MULTI-LAYER PERCEPTRON

LAB 6

Multi-layer perceptron theory:

Multi-layer perceptron is a part of "feedforward" neural networks. It consists of Atleast 3 layers of neurons – Input layer (no. of neurons = input features + 1 for bias), hidden layers (can be one or multiple) and output layer (no. of output neurons is equal to the number of output classes). The MLP uses the backpropagation technique for training.

There are a few thumb rules to decide on the number of hidden layers and size of each hidden layer. Usually, even a single hidden layer is sufficient for majority of the problems. In this case the number of neurons in the hidden layer should be equal to the mean of the number of neurons in the input and output layer. If multiple hidden layers are used, the total number of neurons in the hidden layer should be lesser than the total number of (input + output) layer neurons.

PART 1: BANKNOTES Dataset

First, we visualized the dataset by plotting pair-plots among the 4 attributes. It was clearly visible that the points are not linearly separable and thus we would have to employ MLP. To implement the multi-layer perceptron, MLPClassifier was imported from sklearn.neural_network. Next, the dataset was split into X_train, X_test, y_train and y_test.

Type 1 implementation involved using the 3 different solvers – sgd, adam and lbfgs for a single hidden layer containing 4 neurons.

Type 2 implementation involved using multiple hidden layers with varying number of neurons per layer. There were two hidden layers having sizes 3 and 3. This again was implemented for all the three solvers. The details are in the tables that follow.

TYPE	SOLVER	NO OF NEURONS IN HIDDEN LAYER	NUMBER OF WRONGLY CLASSIFIED POINTS
1	sgd	4	141
1	adam	4	16
1	lbfgs	4	0
2	sgd	3, 3	2
2	adam	3, 3	186
2	lbfgs	3, 3	0

Thus, we observe that 'lbfgs' solver performed better than the other two.

PART 2: ABSENTEEISM FROM WORK Dataset

From the previous lab implementations on this dataset, it was found that plot between "weight and body mass index" with hue set to education, is not linearly separable. Hence it was a fit case to employ MLP. The data for the 3 classes – graduates, postgraduates and doctorates were obtained and the labels for the classes were set to 1,2 and 3. Method 1 was done with respect to the book's algorithm. It showed us the plot between the cost and epochs. However, the accuracy was low for both training (62.2%) and testing (58.9%) data. Next, we used Sklearn's MLPClassifier. Like the banknotes dataset, the implementation was done in two types.

Type 1 implementation involved using the 3 different solvers – sgd, adam and lbfgs for a single

hidden layer containing 13 neurons.

Type 2 implementation involved using multiple hidden layers with varying number of neurons per layer. There were three hidden layers having sizes 7, 5 and 4. This again was implemented for all the three solvers. The details are in the tables that follow.

TYPE	SOLVER	NO OF NEURONS IN HIDDEN LAYER	NUMBER OF WRONGLY CLASSIFIED POINTS
1	sgd	13	6
1	adam	13	4
1	lbfgs	13	4
2	sgd	7, 5, 4	7
2	adam	7, 5, 4	8
2	lbfgs	7, 5, 4	0

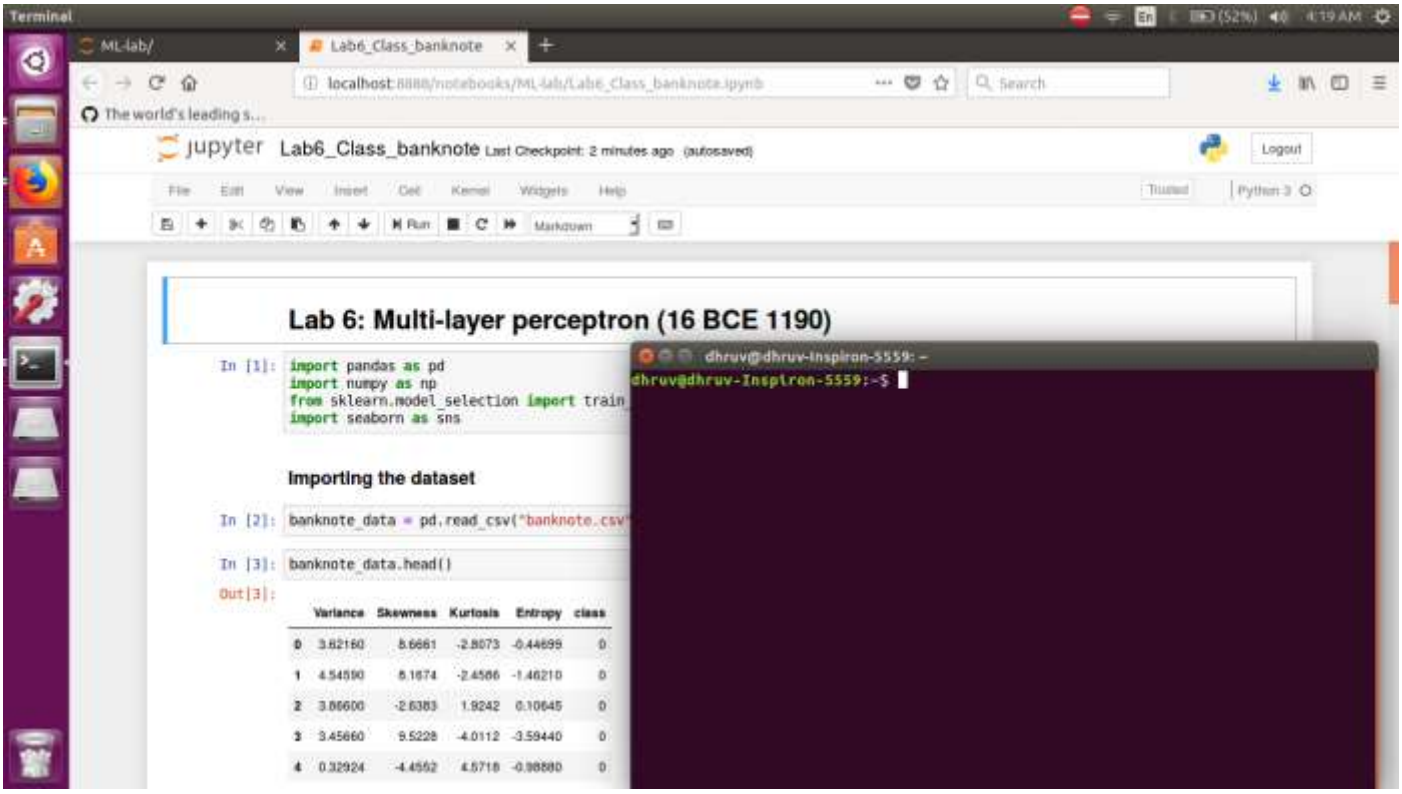
Thus, we observe that in Type 1 implementation, 'adam' and 'lbfgs' performed equally well. However in Type 2 implementation the 'lbfgs' solver performed better than the other two.

PART 3: TENSORFLOW PLAYGROUND

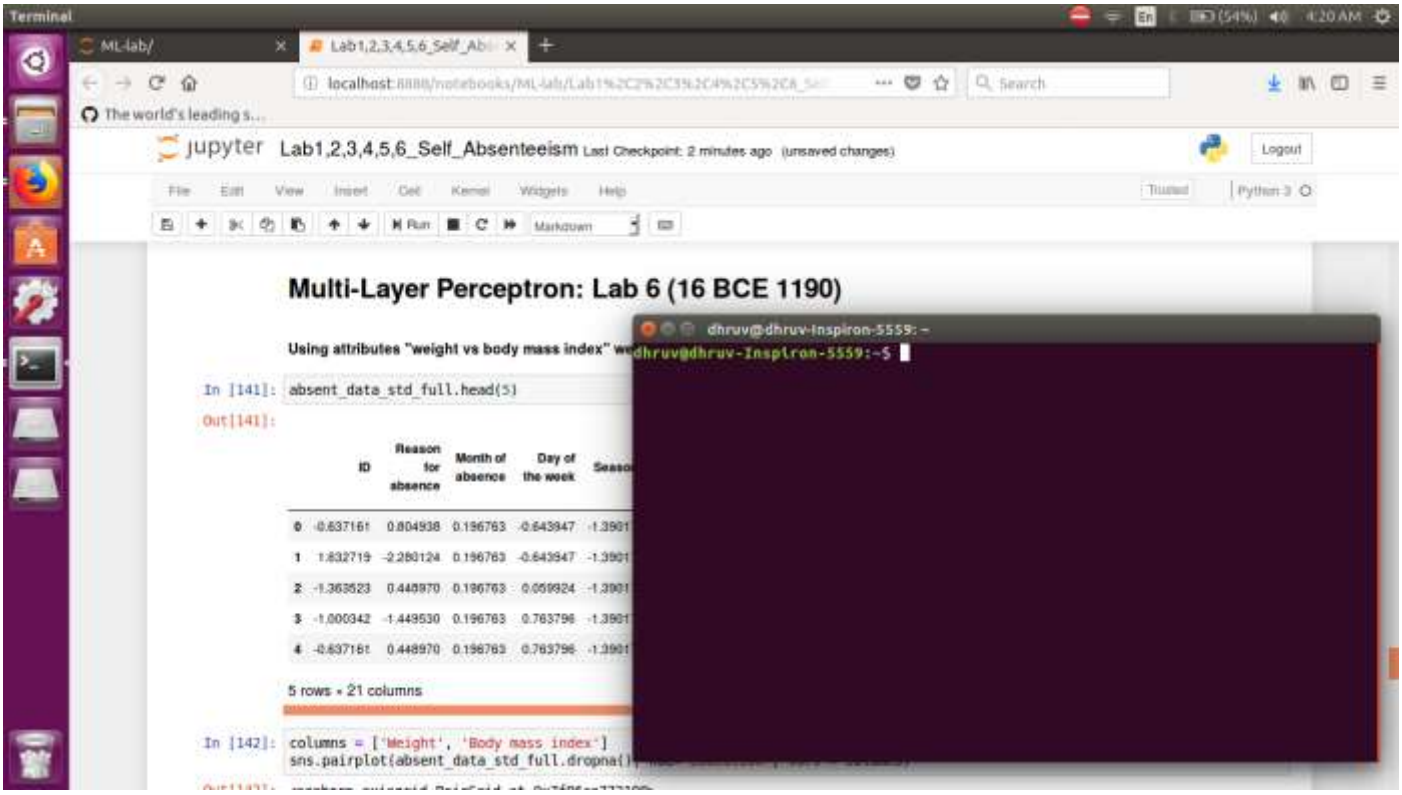
As observed from the "Spiral" pattern in the Tensorflow Playground, the limit for multi-layer perceptrons was tested. To get the pattern correctly, multiple features, multiple hidden layers were used and it was not easy to clearly understand which combination would give the correct pattern. Thus, with very complex models like these, the understandability of the model goes down.

SCREENSHOTS

Banknotes-dataset



Chosen dataset – Absenteeism at work



Tensorflow playground