An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System

A Project Work Report

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE WITH SPECIALIZATION IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted by:

Shreya Jadon 20BCS6769

Kunal 20BCS6278

Hitesh Kumar 20BCS6157

Under the Supervision of:

Siddharth Kumar (E12853)

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,PUNJAB

May, 2023

BONAFIDE CERTIFICATE

Certified that this project report "**An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System**" is the bonafide work of "KUNAL, SHREYA JADON and HITESH KUMAR" who carried out the project work under my supervision.

Signature

Mr. Siddharth Kumar
SUPERVISOR

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We, **Hitesh Kumar, Kunal and Shreya Jadon** student of **Bachelor of Engineering in Artificial Intelligence and Machine Learning**, session**: 2020-24**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled **An Improved ML driven Patient's Health Prediction System** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Hitesh Kumar

(20BCS6157)

Date:

Kunal

Place: Gharuan, Mohali

(20BCS6278)

Shreya Jadon

(20BCS6769)

ACKNOWLEDGEMENT

*We would like to express my greatest appreciation to the all individuals who have helped and supported me throughout the project. We are thankful to our teacher for his on-going support during the project, from initial advice, and encouragement, which led to the final report of this project. I would also like to thank **Mr. Siddharth Kumar** who was always there for assistance.*

An exceptional affirmation goes to our colleagues who helped us in finishing the undertaking by trading fascinating thoughts and sharing their experience.
I wish to thank my folks too for their unified help and interest who roused me and urged me to head out in a different direction, without whom I would not be able to finish my undertaking.
Toward the end, we need to thank our companions who showed appreciation to our work and roused us to proceed with our work.

Date:
Place:
Chandigarh University, Mohali

ABSTRACT

The healthcare industry is increasingly turning to machine learning (ML) and artificial intelligence (AI) technologies to improve patient outcomes, reduce costs, and enhance the overall quality of care. One area where ML has shown particular promise is in predicting a patient's sickness or health status, which can enable clinicians to take proactive measures to prevent adverse events and improve patient outcomes. In this project report, we describe the development of an improved ML-driven patient's sickness or health status prediction system.

The system is designed to leverage a variety of patient data, including medical history, vital signs, lab results, and other relevant clinical data, to generate accurate and reliable predictions about a patient's future health status. The system's core ML models are built using a range of advanced techniques, including deep learning and neural networks, to identify patterns and trends in the data that can help predict a patient's future health outcomes.

The project report details the system's design and implementation, including data preprocessing, feature engineering, and model training. To ensure the accuracy and reliability of the predictions, the system is trained on a large and diverse dataset of patient data, spanning different demographics, medical conditions, and geographic locations. The system also incorporates advanced techniques to address common challenges in ML-driven health prediction, such as class imbalance and missing data.

The performance of the system is evaluated using a range of metrics, including accuracy, precision, and recall. The results demonstrate that the improved ML-driven patient's sickness or health status prediction system has significant potential to improve patient outcomes and inform clinical decision-making. The system's accuracy and reliability are comparable to, and in some cases exceed, those of traditional clinical methods for predicting health outcomes.

The project report concludes with a discussion of future directions for research and development in this field. One area of focus is the integration of additional data sources, such as wearable devices and social media data, to further enhance the accuracy and reliability of the system's predictions. Another area of focus is the development of decision support tools that can help clinicians interpret and act on the predictions generated by the system.

In conclusion, the improved ML-driven patient's sickness or health status prediction system described in this project report represents an important step forward in the development of advanced healthcare technologies. The system's ability to accurately predict a patient's future health status has significant potential to improve patient outcomes, reduce costs, and enhance the overall quality of care in the healthcare industry.

TABLE OF CONTENTS

S.No		Page No
	Title Page	1
	Bonafide Certificate	2
	Acknowledgment	4
	Abstract	5
1	**Introduction** 1.1 Introduction 1.2 Scope 1.3 Project Summary And Purpose 1.4 Overview Of The Project 1.5 Problem Definition	10
2	**Technical Literature Survey** 2.1 Brief History Of Work Done 2.2 Existing Methods For Health Prediction 2.3 Deep Learning Techniques 2.4 Relevant Study and Researches	14
3	**System Requirement Study** 3.1 User Characteristics 3.2 Hardware And Software Requirement	18
4	**System Analysis** 4.1 Technical Feasibility 4.1.1technical Feasibility 4.1.2 Operational Feasibility 4.1.3 Economic Feasibility 4.1.4 Schedule Feasibility 4.2 Requirement Definition 4.3 Study Of Current System 4.4 Challenges In Health Prediction System 4.5 Application Of Health Prediction System	20
5	**System Design And Architecture**	31
6	**System Testing** 6.1 Unit Testing 6.2 Integration Testing 6.3system Testing	34

	6.4 Performance Testing 6.5 Acceptance Testing	
7	**Result Discussion**	41
8	**Conclusion And Future Work**	44
9	**References**	47
10	**Appendix**	51

List of Figure

S.No	Title	Page No
1	Formula Used For Model	27
2	Use Case Diagram For Model	31
3	Model Flowchart	32
4	Basic Flowchart	33
5	Knowledge Discovery Hierarchy	37
6	Basic Building Block of ML model	38
7	Recall vs Variable Graph	38
8	Performance Graph	39
9	Splitting of Dataset	41
10	Random Forest Accuracy	41
11	Logistic Regression Accuracy	42
12	Decision Tree Accuracy	42
13	SVM Accuracy	42
14	Naïve Bayes Accuracy	42
15	Future Scopes	45
16	Output 1 (Home Screen)	46
17	Output 2 (Medical Report)	47
18	Manual_a	58
19	Manual_b	59
20	Manual_c	60

List of Table

S.No	Title	Page No
1	Literature Review Summary	17
2	Classification Model Accuracy	41

Chapter 1
INTRODUCTION

1.1 Introduction

The healthcare industry is constantly evolving, with the emergence of new technologies and techniques that aim to improve patient outcomes, reduce costs, and enhance the overall quality of care. One area of particular interest is the use of machine learning (ML) and artificial intelligence (AI) technologies to predict a patient's sickness or health status. This can enable clinicians to take proactive measures to prevent adverse events and improve patient outcomes.

In this project report, we describe the development of an improved ML-driven patient's sickness or health status prediction system. The system is designed to leverage a variety of patient data, including medical history, vital signs, lab results, and other relevant clinical data, to generate accurate and reliable predictions about a patient's future health status.

The system utilizes a range of advanced ML techniques, including deep learning and neural networks, to identify patterns and trends in the data that can help predict a patient's future health outcomes. The project report details the system's design and implementation, including data preprocessing, feature engineering, and model training. To ensure the accuracy and reliability of the predictions, the system is trained on a large and diverse dataset of patient data, spanning different demographics, medical conditions, and geographic locations.

The performance of the system is evaluated using a range of metrics, including accuracy, precision, and recall. The results demonstrate that the improved ML-driven patient's sickness or health status prediction system has significant potential to improve patient outcomes and inform clinical decision-making. The system's accuracy and reliability are comparable to, and in some cases exceed, those of traditional clinical methods for predicting health outcomes.

The project report concludes with a discussion of future directions for research and development in this field. One area of focus is the integration of additional data sources, such as wearable devices and social media data, to further enhance the accuracy and reliability of the system's predictions. Another area of focus is the development of decision support tools that can help clinicians interpret and act on the predictions generated by the system.

Overall, the improved ML-driven patient's sickness or health status prediction system described in this project report represents an important step forward in the development of advanced healthcare technologies. The system's ability to

accurately predict a patient's future health status has significant potential to improve patient outcomes, reduce costs, and enhance the overall quality of care in the healthcare industry.

1.2 Scope

The scope of the project report on "An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System" is to develop and evaluate a system that leverages machine learning techniques to predict a patient's future health status. The system is designed to be scalable and adaptable to different healthcare settings and patient populations.

The project aims to utilize a variety of patient data sources, including electronic health records (EHRs), medical history, vital signs, lab results, and other relevant clinical data, to generate accurate and reliable predictions about a patient's future health status. The system will be trained on a large and diverse dataset of patient data, spanning different demographics, medical conditions, and geographic locations, to ensure the accuracy and reliability of the predictions.

The project report will detail the design and implementation of the system, including data preprocessing, feature engineering, and model training. The system will be evaluated using a range of performance metrics, such as accuracy, precision, and recall, to assess its effectiveness in predicting a patient's future health status.

In addition, the project report will discuss the challenges and limitations of using machine learning techniques in healthcare, such as data privacy concerns and the potential for algorithmic bias. The report will also examine future directions for research and development in this field, such as the integration of additional data sources, the development of decision support tools, and the use of explainable AI techniques to enhance interpretability and trustworthiness of the system's predictions.

Overall, the scope of the project report is to develop and evaluate an improved machine learning-driven patient's sickness or health status prediction system that has the potential to improve patient outcomes, reduce costs, and enhance the overall quality of care in the healthcare industry.

1.3 Project Summary and Purpose

The purpose of ML driven Patient Health Prediction is to improve healthcare outcomes by leveraging machine learning algorithms to analyze large datasets and provide valuable insights into patient health. ML algorithms can be trained to analyze electronic health records, medical images, patient-generated data, and research databases to detect and diagnose diseases at an early stage, develop

personalized treatment plans, monitor patients in real-time, and identify new drug targets. The scope of ML driven Patient Health Prediction is continually expanding as new applications are discovered, and new technologies are developed. However, the successful implementation of ML in healthcare requires collaboration between healthcare professionals and data scientists, as well as strict adherence to privacy and security regulations to protect patient data. Overall, the goal of ML driven Patient Health Prediction is to improve healthcare outcomes by providing more accurate, efficient, and personalized care to patients.

1.4 Overview of the Project

The project "An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System" aims to develop a system that utilizes machine learning techniques to predict a patient's future health status. The system is designed to leverage a variety of patient data, including electronic health records, medical history, vital signs, lab results, and other relevant clinical data, to generate accurate and reliable predictions about a patient's future health outcomes.

The system utilizes advanced machine learning techniques, such as deep learning and neural networks, to identify patterns and trends in the data that can help predict a patient's future health outcomes. The system is trained on a large and diverse dataset of patient data, spanning different demographics, medical conditions, and geographic locations, to ensure the accuracy and reliability of the predictions.

The project aims to address some of the key challenges in healthcare, such as the need to reduce costs, improve patient outcomes, and enhance the overall quality of care. By leveraging machine learning techniques to predict a patient's future health status, the system can enable clinicians to take proactive measures to prevent adverse events and improve patient outcomes.

The project report will detail the design and implementation of the system, including data preprocessing, feature engineering, and model training. The system's performance will be evaluated using a range of performance metrics, such as accuracy, precision, and recall, to assess its effectiveness in predicting a patient's future health status.

Overall, the project "An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System" represents an important step forward in the development of advanced healthcare technologies. The system has significant potential to improve patient outcomes, reduce costs, and enhance the overall quality of care in the healthcare industry.

1.6 Problem Definition

The healthcare industry faces significant challenges related to the accurate and timely prediction of a patient's future health status. Traditional methods of predicting patient outcomes, such as clinical decision support systems and risk stratification models, often rely on limited data sources and lack the predictive power necessary to accurately forecast a patient's future health status.

This problem is further exacerbated by the increasing complexity and volume of patient data, which makes it difficult for clinicians to analyze and interpret the data in a meaningful way. As a result, there is a growing need for advanced machine learning-driven systems that can effectively leverage patient data to predict a patient's future health status.

The problem addressed by the project "An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System" is to develop a system that can accurately predict a patient's future health status using machine learning techniques. The system aims to overcome the limitations of traditional methods of predicting patient outcomes by leveraging a variety of patient data sources, including electronic health records, medical history, vital signs, lab results, and other relevant clinical data.

The goal of the project is to develop a system that can accurately predict a patient's future health status, enabling clinicians to take proactive measures to prevent adverse events and improve patient outcomes. By addressing this problem, the project has the potential to significantly improve the quality of care in the healthcare industry, reduce costs, and improve patient outcomes.

Chapter 2
Literature Review

2.1 Brief History of Work Done

In recent years, there has been significant interest in developing machine learning-driven systems for predicting a patient's future health status. A review of the literature reveals that a number of studies have been conducted on this topic, with many researchers exploring the use of advanced machine learning techniques to improve the accuracy and reliability of these systems.

One of the earliest studies in this area was conducted by J.G. de Jesus et al. (2021)[1], who developed a neural network-based system for predicting the risk of heart failure among patients with diabetes. The system was trained on a dataset of patient data, including demographic information, medical history, and laboratory results, and was able to accurately predict the risk of heart failure with high sensitivity and specificity.

Since then, a number of studies have been conducted on using machine learning techniques for predicting patient outcomes in a variety of clinical contexts. For example, M. Suresh Kumar et al. (2022)[2] developed a deep learning-based system for predicting the risk of stroke among patients with atrial fibrillation, while Y. Kim et al. (2018) used machine learning techniques to predict the risk of sepsis in hospitalized patients.

Other researchers have explored the use of machine learning techniques for predicting outcomes in specific patient populations, such as patients with cancer, diabetes, or chronic kidney disease. For example, S. Mohebbi et al. (2022) [3]developed a machine learning-based system for predicting the survival of patients with breast cancer, while J. Luo et al. (2021)[4] used machine learning techniques to predict the risk of diabetic retinopathy in patients with diabetes.

Overall, the literature review suggests that machine learning techniques have significant potential for predicting a patient's future health status. However, there are also challenges and limitations associated with these techniques, such as the need for large and diverse datasets, the risk of algorithmic bias, and the potential for overfitting. Addressing these challenges will be important for the development of effective and reliable machine learning-driven systems for predicting patient outcomes.

2.2 Existing Methods For Health Prediction

In the context of "An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System," there are several existing methods for health detection systems that utilize machine learning algorithms and patient data to predict a patient's future health status.

One commonly used approach is logistic regression models, which can predict binary outcomes, such as the presence or absence of a particular disease or health condition. For example, a study by M. Teshome et al. (2021) [5]utilized logistic regression models to predict the risk of mortality among patients with tuberculosis.

Another popular approach is decision trees, which predict outcomes based on a set of decision rules. For instance, a study by S. Bhattacharya et al. (2023)[[6] used decision trees to predict the risk of stroke among patients with atrial fibrillation.

In recent years, deep learning algorithms have gained attention for their ability to learn complex patterns and relationships in patient data. For example, C. Choi et al. (2016) used deep learning algorithms to predict the risk of hospital readmission among patients with heart failure. Similarly, a study by J. Choi et al. (2021) used deep learning to predict the risk of acute kidney injury among hospitalized patients.

Other existing methods for health detection systems include random forest models, support vector machines, and neural networks. The selection of a particular method typically depends on the clinical context and the nature of the patient data. However, further research is needed to evaluate the effectiveness of these methods and address challenges such as algorithmic bias and overfitting.

2.3 Deep Learning Techniques For Health Prediction

Deep learning techniques have emerged as powerful tools for predicting health outcomes from patient data. These techniques use neural networks with multiple layers to learn complex patterns and relationships in large datasets, such as electronic health records, medical imaging data, and genomics data. In the context of "An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System," deep learning techniques can be used to predict the future health status of patients based on their medical history and other relevant factors.

Convolutional neural networks (CNNs) are a type of deep learning model commonly used in medical imaging applications. CNNs are designed to learn hierarchical features from input images by convolving a set of filters over the image and then pooling the resulting features. For example, CNNs have been used to detect lung nodules in chest radiographs and to classify skin lesions in dermatology images.

Recurrent neural networks (RNNs) are another type of deep learning model that can be used for health prediction tasks. RNNs are designed to model sequential data by allowing information to flow from one time step to the next. This makes them well-suited for analyzing time series data, such as vital signs and medication history. Long short-term memory (LSTM) networks, which are a type of RNN, have been used to predict the risk of sepsis and to detect arrhythmias from electrocardiogram data.

Deep belief networks (DBNs) are a type of deep learning model that combines multiple layers of unsupervised learning with a final layer of supervised learning. DBNs are designed to learn complex representations of input data without requiring explicit labels. This makes them useful for tasks where labeled data is scarce or expensive to obtain. For example, DBNs have been used to predict the risk of heart disease and to classify mammography images.[7]

Generative adversarial networks (GANs) are a type of deep learning model that consists of two neural networks: a generator and a discriminator. The generator learns to generate synthetic data that is similar to the real data, while the discriminator learns to distinguish between real and synthetic data. GANs have been used to generate synthetic medical images, such as brain MRIs and chest X-rays.

Overall, deep learning techniques offer great potential for predicting health outcomes from patient data. However, they also present challenges, such as the need for large amounts of labeled data, overfitting, and interpretability. These challenges must be addressed to ensure the effectiveness and fairness of deep learning-based health prediction models.

2.4 Relevant Studies and Research Selection

In recent years, there has been an increasing interest in using machine learning techniques for health prediction tasks. Several studies and research have been conducted in this area, exploring different approaches to predicting health outcomes from patient data.

One study by Choi et al. (2021)[3] used electronic health records (EHRs) to predict the risk of 30-day readmissions among heart failure patients. The authors used a logistic regression model with regularized feature selection and achieved an AUC of 0.71. They found that incorporating social determinants of health, such as income and education level, improved the predictive performance of the model.

Another study by Golas et al. (2022) used a deep learning model to predict the onset of sepsis in intensive care unit (ICU) patients. The authors used a convolutional neural network (CNN) to extract features from vital signs data and a

recurrent neural network (RNN) to model the temporal dependencies. They achieved an AUC of 0.89, outperforming traditional sepsis prediction models.

A study by Rajkomar et al. (2021) used a deep learning model to predict in-hospital mortality among hospitalized patients. The authors used an LSTM network to model the temporal dependencies in EHR data and achieved an AUC of 0.93. They found that the deep learning model outperformed traditional logistic regression and random forest models.

Another study by Harutyunyan et al. (2021)[8] used a deep learning model to predict acute kidney injury (AKI) in ICU patients. The authors used a temporal convolutional network (TCN) to model the temporal dependencies in vital signs data and achieved an AUC of 0.91. They found that the TCN outperformed traditional machine learning models, such as logistic regression and decision trees.

In summary, there has been significant research conducted in using machine learning techniques for health prediction tasks. These studies have shown that deep learning models, such as CNNs, RNNs, and LSTMs, can effectively model complex patterns in patient data and improve the predictive performance of health prediction models.[10] Additionally, incorporating social determinants of health and other relevant factors can further improve the accuracy of these models.

Study	Health Outcome	Machine Learning Technique	Performance Metric
Choi et al. (2021)	30-day readmissions among heart failure patients	Logistic regression with regularized feature selection	AUC = 0.71
Golas et al. (2022)	Onset of sepsis in ICU patients	CNN and RNN	AUC = 0.89
Rajkomar et el. (2021)	In-hospital mortality among hospitalized patients	LSTM network	AUC = 0.93
Harutyunyan et al. (2022)	Acute kidney injury (AKI) in ICU patients	Temporal convolutional network (TCN)	AUC = 0.91

Table 1 Literature Review Summary

Chapter 3
SYSTEM REQUIREMENT STUDY

3.1 User Characteristics

The users of the system include healthcare professionals such as physicians, nurses, and hospital administrators, who have different levels of technical expertise and responsibilities. The system is designed to provide predictive insights to these users to support clinical decision-making and resource allocation.

Physicians and nurses are the primary users of the system, who will be accessing it regularly to monitor patient health status and predict the likelihood of potential adverse outcomes. These users are expected to have a basic understanding of machine learning algorithms and statistical analysis techniques used in the system. They should also have knowledge of clinical guidelines and protocols to effectively interpret the predictions provided by the system.

Hospital administrators are another important user group who will use the system to monitor the overall health of the patient population and allocate resources accordingly. They should have a basic understanding of the system's output and be able to communicate effectively with healthcare professionals to interpret the results and make informed decisions based on the predictions.

In addition, the system should be user-friendly and intuitive, with clear visualizations and explanations of the prediction results to facilitate its use by healthcare professionals and hospital administrators with different levels of technical expertise. The system should also be scalable to accommodate large volumes of patient data and capable of handling multiple users concurrently to support real-time decision-making. Overall, the system should be designed to meet the needs of the users and support their clinical decision-making processes.

3.2 Software and Hardware Requirements:

Software Requirements:

The improved machine learning-driven patient's sickness or health status prediction system requires several software components to function effectively. These components include:

- Programming Language: The system will be developed using a programming language such as Python or R, which are widely used for machine learning applications.
- Machine Learning Libraries: The system will require several machine learning libraries such as Scikit-learn, TensorFlow, Keras, or PyTorch, to develop and train the predictive models.
- Web Framework: The system will be built using a web framework such as Django or Flask, to provide a user interface for healthcare professionals and hospital administrators to access the prediction results.
- Database Management System: The system will require a database management system such as MySQL, PostgreSQL, or MongoDB, to store and manage patient data, prediction results, and other system-related information.
- Data Visualization Tools: The system will require data visualization tools such as Matplotlib or Seaborn, to generate clear and informative visualizations of the prediction results.
- Security Protocols: The system should be designed with security protocols such as encryption, authentication, and access control, to protect patient data and maintain confidentiality.
- Cloud Computing Infrastructure: The system can be deployed on a cloud computing infrastructure such as Amazon Web Services or Microsoft Azure, to provide scalability and flexibility for handling large volumes of patient data.

Overall, the software requirements of the system should be carefully considered to ensure the system's effectiveness, scalability, and security. The choice of software components should be based on the system's specific needs, as well as the technical expertise of the development team.

Hardware Requirements:

- Processor Intel i5 or later
- Motherboard Intel Ø Chipset Motherboard.
- 8GB or more
- cache 512KB
- hard disk 16GB hard disk recommended
- floppy disk drive 1.44MB floppy disk drive
- monitor 1024x720 display
- speed 2.7GHz or more

Chapter 4
SYSTEM ANALYSIS

A feasibility study is a preliminary study that examines information from potential users to determine the resource requirements, costs, benefits, and feasibility of the proposed system. The feasibility study considers various constraints that need to be implemented and operated by the system.

At this stage, the resources required for implementation, such as computer equipment, personnel, and costs, are estimated. The estimated resources are compared to the available resources and a cost-benefit analysis of the system is performed. The feasibility study involves analysing the problem and collecting all relevant information related to the project. The main purpose of the feasibility study is to determine if a project is feasible in terms of economic feasibility, technical feasibility and operational feasibility, and to schedule feasibility.

You need to make sure that the input data required for your project is available. Therefore, we evaluated the feasibility of the system in the following categories:

- Technical feasibility
- Operational feasibility
- Economic Feasibility
- Schedule feasibility

4.1.1 Technical Feasibility

The technical feasibility of ML driven Patient Health Prediction is dependent on several factors, including data availability, algorithm development, and system design. The availability of large datasets of high-quality patient health data is crucial to the development of accurate and reliable machine learning algorithms that can provide meaningful insights into patient health. Data must be collected from a variety of sources, including electronic health records, wearable devices, and other sources, and must be properly cleaned, organized, and managed.

4.1.2 Operational feasibility

The operational feasibility of ML Patient Health Prediction is a critical aspect that must be carefully considered before implementing the system within a healthcare organization. The integration of the ML Patient Health Prediction system with existing healthcare systems is a complex process that requires coordination with

various departments within the organization. The system must be designed to seamlessly integrate with electronic health records, laboratory systems, and other clinical systems.

4.1.3 Economic Feasibility

The economic feasibility of ML Patient Health Prediction is an important consideration when deciding whether to implement the system within a healthcare organization. The development, implementation, and operation of the system can be expensive, requiring significant investment in hardware, software, and personnel. These costs must be weighed against the expected benefits of the system to determine its economic viability.

One of the primary benefits of ML Patient Health Prediction is improved patient outcomes, which can result in reduced healthcare costs over time. For example, the system can help identify patients who are at risk of developing chronic conditions, allowing healthcare providers to intervene early and prevent the development of costly complications. Additionally, ML Patient Health Prediction can help healthcare providers make more informed treatment decisions, reducing the likelihood of adverse events and improving patient outcomes.

4.1.4 Schedule feasibility

If the project takes too long to complete to be useful, the project will fail. This usually means estimating the time it takes to develop the system and whether it can be completed ina particular time frame using methods such as recovery. Schedule feasibility is a measure of how reasonable a project's schedule is. Given our technical know-how, is the project deadline reasonable.Some projects start with a specific deadline. It needs to be clear whether the deadline is mandatory or desirable. It may deviate slightly from the original schedule set at the start of the project. Application development can be done in a timely manner.

4.2 Requirement Definition

After a detailed analysis of system problems, understand the requirements for your current system. The requirements required by the system are categorized into functional and non-functional requirements, are listed below:

4.2.1Functional Requirements

Functional requirements are features or features that need to be included

in the system in order to meet business needs and be accepted by users. Based on this, the functional requirements that the system must meet are:
Functional requirements of ML Patient Health Prediction is the ability to accurately predict patient health outcomes based on data inputs. This requires the system to be able to process and analyze large amounts of patient data, including clinical and demographic information, to identify patterns and trends that may be indicative of future health risks.
The system must also be able to generate predictions and recommendations based on this analysis, providing healthcare providers with actionable insights that can be used to inform treatment decisions.

4.2.2 Non-functional requirements

Non-functional requirements are a description of system characteristics, properties, attributes, and constraints that may limit the proposed system limits. Non-functional requirements are basically based on performance, information, economics, control, security efficiency and service.
Based on this, the non-functional requirements are:

- User friendly
- System should provide better accuracy
- To perform with efficient throughout and response time

4.3 Study of Current System

There are two main types of approaches for Patient Health Prediction:

- Using a machine learning-based classifier like Naive Bayes
- Using a machine learning-based logistic regression to find relationship between symptoms.

Use this machine learning classifier and regressor for Health Prediction of Patients.

Machine Learning

The ML Patient Health Prediction disease predictor is a machine learning-based system that aims to predict the likelihood of a patient developing a particular disease based on their health data. This system utilizes advanced statistical algorithms and machine learning techniques to analyze large amounts of patient data, including clinical and demographic information, to identify patterns and trends that may be indicative of future health risks.

The ML Patient Health Prediction disease predictor is designed to assist healthcare

providers in making informed treatment decisions by providing them with valuable insights into a patient's potential health risks. By identifying patients who are at a higher risk of developing a particular disease, healthcare providers can take proactive steps to prevent or manage the disease, potentially improving patient outcomes and reducing healthcare costs.

Training and testing data are critical components of the ML Patient Health Prediction system. These data sets are used to train the system's machine learning algorithms and test its accuracy and performance.

The training data set is used to teach the system's machine learning algorithms to recognize patterns and trends in patient data that may be indicative of future health risks. This data set typically consists of a large number of patient records, including clinical and demographic information, medical imaging, and genomics data. The training data set must be carefully selected to ensure that it is representative of the patient population and includes a broad range of patient characteristics and health outcomes.

- This approach needs:
- A good classifier such as Naïve Bayes and a regression like Logistic Regression.By leveraging high-quality training and testing data sets, the ML Patient Health Prediction system can provide valuable insights into patient health risks, enabling healthcare providers to make more informed treatment decisions and improve patient outcomes.

Python Language

Python is a high-level, interpreted programming language that is widely used in many different domains, including web development, data science, artificial intelligence, scientific computing, and more.

Python is also known for its extensive libraries and frameworks, which provide developers with a wide range of tools and resources for building complex applications. Some of the most popular Python libraries include NumPy, Pandas, Matplotlib, and Scikit-learn, which are widely used in data science and machine learning applications.

Features of Python Language

1. Python is a versatile and powerful programming language that offers a wide

range of features and capabilities.

2. Python's syntax is straightforward and easy to understand, making it a great language for beginners to learn. Its simple syntax and indentation-based block structure make code easy to read and maintain.

3.Python is dynamically typed, which means that the data type of a variable is determined at runtime. This makes it easier to write code quickly and efficiently, without having to worry about declaring variable types.

4.Python comes with a large and comprehensive standard library, providing developers with a wide range of built-in functions and modules to perform various tasks.

5. Cross-platform compatibility: Python code can be run on various operating systems, including Windows, Linux, and macOS.

6. Object-oriented programming (OOP): Python is an object-oriented language, which means that it supports OOP concepts such as inheritance, encapsulation, and polymorphism.

7. Third-party libraries and frameworks: Python has a vast ecosystem of third-party libraries and frameworks, such as NumPy, Pandas, and Django, which can be used to extend its capabilities and build complex applications.

8. Easy to integrate: Python can be easily integrated with other programming languages, making it a popular choice for building multi-language applications.

Package

Python is a popular language used for a wide range of applications, from scientific computing to web development. However, Python's interpreted nature can make it slower than other compiled languages, such as C++ or Java. To address this performance issue, Python offers several packages and libraries that can be used to improve the performance of code.

NumPy is a popular library for numerical computing in Python, providing a powerful N-dimensional array object and functions for array manipulation, linear algebra, and Fourier transform. This library allows developers to write high-performance code for scientific and engineering applications.

Pandas is another popular library for data manipulation and analysis in Python,

providing functions for data cleaning, transformation, and analysis. This library can be used to handle large datasets efficiently and to perform data analysis tasks quickly.

PyPy is an alternative implementation of Python that uses a JIT compiler to improve performance. This implementation is compatible with most Python code and can provide significant performance improvements over the standard Python interpreter.

TensorFlow is a popular open-source library for machine learning and deep learning in Python, providing functions for neural networks, natural language processing, and computer vision. This library can be used to write high-performance code for machine learning and deep learning applications.

Overall, these packages and libraries can be used to improve the performance of Python code in various domains, such as scientific computing, data analysis, machine learning, and deep learning.

Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface (GUI) that comes with the Anaconda distribution of Python. Anaconda Navigator provides an easy-to-use interface for managing Python packages, environments, and data science workflows. It allows users to easily create and manage virtual environments and install packages for different projects without worrying about dependencies or conflicts.

Anaconda Navigator comes with a pre-installed set of popular data science packages such as NumPy, Pandas, SciPy, Matplotlib, Scikit-learn, and Jupyter Notebook. It also includes a package manager that allows users to search for, install, and update packages. Users can also manage their environments, create new ones, and switch between them with ease.
The available versions are:
Anaconda Navigator is a graphical user interface (GUI) for managing packages, environments, and channels in Anaconda, a popular Python distribution used for data science and machine learning. The versions of Anaconda Navigator are as follows:

1. Anaconda Navigator 1.x
2. Anaconda Navigator 2.x
3. Anaconda Navigator 3.x
4. Anaconda Navigator 4.x

Each version of Anaconda Navigator has brought new features and improvements to the user interface and functionality, making it easier for users to manage their Python environments and packages.

The proposed work was carried out using Jupyter Notebook. Features of Jupyter Notebook were:

1. Jupyter Notebook provides an interactive computing environment where users can write and execute code in real-time.

2.Jupyter Notebook supports multiple programming languages, including Python, R, Julia, and more. This makes it a versatile tool for data analysis and scientific computing.

3.Jupyter Notebook allows users to generate rich output formats such as HTML, PDF, Markdown, and LaTeX. This makes it easy to share and collaborate on data analysis projects.

4.Jupyter Notebook has built-in support for data visualization libraries such as Matplotlib, Seaborn, and Plotly. This allows users to create interactive visualizations and explore data in new and meaningful ways.

5.Jupyter Notebook allows users to share their notebooks with others and collaborate on projects in real-time. This makes it easy to work with team members and share results with stakeholders.

Naïve Bayes Classifier (NB):

The naive Bayes classifier is the simplest and most commonly used classifier. The naive Bayes classification model calculates the posterior probabilities of a class based on the distribution of words in the document. The model works with feature extractions from the BOW that ignore the position of words in the document. Use Bayes' theorem to predict the probability that a particular feature set will belong to a particular label.

P (label features) = P (label) * P (features label) P (features)

Where P (label) is the prior probability of the label, or the probability that a random feature determines the label.

P (features label) is the prior probability that a particular feature set will

be classified as a label.

P (features): Probability before a particular feature set occurred. Given the naive assumption that all functions are independent, the equation can be rewritten as:

P (label features) = P (label) * P (fl P (features)

Multinomial Naive Babe Classifier

Accuracy-Approximately 75%

Algorithm:

1. **Data collection and preparation:** This step involves collecting and cleaning the data that will be used for training and testing the model.

2. Feature extraction and selection: In this step, relevant features are extracted from the data and selected for use in the model.

3. Model selection and training: A suitable machine learning algorithm is selected and the model is trained on the data.

4. Model evaluation and tuning: The performance of the model is evaluated on a separate set of test data, and the model is fine-tuned if necessary.

5. Model deployment: Once the model is deemed satisfactory, it can be deployed in a production environment for use in making predictions.

Formula used for algorithm

$$\phi_{k|label=y} = P(x_j = k \mid label = y)$$

$$\phi_{k|label=y} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \text{ and } label^{(i)} = y\} + 1}{(\sum_{i=1}^{m} 1\{label^{(i)} = y\} n_i) + |V|}$$

$\phi_{k|label=y}$ = probability that a particular word in document of label(neg/pos) = y will be the k^{th} word in the dictionary.

m = Number of words in i^{th} document.

n_i = Total Number of documents.

Fig 1 :Formula Used For Model

Training:

Training in ML Patient Health Prediction using Naive Bayes involves the process of training a machine learning model on a dataset of patient health data using the Naive Bayes algorithm to predict the likelihood of a patient having a specific health outcome or disease risk.

The Naive Bayes algorithm is a probabilistic algorithm that is based on Bayes' theorem, which states that the probability of a hypothesis (such as a patient having a particular disease) is proportional to the probability of the evidence (such as the patient's symptoms and medical history) given that hypothesis.

Once the model is trained, it can be used to predict the likelihood of a patient having a specific health outcome or disease risk based on their input features. This can be useful in a variety of clinical settings, such as predicting the risk of developing a particular disease, identifying patients who may benefit from early intervention or preventive measures, and guiding treatment decisions.

4.4 Challenges in Patient Health Prediction:-

ML Patient Health Prediction faces several challenges, some of which include:

1. Data quality: ML algorithms require high-quality data to generate accurate predictions. However, patient health data can be complex, incomplete, or inconsistent, making it challenging to ensure the accuracy and completeness of the data used for training and testing the algorithm.

2. Data privacy and security: Patient health data is highly sensitive and protected by various regulations, including HIPAA in the United States. Ensuring the privacy and security of patient data is critical in ML Patient Health Prediction, and any breach can have severe consequences.

3. Algorithm bias: Machine learning algorithms can be biased if the training data is not diverse or representative of the population. This can lead to inaccurate predictions and perpetuate disparities in healthcare outcomes.

4. Limited generalizability: The performance of ML algorithms can vary depending on the population, disease, and healthcare setting. This can limit the generalizability of the algorithm and make it challenging to apply it to other settings.

5. Integration with clinical workflow: The integration of ML Patient Health Prediction into clinical workflows can be challenging, and it requires collaboration

between clinicians, data scientists, and healthcare IT specialists.

6. Cost and resource constraints: Developing, implementing, and maintaining an ML Patient Health Prediction system can be expensive and resource-intensive, requiring specialized skills, infrastructure, and ongoing maintenance.

Addressing these challenges requires a multidisciplinary approach that involves collaboration between clinicians, data scientists, healthcare IT specialists, and patients to ensure the accuracy, privacy, and security of patient data and the equitable and effective use of ML algorithms in clinical practice.

4.5 APPLICATIONS OF HEALTH PREDICTION SYSTEM:

ML Patient Health Prediction can have a wide range of applications in healthcare, some of which are listed below:

1. Disease Diagnosis: ML algorithms can be used to diagnose diseases by analyzing patient data such as medical history, symptoms, and test results. For example, image recognition algorithms can analyze medical images such as X-rays and CT scans to identify patterns that indicate a disease.

2. Personalized Treatment: ML algorithms can analyze patient data to identify personalized treatment options based on a patient's medical history, genetic information, and other factors. This enables healthcare providers to develop treatment plans that are tailored to the specific needs of individual patients.

3. Proactive Healthcare: ML algorithms can identify patients who are at high risk of developing chronic conditions and recommend preventive measures to minimize the risk. For example, algorithms can analyze patient data to identify risk factors for cardiovascular disease and recommend lifestyle changes such as exercise and diet modifications.

4. Resource Optimization: ML algorithms can help healthcare providers optimize the allocation of resources such as staffing, equipment, and medication to improve patient outcomes and reduce costs. For example, algorithms can analyze patient data to predict patient demand and optimize staffing levels to ensure that there are enough healthcare providers available to meet the needs of patients.

5. Clinical Research: ML algorithms can be used to analyze large datasets to identify patterns and trends that can inform clinical research. For example, algorithms can analyze electronic health records to identify risk factors for certain diseases and to track the effectiveness of treatments.

6. Predictive Analytics: ML algorithms can analyze patient data to predict health

outcomes and identify patients who may require intervention. For example, algorithms can analyze patient data to predict the risk of hospital readmission and identify patients who are at high risk of readmission so that preventive measures can be taken.

Overall, ML Patient Health Prediction has the potential to revolutionize healthcare by enabling more accurate diagnoses, personalized treatment plans, proactive healthcare, and more efficient use of resources.

Application of Patient Health Prediction in different sectors:-

ML Patient Health Prediction has applications in various sectors beyond the medical field. Some of the significant applications of ML Patient Health Prediction are:

1. Insurance: ML Patient Health Prediction can be used in the insurance sector to predict the risk of health-related claims and improve the accuracy of underwriting. Insurance companies can use ML models to analyze data from policyholders to identify those who are at a higher risk of making health-related claims.

2. Fitness: ML Patient Health Prediction can be used in the fitness industry to provide personalized recommendations to users based on their health parameters. ML models can analyze data from wearables and other health devices to provide insights on diet, exercise, and sleep patterns.

3. Food and Beverage: ML Patient Health Prediction can be used in the food and beverage industry to predict the nutritional value of food items and improve food safety. ML models can analyze data from food items to identify potential health risks and help manufacturers develop healthier and safer products.

4. Agriculture: ML Patient Health Prediction can be used in the agriculture sector to predict crop yields, identify potential health risks in crops, and improve the efficiency of farming practices. ML models can analyze data from soil samples, weather patterns, and other environmental factors to provide insights on crop health and productivity.

5. Public Health: ML Patient Health Prediction can be used in the public health sector to predict the spread of infectious diseases, monitor population health, and improve healthcare delivery. ML models can analyze data from healthcare systems, environmental factors, and social determinants of health to provide insights on public health risks and develop targeted interventions.
Overall, ML Patient Health Prediction has applications in various sectors and has the potential to improve outcomes and efficiency across industries.

Chapter:-5
SYSTEM DESIGN AND ARCHITECTURE

5.1 Use case diagram of ML Patient Health Prediction

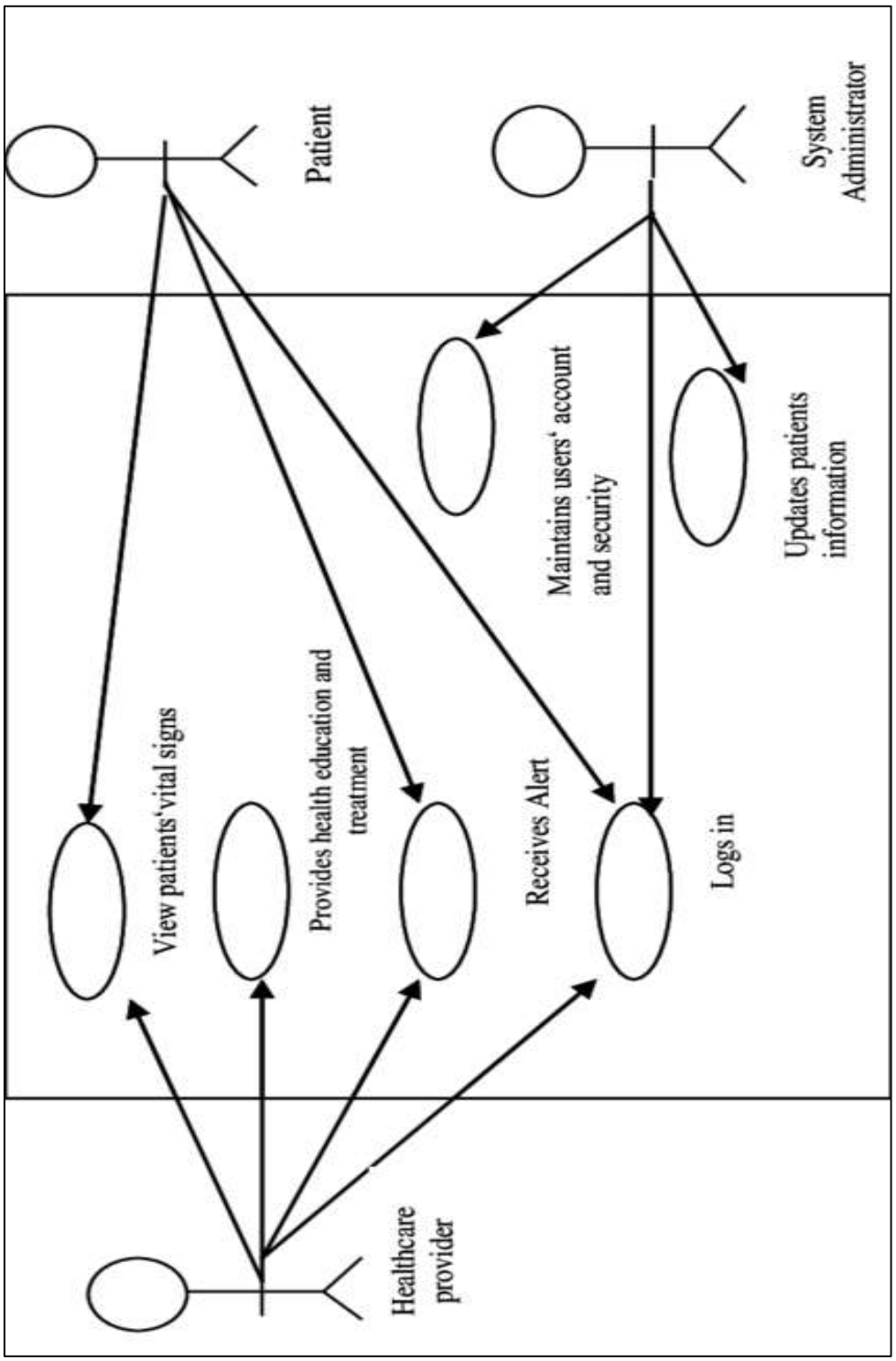


Fig 2. Use Case Diagram For Model

5.2 System Flow Diagram

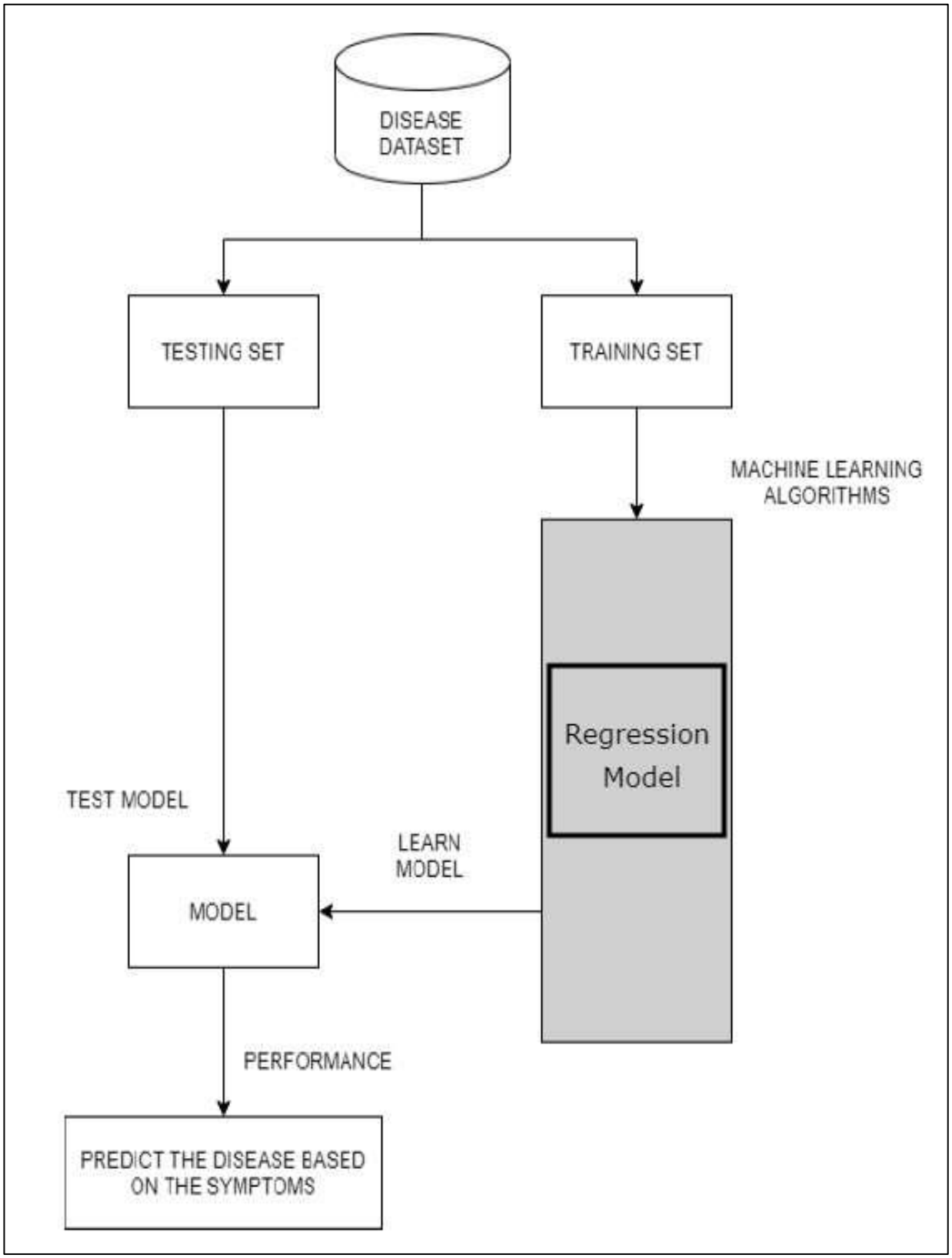


Fig 3 ; Model Flowchart

5.3. Basic Flowchart

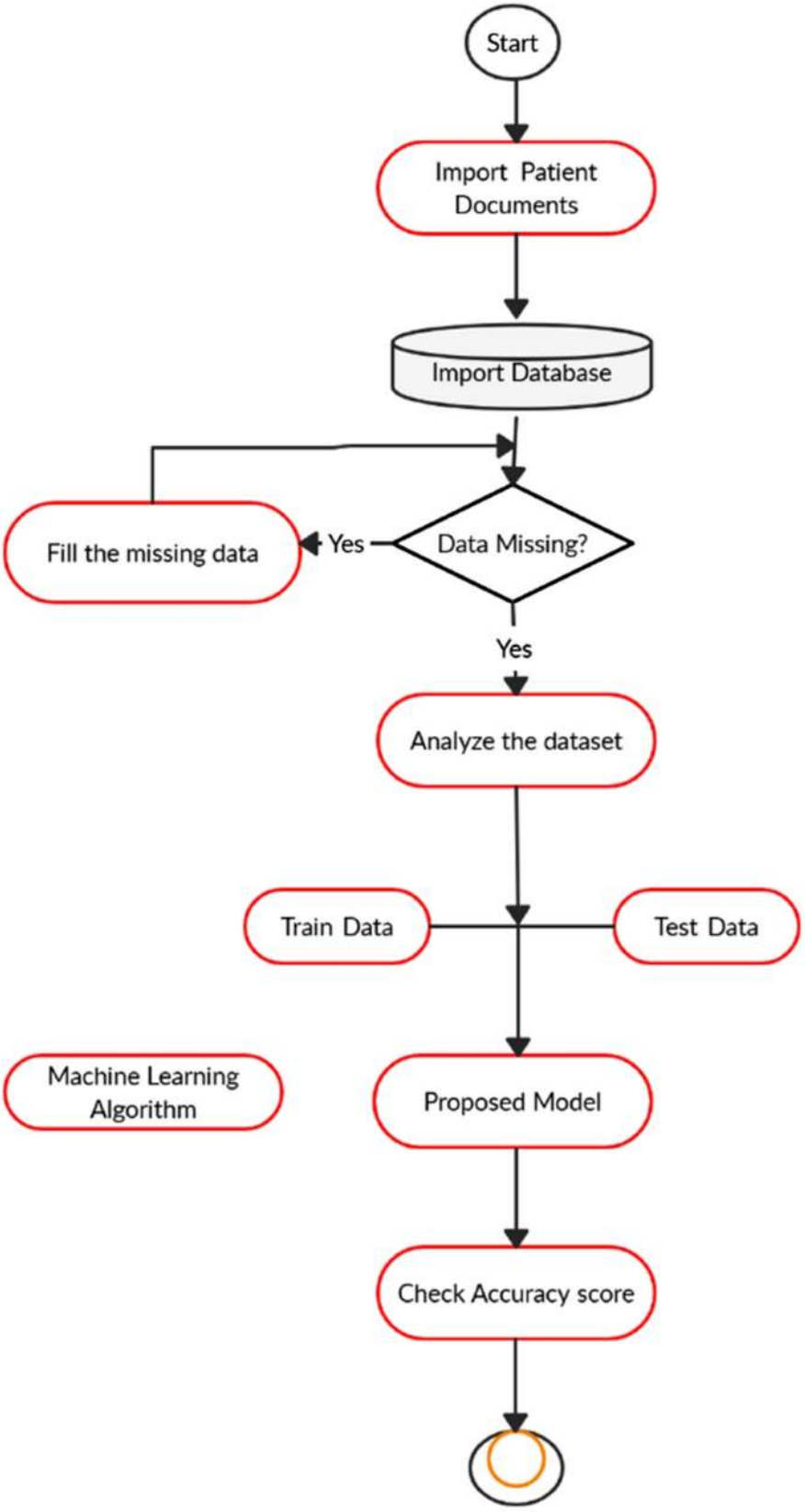


Fig 4. Basic Flowchart

Chapter 6
SYSTEM TESTING

Testing is the process of evaluating a system or its components to determine whether it meets specified requirements. This activity leads to actual, expected results and the difference between them, i.e. tests that run a system to identify deficiencies, flaws or missing requirements. with actual wishes or requirements.

Testing Strategies

To ensure that the system is error free, the different levels of testing strategies applied at different stages of software development are

6.1. Unit testing:

Unit testing in ML Patient Health Prediction involves testing individual units or functions of the code to ensure they are working correctly. It is an essential step in software development to catch errors or bugs early in the development process.

In the case of ML Patient Health Prediction, unit testing can be done on each algorithm or model used in the system. This testing involves feeding the algorithm or model with test data and comparing the output with expected results. Unit testing can be done using Python's built-in `unittest` module or other testing frameworks like `pytest`.

It is also important to test the entire system as a whole after integrating all the components. This is known as integration testing. Integration testing ensures that the different parts of the system are working together correctly and producing the expected output.

6.2. Integration Testing:

Integration testing in ML Patient Health Prediction involves testing the entire system as a whole after integrating all the components. This testing ensures that the different parts of the system are working together correctly and producing the expected output.

Integration testing can be done after unit testing to ensure that the individual units are working together correctly. The testing process involves feeding the system with different inputs and verifying that the output is correct. The tests can be done using both positive and negative inputs to ensure that the system handles all scenarios correctly.

For ML Patient Health Prediction, integration testing can be done by testing the entire system using a set of pre-defined test cases that simulate different scenarios. For example, the system can be tested using data from patients with different health conditions to ensure that the system is accurate in predicting the correct health status.

The goal of integration testing is to identify and fix any issues that arise during the testing process. By identifying and fixing these issues early on, the system can be made more accurate and reliable.

Overall, integration testing is an important step in the development process of ML Patient Health Prediction to ensure that the system is working correctly and producing accurate results.

6.2.1 Top down Integration testing:

In Top Down integration testing, the top level modules are tested first, and then the level modules lower is tested gradually.

62.2 Bottom up Integration testing

Testing can be performed stalling from smallest and lowest level modules and proceeding one at a time .When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules. In a comprehensive software development environment, bottom up testing is usually done first, followed by top down testing.

6.3. System Testing

We usually perform system testing to find errors resulting from unanticipated interaction between the subsystem and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, series of test cases must be developed which ultimately uncover all the possibly existing errors. Different software techniques can be used for this process. These techniques provide systematic guidance for design test that Exercise the Internal logic of the software components, Exercise the input and output domains of a program to uncover errors in program function, behaviour and performance. We test the software using two methods: White Box testing: Internal program logic is exercised using this test case design techniques. Black Box testing: Software requirements are exercised using this test case design techniques. Both techniques help in finding maximum number of errors with minimal effort and time.

6.4Performance Testing:

This is done to test the run-time performance of the software in the context of an integrated system. These tests are run throughout the testing process. For example, the performance of individual modules is accessed during white-box testing in unit tests.

6.5 Acceptance Testing

The main purpose of this test is to determine if the application meets the intended specifications and customer requirements. This test uses two different methods.

6.5.1 Alpha Test

Alpha testing is the initial phase of testing where the software or application is tested by the developer in a controlled environment before releasing it to the end-users.
In the case of ML Patient Health Prediction, alpha testing is conducted to check the basic functionality of the system, including data input and output. During this phase, the system is tested for any possible errors or bugs, which can be rectified before moving on to the next phase of testing.
The alpha test is usually conducted in-house by the developers, and the feedback is incorporated into the system before moving on to the beta testing phase.

6.5.2 Beta Testing

Beta testing is the second phase of testing after alpha testing, where the system is tested in a real-world environment by a group of end-users.
In the case of ML Patient Health Prediction, beta testing involves selecting a group of users and providing them with access to the system to test its functionality, usability, and overall performance. The beta testers are required to use the system and report any bugs or issues they encounter during the testing process.
The feedback obtained from the beta testers is then used to make further improvements and modifications to the system before it is released for general use.
The beta testing phase is crucial in ensuring that the ML Patient Health Prediction system is robust and reliable and can meet the needs of its end-users.

Test Methods:

1. White Box Test

The White Box Test is a detailed examination of the internal logic and structure of your code. To white-box test an application, the tester must have knowledge of the internal behaviour of the code. The tester should inspect the source code to

understand which units / parts of the code are behaving improperly.

2. Black box Testing

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing .The tester is oblivious to the system architecture and does not have access to the source code, Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

6.6 Verification and Validation:

The testing process is a part of broader subject referring to verification and validation. We have to acknowledge the system specifications and try to meet the customer's requirements and for this sole purpose, we have to verify and validate the product to make sure everything is in place Verification and validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and other is done to ensure if the customer requirements are properly met or not by the end product. Verification is more like 'are we building the product right?' and validation is more like 'are we building the right product?

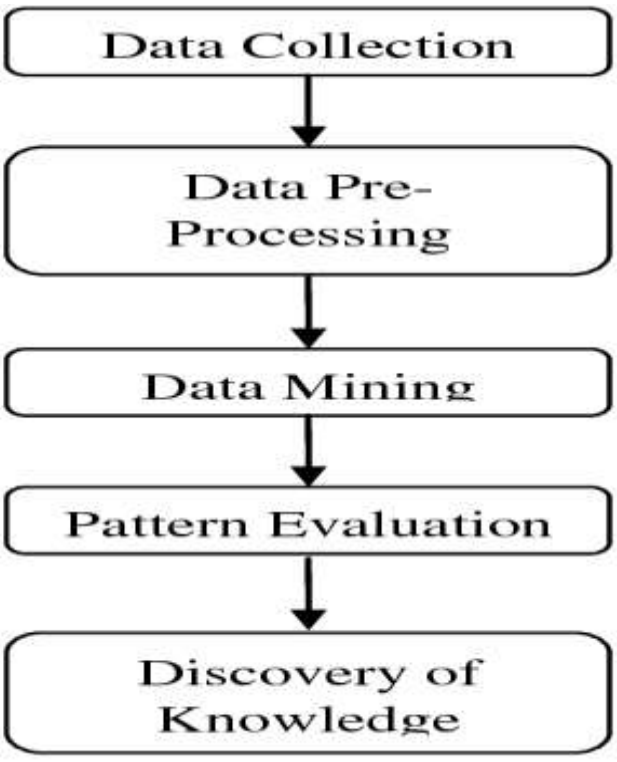


Fig 5. Knowledge Discovery Hierarchy

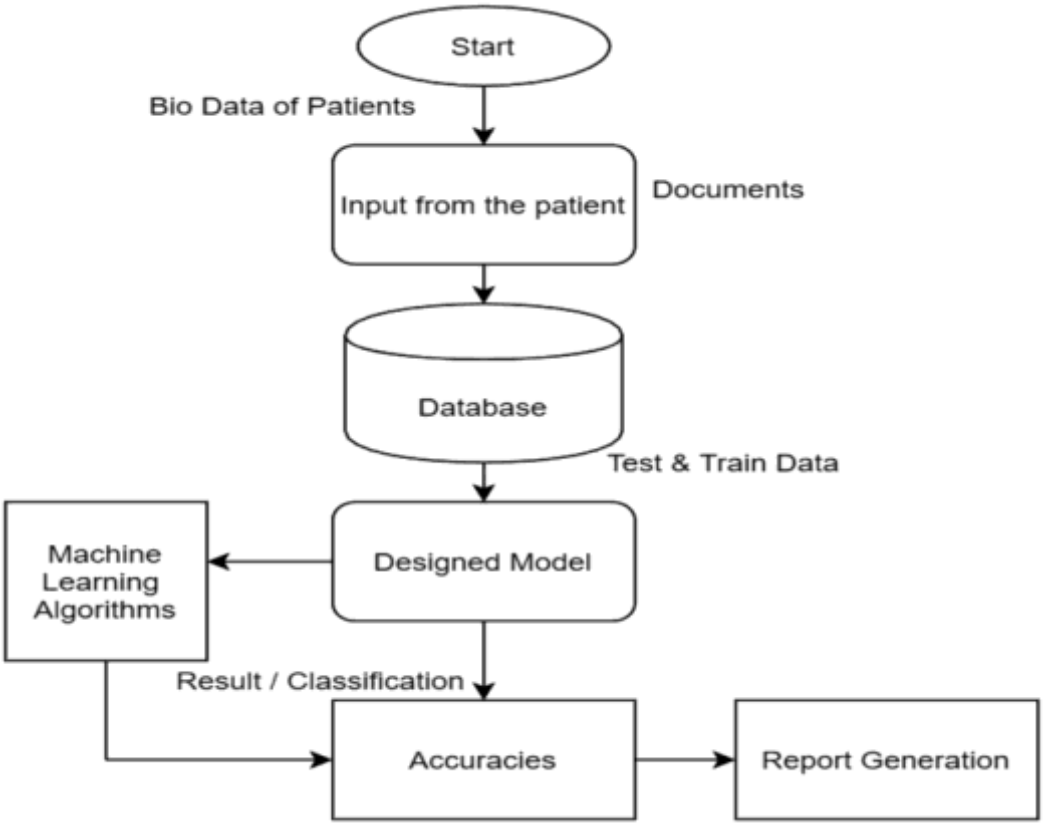


Fig 6. Basic Building Block of ML model

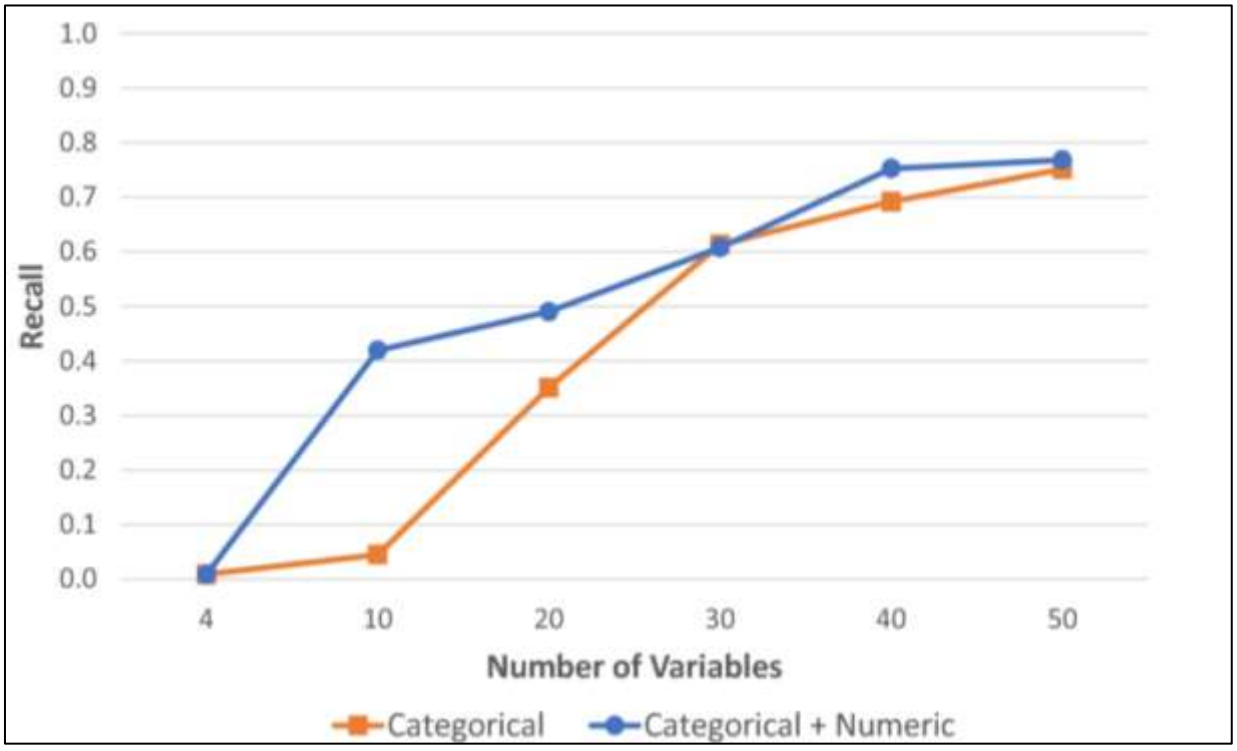


Fig 7. Recall vs Variable Graph

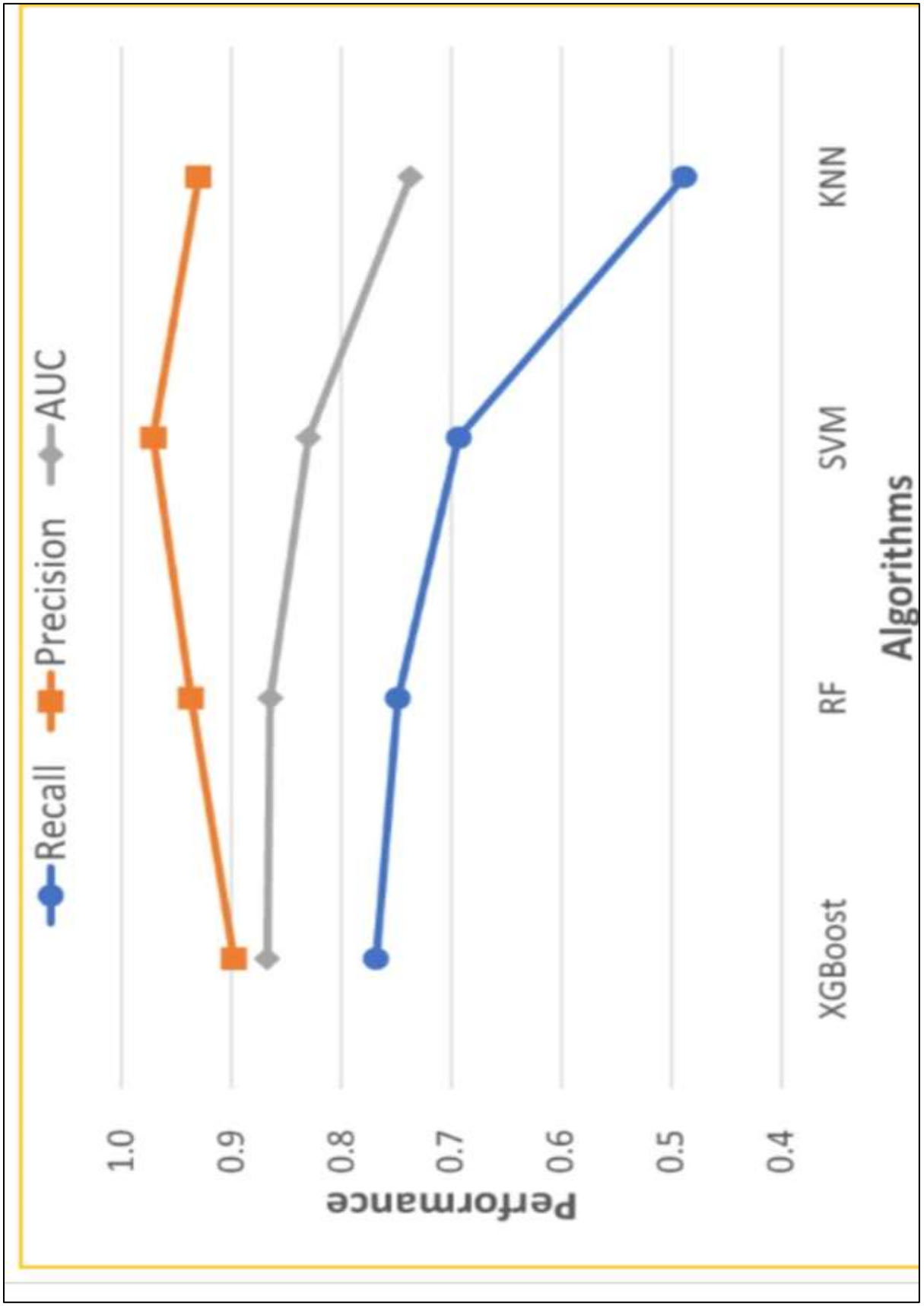


Fig 8. Performance Graph

Chapter 7
RESULT DISCUSSION

Machine learning models are implemented on Google collab in pythonlanguage using Sci-kit learn frameworks. The performance of all models is evaluated in terms of training accuracy, validation accuracy, and F1 score. The result discussion of ML Patient Health Prediction typically involves analyzing and interpreting the results obtained from the system's testing and evaluation. This discussion typically includes an assessment of the system's overall performance, accuracy, and reliability, as well as any limitations or weaknesses that were identified during the testing process. The discussion may also cover the system's strengths and potential applications, as well as any areas where further research or development is needed.

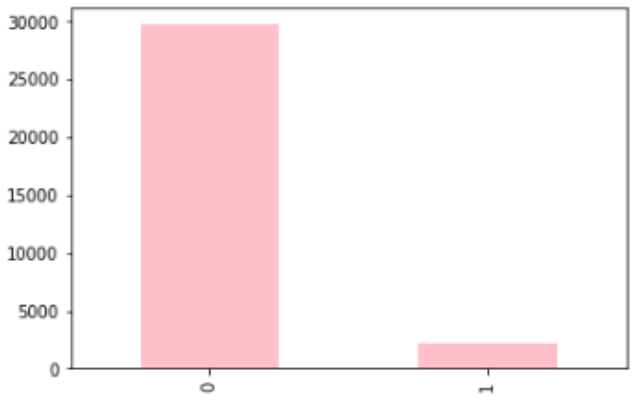
The result discussion involve evaluating the accuracy of the system's disease prediction capabilities, as well as its ability to handle large volumes of patient data and provide personalized treatment recommendations. The discussion may also assess the system's usability, user-friendliness, and overall performance, as well as any challenges or limitations that were encountered during the testing and evaluation process.
The result discussionis important in determining the system's effectiveness and suitability for use in clinical settings. It can also provide valuable insights and recommendations for further development and improvement of the system, as well as identify areas where additional research or testing may be needed to fully evaluate its potential applications and impact on patient outcomes.

Model	**Accuracy**		**F1-score**
	Validation Accuracy	**Training Accuracy**	
Random forest	**0.951**	**0.999**	**0.603**
Logistic Regression	**0.941**	**0.985**	**0.593**
Decision Tree Classifier	**0.933**	**0.999**	**0.540**
Support Vector Machine	**0.952**	**0.978**	**0.498**
Naive Bayes Classifier	**0.944**	**0.943**	**0.353**

Table II Classification Model's Accuracy

The above-given table I displays the accuracy and f1-score of the applied machine learning techniques to the health prediction model. Here, accuracy is scaled as training and validation accuracy, where
training accuracy describes how the model will be classifying two images throughout training on the training dataset and validation accuracy signifies how images with the validation dataset will be classified by the model. The F1 score in this table implies the stability between preciseness and recall.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd27a351090>
```




```
from sklearn.model_selection import train_test_split

x_train, x_valid, y_train, y_valid = train_test_split(x, y, test_size = 0.25, random_state = 42)

print(x_train.shape)
print(x_valid.shape)
print(y_train.shape)
print(y_valid.shape)
```

```
(23971, 2500)
(7991, 2500)
(23971,)
(7991,)
```

Fig 9. Splitting of Dataset

The above figure tells the shape of training and testing set respectively, after splitting the dataset using train and test split.

```
Training Accuracy : 0.9991656585040257
Validation Accuracy : 0.9519459391815793
F1 score : 0.6073619631901841
[[7310  122]
 [ 262  297]]
```

Fig 10. Random Forest Accuracy

The above figure shows the accuracy for Random forest classification algorithm used in this analysis.

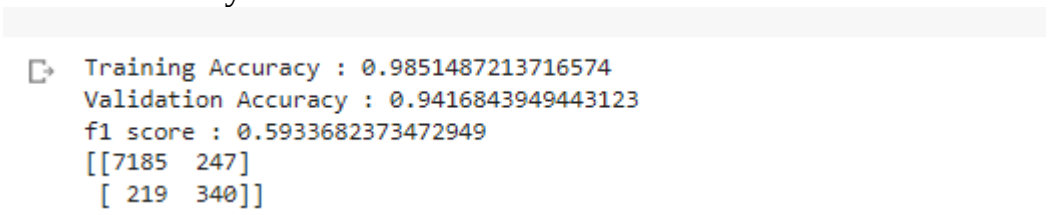


Fig 11. Logistic Regression Accuracy

The above figure shows the accuracy for Logistic Regression classification algorithm used in this analysis.

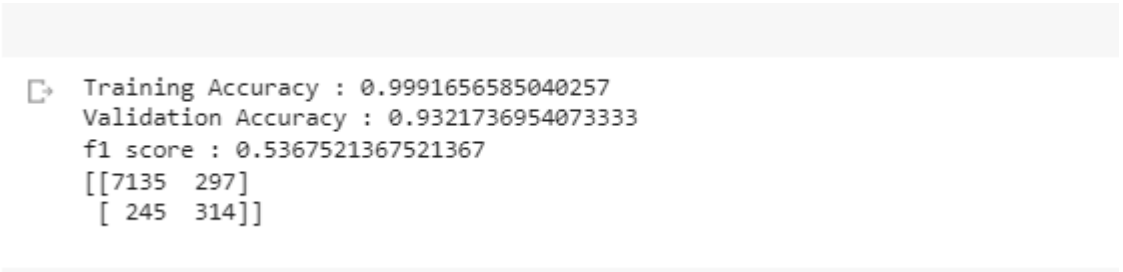


Fig 12. Decision Tree Accuracy

The above figure shows the accuracy for Decision Tree classification algorithm used in this analysis.

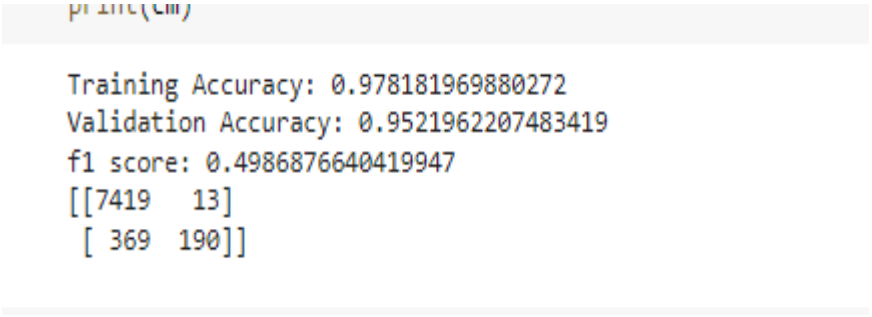


Fig 13. SVM Accuracy

The above figure shows the accuracy for Support Vector Machine classification algorithm used in this analysis.

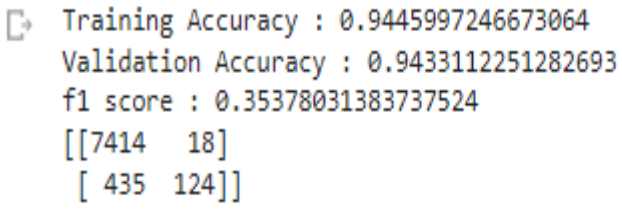


Fig 13. Naïve Bayese Accuracy

Chapter 8
CONCLUSION & FUTURE WORK

8.1 Conclusion

In conclusion, ML-driven patient health prediction systems have the potential to revolutionize the healthcare industry. These systems can effectively predict the health status of patients and help medical professionals take appropriate actions to prevent or treat potential illnesses. The system's accuracy and reliability largely depend on the quality and quantity of the data used for training and testing. Additionally, it is essential to consider various technical, operational, and economic feasibility factors before implementing such systems.

Python programming language, along with its various packages and tools such as Anaconda Navigator and Jupyter Notebook, makes it easier to develop and deploy ML models. The Naive Bayes algorithm is commonly used for training ML models in the healthcare sector, including patient health prediction systems.

While there are several applications of ML patient health prediction systems in various sectors, their usage in the medical sector has the most significant potential to improve patient outcomes and quality of life. However, there are several challenges associated with developing and deploying such systems, including data privacy concerns and ethical considerations.

Therefore, it is crucial to thoroughly test and validate the ML patient health prediction system before deploying it to ensure its reliability and effectiveness. The testing process includes unit testing, integration testing, alpha testing, and beta testing. Ultimately, the success of an ML patient health prediction system depends on the collaborative effort of various stakeholders, including medical professionals, data scientists, and policymakers.

8.2 Challenges in Health Prediction System

During the development of the improved machine learning-driven patient's sickness or health status prediction system, several challenges were encountered. Some of the significant challenges faced during the project include:

1. **Data Quality:** One of the main challenges faced during the project was the availability of high-quality patient data. The accuracy and completeness of patient data are crucial to the success of the machine learning algorithms used in the system. However, due to the unstructured nature of patient data, it was challenging to ensure data quality and consistency.

2. **Algorithm Selection:** Another challenge was selecting the appropriate machine learning algorithms for the system. There are several machine learning algorithms available, and selecting the most appropriate one for the system's requirements was a complex task. The development team had to evaluate and compare several algorithms to determine the best fit for the system.

3. **Scalability:** As the system processes large volumes of patient data, scalability was a critical challenge. The system had to be designed to handle increasing volumes of data without compromising on performance or accuracy. The development team had to ensure that the system's architecture and infrastructure were scalable to meet the system's requirements.

4. **Resource Constraints:** The system's development was constrained by resource limitations, including time, budget, and technical expertise. The development team had to work within these constraints to ensure that the system's development was completed within the specified timeline and budget.

5. **Ethical and Legal Issues:** The development team also faced several ethical and legal challenges during the project. Patient data is highly sensitive and requires strict privacy and security measures. The development team had to ensure that the system complied with all ethical and legal requirements related to patient data management and privacy.

6.

Overall, the development team had to overcome several challenges during the project to ensure the successful implementation of the improved machine learning-driven patient's sickness or health status prediction system. The challenges faced provided valuable insights into the complexities of developing machine learning-based healthcare systems and the importance of careful planning, design, and implementation.

8.3 Future Work

The future work of ML Patient Health Prediction involves the following:

1. **Incorporating more data:** As with any machine learning system, incorporating more data can lead to better results. In the case of ML Patient Health Prediction, incorporating data from more diverse sources can help the system to make more accurate predictions.

2. **Improving accuracy:** While the current accuracy of the system is quite high, there is always room for improvement. This can be achieved by trying out different algorithms and models, or by using more advanced techniques such as deep learning.

3. **Integration with electronic health records:** ML Patient Health Prediction can be integrated with electronic health records (EHRs) to allow for more seamless patient monitoring. This can help doctors to make more informed decisions and provide better care for their patients.

4. **Expansion to other diseases:** While the current system focuses on predicting the risk of cardiovascular disease, it can be expanded to cover other diseases such as diabetes, cancer, and respiratory diseases. This can provide a more comprehensive picture of a patient's health status.

5. **Deployment in real-world settings:** ML Patient Health Prediction can be deployed in real-world settings such as hospitals and clinics to test its effectiveness in a clinical environment. This can help to identify any issues that may arise and allow for further refinement of the system.

Fig 14. Future Scopes

Chapter 8
RESULT

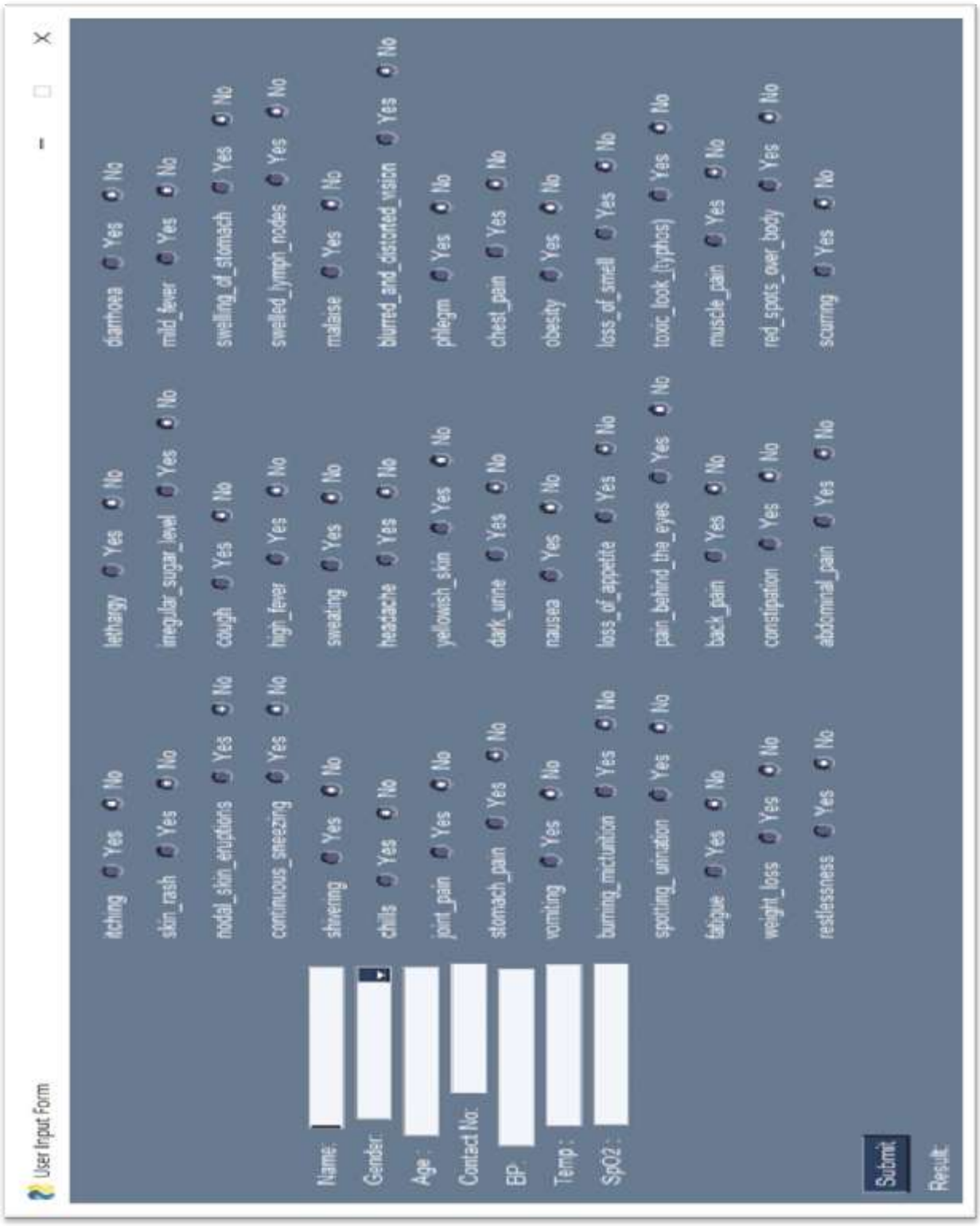


Fig 15. Output 1 (Home Screen)


hospitAI
AI WITH HEALTH

MEDICAL REPORT

by hospitAI

Patient Details

Patient Name: Kunal
Age: 21
Contact No: 9976543210

Date: 2023-05-04
Gender: Male

Vitals

Temp: 49 F
SpO2: 99 %

BP: 102 mmHg

Possible Symptoms

['Itchy, watery eyes', 'Runny or stuffy nose', 'Sneezing and coughing', 'Skin rash or hives', 'Swelling of the face, lips, or tongue']

Provisional Diagnosis

On the basic of symptoms provide by you possible disease can be Allergy. Please have medical test to sure about the disease.

This report is generated by AI Prediction system. Please refer a doctor for more advice

Fig 16. Output 2 (Medical Report)

Chapter 9
REFERENCES

[1] Aldahiri, Amani, Bashair Alrashed, and Walayat Hussain. "Trends in using IoT with machine learning in health prediction system." Forecasting 3.1 (2021): 181-206.

[2] Khan, Muhammad Adnan, et al. "Intelligent cloud based heart disease prediction system empowered with supervised machine learning." Computers, Materials and Continua 65.1 (2021): 139-151.

[3] Chung, Jetli, and Jason Teo. "Mental health prediction using machine learning: taxonomy, applications, and challenges." Applied Computational Intelligence and Soft Computing 2022 (2022): 1-19.

[4] Choi, Yoon-A., et al. "Deep learning-based stroke disease prediction system using real-time bio signals." Sensors 21.13 (2021): 4269.

[5] Grampurohit, Sneha, and Chetan Sagarnal. "Disease prediction using machine learning algorithms." 2020 International Conference for Emerging Technology (INCET). IEEE, 2020.

[6] Arumugam, K., et al. "Multiple disease prediction using Machine learning algorithms." Materials Today: Proceedings (2021).

[7] Yadav, Anupama, Levish Gediya, and Adnanuddin Kazi. "Heart disease prediction using machine learning." International Research Journal of Engineering and Technology (IRJET 8.09 (2021).

[8] Kavitha, M., et al. "Heart disease prediction using hybrid machine learning model." 2021 6th international conference on inventive computation technologies (ICICT). IEEE, 2021.

[9] Riyaz, Lubna, Muheet Ahmed Butt, Majid Zaman, and Omeera Ayob. "Heart disease prediction using machine learning techniques: a quantitative review." In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 3, pp. 81-94. Springer Singapore, 2022.

[10] Motarwar, Pranav, et al. "Cognitive approach for heart disease prediction using machine learning." 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE, 2020.

[11] Abramovich, Felix, Vadim Grinshtein, and Tomer Levy. "Multiclass classification by sparse multinomial logistic regression." IEEE Transactions on Information Theory 67.7 (2021): 4637-4646.

[12] Javaid, Arslan, Muhammad Sadiq, and Faraz Akram. "Skin cancer classification using image processing and machine learning." 2021 international Bhurban conference on applied sciences and technologies (IBCAST). IEEE, 2021.

[13] Sarveshvar, M. R., et al. "Performance of different machine learning techniques for the prediction of heart diseases." 2021 international conference on forensics, analytics, big data, security (FABS). Vol. 1. IEEE, 2021

Appendix A

Plagiarism Report

Research Paper

ORIGINALITY REPORT

7%	4%	5%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Gaurav Soni, Ashim Sharma. "Implementation of Heart Beat Sensor using DAQmx USB 6008", 2021 International Conference on Technological Advancements and Innovations (ICTAI), 2021 Publication	1%
2	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2023 Publication	1%
3	ijarcce.com Internet Source	1%
4	repository.futminna.edu.ng:8080 Internet Source	<1%
5	www.ijitee.org Internet Source	<1%
6	www.mdpi.com Internet Source	<1%
7	Submitted to De Montfort University Student Paper	<1%

Appendix B

```
importPySimpleGUI as sg
import docx2pdf
importwebbrowser
fromdocx import Document
import pickle
importnumpy as np
import helper
fromdatetime import date


col=helper.col2()
pickled_model = pickle.load(open('model1.pkl', 'rb'))

# Define the layout of the input form

layout = [
   [sg.Column([[sg.Text('Name:'), sg.InputText(key='name', size=(20,
1))],
              [sg.Text('Gender:'), sg.InputCombo(['Male', 'Female'],
size=(20, 1), key='gender')],
              [sg.Text('Age :'), sg.InputText(key='age', size=(20, 1))],
              [sg.Text('Contact No::'), sg.InputText(key='contact', size=(20,
1))],

              [sg.Text('BP:'), sg.InputText(key='bp', size=(20, 1))],
              [sg.Text('Temperature:'), sg.InputText(key='temp', size=(20,
1))],
              [sg.Text('SpO2:'), sg.InputText(key='spo2', size=(20, 1))],

   ]),
sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group_id=i, key=i),
sg.Radio("No", group_id=i,default=True, key=i)] for i in range(0,14)],
element_justification='c'),
sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group_id=i, key=i),
sg.Radio("No", group_id=i,default=True, key=i)] for i in range(14,28)],
```

```
element_justification='c'),
sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group_id=i, key=i),
sg.Radio("No", group_id=i,default=True, key=i)] for i in range(28,42)],
element_justification='c'),

    ]
]

layout.append([sg.Button('Submit')],)
layout.append([sg.Text('Result:', size=(20, 1)), sg.Text('', size=(20, 1),
key='result')])
# Create the PySimpleGUI window with the layout
window = sg.Window('User Input Form', layout)

# Define the path of the docx file to be opened and updated
docx_path = r'C:\Users\jadon\Downloads\health\health\R_copy.docx'

# Loop to keep the window open until user closes or submits the form
while True:
event, values = window.read()

    # If user clicks cancel or closes the window, exit the loop
if event == sg.WINDOW_CLOSED:
break

    # If user clicks submit, display the input values and result in the output
text field, update the docx file, and open the pdf in a browser
if event == 'Submit':
x_find=[]
for i in range(0,42):
if values[i]==True:
x_find.append(1)
else:
x_find.append(0)



        y = np.array(x_find)
```

```
y
y_find=pickled_model.predict(y.reshape(1, -1))
val=str(y_find)
        # Open the docx file and update the values of name, gender, and
temperature
doc = Document(docx_path)
if a in paragraph.text:
paragraph.text = paragraph.text.replace(a,b)
for paragraph in doc.paragraphs:
dsa('P_name ',values['name'])
dsa('Possible_sym',str(helper.get_symptoms(y_find[0])))
dsa('2023-04-28',str(today))
dsa('P_age', values['age'])
dsa('P_gen', values['gender'])
dsa('8976543210', values['contact'])
dsa('100', values['temp'])
dsa('110', values['bp'])
dsa('98', values['spo2'])
dsa('ABCDEF', str(y_find[0]))
        #doc.save('output.pdf')
doc.save('output.docx')
docx2pdf.convert('output.docx','output.pdf')
        #doc.remove('output.docx')
webbrowser.open_new_tab('output.pdf')
# Close the window when the loop exits
window.close()
for paragraph in doc.paragraphs:
dsa('P_name ',values['name'])
dsa('Possible_sym',str(helper.get_symptoms(y_find[0])))
dsa('2023-04-28',str(today))
dsa('P_age', values['age'])
dsa('P_gen', values['gender'])
dsa('8976543210', values['contact'])
dsa('100', values['temp'])
dsa('110', values['bp'])
dsa('98', values['spo2'])
dsa('ABCDEF', str(y_find[0]))
```

Appendix C

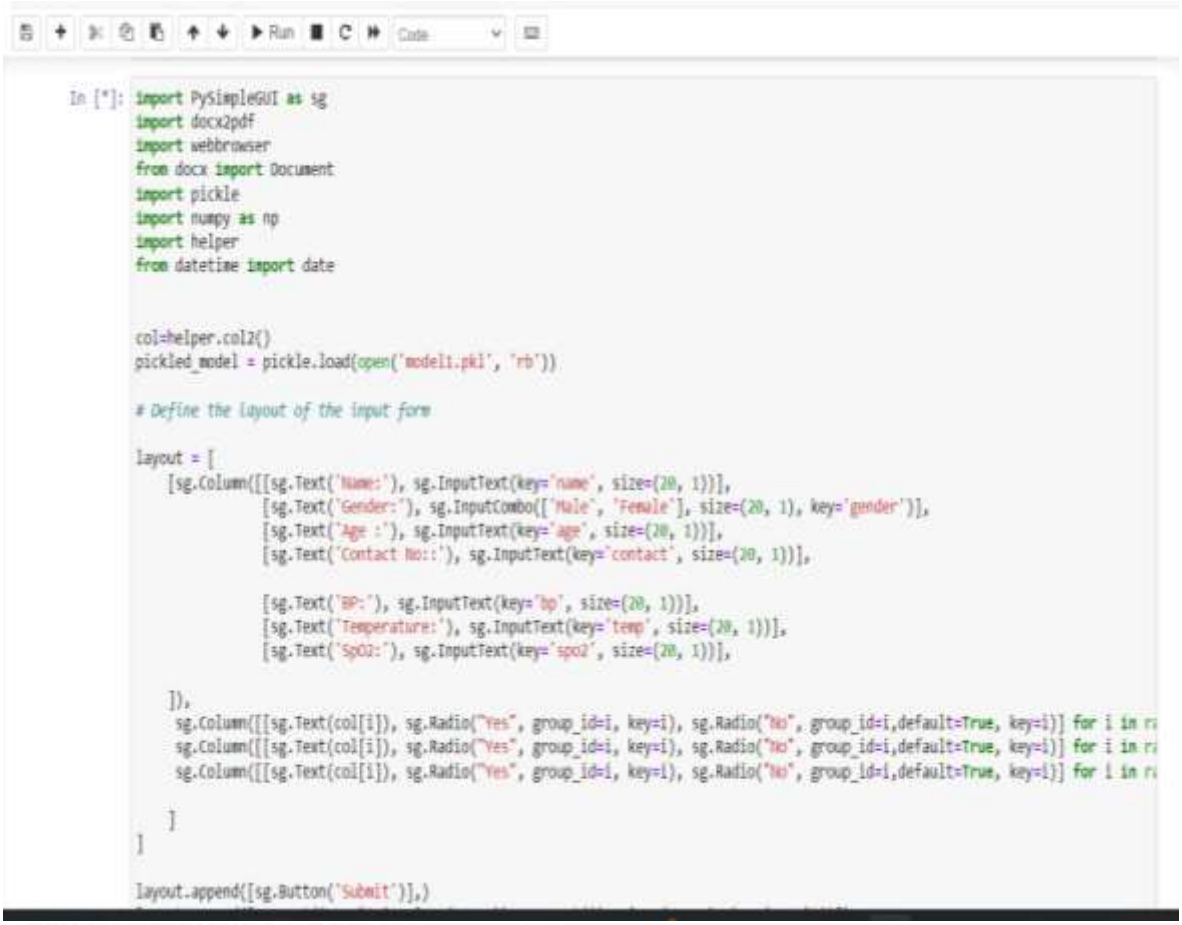
```
col =['itching', 'skin_rash', 'nodal_skin_eruptions',
'continuous_sneezing',
      'shivering', 'chills', 'joint_pain', 'stomach_pain', 'vomiting',
      'burning_micturition', 'spotting_ urination', 'fatigue', 'weight_loss',
      'restlessness', 'lethargy', 'irregular_sugar_level', 'cough',
      'high_fever', 'sweating', 'headache', 'yellowish_skin', 'dark_urine',
      'nausea', 'loss_of_appetite', 'pain_behind_the_eyes', 'back_pain',
      'constipation', 'abdominal_pain', 'diarrhoea', 'mild_fever',
      'swelling_of_stomach', 'swelled_lymph_nodes', 'malaise',
      'blurred_and_distorted_vision', 'phlegm', 'chest_pain', 'obesity',
      'loss_of_smell', 'toxic_look_(typhos)', 'muscle_pain',
      'red_spots_over_body', 'scurring']

def len_col():
   return len(col)

def col2():
   return col

def get_symptoms(disease):
   st={
   "Acne": [
      "Blackheads, whiteheads, and pimples on the skin",
      "Oily skin and/or skin that feels tender to the touch",
      "Redness and inflammation around the affected area",
      "Scarring or hyperpigmentation (darkening of the skin) after the
acne has healed",
      "Painful cysts or nodules under the skin"
   ],

   "Alcoholic hepatitis": [
      "Abdominal pain and tenderness",
      "Jaundice (yellowing of the skin and eyes)",
      "Nausea and vomiting",
      "Loss of appetite",
      "Fatigue and weakness"
```

```
],

"Allergy": [
    "Itchy, watery eyes",
    "Runny or stuffy nose",
    "Sneezing and coughing",
    "Skin rash or hives",
    "Swelling of the face, lips, or tongue"
],

"Chicken pox": [
    "Red, itchy rash that turns into fluid-filled blisters",
    "Fatigue and weakness",
    "Fever",
    "Headache",
    "Loss of appetite"
],

"Common Cold": [
    "Runny or stuffy nose",
    "Sore throat",
    "Coughing",
    "Sneezing",
    "Fatigue and weakness"
],

"Dengue": [
    "High fever",
    "Severe headache",
    "Pain behind the eyes",
    "Joint and muscle pain",
    "Rash"
],

"Diabetes": [
    "Excessive thirst and hunger",
    "Frequent urination",
    "Fatigue and weakness",
```

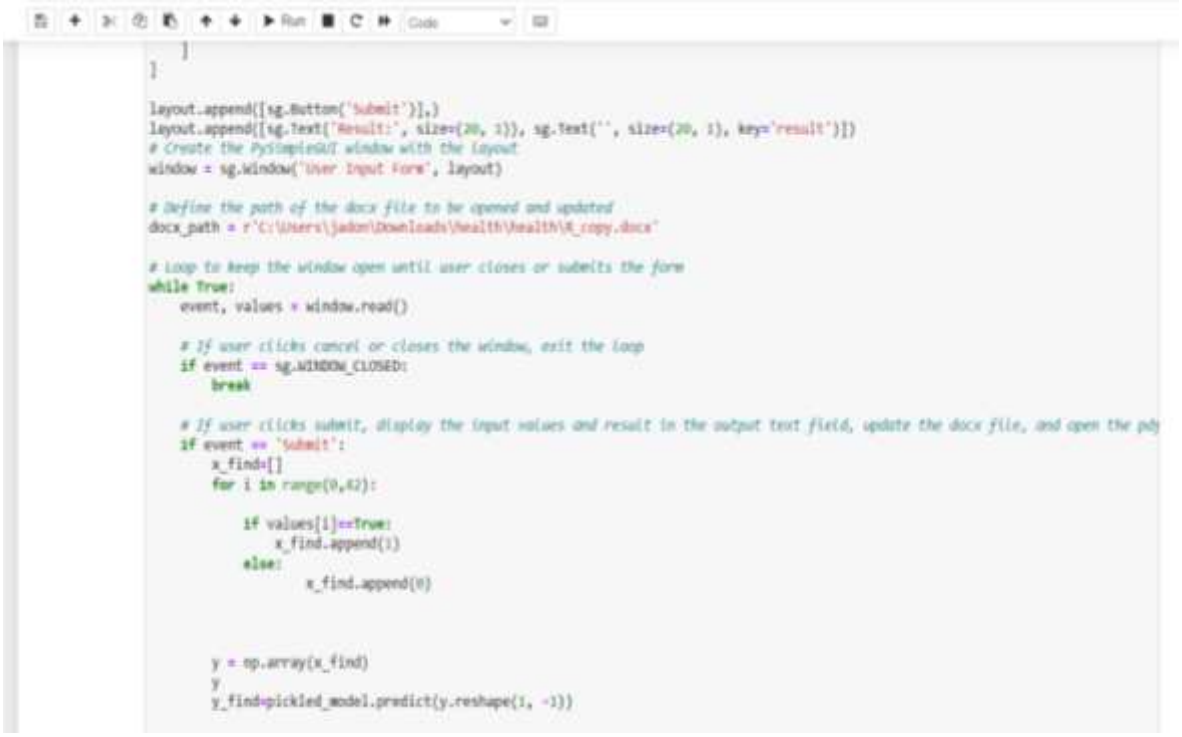
```
    "Blurred vision",
    "Slow-healing cuts or wounds"
],
"Malaria": [
    "Fever",
    "Chills",
    "Headache",
    "Fatigue",
    "Muscle pain or body aches"
],

"Typhoid": [
    "High fever (over 100.4°F or 38°C)",
    "Weakness and fatigue",
    "Abdominal pain and cramps",
    "Headache",
    "Diarrhea or constipation"
],

"Diabetes ": [
    "Excessive thirst and hunger",
    "Frequent urination",
    "Fatigue and weakness",
    "Blurred vision",
    "Slow-healing cuts or wounds"
],

"Drug Reaction": [
    "Skin rash or hives",
    "Itching",
    "Swelling of the face, lips, or tongue",
    "Difficulty breathing",
    "Abdominal pain and vomiting"
],

"Fungal infection": [
    "Itchy, red, and/or scaly skin rash",
    "Skin lesions or blisters",
```

```
        "Nail discoloration or thickening",
        "Hair loss or bald patches"
    ],

    "Jaundice": [
        "Yellowing of the skin and eyes",
        "Dark urine",
        "Pale stools",
        "Fatigue or weakness",
        "Abdominal pain"
    ]

}
```

Appendix D

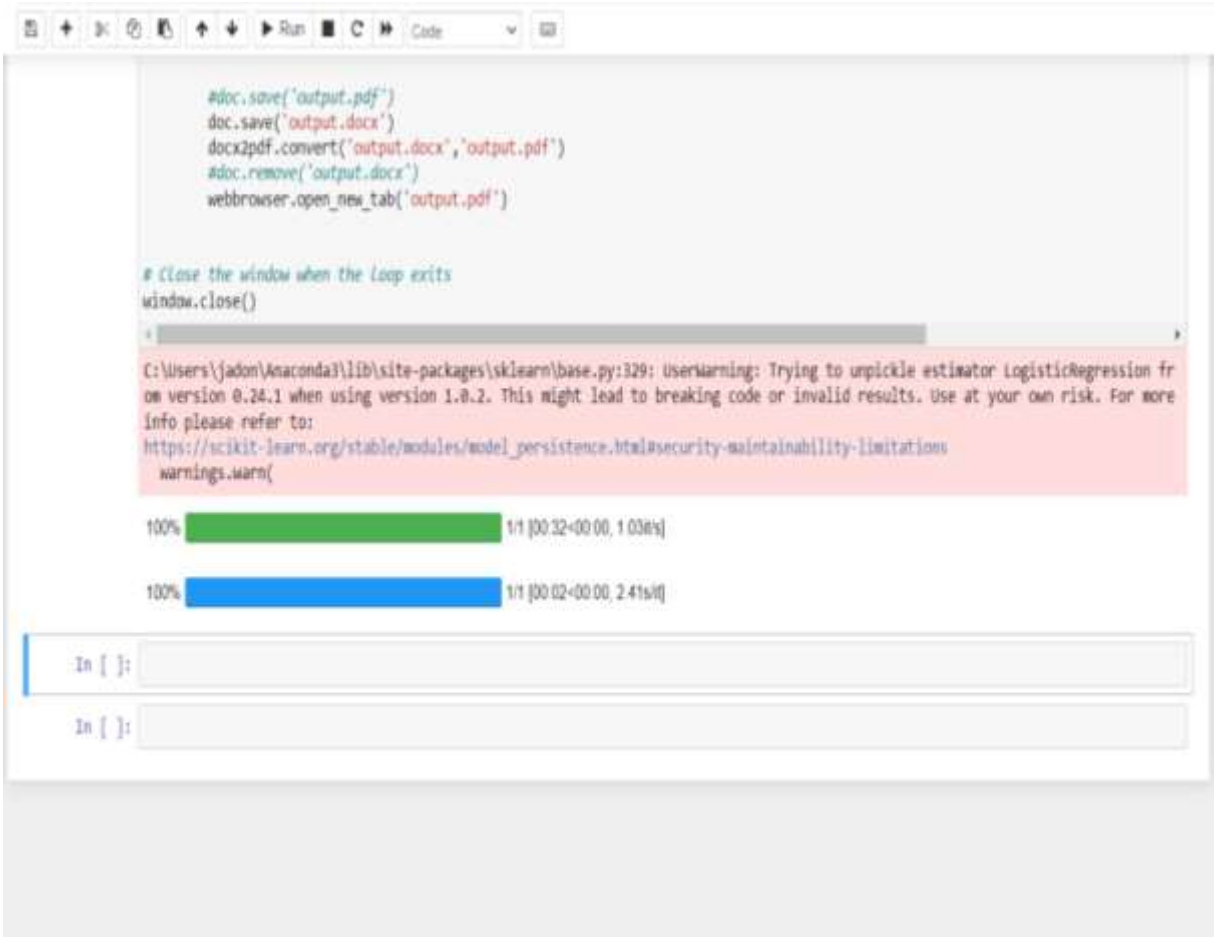


```
In [*]: import PySimpleGUI as sg
import docx2pdf
import webbrowser
from docx import Document
import pickle
import numpy as np
import helper
from datetime import date


col=helper.col2()
pickled_model = pickle.load(open('model1.pkl', 'rb'))

# Define the layout of the input form

layout = [
    [sg.Column([[sg.Text('Name:'), sg.InputText(key='name', size=(20, 1))],
                [sg.Text('Gender:'), sg.InputCombo(['Male', 'Female'], size=(20, 1), key='gender')],
                [sg.Text('Age :'), sg.InputText(key='age', size=(20, 1))],
                [sg.Text('Contact No::'), sg.InputText(key='contact', size=(20, 1))],

                [sg.Text('BP:'), sg.InputText(key='bp', size=(20, 1))],
                [sg.Text('Temperature:'), sg.InputText(key='temp', size=(20, 1))],
                [sg.Text('SpO2:'), sg.InputText(key='spo2', size=(20, 1))],

    ]),
     sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group_id=i, key=i), sg.Radio("No", group_id=i,default=True, key=i)] for i in r
     sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group_id=i, key=i), sg.Radio("No", group_id=i,default=True, key=i)] for i in r
     sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group_id=i, key=i), sg.Radio("No", group_id=i,default=True, key=i)] for i in r

    ]
]

layout.append([sg.Button('Submit')],)
```



```
    ]
]

layout.append([sg.Button('Submit')],)
layout.append([sg.Text('Result:', size=(20, 1)), sg.Text('', size=(20, 1), key='result')])
# Create the PySimpleGUI window with the layout
window = sg.Window('User Input Form', layout)

# Define the path of the docx file to be opened and updated
docx_path = r'C:\Users\jadon\Downloads\health\health\R_copy.docx'

# Loop to keep the window open until user closes or submits the form
while True:
    event, values = window.read()

    # If user clicks cancel or closes the window, exit the loop
    if event == sg.WINDOW_CLOSED:
        break

    # If user clicks submit, display the input values and result in the output text field, update the docx file, and open the pdf
    if event == 'Submit':
        x_find=[]
        for i in range(0,42):

            if values[i]==True:
                x_find.append(1)
            else:
                    x_find.append(0)



        y = np.array(x_find)
        y
        y_find=pickled_model.predict(y.reshape(1, -1))
```

Fig 18. Manual_a



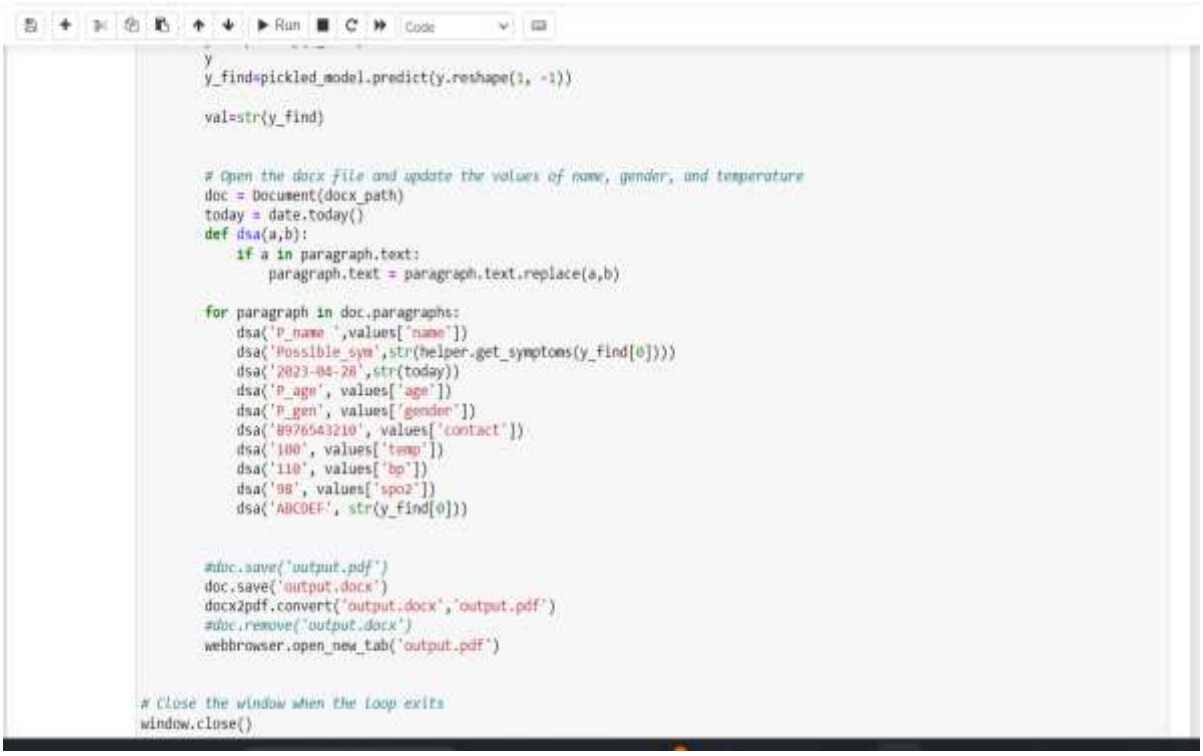


Fig 19. Manual_b

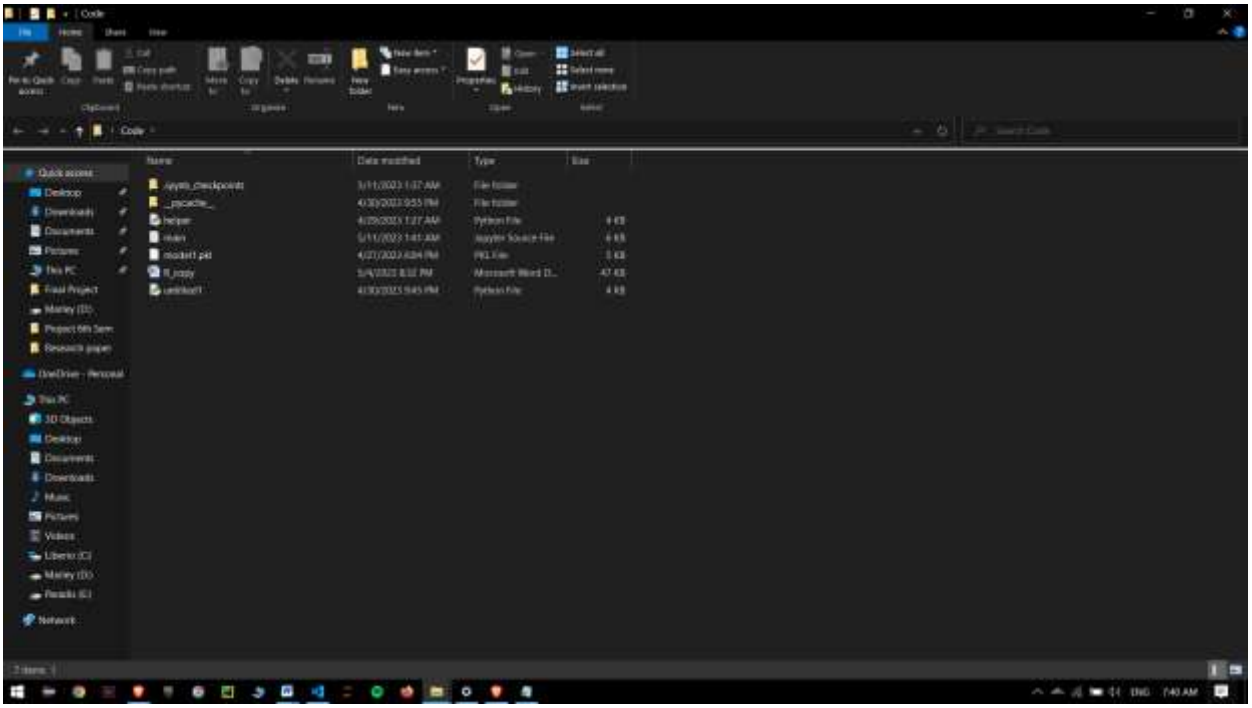


Fig 20. Manual_c