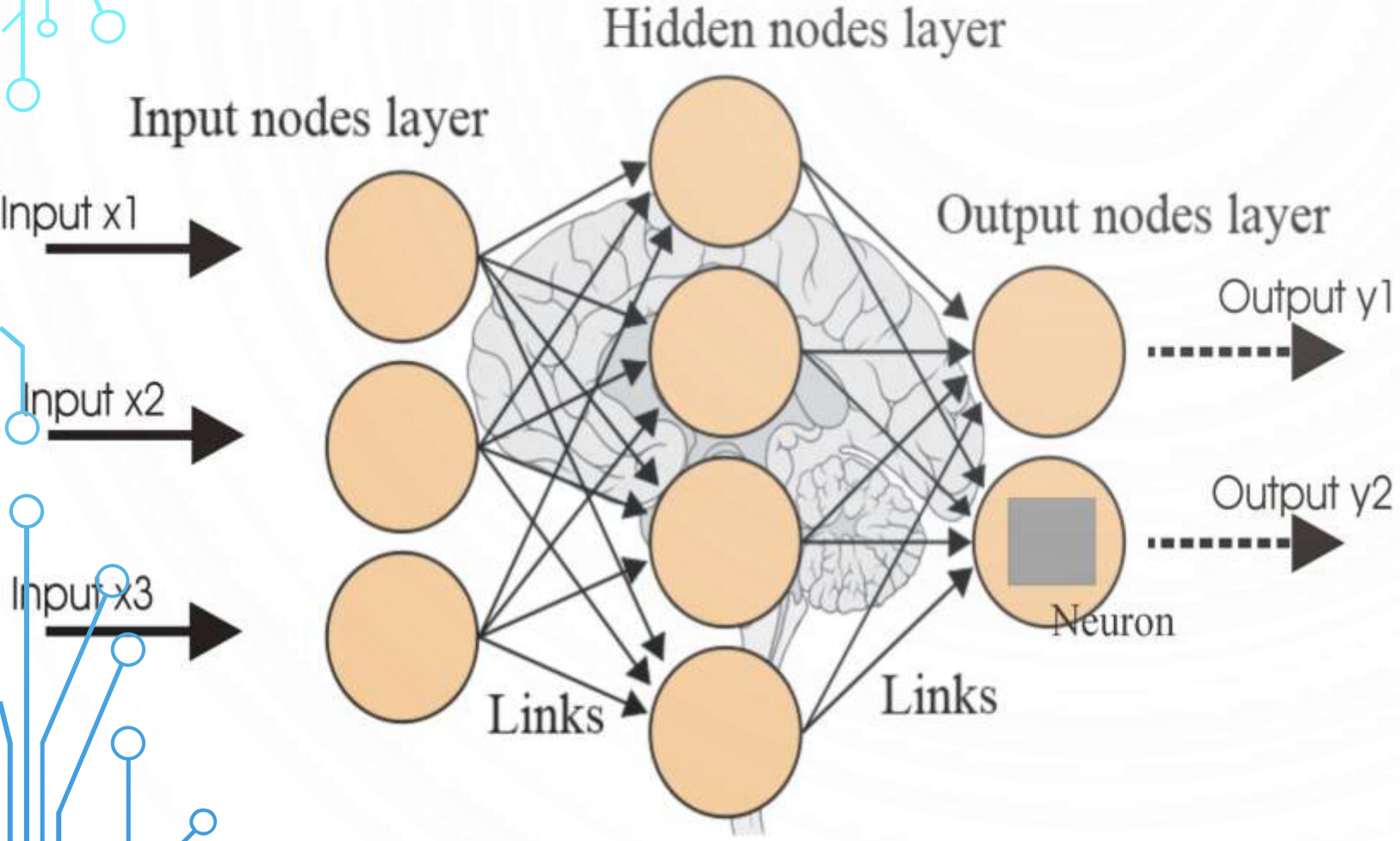
NEURAL NETWORK AND FUZZY CONTROL

- Project: Predicting Diabetes with multilayer perceptron
- Under the guidance of: Dr. Sathish Kumar K
- Slot: D2

Team Members	Registration number
Shreya Biswas	20BEI0072
Pratik Jain	20BEI0091
Pulkit Saraf	20BEI0092

ACKNOWLEDGEMENT



We are doing this project under the course Neural Network And Fuzzy Control. We would like to express our sincere gratitude to our faculty Dr. Sathish Kumar K for providing his valuable guidance, support, comments, suggestions, showing keen interest and encouragement throughout the course of the project .

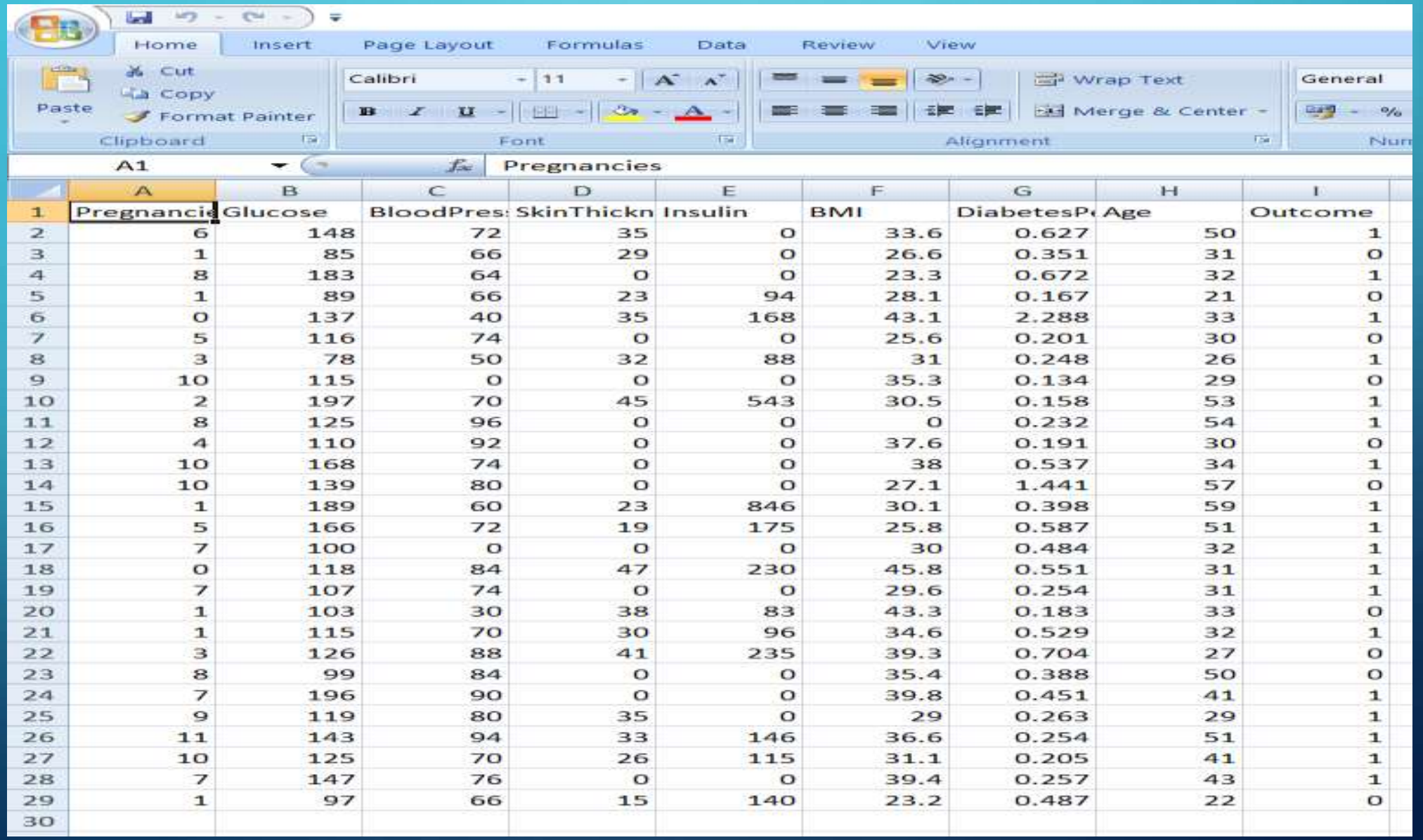
INTRODUCTION

Diabetes is a chronic medical condition that is associated with elevated blood sugar levels in the body. Diabetes often leads to cardiovascular disease, stroke, kidney damage and long term damage to the extremities (i.e. limbs and eyes). It is estimated that there are 415 million people in the world suffering from diabetes, with up to 5 million deaths every year attributed to diabetes-related complications. Clearly, diabetes is a cause of concern to the wellbeing of modern society. Diabetes can be divided into two subtypes: type 1 and type 2

ABSTRACT

- Type 1 diabetes results from the body's inability to produce enough insulin. It is relatively rare compared to type 2 diabetes, and it only accounts for 5% of diabetes. Unfortunately, the exact cause of type 1 diabetes is unknown and therefore, it is difficult to prevent the onset of type 1 diabetes. Type 2 diabetes results from the body's gradual resistance to insulin. Type 2 diabetes is the prevalent form of diabetes in the world, and it is caused by excessive body weight, irregular exercise, and a poor diet. Fortunately, the onset of type 2 diabetes can be prevented and reversed if diagnosed early. One of the barriers for early detection and diagnosis of diabetes is that the early stages of diabetes are often non-symptomatic. People who are on the path to diabetes (also known as prediabetes) often do not know they have diabetes until it is too late.

THE DIABETES MELLITUS DATASET



Home | Insert | Page Layout | Formulas | Data | Review | View

Cut | Copy | Format Painter | Paste | Clipboard

Calibri | 11 | Font

Wrap Text | Merge & Center | Alignment

General | Num

A1 | Pregnancies

	A	B	C	D	E	F	G	H	I
1	Pregnancie	Glucose	BloodPres	SkinThickn	Insulin	BMI	DiabetesP	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1
9	10	115	0	0	0	35.3	0.134	29	0
10	2	197	70	45	543	30.5	0.158	53	1
11	8	125	96	0	0	0	0.232	54	1
12	4	110	92	0	0	37.6	0.191	30	0
13	10	168	74	0	0	38	0.537	34	1
14	10	139	80	0	0	27.1	1.441	57	0
15	1	189	60	23	846	30.1	0.398	59	1
16	5	166	72	19	175	25.8	0.587	51	1
17	7	100	0	0	0	30	0.484	32	1
18	0	118	84	47	230	45.8	0.551	31	1
19	7	107	74	0	0	29.6	0.254	31	1
20	1	103	30	38	83	43.3	0.183	33	0
21	1	115	70	30	96	34.6	0.529	32	1
22	3	126	88	41	235	39.3	0.704	27	0
23	8	99	84	0	0	35.4	0.388	50	0
24	7	196	90	0	0	39.8	0.451	41	1
25	9	119	80	35	0	29	0.263	29	1
26	11	143	94	33	146	36.6	0.254	51	1
27	10	125	70	26	115	31.1	0.205	41	1
28	7	147	76	0	0	39.4	0.257	43	1
29	1	97	66	15	140	23.2	0.487	22	0
30									

Overview of the given dataset

1. Pregnancies – Number of previous pregnancies
2. Glucose – Plasma glucose concentration
3. Blood Pressure – Diastolic blood pressure
4. Skin Thickness – Skin fold thickness measured from the triceps
5. Insulin – Blood serum insulin concentration
6. BMI – Body Mass Index
7. Diabetes Pedigree Function – A summarized score that indicates the genetic predisposition of the patient of diabetes, as extrapolated from the patient's family record for diabetes
8. Age – Age in years
9. Outcome – The target variable we are trying to predict, 1 for patients that developed diabetes within five years of the initial measurement, and 0 otherwise.

CODE

Editor - C:\Users\91626\OneDrive\Documents\MATLAB\Diabetes.m

EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Run | Run and Advance | Run Section | Advance | Run and Time — RUN

Diabetes.m

```
%Read data
clc
clear all
close all
warning off
data=readtable('Dia.csv');
%%
%Checking if null value present or not
a=sum(ismissing(data));
%%
%Feature Scaling
data.Pregnancies=(data.Pregnancies-min(data.Pregnancies))/(max(data.Pregnancies)-min(data.Pregnancies));
data.Glucose=(data.Glucose-min(data.Glucose))/(max(data.Glucose)-min(data.Glucose));
data.BloodPressure=(data.BloodPressure-min(data.BloodPressure))/(max(data.BloodPressure)-min(data.BloodPressure));
data.SkinThickness=(data.SkinThickness-min(data.SkinThickness))/(max(data.SkinThickness)-min(data.SkinThickness));
data.Insulin=(data.Insulin-min(data.Insulin))/(max(data.Insulin)-min(data.Insulin));
data.BMI=(data.BMI-min(data.BMI))/(max(data.BMI)-min(data.BMI));
data.DiabetesPedigreeFunction=(data.DiabetesPedigreeFunction-min(data.DiabetesPedigreeFunction))/(max(data.DiabetesPedigreeFunction)-min(data.DiabetesPedigreeFunction));
data.Age=(data.Age-min(data.Age))/(max(data.Age)-min(data.Age));
%%
%Split data into train test
cv = cvpartition(size(data,1),'HoldOut',0.3);
idx = cv.test;
% Separate to training and test data
dataTrain = data(~idx,:);
dataTest  = data(idx,:);
%%
%Handle the 0 values for some features of training set
g=(dataTrain.Glucose==0);
k=(dataTrain.BloodPressure==0);
l=(dataTrain.SkinThickness==0);
m=(dataTrain.Insulin==0);
n=(dataTrain.BMI==0);
```

Editor - C:\Users\91626\OneDrive\Documents\MATLAB\Diabetes.m

EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Run | Run and Advance | Run Section | Advance | Run and Time — RUN

Diabetes.m

```
34 -    o=(dataTrain.DiabetesPedigreeFunction==0);
35 -    p=(dataTrain.Age==0);
36 -    impractical=[sum(g) sum(k) sum(l) sum(m) sum(n) sum(o) sum(p)];
37      %%
38      %Replace those with average of the corresponding columns of training data
39 -    g=(dataTrain.Glucose==0);
40 -    msa=(dataTrain.Outcome==1);
41 -    ks=mean(dataTrain.Glucose(~g & msa));
42 -    dataTrain.Glucose(g & msa)=ks;
43 -    ks=mean(dataTrain.Glucose(~g & ~msa));
44 -    dataTrain.Glucose(g & ~msa)=ks;
45 -    g=(dataTrain.BloodPressure==0);
46 -    ks=mean(dataTrain.BloodPressure(~g & msa));
47 -    dataTrain.BloodPressure(g & msa)=ks;
48 -    ks=mean(dataTrain.BloodPressure(~g & ~msa));
49 -    dataTrain.BloodPressure(g & ~msa)=ks;
50 -    g=(dataTrain.SkinThickness==0);
51 -    ks=mean(dataTrain.SkinThickness(~g & msa));
52 -    dataTrain.SkinThickness(g & msa)=ks;
53 -    ks=mean(dataTrain.SkinThickness(~g & ~msa));
54 -    dataTrain.SkinThickness(g & ~msa)=ks;
55 -    g=(dataTrain.Insulin==0);
56 -    ks=mean(dataTrain.Insulin(~g & msa));
57 -    dataTrain.Insulin(g & msa)=ks;
58 -    ks=mean(dataTrain.Insulin(~g & ~msa));
59 -    dataTrain.Insulin(g & ~msa)=ks;
60 -    g=(dataTrain.BMI==0);
61 -    ks=mean(dataTrain.BMI(~g & msa));
62 -    dataTrain.BMI(g & msa)=ks;
63 -    ks=mean(dataTrain.BMI(~g & ~msa));
64 -    dataTrain.BMI(g & ~msa)=ks;
65 -    g=(dataTrain.DiabetesPedigreeFunction==0);
66 -    ks=mean(dataTrain.DiabetesPedigreeFunction(~g & msa));
```

Editor - C:\Users\91626\OneDrive\Documents\MATLAB\Diabetes.m

EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Run | Run and Advance | Run Section | Advance | Run and Time — RUN

Diabetes.m

```
67 -    dataTrain.DiabetesPedigreeFunction(g & msa)=ks;
68 -    ks=mean(dataTrain.DiabetesPedigreeFunction(~g & ~msa));
69 -    dataTrain.DiabetesPedigreeFunction(g & ~msa)=ks;
70 -    dataTrain.Age(p)=mean(dataTrain.Age(~p));
71      %%
72      %Handle the 0 values for some features of test set
73 -    g=(dataTest.Glucose==0);
74 -    k=(dataTest.BloodPressure==0);
75 -    l=(dataTest.SkinThickness==0);
76 -    m=(dataTest.Insulin==0);
77 -    n=(dataTest.BMI==0);
78 -    o=(dataTest.DiabetesPedigreeFunction==0);
79 -    p=(dataTest.Age==0);
80 -    impractical=[sum(g) sum(k) sum(l) sum(m) sum(n) sum(o) sum(p)];
81      %%
82      %Replace those with average of the corresponding columns of test data
83 -    g=(dataTest.Glucose==0);
84 -    msa=(dataTest.Outcome==1);
85 -    ks=mean(dataTest.Glucose(~g & msa));
86 -    dataTest.Glucose(g & msa)=ks;
87 -    ks=mean(dataTest.Glucose(~g & ~msa));
88 -    dataTest.Glucose(g & ~msa)=ks;
89 -    g=(dataTest.BloodPressure==0);
90 -    ks=mean(dataTest.BloodPressure(~g & msa));
91 -    dataTest.BloodPressure(g & msa)=ks;
92 -    ks=mean(dataTest.BloodPressure(~g & ~msa));
93 -    dataTest.BloodPressure(g & ~msa)=ks;
94 -    g=(dataTest.SkinThickness==0);
95 -    ks=mean(dataTest.SkinThickness(~g & msa));
96 -    dataTest.SkinThickness(g & msa)=ks;
97 -    ks=mean(dataTest.SkinThickness(~g & ~msa));
98 -    dataTest.SkinThickness(g & ~msa)=ks;
99 -    g=(dataTest.Insulin==0);
```

Editor - C:\Users\91626\OneDrive\Documents\MATLAB\Diabetes.m

EDITOR | PUBLISH | VIEW

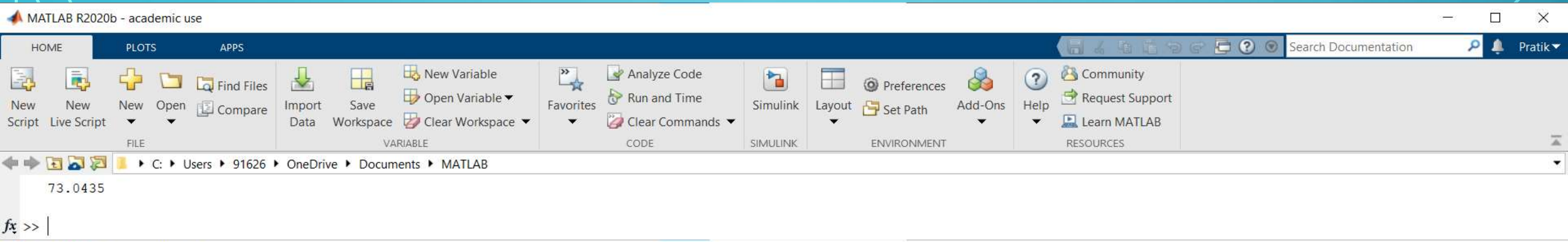
New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Run | Run and Advance | Run Section | Advance | Run and Time — RUN

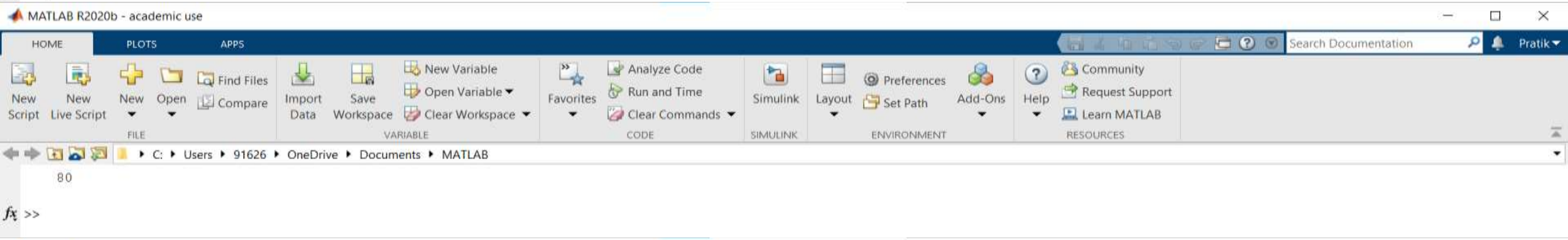
Diabetes.m

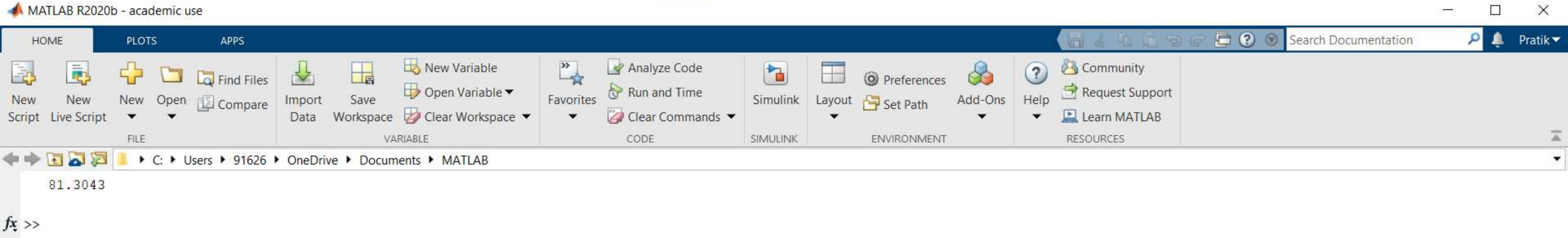
```
91 -    dataTest.BloodPressure(g & msa)=ks;
92 -    ks=mean(dataTest.BloodPressure(~g & ~msa));
93 -    dataTest.BloodPressure(g & ~msa)=ks;
94 -    g=(dataTest.SkinThickness==0);
95 -    ks=mean(dataTest.SkinThickness(~g & msa));
96 -    dataTest.SkinThickness(g & msa)=ks;
97 -    ks=mean(dataTest.SkinThickness(~g & ~msa));
98 -    dataTest.SkinThickness(g & ~msa)=ks;
99 -    g=(dataTest.Insulin==0);
100 -   ks=mean(dataTest.Insulin(~g & msa));
101 -   dataTest.Insulin(g & msa)=ks;
102 -   ks=mean(dataTest.Insulin(~g & ~msa));
103 -   dataTest.Insulin(g & ~msa)=ks;
104 -   g=(dataTest.BMI==0);
105 -   ks=mean(dataTest.BMI(~g & msa));
106 -   dataTest.BMI(g & msa)=ks;
107 -   ks=mean(dataTest.BMI(~g & ~msa));
108 -   dataTest.BMI(g & ~msa)=ks;
109 -   g=(dataTest.DiabetesPedigreeFunction==0);
110 -   ks=mean(dataTest.DiabetesPedigreeFunction(~g & msa));
111 -   dataTest.DiabetesPedigreeFunction(g & msa)=ks;
112 -   ks=mean(dataTest.DiabetesPedigreeFunction(~g & ~msa));
113 -   dataTest.DiabetesPedigreeFunction(g & ~msa)=ks;
114 -   dataTest.Age(p)=mean(dataTest.Age(~p));
115     %%
116     %Train model using training sets
117 -   classification_model=fitcsvm(dataTrain,'Outcome~Pregnancies+Glucose+BloodPressure+SkinThickness+Insulin+BMI+DiabetesPedigreeFunction+Age');
118     %%
119     %Accuracy
120 -   gs=dataTest(:,1:8);
121 -   mk=predict(classification_model,gs);
122 -   accurancy_check=((sum((mk==table2array(dataTest(:,9)))))/size(dataTest,1))*100;
123 -   disp(accurancy_check);
```

UTF-8 | script

OUTPUT


MATLAB R2020b - academic use
HOME
PLOTS
APPS
Search Documentation
Pratik
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: Users 91626 OneDrive Documents MATLAB
73.0435
>>


MATLAB R2020b - academic use
HOME
PLOTS
APPS
Search Documentation
Pratik
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: Users 91626 OneDrive Documents MATLAB
80
>>

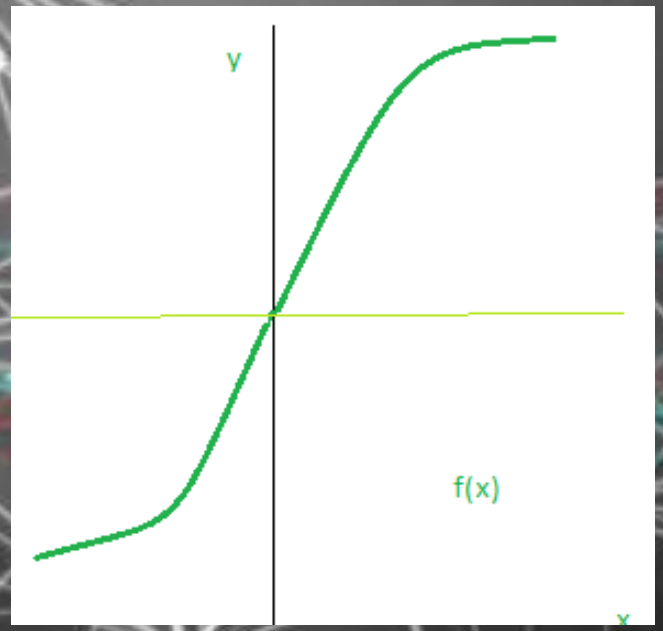

MATLAB R2020b - academic use
HOME
PLOTS
APPS
Search Documentation
Pratik
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: Users 91626 OneDrive Documents MATLAB
81.3043
>>

ACTIVATION FUNCTION

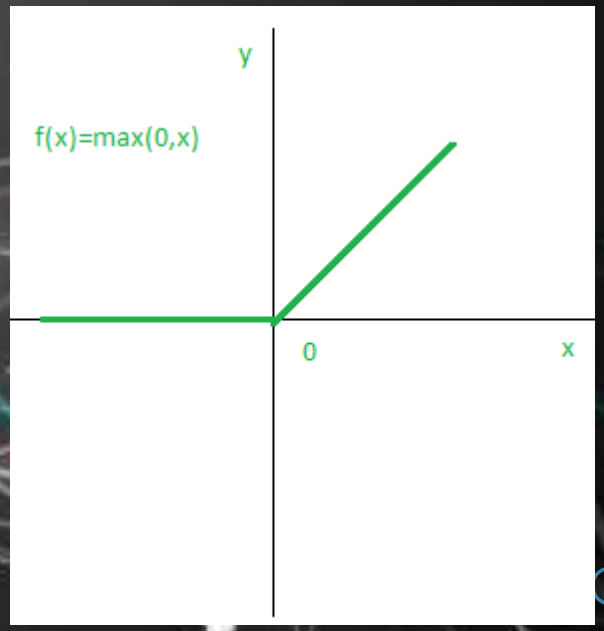
In this project, we will use the rectified linear unit (ReLU) and the sigmoid as our activation functions. ReLU is always used as the activation function for our intermediate hidden layers.

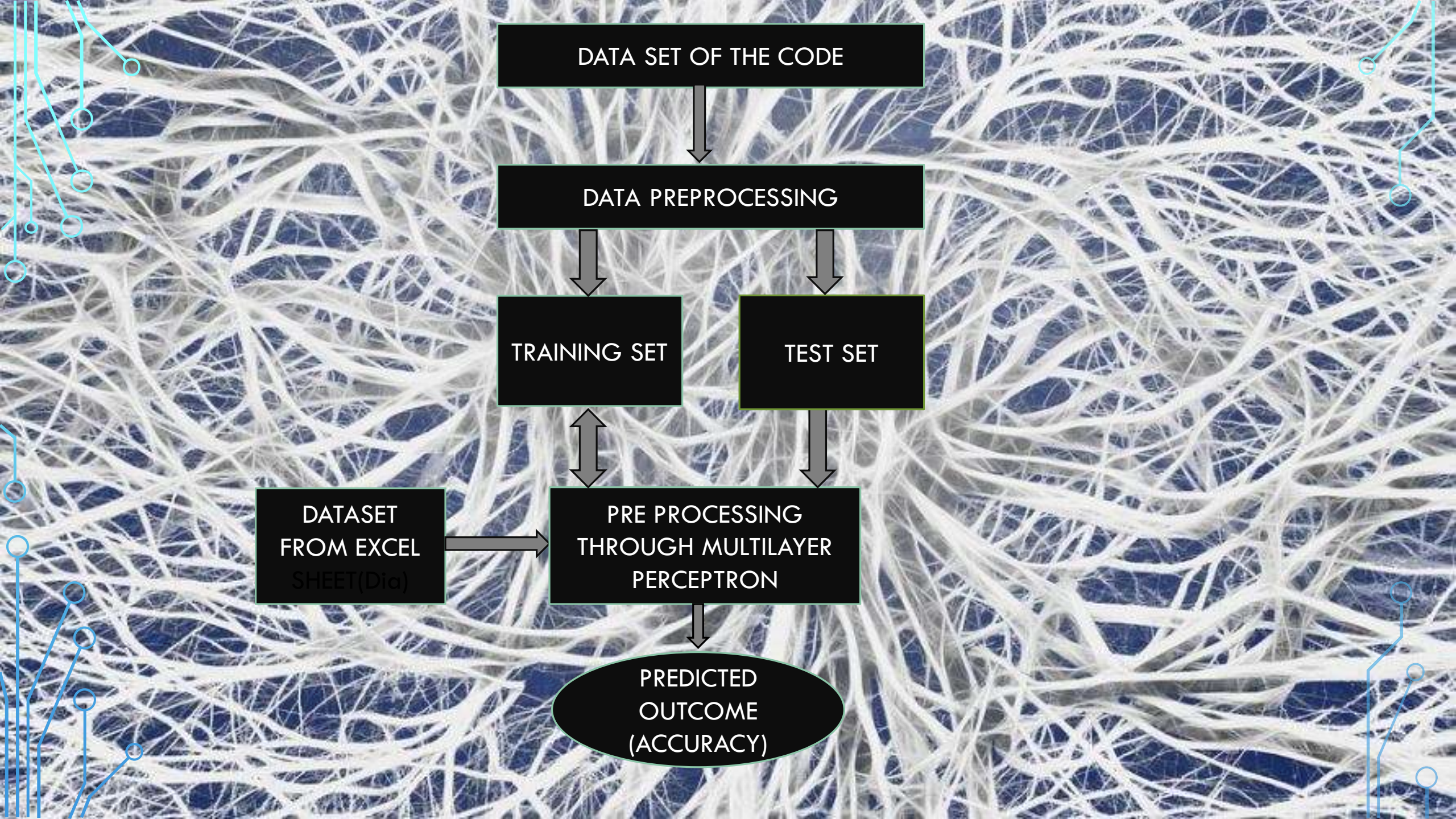
Mathematically, it can be represented as: f(x) = max(0,x)
Sigmoid is used as the activation function for the output layer.
Mathematically, it can be represented as: F(x) = 1/(1+e^-x)

SIGMOID FUNCTION



ReLU FUNCTION




DATA SET OF THE CODE
DATA PREPROCESSING
TRAINING SET
TEST SET
DATASET FROM EXCEL SHEET(Dia)
PRE PROCESSING THROUGH MULTILAYER PERCEPTRON
PREDICTED OUTCOME (ACCURACY)

REFERENCE

1. Anjali Negi and Varun Jaiswal, “A first attempt to develop a diabetes prediction method based on different global datasets.”

2. Jatin N Bagrecha, Chaithra G S and Jeevitha S, “Diabetes Disease Prediction using Neural Networks.”

3. Sumi Alice Saji and K Balachandran, “Performance Analysis of training algorithms of multilayer perceptrons in diabetes prediction.”

4. Talha Mehboob Alam, Muhammad Atif Iqbal, Yasir Ali, “A model for early prediction for diabetes.” 5. James Loy, “Neural Networks projects with Python.


THANK
YOU