

DISEASE PREDICTION AND HOSPITAL RECOMMENDATION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in
Computer Science and Engineering

By

ONTEDDU MANASA(Reg.No.38110375)

PASUPULETI ANKITA NAGA SAI SWATHI(Reg.No.38110391)



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

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY) Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119**

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar 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 bonafied work of **ONTEDDU MANASA(Reg.No.38110375)** and **PASUPULETI ANKITA NAGA SAI SWATHI(Reg.No.38110391)** who carried out the project entitled "**Disease Prediction And Hospital Recommendation Using Machine Learning**" under my supervision from August 2021 to March 2022.

Internal Guide

Dr. M. SUGANIYA, M.E.,P.h.D

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **ONTEDDU MANASA** and **PASUPULETI ANKITA NAGA SAI SWATHI** hereby declare that the Project Report entitled “**Disease Prediction And Hospital Recommendation Using Machine Learning**” is done by me under the guidance of **Dr. M. SUGANIYA, M.E.,P.h.D** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M. SUGANIYA, M.E., P.h.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my 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

Hospital recommendation System is considered as an important factor in health care sector for managing the administrative, financial and clinical aspects of a hospital. Due to data mining progress in biomedical and healthcare communities, accurate study of medical data benefits early disease recognition, patient care and community services. When the quality of medical data is incomplete the exactness of study is reduced. Moreover, different regions exhibit unique appearances of certain regional diseases, which may results in weakening the prediction of disease outbreaks. In the proposed system, it provides machine learning algorithms for effective prediction of various disease occurrences in disease-frequent societies. It experiment the altered estimate models over real-life hospital data collected. To overcome the difficulty of incomplete data, it use a latent factor model to rebuild the missing data. It experiment on a regional chronic illness of cerebral infarction. Using structured and unstructured data from hospital it use Machine Learning Decision Tree algorithm. It predicts probable diseases and hospitals by mining data sets. To the best of our knowledge in the area of medical big data analytics none of the existing work focused on both data types. Compared to several typical estimate algorithms, the calculation exactness of our proposed algorithm reaches 94.8% with a convergence speed which is faster than that of the Decision tree disease risk prediction algorithm.

Table of contents

Chapter No.	Title	Page No.
	ABSTRACT	v
	LIST OF FIGURES	vii
1.	INTRODUCTION	1
	1.1 Overview	1
	1.2 Scope and Objective	2
	1.3 Domain Overview	2
	1.3.1 Machine Learning	3
	1.3.2 Proposed algorithms	6
2.	LITERATURE SURVEY	11
	2.1 Analysis of literature	11
	2.2 Literary review	12
3.	SYSTEM ANALYSIS	15
	3.1 Existing System	15
	3.2 Proposed System	15
	3.3 System Architecture	16
	3.4 System Requirement	17
	3.5 Software Description	17
	3.5.1 Applications of python	18
	3.5.2 Features of python	18
4.	SYSTEM IMPLEMENTATION	19
	4.1 User Module	19
	4.1.1 Login Module	19
	4.1.2 Registration Module	19
	4.2 Administration Module	20
	4.2.1 Add and update hospital information	20
	4.2.2 View feedback	20
	4.3 Disease Prediction Module	20

	4.4 Hospital Recommendation Module	20
5.	SOFTWARE TESTING	21
	5.1 General	22
	5.2 Test Driven Development	23
	5.3 Unit Testing	23
	5.4 Black Box Testing	24
	5.5 Integration Testing	24
	5.6 System Testing	25
	5.7 Regression Testing	26
6.	Result and Discussion	28
7.	CONCLUSION AND FUTURE WORK	34
	7.1 Conclusion	34
	7.2 Future Enhancement	34
	REFERENCES	35
	APPENDIX	36
	A. Sample coding	36
	B. Plagiarism Report	47

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
1.1	Supervised Architecture	4
1.2	Block Diagram	5
1.3	Structure of Decision tree	7
1.4	Structure of Random Forest	9
3.1	Architecture diagram	16
4.1	Data set for Disease prediction	20
5.1	Integration testing	25
5.2	Module testing	25
6.1	Home page	28
6.2	Web page for details	29
6.3	Safe Zone	29
6.4	Danger Zone	30
6.5	Gathering Multiple Symptoms	30
6.6	Predicting particular Disease	31
6.7	Page for Kidney Disease Detector	31
6.8	Detecting Safe	32
6.9	Page for heart disease Detector	32
6.10	Detecting danger	33

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

According to the statistics of National Health and Family Planning Commission, there are a total of 991,632 medical institutions in China until Nov. 2016 where new institutions are increased from Nov. 2015. Nevertheless, the difficulty of getting medical care remains as one of China's major livelihood issues. A survey on Peking University First Hospital, which is one of the most famous hospitals in Beijing, indicates that more than 45% outpatients have to wait for over two hours after the registration, whereas 85% of them have less than 10 minutes for the doctor's inquiry. This phenomenon is actually common in China's 776 Top-Class hospitals¹. The reason is that many hospitals lack publicity and are not familiar to most patients. Having no way of knowing whether there is a good enough and less crowded hospital nearby, people have no choice but go to those congested famous hospitals regardless of the severity of the disease. In fact, most outpatients with a mild disease may expect a quick treatment yet have a low requirement for the hospital's treatment ability. It is imperative to find a simple and convenient way to access the crowd status and basic information of the neighboring hospitals. There are several standard ways for crowd counting, such as approaches based on video or beacon. However, these methods rely on surveillance data or wireless network data, and any company or non-governmental organization can hardly gather these data of all hospitals even in one city, not to mention in any larger range. Fortunately, location-based service (LBS) big data offers a potential solution to this dilemma. LBS data have two unique properties: a) LBS data naturally belongs to the service providers. Using it for population density estimation does not involve asking hospitals for any help. b) LBS data is sufficient for population density estimation with its copiousness and vast area coverage. As smartphone is popular, several main LBS providers in China have hundreds of millions of users and preserve billions LBS request logs every day. For instance, Baidu Map has over 300 million active users and gets 23 billion LBS request logs in China each day on average. Taking advantage of these two properties, we present a novel (nearly) real-time model for counting and predicting the crowd of all hospitals in the city based on LBS big data. What's more, the distribution of people's residence time can be figured out with the LBS data. Relying on the location logs of Baidu LBS and Baidu Points of Interests data, we have designed an APP² to provide users with a) hospitals' real-time and predicted outpatient density, b) official level,

c) distance from the user to each hospital and the corresponding path planning.

To the best of our knowledge, it is the first hospital recommendation system based on the population density analysis with LBS big data. Note that Baidu LBS data has incorporated the information of GPS, WiFi, and Cellular network, so it is a relatively high-quality data source. Since the LBS data we use is location logs of mobile APPs, we will use data and logs interchangeably to denote the LBS data. Our contributions can be summarized into three innovations conquering three main challenges of this work. The first innovation is the model for detecting the types of people around hospitals to address highly noisy LBS data. Although our data source is relatively high-quality, it is still temporally.

1.2 SCOPE AND OBJECTIVE

The proposed software product is the hospital recommendation management system the system will be used in any hospital dispensary or pathology in any of the near location for hospital we have choose that one the current system in use paper based system it is slow and cannot provide update lists of patients within a reasonable timeframe.

The project of the hospital recommendation system is aimed to develop to maintain the day to day state of admission to computerize all details regarding patient details in this we have near location any hospital.

1.3 DOMAIN OVERVIEW

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future. Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

1.3.1 MACHINE LEARNING

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine learning models and finally, computer science is used for implementing algorithms. However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyper parameters to reach an optimum result.

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry. Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms.

With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it. These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions.

Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

Types of Machine Learning

Machine Learning Algorithms can be classified into 3 types as follows –

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

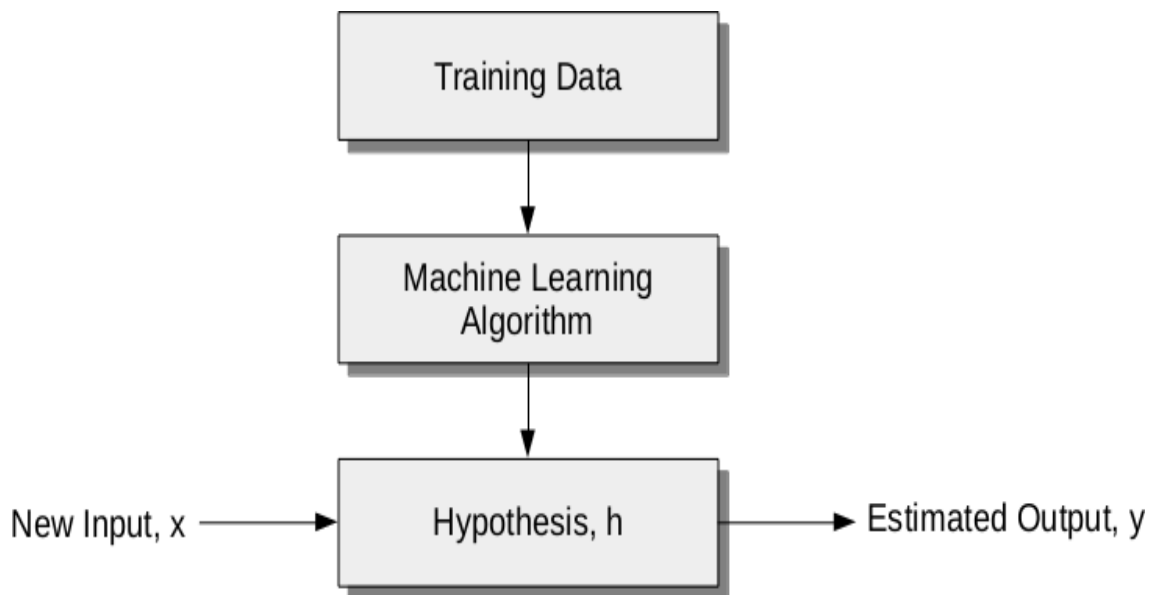


Fig -1.1 Supervised Architecture

SUPERVISED LEARNING

In the majority of supervised learning applications, the ultimate goal is to develop a finely tuned predictor function $h(x)$ (sometimes called the “hypothesis”). “Learning” consists of using sophisticated mathematical algorithms to optimize this function so that, given input data x about a certain domain (say, square footage of a house), it will accurately predict some interesting value $h(x)$ (say, market price for said house).

$$h(x_1, x_2, x_3, x_4) = \theta_0 + \theta_1 x_1 + \theta_2 x_3^2 + \theta_3 x_3 x_4 + \theta_4 x_1^3 x_2^2 + \theta_5 x_2 x_3^4 x_4^2$$

This function takes input in four dimensions and has a variety of polynomial terms. Deriving a normal equation for this function is a significant challenge. Many modern machine learning problems take thousands or even millions of dimensions of data to build predictions using hundreds of coefficients. Predicting how an organism’s genome will be expressed, or what the climate will be like in fifty years, are examples of such complex problems.

Under supervised ML, two major subcategories are:

- Regression machine learning systems: Systems where the value being predicted falls somewhere on a continuous spectrum.
- Classification machine learning systems: Systems where we seek a yes-or-no prediction.

In practice, x almost always represents multiple data points. So, for example, a housing price predictor might take not only square-footage (x_1) but also number of bedrooms (x_2), number of bathrooms (x_3), number of floors (x_4), year built (x_5), zip code (x_6), and so forth. Determining which inputs to use is an important part of ML design. However, for the sake of explanation, it is easiest to assume a single input value is used.

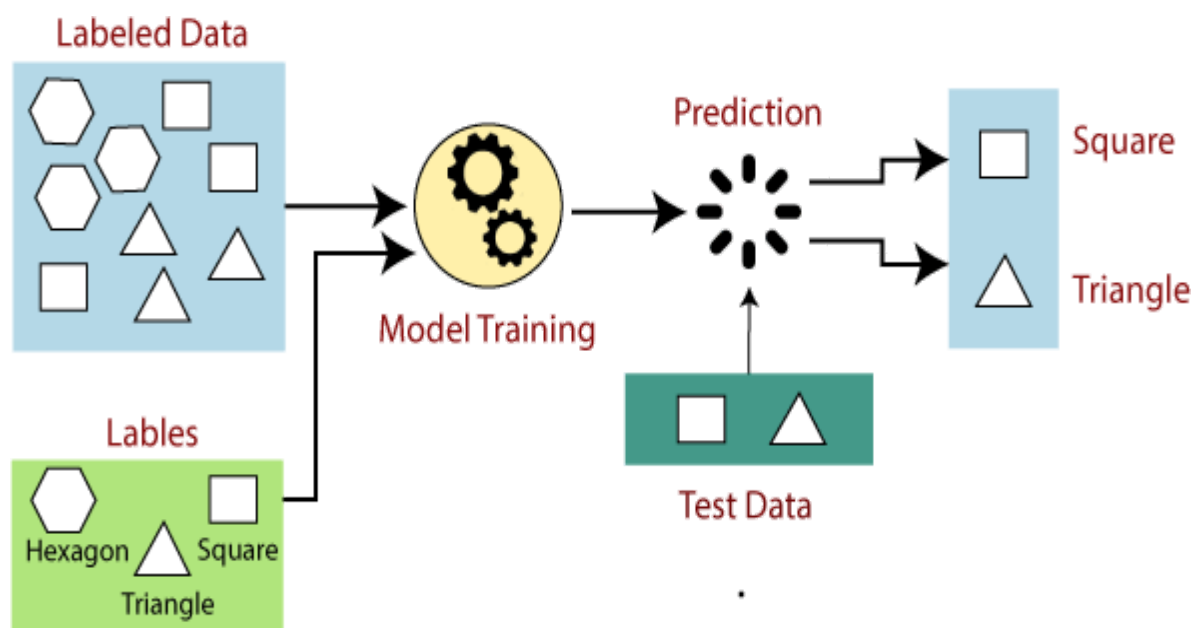


Fig-1.2 Block Diagram

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set.

REGRESSION

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.

1. Linear Regression
2. Regression Trees
3. Non-Linear Regression
4. Bayesian Linear Regression
5. Polynomial Regression

CLASSIFICATION

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Spam Filtering,

- Random Forest
- Decision Tree
- Logistic Regression
- Support vector Machines

1.3.2 PROPOSED ALGORITHMS

Decision Tree Classification Algorithm

- Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.

- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into sub trees.
- Below diagram explains the general structure of a decision tree:

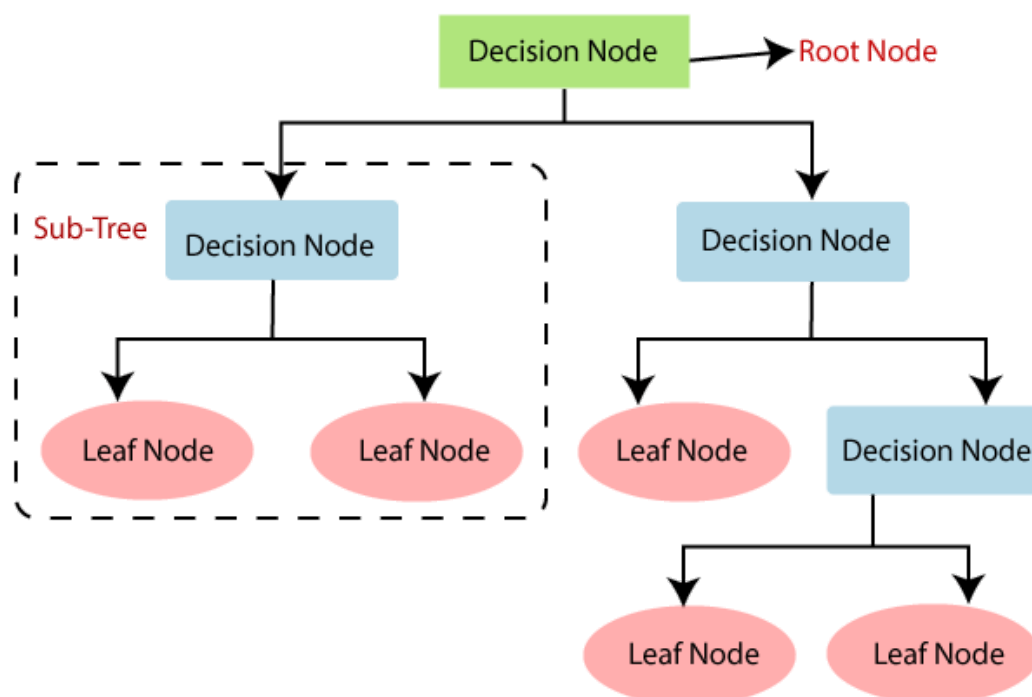


Fig-1.3 Structure of decision tree

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

- Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

- Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- A tree formed by splitting the tree known as branch tree.
- Pruning is the process of removing the unwanted branches from the tree.
- The root node of the tree is called the parent node, and other nodes are called the child nodes.

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an over fitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase.

RANDOM FOREST ALGORITHM

Random forest algorithm can use both for classification and the regression kind of problems. In this you are going to learn, how the random forest algorithm works in machine learning for the classification task.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression

problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The below diagram explains the working of the Random Forest algorithm:

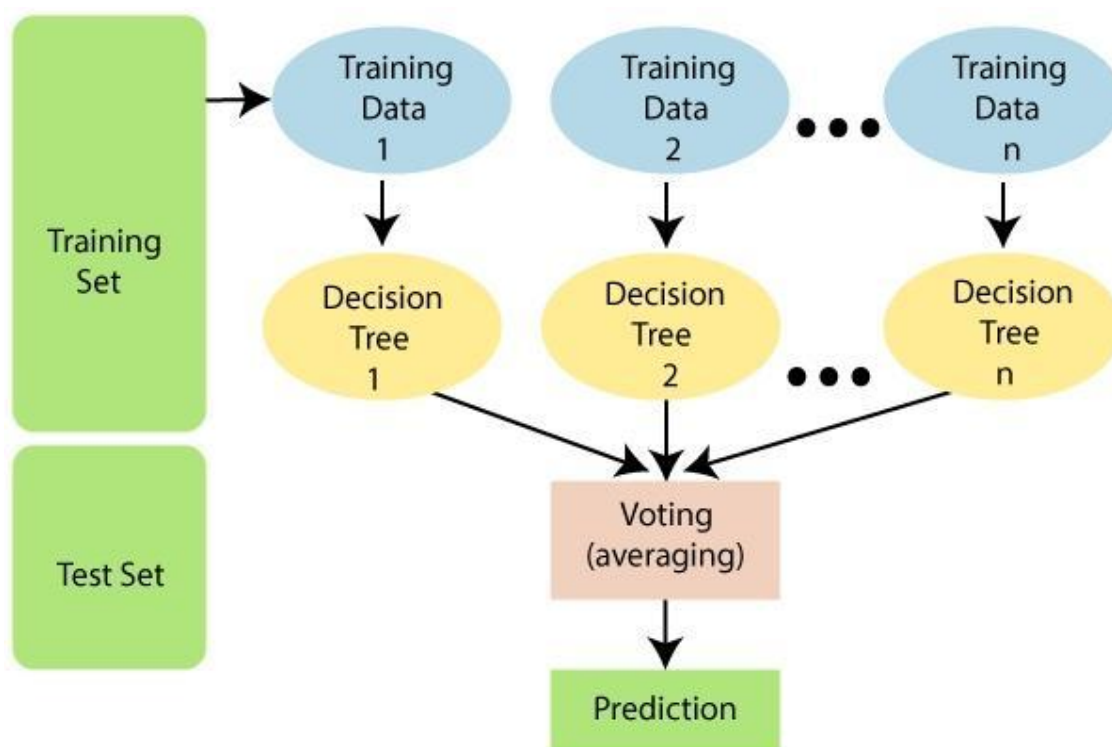


Fig-1.4 Structure of Random Forest

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Features of a Random Forest Algorithm

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of over fitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes. A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.

Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the over fitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

CHAPTER 2

LITERATURE SURVEY

2.1 ANALYSIS OF THE LITERATURE

Literature survey is the main advance in programming improvement measure. Prior to building up the instrument it is important to decide the time factor, economy and friends strength. When these things are fulfilled, at that point the subsequent stage is to figure out which working framework and language can be utilized for building up the device. When the developers begin assembling the apparatus the software engineers need parcel of outer help. This help can be gotten from senior developers, from book or from sites. The major part of the project development sector considers and fully survey all the required needs for developing the project. 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. Prior to building the framework the above thought are considered for building up the proposed framework. The significant piece of the undertaking advancement area considers and completely survey all the necessary requirements for building up the venture. For each undertaking Literature survey is the main area in programming improvement measure. Prior to building up the instruments and the related planning it is important to decide and survey the time factor, asset prerequisite, labor, economy, and friends strength. When these things are fulfilled and completely surveyed, at that point the following stage is to decide about the product details in the separate framework, for example, what kind of working framework the venture would require and what are largely the important programming are expected to continue with the subsequent stage like building up the apparatuses, and the related activities. Here we have taken the general surveys of different creators and noted down the fundamental central issues with respect to their work. In this venture literature survey assumes a prevailing part in get assets from different areas and all the connected points that are exceptionally valuable under this segment. The most awesome aspect if this is the manner in which things get all together and encourages us to suite our work according to the current information.

2.2 LITERARY REVIEWS

A New Load Prediction Method and Management of Distributed Power System in Island Mode of a Large Hospital (IEEE 2018)

Yuji Mizuno et al. described a new load prediction method and management of distributed power system in island mode of a large hospital. The purpose of this paper is to analysis a frequency stability of a large hospital grid for a combination of emergency generators (EGs) with a photovoltaic power generation (PV). A system frequency of the grid can be controlled by a droop controller of emergency generator system when the system has installed only one EG without PV system. When the PV system will install on the grid, the emergency generator system should be distributed to some generators of small type. It is reason why the power balance cannot keep the stabilize of the system frequency within all of power range due to too large EG. Since distributed generation system need a demand for each generator, it is important to predict a load. This paper presents a new load prediction method and management for stabilization with island mode in the large hospital power grid. It is clarified that the deviation of the system frequency has improved a stabilization using the proposed prediction method.

Social topic modeling for point-of-interest recommendation in location-based social networks (IEEE 2015)

B. Hu and M. Ester explained a social topic modeling point-of-interest recommendation in location-based social networks on data mining. In this paper, we address the problem of recommending Point-of-Interests (POIs) to users in a location-based social network. To the best of our knowledge, we are the first to propose the ST (Social Topic) model capturing both the social and topic aspects of user check-ins. We conduct experiments on real life data sets from Foursquare and Yelp. We evaluate the effectiveness of ST by evaluating the accuracy of top-k POI recommendation. The experimental results show that ST achieves better performance than the state-of-the-art models in the areas of social network-based recommender systems, and exploits the power of the location-based social network that has never been utilized before.

A Hospital Recommendation System Based on Patient Satisfaction Survey-2017

Mohammad Reza et al. This article is about Hospital Recommendation System based on patient satisfaction survey. Surveys are used by hospitals to evaluate patient satisfaction and to improve general hospital operations. Collected satisfaction data is usually represented to the hospital administration by using statistical charts and graphs.

Although such visualization is helpful, typically no deeper data analysis is performed to identify important factors which contribute to patient satisfaction. This work presents an unsupervised data-driven methodology for analyzing patient satisfaction survey data. The goal of the proposed exploratory data analysis is to identify patient communities with similar satisfaction levels and the major factors, which contribute to their satisfaction. This type of data analysis will help hospitals to pinpoint the prevalence of certain satisfaction factors in specific patient communities or clusters of individuals and to implement more proactive measures to improve patient experience and care.

Poi recommendation: A temporal matching between poi popularity and user regularity (IEEE 2016)

Z. Yao and Y. Fu states a temporal matching between poi popularity and user regularity, in 2016. Navigation is the most common way of deciding to know a choice, gathering data, and assessing different arrangements. Point of interest (POI) recommendation, which provides personalized recommendation of places to mobile users, is an important task in location-based social networks (LBSNs). However, quite different from traditional interest-oriented merchandise recommendation, POI recommendation is more complex due to the timing effects: we need to examine whether the POI fits a user's availability. While there are some prior studies which included the temporal effect into POI recommendations, they overlooked the compatibility between time-varying popularity of POIs and regular availability of users, which we believe has a non-negligible impact on user decision-making. To this end, in this paper, we present a novel method which incorporates the degree of temporal matching between users and POIs into personalized POI recommendations. Specifically, we first profile the temporal popularity of POIs to show when a POI is popular for visit by mining the spatio-temporal human mobility and POI category data. Secondly, we propose latent user regularities to characterize when a user is regularly available for exploring POIs, which is learned with a user-POI temporal matching function. Finally, results of extensive experiments with real-world POI check-in and human mobility data demonstrate that our proposed user-POI temporal matching method delivers substantial advantages over baseline models for POI recommendation tasks.

Disease prediction by Machine Learning over big data from Healthcare Communities (IEEE 2017)

Min Chen described about disease prediction by Machine Learning over big data from Healthcare Communities. With big data growth in biomedical and healthcare communities, accurate analysis of medical data benefits early disease detection, patient

care, and community services. However, 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. In this paper, we streamline machine learning algorithms for effective prediction of chronic disease outbreak in disease-frequent communities. We experiment the modified prediction models over real-life hospital data collected from central China in 2013-2015. To overcome the difficulty of incomplete data, we use a latent factor model to reconstruct the missing data. We experiment on a regional chronic disease of cerebral infarction. We propose a new convolutional neural network (CNN)-based multimodal disease risk prediction algorithm using structured and unstructured data from hospital. To the best of our knowledge, none of the existing work focused on both data types in the area of medical big data analytics. Compared with several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed, which is faster than that of the CNN-based uni modal disease risk prediction algorithm.

Social topic modelling for point of interest recommendation in location based social networks (IEEE 2014)

Bo Hu and Martin Ester suggests social topic modelling for point of interest recommendation in location based social networks. In this paper, we address the problem of recommending Point-of-Interests (POIs) to users in a location-based social network. To the best of our knowledge, we are the first to propose the ST (Social Topic) model capturing both the social and topic aspects of user check-ins. We conduct experiments on real life data sets from Foursquare and Yelp. We evaluate the effectiveness of ST by evaluating the accuracy of top-k POI recommendation. The experimental results show that ST achieves better performance than the state-of-the-art models in the areas of social network-based recommender systems, and exploits the power of the location-based social network that has never been utilized before.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Hospital currently use a manual system for the management and maintenance of critical information the current system requires numerous paper form with data stores spread through the hospital infrastructure often information's incomplete or does not follow recommendation standards form are often are lost in transmit between department requiring a comprehensive auditing process to ensure that no vital information is lost multiple copies of the same information exists in the hospital and may lead to inconsistence in data in various data stores here we have to developed the hospital recommendation and we apply the proposed model to 113 hospitals , and constructed an online hospital recommendation system which can provide users with a hospital rank list basing the real-time population density information and the hospitals' basic information such as hospitals' levels and their distances

DRAWBACKS OF EXISTING SYSTEM

- The most important disadvantage included reducing access,
- Reducing the rate of hospital admissions and increasing employees' workload and dissatisfaction.

3.2 PROPOSED SYSTEM

The hospital recommendation system is designed for any hospital to replace their existing manual paper based system the new system is to control the following information patient information here we have to the recommended the people near any hospitals with success rate can search in online these service are to be provided in an efficient with the goal of reducing the time and resources currently required for such task.

ADVANTAGES FOR PROPOSED SYSTEM

- The most important benefits were cost reduction.
- Increasing patient satisfaction.
- Increasing home care and outpatient services.

3.3 SYSTEM ARCHITECTURE

There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc.

For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support.

It describes the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.

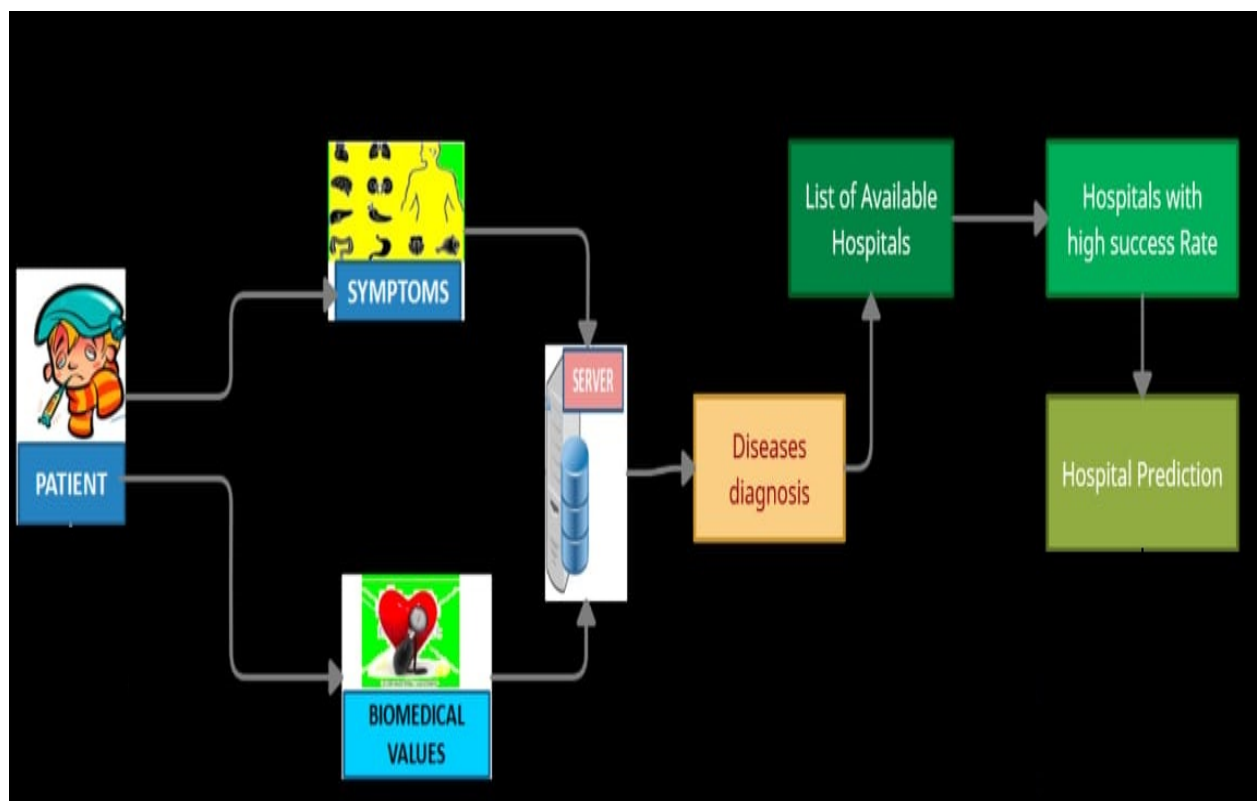


Fig- 3.1 Architecture diagram

3.4 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

System	: Pentium i3 Processor
Hard Disk	: 500 GB.
Monitor	: 15" LED
Input Devices	: Keyboard, Mouse
Ram	: 2 GB

SOFTWARE REQUIREMENTS

Operating system	: Windows 10
Coding Language	: Python

3.5 SOFTWARE DESCRIPTION

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems. Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.

In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).It was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.It is copyrighted. Like Perl, Python source code is now available under the GNU General Public License

(GPL). It is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

3.5.1 APPLICATIONS OF PYTHON

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

3.5.2 FEATURES OF PYTHON

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

CHAPTER 4

SYSTEM IMPLEMENTATION

LIST OF MODULES

- User Module
- Administration Module
- Disease prediction Module
- Hospital Recommendation Module

4.1 USER MODULE

4.1.1 Login Module

This is the first activity that opens when user open the website. User needs to provide a correct contact number and a password, which user enters while registering, in order to login into the website. If information provided by the user matches with the data in the database table then user successfully login into the website 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.

4.1.2 Registration Module

A new user who wants to access the webpage 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 webpage. Validations are applied on all the textboxes for proper functioning of the webpage. 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 webpage 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 webpage.

4.2 ADMINISTRATION MODULE

4.2.1 Add and update hospital information

This module, admin can check the hospital and then add, update some hospital information for using People.

4.2.2 View feedback

This module, admin can view the feedback about hospital application from user.

4.3 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.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
1	Disease	bloody_sto	fecal_leak	swelling	dizziness	confusion	fatigue	itching	vomiting	arm_pain	cough	muscle_pai	depression	fever	painful_bo	urine_bloo	sweating	nausea	stif
2	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
3	Malaria	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
4	Malaria	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0
5	Malaria	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0
6	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0
7	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0
8	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
9	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
10	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
11	Malaria	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0
12	Malaria	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
13	Malaria	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
14	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
15	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0
16	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
17	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
18	Malaria	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
19	Malaria	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
20	Malaria	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
21	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
22	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0
23	Malaria	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
24	Dengue	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	1

FIG 4.1 Data set for Disease Prediction

4.4 HOSPITAL RECOMMENDATION MODULE

The data collection, generation of results and analysis is done on the basis of structure, people, technology, and process perspective originating from the sociotechnical model. Findings of this study are presented in the form of hospital recommendations with predictive success rate.

CHAPTER 5

SOFTWARE TESTING

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases. In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed. The most popular testing documentation files are test reports, plans, and checklists. These documents are used to outline the team's workload and keep track of the process. Let's take a look at the key requirements for these files and see how they contribute to the process. Test strategy. An outline of the full approach to product testing. As the project moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

Test data. The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end-user in a ready solution.

Test plans. A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. It's the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

Test scenarios. In scenarios, testers break down the product's functionality and interface by modules and provide real-time status updates at all testing stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

- Testing can be done in the early phases of the software development lifecycle when other modules may not be available for integration
- Fixing an issue in Unit Testing can fix many other issues occurring in later development and testing stages
- Cost of fixing a defect found in Unit Testing is very less than the one found in the system or acceptance testing
- Code coverage can be measured

- Fewer bugs in the System and Acceptance testing
- Code completeness can be demonstrated using unit tests. This is more useful in the agile process. Testers don't get the functional builds to test until integration is completed.
- Code completion cannot be justified by showing that you have written and checked in the code. But running Unit tests can demonstrate code completeness.
- Expect robust design and development as developers write test cases by understanding the specifications first.
- Easily identify who broke the build
- Saves development time: Code completion may take more time but due to decreased defect count overall development time can be saved.

5.1 GENERAL

Unit Testing frameworks are mostly used to help write unit tests quickly and easily. Most of the programming languages do not support unit testing with the inbuilt compiler. Third-party open source and commercial tools can be used to make unit testing even more fun.

List of popular Unit Testing tools for different programming languages:

- Java framework – JUnit
- PHP framework – PHPUnit
- C++ frameworks – UnitTest++ and Google C++
- .NET framework – NUnit
- Python framework – py.test

Functional Testing is a type of black box testing whereby each part of the system is tested against functional specification/requirements. For instance, seek answers to the following questions,

Are you able to login to a system after entering correct credentials?
Does your payment gateway prompt an error message when you enter incorrect card number? Does your “add a customer” screen adds a customer to your records successfully?

Well, the above questions are mere samples to perform full-fledged functional testing of a system.

5.2 TEST DRIVEN DEVELOPMENT

Test Driven Development, or TDD, is a code design technique where the programmer writes a test before any production code, and then writes the code that will make that test pass. The idea is that with a tiny bit of assurance from that initial test, the programmer can feel free to refactor and refactor some more to get the cleanest code they know how to write. The idea is simple, but like most simple things, the execution is hard. TDD requires a completely different mind set from what most people are used to and the tenacity to deal with a learning curve that may slow you down at first.

5.3 UNIT TESTING

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details. Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development). Most of us might know the classic definition of Unit Testing – “Unit Testing is the method of verifying the smallest piece of testable code against its purpose.” If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

5.4 BLACKBOX TESTING

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks app's functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

5.5 INTEGRATION TESTING:

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

COMPONENT INTEGRATION TESTING:

Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger project, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

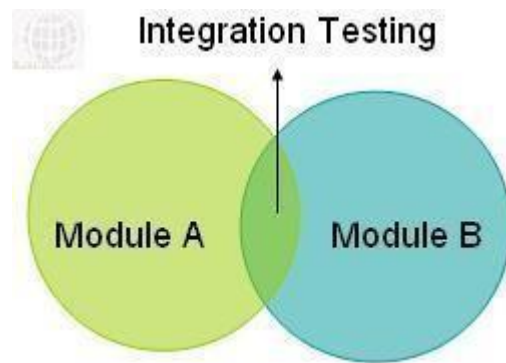


Fig 5.1 Integration Testing

5.6 SYSTEM TESTING

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not. System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.

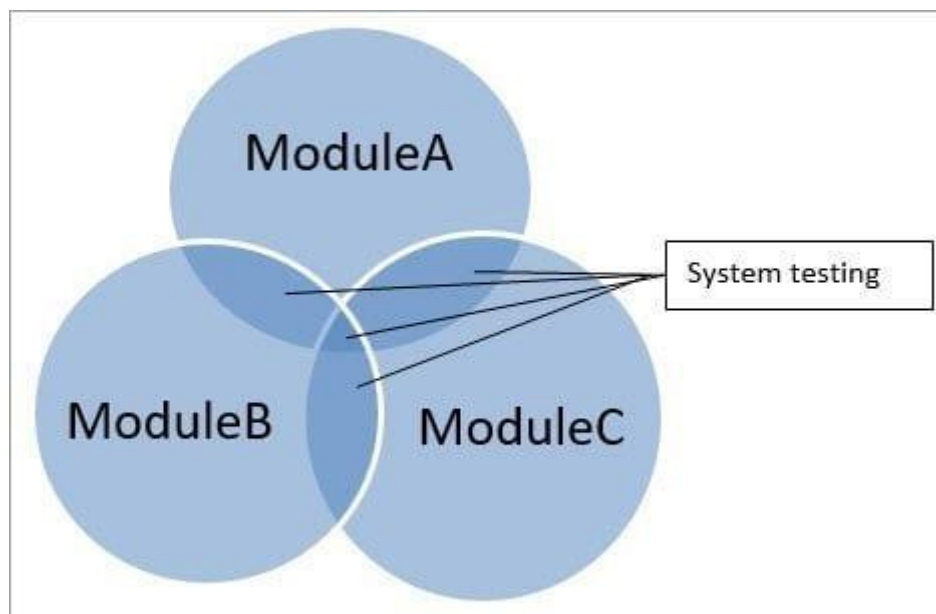


Fig 5.2 Module Testing

System testing is a method of monitoring and assessing the behaviour of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal

knowledge of the coding, programming, design, etc., and is completely based on users-perspective.

A black box testing type, system testing is the first testing technique that carries out the task of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is done next to the Integration testing, as this testing should be covering the end-to-end system's actual routine. This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the customer's point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing; it seeks to detect both defects within the "inter-assemblages".

5.7 REGRESSION TESTING

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs. This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix. For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough variety—such as well-designed frontend UI automated tests alongside targeted unit testing, based on

smart risk prioritization—to prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule.

But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent projects from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your company's bottom line. Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. The process incorporates "performance" indicators such as:

- Browser, page, and network response times
- Server request processing times
- Acceptable concurrent user volumes

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as "Endurance testing" and "Volume testing". The main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load. Load testing comes under the Non Functional Testing & it is designed to test the non-functional requirements of a software application.

Load testing is perform to make sure that what amount of load can be withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.

CHAPTER 6

RESULTS AND DISCUSSION

As a result of this project Figure-6.1 shows the homepage of our application provided information for predicting diseases like coronavirus, kidney failure and cardiac related problems.

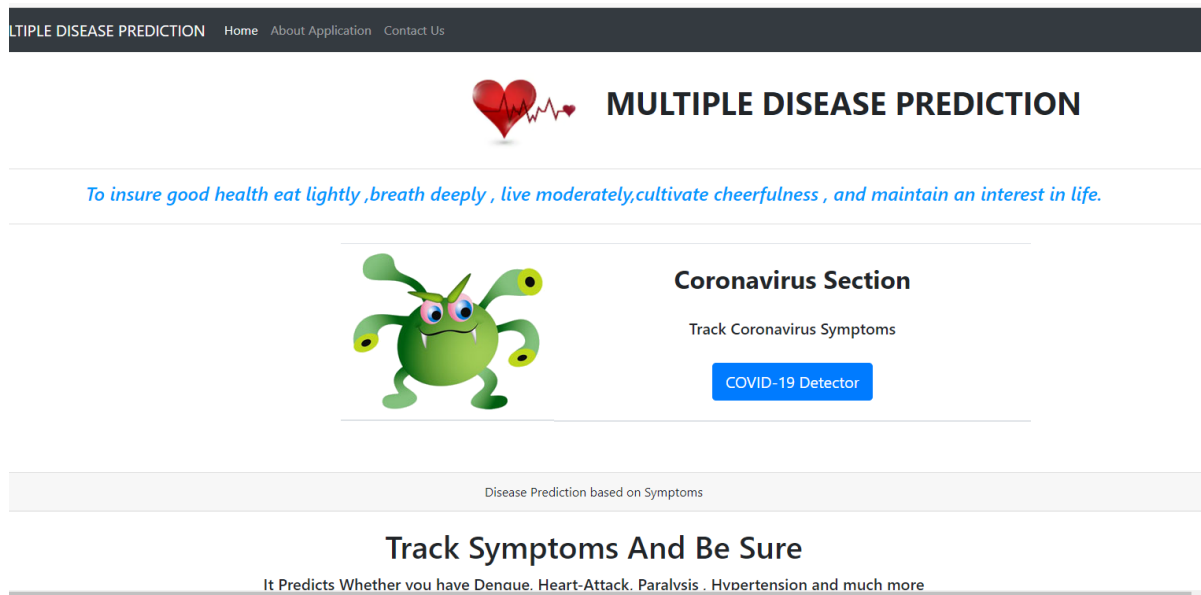


Figure-6.1 Homepage

The Figure-6.2 gives information of a page to fill the details and symptoms of patient like temperature, cough, cold, sore throat etc.

Coronavirus Probability Detector



Enter Body Temperature :

Age :

Cough : ☒ Yes ☐ No

Cold : ☒ Yes ☐ No

Sore Throat : ☒ Yes ☐ No

Body Pain : ☒ Yes ☐ No

Fatigue : ☐ Yes ☒ No

Headache : ☒ Yes ☐ No

Diarrhea : ☐ Yes ☒ No

Difficulty in breathing : ☒ Yes ☐ No

Have you travelled recently during past 14 days :

Do you have a travel history to a COVID-19 INFECTED AREA :

Do you have direct contact or is taking care of a positive COVID-19 PATIENT :

Figure-6.2 Webpage for details

The Figure-6.3 shows the analysis of patient symptoms and gives the patient condition is safe if the patient is tested as negative

[Impulse](#) [Home](#) [About Application](#) [Contact Us](#)

Safe

You are Safe !!

To insure good health eat lightly ,breath deeply ,live moderately,
cultivate cheerfulness , and maintain an interest in life .

Return to Home

Figure-6.3 Safe zone

The Figure-6.4 shows the analysis of patient symptoms and gives the patient condition is danger if the patient tested as positive also provide some of the best hospitals.

Danger

You maybe suffering from Coronavirus

Most Probable Hospital For Coronavirus

1)Apollo Hospital, Greams Road with 99% successrate
 Doctor 1: Dr A R Gayathri(95)
 Doctor 2: Dr Jebini Roger S(90)

2)Appasamy Hospital, Arumbakkam with 96% successrate
 Doctor 1: Dr.Priitam Moon(91)
 Doctor 2: Dr.Belram(91)

3)ArunVijaya Hospitals, KK Nagar with 85% successrate
 Doctor 1: Dr.Arunachalam(80)
 Doctor 2: Dr.Rajendiran N(73)

Figure-6.4 Danger zone

Figure-6.5 tells about the medical condition prediction page for filling the information and multiple symptoms of patient for any disease

Gmail YouTube Maps


Medical Condition Prediction

Are you experiencing any of these symptoms below (mark all those applicable)

☒ Fever
☐ Stomach Pain
☐ Itching
☐ Sweating

☒ Cough
☐ Vision problem
☐ Muscle Pain

☐ Vomiting
☐ Confusion
☐ Depression
☐ Shoulder pain

☐ Cold
☐ Arm pain
☐ Headache

☐ Stiff Neck
☐ Chest Pain
☐ Skin rash
☒ Back Pain

☐ Low heartbeat
☐ Yawning
☐ Swelling

☐ Nausea
☐ Low breath
☐ Fatigue
☐ Painful Bowel moments

☐ Dry mouth
☐ Stool pressure
☐ Urine Blood

Figure-6.5 Gathering multiple symptoms

Figure-6.6 Tracking multiple Symptoms and predicting whether you have Dengue, Heart-Attack, Paralysis , Hypertension and much more diseases based on the analysis of patient symptoms and provide best hospitals for the particular disease.

Multiple Disease Prediction Home About Application Contact Us

Danger

You maybe suffering from Heart_Attack

Most Probable Hospital's

1)Be Well Hospital with 95% successrate

Doctor 1: Dr. KR. Sekar(92)
Doctor 2: Dr. K.M. Srirem(89)

2)SRM Healthcare,Chennai with 90% successrate

Doctor 1: Dr.Rajasekaran(86)
Doctor 2: Dr.Saidhar Hosmi(78)

3)Fortis Malar Hospital, Chennai with 83% successrate

1) HOSPITAL RECORDS Show all

Figure-6.6 Predicting a particular disease

The Figure-6.7 shows the page for filling the details like hemoglobin, hypertension and some other values regarding Kidney disease

← → ↻ ⓘ 127.0.0.1:5000/CKD ⌵ ☆ ⚙ □ S ⋮

Chronic Kidney Disease Home About Application Contact Us

Chronic Kidney Disease Detector 

Specific gravity :

Albumin :

Serum chritening :

Hemoglobin :

Packed cell volume :

Hypertension : ☒ Yes ☐ No

Fig-6.7 Page for Kidney disease detector

The Figure-6.8 shows the analysis of patient is suffering from kidney disease or not based on the given symptoms.

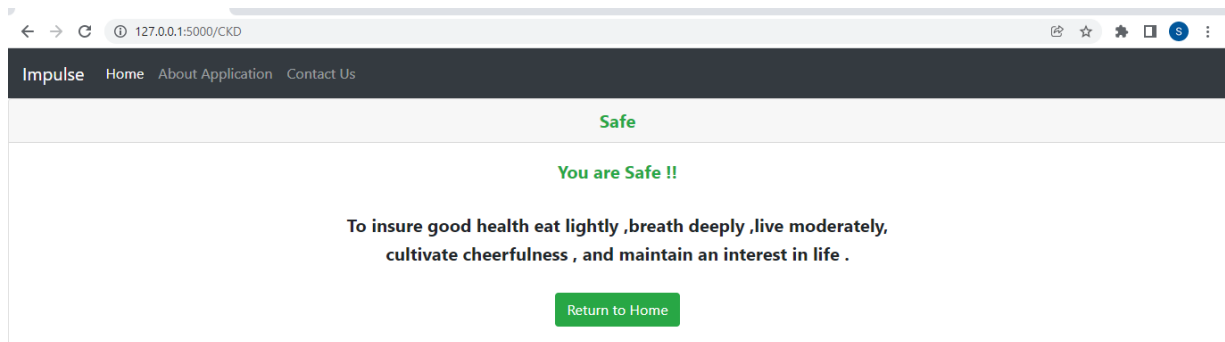


Fig 6.8-Detecting Safe

The Figure-6.9 shows the page for filling the details like blood pressure, glucose, cholesterol regarding Heart disease.

The screenshot shows a web browser window with the URL 127.0.0.1:5000/HeartDisease. The page has a dark header with 'Heart Disease Detector' and links to 'Home', 'About Application', and 'Contact Us'. The main content area is white and features the title 'Heart Disease Probabilty Detector' with a red heart icon. Below the title, there are several input fields and radio buttons: 'Age' (50), 'Gender' (Male selected, Female unselected), 'Height' (160), 'Weight' (80), 'Systolic blood pressure' (160), 'Diastolic blood pressure' (80), 'Cholesterol' (High unselected, Medium selected, Low unselected), and 'Glucose' (High selected, Medium unselected, Low unselected).

Fig-6.9 Page for heart disease detector

The Figure-6.10 shows the analysis of patient is suffering from Heart disease or not and provide best hospitals to cure the disease.

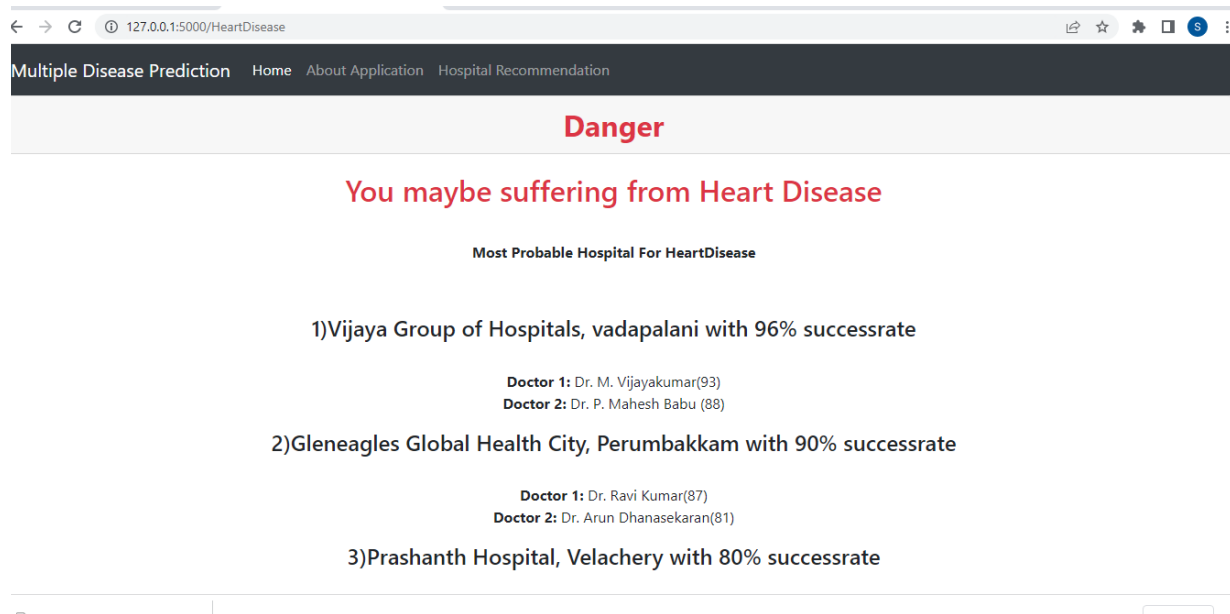


Fig-6.10 Detecting Danger

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

We presented a novel approach for hospital population density estimation based on Symptoms and Disease. In this paper, we propose an approach for hospital recommendation based on disease. 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.

7.2 Future Enhancement

The analysis of collected automated medical data has been established to have the potential in providing adequate recommendations for the improvement and management of patients with special needs in the hospitals through different algorithm developments and knowledge discovery.

REFERENCES

[1] Yuji Mizuno; Teppei Baba; Yoshito Tanaka; Fujio Kurokawa; Masaharu Tanaka; Ilhami Colak; Nobumasa Matsu "A New Load Prediction Method and Management of Distributed Power System in Island Mode of a Large Hospital"/2018

[2]B. Hu and M. Ester, "Social topic modeling for point-of-interest recommendation in location-based social networks," in 2014 IEEE International Conference on Data Mining (ICDM), pp. 845–850, IEEE, 2014.

[3]Mohammad Reza Khoie, Tannaz Sattari Tabrizi,Elham Sahebkar Khorasani, Shahram Rahimi and Nina Marhamati "A Hospital Recommendation System Based on Patient Satisfaction Survey/2017"

[4]E. Velasco, T. Agheneza, K. Denecke, G. Kirchner, and T. Eckmanns, "Social media and internet-based data in global systems for public health surveillance: a systematic review," *Milbank Quarterly*, vol. 92, no. 1, pp. 7–33, 2014.

[5]Z. Yao, Y. Fu, B. Liu, Y. Liu, and H. Xiong, "Poi recommendation: A temporal matching between poi popularity and user regularity," in 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 549– 558, IEEE, 2016

[6]Min Chen, Yixue Hao, Kai Hwang, Lu Wang. "Disease prediction by Machine Learning over big data from Healthcare Communities" IEEE-2015

[7]Bo Hu and Martin Ester "Social topic modelling for point of interest recommendation in location based social networks"/2014 IEEE

APPENDIX

A.SAMPLE CODING

```
from flask import Flask, render_template, request, url_for, send_file, flash,
redirect, make_response

import pickle

import numpy as np

import os

import json

import termcolor

import smtplib


import CurrentStats

# import CancerModel

# import PdfConverter

# from PdfConverter import PDFPageCountError

import DiseasePred


# import warnings


app = Flask(__name__)

app.config['SECRET_KEY'] = '73a4b6ca8cb647a20b71423e31492452'


# For Coronavirus

with open("Coronavirus_logistic", "rb") as f:

    logisticRegression = pickle.load(f)
```

```
# For Chronic kidney disease
with open("CKD_Model", "rb") as f:
    decisionTree = pickle.load(f)
```

```
# For Heart Disease
with open("HeartDisease", "rb") as f:
    randomForest = pickle.load(f)
```

```
@app.route("/")
@app.route("/home")
def Homepage():
    # cases, cured, death = CurrentStats.currentStatus()
    return render_template("Homepage.html", feedback="False")
```

```
@app.errorhandler(404)
def page_not_found(e):
    return render_template("PageNotFound.html")
```

```
# @app.route("/currentstats", methods=["POST", "GET"])
# def CurrentStatus():
#     cases, cured, death = CurrentStats.currentStatus()
#     scases = scured = sdeath = 0
#     state = ""
#     try:
#         if request.method == "POST":
```

```

#         # print(request.form)
#         formDict = request.form
#         state = formDict['state']
#         # print(state)
#         scases, scured, sdeath = CurrentStats.StateStatus(state)
#     except UnboundLocalError:
#         flash("The State is not Affected Yet")
#     except ValueError:
#         flash("The State is not Affected Yet")
#     return render_template("CurrentStats.html", state=state, scases=scases,
# scured=scured, sdeath=sdeath, cases=cases, cured=cured, death=death,
# title="Current Statistics", navTitle="Current Status", headText="Coronavirus
# Current Stats Statewise", ImagePath="/static/Virus.png")

```

```
@app.route("/about")
```

```
def About():
```

```
    return render_template("About.html")
```

```
@app.route("/contact", methods=["POST", "GET"])
```

```
def Contact():
```

```
    if request.method == "POST":
```

```
        # print(request.form)
```

```
        contactDict = request.form
```

```
        firstname = contactDict['firstname']
```

```
        lastname = contactDict['lastname']
```

```
        email = contactDict['email']
```

```
        phone = int(contactDict['phone'])
```

```

description = contactDict['description']

subject = "Medical Website feedback !!"

message = f"First Name : {firstname} \nLast Name : {lastname} \nEmail :
{email}\nPhone Number : {phone}\nDescription : {description}\n"

content = f"Subject : {subject} \n\n{message}"

sender = "intmain1221@gmail.com"

receiver = "intmain1221@gmail.com"

password = "intmain@11"

print(content)

try:

    with smtplib.SMTP("smtp.gmail.com", 587) as mail:

        mail.ehlo()

        mail.starttls()

        mail.login(sender, password)

        mail.ehlo()

        mail.sendmail(sender, receiver, content)

    print("Mail Send Successfully !")

    cases, cured, death = CurrentStats.currentStatus()

    return render_template("Homepage.html", cases=cases, cured=cured,
death=death, feedback="True")

except:

    pass

return render_template("Contact.html")

```

```
@app.route("/infected")
def Infected():
    return render_template("Infected.htm", disease="Nothing")
```

```
@app.route("/noninfected")
def NonInfected():
    return render_template("NonInfected.htm")
```

```
@app.route("/download")
def Download():
    file = "static/Example.docx"
    return send_file(file, as_attachment=True)
```

```
@app.route("/BreastCancer", methods=["POST", "GET"])
def BreastCancer():
    # if request.method == "POST":
    #     f = request.files['inputFile']

    #     name, extension = os.path.splitext(f.filename)
    #     print(extension)
    #     try:
    #         if extension == ".png" or extension == ".jpg" or extension == ".jpeg"
or extension == ".pdf":
            location = os.path.join("Received_Files", f.filename)
            f.save(location)
            print("File Saved !")
```

```

#         if extension == ".pdf":
#             PdfConverter.Convert(f.filename)
#             image = name + ".png"
#             prediction = CancerModel.Predict(
#                 os.path.join("Received_Files", image))
#         else:
#             prediction = CancerModel.Predict(
#                 os.path.join("Received_Files", f.filename))
#         print(prediction)
#         if prediction:
#             return render_template("Infected.htm", disease="Breast Cancer
")
#         else:
#             return render_template("NonInfected.htm")

#     else:
#         flash(
#             "Please upload files with extension 'png', 'pdf' , 'jpg' or 'jpeg'")

# except ValueError:
#     flash("Please Upload Only Valid Files ")
# except PDFPageCountError:
#     flash("Please Upload Only Valid Files , or try again later")
return render_template("breastcacernotfound.html")

```

```

@app.route("/HeartDisease", methods=["POST", "GET"])

```

```

def Heart_disease():

```

```

    if request.method == "POST":

```

```

# print(request.form)

heart_dict = request.form
age = int(heart_dict['age'])
gender = int(heart_dict['gender'])
height = int(heart_dict['height'])
weight = int(heart_dict['weight'])
sbp = int(heart_dict['sbp'])
dbp = int(heart_dict['dbp'])
cholesterol = int(heart_dict['cholesterol'])
glucose = int(heart_dict['glucose'])
smoke = int(heart_dict['smoke'])
alcohol = int(heart_dict['alcohol'])
active = int(heart_dict['active'])
age = age*365

model_input = [age, gender, height, weight, sbp,
               dbp, cholesterol, glucose, smoke, alcohol, active]
prediction = randomForest.predict([model_input])[0]

if prediction:
    return render_template("HeartInfected.htm", disease="Heart Disease")
else:
    return render_template("NonInfected.htm")

return render_template("HeartDisease.html", title="Heart Disease Detector",
navTitle="Heart Disease Detector", headText="Heart Disease Probabilty
Detector", ImagePath="/static/HeartPulse.png")

@app.route("/diseaseprediction", methods=["POST", "GET"])

```

```

def Disease():
    symptoms = []
    if request.method == "POST":
        rf = request.form
        # print(rf)
        for key, value in rf.items():
            # print(key)
            symptoms.append(value)
        print(symptoms)

    if len(symptoms) < 5 or len(symptoms) > 8:
        flash("Please Select symptoms only between 5 and 8 Inclusive")
    else:
        prediction = DiseasePred.predicts(symptoms)
        if prediction:
            return render_template("Infected.htm", disease=prediction)
        else:
            return render_template("NonInfected.htm")
    return render_template("dp.html")

```

```
@app.route("/CKD", methods=["POST", "GET"])
```

```
def CKD():
```

```

    if request.method == "POST":
        submitted_values = request.form
        sg = str(float(submitted_values["sg"].strip()))
        albumin = str(float(submitted_values["albumin"].strip()))
        hemoglobin = str(float(submitted_values["hemoglobin"].strip()))
        pcv = str(float(submitted_values["pcv"].strip()))

```

```

hypertension = str(float(submitted_values["hypertension"].strip()))
sc = str(float(submitted_values["sc"].strip()))

ckd_inputs1 = [sg, albumin, sc, hemoglobin, pcv, hypertension]
prediction = decisionTree.predict([ckd_inputs1])
# print("*****          ", prediction)

if not prediction:
    return render_template("CKDInfected.htm", disease="Chronic Kidney
Disease")
else:
    return render_template("NonInfected.htm")

return render_template("ChronicKidney.html", title="Chronic Kidney Disease",
navTitle="Chronic Kidney Disease", headText="Chronic Kidney Disease
Detector", ImagePath="/static/Chronic_Kidney.png")

@app.route("/CoronavirusPrediction", methods=["POST", "GET"])
def Coronavirus():
    if request.method == "POST":
        # print(request.form)
        submitted_values = request.form
        temperature = float(submitted_values["temperature"].strip())
        age = int(submitted_values["age"])
        cough = int(submitted_values["cough"])
        cold = int(submitted_values["cold"])
        sore_throat = int(submitted_values["sore_throat"])
        body_pain = int(submitted_values["body_pain"])
        fatigue = int(submitted_values["fatigue"])

```

```

headache = int(submitted_values["headache"])
diarrhea = int(submitted_values["diarrhea"])
difficult_breathing = int(submitted_values["difficult_breathing"])
travelled14 = int(submitted_values["travelled14"])
travel_covid = int(submitted_values["travel_covid"])
covid_contact = int(submitted_values["covid_contact"])

age = 2 if (age > 50 or age < 10) else 0
temperature = 1 if temperature > 98 else 0
difficult_breathing = 2 if difficult_breathing else 0
travelled14 = 3 if travelled14 else 0
travel_covid = 3 if travel_covid else 0
covid_contact = 3 if covid_contact else 0

model_inputs = [cough, cold, diarrhea,
                 sore_throat, body_pain, headache, temperature,
difficult_breathing, fatigue, travelled14, travel_covid, covid_contact, age]

prediction = logisticRegression.predict([model_inputs])[0]
# print("*****", prediction)

if prediction:
    return render_template("CoronaInfected.htm", disease="Coronavirus")
else:
    return render_template("NonInfected.htm")

return render_template("Coronavirus.htm", title="Coronavirus Prediction",
navTitle="COVID-19 Detector", headText="Coronavirus Probability Detector",
ImagePath="/static/VirusImage.png")

```

```
if __name__ == '__main__':
    app.run(threaded=True, debug=True)
```

C. PLAGIARISM REPORT :

**DISEASE PREDICTION AND HOSPITAL RECOMMENDATION USING
MACHINE LEARNING**

P.AnkitaNagaSaiSwathi Student professor Department of CSE Sathyabama Institute of Science and Technology Rajahmundry,India swathipasupuketi0309@gmail.com	Onteddu Manasa Student Department of CSE Sathyabama Institute of Science and Technology Kadapa,India manasaonteddu@gmail.com suganiya_cse@sathyabama.ac.in	Dr. M.Suganiya Assistant Department of CSE Sathyabama Institute of and Technology Chennai,India
--	---	--

Abstract:

The hospital conveyance framework is viewed as a significant piece of the wellbeing area in emergency hospital the executives, money, and clinical consideration. In the event that the nature of clinical data isn't completely met, exploration will be decreased. Moreover, various locales show explicit elements of a specific illness in a given area, which is a disservice in anticipating infection. In the proposed systems, it gives AI research calculations to anticipate the different illnesses that happen in the weakest families. Use hospital introduced and unstructured data utilizing the Machine Learning Certificate Tree calculation. Hospitalization is anticipated by mining.

Keywords: Hospital recommendation, Machine Learning Decision Tree algorithm, KNN.

Introduction:

As per figures from the National Commission for Health and Family Planning, 991,632 wellbeing communities have been implicit China since November 2016, and new hardware has been added since November 2015. Notwithstanding, admittance to medical services stays one of the biggest China social issues. As indicated by a review led by Peking University's first medical hospital, one of Beijing's most famous emergency hospitals, in excess of 45% of patients stand by over two hours later enrollment, while 85% go through under 10 minutes with a specialist. Area. This peculiarity is generally normal in 776 top Chinese hospitals¹. This is on the grounds that numerous hospitals need mindfulness and are inexperienced with numerous patients. Individuals who couldn't say whether there is a sufficient hospital, with a low-speed close by, paying little mind to the seriousness of the sickness, should go to that notable and exceptional medical hospital. Indeed, numerous patients with minor diseases are relied upon to be dealt with critically, however don't should be dealt with. It is vital to track down a basic and simple method for getting fundamental data and how to meet in a close by hospital. There are numerous normal strategies for computing, like video or light-based techniques. Be that as it may, this strategy depends on control measurements or organization data, and no non-legislative association or association can gather data from all medical hospitals in a similar city, with the exception of a more extensive level. Fortunately, location-based service (LBS) gives extraordinary data that gives the most ideal answer for this issue. LBS data has two particular elements: a) True LBS data, the specialist co-op. Utilizing it in a populace enumeration doesn't need hospital

help. b) LBS information are adequate to appraise populace thickness and appropriation in huge regions. As indicated by well-known cell phones, numerous LBS suppliers in China have countless clients and stay aware of the billions of LBS required every day. For instance, the Baidu Card, which has in excess of 300 million clients and registers a normal of 23 billion LBS each day in China. With the assistance of these two assets, we are presenting a new (close consistent) method of counting and anticipating all city gatherings dependent on LBS enormous information. Also, the clearing system not really settled from LBS. Every emergency hospital and related street arrangement. Apparently, this is the primary method for conveying local area based medical services with huge LBS. If it's not too much trouble, note that Baidu LBS data is a decent wellspring of data as it incorporates GPS, Wi-Fi and cell network data. Since the LBS data we use is an archive showing the area of the APP, we will utilize the data and data we trade to decide the LBS data. Our commitment can be separated into three developments that have conquered three key business challenges. The primary advancement is an example of individuals close to the medical hospital to address LBS clamor filled data. The beginning of our news is awesome; however, it is brief.

Scope and objective:

The arranged product offering is a conveyance framework, which can be utilized in any medical hospital or in any piece of the medical hospital, and we have decided to give a quicker, more cutting-edge list than the paper framework. We currently use to the patient at the ideal opportunity.

Literature survey:

Yuji Mizuno and others. characterized another strategy for overseeing power and anticipating diminished burdens as enormous clinic beds.

Neighbours are a straightforward calculation that stores generally potential cases and makes another case dependent on comparative rules. KNN has been utilized to look at measurements and decide the construction that started to be utilized as an alternate method during the 1970s calculation.

B. Hu and M.Ester disclosed their longing to show online media on an intriguing theme via web-based media stages. A Bayesian image is a calculation that utilizes the Bayes hypothesis to figure things out. ... These classes are generally utilized in AI since they are not difficult to execute. Guiltless Bays are normal or Bay is free.

Mohammad Reza nar. This article is about the procedure for exhorting in light of fascinating examination. Here we utilize the K-Means clinical endorsement procedure. Lessen hospitalization and increment staff turnover and disappointment. The fundamental benefit was cost decrease.

E. Velasco and others. Person to person communication destinations and online data in the worldwide general wellbeing checking framework. This article diminishes the extent of general wellbeing reconnaissance by normalized and episode based observing, every one of which has been featured. We give a total outline of the articles distributed in the PubMed, Scopus, and chronicles somewhere in the range of 1990 and 2011.

As per Z.Yao and Yu.Fu, in 2016, there was a long hole between poi fame and customer backing. Strolling is a typical method for knowing your choices, to assemble data, and

to assess various settings. Going for a little stroll can assist you with capitalizing on your learning and tomfoolery.

Ming Chen clarifies the illness guess made by Machine Learning on enormous information from Life. Here they used CNN depends on various illnesses that anticipate the danger of calculations utilizing hospital caught and non-caught information. In this article, we will improve on AI calculations to all the more likely recognize ongoing infections in the populace.

Bo Hu and Martin Esther give social analysis to communicate their advantage in open proprietorship. The issue of asking for use out in the open spots has been settled. This is an incredible method for utilizing your innocent Bayes classifier calculation to flaunt your interactive abilities and sort things out.

Today, hospitals utilize a portable framework to oversee and store significant data, and the current framework requires a great deal of paper and data stockpiling to disseminate to emergency hospital foundation. Fragmented or unacceptable data might be shipped off the Central Audit Office to forestall the deficiency of significant data, just as to forestall the deficiency of numerous duplicates of some clinical data, prompting distortion of data in different media information bases. There is a need to further develop hospital needs and make an internet-based hospital conveyance framework that can be utilized on the case of 113 medical hospitals. It can give clients a rundown of hospitals dependent on fundamental clinical data, for example, genuine populace thickness, hospital level, and distance. The main disadvantages in the existing system are the principle impact is to decrease the interaction and diminish hospitalizations,

increase the worker fulfillment and disappointment.

Methodology:

The medical clinic application framework is relied upon to supplant the expected records in the new emergency clinic framework with the going with patient data observing.

We really want to ask individuals close to the emergency hospital to offer this assistance to decrease the time and assets needed for such exercises. The principle benefits were decreased expenses, expanded patient fulfillment, and expanded in-home medical services and out-of-home consideration.

Figure-1 Architecture diagram

Dataset:

This data is used for disease identification for patient suffering from certain symptoms. After identifying the disease based on the prediction here we are recommending a best hospital in their nearby location.

Modules:

- User Module
- Administration Module
- Disease prediction Module
- Hospital Recommendation Module

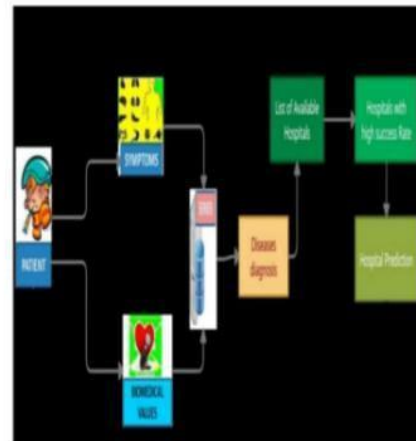
User Module

i) Login Module

These are the initial steps to open when a client opens a site. All together for a client to sign in to the site, the client should enter a substantial secret phrase and secret phrase to sign in. Assuming that the data given by the client is in the information base, the client has effectively signed in to the site, in any case the login message will fall flat and the client should affirm once more. A connect to the enrollment interaction is likewise included to enlist new clients.

ii) Registration Module

New clients wishing to get to the site should enlist prior to signing in. Enrolled clients presently need to login to get to the site. All containers are authorized for legitimate



activity of the site. Similarly as with the data in each container, it is critical that each crate is clear when you register, like a name, connection, secret key, or secret word check. Assuming there is a clear record, the program will give the message that ought to be in each case. You should likewise join secret phrase data and secret key confirmation to enroll appropriately. On the other hand, the contact telephone number check should be a 10-digit

number. At the point when the site is vacant, a message will show up. Assuming all of this data is right, it will lead the client to the login cycle to get to the site.

Administration Module

Login: To sign in to the application, the client should enter a substantial secret phrase and secret key that the client entered during enlistment. Assuming that the data given by the client is in the chronicle table, the client has effectively signed in to the application, in any case the login message will come up short and the client should affirm once more. A connect to the enlistment cycle is likewise included to enroll new clients. Adding and Updating Medical Information In this module, the administrator can really take a look at the emergency hospital and afterward add and update clinical data to utilize People. View Comments This module permits administrators to see clients' remarks about the emergency hospital.

Disease prediction Module

The patient will show side effects brought about by his disease. The framework will pose explicit inquiries about their disease, and the framework will design the ailment dependent on the patient's side effects and allude it to doctors dependent on the sickness.

Hospital Recommendation Module

Information assortment, creation, and examination are in fact dependent on structure, individuals, innovation, and activities. The aftereffects of this review are introduced as clinical applications and achievement rates.

ALGORITHM:

Decision Trees for Classification: A Machine Learning Algorithm:

Introduction

The choice tree is a sort of control machine where data is continually conveyed by specific standards (this implies information and result are remembered for the activity). A tree can be characterized by two elements: navigation and insidiousness. Leaves are an ultimate choice or arrangement. Choices are made based on data sharing. An illustration of a tree arrangement can be clarified utilizing a double tree above. Assume you need to anticipate an individual's life as indicated by data like age, dietary patterns, and proactive tasks. Also, the leaves are the "right" or "wrong" arrangement. For this situation, it was a two-section issue (type no issue).

K-Nearest Neighbor (KNN) classification method:

model sort dependent on learning, or lethargic realizing, where execution is slow and slow until the computation is totally carried out. The k-NN calculation is the least complex of all AI calculations. Neighbors were eliminated from things known by classification

Stage 1: Get Started

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

Stage 3: another instance of carrying out $x = (x_1 \dots x_n)$

Stage 4: Calculate $d(x_i, x)$ for each sign (x_i, c_i) .

Stage 5: Sort $d(x_i, x)$ from base to top, $(I = 1 \dots N)$

Stage 6: x: Select K close to the model in Dkx

Stage 7: Set x to the most widely recognized classification of Dkx

Stage 8: Conclusion

Result:

Figure-2 shows the homepage of our application provided information for predicting diseases like coronavirus, kidney failure and cardiac related problems.



Figure-2 Homepage

Figure-3 gives information of a page to fill the details and symptoms of patient like temperature, cough, cold, sore throat etc.



Figure-3 Webpage for details

Figure-4&5 shows the analysis of patient symptoms and gives the patient condition is safe if the patient is tested as negative and danger if the patient tested as positive also provide some of the best hospitals.



Figure-4 Safe zone



Figure-5 Danger zone

Figure-6 tells about the medical condition prediction page for filling the information and multiple symptoms of patient for any disease

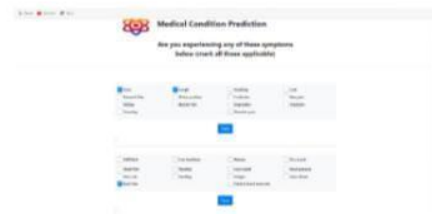


Figure-6 Gathering multiple symptoms

Figure-7 Tracking multiple Symptoms and predicting whether you have Dengue, Heart-Attack, Paralysis, Hypertension and much more diseases based on the analysis of patient symptoms and provide best hospitals for the particular disease.



Figure-7 Predicting a particular disease

Conclusion:

We have shown another technique for assessing the emergency hospital populace dependent on proof and illness. In this article we offer how to apply to the medical hospital based on the illness. Our emphasis is on the most proficient method to find, rank, and give explicit data dependent on deficient data. To conquer deficient and boisterous data, proper game plans and terminations are permitted when addressing questions. We additionally offer inside and out answers for assist clients with understanding the normal outcomes and work on their presentation.

References:

- [1] Yuji Mizuno; Teppei Baba; Yoshito Tanaka; Fujio Kurokawa; Masaharu Tanaka; Ilhami Colak; Nobumasa Matsu "A New Load Prediction Method and Management of Distributed Power System in Island Mode of a Large Hospital"/2018
- [2]B. Hu and M. Ester, "Social topic modeling for point-of-interest recommendation in location-based social networks," in 2014 IEEE International Conference on Data Mining (ICDM), pp. 845–850, IEEE, 2014.
- [3]Mohammad Reza Khoie, Tannaz Sattari Tabrizi,Elham Sahebkar Khorasani, Shahram Rahimi and Nina Marhamati "A Hospital Recommendation System Based on Patient Satisfaction Survey/2017"
- [4]E. Velasco, T. Agheneza, K. Denecke, G. Kirchner, and T. Eckmanns, "Social media and internet-based data in global systems for public health surveillance: a systematic review," *Milbank Quarterly*, vol. 92, no. 1, pp. 7–33, 2014.
- [5]Z. Yao, Y. Fu, B. Liu, Y. Liu, and H. Xiong, "Poi recommendation: A temporal matching between poi popularity and user regularity," in 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 549–558, IEEE, 2016
- [6]Min Chen, Yixue Hao, Kai Hwang, Lu Wang. "Disease prediction by Machine Learning over big data from Healthcare Communities" IEEE-2015
- [7]Bo Hu and Martin Ester "Social topic modelling for point of interest recommendation in location based social networks"/2014 IEEE

DISEASE PREDICTION AND HOSPITAL RECOMMENDATION USING MACHINE LEARNING (1).docx

ORIGINALITY REPORT

1 %	1 %	1 %	0 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	S. Gowri, J. Jabez, Jeberson Retna Raj, Senduru Srinivasulu, Sudha. "An Enhanced Big Data Handling Architecture for Privacy Preservation of Cloud Data", 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), 2021 Publication	1 %
2	www.ijert.org Internet Source	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches Off