Heart Disease Prediction Using K-Means Clustering And Naive Bayes Algorithm

A PROJECT REPORT

Submitted in fulfilment of the award of Degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING

Submitted By

K.VINEETH VARMA	**15951A05P0**
DAMA VISHAL	**15951A05P2**
P VINEETH	**15951A05N7**
SURYA VAMSHI	**15951A05K8**

Under the Supervision of
Dr. Y MOHANA ROOPA
Professor, CSE

COMPUTER SCIENCE AND ENGINEERING
INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal – 500043, Hyderabad, Telangana

2018-2019

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal – 500043, Hyderabad, Telangana

COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that work embodied in this dissertation entitled **"HEART DISEASE PREDICTION USING K MEANS CLUSTERING AND NAIVE BAYES ALGORITHM"** being submitted by

K VineethVarma	**15951A05P0**
Dama Vishal	**15951A05P2**
P Vineeth	**15951A05N7**
Surya vamshi	**15951A05K8**

for partial fulfilment of the requirement for the award of **'Bachelor of Technology'** in **'Computer Science and Engineering'** discipline to Institute of Aeronautical Engineering, Dundigal, Hyderabad during the academic year 2018-2019 is a record of bonafide piece of work, undertaken by him/her the supervision of the undersigned.

Approved and Supervised by

Dr. Y MOHANA ROOPA
Professor, CSE

Forwarded by

Dr. L.V. Narasimha Prasad
Principal

Dr. M Madhu Bala
Professor and Head

EXTERNAL EXAMINER

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal – 500043, Hyderabad, Telangana

COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We **'K Vineethvarma, Dama Vishal, P Vineeth, Surya vamshi'**, are students of **'Bachelor of technology** in **Computer Science and Engineering'**, **session: 2018-2019**, Institute of Aeronautical Engineering, Dundigal, Hyderabad, hereby declare that the work presented in this Project Work entitled **'HEART DISEASE PREDICTION USING K MEANS CLUSTERING AND NAIVE BAYES ALGORITHM'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering ethics. It contains no material previously published or written by another person nor material which had been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement had been made in the text.

K.VINEETH VARMA	**15951A05P0**
DAMA VISHAL	**15951A05P2**
P VINEETH	**15951A05N7**
SURYA VAMSHI	**15951A05K8**

Date:

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whole constant guidance and encouragement crown all the efforts with success. We thank our college management and respected **Sri M. Rajashekar Reddy**, Chairman, IARE, Dundigal for providing us the necessary infrastructure to carry out the project work.

We express our sincere thanks to our beloved Principal, **Dr.L.V. Narasimha Prasad**, who has been a great source of information for our work and, **Dr. M Madhu Bala**, Professor and Head, Department of CSE for extending her support to carry on this project work.

We are especially thankful to our internal supervisor **Dr.Y Mohana Roopa**, Professor, Department of CSE, for his internal support and professionalism who helped us in shaping the project into successful one.

We take this opportunity to express our thanks to one and all who directly or indirectly helped us in bringing this effort to present form.

K VineethVarma	**15951A05P0**
Dama Vishal	**15951A05P2**
P Vineeth	**15951A05N7**
Surya vamshi	**15951A05K8**

ABSTRACT

Nowadays people work on computers for hours and hours they don't have time to take care of themselves. Due to hectic schedules and consumption of junk food it affects the health of people and mainly heart. So we are implementing an heart disease prediction system using data mining technique Naïve Bayes and k-means clustering algorithm. It is the combination of both the algorithms. It helps in predicting the heart disease using various attributes and it predicts the output as in the prediction form. For grouping of various attributes it uses k-means algorithm and for predicting it uses naïve bayes algorithm.

TABLE OF CONTENTS

Chapter			Particulars	page no.
1			Introduction	1
	1.1		Introduction	1
	1.2		Existing System	2
		1.2.1	Functionality	2
		1.2.2	Drawbacks	2
	1.3		Proposed System	2
		1.3.1	Functionality	2
		1.3.2	K-Means Algorithm	2
		1.3.3	Naive Bayes Algorithm	3
		1.3.4	Advantages	4
2			Problem Statement and Requirement Specifications	5
	2.1		Problem Statement	5
	2.2		Requirement Specifications	5
		2.2.1	Functional Requirements	5
		2.2.2	Non-Functional Requirements	5
	2.3		Software Requirements	7
	2.4		Hardware Requirements	7
3			Methodology	8
	3.1		Block Diagram	8
	3.2		UML Diagrams	8
		3.2.1	Class Diagram	8
		3.2.2	Use Case Diagram	9
		3.2.3	Sequence Diagram	9
		3.2.4	Collaboration Diagram	10
		3.2.5	Deployment Diagram	10
	3.3		Modules Description	11
4			Implementation	19

	4.1		Sample Code	19
5			Results	30
	5.1		5.1 Outputs	30
		5.1.1	Home Page	30
		5.1.2	Admin Page	30
		5.1.3	Data Set	31
		5.1.4	Uploading Data Set	31
		5.1.5	K Means Algorithm	32
		5.1.6	Naive Bayes Algorithm	33
		5.1.7	Frequency Tables	34
		5.1.8	User Page	38
		5.1.9	User Details	39
		5.2.0	Final Result	39
6			Conclusion	41
7			References	42

LIST OF FIGURES

Figure no.	Name	Page no.
1.1	Flow Chart for K-Means Algorithm	3
1.2	Naive Bayes Algorithm	4
3.1	Block Diagram	8
3.2	Class Diagram	8
3.3	Use Case Diagram	9
3.4	Sequence Diagram	9
3.5	Collaboration Diagram	10
3.6	Deployment Diagram	10
3.7	JDBC Architecture	15
5.1	Home Page	30
5.2	Admin Page	30
5.3	Data Set	31
5.4	Uploading Data Set	31
5.5	Data Set Records	32
5.6	K Means Algorithm	32
5.7	Formation of Clusters	33
5.8	Naive Bayes Algorithm	33
5.9	Frequency Table For Age Data	34
5.10	Frequency Table For Blood Pressure Data	34
5.11	Frequency Table For calculation	35
5.12	Frequency Table For Chest Pain Data	35
5.13	Frequency Table For Cholesterol Data	36
5.14	Frequency Table For Gender Data	36
5.15	Frequency Table For Heart Rate Data	37
5.16	Frequency Table For Smoker Data	37
5.17	Prediction Page	38
5.18	User Page	38
5.19	User Details	39
5.20	Final Result	39
5.21	Frequency Table For Whole Data	40

CHAPTER-1
INTRODUCTION

1.1 INTRODUCTION

The practice of examining large pre existing data bases in order to generate new information. converts raw data into useful information. It analyzes the data for relationships that are previous been discovered. The steps of data mining are: data cleaning, data integration, data selection, transformation, data mining, pattern evaluation and knowledge representation. Medical data is a domain of lot of imprecision and uncertainity. The clinical decisions are usually based on doctors intuition. Therefore this may lead to disastrous sequences. Due to this there are many e in the clinical decisions and it results in excessive medical costs. It converts the data objects in streams of bytes and stores it into data base.

Most hospitals today use decision-support systems, but to get the results of the disease are largely limited i.e., Determine based on gender, marital status and who have been treated for heart failure.

Solutions are always made in a hospital based on intuition and experience of doctors, and not on the rich knowledge data that are hidden in the database. This process leads to undesirable biases, errors and unnecessary health care costs, which affects the quality of services provided to patients. Machine learning can be used to determine the automatic conclusion of diagnostic rules from the past descriptions, successfully treat a patient, as well as experts and specialists will help make the diagnostic process more objective and more reliable.

Intelligent decision support systems are defined as interactive computer systems to help make decisions in the use of data sets and models to find problems, solve problems and make decisions. The proposed system uses the analysis to integrate and make the right decision at the clinic with a computer system. This patient record can reduce the number of patients to improve the safety of medical decisions errors, reduces unwanted changes in practice and improve patient outcomes. This proposal is promising, as modelling and analysis instruments, such as data mining, have the ability to generate knowledge-rich environment that can help to significantly improve the quality of clinical decisions.

1.2 EXISTING SYSTEM

1.2.1 FUNCTIONALITIES

- Most hospitals today use decision-support systems, but to get the results of the disease are largely limited.
- In the existing model naive bayes algorithm is implemented on the data set as a result large number of outputs are generated which increases complexity.
- Even in some cases, the testing process is manually done.

1.2.2 DRAWBACKS

- It is a time taking process.
- It requires a lot of effort.
- The accuracy of the prediction is decreased.

.

1.3 PROPOSED SYSTEM

1.3.1 FUNCTIONALITIES

- In the proposed system Naïve Bayes algorithm and K-means algorithm are implemented on the dataset.
- Initially the dataset is divided into some number of clusters using k means algorithm.
- To the clusters formed naive bayes algorithm is implemented and frequency tables are generated.
- Intelligent decision support systems can be used to help make decisions and to solve problems.

1.3.2 K-MEANS ALGORITHM

K-means Clustering is often referred to as unsupervised learning. Because there is no need for a marked data, learning algorithms without a teacher are suitable for many applications where the labelled data are difficult to obtain. Uncontrolled tasks such as clustering, as often used, which would investigate and characterize the data set before

starting a controlled learning objective. Since clustering is performed without using a class label, some idea of the similarity must be determined on the basis of object

attributes. Description of similarities and method in which dots are grouped differ based clustering algorithm is used.

K-means algorithm is a simple iterative clustering algorithm, which divides a particular set of data on the number of clusters, k specified by the user. The algorithm is simple to implement and run relatively fast, easily adaptable and common in practice. It is historically one of the most important data mining algorithms.

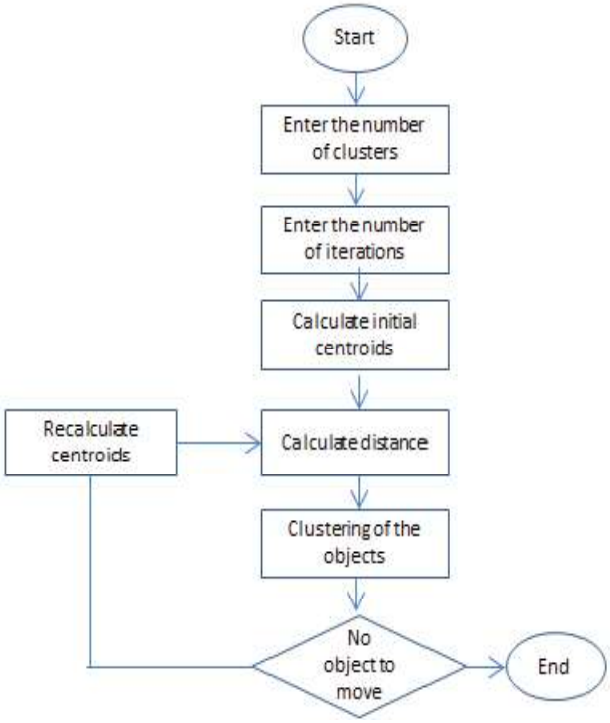

Fig 1.1: flow chart for k-means algorithm

1.3.3 NAÏVE BAYES ALGORITHM

Bayesian classification presents supervised learning method, and the method of statistical classification. It assumes a basic model of probability, and it allows us to capture some uncertainty about the model by determining the probability of the outcome source. It can solve problems of diagnosis and prognostic. Bayesian approach to classification is based on the theorem, which states that if the density distribution of

each of the classes is known, the desired algorithm can be written in explicit analytic form. Furthermore, this algorithm is optimal, that is, it has the minimum error

probability. It calculates the precise probability of the hypothesis, and it is resistant to the noise at the input data.

Naïve Bayesian method is particularly relevant for the problems of high dimensionality of the input space, in case of problems with a large number of input variables. Despite its simplicity, Naive Bayes often superior to other more sophisticated classification methods and this model takes on different characteristics of the patients with different diseases, define and determines the probability of each input source for a predictable state. Bayes' Rules one of the main and basic algorithms of machine learning and data mining methods. The algorithm is used to create models with predictive capabilities. It provides all possible new ways of learning and understanding the data.

Naive Bayes is mainly preferred for the following purposes:

1) When working with very large datasets.
2) When the attributes are independent of each other.

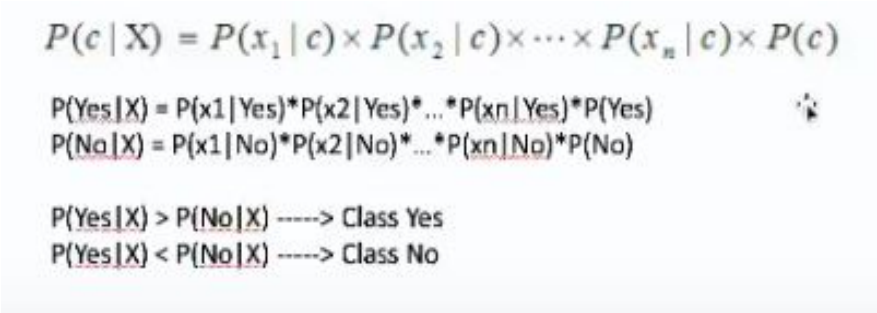

Fig 1.2: Naive bayes algorithm

1.3.4 ADVANTAGES

- The time taken for this process is less when compared to existing system.
- It is easy and fast to predict the result by using data set.

CHAPTER-2

PROBLEM STATEMENT AND REQUIREMENT SPECIFICATIONS

2.1 PROBLEM STATEMENT

Due to the presence of large number of numerical attributes, when the naive bayes algorithm is applied various complex outputs are generated as a result the accuracy in prediction is decreased.

2.2 REQUIREMENT SPECIFICATIONS

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports.

2.2.1 FUNCTIONAL REQUIREMENTS

ADMIN

- Admin can login and upload the datasets.
- These datasets are then clustered and formed into frequency tables based on the datasets.

USER

- User can login and fill the appropriate details in the given text field.
- After the process the result is displayed.

LOGOUT

- After performing all the functionalities then the user and admin needs to logout.

2.2.2 NON-FUNCTIONAL REQUIREMENTS:

- **FLEXIBILITY AND SCALABILITY**

Oracle itself has given a set of applications with JDK but the whole developer community can develop their own applications and they have access to same resources

and public API which are accessible to core applications.

- **ROBUST**

The application is fault tolerant with respect to illegal user/receiver inputs. Error checking has been built in the system to prevent system failure.

- **FRAGMENTATION**

Java gave the same environment which is open; the entire API's which is open to all the devices which reduces fragmentation. If you develop an java application, it will run on all the devices.

- **OPEN SOURCE**

Java open source is free and easy to download. Java is a platform in depended based programming language and The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine.

- **SCALABILITY**

The system can be extended to integrate the modifications done in the present application to improve the quality of the product. This is meant for the future works that is to be done on the application.

- **RELIABILITY**

Since the application is being developed through java the most famous, efficient and reliable language, so it is reliable in every aspect until and unless there is an error in the programming side. Thus, the application can be a compatible and reliable one.

- **PORTABILITY**

This System must be intuitive enough such that user with average background in using mobile phones can quickly experiment with the system and learn how to use the project. The system has user friendly interface.

2.3 SOFTWARE REQUIREMENTS

- Operating System : Windows family
- Technology : Java (1.7/1.8)
- Web Technologies : Html, CSS, JSP
- Web Server : Tomcat 7/8
- Database : My SQL5

2.4 HARDWARE REQUIREMENTS:

- Processer : Any Update Processer
- Ram : Min 1 GB
- Hard Disk : Min 100 GB

3. METHODOLOGY

3.1 BLOCK DIAGRAM

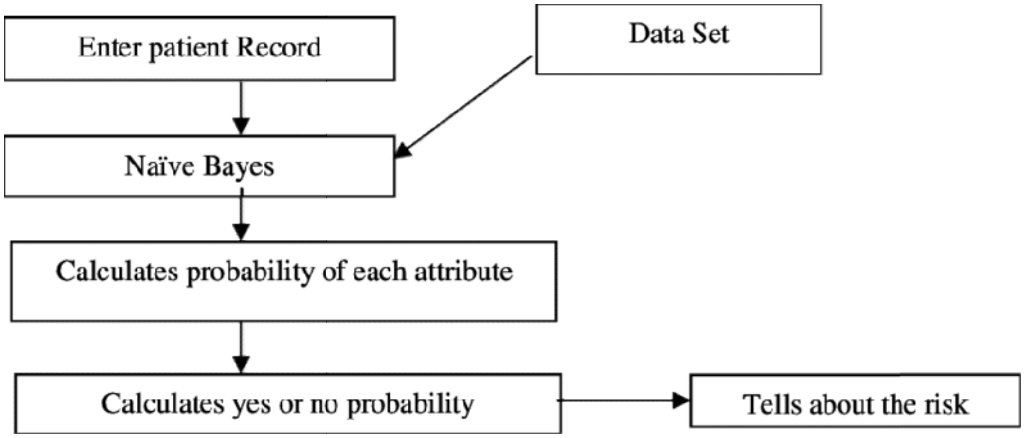

Fig 3.1: Block Diagram

3.2 UML DIAGRAMS

3.2.1 CLASS DIAGRAM

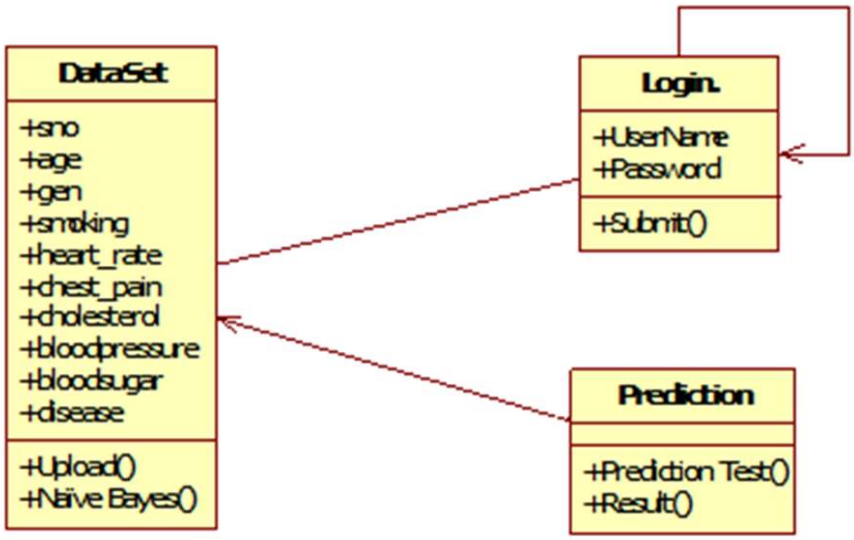

Fig 3.2: Class Diagram

3.2.2 USE CASE DIAGRAM

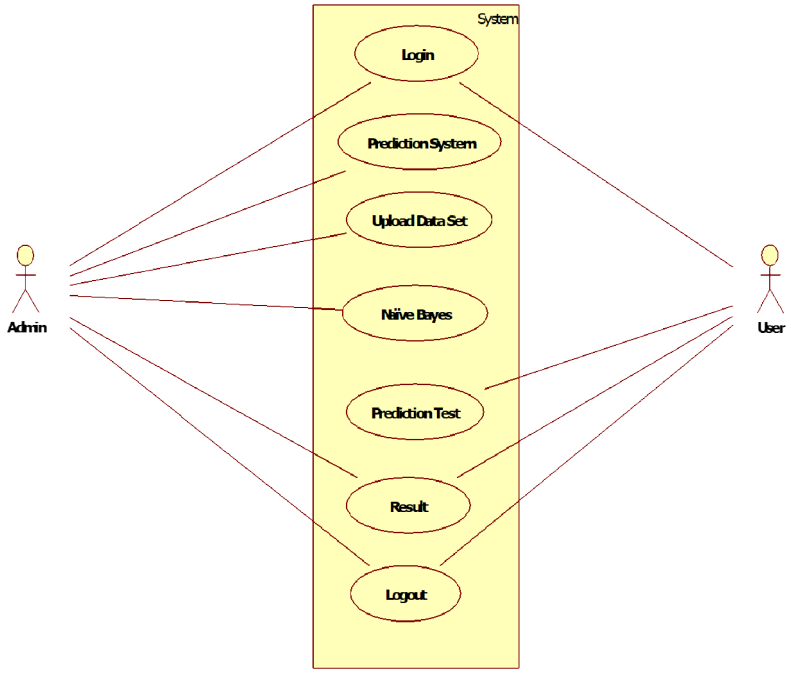

Fig 3.3: Use Case Diagram

3.2.3 SEQUENCE DIAGRAM

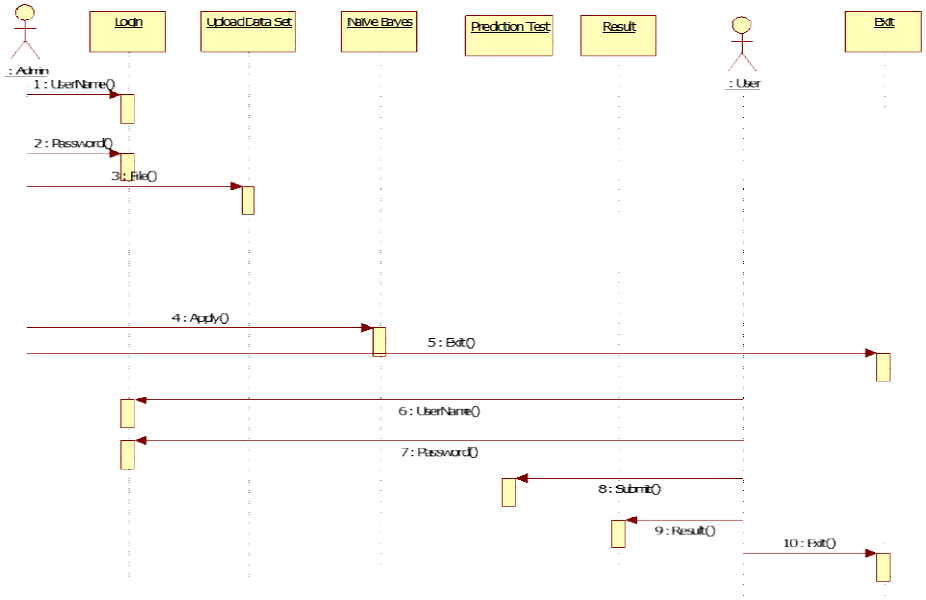

Fig 3.4: Sequence Diagram

3.2.4 COLLABORATION DIAGRAM

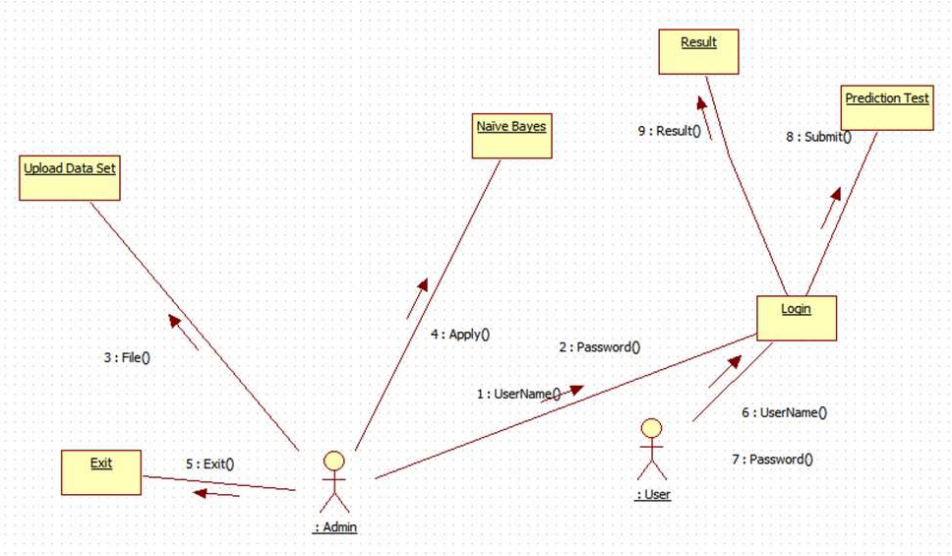

Fig 3.5: Collaboration Diagram

3.2.5 DEPLOYMENT DIAGRAM

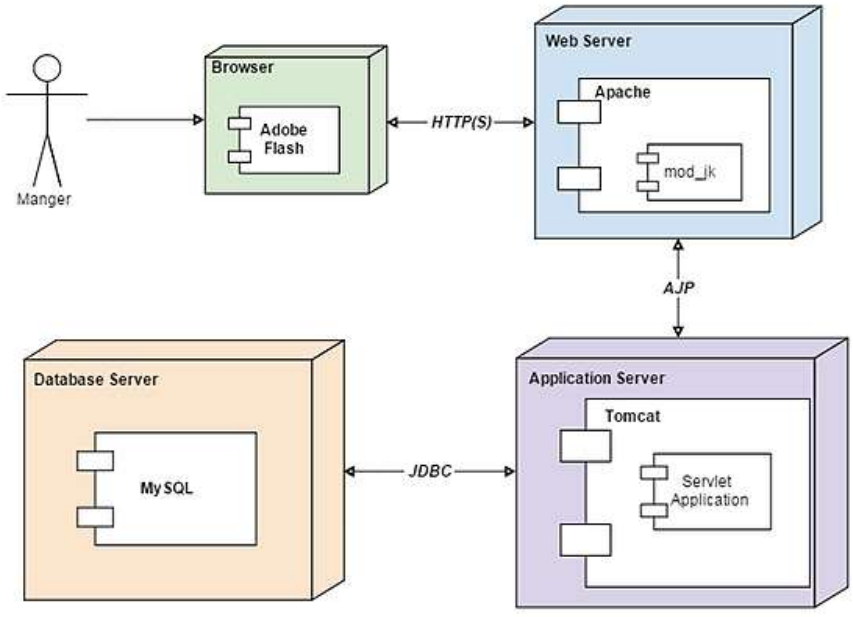

Fig 3.6: Deployment Diagram

3.3MODULES DESCRIPTION

LOGIN

- The login interface allows user and admin to sign in.

There are two user types:

- Administrator.
- User (Patient).

ADMINISTRATOR MODULE

- Admin can login and upload the datasets.
- These datasets are then clustered and formed into frequency tables based on the datasets**.**

USER MODULE

- User can login and fill the appropriate details in the given text field.
- After the process the result is displayed.

JDK

The Java Development Kit (JDK) is provided by Sun Microsystems as a basic development environment for Java. The JDK provides similar facilities to the cc compiler for C programs, plus a JVM simulator and some additional facilities such as a debugger. To use the JDK, programs are constructed as ascii text files (by using an editor, for example). The program files are compiled, which translates the Java code to JVM byte code in .class files.

Each public class must be in a file having the class name (case sensitive on Unix) followed by a .java suffix. There may be any number of classes defined in a .java file, but the compiler produces a separate .class file for each class. A file is compiled with the **javac** command, which is similar to the cc (or gcc) command. A class is executed (or more precisely, the method **main** in a class is executed) by the command **java** with the class name (not the .class file) as the parameter. Thus, for example, to compile the program in file Hi.java, we would use the command

Javac Hi.java

and then to execute the program we would use the command

java Hi

Both compile-time and execution-time (exceptions) error messages include the file name and line where the error occurred. No .class file is produced if there is a compile-time error.

JAVA PLATFORM

The programmer writes Java source code in a text editor which supports plain text. Normally the programmer uses an Integrated Development Environment(IDE) for programming. An IDE supports the programmer in the task of writing code, e.g. it provides auto-formatting of the source code, highlighting of the important keywords, etc.

At some point the programmer (or the IDE) calls the Java compiler (javac). The Java compiler creates the byte code instructions. These instructions are stored in .class files and can be executed by the Java Virtual Machine.

J2SE

J2SE is a collection of Java Programming Language API (Application programming interface) that is very useful to many Java platform programs. It is derived from one of the most dynamic programming languages known as "JAVA" and one of its three basic editions of Java known as Java standard edition being used for writing Applets and other web-based applications.

J2SE platform has been developed under the Java umbrella and primarily used for writing applets and other Java-based applications. It is mostly used for individual computers. Applet is a type of fast-working subroutine of Java that is platform-independent but work within other frameworks. It is a mini application that performs a variety of functions, large and small, ordinary and dynamic, within the framework of larger applications.

J2SE provide the facility to users to see Flash movies or hear audio files by clicking on a Web page link. As the user clicks, page goes into the browser environment and begins the process of launching application-within-an-application to play the requested video or sound application. So many online games are being developed on J2SE. JavaBeans can also be developed by using J2SE.

J2EE

Java***2*** Platform ***E***nterprise ***E***dition. J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multitier, Web-based applications.

KEY FEATURES AND SERVICES OF J2EE:

- At the client tier, J2EE supports pure HTML, as well as Java applets or applications. It relies on Java Server Pages and servlet code to create HTML or other formatted data for the client.
- Enterprise JavaBeans (EJBs) provide another layer where the platform's logic is stored. An EJB server provides functions such as threading, concurrency, security and memory management. These services are transparent to the author.
- Java Database Connectivity (JDBC), which is the Java equivalent to ODBC, is the standard interface for Java databases.
- The Java servlet API enhances consistency for developers without requiring a graphical user interface.

JAVA DATA BASE CONNECTION

JDBC stands for Java Database Connectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks commonly associated with database usage:

- Making a connection to a database.
- Creating SQL or MYSQL statements.
- Executing that SQL or MYSQL queries in the database.
- Viewing & Modifying the resulting records.

Fundamentally, JDBC is a specification that provides a complete set of interfaces that allows for portable access to an underlying database. Java can be used to write different types of executables, such as:

- Java Applications.
- Java Applets.
- Java Servlets.
- Java ServerPages (JSPs).
- Enterprise JavaBeans (EJBs).

All of these different executables are able to use a JDBC driver to access a database and take advantage of the stored data.

JDBC provides the same capabilities as ODBC, allowing Java programs to contain database-independent code.

JDBC ARCHITECTURE:

The Java Database Connectivity (JDBC) API provides universal data access from the Java programming language. Using the JDBC API, you can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built.The Driver Manager class acts as an interface between user and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. The Driver Manager class maintains a list ofDriver classes that have registered themselves by calling the method DriverManager.registerDriver().

The JDBC API supports both two-tier and three-tier processing models for database access but in general JDBC Architecture consists of two layers:

- **JDBC API:** This provides the application-to-JDBC Manager connection.
- **JDBC Driver API:** This supports the JDBC Manager-to-Driver Connection.

The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases. The JDBC driver manager ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

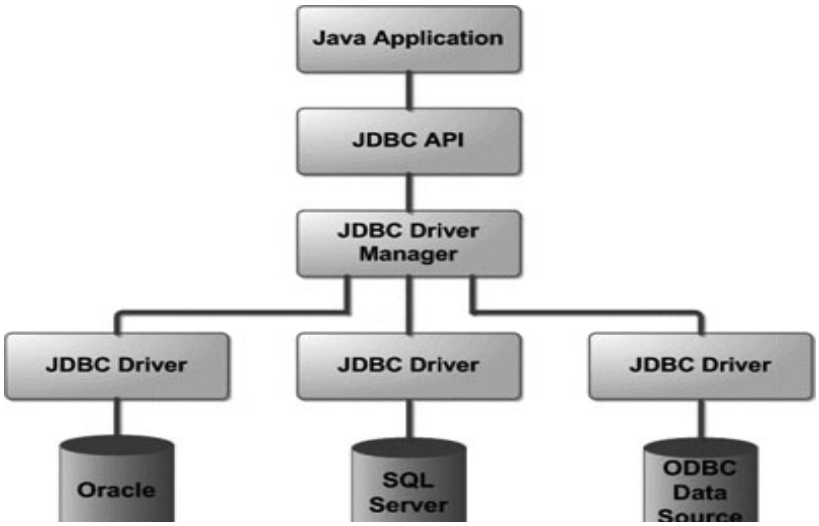

Fig 3.7: JDBC Architecture

The above is the architectural diagram, which shows the location of the driver manager with respect to the JDBC drivers and the Java application.

TOMCAT 7.0 WEB SERVERS

Apache Tomcat is a web container developed at the Apache Software Foundation (ASF). Tomcat implements the servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own HTTP server internally.

ENVIRONMENT

Tomcat is a web server that supports servlets and JSPs. The accompanying Tomcat Jasper compiler compiles JSPs into servlets. The Tomcat servlet engine is often used in combination with an Apache HTTP Server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Tomcat is cross-platform, running on any operating system that has a Java Runtime Environment. The following properties can be specified, either as system properties, or by using a deployer. Properties file located in the root folder of the deployer package:

- Build: The build folder used will be, by default, ${build}/webapp${path}. After the end of the execution of the compile target, the web application WAR will be located at ${build}/webapp${path}.war.
- Webapp: Folder containing the expanded web application which will be compiled and validated. By default, the folder is myapp.
- Path: Deployed context path of the web application, by default /myapp.
- URL: Absolute URL to the manager web application of a running Tomcat server, which will be used to deploy and undeploy the web application. By default, the deployer will attempt to access a Tomcat instance running on local host, at http://localhost:8080/manager.
- Username: Username to be used to connect to the Tomcat manager.

- Password: Password to be used to connect to the Tomcat manager

A web application which is programmatically developed by the developer is stored in the web apps folder and WEB-INF folder also saved in that location which consists of folder named classes which supports to run application automatically.

The deplorer package includes a ready to use Ant script, with the following targets:

- compile (default): Compile and validate the web application. This can be used standalone and does not need a running Tomcat server. The compiled application will only run on the associated Tomcat 5.0.x server release and is not guaranteed to work on another Tomcat release, as the code generated by Jasper depends on its runtime component. It should also be noted that this target will also compile automatically any Java source file located in the /WEB-INF/classes folder of the web application.

- deploy: Deploy a web application (compiled or not) to a Tomcat server
- undeploy: Undeploy a web application
- Start: Start web application
- reload: Reload web application
- Stop: Stop web application

MYSQL

The MYSQL NAÏVE software delivers a very fast, multi-threaded, multi-user, and robustSQL(Structured Query Language) database server. MYSQL Server is intended for mission critical, heavy-load production systems as well as for embedding into mass-deployed software. MYSQL is a trademark of MYSQL AB. The MYSQL software has Dual Licensing, which means you can use the

MYSQL software free of charge under the GNU General Public License (http://www.gnu.org/licenses/). We can also purchase commercial MYSQL licenses from MYSQL AB if you do not wish to be bound by the terms of the GPL. The MYSQL web site (http://www.mysql.com/) provides the latest information about the MYSQL software.

WHAT IS MYSQL?

MYSQL, the most popular Open Source SQL database, is developed and provided by

MYSQL AB. MYSQL AB is a commercial company that builds its business providing services around the MYSQL database. The MYSQL web site (http://www.mysql.com/) provides the latest information about MYSQL software and MYSQL AB.

MYSQL IS A RELATIONAL DATABASE MANAGEMENT SYSTEM

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The tables are linked by defined relations making it possible to combine data from several tables on request. The SQL part of "MYSQL" stands for "Structured Query Language" the most common standardized language used to access databases. The MYSQL Database Server is very fast, reliable, and easy to use. If that is what you are looking for, you should give it a try. MYSQL Server also has a practical set of features developed in close cooperation with our users. You can find a performance comparison of MYSQL Server to some other database managers on our benchmark page.

TECHNICAL FEATURE OF MYSQL

For advanced technical information, The MYSQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different backend, several different client programs and libraries, administrative tools, and a wide range of programming interfaces (APIs).

JAVA SERVER PAGES

Jsp technology enables you to mix regular static html with dynamically generated content from servlets. Separating the static html from the dynamic content provides a number of benefits over servlets alone.Jsp is easy to learn and allows developers to quickly produce wed sites and application in an open and standard way. Jsp is based on java, an object-oriented language. Jsp offers a robust platform for web development.

Main reasons to Jsp:

- Multi-platform
- Component reuse by using java beans and Ejb
- Advantages if java

We can take one Jsp file and move it to another platform, web server or Jsp servlet engine.

JSP Architecture:

Jsp are built on top of sun's servlet technology. Jsp is essentially an html page with special jsp tags embedded. These jsp tags can contain java code. The jsp file extension is '. jsp' rather than 'htm' or '.html'. The jsp engine parses the jsp and creates a java servlet source file. It then compiles the source file into a class file; this is done the first time and this why the jsp is probably slower the first time it is accessed. Any time after this, the special compiled servlet is executed and is therefore returns faster.

CHAPTER-4

IMPLEMENTATION

4.1SAMPLE CODE

ADMIN CODE

```
<%@ include file="header.jsp"%>
<font size="+1">
<h1><font align="center" size="" color="#ff9900">Admin Page</h1></font><br>
<form method="post" action="alogin.jsp">
<table align=left width="50%"><tr><td>
<tr><td><div class="input-group input-group-lg">
<span class="input-group-addon" id="sizing-addon1">ID</span>
<input type="text" name ="uid" required class="form-control"
placeholder="Username" aria-describedby="sizing-addon1"></div>

<tr><td><div class="input-group input-group-lg">
<span class="input-group-addon" id="sizing-addon1">***</span>
<input type="text" name ="pwd" requiredclass="form-control"
placeholder="Password" aria-describedby="sizing-addon1"></div>

<tr><td>       <span class="input-group-btn">
<button class="btnbtn-default" type="submit">Login</button>
                </span>
<td></tr>
</table>
<table align=right ><tr><td><td>
```

```
<%
String m=request.getParameter("id");
if (m! =null &&m.equalsIgnoreCase("fail"))          {
out.println("<font  color='red'><blink><h3>   Login Fail
!! </blink></font></h3>");                                  }
%>


</table>
</form>
<br><br><br><br>
<%@ include file="footer.jsp"%>
```

NAÏVE BAYES CODE

```
%@ include file=" aheader.jsp" %>
<%@ page import=" java.sql. *" import=" java.util.*" import=" databaseconnection. *"
import=" CT.*" %>
<%
int y=0, n=0; double tot=0;
%>
<font size="" color=" #cc0033">
<h2><font size="" color=" #00ff33">Forming into Clustures</font></h2>
<table cellspacing="10" align=" center">
<tr><td><b>Sno<td><b>Age<td><b>Gender<td><b>Smoker<td><b>HeartRate<td><b>Ch
estPain<td><b>Cholesterol<td><b>BloodPressure<td><b>BloodSugar<td><b>Heart Attack
<%
try{
        Connection con2 = databasecon.getconnection();
        Statement st5 = con2.createStatement();
        Statement st2 = con2.createStatement();
        st2.executeUpdate("delete from prediction");
```

```
st2.executeUpdate("delete from calc");
String sss1 = "select * from setdata";

ResultSet rs1=st5.executeQuery(sss1);
While(rs1.next())
{
%><tr><td><H4><%=rs1.getString(1)%><td><H4><%=Cluster.age(rs1.getInt(2))%>
<td><H4><%=rs1.getString(3)%><td><H4><%=rs1.getString(4)%><td><H4><%=Cluster.h
eartrate(rs1.getInt(5))%><td><H4><%=Cluster.chestpain(rs1.getInt(6))%><td><H4><%=Cl
uster.cholesterol(rs1.getInt(7))%><td><H4><%=Cluster.bloodpressure(rs1.getInt(8))%><td>
<H4><%=Cluster.bloodsugar(rs1.getInt(9))%><td><H4><%=rs1.getString(10)%>
<%
st2.executeUpdate("insertintoprediction values ('" +rs1.getString(1) +"','" +Cluster. Age
(rs1.getInt(2)) +"', '" +rs1.getString(3) +"', '" +rs1.getString(4) +"', '" +
Cluster.Heartrate(rs1.getInt(5)) +"', '" +Cluster.chestpain(rs1.getInt(6)) +"',
'" +Cluster.cholesterol(rs1.getInt(7)) +"', '" +Cluster.bloodpressure(rs1.getInt(8)) +"',
'" +Cluster.bloodsugar(rs1.getInt(9)) +"','" +rs1.getString(10) +"')");
}
rs1=st5.executeQuery("SELECT disease_ct, COUNT (*) FROM prediction WHERE
disease_ct='Y' GROUP BY disease_ct");
if(rs1.next())
{
y=rs1.getInt(2);
}
rs1=st5.executeQuery("SELECT disease_ct, COUNT (*) FROM prediction WHERE
disease_ct='N' GROUP BY disease_ct");
if(rs1.next())
{
n=rs1.getInt(2);
}
tot=y+n;
st5.executeUpdate("insert into calcvalues ('" +y+"'," +y/tot+",'" +n+"'," +n/tot+","
+tot+")");
```

```
}


catch (Exception e1)
{
out.println(e1);
}
%>
</table>
<form method=" post" action=" naïve2.jsp">
        <input type=" submit" value=" Next">
</form>
<%@ include file=" footer.jsp" %>
---------------------------------------------------------------------
<%@ include file="aheader.jsp"%>
<%@ page import="java.sql. *" import="java.util.*" import="databaseconnection. *"
import="CT.*" %>
<%
int y=0, n=0; double tot=0;
%>
<font size="" color="#cc0033">
<%
try {
        Connection con2 = databasecon.getconnection();
        Statement st5 = con2.createStatement();
        Statement st2 = con2.createStatement();
        String sss1 = "SELECT age_ct, COUNT(disease_ct) FROM `prediction` WHERE
        disease_ct='Y' GROUP BY age_ct ";
        ResultSet rs1=st5.executeQuery(sss1);
        while (rs1.next())
        {
```

```
                st2.executeUpdate("update age_tab set y_ct="+rs1.getInt(2)+" where
age_ct="'+rs1.getString(1)+"' ");
        }


sss1 = "SELECT age_ct, COUNT(disease_ct) FROM `prediction` WHEREdisease_ct='N'
GROUP BY age_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update age_tab set n_ct="+rs1.getInt(2) +" where age_ct=
'"+rs1.getString(1) +"' ");
}
//==================

sss1 = "SELECT gen_ct, COUNT(disease_ct) FROM `prediction` WHEREdisease_ct='Y'
GROUP BY gen_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update gen_tab set y_ct="+rs1.getInt(2) +" where
gen_ct="'+rs1.getString(1) +"' ");
}

sss1 = "SELECT gen_ct, COUNT(disease_ct) FROM `prediction` WHEREdisease_ct='N'
GROUP BY gen_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update gen_tab set n_ct="+rs1.getInt(2) +" where
gen_ct="'+rs1.getString(1)+"' ");
}
```

```
//=========================



sss1 = "SELECT smoking_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='Y' GROUP BY smoking_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update smoker_tab set y_ct="+rs1.getInt(2) +" where
smoker_ct="'+rs1.getString(1) +"' ");
}

sss1 = "SELECT smoking_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='N' GROUP BY smoking_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update smoker_tab set n_ct="+rs1.getInt(2) +" where
smoker_ct="'+rs1.getString(1)+"' ");
}
//=========================

//=========================

sss1 = "SELECT heart_rate_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='Y' GROUP BY heart_rate_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update heartrate_tab set y_ct="+rs1.getInt(2) +" where
```

```
heartrate_ct='"+rs1.getString(1) +"' ");

}




sss1 = "SELECT heart_rate_ct, COUNT(disease_ct) FROM prediction WHERE disease_ct='N'

GROUP BY heart_rate_ct ";

rs1=st5.executeQuery(sss1);

while (rs1.next())

{

st2.executeUpdate("update heartrate_tab set n_ct="+rs1.getInt(2) +" where heartrate_ct='"+rs1.getString(1) +"' ");

}

//=======================


sss1 = "SELECT chest_pain_ct, COUNT(disease_ct) FROM `prediction` WHERE

disease_ct='Y' GROUP BY chest_pain_ct ";

rs1=st5.executeQuery(sss1);

while (rs1.next())

{

st2.executeUpdate("update chestpain_tab set y_ct="+rs1.getInt(2) +" where

chestpain_ct='"+rs1.getString(1) +"' ");

}


sss1 = "SELECT chest_pain_ct, COUNT(disease_ct) FROM `prediction` WHERE

disease_ct='N' GROUP BY chest_pain_ct ";

rs1=st5.executeQuery(sss1);

while (rs1.next())

{

st2.executeUpdate("update chestpain_tab set n_ct="+rs1.getInt(2) +" where
```

```
chestpain_ct='"+rs1.getString(1) +"' ");
}
//=======================



sss1 = "SELECT cholesterol_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='Y' GROUP BY cholesterol_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update cholesterol_tab set y_ct="+rs1.getInt(2) +" where
cholesterol_ct='"+rs1.getString(1) +"' ");
}

sss1 = "SELECT cholesterol_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='N' GROUP BY cholesterol_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update cholesterol_tab set n_ct="+rs1.getInt(2) +" where
cholesterol_ct='"+rs1.getString(1) +"' ");
}

//=======================

sss1 = "SELECT bloodpressure_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='Y' GROUP BY bloodpressure_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
```

```
{
st2.executeUpdate("update bp_tab set y_ct="+rs1.getInt(2) +" where
bp_ct='"+rs1.getString(1) +"' ");
}



sss1 = "SELECT bloodpressure_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='N' GROUP BY bloodpressure_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update bp_tab set n_ct="+rs1.getInt(2) +" where
bp_ct='"+rs1.getString(1)+"' ");
}
//====================================


sss1 = "SELECT bloodsugar_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='Y' GROUP BY bloodsugar_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update bs_tab set y_ct="+rs1.getInt(2) +" where
bs_ct='"+rs1.getString(1) +"' ");
}


sss1 = "SELECT bloodsugar_ct, COUNT(disease_ct) FROM `prediction` WHERE
disease_ct='N' GROUP BY bloodsugar_ct ";
rs1=st5.executeQuery(sss1);
while (rs1.next())
{
st2.executeUpdate("update bs_tab set n_ct="+rs1.getInt(2) +" where
```

```
bs_ct="'+rs1.getString(1) +"' ");
}

}
catch (Exception e1)
{
out.println(e1);
}
response.sendRedirect("age_tab.jsp");
%><br><br><br><br><br><br><br><br><br><br><br><br>
<%@ include file="footer.jsp"%>
```

USER CODE

```
<%
String mm=request.getParameter("id");
if (mm! =null &&mm.equalsIgnoreCase("exp"))
{
out.println("<script type=text/javascript>alert('Sorry, your session expired, login again '); </script>");
}
%>

<%@ include file="header.jsp"%>
<font size="+1">
<h1><font align="center" size="" color="#ff3300">User Page</h1></font><br>
<form method="post" action="ulogin.jsp">
<table align=left width="50%"><tr><td>
<tr><td><div class="input-group input-group-lg">
<span class="input-group-addon" id="sizing-addon1">UID</span>
<input type="text" name ="uid" class="form-control" placeholder="Username" aria-
describedby="sizing-addon1"></div>
```

```
<tr><td><div class="input-group input-group-lg">


<span class="input-group-addon" id="sizing-addon1">***</span>


<input type="password" name ="pwd" class="form-control" placeholder="Password" aria-describedby="sizing-addon1"></div>
<tr><td>       <span class="input-group-btn">
<button class="btnbtn-default" type="submit">Login</button>
                </span>
</div><td>
</tr>
</table>
<table align=right ><tr><td><td>


<%
String m=request.getParameter("id");
if(m!=null &&m.equalsIgnoreCase("fail"))
{
out.println("<font color='red'><blink><h3>   Login Fail !! </blink></font></h3>");
}
%>
</table>


</form>
<br><br><br><br>
<br><br><br><br><br><br>


<%@ include file="footer.jsp"%>
```

CHAPTER-5

RESULTS

5.1 OUTPUT SCREENSHOTS

5.1.1 HOME PAGE

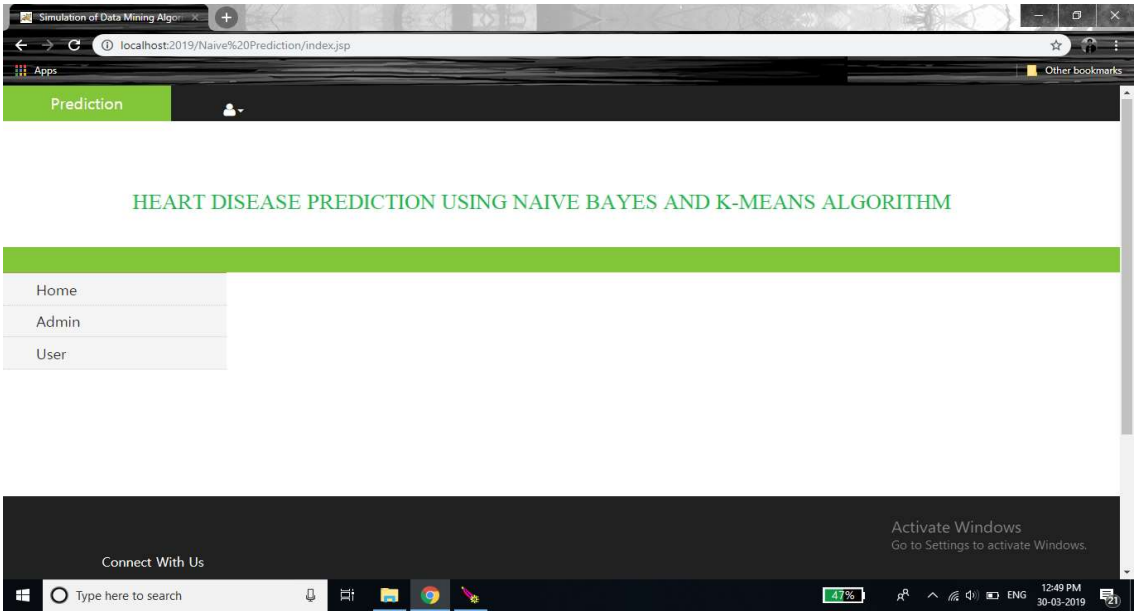

Fig 5.1: Home Page

5.1.2 ADMIN PAGE

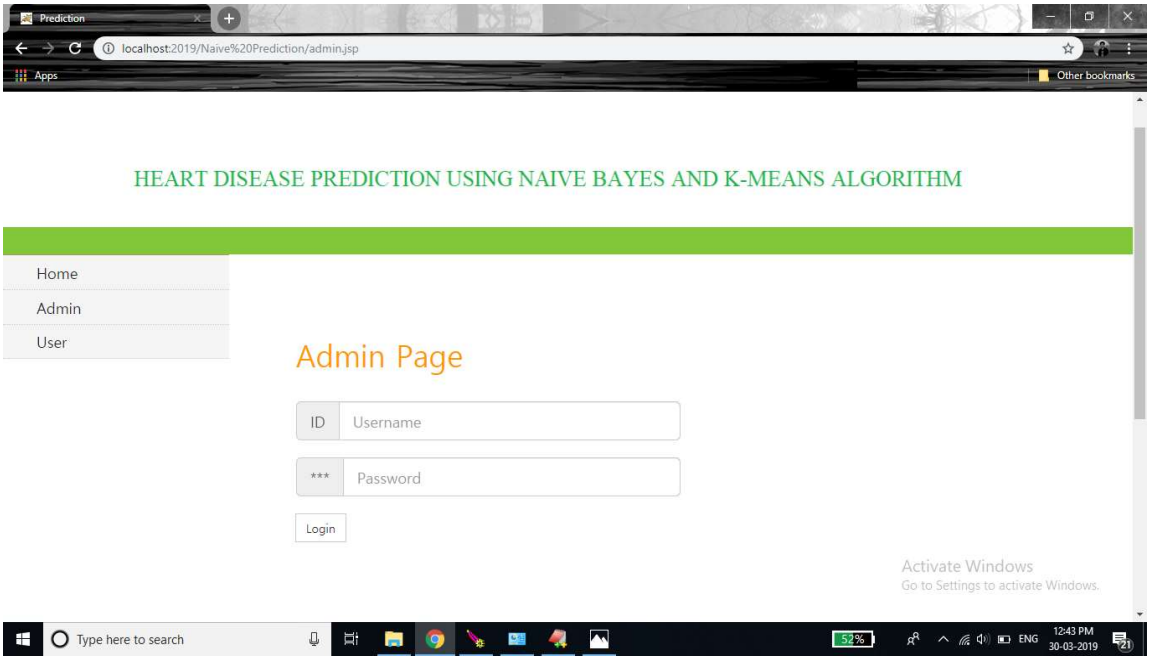

Fig 5.2: Admin Page

5.1.3 DATA SET

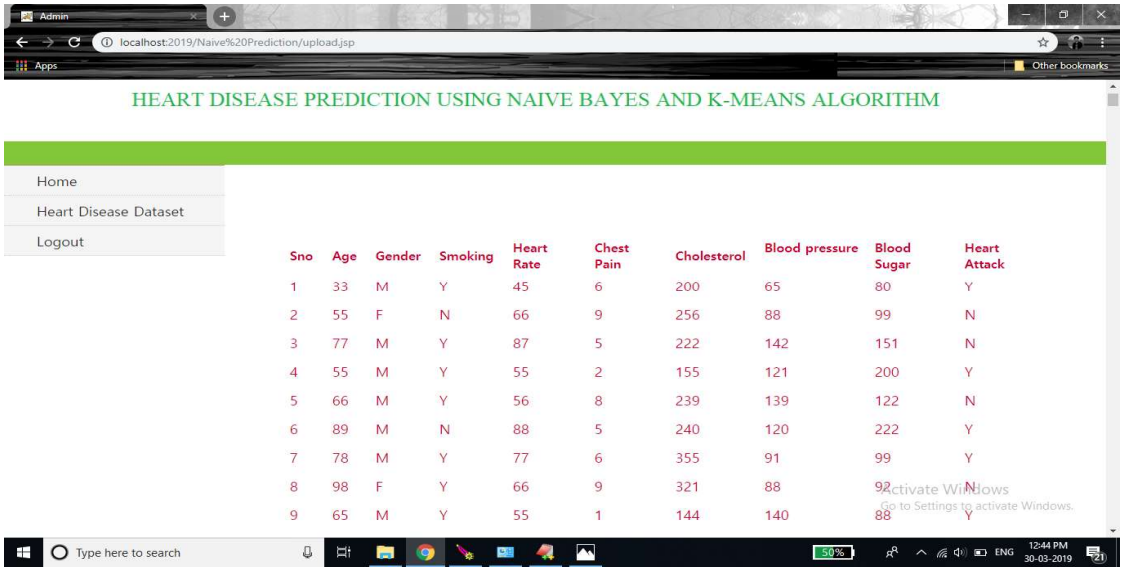

Fig 5.3: Data Set

5.1.4 UPLOADING DATA SET

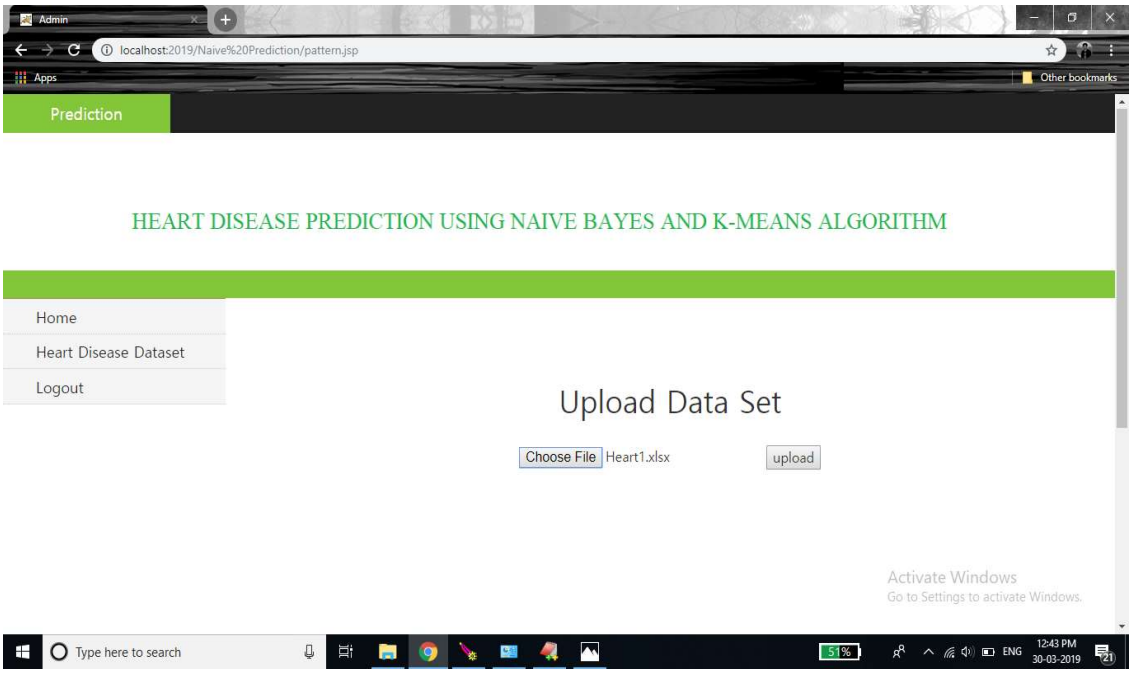

Fig 5.4: Uploading Data Set

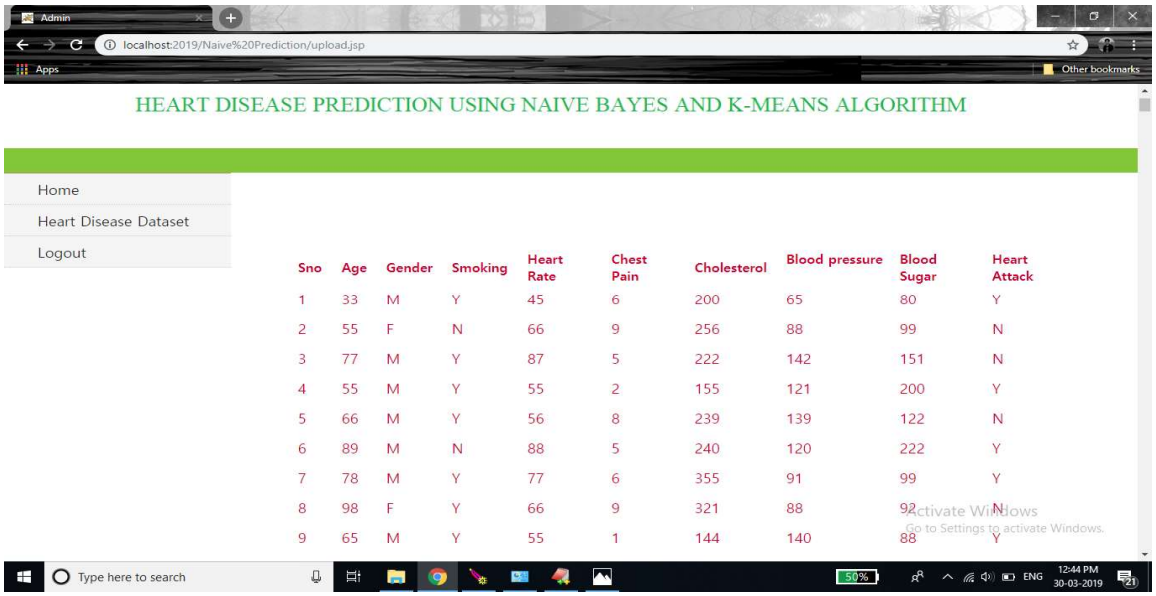

Fig 5.5: Data Set Records

5.1.5 K-MEANS ALGORITHM

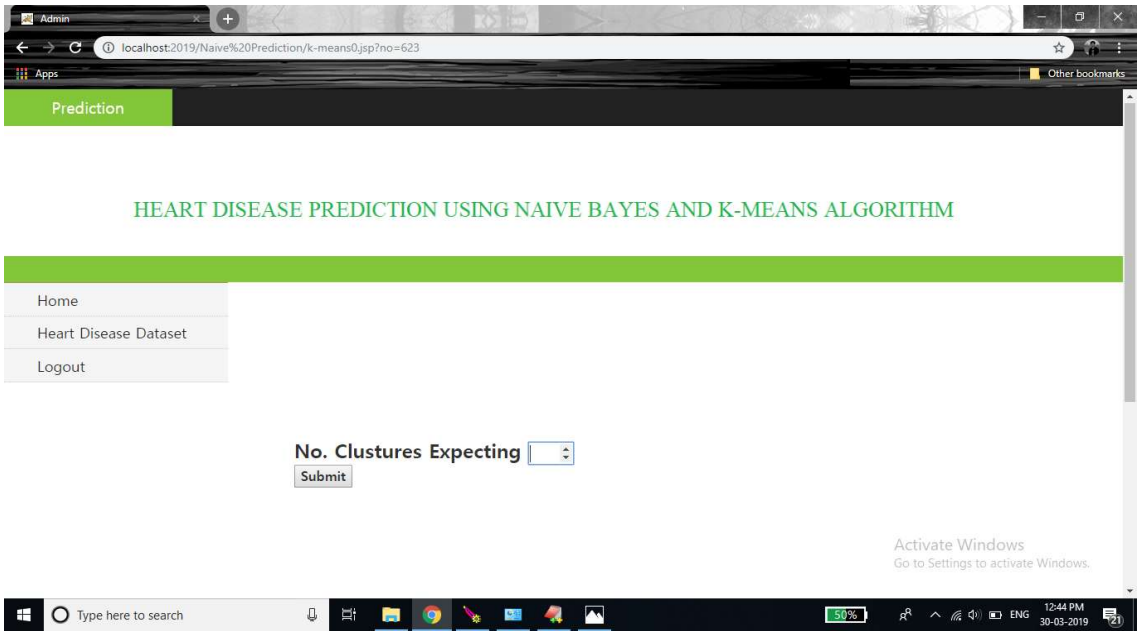

Fig 5.6 : K Means Algorithm

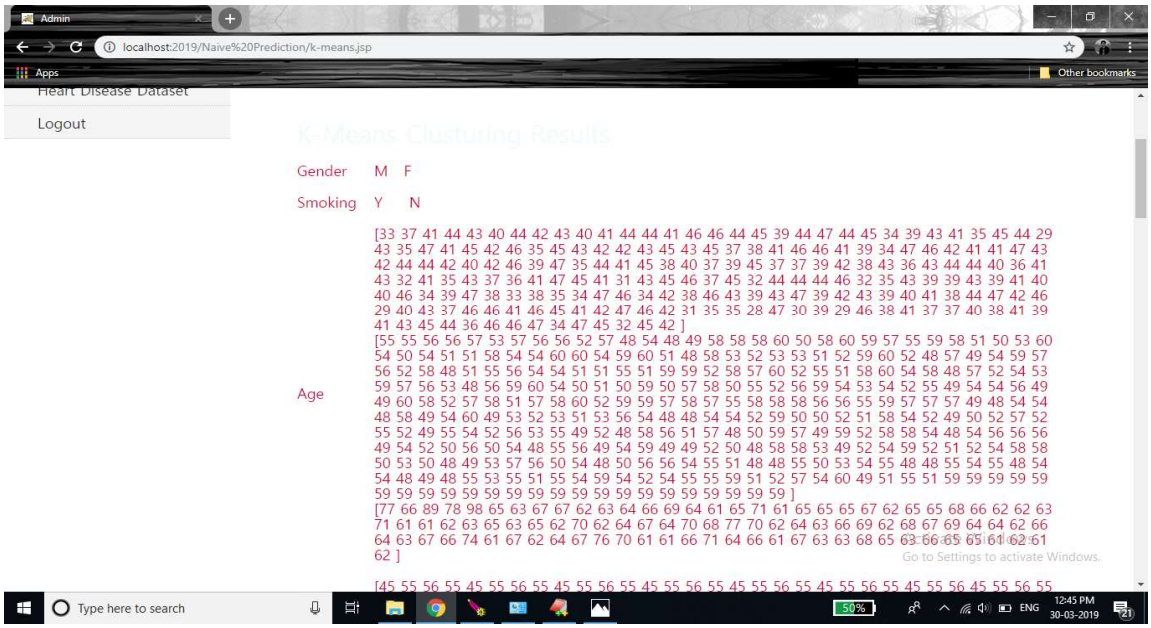

Fig 5.7: Formation of Clusters

5.1.6 NAÏVE BAYES ALGORITHM

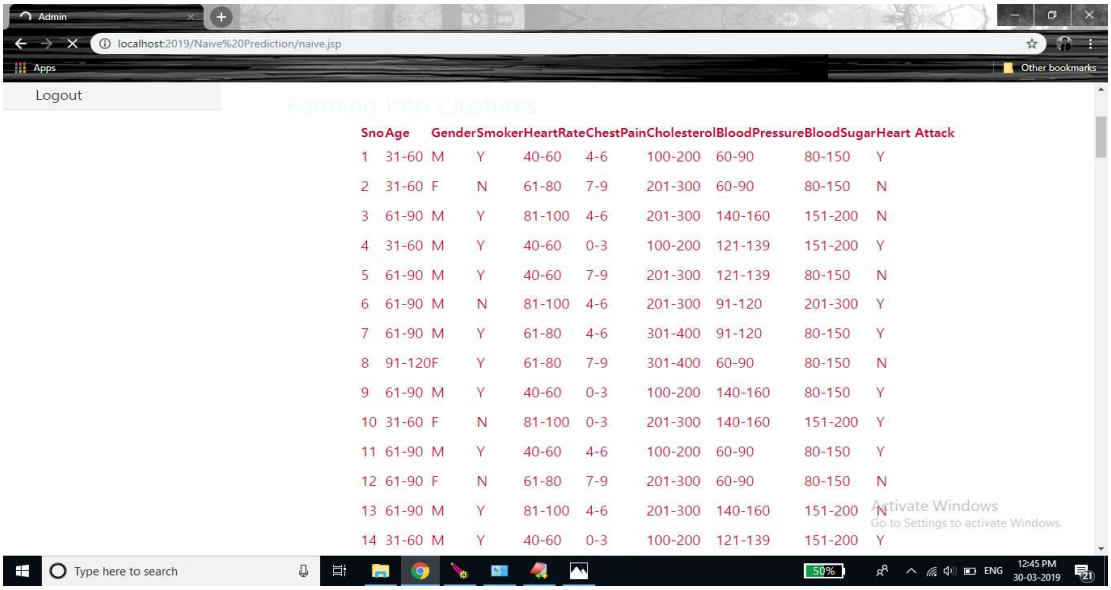

Fig 5.8 :Naive Bayes Algorithm

5.1.7 FREQUENCY TABLES

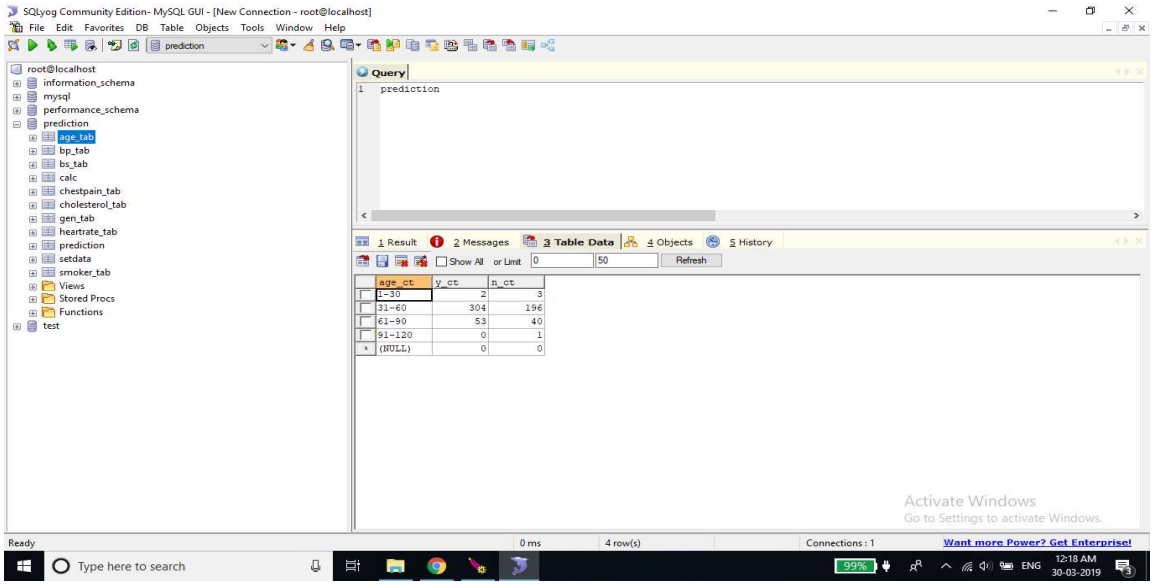

Fig 5.9: Frequency table for Age Data

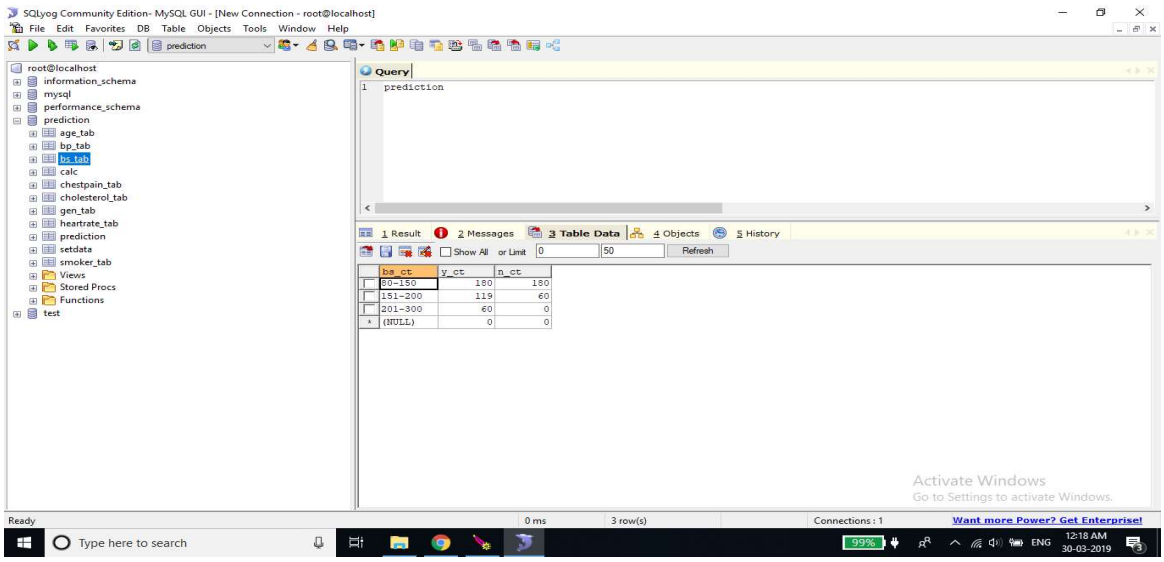

Fig 5.10: Frequency table for Blood Pressure Data

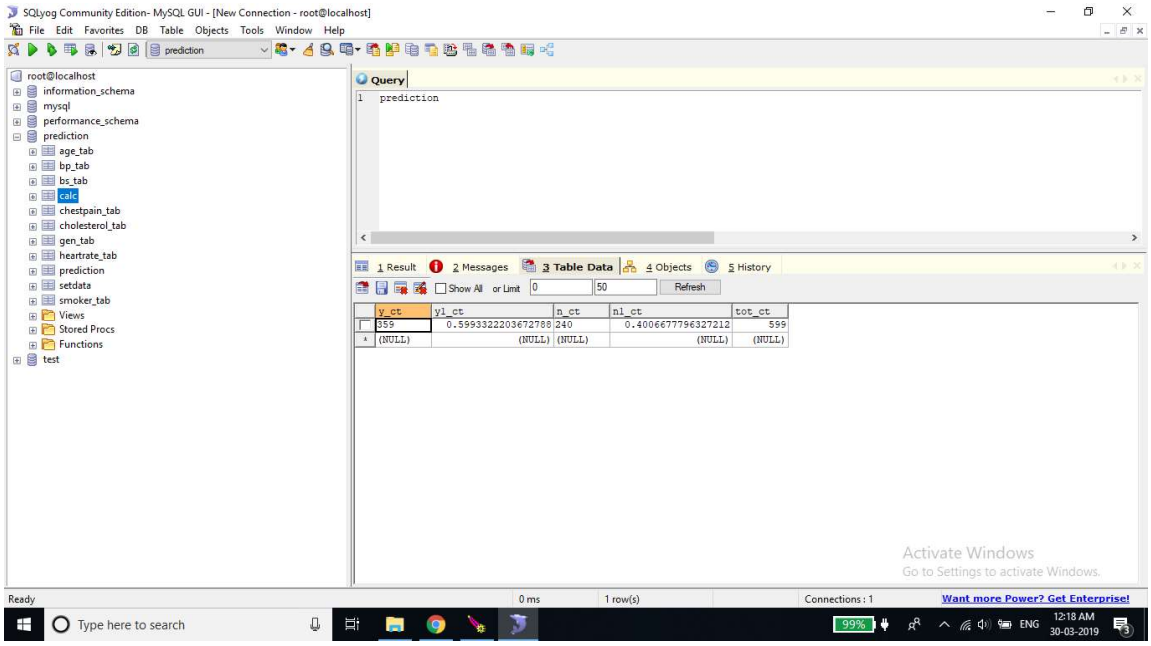

Fig 5.11: Frequency table for calculation

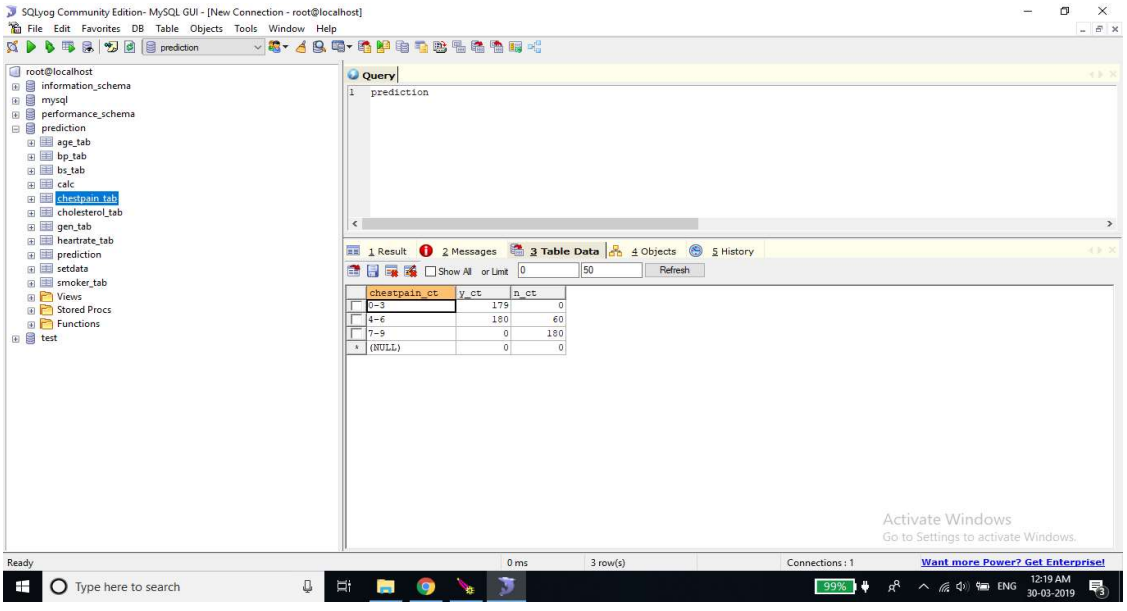

Fig 5.12: Frequency table for Chest Pain Data

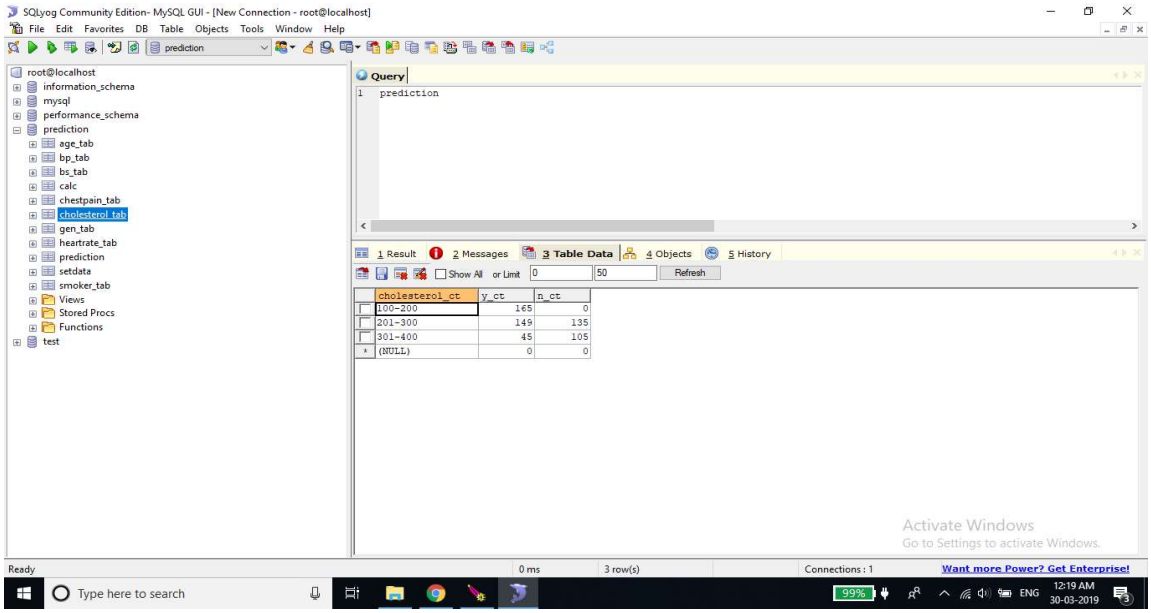

Fig 5.13: Frequency table for Cholesterol Data

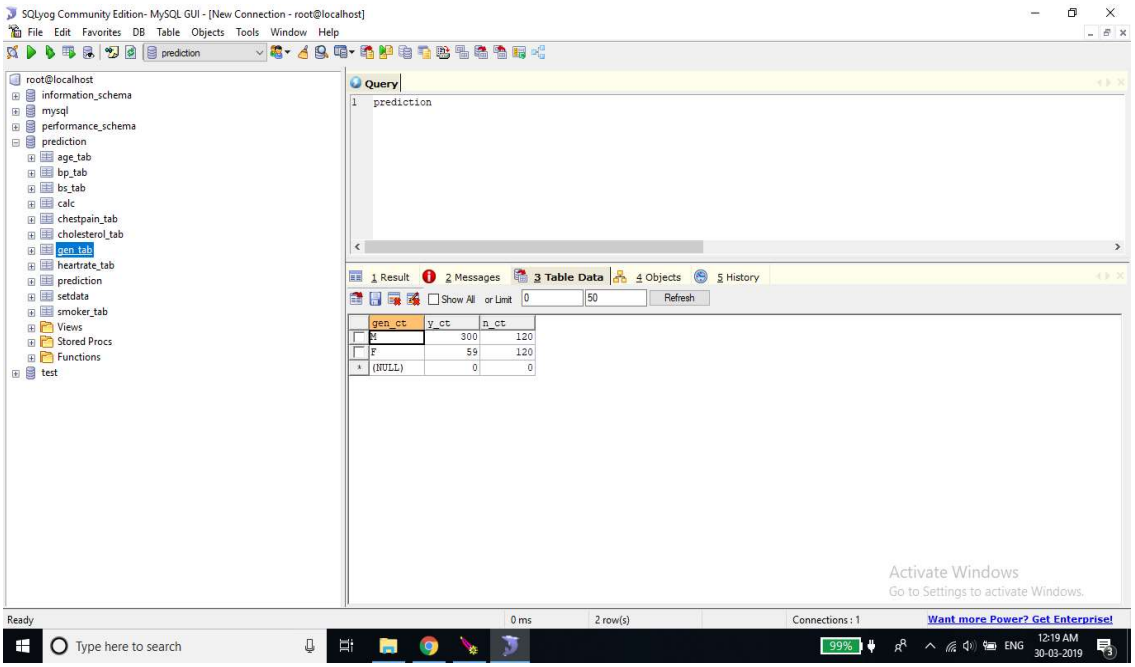

Fig 5.14: Frequency table for Gender Data

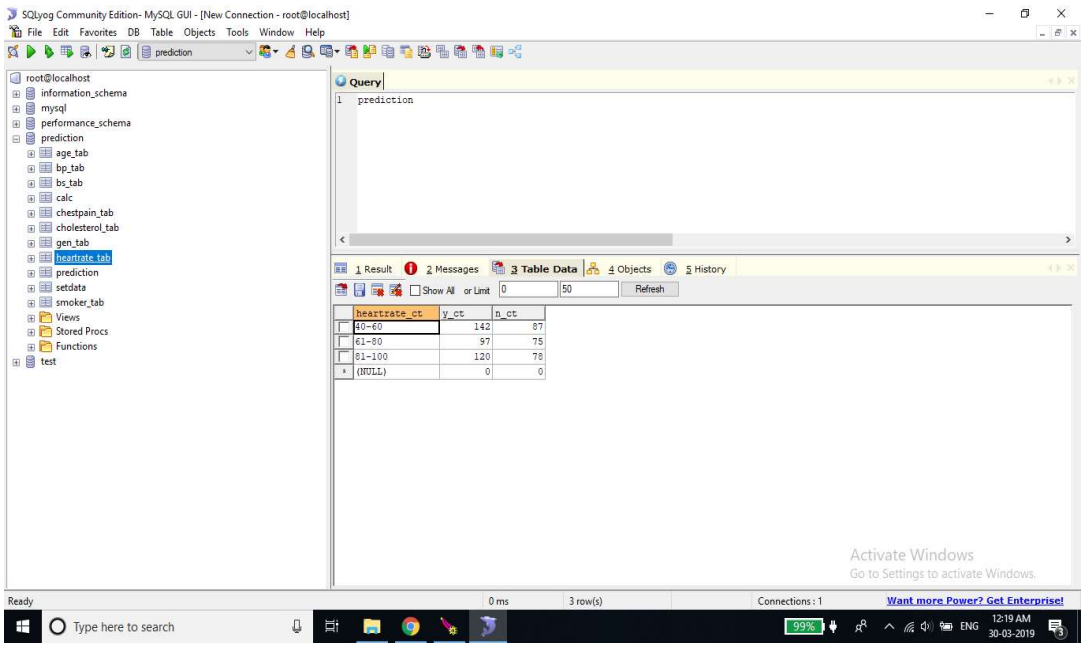

Fig 5.15: Frequency table for Heart rate Data

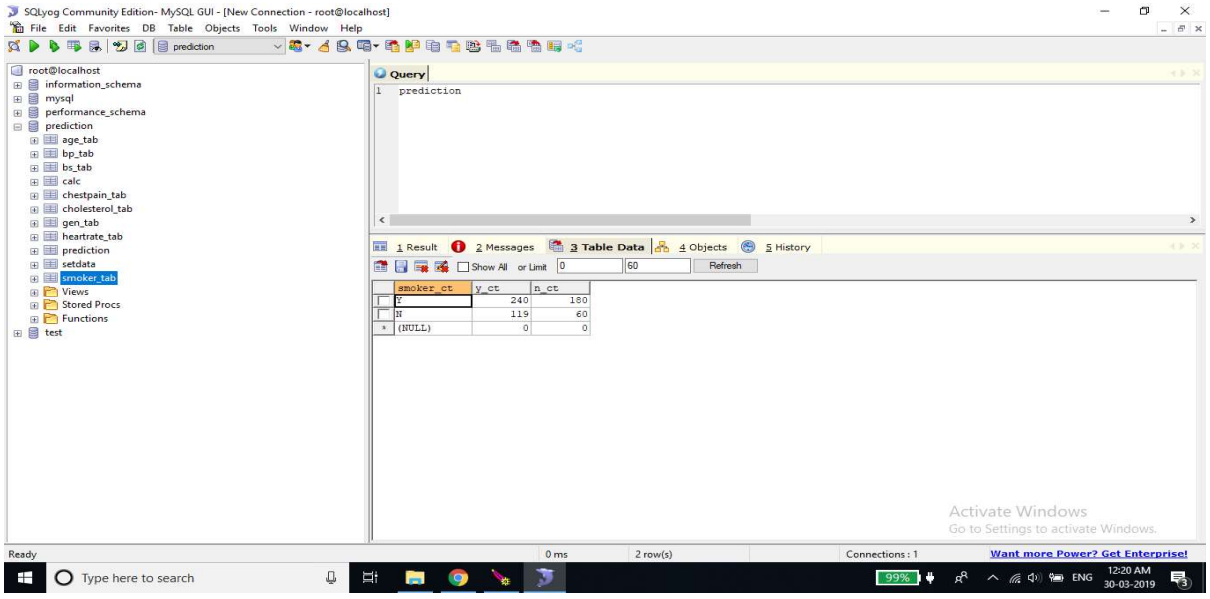

Fig 5.16: Frequency table for Smoker Data

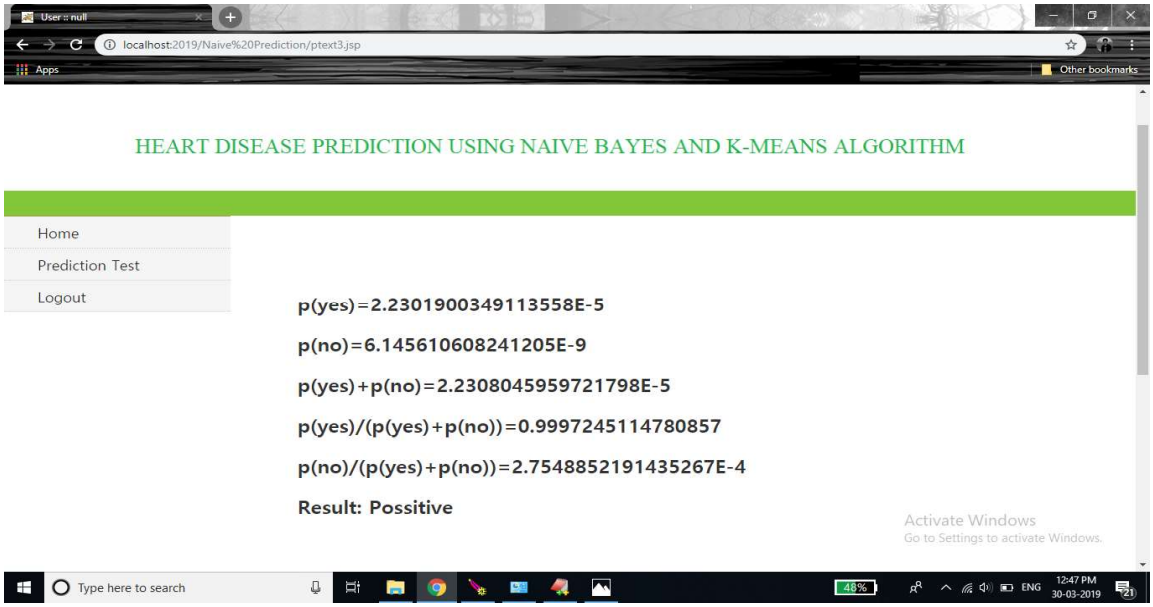

Fig 5.17: Prediction page

5.1.8 USER PAGE

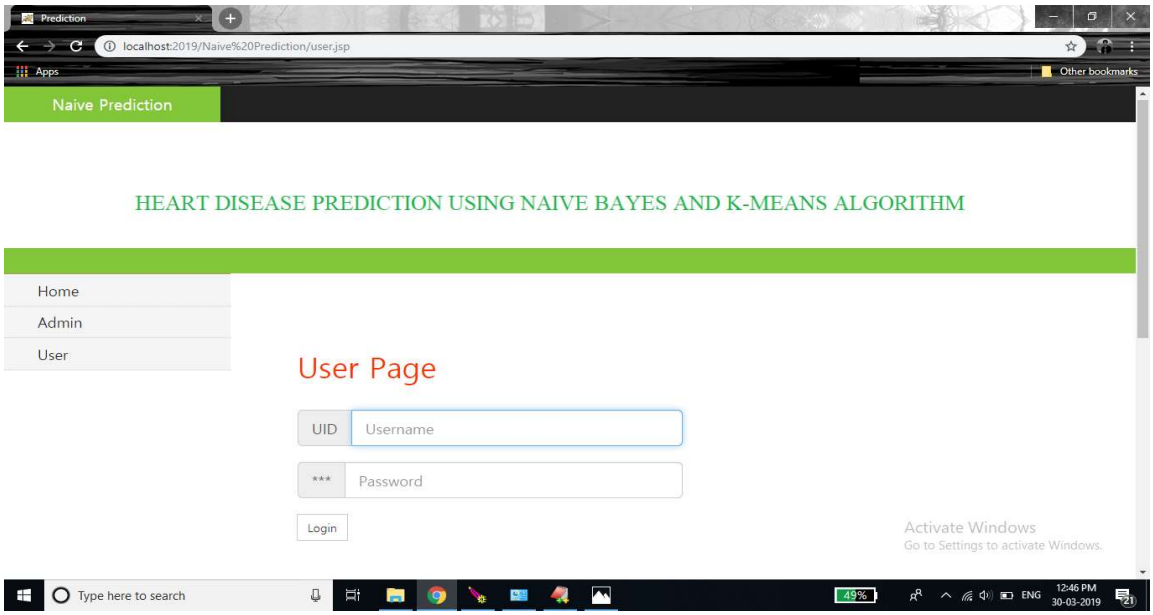

Fig 5.18: User Page

5.1.9USER DETAILS

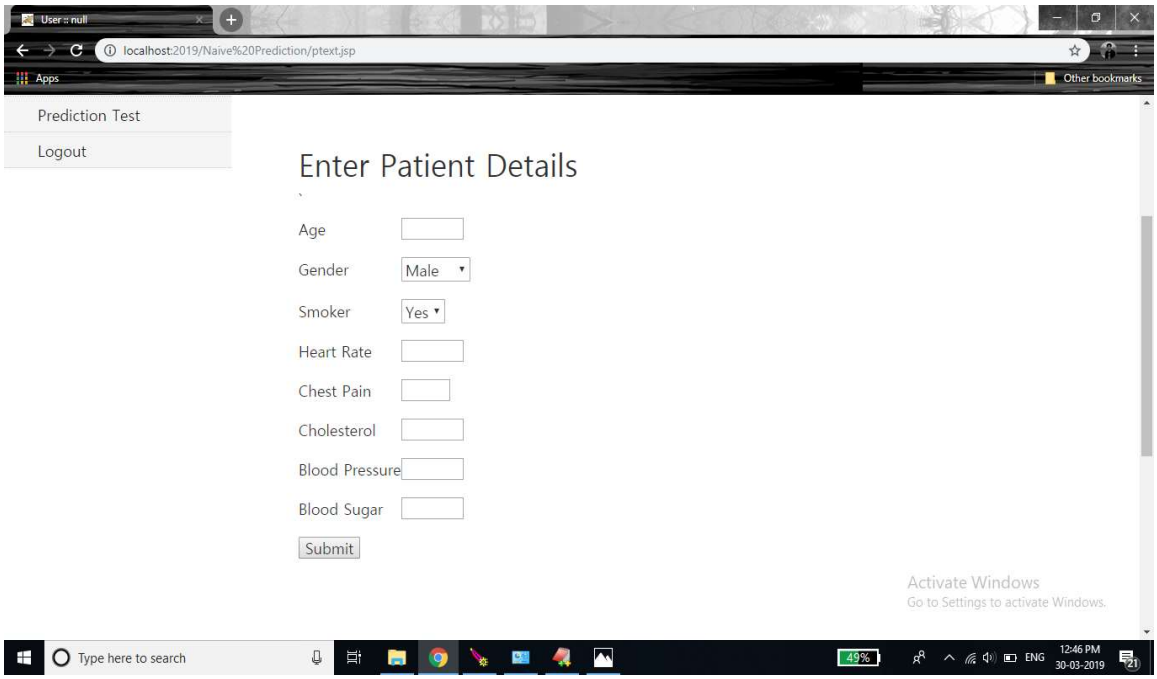

Fig 5.19: Input User Details

5.2.0 FINAL RESULT

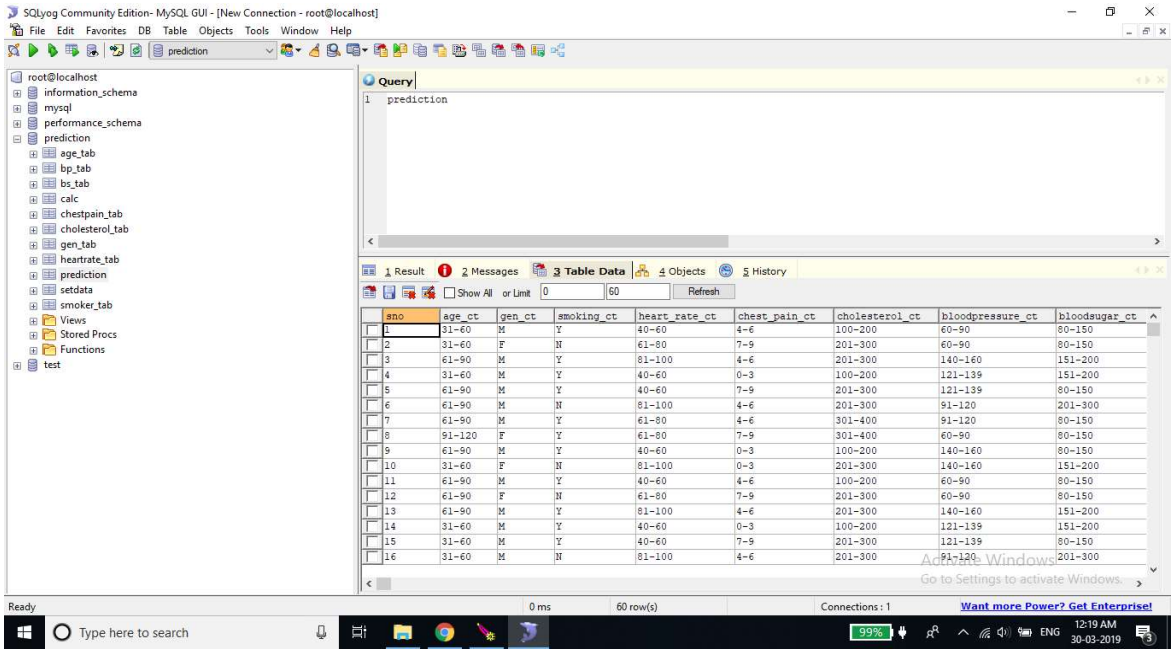

Fig 5.20: Final Result

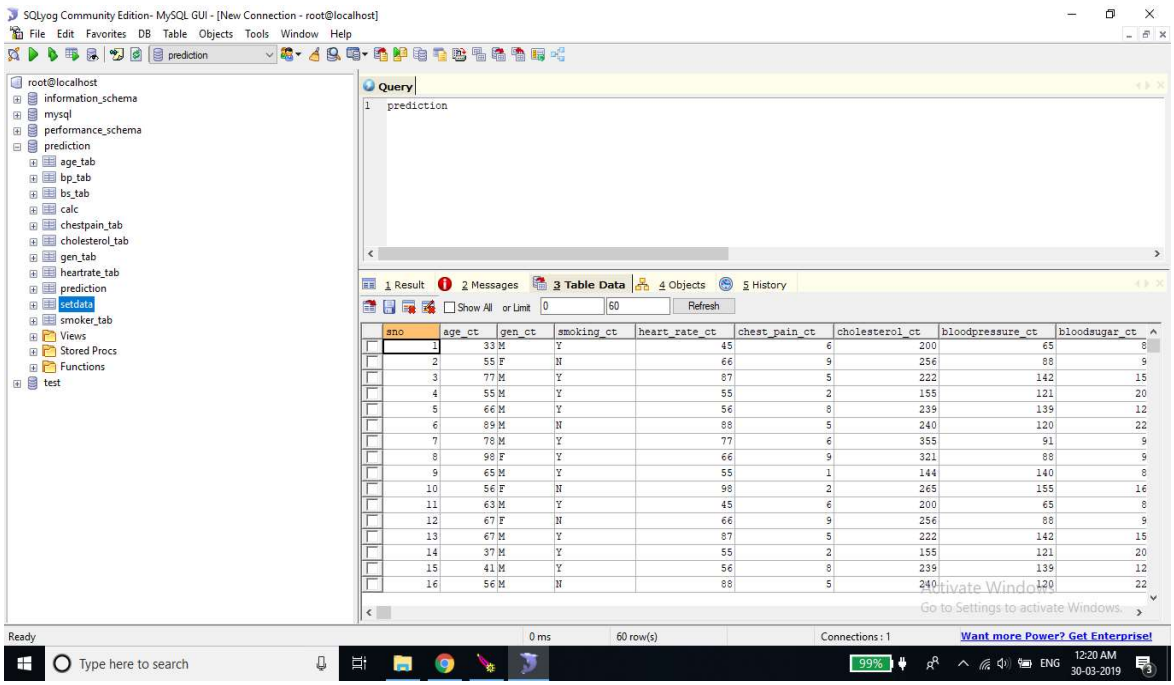

Fig 5.21: Frequency table for Whole Data

CHAPTER-6

CONCLUSION

Today exists many expert systems that operate on the basis of different methods and are used in many areas of medicine. However, a common feature of medical expert systems can be identified with lack of unified technology of their creation for the most part, in development and in the full operating systems, based on different operation algorithms. In this article, we show the effectiveness of unsupervised learning techniques of data, which used K-means clustering algorithm to improving supervised learning technique which is Naive Bayes. It explores the integration of K-means clustering with Naive Bayes in the structure of the decision-making system in medicine. It also investigates different methods of initial centroid selection of the K-means clustering such as range, random attribute values and random row methods in the diagnosis of the patients. The results indicate that the integration of the K-means clustering with Naïve Bayes algorithm with different initial selection can improve the accuracy in diagnosis of patient. Also, Big Data processing through artificial agents and human's knowledge-based shows, and their integration we represented. Data mining approach for processing Big Data can solve many problems of analysing large and growing data sets.

We are proposing heart disease prediction system using Naïve Bayes. We are using Naïve Bayes clustering for increasing the efficiency of the output. This is the most effective model to predict patients with heart disease. This model could answer complex queries, each with its own strength with respect to ease of model interpretation, access to detailed information and accuracy. By using this algorithm, it is easy to predict the disease and admin are allowed to operate and perform with ease. This process can be predicted from any place through this web source. The system is designed to be user friendly. During the research an intelligent decision support system model had been created. Then, for future work, we will improve this intelligent decision-making system by using other new models and apply them to other environments.

CHAPTER-7

REFERENCES

[1] SellappanPalaniappan, RafiahAwang “Intelligent Heart Disease Prediction System Using Data Mining Techniques” Department of Information Technology Malaysia University of Science and Technology Block C, Kelana Square, Jalan SS7/26 Kelana Jaya,47301 Petaling Jaya, Selangor, Malaysia.

[2] "CSV File Reading and Writing" (http:/ / docs. python. org/ library/csv. html). Retrieved July 24, 2011. "is no "CSV standard""

[3] Y. Shafranovich. "Common Format and MIME Type for Comma-Separated Values (CSV) Files" (http:/ / tools. ietf. org/ html/rfc4180) Retrieved September 12, 2011.

[4] JesminNahar, TasadduqImama, Kevin S. Tickle, Yi-PingPhoebe Chen “Association rule mining to detect factors which contribute to heart disease in males and females” Expert Systems

[5] Shadab Adam Pattekari and AsmaParveen “Prediction Systemfor Heart Disease Using Naïve Bayes” International Journal of Advanced Computer and Mathematical Sciences ISSN 2230-9624.Vol 3, Issue 3, 2012, pp 290-294.

[6] Mrs.GSubbalakshmi (M.Tech), Mr. K. Ramesh M.Tech, Asst.Professor Mr.M. ChinnaRao, M.Tech, (Ph.D.) Asst. Professor, “Decision Support in Heart Disease Prediction System using Naïve Bayes” G. Subbalakshmi et al. / Indian Journal of Computer Science and Engineering (IJCSE)2011.