Utrecht University

Automated Interpretable Machine Learning for Medical Classification Tasks

by

Tessel Haagen

Master of Science Submitted to the Artificial Intelligence Graduate Program

in partial fulfillment of the requirements for the degree of

Master of Science

Graduate Program in Artificial Intelligence

Utrecht University

2023

Automated Interpretable Machine Learning
for Medical Classification Tasks

APPROVED BY:

dr. Heysem Kaya .....
(Thesis Supervisor)

dr. Almila Akdag .....

DATE OF APPROVAL: 21.06.2023

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Heysem Kaya, for his guidance, support, and valuable insights throughout the duration of this project. His expertise and feedback have been invaluable in shaping the direction and quality of this work.

I would also like to extend my appreciation to my supervisor from Info Support, Joop Snijder, for providing me with the opportunity to undertake this project. His unwavering support, encouragement, and generous compliments have been a source of motivation and confidence throughout the entire journey. I am truly grateful for his belief in my abilities and for his constant guidance, which has been instrumental in my personal and professional growth.

Furthermore, I would like to acknowledge the invaluable assistance and mentorship provided by Nikki Thissen, my process mentor at Info Support. Our weekly appointments not only challenged me to enhance my soft skills but helped me deal with struggles throughout this project as well.

Lastly, I would like to express my gratitude to Willem Meints for his valuable contributions and expertise in the field of AI. His involvement as my fill-in supervisor in the initial stages of the project greatly enriched the overall outcome.

I am grateful to all those mentioned above and to the countless others who have supported me throughout this journey. Their contributions have played a crucial role in the completion of this project.

ABSTRACT

Automated Interpretable Machine Learning for Medical Classification Tasks

Machine learning (ML) algorithms are increasingly used in high-stake domains like healthcare. While ML systems frequently outperform humans in specific tasks, ensuring safety and transparency is critical in these domains. Interpretability, therefore, plays a crucial role in understanding the decision-making process, auditing and correction of ML models and establishing trust. Furthermore, there is a growing demand for automated machine learning (AutoML) to facilitate model development without expert intervention. However, the combination of interpretability and AutoML has received limited attention thus far. In this study, we propose two objective model-agnostic measures of interpretability to quantify model compactness and explanation stability, embedded within an automated interpretable ML pipeline. We experiment with a set of interpretable models on medical classification tasks reporting the proposed measures along with the predictive performances. We further conduct a user study with domain experts to evaluate the correlation between these measures and the subjective concept of interpretability. Our findings demonstrate the effectiveness of the proposed measures, affirming their success and validating their utility in creating an interpretable automated pipeline.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AI	Artificial Intelligence
AutoML	Automated Machine Learning
DCP	Dominance Classifier Predictor
DT	Decision Tree
ExMo	Explainable AI Model using Inverse Frequency Decision Rules
EBM	Explainable Boosting Machine
FNA	fine needle aspirate
GAM	Generalized Additive Models
GA^2M	Generalized Additive Models plus Interaction
ITR	Information transfer rate
NODE	Neural Oblivious Decision Trees
MIMIC	Medical Information Mart for Intensive Care
ML	Machine Learning
LIME	Local Interpretable Model-Agnostic Explanations
PD	Pharmacodynamic
PK	Pharmacokinetic
RPPR	Reverse Prediction Pattern Recognition
SCS	System Causability Scale
SVM	Support Vector Machine
WDBC	Winconsin Diagnostic Brest Cancer
XAI	eXplainable Artificial Intelligence
XGBoost	eXtreem Gradient Boosting

1. Introduction

Machine learning (ML) is a growing approach for generating decisions, even for high-risk decisions in court [2] or medical field [3]. The fact that ML systems often outperform humans on specific tasks [4] is driving this growth. But the task performance is not the only criterion for ML systems in these high-stake fields. Since the decisions deeply impact human lives, they must be safe and transparent as well. Therefore the growth of using ML models in high-stake domains raises the significance of models' interpretability. It is important humans can check the decision-making process in order to prevent catastrophic failures in real-life decisions, such as a mistaken verdict or incorrect prescription caused by biases in the training data. Hence the decisions made by ML models, and therefore the ML models themselves, must be understood. This is only possible if the model can explain its reasoning because then we can verify whether that reasoning is sound concerning the criteria [5]. Interpretability is, therefore, a crucial component of trust or distrust in ML models [6] and is a necessity in high-stake domains.

The recognition of the need for explanations extends beyond researchers. The introduction of the General Data Protection Regulation (GDPR) in 2018 emphasized the importance of transparency in automated decision-making involving personal data. Article 13 (2f) of the GDPR explicitly requires organizations to provide "meaningful information about the logic involved" in such decision-making processes, granting individuals the "right to explanation" [7]. In parallel, the European Commission established the High-Level Expert Group on AI, comprising esteemed experts tasked with advising on AI-related matters. This group has since published several works, including a set of 7 essentials for trustworthy AI, as detailed in Table 1.1 [1]. Furthermore, on April 21, 2021, the European Commission unveiled the AI Act [8], a proposed legislation aimed at categorizing AI applications into three risk categories. Alongside other regulatory provisions, the AI Act mandates risk management, transparency, and human oversight for high-risk AI systems.

Guidelines for trustworthy AI
Human agency and oversight
Robustness and safety
Privacy and data governance
Transparency
Diversity, non-discrimination, and fairness
Societal and environmental well-being
Accountability

Table 1.1: Guidelines for trustworthy AI, adapted from the European Commission's High-Level Expert Group [1].

The growth of machine learning in the industry also demands hands-free solutions. Building accurate ML models with high performances is an expensive and time-consuming job. Therefore, there is a rise in the field of automated machine learning (AutoML). The idea is to build an ML model without the need for a machine learning expert or data scientist. This can include all sorts of aspects of automation in the building process, such as automating data cleanup, feature selection, data transformation, model selection, criticism, and so on. The field of AutoML aims to make all decisions made during the building in a data-driven, objective, and automated way: the user simply provides data, and the AutoML system automatically determines the approach that performs best for the particular application. Since it has been shown that AutoML can outperform experts in the machine learning field, it will not only lead to a more efficient practice but also a better performance of machine learning [9].

Both of these emerging themes have been the focus of several studies, but hardly any to none combines those into an automated interpretable machine learning pipeline. If the combination of these concepts will be possible, highly risky decisions will be made by machines, automated, effective and interpretable. This will be of great asset for fields where daily highly risked decisions are made, since the decision made by the pipeline will have a higher success rate and is generated automatically, but can be discarded for having the wrong justification.

In this thesis, the possibilities of an automated interpretable machine learning pipeline for multi-class classification that can be used in the medical field will be studied. To test if the pipeline can be used in this field, a user test will be performed with domain experts to see if the pipeline is helpful.

1.1. Research Questions

This thesis intends to address the following question:

(**RQ**) **Is it possible to create an interpretable automated machine learning pipeline for medical classification tasks with an F1-performance comparable with existing non-interpretable tools?**

In this thesis, the scope is limited to the medical industry. The medical industry is an excellent fit since the decisions are critical, and thus they require not only explanations but also interpretable models. If high-risk decisions with the wrong reasons are made, they may have catastrophic consequences. Additionally, since the workload is already excessive, it would be ideal if the decision process will be mostly automated. It would make the process effective, which is crucial.

As this would be the first attempt at interpretable automated machine learning, the thesis will solely concentrate on tabular data, leaving the data-preprocessing step for further work. Comparable F1-performance is defined as a F1-score which is not statistically significantly worse than the existing AutoML tools. We will compare the F1-score with three existing AutoML tools. These tools will be state-of-the art and easy to use. We chose the F1-score since sensitivity is more important than simple accuracy in the medical domain.

To answer the main research question there are a couple of sub-questions that need to be addressed:

(**SQ1**) **Which existing interpretable models are suitable for an automated pipeline?**

Before building an automated pipeline, it must be clear which models will be implemented. A model is suited for Interpretable AutoML if it has no need for human interaction, has an interpretable outcome, and has hyperparameter settings which can be tuned such as regularization methods to boost interpretability. This question will mostly be answered by a literature study.

(**SQ2**) **Which interpretability measures can be used for automatic model selection?**

To evaluate the models trained in the pipeline, a measure for interpretability is needed. Because this measure is used in an automated pipeline, it should be objective and quantitative. To answer this question, we will transform some of the most prominent desiderata of interpretability in the literature into objective quantitative measures. We will report experimental results using these proposed measures as well as predictive performance and analyze them against the rules of thumb.

(**SQ3**) **How interpretable are the resulting models for domain experts?**

To validate the resulting pipeline, a user study will be conducted on domain experts. A dataset that the experts are familiar with will be used and the interpretability will be validated with subjective measures through a standardized questionnaire. The literature study will provide a suitable questionnaire.

2. Background and Related work

2.1. Interpretability

It is challenging to define the ambiguous concept of interpretability since it is a broad, vague, and generally known concept. Murdoch et al. define interpretable machine learning as "the extraction of relevant knowledge from the model concerning relationships either contained in data or learned by the model". Where knowledge is relevant if it provides insight for a particular audience into a chosen problem [10]. But, this definition includes the heading explainable machine learning, while it is important to distinguish the two. Explainable AI (XAI) is where one attempts to explain a black box model by approximating its predictions by a simpler model [6], while interpretable models are understood by their own reasoning for the decisions.

According to Miller, interpretability is "the degree to which a human can understand the cause of a decision" [11]. However, the definition of a machine learning model's interpretability is more complex, since humans need to understand the causes of all decisions the model makes. Therefore, interpretability is better defined as the degree to which a human can consistently rationally predict the model's result [12]. This is known as human-simulatability.

A model needs to meet a few requirements in order to be interpretable and achieve human-simulatability. First of all, the input and features must be intelligible, since these are the primary components of the reasoning process. Secondly, regularizers for the complexity of the model must be added. Lage et al. [13] describe three different types of complexity that must be reduced to create an interpretable model: model size, cognitive chunks, and repeated terms. Using a small number of features is crucial because most people are unable to process a huge amount of data. It is well known that humans can only hold around seven items in their working-memory [14], suggesting explanations that are understandable by humans should adhere to some sort of capacity

restriction. These items can correspond to complex cognitive chunks, representing a complex item in multiple meaningful units. The number of times that important features appear in the model must be kept to a minimum; otherwise, users may find it difficult to complete the tasks since they will have to carefully read each justification where a relevant feature appears to provide the correct answer.

A black box model is the opposite of an interpretable model. Either the model's internal workings, such as neural networks, are too complicated for humans to comprehend, or it provides a formula that is too complex to understand. Furthermore, these black box models frequently predict correct outcomes for incorrect reasons, a phenomenon known as the 'Clever Hans' effect [6]. Unfortunately, it is nearly impossible to correct these incorrect causations since the model is not understood. The field of XAI tries to create insight into these black-box models, by applying an explanation method on the outcomes. But there are several problems with explainable AI; explanations are often not reliable and can be misleading [15]. Furthermore, different methods can give conflicting explanations, which is called the disagreement problem [16]. It raises the issue of which explanation to trust.

Unfortunately, black box models generally have better performances, since most black box models outperform white box models on the same task. Therefore, research says there is a trade-off between accuracy and interpretability, a trade-off that is not wanted in a high-risk decision-making process [17]. However, Rudin argues this trade-off is a myth [15], claiming that there is no significant difference between complex models and simpler models when the task has structured data with meaningful features.

2.1.1. Interpretable models

This section intends to answer **SQ1**.

There are various families of interpretable models, such as rule lists, decision trees, and linear models. If the model-specific conditions are satisfied, these common

three families are interpretable. Implementing these requirements, however, frequently results in a decline in performance. Many new models and methods have been developed over the last few years to reduce the accuracy gap between deep learning models and interpretable models. A brief overview of the different approaches is provided below.

2.1.1.1. Decision Trees. A *decision tree* is perhaps the most used machine learning model. Tree-based models repeatedly separate the data with cutoff values in the features, resulting in a tree structure with nodes and leafs. Each node splits the data into children nodes and each leaf has a final prediction based on the majority of the training data in the leaf. The resulting tree is easily visualized, an example is shown in figure 2.1.

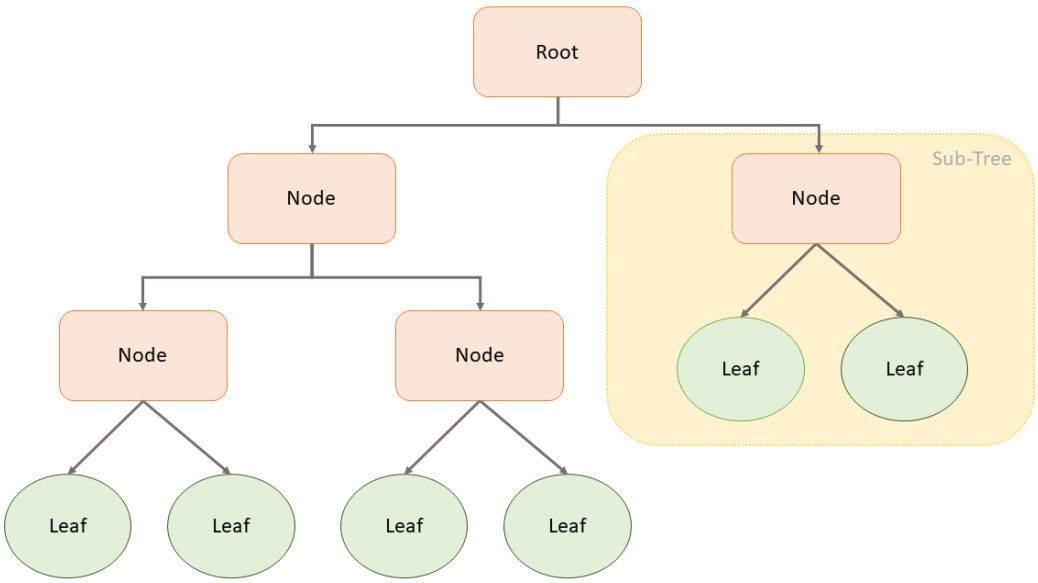


Figure 2.1: Example visualization of a decision tree.

The cutoff values are learned with the training data. The principle is to select the best split, with an attribute selection measure, split the data and repeat this for the children to create sub-trees until all samples in a node have the same class or the impurity measure reduces below a threshold, or there are no attributes left to split

on. These nodes become leaves. There are numerous methods for growing trees, with differences in the tree structure, criteria for finding the best splits, choosing which nodes to set as leafs and how to estimate the prediction in the leaves.

Decision trees are white-box models, since they share internal decision-making logic by the visualization of the tree or a textual explanation with all rules listed, with a conjunction of all conditions of each node from root to leaf. But to be fully interpretable, there has to be some regularization to avoid repeating terms and restricting the size of the tree to the cognitive capacity of humans.

2.1.1.2. The Family of Linear Models. Linear models generate a formula to create a best-fit line to predict unknown values. Linear models tend to be interpretable since it is very easy to see the contribution of each feature to the prediction by their weights. A range of linear models exist. A hierarchy of the linear family is briefly shown in figure 2.2.

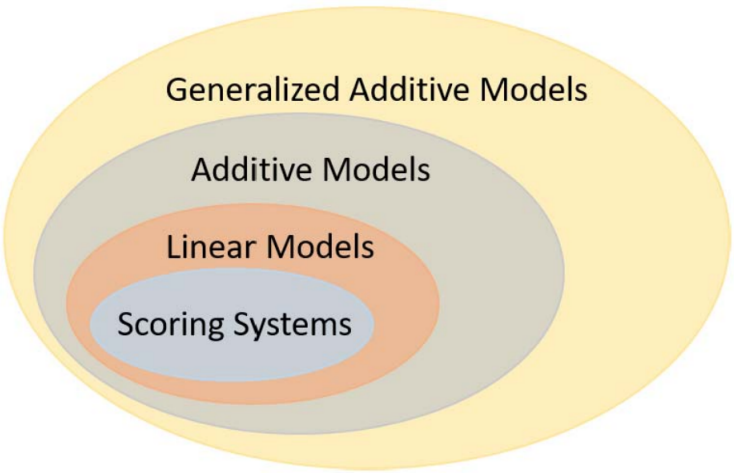


Figure 2.2: Hierarchy of the linear family.
Figure adapted from [6]

- *Scoring systems* are very simple linear classification models where users add, subtract and multiply a few features with their weights in order to make a prediction. These models are used a lot in the medical industry to calculate risks for certain

conditions because they are easy to use and highly interpretable. Feature weights can only be integers, which makes the computation easy and human-friendly to work with.

- *Linear regression* is a very common machine learning method. The idea is to fit the following function to the training data:

$$y = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \epsilon, \tag{2.1}$$

where $\beta_1, \ldots \beta_n$ are the linear coefficients, β_0 is a constant and ϵ are random variables representing the possible errors. The coefficients are learned by optimizing the sum of the squared errors:

$$SE = \sum_{n}^{i=1} (Y_i - (\beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n))^2. \tag{2.2}$$

The lower the summed error SE, the more closely the function matches the training data.

- *Generalized Additive Model* [18], also known as GAM, is an improved linear model. GAMs disregard the requirement that the relationship must be a simple weighted sum and instead assume that the outcome can be represented by a sum of arbitrary functions of each feature. Hence GAMs allow us to learn non-linear features. By simply replacing the linear coefficients with splines, complex flexible functions, in the linear model function, we created a GAM.

$$g(E\,[y]) = w_0 + s_1(x_1) + s_2(x_2) + \ldots + s_n(x_n), \tag{2.3}$$

here w_0 is the basis weight, s_i are the smoothed functions called splines $s_i(x) = \sum_{k=1}^{k} \beta_k b_k(x)$ where β_k are the weights again and b_k is a basis expansion of x. GAMs are trained with fitting methods. The original GAM estimates the splines using non-parametric smoothers through iterative smoothing of partial residuals, known as the backfitting algorithm.

An extension on GAMs is Generalized Additive Models plus Interactions (GA^2M)

[19]. Lou et al. proposed adding two-dimensional interactions:

$$g(E[y]) = \sum s_i(x_i) + \sum s_{ij}(x_i, x_j). \tag{2.4}$$

Furthermore, they propose a method called FAST to measure and rank the pairwise interactions, to scale the used interacted features.

- A very popular interpretable machine learning method is *Explainable Boosting Machine* which is provided in the python package InterpretML [20]. EBM is an improved GA^{2}M since EBM learns the feature functions using modern machine learning techniques like bagging and gradient boosting. The training is restricted to one feature at a time and round-robin cycles are used through features to mitigate the effects of co-linearity.
- *NODE-GAM or NODE-GA^{2}M* is a combination of designs from GAMs and Neural Oblivious Decision Trees (NODEs) [21]. Chang et al. [22] introduced this new approach as an interpretable model with the advantages of deep learning models. To make NODE a GAM, they made three key changes to avoid any feature interactions in the architecture. This will make NODE-GAMs interpretable as well. They compared the performance of the model on six popular binary classification datasets and two regression datasets with different GAMs, GA^{2}Ms, and full complexity models NODE, eXtreme Gradient Boosting, and Random Forest. They showed all GAMs perform similarly, also the NODE-GA^{2}M (0.808) is similar to EBM-GA^{2}M (0.812). Furthermore, NODE-GA^{2}M slightly outperforms full-complexity models (NODE: 0.737, XGB: 0.808, RF: 0.301).

As mentioned, linear models are interpreted by the learned weights. Depending on the type of feature, different interpretations are used for the weights. Increasing a numerical feature by one unit changes the estimated outcome by its weight. Changing a binary feature from category changes the estimated outcome by its weight as well. For categorical features which are one-hot-encoded, each category will be interpreted separately, with the same interpretation as binary features. The constant β_0, also known as the intercept, is the model's prediction if all the features are set to 0 or

default. The interpretation is only meaningful if the features have been standardized because the intercept then reflects the predicted outcome if all features are at their mean value. Secondly, the importance of a feature can be calculated by the absolute value of its t-statistics:

$$t_{\beta_k} = \frac{\beta_j}{SE(\beta_j)}. \tag{2.5}$$

The importance increases with increasing weight and a lower standard error (= the more certain it is the correct value).

2.1.1.3. Rule lists. Another popular interpretable machine learning technique is *Rule lists*. The algorithm learns a list of IF-THEN statements with a decision rule in the condition and a prediction score in the conclusion. If neither the condition nor the list itself is too long, these are easily interpreted lists.

- *Explainable AI Model using Inverse Frequency Decision Rules* (ExMo) is a method to compute these decision rules, proposed by Mainali et al. [23]. The ExMo model extracts the decision rules using term-frequency-inverse document frequency (TF-IDF). The experiment conducted by the authors is done on the IEEE-CIS Fraud dataset and compared eXtreme Gradient Boosting, a three-layered neural network, the original Bayesian Rule List, and three different settings of ExMo. The results show that ExMo (0.89) becomes close to the accuracy of the neural network (0.92) and XGBoost (0.93) and provides a significant improvement over the original BRL algorithm (0.69).
- *The Dominance Classifier and Predictor algorithm boosted with Reverse Prediction Pattern Recognition* is a new approach proposed by Neuhaus and Kovalerchuk [24]. First, the dominance classifier structure is produced, this is a table containing intervals and the number of cases of each class on the training data within the respective interval on each predictor attribute. The dominant class is then computed for a given case in each attribute. Next, there is a voting method used to combine dominances for class prediction, which produces sim-

ple single-attribute prediction rules, and afterwards, the voting method is used for combining the single-attribute prediction rules into a classification rule. This method is boosted by Reverse Prediction Pattern Recognition, and can only be applied to binary classification. First, the elements of the DCP algorithm as Boolean vectors and training cases that are misclassified are found. Next all unique pairs for DCP False-Negative and False-Positive from the training data are discovered. These points are then found in the validation or test data and the prediction by DCP is then reversed. This method is tested by an experiment with three different datasets and compared with several neural network methods. On all three datasets, DCP/RPPR outperforms all black box models.

2.1.1.4. Reinforcement Learning. There has been some research towards *Interpretable Reinforcement Learning*, as summarized by Glanois et al. [25]. They argue that an RL agent has different AI tasks to perform and therefore the whole progress has different components that need to be interpretable. First and foremost, the input must be interpretable, as must the processing of this data. To accomplish this, a symbolic representation of the environment needs to be learned, but this is very hard and needed to be made by human experts. Secondly, the transition and preference models need to be interpretable, since the reasoning behind the decision-making progress is in these models. Recent work on interpretable transition models is based on neural networks in order to process high-dimensional inputs and has adopted an object-centric approach. While neural networks hinder the intelligibility of the method, the decomposition into object dynamics can help scale and add transparency to the transition model. Research in interpretable preference models is not advanced enough, though these models are even more important to be interpretable since these will give more insight into the environment dynamics. Lastly, the decision-making itself can be ensured to be interpretable by its policy. There exist various approaches to obtain interpretable policies, but the main issue here is to achieve good enough interpretability, the task must be simple enough.

2.1.1.5. Interactive learning. Another promising approach would be an *Interactive Interpretable Machine Learning* model. As far as I'm aware, the literature hasn't addressed this subject. It is based on the idea that there should be a feedback loop between the user and the model in order for the model to learn from its errors. Particularly with interpretable machine learning, it would be simple to pinpoint the causation of the error because its justification is provided. It has the potential to improve performance and foster user confidence in the model.

Although interactive interpretable machine learning has not yet been explored, explainable interaction has. Teso et al. propose the framework of *Explanatory Interactive Machine Learning* [26], where the user interacts by both responding to the query and correcting the explanation given by the model at each stage. In their study, they used an SVM and a neural network with Local Interpretable Model-Agnostic Explanations (LIME) [27] to demonstrate that this framework boosts the predictive and explanatory powers of the model. LIME is a post-hoc local explanation method, which fits a surrogate model to approximate weights for data points to explain each individual prediction. Furthermore, by conducting a user study, the writers also conclude this cooperative process can encourage or discourage trust in the model.

2.1.2. Methods for measuring interpretability

Since a uniform definition is a challenge in the research on interpretability, it is difficult to compare the quality of interpretable models across studies using benchmark tests.

To provide the predictions of a machine learning model with explanations, an explanation method is used. An explanation typically makes a clear connection between the features values of an instance to its model prediction in a humanly understandable way. There are several properties of explanation methods that can be used to assess the quality of a method. Expressive Power refers to the structure of the explanations the method produces. The level of translucency specifies the extent to which the

explanation method relies on looking into the machine learning model, like its parameters. Portability describes the variety of machine learning models that the explanation method can be used on. Methods with a low translucency have a higher portability. The term "algorithmic complexity" refers to the method's computational difficulty.

Since interpretability demands reasoning instead of an explanation behind the predictions, the model must include an intrinsic explanation method. Therefore, it becomes very translucent, but not portable.

We can also examine individual explanations of interpretable models, instead of the whole model. A well-known property is accuracy, which answers the question of how well an explanation predicts unseen data. Perhaps the most important property of explanations is fidelity, which tells how well the explanation approximates the prediction. Since interpretable models have intrinsic explanation methods, the fidelity of their explanations is perfect. The level of consistency identifies how distinct the explanation is from those of other models trained on the same task and generating similar predictions. The similarity of the explanations for comparable instances is characterized by stability. Perhaps the most significant yet challenging property is comprehensibility, which gauges how well the explanations are understood by humans. The degree of importance indicates how accurately the explanation captures the significance of the explanation's features or components. Furthermore, an explanation should state its certainty and novelty, since these properties can tell us more about the inaccuracy it may have. At last, representativeness measures how many instances an explanation covers, which can be an individual explanation or the entire model.

Although there is a clear concept of the qualities a good explanation method or individual possesses, the absence of a baseline measure for most of these properties is a significant challenge, since it is not clear how to measure them correctly.

Schmidt and Biessmann [28] address this issue by proposing a metric based on the increase in information transfer rate in an annotation task when model justifications

are provided. According to their theory, annotators will be able to replicate a given ML model's decisions more quickly and precisely the better the model's interpretability is. The *information transfer rate* (ITR) is measured in bits per second:

$$\text{ITR} = \frac{I(\hat{\mathcal{Y}}_H, \hat{\mathcal{Y}}_{ML})}{t}, \tag{2.6}$$

where t is the average response time in the annotation task and $I(\hat{\mathcal{Y}}_H, \hat{\mathcal{Y}}_{ML})$ is the mutual information between the prediction from the human labellers $\hat{\mathcal{Y}}_H$, and the model's predictions $\hat{\mathcal{Y}}_{ML}$, this can be computed as

$$I(\hat{\mathcal{Y}}_H, \hat{\mathcal{Y}}_{ML}) = \sum_{\hat{y}_{ML}, \hat{y}_H} p(\hat{y}_{ML}, \hat{y}_H) \log \frac{p(\hat{y}_{ML}, \hat{y}_H)}{p(\hat{y}_{ML}) p(\hat{y}_H)}. \tag{2.7}$$

To obtain the interpretability quality, the ITR increase between the annotation task when no explanations are provided and when annotators are provided the justifications of the model are computed. The actual predictions are not shown. The higher the increase, the better the justifications are interpretable.

However, the demand for a participant-based experiment for this metric is a serious complication. This cannot be done automatically and requires a significant investment in time and money. Therefore, other researchers use proxy functions to reflect a specific component that makes the explanations interpretable.

For example, Silva et al. [29] argue that a good explanation should maximize the three C's: completeness, correctness, and compactness. An explanation is complete if it is capable of being used in other situations so that the audience can confirm the accuracy of that explanation. Correctness is accomplished by being accurate and therefore generating trust. As stated earlier, a human-friendly explanation is compact by a low number of cognitive chunks. To measure the quality of these properties, they used accuracy for correctness, the fraction of the training set covered by the explanation as completeness, and the size in bytes of the explanation after compression using the Deflate algorithm as compactness.

The problem with a proxy function is that it may be completely inaccurate; in particular, the size in bytes does not reflect the number of cognitive chunks. The explanation itself can be very compact, while the use of an extensive data structure might be seen as an excessive explanation.

2.2. Automated Machine Learning

Automated Machine Learning promises major productivity boosts for ML engineers by reducing repetitive tasks in machine learning pipelines. An AutoML pipeline consists of three main steps.

Typical the first task in an AutoML pipeline is data pre-processing. Currently, this task is not handled very well and still needs human interaction with most tools. The pipeline must identify data types and use domain knowledge to extract features from the data. This is challenging since computers do not have the right domain knowledge for selecting the right features. Examples of data encoding techniques are one-hot encoding, target encoding, and count encoding. One-hot encoding is a commonly used technique, where categorical variables are transformed into binary vectors, representing the presence or absence of each category [30]. Another approach is target encoding, also known as mean encoding, which replaces categorical values with aggregated statistics of the target variable for each category. Micci-Barreca [31] provides insights into the application of target encoding and its preprocessing scheme for high-cardinality categorical attributes. Count encoding, on the other hand, represents each category with the count of its occurrences in the dataset [32]. These techniques can be implemented for transforming categorical to numeric features. There also exist techniques to transform the data from numeric to categorical, the other way around, which are mostly used to simplify the data to increase interpretability. In decision trees, for example, this can be a valuable implementation to boost interpretability. Examples are binning or discretization [33], thresholding [34] or rank-based encoding [31].

The next step in an AutoML pipeline is optimization, which involves finding

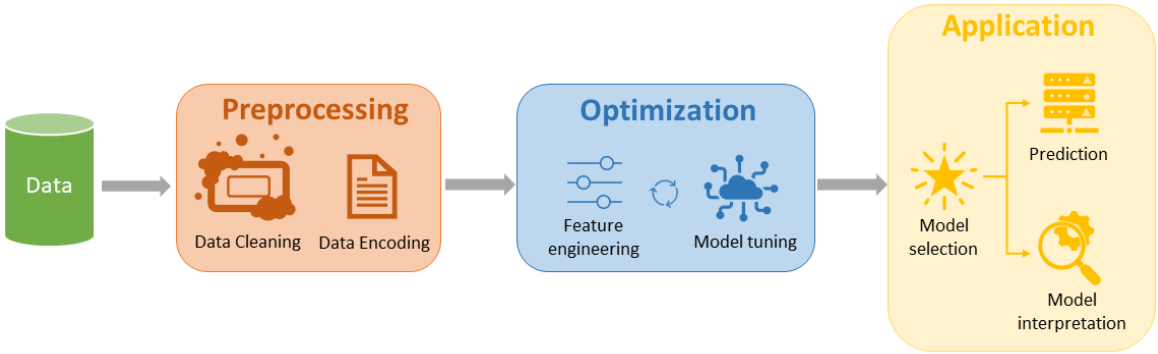


Figure 2.3: Pipeline for AutoML.

the best optimal hyperparameter settings through a search process. Various studies, including those by Bergstra and Bengio [35] and Feurer et al. [36], emphasize the importance of effectively optimizing hyperparameters to improve the performance of machine learning models in AutoML. An AutoML pipeline typically involves selecting from a range of models, each with its own set of hyperparameters. To efficiently search for the best hyperparameter configurations, time-saving techniques such as pruning and search strategies are commonly employed. For instance, one commonly used method is random search, which involves randomly sampling hyperparameter combinations from a predefined search space. Additionally, more advanced search techniques like Bayesian optimization and genetic algorithms can be employed to refine the initial hyperparameter settings. These techniques leverage information from previous iterations to guide the search towards more promising regions of the hyperparameter space. The goal of the optimization step is to identify the best hyperparameter settings that optimize the performance metrics of interest, such as accuracy, precision, recall, or F1 score.

Lastly, the AutoML pipeline generates a report showcasing the results of the best model selected from the optimization phase. This report preferable includes not only the performance metrics of the model but also valuable insights into the important features that contributed to the model's predictions or certain visualizations to facilitate the interpretability and explainability of the model.

There exists a multitude of AutoML tools, both open source and commercial [37]. In table 2.1, a summary of certain properties from seven frequently used open-source

AutoML Tool	Ease of Use	Requires additional explanation	Requires programming skills	Human in the loop	Global explanation	Local explanation	Feature selection and engineering
Auto-sklearn [36]	High	Yes	Some	No	Yes	Yes	Yes
Auto-WEKA [38]	Moderate	No	No	Some	No	No	Yes
Auto-keras [39]	High	Yes	Some	Some	No	Yes	Yes
hyperopt-sklearn [40]	Moderate	Yes	Some	No	No	No	Yes
TPOT [41]	Low	Yes	Yes	No	Yes	No	No
H2O-AutoML [42]	Moderate	Yes	Some	Some	Yes	Yes	Yes
AutoGLUON [43]	High	No	No	Some	No	No	Yes

Table 2.1: Summary of properties of the frequently used AutoML open source tools for tabular data.

AutoML tools is shown. 'Ease of use' is described as high, moderate, or low, depending on how much knowledge is needed and how many steps needs to be followed. The property 'Requires additional explanation' denotes if the model evaluation is interpretable for non-technical persons. 'Requires programming skills' is denoted with none, some, and yes. 'Human in the loop' denotes if the tool allows feedback or editing after finding the best model. The property 'Global explanation' denotes if feature importance is provided and 'Local explanation' denotes if some local explanation method is provided. The remaining properties are answered with Yes/No, which denotes if it is feasible by the tool.

No AutoML tools are fully interpretable, they are only explainable or not at all. Furthermore, the whole process itself is a black box. Therefore, Das et al. propose a white box AutoML solution at scale, Amazon SageMaker Autopilot [44]. Autopilot de-

termines the problem type, analyzes the data, and creates a variety of full ML pipelines, including feature preprocessing and tuned ML algorithms, to produce a scoreboard of potential models. If the performance is unsatisfactory, a data scientist can inspect and adjust the suggested ML pipelines to add their experience and domain knowledge without having to switch to a manual solution.

2.3. Machine Learning for medical classification tasks

As mentioned above, effective machine learning models would be quite helpful for the medical industry. Machine learning can be utilized for risk score improvement, dose determination, and diagnosis. But the trade-off between accuracy and interpretability, both of which are necessary before a high-risk decision can be taken, is a fundamental issue with employing these models in real life.

2.3.1. Interpretability in the medical domain

Loreaux et al. [45] addressed the issue that clinical risk scores generated by machine learning algorithms suffer from a lack of interpretability. They suggest boosting the interpretability of the algorithms to simultaneously predict the risk score and a collection of clinical interventions. These additional predictions can show the clinician that extremely comparable risk scores for two patients with similar characteristics are actually based on different clinical trajectories. In the study, they compared a neural network with two logistic regression models based on features in SOFA [46] and SAPS-II [47], two commonly used risk scores to estimate the probability of mortality for ICU patients. Although the model becomes more interpretable, it still lacks transparency. The understanding of how the specific model learned the intervention prediction is missing.

Teng et al. [48] summarized various applications of explanation methods for disease diagnosis. An overview of these methods is shown in figure 2.4. Since the taxonomy on interpretability is not yet converged to a consensus, Teng et al. use interpretability

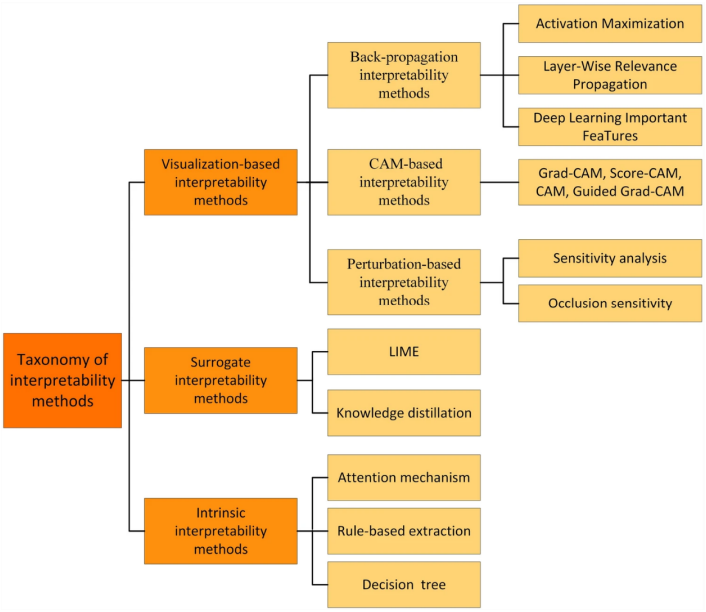


Figure 2.4: Taxonomy of interpretability methods.
Figure adapted from [48].

in contrast to the definitions previously given. According to the given definition, the visualization-based and surrogate methods are explanatory whereas only the inherent interpretability methods are actually interpretable. The survey shows that medical diagnosis with explainability is making convincing progress, but they address a couple of challenges before it can be applied in clinical practice. One of the biggest issues is the lack of a standard concept or evaluation of interpretability and explainability. Since an intuitive evaluation may potentially have an impact on the final diagnosis, the medical domain needs a quantitative analysis of the interpretability or explanation methods. Another challenge is that currently general methods are used in previous work, but the medical domain is sensitive and therefore general methods are not always suitable for this field. In order to make a suitable method, experts should be involved in the model design process, to provide expertise and understanding of the model's decision-making process. Lastly, most researchers adopt a single method to provide predictions and explanations, for example only using medical images and not the patients' records. Adopting multiple modalities potentially enhances accuracy and provides explanations that can be interpreted more holistically.

Tonekaboni et al. [49] studied the specific aspects of explainability that may cat-

alyze building trust in ML models in clinical practice. They interviewed 10 clinicians to identify their needs for building ML systems for their respective clinical practice. They determined four criteria by which an explanation should augment or supplement clinical ML systems to be considered well-designed. The explanation should: (a) recalibrate clinician confidence in model predictions; (b) provide a level of transparency that enables users to validate model outputs with domain knowledge; (c) dependably disseminate model prediction using task-specific representations; and (d) provide concise and actionable steps for clinicians. Furthermore, they curated classes of explanation from qualitative assessments which were identified as most effective to complement the model's predictions.

- The clinicians repeatedly noted how important it is to understand which features contribute to the model's prediction. It allows them to compare the model's decision to their professional judgment. *Feature importance* is, therefore, a key metric to draw the attention of clinicians to specific patient characteristics to help them decide how to proceed.
- Clinicians believe that although patients may have comparable outcomes, they may differ significantly in the clinical course they took to get there, and vice versa patients with similar courses may differ in outcome. Therefore *instance level explanations* by providing similar data instances as explanations are not seen as beneficial for most clinical applications.
- Clinicians view providing a *certainty* score for model performance or predictions as a kind of explanation that completes the output result. A difficulty is that even models that perform well on average can have significant individual mistakes.
- An important aspect reported is that ML models should be able to provide *temporal explanations*, explaining their prediction based on changes in individual patient state.
- Clinicians emphasized the necessity for models that mimic the process of the established methodology of evidence-based medical decision-making. A model with a *transparent design* can facilitate the rationalization of model behaviour.

2.3.2. Bleeding risks

This section focuses on machine learning applications for anticoagulant medicine causing bleeding risks. Because the new dataset contains thrombosis patients and the aim is to predict whether severe bleeding will occur in the near future, the related work will provide further insight into this area.

The study of Lu et al. [50] serves as a perfect illustration of the benefits of machine learning in the medical industry. They studied the performance of several multi-label machine learning models for predicting stroke and bleeding risk of patients in comparison to state-of-the-art risk scores. In the cohort study, data from 9670 patients are used. The patients included were hospitalized with non-valvular atrial fibrillation, received oral anticoagulant medication and had a 1-year follow-up. Outcomes of this follow-up are labelled with ischemic stroke, major bleeding, all-cause death, or event-free survival. A classifier chain was used for three binary classification algorithms; support vector machine, gradient boosting machine, and multi-layer neural networks. As a performance measure, they used the area under the curve. Results indicate that multi-label machine learning models outperform clinical risk scores. A limitation of this study is that they ignored the cause of death, as a result, death caused by ischemic stroke or major bleeding was counted as all-cause death, too. Furthermore, they ensured clinical utility with a threshold of 99% sensitivity for stroke, bleeding, and death based on the area under the curve, and a slightly lower sensitivity was also allowed if there were no matching thresholds. Since interpretability is crucial for healthcare, one can claim that this assurance is insufficient.

Zadeh et al. [51] use deep reinforcement learning to determine warfarin dosing. Warfarin is a commonly used anticoagulant, with a highly variable individual response. Therefore, the prescribed dose should be individualized. Furthermore, a wrong dosage has catastrophic consequences, since an underdose will lead to blood clotting and an overdose to major bleeding events. They used a Pharmacokinetic/ Pharmacodynamic (PK/PD) model to simulate dose responses. Here the PK component models the ab-

sorption, distribution, metabolism, and elimination of the drug, and the PD component shows how the drug affects the patient. The optimization problem is modelled as a Markov Decision Process. To approximate the optimal policy, Q-learning was used, which results in an International Normalized Ratio of the blood coagulability of a patient as output. The authors argue that this setup will result in a more reliable and effective dosing protocol because it incorporates a better representation of reality by adding knowledge of the human body and metabolism to the observations. Results show this approach is more effective than common dosing regimens. However, a big limitation of this work is the lack of interpretability. The deep Q-learning algorithm gives a value for each state-action pair, but it is very hard to find the reason behind these values. So while the PK/PD model has a good interpretable setup and a good reflection of reality, the Q-learning approach fails to be transparent.

2.4. Summary

Previous work shows that there are numerous new promising developments in the field of interpretable machine learning, reducing the performance gap between black box approaches. Furthermore, there are enough interpretable models suited for automated machine learning, described in 2.1.1. But existing AutoML tools use mostly black box models and post-hoc explanation methods to give insights into the predictions. The literature lacks an interpretable automated machine learning pipeline, which would be a practical tool for fields where high-risk decisions are made on a daily basis.

Furthermore, a large body of research has been done on machine learning in the medical field, showing it outperforms human-made decisions and commonly used risk scores. But machine learning algorithms need to be interpretable to establish (dis)trust, a necessity in the medical field.

The goal of this thesis is to fill this gap in the literature with a first step toward interpretable automated machine learning.

3. Proposed method

3.1. The Automated Pipeline

In an automated pipeline, several steps are considered. In Figure 3.1, the proposed pipeline is illustrated. In this section, all steps of this pipeline will be explained.

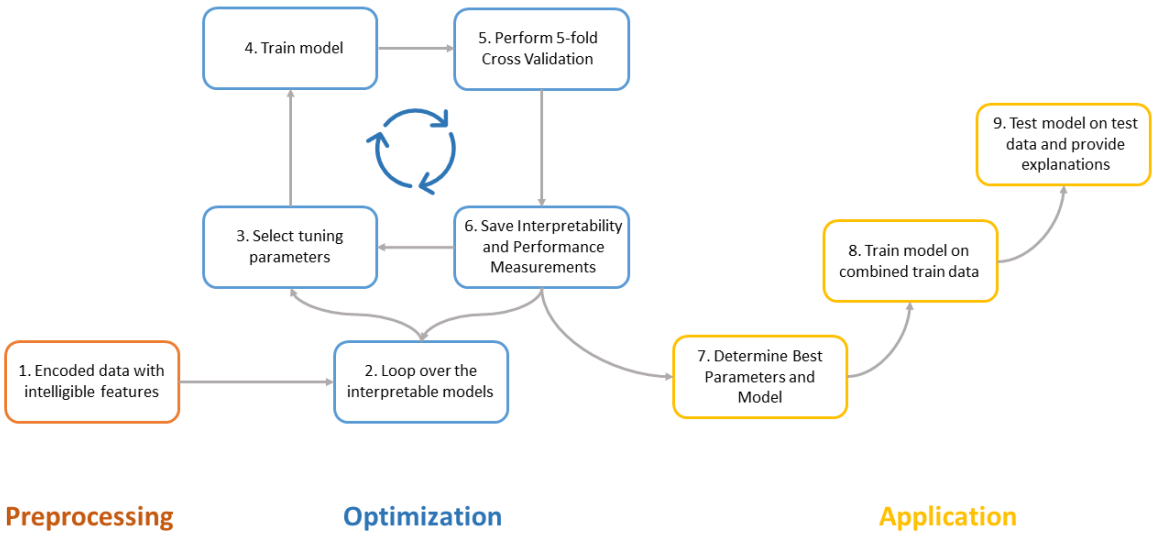


Figure 3.1: Extended AutoML pipeline with interpretability.

3.1.1. Preprocessing

The first stage in the pipeline involves data preprocessing to prepare it for further processing. Prior to preprocessing, the data is divided into a 90% training set and a 10% test set. The training data is subsequently partitioned into 5 folds. For each combination of 4 training folds and 1 validation fold, as well as the entire training set together with the test set, steps 1 to 3 are executed.

Step 1: If categorical features are present in the data, numeric features are extracted to prepare the data for use in machine learning models in this step. The

columns with strings or lists of strings are converted to numeric features using one-hot encoding, count encoding, and target encoding. These encoders are exclusively fitted on the training set. When a column contains a list, it is converted into four separate columns for minimum, maximum, mean, and standard deviation. To ensure that all features are on the same scale, which helps the models perform better, the extracted features are normalized with standardization. This process yields an excessive number of features, leading to redundancy and overlap between those features. Hence, a feature selection step becomes imperative.

Step 2: Secondly, the minimum redundancy maximum relevance (mRMR) [52] technique is employed to rank the features based on their relevance to the target variable and their redundancy with respect to each other. This method allows to select a subset of features that are most informative for the prediction task while minimizing the overlap between them. Once the features are ranked, the top n features can be selected as input parameters in the optimization step. This helps to reduce the computational burden and potentially improve the model's performance by focusing on the most informative features. The selection of the optimal number of features per model will be part of the optimisation step.

Step 3: Unbalanced data sets are commonly encountered in medical datasets, which can lead to biased models favouring the majority class [53, 54]. To address this issue, undersampling is a commonly used approach to balance the classes and improve model performance. By reducing the size of the majority class, undersampling enables the models to capture patterns in the minority class better, leading to improved accuracy and other performance metrics. Previous studies by Kavitha and K. Duraiswamy [55], Khoshgoftaar et al. [56], and Khanteymoori et al. [57] have demonstrated that random undersampling is a viable method for handling imbalanced medical datasets. The last step of the preprocessing stage in the pipeline, undersampling is performed for a class if the ratio of the minority class to this current class is less than 1:9. It involves randomly selecting a subset of samples from the current class. Hence, the ratio becomes 1:3. We note that this approach has been shown to be effective in

reducing the impact of class imbalance and improving model performance on medical datasets.

3.1.2. Optimization

Step 4: The pipeline starts with looping over the four models; DTs, EBMs, NODE-GA(^{2})Ms and DCPs. These models are suited for Interpretable AutoML, since they have no need for human interaction, have an interpretable outcome, and have hyperparameter settings which can be tuned such as regularization methods to boost interpretability. Due to time constraints, this first pipeline has only four different models, but more models can be implemented in further attempts.

Step 5: To optimize the performance of our model, we utilized various parameter settings and feature sets. The parameters were carefully selected to improve interpretability and are documented per model in Table 3.1. Changing the maximum depth, maximum leaf nodes, and minimum samples per leaf of a decision tree can improve interpretability by reducing model complexity and increasing the generalization capability of the model [58, 59]. Furthermore, lowering maximum depth and maximum leaf nodes can prevent overfitting while increasing minimum samples per leaf can lead to simpler and more interpretable splits [59, 60]. To improve the interpretability of Explainable Boosting Machines, interactions can be limited to simplify the model and prevent overfitting and a lower learning rate can reduce model complexity and prevent overfitting [61]. Increasing minimum samples per leaf can also lead to simpler and more interpretable trees by preventing small leaves with high variance [60]. Limiting interactions and increasing the ℓ_2-λ coefficient in NODE-GA(^{2})Ms can improve interpretability. These changes reduce model complexity, prevent overfitting, and improve generalization capability, resulting in a simpler and more interpretable model. To improve the interpretability of Dominance Classifier Predictors, by setting a higher threshold T, the model focuses on the most dominant features, leading to a simpler and more interpretable model. Additionally, using a different voting method will have an impact on the calculation of feature importance and can improve the model's robust-

ness to noise and outliers, which provides a more interpretable ranking of the feature importances. The feature sets included the top 7 ± 2 ranked features obtained from the preprocessing step, due to the psychological studies suggesting that humans can process 7 cognitive chunks efficiently [14]. This leaves a search space of 246 various models with different parameter settings.

Table 3.1: Hyperparameter ranges for the different models.

Maximum Depth	**Maximum leaf nodes**	**Minimum samples per leaf**
None, 10, 3	None, 30, 5	1, 10, 20

(a) Decision Trees

T	**Voting Method**
0.5, 0.6, 0.7, 0.8, 0.9	1, 2, 3, 4

(b) Dominance Classifier Predictors

Interactions	**Learning Rate**	**Early stopping rounds**
0, $\lfloor \frac{\lvert features \rvert}{2} \rfloor$, $\lvert features \rvert$	0.05, 0.01, 0.005	5, 15, 25

(c) Explainable Boosting Machines

Interactions	ℓ_2-λ
False, True	0, 0.00001, 0.0001, 0.001

(d) Node-GAMs

Step 6 & 7: For each of these models, we utilized 5-fold cross-validation with a weighted performance measure based on F1-score, Compactness, and Stability. The F1-score is a commonly used performance measure for medical data, as it accounts for sensitivity. Compactness and Stability are measures of interpretability, and their calculation is explained in detail in Section 3.2. Both measures are based on local explanations. However, Compactness can also be applied to a global explanation. If local explanations are not supported by the model, the pipeline will calculate Compactness on the global explanation, return `N/A` for Stability, and report a weighted score without Stability. Therefore, these models are hard to compare with the others.

3.1.3. Application

Step 8, 9 & 10 After the optimization step, we select the top 3 parameter settings for each model and train them using the entire training dataset. We evaluate their performance on the test set using the same weighted performance measures (F1-score, Compactness, and Stability) as in the optimization step. The results are stored in a CSV file and the pipeline returns the best-performing model. A PNG file containing the global explanation of the best model is also saved. In addition, local explanations can be extracted from the returned model for further analysis.

3.2. Proposed Interpretability Measures

Model-agnostic measures for interpretability are missing from the literature, which is a necessary step for the automated pipeline because it would otherwise be impossible to compare different models with one another. Hence, in order to answer SQ2, we introduce two interpretability measures that are model-agnostic: Compactness and Stability. These selected concepts for interpretability, derived from the literature, encompass important characteristics. Compactness emerges as a significant factor strongly associated with interpretability. It indicates the degree of conciseness and clarity in the explanations provided. Stability, both at the macro and micro levels, plays a crucial role in establishing the reliability and trustworthiness of the models.

3.2.1. Compactness

An explanation is considered compact if it conveys the relevant information using a small number of features, which correspond to cognitive chunks. To formally define compactness, we use the following equation:

$$\textit{Explanation compactness} = 1 - \frac{|F_{ex}| - 1}{|F|}, \tag{3.1}$$

where F is the set of features used in the model, and F_{ex} is the subset of features used in the explanation. The expression $(|F_{ex}| - 1)/|F|$ represents the proportion of features used in the explanation relative to the total number of features. Since explanations should use at least one feature, we subtract 1 from $|F_{ex}|$ to avoid penalizing single-feature explanations. Finally, we subtract this proportion from 1 to obtain a compactness score between 0 and 1, where higher values indicate more compact explanations.

To evaluate the compactness of a model, we apply the same formula to each individual explanation and compute their average:

$$\textit{Model compactness} = \frac{1}{|E|} \cdot \sum_{ex \in E} \left(1 - \frac{|F_{ex}| - 1}{|F|}\right), \tag{3.2}$$

where E is the set of explanations generated by the model, and $|E|$ is the total number of explanations. In our experiments, we compute the compactness of models trained on decision rules, decision trees, and linear models, as described below.

For decision rules, we use the number of splits in each rule as the number of features used in the explanation. Specifically, we consider each rule in the validation set as a separate explanation and compute its compactness score using the explanation compactness equation.

For decision trees, we can convert them into rule lists and apply the same method as for decision rules. Alternatively, we can compute the compactness of a decision tree by aggregating the compactness scores of its rules. Specifically, we consider each leaf node as a separate explanation and assign to it the set of features used in the path from the root to that node which results in the explanation's compactness. We can then apply the models' compactness equation.

For linear models, we need to determine the subset of features used in each explanation, since including all non-zero features could result in an overwhelming number

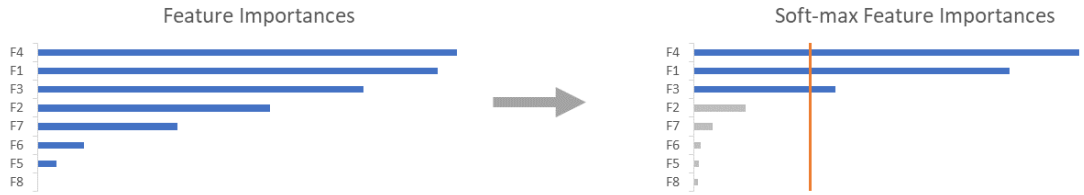


Figure 3.2: Illustration of the feature set of the explanation for linear models.

of features, where some might not even be relevant. To this end, we use a two-step process as illustrated in 3.2. First, we calculate the importance of each feature used in an explanation, which is measured by the absolute value of its t-statistic:

$$I_{ex} = \left\{ \left| \frac{\beta_j}{SE(\beta_j)} \right| \;\middle|\; j \in F \right\}, \tag{3.3}$$

where β_j is the coefficient of feature j, and $SE(\beta_j)$ is its standard error. Second, we apply a softmax function to these feature importances to obtain a probability distribution over the features:

$$P(I_{ex}) = \left\{ \frac{e^i}{\sum_{i \in I_{ex}} e^i} \;\middle|\; i \in I_{ex} \right\}. \tag{3.4}$$

Finally, we use a threshold to select the most important features for the explanation. If all features are used equally, the softmax values of all features will be $1/|F|$. A feature is included in the explanation if its softmax value is equal to or higher than this number:

$$F_{ex} = \left\{ f \;\middle|\; P(I_{ex})_f \geq \frac{1}{|F|} \right\}. \tag{3.5}$$

We can then apply the model compactness equation (3.2) to this feature set to compute the explanation's Compactness score.

3.2.2. Stability

An explanation method is stable if for similar instances similar explanations are provided. The proposed method is based on Turney [62] and Zatar et al. [63].

Zatar et al. proposed a clustering-based approach to improve the stability of LIME. Since LIME's explanations can be unstable due to the high sensitivity of the surrogate models to small changes in the input data, it is important to stabilize the explanations. Zatar et al. addressed this issue by first clustering the data points and then using LIME on similar instances within each cluster to obtain more stable explanations. Although Zatar et al.'s method does not have a direct stability metric, it is based on the assumption that clustering similar instances together will result in more consistent and interpretable explanations. This assumption will be used in the proposed method.

According to Turney, the stability of a classification algorithm is the degree to which it generates repeatable results, given different batches of data from the same process. He evaluated stability by looking at the degree of agreement between predictions. He suggested using $m \times 2$-fold cross-validation. Two classifiers are built on the two folds at each of the m steps and tested using artificially produced instances drawn from a population distribution. The agreement measure is the percentage of cases where the predictions of the two classifiers match. The final assessment of the learning process' stability is based on the average agreement over the m runs. Agreement is a semantic metric that has the benefit of being model-agnostic. Since our definition of stability is slightly different and is used for explanations, the approach is modified according to our definition. We need to measure the agreement of the instances and the explanations of one classification algorithm.

To measure stability, we need to assess the agreement of the instances and the explanations of a single classification algorithm. The proposed method consists of the following steps:

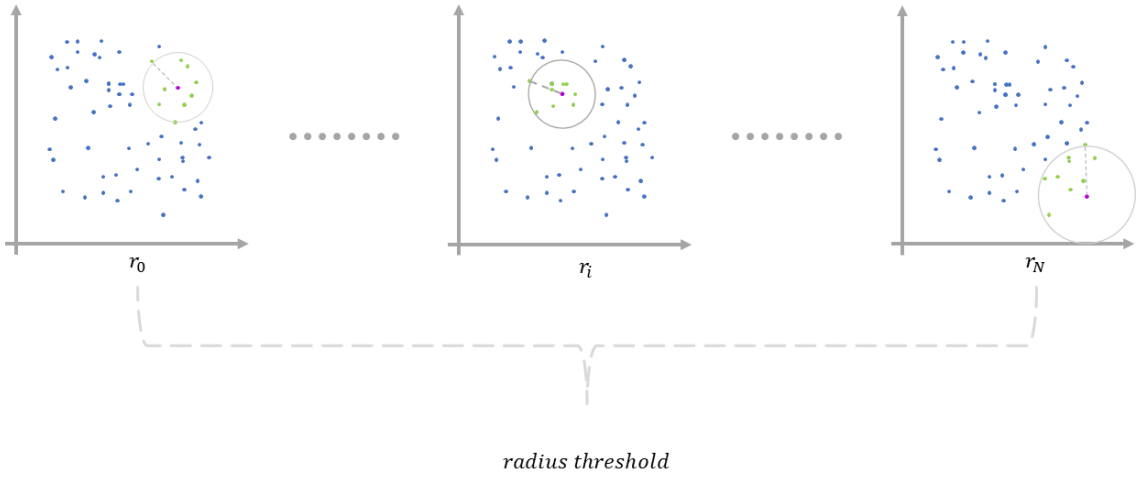


Figure 3.3: Illustration of the radius threshold calculation. For each instance i (purple), we find the nearest 9 neighbors (green) and get the maximum radius r_i, the average gives the radius threshold.

1. **Finding similar instances per instance:** We use the k-nearest neighbour algorithm to find the neighbours of each instance in the training data set. At this moment, the k is set to 9. However, the k can be changed accordingly to the size of the dataset and the strictness for stability. Next, we find the average of the radius r_i used to determine k-neighbors of each instance i. This becomes our radius threshold for the test set:

$$T_r = \frac{\sum_{i=0}^{N} t_i}{N}, \tag{3.6}$$

   where N is the size of the training set. Figure 3.3 illustrates this step.
2. **Creating the instance space:** We find similar instances in the test data using the radius threshold (T_r) to determine the neighbours for each instance. This results in the instance space.
3. **Creating the explanation space:** For decision trees and decision rules, each leaf or rule represents the common explanation of the instances that fall into this leaf node or meet all the conditions of the corresponding rule. For models with feature importances, we use the same method as the instance space, but with the feature importances instead of the feature values. The explanation space consists

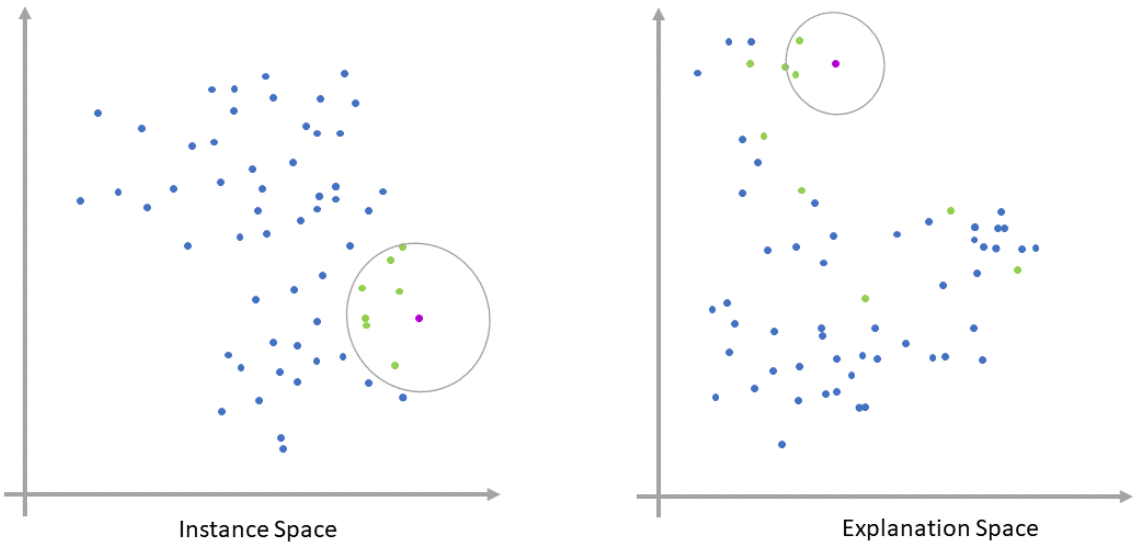


Figure 3.4: Illustration of the instance space and explanation space. The example instance is purple, and neighbours from the instance space are green.

of the neighbours of each instance that share the same decision rule or leaf (for decision trees and decision rules) or similar feature importantances (for models with feature importances).

4. **Calculating the Stability measure:** From the instance space (Is) and the explanation space (Es), we can calculate the agreement of neighbours, which results in the Stability measure:

$$stability = \frac{|Is \cap Es|}{|Is|}. \tag{3.7}$$

Figure 3.4 illustrates the process with the instance space and the explanation space.

4. Experimental results

4.1. Data

In this thesis, three datasets will be used with different classification tasks and information. We focused on tabular datasets in the medical field. For reproducibility purposes, two open-source datasets are used. Both datasets are used in related work, which offers benchmarks for performance.

4.1.1. Wisconsin

Wisconsin Diagnostic Breast Cancer (WDBC) is a dataset from the University of California at Irvine machine learning repository [64]. It consists of 569 samples with 30 features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass [65]. For each image, the mean, standard error and "worst" or largest of the following ten characteristics of the cell nuclei present in the image are computed:

(i) Radius (mean of distances from the centre to points on the perimeter)
(ii) Texture (standard deviation of grey-scale values)
(iii) Perimeter
(iv) Area
(v) Smoothness (local variation in radius lengths)
(vi) Compactness (perimeter^2 / area - 1.0)
(vii) Concavity (severity of concave portions of the contour)
(viii) Concave points (number of concave portions of the contour)
(ix) Symmetry
(x) Fractal dimension ("coastline approximation" - 1)

The target column is the diagnosis, which is M for malignant or B for benign. The class distribution is slightly biased since there are 357 B and 212 M samples.

4.1.2. MIMIC-IV

Medical Information Mart for Intensive Care (MIMIC)-IV [66] is a freely available medical dataset for research providing critical care data for 43,005 patients admitted to intensive care units at the Beth Israel Deaconess Medical Center (BIDMC). In this thesis, the module Intensive Care Unit (*icu*) will be used. The *icu* module contains data sourced from the clinical information system at the BIDMC: MetaVision (iMDSoft). MetaVision tables were denormalized to create a star schema where the *icustays* and *d_items* tables link to a set of data tables all suffixed with "events". Data documented include:

- Intravenous and fluid inputs
- Patient outputs
- Procedures
- Information documented as a date or time
- Charted information occurring during the ICU stay. Contains the majority of information documented in the ICU.

All events tables contain a *stay_id* column allowing identification of the associated ICU patient in *icustays*, and an *itemid* column allowing identification of the concept documented in *d_items*. Each patient is labelled with yes/no in-ICU mortality.

The initial step involves the selection of the cohort, specifically choosing patients who are aged 15 or older and have had an in-ICU stay lasting between 12 hours and 10 days. Furthermore, only their first stays are considered for analysis. This cohort selection approach, as described by Wang et al. [67], has been widely adopted by other researchers in the field. Subsequently, the dataset is constructed by calculating the mean values of all intravenous and fluid inputs, patient outputs, and charted events.

4.1.3. Trombose dataset

The new dataset used for this research is the Atalmedial Thrombosis dataset, which consists of anonymized patient records of individuals with thrombosis. It includes information on blood values, medical events, and medication history from the past 60 days. Each observation in the dataset is labelled as either "yes" or "no" to indicate whether the patient experienced severe bleeding. The dataset includes several features per patient, including:

- Bleedings: Information related to occurrences of bleeding events.
- Kind of meeting: Describes the type or nature of the medical appointments.
- Indication: Specifies the reason or purpose behind a particular medical treatment.
- Contraindication: Indicates any factors or conditions that may prevent using specific medications or treatments.
- MedicationCode: Indicates which medication is administered to the patient.

The dataset exhibits a significant class imbalance, with 47 instances with severe bleeding and 5544 instances without.

To prepare the data for analysis, the features undergo preprocessing steps described in Section 3.1 about the automated pipeline, specifically steps 1 to 3. Since the features represent data from the last 60 days, they are presented to the machine learning model as lists.

4.2. Results

All experiments are done on a laptop with an Intel Core i7 processor and 32GB of RAM. It took 6 hours to run the pipeline on WDBC, 108 hours on MIMIC and 5 hours on Atalmedial Thrombosis dataset.

Firstly, we observe that for each dataset the three DTs exhibit identical perfor-

mance across all measures, which is due to the fact that they all train the same trees. Because the hyperparameters consist of minimum and maximum constraints, the actual optimal settings obtained are the exact same. Secondly, we will not compare NODE-GAM with the other models on interpretability and weighted score, since it has `N/A` values for Stability, and Compactness is calculated on the global explanation.

By evaluating the task performance of the different models on each dataset, we conclude the F1-scores achieved by the pipeline were found to be high, indicating their strong performance in capturing both precision and recall. The results presented in Table 4.1a reveal that the top three EBM models and DCPs models attained the highest F1-score of 0.982 on the WDBC dataset. Similarly, in Table 4.1b, the EBM model with no interactions between features, a learning rate of 0.005, early stopping round set to 5, and was presented with the top 7 features, achieved the highest F1-score of 0.951 on the MIMIC dataset. Turning to Table 4.1c, we observe that on the Atalmedial Thrombosis dataset, all EBMs, all DCPs models and the NODE-GAM with no interactions, a ℓ_2-λ coefficient of 0.05 and trained on the top 7 features, obtained the highest F1-score of 0.988.

Furthermore, when considering the interpretability scores, we note that the DTs obtained the highest Compactness score of 0.844 on the WDBC dataset, while the DCPs exhibited perfect Stability scores of 1.0. Similarly, on the MIMIC dataset, the DTs demonstrated the best Compactness scores of 0.844, and all models achieved perfect Stability scores. Examining the Atalmedial Thrombosis dataset, we found that the DTs achieved a Compactness score of 0.880, while the DCPs attained the highest Stability score of 0.957.

When evaluating both task performance and interpretability scores, we can determine the best overall models. The weighted scores are calculated using a balanced ratio of 1:1:1. In the case of the WDBC dataset, the DCP model with a threshold of 0.9, utilizing voting method 4, and trained on the top 7 features, achieved the highest weighted score of 0.930. Shifting our focus to the MIMIC dataset, all DTs models

received the highest weighted score of 0.924. Lastly, for the Atalmedial Thrombosis dataset, all DCP models attained the highest weighted score of 0.830, showcasing their performance and interpretability.

The pipeline provides together with the scores, an image of the global explanation. Examples of these explanations are shown in Appendix A.

Table 4.1: Top 3 per model with hyperparameter settings per dataset. Reporting 5-fold Cross Validation (5-CV), F1-score, Compactness (C), Stability (S) and Weighted Score (WS) and parameter settings ia: interactions, lr: learning-rate, esr: early stopping rounds, n: number of features, md: maximum depth, mln: minimum leaf nodes, msl: minimum samples per leaf, T: ratio-threshold, vm: voting method, *N* indicates None.
(a) Results on WDBC dataset
* NODE-GA^{2}M achieves the highest weighted score, however, since there is no Stability score available, it is not considered in the comparison and therefore not the definitive best model.

Model	**Parameters**	**WS** 5-CV	**F1** Test	**C** Test	**S** Test	**WS** Test
EBM	ia: 0, lr: 0.05, esr: 25, n: 5	0.736	0.982	0.511	0.890	0.795
EBM	ia: 0, lr: 0.01, esr: 5, n: 5	0.736	0.982	0.533	0.884	0.800
EBM	ia: 0, lr: 0.01, esr: 25, n: 5	0.736	0.982	0.511	0.890	0.795
DT	md: *N*, mln: 5, msl: 20, n: 9	0.921	0.965	0.844	0.909	0.844
DT	md: 10, mln: 5, msl: 20, n: 9	0.921	0.965	0.844	0.909	0.844
DT	md: 3, mln: *N*, msl: 20, n: 9	0.921	0.965	0.844	0.909	0.844
DCP	**T: 0.9, vm: 4, n: 7**	**0.914**	**0.982**	**0.833**	**1.0**	**0.939**
DCP	T: 0.8, vm: 2, n: 9	0.910	0.982	0.808	1.0	0.930
DCP	T: 0.8, vm: 3, n: 9	0.910	0.982	0.808	1.0	0.930
NG	ia: 0, ℓ_2-λ: 0.05, n: 7	0.972	0.956	1.0	N/A	**0.978***
NG	ia: 0, ℓ_2-λ: 0, n: 7	0.968	0.947	1.0	N/A	0.974
NG	ia: 0, ℓ_2-λ: 0.1, n: 5	0.967	0.956	1.0	N/A	**0.978***

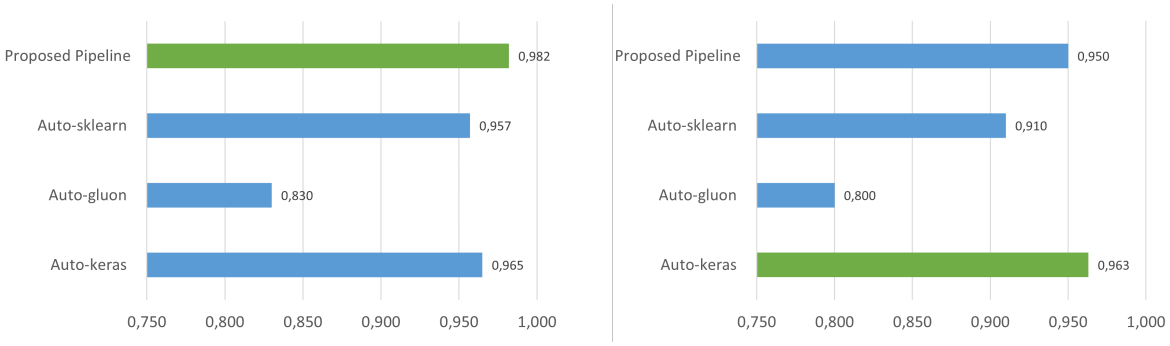
(b) Results on MIMIC dataset

Model	**Parameters**	**WS** 5-CV	**F1** Test	**C** Test	**S** Test	**WS** Test
EBM	ia: 0, lr: 0.005, esr: 5, n: 7	0.823	0.951	0.476	1.0	0.809
EBM	ia: 0, lr: 0.01, esr: 15, n: 7	0.808	0.950	0.541	1.0	0.830
EBM	ia: 0, lr: 0.01, esr: 25, n: 7	0.808	0.949	0.519	1.0	0.823
DT	**md: *N*, mln: 5, msl: 1, n: 9**	**0.962**	**0.923**	**0.844**	**1.0**	**0.924**
DT	**md: *N*, mln: 5, msl: 10, n: 9**	**0.962**	**0.923**	**0.844**	**1.0**	**0.924**
DT	**md: *N*, mln: 5, msl: 20, n: 9**	**0.962**	**0.923**	**0.844**	**1.0**	**0.924**
DCP	T: 0.5, vm: 2, n: 9	0.915	0.941	0.803	1.0	0.915
DCP	T: 0.6, vm: 2, n: 9	0.915	0.941	0.803	1.0	0.915
DCP	T: 0.6, vm: 3, n: 9	0.913	0.932	0.808	1.0	0.913
NG	ia: 1, ℓ_2-λ: 0.1, n: 5	0.904	0.942	0.867	`N/A`	0.904
NG	ia: 0, ℓ_2-λ: 0.05, n: 5	0.873	0.944	0.800	`N/A`	0.872
NG	ia: 0, ℓ_2-λ: 0.1, n: 5	0.872	0.944	0.800	`N/A`	0.872

(c) Results on Atalmedial Thrombosis dataset

Model	**Parameters**	**WS** 5-CV	**F1** Test	**C** Test	**S** Test	**WS** Test
EBM	ia: 9, lr: 0.005, esr: 5, n: 9	0.670	0.988	0.808	0.486	0.761
EBM	ia: 9, lr: 0.01, esr: 15, n: 9	0.669	0.988	0.808	0.485	0.760
EBM	ia: 0, lr: 0.01, esr: 5 , n: 9	0.669	0.988	0.808	0.485	0.760
DT	md: *N*, mln: *N*, msl: 20, n: 5	0.835	0.937	0.880	0.426	0.747
DT	md: *N*, mln: 30, msl: 20, n: 5	0.835	0.937	0.880	0.426	0.747
DT	md: *N*, mln: 5, msl: 20, n: 5	0.835	0.937	0.880	0.426	0.747
DCP	**T: 0.5, vm: 4, n: 5**	**0.860**	**0.988**	**0.545**	**0.957**	**0.830**
DCP	**T: 0.6, vm: 4, n: 5**	**0.860**	**0.988**	**0.545**	**0.957**	**0.830**
DCP	**T: 0.7, vm: 4, n: 5**	**0.860**	**0.988**	**0.545**	**0.957**	**0.830**
NG	ia: 0, ℓ_2-λ: 0, n: 7	0.921	0.981	0.571	`N/A`	0.776
NG	ia: 0, ℓ_2-λ: 0.001, n: 7	0.921	0.981	0.571	`N/A`	0.776
NG	ia: 0, ℓ_2-λ: 0.05, n: 7	0.921	0.988	0.571	`N/A`	0.780

Figure 4.1: F1-scores of existing AutoML tools on WDBC and MIMIC datasets in com- parison to the proposed pipeline.



(a) WDBC dataset

(b) MIMIC dataset

4.3. Comparison with literature

To evaluate the performance of the proposed pipeline, we compared its F1-score on two datasets, namely MIMIC and WDBC, against three well-established open-source AutoML tools: Auto-sklearn, Auto-keras, and AutoGLUON. These tools were selected due to their high ease of use, making them suitable for practical applications (Table 2.1). All experiments were conducted on a Google Colab Jupyter notebook using default settings, except for the Auto-keras on MIMIC dataset, where modifications were made to address memory and time constraints. Specifically, we set the "max_trials" parameter, representing the maximum number of different Keras models to try, to 10, and reduced the number of epochs to 50 from the default value of 1000.

The F1-scores achieved by the existing AutoML tools and the proposed pipeline are presented in Figure 4.1. Among the existing AutoML tools, Auto-keras exhibited the highest performance on both datasets, achieving an F1-score of 0.965 on the WDBC dataset and 0.963 on the MIMIC dataset. The proposed pipeline achieved comparable F1-scores to the existing AutoML tools, demonstrating no significant loss in task performance.

It is worth mentioning that the selection of the AutoML tools was based on their ease of use and practicality, allowing for straightforward integration into real-world

scenarios. Furthermore, the proposed pipeline outperformed the existing tools on the WDBC dataset, attaining an F1-score of 0.982. These results highlight the efficacy of the proposed pipeline in achieving competitive performance.

4.4. Discussion

The findings in this section unequivocally confirm the feasibility and effectiveness of developing an interpretable automated machine learning pipeline for medical classification tasks that achieves a comparable F1-performance to existing non-interpretable tools.

According to the overall performance of the various models, EBMs and DCPs perform with the highest level of accuracy on different tasks, DTs with the highest level of Compactness, and DCPs with the highest level of Stability. The fact that DTs' maximum depths can be set to a low value, ensuring that only a few features are used in an explanation accounts for their high Compactness scores. DCPs' superior Stability ratings are a result of their design. The same prediction and inference are made for similar instances. Consequently, the explanation is comparable for similar instances too.

Based on these observations, we can conclude that there exists a trade-off between model interpretability and performance. Despite the fact that EBMs have performed admirably, their interpretability differs substantially depending on the dataset. The DT models, on the other hand, slightly sacrifice performance but provide better interpretability because of their logical decision-making processes.

The trade-off between interpretability and performance should be carefully evaluated in real-world tasks, considering the impact of decision-making transparency and the domain-specific requirements. By adjusting the weighted score ratio in the pipeline, it becomes possible to implement the desired trade-off and select the best model for the given task.

Lastly, it is intriguing to observe that EBM and decision trees (DTs) exhibit high Compactness scores but lower Stability scores, whereas DCP shows the opposite pattern. This raises the question of whether there exists a trade-off between these measures. In the case of decision trees, increasing the number of minimum samples per leaf can enhance Stability, yet it does not directly influence Compactness. Conversely, adjusting the maximum depth can ensure Compactness without significantly impacting Stability. For EBMs, early stopping rounds are employed to prevent overfitting, leading to improved Stability, while the number of interactions influences Compactness. For DCPs, the voting method employed influences the feature importances, which in turn impacts the Compactness measure. Additionally, the internal design of DCPs contributes to their remarkably high Stability.

This suggests that achieving high levels of Compactness and Stability simultaneously may pose a challenge, as certain factors that enhance one measure do not necessarily affect the other. It emphasizes the need for careful consideration and striking a balance between these two aspects when interpreting and evaluating the models. Further exploration and experimentation can lead to a better understanding of the relationship and potential trade-offs between Compactness and Stability. Another promising direction is to explore the possibility of combining the strengths of DCP, known for achieving high Stability, with the Compactness of DT. By integrating the two methods, it may be possible to develop a novel machine learning approach that can achieve exceptional levels of Compactness and Stability simultaneously.

5. Human evaluation

To assess the extent to which the proposed measures align with the subjective interpretability concept as perceived by humans, a user study was conducted.

5.1. Methodology

5.1.1. Environment

The user study was conducted using Qualtrics, a survey platform provided by the University of Utrecht. Participants independently answered the study questions on a computer. Prior to responding, participants were provided with a detailed explanation of the study's purpose, their task, and the potential usage of their answers for research purposes. Ethical considerations were taken into account, and informed consent was obtained from all participants.

The participants in the user study were employees of Atalmedial, including dosing advisors, thrombosis doctors, and lab chemists. This diverse group of participants ensured a range of expertise and perspectives in evaluating the interpretability of the models. The user study involved a total of 9 participants.

5.1.2. Questions

In the user study, participants were presented with four patient cases, each with their specific features. Explanations for these cases were generated by the best EBM, DT, and DCP models. Participants had the opportunity to review the explanations provided by one model at a time for all four patient cases.

After reviewing the explanations, participants were asked to complete the System Causability Scale (SCS). The SCS is a scale used to assess the interpretability or

causability of a system, as defined by Holzinger et al. [68]. The scale consists of 10 statements related to interpretability aspects, which participants rate using a Likert scale (e.g., from 1 to 5). The SCS score is then computed by summing the scores and dividing by 50, indicating their level of agreement or perception. The SCS consists of the following statements:

1. I found that the data included all relevant known causal factors with sufficient precision and granularity.
2. I understood the explanations within the context of my work.
3. I could change the level of detail on demand.
4. I did not need support to understand the explanations.
5. I found the explanations helped me to understand causality.
6. I was able to use the explanations with my knowledge base.
7. I did not find inconsistencies between explanations.
8. I think that most people would learn to understand the explanations very quickly.
9. I did not need more references in the explanations: e.g., medical guidelines, regulations.
10. I received the explanations in a timely and efficient manner.

By completing the SCS for each model, participants were able to express their perceptions of interpretability regarding the explanations provided by each model. This approach allowed for a comparative evaluation of the models' interpretability based on participant feedback.

5.1.3. Hypotheses

The following null and alternative hypotheses are proposed to investigate potential differences in user perceptions of interpretability among the models based on the System Causability Scale (SCS) scores:

Null Hypothesis (H0): {Model 1} has not a higher perceptions of interpretabil-

ity then {Model 2}.

Alternative Hypothesis (Ha): {Model 1} has a higher perceptions of interpretability then {Model 2}.

For the purposes of this study, three different null hypotheses will be tested, representing each pair of models: comparing EBM and DT, DT and DCP, and DCP and EBM.

Furthermore, the following null and alternative hypotheses are proposed for investigating differences in particular questions in the System Causability Scale:

Null Hypothesis (H0): The median of the differences between the paired data on {Question n} is zero.

Alternative Hypothesis (Ha): The median of the differences between the paired data on {Question n} is {*positive/negative*} ($\{Model\ 1\} \lessgtr \{Model\ 2\} + \mu_0$).

5.1.4. Statistical test

To test the overall hypotheses, a paired two-tailed t-test will be conducted to compare the SCS scores obtained by the models. The paired t-test will assess whether any significant differences exist between two of the three models regarding their perceived interpretability. The significance level (α) will be set at 0.05.

The hypothesis testing will provide insights into whether the proposed measures capture distinct levels of interpretability as perceived by users, enabling a comparative assessment of the three models' effectiveness in conveying understandable explanations.

To test the question hypotheses, a Wilcoxon signed-rank test [69] will be conducted to compare the Likert scales per paired question on the models. The Wilcoxon

signed-rank test will assess whether any significant differences exist between two models regarding particular questions. The significance level (α) will be set at 0.05.

5.2. Results

5.2.1. Overall SCS Significance Analysis

The individual scores of the SCS for each model are presented in Table 5.1. The results of a paired-t test indicated a significant difference between the DT model ($M = 0.607$, $SD = 0.065$) and the DCP model ($M = 0.431$, $SD = 0.124$), $t(8) = 3.3$, $p = .010$. Similarly, a significant difference was observed between the EBM model ($M = 0.662$, $SD = 0.076$) and the DCP model, $t(8) = 4.7$, $p = .002$. Lastly, a significant difference was found between the DT and EBM models, $t(8) = 2.6$, $p = .030$.

Table 5.1: The System Causability Scale per model per participant. AVG = average and STD = standard deviation.

Participant	**DT**	**EBM**	**DCP**
#1	0.58	0.56	0.54
#2	0.68	0.66	0.28
#3	0.66	0.66	0.28
#4	0.58	0.66	0.28
#5	0.60	0.78	0.46
#6	0.66	0.72	0.42
#7	0.54	0.60	0.50
#8	0.48	0.56	0.46
#9	0.68	0.76	0.66
AVG	0.607	0.662	0.431
STD	0.065	0.076	0.124

Table 5.2: Results of the Wilcoxon signed-rank test on paired question per two models. Question numbers are the same as used in section 5.1.2. H_a refers to whether the alternative hypothesis was less or greater than μ_0 ($\{Model\ 1\} \lesseqgtr \{Model\ 2\} + \mu_0$). Reported are the W-value and p-value obtained from the tests. `N/A` indicates there were too many similar answers that prevent the calculation of the Wilcoxon test, * indicates $p \leq \alpha$.

Q	H_a	W	p
#1	$<$	0	.010*
#2	$<$	1.5	.020*
#3	N/A	N/A	N/A
#4	$<$	14	.164
#5	$<$	0	.007*
#6	$<$	2.5	.017*
#7	$>$	5.5	.165
#8	$<$	0	.017*
#9	$<$	3	.117
#10	N/A	N/A	N/A

(a) Comparison between DCP and DT

Q	H_a	W	p
#1	$<$	0	.007*
#2	$<$	0	.017*
#3	$<$	5	.286
#4	$<$	7	.062
#5	$<$	2	.014*
#6	$<$	0	.017*
#7	N/A	N/A	N/A
#8	$<$	0	.027*
#9	N/A	N/A	N/A
#10	$<$	0	.017*

(b) Comparison between DCP and DT

Q	H_a	W	p
#1	N/A	N/A	N/A
#2	N/A	N/A	N/A
#3	N/A	N/A	N/A
#4	N/A	N/A	N/A
#5	N/A	N/A	N/A
#6	N/A	N/A	N/A
#7	$>$	0	.013*
#8	$<$	7	.260
#9	$<$	9	.412
#10	$>$	0	.010*

(c) Comparison between EBM and DT

5.2.2. Individual Question Significance Analysis

Table 5.2 presents the results of the Wilcoxon signed-rank test on the question pairs for the model comparisons. In the comparison between DT and DCP shown in Table 5.2a, DT achieves significantly higher scores on questions 1, 2, 5, 6, and 8. These questions primarily focus on understandability and causality, indicating that the DT model provides explanations that are easier to comprehend and better capture causal relationships compared to the DCP model. Additionally, since questions 2 and 6 relate to expert knowledge, it suggests that the DT model excels in providing explanations that domain experts can readily understand.

The comparison between EBM and DCP in Table 5.2b reveals significantly higher scores for EBM than DCP on questions 1, 2, 5, 6, 8, and 10. Similar to the DT - DCP comparison, the EBM model demonstrates superior performance in terms of understandability, capturing causality, providing explanations understandable for domain experts, and delivering explanations in a more efficient manner compared to the DCP model.

Lastly, in the DT - EBM comparison shown in Table 5.2c, the EBM model attains significantly higher scores on questions 7 and 10. This indicates that experts found more inconsistencies in the DT model than in the EBM model, and the EBM model delivers explanations more efficiently.

5.3. Conclusion

The findings from the overall interpretability analyses suggest that the DCP model's interpretability, as perceived by the study participants, significantly differs from that of the DT and EBM models. It appears that the DCP model faces challenges or limitations in providing understandable explanations compared to the other models. However, it is important to note that in the experimental results presented in Chapter 4, the DCP model is ranked as the best model based on the weighted scores. This apparent contradiction can be attributed to the high Stability score achieved by the DCP model, which is an inherent characteristic of its design. By assigning appropriate weights to the different measures, this discrepancy can be effectively addressed and accounted for in the overall evaluation.

Further item-wise examination of the SCS questionnaire results provides a deeper understanding of the DCP model's interpretability issues. The results reveal that the DCP model struggles in capturing causality and is comparatively harder to comprehend compared to the EBM and DT models.

Interestingly, the significantly higher score of the EBM model compared to the DT

model in question 7 highlights the presence of more inconsistencies in the explanations provided by the DT model. Inconsistency is typically associated with lower stability, yet this substantial difference in inconsistency is not adequately captured by the stability measures discussed in Chapter 4, where the DT model receives only a slightly lower Stability score. This suggests that there may be other factors or aspects contributing to the observed inconsistencies in the DT model's explanations, or that the Stability measure does not reflect the subjective perception of stability. This requires further investigation and consideration beyond the scope of the Stability measures alone.

Overall, these findings highlight the discrepancies in interpretability among the models, with the DCP model exhibiting limitations in terms of capturing causality and understandability, while the DT model demonstrates inconsistencies in its explanations. These insights emphasize the importance of evaluating interpretability from multiple angles to gain a comprehensive understanding of model performance.

6. Conclusion

In this study, our primary objective was to explore the feasibility of developing an interpretable automated machine learning pipeline for medical classification tasks while maintaining F1-performance comparable to existing non-interpretable tools.

To achieve this goal, we divided our investigation into subquestions, each addressing a specific aspect of the research. Let's further elaborate on these subquestions and their corresponding findings:

(**SQ1**) **Which existing interpretable models are suitable for an automated pipeline?**

Through an extensive literature review, we identified several interpretable models that hold promise for our automated pipeline. Considering the diverse range of model families, we focused our research on a subset that included all different families: Decision Trees, Explainable Boosting Machine, Dominance Classifier Predictors, and NODE-GA(^{2})Ms. These models were selected for their advantage of finding an robust model without human intervention and flexible hyperparameter settings with regularizations to boost interpretability. However, we discovered that NODE-GA(^{2})Ms were less suitable due to their limitations in producing local explanations.

(**SQ2**) **Which interpretability measures can be used for automatical model selection?**

Our investigation revealed a lack of available automatic measures for interpretability in the literature. To address this gap, we proposed two model-agnostic objective measures: Compactness and Stability. These measures were derived from interpretability characteristics identified in the literature. Additionally, we validated the interpretability measures by comparing them to subjective measures obtained through a questionnaire. Particularly, the Compactness measure is closely aligned with the subjective concept of interpretability, showing the effectiveness

of this measure.

(**SQ3**) **How interpretable are the resulting models for domain experts?**
To assess the interpretability of the resulting pipeline, we conducted a user study involving domain experts familiar with the dataset. Some of the best models obtained a satisfactory SCS score, indicating a good level of interpretability. However, the best model for the Atalmedial thrombosis dataset received the lowest SCS score. From analyses of the individual question, the DCP algorithm showed a lack of capturing causality and quick understanding. These characteristics are not measured with Compactness and Stability, and therefore more measures for interpretability are needed. However, the validation from domain experts that the pipeline can provide satisfactory interpretable models reinforces the interpretability of the pipeline and its potential application in high-stakes domains.

Considering the comprehensive analysis and evaluation of these subquestions, we have gained valuable insights into the main research question:

(**RQ**) **Is it possible to create an interpretable automated machine learning pipeline for medical classification tasks with an F1-performance comparable with existing non-interpretable tools?**

The successful integration of suitable interpretable models, the proposal of model-agnostic interpretability measures, and the interpretability validation by domain experts contribute to the development of an interpretable automated machine learning pipeline. Furthermore, the comparison of F1-scores with non-interpretable tools demonstrates no loss in performance by using the proposed pipeline. Therefore, our findings demonstrate that it is indeed possible to create such a pipeline without sacrificing F1-performance compared to non-interpretable tools. These findings provide valuable implications for the field of automated decision-making in high-stakes domains.

7. Discussion and future work

This study highlights the potential of an interpretable automated pipeline, particularly in the context of healthcare decision-making. The proposed pipeline achieves comparable F1-scores when compared to existing AutoML tools, indicating that using exclusively interpretable models in an automated machine learning pipeline has no significant impact on performance. This finding is vital, as trust and clarity are paramount in healthcare decisions where interpretability is crucial.

By utilizing only interpretable models, the proposed pipeline aligns with the healthcare domain's requirements for transparent and understandable decision-making processes. The ability to interpret and explain model predictions is highly valuable in healthcare, as it facilitates trust-building between clinicians and machine learning systems. The proposed pipeline offers a suitable solution in this regard.

The effectiveness of the proposed interpretability measures is another notable point to discuss. In particular, Compactness measure exhibits a strong correlation with the perceived level of interpretability by the participants in the user study. This finding indicates that the Compactness measure effectively captures the key aspects of interpretability since it aligns with the subjective understanding of interpretability held by human evaluators. The model-agnostic and objective nature of these measures makes them well-suited for integration into an AutoML framework.

Lastly, the ability to customize the trade-off between interpretability and task performance makes the proposed pipeline a valuable tool for practitioners seeking an optimal solution tailored to their specific needs. This adaptability empowers decision-makers to strike the right balance, taking into account the domain-specific constraints, ethical considerations, and desired levels of interpretability and performance. By adjusting the weighted score ratio, stakeholders can align the model's characteristics with the requirements of the task at hand.

7.1. Limitations

- **Need for local explanations:** Both proposed interpretability measures, Compactness and Stability, have a requirement for local explanations to effectively capture interpretability. While Compactness can utilize global feature importances as input, local feature importances provide a more comprehensive understanding. In the case of the Stability measure, local explanations are crucial as a global explanation can not adequately differentiate the concept of stability. This limitation excludes models that could not provide local explanations from being fully evaluated for their interpretability. In this study, NODE-GA(^{2})M is included as one such model, which cannot provide local explanations. As a result, it received a "NaN" value for Stability and cannot be directly compared with other models. Future research should explore techniques for incorporating global explanations into the proposed interpretability measures, allowing for a more comprehensive assessment across a broader range of models.
- **User Study Questions:** The user study conducted to evaluate the subjective interpretability concept could benefit from certain design considerations that warrant improvement. One limitation is that the study focused on using the System Causability Scale (SCS) method to measure the overall explanatory power. In contrast, only Compactness and Stability measures were utilized in this study. To address this limitation in future research, it is recommended to include additional questions that specifically target and validate the individual measures used in the study. This would provide a more comprehensive assessment of the interpretability of the models and their respective measures. Additionally, the user study could be expanded to incorporate qualitative data collection methods, such as interviews or focus groups. These methods would allow for gathering more in-depth insights into participants' interpretability experiences and perceptions. Qualitative data can provide valuable contextual information and nuanced perspectives that quantitative measures alone may not capture. By including qualitative data, researchers can obtain a richer understanding of participants' subjective experiences and gather more detailed feedback on the interpretability of the models.

Lastly, the use of an objective method, such as the ITR method, could be deployed in future studies to complement the subjective measures. The ITR approach is based on intuition and needs less explanation of the study itself, reducing potential bias and subjective influence. By incorporating the ITR method, the study would benefit from a more objective assessment of interpretability, enhancing the reliability and validity of the findings. These improvements would enhance the reliability and validity of the user study findings.

- **Generalizability and Sample Size Considerations:** It is important to note that the user study involved a relatively small sample size of 9 participants from a specific organization. While the study provides valuable insights, caution should be exercised when generalizing the findings to a larger population or different contexts. Conducting similar studies with larger and more diverse participant groups from other healthcare institutions can strengthen the generalizability of the results and provide a broader understanding of interpretability perceptions. Furthermore, it is worth noting that this study focused specifically on the healthcare domain, and participants with expertise in the medical field were involved in the survey. To assess the effectiveness of the interpretability measures in other domains, such as law or finance, it would be necessary to involve experts from those respective domains in future studies. This would provide a more nuanced evaluation and ensure the interpretability measures are effective across different domains and contexts.
- **Other Limitations:** It is essential to acknowledge that the proposed pipeline and interpretability measures may have additional limitations that were not explicitly addressed in this study. For example, the interpretability measures used in this research focused on specific aspects of interpretability, and other important dimensions, such as domain-specific constraints or fairness considerations, were not thoroughly examined.

7.2. Future work

The current study has provided valuable insights into the interpretability of the models under investigation and is a good step towards automated interpretable machine learning. However, several avenues for future work can expand and enhance the pipeline. The following suggestions can serve as potential directions for further research:

- **Expanding the Range of Interpretable Models:** While the study included four interpretable models, various other interpretable algorithms and techniques could be incorporated. For instance, including basic linear regression models or decision rule algorithms can offer a broader perspective on interpretability and facilitate comparisons among a wider range of models.
- **Exploring a Larger Parameter Space:** Due to time constraints, the study only explored a limited set of parameter settings for the models. To gain a more comprehensive understanding of the models' interpretability, future research could consider expanding the parameter search space. This can involve exploring a wider range of parameter values, incorporating default and non-default settings, and conducting experiments with more varied configurations. This expanded search can reveal the impact on interpretability with different parameter choices and provide better models.
- **Optimizing the Search Process:** To validate a larger search area, optimizing the search process itself can be beneficial. Techniques such as grid search or genetic algorithms can be employed to efficiently explore a wider range of parameter settings and identify optimal configurations. This optimization can improve the pipeline's performance and provide further insights into the efficiency of the proposed interpretability measures.
- **Incorporating Additional Interpretability Measures:** While the current study employed Compactness and Stability to assess interpretability, other measures could be included in future research. The field of interpretability has identified various concepts and characteristics that contribute to the overall inter-

pretability of a model. By incorporating additional measures, such as complexity or completeness, a more comprehensive evaluation of interpretability can be achieved.

- **Subjective Analysis of the Interpretability Measures:** While the current study does analyse the overall interpretability subjective alignment with the measures, the questions of the user study were not specifically related to the individual interpretability measures. The measures will get a more elevated significance if they are supported by specific user studies on the compactness and stability of the explanations.
- **Improving the Data Encoding Step:** The proposed pipeline incorporates simple data encoding techniques, namely one-hot encoding, target encoding, and count encoding. However, an interesting direction for future research is to enhance this step by exploring a broader range of techniques and selecting the most suitable ones for different models. So are the current techniques focussed on converting categorical data to numeric representations, while exploring the reverse direction of numeric to categorical encoding could potentially enhance the interpretability of decision tree models. By expanding the repertoire of data encoding techniques and employing the most effective approaches for the specific models, we can further optimize the pipeline's performance and interpretability. This investigation holds promise for refining the data preprocessing phase.

These future work options present promising opportunities for advancing and improving the proposed pipeline. By implementing these enhancements, we can create a more robust and effective automated pipeline that can be seamlessly integrated into daily practice.

7.3. Conclusion

The proposed pipeline and interpretability measures provide valuable insights and contributions to the field of automated machine learning. By prioritizing interpretability and incorporating model-agnostic measures, the pipeline promotes transparency

and comprehension in the decision-making process. These findings can encourage the development of future AutoML systems in various domains, ensuring the provision of interpretable and trustworthy machine learning models and enabling automatic decision-making processes in high-stake domains such as healthcare in line with the regulations of the European Commission.

REFERENCES

1. Europeia, C., "Artificial Intelligence: Commission Takes Forward its Work on Ethics Guidelines", , April 2019, URL.

2. Medvedeva, M., M. Vols and M. Wieling, "Using machine learning to predict decisions of the European Court of Human Rights", *Artificial Intelligence and Law*, Vol. 28, No. 2, pp. 237–266, 2020.

3. Safdar, S., S. Zafar, N. Zafar and N. F. Khan, "Machine learning based decision support systems (DSS) for heart disease diagnosis: a review", *Artificial Intelligence Review*, Vol. 50, No. 4, pp. 597–623, 2018.

4. Grace, K., J. Salvatier, A. Dafoe, B. Zhang and O. Evans, "When will AI exceed human performance? Evidence from AI experts", *Journal of Artificial Intelligence Research*, Vol. 62, pp. 729–754, 2018.

5. Doshi-Velez, F. and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning", , 2017, https://arxiv.org/abs/1702.08608.

6. Rudin, C., C. Chen, Z. Chen, H. Huang, L. Semenova and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges", *Statistics Surveys*, Vol. 16, No. none, pp. 1 – 85, 2022, https://doi.org/10.1214/21-SS133.

7. "Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)", https://eur-lex.europa.eu/eli/reg/2016/679/oj.

8. Commission, E., "Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts", , 2021, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206.

9. Hutter, F., L. Kotthoff and J. Vanschoren, *Automated machine learning: methods, systems, challenges*, Springer Nature, 2019.

10. Murdoch, W. J., C. Singh, K. Kumbier, R. Abbasi-Asl and B. Yu, "Definitions, methods, and applications in interpretable machine learning", *Proceedings of the National Academy of Sciences*, Vol. 116, No. 44, pp. 22071–22080, 2019, https://www.pnas.org/doi/abs/10.1073/pnas.1900654116.

11. Miller, T., "Explanation in artificial intelligence: Insights from the social sciences", *Artificial intelligence*, Vol. 267, pp. 1–38, 2019.

12. Molnar, C., *Interpretable machine learning*, Lulu. com, 2020.

13. Lage, I., E. Chen, J. He, M. Narayanan, B. Kim, S. J. Gershman and F. Doshi-Velez, "Human Evaluation of Models Built for Interpretability", *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 7, No. 1, pp. 59–67, Oct. 2019, https://ojs.aaai.org/index.php/HCOMP/article/view/5280.

14. Miller, G. A., "The magical number seven, plus or minus two: Some limits on our capacity for processing information.", *Psychological review*, Vol. 63, No. 2, p. 81, 1956.

15. Rudin, C., "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead", *Nature Machine Intelligence*, Vol. 1, No. 5, pp. 206–215, 2019.

16. Krishna, S., T. Han, A. Gu, J. Pombra, S. Jabbari, S. Wu and H. Lakkaraju, "The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective", , 2022, https://arxiv.org/abs/2202.01602.

17. Kovalerchuk, B. and N. Neuhaus, "Toward Efficient Automation of Interpretable Machine Learning", *2018 IEEE International Conference on Big Data (Big Data)*,

pp. 4940–4947, 2018.

18. Hastie, T. and R. Tibshirani, "Generalized additive models: some applications", *Journal of the American Statistical Association*, Vol. 82, No. 398, pp. 371–386, 1987.

19. Lou, Y., R. Caruana, J. Gehrke and G. Hooker, "Accurate intelligible models with pairwise interactions", *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 623–631, 2013.

20. Nori, H., S. Jenkins, P. Koch and R. Caruana, "Interpretml: A unified framework for machine learning interpretability", *arXiv preprint arXiv:1909.09223*, 2019.

21. Popov, S., S. Morozov and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data", *arXiv preprint arXiv:1909.06312*, 2019.

22. Chang, C.-H., R. Caruana and A. Goldenberg, "NODE-GAM: Neural Generalized Additive Model for Interpretable Deep Learning", , 2021, `https://openreview.net/forum?id=g8NJR6fCCl8`.

23. Mainali, P., I. Psychoula and F. A. P. Petitcolas, "ExMo: Explainable AI Model using Inverse Frequency Decision Rules", , 2022, `https://arxiv.org/abs/2205.10045`.

24. Neuhaus, N. and B. Kovalerchuk, "Interpretable Machine Learning with Boosting by Boolean Algorithm", *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 307–311, 2019.

25. Glanois, C., P. Weng, M. Zimmer, D. Li, T. Yang, J. Hao and W. Liu, "A Survey on Interpretable Reinforcement Learning", , 2021, `https://arxiv.org/abs/2112.13112`.

26. Teso, S. and K. Kersting, "Explanatory interactive machine learning", *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 239–245, 2019.

27. Ribeiro, M. T., S. Singh and C. Guestrin, ""Why should i trust you?" Explaining the predictions of any classifier", *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

28. Schmidt, P. and F. Biessmann, "Quantifying Interpretability and Trust in Machine Learning Systems", , 2019, `https://arxiv.org/abs/1901.08558`.

29. Silva, W., K. Fernandes, M. J. Cardoso and J. S. Cardoso, "Towards complementary explanations using deep neural networks", *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pp. 133–140, Springer, 2018.

30. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.

31. Micci-Barreca, D., "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems", *ACM SIGKDD Explorations Newsletter*, Vol. 3, No. 1, pp. 27–32, 2001.

32. Van den Bossche, J., J. De Bock and J. De Brabanter, "Efficient categorical variable encoding for multiclass classification", *Machine Learning and Knowledge Extraction*, Vol. 1, No. 1, pp. 101–121, 2015.

33. Lichman, M., "Machine learning in R: using caret", *Journal of Statistical Software*, Vol. 58, No. 10, pp. 1–26, 2013.

34. Liu, H., R. Setiono and H. Zhu, "Feature selection in knowledge discovery and data mining", *Data Mining and Knowledge Discovery*, Vol. 2, No. 4, pp. 359–394, 1996.

35. Bergstra, J. and Y. Bengio, "Random search for hyper-parameter optimization", *Journal of Machine Learning Research*, Vol. 13, pp. 281–305, 2012.

36. Feurer, M., A. Klein, K. Eggensperger, J. Springenberg, M. Blum and F. Hutter, "Efficient and robust automated machine learning", *Advances in neural information processing systems*, Vol. 28, 2015.

37. Truong, A., A. Walters, J. Goodsitt, K. Hines, C. B. Bruss and R. Farivar, "Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools", , 2019, `https://arxiv.org/abs/1908.05557`.

38. Thornton, C., F. Hutter, H. H. Hoos and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms", *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 847–855, 2013.

39. Jin, H., Q. Song and X. Hu, "Auto-keras: An efficient neural architecture search system", *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1946–1956, 2019.

40. Komer, B., J. Bergstra and C. Eliasmith, "Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn", *ICML workshop on AutoML*, Vol. 9, p. 50, Citeseer, 2014.

41. Olson, R. S., N. Bartley, R. J. Urbanowicz and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science", *Proceedings of the genetic and evolutionary computation conference 2016*, pp. 485–492, 2016.

42. LeDell, E. and S. Poirier, "H2O AutoML: Scalable Automatic Machine Learning", *7th ICML Workshop on Automated Machine Learning (AutoML)*, July 2020, `https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf`.

43. Erickson, N., J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li and A. Smola, "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data", *arXiv preprint arXiv:2003.06505*, 2020.

44. Das, P., V. Perrone, N. Ivkin, T. Bansal, Z. Karnin, H. Shen, I. Shcherbatyi, Y. Elor, W. Wu, A. Zolic, T. Lienart, A. Tang, A. Ahmed, J. B. Faddoul, R. Jenatton, F. Winkelmolen, P. Gautier, L. Dirac, A. Perunicic, M. Miladinovic, G. Zappella, C. Archambeau, M. Seeger, B. Dutt and L. Rouesnel, "Amazon SageMaker Autopilot: a white box AutoML solution at scale", , 2020, https://arxiv.org/abs/2012.08483.

45. Loreaux, E., K. Yu, J. Kemp, M. Seneviratne, C. Chen, S. Roy, I. Protsyuk, N. Harris, A. D'Amour, S. Yadlowsky and M.-J. Chen, "Boosting the interpretability of clinical risk scores with intervention predictions", , 2022, https://arxiv.org/abs/2207.02941.

46. Vincent, J.-L., R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter and L. G. Thijs, "The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure", , 1996.

47. Le Gall, J.-R., S. Lemeshow and F. Saulnier, "A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study", *Jama*, Vol. 270, No. 24, pp. 2957–2963, 1993.

48. Teng, Q., Z. Liu, Y. Song, K. Han and Y. Lu, "A survey on the interpretability of deep learning in medical diagnosis", *Multimedia Systems*, pp. 1–21, 2022.

49. Tonekaboni, S., S. Joshi, M. D. McCradden and A. Goldenberg, "What clinicians want: contextualizing explainable machine learning for clinical end use", *Machine learning for healthcare conference*, pp. 359–380, PMLR, 2019.

50. Lu, J., R. Hutchens, J. Hung, M. Bennamoun, B. McQuillan, T. Briffa, F. Sohel, K. Murray, J. Stewart, B. Chow, F. Sanfilippo and G. Dwivedi, "Performance of multilabel machine learning models and risk stratification schemas for predicting stroke and bleeding risk in patients with non-valvular atrial fibrillation", , 2022, https://arxiv.org/abs/2202.01975.

51. Zadeh, S. A., W. N. Street and B. W. Thomas, "Optimizing Warfarin Dosing using Deep Reinforcement Learning", , 2022, https://arxiv.org/abs/2202.03486.

52. Peng, H., F. Long and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, pp. 1226–1238, 2005.

53. Fernández, A., S. Garcia and F. Herrera, "Learning from imbalanced data sets", *ACM SIGKDD Explorations Newsletter*, Vol. 6, No. 1, pp. 1–12, 2004.

54. Landis, J. R., K. E. Kocher, M. Diaz and A. M. Golden, "Challenges of Machine Learning in Electronic Health Records", *American Journal of Preventive Medicine*, Vol. 53, No. 3, pp. 378–379, 2017.

55. Kavitha, G. and K. Duraiswamy, "Handling Imbalanced Data using Random Undersampling for Medical Diagnosis", *International Journal of Computer Applications*, Vol. 170, No. 5, pp. 12–15, 2017.

56. Khoshgoftaar, T. M., K. Gao, A. Napolitano, R. Wald and A. N. Richter, "Analysis and Experimental Results for Undersampling Imbalanced Healthcare Data", *IEEE Access*, Vol. 7, pp. 75911–75922, 2019.

57. Khanteymoori, A. and M. Kazemi, "Undersampling for Handling Class Imbalance in Medical Datasets", *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–8, IEEE, 2015.

58. Quinlan, J. R., "Simplifying decision trees", *International journal of man-machine studies*, Vol. 27, No. 3, pp. 221–234, 1987.

59. Breiman, L., J. Friedman, C. J. Stone and R. A. Olshen, *Classification and regression trees (No. 5)*, CRC press, 1984.

60. Liu, X. Y., J. Wu and Z. H. Zhou, "Towards interpretable deep neural networks by leveraging pretrained models and supervision", *arXiv preprint arXiv:1811.04551*, 2018.

61. Chen, T. and C. Guestrin, "XGBoost: A scalable tree boosting system", *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794, 2019.

62. Turney, P., "Bias and the quantification of stability", *Machine Learning*, Vol. 20, No. 1, pp. 23–33, 1995.

63. Zafar, M. R. and N. Khan, "Deterministic local interpretable model-agnostic explanations for stable explainability", *Machine Learning and Knowledge Extraction*, Vol. 3, No. 3, pp. 525–541, 2021.

64. Dua, D. and C. Graff, "UCI Machine Learning Repository", , 2017, `http://archive.ics.uci.edu/ml`.

65. Street, W. N., W. H. Wolberg and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis", *Biomedical image processing and biomedical visualization*, Vol. 1905, pp. 861–870, SPIE, 1993.

66. Johnson, A., L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi and R. Mark, "Mimic-iv", *PhysioNet. Available online at: https://physionet.org/content/mimiciv/1.0/(accessed August 23, 2021)*, 2020.

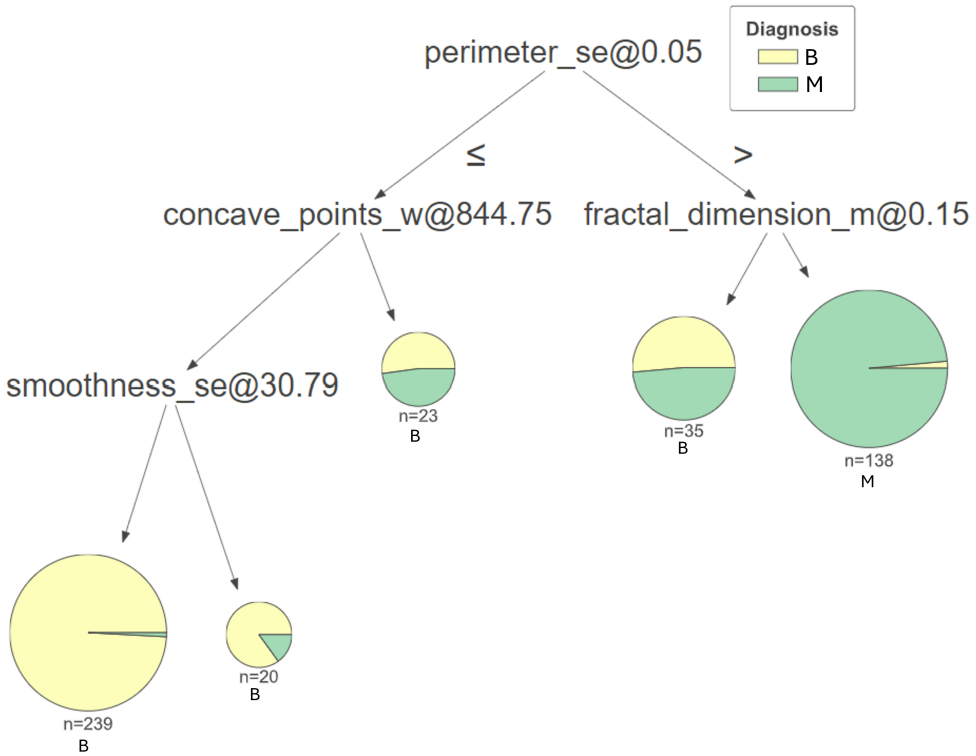
67. Wang, S., M. B. A. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes and T. Naumann, "MIMIC-Extract", *Proceedings of the ACM Conference on Health, Inference, and Learning*, ACM, apr 2020.

68. Holzinger, A., A. Carrington and H. Müller, "Measuring the quality of explanations: the system causability scale (SCS)", *KI-Künstliche Intelligenz*, Vol. 34, No. 2, pp. 193–198, 2020.

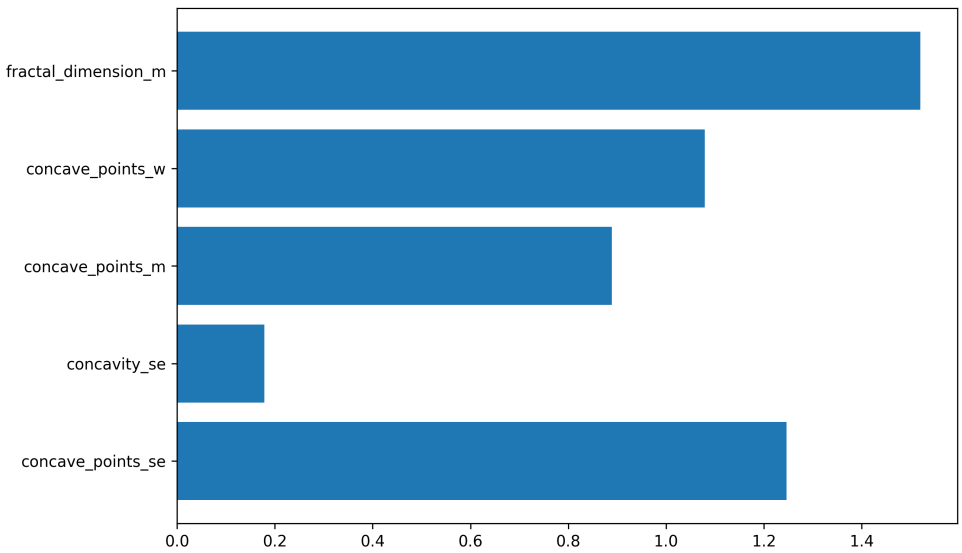
69. Wilcoxon, F., "Individual comparisons by ranking methods", *Biometrics Bulletin*, Vol. 1, No. 6, pp. 80–83, 1945.

APPENDIX A: Example explanations

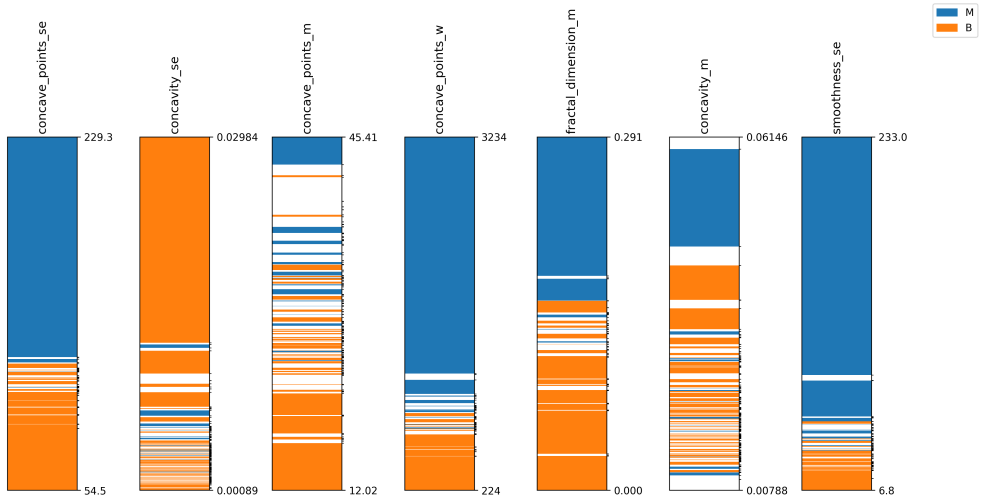
Figure A.1: Global Explanation of the top model per ML method on the WDBC dataset.



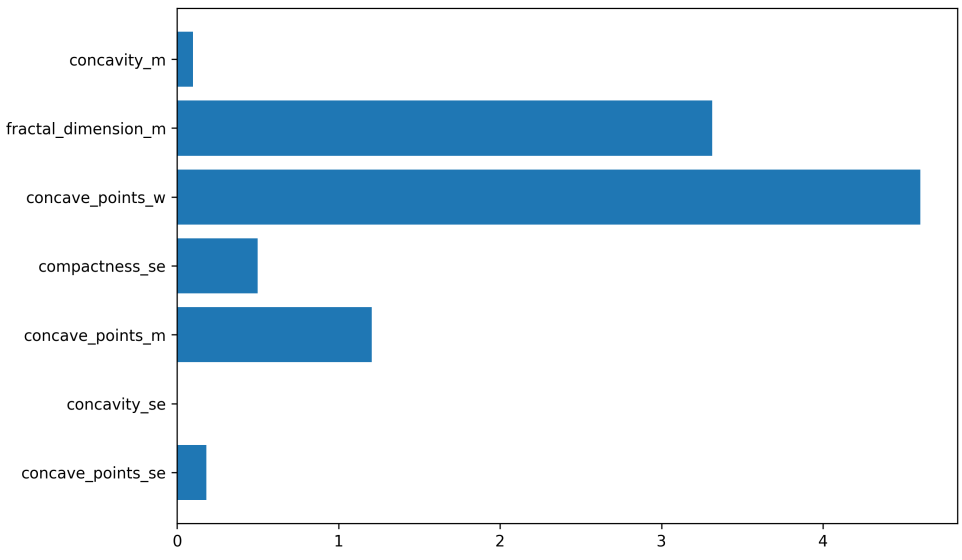
(a) DT with maximum depth on *None*, maximum number of leaf nodes set to 5, maximum samples per leaf set at 20 and using 9 features.



(b) EBM with zero interactions, a learning rate of 0.01, early stopping rounds set at 5 and using 5 features.



(c) DCP with a ratio threshold of 0.9, using voting method 4, and 7 features.



(d) NG with zero interactions, a ℓ_2-λ of 0.05 and using 7 features.