

# Lab 3 – Perceptron

MACHINE LEARNING

SAUMAY AGRAWAL

16BCE1151

---

## EXPERIMENT

- Implement the perceptron algorithm.
- Test the implementation using Iris Dataset.
- Redo the experiment with sklearn and for your own dataset. Compare the performance of the implementation and for the different datasets.

## ALGORITHM

The perceptron algorithm is analogous to the biological mechanism of the neurons in our mind. Neurons transmit data in the form of signals across our brain, and a network of neurons work together while we are making a decision. A bigger and more complex network of neurons gets involved in more complex decision making.

The perceptron algorithm works on a similar basis. The input values are multiplied by a weight and a scalar value is achieved after including bias, to get the predicted results. The difference between predicted and expected results is fed to a step function or activation function, which gives the result as 0 or 1, based on the relative position of difference with respect to the threshold value. If the prediction was wrong, then the weights are changed accordingly by multiplying a learning factor. The learning factor decides the change in the magnitude of weights, and is not learned by the model. It is instead given by the user.

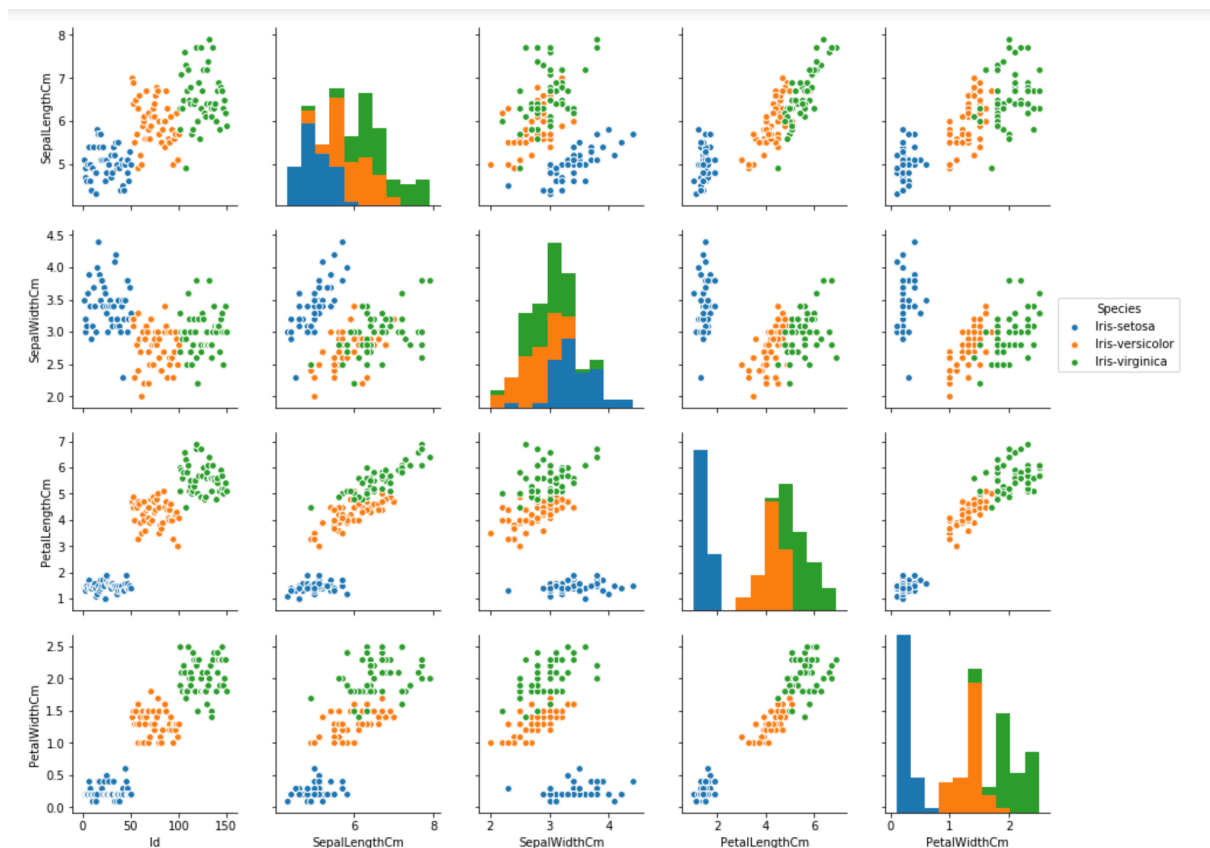
When the model has repeated this process on all the rows, an epoch is said to be completed. The process is repeated over multiple epochs, to make sure that the model gets adopted appropriately with respect to all the input attributes. This prevents the overfitting and underfitting of the model, ensuring maximum accuracy that can be achieved for given input attributes and expected attribute.

The perceptron model helps in binary classification. It creates a decision boundary dividing the data into two classes. Thus, it works best when the data is linearly separable. The accuracy of the model generated is directly proportional to the extent to which the data is linearly separable. However, if the dataset is not linearly separable, then multiple layers of perceptrons can be combined together for decision making, which then constitutes of a neural network.

## OBSERVATIONS

### IRIS DATASET

- The dataset can be classified easily, as we have seen earlier. The classes 'setosa' is completely linearly separable from the classes 'versicolor' and 'virginica'. However, the classes 'versicolor' and 'virginica' overlap a bit. So the accuracy of prediction may get reduced.

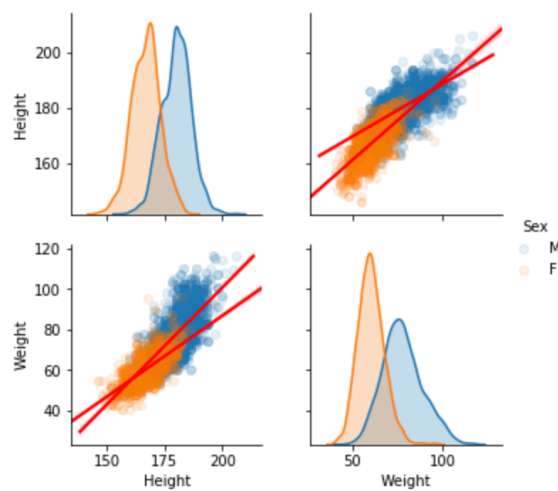


- Upon applying the perceptron model for the target classes 'versicolor' and 'virginica', for the input attributes 'petal length' and 'petal width', the accuracy of the trained model was found to be '1.00' surprisingly. We also found out that bias = 5000,  $w_1 = 27760$ ,  $w_2 = 10130$ .

### OLYMPICS DATASET

- Upon pair plotting, I found out that it was difficult to get the data as linearly separable, as there was a wide range of values in these attributes, for all the entries. So I tried the pair plotting on various categorical attributes, like Sports, Medals, and Sex.
- The Sports plot had way too many classes, and there wasn't a clear distinction between the classes in the other two plots. So I decided to classify on the basis of Sex (as it is binary in this dataset), by reducing the dimensions of the dataset.

- According to the structure of this dataset, the appropriate input attributes for the perceptron model seems to be
  - Age
  - Height, and
  - Weight
- I chose the height, weight and sex attributes of the athletes, and reduced the dimensions of dataset on various parameters. Upon a lot of trial and error, I found out the data which was linearly separable up to some extent. So, I decided to use the height and weight values to predict the sex of an athlete (who participated in Olympics 2014).



- Some algorithms (like decision trees) can work fine with the categorical data. However, the perceptron algorithm need its data to be numeric. For this sole purpose, we convert the values of the attribute sex, using the Label encoding.
- Then we initialize our perceptron for this sub-dataset, with number of epochs = 1000 and learning rate = 0.1.
- Upon comparing the predicted and expected values, we got the following accuracies of our model on different subsampled datasets.
  - 0.87 on the 20 years dataset.
  - 0.75 on the 40 years dataset.
  - 0.79 on the whole 120 years dataset.

This accuracy is justified, as it can be seen that the values are not completely linearly separable in the above graph.

## INFERENCE

- The perceptron algorithm is good for binary classification, where the data is linearly separable upto some extent.
- If the data isn't linearly separable, like in this case, a single perceptron fails. This is the case with most of the real world data. So this model is not suited for such data.

- For such purposes, we might need to construct multiple layers each consisting of multiple perceptron, giving rise to a neural network. An example is given below.

