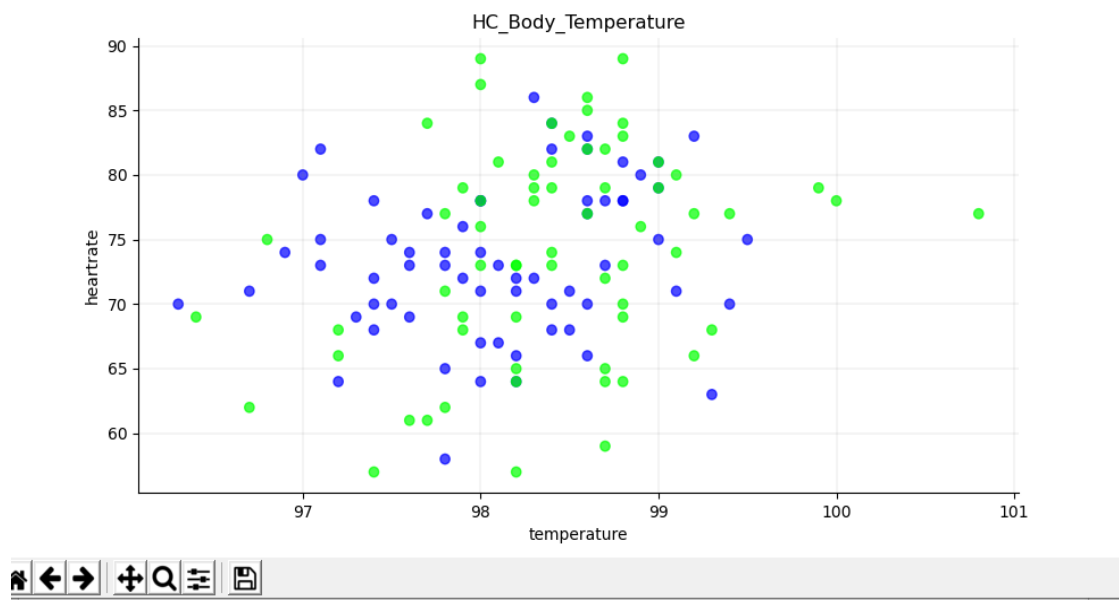


Machine Learning- AdaBoost

Below is the data represented by points when the X axis shows the temperature and the Y axis shows the heart rate. The purpose of the graph is to show an initial impression of the data and to understand what we do in the learning even before writing the code.

Blue-gender 1
Green-gender 2



The data shows in python using `import matplotlib.pyplot`

After writing the code ,we ran the code the rectangles with Adaboost algo.

Here are the results we got.(Honest moment, because the computer is i3 and it took too long to run the code on 100 round, we downgraded to 50.)

```
AdaBoost x
C:\Users\Shira\PycharmProjects\tirgull1\venv\Scripts\python.exe C:/Users/Shira/PycharmProjects/tirgull1/AdaBoost.py
-----rectangle-----
number of mistakes on T in round 1 : 0.2650769230769231
number of mistakes on R in round 1 : 0.25646153846153846

number of mistakes on T in round 2 : 0.26876923076923076
number of mistakes on R in round 2 : 0.2569230769230769

number of mistakes on T in round 3 : 0.2652307692307692
number of mistakes on R in round 3 : 0.2589230769230769

number of mistakes on T in round 4 : 0.29307692307692307
number of mistakes on R in round 4 : 0.2766153846153846

number of mistakes on T in round 5 : 0.26184615384615384
number of mistakes on R in round 5 : 0.24630769230769234

number of mistakes on T in round 6 : 0.24969230769230769
number of mistakes on R in round 6 : 0.2612307692307692

number of mistakes on T in round 7 : 0.27615384615384614
number of mistakes on R in round 7 : 0.25984615384615384

number of mistakes on T in round 8 : 0.26646153846153847
number of mistakes on R in round 8 : 0.25553846153846155
```

Here are the results we got for the circles with Ada boost algo.(Honest moment, because the computer is i3 and it took too long to run the code on 100 round, we downgraded to 50.)

```
AdaBoost <
-----circles-----
number of mistakes on T in round 1 : 0.2433846153846154
number of mistakes on R in round 1 : 0.24661538461538463
number of mistakes on T in round 2 : 0.24215384615384616
number of mistakes on R in round 2 : 0.24784615384615383
number of mistakes on T in round 3 : 0.24215384615384616
number of mistakes on R in round 3 : 0.24784615384615383
number of mistakes on T in round 4 : 0.24646153846153845
number of mistakes on R in round 4 : 0.24353846153846154
number of mistakes on T in round 5 : 0.2456923076923077
number of mistakes on R in round 5 : 0.2443076923076923
number of mistakes on T in round 6 : 0.24553846153846154
number of mistakes on R in round 6 : 0.24446153846153848
number of mistakes on T in round 7 : 0.23707692307692307
number of mistakes on R in round 7 : 0.2529230769230769
number of mistakes on T in round 8 : 0.24676923076923077
number of mistakes on R in round 8 : 0.24323076923076925
```

Attached the code in the Python on which we ran the staff.