Selection and Application of Machine Learning-Algorithms in Production Quality

Jonathan Krauß[1], Maik Frye[1], Gustavo Teodoro Döhler Beck[1], Robert H. Schmitt[2]

[1] Fraunhofer Institute for Production Technology IPT
Steinbachstr. 17, Aachen 52074, Germany
{jonathan.krauss,maik.frye,gustavo.beck}@ipt.fraunhofer.de

[2] Laboratory for Machine Tools WZL RWTH Aachen University
Steinbachstr. 19, Aachen 52074, Germany
robert.schmitt@rwth-aachen.de

Abstract. Due to the increase in digitalization Machine Learning (ML)-algorithms bare high potentials for process optimization in the production quality-domain. Nowadays, ML-algorithms are hardly implemented in the production environment. In this paper, we present a tangible use case in which ML-algorithms are applied for predicting the quality of products in a process chain and present the lessons learned we extracted from the application. In the described project, the process of choosing ML-algorithms was a bottleneck. Therefore we describe a promising approach how a decision making tool can help selecting ML-algorithms problem-specifically.

1 Data-Driven Modeling in the Production Quality

Digitalization has led to a steady increase in data in recent years. Through higher computing power, it is possible to process the large amount of data [1]. Analyzing the acquired data can enhance both the understanding and the process efficiency - or to describe it in the words of Peter Sondergaard: “Information is the oil of the 21st century, and analytics is the combustion engine” [2]. Especially sectors like the financing-domain or the marketing-domain are leading when it comes to generate value from data [3]. In particular, the use of Machine Learning (ML)-algorithms increased over the last decade. The main reasons for this trend, apart from the higher computing power and data input mentioned above, are the increasing reliability of the algorithms, the simpler implementation of the algorithms as well as the easier data acquisition. [1]

Even though the application of ML-algorithms is well established in other domains, it is not common in the context of production quality. For process optimization in the production quality-domain, physically based modeling (PBM) is commonly used. While PBM offers the advantage of describing the current and future state of a system by physical dependencies, data-driven models use the information from observed data to identify current system characteristics and to predict the future state without requiring a deeper understanding of the physical interdependencies of the process. [4] The development of data-driven models thus shows a high potential for even further optimization of production processes. In the presented case we chose to transform the data into a data-driven model by applying ML-algorithms.

J. Beyerer et al. (Eds.), *Machine Learning for Cyber Physical Systems*, Technologien für die intelligente Automation 9, https://doi.org/10.1007/978-3-662-58485-9_6

2 Application of Machine Learning in the Production Quality

2.1 Prediction of Product Quality in a Process Chain

In the following, we want to show in a tangible use case at a German manufacturing company that the application of ML-algorithms is worthwhile and to encourage companies to use ML-algorithms for process optimization. To introduce data-driven modeling for process optimization, the Cross-industry standard process for data mining (CRISP-DM) procedure can generally be used [5]. The first step is to understand the corresponding business in more detail. After an initial data acquisition, the characteristics of the data are determined in order to understand the data. The data is subsequently prepared for the application of a suitable ML-algorithm. Based on the data preparation, the implementation of the selected ML-algorithm is described. Finally, the results of the model are evaluated, whereby various criteria are taken into account. Tangible lessons learned will be presented extensively.

The first step of the CRISP-DM is the Business Understanding. The company in this specific use case aims to enhance the efficiency of a process chain, which consists of six different processes. Each product runs through every process sequentially with some processes taking several hours or even days. In order to get a better understanding of the process chain and the corresponding data, we conducted several workshops and web conferences with the company's process engineers. The process chain is depicted in **Fig. 1**.

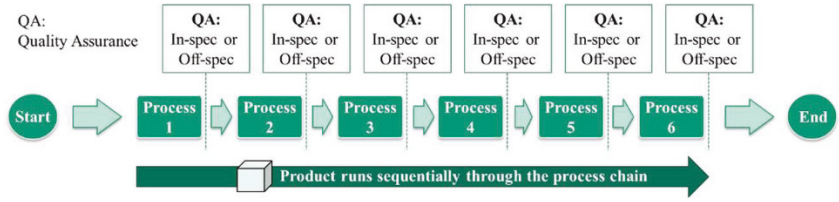

Fig. 1. Illustration of the process chain

Whether a product is an in-spec product can be determined after the completion of each process. Since the cycle time of the entire process chain takes several days, it would be useful to predict whether a product will run out of specification in a process already in earlier stages. If it can accurately predicted that a product will run out of specification, the machines could be equipped with other products. This leads to higher efficiency as well as flexibility of the entire process chain.

Data Understanding as the second step of the CRISP-DM process shows a strong relation to the Business Understanding to the effect that both steps require multiple loops and iterations. The acquired data is stored in separate product-related databases for each of the six processes as semi-structured CSV-files. Due to acquiring a large number of measuring values, there are more than 500 values per process for each product. Different data types like integer, float or string parameters characterize this high

amount of dimensions. Besides a multitude of missing values, the data set is also imbalanced. In this context, an imbalanced dataset means that more products are in-spec than off-spec.

To predict the product quality it is necessary to trace the product data throughout the entire process chain. For that reason, the six different CSV-files need to be linked. This link is created using a product identification number. Since the CSV-files are not uniformly structured, the files need to be transformed multiple times. After the product-related link, the data is cleaned by deleting empty values, apparent correlations as well as by reducing dimensions. Overall, the process of data understanding and preparation took about 80 % of the time regarding the entire CRISP-DM procedure.

In the beginning of the modeling step, a suitable approach how to create a model needs to be selected. Due to the time it takes to learn a data-driven model with an ML-algorithm, only a small number of algorithms can be applied. The process of selecting ML-algorithms depends highly on the use case, the appearance of the data set and the personal experience of the involved data scientists. In this specific case, we interpret the prediction whether a product will be in-spec or off-spec as a classification problem. One class includes all products that run through the process chain being in-spec. Since the quality of the product is measured after each process, the product can become off-spec after each process resulting in six additional classes. Because we are able to label the data set, this multiclass classification problem can be solved using supervised learning algorithms. **Fig. 2** shows a visualization of the processes and the even classes.

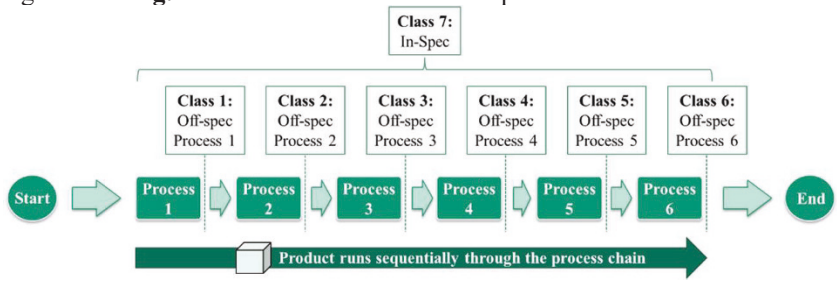

Fig. 2. Visualization of the processes as well as the seven classes

The characteristics of the data set result in the requirements for the algorithm that has to deal with an imbalanced data set, few samples as well as many dimensions. Best practices in other sectors with similar problems are taken from the literature. Besides the results of the literature research, own experiences show beneficial results when decision tree algorithms are applied. Considering the mentioned explanations, the decision tree algorithm Classification and Regression Tree (CART) is selected for this use case [6]. CART can handle high dimensional data sets and has the further advantage that process owners can understand the results of the analysis very quickly and intuitively. The localization, in which the prediction states that the product will run out of tolerance, can be easily detected. Furthermore, the implementation and validation of the decision tree algorithm is simple.

There exist many different platforms for Data Mining as well as ML-algorithm implementation [7]. These platforms can be divided in “Data Science and Machine Learning platforms” like Matlab or RapidMiner and “open source platforms” like Python and

R. Data Science and Machine Learning platforms are characterized by easy handling and fast model development [8]. Nevertheless, operating the platform can result in high licensing costs [9]. Open source platforms like Python and R play an increasingly important role in the data science market because they are free of charge and are the most common programming languages for ML-implementation [8]. We decided to use the "open source platform" Python because the libraries that can be called, such as TensorFlow and scikit-learn, are undergoing strong development. The algorithm is implemented in Python by calling the decision tree algorithm via the scikit-learn library. Scikit-learn uses an optimized version of the CART algorithm [10].

To achieve betters performances of the ML-algorithm, hyperparameter must be set. Hyperparameters are the configuration that is external to the model and whose values cannot be estimated from the data set [11]. They are initially set when the algorithm is called by scikit-learn and need to be optimized. Hyperparameters of the decision tree algorithm are e.g. the maximum depth and the minimum size of the tree. There are different approaches to optimize hyperparameters. For this use case, the basic approach, called random search, is applied on the decision tree algorithm. Random search randomly selects any combination of the hyperparameters to be set within an interval of possible hyperparameters. If this combination of hyperparameters lead to better results, the parameters are updated. Basic approaches to set and tune hyperparameters are grid-search and random-search. Over the last years, other tuning approaches like Bayesian Optimization and Gradient Descent became popular [12]. In addition to these advanced approaches, research institutes try to apply heuristics to the hyperparameter tuning-problem. These academic approaches include metaheuristics like Particle Swarm Optimization, Ant Colony Optimization and Harmony Search [13].

After running, the performance of the model can be evaluated by a multitude of metrics. The basis of measuring the performance of a classification model is the confusion matrix. The rows of the 2x2 confusion matrix represent the instances in a predicted class while the columns represent the instances in an actual class [14]. If the classification model correctly classifies the input as positive (in-spec) or negative (off-spec), they are considered as true positives (TP) or true negatives (TN). Classifying products falsely as positive or negative counts as false positive (FP) or false negative (FN). Based on the confusion matrix, we can derive different metrics.

Metrics that can be easily derived from the confusion matrix are accuracy and error rate. Other single-value metrics like the F1-Score and Mathew Correlation Coefficient (MCC) are more complex to set up but can still be derived from the confusion matrix. In order to evaluate the performance of the CART algorithm in this specific use case, the MCC is selected. MCC considers imbalanced data sets more efficiently than accuracy and error rate [14]. The mathematical relationship can be taken from equation **(1)**.

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP}) \cdot (\mathrm{TP}+\mathrm{FN}) \cdot (\mathrm{TN}+\mathrm{FP}) \cdot (\mathrm{TN}+\mathrm{FN})}} \tag{1}$$

The MCC considers both mutual accuracies and error rates on both classes. Furthermore, the MCC is a coefficient between the observed and predicted classifications and

returns a value between "−1" and "+1". A coefficient of "+1" represents a perfect prediction, "0" no better than random prediction and "−1" indicates total disagreement between prediction and observation. [14]

In order to predict the product quality after each process, different CART-algorithms need to be trained because at each process, different amount of data is available to train the CART-algorithm. This leads to four different CART-algorithms, whose performances are depicted in **Fig. 3**. The results include the decision trees that were created after the hyperparameter tuning. By applying random search, the results could be improved by 30% which can be observed in other cases as well [15]. Since no new data is generated in the fourth process, no new decision tree was learned for the change from the fourth to the fifth process.

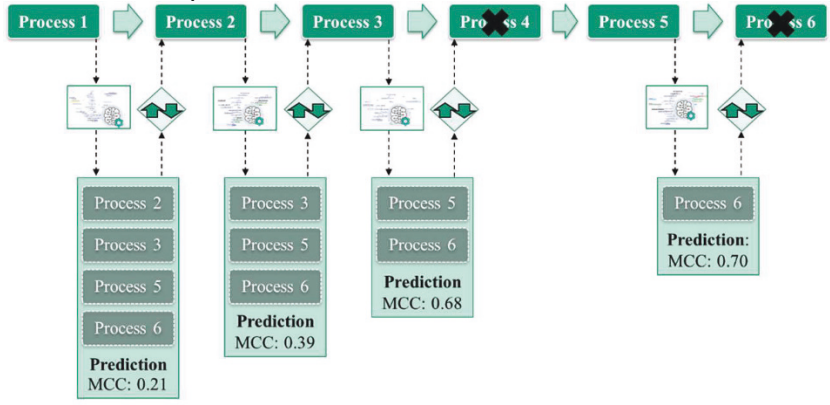

Fig. 3. Performance of the decision tree algorithm

The metric MCC shows the performance of the algorithm in predicting the actual classes of the process. For the first process step the metric is MCC = 0.21. This means that there is a match between predicted and actual class, which is relatively low, but better than random prediction. The MCC increases the more processes are accomplished and the fewer processes have to be carried out. The quality of the model improves when more data points are used for the learning task. In addition, less processes and results need to be predicted for the future. After the completion of the fifth process, the metric value is MCC = 0.70, which means that the decision tree is a suitable algorithm to predict the product quality sufficiently [16].

2.2 Lessons Learned

In the following, tangible lessons learned are presented, starting with the management level. Then, there will be a focus on the lessons learned for project managers as well as for computer scientists and developers. Two central research needs result from the presented method.

Lessons learned from the managers' perspective:

- In principle, only available data can be analyzed. Big data only leads to beneficial results if the quality of acquired data is acceptable.

- Managers need to have a precise idea, which results are to be achieved by the analysis of data. Expectations for initial projects must be appropriate.
- Data science projects should start at small scales. As mentioned, simple algorithms like the decision tree can already lead to beneficial results. If the first projects proceed favorable, the projects can be scaled up.

Lessons learned from the project managers' perspective:

- Understanding the business goals and formulating precise objectives is essential when ML-algorithms are implemented.
- A very close collaboration between process engineers and data scientists is unalterable.
- It is natural that many iterations are necessary to get to know technical interdependencies and the characteristics of the data set. The processes of data understanding and data preparation takes a long time compared to the implementation of the ML-algorithm in the end.
- A project manager should be aware of whether it is a quick-win project, low complexity project or long-term commitment.

Lessons learned from the computer scientists and developers' perspective:

- Python notebooks like Jupyter are able to segment the entire code in sensible parts [17]. With Python notebooks, the code can be sequentially updated, which makes coding easier and faster. Introduced variables should still be readable and understandable at the very end of the project.
- Since the selection of a suitable ML-algorithm depends highly on the use case, the appearance of the data set and the personal experience of the involved data scientists, the choice for the ML-algorithm is difficult and sophisticated.
- The hyperparameter tuning is unalterable to solve the classification problem optimally. An optimization based on random search was successful, but advanced optimizations can lead to better solutions.

Overall, we recommend that companies should start with the first data science projects and make their own experiences. Based on first it can be obtained what specific challenges will happen. To describe it in other words - practice makes it perfect!
In addition to the lessons learned, we can derive two central research needs from the presented procedure and the lessons learned. First, it should be evaluated whether more complex hyperparameter optimization methods are capable of outperforming basic approaches like random search and grid search. Second, the procedure of selecting the suitable ML-algorithm was built up on the experiences we had and by comparing the learning task with the literature. A tool supporting us in selecting an appropriate ML-algorithm would have made the process more transparent and reproducible. In the following, we propose a concept how such a tool can function.

3 Selection of Machine Learning-Algorithms

The use of methodologies to solve a specific task creates comprehensible and reproducible results. Therefore, methodologies were developed especially for data mining

and knowledge discovery [18]. Due to the mentioned benefits, they are used in the majority of corresponding projects [3].

CRISP-DM, SEMMA (Sample, Explore, Modify, Model, and Assess) and KDD (Knowledge Discovery in Databases) as the top three methodologies all include a phase specifically designated to create the model for the problem [19]. Due to the generic nature of the three methodologies, the activities in the phase of "Modeling" can be on a different level of complexity ranging from the application of linear regression up to deep learning. Therefore, a data scientist has to decide how to conduct the phase of "Modeling" e.g. by applying an ML-algorithm. Normally the following three aspects are included in this decision: Personal experience, appearance of the data set and literature review. [20]

The problems and corresponding data sets that need to be tackled are domain-specific. Tools that support the data scientist in selecting an ML-algorithm are mostly so called "cheat sheets" [21]. Team members solely bring domain-specific knowledge into the solution. The process of choosing the ML-algorithm is therefore highly dependent on the expertise of the data scientist. Since neither methodologies nor tools include this domain-specific knowledge, the process of selecting the ML-algorithm is not reproducible. Not all domain-specific knowledge can be integrated into a tool. The process of selecting the ML-algorithm stands out by the required creativity of the data scientist. Therefore a decision making tool cannot dismiss the data scientist from his responsibility, but can serve as a support in fulfilling that task. In the following, we present a concept how to set up such a domain-specific decision making tool.

4 Decision Making Tool for Production Quality

The decision making tool (DMT) works as a domain-specific support for the data scientist in selecting an appropriate ML-algorithm to create a model that fulfils problem-specific requirements. This is done by including three main aspects as depicted in **Fig. 4**: Appearance of the data input, requirements of the model to be created and domain-specific knowledge regarding the considered use case. All three factors are included when providing the user a recommendation.

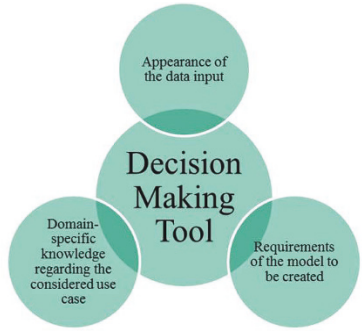

Fig. 4. Factors to be considered when selecting an ML-algorithm

The data scientist interacts with the DMT over a user interface (UI), which he utilizes to describe the specific case he wants to model applying ML-algorithms. The DMT compares the input with historical assessments and problems, including their evaluation. Afterwards the DMT provides the data scientist a list of ML-algorithms probably suitable for the specific use case and additional information about the corresponding selection process. The concept of the DMT is depicted in **Fig. 5** and described in detail in the following.

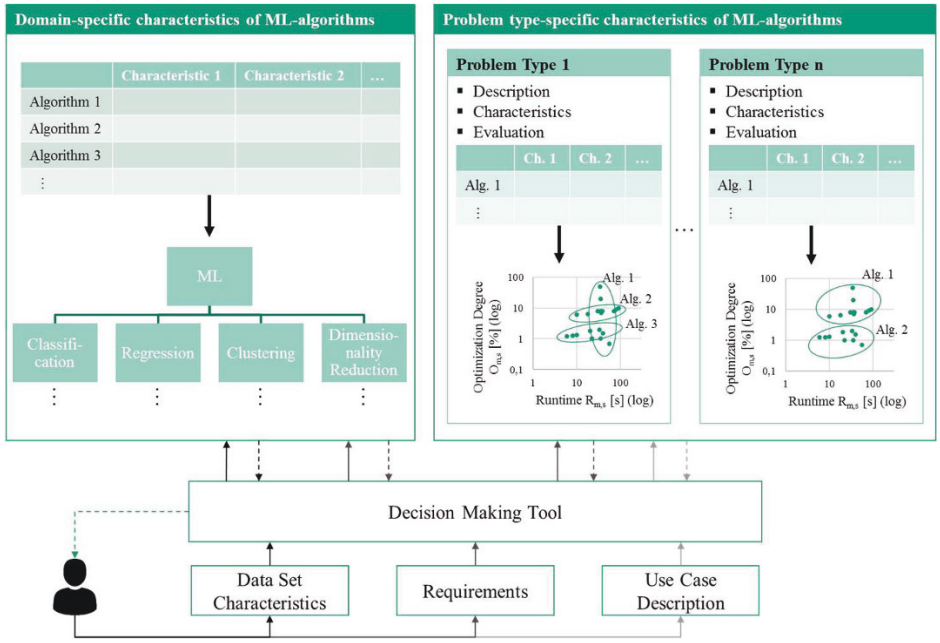

Fig. 5. General Concept of the Decision Making Tool

Using the UI, the data scientist loads the characteristics of the data set, the requirements of the model to be created and a description of the use case into the DMT. Characteristics of the data set are for example the dimensionality of the data, number of features, number of data points, data quality, data distribution or data noise. Requirements of the model to be created are for instance the learning time, performance of the model or transparency of the model. The description of the use case includes information about the type of the use case, e.g. predictive maintenance or product quality prediction. Characteristics like the dimensionality or the maximum running time are quantitative and can directly be loaded into the DMT. Others like the transparency of the model need to be transformed from their qualitative state into a measurable form using for example goal question metrics [22]. This influences the degree of automation to which the characteristics can be loaded into the DMT.

Two main databases function as the backbone of the DMT: A database that includes the domain-specific characteristics of ML-algorithms and a database that stores problem-specific characteristics of ML-algorithms.

The domain-specific characteristics include the attributes of ML-algorithms that are important in the context of the production quality-domain. This includes characteristics and an assessment to which degree the algorithms resp. the learned models meet these characteristics such as interpretability, decomposability, speed, accuracy or learning time. The database is set up and maintained by data scientists working in the production quality-domain.

The problem-specific characteristics are structured by the different types of problems occurring in the production quality-domain such as machine downtime prediction or product failure prediction. For each type of problem, the corresponding description and attributes are available, so that the use case provided by the user can be matched to the most-fitting problem-type in the database. For each problem type from the production quality-domain, different ML-algorithms have been implemented in the past. The information, which algorithms are suitable for the problem-type and the evaluation of their performance is stored accordingly. This is realized by using algorithm maps also known as optimization maps . Each time new types of problems or new evaluations are created, responsible data scientists update the database consequently. This ensures that the specific demands of the production quality-domain and the problem-specific evaluations are considered in the selection process.

The DMT creates a list of algorithms that are promising for the use case by comparing the characteristics of the data set, the requirements of the model to be created and the description of the use case with the historical information stored in the two data bases.

5 Conclusion

In this paper, we presented how ML-algorithms can be applied in a tangible use case from the production quality-domain. In a process chain consisting of six processes, it should be predicted after completion of each individual process whether the product would be off-spec in the following processes. In order to achieve beneficial results, the methodology CRISP-DM was followed. After focusing on the process understanding, data was initially acquired. Afterwards, formats as well as characteristics of the data set has been explored. The preparation of the data comprised the cleaning, transforming and dimensionality reduction in order to apply the ML-algorithm sufficiently. Since we have a multiclass classification problem, the decision tree algorithm CART was selected. The evaluation of the CART algorithm showed that both the methodology and the application of ML-algorithms could lead to beneficial results. On the basis of the mentioned use case, tangible lessons learned could be derived and were divided into lessons learned on the management, project and technology level.

Based on the variety of ML-algorithms, it is difficult to determine, which ML-algorithm is the most suitable for predicting the product quality. In this use case, we compared the performance of different algorithms. These algorithms were selected by the character of the problem, by analyzing the data, by reviewing literature and by the

authors own experience. This process of choosing the ML-algorithm is highly dependent on the expertise of the involved team members. Therefore, a tool that supports the user selecting the ML-algorithm could help in making the process more reliable.

We explained why methodologies are widely used in data mining-projects but why they are just a footnote when choosing ML-algorithm for a specific problem. A concept how a DMT can support data scientists in selecting ML-algorithms for a specific problem was presented. The DMT takes domain-specific demands into account and characterizes ML-algorithms accordingly. Problem type-specific evaluations of ML-algorithms are included in the recommendations. Nevertheless, domain-specific knowledge, expertise regarding selection and implementation of ML-algorithms and the creativity of data scientists will not become obsolete.

6 Funding notes

“The IGF promotion plan 18504N of the Research Community for Quality (FQS), August-Schanz-Str. 21A, 60433 Frankfurt/Main has been funded by the AiF within the programme for sponsorship by Industrial Joint Research (IGF) of the German Federal Ministry of Economic Affairs and Energy based on an enactment of the German Parliament.”

7 References

1. Michael Driscoll (2011) Building data startups: Fast, big, and focused: Low costs and cloud tools are empowering new data startups. http://radar.oreilly.com/2011/08/building-data-startups.html. Accessed 14 May 2018
2. Peter Sondergaard (2011) Gartner Says Worldwide Enterprise IT Spending to Reach $2.7 Trillion in 2012. https://www.gartner.com/newsroom/id/1824919. Accessed 14 May 2018
3. Piatetsky-Shapiro G (2014) What main methodology are you using for your analytics, data mining, or data science projects? https://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html. Accessed 14 May 2018
4. Datong P. Zhou, Qie Hu, Claire J. Tomlin (2017) Quantitative comparison of data-driven and physics-based models for commercial building HVAC systems
5. Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer and Rüdiger Wirth CRISP-DM: Step-by-step data mining guide
6. Scikit-learn Developers (2018) Decision Tree Classifier. http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. Accessed 14 May 2018
7. Gregory Piatetsky (2018) Survey regarding data mining platforms: What software you used for Analytics, Data Mining, Data Science, Machine Learning projects in the past 12 months? http://vote.sparklit.com/poll.spark/203792. Accessed 14 May 2018
8. Gregory Piatetsky (2018) Gainers and Losers in Gartner 2018 Magic Quadrant for Data Science and Machine Learning Platforms. https://www.kdnuggets.com/2018/02/gartner-2018-mq-data-science-machine-learning-changes.html. Accessed 14 May 2018
9. Gregory Piatetsky (2017) Forrester vs Gartner on Data Science Platforms and Machine Learning Solutions. https://www.kdnuggets.com/2017/04/forrester-gartner-data-science-platforms-machine-learning.html. Accessed 14 May 2018
10. Scikit-learn Developers (2018) Decision Trees. http://scikit-learn.org/stable/modules/tree.html. Accessed 14 May 2018
11. Dr. Jason Brownlee (2017) What is the Difference Between a Parameter and a Hyperparameter? https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/. Accessed 14 May 2018
12. Rafael G. Mantovani, Tomáš Horváth, Ricardo Cerri, Joaquin Vanschoren, André C.P.L.F. de_Carvalho (2016) Hyperparameter Tuning of a Decision Tree Induction Algorithm. IEEE, Piscataway, NJ
13. Pawel Matuszyk, Rene Tatua Castillo, Daniel Kottke A Comparative Study on Hyperparameter Optimization for Recommender Systems

14. Mohamed Bekkar, Dr.Hassiba Kheliouane Djemaa, Dr.Taklit Akrouf Alitouche Evaluation Measures for Models Assessment over Imbalanced Data Sets. 2013
15. Patrick Koch, Brett Wujek, Oleg Golovidov et al. (2017) Automated Hyperparameter Tuning for Effective Machine Learning
16. Thusberg J, Olatubosun A, Vihinen M (2011) Performance of mutation pathogenicity prediction methods on missense variants. Hum Mutat 32(4): 358–368. doi: 10.1002/humu.21445
17. (2018) Jupyter Notebook. https://jupyter.readthedocs.io/en/latest/architecture/how_jupyter_ipython_work.html. Accessed 14 May 2018
18. Mariscal G, Marbán Ó, Fernández C (2010) A survey of data mining and knowledge discovery process models and methodologies. The Knowledge Engineering Review 25(02): 137–166. doi: 10.1017/S0269888910000032
19. Azevedo A, Santos MF (2008) KDD, SEMMA and CRISP-DM: a parallel overview July 24-26, 2008. Proceedings. In: Abraham A (ed) IADIS European Conference on Data Mining 2008, Amsterdam, The Netherlands, July 24-26, 2008. Proceedings. IADIS, pp 182–185
20. Piatetsky-Shapiro G (2017) Top Data Science and Machine Learning Methods Used in 2017. https://www.kdnuggets.com/2017/12/top-data-science-machine-learning-methods.html. Accessed 14 May 2018
21. pakalra, olprod, OpenLocalizationService (2017) Machine Learning – Cheat Sheet für Algorithmen für Microsoft Azure Machine Learning Studio. https://docs.microsoft.com/de-de/azure/machine-learning/studio/algorithm-cheat-sheet. Accessed 14 May 2018
22. Basili VR, Caldiera G, Rombach HD (1994) The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. Wiley
23. Pitzer E, Affenzeller M (2012) A Comprehensive Survey on Fitness Landscape Analysis. In: Fodor J, Klempous R, Suárez Araujo CP (eds) Recent Advances in Intelligent Engineering Systems, vol 378. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 161–191