2 Multi Layer Perceptron Classification

In this task, we implement MLP classification from scratch
using numpy, pandas and experiment with various activation functions
and optimization techniques, evaluate the model's performance, and draw comparisons
with the previously implemented multinomial logistic regression.

Arnav Negi

Model Implementation

We have a Multi Layer Perceptron classifier class with the following hyperparameters:

- hidden layers: describes architecture of MLP
- epochs
- learning rate
- activation function: [sigmoid, tanh or relu]
- optimizer : [Batch, SGD or mini batch]

We use the Wine Quality dataset as in Task 1.

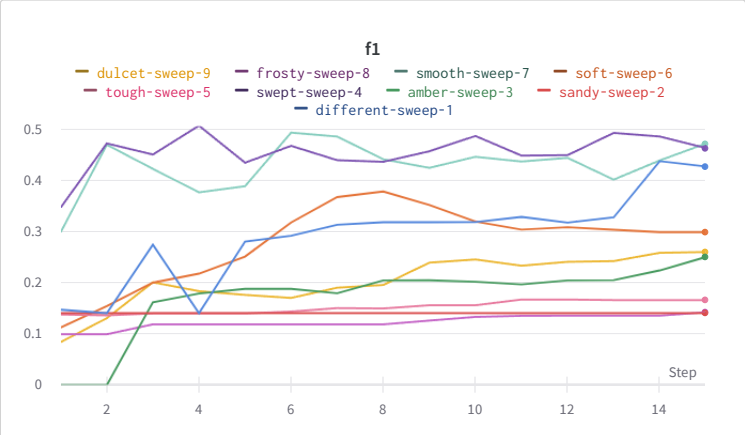
Model hyperparameter tuning

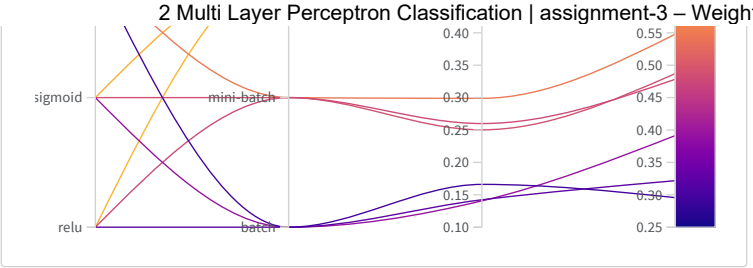
We tune the hyperparameters for maximum f1 score

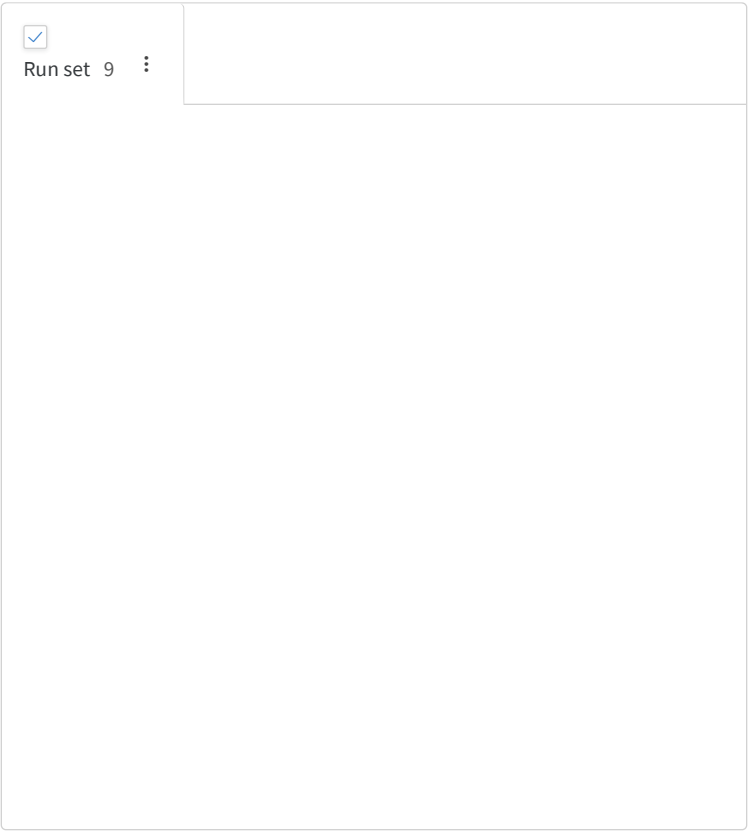
We track the following metrics on val set

- Loss
- Accuracy
- F1 score
- Precision
- Recall

First we tune for activation and optimizer, keeping rest fixed:



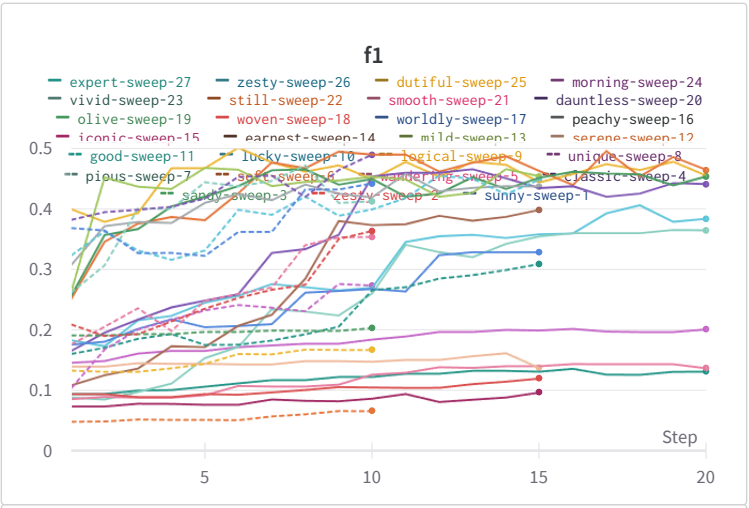


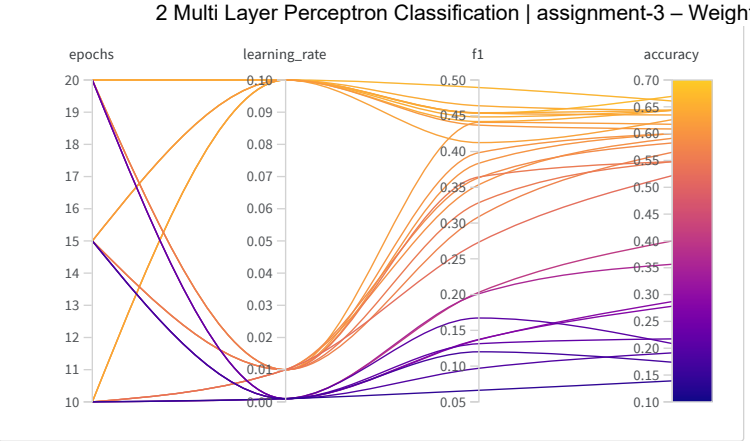


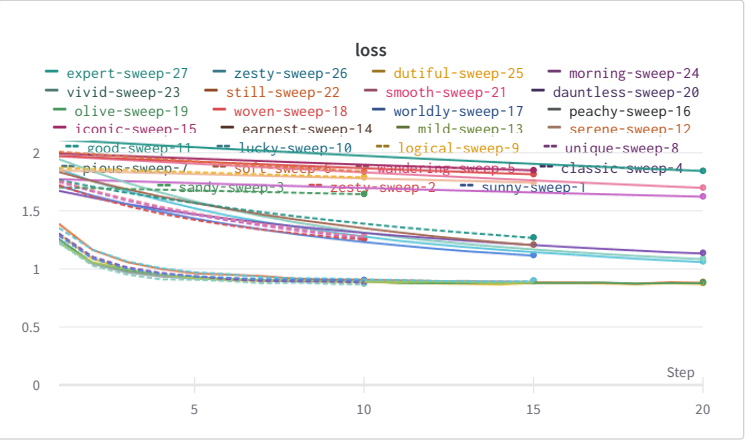
As we see the best performing model had 0.4717 f1 score and 62.61% accuracy

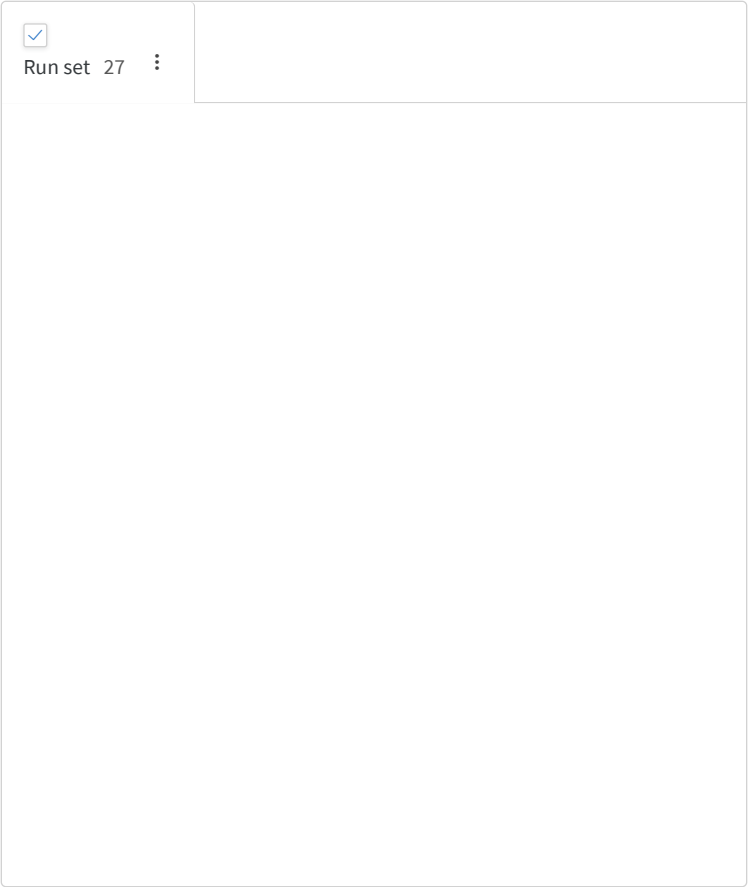
- activation = tanh
- optimizer = SGD

Next we tuned epochs, learning rate and hidden layers. Hidden layers were chosen from [10, 10], [20, 20] and [30,30].









Here we can see the best performing model had 0.4895 f1 score and 66.09% accuracy

- epochs = 20
- learning rate = 0.1

- hidden layers = [20,20]

Comparison with Multinomial Regression

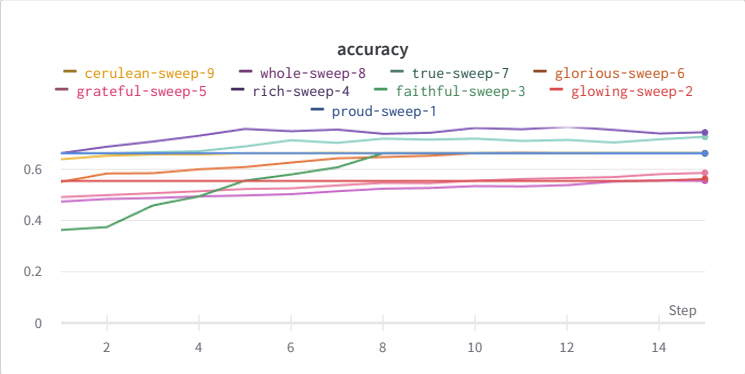
With the right hyperparameters, MLP models beat Multinomial regression on this task. Furthermore, given more data MLP models have more parameters to train on and hence can adapt better than simple multinomial regression.

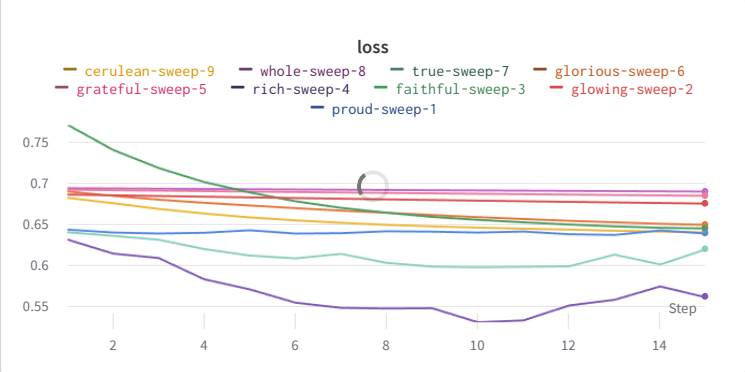
Multi Label classification with MLP

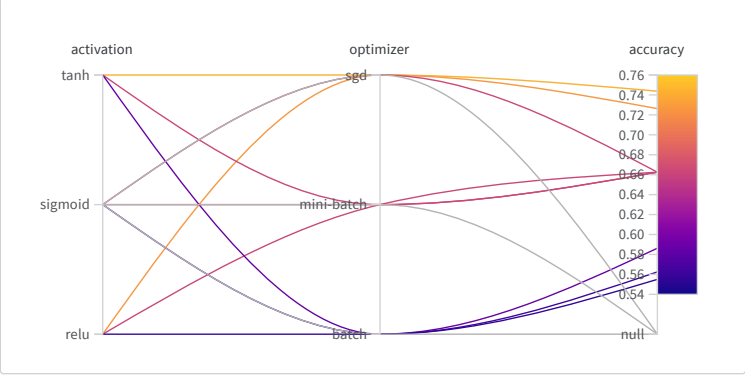
We modify our MLP class for multi label classification task. Then apply it on the customer's dataset.

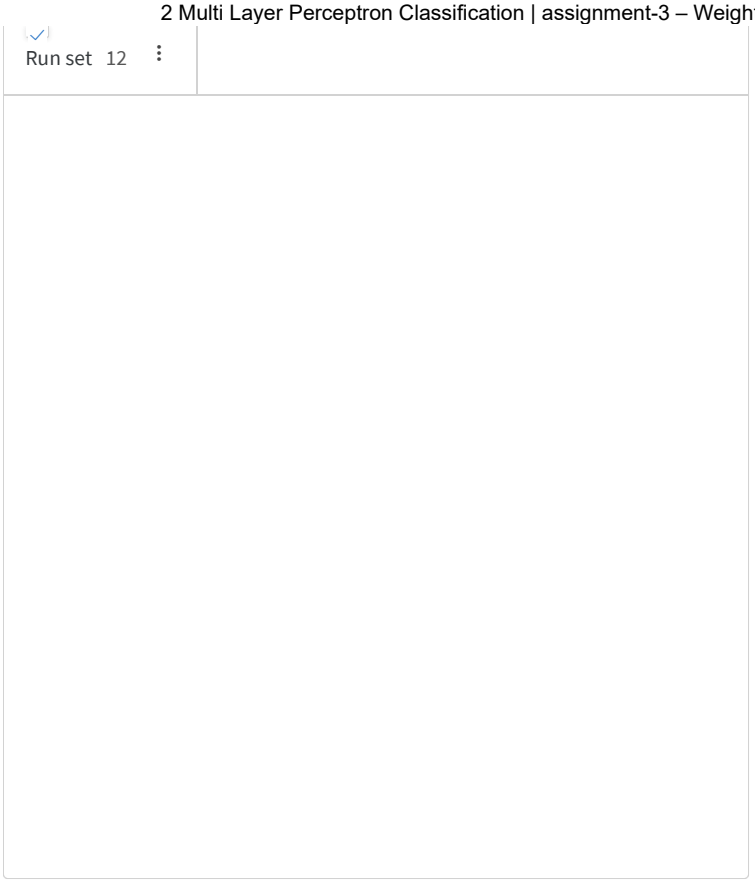
The hyperparameters stay the same. We tune them as follows.

First we tune for activation and optimizer, keeping rest fixed:





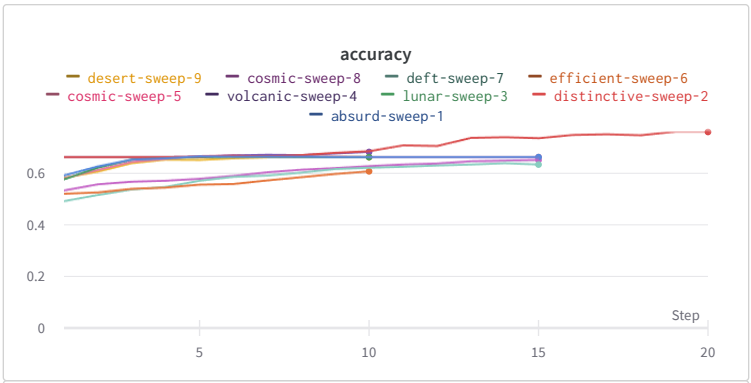


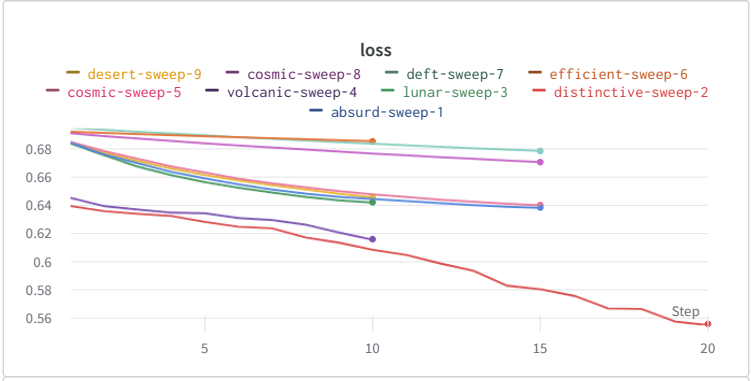


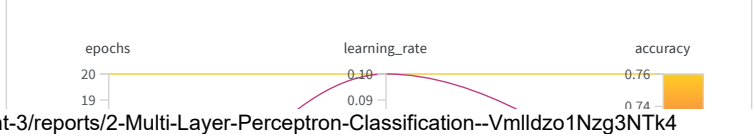
This shows the best model has 74.38% accuracy and

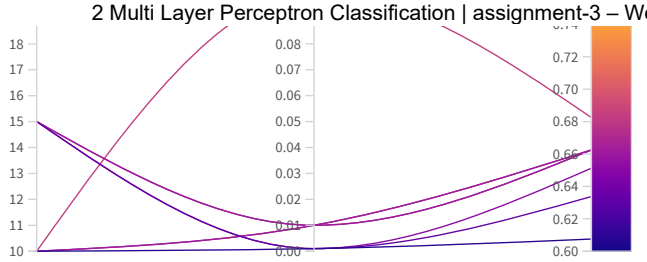
- activation = tanh
- optimizer = SGD

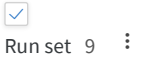
Next we tuned epochs, learning rate and hidden layers. Hidden layers were chosen from [10, 10], [20, 20] and [30,30].











This shows the best model has 76% accuracy and

- epochs = 20
- learning rate = 0.1
- hidden layers = [20, 20]

Created with ❤ on Weights & Biases.

https://wandb.ai/arnav-team/assignment-3/reports/2-Multi-Layer-Perceptron-Classification--Vmlldzo1Nzg3NTk4