Foundations of Data Analysis - SS2025
Lab assignment
Supervised learning

Due date: 09:45 am on 28.05.2025

This assignment consists of two parts. In Part I, you are not allowed to use AI tools of any kind. In Part II, you are allowed to use Microsoft Copilot (see here for details). Both parts are **designed to instruct as much as to evaluate**. Part I guides you more closely through the task and helps you build intuition around a lecture topic (Neural Networks). In Part II, you are given a high-level outline and are expected to use AI tools to complete the task. The goal of Part II is to help you learn how to use AI effectively to tackle problems that would normally be too challenging with your current knowledge alone within a limited timeframe.

Instructions

- The maximum number of points achievable in this assignment is 100. Kindly follow these instructions and submission requirements to the point, as failing to do so may disqualify your assignment for grading.
- Remember to cite every external source that you use at the end of the assignment.
- Any act of plagiarism will be taken very seriously and handled according to university guidelines. Plagiarism includes copying any content from other students and using AI tools (ChatGPT, Perplexity, Copilot, Grammarly, etc.) where it is prohibited. You will gain much more by honestly doing the assignment.
- All results in your report—whether plots, figures, or numbers—must be generated by you using Python code. The corresponding code and results should be clearly documented in a Jupyter notebook.
- If you have any questions about the assignment, please post in the Moodle forum or write me an email (akshey.kumar@univie.ac.at) no later than 3 days before the submission deadline. Kindly refrain from posting answers on the Moodle forum.
- Please read through the entire assignment before you begin.

Submission requirements

- You must submit **exactly two PDFs** on Moodle consisting of a report and code:
  1. `report_<your_full_name>.pdf`
     (a) A self-written report maximum 4 pages long, font size 10pt, including figures. (No AI tools allowed in writing)
     (b) Supplementary material (optional)
     (c) Instructions on how to set up environment to reproduce your results, Python version, etc.
     (d) Sources
  2. `code_<your_full_name>.pdf`
     (a) The Jupyter notebook you used to solve the asignment tasks with clean, readable code, exported as a PDF. Do not upload the `.ipynb` file.
     (b) All results and figures in the report must be generated within this notebook
     (c) Define all used functions/classes within the notebook (no external files)
- You will be graded purely based on your report. We will check your jupyter notebook PDF, and if we are not convinced that the results in your report were produced in the notebook, you will be summoned to explain your results.
- The report should demonstrate your understanding of the topic and convince the reader of your findings.

Recommended Packages

`numpy`, `matplotlib`, `scikit-learn`, `torch`, `torch-geometric`

Part I — AI tools prohibited (40 points)

AI tools (e.g., ChatGPT, Copilot, Perplexity etc.) are prohibited in Part I. However, you are encouraged to use official documentation, online tutorials, or textbooks to understand the tools and concepts.

1. **Understanding the Data.** Download and unzip the data from `https://ucloud.univie.ac.at/index.php/s/nnPYZcWPrZ4GpH9`. Load the files `features.npy` and `labels.npy` and answer the following:
   (a) How many features and samples are present in the dataset?
   (b) Is the task a regression, binary classification, or multiclass classification?
   (c) Plot a histogram of the label distribution. What does this distribution tell you?
2. **Data visualisation.** Create a 2D or 3D scatter plot of the data, using color to represent the labels.
   (a) Include this figure in your report with an appropriate caption and legend.
   (b) Describe what you see. Is the data linearly separable at first glance?
   (c) Could a linear classifier achieve zero empirical risk? Justify your answer.
   (d) How might the Bayes optimal classifier look in this scenario? Could it achieve zero empirical risk?
3. **Fitting the first classifier.** Before diving into more advanced models, you will start by fitting a simple linear classifier to the data and evaluating its behavior.
   (a) Split the data into a training set and a test set. All model training should be done using only the training set, and all evaluation reported only on the test set. Why is this split necessary?
   (b) Pick the most appropriate of the following linear classifiers from `scikit-learn`:
      - Linear Discriminant Analysis (LDA)
      - Logistic Regression
      - Linear Support Vector Machine (SVM)

      Train your chosen model on the training data. In your report, justify your choice of classifier.
   (c) What is the classification accuracy on the held-out test set? Based on this result, do you think the classifier is learning any meaningful structure in the data? Why or why not?
   (d) Plot the decision boundary of your trained classifier in the original 2D feature space. Shade the regions in feature space according to the class predicted by the model.
4. **Does the model perform better than chance?** Just because a classifier achieves a certain accuracy does not mean it has learned something meaningful. In this section, you will assess whether your model's performance is better than what would be expected from random chance.
   (a) Estimate the chance accuracy by randomly shuffling the test labels and using these shuffled labels as predictions. What accuracy do you get?
   (b) Repeat this process multiple times (e.g., 1000 times) to get a distribution of chance accuracies. Plot a histogram of these values. Mark your model's actual test accuracy on this histogram.
   (c) What does this plot suggest? Could the accuracy your model achieved have occurred by chance?
   (d) How many times did you repeat the shuffling? Justify your choice. Why not fewer?
   (e) Estimate the accuracy of a simple baseline classifier that always predicts the most frequent class in the test set. Is this the same as the chance accuracy? Why?
   (f) Give your interpretation of these results in at most 3 sentences.
5. **Neural nets.** Here, you will use a neural network to fit a non-linear classifier.
   (a) Create a neural network architecture and train it on the data. Refer to PyTorch tutorials on how to implement a neural net classifier.
   (b) Plot the training and test accuracy versus epochs. What can you deduce from the training curve?
   (c) Tune your architecture and learning hyperparameters using cross-validation and draw the chosen architecture (You can draw by hand or use digital tools). Describe your implementation choices (layers, activation, etc.) in less than 4 sentences.
   (d) Plot the decision boundary in the original feature space. Shade regions according to the class predicted by the neural net.

Part II — AI-assisted Assignment (60 points)

In this part, you will learn how to use AI tools to teach yourself new topics and generate code in unfamiliar domains, and decide when AI is helpful and when it is not. You are expected to use AI deliberately not mindlessly. That means breaking the task into manageable steps, thinking carefully about how to frame prompts, and relying on your own reasoning and interpretation. You are only allowed to use Microsoft Copilot as your AI tool. Keep in mind that sometimes the most effective strategy may not involve AI, but rather other resources including research papers, documentation, tutorials, textbooks, Google, or your own brain. For each task, in addition to answering the question, submit the prompts that you used in Microsoft Copilot, and any revisions of the prompts that were necessary to achieve the task. You will also be graded based on the quality of your prompts and your approach to using AI.

Introduction

Neural networks can be adapted to different data types by designing architectures that account for domain-specific properties—for example, convolutional networks for images, recurrent networks for sequences, and graph neural networks for graph data. In this part, you will apply neural networks to graph data and use this to classify whether a molecule is mutagenic or not. Before you begin, familiarise yourself with the mathematical concept of a graph, nodes edges, adjacency matrix, coordinate matrix.

1. Download the MUTAG dataset from TUDatasets. It is available via PyTorch Geometric. Answer the following:
   (a) How many samples, features and classes are in the dataset?
   (b) What real-world object does a sample represent? What do the nodes, edges, features, and labels correspond to?
   (c) How many nodes and edges are present in the first sample (Python index 0)?
   (d) Plot the first graph with a suitable python package and colour the nodes according to the second feature (Python index 1).
2. Follow the procedure introduced in Part I to train and evaluate both a linear model and a feedforward neural network without considering the graph information. i.e. use only the feature vectors for each sample without edge information. Test if this model performs significantly better than chance prediction. In your report add
   (a) How you prepared the data, the architecture of your neural network, and how you tuned for optimal parameters.
   (b) The training curves (train and test accuracy versus epochs) and what you deduce from their shape.
   (c) The test accuracy obtained for both the linear model and the neural network. Provide evidence if your model (does or does not) perform signficantly better than some chosen baseline models with the aim of convincing us that your models indeed learns something from the features.
3. Now fit a graph neural network to the data and rigorously evaluate the predicitions wit the same work flow as in Part I and answer the following.
   (a) Explicitly draw the architecture of your model in the non-graph neural network (Question 2), and the graph neural network and describe the key differences.
   (b) How does your graph neural network take into account graph information? What does the depth of the network signify?
   (c) Perform model validation, find optimal hyperparameters, and describe your chosen process and results less that 3 sentences.
   (d) What accuracy do you obtain with the optimal model? Does it perform significantly better than chance? Substantiate your argument with evidence and a p-value.
4. Now build a deep neural network using only layers with fewer neurons than the original feature space dimension. The final layer must be an embedding layer that projects the data to 2 dimensions. Use this network to answer the following:
   (a) Visualise the embedded data just before the final layer. Label the samples according to the true label, the predicted label. Draw the decision boundary in the embedded space and shade the feature space according to what label your model assigns them.

(b) How many layers do you need to achieve optimal performance? How many neurons per layer?

(c) Is there any statistically significant difference in performance compared to your earlier models? Why is it possible (or impossible) to classify well without projecting to a higher dimensional space?

5. (a) Summarise how you used AI to help you work on Part II in no more than 4 sentences.

   (b) How did you check if the AI responses were correct? Provide evidence.

   (c) Give an example, along with evidence, where AI was particularly helpful, and another example where it was not helpful. How did you overcome this?