Johannes Brinkrolf, Valerie Vaquet Bielefeld University

Introduction to Machine Learning (WS 2021/22)
3rd Project

Released: Wednesday, 22.12.2021.
Due: Please solve the exercises in groups of three and submit your report and your code as an executable python script (`.py`) to LernraumPlus by **Tuesday, 25.01.2022, 11:59pm**.

- Please note that the course language is **English**. However, you might hand-in your report in German as well as in English, but be consistent. We will not substract any points for errors regarding language as long as your report is understandable.
- We provide a LaTeX template `report_template.tex` which might help you.
- There is a Q&A tutorial for this project on Monday, 17.01.2022. You can use this session to work on this sheet and/or ask your tutor if you get into trouble.
- Submit your *pdf* and your code as *py*-file (you can export a py-file out of jupyter-notebook) to the LernraumPlus!
- If you use any code from the internet put a link to the source as an comment into your code for reference.
- The project will be discussed during the tutorials on Monday, 31.01.2022.
- If you have any questions, please ask your tutor or write an email to intromachlearn@techfak.uni-bielefeld.de.

In this project you will work on both artificial and real world data. As for the previous project there will be three parts, where the first two are mandatory and for the third you can select one of two options. You can gain 100 points in total and there is no peer-review this time.

Work on the tasks described below and write a **full-text** report on **max. four** pages. Additionally, you can put all tables and figures, which do not fit into the page limit into the appendix. Please make sure to refer to them in your text. Your report should contain

- a description of the data sets used (now that you are partly working with real world data, your description might be a little more detailed here as you have information about what the features actually mean),
- descriptions of models and techniques used,
- a documentation of your experimental set-up (what choices did you make? Why?),
- your results and an analysis,
- and, of course, a short introduction and conclusion.

When writing your report, please make sure that your descriptions are complete and precise. Based on your text we should be able to reproduce your pipeline and your results. Besides, you will have quite a lot of results. Consider one part of the task it to present them in a consist way. For example, you can consider visualizing your results in plots or creating tables.

When putting together your report pay attention to the structure. This is very important, as it is hard grasp work described in a badly structured report.

Hint: When creating plots using matplotlib, you can save the current figure by `plt.savefig('path/to/file.eps', format='eps')`. If you are using LaTeX for the first time `www.tablesgenerator.com` might be a helpful resource for easily creating clean looking tables.

1 Decision Trees and Feature Importances

(a) Train and evaluate a random forest classifier, and the other classifiers you know from the lecture on `dataset1.npz`.

(b) Take a look at the feature importances of the random forest. Analyze them with respect to the data. Rerun your experiments on a suitable subset of the features.

2 Clustering

(a) Apply and evaluate at least three different clustering algorithms from the lecture on `dataset2.npz`. Analyze the impact of different hyperparameter for the different models. For doing a meaningful analysis choose hyperparameters such that the number of clusters varies. In this case, you should both consider evaluation scores and visualization of the clustering for your analysis. Do they align? Why/Why not?

(b) Load the three given pictures and compress them by clustering the pixels of each picture according to their color values. Then, represent each pixel of a cluster in the mean color of the cluster. Visualize your results for different reasonable numbers of clusters.
Hints:

- You do not need to actually compress the pictures so they use less disk space. Here, we only go the first step of that and assign new color values to each pixel.
- You can use the library `skimage (you might need to install it via pip install)`
- Have a look into `loadAndDisplayImagesExampe.py`
- Calculation may need some time (up to a minute per picture)

3 Further steps

It's time again to choose which direction you want your project to go. Pick one of the following options:

(a) So called *extra trees* are another option for base-learners for a random forest. Repeat the experiment of Task 1 for them and report your results. Besides, present the idea of extra trees in a short video of about 3-4 minutes. As you are explaining them in your video you do not need to include a description in your report. Of course, you still need to state your used (hyper-) parameter in the experiments.

(b) If you are interested in another topic related to the lecture, please prepare a 5-6 minute long video presenting this topic. You should do your own research on the topic but build upon concepts that were part of the lecture. If you are not sure whether your topic is suitable, please discuss this with your tutor. Please, state your core message/question of your video in one or two sentences and send them directly to your tutor, at latest on **16th January**. In case the core message is not suitable or too broad, your tutor will contact you to come up with a more suitable option.

For all the videos, it is required, that **all group members** are presenting for approximately the same amount of time. You can simply record the video with your webcam or smartphone. Simple cutting can be done with ffmpeg[1]. Please do not speed up or slow down the video to match the time requirements. If you prefer to record your video in German, that is fine for us.
We are not looking for standard slide-based presentations. Instead, you can use drawings, sketches, hand-crafted things that visualize your point. If you need to rely on images, plots, or formulas you can show them as a screen recording but please don't use any text.

[1] `https://ffmpeg.org/`
If the codecs are the same, you can simply use the following command: `ffmpeg -f concat -safe 0 -i mylist.txt finalVideo.mp4`
where `mylist.txt` is a text file containing all names of the videos or

```
ffmpeg -i input1.mp4 -i input2.webm -i input3.mov -filter_complex "[0:v:0][0:a:0][1:v:0][1:a:0][2:v:0][2:a:0]
concat=n=3:v=1:a=1[outv][outa]" -map "[outv]" -map "[outa]" finalVideo.mp4
```

For further information have a look at `https://trac.ffmpeg.org/wiki/Concatenate`