

Medical artificial intelligence toolbox (MAIT): an explainable machine learning framework for binary classification, survival modelling, and regression analyses

Ramtin Zargari Marandi (1), Anne Svane Frahm (1), Jens Lundgren (1), Daniel Dawson Murray (1), Maja Milojevic (1)

(1) Centre of Excellence for Health, Immunity and Infections (CHIP), Rigshospitalet, Copenhagen University Hospital, Blegdamsvej 9, DK-2100 Copenhagen, Denmark.

While machine learning offers diverse techniques suitable for exploring various medical research questions, a cohesive synergistic framework can facilitate the integration and understanding of new approaches within unified model development and interpretation. We therefore introduce the Medical Artificial Intelligence Toolbox (MAIT), an explainable, open-source Python pipeline for developing and evaluating binary classification, regression, and survival models on tabular datasets. MAIT addresses key challenges (e.g., high dimensionality, class imbalance, mixed variable types, and missingness) while promoting transparency in reporting (TRIPOD+AI compliant). Offering automated configurations for beginners and customizable source code for experts, MAIT streamlines two primary use cases: Discovery (feature importance via unified scoring, e.g., SHapley Additive exPlanations - SHAP) and Prediction (model development and deployment with optimized solutions). Moreover, MAIT proposes new techniques including fine-tuning of probability threshold in binary classification, translation of cumulative hazard curves to binary classification, enhanced visualizations for model interpretation for mixed data types, and handling censoring through semi-supervised learning, to adapt to a wide set of data constraints and study designs. We provide detailed tutorials on GitHub, using four open-access data sets, to demonstrate how MAIT can be used to improve implementation and interpretation of ML models in medical research.

keywords

data-driven modeling, simulation, mathematical models, supervised learning, clinical applications

Introduction

Machine learning (ML) has demonstrated an ability to enhance data processing across various domains including medicine^{1,2}. Despite the advancement of ML methods and algorithms, there are still important factors creating gaps in the utilization of ML for clinical research. A hindering factor in clinical research may be attributed to the complexities in implementation of ML methodologies tailored to working with medical datasets and study designs³. In contrast, conventional statistical approaches, like logistic regression and Cox models are popular within the medical community due to the extensive historical usage of terminologies and methodologies ingrained over many years.

The rise in popularity of ML can be attributed to several factors. One reason is the ability of ML to incorporate both well-established statistical models like logistic regression, and novel algorithms (e.g., LightGBM⁴), to extract impactful information and solve complex problems. In addition, ML models are often known for their data-driven approach and are less reliant on strong assumptions as compared to some statistical models. Another useful aspect of ML models is the ability to continuously learn from new data to mitigate the risk of model drift over time, which is important in cases such as the use of digital tools for patient screening. The rise in more wide-spread ML use also coincides with the increasing availability of computational infrastructure (e.g., high-performance computing clusters and graphics processing units - GPUs) which is needed to process the ever-growing amounts of big data⁵.

In medical research, non-image patient data is often organized in tabular formats with features or risk factors as columns and individuals as rows. Binary classification within these data structures serves two critical roles: 1) discovery - where patterns and predictive relationships between variables are identified, and 2) prediction – which focuses on building deployment-ready models to forecast outcomes from current health indicators. Prediction models can be used as diagnostic tools to aid in immediate clinical interventions, or as prognostic indicators, anticipating future medical needs. For time-to-event data, survival analysis using methods like random survival forests offers an alternative analytical approach that accounts for censoring and avoids the potential biases inherent to standard binary classification approaches.

MAIT is a highly versatile tool allowing users to address the multifaceted challenges of ML applications within medical research, particularly when confronted with high-dimensional datasets with mixed variable types and missing values. It provides an accessible and customizable ML framework for developing models with varying levels of user expertise while maintaining transparency aligned with the TRIPOD+AI statement⁶. MAIT automates many data preprocessing and modeling steps while allowing for customization of the processes within a standard framework. In addition, MAIT bridges the analytical gap between survival analyses and binary classification, an integral aspect for clinical decision-making based on ML analyses. The novelty of MAIT relies on its implemented techniques used to facilitate the preprocessing of tabular data for compatibility for a wide variety of algorithms, to provide comprehensive model evaluation and validation, and allow for unified and multi-level model interpretation. Consequently, this consolidation of methodologies into one cohesive synergistic system boosts the potential to clarify complex ML applications in medical research. Our aims are therefore to provide a semi-automated framework in a form of an explainable ML pipeline, where there is a reasonable level of human involvement in ML design⁷ while maintaining a consistent framework for model evaluation and reporting, as well as addressing some common technical constraints.

Overview

The current version of MAIT, v. 1.0.0, is implemented in a Jupyter Notebook using Python packages (modules) commonly employed in ML tasks like Scikit-Learn⁸. MAIT is executable within a Conda environment from Anaconda Software Distribution (Anaconda Inc.) as instructed in MAIT documentations (publicly available on GitHub, <https://github.com/PERSIMUNE/MAIT>, containing instructions, tutorials for common use cases, and details on all the methods and algorithms). The environment to execute MAIT is also available as a Docker Image (Docker Inc.). Here, we provide a summary of the features and functionalities of MAIT (Figure 1), as well as additional documentation thoroughly describing the technical details.

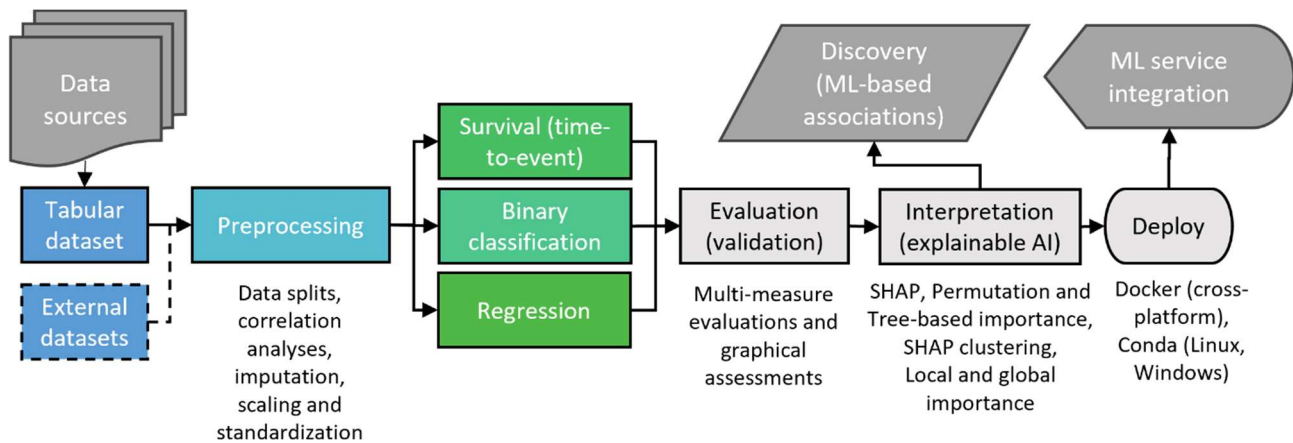


Figure 1 Overview of the analytical blocks of MAIT. It includes several methods and optional functions to address common methodological challenges for binary classification, survival, and regression analyses. The pipeline begins with user-defined settings selecting the methods to use and providing information about the input data. First, the data is imported and quality checked, then MAIT's core functionality of binary classification is applied. The pipeline also allows for survival and regression modeling. The blocks are in a sequence of code chunks within a Jupyter Notebook file for readability. For a complete list of all MAIT functionalities see <https://github.com/PERSIMUNE/MAIT>. Acronyms: area under the receiver operating characteristics curve (AUC), area under the precision recall curve (PR-AUC), Matthews correlation coefficient (MCC), positive predictive value (PPV), negative predictive value (NPV), SHapley Additive exPlanations (SHAP), and histogram-based gradient boosting classifier (HGBC).

Pipeline initiation

For pipeline initiation, analysis metadata (e.g., data localization, definition of variables, methods and algorithms to use) and computational resources (e.g., GPU usage) must first be configured. There are also extra options available to the user, such as enabling feature selection and its detailed options (e.g., number of features). MAIT requires datasets for ML analyses to be structured as a tabular dataset where each row is an observation (e.g., patient), and each column is a variable, such as a patient characteristics like age or gender. Such formats are common for electronic health records stored as tab- or comma-separated values and excel files. In binary classification, a dataset is required to have a column for the outcome variable in a binary (i.e., two categories). In case of outcome censoring (e.g., lost to follow up), semi-supervised learning, through a label propagation method⁹, is available to address relabeling of instances with unknown labels.

For diagnosis or prognosis, validation on an independent dataset, usually called a test set, is necessary. The option for data splitting of a single dataset into development and testing sets is therefore available in MAIT. Alternatively, two separate, independent datasets can be used for development and testing. MAIT ensures that all data processing steps are performed independently (in isolation) for each dataset (i.e., development or train(ing), test(ing), and external validation sets) to avoid data leakage¹⁰. In case without data splitting, the pipeline can serve for association studies based on ML models for discovery purposes rather than for deployment of models for prediction.

Data representativeness is crucial for fair assessments of model performance and generalizability¹¹. Depending on the data, there may be multiple criteria that need consideration to ensure that both the development and test sets adequately represent the underlying population. For instance, if one population sub-group or patient characteristic accounts for 10% of all samples, it is important to maintain a similar proportion in both sets. In such cases, if the outcome, such as mortality with a prevalence of 20%, is considered, the two factors can be combined in MAIT to form a composite feature for stratified data splitting.

The pipeline checks data quality for missingness, rare categories, and feature variance. MAIT also assesses correlation of the features to each other and the binary outcome variable by Spearman's rank correlation. Spearman's rank correlation is often preferred over Pearson's correlation due to its ability to capture non-linear relationships, be more robust to outliers, and not rely on assumptions of normality. To complement Spearman's rank correlation using a non-ranking method, Point-Biserial correlation and mutual information are used to assess the correlation of all features with the binary outcome. MAIT also allows filtering out of highly missing instances and features, merging highly rare categories in categorical features, and outlier detection via the isolation forest algorithm¹². The data exploration analyses, performed prior to ML, provide preliminary assessments of the data distribution and the strength of associations in the data.

MAIT includes algorithms that can handle missingness and categorical features, like CatBoost¹³ and LightGBM⁴, as well as algorithms that require some data transformation. As such, one-hot encoding is applied for categorical variables to transform them into binary variables and k-nearest neighbors is used to impute missingness in continuous variables (*KNNImpute* function in *Scikit-Learn*). An optional data scaling method based on interquartile range is also available (e.g., *robust_scale* function in *Scikit-Learn*). Data scaling is

optional due to tree-based ensemble models being highly robust to skewed data distributions and outliers, and use of the original data distribution makes the interpretation of the models straightforward.

Another factor that can cause technical constraints in ML pipelines by increasing computational costs and through the presence of noise and multicollinearity, is high dimensionality, which is common in some ML datasets (e.g., multi-omics). This is addressed in MAIT by applying feature selection using minimum redundancy maximum relevance (mRMR) algorithm¹⁴ to select the most relevant features based on the development set.

Binary classification

Binary classification simplifies decision making by providing two possible prediction outcomes (e.g., survival vs. death). MAIT includes a selection of seven algorithms for binary classification: (1) LightGBM, (2) CatBoost, (3) Random Forest¹⁵, (4) HGBC, (5) QLattice¹⁶, as well as (6) Logistic Regression, and (7) (Gaussian) Naïve Bayes. The list of algorithms predominantly consists of tree-based ensemble models (1-4), renowned for their versatility and high performance. Simpler models, i.e., Naïve Bayes and Logistic Regression, are used as statistical alternatives to the tree-based ensemble models for performance comparison. QLattice, a more recent algorithm, can provide higher transparency than the other algorithms by producing a closed-form mathematical equation and a block diagram for visualizing feature interactions related to the outcome variable.

The models include free parameters, i.e., hyperparameters, which require configuration and tuning for optimal performance¹⁷. Hyperparameters, (e.g., tree depth in ensemble models, L1 regularization in logistic regression¹⁸, etc.) are important for reasons such as regularization to counteract overfitting¹⁸. Certain parameters are particularly sensitive to variations in dataset characteristics, including the number of features and samples, and class distribution. Typically, these parameters are determined by exploring a parameter space using hyperparameter tuning methods or are set manually based on dataset attributes. MAIT allows hyperparameter tuning via random search and cross validation where the search space is adaptive to dataset characteristics. Hyperparameter tuning can be iterated multiple times. The cross-validation procedure is nested within the overarching cross-validation framework used for comparing different models. Evaluation metrics such as area under the receiver operating characteristic curve (AUC) and precision-recall curve (PR-AUC) are employed during hyperparameter tuning to assess model performance.

For model performance assessment and ensuring consistency across subsets of the development set, cross-validation is conducted, reporting the mean and standard deviation of various evaluation metrics. The model exhibiting the highest average performance across folds, as determined by multiple classification metrics is selected for further analysis. Relying solely on a single performance measure for model selection can introduce bias, particularly in the context of imbalanced datasets¹⁹. Therefore, the cross-validation procedure is stratified by the outcome variable to address class distribution imbalances and minimize bias in classification tasks.

MAIT employs various techniques to handle class imbalance²⁰ where one class is substantially more prevalent than the other. Sample weights are computed based on class proportions and utilized in the models to ensure balanced learning and prevent biased learning outcomes. Model performance for binary classification is evaluated using multiple metrics²¹ to mitigate the risk of biased assessment associated with a single metric, and includes computation of several metrics: AUC, PR-AUC, MCC, positive predictive value, negative predictive value, sensitivity, specificity, F1 score, balanced accuracy, and Brier score. These metrics are recognized as robust measures for evaluating binary classification models²². The best performing model, determined by the highest grand average of MCC, AUC, and PR-AUC, is subsequently trained on the entire

development set and evaluated on an unseen test set to assess its generalizability. Visualizations of ROC curves, precision recall curve, and confusion matrices enrich the evaluation process. Moreover, oversampling methods are available in MAIT to further address class imbalance.

A core component of MAIT to improve its utility for clinical research is the utilization of methods to assess model behavior, prominently featuring SHAP and other compatible methods tailored to each model type. Four methods for model interpretation are available in MAIT: SHAP, permutation-based feature importance, tree-based feature importance, and QLattice-specific feature diagram. Feature importance is estimated for each model during cross-validation and on the test set if data splitting is employed. Feature importance based only on correctly classified samples is also available in MAIT and cluster analysis based on SHAP values of features can be employed to uncover unique patient profiles²³. As a functionality of MAIT, feature importance is depicted for each of the detected clusters as well as the model performance for the clusters. A unique capability of MAIT is the enhancement of SHAP plots by displaying feature categories.

MAIT provides clinical utility assessment through cost-sensitive decision curve analysis for binary classification tasks by comparing the net benefit²⁴ of a selected model to alternative models, extreme cases (e.g., all negatives), and random decisions. Additionally, MAIT enables model calibration using isotonic regression and conformal predictions, which is recommended when there is sufficient data as the approach requires an extra data split for a calibration set. MAIT also provides the option to optimize the probability threshold (cut-off value between 0 and 1) for classification models. In this scenario, the mean value of predicted probabilities for both classes is calculated for each fold in cross-validation, with the average used as an empirical estimation to approach the optimal probability threshold. The probability threshold used in each fold's model performance calculation is the median of estimated probability thresholds from previous folds, with the initial threshold value set as the proportion of samples from the minority class in the total number of samples. An additional option (model uncertainty reduction module) is proposed in MAIT to discard uncertain predictions determined by SHAP values and predicted probabilities in close proximity to the chance (neutral) level.

Survival analysis

Many datasets contain information suitable for survival analysis, particularly evident in longitudinal data from cohort studies. Such datasets often feature a baseline date (e.g., transplant time) and an event occurring subsequent to this date (e.g., post-transplant complications or death)⁵. In these scenarios, the dataset may include a time-to-event variable spanning a follow-up period, facilitating the development and evaluation of survival models implemented by the *scikit-survival* package²⁵, such as random survival forest (RSF) model and Cox's proportional hazards model (CPH) with elastic net penalty.

A typical task formulation involves aggregating observations from the same patients, such as for the first month post-transplant, to predict the likelihood of developing complications within a subsequent observation period, e.g., one-year post-aggregation. Patients experiencing events beyond the observation period are censored, while those within the period have their time-to-event recorded from baseline to event date. Patients without events are assigned the time-to-event as the time to their last follow-up within the observation period.

Performance comparison of the RSF and CPH survival models typically relies on concordance index (CI) and integrated Brier score (IBS)²⁶. Translation of survival model results to binary classification, as proposed in MAIT, involves generating cumulative hazard curves over time for all samples in the development set. The median cumulative hazard for each class (e.g., event and event-free) is computed, and the Euclidean distance of test set samples' cumulative hazard curves to these medians determines class assignment. This approach

enables the interpretation of survival model outputs as predicted cumulative hazard curves transformed into predicted binary classes where the prediction performance of the survival model can be presented as a confusion matrix like a binary classification model.

Additionally, time-updated AUC and predicted risk scores are reported, alongside visualization of cumulative hazard curves. The survival modeling component is considered supplementary to binary classification when time-to-event data is available, enhancing the pipeline's analytical scope. RSF is favored over CPH for its ability to capture nonlinear information and feature interactions, and is interpreted by default, using feature permutation and SHAP methods via *SurvSHAP(t)*²⁷. A great advantage of the survival models compared to the binary classification models is that censoring is addressed without making assumptions.

Regression analysis

Regression analysis is a common approach used for predicting continuous outcome variables. In this context, two models are available: the Random Forest Regressor and Linear Regression. Both models are widely recognized and utilized. However, the Random Forest Regressor is typically expected to outperform the Linear Regression model in most scenarios, thus it is prioritized for explanation using the SHAP method.

Benchmarking report and exporting

Upon completion of a pipeline run, the user will be provided with a detailed report in an HTML file and a folder (directory) containing all figures and tables generated in the user-specified analyses. The HTML file includes the outputs of all blocks of executed codes including figures describing models and datasets as well as tables featuring the values in benchmarking models (e.g., cross validation) that could easily be reviewed and compiled for publications.

Software framework

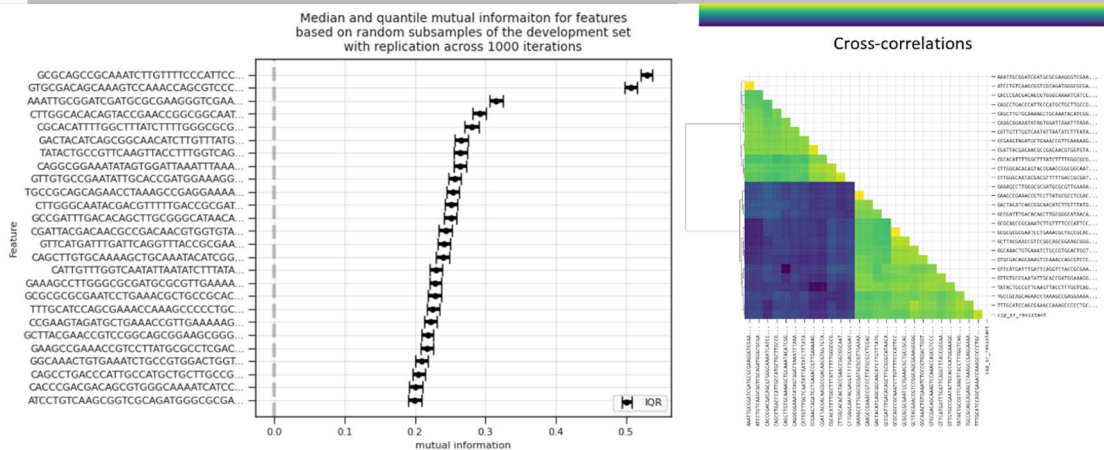
MAIT is formatted as a monolithic notebook with sequential operations. The current version of MAIT offers a stable implementation compatible with both Linux and Windows operating systems. Additionally, it is distributed as a Docker image, enhancing its stability and reproducibility. The development strategy of MAIT prioritized simplicity for ease of comprehension and customization. Users can execute MAIT independently or within another Jupyter Notebook, leveraging tools like the *papermill* package for configuring multiple parameters. Customizability extends to the selective inclusion or exclusion of features to optimize runtime, such as limiting the number of models or bypassing hyperparameter tuning. To maximize efficiency, MAIT employs vectorization and parallelization techniques, with GPU utilization enabled for compatible models like CatBoost. Parallel processing is implemented for several functions to speed up bootstrapping methods for correlation analyses (point-biserial and mutual information), hyperparameter tuning in QLattice model, and extracting SHAP values for survival models. The suite of models and methods within MAIT is designed to accommodate a broad range of ML studies focusing on tabular data. Successful implementation of ML analyses relies on robust, well-documented Python packages for ML, statistical analyses, visualization, and model interpretation, ensuring a solid foundation for future updates and cross-platform compatibility. The most important advantage of MAIT's monolithic notebook form is that it provides a self-contained, reproducible workflow, making it easy to follow the entire process from start to finish at the cost of complex scalability as it grows.

Tutorials and practical examples.

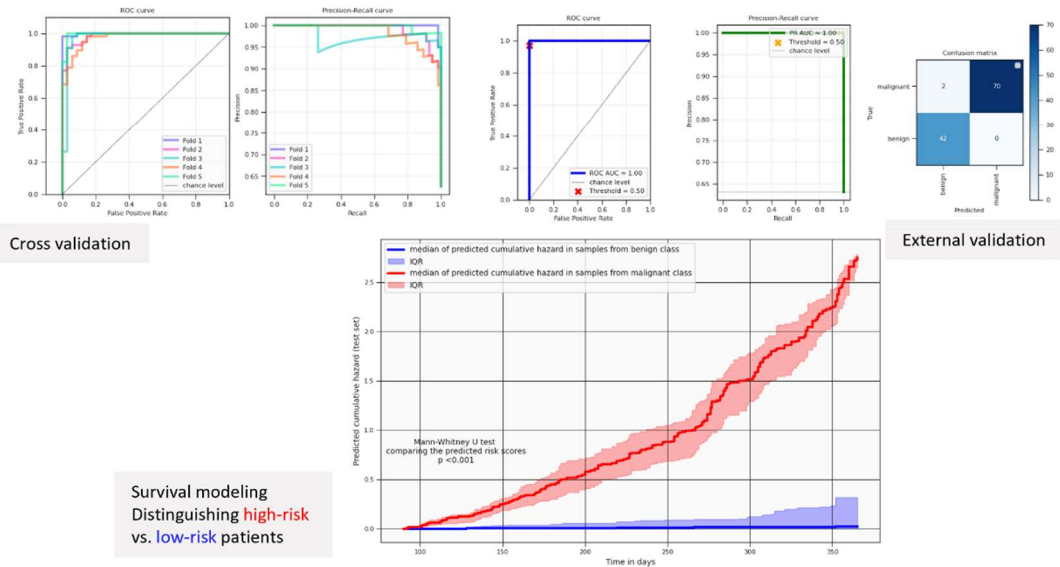
We have curated four tutorials based on four medical datasets that provide practical use cases for MAIT. The tutorials include examples of how to conduct studies based on classification, regression, and survival analyses. One of the tutorials is based on the Wisconsin Breast Cancer dataset²⁸ that includes features

computed from digitized images of fine needle aspirate of breast masses, with the primary purpose of the ML task being classification of breast tumors as benign or malignant. The tutorial on the Dementia Prediction Dataset²⁹ presents an example of a binary classification task when both categorical and continuous features are available. In addition, we provided two tutorials based on two datasets that include genetic data on antibiotic resistance³⁰. The tutorials demonstrate the strengths and versatility of MAIT and provide examples of robust prediction models. Each tutorial's dataset details and results are accessible on our GitHub repository, allowing users to select the most relevant example to guide their machine learning study needs (see Figure 2 as examples of MAIT outputs for different analyses).

(1) Examples of data exploration



(2) Examples of modeling, evaluation and validation



(3) Examples of model interpretation – explainable AI

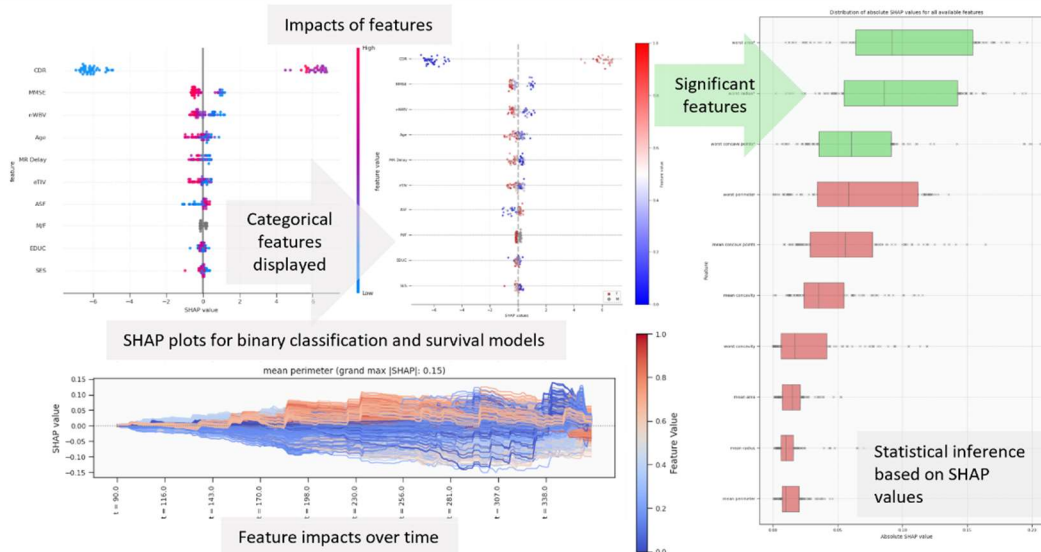


Figure 2 Examples of outputs from tutorials using MAIT. A subset of output figures is depicted from each analytical step as a result of conducting machine learning analyses using MAIT. The process begins with data exploration where figures display the strength of associations between variables, and is followed by model validation and evaluation exemplified by cross validation results. Finally, the associations and impacts of the variables (features) for each model type are illustrated by SHAP plots.

Strengths and limitations of MAIT

An overview of MAIT's features is outlined in Table 1, and briefly describes key strengths and limitations of MAIT. This overview can guide researchers on the suitability of MAIT for their study purposes in comparison to AutoML frameworks. The MAIT vs AutoML comparison section lists several novel approaches incorporated into MAIT to resolve bottlenecks in ML analysis with focus on explainable AI.

Table 1 overview of scientific and technical features of MAIT. Unique functionalities provided in MAIT are outlined in the list, as well as its limitations and comparison with AutoML frameworks in general terms.

Features	Description
Strengths of MAIT	<p>Data Quality & Robustness</p> <ul style="list-style-type: none"> Handles imbalanced data and incomplete datasets <p>Explainability & Transparency</p> <ul style="list-style-type: none"> Model-specific and model-agnostic explainability <p>Standardization & Compliance</p> <ul style="list-style-type: none"> Standardizes ML pipelines – compatible with TRIPOD+AI reporting framework <p>Comprehensive Evaluation</p> <ul style="list-style-type: none"> Multi-metrics performance measurements and optimization <p>Adaptability & Efficiency</p> <ul style="list-style-type: none"> Adaptive to data characteristics – e.g., for hyperparameter tuning and visualization <p>Reliability Enhancement</p> <ul style="list-style-type: none"> Model uncertainty reduction module – to discard uncertain predictions <p>Effective Communication</p> <ul style="list-style-type: none"> Transparent and convenient reporting of scientific and technical ML results <p>Research & Development Flexibility</p> <ul style="list-style-type: none"> Customizable – several selective options for experimentation and hypotheses testing <p>Broad Applicability</p> <ul style="list-style-type: none"> Simultaneous support for binary classification, regression, and survival models <p>Feature Analysis</p> <ul style="list-style-type: none"> Unified feature importance (normalization of feature scores from multiple methods) Explores interactions between pairs of features based on SHAP and permutation methods <p>Evaluation Scope</p> <ul style="list-style-type: none"> Ubiquitous analyses on multiple datasets (training, testing, and external validation) <p>Healthcare & Time-to-Event Applications</p> <ul style="list-style-type: none"> Translation of survival curves to binary classifications Cost-sensitive model evaluation for clinical utility <p>Advanced Analytics Capabilities</p> <ul style="list-style-type: none"> Statistical inference in significance of feature importance Model interpretation on all levels (population, subgroups, and individuals) Multi-level model interpretation (e.g., within cross-validation, after excluding incorrect predictions, within data-driven subgroups)
Limitations of MAIT	<ul style="list-style-type: none"> Does not handle multi-class classification directly Does not support multi-task learning and causal ML Does not include deep learning
MAIT vs. AutoML	<ul style="list-style-type: none"> More efficient than AutoML usually rely on testing all available configuration More interpretable than AutoML as they usually provide very limited options More customizable than AutoML for manual modifications

- | | |
|--|---|
| | <ul style="list-style-type: none"> • Less prone to bias than AutoML that tend to produce misleading results without expert supervision |
|--|---|

An important focus of MAIT is the derivation of feature importance. Multi-level model interpretation is used to determine feature importance for different subsets of the data (e.g., data-driven clusters based on SHAP values²³) and for different conditions (e.g., using only correct predictions in binary classification). Multi-level model interpretation contributes to guided determination of feature importance as opposed to, for instance, using the whole dataset to determine feature importance without consideration that samples with incorrect predictions could be adding irrelevant information to complicate interpretation. In addition, MAIT provides a significance test using bootstrap subsampling (95% confidence) that identifies statistically significant features based on their SHAP values, marking those that have an Interquartile Range (IQR) crossing a data-driven threshold less than 5% of the time. Furthermore, the availability of the other methods to determine feature importance like the permutation method¹⁵, strengthens model interpretation on different levels (e.g., within cross validation and on the test set). MAIT also analyzes feature interactions using two methods: SHAP values, and a feature permutation method, which calculates importance scores for all possible feature pairs. These are key contributions of MAIT in addition to rich visualizations that can reveal important aspects of models and the underlying data, addressing shortcomings of existing software solutions (e.g., SHAP plots with the presence of categorical features).

To the best of our knowledge, MAIT provides a unique solution to unify and compare findings from a survival model (time-to-event outcome variable) and its binary classification counterpart (when the outcome variable is defined as binary classes) for the same dataset. It translates predicted cumulative hazard curves to binary classifications by comparing the difference of the predicted curves from a test set with the median curves representing each of the two classes (based on non-censored samples from the training set). The translation to binary classification is useful for decision-making processes where a predicted cumulative hazard curve alone is less interpretable and actionable. This also allows for comparing feature importance between the two models of different types.

AutoML frameworks (like H2O AutoML³¹), an alternative to MAIT, boast ease of use but suffer from significant limitations³². They incur high computational costs due to exhaustive method and model testing, often yielding negligible performance differences. In contrast, MAIT optimizes resource utilization by allowing users to specify desired resources and providing a curated set of methods and models with user guidance. Beyond computational efficiency, AutoML frameworks are criticized for their limited customizability, lack of model transparency, and overemphasis on top-performing models without regard for interpretability—a crucial aspect in clinical research. MAIT addresses these concerns by supplying an open-source, modifiable Jupyter Notebook with configurable pipeline options, and integrated model interpretation methods for clearer reasoning explanations. In AutoML frameworks, there is a risk of overlooking important details such as imbalanced class distribution and stratified data splitting. In contrast, MAIT employs a multi-measure approach to robust validation, offers data imputation and encoding for incomplete datasets, and contributes to model generalizability through adaptive hyperparameter tuning.

An alternative approach to AutoML is to design and implement a new pipeline from scratch, which could be inefficient and prone to errors. Instead, MAIT presents a standardized, extensible framework based on ML best practices^{1,21,33,34}. Notably, MAIT uniquely supports simultaneous development of binary classification, regression, and survival models derived from a single dataset whilst supporting external validation. This feature allows direct head-to-head comparisons across models that are otherwise almost infeasible (due to

technical requirements including identical computerized randomizations and software environments). This capability is facilitated by using the innovative translations of predicted cumulative hazard curves to binary classifications for comparable predictions across the two model types.

MAIT advocates for the human-in-the-loop formalism⁷ while maintaining a balance between arbitrary ML designs and autopilot approaches based on AutoML frameworks for high quality research. The efficient set of models and methods with selective options and room for customisation are the key to approach this aim. The MAIT framework introduces novel synergies, such as combining feature importance rankings (SHAP, permutation, tree-based) and a distance-based approach for survival to binary predictions. Method selections prioritize efficiency and stability, such as mRMR for feature selection and kNN for data imputation, with thoughtful omissions (e.g., SMOTE³⁵, MICE³⁶) to maintain simplicity and performance.

Conclusion

MAIT provides a novel framework for explainable machine learning (ML) research on tabular datasets, synergistically integrating established methodologies with cutting-edge algorithms and novel approaches for improved visualizations to interpret models, supporting statistical inference for improved confidence in feature importance derivations, and model validation. By presenting a standardized, adaptable, and user-friendly platform, MAIT accelerates the exploration of complex datasets, facilitates the development of robust and deployable prediction models tailored to diverse patient profiles. MAIT is a powerful tool with potential applications in enhancing the translational relevance of ML insights in clinical settings and driving advancements in personalized medicine and evidence-based healthcare decision-making.

Funding

This work was supported by the Danish National Research Foundation [DNRF126].

Data availability

Examples of execution of MAIT on publicly available datasets are available at <https://github.com/PERSIMUNE/MAIT>.

Conflict of interest

None declared.

Code availability

The source codes for MAIT are publicly available at <https://github.com/PERSIMUNE/MAIT>.

Acknowledgements

We would like to thank Henrik Lassen at Centre of Excellence for Health, Immunity and Infections (CHIP), Rigshospitalet, Copenhagen University Hospital, Denmark for his help in maintaining CHIP HPC servers, that were the computing infrastructure used to develop and test the pipeline.

References

1. Rajkomar, A., Dean, J. & Kohane, I. Machine Learning in Medicine. *N. Engl. J. Med.* **380**, 1347–1358

(2019).

2. Niazi, S. K. The coming of age of AI/ML in drug discovery, development, clinical testing, and manufacturing: The FDA Perspectives. *Drug Des. Devel. Ther.* **17**, 2691–2725 (2023).
3. Kantidakis, G., Putter, H., Litière, S. & Fiocco, M. Statistical models versus machine learning for competing risks: development and validation of prognostic models. *BMC Med. Res. Methodol.* **23**, 51 (2023).
4. Ke, G. *et al.* LightGBM: A highly efficient gradient boosting decision tree. in *Advances in Neural Information Processing Systems* (2017).
5. Zargari Marandi, R. *et al.* Pre-Transplant Prediction of Acute Graft-versus-Host Disease Using the Gut Microbiome. *Cells* **11**, 4089 (2022).
6. Collins, G. S. *et al.* TRIPOD+ AI statement: updated guidance for reporting clinical prediction models that use regression or machine learning methods. *bmj* **385**, (2024).
7. Crisan, A. & Fiore-Gartland, B. Fits and starts: Enterprise use of automl and the role of humans in the loop. in *Proceedings of the 2021 CHI Conference on human factors in computing systems* 1–15 (2021).
8. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
9. Zhu, X. & Ghahramani, Z. *Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107* (2002).
10. Barnett, J. *et al.* Guiding questions to avoid data leakage in biological machine learning applications. *Nat. Methods* **21**, 1444–1453 (2024).
11. Schat, E., van de Schoot, R., Kouw, W. M., Veen, D. & Mendrik, A. M. The data representativeness criterion: Predicting the performance of supervised classification based on data set similarity. *PLoS One* **15**, e0237009 (2020).
12. Liu, F. T., Ting, K. M. & Zhou, Z.-H. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. from Data* **6**, 1–39 (2012).
13. Dorogush, A. V., Ershov, V. & Gulin, A. CatBoost: gradient boosting with categorical features support. (2018).
14. Peng, H. C., Long, F. H. & Ding, C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1226–1238 (2005).
15. Breiman, L. Random Forest. *Mach. Learn.* **45**, 5–32 (2001).
16. Broløs, K. R. *et al.* An Approach to Symbolic Regression Using Feyn. (2021).
17. Villalobos-Arias, L., Quesada-López, C., Guevara-Coto, J., Martínez, A. & Jenkins, M. Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation. in *Proceedings of the 16th ACM international conference on predictive models and data analytics in software engineering* 31–40 (2020).
18. Lee, S.-I., Lee, H., Abbeel, P. & Ng, A. Y. Efficient L1 regularized logistic regression. in *Aaai* vol. 6 401–408 (2006).

19. Wynants, L. *et al.* Prediction models for diagnosis and prognosis of covid-19 infection: systematic review and critical appraisal. *bmj* **369**, (2020).
20. Boughorbel, S., Jarray, F. & El-Anbari, M. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLoS One* **12**, e0177678 (2017).
21. Chicco, D. & Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* **21**, 6 (2020).
22. Zargari Marandi, R. *et al.* Development of a machine learning model for early prediction of plasma leakage in suspected dengue patients. *PLoS Negl. Trop. Dis.* **17**, e0010758 (2023).
23. Zargari Marandi, R. ExplainerR: an R package to explain machine learning models. *Bioinforma. Adv.* **4**, vbae049 (2024).
24. Vickers, A. J. & Elkin, E. B. Decision curve analysis: A novel method for evaluating prediction models. *Med. Decis. Mak.* (2006) doi:10.1177/0272989X06295361.
25. Pölsterl, S. scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn. *J. Mach. Learn. Res.* **21**, 1–6 (2020).
26. Fernandez, C., Chen, C. S., Gaillard, C. P. & Silva, A. Experimental Comparison of Ensemble Methods and Time-to-Event Analysis Models Through Integrated Brier Score and Concordance Index. *arXiv Prepr. arXiv2403.07460* (2024).
27. Krzyżiński, M., Spytek, M., Baniecki, H. & Biecek, P. SurvSHAP (t): time-dependent explanations of machine learning survival models. *Knowledge-Based Syst.* **262**, 110234 (2023).
28. Wolberg, W., Mangasarian, O., Street, N. & Street, W. Diagnostic Wisconsin Breast Cancer Database. *UCI Machine Learning Repository* (1993) doi:10.24432/C5DW2B.
29. Battineni, G., Chintalapudi, N. & Amenta, F. Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). *Informatics Med. Unlocked* **16**, 100200 (2019).
30. Jaillard, M. *et al.* A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. *PLoS Genet.* **14**, e1007758 (2018).
31. LeDell, E. & Poirier, S. H2o automl: Scalable automatic machine learning. in *Proceedings of the AutoML Workshop at ICML* vol. 2020 (2020).
32. Azevedo, K., Quaranta, L., Calefato, F. & Kalinowski, M. A Multivocal Literature Review on the Benefits and Limitations of Automated Machine Learning Tools. *arXiv Prepr. arXiv2401.11366* (2024).
33. Deo, R. C. Machine learning in medicine. *Circulation* **132**, 1920–1930 (2015).
34. Li, J. J. & Tong, X. Statistical hypothesis testing versus machine learning binary classification: Distinctions and guidelines. *Patterns* **1**, (2020).
35. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002).
36. Van Buuren, S. & Groothuis-Oudshoorn, K. mice: Multivariate imputation by chained equations in R. *J. Stat. Softw.* **45**, 1–67 (2011).