# **ML Assignment 2**

**This assignment is done in pair.**

**Partner 1: Ankit Juneja (2221327), 1CSD (Fully responsible for Perceptron algorithm and comparison with inbuilt perceptron function, along with perceptron graphs)**

**Partner 2: Aditya Pimpalkar (22226150) ,1CSD (Fully responsible for multi-layer perceptron algorithm and comparison with inbuilt perceptron function, along with multilayer perceptron graphs)**

**Description of Perceptron Algorithm:**

Perceptron algorithm is a type of neural network which comes under the supervised learning, also perceptron is known as single-layer neural network. The main objective of this algorithm is to analyse the linearly separable objects with binary outcomes, Initially, it does not contain any recorded data but later it makes the sum of all inputs and weights. Therefore, If the total sum of all inputs is more than a pre-determined value, the model gets activated.

**MLPerceptron Algorithm:**

**Step1:** Import the necessary libraries.

**Step2:** Read the data and store in variable

**Step 3**: Apply data pre-processing technique MinMax scaling to scale the data.

**Step 4**: Replace the ‘yes’ and ‘no’ value with 1 and 0 in dependent variable for better calculations and separated the dependent variable from other features.

**Step 5:** Created **MLPerceptron** Algorithm: *It consist of two main methods* ***fit*** *and* ***predict****.*

Fit Method:

* Fit method take input only X as independent features and y as dependent features It has fixed value of learning rate 0.01 and number of Iterations is 1200.
* Initially an array created of 9 weights and each weight assign as value 0. Also, Bias is taken as 0.
* Initialize for loop to get maximum accuracy of weights and bias by updating their values with a difference of predicted dependent variable and actual dependent variable.

Predict Method:

* Created empty an array.
* Calculate dependent variable prediction by using weight and bias that already set in fit method.
* Store each value into an empty array to get all dependent variables.

**Step 6:** Compare the accuracy of MLPerceptron algorithm vs Inbuilt perceptron function by passing 5 times different random shuffled data, also represented through various graphs.

A picture containing chart

Description automatically generated

**Decision Designs of perceptron Algorithms:**

**Input:** In our dataset we have total 9 features, so as an input we take all these features.

**Weights and Bias:** Initially, weights and bias are assigned with 0. Later it will update in each iteration with the help of learning rate.

**Learning Rate:** It plays an important role in terms of accuracy in this algorithm. It is fixed with 0.01 in LR variable.

**Iterations:** It helps to change the weights till convergence, in this algorithm total iterations are 1200 predicted and it stored in N\_iters variable and it try to iterate till convergence.

**Activation function:** It is responsible to give the binary output (0 and 1), If the input value of function is greater than the threshold value then it gives 1 else it gives 0. In my algorithm instead of creating separate activation function I code from scratch to fulfil the same purpose.

**Test and Results:**

In perceptron algorithm, I split my data into multiple ratios train and each time I shuffle training data to get different values, till 5 times it repeats the process. There is a graph that represent the accuracy of each time in both perceptron algorithm and inbuilt perceptron function from scikit-learn.

Chart, line chart

Description automatically generated

Also, there is another graph (Graph 1) which represent the values of dependent variable (Fire and No Fire) between predicted by perceptron and test data of dependent variables.

Moreover, one more similar graph (Graph 2) which represent the values of dependent variable (Fire and No Fire) between predicted by Inbuilt function and test data of dependent variables.

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Below is the ROC graph that represent the prediction for both Perceptron algorithm and inbuilt function of perceptron from scikit library, also it has observed that accuracy achieve by perceptron algorithm is similar to inbuilt function.

Chart, line chart

Description automatically generated

**Conclusion:** It is crystal clear that this single layer perceptron algorithm is recommended for linear datasets and activation function plays an important role in this algorithm to predict the values.

**Observations:** It is clearly observed that perceptron algorithm is totally based upon weight, bias and learning rate, the more accurate these variables are the more accurate prediction it reflects. Also, in comparison of inbuilt scikit function the accuracy of MLPerceptron algorithm is similar but sometimes it depends upon training data as well.

\_Ankit Juneja (22221327)

--------------------------------------------------------------------------------------------------------------------------------------

**Multilayer perceptron:**

An artificial neural network or multilayer perceptron are the most useful types of neural networks which are used to solve challenging computation tasks like predictive modelling in machine learning. This kind of neural network learns mapping of the data it’s provided and how to represent the output variable you want it to predict. MLP model has an Input layer, hidden layer and an output layer.

**My model design: -**

**Input layer** = 9 | **Hidden layer** = 10 | **Output layer** = 1

Since I have 9 features in our dataset, I assign the neurons to 9. For our deep learning I took hidden layers as 10. Our output layer is 1 because we are classifying data as 1 or 0. **Bias** value for both hidden and output layers have been taken as -1 along with starting weights which have been assigned a random value between -1 and 1. **Learning rate** which controls how quickly the model is adapted to the problem – is set to 0.04 -1 on both layers. **Tanh** was used for **activation function**. 1000 **Epochs** are taken

Scikit-learn was used as a reference implementation to compare results against my own algorithm.

**Tests and results**

For MLP, I have allocated the data as 80% to training and 20% to testing and shuffled it through 5 different iterations, ran these datasets through my algorithm written from scratch along with the MLPClassifier of Scikit-learn.

Chart, line chart

Description automatically generated

The line graph above, shows us the accuracy scores across five different dataset iterations. We see my implementation (blue line) showing the results almost the same as scikit-learn but mostly just a tad bit low accuracy than the reference implementation (orange line).

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

The two bar plots above, shows us the scores of how accurately the models predicted against test data over five different dataset divisions. The first bar chart is my implementation’s prediction scores and the second is the scikit-learns predictions. We can see that only on some iterations, my algorithm is matching the results of the scikit library. Can see the accuracies matching the most in the fifth iteration.

Line chart

Description automatically generated with medium confidence

The above graph shows the ROC curve of both algorithms, we can see that they have almost the same curve along the positive and negative rate.

**Conclusion:**

We can see my algorithm has been performing almost as well as scikit-learns. We can get a better accuracy result by changing the bias range, the learning rate or change the epochs. Changing range of the weights can also be a contributing factor to reducing the error rate and getting a better accuracy.

\_Aditya Pimpalkar (22226150)

--------------------------------------------------------------------------------------------------------------------------------------