International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Diseases Predictor using Machine Learning

1. PARDEEP SINGH B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310
pardeepsingh231098@gmail.com

2. ANURAG SINGH B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310
chaudharyanurag70@gmail.com

3. ARUN SHARMA B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310
arunsharma1932003@gmail.com

4. ASHISH PAL B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310
ashishpal28022001@gmail.com

LAVEENA SEHGAL (Mentor) ,
laveena_gn@iimtindia.net

Abstract

For disease management and treatment to be successful in the healthcare industry, prompt and correct diagnosis is crucial. The potential for creating intelligent disease prediction systems has greatly increased with the use of machine learning techniques. This study describes Project Ailment Analysis, a Python-based disease prediction program that makes use of the Naive Bayes, Decision Tree, and Random Forest machine learning methods. This project's primary objective is to identify the most likely illness from the patient's records and symptoms. The system seeks to optimize the diagnosis process and assist medical practitioners in making well-informed decisions by leveraging the power of these algorithms. The article discusses the rationale behind the project, presents a literature review of related work, describes the objectives, the proposed approach, and the implemented algorithms. Furthermore, the methodology, implementation details and future scope of the project are discussed and concluded with a summary of the research findings.

Introduction

The process of diagnosing an illness is intricate and necessitates a high level of medical expertise. However, there is a rising chance to create intelligent systems that can help healthcare providers with the diagnosis process thanks to the availability of patient data and advancements in machine learning algorithms. Disease prediction software called Project Ailment Analysis uses machine learning to analyse patient symptoms and deliver the most likely diagnosis.
Three popular machine learning algorithms are used by the software: Random Forest, Decision Tree, and Naive Bayes. These algorithms are especially well-suited for jobs involving disease prediction because of their reputation for being reliable and effective when handling classification challenges.

Motivation

The main goal of the project's illness analysis is to solve the difficulties that medical practitioners encounter while making correct diagnoses. It can take a while and be prone to human error to analyze a patient's medical history, history of symptoms, and other data as part of the diagnostic procedure. Accelerating the diagnostic process, lowering the possibility of a misdiagnosis, and eventually improving patient outcomes are the objectives of creating an intelligent disease prediction system. Furthermore, the opportunity to employ machine learning techniques to develop prediction models is made possible by the availability of vast datasets of patient symptoms and diseases. The software can learn patterns and links between symptoms and diseases by training on these data sets, which enables it to make precise predictions.

Literature Survey

According to Tom Mitchell, who is discussing artificial intelligence, "a PC program is said to gain as a matter of fact and from certain undertakings and some presentation on."

assessed by and improves with repetition. Since artificial intelligence is essentially a sophisticated network of links, most of its computations are centered around finding and potentially using connections between various datasets. The model can either use these connections to anticipate future perceptions or aggregate the data to identify intriguing situations when Machine Learning Algorithms are able to focus on specific relationships. Machine learning makes use of a wide range of calculations, such as logistic regression, regression, Naive Bayes classifier, Bayes speculation, KNN (K-Nearest Neighbor classifier), Decision Tress, entropy, ID3, SVM (Support Vector Machines), K-infers algorithm, Random Forest, etc.

In 1959, Arthur Samuel coined the term AI. AI investigates the assessment and development of calculations that can profit from and establish expectations based on data. Computational measurements, which are also centered on PC forecasting, are strongly associated with (and frequently comprise) machine learning. It has a close relationship to numerical advancement, which gives the field its ideas, methodologies, and fields of application. Information mining, also known as solo learning and a subset of artificial intelligence that concentrates primarily on exploratory information discovery, is sometimes mistaken for artificial intelligence.

Artificial intelligence (AI) is a technology that makes complex models and computations that are amenable to prediction in the evaluation of information;
In the business sector, this is known as foresight investigation. These scientific models allow specialists, information researchers, designers, and examiners to "produce solid, repeatable choices and results" and uncover "stowed away experiences" by using the material's verified linkages and patterns as a source of learning.

AI initiatives Machine learning tasks are generally classified into numerous major categories:
Managed instruction: To the PC is delivered model data.

sources and their desired results, given by a "teacher," with the goal of mastering an overarching principle that governs yield contributions. The information indication might only be partially shown or restricted to a single critique in some extremely uncommon situations.
Participants were therefore given the chance to submit their baseline data from this study, which is compared, and the data and the heart state are evaluated. The objective of creating a classifier framework with AI computations is to greatly aid in the resolution of health-related issues by empowering medical practitioners to anticipate and assess illnesses at an early stage. Examining

a sample data set of 4920 that wasn't precisely restricted to 41 illnesses was decided upon. A dependent variable was generated from 41 disorders. Ninety-five of the 132 independent variables (symptoms) that showed a substantial correlation with infections were chosen and investigated. The completed analysis work presents the infection expectation framework, which was created with the use of machine learning algorithms like Random Woods and Naïve Bayes, and Decision Tree classifiers.

Objective

The primary aim of the disease analysis project is to create software for disease prediction that can precisely identify potential illnesses based on symptoms and data submitted by the patient. The particular goals are as follows:

1. Use three learning algorithms for disease prediction tasks: Random Forest, Decision Tree, and Naive Bayes. Evaluate and contrast their results.

2. Provide a user-friendly interface where users can input symptoms and associated data for patients.

3. Determine the likelihood of each condition by ranking them according to the algorithm's predictions and probability.

4. Using the proper indicators and methodologies, assess the produced system's correctness and performance.

5. Determine possible directions for the future development of the illness prediction software.

Proposed Approach

By combining the Naive Bayes, Decision Tree, and Random Forest machine learning algorithms, Project Disease Analysis offers a thorough method for predicting diseases. The following actions are part of the suggested strategy:

1. Gathering and preparing data: Gather pertinent patient data, such as medical history, symptoms, and other pertinent data. Handle missing values, eliminate outliers, and encode categorical variables as part of the preprocessing step.

2. Choosing features: Determine which features—symptoms and patient characteristics—are the most useful and have the biggest impact on the illness prediction task. Methods like domain knowledge, recursive feature elimination, and correlation analysis can be used to complete this stage.

3. Model Training: Train the Naive Bayes, Decision Tree and Random Forest algorithms on the pre-processed dataset, using the selected features as input and the corresponding disease labels as variables in the target.

4. Model evaluation: Assess trained models' performance with suitable measures including F1 score, accuracy, precision, and recall. For models that are reliable and generalizable, employ strategies like retention testing and cross-validation.

5. Model selection: Choose which model, or models, perform the best and include them into the illness prediction software based on the evaluation findings.

6. Software development: Provide an intuitive user interface that enables users to input patient symptoms and relevant data. Utilizing user input, incorporate the selected model or models into software to forecast illnesses.

7. Prediction and classification: Apply the

trained model(s) to predict which diseases are most likely to manifest based on patient data and user-submitted symptoms. Arrange the predicted illnesses based on the probabilities or confidence levels attached to each.

8. Interpretation and Visualization: Offers comprehensible and educational forecast results visualizations so users can comprehend the reasoning behind forecasts and come to well-informed judgments.

Algorithms Implemented

Naive Bayes

Based on the Bayes theorem, the probabilistic machine learning method Naive Bayes presupposes independence between qualities (symptoms) within a particular class (illness). based on the collection of recorded symptoms, determines the probability of each disease and makes a prognosis about the condition with the highest probability.

Decision Tree

A non-parametric supervised learning approach called decision trees builds a tree model of decisions and their potential results. Using an iterative process, the algorithm creates branches that represent various solutions or outcomes by partitioning the data according to the most informative attributes. Using the input features as a guide, the tree is traversed to make the final prediction.

Random Forest

A set of learning techniques called Random Forest combines several decision trees to improve prediction accuracy and decrease redundancy. Make a collection of decision trees, each with a random subset of the characteristics and trained data as its foundation. The data from each tree is summarized in a final report, typically using averages or statistics.

Methodology

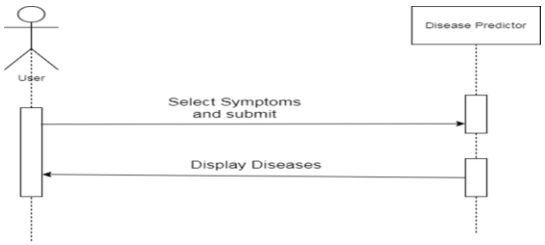


Fig 1- working methodology

The main diagnostic criteria are as follows.

1. Data collection: Compile a thorough set of information on the patient, including their medical history, symptoms, and pertinent illness labels. It should be mentioned that the datasets reflect various diseases and differ from one another.

2. Data Processing: Handle missing values, eliminate anomalies, and code category variables in order to process the gathered data. Ensuring the quality and consistency of the data utilized to train the machine learning model is a crucial step that must be completed.

3. Feature selection: Determine which features—symptoms and patient traits—offer the most valuable insights into the prognosis of the disease. Techniques like correlation analysis, repeated feature extraction, and industry experts' knowledge can be applied to this.

4. Segment the dataset: Prior to processing, split the dataset in training and test halves. The set of tests is used to assess performance, and the training set are used to train models based on machine learning.

5. Training models: Using specific characteristics as input and loss points as targeted variables, training datasets are utilized to train naïve Bayesian, decision tree, and randomization algorithms.

6. Parameter optimization: Increase forecast accuracy by fine-tuning each algorithm's parameters. Techniques like random and grid searches can be used for this.

7. Model Evaluation: Assess the trained models' performance using relevant measures, such as F1 score, accuracy, precision, and recall. To make sure the sample is robust and generalizable, employ techniques like retention tests and cross-validation.

8. Model selection: Choose the best model or group of models to incorporate into the illness prediction software following the evaluation results.

9. Software Development: Create an intuitive user interface that enables users to input patient symptoms and associated information. Put the chosen model into the program, and it will use user input to predict diseases.

10. Test and Usability: Examining the generated program for accuracy, performance, and usability. Distribute the program to healthcare professionals and other interested individuals for optimal outcomes.

Implementation

Conducting a project analysis includes the following key elements:

1. Data entry: Create a form that will be used to load and initialize patient data sets. This entails grouping the data set for testing and training, coding variables based on categories, and managing missing values.

2. Use feature engineering approaches to find the most useful characteristics for disease prediction by applying feature selection strategies. Correlation analysis, covariate removal, or the expertise of industry specialists can all be used to accomplish this.

3. Training models. To teach decision trees, random forests, and naive Bayes algorithms, create your own modules. Preparing data, training models, and optimizing hyperparameters should all be covered in this subject.

4. Model assessment. Assess the trained model's performance using assessment metrics including F1 score, recall, accuracy, and precision. Make use of methods like robustness testing and cross-validation to guarantee generalizability and dependability.

5. Model selection: Choose the best model or combination of models to utilize in the illness prediction program based on the analysis's results.

6. User Interface: Create a user-friendly command line or graphical interface that enables users to input information about patients, including their characteristics. It should be simple to use and intuitive to traverse this UI.

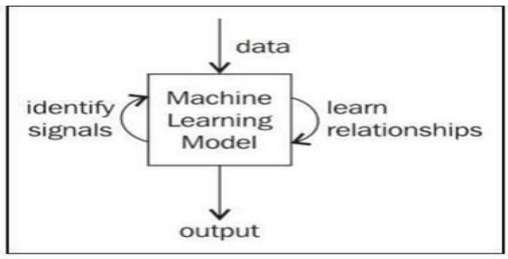
7. Illness prediction. Utilizing user data, integrate specific models in software to generate disease forecasts. Utilize tools for classification and visualization to show the possibility or safety of projected diseases.

8. Putting it to use. Use a suitable platform (web, desktop, cloud) to deploy illness prediction software so that healthcare providers and other stakeholders can easily access and utilize it.

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

Fig 2 - The given dataset is divided into two parts namely feature matrix and response vector.

The application is written in Python and makes use of well-known machine learning tools like Panda, NumPy, and scikit-learn. Furthermore, data visualization and interpretation are possible with libraries like Seaborn and Matplotlib.



An overview of machine learning models

Model Selection

Future Scope

The project diagnosis presents various avenues for enhancement and growth:

1. Make additional algorithms available. Investigate whether you can increase prediction accuracy and dependability by including additional machine learning algorithms, such as neural networks, support vector machines, or ensemble techniques like gradient boosting.

2. Integration of data across multiple modes. Expand the software's functionality to incorporate multimodal data sources, including genetic, laboratory, and medical pictures, to boost predictive capacity and offer a more complete illness prediction solution.

3. Give an explanation of AI. By using explainable AI techniques, you may make predictions that are clear and easy to understand. This will boost healthcare professionals' confidence in the system by enabling them to comprehend the logic behind the forecasts.

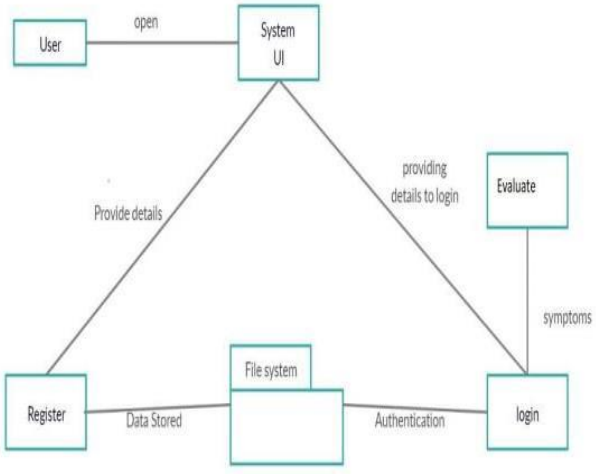
4. Lifelong learning. Provide techniques for the model to be updated and learned continuously so that when fresh patient data becomes available, the system can adjust and improve its predictions.

5. Individual forecasts. Examine the potential for implementing customized treatment plans that consider the patient's distinct traits, medical background, and genetic variables in order to provide a more accurate and customized prognosis.

6. Connectivity to the electronic health record (EMR). creating interfaces between illness prediction software and the current EHR systems in order to facilitate software implementation in healthcare facilities and guarantee smooth data interchange.

7. Combining wearables and mobile devices. Expand the software's compatibility to include wearables and mobile devices to allow for real-time disease prediction and symptom monitoring, facilitating early detection and preventative actions.

Result



Conclusion

A software for predicting diseases, Software Diagnosis makes use of machine learning methods including Random Forest, Decision Tree, and Naive Bayes. The software attempts to produce precise and trustworthy disease predictions by merging various apps, taking into account pertinent data and the symptoms reported by the patients.
In addition to outlining the project's justification, the study article lists potential advantages of mental disease prediction methods as well as diagnostic obstacles. The paper highlights gaps in the detection of porcine disease and provides an overview of recent studies conducted in the subject through a review of the literature.

The suggested methodology covers the following steps: gathering data, preprocessing, choosing a product, studying models, analyzing data, and developing software. Essential aspects including data collection, modeling training and assessment, architecture, user data production, and prediction are included in program components.

The future plans for the project are also outlined in the paper, along with potential enhancements that could be made, like incorporating more algorithms, combining datasets, enhancing artificial intelligence comprehension, learning to make more customized predictions, integrating with electronic health records, and supporting wearables and mobile devices.

Software Diagnostic strives to simplify the diagnostic process, assist healthcare providers in making better decisions, and enhance patient outcomes by providing user-friendly machine learning and diagnosis software. Ongoing efforts to leverage technology to improve health care outcomes are supported by research findings and suggested applications.

References

[1] Disease Prediction and Doctor Recommendation System by www.irjet.net

[2] Disease Prediction Based on Prior Knowledge by www.hcupus.ahrq.gov/nisoverview.jsp

[3] GDPS - General Disease Prediction System by www.irjet.net

[4] Kaveeshwar, S.A., and Cornwall, J., 2014, "The current state of disease mellitus in India". AMJ, 7(1), pp. 45-48.

[5] Dean, L., McEntyre, J., 2004, "The Genetic Landscape of Disease [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); Chapter 1, Introduction to Disease. 2004 Jul 7.

[6] Machine Learning Methods Used in Disease by www.wikipedia.com https://www.researchgate.net/publication/32116774_disease_prediction_using_machine_learning_techniques

[7] Disease Prediction Using Machine Learning by International Research Journal of Engineering and Technology (IRJET).

[8] Kaveeshwar, S.A., and Cornwall, J., 2014, "The current state of disease mellitus in India". AMJ, 7(1), pp. 45-48.

[9] Dean, L., McEntyre, J., 2004, "The Genetic Landscape of Disease [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); Chapter 1, Introduction to Disease. 2004 Jul 7.

[10] Machine Learning Methods Used in Disease by www.wikipedia.com

[11] https://www.researchgate.net/publication/3251 16774_disease_prediction_using_machine_learning_techniques

[12] https://ieeexplore.ieee.org/document/8819782/ disease_prediction

[13] Algorithms Details from www.dataspirant.com

[14] https://www.youtube.com/disease_prediction

[15] https://www.slideshare.com/disease_prediction

[16] https:/en.wikipedia.org/machine_learning_algorithms

[17] https://en.wikipedia.org/wiki/Python_(programming_language)

[18] https://wiki.python.org/TkInter

[19] https://creately.com/lp/uml-diagram-tool/

[20] https://app.diagrams.net/

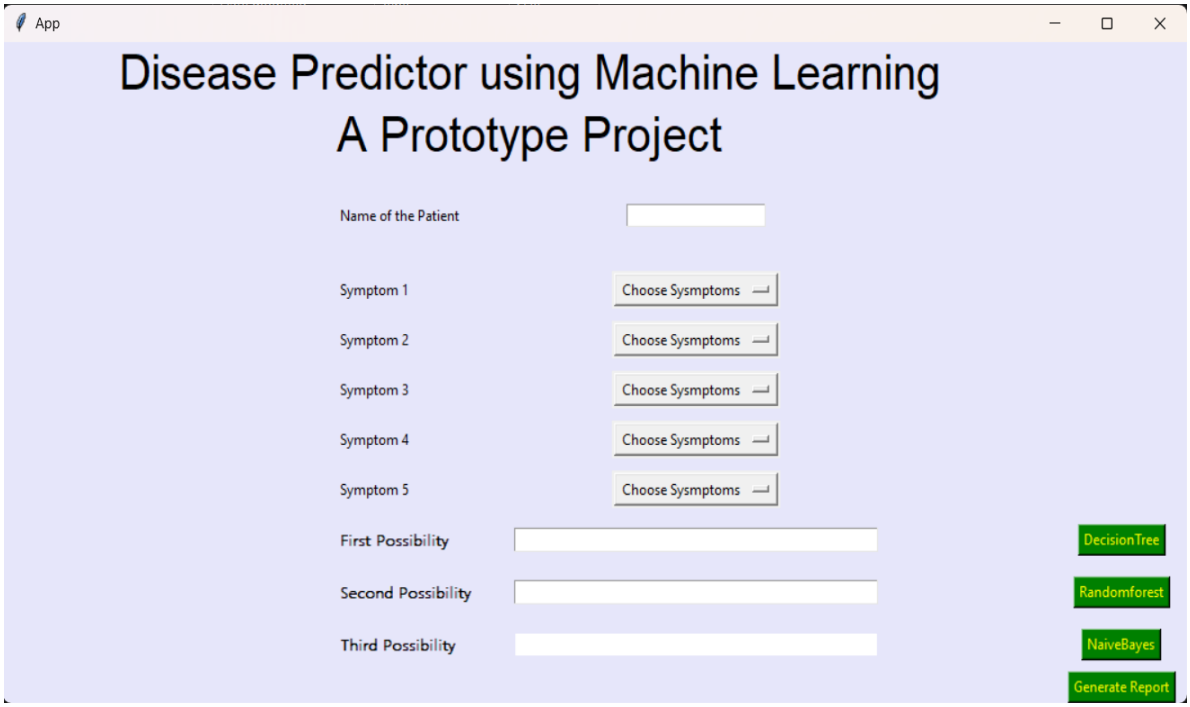
Coding :

```
from tkinter import *
import numpy as np
import pandas as pd
# from gui_stuff import *

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','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','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']

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)',
'Heartattack','Varicoseveins','Hypothyroidism','Hyperthyroidism','Hypoglycemia','Osteoarthristis',
'Arthritis','(vertigo) Paroymsal  Positional Vertigo','Acne','Urinary tract infection','Psoriasis',
'Impetigo']

l2=[]
for x in range(0,len(l1)):
    l2.append(0)

# TESTING DATA df -------------------------------------------------------------------------------
df=pd.read_csv("Training.csv")

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':19,
'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,'Hypothyroidism':31,
'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)

# print(df.head())

X= df[l1]

y = df[["prognosis"]]
np.ravel(y)
# print(y)

# TRAINING DATA tr ----------------------------------------------------------------------------------
tr=pd.read_csv("Testing.csv")
tr.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':19,
'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,'Hypothyroidism':31,
'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)

X_test= tr[l1]
y_test = tr[["prognosis"]]
np.ravel(y_test)
# ----------------------------------------------------------------------------------------------------

def DecisionTree():

    from sklearn import tree

    clf3 = tree.DecisionTreeClassifier()   # empty model of the decision tree
    clf3 = clf3.fit(X,y)

    # calculating accuracy-------------------------------------------------------------------
    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))
    # ---------------------------------------------------

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

    for k in range(0,len(l1)):
        # print (k,)
        for z in psymptoms:
            if(z==l1[k]):
```

```
            l2[k]=1

    inputtest = [l2]
    predict = clf3.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t1.delete("1.0", END)
        t1.insert(END, disease[a])
    else:
        t1.delete("1.0", END)
        t1.insert(END, "Not Found")

def randomforest():
    from sklearn.ensemble import RandomForestClassifier
    clf4 = RandomForestClassifier()
    clf4 = clf4.fit(X,np.ravel(y))

    # calculating accuracy-------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=clf4.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # ---------------------------------------------------

    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 = clf4.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t2.delete("1.0", END)
        t2.insert(END, disease[a])
    else:
```

Output :


App
Disease Predictor using Machine Learning
A Prototype Project
Name of the Patient
Symptom 1
Choose Sysmptoms
Symptom 2
Choose Sysmptoms
Symptom 3
Choose Sysmptoms
Symptom 4
Choose Sysmptoms
Symptom 5
Choose Sysmptoms
First Possibility
Second Possibility
Third Possibility
DecisionTree
Randomforest
NaiveBayes
Generate Report