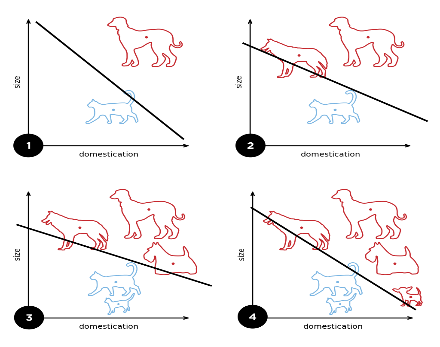
Bc"d

Submitted by Naor Eliav and Amir hoshen

***Perceptron algorithm***

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), the **perceptron** is an algorithm for [supervised learning](https://en.wikipedia.org/wiki/Supervised_classification) of [binary classifiers](https://en.wikipedia.org/wiki/Binary_classification). A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It is a type of [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier). A classification algorithm that makes its predictions based on a [linear predictor function](https://en.wikipedia.org/wiki/Linear_predictor_function) combining a set of weights with the [feature vector](https://en.wikipedia.org/wiki/Feature_vector).

This is an example of predicting whether the animal is cat or dog.

To learn and understand the algorithm I searched the internet and found a guide on the Udemi website and from there I also took the code and changed it. Attached is the source link: <https://www.udemy.com>. After learning the algorithm I worked on the given dataset. I saw that the data-set is not linear separated and as we learn that the perceptron doesn’t work well with such data. I understood that after plotting the the data on the graph. There was a problem with the data set that was missing information and I read that there are some solutions for that. 1. Deleting the row that contain the missing data. 2. Fill the missing data with the average of the future. I chose to delete the rows because there wasn’t much missing data. Then I normalized the data by division each row with the max value in the row so the values will be between 0 and 1. Another problem with the data- set was imbalance. To overcome this I looked in the web and saw that there are some solutions. One of them was to copy each Non-Recurring patient 3 times in order to balance the data.

Training proccess

The learning rate was set to 0.01, weights and bias were randomly chosen

between 0 to 1 using numpy and pandas in python.

Then we trained the model. The time of the training was very fast. It took 1.5 seconds. Its make sense because perceptron is single neuron simple model.

We devided the data to two classes: 66% for train and 33% for test.

we hava changed the choise of 33% randomly by choosing different start index each time.

The results were:

47.784545%, 54.5687% and 56.45454%.

The standart deviation is: **3.7230507930646**

|  |  |  |
| --- | --- | --- |
|  | Predicted: yes | Predicted: no |
| Actual: yes | True Positive (mean+/-std) | False Negative |
| Actual: no | False Positive | True Negative |

Ease:

In the algorithm there is an option to choose the learning rate( default: 0.01), the number of the iterations, the size of the train and test set.

Discussion:

As I assumed in advanced the perceptron algorithm insufficient when it comes to nonlinear separated data. When the data is separate the algorithm can predict with 85 % of success. I think also that the data set wasn’t satisfy because the imbalance.