

# **Automatic Ranking of Classification Algorithms**

**A Comparison of Regression-Based Ranking and Preference Models**

**Bachelor Thesis Proposal & Work Plan**

Helena Graf

matriculation number: 7011643

hgraf@mail.upb.de

November 27, 2017

Supervisors:

Prof. Dr. Eyke Hüllermeier

Prof. Dr. Axel-Cyrille Ngonga Ngomo

# Contents

|          |                                       |          |
|----------|---------------------------------------|----------|
| <b>1</b> | <b>Motivation</b>                     | <b>3</b> |
| <b>2</b> | <b>Related Work</b>                   | <b>4</b> |
| <b>3</b> | <b>Goals</b>                          | <b>5</b> |
| 3.1      | Required Goals . . . . .              | 5        |
| 3.2      | Optional Goals . . . . .              | 6        |
| <b>4</b> | <b>Approach</b>                       | <b>6</b> |
| <b>5</b> | <b>Preliminary Document Structure</b> | <b>8</b> |
| <b>6</b> | <b>Schedule</b>                       | <b>9</b> |

# 1 Motivation

As the potential of big data is evident, an increasing amount of data is collected and available for analyzation - but this potential is not utilized. In a white paper, the international data corporation claims that in 2012, out of the 2.8 zettabytes (ZB) of available data, only 3% were tagged as to enable further processing, and only 0.5% were analyzed. While the claim of a follow-up paper in 2017 projects that in 2025, 15% of the then estimated 163ZB of global data will be tagged, and approximately 3% analyzed, is more optimistic, it still shows that there is a huge amount of data that could potentially be useful, but this is not exploited. This shows that the demand of data to be analyzed cannot be covered by data scientists alone, and thus calls for automation of the process, in a way that not much expertise in the field of machine learning is needed to gain insights about the collected data.[citations will follow]

Often, the problem at hand is classification: Assigning an instance a certain class, for example deciding if a client of a bank will be given a credit, based on factors like other existing credits or the job of the client. But selecting a fitting classifier for a new dataset is difficult, since algorithm performances can vary substantially among datasets, and it is not feasible to simply apply a large number of them to empirically find a good match. For example, on a dataset about the electricity prices in the Australian state New South Wales, the predictive accuracy for the Multilayer Perceptron<sup>1</sup> is 0.7887. The predictive accuracy of the Random Forests<sup>2</sup> algorithm on the same data set is 0.9236, a much higher value. On a different data set, with the topic of vehicle silhouettes, we get a predictive accuracy of 0.7979 for the Multilayer Perceptron, and 0.7518 for Random Forests, showing an advantage of the former on this dataset<sup>3</sup>. So in each case, one would have picked a different algorithm in order to achieve the best results. In general, this means that for a different dataset, a different algorithm might yield the best performance.[citations will follow]

Since there is no one best classifier for all datasets, it is likely that how well a classifier performs on a given dataset is to at least some degree dependent on properties of the dataset. Combined with the need for automatic machine learning, this calls for an approach that takes past performances of classifiers, dependent on properties of datasets, in consideration to automatically make suggestions for classifiers for a given dataset.

---

<sup>1</sup> With standart hyperparameters (L:0.3,M:0.2,N:500,V:0,S:0,E:20,H:a).

<sup>2</sup> With standart hyperparameters (P:100,I:100,num-slots:1,K:0,M:1.0,V:0.001,S:1).

<sup>3</sup> Hyperparameters as above.

## 2 Related Work

The demand for aid in the process of selecting an algorithm has already lead to the development of tools that automate machine-learning (Auto-ML). In the following paragraphs, The way that three of those tools work is outlined briefly.

Auto-WEKA is an Auto-ML tool that both selects a machine learning algorithm and sets its hyperparameters by using Bayesian optimization [5]. It was first released in 2013 as an extension to the popular data mining software WEKA [2] to assist the large number of novice users of the software in selecting parameterized algorithms for their problems. The tool has since grown in popularity and is in version 2.0 as of March 2016 [4]. In Auto-WEKA, the problem of selecting an algorithm and its hyperparameters is combined by treating the algorithm itself as a hyperparameter and searching the joint space of algorithms and hyperparameters for the best solution. An input dataset is first preprocessed by means of feature selection. Then, Sequential Model-Based Optimization is used to 'iterate[...] between fitting models and using them to make choices about which configurations to investigate' [3]. In the case of Auto-WEKA, this means that during the optimization process, a model is built, a configuration of hyperparameters that is promising regarding the current model and training data is tried out, and the result is integrated into the model. This cycle is then repeated until the allocated time has run out. While Auto-WEKA works reasonably well, it does not use meta-knowledge, that is the algorithms performance regarding properties of the datasets, to make its decisions.

AUTO-SKLEARN has been described as a sister-package to Auto-WEKA and is an Auto-ML tool which is based on scikit-learn [1]. It works very similar to Auto-ML but extends it by adding a meta-learning pre-processing step to warm-start the Bayesian Optimization and automatically constructing ensembles during optimization. During the pre-processing phase, performance values for the classifiers available in AUTO-SKLEARN are recorded on a set of datasets. Then, for each dataset, the algorithm which shows the best empirical performance is noted for each dataset. Then, certain meta-features are calculated for each dataset. For a new dataset, at first, its meta-features are computed. Then, the Manhattan distance to the other datasets is determined according to the meta-features, and the algorithms that are associated with the k-nearest datasets are used as a starting point for further optimization. The authors observe that the additional meta-learning and ensemble construction result in a more efficient and robust way. Their results show that Meta-Learning can be used to improve the overall Auto-ML process.

ML-Plan is an Auto-ML tool that instead of concentrating on hyperparameter optimization, aims to optimize the whole machine-learning pipeline [Reference will follow]. This is achieved by viewing machine-learning as a task, building a hierarchical task network out of those tasks, and then searching for a solution in the induced tree structure. In the tree, the root node is a complex task, middle nodes represent incomplete pipelines, and leaf nodes are complete pipelines. An example of this might be 'classify' as the root node, with an intermediate node on some level that contains the tasks 'build NN', 'train NN' and 'predict from NN'. The first of these tasks would then further be decomposed down the tree and could lead to a leaf node with  $n$  tasks 'Add layer', 'build NN' and 'predict from NN', which are all so-called primitive tasks that do not need to be further decomposed. A best-first search algorithm in a modified variant is used to find the good solutions in this pipeline. While this variant does not use meta-learning in the process of optimizing the pipeline, the authors find that their results exceed those achieved by Auto-WEKA.

## 3 Goals

The aim of this thesis is to test the assumption that given a new dataset, one can derive an accuracy-based ranking of classification algorithms from their past performances by means of regression-based compared with preference-based ranking, and to compare the two methods. It is derived from the need for Auto-ML tools and the fact that related solutions so far have not exploited Meta-Learning in this way. If the hypothesis is true, this tool could be used as a pre-processing step in another Auto-ML framework similarly to how Meta-Learning is used in AUTO-SKLEARN, as a pre-processing step to speed up the whole process. The aim presented consists of multiple goals, which can be divided into required and optional goals.

### 3.1 Required Goals

Based on the general objective of the thesis, several goals should be achieved:

1. Implementing a tool with the following functionality:
  - 1.1 A pre-processing phase in which the performances of the available set of classifiers is calculated for a set of datasets, a regression model is fit to each of the classifiers, and a preference-ranker is fit to the ranks of the algorithms which can be deduced from their performance values.

- 1.2 A method where the tool is given a dataset and returns a ranking of classifiers for this dataset.
- 1.3 The possibility to choose between regression-based and preference-based ranking.
2. Evaluate the performance of both the regression and preference-ranking variant of the implementation by comparing them against each other, an oracle, and a baseline.

## 3.2 Optional Goals

If the required goals have been met in less time than originally allocated, addressing additional goals may be considered. The following list suggests possible optional goals. However, they may be replaced if while working on the thesis it becomes clear that more appropriate expansions exists.

1. Extend the evaluation to compare the tool with other Auto-ML solutions.
2. Extend the evaluation to compare with a more intelligent baseline.
3. Adding another layer to search by including hyperparameters, possibly by means of iterative requests to tool.
4. If the accuracy prediction is succesful, add runtime and / or complexity predictions and make combined predictions

## 4 Approach

As described in chapter 3, the goal is to implement an Auto-ML tool that returns a ranking of classification algorithms for a given dataset based on past performances of the classifiers by using regression and preference-ranking, and comparing and evaluating the results.

The implementation of the tool will be done in Java 8, and will consist of two phases. In the first phase, a basic functionality is going to be ensured, but placeholder regression and preference-ranking models are going to be used. The first step of this phase is to implement the first preprocessing phase, which consist of computing the performances of classifiers on datasets. For the implementation of the classifiers, WEKA 3.8.1 is going to be used, which dictates the set of classifiers that will be included. The datasets will be fetched from OpenML.org, which is an open-source project where users can contribute datasets, implementations for machine learning algorithms, and run said algorithms on the datasets. Since the

results of the runs are available publicly on the website, not all of the required base data will be needed to be computed, but can be extracted instead. OpenML.org provides the datasets in .ARFF format, and supplies a plugin that works from WEKA, which should ensure that these libraries work together smoothly. The second step of the first phase is to add the functionality to fit a preliminary regression model and preference ranker to the performance data, based on properties of the datasets. This step also includes providing a method that can be given a dataset and returns a ranking, based on the choice of ranking method.

In the second phase of the implementation, the preliminary algorithms used in the first phase will be replaced. The regression models are going to be customized for each classifier by using ML-Plan (described in section 2) and configuring it to work with regression models. To do this, the configuration file of ML-Plan has to be manipulated. The ranking model is going to be optimized in this phase as well, using the jPL framework [citation will follow]. The project provides a framework for the evaluation of preference learners, including label ranking, which is the problem at hand.

For the evaluation, it is necessary to generate a baseline and an oracle. Both can be obtained from the data of the preprocessing phase. Since due to all performances of classifiers being recorded on all sets, one can obtain a best ranking of classifiers for each dataset from the same table. The baseline which is going to be used is the best algorithm, which performs best on the largest number of datasets. This can be obtained from the table by counting the number of times each algorithm performs the best. The rankings of the oracle, the baseline, and the results of the tool are then going to be compared using the Kendall Rank Correlation Coefficient.

## 5 Preliminary Document Structure

The following structure will serve as an orientation during the writing process. While the chapters should remain the same, subsections may be added, removed or moved.

1. Introduction
2. Fundamentals
  - 2.1 Meta-Learning
  - 2.2 Kendall Rank Correlation Coefficient
  - 2.3 JPL
  - 2.4 WEKA
3. Approach
4. Implementation
  - 4.1 Regression-based Ranking
    - 4.1.1 Preliminary Regression Algorithm
    - 4.1.1 Automatic Algorithm Selection
  - 4.2 Preference Models
    - 4.2.2 Preliminary Ranking Algorithm
    - 4.2.2 Automatic Algorithm Selection
5. Evaluation
  - 5.1 Computing the Accuracy of the Results
  - 5.2 Assessment of the Accuracy
6. Related Work
7. Conclusion
8. Literature
9. Appendix



## 6 Schedule

The working time is going to be structured according to the schedule shown in figure 1. Variations due to unforeseeable circumstances may occur.

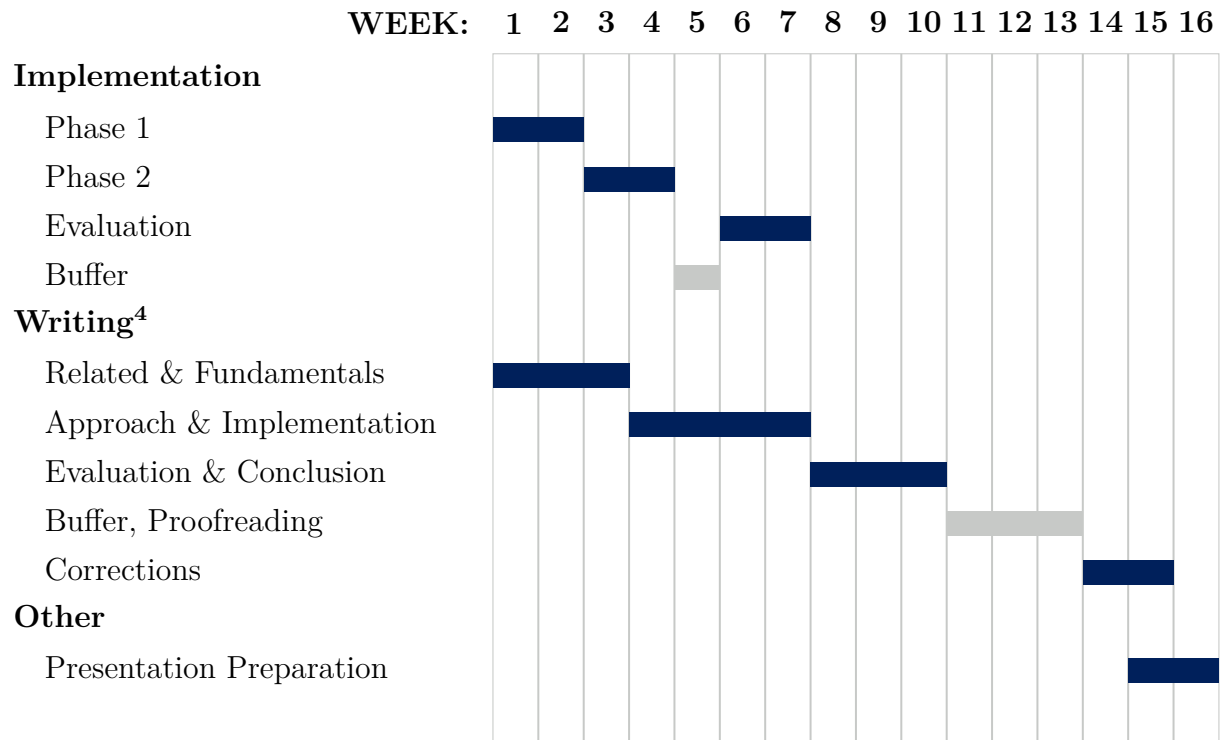


Figure 1: Sketch of the Work Schedule for the thesis.

<sup>4</sup> The chapter 'Introduction' is not included since it will be written in parallel to other chapters, in small bits according to the progress of the thesis.

## References

- [1] Matthias Feurer et al. “Efficient and robust automated machine learning.” In: *Advances in Neural Information Processing Systems*. 2015, pp. 2962–2970.
- [2] Mark Hall et al. “The WEKA data mining software: an update.” In: *ACM SIGKDD explorations newsletter* 11.1 (2009), pp. 10–18.
- [3] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. “Sequential Model-Based Optimization for General Algorithm Configuration.” In: *LION* 5 (2011), pp. 507–523.
- [4] Lars Kotthoff et al. “Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA.” In: *Journal of Machine Learning Research* 17 (2016), pp. 1–5.
- [5] Chris Thornton et al. “Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms.” In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 847–855.

Hereby supervisor and student confirm that this proposal is the basis for the topic assignment of the described work. The timetable and topic description are accepted by both sides as laid out in this proposal.

---

Supervisor  
(Prof. Dr. Eyke Hüllermeier)

---

Student  
(Helena Graf)