Final Project

On

“Disease Prediction using Machine Learning”

Prepared by:

885189357 | Tharunaa Shoban Babu | tharunaa@csu.fullerton.edu

885189761 | Gouri Babasaheb Sabale | gourisabale123@csu.fullerton.edu

Report Submitted to

Prof. Mrs. Lidia Morrison
Department of Computer Science
College of Engineering and Computer Science (ECS)
California State University, Fullerton (CSUF)
for the Fulfillment of the Requirements for the

CPSC 597: Project
California State University, Fullerton
May 2024

Table of contents

Table of Figures

Abstract

Addressing minor health issues often requires humans to personally visit the hospital for check-ups which leads to time consumption, handling over the call doctor appointments gets hectic and often people tend to search the symptoms over the internet to self-diagnose, which can yield inaccurate results. The increased burden on healthcare systems, leveraging data-driven disease prediction has become crucial for systematic and cost-efficient healthcare. The growing volume of medical data makes it essential to explore different effective data mining techniques to find valuable patterns and trends. Despite the large amounts of data being generated by Hospital Information Systems (HIS), extracting meaningful insights from diagnostic case data can be challenging. The project outlines the significance of disease prediction systems based on patient-provided symptoms, using machine learning algorithms and a user-friendly and aims to facilitate early disease prediction, helping physicians in timely diagnosis. Comparison of several machine learning algorithms are used such as random forest, decision tree classifiers, support vector machines, and naïve Bayes classifiers. The highest accuracy is used to deploy the model as a web application.

Keywords: Hospital Information Systems, Machine Learning, Random Forest, Support Vector Machines, naïve Bayes classifiers, Decision Tree.

Introduction

The human population on the earth has been increasing rapidly, as well as illnesses becoming more complex each year, healthcare systems worldwide are grappling with increasing demands and rising costs. Many medical conditions require precise clinical guidance, making accurate disease prediction a crucial aspect of effective treatment. If people are in areas with limited access to doctors and hospitals, identifying diseases can be challenging. Implementing an automated program that enables disease prediction could save both time and money, thereby simplifying the process and making it more accessible for patients.

Disease Prediction is a Machine learning - web based application that predicts the disease of the user with respect to the symptoms given by the user. The Disease Prediction system has data sets collected from different health related sites. With the help of the Disease Prediction system, the user will be able to know the probability of the disease with the given symptoms. People are constantly curious to learn new things, and internet usage is expanding daily. When an issue comes up, people always try to turn to the internet for assistance. More people than hospitals and physicians have access to the internet. People do not have immediate options when they suffer from a particular disease. Given that people have access to the internet around-the-clock, this approach may be useful to them.

The goal is to streamline disease prediction by offering promising prospects for the future of the healthcare system. The project will carefully examine distinct ML algorithms and their accuracy in disease prediction. The implications of such a system are profound, potentially revolutionizing medical treatment paradigms.

I. Problem statement

Develop a robust machine learning system that accurately predicts the risk of various diseases based on individual health data, develop a user-friendly platform, ensuring data privacy and regulatory compliance to enhance patient care and healthcare efficiency.

II. Project Objective

This project's goal is to create an advanced machine learning-based illness prediction system that will increase the precision and effectiveness of medical diagnostics. This system is intended to enhance patient outcomes by accurately and precisely predicting diseases by utilizing cutting-edge technology, data analytics, and machine learning. The system's overarching goals are as follows:

1. Early Diagnosis

- Early Disease Detection: This method uses a variety of data sources, such as genetic, lifestyle, and medical records, to identify diseases in their early stages.
- Risk assessment: Determines a person's likelihood of contracting a particular disease by looking at their medical history, genetic predispositions, and other relevant variables.2. Cost Reduction and Resource Optimization

2. Improved Precision of Treatment

- Support for Doctors: Provides medical practitioners with increased accuracy in disease diagnosis, particularly in cases when vague indications and symptoms are present.
- Accuracy in Disease Classification: Takes into account the intricacies of disease classification in order to provide more conclusive and accurate diagnosis.

3. Minimizing Expenses and Optimizing Resources

- Preventive Measures: Reduces long-term healthcare expenditures by focusing on early detection and preventive care while treating more severe or chronic illnesses.
- Effective Resource Allocation: Enhances the distribution of health resources by ranking interventions according to the anticipated risks and seriousness of illnesses.

III. Scope

To develop a robust and scalable machine learning-based platform for accurate disease prediction and personalized treatment recommendations. This system aims to improve patient outcomes and enhance the efficiency of healthcare systems by enabling early disease detection, which facilitates timely and proactive interventions. Efficient resource allocation and early interventions can significantly impact health management positively.

IV. Key issues

Increasing healthcare demands and rising costs are major challenges in the healthcare system. The challenge of medication recommendation and the complexities of disease classification are some of the significant issues in this domain.

One of the major key issues related to disease prediction systems is disease classification. Because many diseases have subtypes or stages, each with unique characteristics and progression patterns. Classifying patients into these subgroups is challenging but crucial for personalized treatment. It is difficult to identify diseases with overlapping or common systems. Another issue is that patient data can vary in format, quality, and completeness, making it challenging to standardize and use for classification tasks.

V. Overview of the existing system

Given the contemporary environment, which is characterized by an increase in reliance on technology, there is a greater need for clever solutions that improve accuracy. People's reliance on technology is growing as the twenty-first century goes on since it permeates every facet of daily life. Nevertheless, there is still a general lack of interest in personal health management despite this absorption in technology. Many people would rather stay out of the hospital when they have small problems that could develop into more serious ailments. As a result, there is a growing trend of using online forums with question and answer sections to handle health-related queries instead of having to go through voluminous web documentation.

To address these challenges, our project aims to develop a sophisticated disease prediction system that utilizes user-input symptoms to predict diseases and their severity. The system leverages a robust database containing labeled diseases and associated symptoms. Key to this system is the implementation of advanced machine learning algorithms, including Naïve Bayes, Decision Trees, and Random Forest. These algorithms are selected for their ability to deliver fast and efficient results, with a comparative analysis to identify the most effective among them. According to our research, Naïve Bayes, in particular, has shown remarkable accuracy, making it a preferred choice for disease prediction in large datasets.

Further research in recent years has underscored the potential of machine learning techniques in disease prediction and drug recommendation. Studies have highlighted the importance of comprehensive datasets that cover a wide array of symptoms and diseases. Various machine learning algorithms have been employed, such as Random Forest, Support Vector Machine, Decision Tree, K-Nearest Neighbors, and Naïve Bayes, with accuracy rates between 93% and 95.63%. The integration of these algorithms into our system is anticipated to significantly enhance the predictive accuracy and efficiency of diagnosing diseases. This approach not only aims to improve health outcomes but also contributes to the optimization of healthcare resources by facilitating early detection and intervention.

VI. Proposed solution

The proposed system would show enhanced performance when it comes to identifying diseases because it will use a combination of ML algorithms for accurate disease prediction. Along with the patient's existing symptoms, we will also consider domain expertise and medical history of the patient so that the right drug will be recommended to the patient.

In our proposed system, we will use a modern approach that involves the utilization of the Random Forest algorithm and Naive Bayes classifier to enhance the accuracy and reliability of disease predictions. Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. In disease prediction, this ensemble approach helps improve accuracy by reducing overfitting and capturing complex relationships in the data. Naive Bayes is

a probabilistic classifier that calculates the probability of an instance belonging to a particular class. In disease prediction, this probabilistic approach allows for the quantification of uncertainty in predictions.

The system offers a holistic solution that addresses the complexities and ambiguities associated with healthcare datasets. This innovative model not only provides high predictive accuracy but also enhances interpretability, making it a valuable tool for early disease detection.

Literature Survey

The first research paper [1] aimed to develop an effective disease prediction model using machine learning techniques in the healthcare domain. Leveraging Python and cutting-edge AI tools, the authors tackled the critical challenge of early disease diagnosis based on patient symptoms. They harnessed a substantial dataset encompassing 132 symptoms and 41 types of diseases, emphasizing the importance of robust data for machine learning success. Data preprocessing was diligently performed to ensure data cleanliness and suitability for AI algorithms. This study explored various machine learning models, including Random Forest, Support Vector Machine, Decision Tree, and Naïve Bayes, achieving accuracy rates ranging from 93% to 95.63%. Notably, Naïve Bayes emerged as the top-performing algorithm for disease prediction, demonstrating its potential for enhancing healthcare outcomes.

The proposed work in the research paper [2] focuses mainly on the development of a system for predicting illnesses based on user-provided symptoms, utilizing four distinct algorithms. This system achieves an average accuracy of approximately 94%. The primary motivation behind this system is to alleviate the congestion in hospital outpatient departments (OPDs) and ease the workload on medical staff. By offering a user-friendly and adaptable approach, this system aims to cater to the specific requirements of the medical sector. Additionally, the project emphasizes the importance of visualizing and presenting the research findings effectively.

The research paper [3] introduced an innovative machine-learning methodology. This study utilized a dataset named Disease (DD), which encompasses eight distinct classifications, including anemia, chronic renal failure, delta hepatitis, high-fat content, jaundice, lipid disorders, liver dysfunction, and normal cases. Several machine learning techniques, such as SVM, DT(Decision Trees), KNN (K-Nearest Neighbors), and NB (Naïve Bayes), were employed for analysis and prediction.

Software requirements

I. Functional requirements

1. User Authentication and Authorization

- **Description**: The system will incorporate secure authentication mechanisms to ensure only authorized users can access the application. Different access levels will be implemented for various user roles including administrators, doctors, and patients.
- **Precondition**: Users must have valid credentials corresponding to their roles.
- **Postcondition**: Access is granted based on the role, ensuring security and appropriate user experience.

2. Patient Data Management

- **Description**: The system will allow medical professionals to securely input and manage patient data. It will support the storage of demographics, medical history, symptoms, medical records, and other relevant patient data.
- **Precondition**: Medical professionals must be authenticated and authorized to access this feature.
- **Postcondition**: Patient data is securely stored and accessible for medical use, ensuring data integrity and confidentiality.

3. Machine Learning Models

- **Description**: Integrate advanced machine learning models for disease prediction using classification, regression, and clustering techniques. The system will allow for the training and periodic retraining of these models to incorporate new data and improve prediction accuracy.
- **Precondition**: Sufficient data must be available for model training.
- **Postcondition**: Machine learning models are effectively integrated, offering high accuracy and reducing the risk of overfitting.

4. Alerts and Notifications

- **Description**: Implement a comprehensive alert system that notifies healthcare professionals and patients about potential health risks and upcoming appointments. Alerts will be customized based on the severity of disease risk and the urgency of actions required.
- **Precondition**: Users are registered and have provided necessary health information.
- **Postcondition**: Timely and relevant alerts enhance patient care and proactive health management.

5. User-friendly Interface

- **Description**: The system will feature a user-friendly interface that facilitates easy navigation and interaction for all users, including patients and doctors. This interface aims to enhance user engagement and satisfaction.
- **Precondition**: Users must access the system through compatible devices.
- **Postcondition**: Users experience seamless navigation and efficient interaction with the system, enhancing overall usability and satisfaction.

II. Non-functional requirements

- Performance:
    - Response Time: Specify the higher acceptable time for a system to provide disease predictions and drug recommendations by ensuring timely results.
- Reliability:
    - Availability: Define the minimum acceptable time to ensure the system is readily accessible to medical professionals.
    - Fault Tolerance: Specify measures for handling errors to reduce downtime and data loss.

- Security:
    - Data Privacy: Ensure confidentiality and privacy of patient data by adhering to healthcare standards and regulations.
    - Access Control: Implement role-based access control to restrict sensitive information access based on user roles.
- Scalability:
    - User Base Growth: Define how the system will scale with growing user and patient volumes.
    - Data Volume: Address how the system will handle increasing amounts of patient data while maintaining performance.
- Usability:
    - User Interface Design: Create a better user-friendly interface for healthcare professionals.
    - Training Requirements: Specify the level of training needed for effective system use.
- Interoperability:
    - Data Exchange Standards: Define standards for seamless exchange of data with other healthcare systems.
    - Compatibility: Ensure compatibility with different healthcare devices and platforms.
- Maintainability:
    - Modularity: Design a modular system for easy updates and maintenance.
    - Documentation: Provide comprehensive documentation for troubleshooting and system understanding.
- Compliance:
    - Regulatory Compliance: Ensure adherence to relevant healthcare regulations and standards.

Environment

I. Software

- **Python 3.10 or higher**: Python is a versatile and largely-used programming language. Python with machine learning libraries can be used to develop predictive models for disease prediction.
- **Scikit-learn**: Python library used for traditional machine learning algorithms Numpy and Panda: Powerful data analysis libraries in Python.They can be used for efficient data analysis and manipulation.
- **React JS**: React is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and build interactive user interfaces efficiently.
- **Bootstrap**: Bootstrap is a front-end framework that simplifies the process of designing and styling web pages. It provides pre-designed components and styles that can be easily customized.
- **Django Framework**: Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django can be utilized to deploy and manage the web application, handling tasks such as routing, authentication, and database interactions.

II. Hardware

High-Performance CPU (e.g., Intel Core i7 or AMD Ryzen 9): Machine learning tasks where we train complex models, can be computationally intensive. A high-performance CPU is used for handling the processing demands of training and running predictive models. The Intel Core i7 or AMD Ryzen 9 series processors are known for their multiple cores and threads, which can significantly accelerate parallel processing tasks.

Dedicated GPU (e.g., NVIDIA GeForce RTX 3080):GPUs are famous for handling the matrix operations and numerical calculations involved in training neural networks. The NVIDIA

GeForce RTX 3080 is a powerful GPU with CUDA cores, making it suitable for fastening machine learning workloads.

At Least 16GB of RAM: Random Access Memory (RAM) is essential for storing and accessing data that is actively being used by the computer. In machine learning, having enough RAM is crucial for loading large datasets and performing computations on them.

SSD with a Minimum of 512GB: A Solid State Drive (SSD) is crucial for fast data access and retrieval. In machine learning applications, where large datasets and models are involved, having a high-capacity SSD helps in quick loading of data and models.

Modules

1. Data Collection Module

This module helps to gather all data which is required for training disease prediction models.It includes following tasks:

- Collect the data related to multiple diseases and their corresponding symptoms from Columbia University Dataset.It contains data of about 50 diseases and 132 different symptoms which are associated with them.
- Import the dataset which consists of diseases and symptoms into a relational database MYSQL.

2. Data Preparation Module

This module converts the data into appropriate format so that it could be used by machine learning model for training purposes**.** It includes following tasks:

- Preprocess the dataset which consists of diseases and symptoms.
    - Fill in the values which are missing.
- Retrieve the names of diseases and the symptoms associated with it.
- Process the names of diseases and the symptoms associated with it.
    - Exclude the disease id and include the name of disease only.
    - Extract the corresponding symptoms.
    - Get the Names of the disease.
    - Get the Symptoms with respect to the Diseases.
    - Obtain dummy values of corresponding symptoms of a given disease
- For each of the disease names, convert every one of them to a natural number.

3. Disease Prediction Module

This module predicts the disease based on symptoms seen by the patient as well as demographic information of the patient. Around 50 different diseases can be accurately predicted by making use of this model. The input Features could include patient demographics, medical history, symptoms, and possibly genetic information. The label/output is the disease status means whether a patient has a specific disease or not. Given a set of input features, this module will predict the likelihood of having a particular disease.

It includes following tasks:

- Splitting the dataset into training and testing data.
- Train the model using different Machine Learning Algorithms.
- Verify the accuracy of those models.
- Use the model with the highest accuracy for predicting the diseases.

4. Location Finder Module

Once the disease is predicted, this module will guide patients to find nearby hospitals with respect to their current locations. So that they can do the health checkup and initiate the medical treatment process if needed. In this module:

- Patient will enter the zip code or his current address. Based on that choice, we will find the longitude and latitude of that location.
- We will display the Map View for those hospitals by using Google Maps API Key.

5. Use Case Diagram

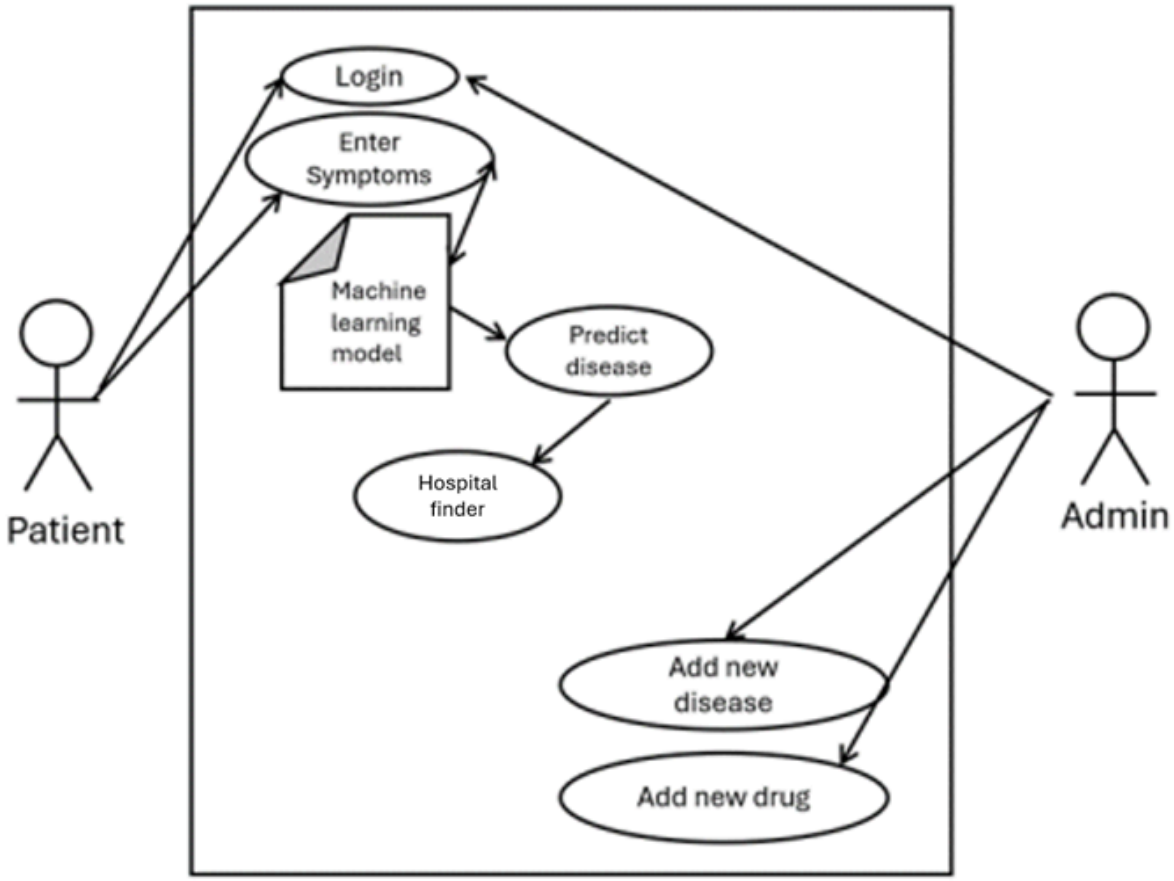


Figure No. 1

The above figure shows the use case diagram of Disease Prediction and Drug Recommendation System with two main actors, that are the Patient and the Admin. The Patient begins by logging into the system, proceeding to enter symptoms which are then processed by the Machine Learning Model. The fundamental analytical tool that makes prospective disease predictions based on given symptoms is this model. The program also displays the list of nearest hospitals based on the patient's location.

To keep the Machine Learning Model's predictions and suggestions current with the most recent medical knowledge, the Admin can improve the system's knowledge base by adding new illnesses and the symptoms to the database. The goal of this partnership between the updates

from the admin and the patient's inputs is to deliver precise and rapid medical advice and predictions.

A. a.Actor- Patient

- **Login:** Patient has to do registration and login before accessing the disease prediction system and its results. Once the patient gets logged in to the system successfully, he can view the Patient Dashboard.
- **Create Profile:** This allows a patient to create a profile where he can enter his personal details as well as other demographic information like birth date, gender, the place where he resides. Some of this information will be used later on while predicting the diseases and locating the hospitals.
- **Enter Symptoms:** This action allows users to enter different symptoms which they are seeing currently. There will be a dropdown menu while selecting these symptoms which consists of around 132 different symptoms.
- **Predict Diseases:** Once a user clicks on the predict disease button, one of the possible diseases will be shown on the screen based on symptoms he has chosen. The ML Learning model runs in the background which will generate the diagnosis results and it will display those results on the disease prediction panel.
- **Find Hospitals:** Once the disease is predicted , users can find the nearest hospitals from its current location using this functionality.

B. Actor- Admin

- **Develop and maintain the code:** Admin can work on the functionalities like do the web development , test and maintain the code.
- **Load ML Model:** Admin can load the Machine learning model to the existing code once if it is giving me better accuracy.

6. Sequence Diagram

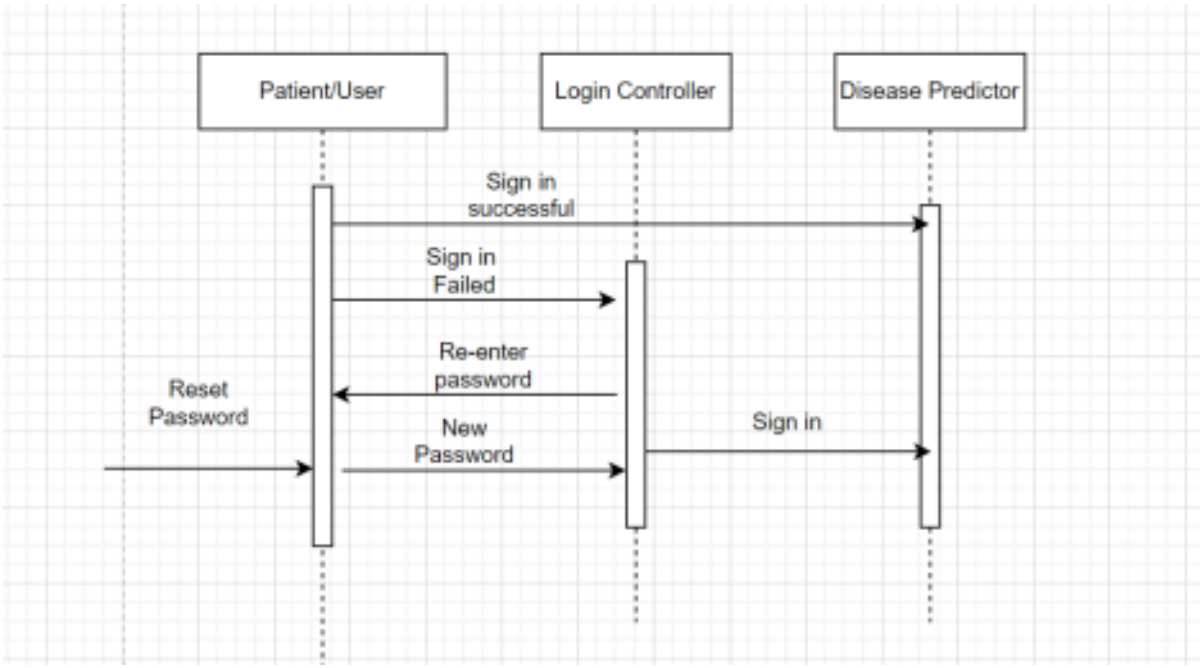


Figure No. 2
Sequence Diagram for User Login

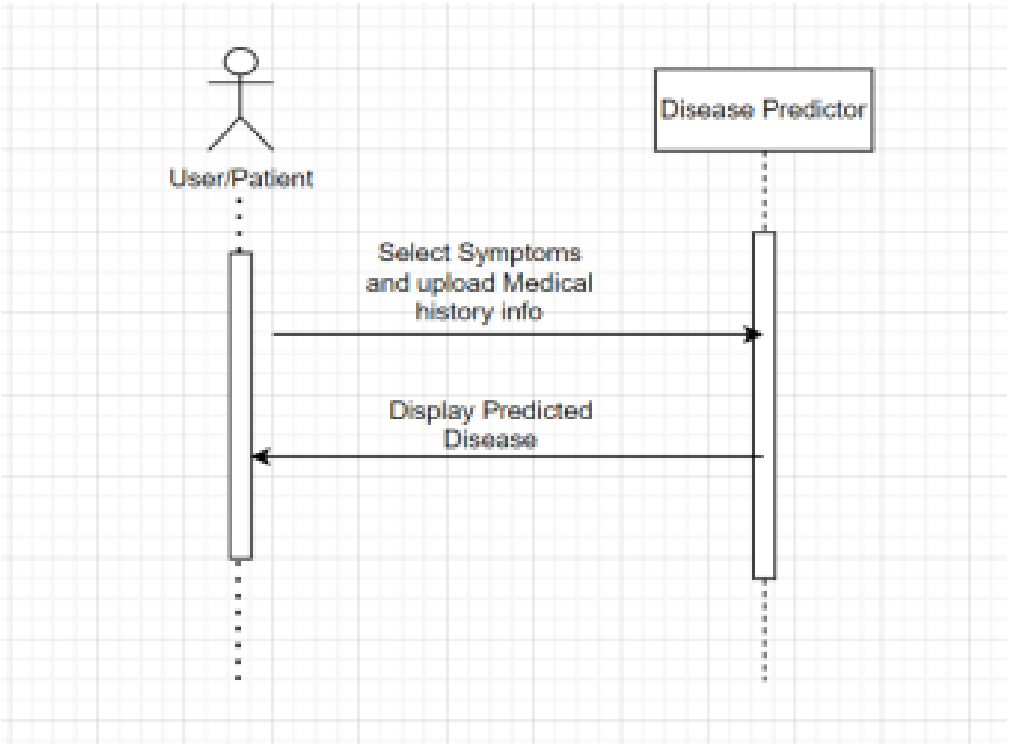


Figure No. 3
Sequence Diagram for Disease Prediction

A sequence diagram is a type of Unified Modeling Language (UML) diagram that illustrates the interactions and order of events between different components or actors in a system over a specific period of time.

Participants:

Patient: Initiates the process by requesting disease prediction and hospital finder.

System Components:

- Login Controller: The Patient initiates the process by sending a request for disease prediction to the system.Login Controller authenticates users based on their credentials and provides access to the system.Upon successful authentication, the Login Controller grants access to the system for the Patient.
- Disease Predictor: The Patient's data is then forwarded to the Disease Predictor component. Disease Predictor analyzes patient data including symptoms and predicts the likelihood of a disease.
- Hospital Finder: Once the disease is predicted, patients can use this service to locate nearest hospitals based on its current location.

7. Activity Diagram

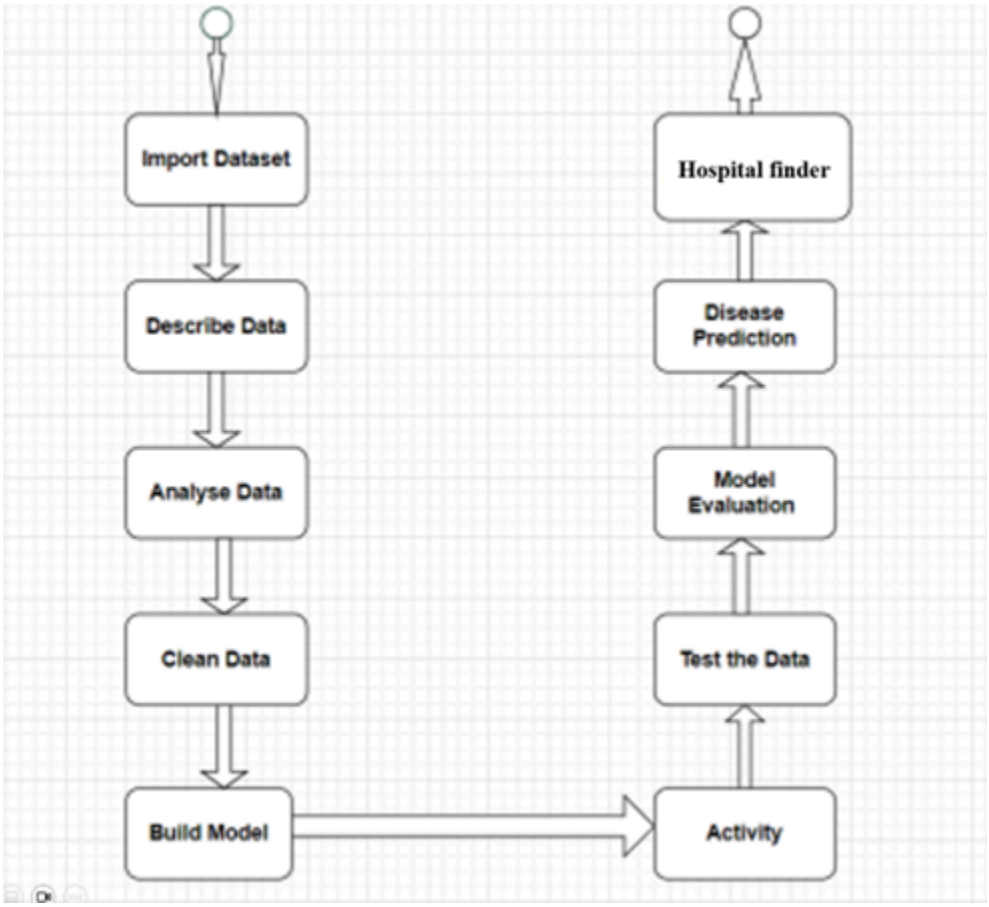


Figure No. 4

The figure given above represents an activity diagram that visualizes the flow of activities or processes within our system. This diagram illustrates the sequential flow of activities in a disease prediction and drug recommendation system, providing a high-level overview of the system's dynamic behavior.

8. System Architecture

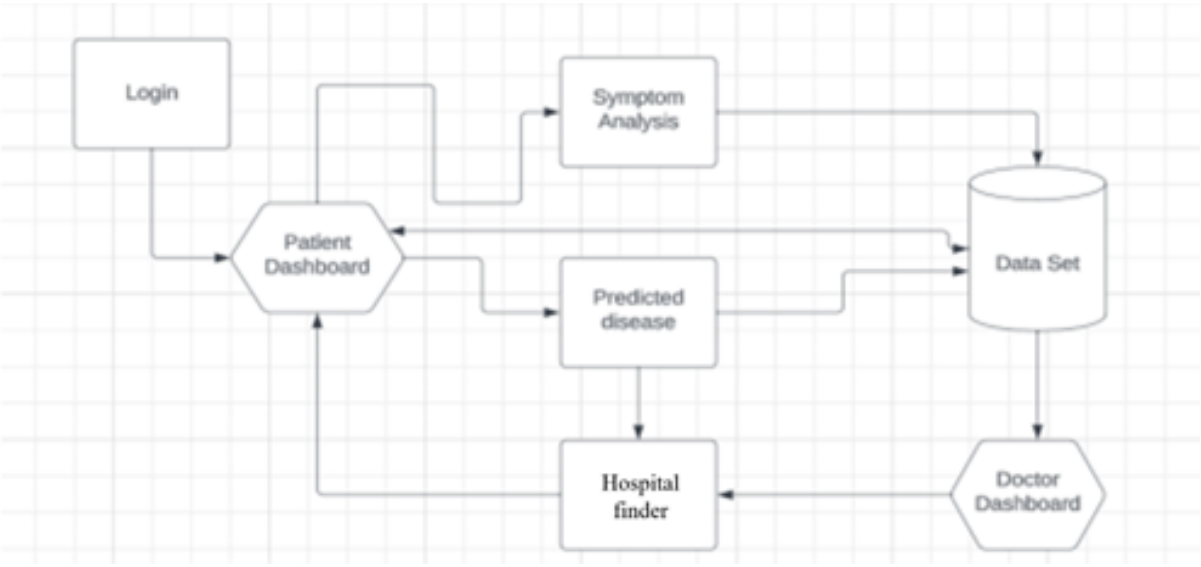


Figure No. 5

The system architecture of disease prediction and drug recommendation system is shown above. It consists of 5 main modules:

1) **Patient Dashboard:** It provides a user interface for patients where they can enter their personal details along with the symptoms that they currently see and their medical history.

3) **Symptoms Analysis:** Symptoms analysis is done by collecting the symptoms collected by patients as well as by analyzing the demographic information such as gender,age and medical history of the patient.

4) **Disease Prediction:** The output of the patient dashboard is predicted disease and this information is used later on to recommend specific drugs to the patient.

5) **Hospitals Finder Service :**The output of the patient dashboard is given to the hospital finder API which considers the current location of the patient/ zip code to locate the hospitals and displays it on Google Maps API.

Implementation

I. Backend of the disease Prediction System

Model Building using Machine Learning

Steps involved:

1. Loading of Dataset

- Import necessary python libraries for data analysis and manipulation.
- Load Symptoms dataset from training.csv file.

```
In [3]:
#IMport necessary Libraries
import pandas as pd
from pandas import read_csv
import numpy as np
import matplotlib.pyplot as plt
```

```
In [4]:
#Loading dataset
filename="Training.csv"
data=read_csv(filename)
data.head()
```

Out[4]:

blackheads	scurring	skin_peeling	silver_like_dusting	small_dents_in_nails	inflammatory_nails	blister	red_sore_around_nose	yellow_crust_ooze	prognosis
0	0	0	0	0	0	0	0	0	Fungal infection
0	0	0	0	0	0	0	0	0	Fungal infection
0	0	0	0	0	0	0	0	0	Fungal infection
0	0	0	0	0	0	0	0	0	Fungal infection
0	0	0	0	0	0	0	0	0	Fungal infection

Figure No. 6

```
In [7]: data.tail(10)
```

Out[7]:

ıeads	scurring	skin_peeling	silver_like_dusting	small_dents_in_nails	inflammatory_nails	blister	red_sore_around_nose	yellow_crust_ooze	prognosis
0	0	0	0	0	0	0	0	0	Hypothyroidism
0	0	0	0	0	0	0	0	0	Hyperthyroidism
0	0	0	0	0	0	0	0	0	Hypoglycemia
0	0	0	0	0	0	0	0	0	Osteoarthristis
0	0	0	0	0	0	0	0	0	Arthritis
0	0	0	0	0	0	0	0	0	(vertigo) Paroymsal Positional Vertigo
1	1	0	0	0	0	0	0	0	Acne
0	0	0	0	0	0	0	0	0	Urinary tract infection
0	0	1	1	1	1	0	0	0	Psoriasis
0	0	0	0	0	0	1	1	1	Impetigo

Figure No. 7

- Here starting columns will show disease symptoms whereas the last column will tell us about prognosis i.e. disease diagnosis.
    - 0-> symptom not seen
    - 1->symptom is seen

2. Feature Selection and Train Test Data Split

- In this step we select high level features i.e. symptoms from our training dataset which we will need for training our Machine Learning model for disease prediction.
- These selected features will serve as the essential inputs for training our machine learning model in the prediction of diseases. Additionally, we will proceed to partition our dataset into training and testing sets to facilitate the model training process.

```
In [8]: #Feature Selection and Train Test Data Split
        from sklearn.model_selection import train_test_split

In [9]: #feature selection
        df_x=data[['itching','skin_rash','nodal_skin_eruptions','continuous_sneezing','shivering','chills','joint_pain','stomach_pain
        ]]
        df_y=data[['prognosis']]

In [10]: df_y.head()
```

Out[10]:

	prognosis
0	Fungal infection
1	Fungal infection
2	Fungal infection
3	Fungal infection
4	Fungal infection

```
In [11]: #Train Test Split
         X_train, X_test, y_train, y_test = train_test_split(df_x, df_y, test_size=0.2, random_state=0)

In [12]: X_train.shape
```

Figure No. 8

3. Fitting the model and making confusion Matrix

- Fit the model using Training data and use gaussian naive bayes classifier for training purpose.
    - df_x- input features (disease symptoms)
    - df_y->target variable (disease diagnoses)
- Use Gaussian Naive Bayes classifier to make predictions on the test set (X_test), and the results are stored in y_pred.
- In machine learning, a confusion matrix is a table that is often used to evaluate the performance of a classification model. It provides a summary of the predictions made by a model on a set of data compared to the actual labels.

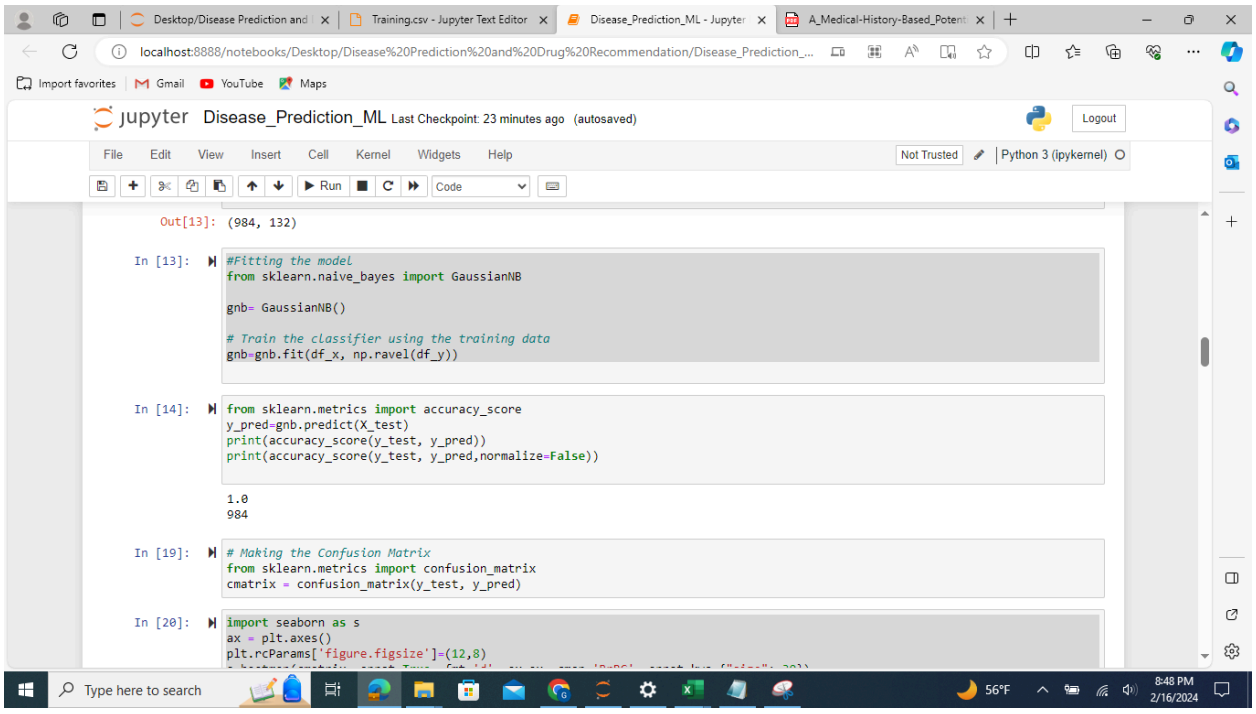


Figure No. 9

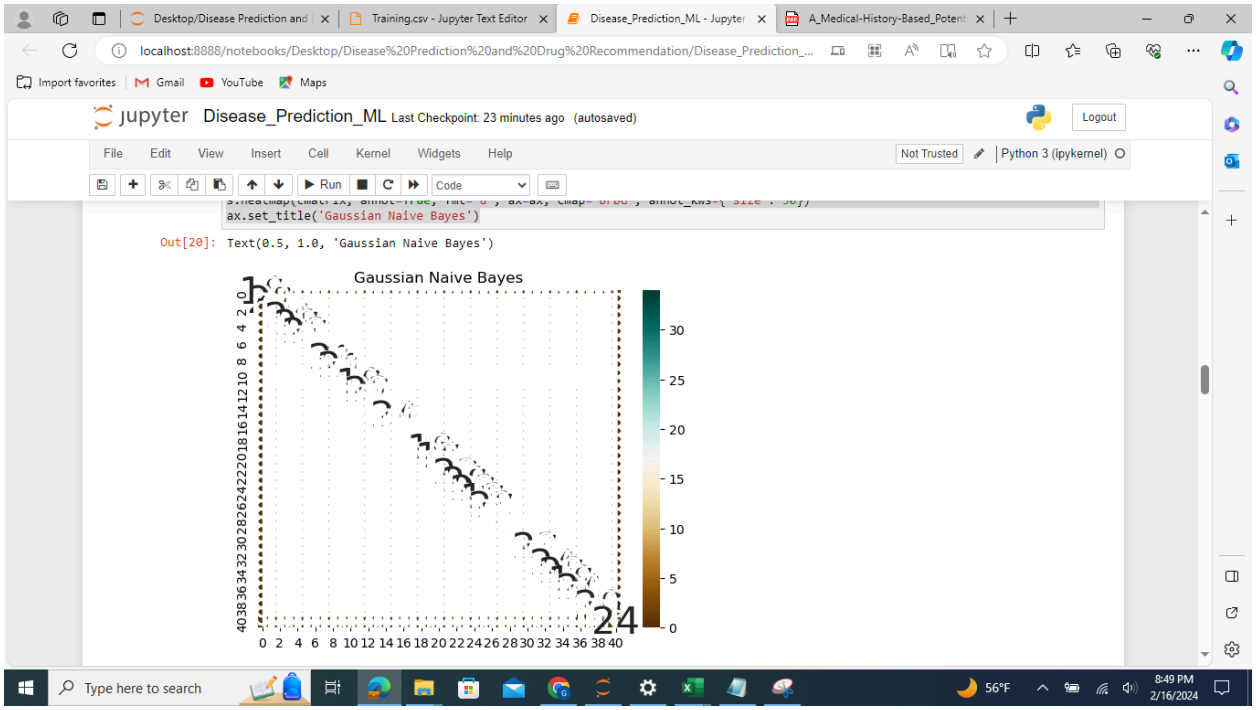


Figure No. 10

4. Save the model and load it

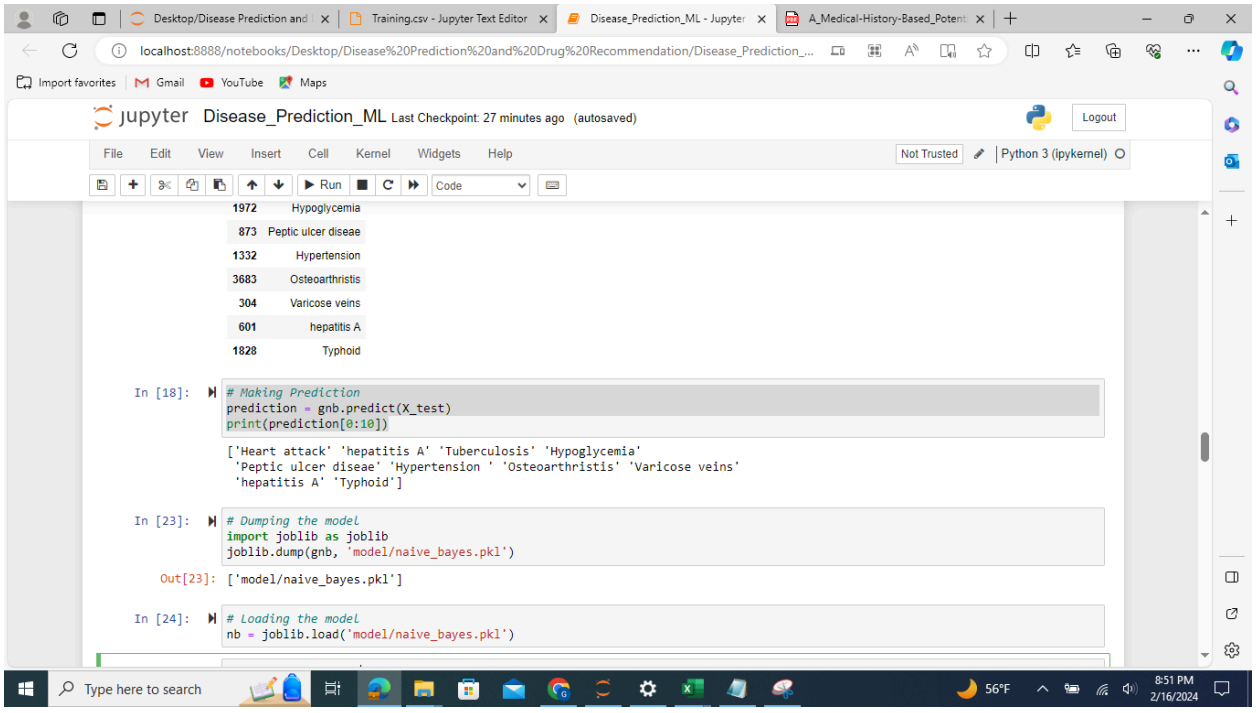


Figure No. 11

5. Predict Disease with Guassian Naive Bayes Classifier ML model

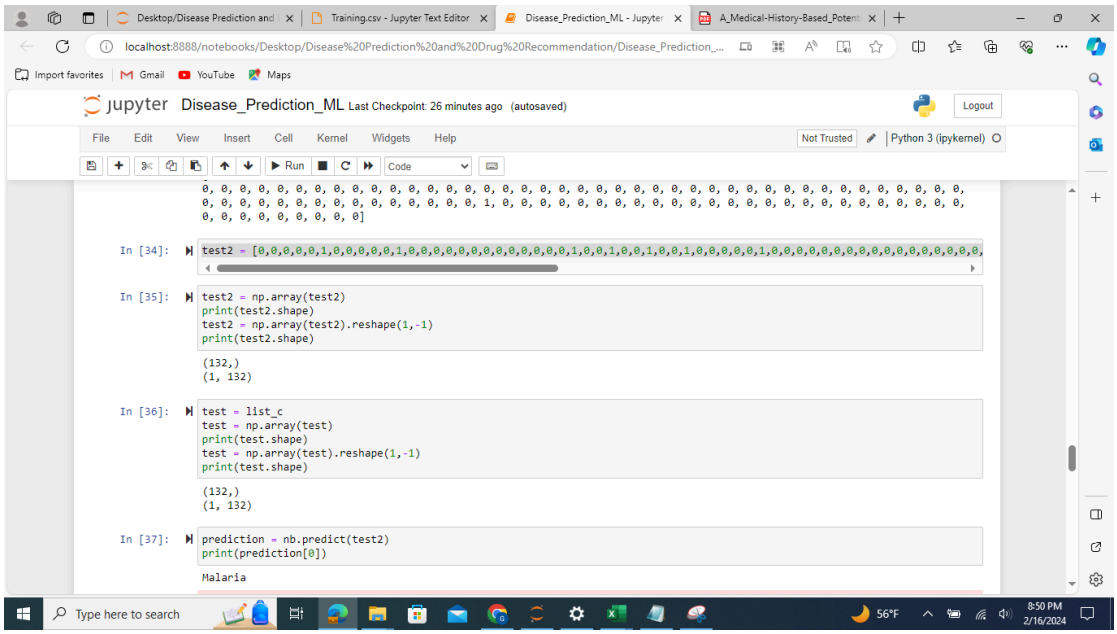


Figure No. 12

6. Predict Disease with Random Forest ML Model:

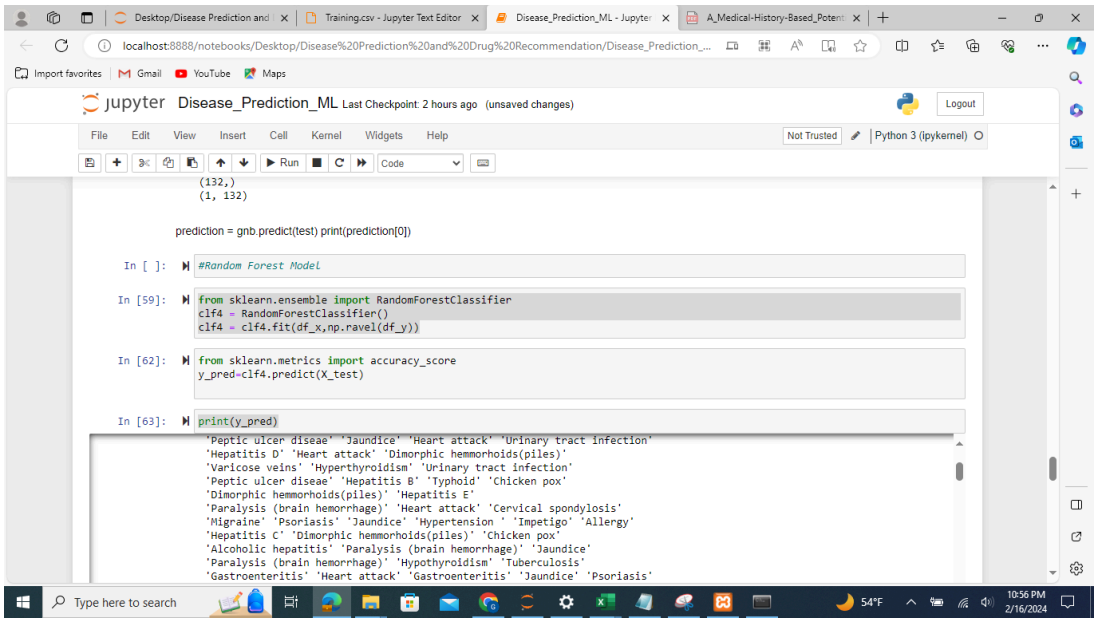


Figure No. 13

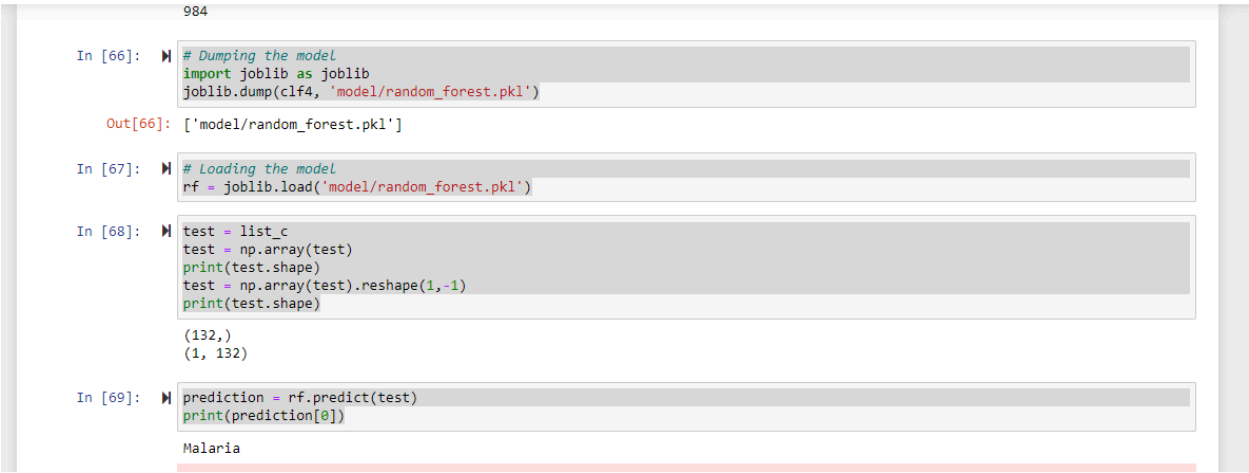


Figure No. 14

7. Predict Disease Logistic Regression Model:

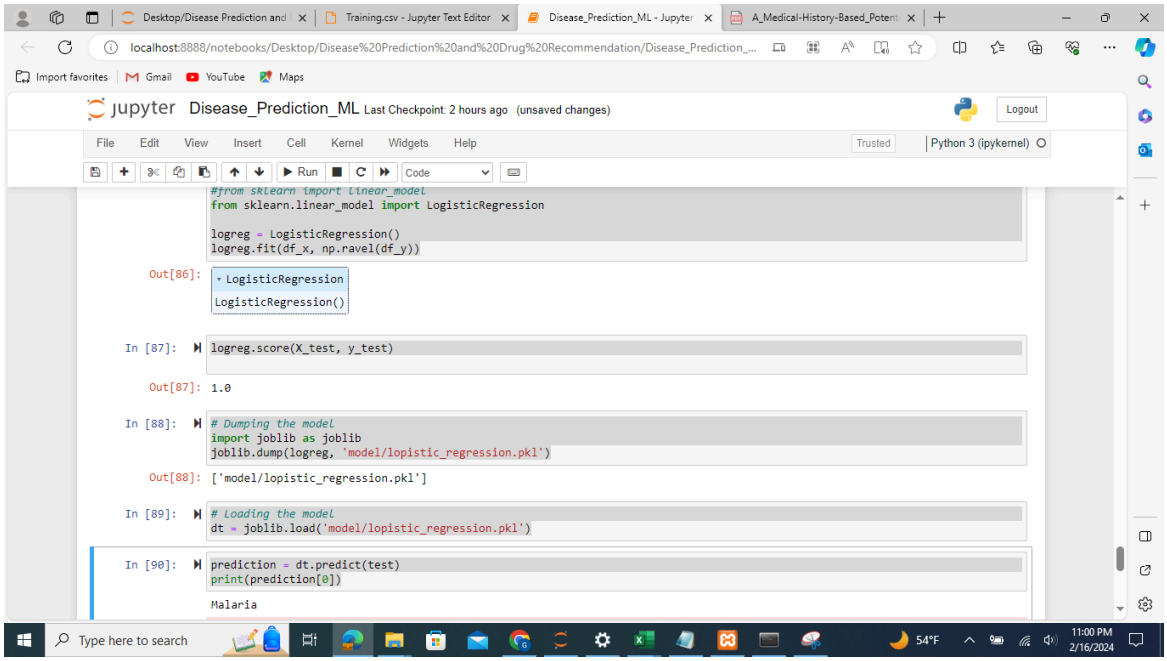


Figure No. 15

8. Predict Disease with Decision Tree ML model

```
In [101]:
from sklearn import tree

clf3 = tree.DecisionTreeClassifier()   # empty model of the decision tree
clf3 = clf3.fit(df_x,df_y)

In [102]:
from sklearn.metrics import accuracy_score
y_pred=clf3.predict(X_test)
print(accuracy_score(y_test, y_pred))
print(accuracy_score(y_test, y_pred,normalize=False))

1.0
984

In [103]:
# Dumping the model
import joblib as joblib
joblib.dump(clf3, 'model/decision_tree.pkl')

Out[103]: ['model/decision_tree.pkl']

In [104]:
# Loading the model
dt = joblib.load('model/decision_tree.pkl')

In [105]:
prediction = dt.predict(test)
print(prediction[0])

Malaria
```

Figure No. 16

We are developing our code using the Django web application framework in Python which is used for rapid web development. The Project structure is as follows:

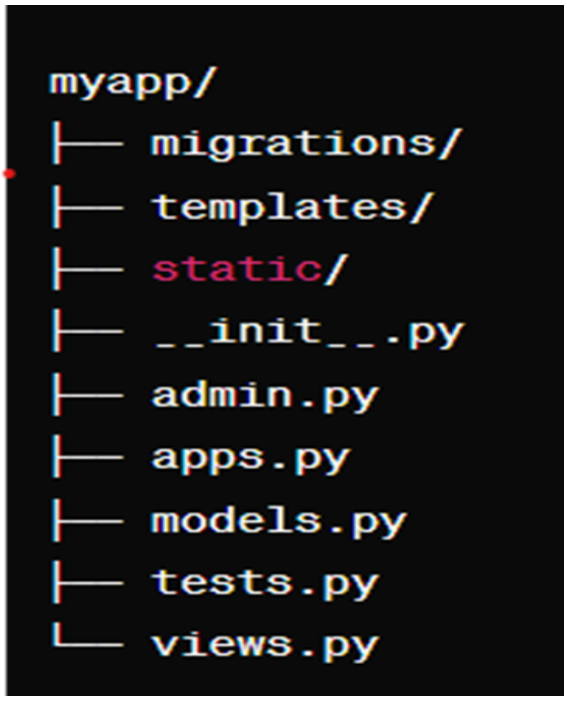


Figure No. 17

models.py: Defines database models for the app.

views.py: Contains view functions or classes to handle HTTP requests.

urls.py: Defines URL patterns specific to the app.

templates/: Directory for HTML templates specific to the app.

static/: Directory for static files (e.g., CSS, JavaScript) specific to the app.

migrations/: Directory for database migrations generated by Django's ORM.

II. Front End

Front End : Home Page

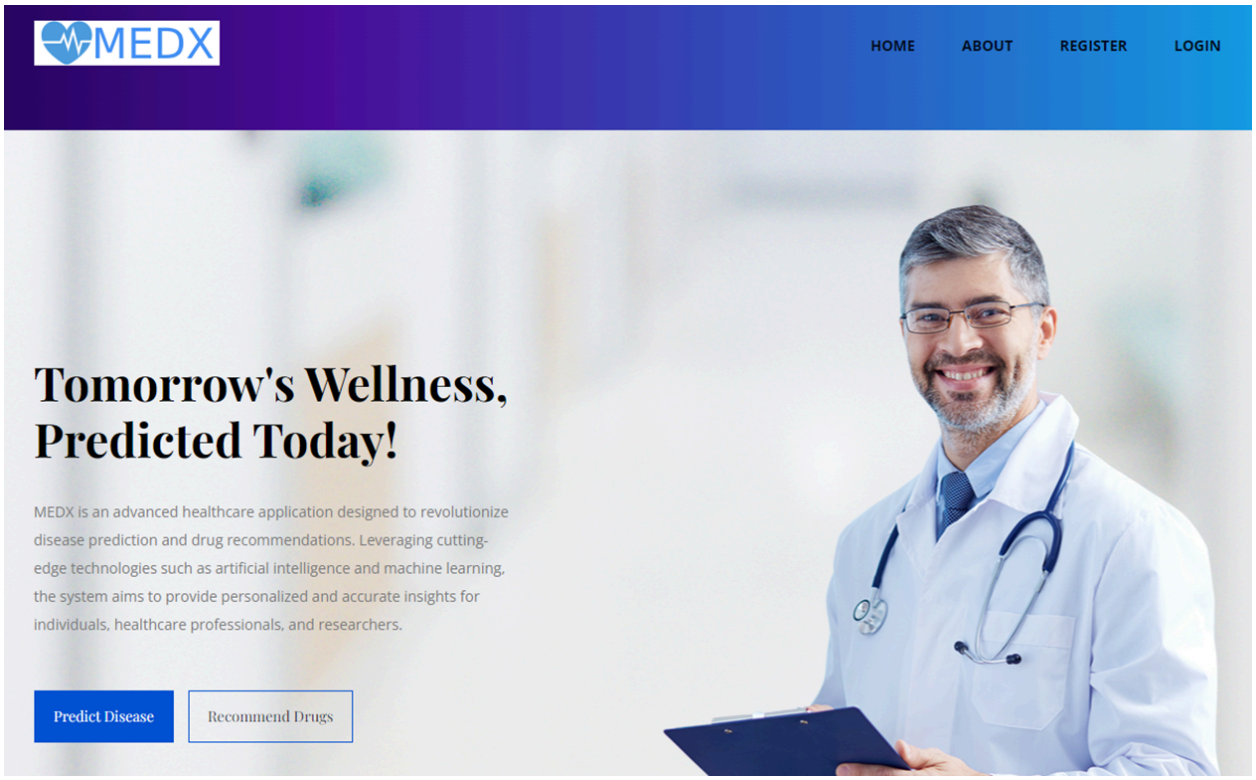
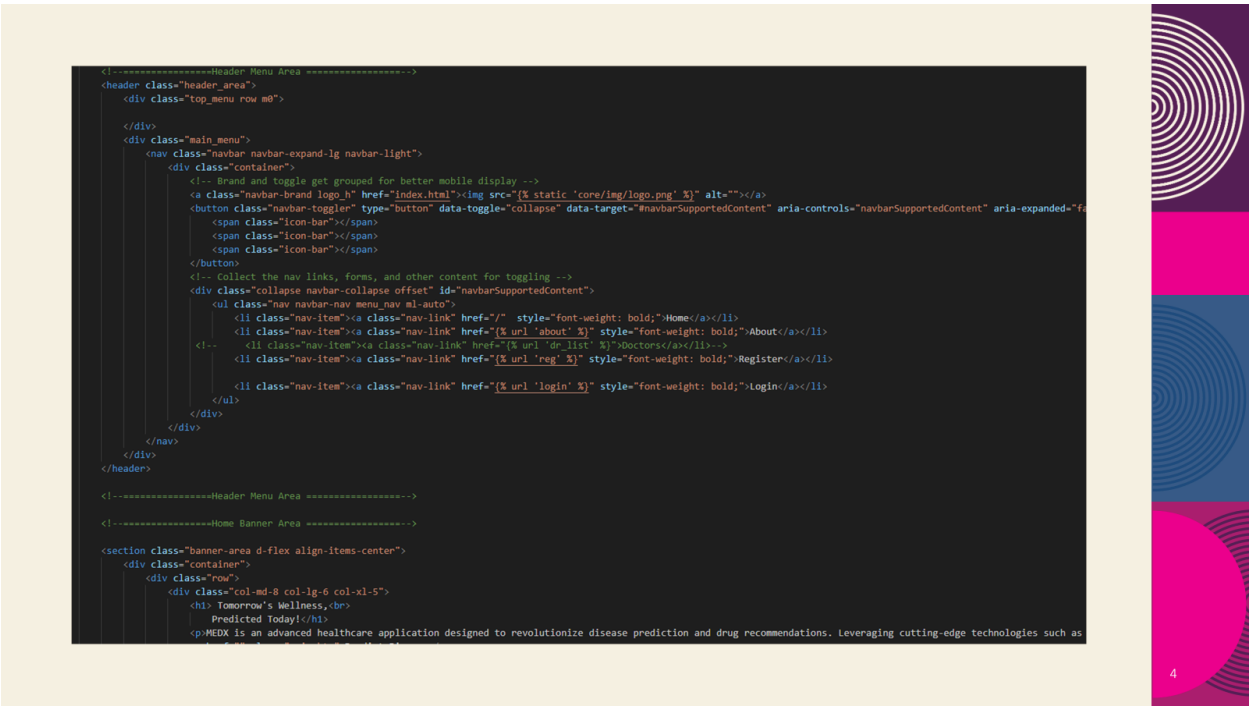


Figure No. 18



```
<!--================Header Menu Area =================-->
<header class="header_area">
    <div class="top_menu row m0">

    </div>
    <div class="main_menu">
        <nav class="navbar navbar-expand-lg navbar-light">
            <div class="container">
                <!-- Brand and toggle get grouped for better mobile display -->
                <a class="navbar-brand logo_h" href="index.html"><img src="{% static 'core/img/logo.png' %}" alt=""></a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="fa
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <!-- Collect the nav links, forms, and other content for toggling -->
                <div class="collapse navbar-collapse offset" id="navbarSupportedContent">
                    <ul class="nav navbar-nav menu_nav ml-auto">
                        <li class="nav-item"><a class="nav-link" href="/"  style="font-weight: bold;">Home</a></li>
                        <li class="nav-item"><a class="nav-link" href="{% url 'about' %}" style="font-weight: bold;">About</a></li>
                    <!--    <li class="nav-item"><a class="nav-link" href="{% url 'dr_list' %}">Doctors</a></li>-->
                        <li class="nav-item"><a class="nav-link" href="{% url 'reg' %}" style="font-weight: bold;">Register</a></li>

                        <li class="nav-item"><a class="nav-link" href="{% url 'login' %}" style="font-weight: bold;">Login</a></li>
                    </ul>
                </div>
            </div>
        </nav>
    </div>
</header>

<!--================Header Menu Area =================-->

<!--================Home Banner Area =================-->

<section class="banner-area d-flex align-items-center">
    <div class="container">
        <div class="row">
            <div class="col-md-8 col-lg-6 col-xl-5">
                <h1> Tomorrow's Wellness,<br>
                    Predicted Today!</h1>
                <p>MEDX is an advanced healthcare application designed to revolutionize disease prediction and drug recommendations. Leveraging cutting-edge technologies such as
```


Figure No. 19

Front End : Registration Page

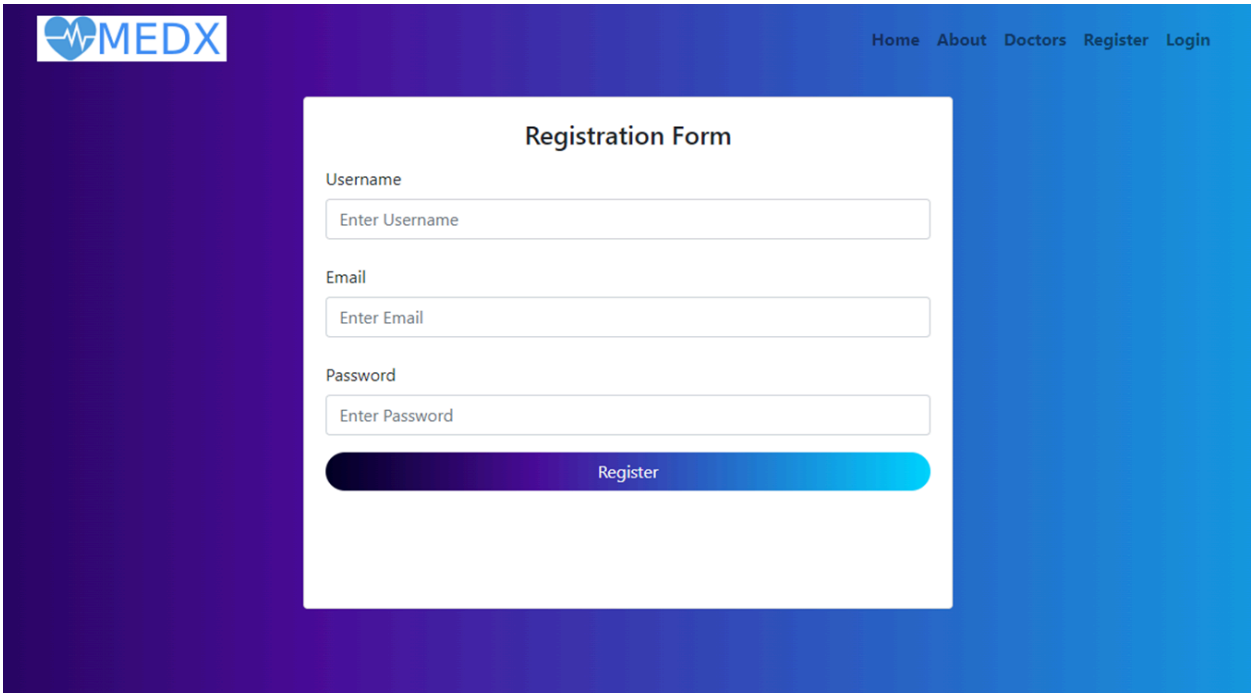


Figure No. 20

```
<div class="container-fluid">

<div class="row">


    <div class="col-sm-4">


    </div>


    <div class="col-sm-4" style="padding-top: 20px">

        <div class="card body" style="padding-bottom: 70px; padding-left: 20px; padding-right: 20px; padding-top: 20px" id="login_card">
        <h4 class="text-center">Registration Form</h4>


        <div class="reg_form">
            <form method="POST" action="{% url 'reg_user' %}">
                {% csrf_token %}
              <div class="form-group">

        <label class="col-form-label">Username</label>

        <div class="row">
                <div class="col"><input type="text" class="form-control" name="username" placeholder="Enter Username" value="" required=""></div>

        </div>
 </div>

 <div class="form-group">
    <label class="col-form-label">Email </label>
    <div class="row">
     <div class="col">
         <input type="email" class="form-control" name="email" placeholder="Enter Email" required="" value="">
     </div>
     </div>
 </div>
```

Figure No. 21

Front End : Login Page

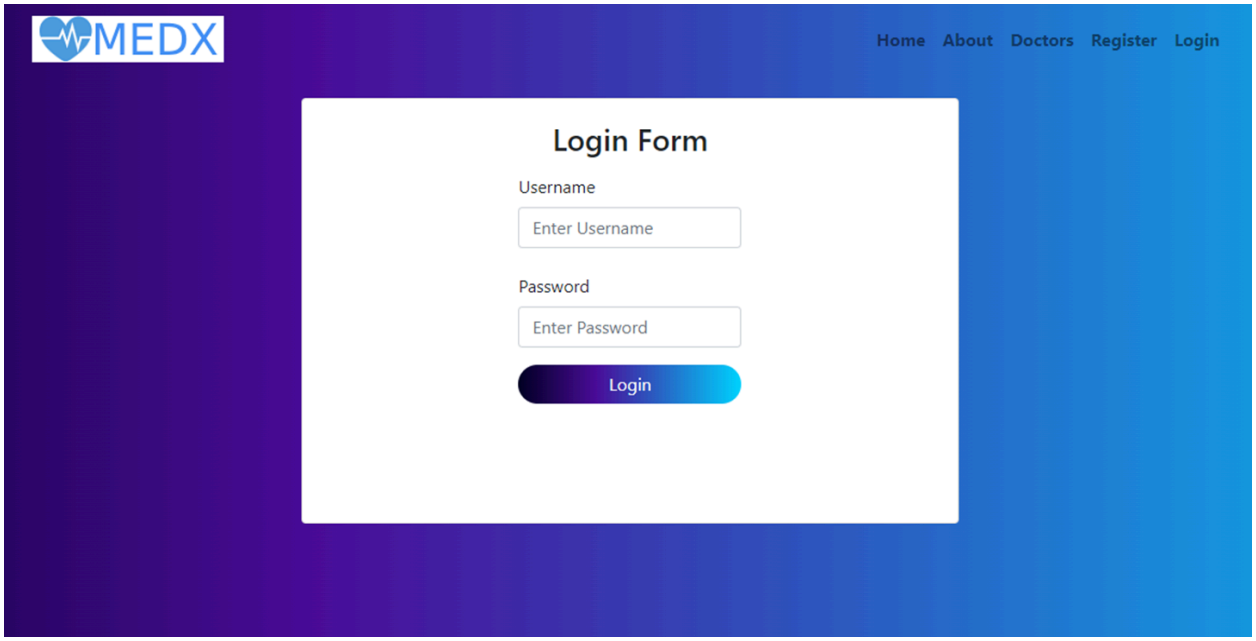


Figure No. 22

```
<div class="container-fluid">

<div class="row">


    <div class="col-sm-4">


    </div>


    <div class="col-sm-4" style="padding-top: 20px">

        <div class="card body" style="padding-bottom: 70px; padding-left: 20px; padding-right: 20px; padding-top: 20px" id="login_card">


        <form class="mx-auto" method="POST" action="">
        {% csrf_token %}

         <h3 class="text-center">Login Form</h3>
        <div class="login_form">
             <div class="form-group">
        <label class="col-form-label">Username</label>

        <div class="row">
             <div class="col"><input type="text" class="form-control" id="org" placeholder="Enter Username" required="" value="" name="username"></div>

        </div>
 </div>



        <div class="form-group">
          <label class="col-form-label">Password </label>
          <div class="row">
           <div class="col">
              <input type="password" class="form-control" id="org" placeholder="Enter Password" required="" value="" name="password">
           </div>
           </div>
        </div>

 <button type="submit" id="login_btn" class="btn btn-primary mt5" style>Login</button>
```

Figure No. 23

Front End : Patient Dashboard

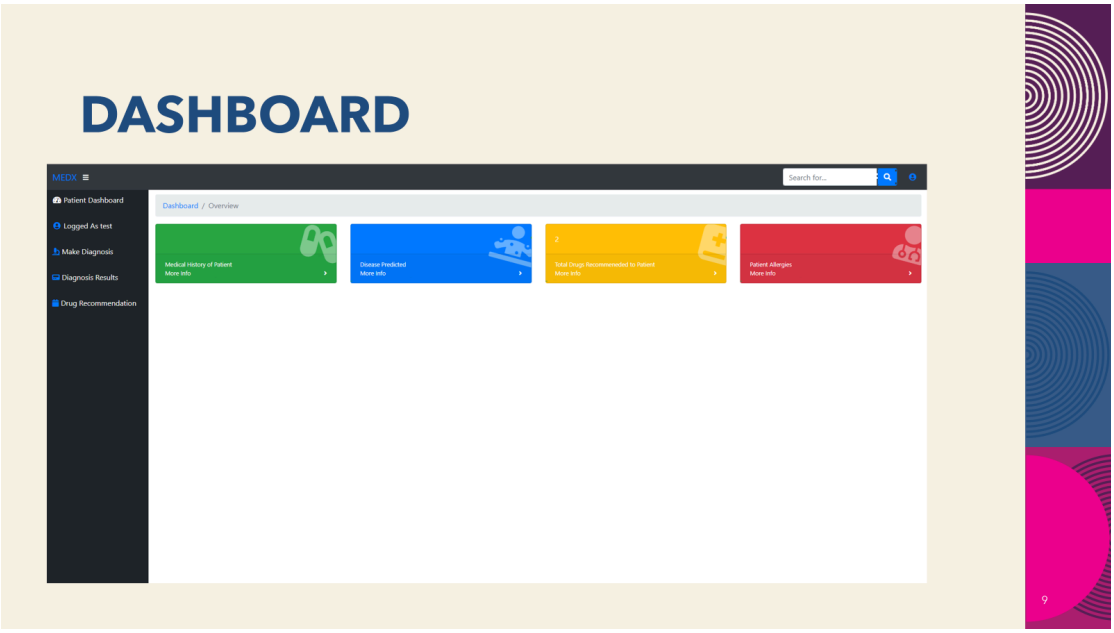


Figure No. 24

Front End : Disease Prediction Panel

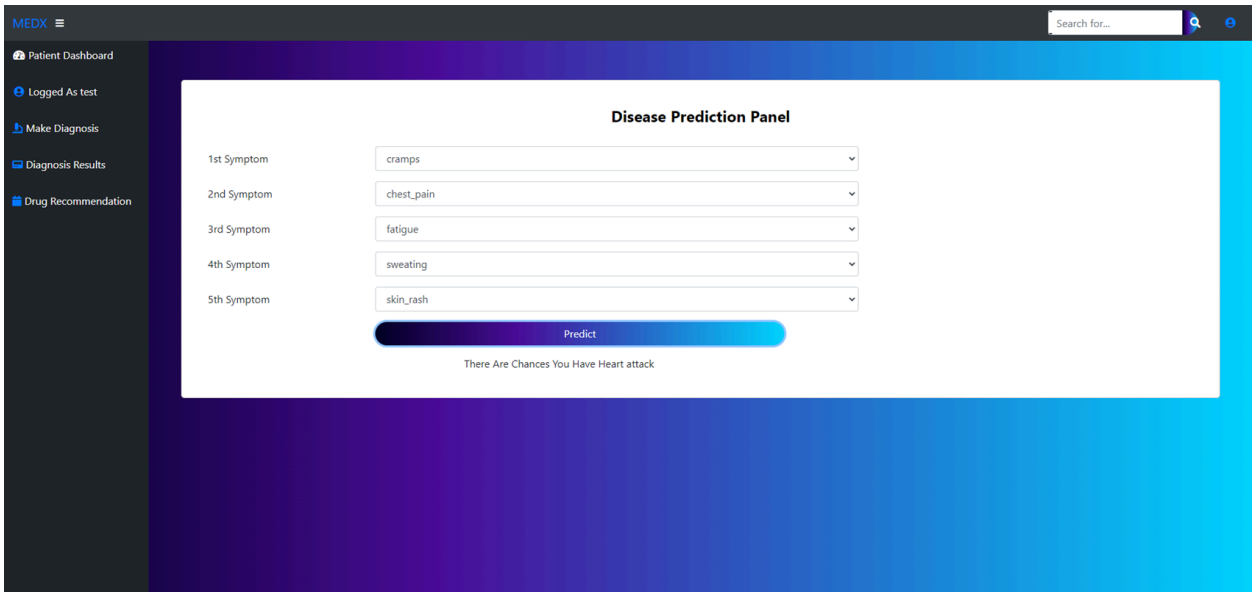


Figure No. 25

Conclusion

In conclusion, the ever-increasing burden on healthcare systems has necessitated a shift towards data-driven multi disease prediction systems. This project has explored the potential of leveraging machine learning algorithms to predict diseases based on patient symptoms, with the aim of facilitating early disease prediction and enhancing the efficiency and cost-effectiveness of healthcare.

Additionally, through this project we are predicting around 49 different diseases with the help of a total 132 symptoms. In an era dominated by Machine Learning (ML), these systems offer precise and reliable clinical predictions while conserving valuable healthcare resources. They provide patients and healthcare professionals with tailored predictions, taking into account individual symptoms, vital signs, and health parameters. Such systems are invaluable during medical emergencies, offering swift and safe medical diagnosis , thereby enhancing patient care while maintaining data privacy and integrity.

In conclusion, the fusion of machine learning and healthcare holds the promise of transforming disease prediction, ultimately leading to more efficient, cost-effective, and patient-centric healthcare systems.

Future Work

This project primarily concentrates on the creation of machine learning models designed to predict diseases and suggest suitable medications. Although this project offers valuable insights and solutions, the technology in this domain has continuously progressed, providing more encompassing solutions. With the continued growth of electronic health records (EHRs) and healthcare data repositories, researchers may have access to even larger and more diverse datasets for training and testing machine learning models. So they can utilize larger datasets to ensure machine learning model's accuracy and prediction results.

Future research work can be focused on making machine learning models for disease prediction more interpretable and explainable, especially in the healthcare domain where trust and transparency are crucial. Collaborative AI systems that involve both machine and human intelligence to improve disease diagnosis and treatment planning may have been developed. The ethical implications of using AI in healthcare, such as bias and fairness, may have been addressed more comprehensively.

Researchers may have increasingly utilized deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to improve disease prediction accuracy. Deep learning models are known for their ability to automatically extract relevant features from raw data.

Installation Instructions

1. **Clone the project repository using this link:**
   https://github.com/gouri-sabale-123/Disease_Prediction_And_Drug_Recommendation_Using_Machine_Learning.git

2. **Setup XAMPP Server on localhost by following the XAMPP installation guidelines:**
   https://www.apachefriends.org/download.html

3. **Execute healthcare_db.sql script present in project folder into MYSQL DB**
   It is available at http://localhost/phpmyadmin/

4. **Once the script is executed successfully, start Apache and MYSQL from XAMPP**

5. **Install Anaconda Python using following link:**
   https://docs.anaconda.com/free/anaconda/install/index.html

6. **Start anaconda command prompt and run it as administrator**

7. **Navigate to the project folder cloned on your local system where you will see manage.py**

8. **Run following command in anaconda command prompt**
   python runserver manage.py

Glossary of disease Prediction Terms

1. **Electronic Health Records(ETH):** These are digital versions of a patient's paper information which contain all the information about a patient's medical history , symptoms, diagnoses and other demographic information.
2. **Naive Bayes Classifier:** It is a probabilistic ML algorithm used for performing classification tasks in Machine Learning.
3. **Decision Tree Algorithm:** Supervised Machine Learning algorithms used for classification and regression tasks.
4. **Random Forest Classifier:** It builds multiple decision trees during training and outputs mode of all the classes.

References

[1] Gomathi, R. M., K, D. J., Ajitha, P., Sivasangari, A., Anandhi, T., & Rani, V. N. (2022). Flawless multi perspective vision for prediction of disease using machine learning approach. *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*. https://doi.org/10.1109/icoei53556.2022.9776787

[2] Gupta, A., & Gupta, M. K. (2022). Prediction of diseases using different machine learning approaches. *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*. https://doi.org/10.1109/iciem54221.2022.9853132

[3] Rasheed, S. S., & Glob, I. H. (2022). Classifying and prediction for patient disease using machine learning algorithms. *2022 3rd Information Technology To Enhance E-Learning and Other Application (IT-ELA)*. https://doi.org/10.1109/it-ela57378.2022.10107935

[4] Silpa, C., Sravani, B., Vinay, D., Mounika, C., & Poorvitha, K. (2023). Drug recommendation system in medical emergencies using machine learning. *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*. https://doi.org/10.1109/icidca56705.2023.10099607

[5] Nayak, S. K., Garanayak, M., Swain, S. K., Panda, S. K., & Godavarthi, D. (2023). An intelligent disease prediction and drug recommendation prototype by using multiple approaches of machine learning algorithms. *IEEE Access*, *11*, 99304–99318. https://doi.org/10.1109/access.2023.3314332

[6] https://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html