Semester Project for Advanced Topics in Machine Learning

In order to successfully finish the course *Advanced Topics in Machine Learning*, you are required to do a small project. This project consists of a conceptual, a programming (alternatively: usage of fitting libraries) and a presentation part. The work in a team is required, thus you need gather a team of 3 to 5 students. In total there are 4 deadlines, that you need to be aware of:

1) *7th of May 2020*: Please send us an e-mail (sayantan.polley@ovgu.de) regarding your team. Changes to the team can only be done afterwards if absolutely necessary.

2) *14th of May 2020*: Send us a short document (about one page or less) explaining what you plan to do for the project via Moodle.

3) *25th of June 2020*: Hand in your assignment along with the source code, documentation and explanation(s) uploaded in Moodle. This deadline is non-negotiable!

4) *29th June/1st of July and 6th/8th July 2020*: Be prepared to present your results (via Zoom) with slides in about 5 to 10 (maximum!) minutes. (Try to stick around 7 minutes, the presentation dates may change slightly to accommodate all groups)

Please note, that it is not enough to just hand in documented source code or a short description of what you did. The most important part of the project is the **documentation of all necessary steps** of the data preparation, modeling and evaluation, that you have done. We will not have a deeper look at your source code or the tools, that you have used for solving the problem. Instead we will focus on your **explanations** and whether you actually have understood, what is important for analyzing data sets, especially in the context of the given problem. Therefore you should concentrate on presenting the individual design steps and of course the results (**evaluation**). For the latter part, you should **include visualisations** (confusion matrices, diagrams, charts, time series, etc.) in order to support the understanding. See the example structure on the last page.
The project should be written in either Python 3+ or Java 7+. Please do not use other languages. You may use any freely available libraries for the purpose of the task. Please do not forget to submit a list of necessary packages or better a routine for automatically installing those packages (like maven for Java).

Genre Identification on (a sub-set of) Gutenberg Corpus

Consider this set of books belonging to the *19th Century English Fiction*[1]. The data set is created from *Project Gutenberg*[2]. The data set consists of about 1000 books and roughly 10 genres. The task here consists of detection (i.e. classification) of *genre*[3] of a book. Each data-point in this classification task is a fiction book with a label (genre). Please note the following three main challenges, that you have to consider for your work:

1) Extract features that are relevant to fiction books, which may include ideas like sentiment, setting[4] and so on, using appropriate libraries if possible. You cannot hand-in a solution with bag-of-words or simple term incidences, as your feature representation.

2) Outline which models you intend on using and why and how models selection was performed.

3) Explain how the evaluation of the model is being done and how the data set is to be partitioned while taking into account potential challenges like class imbalances and similar.

We have one task as a nice-to-have feature (optional): Do you think you can measure the bias and variance of at least one of your models and also visualize it?

Please **document every step** that has been made in order to solve the challenges. If you use any libraries in your programming, reference where you have got them and which version your program uses. At best include the libraries in your contribution if possible (please do not exceed a certain size in your submission). It is not enough to just use existing algorithms. **We expect you to extract at least a few features that may be relevant to fiction books. You are free to make simplified assumptions to arrive at the features. But mention those assumptions clearly!** You need to have an understanding, what exactly happens to the data. Therefore you should be able to explain (and also document!), why certain methods could or could not be used for your problem. We do not expect perfect results, especially since the data set is relatively difficult to handle, but the challenges have to be addressed and we must have the feeling in the end, that you tried to solve them sensibly.

Submissions will happen via Moodle. This task will be available in Moodle where you can upload the initial one page proposal, final report, source code in a zipped format. The format of the final report is IEEE two column format[5] not exceeding eight pages with references. We have defined some illustrative sections in the next page that you may use for your report, but overall the final deliverable should be as IEEE two column format.

[1] http://dke.ovgu.de/findke/en/Research/Data+Sets-p-1140.html
[2] https://www.gutenberg.org/
[3] https://en.wikipedia.org/wiki/Genre
[4] https://web.csulb.edu/~yamadaty/EleFic.html
[5] https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhhncsfqn

Example Structure for the Project Report

1 Motivation and Problem Statement

- Shortly motivate the task surrounding the data set and explain the problem.
- This should cover not more than half a page for such a report.
- Ideally, you want to raise one or several questions, that are answered later on in the evaluation.

2 Data set

- Shortly explain how the data set is structured.
- Are there any parts in the data set, that need to be addressed regarding the previously mentioned problem?
- Already try to give some short insights about the data set. E.g. you could add and explain a scatter-plot (if feasible and sensible).

3 Concept

- Which methods should be tested on the data set and why (e.g. what makes them suitable for the task)?
- Explain, how do you plan to represent features and extract them.
- Give some thoughts on the advantages and disadvantages of the chosen methods.

4 Implementation

- Very shortly give some details about the implementation or used tools.
- How did you implement the concepts? Are they any special things, that you needed to take care of?
- What assumptions are being made (especially interesting when using a tool set) and how could they affect the results?

5 Evaluation

- How did you perform model selection?
- Introduce your concept for evaluation. How are you using the data set for training and testing?

- How many iterations are you testing? How many different parameter combinations? What are your evaluation measures and why are you using those? Hint: Accuracy alone may not be enough.
- What results are you getting? How do the two methods compare to each other?
- How do the methods perform in the context of the original problem statement (question)? Does it fulfill the requirements? If yes, why? If not, what needs to be different?

6 Conclusion

- Shortly conclude on your overall project, the results and research question/problem.
- What went well, what was not so good? You can also think about runtime performance here, if interesting.
- What could be done with your results in the context of the motivation?
- What else could have been done in the project, which is sensible in your context? (Don't just write: We can try different models.)