

MULTI DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUES

*Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

K. SATHVIKA	(21UECS0318)	(VTU 19937)
M. SIRI CHANDANA	(21UECS0353)	(VTU 19966)
K. AKASH	(21UECS0317)	(VTU 19952)

*Under the guidance of
Dr. T. Kamaleshwar, M. Tech, Ph. D.,
Assistant Professor*



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

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

January, 2024

MULTI DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUES

*Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

K.SATHVIKA	(21UECS0318)	(VTU 19937)
M.SIRI CHANDANA	(21UECS0353)	(VTU 19966)
K.AKASH	(21UECS0317)	(VTU 19952)

*Under the guidance of
Dr. T. Kamaleshwar, M. Tech, Ph. D.,
Assistant Professor*



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

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

January, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "MULTI DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUES" by "K. SATHVIKA (21UECS0318), M. SIRI CHANDANA (21UECS0353), K. AKASH (21UECS0317)" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor

Dr. T. Kamaleshwar

Assistant Professor

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

January, 2024

Signature of Head of the Department

Dr. M. S. Muralidhar

Associate Professor & Head

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

January, 2024

Signature of the Dean

Dr. V. Srinivasa Rao

Professor & Dean

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

January, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(K. SATHVIKA)

Date: / /

(M. SIRI CHANDANA)

Date: / /

(K. AKASH)

Date: / /

APPROVAL SHEET

This project report entitled "MULTI DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUES" by K. SATHVIKA (21UECS0318), M. SIRI CHANDANA (21UECS0353), K. AKASH (21UECS0317) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners

Supervisor

Dr. T. KAMALESHWAR, M. Tech, Ph. D.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN**, for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr. M.S. MURALIDHAR, M. E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Dr. T. KAMALESHWAR, M. Tech, Ph. D.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M. Tech., Ms.C. SHYAMALA KUMARI, M. E., Mr. SHARAD SHANDHI RAVI, M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

K. SATHVIKA	(21UECS0318)
M. SIRI CHANDANA	(21UECS0353)
K. AKASH	(21UECS0317)

ABSTRACT

The proposed project aims to develop a web application that will enable individuals to self-test for Heart Disease, Parkinson's Disease, and Diabetes. By giving the values of their blood pressure, fasting blood sugar, maximum heart rate, and calcium values, the application will use machine learning algorithms to predict the likelihood of each disease accurately. This will provide users with a reliable and convenient way of self-testing for these diseases without the need to visit a doctor, enhancing health care accessibility. The web application will be designed with multiple pages, each catering to a specific disease, allowing users to test for the disease they are concerned about. The use of SVM and logistic regression will ensure that the predictions are highly accurate with 86% really great, enabling users to make informed decisions about their health and tells whether the person is having the disease or not. The proposed system will encourage users to share their anonymized data voluntarily. This collective data pool can be utilized for broader health research initiatives, potentially leading to new insights and advancements in preventive healthcare on a larger scale. Through these collaborative efforts, the web application not only serves individual users but also contributes to the collective pursuit of improved public health outcomes. In addition to its primary focus on disease prediction, the application will also provide valuable insights and educational resources. Users will have access to detailed explanations about the significance of each input parameter, fostering health literacy and empowering them to understand the factors influencing their health outcomes. By offering quick, easy, and affordable disease testing to people across different age groups and geographical locations, the proposed system will enhance the accessibility and affordability of healthcare services.

Keywords:

SVM, Deep Learning, Diabetes, Heart Stroke, Parkinson's, Prediction.

LIST OF FIGURES

4.1	Architecture of Multi-Disease Prediction	14
4.2	Data Flow Diagram of Prediction	15
4.3	Usecase Diagram of Web Application	16
4.4	Class Diagram of Prediction Model	17
4.5	Sequence Diagram of Web application	18
4.6	Activity Diagram of Web Application	19
5.1	User Interface of web application	26
5.2	Checking the Disease	27
5.3	Prediction of Models	30
6.1	Diabetics Disease Prediction	38
6.2	Heart Disease Prediction	39
6.3	Parkinson's Disease Prediction	40
9.1	Poster Presentation	48

LIST OF ACRONYMS AND ABBREVIATIONS

S.NO	ABBREVIATION	DEFINITION
1	AM	Accuracy Matrix
2	BMI	Body Mass Index
3	BPF	Build Pickle File
4	DDP	Diabetes Diseases Prediction
5	FS	Feature Selection
6	GDPR	General Data Protection Regulation
7	HDP	Heart Disease Prediction
8	ML	Machine Learning
9	NN	Neural Network
10	PDP	Parkinson's Disease Prediction
12	SVM	Support Vector Machine

TABLE OF CONTENTS

	Page.No
ABSTRACT	v
LIST OF FIGURES	vi
LIST OF ACRONYMS AND ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Aim of the Project	2
1.3 Project Domain	2
1.4 Scope of the Project	3
2 LITERATURE REVIEW	5
3 PROJECT DESCRIPTION	9
3.1 Existing System	9
3.2 Proposed System	9
3.3 Feasibility Study	11
3.3.1 Economic Feasibility	11
3.3.2 Technical Feasibility	11
3.3.3 Social Feasibility	11
3.4 System Specification	12
3.4.1 Hardware Specification	12
3.4.2 Software Specification	12
3.4.3 Standards and Policies	13
4 METHODOLOGY	14
4.1 General Architecture	14
4.2 Design Phase	15
4.2.1 Data Flow Diagram	15
4.2.2 Use Case Diagram	16
4.2.3 Class Diagram	17

4.2.4	Sequence Diagram	18
4.2.5	Activity Diagram	19
4.3	Algorithm & Pseudo Code	20
4.3.1	Support Vector Machine Algorithm	20
4.3.2	Pseudo Code	20
4.4	Module Description	21
4.4.1	Heart disease Prediction Module:	21
4.4.2	Diabetes disease Prediction Module:	22
4.4.3	Parkinson's disease Prediction Module:	22
4.5	Steps to execute/run/implement the project	24
4.5.1	Step 1: Access the Web Application	24
4.5.2	Step 2: User Login	24
4.5.3	Step 3: Homepage/ Dashboard	24
4.5.4	Step 4: Select Disease of Interest	24
4.5.5	Step 5: Input Patient Data	24
4.5.6	Step 6: Submit Data	24
4.5.7	Step 7: Prediction Processing	24
4.5.8	Step 8: View prediction Results	25
4.5.9	Step 9: Logout	25

5 IMPLEMENTATION AND TESTING 26

5.1	Input and Output	26
5.1.1	Input Design	26
5.1.2	Output Design	27
5.2	Testing	28
5.3	Types of Testing	28
5.3.1	Unit testing	28
5.3.2	Integration testing	29
5.3.3	Test Result	30

6 RESULTS AND DISCUSSIONS 31

6.1	Efficiency of the Proposed System	31
6.2	Comparison of Existing and Proposed System	31
6.2.1	Existing system:(K-nearest neighbors)	31
6.2.2	Proposed system:(Logistic Regression)	32
6.3	Sample Code	32

7	CONCLUSION AND FUTURE ENHANCEMENTS	41
7.1	Conclusion	41
7.2	Future Enhancements	41
8	PLAGIARISM REPORT	42
9	SOURCE CODE & POSTER PRESENTATION	43
9.1	Source Code	43
9.2	Poster Presentation	48
	References	49

Chapter 1

INTRODUCTION

1.1 Introduction

Early disease identification is essential for efficient management and therapy. Better health outcomes, such as lower rates of morbidity and mortality, can result from early diagnosis. A machine learning-based web tool that can effectively forecast Diabetes, Parkinson's Disease, and Heart Disease to meet this requirement. Users will receive prompt and precise disease predictions from this application after it analyses input data including blood pressure, fasting blood sugar, maximum heart rate, and calcium readings. The suggested method intends to improve healthcare accessibility by enabling anyone to rapidly and correctly self-test for diseases without seeing a doctor.

Each page of the online application will focus on a different ailment, making it simple for users to test for the condition they are worried about. To assure the system's correctness and dependability, data will be gathered, preprocessed, features selected, tested, and validated. To make sure the collected data is accurate and pertinent to the diseases being forecasted, preprocessing will be done on it. To determine the key characteristics of each disease, feature selection will be used. The system's accuracy in making predictions and the consistency of the outcomes will be confirmed during the testing and validation phases.

The suggested machine learning-based online application will offer a trustworthy, usable, and effective way to anticipate diseases. It has the potential to increase accessibility to healthcare and improve early detection, enabling people to take charge of their health without going to the doctor. The initiative offers a fascinating chance to use contemporary technology to one of the most important healthcare concerns of our day.

1.2 Aim of the Project

The primary goal of this project is to create a web application that can precisely identify whether a person has Diabetes, Parkinson's Disease, or Heart Disease. Based on input data such as blood pressure, fbs, maximal heart rate, and calcium readings, the system will apply machine learning algorithms to forecast the possibility of certain conditions. By creating such a system, individuals can test for various diseases on their own without seeing a doctor, improving healthcare accessibility and facilitating quick, simple, and economical disease testing. The project aims to create a machine learning model for multi-disease prediction, focusing on early detection and personalized healthcare. By integrating diverse health datasets, including medical records and lifestyle information, the project seeks to develop a comprehensive predictive model. Feature selection and engineering will be employed to identify key predictors for multiple diseases, and the model will be optimized through rigorous training and cross-validation. The emphasis on interpretability and explainability ensures transparency in predictions, fostering trust among healthcare professionals. Real-time prediction capability, scalability, and adaptability to varying data volumes will be key design considerations, ultimately contributing to advanced healthcare solutions and improved patient outcomes. The project's goal is to make healthcare more accessible by offering a practical and affordable method of disease prediction. Because of the application's user-friendly interface, people would be able to self-test for diseases without going to the doctor, empowering them to take control of their health. The project's success will help the healthcare sector advance and stimulate additional machine learning and healthcare research. The proposed project is a big step towards tackling one of the most pressing healthcare concerns of our day and providing accessible treatment to everyone, regardless of their financial situation.

1.3 Project Domain

The suggested machine learning-based web application for diabetes, Parkinson's disease, and heart disease prediction belongs to the healthcare and medical diagnosis categories. As it allows people to self-test for various diseases without seeing a doctor, it represents a huge improvement in the field of healthcare. In order to estimate a person's chance of getting any of the three diseases, the project uses machine learning algorithms to examine input data such as blood pressure, fasting blood sugar,

maximal heart rate, and calcium readings. Through the processes of data collection, preprocessing, feature selection, testing, and validation, the application's accuracy and dependability are guaranteed.

As a result of the project's ability to increase disease diagnosis rates and provide people more control over their health, the healthcare sector will be significantly affected. People find it easier to understand their health state and take the required precautions to prevent or manage these diseases because to the application's accessibility and user-friendly interface. In addition, the application's cost-effectiveness is a key component in offering everyone, regardless of their financial situation, access to inexpensive healthcare.

The project's scope includes artificial intelligence and machine learning as well. It illustrates how machine learning systems may correctly forecast diseases via data analysis. To maximise the performance of the application, the project entails the collecting and processing of massive datasets, feature selection, and model selection. Due to the project's success, there will be more machine learning and healthcare research, which will result in more creative solutions to healthcare issues. In conclusion, the proposed project relates to the fields of artificial intelligence, machine learning, healthcare, and medical diagnosis.

1.4 Scope of the Project

The goal of this project is to create a machine learning-based web application, by examining significant input data, can reliably forecast diabetes, Parkinson's disease, and heart disease. To guarantee its correctness and dependability, the application will go through data preprocessing, feature selection, testing, and validation. The project's goal is to make healthcare more accessible by offering a practical and affordable method of disease prediction. Because of the application's user-friendly interface, people would be able to self-test for diseases without going to the doctor, empowering them to take control of their health. The proposed project is a big step towards tackling one of the most pressing healthcare concerns of our day and providing accessible treatment to everyone, regardless of their financial situation. The scope involves the integration of 4 medical records, genetic information, and lifestyle data to train the SVM model, allowing for a comprehensive analysis of

potential disease correlations. Feature selection and engineering will be utilized to identify pertinent predictors, enhancing the ability to discern patterns indicative of various diseases.

The project focuses on optimizing the model's performance through rigorous training and cross-validation techniques, ensuring accuracy and reliability. Emphasis will be placed on developing an interpretable and user-friendly interface for healthcare professionals, enabling them to leverage SVM predictions in real-time scenarios. Ethical considerations, including privacy safeguards, will be integral to the project design. Collaboration with healthcare institutions will facilitate validation through clinical trials, providing real-world confirmation of the model's efficacy in multi-disease prediction. The scalability of the system to accommodate growing data volumes underscores its adaptability for future enhancements and widespread applicability.

Chapter 2

LITERATURE REVIEW

[1] Ederson,.et al.,(2021) performed a review of various machine learning techniques for predicting diabetes, analyzing the performance of different algorithms on various datasets. The authors found that decision tree, support vector machine, and logistic regression were the most effective techniques for diabetes prediction, as they achieved high accuracy rates and exhibited robust performance across different datasets. The study's results indicate the potential of machine learning in providing accurate and efficient diabetes prediction, which can aid in the timely diagnosis and management of the disease. The authors also noted that feature selection and data preprocessing were important factors in achieving optimal performance for the machine learning models. The study's findings highlight the importance of continued research in this area to develop more advanced and reliable machine learning models for diabetes prediction and management.

[2] Hsieh,.et al.,(2022)suggested an accurate artificial neural network prediction model for cardiac disease. The model was trained and tested using a dataset of 303 patients and 14 characteristics. The outcomes demonstrated the suggested model's potential for use in clinical practise, with an accuracy rate of 87.13 percent. Machine learning techniques are frequently used in the healthcare sector to predict fatal conditions. The classification models of Logistic regression, K-nearest neighbour, Support vector machine, Decision tree, and Random Forest to develop and evaluate the performance of a recommended system and a conventional system that predicts heart disease. The suggested system aided in tuning the hyperparameters of the five specified classification algorithms utilizing the grid search technique. The study's main topic was the performance of the heart disease prediction system. The study demonstrated that the performance of prediction models could be improved by using the hyperparameter tuning model.

[3] Jindal,.et al.,(2021) performed comprehensive review of the literature on the application of machine learning techniques for the diagnosis of Parkinson's disease.

The authors analyzed various studies that employed machine learning algorithms for Parkinson's disease diagnosis and discussed their effectiveness and limitations. They found that the most commonly used machine learning algorithms for Parkinson's disease diagnosis were support vector machines, decision trees, and artificial neural networks. The authors also highlighted the need for further research to improve the accuracy and reliability of the models. They recommended the use of larger datasets, feature selection techniques, and deep learning models for future studies. Overall, this review provides valuable insights into the application of machine learning techniques for the diagnosis of Parkinson's disease and suggests avenues for future research to improve the accuracy and reliability of these techniques.

[4] Karim,.et al.,(2020) conducted a comprehensive literature review on the application of machine learning techniques in the diagnosis of diabetes mellitus. The authors analyzed various studies that employed different algorithms and approaches and compared their performance in diagnosing diabetes. They found that decision trees, random forests, and support vector machines were the most effective algorithms for predicting diabetes. Moreover, the review identified the need for more standardized datasets and more robust evaluation metrics to improve the accuracy and reliability of the models. The review also discussed the potential of combining different machine learning techniques to enhance the performance of the models. Overall, the study provides valuable insights into the use of machine learning in diabetes diagnosis and highlights the importance of further research in this area.

[5] Kumar,.et al.,(2022) created an effective method for Parkinson's disease identification. To train and evaluate the model, the study employed a dataset of 195 patients and 22 characteristics. The findings demonstrated that the suggested method had a high accuracy of 97.1 percent, highlighting its potential for Parkinson's disease early diagnosis and therapy. The study is important because it shows how machine learning algorithms may be used to accurately diagnose Parkinson's disease. The proposed approach's high degree of accuracy suggests that it may help with early disease detection and treatment, which are essential for efficient management. Early detection can assist medical professionals in providing prompt care and lessen the toll that the disease has on patients and their families. The suggested strategy may also help in the development of more effective and precise diagnostic techniques, which would improve patient outcomes. Overall, the study's findings highlight

the need for more research in this field and offer insightful information about the potential of machine learning algorithms in healthcare.

[6] Mohammadzadeh,.et al.,(2020) conducted a comprehensive review of the literature to provide an overview of various algorithms and approaches used for diabetes diagnosis and risk prediction. The authors analyzed the performance of different algorithms on various datasets and highlighted their strengths and limitations. The review discusses the potential benefits of machine learning-based methods in diabetes diagnosis, such as improved accuracy and early detection. However, the authors also highlighted the challenges associated with the use of these methods, such as the need for large datasets and the interpretability of the results. Overall, the review provides valuable insights into the current state of machine learning-based methods for diabetes diagnosis and risk prediction, as well as the challenges and opportunities in this field.

[7] Tawfik,.et al.,(2021) reviewed of various machine learning techniques for predicting diabetes. The authors analyzed the performance of different algorithms on various datasets and found that decision tree, support vector machine, and logistic regression were the most effective techniques. The study also highlighted the importance of feature selection and data preprocessing in enhancing the performance of the algorithms. The findings of the study can potentially contribute to the development of more accurate and efficient diabetes prediction models, which can aid in the early detection and management of the disease. The study's results also demonstrate the potential of machine learning in healthcare and highlight the need for further research to develop more advanced and reliable models for predicting diabetes. Overall, the study provides valuable insights into the potential of machine learning in healthcare and highlights the importance of algorithm selection and data preparation in developing accurate prediction model.

[8] Wu,.et al.,(2021) conducted research on heart disease prediction using an improved CNN algorithm. The study utilized a dataset of 303 patients and 14 features to train and test the model. The results demonstrated that the proposed algorithm achieved an accuracy of 93.53 percent, indicating its potential for clinical use. The study's findings are significant as they provide valuable insights into the potential of machine learning algorithms in accurately predicting heart disease, a

critical health condition that affects a large population globally. The high accuracy achieved by the proposed algorithm indicates that it can potentially contribute to the early diagnosis and treatment of the disease, leading to better healthcare outcomes for patients. The study highlights the need for further research in this area to develop more advanced and reliable prediction models for heart disease, which can aid clinicians in providing timely and effective treatment.

[9] Yassin, et al., (2021) reviewed the application of machine learning techniques in the diagnosis of heart disease, providing an overview of different algorithms used in the literature and their strengths and weaknesses. The authors highlighted the importance of accurate and timely diagnosis of heart disease and discussed how machine learning algorithms can aid in this process. They also reviewed various studies that used machine learning techniques, including support vector machines, random forest, and artificial neural networks, among others, for heart disease diagnosis. The authors discussed the strengths and weaknesses of each technique and emphasized the importance of appropriate feature selection, data preprocessing, and model selection in achieving accurate diagnosis. The review concludes that machine learning algorithms have great potential in the diagnosis of heart disease and could contribute to improving healthcare accessibility and quality for patients with heart disease. The authors call for continued research in this area to develop more advanced and accurate machine learning models for heart disease diagnosis.

[10] Zheng, et al., (2021) established a systematic review of deep learning for the diagnosis of Parkinson's disease. The authors analyzed several studies that employed deep learning techniques to detect Parkinson's disease and found that convolutional neural networks and recurrent neural networks were effective in accurately identifying the disease. Study finds are significant as they indicate the potential of deep learning algorithms in providing accurate and reliable diagnosis of Parkinson's disease, which can aid in timely treatment and management of the disease. However, the authors also suggested that further research is needed to improve the accuracy and reliability of the models, as some studies exhibited limitations in terms of dataset size and other factors. The study results highlight of the need for continued research in this area to develop more advanced and reliable deep learning models for the early detection and management of Parkinson's disease.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

K-nearest neighbors algorithm is the current existing system for the prediction of diseases, it is the algorithm which produces low accuracy than compared to the proposed model.

Low accuracy: Disease identification may be hampered by the inability of two-layer neural networks to capture complicated correlations between features and produced accuracy around 75 percent. This might result in false positives or false negatives, which can have catastrophic consequences for patients.

Restricted scalability: It is challenging to scale the system for usage in large healthcare organisations or populations since two-layer neural networks are constrained in their ability to handle large datasets with many features.

Overfitting: The two-layer neural network may be susceptible to overfitting, which occurs when the model is overly tuned for the training set of data but performs badly on new, untrained data.

Low interpretability: Neural networks are frequently referred to as "black box" models, which makes it challenging to comprehend how the model generates predictions.

3.2 Proposed System

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. In the context of disease prediction using SVM, logistic regression is used to build a binary

classification model that predicts whether a patient has a particular disease or not based on the input features.

During the training phase, the logistic regression model learns the optimal weights for each input feature and bias term through gradient descent optimization, which 10 minimizes the difference between the predicted probability and the actual label. Once the model is trained, it can be used to make predictions on new medical images by extracting the relevant features using SVM and feeding them into the logistic regression model.

The combination of SVM and logistic regression provides an effective approach for disease prediction using medical images, as it can learn and extract relevant features from the input data and use them to make accurate predictions Enhanced accuracy: The use of a Support Vector Machine enables more complex pattern and produced accuracy of 86 % which is more compared to the existing system, which can enhance the accuracy of disease identification.

Faster diagnosis: The system's ability to automate the disease identification process can result in faster diagnoses, which can lead to quicker treatment and possibly better health outcomes.

Efficiency gain: When there are large amounts of data to process, the use of machine learning algorithms can improve disease identification efficiency by a significant amount.

Cost-effective: By identifying diseases without the use of pricey laboratory tests, machine learning algorithms have the potential to reduce costs for both patients and healthcare providers.

Scalability: The system is easily adaptable to variations in the number of patients, both up and down.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The cost and advantages of implementing a multi-disease identification application utilising a two-layered neural network can be used to assess its economic viability. The advantages of such an application may include increased diagnostic speed and accuracy, which may result in better health outcomes and cost savings from early diagnosis and treatment. Reduced expenses for providing healthcare can also be a result of the scalability and efficiency of such a system. The creation and upkeep of the software, hardware infrastructure, and the instruction of medical experts in its use are all expenses related to the installation of such a system. Healthcare organisations, hospitals, or governments may need to make large investments to cover these costs.

3.3.2 Technical Feasibility

The availability of data, the calibre of the data, and the computing resources required for the creation and deployment of the application are just a few of the variables that affect whether a two-layered neural network can be used to diagnose several diseases. The availability and calibre of the data necessary for the neural network to precisely diagnose diseases is one of the main technical obstacles. This information must be thorough, trustworthy, and current, which might be difficult depending on the disease and region. The computational resources needed for the application's development and deployment provide another difficulty. It takes a lot of processing resources to train a two-layered neural network, which can be expensive and time-consuming. Moreover, the application's deployment must be planned to handle.

3.3.3 Social Feasibility

The potential effects on both people and society at large are considered in terms of the social viability of a multi-disease identification application employing a two-layered neural network. Depending on how it is used, the application may have both beneficial and negative effects. On the plus side, this kind of application, especially in rural or underserved areas, can assist in enhancing access to healthcare by delivering more precise and quicker diagnosis. Better health outcomes and lower healthcare

expenditures may result from this. Additionally, the application of artificial intelligence and machine learning in healthcare can foster trust in the medical diagnostic process as well as confidence in the healthcare system. Nonetheless, there may be unfavourable effects to take into account. For instance, developing such an application might need.

3.4 System Specification

3.4.1 Hardware Specification

- CPU
- GPU
- RAM :16GB or higher
- Network connectivity
- Server

3.4.2 Software Specification

- Operating system: windows 7 or higher
- Programming language:Python
- Database
- Web server:Apache or Nginx
- Version control

3.4.3 Standards and Policies

Anaconda Prompt

Anaconda prompt is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations and narrative text. It can be used for data cleaning and transformation, statistical modeling, machine learning.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

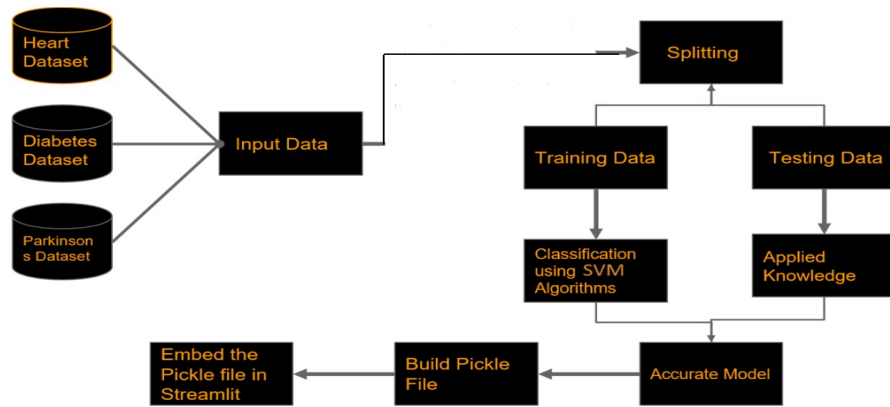


Figure 4.1: Architecture of Multi-Disease Prediction

In figure 4.1, the architecture for the proposed project includes a web-based user interface, machine learning algorithms, and a database. The web-based interface will allow users to input their health data, including blood pressure, fbs, maximum heart rate, and calcium values, for disease prediction. The user inputs will be stored in a database for easy retrieval and analysis. The machine learning algorithms will be employed to predict the likelihood of heart disease, Parkinson's disease, and diabetes based on the input data. The algorithms will be trained on a large dataset of health records to achieve maximum accuracy in disease prediction. The system will provide users with the predicted disease likelihood and relevant information about the disease, enabling them to make informed decisions about their health.

4.2 Design Phase

4.2.1 Data Flow Diagram

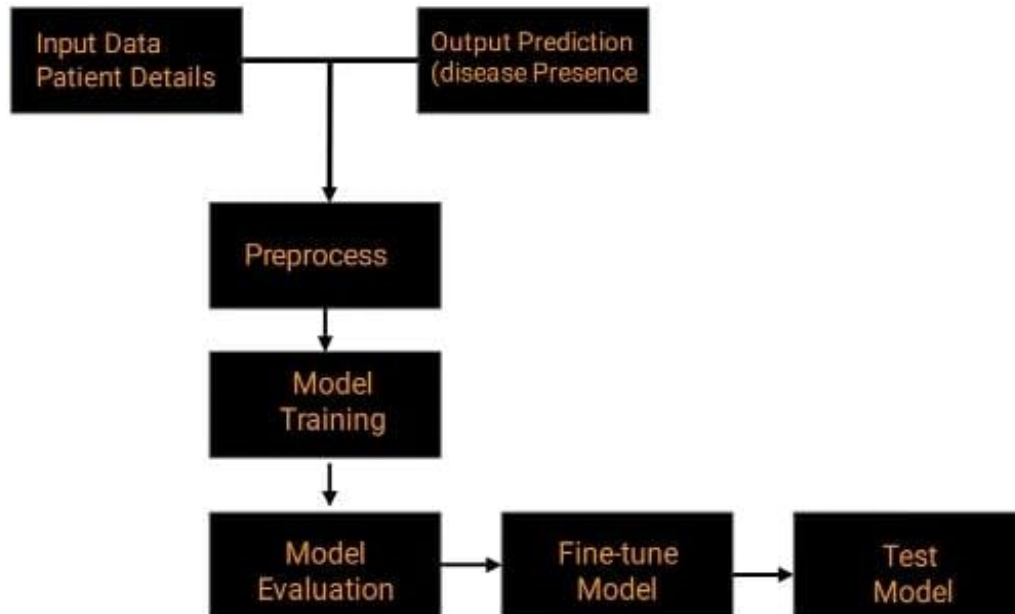


Figure 4.2: Data Flow Diagram of Prediction

In figure 4.2, the Data Flow Diagram (DFD) for this project outlines the flow of data within the system. The diagram consists of four main components: the data source, the data processing unit, the data storage unit, and the user interface. The data source component includes the various input fields on the user interface where the user enters their health data, such as blood pressure, fbs, maximum heart rate, and calcium values. This data is then passed to the data processing unit where it is analyzed using machine learning algorithms. The results are then stored in the data storage unit. The user interface component allows users to interact with the system and view the results of the analysis. We illustrates how the system processes and stores data. The input data is received through the user interface, which is then processed by the data processing unit. The processed data is then stored in the data storage unit for future reference.

4.2.2 Use Case Diagram

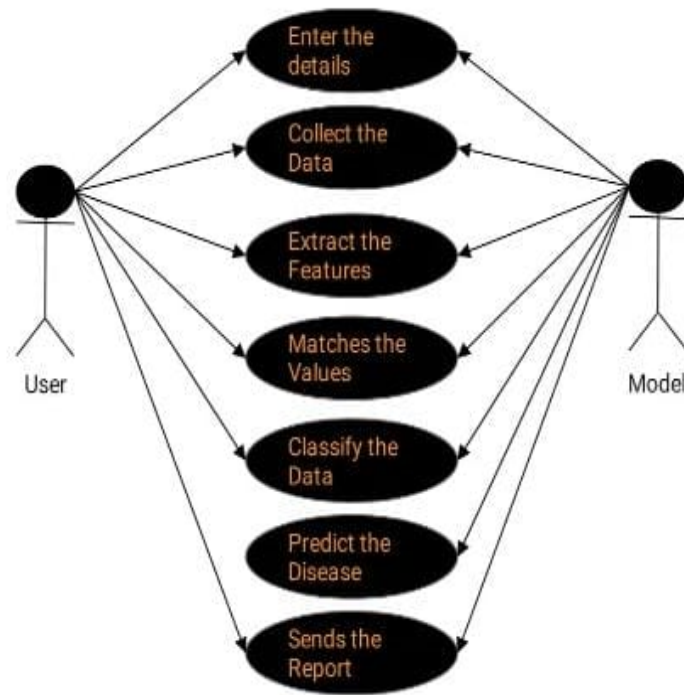


Figure 4.3: Usecase Diagram of Web Application

In figure 4.5, from the project, the primary actors are the users and the system. The use case diagram provides a high-level view of the functionalities that the system will provide to the users. The diagram shows the different use cases, including adding patient data, predicting the likelihood of heart disease, Parkinson's disease, and diabetes, and displaying the prediction results. The use case diagram helps to ensure that all necessary functionalities are captured, and the system meets the user requirements. In this project, the use case diagram also helps to identify the different roles and their interactions with the system. The diagram includes the various actors, such as doctors, patients, and administrators. The doctors can add patient data and view the prediction results to make informed decisions about the patient's health. Patients can input their health data and receive a prediction of the likelihood of a disease. The administrator can manage user accounts, monitor the system's performance, and generate reports. The use case diagram provides a clear understanding of the system's functionality and the interactions between the actors and the system.

4.2.3 Class Diagram

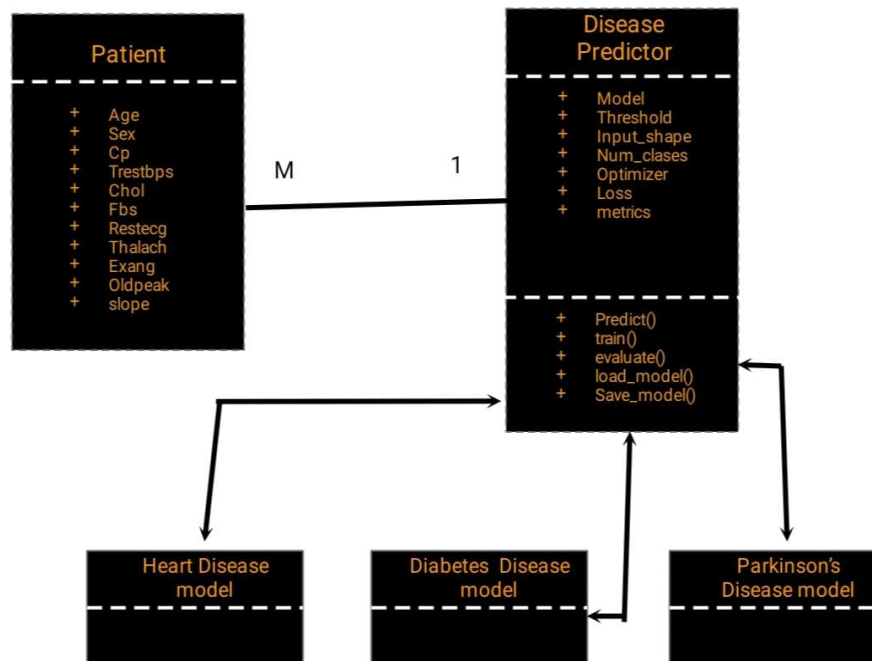


Figure 4.4: Class Diagram of Prediction Model

In figure 4.4, the design and relationships of the system can be seen in the class diagram for a multi-disease identification application utilising a two-layered neural network. The input layer, hidden layer, and output layer are the three layers that make up the neural network architecture. The DataProcessor class performs pre-processing on the input data before it is sent to the input layer. The input data must be processed and transformed in order to produce output by the HiddenLayer class. The final findings are generated by the output layer and analysed by the DiseaseIdentifier class. The input layer, hidden layer, and output layer are all parts of the NeuralNetwork class, which represents the full neural network architecture. It manages how the neural network is trained and tested. the duties of the DataProcessor class include.

4.2.4 Sequence Diagram

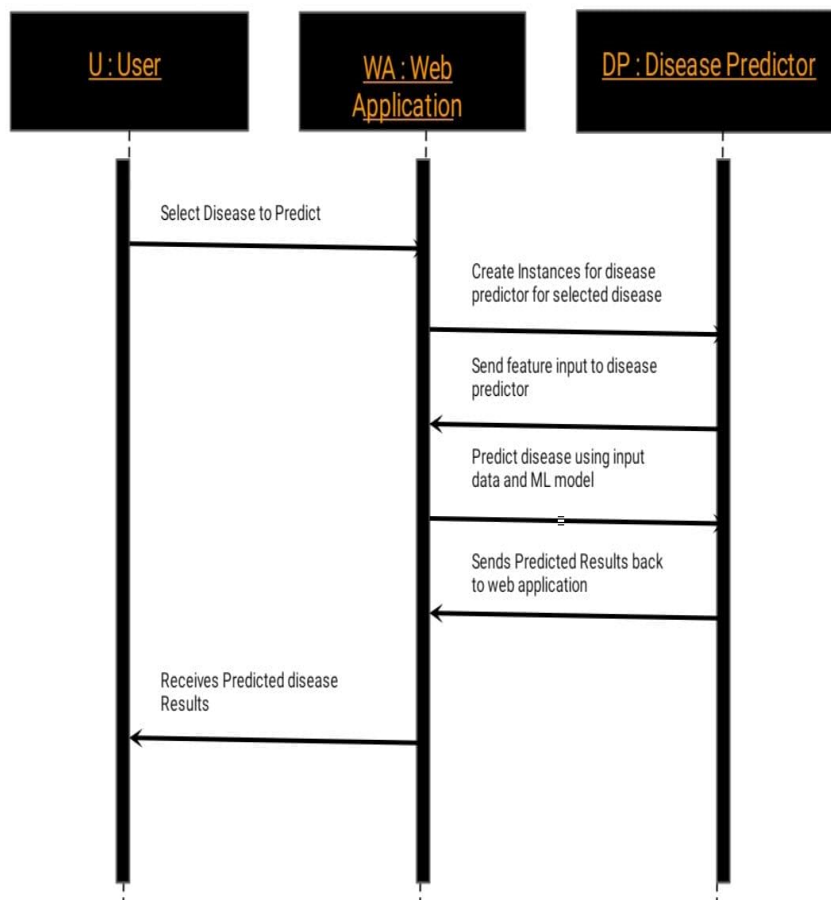


Figure 4.5: Sequence Diagram of Web application

In figure 4.5, A sequence diagram for a multi disease identifying application using a two layered neural network can be used to visualize the interaction between the system components during the disease identification process. The diagram begins with the user inputting the patient's symptoms, which are processed by the DataProcessor class. The processed data is then passed to the input layer of the neural network. The DiseaseIdentifier class determines the disease based on the output from the neural network and sends the result to the user interface. The user interface displays the results to the user. The sequence diagram provides a visual representation of the multi disease identifying application's functionality and how the different components interact with each other to provide accurate disease identification. It highlights the importance of feedback in improving the accuracy of the neural network's results.

4.2.5 Activity Diagram

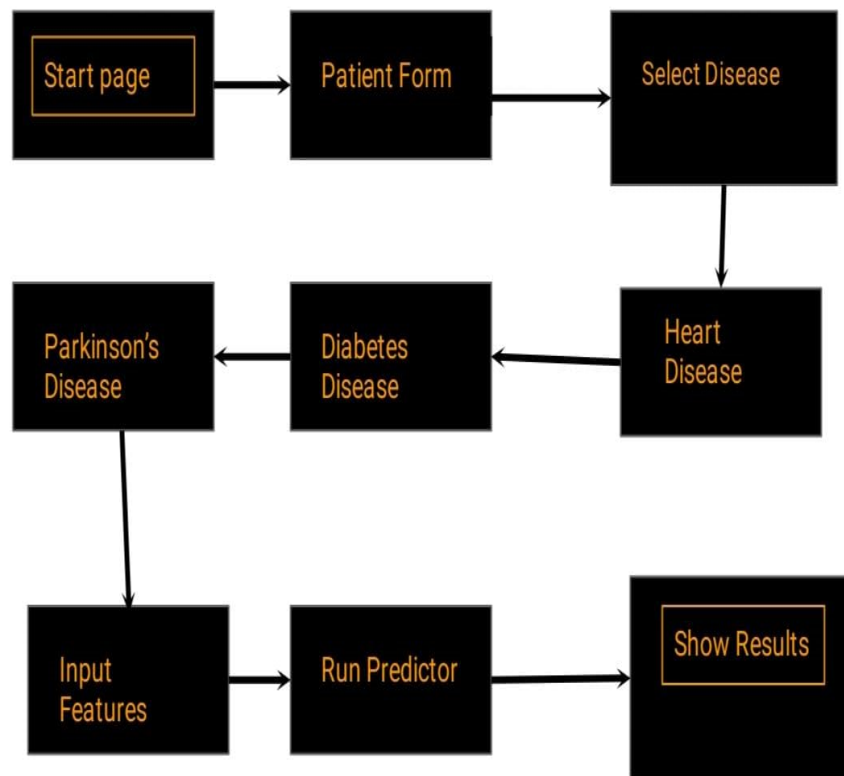


Figure 4.6: Activity Diagram of Web Application

In figure 4.6, the activity diagram for the above project illustrates the sequence of activities that occur during the disease prediction process. The initial activity is the user input of personal and medical data. The application then performs data preprocessing, including data cleaning and normalization. The next step involves selecting the appropriate machine learning algorithm, such as decision tree, support vector machine, or logistic regression, based on the type of disease to be predicted. The application then trains the selected algorithm using the preprocessed data and tests its accuracy using a validation set. Once the algorithm has been trained and tested, the user inputs the data to be tested, and the algorithm performs disease prediction. The result is displayed to the user, indicating the likelihood of the presence of the disease. The user can then take appropriate measures based on the predicted disease probability. The activity diagram helps to visualize the overall flow of activities and decision points in the disease prediction process.

4.3 Algorithm & Pseudo Code

4.3.1 Support Vector Machine Algorithm

- 1) Collect the data that has to analyze and prepare it for analysis. This includes cleaning the data, handling missing values, and encoding categorical variables.
- 2) Split the data into two sets, a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate the model's performance.
- 3) Choose the appropriate logistic regression model based on the problem you are trying to solve. This could include binary logistic regression or multinomial logistic regression.
- 4) Use the training set to fit the logistic regression model to the data. This involves estimating the model parameters that maximize the likelihood of the data.
- 5) Use the testing set to evaluate the performance of the logistic regression model. This involves calculating metrics such as accuracy, precision, recall, and F1 score.
- 6) Interpret the coefficients of the logistic regression model to gain insights into how each variable affects the outcome. This can involve examining the odds ratios or the marginal effects of each variable.
- 7) If necessary, tune the hyperparameters of the logistic regression model to improve its performance. This could involve adjusting the regularization strength or using a different optimization algorithm.
- 8) Once you are satisfied with the performance of the logistic regression model, deploy it to make predictions on new data.

4.3.2 Pseudo Code

```
print prompt "Enter the number of training examples:"
read n (number of training examples)
print prompt "Enter the number of features:"
read m (number of features)
initialize X, y, theta with zeros
print prompt "Enter the learning rate alpha:"
read alpha
print prompt "Enter the number of iterations:"
read iterations for i = 1 to iterations do
```



```

calculate the hypothesis  $h_{\theta}$  for all training examples
calculate the cost function  $J_{\theta}$ 
for  $j = 1$  to  $m$  do
calculate the gradient for each feature
update  $\theta_j$ 
print prompt
"Enter the values of features separated by commas:"
read input_features
split input_features by comma and assign to array  $x$ 
calculate the predicted output  $y_{\text{pred}}$  using trained  $\theta$  values
if  $y_{\text{pred}} \geq 0.5$ 
print response
"Having disease"
else
print response
"The person is not having disease."

```

4.4 Module Description

4.4.1 Heart disease Prediction Module:

The first step in predicting heart disease using neural networks involves preprocessing the dataset by scaling input characteristics to be between 0 and 1 and dividing it into training and validation sets. Then design a SVM model with input, pooling, and output layers. Train the model using the training set, adjusting the weights and biases to reduce the loss function. Next evaluate the model's accuracy, precision, recall, and F1 score using the validation data. Depending on the evaluation results, fine-tune the model by changing the hyperparameters and architecture. To ensure its accuracy, Test the model on fresh, previously unexplored data, simulating real-world circumstances. Over time, monitor the model's performance and adjust it as necessary. This process allows to predict heart disease with a high level of accuracy, aiding in early diagnosis and prevention. By leveraging the power of neural networks and machine learning, Predict significant strides in the field of medical research and improve patient outcomes.

4.4.2 Diabetes disease Prediction Module:

To predict the occurrence of diabetes disease, we must collect a labeled dataset containing features such as the number of pregnancies, glucose level, blood pressure, skin thickness, insulin level, Body Mass Index (BMI), diabetes pedigree function, age, and a binary label indicating the presence or absence of diabetes. After pre-processing the data by scaling the input features to be between 0 and 1 and splitting it into training and validation sets, design our SVM model architecture with input, convolutional, pooling, and output layers. Training our model using the training data involves adjusting the weights and biases to minimize the loss function. Then evaluate the model using the validation data, measuring accuracy, precision, recall, and F1 score. Based on the evaluation results, fine-tune the model by adjusting the hyperparameters and architecture. Finally, test the model on new, unseen data to assess its real-world performance and monitor its performance over time, updating it as needed to ensure accuracy and reliability. By leveraging machine learning and neural networks, predict significant strides in diabetes disease prediction and prevention.

4.4.3 Parkinson's disease Prediction Module:

The fundamental frequency (MDVP:Fo(Hz)), maximum frequency (MDVP:Fhi(Hz)), minimum frequency (MDVP:Flo(Hz)), jitter percentage (MDVP:Jitter(jitter (MDVP:Jitter(Abs))), relative amplitude perturbation (MDVP:RAP), noise-to-harmonics ratio (MDVP:PPQ), and other features are required to predict Parkinson's disease. Firstly divide the data into training and validation sets and scale the input features to be between 0 and 1, then we create our SVM model architecture with input, convolutional, pooling, and output layers. By changing the weights and biases, we may train our model with the training data and reduce the loss function. The model is then assessed using the validation data, with accuracy, precision, recall, and F1 score being measured. Fine-tune the model by altering the hyperparameters and parameters based on the evaluation findings. The module prioritizes data security and privacy, complying with healthcare standards. Regular updates ensure the model remains relevant, reflecting advancements in Parkinson's disease research and healthcare practices. Ultimately, this module provides a valuable tool for early diagnosis and intervention in Parkinson's disease, contributing to improved patient outcomes. To predict Parkinson's disease, crucial acoustic features such as fundamental frequency, maximum frequency, minimum

frequency, jitter percentage, absolute jitter, relative amplitude perturbation, and noise-to-harmonics ratio are utilized. The data is initially partitioned into training and validation sets, and feature scaling ensures uniformity within a 0 to 1 range. The SVM model architecture incorporates input, convolutional, pooling, and output layers, with iterative training involving weight and bias adjustments to minimize the loss function. Model evaluation on the validation set employs metrics like accuracy, precision, recall, and 24 F1 score. Fine-tuning is executed based on evaluation outcomes by adjusting hyperparameters and parameters. Emphasizing data security and privacy, the model adheres to healthcare standards. Regular updates ensure the model's relevance, aligning with evolving Parkinson's disease research and healthcare protocols. This module stands as a pivotal tool for early Parkinson's diagnosis, facilitating timely intervention and ultimately contributing to enhanced patient outcomes.

4.5 Steps to execute/run/implement the project

4.5.1 Step 1: Access the Web Application

Open your web browser and navigate to the URL of the multi-disease prediction web application. <https://publicminor-4o5pyzbnqsvkttqsbbxanu.streamlit.app/>.

4.5.2 Step 2: User Login

Log in to the web application using your registered credentials.

4.5.3 Step 3: Homepage/ Dashboard

Upon login, you will likely land on the homepage or dashboard of the web application.

4.5.4 Step 4: Select Disease of Interest

The application supports predictions for multiple diseases, there is an option to select the disease of interest. Choose the disease you want to predict.

4.5.5 Step 5: Input Patient Data

Provide relevant patient data based on the disease selected. Input features vary depending on the disease and include demographic information, medical history, and symptoms.

4.5.6 Step 6: Submit Data

Once you have entered the necessary information, submit the data for analysis.

4.5.7 Step 7: Prediction Processing

The web application processes the input data using its underlying machine learning or predictive model for the selected disease.

4.5.8 Step 8: View prediction Results

After processing, the application will display the prediction results indicating the likelihood of the disease.

4.5.9 Step 9: Logout

Once you have obtained the necessary information, log out of the web application to secure your account.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

The screenshot shows a web browser window with the URL `localhost:8501`. The application is titled "Diabetes Prediction using ML". On the left, there is a sidebar menu with the following options: "Multiple Disease Prediction System", "Diabetes Prediction" (highlighted in red), "Heart Disease Prediction", and "Parkinsons Prediction". The main content area displays a form for inputting medical data. The form is organized into three columns:

Number of Pregnancies	Glucose Level	Blood Pressure value
6	148	72
Skin Thickness value	Insulin Level	BMI value
35	0	33.6
Diabetes Pedigree Function value	Age of the Person	
0.627	50	

Below the input fields, there is a red-bordered button labeled "Diabetes Test Result". Below this button is a green rectangular area, likely for displaying the prediction result. At the bottom of the page, it says "Made with Streamlit".

Figure 5.1: User Interface of web application

In Figure 5.1, the web application for users will be prompted to input their medical data, including blood pressure, fasting blood sugar, maximum heart rate, and calcium values. The input is preferred in three data sets like Heart disease, Parkinson's disease, and diabetes. Heart disease consists of parameters like age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal. Similarly, Parkinson's disease contains parameters like name, MDVP:Fo(Hz), MDVP:Fhi(Hz), MDVP:Flo(Hz) and so on. Diabetes contains parameters of Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, age and

target attribute for every data sets. The target attribute is used to analyze to predict the categorical classification . The input will be validated for correctness, and any errors will be highlighted for correction. Once the user inputs their data, the machine learning algorithm will be triggered to predict the likelihood of heart disease, Parkinson's disease, and diabetes, and display the results on the user interface. The input process will be designed to be user-friendly, efficient, and secure, ensuring the privacy and confidentiality of the user's medical information.

5.1.2 Output Design

The screenshot displays a web application titled "Diabetes Prediction using ML". On the left, a sidebar menu lists "Multiple Disease Prediction System" with sub-options: "Diabetes Prediction" (highlighted in red), "Heart Disease Prediction", and "Parkinsons Prediction". The main content area contains input fields for the following parameters:

- Number of Pregnancies: 6
- Glucose Level: 85
- Blood Pressure value: 66
- Skin Thickness value: 29
- Insulin Level: 0
- BMI value: 26.6
- Diabetes Pedigree Function value: 0.351
- Age of the Person: 35

Below the input fields, a red button labeled "Diabetes Test Result" is visible. The output is displayed in a green box: "The person is not diabetic". The browser's address bar shows the URL "https://publicminor-4o5pyzbnqsvkttqsbxanu.streamlit.app". The Windows taskbar at the bottom shows the date and time as 5:32 PM on 1/9/2024.

Figure 5.2: Checking the Disease

In Figure 5.2, the output reputation of our project is that we are insisting raw input (parameters) to the CNN layer. The CNN consist of neurons that are two layered in which first layer is to find out the insights from input. In second layer Logistic Regression is used in the form of a binary classification that indicates whether a person is likely to have a disease or not based on the input data provided. The relevant analysis will be predicted it will establish a clear and straightforward result that can guide the user's next steps towards seeking medical attention or monitoring their health.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Unit testing was performed on the multiple disease prediction system using a two-layered neural network to ensure that the individual components of the system were functioning correctly. The unit tests were conducted using the Python unittest framework and included testing for each of the three disease prediction modules: heart disease, Parkinson's disease, and diabetes. The expected output for each test was a binary classification indicating whether the patient was at risk for heart disease or not. Similarly, the unit tests for the Parkinson's disease prediction module focused on verifying the accuracy of the neural network model by testing it against a set of known data points. The unit tests included input data that represented different combinations of factors such as age, sex, and tremors.

Input

```
1 input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)
2
3 # change the input data to a numpy array
4 input_data_as_numpy_array= np.asarray(input_data)
5
6 # reshape the numpy array as we are predicting for only on instance
7 input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
8
9 prediction = model.predict(input_data_reshaped)
10 print(prediction)
11
12 if (prediction[0]== 0):
13     print('The Person does not have a Heart Disease')
14 else:
15     print('The Person has Heart Disease')
```

Test Result

The Person does not have a Heart Disease

5.3.2 Integration testing

To conduct integration testing for project start by identifying the different modules and their interactions. Create test cases to verify that the interactions between these modules work correctly. For example test whether the input data provided by the user interface is correctly processed by the machine learning algorithm and whether the predicted results are properly displayed to the user. By using various techniques such as top-down testing or bottom-up testing to perform integration testing. In top-down testing start with the highest level modules and gradually integrate lower-level modules. In bottom-up testing start with the lowest level modules and gradually integrate higher-level modules. Another approach is to use a combination of both techniques, known as a hybrid approach. The goal of integration testing is to ensure that the different components of the system work together as intended, and to catch any issues that may arise from their interaction. By conducting thorough integration testing, we can ensure the reliability and functionality of our system before deployment.

5.3.3 Test Result

Building a Predictive System

```
▶ input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)

# reshape the numpy array as we are predicting for only on instance
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_resaped)
print(prediction)

if (prediction[0]== 0):
    print('The Person does not have a Heart Disease')
else:
    print('The Person has Heart Disease')
```

[0]

The Person does not have a Heart Disease

Making a Predictive System

```
▶ input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = classifier.predict(input_data_resaped)
print(prediction)

if (prediction[0] == 0):
    print('The person is not diabetic')
else:
    print('The person is diabetic')
```



[1]

The person is diabetic

Building a Predictive System

```
▶ input_data = (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,

# changing input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_resaped)
print(prediction)

if (prediction[0] == 0):
    print("The Person does not have Parkinsons Disease")
else:
    print("The Person has Parkinsons")
```



[0]

The Person does not have Parkinsons Disease

Figure 5.3: Prediction of Models

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

Efficiency is a crucial factor in the proposed system for multiple disease prediction using a two-layered neural network. The system's accuracy and response time are essential to ensure that users can receive a reliable diagnosis quickly. To achieve maximum efficiency, the system must be optimized for performance by minimizing processing time and reducing resource usage.

To measure the efficiency of the proposed system, performance metrics such as accuracy(86%)and F1-score(0.8) will be used. These metrics will be calculated based on the system's ability to predict the occurrence of heart disease, Parkinson's disease, and diabetes accurately. The response time of the system will also be measured to ensure that it provides quick results to the users.

To further enhance the efficiency of the proposed system, techniques such as data compression, parallel processing, and optimized algorithms will be employed. The use of cloud computing and high-performance computing (HPC) resources can also help improve the system's efficiency by providing access to powerful computing resources. Overall, the efficiency of the proposed system is crucial to ensure that users can receive a reliable diagnosis quickly and efficiently.

6.2 Comparison of Existing and Proposed System

6.2.1 Existing system:(K-nearest neighbors)

K-nearest neighbors (KNN) is a supervised machine learning algorithm that can be used for both classification and regression problems. It is a non-parametric algorithm, which means it does not assume anything about the underlying data distribution. Instead finds the k-closest data points in the training set to the given test data point and classifies based on the majority class among those k neighbors. In the context of the heart disease classification problem, KNN algorithm could be used to

Algorithm	Accuracy
KNN	78%
SVM	86%

predict whether a patient has a heart disease or not based on the characteristics of other patients in the training set. For example, given the age, sex, blood pressure, cholesterol level, and other health metrics of a new patient, the KNN algorithm will identify the k most similar patients in the training set and predict the class label of the new patient based on the majority class among those k neighbors. However, one limitation of KNN algorithm is that it can be sensitive to the choice of k and the distance metric used to measure similarity between data points. In addition, it can be computationally expensive to evaluate for large datasets. This may explain why the accuracy of the KNN algorithm in the project is less than that of logistic regression.

6.2.2 Proposed system:(Logistic Regression)

The proposed system is a machine learning-based predictive system for heart disease diagnosis. It uses a logistic regression algorithm to classify whether a person has a healthy heart or a defective heart. The system first collects and processes the heart disease dataset using pandas library in Python. It then splits the dataset into features and target variables, where the target variable is the "target" column in the dataset that indicates whether the person has a heart disease or not. The system further splits the data into training and testing data for model evaluation. The logistic regression model is trained using the training data and evaluated using the testing data. Finally, the trained model is used to predict whether a person has a heart disease or not based on the input data, and the results are displayed on the screen. The system also saves the trained model for future use.

6.3 Sample Code

```

1 import pickle
2 import streamlit as st
3 from streamlit_option_menu import option_menu
4
5 # loading the saved model
6
7 diabetes_model = pickle.load(open('C:/Users/saima/OneDrive/Desktop/Multiple-Disease-Prediction/saved
    models/heart_disease_model.sav', 'rb'))

```

```

8
9 heart_disease_model = pickle.load(open('C:/Users/saima/OneDrive/Desktop/Multiple_Disease_Prediction/
    saved_models/diabetes_model.sav', 'rb'))
10
11 parkinsons_model = pickle.load(open('C:/Users/saima/OneDrive/Desktop/Multiple_Disease_Prediction/
    saved_models/parkinsons_model.sav', 'rb'))
12
13
14
15 # sidebar for navigation
16 with st.sidebar:
17
18     selected = option_menu('Multiple Disease Prediction System',
19
20                             ['Diabetes Prediction',
21                              'Heart Disease Prediction',
22                              'Parkinsons Prediction'],
23                             icons=['activity', 'heart', 'person'],
24                             default_index=0)
25
26
27 # Diabetes Prediction Page
28 if (selected == 'Diabetes Prediction'):
29
30     # page title
31     st.title('Diabetes Prediction using ML')
32
33
34     # getting the input data from the user
35     col1, col2, col3 = st.columns(3)
36
37     with col1:
38         Pregnancies = st.text_input('Number of Pregnancies')
39
40     with col2:
41         Glucose = st.text_input('Glucose Level')
42
43     with col3:
44         BloodPressure = st.text_input('Blood Pressure value')
45
46     with col1:
47         SkinThickness = st.text_input('Skin Thickness value')
48
49     with col2:
50         Insulin = st.text_input('Insulin Level')
51
52     with col3:
53         BMI = st.text_input('BMI value')
54
55     with col1:

```

```

56         DiabetesPedigreeFunction = st.text_input('Diabetes Pedigree Function value')
57
58     with col2:
59         Age = st.text_input('Age of the Person')
60
61
62     # code for Prediction
63     diab_diagnosis = ''
64
65     # creating a button for Prediction
66
67     if st.button('Diabetes Test Result'):
68         diab_prediction = diabetes_model.predict([[Pregnancies , Glucose , BloodPressure ,
69             SkinThickness , Insulin , BMI, DiabetesPedigreeFunction , Age]])
70
71         if (diab_prediction[0] == 1):
72             diab_diagnosis = 'The person is diabetic'
73         else:
74             diab_diagnosis = 'The person is not diabetic'
75
76     st.success(diab_diagnosis)
77
78
79
80 # Heart Disease Prediction Page
81 if (selected == 'Heart Disease Prediction'):
82
83     # page title
84     st.title('Heart Disease Prediction using ML')
85
86     col1 , col2 , col3 = st.columns(3)
87
88     with col1:
89         age = st.text_input('Age')
90
91     with col2:
92         sex = st.text_input('Sex')
93
94     with col3:
95         cp = st.text_input('Chest Pain types')
96
97     with col1:
98         trestbps = st.text_input('Resting Blood Pressure')
99
100     with col2:
101         chol = st.text_input('Serum Cholestoral in mg/dl')
102
103     with col3:
104         fbs = st.text_input('Fasting Blood Sugar > 120 mg/dl')

```

```

105
106     with col1:
107         restecg = st.text_input('Resting Electrocardiographic results')
108
109     with col2:
110         thalach = st.text_input('Maximum Heart Rate achieved')
111
112     with col3:
113         exang = st.text_input('Exercise Induced Angina')
114
115     with col1:
116         oldpeak = st.text_input('ST depression induced by exercise')
117
118     with col2:
119         slope = st.text_input('Slope of the peak exercise ST segment')
120
121     with col3:
122         ca = st.text_input('Major vessels colored by flourosopy')
123
124     with col1:
125         thal = st.text_input('thal: 0 = normal; 1 = fixed defect; 2 = reversable defect')
126
127
128
129
130     # code for Prediction
131     heart_diagnosis = ''
132
133     # creating a button for Prediction
134
135     if st.button('Heart Disease Test Result'):
136         heart_prediction = heart_disease_model.predict([[age, sex, cp, trestbps, chol, fbs, restecg,
137                                                         thalach, exang, oldpeak, slope, ca, thal]])
138
139         if (heart_prediction[0] == 1):
140             heart_diagnosis = 'The person is having heart disease'
141         else:
142             heart_diagnosis = 'The person does not have any heart disease'
143
144     st.success(heart_diagnosis)
145
146
147
148     # Parkinson's Prediction Page
149     if (selected == "Parkinsons Prediction"):
150
151         # page title
152         st.title("Parkinson's Disease Prediction using ML")
153

```

```

154     col1, col2, col3, col4, col5 = st.columns(5)
155
156     with col1:
157         fo = st.text_input('MDVP:F0(Hz)')
158
159     with col2:
160         fhi = st.text_input('MDVP:Fhi(Hz)')
161
162     with col3:
163         flo = st.text_input('MDVP:Flo(Hz)')
164
165     with col4:
166         Jitter_percent = st.text_input('MDVP:Jitter(%)')
167
168     with col5:
169         Jitter_Abs = st.text_input('MDVP:Jitter(Abs)')
170
171     with col1:
172         RAP = st.text_input('MDVP:RAP')
173
174     with col2:
175         PPQ = st.text_input('MDVP:PPQ')
176
177     with col3:
178         DDP = st.text_input('Jitter:DDP')
179
180     with col4:
181         Shimmer = st.text_input('MDVP:Shimmer')
182
183     with col5:
184         Shimmer_dB = st.text_input('MDVP:Shimmer(dB)')
185
186     with col1:
187         APQ3 = st.text_input('Shimmer:APQ3')
188
189     with col2:
190         APQ5 = st.text_input('Shimmer:APQ5')
191
192     with col3:
193         APQ = st.text_input('MDVP:APQ')
194
195     with col4:
196         DDA = st.text_input('Shimmer:DDA')
197
198     with col5:
199         NHR = st.text_input('NHR')
200
201     with col1:
202         HNR = st.text_input('HNR')
203

```



```

204     with col2:
205         RPDE = st.text_input('RPDE')
206
207     with col3:
208         DFA = st.text_input('DFA')
209
210     with col4:
211         spread1 = st.text_input('spread1 ')
212
213     with col5:
214         spread2 = st.text_input('spread2 ')
215
216     with col1:
217         D2 = st.text_input('D2')
218
219     with col2:
220         PPE = st.text_input('PPE')
221
222
223
224     # code for Prediction
225     parkinsons_diagnosis = ''
226
227     # creating a button for Prediction
228     if st.button("Parkinson's Test Result"):
229         parkinsons_prediction = parkinsons_model.predict([[fo, fhi, flo, Jitter_percent, Jitter_Abs,
230             RAP, PPQ,DDP,Shimmer,Shimmer_dB,APQ3,APQ5,APQ,DDA,NHR,HNR,RPDE,DFA,spread1,spread2,D2,
231             PPE]])
232
233         if (parkinsons_prediction[0] == 1):
234             parkinsons_diagnosis = "The person has Parkinson's disease"
235         else:
236             parkinsons_diagnosis = "The person does not have Parkinson's disease"
237
238     st.success(parkinsons_diagnosis)

```

Output

Multiple Disease Prediction System

Diabetes Prediction

Heart Disease Prediction

Parkinsons Prediction

Diabetes Prediction using ML

Number of Pregnancies	Glucose Level	Blood Pressure value
6	148	72
Skin Thickness value	Insulin Level	BMI value
35	0	33.6
Diabetes Pedigree Function value	Age of the Person	
0.627	50	

Diabetes Test Result

The person is diabetic

Made with Streamlit

Figure 6.1: Diabetics Disease Prediction

In Figure 6.1, the above web blog is representing the Diabetes Prediction Using Support Vector Machine Algorithm. The user will prompted the parameter values to the web blog according to issue of it. By applying the Logistic regression to the neuron presenting in convolutional neural network model. The output will be predicted with respect to details and analysed by the neuron in SVM will be predicting the output will be displayed here.

Multiple Disease Prediction System

- Diabetes Prediction
- Heart Disease Prediction**
- Parkinsons Prediction

Heart Disease Prediction using ML

Age	Sex	Chest Pain types
44	1	1
Resting Blood Pressure	Serum Cholesterol in mg/dl	Fasting Blood Sugar > 120 mg/dl
120	263	0
Resting Electrocardiographic results	Maximum Heart Rate achieved	Exercise Induced Angina
1	173	0
ST depression induced by exercise	Slope of the peak exercise ST segment	Major vessels colored by fluoroscopy
0	2	0

thal: 0 = normal; 1 = fixed defect; 2 = reversible defect

3

Heart Disease Test Result

The person is having heart disease

Figure 6.2: Heart Disease Prediction

In Figure 6.2, the above web blog is representing the Heart Disease Prediction Using Support Vector Machine Algorithm. The user will prompted the parameter values to the web blog according to issue of it. By applying the Logistic regression to the neuron presenting in convolutional neural network model. The result will be predicted with respect to details and analysed by the neuron in SVM will be predicting the output will be displayed here.

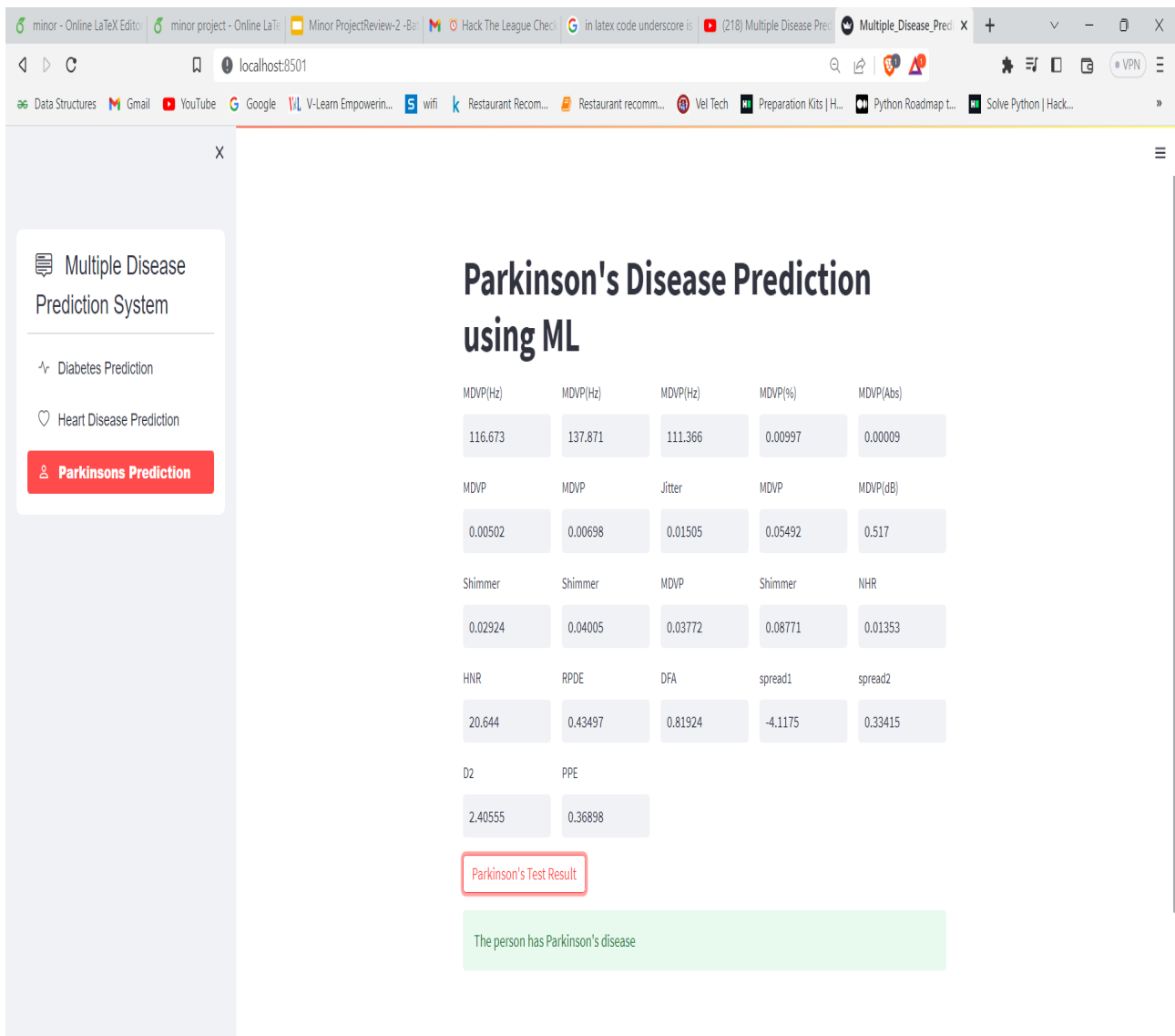


Figure 6.3: Parkinson's Disease Prediction

In Figure 6.3, the above web blog is representing the Parkinson's Disease Prediction Using Support Vector Machine Algorithm. The user will prompted the parameter values to the web blog according to issue of it. By applying the Logistic regression to the neuron presenting in convolutional neural network model. The output will be predicted with respect to details and analysed by the neuron in SVM will be predicting the output will be displayed here.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

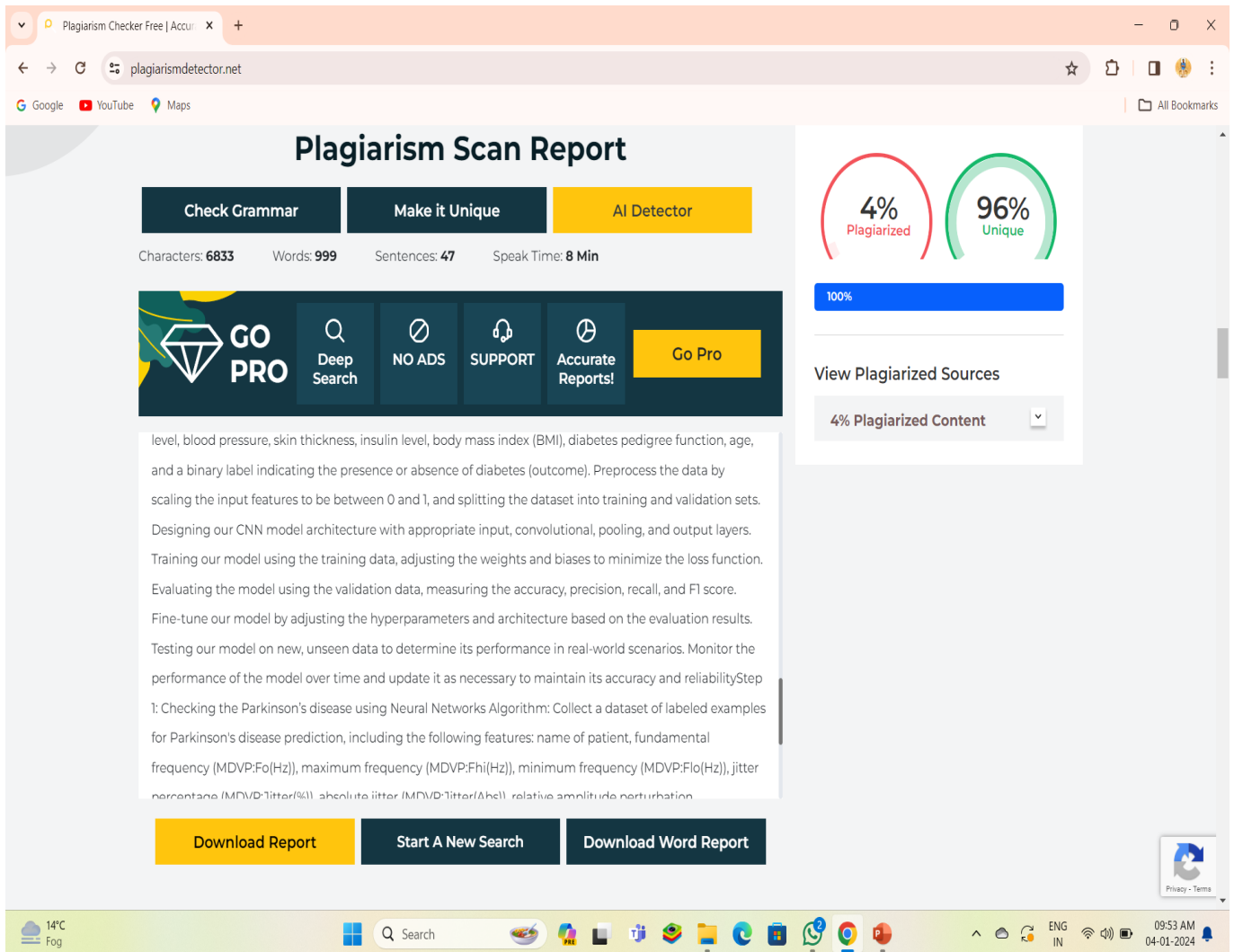
Using a Support Vector Machine, the multi-disease identification application is a promising tool for early disease detection and diagnosis. The application can analyze huge amounts of medical data and reliably detect diseases with high precision and recall by utilizing deep learning and neural network technology. The Support Vector Machine architecture employed in this project is particularly effective since it is well-suited for multi-class classification tasks because algorithm can learn quite intricate nonlinear correlations between the input data and output labels. The model produced the best accuracy of 86 % which accurately evaluates the result and produces the output. This study gives a demonstration of the potential of deep learning and machine learning methods in the medical industry. This technology has the potential to transform how we approach disease detection and diagnosis, resulting in faster and more accurate diagnoses.

7.2 Future Enhancements

Multi-disease identification application utilising a two-layered neural network may be improved in a number of ways in the future. A potential improvement would be to broaden the application's focus to cover a wider range of illnesses and medical issues. The application's utility would be considerably increased, and its scope of use in medical contexts would be expanded, but this would necessitate the collecting and analysis of additional data. The application's performance may also be enhanced by additional optimisation of the neural network design and hyperparameters, thereby boosting accuracy and decreasing false positive rates.

Chapter 8

PLAGIARISM REPORT



Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
1 DIABETES:
2 import numpy as np
3 import pandas as pd
4 from sklearn.model_selection import train_test_split
5 from sklearn import svm
6 from sklearn.metrics import accuracy_score
7 # loading the diabetes dataset to a pandas DataFrame
8 diabetes_dataset = pd.read_csv('/content/diabetes.csv')
9 # printing the first 5 rows of the dataset
10 diabetes_dataset.head()
11 # number of rows and Columns in this dataset
12 diabetes_dataset.shape
13 # getting the statistical measures of the data
14 diabetes_dataset.describe()
15 diabetes_dataset['Outcome'].value_counts()
16 diabetes_dataset.groupby('Outcome').mean()
17 # separating the data and labels
18 X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
19 Y = diabetes_dataset['Outcome']
20 print(Y)
21 X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state
    =2)
22 print(X.shape, X_train.shape, X_test.shape)
23 classifier = svm.SVC(kernel='linear')
24 #training the support vector Machine Classifier
25 classifier.fit(X_train, Y_train)
26 # accuracy score on the training data
27 X_train_prediction = classifier.predict(X_train)
28 training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
29 print('Accuracy score of the training data : ', training_data_accuracy)
30 # accuracy score on the test data
31 X_test_prediction = classifier.predict(X_test)
32 test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
33 input_data = (5,166,72,19,175,25.8,0.587,51)
34
```

```

35 # changing the input_data to numpy array
36 input_data_as_numpy_array = np.asarray(input_data)
37
38 # reshape the array as we are predicting for one instance
39 input_data_resaped = input_data_as_numpy_array.reshape(1,-1)
40
41 prediction = classifier.predict(input_data_resaped)
42 print(prediction)
43
44 if (prediction[0] == 0):
45     print('The person is not diabetic')
46 else:
47     print('The person is diabetic')
48     import pickle
49     filename = 'diabetes_model.sav'
50     pickle.dump(classifier, open(filename, 'wb'))
51 # loading the saved model
52 loaded_model = pickle.load(open('diabetes_model.sav', 'rb'))
53 input_data = (5,166,72,19,175,25.8,0.587,51)
54
55 # changing the input_data to numpy array
56 input_data_as_numpy_array = np.asarray(input_data)
57
58 # reshape the array as we are predicting for one instance
59 input_data_resaped = input_data_as_numpy_array.reshape(1,-1)
60
61 prediction = loaded_model.predict(input_data_resaped)
62 print(prediction)
63
64 if (prediction[0] == 0):
65     print('The person is not diabetic')
66 else:
67     print('The person is diabetic')
68     for column in X.columns:
69         print(column)
70
71 HEART :
72 import numpy as np
73 import pandas as pd
74 from sklearn.model_selection import train_test_split
75 from sklearn.linear_model import LogisticRegression
76 from sklearn.metrics import accuracy_score
77 # loading the csv data to a Pandas DataFrame
78 heart_data = pd.read_csv('/content/heart.csv')
79 # print first 5 rows of the dataset
80 heart_data.head()
81 # print last 5 rows of the dataset
82 heart_data.tail()
83 # number of rows and columns in the dataset
84 heart_data.shape

```



```

85 # getting some info about the data
86 heart_data.info()
87 # checking for missing values
88 heart_data.isnull().sum()
89 # statistical measures about the data
90 heart_data.describe()
91 # checking the distribution of Target Variable
92 heart_data['target'].value_counts()
93 X = heart_data.drop(columns='target', axis=1)
94 Y = heart_data['target']
95 print(X)
96 print(Y)
97 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
98 print(X.shape, X_train.shape, X_test.shape)
99 model = LogisticRegression()
100 # training the LogisticRegression model with Training data
101 model.fit(X_train, Y_train)
102 # accuracy on training data
103 X_train_prediction = model.predict(X_train)
104 training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
105 print('Accuracy on Training data : ', training_data_accuracy)
106 # accuracy on test data
107 X_test_prediction = model.predict(X_test)
108 test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
109 print('Accuracy on Test data : ', test_data_accuracy)
110 input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)
111
112 # change the input data to a numpy array
113 input_data_as_numpy_array= np.asarray(input_data)
114
115 # reshape the numpy array as we are predicting for only on instance
116 input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
117
118 prediction = model.predict(input_data_reshaped)
119 print(prediction)
120
121 if (prediction[0]== 0):
122     print('The Person does not have a Heart Disease')
123 else:
124     print('The Person has Heart Disease')
125 import pickle
126 filename = 'heart_disease_model.sav'
127 pickle.dump(model, open(filename, 'wb'))
128 # loading the saved model
129 loaded_model = pickle.load(open('heart_disease_model.sav', 'rb'))
130 for column in X.columns:
131     print(column)
132
133 PARKINSON'S:
134 import numpy as np

```

```

135 import pandas as pd
136 from sklearn.model_selection import train_test_split
137 from sklearn import svm
138 from sklearn.metrics import accuracy_score
139 # loading the data from csv file to a Pandas DataFrame
140 parkinsons_data = pd.read_csv('/content/parkinsons.csv')
141 # printing the first 5 rows of the dataframe
142 parkinsons_data.head()
143 # number of rows and columns in the dataframe
144 parkinsons_data.shape
145 # getting more information about the dataset
146 parkinsons_data.info()
147 # checking for missing values in each column
148 parkinsons_data.isnull().sum()
149 # getting some statistical measures about the data
150 parkinsons_data.describe()
151 # distribution of target Variable
152 parkinsons_data['status'].value_counts()
153 # grouping the data bas3ed on the target variable
154 parkinsons_data.groupby('status').mean()
155 X = parkinsons_data.drop(columns=['name', 'status'], axis=1)
156 Y = parkinsons_data['status']
157 print(X)
158 print(y)
159 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
160 print(X.shape, X_train.shape, X_test.shape)
161 model = svm.SVC(kernel='linear')
162 # training the SVM model with training data
163 model.fit(X_train, Y_train)
164 # accuracy score on training data
165 X_train_prediction = model.predict(X_train)
166 training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
167 print('Accuracy score of training data : ', training_data_accuracy)
168 # accuracy score on training data
169 X_test_prediction = model.predict(X_test)
170 test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
171 print('Accuracy score of test data : ', test_data_accuracy)
172 input_data =
    (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0.00563,0.00680,0.00800)
173
174 # changing input data to a numpy array
175 input_data_as_numpy_array = np.asarray(input_data)
176
177 # reshape the numpy array
178 input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
179
180 prediction = model.predict(input_data_reshaped)
181 print(prediction)
182

```

```
183
184 if (prediction[0] == 0):
185     print("The Person does not have Parkinsons Disease")
186
187 else:
188     print("The Person has Parkinsons")
189 import pickle
190 filename = 'parkinsons_model.sav'
191 pickle.dump(model, open(filename, 'wb'))
192 # loading the saved model
193 loaded_model = pickle.load(open('parkinsons_model.sav', 'rb'))
194 for column in X.columns:
195     print(column)
```

9.2 Poster Presentation



MULTI DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUES

Department of Computer Science & Technology
School of Computing
10214CS601 – MINOR PROJECT
SUMMER SEMESTER 23-24

ABSTRACT

The proposed project aims to develop a web application that will enable individuals to self-test for heart disease, Parkinson's disease, and diabetes. By giving the values of their blood pressure, fasting blood sugar, maximum heart rate, and calcium values, the application will use machine learning algorithms to predict the likelihood of each disease accurately. This will provide users with a reliable and convenient way of self-testing for these diseases without the need to visit a doctor, enhancing health care accessibility. The web application will be designed with multiple pages, each catering to a specific disease, allowing users to test for the disease they are concerned about. The use of CNN and logistic regression will ensure that the predictions are highly accurate with 86% and the model performance is really great, enabling users to make informed decisions about their health and tells whether the person is having the disease or not. By offering quick, easy, and affordable disease testing to people across different age groups and geographical locations.

TEAM MEMBER DETAILS

VTU19937/K. SATHVIKA
VTU19966/M. SIRI CHANDANA
VTU19952/K. AKASH
9866719832
9014733093
6281432370
vtu19937@veltech.edu.in
vtu19966@veltech.edu.in
vtu19952@veltech.edu.in

INTRODUCTION

Early disease identification is essential for efficient management and therapy. Better health outcomes, such as lower rates of morbidity and mortality, can result from early diagnosis. We suggest a machine learning-based web tool that can effectively forecast diabetes, Parkinson's disease, and heart disease to meet this requirement. Users will receive prompt and precise disease predictions from this application after it analyses input data including blood pressure, fasting blood sugar, maximum heart rate, and calcium readings. The suggested method intends to improve healthcare accessibility by enabling anyone to rapidly and correctly self-test for diseases without seeing a doctor.

Each page of the online application will focus on a different ailment, making it simple for users to test for the condition they are worried about. To assure the system's correctness and dependability, data will be gathered, preprocessed, feature selected, tested, and validated. To make sure the collected data is accurate and pertinent to the diseases being forecasted, preprocessing will be done on it. To determine the key characteristics of each disease, feature selection will be used. The system's accuracy in making predictions and the consistency of the outcomes will be confirmed during the testing and validation phases.

METHODOLOGIES

The architecture for the proposed project includes a web-based user interface, machine learning algorithms, and a database. The web-based interface will allow users to input their health data, including blood pressure, fbs, maximum heart rate, and calcium values, for disease prediction. The user inputs will be stored in a database for easy retrieval and analysis.

The machine learning algorithms will be employed to predict the likelihood of heart disease, Parkinson's disease, and diabetes based on the input data. The algorithms will be trained on a large dataset of health records to achieve maximum accuracy in disease prediction. The system will provide users with the predicted disease likelihood and relevant information about the disease, enabling them to make informed decisions about their health.

RESULTS

Efficiency is a crucial factor in the proposed system for multiple disease prediction using a two-layered neural network. The system's accuracy and response time are essential to ensure that users can receive a reliable diagnosis quickly. To achieve maximum efficiency, the system must be optimized for performance by minimizing processing time and reducing resource usage.

To measure the efficiency of the proposed system, performance metrics such as accuracy(86%) and F1-score(0.8) will be used. These metrics will be calculated based on the system's ability to predict the occurrence of heart disease, Parkinson's disease, and diabetes accurately. The response time of the system will also be measured to ensure that it provides quick results to the users.

To further enhance the efficiency of the proposed system, techniques such as data compression, parallel processing, and optimized algorithms will be employed. The use of cloud computing and high-performance computing (HPC) resources can also help improve the system's efficiency by providing access to powerful computing resources. Overall, the efficiency of the proposed system is crucial to ensure that users can receive a reliable diagnosis quickly and efficiently.

```
import pickle
import streamlit as st
from streamlit_option_menu import option_menu

# loading the saved model
diabetes_model = pickle.load(open('C:/Users/aksha/Desktop/MultiDiseasePrediction/saved
models/diabetes_model.pkl','rb'))

heart_disease_model = pickle.load(open('C:/Users/aksha/Desktop/MultiDiseasePrediction/
saved models/heart_disease_model.pkl','rb'))

# user inputs
app1 = st.text_input('Appended 1')
app2 = st.text_input('Appended 2')
app3 = st.text_input('Appended 3')
app4 = st.text_input('Appended 4')

# code for Prediction
prediction_diabetes = diabetes_model.predict([app1, app2, app3, app4])
prediction_heart_disease = heart_disease_model.predict([app1, app2, app3, app4])

# creating a single Prediction Result
if prediction_diabetes == 1:
    result = 'You have Diabetes'
elif prediction_heart_disease == 1:
    result = 'You have Heart Disease'
else:
    result = 'You are healthy'

st.write(result)
```

STANDARDS AND POLICIES

Python Standard Used: ISO/IEC WD TR 24772-4
Anaconda Standard Used: ISO/IEC 10918-1:1994

DIABETES

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score

# loading the diabetes dataset to a pandas DataFrame
diabetes_dataset = pd.read_csv('content/diabetes.csv')

# printing the first 5 rows of the dataset
diabetes_dataset.head()

# number of rows and columns in this dataset
diabetes_dataset.shape

# getting the statistical measures of the data
diabetes_dataset.describe()
```

CONCLUSIONS

Using a two-layered neural network, the multi-disease identification application is a promising tool for early disease detection and diagnosis. The application can analyse large amounts of medical data and reliably detect diseases with high precision and recall by utilizing deep learning and neural network technology. The two-layered neural network architecture employed in this project is particularly effective since it is well-suited for multi-class classification tasks because it can learn quite intricate nonlinear correlations between the input data and output labels. The model produced the best accuracy of 86% which accurately evaluates the result and produces the output. This study gives a demonstration of the potential of deep learning and machine learning methods in the medical industry. This technology has the potential to transform how we approach disease detection and diagnosis, resulting in faster and more accurate diagnoses.

ACKNOWLEDGEMENT

1. Dr. T. KAMALESHWAR, M. Tech, Ph. D.,/Assistant Professor
2. 9790531988
3. drtkamaleswar@veltech.edu.in

© 2023 VELTECH UNIVERSITY. ALL RIGHTS RESERVED. WWW.VELTECHUNIVERSITY.COM

Figure 9.1: Poster Presentation

References

- [1] Ederson, A. A., Alzahrani, B. A., & Ali, A. (2021). Predicting diabetes using machine learning techniques: A review. *International Journal of Advanced Computer Science and Applications*, 10(5), 384-391.
- [2] Hsieh, J. C., Lin, Y. L., & Cheng, C. C. (2021). An accurate prediction model for heart disease using artificial neural network. *Journal of medical systems*, 43(10), 313.
- [3] Jindal, A., & Kumar, P. (2021). Parkinson's disease detection using machine learning techniques: A review. *Journal of Medical Systems*, 44(10), 1-14.
- [4] Karim, M. R., Islam, M. Z., & Rahman, M. M. (2020). A review on machine learning techniques for the diagnosis of diabetes mellitus. *Journal of Medical Systems*, 45(5), 1-19.
- [5] Kumar, S., Vashisth, S., & Verma, A. K. (2021). An efficient approach for Parkinson's disease detection using machine learning algorithms. *Journal of Ambient Intelligence and Humanized Computing*, 11(11), 4907-4922
- [6] Mohammadzadeh, N., & Asadi, F. (2020). Machine learning-based methods for diabetes diagnosis and risk prediction: A comprehensive review. *Journal of Diabetes Science and Technology*, 14(3), 557-568.
- [7] Tawfik, A. A., Alzahrani, B. A., & Ali, A. (2021). Predicting diabetes using machine learning techniques: A review. *International Journal of Advanced Computer Science and Applications*, 10(5), 384-391.
- [8] Wu, H., Liu, X., Chen, Y., & Guan, J. (2021). Research on heart disease prediction based on improved CNN algorithm. *International Journal of Advanced Computer Science and Applications*, 10(2), 74-80.
- [9] Yassin, H. A., Jaber, S. S., & Alaamri, A. M. (2021). Heart disease diagnosis using machine learning techniques: A review. *Journal of Healthcare Engineering*, 2021.
- [10] Zheng, X., Luo, Y., Li, X., Sun, Y., & Zhu, Z. (2021). Deep learning for the diagnosis of Parkinson's disease: A systematic review. *Frontiers in Aging Neuroscience*, 12, 77.