Hands on Lab on Perceptron Algorithm

AGENDA OF THE VIDEO

- What is Perceptron Algorithm?
- What type of task does it solve?
- Concept of Neuron..
- What is an Artificial Neuron?
- How does the algorithm work?
- Pseudocode for perceptron algorithm
- Loan Default Toy Dataset
- How to plot the data in Matplotlib?
- Coding the algorithm from scratch
- Perceptron Class in sklearn
- Task For you……….

What is Perceptron Algorithm?

Perceptron algorithm is one of the early algorithms by Frank Rosenblatt in 1957. This algorithm is based on the artificial neurons. The basic idea behind the algorithm is motivated by the way biological neurons work in the human brain.

“Neuron Fires or not Fires Decide almost all the things that we do on a constant basis.”

What type of task does it solve?

There are many different types of tasks that are unsolvable by traditional Programming methods.

Machine Learning offers various algorithms to solve these tasks.

The perceptron algorithm in particular solve the task of Binary Classification.

Concept of Neurons

Neurons are a Biological Units in a Human Brain which decide various functions in Human Body.

Essentially, what we need to understand is whether the neuron is firing or not.

Every Human action or function in the body, can be traced back to firing of these neurons. They are billions in number.

Concept of Artificial Neurons

So, Can we create or somehow mimic the neuron function in real world?

If we can, then that would be called as Artificial Neurons….

Essentially, what we want to create is a mechanism which takes some input and produces some output. Something happens inside the neuron to mimic whether it has fired or not.

Concepts

Given (X,y) where y is of discrete categorical type (Binary).

Make a Decision Rule based on X to predict y.

For example:
Rule:
Predict “Spam” if w.x > 0
“Ham” if w.x <= 0

How does the algorithm work?

The idea is that we have some inputs and we want to learn (find) some weights.
So, we start with some random weights and loop through the training data set and do the following:

1. Get Prediction based on the Input and the current_weight.
2. Update the weight by doing:
    a. weight += (actual - predicted) * Learning_Rate * Corresponding_x

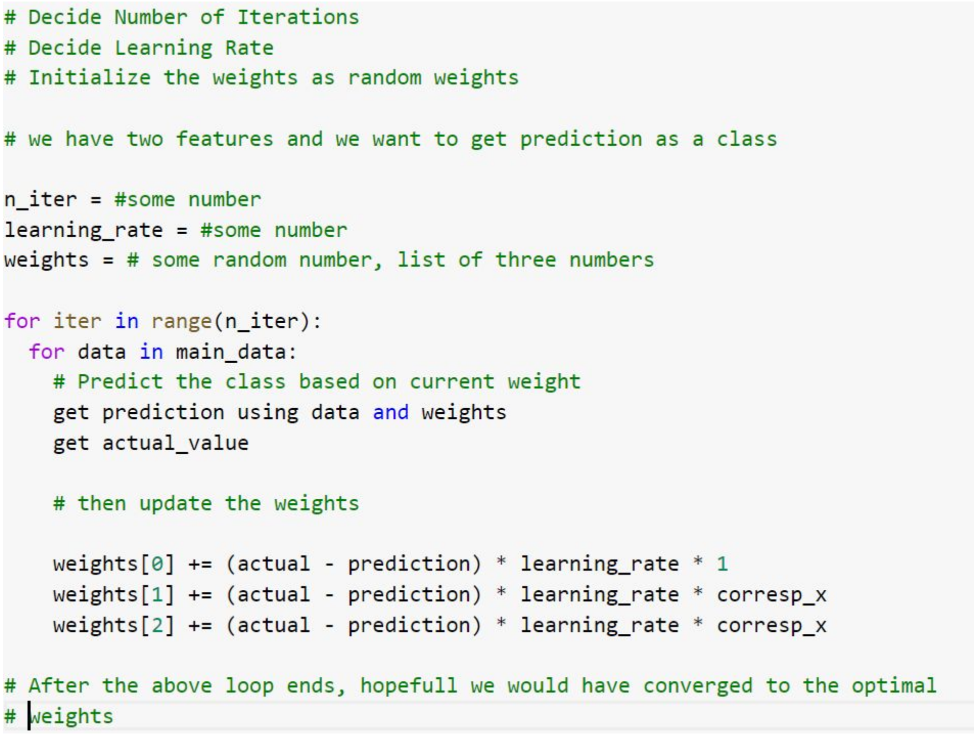
Why does the weight update make sense?

The weight update can be understood quite intuitively. It is one of most beautiful results in machine learning.

When the actual and predicted values are same, the weights will not get updated. The reason for this is that (actual - predicted) will become zero.

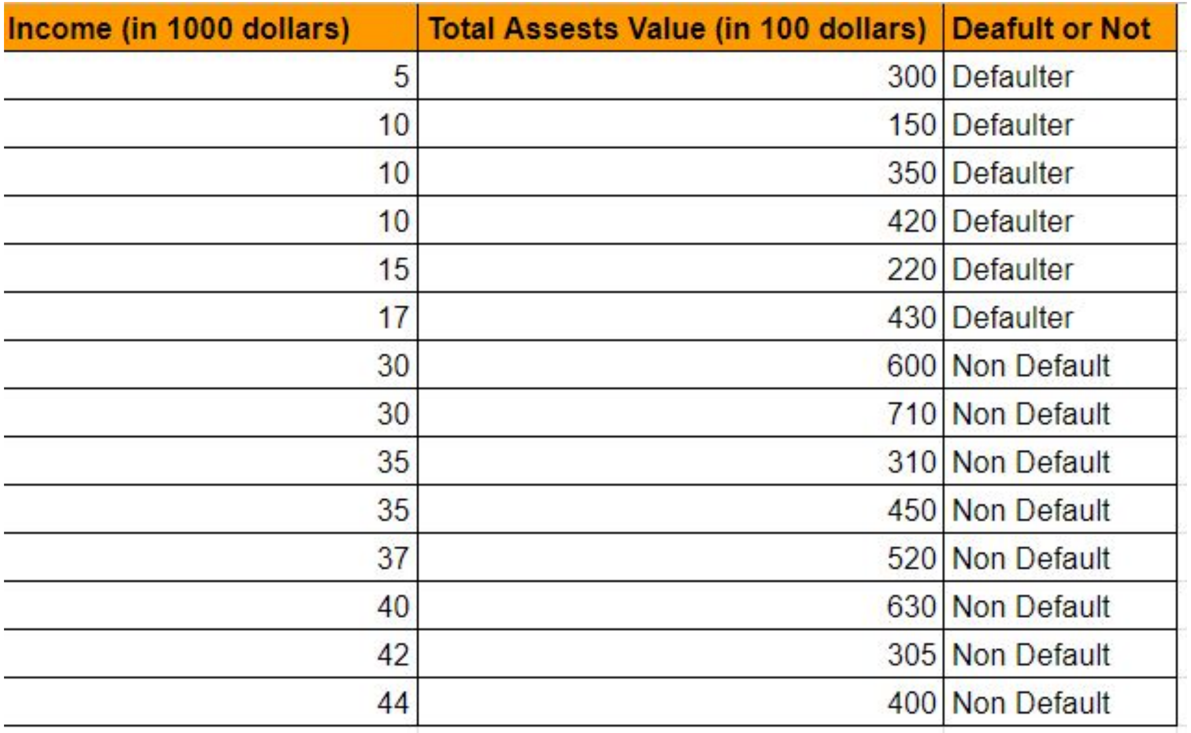
Now, the weights will be updated only in other situations where the actual and predicted classes differs.

Pseudocode for Perceptron Algorithm



```
# Decide Number of Iterations
# Decide Learning Rate
# Initialize the weights as random weights

# we have two features and we want to get prediction as a class

n_iter = #some number
learning_rate = #some number
weights = # some random number, list of three numbers

for iter in range(n_iter):
  for data in main_data:
    # Predict the class based on current weight
    get prediction using data and weights
    get actual_value

    # then update the weights

    weights[0] += (actual - prediction) * learning_rate * 1
    weights[1] += (actual - prediction) * learning_rate * corresp_x
    weights[2] += (actual - prediction) * learning_rate * corresp_x

# After the above loop ends, hopefull we would have converged to the optimal
# weights
```

Loan Default Toy Dataset



Income (in 1000 dollars)	Total Assests Value (in 100 dollars)	Deafult or Not
5	300	Defaulter
10	150	Defaulter
10	350	Defaulter
10	420	Defaulter
15	220	Defaulter
17	430	Defaulter
30	600	Non Default
30	710	Non Default
35	310	Non Default
35	450	Non Default
37	520	Non Default
40	630	Non Default
42	305	Non Default
44	400	Non Default

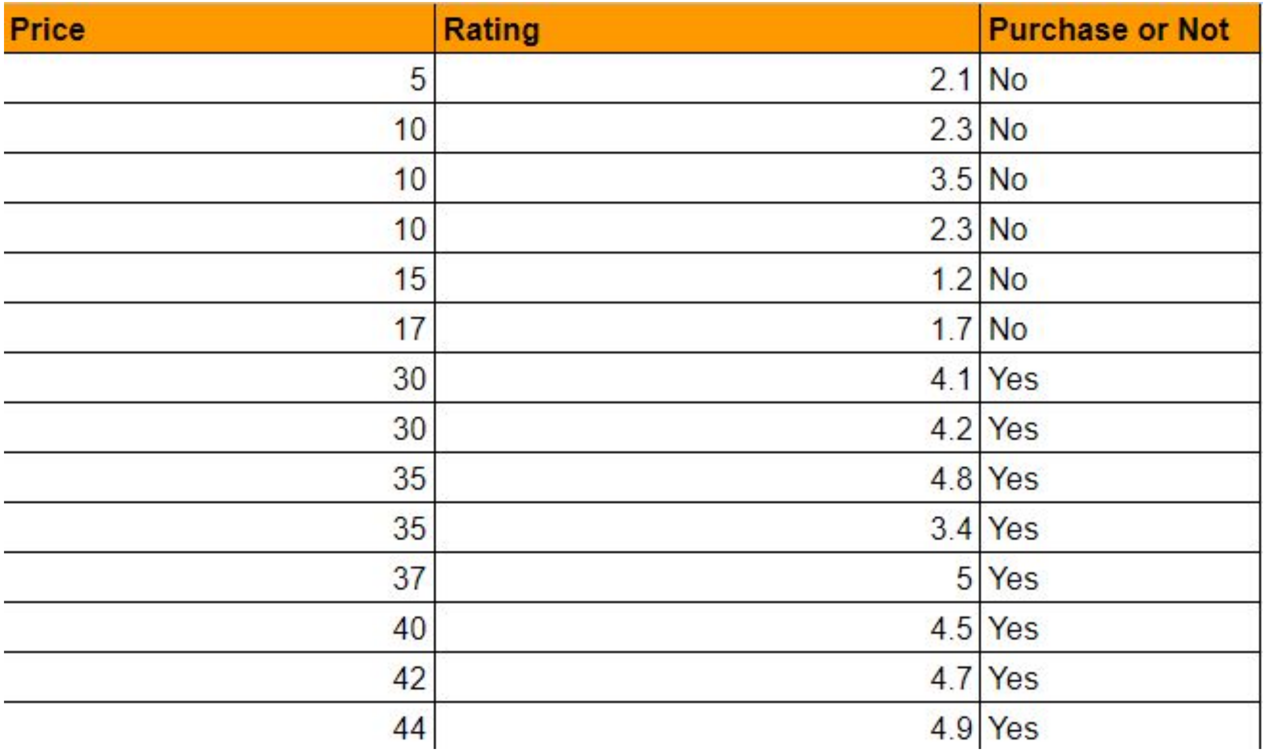
How to plot the data in Matplotlib?

Coding the algorithm from Scratch in Python

Let us now see Perceptron Class in Sklearn in action

Task For you...

Build a Perceptron Classification Model to predict whether an Individual (say A) is going to Purchase a Product with a given Price and Rating or not.



Price	Rating	Purchase or Not
5	2.1	No
10	2.3	No
10	3.5	No
10	2.3	No
15	1.2	No
17	1.7	No
30	4.1	Yes
30	4.2	Yes
35	4.8	Yes
35	3.4	Yes
37	5	Yes
40	4.5	Yes
42	4.7	Yes
44	4.9	Yes

Thank You