Exercise 1 – Language identification with sklearn and skorch

From Linear to Deep

Deadlines

Deadline for Exercise 1: **15.10.2023, 23:59 (Zurich Time)**.
Deadline for the peer review: **23.10.2023, 23:59 (Zurich Time).** You will find instructions for the peer review process at the end of this document.

Learning goals

This exercise consists of two parts: the first part aims at deepening your understanding of linear models. The second part will already target a simple kind of multi-layered network; the multilayer perceptron (MLP). Don't worry if you don't know anything about MLPs when reading. We will cover all you need to know to solve the second part of this exercise next week in class and in the tutorial session. The learning goals of this exercise are…

- to understand linear models and use them for multiclass classification tasks.
- to be able to implement different machine learning models, including MLPs, in scikit-learn and skorch.
- to understand the role of hyper-parameters and regularization.
- to perform an error analysis of machine learning models.
- to observe the most important features that lead to a prediction of a specific class and explain to some degree what the model does (XAI).

Please keep in mind that you can always consult and use the exercise forum if you get stuck (note that we have a separate forum for the exercises).

Deliverables

We request that you hand in your solutions as self-contained Jupyter Notebooks (ipynb files). That way, your reviewers can view and execute your code without potential dependency problems and/or installing new packages or versions of packages.

This exercise consists of two parts. Please hand in your solutions for each part in separate notebooks. The notebooks contain your well-documented, EXECUTED, and EXECUTABLE code. You will also have to write a lab report. The lab report should contain a detailed description of the approaches you have used to solve this exercise. Please also include the results. We highlight sections in this sheet **in green** where we expect a statement about an issue in your lab report.

We encourage you to solve the exercise on Google Colab (in particular for part II) and download the ipynb file when everything is completed. Besides submitting the required ipynb files, you also need to submit a lab report in PDF format.

Please submit a zip folder named ex01.zip containing the following 3 files, named as follows:

- ex01_lr.ipynb
- ex01_nn.ipynb
- ex01_report.pdf

Please note:

- Organize your code such that it is executable when assuming that the data is in the same folder as the scripts. However, DO NOT submit the data files!
- The cells must have already been run and the output must be visible to anyone checking your notebook without having to run the code again.
- Your assessors must be able to run your code. If it doesn't work, you can't get the maximum score.

Data

For both parts of this exercise, you will work with the same data. The goal is to classify languages based on text sections. This is a challenging extension of the problem described in Goldberg: chapter 2. However, we will work with more languages than just six and the text segments we need to classify are a bit shorter. The material folder in the exercise section of OLAT contains the four files files `x_train.txt` and `y_train.txt`, `x_test.txt` and `y_test.txt`.

Part 1 - Language identification with linear classification

To facilitate the start, you can use this notebook in Google Colab which loads the files using the public links. If you like, you can just continue the exercise in your own copy of that notebook. If you choose to work locally, download the files to your computer.

Scikit-learn is a useful Python library for all kinds of machine learning tasks. In the following, you will train several models in sklearn to solve this task. The aim is to become familiar with a few different classifiers, as well as with the basic functionality of sklearn.

1. First, please solve the tasks (cells marked with "T") in the provided jupyter notebook.
2. Create a suitable pipeline in sklearn to preprocess the data. Think about extending the feature space. **What other features could you use to determine the language? Please include additional linguistic features to your machine learning model for this task.**
3. Train the following classifier: LogisticRegression
4. To find the optimal hyperparameter settings for the classifier, use sklearn's GridSearchCV. [hint: don't overdo it at the beginning, since runtime might go up fast] You are supposed to experiment with the following hyperparameters[1]:
    a. **Penalty (Regularization)**
    b. **Solver**
    c. **Experiment with parameters of the Vectorizer (not required, highly advised)**

Report the hyperparameter combination for your best-performing model on the test set.
What is the advantage of grid search cross-validation? Use a confusion matrix to do your error analysis and summarize your answers in your report.

Now that you have your best model, it's time to dive deep into understanding how the model makes predictions.
It is important that we can explain and visualize our models to improve task performance. Explainable models help characterize model fairness, transparency, and outcomes. Let's try to understand what our best-performing logistic regression classification model has learned. **Generate a feature importance table for the top ten features (please have the features named) for the languages English, Swedish, Norwegian, and Japanese. What is more important, extra features or the outputs of the vectorizer, discuss.**
We recommend using the ELI5 library as it supports sklearn pipelines to explain the model weights. For more details, see their documentation on dealing with text classification. We will accept answers from any explanation library/method as long as the explanations for the model weights are provided in a structured/clear way.

Lastly, you will condact a small ablation study. First, choose the two languages for which the classifier worked best. Next, re-fit the best working model several times, each time reducing the number of characters per instance in the training set (1. All characters, 2. 500 characters, 3. 100 characters). **How does the ablation affect the performance of the classifier?**

Part 2 - Your first Neural Network

Let's see if you can beat the best linear model you've trained with sklearn with a simple neural network using skorch. Please start using the following notebook.

1. First, please solve all the tasks (cells marked with "T") given in the notebook.
2. Please improve the neural network to at least up to 80% accuracy. You can also use GridSearchCV but be aware that training a neural network takes much more time. **Play around with 5 different sets of hyperparameters including layer sizes, activation functions, solvers, early stopping, vectorizer parameters, and report your best hyperparameter combination Do you achieve higher performance? Why/why not?** Importantly, please use the same data splits as for Part 1.

[1] In general, we expect you to engineer the full GridSearch optimization Pipeline. However, if runtimes take long, we recommend taking shortcuts on the optimization methods (only 1 option per hyperparameter, max_iter being a ridiculously low number). Performance is not critical but being able to build the complete system is, which is what we want to see in the code and in the report.

Submission & Peer Review Guidelines:

Peer Reviews will be carried out on OLAT.
As soon as the deadline for handing in the exercise expires, you will have time to review the submissions of your peers. You need to do **3 reviews** to get the maximum points for this exercise.
Here are some additional rules:

- **All file submissions are anonymous (for peer review purposes): Do not write your name into the Python scripts, the lab report, or the file names.**
- **ONLY ONE** member of each team submits on OLAT.
- Please submit a zip folder containing all the deliverables.

Groups & Peer Reviews:

- You can create groups of up to three people to solve the exercise together. Each member should contribute equally!
- If you did not already work together for the previous exercise (or already submitted a post with the same team members), write a small post in the "Assignment Team Submission Thread" in the exercise forum on OLAT to notify the instructors about the group.
- Only **one team member submits the solutions.**
- Only the submitting team member will have access to the peer review, however, you should distribute the workload evenly!
- If you do not submit 3 reviews, the maximum number of points you can achieve is 0.75 (out of 1).
- Please use full sentences when giving feedback.
- Be critical, helpful, and fair!
- Please answer all the review questions of the peer review.