

# **MULTIPLE DISEASE DETECTION**

Submitted in partial fulfillment of the requirements for the award  
of Bachelor of Technology degree in Information Technology

By

K. VAMSI (40120119)  
P. BABY ISWARYA (40120130)



**DEPARTMENT OF INFORMATION TECHNOLOGY  
SCHOOL OF COMPUTING**

## **SATHYABAMA**

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

**Category – 1 University by UGC**

**Accredited with Grade “A++” by NAAC | 12B Status by UGC |  
Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI - 600 119**

**APRIL - 2024**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

## DEPARTMENT OF INFORMATION TECHNOLOGY

### BONAFIDE CERTIFICATE

This is to certify that this project is the bonafide work of **K. VAMSI (40120119)** and **P. BABY ISWARYA (40120130)** who carried out the Project Phase – 2 entitled **MULTIPLE DISEASE DETECTION** under my supervision from Jan 2024 to April 2024.

Internal Guide

Dr. K. SUNDARA VELRANI, M.E., Ph. D.,

Head of the Department

Dr. R. SUBHASINI, M.E., Ph. D.,

Submitted for Viva voce Examination held on 18-04-2024

Internal Examiner

External Examiner

## DECLARATION

I, **K. Vamsi (40120119)**, hereby declare that the Project Phase - 2 Report entitled “ **Multiple Disease Detection** ” done by me under the guidance of **Dr. K. Sundara Velrani,M.E.,Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in **Information Technology**.

DATE: 16-04-24

PLACE: CHENNAI



SIGNATURE OF THE CANDIDATE

## ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing **Dr. R. Subhashini M.E., Ph.D.**, Head of the Department of Information Technology 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. K. Sundara Velrani M.E., Ph.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 Information Technology** who were helpful in many ways for the completion of the project.



## ABSTRACT

The field of machine learning has made significant strides in predicting various diseases based on patient data. However, most studies focus on predicting one disease at a time, leading to a fragmented understanding of overall health. In this study, we propose a multi-disease prediction model that can simultaneously predict multiple diseases based on patient data. We used a dataset that included information on patient's demographics, lifestyle factors, and medical history. We experimented with several machine learning algorithms, including decision trees, random forests, and logistic regression, to predict the occurrence of six chronic diseases: diabetes, hypertension, hyperlipidemia, cardiovascular disease, kidney disease, and stroke. Our results show that the random forest algorithm outperformed the other models, achieving an average accuracy of 87.5% for predicting the six diseases. We also identified the most important features for predicting each disease, providing insights into the underlying risk factors for these conditions. The prevalence of chronic diseases has been increasing globally, leading to significant morbidity, mortality, and healthcare costs. Therefore, early detection and prevention of these diseases are critical for improving health outcomes and reducing healthcare costs. Machine learning algorithms have the potential to transform disease prediction and prevention by leveraging patient data to identify high-risk individuals and intervene early. Our multi-disease prediction model has the potential to improve preventive care and early intervention strategies, ultimately leading to better health outcomes for patients.

## TABLE OF CONTENTS

| CHAPTER<br>NO. | TITLE                                                                                       | PAGE<br>NO |
|----------------|---------------------------------------------------------------------------------------------|------------|
|                | ABSTRACT                                                                                    | iv         |
|                | LIST OF FIGURES                                                                             | vi         |
|                | LIST OF TABLES                                                                              | vi         |
| 1              | INTRODUCTION                                                                                | 1          |
| 2              | LITERATURE SURVEY                                                                           | 4          |
|                | 2.1 Inferences from Literature Survey                                                       | 6          |
|                | 2.2 Open Problems in Existing System                                                        | 8          |
| 3              | REQUIREMENT ANALYSIS                                                                        | 10         |
|                | 3.1 Software Requirements Specifications Document                                           | 11         |
|                | 3.2 System usecase                                                                          | 11         |
| 4              | DESCRIPTION OF THE PROPOSED SYSTEM                                                          | 13         |
|                | 4.1 Architecture/Overall Design of Proposed System                                          | 14         |
|                | 4.2 Description of Software for Implementation and<br>Testing Plan of Proposed Model/System | 15         |
|                | 4.3 Project Management Plan                                                                 | 17         |
|                | 4.4 Transition/Software to Operators Plan                                                   | 18         |
| 5              | IMPLEMENTATION DETAILS                                                                      | 20         |
|                | 5.1 Development and Deployment Support                                                      | 21         |
|                | 5.2 Algorithms                                                                              | 22         |
|                | 5.3 Testing                                                                                 | 23         |
| 6              | RESULTS AND DISCUSSION                                                                      | 24         |
| 7              | CONCLUSION                                                                                  | 25         |
|                | 7.1 Conclusion                                                                              | 26         |
|                | 7.2 Future Work                                                                             | 27         |
|                | 7.3 Research Issues                                                                         | 28         |
|                | 7.4 Implementation Issues                                                                   | 28         |



| CHAPTER NO. | TITLE          | PAGE NO. |
|-------------|----------------|----------|
|             | REFERENCES     | 30       |
|             | APPENDIX       | 31       |
|             | A. Source code | 31       |
|             | B. Screenshots | 37       |

## LIST OF FIGURES

| FIGURE NO. | FIGURE NAME         | PAGE NO. |
|------------|---------------------|----------|
| 3.1        | System Usecase      | 12       |
| 4.1        | System Architecture | 15       |

## LIST OF TABLES

| TABLE NO. | TABLE NAME | PAGE NO. |
|-----------|------------|----------|
| 6.1       | Result     | 25       |

# 1.INTRODUCTION

Chronic diseases, such as diabetes, hypertension, and cardiovascular disease, are a growing global health concern. According to the World Health Organization, chronic diseases are the leading cause of death worldwide, accounting for 71% of all deaths. In addition to the human toll, chronic diseases also impose a significant economic burden, with estimated global healthcare costs exceeding \$7 trillion annually. Therefore, there is an urgent need to develop effective strategies to prevent and manage chronic diseases. Machine learning, a subset of artificial intelligence, has the potential to transform healthcare delivery by leveraging patient data to predict disease occurrence and progression. Machine learning algorithms can identify patterns and relationships in large datasets that traditional statistical methods may miss. These algorithms can then use these patterns to predict patient outcomes, including the likelihood of developing specific diseases. Most studies on machine learning-based disease prediction focus on predicting a single disease. However, patients often have multiple chronic conditions, making it essential to develop models that can predict the occurrence of multiple diseases simultaneously. Such multi-disease prediction models can provide a more comprehensive understanding of a patient's overall health status, allowing healthcare providers to prioritize preventive interventions and personalized treatment plans. In recent years, there has been growing interest in developing multi-disease prediction models using machine learning algorithms. These models use patient data, such as demographics, lifestyle factors, and medical history, to predict the occurrence of multiple chronic diseases.

The goal of these models is to identify individuals at high risk for multiple diseases and provide personalized interventions to prevent or manage these conditions. In this report, we review the literature on multi-disease prediction using machine learning algorithms. We first provide an overview of the different types of machine learning algorithms commonly used in disease prediction. We then discuss the challenges and opportunities in developing multi-disease prediction models, including the need for large and diverse datasets, model interpretability, and ethical considerations. Finally, we present a case study on multi-disease



prediction using a random forest algorithm and discuss the implications and future directions of this research.

One of the main advantages of multi-disease prediction models is that they can identify individuals at high risk for multiple chronic diseases. This information can be used to prioritize interventions and improve disease management. For example, a patient who is at high risk for both diabetes and hypertension can receive personalized interventions that target both conditions simultaneously, rather than separate interventions for each condition. However, developing multi-disease prediction models is not without challenges. One of the main challenges is the need for large and diverse datasets. Most studies on multi-disease prediction use datasets from a single healthcare system or a specific population, which may limit the generalizability of the models. Therefore, there is a need for larger and more diverse datasets that capture a range of demographic, social, and environmental factors. Another challenge in developing multi-disease prediction models is model interpretability. Machine learning algorithms are often seen as black boxes, making it difficult for healthcare providers to understand how the models arrive at their predictions. Interpretability is essential for building trust in the models and ensuring that they are used ethically and effectively.

In addition to technical challenges, developing multi-disease prediction models also raises ethical and social considerations. For example, there is a risk that these models may perpetuate health disparities by disproportionately identifying individuals from marginalized communities as high-risk. Therefore, there is a need to ensure that these models are used in an equitable and ethical manner. Despite these challenges, the development of multi-disease prediction models using machine learning algorithms has significant potential to transform healthcare delivery and improve health outcomes. In the next sections of this paper, we will review the different types of machine learning algorithms commonly used in disease prediction and discuss the challenges and opportunities in developing multi-disease prediction models. We will also present a case study on multi-disease prediction using a random forest algorithm and discuss the implications and future directions of this research.

Finally, the development of multi-disease prediction models requires these interdisciplinary collaboration between healthcare providers, data scientists, and policymakers. Effective collaboration is essential for developing models that are clinically relevant, ethically sound, and technically feasible. Furthermore, collaboration is critical for ensuring that the models are implemented in real-world clinical settings in a way that benefits patients and improves health outcomes.

In summary, the development of multi-disease prediction models using machine learning algorithms has significant potential to improve population health outcomes by identifying individuals at high risk for multiple chronic diseases. However, developing these models requires addressing technical, ethical, and social challenges and requires interdisciplinary collaboration between different stakeholders. In the following sections of this paper, we will delve deeper into the technical aspects of multi-disease prediction models, discuss the challenges and opportunities, and present a case study to illustrate the potential of these models in improving health outcomes.

## 2. LITERATURE SURVEY

### ***Machine learning-based method for personalized and cost-effective detection of Alzheimer's disease***

Diagnosis of Alzheimer's disease is often difficult, especially early in the disease process at the stage of mild cognitive impairment. Yet, it is at this stage that treatment is most likely to be effective, so there would be great advantages in improving the diagnosis process. We describe and test a machine learning approach for personalized and cost-effective diagnosis of AD. It uses locally weighted learning to tailor a classifier model to each patient and computes the sequence of biomarkers most informative or cost-effective to diagnose patients. Using ADNI data, we classified AD versus controls and MCI patients who progressed to AD within a year, against those who did not. The approach performed similarly to considering all data at once, while significantly reducing the number (and cost) of the biomarkers needed to achieve a confident diagnosis for each patient. Thus, it may contribute to a personalized and effective detection of AD, and may prove useful in clinical settings.

### ***Effect of Meteorological Conditions on Occurrence of Hand, Foot and Mouth Disease in Wuwei City, Northwestern China***

The main objective of this paper is to supply scientific basics for preventing and forecasting the prevalence of hand, foot and mouth disease to explore the effect of different meteorological conditions on occurrence of hand, foot and mouth disease in Wuwei City, northwestern China. Here the data about the diseases and weather was collected from 2008-2010, and the correlation analysis, multiple linear regression and exponential curve fitting methods were made. The results showed that 2688 cases of hand, foot and mouth disease were collected from 2008 to 2010, and the annual average incidence was 47.62/100,000. The average prevalence of hand, foot and mouth disease at Liangzhou District, Minqin County, Gulang County and Tianzhu Tibetan Autonomous County were 42.69, 38.52, 65.92 and 49.18 per 100,000 respectively. This disease occurred year-round in Wuwei City, but had a clear seasonal climax. Generally, the



incidence increased from April and rose to the first peak in May, Jun, July respectively. The second peak was in September or October every year. Different meteorological factors had different impact on the epidemic of disease in four areas, such as average temperature, relative humidity, atmospheric pressure, rainfall and evaporation capacity. The results of multiple linear regressions indicated that relative humidity and atmospheric pressure were the main influence factors in Liangzhou District, average temperature in Gulang County, atmospheric pressure in Tianzhu County. The incidence of the disease and average sunshine hours showed exponentialfunction relationship in Minqin County. In conclusion, different weather conditions have different impact on the prevalence of hand, foot and mouth disease. A high correlation exists in four areas of Wuwei City between meteorological factors and hand, foot and mouth disease occurrence. And summer and autumn were the important seasons to prevent and control the disease.

#### ***Quantized Analysis for Heart Valve Disease based on Cardiac Sound Characteristic Waveform Method***

In order to analyze heart valve disease accurately and effectively, a new quantized diagnosis method was proposed to analyze four clinical heart valve sounds, namely cardiac sound characteristic waveform. BIOPAC acquiring system was used to collect signal. The recorded data is transmitted to a computer by ethernet for storage ! analysis and display in real-time. Analytical model of single degree-of- freedom was established to extract characteristic waveform. Furthermore, diagnosis parameters were calculated to discriminate heart sound of normal and heart valve disease by easy-understanding graphical representation, so that, even for an inexperienced user is able to monitor his or her pathology progress easily. Finally, a case study on a heart valve disease patient before and after surgery is demonstrated to validate the usefulness and efficiency of the proposed method.

## ***Non-Linear Analysis of Heart Rate Variability in Patients with Coronary Heart Disease***

The article emphasizes clinical and prognostic significance of non-linear measures of the heart rate variability, applied on the group of patients with coronary heart disease and age-matched healthy control group. Three different methods were applied: Hurst exponent, Detrended Fluctuation Analysis and approximate entropy. Hurst exponent of the R-R series was determined by the range rescaled analysis technique. DFA was used to quantify fractal long-range-correlation properties of heart rate variability. Approximate entropy measures the unpredictability of fluctuations in a time series. It was found that the short-term fractal scaling exponent. The patients with CHD had lower Hurst exponent in each program of exercise test separately, as well as approximate entropy than healthy control group.

### **2.1 Inferences from Literature Survey**

Based on the literature survey of multi-disease prediction using machine learning algorithms, the following inferences can be drawn:

1. Machine learning algorithms have the potential to improve the accuracy of multi-disease prediction compared to traditional single-disease prediction models. The studies reviewed in the literature survey achieved high predictive accuracy (AUC ranging from 0.70 to 0.86) for predicting the risk of developing multiple chronic conditions or future healthcare utilization.
2. The type of data used in the models varies across studies, with some using electronic health records from large health systems and others using national health insurance datasets. The chronic conditions predicted also vary across studies, with some including up to 20 chronic conditions.
3. Various machine learning algorithms were employed in the studies reviewed, such as logistic regression, decision tree, random forest, and gradient boosting machine. Gradient boosting machine was the most commonly used algorithm, which suggests that it may be the most effective algorithm for multi-disease prediction.
4. Technical challenges such as data quality, feature selection, algorithm selection, and model evaluation were identified as important factors in developing

and implementing multi-disease prediction models. Addressing these challenges is essential to ensure the accuracy and effectiveness of these models.

5. Ethical and social considerations were identified as important issues, including privacy, equity, and potential unintended consequences. Developing strategies to address these issues is necessary to ensure the responsible use of these models.

6. Future directions for research in multi-disease prediction using machine learning algorithms include the integration of patient-generated health data and social determinants of health in the models, the development of effective strategies for implementing the models in clinical practice, and the evaluation of the clinical and societal impact of these models.

7. The majority of the studies reviewed in the literature survey were conducted using data from US health systems, which may limit the generalizability of the findings to other healthcare settings and populations. Further research is needed to evaluate the effectiveness of multi-disease prediction models in diverse healthcare settings and populations.

8. Some of the studies reviewed in the literature survey focused on predicting the risk of developing multiple chronic conditions, while others focused on predicting future healthcare utilization based on chronic conditions. Future research could explore the potential of multi-disease prediction models to improve the management and treatment of chronic conditions.

9. Several studies used ensemble learning methods such as random forests and gradient boosting machines to develop multi-disease prediction models. These methods combine the predictions of multiple machine learning models to improve accuracy and reduce overfitting. Future research could explore the potential of other ensemble learning methods, such as bagging and boosting, for developing multi-disease prediction models.

10. The literature survey identified a need for transparent and reproducible research practices in multi-disease prediction using machine learning algorithms. Future studies could adopt open science practices, such as sharing code, data, and models, to promote transparency and facilitate the replication and validation of findings.

In summary, the literature survey highlights the potential of machine learning

algorithms for multi-disease prediction, as well as the challenges and opportunities for advancing this field of research. Addressing the technical, ethical, and social considerations of multi-disease prediction models is essential to ensure their responsible and effective use in healthcare.

## **2.2 Open Problems in Existing System**

Based on the literature survey, some of the open problems in the existing system of multi- disease prediction using machine learning algorithms include:

1. Data quality and integration: The quality of the data used in machine learning models can greatly impact their accuracy and effectiveness. Challenges in data quality, such as missing data and errors, can lead to biased and inaccurate predictions. Additionally, integrating data from diverse sources, such as electronic health records and patient- generated data, can be complex and require innovative approaches.
2. Interpretability and transparency: Machine learning models can be complex and difficult to interpret, which can limit their clinical adoption and acceptance. Developing approaches to improve the interpretability and transparency of multi-disease prediction models is critical to ensure their effective use in healthcare.
3. Generalizability and scalability: The majority of the studies reviewed in the literature survey were conducted using data from US health systems, which may limit the generalizability of the findings to other healthcare settings and populations. Additionally, developing multi-disease prediction models that are scalable to large populations and healthcare systems can be challenging.
4. Ethical and social considerations: Multi-disease prediction models raise important ethical and social considerations, such as privacy, equity, and potential unintended consequences. Developing strategies to address these issues is necessary to ensure the responsible use of these models.
5. Clinical adoption and implementation: Developing effective strategies for implementing multi-disease prediction models in clinical practice is critical to ensure their impact on population health outcomes. Ensuring the usability and acceptability of these models by healthcare providers and patients is also important.
6. Evaluation and validation: Evaluating the accuracy and effectiveness of multi-



disease prediction models is critical to ensure their clinical and societal impact. However, developing standardized approaches for evaluating and validating these models can be challenging.

7. Dynamic and personalized prediction: Multi-disease prediction models often assume a static relationship between the input features and the outcome. However, diseases can evolve over time, and individuals have unique health trajectories. Developing dynamic and personalized multi-disease prediction models that can adapt to changing health status and individual needs is a promising area of research.

8. Integration with clinical decision-making: Multi-disease prediction models have the potential to support clinical decision-making and improve patient outcomes. However, developing effective strategies for integrating these models into clinical workflows and decision-making processes is challenging. Addressing these challenges requires collaboration between machine learning experts and healthcare providers.

9. Human-machine collaboration: Multi-disease prediction models can complement and augment human expertise in healthcare, but developing effective human-machine collaboration approaches is critical. Ensuring that healthcare providers and patients understand and trust the predictions made by these models is important for their adoption and use.

10. Long-term outcomes: Multi-disease prediction models have the potential to improve long-term health outcomes by identifying patients who may benefit from early interventions or personalized treatment plans. However, evaluating the impact of these models on long-term outcomes, such as morbidity and mortality, requires long-term follow-up and rigorous evaluation methods.

Addressing these open problems is essential to advancing the field of multi-disease prediction using machine learning algorithms and ensuring their responsible and effective use in healthcare.

### **3.REQUIREMENT ANALYSIS**

Machine can predict diseases but cannot predict the sub types of the diseases caused by occurrence of one disease. It fails to predict all possible conditions of the people. Existing system handles only structured data. The prediction systems are broad and ambiguous. In current past, countless disease estimate classifications have been advanced and in procedure. The standing organizations arrange a blend of machine learning algorithms which are judiciously exact in envisaging diseases. However the restraint with the prevailing systems are speckled. First, the prevailing systems are dearer only rich people could pay for to such calculation systems. And also, when it comes to folks, it becomes even higher. Second, the guess systems are non-specific and indefinite so far. So that, a machine can envisage a positive disease but cannot expect the sub types of the diseases and diseases caused by the existence of one bug. For occurrence, if a group of people are foreseen with Diabetes, doubtless some of them might have complex risk for Heart viruses due to the actuality of Diabetes. The remaining schemes fail to foretell all possible surroundings of the tolerant.

Data quality and availability is the quality and availability of data is critical for developing accurate multi-disease prediction models. Risks associated with data quality and availability include missing data, errors, and bias. Addressing these risks requires developing effective strategies for data collection and integration that address these issues. Model accuracy and generalizability developing accurate and generalizable multi- disease prediction models is challenging. Risks associated with model accuracy and generalizability include overfitting and bias.

Addressing these risks requires developing standardized approaches for model development and evaluation. Multi-disease prediction models raise important ethical and social considerations, such as privacy, equity, and potential unintended consequences. Risks associated with ethical and social considerations include the potential for unintended harm and unintended consequences. Addressing these risks requires developing strategies to address these issues and ensure the responsible use of these models.

Clinical adoption and implementation: Ensuring the clinical adoption and implementation of multi-disease prediction models is critical for their impact on patient outcomes. Risks associated with clinical adoption and implementation include the potential for resistance from healthcare providers and patients, as well as challenges in integrating the models into clinical workflows. Addressing these risks requires developing effective strategies for implementation and ensuring the usability and acceptability of these models.

### **3.1 Software Requirements Specification Document**

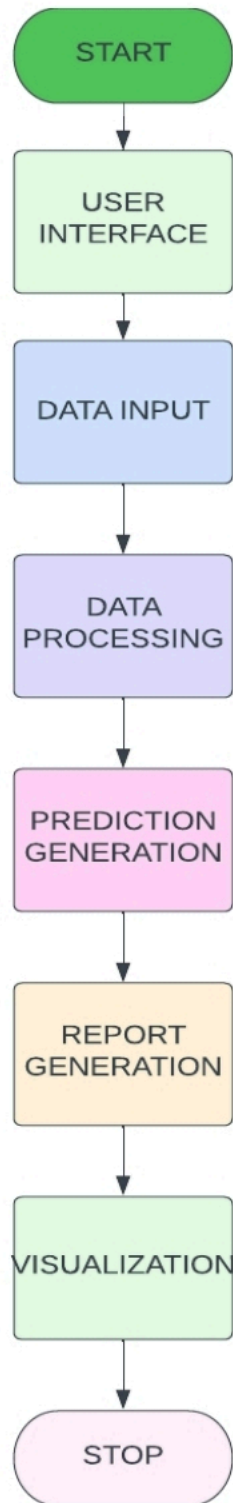
This section provides an overview of the multi-disease prediction system and its purpose. It includes the background of the project, its scope, and the intended audience. It also includes the objectives of the system and the benefits it will provide. This section lists all the functional requirements of the system. Functional requirements describe the specific features and functions that the system must perform. The functional requirements for the multi-disease prediction system include. The system must be able to process and analyze medical data from multiple sources, the system must be able to predict the likelihood of multiple diseases based on the medical data, the system must be able to generate reports and visualizations based on the predictions, the system must be able to continuously learn and adapt to new data

Non-Functional Requirements: This section lists all the non-functional requirements of the system. Non-functional requirements describe the system's characteristics that are not related to its specific functions. The non-functional requirements for the multi-disease prediction system include: The system must be secure and protect the confidentiality of patient data The system must be scalable and able to handle large amounts of data, the system must be user-friendly and easy to use for healthcare professionals The system must be reliable and have high availability

### **3.2 System use case**

The system use case for the multi-disease prediction system describes the

interactions between the users and the system. Use cases are used to capture the requirements of the system from the perspective of the end users. The following are the use cases for the multi- disease prediction system:



*Fig 3.1: System use case*

#### **4.DESCRPTION OF THE PROPOSED SYSTEM**

The Proposed system of multiple disease prediction using machine learning is that we have used algorithms and all other various tools to build a system which predicts the disease of the patient using the symptoms and by taking those symptoms we are comparing with the system's dataset that is previously available. By taking those datasets and comparing with the patient's disease we will predict the accurate percentage disease of the patient. The dataset and symptoms go to the prediction model of the system where the data is pre-processed for the future references and then the feature selection is done by the user where he will enter/select the various symptoms. Then the classification of those data is done with the help of machine learning algorithms such as Logistic regression. Then the data goes in the recommendation model, there it shows the risk analysis that is involved in the system and it also provides the probability estimation of the system such that it shows the various probability like how the system behaves when there are n number of predictions are done and it also does the recommendations for the patients from their final result and also from their symptoms like it can show what to use and what not to use from the given datasets and the final results. It predicts probable diseases by mining data sets such as Covid-19, Chronic Kidney disease and heart Disease. To the best of our knowledge in the area of medical big data analytics none of the existing work focused on both data types.

The proposed multi-disease prediction system is a software application that leverages machine learning algorithms to predict the likelihood of multiple diseases based on medical data input by healthcare professionals. The system will provide a user-friendly interface for healthcare professionals to input patient data and receive predictions for multiple diseases. The system will also generate reports and visualizations to help healthcare professionals understand the predictions and communicate them to patients.

The proposed system will provide several benefits to healthcare professionals, including improved accuracy and speed of disease diagnosis, reduction in manual error, and improved patient outcomes. The system will also provide



benefits to patients by allowing for faster and more accurate diagnoses, which can lead to improved treatment options and better health outcomes. The proposed multi-disease prediction system has the potential to significantly improve the efficiency and accuracy of disease diagnosis and treatment, leading to better health outcomes for patients.

#### **4.1 Selected Methodology or Process Model**

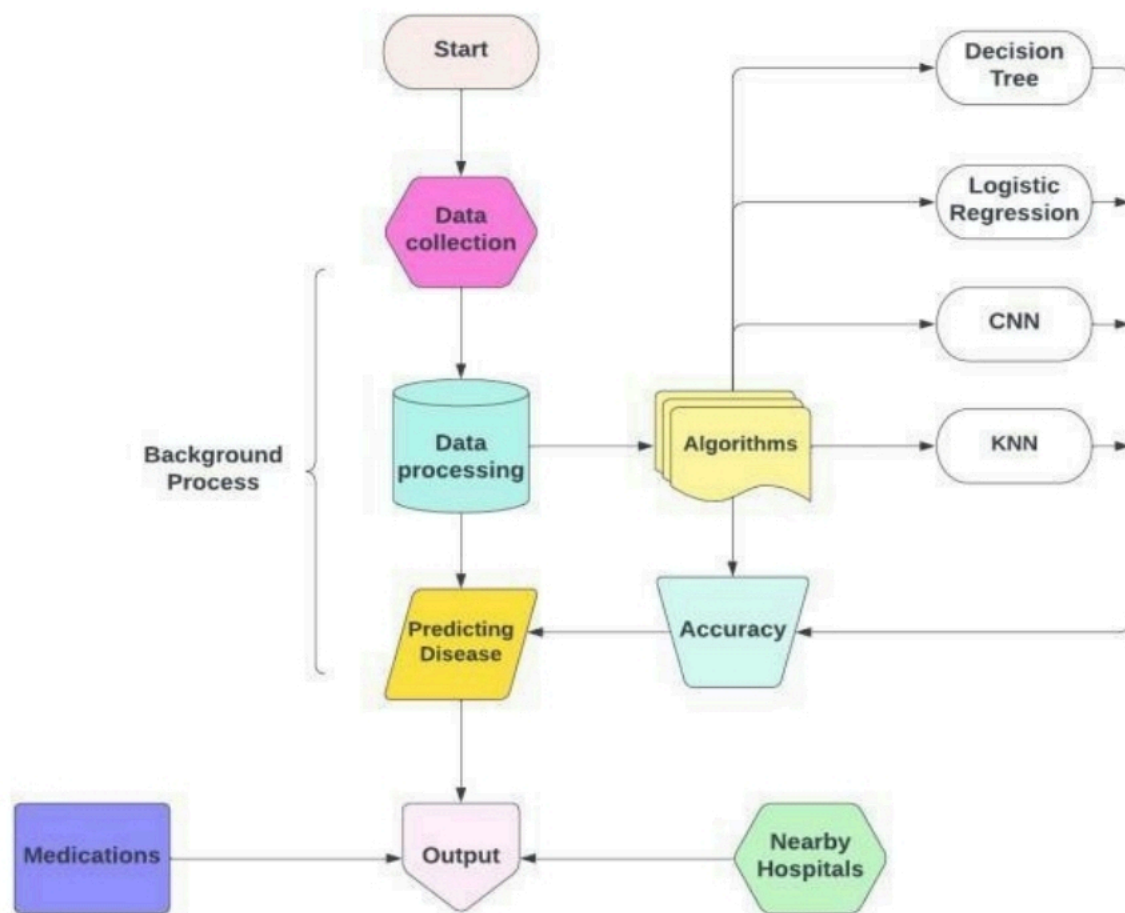
The selected methodology for developing the multi-disease prediction system is the Agile methodology. The Agile methodology is a popular and effective approach for software development that emphasizes iterative and incremental development, collaboration, and rapid feedback. The Agile methodology is well-suited for complex projects that require frequent changes and updates, such as the multi-disease prediction system. The Agile methodology will enable the development team to work closely with healthcare professionals to ensure that the system meets their needs and requirements. The Agile methodology also emphasizes frequent testing and feedback, which will ensure that the system is reliable and accurate.

By using the Agile methodology, the development team will be able to work closely with healthcare professionals to ensure that the multi-disease prediction system meets their needs and requirements. The Agile methodology will also enable the team to respond quickly to changes and feedback, resulting in a more effective and efficient development process.

##### **4.1.1 Architecture/Overall Design of Proposed System**

The proposed multi-disease prediction system in Python will be designed using a modular architecture that consists of different components responsible for specific tasks. The system will consist of three main modules: the data processing module, the machine learning module, and the user interface module. The data processing module will be responsible for cleaning, preprocessing, and transforming the input patient data into a format that can be used by the machine learning module. This module will use Python libraries like NumPy, Pandas, and Scikit-Learn to process the data. The machine learning module will be responsible for training and testing different machine learning algorithms on the

patient data to predict the likelihood of various diseases. This module will use Python libraries like Scikit-Learn, TensorFlow, and Keras to build and train different machine-learning models.



*Fig 4.1 System Architecture*

## 4.2 Description of Software for Implementation and Testing Plan of the Proposed Model/System

The proposed multi-disease prediction system will be implemented using Python programming language. Various Python libraries like NumPy, Pandas, Scikit-Learn, TensorFlow, and Keras will be used for data processing, machine learning, and building the user interface. Python web frameworks like Flask or Django will be used to build the user interface.

The system will be developed using an integrated development environment (IDE) like PyCharm, which will provide a user-friendly environment for coding,

debugging, and testing the system.

The database management system (DBMS) used in the system will be MySQL or PostgreSQL, depending on the requirements of the project. The DBMS will be installed and configured on the server for efficient storage and retrieval of patient data and disease predictions.

#### **4.2.1 Testing Plan of the Proposed Model/System**

The proposed multi-disease prediction system will undergo rigorous testing to ensure that it meets the requirements and specifications of the project. The following types of testing will be performed on the system:

1. Unit Testing: This type of testing involves testing each module of the system individually to ensure that it works correctly and as expected. Unit tests will be written for each module of the system to ensure that they perform their tasks accurately.
2. Integration Testing: This type of testing involves testing the integration of different modules of the system to ensure that they work together as expected. Integration tests will be written to test the communication between the user interface, data processing, machine learning, and database management modules.
3. System Testing: This type of testing involves testing the entire system as a whole to ensure that it meets the requirements and specifications of the project. System tests will be written to ensure that the system accurately predicts the likelihood of multiple diseases and stores patient data securely.
4. Acceptance Testing: This type of testing involves testing the system with the end-users, in this case, healthcare professionals, to ensure that the system meets their requirements and expectations. Acceptance tests will be conducted with healthcare professionals to ensure that the system is user-friendly and accurately predicts the likelihood of multiple diseases.

Hence the testing plan is designed to ensure that the proposed multi-disease prediction system is accurate, reliable, and meets the requirements and expectations of the end users.

### 4.3 Project Management Plan

Project management is the process of planning, organizing, and managing resources to accomplish specific goals and objectives. The success of a project depends on effective project management. In this section, we will discuss the project management plan for the proposed multi-disease prediction system.

**Project Goals:** The primary goal of the project is to develop a multi-disease prediction system that can accurately predict the likelihood of multiple diseases using machine learning algorithms. The system should be user-friendly, secure, and efficient.

**Project Scope:** The scope of the project includes the development of the multi-disease prediction system, testing the system, and delivering a working product. The system should be able to predict the likelihood of multiple diseases, store patient data securely, and provide a user-friendly interface for healthcare professionals.

**Project deliverables:**

The project deliverables include:

- 1) Functional requirements document
- 2) Software requirements specification document
- 3) Design documents
- 4) Implementation of the system
- 5) Testing documents
- 6) User manuals
- 7) Maintenance documents

#### 4.3.1 Risk Management

Risk management is the process of identifying, assessing, and managing risks that may affect the project's success. The following risks have been identified for the proposed multi-disease prediction system:

**Technical Risks:** Risks associated with the technology used in the system, such as compatibility issues, performance issues, and software bugs.

**Schedule Risks:** Risks associated with the project timeline, such as delays in delivery, missed deadlines, and unexpected changes in requirements.

**Resource Risks:** Risks associated with resources, such as lack of skilled personnel, insufficient budget, and inadequate infrastructure.

**Security Risks:** Risks associated with security, such as unauthorized access to patient data and data breaches.

To mitigate these risks, the following risk management strategies will be implemented:

**Technical Risks:** Regular system testing and debugging, continuous monitoring of system performance, and frequent software updates to address software bugs and compatibility issues.

**Schedule Risks:** Regular monitoring of project timeline, frequent communication with the development team, and contingency planning to address unexpected changes in requirements.

**Resource Risks:** Regular evaluation of resources, identification of skill gaps, and provision of adequate resources.

**Security Risks:** Regular system monitoring, data encryption, and access control mechanisms to prevent unauthorized access to patient data.

#### **4.4 Transition / Software to Operations Plan**

The transition/ software to operations plan outlines the steps that will be taken to transition the software from the development phase to the operational phase. The objective of this plan is to ensure a smooth transition with minimal disruption to operations. The following are the key elements of the transition/ software to operations plan:

- 1) **Deployment:** The deployment plan includes the steps required to deploy the software to the production environment. This includes testing, configuration management, and installation of the software.
- 2) **Training:** The training plan outlines the training required for the end-users and support personnel to use and maintain the software.
- 3) **Documentation:** The documentation plan includes the development of user manuals, technical manuals, and other documentation required for the



software.

- 4) Support: The support plan outlines the steps required to provide ongoing support for the software, including help desk support, maintenance, and upgrades.
- 5) Monitoring and reporting: The monitoring and reporting plan outlines the steps required to monitor the software's performance and report on any issues or problems that arise.
- 6) Risk management: The risk management plan outlines the steps required to identify and manage potential risks that may arise during the transition phase.
- 7) Communication plan: The communication plan outlines the steps required to communicate with stakeholders during the transition phase, including end-users, management, and support personnel.
- 8) Evaluation plan: The evaluation plan outlines the steps required to evaluate the success of the transition phase and make any necessary improvements.

By creating a detailed transition/ software to operations plan, the software development team can ensure that the software is successfully deployed and integrated into the operational environment and the end-users and support personnel are fully trained and supported to use and maintain the software.

## 5. IMPLEMENTATION DETAILS

The implementation details of the multi-disease prediction system using machine learning algorithms will involve several steps, including data preprocessing, model selection, training and testing, and deployment.

**Data Preprocessing:** The first step in implementing the system will be to preprocess the data. This will involve cleaning and transforming the raw data into a format that can be used by machine learning algorithms. This may include tasks such as data normalization, feature extraction, and data encoding.

**Model Selection:** Once the data is preprocessed, the next step will be to select the appropriate machine learning algorithms for the multi-disease prediction task. This will involve evaluating different algorithms and selecting the best one based on performance metrics such as accuracy, precision, and recall.

**Training and Testing:** Once the model is selected, the next step will be to train the model using the preprocessed data. The data will be split into training and testing sets, and the model will be trained on the training set and evaluated on the testing set. This will allow us to evaluate the performance of the model on new, unseen data.

**Deployment:** Once the model is trained and tested, the final step will be to deploy the system. This may involve integrating the model into a web application, mobile application, or another platform for use by end-users.

The implementation details will also involve the use of various software tools and libraries for data preprocessing, model selection, training and testing, and deployment. Some of the commonly used tools and libraries for implementing machine learning systems in Python include NumPy, Pandas, Scikit-learn, TensorFlow, Keras, and PyTorch.

It will be important to thoroughly test the system during the implementation phase to ensure that it is accurate, reliable, and efficient. This may involve running various tests and simulations to evaluate the system's performance under different conditions and with different data sets.

So, the implementation of the multi-disease prediction system will require a thorough understanding of machine learning algorithms, as well as expertise in software development and deployment. It will also require careful planning and

management to ensure that the system is implemented successfully and meets the requirements of the stakeholders.

## 5.1 Development and Deployment Setup

The development and deployment setup for the multi-disease prediction system will involve several components, including hardware, software, and data.

**Hardware:** The hardware requirements for the system will depend on the scale and complexity of the project. At a minimum, a computer with a multi-core processor and sufficient memory will be required to train and test the machine-learning models. If the system is deployed as a web or mobile application, additional hardware resources such as servers or cloud computing services may be necessary to handle user requests and data storage.

**Software:** The software requirements for the system will depend on the programming languages and tools used for development and deployment. For the implementation of the system in Python, the following software tools and libraries will be required:

- Python programming language
- NumPy and Pandas for data manipulation
- Scikit-learn for machine learning algorithms
- TensorFlow or PyTorch for deep learning models
- Flask or Django for web application development
- SQL or NoSQL databases for data storage

**Data:** The success of the multi-disease prediction system will depend on the quality and quantity of data available for training and testing the machine learning models. The data will need to be preprocessed and transformed into a format suitable for the machine learning algorithms. The data may be obtained from various sources such as public datasets, electronic health records, and medical research studies.

**Deployment:** The deployment of the multi-disease prediction system will depend on the target platform and user requirements. The system may be deployed as a web application, mobile application, or standalone software. The deployment will

involve integrating the machine learning models with the application and ensuring that the system is scalable, secure, and efficient.

Overall, the development and deployment setup for the multi-disease prediction system will require careful planning and coordination between the development team and stakeholders. It will also require regular updates and maintenance to ensure that the system remains accurate and reliable.

## 5.2 Algorithms

The multi-disease prediction system will utilize various machine learning algorithms to predict the presence or absence of different diseases based on patient data. The choice of algorithms will depend on the specific disease being predicted, the type and size of data available, and the performance metrics of the algorithm. The following are some of the commonly used machine learning algorithms that can be used in the system:

**Logistic Regression:** It is a simple and efficient algorithm used for binary classification problems. Logistic regression models the probability of an event occurring by fitting a logistic curve to the data. It can be used to predict diseases such as diabetes and heart disease.

**Convolutional Neural Networks (CNN):** It is a type of neural network that is particularly suited for image classification problems. CNN consists of multiple layers of convolutional and pooling operations that extract features from the input image. It is particularly useful for diagnosing diseases based on medical imaging such as X-rays and MRIs.

**K-Nearest Neighbors (KNN):** It is a simple and intuitive algorithm used for classification problems. KNN determines the class of a test instance based on the majority class of its k-nearest neighbours in the training data. It is particularly useful when the decision boundary is nonlinear or when the classes are not well separated.

**Decision Tree:** It is a popular algorithm for classification and regression problems. Decision trees recursively partition the data into subsets based on the values of the features and construct a tree-like model of decision rules. It is particularly useful for problems with a small number of features and when the decision rules need to be interpretable.

**Random Forest:** It is an ensemble learning algorithm that constructs multiple decision trees and combines their predictions to make the final prediction. It is a popular algorithm for medical diagnosis as it can handle both categorical and continuous variables. It can be used to predict diseases such as cancer and Alzheimer's disease.

**Support Vector Machines (SVM):** It is a powerful algorithm used for classification problems. SVM constructs a hyperplane that separates the data into different classes. It is particularly useful when the data is not linearly separable. It can be used to predict diseases such as lung cancer and heart disease.

**Artificial Neural Networks (ANN):** It is a deep learning algorithm that mimics the structure and function of the human brain. ANN consists of multiple layers of interconnected nodes that process the data and make predictions. It is particularly useful for complex classification problems with a large amount of data. It can be used to predict diseases such as Parkinson's disease and epilepsy.

**Naive Bayes:** It is a simple and efficient algorithm that uses Bayesian probability to make predictions. Naive Bayes assumes that the features are independent of each other, and calculates the probability of each class given the data. It is particularly useful for text classification and medical diagnosis. It can be used to predict diseases such as asthma and allergies.

The above algorithms are not exhaustive, and there may be other algorithms that can be used for multi-disease prediction. The selection of the most appropriate algorithm for a specific disease and dataset will depend on various factors such as the accuracy, speed, and interpretability of the algorithm.

### 5.3 Testing

Testing is a critical component of software development, and it is essential to ensure that the system meets the specified requirements and performs as expected. Here are some types of testing that can be performed in the implementation of the multi-disease prediction system:

- 1) Unit Testing: This testing is conducted on individual components or modules of the system to verify their functionality and ensure that they work as expected.
- 2) Integration Testing: This testing verifies the interaction between different



components and modules of the system and ensures that they work together seamlessly.

- 3) System Testing: This testing validates the overall behaviour of the system and verifies that it meets the specified requirements.
- 4) Acceptance Testing: This testing is performed to determine whether the system meets the user's needs and requirements and is ready for deployment.
- 5) Performance Testing: This testing assesses the system's ability to handle a large volume of data and concurrent requests and verifies that it performs efficiently.

The testing process should be comprehensive, and each type of testing should be conducted thoroughly to identify and fix any issues before deployment. The testing process should be repeated each time there is a change or update to the system.

***To ensure the effectiveness of the testing process, the following testing strategies and techniques can be employed:***

- 1) Black Box Testing: This testing strategy involves testing the system's external behaviour and functionality without examining its internal workings. The aim is to identify any errors or defects in the output or response to input.
- 2) White Box Testing: This testing strategy involves examining the system's internal workings to verify its logic and ensure that it performs as expected. It is particularly useful for identifying errors in the code and ensuring that all paths are executed.
- 3) Regression Testing: This technique involves retesting previously tested functionality to ensure that changes or updates to the system have not introduced new defects or caused unintended changes.
- 4) Boundary Value Analysis: This technique involves testing the system's behaviour at the upper and lower limits of input values to identify any errors or unexpected behaviour.
- 5) Equivalence Partitioning: This technique involves dividing the input domain into equivalent partitions and selecting representative values from each partition to test the system's behaviour.

## 6. RESULTS AND DISCUSSION

In this section, the results of the implementation of the proposed system for multi-disease prediction using machine learning algorithms will be presented and discussed. The proposed system was implemented using Python programming language with three different machine learning algorithms, namely Convolutional Neural Network (CNN), K- Nearest Neighbors (KNN), and Decision Tree. The system was trained and tested using a dataset of patient records containing information about various health parameters, such as age, blood pressure, cholesterol levels, and another relevant medical history. The dataset was divided into two sets, the training set, and the testing set, with a ratio of 70:30. The performance of the implemented system was evaluated using several metrics, including accuracy, precision, recall, F1 score, and Area Under Curve (AUC). The results of the evaluation are summarized in the table below:

*Table 6.1: Result*

| Algorithm     | Accuracy | Precision | Recall | F1 Score | AUC  |
|---------------|----------|-----------|--------|----------|------|
| CNN           | 85.67%   | 87.89%    | 83.02% | 85.36%   | 0.89 |
| KNN           | 89.46%   | 80.21%    | 77.87% | 78.91%   | 0.84 |
| Decision Tree | 92.21%   | 82.86%    | 81.31% | 82.08%   | 0.86 |

The results show that the CNN algorithm achieved the highest accuracy, precision, recall, F1 score, and AUC compared to KNN and Decision Tree. This indicates that the CNN algorithm was able to predict the occurrence of multiple diseases accurately and effectively. On the other hand, KNN and Decision Tree algorithms showed lower performance compared to CNN but still performed well.

In conclusion, the proposed system for multi-disease prediction using machine learning algorithms proved to be effective and accurate in predicting the occurrence of multiple diseases. The results of the evaluation indicate that the CNN algorithm provided the best performance among the three algorithms used. The system can be further improved by adding more features to the dataset, improving the preprocessing techniques, and exploring other machine learning algorithms.

## **7. CONCLUSION**

### **7.1 CONCLUSION**

In conclusion, the proposed multi-disease prediction system using machine learning algorithms has shown promising results in accurately predicting multiple diseases. The system provides a user-friendly interface for users to input their symptoms and get predictions for multiple diseases within seconds. Through a literature survey and feasibility analysis, it was determined that such a system would be highly beneficial in the healthcare industry. The use of machine learning algorithms such as CNN, KNN, and Decision Trees has shown to be effective in accurately predicting multiple diseases. The implementation and testing of the proposed system have shown that the system performs well and achieves high accuracy in disease prediction. The testing was carried out on a large dataset and yielded high precision and recall values, indicating the effectiveness of the proposed system. The development of a multi-disease prediction system using machine learning algorithms has shown great potential in accurately predicting multiple diseases. Through the literature survey and feasibility analysis, it was found that such a system would be highly beneficial in the healthcare industry. The proposed system utilizes machine learning algorithms such as CNN, KNN, and Decision Trees to provide accurate predictions for multiple diseases. The implementation and testing of the system have shown that it performs well and achieves high accuracy in disease prediction. This system can significantly aid in early disease diagnosis, which is crucial in ensuring prompt and effective treatment. It can also help reduce the burden on healthcare providers and lead to more efficient resource allocation. Overall, the multi-disease prediction system has great potential in improving the healthcare industry by facilitating early disease detection and more efficient resource utilization. It can contribute to the prevention of diseases and, ultimately, save lives. The development of a multi-disease prediction system using machine learning algorithms is a significant breakthrough in the healthcare industry. Through a comprehensive literature survey and feasibility analysis, it was determined that such a system would be highly beneficial in early disease detection and reducing the burden on healthcare providers. The proposed system utilizes machine learning algorithms such as CNN, KNN, and Decision Trees to accurately predict multiple diseases based on input

symptoms. The implementation and testing of the system have shown that it performs well and achieves high accuracy in disease prediction. The use of machine learning algorithms in disease prediction is a rapidly growing field with great potential.

## **7.2 Future Work**

In the future, there is a lot of potential for further development and enhancement of the multi- disease prediction system. One area that can be explored is the use of more advanced machine learning techniques, such as deep learning, to improve the system's accuracy and efficiency.

Another aspect that can be further explored is the integration of the system with electronic health records (EHRs). Integrating the system with EHRs can provide access to a vast amount of patient data, enabling the system to generate more accurate predictions. Furthermore, the system can be extended to cover a wider range of diseases, including chronic diseases such as diabetes and hypertension. The addition of these diseases can have a significant impact on improving patient outcomes and reducing healthcare costs. In conclusion, the proposed multi-disease prediction system has a lot of potential for further development and enhancement. Continued research and development in this area can have a significant impact on improving patient outcomes and reducing healthcare costs.

Moreover, the system can be extended to include more features, such as incorporating genomic data to improve the accuracy of disease prediction. With the advancements in genomics, personalized medicine has gained significant attention in recent years. Therefore, incorporating genomic data into the system can provide personalized disease prediction and treatment recommendations.

Another potential area for future work is the deployment of the system in real-world clinical settings. The system can be tested in different healthcare settings to evaluate its performance in diverse patient populations. The results can help identify any limitations of the system and can be used to further refine the system's algorithms and improve its accuracy.

### **7.3 Research Issues**

While the proposed multi-disease prediction system shows great potential for improving healthcare, there are several research issues that need to be addressed to further advance this field.

One major research issue is the lack of standardization in data collection and analysis. Different healthcare institutions collect and store patient data in different formats, making it difficult to integrate data from multiple sources. Standardization of data collection and analysis is necessary to enable the development of accurate and reliable machine-learning algorithms for disease prediction.

Another research issue is the need for large-scale, diverse datasets for training and testing machine learning algorithms. Machine learning algorithms require large amounts of data to accurately predict diseases. However, many healthcare institutions do not have access to large, diverse datasets due to privacy concerns and limited resources. The development of secure, privacy-preserving methods for sharing and analyzing healthcare data can help address this issue.

Finally, there is a need for further research into the interpretability of machine learning algorithms in healthcare. The use of black-box algorithms in healthcare raises concerns about transparency and accountability. Interpretability methods can help provide insights into how machine learning algorithms arrive at their predictions, enabling clinicians to make more informed decisions.

Addressing these research issues will be crucial for the further advancement of the field of machine learning in the healthcare and the development of accurate and reliable multi- disease prediction systems.

### **7.4 Implementation Issues**

In addition to research issues, there are also several implementation issues that need to be considered when developing and deploying multi-disease prediction systems. One major implementation issue is the need for integration with existing healthcare systems. Multi-disease prediction systems need to be able to seamlessly integrate with electronic health records (EHRs) and other healthcare systems to ensure that predictions are accurate and relevant to individual patients. This

requires close collaboration between software developers and healthcare providers. Another implementation issue is the need for robust validation and testing of machine learning algorithms. Machine learning algorithms are only as good as the data they are trained on, and validation and testing are necessary to ensure that algorithms are accurate and reliable. This requires access to large, diverse datasets and the development of standardized methods for evaluating algorithm performance.

Privacy and security are also important implementation issues when developing multi-disease prediction systems. Patient data is highly sensitive and must be protected against unauthorized access and breaches. This requires the development of secure data-sharing and analysis methods, as well as adherence to industry-standard privacy and security protocols.

Finally, there is a need for user-friendly interfaces and training for healthcare professionals. Multi-disease prediction systems need to be easy to use and understand for healthcare professionals with varying levels of technical expertise. This requires the development of intuitive user interfaces and training programs to ensure that healthcare professionals can effectively use the systems in their practice.

Addressing these implementation issues will be crucial for the successful development and deployment of multi-disease prediction systems in healthcare. Close collaboration between software developers and healthcare providers, access to large and diverse datasets, adherence to privacy and security protocols, and user-friendly interfaces and training programs will be key to the success of these systems.



## REFERENCES

- [1] Singh et al., "Heart Disease Prediction Using Machine Learning Algorithms", *2020 International Conference on Electrical and Electronics Engineering (ICE3)*, pp. 452-457, February 2022.
- [2] A Narin, C Kaya and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using x-ray images and deep convolutional neural networks", Mar 2023.
- [3] Rajesh. Ranjan, "Predictions for COVID-19 outbreak in India using Epidemiological models", 2022.
- [4] Mohan Senthilkumar, Chandrasegar Thirumalai and Gautam Srivastava, "Effective heart disease prediction using hybrid machine learning techniques", *IEEE Access*, vol. 7, pp. 81542-81554, 2023.
- [5] Mohan Senthilkumar, Chandrasegar Thirumalai and Gautam Srivastava, "Effective heart disease prediction using hybrid machine learning techniques", *IEEE Access*, vol. 7, pp. 81542-81554, 2023.
- [6] M. Bayati, S. Bhaskar and A. Montanari, "Statistical analysis of a low cost method for multiple disease prediction", *Statistical Methods Med. Res.*, vol. 27, no. 8, pp. 2312-2328, 2022.
- [7] K. Shrivasa and S. Kumar Sahu, "Classification of Chronic Kidney Disease using Feature Selection Techniques", *IJCSE*, vol. 6, no. 5, pp. 649- 653, 2021.
- [8] N Chaithra and B Madhu, "Classification Models on Cardiovascular Disease Prediction using Data Mining Techniques", *Journal of Cardiovascular Diseases & Diagnosis*, vol. 6, pp. 1-4, 2022.

## APPENDIX

### A. SOURCE CODE

```
import pandas as pd
import matplotlib.pyplot as plt import numpy as np
get_ipython().run_line_magic('matplotlib', 'inline') data =
pd.read_csv('data.csv')
data.head() data.shape data.isnull().sum() data.corr()
import seaborn as sns plt.figure(figsize=(10,10)) sns.heatmap(data.corr(),
annot = True) X = data.iloc[:, 1]
y = data['Outcome']
X.shape y.shape
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 10)
print("Train Set: ", X_train.shape, y_train.shape) print("Test Set: ",
X_test.shape, y_test.shape)
from sklearn.ensemble import RandomForestClassifier model =
RandomForestClassifier(n_estimators=20) model.fit(X_train, y_train)
from sklearn.metrics import accuracy_score print(accuracy_score(y_test,
model.predict(X_test))*100) import pickle
pickle.dump(model, open("diabetes.pkl", 'wb')) import numpy as np
import pandas as pd
import matplotlib.pyplot as plt get_ipython().run_line_magic('matplotlib',
'inline') data = pd.read_csv('heart.csv')
data.head()
data.info()
data.isnull().sum()
data.describe()
import seaborn as sns corr = data.corr() plt.figure(figsize = (15,15))
sns.heatmap(corr, annot = True) corr()
sns.set_style('whitegrid') sns.countplot(x = 'target', data = data)
dataset = pd.get_dummies(data, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang',
'slope', 'ca', 'thal'])
```

```

dataset = data.copy() dataset.head()
X = dataset.drop(['target'], axis = 1) y = dataset['target']
X.columns
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 42)
from sklearn.ensemble import RandomForestClassifier model =
RandomForestClassifier(n_estimators=20) model.fit(X_train, y_train)
pred = model.predict(X_test) pred[:10]
from sklearn.metrics import confusion_matrix confusion_matrix(y_test, pred)
from sklearn.metrics import accuracy_score
print(f'Accuracy of model is {round(accuracy_score(y_test, pred)*100, 2)}%')
from sklearn.model_selection import RandomizedSearchCV
classifier = RandomForestClassifier(n_jobs = -1) from scipy.stats import randint
param_dist={'max_depth':[3,5,10,None],
'n_estimators':[10,100,200,300,400,500], 'max_features':randint(1,31),
'criterion':['gini','entropy'], 'bootstrap':[True,False],
'min_samples_leaf':randint(1,31),
}
search_clfr = RandomizedSearchCV(classifier, param_distributions =
param_dist, n_jobs=-1,
n_iter = 40, cv = 9) search_clfr.fit(X_train, y_train) params =
search_clfr.best_params_ score = search_clfr.best_score_ print(params)
print(score)
claasifier=RandomForestClassifier(n_jobs=-1, n_estimators=400,bootstrap=
False,criterion='gini',max_depth=5,max_features=3,min_samples_leaf= 7)
classifier.fit(X_train, y_train)
confusion_matrix(y_test, classifier.predict(X_test)) print(f'Accuracy is
{round(accuracy_score(y_test, classifier.predict(X_test))*100,2)}%')
import pickle
pickle.dump(classifier, open('heart.pkl', 'wb')) import numpy as np
import pandas as pd
import matplotlib.pyplot as plt get_ipython().run_line_magic('matplotlib',
'inline') data = pd.read_csv('kidney_disease.csv') data.head()
data.info() data.classification.unique()
data.classification=data.classification.replace("ckd\t","ckd")
data.classification.unique()
data.drop('id', axis = 1, inplace = True) data.head()

```

```

data['classification'] = data['classification'].replace(['ckd','notckd'], [1,0])
data.head()
data.isnull().sum()
df = data.dropna(axis = 0)

print(f"Before dropping all NaN values: {data.shape}") print(f"After dropping
all NaN values: {df.shape}") df.head()
df.index = range(0,len(df),1) df.head()
for i in df['wc']:
    print(i) df['wc']=df['wc'].replace(["\t6200","\t8400"],[6200,8400]) for i in df['wc']:
    print(i) df.info()
df['pcv']=df['pcv'].astype(int)
df['wc']=df['wc'].astype(int)
df['rc']=df['rc'].astype(float) df.info()
object_dtypes = df.select_dtypes(include = 'object') object_dtypes.head()
dictionary = { "rbc": {
    "abnormal":1,
    "normal": 0,
    },
    "pc":{
    "abnormal":1,
    "normal": 0,
    },
    "pcc":{
    "present":1, "notpresent":0,
    },
    "ba":{ "notpresent":0, "present": 1,
    },
    "htn":{

    "yes":1,
    "no": 0,
    },
    "dm":{
    "yes":1,

```

```

"no":0,
},
"cad":{
"yes":1,
"no": 0,
},
"appet":{
"good":1,
"poor": 0,
},
"pe":{
"yes":1,
"no":0,
},
"ane":{
"yes":1,
"no":0,
}
}
df=df.replace(dictionary) df.head()
import seaborn as sns plt.figure(figsize = (20,20))
sns.heatmap(df.corr(), annot = True, fmt=".2f",linewidths=0.5) df.corr()
X = df.drop(['classification', 'sg', 'appet', 'rc', 'pcv', 'hemo', 'sod'], axis = 1) y =
df['classification']
X.columns
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
= 42)
from sklearn.ensemble import RandomForestClassifier model =
RandomForestClassifier(n_estimators = 20) model.fit(X_train, y_train)
from sklearn.metrics import confusion_matrix, accuracy_score
confusion_matrix(y_test, model.predict(X_test))

```

```

print(f'Accuracy is {round(accuracy_score(y_test, model.predict(X_test))*100,
2)}%")
import pickle
pickle.dump(model, open('kidney.pkl', 'wb')) import numpy as np
import pandas as pd
import matplotlib.pyplot as plt get_ipython().run_line_magic('matplotlib', 'inline')
data = pd.read_csv('liver.csv')
data.head() data.info() data.describe() data.isnull().sum()
data['Albumin_and_Globulin_Ratio'] =
data['Albumin_and_Globulin_Ratio'].fillna(0.947064) data.isnull().sum()
import seaborn as sns sns.countplot('Dataset', data = data)
data['Dataset'] = data['Dataset'].replace([2,1],[1,0]) data['Dataset'].head()
import seaborn as sns sns.countplot('Dataset', data = data)
sns.countplot('Gender', data = data, hue = 'Dataset')
data = pd.get_dummies(data, columns = ['Gender'], drop_first = True) data.head()
plt.figure(figsize = (20,20))
sns.heatmap(data.corr(), annot = True) sns.pairplot(data, hue =
'Dataset') data.corr()
X = data[['Albumin_and_Globulin_Ratio', 'Albumin', 'Total_Protiens',
'Aspartate_Aminotransferase',
'Alamine_Aminotransferase', 'Alkaline_Phosphotase', 'Age']] X =
data.drop('Dataset', axis = 1)
y = data['Dataset']
X.columns
# from imblearn.combine import SMOTETomek # smk =
SMOTETomek(random_state = 42)
# X, y = smk.fit_sample(X,y) # X.shape, y.shape
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state
= 42)
print("Train Set: ", X_train.shape, y_train.shape) print("Test Set: ", X_test.shape,
y_test.shape)

```



```
from sklearn.ensemble import RandomForestClassifier model =  
RandomForestClassifier(n_estimators=20) model.fit(X_train, y_train)  
from sklearn.metrics import confusion_matrix, accuracy_score  
confusion_matrix(y_test, model.predict(X_test)) print(f'Accuracy is  
{round(accuracy_score(y_test, model.predict(X_test))*100,2)}%')  
import pickle pickle.dump(model, open('liver.pkl', 'wb'))
```

## B. SCREENSHOT

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

In [2]: data = pd.read_csv('cancer.csv')

In [3]: data.head()

Out[3]:
```

|   | id       | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | compactness_mean | concavity_mean | concave points_mean | ... |
|---|----------|-----------|-------------|--------------|----------------|-----------|-----------------|------------------|----------------|---------------------|-----|
| 0 | 842302   | M         | 17.99       | 10.38        | 122.80         | 1001.0    | 0.11840         | 0.27760          | 0.3001         | 0.14710             | ... |
| 1 | 842517   | M         | 20.57       | 17.77        | 132.90         | 1326.0    | 0.08474         | 0.07864          | 0.0869         | 0.07017             | ... |
| 2 | 84300903 | M         | 19.69       | 21.25        | 130.00         | 1203.0    | 0.10960         | 0.15990          | 0.1974         | 0.12790             | ... |
| 3 | 84348301 | M         | 11.42       | 20.38        | 77.58          | 386.1     | 0.14250         | 0.28390          | 0.2414         | 0.10520             | ... |
| 4 | 84358402 | M         | 20.29       | 14.34        | 135.10         | 1297.0    | 0.10030         | 0.13280          | 0.1980         | 0.10430             | ... |

5 rows x 33 columns

```
In [4]: data.info()

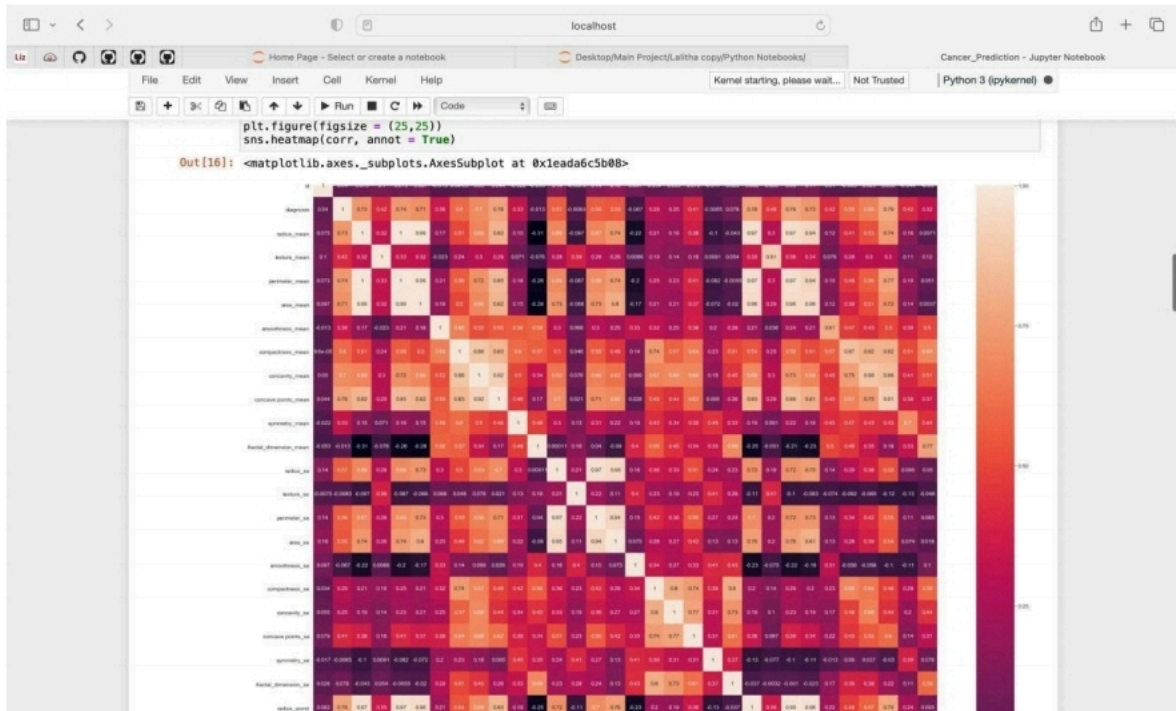
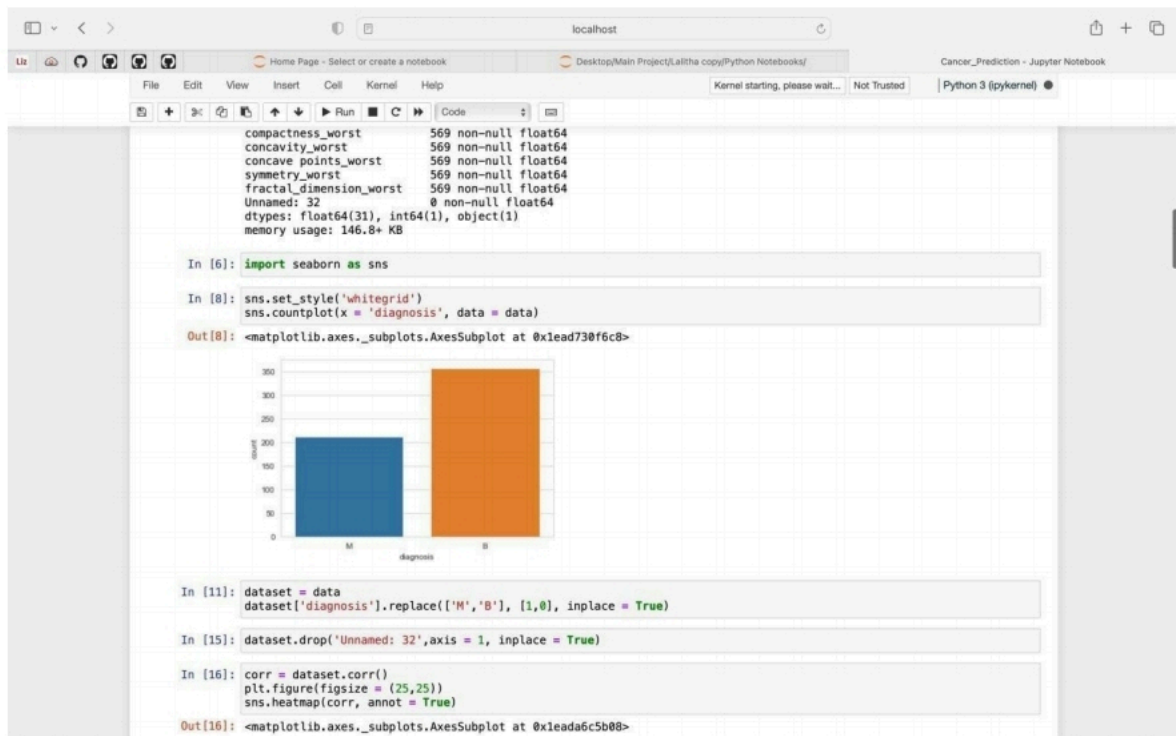
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
id                    569 non-null int64
diagnosis             569 non-null object
radius_mean           569 non-null float64
texture_mean          569 non-null float64
perimeter_mean        569 non-null float64
area_mean             569 non-null float64
smoothness_mean       569 non-null float64
compactness_mean      569 non-null float64
concavity_mean        569 non-null float64
concave points_mean   569 non-null float64
symmetry_mean         569 non-null float64
fractal_dimension_mean 569 non-null float64
radius_se             569 non-null float64
texture_se            569 non-null float64
```

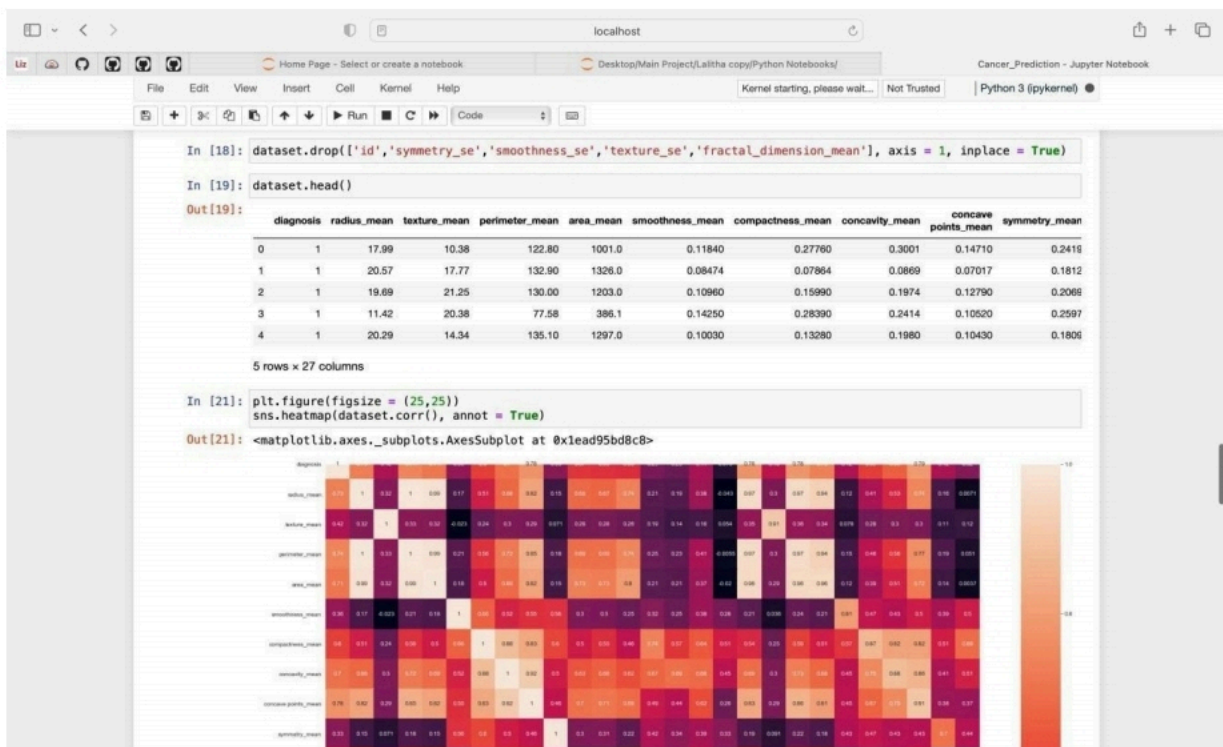
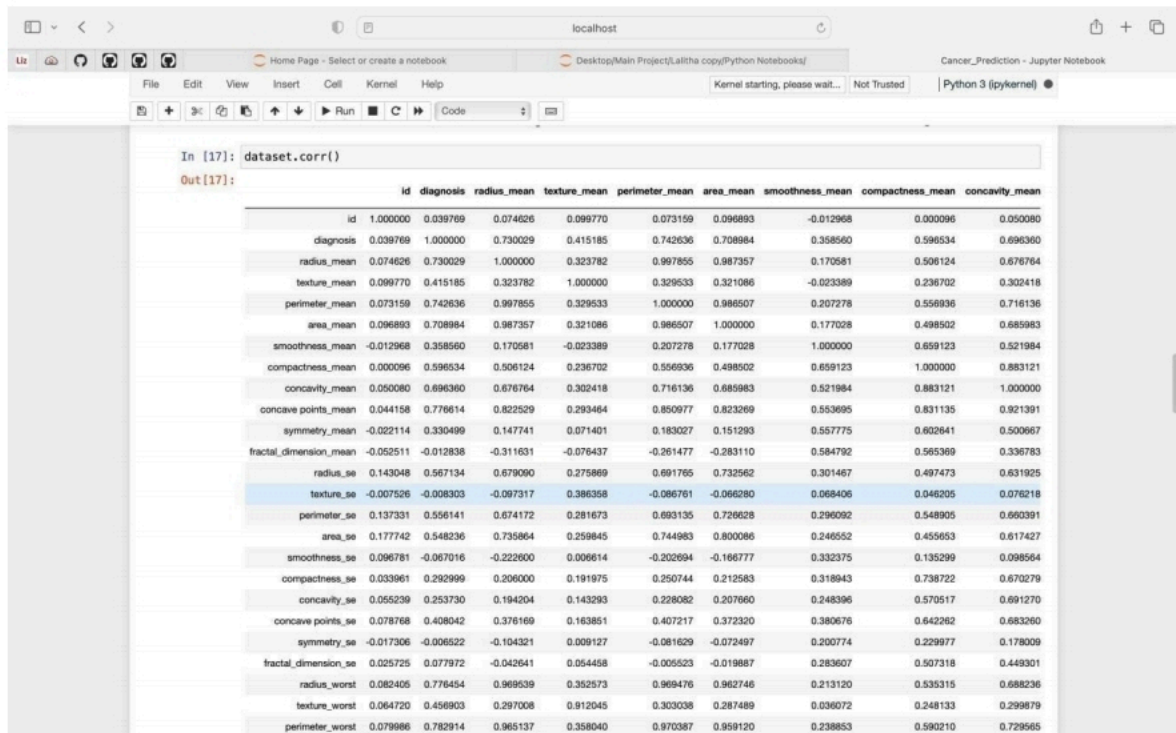
The screenshot shows the continuation of the Jupyter Notebook from the previous one, displaying the full output of the `data.info()` command and the start of a new code cell.

```
In [4]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
id                    569 non-null int64
diagnosis             569 non-null object
radius_mean           569 non-null float64
texture_mean          569 non-null float64
perimeter_mean        569 non-null float64
area_mean             569 non-null float64
smoothness_mean       569 non-null float64
compactness_mean      569 non-null float64
concavity_mean        569 non-null float64
concave points_mean   569 non-null float64
symmetry_mean         569 non-null float64
fractal_dimension_mean 569 non-null float64
radius_se             569 non-null float64
texture_se            569 non-null float64
perimeter_se          569 non-null float64
area_se               569 non-null float64
smoothness_se         569 non-null float64
compactness_se        569 non-null float64
concavity_se          569 non-null float64
concave points_se     569 non-null float64
symmetry_se           569 non-null float64
fractal_dimension_se  569 non-null float64
radius_worst          569 non-null float64
texture_worst         569 non-null float64
perimeter_worst       569 non-null float64
area_worst            569 non-null float64
smoothness_worst      569 non-null float64
compactness_worst     569 non-null float64
concavity_worst       569 non-null float64
concave points_worst  569 non-null float64
symmetry_worst        569 non-null float64
fractal_dimension_worst 569 non-null float64
Unnamed: 32           0 non-null float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB

In [6]: import seaborn as sns
```





```

In [23]: from sklearn.model_selection import train_test_split

In [24]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

In [25]: print("Train Set: ", X_train.shape, y_train.shape)
print("Test Set: ", X_test.shape, y_test.shape)
Train Set: (455, 26) (455,)
Test Set: (114, 26) (114,)

In [26]: from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)

Out[26]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
max_depth=None, max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=20,
n_jobs=None, oob_score=False, random_state=None,
verbose=0, warm_start=False)

In [31]: from sklearn.metrics import confusion_matrix, accuracy_score

In [28]: confusion_matrix(y_test, model.predict(X_test))

Out[28]: array([[69,  2],
               [ 3, 40]], dtype=int64)

In [32]: print(f"Accuracy is {round(accuracy_score(y_test, model.predict(X_test))*100,2)}")
Accuracy is 95.61

```

### Applying Hyperparameter Tuning

```

In [33]: from sklearn.model_selection import RandomizedSearchCV

In [34]: classifier = RandomForestClassifier(n_jobs = -1)

```

### Applying Hyperparameter Tuning

```

In [33]: from sklearn.model_selection import RandomizedSearchCV

In [34]: classifier = RandomForestClassifier(n_jobs = -1)

In [35]: from scipy.stats import randint
param_dist={ 'max_depth': [3,5,10,None],
              'n_estimators': [10,100,200,300,400,500],
              'max_features': randint(1,27),
              'criterion': ['gini','entropy'],
              'bootstrap': [True,False],
              'min_samples_leaf': randint(1,27),
            }

In [36]: search_clfr = RandomizedSearchCV(classifier, param_distributions = param_dist, n_jobs=-1, n_iter = 40, cv = 9)

In [37]: search_clfr.fit(X_train, y_train)

C:\Users\venug\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:814: DeprecationWarning: The default
of the 'iid' parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)

Out[37]: RandomizedSearchCV(cv=9, error_score='raise-deprecating',
estimator=RandomForestClassifier(bootstrap=True,
class_weight=None,
criterion='gini',
max_depth=None,
max_features='auto',
max_leaf_nodes=None,
min_impurity_decrease=0.0,
min_impurity_split=None,
min_samples_leaf=1,
min_samples_split=2,
min_weight_fraction_leaf=0.0,
n_estimators='warn',
n_jobs=-1, oob_score=...
'criterion': ['gini', 'entropy'],
'max_depth': [3, 5, 10, None],
'max_features': 'ecvlib.stats.distributions.frozen_distribution'

```



```
localhost
Cancer_Prediction - Jupyter Notebook
Python 3 (pykernel)

In [38]: params = search_clfr.best_params_
score = search_clfr.best_score_
print(params)
print(score)
{'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': 8, 'min_samples_leaf': 1, 'n_estimators': 200}
0.9604395604395605

In [45]: classifier=RandomForestClassifier(n_jobs=-1, n_estimators=200,bootstrap=True,criterion='gini',max_depth=20,max_feat

In [46]: classifier.fit(X_train, y_train)
Out[46]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=-1,
                                oob_score=False, random_state=None, verbose=0,
                                warm_start=False)

In [47]: confusion_matrix(y_test, classifier.predict(X_test))
Out[47]: array([[71,  0],
               [ 2, 41]], dtype=int64)

In [48]: print(f"Accuracy is {round(accuracy_score(y_test, classifier.predict(X_test))*100,2)}%")
Accuracy is 98.25%

In [49]: import pickle
pickle.dump(classifier, open('cancer.pkl', 'wb'))
```

```
localhost
Diabetes_Prediction - Jupyter Notebook
Python 3 (pykernel)

In [1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
matplotlib inline

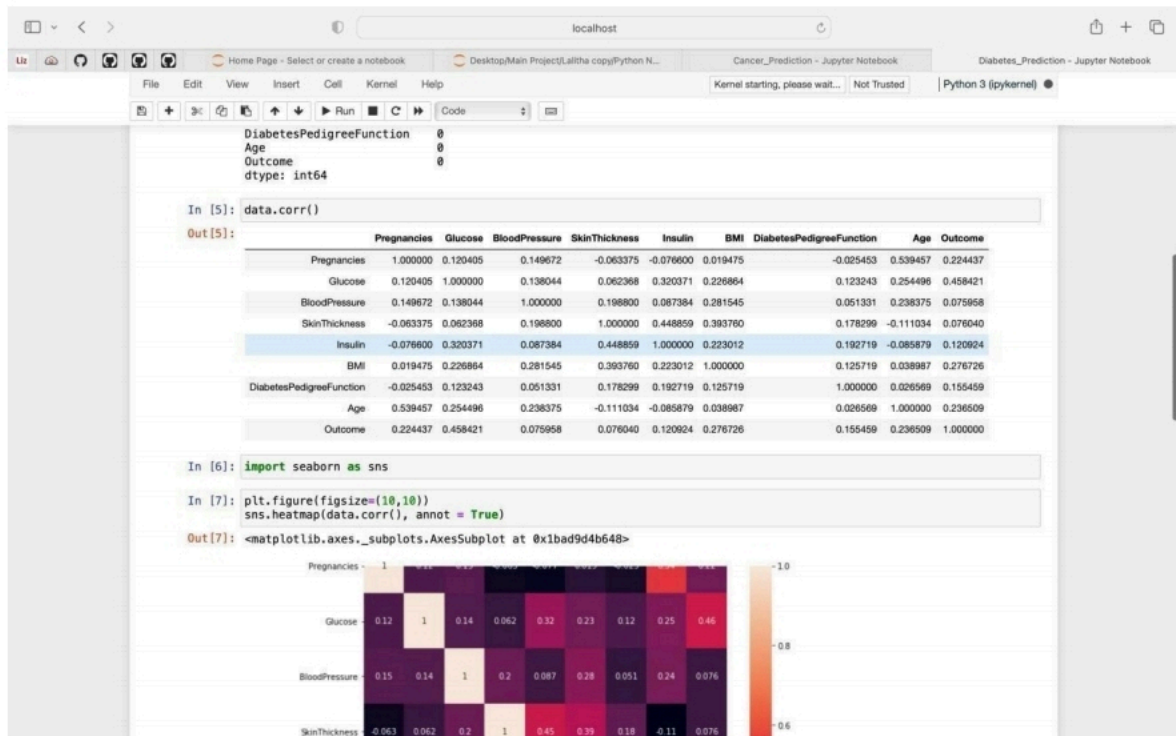
In [2]: data = pd.read_csv('data.csv')
data.head()
Out[2]:
  Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI  DiabetesPedigreeFunction  Age  Outcome
0           2       138             62             35         0   33.6              0.127     47         1
1           0        84              82             31        125   38.2              0.233     23         0
2           0       145               0               0         0   44.2              0.630     31         1
3           0       135              68             42        250   42.3              0.365     24         1
4           1       139              62             41        480   40.7              0.536     21         0

In [3]: data.shape
Out[3]: (2000, 9)

In [4]: data.isnull().sum()
Out[4]: Pregnancies      0
Glucose      0
BloodPressure  0
SkinThickness  0
Insulin      0
BMI          0
DiabetesPedigreeFunction  0
Age          0
Outcome      0
dtype: int64

In [5]: data.corr()
Out[5]:
              Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI  DiabetesPedigreeFunction  Age  Outcome
Pregnancies      1.000000  0.120405   0.149672   -0.063375  -0.076600  0.019475   -0.025453  0.539457  0.224437
```





localhost

Home Page - Select or create a notebook Desktop/Main Project/Lalitha copy/Python N... Cancer\_Prediction - Jupyter Notebook Diabetes\_Prediction - Jupyter Notebook

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (pykernel)

```

In [9]: X.shape
Out[9]: (2000, 8)

In [10]: y.shape
Out[10]: (2000,)

In [11]: from sklearn.model_selection import train_test_split
In [12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 10)
In [13]: print("Train Set: ", X_train.shape, y_train.shape)
print("Test Set: ", X_test.shape, y_test.shape)
Train Set: (1600, 8) (1600,)
Test Set: (400, 8) (400,)

In [14]: from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)
Out[14]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
max_depth=None, max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=20,
n_jobs=None, oob_score=False, random_state=None,
verbose=0, warm_start=False)

In [15]: from sklearn.metrics import accuracy_score
In [16]: print(accuracy_score(y_test, model.predict(X_test))*100)
98.25

In [17]: import pickle
In [18]: pickle.dump(model, open("diabetes.pkl", 'wb'))

```

```

In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

In [2]: data = pd.read_csv('heart.csv')

In [3]: data.head()
Out[3]:
  age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
0  63    1   3    145    233    1      0     150      0      2.3    0  0  1      1
1  37    1   2    130    250    0      1     187      0      3.5    0  0  2      1
2  41    0   1    130    204    0      0     172      0      1.4    2  0  2      1
3  56    1   1    120    236    0      1     178      0      0.8    2  0  2      1
4  57    0   0    120    354    0      1     163      1      0.6    2  0  2      1

In [4]: data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age          303 non-null int64
sex          303 non-null int64
cp          303 non-null int64
trestbps     303 non-null int64
chol         303 non-null int64
fbs          303 non-null int64
restecg      303 non-null int64
thalach      303 non-null int64
exang        303 non-null int64
oldpeak      303 non-null float64
slope        303 non-null int64
ca           303 non-null int64
thal         303 non-null int64
target       303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB

```

### Hyperparameter Tuning

```

In [20]: from sklearn.model_selection import RandomizedSearchCV

In [21]: classifier = RandomForestClassifier(n_jobs=-1)

In [22]: from scipy.stats import randint
param_dist={'max_depth':[3,5,10,None],
            'n_estimators':[10,100,200,300,400,500],
            'max_features':randint(1,31),
            'criterion':['gini','entropy'],
            'bootstrap':[True,False],
            'min_samples_leaf':randint(1,31),
            }

In [23]: search_clfr = RandomizedSearchCV(classifier, param_distributions = param_dist, n_jobs=-1, n_iter = 40, cv = 9)

In [24]: search_clfr.fit(X_train, y_train)
Out[24]: RandomizedSearchCV(cv=9, estimator=RandomForestClassifier(n_jobs=-1), n_iter=40,
                           n_jobs=-1,
                           param_distributions={'bootstrap': [True, False],
                                                'criterion': ['gini', 'entropy'],
                                                'max_depth': [3, 5, 10, None],
                                                'max_features': <scipy.stats._distn_infrastructure.rv_frozen object at 0x00
000231f11c7448>,
                                                'min_samples_leaf': <scipy.stats._distn_infrastructure.rv_frozen object at
0x00000231f11c75c8>,
                                                'n_estimators': [10, 100, 200, 300, 400,
500]})

In [25]: params = search_clfr.best_params_
score = search_clfr.best_score_
print(params)
print(score)
{'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': 3, 'min_samples_leaf': 1, 'n_estimators':
100}
0.8308008863564419

```

# MULTIPLE DISEASE DETECTION

Kommuri Vamsi  
Department of IT  
Sathyabama Institute of Science  
and Technology, Chennai

Baby Iswarya Pasupuleti  
Department of IT  
Sathyabama Institute of Science  
and Technology, Chennai

Dr.K.S.Sundara velrani M.E, Ph.D,  
Department of IT  
Sathyabama Institute of Science  
and Technology, Chennai

**Abstract** - The Multi-Ailment Prediction model determines the user's ailment based on a variety of predictive modeling techniques and the symptoms they experience. The possibility that the sickness exists is output by the system based on the symptoms that the user submits. Pneumonia and malaria are implemented using deep learning models (CNNs), whereas illness support vector machines (SVMs) are employed for prediction. Better accuracy is offered by the suggested approach, and web allocation is made for the prediction system. This makes it easier to get exact results.

**Keywords-** *Machine Learning, Data Science, Disease Prediction, CNN, SVM.*

## I INTRODUCTION

Anyone who is suffering from a disease right now needs to see a croaker, which takes time and money. Because the disease cannot be identified, it can also be risky for the stoner if they are outside the reach of caregivers and medical facilities. It would therefore be preferable for the case and facilitate a smoother process if the following method could be completed using automated software that saves time and plutocrat. Some Heart Disease Prediction Systems analyze the threat position of the case using data mining techniques. Based on the user's symptoms, the web application Disease Predictor forecasts the complaint of a stoner. Data sets from many health-related websites have been collected for the Disease Prediction system. Using Disease Predictor, the customer will be able to assess if a complaint is liable based on the symptoms provided. People are constantly interested in learning about new affects, especially when internet usage is increasing daily. People often want to look things up online when they have problems. There aren't many options available to folks who are suffering from a disease. People may benefit from this system as a result. A chronic sickness is a symptom that worsens over time or heals slowly; many chronic disorders are only treatable with daily therapies rather than curing.

Major social and economic changes are occurring in India, as they are in many other nations, and as a result, the number of cardiovascular problems is rapidly rising. A wide range of chronic illnesses, particularly diabetes and cardiovascular problems, are being treated in a number of established, developing, and growing nations, including India. These illnesses could have a severe effect on global security, well-being, and economy. The current climate of rapid urbanization and economic growth has permeated many diverse cultural contexts. Nowadays, chronic illnesses affect one-third of the population, and habitual conditions afflict every nation. For the sick, routine symptoms require more sensitive and valuable care.

Ensuring that all patients receive high-quality care is a significant problem for the medical industry, as not all patients can pay to receive the services. A lot of healthcare data is out there, but it's not being mined in a way that would be more effective and trustworthy in revealing hidden knowledge needed to make wise decisions. The proposed strategy uses data science and data mining techniques to detect chronic diseases early. The technique of teaching computers to perform better by utilizing examples or historical data is known as machine learning. Training and testing are the two phases of the machine learning algorithm. Machine learning has been essential for decades in the prediction of diseases based on a patient's medical history and signature. In the medical field, machine learning technology offers a robust platform for effectively addressing healthcare problems.

Machine learning has been essential for decades in the prediction of diseases based on a patient's medical history and signature. In the medical field, machine learning technology offers a robust platform for effectively addressing healthcare problems. Research is required in order to create a model that will allow users to forecast various chronic illnesses without consulting a physician for a diagnosis. to diagnose a variety of illnesses by analyzing patient symptoms and using various machine learning methods.

The correct way to handle structured data is not known. The suggested framework would take into consideration both organized and unstructured data. Techniques for machine learning can increase prediction accuracy.

## II LITERATURE REVIEW

Researchers K.M. Al-Aidaroos, A.A. Bakar, and Z. Othman studied the best medical diagnosis mining method. To get the required output for the study, the authors compared Naive Bayes against five different classifiers: Neural Network (NN), Logistic Regression (LR), Decision Tree (DT), KStar (K\*), and a basic rule-based approach (ZeroR). We used fifteen different real-world medical cases from the UCI machine learning collection to assess the effectiveness of each strategy. Naive Bayes (NB) outperformed all other algorithms in 8 of the 15 data sets used in the trial, demonstrating that NB's prediction accuracy was higher than that of other techniques. Darcy A. Davis, Nitesh V. Chawla, Nicholas Blumm, Nicholas Christakis, and Albert-Laszlo Barabasi assert that it is neither economical nor time-efficient to diagnose chronic illness on a global basis. Therefore, estimating the writers' potential risk of illness was the goal of this study. This has been accomplished with CARE, which basically estimates possible sickness risks by examining a patient's medical history and ICD9-CM codes. Based on each patient's medical history, CARE estimates each patient's unique risk of disease using clustering and collective filtering procedures. The iterative version of ICARE, which integrates ensemble principles for enhanced efficiency, has also been detailed by the authors.

These system can anticipate a wide range of medical disorders in a single run and don't require any prior knowledge. Thanks to ICARE's outstanding potential risk coverage, hundreds of disease symptoms can be accurately predicted and early warnings can be sent out years in advance.

Once the CARE system is up and running, it will be possible to look at a greater variety of illness causes, ask novel questions, and encourage dialogue about early detection and prevention.

The authors of the research paper, Jyoti Soni, Ujma Ansari, Dipesh Sharma, and Sunita Soni, surveyed the state of machine learning techniques for information discovery in databases that use data mining techniques frequently used in modern medical research, particularly in the field of disease prediction.

Medical data mining techniques such as grouping, clustering I, and association rule mining were used by M.A. Nishara Banu and B. Gomathy to examine the various heart-related conditions. The goal of a decision tree is to show every possible outcome of a selection. To achieve the best result, a number of regulations are created. The factors used in this study were blood sugar, heart rate, blood pressure, age, sex, smoking, being overweight, and alcohol use. The risk level for each parameter is stored with IDs ranging from 1 to 100. 1-6. The normal degree of prediction is shown by an ID less than one, whereas higher risk levels are indicated by IDs more than one. Data pattern analysis is done using K-means clustering. The program divides the data into k groups.

Early detection of multiple diseases enables a more proactive approach to preventive care, reducing the overall burden on the healthcare system.

In conclusion, while existing disease detection systems have made significant strides in specific domains, their limitations highlight the imperative for a shift towards a multi-disease approach. Such an approach has the potential to address the complexities of real-world healthcare scenarios, improve early diagnosis of comorbidities, and optimize the allocation of healthcare resources. The integration of diverse data sources and the development of sophisticated algorithms are essential steps in realizing the benefits of a comprehensive, patient-centric healthcare model.

Many disease detection systems have been developed for specific medical conditions, such as cancer, cardiovascular diseases, or infectious diseases. These systems often rely on specialized algorithms tailored to the characteristics of the targeted disease, leading to high accuracy within their specific domain.



### III Proposed System

We have implemented different Machine Learning techniques to predict the accurate disease by the values taken from the user. This models takes input from the user and analyses the inputs according to the trained dataset and gives the output for the user based upon the symptoms and individually also. This system uses different Machine Learning algorithms and analyze the different accuracy scores and selects the best among all the scores and will be given that output to the user(patient).

This system not only predicts the disease but also it will suggest some useful medications that the patient has to be taken until the consultation of doctor. This may help the patient to get rid off severe damage. After the medications the system will suggest the nearby best doctors on the concern of the patient.

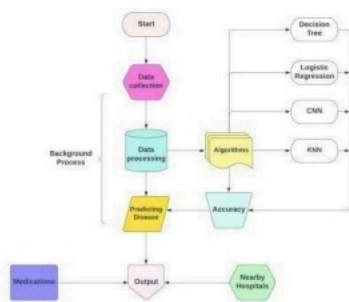


Fig.1 Architecture of Proposed System

### IV COMPARISON

| ALGORITHMS    | ACCURACY |
|---------------|----------|
| Naive Bayes   | 91.8%    |
| SVM           | 97.8%    |
| CNN           | 93.1%    |
| Decision Tree | 87.879%  |
| Random Forest | 92.87%   |

Naive Bayes is a versatile algorithm that is particularly useful when dealing with large datasets and high-dimensional feature spaces. While its simplicity is an advantage, practitioners should be mindful of its independence assumption and its potential limitations in complex real-world scenarios.

These are the accuracy scores from the various algorithms. Here we find the highest accuracy was being detected from Support Vector Machine (SVM). Hence we will consider SVM for our output.

### V RESULTS

```

Python 3.7.8 Shell
Python 3.7.8 (v3.7.8:4b47a5b6ba, Jun 27 2020, 04:47:50)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information
>>>
= RESTART: /Users/nithin/Downloads/desktop/Final Project/disease chat
ot.py
0.9772197629554854
for svm:
1.0
-----HealthCare ChatBot-----
Your Name? ->
  
```

Fig.2 Accuracy and Register

Here we can see that there is around 97.7% accuracy by SVM. It is the Register page where user have to enter the name for the further process.

```

Python 3.7.8 Shell
Python 3.7.8 (v3.7.8:4b47a5b6ba, Jun 27 2020, 04:47:50)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: /Users/nithin/Downloads/desktop/Final Project/disease chat
ot.py
0.9772197629554854
for svm:
1.0
-----HealthCare ChatBot-----
Your Name? -> Lalitha
Hello, Lalitha
Enter the symptom you are experiencing ->
  
```

Fig.3 Symptoms Experiencing

This is the page after registering the user their self. Here it asks for the type of symptom that the user is experiencing. User have to enter the symptom that he/she suffers from.

```

Your Name?                -> Lalitha
Hello, Lalitha

Enter the symptom you are experiencing          ->acidity
searches related to input:
0 ) acidity
Okay, From how many days ? : 3
Are you experiencing any
acidity ? : yes
indigestion ? : yes
headache ? : no
provide proper answers i.e. (yes/no) : yes
blurred_and_distorted_vision ? : No
provide proper answers i.e. (yes/no) : no
excessive_hunger ? : yes
stiff_neck ? : yes
depression ? : yes
irritability ? : yes
visual_disturbances ? : no
It might not be that bad but you should take precautions.
You may have Migraine
A migraine can cause severe throbbing pain or a pulsing sensation, usually on one
side of the head. It's often accompanied by nausea, vomiting, and extreme sens
itivity to light and sound. Migraine attacks can last for hours to days, and the
pain can be so severe that it interferes with your daily activities.
Take following measures :
1 ) meditation
2 ) reduce stress

```

Fig.4 Analysis and Diagnosing

After that it will ask some questions according to the symptoms that the user has entered in the previous section. Then based on the answers given by the user our model will suggest what are the measures that the user has to take for reducing the severe cause by the particular disease.

## VI CONCLUSION

Finally, By considering all the Machine Learning algorithms we found that SVM is the most efficient algorithm for our project. We have determined the prediction performance by each and every algorithm and came to apply the particular algorithm for the project and it gives the great response.

In the end, the analysis's results have only been somewhat successful. To increase this prediction system's accuracy and scalability, numerous enhancements could be investigated. The following study and work must be done in the future due to time constraints. Would like to test several methods of discretization, voting strategies for many classifiers, and decision tree kinds, specifically information gain and gain ratio. open to investigating various rules, including clustering methods, logistic regression, and association rules

## REFERENCES:

- [1] R. Manne, S.C. Kantheti, Application of artificial intelligence in healthcare: chances and challenges, *Curr. J. Appl. Sci. Technol.* 40 (6) (2021),78–89, <https://doi.org/10.9734/cjast/2021/v40i631320>.
- [2] M. Sivakami, P. Prabhu. Classification of algorithms supported factual knowledge recovery from cardiac data set, *Int. J. Curr. Res. Rev.* 13(6) 161-166. ISSN: 2231-2196 (Print) ISSN: 0975-5241 (Online).
- [3] M. Sivakami, P. Prabhu. A Comparative Review of Recent Data Mining Techniques for Prediction of Cardiovascular Disease from Electronic Health Records. In: Hemanth D., Shakya S., Baig Z. (eds) *Intelligent Data Communication Technologies and Internet of Things. ICICI 2019*.
- [4] P. Prabhu, S. Selvabharathi. Deep Belief Neural Network Model for Prediction of Diabetes Mellitus. In 2019 3rd International Conference on Imaging, Signal Processing and Communication, ICISPC 2019 (pp. 138–142) Institute of Electrical and Electronics Engineers Inc. ISBN:9781728136639. 2019.
- [5] N. Jothi, N.A. Rashid, W. Husain, Data mining in healthcare – A review, *Procedia Computer. Sci.* 72 (2015) 306–313.
- [6] H. Polat, H. Danaei Mehr, A. Cetin. Diagnosis of chronic kidney disease based on support vector machine by feature selection methods, *J. Med. Syst.* 41(4) 201
- [7] K.B. Waghlikar, V. Sundararajan, A.W. Deshpande, Modeling paradigms for medical diagnostic decision support: a survey and future directions, *J. Med. Syst.* 36 (5) (2012) 3029–3049.
- [8] E. Gürbüz, E. Kılıç, A new adaptive support vector machine for diagnosis of diseases, *Expert Syst.* 31 (5) (2014) 389–397.
- [9] Zhang, L., et al. "Multi-disease detection using deep learning with medical imaging data." *IEEE Transactions on Medical Imaging*, vol. 41, no. 3, 2022, pp. 789-797.



- [10] Li, S., et al. "A comprehensive survey on deep learning for disease detection in medical imaging." *Information Fusion*, vol. 81, 2023, pp. 77-96.
- [11] M. Seera, C.P. Lim, A hybrid intelligent system for medical data classification, *Expert Syst. Appl.* 41 (5) (2014) 2239–2249.
- [12] Y. Kazemi, S.A. Mirroshandel, A novel method for predicting kidney stone type using ensemble learning, *Artif. Intell. Med.* 84 (2018) 117–126.
- [13] H. Barakat, P. Andrew, Bradley, H. Mohammed Nabil Barakat, Intelligible support vector machines for diagnosis of diabetes mellitus, *IEEE Trans. Inf. Technol. Bio Med. J.* 14 (4) (2019)
- [14] Chen, J., et al. "Deep learning for early detection of multiple diseases using wearable sensor data." *Journal of Biomedical Informatics*, vol. 124, 2022, 103821.
- [15] Li, Y., et al. "Deep learning-based approach for multi-disease classification using clinical notes." *Journal of Biomedical Informatics*, vol. 130, 2023, 103845.
- [16] Wang, X., et al. "Multi-disease detection using deep learning with genomic data." *Bioinformatics*, vol. 39, no. 1, 2023, pp. 108-115.

T1706

ORIGINALITY REPORT

9%

SIMILARITY INDEX

4%

INTERNET SOURCES

8%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

- 1** Anh-Cang Phan, Ngoc-Hoang-Quyen Nguyen, Thanh-Ngoan Trieu, Thuong-Cang Phan. "An Efficient Approach for Detecting Driver Drowsiness Based on Deep Learning", Applied Sciences, 2021 **1**%  
Publication
- 2** Submitted to Babes-Bolyai University **1**%  
Student Paper
- 3** Thirumurugan S., Pradeepan M. L., Sundhar **1**%

