**MULTIPLE DISEASE PREDICTION SYSTEM**



***A project report submitted to***

***Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal***

***in partial fulfillment for the award of***

***the degree of***

***Bachelor of Technology***

***in***

***Computer Science & Engineering***

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY

**INDORE- 453331**

**Jan – June 2023**

**MULTIPLE DISEASE PREDICTION SYSTEM**

***A project report submitted to***

***Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal***

***in partial fulfillment for the award of***

***the degree of***

***Bachelor of Technology***

***in***

***Computer Science & Engineering***

**PROJECT GUIDE SUBMITTED BY**

Prof. Pooja Hardiya Ishita Agrawal (0829CS191060)

Ankita Vishwas (0829CS191020)

Tanuja Upadhyay (0829CS19171)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY

**INDORE- 453331**

**Jan - June 2023**

**ACKNOWLEDGEMENT**

With great pleasure and a sense of obligation, we express our heartfelt gratitude to **Prof. Pooja Hardiya** our project guide and project coordinator. We are highly indebted to her for her valuable guidance and ever-ready support in completing this project in the stipulated time. Also, we express our gratitude to her for her persisting encouragement, perpetual motivation, everlasting patience, and excellent expertise in discussions, during the progress of the project work, which has benefited us to an extent, which is beyond expressing. Her contributions are beyond the preview of acknowledgment.

We are again grateful to **Prof. Atul Agrawal**, Head of the Computer Science Department, SDBCT, Indore, for his encouragement throughout my project work.

We would like to warmly thank **Dr. Premanand Singh Chouhan** Director of SDBC, Indore for providing all facilities and academic environment for my project work.

We sincerely wish to express our gratefulness to all the members of the staff. We sincerely thank all our friends and well-wishers for directly or indirectly helping during work.

**Ishita Agrawal 0829CS191060**

**Ankita Vishwas 0829CS191020**

**Tanuja Upadhyay 0829CS191060**

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY

**INDORE, 453331**

****

**CERTIFICATE**

This is to certify that **ISHITA AGRAWAL(0829CS191060), ANKITA VISHWAS (0829CS191020), and TANUJA UPADHYAY (0829CS191171)** have completed their project work, titled **“MULTIPLE DISEASE PREDICTION SYSTEM”**  as per the syllabus and have submitted a satisfactory report on this project as a part of fulfillment towards the degree of **“BACHELOR OF TECHNOLOGY” (Computer Science & Engineering)**  from **RAJIV GANDHI PROUDYOGIKI VISHWAVIDHYALAYA, BHOPAL.**

**HEAD OF THE DEPARTMENT PROJECT GUIDE**

**DIRECTOR**

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY

**INDORE, 453331**



**CERTIFICATE**

This is to certify that **ISHITA AGRAWAL (8029CS191060), ANKITA VISHWAS(0829CS191020), and TANUJA UPADHYAY(0829CS191171)** have completed their project work, titled **“MULTIPLE DISEASE PREDICTION SYSTEM”** as per the syllabus and have submitted a satisfactory report on this project as a part of fulfillment towards the degree of **“BACHELOR OF TECHNOLOGY” (Computer Science & Engineering)**  from **RAJIV GANDHI PROUDYOGIKI VISHWAVIDHYALAYA, BHOPAL.**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| CHAPTER | TITLE | PAGE NO. |
|  | **ABSTRACT** | **i** |
|  | **LIST OF TABLES** | **ii** |
|  | **LIST OF FIGURES** | **iii** |
|  | **LIST OF ABBREVIATIONS AND SYMBOLS** | **iv** |
| 1. | Software Requirement Specification  1.1 Introduction   * + 1. Purpose     2. Scope     3. Problem in the existing system     4. Statement of Problem | 1  1  1  1  1  2 |
| 2. | System Requirement Analysis  2.1 Introduction   1. Purpose 2. Intended Audience and Reading Suggestions 3. Product scope   2.2 Overall Description   1. Product Perspective 2. Product Functions 3. User Classes and Characteristics 4. Operating Environment   2.3 External Interface Requirements  2.3.1 User Interface  2.3.2 Hardware Interface Requirements  2.3.3 Software Interface Requirements  2.3.4 Communication Requirements   * 1. Functional Requirement      1. Dataset Preparation and Pre-processing      2. Data Visualization      3. Data Labelling      4. Data Selection      5. Data Pre-processing      6. Data Splitting      7. Modeling      8. Model Deployment   2. Nonfunctional Requirements      1. Usability      2. Reliability      3. Performance      4. Supportability   3. Project Plan      1. Team Members      2. Division of Work      3. Time Schedule | 3-12  3  3  3  4  4  4  4  5  5  5  5  5  6  6  6  6  7  8  8  9  9  10  10  11  11  11  11  11  11  11  12  12 |
| 3. | Analysis   * 1. Methodology Used   2. Use Case diagram   3. Data Flow Diagram   4. Entity Relationship Model   5. Process Specification   6. Control Flow Diagram | 12  12  13  14  15  16  17 |
| 4. | Design   * 1. Architectural Design      1. System Architecture Diagram      2. Description of Architectural Design   2. Database Design   3. Component Design      1. Flow Chart   4. Interface Design      1. Screenshots | 18  18  18  19  22  24  24  25  25 |
| 5. | Conclusion | 29 |
| 6. | Future Scope | 30 |
| 7. | References | 31 |

**ABSTRACT**

Machine learning and Artificial Intelligence are playing a huge role in today’s world. From self-driving cars to medical fields, we can find them everywhere. The medical industry generates a huge amount of patient data which can be processed in a lot of ways. So, with the help of machine learning, we have created a Prediction System that can detect more than one disease at a time. Many of the existing systems can predict only one disease at a time and that too with lower accuracy. Lower accuracy can seriously put a patient’s health in danger. We have considered three diseases for now which are Heart, Parkinson’s, and Diabetes, and in the future, many more diseases can be added. The user has to enter various parameters of the disease and the system would display the output whether he/she has the disease or not. This project can help a lot of people as one can monitor the person’s condition and take the necessary precautions thus increasing life expectancy.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig. no.** | **Name** | **Page no.** |
| 2.1 | Data Visualization | 16 |
| 2.5 | Data Splitting | 17 |
| 3.1 | Agile Methodology | 20 |
| 3.2 | Scrum | 20 |
| 3.3 | Use Case Diagram | 21 |
| 3.4 | Data Flow Diagram | 21 |
| 3.5 | ER Model | 22 |
| 3.6 | Process Specification | 23 |
| 3.7 | CF Diagram | 24 |
| 4.1 | System Architecture Diagram | 25 |
| 4.2 | S – Shaped Curve | 27 |
| 4.3 | SVM | 27 |
| 4.4 | (a) Parkinson’s Disease Dataset  (b) Diabetes Disease Dataset  (c) Heart Disease Dataset | 28  29  29 |
| 4.5 | Methodology | 30 |
| 4.6 | Work Flow | 30 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name** | **Page No.** |
| 2.1 | Software Requirements | 15 |
| 2.2 | Hardware Requirements | 15 |
| 2.3 | Time Schedule | 19 |

**LIST OF ABBREVIATIONS AND SYMBOLS**

|  |  |
| --- | --- |
| ECG | Electrocardiography |
| PPE | Pitch Period Entropy |
| DFA | Detrended Fluctuation Analysis |
| RPDE | Recurrence Period Density Entropy |
| SVM | Support Vector Machine |
| CSV | Comma Separated Values |
| API | Application Programming Interface |

1. Software Requirement Specification

* 1. Introduction
     1. Purpose

The main motivation for doing this research is to present a multiple disease prediction model for the prediction of the occurrence of heart, diabetes, and Parkinson’s disease. Further, this research work is aimed at identifying the best classification algorithm for identifying the possibility of disease in a patient. This work is justified by performing a comparative study and analysis using the classification algorithms namely Logistic Regression and SV, which are used at different levels of evaluation. Although these are commonly used machine learning algorithms, heart disease prediction is a vital task involving the highest possible accuracy. Hence, both algorithms are evaluated at numerous levels and types of evaluation strategies. This will provide researchers and medical practitioners to establish a better.

* + 1. Scope

As we know today life has become so fast and anyone can get a heart attack anytime and anywhere. People hardly get time to go to the hospital far away from their places and get a body(heart) checkup so we have attempted so that people can have fast and easy access to their heart-related health measures whenever and wherever required. We have tried our level best to make software that provides everyone with this facility and ease.

* + 1. Problem in the existing system

Many of the existing machine learning models for health care analysis are concentrating on one disease per analysis. For example first is for liver analysis, one for cancer analysis, and one for lung diseases like that. If a user wants to predict more than one disease, he/she has to go through different sites. There is no common system where one analysis can perform more than one disease prediction. Some of the models have lower accuracy which can seriously affect patients’ health. When an organization wants to analyze their patient’s health reports, they have to deploy many models which in turn increases the cost as well as time. Some existing systems consider very few parameters that can yield false results.

* + 1. Statement of Problem

In multiple disease prediction, it is possible to predict more than one disease at a time. So the user doesn’t need to traverse different sites to predict the diseases. We are taking three diseases that are Parkinson’s, Diabetes, and Heart. As all three diseases are correlated with each other. To implement multiple disease analyses we are going to use machine learning algorithms and Streamlit. When the user is accessing this API, the user has to send the parameters of the disease along with the disease name. Streamlit will invoke the correspond-ding model and returns the status of the patient.

2. System Requirement Analysis

2.1 Introduction

2.1.1 Purpose

The purpose of a multiple-disease prediction system is to assist healthcare providers in the early diagnosis and treatment of various diseases by predicting the likelihood of multiple diseases based on patient data. The system uses machine learning algorithms to analyze a patient's medical history, symptoms, and test results and provides insights into the possible diseases they may be susceptible to. This helps in reducing the number of unnecessary tests, providing personalized treatment plans, and improving patient outcomes. The system can also be used to prioritize patients who require urgent care and assist healthcare providers in making more informed decisions.

2.1.2 Intended Audience and Reading Suggestions

The intended audience for a multiple disease prediction system using machine learning could include healthcare providers, medical researchers, data scientists, and individuals who are interested in healthcare technology.

For healthcare providers and medical researchers, it is recommended to read research papers and articles related to machine learning-based disease prediction systems, such as those published in medical journals or presented at conferences. This will provide a deeper understanding of the underlying concepts and techniques used in such systems.

For data scientists, it is recommended to read literature on machine learning algorithms, especially those used in healthcare, such as deep learning, decision trees, and random forests. It is also suggested to study medical terminologies, electronic health records, and data management principles.

For individuals interested in healthcare technology, it is recommended to read articles and blog posts on machine learning-based healthcare systems, such as those published in healthcare technology magazines or websites. This will provide a broad understanding of the potential applications of machine learning in healthcare and the benefits of disease prediction systems.

Overall, the reading suggestions would vary based on the individual's level of familiarity with machine learning, healthcare, and related topics.

2.1.3 Product Scope

The product scope for a multiple disease prediction system includes the development of a machine learning model that can accurately predict the likelihood of multiple diseases based on patient data. The system should be able to handle various patient attributes, such as medical history, symptoms, and test results, and provide a comprehensive report of the possible diseases a patient may be susceptible to. The system should be user-friendly and accessible to healthcare providers and patients alike, with the ability to integrate with existing healthcare systems. Additionally, the system should be scalable and adaptable to accommodate new diseases and medical data as they become available.

2.2 Overall Description

2.1.1 Product Perspective

The perspective of this project called “Multiple Disease Prediction System” is to predict the accurate disease of the patient using all their general information and also the symptoms. Using this information there we will compare with our datasets of the patients and predicts the disease which he/she is been through.

2.1.2 Product Functions

The main function of this product is that it can predict multiple diseases for a patient in one platform only. The user does not have to download new software to predict his/her multiple diseases.

2.1.3 User Classes and Characteristics

Here are some potential user classes and characteristics for a multiple disease prediction system:

1. Healthcare providers: This includes doctors, nurses, and other healthcare professionals who will be using the system to assist in diagnosis and treatment. They should have a strong medical background and understanding of disease diagnosis and treatment.

2. Patients: Patients are the individuals who will provide the data used by the system to make predictions. They may have varying levels of health literacy and may require different levels of guidance in providing accurate and complete information.

3. Data analysts: This group of users may work in the background, analyzing the data generated by the system to identify trends and improve the accuracy of predictions. They should have strong analytical skills and experience in data science.

4. System administrators: These are the individuals responsible for managing the system, ensuring its smooth operation, and maintaining the privacy and security of patient data. They should have experience in managing software systems and be knowledgeable about cybersecurity best practices.

5. Researchers: Researchers may use the system to analyze trends in disease diagnosis and treatment, identify new areas for research, and improve overall healthcare outcomes. They should have a strong understanding of healthcare research methods and data analysis techniques.

The characteristics that may be common across these user classes include a high level of trust in the system's accuracy and privacy protections, a need for clear and understandable outputs and user interfaces, and an ability to integrate the system with existing healthcare workflows and practices.

2.2.4 Operating Environment

The operating environment for this system will require a powerful hardware setup, various software tools, a large dataset, cloud computing, collaboration tools, and documentation tools

2.3 External Interface Requirements

2.3.1 User Interface

A user interface for a multiple disease prediction system can be designed to be intuitive and user-friendly. The interface can provide a form for the user to input patient information such as medical history, symptoms, and test results. The system can then use this information to generate a report that predicts the likelihood of multiple diseases. The report can be displayed in an organized and easy-to-read format and can include recommendations for further testing and treatment. The interface can also provide options for the user to customize the report, such as selecting specific diseases to be included or excluded from the prediction. Additionally, the interface can have a search function to retrieve past reports for comparison and tracking.

2.3.2 Software Requirements

|  |  |
| --- | --- |
| Name of Component | Specification |
| Operating System | Windows 9, Windows XP, Windows 7, Linux |
| Language | Python 3.7 |
| Database | MS Excel |
| Browser | Any of Mozilla, Opera, Chrome, etc. |
| Web Server | Streamlit |
| Software Development Kit | Streamlit |
| Scripting Language Enable | Python Script |

Table 2.1 – Software Requirements

2.3.3 Hardware Requirements

|  |  |
| --- | --- |
| Name of Component | Specification |
| Processor | Pentium III 630 MHz |
| RAM | 128 MB |
| Hard Disk | 20 GB |
| Monitor | 15’’ color monitor |
| Keyboard | 122 keys |

Table 2.2 – Hardware Requirements

2.3.4 Communication requirements

A multiple disease prediction system requires effective communication between healthcare providers and the system to ensure accurate and timely diagnosis of diseases. The system should be able to communicate the predicted diseases and their likelihood to the healthcare providers clearly and understandably. Additionally, the system should be able to receive feedback from the health providers and incorporate it into the prediction model. The system should also have the ability to communicate with various healthcare systems and exchange patient data securely. It The communication between the system and healthcare providers must be efficient and effective, to provide timely and accurate diagnosis and treatment for patients.

2.4 Functional Requirement

2.4.1 Dataset Preparation and Pre-processing

Data collection is defined as the procedure of collecting, measuring, and analyzing accurate insights for research

using standard validated techniques. A researcher can evaluate their hypothesis based on collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The approach to data collection is different for different fields of study, depending on the required information. The most critical objective of data collection is ensuring that information-rich and reliable data is collected for statistical analysis so that data-driven decisions can be made for research.

2.4.2 Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Example –

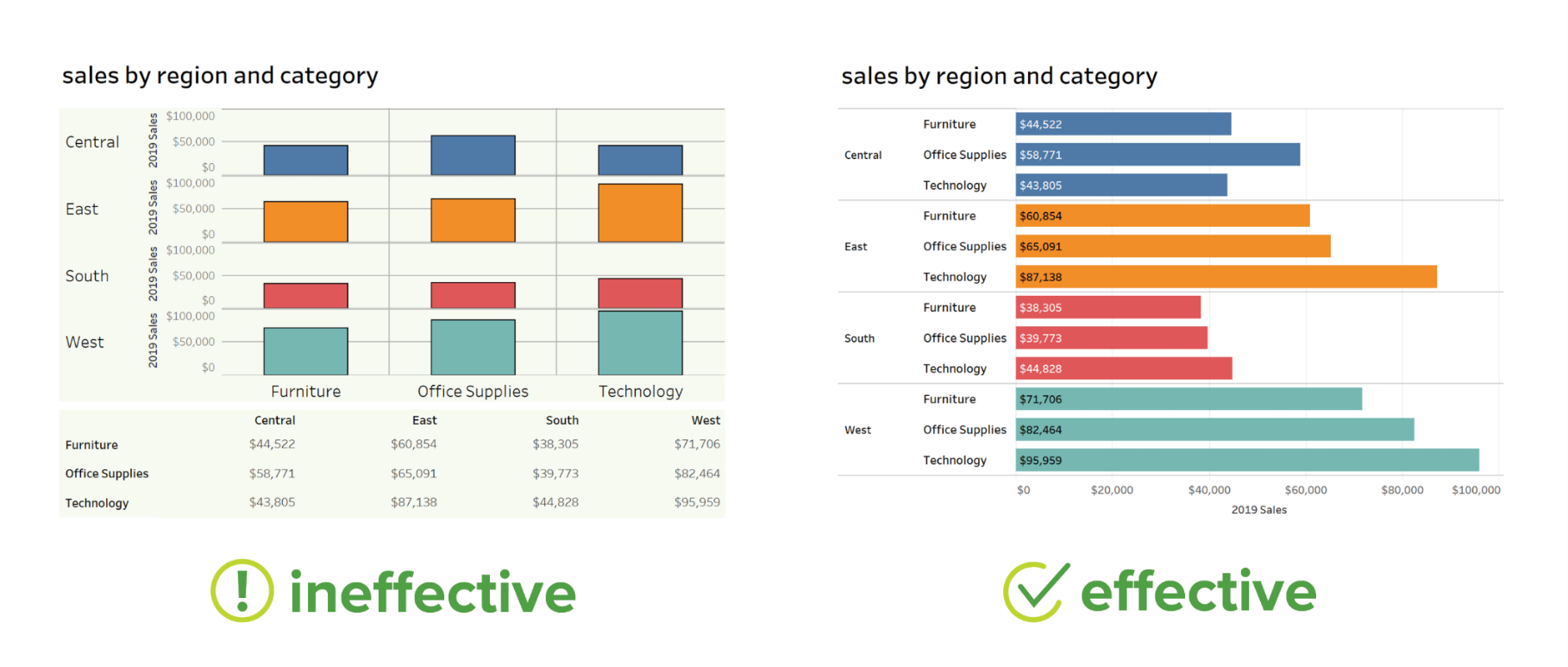


Fig. 2.1 – Data Visualization

2.4.3 Data Labelling

Supervised machine learning, which we’ll talk about below, entails training a predictive model on historical data with predefined target answers. An algorithm must be shown which target answers or attributes to look for. Mapping these target attributes in a dataset is called labeling. Data labeling takes much time and effort as datasets sufficient for machine learning may require thousands of records to be labeled. For instance, if your image recognition algorithm must classify types of bicycles, these types should be clearly defined and labeled in a dataset.

2.4.4 Data Selection

Data selection is defined as the process of determining the appropriate data type and source, as well

as suitable instruments to collect data. Data selection precedes the actual practice of data collection. This definition distinguishes data selection from selective data reporting (selectively excluding data that is not supportive of a research hypothesis) and interactive/active data selection (using collected data for monitoring activities/events or conducting secondary data analyses). The process of selecting suitable data for a research project can impact data integrity. After having collected all information, a data analyst chooses a subgroup of data to solve the defined problem. For instance, if you save your customers’ geographical location, you don’t need to add their cell phones and bank card numbers to a

dataset. But purchase history would be necessary. The selected data includes attributes that need to be considered when building a predictive model.

2.3.5 Data Pre-processing

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. The purpose of pre-processing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling.

2.3.6 Data Splitting

A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

Training set - A data scientist uses a training set to train a model and define its optimal parameters it must learn from data.

Test set - A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model’s ability to identify patterns in new unseen data after having been trained over training data. It is crucial to use different subsets for training and testing to avoid model overfitting, which is the incapacity for generalization we mentioned above.

Validation set - The purpose of a validation set is to tweak a model’s hyperparameters-higher-level structural settings that cannot be directly learned from data. These settings can express, for instance, how complex a model is and how fast it finds patterns in data.

66-80 percent 20-33 percent

Test Set

Training Set

Validation Set

Fig 2.2 – Data Splitting

2.3.7 Modelling

After pre-processing the collected data and splitting it into three subsets, we can proceed with model training. This process entails “feeding” the algorithm with training data. An algorithm will process data and output a model that can find a target value (attribute) in new data - an answer you want to get with predictive analysis. The purpose of model training is to develop a model.

2.3.8 Model Deployment

Deployment is the method by which you integrate a machine-learning model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the machine learning life cycle and can be one of the most cumbersome. Often, an organization’s IT systems are incompatible with traditional model-building languages, forcing data scientists and programmers to spend valuable time and brainpower rewriting them.

2.4 Nonfunctional Requirements

2.4.1 Usability

A multiple disease prediction system has various usability factors, some of which include:

1. Early Diagnosis: The system can help in the early diagnosis of multiple diseases, allowing healthcare providers to initiate treatment early, which can result in better patient outcomes.

2. Personalized Treatment: By providing insights into the possible diseases a patient may be susceptible to, the system can assist healthcare providers in creating personalized treatment plans for individual patients.

3. Reduced Healthcare Costs: By reducing the number of unnecessary tests, the system can help in reducing healthcare costs for both patients and providers.

4. Time Efficiency: The system can save time for healthcare providers by automating the process of disease prediction and providing quick and accurate results.

5. Improved Decision-making: The system can assist healthcare providers in making more informed decisions by providing insights into the likelihood of multiple diseases based on patient data.

6. Accessibility: The system can be accessed from anywhere with an internet connection, allowing healthcare providers to access it from remote locations.

Overall, the multiple disease prediction system can improve patient outcomes, reduce healthcare costs, and increase efficiency in healthcare provision by providing accurate disease predictions and insights.

2.4.2 Reliability

This software will be developed with machine learning, feature engineering, and deep learning

techniques. So, in this step, there is no certain reliable percentage that is measurable. Also, user-provided data will be used to compare with results and measure reliability. With recent machine learning techniques, user-gained data should be enough for reliability if enough data is obtained.

2.4.3 Performance

Processing time and response time are as little as possible providing the result at a faster rate when compared to other methods.

2.4.4 Supportability

The system should require Python knowledge to maintain. If any problem acquires on the user side and deep learning methods, it requires code knowledge and deep learning background to solve.

2.5 Project Plan

2.5.1 Team Members

* Ishita Agrawal

Roll no. - 0829CS191060

* Ankita Vishwas

Roll No. - 0829CS191020

* Tanuja Upadhyay

Roll No. – 0829CS191060

2.5.2 Division of Work

Ankita Vishwas

* Project Management

Tanuja Upadhyay

* Software Testing

Ishita Agrawal

* Freezing og algorithm
* Framework
* Software
* Coding and Implementation

2.5.3 Time Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint 1 | Sprint 2 | Sprint 3 | Sprint4 |
| December 2022 | Jan 2023 – Feb 2022 | March 2022 | April 2022 |
| Discussion of Project Selection | Literature survey, Synopsis, SDD, Research Notes | Freezing of algorithm, framework, and software | Coding and implementation, Testing and adding last-minute features |

Table 2.3 – Time Schedule

3. Analysis

* 1. Methodology Used

Agile:

Agile is a process by which a team can manage a project by breaking it up into several stages and involving constant collaboration with stakeholders and continuous improvement and iteration at every stage. It promotes continuous iteration of development and testing throughout the software development life cycle of the project. Both development and testing activities are concurrent.

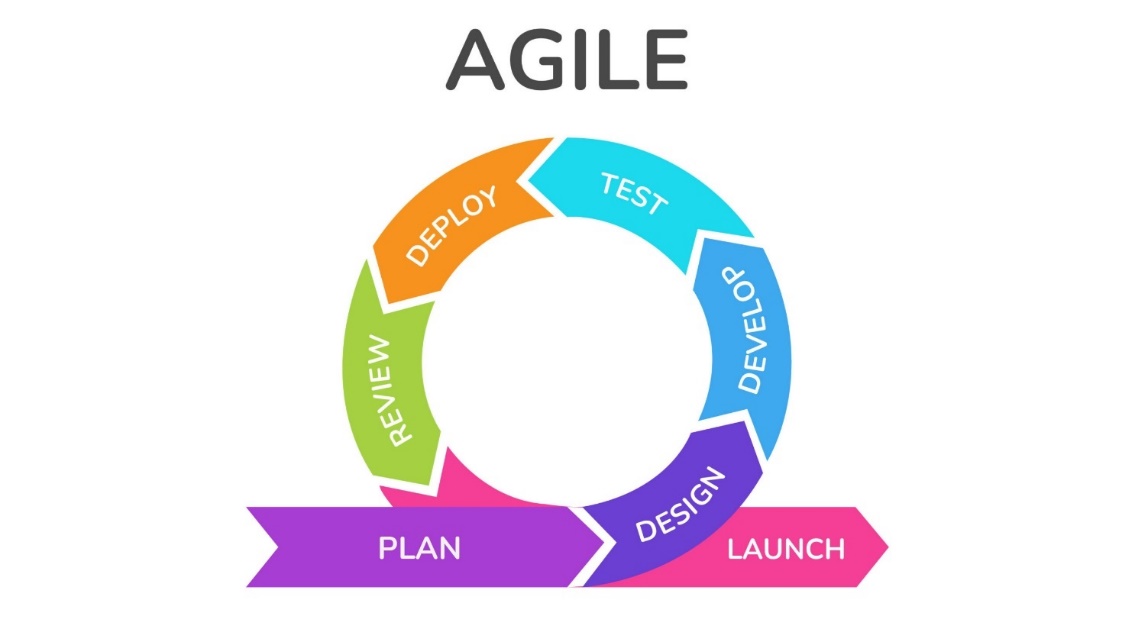


Fig 3.1 – Agile Methodology

Scrum:

SCRUM is an agile development method that concentrates specifically on how to manage tasks within a team-based development environment. Scrum encourages teams to learn through experiences and self-organize while working on a problem.

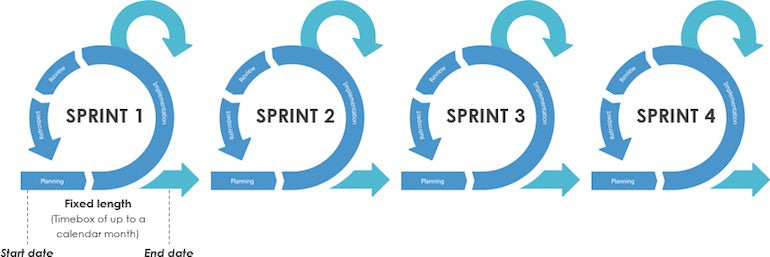
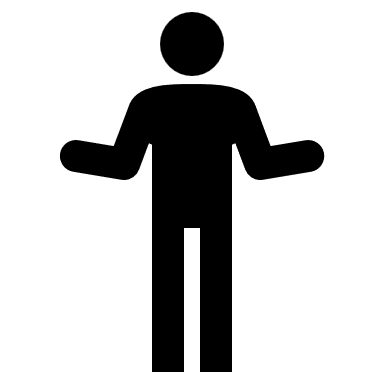


Fig 3.2 - Scrum

* 1. Use Case diagram

Though use cases are not recommended by agile development or feature-driven development, use cases were created to better understand the requirements of the system. The use cases are packaged according to domain areas. A detailed description of only some of the use cases is present in this section.



ParkinsonsDisease Prediction

Heart Disease Prediction

Diabetes Disease Prediction

The person have Parkinsons disease

The person does not have Parkinsons Disease

The person does not have Heart Disease

The person have a Heart Disease

The person does not have Diabetes

The person have Diabetes

Fig 3.3 – Use Case Diagram

Method Invocations

Events

* 1. Data Flow Diagrams

MODEL

-Encapsulates application state

-Responds to state queries

-Exposes application functionality

-Notifies views of changes

State Change

State Querry

Change

Notification

View Selection

CONTROLLER

-Defines application behavior

-Maps user actions to model updates

-Select view for response

-Uses one for each functionality

VIEW

-Renders the models

-Requests updates from models

-Sends user gestures to controller

-Allows controller to select view

User

Gestures

Fig 3.4 – Data Flow Diagram

* 1. ER Model

Fig 3.5 – ER Model

Check Accuracy Score

ML Model

Test Data

Train Data

Analyze the dataset

Fill the missing data

Data Missing ?

Import Database

Import Patient’ s Documents

* 1. Process Specification

Fig 3.6 – Process Specification

Prediction of Heart Disease

Comparison of Results

Collection of Results

Applying LR Model

Applying Decision

Tree Algorithm

Applying SVM Algorithm

Testing the Data

Training the Data

Cleaned

Database

Data Pre-Processing

Patient

Database

* 1. CF Diagram

Start

The patient does not have the disease

The patient has the disease

Apply the algorithms to predict

Apply the Data Pre-Processing techniques

Collection of the Dataset for prediction process

Decise Prediction

Fig 3.7 – CF Diagram

4. Design

Speech Dataset

* 1. Architectural Design
     1. System Architecture Diagram

Pre-Processing Data

Training Data

Apply ML algorithms

LR

SVM

Test Data

Output

Fig 4.1 – System Architecture Diagram

* + 1. Description of Architectural Design

Speech Dataset:

The main aim of this step is to spot and acquire all data-related problems. during this step, we'd like to spot the various data sources, as data are often collected from various speech dataset pre-processing data training data apply machine learning algorithms SVM and logistic regression test data output sources like files and databases. The number and quality of the collected data will determine the efficiency of the output. The more is going to be the info, the more accurate is going to be the prediction. We’ve collected our data from the Kaggle website.

Data Pre-Processing

The main aim of this step is to study and understand the nature of data that was acquired in the

previous step and also to know the quality of data. Real-world data generally contains noises,

and missing values, and may be in an unusable format that cannot be directly used for machine learning models. Data pre-processing is a required task for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model. Identifying duplicates in the dataset and removing them is also done in this step.

Training data:

Splitting the dataset into a Training set and testing set:

In machine learning data preprocessing, we have to break our dataset into both a training set and a test set. This is often one of the crucial steps of knowledge preprocessing as by doing this, we will enhance the performance of our machine learning model.

If we train our model all right and its training accuracy is additionally very high, but we offer a replacement dataset there too, then it'll decrease the performance. So we always attempt to make a machine learning model which performs well with the training set and also with the test dataset.

Apply Machine Learning Algorithms

Now, we've both the train and test data. The subsequent step is to spot the possible training methods and train our models. As this is often a classification problem, we've used two different classification methods SVM and Logistic Regression. Both algorithms have been run over the Training dataset and their performance in terms of accuracy is evaluated alongside the prediction wiped out of the testing data set.

* + - * 1. Logistic Regression

Logistic regression is additionally one of the foremost popular Machine Learning algorithms, which comes under the Supervised Learning technique. it's used for predicting the specific variable employing a given set of independent variables. It becomes a classification technique only a choice threshold is brought into the image. The setting of the edge value may be a vital aspect of Logistic regression and depends on the classification problem itself.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, whether a mouse is obese or not based on its weight, etc.



Fig 4.2 – S-Shaped Curve

Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence algorithm is termed a Support Vector Machine. Consider the below diagram in which two different categories are classified using a decision boundary or hyperplane:



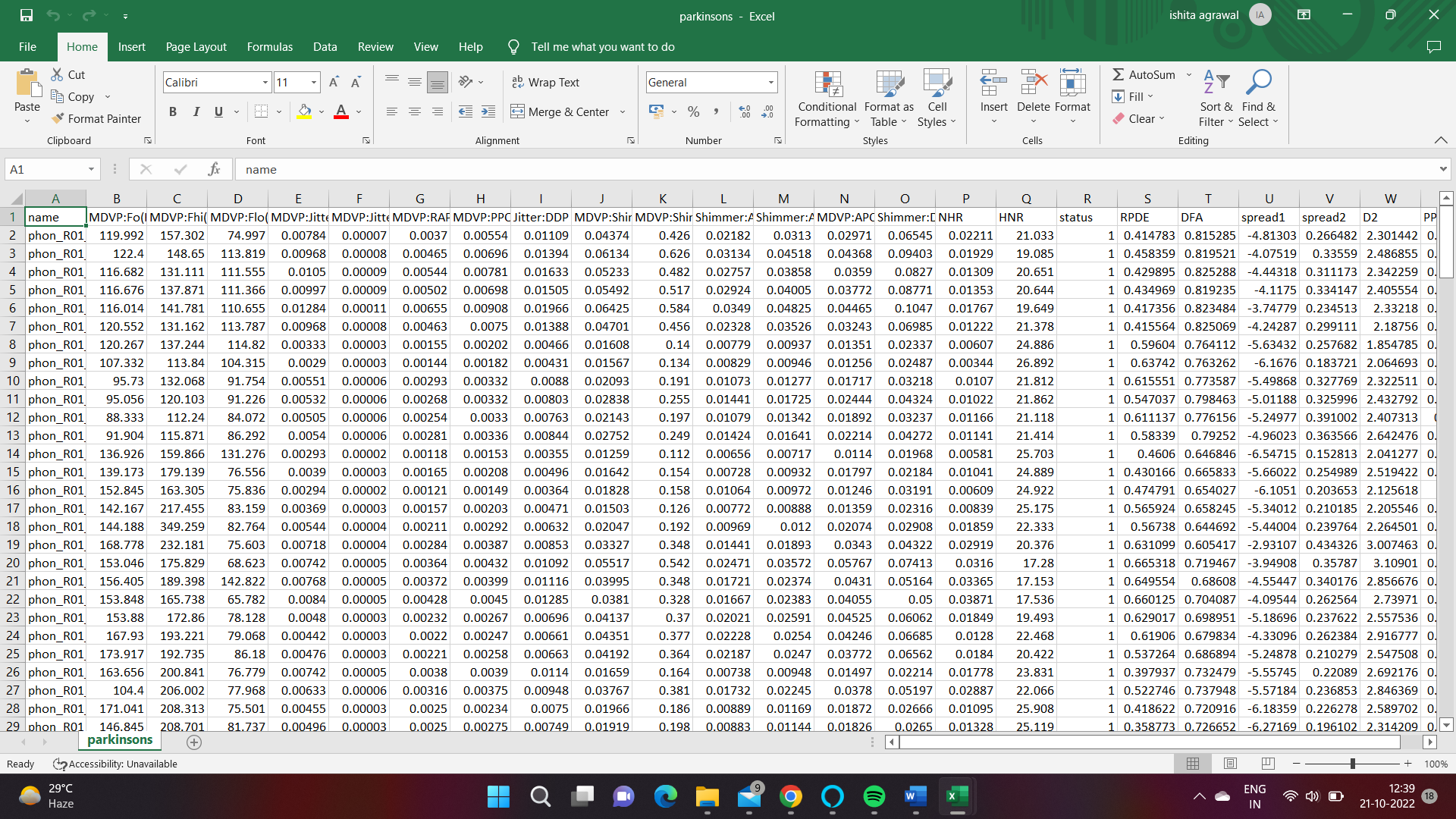
Fig 4.3 - SVM

Testing Data

Once the disease’s Prediction model has been trained on the pre-processed dataset, then the model is tested using different data points. In this testing step, the model is checked for correctness and accuracy by providing a test dataset to it. All the training methods need to be verified for finding out the best model to be used. After fitting our model with training data, we used this model to predict values for the test dataset. These predicted values on testing data are used for model comparison and accurate calculation.

User Interface

Our Front-End implementation is completed using Python and Streamlit Framework in Scientific Python Development Environment (Spyder). The user interface is extremely essential for any project because everyone who tries to utilize the system for a purpose will attempt to access it using an interface. Indeed, our system also features a user interface built to facilitate users to utilize the services we provide. Where we have used Streamlit a Python library, utilized for creating websites. Streamlit gives the sorts of choices when developing web applications, it provides you with tools, methods, and mechanics that allow you to build, and create various applications but it will not enforce any dependencies or tell you the way how the project should look like.

* 1. Database DesignFig 4.4(a) – Parkinsons Disease Database

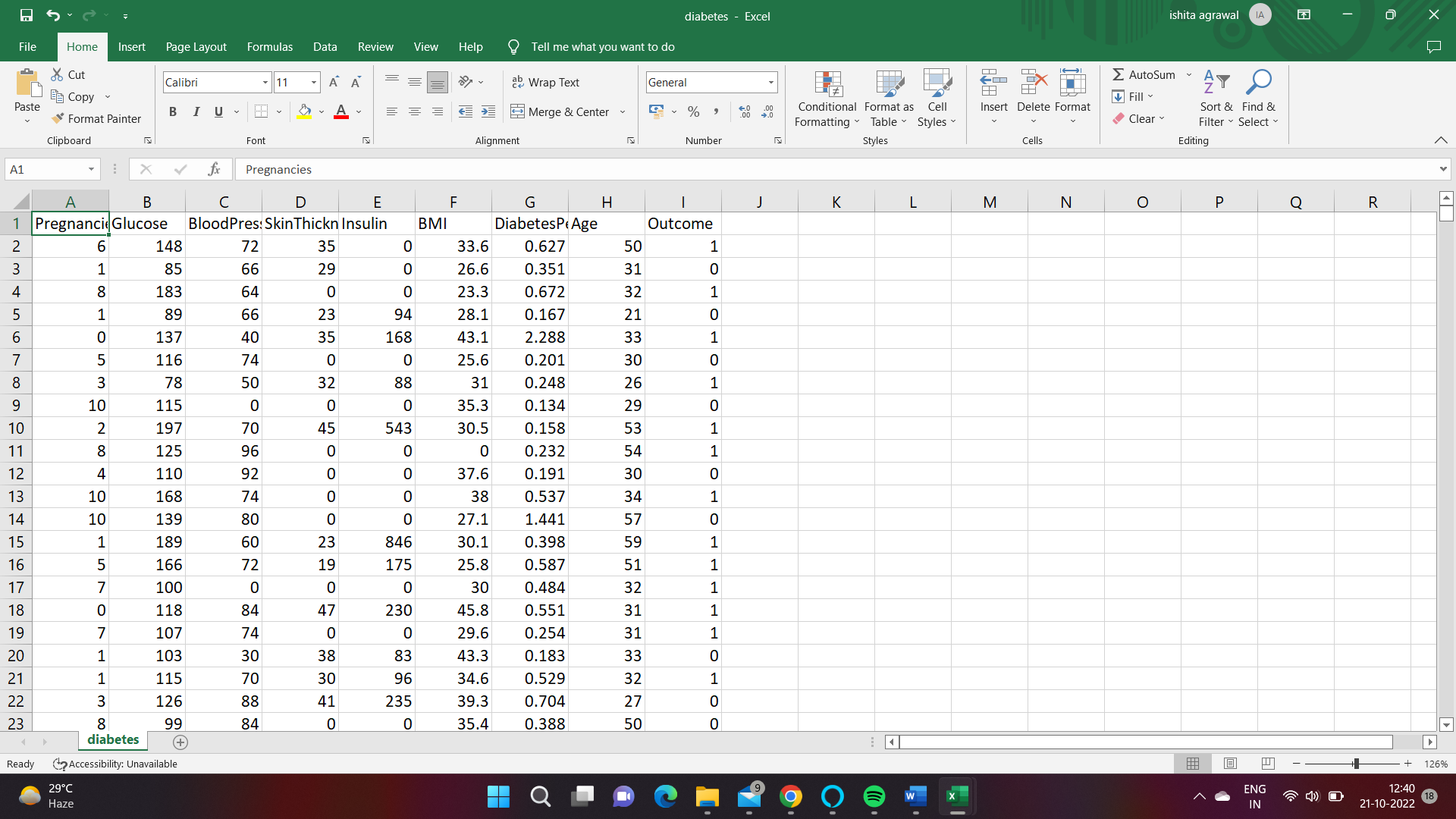


Fig 4.4(b) – Diabetes Disease Database

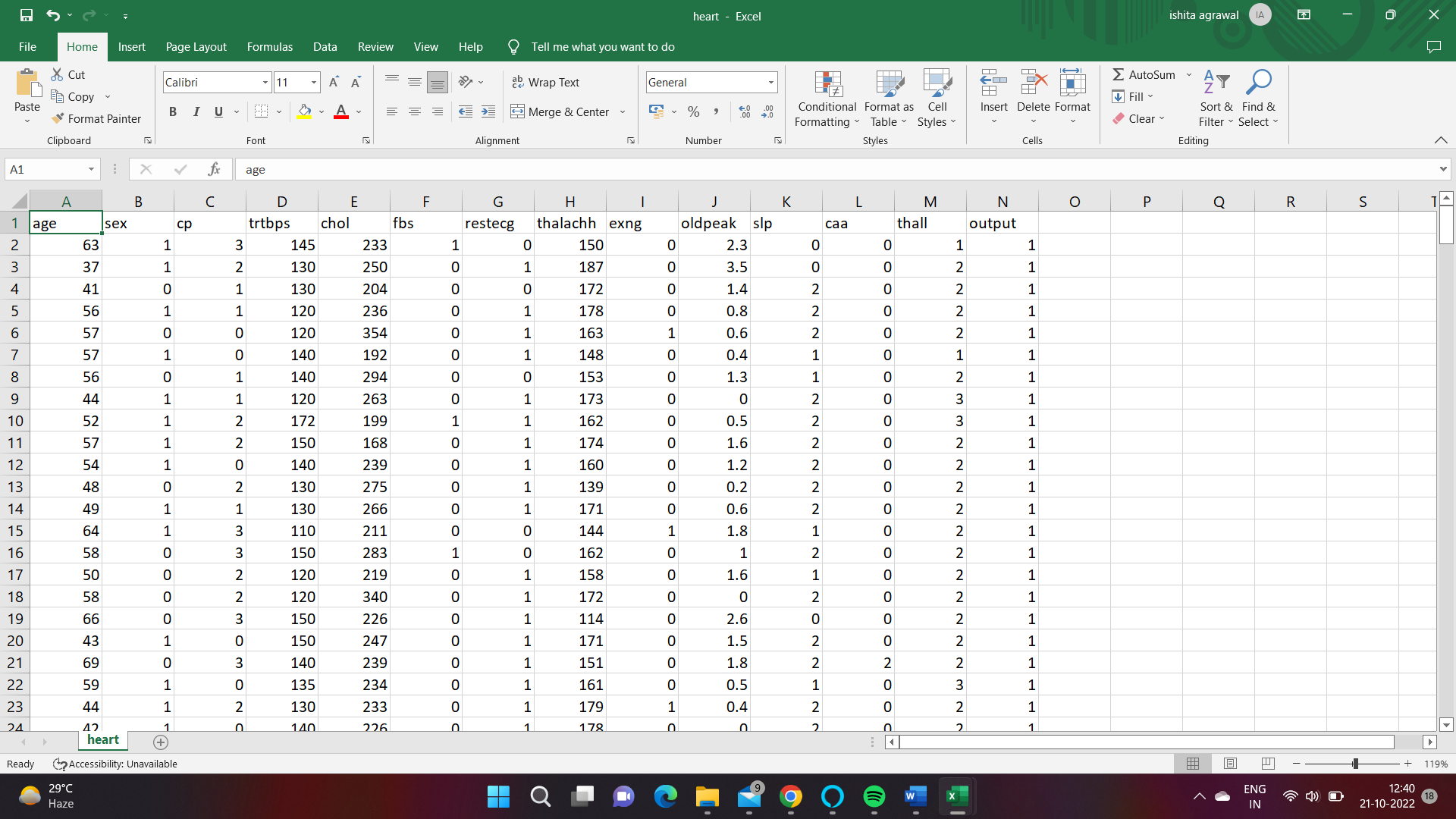


Fig 4.4(c) – Heart Disease Database

* 1. Component Design
     1. Flow Chart

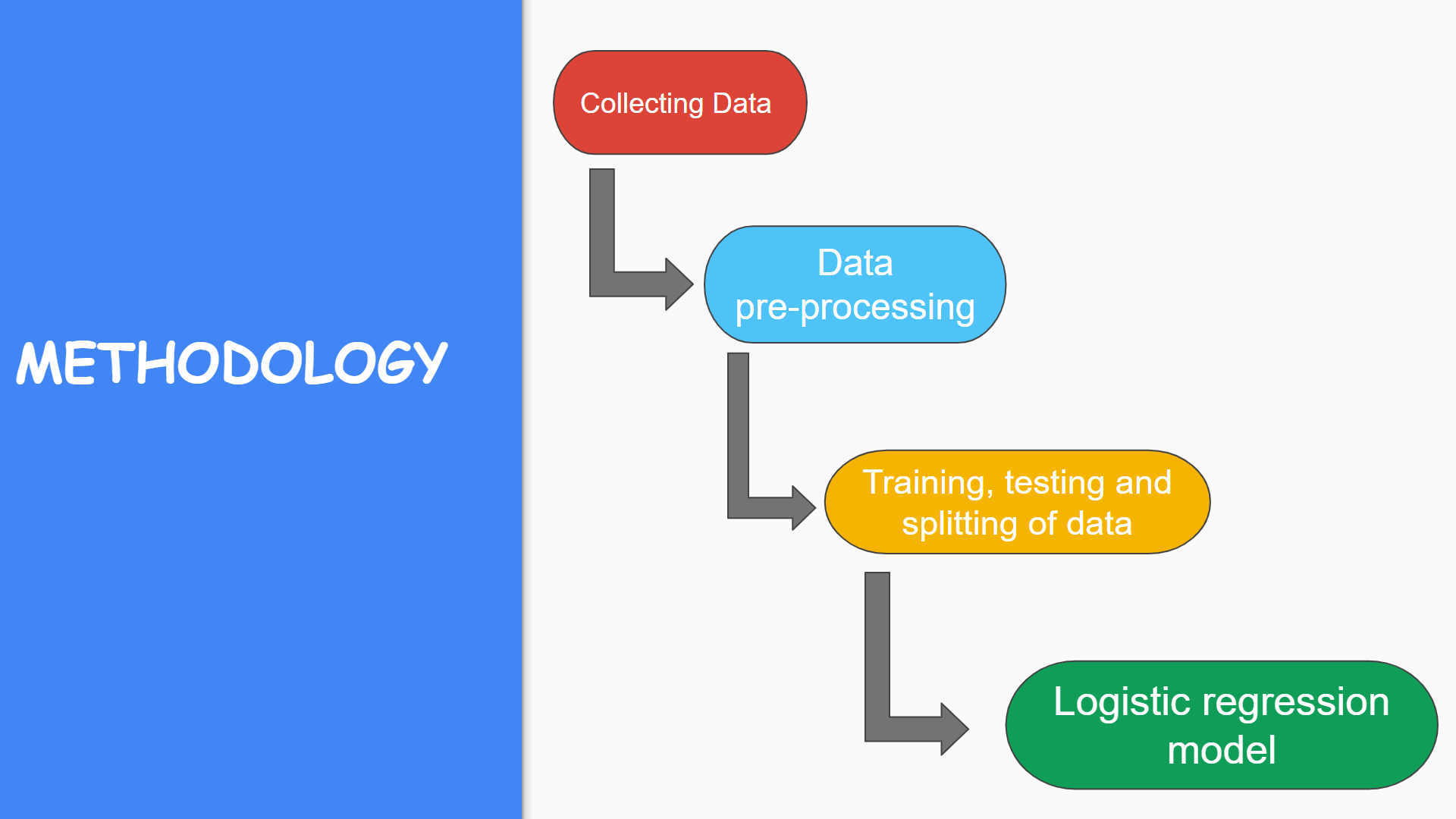


Fig 4.5 – Methodology

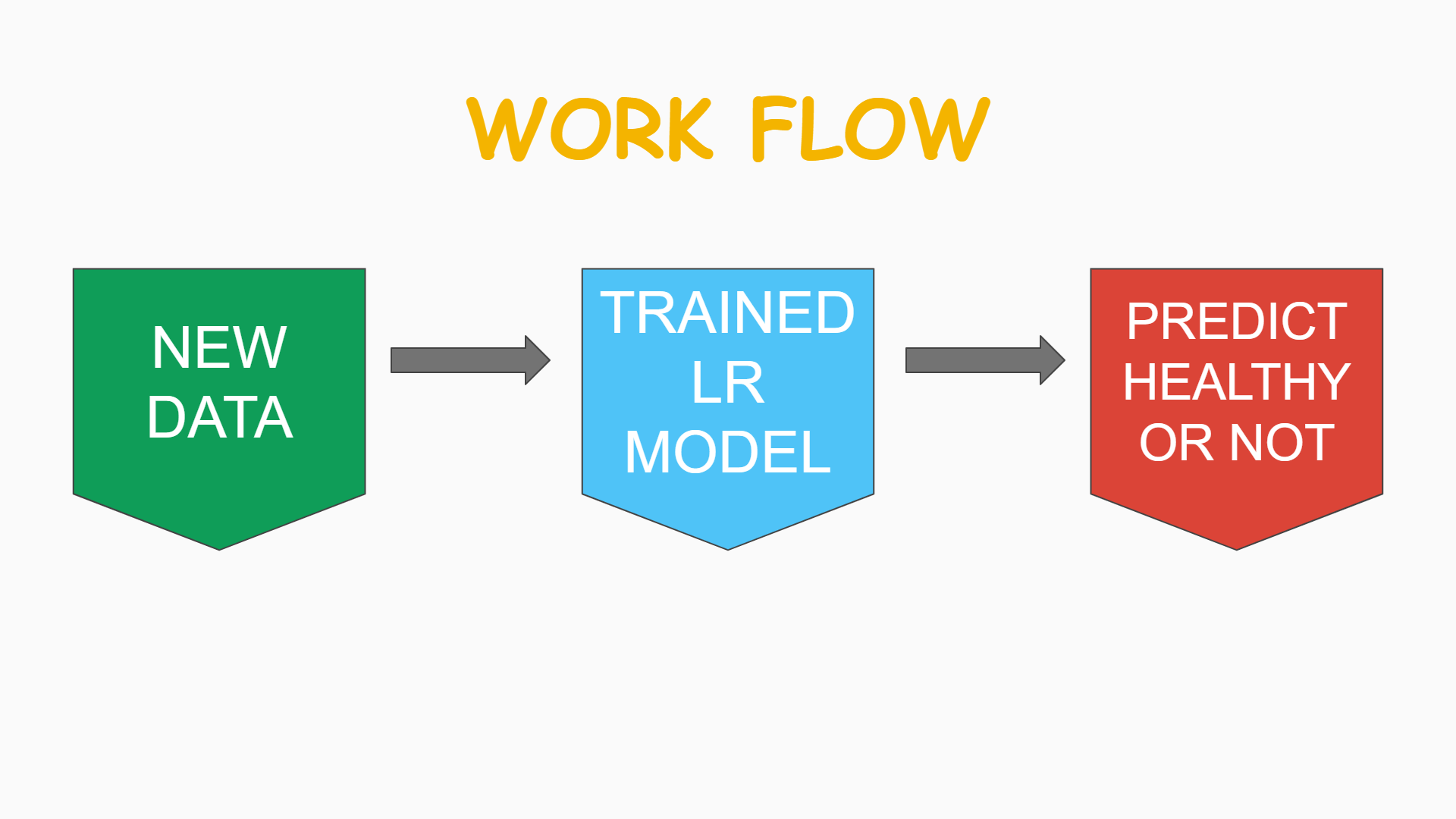
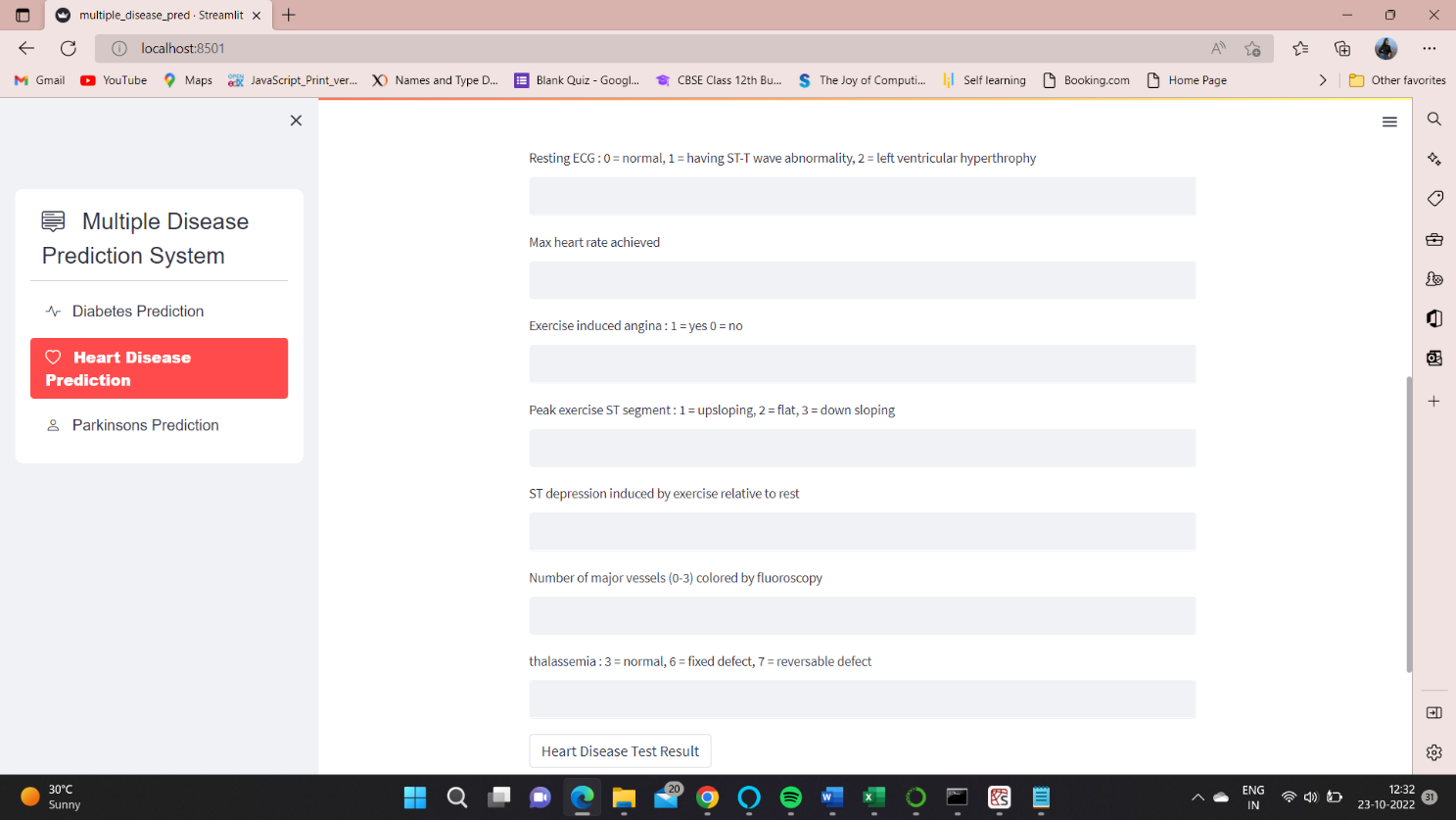
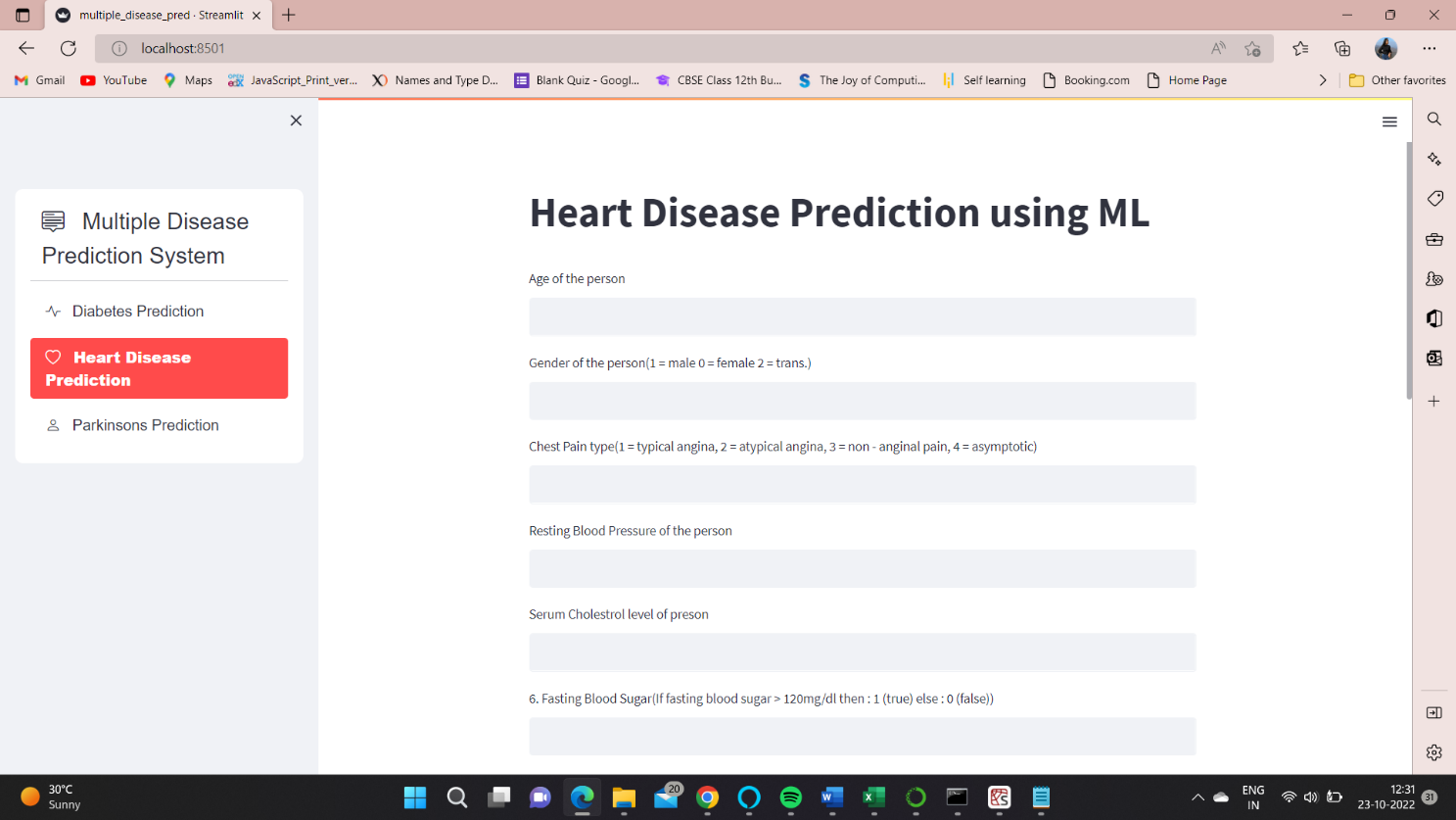
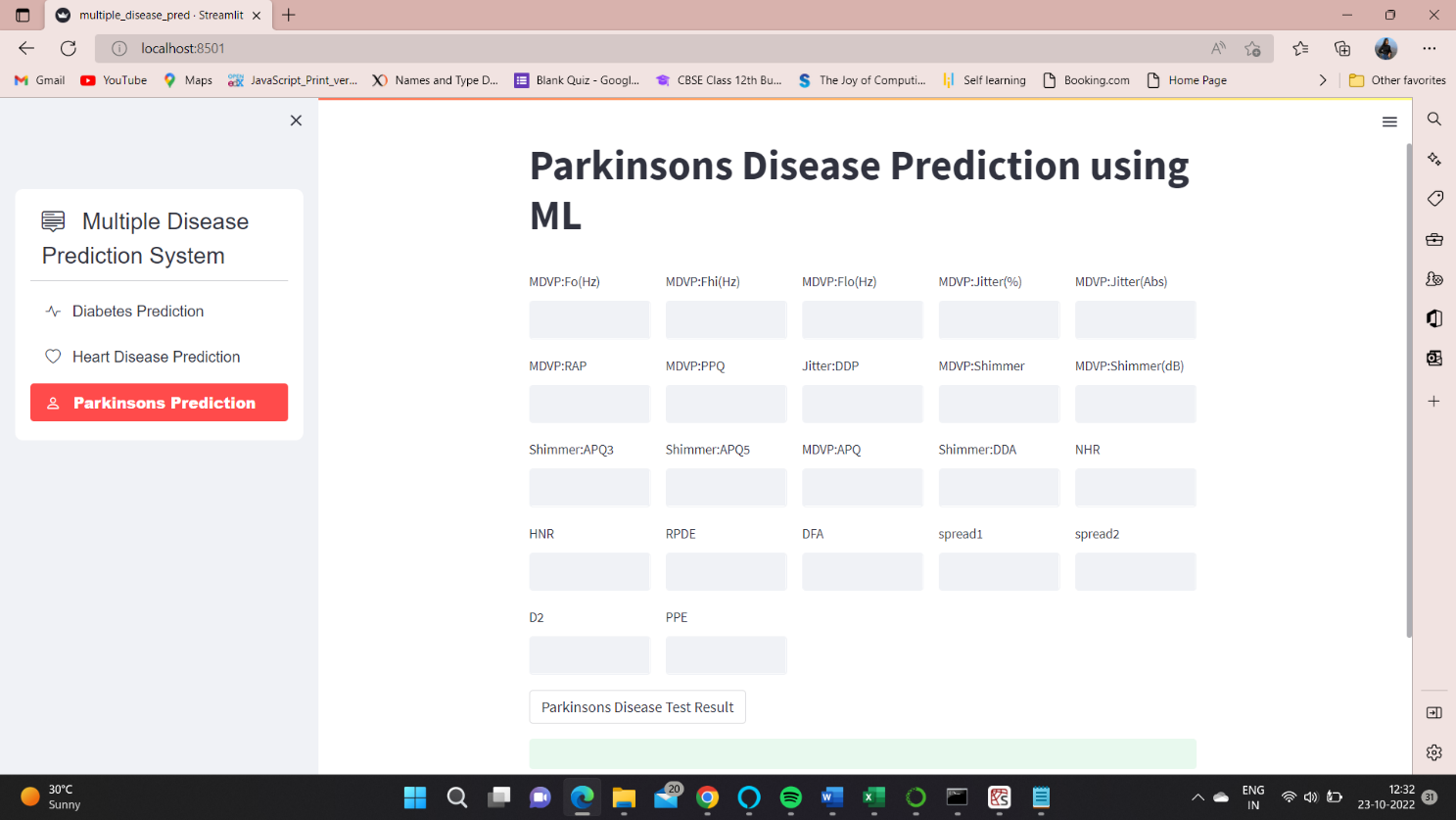


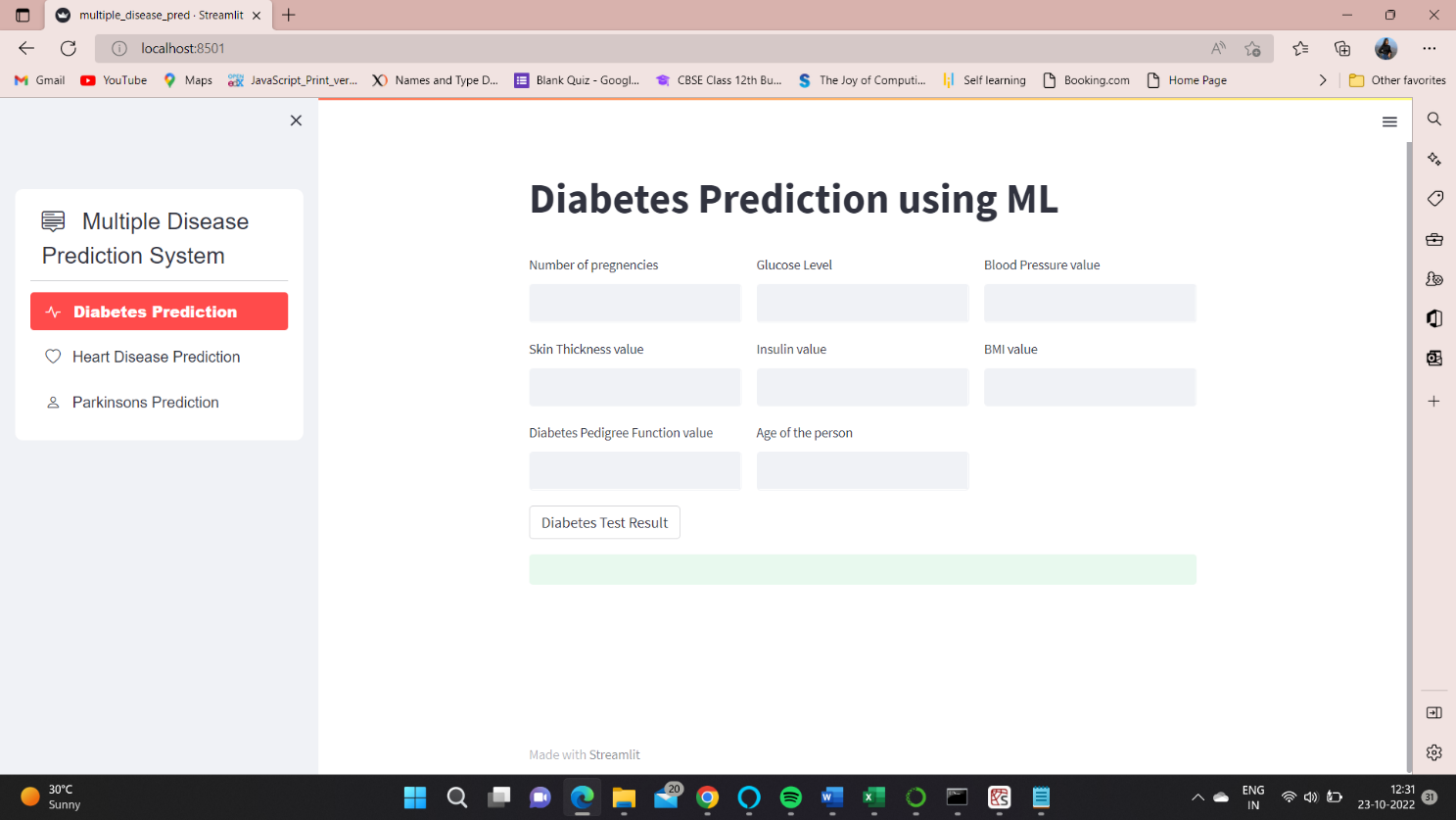
Fig 4.6 – WorkFlow

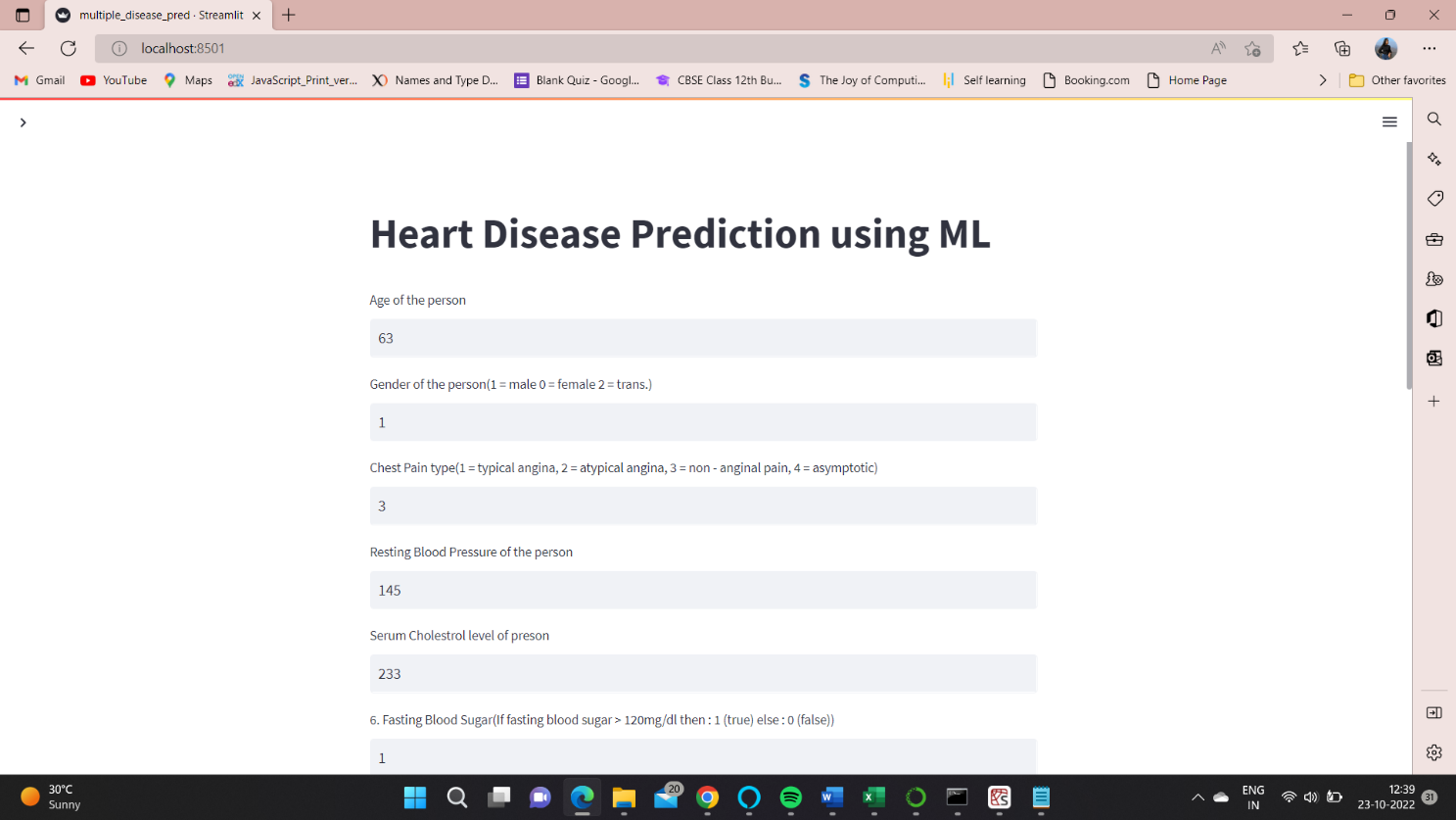
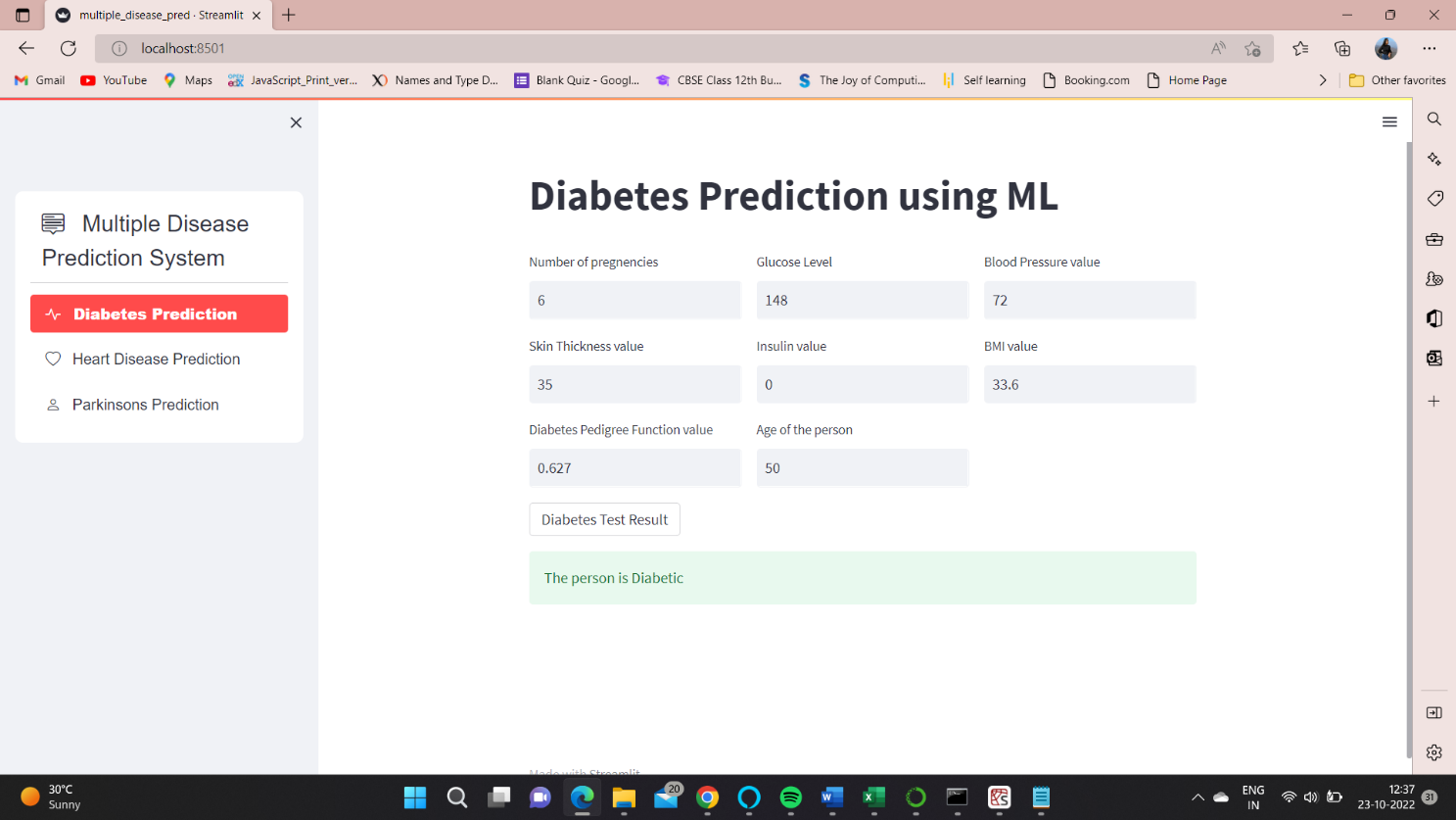
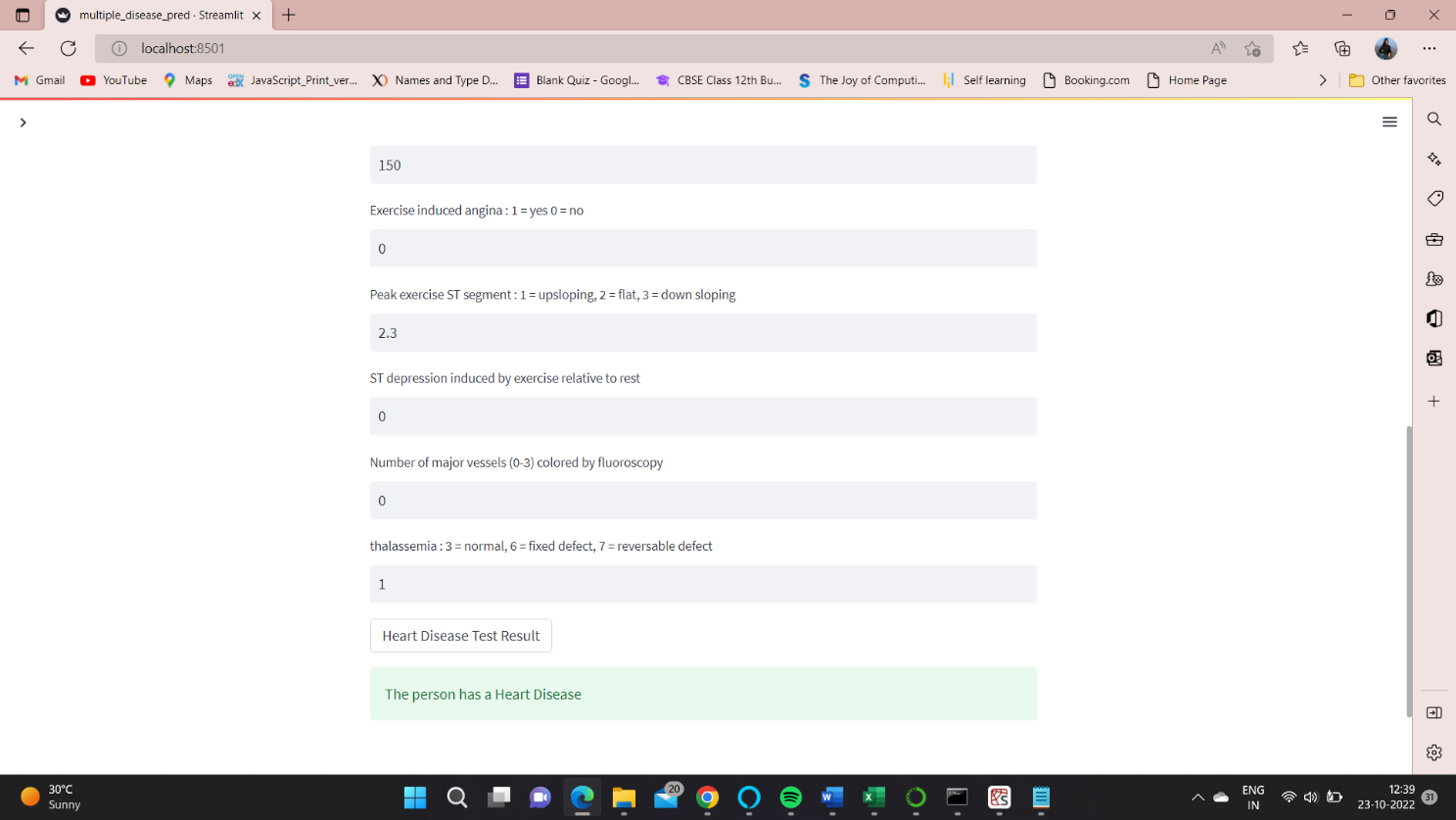
* 1. Interface Design
     1. Screenshots

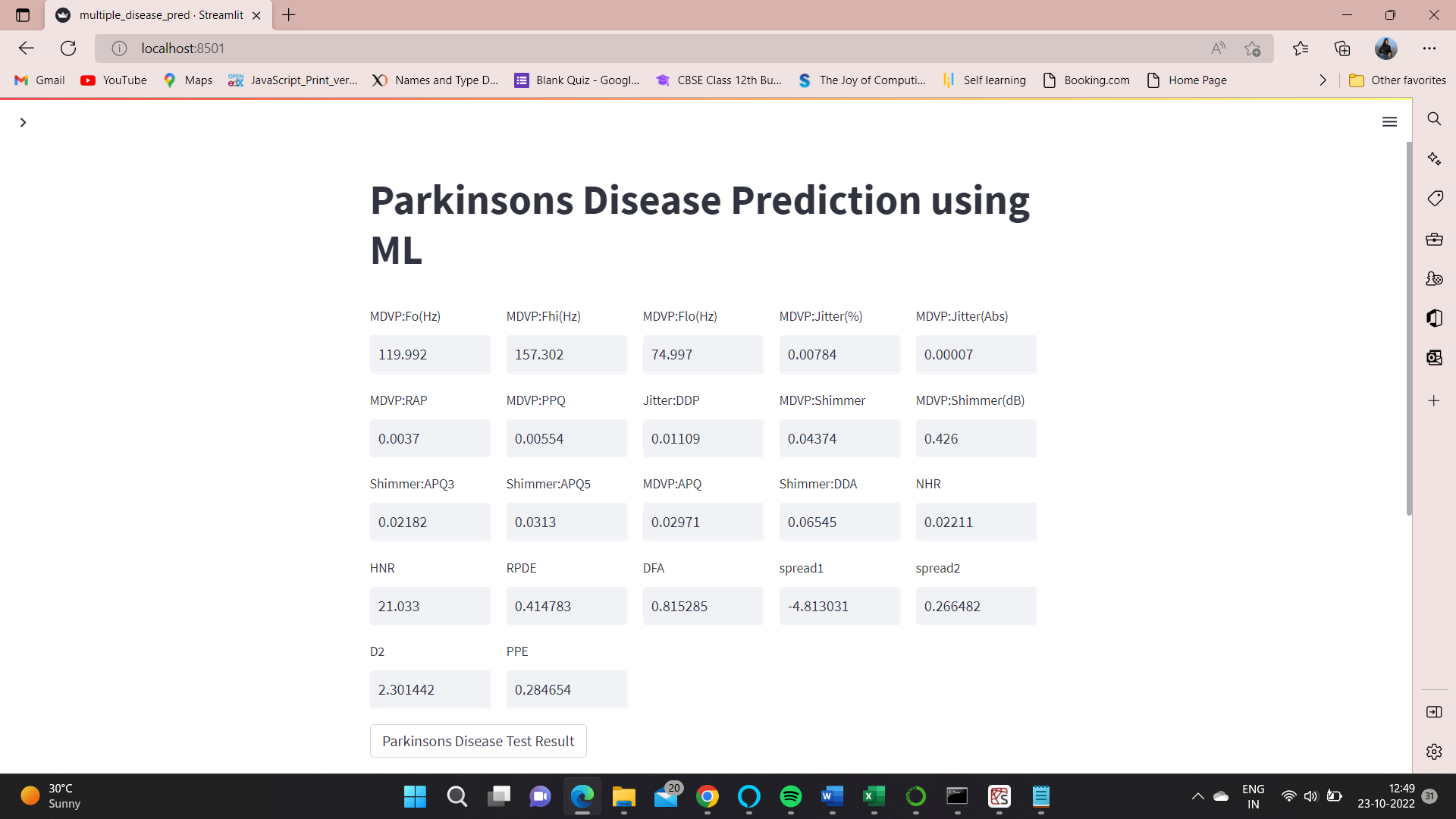








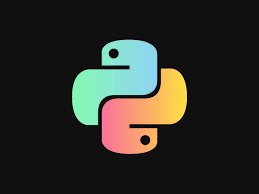


5. Implementation

5.1 Language and database system used for the implementation

5.1.1 Python

Python is a popular high-level programming language that is widely used for general-purpose programming, data science, machine learning, web development, scientific computing, and more. It was created by Guido van Rossum and first released in 1991. One of the defining features of Python is its simplicity and readability. Its syntax is designed to be easy to read and write, with a focus on reducing the amount of code needed to accomplish a task. This makes Python a great language for beginners to learn and for rapid prototyping. Python is also known for its vast collection of libraries and frameworks, which make it a powerful tool for a wide range of applications. Some of the most popular libraries in Python include NumPy and Pandas for data analysis, Matplotlib, and Seaborn for data visualization, TensorFlow and PyTorch for machine learning, and Flask and Django for web development. Python is an interpreted language, which means that code is executed line by line by an interpreter rather than compiled into machine code. This makes it easier to develop and debug code, as changes can be made and tested quickly without the need for compiling.



5.1.2 Streamlit

Streamlit is an open-source framework for building interactive data science web applications. It simplifies the process of building data-driven applications by providing pre-built widgets and components that allow developers to quickly create and deploy applications without needing to write complex web code. Streamlit enables developers to build and share applications that are easy to use, intuitive, and provide rich and interactive data visualizations. With Streamlit, data scientists and developers can create customized web applications that allow users to interact with data in real time, visualize results, and gain insights from their data. It is a popular choice among data scientists, machine learning engineers, and developers for building data-driven applications with minimal code.



5.1.3 Kaggle Database

Kaggle is a popular online platform for data science and machine learning competitions, as well as a community for data scientists to collaborate and share their work. It was founded in 2010 by Anthony Goldbloom and Ben Hamner and acquired by Google in 2017.

One of the main features of Kaggle is its competition, which allows data scientists to compete with each other to develop the best models for a given problem. Kaggle provides a wide range of datasets, both public and private, that users can download and use for their projects. These datasets cover a wide range of topics, including social media, healthcare, finance, and more. Users can also upload their datasets and make them available to the community. In addition to competitions and datasets, Kaggle also provides a community forum where users can ask and answer questions, share their work, and collaborate on projects.

Kaggle has become a popular platform for companies and organizations to host data science challenges and recruit talent. It has also been used by researchers to develop models for scientific applications, such as drug discovery and genomics. Overall, Kaggle has become a valuable resource for the data science and machine learning community, providing a platform for collaboration, learning, and innovation. Its competitions, datasets, and community forum have helped advance the field of data science and machine learning and democratize access to these technologies.



5.2 Features of Language and database used for the Project

* + 1. Features of Python
* Easy to learn: Python is designed to be easy to learn and understand. Its syntax is simple and intuitive, making it accessible to beginners.
* Interpreted language: Python is an interpreted language, which means that code can be executed directly without the need for compilation. This makes it easy to write and test code quickly.
* Cross-platform compatibility: Python is a cross-platform language, which means that it can run on different operating systems, such as Windows, macOS, and Linux.
* Large standard library: Python comes with a large standard library that provides a wide range of functionality for tasks such as web development, data analysis, and scientific computing. This reduces the need for external libraries and makes it easy to get started with Python.
* Dynamic typing: Python is a dynamically typed language, which means that variable types are determined at runtime. This makes it easier to write code quickly, but can also lead to type errors if not used carefully.
* Object-oriented programming: Python supports object-oriented programming, allowing developers to create reusable and modular code.
* High-level language: Python is a high-level language, meaning that it abstracts away many low-level details, making it easier to focus on the logic of a program.
* Extensible: Python can be extended with external libraries written in other languages such as
* C and C++. This allows developers to use existing libraries and tools for tasks such as scientific

computing and machine learning.

* + 1. Features of Database
  1. Description of the third-party tools used

Jupyter Notebook is an open-source web application that is widely used for data science and machine learning tasks in Python. It provides an interactive environment where users can write, execute and test code in real time, making it an excellent tool for prototyping and experimenting with new ideas.

The notebook interface supports various types of cells, including code cells, markdown cells, and raw cells. Code cells allow users to write and execute Python code snippets, while markdown cells enable users to write formatted text, equations, and documentation. Raw cells allow users to write arbitrary content without any processing.

Jupyter Notebook integrates seamlessly with various Python libraries and frameworks, including NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, PyTorch, and many others. Users can import these libraries into their notebooks and use them to manipulate data, visualize results, and train machine-learning models.



Jupyter Notebook also supports various keyboard shortcuts and built-in tools that make it easy to work with code, data, and visualizations. Users can export their notebooks to various formats, including HTML, PDF, and LaTeX, and share their work with others who may not have access to Jupyter Notebook.

Overall, Jupyter Notebook is a powerful tool for Python developers, data scientists, and machine learning engineers. Its interactive and flexible nature makes it an excellent choice for prototyping and experimenting with new ideas, while its integration with Python libraries and frameworks makes it an essential tool for data science workflows.

6. Testing

6.1 Features to be tested

Several features should be tested for a multiple disease prediction system to ensure that it is accurate, reliable, and effective. Some of these features include:

* + - * Data quality: The system should be tested to ensure that it can handle large amounts of patient data and that the data is accurate, complete, and relevant.
      * Accuracy: The accuracy of the system should be tested using different evaluation metrics such as sensitivity, specificity, and area under the ROC curve to ensure that it provides reliable predictions for different diseases.
      * Speed and efficiency: The system should be tested to ensure that it can provide predictions in a reasonable amount of time, even with large amounts of patient data.
      * User-friendliness: The system should be tested to ensure that it is easy to use and that users can interact with it without requiring extensive training or technical knowledge.
      * Robustness: The system should be tested to ensure that it can handle various input types, handle missing data, and provide accurate predictions even when presented with noisy or incomplete data.
      * Security: The system should be tested to ensure that it has appropriate security measures to protect patient data from unauthorized access or breaches.
      * Scalability: The system should be tested to ensure that it can handle large amounts of patient data and can be scaled up as the volume of data increases over time.

6.2 Features not to be Tested

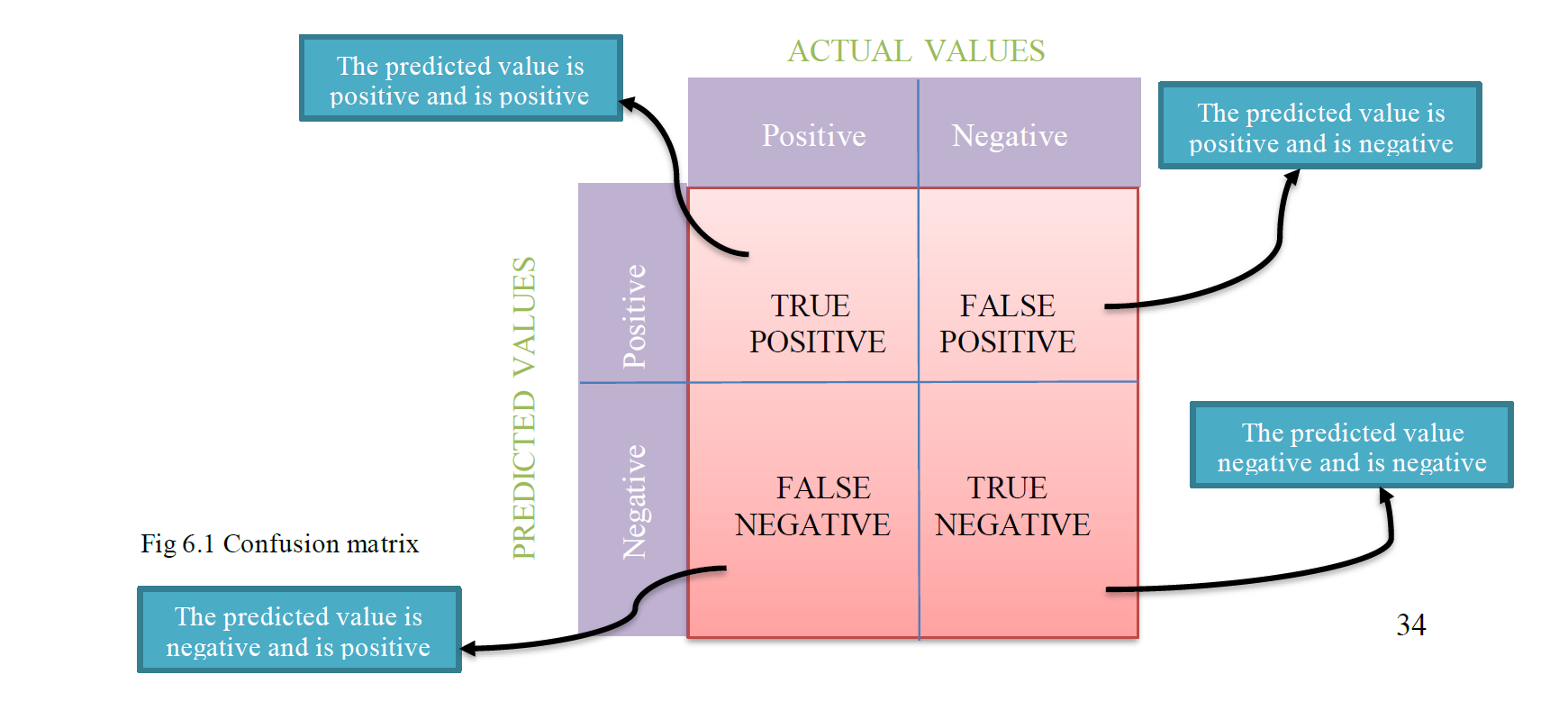
When testing a multiple disease prediction system, some features should not be tested. These include:

* + External factors: The system should not be tested for factors outside of its control, such as the accuracy of the medical data provided or the actions of healthcare professionals.
  + User interface design: While the user interface is important, testing it is not directly related to the system's ability to predict diseases accurately. User interface design should be tested separately from the system's functionality.
  + Third-party integrations: If the system integrates with other healthcare systems or third-party software, these integrations should be tested separately from the disease prediction system.
  + Network connectivity: Testing network connectivity and infrastructure is important, but it is not directly related to the system's ability to predict diseases accurately.
  + Hardware performance: The performance of the hardware used to run the system should be tested separately and is not directly related to the system's ability to predict diseases accurately.
  + Legal and ethical considerations: While legal and ethical considerations are important, they are not directly related to the system's functionality and should be addressed separately.

6.3 Testing Techniques

6.3.1 Confusion Matrix

A confusion matrix can be used to evaluate the performance of the classification model. It shows the number of true positives, true negatives, false positives, and false negatives. This can help identify any areas where the model is making mistakes.



6.3.2 Cross-Validation

Cross-validation is a technique used to evaluate the performance of a classification model. It involves splitting the data into training and testing sets and evaluating the model on multiple different splits of the data.



Fig 6.2 Cross Validation

**CONCLUSION**

We developed a Prediction Engine that enables the user to check whether he/she has diabetes/ heart disease/ Parkinsons Disease. The user interacts with the Prediction Engine by filling out a form that holds the parameter set provided as input to the trained models. The Prediction engine provides an optimal performance compared to other states of art approaches. The Prediction Engine makes use of two algorithms to predict the presence of a disease namely: Logistic Regression and Support Vector Machine (SVM). The reasons to choose these algorithms are:

* It is effective if the training data is large.
* A single dataset can be provided as an input to this algorithm with minimal or no modification.
* A common scalar can be used to normalize the input provided to this algorithm.

**FUTURE SCOPE**

* To enhance the functionality of the prediction engine providing the details of 5 nearest hospitals or medical facilities to the user input location.
* Provide a user account that allows the user to keep track of their medical test data and get suggestions or support to meet the right specialists or the tests to be taken.
* Provide admin controls to upload, and delete the dataset which will be used to train the model.
* Automate the process of training the model and extracting pickle files of the trained models which will be consumed by the APIs to predict the disease.
* Mail the detailed report of the prediction engine results along with the information of 5 nearest medical facilities details having location and contact information.

**REFERENCES**

1. *Priyanka Sonar, Prof. K. JayaMalini*, “Diabetes Prediction Using Different Machine Learning Approaches”, Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019) IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4.
2. *Samrat Kumar Dey, Ashraf Hossain, Md. Mahbubur Rahman*, “Implementation of a Web Application to Predict Diabetes Disease: An Approach Using Machine Learning Algorithm”, 2018 21st International Conference of Computer and Information Technology (ICCIT).
3. *Deeraj Shetty, Kishor Rit, Sohail Shaikh, Nikita Patil*, “Diabetes Disease Prediction Using Data Mining”, 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS).
4. *Zhilbert Tafa, Nerxhivane Pervetica, Bertran Karahoda*, “An Intelligent System for Diabetes Prediction”, 4thMediterranean Conference on Embedded Computing MECO – 2015 Budva, Montenegro.