Integration of an Explainable Predictive Process Monitoring System into IBM Process Mining Suite (Extended Abstract)

Riccardo Galanti*†, Massimiliano de Leoni†, Alan Marazzi*,
Giacomo Bottazzi*, Massimiliano Delsante*, Andrea Folli*

*IBM, Bologna, Italy, †University of Padua, Padua, Italy
{Riccardo.Galanti,Alan.Marazzi,Giacomo.Bottazzi,Massimiliano.Delsante,Andrea.Folli}@ibm.com, deleoni@math.unipd.it

I. Introduction

Predictive process monitoring aims to forecast the running process instances with the purpose of timely signalling those that require special attention (those that may take too long, cost too much, etc.). Within the field of Process Mining, this problem has received significant attention in the last years, yielding a large body of techniques capable to give accurate predictions [1].

However, an industrial application of predictive monitoring requires to build trust among process participants, who otherwise would be reluctant to believe the forecasts and adopt this type of predictive systems [2], [3]. An important step towards building trust is to provide users with explanations of the reasons why a given process execution is predicted to behave in a certain way [3], [4].

This paper reports on the introduction of explainable predictive process monitoring functionalities into the IBM Process Mining Suite.[1] After training the prediction model, the IBM Process Mining Suite performs a prediction of the completion time and cost for each running case. The prediction returned for each running case is also accompanied by a set of explanations of the attributes that influence these predictions and how. Each explanation is of form *a*=*x* to indicate that the prediction is driven by the fact that the case attribute a took on value x. Explanations for all running cases are also grouped and shown through a bar chart to highlight the most predominant.

II. Overview

The implementation of explainable predictive monitoring within the IBM Process Mining suite builds on the explanation framework discussed in [5], which is based on SHAP [6][2]. SHAP is based on the strong theoretical foundation of the original game theory approach to explain the variables that contribute to the predictions, and it is independent of the specific prediction technique by nature, as opposite to attention-based mechanisms, which only apply to a neural network [7]. To provide further evidence of this, the original prototype in [5] built on training LSTM models, whereas the implementation within the IBM software uses Catboost, a high-performance open source framework for gradient boosting on decision trees [8]. The choice of replacing LSTM models with Catboost is motivated by the fact that Catboost reduces the training time of ca. 20-30 times in all the experiments that we carried out, while returning models with similar accuracy.

The back-end of the predictive monitoring is based on the Azure infrastructure. This enables to deploy the technique in the cloud and develop a whole system around it. The system is in charge of processing requests coming from the IBM Process Mining suite, preparing a computing instance to execute our framework, and deliver the results back.[3] In particular, it can handle and process multiple requests coming from different users and, in case a customer requests it, multiple compute instances can be easily provided by allocating new clusters, enabling to scale on demand. The system has been tested to work with datasets up to 10 million of events.

Figure 1 shows a screenshot of the *Analytics Dashboard* within the IBM Process Mining suite for prediction of the total time of cases, namely the time necessary to complete a case. The use case presented here is based on a process executed at an Italian Banking Institution. The process deals with the closure of customer's accounts, which may be requested by the customer or by the bank, for several reasons. It uses event data consisting of 730336 events belonging to 116566 cases.

The upper-left corner reports on general process statistics, such as the number of running cases and the average case total time (here labeled as *Completed Time*) and cost. The bottom-right corner lists the running cases, each associated with the case identifier, and the last performed activity; since this dashboard refers to the process total time, each case is also associated with the elapsed time, the expected total time as forecasted by the predictive monitor, and its difference wrt. the average completion time, here also named as target. When one clicks on a specific running case (e.g. with id

[1] IBM Process Mining – https://www.ibm.com/cloud/cloud-pak-for-business-automation/process-mining

[2] Welcome to the SHAP documentation – https://shap.readthedocs.io

[3] A tutorial and a video showing how to use the tool can be found at https://github.com/PyRicky/explainable_predictive_system

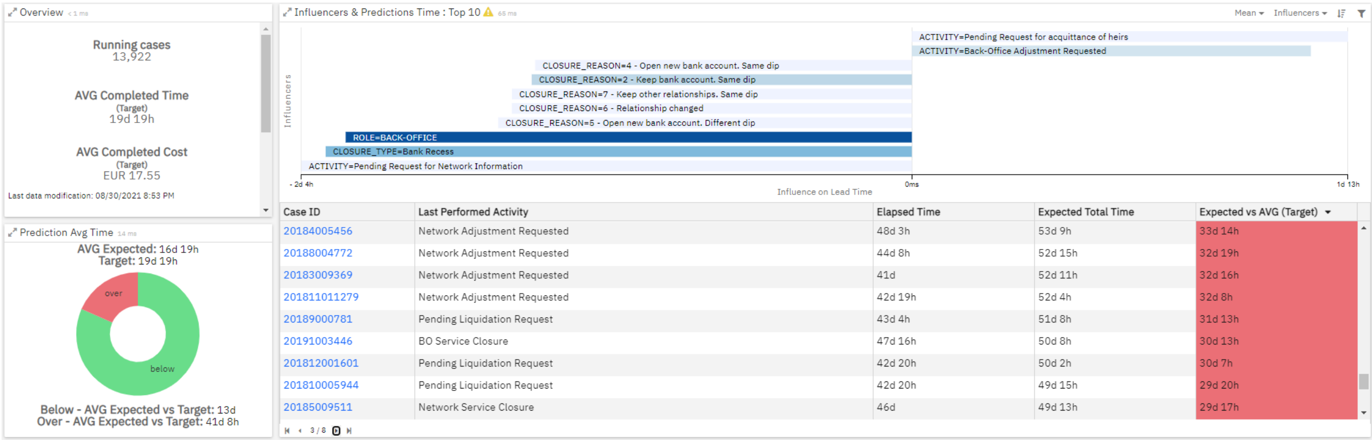

Fig. 1: The Analytics dashboard for Explainable Predictive Process Monitoring.

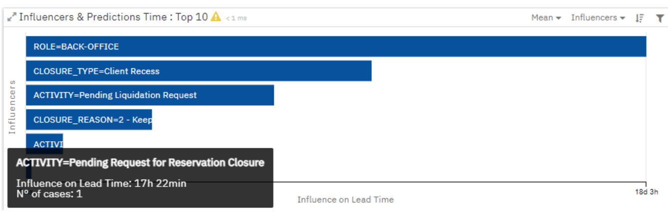

Fig. 2: Explanations related to one running case.

20181014067), one can see the the explanations for that case, named *influencers* in the tool (see Figure 2). Each explanation is of form a=x and is associated with a so-called *Shapley value* n computed via SHAP. In accordance with the SHAP theory, the explanation's interpretation is as follows: since attribute a takes on value x for this running case, the total-time prediction deviates n time units from the average total time of cases.

Let us consider again the all-cases dashboard in Figure 1: the bar chart in the top-right corner provides an helicopter view of the explanations. In particular, each row of the bar chart represents an explanation, and extends towards left or right, depending whether the average Shapley value for the explanation is negative or positive. The colour indicates the frequency of an explanation, with darker colours indicating a large number of running cases with that explanation.

As an example, explanation *ACTIVITY=Pending Request for Network Information* has a large bar with a light colour: this means that, for a small number of cases, the fact that the latest activity has been a *Pending Request for Network Information* has contributed to reduce the predicted total case duration by an average value of 2 days and 4 hours (the average shapley value). The explanation *CLOSURE_TYPE=Bank Recess* is conversely associated with a darker colour, namely with a large number of cases. The average shapley value is equal to -2 days: when the closure of the bank account is requested directly by the bank, the total time reduces by 2 days wrt. the average. This is indeed considered a simpler situation that does not require much interaction with the bank account holder. On the other side of the spectrum, explanation *ACTIVITY=Pending request for acquittance of heirs* has the largest positive shapley value: 1 day and 13 hours. This can also be justified: when the bank account is aimed at closure because of the holder's decease, the execution takes longer due to the involvements of the heirs.

III. Conclusions

In this paper we presented our ready-to-use explainable predictive module, which can work directly with the data processed by the process mining engine, without requiring any additional intervention or technical knowledge and providing almost immediate insights to the process stakeholder.

Our framework is fully integrated in the IBM Process Mining suite, and is ready for evaluation with the users; it can be leveraged directly by process stakeholders with no need for customization for each specific project, and is scalable thanks to a cloud-based infrastructure. The interface is the result of integrating feedback collected by process analysts and consultants within IBM. However, we aim at a more extensive user evaluation to further improve the user experience.

References

[1] A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés, "Predictive monitoring of business processes: A survey," *IEEE Transaction on Services Computing*, vol. 11, no. 6, pp. 962–977, 2018.

[2] I. Nunes and D. Jannach, "A systematic review and taxonomy of explanations in decision support and recommender systems," *User Modeling and User-Adapted Interaction*, vol. 27, no. 3–5, p. 393–444, Dec. 2017.

[3] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco*, 2016, pp. 1135–1144.

[5] R. Galanti, B. Coma-Puig, M. de Leoni, J. Carmona, and N. Navarin, "Explainable predictive process monitoring," in *2nd International Conference on Process Mining, ICPM 2020*. IEEE, pp. 1–8.

[6] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *The 3rd International Conference on Learning Representations, ICLR 2015*.

[8] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," in *Proceedings of the Workshop on ML Systems at NIPS 2017*, 2017.