**Introduction to** **Neural Networks - Dr. Amos Azaria**

**Mini-Project -Predict endometriosis part 2**Github: https://github.com/shani-co/-Deeplearning-Endo

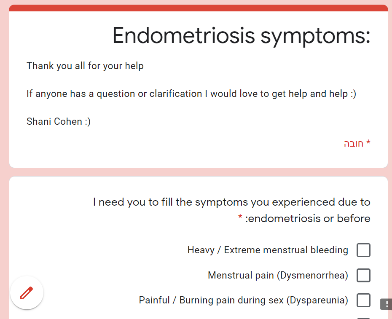
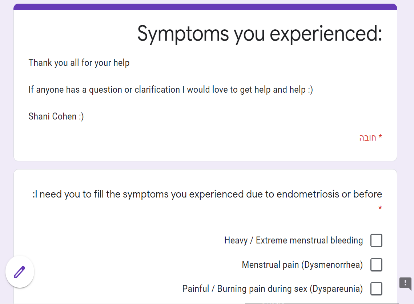
**Author**

**Shani Cohen -313246662**

**Naor Dahan-308399393**

**Dolev Hindi-312126642**

We are trying to predict a chronic disease called: endometriosis in order to facilitate the identification of the disease.

We created a data set with the help of 2 surveys, which were answered by different women.

**Data Set:**

Sick women (Label-1).

Unsick women (Label-0).

And 58 features (the symptoms are verity, and there are some that are necessary for endometriosis)

We combined this two data sets into one, and created a data set ,that contain the women who answered the 2 different surveys with the same features (symptoms)

Total: 474 with label 1

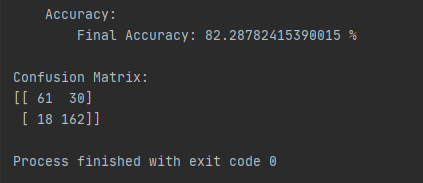
411 with label 0.

In all cases I divided the data as : 60% for the training and 40% for the testing

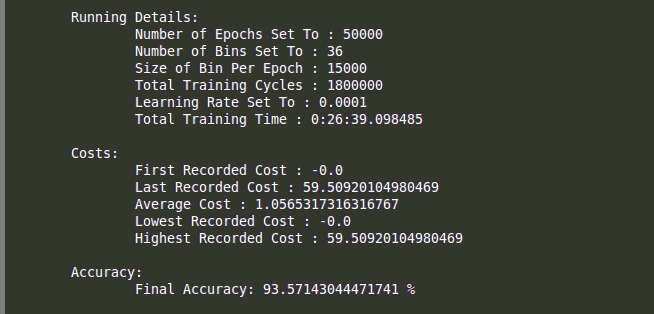
**Issus:**

\*The first e was the unbalanced data set -

1) When I used the entire data I noticed that the actual result guessing acts in a strange way.

 Despite the accuracy of the results was high but the false negative prediction was high ass well, it accures directly bacause of the ratio between the high amount of non recured cases to small amount of recured cases.

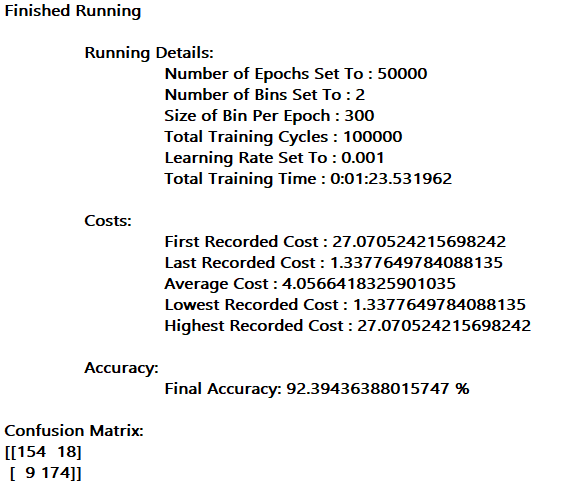
2) When I manualy balanced the data set by adding more unsick row to the data samples, the accuracy drops significantly but the ration between true positive, false positive, true negative and false negative were much more stable.

\*The secound issue was that, When I divided the data by bin\_size ,

Thare was a enormous drop in Loss after one run to -0.0.

In first phase I tried to solve this issue by adjusting two parameters (the learning rate and the bin\_size) with a succes. Then we realize that we chose too much data to check per each round.

**Logistic Regression Algorithm:**

****

\* weigths – adjusted by small random numbers, also checked the case when the weigths were initialized to 0’s , seems to make no difference.

\* learning rate – at the beginnig I tried with 0.00001 but when decreaced it to about 0.001-0.0001 the accuracy was better

\* iterations – started with just 10 but after increasing it to 100000 the accuracy got better.

**Conclusions:**

Our conclusions is that we reduced the alpha, the step was too small .and then our Lost did not decrease, so we increased it until the lust stabilized to a minimum and avoided reaching a Lost result : 0.

We have also get state of over fitting because the unbalanced data.  
Then we ran the machine on a new and balanced data set.

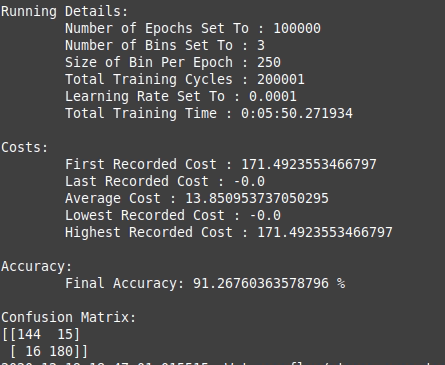
The implications are that the data set needs to be balanced so that the machine learns the data objectively and does not predict often and even when it does not need to "yes".

**MLP\_model Algorithm:**

**Issues:**

1) The first issue was that we didn’t know how much hidden level we need and wich size they need to be.

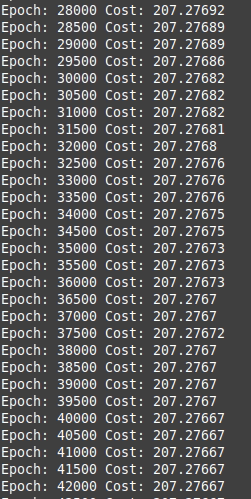
\*We did a trial and error this phase we decided to add 2 layers the first layer with 15 neurons and the second layer with 30 neurons based on a trial-and-error method.

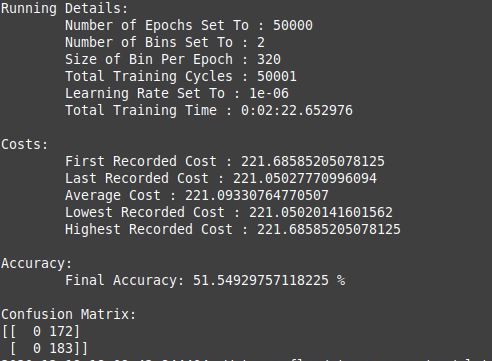


2) The second issue was that, When I divided the data by bin\_size ,

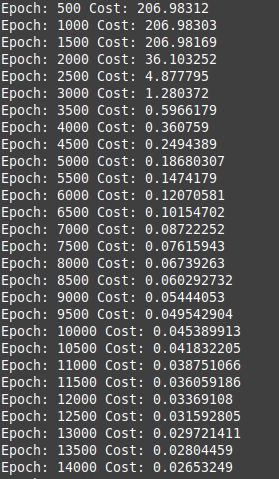
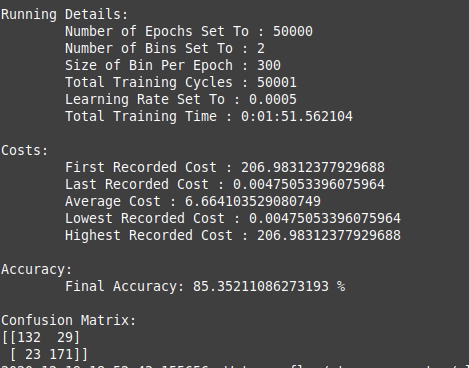
\*There was a enormous drop in Loss after one run to -0.0.

In first phase I tried to solve this issue by adjusting two parameters



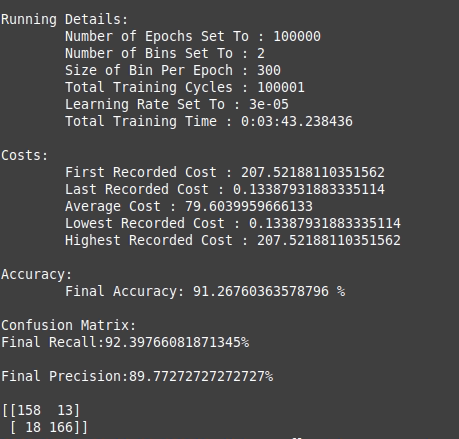
3) The three issue was that the loss didn’t go down and stayed on a fixed value.

\* the problem was caused by a very small alpha (0.00001) and the loss barely went down. After a few test we decided that 0.0003 is the oprimal alpha.



4) The fourth issue we had is a drastic drop of the loss

\*we solved that problem by modifying the alpha (the learning rate), we tried an alpha in range 0.001-0.0001 and we choose an alpha of 0.00003 since it gave us the best results

**Conclusions:**

Our conclusions is that after running the natural network model we saw an improvement in the performance of the model and in the Lost.

Where the lowest accuracy in the natural network was 91~%.

The improvement in the Lost is by 1.0~%.