Introduction to HealthCare Chatbot

The purpose of this project to provide admin has to collect the patients medical history of records and filter it appropriately by applying data preprocessing techniques. Admin's functionalities are to Collecting the appropriate medical records of the patients, handle missing values, handling categorical values,Creating sparse matrix representation, Feeding data to the autonomous pipeline for predictions, selecting and training an appropriate machine learning algorithm.

Visitor can perform the basic task of visitor is to access the chat bot from the front end and reply to its queries with a binary response (Yes/No). The visitor will be shown a confidence interval related to a certain prognosis which needs to be further investigated and experimented with for better results.

The first step is to start their procedure, then one by one all the symtoms come in clients screen. They will have to reply with yes or no answer.

Once a problem will found then they will have to click yes, then patient can see their problem in screen.

The Best Part is that it will provide docter's information like Docter name and his/her website link. So that one can easily find their docter with don't face any type of problem, and start their treatment.

This will prepare with the help of chatbot so that one can even check their problem at any time.You have to just reply with clicking of button Yes or No.

Software Requirement Analysis

Requirement Analysis is a software engineering task that bridges the gap between system level software allocation and software design. It provides the system engineer to specify software function and performance indicate software's interface with the other system elements and establish constraints that software must meet.

The basic aim of this stage is to obtain a clear picture of the needs and requirements of the end-user and also the organization. Analysis involves interaction between the clients and the analysis. Usually analysts research a problem from any questions asked and reading existing documents. The analysts have to uncover the real needs of the user even if they don't know them clearly. During analysis it is essential that a complete and consistent set of specifications emerge for the system. Here it is essential to resolve the contradictions that could emerge from information got from various parties. This is essential to ensure that the final specifications are consistent.

It may be divided into 5 areas of effort.

- Problem recognition
- Evaluation and synthesis
- Modelling
- Specification
- Review

Each Requirement analysis method has a unique point of view. However all analysis methods are related by a set of operational principles. They are

- The information domain of the problem must be represented and understood.
- The functions that the software is to perform must be defined.
- The behaviour of the software as a consequence of external events must be defined.
- The models that depict information function and behavior must be partitioned in a hierarchical or layered fashion.

- The analysis process must move from essential information to implementation detail.

Software Requirements Specification

Software Requirements Specification plays an important role in creating quality software solutions. Specification is basically a representation process. Requirements are represented in a manner that ultimately leads to successful software implementation.

Requirements may be specified in a variety of ways. However there are some guidelines worth following: -

- Representation format and content should be relevant to the problem
- Information contained within the specification should be nested
- Diagrams and other notational forms should be restricted in number and consistent in use.
- Representations should be revisable.

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system engineering are refined by establishing a complete information description, a detailed functional and behavioural description, and indication of performance requirements and design constraints, appropriate validation criteria and other data pertinent to requirements

System Design

System design is the most creative phase of the system development. The term describes a final system and the process by which it is developed. The question in system design is:

How the problem is to be solved?

A systematic method has to achieve the beneficial results at the end. It involves starting with a vague idea and developing it into a series of steps. The series of steps for successful system design are:

- First step is to study the problem completely because we should know the goal. We should see what kind of output we require and what king of input we give so that we can get desired result. We should see what kind of program should be developed to reach the final goal.
- Then we write individual programs, which later on joining solve the specified problem.
- Then we test these programs and make necessary corrections to achieve target of the programs.

ER Diagram

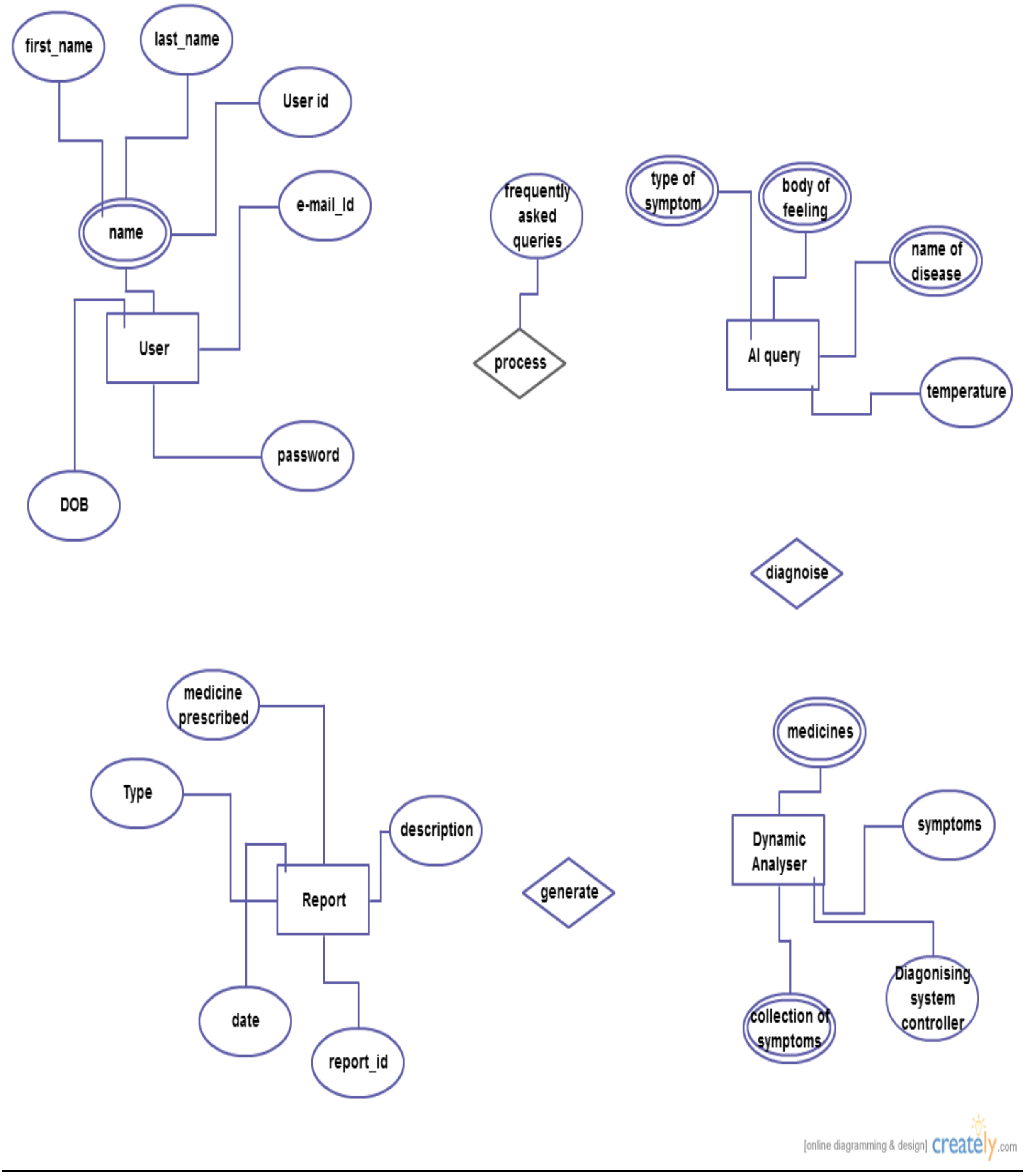


Fig.1:-ER diagram

Testing Phase

As testing is the last phase before the final software is delivered, it has the enormous responsibility of detecting any type of error that may in the software. A software typically undergoes changes even after it has been delivered. And to validate that a change has not affected some old functionality of software regression testing is performed

Levels of Testing

The basic levels of testing are unit testing, integration testing and system and acceptance testing. These different levels of testing attempt to detect different types of faults.

Code/Unit Testing

Code testing and implementation is a critical process that can even consume more than sixty percent of the development time.

Testing

The system development life cycle involves the phases of testing and debugging after the requirement analysis, designing and coding. The project in question was tested , debugged and implemented successfully.

Two strategies of software testing adopted for the new system are as follows

Code testing

Code testing was carried out to see the correctness of the logic involved and the correctness of the modules. Tests were conducted based upon sample. All the modules are checked separately for assuming correctness and accuracy in all the calculations.

Specification testing

It examines the specification stating about what program should do and how it performs under various conditions. This testing strategy is better strategy since it focuses on the way the software is expected to work.

Unit Testing

During the phase of unit testing different constituent modules were testing against the specifications produced during the design for the modules. Unit testing is essentially for the verification of the code produced during the coding the phase, and goal is to test the internal logic of the modules. The modules once tested were then considered for integration and use by others.

Test Planning

Testing needs to be planned, to be cost and time effective. Planning is setting out standards for tests. Test plans set out the context in which individual engineers can place their own work. Typical test plan contains:

Overview of testing process

- Requirements trace ability (to ensure that all requirements are tested)
- List of item to be tested
- Schedule
- Recording procedures so that test results can be audited
- Hardware and software requirements
- Constraints

Implementation

Implementation is the stage in the project where the theoretical design is turned into the working system and is giving confidence to the new system for the users i.e. will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of method to achieve the change over, an evaluation, of change over methods. A part from planning major task of preparing the implementation is education of users. The more complex system is implemented, the more involved will be the system analysis and design effort required just for implementation. An implementation coordinating committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation for the system. According to this plan, the activities are to be carried out, discussions may regarding the equipment has to be acquired to implement the new system.

Implementation is the final and important phase. The most critical stage is in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain types of transaction while using the new system.

The major elements of implementation plan are test plan, training plan, equipment installation plan, and a conversion plan.

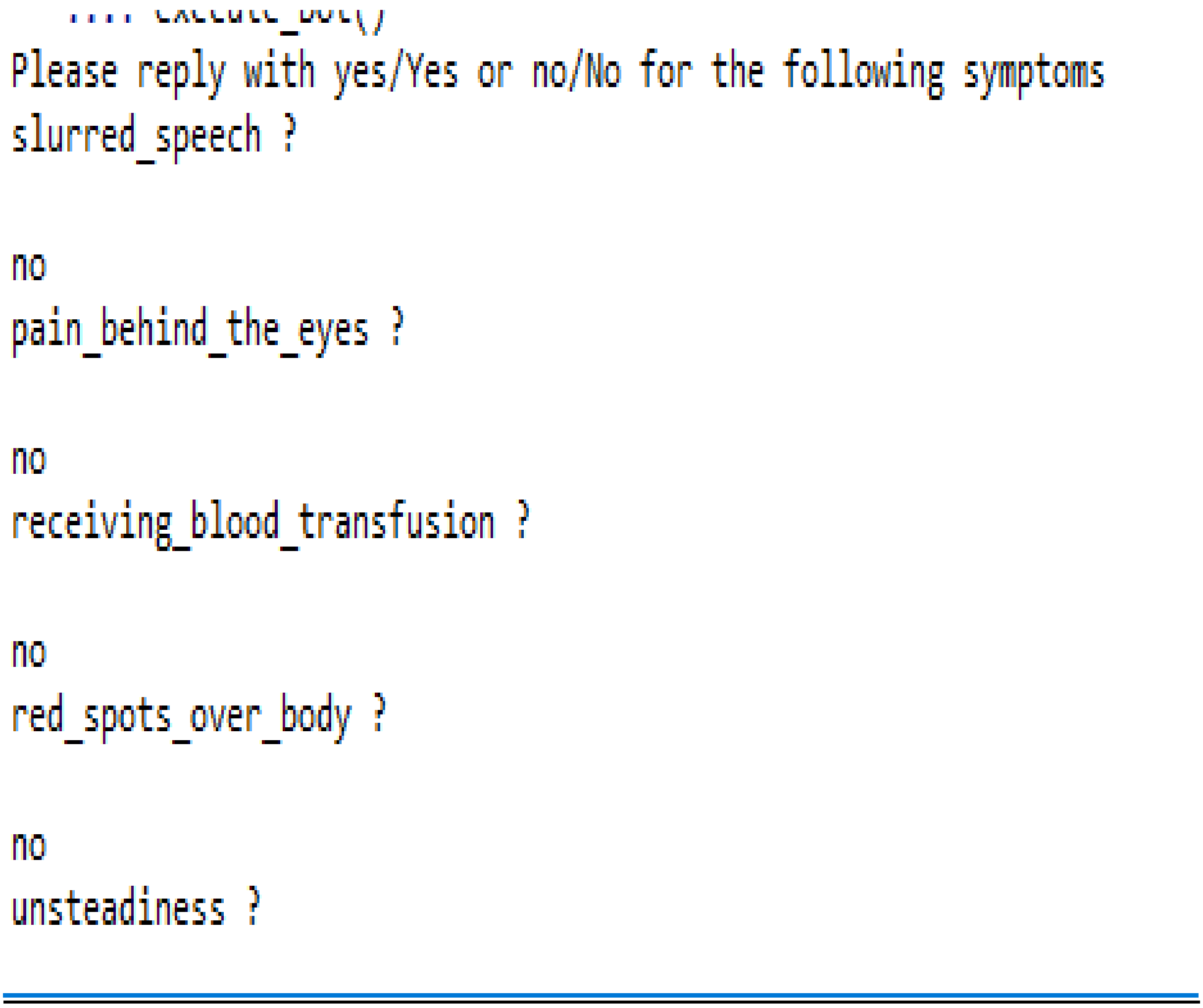
Interface

Console Based

In this, we have to write Yes or No only.

If our Symtoms are not matched then we have to write no in our screen.

When our Symtoms will be matched then we just have to write yes.

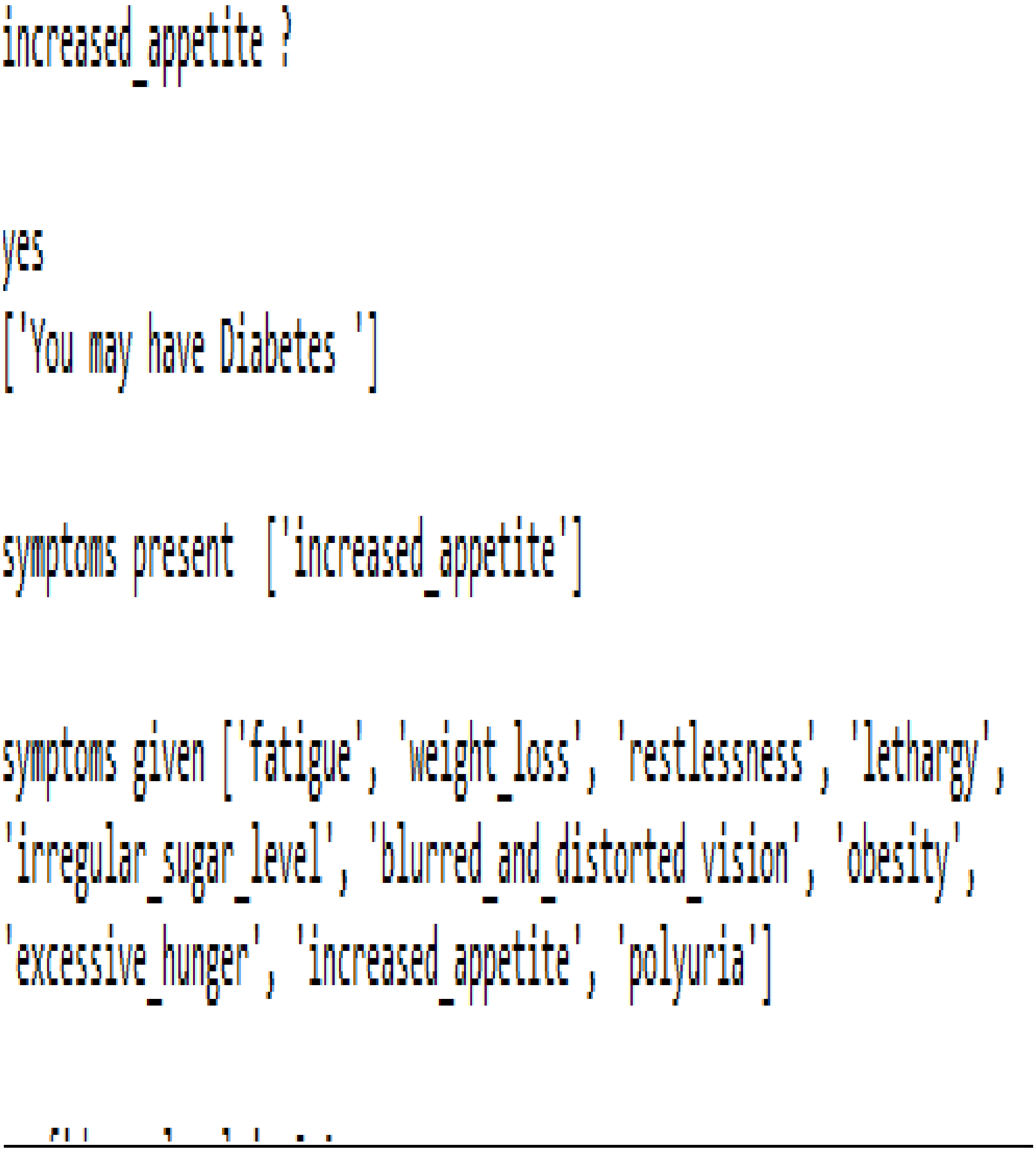


```
    .... execute_bot()
Please reply with yes/Yes or no/No for the following symptoms
slurred_speech ?

no
pain_behind_the_eyes ?

no
receiving_blood_transfusion ?

no
red_spots_over_body ?

no
unsteadiness ?
```

Fig.2:-Console based

Symptoms Window

When we write Yes on our console screen, then our matched problem will be found on screen. And it will also tell the Symptoms which may a patient have.



```
increased_appetite ?

yes
['You may have Diabetes ']

symptoms present  ['increased_appetite']

symptoms given ['fatigue', 'weight_loss', 'restlessness', 'lethargy',
'irregular_sugar_level', 'blurred_and_distorted_vision', 'obesity',
'excessive_hunger', 'increased_appetite', 'polyuria']
```

Fig.2.1:-Symptoms Window

Bibliography

Books

- Learn PYTHON the HARD WAY(Third Edition)
- Introduction to Machine Learning
- Machine Learning with Python Cookbook

Website

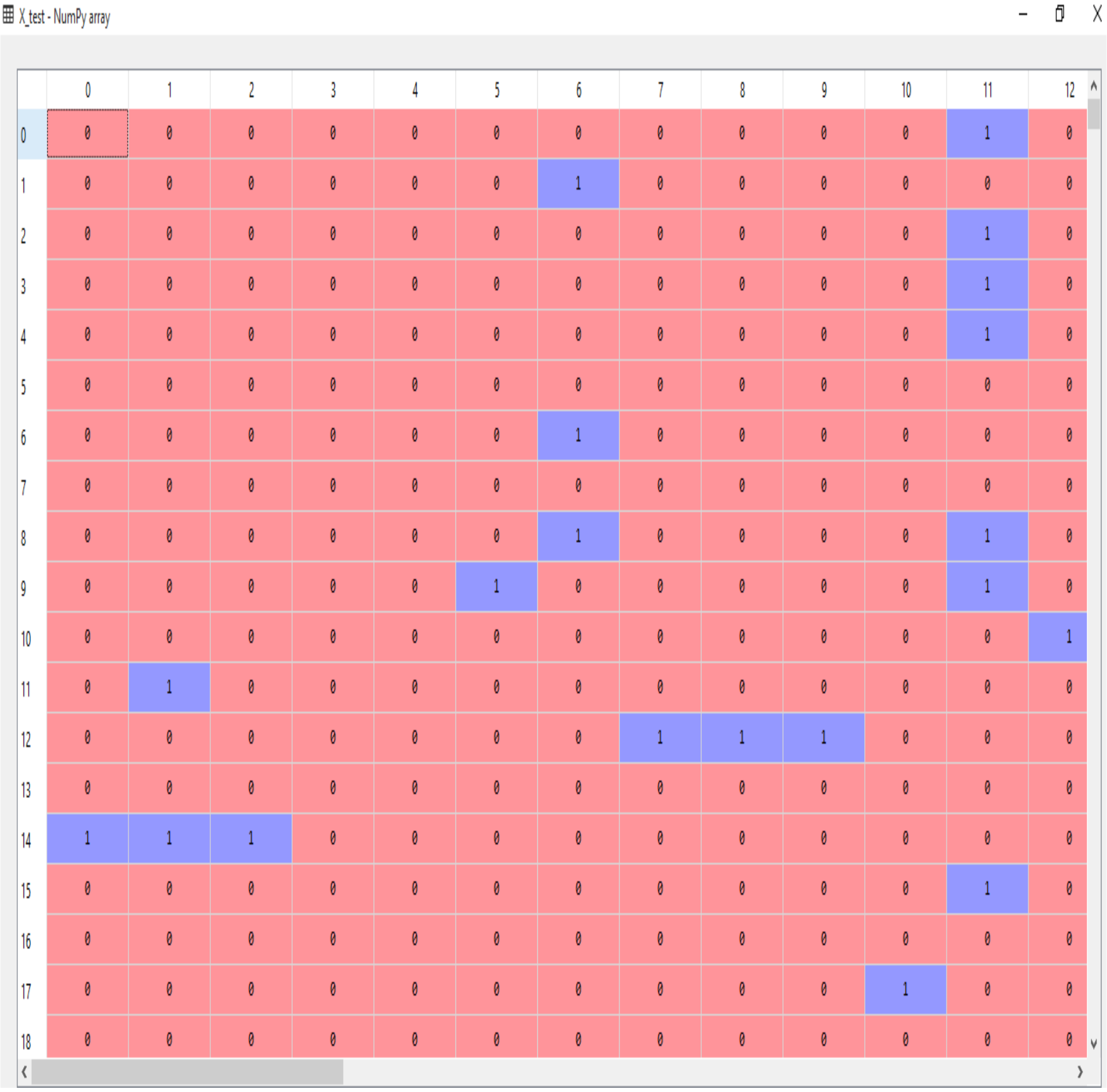
- www.we3schools.com
- www.stackoverflow.com
- www.it-ebooks.com

Appendices

Database Tabels

X_Test Dataset

It divides the x just to check if our output will be correct or not.

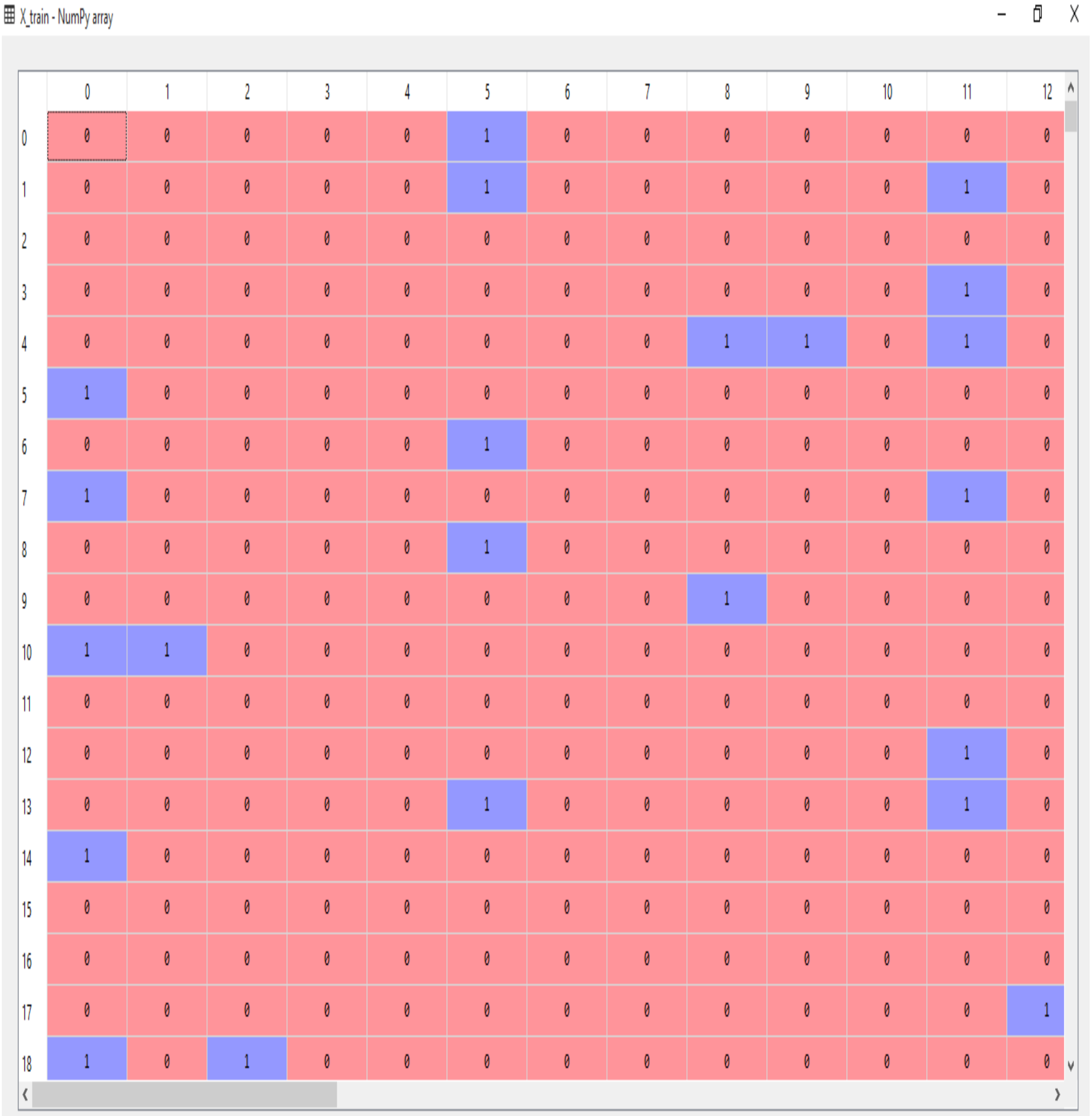


	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	1	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	1	1	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	1	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig.3:-X_Test Dataset

X_Train Dataset

It also divide x dataset into another part called as training set.

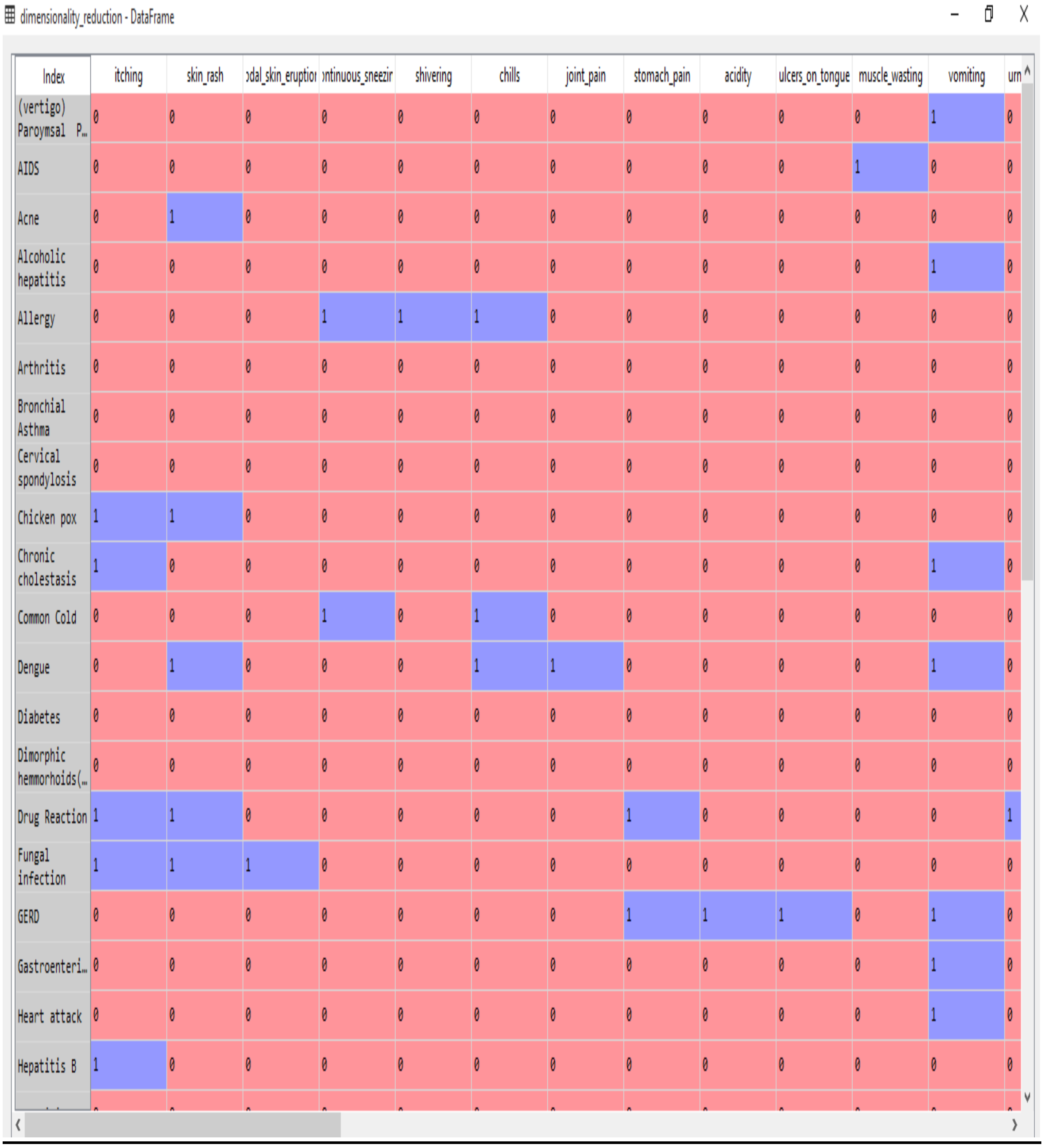


	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	1	1	0	1	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0
10	1	1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	1	0	0	0	0	0	1	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	1
18	1	0	1	0	0	0	0	0	0	0	0	0	0

Fig.3.1:-X_Train Dataset

Dimensionaly_Reduction

To avoid repeat value and contain unique value.



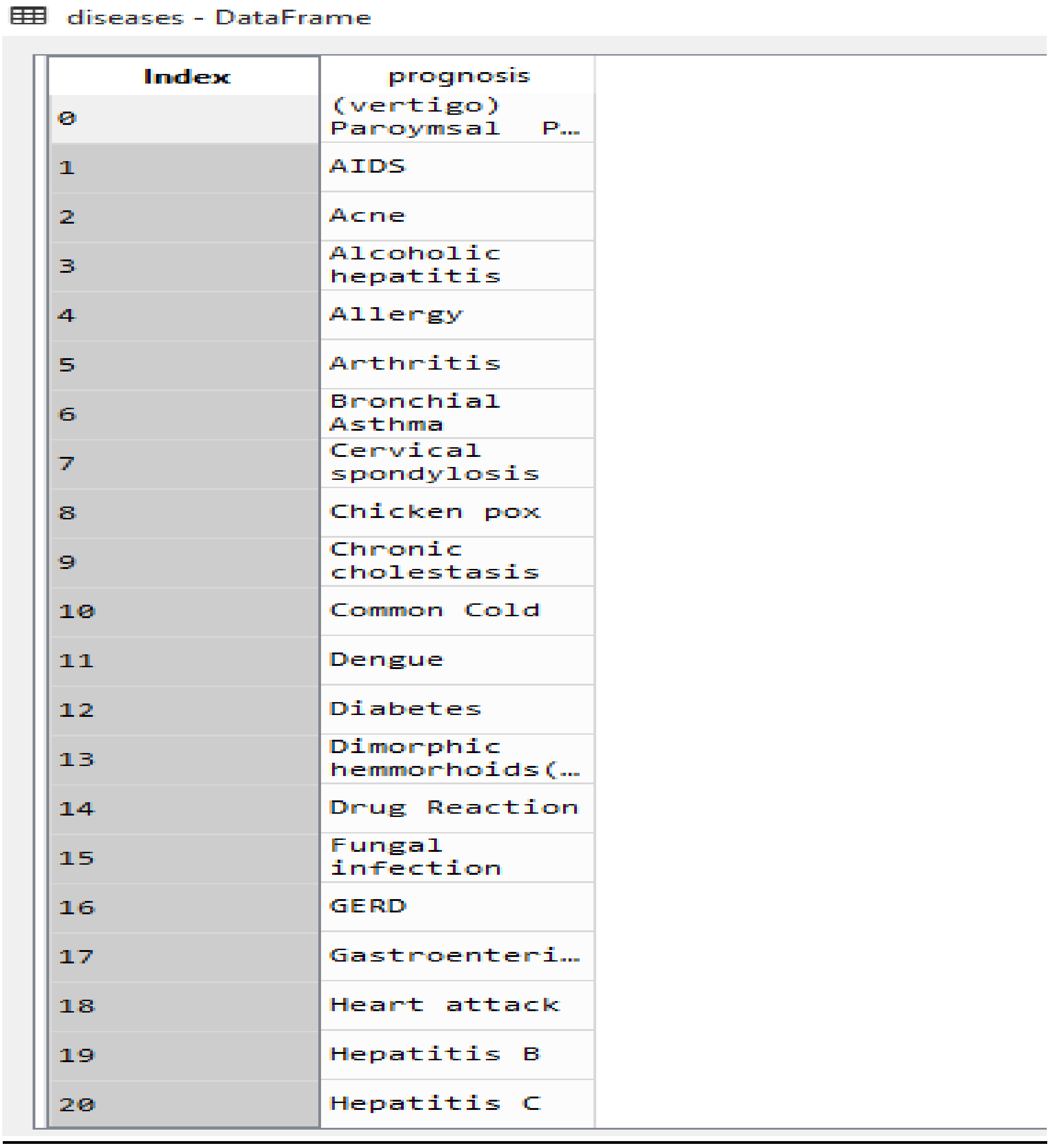
dimensionality_reduction - DataFrame

Index	itching	skin_rash	›dal_skin_eruptior	›ntinuous_sneezir	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	muscle_wasting	vomiting	urn
(vertigo) Paroymsal P…	0	0	0	0	0	0	0	0	0	0	0	1	0
AIDS	0	0	0	0	0	0	0	0	0	0	1	0	0
Acne	0	1	0	0	0	0	0	0	0	0	0	0	0
Alcoholic hepatitis	0	0	0	0	0	0	0	0	0	0	0	1	0
Allergy	0	0	0	1	1	1	0	0	0	0	0	0	0
Arthritis	0	0	0	0	0	0	0	0	0	0	0	0	0
Bronchial Asthma	0	0	0	0	0	0	0	0	0	0	0	0	0
Cervical spondylosis	0	0	0	0	0	0	0	0	0	0	0	0	0
Chicken pox	1	1	0	0	0	0	0	0	0	0	0	0	0
Chronic cholestasis	1	0	0	0	0	0	0	0	0	0	0	1	0
Common Cold	0	0	0	1	0	1	0	0	0	0	0	0	0
Dengue	0	1	0	0	0	1	1	0	0	0	0	1	0
Diabetes	0	0	0	0	0	0	0	0	0	0	0	0	0
Dimorphic hemmorhoids(…	0	0	0	0	0	0	0	0	0	0	0	0	0
Drug Reaction	1	1	0	0	0	0	0	1	0	0	0	0	1
Fungal infection	1	1	1	0	0	0	0	0	0	0	0	0	0
GERD	0	0	0	0	0	0	0	1	1	1	0	1	0
Gastroenteri…	0	0	0	0	0	0	0	0	0	0	0	1	0
Heart attack	0	0	0	0	0	0	0	0	0	0	0	1	0
Hepatitis B	1	0	0	0	0	0	0	0	0	0	0	0	0

Fig.3.2:Dimensionaliy Reduction

Diseases

Number of Diseases.



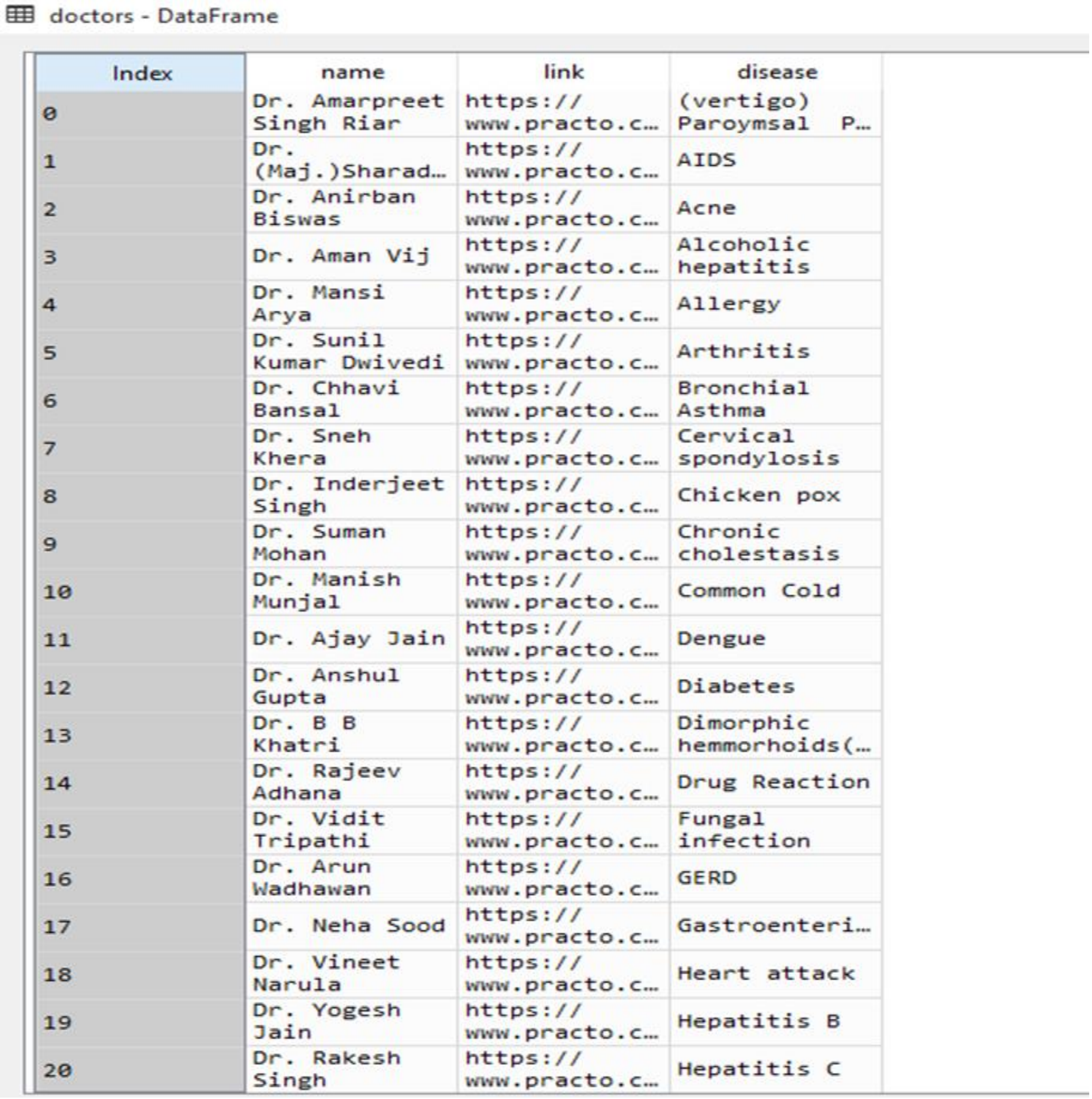
diseases - DataFrame

Index	prognosis
0	(vertigo) Paroymsal P…
1	AIDS
2	Acne
3	Alcoholic hepatitis
4	Allergy
5	Arthritis
6	Bronchial Asthma
7	Cervical spondylosis
8	Chicken pox
9	Chronic cholestasis
10	Common Cold
11	Dengue
12	Diabetes
13	Dimorphic hemmorhoids(…
14	Drug Reaction
15	Fungal infection
16	GERD
17	Gastroenteri…
18	Heart attack
19	Hepatitis B
20	Hepatitis C

Fig.3.3:-Diseases Dataset

Doctor Dataset

It will provide Doctor name with its link where one can visit his/her site and also tell problem. It contain multiple rows and three columns.



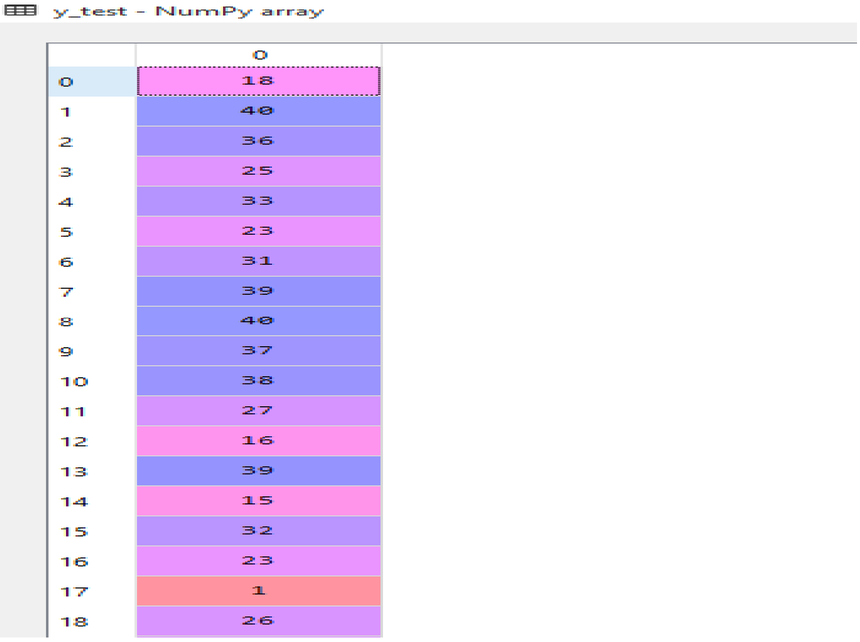
doctors - DataFrame

Index	name	link	disease
0	Dr. Amarpreet Singh Riar	https:// www.practo.c…	(vertigo) Paroymsal P…
1	Dr. (Maj.)Sharad…	https:// www.practo.c…	AIDS
2	Dr. Anirban Biswas	https:// www.practo.c…	Acne
3	Dr. Aman Vij	https:// www.practo.c…	Alcoholic hepatitis
4	Dr. Mansi Arya	https:// www.practo.c…	Allergy
5	Dr. Sunil Kumar Dwivedi	https:// www.practo.c…	Arthritis
6	Dr. Chhavi Bansal	https:// www.practo.c…	Bronchial Asthma
7	Dr. Sneh Khera	https:// www.practo.c…	Cervical spondylosis
8	Dr. Inderjeet Singh	https:// www.practo.c…	Chicken pox
9	Dr. Suman Mohan	https:// www.practo.c…	Chronic cholestasis
10	Dr. Manish Munjal	https:// www.practo.c…	Common Cold
11	Dr. Ajay Jain	https:// www.practo.c…	Dengue
12	Dr. Anshul Gupta	https:// www.practo.c…	Diabetes
13	Dr. B B Khatri	https:// www.practo.c…	Dimorphic hemmorhoids(…
14	Dr. Rajeev Adhana	https:// www.practo.c…	Drug Reaction
15	Dr. Vidit Tripathi	https:// www.practo.c…	Fungal infection
16	Dr. Arun Wadhawan	https:// www.practo.c…	GERD
17	Dr. Neha Sood	https:// www.practo.c…	Gastroenteri…
18	Dr. Vineet Narula	https:// www.practo.c…	Heart attack
19	Dr. Yogesh Jain	https:// www.practo.c…	Hepatitis B
20	Dr. Rakesh Singh	https:// www.practo.c…	Hepatitis C

Fig.3.4:-Doctors Dataset

Y _Test Dataset

It contain output only for test dataset.

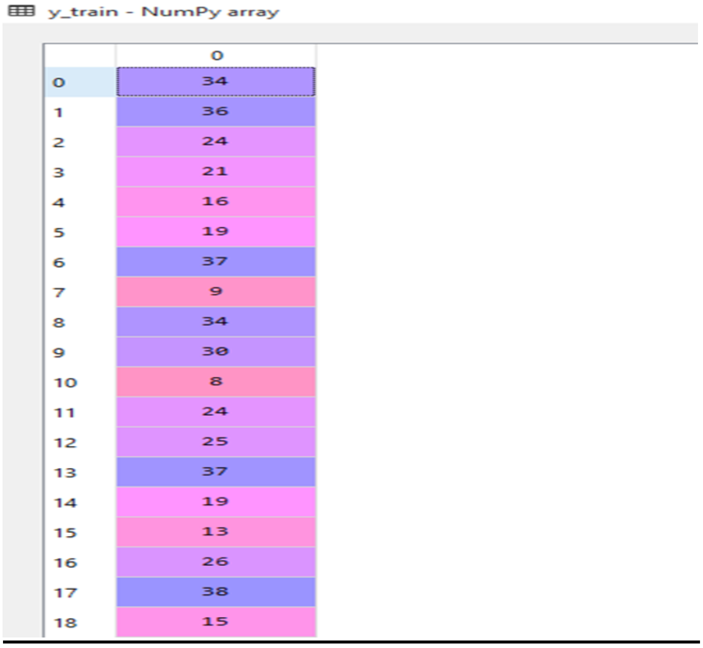


y_test - NumPy array

	0
0	18
1	40
2	36
3	25
4	33
5	23
6	31
7	39
8	40
9	37
10	38
11	27
12	16
13	39
14	15
15	32
16	23
17	1
18	26

Fig.3.5:-Y_Test Dataset

Y_Train Dataset



y_train - NumPy array

	0
0	34
1	36
2	24
3	21
4	16
5	19
6	37
7	9
8	34
9	30
10	8
11	24
12	25
13	37
14	19
15	13
16	26
17	38
18	15

Fig.3.6:-Y_Train Dataset

Code

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

training_dataset = pd.read_csv('Training.csv')

test_dataset = pd.read_csv('Testing.csv')

X = training_dataset.iloc[:, 0:132].values

y = training_dataset.iloc[:, -1].values

dimensionality_reduction = training_dataset.groupby(training_dataset['prognosis']).max()

from sklearn.preprocessing import LabelEncoder

labelencoder = LabelEncoder()

y = labelencoder.fit_transform(y)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier()

classifier.fit(X_train, y_train)

cols     = training_dataset.columns

cols     = cols[:-1]

importances = classifier.feature_importances_

indices = np.argsort(importances)[::-1]
```

```
features = cols

from sklearn.tree import _tree

def execute_bot():

    print("Please reply with yes/Yes or no/No for the following symptoms")

    def print_disease(node):

               node = node[0]

        val  = node.nonzero()

        disease = labelencoder.inverse_transform(val[0])

        return disease

    def tree_to_code(tree, feature_names):

        tree_ = tree.tree_

        feature_name = [

            feature_names[i] if i != _tree.TREE_UNDEFINED else "undefined!"

            for i in tree_.feature

        ]

        symptoms_present = []

        def recurse(node, depth):

            indent = "  " * depth

            if tree_.feature[node] != _tree.TREE_UNDEFINED:

                name = feature_name[node]

                threshold = tree_.threshold[node]

                print(name + " ?")

                ans = input()
                ans = ans.lower()
                if ans == 'yes':
                    val = 1
                else:
                    val = 0
                if  val <= threshold:
                    recurse(tree_.children_left[node], depth + 1)
                else:
                    symptoms_present.append(name)
                    recurse(tree_.children_right[node], depth + 1)
            else:
                present_disease = print_disease(tree_.value[node])
                print( "You may have " +  present_disease )
                print()
                red_cols = dimensionality_reduction.columns
                symptoms_given =
red_cols[dimensionality_reduction.loc[present_disease].values[0].nonzero()]
                print("symptoms present  " + str(list(symptoms_present)))
                print()
                print("symptoms given "  +  str(list(symptoms_given)) )
                print()
                confidence_level =
(1.0*len(symptoms_present))/len(symptoms_given)
```

```
                print("confidence level is " + str(confidence_level))

                print()

                print('The model suggests:')

                print()

                row = doctors[doctors['disease'] == present_disease[0]]

                print('Consult ', str(row['name'].values))

                print()

                print('Visit ', str(row['link'].values))

        recurse(0, 1)

    tree_to_code(classifier,cols)

doc_dataset = pd.read_csv('doctors_dataset.csv', names = ['Name', 'Description'])

diseases = dimensionality_reduction.index

diseases = pd.DataFrame(diseases)

doctors = pd.DataFrame()

doctors['name'] = np.nan

doctors['link'] = np.nan

doctors['disease'] = np.nan

doctors['disease'] = diseases['prognosis']

doctors['name'] = doc_dataset['Name']

doctors['link'] = doc_dataset['Description']

record = doctors[doctors['disease'] == 'AIDS']

record['name']
```

```
record['link']

execute_bot()
```