INTERNATIONAL JOURNAL OF EDUCATION AND INFORMATION TECHNOLOGIES
DOI: 10.46300/9109.2022.16.5 Volume 16, 2022

Model reports, a supervision tool for Machine Learning engineers and users

Amine Saboni
Data User-Centered Tribe
OCTO Technology
Paris, France
amine.saboni@octo.com

Mohamed Ridha Ouamane
Department of Electrical Engineering
PRISME Laboratory, INSA Centre Val de Loire
Bourges, France
mohammed.ouamane@insa-cvl.fr

Ouafae Bennis
Departement of Automatic
PRISME Laboratory, University of Orleans
Chartres, France
ouafae.bennis@univ-orleans.fr

Frédéric Kratz
Departement of Automatic
PRISME Laboratory, , INSA Centre Val de Loire
Bourges, France
frederic.kratz@insa-cvl.fr

Received: March 20, 2021. Revised: November 13, 2021. Accepted: December 11, 2021. Published: January 6, 2022.

***Abstract*— This article investigates a methodology to design an automated supervision report, ensuring the suitability between the designers and the users of an algorithm. For this purpose, we built a super-vision tool, focused on error diagnosis.**

The argumentation of the article relies first on the exposition of the reasons to use model reports as a supervision artefact, with a prototype of implementation at an organization level, describing the necessary tooling to industrialize its production.

Finally, we propose a method for supervising machine learning algorithms in a responsible and sustainable way, starting from the conception of the algorithm, along its development and dur-ing its operating phase.

***Keywords*— Artificial intelligence (AI), error diagnosis, machine learn-ing (ML) supervision, operations**

I. Introduction and State of the Art

THE recent advances of the responsible AI field demonstrate a certain maturity and impact outside of the academic sphere [1]. The ethical guidelines proposed by a diversity of actors have focused over five main principles: transparency, justice and fairness, non-maleficence, accountability and privacy preserving [2].

To address the first two principles, the new generation of interpretability tooling [3, 4], have let emerge different interactions with users. Indeed, where the previous generation focused on providing statistical description of how predictions are computed by an algorithm, this new tooling formulates information in a digest way for non-technical users which can better apprehend the algorithm's behaviors. The transparency offered by interpretability tooling have led to better auditing tools and the model cards represent a standard, implemented in the industry, for algorithmic fairness evaluation [6].

The MLUX approach digs even deeper into the Human-AI collaboration, as it expresses explicit metrics to optimize, in a context of AI-assisted decision making, with notions of Dissonance, Trust Compatibility Score and Error Compatibility Score, to evaluate the continuous improvement of algorithms in their iterative development phase [5].

On another side, the global ML engineering practices have gain experience and accuracy [17], conducting the industrialization of ML algorithms production pipelines to observe a lot more parameters [18, 19]. Operating engineers have now a wider comprehension of how the models behave in production, at least at a technical level. This push the transparency and accountability limits to a new level, with a finer grain of understanding of how the technical process that leads to the training and the operating of a ML algorithm works.

Finally, privacy preserving machine learning techniques, such as federated learning [27] have been widely researched deployed by industrials such as Google. The health applications of those techniques seems to become the future of the discipline [28]. Other methods such as homomorphic encryption seems to promise a future where machine learning can be possible without compromising users personal data [29].

We propose in this article a supervision tool concept, focused on error diagnosis based mainly on the transparency, fairness and accountability principles. As algorithmic failures involve an increasing number of parameters, the need of supervision for ethical purposes grows. As ML-based products have a deep impact on its users, we propose to include them directly in the definition of the supervision report construction.

In order to isolate the accountability of each actor of an intelligent system chain, we simplify in this article the chain around three main roles: designers, who collect knowledge about a specific behavior in order to transform it into an intelligent system; operators who maintain this system; and users, who can have a deep feedback on the intelligent systems they use [8].

II. WHAT TO SHOW IN A MODEL REPORT

In order to supervise a system, web-oriented applications, or industrial software have developed strong methodology and tooling. These sub-fields of computer science have specified operational processes to optimize products quality and human safety [9, 15].

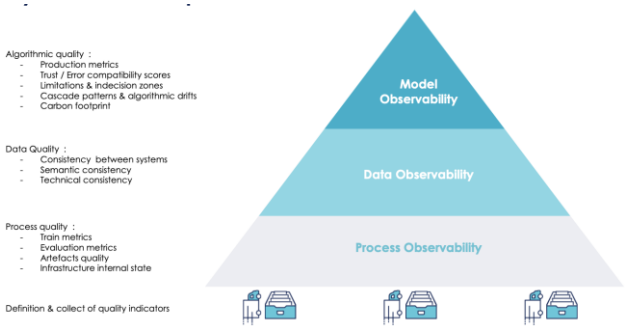


Fig1. Model supervision pyramid

To help supervise a model production pipeline, which will orchestrate a recipe of operations, applied sequentially, in a distributed environment and to specific data, we must observe two types of behaviors.

We will first find technical behaviors, which will help define the model observability layer, based on the data and process foundation layers' integrity. Indicators to observe may be in the following behaviors:

- Automated processing, to deduce software artefacts quality, as well as their underlying operational infrastructure. The indicators on these criteria can be expressed as the ML model specification consistency.
- Production algorithm's quality, based on quantitative analysis on production context (production data, inference service performances, etc.).
- Drift operations on operational data, along the measure of process quality.
- Carbon footprint, using appropriate tooling to evaluate each process electricity consumption and their carbon emission equivalence on the relying infrastructure [10]-[11]

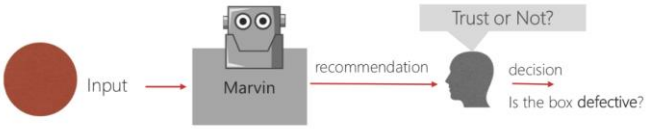
By assessing those behaviors, an essential knowledge is developed about the model. During its production phase, a technical view of its state, compared to its development phase can be built to detect any variation from its nominal state. A clear communication between operators and designers is needed, through a shared comprehension of the previously listed indicators.

We can also measure less technical behaviors, which will be linked to the algorithm's usage and to the effectiveness of its collaboration with its users. A specific methodology, detailed in the *Error matrix workshop* [23] proposes to link a business interpretation of a baseline algorithm to measurable and actionable indicators, focusing on errors or trips of the algorithm. By designing the user's error experience, this workshop makes users ask themselves mitigating questions as:

- What is the operational cost of an error?
- How to prevent the algorithm from failing in the identified corner cases?
- In case of recommender systems or decision assisting algorithms, how to identify and move beyond a bad recommendation ?

A domain that expresses an urgent need for qualitative recommendations systems is scientific communication, especially on the internet where the hierarchization of the information during the COVID-19 pandemic, because of its impact on the public health [12].

The collaboration between designers and users helps design a safer system, with the inclusion of more tripping patterns and methods to mitigate them. The MSResearch



Responsible AI team have proposed a user-centred development approach, proposing a list of conception patterns embedded in an iterative methodology to explore model limitations in collaboration with the system's users.

Fig2: Caja exploration workflow, from input data to user's feedback collection [24]

This Bayesian iteration on model parameters, based on user feedback on the algorithm's recommendations, leads to a more robust production algorithm [24]. An interesting feature of this methodology is that it optimizes criteria that are neither technical nor related to the business objectives behind the algorithm's design: it looks forward to increasing the trust of a user. During this iterative process, they define two types of metrics: descending compatibility between two versions of the same algorithms, and its antagonist : dissonance between a version of a model and its update toward the input data.

For a specific cohort, it is also possible to compute Trust & Error Compatibility scores. which will help model's operators anticipate a new version produced by its designers. Those scores might also be interesting to supervise at a global level, from the users, as they increase the model explainability and the trust users can give to its decisions.

Those technical but user-oriented evaluation criteria may be completed by socio-technical ones, as suggested by Upol Ehsan in his methodology for a shared exploration [14], among all available information about the model design & development. Data sources and their transformations, organizational constraints and business objectives, helps the user understand the behavior of the model and lead to the expression of an optimal definition of the model and its

behaviors, especially on failing mode.

One can argue that those interpretability analysis tools and methods, mainly based on the computation of Shapley values contributions or other statistical methods may be difficult to monitor on production, by their versatile nature and the fact that they cannot collect all the information used by an algorithm to score a prediction. Computing multiple times the Shapley values for a single version of an algorithm could lead to very different values thus a decrease of comprehension and trust for the users. As this versatility has not been issued by the literature, few mitigation process exists except computing those values in limited occasions, to avoid multiplicity of values.

III. How to Supervise a Model

A. Start at conception

In the development phase of a ML production pipeline, the baseline algorithm helps define on which metrics a feedback (which can be, depending on the underlying use case: a highly contributing feature, ground truth for numerical values of true label for classification algorithms) would be used in the production phase to build the adapted collection pipeline. On those data collection and feedback collection pipelines, technical indicators can be easy to retrieve and automatize [20]- [21].

But to determine which indicators would be the most impactful on user's trust and comprehension, a co-development is necessary of the algorithmic part of the ML pipeline.

We propose the *Error matrix workshop* [30], to help expert users ask valuable questions to mock an optimal diagnosis dashboard. Guided by the responsible AI principles (inclusion / fairness, accountability, non-maleficence, and privacy), a common exploration can lead to a shared comprehension on why the algorithm may fail in specific situations, and when those corner cases can happen. An example of dilemma that can let the user understand the statistical complexity abstracted by the algorithm is, in case of classification, to propose to them different configurations of F1 scores to optimize, explaining how it will impact the predictions:

Would the user prefer an algorithm that retrieves all positives, producing then more false positives? The drawback of that configuration is pretty clear for a user: a human review would be needed to relabel false positives. Discussing those impacts can lead to the emergence of more valuable features or the acknowledgment of more precise data sources, which will increase the trust of the users in the intelligent system embedding the algorithm, as they are a critical part of its development.

As the exploration deepens, the user will be able to apprehend with more ease the statistical complexity and will lead to a deeper comprehension of the algorithm's conception. Mixing that information with socio-technical information around the algorithm's development as design and implementations will sharpen the shared comprehension and potential flaws of the algorithm.

B. Build a prototype to answer to the right questions

Based on patterns identified during the *Error Matrix Workshop*, an interface can quickly be built using Wizard of Oz steps, to provide an asynchronous exploration dashboard to operators and users. The purpose of this interface is first to start the automation of feedback collection, starting with some basic interpretability tooling. The Shapash toolbox [4] let its users build in a few line of codes a wrapped explainable model, which will help identifying the main characteristics of a trained model.

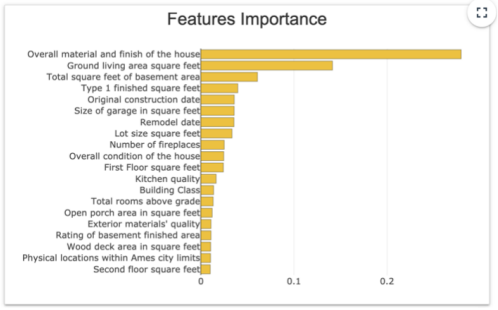


Fig 3: Shapash global explanation graph, feature importance

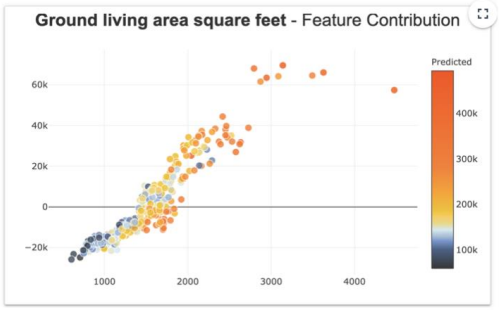


Fig 4: Shapash global explanation graph, feature contribution

Knowing which feature will have the most impact on a prediction made by the algorithm will help understand where it may fail in the future. Indeed, challenging the most impactful data used to predict a value with users can lead to a better understanding of the modelled behaviour and enlighten abusive correlations statistically learnt by the algorithm.

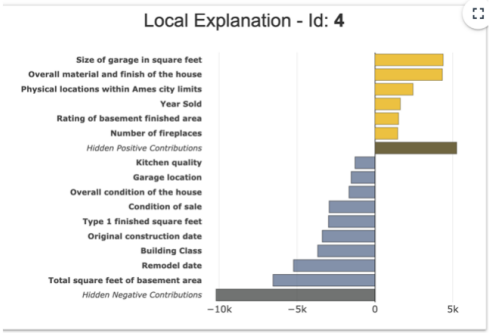


Fig 5: Shapash local explanation graph

In addition to global explanations, a second view of the local explanations is also helpful in the errors or tripping diagnosis. Indeed, in case of a bad prediction or recommendation, auditing the local explanation of the algorithm can answer to the following questions:

- Why has the algorithm been failing to predict this specific case?
- Is the class of this example enough represented in the train dataset?
- Can we identify which corner cases will always make the algorithm fail in a specific parameter?

Studying the answers to these questions, with users and domain experts should lead to building representative indicators that may be specified following a test referenced by the supervision pyramid. Exposing the evolution of the indicators in a supervision dashboard, accessible and understandable by non-technical users can provide a first level of feedback loop, if they are able to comment them.

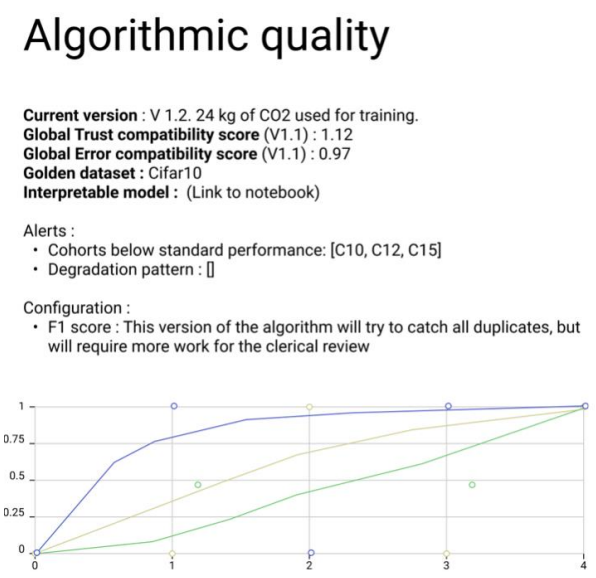


Fig. 6: Algorithm quality layer represented on a supervision dashboard

To summarize the state of the model as observed, three kind of metrics could be represented in the report:

- Train metrics, with performance and carbon footprint computed during the training phase.
- Evaluation metrics, computed on a reference test set, and compared between different versions of an algorithm. Usage of Trust and Error compatibility scores must be defined with a shared comprehension between designers, operators and users.
- Data oriented metrics, to understand which behaviors the algorithm will have toward specific cohorts of the reference test set.

C. Automatise the collection of indicators, and iterate with users

Starting with foundations, indicators can be automatically extracted on the process layer of the pyramid. As good data collection always starts with manual production (a procedure detailed in the *emerging data architecture* [22]). This first view of the supervision dashboard, updated on a regular basis, will help non-technical users apprehend the complex object that has to be supervised.

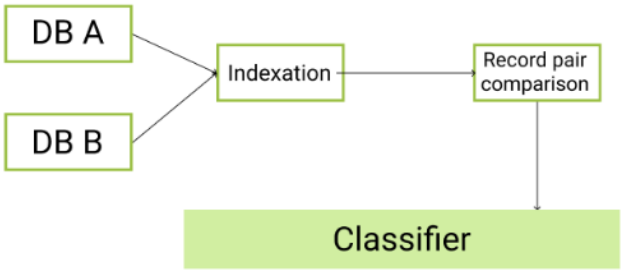


Fig. 7 : Data & process quality layer

To ensure the model production process do not induce errors in the algorithm's prediction, the process can be broken into unit operations, with custom indicators that reflects the quality of the operation directly on the dashboard.

In the fig 7, a data deduplication pipeline is define by four main operations: data extraction from several databases, indexation of the extracted data, record pair comparison based on the indexation and a final classification. Highlighting the state of each operation on the production data will increase the trust of the user in the algorithm's production pipeline. As the four operations shows a green status, the processes can be excluded from a potential error diagnosis.

Once users are able to understand this level of information, an iterative work on the pertinence of shown indicators, focusing on getting feedback from users about the potential causes of errors.

Finally, the dashboard can be considered as fully operational when the three layers of the pyramid are represented, providing insightful information for all the actors interacting with it. To ensure that users can have a real impact on the algorithm, dedicated space to provide feedback on the decisions the algorithm have proposed should be available directly through the report interface. Ensuring this capability for users will lead to a continuous improvement of the algorithms by its operators.

IV. Conclusion

We have explored in this article the main benefits of automatising the supervision of an algorithm, through a non-technical user-friendly dashboard. Exposing up-to-date information helps designers build and update predictive models in a sustainable and responsible way. It gives the opportunity to operators to monitor standardized procedures, to limit the potential failures of the model.

The supervision also benefits to users, who could be domain experts, by giving them the ability to provide feedback to the model, especially when they can identify well know trip patterns.

To achieve this supervision, we have seen that an active collaboration is needed. Indeed, collaboration between users and designers when building a decision-making predictive algorithm have been proved of a better quality if users understand how it is behaving, and where potential flaws can appear, through the *Error matrix workshop.* The handling of those flaws can be directly integrated into the system hosting the algorithm, as the exchange between users and designers improves the shared mental model of what the system should be in its operating phase.

Finally, in order to expose these flaws, three kind of indicators have been identified in the pyramid of the supervision (process, data & algorithm quality), and an iterative process has been proposed to build an optimal reporting dashboard, with feedback collection features, to let users have an impact on the algorithm.

References

[1] Rakova, B., Yang, J., Cramer, H., & Chowdhury, R. (2021). Where responsible AI meets reality: Practitioner perspectives on enablers for shifting organizational practices. Proceedings of the ACM on Human-Computer Interaction, 5(CSCW1), 1-23.

[2] Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1(9), 389-399

[3] Responsible AI widgets for Error Analysis, Microsoft Research

[4] Shapash responsible AI tooling (2020), MAIF

[5] Nushi, B., Kamar, E., & Horvitz, E. (2018, June). Towards accountable ai: Hybrid human-machine analyses for characterizing system failure. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (Vol. 6, No. 1)

[6] Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... & Gebru, T. (2019, January). Model cards for model reporting. In Proceedings of the conference on fairness, accountability, and transparency (pp. 220-229)

[7] Salesforce, Model Cards for ai model transparency (2020) https://blog.einstein.ai/model-cards-for-ai-model-transparency/

[8] People+AI Research Group, Google, user needs https://pair.withgoogle.com/chapter/user-needs/

[9] Davidson, E. M., McArthur, S. D., McDonald, J. R., Cumming, T., & Watt, I. (2006). Applying multi-agent system technology in practice: Automated management and analysis of SCADA and digital fault recorder data. IEEE Transactions on Power Systems, 21(2), 559-567

[10] Victor Schmidt, CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing, https://github.com/mlco2/codecarbon

[11] Bourdon, A., Noureddine, A., Rouvoy, R., & Seinturier, L. (2013). Powerapi: A software library to monitor the energy consumed at the process-level. ERCIM News, 2013(92).

[12] Hoang, L. N. (2020). Science communication desperately needs more aligned recommendation algorithms. Frontiers in Communication, 5, 115

[13] Tournesol public wiki homepage, https://wiki.tournesol.app/index.php/Main_Page

[14] Upol Ehsan, Q. Vera Liao, Michael Muller, Mark O. Riedl, and Justin D. Weisz. (2021). Expanding Explainability: Towards Social Transparency in AI systems. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 82, 1–19.

[15] *Humble, J., & Kim, G. (2018). Accelerate: The science of lean software and devops: Building and scaling high performing technology organizations. IT Revolution.*

[16] Blangero, Grimprel, La confiance des utilisateurs dans les systèmes impliquant de l'IA, 2019 https://blog.octo.com/la-confiance-des-utilisateurs-dans-les-systemes-impliquant-de-lintelligence-artificielle/

[17] Muccini, H., & Vaidhyanathan, K. (2021). Software Architecture for ML-based Systems: What Exists and What Lies Ahead. arXiv preprint arXiv:2103.07950.

[18] Lachheb, I. (2021) Le feature store, nouvel outil pour les projets de data science, https://blog.octo.com/le-feature-store-nouvel-outil-pour-les-projets-data-science/

[19] Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P. K., & Aroyo, L. M. (2021). " Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI.

[20] Ait Bachir, S. (2021) Les tests automatisés en delivery de machine learning https://blog.octo.com/les-tests-automatises-en-delivery-de-machine-learning/

[21] Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017, December). The ml test score: A rubric for ml production readiness and technical debt reduction. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 1123-1132). IEEE.

[22] Histoire d'une architecture émergente, Compte-rendu du talk d'Emmanuel Lin Toulemonde, Duck Conf 2021, Alessandro Mosca.

[23] Atelier de matrice d'erreur, publication in progress at https://blog.octo.com

[24] Bansal, G., Nushi, B., Kamar, E., Weld, D. S., Lasecki, W. S., & Horvitz, E. (2019, July). Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 2429-2437).

[25] Kumar, I. E., Scheidegger, C., Venkatasubramanian, S., & Friedler, S. (2020, January). Shapley Residuals: Quantifying the limits of the Shapley value for explanations. In ICML Workshop on Workshop on Human Interpretability in Machine Learning (WHI).

[26] Kara Combs, Mary Fendley, Trevor Bihl, A Preliminary Look at Heuristic Analysis for Assessing Artificial Intelligence Explainability, WSEAS Transaction on Computer Research, Volume 8, 2020, pp. 61-72.

[27] Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al, Advances and open problems in federated learning, arXiv preprint arXiv:1912.04977, 2

[28] RIEKE, Nicola, HANCOX, Jonny, LI, Wenqi, et al. The future of digital health with federated learning. NPJ digital medicine, 2020, vol. 3, no 1, p. 1-7.

[29] ASLETT, Louis JM, ESPERANÇA, Pedro M., et HOLMES, Chris C. A review of homomorphic encryption and software tools for encrypted statistical machine learning. arXiv preprint arXiv:1508.06574, 2015.

[30] Roussel F., Pemodjo M., Saboni A. L'atelier matrice d'erreur, démystifier les performances du ML avec ses utilisateurs, https://blog.octo.com/latelier-matrice-derreur-demystifier-les-performances-du-ml-avec-ses-utilisateurs/