auto-xFS: An explanation-based feature selection tool for more meaningful and trustworthy machine learning models

Haomiao Wang, Julien Aligon, Haoran Zhou, Chantal Soulé-Dupuy, Paul Monsarrat

SoftwareX 31 (2025) 102268

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

auto-xFS: An explanation-based feature selection tool for more meaningful and trustworthy machine learning models

Haomiao Wang [b], Julien Aligon [a], Haoran Zhou [a], Chantal Soulé-Dupuy [a], Paul Monsarrat [b,c,*]

[a] Université Toulouse Capitole, Institute of Research in Informatics (IRIT) of Toulouse, CNRS – UMR5505, Toulouse, 31000, France
[b] RESTORE Research Center, Université de Toulouse, INSERM 1301, CNRS 5070, EFS, ENVT, 4bis Avenue Hubert Curien, Toulouse, 31100, France
[c] Oral Medicine Department and Hospital of Toulouse, CHU de Toulouse, Toulouse, 31062 Cedex 09, France

ARTICLE INFO

Keywords:
Feature selection
Explainability
Machine learning
Discernibility

ABSTRACT

auto-xFS is a novel tool for feature selection (FS) based on a three-dimensional perspective encompassing feature retention rate, machine learning (ML) model performance, and explainability (XML). The application is designed to streamline the user's workflow by autonomously hyperparameterizing FS techniques, ML models, and XML methods using a meta-learning approach. Our findings demonstrate that prioritizing FS that yields highly precise prediction explanations even at the expense of a slight reduction in model accuracy, can ensure more meaningful information for the user. *auto-xFS* is totally suited to FS in critical areas such as biomedicine where user confidence is crucial.

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-25-00241
Permanent link to Reproducible Capsule	https://reco-xfs.irit.fr
Legal Code License	GNU General Public License v2.0
Code versioning system used	git
Software code languages, tools, and services used	python
Compilation requirements, operating environments & dependencies	dash, redis, SQLite; Operating systems: Linux/WSL; Dependency management tool: conda
If available Link to developer documentation/manual	https://github.com/jaligon/auto-xFS/blob/main/README.md
Support email for questions	julien.aligon@irit.fr

1. Motivation and significance

According to the Hughes phenomenon [1], a model's machine learning (ML) capacity increases with the number of features up to a certain threshold, beyond which it decreases due to limitations in sampling and computation. As the number of observations and attributes increases, the complexity of learned models grows, potentially leading to overfitting and reduced performance on new data. Additionally, incorporating irrelevant features negatively impacts model accuracy. Dimensionality reduction methods, particularly Feature Extraction and Feature Selection (FS), are designed to address these challenges by simplifying data and enhancing its quality. A plethora of FS methods have been documented in literature and are conventionally classified into three categories: filter, wrapper, and embedded. Filter methods operate independently of the ML model and rely on an evaluation metric to differentiate features, typically based on similarity or other statistical techniques [2]. Wrapper methods, in contrast, use the ML model itself to assess the performance of selected feature subsets. These methods can be framed as an optimization problem, employing techniques such as genetic algorithms [3] or swarm intelligence [4]. Embedded methods integrate feature selection directly within the ML model. In tree-based models, for instance, the learning phase determines feature importance

* Corresponding author at: RESTORE Research Center, Université de Toulouse, INSERM 1301, CNRS 5070, EFS, ENVT, 4bis Avenue Hubert Curien, Toulouse, 31100, France.
E-mail address: paul.monsarrat@inserm.fr (P. Monsarrat).

https://doi.org/10.1016/j.softx.2025.102268
Received 11 April 2025; Received in revised form 10 June 2025; Accepted 8 July 2025
Available online 31 July 2025

by generating splits and identifying features to discard. Given the wide array of potential techniques, it is challenging to ascertain the most suitable method for a given dataset, particularly given the absence of a universally optimal solution that would apply to all situations (the "no free lunch" theorem [5]). Moreover, the FS evaluation strategy remains a bottleneck due to the absence of a known a priori optimal subset for real datasets. This evaluation is inherently indirect, as it relies on the outputs of a model, rather than the model itself, thereby introducing potential biases into the assessment methodology [6,7].

In critical application fields, such as biomedicine, practitioners require that FS, or the entire learning procedure, be explainable, which contrasts with the black-box nature of many machine learning models [8]. As a result, explainable machine learning (XML) has gained significant attention in recent years. Among all methods, local attributive XML methods such as LIME [9] and SHAP [10] are particularly effective as they can explain model predictions in terms of the contributions of each feature for each instance. In addition to the well-documented benefits of these methods, including their ease of visualization and user-friendliness [11], these approaches guarantee that a single explanation is provided for each instance (the size of the two spaces is equivalent). This distinction is notable when compared to alternative methods, such as counterfactuals [12], which do not share this characteristic. Furthermore, these methods can perform both local and global analyses, aggregated over a subset of instances. As a result, they are particularly well-suited for supporting data analysis [13,14].

The literature primarily evaluates the relevance of feature selection based on ML model performance and the reduction in the number of features compared to the initial dataset. But what about the quality of the explanations generated? This question was addressed in a previous study [15], which highlighted the significant impact of feature selection on prediction explanations.

This paper aims to demonstrate the importance of automatically recommending feature subsets that account for the evaluation of prediction explanations, alongside traditional assessments of model performance and feature retention. We argue that feature selection should be framed as a trade-off among these three dimensions. The proposed application allows users to analyze feature selections based on their explainability in relation to ML model performance. For instance, one might prioritize a feature selection that yields highly precise prediction explanations – according to a chosen explanation evaluation metric – even at the expense of a slight reduction in model accuracy. In addition, the application architecture has been meticulously designed to automate the hyperparameterization of ML and XML models as much as possible, notably through a meta-learning strategy.

2. Software description

2.1. Software architecture

The platform is implemented as a web application to ensure optimal accessibility. It integrates a modular front-end built with callback mechanisms to handle sequential pipeline stages (file upload, hyperparameter tuning of ML and XML models) and provides interactive visualizations of possible subsets of features using Plotly Dash.[1] The Python-based backend ensures extensibility for computational tasks. While calculations are currently performed server-side, user data and computation results are not stored to maintain data security and confidentiality. Only memory cache is utilized for calculations and visualizations. However, an SQLite[2] database is reserved to store metadata in order to recommend hyperparameter for ML models through meta-learning. The backend component also includes feature selection computation as well as ML model training and XML calculations. Fig. 1 illustrates the global architecture of the system, highlighting the sequential steps and interactions between components.

[1] https://dash.plotly.com

[2] https://www.sqlite.org

2.2. Software functionalities

Fig. 2 illustrates the parameterization options available to the user. The software can process tabular datasets for both classification and regression tasks, accommodating continuous, binary, and ordinal categorical features. However, it does not support non-ordinal categorical features. In such cases, a one-hot encoding will be applied to the data before training, with a notification for the user. The following ML models are available for use: Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), XGBoost, and TensorFlow Sequential. The selection of XML methods depends on the selected ML model and covers most of the attributive strategies available in the literature, including: LIME, SHAP (and its variants TreeSHAP, DeepSHAP), Shapley values (complete method), and coalitional methods [16,17]. Details about FS methods and evaluation metrics are given below.

2.2.1. Choosing FS methods

The software supports a variety of FS methods considering different FS families (filter, wrapper and embedded) as well as diverse computational approaches within each category. Within the filter-based family, several techniques are available: similarity-based methods (*Fisher*, *ReliefF* [18], and *SPEC* [19]), statistical approaches (*F-test* and *Chi-squared*), sparse-learning-based methods (*RFS* [20]), and information-theory-based techniques (*mRMR* [21], *CMIM* [22] and *JMI* [23]). For embedded methods, feature ranking are determined using feature importance scores from Random Forest (*RF*) or coefficients such as Linear Support Vector Machines (*SVM*) and Logistic Regression (*LG*). In the wrapper method, *BorutaShap* [24] (Boruta algorithm with SHAP values) is considered.

Clearly, the built-in methods do not cover all existing feature selection techniques comprehensively. According to the "no free lunch" theorem, no single method is suitable for every situation, and users might want to explore other potential subsets to generate new hypotheses. Therefore, users have the option to add customized subsets in addition to those proposed by predefined feature selection methods. These customized subsets could originate from sophisticated feature selection techniques, such as metaheuristic methods, manual selection by experts, or previous analysis rounds with different parameters. The subset mask is then ready to be copied into the visualization phase to facilitate further analysis.

2.2.2. Evaluation metrics and recommendation strategy

The evaluation metrics encompass those traditionally associated with ML model (accuracy, coefficient of determination (R^2), mean absolute error (MAE), and root mean square error (RMSE), contingent upon the learning task). Additionally, the retention rate is considered. The primary novelty lies in the methodology used to evaluate explainability, leveraging attributive methods through the lens of feature selection. To this end, the metrics proposed in [15] are considered. First, the Kendall rank correlation coefficient measures the differences in the ranking of the most influential features before and after feature selection. Second, the relative influence change quantifies variations in feature influence values. Third, the composite metric RI integrates the scores of these two metrics. Fourth, the composite metric RIA extends RI by incorporating the ML model's accuracy. Finally, discernibility [25] assesses the correlation between feature influence values and the actual feature values, ensuring the presence of potentially meaningful information for the user. Although such a relationship between data and explanations cannot be translated directly into causality, it remains an argument in favor of such a link [25,26].

To facilitate comparative analysis of the selected feature subsets, the software offers four distinct recommendation strategies based on the previously outlined evaluation metrics. The first strategy relies on the Pareto front, aiming to optimize the selected metrics as effectively as possible. The second and third strategies recommend feature subsets by aggregating either the scores or ranks of the considered metrics. The fourth strategy generates recommendations based on the user's specified preferences, prioritizing the selected metrics accordingly.

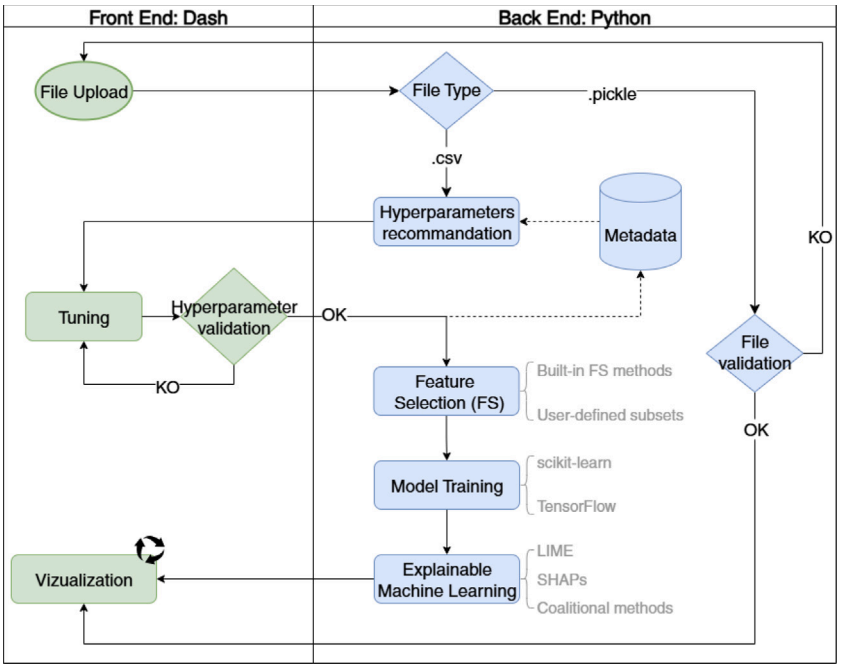

Fig. 1. Application architecture.

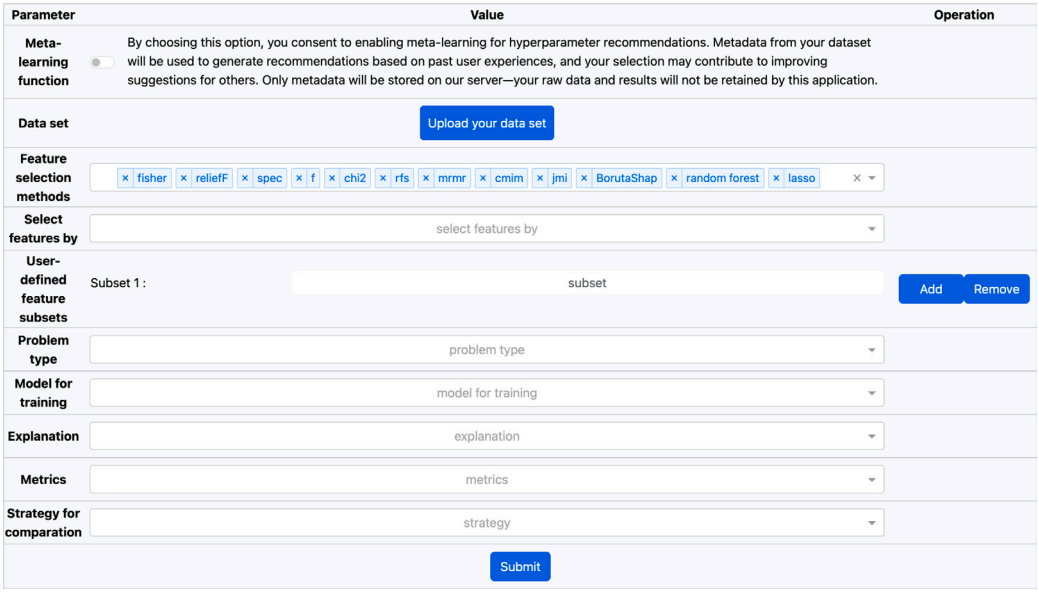

Fig. 2. User input interface.

2.2.3. Result investigation

Fig. 3 shows an example of the result using the Pareto recommendation strategy. Each point on the three-dimensional scatter plot represents a subset of features, colored according to whether it pertains to a Pareto-optimal solution or not. Users can change the metrics on the axis and rotate the cube for better investigation. Detailed information (feature names, feature selection methods, and summary plot of the explanation) is displayed on the right when a solution (point) is selected within the cube.

2.2.4. Model tuning and meta-learning based assistance

As machine learning algorithms become more sophisticated, the number of hyperparameters that need to be managed also increases, making hyperparameter tuning a challenging task, even for machine learning experts [27]. In the current version of the system, users can modify, manually, the hyperparameters of the pipeline; but if they do not, default values are applied. Thus, the software offers to add an automatic hyperparameterization assistance function based on a meta-learning approach. This support should cover the entire pipeline, not only the machine learning model, but also feature selection methods, explanation techniques, and user preferences for recommendations.

This challenge is well recognized within the field of Automated Machine Learning (AutoML), where the goal is to automate the construction of machine learning pipelines. A fundamental solution to this challenge is meta-learning, which involves recommending hyperparameters based on prior knowledge or experiences. Typically, meta-learning can utilize various sources of information, such as model evaluations, task properties, and previous models [28]. However, in the feature selection recommendation system, model evaluation is prohibitively expensive, and no pre-existing models are available for the newly

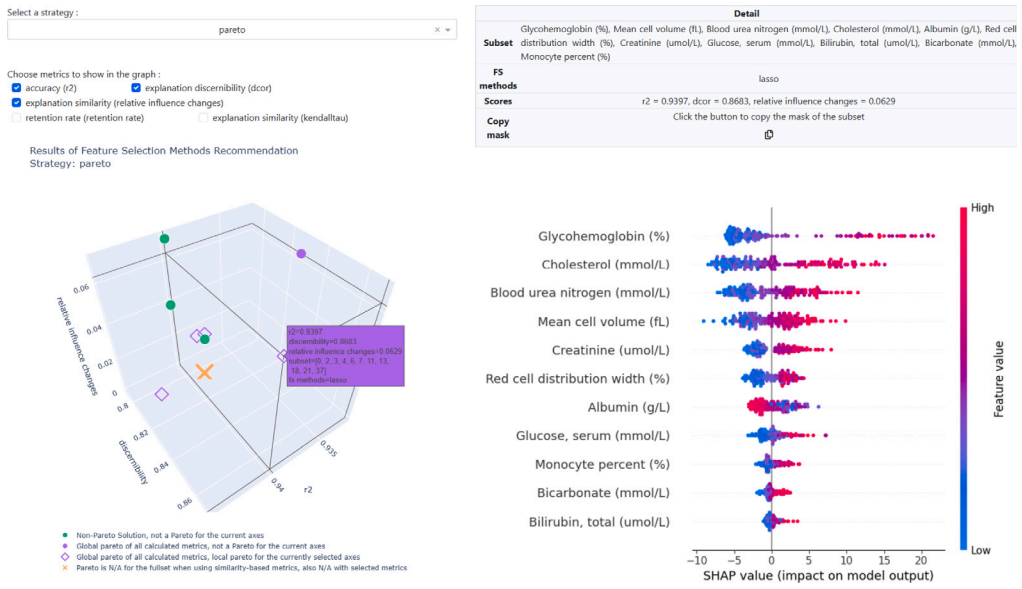

Fig. 3. Visualization and investigation of results: Pareto strategy.

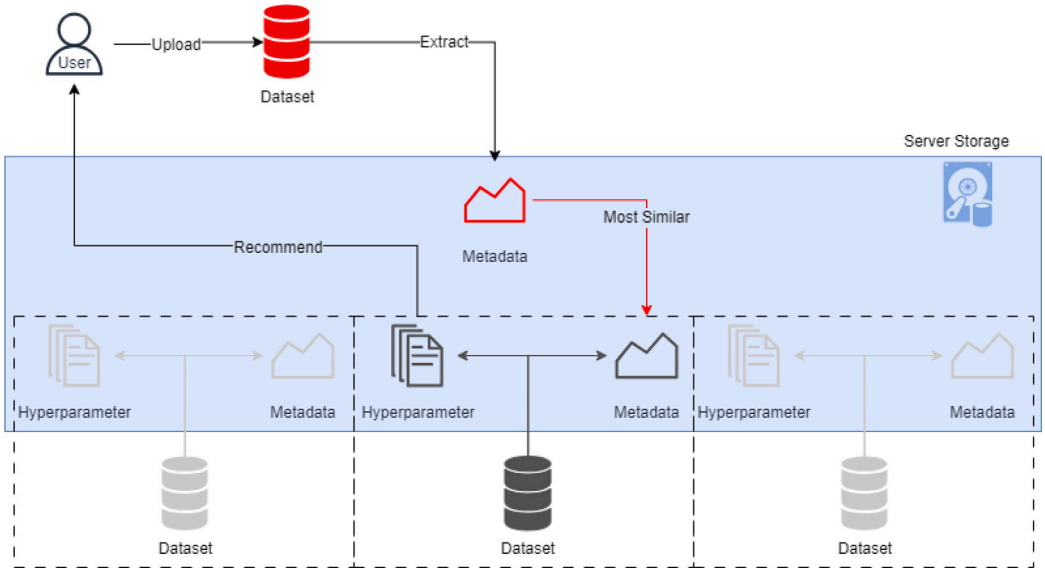

Fig. 4. Hyperparameter recommendation.

uploaded dataset. Therefore, our meta-learning approach is constrained to using the task properties, *i.e.,* metadata, for making the recommendations. More specifically, this means that recommendations are informed by peer experience, as similar datasets often share common analysis objectives. With the user's consent, insights from previous users can be leveraged to enhance the recommendations for future users. The core concept is straightforward: the more similar two tasks are in terms of dataset characteristics and objectives, the more their optimal workflows tend to align. Metadata allows for the quantification of task similarity. Numerous researchers [29–31] have explored metadata lists to identify and construct the most relevant set of metadata for different datasets. These metadata range from simple descriptive or statistical indicators to model-based metrics or landmarks. The main steps of the recommendation process implemented in this software are illustrated in Fig. 4.

2.2.5. Data privacy and confidence

Due to potentially very expensive calculations (training ML models, calculating feature subsets, calculating explanations, etc.), a result file is directly downloadable once all the calculations are done. The user then has the option to upload this file in order to avoid the same calculations and to have access to the visualizations of the feature subsets. Although calculations are currently performed server-side, user data and computation results are not stored for data security and confidentiality reasons. Only the memory cache is used for calculations and visualizations. However, some dataset metadata may be collected with user consent, which could help to recommend hyperparameters to beginner users in the future through meta-learning.

3. Illustrative example

In order to illustrate the merits of our application, we posit that a user would be interested in selecting relevant features from the Lung Cancer dataset,[3] which comprises 16 features (including target) and

[3] https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer

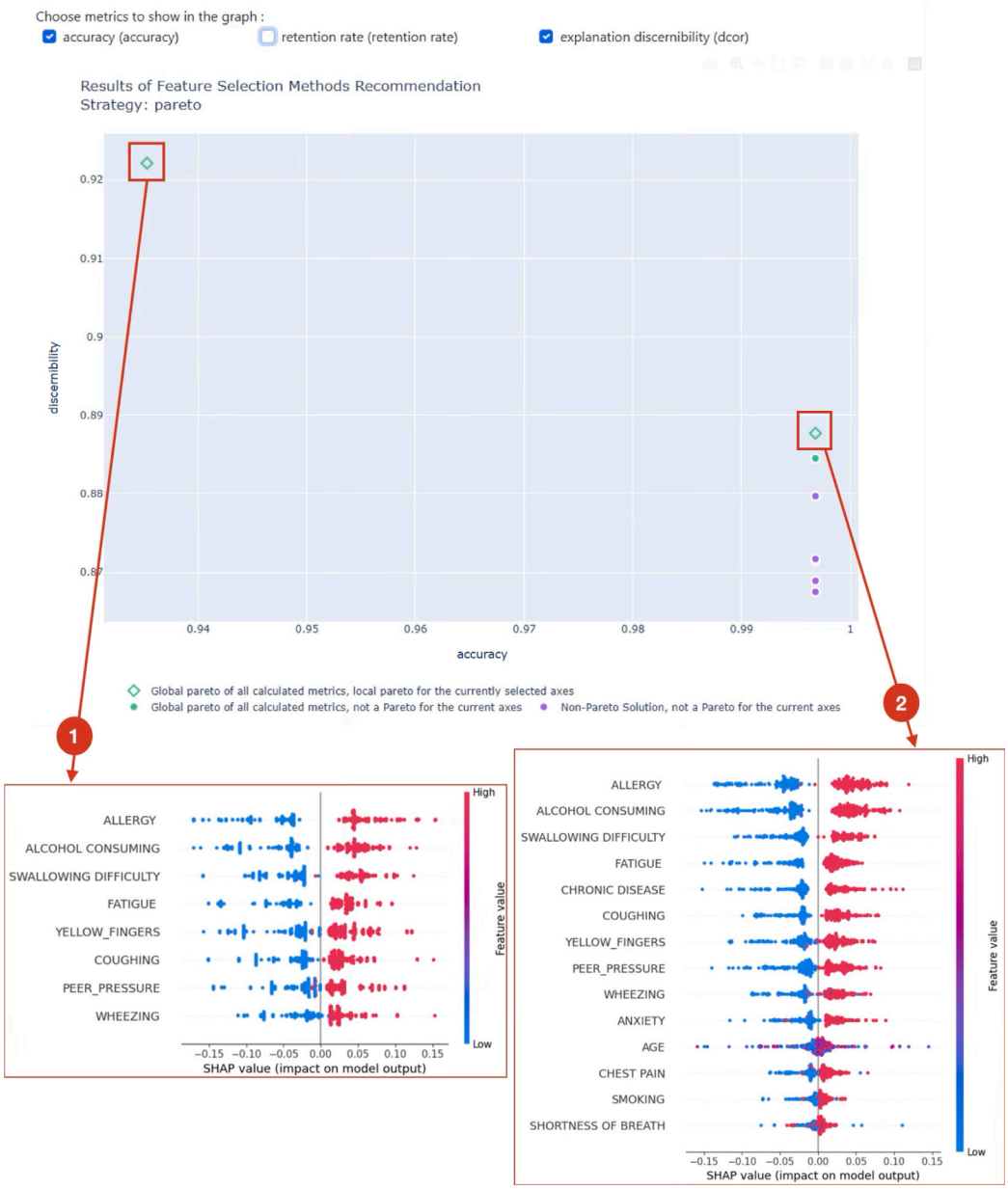

Fig. 5. Illustrative example using Lung cancer dataset on *auto-xFS*.

284 instances. The problem pertains to binary classification, specifically determining whether or not a patient has lung cancer. After the dataset is uploaded to the web application, a one-hot encoder is applied. Concerning parameterization, we consider all the feature selection methods proposed by the system. The system automatically detects that it is a classification problem. We elect to use a Random Forest model and a XML TreeSHAP model. We consider the following evaluation metrics: accuracy, retention rate, and explanation discernibility. We also consider the Pareto strategy for feature subset recommendation. The result is illustrated in Fig. 5. For the sake of brevity, only axes accuracy and discernibility have been considered to represent the subset of features. It can be noted that the Pareto front (local to these two axes) contains two possible solutions (red squares): one with higher accuracy and another with higher discernibility. Global explanations are also provided. It is noteworthy that explanations with the highest discernibility yield clearer and more concise explanations (1) than those with the highest accuracy (2); the age feature is particularly unreadable. Consequently, the user opts to select the subset with the best discernibility, albeit at the expense of a slight loss in accuracy.

4. Impact

auto-xFS endeavors to refine the conventional conception of feature selection by incorporating the concept of explainability. In the context of machine learning, the primary objective of feature selection is to identify the most minimal subset of features that enables the acquisition of the most efficient ML model. We contend that this paradigm is inadequate in the current era, where the black-box effect of ML models hinders our capacity to comprehend the rationale behind predictions [32]. Indeed, the pre-processing tasks, such as feature selection, wield a considerable influence on the explainability of an ML model. As previously demonstrated [15], feature selection can lead to explanations that, while accompanying a high-performing ML model, may be uninformative or lack practical value. This software builds on previous research [13–15,33,34], with the primary objective of leveraging explainability to enhance data analysis processes. The software recommends feature subsets evaluated across three key dimensions: model performance, feature retention rate, and explanatory power. This approach allows users to navigate trade-offs between these

factors, tailoring feature selection to their specific needs. Designed for users interested in machine learning and feature selection, the software is particularly beneficial for domain experts – especially in biomedical fields – who seek to automatically identify and visualize the most relevant features for their dataset. The application ensures efficient data management by storing only the metadata necessary for automated hyperparameter tuning (AutoML) and feature visualization. The provided explanations are based on attributive methods, which are user-friendly, easily interpretable, and well-suited for both global and concise dataset representations [11]. *auto-xFS* is one of the few tools in the literature [35] to demonstrate a concrete use case for XML evaluation metrics. Measuring explanation quality remains a major and timely challenge [36]. This tool represents a promising avenue for integrating a broader range of quality metrics documented in the literature, as well as for exploring new research hypotheses.

5. Conclusion

We present *auto-xFS*, a novel tool for feature selection based on a three-dimensional perspective encompassing feature retention rate, ML model performance, and explainability. The application is designed to streamline the user's workflow by autonomously hyperparameterizing FS techniques, ML models, and XML methods using a meta-learning approach. The primary strength of *auto-xFS* lies in its ability to visualize and analyze feature subsets according to the selected evaluation metrics, corresponding to the three analysis dimensions. This enables users to assess the impact of feature selection on a predictive model, particularly regarding its performance and the explanations generated. Our findings demonstrate that prioritizing feature selection with slightly lower predictive accuracy can lead to more informative explanations. Given the numerous contributions from the literature in this field, we plan to integrate additional explainability evaluation metrics. In the long term, our goal is to establish a metadata repository to further automate feature selection via meta-learning, particularly for recommending evaluation metrics. We also plan to extend this work by taking into account the user's profile. Indeed, a data scientist might be more familiar with analyzing raw performance scores, rather than a biologist or medical practitioner more interested in ensuring that the selected variables correspond to his or her business needs. Data storytelling [37,38] could be an interesting avenue of research for automatically proposing an analysis of variables relevant to a given user profile.

CRediT authorship contribution statement

Haomiao Wang: Writing – original draft, Software, Conceptualization, Visualization, Methodology. **Julien Aligon:** Writing – review & editing, Validation, Project administration, Writing – original draft, Resources, Methodology, Formal analysis, Funding acquisition, Conceptualization. **Haoran Zhou:** Software, Writing – original draft, Conceptualization. **Chantal Soulé-Dupuy:** Validation, Project administration, Conceptualization, Writing – review & editing, Resources, Funding acquisition. **Paul Monsarrat:** Validation, Resources, Methodology, Funding acquisition, Conceptualization, Writing – review & editing, Supervision, Project administration, Investigation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Occitanie Region, the Federal University of Toulouse Midi-Pyrénées (grant ADI 2021, N° ALDOCT89533). This study has been partially supported by the Agence Nationale pour la Recherche, through the grant EUR CARe N° ANR-18-EURE-0003 and the national infrastructure "ECELLFrance: Development of mesenchymal stem cell based therapies" (PIA-11-INBS-0005) in the framework of the Programme des Investissements d'Avenir.

References

[1] Hughes G. On the mean accuracy of statistical pattern recognizers. IEEE Trans Inform Theory 1968;14(1):55–63.
[2] Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, et al. Feature selection: A data perspective. ACM Comput Surv 2018;50(6):94.
[3] Oreski S, Oreski G. Genetic algorithm-based heuristic for feature selection in credit risk assessment. Expert Syst Appl 2014;41(4):2052–64.
[4] Rostami M, Berahmand K, Nasiri E, Forouzandeh S. Review of swarm intelligence-based feature selection methods. Eng Appl Artif Intell 2021;100:104210.
[5] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82.
[6] McNee SM, Riedl J, Konstan JA. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In: CHI '06 extended abstracts on human factors in computing systems. CHI EA '06, New York, NY, USA: Association for Computing Machinery; 2006, p. 1097–101. http://dx.doi.org/10.1145/1125451.1125659.
[7] Hossin M, Sulaiman MN. A review on evaluation metrics for data classification evaluations. Int J Data Min Knowl Manag Process 2015;5(2):1.
[8] Loyola-González O. Black-box vs. White-box: Understanding their advantages and weaknesses from a practical point of view. IEEE Access 2019;7:154096–113. http://dx.doi.org/10.1109/ACCESS.2019.2949286.
[9] Ribeiro MT, Singh S, Guestrin C. " 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. 2016, p. 1135–44.
[10] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 2017;30.
[11] Arunika M, Saranya S, Charulekha S, Kabilarajan S, Kesavan G. A survey on explainable AI using machine learning algorithms shap and lime. In: 2024 15th international conference on computing communication and networking technologies. IEEE; 2024, p. 1–6.
[12] Guidotti R. Counterfactual explanations and how to find them: literature review and benchmarking. Data Min Knowl Discov 2024;38(5):2770–824.
[13] Cooper A, Doyle O, Bourke A. Supervised clustering for subgroup discovery: an application to COVID-19 symptomatology. In: Joint European conference on machine learning and knowledge discovery in databases. Springer; 2021, p. 408–22.
[14] Escriva E, Lefrere T, Martin M, Aligon J, Chanson A, Excoffier J-B, et al. Effective data exploration through clustering of local attributive explanations. Inf Syst 2025;127:102464.
[15] Wang H, Doumard E, Soulé-Dupuy C, Kémoun P, Aligon J, Monsarrat P. Explanations as a new metric for feature selection: A systematic approach. IEEE J Biomed Heal Inform. 2023;27(8):4131–42. http://dx.doi.org/10.1109/JBHI.2023.3279340.
[16] Ferrettini G, Aligon J, Soulé-Dupuy C. Improving on coalitional prediction explanation. In: Advances in databases and information systems: 24th European conference, ADBIS 2020, lyon, France, August 25–27, 2020, proceedings 24. Springer; 2020, p. 122–35.
[17] Ferrettini G, Escriva E, Aligon J, Excoffier J-B, Soulé-Dupuy C. Coalitional strategies for efficient individual prediction explanation. Inf Syst Front 2022;24(1):49–75.
[18] Kira K, Rendell LA. A practical approach to feature selection. In: Machine learning proceedings 1992. Elsevier; 1992, p. 249–56.
[19] Zhao Z, Liu H. Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th international conference on machine learning. 2007, p. 1151–7.
[20] Nie F, Huang H, Cai X, Ding C. Efficient and robust feature selection via joint ℓ2, 1-norms minimization. Adv Neural Inf Process Syst 2010;23.
[21] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 2005;27(8):1226–38.
[22] Fleuret F. Fast binary feature selection with conditional mutual information. J Mach Learn Res 2004;5(9).
[23] Yang H, Moody J. Feature selection based on joint mutual information. In: Proceedings of international ICSC symposium on advances in intelligent data analysis. 1999, Citeseer; 1999, p. 22–5.

[24] Keany E. Is this the best feature selection algorithm "BorutaShap"?. 2020, https://medium.com/analytics-vidhya/is-this-the-best-feature-selection-algorithm-borutashap-8bc238aa1677. [Online; Accessed 8 September 2022].
[25] Wang H, Aligon J, May J, Doumard E, Labroche N, Delpierre C, et al. Discernibility in explanations: Designing more acceptable and meaningful machine learning models for medicine. Comput Struct Biotechnol J 2025.
[26] Shimonovich M, Thomson H, Pearce A, Katikireddi SV. Applying bradford hill to assessing causality in systematic reviews: A transparent approach using process tracing. Res Synth Methods 2024;15(6):826–38. http://dx.doi.org/10.1002/jrsm.1730, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jrsm.1730. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jrsm.1730.
[27] Probst P, Boulesteix A-L, Bischl B. Tunability: Importance of hyperparameters of machine learning algorithms. J Mach Learn Res 2019;20(53):1–32.
[28] Vanschoren J. Meta-learning. Autom Mach Learn: Methods Syst Challenges 2019;35–61.
[29] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 2009;11(1):10–8.
[30] Van Rijn JN, Bischl B, Torgo L, Gao B, Umaashankar V, Fischer S, et al. OpenML: A collaborative science platform. In: Joint European conference on machine learning and knowledge discovery in databases. Springer; 2013, p. 645–9.
[31] Rivolli A, Garcia LP, Soares C, Vanschoren J, de Carvalho AC. Meta-features for meta-learning. Knowl-Based Syst 2022;240:108101.
[32] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM Comput Surv 2018;51(5):1–42.
[33] Singh R, Miller T, Lyons H, Sonenberg L, Velloso E, Vetere F, et al. Directive explanations for actionable explainability in machine learning applications. ACM Trans Interact Intell Syst 2023;13(4). http://dx.doi.org/10.1145/3579363.
[34] Kanagarla K. Explainable AI in data analytics: Enhancing transparency and trust in complex machine learning models. 2024, Available At SSRN 5012468.
[35] Kadir MA, Mosavi A, Sonntag D. Evaluation metrics for XAI: A review, taxonomy, and practical applications. In: 2023 IEEE 27th international conference on intelligent engineering systems. IEEE; 2023, p. 000111–24.
[36] Pawlicki M, Pawlicka A, Uccello F, Szelest S, D'Antonio S, Kozik R, et al. Evaluating the necessity of the multiple metrics for assessing explainable AI: A critical examination. Neurocomputing 2024;602:128282.
[37] Chanson A, Labroche N, Marcel P, Rizzi S, T'kindt V, et al. Automatic generation of comparison notebooks for interactive data exploration. In: Advances in database technology. proceedings 25th international conference on extending database technology, vol. 25, 2022, p. 274–84.
[38] Amer-Yahia S, Marcel P, Peralta V. Data narration for the people: challenges and opportunities. In: 26th international conference on extending database technology. 2023.