

CS 165B – Machine Learning, Spring 2023

Machine Problem #3 Due Saturday, June 3 by 11:00pm

TA: Danqing Wang

Email: danqingwang@ucsb.edu

Note: This assignment should be completed individually. You are not supposed to copy code directly from anywhere else, and we will review the similarity of your submissions. You should write the code from scratch, and you **are not allowed** to directly use any third-party tools such as sklearn. You can use any default Python libraries and the imported packages at the beginning of *mp3.py*

In this project, you should **start from scratch** to implement one perceptron. Remember to keep the function name and argument of *run_train_test(training_data, testing_data)* unchanged.

As MP2, we use **Pandas** library to deal with the data.

Problem Overview:

Change the **Python 3** program *mp3.py* that creates a perceptron classifier for the binary classification problem. The training and testing data sets are available in the starter package. For each data point, there are 5 real-value features (x_1, x_2, x_3, x_4, x_5). The target is 0/1, indicating which class it belongs to.

- *data/**: include *train.csv* for training and *dev.csv* for development. We will have an extra private test set for the final grading.
- *mp3.py*: the python file you need to work on and submit.
- *MP3.pdf*: this instruction file

Code Instructions

In this project, you need to implement one perceptron for binary classification. You are only allowed to use one perceptron (as defined in Lecture 10 P5). Specifically, you need to update the weight based on the misclassified data until it converges or reach the maximum epoch you set. There are several hyperparameters that might be helpful:

1. Learning rate: it decides the pace of parameter update.
2. Maximum epoch: the maximum epoch to update parameters. If the algorithm doesn't converge, it will stop after the maximum epoch.

`run_train_test(training_data, testing_data)` function is called to update the parameters in the perceptron and get the prediction on `testing_data` via the trained perceptron.

Evaluation Instructions

You can test your program with the **train.csv** and **dev.csv** provided in this starter package. You can directly run ***mp3.py*** for checking:

```
$ python mp3.py
```

This will output the accuracy you get on the dev dataset. The final score will be graded on another private test set (use the same training set).

Submission Instructions

You should upload ***mp3.py*** to Gradescope for grading.

Grading Instruction:

The score on Gradescope are computed based on the accuracy using the following rule:

Manual Grader (60%): If your code can run successfully, you will get full points. Otherwise, your score will be based on the completeness.

AutoGrader (40%): Your score depends on the accuracy on the private test:

- If your accuracy is above 50%: $Score = \min(100, 55 + 100 * (Accuracy - 50\%))$
- If your accuracy is below 50%: $Score = \max(0, 55 - 100 * (50\% - Accuracy))$

Late submissions are accepted with 20% reduced credit each day. For example, you will get 80% of full credit if one day late, 60% if two days late and so on so forth.