**“DISEASE PREDICTION SYSTEM USING ML ”**

**A Major Project Report Submitted in the Partial Fulfillment of Requirement for the Award of Degree of**

**Bachelor of Technology**

**In**

**Computer science and Engineering**

**Submitted To**

****

**RAJIV GANDHI PROUDYOGIKI**

**VISHWAVIDYALAYA BHOPAL (M.P.)**

**Submitted By**

**Rohan Ghosh(ENROLL. NO. 0546CS181033)**

**Prasanjeet Paul(ENROLL. NO. 0546CS181029)**

**Vivek Sarkar(ENROLL. NO. 0546CS181053)**

**Submitted To**

**Mrs.Rajni Mishra**

**(Asst.Prof. VITS BHOPAL)**

**Under the supervision of Prof.SUMIT SHARMA**

**(HOD CSE Dept.)**

**Dept. – CSE, VITS Bhopal**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Vaishnavi Institute of Technology**

**& Science BHOPAL**

**June – 2022**

**VAISHNAVI INSTITUTE OF TECHNOLOGY & SCIENCE BHOPAL**

****

**CERTIFICATE**

****

This is to certify that the dissertation entitled **“DISEASE PREDICTION SYSTEM USING ML .**” submitted to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) by Rohan Ghosh(Enroll no:0546CS181033),Prasanjeet Paul(Enroll no:0546CS181029), Vivek Sarkar(Enroll no:0546CS181053), is a partial fulfillment of the requirement for the award of degree of **Bachelor of Technology**

in computer science and Engineering. The matter embodied is the actual work done by Rohan Ghosh,Prasanjeet Paul, vivek Sarkar(Enroll no: 0546CS181033, 0546CS181029, 0546CS181053) and is a record of bonafide work done by him under my supervision.

|  |  |
| --- | --- |
| **Dr. R. S. Sikarwar** | **Prof. Sumit Sharma** |
| (Director) | **(**Supervisor & HOD , CSE) |
| VITS, Bhopal | VITS, Bhopal |

**ACKNOWLEDGEMENTS**

Any assignment puts to litmus, test of individual’s knowledge, credibility or experience and thus sole efforts of an individual are not sufficient to accomplish the desired task. Words shall never be able to describe neither the spirit with which we worked together nor shall they ever be able to express the feeling we felt towards our guides. Successful completion of a project involves interests and efforts of many people so it becomes obligatory on our part to extend our thanks to them.

I take this opportunity to thanks **Mrs**.**Rajni Mishra (Asst.Prof. VITS Bhopal)** Supervisor and H.O.D. **Prof. Sumit Sharma VITS, Bhopal** for accepting me to work under his valuable guidance, closely supervising this work over the past few months and offering many innovative ideas and helpful suggestions as and when required. His valuable advice and support, was an inspiration and driving force for me. He has constantly enriched my raw ideas with his experience and knowledge. Indeed, it was a matter of great felicity to have worked under his aegis.

I would like to give thanks to **Dr. R. S. Sikarwar**, Director and Co-Supervisor **VITS, Bhopal** for his valuable guidance and motivation.

I also wish to thanks my respected teachers of **Computer Science, VITS**, for their constant support and guidance during my research work. I extend my gratitude to all teachers of department of Computer Science and colleagues, who have always been by my side, through thick and thin during these years and helped me in several ways.

Last but not least I’d like to thank “**The Almighty God”**, my Parents, my family and my friends who directly or indirectly helped me in this endeavor.

**NAME -ROHANGHOSH ENROLL- 0546CS181033**

**PRASANJEET PAUL**

**ENROLL-0546CS181029**

**VIVEK SARKAR**

**ENROLL-0546CS181053**

**INDEX**

# Chapter 1 Introduction

* Overview
* Problem Statement
* Objective
* Applications

**Chapter 2 Requirement Analysis**

* Gantt Charts

# Chapter 3 Designing & Methodology

* Overview of methodology o COLLECTION OF DATASET o SELECTION OF ATTRIBUTES o PRE-PROCESSING OF DATA o PREDICTION OF DISEASE
* Machine Learning o Supervised Learning o Unsupervised learning o Reinforcement learning
* Algorithm o SUPPORT VECTOR MACHINE (SVM)
* Architecture Diagram
* Diagrams (Flow Charts: Use Case, Class Diagram, Sequence Diagram, DFD with explanation
* E-R Diagrams
* Coding

# Chapter 4 Implementation and Results

* External Interface Requirements o User Interfaces o Software Interfaces
* Screen Shots with Description
* Advantages of project

# Chapter 5 Conclusion & Future Scope

Conclusion

Future Scope

# References User Manual

**LIST OF FIGURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **S**  **No** | **Figure No** | **Figure Name** | **Page No** |
| **1** | Figure 1 | Collection of data | **9** |
| **2** | Figure 2 | Preprocess of data | **10** |
| **3** | Figure 3 | Prediction of Disease | **11** |
| **4** | Figure 4 | Support Vector Algorithm | **14** |
| **5** | Figure 5 | Architecture Diagram | **15** |
| **6** | Figure 6 | E-R Diagram | **17** |

**CHAPTER 1**

# INTRODUCTION

**1.1 OVERVIEW-**

Our project is based on disease prediction according to the symptoms shown by the patient. This model which we have built comes under the umbrella of data analysis. For this we are using python as a platform to run our machine learning algorithms. The first step to any analysis is to decide the problem we want to solve. Then getting the dataset to work on .Then we visualize our data with the help of scatter plot or any different plot and see it on an excel file by doing this we can the redundancy in our data i.e. outliers, missing values etc. Then we treat our data by replacing the missing values, as python is a case sensitive programing language we transform all the letters into capital. Creating dummy variables to sort our data into mutually exclusive categories also the no of dummy variables should be less than the no of categories of a qualitative variable. Also many people do the mistake of replacing the missing values with mean of that variable but by doing so you can miss very important variations in the data.

**1.2 PROBLEM STATEMENT**

Healthcare industry has become big business. The healthcare industry produces large amounts of health-care data daily that can be used to extract information for predicting disease that can happen to a patient in future while using the treatment history and health data. This hidden information in the healthcare data will be later used for affective decision making for patient’s health. Also, this area need improvement by using the informative data in healthcare. Major challenge is how to extract the information from these data because the amount is very large so some data mining and machine learning techniques can be used. Also, the expected outcome and scope of this project is that if disease can be predicted than early treatment can be given to the patients which can reduce the risk of life and save life of patients and cost to get treatment of diseases can be reduced up to some extent by early recognition.

For this problem, a probabilistic modeling and deep learning approach will train a Long Short-Term Memory recurrent neural network and two convolutional neural networks for prediction of disease. The rapid adoption of electronic health records has created a wealth of new data about patients, which is a goldmine for improving the understanding of human health. The above method is used to predict diseases using patient treatment history and health data.

**1.3 OBJECTIVE-**

We want to predict the disease suffered by a patient depending upon the symptoms.

**1.4 APPLICATON-**

The application of machine learning in the field of medical diagnosis is increasing gradually. This can be contributed primarily to the improvement in the classification and recognition systems used in disease diagnosis which is able to provide data that aids medical experts in early detection of fatal diseases and therefore, increase the survival rate of patients significantly. In this paper, we apply different classification algorithms, each with its own advantage on three separate databases of disease (Heart, Breast cancer, Diabetes) available in UCI repository for disease prediction. The feature selection for each dataset was accomplished by backward modeling using the p-value test. The results of the study strengthen the idea of the application of machine learning in early detection of diseases.

**CHAPTER 2**

**REQUIRMENT AND ANALYSIS**

# 2.1 Gantt Charts

**Weeks 1 2 3 4 5 6 7 8**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

**Planning**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

**Design**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

**Coding**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

**Testing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

**Delivery**

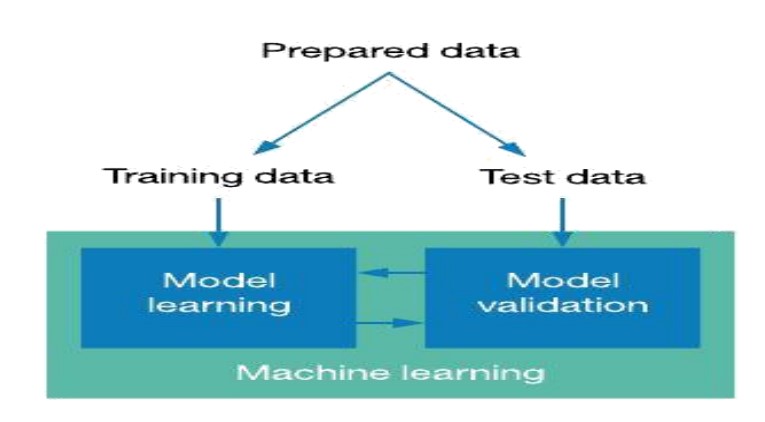
**CHAPTER 3**

# DESIGNING AND METHODOLOGY

**3.1 OVERVIEW OF METHODOLOGY**

## 3.1.1 COLLECTION OF DATASET

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing. The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.



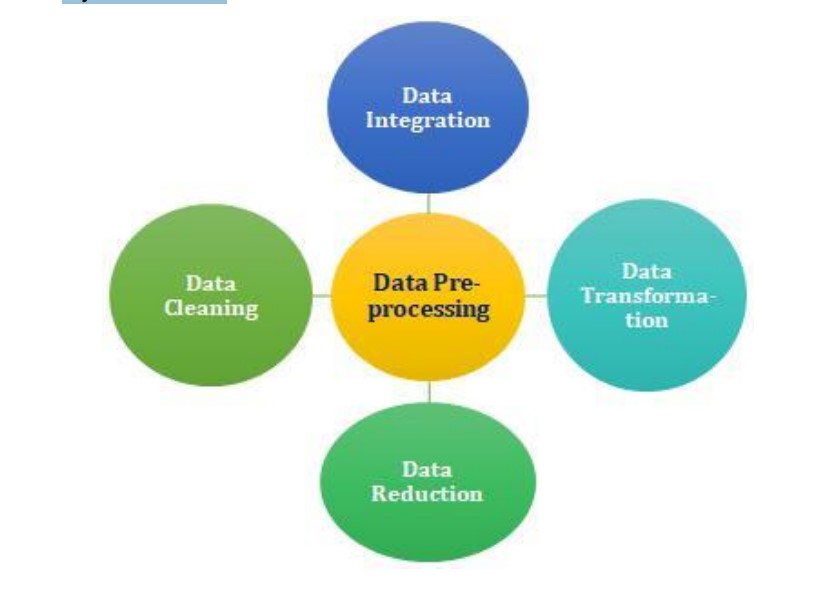
**Fig 1-Collection of data**

## 3.1.2 SELECTION OF ATTRIBUTES

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exange, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

## 3.1.3 PRE-PROCESSING OF DATA

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In preprocessing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.



## Fig2-Preprocess of data

### 3.1.4 PREDICTION OF DISEASE

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Tree, Logistic Regression, Ada-boost, Xgboost are there but we used SVM for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for disease prediction.

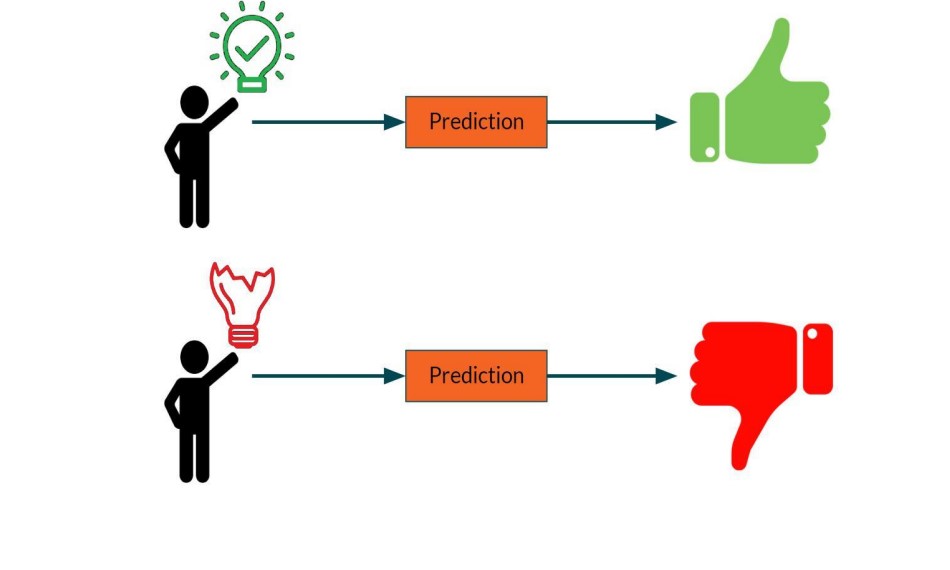


Fig 3-Prediction of Disease

**3.2 MACHINE LEARNING** In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

### 3.2.1 Supervised Learning

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output. In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher. Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

**3.2.2** **Unsupervised learning** :

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

* Unsupervised learning is helpful for finding useful insights from the data.
* Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.
* Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
* In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

### 3.2.3 Reinforcement learning –

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is 11 trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

### 3.3 ALGORITHMS

**SUPPORT VECTOR MACHINE (SVM):** 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 the algorithm is termed as Support Vector Machine. Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables. The followings are important concepts in SVM - Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points. Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes. Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the 12 support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin. Types of SVM: SVM can be of two types:

* Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier. The objective of the support vector machine algorithm is to find a hyperplane in an Ndimensional space (N - the number of features) that distinctly classifies the data points. The advantages of support vector machines are:
* Effective in high dimensional spaces.
* Still effective in cases where the number of dimensions is greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels. The disadvantages of support vector machines include:
* If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
* **ALGORITHMS**

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is.

The Linear Support Vector Classifier (SVC) method applies a linear kernel function to perform classification and it performs well with a large number of samples. If we compare it with the SVC model, the Linear SVC has additional parameters such as penalty normalization which applies 'L1' or 'L2' and loss function.

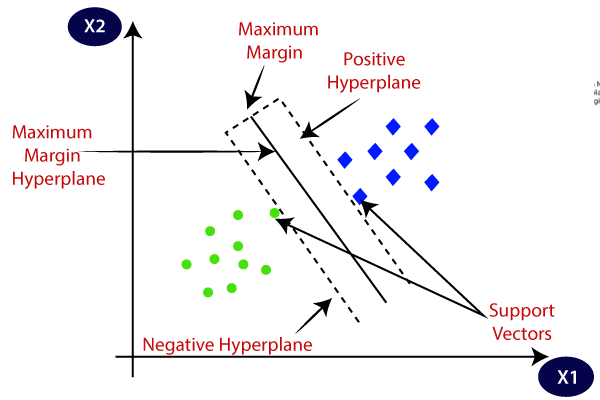


Fig4- Support Vector Algorithm

* **ARCHITECTURE DIAGRAM-**

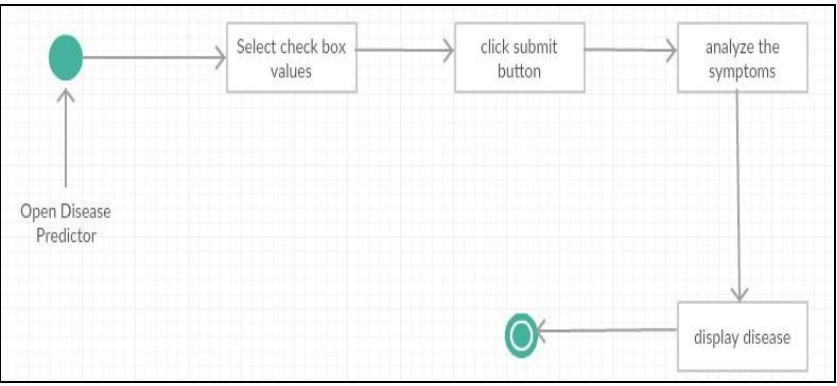
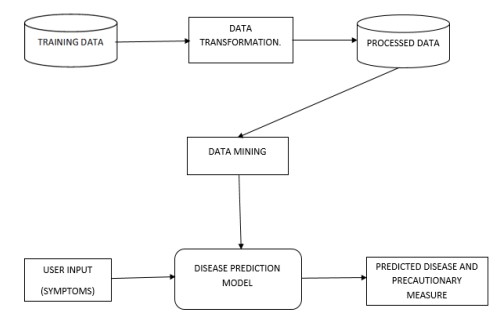


Fig 5 Architecture Diagram

* **DIAGRAM-**

The general disease prediction system predicts chance of presence of a disease present in a patient on the basis of their symptoms. It will also recommend necessary precautionary measures required to treat the predicted disease. The system will initially be fed data from different sources i.e. patients, the data will then be pre-processed before further process is carried out, this is done so as to get clean data from the raw initial data, as the raw data would be noisy, or flawed. This data will be processed using Data mining algorithms, the system, will be trained so as to predict the disease based on the input data given by the user.



**Block Diagram for disease prediction system.**

The system is implemented into two parts, admin part and the user part. The duty of the admin is training the system for creation of the disease prediction model. The user uses the services provided by the model after logging in as the user, entering the symptoms into the model, which in turn returns the predicted results and necessary precautionary measures.

#### O E-R DIAGRAM-

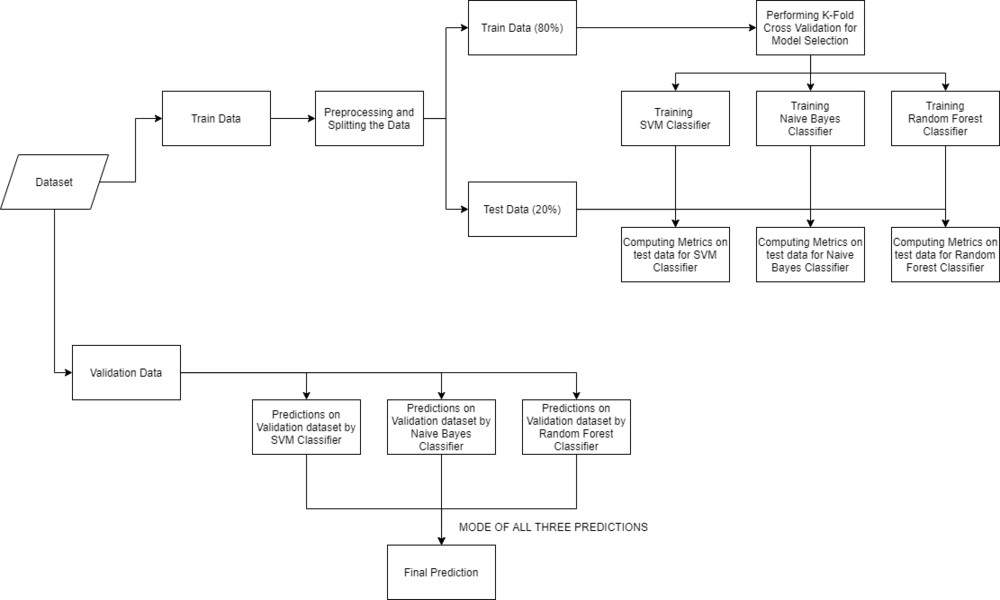


Fig 6 – E-R Diagram

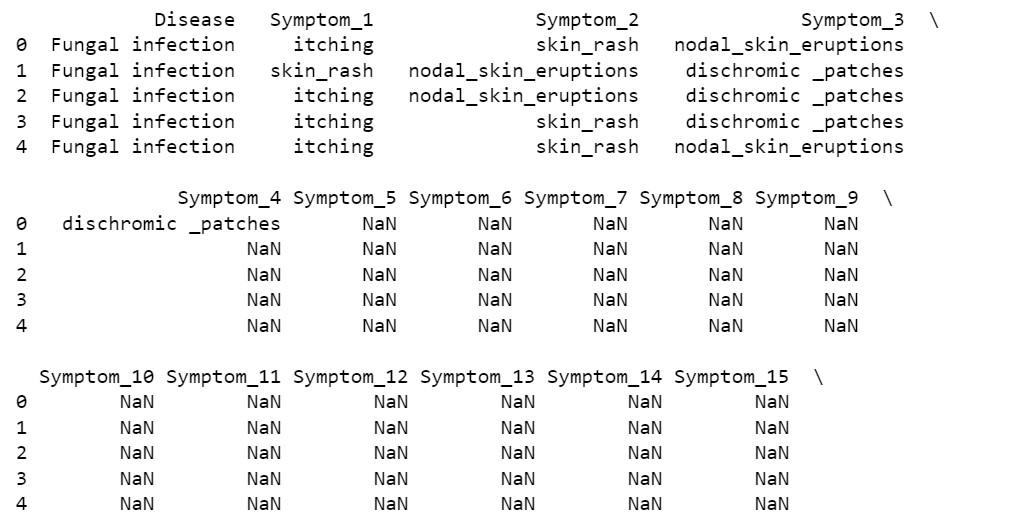
# 3.4 CODING

**import pandas as pd import numpy as np import matplotlib.pyplot as plt from sklearn.model\_selection import train\_test\_split from sklearn.svm import SVC**

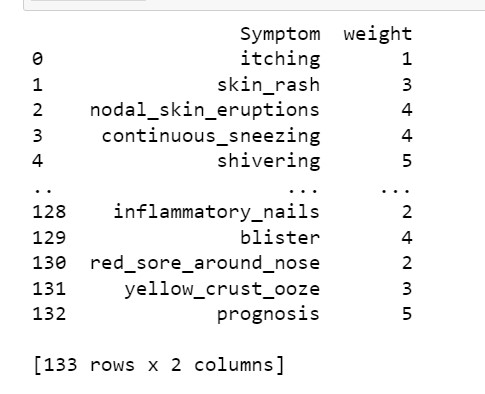
**from sklearn.metrics import f1\_score, accuracy\_score, confusion\_matrix import seaborn as sns from tkinter import \* from tkinter import messagebox import sys import urllib import urllib.request**

**df = pd.read\_csv('dataset.csv') print(df.head()) #df.describe()**

## OUTPUT-



**df1 = pd.read\_csv('Symptom-severity.csv') print(df1)**



**df.isna().sum() df.isnull().sum()**

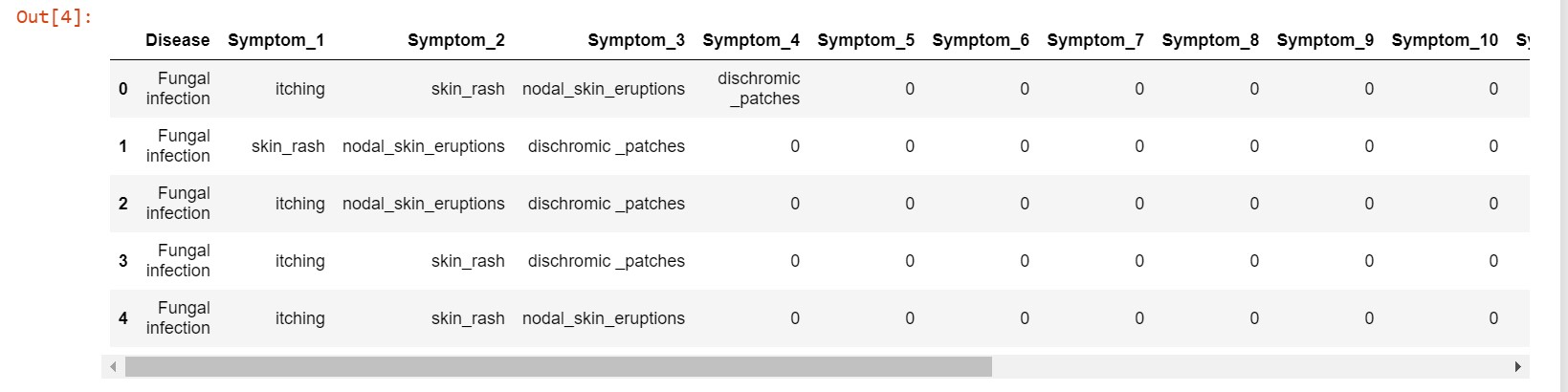
**cols = df.columns data = df[cols].values.flatten()**

**s = pd.Series(data) s = s.str.strip() s = s.values.reshape(df.shape)**

**df = pd.DataFrame(s, columns=df.columns)**

**df = df.fillna(0) df.head()**

**OUTPUT-**



**vals = df.values**

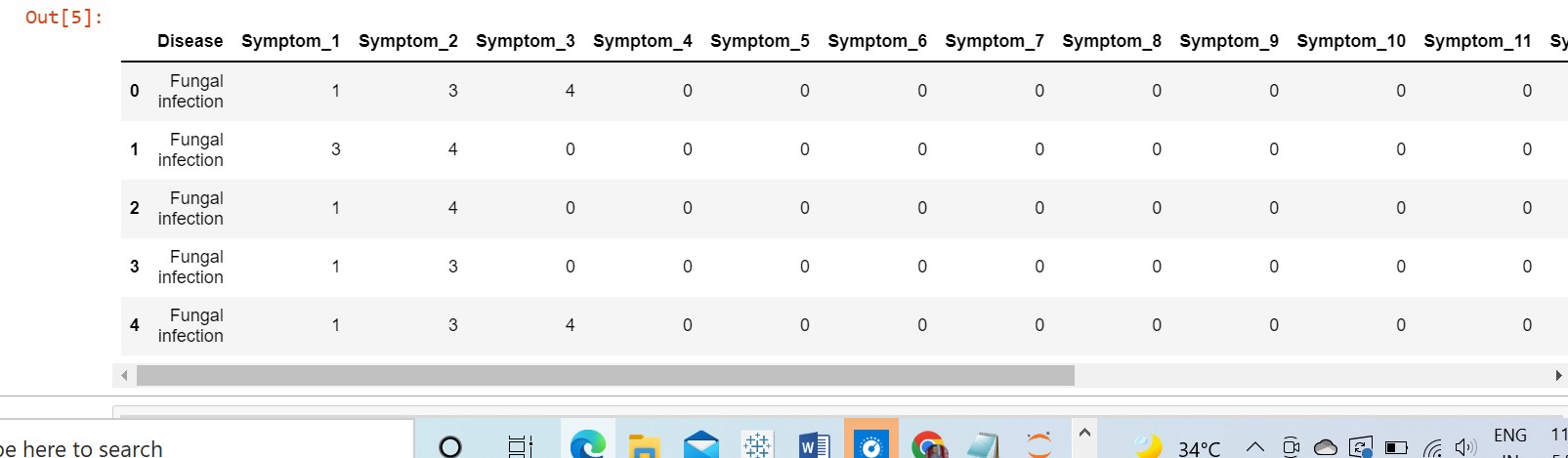
**symptoms = df1['Symptom'].unique()**

**for i in range(len(symptoms)): vals[vals == symptoms[i]] = df1[df1['Symptom'] == symptoms[i]]['weight'].values[0]**

**d = pd.DataFrame(vals, columns=cols)**

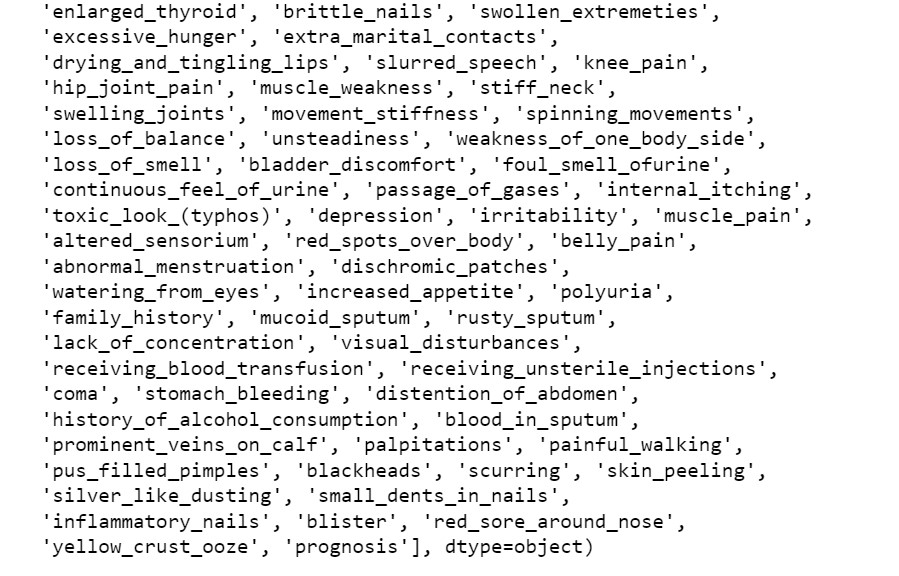
**d = d.replace('dischromic \_patches', 0) d = d.replace('spotting\_ urination',0) df = d.replace('foul\_smell\_of urine',0) df.head()**

**OUTPUT-**



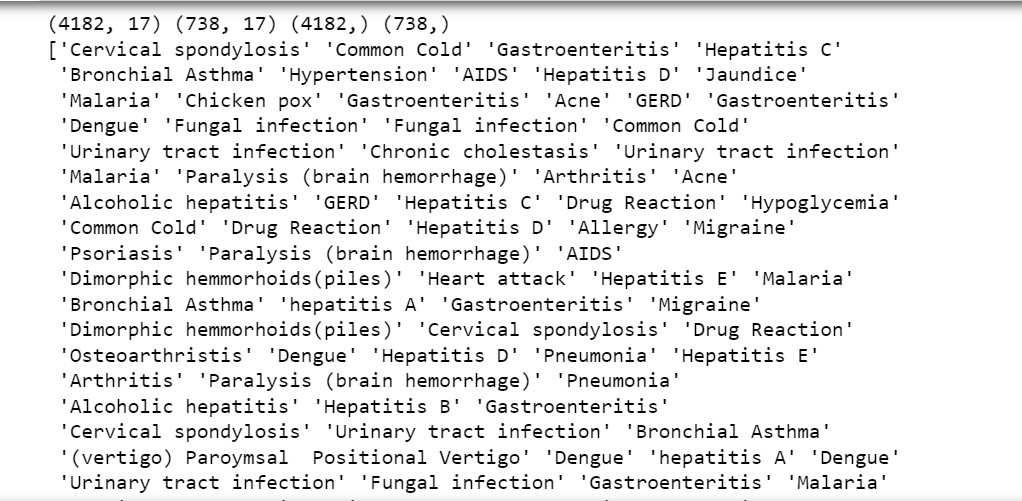
**(df[cols] == 0).all() df['Disease'].value\_counts() df['Disease'].unique() OUTPUT-**





**data = df.iloc[:,1:].values labels = df['Disease'].values x\_train, x\_test, y\_train, y\_test = train\_test\_split(data, labels, shuffle=True, train\_size = 0.85) print(x\_train.shape, x\_test.shape, y\_train.shape, y\_test.shape) model = SVC() model.fit(x\_train, y\_train)**

**preds = model.predict(x\_test) print(preds)**

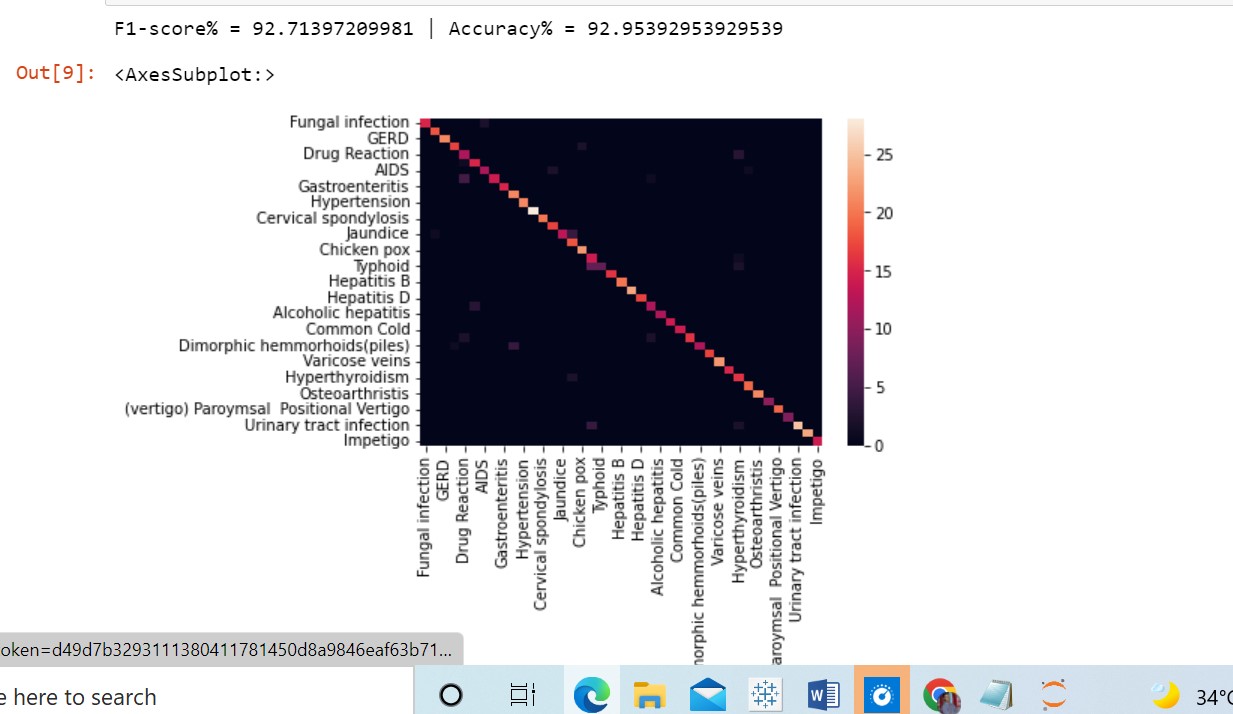


**conf\_mat = confusion\_matrix(y\_test, preds)**

**df\_cm = pd.DataFrame(conf\_mat, index=df['Disease'].unique(), columns=df['Disease'].unique())**

**print('F1-score% =', f1\_score(y\_test, preds, average='macro')\*100,**

**'|', 'Accuracy% =', accuracy\_score(y\_test, preds)\*100) sns.heatmap(df\_cm)**



**def message():**

**if (Symptom1.get() == "None" and Symptom2.get() == "None" and Symptom3.get() == "None" and Symptom4.get() == "None" and Symptom5.get() == "None"):**

**messagebox.showinfo("OPPS!!", "ENTER SYMPTOMS PLEASE") else :**

**SVM()**

**def SVM():**

**psymptoms =**

**[Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Sy mptom5.get()]**

**loc = location.get() a = np.array(df1["Symptom"]) b = np.array(df1["weight"]) for j in range(len(psymptoms)):**

**for k in range(len(a)): if psymptoms[j]==a[k]: psymptoms[j]=b[k]**

**nulls = [0,0,0,0,0,0,0,0,0,0,0,0] psy = [psymptoms + nulls] pred2 = model.predict(psy) t3.delete("1.0", END) t3.insert(END, pred2[0])**

**if(pred2[0]=="GERD"):**

**z=urllib.request.urlopen('https://api.thingspeak.com/update?api\_key =MP77HD9B13Z7N6BO&field1=1&field2=0&field3='+str(loc)) z.read() if(pred2[0]=="Hepatitis C"):**

**r=urllib.request.urlopen('https://api.thingspeak.com/update?api\_key =MP77HD9B13Z7N6BO&field1=0&field2=1&field3='+str(loc)) r.read()**

**root = Tk()**

**root.title(" Disease Prediction From Symptoms") root.configure(bg='salmon1')**

**Symptom1 = StringVar()**

**Symptom1.set(None)**

**Symptom2 = StringVar()**

**Symptom2.set(None)**

**Symptom3 = StringVar()**

**Symptom3.set(None)**

**Symptom4 = StringVar()**

**Symptom4.set(None)**

**Symptom5 = StringVar() Symptom5.set(None) location = StringVar() location.set(None)**

**w2 = Label(root, justify=CENTER, text=" Disease Prediction From Symptoms ")**

**w2.config(bg='orange red',font=("Times New Roman", 50)) w2.grid(row=1, column=0, columnspan=2, padx=100)**

**NameLb1 = Label(root, text="")**

**NameLb1.config(bg='salmon1',font=("Helvetica", 20))**

**NameLb1.grid(row=5, column=1, pady=10, sticky=W)**

**S1Lb = Label(root, text="Symptom 1")**

**S1Lb.config(bg='khaki1',font=("Helvetica", 15))**

**S1Lb.grid(row=7, column=1, pady=10 , sticky=W)**

**S2Lb = Label(root, text="Symptom 2")**

**S2Lb.config(bg='khaki1',font=("Helvetica", 15))**

**S2Lb.grid(row=8, column=1, pady=10, sticky=W)**

**S3Lb = Label(root, text="Symptom 3")**

**S3Lb.config(bg='khaki1',font=("Helvetica", 15))**

**S3Lb.grid(row=9, column=1, pady=10, sticky=W)**

**S4Lb = Label(root, text="Symptom 4")**

**S4Lb.config(bg='khaki1',font=("Helvetica", 15))**

**S4Lb.grid(row=10, column=1, pady=10, sticky=W)**

**S5Lb = Label(root, text="Symptom 5")**

**S5Lb.config(bg='khaki1',font=("Helvetica", 15)) S5Lb.grid(row=11, column=1, pady=10, sticky=W)**

**locLb = Label(root, text="Location") locLb.config(bg='khaki1',font=("Helvetica", 15)) locLb.grid(row=12, column=1, pady=10, sticky=W)**

**lr = Button(root, text="Predict",bg='gold',height=2, width=20,**

**command=message)**

**lr.config(font=("Helvetica", 15)) lr.grid(row=15, column=1,pady=10)**

**#OPTIONS = sorted(symptoms)**

**OPTIONS = ['itching', 'skin\_rash', 'nodal\_skin\_eruptions',**

**'continuous\_sneezing', 'shivering', 'chills', 'joint\_pain',**

**'stomach\_pain', 'acidity', 'ulcers\_on\_tongue', 'muscle\_wasting',**

**'vomiting', 'burning\_micturition', 'spotting\_urination', 'fatigue',**

**'weight\_gain', 'anxiety', 'cold\_hands\_and\_feets', 'mood\_swings',**

**'weight\_loss', 'restlessness', 'lethargy', 'patches\_in\_throat',**

**'irregular\_sugar\_level', 'cough', 'high\_fever', 'sunken\_eyes',**

**'breathlessness', 'sweating', 'dehydration', 'indigestion',**

**'headache', 'yellowish\_skin', 'dark\_urine', 'nausea',**

**'loss\_of\_appetite', 'pain\_behind\_the\_eyes', 'back\_pain',**

**'constipation', 'abdominal\_pain', 'diarrhoea', 'mild\_fever',**

**'yellow\_urine', 'yellowing\_of\_eyes', 'acute\_liver\_failure',**

**'fluid\_overload', 'swelling\_of\_stomach', 'swelled\_lymph\_nodes',**

**'malaise', 'blurred\_and\_distorted\_vision', 'phlegm',**

**'throat\_irritation', 'redness\_of\_eyes', 'sinus\_pressure', 'runny\_nose', 'congestion', 'chest\_pain', 'weakness\_in\_limbs',**

**'fast\_heart\_rate', 'pain\_during\_bowel\_movements',**

**'pain\_in\_anal\_region', 'bloody\_stool', 'irritation\_in\_anus',**

**'neck\_pain', 'dizziness', 'cramps', 'bruising', 'obesity',**

**'swollen\_legs', 'swollen\_blood\_vessels', 'puffy\_face\_and\_eyes',**

**'enlarged\_thyroid', 'brittle\_nails', 'swollen\_extremeties',**

**'excessive\_hunger', 'extra\_marital\_contacts',**

**'drying\_and\_tingling\_lips', 'slurred\_speech', 'knee\_pain',**

**'hip\_joint\_pain', 'muscle\_weakness', 'stiff\_neck',**

**'swelling\_joints', 'movement\_stiffness', 'spinning\_movements',**

**'loss\_of\_balance', 'unsteadiness', 'weakness\_of\_one\_body\_side',**

**'loss\_of\_smell', 'bladder\_discomfort', 'foul\_smell\_ofurine',**

**'continuous\_feel\_of\_urine', 'passage\_of\_gases', 'internal\_itching',**

**'toxic\_look\_(typhos)', 'depression', 'irritability', 'muscle\_pain',**

**'altered\_sensorium', 'red\_spots\_over\_body', 'belly\_pain',**

**'abnormal\_menstruation', 'dischromic\_patches',**

**'watering\_from\_eyes', 'increased\_appetite', 'polyuria',**

**'family\_history', 'mucoid\_sputum', 'rusty\_sputum',**

**'lack\_of\_concentration', 'visual\_disturbances',**

**'receiving\_blood\_transfusion', 'receiving\_unsterile\_injections',**

**'coma', 'stomach\_bleeding', 'distention\_of\_abdomen',**

**'history\_of\_alcohol\_consumption', 'blood\_in\_sputum',**

**'prominent\_veins\_on\_calf', 'palpitations', 'painful\_walking', 'pus\_filled\_pimples', 'blackheads', 'scurring', 'skin\_peeling',**

**'silver\_like\_dusting', 'small\_dents\_in\_nails',**

**'inflammatory\_nails', 'blister', 'red\_sore\_around\_nose',**

**'yellow\_crust\_ooze', 'prognosis']**

**LOCATIONS = ["New Delhi", "Mumbai", "Chennai", "Kolkata", "Bengaluru","Bhopal"]**

**S1En = OptionMenu(root, Symptom1,\*OPTIONS)**

**S1En.config(bg='khaki1',font=("Helvetica", 12)) S1En.grid(row=7, column=1)**

**S2En = OptionMenu(root, Symptom2,\*OPTIONS)**

**S2En.config(bg='khaki1',font=("Helvetica", 12)) S2En.grid(row=8, column=1)**

**S3En = OptionMenu(root, Symptom3,\*OPTIONS)**

**S3En.config(bg='khaki1',font=("Helvetica", 12)) S3En.grid(row=9, column=1)**

**S4En = OptionMenu(root, Symptom4,\*OPTIONS) S4En.config(bg='khaki1',font=("Helvetica", 12)) S4En.grid(row=10, column=1)**

**S5En = OptionMenu(root, Symptom5,\*OPTIONS)**

**S5En.config(bg='khaki1',font=("Helvetica", 12)) S5En.grid(row=11, column=1)**

**LocEn = OptionMenu(root, location,\*LOCATIONS)**

**LocEn.config(bg='khaki1',font=("Helvetica", 12)) LocEn.grid(row=12, column=1)**

**NameLb = Label(root, text="")**

**NameLb.config(bg='salmon1',font=("Helvetica", 20))**

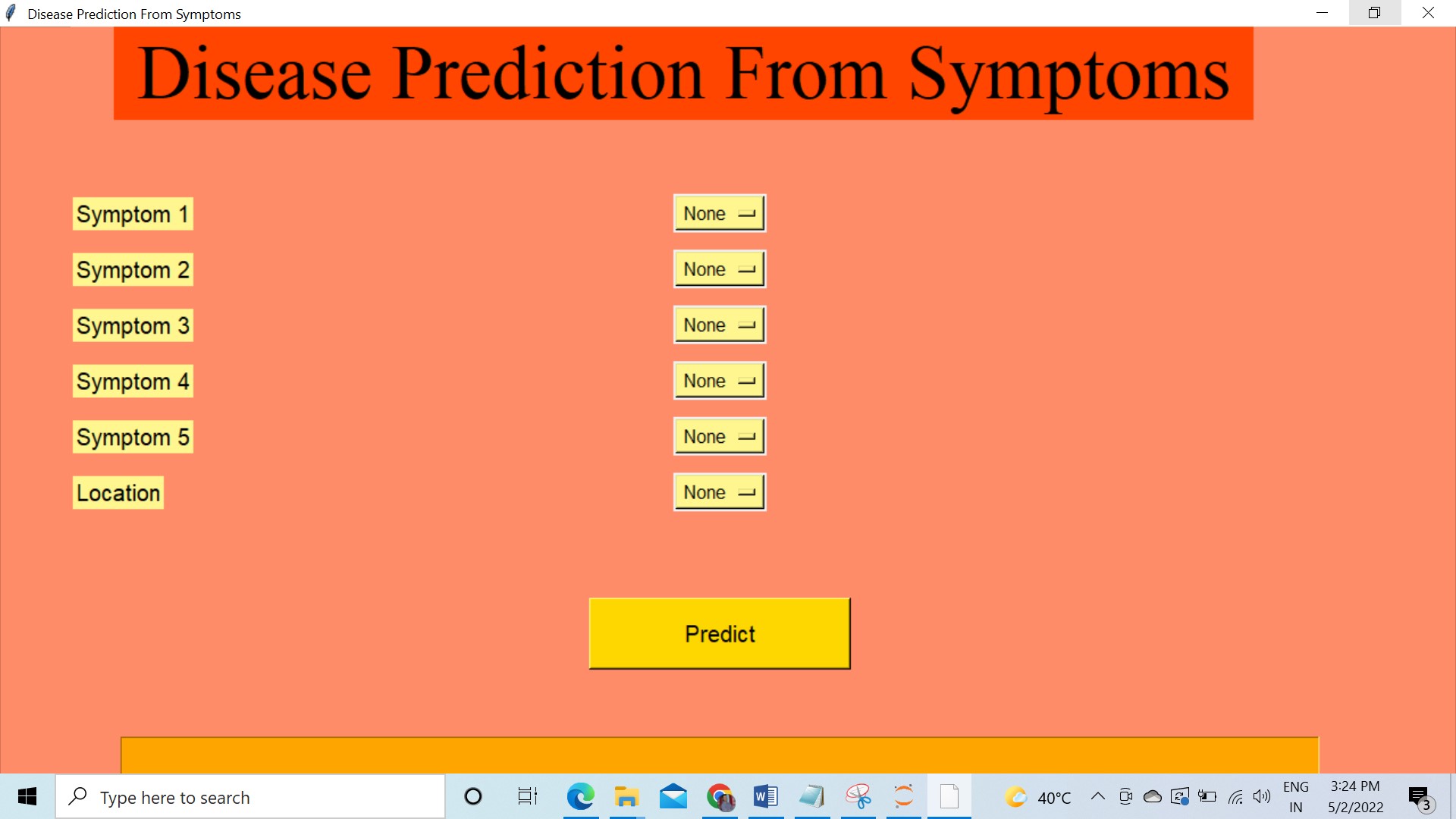
**NameLb.grid(row=13, column=1, pady=10, sticky=W)**

**NameLb = Label(root, text="")**

**NameLb.config(bg='salmon1',font=("Helvetica", 15)) NameLb.grid(row=18, column=1, pady=10, sticky=W)**

**t3 = Text(root, height=2, width=70) t3.config(bg='orange',font=("Helvetica", 20)) t3.grid(row=19, column=1 , padx=10) root.mainloop()**

# OUTPUT



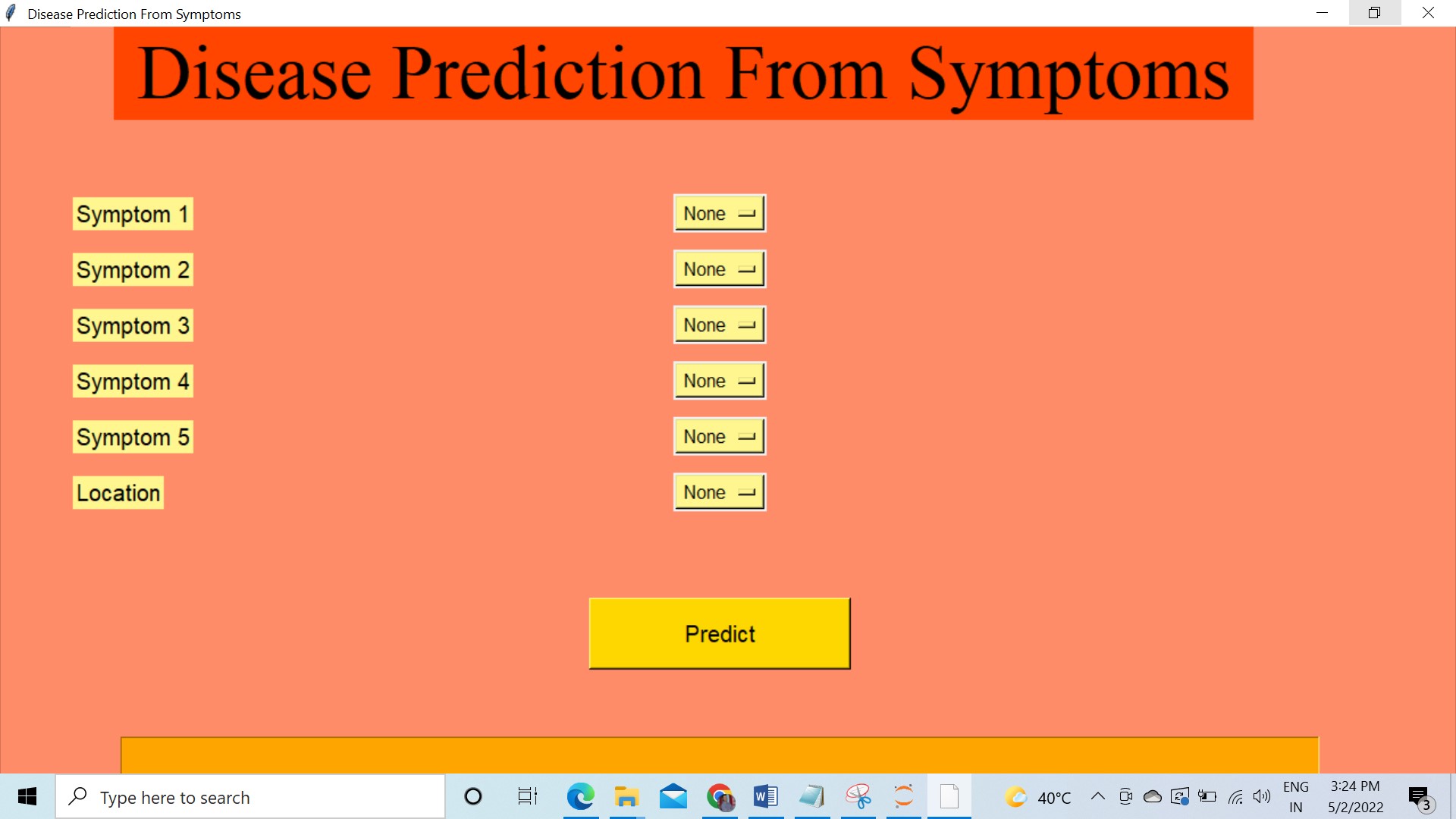
# CHAPTER 4 IMPLEMENTATION OF RESULTS

**4.1 EXTERNAL INTERFACE REQUIREMENTS**

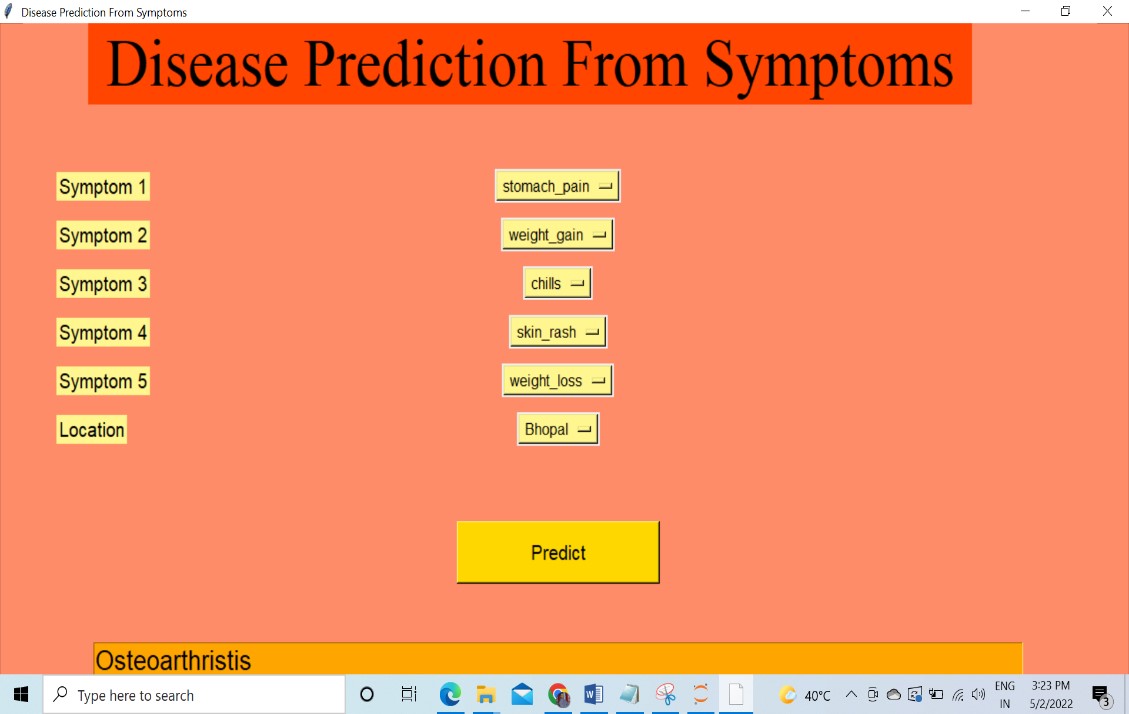
These requirements include user interfaces (interaction logic between software and user), screen layouts, buttons, functions on every screen, hardware interfaces (here a team describes what devices the software is created for), and other relevant particularities.

## 4.1.1 USER INTERFACE-

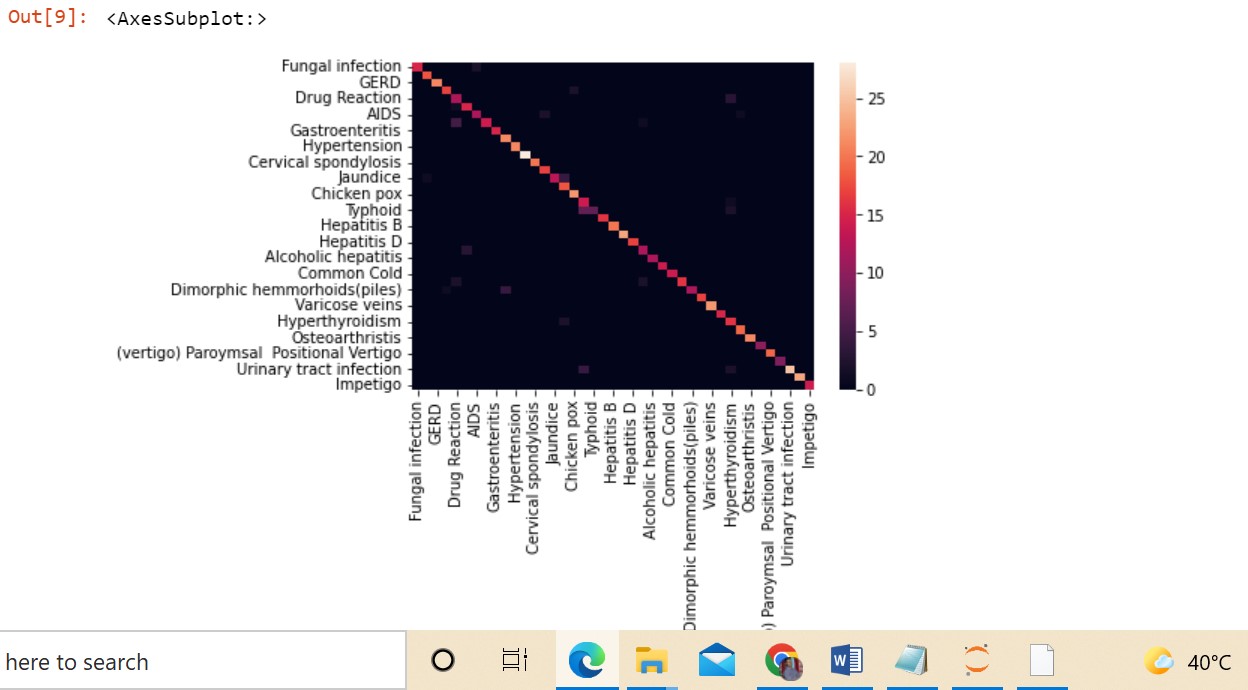
User interface (UI) design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable. UI design refers to graphical user interfaces and other forms

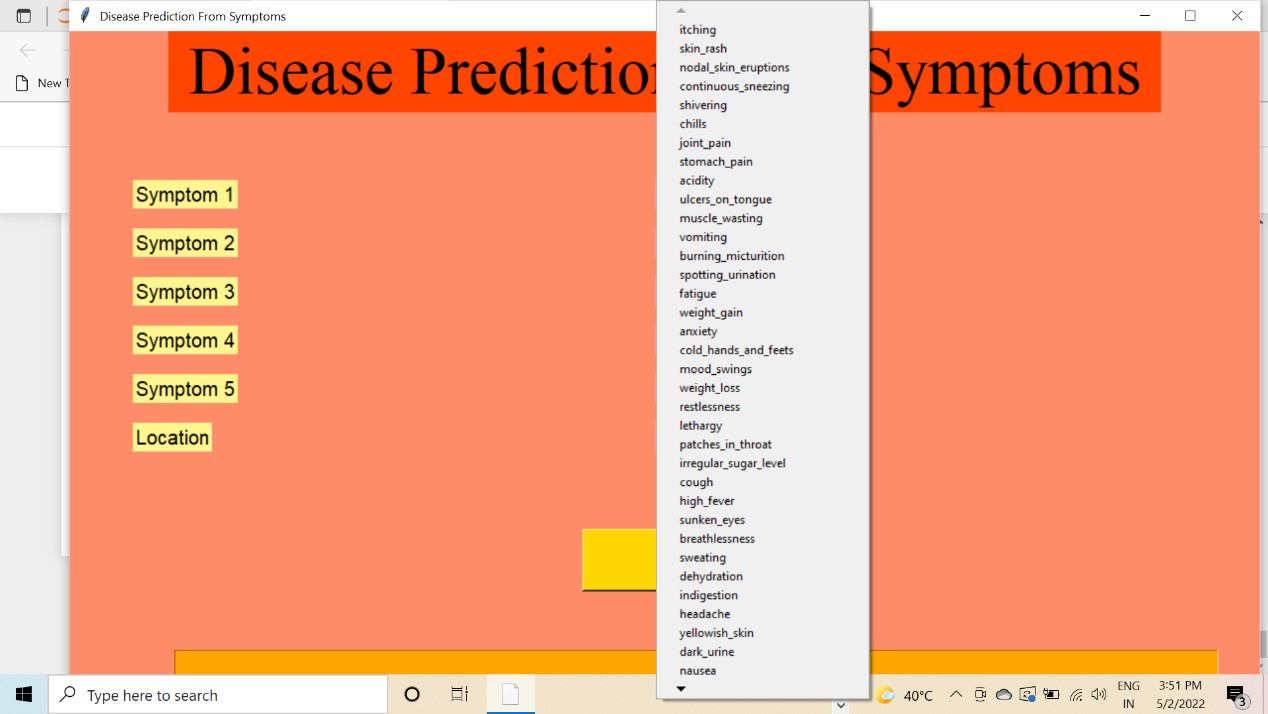


##  SOFTWARE INTERFACE-



# 4.2 SCREEN SHOT WITH DISCRIPTION





From the above output, we can notice that our machine learning algorithm are performing very well and the mean scores after k fold cross-validation are also very high. To build a robust model we can combine i.e. take the mode of the predictions of all three models so that even one of the models makes wrong predictions and the other two make correct predictions then the final output would be the correct one. This approach will help us to keep the predictions much more accurate on

completely unseen data. In the below code we will be training all the three models on the train data, checking the quality of our models using a confusion matrix, and then combine the predictions of all the three models.

**4.3** **ADVANTAGES**

* Increased accuracy for effective disease.
* Handles roughest (enormous) amount of data using Support Vector Machine algorithm and feature selection.
* Reduce the time complexity of doctors. Cost effective for patients.

C**HAPTER 5**

# CONCLUSION AND FUTURE SCOPE

**5.1 CONCLUSION**

diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the seven different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting applied on the dataset.

The system has been implemented with the accuracy of 92.95392953929539 % on the dataset of patient data. The current system covers only the diseases or the more commonly occurring disease, the plan is to include disease of higher fatality, like various cancers in future, so that early prediction and treatment could be done, and the fatality rate of deadly diseases like cancer decreases, with the economic benefit in long sight as well.

The expected attributes leading to heart disease in patients are available in the dataset which contains 76 features and 14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done. In this n features have to be selected for evaluating the model which gives more accuracy. The correlation of some features in the dataset is almost equal and so they are removed. If all the attributes present in the dataset are taken into account then the efficiency decreases considerably.

# 5.2 FUTURE SCOPE

The goal of our disease prediction project is **to determine if a patient should be diagnosed with disease or not**, which is a binary outcome, so: Positive result = 1, the patient will be diagnosed with heart disease. Negative result = 0, the patient will not be diagnosed with heart disease

38