An End-to-End Process Model for Supervised Machine Learning Classification: From Problem to Deployment in Information Systems

Robin Hirt, Niklas Kühl, and Gerhard Satzger

Karlsruhe Service Research Institute (KSRI),
Karlsruhe Institute of Technology (KIT),
Kaiserstr. 89, 76131 Karlsruhe
{hirt,kuehl,gerhard.satzger}@kit.edu
http://www.kit.edu

Abstract. Extracting meaningful knowledge from (big) data represents a key success factor in many industries today. Supervised machine learning (SML) has emerged as a popular technique to learn patterns in complex data sets and to identify hidden correlations. When this insight is turned into action, business value is created. However, common data mining processes are generally not tailored to SML. In addition, they fall short of providing an end-to-end view that not only supports building a "one off" model, but also covers its operational deployment within an information system.
In this research-in-progress work we apply a Design Science Research (DSR) approach to develop a SML process model artifact that comprises model initiation, error estimation and deployment. In a first cycle, we evaluate the artifact in an illustrative scenario to demonstrate suitability. The results encourage us to further refine the approach and to prepare evaluations in concrete use cases. Thus, we move towards contributing a general process model that supports the systematic design of machine learning solutions to turn insights into continuous action.

Keywords: Data Mining Process, Supervised Machine Learning, Information Systems

1 Introduction & Methodology

In parallel to the "data tsunami" triggered by sensor or social media data [1], also the availability of methods and tools to exploit data has quickly picked up. Thus, possibilities to take advantage from insights drawn from (big) data have dramatically increased [2]. While many early attempts in knowledge discovery or data mining have focused on one time analyses, organizations increasingly embed such machine learning approaches in operational processes to reap ongoing benefits, e.g. predictive maintenance provision, forecasting processes, or customer churn predictions. Significant importance has been attributed to supervised machine learning approaches [3, 4]—where developed models can be turned into analytics services embedded within larger applications [5].

While there are several different process models on data mining in general and although these models are widespread [6], existing approaches bear deficiencies in at least two aspects—as we will show in more detail later: First, they are not tailored to SML classification and, thus, not granular enough to serve as a hands-on guidance for data analysts. Second, they typically do not cover the critical step of the model error estimation and the ultimate process step to deploy an analytics service within an information system. In order to address both of these gaps, we apply a Design Science Research (DSR) approach [7] to design a comprehensive, end-to-end process model specifically for classification using SML. This artifact describes the activities and the data flow during the initiation, error estimation and deployment of a generic SML classification model built to predict a certain attribute from a given dataset. Thus, we aim to add knowledge in the form of operational principles/architecture, thus making a *"level 2"* contribution to knowledge ("nascent design theory") [8, p. 341]. As we aim to develop a new solution for a known problem, the DSR contribution type is an *improvement* [9]. Figure 1 depicts our approach—with the individual steps

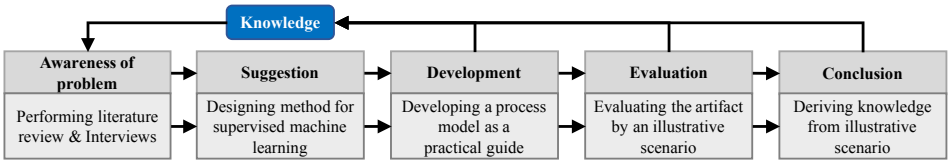

Fig. 1. DSR activities of the first cycle, according to Peffers et al. [10]

also serving to structure the remaining paper: At first, we review relevant literature and conduct exploratory interviews with two experts from industry—both confirming awareness of an issue (section 2). Then, we explain the suggestion and development of our novel process model (section 3). In the evaluation step, we test the artifact for suitability in an illustrative scenario [11] (section 4). According to Peffers et al. [10], an illustrative scenario is an evaluation method type and defined as "Application of an artifact to a synthetic or real-world situation aimed at illustrating suitability or utility of the artifact" [10]. In our case we apply the process model (*artifact*) to the development of a SML classification service to predict the age of Twitter users (*real-world situation*) to illustrate its suitability. Finally, we derive knowledge out of the completed design cycle—which then leads us into subsequent design cycles (section 5).

2 Awareness of Problem

There is a variety of different process models for data mining [6], common representatives amongst researchers and practitioners being Knowledge Discovery in Databases (KDD) [12] and Cross Industry Standard Process for Data Mining (CRISP-DM) [13]. While these process models are highly popular, they are either pursuing particular objectives or are focusing on a limited part of the overall process only [6]. None is specifically tailored to SML classification challenges nor

does any of them include the error estimation and deployment steps of a predictive model within an information system. Table 1 compares the most common models as to the full coverage of a SML process from model instantiation to model deployment.

Table 1. Comparison of different data mining process models regarding a holistic supervised machine learning process

Source	Model initiation	Model error estimation	Model deployment
Fayyad et al. [12] (KDD)	◐	◐	○
Chapman et al. [13] (Crisp-DM)	●	○	●
Witten et al. [14]	○	○	●
Cabena [15]	◐	○	○
Anand & Büchner [16]	●	○	○
Cios et al. [17]	◐	○	○
Brodley & Smyth [18]	●	●	○

○= Not addressed, ◐= Partially addressed, ●= Fully addressed

As the table shows, related work is very much focused on the process steps of the model initiation (like preprocessing and model training)—but usually misses out on the error estimation as well as the important final step to embed the generated models within an IS artifact (and thus to create an actionable analytics service). Furthermore, none of the existing processes is describing the data flow during the process, which is a crucial aspect for creating a well-performing SML model.

Therefore, in this research-in-progress paper we propose a holistic *process model* for SML classification—from problem to final deployment. Such an artifact can then be used for arbitrary *SML model* scenarios aiming at actionable SML analytics services, e.g. in predictive maintenance applications or any of the other scenarios mentioned before.

In order to contrast evidence from literature to current industry perception, we additionally conduct exploratory interviews with two experts that analyze data on a daily basis. They confirm the necessity of a fine-grained process model tailored to SML classification tasks and emphasize the importance of standardization in this context. According to them, currently established processes (e.g. KDD) are insufficient as a guideline for building a SML classification model that serves to derive knowledge out of data and is deployed for continuous use. Thus, insights from both the literature review and industry interviews confirm the lack of a holistic process model tailored to SML applications.

3 Suggestion & Development

Our goal is to design a process model that depicts the activities as well as the data flow throughout the initiation, error estimation and deployment of a SML classification process to predict a certain attribute from a large set of data. The process runs through three consecutive phases, and ends with a deployed SML classification model (figure 2). From a data perspective, we have to keep in mind that for any supervised learning a *ground truth data set* (where the targeted

attribute is known) is needed to train and test the model. To achieve this, the process starts with a well-defined problem that describes the (business) setting as well as the objective to be predicted. In the first phase, the *model initiation*, the *problem space is explored* to gain insights. Methods, like an exploratory data analysis [19], can be of use to find meaningful patterns and to identify the relevant data (features) that can later on be processed. Furthermore, the performance metrics are selected that are afterwards used to validate and test the model. Common metrics include F_β-score, ROC-AUC, sensitivity, recall, specificity, accuracy, Cohen's Cappa and others. The metrics are selected in light of the specific problem setting, e.g. reflecting classification error impacts.

After that, the *raw data* is gathered. Should the target attribute not already be included in the data, a manual *labelling* process is necessary. In that case, to ensure correctness of the ground truth data, it is advisable to categorize the data by more than one human assessor to minimize the manual classification bias [20]. Next, this data is cleaned and structured and further *preprocessed* (if necessary). Depending on the data input, fundamental preprocessing might be necessary. For instance, if text needs to be analyzed, common preprocessing techniques from the field of natural language processing (NLP) would be a n-gram generation [21] or stemming [22]. The result of this step is the *ground truth data*, which serves as the basis for the remaining process. The only remaining step is *choosing a classification algorithm.*

The second main phase is the *model error estimation*, where the goal is to estimate the expected performance of a model on unseen data. This is necessary, as a model selection without a previous error estimation cannot make meaningful estimations about performances and will result in too optimistic results [23]. As we aim at identifying a solid model, we also need to regard different *algorithm parameters*, which will characterize our machine learning model. A parameter, for instance, is the error term penalty in case of evaluating a support vector machine [24]. For each parameter a range of values has to be chosen (*search space*). Then, *data is split* to be handled in an outer and inner cross validation (*nested cross validation*). In the inner iteration, the *model is trained and validated* for different parameter sets towards its fit for the given problem. Various techniques for parameter tuning, such as a grid search, a bayesian optimization [25], a gradient-based optimization [26] or a random select [27] can be used. Ultimately, the previously defined metrics determine the *best performing parameters* on the inner iteration. After iterating through all possibilities (in case of a grid search), the best performing pair of parameters (on average) is then used to train on the whole data set from the inner fold and validate it on completely unseen data from the outer fold. Only by doing this, we can gain insight on how well the model would perform on new, unseen data—and how the results vary across different scenarios [28]. This process is repeated multiple times—depending on the amount of folds and runs from the outer cross validation. After all outer fold are iterated, we can calculate the mean, deviation and confidence interval for the different acquired performances from their validation on the outer folds. This result gives then insight about the expected performance of our model on

unseen data and how widely it varies—and therefore how stable the model is and how much it tends to over-fit [29].

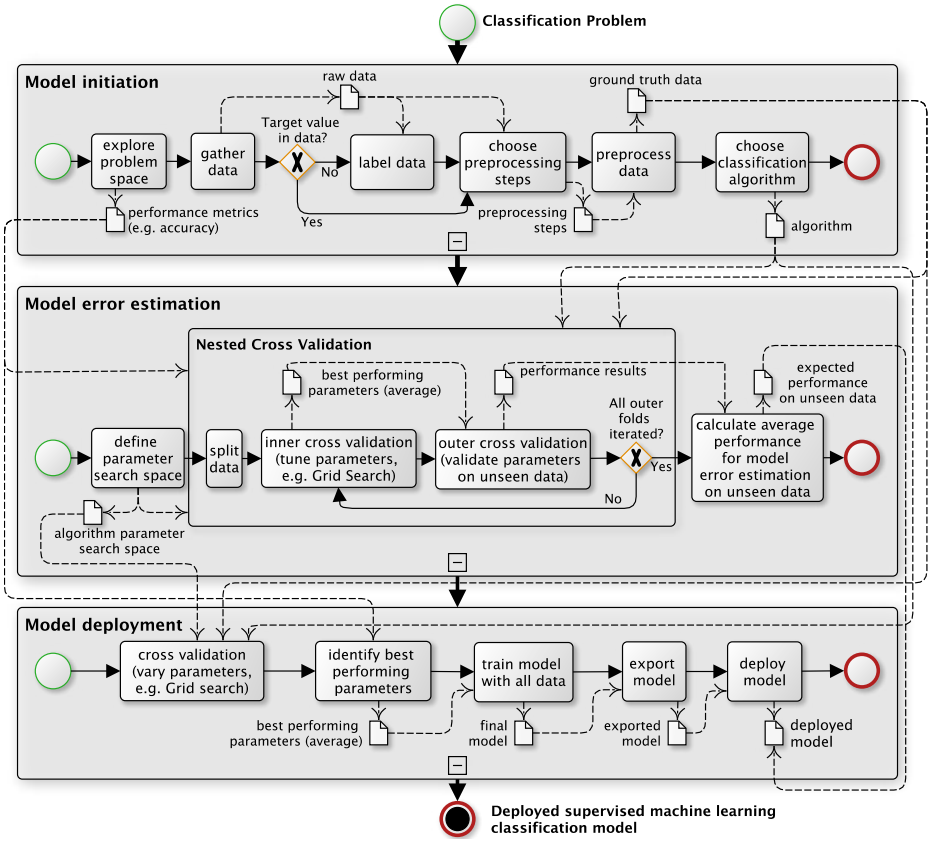

Fig. 2. End-to-end process model for SML classification including the data flow, divided into model initiation, error estimation and deployment.

The goal of the final *model deployment* phase is the generation, implementation and distribution of a built SML model—potentially as an analytics service—within an information system. First, we perform another *cross validation* with the identical parameter search space from the error estimation. We *identify the combination of parameters* which achieves the best results regarding our previously defined performance metrics. As data is always valuable and in most cases scarce, the complete data set is used to train the final machine learning classification model using the previously selected parameters. Then, an *export of the final model* (serialization) is needed to save the state of the model and the used preprocessing pipeline for further usage. Now, the serialized object can be included into a workflow, such as a connected web service, to predict the target value of new, incoming data. Hereby, data gets sent to the serialized object to be preprocessed and classified by the model. The *deployed SML classification*

model, including the preprocessing steps, is the final output of the third phase and the overall process.

4 Evaluation

The developed artifact is evaluated by performing an illustrative scenario [10] to show its suitability as an SML classification process model. As a real-world scenario, we want to predict the age-class (1-17, 18-24, 25+ years of age) of a Twitter user by applying NLP and SML to the user's tweets. This information would be valuable, e.g., to analyze demographics of trending topics or the automatic elicitation of customer needs [20].

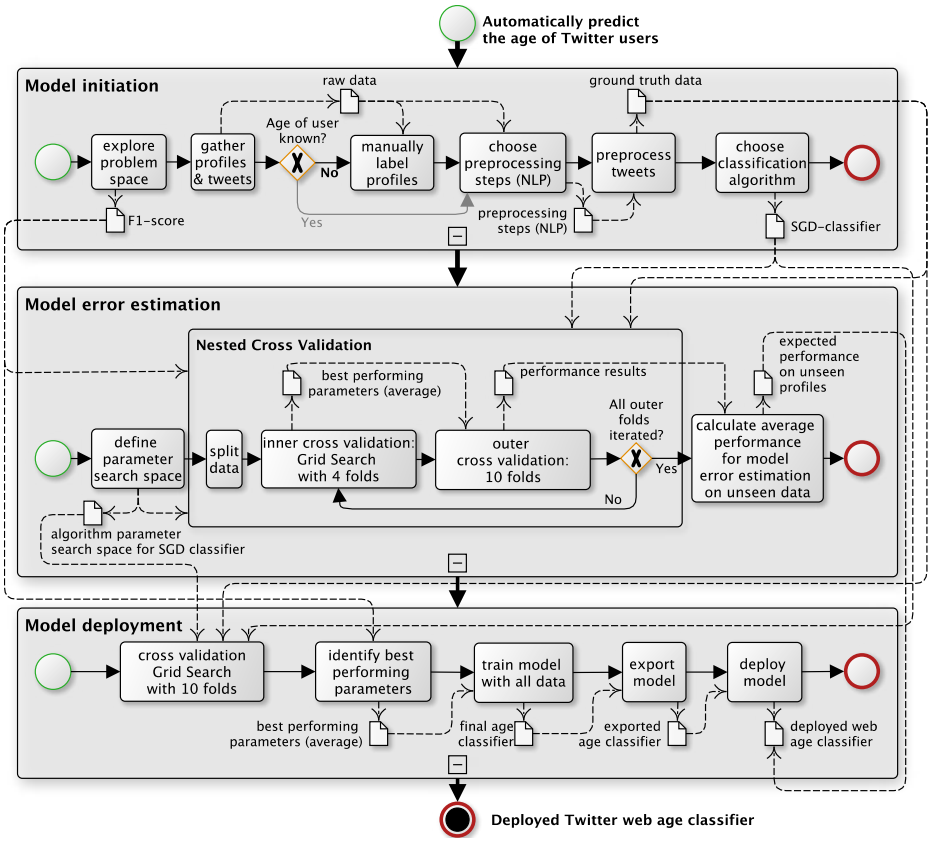

Fig. 3. Overview over the process that includes the three main sub-processes of solving a text mining problem.

As depicted in fig. 3, we follow the developed process to cope this challenge as an illustrative scenario. During the *model initiation*, we explore the problem space by comparing research about age classification and the feasibility of a tweet-based age classifier [30]. We define the F_1-score as our *performance metric*,

which represents a trade-off between recall and precision. After that, we use the official Twitter API[1] *to gather profiles and tweets* of Twitter users who have at least 20 tweets and who mention their age in the profile description. Manually, we sort out profiles that might be misleading, e.g. bots (structure & clean data). We then link the profiles to the corresponding age class (*manually label profiles*). For each profile we concatenate 20 tweets that represent a text with a corresponding age-class. This yields in a imbalanced data set with 781 categorized texts with a distribution of 305 (1-17) to 285 (18-24) to 191 (25+). We use Natural Language Processing (NLP) techniques as preprocessing [30], such as an n-gram processing, emoticon-count, hashtag-count and hyperlink-count, and preprocess the tweets. The output now represents our ground truth data set. Now we perform a pretest using several SML classifier algorithms and choose the Stochastic Gradient Descent classifier as the best performing algorithm.

During the *model error estimation* phase, we first define a range of parameters to tune the SML algorithm which defines our parameter search space. Now we perform a *nested cross validation* with an inner 4-fold cross validation and an outer 10-fold cross validation, to first tune and then validate the classifier parameters. As an output of the inner cross validation, we get the *best performing parameters*. Then, the outer cross validation outputs the *performance results* that we use to calculate the *expected performance on unseen data*. The mean is 47.47% F_1-score, a deviation of 5.05% and confidence interval of $[36.12; 57.04]$.

In a last step, the *model deployment*, we perform a *10-fold cross validation* with a Grid Search to *identify the best performing parameters* out of the parameter search space. The best run scores a F_1-score of 52.02% which lays inside the confidence interval identified during the model error estimation phase. Now all ground truth data is used to train the final model. This final text classifier is *exported and deployed* in a web service architecture. Providing a REST-API, concatenated tweets of Twitter users can be sent to the API, triggering the classifier to predict the age and return the the response. The deployed model now has an expected performance on unseen data derived during the model error estimation phase. The classifier is made publicly accessible as a web service through a user interface for demonstration[2]. With that, the age predicting SML classifier can be embedded in other applications to dynamically segment groups of users. As we tightly adhere to the developed process and succeed in building a text-based age classifier for Twitter users, we demonstrate that the developed artifact is suitable and helpful to guide both activities and data flow during the initiation, generation and deployment of a SML classification model.

5 Conclusion

We suggest and develop a holistic process model as an artifact to systematically build SML classification models that predict an attribute from a given dataset. It both attends to the specifics of a SML classification model and includes the

[1] https://dev.twitter.com/rest/public, last accessed on 23-02-2017.
[2] http://age-prediction.science, last accessed on 21-04-2017.

deployment into an information system for continuous use and actionability. In a first design cycle we run a first validation to test the suitability of the artifact via an illustrative scenario [11]: We build a text-based age classifier for Twitter users applying a step-by-step mapping to the developed process model.

Our future research will include additional evaluations of the artifact—in concrete case studies and in experimental benchmarks vs. established process models. We expect to generate more insights from the evaluations that will help us to further refine the artifact in future design cycles, e.g. by detailing sub-steps or providing decision guidance.

Thus, we contribute a first version of an artifact that will augment the inventory of concepts and methods in knowledge discovery. The managerial implications of a comprehensive process model for supervised machine learning are evident. As data availability and SML classification approaches soar in importance, a standardized and holistic process model is key to ensure flawless, high prediction quality and efficient SML classification models that can also be embedded in information systems for continuous support of decisions and actions.

Acknowledgements The authors would like to thank Björn Schmitz for his input and contributions to the process model.

References

1. Van der Aalst, W.M.: Data scientist: The engineer of the future. In: Enterprise Interoperability VI. Springer (2014) 13–26
2. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U.: The rise of big data on cloud computing: Review and open research issues. Information Systems **47** (2015) 98–115
3. Jensen, L.J., Bateman, A.: The rise and fall of supervised machine learning techniques (2011)
4. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine learning: An artificial intelligence approach. Springer Science & Business Media (2013)
5. Stokic, D., Scholze, S., Barata, J.: Self-learning embedded services for integration of complex, flexible production systems. In: IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society, IEEE (2011) 415–420
6. Kurgan, L.A., Musilek, P.: A survey of knowledge discovery and data mining process models. The Knowledge Engineering Review **21**(01) (2006) 1–24
7. Kuechler, B., Vaishnavi, V., Systems, C.I.: Theory Development in Design Science Research: Anatomy of a Research Project. Conference on Design Science Research in Information Systems and Technology (2007) pp. 1–15
8. Gregor, S., Hevner, A.R.: Positioning and Presenting Design Science Types of Knowledge in Science Research. MIS Quarterly **37**(2) (2013) 337–355
9. March, S., G., S.: Design and Natural Science Research on Information Technology. Decision Support Systems **15** (1995) 251–266
10. Peffers, K., Rothenberger, M., Tuunanen, T., Vaezi, R.: Design Science Research Evaluation. Design Science Research in Information Systems. Advances in Theory and Practice (2012) 398–410

11. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems **24**(3) (2007) 45–77
12. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The kdd process for extracting useful knowledge from volumes of data. Communications of the ACM **39**(11) (1996) 27–34
13. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: Crisp-dm 1.0 step-by-step data mining guide. (2000)
14. Witten, I., Frank, E., Hall, M., Pal, C.: Data Mining: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science (2016)
15. Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., Zanasi, A.: Discovering data mining: from concept to implementation. Prentice-Hall, Inc. (1998)
16. Anand, S.S., Büchner, A.G.: Decision support using data mining. Financial Times Management (1998)
17. Cios, K.J., Teresinska, A., Konieczna, S., Potocka, J., Sharma, S.: Diagnosing myocardial perfusion from pect bulls-eye maps-a knowledge discovery approach. IEEE Engineering in Medicine and Biology Magazine **19**(4) (2000) 17–25
18. Bradley, C.E., Smyth, P.: The process of applying machine learning algorithms. Applying Machine Learning in Practice IMLC-95 (1998)
19. Borcard, D., Gillet, F., Legendre, P.: Exploratory Data Analysis. Applied Spatial Data Analysis with R **2**(1999) (2008) 21–54
20. Kuehl, N., Scheurenbrand, J., Satzger, G.: NEEDMINING: IDENTIFYING MICRO BLOG DATA CONTAINING CUSTOMER NEEDS. In: Proceedings of the 24th European Conference of Information Systems, AIS (2016)
21. Pustejovsky, J., Stubbs, a.: Natural language annotation for machine learning. (2013)
22. Andrews, N.O., Fox, E.A.: Recent developments in document clustering. Citeseer (2007) 1–25
23. Cawley, G.C., Talbot, N.L.: On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research **11**(Jul) (2010) 2079–2107
24. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Lecture Notes in Computer Science **1398** (1998) 137–142
25. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian Optimization of Machine Learning Algorithms. Adv. Neural Inf. Process. Syst. 25 (2012) 1–9
26. Maclaurin, D., Duvenaud, D., Adams, R.P.: Gradient-based Hyperparameter Optimization through Reversible Learning. Proceedings of the 32nd International Conference on Machine Learning (2015) 1–10
27. Bergstra, J., Yoshua Bengio, U.: Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research **13** (2012) 281–305
28. Beleites, C., Salzer, R.: Assessing and improving the stability of chemometric models in small sample size situations. Analytical and Bioanalytical Chemistry **390**(5) (2008) 1261–1271
29. Varma, S., Simon, R.: Bias in error estimation when using cross-validation for model selection. BMC bioinformatics **7**(1) (2006) 91
30. Burger, J.D., Henderson, J., Kim, G., Zarrella, G.: Discriminating Gender on Twitter. Association for Computational Linguistics **146** (2011) 1301–1309