Self-service Data Science in Healthcare: using AutoML in the Knowledge Discovery Process

Master's Thesis

Utrecht University
Department of Information and Computing Sciences

Richard Lodewijk Jacobus Ooms

r.l.j.ooms@students.uu.nl
Student number: 4087437

First supervisor: Dr. Marco Spruit
Second supervisor: Dr. Matthieu Brinkhuis

Daily supervisor: Sebastiaan Candel, MSc

“Hiding within those mounds of data is the knowledge that could change the life of a patient, or change the world.” – Atul Butte

Abstract

Keywords: **Automated Machine Learning, Applied Data Science, Healthcare Analytics, Benchmark, Researcher-physicians**

Introduction: The healthcare industry has been lagging in the adoption of analytics. One of the reasons for lagging is the shortage of data scientists in the healthcare sector. Advancements in Machine Learning (ML) and research on its accessibility for non-experts sparked the research field of Automated Machine Learning (AutoML). Because AutoML is designed to make ML accessible to non-expert users, this research aims to find out how researcher-physicians can be supported in their knowledge discovery process by applying AutoML as part of the research field of Applied Data Science (ADS). This is the first study, to the best of our knowledge, to test AutoML methods with domain experts in the healthcare domain.

Method: The method used in this research is design science. First, we selected TPOT as AutoML method based on the results of a benchmark test and requirements from researcher-physicians. We integrated TPOT into two artefacts, a web-application and a notebook. We have evaluated the artefacts with the framework for evaluation in design science to find out which method suits researcher-physicians best.

Results: The benchmark test found that there was no AutoML method that consistently outperformed all other methods one-hour and four-hour budgets. However, TPOT and Auto-Sklearn performed best on both tests. As TPOT was the method that satisfied most requirements, we integrated TPOT into two artefacts. Both artefacts had a similar workflow, but different user interfaces because of a conflict in requirements. Artefact A, a web-application, was perceived better for uploading a dataset and comparing results. Artefact B, a Jupiter notebook, was perceived better regarding the workflow and being in control of model construction. Thus, a hybrid artefact would be best for researcher-physicians. However, both artefacts missed model explainability and an explanation of variable importance for the created model. Hence, the researcher-physicians indicated that they would only use AutoML for the explorative phase of their knowledge discovery process.

Discussion: The results suggest that AutoML methods need work on explaining the created models and their route to model creation. Another issue is the stability of the (Auto)ML models; the models created by an evolutionary algorithm based AutoML methods are hard to reproduce due to their random inception. As much as changing the seed can change the outcome for a single patient.

Acknowledgements

In March 2019, I started on this research project, my master’s thesis. I researched the high paced field of Automated Machine Learning and its possible contribution to enable self-service data science in healthcare. The past few months have brought me much knowledge about both the technique and the domain. The research report that I have written on this subject is the final piece of my master’s studies. To share the knowledge, results, insights and lessons learned from this process, I have written this document. As I could not have done this on my own, I would like to thank the people who contributed to the research and supported me.

First, I would like to thank Marco Spruit, my first supervisor, for your guidance, insights and discussions throughout the project. You helped me by asking the right questions, providing precise feedback and valuable discussions during our meetings.

Second, I would like to thank Matthieu Brinkhuis, my second supervisor, for his insights regarding the experiments, data visualisation and structure. I am grateful for the times you took time for me in when I got stuck in my research.

Third, I would like to thank Deloitte Consulting for providing me with a dynamical environment which helped find the right contacts in the medical sector and an opportunity to get in touch and collaborate with highly skilled data scientists. From the many people in the Deloitte organisation, I would like to thank Sebastiaan Candel in particular, for challenging me and guiding me though the organisation.

Fourth, I would like to thank Sebastien Poirier for contributing significantly to the set-up of the benchmark test. Without your help, it would have been technically impossible to run the test. Your swift replies and problem-solving encouraged me.

Fifth, I would like to thank Fenna Heyning and Lucien Engelen their insights in the medical world and for providing me with contacts that helped me to share the results, and lessons learned from this research.

Sixth, I would like to thank the participants and domain experts that were willing to share their time and insights with me. Their input has been valuable to me, not only for this research but also to get a better understanding of the medical world.

Finally, but definitely not in the last place, I would like to thank my Mom, Dad, brother and girlfriend. Throughout the process of this research, they have always stood by me, supported me, provided me with valuable feedback, moments to relax and helped to make this research a success.

Abbreviations

Abbreviation	Explanation
3PM	Three Phases Method
ADS	Applied Data Science
AUROC	Area Under the Receiver Operator Curve
ATM	Auto-Tuned Models
AutoML	Automated Machine Learning
BO	Bayesian Optimisation
CASH	Combined Algorithm Selection and Hyperparameter optimisation
CRISP-DM	Cross-Industry Standard Process for Data Mining
DM	Data Mining
DS-BoK	Data Science Body of Knowledge
EA	Evolutionary Algorithms
EHR	Electronic Health Record
FEDS	Framework for Evaluating Design Science research
FLASH	Fast LineAr SearcH
GP	Genetic Programming
GUI	Graphical User Interface
HPO	Hyperparameter Optimization
HTN	Hierarchical Task Networks
KD	Knowledge Discovery
KDD	Knowledge Discovery in Databases
LIME	Local Interpretable Model-agnostic Explanations
ML	Machine learning
NN	Neural Network
n.s.	Not significant
PCA	Principal Component Analysis
PoSH	Portfolio Successive Halving
RECIPE	Resilient ClassifIcation Pipeline Evaluation
ROAR	Random Online Adaptive Racing
SMAC	Sequential Model-based Algorithm Configuration
SMBO	Sequential Model Based global Optimization
STZ	Samenwerkende Topklinische opleidingsZiekenhuizen
TPE	Tree-based Parzen Estimator
TPOT	Tree-based Pipeline Optimisation Tool

Table of contents

List of figures

List of tables

1 Introduction

This chapter introduces this research in three sections. First, we introduce the problem statement and research objective. Second, we discuss the research questions. Finally, we discuss the scientific and social relevance of this research.

1.1 Problem statement & Research objective

Data is considered to be ‘the new oil’ in modern society (The Economist, 2017). Where the trade, refinery and smart use of oil could bring one great prosperity in the twentieth century, data is the way to go in the twenty-first century. Data differs from oil in the sense that it is not a physical asset. However, both products have in common that they increase significantly in value when refined in the right way. The most valuable companies in 1999[1] were thriving on oil or oil-related products (Fortune, 2019). In contrast, the five[2] most valuable companies today thrive on smart refinery of data (Statista, 2018). Because of the success of these five companies, many industries have adopted analytics to get the most value out of the ‘new oil’ that they possess.

In the healthcare industry, the adoption of data analytics can be used for cost reduction, improvements of treatment, and increasing patient satisfaction (Feldman, Martin, & Skotnes, 2012; Lee & Yoon, 2017; Malik, Abdallah, & Ala’raj, 2016; X. Wang, Noor-E-Alam, Islam, Hasan, & Germack, 2018; Y. Wang, Kung, & Byrd, 2016).

Although there is enormous potential in analytics, the healthcare sector has been slow in adopting it in their daily practice compared to other industries (Koh & Tan, 2005). Because of the late adoption of analytics, the healthcare industry is lagging compared to other industries considering analytics. When asked, a medical researcher stated the following about the state of analytics in healthcare: *“I seriously believe that we are in the middle ages. I look at my iPhone and think about everything that’s possible and yet here in the hospital, you still get a piece of paper with your appointment.”*(Vries de, 2018, p. 20).

In addition to that there is a shortage of data scientists in healthcare (Gibert, Horsburgh, Athanasiadis, & Holmes, 2018; Harris, Shetterley, Alter, & Schnell, 2017; Manyika et al., 2011; Markow, Braganza, Taska, Hughes, & Miller, 2017). This ever-growing shortage of data scientists hinders the adoption and development of analytics in the healthcare sector (Davenport & Patil, 2012). To improve the adoption of healthcare analytics, one of the focus areas in healthcare research should be making analytics accessible to domain experts (X. Wang et al., 2018).

Enabling domain experts to perform analytics is referred to as Applied Data Science (ADS) (Spruit & Jagesar, 2016). One of the challenges in ADS is making Machine Learning (ML) accessible for domain experts. Making ML available for domain experts is part of the selection vs configuration challenge: As no algorithm configuration works best on all datasets, one cannot provide a single algorithm to domain experts to solve

[1] Exxon mobile, Ford Motor, General Electric, General Motors and Wall-Mart

[2]Amazon, Apple, Facebook, Google and Microsoft

all problems. Domain experts, on the other hand, do not have the expertise to choose and configure an algorithm given their problem and corresponding dataset.

The ML community has been working on making ML more accessible to non-expert users. Thornton, Hutter, Hoos, & Leyton-Brown (2013) developed a tool, Auto-WEKA, to automatically select an algorithm and its optimal settings, given a dataset and a performance metric. The inception of Auto-WEKA sparked a sub-discipline of ML research: Automated Machine Learning (AutoML) (Hutter, Kotthoff, & Vanschoren, 2019). The goal of AutoML is to automate the creation of a machine learning pipeline in order to make ML accessible to non-expert users and create reproducible solutions (Hutter et al., 2019; Thornton et al., 2013).

The objective of this research is to find out how healthcare practitioners can be supported in their knowledge discovery process by using AutoML.

1.2 Research questions

In this research, we defined the main research question as: *How can we support healthcare professionals in their knowledge discovery process by applying AutoML?* The healthcare professionals that we refer to in this research are researcher-physicians. Researcher-physicians are domain experts who are active in both medical research and clinical practice. The main research question is split up into five sub-research questions to structure this research:

1. *What is the knowledge discovery process for healthcare professionals in their research?*

To answer this question, we first provide a brief overview of the history of data science, a definition of data science, and introduce ADS. Second, we provide an overview of knowledge discovery (KD) methods. Finally, we discuss the application of data science in healthcare in Chapter 3.

2. *What are the capabilities of AutoML?*

To answer this question, we define ML and introduce core concepts. Second, we provide insight into the architecture of AutoML methods. Finally, we provide an overview and synthesis of currently available AutoML methods in Chapter 4.

3. *Which AutoML method performs best on a benchmark test, given medical datasets?*

To answer this question, we will use a benchmark suite on a set of AutoML methods with medical datasets. Building on the work of (Gijsbers et al., 2019), we answer this question in Chapter 5.

4. *What are the requirements of healthcare professionals for starting to use AutoML in their daily practice?*

To answer this question, we elicit requirements from healthcare professionals regarding analytics. Based on these requirements, we select a subset of AutoML methods that are considered suitable to support the data mining process for healthcare professionals. The results are available in Chapter 6.

5. *How does the selected AutoML method suit healthcare professionals in their knowledge discovery process?*

To answer this question, we create two artefacts containing an AutoML method. We evaluate the artefacts containing the AutoML method with the healthcare professionals to see how the artefacts and the AutoML method are perceived. The artefact description and evaluation are available in Chapter 7.

1.3 Relevance of research

Scientific relevance

After reviewing the literature on the state of (big) data analysis in healthcare, it becomes evident that there is a lot of progress to be made (Chawla & Davis, 2013; X. Wang et al., 2018). Progress in the automation of the KD process in healthcare is relevant to this research (X. Wang et al., 2018). Automation of the KD process can increase the adoption of analytics by domain experts.

In addition to this focus area of automating healthcare analytics, the ML community has noticed the need to enable access for non-expert users to ML techniques. The need to enable non-experts to use machine learning is one of the drivers that gave birth to the fast-paced research area of AutoML (Hutter et al., 2019; Thornton et al., 2013). The AutoML community aims to automate all steps in the process of creating a machine learning pipeline. However, to the best of our knowledge, no AutoML applications were tested in real-world situations with non-expert users in the healthcare domain.

AutoPrognosis (Alaa & van der Schaar, 2018) is an AutoML method that is developed for healthcare but has not been made available at the time of writing. In addition to its core functionality, AutoPrognosis has an explanation function to justify its choices to clinicians, something that is regarded as valuable in medicine (Cabitza, Rasoini, & Gensini, 2017; Dedding, 2018). This research aims to explore and overcome the boundaries to AutoML adoption in healthcare with a method-agnostic approach as it is the first study, to the best of our knowledge, to experiment with the usage of AutoML methods by domain experts in the medical domain.

Societal relevance:

In the Netherlands, 10% of the GDP is spent on healthcare (OECD, 2019). With an ageing population, this spending is expected to double in 2040 (Rijksinstituut voor Volksgezondheid en Milieu, 2019). With the growing burden of healthcare costs on society, it is vital to improve efficiency and reduce the costs of healthcare. Improvements can be in the supply chain of hospitals (Y. Wang & Hajli, 2016), development of personalised care plans to improve quality and experience of patients (X. Wang et al., 2018) and improve operational efficiency (Y. Wang et al., 2016). This research aims to catalyse the adoption of analytics in healthcare by finding out how we can support the knowledge discovery process of domain experts with AutoML.

2 Research method

This chapter describes the research method used in this research. First, we discuss design science. After that, we discuss how we answer every research question. Finally, we connect the different research questions to the design science cycles.

2.1 Research approach

This research is conducted using the design science method as proposed by Hevner, Ram, March, and Park (2004). Design science is a method which is useful in applied information systems science. Because this research is on ADS, and we developed an information system to aid data analysis, design science is the right choice according to the norms. We position our research as improvement design science (Gregor & Hevner, 2013). The goal of improvement in design science is to create a more efficient and effective product for self-service data science. To do so, we create a 'level 1' artefact to create specific knowledge about this topic (Gregor & Hevner, 2013).

Design science uses a three-cycle research approach combining the relevance, design and rigor cycle, as Figure 1 illustrates (Hevner et al., 2004). The relevance cycle initiates and concludes design science research. With the initiation, the relevance cycle identifies the problem or opportunity that is addressed by design science. The second part of the initiation is used to ask the question: *"Does the design artefact improve the environment and how can this improvement be measured?"* (Hevner, 2007, p. 3). After the research, the relevance cycle is used to measure the success of the artefact within the application domain.

The rigor cycle is what distinguishes design science from application development. The rigor cycle ensures that design science research is grounded in a knowledge base consisting of scientific theories, existing expertise in the application domain of the research and existing application in research (Hevner, 2007). More importantly, we add results from the study to the existing knowledge base.

The design cycle is the heart of design science research. The goal is to create alternative designs and evaluate these based on requirements and the requested methods and theories. These requirements are derived from the relevance cycle, whereas we draw evaluation methods from the rigor cycle. We use the framework for evaluation in design science research (FEDS) to evaluate the artefacts using the technical risk and efficacy strategy (Venable, Pries-Heje, & Baskerville, 2016). We selected the risk and efficacy strategy for two reasons. First, it is too expensive to evaluate and integrate the proposed artefacts within the real setting. Second, the significant design risk is technically oriented, as AutoML is a new technique. FEDS states that the properties of an artefact are evaluated after choosing a strategy for evaluation. The final step is creating an evaluation to assess these properties based on the selected strategy.

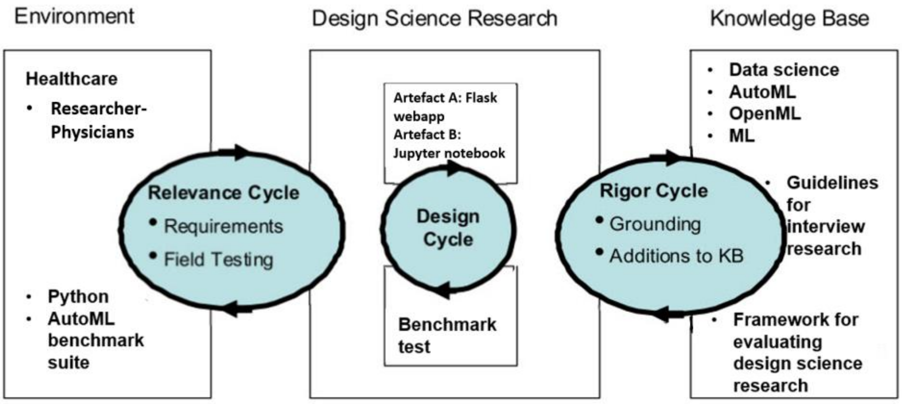


Figure 1*:* Adapted Information Systems Research Framework

2.2 Literature study

A literature study answered the first two research questions. The literature study is part of the rigor cycle and is conducted by back and forth snowballing after selecting influential papers based on a set of search queries (Appendix 1: Search queries). We snowballed for at least two levels and used literature from recent AutoML conferences as starting points to find literature. Furthermore, we used literature that is written by the founders of the CASH problem to reverse-snowball as they were the first authors to address AutoML in literature. Snowballing is considered to be an efficient method for a literature study (Webster & Watson, 2002). Besides that, its results do not significantly differ from the results of a systematic literature review (Jalali & Wohlin, 2012). Due to the time constraints as well as the novelty of the topic of this research, we choose snowballing over a systematic literature review.

2.3 Benchmark test

A benchmark test answered the third research question as part of both the design- and rigor cycle. To benchmark the AutoML methods, we used the AutoML framework provided by Gijsbers et al. (2019). We used this open-source framework to ensure a reproducible benchmark test by using the default settings (Balaji & Allen, 2018). Four medical datasets from the OpenCC18 (Rijn van, 2019) are used to benchmark the AutoML methods. The OpenCC18 is the successor of the OpenML100 (Bischl et al., 2017), a collection of datasets selected for benchmarking. The requirements for inclusion in the OpenML100 are available in Appendix 2: Requirements for OpenML100 datasets. All tests have been run on Amazon Web Services using m5.2xlarge machines,[3]

[3] 32 GB memory, 8 vCPUs (Intel Xeon Platinum 8000 series Skylake-SP processor with a sustained all core Turbo CPU clock speed of up to 3.1 GHz). The operating system on these machines is Amazon Linux. https://aws.amazon.com/ec2/instance-types/m5/

to get constant circumstances and enough computing power for the AutoML methods. The machines are chosen to build on the work of Gijsbers et al. (2019).

All selected methods received a time budget of one hour in a 10-fold cross-validation set-up to create the best pipeline on the given datasets. The time limit is set on one hour, as longer runs do not significantly provide better results (Gijsbers et al., 2019). To baseline the performance of the AutoML methods in the benchmark test, we added a decision tree and a constant predictor. Following Gijsbers et al. (2019), we used Area Under the Receiver Operator Curve (AUROC) for scoring.

To validate the statements of Gijsbers et al. (2019) about performance improvement of the AutoML methods with a longer time budget, we have selected three methods to run again on the same datasets with a time budget of four hours. We chose one evolutionary algorithm (EA) and a Bayesian Optimization (BO) method based on their performance in the one-hour test. A third method was selected because its performance lagged significantly in the 1-hour test, to see if a larger time budget would help improve its score.

2.4 Requirements elicitation

Requirements elicitation was used to answer the fourth research question using interviewees that participated in the study as part of the relevance cycle. We describe the interviewees that participated in the next paragraph. The requirements were elicited using semi-structured interviews. We selected semi-structured interviews as the best method for requirements elicitation for three reasons. 1) Semi-structured interviews are considered to be the most effective way for requirements elicitation (Davis, Dieste, Hickey, Juristo, & Moreno, 2006); 2) It is an accepted method for conducting qualitative research in healthcare (Al-Busaidi, 2008); 3) Semi-structured interviews have the benefits of eliciting people’s own views and uncovering issues or concerns that have not been considered beforehand by the researcher (Pope, van Royen, & Baker, 2002).

To get a complete view of requirements in healthcare, we analysed five interviews that are part of an earlier, related, research project by De Vries (2018) for requirements elicitation.

For our interviews, we constructed an interview protocol following the guidelines for interview research (Castillo-Montoya, 2016). The interview protocol is available in Appendix 11.4. We transcribed all the interviews, which lasted between 30 and 45 minutes the transcripts are available in Appendix 11.5.

From the interviews, we elicited requirements. We sent the requirements to the interviewees for confirmation. After the confirmation, we categorised and analysed the requirements. We tested the functional capabilities of the AutoML methods that participated in the benchmark test against the elicited requirements.

Sample

All interviewees are related to the scientific department of a regional hospital in the Netherlands. The interviewees have decided to participate voluntarily and hold different roles and medical expertise within the hospital. The interviewees are active in the research fields of cancer, orthopedy, and cardiology and participate in medical research, either full-time or part-time. The sample consists of three women and two men. The sample is restricted to five interviewees for the interviews and the case study due to the time constraints of this study. An overview of the sample is available in Table 1.

Table 1: Sample description

Interviewee	**Experience**	**Speciality**	**Hospital type**
Interviewee 1	3 years	Oncology	Non-academic
Interviewee 2	6 years	Orthopaedic surgery	Non-academic
Interviewee 3	10 years	Research, no specific field	Non-academic
Interviewee 4	18 years	Cardiology	Non-academic
Interviewee 5	22 years	Immunology	Non-academic

2.5 Evaluation of AutoML artefacts

As part of the relevance cycle, we answered the fifth research question. In this process, the interviewees evaluate the created artefacts. Design evaluation is crucial to design science (Hevner et al., 2004). To evaluate the artefacts, we combine two of the methods that Hevner et al. (2004) suggest: an observational and experimental evaluation. We study the artefacts in depth using mock data as part of the experimental set-up. However, the interviewees were allowed to use their data to execute the experiment if preferred. The observational part of the evaluation method is used to study the artefacts in a simulated business environment during the experiment.

According to Hevner et al. (2004), observational evaluation is part of a case study. The choice for a case study is supported by Roethlisberger (1977), as he argues that case-research is well-suited for problems in which research about phenomena is at the early and formative stages, as is the case of the application of AutoML for KD in the medical domain. Furthermore, case-study research allows answering “how” questions (Benbasat, Goldstein, & Mead, 1987).

To evaluate the artefacts, we used artificial summative evaluation as part of the framework for evaluating design science research (Venable et al., 2016). We evaluated the artefacts on the user-story categories from the previous research question. To be able to evaluate the artefact properties, we have created refined hypotheses (Offermann, Levina, Schönherr, & Bub, 2009). Based on these refined hypotheses, we created an artefact evaluation protocol available in Appendix 11.7. The first artefact presented to the interviewees will differ per participant to decrease learning bias. The interviewees evaluated the artefacts based on the artefact evaluation protocol.

2.6 Conclusion

We answer all sub-research questions as part of one of the cycles in the design science research method. A literature study is conducted to provide a theoretical foundation for this study. After that, two research questions are asked to be able to select the ideal AutoML method based on functional requirements and performance. Finally, we evaluate the created artefacts containing an AutoML method with the interviewees.

A summary of each research question, including the corresponding design cycle type and execution, is listed in Table 2.

Table 2: Research approach overview

Research question	Cycle	Practical execution
RQ1	Rigor	Literature review
RQ2	Rigor	Literature review
RQ3	Design & Rigor	Compare AutoML methods in a benchmark test.
RQ4	Relevance	Elicit requirements from literature. Conduct interviews. Elicit requirements from interview transcripts. Revise requirements based on interviewee feedback.
RQ5	Relevance	Evaluate the implementation of the selected AutoML method.

3 Theoretical background

This chapter describes the theoretical background of this research. The first section elaborates on the inception of data science, various definitions of data science, and Applied Data Science (ADS) as a sub-discipline of data science. The second section discusses seven Knowledge Discovery (KD) methods. The third section covers the application of data science in the healthcare domain. The final section provides a conclusion of this chapter to scope the remainder of this research by answering the first research question.

3.1 Data science

Within the statistics community, there has been a lively debate about the role of statisticians in data science, as statisticians long dominated data science (Chang et al., 2018). John Tukey was the first statistician to argue in 1962 that statisticians should look beyond the theory of statistics (Cao, 2017; Donoho, 2017). He identified four driving forces in statistics that have led to what we know as data science today: 1) The formal theory of statistics; 2) Accelerating developments in computers and display devices; 3) The challenge, in many fields, of more and ever-larger bodies of data; 4) The emphasis on quantification in an ever-wider variety of disciplines (Tukey, 1962). In the remainder of this section, we describe different views on data science and conclude with our definition based on the literature. After that, we will discuss views on ADS as a sub-discipline of data science and conclude with a definition.

The definition of data science

Since Tukey's argument in 1962, the term data science was first coined by Peter Daur in 1974 (Cao, 2017). Since then, there has been an active discussion on the definition of data science. Jeff Wu (1997) was a statistician who asked the question if statistics is the same as data science in his lecture: 'Statistics = Data Science?'. Many disagree with Wu on this and see statistics as one of the fundamental parts of data science but argue that data science is broader than just statistics. Some view data science as a culture, whereas others see it as an interdisciplinary field. We agree with the latter and view data science as an interdisciplinary field of domain knowledge, mathematics & statistics, and computer science.

Data science as an interdisciplinary field

There are four perspectives on data science as an interdisciplinary field. All perspectives state that data science is the intersection or superset of three disciplines but differ in what these disciplines are. Cao defines data science as *"a new interdisciplinary field that synthesises and builds on computing, communication, management, and sociology to study data"* (2017, p. 9).

Yu (2014), just like Cao, defined data science as an interdisciplinary field. She states that Data science has three pillars: computer science, statistics/mathematics, and domain knowledge. She leaves it in the middle whether data science is the union or the intersection of these three fields. She disagrees with Cao over the fields that create the interdisciplinary field of data science. Yu mentions statistics/mathematics as part of the triangle, whereas Cao does not.

Blei and Smyth (2017) state that there are three perspectives on data science: The statistical perspective, the computational perspective, and the human perspective. They state that the potential of data science is in crossing the boundaries of each perspective. Furthermore, they state that it is more than a combination of the disciplines; it is about fitting them in a broader framework to answer discipline-specific questions. Hence, Blei and Smith agree with both Cao and Yu that data science is an interdisciplinary field. We can map the three views presented by Blei and Smith to the three pillars that make up Data science as described by Yu when we argue that the human perspective in a specific discipline is the equivalent to domain knowledge.

Chang et al. (2018) provide a Venn diagram, depicted in Figure 2, in the NIST Big Data Interoperability framework building on an earlier version of Pritzker and May (2015). It shows three pillars of data science: math & statistics, domain expertise, and computer science. They state that: *"Data Science is a super-set of the fields of statistics and DM and machine learning (ML) to include the analysis of big data."* (Chang et al., 2018, p. 23). We find that these three fields that are similar to those mentioned by Yu, Blei and Smith. Furthermore, they agree with the notion that data science is a multidisciplinary field.

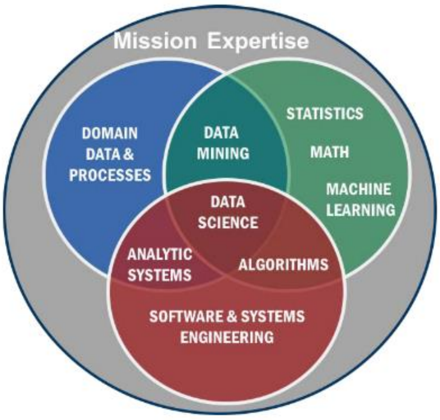


Figure 2: Data science and its sub-disciplines (Chang et al., 2018)

The Data Science Body of Knowledge (DS-BoK) agrees with the definition of Pritzker and May after evaluating relevant bodies of knowledge regarding data science (Demchenko, Manieri, & Belloum, 2017). The goal of the DS-BoK is to create a competency framework and a standard for the Data science community. This standard and framework should be for both professionals and academics but is still in a draft phase.

In the literature we see a consensus on the definition of data science: it is an interdisciplinary field, and it consists of three pillars: domain knowledge, mathematics & statistics, and computer science (Chang et al., 2018; Demchenko et al., 2017; Pritzker, P., and May, 2015; B. Yu, 2014). Hence, we use the definition of Chang et al. (2018)

in the remainder of this research when we discuss data science *"Data Science is a super-set of the fields of statistics and data mining and machine learning to include the analysis of big data."* (Chang et al., 2018, p. 23).

Applied Data Science

The DS-BoK (Demchenko et al., 2017) describes the practitioners of data science as data scientists. Because of the reduction of economic costs, algorithmic advances (see Chapter 4) and improved software for data analysis, data scientists are becoming increasingly important, and the demand for these specialists is soaring (Carmichael & Marron, 2018). Davenport and Patil (2012) recognised the growing demand for data scientists earlier and branded the job of data scientist as: "*the sexiest job of the 21st century*". Besides Davenport and Patil, many others in literature predicted and noted a growing shortage of data scientists (Gibert et al., 2018; Harris et al., 2017; Manyika et al., 2011; Markow et al., 2017). This shortage of data scientists gave birth to a sub-discipline of data science: Applied Data Science (ADS). ADS aims to enable domain experts to do data science by developing information systems suited to them.

ADS covers a part of the three-pillar Venn diagram in Figure 2 as it is a sub-discipline of data science. There is not such an extensive discussion around the formal definition of ADS as is the case for data science. We review three takes on the definition of ADS.

An interdisciplinary focus group of scientists on ADS from Utrecht University defines ADS as: *"All applications of Data science methodology and engineering to scientific domains. Including the fundamental research from which new methodologies and tools are created and studied from an application-oriented perspective for one or more domains"* (Eijnatten et al., 2017, p. 4).

The KDD conference has a distinct call for ADS papers next to the call for data science papers. ADS papers have a specific distinction in this call: *"ADS papers focus on real-world problems and systems that are deployed or are in the process of being deployed."* (KDD, 2019, p. 1).

In the paper Power to the people! Spruit & Jagesar define ADS as: *"The knowledge discovery process in which analytical applications are designed and evaluated to improve the daily practices of domain experts."* (Spruit & Jagesar, 2016, p. 1).

All three views on ADS emphasise the use of tools or applications to solve data-driven, domain-specific problems. We use the definition of Spruit and Jagesar (2016) for ADS in the remainder of this research, as it is the most comprehensive definition and builds on the definition of data science provided above. The KD process mentioned in this definition is elaborated on in the section on Knowledge Discovery Methods. Figure 3 displays which part of data science is covered by ADS.

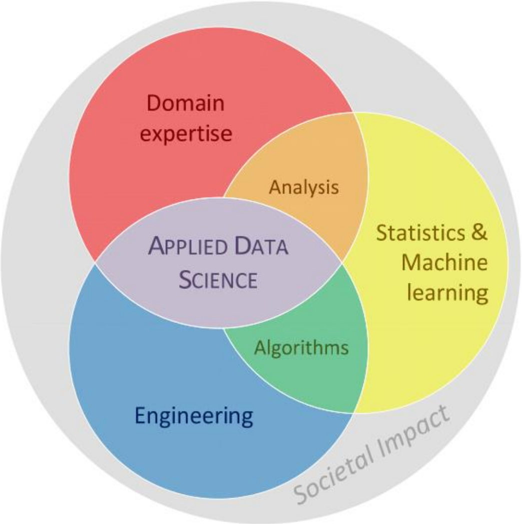


Figure 3*:* ADS in Context (Spruit & Jagesar, 2016)

3.2 Knowledge Discovery Methods

The KD process in ADS is often referred to as Data Mining (DM). Fayyad et al. define DM as: *"The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data."* (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a, p. 30). We consider the KD and DM processes as the same processes. All DM methodologies have evolved from the Knowledge Discovery in Databases (KDD) process or its benchmark successor, the Cross-Industry Standard Process for Data Mining (CRISP-DM) (Mariscal, Marbán, & Fernández, 2010). In this section, we discuss seven available methods. KDD, CRISP-DM and, SEMMA are discussed because they are well-established methods in DM (Azevedo & Santos, 2008; Mariscal et al., 2010; Shafique & Qaiser, 2014).

Additionally, we evaluate The Three Phases Method (3PM) and the Advanced Analytics project methodology, as these are both methods that focus on value delivery for a client organisation. Finally, we discuss the Epicycles of Analysis and the Human-Centred process. The focus of the latter two is on self-service data science. Thus, these could be more useful for interpretability by domain experts, as most practitioners do not use scientific literature due to its limited availability and complexity (Vlaanderen, Brinkkemper, & van de Weerd, 2012).

KDD

KDD was the first structured method for knowledge discovery in databases, described by Fayyad, Piatetsky-Shapiro, and Smyth (1996b). The focus of the method is on extracting knowledge from databases from a scientific perspective and hence might be perceived as technical.

KDD consists of nine steps: learning the application domain, creating a target data set, data cleaning and pre-processing, data reduction and projection, choosing the function of data mining, choosing data mining algorithm(s), data mining, interpretation, and discovering knowledge (Fayyad et al., 1996b). Figure 4 summarises the KDD process**.** The visualisation shows that the process is iterative.

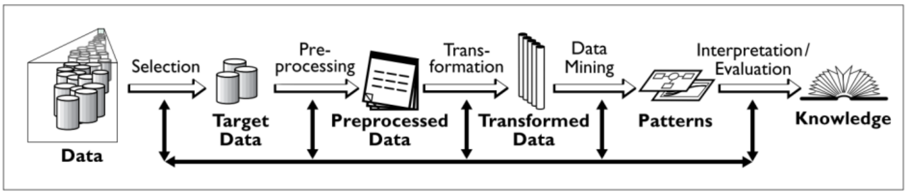


Figure 4: KDD process (Fayyad et al., 1996b)

CRISP-DM

CRISP-DM is a method developed by a consortium of DaimlerChrysler, NCR and, SPSS using Clementine, a tool for data mining. The goal of the method is to have a standardised process for the usage of data mining within organisations (Chapman et al., 2000). CRISP-DM is the most widely used data mining method in practice (Mariscal et al., 2010; Piatetsky, 2014).

CRISP-DM consists of six phases and is flexible but carefully described based on the hierarchical levels below the phases. The six phases in the CRISP-DM method are business understanding, data understanding, data preparation, modelling, evaluation, and deployment. Each phase consists of one or more generic tasks, specific tasks, and process instances. Generic tasks are created to cover all possible data mining situations in a phase. Specific tasks are created to describe how generic tasks should be completed, given a specific situation and goal. The process instances record what happened in an actual engagement of the method (Chapman et al., 2000). Figure 5 visualises the hierarchy of the CRISP-DM methodology**.**

An example of a generic task for business understanding is to determine business objectives. A specialised task that is accompanying this generic task is identifying key persons in the business.

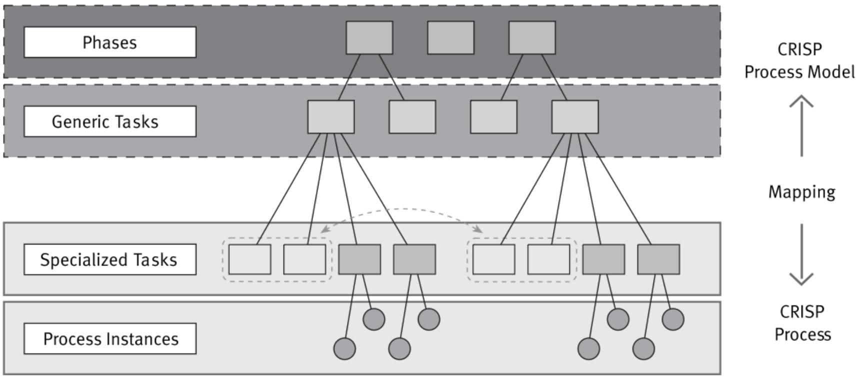


Figure 5: CRISP-DM method hierarchy (Chapman et al., 2000)

According to Chapman et al., one does not have to follow the phases in a specific order. Although Figure 6 depicts an order. Furthermore, the stages are not to be completed once, as the method is iterative.

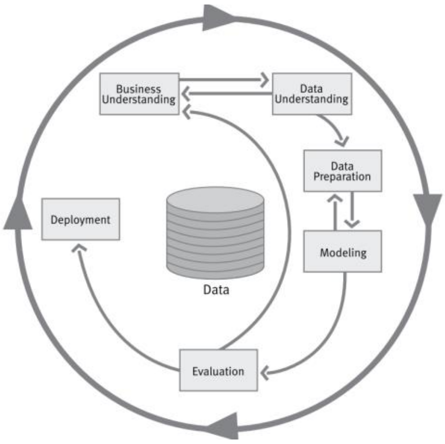


Figure 6: CRISP-DM reference model (Chapman et al., 2000)

SEMMA

The SAS Institute developed SEMMA, SEMMA stands for sampling, exploring, modifying, model, and assessing the data. It is developed to complement their ERP package and their analysis tool Enterprise Miner (Rohanizadeh & Moghadam, 2009). Hence, we conclude that SEMMA focusses on data mining in organisations that use SAS. Although SAS presents SEMMA as a method, it solely focusses on the technical part of data mining, the organisation side of the process is neglected (Marbon, Mariscal, & Segovi, 2009). The focus of SEMMA is on sampling the dataset at hand, as can be deduced from the acronym’s meaning (SAS, 2018). As can be deducted from Figure 7, SEMMA is a linear method.

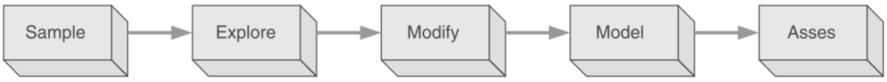


Figure 7: SEMMA method (Mariscal et al., 2010)

Three Phases Method (3PM)

Vleugel et al. (2010) developed the 3PM for organisations that outsource their data mining process and organisations that deliver value by performing the DM process. Building on both CRISP-DM and KDD the method was constructed with a clear distinction in roles for the case company and the third party who executes the data mining. The method is created to support both companies in the outsourcing process (Vleugel et al., 2010).

The method consists of three phases: data retrieval, data analysis, and results implementation. The goal of the first phase is to align the case company and the third party. The data analysis phase is about selecting the right data mining technique to solve the case company's questions. The final phase aims to embed the results in the business processes or to deliver a recommendation report. All phases consist of activities with their corresponding sub-activities and deliverables. 3PM is an iterative method; all its activities are executed by either one party or collaboratively (Ooms, Spruit, & Overbeek, 2019). The 3PM is an iterative method as can be deducted from Figure 8**.**

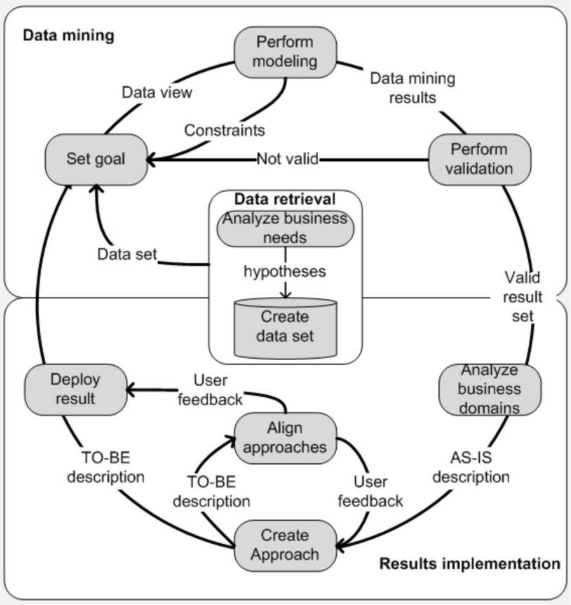


Figure 8: 3PM (Vleugel et al., 2010)

Advanced Analytics project methodology

The Advanced Analytics project methodology is designed to have a standard process for performing projects at clients of Deloitte. The focus of the method is on the assignment of tasks within the project team and the deliverables for the client in each phase of the project. The method consists of five project phases at the client: 1) Problem framing; 2) Acquire and understand data; 3) Prepare and structure data; 4) Analysis and modelling; 5) Report and implement (Deloitte, 2016). Besides the phases during the project, there are two phases outside of the project: Proposal and preliminary activities, and project evaluation. These two phases are not performed at the client but are crucial for the business processes within Deloitte. All deliverables and the division of roles within a project team are carefully described, as well as the criteria to go to the next phase of the project. The methodology is an iterative method, as is shown in Figure 9.

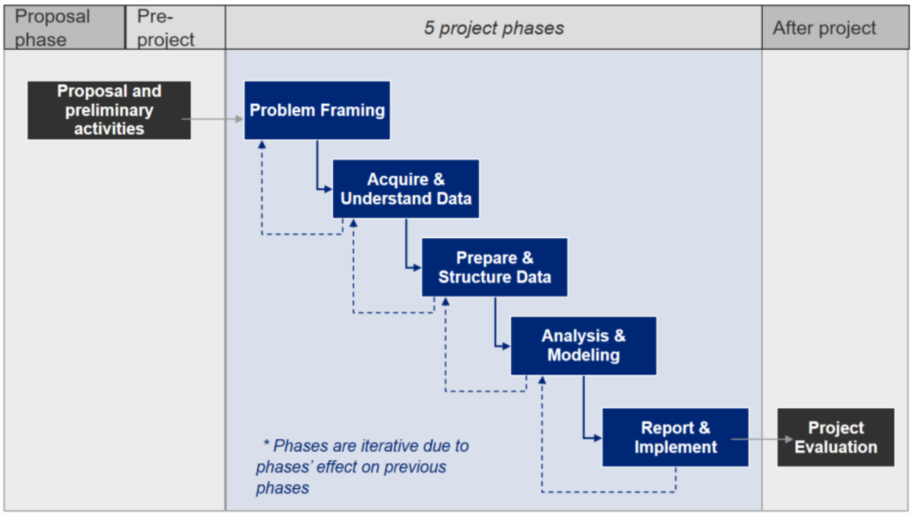


Figure 9: Advanced Analytics project method (Deloitte, 2016)

Epicycles of analysis

The Epicycles of analysis is a general framework to take in mind when working with data. The epicycles of analysis are for ‘anyone who works with data’, researchers and business people, professionals and amateurs. It describes five core activities: Stating and refining the question, exploring the data, building formal statistical models, interpreting the results, and communicating the results (Peng & Matsui, 2016). At each of the activities, it is crucial to engage in three steps: 1) Setting expectations 2) Collecting information and compare these to the expectations. If these do not match: 3) Revising expectations or fix the data, so your data and your expectations match (Peng & Matsui, 2016). The epicycles of analysis are an iterative method as can be inferred from Figure 10**.**

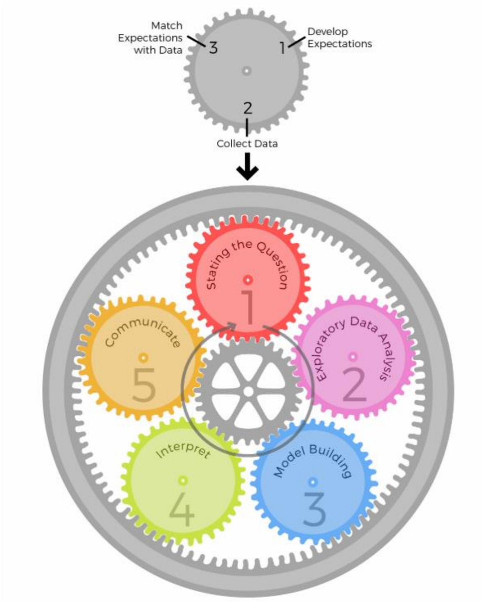


Figure 10*:* Epicycles of analysis (Peng & Matsui, 2016)

Human-centred process

The human-centred approach is an elaboration on the KDD method described above. Because the KDD method can be complicated to solve real-life tasks Brachman and Anand (1996) developed the human-centred process. Gertosio and Dussauchoy (2004) describe the human-centred process as the realistic steps of the KDD approach. The focus of the human-centred process is on the decisions that the data scientist has to take. Furthermore, it describes general tools and deliverables for the process steps. The six steps in the process are similar to those in KDD process: 1) Task discovery; 2) Data discovery; 3) Data cleaning; 4) Model development; 5) Data analysis; 6) Output generation. Figure 11 depicts the iterative nature of the method, and the process, inputs, and outputs.

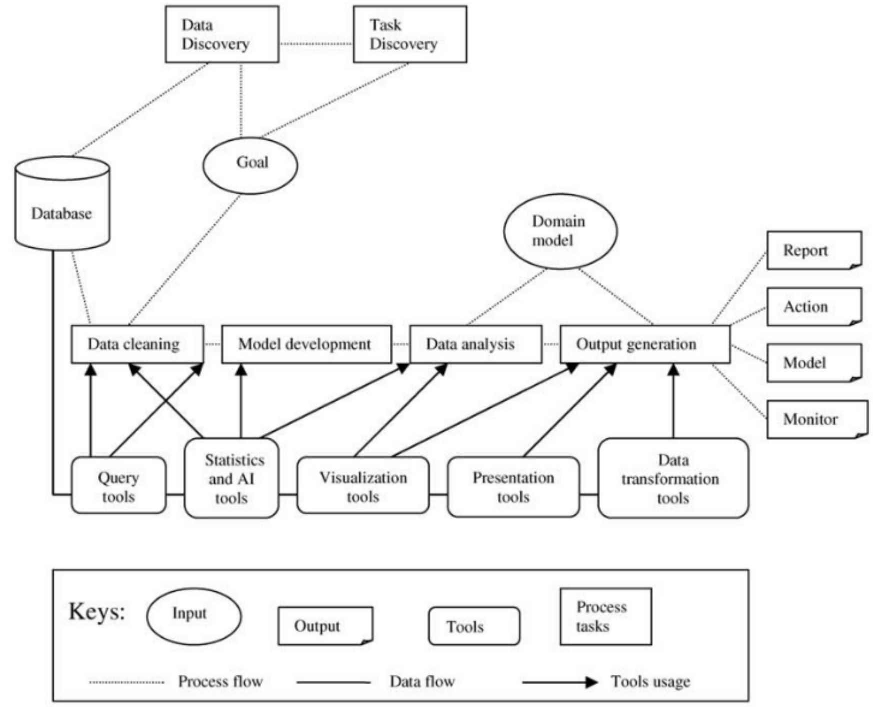


Figure 11: Human-centred process (Gertosio & Dussauchoy, 2004)

Data mining method overview

In this section, we described seven methods: Three methods focus on data mining conducted by a single party. Two methods focus on delivering value for a client organisation when outsourcing the data mining process. The last two methods focus on the decisions that have to be made by practitioners during the KD process.

Besides SEMMA, all models described above are iterative. This finding is not surprising, as iterative development is efficient in KD (Larson & Chang, 2016). Besides that, we find that the Deloitte phases during the project identically resemble the CRISP-DM phases. This observation is unsurprising, as Mariscal et al. (2010) observed that most methods evolved from either CRISP or KDD and CRISP-DM is the most widely used method in practice (Piatetsky, 2014). The addition of the extra phases is to support the business processes of the company.

We choose CRISP-DM as the default KD method in this research. It is the base of most methods (Mariscal et al., 2010) and is the most used method in data science (Piatetsky, 2014). CRISP-DM is highly flexible to adapt to healthcare due to its hierarchical structure (Koh & Tan, 2005). When discussing the KD process of the ADS definition, we are referring to CRISP-DM in the remainder of this research. A tabular overview of the methods, the level of detail of the method description in literature and method characteristics are in Table 3.

Table 3: Data mining method overview

Method	Process type	Description	Characteristics
KDD	Iterative	General	For on scientific discovery in datasets, technically oriented
CRISP-DM	Iterative	Detailed	Focus on industry applications, the most used model for data mining. Easily adaptable.
SEMMA	Linear	General	Focus on SAS applications covers the technical part of the process.
3PM	Iterative	Detailed	Focus on two parties in the data mining process
Advanced Analytics project methodology	Iterative	Detailed	Focus on delivering value for the client, deliverable focussed. Sets requirements for going to the next step.
Epicycles of analysis	Iterative	General	General method focussed on non-experts in data mining
Human-Centred process	Iterative	General	Emphasis on the perspective of the data scientists. Shows input and output

3.3 Knowledge Discovery in healthcare

Compared to other industries, the healthcare industry has been late in the adoption of data mining (Koh & Tan, 2005). Hence, the healthcare industry has not yet grasped the full potential of DM (Feldman et al., 2012; Lee & Yoon, 2017; Malik et al., 2016; X. Wang et al., 2018; Y. Wang & Hajli, 2016). Well documented reasons for this are: a lack of understanding of the impact on strategic and managerial perspective (Raghupathi & Raghupathi, 2014; X. Wang et al., 2018); the complexity of the healthcare system (Spruit & Lytras, 2018); costs of adoption (Dedding, 2018; Neff, 2013), patient privacy (Neff, 2013; Patil & Seshadri, 2014), data quality in Electronic Health Records (EHR) (Koh & Tan, 2005; Lee & Yoon, 2017) and integration of various data sources (Koh & Tan, 2005; Y. Wang et al., 2016).

Besides these barriers to adoption, healthcare can benefit from implementing analytics in three ways: cost reduction (Bates, Saria, Ohno-Machado, Shah, & Escobar, 2014), increased operational efficiency (Malik et al., 2016) and increased patient satisfaction (Kimberly & Cronk, 2016). Viewing hospitals and their suppliers as a supply chain and operations costs can be used to reduce costs (Malik et al., 2016). Another option is identifying high-risk patients at an early stage (Bates et al., 2014). Approaching healthcare as a value delivery process could help improve the patient's satisfaction (Kimberly & Cronk, 2016). Finally, implementing patient-centric analysis is proposed as a benefit for healthcare by creating unique treatment plans for each patient. Implementing a patient-centric analysis could reduce costs and improve both patient satisfaction and operational efficiency (Chawla & Davis, 2013; Duan, Street, & Xu, 2011; Martin & Félix-Bortolotti, 2014).

DM is used in healthcare by researcher-physicians for research purposes and KD. Researcher-physicians are former medical students who perform research in order to obtain a medical PhD degree or people who do research next to their clinical practice (Ley & Rosenberg, 2005). As medical students have limited knowledge in statistics, this is often the case for researcher-physicians. Skills outside of the medical domain in statistics or computer science are needed for clinical research but not mastered by researcher-physicians (Sung et al., 2003). The need for clinical data analysis is surging, but the lack of skilled professionals is hindering this development (Markow et al., 2017).

3.4 Conclusion

In this chapter, we defined ADS as *"The knowledge discovery process in which analytical applications are designed and evaluated to improve the daily practices of domain experts."* (Spruit & Jagesar, 2016, p. 1). We refer to CRISP-DM as the KD process in this definition. Because CRISP-DM is the most widely adopted method in data science (Piatetsky, 2014), it is easily adaptable and many different models have been derived from CRISP-DM (Mariscal et al., 2010). Finally, we described the possibilities of applying analytics in healthcare. Because of the surge in clinical data analysis and the lack of data scientists, there is enormous potential for the application of KD and ADS in healthcare (Markow et al., 2017; X. Wang et al., 2018). While designing new analytical systems, constraints and barriers to adoption need to be taken into account to increase the adoption rate with healthcare professionals (Neff, 2013).

4 Automated Machine Learning

This chapter describes AutoML by discussing its origin, related concepts, and definition. Furthermore, it provides an overview of AutoML methods. The first section introduces ML and related concepts to provide context for the remainder of this chapter. The second section describes AutoML, including related concepts. The third section describes a selection of AutoML methods and their contribution to literature. The last section provides a synthesis of the AutoML methods discussed in section 4.3.

4.1 Machine Learning

Machine Learning (ML) is formally defined as: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."* (Mitchell, 1997, p. 2). ML first originated in the 1950s as a sub-discipline of statistics (Portugal, Alencar, & Cowan, 2018) and is often characterised as teaching computers to learn from experience by developing algorithms (Breiman, 2001; Jordan & Mitchell, 2015; Obermeyer & Emanuel, 2011). Another take on ML is teaching computers how to learn without being explicitly programmed (Olson & Moore, 2016). Next, we will discuss essential concepts in ML: The ML pipeline, Learning functions, Model evaluation, and Model tuning.

ML Pipeline

Data scientists often refer to a 'pipeline' when they are talking about the result of their work. This pipeline is an analogy for the process through which the data progresses during data analysis. A pipeline consists of data collection, data pre-procession, and analytical processing. When projecting a pipeline on CRISP-DM (section 3.2), it covers the phases of data preparation, and modelling (Aggarwal, 2015). ML techniques are often used as the building blocks of the analytical processing part of the pipeline, as shown in Figure 12. These building blocks consist of algorithms and pre-processing methods. Data scientists spend most of their time on data pre-processing and the configuration of the ML techniques (Olson, Bartley, Urbanowicz, & Moore, 2016).

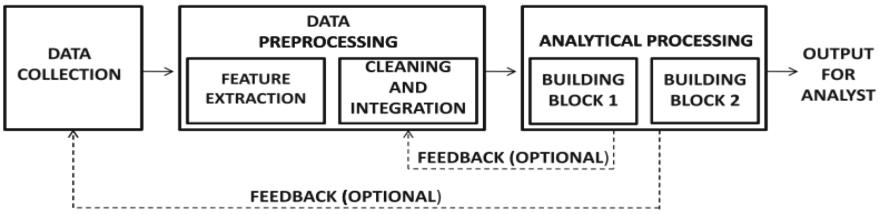


Figure 12: DM Pipeline (Aggarwal, 2015)

Learning functions

ML techniques have different ways to learn patterns from data. In ML, there are four types of learning functions: supervised, unsupervised, semi-supervised, and reinforcement learning (Antonoglou et al., 2015; Gareth, Witten, Hastie, & Tibshirani, 2013). For the same purpose, different learning types are suitable. Learning types are families of algorithms. For example, classification can be done by both supervised learning using logistic regression, and unsupervised learning using k-means clustering.

Supervised learning is learning from a set of input variables, $X = \{x_1, x_2, \dots, x_k\}$ for which the output label Y is known. Supervised learning is used to gain understanding in which variables in x_k in X influence the outcome Y, as it does not deliver the best predictive performance on a dataset (Gareth et al., 2013; Kotsiantis, 2007). Examples of supervised learning models are linear and logistic regression, decision trees, and Naïve Bayes classifiers (Kotsiantis, 2007).

Unsupervised learning is the opposite of supervised learning. There is no output label Y for the input variables in X. The goal of unsupervised learning is to find similar cases or to find similarities between variables in X. Examples of unsupervised learning models are principal component analysis and K-means clustering (Gareth et al., 2013).

Semi-supervised learning is a form of learning in which a dataset contains both labelled and unlabelled data. Semi-supervised learning is used in situations where it is cheap to find the input variables in X and expensive to label the data with output variable Y (Gareth et al., 2013; Zhu, 2005). With semi-supervised learning, both labelled and unlabelled data is used to train the algorithm. Under the right assumptions, it is possible to use the unlabelled data to improve the performance of the learning model (Zhu, 2005).

Reinforcement learning is training an algorithm by a feedback loop. The algorithm does not know what its goal is but gets both positive and negative feedback based on the decisions that it takes. In this way, the system is learning from experience (Antonoglou et al., 2015). This way of learning is often used to teach a computer to play games.

Model tuning

The performance of each algorithm can be fine-tuned using model parameters and hyperparameters. Model parameters are variables in X which are in- or excluded in the model. An example of tuning model parameters is the in- or exclusion of gender for a model on credit card fraud (Gareth et al., 2013). Hyperparameters are used to configure the selected algorithm to optimise performance (Mohr, Wever, & Hüllermeier, 2018). The number of trees in a random forest is an example of a hyperparameter setting (Olson et al., 2016). The actual model consists of a selected algorithm and its tuned (hyper)parameters (Bergstra & Bengio, 2012). As model tuning is complex and time consuming for data scientists, research has focussed on its automation (Bergstra, Bardenet, Bengio, & Kégl, 2011; Bergstra & Bengio, 2012; Thornton et al., 2013).

Model evaluation

In the world of a data scientist, there is no free lunch (Wolpert & Macready, 1996). That is, no model performs best on all datasets. To evaluate models, data scientists divide the dataset at hand in a training set and a validation set. After training, data scientists test a model on a validation set to find out if a model generalises well to unseen data. The validation is executed to make sure that the created model has not adapted too much on extremes within the training set.

A model that is sensitive to extremes in a training set is said to have high variance and low bias. If a model predicts the mean of the outcome variable, it is said to have high bias and low variance. The goal of training the model is to get the right balance between bias and variance, so the model performs well on unseen data. This balancing act between model complexity and model performance is the bias-variance trade-off (Figure 13) (Gareth et al., 2013; L. Yu, Lai, Wang, & Huang, 2006).

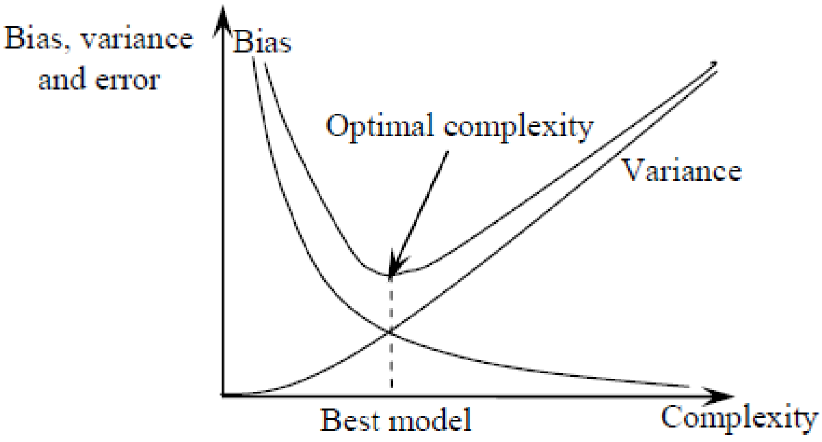


Figure 13: Bias-variance trade-off and model error (L. Yu et al., 2006)

Besides the choice of model complexity, there are two other choices to be made by the data scientist when evaluating a model: the division of data into a training- and validation set, and the evaluation metric. The method of data division and the evaluation metric is essential, as it has a significant impact on the performance of the algorithm on an unseen dataset (Mohr et al., 2018). To divide a dataset into a training- and validation set for evaluation, one can use a single divide, k-fold cross-validation, bagging or boosting (Gareth et al., 2013). Examples of evaluation metrics are accuracy, precision, recall, F1-score, area under the curve, root mean squared error, and the R^2 adjusted statistic (Gareth et al., 2013).

4.2 Automated Machine Learning

AutoML aims to automate the creation of an ML pipeline in order to make ML accessible to non-experts and to improve the reproducibility of ML solutions (Feurer et al., 2015; Hutter et al., 2019; Thornton et al., 2013). AutoML aims to improve the quality of solutions as some AutoML systems can outperform human experts in configuring pipelines. (Jin, Song, & Hu, 2018; Sá de, Pinto, Oliveira, & Pappa, 2017).

Based on the definition of ML in section 4.1, Quanming et al. (2018) define AutoML as: *"AutoML attempts to construct machine learning programs (specified by E, T, and P in the definition of ML), without human assistance and within limited computational budgets.*". From this definition, we can derive that there is a focus on automating the construction of the ML pipeline with the constraint of a computational budget. Next, we will discuss essential concepts in AutoML: The CASH problem, search strategies and the architecture of AutoML systems.

The CASH problem

AutoML has converged from the fields of automatic model selection and hyperparameter configuration. Thornton et al. (2013) were the first to combine both model selection and hyperparameter configuration. They did so by defining the combined algorithm selection and hyperparameter optimisation problem (CASH). The CASH equation, available in Equation 1, consists of algorithms A = {A(1), ... , A(n)} with an associated hyperparameter space Λ(1), ... , Λ(n), a loss function $\mathcal{L}$, and a dataset D. The dataset D is divided into a training and validation set for each fold, denoted by *k*. The goal of the CASH problem is to select an algorithm A* as the optimal value for A and associated hyperparameters λ* as the optimal values of the hyperparameters that minimise the loss on the given dataset.

$$A^* \in \underset{A \in \mathcal{A}}{\operatorname{argmin}} \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}(A, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{valid}}^{(i)})$$

Equation 1: CASH problem (Thornton et al., 2013).

By defining the CASH problem, Thornton et al. (2013) defined the search space in which AutoML methods have to work: all combinations of algorithms and their hyperparameter configuration. As algorithm evaluation is expensive in terms of computing power and time, it is vital to use an efficient search strategy to find a solution to the CASH problem.

Search Strategies

Due to the budget constraint in AutoML and the high costs of evaluation, it is crucial to use the optimal search strategy for pipeline construction. As the search space for AutoML systems is highly dimensional, it is not feasible to use brute-force search strategies. This section describes different search strategies. First, we discuss grid and random search. Second, we discuss Bayesian Optimization (BO) methods. Third, we discuss warm-starting the search process. Fourth, we discuss Evolutionary Algorithms (EA) as a strategy for search. Finally, we discuss a way to search for suitable neural architectures.

Grid and Random search

Grid search structurally explores the combination of different dimensions in set intervals. Random search does the same but with random intervals. Both grid- and random search are slightly more efficient than a manual search for hyperparameter optimisation (HPO) (Bergstra & Bengio, 2012). Grid and random search are slightly better than humans as they do not use feedback from previous loops and thus spend less time evaluating infeasible ranges of hyperparameter values.

Random search is considered to be more effective than grid search. When one parameter is more important than the other parameter in a search space with two axes, random search is more likely to find optima as it tests on nine different points on each parameter instead of three points for each parameter, as is illustrated with the dots on green surface in Figure 14 (Bergstra & Bengio, 2012).

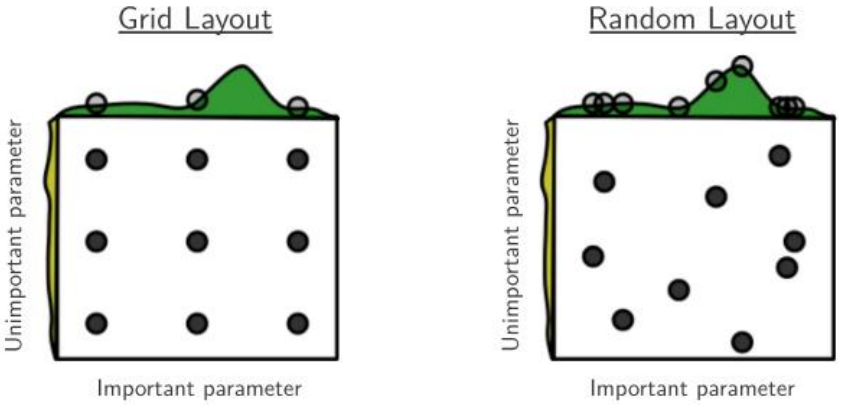


Figure 14: Grid vs Random search (Bergstra & Bengio, 2012)

Bayesian Optimization

Bayesian Optimization (BO) methods use a probabilistic model based on expectations and past experiences to create a new model in the search process. BO starts with a simple function called a ‘surrogate’. This function is iteratively improved based on the scores of evaluating the surrogate model. (Dewancker, McCourt, & Clark, 2015; Fenton, 2019).

Sequential Model-Based global Optimization (SMBO) is a BO method used in AutoML (Dewancker et al., 2015; Hutter, Hoos, & Leyton-Brown, 2010). SMBO first selects an algorithm before selecting the hyperparameter configurations. In SMBO implementations two types of probabilistic models are used: Gaussian processes (Bergstra et al., 2011; Rasmussen & Williams, 2006), and Tree Parzen Estimators (TPE) (Bergstra et al., 2011; Dewancker et al., 2015). Sequential Model-based Algorithm Configuration (SMAC) and Random Online Adaptive Racing (ROAR) (Hutter et al., 2010) are adaptations of SMBO to improve the performance of searching for the right configurations based on random forests. SMAC is an extension of ROAR and is often used in the first AutoML methods (Hutter et al., 2010). All SMBO variants work with a feedback loop of expected improvement based on BO. In each iteration, they evaluate where to search in the search space to get the most substantial expected improvement on the score of the current algorithm-hyperparameter configuration. Figure 15 illustrates BO on a one-dimensional function. The dotted line demonstrates the objective function, the black dots demonstrate points in which the objective function is evaluated, the solid line demonstrates the function created by BO. The blue area depicts the guessed uncertainty ratio and the orange area depicts amount of information that can be gained by a new observation (Hutter et al., 2019).

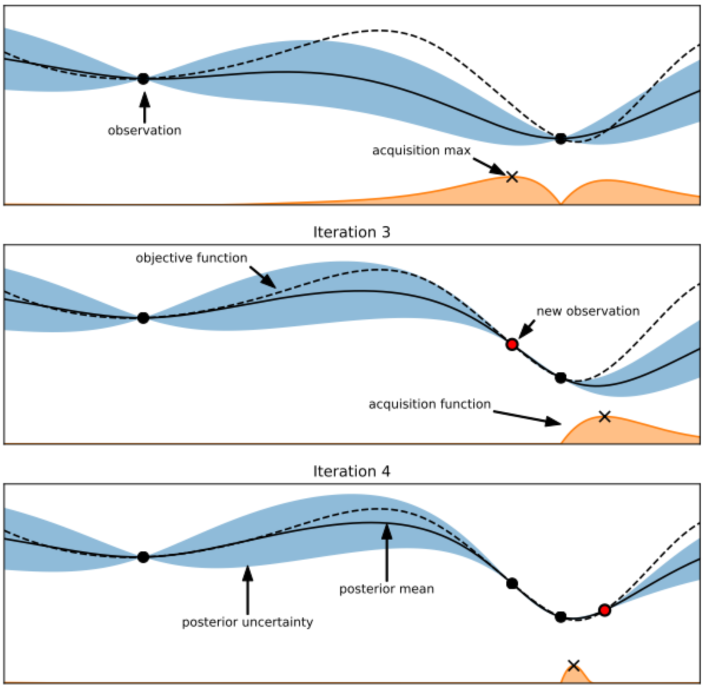


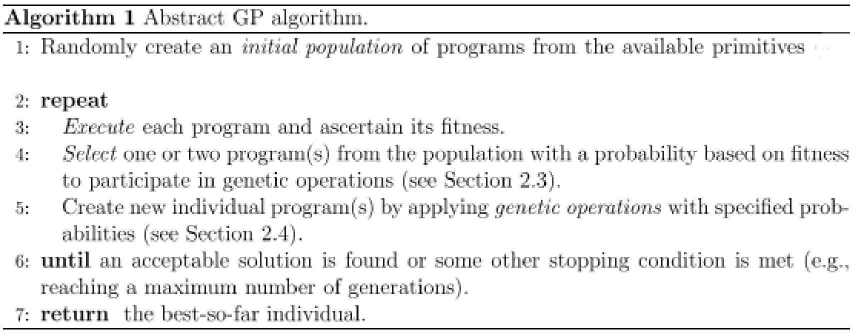
Figure 15: Bayesian Optimisation (Hutter et al., 2019)

Warm-start

Warm-starting is a strategy to speed search in HPO for BO methods. Based on statistical similarities between the dataset at hand and previously seen datasets, it creates a surrogate model (Feurer et al., 2015). The creation of a surrogate model is called meta-learning and is similar to how data scientists work. Data scientists use experiences and approaches that have worked on similar sets to speed up their search. (Brazdil, Carrier, Soares, & Vilalta, 2008). AutoML methods save the statistical properties of previous datasets in conjunction with their optimal algorithm configuration to apply meta-learning. Meta-learning has proven to speed up search significantly (Kalousis, 2002).

Evolutionary Algorithms

An alternative to the Bayesian methods for pipeline construction is Evolutionary Algorithms (EA)s. *‘EAs automatically solve problems based on a high-level statement of* *what needs to be done'* (Poli, Landon, McPhee, & Koza, 2007, p. 5). Genetic programming (GP) is a form of EA based on genetic evolution in biological processes. GP assembles primitives into individuals who make up populations. A population is a set of multiple constructed pipelines. Initial populations consist of individuals that represent random combinations of primitives. Before moving on to the next generation, the algorithm assesses each population member's fitness. In other words, the algorithm evaluates each pipeline's performance. The best pipelines of a population can then crossover with each other to create a new generation by randomly recombining pipelines. Another option is to alter the best pipelines by pre-defined mutations. Mutations can be the removal, replacement or addition of a primitive. GP is iterative and can use many generations to find an optimal solution. When the stopping conditions are satisfied, GP stops breeding and evolving (Poli et al., 2007). Figure 16 depicts an abstract GP algorithm.



Algorithm 1 Abstract GP algorithm.

```
Randomly create an *initial population* of programs from the available primitives
repeat
    *Execute* each program and ascertain its fitness.
    *Select* one or two program(s) from the population with a probability based on fitness
    to participate in genetic operations (see Section 2.3).
    Create new individual program(s) by applying *genetic operations* with specified probabilities (see Section 2.4).
until an acceptable solution is found or some other stopping condition is met (e.g.,
reaching a maximum number of generations).
return the best-so-far individual.
```

Figure 16: Abstract GP Algorithm (Poli et al., 2007)

AI Planning

Hierarchical Task Networks (HTN) is a form of AI planning that is used in AutoML (Hutter et al., 2019). The goal of HTN is to create a sequence of actions to perform a task (Nau et al., 2003). HTN connects sub-tasks to create task networks. There are three types of tasks: goal tasks, compound tasks, and primitive tasks. Goal tasks are properties that are to be made true. In the case of AutoML, this is creating a pipeline. Primitive tasks are the most granular tasks. These tasks can be completed directly by executing an action. In AutoML, this is selecting an algorithm or tuning a hyperparameter. Tasks that cannot be either a goal task or a primitive task are compound tasks. In AutoML this is optimising an algorithm and its parameters. The planning of the network of tasks is done by expanding tasks and iteratively resolving conflicts between the compound- and primitive tasks until a plan with primitive tasks is established to accomplish the common goal.

The architecture of AutoML systems

An AutoML system consists of a toolbox and a controller. The controller consists of an optimiser and an evaluator. The evaluator measures the performance of the models that are created by the optimiser and gives feedback to the optimiser based on their performance. AutoML systems differ in what items are in their toolbox to construct the pipeline. AutoML methods use supervised and reinforcement learning methods to enable the evaluator to provide feedback to the optimiser. Figure 17 depicts a visual representation of an AutoML architecture.

The toolbox defines the scope of an AutoML. The toolbox contains a set of pre-processing methods and algorithms. Most systems use Scikit-learn (Pedregosa et al., 2011) as their toolbox. However, AutoML systems are not limited to use a single library in their toolbox. They can use multiple libraries at the same time. An AutoML system is not able to function without a toolbox.

The optimiser searches the toolbox containing the pre-processing methods, algorithms, and their corresponding hyperparameter values. The optimiser tries to find the optimal configuration as fast as possible and uses feedback from the evaluator to select new configurations.

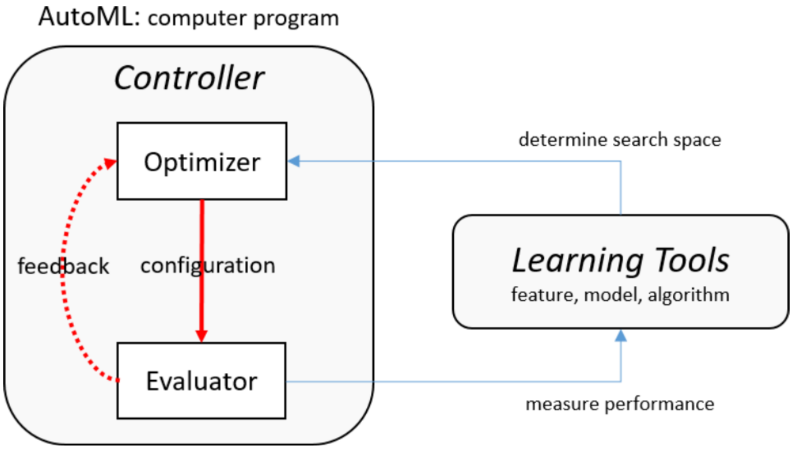


Figure 17: AutoML Architecture (Quanming et al., 2018)

Comparing AutoML systems

Since the inception of AutoML, there have been competitions to determine what the best method is. The learning task in these challenges was supervised classification (Hutter et al., 2019). These challenges have attracted many competitors who hand in tweaked versions of the AutoML methods that we will discuss in the next section. At the time of writing, the most recent winner is PoSH-AutoSklearn (Feurer, Eggensperger, Falkner, Lindauer, & Hutter, 2018), an extension of Auto-Sklearn (Feurer et al., 2015). In the AutoML challenge, a method gets a computational budget on a single machine with a set amount of CPU power. All methods get unseen datasets in different rounds of varying difficulty (Hutter et al., 2019).

To benchmark their AutoML method each author chooses one or more datasets from OpenML (Vanschoren, Rijn, Bischl, & Torgo, 2014) or UCI (Bay, Kibler, Pazzani, & Smyth, 2000). At this moment there is not a single benchmark dataset set or collection of datasets to compare AutoML methods on (Olson, La Cava, Orzechowski, Urbanowicz, & Moore, 2017), nor is there a standard on the budget that needs to be set to do a benchmark test. However, Gijsbers et al. (2019) recently released an open-source benchmark framework for AutoML systems. They found that there is no significant difference in one-hour and four-hour budgets, but they do not propose a standard budget for AutoML benchmarks. Their work is used in Chapter 5.

4.3 An overview of existing AutoML methods

This section provides an overview of existing AutoML methods. For each method, we will describe the origin, its contribution to the existing knowledge base, the search strategy and used libraries. We only discuss distinctive methods that contribute to the body of knowledge of the AutoML community. Excluded methods are adaptions to existing systems without scientific additions like the Mondrian forest optimiser by Kim, Jeong, & Choi (2016), as it is a tweaked version of Auto-Sklearn. We excluded the Automated Statistician (Steinruecken, Smith, Janz, & Lloyd, 2018) as it is a large research project without any concrete implementation. Commercial applications like Google Cloud AutoML (Google, 2019) and Prophet (Taylor & Letham, 2018) are left out of scope because costs and privacy issues are barriers to adoption of analytics in healthcare (Dedding, 2018; Neff, 2013). In the following paragraphs, we discuss different AutoML methods.

Auto-WEKA

Auto-WEKA (Thornton et al., 2013) was the first available AutoML method (Quanming et al., 2018). It is built based on the Java-based WEKA library (Reutemann et al., 2009) and has two versions: 1.0 (Thornton et al., 2013) and 2.0 (Kotthoff, Thornton, Hoos, Hutter, & Leyton-Brown, 2017). The first version searches the search space by a tree-based BO method: SMAC (Bergstra et al., 2011; Thornton et al., 2013). Auto-WEKA consists of a learning algorithm and does not have any pre-processing features. It is only able to perform classification tasks (Thornton et al., 2013).

The second version of Auto-WEKA (Kotthoff et al., 2017) has made four significant improvements over the first version: First, it added regression tasks to the search space. Second, it can optimise all performance metrics supported by WEKA. Third, parallel runs on the same machine can be executed to improve performance. Finally, the new version provides complete integration with WEKA. Users need to provide a dataset and set a time budget constraint (Kotthoff et al., 2017).

Hyperopt-Sklearn

Hyperopt-Sklearn (Komer, Bergstra, & Eliasmith, 2014) was developed as a reaction to Auto-WEKA with the purpose to provide AutoML to the users of Python and the scikit-learn library (Hutter et al., 2019; Komer et al., 2014). Python is used instead of Java because Python applications are scalable (Komer et al., 2014). Hyperopt-Sklearn provides both pre-processing of data and classification. Hyperopt-Sklearn has fixed pipelines; they can contain one pre-processor and one classifier (Komer et al., 2014). This search space is searched using Hyperopt, which makes use of either random search or Tree Parzen Estimators (TPE) with BO (Bergstra, Yamins, & Cox, 2013). To improve efficiency in search, Hyperopt-Sklearn makes a distinction between conditional and non-conditional hyperparameters. Conditional parameters always need to be assigned, and non-conditional parameters depend on the chosen algorithm in the pipeline (Komer et al., 2014).

Auto-Sklearn

Auto-Sklearn (Feurer et al., 2015) is a Python-based AutoML built on the scikit-learn library. It extends the approach of Auto-WEKA to improve both efficiency and robustness of the AutoML process. Auto-Sklearn uses meta-learning on statistical properties of the dataset at hand as a means to warm-start the BO process. The search space is searched using SMAC (Feurer et al., 2015; Hutter et al., 2019). The first improvement of Auto-Sklearn over the previous methods is in improving efficiency by implementing a warm start module. The second improvement of Auto-Sklearn is on robustness, instead of discarding all the classification algorithms except for the best one, Auto-Sklearn saves the models that perform almost as good as the best method and ensembles these methods to improve performance (Feurer et al., 2015). Auto-Sklearn creates pipelines that consist of a data- and a feature pre-processor with a classifier of fixed length. The best pipelines are ensembled to improve predictions. Figure 18 depicts the architecture of Auto-Sklearn (Feurer et al., 2015).

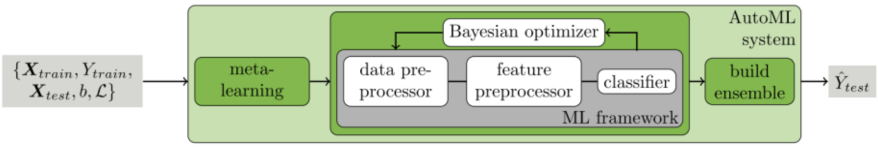


Figure 18: Auto-Sklearn model (Feurer et al., 2015)

Auto-Net

Auto-Net (Mendoza, Klein, Feurer, Springenberg, & Hutter, 2016) is the first AutoML method that configures a Neural Network (NN). Auto-Net has two versions: 1.0 (Mendoza et al., 2016) and 2.0 (Hutter et al., 2019, Chapter 7). Both Auto-WEKA and Auto-Sklearn inspired its architecture. To optimise the NN, SMAC is used (Mendoza et al., 2016). Auto-Net 1.0 integrated with Auto-Sklearn to make use of its architecture. Furthermore, it has added more classification algorithms and regression algorithms to the toolbox. Auto-Net makes use of feed-forward NN and is built on the Python library Lasagne (Mendoza et al., 2016). The depth of the NN is constrained to six layers to reduce the search space. Stochastic Gradient Descent is used to configure the internal weights of the nodes in the NN (Bottou, 2010). Gradual decay (Goodfellow, Bengio, & Courville, 2016) is applied to prevent local optimum bias (Mendoza et al., 2016).

Auto-Net 2.0 differs in three aspects from its predecessor. First, it uses PyTorch instead of Lasagne as a library because the support for Lasagne ended. PyTorch is selected as an alternative because it is one of the most popular ML libraries in Python. Second, Auto-Net 2.0 has expanded the search space compared to the first version by offering four network types: Multi-Layer Perceptrons, Residual NN, Shaped Multi-Layered Perceptrons and Shaped Residual Networks. Finally, it uses BO in combination with HyperBand (Li, Jamieson, DeSalvo, Rostamizadeh, & Talwalkar, 2018) instead of SMAC to improve the efficiency of finding well-performing NNs (Hutter et al., 2019). Auto-Net 2.0 does not use an ensemble for post-processing (Hutter et al., 2019).

TPOT

TPOT (Olson & Moore, 2016) is a Tree-based Pipeline Optimisation Tool which is a wrapper around the Python package scikit-learn. The incentive to develop TPOT was a reaction to the fixed-length pipeline methods discussed above. TPOT constructs pipelines of arbitrary length which can use multiple modified copies of a dataset as an input and consists of feature pre-processing- and selection methods and supervised learning classification methods (Olson & Moore, 2016). An example of such a pipeline is in Figure 19.

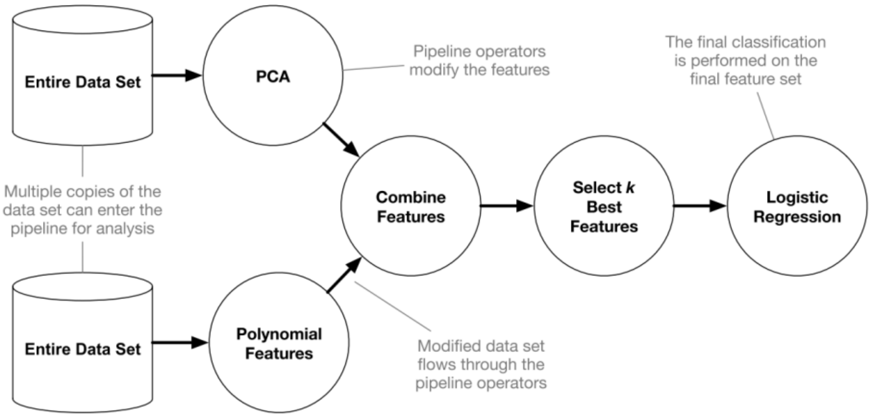

Entire Data Set
PCA
Pipeline operators modify the features
The final classification is performed on the final feature set
Multiple copies of the data set can enter the pipeline for analysis
Combine Features
Select *k* Best Features
Logistic Regression
Entire Data Set
Polynomial Features
Modified data set flows through the pipeline operators

Figure 19: TPOT example pipeline (Olson & Moore, 2016)

TPOT uses GP to construct pipelines; the building blocks of a pipeline are considered to be GP primitives to build a tree. These trees are an arbitrary representation of the ML pipeline, consisting of multiple datasets, pre-processors, and classification operators. Each node uses the output of its preceding node as input. TPOT divides a dataset into a training and a validation set. It gives each record an additional variable to mark this division. It adds variables for the true class and the pipeline's last guess of the value of the particular record because the set is not explicitly split (Olson & Moore, 2016).

To optimise and generate the pipelines, TPOT uses the Python package DEAP. The algorithm in the package generates 100 candidate pipelines and selects the top 20 pipelines that have the best balance in prediction accuracy and complexity. These top 20 pipelines are copied five times to function as input for the next generation. In the next generation, 5% of the new population crosses over with another copy doing a one-point crossover. The remaining 90% mutates at one point by having to lose a node, randomly insert a new node or mutate a node with each of the operations having a probability of 1/3rd to happen. A set amount of generations is generated and evaluated by TPOT before a pipeline is selected (Olson & Moore, 2016).

Layered TPOT

Layered TPOT (Gijsbers, Vanschoren, & Olson, 2017) is the successor of TPOT with a focus on improving the efficiency of the pipeline generation. It improves efficiency by implementing the idea of an Age Layered Population Structure (Hornby, 2006) in addition to the original TPOT algorithm. The individuals in the population are divided into ordered layers and trained and tested on different subsets of data. The first layer contains the smallest subset and all subsets increase in size at every layer. The individuals are trained and tested on different subsets of data. Thus, the algorithm cannot compare them. The next layer consists of the best pipelines from all subsets. The evaluation time of layered TPOT is dependent on the sample size. The dependence is built in to prevent a single pipeline from halting the algorithm, and to improve the algorithm performance. Evaluations are automatically stopped and marked as a failure if an evaluation exceeds the time limit. The stop is only executed to evaluate the best pipelines on the full set, as shown in Figure 20. The algorithm can discard pipelines that work well on the entire set, but not on subsets. This is considered to be a drawback of this method. (Gijsbers et al., 2017).

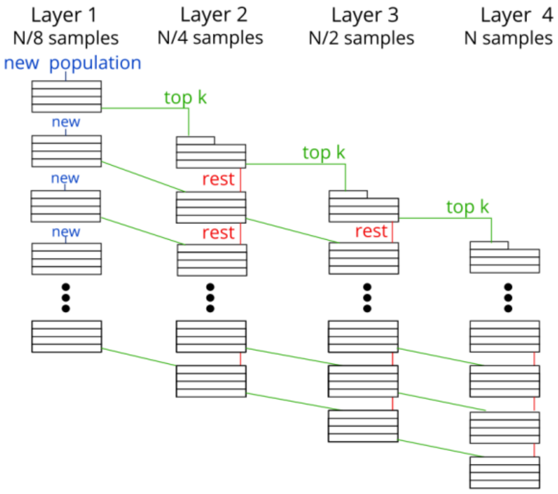


Figure 20: Layered TPOT (Gijsbers et al., 2017)

FLASH

Fast LineAr SearH (FLASH) (Zhang, Bahadori, Su, & Sun, 2016) proposes a two-layer Bayesian Network approach to improve search efficiency. FLASH is capable of predictive and descriptive modelling and makes use of the scikit-learn library (Zhang et al., 2016).

FLASH poses three main improvements over existing methods: First, the proposal of a two-layer hybrid model consisting of a parametric approach and a non-parametric approach. The first phase of the search is based on expected improvement in combination with BO to find the best *k* pipeline paths. The search then prunes the paths before it fine-tunes the paths. To fine-tune the paths, FLASH uses either SMAC or TPE. Second, the hyperparameter tuning algorithm is initialised using the optimal design strategy to improve the efficiency of FLASH overusing random search. Finally FLASH introduces a caching mechanism that can save time in the tuning process by reducing the number of redundant operations that are performed when calculating the optimal configuration of a method (Zhang et al., 2016).

RECIPE

REsilient ClassifIcation Pipeline Evolution (RECIPE) (Sá de et al., 2017) is an evolutionary method based on GP that makes use of grammar to improve its efficiency compared to other GP methods. RECIPE uses grammar to prevent the creation of invalid pipelines and focusses on classification tasks (Sá de et al., 2017).

RECIPE has three additions to previously used evolutionary methods. First, it uses grammar to describe the characteristics of an ML pipeline to avoid assembling and validating invalid pipelines. RECIPE uses the Scikit-Learn library to create pipelines. Within its grammar, it specifies the possible forms that a pipeline can have. It defines that a pipeline needs a dataset, classification algorithm, and evaluation as mandatory parts. Also, pre- and post-processing operators are possible components of a pipeline. Furthermore, the grammar in RECIPE is flexible. It can be extended beyond classification pipelines (Sá de et al., 2017). Second, it works with a larger search space than TPOT and Auto-Sklearn. Third, the global guided search based on the grammar makes it possible to evaluate the whole pipeline instead of parts of it like Auto-Sklearn does (Sá de et al., 2017).

AutoPrognosis

AutoPrognosis (Alaa & van der Schaar, 2018) is an AutoML method developed for clinicians. It follows a principled Bayesian approach in all components. AutoPrognosis uses Bayesian model averaging for pipeline construction. Meta-learning is used to find similar groups of patients. Both clinical and statistical features of datasets are used for Warm-starting. Figure 21 shows the architecture of AutoPrognosis(Alaa & van der Schaar, 2018).

AutoPrognosis uses Scikit-learn and is capable of missing data imputation, feature pre-processing, prediction, and calibration. For prediction, it can operate in three different modes: classification, temporal, and survival mode. Classification is used to predict binary clinical outcomes. The temporal mode is used to handle time-series data by using classification methods on a sliding window. Survival mode predicts the time to a clinical event in addition to survival models to predict a patient's clinical journey (Alaa & van der Schaar, 2018). As explanation is critical to clinical decision making based on computational input (Cabitza et al., 2017), the authors added a rule-based approximation to explain de decisions of AutoPrognosis to clinicians (Alaa & van der Schaar, 2018).

To enable BO based on Gaussian Processes, Alaa and van der Schaar (2018) generalised the CASH problem to the pipeline selection and configuration problem. By subdividing the search space into several sub-spaces with a maximum number of dimensions, the problem is generalised. With the reduced search space complexity, it is possible to use Gaussian Processes for optimisation as these are only feasible to use when having ten dimensions or less (Z. Wang, Zoghi, Hutter, Matheson, & De Feitas, 2013). Alaa and van der Schaar chose to use Gaussian Processes because they considered them the best performing BO method (Alaa & van der Schaar, 2018).

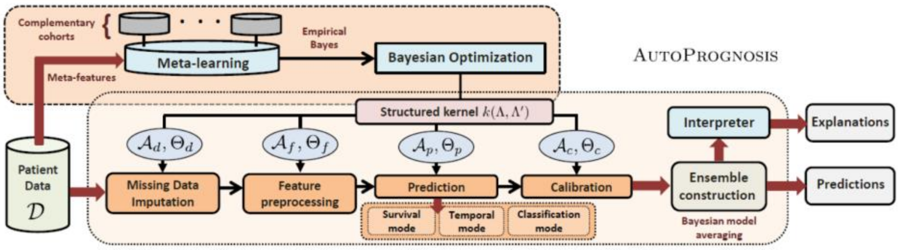


Figure 21: AutoPrognosis architecture (Alaa & van der Schaar, 2018)

Autostacker

Autostacker (Chen, Wu, Mo, Chattopadhyay, & Lipson, 2018) is an AutoML method that focusses on providing a set of potential useful pipelines for users without any pre-processing steps based on the scikit-learn and the XGBoost library. Pipelines are built using a stacking mechanism based on an EA algorithm and should generalise well to new data (Chen et al., 2018).

Autostacker has three properties that make it able to generalise well to new data. First, it uses cascading to handle small and sparse datasets. To prevent bias, it uses the original dataset to prevent bias from earlier operations on the data. Second, Autostacker uses combinations of different ML components to create flexible pipelines. Finally, EAs are used to search in the vast space of possibilities. Both stacking and cascading have not been used before in the discipline of AutoML (Chen et al., 2018).

Autostacker delivers a set of ten possible pipelines to the user to allow for flexibility and choice for the user. The argument for delivering multiple pipelines is that two pipelines can have different performances on an unseen dataset (Chen et al., 2018). Another advantage of Autostacker is that the system is scalable over multiple instances, as the worker nodes only have to share the validation results of different pipelines.

ML-Plan

ML-Plan (Mohr et al., 2018) uses a hybrid approach to construct pipelines. It combines the ideas and concepts from two different approaches: the idea of evaluating candidates at runtime (Thornton et al., 2013) and the idea of using hierarchical task networks (Nau et al., 2003) for pipeline planning (Mohr et al., 2018).

ML-Plan contributes to the existing knowledge base by proposing a two-phase search with HTN and a dedicated system to prevent overfitting. Mohr et al. (2018) claim to have invented the first AutoML technique that prevents overfitting. ML-Plan divides modelling into two phases, which should be considered as regions of the search space to prevent overfitting. The first phase collects a set of candidate pipelines based on the entire search space. The second phase takes these candidates and selects which ones minimise the generalisation error. Phase two operates on small subsets and discards high-variance models that work well on the complete validation set.

ML-plan creates new pipelines which consist of a pre-processor and a classifier. To evaluate all combination of classifiers and pre-processors the alterations to complete the pipeline are executed at random. (Mohr et al., 2018).

ML-plan chooses parameter values from a set of pre-defined possible values. Although this is a limitation of the technique, it is often sufficient for a good result. If the pre-defined range of the hyperparameter is not too far from the optimum, the algorithm performs well. ML-plan can run on both the WEKA and the scikit-learn libraries.

AlphaD3M

AlphaD3M (Drori et al., 2018) approaches the construction of a pipeline as a single-player game by having the player either insert, delete, or replace a part of the pipeline in each turn. AlphaD3M is based on AlphaZero (Silver et al., 2017), which is a generalisation of AlphaGo (Silver et al., 2017). AlphaGo is the AI program which famously defeated the world champion in the game Go. With this method, they create a pipeline which is explainable by including the ‘thinking’ behind the actions that lead to the construction of the pipeline. It leverages deep reinforcement learning to build the pipeline in this single-player game paradigm (Drori et al., 2018).

By learning these patterns in the game paradigm using self-play, the network learns to recognise patterns in the search space just like a human would. The learning method is built on a neural network following a Monte-Carlo tree search (Silver et al., 2017) based on PyTorch. By using this technique, AlphaD3M speeded up search compared to other methods ranging from a factor 3 to a factor 32 (Drori et al., 2018).

PoSH Auto-Sklearn

Portfolio Successive Halving (PoSH) Auto-Sklearn (Feurer et al., 2018) is the winner of the 2018 ChaCha AutoML challenge (ChaLearn, 2019). It is an extension of Auto-Sklearn method described above and builds on the scikit-learn library.

Successive halving was introduced in the search process to improve the efficiency of the method. Successive halving starts with a sample of the dataset and a small budget and continues with pipelines that perform well in this first round. It doubles the amount of data and computational budget in each round while halving the number of candidate pipelines. In addition to this, they build a portfolio based on OpenML datasets (Vanschoren et al., 2014), which contained meta-information to warm-start the search process. Finally, the ensemble technique of Auto-Sklearn was altered to exclude poor performing models. In PoSH Auto-Sklearn a model is not added to the ensemble if it performs over 3% worse than the best model. Figure 22 displays the architecture of PoSH Auto-Sklearn.

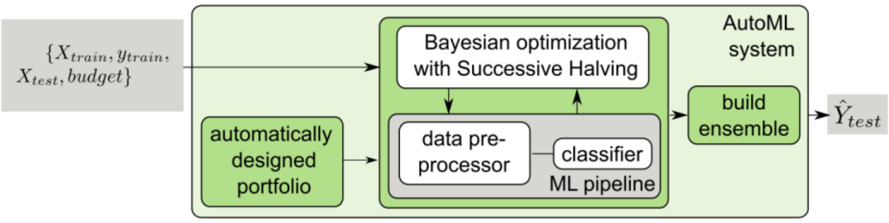

$\{X_{train}, y_{train}, X_{test}, budget\}$
AutoML system
Bayesian optimization with Successive Halving
data pre-processor
classifier
ML pipeline
automatically designed portfolio
build ensemble
$\hat{Y}_{test}$

Figure 22: PoSH Auto-Sklearn architecture (Feurer et al., 2018)

ATM

Auto-Tuned Models (Swearingen et al., 2017) are developed to support a multi-user machine learning platform in a cloud or cluster. The aim is to provide standardised abstractions in a library to become as influential for the AutoML community with a library as scikit-learn is to the machine learning community (Swearingen et al., 2017).

ATM contributes to the AutoML community by four additions. First, by implementing the first distributed AutoML system. The distributed system can handle multiple different AutoML requests at once from different users. Second, it has a database with previous requests to warm-start the search. Third, ATM has a new way to organise the search space by defining the search space into conditional parameter trees to speed up the search. Figure 23 shows how conditional parameter trees express the search space in a conditional tree with its hyperparameters below it as branches or leaves. The figure depicts the pruning of the search space of a support vector machine. Finally, ATM uses abstractions to enable the integration of different AutoML methods in the library (Swearingen et al., 2017).

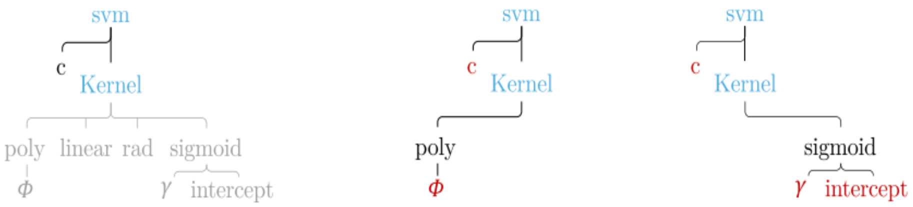


Figure 23: Conditional Parameter Tree (Swearingen et al., 2017)

ATM aims at three categories of users: Data Scientists, who can upload a dataset, select methods, and hyperparameter range to search over. AutoML experts, who can contribute to the library by expanding the presented framework. ML enthusiasts, who can contribute to the toolbox by adding new methods or implementations of the framework (Swearingen et al., 2017).

Auto-Keras

Auto-Keras (Jin et al., 2018) aims to provide an efficient way of finding NN architectures based on scikit-learn. The NNs are created using network morphism based on BO (Jin et al., 2018). Auto-Keras contributes to the AutoML literature in three ways. First, Auto-Keras presents a BO guided network morphism search for neural architectures, which is more efficient than previous methods for neural architecture search in AutoML. Second, the authors propose a NN kernel for BO: a tree-structured acquisition function optimiser with graph-level morphism. This kernel is used to enable BO to function outside of a Euclidian search space in which BO typically operates. By doing so, it becomes possible to search in a multi-dimensional space with BO. Third, Jin et al. (2018) provide an algorithm for optimising the acquisition function in this newly structured search space.

4.4 AutoML synthesis

In this section, we summarise and categorise the AutoML methods from the previous section. We only consider the newest versions of the methods. First, we compare the two AutoML methods developed for the healthcare domain. Second, we discuss methods with a fixed pipeline length. Third, we discuss the AutoML methods that build neural nets. Fourth, we discuss evolutionary methods. Fifth, we discuss distributed methods. Finally, we provide a detailed overview of the AutoML methods discussed in this chapter.

Healthcare

FLASH (Zhang et al., 2016) and AutoPrognosis (Alaa & van der Schaar, 2018) have both been developed for healthcare or with funds for healthcare, but with different incentives. FLASH was developed to improve the efficiency of creating and evaluating pipelines. AutoPrognosis is developed with the practitioner in mind. FLASH is a black-box tool, as most AutoML tools are. In contrast, AutoPrognosis is the only AutoML method that contains an explainer to justify its recommendations to a clinician.

We cannot compare the performances of both methods, as there has been no test featuring both. FLASH tested its performance on a medical dataset with the binary classification task of predicting drug non-responders. In this case, it outperformed other methods based on TPE and SMAC using error rate as the performance metric (Zhang et al., 2016). AutoPrognosis outperformed Auto-WEKA, Auto-Sklearn, and TPOT on multiple datasets in its own comparison (Alaa & van der Schaar, 2018).

Fixed pipelines

Auto-WEKA, Hyperopt-Sklearn, Auto-Sklearn, PoSH Auto-Sklearn, and ML-Plan are all methods that have a fixed pipeline length. PoSH Auto-Sklearn outperforms all other methods as it is the winner of the latest AutoML competition (Hutter et al., 2019, Chapter 10).

Auto-WEKA, Hyperopt-Sklearn, and Auto-Sklearn were the first three methods that were developed to tackle the CASH problem. What is interesting to see is that Auto-Sklearn has served as a basis for multiple other AutoML systems as depicted in Figure 24, whereas the other two methods have not. We assume that this is due to the warm-start procedure in Auto-Sklearn.

Neural Networks

The first version of AutoNet was the first to automate the configuration of a NN. It laid the groundwork for its successor and the inception of AlphaD3M and Auto-Keras. Besides laying the groundwork for these applications, it incentivised the inception of a lot of commercial applications. Most commercial applications that automatically tune NN are inspired on the first version of Auto-Net (Hutter et al., 2019). This is because Auto-Net was the first AutoML program to beat human experts in configuring a pipeline (Hutter et al., 2019; Mendoza et al., 2016).

AlphaD3M is the only AutoML method that makes use of reinforcement learning and is much faster than any other method in the field. In one case, it was 32 times faster than TPOT. However, AlphaD3M does not outperform other AutoML methods. In comparison with three other methods, its average rank is third, based on mean scores. As AlphaD3M ranks first on some datasets, it is still competitive (Drori et al., 2018). It is interesting to see if reinforcement learning gets widely adopted as a search strategy.

Evolutionary methods

Evolutionary methods can create pipelines of flexible length. These are TPOT, LTPOT, RECIPE, and Autostacker. They can do so due to their search strategy. The downside of evolutionary algorithms is that they can produce invalid pipelines and get stuck at local optima. RECIPE and LTPOT have independently overcome these downsides. It would be interesting to see when a hybrid version of these two strategies emerges.

Distributed methods

Two AutoML methods can process data in a distributed matter: Autostacker and ATM. It is remarkable, that there are only two systems that can run in a distributed manner when taking the computing cost of creating a pipeline in mind. Autostacker can use parallel processing as it proposes the best pipelines to its user. Hence it needs the performance scores of the pipelines. ATM is the only method which can run in parallel on different machines and is set up to be distributed and scalable. The development of ATM and integration with the methods described above is one of the most attractive developments in AutoML. An incentive to spur this development could be having a separate performance challenge for distributed AutoML methods.

Overview of methods.

Figure 24 demonstrates the relations between AutoML methods discussed in this chapter. It makes a distinction between methods that build NNs and methods that use traditional classifiers or regressors in their pipeline. Arrows between methods point out a relationship between methods. The colours of the methods indicate the search strategy that is applied in a method to create a pipeline. A tabular overview of the discussed AutoML methods is in Table 4, including, the prediction tasks, a link to the code repository and information about the pipeline creation.

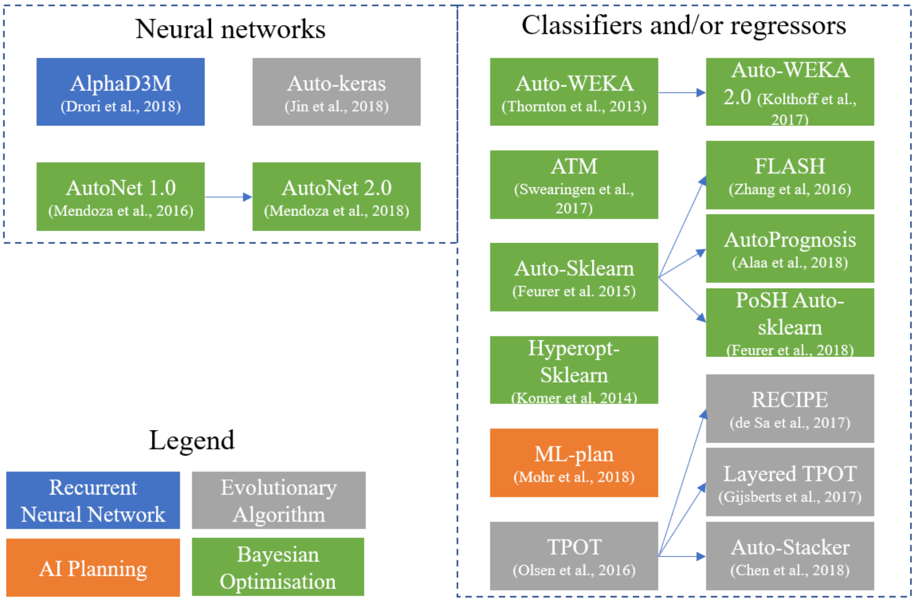

Neural networks
AlphaD3M (Drori et al., 2018)
Auto-keras (Jin et al., 2018)
AutoNet 1.0 (Mendoza et al., 2016)
AutoNet 2.0 (Mendoza et al., 2018)
Classifiers and/or regressors
Auto-WEKA (Thornton et al., 2013)
Auto-WEKA 2.0 (Kolthoff et al., 2017)
ATM (Swearingen et al., 2017)
FLASH (Zhang et al, 2016)
Auto-Sklearn (Feurer et al. 2015)
AutoPrognosis (Alaa et al., 2018)
PoSH Auto-sklearn (Feurer et al., 2018)
Hyperopt-Sklearn (Komer et al. 2014)
RECIPE (de Sa et al., 2017)
ML-plan (Mohr et al., 2018)
Layered TPOT (Gijsberts et al., 2017)
TPOT (Olsen et al., 2016)
Auto-Stacker (Chen et al., 2018)
Legend
Recurrent Neural Network
Evolutionary Algorithm
AI Planning
Bayesian Optimisation

Figure 24: Overview of AutoML methods

Table 4: Overview of existing AutoML methods[4]

Tool	Library/ package	Optimization	Pre-processor	Post-processor	Extra Feature(s)	Analysis capabilities	Code link
Auto-Weka 2.0	WEKA	Tree-based hierarchical BO	Yes	No		Binary classification Multi-label classification Regression	https://github.com/automl/autoweka
Auto-Sklearn	scikit-learn	Tree-based BO	Yes	Yes	Meta-learner	Binary classification Multi-label classification Regression	https://github.com/automl/auto-sklearn
Hyperopt-Sklearn	scikit-learn	Tree-based BO	Yes	No		Binary classification Multi-label classification	https://github.com/hyperopt/hyperopt-sklearn
TPOT	scikit-learn DEAP	Tree-based GP	Yes	No		Binary classification Multi-label classification Regression	https://github.com/EpistasisLab/tpot
Layered TPOT	scikit-learn	Tree-based GP	Yes	No		Binary classification Multi-label classification Regression	https://github.com/PG-TUe/tpot/tree/layered
Auto-Net 1.0	Lasagne	Feed-forward NN on Stochastic Gradient Descent	Yes	No		Binary classification Multi-label classification Regression	No implementation found
Auto-Net 2.0	PyTorch	BO and Hyperband (BOHB)	Yes	No		Binary classification Multi-label classification Regression	No implementation found
FLASH	scikit-learn	BO with expected improvement	Yes	No	Pipeline caching	Binary classification	https://github.com/yuyuz/FLASH

[4] Table continues on next page

Tool	**Library/ package**	**Optimization**	**Pre-processor**	**Post-processor**	**Extra Feature(s)**	**Analysis capabilities**	**Code link**
RECIPE	scikit-learn	Grammar-based GP	Yes	No		Binary classification	https://github.com/RecipeML/Recipe
AutoPrognosis	scikit-learn	BO and GP	Yes	Yes	Meta-learner, Explainer	Binary classification Survival analysis Temporal analysis	https://github.com/ahmedmalaa/AutoPrognosis[5]
ML-Plan	WEKA scikit-learn	HTN and EA	Yes	No		Binary classification Multi-label classification	https://github.com/fmohr/ML-Plan
Auto-stacker	scikit-learn XGBoost	Hierarchical stacking and EA	No	No		Binary classification Multi-label classification	No implementation found
Alpha3DM	PyTorch	NN and Monte Carlo Tree Search	Yes	Yes		Binary classification Multi-label classification Regression	No implementation found
PoSH AUTO-sklearn	scikit-learn	BO with successive halving	Yes	Yes	Meta-learner	Binary classification Multi-label classification Regression	No implementation found
Auto-Keras	scikit-learn	BO guided network morphism	Yes	No		Binary classification Multi-label classification	https://autokeras.com/
ATM	scikit-learn	Conditional Parameter Tree	Yes	No	Meta-learner	Binary classification Multi-label classification	https://github.com/HDI-Project/ATM

[5] At the time of writing no files were found in the folder, the author promised to upload his files in this folder soon.

5 Benchmark test

In this chapter, we describe the benchmark test for the different AutoML methods. We benchmark the performance of AutoML methods on medical datasets. First, we describe the datasets for benchmarking. Second, we discuss the set-up of the benchmark test. Finally, we discuss the results of the benchmark test.

5.1 Datasets

For the benchmark test, we use four datasets from the OpenML-CC18 library (Rijn van, 2019). OpenML-CC18 is the successor of the OpenML100 library, which was designed for delivering datasets that are suitable for benchmarking (Bischl et al., 2017). It has strict criteria for in- and exclusion of datasets to improve the reproducibility of benchmark tests (see Appendix 11.2). Furthermore, the OpenML100 library provides APIs for easy access to the datasets and is designed to improve the reproducibility of benchmark tests (Bischl et al., 2017). From the collection of OpenML-CC18 datasets, we have selected all medical datasets suited for binary classification problems; breast cancer, diabetes, Indian liver patients and sick. Non-numerical values in the datasets have been label encoded to prepare the sets for the benchmark test.

Breast cancer

The breast cancer dataset (Mangasariona & Wolberg, 1990) is from the University of Wisconsin Hospitals, created by Dr William Wolberg. The set represents the digitalisation of a fine needle aspirate of a breast mass to predict a prognosis (malignant or benign). The features describe the characteristics of the present cell nuclei in the image. The set consists of nine predictive features: 1) Clump thickness; 2) Cell size Uniformity; 3) Cell shape uniformity; 4) Marginal adhesion; 5) Single epi cell size; 6) Bare nuclei; 7) Bland chromatin; 8) Normal nucleoli, and 9) Mitoses. The dataset contains 699 data points and has no missing values. Appendix 11.3 contains an overview of the distribution of the variables. The class variable consists of 458 benign and 241 malignant cases.

Diabetes

The diabetes dataset (Dua & Graff, 2019) is from the National Institute of Diabetes and Digestive and Kidney Diseases. All subjects in the dataset are females of at least 21 years old of Pima Indian Heritage living near Phoenix, Arizona, USA. The goal of the dataset is to predict if a patient shows signs of diabetes according to the standards of the World Health Organisation. The features in this set describe characteristics of the women in the dataset. The dataset consists of eight features: 1) Number of times pregnant; 2) Plasma glucose concentration; 3) Diastolic blood pressure; 4) Triceps skinfold; 5) 2-hour serum insulin; 6) Body mass index; 7) Diabetes pedigree function, and 8) Age. The dataset contains 768 data points and has no missing values. Appendix 11.3

contains an overview of the distribution of the variables. The class variable contains 500 values for tested_negative and 268 values for tested_positive.

Indian liver patients

The Indian Liver Patient dataset (Dua & Graff, 2019) is from Venkata Ramana, Babu, & Venkateswarlu (2011). The dataset consists of data of both liver and non-liver patients from the northeast of Andhra Pradesh, India. The goal is to predict if a patient is a liver patient or not. The dataset contains ten features: 1) Age; 2) Gender; 3) Total bilirubin; 4) Direct bilirubin; 5) Alkphos alkaline phosphatase; 6) Sgpt Alanine Aminotransferase; 7) Sgpt Aspartate Aminotransferase; 8) Total proteins; 9) Albumin; and 10) A/G ratio albumin and globulin Ratio. The dataset consists of 583 instances and has no missing values. Appendix 11.3 contains an overview of the distribution of the variables. The class variable is distributed with 416 liver patients and 167 non-liver patients.

Sick

The sick dataset (Quinlan, 1986) is from the Garavan Institute and Ross Quinlan from the New South Wales Institute from Sydney, Australia. The goal is to predict if a patient has Thyroid disease. The dataset contains twenty-nine features: 1) Age; 2) Sex; 3) On thyroxine; 4) Query on thyroxine; 5) On antithyroid medication; 6) Sick; 7) Pregnant; 8) Thyroid surgery; 9) l131 treatment; 10) Query hypothyroid; 11) Query hypothyroid; 12) Lithium; 13) Goitre; 14) Tumor; 15) Hypopipuitary; 16) Psych; 17) TSH measured; 18) TSH Real 0%; 19) T3 measured; 20) T3 real 0%; 21) TT4 measured; 22) TT4 Real 0%; 23) T4U Measured; 24) T4U real 0%; 25) FTI measured; 26) FTI real 0%; 27) TBG measured; 28) TBG real; and 29) Referral source. The dataset contains 3772 instances and has 6064 missing values. Appendix 11.3 contains an overview of the distribution of the variables. The class variable contains 3541 negative and 231 sick cases.

5.2 Set-up

As described in the methods section, we use an open-source AutoML benchmark suite for the benchmark test (Gijsbers et al., 2019). With different datasets, we compare four AutoML methods. The benchmark suite is created to compare AutoML methods on various datasets. As the suite is open-source, authors can add their frameworks. Users of the benchmark set can use their dataset to run the benchmark tests. The methods that are available within the benchmark at the time of writing are TPOT, Auto-Weka, Auto-Sklearn, H2O, Hyperopt-Sklearn, and Oboe. As a baseline, four methods have been included in the open-source framework: a constant predictor, a decision tree, a random forest, and a tuned random forest.

We ran two tests, the first test included all methods and had a budget of one hour. We used a budget of one hour as longer runs *'bring only slight score improvements'* (Gijsbers et al., 2019). To verify this statement, we ran the second test with a four-hour budget for one method based on BO, one method based on EA and the worst performer in the one-hour test. In accordance with Gijsbers et al., we use AUROC as a performance metric.

One-hour budget

In our set-up, we exclude H2O and Oboe from the benchmark test. We exclude H2O because there is no paper published about it. Oboe is excluded as it is still in the early stages of development (Gijsbers et al., 2019). We include the TPOT, Auto-WEKA, Auto-Sklearn and Hyperopt-Sklearn and compare their performance to a constant predictor and a decision tree as a baseline. If a method did not show a result in time, we have considered this as a missing value.

Four-hour budget

For the four-hour test, we included Auto-Sklearn, TPOT and Hyperopt-Sklearn. Auto-Sklearn and TPOT are included to find out what the result is of a three-hour budget increase for a BO and an EA method. We included Hyperopt-Sklearn to check if the time budget limited the method in the first benchmark test. If a method did not show a result in time, we have considered this as a missing value.

5.3 Results

In this section, we first discuss the results of the one-hour benchmark test. After that, we discuss the results of the four-hour benchmark test and compare the performance of the individual methods with different time budgets.

One-hour budget

We ran one benchmark using a time budget of one hour with a total of 160 hours of computational budget time. Figure 25 contains the visualisation of results; the x-axis contains the different datasets, the y-axis shows the AUROC. A coloured dot marks the score for an AutoML method on each of the ten folds. The minimum, maximum and median score of each AutoML method are available in Table 12.

A Kruskal-Wallis H test indicated that there was a statistically significant difference in the distribution for the Breast (H =11.36, p <.001), Diabetes (H =18.64, p <.001), Liver (H =17.93, p <.001) and Sick dataset (H =27.87, p <.001) between the AutoML methods, see Table 5 for statistics.

What is interesting to see in Figure 25 is that on the liver dataset the decision tree and Hyperopt-Sklearn do not always outperform the constant predictor. On the diabetes dataset, Hyperopt-Sklearn lags behind the three other methods, but performs better than the constant predictor and has a similar performance to the decision tree. On the breast dataset, all AutoML methods have the maximum score in at least one fold. All methods perform well on the breast set, given their median scores and distribution. The performance of the decision tree indicates that it is not a hard prediction problem. For the results on the sick dataset, we see that TPOT and Auto-Sklearn outperform the other two methods in both consistency and score of their predictions despite the fact that the set has missing values. Hyperopt-Sklearn is again not better than the decision tree.

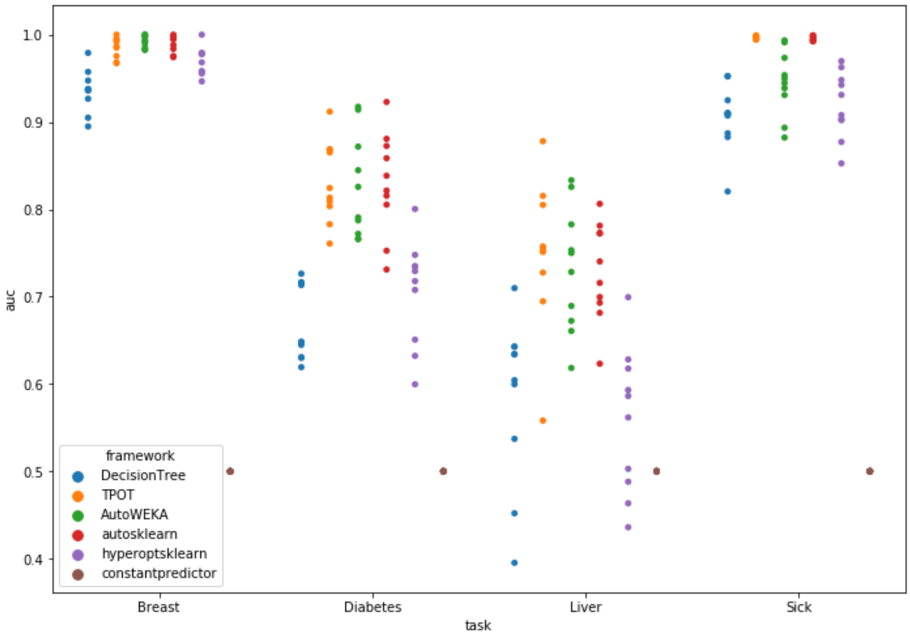


Figure 25: One-hour benchmark test results

Overall, TPOT registered the highest median score after running for one-hour on all sets but the Breast dataset, in which Auto-Sklearn registered the best performance. A Mann-Whitney U test indicated that Auto-Sklearn significantly outperforms the decision tree (*U* =2.0, *p* <.001) and Hyperopt-Sklearn (*U* =17.5, *p* <.01) on the Breast set. TPOT significantly outperforms the decision tree on the datasets Diabetes (U =0.0, *p* <.001), Liver (*U* =8.0, *p* <.001) and Sick (*U* =0.0, *p* <.001). Furthermore, TPOT outperforms Hyperopt-Sklearn significantly on the datasets Diabetes (U = 2.0, *p* <.001), Liver (*U* =7.0, *p* <.001) and Sick (*U* =0.0, *p* <.001). Finally, Auto-WEKA is significantly outperformed on the Sick dataset by TPOT (*U* =0.0, *p* <.001). This is probably because Auto-WEKA does not impute data for missing values. Auto-Sklearn and TPOT impute values for missing data and both have a more condensed distribution in their results. The statistics for the Mann-Whitney U test are available in
Table 6. The table shows the *p*-values and U statistics for each method compared to the best performing method on each dataset. TPOT and Auto-Sklearn do not significantly differ in performance for any of the datasets.

Table 5: Statistics for Kruskal Wallis test

	Breast	**Diabetes**	**Liver**	**Sick**
***H*-statistic**	11.36	18.64	17.93	27.87
***p*-value**	$.995^{**}$	$.324^{**}$	$.455^{**}$	$.386^{**}$

** *p* < 0.001

Table 6: P-values and U statistic for Mann-Whitney U test compared to the best performer

	Breast		**Diabetes**		**Liver**		**Sick**	
	p	***U***	***p***	***U***	***p***	***U***	***p***	***U***
Decision Tree	$\mathbf{.164^{**}}$	**2.0**	$\mathbf{.908^{**}}$	**0.0**	$\mathbf{.895^{**}}$	**8.0**	$\mathbf{.913^{**}}$	**0.0**
TPOT	.236	40.0						
Auto-WEKA	.395	46.0	.455	48.0	.263	40.0	$\mathbf{.913^{**}}$	**0.0**
Auto-Sklearn			.425	47.0	.213	39.0	.5	49.5
Hyperopt-Sklearn	$\mathbf{.8^{*}}$	**17.5**	$\mathbf{.164^{**}}$	**2.0**	$\mathbf{.657^{**}}$	**7.0**	$\mathbf{.913^{**}}$	**0.0**

* *p* < 0.01 ** *p* < 0.001

Four-hour budget

We ran the second benchmark using a time budget of four hours with a total of 480 hours of computational budget time. The four-hour budget results in Figure 26 show a similar pattern to Figure 25. A Kruskal-Wallis test indicated that there was a statistically significant difference in the distribution for the Breast (*H* =13.10, *p* <.001), Diabetes (*H* =31.10, *p* <.001), Liver (*H* =31.51, *p* <.001) and Sick dataset (*H* =28.93, *p* <.001) between the AutoML methods, see Table 7 for statistics.

A Mann-Whitney U test indicates that the best performer again only significantly outperforms Hyperopt-Sklearn on the datasets Breast, (*U* =1.0, *p* <.01), Diabetes (*U* =2.0, *p* <.001), Liver (*U* =5.0, *p* <.001) and Sick (*U* = 0.0, *p* <.001). The performances of TPOT and Auto-Sklearn do not significantly differ from each other on the datasets. The statistics for the Mann-Whitney U test are available in Table 8. Although the difference in performance is not significant, TPOT again registered the highest score median on three of the four datasets. Auto-Sklearn only records a better median score on the breast dataset. In the next sections, we compare the performance of each method on the one-hour and the four-hour runs. The results of the four-hour runs are available in Table 13.

Table 7: Statistics for Kruskal Wallis test

	Breast	**Diabetes**	**Liver**	**Sick**
***H*-statistic**	13.10	31.10	31.51	28.93
***p*-value**	$.442^{**}$	$.809^{**}$	$.665^{**}$	$.232^{**}$

** *p* < 0.001

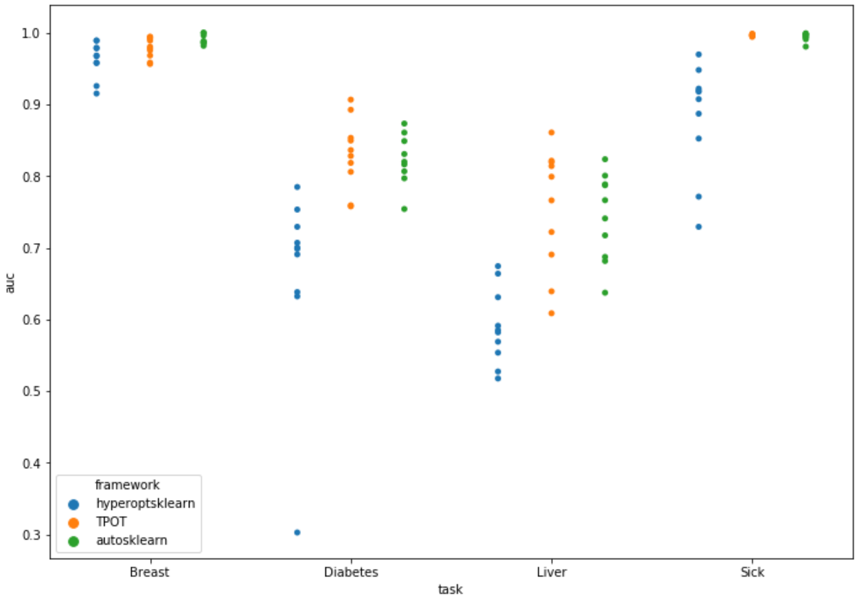


Figure 26: Four-hour benchmark test results

Table 8: P-values and U-statistic for Mann Whitney U test compared to the best performer

	Breast		**Diabetes**		**Liver**		**Sick**	
	p	***U***	***p***	***U***	***p***	***U***	***p***	***U***
TPOT	.485	49.0						
Auto-Sklearn			0.485	49.0	.285	42.0	.455	48.0
Hyperopt-Sklearn	**.123****	**1.0**	**.164****	**2.0**	**.384****	**5.0**	**.913****	**0.0**

** $p < 0.001$

Auto-Sklearn

When we compare the performance of Auto-Sklearn on the four datasets for the different time budgets using a Mann-Whitney U test, we can see that the results are from the same distribution. Figure 27 illustrates the difference in performance between the one-hour and the four-hour budget runs. For two datasets, the recorded median score decreases with a four-hour budget. The score decrease might be the result of overfitting. On the other two datasets, the recorded median scores improve for the four-hour budget. However, these differences are not significant for all datasets, both sets of scores are from the same distributions: Breast (U =39.5 , p >=.05), Diabetes (U =49.0 , p >=.05), Liver (U =41.5 , p >=.05), Sick (U =44.0 , p >=.05). The median scores are available in Table 9 along with the statistics of the Mann-Whitney U test.

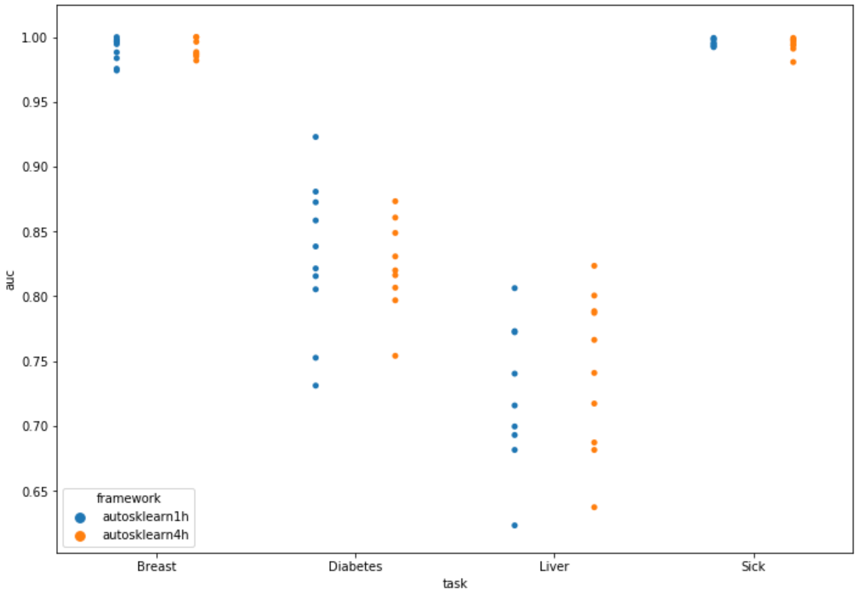


Figure 27: One-hour vs four-hour performance comparison for Auto-Sklearn

Table 9: Auto-Sklearn results comparison

Dataset	1-hour median	4-hour median	*p*-value	*U*-statistic
Breast	**0.995**	0.988	.224	39.5
Diabetes	**0.830**	0.82	.485	49.0
Liver	0.728	**0.754**	.273	41.5
Sick	0.995	**0.996**	.339	44.0

TPOT

When we compare the performance of TPOT on the four datasets for the different time budgets, we can see from Figure 28 that there is no difference in performance between the 1-hour and the 4-hour budget runs. The median scores improve for all sets but the breast dataset, but all scores are from the same distribution. However, these differences are not significant for all datasets, both sets of scores are from the same distributions: Breast (*U* =34.0, *p* >=.05), Diabetes (*U* =50.0 , *p* >=.05), Liver (*U* =46.0 , *p* >=.05), Sick (*U* =49.0 , *p* >=.05).

The median scores are available in Table 10 along with the statistics of the Mann-Whitney U test.

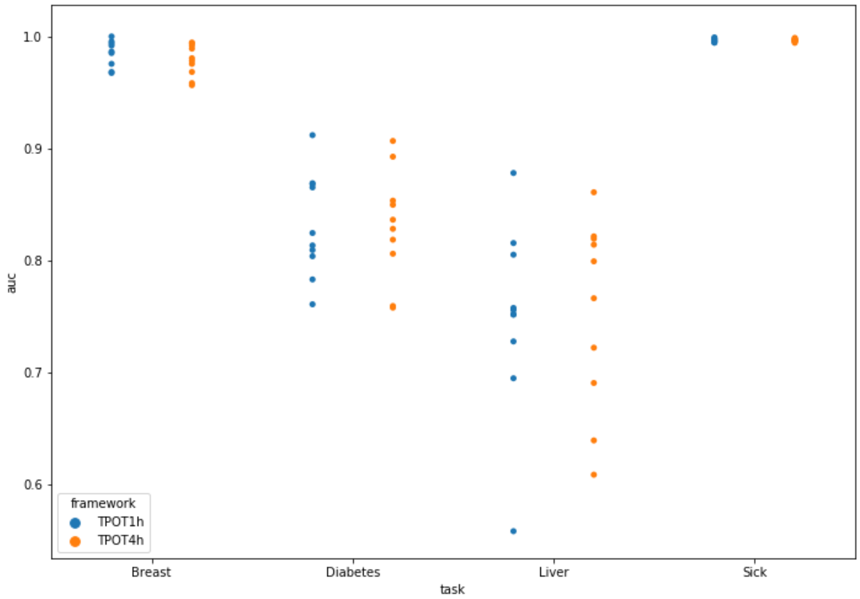


Figure 28: One-hour vs four-hour performance comparison for TPOT

Table 10: TPOT results comparison

Dataset	1-hour median	4-hour median	*p*-value	*U*-statistic
Breast	**0.986**	0.979	.120	34.0
Diabetes	0.819	**0.832**	.485	50.0
Liver	0.754	**0.783**	.396	46.0
Sick	0.996	**0.997**	.485	49.0

Hyperopt-Sklearn

When we compare the performance of Hyperopt-Sklearn on the four datasets for the different time budgets, we can see from Figure 29 that there is no significant difference in performance between the one-hour and the four-hour budget runs. The median scores decrease for all sets but the liver dataset, but not significantly. The median scores are available in Table 11 along with the statistics of the Mann-Whitney U test. Hyperopt-Sklearn has a maximum of 1000 evaluations as a default parameter. We did not alter this, as recommended by Balaji and Allen (2018), to improve the fairness of evaluation and reproducibility for the benchmark test. This parameter caused an early stop for all datasets but the sick dataset. However, the early stop does not seem to have influenced the results. The differences for the runs are not significant for all datasets, both sets of scores are from the same distributions: Breast (*U* =48.5 , *p* >=.05), Diabetes (*U* =39.0 , *p* >=.05), Liver (*U* =40.0, *p* >=.05), Sick (*U* =38.0 , *p* >=.05). if we look at the scores for the sick dataset.

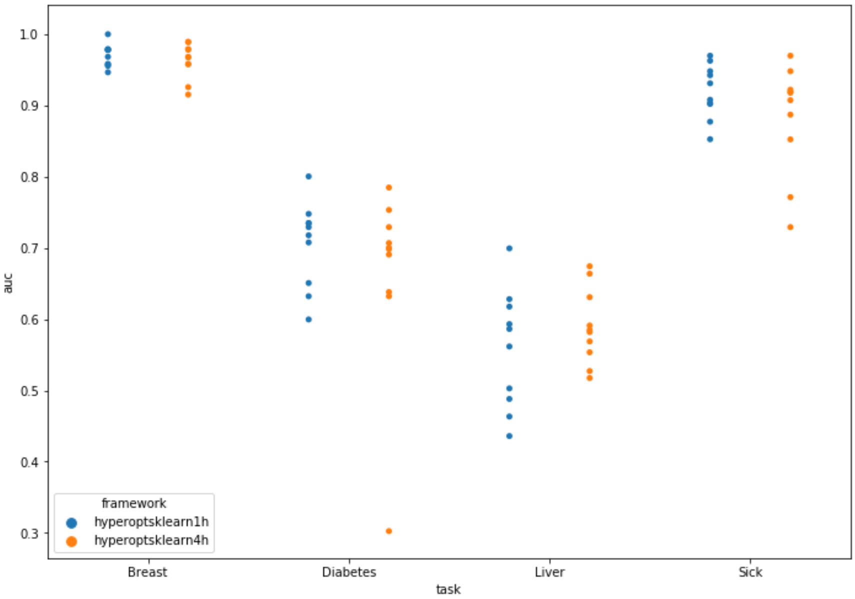


Figure 29: One-hour vs four-hour performance comparison Hyperopt-Sklearn

Table 11: Hyperopt-Sklearn results comparison

Dataset	1-hour median	4-hour median	*p*-value	*U*-statistic
Breast	**0.973**	0.968	.470	48.5
Diabetes	**0.724**	0.699	.213	39.0
Liver	0.574	**0.583**	.236	40.0
Sick	**0.920**	0.913	.192	38.0

5.4 Conclusion

From the conducted benchmark tests, we can conclude that no method consistently outperforms all others. However, we see that TPOT records the highest median scores on three of the four tasks in this test for both time budgets. However, this performance is not significantly better than the performance of Auto-Sklearn and Auto-WEKA. Auto-Sklearn gets similar results to TPOT given the one-hour and four-hour tests, Auto-WEKA gets similar results but is only outperformed on the Sick dataset. Finally, Hyperopt-Sklearn performs the worst on all tasks and even predicts scores that are worse than the constant predictor.

Furthermore, we can confirm the findings of Gijsbers et al. (2019) that the increase of time budget from one to four hours does not result in a significant score improvement using a Mann-Whitney U test, all results come from the same distributions.

Based on the results of the one- and four-hour benchmark tests, we conclude that no method consistently outperforms the all methods.

Table 12: One-hour benchmark test results

Framework **Dataset**	**TPOT**			**Auto-WEKA**			**Auto-Sklearn**			**Hyperopt-Sklearn**		
	Min	Med	Max	Min	Med	Max	Min	Med	Max	Min	Med	Max
Breast	0.967	0.986	1.0	0.983	0.993	1.0	0.974	**0.995**	1.0	0.947	0.973	1.0
Diabetes	0.761	**0.819**	0.911	0.766	0.808	0.917	0.731	0.830	0.923	0.600	0.725	0.800
Sick	0.994	**0.996**	0.999	0.882	0.947	0.993	0.992	0.995	0.999	0.853	0.920	0.970
Liver	0.558	**0.754**	0.878	0.618	0.739	0.833	0.623	0.728	0.806	0.436	0.574	0.699

Table 13: Four-hour benchmark test results

Framework **Dataset**	**TPOT**			**Auto-Sklearn**			**Hyperopt-Sklearn**		
	Min	Med	Max	Min	Med	Max	Min	Med	Max
Breast	0.957	0.979	0.995	0.982	**0.988**	1.0	0.915	0.968	0.989
Diabetes	0.758	**0.832**	0.907	0.754	0.82	0.873	0.303	0.699	0.785
Sick	0.994	**0.997**	0.999	0.981	0.996	0.999	0.729	0.913	0.970
Liver	0.609	**0.783**	0.861	0.637	0.754	0.824	0.518	0.583	0.674

6 Requirements for AutoML methods

This chapter describes the requirements for AutoML methods as we develop applications to enable researcher-physicians to start with self-service data science. First, we describe and categorise the requirements for AutoML methods. Second, we analyse if AutoML methods support the requirements. Finally, we conclude which AutoML method would be the best to implement into the artefact based on the requirements.

6.1 User-stories

We present the requirements in this paragraph in the form of user-stories. For user-stories, we use the template of Cohen (2004): *“As a 〈type of user〉, I want 〈goal〉, [so that 〈some reason〉].”*. An example of such a user-story could be: As a researcher-physician, I want a graphical user interface, so that I do not have to learn a programming language to make use of AutoML. User-stories are useful to describe the functional requirements of users. Furthermore, it is an excellent way to summarise the needs of a user in a comprehensive and atomic way (Cohen, 2004).

Besides the requirements elicitation from our interviews, we have analysed five interviews from previous research on the application of self-service data science with ML in healthcare (Vries de, 2018). From both sets of interviews, we have elicited requirements and put them into user-stories, sorted on their frequencies in Table 14. Transcripts of our interviews are available in Appendix 11.5.

Table 14: User-stories

No.	User-story	Frequency
1	As a researcher-physician, I want to know how a prediction mechanism works, so that I can trust it more easily.	5
2	As a researcher-physician, I want to be able to perform ML without having to code, so that I do not have to spend time learning how to program.	5
3	As a researcher-physician, I want to use logistic regression, so that I can follow the medical guidelines for research.	4
4	As a researcher-physician, I want to see which variables are included and excluded in the model, so that I can assess variable importance.	4
5	As a researcher-physician, I want to see the difference between models with different variables included so that I can assess variable importance.	4
6	As a researcher-physician, I want to transfer my model into a calculation tool, so that it can be used in clinical practice.	4

7	As a researcher-physician, I want to have results within a day, so that I do not have to wait.	4
8	As a researcher-physician, I want to know the statistical power of a created model, so that I know if I can use it.	4
9	As a researcher-physician, I want that the AutoML method explains its decisions, so that I can check its reasoning.	3
10	As a researcher-physician, I want to have a graphical user interface, so that the chance of making errors is less than while coding.	2
11	As a researcher-physician, I want to see the importance of each variable. So that I can check the reasoning of the computer.	2
12	As a researcher-physician, I want to use code, so that I can trace back the decisions that I have made	2
13	As a researcher-physician, I want to know what happens with missing data, so that I can evaluate the model correctly.	2
14	As a researcher-physician, I want to get suggestions for variables to include by the computer, so that I can improve my models.	2
15	As a researcher-physician, I want to see the amount of variance that is explained by my model, so that I can assess the model quality.	2
16	As a researcher-physician, I want to see multiple alternatives for a model, so that I am more in control of the machine.	1
17	As a researcher-physician, I want to see an overview of the data, so that I know what I am using for my analysis.	1
18	As a researcher-physician, I want to be able to micro-target so that I can get better results for my patients.	1
19	As a researcher-physician, I want to be able to transfer files directly to a machine learning tool, so that I have a fast process.	1
20	As a researcher-physician, I want to be able to include a patient in the decision, so that I can explain a decision as part of shared decision making.	1
21	As a researcher-physician, I want to see frequencies and graphs so that I can get into the data real fast.	1

User-stories categorised

We grouped the user-stories from Table 14 into four categories: 1) User interaction; 2) Model construction; 3) Model explanation, and 4) Model usage. We only include user stories that are requested by two or more interviewees. Hence, we excluded user-story sixteen to twenty-one. We categorise and discuss all other user-stories from Table 14 below.

User interaction

User-stories ten and twelve consider user interaction with an AutoML method. User-story ten shows that some users prefer a Graphical User Interface (GUI) over coding. **quotes removed for confidentiality**

Model construction

User-stories two, three, seven, thirteen and fourteen are about model construction. User story two is about not having to code to construct an ML model. **quotes removed for confidentiality**

Regarding user-story three, four interviewees mention that they only want to use AutoML to create logistic regression models, as that is the standard in medical practice. **quotes removed for confidentiality**

User-story seven is about the time of the model construction; most interviewees think that one day is a reasonable time for model construction. **quotes removed for confidentiality**

User-story thirteen is about how the model construction handles missing data when building the model. **quotes removed for confidentiality**

User-story fourteen is about the AutoML method suggesting which variables to include or exclude while creating the model, even if the variables are not present in the uploaded dataset. **quotes removed for confidentiality**

Model explanation

User-stories one, four, five, eleven, twelve and fifteen, are all about model explanation. User story one is about knowing how a prediction mechanism works. The AutoML method should explain the selected model. **quotes removed for confidentiality**

Preferably the AutoML method should use logistic regression for model construction, as mentioned in the paragraph above.

User-story four, nine and eleven are about the importance of variables that are included in the model, as variable importance is crucial for the interviewees. The importance of variables explains why they have been in- or excluded in a model: "*You are not allowed to use all data that you have available as predictors in such a model*" **quotes removed for confidentiality**

User-story five is about seeing the difference in the performance of two models where some variables are in- or excluded. **quotes removed for confidentiality**

User-story twelve and fifteen are about knowing the statistical power and explained variance of the constructed model; this is used by the interviewees to explain how useful their models are. **quotes removed for confidentiality**

Model usage

User story six is about using the model in practice. All practitioners have discussed the necessity of getting to use their models in practice: **quotes removed for confidentiality**

6.2 User-story analysis

In this section, we analyse the user-stories from the previous section. For each of the categories from the previous section, we compare the functional requirements of the user stories to the capabilities of the AutoML methods.

User interaction

In the category user interaction, two user-stories are in conflict. User-story ten conflicts with user story twelve. Some researcher-physicians prefer to use code to do their analysis, so they have more control over what is happening. In contrast, other researcher-physicians prefer to have a GUI, as coding errors cost a lot of time to solve.

Model construction

The five user-stories for model construction are two, three, seven, thirteen and fourteen. User-story two, automatic model construction, is covered by the concept of AutoML. User-story three is about only creating logistic regression models. Auto-WEKA and TPOT are the only methods that support logistic regression. User-story seven is about delivering results within a day. The time constraint is possible for all methods.

User-story thirteen is about the processing of missing data. Auto-Sklearn and TPOT impute missing data with the median. Auto-WEKA and Hyperopt-Sklearn do not provide any documentation on their strategy for handling missing data. The Hyperopt-Sklearn documentation does state that it does not do any pre-processing by default. AutoML is unable to suggest variables as requested in user story fourteen. Table 15 is a matrix containing the AutoML methods, user stories and their matches.

Table 15: AutoML methods and model construction user-stories

User-story	Auto-Sklearn	Auto-WEKA	TPOT	Hyperopt-Sklearn
Automatic model configuration (2)	X	X	X	X
Only logistic regression (3)		X	X	
Results within a day (7)	X	X	X	X
Explain missing data handling (13)	X		X	
Suggest variables (14)				
Total matches	**3**	**3**	**4**	**2**

Model explanation

None of the AutoML methods can support any user-stories about model explanation. The AutoML methods do not explain the selected prediction mechanisms, nor do they explain the variables that are in- or excluded in a model or their importance. The methods also do not explain statistical power and explained variance. However, requirement one and five are possible to integrate into an artefact. An empty matrix demonstrates the mismatch between the capabilities of the AutoML methods and the requirements from user-stories regarding model explanation. Table 16 demonstrates the empty matrix with user-stories and AutoML methods.

Table 16: AutoML methods and model explanation user-stories

User-story/method	Auto-Sklearn	Auto-WEKA	TPOT	Hyperopt-Sklearn
Prediction mechanism explanation (1)				
Variable importance (4, 11)				
Model comparison (5)				
Statistical power (8)				
Explain decisions made (9)				
Explained variance (15)				
Total matches	**0**	**0**	**0**	**0**

Model usage

User-story six is about using the created model in practice. All four methods support exporting the model so that it can be applied to unseen data or to predict a single case.

6.3 Conclusion

After comparing the four AutoML methods to the user stories, we can conclude that TPOT is the best AutoML method for this set of requirements. TPOT satisfies five of the fifteen assessed requirements compared to four out of fifteen by Auto-WEKA and Auto-Sklearn. What is interesting to note is the inability of all AutoML methods to explain the created models. The need for explainability is evident: Model explanation is the biggest category in the user-story categorisation and described in the literature as an important factor (Vollmer et al., 2018). Besides that, explaining model decisions is obligatory in Europe since the introduction of the General Data Protection Regulation Law (Janssen, 2019, pp. 40–42). Table 17 contains an overview of the number of requirements satisfied by the AutoML methods in each category. We have not included the user-interaction category as it contains conflicting user-stories and does not apply to AutoML methods, only to the artefacts.

Table 17: AutoML method scores on user-story categories

Category/method	Auto-Sklearn	Auto-WEKA	TPOT	Hyperopt-Sklearn
User interaction	n/a	n/a	n/a	n/a
Model construction	3	3	4	2
Model explanation	0	0	0	0
Model usage	1	1	1	1
Total matches	**4**	**4**	**5**	**3**

7 Results

This chapter describes the created artefacts and the results of the artefact evaluation. First, we describe the designed artefacts. Second, we describe the evaluation strategy. Third, we describe the results of the artefact evaluation.

7.1 Artefact design

The designed artefacts are created to automate a part of the data preparation phase and to automate the complete modelling phase of CRISP-DM. The data preparation activities involve the possibility to in- or exclude variables, data imputation for missing values and the recoding of categorical variables to numerical variables, as TPOT cannot handle non-numerical input data. Due to the conflict in user-interaction, we will design two artefacts with the same functionality but a different interface.

Both artefacts are designed based on the requirements from the previous chapter. The artefacts can be used to create logistic regression models within the timespan of a day and users who do not know how to code should be able to use the artefacts. The artefacts contain a description of missing data handling, as well as the possibility to compare the different models. The two artefacts are a web-application and a notebook. Table 18 contains an overview of the user-stories that are in- and excluded from the artefacts. We have decided not to include user-story six, as the focus of the artefacts is on the data preparation phase and modelling phase of CRISP-DM, not on the deployment phase. All excluded user-stories were impossible to integrate into the artefacts.

Artefact A: Flask web-application

Based on user-story ten, we have developed a web application in Python based on the Flask Framework. This web application allows users to upload a dataset, create subsets of these datasets and create a pipeline using AutoML. Within the application, a user can access overviews of the uploaded datasets, created subsets and models. The Flasky application by Grinberg (2014) is the basis for the architecture of the application. Heroku[6] is used to deploy the application. To construct the AutoML methods, we used a Redis[7] background server to enable the user to use the application during model construction. The code of the application is available on git, a link to the git and screenshots of the artefact are available in Appendix 6: Artefacts for researcher-physicians.

Artefact B: Jupiter notebook

For the researcher-physicians who indicate that they prefer to use code over a GUI (user-story twelve), we have prepared a Jupiter Notebook.[8] A notebook is a document that contains both computer code and rich-text items. We provide a notebook, displaying and explaining every line of code. Access to the code gives the users control over

[6] https://www.heroku.com

[7] https://redis.io/

[8] https://jupyter-notebook.readthedocs.io/en/stable/

their knowledge discovery process. They can edit every part of the code to find out how it influences the eventual outcome. A link to the git and screenshots of the artefact are available in Appendix 6: Artefacts for researcher-physicians.

Table 18: User-stories per artefact

	Artefact A	**Artefact B**
AutoML method	TPOT	TPOT
User interaction	GUI (10)	Code-based interface (12)
Model creation	1, 2, 3, 7	1, 2, 3, 7
Model explanation	5, 13	5, 13
Model usage	n/a	n/a
Excluded	4, 6, 8, 9, 11, 14 & 15	4, 6, 8, 9, 11, 14 & 15

7.2 Artefact evaluation strategy

All interviewees will evaluate both artefacts. The artefacts will be evaluated using the risk and efficacy strategy from the framework for evaluation in design science (Venable et al., 2016). In line with the framework, we have created a set of refined hypotheses based on the user-story categories. The category model usage was excluded for evaluation, as it is not part of the scope of this research. The interviewees have to upload a dataset, create two or more subsets and compare the results as part of the evaluation. We randomized the order of the first demonstrated artefact to avoid learning bias.

User interaction

To test user interaction, we have created four refined hypotheses to test which artefact is preferred over the other for user interaction. Two hypotheses are about the ease of use of uploading a dataset and creating a subset. The other two hypotheses are about the explanation of the steps within the artefact and how easy these are to find. The hypotheses are:

1. Artefact A is preferred over Artefact B to upload a dataset.
2. Artefact A is preferred over Artefact B to create a subset.
3. Artefact B is preferred over Artefact A to find the way in the different steps.
4. Artefact B is preferred over Artefact A for the explanations of the workflow.

Model construction

We have created two hypotheses on model construction. The hypotheses are:

1. Artefact B is preferred over Artefact A for progress reporting on model construction.
2. Artefact B is preferred over Artefact A for model construction.

Model explanation

Because no user-stories matched a model explanation requirement, the first two hypotheses are about the model comparison. The latter hypotheses are about the desired explainability mentioned in the user-stories.

1. Artefact A is preferred over Artefact B for comparing results of model creation.
2. Artefact A is preferred over Artefact B for the explanation of missing data handling.
3. Artefact B is preferred over Artefact A for reading the produced model.
4. Users consider accuracy to be a good measure of model performance.
5. Users want to know the statistical power of the created model.
6. Users want to know the importance of each variable in the created model.

To test these hypotheses, the users first evaluate each artefact individually. For each statement, based on a user story, they can answer on a three-point scale: positive, neutral or negative. Each question provides the space to comment on a decision. Besides these hypotheses, the interviewees get the liberty to comment on parts that they would like to improve or remove for each of the artefacts. After that, they are asked to choose between the two artefacts for each hypothesis. The evaluation protocol is included in Appendix 7: Artefact evaluation protocol.

7.3 Artefact evaluation

This section describes the results of the artefact evaluation. First, we evaluate the hypotheses that are stated above for the three categories. Second, we discuss improvements for the artefacts.

Refined hypotheses testing

To evaluate the artefacts, we have tested our refined hypotheses with the interviewees. The interviewees had completed the same process on both artefacts to test the refined hypotheses. After evaluating the artefacts individually, the interviewees were asked to choose one artefact over the other for the different topics. The results of the artefact preference evaluation are available in Table 19; the individual assessments of the artefacts are available in Appendix 8: Artefact evaluations.

For the user interaction category, we can confirm three of the four hypotheses. Most users liked to upload a dataset with Artefact A and found the workflow and workflow explanation of artefact B to be preferable. We have to reject hypothesis two of the user-interaction. Users preferred Artefact B over Artefact A to create a subset with their data. Using code gives the users a feeling of insight and control over the process. **quotes removed for confidentiality**. The preference for Artefact B is minimal if we look to the individual artefact evaluation. Both A and B have four likes; the only difference is a dislike compared to an indifferent. Interesting to note is the difference in the individual evaluations and the overall performance. The workflow in Artefact A is preferred over the workflow in Artefact B if we consider the individual evaluations. However, the interviewees indicate a preference for artefact B when comparing the two artefacts.

Being in control is a central theme in the model construction category. **quotes removed for confidentiality**. We can see from Table 19 that only one interviewee preferred Artefact A over artefact B for progress reporting. All other interviewees prefer Artefact B over artefact A for both progress reporting and model construction, as we hypothesised. The individual evaluations of the artefacts confirm these findings. Artefact B had the most occurring value of ‘I like it’ for both aspects. Artefact A had the most occurring values of ‘I do not like it’ and ‘I am indifferent’ for progress reporting and model construction.

The model explanation part is all about understanding what has happened. Most researcher-physicians want to know why the computer created a model. For this category, we can only confirm the hypothesis about comparing results. Comparing results is perceived as more pleasant for Artefact A. **quotes removed for confidentiality.*

However, the individual evaluation of the artefact demonstrates that the researcher-physicians do not like the output of both artefacts because the AutoML methods do not offer explanations for the created models. The word cloud in Figure 30 illustrates the need for an explanation of the variables. The word cloud contains all answers to the why questions on the individual evaluations of the interviewees. The words variable, model, happening, explanation, control and understand all stand out. Most interviewees do not consider the output of TPOT as a model: **quotes removed for confidentiality**..

Table 19: Artefact evaluation preferences

	S1	S2	S3	S4	S5
User interaction					
Upload dataset	A	A	A	A	B
Create a subset	B	A	B	A	B
Workflow	B	A	B	B	B
Workflow explanation	B	B	B	A	B
Model construction					
Progress reporting	B	B	B	A	B
Model construction	B	B	B	B	B
Model explanation					
Compare results	B	A	A	A	A
Explanation missing data	B	B	B	A	B
Readability	B	A	A	A	B
Accuracy is a good measure	No	No	Yes	Yes	Yes
Statistical power	Yes	Yes	Yes	Yes	Yes
Variable importance	Yes	Yes	Yes	Yes	Yes

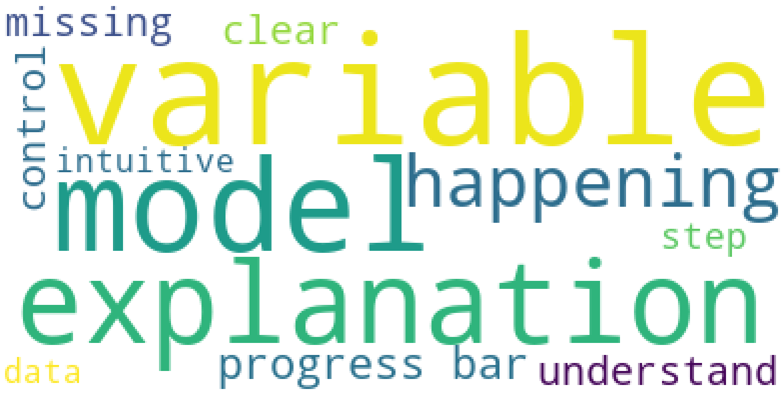


Figure 30: Word cloud of all comments on artefact evaluation questions

Artefact improvement

Part of design science is artefact improvement. Hence, we have asked the interviewees what they wanted to improve about the artefacts. Because the artefacts only differed in presentation form, we first discuss what can be improved for both artefacts and discuss improvements four for each of the two artefacts separately. Finally, we discuss the reaction to the suggestion of using AutoML in its current form for the interviewee's research.

The thing that was touched upon by all interviewees is the model output that TPOT gives. The output is not human-readable, and it is impossible to infer the contribution of each variable to the outcome. **quotes removed for confidentiality**

Variable importance and the statistical metrics of the created model are essential to researcher-physicians, as answered by all five interviewees and visible in the word cloud. Besides the variable importance, the interviewees want to have a more specific option to control the time used for model construction and the score measure that is used to evaluate the models. **quotes removed for confidentiality**. Both options were not present in Artefact A and not explained in Artefact B, but give the researcher-physicians a feeling of control over the process of model development.

Artefact A: Flask web-application

For artefact A, the interviewees mentioned four improvements: feedback on process steps, traceability of process steps, improvement of model comparison, and navigation. The first improvement is feedback on the results of the process within the web-application. The interviewees would like to see a message after they uploaded a dataset or created a subset. The second improvement is the traceability of the previous steps of the interviewee. A suggestion was the use of breadcrumbs to give the interviewees a view of what they have done in previous steps and where they are in the process of model creation. The third improvement was on the model overview, as mentioned above, the model used did not say anything to the interviewees, whereas the variables in the subset do. Hence, the interviewees suggested swapping these for each other. The fourth improvement is navigation. It was not self-explanatory for all interviewees; some interviewees suggested guidance through the whole process instead of coming back to the homepage after each step. Other interviewees suggested more links to the same actions or using numbers on the buttons in the steps of the process.

Artefact B: Jupyter notebook

For Artefact B, the interviewees mentioned four improvements: when to edit code, code explanation, reset the notebook to default and an explanation on how to run code blocks. The first improvement is being more explicit in where they have to edit code and where not. Suggestions are the collapse of code blocks, in which they do not have to edit something and highlights within the code about what parts to edit. The second improvement is a more thorough explanation of the visible code, both in-line and around the code. The explanation is requested so that the researcher-physicians understand what is happening; this helps them to feel in control. The third improvement was to have a function to roll back the notebook to its original state for when they broke something. The fourth improvement is a better explanation of how to run the code blocks and their output. For most interviewees, it was unclear how to run the code blocks and when the code blocks were done processing the contained code.

Artefact usage

After the artefact evaluation, we asked the interviewees if they would use AutoML in their research. Two of the interviewees mentioned that they would find it useful for data exploration and performance comparison of their models. Another interviewee mentioned that it believed in the power of (Auto)ML, but for adoption, the methods need to improve on explainability and transparency. The last two interviewees only would use AutoML in their research if the variable importance is part of the output of the AutoML method.

7.4 Conclusion

Based on the requirements from the previous chapter, we have created two similar artefacts with a different interface. We compared and evaluated these artefacts to find the best way to present AutoML to researcher-physicians in their knowledge discovery process. What we found is that a hybrid version of the two artefacts is preferred to interact with AutoML by the interviewees. Furthermore, AutoML needs to explain variable importance to make it usable in their research practice.

In the user interaction category, we found that for model interaction uploading a dataset is preferred with Artefact A. For subset selection, the workflow and its explanation are preferred to do in Artefact B. The preference for Artefact B is also present for model creation and progress reporting on model construction. Both artefacts scored low on the model explanation part. Artefact A was preferred over Artefact B as it provided a better overview. However, explaining variable importance is considered a must-have for the adoption of AutoML by researcher-physicians.

8 Conclusion

This research started with the question: '*How can we support healthcare professionals in their knowledge discovery process by applying AutoML?'.* To answer this question, we first answered five sub-questions. In the first section of this chapter, we discuss the main findings for all sub-questions individually. The second section will address the main research question.

8.1 Sub-questions

The first sub-question is *'What is the knowledge discovery process for healthcare professionals in their research?'*. In Chapter 3, we found that there is a need for the design of applications that enable domain experts to execute self-service data science as part of ADS. In healthcare, there is enormous potential for the usage of analytics. However, there is a lack of skilled people and no standard methodology. We selected CRISP-DM as the knowledge discovery process for this research, as it is widely adopted, adaptable to specific situations, and many methods derived from it.

The second sub-question is *'What are the capabilities of AutoML?'*. In Chapter 4, we described that AutoML is capable of selecting and configuring ML algorithms and their corresponding hyperparameters to optimise performance for a given dataset. Some AutoML methods combine ML algorithms with pre-processor steps to automatically create an ML pipeline that for optimal performance. Furthermore, we provided a synthesis in which we categorized the discussed AutoML methods.

The third sub-question is *'Which AutoML method performs best on a benchmark test given medical datasets?'*. To answer this question, we ran a benchmark test with four AutoML methods. From the benchmark test, we conclude that no method consistently significantly outperforms all other methods. However, we see that TPOT and Auto-Sklearn outperform Auto-WEKA on the sick dataset with a one-hour budget. However, Hyperopt-Sklearn is outperformed on both the one-hour and four-hour time budget for all datasets. When we extend the time limit from one hour to four hours, we do not see a significant improvement in model performance for any of the AutoML methods.

The fourth sub-question is *"What are the requirements of healthcare professionals to start using AutoML in their daily practice?"*. The most requirements of the healthcare professionals on AutoML were in the category model explanation. The interviewees stated that they wanted to know how model construction works and which choices are made by the AutoML method in the process of model construction. Furthermore, they want to know the importance of different variables in the created models. However, no assessed AutoML method could satisfy a single explainability requirement. TPOT satisfied most requirements in the other categories; user interaction, model construction and model usage. Hence, TPOT was selected as AutoML to integrate into the artefacts.

The fifth sub-question was *‘How does the selected AutoML method suit healthcare professionals in their knowledge discovery process?’*. To answer this question, we created two artefacts: a notebook and a web-application. We found that the notebook was preferred over the web-application to create a model with AutoML. The preference was due to the visibility of the code. The visibility provided a feeling of insight and control over what was happening. All interviewees were disappointed with the explanation of the created models.

We concluded that a hybrid version of the two artefacts would suit researcher-physicians best. A notebook part provides insight into the code and gives the interviewees a feeling of being in control whereas the web-application is preferred over the notebook regarding the uploading the dataset and the comparison of created models. The interviewees stated that if they would use AutoML in their research that it would be useful for data exploration. They would not use it to create models for their research.

8.2 Main research question

The main research question of this research was ‘*How can we support healthcare professionals in their knowledge discovery process by applying AutoML?’*. We found that AutoML is currently only suitable for the data understanding phase of the CRISP-DM method in this first study on possibilities for AutoML adoption in healthcare. Although AutoML is capable of modelling and data pre-processing, it misses an explanation for the decisions made in the modelling process. Part of medical knowledge discovery is finding the cause of a medical event. Because modelling decisions are not shared, and variable importance is absent in the result, AutoML does not support the discovery of new knowledge. However, the researcher-physicians point out that they see the added value of automatically finding out the best possible scores for their datasets. Furthermore, they mention that AutoML can help them in getting an understanding of their data in the data understanding phase of their knowledge discovery process.

9 Discussion

This chapter discusses the conducted research. First, we discuss the lessons learned. Second, we discuss possibilities for future research. Finally, we discuss the validity of this research.

9.1 Lessons learned

In this section, we discuss two lessons learned during this research. First, we discuss the suitability of AutoML for healthcare practitioners. Second, we discuss the bias in medical (analytics) publications.

Suitability of AutoML methods for researcher-physicians

After the elicitation of requirements, we found that researcher-physicians prefer to create logistic regression models in their research. There are multiple reasons why we find that AutoML in the way we assessed it, is not the best way to create models for researcher-physicians. First, AutoML originates from the CASH problem. If the algorithm type is pre-selected, AutoML can only contribute to HPO. As logistic regression only has two hyperparameters (Thornton et al., 2013), we question the need for using AutoML to tune these hyperparameters.

Second, even if we drop the constraint of logistic regression for model construction, previous research found that other ML models do not significantly outperform logistic regression models in medical studies (Christodoulou et al., 2019). Even when we drop the logistic regression model constraint, we still doubt the usability of AutoML due to the results of Christodoulou et al. and the inability of AutoML to explain the created models, although other researchers do not find the same results on general datasets (Gareth et al., 2013; Kotsiantis, 2007).

Third, there is no explanation of variable importance by the tested AutoML method. As model explainability and variable importance are essential requirements for researcher-physicians, this makes AutoML unsuitable in its current form. If the given explanations are satisfactory, AutoML might be useful in research. More on variable importance is available in the future work section below (9.2). If AutoPrognosis can deliver on its promises it can be a promising technology considering the user stories on model explanation.

Fourth, there is no structure in the pipelines created by TPOT; this can lead to very complicated pipelines in with three or four logistic regression models, all using each other's results as input. These constructions are hard to understand for domain experts. If we would use grammar to represent the pipeline like in RECIPE (Sá de et al., 2017) the grammar can help to create more understandable pipelines for researcher-physicians. Another option could be using fixed-pipeline methods based on BO. Finally, we have learned that there is a gap in the knowledge level of ML between literature and practice for researcher-physicians. The literature on AutoML states that AutoML aims to aid non-expert users of ML techniques (Thornton et al., 2013). However, we find that most non-expert users in the medical domain have no knowledge or education in programming. Hence, the current offering of AutoML techniques are still too technical for non-expert users in healthcare.

Bias in medical analytics publications

Most publications about healthcare analytics that we have come across during this research, have all published a positive result. Vollmer et al. (2018) noticed the same: there is a need to publish every positive result, but there are no real tests for the value of ML solutions in healthcare. If we add the findings of Christodoulou et al. (2019) to the fact that ML models do not perform significantly better than traditional methods in medical research we find a discrepancy between what is published and what is improving healthcare. Hence, we think that it would be helpful for the development of accurate methods for analytics in healthcare to publish results that do not provide a direct benefit. In this way, funding for research can be either used for improving methods that do not yet work in healthcare or in researching different methods for solving the same problem. When negative results are not published, we fear a waste of research funding by researchers continually reinventing the wheel and thus stalling research and innovation.

9.2 Future work

To build on this research, we discuss six options to extend this research. First, we discuss the validation of ML studies in healthcare. Second, we discuss AutoML model uncertainty. Third, we discuss new AutoML use-cases for healthcare. Fourth, we discuss improvements for the benchmark test and framework. Fifth, we discuss the interpretability of AutoML methods. Finally, we discuss the improvement of the artefacts.

Validation of ML studies in healthcare

Currently, there are no evaluation protocols in the medical domain to assess the added value of the application of ML. Christodoulou et al. (2019) found that this led to several studies where ML was preferred over logistic regression in algorithm performance, although there was no significant improvement. Vollmer et al. (2018) address the need for evaluation protocols of ML in healthcare. Such a framework should help to increase the confidence in digital healthcare solution and incorporate all stakeholders. One possible solution for this is the digital health scorecard, as it considers four different perspectives on ML in healthcare: technical, clinical, usability and costs (Mathews et al., 2019).

AutoML model uncertainty

One of the benefits that are proposed by AutoML is the reproducibility of created ML pipelines (Hutter et al., 2019; Kotthoff et al., 2017; Thornton et al., 2013). However, these authors state that the outcome of the creation of an ML pipeline with an AutoML method is dependent on the time budget allocated to the AutoML method. Besides that, EA based AutoML methods start with a random population. Thus, it is harder to reproduce the result of a single run without explicitly setting the seed.

Dusenberry et al. (2019) investigated model uncertainty in a medical context. They have found that as much as changing the seed can influence the prediction outcome for an individual patient. Hence, we argue that there should be more research on the stability of AutoML pipelines in the medical domain.

AutoML use-cases

In our overview in Chapter 4, we demonstrated that AutoML is applicable for tuning neural networks and creating classification and regression models. In healthcare, the fields with the most significant potential for the application of ML are image recognition and natural language processing (Vollmer et al., 2018). Because researcher-physicians accept black boxes in image recognition and natural language processing more than in traditional research, these ML tasks might be better suited to enable domain experts to work with AutoML in healthcare. Hence, we argue that the scope of AutoML use-cases could be widened to NLP and image recognition tasks if we want to accelerate the adoption of analytics in healthcare.

Benchmark test

To improve the benchmark test, as described in Chapter 5, we suggest three additions. The first is the inclusion of more AutoML models. AutoPrognosis (Alaa & van der Schaar, 2018), ML-Plan (Mohr et al., 2018) and Auto-Keras (Jin et al., 2018) are examples of different types of methods to include. The introduction of these methods would provide insights into the performance of a method designed for clinical practice, a method based on hierarchical task networks and a method for creating neural networks. The second addition would be the inclusion of new types of tasks to the benchmark test. In medical practice, there are more types of tasks than classification and regression. For example, survival analysis and multi-classification tasks are important prediction tasks in medical practice. Finally, the addition of more medical datasets that fit the requirements of Bischl et al. (2017) for benchmarking would be of value. The addition of more datasets would allow for a more reliable comparison of the available AutoML methods in the medical domain.

AutoML interpretability

As pointed out in Chapter 6 and 7, the explainability of ML models is crucial to adoption for researcher-physicians (Sung et al., 2003). Molnar (2019) argues that ML interpretability is crucial to the adoption of black-box algorithms in every sector. In healthcare, this barrier to adoption is even higher, as being able to explain decisions is part of the medical culture and vital to patient-doctor interaction. Hence, AutoML methods must become more interpretable for non-expert users. The interpretability technique should be model agnostic. In that way, the technique is suitable for all pipelines created by the AutoML methods (Ribeiro, Singh, & Guestrin, 2016). To improve the interpretability of AutoML models we propose three areas for further research on the interpretability of AutoML: Surrogate models, Local Interpretable Model-agnostic Explanations (LIME) (Molnar, 2019) and Shapley values (Shapley, 1953). However, others argue that we should improve the trust in artificial intelligence in healthcare rather than improving the interpretability. If the trust is high enough, the researcher-physicians will start using the black boxes (Bartoletti, 2019). Most researcher-physicians also do not precisely know how a car works. However, they still use cars in their daily lives.

Surrogate models

A surrogate model is a white-box model created parallel to a black-box solution. A surrogate model can be used to give the user of the black-box model a sense of understanding of what is happening inside by looking at the white-box model. There is no certainty that a surrogate model depicts the internal process of a black-box model, nor a clear-cut line to state when a surrogate model explains a black-box model well enough (Molnar, 2019). An example of a surrogate model is the usage of a logistic regression model to explain the inner workings of a neural network. Logistic regression models have explainable variable importance and can hence be used to mimic the neural network.

LIME

LIME models use artificial variations of data and inputs these into the created black-box function to create local surrogates of the black-box model. The local surrogates result in human-friendly explanations. A disadvantage of LIME is its locality; it can give conflicting explanations to explain the same model. This instability is considered to be a problem in the implementation of LIME in practice. *“Local surrogate models, with LIME as a concrete implementation, are very promising. But the method is still in the development phase and many problems need to be solved before it can be safely applied.”* (Alvarez-Melis & Jaakkola, 2018).

Shapley values

Shapley values (Shapley, 1953) are part of game theory. Each feature is called a player and the predicted value is called the pay-out. The goal of the Shapley values is to determine what the contribution for each player is. Contributions could be a combination of players (features) *“The Shapley value might be the only method to deliver a full explanation. In situations where the law requires explainability (...) the Shapley value might be the only legally compliant method because it is based on solid theory and distributes the effects fairly”* (Molnar, 2019). A disadvantage of Shapley values is that Shapley values return a simple value per feature, but no prediction model like LIME does. The absence of models means that Shapley values cannot be used to make statements about changes in prediction for changes in the input, such as: *“If my BMI were 5 points lower, I would have been considered for surgery.”*.

Artefact improvement

As we found in Chapter 7, the interviewees of this study prefer to interact with a ‘hybrid’ artefact. They want to have an artefact that has both GUI components and notebook components. Hence, we propose the development of an artefact that has a GUI for uploading a dataset and comparing model results. We propose more research on the interface for the creation of a subset. The interviewees had a slight preference for the notebook, but this was only by a narrow margin. To get a better overview of user-preferences, a larger sample needs to be involved.

The combined artefact should also make room for more ways of assessing model performance and time budget. Building a choice for evaluation metrics other than accuracy and a field in which the interviewees can set the time budget. Besides the addition of choices, an extension of the workflow could be added — especially which parts of the code are mandatory to edit, and which are optional.

9.3 Validity

In this section, we address the “subjective” nature of the data collection and analysis of this research (Kaplan & Maxwell, 2005). To assess the validity of this study, we look at three of the five aspects of validity for qualitative research, as proposed by Burke Johnson (1997). We do not discuss theoretical validity as the goal of design science research is on artefact creation instead of theory creation. We do not aim to explain a phenomenon. We also do not discuss Internal validity, as we do not aim to answer a question about a causal phenomenon. The main research question of this research is a ‘how’-question.

Descriptive validity

Descriptive validity is on the factual accuracy of the account of events as reported by the researcher (Burke Johnson, 1997). As a single researcher has conducted this research, hence there is a bias in data collection. A researcher is subjective by nature, and so are his data collection and analysis (Kaplan & Maxwell, 2005). Besides that, the relationship between the researcher and participants significantly influences what the participants reveal to the researcher (Kiegelmann, 2002, pp. 11–30). To mitigate this validity threat, we used a framework to set up the semi-structured interviews and the framework for evaluation of design science to set up the artefact evaluation. We recorded all interactions with the participants, and we took part in sessions to obtain peer-feedback on our research to increase the descriptive validity.

Interpretive validity

Interpretive validity is about accurately portraying the meaning that was attached by the participants to the objects that were studied (Burke Johnson, 1997). To mitigate this threat, we have sent the elicited user-stories to the participants to obtain feedback on our findings (Kaplan & Maxwell, 2005). Furthermore, we used data-triangulation by tapping into other sources to confirm our findings. To make sure we portrayed the meaning of the participants well, we used low inference descriptors by quoting participants in this research (Burke Johnson, 1997).

External validity

External validity is crucial if we want to generalise our findings to a larger part of the population (Burke Johnson, 1997). Although generalizability is not the primary purpose of this research, we will touch upon the subject. As this research conducts a case-study, the best way to generalise its findings is to find the similarity in subjects, objects and issues (Polit & Beck, 2010). As the sample size and characteristics are not valid for generalizability, the best method to generalise our findings is to identify similarity in other situations.

Characteristics that make the sample inapplicable for generalisation to medical professionals are the limited set of medical domains in which the participants operate and the fact that all participants decided to participate voluntarily. Hence, insights derived from this study are hard to generalise but could be a stepping stone for future research.

10 Bibliography

Aggarwal, C. (2015). *Data Mining The Textbook*. Springer. https://doi.org/10.1016/0304-3835(81)90152-X

Al-Busaidi, Z. Q. (2008). Qualitative research and its uses in health care. *Sultan Qaboos University Medical Journal*, *8*(1), 11–19. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/21654952%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3087733

Alaa, A. M., & van der Schaar, M. (2018). AutoPrognosis: Automated Clinical Prognostic Modeling via Bayesian Optimization with Structured Kernel Learning. In *International Conference on Machine Learning* (pp. 139–148). Retrieved from http://arxiv.org/abs/1802.07207

Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the Robustness of Interpretability Methods. *Arxiv Pre-Print*. Retrieved from http://arxiv.org/abs/1806.08049

Antonoglou, I., Fidjeland, A. K., Wierstra, D., King, H., Bellemare, M. G., Legg, S., … Mnih, V. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: a parallel overview. *IADIS European Conference Data Mining*, (January), 182–185. Retrieved from http://recipp.ipp.pt/handle/10400.22/136

Balaji, A., & Allen, A. (2018). Benchmarking Automatic Machine Learning Frameworks. *Arxiv Pre-Print*. Retrieved from http://arxiv.org/abs/1808.06492

Bartoletti, I. (2019). AI in Healthcare: Ethical and Privacy Challenges. In D. Riaño, S. Wilk, & A. ten Teije (Eds.), *Artificial Intelligence in Medicine* (pp. 7–10). Cham: Springer International Publishing.

Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., & Escobar, G. (2014). Big Data In Health Care: Using Analytics To Identify And Manage High-Risk And High-Cost Patients. *Health Affairs*, *33*(7), 1123–1131. https://doi.org/10.1377/hlthaff.2014.0041

Bay, S. D., Kibler, D., Pazzani, M. J., & Smyth, P. (2000). The UCI KDD archive of large data sets for data mining research and experimentation. *ACM SIGKDD Explorations Newsletter*, *2*(2), 14–18. https://doi.org/10.1145/380995.381030

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, *11*(3), 369–386. https://doi.org/10.2307/248684

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems*, 1–9. https://doi.org/2012arXiv1206.2944S

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*, 281–305. https://doi.org/10.1162/153244303322533223

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms. *Computational Science & Discovery*, 1–8.

Bischl, B., Casalicchio, G., Feurer, M., Hutter, F., Lang, M., Mantovani, R. G., …

Vanschoren, J. (2017). OpenML Benchmarking Suites and the OpenML100. *Arxiv Pre-Print*, 1–6. Retrieved from http://arxiv.org/abs/1708.03731

Blei, D. M., & Smyth, P. (2017). Science and data science. *Proceedings of the National Academy of Sciences*, *114*(33), 8689–8692. https://doi.org/10.1073/PNAS.1702076114

Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. In *Proceedings of COMPSTAT'2010* (pp. 177–186). https://doi.org/10.1007/978-3-7908-2604-3

Brachman, R. J., & Anand, T. (1996). The process of knowledge discovery in databases: A human-centered approach. *AAAI Press/ The MIT Press, Menlo Park, CA*, 37–57.

Brazdil, P., Carrier, C. G., Soares, C., & Vilalta, R. (2008). *Metalearning: Applications to Data Mining*. *Metalearning: Applications to Data Mining*. https://doi.org/10.1007/978-3-540-73263-1

Burke Johnson, R. (1997). Examining the Validity Structure of Qualitative Research. *Education*, *118*(Winter), 282–292.

Cabitza, F., Rasoini, R., & Gensini, G. F. (2017). Unintended consequences of machine learning in medicine. *JAMA - Journal of the American Medical Association*, *318*(6), 517–518. https://doi.org/10.1001/jama.2017.7797

Cao, L. (2017). Data Science: A Comprehensive Overview. *ACM Computing Surveys*, *95*(4), 714–716. https://doi.org/10.1215/00182168-3161769

Carmichael, I., & Marron, J. S. (2018). Data Science vs. Statistics: Two Cultures? *Japanese Journal of Statistics and Data Science*, *1*(1), 117–138. https://doi.org/10.1007/s42081-018-0009-3

Castillo-Montoya, M. (2016). Preparing for interview research: The interview protocol refinement framework. *How To Article*, *21*(5), 811–831. https://doi.org/Retrieved from: http://nsuworks.nova.edu/cgi/viewcontent.cgi?article=2337&context=tqr

ChaLearn. (2019). AutoML3 :: AutoML for Lifelong Machine Learning. Retrieved March 28, 2019, from https://competitions.codalab.org/competitions/19836

Chang, W., Roy, A., Grady, N., Reinsch, R., Underwood, M., Fox, G., … Laszewski von, G. (2018). NIST Big Data Interoperability Framework. *NIST*. https://doi.org/10.6028/NIST.SP.1500-1r1

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). Crisp-Dm 1.0. *CRISP-DM Consortium*, 76. https://doi.org/10.1109/ICETET.2008.239

Chawla, N. V., & Davis, D. A. (2013). Bringing big data to personalized healthcare: A patient-centered framework. *Journal of General Internal Medicine*, *28*(SUPPL.3), 660–665. https://doi.org/10.1007/s11606-013-2455-8

Chen, B., Wu, H., Mo, W., Chattopadhyay, I., & Lipson, H. (2018). Autostacker: A Compositional Evolutionary Learning System. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 402–409). ACM. https://doi.org/10.1145/3205455.3205586

Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., & Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal* *of Clinical Epidemiology*, *110*, 12–22. https://doi.org/10.1016/j.jclinepi.2019.02.004

Cohen, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Davenport, T. H., & Patil, D. J. (2012). Data scientist: The sexiest job of the 21st century. *Harvard Business Review*, *90*(10), 5. https://doi.org/10.1074/jbc.C900990199

Davis, A., Dieste, O., Hickey, A., Juristo, N., & Moreno, A. M. (2006). Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review. In *14th IEEE International Requirements Engineering Conference* (pp. 179–188).

Dedding, T. (2018). *Knowledge discovery for domain experts: A data preparation approach (MSc. Thesis)*. Utrecht University.

Deloitte. (2016). Advanced Analytics Standardized Project Methodology v6 0.

Demchenko, Y., Manieri, A., & Belloum, A. (2017). Part 2. Data Science Body of Knowledge (DS-BoK) Release 2, (January). https://doi.org/10.5281/zenodo.167591

Dewancker, I., McCourt, M., & Clark, S. (2015). Bayesian Optimization Primer. *SigOpt*, 2–5. Retrieved from https://sigopt.com/static/pdf/SigOpt_Bayesian_Optimization_Primer.pdf

Donoho, D. (2017). 50 Years of Data Science. *Journal of Computational and Graphical Statistics*, *26*(4), 745–766. https://doi.org/10.1080/10618600.2017.1384734

Drori, I., Krishnamurty, Y., Rampin, R., Paula Lourenco de, R., Piazentin Ono, J., Cho, K., … Freire, J. (2018). AlphaD3M : Machine Learning Pipeline Synthesis. In *ICML 2018 AutoML Workshop*.

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository. Retrieved from https://archive.ics.uci.edu/ml/index.php

Duan, L., Street, W. N., & Xu, E. (2011). Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems*, *5*(2), 169–181. https://doi.org/10.1080/17517575.2010.541287

Dusenberry, M. W., Tran, D., Choi, E., Kemp, J., Nixon, J., Jerfel, G., … Dai, A. M. (2019). Analyzing the Role of Model Uncertainty for Electronic Health Records. *Arxiv Pre-Print*, 1–14. Retrieved from http://arxiv.org/abs/1906.03842

Eijnatten, V., Gaag, L. Van Der, Grobbee, R., Jong, S. De, Karssenberg, D., Kemner, C., … Rijt, A. Van De. (2017). Focus area The Utrecht Platform for Applied Data Science (UPADS).

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996a). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, *17*(3), 37. https://doi.org/10.1609/aimag.v17i3.1230

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996b). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27–34. https://doi.org/10.1145/240455.240464

Feldman, B., Martin, E. M., & Skotnes, T. (2012). *Big Data in Healthcare Hype and Hope*. Retrieved from http://www.riss.kr/link?id=A99883549

Fenton, N. (2019). Bayes rule. Retrieved March 26, 2019, from http://www.eecs.qmul.ac.uk/~norman/BBNs/Bayes_rule.htm

Feurer, M., Eggensperger, K., Falkner, S., Lindauer, & Hutter, F. (2018). Practical Automated Machine Learning for the AutoML Challenge 2018. In *ICML 2018 AutoML Workshop*.

Feurer, M., Springenberg, J. T., Klein, A., Blum, M., Eggensperger, K., & Hutter, F. (2015). Efficient and Robust Automated Machine Learning. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2755–2763. https://doi.org/10.1016/j.sbspro.2015.09.090

Fortune. (2019). FORTUNE 500: 1999 Archive Full List 1-100. Retrieved March 28, 2019, from http://archive.fortune.com/magazines/fortune/fortune500_archive/full/1999/

Gareth, J., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. *Springer*. https://doi.org/10.1016/j.peva.2007.06.006

Gertosio, C., & Dussauchoy, A. (2004). Knowledge discovery from industrial databases. *Journal of Intelligent Manufacturing*, *15*(1), 29–37. https://doi.org/10.1023/B:JIMS.0000010073.54241.e7

Gibert, K., Horsburgh, J. S., Athanasiadis, I. N., & Holmes, G. (2018). Environmental Data Science. *Environmental Modelling and Software*, *106*, 4–12. https://doi.org/10.1016/j.envsoft.2018.04.005

Gijsbers, P., Ledell, E., Thomas, J., Poirier, S., Bischl, B., & Vanschoren, J. (2019). An Open Source AutoML Benchmark. In *ICML workshop on AutoML* (pp. 1–8).

Gijsbers, P., Vanschoren, J., & Olson, R. S. (2017). Layered TPOT: Speeding up tree-based pipeline optimization. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (pp. 49–68).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Retrieved from https://www.deeplearningbook.org/front_matter.pdf

Google. (2019). Cloud AutoML - Custom modellen voor machine learning | AutoML | Google Cloud. Retrieved March 22, 2019, from https://cloud.google.com/automl/

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337–355.

Grinberg, M. (2014). *Flask Web Development*. O'Rilley.

Harris, J., Shetterley, N., Alter, A., & Schnell, K. (2017). It Takes Teams to Solve the Data Scientist Shortage. *CIO Journal. - WSJ Blogs*, 2–5. Retrieved from https://blogs.wsj.com/cio/2014/02/14/it-takes-teams-to-solve-the-data-scientist-shortage/

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*(2). Retrieved from http://aisel.aisnet.org/sjis/vol19/iss2/4

Hevner, A. R., Ram, S., March, S., & Park, J. (2004). Design Science in Information Systems. *MIS Quarterly*, *28*(1), 75–105.

Hornby, G. S. (2006). ALPS : The Age-Layered Population Structure for Reducing the Problem of Premature Convergence. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2010). *Sequential model-based optimization for general algorithm configuration*. *Technical Report TR-2010–10, University of British Columbia, Computer Science*. https://doi.org/10.1007/978-3-642-25566-3_40

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automatic machine learning: methods, systems, challenges*. https://doi.org/10.1007/978-3-030-05318-5

Jalali, S., & Wohlin, C. (2012). Systematic literature studies: Database searches vs. backward snowballing. *Proceedings of the 6th ACM-IEEE International Symposium on Emprical Software Engineering and Measurement, ESEM*, 29–38. https://doi.org/10.1145/2372251.2372257

Janssen, J. H. N. (2019). *The right to explanation: means for 'white-boxing' the black-box? (MSc. Thesis)*. Tilburg University.

Jin, H., Song, Q., & Hu, X. (2018). Auto-Keras: Efficient Neural Architecture Search with Network Morphism. *Arxiv Pre-Print*. https://doi.org/10.1016/j.cardiores.2005.02.015

Kalousis, A. (2002). Algorithm Selection via Meta-Learning. *University of Geneva, Genebra*, 283.

Kaplan, B., & Maxwell, J. (2005). Qualitative Research Methods for Evaluating Computer Information Systems. In *Evaluating the organizational impact of healthcare information systems*. Springer New York.

KDD. (2019). KDD 2019 Call for Applied Data Science Papers. Retrieved February 27, 2019, from https://www.kdd.org/kdd2019/calls/view/kdd-2019-call-for-applied-data-science-papers

Kiegelmann, M. (2002). *The role of the researcher in qualitative psychology*. Ingeborg Huber Verlag.

Kim, J., Jeong, J., & Choi, S. (2016). AutoML Challenge : AutoML Framework Using Random Space Partitioning Optimizer. In *ICML, AutoML workshop* (pp. 1–4).

Kimberly, J., & Cronk, I. (2016). Making value a priority: how this paradigm shift is changing the landscape in health care. *Annals of the New York Academy of Sciences*, *1381*(1), 162–167. https://doi.org/10.1111/nyas.13209

Koh, H. C., & Tan, G. (2005). Data mining applications in healthcare, *19*(2), 64–72.

Komer, B., Bergstra, J., & Eliasmith, C. (2014). Hyperopt-Sklearn: Automatic HyperparameterConfiguration for Scikit-Learn. *ICML Workshop on AutoML*, 2825–2830.

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, *31*, 249–268. https://doi.org/10.1007/s10462-007-9052-3

Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Physics B: Atomic, Molecular and Optical Physics*, *18*, 1–5. https://doi.org/10.1088/0953-4075/40/9/S11

Larson, D., & Chang, V. (2016). A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, *36*(5), 700–710. https://doi.org/10.1016/j.ijinfomgt.2016.04.013

Lee, C. H., & Yoon, H.-J. (2017). Medical big data: promise and challenges. *Kidney*

Research and Clinical Practice, *36*(1), 3–11. https://doi.org/10.23876/j.krcp.2017.36.1.3

Ley, T. J., & Rosenberg, L. E. (2005). The Physician-Scientist Career Pipeline in 2005. *Jama*, *294*(11), 1343. https://doi.org/10.1001/jama.294.11.1343

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, *18*, 1–52. https://doi.org/10.1353/sof.0.0257

Malik, M. M., Abdallah, S., & Ala'raj, M. (2016). Data mining and predictive analytics applications for the delivery of healthcare services: a systematic literature review. *Annals of Operations Research*, *270*(1–2), 287–312. https://doi.org/10.1007/s10479-016-2393-z

Mangasariona, O. L., & Wolberg, W. H. (1990). Cancer Diagnosis via Linear Programming. *SIAM News*, *23*(5), 1–18.

Manyika, J., Chui, M., B., B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). Big data: The next frontier for innovation, competition and productivity. *McKinsey Global Institute*, (May).

Marbon, O., Mariscal, G., & Segovi, J. (2009). A Data Mining &amp; Knowledge Discovery Process Model. *Data Mining and Knowledge Discovery in Real Life Applications*. https://doi.org/10.5772/6438

Mariscal, G., Marbán, Ó., & Fernández, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. *Knowledge Engineering Review*, *25*(2), 137–166. https://doi.org/10.1017/S0269888910000032

Markow, W., Braganza, S., Taska, B., Hughes, D., & Miller, S. (2017). *The Quant Crunch*. Retrieved from https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=IML14576USEN

Martin, C. M., & Félix-Bortolotti, M. (2014). Person-centred health care: A critical assessment of current and emerging research approaches. *Journal of Evaluation in Clinical Practice*, *20*(6), 1056–1064. https://doi.org/10.1111/jep.12283

Mathews, S. C., McShea, M. J., Hanley, C. L., Ravitz, A., Labrique, A. B., & Cohen, A. B. (2019). Digital health: a path to validation. *Nature Digital Medicine*, *2*(1), 1–9. https://doi.org/10.1038/s41746-019-0111-3

Mendoza, H., Klein, A., Feurer, M., Springenberg, J. T., & Hutter, F. (2016). Towards Automatically-Tuned Neural Networks. *Proceedings of the Workshop on Automatic Machine Learning*, 58–65.

Mitchell, T. M. (1997). *Machine learning*. New York, NY: McGraw-Hill, Inc.

Mohr, F., Wever, M., & Hüllermeier, E. (2018). ML-Plan: Automated machine learning via hierarchical planning. *Machine Learning*, *107*(8–10), 1495–1515. https://doi.org/10.1007/s10994-018-5735-z

Molnar, C. (2019). *Interpretable Machine Learning*. leanpub.com. Retrieved from https://christophm.github.io/interpretable-ml-book/index.html

Nau, D., Au, T. C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research, 20,* 379–404.

Neff, G. (2013). Why Big Data Won't Cure Us. *Big Data*, *1*(3), 117–123. https://doi.org/10.1089/big.2013.0029

OECD. (2019). Health expenditure and financing. Retrieved March 28, 2019, from https://stats.oecd.org/Index.aspx?DataSetCode=SHA

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). Outline of a Design Science Research Project. *4th International Conference on Design Science Research in Information Systems and Technology*, 11. https://doi.org/10.1145/1555619.1555629

Olson, R. S., Bartley, N., Urbanowicz, R. J., & Moore, J. H. (2016). *Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science*. *Automated Machine Learning*. https://doi.org/10.1145/2908812.2908918

Olson, R. S., La Cava, W., Orzechowski, P., Urbanowicz, R. J., & Moore, J. H. (2017). PMLB: A large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, *10*(1), 1–13. https://doi.org/10.1186/s13040-017-0154-4

Olson, R. S., & Moore, J. H. (2016). TPOT: A tree-based pipeline optimization tool for automating machine learning. *Workshop on Automatic Machine Learning*, 66–74. Retrieved from http://proceedings.mlr.press/v64/olson_tpot_2016.html

Ooms, R., Spruit, M., & Overbeek, S. (2019). 3PM Revisited : Dissecting The Three Phases Method For Outsourcing Knowledge Discovery. *International Journal of Business Intelligence Research (IJBIR)*, *10*(1), 80–93.

Patil, H. K., & Seshadri, R. (2014). Big data security and privacy issues in healthcare. *Proceedings - 2014 IEEE International Congress on Big Data, BigData Congress 2014*. https://doi.org/10.1109/BigData.Congress.2014.112

Pedregosa, F., Michel, V., Grisel OLIVIER, Blondel, M., Prettenhofer, P., Weiss, R., … Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://doi.org/10.1007/s13398-014-0173-7.2

Peng, R. D., & Matsui, E. (2016). The Art of Data Science: A guide for anyone who works with Data. *Leanpub*, *53*, 160. https://doi.org/10.1017/CBO9781107415324.004

Piatetsky, G. (2014). CRISP-DM, still the top methodology for analytics, data mining, or data science projects. Retrieved February 18, 2018, from https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

Poli, R., Landon, W. B., McPhee, N. F., & Koza, J. R. (2007). *Genetic Programming An Introductory Tutorial and a Survey of Techniques and Applications* (Vol. 115). https://doi.org/10.1007/978-3-540-78293-3

Polit, D. F., & Beck, C. T. (2010). Generalization in quantitative and qualitative research: Myths and strategies. *International Journal of Nursing Studies*, *47*(11), 1451–1458. https://doi.org/10.1016/j.ijnurstu.2010.06.004

Pope, C., van Royen, P., & Baker, R. (2002). Qualitative methods in research on healthcare quality. *Quality & Safety in Health Care*, *11*(2), 148–152. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/12448807%0Ahttp://www.pubmedcentra l.nih.gov/articlerender.fcgi?artid=PMC1743608

Pritzker, P., and May, W. (2015). *NIST Big Data Interoperability Framework: Volume 1, Definitions*. *NIST Big Data Public Working Group Definitions and Taxonomies Subgroup* (Vol. 1). https://doi.org/10.6028/NIST.SP.1500-1

Pritzker, P., & May, W. (2015). *NIST Big Data Interoperability Framework: Volume 1, Definitions*. *NIST Big Data Public Working Group Definitions and Taxonomies Subgroup* (Vol. 1). https://doi.org/10.6028/NIST.SP.1500-1

Quanming, Y., Mengshuo, W., Hugo, J. E., Isabelle, G., Yi-Qi, H., Yu-Feng, L., … Yang, Y. (2018). Taking Human out of Learning Applications: A Survey on Automated Machine Learning. *Arxiv Pre-Print*, (November). https://doi.org/arXiv:1810.13306v1

Quinlan, J. R. (1986). Simplifying Decision Trees. *MIT Artificial Intelligence Laboratory*, 81–106.

Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, *2*(1), 1–10. https://doi.org/10.1186/2047-2501-2-3

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT press. https://doi.org/10.1142/S0129065704001899

Reutemann, P., Hall, M., Frank, E., Witten, I. H., Holmes, G., & Pfahringer, B. (2009). The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10. https://doi.org/10.1145/1656274.1656278

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-Agnostic Interpretability of Machine Learning. In *ICML Workshop on Human Interpretability in Machine Learning*. Retrieved from http://arxiv.org/abs/1606.05386

Rijksinstituut voor Volksgezondheid en Milieu. (2019). Zorguitgaven | Volksgezondheid Toekomst Verkenning. Retrieved March 28, 2019, from https://www.vtv2018.nl/zorguitgaven

Rijn van, J. (2019). OpenML OpenML Benchmarking Suites and the OpenML-CC18. Retrieved June 27, 2019, from https://www.openml.org/s/99

Roethlisberger, F. J. (1977). *The Elusive Phenomena*. Harvard Business School.

Rohanizadeh, S. S., & Moghadam, M. B. (2009). A Proposed Data Mining Methodology and its Application to Industrial Procedures. *Journal of Industrial Engineering*, *4*, 37–50. Retrieved from http://www.qjie.ir/?_action=showPDF&article=31&_ob=2e9f779810eaef02d9bcc00959616080&fileName=full_text.pdf

Sá de, A. G. C., Pinto, W. J. G. S., Oliveira, L. O. V. B., & Pappa, G. L. (2017). RECIPE: A grammar-based framework for automatically evolving classification pipelines. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10196 LNCS*, 246–261. https://doi.org/10.1007/978-3-319-55696-3_16

SAS. (2018). Getting Started with SAS(R) Enterprise Miner(TM) 4.3. Retrieved February 18, 2018, from http://support.sas.com/documentation/cdl/en/emgs/59885/HTML/default/viewer.htm#a000167823.htm

Shafique, U., & Qaiser, H. (2014). A Comparative Study of Data Mining Process

Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research ISSN*, *12*(1), 2351–8014. Retrieved from http://www.ijisr.issr-journals.org/

Shapley, L. S. (1953). A value for n-person games. In *Contributions to the Theory of Games* (pp. 307–317).

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., … Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *Arxiv Pre-Print*, 1–19. https://doi.org/10.1002/acn3.501

Spruit, M., & Jagesar, R. (2016). Power to the People! - Meta-Algorithmic Modelling in Applied Data Science. *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, (January), 400–406. https://doi.org/10.5220/0006081604000406

Spruit, M., & Lytras, M. (2018). Applied data science in patient-centric healthcare: Adaptive analytic systems for empowering physicians and patients. *Telematics and Informatics*, *35*(4), 643–653. https://doi.org/10.1016/j.tele.2018.04.002

Statista. (2018). Biggest companies in the world 2018 | Statista. Retrieved March 28, 2019, from https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-value/

Steinruecken, C., Smith, E., Janz, D., & Lloyd, J. (2018). The Automatic Statistician. In *Automatic machine learning: methods, systems, challenges* (pp. 175–188).

Sung, N., Crowley, W. F., Genel, M., Salber, P., Sandy, L., Sherwood, L. M., … Rimoin, D. (2003). Central Challenges Facing the National Clinical Research Enterprise. *JAMA - Journal of the American Medical Association*, *289*(10), 1278–1287.

Swearingen, T., Drevo, W., Cyphers, B., Cuesta-Infante, A., Ross, A., & Veeramachaneni, K. (2017). ATM: A distributed, collaborative, scalable system for automated machine learning. In *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017* (pp. 151–162). https://doi.org/10.1109/BigData.2017.8257923

Taylor, S. J., & Letham, B. (2018). *Forecasting at Scale*. https://doi.org/10.7287/peerj.preprints.3190v2

The Economist. (2017). The world’s most valuable resource is no longer oil, but data - Regulating the internet giants. Retrieved March 28, 2019, from https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2487575.2487629

Tukey, J. (1962). The future of Data Analysis, *37*(3), 688–697. https://doi.org/10.1214/aoms/1177705148

Vanschoren, J., Rijn, J. N. Van, Bischl, B., & Torgo, L. (2014). OpenML : networked science in machine learning. *ACM SIGKDD Explorations Newsletter*.

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. https://doi.org/10.1057/ejis.2014.36

Venkata Ramana, B., Babu, M. S. P., & Venkateswarlu, N. . (2011). A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis. *International Journal of Database Management Systems*, *3*(2), 101–114. https://doi.org/10.5121/ijdms.2011.3207

Vlaanderen, K., Brinkkemper, S., & van de Weerd, I. (2012). On the Design of a Knowledge Management System for Incremental Process Improvement for Software Product Management. *International Journal of Information System Modeling and Design*, 46–66.

Vleugel, A., Spruit, M., & Daal, A. Van. (2010). Historical Data Analysis through Data Mining From an Outsourcing Perspective: The Three-Phases Model. *International Journal of Business ...*, 1–21. https://doi.org/10.4018/jbir.2010070104

Vollmer, S., Mateen, B. A., Bohner, G., Király, F. J., Ghani, R., Jonsson, P., … Hemingway, H. (2018). Machine learning and AI research for Patient Benefit: 20 Critical Questions on Transparency, Replicability, Ethics and Effectiveness. In *The Alan Turing Institute*. Retrieved from http://arxiv.org/abs/1812.10404

Vries de, N. (2018). *Making machine learning accessible to healthcare professionals for the purpose of predicting medical adverse events (MSc. Thesis)*. Utrecht University.

Wang, X., Noor-E-Alam, M., Islam, M., Hasan, M., & Germack, H. (2018). A Systematic Review on Healthcare Analytics: Application and Theoretical Perspective of Data Mining. *Healthcare*, *6*(2), 54. https://doi.org/10.3390/healthcare6020054

Wang, Y., & Hajli, N. (2016). Exploring the path to big data analytics success in healthcare. *Journal of Business Research*, *70*, 287–299. https://doi.org/10.1016/j.jbusres.2016.08.002

Wang, Y., Kung, L. A., & Byrd, T. A. (2016). Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting and Social Change*, *126*, 3–13. https://doi.org/10.1016/j.techfore.2015.12.019

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., & De Feitas, N. (2013). Bayesian Optimization in a Billion Dimensions via Random Embeddings. In *Twenty-Third International Joint Conference on Artificial Intelligence* (Vol. 55, pp. 1778–1784). https://doi.org/10.1613/jair.4806

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Review. *Management Information Systems Quarterly*, *26*(2), xiii–xxiii. https://doi.org/10.2307/4132319

Wolpert, D. H., & Macready, W. G. (1996). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1145/1389095.1389254

Wu, C. F. J. (1997). Statistics=Data Science? *Identity of Statistics in Science Examined*.

Yu, B. (2014). Let us own Data Science. *IMS Bulletin*, *43*(7), 1–16.

Yu, L., Lai, K. K., Wang, S., & Huang, W. (2006). A Bias-Variance-Complexity Trade-Off Framework for Complex System Modeling. In *Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-642-21887-3

Zhang, Y., Bahadori, M. T., Su, H., & Sun, J. (2016). FLASH: Fast Bayesian Optimization for Data Analytic Pipelines, 1–21. https://doi.org/10.1145/2939672.2939829

Zhu, X. (2005). *Semi-supervised learning literature survey (Thesis)*. University of Wisconsin Madison.

11 Appendices

11.1 Appendix 1: Search queries

During the literature review, we have used the search terms listed below as a basis to find the papers that are found using snowballing.

- "Big data"
- "Data Science"
- Statistics
- "Machine Learning"
- "Automated machine learning" or AutoML
- Analytics
- All above in conjunction with either:
    - Applied
    - Self-service
    - Healthcare
    - "Patient-centric" or "personalised healthcare"
    - Physician-researcher
    - Researcher-physician
- "Automated statistician"
- "Automated algorithm selection and configuration"

Example of literature used as a starting point:

- Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). Automated Machine Learning-Methods, Systems, Challenges. *Automated Machine Learning*.
- Quanming, Y., Mengshuo, W., Hugo, J. E., Isabelle, G., Yi-Qi, H., Yu-Feng, L., ... & Yang, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*.
- Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013, August). 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* (pp. 847-855). ACM.

11.2 Appendix 2: Requirements for OpenML100 datasets

The OpenML website states the following requirements to a dataset to become a verified OpenML100 dataset (Bischl et al., 2017):

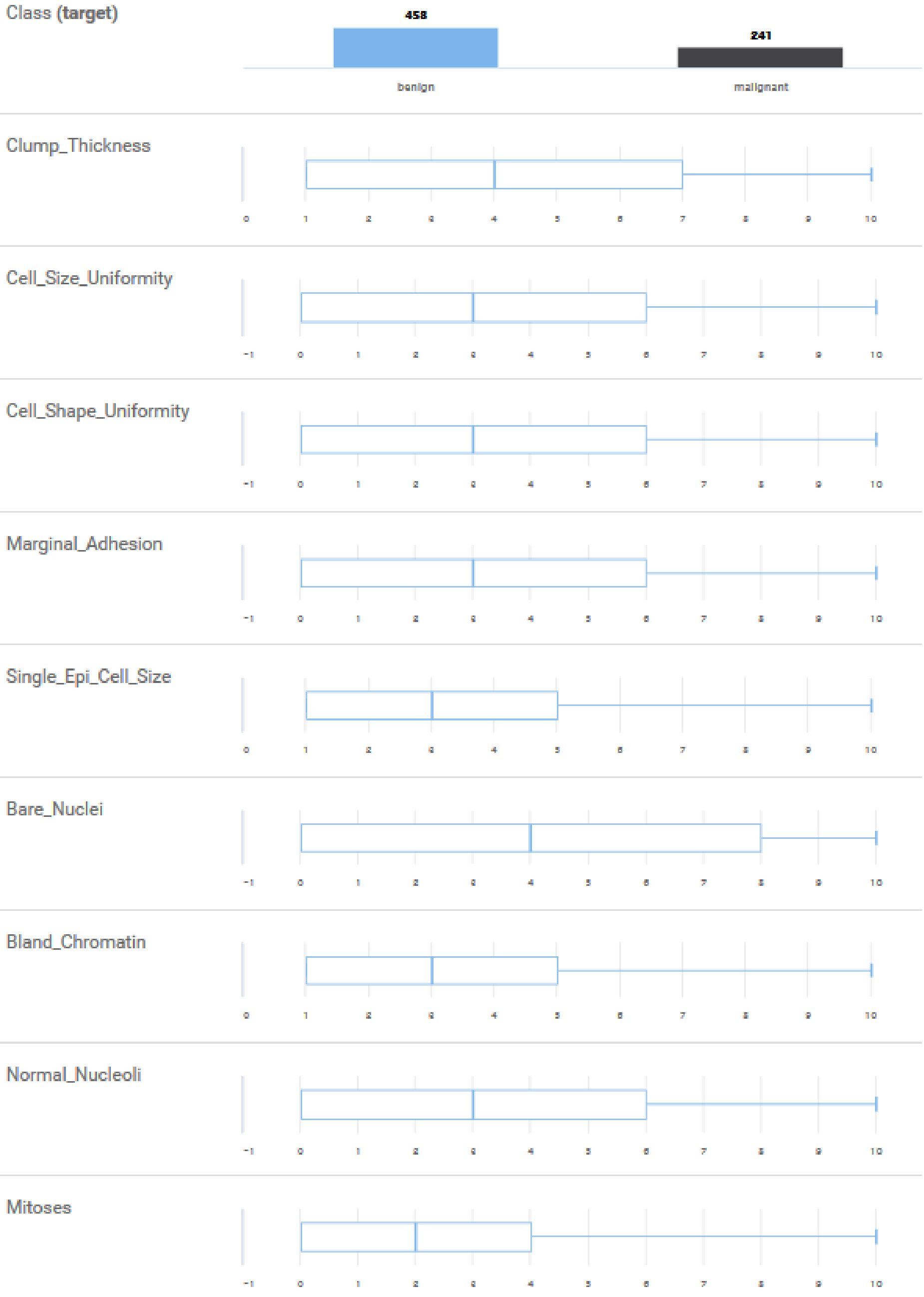
- The number of observations is between 500 and 100000 to focus on medium-sized datasets, that are not too small and not too big,
- The number of features does not exceed 5000 features to keep the runtime of algorithms low
- The target attribute has at least two classes
- Have classes with less than 20 observations
- The ratio of the minority class and the majority class is above 0.05, to eliminate highly imbalanced datasets, which require special treatment for both algorithms and evaluation measures.

The authors of OpenML excluded datasets which:

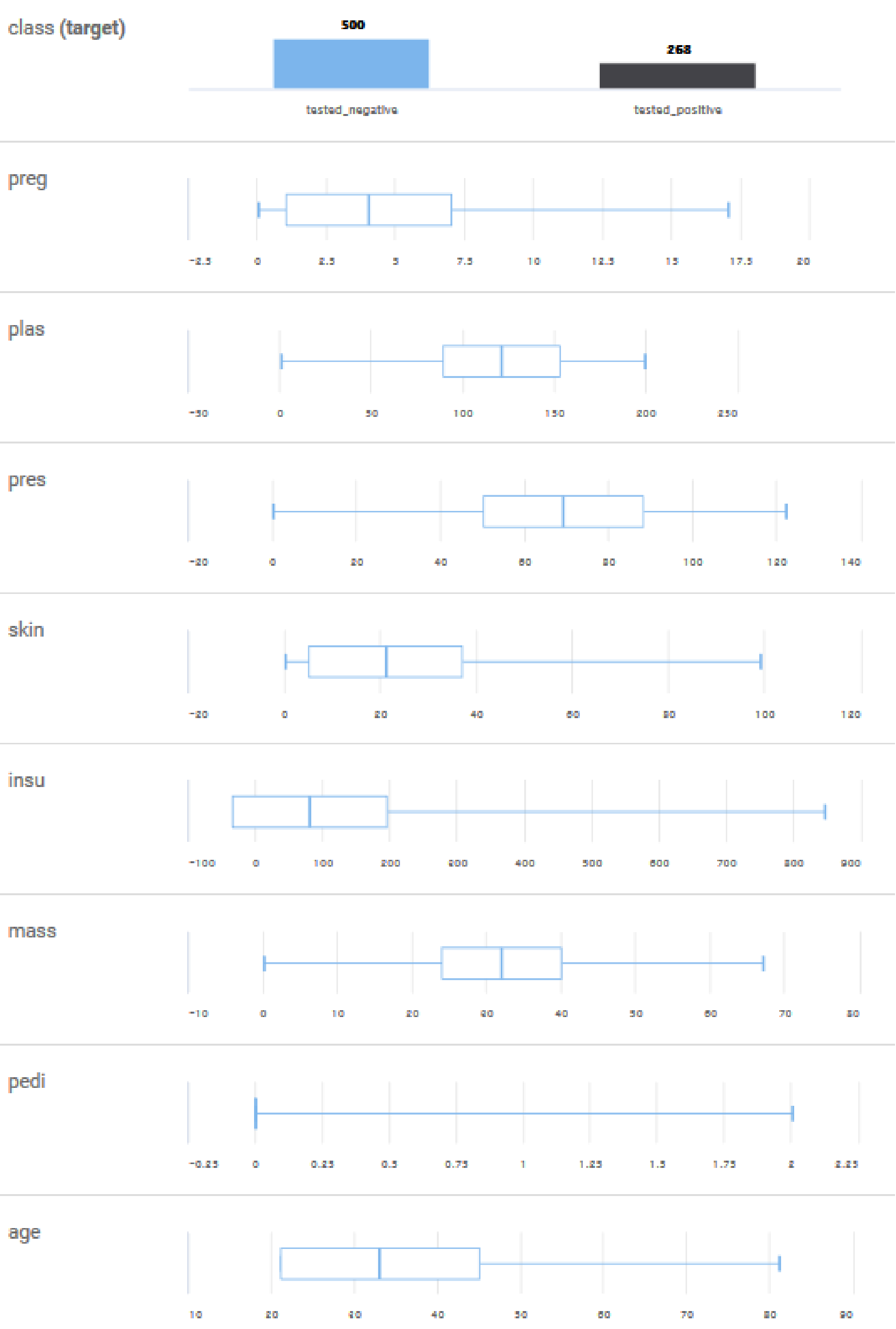
- Are artificially generated (not to confuse with simulated)
- Cannot be randomised via 10-fold cross-validation due to grouped samples or because they are time series or data streams
- Are a subset of a larger dataset
- Have no source or reference available
- Can be perfectly classified by a single attribute or a decision stump
- Allow a decision tree to achieve 100% accuracy on a 10-fold cross-validation task
- Have more than 5000 features after one-hot-encoding categorical features
- Are created by binarisation of regression tasks or multiclass classification tasks, or are sparse data (e.g., text mining data sets)

11.3 Appendix 3: Benchmark dataset descriptions

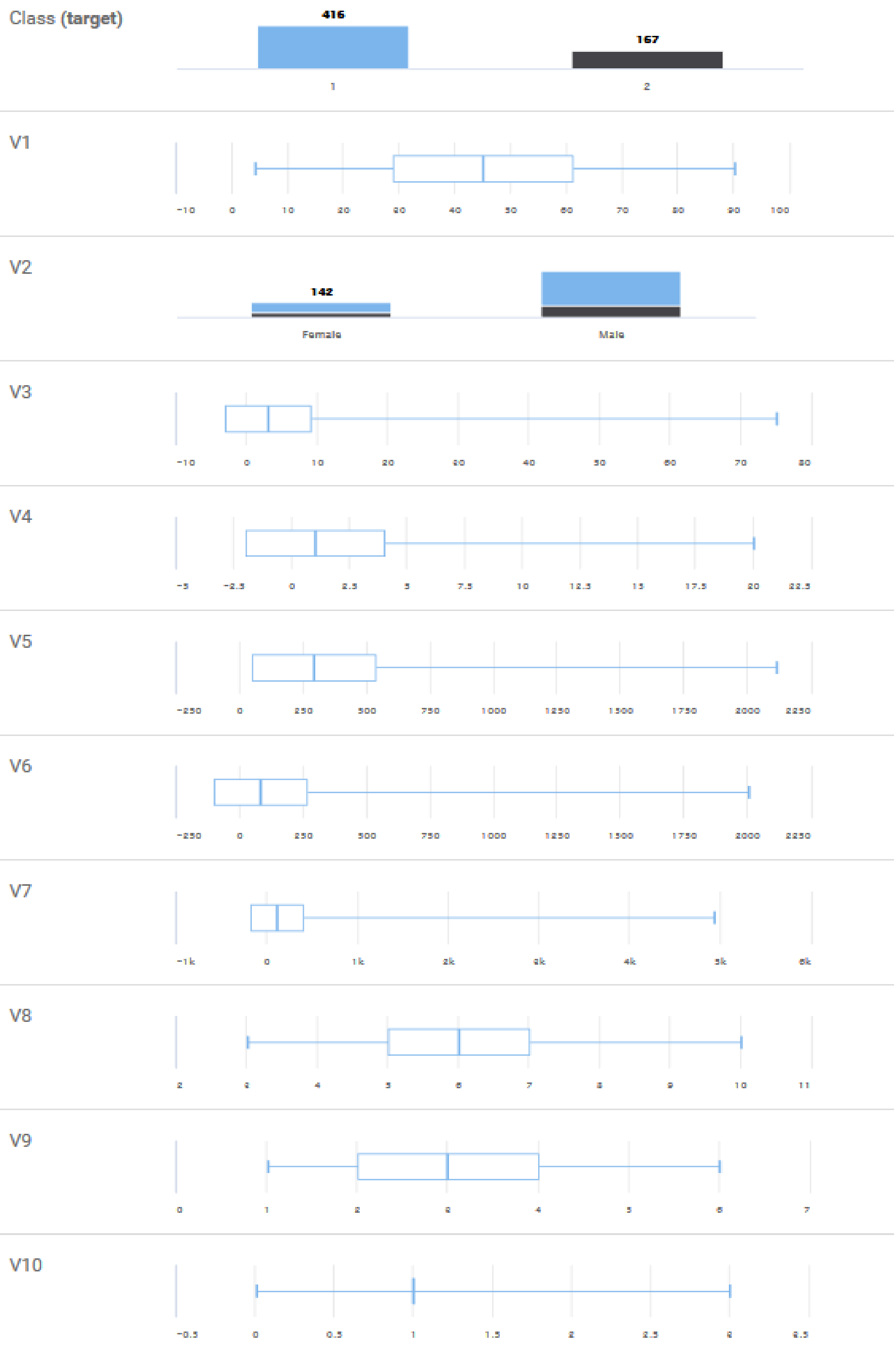
Breast cancer



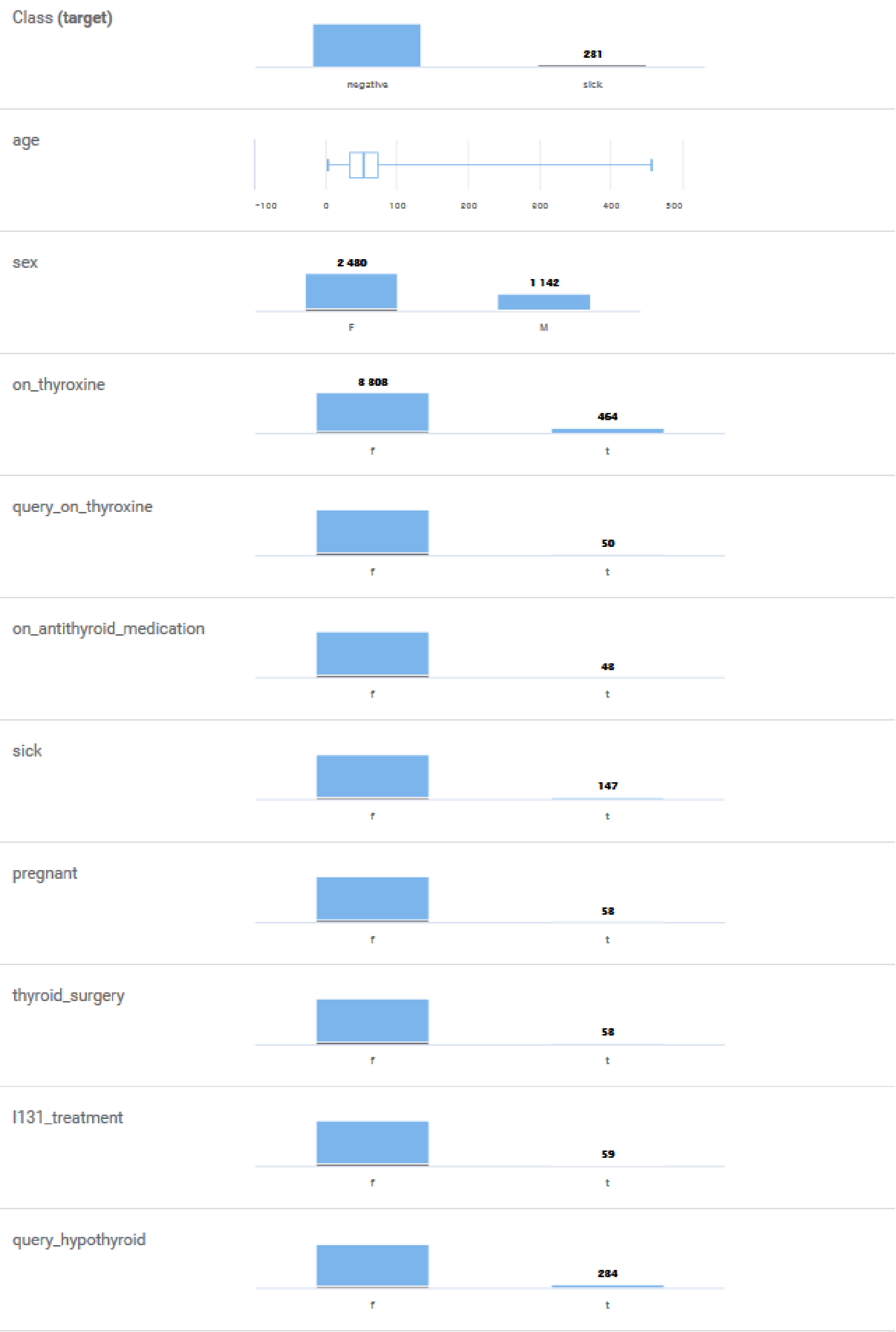
Diabetes


class (target)
500
268
tested_negative
tested_positive
preg
-2.5
0
2.5
5
7.5
10
12.5
15
17.5
20
plas
-50
0
50
100
150
200
250
pres
-20
0
20
40
60
80
100
120
140
skin
-20
0
20
40
60
80
100
120
insu
-100
0
100
200
300
400
500
600
700
800
900
mass
-10
0
10
20
30
40
50
60
70
80
pedi
-0.25
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
2.25
age
10
20
30
40
50
60
70
80
90

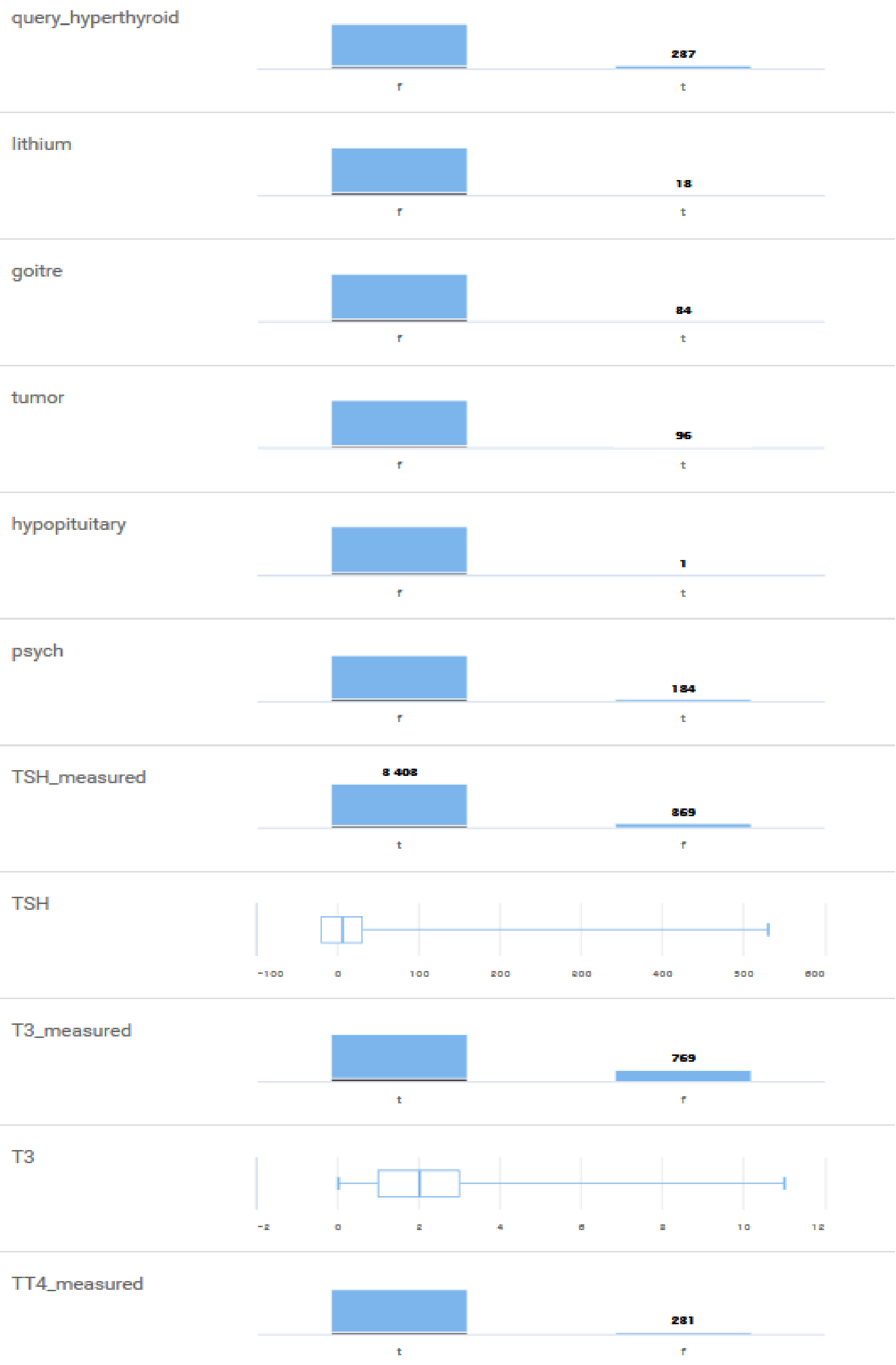
Indian Liver Patients



Sick

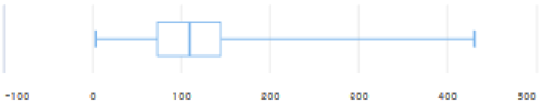


(continues on next page)

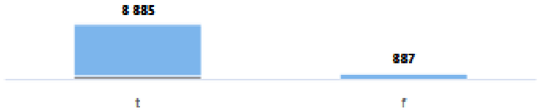


(continues on next page)

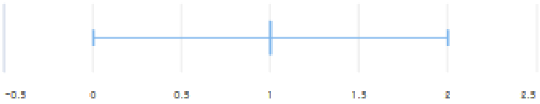
TT4



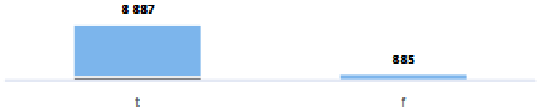
T4U_measured



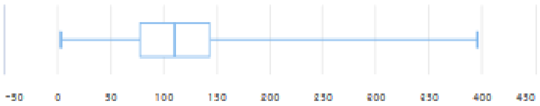
T4U



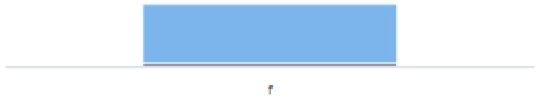
FTI_measured



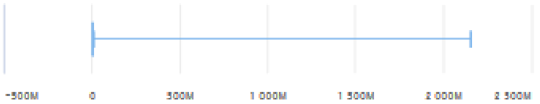
FTI



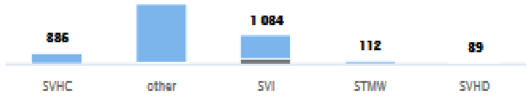
TBG_measured



TBG



referral_source



11.4 Appendix 4: Interview protocol

Introduction

- Thank the subject for participating
- Introduce myself and the research

Before we start, I am asking for your permission to record this interview and later transcribing it. All information will be anonymised and only be used for scientific research. It will not be shared outside of the university.

Start recording

Collect information on participant and their research

Name:
Function:
Experience:
Research topic

Collect information about the knowledge discovery process

1) Do you use a standard methodology for your knowledge discovery process?
2) How do you process data, if you do so, in your research?
3) Do you have support in data processing in your research?
4) Have you had an education in data processing?
5) What tools do you use for data processing?
6) What kind of challenges have you encounter while using data?

Collect information about Machine learning

7) What do you know about Machine Learning?
8) Is machine learning used in practice?
9) What do you know about the pitfalls in Machine Learning?
10) What do you know about the application of Machine Learning in healthcare?
    1. What do you think is important in applying machine learning in healthcare?
11) What kinds of statistical analysis/machine learning do you apply in your research? (if applicable)
12) How do you apply these techniques?
    1. Which tools do you use?
    2. What is your opinion on the tools used?
        i) What do you like?
        ii) What do you dislike about these languages?

Collect information on requirements on AutoML and automatic model delivery

- Delivery time
- Interpretability

- Alternatives

Stop recording

- Reassure confidentiality
- Close interview, thank interviewee

11.5 Appendix 5: Interview transcripts

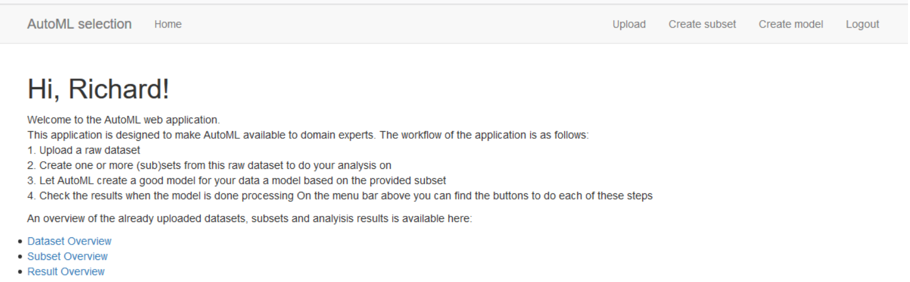
All interviews have been transcribed in Dutch, as they were held in Dutch. The interviewer is denoted by the letter I, the interviewee by the letter S.

The text of the interviews is removed for confidentiality

11.6 Appendix 6: Artefacts for researcher-physicians

The source code for the web as well as the notebook is available at this URL: https://github.com/richooms/healthcare_automl. Screenshots of the artefacts are available below.

Artefact A: Flask web-application



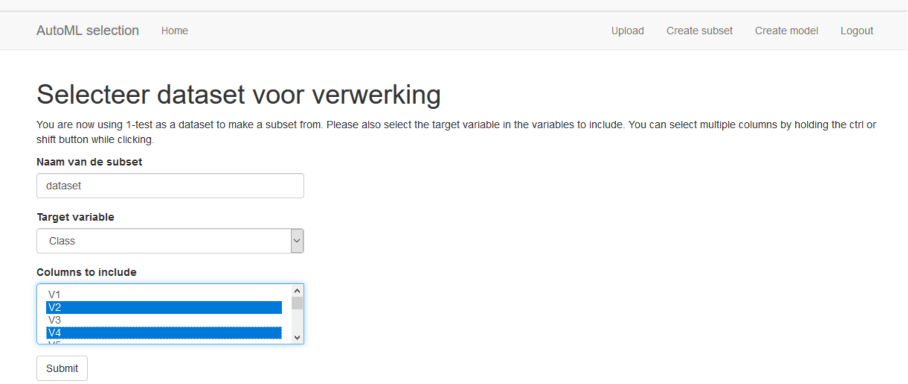
Hi, Richard!

Welcome to the AutoML web application.
This application is designed to make AutoML available to domain experts. The workflow of the application is as follows:
1. Upload a raw dataset
2. Create one or more (sub)sets from this raw dataset to do your analysis on
3. Let AutoML create a good model for your data a model based on the provided subset
4. Check the results when the model is done processing On the menu bar above you can find the buttons to do each of these steps

An overview of the already uploaded datasets, subsets and analysis results is available here:

- Dataset Overview
- Subset Overview
- Result Overview

Figure 31: Screenshot 1 of artefact A



Selecteer dataset voor verwerking

You are now using 1-test as a dataset to make a subset from. Please also select the target variable in the variables to include. You can select multiple columns by holding the ctrl or shift button while clicking.

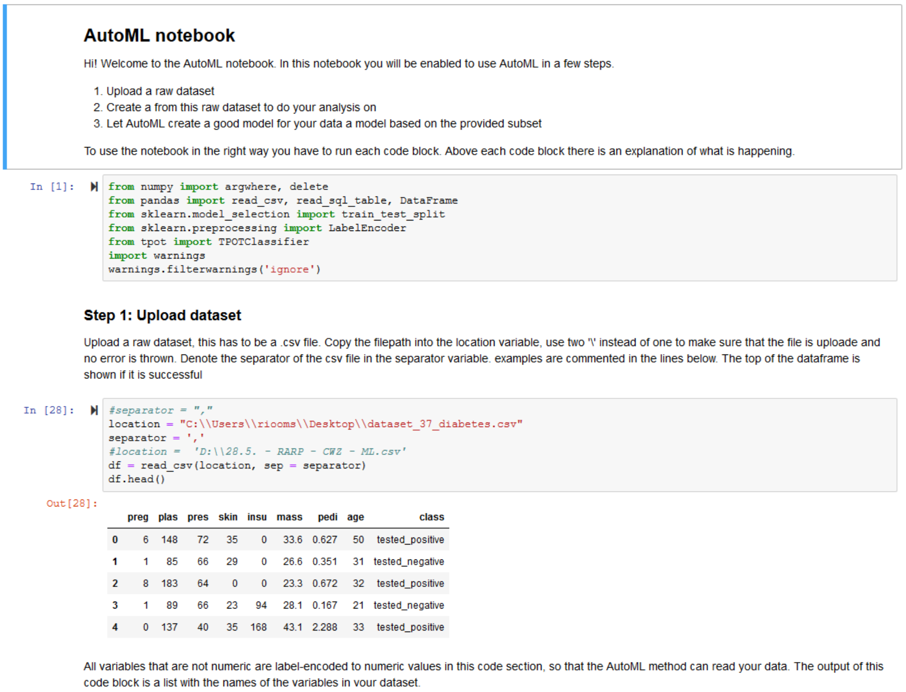
Naam van de subset: dataset

Target variable: Class

Columns to include: V1, V2 (selected), V3, V4 (selected)

Figure 32: Screenshot 2 of artefact A

Artefact B: Jupyter notebook



AutoML notebook

Hi! Welcome to the AutoML notebook. In this notebook you will be enabled to use AutoML in a few steps.

1. Upload a raw dataset
2. Create a from this raw dataset to do your analysis on
3. Let AutoML create a good model for your data a model based on the provided subset

To use the notebook in the right way you have to run each code block. Above each code block there is an explanation of what is happening.

In [1]:

```
from numpy import argwhere, delete
from pandas import read_csv, read_sql_table, DataFrame
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tpot import TPOTClassifier
import warnings
warnings.filterwarnings('ignore')
```

Step 1: Upload dataset

Upload a raw dataset, this has to be a .csv file. Copy the filepath into the location variable, use two '\' instead of one to make sure that the file is uploade and no error is thrown. Denote the separator of the csv file in the separator variable. examples are commented in the lines below. The top of the dataframe is shown if it is successful

In [28]:

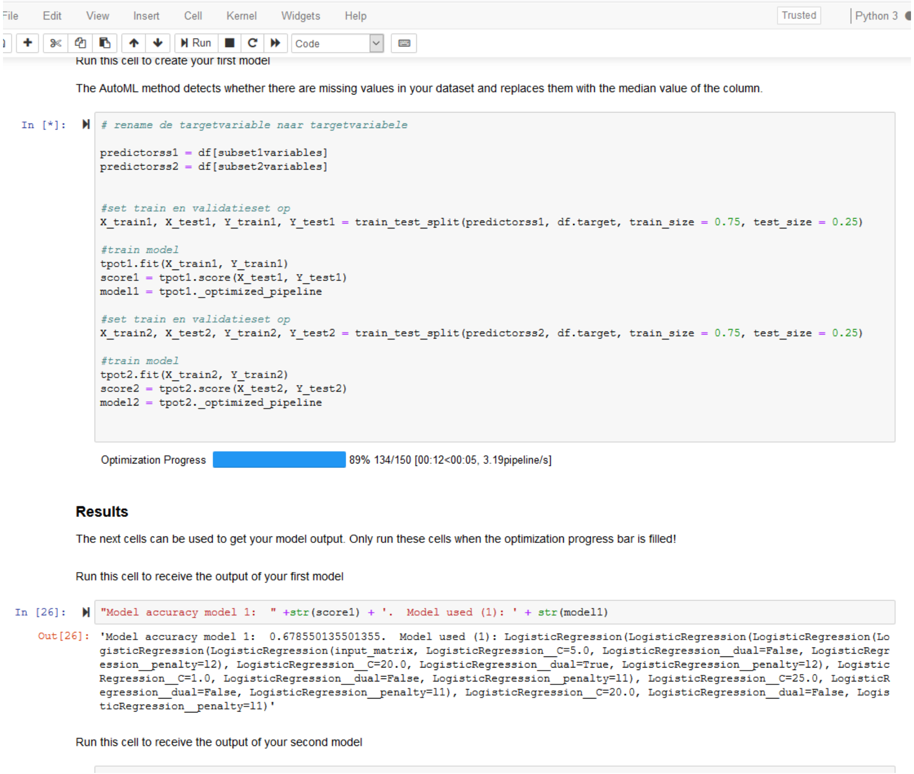
```
#separator = ","
location = "C:\\Users\\riooms\\Desktop\\dataset_37_diabetes.csv"
separator = ','
#location =  'D:\\28.5. - RARP - CWZ - ML.csv'
df = read_csv(location, sep = separator)
df.head()
```

Out[28]:

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive

All variables that are not numeric are label-encoded to numeric values in this code section, so that the AutoML method can read your data. The output of this code block is a list with the names of the variables in your dataset.

Figure 33: Screenshot 1 of artefact B



Run this cell to create your first model

The AutoML method detects whether there are missing values in your dataset and replaces them with the median value of the column.

```
In [*]:

# rename de targetvariable naar targetvariabele

predictorss1 = df[subset1variables]
predictorss2 = df[subset2variables]


#set train en validatieset op
X_train1, X_test1, Y_train1, Y_test1 = train_test_split(predictorss1, df.target, train_size = 0.75, test_size = 0.25)

#train model
tpot1.fit(X_train1, Y_train1)
score1 = tpot1.score(X_test1, Y_test1)
model1 = tpot1._optimized_pipeline

#set train en validatieset op
X_train2, X_test2, Y_train2, Y_test2 = train_test_split(predictorss2, df.target, train_size = 0.75, test_size = 0.25)

#train model
tpot2.fit(X_train2, Y_train2)
score2 = tpot2.score(X_test2, Y_test2)
model2 = tpot2._optimized_pipeline
```

```
Optimization Progress  89% 134/150 [00:12<00:05, 3.19pipeline/s]
```

Results

The next cells can be used to get your model output. Only run these cells when the optimization progress bar is filled!

Run this cell to receive the output of your first model

```
In [26]:  "Model accuracy model 1:  " +str(score1) + '.  Model used (1): ' + str(model1)

Out[26]: 'Model accuracy model 1:  0.678550135501355.  Model used (1): LogisticRegression(LogisticRegression(LogisticRegression(Lo
gisticRegression(LogisticRegression(input_matrix, LogisticRegression__C=5.0, LogisticRegression__dual=False, LogisticRegr
ession__penalty=l2), LogisticRegression__C=20.0, LogisticRegression__dual=True, LogisticRegression__penalty=l2), Logistic
Regression__C=1.0, LogisticRegression__dual=False, LogisticRegression__penalty=l1), LogisticRegression__C=25.0, LogisticR
egression__dual=False, LogisticRegression__penalty=l1), LogisticRegression__C=20.0, LogisticRegression__dual=False, Logis
ticRegression__penalty=l1)'
```

Run this cell to receive the output of your second model

Figure 34: Screenshot 2 of artefact B

11.7 Appendix 7: Artefact evaluation protocol

Introduction

- Thank the interviewee for participating
- Introduce the second experiment of the research

Before we start, I am asking for your permission to record the session in which we evaluate the artefacts and later transcribing it. All information will be anonymised and only be used for scientific research. It will not be shared outside of the university.

Introduce dataset

Explain the diabetes dataset. Pima Indians, consisting of women of 21 years and older 1) Number of times pregnant; 2) Plasma glucose concentration; 3) Diastolic blood pressure; 4) Triceps skinfold; 5) 2-hour serum insulin; 6) Body mass index; 7) Diabetes pedigree function, and 8) Age.

Introduce the purpose of the experiment.

Provide and introduce the first artefact.
Interact and evaluate the first artefact according to the artefact evaluation scheme
Evaluate results (model explanation part of the evaluation)

Provide/introduce the second artefact.
Interact and evaluate the second artefact according to the artefact evaluation scheme
Evaluate results (model explanation part of the evaluation)

Comparative experiment

Let the interviewee answer the questions of preference for two artefacts on the different aspects.

Conclusion

Thank the interviewee for participating. Ensure confidentiality

Artefact evaluation scheme

Interact with the artefact and start model creation (takes 15 minutes)

- During model creation, evaluate sections of user interaction and model construction
- After model creation, evaluate the model explanation

Individual evaluation artefact

User interaction			
Upload a dataset	I did not like it	I am indifferent	I like it
Why?			

Create a subset	I did not like it	I am indifferent	I like it
Why?			
Workflow	I did not like it	I am indifferent	I like it
Why?			
Workflow explanation	I did not like it	I am indifferent	I like it
Why?			
Model construction			
Progress reporting on model construction	I did not like it	I am indifferent	I like it
Why?			
Model construction	I did not like it	I am indifferent	I like it
Why?			
Model explanation			
Comparing results	I did not like it	I am indifferent	I like it
Why?			
Explanation of missing data handling	I did not like it	I am indifferent	I like it
Why?			
Readability of created model	I did not like it	I am indifferent	I like it
Why?			

What would you add to the artefact if possible?

What would you remove from the artefact if possible?

Comparative evaluation artefact A & B

User interaction		
Upload a dataset	Artefact A	Artefact B
Why?		
Create a subset	Artefact A	Artefact B
Why?		
Workflow	Artefact A	Artefact B
Why?		
Workflow explanation	Artefact A	Artefact B
Why?		

Model construction		
Progress reporting on model construction	Artefact A	Artefact B
Why?		
Model construction	Artefact A	Artefact B
Why?		
Model explanation		
Comparing results	Artefact A	Artefact B
Why?		
Explanation of missing data handling	Artefact A	Artefact B
Why?		
Readability of created model	Artefact A	Artefact B
Why?		
Accuracy is a good measure of model performance	I disagree	I agree
I want to know statistical power	I disagree	I agree
I want to know variable importance	I disagree	I agree

Discuss use-cases in which researcher-physicians think AutoML could contribute to their research.

Would you use AutoML in your research?

If they do not want to use it, what would be needed to have them start using it?

11.8 Appendix 8: Artefact evaluations

Interviewee 1

Individual evaluation artefact A (web app)

User interaction			
Upload a dataset			I like it
Why?	It is easy for new users		
Create a subset			I like it
Why?	It is easy with clicking		
Workflow		I am indifferent	
Why?	It all feels black box		
Workflow explanation	I did not like it		
Why?	There is no explanation in the app		

Model construction			
Progress reporting on model construction	I did not like it		
Why?	The computer does not show how accuracy is determined		
Model construction		I am indifferent	
Why?	I do not understand how it is constructed		

Model explanation			
Comparing results		I am indifferent	
Why?			
Explanation of missing data handling		I am indifferent	
Why?	I don't understand how missings were handled		
Readability of created model	I did not like it		
Why?	The model used formula is difficult to understand		

What would you add to the artefact if possible?

- Being able to select other measurements for model performance
- Calculation of:
  - AUC
  - Calibration
  - Sensitivity
  - Specificity
  - The net benefit of decision curve analysis

What would you remove from the artefact if possible?

- The usage of accuracy

Individual evaluation artefact B (notebook)

User interaction			
Upload a dataset			I like it
Why?	It is relatively easy		
Create a subset			I like it
Why?	It is easy		
Workflow		I am indifferent	
Why?	Steps were easy		
Workflow explanation		I am indifferent	
Why?	The code gives more information		

Model construction			
Progress reporting on model construction		I am indifferent	
Why?			
Model construction			I like it
Why?	It is easy to create more models		

Model explanation			
Comparing results	I did not like it		
Why?			
Explanation of missing data handling	I did not like it		
Why?	No explanation		
Readability of created model	I did not like it		
Why?	Difficult to understand for me, I want to know the coefficients of the variables		

What would you add to the artefact if possible?

- Being able to select other measurements for model performance
- Calculation of:
    - AUC
    - Calibration
    - Sensitivity
    - Specificity
    - The net benefit of decision curve analysis

What would you remove from the artefact if possible?

Comparative evaluation artefact A & B

User interaction		
Upload a dataset	Artefact A	
Why?	It is easier to select	
Create a subset		Artefact B
Why?	I like to see the variables in the code sheet	
Workflow		Artefact B
Why?	Steps are easier to follow	
Workflow explanation		Artefact B
Why?	Steps are easier to follow	

Model construction		
Progress reporting on model construction		Artefact B
Why?	It gives you more information	
Model construction		Artefact B
Why?	It feels like more info is given	

Model explanation		
Comparing results		Artefact B
Why?	I have to make a choice, but I’m indifferent	
Explanation of missing data handling		Artefact B
Why?	I am indifferent. I miss insight into missing data in both models	
Readability of created model		Artefact B
Why?	You can see more specifics of the created model	
Accuracy is a good measure of model performance	I disagree	
I want to know statistical power		I agree
I want to know variable importance		I agree

Discuss use-cases in which researcher-physicians think AutoML could contribute to their research.

Would you use AutoML in your research?
Yes, if it would provide more statistics for prediction models.
If they do not want to use it, what would be needed to have them start using it?

I would like a good tutorial on what machine learning or automated machine learning really does.

Remark: People should be wary about what they do with machine learning models in medical practice. Soon there, you will need a certificate to prove the robustness of your model (CE certificate).

Interviewee 2
Interact with the artefact and start model creation (takes 15 minutes)

- During model creation, evaluate sections of user interaction and model construction
- After model creation, evaluate the model explanation

Individual evaluation artefact A (web-application)

User interaction			
Upload a dataset			I like it
Why?	The process is good and intuitive		
Create a subset			I like it
Why?	It is intuitive to me		
Workflow			I like it
Why?	It is clean		
Workflow explanation			I like it
Why?	There is not too much information.		

Model construction			
Progress reporting on model construction	I did not like it		
Why?	I want to see a timer or progress bar. I have to know if I can get a cup of coffee		
Model construction		I am indifferent	
Why?	The usability is high, but my stomach gives me an uncomfortable (unheimlich) feeling.		

Model explanation			
Comparing results	I did not like it		

Why?	It does not show an answer to my question. I do not consider this a model.		
Explanation of missing data handling	I did not like it		
Why?	With the model overview, it is a bit late; I'd rather have it during the model creation phase.		
Readability of created model	I did not like it		
Why?	I cannot read or interpret this.		

What would you add to the artefact if possible?

- More flexibility in uploading of the format of a dataset (also accept; as a separator for CSV files.)
- Feedback if an operation (uploading or subset creation) is successful.
- Numbers at the buttons in the top bar
- Links in the steps
- A bigger field for the variable selection
- The assumption that your just created model is your default
- Being led through the process instead of returning to the home page each time
- Showing subset variables in the results overview and see what has happened to their importance.

What would you remove from the artefact if possible?

- Model used text, because it does not tell me anything

Individual evaluation artefact B (notebook)

User interaction			
Upload a dataset			I like it
Why?	It was doable for me		
Create a subset		I am indifferent	
Why?	It was okay, but not intuitive		
Workflow		I am indifferent	
Why?	The idea appeals to me, but it has to be clearer when I have to do something, or not		
Workflow explanation		I am indifferent	
Why?	The text can be clearer in what every step does		

Model construction			
Progress reporting on model construction			I like it
Why?	I liked the percentage and progress bar		

Model construction			I like it
Why?	Although it is a black box, it is user-friendly and gives me a sense of control		

Model explanation			
Comparing results	I did not like it		
Why?	It is hard to find out what the role of a variable is.		
Explanation of missing data handling	I did not like it		
Why?	This is redundant, no missing data is accepted during the recoding process		
Readability of created model	I did not like it		
Why?	It does not answer my question.		

What would you add to the artefact if possible?

- Make the text more foolproof, tell me where I can edit something and where not
- Use a green bar upon completion
- Let me input my own time constraint options
- I want more information and choice on the type of output, as well as the choices that are made on the in or exclusion of variables
- Fold the code blocks that you don’t have to edit
- Use more colours and highlights in the text and code that I can or have to edit.

What would you remove from the artefact if possible?

Comparative evaluation artefact A & B

User interaction		
Upload a dataset	Artefact A	
Why?	It is less typing and hassle for me	
Create a subset	Artefact A	
Why?	Less typing, no issues with comma’s etc	
Workflow	Artefact A	
Why?	Usability is better	
Workflow explanation		Artefact B
Why?	It gives me a better understanding of what is happening	

Model construction		
Progress reporting on model construction		Artefact B

Why?	The progress bar is a unique selling point	
Model construction		Artefact B
Why?	I feel more engaged in the process of model construction	

Model explanation		
Comparing results	Artefact A	
Why?	There is a better overview of what happened	
Explanation of missing data handling		Artefact B
Why?	The place in the process is better	
Readability of created model	Artefact A	
Why?	Just the layout. I still consider the model unreadable	
Accuracy is a good measure of model performance	I disagree, I want to know more statistical properties, sensitivity, AUROC, specificity and more	
I want to know statistical power		I agree I want to know margins and confidence intervals in a results page
I want to know variable importance		I agree, to make this usable I need to know which variable has influence

Discuss use-cases in which researcher-physicians think AutoML could contribute to their research.

Would you use AutoML in your research? No, not at this point in time. I cannot see what the relative importance of a variable is. In my opinion, that is the key to adoption. Give a user more insight into the process of variable selection. I do believe in the power of machine learning.

If they do not want to use it, what would be needed to have them start using it?

Remark:
It is very nice how you mimicked the difference in the two artefacts between R and SPSS in your artefacts

Interviewee 3

Evaluation artefact A

User interaction			
Upload a dataset		I am indifferent	

Why?	It worked as I expected		
Create a subset	I did not like it		
Why?	There was no clear overview. I want to see what happens and to see my variables		
Workflow	I did not like it		
Why?	To me, there is no difference in creating a model or a subset. So this seemed to be a redundant and unnatural step for me		
Workflow explanation		I am indifferent	
Why?	It was not intuitive, but I also do not dislike it.		

Model construction			
Progress reporting on model construction	I did not like it		
Why?	I want to see the progress and see what is happening ‘under the hood.’		
Model construction	I did not like it		
Why?	I missed an overview and control of what was happening		

Model explanation			
Comparing results	I did not like it		
Why?	In my opinion, there is no result, as there is no model. At least not what I consider a model. I cannot see which variable had which influence.		
Explanation of missing data handling	I did not like it		
Why?	It is the wrong way of going about data imputation. I should be able to choose what happens with rows containing missing values.		
Readability of created model	I did not like it		
Why?	Again, this is not a model, in my opinion. Nor was it readable		

What would you add to the artefact if possible?

- I would like to have a more transparent process. I want to see the steps in one big overview.

What would you remove from the artefact if possible?

- The distinction between the subset and a model that is one thing, in my opinion.

Evaluation artefact B

User interaction			
Upload a dataset		I am indifferent	
Why?	There is a learning curve to it, but it works fine.		
Create a subset			I like it

Why?	I enjoy the typing, feels like I am in control.		
Workflow		I am indifferent	
Why?	I do not understand all the code. But being able to see the code makes it feel like I'm more in control		
Workflow explanation	I did not like it		
Why?	I need more explanation to understand everything that is happening in the code		

Model construction			
Progress reporting on model construction			I like it
Why?	I like the progress bar		
Model construction			I like it
Why?	I feel in control, and I have an overview of what has happened in the first step		

Model explanation			
Comparing results	I did not like it		
Why?	These are not results to me		
Explanation of missing data handling	I did not like it		
Why?	I think you should get a choice for handling missing data. Median imputation is not the way to do it.		
Readability of created model	I did not like it		
Why?	It is not a model, and I do not see variable importance		

What would you add to the artefact if possible?

- More explanation of the code
- A better, result. In other words a model instead of some text that does not tell me anything

What would you remove from the artefact if possible?

- The text in the blocks could be replaced by help boxes that pop up to explain to you what is happening instead of the text

Comparative evaluation artefact A & B

User interaction		
Upload a dataset	Artefact A	
Why?	It is like it works in all other places as well	

Create a subset		Artefact B
Why?	I have more control over what is happening and which variables are kept in a subset	
Workflow		Artefact B
Why?	Everything that happens is visible and listed in one screen	
Workflow explanation		Artefact B
Why?	It explains it better in my opinion	

Model construction		
Progress reporting on model construction		Artefact B
Why?	Because of the progress bar and the fact that it is one document, hence I can see what is happening and how far I am in the process	
Model construction		Artefact B
Why?	I have a better overview, and I feel more in control	

Model explanation		
Comparing results	Artefact A	
Why?	The columns make it more comprehensible	
Explanation of missing data handling		Artefact B
Why?	I missed in in artefact A	
Readability of created model	Artefact A	
Why?	The tabular overview gives me a better overview	
Accuracy is a good measure of model performance		I agree if there is an actual model being created.
I want to know statistical power		I agree, as I want to know more: AUROC, R^2, precision, sensitivity, recall etc.
I want to know variable importance		I agree that is what a model is all about in my opinion

Discuss use-cases in which researcher-physicians think AutoML could contribute to their research.

Would you use AutoML in your research?

NO, it would not be useful at all. It does not provide me with any insights. I need a model as an outcome which shows variable importance and a rationale in why variables are selected or dropped from a model.

If they do not want to use it, what would be needed to have them start using it?
Explainability of the inner workings of the AutoML method and its created pipeline

Interviewee 4
Interact with the artefact and start model creation (takes 15 minutes)

- During model creation, evaluate sections of user interaction and model construction
- After model creation, evaluate the model explanation

Individual evaluation artefact A (web app)

User interaction			
Upload a dataset			I like it
Why?	It is very easy		
Create a subset			I like it
Why?	It is very easy		
Workflow			I like it
Why?	It goes very smooth		
Workflow explanation			I like it
Why?	Everything works as it should be. “speaks for itself.”		

Model construction			
Progress reporting on model construction		I am indifferent	
Why?	Some countdown or progress bar would be nice; I want to know if I can get coffee		
Model construction			I like it
Why?	It works well, depending on the outcome		

Model explanation			
Comparing results	I did not like it		
Why?	I do not see what all variables do and/or contribute to the solution		
Explanation of missing data handling		I am indifferent	
Why?	It does not grab attention; I was looking for it due to the evaluation of the other artefact		

Readability of created model	I did not like it		
Why?	I don’t understand what it does and says. I’d rather have variables and their importance. This gives me an insight into their score		

What would you add to the artefact if possible?

- I want to see what has happened. I want to see more traceability and want to find out what I have done before. Traceback my own steps
- I want to see the variable importance as part of the result
- I want to see why variables are important, building on the case above.
- I want to see the covariance between the variables.

What would you remove from the artefact if possible?

Individual evaluation artefact B (notebook)

User interaction			
Upload a dataset			I like it
Why?	I see what I do, and I like that		
Create a subset			I like it
Why?	I see what I do; I like that		
Workflow	I did not like it		
Why?	It is unclear if there is a result after running a code block		
Workflow explanation		I am indifferent	
Why?	It should be more clear if I have to do something or not; I do not understand what is happening and what choices I have intuitively		

Model construction			
Progress reporting on model construction		I am indifferent	
Why?	I want to see the remaining time. Can I get coffee or not?		
Model construction			I like it
Why?	It gives me the feeling of being in control. I can seem to understand what is happening.		

Model explanation			
Comparing results	I did not like it		
Why?	I miss parts, such as a confidence interval. That is the stuff I want to know		
Explanation of missing data handling	I did not like it		
Why?	I want to have more explanation per line of code		

Readability of created model	I did not like it		
Why?	I am only interested in the variables. The other stuff is unnecessary in my opinion		

What would you add to the artefact if possible?

- A completed message after code blocks have ran
- A guide on what happens if I break something or remove something that I should not remove
- The confidence interval for all variables
- More comments in the code instead of the text around it
- Highlights on the pieces that I am allowed to change as a user.

What would you remove from the artefact if possible?

- The created ‘model.’

Comparative evaluation artefact A & B

User interaction		
Upload a dataset	Artefact A	
Why?	It is super intuitive; it is click, click done!	
Create a subset	Artefact A	
Why?	It is super intuitive. It is click, click done!	
Workflow		Artefact B
Why?	I feel more in control; I can see better what is happening behind the scenes.	
Workflow explanation	Artefact A	
Why?	There is less text, and the workflow is more clear.	

Model construction		
Progress reporting on model construction	Artefact A	
Why?	The message is clearer than the progress bar	
Model construction		Artefact B
Why?	I can see better what is happening	

Model explanation		
Comparing results	Artefact A	
Why?	The overview is more clear	
Explanation of missing data handling	Artefact A	

Why?	But here I'm biased because I paid attention to it after first evaluating artefact B and your question about it.	
Readability of created model	Artefact A	
Why?	It is more readable, but still not enough for me to use it.	
Accuracy is a good measure of model performance		I agree it helps me to find out what to remove or retain in a model
I want to know statistical power		I agree I want to know at least the power and the confidence interval of models
I want to know variable importance		I agree this is what I care about most.

Discuss use-cases in which researcher-physicians think AutoML could contribute to their research.

Would you use AutoML in your research?
No, I cannot use this to build a model and compare the variable power of different variables in the different models that I create. It is hard to compare, and there is no why explanation.

If they do not want to use it, what would be needed to have them start using it?

Overall evaluation:

My perfect artefact would be a combination of both artefacts that I have just seen. For the basic stuff, I don't want to use code and clicking is just perfect. For the harder part, in which we create models, I prefer to see what is happening. Hence I'd like a hybrid version of these two approaches.

Interviewee 5

Individual evaluation artefact A (web-application)

User interaction			
Upload a dataset			**I like it**
Why?	I want to know which format it has to be. You should add that the format needs to be in .csv		
Create a subset			**I like it**
Why?	You could do something on the layout and on the explanation of what a target variable is etc		
Workflow			**I like it**
Why?	I liked it, but sometimes it was a lot of switching between screens.		

Workflow explanation	**I did not like it**		
Why?	A lot of information was missing in my opinion		

Model construction			
Progress reporting on model construction		**I am indifferent**	
Why?	I would like to see a progress bar or a start time, so I know when my 15 minutes are over		
Model construction			**I like it**
Why?	For exploration purposes, it seems useful		

Model explanation			
Comparing results			**I like it**
Why?	I miss the contribution of each variable to the model		
Explanation of missing data handling	**I did not like it**		
Why?	The letter size is too small. Furthermore, I want an extra explanation on what the consequences are for missing data handling		
Readability of created model	**I did not like it**		
Why?	The letter size is too small, and I do not know what everything means.		

What would you add to the artefact if possible?

- Larger letters to improve readability
- More explanations on where the buttons are, it is not very self-guiding
- Change target for class variable
- More explanations
- An explanation of what I can and cannot do when the artefact is working on something.
- An explanation on the model construction (evolutionary algorithms) at the results page

What would you remove from the artefact if possible?

- The restriction on logistic regression

Individual evaluation artefact B

User interaction			
Upload a dataset		**I am indifferent**	
Why?	I miss some explanation		
Create a subset			**I like it**
Why?	It is clear. I like the example		
Workflow		**I am indifferent**	
Why?	Different letter sizes between code and text would help me a lot.		
Workflow explanation		**I am indifferent**	
Why?	Maybe you could be clearer on when I have to edit code and when not. Another option could be to collapse the code blocks that I don't have to edit. Showing the code does not have value to me.		

Model construction			
Progress reporting on model construction			**I like it**
Why?	I liked the progress bar		
Model construction			**I like it**
Why?	It is more clear what is happening 'under the hood.'		

Model explanation			
Comparing results			**I like it**
Why?	It is a bit hard to read, but it's clear		
Explanation of missing data handling			**I like it**
Why?	I came across it very consciously		
Readability of created model	**I did not like it**		
Why?	It's just hard to read		

What would you add to the artefact if possible?

- A clearer explanation of how to run the artefacts
- Which code block I have to do something and which I did not
- Be more clear what the demonstrated output is
- Larger letter size

What would you remove from the artefact if possible?

Comparative evaluation artefact A & B

User interaction		
Upload a dataset		**Artefact B**
Why?	It gets me to think more about my data	
Create a subset		**Artefact B**
Why?	It provides me with more insights into what I am doing	
Workflow		**Artefact B**
Why?	This is more step-by-step	
Workflow explanation		**Artefact B**
Why?	The code makes the explanations more clear to me	

Model construction		
Progress reporting on model construction		**Artefact B**
Why?	The progress bar helps me a lot to see that something is actually happening	
Model construction		**Artefact B**
Why?	The visibility of the process is more clear	

Model explanation		
Comparing results	**Artefact A**	
Why?	The difference in model performance is easier to see at once	
Explanation of missing data handling		**Artefact B**
Why?	Better integrated into the process	
Readability of created model		**Artefact B**
Why?	The layout is more clear	
Accuracy is a good measure of model performance		**I agree** for explorative purposes. For other purposes, I'd like to know other statistics.
I want to know statistical power		**I agree,** to go beyond exploration with AutoML. Confidence intervals or effect sizes would be very nice to have
I want to know variable importance		**I agree**, what would be perfect if I could get a matrix with variable combinations

		that bring the best performance.

Discuss use-cases in which researcher-physicians think AutoML could contribute to their research.

- At this moment in time, with this stage of development. It would be only useful for explorative research and guidance in which model to pick. It could help broaden my vision from only logistic regression models to the usage of other models

Would you use AutoML in your research?

- Yes, for explorative research. To use it in publications, it needs to be more mature in what the end-user wants to see. Variables, statistical properties of the created model

Overview of artefact evaluations

These tables summarize the evaluations of the artefacts. First, we present numerical evaluations, later we demonstrate the textual evaluations of the artefacts. The values are:
1 = I dislike it
2 = I am indifferent
3 = I like it

Artefact A	S1	S2	S3	S4	S5	Average
User interaction						
Upload dataset	3	3	2	3	3	2,8
Create subset	3	3	1	3	3	2,6
Workflow	2	3	1	3	3	2,4
Workflow explanation	1	3	2	3	1	2
Model construction						
Progress	1	1	1	2	2	1,4
Model construction	2	2	1	3	3	2,2
Model explanation						
Compare results	2	1	1	1	3	1,6
Explanation missing data	2	1	1	2	1	1,4
Readability	1	1	1	1	1	1

Artefact B	S1	S2	S3	S4	S5	Average
User interaction						
Upload dataset	3	3	2	3	2	2,6
Create subset	3	2	3	3	3	2,8

Workflow	2	2	2	1	2	1,8
Workflow explanation	2	2	1	2	2	1,8
Model construction						
Progress	2	3	3	2	3	2,6
Model construction	3	3	3	3	3	3
Model explanation						
Compare results	1	1	1	1	3	1,4
Explanation missing data	1	1	1	1	3	1,4
Readability	1	1	1	1	1	1

Artefact A	S1	S2	S3	S4	S5	Majority
User interaction						
Upload dataset	Like	Like	Indifferent	Like	Like	Like
Create subset	Like	Like	Dislike	Like	Like	Like
Workflow	Indifferent	Like	Dislike	Like	Like	Like
Workflow explanation	Dislike	Like	Indifferent	Like	Dislike	Dislike
Model construction						
Progress	Dislike	Dislike	Dislike	Indifferent	Indifferent	Dislike
Model construction	Indifferent	Indifferent	Dislike	Like	Like	Indifferent
Model explanation						
Compare results	Indifferent	Dislike	Dislike	Dislike	Like	Dislike
Explanation missing data	Indifferent	Dislike	Dislike	Indifferent	Dislike	Dislike
Readability	Dislike	Dislike	Dislike	Dislike	Dislike	Dislike

Artefact B	S1	S2	S3	S4	S5	Majority
User interaction						
Upload dataset	Like	Like	Indifferent	Like	Indifferent	Like
Create a subset	Like	Indifferent	Like	Like	Like	Like
Workflow	Indifferent	Indifferent	Indifferent	Dislike	Indifferent	Indifferent

Workflow explanation	Indifferent	Indifferent	Dislike	Indifferent	Indifferent	Indifferent
Model construction						
Progress	Indifferent	Like	Like	Indifferent	Like	Like
Model construction	Like	Like	Like	Like	Like	Like
Model explanation						
Compare results	Dislike	Dislike	Dislike	Dislike	Like	Dislike
Explanation missing data	Dislike	Dislike	Dislike	Dislike	Like	Dislike
Readability	Dislike	Dislike	Dislike	Dislike	Dislike	Dislike

11.9 Appendix 9: Publication and blog for ICT& Health

This appendix features a two-page publication in Dutch in the magazine ICT&Health as well as a blog that was published on their website.

Publication

[chapeau]

Artsen willen inzicht in black box

[kop]

Onvoldoende transparantie struikelblok voor Automated Machine Learning

[intro]

In de gezondheidszorg is nog veel terrein te winnen op het gebied van gebruik en slim verwerken van data. Automated Machine Learning (AutoML) zou hieraan kunnen bijdragen door artsen en andere zorgprofessionals in staat te stellen met hun eigen data aan de slag te gaan. Technieken die waarde uit data halen, zijn via AutoML toegankelijk te maken voor een groter publiek. Deze self-service data-science kan datagedreven ontdekkingen en ontwikkelingen in de medische wereld versnellen. Een veelbelovende ontwikkeling, maar om het vertrouwen van artsen voor deze technologie te winnen, blijkt meer transparantie van de onderliggende algoritmiek nodig.

[platte tekst]

AutoML is te vergelijken met een zelfrijdende auto. Om gebruik te kunnen maken van Machine Learning heb je een opleiding plus kennis en kunde van de techniek nodig. Certificaten van cursussen over Machine Learning zijn te vergelijken met een rijbewijs voor een normale auto. Je hebt een rijbewijs nodig om zelfstandig in een auto te mogen stappen, als bewijs van jouw kennis, kunde en ervaring over het besturen van een auto.

Bij het instappen van een zelfrijdende auto hoeft dit niet, net als dat je geen rijbewijs nodig hebt om AutoML te gebruiken. Wanneer je in een zelfrijdende auto stapt, geef je deze jouw bestemming op en eventueel het type route, dat je wilt rijden: een route met mooie uitzichten of de snelste. Met AutoML verloopt dit bijna hetzelfde. Het enige dat je extra moet leveren, is de kaart (dataset). Verder geef je aan van welke variabele je de waardes wilt voorspellen (adres) en op welke manier de voorspelling geoptimaliseerd moet worden (type route).

[tk]

Zelfrijdende functies

Door de zelfrijdende functies is AutoML heel interessant voor de zorg. Op basis van de gegeven dataset, het doel en type route kiest de AutoML-methode zelf welk van de beschikbare algoritmes het beste resultaat oplevert voor jouw dataset en jouw doel. Daarnaast wordt het gekozen algoritme ook gefinetuned voor de optimale performance, en dat alles zonder menselijke tussenkomst.

Vanuit technisch oogpunt lijkt AutoML een perfecte basis om self-service data-science mee op te zetten. Zonder tussenkomst van een expert kan een arts zelf aan de slag met zijn eigen data om nieuwe inzichten te verkrijgen.

Deze belofte enthousiasmeerde een aantal arts(-onderzoekers) om mee te doen aan het afstudeeronderzoek ‘AutoML voor self-service data-science in de gezondheidszorg’, dat in de periode maart-september 2019 is uitgevoerd in drie topklinische ziekenhuizen met vijf arts-onderzoekers.

[tk]
Alleen regressiemethoden

Uit interviews met de deelnemende artsen kwamen de eisen naar voren, die zij stellen aan AutoML. Zo zijn ze voor onderzoek en nieuwe ontdekkingen niet geïnteresseerd in het gebruik van meerdere types algoritmen. In medisch onderzoek wordt vrijwel alleen gebruik gemaakt van regressiemethoden. AutoML heeft echter meer in zijn mars dan alleen regressie. De eerste stap is de keuze voor het type algoritme: regressie of een ander soort algoritme.

Als reden voor het gebruik van uitsluitend regressie noemden de deelnemende artsen dat ze willen kunnen uitleggen waarom bepaalde keuzes gemaakt worden. Waarom welke variabelen de meeste invloed hebben op een voorspelling, is voor hen interessanter dan het verbeteren van een score voor medisch onderzoek. De variabelen en hun gewicht kunnen gebruikt worden in het zorgproces.

Naast de uitleg van het resultaat vroegen de artsen ook naar het proces van het samenstellen van het model. Ze willen graag weten waarom een methode juist voor dit algoritme en deze variabelen had gekozen.

De deelnemende artsen stelden dat een black box-methode zonder uitleg niet gebruikt gaat worden. Ze willen precies weten hoe de vork in de steel zit. De boodschap voor de onderzoeker was dan ook: faciliteer het gebruik van regressies en andere transparante algoritmen voor onderzoek om transparantie in zowel het proces als de uitkomst van het onderzoek te ondersteunen.

[tk]
Transparantie ontbreekt

Omdat AutoML-methoden alleen te gebruiken zijn tijdens het programmeren in Python, moest hier nog een ‘laagje’ overheen ontwikkeld worden, zodat de artsen niet hoeven te programmeren. Tijdens het creëren van dit laagje bleek dat het belangrijkste onderdeel, de transparantie, niet geleverd werd door de AutoML-methoden.

Waar je bij een zelfrijdende auto instapt en het landschap langs je ziet gaan, was dat hier niet het geval. Je voert een kaart, adres en type route in en na een tijdje rekenen zonder updates krijg je het resultaat. Je kunt niet meekijken tijdens het proces en krijgt alleen de score en een technische beschrijving van het model als uitkomst. Over de gebruikte variabelen kom je niets te weten.

Na deze ‘zelfrijdende auto’ getest te hebben met de artsen werd bevestigd dat de techniek, in haar huidige staat, niet de sleutel is tot self-service data-science voor artsen. Zij geven aan inzicht in en controle over het process te willen hebben.

[tk]
Ervaringen delen

Tijdens dit onderzoek naar AutoML voor de zorg is de sleutel voor self-service data-science voor artsen nog niet gevonden. Ondanks deze uitkomst is het van belang de opgedane ervaring te delen. Over AutoML of Machine Learning in de gezondheidszorg zijn tot nu toe nagenoeg alleen positieve ervaringen gepubliceerd. Volgens de wetenschappelijke literatuur en nieuwsartikelen leveren ze alleen maar succes en vooruitgang op.

Er moet echter ook aandacht zijn voor technieken en oplossingen, die nog niet volwassen genoeg zijn om te gebruiken in de praktijk. Zo kan worden voorkomen dat het wiel opnieuw wordt uitgevonden en een volgende student, promovendus of start-up AutoML in haar huidige staat gaat testen of inzetten bij andere ziekenhuizen. Hun tijd, geld en energie kan beter gebruikt worden om deze technologie te verbeteren, zodat artsen via AutoML op afzienbare termijn wel in staat worden gesteld om gemakkelijk met hun vragen over eigen data aan de slag te gaan. Het oprichten van een platform voor self-service data-science zou aan deze ontwikkeling kunnen bijdragen.

[tk]

Vooroordelen wegnemen

Juist in de zorg is transparantie over processen erg belangrijk. Overal in de samenleving worden zorgen geuit over vooroordelen in algoritmen. Voor toekomstige onderzoekers en entrepreneurs ligt de uitdaging in het creëren van transparantie van het proces en het opbouwen van vertrouwen in de black box oplossingen. Artsen rijden immers ook in auto’s, zonder dat ze exact weten wat deze doen. De uitdaging is om hen achter het stuur te krijgen van de black box methoden om op basis van positieve ervaringen vertrouwen te creëren.

===tekst kader/CV===

Richard Ooms is master student Business Informatics aan de Universiteit Utrecht, met een Applied Data Science profiel. Door de ziekte van zijn vader is hij gemotiveerd om de gezondheidszorg te verbeteren vanuit zijn eigen vakgebied. Zijn masterscriptie, onder supervisie van Dr. Marco Spruit, is het resultaat van een scriptiestage bij het Analytics & Cognitive team van Deloitte.

===einde tekst kader/CV===

===tekst kader/CV===

Dr. Fenna Heyning is directeur van STZ, de vereniging van samenwerkende topklinische opleidingsziekenhuizen en redactieraadlid van ICT&health. Opgeleid als internist-hematoloog heeft de rol van cultuur en gedrag bij implementatie van innovatie in de zorg haar grote interesse.

===einde tekst kader/CV===

Blog voor ICT & Health - https://www.icthealth.nl/blog/

Medisch-specialisten op de werkvloer over machine learning: “een black box methode zonder uitleg gaat niet gebruikt worden, wij willen weten hoe de vork in de steel zit”

Vraagt u zich ook wel eens af wanneer Artificieel Intelligence (AI) en black-box algoritmen hun intrede gaan maken in uw spreekkamer? Wat zou u doen als een algoritme u een behandeling of advies voor een patiënt aanraadt wat tegen uw gevoel in gaat? Wellicht nog interessanter, wat zou u ervan vinden om zelf met verschillende black-box algoritmen op uw eigen data aan de slag te gaan met een zelf rijdende auto voor machine learning?

De belofte van een zelfrijdende auto voor machine learning enthousiasmeerde een aantal artsen en arts-onderzoekers om mee te doen aan mijn scriptieonderzoek voor mijn masters in business informatics. Als eerste stap ben ik artsen gaan bevragen over hun wensen op de werkvloer voor dit soort technieken. Na een paar interviews werd me een aantal dingen duidelijk over hun eisen. Ten eerste zijn de artsen voor onderzoek en nieuwe ontdekkingen niet geïnteresseerd in het gebruik van meerdere type algoritmes, in medisch onderzoek wordt vrijwel alleen gebruik gemaakt van transparante regressie methoden. Een black box methode zonder uitleg gaat niet gebruikt worden, want artsen en onderzoekers willen weten hoe de vork in de steel zit.

Toch denk ik dat andere methoden dan regressie ook grote meerwaarde kunnen hebben voor het onderzoek van artsen. In gesprekken hierover met Fenna Heyning kwam naar voren dat artsen veel minder bekend zijn met andere methodes dan regressie. Van oudsher is men klassieke statistische methodes geleerd, maar inmiddels is er veel meer mogelijk. Door het gebruik van andere algoritmen dan regressie is het mogelijk om tot nieuwe inzichten en betere voorspellingen te komen met dezelfde data. Het per definitie afwijzen van alles wat geen regressie is, kost meer dan het oplevert. Andere algoritmen kunnen nieuwe invalshoeken en betere resultaten opleveren: het in gebruik nemen van een nieuw medicijn kan minder complicaties en meer succesvolle trajecten met patiënten opleveren. Echter moet iemand ze wel ooit in gebruik durven te nemen. Een ander voordeel van het gebruik van nieuwe algoritmen is dat trials ook sneller en flexibeler van opzet worden, met resultaten die meer directe relevantie hebben voor de patiënt en diens behandelaar.

Ondanks dat de stap onwennig zou kunnen zijn pleiten wij ervoor om adoptie van nieuwe algoritmen in onderzoek in de zorg te stimuleren. Zolang dit zorgvuldig gebeurt, kan dit veel nieuwe en relevante inzichten opleveren voor patiënt en behandelaar. Belangrijk hierbij is het delen als een methode of techniek niet werkt, alleen op die manier kunnen we van elkaars fouten leren en stappen vooruit zetten.

Part of an article in ‘Het Financiele dagblad’

Richard Ooms, student business informatics in Utrecht, zocht medewerking van artsen voor een project met 'automated machine learning'. Dat is een speciale vorm van kunstmatige intelligentie. De computer kiest dan zelf welke statistische methode het beste werkt bij bepaalde data. Artsen uit drie verschillende ziekenhuizen, aangesloten bij de koepel voor 'topklinische' ziekenhuizen STZ, deden mee. Ooms vroeg de artsen hoe AI hun werk zou kunnen verbeteren. Hun antwoord: als we zelf kunnen uitleggen waarom de computer een conclusie trekt. 'Ze wilden controle houden', was zijn conclusie. 'Voor een black box hadden ze geen belangstelling.' Bovendien waren ze alleen nieuwsgierig naar één statistische methode. Er was geen behoefte aan AI die zelf een methode kiest. 'Een enorme beperking', zegt de jonge onderzoeker. 'Hiermee kon de techniek niet optimaal worden benut.' Hij ging aan de slag met een data van diabetici. In Nederland kon hij ze niet krijgen. Zo kwam hij onder meer uit bij de Pima, een indianenstam uit de Amerikaanse staat Arizona. Deze mensen stapten ooit over van een traditioneel dieet naar voedsel uit de Amerikaanse supermarkt, en kregen sindsdien schrikbarend vaak suikerziekte. Hun data is vrij beschikbaar. De artsen vroegen een exacte voorspelling voor het krijgen van diabetes bij allerlei factoren, zoals bloeddruk, dikte van de huid, glucose niveau, bmi of zwangerschap. Zo ver bleek de techniek nog niet te zijn. Maar het onderzoek leidde tot een conclusie die minstens zo belangrijk is, zegt Ooms: 'We weten nu dat artsen zich pas vertrouwd voelen met kunstmatige intelligentie, als ze volledige controle hebben. De psychologie is even belangrijk als de techniek.'

11.10 Appendix 10: Draft of Scientific paper

The next pages contain a draft of a scientific paper. We aim to submit this paper for the SIGKDD 2020 conference on the call for applied data science papers.

Self-Service Data Science in Healthcare

Using AutoML in the knowledge discovery process

Richard Ooms
Department of Information and Computing Sciences
Utrecht University
Utrecht, The Netherlands
r.l.j.ooms@students.uu.nl

Marco Spruit
Department of Information and Computing Sciences
Utrecht University
Utrecht, The Netherlands
m.r.spruit@uu.nl

ABSTRACT

Introduction: The healthcare industry has been lagging in the adoption of analytics. One of the reasons for lagging is the shortage of data scientists in the healthcare sector. Advancements in Machine Learning (ML) and research on its accessibility for non-experts sparked the research field of Automated Machine Learning (AutoML). Because AutoML is designed to make ML accessible to non-expert users, this research aims to find out how researcher-physicians can be supported in their knowledge discovery process by applying AutoML as part of the research field of Applied Data Science (ADS). This is the first study, to the best of our knowledge, to test AutoML methods with domain experts in the healthcare domain.

Method: The method used in this research is design science. First, we selected TPOT as AutoML method based on the results of a benchmark test and requirements from researcher-physicians. We integrated TPOT into two artefacts, a web-application and a notebook. We have evaluated the artefacts with the framework for evaluation in design science to find out which method suits researcher-physicians best.

Results: The benchmark test found that there was no AutoML method that consistently outperformed all other methods one-hour and four-hour budgets. However, TPOT and Auto-Sklearn performed best on both tests. As TPOT was the method that satisfied most requirements, we integrated TPOT into two artefacts. Both artefacts had a similar workflow, but different user interfaces because of a conflict in requirements. Artefact A, a web-application, was perceived better for uploading a dataset and comparing results. Artefact B, a Jupiter notebook, was perceived better regarding the workflow and being in control of model construction. Thus, a hybrid artefact would be best for researcher-physicians. However, both artefacts missed model explainability and an explanation of variable importance for the created model. Hence, the researcher-physicians indicated that they would only use AutoML for the explorative phase of their knowledge discovery process.

Discussion: The results suggest that AutoML methods need work on explaining the created models and their route to model creation. Another issue is the stability of the (Auto)ML models; the models created by an evolutionary algorithm based AutoML methods are hard to reproduce due to their random inception. As much as changing the seed can change the outcome for a single patient

KEYWORDS

Automated Machine Learning, Applied Data Science, Healthcare Analytics, Benchmark test, Researcher-physicians

1 Introduction

In the Netherlands, 10% of the GDP is spent on healthcare (OECD, 2019). With an ageing population, this spending is expected to double in 2040 [1]. With the growing burden of healthcare costs on society, it is vital to improve efficiency and reduce the costs of healthcare. Improvements can be in the supply chain of hospitals [2], development of personalised care plans to improve quality and experience of patients [3] and improve operational efficiency [2]. This research aims to catalyse the adoption of analytics in healthcare by finding out how we can support the knowledge discovery process of domain experts with AutoML in the field of Applied Data Science (ADS) as defined by Spruit and Jagesar [4, p. 1]: *"The knowledge discovery process in which analytical applications are designed and evaluated to improve the daily practices of domain experts."*

Although there is enormous potential in analytics, the healthcare sector has been slow in adopting it in their daily practice compared to other industries [5]. Because of the late adoption of analytics, the healthcare industry is lagging compared to other industries considering analytics. When asked, a medical researcher stated the following about the state of analytics in healthcare: *"I seriously believe that we are in the middle ages. I look at my iPhone and think about everything that's possible and yet here in the hospital, you still get a piece of paper with your appointment."*[6, p. 20]. It becomes evident that there is a lot of progress to be made in the application of analytics in healthcare [3], [7]. Furthermore, there is a shortage of data scientists in healthcare [8]–[11]. This ever-growing shortage of data scientists hinders the adoption and development of analytics in the healthcare sector [12]. To improve the adoption of healthcare analytics, one of the focus areas in healthcare research should be making analytics accessible to domain experts [3]. Automation of the knowledge discovery process can increase the adoption of analytics by enabling domain experts to contribute to the knowledge discovery in the field using state-of the art techniques. Enabling domain experts to perform analytics is referred to as ADS [4].

The Machine Learning (ML) community has also noticed the need to enable access for non-expert users to ML techniques. The need to enable non-experts to use machine learning is one of the drivers that gave birth to the fast-paced research area of AutoML [13], [14]. The AutoML community aims to automate all steps in the process of creating a machine learning pipeline. However, to the best of our knowledge, no AutoML applications were tested in real-world situations with non-expert users in the healthcare domain. Hence our research question is: *How can we support the knowledge discovery process of domain experts in healthcare using AutoML?* CRISP-DM [15] is considered to be the knowledge discovery process, when referred to in this paper.

This research aims to explore and overcome the boundaries to AutoML adoption in healthcare with a method-agnostic approach as it is the first study, to the best of our knowledge, to assess adoption of AutoML methods by domain experts in the healthcare domain. In the remainder of this paper we first discuss

2 Overview of AutoML methods

In this section, we summarise and categorise the AutoML methods that we found during a literature review. To do so we first define Machine Learning (ML): *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."* [16, p. 2] and AutoML: *"AutoML attempts to construct machine learning programs (specified by E, T and P), without human assistance and within limited computational budgets"*[17]. Most papers about AutoML describe the construction of a ML pipeline. This pipeline is an analogy for the process through which the data progresses during data analysis. A pipeline consists of data collection, data pre-procession, and analytical processing.

In our literature review we only consider the newest versions of the methods and only include non-commercial AutoML methods. First, we compare two AutoML methods that are developed for the healthcare domain. Second, we discuss methods with a fixed pipeline length. Third, we discuss the AutoML methods that build neural nets. Fourth, we discuss evolutionary methods. Fifth, we discuss distributed methods. Finally, we provide a detailed overview of the AutoML methods discussed in this section.

2.1 Healthcare

FLASH [18] and AutoPrognosis [19] have both been developed for healthcare or with funds for healthcare, but with different incentives. FLASH was developed to improve the efficiency of creating and evaluating pipelines. AutoPrognosis is developed with the practitioner in mind. FLASH is a black-box tool, as most AutoML tools are. In contrast, AutoPrognosis is the only AutoML method that contains an explainer to justify its recommendations to a clinician.

We cannot compare the performances of both methods, as there is no benchmark test available featuring both methods at the time of writing. FLASH tested its performance on a medical dataset with the binary classification task of predicting drug non-responders. In this case, it outperformed other methods based on TPE and SMAC using error rate as the performance metric [18]. AutoPrognosis outperformed Auto-WEKA, Auto-Sklearn, and TPOT on multiple datasets in its own comparison with other methods [19].

2.2 Fixed pipelines

Auto-WEKA [20], Hyperopt-Sklearn [21], Auto-Sklearn [22], PoSH Auto-Sklearn [23], and ML-Plan [24] are all methods that have a fixed pipeline length. PoSH Auto-Sklearn outperforms all other methods as it is the winner of the latest AutoML competition at the time of writing [14, Ch. 10]. Auto-WEKA, Hyperopt-Sklearn, and Auto-Sklearn were the first three methods that were developed to tackle the CASH problem. What is interesting to see is that Auto-Sklearn has served as a basis for multiple other AutoML systems, whereas the other two methods have not. We assume that this is due to the warm-start procedure that is built into Auto-Sklearn.

2.3 Neural Networks

The first version of AutoNet [25] was the first to automate the configuration of a Neural Network. It laid the groundwork for its successor and the inception of AlphaD3M [26] and Auto-Keras [27]. Besides laying the groundwork for these applications, it incentivised the inception of a lot of commercial applications. Most commercial applications that automatically tune Neural Networks are inspired by the first version of Auto-Net [14]. This is because Auto-Net was the first AutoML program to beat human experts in configuring a pipeline [14], [25].

AlphaD3M is the only AutoML method that makes use of reinforcement learning and is much faster than any other method in the field. In one case, it was 32 times faster than TPOT. However, AlphaD3M does not outperform other AutoML methods. In comparison with three other methods, its average rank is third, based on mean scores. As AlphaD3M ranks first on some datasets, it is still competitive [26]. It is interesting to see if reinforcement learning gets widely adopted as a search strategy for pipeline configuration.

2.4 Evolutionary methods

Evolutionary methods can create pipelines of flexible length. These are TPOT [28], LTPOT [29], RECIPE [30], and Autostacker [31]. They can do so due to their search strategy. The downside of evolutionary algorithms is that they can produce invalid pipelines and get stuck at local optima. RECIPE and LTPOT have independently overcome these downsides. RECIPE uses grammar to overcome this whereas LTPOT uses a maximum evaluation time for a pipeline.

2.5 Distributed methods

Two AutoML methods can process data in a distributed matter: Autostacker [31] and ATM [32]. It is remarkable, that there are only two systems that can run in a distributed manner, when taking the computing cost of creating a pipeline in mind. Autostacker can use parallel processing as it proposes the best pipelines to its user. Hence it needs the performance scores of the pipelines. ATM is the only method which can run in parallel on different machines and is set up to be distributed and scalable. The development of ATM and integration with the methods described above is one of the most attractive developments in AutoML. An incentive to spur this development could be having a separate performance challenge for distributed AutoML methods.

2.6 Overview of methods

Figure 1 demonstrates the relations between AutoML methods discussed in this section. It makes a distinction between methods that build NNs and methods that use traditional classifiers or regressors in their pipeline. Arrows between methods point out an source relationship between methods. The colors of the methods indicate the search strategy that is applied in a method to create a pipeline. A tabular overview of the discussed AutoML methods is in Table 4, including, the prediction tasks, a link to the code repository and information about the pipeline creation.

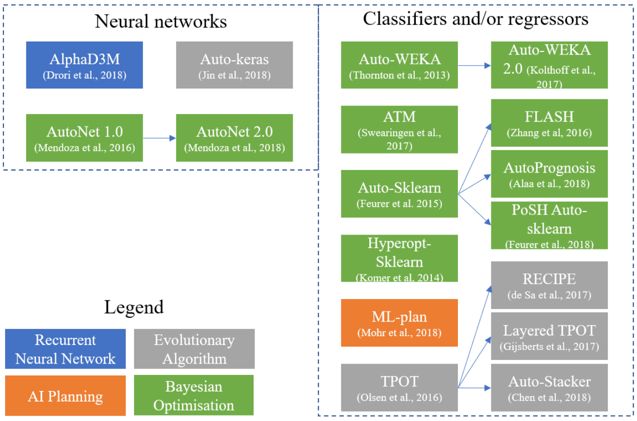

Neural networks
AlphaD3M (Drori et al., 2018)
Auto-keras (Jin et al., 2018)
AutoNet 1.0 (Mendoza et al., 2016)
AutoNet 2.0 (Mendoza et al., 2018)
Classifiers and/or regressors
Auto-WEKA (Thornton et al., 2013)
Auto-WEKA 2.0 (Kolthoff et al., 2017)
ATM (Swearingen et al., 2017)
FLASH (Zhang et al, 2016)
AutoPrognosis (Alaa et al., 2018)
Auto-Sklearn (Feurer et al. 2015)
PoSH Auto-sklearn (Feurer et al., 2018)
Hyperopt-Sklearn (Komer et al, 2014)
RECIPE (de Sa et al., 2017)
ML-plan (Mohr et al., 2018)
Layered TPOT (Gijsberts et al., 2017)
TPOT (Olsen et al., 2016)
Auto-Stacker (Chen et al., 2018)
Legend
Recurrent Neural Network
Evolutionary Algorithm
AI Planning
Bayesian Optimisation

Figure 35: Overview of AutoML methods

3 Method

We used the Design Science research framework [33] to answer the research question. To do so, we first performed a benchmark test using the framework of Gijsbers et al. (2019) on all available medical datasets from the OpenML100 [35] to find out if one AutoML method performed best on medical tasks. We ran medical binary-classification tasks on four datasets: breast cancer [36], diabetes [37], Indian Liver Patient [37] and sick dataset [38]. All selected AutoML methods received a time budget of one hour in a 10-fold cross-validation set-up to create the best pipeline on the given datasets. The time limit is set on one hour, as longer runs do not significantly provide better results [34]. To baseline the performance of the AutoML methods in the benchmark test, we added a decision tree and a constant predictor. Following Gijsbers et al. [34], we used Area Under the Receiver Operator Curve (AUROC) for scoring. All tests have been run on Amazon Web Services using m5.2xlarge machine[9], to get constant circumstances and enough computing power for the AutoML methods.

To understand the needs of domain experts we elicited requirements using semi-structured interviews. We selected semi-structured interviews as the best method for requirements elicitation for three reasons. 1) Semi-structured interviews are considered to be the most effective way for requirements elicitation [39]; 2) It is an accepted method for conducting qualitative research in healthcare [40]; 3) Semi-structured interviews have the benefits of eliciting people's own views and uncovering issues or concerns that have not been considered beforehand by the researcher [41]. For our interviews, we constructed an interview protocol following the guidelines for interview research [42]. The sample consisted of five domain experts who were active in the scientific department of a regional hospital in the Netherlands. The interviewees have decided to participate voluntarily and hold different roles and medical expertise within non-academic hospitals in the Netherlands. The interviewees are active in the research fields of cancer, orthopedy, and cardiology and participate in medical research, either full-time or part-time. The sample consists of three women and two men. We evaluated the capabilities of the AutoML methods to the requirements of the domain experts.

Based on the requirements we created two artefacts and evaluated these with the interviewees. The first artefact had a graphical user-interface (GUI), the second artefact had a code-based interface. We used artificial summative evaluation as part of the framework for evaluating design science research [43]. We evaluated the artefacts on the user-story categories from the previous research question. To be able to evaluate the artefact properties, we used refined hypotheses [44]. Half of the subject received the GUI artefact first and half of the subject received the code-based artefact first to prevent learning bias. The sample consisted of five researcher-physicians based in three different hospitals in the Netherlands. All interviewees had different specialities.

3.1 Validity

In this section, we address the "subjective" nature of the data collection and analysis of this research [45]. To assess the validity of this study, we look at three of the five aspects of validity for qualitative research, as proposed by Burke Johnson [46]. We do not discuss theoretical validity as the goal of design science research is on artefact creation instead of theory creation. We do not aim to explain a phenomenon. We also do not discuss Internal validity, as we do not aim to answer a question about a causal phenomenon. The main research question of this research is a 'how'-question.

Descriptive validity

Descriptive validity is on the factual accuracy of the account of events as reported by the researcher [46]. As a single researcher has conducted this research, hence there is a bias in data collection. A researcher is subjective by nature, and so are his data collection and analysis [45]. Besides that, the relationship between the researcher and participants significantly influences what the participants reveal to the researcher [47, pp. 11–30]. To mitigate this validity threat, we used a framework to set up the semi-structured interviews and the framework for evaluation of design science to set up the artefact evaluation. We recorded all interactions with the participants, and we took part in sessions to obtain peer-feedback on our research to increase the descriptive validity.

Interpretive validity

Interpretive validity is about accurately portraying the meaning that was attached by the participants to the objects that were studied [46]. To mitigate this threat, we have sent the elicited user-stories to the participants to obtain feedback on our findings [45]. Furthermore, we used data-triangulation by tapping into other sources to confirm our findings. To make sure we portrayed the meaning of the participants well, we used low inference descriptors by quoting participants in this research [46].

External validity

[9] 32 GB memory, 8 vCPUs (Intel Xeon Platinum 8000 series Skylake-SP processor with a sustained all core Turbo CPU clock speed of up to 3.1 GHz). OS used is Amazon Linux. https://aws.amazon.com/ec2/instance-types/m5/

External validity is crucial if we want to generalise our findings to a larger part of the population [46]. Although generalizability is not the primary purpose of this research, we will touch upon the subject. As this research conducts a case-study, the best way to generalise its findings is to find the similarity in subjects, objects and issues [48]. As the sample size and characteristics are not valid for generalizability, the best method to generalise our findings is to identify similarity in other situations.

Characteristics that make the sample inapplicable for generalisation to medical professionals are the limited set of medical domains in which the participants operate and the fact that all participants decided to participate voluntarily. Hence, insights derived from this study are hard to generalise but could be a stepping stone for future research.

4 Benchmark test

The benchmark test was ran using a time budget of one hour with a total of 160 hours of computational time. Figure 25 contains the visualisation of results; the datasets are on the x-axis, the y-axis shows the AUROC score. A coloured dot marks the score for an AutoML method on each of the ten folds. A Kruskal-Wallis test indicated that there was a statistically significant difference in the distribution for the Breast (*H* =13.10, *p* <.001), Diabetes (*H* =31.10, *p* <.001), Liver (*H* =31.51, *p* <.001) and Sick dataset (*H* =28.93, *p* <.001) between the AutoML methods, see Table 5 for statistics.

What is interesting to see in Figure 25 is that on the liver dataset, the decision tree and Hyperopt-Sklearn do not always outperform the constant predictor. On the diabetes dataset, Hyperopt-Sklearn lags behind the three other methods, but performs better than the constant predictor and has a similar performance to the decision tree. On the breast dataset, all AutoML methods have the maximum score in at least one fold. All methods perform well on the breast set, given their median scores and distribution. The performance of the decision tree indicates that it is not a hard prediction problem. For the results on the sick dataset, we see that TPOT and Auto-Sklearn outperform the other two methods in both consistency and score of their predictions despite the fact that the set has missing values. Hyperopt-Sklearn is again not better than the decision tree.

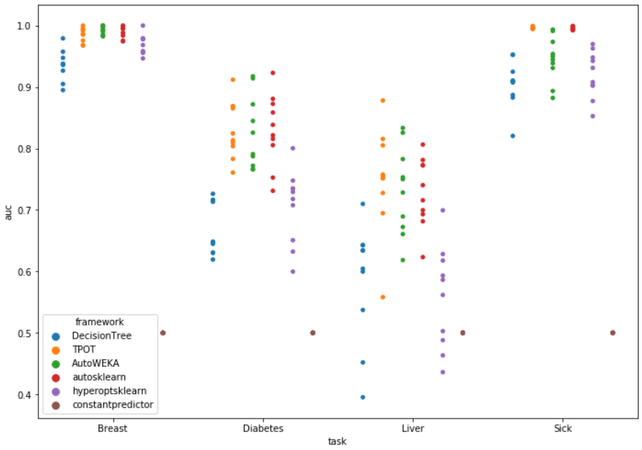


Figure 36: One-hour benchmark test results

Overall, TPOT registered the highest median score after running for one-hour on all sets but the Breast dataset, in which Auto-Sklearn registered the best performance. A Mann-Whitney U test indicated that Auto-Sklearn significantly outperforms the decision tree (*U* =2.0, *p* <.001) and Hyperopt-Sklearn (*U* =17.5, *p* <.01) on the Breast set. TPOT significantly outperforms the decision tree on the datasets Diabetes (U =0.0, *p* <.001), Liver (*U* =8.0, *p* <.001) and Sick (*U* =0.0, *p* <.001). Furthermore, TPOT outperforms Hyperopt-Sklearn significantly on the datasets Diabetes (U = 2.0, *p* <.001), Liver (*U* =7.0, *p* <.001) and Sick (*U* =0.0, *p* <.001). Finally, Auto-WEKA is significantly outperformed on the Sick dataset by TPOT (*U* =0.0, *p* <.001). This is probably due to the fact that Auto-WEKA does not impute data for missing values. Auto-Sklearn and TPOT do impute values for missing data and have significantly better results. The statistics for the Mann-Whitney U test are available in Table 21. The table shows the p-values and U statistics for each method compared to the best performing method on each dataset. TPOT and Auto-Sklearn do not significantly differ in performance for any of the datasets.

Table 20: Statistics for Kruskal Wallis test

	Breast	Diabetes	Liver	Sick
H-statistic	11.36	18.64	17.93	27.87
P-value	0.995**	0.324**	0.455**	0.386**

** p < 0.001

Table 21: P-values and U statistic for Mann Whitney U test compared to best performer

	Breast		Diabetes		Liver		Sick	
	p	U	p	U	p	U	p	U
Decision Tree	**0.164****	**2.0**	**0.908****	**0.0**	**0.895****	**8.0**	**0.913****	**0.0**
TPOT	0.236	40.0						
Auto-WEKA	0.395	46.0	0.455	48.0	0.263	40.0	**0.913****	**0.0**
Auto-Sklearn			0.425	47.0	0.213	39.0	0.5	49.5
Hyperopt-Sklearn	**0.8***	**17.5**	**0.164****	**2.0**	**0.657****	**7.0**	**0.913****	**0.0**

* p < 0.01 ** p < 0.001

From the benchmark test we can conclude that no method consistently outperforms all others. However, we see that TPOT records the highest median scores on three of the four tasks in this test but this performance is not significantly better than the performance of other methods. Auto-Sklearn and Auto-WEKA get similar results to TPOT, only Auto-WEKA is outperformed on the Sick dataset.

5 AutoML requirements evaluation

We elicited twenty-one requirements of domain-experts in the form of user-stories. These user-stories are categorized in four categories: User interaction, model construction, model usage and model explanation. We only consider the AutoML methods that participated in the benchmark test for comparison with user-stories. Furthermore, we only include user-stories that were mentioned by at least two participants, leaving us with fifteen user-stories. For user-stories, we use the template of Cohen (2004): *“As a 〈type of user〉 , I want 〈goal〉, [so that 〈some reason〉].”*. Examples of the user-stories are: ‘*As a researcher-physician, I want to know how a prediction mechanism works, so that I can trust it more easily.*’ and *‘As a researcher-physician, I want to transfer my model into a calculation tool, so that it can be used in clinical practice.’.*

There are two user-stories in the user interaction category with the AutoML artefact. These are in conflict, three out of five domain-experts prefer to use code to do their analysis, so they have more control over what is happening. In contrast, two out of five domain-experts prefer to have a GUI, as small coding errors cost a lot of time to solve. *“*quotes removed for confidentiality.*

For model construction there are five user-stories. Automatic model configuration as well as the possibility to set a time budget is covered by the concept of AutoML. The restriction of only using logistic regression can only be satisfied by TPOT and Auto-WEKA. Auto-Sklearn and TPOT are the methods that satisfy the requirement of an explanation on handling missing data. There is only one user-story about using the created model in practice. Using the created model on unseen data is supported by all assessed AutoML methods.

*quotes removed for confidentiality*

Almost half of the user stories are about model explanation, the domain-experts are interested in what variables are important, what the statistical power of a model is and what decisions are made during model creation. Model explanation was not only the category with the most user-stories, but also However, none of the assessed AutoML methods could satisfy any of these requirements. It is possible to integrate comparison of created models and an explanation of regression in the artefacts. **quotes removed for confidentiality**

All practitioners have discussed the necessity of getting to use their models in practice: **quotes removed for confidentiality**

After comparing the four AutoML methods to the user stories, we can conclude that TPOT is the best AutoML method for this set of requirements. TPOT satisfies five of the fifteen assessed requirements compared to four out of fifteen by Auto-WEKA and Auto-Sklearn. What is interesting to note is the inability of all AutoML methods to explain the created models. The need for explainability is evident: Model explanation is the biggest category in the user-story categorisation. Besides that, explaining model decisions is obligatory in Europe since the introduction of the General Data Protection Regulation Law [50, pp. 40–42]. Table 17 contains an overview of the AutoML methods in each category. We have not included the user-interaction category into the table as it contains conflicting user-stories and does not apply to AutoML methods, only to the artefacts. As TPOT performs better than Auto-Sklearn on the requirements we integrated TPOT in the artefacts.

Table 22: AutoML method scores on user-story categories

Category/method	Auto Sklearn	Auto-WEKA	TPOT	Hyper-opt-Sklearn
User interaction	n/a	n/a	n/a	n/a
Model construction	3	3	4	2
Model explanation	0	0	0	0
Model usage	1	1	1	1
Total matches	**4**	**4**	**5**	**3**

6 Artefact evaluation

Based on the user-stories we created two artefacts to automate a part of the data preparation phase and the complete modelling phase of CRISP-DM [15]. The data preparation activities involve the possibility to in- or exclude variables, data imputation for missing values and the recoding of categorical variables to numerical variables, as TPOT cannot handle non-numerical data as input. Due to the conflict in user-interaction requirements, we designed two artefacts with the same functionalities, but a different user-interface. The artefacts can be used to create logistic regression models and users who do not know how to code should be able to use the artefacts. The artefacts contain a description of missing data handling, as well as the possibility to compare the different models. The two artefacts are a Flask web-application (GUI artefact) to satisfy the graphical user interface preference and a Jupyter notebook (code artefact) to satisfy the coding preference.

Using the risk and efficacy strategy from the framework for evaluation in design science [43]. We created a set of refined hypotheses based on the user-story categories to evaluate the artefacts with the domain-experts. For the user-interaction we tested on four elements: Uploading a dataset, creating a subset, the workflow of the application and the workflow explanation. Only for uploading a dataset the GUI artefact was preferred. For all other actions the code artefact was preferred, as the interviewees felt more in control of the process. The same was the case for model construction and the progress reporting on model construction as parts of the model construction category. The model explanation category consisted of comparing of different results, explanation of missing data handling and readability. In this category artefact A was preferred over artefact B for all interactions except for explanation of missing data handling. An overview of the preferences and categories is available in Table 23.

Table 23: Artefact preferences

Category	Preference	Score
User interaction		
Upload dataset	GUI	4/5
Create a subset	Code	3/5
Workflow	Code	4/5
Workflow explanation	Code	4/5
Model construction		
Progress reporting	Code	4/5
Model construction	Code	5/5
Model explanation		
Compare results	GUI	4/5
Explanation missing data	Code	4/5
Readability	GUI	4/5

What we found is that a hybrid version of the two artefacts is preferred to interact with AutoML by the interviewees. To keep control over the process of creating a model they prefer coding: **quotes removed for confidentiality**

For the relatively simple tasks such as uploading a set or comparing results, they prefer a graphical user interface: **quotes removed for confidentiality**

Whereas for model construction the code-based interface is preferred: *"*Furthermore, the models produced by TPOT need to explain variable importance to make it usable in research practice as it is considered a must-have for the adoption of AutoML by domain-experts. Most interviewees do not consider the output of TPOT as a model: **quotes removed for confidentiality**

After the artefact evaluation, we asked the interviewees if they would use AutoML in their research. Two of the interviewees mentioned that they would find it useful for data exploration and performance comparison of their models. Another interviewee mentioned that it believed in the power of (Auto)ML, but for adoption, the methods need to improve on explainability and transparency. The last two interviewees only would use AutoML in their research if the variable importance is part of the output of the AutoML method.

7 Conclusion

The main research question of this study was '*How can we support healthcare professionals in their knowledge discovery process by applying AutoML?'.* We found that AutoML is currently only suitable for the data understanding phase of the CRISP-DM method in this first study on possibilities for AutoML adoption in healthcare.

TPOT performed best on the benchmark test along with Auto-Sklearn, but satisfied more requirements than Auto-Sklearn regarding usability. Considering the interaction of the users with the presented artefacts, a web-application and a notebook, we see that the domain experts prefer a hybrid artefact to interact with the TPOT in this case.

Although the assessed AutoML methods are capable of modelling and data pre-processing, they miss an explanation for the decisions made in the modelling process. Part of medical knowledge discovery is finding the cause of a medical event. Because modelling decisions are not shared and variable importance is absent in the result, AutoML does not support the discovery of new knowledge. However, the domain-experts point out that they see the added value of automatically finding out possible scores for their datasets. Furthermore, they mention that AutoML can help them in getting an understanding of their data in the data understanding phase of their knowledge discovery process.

8 Discussion

In this section we discuss the lessons learned during this research. First we discuss the suitability of AutoML methods for researcher-physicians, second we discuss biases in medical analytics publications towards positive outcomes.

8.1 Suitability of AutoML methods for researcher-physicians

After the elicitation of requirements, we found that researcher-physicians prefer to create logistic regression models in their research. There are multiple reasons why we find that AutoML in the way we assessed it, is not the best way to create models for researcher-physicians. First, AutoML originates from the CASH problem. If the algorithm type is pre-selected, AutoML can only contribute to HPO. As logistic regression only has two hyperparameters [13], we question the need for using AutoML to tune these hyperparameters.

Second, even if we drop the constraint of logistic regression for model construction, previous research found that other ML models do not significantly outperform logistic regression models in medical studies [51]. Even when we drop the logistic regression model constraint, we still doubt the usability of AutoML due to the results of Christodoulou et al. and the inability of AutoML to explain the created models, although other researchers do not find the same results on general datasets [52], [53].

Third, there is no explanation of variable importance by the tested AutoML method. As model explainability and variable importance are essential requirements for researcher-physicians, this makes AutoML unsuitable in its current form. If the given explanations are satisfactory, AutoML might be useful in research. More on variable importance is available in the future work section below (9.2). If AutoPrognosis can deliver on its promises it can be a promising technology considering the user stories on model explanation.

Fourth, there is no structure in the pipelines created by TPOT; this can lead to very complicated pipelines in with three or four logistic regression models, all using each other's results as input. These constructions are hard to understand for domain experts. If we would use grammar to represent the pipeline like in RECIPE [30] the grammar can help to create more understandable pipelines for researcher-physicians. Another option could be using fixed-pipeline methods based on BO. Finally, we have learned that there is a gap in the knowledge level of ML between literature and practice for researcher-physicians. The literature on AutoML states that AutoML aims to aid non-expert users of ML techniques [13]. However, we find that most non-expert users in the medical domain have no knowledge or education in programming. Hence, the current offering of AutoML techniques are still too technical for non-expert users in healthcare.

Bias in medical analytics publications

Most publications about healthcare analytics that we have come across during this research, have all published a positive result. Vollmer et al. [54] noticed the same: there is a need to publish every positive result, but there are no real tests for the value of ML solutions in healthcare. If we add the findings of Christodoulou et al. [51] to the fact that ML models do not perform significantly better than traditional methods in medical research we find a discrepancy between what is published and what is improving healthcare. Hence, we think that it would be helpful for the development of accurate methods for analytics in healthcare to publish results that do not provide a direct benefit. In this way, funding for research can be either used for improving methods that do not yet work in healthcare or in researching different methods for solving the same problem. When negative results are not published, we fear a waste of research funding by researchers continually reinventing the wheel and thus stalling research and innovation in healthcare.

9 Future research

In this section we discuss three possibilities for future research, first we discuss model uncertainty of AutoML created models. Second, we discuss possible new use-cases for AutoML methods. Finally, we discuss interpretability as a direction for future research.

9.1 AutoML model uncertainty

One of the benefits that are proposed by AutoML is the reproducibility of created ML pipelines [13], [14], [20]. However, these authors state that the outcome of the creation of an ML pipeline with an AutoML method is dependent on the time budget allocated to the AutoML method. Besides that, EA based AutoML methods start with a random population. Thus, it is harder to reproduce the result of a single run without explicitly setting the seed. Dusenberry et al. (2019) investigated model uncertainty in a medical context. They have found that as much as changing the seed can influence the prediction outcome for an individual patient. Hence, we argue that there should be more research on the stability of AutoML pipelines in the medical domain.

9.2 AutoML use-cases

In our synthesis in section **Error! Reference source not found.**, we demonstrated that AutoML is applicable for tuning neural networks and creating classification and regression models. In healthcare, the fields with the most significant potential for the application of ML are image recognition and natural language processing [54]. Because researcher-physicians accept black boxes in image recognition and natural language processing more than in traditional research, these ML tasks might be better suited to enable domain experts to work with AutoML in healthcare. Hence, we argue that the scope of AutoML use-cases could be widened to NLP and image recognition tasks if we want to accelerate the adoption of analytics in healthcare.

9.3 AutoML interpretability

As pointed out in section 6, the explainability of ML models is crucial to adoption for domain-experts in healthcare [56]. Molnar (2019) argues that ML interpretability is crucial to the adoption of black-box algorithms in every sector. In healthcare, this barrier to adoption is even higher, as being able to explain decisions is part of the medical culture and vital to patient-doctor interaction. Hence, AutoML methods must become more interpretable for non-expert users. The interpretability technique should be model agnostic. In that way, the technique is suitable for all pipelines created by the AutoML methods [58]. To improve the interpretability of AutoML models we propose three areas for further research on the interpretability of AutoML: Surrogate models, Local Interpretable Model-agnostic Explanations (LIME) [57] and Shapley values [59]. However, others argue that we should improve the trust in artificial intelligence in healthcare rather than improving the interpretability. If the trust is high enough, the researcher-physicians will start using the black boxes [60]. Most researcher-physicians also do not precisely know how a car works. However, they still use cars in their daily lives.

At the time of writing, AutoPrognosis [19] was not yet readily available. If AutoPrognosis can deliver on its promises it can be a promising technology considering the user stories on model explanation. Although it does not provide specific values for variables, it does pay attention to the explanation part of a decision of the AutoML method.

We assume that the addition of interpretability will ignite the adoption rate of AutoML methods in healthcare and other sectors to enable self-service data-science.

REFERENCES

[1] Rijksinstituut voor Volksgezondheid en Milieu, “Zorguitgaven | Volksgezondheid Toekomst Verkenning,” 2019. [Online]. Available: https://www.vtv2018.nl/zorguitgaven. [Accessed: 28-Mar-2019].

[2] Y. Wang and N. Hajli, “Exploring the path to big data analytics success in healthcare,” *J. Bus. Res.*, vol. 70, pp. 287–299, 2016.

[3] X. Wang, M. Noor-E-Alam, M. Islam, M. Hasan, and H. Germack, “A Systematic Review on Healthcare Analytics: Application and Theoretical Perspective of Data Mining,” *Healthcare*, vol. 6, no. 2, p. 54, 2018.

[4] M. Spruit and R. Jagesar, “Power to the People! - Meta-Algorithmic Modelling in Applied Data Science,” *Proc. 8th Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl. Manag.*, no. January, pp. 400–406, 2016.

[5] H. C. Koh and G. Tan, “Data mining applications in healthcare,” vol. 19, no. 2, pp. 64–72, 2005.

[6] N. Vries de, “Making machine learning accessible to healthcare professionals for the purpose of predicting medical adverse events (MSc. Thesis),” Utrecht University, 2018.

[7] N. V. Chawla and D. A. Davis, “Bringing big data to personalized healthcare: A patient-centered framework,” *J. Gen. Intern. Med.*, vol. 28, no. SUPPL.3, pp. 660–665, 2013.

[8] J. Manyika, B. Chui, M., J. B., Bughin, R. Dobbs, C. Roxburgh, and A. Hung Byers, “Big data: The next frontier for innovation, competition and productivity,” *McKinsey Glob. Inst.*, no. May, 2011.

[9] J. Harris, N. Shetterley, A. Alter, and K. Schnell, “It Takes Teams to Solve the Data Scientist Shortage,” *CIO Journal. - WSJ blogs*, pp. 2–5, 2017.

[10] W. Markow, S. Braganza, B. Taska, D. Hughes, and S. Miller, “The Quant Crunch,” p. 25, 2017.

[11] K. Gibert, J. S. Horsburgh, I. N. Athanasiadis, and G. Holmes, “Environmental Data Science,” *Environ. Model. Softw.*, vol. 106, pp. 4–12, 2018.

[12] T. H. Davenport and D. J. Patil, “Data scientist: The sexiest job of the 21st century,” *Harv. Bus. Rev.*, vol. 90, no. 10, p. 5, 2012.

[13] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms,” *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, 2013.

[14] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automatic machine learning: methods, systems, challenges*. 2019.

[15] P. Chapman *et al.*, “Crisp-Dm 1.0,” *Cris. Consort.*, p. 76, 2000.

[16] T. M. Mitchell, *Machine learning*. New York, NY: McGraw-Hill, Inc., 1997.

[17] Y. Quanming *et al.*, “Taking Human out of Learning Applications: A Survey on Automated Machine Learning,” *arxiv pre-print*, no. November, 2018.

[18] Y. Zhang, M. T. Bahadori, H. Su, and J. Sun, “FLASH: Fast Bayesian Optimization for Data Analytic Pipelines,” pp. 1–21, 2016.

[19] A. M. Alaa and M. van der Schaar, “AutoPrognosis: Automated Clinical Prognostic Modeling via Bayesian Optimization with Structured Kernel Learning,” in *International Conference on Machine Learning*, 2018, pp. 139–148.

[20] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, “Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA,” *J. Phys. B At. Mol. Opt. Phys.*, vol. 18, pp. 1–5, 2017.

[21] B. Komer, J. Bergstra, and C. Eliasmith, “Hyperopt-Sklearn: Automatic HyperparameterConfiguration for Scikit-Learn,” *ICML Work. AutoML*, pp. 2825–2830, 2014.

[22] M. Feurer, J. T. Springenberg, A. Klein, M. Blum, K. Eggensperger, and F. Hutter, “Efficient and Robust Automated Machine Learning,” *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, pp. 2755–2763, 2015.

[23] M. Feurer, K. Eggensperger, S. Falkner, Lindauer, and F. Hutter, “Practical Automated Machine Learning for the AutoML Challenge 2018,” in *ICML 2018 AutoML Workshop*, 2018.

[24] F. Mohr, M. Wever, and E. Hüllermeier, “ML-Plan: Automated machine learning via hierarchical planning,” *Mach. Learn.*, vol. 107, no. 8–10, pp. 1495–1515, 2018.

[25] H. Mendoza, A. Klein, M. Feurer, J. T. Springenberg, and F. Hutter, “Towards Automatically-Tuned Neural Networks,” *Proc. Work. Autom. Mach. Learn.*, pp. 58–65, 2016.

[26] I. Drori *et al.*, “AlphaD3M : Machine Learning Pipeline Synthesis,” in *ICML 2018 AutoML Workshop*, 2018.

[27] H. Jin, Q. Song, and X. Hu, “Auto-Keras: Efficient Neural Architecture Search with Network Morphism,” *arxiv pre-print*, 2018.

[28] R. S. Olson and J. H. Moore, “TPOT: A tree-based pipeline optimization tool for automating machine learning,” *Work. Autom. Mach. Learn.*, pp. 66–74, 2016.

[29] P. Gijsbers, J. Vanschoren, and R. S. Olson, “Layered TPOT: Speeding up tree-based pipeline optimization,” in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2017, pp. 49–68.

[30] A. G. C. Sá de, W. J. G. S. Pinto, L. O. V. B. Oliveira, and G. L. Pappa, “RECIPE: A grammar-based framework for automatically evolving classification pipelines,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10196 LNCS, pp. 246–261, 2017.

[31] B. Chen, H. Wu, W. Mo, I. Chattopadhyay, and H. Lipson, “Autostacker:

A Compositional Evolutionary Learning System," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 402–409.

[32] T. Swearingen, W. Drevo, B. Cyphers, A. Cuesta-Infante, A. Ross, and K. Veeramachaneni, "ATM: A distributed, collaborative, scalable system for automated machine learning," in *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, 2017, pp. 151–162.

[33] A. R. Hevner, S. Ram, S. March, and J. Park, "Design Science in Information Systems," *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[34] P. Gijsbers, E. Ledell, J. Thomas, S. Poirier, B. Bischl, and J. Vanschoren, "An Open Source AutoML Benchmark," in *ICML workshop on AutoML*, 2019, pp. 1–8.

[35] B. Bischl *et al.*, "OpenML Benchmarking Suites and the OpenML100," *arxiv pre-print*, pp. 1–6, 2017.

[36] O. L. Mangasariona and W. H. Wolberg, "Cancer Diagnosis via Linear Programming," *SIAM News*, vol. 23, no. 5, pp. 1–18, 1990.

[37] D. Dua and C. Graff, "UCI Machine Learning Repository," 2019. [Online]. Available: https://archive.ics.uci.edu/ml/index.php.

[38] J. R. Quinlan, "Simplifying Decision Trees," *MIT Artif. Intell. Lab.*, pp. 81–106, 1986.

[39] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. M. Moreno, "Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review," in *14th IEEE International Requirements Engineering Conference*, 2006, pp. 179–188.

[40] Z. Q. Al-Busaidi, "Qualitative research and its uses in health care.," *Sultan Qaboos Univ. Med. J.*, vol. 8, no. 1, pp. 11–9, 2008.

[41] C. Pope, P. van Royen, and R. Baker, "Qualitative methods in research on healthcare quality.," *Qual. Saf. Health Care*, vol. 11, no. 2, pp. 148–52, 2002.

[42] M. Castillo-Montoya, "Preparing for interview research: The interview protocol refinement framework," *How To Artic.*, vol. 21, no. 5, pp. 811–831, 2016.

[43] J. Venable, J. Pries-Heje, and R. Baskerville, "FEDS: A Framework for Evaluation in Design Science Research," *Eur. J. Inf. Syst.*, vol. 25, no. 1, pp. 77–89, 2016.

[44] P. Offermann, O. Levina, M. Schönherr, and U. Bub, "Outline of a Design Science Research Project," *4th Int. Conf. Des. Sci. Res. Inf. Syst. Technol.*, p. 11, 2009.

[45] B. Kaplan and J. Maxwell, "Qualitative Research Methods for Evaluating Computer Information Systems," in *Evaluating the organizational impact of healthcare information systems*, Springer New York, 2005.

[46] R. Burke Johnson, "Examining the Validity Structure of Qualitative Research," *Education*, vol. 118, no. Winter, pp. 282–292, 1997.

[47] M. Kiegelmann, *The role of the researcher in qualitative psychology*. Ingeborg Huber Verlag, 2002.

[48] D. F. Polit and C. T. Beck, "Generalization in quantitative and qualitative research: Myths and strategies," *Int. J. Nurs. Stud.*, vol. 47, no. 11, pp. 1451–1458, 2010.

[49] M. Cohen, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[50] J. H. N. Janssen, "The right to explanation: means for 'white-boxing' the black-box? (MSc. Thesis)," Tilburg University, 2019.

[51] E. Christodoulou, J. Ma, G. S. Collins, E. W. Steyerberg, J. Y. Verbakel, and B. Van Calster, "A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models," *J. Clin. Epidemiol.*, vol. 110, pp. 12–22, 2019.

[52] J. Gareth, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*. 2013.

[53] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[54] S. Vollmer *et al.*, "Machine learning and AI research for Patient Benefit: 20 Critical Questions on Transparency, Replicability, Ethics and Effectiveness," in *The Alan Turing Institute*, 2018.

[55] M. W. Dusenberry *et al.*, "Analyzing the Role of Model Uncertainty for Electronic Health Records," *arxiv pre-print*, pp. 1–14, 2019.

[56] N. Sung *et al.*, "Central Challenges Facing the National Clinical Research Enterprise," *JAMA - J. Am. Med. Assoc.*, vol. 289, no. 10, pp. 1278–1287, 2003.

[57] C. Molnar, *Interpretable Machine Learning*. leanpub.com, 2019.

[58] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-Agnostic Interpretability of Machine Learning," in *ICML Workshop on Human Interpretability in Machine Learning*, 2016.

[59] L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games*, 1953, pp. 307–317.

[60] I. Bartoletti, "AI in Healthcare: Ethical and Privacy Challenges," in *Artificial Intelligence in Medicine*, 2019, pp. 7–10.