Disease Prediction Using ML and DL

Project Report

By

Rajnish Singh [CSE/19061/488] , Akash Singh [CSE/19007/434] , Harshit Sharma [CSE/19033/460]

A report submitted to

Indian Institute Of Information Technology Kalyani (W.B)

for 5th semester project evaluation

Bachelor of Technology

in

Computer Science and Engineering

November, 2021

Certificate

This is to certify that the project report entitled “Disease Prediction System" being submitted by **Rajnish Singh**, an undergraduate student **(Roll No: CSE/19061/488)**, **Akash Kumar Singh**, an undergraduate student **(Roll No: CSE/19007/434)** and **Harshit Sharma**, an undergraduate student **(Roll No: CSE/19033/460)** in the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, (W.B.) India, in partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering, is an original research work carried by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulation of IIIT Kalyani and in my opinion, has reached the standards needed for submission. The works, techniques and the results presented have not been submitted to any other university or institute for the award of any other degree or diploma.

(Dr. Debasish Bera)
Assistant Professor
Department of Computer Science and Engineering Indian Institute of Information Technology Kalyani
Kalyani, W.B.-741235, India.

Acknowledgments

We would like to take this opportunity to thank our mentor **Dr. Debasish Bera** for guiding us throughout this project and motivating us to perform our best. He was always there to clear our doubts and provided us with new ideas and resources.

Rajnish Singh
Roll No.: CSE/19061/488
Department of Computer Science and Engineering Indian Institute of Information Technology Kalyani Kalyani, W.B. - 741235, India.

Akash Kumar Singh
Roll No.: CSE/19007/434
Department of Computer Science and Engineering Indian Institute of Information Technology Kalyani Kalyani, W.B. - 741235, India.

Harshit Sharma
Roll No.: CSE/19033/460
Department of Computer Science and Engineering Indian Institute of Information Technology Kalyani Kalyani, W.B. - 741235, India.

Contents

Introduction

A diagnosis procedure usually starts with the patient complaints and the doctor learns more about the patient situation interactively during an interview, as well as by measuring some metrics such as blood pressure or body temperature. The diagnosis is then determined by taking the whole available patients status into account. Then depending on that, a suitable treatment is prescribed, and the whole process might be iterated. In each iteration, the diagnosis might be reconfigured, refined, or even rejected. Epidemics also should be considered and the genetic factors too, and then a diagnosis can be materialized. However, there are still some problems. The work presents an intelligent and automatic disease prediction system using some probabilistic model, i.e. **Artificial Neural Network (ANN), Kth Nearest Neighbour (KNN) and Decision Tree algorithm**. It uses the data set of large number of confirmed cases relationship between symptoms of the patient and make prediction more accurate and the system practically more useful as every disease diagnosis is other than symptoms is based on many others factors so it covers many of them. The diagnosis we done on the basis of comparing the symptoms given by the user and comparing it with the confirmed cases data set giving the most probable disease and the condition of Severness of symptoms.

Pre-Work

We developed a primary diagnosis system that helps people in isolated environments (without proper medical staff) to predict diseases according to their symptoms and other parameters and to select appropriate drugs for particular disese. The system classifies patients' abilities to protect themselves from diseases and ailments through reviews and ratings of the drugs beforehand. In an evaluation of the prototype system, the risk level it determined correlated with the decisions of specialists as well as the reviews sent by practitioners of the drugs in question.

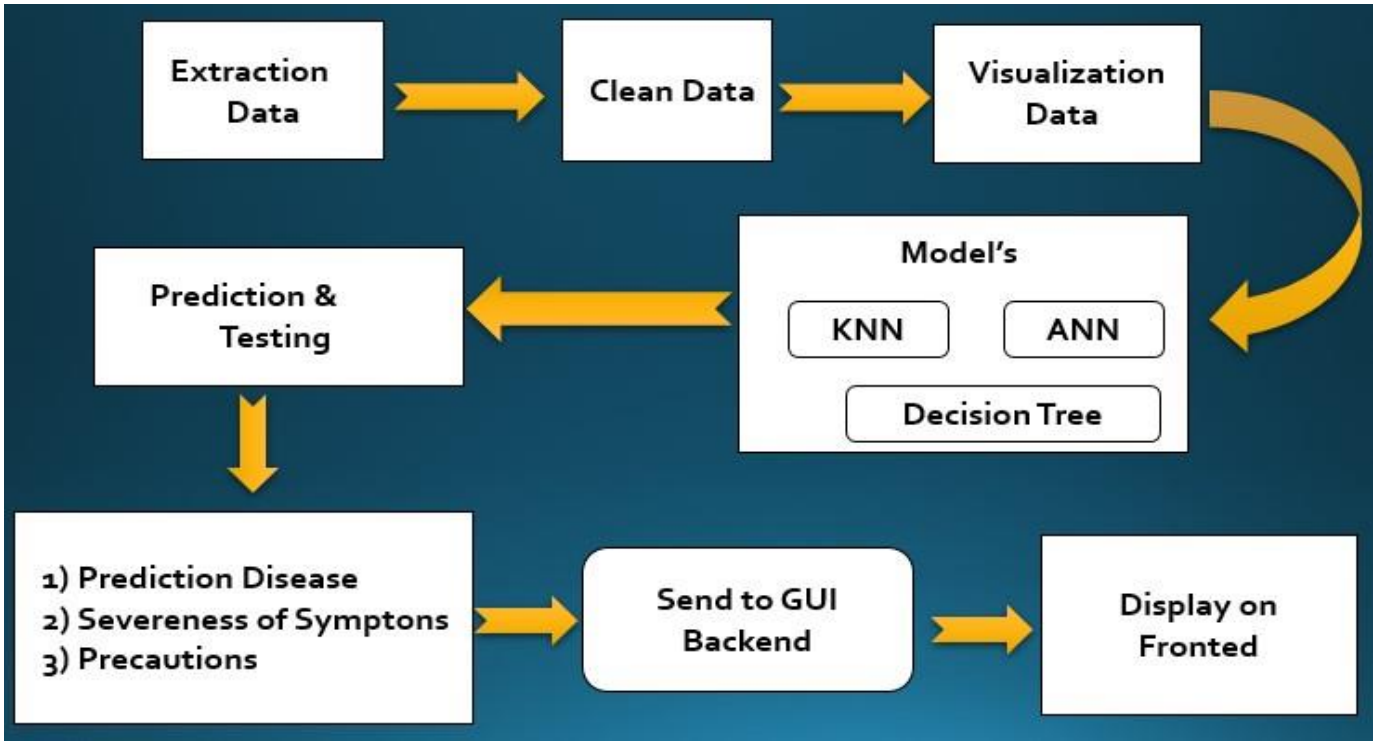
Project Goal

Our goal is to develop a Web-App using Flask enclosed, ML trainable ,primary diagnosis system that can be used to predict diseases and recommending drugs according to the predicted disease in rural areas or isolated areas where medical facilities are not so good .It will use the data set of large number of confirmed cases and it will find relationship between symptoms of the patient to make prediction more accurate and the system practically more useful and to make it more user friendly by developing a GUI and merging it with drug recommendation system.

Methodology

- Extraction of data.
- Cleaning the dataset.
- Data Visulaization.
- Training Model
- Analysis of the manual data.
- Trying Classifier to learn disease from the symptoms.
- Finding the feature importance like severeness of Symptoms.
- Getting the Precaution and Description of Diseases.
- Integrating the GUI for better understanding and for better experience and ease to use.

Flow Chart

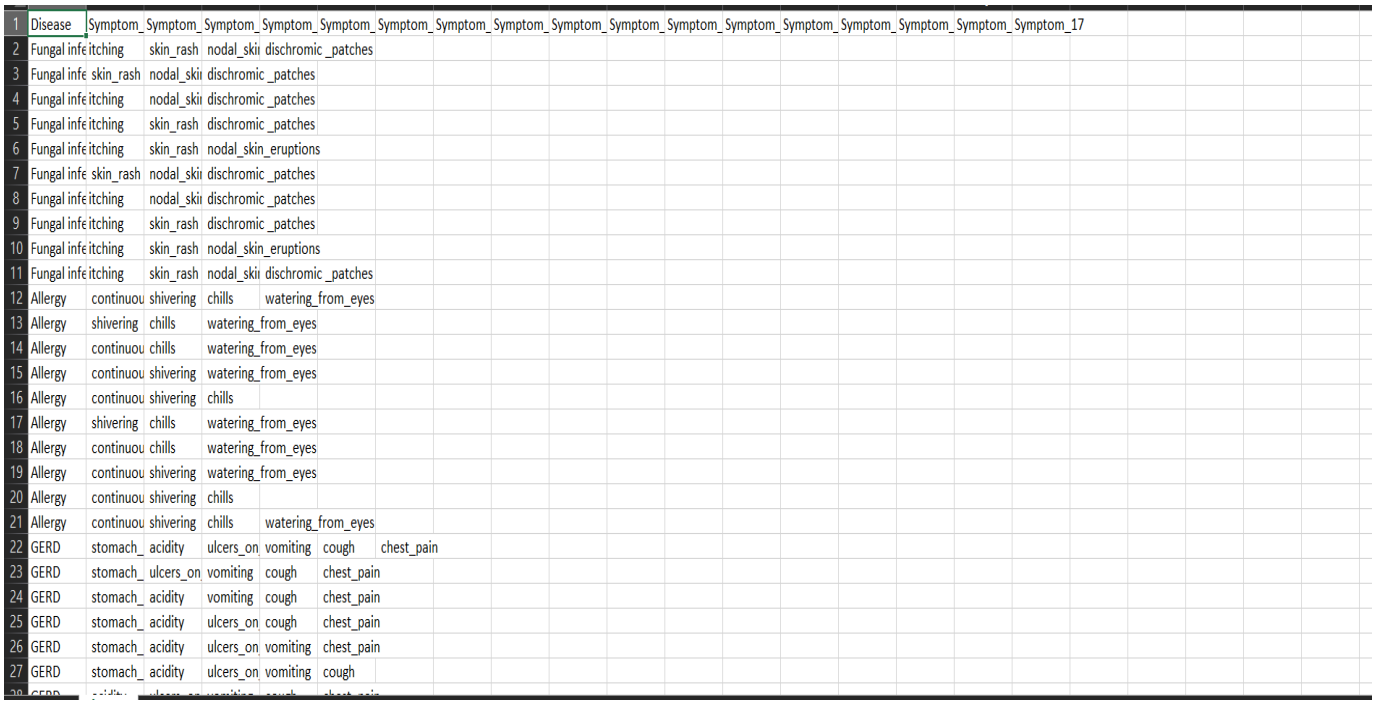

Extraction Data
Clean Data
Visualization Data
Model's
KNN
ANN
Decision Tree
Prediction & Testing
1) Prediction Disease
2) Severeness of Symptons
3) Precautions
Send to GUI Backend
Display on Fronted

Data Collection

The data set used in the project was taken from two different sources:

1. We extracted the data from Kaggle.com:
   https://www.kaggle.com/kaushil268/disease-prediction-using-machine-learning

2. After data extraction we need to clean the data so we removed the punctuation marks and unnecessary terms to scrap the data to get the symptoms and disease associated with them.



Disease	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_	Symptom_17
Fungal infe	itching	skin_rash	nodal_skii	dischromic	_patches														
Fungal infe	skin_rash	nodal_skii	dischromic	_patches															
Fungal infe	itching	nodal_skii	dischromic	_patches															
Fungal infe	itching	skin_rash	dischromic	_patches															
Fungal infe	itching	skin_rash	nodal_skin_eruptions																
Fungal infe	skin_rash	nodal_skii	dischromic	_patches															
Fungal infe	itching	nodal_skii	dischromic	_patches															
Fungal infe	itching	skin_rash	dischromic	_patches															
Fungal infe	itching	skin_rash	nodal_skin_eruptions																
Fungal infe	itching	skin_rash	nodal_skii	dischromic	_patches														
Allergy	continuou	shivering	chills	watering_from_eyes															
Allergy	shivering	chills	watering_from_eyes																
Allergy	continuou	chills	watering_from_eyes																
Allergy	continuou	shivering	watering_from_eyes																
Allergy	continuou	shivering	chills																
Allergy	shivering	chills	watering_from_eyes																
Allergy	continuou	chills	watering_from_eyes																
Allergy	continuou	shivering	watering_from_eyes																
Allergy	continuou	shivering	chills																
Allergy	continuou	shivering	chills	watering_from_eyes															
GERD	stomach_	acidity	ulcers_on	vomiting	cough	chest_pain													
GERD	stomach_	ulcers_on	vomiting	cough	chest_pain														
GERD	stomach_	acidity	vomiting	cough	chest_pain														
GERD	stomach_	acidity	ulcers_on	cough	chest_pain														
GERD	stomach_	acidity	ulcers_on	vomiting	chest_pain														
GERD	stomach_	acidity	ulcers_on	vomiting	cough														

3. Beside this datasets there was another dataset provided of severness of symptom.

Symptom	weight
itching	1
skin_rash	3
nodal_skin_eruptions	4
continuous_sneezing	4
shivering	5
chills	3
joint_pain	3
stomach_pain	5
acidity	3
ulcers_on_tongue	4
muscle_wasting	3
vomiting	5
burning_micturition	6
spotting_urination	6
fatigue	4
weight_gain	3
anxiety	4
cold_hands_and_feets	5
mood_swings	3
weight_loss	3
restlessness	5
lethargy	2
patches_in_throat	6
irregular_sugar_level	5
cough	4
high_fever	7

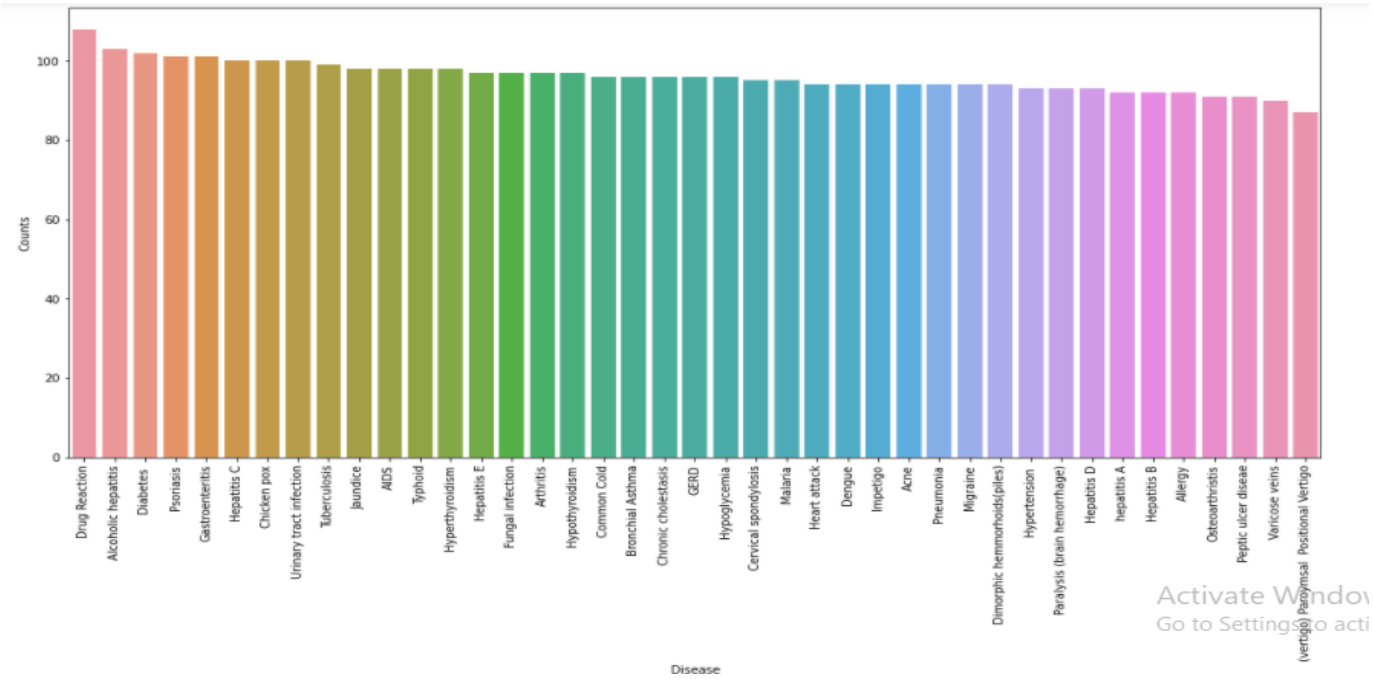
It summarizes how we did the data prepossessing by cleaning it and arranging in the way compatible with the model for best prediction.

Disease	Female	Male	adolesce	adult	child	early-old	infant	middle-a	old	older adı	autumn	spring	summer	winter	abdomer	abdomin	abdomin	abnorma	abnorma	abortion	abscess
hypertensive d	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
diabetes	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
coronary arter	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
coronary heart	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
pneumonia	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
failure heart d	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
depressive dis	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
myocardial int	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
hypercholeste	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
infection	0	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1
infection urin	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
anemia	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
chronic obstru	1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0
dementia	1	0	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	0	0
insufficiency r	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
confusion	0	1	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0
hypothyroidis	1	0	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
anxiety state	1	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
hiv	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
cellulitis	1	1	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1

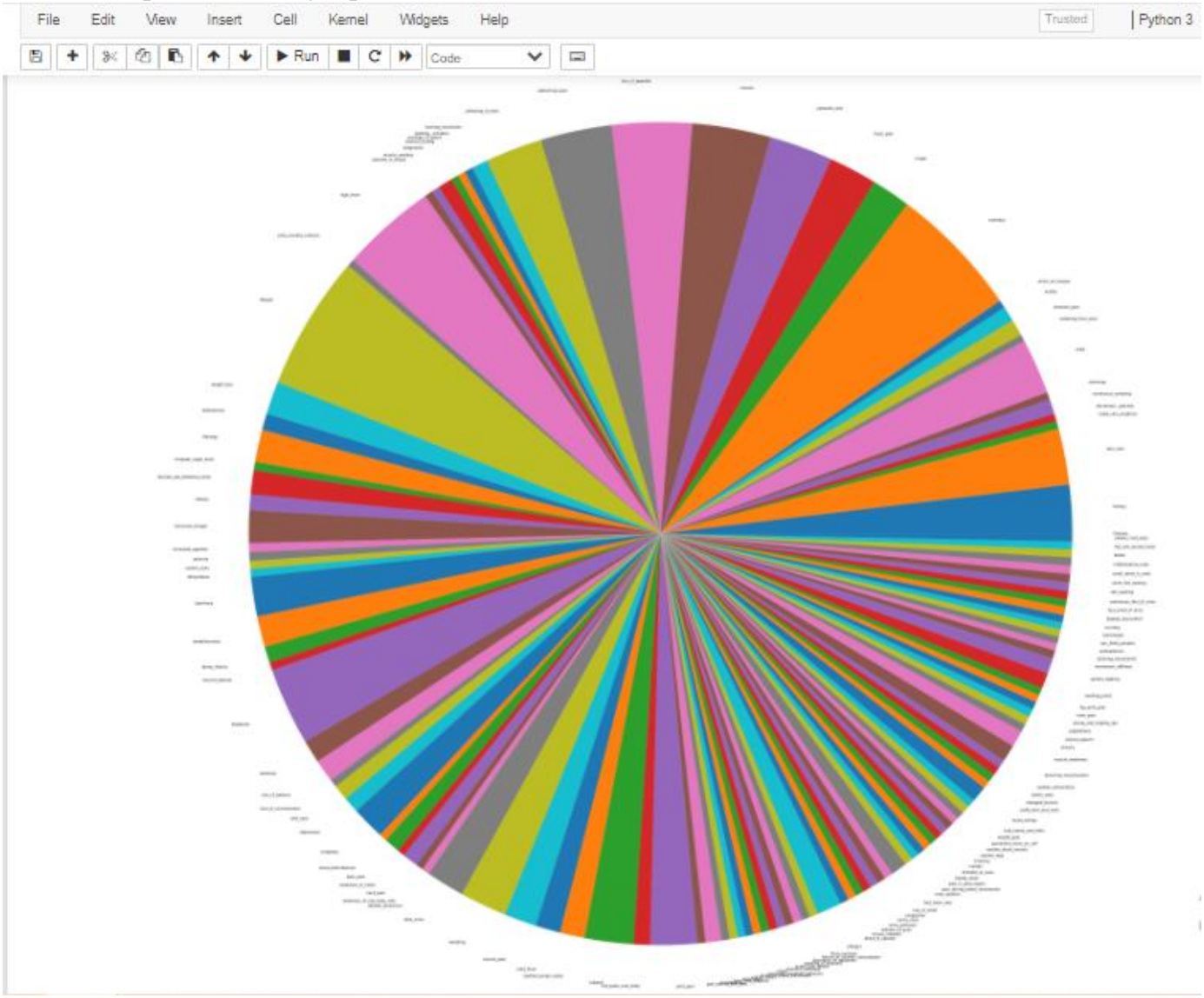
Figure 3.2: Final look of the data set for model training.

Data Analysis

Below figures shows the relation between No. of Diseases and its Occurrence in dataset.



This is the plot for the symptoms vs its Occurrence in Dataset.



ML and DL Model

The Machine Learning model trained on this data set uses the **Artificial Neural Network (ANN), Kth Nearest Neighbour(KNN) and Decision tree**.

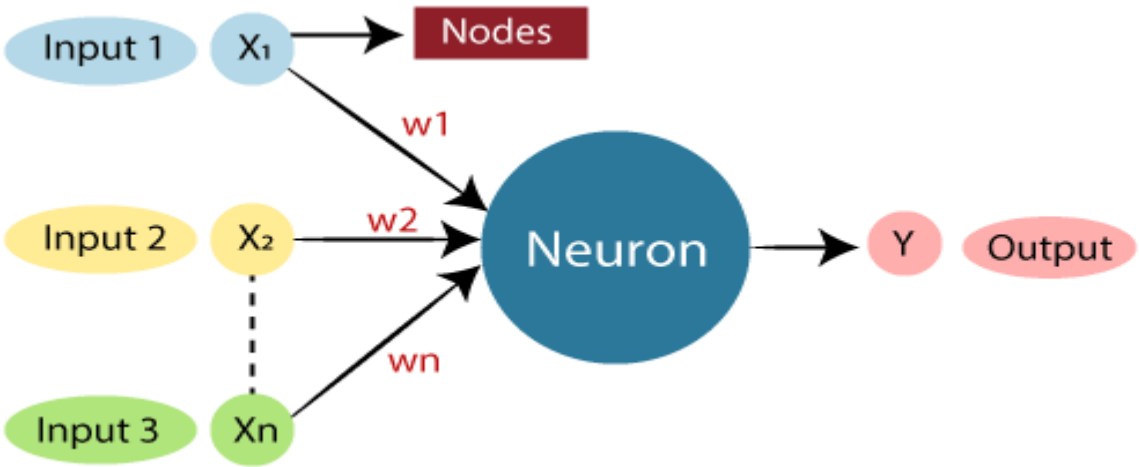
Artificial Neural Network(ANN):

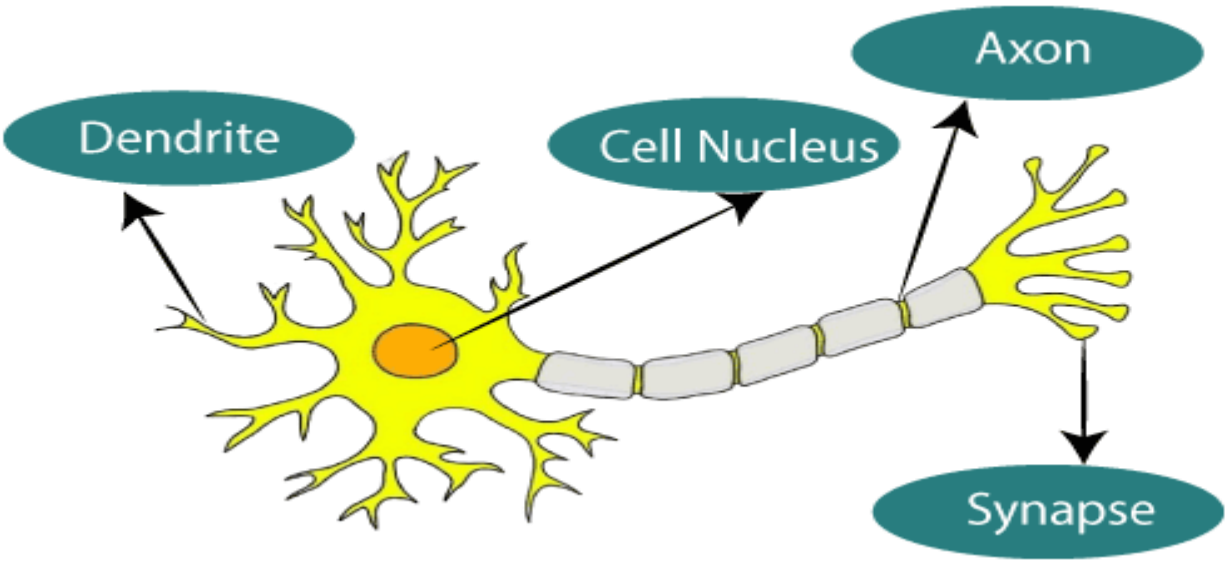
Artificial Neural Network Tutorial provides basic and advanced concepts of ANNs. Our Artificial Neural Network tutorial is developed for beginners as well as professions.

The term "Artificial neural network" refers to a biologically inspired sub-field of artificial intelligence modeled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.

What is Artificial Neural Network?

The term "**Artificial Neural Network**" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

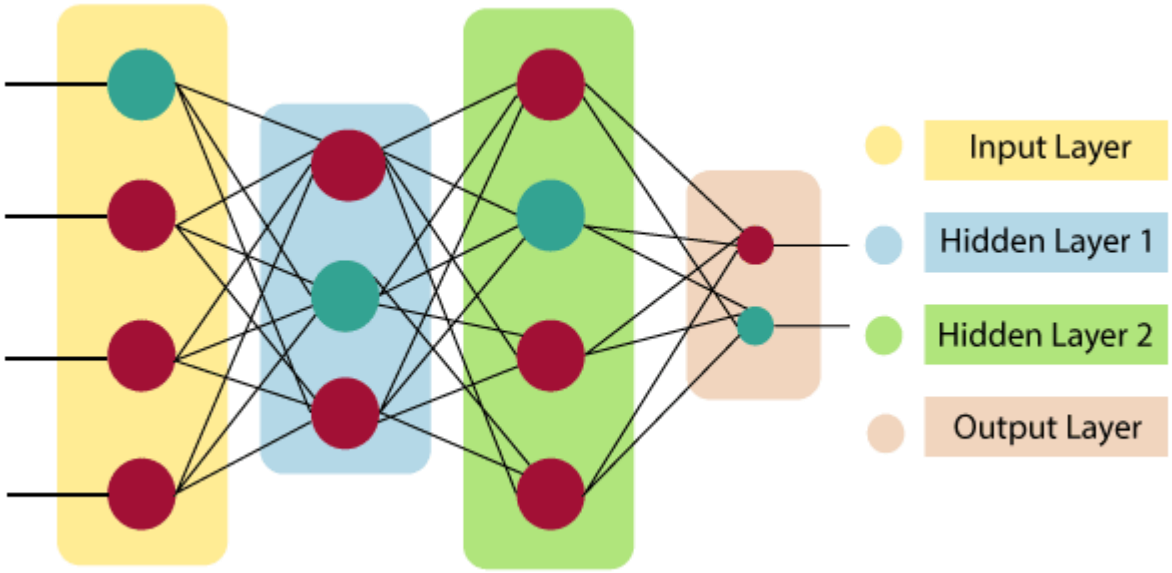




The architecture of an artificial neural network:

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Lets us look at various types of layers available in an artificial neural network.

Artificial Neural Network primarily consists of three layers:



Input Layer:
As the name suggests, it accepts inputs in several different formats provided by the programmer.
Hidden Layer:
The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.
Output Layer:
The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.
The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

Kth Neural Network (KNN):

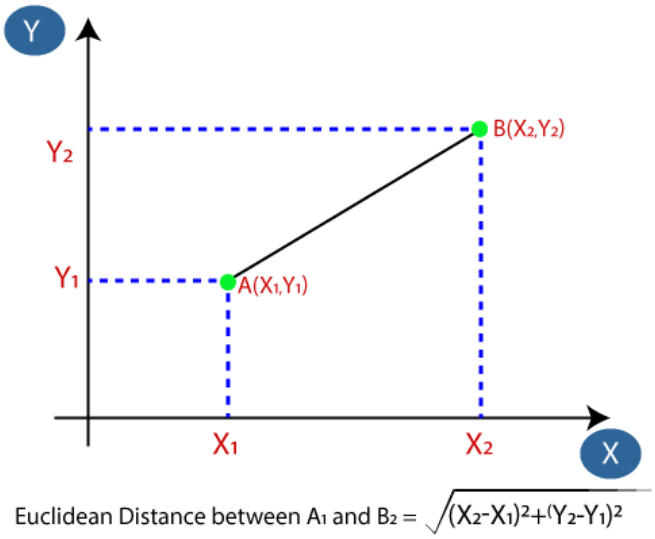
K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. KNN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.KNN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems, KNN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data. It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The K-NN working can be explained on the basis of the below algorithm:

- Select the number K of the neighbors

- Calculate the Euclidean distance of **K number of neighbors**
- Take the K nearest neighbors as per the calculated Euclidean distance.
- Among these k neighbors, count the number of the data points in each category.
- Assign the new data points to that category for which the number of the neighbor is maximum.
- Our model is ready.



Firstly, we will choose the number of neighbors, so we will choose the k=5. Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B.

Decision Tree Algorithm:

In general, Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the data set in different ways based on different conditions. Decisions tress are the most power- ful algorithms that falls under the category of supervised algorithms.

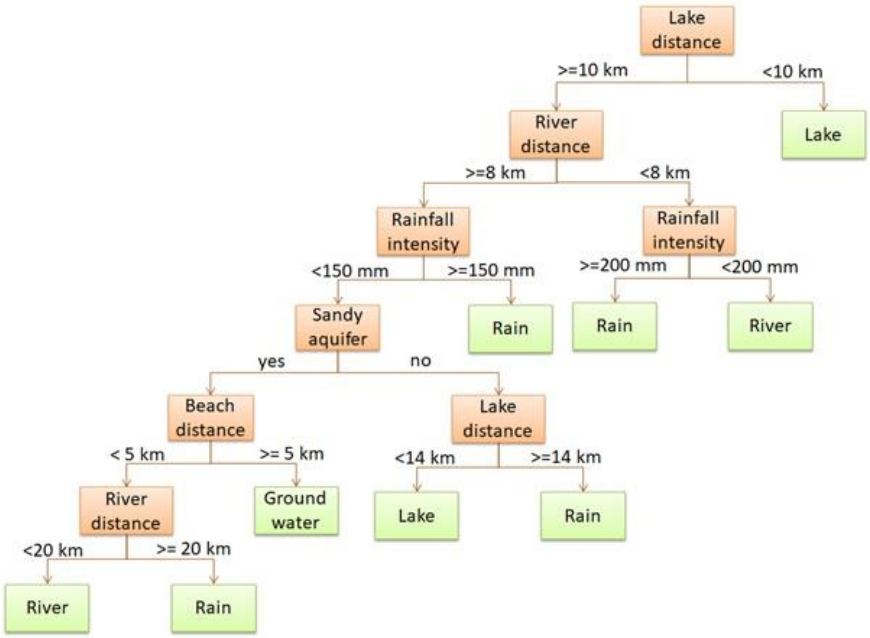
They can be used for both classification and regression tasks. The two main entities of a tree are decision nodes, where the data is split and leaves, where we got outcome. The example of a binary tree for predicting whether

a person is fit or unfit providing various information like age, eating habits and exercise habits, is given below :

In the above decision tree, the question are decision nodes and final outcomes are leaves. We have the following two types of decision trees :

- Classification decision trees: In this kind of decision trees, the decision variable is categorical. The above decision tree is an example of classification decision tree.
- Regression decision trees: In this kind of decision trees, the decision variable is continuous.

Decision tree training is relatively expensive as the complexity and time have taken are more. The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.



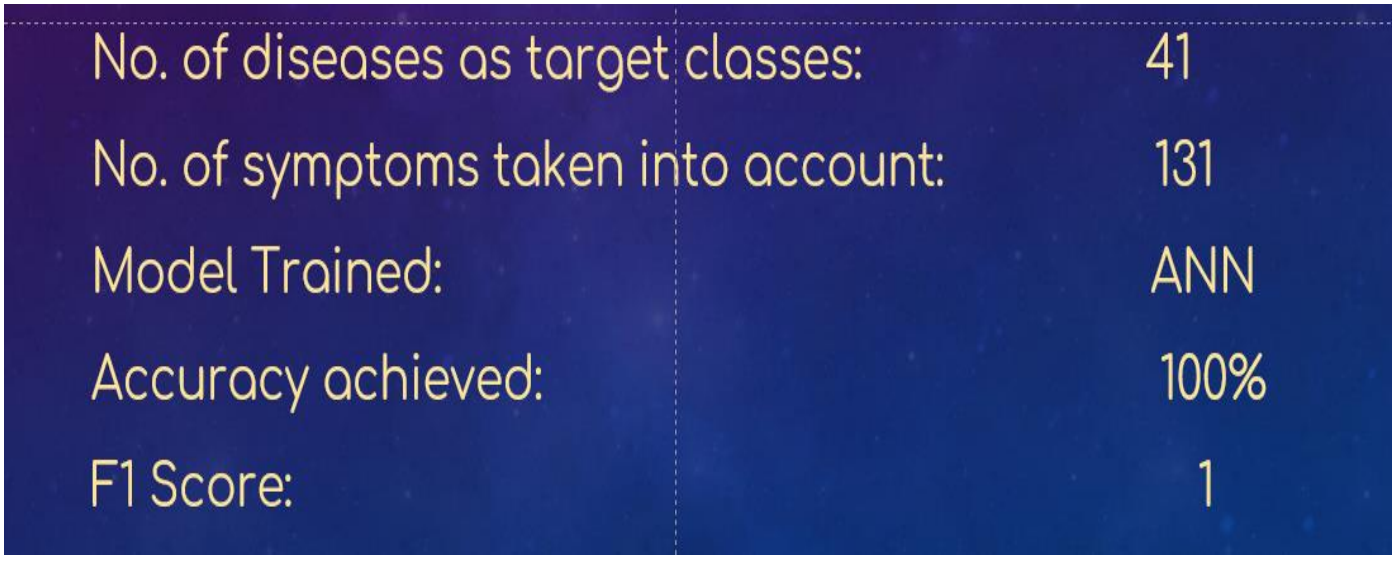
Results

The Result which we are getting by the 3 models are as following and also the GUI
design by us.

Artificial Neural Network (ANN)

The project was implemented in Jupyter Notebook environment using Python 3 language. It uses classifier class from sklearn machine learning module and the tenserflow module to implement the model.

The multi-class Naive Bayes classification model was trained for 41 different diseases as target classes involving 131 different symptoms as features. It achieved a training accuracy of 100%. With the increasing importance of diagnosing disease from symptoms in a short span of time, the framework which We tried to design will surely give fruitful results on real world patient data which is recorded in huge volume in hospitals and nursing homes.

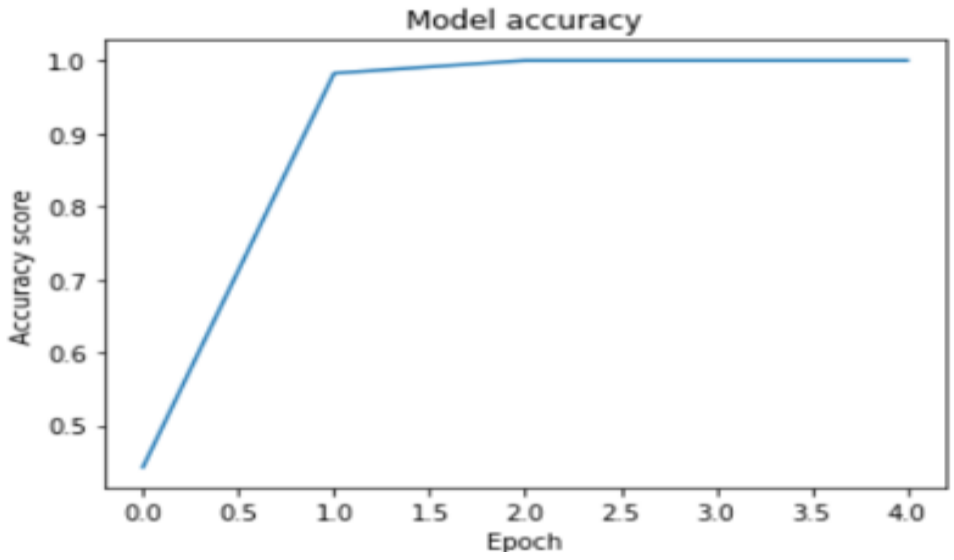


```
In [14]: history = model.fit(X_train, y_train_dum, epochs = 5, batch_size = 30)

Epoch 1/5
132/132 [==============================] - 3s 5ms/step - loss: 2.8370 - accuracy: 0.4428
Epoch 2/5
132/132 [==============================] - 0s 3ms/step - loss: 0.3490 - accuracy: 0.9825
Epoch 3/5
132/132 [==============================] - 0s 2ms/step - loss: 0.0316 - accuracy: 1.0000
Epoch 4/5
132/132 [==============================] - 0s 2ms/step - loss: 0.0120 - accuracy: 1.0000
Epoch 5/5
132/132 [==============================] - 0s 2ms/step - loss: 0.0064 - accuracy: 1.0000
```

The Model given the 100% accuracy after the 3rd epoch

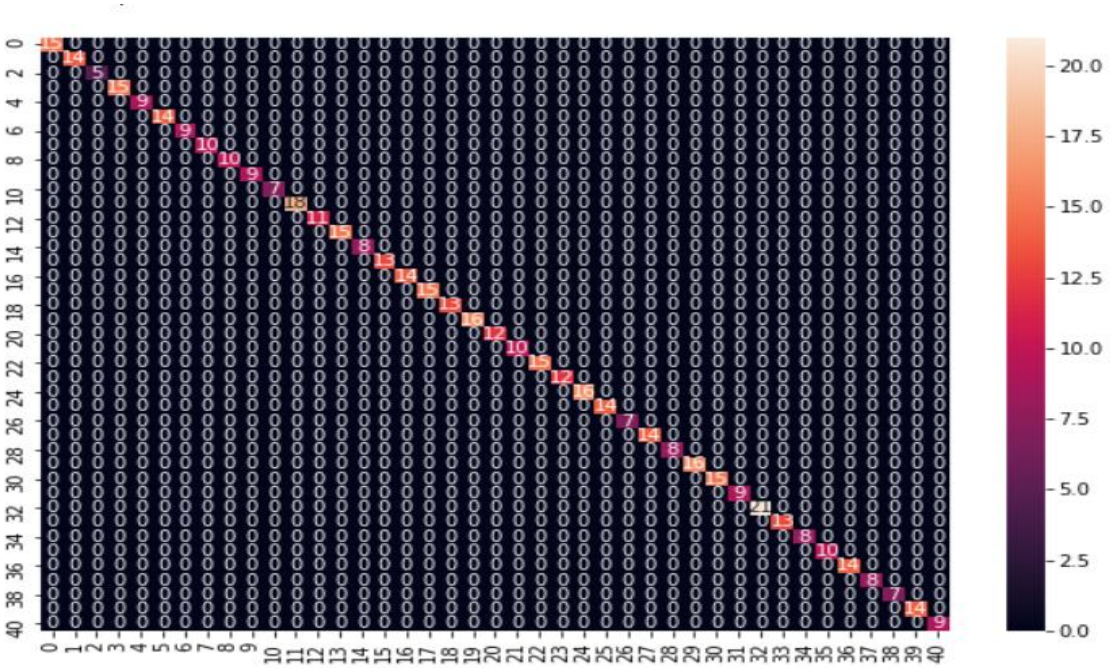
```
In [17]: import matplotlib.pyplot as plt
         plt.plot(history.history["accuracy"])
         plt.title("Model accuracy")
         plt.xlabel("Epoch")
         plt.ylabel("Accuracy score")
         plt.show()
```



Kth Nearest Neighbour (KNN)

The project was implemented in Jupyter Notebook environment using Python 3 language It uses the sklearn machine learning module to implement the KNN. It achieved a training accuracy of 99% .

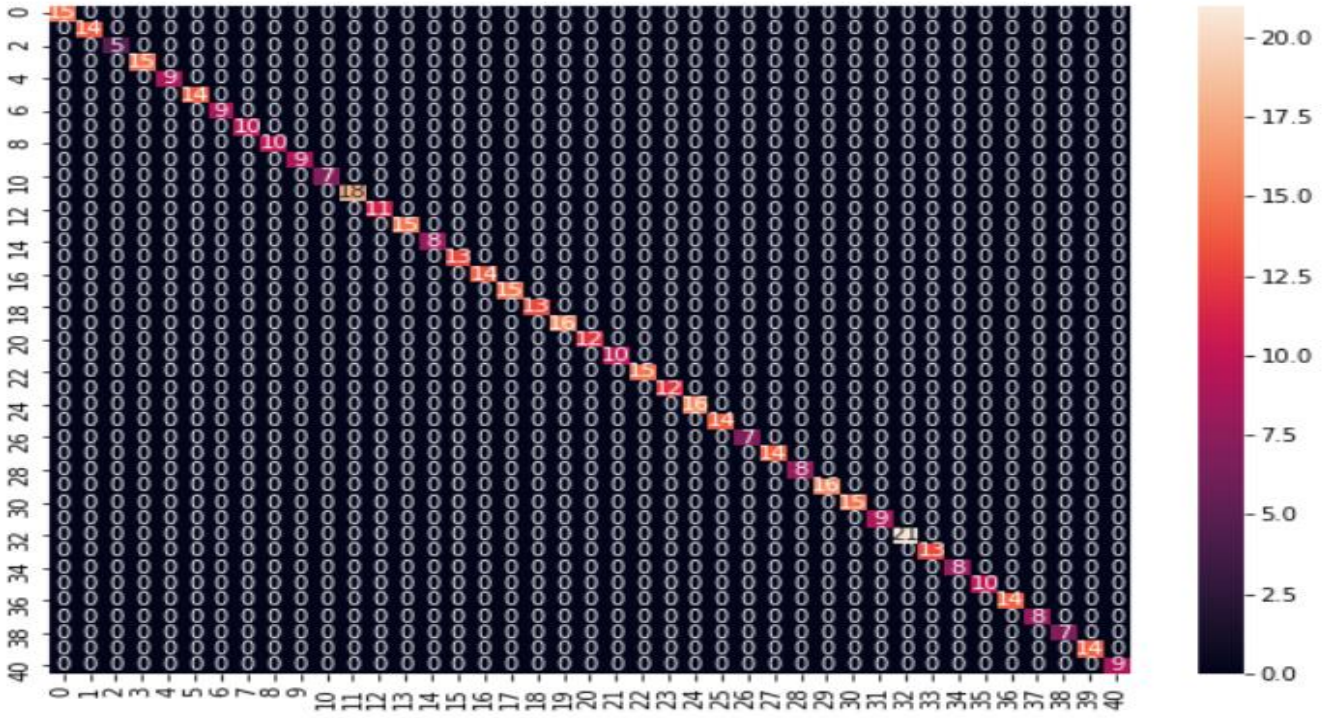
No. of diseases as target classes:	41
No. of symptoms taken into account:	131
Model Trained:	KNN
Accuracy achieved:	99%
F1 Score:	0.99



Decision Tree Model

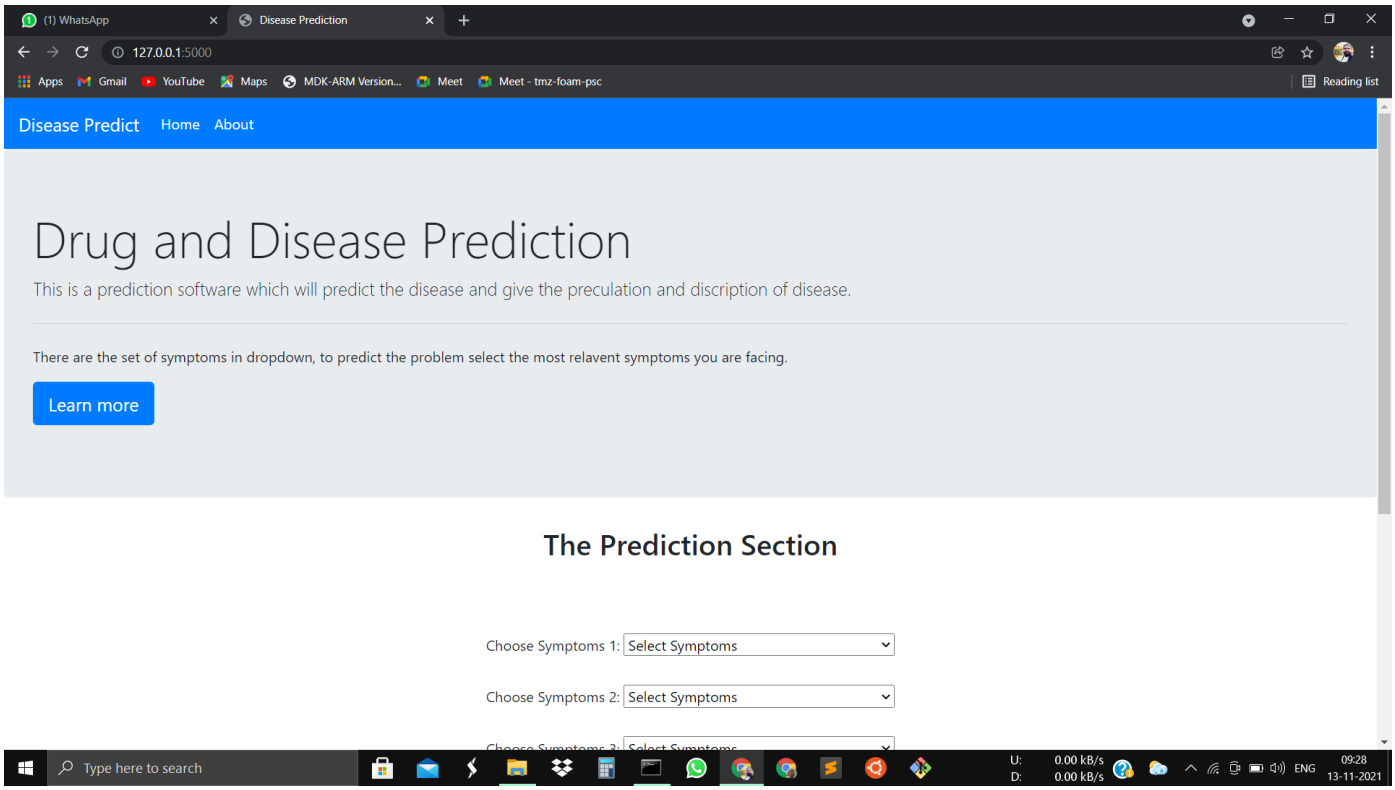
The project was implemented in Jupyter Notebook environment using Python 3 language It uses the sklearn machine learning module to implement the DecisionTreeClassifier. It achieved a training accuracy of 99% .

No. of diseases as target classes:	41
No. of symptoms taken into account:	131
Model Trained:	Decision Tree
Accuracy achieved:	99%
F1 Score:	0.99

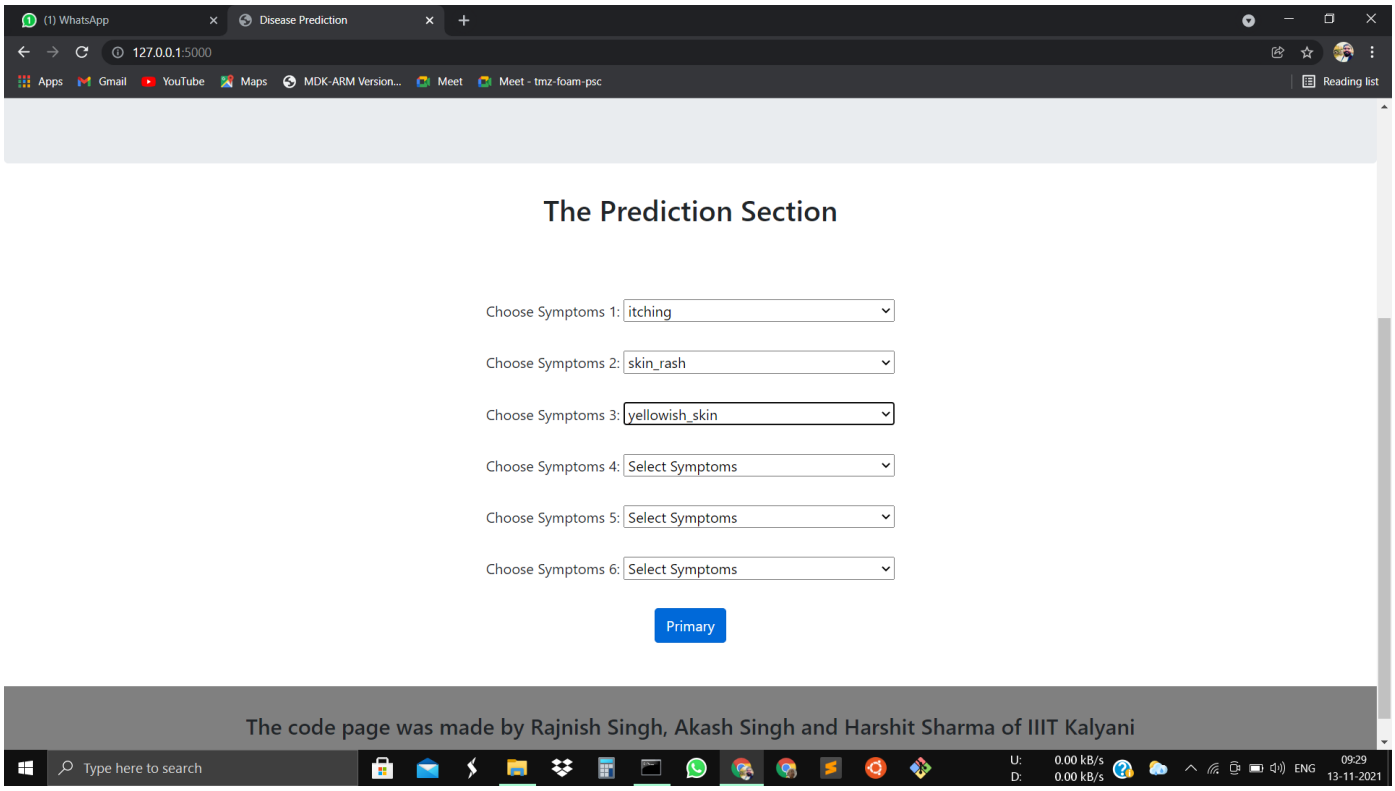


GUI

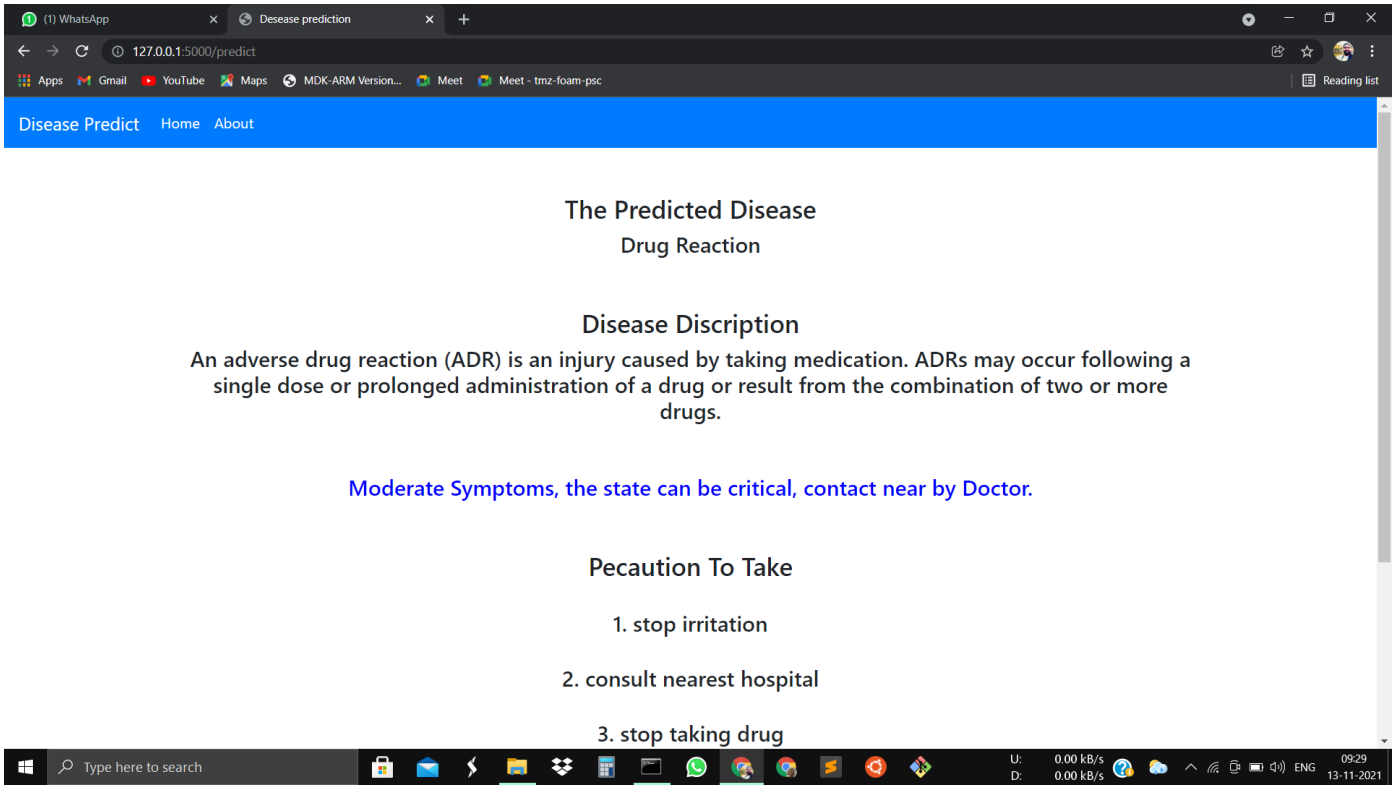
To make our project more user friendly and to make it more presentable we have created GUI. The GUI of the system is designed and implemented using Flask library in python. We have merged both disease prediction system and drug recommendation system in this GUI part. This is the GUI of a complete medical recommendation system.
We have make it more colorful and more presentable.



As shown in figure this is our home page of our medical diagnostic system.
This us the web app so we can launch it on internet. This have a navbar with some dropdown button to select the symptoms.



This is the proper look of the home page were you can select the following Symptoms. After selecting the symptom the click on Primary button that will predict the outcome.



After clicking the button the browser will take you to new page where you will be shown the predicted Disease.

It also has the Discription of Disease which explain the disease give the user the basic idea of what is the Disease is.

Also we have the line which explain the Severeness of the disease with different colour.

1.Green(Miner Symptoms)

2. Blue(Moderate Symptoms)

3. Red (Critical Symptoms)

After the followed by the Precaution of the Disease which Should be Taken by the prevent the more severe effect of disease.

Future Work:

1. In future we plan to work on a larger data set containing patient's diagnosis reports, past medical history and test reports to better diagnose the disease.
2. We need to implement Epidemics also need to be taken into account while predicting the possible disease from patient's symptoms.
3. Apart from ANN, KNN and Decision Tree we will try to Implement other models to check which one gives what accuracy.
4. We are try to build an app and will try to modify web version of this Medical Diagnostic System.
5. And also adding the recommended Drug to be given in every Disease.

Conclusion:

After the analysis of data set we concluded that it was working accurately with given symptoms as parameter but to increase accuracy and make it practically useful we added the multi- parameter as constraints of age, gender, season and past medical history. It's working accurately and with increase in data set it will become more and more accurate and useful.