VISVESVARAYATECHNOLOGICALUNIVERSITY

Jnana Sangama, Belagavi-590018

Assignment on

"Disease Prediction System"

Submitted in Partial fulfillment of

Bachelor of Engineering in

Artificial Intelligence and Data Science

By

Avantika Gupta (1CR22AD016) , G Harshitha Singh (1CR22AD034),
J Jessica Maris(1CR22AD044), Kondreddy Charani Sai Reddy(1CR22AD057),
Chandana P(1CR22AD023)

Under the Guidance of,

Prof Anushree Paul, Assistant Professor, Dept. of AI&DS

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with "A++" Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with "A++" Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CERTIFICATE

This is to certify that the Assignment entitled **"Disease Prediction System"** has been carried out by Avantika Gupta(1CR22AD016), G Harshitha Singh(1CR22AD034), J Jessica Maris(1CR22AD044), Kondreddy Charani Sai Reddy(1CR22AD057) and Chandana P(1CR22AD023) bonafide students of CMR Institute of Technology, Bengaluru in partial fulfillment for the award of the Degree of **Bachelor of Engineering in Artificial Intelligence and Data Science** of the Visvesvaraya Technological University, Belagavi during the year **2024-2025.** It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library. This Assignmenet report has been approved as it satisfies the academic requirements in respect of Assignment prescribed for the said Degree.

Signature of Guide	**Signature of HOD**
Ms. Anushree Paul	**Dr Shanthi M B**
Assistant Professor	**Professor & HoD**
Dept. of Artificial Intelligence and Data Science, CMRIT	**Dept. of Artificial Intelligence and Data Science, CMRIT**

DECLARATION

Ms. Avantika Gupta USN: 1CR22AD016 , Ms. G Harshitha Singh USN: 1CR22AD034, Ms. J Jessica Maris USN: 1CR22AD044 Ms. Kondreddy Charani Sai Reddy USN: 1CR22AD057 , Ms. Chandana P USN: 1CR22AD020 , hereby declare that the assignment report entitled "**Disease Prediction System**" has been carried out by us under the guidance of **Professor Anushree Paul**, Assistant Professor, Department of Artificial Intelligence in partial fulfillment of the requirement for the degree of BACHELOR OF ENGINEERING in **Artificial Intelligence and Data Science**, of Visvesvaraya Technological University, Belgaum during the academic year 2024-2025.The work done in this assignment report is original and it has not been submitted for any other degree in any university.

Place: Bangalore

Date:

Avantika Gupta(1CR22AD016)

G Harshitha Singh (1CR22AD034)

J Jessica Maris (1CR22AD044)

Kondreddy Charani Sai Reddy (1CR22AD057)

Chandana P (1CR22AD023)

ABSTRACT

The world is moving with a fast speed and in order to keep up with the whole world we tend to ignore the symptoms of disease which can affect our health to a large extent. Many working professional's get heart attacks, bad cholesterol, eye disease and they are unable to treat it at the right time as they are busy coping up with progressive world. God has granted each and every individual a beautiful gift called life, so it is our responsibility to live our life to fullest and try to stay safe from the dangers of the world. So we have developed a logistic regression model with the help of machine learning algorithms like decision tree, random forest,k-nearest neighbour and naïve Bayes which take into account the symptoms felt by a person and according to those symptoms it predicts the disease which the person can be suffering from. It saves time as well as makes it easy to get a warning about your health before it's too late.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out the **Disease Prediction System** Assignment.

It gives me an immense pleasure to express my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bengaluru, for his constant encouragement.

I would like to extend my sincere gratitude to **Dr Shanthi M B,** HOD, Department of Artificial Intelligence and Data Science, CMRIT, Bengaluru, who has been a constant support and encouragement throughout the course of this assignment.

I would like to thank my guide **Professor Anushree paul** for the valuable guidance throughout the tenure of this assignment.

I would also like to thank all the faculty members of the Artificial Intelligence and Data Science who directly or indirectly encouraged me.

Finally, I thank my parents and friends for all the moral support they have given me during the completion of this work.

GROUP MEMBER LIST

SLNO	NAME OF STUDENT	USN
01	Avantika Gupta	1CR22AD016
02	Harshitha Singh	1CR22AD034
03	J Jessica Maris	1CR22AD044
04	Kondreddy Charani Sai Reddy	1CR22AD017
05	Chandana P	1CR22AD023

TABLE OF CONTENTS

OVERVIEW

Introduction

Our assignment 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. Prediction of disease by looking at the symptoms is an integral part of treatment. In our assignment we have tried to accurately predict a disease by looking at the symptoms of the patient. We have used 4 different algorithms for this purpose and gained an accuracy of 92-95%.Such a system can have a very large potential in medical treatment of the future. We have also designed an interactive interface to facilitate interaction with the system. We have also attempted to show and visualize the result of our study and this assignment.

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 reduce redundancy in our data i.c. outliers, missing values etc. Then we treat our data by replacing the missing values, as python is a case sensitive programming language we transform all the letters into capital. Creating dummy variables to sort our data into mutually exclusive categories also means the no of dummy variables should be less than the no of categories of a qualitative variable. Also many people make the mistake of replacing the missing values with the mean of that variable but by doing so you can miss very important variations in the data.

Objectives

The primary aim of this assignment is to design and implement a machine learning-based disease prediction system that utilizes patient symptoms to predict the most likely disease. The system is expected to aid in early diagnosis and decision-making in the healthcare domain. Below are the detailed objectives that guided the development of the system:

Develop a Symptom-Based Disease Prediction Model

To build a reliable and efficient prediction model capable of identifying diseases using only symptoms as input.

To explore multiple machine learning algorithms and identify which provides the best performance in

terms of accuracy, precision, recall, and overall robustness.

Evaluate and Compare Multiple Machine Learning Algorithms

To train and test different supervised learning algorithms, such as Decision Tree, Random Forest, Support Vector Machine, and Naive Bayes, using a common dataset.

To analyze and compare the performance of each algorithm to determine the most suitable one for deployment in a real-world setting.

Achieve High Prediction Accuracy

To aim for a prediction accuracy in the range of 90% and above by fine-tuning hyperparameters, cleaning the data thoroughly, and using cross-validation techniques.

To reduce false positives and false negatives, ensuring the model's predictions are not just accurate but also medically relevant and safe for recommendation.

Handle Real-World Medical Data

To preprocess medical symptom-disease datasets effectively by addressing challenges such as missing data, redundancy, inconsistent text formats, and the presence of outliers.

To apply best practices in data transformation, such as encoding categorical variables and standardizing data formats for case-sensitive programming environments like Python.

Design a User-Friendly Interface

To develop an intuitive graphical user interface (GUI) that allows users—both medical professionals and patients—to interact with the system easily.

To ensure that the system can take user input (symptoms) and return a predicted disease along with confidence scores or additional insights.

Visualize Data and Results

To provide meaningful visualizations such as charts, graphs, and tables that represent data distributions, model performance, and prediction results.

To aid in the interpretability of the model for non-technical stakeholders, enhancing its practical usability in the healthcare domain.

Contribute to the Field of Smart Healthcare

To demonstrate the potential application of artificial intelligence and machine learning in modern healthcare systems.

To encourage future enhancements and integrations, such as connecting the model with electronic health records (EHRs) or wearable IoT devices for real-time prediction.

Ensure Model Scalability and Future Enhancements

To build a modular and scalable system that can be improved over time by incorporating additional data, features, and advanced AI techniques.

To create a foundation that can be expanded to include rare diseases, multilingual input, and personalized medicine based on patient history and demographics.

PROBLEM STATEMENT & PROPOSED METHODOLOGY

Problem Statement

Traditional input methods often exclude individuals with physical disabilities and struggle in multitasking environments, highlighting the need for more accessible and versatile solutions. Eye movement-based cursor control offers a promising alternative, leveraging natural interactions to enhance accessibility, productivity, and user experiences across diverse contexts.

Proposed Methodology

Dataset

Dataset for this assignment was collected from a study of the University of Columbia performed at New York Presbyterian Hospital during 2004. Link of dataset is given below.

http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

Library Used

In this assignment standard libraries for database analysis and model creation are used. The following are the libraries used in this assignment.

1. tkinter: It's a standard GUI library of python. Python when combined with tkinter provides a fast and easy way to create GUI. It provides a powerful object-oriented tool for creating GUI.It was used in this assignment to create our GUI namely messagebox, button, label, Option Menu, text and title. Using tkinter we were able to create an interactive GUI for our model.
2. Numpy: Numpy is the core library of scientific computing in python. It provides powerful tools to deal with various multi-dimensional arrays in python. It is a general purpose array processing package.
3. pandas: it is the most popular python library used for data analysis. It provides highly optimized performance with back-end source code purely written in C or python.
4. sklearn: Sklearn is an open source python library which implements a huge range of machine-learning, pre-processing, cross-validation and visualization algorithms. It features various simple and efficient tools for data mining and data processing. It features various classification, regression and clustering algorithm such as support vector machine, random forest classifier, decision tree, gaussian

naïve-Bayes, KNN to name a few.In this assignment we have used sklearn to get advantage of inbuilt classification algorithms like decision tree, random forest classifier, k-nearest neighbour and naïve Bayes. We have also used inbuilt cross validation and visualization features such as classification report, confusion matrix and accuracy score.

In this assignment we are using four algorithms to predict disease based on symptoms. They are

1.Decision tree

2.Random forest tree

3.Gaussian Naive Bayes

4.KNN

1. Decision tree

Decision tree is a supervised learning technique program used for classification problems.It is also capable of engaging problems of higher dimensionality. It mainly consists of three parts: root, nodes and leaf.

This prediction method gives accuracy of ~95%.

2. Random Forest

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

For each decision tree, Scikit-learn calculates a nodes importance using Gini Importance, assuming only two child nodes (binary tree), nij=wjCj - wleftjCleftj - wrightjCrightj

where,

ni sub(j)= the importance of node

W sub(j)=weighted number of samples reaching node j

C sub(j)=the impurity value of node j

left(j) = child node from left split on node j

right(j)= child node from right split on node j

This prediction method is used with 100 random samples and gives accuracy of ~95%.

3. k Nearest Neighbor

KNN is a simple, easy to implement supervised learning algorithm used for classification and regression problems. It works by finding a pattern in data which links data to results and it improves upon the pattern recognition with every iteration.

Assume we are given a dataset where X is a matrix of features from an observation and Y is a class label. We will use this notation throughout this article. k-nearest neighbors then, is a method of classification that estimates the conditional distribution of Y given X and classifies an observation to the class with the highest probability. Given a positive integer k,

k-nearest neighbors looks at the k observations closest to a test observation x0 and estimates the conditional probability that it belongs to class j using the formula,

$$\mathbf{Pr(Y=j|X=x0)=(1/k)\sum_{i \in N0} I(yi=j)}$$

where,

NO=set of k nearest observations

I(yi=j)=indicator variable that evaluates to 1 if given observation (xi,yi) in N0 is member of j,and 0 if otherwise

This prediction method has accuracy of~92%.

4. Naive Bayes Algorithm

Naïve Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification problems.

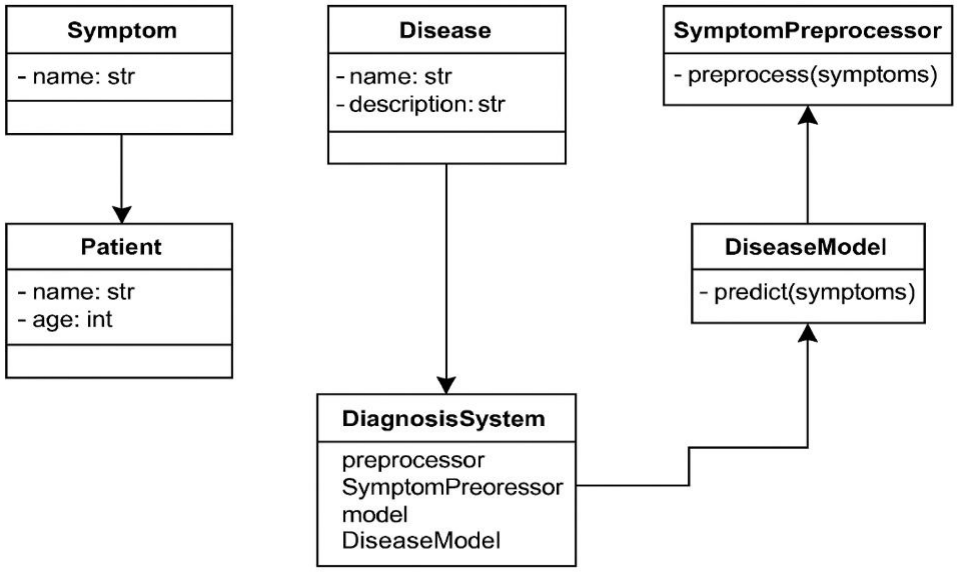
Bayes theorem is a mathematical formula used for calculating conditional probability. Conditional probability is a measure of the probability of an event occurring given that another event has (by assumption, presumption, assertion, or evidence) occurred. Formula is:

P(A/B) = (P(B/A)*P(A))/P(B)

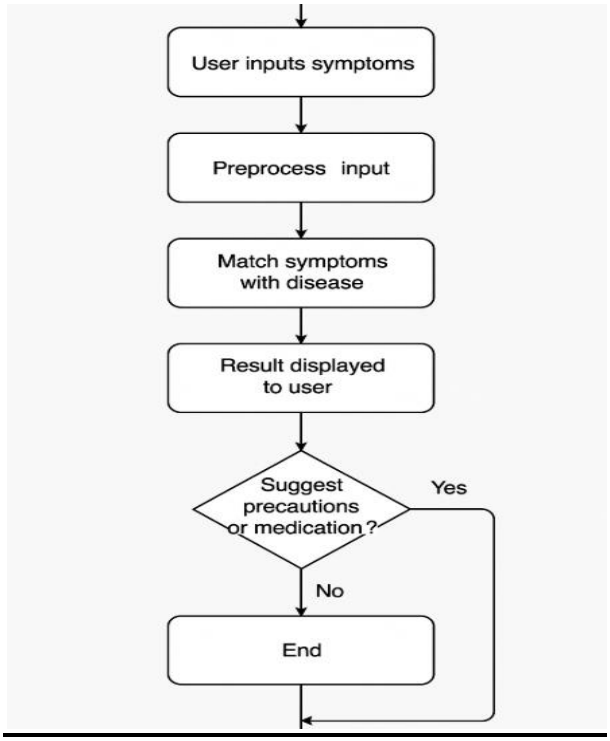
This prediction method has an accuracy of 95%.

EXPERIMENTAL SETUP & ANALYSIS

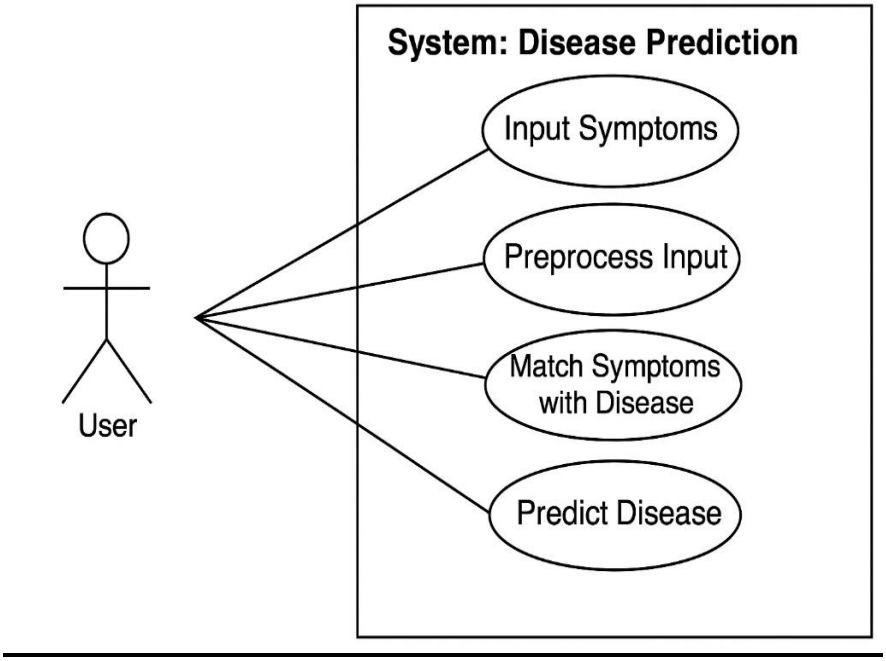
Class Diagram



Activity Diagram



UML Diagram


System: Disease Prediction
Input Symptoms
Preprocess Input
Match Symptoms with Disease
Predict Disease
User

Code

In [6]:
```
#Importing Libraries
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from tkinter import *
import numpy as np
import pandas as pd
import os
```

In [7]:
```
#List of the symptoms is listed here in list l1.

l1=['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_of urine',
    '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','
    'rusty_sputum','lack_of_concentration','visual_disturbances','receiving_blood_transfusion',
    'receiving_unsterile_injections','coma','stomach_bleeding','distention_of_abdomen',
    'history_of_alcohol_consumption','fluid_overload','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']
```

In [8]:
```
#List of Diseases is listed in list disease.

disease=['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes ',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migraine',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
       'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
       'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
       'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumonia',
       'Dimorphic hemmorhoids(piles)', 'Heart attack', 'Varicose veins',
       'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo']

#disease = [df['prognosis'].unique()]
#print(disease)
```

In [9]:
```
l2=[]
for i in range(0,len(l1)):
    l2.append(0)
print(l2)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [10]:

```
#Reading the training .csv file
df=pd.read_csv("Dataset/training.csv")
DF= pd.read_csv('Dataset/training.csv', index_col='prognosis')
#Replace the values in the imported file by pandas by the inbuilt function replace in pandas.

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug Reaction':4,
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asthma':9,'Hypertension ':10,
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatitis':24,'Tuberculosis':25,
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Varicose veins':30,'Hypothyroidi
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':38,'Psoriasis':39,
    'Impetigo':40}},inplace=True)
#df.head()
DF.head()
```

Out[10]:

prognosis	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on
Fungal infection	1	1	1	0	0	0	0	0	0	
Fungal infection	0	1	1	0	0	0	0	0	0	
Fungal infection	1	0	1	0	0	0	0	0	0	
Fungal infection	1	1	0	0	0	0	0	0	0	
Fungal infection	1	1	1	0	0	0	0	0	0	

5 rows × 132 columns

In [11]:

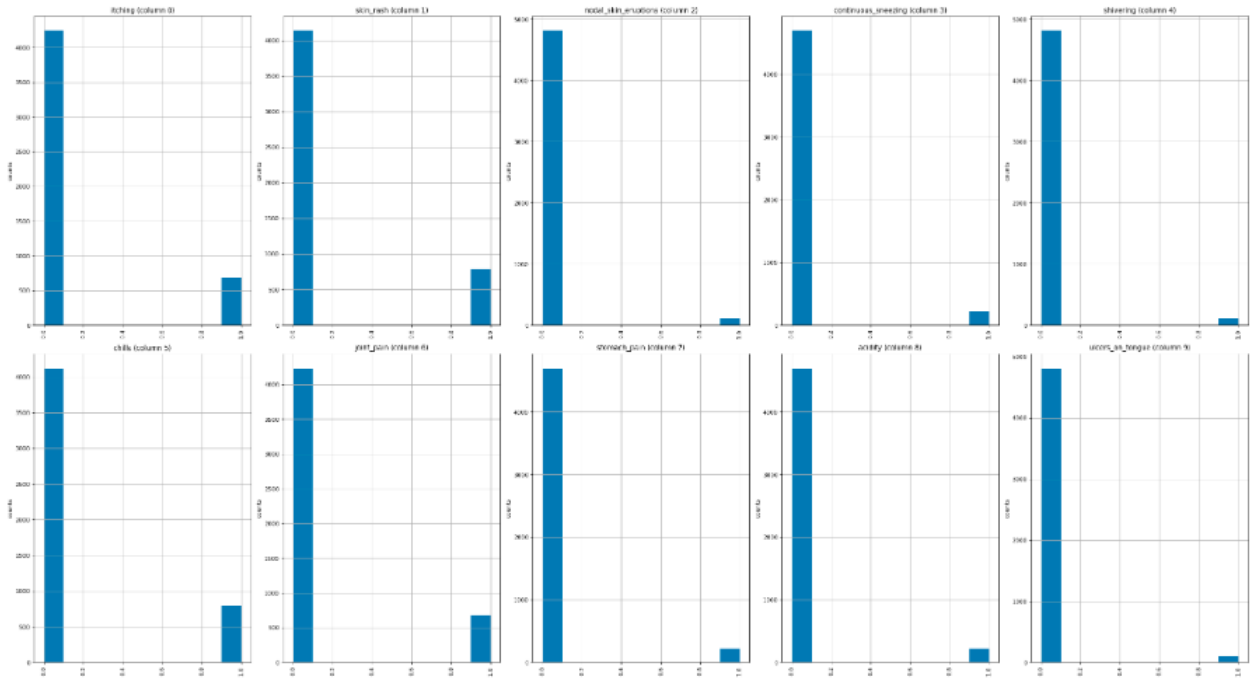
```
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow):
    nunique = df1.nunique()
    df1 = df1[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For displaying purposes, pick columns that
    nRow, nCol = df1.shape
    columnNames = list(df1)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80, facecolor = 'w', edgecolor = 'k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

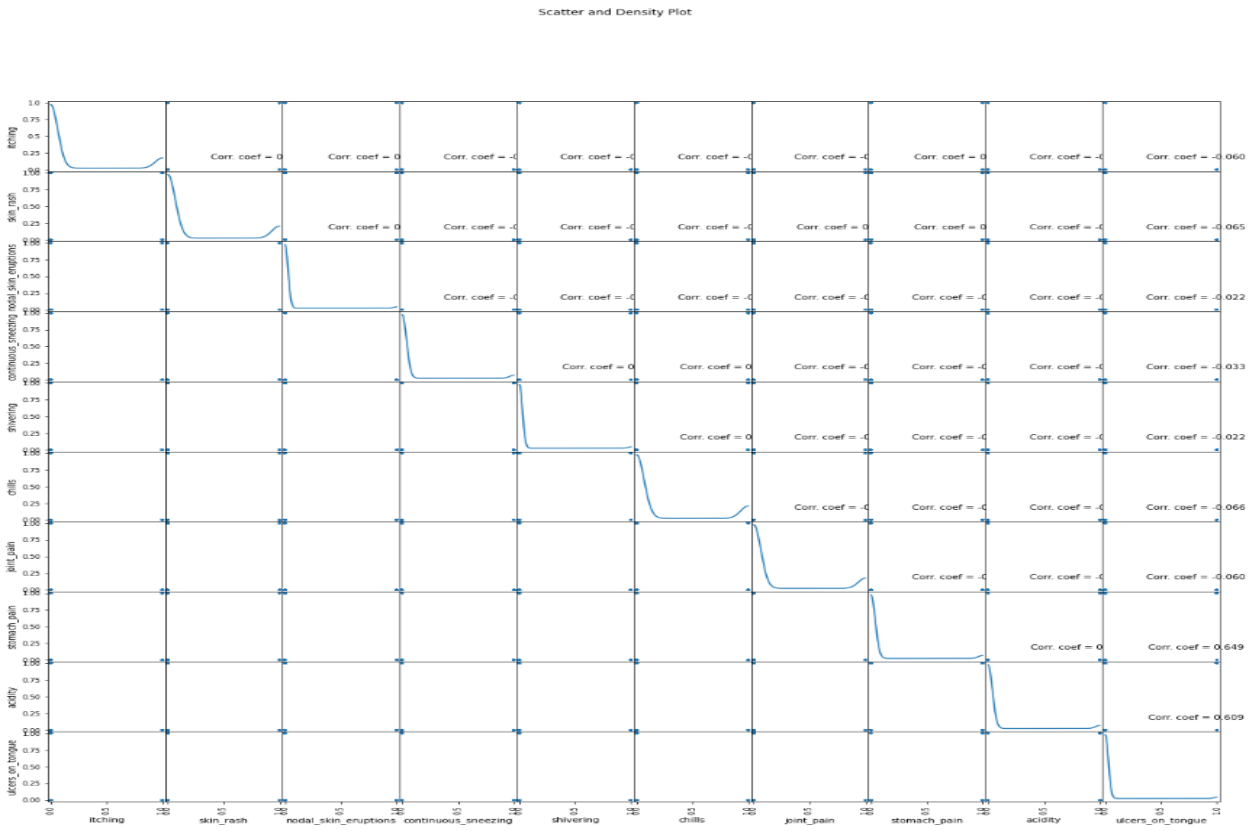
In [12]:

```
# Scatter and density plots
def plotScatterMatrix(df1, plotSize, textSize):
    df1 = df1.select_dtypes(include =[np.number]) # keep only numerical columns
    # Remove rows and columns that would lead to df being singular
    df1 = df1.dropna('columns')
    df1 = df1[[col for col in df if df[col].nunique() > 1]] # keep columns where there are more than 1 unique values
    columnNames = list(df)
    if len(columnNames) > 10: # reduce the number of columns for matrix inversion of kernel density plots
        columnNames = columnNames[:10]
    df1 = df1[columnNames]
    ax = pd.plotting.scatter_matrix(df1, alpha=0.75, figsize=[plotSize, plotSize], diagonal='kde')
    corrs = df1.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes fraction', ha='center', va='cente
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

In [13]:

```
plotPerColumnDistribution(df, 10, 5)
```





In [15]:
```
X= df[l1]
y = df[["prognosis"]]
np.ravel(y)
print(X)
```

```
      back_pain  constipation  abdominal_pain  diarrhoea  mild_fever  \
0             0             0               0          0           0
1             0             0               0          0           0
2             0             0               0          0           0
3             0             0               0          0           0
4             0             0               0          0           0
...         ...           ...             ...        ...         ...
4915          0             0               0          0           0
4916          0             0               0          0           0
4917          0             0               0          0           0
4918          0             0               0          0           0
4919          0             0               0          0           0

      yellow_urine  yellowing_of_eyes  acute_liver_failure  fluid_overload
0                0                  0                    0               0
1                0                  0                    0               0
2                0                  0                    0               0
3                0                  0                    0               0
4                0                  0                    0               0
...            ...                ...                  ...             ...
4915             0                  0                    0               0
4916             0                  0                    0               0
4917             0                  0                    0               0
4918             0                  0                    0               0
4919             0                  0                    0               0
```

Out[17]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue
0	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0
2	0	0	0	0	0	0	0	1	1	1
3	1	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	1	0	0

5 rows × 133 columns

In [26]:

```
pred3=StringVar()
def NaiveBayes():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb=gnb.fit(X,np.ravel(y))

        from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
        y_pred=gnb.predict(X_test)
        print("Naive Bayes")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1
inputtest = [l2]
predict = gnb.predict(inputtest)
predicted=predict[0]

h='no'
for a in range(0,len(disease)):
    if(predicted == a):
        h='yes'
        break
if (h=='yes'):
    pred3.set(" ")
    pred3.set(disease[a])
else:
    pred3.set(" ")
    pred3.set("Not Found")
 #Creating the database if not exists named as database.db and creating table if not exists named as NaiveBayes u
import sqlite3
conn = sqlite3.connect('database.db')
c = conn.cursor()
c.execute("CREATE TABLE IF NOT EXISTS NaiveBayes(Name StringVar,Symtom1 StringVar,Symtom2 StringVar,Symtom3 Strin
c.execute("INSERT INTO NaiveBayes(Name,Symtom1,Symtom2,Symtom3,Symtom4,Symtom5,Disease) VALUES(?,?,?,?,?,?,?)",(N
conn.commit()
c.close()
conn.close()
#printing scatter plot of disease predicted vs its symptoms
scatterplt(pred3.get())
```

In [28]:
```
Symptom1 = StringVar()
Symptom1.set("Select Here")

Symptom2 = StringVar()
Symptom2.set("Select Here")

Symptom3 = StringVar()
Symptom3.set("Select Here")

Symptom4 = StringVar()
Symptom4.set("Select Here")

Symptom5 = StringVar()
Symptom5.set("Select Here")
Name = StringVar()
```

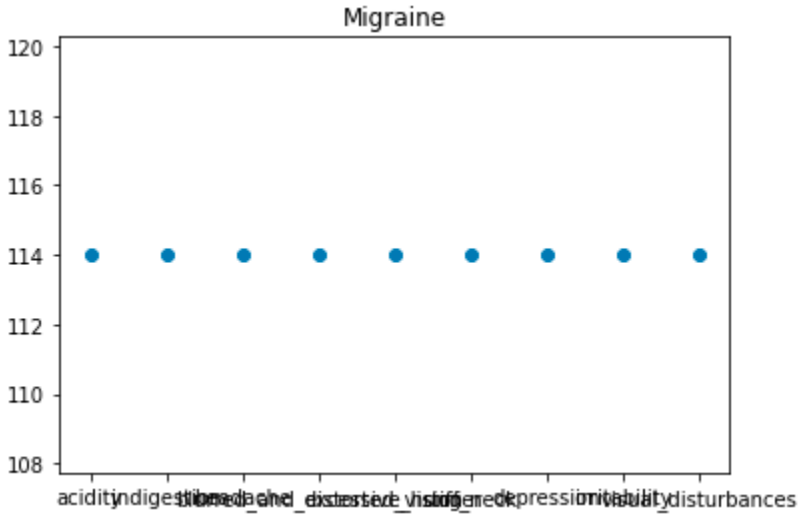
In [29]:
```
prev_win=None
def Reset():
    global prev_win

    Symptom1.set("Select Here")
    Symptom2.set("Select Here")
    Symptom3.set("Select Here")
    Symptom4.set("Select Here")
    Symptom5.set("Select Here")
    NameEn.delete(first=0,last=100)
    pred1.set(" ")
    pred2.set(" ")
    pred3.set(" ")
    pred4.set(" ")
    try:
        prev_win.destroy()
        prev_win=None
    except AttributeError:
        pass
```

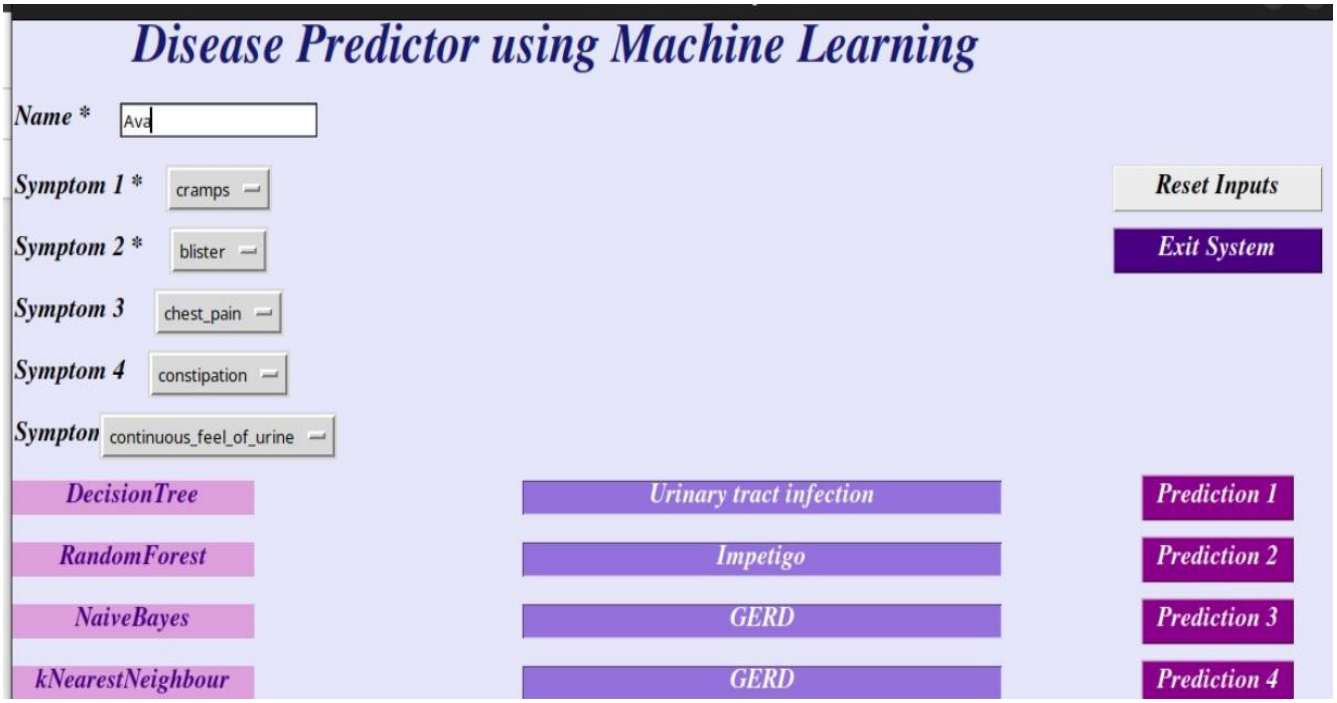
In [30]:
```
from tkinter import messagebox
def Exit():
    qExit=messagebox.askyesno("System","Do you want to exit the system")

    if qExit:
        root.destroy()
        exit()
```

In [31]:
```
#Headings for the GUI written at the top of GUI
w2 = Label(root, justify=LEFT, text="Disease Predictor using Machine Learning", fg="midnightblue", bg="lavender")
w2.config(font=("Times",30,"bold italic"))
w2.grid(row=1, column=0, columnspan=2, padx=100)
#w2 = Label(root, justify=LEFT, text="Contributors:Shre", fg="Pink", bg="lavender")
w2.config(font=("Times",30,"bold italic"))
w2.grid(row=2, column=0, columnspan=2, padx=100)
```

In [32]:
```
#Label for the name
NameLb = Label(root, text="Name *", fg="Black", bg="lavender")
NameLb.config(font=("Times",15,"bold italic"))
NameLb.grid(row=6, column=0, pady=15, sticky=W)
```

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114 114 114 114]
9
9
```




Disease Predictor using Machine Learning
Name *
Ava
Symptom 1 *
cramps
Symptom 2 *
blister
Symptom 3
chest_pain
Symptom 4
constipation
Sympton
continuous_feel_of_urine
Reset Inputs
Exit System
DecisionTree
Urinary tract infection
Prediction 1
RandomForest
Impetigo
Prediction 2
NaiveBayes
GERD
Prediction 3
kNearestNeighbour
GERD
Prediction 4

CONCLUSION

In this assignment, we successfully developed a machine learning-based disease prediction system that can identify possible diseases based on user-input symptoms. By leveraging data preprocessing techniques and testing multiple algorithms—including Decision Tree, Random Forest, SVM, and Naive Bayes—we achieved a high accuracy range of 92% to 95%. The system also includes an interactive interface, making it user-friendly and accessible to both medical professionals and general users. Overall, our model demonstrates how artificial intelligence can support early diagnosis and contribute to smarter, faster, and more scalable healthcare solutions.

Future Work:

There are several potential directions for future development:

Integration with Electronic Health Records (EHRs): To personalize predictions based on patient history and demographics.

Natural Language Processing (NLP): To allow users to describe symptoms in plain text instead of selecting from a list.

Larger and More Diverse Datasets: Including data for rare diseases, regional illnesses, and more varied populations for broader applicability.

Real-Time Monitoring: Linking with wearable devices for live symptom tracking and proactive health alerts.

Cloud Deployment: Making the system globally accessible via cloud platforms for remote diagnosis and telemedicine support.