Foundations of Data Analysis - SS22
Lab assignment
Supervised learning

Due date: 9:45 am on 7.4.2022

Description and instructions

The maximum number of points achievable in this assignment is 100. Kindly follow the submission instructions carefully, as failing to do so **will result in a penalty**.

- You should work on this assignment individually, however you are allowed and encouraged to discuss your approaches to the problems, as well as questions you may have, with your colleagues.
- You are however not allowed to share your code! (Except for very small parts, in order to discuss problems with your peers.)
- Remember to cite every external source that you use (as comments in your code)!
- Any act of plagiarism will be taken very seriously and handled according to university guidelines.

Do not hesitate to email me (Anja Meunier) at `anja.meunier@univie.ac.at` or post on the discussion forum on Moodle with any questions you may have.

1 Introduction

The purpose of this assignment is for you to put the theory about supervised learning algorithms we have discussed in the lectures into practice. You are given a data set of images of letters, and we ask you to try your best to train an image recognition model which is able to distinguish the letters in the photographs. You are free to use any machine learning method you like, both those presented in the course and other methods, as well as any data preprocessing you think can improve your model. We specifically encourage you to do some research on which methods might be particularly suited for this kind of data, play around with your models, and to **use methods beyond those covered in class**. In short: Give this challenge all you can come up with!

2 Formal requirements

Download the files `X_train.csv`, `y_train.csv`, `fda_ss22_lab.yml` and `template.py` from the u:cloud link `https://ucloud.univie.ac.at/index.php/s/3y91LZ9O2mUcar7` (password: `fda_zn7Q!hj2VP`).

`X_train.csv` contains flattened color images of dimensions 25 x 25 x 3, `y_train.csv` contains the corresponding labels, where 0 corresponds to letter 'A', 1 to letter 'B', etc.

`template.py` contains the template of a function `train_predict(X_train,y_train,X_test)` which returns a prediction `y_pred`. You should add imports and additional functions as needed at the top of the file, and adapt the provided function to include your data preprocessing, model definition, training and prediction where indicated. The template also contains checks of the input and output formats. To evaluate your model we will import the function `train_predict` and call it on the provided training data `X_train` and `y_train` and the secret test data `X_test`. The returned predictions `y_pred` will be compared with the secret `y_test` to compute the accuracy of your model.

We will execute your code in an environment like the one provided (`fda_ss22_lab.yml`). We therefore strongly recommend that you recreate this environment locally with `conda env create -f fda_ss22_lab.yml` and work within it. The environment contains the packages discussed in the lecture, and we ask that you complete the assignment with only those. If you absolutely want to use additional packages, contact me to ask permission.

Upload a single python file to Moodle, named `<last_name>_<letter_first_name>.py`, replacing `<last_name>` with your last name(s) and `<letter_first_name>` with the first letter of your first name. (Example: `meunier_a.py`)

Hints:

- Import the data with
```
import pandas as pd
X_train = pd.read_csv("X_train.csv", index_col=0).values
y_train = pd.read_csv("y_train.csv", index_col=0).values
```
- It may be easier to develop your models in the console or a jupyter notebook, and only later add your final processing pipeline to the template.
- Make sure to set aside part of the training data as validation set, in order to estimate the performance of your model on unseen data!

3 Evaluation

We will evaluate your model on a secret test set! You will be awarded the accuracy your model achieves (rounded to full percentages) as points. Additionally, you will be able to 'earn back' up to half of the missing points by

- commenting and documenting your code well,
- using efficient implementations (e.g. use numpy functions, not for loops whenever possible), and
- using challenging methods beyond those covered in class.

Examples:

- You train a fantastic model and achieve 99.5% accuracy. You will get the full 100 points for the assignment.
- If your model achieves 77% accuracy on the test data, you can earn back up to 11.5 points by submitting clean code and using interesting methods. You can thus achieve 88.5 points maximum for this assignment.

What if your code does not run or provide any results? We will take limited time to try and make minor fixes, and will deduct some points from the final result, depending on how severe the problem was. However, if it is not quickly solvable, we will award you 30 points at best, depending on the quality of your code and the models you used.

So please pay close attention to the formal requirements!