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, programming (alternatively: usage of fitting tools) and presentation part. The work in a team is required, thus you need gather a team of 3 to 4 students. Please talk to me beforehand (Marcus Thiel (marcus.thiel@ovgu.de)), if you would like to do it in a team of 2 or even alone. In total there are 4 deadlines, that you need to be aware of:

1. *30th of April 2018*: Please send me an e-mail regarding your team. Changes to the team can only be done afterwards if absolutely necessary.

2. *14th of May 2018*: Send me a short document (about one page or less) explaining what you plan to do for the project.

3. *20th of June 2018*: Hand in your assignment along with the source code (if any), documentation and explanation(s). This deadline is non-negotiable!

4. *25th of June and 2nd of July 2018*: Be prepared to present your results in class with slides in about 5 to 10 (maximum!) minutes. (Try to stick around 7 minutes)

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. I won't have a deeper look at your source code or the tools, that you have used for solving the problem. Instead I will focus on your **explanations** and whether you actually have understood, what is important for analysing data sets. 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 can be written in any language of your choice. Alternatively you can use any available tool that you like. The only requirement is, that I am able to reproduce your results based on your explanations. For that, it is imperative, that the source code can be compiled on my hardware (I've got a Linux (Ubuntu) PC.) or that the tools are available for Linux (Ubuntu) as well. If you really need to use something else (Windows or Mac only software for example or commercial software) please consult me (Marcus Thiel (marcus.thiel@ovgu.de)) first.
Remark: I personally recommend to use **Python Jupyter Notebook** for this task.

Detection of APS failures in Scania trucks

Given is the *APS Failure*[1] at Scania Trucks data set from the *UCI Machine Learning Repository*. The data set is from real world operational data of Scania trucks. The classification is binary (pos and neg) denoted by the label. You've got 171 unnamed features, where some are part of a histogram. In total, there 60.000 training examples and 16.000 test examples. Your task is to find a suitable method (or several) to automatically recognize the label on the provided test data. It is up to you to analyse the data set, clean and prepare it for the task. Please also note the cost of wrongly classified instances in the description. Make use of that when evaluating your classifier. There are several general steps, that you may need to take in order to achieve your goal:

- Analyse the data set and the attributes.
- Clean the data, e.g. getting rid of missing values or outliers. (This could also be done after creating the concept.)
- Create a concept on how you can classify the data. Note down the differences between the online and offline variant.
- You may need to convert the data into a format useable for that concept.
- Use (different, at least 2) suitable machine learning algorithms. You may take existing libraries for that. (Hint: Please choose algorithms, that are not extremely similar. For example, using *only* Decision Trees and Random Forests would be a bad choice.)
- Evaluate the results of your model/concept. Don't just write down raw numbers, but also think about why it either does work or it doesn't.
- Compare the results to different algorithms.
- Present an outlook for improvements on your concept.

Please **document every step** that has been made in order to achieve the goal. If you use any libraries in your programming, reference where you've got them and which version your program uses. At best include the libraries in your contribution if possible. If you are using tools, please provide a link to the used tools and explain, what features have been used. It is not enough to just use the existing algorithms. 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.

In case of questions, send an email to Marcus Thiel (marcus.thiel@ovgu.de).

[1] https://archive.ics.uci.edu/ml/datasets/APS+Failure+at+Scania+Trucks

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 adressed regarding the previously mentioned problem?
- Already try to give some short insights about the data set. E.g. you could add and explain a scatterplot.

3 Concept

- Which methods should be tested on the data set and why (e.g. what makes them suitable for the task)?
- Explain, how the data set has to be used for the specific methods. Therefore also explain, if there are transformations or feature selections needed and/or sensible.
- 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

- 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.)