Economics and Management
ISSN: 2683-1325
Volume: XIX, Issue: 2, Year: 2022, pp. 1-11
DOI: 10.37708/em.swu.v19i2.1

FORECASTING CUSTOMER SUPPORT RESOLUTION TIMES THROUGH AUTOMATED MACHINE LEARNING

Anton A. Gerunov[1]

Received 4.11.2022, Accepted: 20.11.2022

Abstract

This article focuses on modeling and forecasting the resolution time of customer support tickets. To this end we leverage data from a process aware information system and compare manual training of several state-of-the-art benchmark models (neural network, regression, k-Nearest neighbors, random forest, and support vector machine) to automated model training using the H2O framework. The best performer among the automated machine learning models has much higher forecast accuracy than the benchmark models. This indicates that automated machine learning is a feasible way to approach process modeling problems and may be fruitfully utilized to forecast relevant process metrics.

Keywords: *customer support; resolution time; business process mining; prediction; automated machine learning; AutoML; H2O framework*
JEL Codes: *C44, C45, C53*

Introduction

Business activity in the modern organization is often structured along the lines of well-defined processes that aim to standardize and optimize common activities. Historically, it has been the realm of executives and management consultants to reimagine processes and activities in order to maximize their value. This hinged critically upon best practices and recourse to similar process optimization exercises (see Hammer & Champy, 1993 or Harika et al., 2021 for a more recent treatment). However, the growing preponderance of large volumes of data has enabled a more formal analysis of processes beyond the experience-based intuition. Data and event mining have been instrumental in empowering organizations, large and small, to automatically reconstruct their actual (as opposed to normative) processes and procedures and formally identify ways to improve them (dos Santos Garcia et al., 2019). This article takes this strand of literature one step

[1] Ph.D., D. Sc. and Associate Professor at the Faculty of Economics and Business Administration, Sofia University "St. Kliment Ohridski", 125 Tsarigrasko Shosse Blvd., Sofia 1113, e-mail: a.gerunov@feb.uni-sofia.bg, ORCID ID: 0000-0001-9357-7375

further by applying automated machine learning to predict the outcome of the customer support process in a specific organization.

We focus on investigating what drives extreme delays in IT customer support by taking recourse to data from a process-aware information system. This process is clearly crucial as it has direct interface with customers and can critically impinge on overall client loyalty, brand preference and the subsequent purchasing behavior. These in turn affect customer lifetime value and overall organizational profitability. The organization would clearly want short support turnaround times which not only increases customer satisfaction but also economizes on internal resources. It is thus of natural interest to predict which individual tickets (cases) are likely to be delayed and thus take corrective action. Advanced machine learning can be leveraged to achieve this task and we demonstrate how its automation can provide superior results over the standard opportunistic model selection.

The structure is as follows. Section two presents a short literature review, and section three outlines the data used and the proposed methodology. Section four outlines a few stylized fact about the data at hand, and section five applies automated machine learning (so-called AutoML) to try and predict process outcomes. Variable importance is used to investigate the relative contribution of different process drivers. Section six discusses the results, presents a few recommendations, and concludes.

Literature Review

Traditionally, process understanding and optimization has crucially depended on two main perspectives – the abstract modeling and optimization one, and the data driven one (see de Leoni et al., 2016 for further discussion). The former focuses on eliciting a normative description of the business process ("as it should be") and then identifying possible bottlenecks that can be reimagined. This goes very much in the stream of process reengineering thinking a la Hammer & Champy (1993). The latter approach – the data-driven one – relies heavily on collecting process data and identifying key drivers and blockers that can then be leveraged to increase performance in a tangible way. This analysis has more of a positive tinge – defining and improving processes as they actually occur ("as is").

The missing link between the two is the so-called process mining – a concerted effort to collect actual data from process instances that characterizes every step taken and elicit the process model from the data at hand (de Leoni et al., 2016, Amaral et al., 2018). In a modern organization many business processes are carried out completely or to a large extent in dedicated information systems. The data collection exercise then hinges critically upon accessing the collecting data from those transaction systems. This data are usually in the from of registering process steps, information flows, classification types and business rules implementations. More often than not, such data are found in the system logs but additional data may be brought to bear in the data ingestion and enrichment process.

Amaras et al. (2018) point that a large part of this research program is carried out by taking recourse to rather abstract process representations and calls for more concrete and practical ones. Van der Aalst (2016) provides a handbook for process discovery and optimization through event mining. This approach traditionally goes through a set of high-level steps, starting from gaining process understanding through data identification and collection, log and event mining, automated or expert-led optimization through reengineering, implementation, and ultimately – change management. For an overview of the process mining steps and ideas, the reader is pointed to van der Aalst (2022).

This approach has proven to work well and has been applied in a variety of settings and to a large set of problems such as measuring customer reactions to social media advertising (Boonjing & Pimchangthong, 2017), evaluating enterprise resource planning (ERP) systems (Pawełoszek, 2016) or general improvement of various business processes (Grisold et al., 2020). A mapping study of process mining techniques and applications can be found in dos Santos Garcia et al., (2019).

However, the full automation of process optimization remains an elusive task. An important part of this tasks is the automation of modelling and analysis that identifies process bottlenecks or process features that lead to exceptions, delays and failures. This article focuses on applying an automated machine learning framework to model the drivers behind excessive process delays in a typical client-facing business process – providing customer support.

Currently, automated machine learning is a new paradigm for fitting machine learning models. The traditional way to approach modeling is through what may be considered theory-informed opportunistic model fitting. It consists of the following. Once the analyst has an idea of the general type of problem that needs to be solved (i.e., is it a classification or a prediction/regression problem), then a number of models are iteratively fitted and investigated. While there is best practice and a number of references studies that explore the performance of a large number of alternative algorithms (see e.g. Fernandez-Delgado et al., 2014; Makridakis et al., 2020; Gerunov, 2022) this approach remains human-driven and opportunistic at heart.

The automated machine learning (AutoML) paradigm takes a different angle to the modeling exercise. He et al. (2021) define AutoML as a process of automatically constructing the machine learning pipeline on a limited computational budget. It is essentially the place where machine learning meets automation (Yao et al., 2018). AutoML starts by being fed a dataset and from then on it is able to automatically perform data normalization and preparation. After that a potentially very large number of models from a pre-selected list of algorithms is fitted and their parameters are fine-tuned. The best performing algorithms and parameters are automatically selected based on an error metric or an information criterion. The optimal model can then be used to outline process drivers through variable importance metrics and also generate useful predictions or classifications. The interested reader is referred to He et al. (2021) for a more detailed overview of AutoML.

The AutoML approach is practically implemented in a number of frameworks that easily run on traditional languages for statistical and analytic programming such as R and Python.

This article leverages one of the most mature and extensive AutoML framework to solve its modeling tasks – the H2O AutoML. A description of the framework – its concepts, ideas, methods, and implementations can be found in LeDell & Poirier (2020).

Data, Samples and Descriptive Statistics

To model customer support delays, we leverage data from Amaral et al. (2018). The data hails from a process-aware information system (PAIS) and contains standard markers, including type of incident, processor, handler, request IDs, etc. The dataset also includes a rich set of characteristics of the alerts themselves – incident status, activity, number of assignments and modifications, reporting channel, symptoms and logical place of occurrence, presence of an error message, and more. Ratings are also available for problem severity - strength of effect, urgency, priority, double check for priority, as well as closing characteristics – date and time, close code, close employee ID. The total number of those explanatory features in the original dataset is 36, spanning over 141,712 observations.

Amaral et al. (2018) focus on selecting the appropriate features to include in a predictive model, using the Annotated Transition Systems algorithm. They have three experiments with different selection approaches, finding that expert judgments outperform some but not all automated approaches. Additionally, the authors use mean absolute percentage error (MAPE) to measure the predictive accuracy of their approach, but do not report other indicators of predictive accuracy. We build on the results in this paper (Amaral et al., 2018) regarding the selection of predictors but also perform feature selection ourselves.

More concretely, we remove data that contains identifiers of various circumstances (agents, randomly generated request numbers, etc.), as they do not carry meaningful information. We additionally remove all variables that have a large number of missing observations (over 30%). There are seven such variables; five of those over 98% missing values, so their removal does not result in a significant loss of information. On the basis of the time stamps for the moment of creation of the ticket and the moment of its closing, we construct the target variable – the elapsed processing time. Data is further tested for missing values, with very few of those remaining. Wherever possible, missing values are imputed using multivariate imputation by chained equations, and whenever this is not possible – missing observations are removed. This leaves a final complete dataset of 17 predictors over 141,712 observations.

It is notable that most incidents in the array are of relatively advanced status, given that the status variable is coded from 0 to 7, with higher numbers indicating proximity to resolution (6 being "resolved" and 7 being "closed"). On average, a ticket is assigned once, but this can go up to 27 times. In practice, a very small fraction of tickets are reopened (only 2%), but the most serious tickets can reach up to 6 reopens. Even more indicative in this regard are the statistics on number of system modifications – although the average is relatively low ($\mu = 5.08$), the standard deviation is significant ($\sigma = 7.70$), which means that there are isolated cases with

particularly high activity on them. The highest number of ticket status updates we see in the database is 129.

The vast majority of incidents within this database (94%) fall outside the minimum service level (MSL) set out in the service level agreement (SLA). This highlights that tickets tend to concern extreme cases – both in terms of expectations and contractual obligations. Symptom codes and locations of occurrence and incident termination are useful number for machine-led modeling. Those are difficult to interpret by humans due to the high number of missing values, but can still be extremely useful information for automated algorithms when fitting predictive models. The vast majority of alerts are classified as medium impact strength and medium urgency. However, they were mostly classified as high priority incidents on a 4-point scale ($\mu = 2.98$).

A small percentage of tickets use the organization's knowledge base (15%), which may raise questions about the quality and usefulness of the knowledge base itself. An alternative explanation here would be that the vast majority of incidents are relatively trivial and do not require deep knowledge, making them suitable candidates for automation. The resilience of the business process and the information system serving it can be seen in the indicator for an error message generated – it is almost always 1 ($\mu \approx 1.00$), and very rarely the ticket is not associated with such a message ($\sigma = 0.02$). Looking at the skewness and kurtosis values of the variables under consideration, it is immediately obvious that they do not follow a normal distribution. This should be taken into account in subsequent modeling and thus methods that rely on data normality should be applied with caution.

Table no. 1 – Descriptive Statistics for IT Support Data

Variable	**Mean**	**Std. Dev.**	**Median**	**Min**	**Max**	**Skewness**	**Kurtosis**
Incident state	4.05	2.83	2	1	8	0.28	-1.68
Active	1.82	0.38	2	1	2	-1.70	0.89
Count of reassignments	1.11	1.74	1	0	27	3.22	18.26
Count of reopens	0.02	0.19	0	0	6	13.61	261.30
Count of system modifications	5.08	7.70	3	0	129	4.77	35.79
Outside Minimum Service Agreement	0.94	0.24	2	1	2	-3.67	11.48
Channel	4.00	0.12	4	1	5	-4.17	202.18
Location	138.34	55.56	143	2	249	-0.33	-0.92
Category	36.74	13.82	37	2	63	-0.17	-0.83
Sub-category	161.60	69.35	174	2	305	-0.50	0.03
Symptom code	398.26	169.78	491	2	609	-1.21	-0.05
Impact	2.01	0.23	2	1	3	0.42	15.70
Urgency	2.00	0.23	2	1	3	0.20	15.79
Priority	2.98	0.33	3	1	4	-2.78	20.51
Use of knowledge base	1.15	0.36	1	1	2	1.97	1.89
Priority confirmation	1.30	0.46	1	1	2	0.90	-1.20
Notification	1.00	0.02	1	0	1	41.41	1713.18

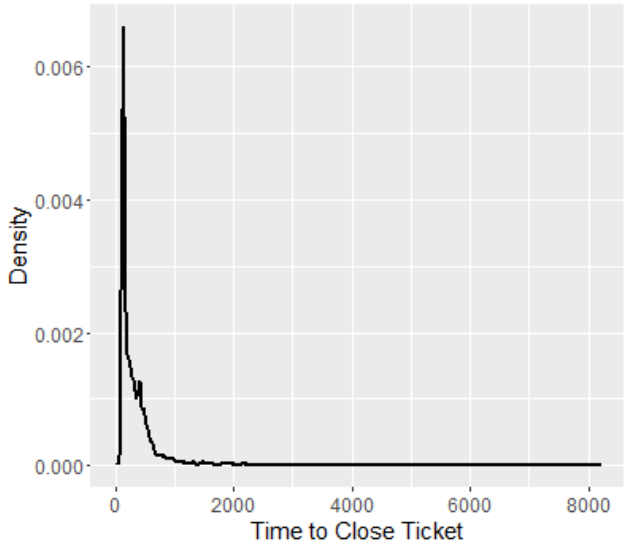
Closed code	6.52	1.73	6	1	17	1.24	9.20
Resolution time, y	400.73	650.97	217.9	0	8140.78	5.58	38.31

Source: Author's calculations based on data by Amaral et al. (2018)

Most tickets in the data set are of relatively advanced status, given that the status variable is coded from 0 to 7, with higher numbers indicating proximity to resolution (6 being "resolved" and 7 being "closed"). On average, a ticket is assigned once, but can be assigned up to 27 times. In practice, a very small fraction of alerts is reopened (only 2%), but the most serious support tickets can reach up to 6 reopens. Even more indicative in this regard are the statistics on new system modifications – although the average is relatively low ($\mu = 5.08$), the standard deviation is significant ($\sigma = 7.70$), which means that there are isolated cases with particularly high activity on them. The highest number of ticket modifications we see in the database is 129.

It is instructive to further investigate the distribution of the target variable – the amount of time it took to close a support ticket. The average time to process and close a ticket is about 400 hours, or 16.7 days. The average is largely driven by the extreme observations in the dataset, with the longest processing time being a whopping 8140.78 hours (339 days) and the shortest being under an hour. The large difference between the different incidents is also visible in the high values of the standard deviation ($\sigma = 650.97$) and the amount of time for resolution has a distribution far from normal.

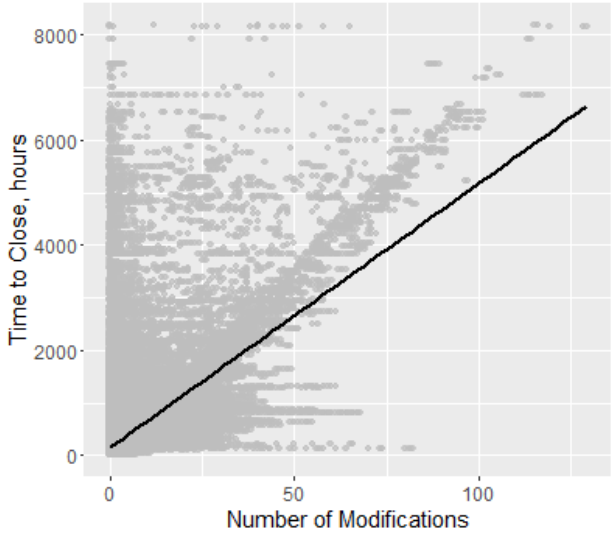
Figure no. 1 Distribution of Target Variable: Ticket Processing Time



Source: Author's visualization based on data by Amaral et al., 2018

In fact, the distribution remarkably resembles a lognormal one with a clearly pronounced peak on the left side and a long tail. Estimating a lognormal distribution on the positive values in the target variable (time to resolution), it seems that the best fitting lognormal has a log mean of 5.39 and a log standard deviation of 0.65. Here we observe a phenomenon that is very typical of the digital world – "winner-takes-all" dynamics underlined by a high degree of inequality. In our instance it is reflected in the fact that very few cases take the lion's share of time and presumably resources, while the vast majority of tickets are rapidly closed.

Figure no. 2 Correlation between number of support tickets modifications and resolution time



Source: Author's visualization based on data by Amaral et al., 2018

As a final step in the descriptive analysis, the variable correlations are investigated further. As a general observation, all but one correlation with the dependent variable (time to close) are of very low values. The one exception is the correlation between the number of system modifications and the resolution time – it is statistically significant and high with $r = 0.59$, $p < 0.005$. The link between those seems mostly linear, and robust (see Figure 2).

Modeling and Predicting Ticket Resolution time

The modeling approach used is a standard one. The overall dataset is randomly divided into two sub-samples – one that is used for training the models (approximately 80% of original data) and one that is used for testing model performances (approximately 20% of original data). Their samples sizes are $N_1 = 113{,}371$ and $N_2 = 28{,}341$, respectively. As a first pass we compare standard machine learning models that can be applied to this task

(see Gerunov, 2022) with automatically generated ones from the H2O AutoML framework. The standard benchmark models we fit are the following: Multiple Linear Regression, Artificial Neural Network, K-Nearest Neighbors, Random Forest, Support Vector Machine. Those benchmark models are then tested of the test set (i.e. data they have never been exposed to) and key accuracy metrics are calculated (the mean error, the root mean squared error, and the mean absolute error). Results are presented in Table 2.

Table no. 2 – Forecast Accuracy Metrics for Alternative Prediction Models

Machine Learning Algorithm	**Mean Error, ME**	**Root Mean Squared Error, RMSE**	**Mean Absolute Error, MAE**
Multiple Linear Regression	-2.39	480.46	214.15
Artificial Neural Network	398.42	759.98	398.42
k-Nearest Neighbors	5.95	351.56	134.57
Random Forest	-3.74	337.09	135.40
Support Vector Machine	77.71	495.42	190.61
AutoML Stacked Model: Ensemble of 100 models	0.04	274.87	111.36
AutoML Best Performer: Gradient Boosting Machine	-1.39	270.05	105.51

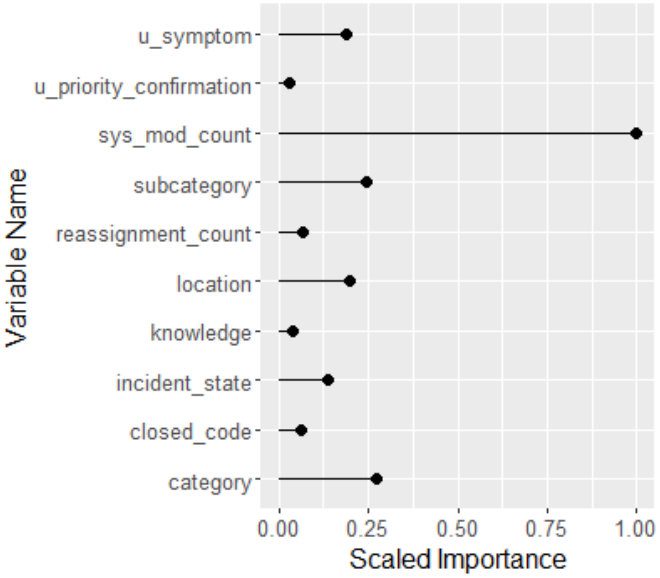
Source: Author's calculations

In contrast to this traditional modeling, we also leverage automated machine learning to fit and find the best model in a rigorous and standardized fashion without the need for expert interferences. The H2O framework automatically fits and evaluated models based on the following algorithms: Distributed Random Forest, Extremely Randomized Trees, Generalized Linear Model (GLM) with regularization, XGBoost Gradient Boosting Machine, H2O Gradient Boosting Machine, Multi-layer Artificial Neural Network, 2 Stacked Models, of which one contains all the models trained, and the other – the best-in-class model. The models are again fitted on the training set and test on the test set to generate out-of-sample accuracy metrics. Results can be seen in Table 2.

It is immediately obvious that AutoML models significantly outperform all other alternatives on all reported metrics. The best performing AutoML models is essentially a stacked ensemble model that combines the predictions of 100 different models, combining 55 neural network models, 2 deep random forest models, 42 gradient boosting machine models, and one general linear model. Its RMSE is by far the lowest, standing at 274.05 as compared to 337.09 (the lowest RMSE of non-AutoML models). The mean error and the mean absolute error metrics are similarly impressive. This result is not unexpected – there is a large body of literatures that shows that ensemble models outperform individual ones (see e.g. Idri et al., 2016 and references therein) but here the magnitude of the difference is notable.

The best performing single models in the AutoML exercise turns out to be a Gradient Boosting Machine (GBM) with 92 trees. Its RMSE is significantly lower that that of competing non-AutoML alternatives, and so are the values of its mean error and its mean absolute error. Remarkably, the GBM's performance is very close to that of the stacked ensemble models – the difference in RMSE is practically negligible – the former one stands at 274.87, while the latter – at 270.05. The mean errors are -1.39 and 0.04, respectively. Thus, for all practical purposes one can use a single GBM model instead of the stacked ensemble. This leads to ease of computation, decreased computational needs and significantly improve model explainability, all coming at the cost of negligible decrease in accuracy.

Figure no. 3 Top 10 Predictors for Resolution Time



Source: Author's visualization

Prediction drivers can be further investigated using the relative variable contributions to the forecast errors. The procedure involves removing every single variable and measuring the difference in average forecast errors. Those differences are then rescaled with the largest normalized to one. The intuition behind this is simple – if the removal of a variable significantly worsens the prediction made, then this variable must be an important one. Results from this exercise on the GBM model are presented in Figure 3.

The most important variable in the model is the system modifications count. The more modifications a ticket gets, the longer its processing time will be. This is then followed by the category of the indent, and the code for symptoms. The category and location of the incident are the remaining variables that drive our predictions. It seems that

6 out the 17 variables under study hold the lion's share of predictive power. All effects are in the expected directions with more critical incidents taking place in specific logical locations being connected with longer resolution times. Trivial tickets get the fastest resolution, and are thus rarely updated, modified, and classified as priority ones.

Discussion and Conclusion

This article has investigated the possibility for automated predictions of business process outcomes. We have taken a classical process found in many contemporary organizations – customer support – and have attempted to model its resolution times. Results are encouraging: automated forecasting models outperform traditional benchmarks, showing that automated prediction is viable. Knowing this, business may choose to increase the level of process automation, delegating it to machine learning algorithms to identify and flag instances with slow resolution times and bring notifications or even undertake actions to speed them up. This holds the potential of both decreasing operational costs as fewer analysts and managers will be needed for monitoring and control, as well as improving service and customer loyalty and retention. The latter may have positive spillovers into actual revenue. The promise of AutoML clearly goes beyond the process investigated here – it can be leveraged across the full spectrum of business processes to make them smarter and more efficient, thus further driving the digital transformation of the organization. Further research will likely elucidate the applications of this approach to analyzing and modeling business activities well beyond customer support and investigate the actual implementations of AutoML in practice.

REFERENCES

Amaral, C. A., Fantinato, M., Reijers, H. A., & Peres, S. M. (2018). Enhancing Completion Time Prediction Through Attribute Selection. In *Information Technology for Management: Emerging Research and Applications* (pp. 3-23). Springer, Cham.

Boonjing, V., & Pimchangthong, D. (2017). Data mining for positive customer reaction to advertising in social media. In *Information Technology for Management. Ongoing Research and Development* (pp. 83-95). Springer, Cham.

De Leoni, M., van der Aalst, W. M., & Dees, M. (2016). A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, *56*, 235-257.

dos Santos Garcia, C., Meincheim, A., Junior, E. R. F., Dallagassa, M. R., Sato, D. M. V., Carvalho, D. R., ... & Scalabrin, E. E. (2019). Process mining techniques and applications–A systematic mapping study. *Expert Systems with Applications*, *133*, 260-295.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems?. *The journal of machine learning research*, *15*(1), 3133-3181.

Gerunov, A. A. (2022). Performance of 109 Machine Learning Algorithms across Five Forecasting Tasks: Employee Behavior Modeling, Online Communication, House Pricing, IT Support and Demand Planning. *Economic Studies journal*, (2), 15-43.

Grisold, T., Mendling, J., Otto, M., & vom Brocke, J. (2020). Adoption, use and management of process mining in practice. *Business Process Management Journal*.

Hammer, M., & Champy, J. (1993). Business process reengineering. *London: Nicholas Brealey*, *444*(10), 730-755.

Harika, A., Sunil Kumar, M., Anantha Natarajan, V., & Kallam, S. (2021). Business process reengineering: issues and challenges. In *Proceedings of Second International Conference on Smart Energy and Communication* (pp. 363-382). Springer, Singapore.

He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, *212*, 106622.

Idri, A., Hosni, M., & Abran, A. (2016). Systematic literature review of ensemble effort estimation. *Journal of Systems and Software*, 118, 151-175.

LeDell, E., & Poirier, S. (2020, July). H2o automl: Scalable automatic machine learning. In *Proceedings of the AutoML Workshop at ICML* (Vol. 2020).

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*(1), 54-74.

Pawełoszek, I. (2016). Data mining approach to assessment of the ERP system from the vendor's perspective. In *Information technology for management* (pp. 125-143). Springer, Cham.

van der Aalst, W. M. (2022). Process mining: a 360 degree overview. In *Process Mining Handbook* (pp. 3-34). Springer, Cham.

Van Der Aalst, W. M., & Dustdar, S. (2012). Process mining put into context. *IEEE Internet Computing*, *16*(1), 82-86.

van der Aalst, W.M.P. (2016). *Process Mining - Discovery, Conformance and Enhancement of Business Processes*, 2nd Edition. Springer: Heidelberg.

Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y. F., Tu, W. W., ... & Yu, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*.