

HEART DISEASE PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award
of
Bachelor of ENGINEERING degree
in
COMPUTER SCIENCE AND ENGINEERING

By

Hruthik (38110550)



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119**

APRIL - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Hruthik (38110550)**

who carried out the project entitled "**HEART DISEASE PREDICTION USING MACHINE LEARNING**" under my supervision from September, 2020 to APRIL, 2021.

Internal Guide

Dr. L. BHARATHI, M.E., Ph.D.,

Head of the Department

Dr. L. Lakshmanan M.E., Ph.D.,

Dr. S. Vigneshwari M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Hruthik (38110550)**

hereby declare that the Project Report entitled “**HEART DISEASE PREIDITION USING MACHINE LEARNING**” done by me under the guidance of **Dr. BHARATHI, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF CANDIDATE:

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. T. SASIKALA ME., Ph.D., Dean, School OF COMPUTING, Dr. S. Vigneshwari, ME., Ph.D. and Dr. L. Lakshmanan, M.E.,Ph.D., Heads of the Department,** Department of Computer Science and Engineering for providing us the necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. BHARATHI** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTRACT

In today's era deaths due to heart disease has become a major issue approximately one person dies per minute due to heart disease. This is considering both male and female category and this ratio may vary according to the region also this ratio is considered for the people of age group. This does not indicate that the people with other age group will not be affected by heart diseases. This problem may start in early age group also and predict the cause and disease is a major challenge nowadays. Here in this paper, we have discussed various algorithms and tools used for prediction of heart diseases.

.

CONTENTS

CHAPTER	TITLE	PAGE.NO
1	INTRODUCTION	1
1.1	OBJECTIVE	2
1.2	EXISTIN SYSTEM	2
1.2.1	DISADVANTAGES	3
1.3	PROPOSED SYSTEMS	3
1.3.1	ADVANTAGES	4
1.4	ARCHITECTURE DIAGRAM	4
1.5	SYSTEM REQUIREMENTS	4
1.5.1	HARDWARE REQUIREMENTS	4
1.5.2	SOFTWARE REQUIREMENTS	5
1.6	INTRODUCTIN TO JAVA	5
1.6.1	APPLICATION OF JAVA	5
1.6.2	FEATURES OF JAVA	6
1.6.3	COLLECTION FRAME WORK	8
1.7	MYSQL	9
1.7.1	SQL SERVER MANAGEMENT STUDIO	10
1.7.2	CREATE A NEW DATABASE	12
1.7.2.1	TABLE	12
1.7.2.2	QUERY	13
2	LITERATURE SURVEY	14
3	AIM AND SCOPE	17
4	METHODOLOGY	19
4.1	NAÏVE BAYES ALGORITHM	19
4.1.1	NAÏVE BAYES ALGORITHM WORKING	20
4.1.2	PROS AND CONS OF NAÏVE BAYES	22
4.1.3	APPLICATON OF NAÏVE BAYES ALGORITHM	22
4.2	NEAREST NEIGHBOUR(KNN) CLASSIFICATION METHOD	23
4.3	MODULES	24

4.3.1	USER MODULE	24
4.3.1.1	LOGIN	24
4.3.1.2	REGISTRATION	24
4.3.2	ADMIN MODULE	25
4.3.3	DISEASE ANALYSIS MODULE	25
4.3.4	DISEASE PREDICTION MODULE	25
4.4	DATA FLOW DIAGRAM	26
4.4.1	ROLE OF DFD	26
5	UML DIAGRAMS	27
5.1	UML DIAGRAMS	27
5.1.1	USECASE DIAGRAM	28
5.1.2	CLASS DIAGRAM	28
5.1.3	ACTIVITY DIAGRAM	29
5.1.4	SEQUENCE DIAGRAM	30
5.1.5	COLLABORATION DIAGRAM	30
5.2	REQUIREMENT ANALYSIS	31
5.3	FUNCTIONAL REQUIREMENTS	31
5.3.1	USABILITY	31
5.3.2	ROBUSTNESS	31
5.3.3	SECURITY	31
5.3.4	RELIABILITY	32
5.3.5	COMPAITIBILITY	32
5.3.6	FLEXIBILITY	32
5.3.7	SAFETY	32
5.4	NON-FUNCTIONAL REQUIREMENTS	32
5.4.1	PORTABILITY	32
5.4.2	PERFORMANCE	32
5.4.3	ACCURACY	32
5.4.4	MAINTAINABILITY	33

6	RESULTS AND DISCUSSION	34
	REFERENCES	36
	A SOURCE CODE	37

LIST OF FIGURES		
FIGURE NO	FIGURE NAME	PAGE NO
1.1	ARCHITECTURE DIAGRAM	4
1.2	PROCESSING OF J2SE	7
1.3	TOOLS USED IN MYSQL	10
1.4	SQL SERVER MANAGEMENT STUDIO	11
1.5	DATA BASE CREATION	12
5.1	USE CASE DIAGRAM	28
5.2	CLASS DIAGRAM	28
5.3	ACTIVITY DIAGRAM	29
5.4	SEQUENCE DIAGRAM	30
5.5	COLLABORATION DIAGRAM	30
6.1	DATA ENTRY PAGE	34
6.2	QUICK TEST MODULE	35
6.3	REPORT RESULT	35

CHAPTER-1

INTRODUCTION

The content is mainly focus on various data mining practices that are valuable in heart disease forecast with the assistance of dissimilar data mining tools that are accessible. If the heart doesn't function properly, this will distress the other parts of the human body such as brain, kidney etc. Heart disease is a kind of disease which effects the functioning of the heart. In today's era heart disease is the primary reason for deaths. WHO-World Health Organization has anticipated that 12 million people die every year because of heart diseases. Some heart diseases are cardiovascular, heart attack, coronary and knock. Knock is a sort of heart disease that occurs due to strengthening, blocking or lessening of blood vessels which drive through the brain or it can also be initiated by high blood pressure. The major challenge that the Healthcare industry faces now-a-days is superiority of facility. Diagnosing the disease correctly & providing effective treatment to patients will define the quality of service. Poor diagnosis causes disastrous consequences that are not accepted. Records or data of medical history is very large, but these are from many dissimilar foundations. The interpretations that are done by physicians are essential components of these data. The data in real world might be noisy, incomplete and inconsistent, so data preprocessing will be required in directive to fill the omitted values in the database. Even if cardiovascular diseases is found as the important source of death in world in ancient years, these have been announced as the most avoidable and manageable diseases. The whole and accurate management of a disease rest on on the well-timed judgment of that disease. An correct and methodical tool for recognizing high-risk patients and mining data for timely analysis of heart infection looks a serious want. Different person body can show different symptoms of heart disease which may vary accordingly. Though, they frequently include back pain, jaw pain, neck pain, stomach disorders, and tininess of breath, chest pain, arms and shoulders pains. There are a variety of different heart diseases which includes heart failure and stroke and coronary artery disease. Heart expert"s create a good and huge record of patient's database and store them. It also delivers a great prospect for mining a valued knowledge from such sort of datasets.

There is huge research going on to determine heart disease risk factors in different patients, different researchers are using various statistical approaches and numerous programs of data mining approaches. Statistical analysis have acknowledged the count of risk factors for heart diseases counting smoking, age, blood pressure, diabetes, total cholesterol, and hypertension, heart disease training in family, obesity and lack of exercise. For prevention and healthcare of patients who are about to have addicted of heart disease it is very important to have awareness of heart diseases.

OBJECTIVE

The analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease outbreaks. However, those existing work mostly considered structured data. There is no proper methods to handle semi structured and unstructured. The proposed system will consider both structured and unstructured data. The analysis accuracy is increased by using Machine Learning algorithm.

EXISTING SYSTEM

The before all existing system works on sets of both Deep learning and data mining. The existing system modules generates comprehensive report by implementing the strong prediction algorithm. The main aims of the existing system to compare and check the before patient whose having disease outputs and new patient disease and determine future possibilities of the heart disease to a particular patient. By Implementing the above mentioned model we will get the goal of developing a system with increased rate of accuracy of estimating the new patient getting heart attack percentage. The model which is proposed for Heart Attack Prediction System is invented for using Deep learning algorithms and approach. But by using all the existing systems the accuracy is very less.

DISADVANTAGES

- Does not analyse the disease
- There is no automatic drugs
- Less security
- There is no feedback system

PROPOSED SYSTEM

This proposed system have a data which classified if patients have heart disease or not according to features in it. This proposed system can try to use this data to create a model which tries predict (reading data and data Exploration) if a patient has this disease or not. In this proposed system, use logistic regression (classification) algorithm. Implements Naive Bayes algorithm to getting accuracy result. Finally analysing the results by the help of Comparing Models and Confusion Matrix. From the data we are having, it should be classified into different structured data based on the features of the patient heart. From the availability of the data, we have to create a model which predicts the patient disease using logistic regression algorithm. First, we have to import the datasets. Read the datasets, the data should contain different variables like age, gender, sex, cp (chest pain), slope, target. The data should be explored so that the information is verified. Create a temporary variable and also build a model for logistic regression. Here, we use sigmoid function which helps in the graphical representation of the classified data. By using logistic regression, naïve Baye's the accuracy rate increases.

ADVANTAGES

- Easily analyse the disease
- Also provides the drugs based on disease

ARCHITECTURE DIAGRAM

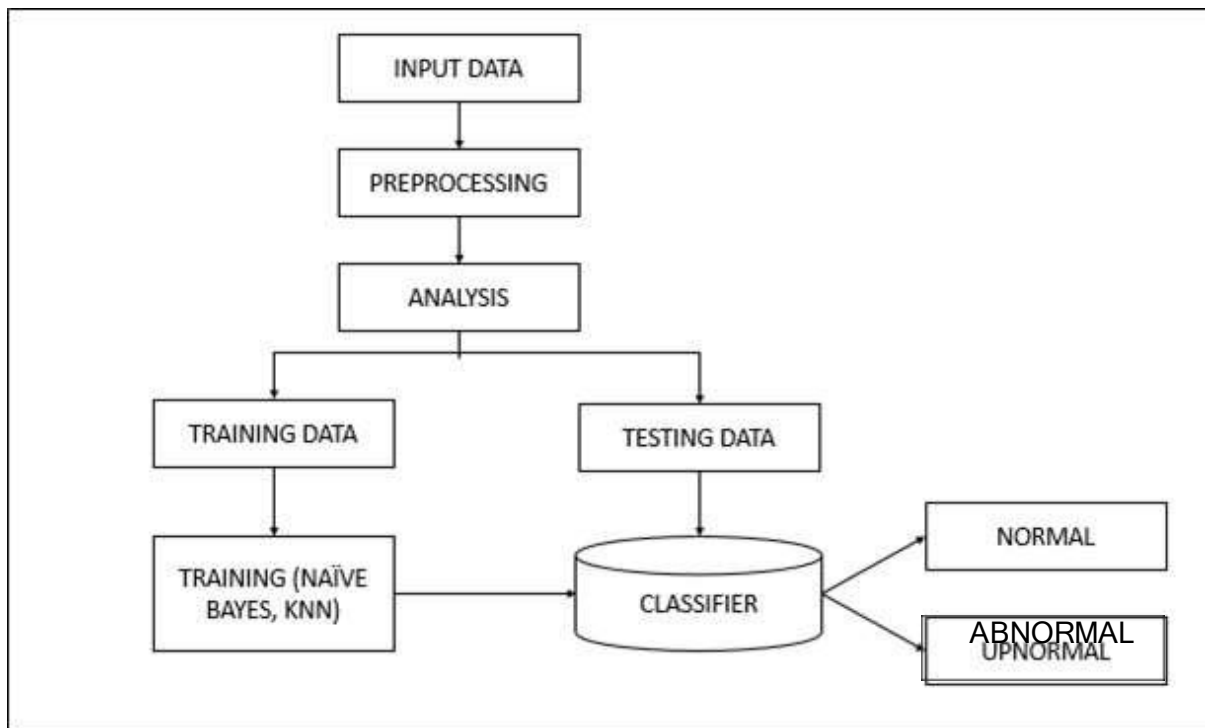


Fig 1.1: Architecture Diagram

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- Processor : Core i3/i5/i7
- RAM : 2-4GB
- HDD : 500 GB

SOFTWARE REQUIREMENTS

- Platform : Windows XP /7/8
- Language : Java
- Front End : Netbeans
- Back End : MYSQL

INTRODUCTION

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown.

APPLICATION OF JAVA

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

1.6.2 Features of java

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

3) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language. On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format

also provide security. Any machine with Java Runtime Environment can run Java Programs.

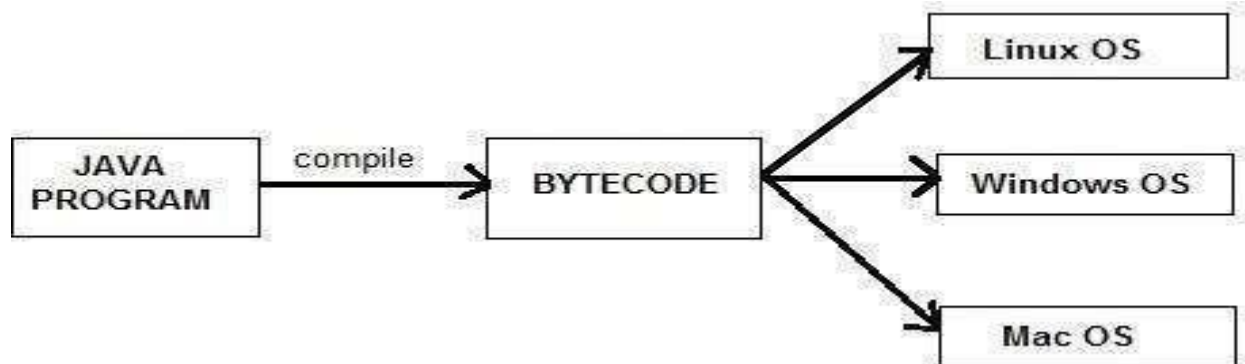


Fig 1.2: Processing of J2SE

5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

7) Portable

Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types.

8) High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

COLLECTION OF FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of a like objects.

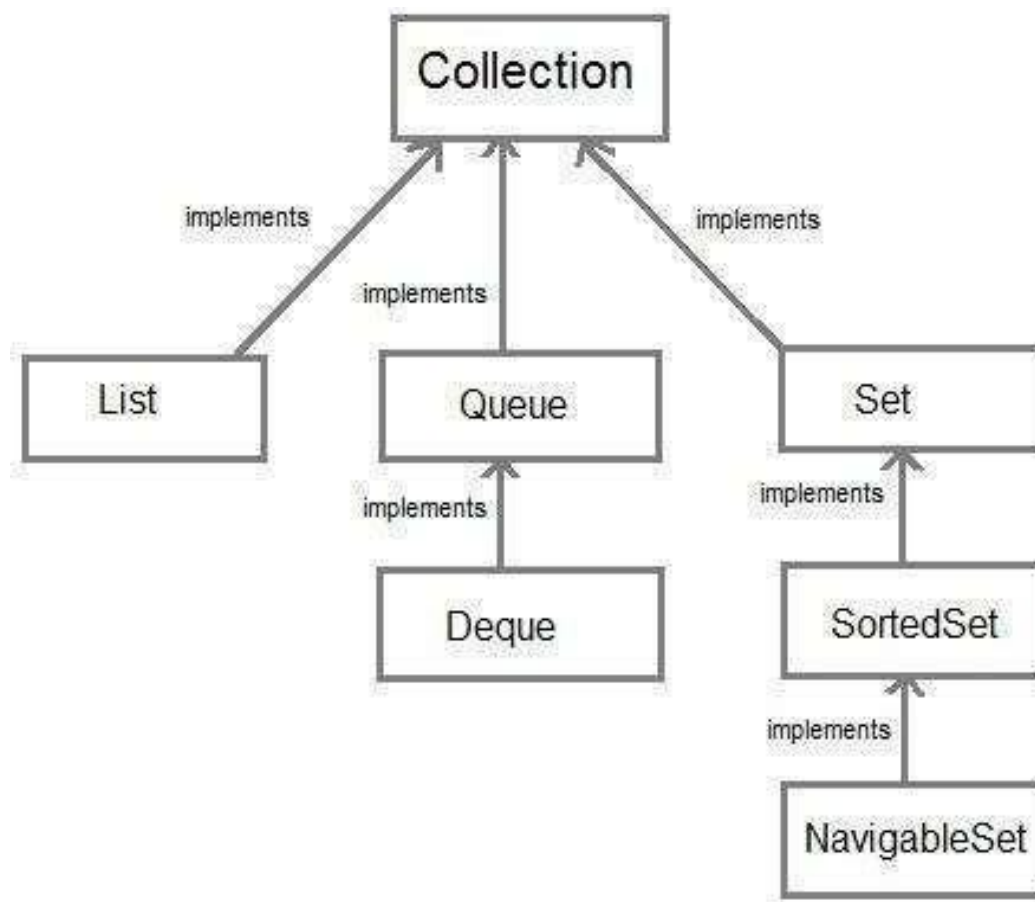


Fig 1.3: Flow chart of collection frame work

MYSQL

MySQL, officially, but also called "My Sequel" is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks).

LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Freesoftware-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr and YouTube. Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use. SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. TSQL is very similar to standard SQL, but in addition it supports some extra functionality, built in functions, etc.

T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, date processing, mathematics, etc. SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your

computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).

SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

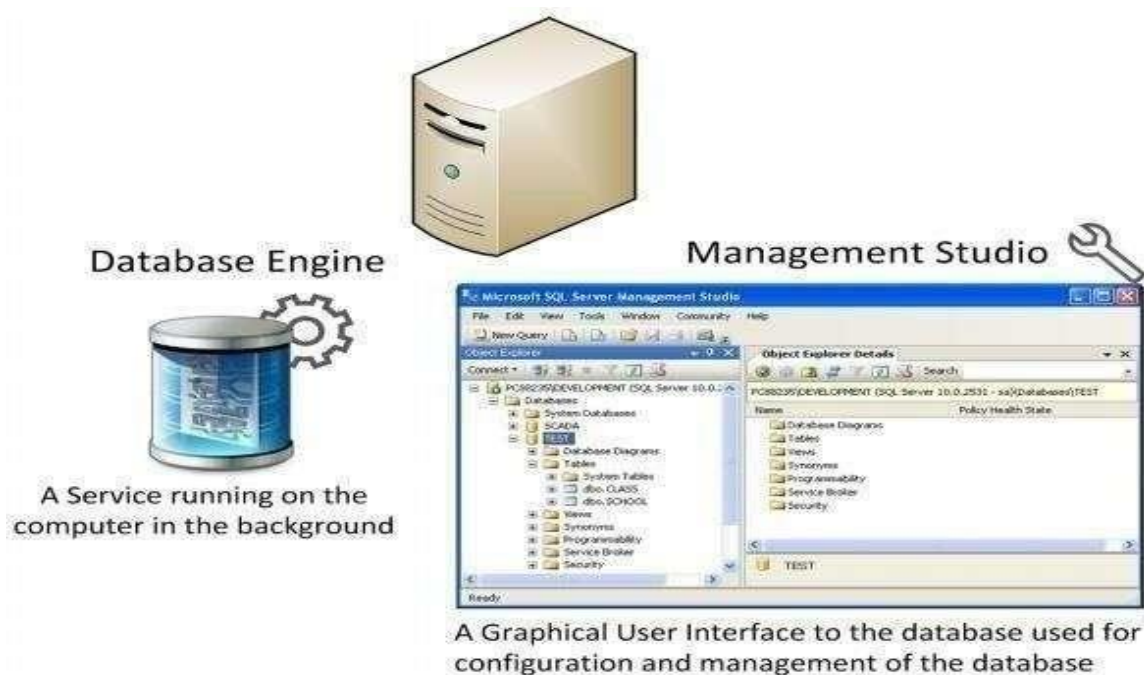


Fig 1.3: Tools used in MYSQL

SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier, version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express.

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

When creating SQL commands and queries, the “Query Editor” (select “New Query” from the Toolbar) is used (shown in the figure above). With SQL and the “Query Editor” we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).



Fig 1.4: SQL server management studio

CREATE A NEW DATABASE

It is quite simple to create a new database in Microsoft SQL Server. Just right- click on the “Databases” node and select “New Database...”



Fig 1.5: Data base creation

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects, they are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

Table

A database is a collection of data about a specific topic.

a. Views of Table

We can work with a table in two types,

1. Design View
2. Datasheet View

1. Design View

To build or modify the structure of a table we work in the table design view.

We can specify what kind of data will be hold.

2. Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

Query

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dyna set (if you edit it) or a snapshot (it cannot be edited). Each time we run query, we get latest information in the dyna set. Access either displays the dyna set or snapshot for us to view or perform an action on it, such as deleting or updating.

CHAPTER-2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

Spectral data have been widely used to estimate the disease severity levels of different plants. However, such data have not been evaluated to estimate the disease stages of the plant. This study aimed at developing a spectral disease index that is able to identify the stages of wheat leaf rust disease at various DS levels. To meet the aim of the study, the reflectance spectra of infected leaves with different symptom fractions and DS levels were measured with a spectro radiometer. Then, pure spectra of the different disease symptoms at the leaf scale were analyzed, and a new function was developed to find the wavelengths most sensitive to disease symptom fraction. The reflectance spectra with highest sensitivity were found at 675 and 775 nm. Finally, the normalized difference of DS and the ratio p_{675}/p_{775} was used as a new SDI to discriminate three different levels of the disease stage at the canopy level. The suggested SDI showed a promising performance to improve the detection disease stages in precision plantprotection.

S.NO	TITLE	AUTHOR	WORKDONE	LIMITATION
1	Machine learning-based method for personalized and cost-effective detection of Alzheimer's disease	Escudero J¹, Ifeachor E, Zajicek JP, Green C, Shearer J, Pearson S	The approach performed similarly to considering all data at once, while significantly reducing the number of the biomarkers needed to achieve a confident diagnosis for each patient. Thus, it may contribute to a personalized and effective detection of AD, and may prove useful in clinical settings.	High cost.
2	Effect of Meteorological Conditions on Occurrence of Hand, Foot and Mouth Disease in Wuwei City, Northwestern China	Shan Zheng ; Minzhen Wang ; Shigong Wang ; Kezheng Shang ; Lili Hu ; Jinrong Dong	The main objective of this paper is to supply scientific basics for preventing and forecasting the prevalence of hand, foot and mouth disease to explore the effect of different meteorological conditions on occurrence of hand, foot and mouth disease in Wuwei City, northwestern China. Here the data about the diseases and weather was collected from 2008-2010, and the correlation analysis, multiple linear regression and exponential curve fitting methods were made.	Difficult to handle this process.
3	Developing an Index for Detection and Identification of Disease Stages	Zhang Y1, Long JD, Mills JA, Warner JH, Lu W, Paulsen JS	Spectral data have been widely used to estimate the disease severity levels of different plants. However, such data have not been evaluated to estimate the disease stages of the plant. This study aimed at developing a spectral disease index that is able to identify the stages of wheat leaf rust disease at various DS levels.	Limitation of this system, such data have not been evaluated to estimate the disease stages of the plant.

4	Quantized Analysis for Heart Valve Disease based on Cardiac Sound Characteristic Waveform Method	Hu Yuliang ; Qiao Junxuan ; Wang Haibin ; Wei Xiubo	In order to analyze heart valve disease accurately and effectively, a new quantized diagnosis method was proposed to analyze four clinical heart valve sounds, namely cardiac sound characteristic waveform.	Increase in pressure value.
---	---	---	---	-----------------------------

CHAPTER-3

AIM AND SCOPE

AIM

In today's era deaths due to heart disease has become a major issue approximately one person dies per minute due to heart disease. This is considering both male and female category and this ratio may vary according to the region also this ratio is considered for the people of age group. This does not indicate that the people with other age group will not be affected by heart diseases. This problem may start in early age group also and predict the cause and disease is a major challenge nowadays. Here in this paper, we have discussed various algorithms and tools used for prediction of heart diseases.

EXISTING SYSTEM

The before all existing system works on sets of both Deep learning and data mining. The existing system modules generates comprehensive report by implementing the strong prediction algorithm The main aims of the existing system to compare and check the before patient whose having disease outputs and new patient disease and determine future possibilities of the heart disease to a particular patient By Implementing the above mentioned model we will get the goal of developing a system with increased rate of accuracy of estimating the new patient getting heart attack percentage. The model which is proposed for Heart Attack Prediction System is invented for using Deep learning algorithms and approach. But by using all the existing systems the accuracy is very less.

PROPOSED SYSTEM

This proposed system have a data which classified if patients have heart disease or not according to features in it. This proposed system can try to use this data to create a model which tries predict(reading data and data Exploration) if a patient has this disease or not. In this proposed system, use logistic regression (classification) algorithm. Implements Naïve Bayes algorithm to getting accuracy result.Finally analyzing the results by the help of Comparing Models and Confusion Matrix. From the data we are having, it should be

classified into different structured data based on the features of the patient heart. From the availability of the data, we have to create a model which predicts the patient disease using logistic regression algorithm. First, we have to import the datasets. Read the datasets, the data should contain different variables like age, gender, sex, CP(chest pain), slope, target. The data should be explored so that the information is verified. Create a temporary variable and also build a model for logistic regression. Here, we use sigmoid function which helps in the graphical representation of the classified data. By using logistic regression, naïve Bayes the accuracy rate increases.

CHAPTER-4

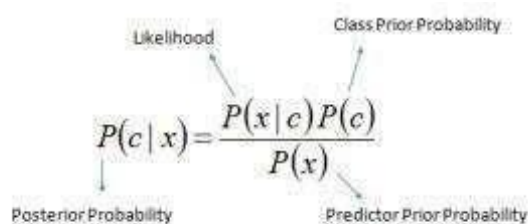
METHODOLOGY

NAIVE BAYES ALGORITHM

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as „Naive“.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. Look at the equation below:



The diagram shows the equation $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$ with arrows pointing from labels to the corresponding parts of the equation. 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c , *target*) given *predictor* (x , *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

NAÏVE BAYES ALGORITHM WORKING

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable „Play“ (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Bayes Rule:

The Bayes rule is a concept of determining $P(W | Z)$ from the provided training datasets which are $P(Z | W)$. To find it, we will replace the X and Y in the above equation respectively.

For test observation, W would be known and Z will be unknown. To calculate each row from the test data set you will calculate the probability of Z and consider W event has already happened.

Now, what will happen when Z will have more than two categories? We will calculate every class probability. The highest will be considered as a win.

Naive Bayes Algorithm:

In above the Bayes rule determines the probability of Z over given W . Now when it comes to the independent feature we will go for the Naive Bayes algorithm. The algorithm is called naive because we consider W 's are independent to one another.

In the case of multiple Z variables, we will assume that Z 's are independent.

Gaussian Naive Bayes

We have seen the computations where W is categorical but the question is how we can compute the probabilities where W is a continuous variable. Suppose if W follows a particular distribution then to compute the likelihood probability, you can plug in the density function of probability for that distribution.

And, if we consider W is following a Gaussian distribution then we can name it as Gaussian Naive Bayes by substituting the corresponding probability density of a Normal distribution. To compute the formula, you simply need mean and variance of W .

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct? We can solve it using above discussed method of posterior probability.

$$P(\text{Yes} \mid \text{Sunny}) = P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have $P(\text{Sunny} \mid \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

Now, $P(\text{Yes} \mid \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

PROS AND CONS OF NAÏVE BAYES

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict probate not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

4.1.3. APPLICATION OF NAÏVE BAYES ALGORITHM

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class

problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam

e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

- **Recommendation System:** Naive Bayes Classifier and Collaborative

Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

NEAREST NEIGHBOR(KNN) CLASSIFICATION METHOD

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) or the object property value (for k-NN regression) is known.

STEP 1: BEGIN

STEP 2: Input: $D = \{(x_1, c_1), \dots, (x_N, c_N)\}$

STEP 3: $x = (x_1, \dots, x_n)$ new instance to be classified

STEP 4: FOR each labelled instance (x_i, c_i) calculate $d(x_i, x)$

STEP 5: Order $d(x_i, x)$ from lowest to highest, $(i = 1 \dots$

$N)$ STEP 6: Select the K nearest instances to x: D_x^k

STEP 7: Assign to x the most frequent class in D_x^k

STEP 8: END

MODULES

- User Module
- Admin Module
- Disease Analysis Module
- Disease prediction Module

USER MODULE

In this Module, Patient can act as a user.

LOGIN

This is the first activity that opens the website. User needs to provide a correct contact number and a password, which user enters while registering, in order to login into the app. If information provided by the user matches with the data in the database table then user successfully login into the app else message of login failed is displayed and user need to reenter correct information. A link to the register activity is also provided for registration of new users.

REGISTRATION

A new user who wants to access the website needs to register first before login. By clicking on register button in login activity, the register activity gets open. A new user registers by entering full name, password and contact number. A user needs to enter password again in confirm password textbox for confirmation. When user enters the information in all textboxes, on the click of register button, the data is transferred to database and user is directed to login.

Activity again. Registered user then needs to login in order to access the app. Validations is applied on all the textboxes for proper functioning of the app. Like information in each textbox is must that is each textbox, either it is of name, contact, password or confirm password, will not be empty while registering. If any such textbox is empty app will give message of information is must in each textbox. Also data in password and confirm password fields must match for successful registration. Another validation is contact number must be valid one that is of 10 digits. If any such validation is violated then registration will be unsuccessful and then user needs to register again. message that app will display when one of the field is empty. If all such information is correct user will be directed to login activity for login into the app.

ADMIN MODULE

In this module, admin can add and view new doctor details, disease details and drug details. And then admin can view feedback provided by various users.

DISEASE ANALYSIS MODULE

In this module, we can analyse the disease and give patient feedbacks if it is life threatening or not and also calculate the how much probability disease will be happening.

DISEASE PREDICTION MODULE

Patient will specify the symptoms caused due to his illness. System will ask certain question regarding his illness and system predict the disease based on the symptoms specified by the patient and system will also suggest doctors based on the disease.

DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is a two-dimensional diagram that describes how data is processed and transmitted in a system. The graphical depiction recognizes each source of data and how it interacts with other data sources to reach a mutual output. In order to draft a data flow diagram one must

- Identify external inputs and outputs
- Determine how the inputs and outputs relate to each other
- Explain with graphics how these connections relate and what they result in.

Role of DFD

- It is a documentation support which is understood by both programmers and nonprogrammers. As DFD postulates only what processes are accomplished not how they are performed.

- A physical DFD postulates where the data flows and who processes the data.

- It permits analyst to isolate areas of interest in the organization and study them by examining the data that enter the process and viewing how they are altered when they leave.

CHAPTER-5

UML DIAGRAMS

UML DIAGRAMS

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

ADVANTAGES

- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

- The class model captures the static structure
- The state model expresses the dynamic behavior of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process

5.1.1.USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse[Type a quote from the document or the summary of an interesting point. You can position the text box anywhere in the document. Use the Drawing Tools tab to change the formatting of the pull quote textbox.]

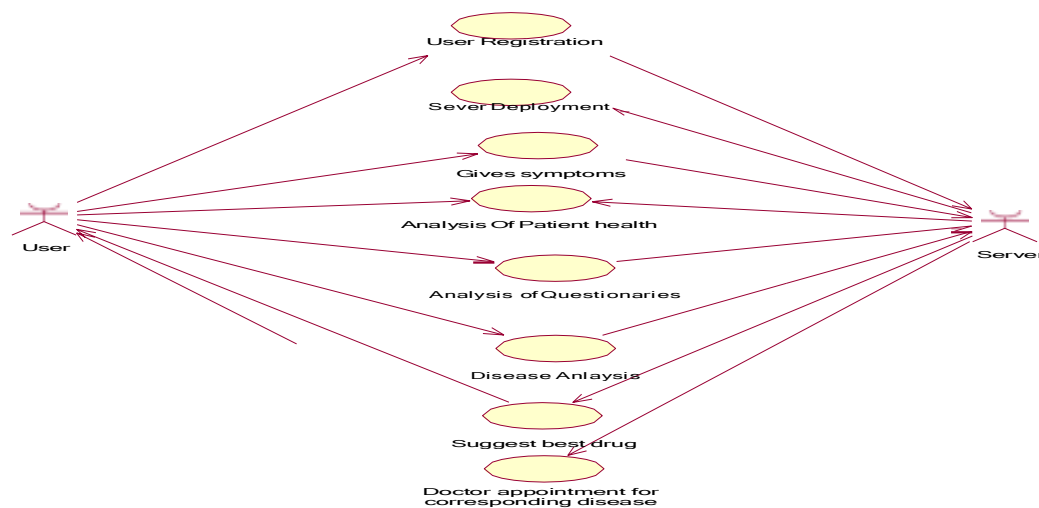


FIG: 5.1: Use case diagram

CLASS DIAGRAM

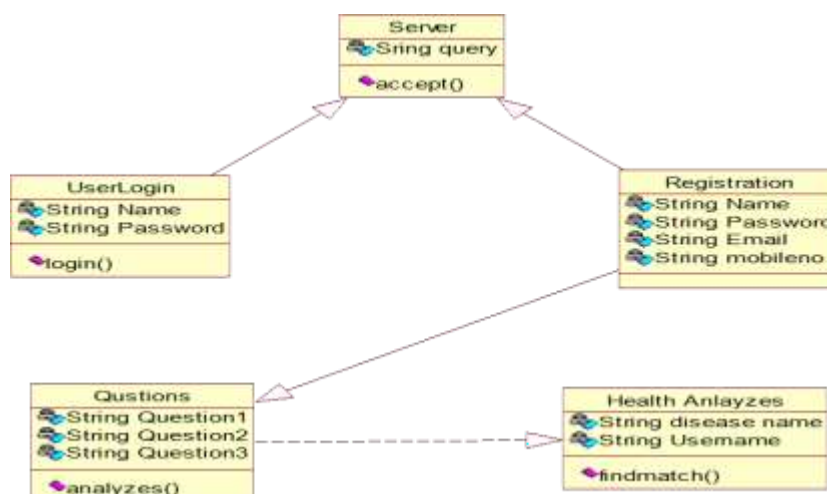


Fig5.2: Class Diagram

ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by- step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.

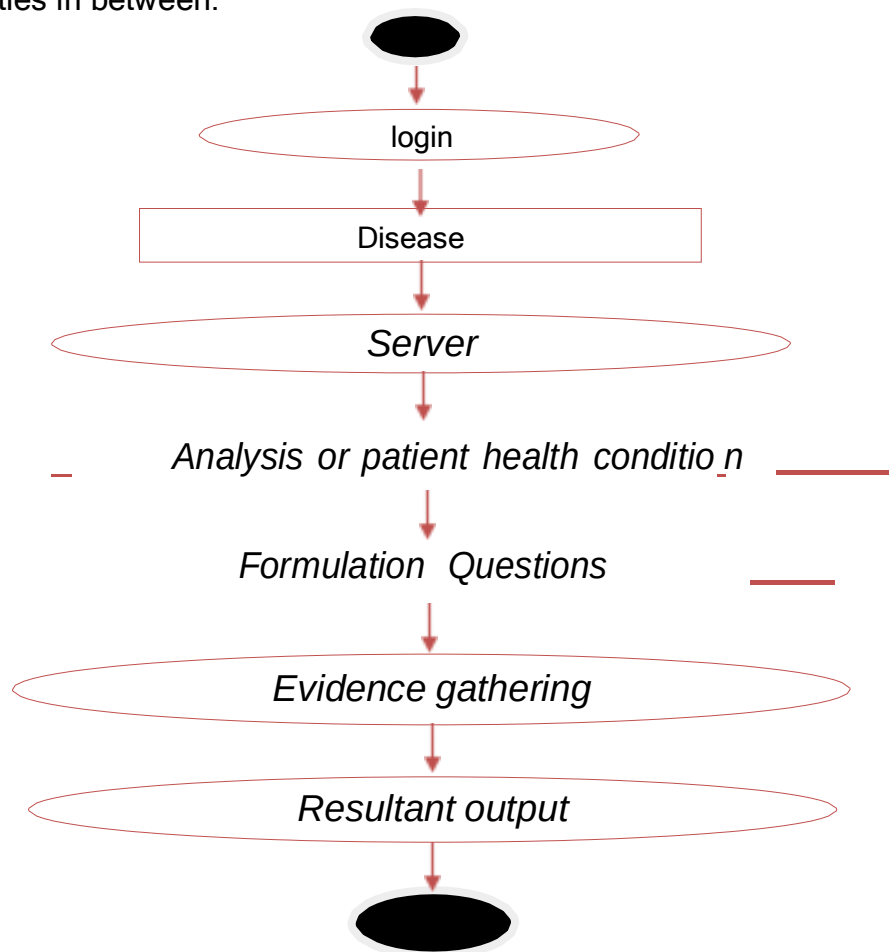


Fig5.4: Activity Diagram

SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.

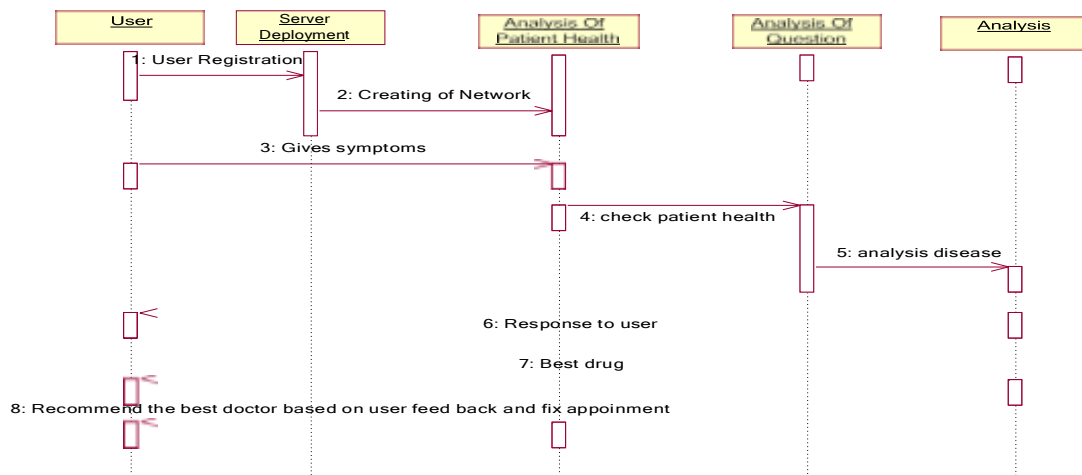


Fig5.4: Sequence Diagram

COLLABORATION DIAGRAM

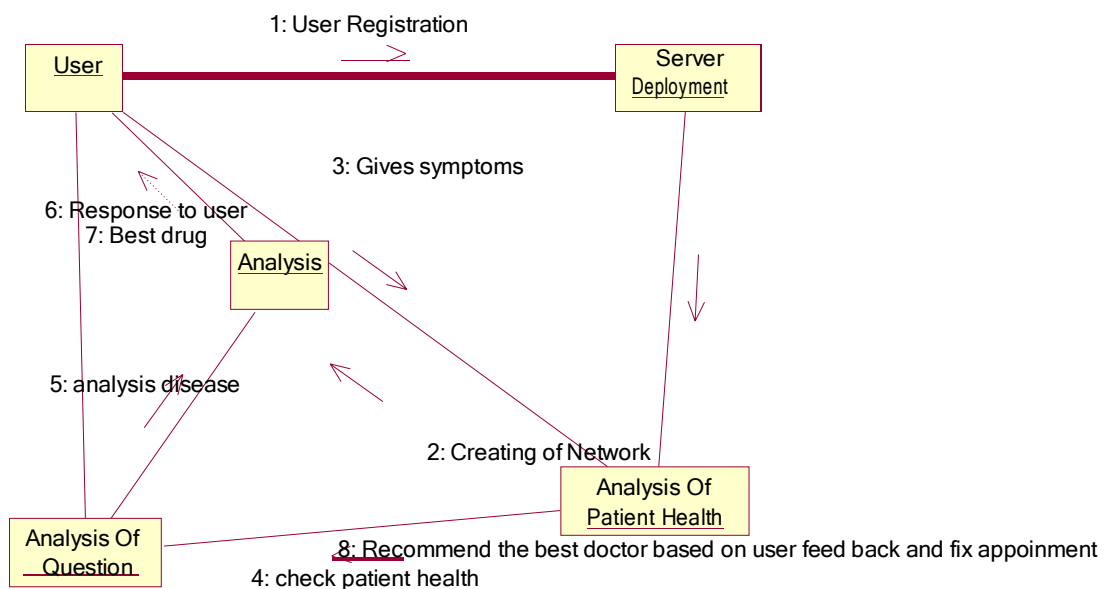


Fig: 5.5 Collaboration diagram

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like local host makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information

NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

5.5. SYSTEM DESIGN

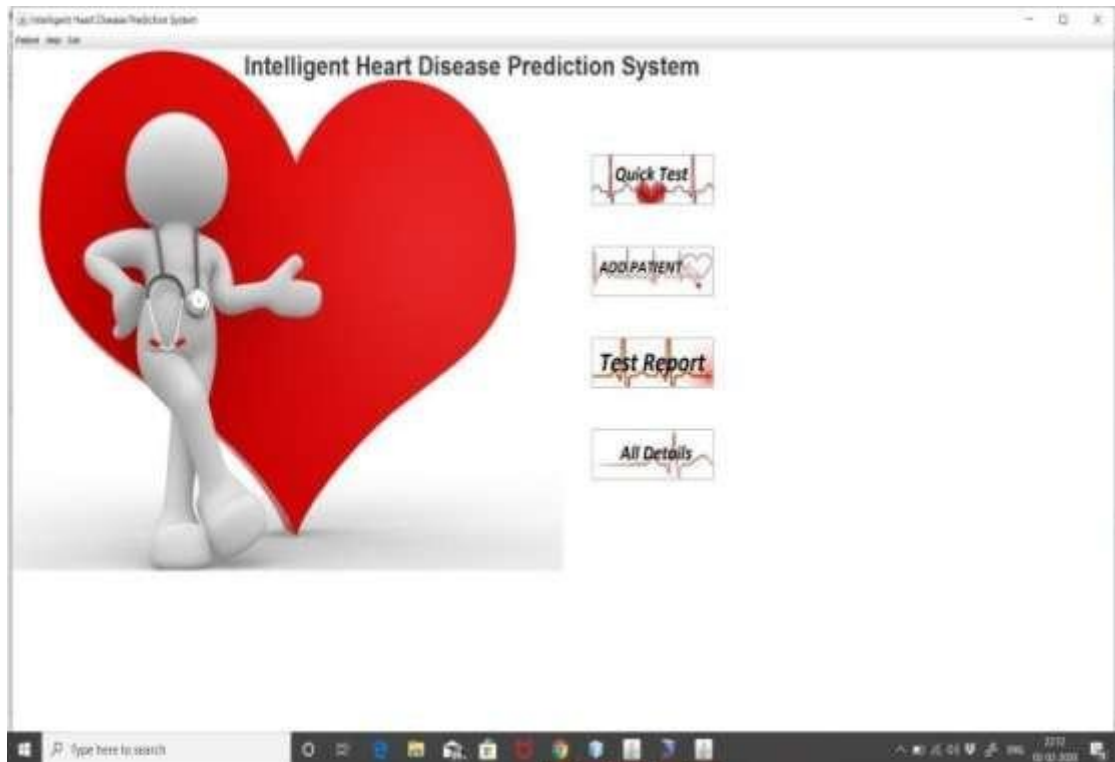


Fig5.6: Data entry page

Once user has logged into the application ,he is able to see the above page .It consists of four fields .They are :Quick test, add patient,test report,All details.

Quick test field is used to test the patient immediately.here we can give the attribute values using dataset which is present in IDE .we can use the previous tests as the reference values for the new tests.

Add patient field is used to add the attribute values of the new patient.

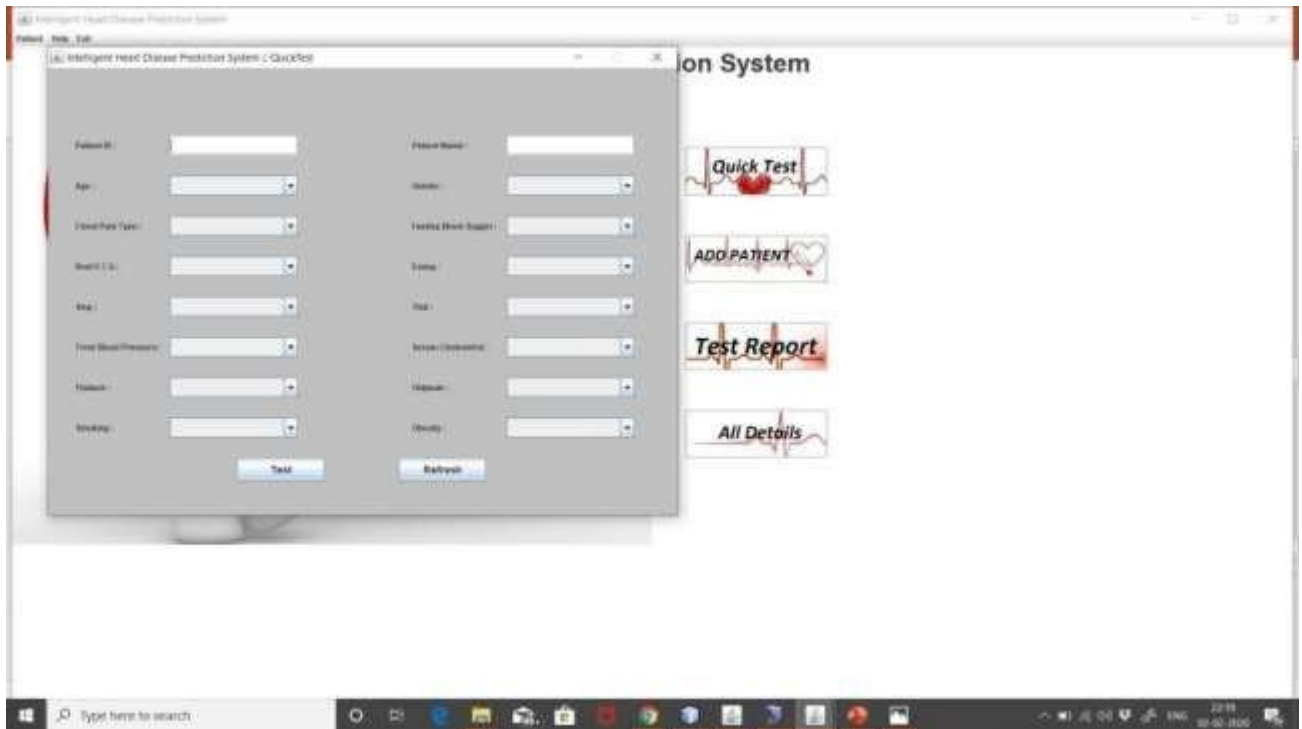


Fig5.7:Quick test module

After adding the details ,we have to click on the test report module. Then it will give positive or negativebased on the values which has given by the admin.



Fig5.8: Report result

CHAPTER-6

CONCLUSION AND FUTURE WORK

CONCLUSION

In this paper, we propose an approach for answering drug queries to support drug prescription. Our focus is on how to obtain and rank answers based on incomplete information and provide personalization. To cope with incomplete and noisy data, we allow both exact and close matches when answering queries. We also present an intuitive approach to display answers to users, which aims to help users to understand the ranked results and possibly refine their queries.

FUTURE WORK

In future work we will answer the drug queries to support drug prescription. Our focus is on how to obtain and rank answers based on incomplete information and provide personalization. And we will provide particular medicine for particular type of heart diseases.

REFERENCES

- [1] S. Khemmarat and L. Gao, “Supporting drug prescription via predictive and personalized query system,” in *PervasiveHealth*. IEEE, 2015.
- [2] C. Knox et al., “Drugbank 3.0: a comprehensive resource for omics research on drugs,” *Nucleic acids research*, vol. 39, no. suppl 1, pp. D1035–D1041, 2011.
- [3] M. Kuhn et al., “A side effect resource to capture phenotypic effects of drugs,” *Molecular systems biology*, vol. 6, no. 1, p. 343, 2010.
- [4] M. Kanehisa and S. Goto, “Kegg: kyoto encyclopedia of genes and genomes,” *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.
- [5] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [6] K. Sangkuhl et al., “Pharmgkb: understanding the effects of individual genetic variants,” *Drug Metab. Rev.*, vol. 40, no. 4, pp. 539–551, 2008.
- [7] A. Langer et al., “A text based drug query system for mobile phones,” *Int. J. Mob. Commun.*, vol. 12, no. 4, pp. 411–429, Jul. 2014.
- [8] C. Doulaverakis et al., “Panacea, a semantic-enabled drug recommendations discovery framework,” *J. Biomed. Semant.*, vol. 5, p. 13, 2014.

A.PROGRAM

```
package com.toushin.IHDPS;
import com.mysql.jdbc.Connection;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
package com.toushin.IHDPS;
import com.mysql.jdbc.Connection;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DecimalFormat;
import java.util.*;
import javax.management.Notification;
import javax.swing.*.*;

public class Add_Patient extends JFrame implements ActionListener{

Add_Patient(){
    super("Intelligent Heart Disease Prediction System ::: Add_Patient");
    panel.setLayout(null);
    setVisible(true);
    setSize(940,700);
    panel.setBackground(Color.LIGHT_GRAY);
    setLocation(60, 60);
    setResizable(false);
    add(panel);
    conn=DB.getConnection();
    AutoLoad();
    ComponentAdd();
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

private void inserdata(){
    int status=0;
    try {
```

```

        st = conn.createStatement();
        PreparedStatement ps=conn.prepareStatement("insert into
TbTrainingDataSet(PatientId,patientName,Age,Gender,ChestPainType,FastingBloodsugger,Rest
ecg,Exang,Slop,Thal,TrestBloodPressure,SerumCholesterol,Thalach,Oldpeak,Smoking,Obesity,
Disease,Probabilty,BestClassName)values('"+lbltxtPatientId.getText()+"','"+txtPatientName.getTe
xt()+"','"+cmbage.getSelectedItemAt().toString()+"','"+cmbgender.getSelectedItemAt().toString()+"','"+
cmbChestPaintype.getSelectedItemAt().toString()+"','"+cmbFastingBloodsugger.getSelectedItemAt().
toString()+"','"+cmbRestecg.getSelectedItemAt().toString()+"','"+cmbExang.getSelectedItemAt().toStri
ng()+"','"+cmbSlop.getSelectedItemAt().toString()+"','"+cmbThal.getSelectedItemAt().toString()+"','"+c
mbTrestBloodPressure.getSelectedItemAt().toString()+"','"+cmbSerumCholesterol.getSelectedItemAt

```

```

().toString()+""+cmbThalach.getSelectedItemAt().toString()+""+cmbOldpeak.getSelectedItemAt().t
oString()+""+cmbSmoking.getSelectedItemAt().toString()+""+cmbObesity.getSelectedItemAt().toSt
ring()+""+txtDisease.getText()+""+txtProbability.getText()+""+""));
        status=ps.executeUpdate();
        if (status > 0) {
            JOptionPane.showMessageDialog(null, "Welcome!!! your records have successfully
inserted!!!");
        }

    } catch (Exception exception) {
        System.out.println(exception.toString());
    }
}

```

```

if(DiseasePositiveCalculation>DiseaseNegativeCalculation)
{
    result="Positive";
        txtDisease.setText(result);
//JOptionPane.showMessageDialog(null,"Heart Disease Result is Positive");
}
else
{
    result="Negative";
        txtDisease.setText(result);
//JOptionPane.showMessageDialog(null,"Heart Disease Result is Negative");
}
}

private void CARTValueCalculation()
{
    if(result == "Positive"){
        try {
            st = conn.createStatement();
            ResultSet iterTage = st.executeQuery("select count(Age)as age from
tbTrainingDataSet where Age=" + cmbage.getSelectedItemAt().toString() + " ");
            while (iterTage.next()) {
                //Object element = iterage.next();
                Cagecount = Double.parseDouble(iterTage.getString(1));
                System.out.println("agecount" + Cagecount);
            }
        } catch (Exception exception) {
        }

        try {
            st = conn.createStatement();
            ResultSet iteragepositive = st.executeQuery("select count(Age)as age from
tbTrainingDataSet where Age=" + cmbage.getSelectedItemAt().toString() + "and

```

```

Disease='Positive' ");
    while (iteragepositive.next()) {
        //Object element = iteragepositive.next();
        CPositiveCount = Double.parseDouble(iteragepositive.getString(1));
        System.out.println("PositiveCount " + CPositiveCount);
    }
} catch (Exception exception) {
}
CAge=CPositiveCount/Cagecount;
    CAge=Double.parseDouble(df.format(CAge));
System.out.println("Age "+CAge);
////////////////////

```

```

try {
    st = conn.createStatement();
    ResultSet itergender = st.executeQuery("select count(Gender)as gender from
tbTrainingDataSet where Gender='" + cmbgender.getSelectedItem().toString() + "'");
    while (itergender.next()) {
        //Object element = itergender.next();
        Cgendercount = Double.parseDouble(itergender.getString(1));
        System.out.println("gendercount" + Cgendercount);
    }
} catch (Exception exception) {
}

try {
    st = conn.createStatement();
    ResultSet itergenderpositive = st.executeQuery("select count(Gender)as Gender from
tbTrainingDataSet where Gender='" + cmbgender.getSelectedItem().toString() + "'and
Disease='Positive' ");
    while (itergenderpositive.next()) {
        //Object element = itergenderpositive.next();
        CgenderPositiveCount = Double.parseDouble(itergenderpositive.getString(1));
        System.out.println("genderPositiveCount " + CgenderPositiveCount);
    }
} catch (Exception exception) {
}

CGender=CgenderPositiveCount/Cgendercount;
CGender=Double.parseDouble(df.format(CGender));
System.out.println("gender "+CGender);

//////////
try {
    st = conn.createStatement();
    ResultSet iterchest = st.executeQuery("select count(ChestPainType)as ChestPainType
from tbTrainingDataSet where ChestPainType='" +
cmbChestPaintype.getSelectedItem().toString() + "'");
    while (iterchest.next()) {
        //Object element = iterchest.next();
        Cchestcount = Double.parseDouble(iterchest.getString(1));
        System.out.println("chestcount" + Cchestcount);
    }
} catch (Exception exception) {
}

try {
    st = conn.createStatement();
    ResultSet iterchestpositive = st.executeQuery("select count(ChestPainType)as
ChestPainType from tbTrainingDataSet where ChestPainType='" +
cmbChestPaintype.getSelectedItem().toString() + "'and Disease='Positive' ");
    while (iterchestpositive.next()) {
        //Object element = iterchestpositive.next();

```

```

        CchestPositiveCount = Double.parseDouble(iterchestpositive.getString(1));
        System.out.println("chestPositiveCount " + CchestPositiveCount);
    }
} catch (Exception exception) {
}
}
Cchest=CchestPositiveCount/Cchestcount;
Cchest=Double.parseDouble(df.format(Cchest));
System.out.println("Chest "+Cchest);
////////////////////

try {
    st = conn.createStatement();
    ResultSet iterFasting = st.executeQuery("select count(FastingBloodsugar)as
FastingBloodsugar from tbTrainingDataSet where FastingBloodsugar='" +
cmbFastingBloodsugar.getSelectedItemAt().toString() + "'");
    while (iterFasting.next()) {
        //Object element = iterFasting.next();
        CFastingcount = Double.parseDouble(iterFasting.getString(1));
        System.out.println("Fastingcount" + CFastingcount);
    }
} catch (Exception exception) {
}

try {
    st = conn.createStatement();
    ResultSet iterFastingpositive = st.executeQuery("select count(FastingBloodsugar)as
FastingBloodsugar from tbTrainingDataSet where FastingBloodsugar='" +
cmbFastingBloodsugar.getSelectedItemAt().toString() + "'and Disease='Positive' ");
    while (iterFastingpositive.next()) {
        //Object element = iterFastingpositive.next();
        CFastingPositiveCount = Double.parseDouble(iterFastingpositive.getString(1));
        System.out.println("FastingPositiveCount " + CFastingPositiveCount);
    }
} catch (Exception exception) {
}
CFasting=CFastingPositiveCount/CFastingcount;
CFasting=Double.parseDouble(df.format(CFasting));
System.out.println("Fasting "+CFasting);
////////////////////

try {
    st = conn.createStatement();
    ResultSet iterRest = st.executeQuery("select count(Restecg)as Restecg from
tbTrainingDataSet where Restecg='" + cmbRestecg.getSelectedItemAt().toString() + "'");
    while (iterRest.next()) {
        //Object element = iterRest.next();
        CRestcount = Double.parseDouble(iterRest.getString(1));
        System.out.println("Restcount" + CRestcount);
    }
} catch (Exception exception) {
}

```

```

    }

    try {
        st = conn.createStatement();
        ResultSet iterRestpositive = st.executeQuery("select count(Restecg)as Restecg from
tbTrainingDataSet where Restecg='" + cmbRestecg.getSelectedItem().toString() + "'and
Disease='Positive' ");
        while (iterRestpositive.next()) {
            //Object element = iterRestpositive.next();
            CRestPositiveCount = Double.parseDouble(iterRestpositive.getString(1));
            System.out.println("RestPositiveCount " + CRestPositiveCount);
        }
    } catch (Exception exception) {
    }
    CRest=CRestPositiveCount/CRestcount;
    CRest=Double.parseDouble(df.format(CRest));
    System.out.println("Rest "+CRest);
    //////////////////////////////////////

    try {
        st = conn.createStatement();
        ResultSet iterExang = st.executeQuery("select count(Exang)as Exang from
tbTrainingDataSet where Exang='" + cmbExang.getSelectedItem().toString() + "' ");
        while (iterExang.next()) {
            //Object element = iterExang.next();
            CExangcount = Double.parseDouble(iterExang.getString(1));
            System.out.println("Exangcount" + CExangcount);
        }
    } catch (Exception exception) {
    }
    try {
        st = conn.createStatement();
        ResultSet iterExangpositive = st.executeQuery("select count(Exang)as Exang from
tbTrainingDataSet where Exang='" + cmbExang.getSelectedItem().toString() + "'and
Disease='Positive' ");
        while (iterExangpositive.next()) {
            //Object element = iterExangpositive.next();
            CExangPositiveCount = Double.parseDouble(iterExangpositive.getString(1));
            System.out.println("ExangPositiveCount " + CExangPositiveCount);
        }
    } catch (Exception exception) {
    }
    Cexang=CExangPositiveCount/CExangcount;
    System.out.println("Exang "+Cexang);
    //////////////////////////////////////

    try {
        st = conn.createStatement();

```

```

        ResultSet iterSlop = st.executeQuery("select count(Slop)as Slop from
tbTrainingDataSet where Slop='" + cmbSlop.getSelectedItem().toString() + "'");
        while (iterSlop.next()) {
            //Object element = iterSlop.next();
            CSlopcount = Double.parseDouble(iterSlop.getString(1));
            System.out.println("Slopcount" + CSlopcount);
        }
    } catch (Exception exception) {
    }

    try {
        st = conn.createStatement();
        ResultSet iterSloppositive = st.executeQuery("select count(Slop)as Slop from
tbTrainingDataSet where Slop='" + cmbSlop.getSelectedItem().toString() + "'and
Disease='Positive' ");
        while (iterSloppositive.next()) {
            //Object element = iterSloppositive.next();
            CSlopPositiveCount = Double.parseDouble(iterSloppositive.getString(1));
            System.out.println("SlopPositiveCount " + CSlopPositiveCount);
        }
    } catch (Exception exception) {
    }

    Cslop=CSlopPositiveCount/CSlopcount;
    Cslop=Double.parseDouble(df.format(Cslop));
    System.out.println("Slop "+Cslop);
    //////////////////////////////////////

    try {
        st = conn.createStatement();
        ResultSet iterthal = st.executeQuery("select count(thal)as thal from tbTrainingDataSet
where thal='" + cmbThal.getSelectedItem().toString() + "'");
        while (iterthal.next()) {
            //Object element = iterthal.next();
            Cthalcount = Double.parseDouble(iterthal.getString(1));
            System.out.println("thalcount" + Cthalcount);
        }
    } catch (Exception exception) {
    }

    try {
        st = conn.createStatement();
        ResultSet iterthalpositive = st.executeQuery("select count(thal)as thal from
tbTrainingDataSet where thal='" + cmbThal.getSelectedItem().toString() + "'and
Disease='Positive' ");
        while (iterthalpositive.next()) {
            //Object element = iterthalpositive.next();
            CthalPositiveCount = Double.parseDouble(iterthalpositive.getString(1));
            System.out.println("thalPositiveCount " + CthalPositiveCount);
        }
    } catch (Exception exception) {
    }

```

```

    }
    Cthal=CthalPositiveCount/Cthalcount;
    Cthal=Double.parseDouble(df.format(Cthal));
    System.out.println("thal "+Cthal);
    //////////////////////////////////////

    try {
        st = conn.createStatement();
        ResultSet itertrest = st.executeQuery("select count(TrestBloodPressure)as
TrestBloodPressure from tbTrainingDataSet where TrestBloodPressure='" +
cmbTrestBloodPressure.getSelectedItem().toString() + "'");
        while (itertrest.next()) {
            //Object element = itertrest.next();
            Ctrestcount = Double.parseDouble(itertrest.getString(1));
            System.out.println("trestcount" + Ctrestcount);
        }
    } catch (Exception exception) {
    }

    try {
        st = conn.createStatement();
        ResultSet itertrestpositive = st.executeQuery("select count(TrestBloodPressure)as
TrestBloodPressure from tbTrainingDataSet where TrestBloodPressure='" +
cmbTrestBloodPressure.getSelectedItem().toString() + "'and Disease='Positive' ");
        while (itertrestpositive.next()) {
            //Object element = itertrestpositive.next();
            CtrestPositiveCount = Double.parseDouble(itertrestpositive.getString(1));
            System.out.println("trestPositiveCount " + CtrestPositiveCount);
        }
    } catch (Exception exception) {
    }

    Ctrest=CtrestPositiveCount/Ctrestcount;
    Ctrest=Double.parseDouble(df.format(Ctrest));
    System.out.println("trest "+Ctrest);
    //////////////////////////////////////

    try {
        st = conn.createStatement();
        ResultSet iterserum = st.executeQuery("select count(SerumCholesterol)as
SerumCholesterol from tbTrainingDataSet where SerumCholesterol='" +
cmbSerumCholesterol.getSelectedItem().toString() + "'");
        while (iterserum.next()) {
            //Object element = iterserum.next();
            Cserumcount = Double.parseDouble(iterserum.getString(1));
            System.out.println("serumcount" + Cserumcount);
        }
    } catch (Exception exception) {
    }

    try {

```

```

        st = conn.createStatement();
        ResultSet iterserumpositive = st.executeQuery("select count(SerumCholesterol)as
SerumCholesterol from tbTrainingDataSet where SerumCholesterol='" +
cmbSerumCholesterol.getSelectedItem().toString() + "'and Disease='Positive' ");
        while (iterserumpositive.next()) {
            //Object element = iterserumpositive.next();
            CserumPositiveCount = Double.parseDouble(iterserumpositive.getString(1));
            System.out.println("serumPositiveCount " + CserumPositiveCount);
        }
    } catch (Exception exception) {
    }
    Cserum=CserumPositiveCount/Cserumcount;
    Cserum=Double.parseDouble(df.format(Cserum));
    System.out.println("serum "+Cserum);

    try {
        st = conn.createStatement();
        ResultSet iterthalach = st.executeQuery("select count(Thalach)as Thalach from
tbTrainingDataSet where Thalach='" + cmbThalach.getSelectedItem().toString() + "'");
        while (iterthalach.next()) {
            //Object element = iterthalach.next();
            Cthalachcount = Double.parseDouble(iterthalach.getString(1));
            System.out.println("thalachcount" + Cthalachcount);
        }
    } catch (Exception exception) {
    }

    try {
        st = conn.createStatement();
        ResultSet iterthalachpositive = st.executeQuery("select count(Thalach)as Thalach from
tbTrainingDataSet where Thalach='" + cmbThalach.getSelectedItem().toString() + "'and
Disease='Positive' ");
        while (iterthalachpositive.next()) {
            //Object element = iterthalachpositive.next();
            CthalachPositiveCount = Double.parseDouble(iterthalachpositive.getString(1));
            System.out.println("thalachPositiveCount " + CthalachPositiveCount);
        }
    } catch (Exception exception) {
    }
    Cthalach=CthalachPositiveCount/Cthalachcount;
    Cthalach=Double.parseDouble(df.format(Cthalach));
    System.out.println("thalach "+Cthalach);
    //////////////////////////////////////

    try {
        st = conn.createStatement();
        ResultSet iteroldpeak = st.executeQuery("select count(Oldpeak)as Oldpeak from
tbTrainingDataSet where Oldpeak='" + cmbOldpeak.getSelectedItem().toString() + "'");
        while (iteroldpeak.next()) {

```

```

        //Object element = iteroldpeak.next();
        Coldpeakcount = Double.parseDouble(iteroldpeak.getString(1));
        System.out.println("oldpeakcount" + Coldpeakcount);
    }
} catch (Exception exception) {
}

try {
    st = conn.createStatement();
    ResultSet iteroldpeakpositive = st.executeQuery("select count(Oldpeak)as Oldpeak
from tbTrainingDataSet where Oldpeak='" + cmbOldpeak.getSelectedItem().toString() + "'and
Disease='Positive' ");
    while (iteroldpeakpositive.next()) {
        //Object element = iteroldpeakpositive.next();
        ColdpeakPositiveCount = Double.parseDouble(iteroldpeakpositive.getString(1));
        System.out.println("oldpeakPositiveCount " + ColdpeakPositiveCount);
    }
} catch (Exception exception) {
}

Coldpeak=ColdpeakPositiveCount/Coldpeakcount;
Coldpeak=Double.parseDouble(df.format(Coldpeak));
System.out.println("oldpeak "+Coldpeak);
////////////////////

try {
    st = conn.createStatement();
    ResultSet itersmoking = st.executeQuery("select count(Smoking)as Smoking from
tbTrainingDataSet where Smoking='" + cmbSmoking.getSelectedItem().toString() + "'");
    while (itersmoking.next()) {
        //Object element = itersmoking.next();
        Csmokingcount = Double.parseDouble(itersmoking.getString(1));
        System.out.println("smokingcount" + Csmokingcount);
    }
} catch (Exception exception) {
}

try {
    st = conn.createStatement();
    ResultSet itersmokingpositive = st.executeQuery("select count(Smoking)as Smoking
from tbTrainingDataSet where Smoking='" + cmbSmoking.getSelectedItem().toString() + "'and
Disease='Positive' ");
    while (itersmokingpositive.next()) {
        //Object element = itersmokingpositive.next();
        CsmokingPositiveCount = Double.parseDouble(itersmokingpositive.getString(1));
        System.out.println("smokingPositiveCount " + CsmokingPositiveCount);
    }
} catch (Exception exception) {
}

Csmoking=CsmokingPositiveCount/Csmokingcount;
Csmoking=Double.parseDouble(df.format(Csmoking));

```

```

System.out.println("smoking "+Csmoking);
////////////////////////////////////
    try {
        st = conn.createStatement();
        ResultSet iterobesity = st.executeQuery("select count(Obesity)as Obesity from
tbTrainingDataSet where Obesity='" + cmbObesity.getSelectedItem().toString() + "'");
        while (iterobesity.next()) {
            //Object element = iterobesity.next();
            Cobesitycount = Double.parseDouble(iterobesity.getString(1));
            System.out.println("obesitycount" + Cobesitycount);
        }
    } catch (Exception exception) {
    }

    try {
        st = conn.createStatement();
        ResultSet iterobesitypositive = st.executeQuery("select count(Obesity)as Obesity from

```

```

tbTrainingDataSet where Obesity="" + cmbObesity.getSelectedItem().toString() + ""and
Disease='Positive' ");
    while (iterobesitypositive.next()) {
        //Object element = iterobesitypositive.next();
        CobesityPositiveCount = Double.parseDouble(iterobesitypositive.getString(1));
        System.out.println("obesityPositiveCount " + CobesityPositiveCount);
    }
} catch (Exception exception) {
}
Cobesity=CobesityPositiveCount/Cobesitycount;
Cobesity=Double.parseDouble(df.format(Cobesity));
System.out.println("obesity "+Cobesity);

System.out.println("=====
=====");

try {
    st = conn.createStatement();
    ResultSet iteragepositive = st.executeQuery("select count(Age)as age from
tbTrainingDataSet where Age="" + cmbage.getSelectedItem().toString() + ""and
Disease='Negative' ");
    while (iteragepositive.next()) {
        //Object element = iteragepositive.next();
        CPositiveCount = Double.parseDouble(iteragepositive.getString(1));
        System.out.println("PositiveCount " + CPositiveCount);
    }
} catch (Exception exception) {
}

```

Submission date: 11-Mar-2020 11:22PM (UTC-0500)

Submission ID: 1274113121

File name: Mani_Chand.pdf (438.73K)

Word count: 2219

Character count: 12039

that infection. A privilege and deliberate gadget for seeing high-peril patients and burrowing data for advantageous assessment of coronary illness looks a real need. Distinctive individual body can show various side effects of coronary illness which may differ likewise. However, they oftentimes incorporate back torment, jaw torment, neck torment, stomach issue, and littleness of breath, chest agony, arms and shoulders torments. There are a wide range of heart ailments which incorporates cardiovascular breakdown and stroke and coronary conduit infection. Heart expert's create a good and huge record of patient's database and store them. It likewise conveys an extraordinary possibility for mining an esteemed information from such kind of datasets. There is tremendous research proceeding to decide coronary illness chance factors in various patients, various scientists are utilizing different factual methodologies and various projects of information mining draws near. Measurable examination has recognized the check for hazard factors of heart maladies tallying smoking, age, circulatory strain, diabetes, all out cholesterol, and hypertension, coronary illness preparing in familys, heftiness and absence for activity. For anticipation and social insurance of patients who are going to have dependent of coronary illness it is imperative to have familiarity with heart ailments.

2 Related Works:

⁸ Escudero J, Ifeakor E, Zajicek JP, Green C, Shear J, Pearson S. The methodology performed also to considering all information without a moment's delay, while fundamentally lessening the quantity of the biomarkers expected to accomplish a sure analysis for every patient. In this way, it might add to a customized and viable location of AD, and may demonstrate helpful in clinical settings.

⁷ Shan Zheng, Minzhen Wang, Shigong Wang, Kezhe Shang, Lili Hu, Jinrong Dong. The basic target of this paper is to supply reasonable essentials for blocking and guaging the normality of hand, foot and mouth disease to look into the impact of various meteorological conditions every so often of hand, foot and mouth issue in Wuwei City, northwestern China. Here the information about the ailments and air was amassed from 2008-2010, and the relationship

assessment, differing direct descend into sin and exponential bend fitting strategies were made.

⁶ Zhang Y, Long JD, Mills JA, Warner JH, Lu W, Paulsen JS. Spectral data have been widely used to estimate the disease severity levels of different plants. Be that as it may, such information have not been assessed to gauge the infection phases of the plant. This examination planned for building up an otherworldly infection record that can recognize the phases of wheat leaf rust illness at different DS levels.

Hu Yuliang, Qiao Junxuan, Wang Haibin, Wei Xiubo. In request to break down heart valve ailment precisely and adequately, another quantized finding technique was proposed to examine four clinical heart valve sounds, to be specific cardiovascular sound trademark waveform.

⁵ Chengyu Liu, David Springer, Qiao Li, Benjamin Moody, Ricardo Abad Juan, Francisco J Chorro. The article underscores clinical and prognostic criticalness of non-straight extents of the beat variability, applied on the social occasion of patients with coronary sickness and age-facilitated strong benchmark gathering. Three specific methods were applied: Hurst model, Detrended Fluctuation Analysis and assembled entropy. Hurst sort of the R-R game-plan was obliged by the range rescaled appraisal system. DFA was utilized to evaluate fractal long-go affiliation properties of heartbeat changeability.

3 Proposed System:

This proposed system have a data which orchestrated if patients have coronary sickness or not as showed by features in it. This proposed structure can endeavor to use this data to make a model which endeavors foresee (getting data and data Exploration) if a patient has this disease or not. At this moment, use determined backslide count. Right now, utilize calculated relapse (arrangement) calculation. Executes Naive Bayes calculation to getting precision result. Finally researching the results by the help of Comparing Models and Confusion Matrix. From the data we are having, it should be portrayed into different sorted out data reliant on the features of the patient heart. From the accessibility of the information, we need to make a model which predicts the patient illness utilizing calculated relapse calculation. To begin with, we need

Heart Disease Prediction Using Machine Learning

M.Sai Shekhar . Y.Mani Chand . L.Mary Gladence

Abstract:

In the present period passing because of coronary illness has become a significant issue roughly one individual bites the dust every moment because of coronary illness. This is thinking about both the genders classification and this part shift as indicated by area likewise this part is taken into consideration for every person old enough to gathering. This doesn't show every individuals those who are having different ages gathering won't get influenced by coronary illness. The above issue might begin starting of the age bunch additionally, foresee the motive and infection is a great take a look at those days. Here right now, was examined distinctive calculations and gadgets applied for forecast of heart maladies. The datasets are prepared in java coding language by using machine learning algorithms like Naïve Bayes, KNN algorithms which are used to predict the disease easily and shows the probability of disease of the patient is either negative or positive.

Keywords: Prediction, Classification, Naïve Bayes, KNN, Coronary illness, Probability, Positive, Negative, Machine Learning

1 Introduction:

The main theme of this paper around distinctive data mining rehearses those are important in coronary illness figure by using unique data mining instruments those are available. In this event that the heart doesn't work appropriately, this will trouble different pieces of the human body, for example, cerebrum, kidney and so on. Coronary illness is a sort of sickness which impacts the working of the heart. In modern era coronary illness is the primary reason for end of. WHO-World Health Organization has foreseen that 13 million individuals kick the bucket each year in view of heart sicknesses. Some heart contaminations are cardiovascular, respiratory disappointment, coronary and pound. Pound is a sort of coronary sickness that occurs because of strengthening, blocking or lessening of veins which cross the thoughts or it can in like manner

be begun by hypertension. The significant test that the Healthcare business faces now-a-days is predominance of office. Diagnosing the sickness precisely and giving convincing treatment to patients will portray the idea of organization. Poor examination causes awful results that are not recognized. Records or information of clinical history is enormous, however these are from numerous disparate establishments. The translations that are finished by doctors are fundamental segments of these information. The data in real world might be noisy, deficient and clashing, so data preprocessing will be required in command to fill the disposed of characteristics in the database. Regardless of whether cardiovascular sicknesses is found as the significant wellspring of death in world in antiquated years, these have been reported as the most avoidable and sensible illnesses. The entire and exact administration of a sickness lay on the very much planned judgment of

to import the datasets. Peruse the datasets, the information ought to contain various factors like age, sexual orientation, sex, cp (chest torment), incline, target. The information ought to be investigated with the goal that the data is checked. Make a brief variable and furthermore assemble a model for calculated relapse. Here, we utilize sigmoid capacity which helps in the graphical portrayal of the grouped information. By utilizing strategic relapse, credulous Bayes the precision rate increments.

9 4 Modules:

1. User Module

2. Admin Module

3. Disease Analysis

4. Disease prediction

4 4.1 User Module:

In this Module, Patient can act as an user.

4.1.1 Login:

This is the fundamental activity that opens the site. Client needs to give a right contact numbers and a secret key, which client enters while enlisting, to logged in the application. On the off chance that data furnished for the client equals the information in the database table then client sufficiently logged into the application else message of logged in fizzled are shown up and client should return right data. An interface with the register improvement is additionally given to selection of new clients.

4.1.2 Registration:

Another client who needs to get to the site needs to enlist first before login. By tapping on register button in login development, the register activity gets open. Another customer enrolls by entering total name, mystery expression and contact number. A customer needs to enter mystery state again in insist mystery word textbox for assertion. At the point when client entered the data in every textboxes, on the snap of register button, the information is moved to database

and clients are coordinated to be logged in action once more. Enlisted client then needs to login so as to get to the application. Approvals are applied on every textboxes for appropriate working for the application. Like information in each textbox is must that is each textbox, it is possible that it is of name, contact, secret phrase or affirm secret key, won't be unfilled while enrolling. On the off chance that any of those textboxes are vacant application will get the message of data is should in every textbox. Additionally information in secret word and affirm secret word fields must counterpart for fruitful enlistment. Another endorsement is contact number must be real one that is of 10 digits. In the event that any such approval is disregarded, at that point enlistment will be fruitless and afterward client needs to enroll once more. message that application will show when one of the field is vacant. On the off chance that all such data is right client will be coordinated to login movement for login into the application.

4 4.2 Admin:

In this module, admin can add and view new doctor tails, disease details and drug details. And then admin can view feedback provided by various users.

4.3 Disease Analysis:

In this module, we can analyse the disease and give patient feedbacks if it is life threatening or not and also calculate the how much probability disease will be happening.

3 4.4 Disease Prediction:

Patient will show the reactions caused in view of his illness. System will represent certain request with respect to his sickness and structure envision the affliction subject to the reactions showed by the patient and structure will moreover suggest pros reliant on the contamination.

5 Naïve Bayes Algorithm:

It is a depiction method subject to Bayes' Theorem with an uncertainty of self-rule among markers. In essential notations, a Naive Bayes methodology specifies that the region of a

particular part in a module is unimportant to the closeness of various different segments. For example, a trademark thing might be seen as an apple in case it is red, circle and all together 3 slithers in partition over. Whether or not this features depends each other are based on the closeness of various features, these features selfrulingly add to the probability where typical thing is an apple and that is clarification is called as 'Fair'. Genuine Bayes structure is certainly not hard to create and particularly strong for colossal informative varieties. Near to ease, Naive Bayes is known to beat the altogether refined solicitation frameworks.

Utilizations of Naive Bayes Algorithms:

Ongoing Prediction: Naive Bayes is a nervous studying technique and it is certain smart. Thusly, it can be used to create wants perpetually.

Multi class Prediction: This include is in like way conspicuous for various module gauge feature. Therefore we can envision the probability of various division of spotted variable.

Content gathering/Spam filtration/Sentimental Analysis: Innocent Bayes methodology generally used in content approach (considering good predicted result in various modules situations and independence sequences) have greater accomplishment percentage when showed up diversely according to different figurings. Properly, it is widely utilized in Spam isolation (see mailbox) and Sentimental prediction (through online life assessment, to perceive good and bad client estimation)

6 K-Nearest Neighbour Classification Method:

1 K-NN is a sort of event based learning, or torpid acknowledging, where the limit is simply approximated locally and all computation is surrendered until game plan. The k-NN count is

1 Among minimal complex of all AI computations. The neighbors are taken from a ton of articles for which the class (for k-NN plan) or the thing property estimation (for k-NN backslide) is known.

6 Output:



6.1 Details of the patients



6.2 Effectiveness of heart disease

7 References:

- [1] S. Khemmarat and L. Gao, "Supporting drug prescription via predictive and personalized query system," in *PervasiveHealth*. IEEE, 2015.
- [2] C. Knox et al., "Drugbank 3.0: a comprehensive resource for omics research on drugs," *Nucleic acids research*, vol. 39, no. suppl 1, pp. D1035–D1041, 2011.
- [3] M. Kuhn et al., "A side effect resource to capture phenotypic effects of drugs," *Molecular systems biology*, vol. 6, no. 1, p. 343, 2010.
- [4] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.
- [5] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [6] K. Sangkuhl et al., "Pharmgkb: understanding the effects of individual genetic variants," *Drug Metab. Rev.*, vol. 40, no. 4, pp. 539–551, 2008.
- [7] A. Langer et al., "A text based drug query system for mobile phones," *Int. J. Mob. Commun.*, vol. 12, no. 4, pp. 411–429, Jul. 2014.
- [8] C. Doulaverakis et al., "Panacea, a semantic-enabled drug recommendations discovery framework," *J. Biomed. Semant.*, vol. 5, p. 13, 2014.

Mani_Chand.pdf

ORIGINALITY REPORT

9%

SIMILARITY INDEX

3%

INTERNET SOURCES

4%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to SASTRA University

Student Paper

2%

2

Shan Zheng, Minzhen Wang, Shigong Wang, Kezheng Shang, Lili Hu, Jinrong Dong. "Effect of meteorological conditions on occurrence of hand, foot and mouth disease in Wuwei City, Northwestern China", 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI), 2011

Publication

2%

3

Submitted to Sreenidhi International School

Student Paper

1%

4

Submitted to Chicago State University

Student Paper

1%

5

arxiv.org

Internet Source

1%

6

www.nature.com

Internet Source

1%

7

dblp.dagstuhl.de

Internet Source

1 %

8

medicine.exeter.ac.uk

Internet Source

1 %

www.scribd.com

9

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On

