FIRST ACTIVATION

```
#These are the imports that will be used in our diabetes AI

import numpy as NAEC
import pandas as Paladin
import matplotlib.pyplot as Paralyzer
from sklearn.model_selection import train_test_split
import tensorflow as Thor

Delta = Paladin.read_csv("diabetes.csv")

"""Delta variable will import the diabetes.csv file for training and testing"""
x_matrics = Delta[Delta.columns[:-1]].values
y_matrics = Delta[Delta.columns[-1]].values

"""Sklearn train_test_split is function that will help us to split the training features and  testing features.
test_size will use 50% of data in datasets for training & testing
random_state will also us to split every single time"""
Training_X, Temporary_X, Training_Y, Temporary_Y = train_test_split(x_matrics, y_matrics, test_size =
0.5, random_state = 0 )

"""Keras is an API that will help us build the Artificial Neural Network model
We used Dense NN model with 20 nodes and two activation functiin ReLU & Sigmoid"""
Ze_Model = Thor.keras.Sequential([
   Thor.keras.layers.Dense(20, activation = 'relu'),
   Thor.keras.layers.Dense(20, activation = 'relu'),
   Thor.keras.layers.Dense(1, activation = 'sigmoid')
])

"""This line of code will help us view the accuracy of the AI's correct prediction"""
Ze_Model.compile(optimizer=Thor.keras.optimizers.Adam(learning_rate=0.001),
            loss=Thor.keras.losses.BinaryCrossentropy(),
            metrics=['Accuracy'])

"""""This code initiate the training process and give us the accuracy"""
Ze_Model.evaluate(Training_X,Training_Y)

"""This will help improve the accuracy of the previous metrics"""
Ze_Model.fit(Training_X, Training_Y, batch_size=20, epochs=30, verbose=2, shuffle=True)
```

SECOND ACTIVATION

```
#These are the imports that will be used in our diabetes AI

import numpy as NAEC
import pandas as Paladin
import matplotlib.pyplot as Paralyzer
from sklearn.model_selection import train_test_split
import tensorflow as Thor

Delta = Paladin.read_csv("diabetes.csv")

"""Delta variable will import the diabetes.csv file for training and testing"""
x_matrics = Delta[Delta.columns[:-1]].values
y_matrics = Delta[Delta.columns[-1]].values

"""Sklearn train_test_split is function that will help us to split the training features and  testing features.
test_size will use 50% of data in datasets for training & testing
random_state will also us to split every single time"""
Training_X, Temporary_X, Training_Y, Temporary_Y = train_test_split(x_matrics, y_matrics, test_size =
0.5, random_state = 0 )

"""Keras is an API that will help us build the Artificial Neural Network model
We used Dense NN model with 20 nodes and two activation functiin ReLU & Sigmoid"""
Ze_Model = Thor.keras.Sequential([
   Thor.keras.layers.Dense(20, activation = 'relu'),
   Thor.keras.layers.Dense(20, activation = 'relu'),
   Thor.keras.layers.Dense(1, activation = 'softmax')
])

"""This line of code will help us view the accuracy of the AI's correct prediction"""
Ze_Model.compile(optimizer=Thor.keras.optimizers.Adam(learning_rate=0.001),
            loss=Thor.keras.losses.BinaryCrossentropy(),
            metrics=['Accuracy'])

"""""This code initiate the training process and give us the accuracy"""
Ze_Model.evaluate(Training_X,Training_Y)

"""This will help improve the accuracy of the previous metrics"""
Ze_Model.fit(Training_X, Training_Y, batch_size=20, epochs=30, verbose=2, shuffle=True)
```