

Prediction of Maintenance Equipment Failures Using Automated Machine Learning

Luís Ferreira^{1,2}[0000-0002-4790-5128], André Pilastrri¹[0000-0002-4380-3220],
Vítor Sousa³, Filipe Romano³, and Paulo Cortez²[0000-0002-7991-2090]

¹ EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute,
Guimarães, Portugal

{luis.ferreira, andre.pilastrri}@ccg.pt

² ALGORITMI Centre, Dep. Information Systems, University of Minho, Guimarães,
Portugal

pcortez@dsi.uminho.pt

³ Valuekeep, Braga, Portugal

{vitor.sousa, filipe.romano}@valuekeep.com

Abstract. Predictive maintenance is a key area that is benefiting from the Industry 4.0 advent. Recently, there have been several attempts to use Machine Learning (ML) in order to optimize the maintenance of equipments and their repairs, with most of these approaches assuming an expert-based ML modeling. In this paper, we explore an Automated Machine Learning (AutoML) approach to address a predictive maintenance task related to a Portuguese software company. Using recently collected data from one of the company clients, we firstly performed a benchmark comparison study that included four open-source modern AutoML technologies to predict the number of days until the next failure of an equipment and also determine if the equipments will fail in a fixed amount of days. Overall, the results were very close among all AutoML tools, with AutoGluon obtaining the best results for all ML tasks. Then, the best AutoML predictive results were compared with a manual ML modeling approach that used the same dataset. The results achieved by the AutoML approach outperformed the manual method, thus demonstrating the quality of the automated modeling for the predictive maintenance domain.

Keywords: Automated Machine Learning · Predictive Maintenance · Supervised Learning

1 Introduction

The Industry 4.0 phenomenon allowed companies to focus on the analysis of historical data to obtain useful insights. In particular, predictive maintenance is a crucial application area that emerged from this context, where the goal is to optimize the maintenance and repair process of equipments through the usage of Machine Learning (ML) algorithms [17]. Indeed, some ML studies try to anticipate the failure of equipments (typically, manufacturing machines), aiming

to reduce the costs of repairs [4, 6, 11, 16]. Other approaches [2, 3, 5, 19] use ML algorithms to predict the behavior of the manufacturing process.

Despite all potential Industry 4.0 benefits, many organizations do not currently apply ML to enhance maintenance activities. And those who do rely mostly on data science experts, the ML models are tuned manually, often requiring a large number of trial-and-error experiments. In effect, we have only found one study that applied AutoML for the maintenance domain [18]. Yet, we note that such study only used synthetic data, which might not reflect the complexities of real industrial maintenance data. In contrast with the “traditional” ML expert design approach, in this paper we apply Automated Machine Learning (AutoML), aiming to automate the ML modeling phase and thus reduce the data to maintenance insights process cycle. Moreover, we apply AutoML using real-world data, collected from the client of a Portuguese software company in the area of maintenance management.

The AutoML was explored for two specific prediction tasks: the number of days until an equipment fails and if the equipments will fail in a fixed number of days. We designed a large set of computational experiments to assess the AutoML predictive performance of four open-source tools. To provide a baseline comparison, we also measure the best AutoML results with a manual ML modeling that was made previously by one of the company’s professionals. The comparison clearly favors the AutoML results, thus attesting the potential of the AutoML approach for the predictive maintenance domain.

2 Materials and Methods

2.1 AutoML Tools

In this article, we apply and compare four modern open-source AutoML tools, based on a recent benchmark study performed in [9]. In order to achieve a more fair comparison, we did not tune the hyperparameters of the AutoML tools. Table 1 summarizes the main characteristics of the four explored AutoML tools:

Table 1. Description of the AutoML tools.

Tool	Framework API		Version
AutoGluon	Gluon	Python	0.2.0
H2O AutoML	H2O	Java, Python, R	3.32.1.3
rminer	rminer	R	1.4.6
TPOT	Scikit-Learn	Python	0.11.7

- **AutoGluon** is an AutoML toolkit based on the Gluon framework [1]. In this work, we only considered the tabular data module, which runs several algorithms and returns a Stacked Ensemble with multiple layers [8].

- **H2O AutoML** is the AutoML module from the H2O framework. H2O AutoML runs several algorithms from H2O and two Stacked Ensembles, one with the best ML model of each family and another with all models [10, 13].
- **rminer** is a library for the R programming language, focused on facilitating the usage of ML algorithms [7]. Rminer also provides AutoML functions that can be highly customized. In this paper, we used the "automl3" template⁴, which runs several ML algorithms and one Stacked Ensemble.
- **TPOT** is a Python AutoML tool that uses Genetic Programming to automate several phases of the ML workflow [12, 15]. It uses the Python Scikit-Learn framework to produce ML pipelines.

2.2 Data

The provided data has a large number of datasets related to predictive maintenance, which are detailed in Fig. 1.

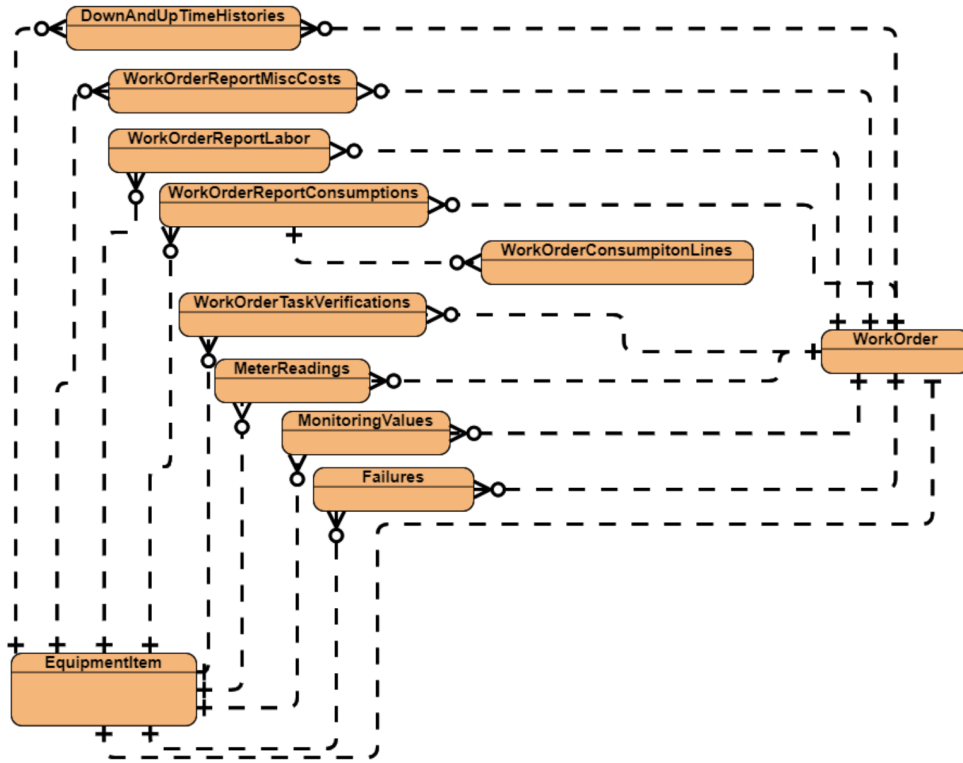


Fig. 1. Entities and relationships between the datasets.

⁴ <https://CRAN.R-project.org/package=rminer>

For the context of this work, we assume a tabular dataset composed of the aggregation of several attributes from each entity. Overall, the data includes 2,608 records and 21 input attributes. Each record represents an action (e.g., a work order) related to one of the company’s equipments (e.g., industrial machine). Each record includes diverse inputs attributes, such as the tasks performed by the machine, consumption of material, and meter readings.

The data also includes five target variables for regression or binary classification tasks. The regression task target (attribute DaysToNextFailure) describes the number of days between that record and the failure of the respective equipment. As for the binary classification targets (attributes FailOn x Days), these describe if the equipment will fail or not in a certain amount of days (e.g., in three days).

Table 2 details the input and output variables (**Attribute**), their description (**Description**), data type (**Type**), number of levels (**Levels**), domain values (**Domain**), and example values from one of the records (**Example**). Half (12) of the 21 input attributes are categorical. Among these, most present a low cardinality (e.g., RecordType, Brand). However, some of the attributes present a very high cardinality (e.g., Part).

2.3 Data Preprocessing

Since several data attributes are of type String, which is not accepted by some AutoML tools, we opted to encode all String attributes into numerical types. For the String attributes that presented a low cardinality (five levels or less), we applied the known One-Hot encoding. Since this method creates one binary column for each level of the original attribute, we applied a different transformation for the columns with a higher cardinality.

Indeed, for the categorical variables with more than five levels, we used the Inverse Document Frequency (IDF) technique [14]. This method is used to convert a categorical column into a numerical column of positive values, based on the frequency of each level of the attribute. IDF uses the function $f(x) = \log(n/fx)$, where n is the length of x and fx is the frequency of x . The benefit of IDF, when compared with One-Hot Encoding, is that the IDF technique does not generate new columns, which is useful for attributes with high cardinality (e.g., the attribute Part has 161 levels).

The remaining attributes (of Integer and Float types) were not altered because most AutoML tools already apply preprocessing techniques to the numerical columns (e.g., normalization, standardization). After applying the transformations, the final dataset had 42 inputs and 5 target columns.

2.4 Evaluation

In order to evaluate the results from the AutoML tools, we adopted a similar approach to the benchmark developed in [9]. For every predictive experiment, we divided the dataset into 10 folds for an external cross-validation and adopted an internal 5-fold cross-validation (i.e., over the training data) to select the best

Table 2. Description of the equipment maintenance dataset attributes.

Attribute	Description	Type	Levels	Domain	Example
RecordType	Type of record	String	5	-	Failure
Brand	Brand of the equipment	String	2	-	Rossi
WOType	Type of work order	String	4	-	Corrective
PriorityLevel	Priority Level of the work order	String	4	-	Urgent
Responsible	Responsible for the work order	String	3	-	R4
Employee	Employee that performed the action	String	12	-	E100
TotalTime	Duration of the action (in hours)	Float	17	[0, 8]	8
Quantity	Consumption quantity	Float	32	[0, 300]	90
Part	Part that was consumed	String	161	-	T-1073
Meter	Meter associated to meter reading	String	11	-	L-0002
MeterCumulativeReading	Cumulative reading of meter	Float	1477	[0, 73636]	22767
IncrementValue	Increment compared to last reading	Float	475	[0, 54570]	168
MaintenancePlan	Maintenance Plan associated to task	String	5	-	P-000011
Task	Executed task	String	5	-	T-0001
AssetWithFailure	Identification of the equipment	String	15	-	A577
ParentAsset	Parent equipment of <i>AssetWithFailure</i>	String	11	-	LINHA2
Day	Day of the month of the record	Integer	31	[1, 31]	4
DayOfWeek	Day of the week of the record	Integer	7	[1, 7]	6
Month	Month of the record	Integer	12	[1, 12]	2
Year	Year of the record	Integer	6	[2015, 2019]	2019
DaysAfterPurchase	Age of the equipment (in days)	Integer	852	[0, 6309]	4479
DaysToNextFailure	Number of Days until the next failure of the equipment	Integer	1015	[0, 1550]	3
FailOn3Days	Indication whether the equipment will fail in the next 3 days	Integer	2	{0,1}	1
FailOn5Days	Indication whether the equipment will fail in the next 5 days	Integer	2	{0,1}	1
FailOn7Days	Indication whether the equipment will fail in the next 7 days	Integer	2	{0,1}	1
FailOn10Days	Indication whether the equipment will fail in the next 10 days	Integer	2	{0,1}	1

algorithm and hyperparameters (executed automatically by the AutoML tools). To evaluate the test set (from external 10-fold validation) predictions we used the Mean Absolute Error (MAE) ($\in [0.0, \infty[$, where 0.0 represents a perfect model) for the regression task and the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) analysis ($\in [0.0, 1.0]$, where 1.0 indicates an ideal classifier) for the binary classification targets. We also used MAE and AUC for the internal validation, responsible for choosing the best ML model.

For all four AutoML tools we defined a maximum training time of one hour (3,600 seconds) and an early stopping of three rounds, when available. The maximum time of one hour was chosen since it is the default value for most of the AutoML tools. We computed the average of the evaluation measures, computed on the test sets of the 10 external folds, to provide an aggregated value. Additionally, we use confidence intervals based on the t -distribution with 95% confidence to verify the statistical significance of the experiments. In order to identify the

best results for each target, we choose the AutoML tool that had the best average predictive performance (with maximum precision of 0.01). All experiments were executed using an Intel Xeon 1.70GHz server with 56 cores and 64GB of RAM.

3 Results

All the experiments were implemented in Python or R (when the tool did not have a Python API) using the AutoML libraries detailed in Section 2.1. For each AutoML tool, we executed five experiments, one for each target variable (DaysToNextFailure and FailOn x Days). Table 3 shows the average external test scores for all 10 folds and the respective confidence intervals (near the \pm symbol).

Table 3. Average results obtained by the AutoML tools (best values in **bold**).

Target	Measure	AutoGluon	H2O	AutoML	rminer	TPOT
DaysToNextFailure	MAE	4.95 \pm 0.57	5.53 \pm 0.62	8.89 \pm 0.75	7.05 \pm 0.57	
FailOn3Days	AUC	0.98 \pm 0.02	0.98 \pm 0.01	0.95 \pm 0.05	0.97 \pm 0.03	
FailOn5Days	AUC	0.97 \pm 0.02	0.96 \pm 0.03	0.93 \pm 0.04	0.96 \pm 0.02	
FailOn7Days	AUC	0.98 \pm 0.01	0.98 \pm 0.01	0.97 \pm 0.03	0.98 \pm 0.01	
FailOn10Days	AUC	0.99 \pm 0.01	0.98 \pm 0.01	0.98 \pm 0.02	0.99 \pm 0.01	

Overall the best AutoML tool was AutoGluon, which produced the highest AUC values for the binary classification tasks and the lowest MAE value for the regression task. For the regression task (DaysToNextFailure), besides AutoGluon, the best AutoML tools were H2O AutoML, TPOT, and rminer. The maximum predictive difference was 3.94 points (days). As for the binary classification, the predictive test set results are more similar: maximum difference of 3 *percentage points (pp)* for FailOn3Days, 4 *pp* for FailOn5Days, 1 *pp* for FailOn7Days, and 1 *pp* for FailOn10Days. AutoGluon was the best tool for all four binary classification targets, followed by H2O AutoML and TPOT (best in two targets each).

Finally, we compare the best AutoML results for each target with the best result achieved by a human ML modeling (held before this study). For each target, Table 4 shows the best predictive result and the respective AutoML tool in rounded brackets. For each AutoML tool, we also show the algorithm that was most often the leader, across the external folds. For the human modeling, we show the best obtained result and the used algorithm (also in rounded brackets).

It should be noted that the human modeling used a distinct preprocessing procedure since it applied the One-Hot encoding to all categorical attributes (and not IDF for the high cardinality ones, as we adopted for the AutoML tools). Nevertheless, the comparison clearly favors the AutoML results for all

Table 4. Comparison between the best AutoML result and human ML modeling result for each target (best values in **bold**).

Target	Measure	Best Results	
		AutoML	Human Modeling
DaysToNextFailure	MAE	4.948 (AutoGluon: Ensemble)	68.361 (Random Forest)
FailOn3Days	AUC	0.979 (H2O AutoML: GBM)	0.500 (Random Forest)
FailOn5Days	AUC	0.971 (AutoGluon: Ensemble)	0.529 (Random Forest)
FailOn7Days	AUC	0.982 (TPOT: Random Forest)	0.581 (Random Forest)
FailOn10Days	AUC	0.988 (AutoGluon: Ensemble)	0.563 (Random Forest)

GBM - Gradient Boosting Machine.

predicted target variables. In particular, for the binary classification task, the human modeling achieved only slightly better results than a random model, while all AutoML tools achieved results that can be considered excellent (e.g., AUC higher than 0.90). For regression, the human modeling achieved an average error of 68.36 days, while the highest MAE obtained by the AutoML tools was 8.89 (achieved by rminer).

4 Conclusions

Predictive maintenance is a key industrial application that is being increasingly enhanced by the adoption of ML. Yet, most ML related works assume an expert ML model design that requires manual effort and time. In this paper, we explore the potential of AutoML to automate predictive maintenance ML modeling. We used real-world data provided by a Portuguese software company within the domain of maintenance management to predict equipment malfunctions.

Our goal was to anticipate failures from several types of equipments (e.g., industrial machines), using two ML tasks: regression - to predict the number of days until the next failure of the equipment; and binary classification - to predict if the equipment will fail in a fixed amount of days (e.g, in three days). For the ML modeling and training, we used four recent state-of-the-art Automated Machine Learning (AutoML) tools: AutoGluon, H2O AutoML, rminer, and TPOT.

Several computational experiments were held, assuming five predictive tasks (one regression and four binary classifications). For all ML tasks, AutoGluon presented the best average results among the AutoML tools. The AutoML results were further compared with a human ML design, performed previously by a professional of the Portuguese company. The comparison favored all AutoML tools, which provided better average results than the manual approach by a large margin. These results confirm the potential of the AutoML modeling, which

can automatically provide high quality predictive models. This is particularly valuable for the predictive maintenance domain since industrial data can arise with a high velocity, thus the predictive models can be dynamically updated through time, reducing the effort of the data analysis.

In future work, we intend to perform experiments with more AutoML tools and from the domain of predictive maintenance datasets. In particular, we intend to experiment with AutoML technologies that can automatically perform feature engineering and selection tasks.

Acknowledgments

This work was executed under the project Cognitive CMMS - Cognitive Computerized Maintenance Management System, NUP: POCI-01-0247-FEDER-033574, co-funded by the Incentive System for Research and Technological Development, from the Thematic Operational Program Competitiveness of the national framework program - Portugal2020.

References

1. Auto-Gluon: AutoGluon: AutoML Toolkit for Deep Learning — AutoGluon Documentation 0.2.0 documentation (2021), <https://auto.gluon.ai/>
2. Ayvaz, S., Alpay, K.: Predictive aintenance system for production lines in manufacturing: A machine learning approach using iot data in real-time. *Expert Systems with Applications* **173**, 114598 (2021), <https://doi.org/10.1016/j.eswa.2021.114598>
3. Butte, S., Prashanth, A., Patil, S.: Machine learning based predictive maintenance strategy: a super learning approach with deep neural networks. In: 2018 IEEE Workshop on Microelectronics and Electron Devices (WMED). pp. 1–5. IEEE (2018)
4. Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., da Piedade Francisco, R., Basto, J.P.T.V., Alcalá, S.G.S.: A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.* **137** (2019). <https://doi.org/10.1016/j.cie.2019.106024>
5. Çınar, Z.M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., Safaei, B.: Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability* **12**(19), 8211 (2020)
6. Cline, B., Niculescu, R.S., Huffman, D., Deckel, B.: Predictive maintenance applications for machine learning. In: 2017 annual reliability and maintainability symposium (RAMS). pp. 1–7. IEEE (2017)
7. Cortez, P.: Data mining with neural networks and support vector machines using the r/rminer tool. In: *Industrial conference on data mining*. pp. 572–583. Springer (2010)
8. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A.: Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505* (2020), <https://arxiv.org/abs/2003.06505>
9. Ferreira, L., Pilastri, A., Martins, C.M., Pires, P.M., Cortez, P.: A Comparison of AutoML Tools for Machine Learning, Deep Learning and XGBoost. In: *Int. Joint Conference on Neural Networks, IJCNN 2021, July*. IEEE (2021)

10. H2O.ai: H2O AutoML (2021), <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>, h2O version 3.32.1.3
11. Kanawaday, A., Sane, A.: Machine learning for predictive maintenance of industrial machines using iot sensor data. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). pp. 87–90. IEEE (2017)
12. Le, T.T., Fu, W., Moore, J.H.: Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics* **36**(1), 250–256 (2020)
13. LeDell, E., Poirier, S.: H2O AutoML: Scalable automatic machine learning. 7th ICML Workshop on Automated Machine Learning (AutoML) (July 2020), https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf
14. Matos, L.M., Cortez, P., Mendes, R., Moreau, A.: A comparison of data-driven approaches for mobile marketing user conversion prediction. In: Jardim-Gonçalves, R., Mendonça, J.P., Jotsov, V., Marques, M., Martins, J., Bierwolf, R.E. (eds.) 9th IEEE International Conference on Intelligent Systems, IS 2018, Funchal, Madeira, Portugal, September 25-27, 2018. pp. 140–146. IEEE (2018). <https://doi.org/10.1109/IS.2018.8710472>
15. Olson, R.S., Urbanowicz, R.J., Andrews, P.C., Lavender, N.A., Kidd, L.C., Moore, J.H.: Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I, chap. Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pp. 123–137. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-31204-0_9
16. Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., Loncarski, J.: Machine learning approach for predictive maintenance in industry 4.0. In: 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, MESA 2018, Oulu, Finland, July 2-4, 2018. pp. 1–6. IEEE (2018). <https://doi.org/10.1109/MESA.2018.8449150>
17. Silva, A.J., Cortez, P., Pereira, C., Pilastrri, A.: Business analytics in industry 4.0: A systematic review. *Expert Systems* p. e12741. <https://doi.org/https://doi.org/10.1111/exsy.12741>
18. Tornede, T., Tornede, A., Wever, M., Mohr, F., Hüllermeier, E.: Automl for predictive maintenance: One tool to RUL them all. In: IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning, ITEM 2020, Belgium, September 14-18, 2020, Revised Selected Papers. *Communications in Computer and Information Science*, vol. 1325, pp. 106–118. Springer (2020). https://doi.org/10.1007/978-3-030-66770-2_8
19. Vazan, P., Janikova, D., Tanuska, P., Kebisek, M., Cervenanska, Z.: Using data mining methods for manufacturing process control. *IFAC-PapersOnLine* **50**(1), 6178–6183 (2017)