

MACHINE LEARNING TECHNIQUES (01AI0603)

MLT MINI PROJECT REPORT {Disease Prediction using Symptoms}

SUBMITTED BY:

Divy Dodeja 91900151045

Smit Devani 91900151001

Girish Purohit 91900151058

Aditya Jain 91900151053

GUIDED BY:

Prof. Kathiresan

Problem Statement

Health information needs are also changing the information seeking behaviour and can be observed around the globe. Challenges faced by many people are looking online for health information regarding diseases, diagnoses and different treatments. If a recommendation system can be made for doctors and medicine while using review mining will save a lot of time. In this type of system, the user face problem in understanding the heterogeneous medical vocabulary as the users are laymen. User is confused because a large amount of medical information on different mediums are available. The idea behind recommender system is to adapt to cope with the special requirements of the health domain related with users.

What we did?

In our project we have tried accurately predict a disease by looking at the symptoms of the patient. We have used Naïve Bayes algorithm 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 visualized the result of our study and this project.

Database Collection

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

<https://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html>

Libraries Used

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

1. tkinter: It's a standard GUI library of python. Python when combined with tkinter provides fast and easy way to create GUI. It provides powerful object-oriented tool for creating GUI.

It provides various widgets to create GUI some of the prominent ones being:

- ☐ Button
- ☐ Canvas
- ☐ Label

- ☐ Entry
- ☐ Check Button
- ☐ List box
- ☐ Message
- ☐ Text
- ☐ Messagebox

Some of these were used in this project 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 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. Numpy's main purpose is to deal with multidimensional homogeneous array. It has tools ranging from array creation to its handling. It makes it easier to create a n dimensional array just by using np.zeros() or handle its contents using various other methods such as replace, arrange, random, save, load it also helps in array processing using methods like sum, mean, std, max, min, all, etc. Array created with numpy also behave differently then arrays created normally when they are operated upon using operators such as +, -, *, /. All the above qualities and services offered by numpy array makes it highly suitable for our purpose of handling data. Data manipulation occurring in arrays while performing various operations need to give the desired results while predicting outputs require such high operational capabilities.

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.

Data in python can be analysed with 2 ways

- ☐ Series
- ☐ Dataframes

Series is one dimensional array defined in pandas used to store any data type.

Dataframes are two-dimensional data structure used in python to store data consisting of rows and columns. Pandas dataframe is used extensively in this project to use datasets required for training and testing the algorithms. Dataframes makes it easier to work with attributes and results. Several of its inbuilt functions such as replace were used in our project for data manipulation and preprocessing.

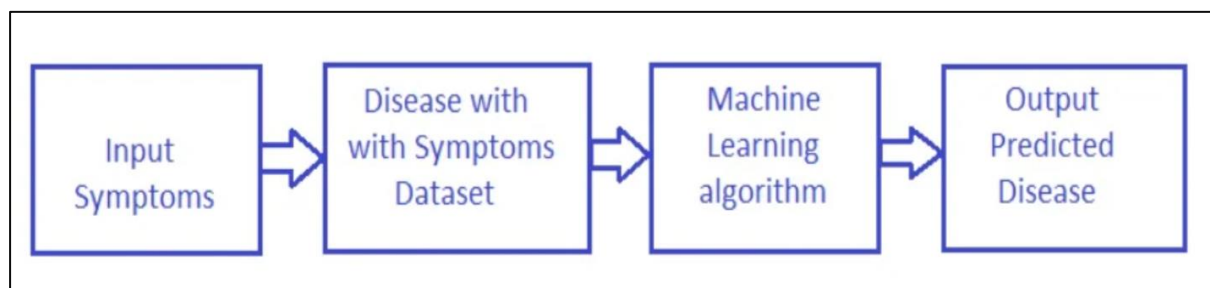
4. sklearn: Sklearn is an open source python library with implements a huge range of machine-learning, pre-processing, cross-validation and visualization algorithms. It features various simple and efficient tools for data mining and data processing. It features various classification, regression and clustering algorithm such as support vector machine, random forest classifier, decision tree, gaussian naïve-Bayes, KNN to name a few. In this project we have used sklearn to get advantage of inbuilt classification algorithms like naïve Bayes. We have also used inbuilt cross validation and visualization features such as classification report, confusion matrix and accuracy score.

Algorithm Used

Naive Bayes algorithm is a family of algorithms based on naïve bayes theorem. They share a common principle that is every pair of prediction is independent of each other. It also makes an assumption that features make an independent and equal contribution to the prediction.

In our project we have used naïve bayes algorithm to gain a ~95% accurate prediction.

Working



Block Diagram for Disease Prediction System using Supervised Learning

Implementation Steps

- Import all the packages required i.e. tkinter for GUI, numpy to perform numerical operations and pandas for reading the csv files.

```
from tkinter import *  
from tkinter import messagebox  
import numpy as np  
import pandas as pd
```

- Create a list which contains all the symptoms which are according the csv file.

```
#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',
```

- Create another list which contain the diseases.

```
#List of Diseases is listed in list disease.
disease=['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
        'Drug Reaction', 'Peptic ulcer disease', 'AIDS', 'Diabetes ',
        'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migraine',
        'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
        'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
```

- Then, create a empty list.

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

L1 and L2, both have equal length.

L1	Sym1	sym2	sym3	sym4	sym5	sym6	sym7	sym8
L2	0	0	0	0	0	0	0	0

- Perform same steps for both testing and training dataset
 1. Using pandas read the CSV file
 2. Replace with index

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

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,
    'Peptic ulcer disease':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asthma':9,
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,
```

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

```
#Reading the testing.csv file
tr=pd.read_csv("testing.csv")

#Using inbuilt function replace in pandas for replacing the values

tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic  
'Peptic ulcer disease':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bron  
'Migraine':11,'Cervical spondylosis':12,
```

```
X_test= tr[l1]
y_test = tr[["prognosis"]]
np.ravel(y_test)
print(X_test)
```

S1	S2	S3	S4	S5	S6	S7	S8	Prognosis
0	0	0	1	0	1	0	0	0
0	1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	2
0	0	0	0	0	0	0	0	3

3. Make X as symptoms and Y as disease.

- GaussianNB() is used to train the model and predict the disease on testing dataset according to symptoms entered by the user. Accuracy of predicting the disease is calculated using accuracy_score and confusion matrix is created using confusion_matrix which are imported from sklearn.metrics.

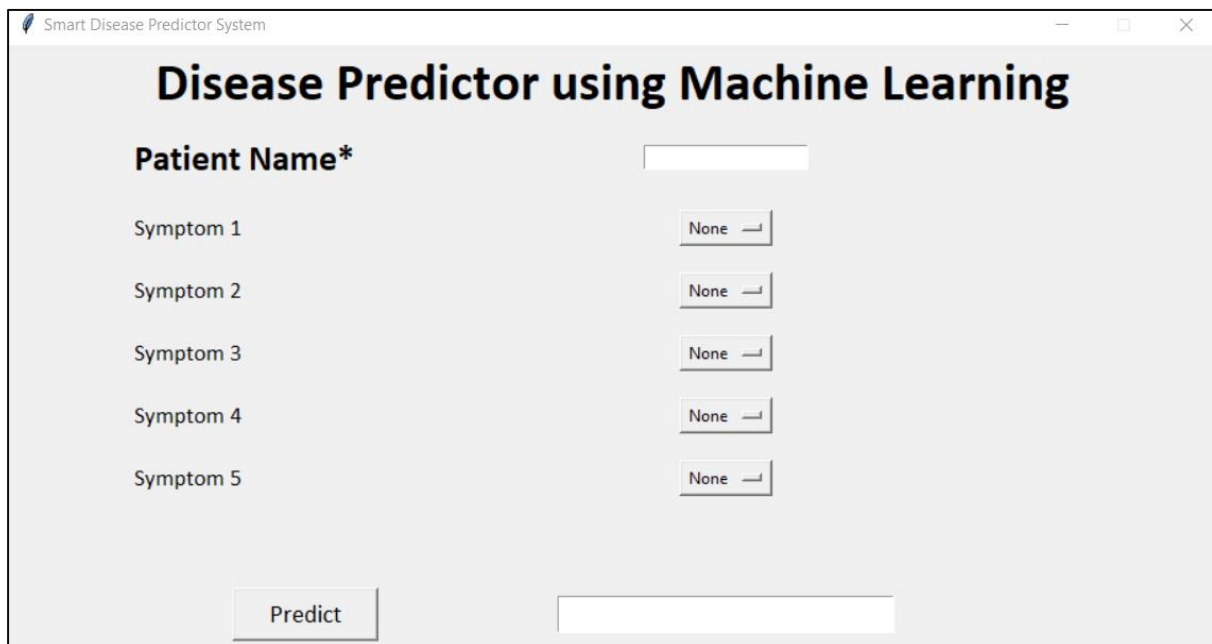
```
def NaiveBayes():
    from sklearn.naive_bayes import GaussianNB
    gnb = GaussianNB()
    gnb=gnb.fit(X,np.ravel(y))
    from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
    y_pred=gnb.predict(X_test)
    print("Accuracy:",end=" ")
    print(accuracy_score(y_test, y_pred))
    print("Number of correctly classified saamples:",end=" ")
    print(accuracy_score(y_test, y_pred,normalize=False))
    print("Confusion matrix:")
    conf_matrix=confusion_matrix(y_test,y_pred)
    print(conf_matrix)
```

- Building the Graphical User Interface:
 - Graphical User Interface is build using tkinter library in Python. Root is used to start the GUI. GUI titlt is given as “Smart Disease Predictor System” using title() function in tkinter library. Resizable function is used to fix the size GUI.
 - Here, variables are defined like Name, Symptom1, Symptom2, etc and they initialised to “None” using set() function in tkinter library.
 - “NameLb1” is the label created for showing the “Name of the Patient *” using label() function in tkinter library. It is configured using config() function and the grid of the label is set using grid() function.
 - Similarly labels for showing the symptoms of the disease are also created.

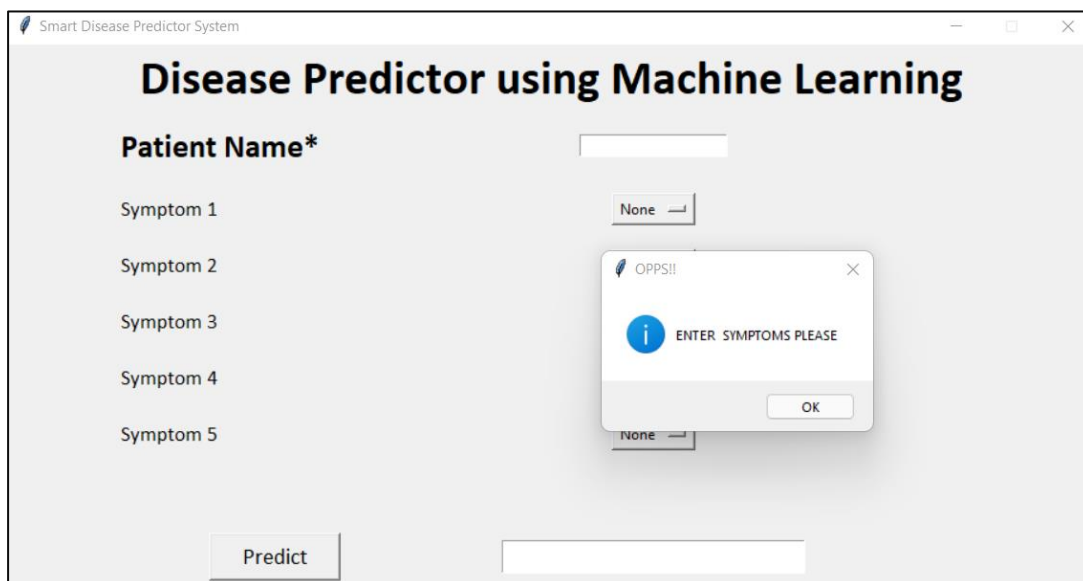
5. NameEn is the entry box created for getting the name of the patient using Entry() function in tkinter library. S1, S2, S3, S4, S5 are the option menu used to get symptoms from the user which is created using Optionmenu in tkinter library. *OPTIONS is the list of unique symptoms.
6. Button is also created for prediction.

Results

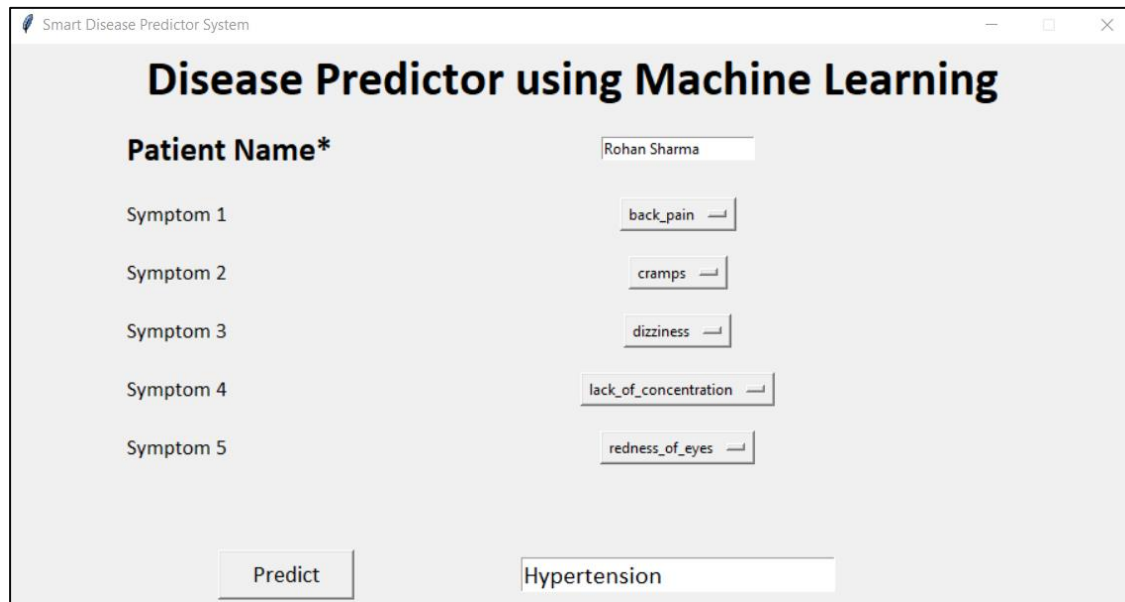
GUI made for this project is a simple tkinter GUI consisting of labels, messagebox, button, text, title and option menu



Messagebox are used to ask for at least two symptoms



After entering patient's name and selecting the symptoms the disease has been predicted



Smart Disease Predictor System

Disease Predictor using Machine Learning

Patient Name* Rohan Sharma

Symptom 1 back_pain

Symptom 2 cramps

Symptom 3 dizziness

Symptom 4 lack_of_concentration

Symptom 5 redness_of_eyes

Predict Hypertension

Accuracy Score and confusion matrix of the model used.

```
Accuracy: 0.9512195121951219
Number of correctly classified saamples: 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]]
```

Conclusion

We set out to create a system which can predict disease on the basis of symptoms given to it. Such a system can decrease the rush at OPDs of hospitals and reduce the workload on medical staff. We were successful in creating such a system to do so. On an average we achieved accuracy of ~94%. Such a system can be largely reliable to do the job. Our system also has an easy to use interface.