MAHATMA GANDHI COLLEGE

IRITTY, KEEZHUR P.O

KANNUR – 670703

A NOVEL APPROACH FOR CLASSIFICATION USING CLUSTERING -

A CASE STUDY ON HEART DISEASE PREDICTION

By

VYSHAK PUTHUSSERI [MG14CCSR11]

SREERAGH M [MG14CCSR28]

BSc(CS) Project Report 2014-2017

Under The Guidance of

Mrs .Reshma P K

CERTIFICATE

Certified that this report titled **" A NOVEL APPROACH FOR CLASSIFICATION USING CLUSTERING - A CASE STUDY ON HEART DISEASE PREDICTION "** is a bonafide record of the project work done by VYSHAK PUTHUSSERI AND SREERAGH M under our supervision and guidance, towards partial fulfillment of the requirement for award of the Degree of B.Sc. Computer Science (2014-2017) of the Kannur University.

Signature of lecturer in charge   Signature of Head of the Dept.

Submitted for the University examination

Examiners:

1.

2.

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show the greatest appreciation to our Head Of the Department Dr.Shijo M Joseph. We can't say thank you enough for his tremendous support and help. We feel motivated and encouraged every time we attend his meeting. Without his encouragement and guidance this project would not have materialized.

Words are boundless to express our sense of gratitude to our guide Mrs.Reshma PK whose valuable guidance and constant encouragement have been of invaluable help to us throughout the course of this project.

We also convey a heartful of thanks to Mr.Rejeesh E for the healthy criticism and Mr.Jithesh K, Mrs.Anupama M for giving support to this project.

The guidance and support received from Mrs.Thulassi, the research scholar under Kannur University was vital for the success of the project. We are grateful for her constant support and help.

DECLARATION

We hereby declare that the project work entitled **“ A NOVEL APPROACH FOR CLASSIFICATION USING CLUSTERING - A CASE STUDY ON HEART DISEASE PREDICTION ”** is a project of the original work done by VYSHAK PUTHUSSERI AND SREERAGH M under the guidance of Mrs.Reshma PK Assistant Professor Mahatma Gandhi College Iritty, towards partial fulfillment of the requirement for award of the Degree of B.Sc. Computer Science (2014-2017) of the Kannur University. To the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Date : Vyshak Puthusseri

Place : Sreeragh M

ABSTRACT

Medical science is one of the extremely fastest growing fields in the present world. The modern medicine systems generate huge amount of data every day. These data without effective computation is totally a waste. To turn these data into useful pattern, mining is a vital task. The medical data mining are useful to produce efficient results on prediction based system in medical oriented fields.

In medical sciences, prediction of Heart disease is identified as one of the most difficult tasks. In India, a main cause of Death is due to Heart Diseases. The deaths due to heart disease in many countries occur due to work overload, mental stress and many other problems. It is found as main reason for death in adults is due to heart disease.

Heart disease is a term for defining a huge amount of healthcare conditions that are related to the heart. This medicinal condition defines the unpredicted health conditions that directly control all the parts of the heart. Different data mining techniques such as association rule mining, classification, clustering etc. are used to predict the heart disease in health care industry.

In this work, the heart disease database is preprocessed to make the mining process more efficient. The preprocessed data is clustered using K-Means to achieve better clustering result. Naïve Bayes classifier algorithm is applied on the clustered data for building the predictive model.

TABLE OF CONTENTS

INTRODUCTION

1.1 Introduction

Data mining is process of extracting useful information from large amount of databases. Data mining is most useful in an exploratory analysis because of nontrivial information in large volumes of data. The data mining techniques are useful for predicting the various diseases in the medical field.

Heart disease is the leading cause of death in the world over the past 10 years. The World Health Organization reported that heart disease is the first leading cause of death in high and low income countries [1]. The European Public Health Alliance reported that heart attacks and other circulatory diseases account for 41% of all deaths [2]. Disease prediction plays an important role in data mining. There are different types of diseases predicting in data mining namely heart diseases, lung cancer and breast cancer. This paper analyzes the heart disease predictions using classification and clustering algorithms. Data mining technology afford an effective approach to latest and indefinite patterns in the data. These hidden patterns can be used for health diagnosis in medicinal data. The information which is identified can be used by the healthcare administrators to get better services. Heart disease is the first leading cause of death in high and low income countries and occurs almost equally in men and women, they continue to rise mainly because preventive measures are inadequate. An estimated over 80% of cardiovascular disease deaths take place in low- and middle-income countries. Tobacco use, an unhealthy diet increases the possibility of heart attacks and strokes. Eating of fruit and vegetables, and limiting the use of salt, also helps to prevent heart attacks and strokes. Heart disease is caused by disorders of the heart and blood vessels. Heart attacks, hypertension,, stroke, peripheral artery disease, rheumatic heart disease and heart failure are also included in this. Use of tobacco and use of alcohol are the major causes of cardio vascular diseases. Three causes of heart diseases are:

- Chest pain
- Stroke
- Heart attack

Data mining techniques like Association Rule Mining, Clustering, and Classification algorithms are used to explore the different kinds of heart based problems. The best-known partitioning clustering algorithm is K-Means algorithm, which is very simple, flexible and straightforward. K-Means clustering algorithm clusters a group of data items into a predefined number of clusters. Clustering process starts with randomly generated initial centroids and keeps reassigning the data objects various clusters based on the similarity between the data object and the cluster centroids until a termination criteria is met (e.g., the fixed number of iterations or stability in movement of data points among clusters). K-Means is the most efficient algorithm in terms of the execution time but it has a drawback that the cluster results are extremely sensitive to the selection of the initial cluster centroids and may converge to the local optimal solution.

It has been proposed to use meta-optimization to improve the processing capabilities of existing classification algorithms. Meta-optimization is an approach which allows using the combination of two or more than two algorithms to achieve a common goal. In current scenario, rather than choosing the training datasets as random, it will be better to choose the samples from the clustered dataset. The results from our experiments indicate that choosing training datasets from a clustered dataset can generate the best compact classification results in comparison with performing the classification algorithm alone.

1.2 Machine Learning

Machine learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include spam filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), search engines and computer vision.

Machine learning is closely related to computational statistics, which also focuses in prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Types of Machine Learning Algorithms:

- Supervised Learning
- Unsupervised Learning

1.2.1. Supervised Learning

Supervised learning is basically a synonym for classification. This type algorithm consists of a target/outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data.

Example:

- k-NN clustering

  The k-nearest neighbor's algorithm (k-NN) is a method for classifying objects based on closest training data in the feature space. k-NN is a type of instance-based learning. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms. But the accuracy of the k-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance.

- Naïve Bayes classifier

  A Naïve Bayes classifier is a term dealing with a simple probabilistic classification based on applying Bayes theorem. In simple terms, a naïve Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

1.2.2 Unsupervised Learning

Unsupervised learning is essentially a synonym for clustering. In this algorithm, we do not have any target or outcome variable to predict / estimate as the input examples are not class labeled. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Example:

- Apriori Algorithm :

  The Apriori algorithm is a seminal algorithm for mining frequent item sets for Boolean association rules. It explores the level-wise mining Apriori property that all nonempty subsets of a frequent itemset must also be frequent. At the k^{th} iteration , it forms frequent k-itemset candidates based on the frequent (k − 1)-itemsets, and scans the database once to find the complete set of frequent k-itemsets,L_k. Variations involving hashing and transaction reduction can be used to make the procedure more efficient. Other variations include partitioning the data (mining on each partition and then combining the results) and sampling the data (mining on a data subset). These variations can reduce the number of data scans required to as little as two or even one.

- K means clustering algorithm :

  A centroid-based partitioning technique uses the centroid of a cluster, C_i, to represent that cluster. Conceptually, the centroid of a cluster is its center point. The centroid can be defined in various ways such as by the mean or medoid of the objects (or points) assigned to the cluster. The difference between an object $p \in C_i$ and c_i, the representative of the cluster, is measured by dist (p, c_i), where dist (x, y) is the Euclidean distance between two points x and y.

1.3. Dataset:

The Cleveland Heart Disease database [3] contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. A total of 270 records with 13 medical attributes (factors) were obtained from the preprocessed Cleveland Heart Disease database [4]. 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. It is assumed that problems such as missing data, inconsistent data, and duplicate data have all been resolved.

The datasets can be visualized in Weka tool as:

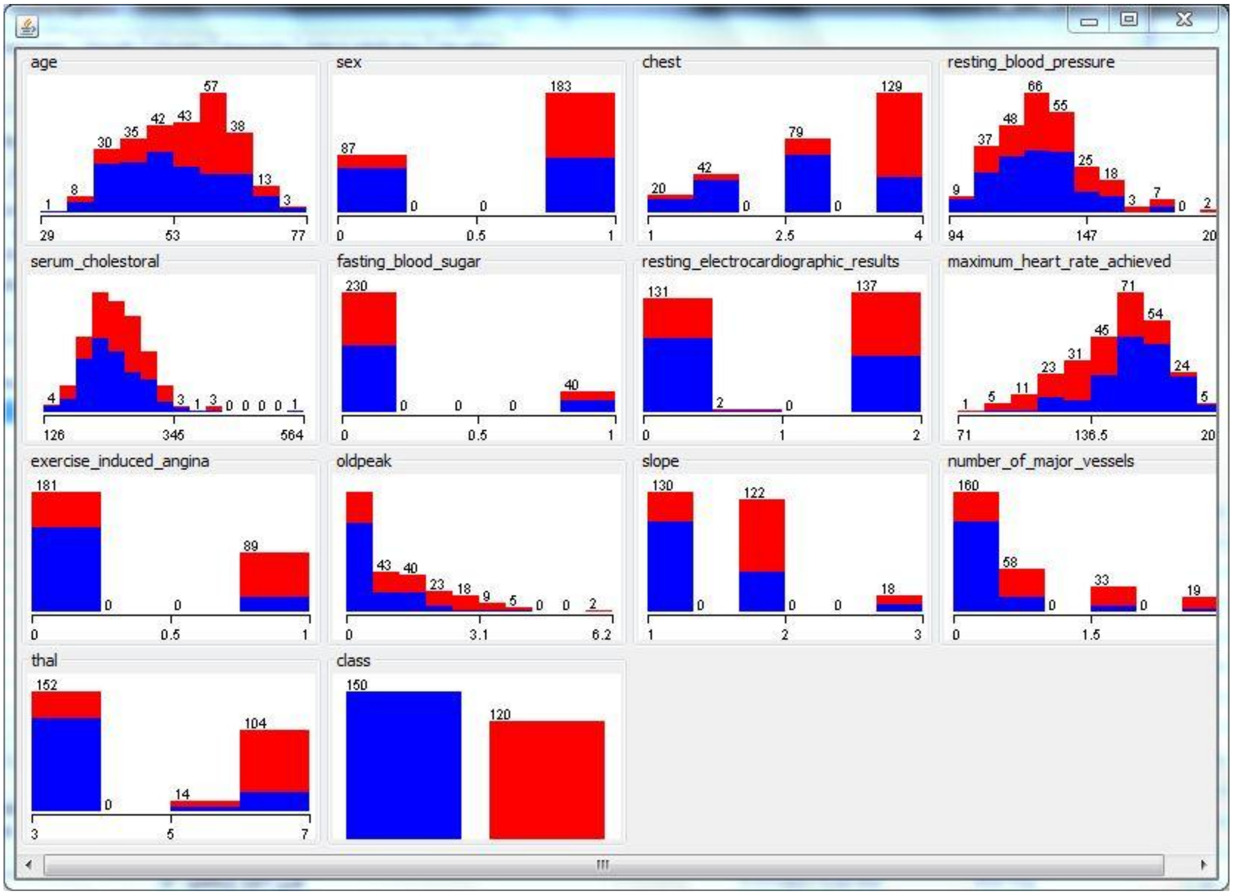


Figure 1: Dataset

1.4 Attributes information.

- Predictable attribute

  Diagnosis (value 0: <50% diameter narrowing (no heart disease;

  value 1: >50% diameter narrowing (has heart disease)

- Input attributes

  1. Age in years
  2. Sex(value 1:male;value 0: female)
  3. chest pain type(
     value 1: typical type 1 angina;
     value 2: typical type angina;
     value 3: non-angina pain;
     value 4: asymptomatic)

  4. resting blood pressure
  5. serum cholestoral in mg/dl
  6. fasting blood sugar > 120 mg/dl
  7. electrocardiographic results(
     value 0: normal;
     value 1: having ST-T wave abnormality;
     value 2: showing definite left ventricular hypertrophy)

  8. maximum heart rate achieved
  9. exercise induced angina(
     value 1: yes;
     value 0: no;

  10. oldpeak = ST depression induced by exercise relative to rest
  11. the slope of the peak exercise ST segment(
      value 1: unsloping;
      value 2: flat;
      value 3: downsloping;

  12. number of major vessels (0-3) colored by flourosopy
  13. thal(
      value 3 : normal;
      value 6 : fixed defect;
      value 7 : reversable defect

1.4.1 Sample Datasets:

Age	Sex	Chest Pain	BP	Cholestoral	Blood Sugar	ECG	Max. Heart Rate	Exercise Induced Angina	Oldpeak	Slope	No. of major vessels	Thal	Class Label
70	1	4	130	322	0	2	109	0	2.4	2	3	3	1
56	1	3	130	256	1	2	142	1	0.6	2	1	6	1
60	1	4	140	293	0	2	170	0	1.2	2	2	7	1
53	1	4	142	226	0	2	111	1	0	1	0	7	0
44	1	3	140	235	0	2	180	0	0	1	0	3	0
61	1	1	134	234	0	0	145	0	2.6	2	2	3	1
46	1	4	140	311	0	0	120	1	1.8	2	2	7	1
44	1	3	130	233	0	0	179	1	0.4	1	0	3	0
59	1	3	126	218	1	0	134	0	2.2	2	1	6	1
65	0	4	150	225	0	2	114	0	1	2	3	7	1

Table 1:Sample Datasets used for prediction.

LITERATURE SURVEY

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. There are number of factors which increase the risk of Heart disease [5].

- Family history of heart disease
- Smoking
- Cholesterol
- Poor diet
- High blood pressure
- High blood cholesterol
- Obesity
- Physical inactivity
- Hyper tension

Different types of studies have been done to focus on prediction of heart disease. Various data mining techniques are used for diagnosis and achieved different accuracy level for different methods [6].Association rule mining, a computational intelligence approach is used to identify the factors that contribute to heart disease and UCI Cleveland data set, a biological data base is considered along with the rule generation algorithm – Apriori [7].

KNN is a non-parametric method which is used for classification and regression. Compared to other machine learning algorithm KNN is the simplest algorithm. This algorithm consist K-closet training examples in the feature space. In this algorithm K is a user defined constant. The test data are classified by assigning a constant value which is most chronic among the K-training samples nearest to the point. Literature shows the KNN has the strong consistency result. Decision tree builds classification models in the form of a tree structure. It breaks the dataset into smaller subset while at the same time an associated decision tree is incrementally developed. The decision tree uses a top-down approach method. The root of the decision tree is the data set and the leaf is the subset of the data set. The risk level of heart disease prediction through hybrid algorithm has been proposed by Shovon K.P ramanik.et al [8]. Hybrid Algorithm is the combination of KNN algorithm and ID3. These algorithms are used for heart disease prediction. The KNN algorithm is used to preprocess the data; it is called as preprocessed algorithm. The preprocessed data are considered as training set and then the data has been classified into a tree structure. The ID3 algorithm is applied for the classifier to predict the heart disease. The incorrect values are classified through KNN Algorithm.

The Naive Bayes classifier algorithm uses conditional independence; it believes that an attribute value of a given class is independent of the values of other attributes. Web based health care detection was proposed by S.Indhumathi.et al [9] has suggested a prediction of high risk heart disease using a Naïve Bayes algorithm. The preprocessed data has been considered as the training set. Two phase namely classification and prediction was discussed in that work. Preprocessing is done in the classification phase. The preprocessing includes cleaning of data, normalization and reduction of data, etc. In the prediction phase the disease types are classified and predicted, i.e. a training set is formed based on the disease type and the test set is formed based on the questions.

ANN, often just called a "neural network", is a mathematical model or computational model used for a biological purpose. In other words, it is an emulation of biological neural system. The prediction method for heart disease using Neural Network has been proposed by ChaitraliS.Dangare.et al [10]. It has mainly three layers, i.e. the input layer, hidden layer and the output layer. The input is given to the input layer and the result is obtained in the output layer. Then the actual output and the expected output are compared. The back propagation has been applied to find the error and to adjust the weight between the output and the previous hidden layers. Once, the back propagation is completed, and then the forward process is started and continued until the error is minimized.

TOOLS

3.1 Algorithms:

3.1.1 *k*-NN Algorithm:

In pattern recognition, the *k*-nearest neighbor's algorithm (*k*-NN) is a non-parametric method used for classification. The input consists of the *k* closest training examples in the feature space. *K-NN* is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms. The neighbors are taken from a set of objects for which the class (for *k*-NN classification) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The optimal choice of the value k is highly data-dependent: in general a larger k suppresses the effects of noise, but makes the classification boundaries less distinct. A shortcoming of the *k*-NN algorithm is that it is sensitive to the local structure of the data.

3.1.2. Naive Bayes Classifier:

- 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.P(H), P(X|H), and P(X) may be estimated from the given data, 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).

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Equation 1: Bayes' theorem

- Naive Bayesian Classification:

i. 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 = (x_1, x_2,...,x_n), depicting n measurements made on the tuple from n attributes, respectively,A_1, A_2,..., A_n .

ii. Suppose that there are m classes, C_1, C_2,..., C_m. 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 Bayesian classifier predicts that tuple X belongs to the class C_i if and only if C_i .

$$P(Ci|X) > P(Cj|X)\ for\ 1 \leq j \leq m\ , j \neq i$$

Thus, we maximize P(C_i|X). The class C_i for which P(C_i |X) is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(Ci|X) = \frac{P(X|Ci)P(Ci)}{P(X)}$$

Equation 2: maximum posteriori hypothesis

iii. As P(X) is constant for all classes, only P(X|C_i)P(C_i) 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|C_i). Otherwise, we maximize P(X|C_i)P(C_i). Note that the class prior probabilities may be estimated by P(C_i) = |C_i,D|/|D|, where |C_i,D| is the number of training tuples of class C_i in D .

iv. Given data sets with many attributes, it would be extremely computationally expensive to compute P(X|C_i). To reduce computation in evaluating P(X|C_i), the naive 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,

$$P(X|Ci) = \prod_{k=1}^{n} P(Xk|Ci)$$

$$= P(x1|Ci) \times P(x2|Ci) \times \ldots\ldots \times P(xn|Ci)$$

We can easily estimate the probabilities P(x_1|C_i), P(x_2|C_i),..., P(x_n|C_i) from the training tuples. Here x_k refers to the value of attribute A_k for tuple X. For each attribute, we look at whether the attribute is categorical or continuous-valued. For instance, to compute P(X|C_i), we consider the following:

a) If A_k is categorical, then P(x_k |C_i) is the number of tuples of class C_i in D having the value x_k for A_k , divided by |C_i,D|, the number of tuples of class C_i in D.
b) If A_k 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, defined by

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \mathrm{e}^{\frac{-(x-\mu)x^2}{2\sigma x^2}}$$

so that

$$P(X_k|C_{i)} = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

We need to compute$\mu_{C_i} and\ \sigma_{C_i}$,which are the mean (i.e., average) and standard deviation, respectively, of the values of attribute A_k for training tuples of class C_i.

v. To predict the class label of X, P(X|C_i)P(C_i) is evaluated for each class C_i. The classifier predicts that the class label of tuple X is the class C_i if and only if,

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad for\ 1 \leq j \leq m, j \neq i$$

In other words, the predicted class label is the class Ci for which P(X|C_i)P(C_i) is the maximum.

3.2 Software Tools:

3.2.1 Flask Framework

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed. The latest stable version of Flask is 0.12 as of December 2016. Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself.

Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation and upload handling, various open authentication technologies and several common framework related tools.

- Jinja(Template Engine)

Jinja is a template engine for the Python programming language and is licensed under a BSD License created by Armin Ronacher. It is similar to the Django template engine but provides Python-like expressions while ensuring that the templates are evaluated in a sandbox. It is a text-based template language and thus can be used to generate any markup as well as source code. The Jinja template engine allows customization of tags, filters, tests, and globals. Also, unlike the Django template engine, Jinja allows the template designer to call functions with arguments on objects. Jinja is Flask's default template engine.

3.2.2 Python **:**

Python is a widely used high-level programming language for general purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems.

➢ Features and philosophy*:*

Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and many language features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)).Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a mix of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution. The design of Python offers some support for functional programming in the Lisp tradition. The core philosophy of the language is summarized by the document *The Zen of Python* (*PEP 20*), which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Rather than requiring all desired functionality to be built into the language's core, Python was designed to be highly extensible. Python can also be embedded in existing applications that need a programmable interface. This design of a small core language with a large standard library and an easily extensible interpreter was intended by Van Rossum from the start because of his frustrations with ABC, which espoused the opposite mind set. While offering choice in coding methodology, the Python philosophy rejects exuberant syntax, in favor of organized grammar. As Alex Martelli put it: "To describe something as clever is *not* considered a compliment in the Python culture." Python's philosophy is in favor of "there should be one—and preferably only one—obvious way to do it".

When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or try using PyPy, a just-in-time compiler. An important goal of Python's developers is to make the program enjoyable for the user.

- Syntax and semantics:

Python is intended to be a highly readable language. It is designed to have an uncluttered visual layout, often using English keywords where other languages use punctuation. Further, Python has fewer syntactic exceptions and special cases than C or Pascal.

- Indentation:

Python uses whitespace indentation to delimit blocks – rather than curly braces or keywords. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. This feature is also sometimes termed the off-side rule.

3.2.3 R Programming Language:

R is an open source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis .R is a GNU package. The source code for the R software environment is written primarily in C, FORTRAN, and R. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. While R has a command line interface, there are several graphical front-ends available.

- Programming Features:

R is an interpreted language; users typically access it through a command-line interpreter. Like other similar languages like MATLAB, R supports matrix arithmetic. R's data structures include vectors, matrices, arrays, data frames (similar to tables in a relational database) and lists. R's extensible object system includes objects for (among others): regression models, time-series and geo-spatial coordinates. The scalar data type was never a data structure of R. Instead, a scalar is represented as a vector with length one. R supports procedural programming with functions and, for some functions, object-oriented programming with generic functions. A generic function acts differently depending on the classes of arguments passed to it. In other words, the generic function dispatches the function (method) specific to that class of object. For example, R has a generic print function that can print almost every class of object in R with simple print(object_name) syntax. Although used mainly by statisticians and other practitioners requiring an environment for statistical computation and software development, R can also operate as a general matrix calculation toolbox – with performance benchmarks comparable to GNU Octave or MATLAB.

PROPOSED SYSTEM

4.1 Prediction using pattern matching:

Medical diagnosis is an on-going research in medical trade. The prediction of the disease is done by a doctor by the pattern matching technique. That is, by finding the match for the symptoms that had seen in the patients with the knowledge and experience he or she had acquired. The same concept can be used to develop a system that can perform the disease prediction duty. The method still requires the doctor's information and experience due to alternative factors starting from medical records to weather conditions, atmosphere, blood pressure and numerous alternative factors. The huge numbers of variables are consider as entire variables that are required to understand the complete working process itself, however no model has analysed successfully. Medical decision could be extremely specialized and difficult job due to alternative factors or incase of rare diseases. The coming era of machine learning can solve this problem which the computers can think and make decision of its own. This method needs a process of elimination or obtaining information that shrinks the probability of candidate conditions to negligible levels. It contains four steps:

- The doctor gather all information about the patients and create a symptoms list.
- The doctor should make a list of all possible causes of symptoms.
- The doctor should prioritize the list by which is the most dangerous possible cause of symptoms put in the top of the list.
- The doctor should rule out or treat the possible causes beginning with the most urgently dangerous conditions.

4.1.1 Results :

If we give the symptoms as follows:

“ Good morning doctor. Sir I have been suffering from fever since yesterday. And also feeling headache and shivering. I’m unable to sleep at nights. ”

The result will be as follows.

“I seems you are having fever.”

4.2 Prediction of heart disease using *k*-NN classifier:

Some diseases like cancer, diabetes, heart disease requires some specific values for identification like blood pressure, blood sugar etc. K-Nearest-Neighbour is one of the most widely used data mining techniques in classification problems. Its simplicity and relatively high convergence speed make it a popular choice. However a main disadvantage of KNN classifiers is the large memory requirement needed to store the whole sample. When the sample is large, response time on a sequential computer is also large. Despite the memory requirement issue, it is showing good performance in classification problems of various datasets.

The algorithm can be viewed as the following steps:

- Handle the data in such a manner that algorithm can use it for the calculation process.
- Calculate the distance between two data instances.
- Locate k most similar data instances.
- Generate a response from a set of data instances.
- Summarize the accuracy of predictions

To calculate the distance between two data instance, Euclidean distance measure can be used. This is defined as the square root of the sum of the squared differences between the two arrays of numbers.

4.2.1 Results:

The following dataset, was applied to the k-NN algorithm.

Age	Sex	Chest Pain	BP	Cholestoral	Blood Sugar	ECG	Max. Heart Rate	Exercise Induced Angina	Oldpeak	Slope	No. of major vessels	Thal	Class Label
70	1	4	130	322	0	2	109	0	2.4	2	3	3	1
44	1	3	140	235	0	2	180	0	0	1	0	3	0

The result was:

Accuracy	Prediction Result
62.222222	Heart Disease
66.666666	Not Heart Disease

4.3 Prediction of heart disease using Naïve Bayes Classifier:

Although *k*-NN is a good classification algorithm, it has some disadvantages which cannot be negligible. As the datasets become large it takes too much of response time and considerable memory location. Naive Bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method. The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability. Naive bases is often described using categorical data because it is easy to describe and calculate using ratios. A more useful version of the algorithm for our purposes supports numeric attributes and assumes the values of each numerical attribute are normally distributed (fall somewhere on a bell curve). Again, this is a strong assumption, but still gives robust results.

The algorithm can be viewed as the following steps:

- Handle the data in such a manner that algorithm can use it for the calculation process.
- Summarize the properties in the training dataset so that we can calculate probabilities and make predictions.
- Use the summaries of the dataset to generate a prediction.
- Summarize the accuracy of predictions.

4.3.1 Results:

Age	Sex	Chest Pain	BP	Cholestoral	Blood Sugar	ECG	Max. Heart Rate	Exercise Induced Angina	Oldpeak	Slope	No. of major vessels	Thal	Class Label
70	1	4	130	322	0	2	109	0	2.4	2	3	3	1
44	1	3	140	235	0	2	180	0	0	1	0	3	0

The following dataset, was applied to the Naïve Bayes Classifier algorithm.

The result was:

Accuracy	Prediction Result
85.555555	Heart Disease
86.666666	Not Heart Disease

REFINED SYSTEM

5.1 Data Pre-processing:

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format which can be used for producing the knowledge.

5.1.1 Why preprocessing?

1. Real world data are generally
   - Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
   - Noisy: containing errors or outliers
   - Inconsistent: containing discrepancies in codes or names
2. Tasks in data preprocessing
   - Data cleaning: fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
   - Data integration: using multiple databases, data cubes, or files.
   - Data transformation: normalization and aggregation.
   - Data reduction: reducing the volume but producing the same or similar analytical results.
   - Data discretization: part of data reduction, replacing numerical attributes with nominal ones.

To improve the accuracy of the predictive model, it was suggested to cluster the datasets such that selection of training and test set cab be chosen from each cluster uniformly. This can overcome the problem with the model having the worst case. The worst case is that the model is built only over a particular type of data. Which can produced a biased result for the prediction

5.1.2 K-Means Clustering:

K-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the *k*-means algorithm; it is also referred to as Lloyd's algorithm, particularly in the computer science community.

Given an initial set of *k* means $m_1,\ldots,m_k$, the algorithm proceeds by alternating between two steps:

- Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares (WCSS). Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean. Each data point is assigned to exactly one cluster, even if it could be assigned to two or more of them.
- Update step: Calculate the new means to be the centroids of the observations in the new clusters. Since the arithmetic mean is a least-square estimator, this also minimize the within-cluster sum of squares objective.

5.2 Refined Naïve Bayes Classifier over clustered datasets:

The datasets are clustered using k-means clustering algorithm to produce two clusters which contains similar datasets. Naïve Bayes Classifier model was applied over the clustered datasets to produce a training set which cannot lead to a biased prediction result. The accuracy was high when the model was built upon a clustered dataset.

4.2.1 Results:

The following dataset, was applied to the refined algorithm.

Age	Sex	Chest Pain	BP	Cholestoral	Blood Sugar	ECG	Max. Heart Rate	Exercise Induced Angina	Oldpeak	Slope	No. of major vessels	Thal	Class Label
70	1	4	130	322	0	2	109	0	2.4	2	3	3	1
44	1	3	140	235	0	2	180	0	0	1	0	3	0

The result was:

Accuracy	Prediction Result
91.111111	Heart Disease
92.222222	Not Heart Disease

CONCLUSION & FUTURE WORKS

6.1 Conclusion:

Data mining in health care management is different from the other fields owing to the fact that the data present are heterogeneous and that certain ethical, legal, and social constraints apply to private medical information. Health care related data are voluminous in nature and they arrive from diverse sources all of them not entirely appropriate in structure or quality. These days, the exploitation of knowledge and experience of numerous specialists and clinical screening data of patients gathered in a database during the diagnosis procedure, has been widely recognized.

Naive Bayes algorithm is the optimum algorithm as the compact time for processing dataset and shows better performance in accuracy prediction.

As a way to validate the proposed method, we have tested with emphasis on heart disease using machine learning data sets taken from UCI repository.

In many papers, the datasets are randomly selected which degrades the predictive capabilities of the model.

So, to overcome such situations, we are first clustering the datasets into two clusters which help the predictive model having high accuracy as the problem with selection from a class imbalanced datasets solves up to an extent.

6.2 Result Analysis:

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

The most basic terms used for confusion matrix:

- True Positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
- True Negatives (TN): We predicted no, and they don't have the disease.
- False Positives (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- False Negatives (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

- ➢ Confusion matrix for k-NN:

n=270	Predicted No	Predicted Yes
Actual No.	TN = 99	FP = 51
Actual Yes	FN =46	TP =74

- ➢ Confusion matrix for Naïve Bayes Classifier:

n=270	Predicted No	Predicted Yes
Actual No.	TN = 129	FP = 21
Actual Yes	FN =18	TP =102

- ➢ Confusion matrix for refined algorithm:

n=270	Predicted No	Predicted Yes
Actual No.	TN = 136	FP = 14

Actual Yes	FN =11	TP =109

6.3 Table of Comparison:

We have used three methods for heart disease prediction and arrived at an inference that the accuracy of the classification model can be increased if the datasets were selected from two clusters. The result of the predictive model using three methods can be summarized as follows:

Algorithm	Accuracy	Prediction Result
K NN	62.222222	Heart Disease
Naïve Bayes Classifier	85.555555	Heart Disease
Classification using Clustering	91.111111	Heart Disease

6.4 Future Work:

As data mining is an area which was going on developing, day by day new methods are emerging. In the case of Heart disease prediction also new methods are developing.

As a future work, we are planning to propose a new algorithm using decision tree as it can provide a high accuracy in prediction. The learning and classification steps of decision tree induction are simple and fast. However, successful use may depend on the data at hand also the tree building process is a computationally expensive task. There may be a chance for the classifier tends to over fit the data i.e. during learning it may incorporate some particular anomalies of the training data that are not present in the general dataset overall. So first we can generate the association rules among datasets which can be used as splitting rules.

But the complexity of association rule generation can leads to the over fit the data. This problem can be overcome by clustering the datasets and association rules are mined from each cluster. The clustering procedure is highly dependent to the initial centroid values. So to overcome such situation we can use an optimization algorithm which results in the optimum initial clusters that can increase the computational complexity of clustering.

BIBLIOGRAPHY

[1] World Health Organization. Factsheet (July 2007-Febuary 2011).[Online]. Available: http://www.who.int/mediacentre/factsheets/fs310.pdf

[2] European Public Health Alliance. (July 2010-Febuary 2011).[Online]. Available: http://www.epha.org/a/2352

[3]David W. Aha "UCI Machine Learning Databases", https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[4] Preprocessed Cleveland Heart Disease datasets. https://github.com/renatopp/arff-datasets/blob/master/classification/heart.statlog.arff

[5] Yanwei, X.; Wang, J.; Zhao, Z.; Gao, Y., "Combination data mining models with new medical data to predict outcome of coronary heart disease". Proceedings International Conference on Convergence Information Technology 2007, pp. 868 – 872

[6] Miss. Chaitrali S. Dangare, Dr. Mrs. Sulabha S. Apte, "Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques", International Journal of Computer Applications (0975 – 888), Volume 47– No.10, pp.44-48, June 2012.

[7] Ibrahim Umar Said1 , AbdullahiHaruna Adam , Dr. Ahmed BaitaGarko,"ASSOCIATION RULE MINING ON MEDICAL DATA TO PREDICT HEART DISEASE" .International Journal of Science Technology and Management Vol.No.4,Issue 08,August 2015

[8] Shovon K. Pramanik, SubrataPramanik, Bimal K. Pramanik, M. K. Islam Molla and Md. Ekramul Hamid, "Hybrid Classification Algorithm for Knowledge Acquisition of Biomedical Data", International Journal of Advanced Science and Technology, Vol. 44, July, 2012

[9] S.Indhumathi, Mr.G.Vijaybaskar, "Web based health care detection using naïve Bayes algorithm", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 4 Issue 9, pp.3532-36, September 2015.

[10] Miss. Chaitrali S. Dangare, Dr. Mrs. Sulabha S. Apte, "A Data mining approach for prediction of heart disease using neural network's", International Journal of Computer Engineering & Technology(IJCET)), Volume 3, Issue 3, October - December (2012), pp. 30-40...

[11] JiaweiHan, MichelineKamber, JianPei "Data Mining Concepts And Techniques", [Online] Available:http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

[12] Thomas A. Powell "HTML: The Complete Reference, Second Edition" [Online] Available http://www.alpcentauri.info/html_complete_reference.pdf

[13] Miguel Grinberg Flask Web Development https://doc.lagout.org/programmation/python/Flask%20Web%20Development_%20Developing%20Web%20Applications%20with%20Python%20%5BGrinberg%202014-05-18%5D.pdf

APPENDIX A :SOURCE CODE

- R Script

```
h=read.csv("h.csv")
View(h)
a<-kmeans(h,2)
aa<-a$cluster
write.csv(aa,file="a.csv")
```

- Python Script

```
import re
import csv
import random
import math
import operator
from flask import Flask, render_template, request
app = Flask(__name__)
@app.route("/")
def main():
    return render_template('index.html')
@app.route("/home")
def main1():
    return render_template('index.html')
@app.route("/about")
def abt():
    return render_template('aboutus.html')
@app.route("/diseasePrediction")
def diseasePre():
    return render_template('diseasePrediction.html')
@app.route("/heartDiseasePrediction")
```

```
def heartDiseasePre():
        return render_template('heartDisease.html')

@app.route("/diseasePredictionSubmit",methods=['POST'])
def usingPatternMatching():
        s = request.form['s']
        s=s.lower()
        dnam = request.form['dname']
        stopwords=["i","are","am","but","that","my","and","to","have","also"]
        sympt=['fever','headache','stomachache',"eyepain","swetting"]
        '''Removimg stopwords from the sentence.'''
        for i in stopwords:
            pattern=r'\b'+i+r'\b'
            s=re.sub(pattern,'',s)
        '''Splitting the string into words'''
        l=[]
        s=s.split()
        for i in s:
            l.append([i,0])
        '''Changeing to dictionary'''
        di=dict(l)
        orr={}
        a=[]
        '''Finding keywords'''
        for i in di.iterkeys():
            for j in sympt:
                if i==j:
                    orr[i]=di[i]
        '''Changeing the keyword into a list '''
        for i in orr.iterkeys():
            a.append(i)
        orr=a
        '''Our disease database'''

do={'flue':['fever','headache','stomachache'],'myophia':['headache','eyepain'
],'hungry':['headache','stomachache','swetting']}
        '''For identifying which disease have max sympt'''
        disease={'flue':0,'myophia':0,'hungry':0}
        '''Counting the no of symptoms in every disease in database'''
        for i in do.iterkeys():
            for j in do[i]:
                for k in orr:
                    if k==j:
                        disease[i]=disease[i]+1
        m=0
        mk=" "
        '''Finding the disease with max number of sympt matched.'''
        for i in disease.iterkeys():
            if m<disease[i]:
                m=disease[i]
                mk=i
        print "Disease is "+mk
        return
render_template('diseasePrediction.html',resultantDisease=mk,rname=dn
am)

@app.route('/heartDisease',methods=['POST'])
def indexHeart():
        testset=[]
        r = request.form['nam']
        #Converting the datas into a uniform type
        age = float(int(request.form['age']))
```

```
    sex = float(request.form['sex'])
    chestPain = float(request.form['chestPain'])
    restingBP = float(request.form['restingBP'])
    cholestoral = float(request.form['cholestoral'])
    sugar = float(request.form['sugar'])
    electroCardiographic = float(request.form['electroCardiographic'])
    maximumHeartRate = float(request.form['maximumHeartRate'])
    angina = float(request.form['angina'])
    oldpeak = float(request.form['oldpeak'])
    slope = float(request.form['slope'])
    majorVessels = float(request.form['majorVessels'])
    thal = float(request.form['thal'])
    testset.extend((age ,sex ,chestPain ,restingBP ,cholestoral ,sugar
,electroCardiographic ,maximumHeartRate ,angina ,oldpeak ,slope
,majorVessels ,thal))
    nreturnValue=m(testset)
    kreturnValue=n(testset)
    creturnValue=o(testset)
    if nreturnValue[1]==1.0:
        string1 = 'Heart Disease'
    else:
        print "return value="
        string1 = 'Not Heart Disease'
    if kreturnValue[1]==1.0:
        string2 = 'Heart Disease'
    else:
        print "return value="
        string2 = 'Not Heart Disease'
    if creturnValue[1]==1.0:
        string3 = 'Heart Disease'
    else:
```

```
                print "return value="
                string3 = 'Not Heart Disease'
            return
render_template('heartDisease.html',rname=r,knnaccuracy=kreturnValue
[0],caccuracy=creturnValue[0],naccuracy=nreturnValue[0],knnresult=stri
ng2,nresult=string1,cresult=string3)

def loadCsv(filename):
        lines = csv.reader(open(filename, "rb"))
        dataset = list(lines)
        for i in range(len(dataset)):
                dataset[i] = [float(x) for x in dataset[i]]
        return dataset

def splitDataset(dataset, splitRatio):
        trainSize = int(len(dataset) * splitRatio)
        trainSet = []
        copy = list(dataset)
        while len(trainSet) < trainSize:
                index = random.randrange(len(copy))
                trainSet.append(copy.pop(index))
        return [trainSet, copy]

def separateByClass(dataset):
        separated = {}
        for i in range(len(dataset)):
                vector = dataset[i]
                if (vector[-1] not in separated):
                        separated[vector[-1]] = []
                separated[vector[-1]].append(vector)
        return separated
```

```
def mean(numbers):
    try:
        return sum(numbers)/float(len(numbers))
    except TypeError:
        numbers=list(numbers)

        #numbers=[int(i) for i in numbers]
        s=0
        s=sum(numbers)
        return s/len(numbers)

def stdev(numbers):
    numbers=[float(i) for i in numbers]
    avg = mean(numbers)
    variance = sum([pow(x-avg,2) for x in numbers])/float(len(numbers)-
1)
    return math.sqrt(variance)

def summarize(dataset):
    summaries = [(mean(attribute), stdev(attribute)) for attribute in
zip(*dataset)]
    del summaries[-1]
    return summaries

def summarizeByClass(dataset):
    separated = separateByClass(dataset)
    summaries = {}
    for classValue, instances in separated.iteritems():
        summaries[classValue] = summarize(instances)
    return summaries
```

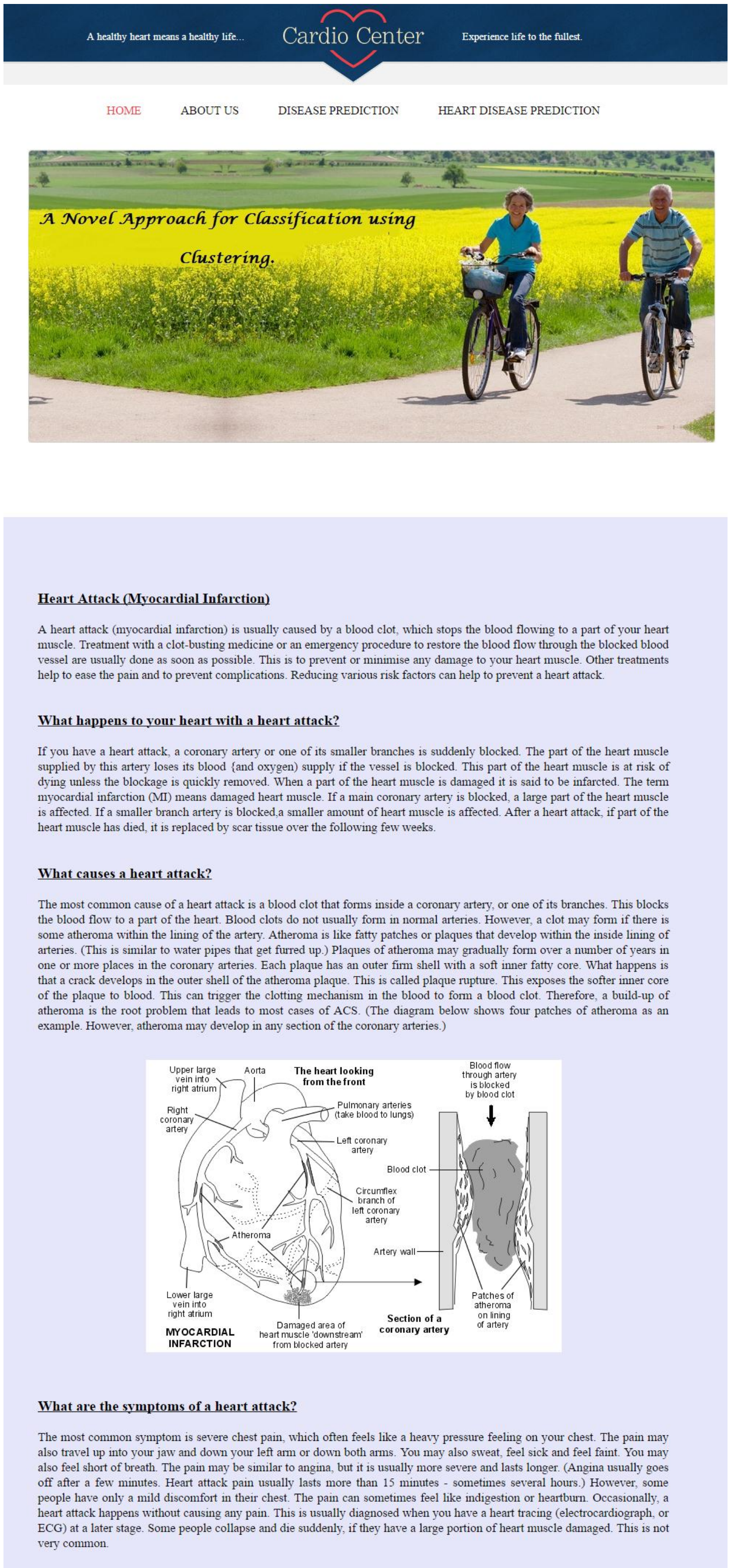
```
def calculateProbability(x, mean, stdev):

    exponent = math.exp(-(math.pow(x-mean,2)/(2*math.pow(stdev,2))))
    return (1 / (math.sqrt(2*math.pi) * stdev)) * exponent

def calculateClassProbabilities(summaries, inputVector):
    probabilities = {}
    for classValue, classSummaries in summaries.iteritems():
        probabilities[classValue] = 1
        for i in range(len(classSummaries)):
            mean, stdev = classSummaries[i]
            try:
                x = inputVector[i]
            except IndexError:
                pass
            probabilities[classValue] *= calculateProbability(x, mean, stdev)
    return probabilities

def predict(summaries, inputVector):
    probabilities = calculateClassProbabilities(summaries, inputVector)
    bestLabel, bestProb = None, -1
    for classValue, probability in probabilities.iteritems():
        if bestLabel is None or probability > bestProb:
            bestProb = probability
            bestLabel = classValue
    return bestLabel

def getPredictions(summaries, testSet):
    predictions = []
    for i in range(len(testSet)):
        result = predict(summaries, testSet[i])
        predictions.append(result)
    return predictions

def getAccuracy(testSet, predictions):
    correct = 0
    for i in range(len(testSet)):
        if testSet[i][-1] == predictions[i]:
            correct += 1
    return (correct/float(len(testSet))) * 100.0

def euclideanDistance(instance1, instance2, length):
    distance = 0
    for x in range(length):
        distance += pow((instance1[x] - instance2[x]), 2)
    return math.sqrt(distance)

def getNeighbors(trainingSet, testInstance, k):
    distances = []
    length = len(testInstance)-1
    for x in range(len(trainingSet)):
        dist = euclideanDistance(testInstance, trainingSet[x], length)
        distances.append((trainingSet[x], dist))
    distances.sort(key=operator.itemgetter(1))
    neighbors = []
    for x in range(k):
        neighbors.append(distances[x][0])
    return neighbors
```

```
def getResponse(neighbors):
    classVotes = {}
    for x in range(len(neighbors)):
        response = neighbors[x][-1]
        if response in classVotes:
            classVotes[response] += 1
        else:
            classVotes[response] = 1
    sortedVotes = sorted(classVotes.iteritems(), key=operator.itemgetter(1), reverse=True)
    return sortedVotes[0][0]

def m(testset):
    returnValue=[]
    filename = '/home/projectvyshu/project/h.csv'
    splitRatio = 0.67
    dataset = loadCsv(filename)
    trainingSet, testSet = splitDataset(dataset, splitRatio)
    print('Split {0} rows into train={1} and test={2} rows').format(len(dataset), len(trainingSet), len(testSet))
    # prepare model
    summaries = summarizeByClass(trainingSet)
    # test model
    predictions = getPredictions(summaries, testSet)
    accuracy = getAccuracy(testSet, predictions)
    result = predict(summaries, testset)
    returnValue.append(accuracy)
    returnValue.append(result)
    print('Accuracy: {0}%').format(accuracy)
    return returnValue
```

```
def o(testset):
    returnValue=[]
    #clustered data
    filename = '/home/projectvyshu/project/a.csv'
    a=loadCsv(filename)
    #main data
    filename = '/home/projectvyshu/project/h.csv'
    lines = csv.reader(open(filename, "rb"))
    data = list(lines)
    for i in range(len(data)):
        data[i] = [float(x) for x in data[i]]
        data[i].append(a[i][1]-1)
    s = separateByClass(data)
    count=True
    for key,values in s.items():
        if count:
            a=values
            count=False
        b=values
    trainingSet1, testSet1 = splitDataset(a, 0.67)
    trainingSet, testSet = splitDataset(b, 0.67)
    trainingSet.extend(trainingSet1)
    testSet.extend(testSet1)
    summaries = summarizeByClass(trainingSet)
    # test model
    predictions = getPredictions(summaries, testSet)
    accuracy = getAccuracy(testSet, predictions)
    result = predict(summaries, testset)
    returnValue.append(accuracy)
    returnValue.append(result)
    print('Accuracy:with clustering {0}%').format(accuracy)
```

```
       return returnValue

def n(testset):
        returnValue=[]
        filename = '/home/projectvyshu/project/h.csv'
        splitRatio = 0.67
        dataset = loadCsv(filename)
        trainingSet, testSet = splitDataset(dataset, splitRatio)
        # generate predictions
        predictions=[]
        k = 3
        for x in range(len(testSet)):
                neighbors = getNeighbors(trainingSet, testSet[x], k)
                result = getResponse(neighbors)
                predictions.append(result)
        accuracy = getAccuracy(testSet, predictions)
        neighbors = getNeighbors(dataset, testset, k)
        result = getResponse(neighbors)
        returnValue.append(accuracy)
        returnValue.append(result)
        print('Accuracy: {0}%').format(accuracy)
        return returnValue
if __name__ == "__main__":
       app.run()
```

APPENDIX B: SCREENSHOTS



A healthy heart means a healthy life...
Cardio Center
Experience life to the fullest.
HOME
ABOUT US
DISEASE PREDICTION
HEART DISEASE PREDICTION
A Novel Approach for Classification using
Clustering.

Heart Attack (Myocardial Infarction)

A heart attack (myocardial infarction) is usually caused by a blood clot, which stops the blood flowing to a part of your heart muscle. Treatment with a clot-busting medicine or an emergency procedure to restore the blood flow through the blocked blood vessel are usually done as soon as possible. This is to prevent or minimise any damage to your heart muscle. Other treatments help to ease the pain and to prevent complications. Reducing various risk factors can help to prevent a heart attack.

What happens to your heart with a heart attack?

If you have a heart attack, a coronary artery or one of its smaller branches is suddenly blocked. The part of the heart muscle supplied by this artery loses its blood {and oxygen) supply if the vessel is blocked. This part of the heart muscle is at risk of dying unless the blockage is quickly removed. When a part of the heart muscle is damaged it is said to be infarcted. The term myocardial infarction (MI) means damaged heart muscle. If a main coronary artery is blocked, a large part of the heart muscle is affected. If a smaller branch artery is blocked,a smaller amount of heart muscle is affected. After a heart attack, if part of the heart muscle has died, it is replaced by scar tissue over the following few weeks.

What causes a heart attack?

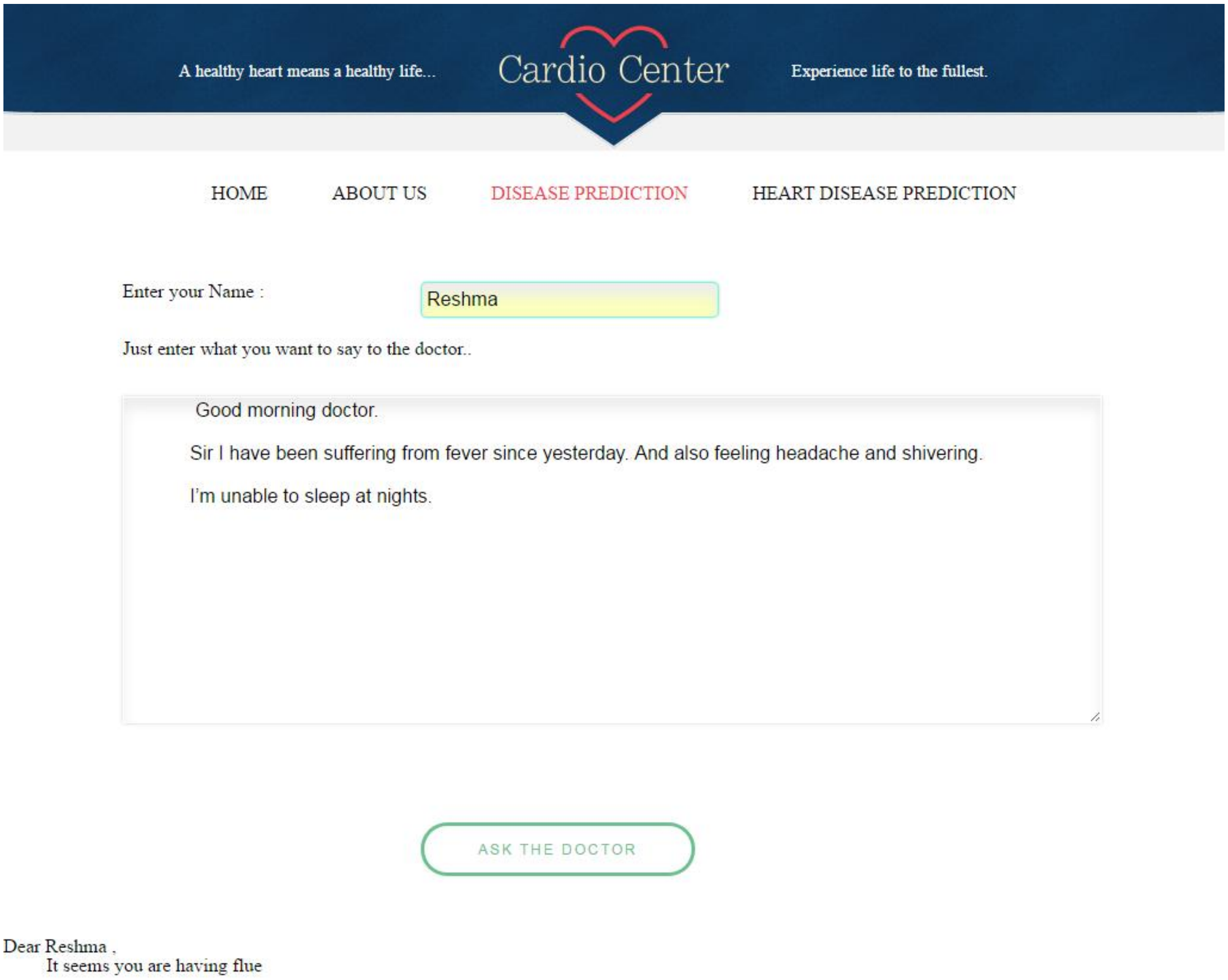
The most common cause of a heart attack is a blood clot that forms inside a coronary artery, or one of its branches. This blocks the blood flow to a part of the heart. Blood clots do not usually form in normal arteries. However, a clot may form if there is some atheroma within the lining of the artery. Atheroma is like fatty patches or plaques that develop within the inside lining of arteries. (This is similar to water pipes that get furred up.) Plaques of atheroma may gradually form over a number of years in one or more places in the coronary arteries. Each plaque has an outer firm shell with a soft inner fatty core. What happens is that a crack develops in the outer shell of the atheroma plaque. This is called plaque rupture. This exposes the softer inner core of the plaque to blood. This can trigger the clotting mechanism in the blood to form a blood clot. Therefore, a build-up of atheroma is the root problem that leads to most cases of ACS. (The diagram below shows four patches of atheroma as an example. However, atheroma may develop in any section of the coronary arteries.)

Upper large vein into right atrium
Aorta
The heart looking from the front
Blood flow through artery is blocked by blood clot
Right coronary artery
Pulmonary arteries (take blood to lungs)
Left coronary artery
Blood clot
Circumflex branch of left coronary artery
Atheroma
Artery wall
Lower large vein into right atrium
Patches of atheroma on lining of artery
Section of a coronary artery
MYOCARDIAL INFARCTION
Damaged area of heart muscle 'downstream' from blocked artery

What are the symptoms of a heart attack?

The most common symptom is severe chest pain, which often feels like a heavy pressure feeling on your chest. The pain may also travel up into your jaw and down your left arm or down both arms. You may also sweat, feel sick and feel faint. You may also feel short of breath. The pain may be similar to angina, but it is usually more severe and lasts longer. (Angina usually goes off after a few minutes. Heart attack pain usually lasts more than 15 minutes - sometimes several hours.) However, some people have only a mild discomfort in their chest. The pain can sometimes feel like indigestion or heartburn. Occasionally, a heart attack happens without causing any pain. This is usually diagnosed when you have a heart tracing (electrocardiograph, or ECG) at a later stage. Some people collapse and die suddenly, if they have a large portion of heart muscle damaged. This is not very common.

Figure 2:Home Page



Enter your Name : Reshma

Just enter what you want to say to the doctor..

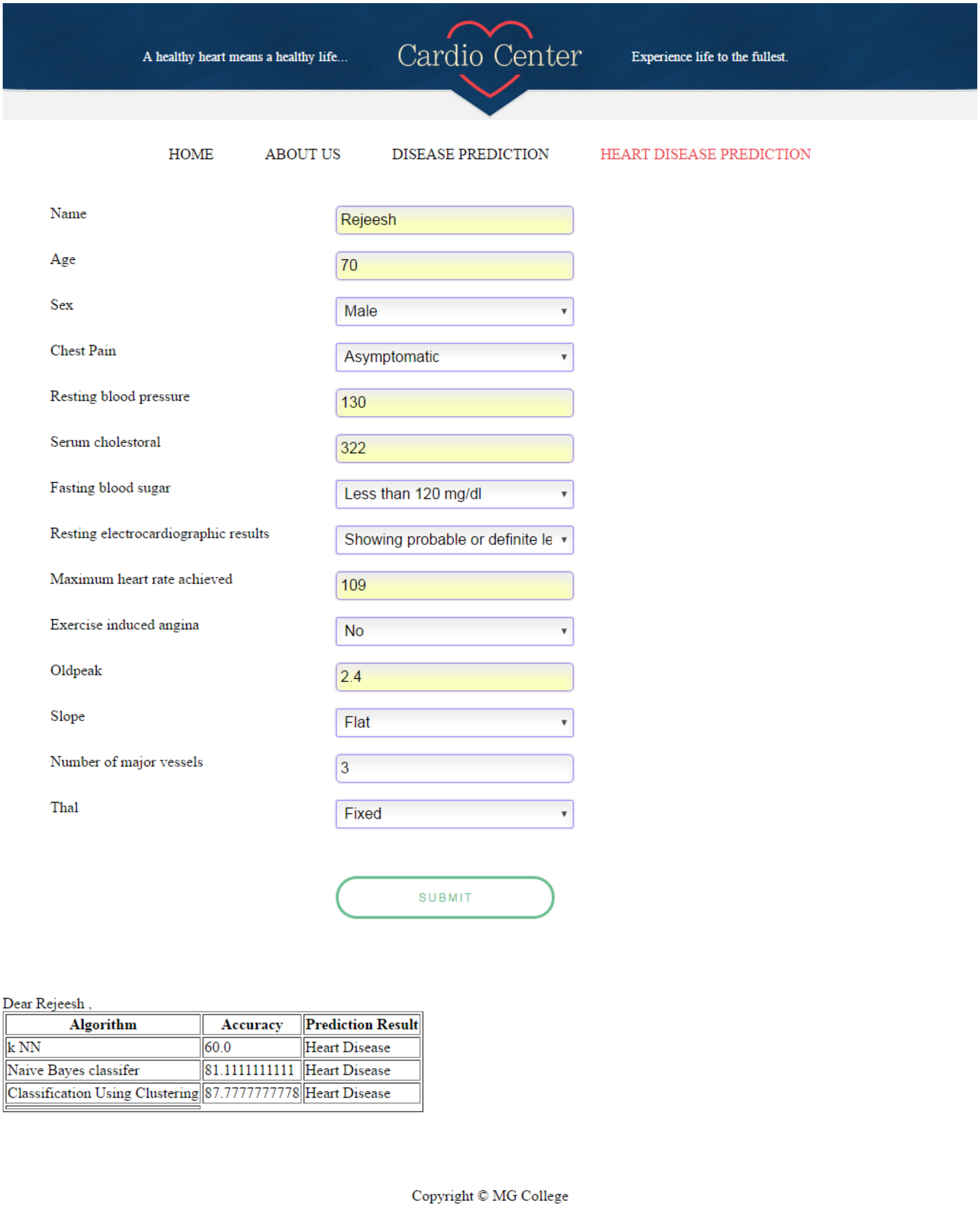
Good morning doctor.

Sir I have been suffering from fever since yesterday. And also feeling headache and shivering.

I'm unable to sleep at nights.

Dear Reshma ,
It seems you are having flue

Figure 3:Disease Prediction Page



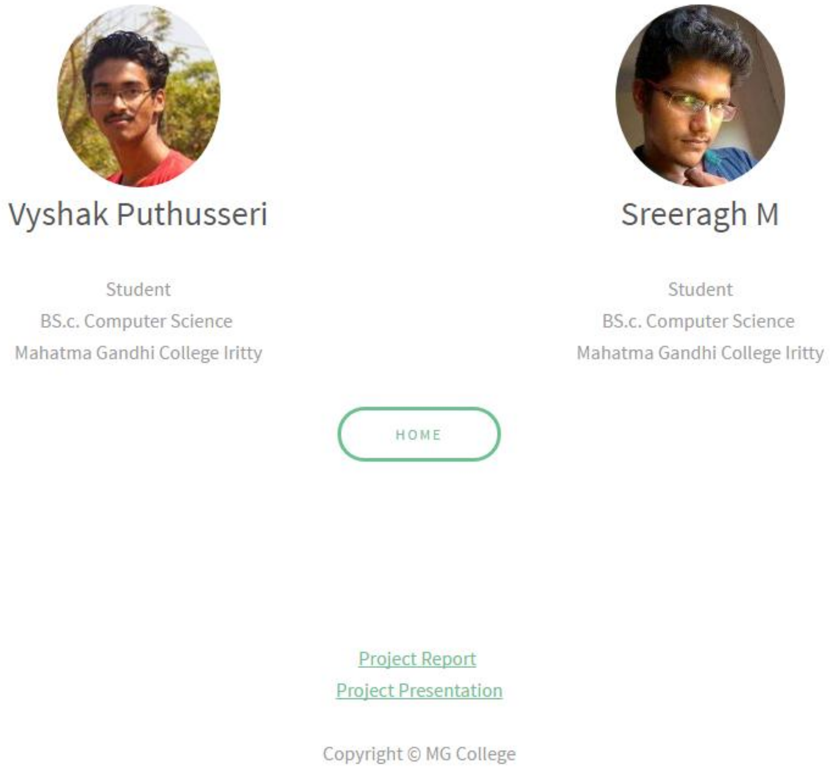
Name	Rejeesh
Age	70
Sex	Male
Chest Pain	Asymptomatic
Resting blood pressure	130
Serum cholestoral	322
Fasting blood sugar	Less than 120 mg/dl
Resting electrocardiographic results	Showing probable or definite le
Maximum heart rate achieved	109
Exercise induced angina	No
Oldpeak	2.4
Slope	Flat
Number of major vessels	3
Thal	Fixed

Dear Rejeesh ,

Algorithm	Accuracy	Prediction Result
k NN	60.0	Heart Disease
Naive Bayes classifer	81.1111111111	Heart Disease
Classification Using Clustering	87.7777777778	Heart Disease

Copyright © MG College

Figure 4:Heart Disease Prediction Page



Vyshak Puthusseri

Student
BS.c. Computer Science
Mahatma Gandhi College Iritty

Sreeragh M

Student
BS.c. Computer Science
Mahatma Gandhi College Iritty

Copyright © MG College

Figure 3:About Us Page