Sarah Schröder, Alexander Schulz Bielefeld University

Introduction to Machine Learning (WS 2025/26)
Programming Assignment 1

Released: Thursday, 06.11.2025
Submission Deadline: Sunday, 16.11.2025 until 23:59 in Moodle.
Discussion: Monday, 17.11.2025 in the tutorials.
Please solve the exercises in groups of three. In order to get the points, at least one group member needs to attend the tutorial and present your solution if asked by the tutor. Submitting your solutions to the Moodle is required to avoid plagiarism and unfair advantages from attending later tutorials.
If you have any questions, please ask your tutor and do not hesitate to write an email to intromachlearn@techfak.uni-bielefeld.de.

1 kNN

(*3 Points*)

Train a k-NN classifier and do hyperparameter search for k based on the code given in `Task1_kNN.py`.

- Adapt the code to train k-NN with $k = 5$ and report the train and test accuracy. What do you observe?
- Find the best value for k. Split the training data into a train and validation set. Plot the validation accuracy for all possible values of k, then train k-NN with the best k and report the test accuracy again. Which values for k did you choose? Did the test accuracy improve as expected?
- Assume you want to train k-NN on another dataset. Can you use the same value of k? Why/ Why not?

The questions are meant to be discussed in the tutorial. Think about them when working on the task.

2 Logistic Regression

(*9 Points*)

Write a custom implementation of Logistic Regression with Gradient Descent as discussed in the lecture. Your implementation should be based on `Task2_LogisticRegression.py`. Solve the following tasks:

(a) *(4 Points)* Implement Logistic Regression as defined in the lecture slides, using only numpy. Then train and evaluate it on dataset2. How does the model perform?

(b) *(2 Points)* Complete the function for plotting the decision boundary (by drawing a line, using the model parameters) and create plots for Logistic Regression fitted on dataset2. What do you observe?

(c) *(3 Points)* Run Gradient Descent with different step sizes $\eta \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ on dataset2 and plot how the NLL changes over n[1] iterations/ until convergence. What do you observe? Which step size is appropriate here?

The questions are meant to be discussed in the tutorial. Think about them when working on the task.

3 k-NN: Guess the dataset

(*3 Points*)

We evaluated k-NN with different values of k on three different datasets (similar to what you did in task 1). You can see the datasets A, B and C in Figure 1 and the accuracy curves from the evaluation in Figure 2.

Which Curve does belong to which dataset? How can you tell?

[1] n should be large enough to see the convergence

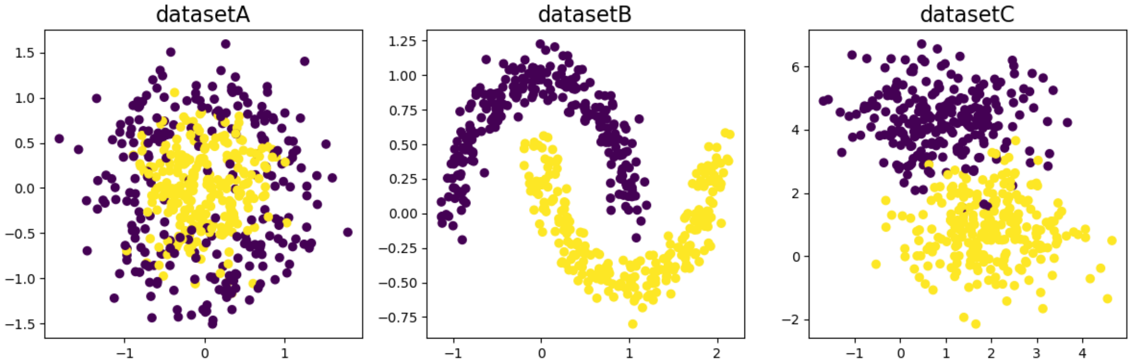

Figure 1: The different datasets. Colors indicate different classes.

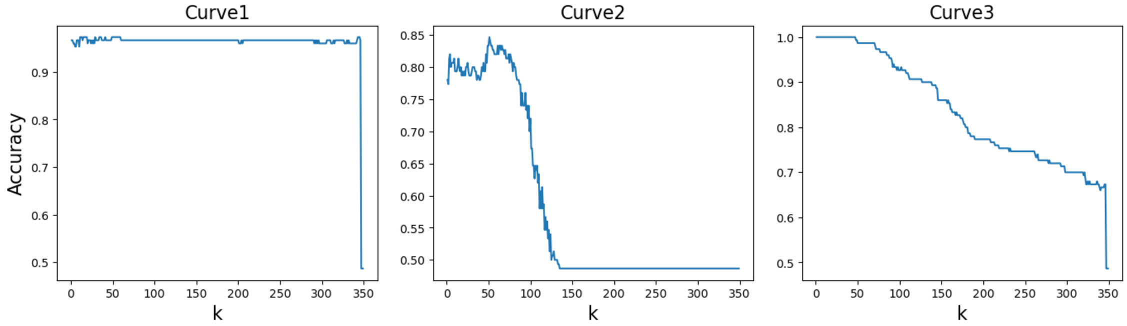

Figure 2: Validation accuracy for different values of k on three different datasets. All datasets have the same number of samples.

4 Code Analysis: Why does Logistic Regression perform so bad?

(*5 Points*)

We trained a Logistic Regression model on dataset3, using the code in `Task4_CodeAnalysis.py`. Unfortunately, the results are not what we hoped for...

$$acc_{train} = 0.5925$$
$$acc_{test} = 0.63$$

Think about the following questions:

- What would be the baseline (e.g. if random guessing) for the accuracy on this particular dataset?
- Why does it perform so poorly?
- How would you solve the issue, i.e. getting a model that properly classifies the dataset?

Modify the code to identify the problem and provide a solution (e.g. look more closely into the dataset properties, change the model/ hyperparameters or how it is trained). Use methods discussed in the lecture and explain your choices.