

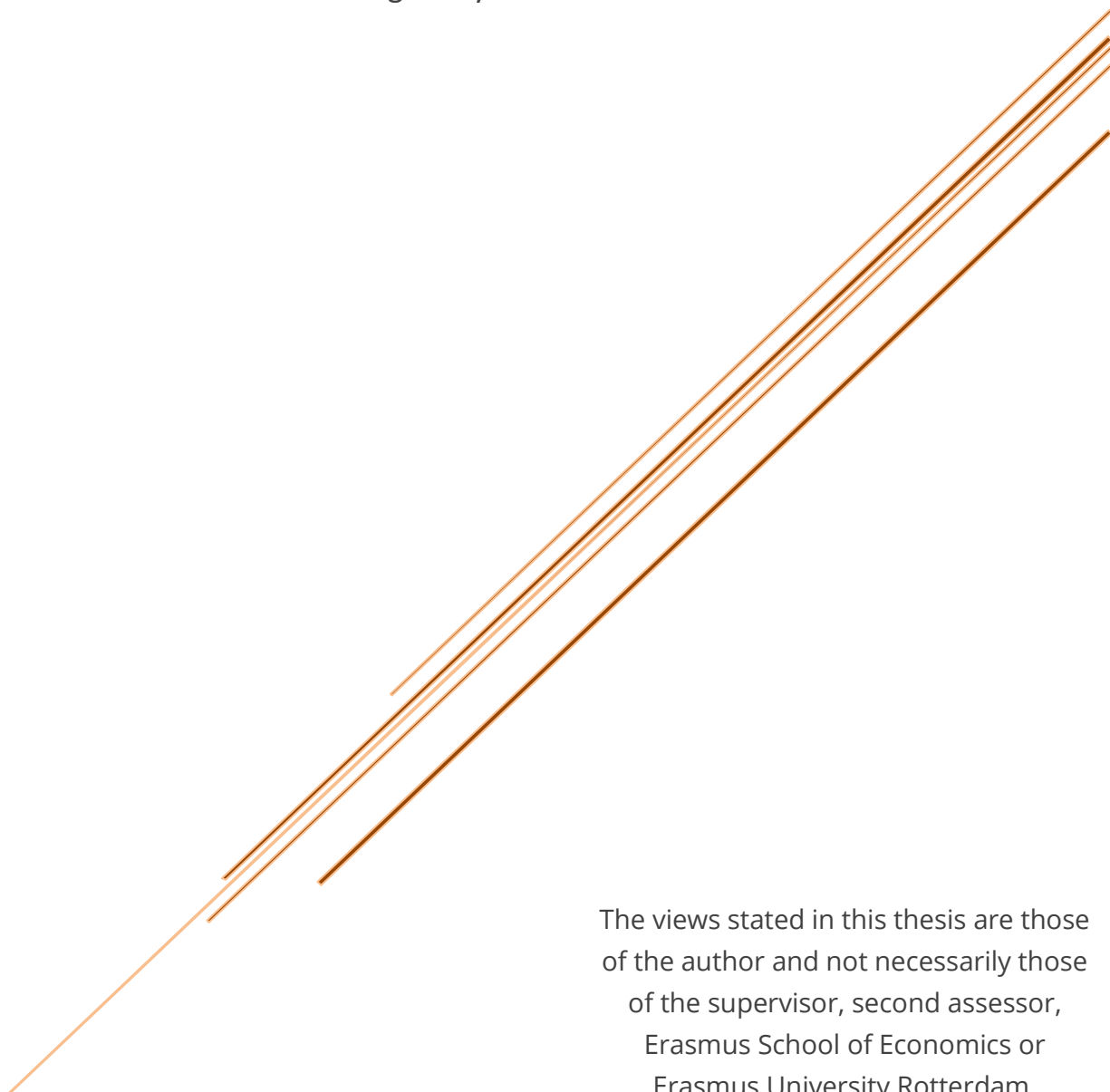
Peeking into the black box: a Rule Fit evaluation.

Author: John Tsaousis 525206

Supervisor: Eran Raviv

Second Assessor:

Erasmus University Rotterdam – Erasmus School of Economics
MSc Data Science & Marketing Analytics



The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Table of Contents

Table of Contents	1
Abstract	3
1. Introduction	4
2. Theoretical Framework	6
2.1 Towards interpretability.....	6
2.1.1 Relevance.....	9
2.1.2 Solutions	10
2.2 Rule Ensembles.....	11
2.2.1 RuleFit	11
2.2.2 Bridging the Gap	12
2.2.3 Related work.....	13
2.2.4 Contribution	14
3. Methodology.....	15
3.1. Ensembles	15
3.2. Extraction of Candidate Rules.....	16
3.3. Meta Ensemble / Rule Fitting.....	17
3.4. Extensions and considerations.....	18
3.4.1 Original features	18
3.4.2 Overlapping rules.....	19

3.4.3 Depth of Trees.....	19
4. Data	21
4.1 Models	22
4.2 Performance Analysis.....	23
5. Results	24
5.1 Performance Results	24
5.2 Interpretability	25
5.2.1 Bank Dataset Demonstration	25
6. Discussion	35
6.1 Performance	35
6.2 Interpretability	36
7. Conclusions.....	39
8. Limitations and Future Work.....	40
Appendix	41
References.....	45

Abstract

Machine Learning models are widely adopted by businesses worldwide. However, without adequately comprehending a model's underlying mechanisms, evaluating its effectiveness can be difficult, leading to potential consequences for companies, policymakers, and the public. Simpler, more transparent models tend to be outperformed by more complex ones (black-box). This is commonly referred to as the interpretability vs. performance trade-off. This paper empirically evaluates the method of RuleFit by Friedman & Popescu (2008) as a solution to this dilemma. RuleFit is assessed in terms of performance and interpretability against a few transparent and black-box models across six datasets. According to this study, RuleFit can achieve a performance level that falls between simple and complex models, striking a meaningful balance, as long as it is tuned toward performance. However, the optimal performance parameters make the global interpretation of RuleFit challenging due to a large set of (extensive) features that need to be considered simultaneously. In that sense, RuleFit is shown to suffer from internal conflicts as its performance must be sacrificed to the level of inherently simpler models to become globally interpretable. On the other hand, RuleFit shows a significant advantage in local interpretation compared to the respective black-box solutions tested. Generally, for RuleFit, finding the optimal parameter levels to achieve an output that is both high-performing and easily interpretable was shown to be a challenging task. Based on these results, RuleFit may not be the optimal method for balancing interpretability and performance in most scenarios. Yet, there are two specific instances where RuleFit could prove advantageous for researchers in its current state. Moving forward, certain areas of improvement are identified for existing RuleFit implementations to address several issues found through this analysis.

1. Introduction

Machine learning models have allowed businesses and researchers to extract essential insights from their data and use them in their operations and services. Ensemble learning methods are among the most powerful, accurate, and commonly used supervised machine learning models. Methods such as Bagging, Boosting, and Random Forests are famous examples that use decision trees as their base learners and are thus known as forests. These models can display a remarkable level of performance even without parameter tuning (Fernández-Delgado et al., 2014; Nalenz & Augustin, 2022). However, to do so, their interpretability is sacrificed to a considerable extent as their complexity increases when fitting the data. This also results in the loss of the intuitive structure of the decision trees. This lack of transparency leads to them being characterized as a “black box.”

Understanding the inner workings of a model is necessary when communicating with the relative stakeholders and decision-makers. As noted by Doshi-Velez and Kim (2017), more than a single metric, such as accuracy, is needed to be a sufficient description of most real-world applications. Furthermore, a model can only be audited when understood, and being unable to explain how a model works opens the risk of biases and discrimination. In other words, comprehending a model means evaluating its desiderata as a machine learning model in general (Doshi-Velez & Kim, 2017).

Complex models with a higher performance suffer in interpretability while more “transparent” models are usually outperformed, a problem known as the interpretability vs. accuracy trade-off. Finding the right balance between the two depends on the use case. For example, predicting the price of a stock may not require a high level of interpretability but high-stake cases such as a medical application might do instead (Nalenz & Augustin, 2022).

One solution proposed by Friedman and Popescu (2008) is RuleFit. It is based on the idea of Rule Ensembles wherein the trees of a forest ensemble are transformed into rule statements and inserted in a (lasso) regression called a meta-learner. This process realizes

the benefits of an ensemble method within a traditional regression framework. The following sections will elaborate upon RuleFit's methodology and related work. Generally, the main idea behind RuleFit is that rules are considered more interpretable than complex trees. At the same time, the meta-learner's penalty factor eliminates the overall model's excess complexity while retaining a satisfactory performance. The paper and model of Friedman and Popescu (2008) will also be the basis of this paper. The goal is empirically testing and analyzing the trade-off between accuracy and interpretability using RuleFit, conventional ensemble methods, and inherently simpler models such as logistic regression. Thus, the main research question is:

“To what extent does RuleFit manage to tighten the gap between accuracy and interpretability?”

Followed by the sub-questions:

- I. *To what extent does RuleFit compete with conventional ensemble approaches in terms of performance?*
- II. *To what extent does RuleFit compete with more straightforward and transparent methods in terms of performance?*
- III. *In which circumstances would using a simpler model be favorable?*
- IV. *In which circumstances would using a more accurate model be favorable?*

The following section provides an overview of the concept and literature of (black box) interpretability. Furthermore, the RuleFit method is described along with its fundamental theoretical concepts and its related studies. Next, the analysis results are presented and followed by a discussion. Finally, conclusions and suggestions are made based on the insights of this research.

2. Theoretical Framework

2.1 Towards interpretability

Interpretability holds no strict definition in machine learning (Lipton, 2016; Doshi-Velez & Kim, 2017). Generally, interpretability relates to the extent to which a model's workings and output can be communicated in understandable terms to a human (Doshi-Velez & Kim, 2017). Researchers commonly interchange the terms interpretability and explainability (Molnar et al., 2020). Still, considering Miller (2017) and Molnar (2022), a distinction of the term "*explanation*" is made in this paper. The term "*explanation*" refers to asking why a prediction is made instead of another one. Some examples are "*Why was this patient diagnosed with a disease?*" or "*What if input X was different?*" (Molnar, 2022).

Lipton (2016) states that the two characteristics of interpretable models are post-hoc (after analysis) interpretability and transparency. Furthermore, Doshi-Velez & Kim (2017) attempt to formalize and formulate a general approach to define further, measure, and evaluate interpretability. They describe interpretability as a multi-faceted concept and propose three evaluation levels, as shown in Figure 1.

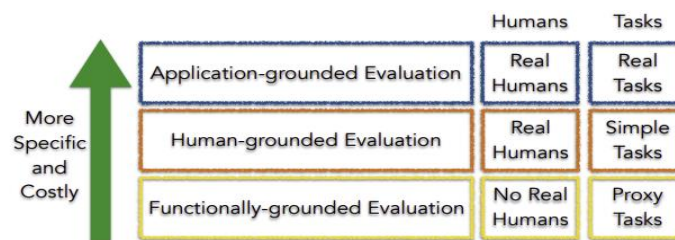


Figure 1 - Taxonomy of evaluation approaches for interpretability, obtained by Doshi-Velez & Kim (2017)

1. Application-grounded: According to the authors, interpretability should be evaluated based on whether the model's predictions align with a user's needs.
2. Human-grounded: This approach evaluates interpretability by how well the model's predictions align with human cognition and decision-making.

-
3. Functionally grounded: This approach evaluates interpretability based on how well the model performs on the task at hand based on domain knowledge.

The authors argue that all three levels are critical when evaluating the interpretability of a model.

The extent to which a model is considered interpretable may also relate to its fundamental theoretical structure. Efron (2020) highlights the differences between traditional regression (transparent) and pure-prediction (black-box) models. First, the author notes that traditional regression models are based on parametric modeling (causality), while pure-prediction algorithms are non-parametric. Non-parametric models may be considered less interpretable as they involve a parameter tuning process related to the model instead of the "data generation process." This is exemplified by Neural Network models, commonly requiring the tuning of hundreds or thousands of parameters. In that sense, parametric models reflect a notion of causality, while the predictions of non-parametric models are much less understood. In other words, traditional regression models are considered "truth-seeking," while pure-prediction models primarily focus on prediction (accuracy), ignoring the underlying truths or mechanisms behind the data.

Furthermore, Efron (2020) explains that attribution (feature importance) in ensemble algorithms can become misleading. It is demonstrated that ensemble algorithms create new high-order combinations of features no longer directly related to the original ones. As a result, identifying the "truly" important features responsible for the model's accuracy becomes complicated. Contrary to that, traditional regression models avoid this issue by attempting to identify the importance of a much smaller set of features. Another difference is that traditional regression models are based on a long history of theoretical development, such as the Maximum Likelihood criterion and Neyman-Pearson inference. On the other hand, pure-prediction algorithms are mostly based on empirical methods, such as the train/test paradigm. The author aspires to reunite the model categories in two ways. The first is through making black-box models more interpretable by becoming more

similar to traditional regression outputs. The second is by realizing the advantages of black-box models within traditional frameworks. The latter can be exemplified by RuleFit in that it brings the accuracy of a Boosting algorithm in a regression framework.

Watson (2022) divides the conceptual obstacles to making a Machine Learning model more interpretable into three categories. The first is *Ambiguous fidelity*. The author highlights that explanations of a model should be faithful to a target. Nevertheless, the exact type of target is underspecified. Explanations can either be faithful to the model (“Why did the model predict x ?”) or to the system (“Which fundamental truth or fact explains the truth conditions of this prediction ?”). This is based on a longstanding debate between two attribution methods, conditional and marginal importance. Marginal measures are considered faithful to the model itself, while conditional measures are considered faithful to the system the model is embedded in. In that sense, the choice of measure must be specified and motivated by the target of interest. The next obstacle is *error rates and severe testing*. According to the author, algorithmic explanations of black-box models cannot be confidently relied upon as they do not undergo severe testing, contrary to regression models as an example. Depending on the complexity of the model, different testing approaches are required. The last obstacle is *process vs. product*. The author claims that algorithmic explanations of black-box predictions produce a static product output. However, it is argued that interpretation should be thought of as more of a process instead. As a solution, Watson (2020) suggests treating explanations as a dynamic exchange between agents. This approach mimics the unfolding of real-world explanations more closely and can more likely lead to understanding by the inquiring agent. The example of a medical diagnosis is used by the author to demonstrate this approach; for the practitioner finding the medical name of the patient’s condition may be sufficient, while the patient may need their condition explained in more familiar terminology instead.

The presence of numerous features and different sources of randomness further compounds the complexity of interpreting a model. Even inherently simpler models, such as a Regression or Decision Tree, may no longer be considered interpretable if hundreds of

parameters and features are included (Molnar et al., 2020). As mentioned, forest ensembles are based on the concept of bootstrapping, a random sampling of observations with replacement. Despite its potential benefits, this source of randomness makes it hard for a human to keep track of the model's exact inputs. The situation becomes even more complicated after multiple different and complex trees are fit on each bootstrapped sample. The issue of randomness as a source of complexity is further emphasized in Random Forest models, which randomly sample a subset of features when fitting each tree.

In understanding machine learning models, it is common to distinguish between two types of interpretability, global and local. Global interpretation refers to understanding the model's behavior as a whole. In contrast, local interpretation focuses on understanding the model's behavior on a particular instance or subset of the data. Global interpretation aims to obtain insights into the overall patterns and relationships in the data. Meanwhile, local interpretation aims to identify which factors drive a model's predictions for a particular observation or group of observations. Both types of interpretation can be important for understanding and improving the performance of machine learning models.

2.1.1 Relevance

As mentioned, Machine Learning algorithms are utilized by many companies for their operations worldwide. Companies that respond successfully to the changes caused by the increase in Artificial Intelligence and Machine Learning applications are expected to survive longer and realize higher profits, increasing by approximately 38% within the next decade (Daugherty & Wilson, 2018). Understanding and creating interpretable algorithms and (machine-learning) models is crucial for companies to safeguard their reputation, mitigate potential legal risks, and maintain positive relationships with stakeholders, policymakers, and the general public. Failure to comprehend the inner workings of these models can have harmful consequences for all parties.

Firstly, algorithms commonly reflect, carry, and perpetuate the biases of their creator (Bogen, 2021; O'Neil, 2016). This raises further ethical concerns and puts democracy at risk. Bogen (2021) discusses the example of hiring algorithms. When "shaping the candidate pool," the algorithms used to reach out to candidates via ads or notifications can re-enforce both gender and racial stereotypes regardless of the company's or creator's intent. As a result, the company may overlook potentially suitable and skilled candidates. At a higher level, this can also lead to discriminatory outcomes, damaging the company's reputation and raising concerns among policymakers and the wider society. Bogen (2021) highlights the need for more thorough regulation and the responsibility of companies to monitor whether their algorithms promote equity. Another relevant example is the COMPAS algorithm, which was regularly used to predict the probability of re-offense by criminals, thus determining their sentences (Angwin et al., 2016). It was later found that black criminals were systematically assigned a higher bias in re-offense probability than white ones. Another detrimental consequence of not understanding the behavior and errors made by an algorithm is exemplified by companies like Uber and Tesla, which experienced significant accidents involving their autopiloting cars. When using the "self-driving" feature, a Tesla car's behavior caused a crash injuring nine people (Kippenstein, 2023), while Uber's autonomous vehicle fatally crashed a pedestrian (Smiley, 2022). These incidents underscore the importance of comprehending the underlying algorithms and ensuring their transparency and explainability. Diakopoulos (2014) states that despite their significant impact on society, most algorithms used regularly remain black boxes.

2.1.2 Solutions

A few methods can extract some level of interpretability from a black box (forest) model after training/post-hoc. Solutions specifically tailored to a method are known as model-specific, while general-fitting ones are known as model-agnostic. Individual Conditional Expectation plots (ICE) (Goldstein et al., 2015), Partial Dependence Plots (PDP) (Friedman, 2001), Variable Importance (Breiman, 2001), and (local) surrogate models (Pruett & Hester, 2016) are to name a few existing solutions. They mostly rely on visually explaining the

model's workings (e.g., PDP, ICE plots) or replicating its behavior more simply (e.g., surrogate). Both approaches aim to increase the transparency of the model. However, regardless of the solution, the intuitive structure of the original decision trees is lost in the context of forests.

Molnar et al. (2022) discuss that (model-agnostic) solutions also have their pitfalls. The authors state that there is no "one-fits-all" solution when choosing a solution for interpretability. They explain that the choice of solution should be motivated by the research goals. Here are a few considerations that the authors discuss regarding the aforementioned solutions. First, they state that feature importance is sensitive to interactions and unable to capture non-linear effects leading to inconsistent results. PDP/ICE may become misleading when a feature marginally captures the effect of other features, or their relationship could be more complex. Surrogate models are characterized as inconsistent as their results depend on the choice of the type of surrogate model. In addition, surrogate models, by definition, lead to a loss of complex interactions found in the original model. Watson (2022) further discusses most of the above concerns in the context of Interpretable Machine Learning. Finally, Molnar et al. (2022) argue that it is often a misconception that simpler models are always expected to be highly outperformed by more complex ones.

2.2 Rule Ensembles

2.2.1 RuleFit

Rule ensembles are one way to create an interpretable machine-learning model, which has remained relatively understudied. One such approach is the one of Friedman and Popescu (2008) known as RuleFit. The idea of Rule Ensembles, and thus of Rule Fit, is that when creating a forest ensemble, instead of directly "learning" its decision trees and their predictions, they are extracted and transformed into rules. This is achieved by traversing the tree from the root to each leaf node. The extracted rules refer to *If-else* statements.

According to the authors, rules are a more intuitive and easily understood construct than deep decision trees (Friedman & Popescu, 2008). Next, Friedman and Popescu (2008) propose using a lasso penalized regression using the extracted rules as predictors. This is commonly referred to as the meta-learner model. The penalty factor reduces unnecessary complexity by shrinking the coefficients of the least important rules to zero. In that sense, one retains satisfactory performance while reducing complexity, bridging the gap between interpretability and accuracy.

2.2.2 Bridging the Gap

The primary way in which RuleFit is expected to provide a higher level of interpretability is that it significantly reduces the final features/coefficients in the model compared to a typical black-box model. In principle, the final output of the RuleFit model is identical to a linear/logistic regression model, meaning that its interpretation also follows the same principles. This aids in interpretability as regression models are relatively well-established and straightforward. Efron (2020) adds that bridging a forest ensemble (i.e., boosting) with a logistic regression aims to retain the advantages of "pure-prediction models within a traditional framework." In terms of performance, the model aims to retain a level similar to a conventional black-box model such as Random Forest. This is because RuleFit leverages the accuracy of complex ensembles to derive its candidate features. This is based on the assumption that RuleFit keeps the high order and complex interactions responsible for the increased performance in the original ensemble method. Molnar (2022) mentioned that RuleFit could also be a way to obtain meaningful and complex interactions instead of requiring the researcher to identify and create them manually.

All mentioned advantages are related to the interpretation of the model on a global level. However, the benefits extend to the local interpretation level. For example, for a given observation or group of observations, it is possible to check which rules are applicable and which are not. Therefore, one can directly understand how a prediction is made for said

observation(s). This can be very useful for applications where micro-level insights are crucial, such as medical cases.

By tightening the gap between accuracy and interpretability, RuleFit can play a crucial role in addressing the challenges faced by businesses and society regarding interpretable models. Providing transparent and understandable insights into the decision-making process can help businesses build customer trust, mitigate reputational risks, and comply with regulations. In the societal context, it can enable individuals to comprehend and question the outcomes of ML algorithms, promoting fairness and accountability. The balance of the model between accuracy and interpretability can allow users to make informed decisions based on comprehensible and reliable outputs while still achieving high predictive performance. By offering a meaningful compromise, RuleFit contributes to a responsible and effective adoption of Machine Learning technologies, benefiting businesses and society. On the other hand, if RuleFit widens the gap, it is essential to examine the potential advantages or trade-offs that come with sacrificing interpretability for a higher accuracy level and vice versa.

2.2.3 Related work

Originally, the candidate rules were obtained from Gradient Boosted decision trees (GBM) (Friedman & Popescu, 2002), but more forests have been explored for this purpose (Nalenz & Villani, 2018). Nalenz & Augustin (2022) argue that the idea of RuleFit suffers from conflicting interests. Specifically, to reduce unnecessary rules (create smooth decision boundaries), RuleFit must first go through a series of different or overlapping rules. Instead, Nalenz & Augustin (2022) propose a univariate clustering of the splits of each covariate. The resulting clusters are referred to as *ensemble conditions* and are used in the meta-learner instead of the individual rules, unlike RuleFit (Nalenz & Augustin, 2022; Friedman & Popescu, 2008). Wei et al. (2019) propose a new regularization method called "smoothly clipped absolute deviation" (SCAD) to assign weights to rules. Its main advantages include an increase in stability and a reduction in overfitting. According to the authors, the

technique can be competitive with more complex models. Lastly, Kundu et al. (2021) show that using accuracy as a measure to assign weights to base learners may worsen the meta-ensemble's final performance, especially in the case of imbalanced datasets. To that extent, they instead propose a function with inputs of various metrics such as Recall, Precision, AUC, and F1 scores.

2.2.4 Contribution

The contribution of this paper can be summarized as follows:

- Expands the literature on the understudied topics of Rule Ensembles and RuleFit, bringing attention to their potential to bridge the accuracy-interpretability trade-off.
- Provides a guideline for researchers on when and how to effectively use RuleFit to strike a balance between interpretability and accuracy, contributing to the ongoing pursuit of narrowing the gap between these two crucial aspects in machine learning models.
- It is the first to utilize the xrf package's implementation of RuleFit empirically, the only package capable of tackling overlapping rule concerns.
- Identifies the shortcomings of the current implementations of RuleFit and makes suggestions for improving the method in its practical application.
- Presents a thorough comparison between the RuleFit method and transparent and black box models. The evaluation is conducted in an objective manner, analyzing the performance and interpretability of each method. The findings offer a clear insight into their individual strengths and weaknesses.

3. Methodology

This section presents a more detailed but not exhaustive description of the methodology of the RuleFit model and its base concepts.

3.1. Ensembles

Firstly, forest-type ensembles are based on the concept of bootstrapping. The main idea is that random sub-samples of the original data are drawn with replacement. Then each decision tree of the forest is trained on its respective bootstrap sample drawn. Each decision tree's training process happens independently or depends on the rest. They are known as parallel and sequential ensembles, depending on the process. Bagging is a popular parallel ensemble, while Boosting is a popular sequential ensemble. Random Forests are an extension of the Bagging method, but they consider random subsets of variables at each split instead of all variables.

As mentioned, the candidate rules were originally obtained from Gradient Boosted Decision trees (GBM) (Friedman & Popescu, 2002; 2008). Regardless, any forest ensemble can be used for the rule extraction process. Each forest method has its potential benefits, such as reducing bias or variance. Therefore, using a different ensemble to extract the candidate rules will affect some aspects of the RuleFit model. Nalenz & Augustin (2022), for example, state that boosted decision trees (GBM) exhibit an elevated level of accuracy, meaning that they are likely to “find interesting subspaces” in the data.

Following the definition of the original RuleFit paper of Friedman & Popescu (2003 & 2008), a general ensemble is mathematically described with the following formula (1):

$$\hat{f}(x) = a_0 + \sum_{m=1}^M a_m \hat{f}_m(x) \quad (1)$$

m refers to the number of trees in the forest with their respective prediction $\hat{f}_m(x)$. The a terms refer to the weight of each tree depending on the ensemble method (i.e., Bagging,

GBM, RF) and the respective method with which predictions are combined (i.e., weighted average, majority-vote). Together all these parameters make up the ensemble prediction function $\hat{f}(x)$.

3.2. Extraction of Candidate Rules

Each node, including the terminal ones, within the forest, produces a candidate rule. As mentioned, rules are obtained by traversing from the root to the node. Mathematically this is the product of all associated indicator functions $I(\cdot)$ within that path. The indicator function of a rule $r_m(x)$, part of m -th tree and for variable x_j , checks whether the condition is true or false, taking a value of 1 and 0, respectively. The following equation summarizes the above:

$$r_m(\mathbf{x}) = \prod_{j \in T_m} I(x_j \in S_{jm}) \quad (2)$$

Wherein T_m refers to the set of features utilized within tree m . S_j refers to all values that input variable x_j can take. S_{jm} is a subset of S_j , whose bounds for feature j are defined by the respective rule conditions (tree-splits). Finally, I is the indicator function taking the value of 1 when input x_j falls within the specified bounds S_{jm} , for each j -th feature, and 0 otherwise.

The type of variable x_j determines the structure of S_{jm} :

- For numerical variables: S_{jm} is an interval with a lower and upper limit defined by the conditions of the extracted rule (e.g., $x_j < X_{S_{jm}, \text{upper}}$).
- For categorical variables, S_{jm} contains a set of numbers corresponding to each category for both ordered and unordered inputs.

Figure 2 visualizes the process of extracting rules from a decision tree using a made-up example based on a few straightforward variables of the *Adult* (1996) dataset of the UCI Machine Learning Repository.

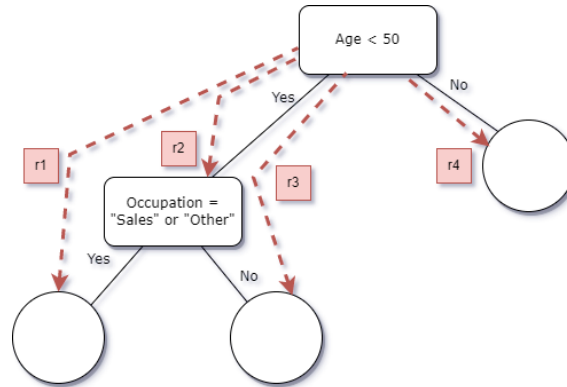


Figure 2 - The process of extracting rules, made-up example created based on variables from the Adult Dataset

The four unique rules that can be extracted from the decision tree of Figure 2 are:

$$r_1(x_{Age}, x_{occupation}) = I(x_{Age} < 50) \cdot I(x_{occupation} \notin \{sales, other\}) ,$$

$$r_2(x_{Age}) = I(x_{Age} < 50) ,$$

$$r_3(x_{Age}, x_{occupation}) = I(x_{Age} < 50) \cdot I(x_{occupation} \in \{sales, other\}) ,$$

$$r_4(x_{Age}) = I(x_{Age} \geq 50)$$

Here rules r_2 and r_4 , are redundant since they are perfectly colinear (evaluate the same conditions) as explained in Fokkema (2020). In such cases, for most implementations of the RuleFit algorithm, the software automatically omits either of the two redundant rules. The same goes if the same rule occurs again within a “child node.” Resultingly, the user has fewer and less complicated rules to keep track of within the final ensemble.

3.3. Meta Ensemble / Rule Fitting

Next, the extracted rules are used as the inputs of a penalized regression, specifically a *lasso* one. In this way, the model pushes rules that do not add “adequate” explanatory power to zero. Theoretically, this retains an elevated level of performance, but also interpretability as fewer parameters are present in the final output. The formula according to which the regression model assigns the respective weights to each rule is the following:

$$\{\hat{a}_m\}_0^M = \arg \min_{\{a_m\}_0^M} \sum_{i=1}^N L(y_i, \hat{f}(x_i)) + \lambda \cdot \sum_{m=1}^M |a_m| \quad (3)$$

The α terms refer to the weight assigned to each rule in the linear model. The function L refers to the Loss Function between the actual value y_i and the predicted $\hat{f}(x_i)$. The exact type of loss function depends on the nature of the data. N is the total number of observations and M is the number of the decision tree in the ensemble. The term λ controls the magnitude of the $L1$ -type lasso penalty. Higher values of λ lead to more shrinkage of the rules’ respective coefficients. Identifying the optimal value of λ is done through the (k -fold) cross-validation process by finding the value which leads to the lowest level of (mean-squared) prediction error on the training data.

3.4. Extensions and considerations

3.4.1 Original features

Fundamentally, the way rules act remains the same as the nodes of a decision tree; they divide the space of the data. However, some relationships, for example, distance in kilometers, are better captured by a continuous variable when compared to rules. Implementations of the RuleFit model by packages such as *H2O*, *pre*, *rulefit*, and *xrf*, allow for both the candidate rules and the original features in the meta-ensemble. In practice, this extends formula (3) to accommodate a term β that assigns weights to the p original features. Both these new parameters are included in the penalization (second term of formula (3)). The structure of the final formula varies per use case; thus, it is not provided

here. In practice, the main idea is that the model becomes more flexible by including the original features.

3.4.2 Overlapping rules.

Another consideration regarding the extraction of rules is that they can overlap. This means that one must keep track of multiple simultaneously true rules. Though it is not necessarily a problem in terms of performance, it could complicate the interpretation of the model. This eventually defeats the purpose of creating an interpretable and straightforward model. The meta-ensemble still follows the same interpretation as a general linear model. Molnar (2022) highlights that the interpretation of the weights in the meta-ensemble becomes unclear in the case of overlapping rules. This is because the standard coefficient interpretation assumes all other variables to be constant (*ceteris paribus*). However, in the case of overlapping rules, since multiple statements are true simultaneously, the “holding all else constant” assumption no longer holds. This problem is visually shown with a made-up example in Figure 3 obtained from Singh et al. (2021). Some implementations of RuleFit from statistical packages, such as the one of *xrf*, can de-overlap such cases to produce fewer but mutually exclusive features. The reader is referred to Holub (2022) for details about the exact process of de-overlapping.

IF $X_1 < 6$ and $X_2 > 6$: $p(+)$ = 0.8
IF $X_1 < 5$ and $X_2 < 7$: $p(+)$ = 0.3
IF $X_1 > 4$ and $X_2 < 4$: $p(+)$ = 0.8

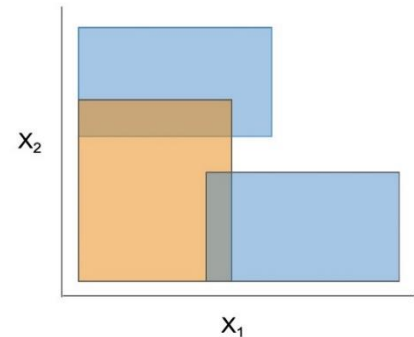


Figure 3 - Overlapping rules visualization obtained by imodels package documentation Singh et al. (2021)

3.4.3 Depth of Trees

One important choice to make prior to generating the initial trees is their depth. The reason behind this is twofold. First, trees with a higher depth can extract more complex interactions and relationships in the training data. This, in turn, enhances the flexibility and performance of the final linear model. On the other hand, highly complex rules are a problem when interpreting the model output, as the user must keep track of an extensive

list of high-order interactions. Moreover, only including high-order interactions could mean that the “main” rules which cover the essential/lower-order interactions are left out. This leads back to the original dilemma between accuracy and interpretability. Friedman & Popescu (2008) discusses that this question is directly related to the nature of the data. If the given research question requires multiple parameters (L), there should mathematically be at least $L + 1$ terminal nodes in the decision tree to generate a rule for each parameter. In other words, a decision tree with L terminal nodes can only generate rules regarding $L - 1$ features (Friedman & Popescu, 2008).

One approach suggested by Friedman & Popescu (2008) is to learn trees of random depth by allowing the depth parameter to vary randomly per tree, revolving around a given average meta-parameter \bar{L} . This ensures that the rules of the linear model include both the main/lower-order and more complex interactions, meaning a more even distribution of the rules. The authors suggest that the optimal level of \bar{L} can be found either via experimentation with cross-validation or based on field knowledge and prior assumptions regarding the data.

4. Data

This section provides an overview of the chosen datasets, the motivations behind their choice, and a brief description of each.

The chosen datasets are obtained from the *UCI Machine Learning Repository* (Dua & Graff, 2019) and are presented in *Table 1* below. They are chosen for the following reasons. First, each dataset has one or multiple characteristics such as a high-class imbalance, numerous features, and multiple observations. Thus, the RuleFit method can be tested under various scenarios. The datasets also require a minimal level of pre-processing. Furthermore, literature has used them to introduce, compare, and evaluate certain methodologies (Van Hulse et al., 2007; Mita et al., 2019; Wei et al., 2019; Fokkema, 2020). Specifically, using some of the mentioned datasets (and more), Mita et al. (2019) introduce and compare the *LIBRE* method, Van Hulse et al. (2007) examine several methods under high class-imbalance cases, Fokkema (2020) introduces the *'pre'* package in R and how it compares to *'RuleFit'*, and finally Wei et al., (2019) introduces and compares the *SIRUS* method. This gives a baseline expectation of the (potential) performance across multiple methods and for each dataset.

Name	# Features	# Observations	Class Proportion (% of Positive Class)	Task
Adult	14	48842	24%	Classification
Bank	17	45211	11%	Classification
Mushroom	22	8124	48%	Classification
Bike Sharing	16	17389	-	Regression
Wine Quality	12	4898	-	Regression
Abalone	9	4177	-	Regression

Table 1 – Datasets obtained by UCI Repository with their observation count, number of features, class proportion, and their task (regression or classification)

The “Adult” dataset is commonly known as “census income.” The goal is to predict whether individuals in the dataset exceed a yearly income of \$50k. The variables in the adult dataset are a mix of categorical and numerical classes. The “Bank” dataset aims to predict whether a customer will subscribe to a term deposit after being contacted via direct marketing. The variables in this dataset are also a mix of categorical and numerical classes. For the

“Mushroom” dataset, multiple goals can be chosen for the classification task. In this case, the goal is to predict whether a mushroom is poisonous or edible. The variables in the dataset are all categorical. The “Bike Sharing” dataset aims to predict the hourly demand for sharing bikes. The variables in the dataset consist of both categorical and numerical data. The “Wine Quality” dataset contains information regarding the composition of various wines next to their assigned ratings by judges. The task is to predict the ratings of a wine based on its composition, such as sugar, sulfate, and alcohol levels. Besides the red or white wine category, the dataset consists of numerical values. Finally, the “Abalone” dataset aims to predict the age of abalone using physical measurements. The variables in the dataset are all numerical except for the sex of the abalone, which is nominal. All datasets required minimal or no pre-processing.

4.1 Models

There are two categories of methods that are compared along with RuleFit. The forest ensemble methods Boosting and Random Forests represent the black box category. On the other hand, the models (penalized) regression and decision trees represent the category of interpretable models. To determine an approximation of the optimal parameter levels in each model, a 5-fold cross-validation process is used on a few hyperparameters. The following parameters are tuned:

- Random Forest: Number of variables tried at each split (mtry)
- Boosting (GBM): Number of trees, Interaction Depth
- Penalized regression: L1 Penalty parameter α
- RuleFit:
 - i. Trees extracted from GBM: Penalty, Shrinkage
 - ii. Trees extracted from Random Forest: Penalty, Number of variables tried at each split (mtry)

The RuleFit models are trained using the *xrf* engine in the *tidymodels* package (*with the rules extension*), which acts as a “wrapper” that simplifies and extends the use of the original *xrf* functionality. The *xrf engine* is chosen instead of other implementations, such as the *RuleFit* and *pre*-packages, as it can de-overlap rules, as mentioned in section 4.2. Furthermore, the *xrf* engine integrates the ability to extract the candidate rules from a Random Forest ensemble. Finally, it is worth noting that by default, the *xrf* package uses 100 trees and a max depth of 3; in the *tidymodels* wrapper, these values are changed to 15 and 6, respectively. Contrary to the *pre* package, however, *xrf* does not natively support sampling trees of random depths as mentioned in section 4.3, proposed by Friedman & Popescu (2008).

In this paper, two versions of RuleFit are constructed, one following the original method by extracting the rules from GBM trees and one using the trees of a Random Forest instead.

4.2 Performance Analysis

In light of Fokkema (2020), the performance of each model is calculated and evaluated using the *real-world data evaluation* design of Hothorn et al. (2005). According to the design, multiple bootstrap samples of the original dataset are generated. In each iteration, a model is trained based on the bootstrap sample while the respective Out of Bag observations are kept aside and used as the test set. With this design, the distribution of a model’s performance is obtained, thus allowing one to assess the model’s stability. The Classification tasks are evaluated using the Area Under the Curve (AUC) metric, while the Root Mean Squared Error (RMSE) is used for regression tasks. Final model performance is measured as the average score across all bootstrap samples.

5. Results

The structure of the following section goes as follows. First, the performance results of a 250-bootstrap process based on the design of Hothorn et al. (2005) are shown and compared across each model and dataset. Then, interpretability is explored and compared within models using examples from the bank dataset.

5.1 Performance Results

The following performance scores are obtained and shown in Table 3. Mean rank refers to the average rank of a model across all datasets and out of the seven models.

Dataset	Random Forest	Decision Tree	GBM	Linear/Logistic Regression	Lasso	RuleFit GBM	RuleFit RF
Adult	0.90 (0.002)	0.76 (0.004)	0.922 (0.002)	0.905 (0.002)	0.853 (0.002)	0.919 (0.002)	0.917 (0.002)
Bank	0.929 (0.002)	0.708 (0.007)	0.932 (0.002)	0.906 (0.003)	0.872 (0.003)	0.911 (0.003)	0.921 (0.003)
Mushroom	1 (0)	0.999 (7.06e-4)	0.999 (2.9e-4)	0.999 (1e-4)	0.974 (0.003)	0.999 (3e-4)	0.999 (0.003)
Bike Sharing	43.33 (0.92)	55.76 (1.41)	43.03 (0.79)	141.91 (1.44)	141.91 (1.45)	56.62 (1.95)	45.97 (1.09)
Wine Quality	0.605 (0.024)	0.744 (0.03)	0.63 (0.02)	0.65 (0.02)	0.64 (0.02)	0.65 (0.03)	0.672 (0.03)
Abalone	2.16 (0.053)	2.56 (0.063)	2.18 (0.057)	2.22 (0.057)	2.24 (0.059)	2.2 (0.052)	2.45 (0.064)
Mean Rank	2	6.3	1.8	4.2	5.5	3.8	4

Table 3 – Average measured performance across 250 Bootstrap iterations following the design of Hothorn et al. (2005) (rounded to three decimal places, Standard Deviation in Parentheses)

The results indicate that, on average, the performance of both RuleFit models lies in between the transparent and black box models, as shown by their mean rank. As expected, the more complex black-box models are consistently the top performers. However, the performance of the simpler models is mixed. Occasionally, they perform similarly to the black-box models and even outperform the RuleFit models, such as in the Abalone and Mushroom datasets. Overall, it can be argued that the performance gap between each model is relatively small on average. This finding is aligned with Molnar et al. (2022), who

argue that it is a misconception that more complex models will consistently outperform simpler models.

Friedman & Popescu (2008) suggest that RuleFit's performance is expected to be competitive with other black-box models. However, Molnar (2022) highlights that in practice, the performance of RuleFit is weaker than what is proposed by Friedman & Popescu (2008). Nevertheless, here the results are partly in line with both authors. Specifically, it is demonstrated that for most classification tasks, RuleFit performs more closely to the black box model category, confirming the suggestions of Friedman & Popescu (2008). On the other hand, for most regression tasks, RuleFit tends to underperform and stand closer to the simple category, as Molnar (2022) implied. One exception within the regression tasks is the Bike dataset, where RuleFit has a significant advantage over the simpler category and performs closest to the black-box category. It must be noted that the relative advantage of RuleFit in classification tasks over regression tasks is, in all likelihood, attributed to the specific datasets and not the type of task.

5.2 Interpretability

The training and testing set for the interpretation demonstration are obtained by randomly sampling one of the bootstrap samples used in the performance evaluation design.

5.2.1 Bank Dataset Demonstration

Rule Fit

The (RF-based) RuleFit model generates 417 features consisting mainly of rules. Interpreting those many features becomes a highly complex process when dealing with lengthy rules. To counter that, a stronger level of penalization is required to reduce the rules in the model. The number of features and test set performance when using different penalty levels can be seen in the Appendix Section (Figure A.7). Initially, RuleFit achieves an AUC of 0.915 on the test set with 417 features. By increasing the lasso penalty and sacrificing the AUC to 0.903, a notable 387 rules are omitted, resulting in only 30 features

remaining in the model. The six largest absolute coefficients out of the remaining 30 features are provided below in Table 4.

Coefficient	Term	Rule Description
-2.7430	(Intercept)	
1.832	Outcome Success P	<i>Original categorical feature of Bank Dataset</i>
1.115	Rule ID 7_83	(age < 57.5) & (contact != 'unknown') & (duration < 640.5) & (duration >= 225.5) & (month == 'jun') & (poutcome != 'success')
1.06	Rule ID 2_57	(day >= 19.5) & (duration < 677.5) & (duration >= 130.5) & (month == 'oct') & (poutcome == 'unknown')
0.777	Rule ID 13_79	(balance < 2250.5) & (duration >= 203.5) & (job != 'retired') & (loan != 'yes') & (month != 'jan') & (pdays >= 382.5)
0.584	Month March	<i>Original categorical feature of Bank Dataset</i>

Table 4 – Six largest absolute coefficients of RuleFit (RF) output (with lambda resulting in 30 total features)

Table 4 shows that the largest coefficient is assigned to the original feature “Outcome Success P” referring to what happened with a client in the previous marketing campaign. As mentioned in Section 3.4.1, including the original features were expected to increase the model’s flexibility. The next largest coefficient is assigned to the extracted rule with ID 7_83. This rule is translated as; *the age of the client is less than 57.5 years, the medium of contact (i.e., phone, email) is unknown, the duration of contact was less than 640.5 but more than 225.5 seconds, the client was called in June, and the last marketing campaign was a success.* When all conditions hold, the predictor takes the value of 1 and 0 otherwise. Therefore, the interpretation of this coefficient would be such that if all else is held constant (*ceteris paribus*), the probability that a client subscribes to a term deposit on average increases by 111.5% compared to the reference category.

The model has 24 further rules/features of similar length that are not listed in Table 4 yet need to be considered. Given the number of conditions in each rule, this can become a challenging and time-consuming process. Rule length is directly related to the depth of the original trees, as mentioned in Section 3.4.3. At this date, the *xrf* package does not support or document a function to sample trees of random depth, as suggested by Friedman & Popescu (2008). Again, doing so is expected to result in more lower-level/base rules which

can make interpretation relatively more straightforward. Lastly, for global interpretability, an adapted method for Variable Importance is shown by Friedman & Popescu (2008). Though the authors' approach is relatively straightforward, there is no known support for the exact method in combination with the *xrf* package (or *R* in general).

To examine the local interpretability of RuleFit, a bank client is randomly sampled from the test set. The sampled client was predicted to subscribe to a term deposit with a probability of 77.6% by the RuleFit model. For this specific client, only three rules and one original feature were applicable out of the 30 total coefficients of the RuleFit model. The conditions of the remaining rules in the model were not met for this individual; thus, their coefficients can be ignored. The contributions of each applicable coefficient towards the prediction of 77.6% probability are shown in Figure 4 below.

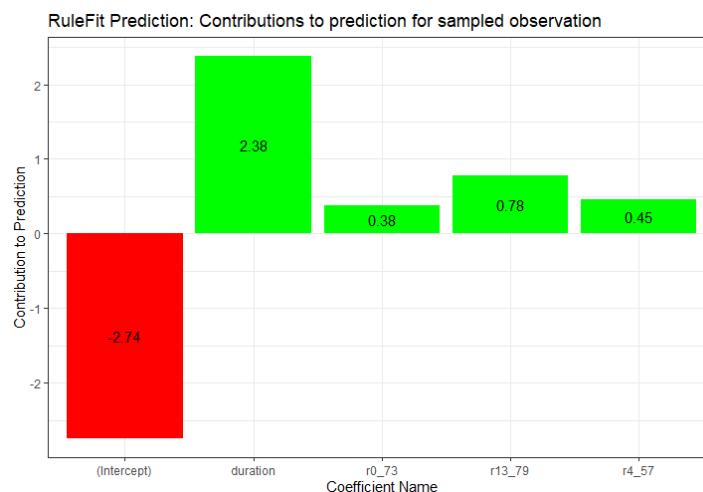


Figure 4 - Contributions to the prediction of RuleFit (RF-based) towards the randomly sampled observation (made in R using ggplot package)

Though the model has 30 features, only 3 rules apply to this specific client. Furthermore, Rule 7_83 was only valid for approximately 1% of clients in the dataset. A researcher can thus identify observations of interest and assess which rules/features apply to them. Assuming that the number of applicable rules to a specific observation remains low even with increased total rules in the model: the local interpretation of RuleFit is shown to be efficient and straightforward, making the model transparent.

Finally, regarding the global and local interpretation, since the last step of RuleFit is a penalized regression, there are no available coefficient p-values. In this case, Wei et al. (2019) observe the largest coefficient values to assess their significance. Nevertheless, they propose for future work to distinguish between original features, low-order rules, and high-order rules. This suggestion is based on Friedman & Popescu (2008), who also propose a slightly refined method for assessing the importance of original features.

Simpler Models

The logistic regression model has 43 different features, and the lasso 29 after 14 coefficients are shrunk to 0. It can be assumed that with fewer features, interpretation becomes clearer. At the same time, in the lasso model, there is a lack of p-values to signify the importance of each variable, in contrast to the logistic model. As explained earlier, a rule of thumb would be to observe the largest coefficients. For this specific test-set, both models give an AUC of 0.9.

The main benefit of the Decision Tree model is its visual interpretability. Figure 5 below shows the structure of the Decision Tree model for the Bank dataset.

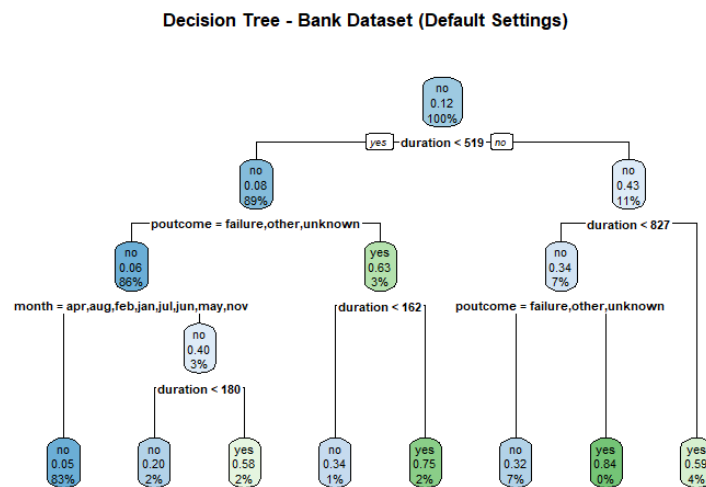


Figure 5 - Visualized Decision Tree model for Bank Dataset (made using rpart in R)

By looking at Figure 5, the classification process can be understood relatively easily. The words “yes” and “no” refer to the model’s prediction, thus whether the client will subscribe to a term deposit.

Despite its straightforward interpretation, the model is by far the weakest performing, scoring an AUC of 0.803 in the test set. Performance could be enhanced by tuning the Decision Tree’s parameters that were now left at default values. However, by allowing the model to become more complex, its length and number of splits would increase considerably. As a result, the model may no longer be considered equally interpretable.

Black-box models

For the black-box model interpretability solutions, Variable Importance Plots and Partial Dependence Plots are utilized for global interpretation. Individual Conditional Expectation (ICE) plots and Local Surrogate (LIME) methods are used for local interpretation. For the demonstration, the random forest model represents the black-box model category. In terms of performance, the random forest classifier scores an AUC score of 93.1% in the test data.

Starting with the Variable Importance Plot in Figure 6, it is easy to obtain an overview of the most important variables. Here importance is measured as the mean decrease in the GINI coefficient. Features that lead to a more significant decrease in the GINI coefficient are considered more important than features that lead to a smaller decrease.

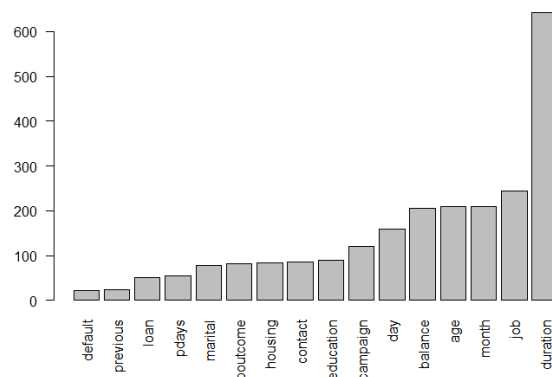


Figure 4 - Bank Dataset Random Forest variable importance measured in the mean decrease in the GINI coefficient (made in R using the RandomForest package)

Figure 6 shows that the duration variable is the most important for the model's splits. It is, however, impossible to establish the direction or relationship of duration with the dependent variable. Hence, whether a higher or lower level of call duration leads to a higher probability of subscribing to a term deposit is unclear. The only insight that can thus be derived is that the duration of the call is a strong determinant of the model's behavior.

A solution to obtain more insight into the relationship of a feature with the prediction of the dependent variable is Partial Dependence Plots (PDPs). PDPs can accommodate either one or two features when visualized. A one-feature PDP shows how the model's predictions vary for different values of the feature holding all other features constant on their average value. For two features, the PDP shows how the predictions of the model change when both features of interest change while holding the remaining features constant on their average values.

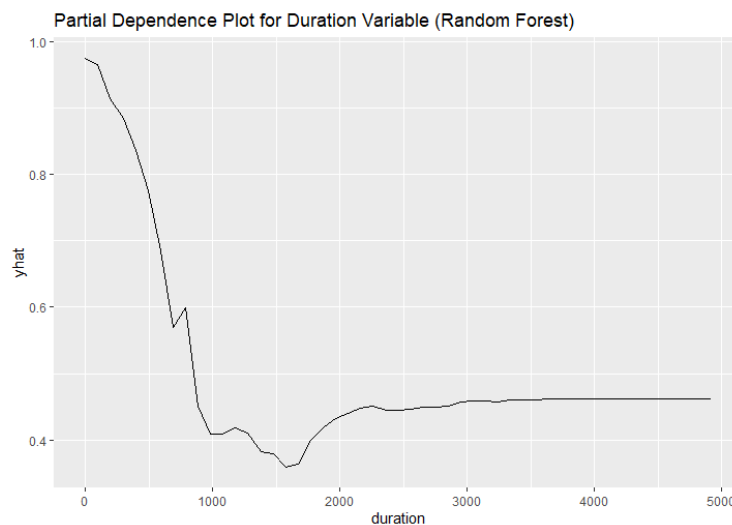


Figure 5 - PDP (Random Forest) change in model predictions for different values of the "duration" variable (made in r using pdp package)

To demonstrate, Figure 7 displays a one-feature PDP in which the model assigns a higher probability of subscribing to a term deposit when a call is shorter than a thousand seconds (approximately 16 minutes).

One concern regarding PDPs is that the features displayed in a PDP are selected by the researcher. As a result, they do not necessarily display the most important features

responsible for the model's behavior. Another concern is that PDPs are susceptible to confounding relationships of features (Molnar et al., 2022). In this case, it could be that the *duration* feature is related to whether the call to a client was placed during working hours, as it would likely last shorter. In that sense, the effect of "duration" would be mediated by the employment status instead. Lastly, PDPs are unable to capture interactions or non-linear feature relationships.

Individual Conditional Expectation (ICE) plots are similar to Partial Dependency Plots (PDPs) in that they effectively visualize the relationship between a change in a feature and the corresponding change in the model's predictions. ICE plots, however, show a more detailed image by showing the change in the model's prediction for all observations instead of only the averaged curve as in the PDP. The benefits of ICE plots are that one can identify any heterogeneity in the data along the variable of interest or spot outliers that deviate from the average trend. For example, Figure 8 below shows the ICE plot for the "duration" variable in the bank dataset. The red line represents the average across all observations equivalent to the PDP in Figure 7.

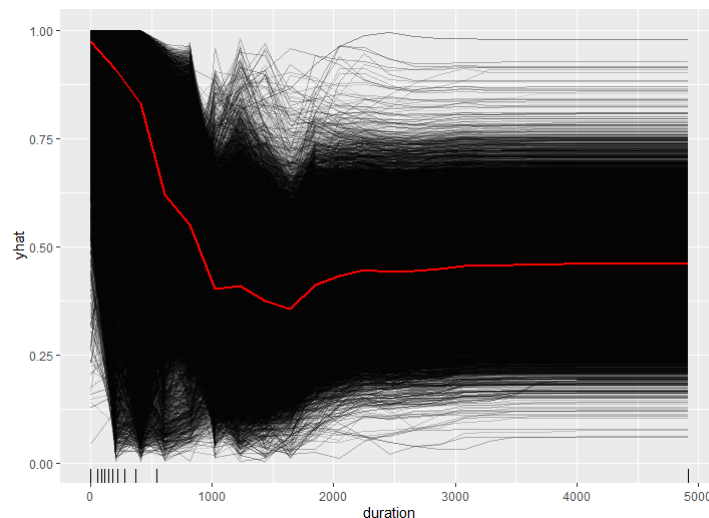


Figure 8 - ICE plot for Random Forest model for duration variable from bank dataset. Each line represents the conditional expectation of a bank client. The red line shows the average of all individual clients (made in R using pdp package)

Figure 8 reveals that there is a lot of variation in the model's predictions among the bank clients in all duration levels. This is indicated by the large spread of the model's predictions

among each observation. Beyond the calling duration of approximately 700 seconds, more predictions start to stand out, and some outliers appear. Overall, the ICE plot suggests that the model depends on more features for its predictions, which is evident as clients with similar call durations are predicted to have considerably different subscribing probabilities.

The concerns of ICEs, similar to PDPs, include being susceptible to confounding variable relationships and being unable to capture interactions or complex and non-linear relationships. Furthermore, in the example of Figure 8, the graph becomes overpopulated due to a larger dataset, making it harder to identify the details in the plot.

Lastly, a LIME solution is demonstrated for the Random Forest. LIME aims to explain how a black-box model predicts one or a few specific observations. To do so, LIME generates a new dataset containing perturbed samples of the chosen observation(s) and records the predictions of the black-box model for them. In other words, it records how the predictions of the black-box model change when the features of the chosen observation are slightly altered. These newly sampled observations are weighted based on their proximity to the original observation of interest. A simpler model is then trained on the predictions of the black box and with respect to the proximity weights. For the complete implementation of local surrogate (LIME) models, the reader is directed to the paper of Ribeiro et al. (2016).

Figure 9 below depicts the results of a LIME explanation using a linear model with 10 features to explain the Random Forest's prediction. All remaining parameters are left at the defaults of the *lime* package. The observation chosen to be explained by LIME is the same one that was sampled for RuleFit's local interpretation.

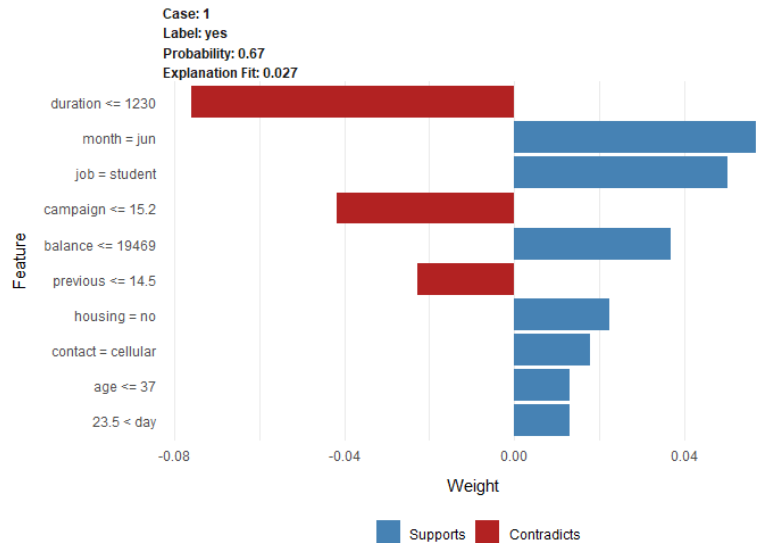


Figure 9 - LIME explanation of random forest model for the randomly sampled observation seen in RuleFit interpretation. Explanation using a linear model with 10 features and default settings (made in r using lime package).

Figure 9 shows that the Random Forest model assigns the client a 67% probability of subscribing to a term deposit. LIME shows features that increased or decreased the probability of subscribing to a term deposit in blue and red, respectively. Overall, features relating to the client's contact have a strong negative impact on the likelihood of subscribing to a term deposit. On the contrary, features relating to the client's personal details increased the likelihood of subscribing to a term deposit.

Despite the benefits of LIME's visual prediction representation, concerns exist. Firstly, a critical concern of LIME is that user changes can heavily influence the outcome, reducing the confidence in LIME's explanations. A crucial parameter in LIME that has to be decided by the user is the definition of proximity weights. This is commonly known as defining the neighborhood around the point of interest. Molnar (2018; 2020) discusses that this choice can turn around LIME's results. The same applies to the remaining parameters that must be chosen by the user, such as the type of simple model and the number of features in it. Yet, there is no correct way to estimate which parameter levels are preferable. Slack et al. (2020) even show that a user/scientist can tweak LIME's parameters to hide biases in the black-box model. Another potential concern is that LIME makes the strong assumption that

linear parameters locally explain the black-box model's prediction. All in all, due to the instability of results, it is difficult to establish trust in the explanations of LIME. Molnar (2020) concludes that the method is promising, but some critical issues must be addressed before confidently using LIME.

6. Discussion

This paper aimed to empirically analyze whether RuleFit can tighten the gap between interpretability and performance and to identify which factors support or hinder this process. In this section, the insights of the results are discussed concerning the research question and the relevant sub-questions.

6.1 Performance

There are two main insights regarding model performances in Table 3 Section 5.2. Firstly, both RuleFit models perform on average between the simpler and black-box models. Based on Friedman & Popescu (2008), the performance of RuleFit was expected to be competitive with black-box models, whereas Molnar (2022) suggests that in practice, RuleFit performs less competitively. In this paper, both propositions are partly supported. This finding may be explained by the following. Theoretically, RuleFit achieves a high accuracy through the complex rules extracted from the initial ensemble. On the other hand, that complexity is later diminished due to the penalty of the lasso meta-learner. This can be considered a case of conflicting interests within RuleFit, supported by Figure A.7 (Appendix), wherein 387 extra rules are required to go from an AUC of 0.903 to 0.913. Therefore, the features responsible for increasing the performance of RuleFit closer to one of the black-box methods are largely omitted through the penalization process, moving performance progressively towards simpler models. From a different angle, this means that when aiming for significantly better performance, RuleFit becomes less easily understandable.

Another insight in terms of model performance is that even though the simpler models remain the weakest performers, their gap with more complex models is relatively smaller than expected, given their complexity. This may be related to the fact that the datasets used in this study required minimal pre-processing with pre-determined tasks, meaning that they were used for accessibility and comparability. It can be expected that with “real-

world” datasets and tasks where complexity increases, the performance gap of simple and black-box models may further widen.

This paper’s first two sub-questions can thus be addressed.

Sub-question i: *To what extent does RuleFit compete with conventional ensemble approaches in terms of performance?*

When tuned for performance, the average performance of RuleFit is similar but still short of conventional forest black-box methods. When tuned toward interpretability, RuleFit scores a much lower level of performance relative to other black-box models.

Sub-question ii: *To what extent does RuleFit compete with more straightforward and transparent methods in terms of performance?*

When tuned for performance, the average performance of RuleFit is higher than that of the simpler models tested, though the gap between them is relatively small. When tuned towards interpretability, RuleFit performs similarly or slightly better than simpler models.

6.2 Interpretability

In Section 5.2, the demonstration of interpretability showcases the difficulties and advantages of utilizing RuleFit. The primary issue with interpreting RuleFit globally was directly related to its conflicting internal interests. When tuned in for performance, RuleFit reaches a meaningful middle ground between simple and black-box models, but many features must be omitted for RuleFit to become globally interpretable. In addition, as shown in the Bank dataset in Table 4, the extracted rules can be lengthy. Even with fewer features, the interpretation process may still be considered relatively complex. One potential solution for addressing rule length is to sample trees of varying lengths. This is expected to result and fewer and less complex features in the model, aiding with interpretation. However, this approach is not currently supported by the *xrf* package. Using model-agnostic solutions such as Variable Importance plots and PDPs could aid the

interpretability of RuleFit at the global level. Yet again, there is no support for either method in r for RuleFit-type models. Overall, the above renders RuleFit redundant, making an inherently simpler model such as a simple regression preferred in most cases.

The main benefit of RuleFit is found at the local interpretation level. A desideratum of an explainable model is the existence of a causal account; *“Why did the model predict x ?”* (Watson, 2022). For each observation of interest, RuleFit was shown to be transparent and explainable, meeting this desideratum reliably. The bank dataset demonstrated that only a few rules are true for a single observation, meaning that the researcher can efficiently understand how a prediction was made (Figure 4). One practical limitation regarding the current RuleFit implementations in r is that users must manually check which rules apply to an observation. This can be a time-consuming task, even with fewer features. Automating this process would enable the user to efficiently retain more features in the model. By doing so, the model becomes more accurate, as Figure A.7 (Appendix) suggests. More rules, however, would come at the expense of global interpretability.

Finding the right balance between performance and interpretability for RuleFit was shown to be a difficult task that requires a trial-and-error approach to determine the optimal combination of parameters. This conclusion applies to both local and global interpretation levels.

In comparison, the models of the simple category are established to be transparent and explainable both locally and globally. Furthermore, several practical solutions were assessed for the black-box category on a global level. Solutions, such as Variable Permutation Importance and Partial Dependency Plots, provided an understanding of a black-box model's essential functions. However, these methods are limited when capturing interaction effects, confounding variables, and complex non-linear relationships (Molnar et al., 2022). These limitations were also explained in the demonstration of the Bank dataset. Regarding the local interpretation of black-box models, Individual Conditional Expectation (ICE) plots and LIME solutions were utilized in this study. Yet, ICE plots are susceptible to the

same concerns as PDPs (Molnar et al., 2022). In addition, as discussed in section 5.2.1, they offered only a limited understanding of the random forest's local interpretation in the bank dataset. LIME, however, proved valuable and straightforward by visually explaining the prediction of a black-box model for a specific observation. However, as concluded by Molnar (2020), concerns still need to be addressed before establishing confidence in LIME's explanations.

The remaining two research subquestions can thus be addressed.

Subquestion iii: *In which circumstances would using a simpler model be favorable?*

A simpler model is almost always preferred when it displays a sufficient level of performance compared to a more complex method, especially when a higher transparency level is required. This case would apply to almost all datasets examined earlier.

Subquestion iv: *In which circumstances would using a more accurate model be favorable?*

A more complex model is preferred when there is a relatively large performance gap compared with simpler models, and a compromise in interpretability is acceptable. One such example in this paper is the bike dataset.

7. Conclusions

Ultimately the primary research question *“To what extent does RuleFit manage to tighten the gap between accuracy and interpretability?”* can be addressed.

Regarding performance, it was shown that RuleFit could reach a meaningful middle ground between transparent and black box models. However, the performance had to be sacrificed to achieve an interpretable model output, down to the level of inherently simpler models. This was shown to be the main weakness of RuleFit, regarding interpretability. This process can also be regarded as an internal conflict of interest within RuleFit’s parameters. Additionally, the simple models performed relatively better than expected, given their complexity compared to the black box models. The most vital point of RuleFit was its local interpretation, especially compared to other model-agnostic solutions for local interpretability. For RuleFit, only a few rules applied to a specific observation, making the model explainable and its inner workings transparent. Overall, the method could benefit from using model-agnostic solutions, but currently, there is no support in r for any of the methods shown.

In conclusion, even though the method is promising, for most cases, RuleFit does not sufficiently tighten the gap between interpretability and performance at its current state. Yet, there are two scenarios where RuleFit may be helpful to a researcher.

1. When the performance gap between transparent and black box models is adequately large, RuleFit could still reach a meaningful middle ground despite the increased penalty required for easier interpretation.
2. When the researcher’s interest primarily lies on the local level and a higher performance level than a simpler model is required, RuleFit can realize an advantage compared to the existing black-box solutions.

8. Limitations and Future Work

This research comes with limitations and suggestions for future work. The datasets utilized in this study were relatively straightforward and intended for accessibility and comparability. This lack of complexity may explain the similarity in the performance between all models. Future studies could use “real word” datasets and tasks to assess both the performance and interpretation of RuleFit. With a higher degree of complexity, real-world datasets may show a change in the performance gap between the different model categories. Another limitation of this study is the number of datasets used. Future studies could utilize more datasets to better understand RuleFit's functionality from both perspectives. Next, a more extensive cross-validation process could be utilized for RuleFit. This may translate into using a different performance measure, a higher number of folds, or a more extensive list of parameters tuned. This way, both aspects of interpretability and performance could be improved.

Moreover, in this paper, the function of sampling trees of random depth was not utilized as it was not supported by the implementation of RuleFit chosen. On the other hand, implementations that allow this functionality do not support the de-overlapping of rules. Both functions are essential to the interpretability of the model. Future work could focus on enabling both functionalities and assessing their joint impact. Similarly, model-agnostic tools could be applied to RuleFit to aid interpretation, especially globally. However, no support exists within r to apply solution methods to RuleFit-type models. Building support for RuleFit models and model-agnostic solutions can significantly improve RuleFit's functionality and reach a more meaningful compromise between accuracy and interpretability. Future work could build on existing packages to support RuleFit-type models and thus address these limitations.

Appendix

Dataset	Random Forest Mtry	GBM # of Trees	GBM Interaction Depth	Lasso Penalty	RuleFit-Penalty		RuleFit (RF) Mtry	RuleFit (GBM) Shrinkage
					RF	GBM		
Adult	13	300	8	0.001	0.003	0.002	10	0.1
Bank	10	400	8	0.003	0.002	0.002	11	0.1
Mushroom	4	50	5	0.001	0.001	1e-4	20	0.001
Bike Sharing	7	400	8	0.231	1e-10	0.1	12	0.2
Wine Quality	6	50	7	0.012	0.01	0.1	9	0.2
Abalone	2	150	5	0.002	0.01	1e-5	6	0.001

Table A1 – Final parameter values obtained from a 5-fold cross-validation process using the packages *caret* and *tidymodels* in R (rounded to three decimal places). **Notes:** For the Random Forest models, the number of trees is held constant at 500. For the GBM models, the shrinkage parameter is constant at 0.1, and the minimum node size is constant at 10.

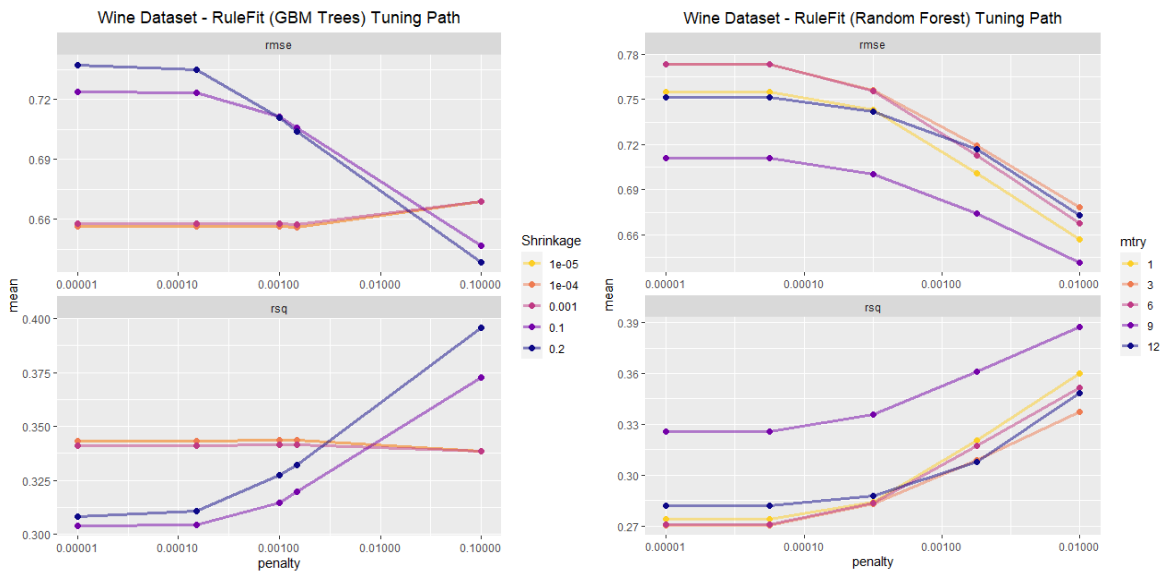


Figure A.1 - RuleFit Cross-Validation Performance path (RMSE & R^2) for the Wine dataset. Left side shows the parameters of penalty and shrinkage for the GBM version. Right side shows the parameters of penalty and mtry for the RandomForest version (made using *tidymodels* package)

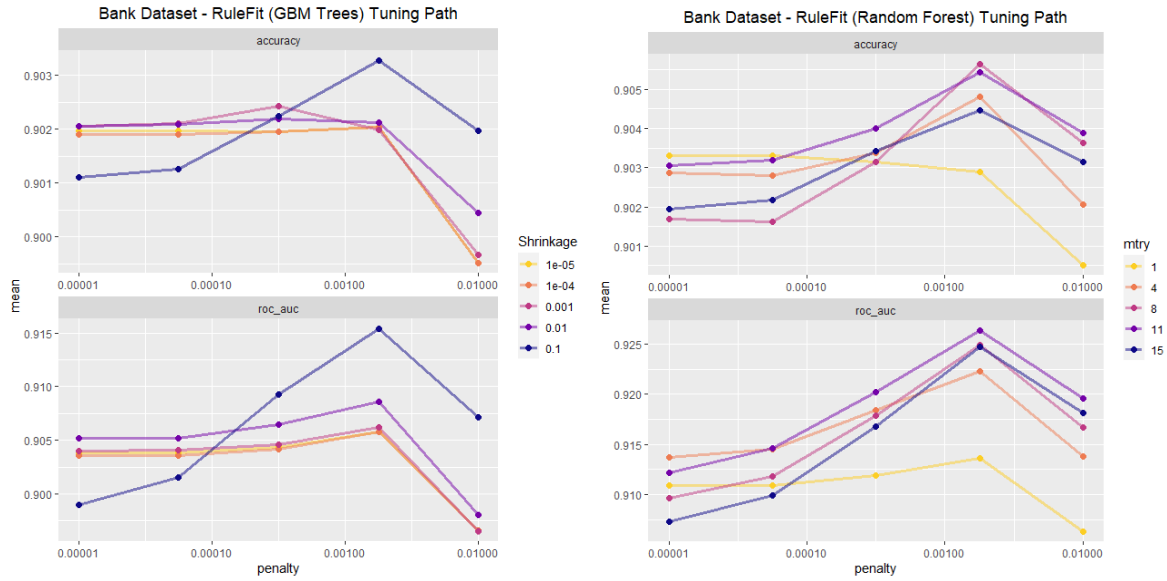


Figure A.2 - RuleFit Cross-Validation Performance path (Accuracy & AUC) for the Bank dataset. Left side shows the parameters of penalty and shrinkage for the GBM version. Right side shows the parameters of penalty and mtry for the RandomForest version (made using tidymodels package)

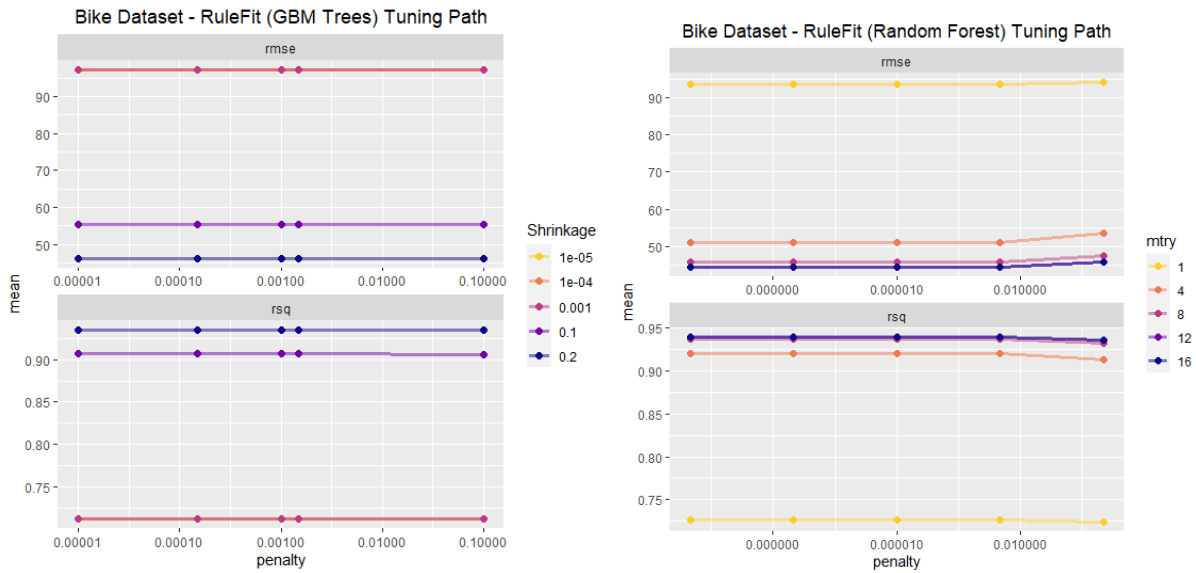


Figure A.3 - RuleFit Cross-Validation Performance path (RMSE & R^2) for the Bike Sharing dataset. Left side shows the parameters of penalty and shrinkage for the GBM version. Right side shows the parameters of penalty and mtry for the RandomForest version (made using tidymodels package)

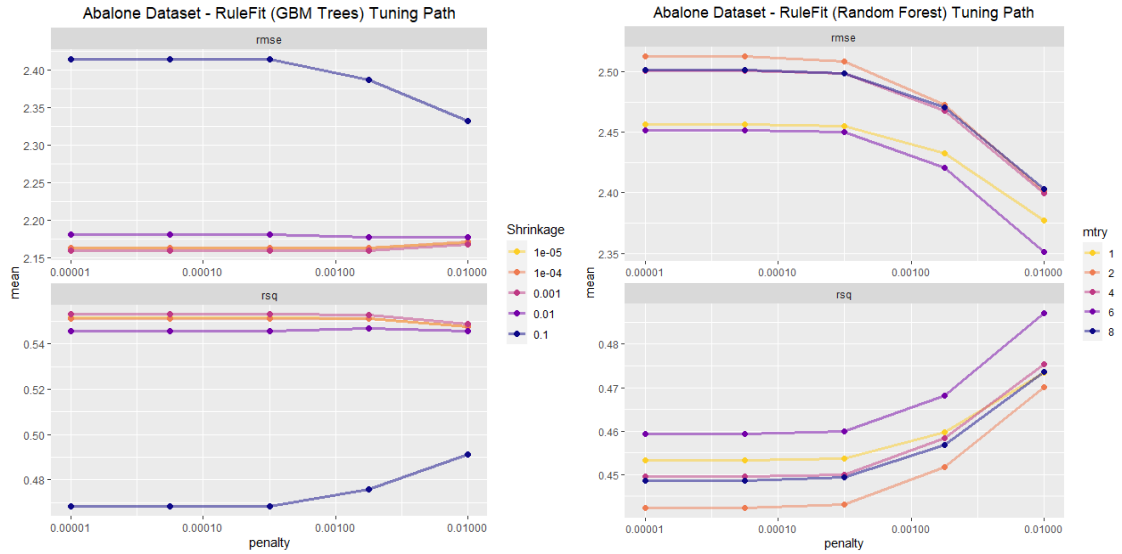


Figure A.4 - RuleFit Cross-Validation Performance path (RMSE & R^2) for the Abalone dataset. Left side shows the parameters of penalty and shrinkage for the GBM version. Right side shows the parameters of penalty and mtry for the RandomForest version (made using tidymodels package)

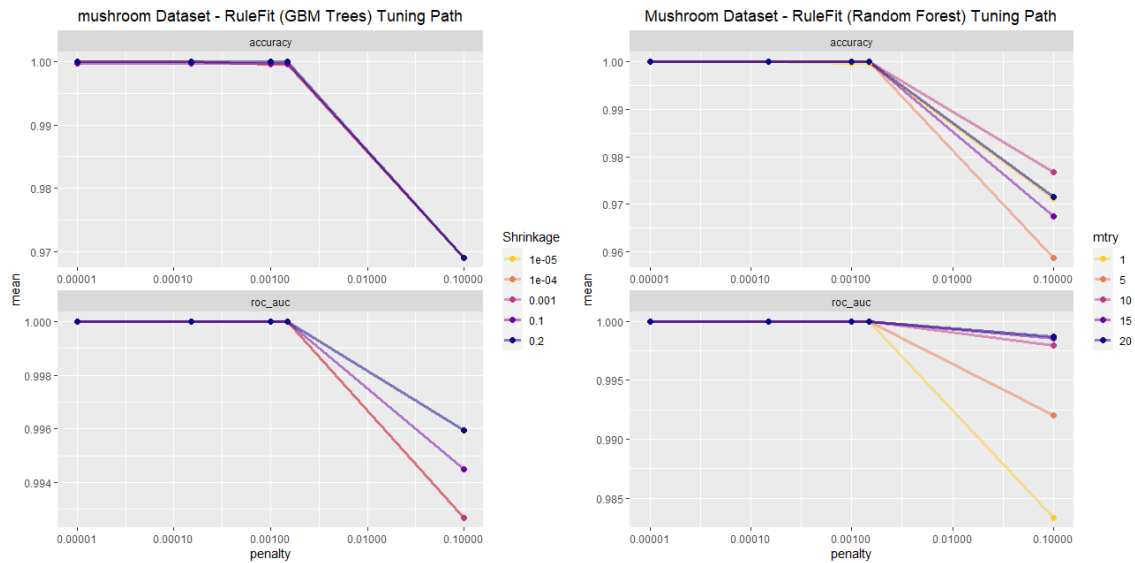


Figure A.5 - RuleFit Cross-Validation Performance path (Accuracy & AUC) for the Mushroom dataset. Left side shows the parameters of penalty and shrinkage for the GBM version. Right side shows the parameters of penalty and mtry for the RandomForest version (made using tidymodels package)

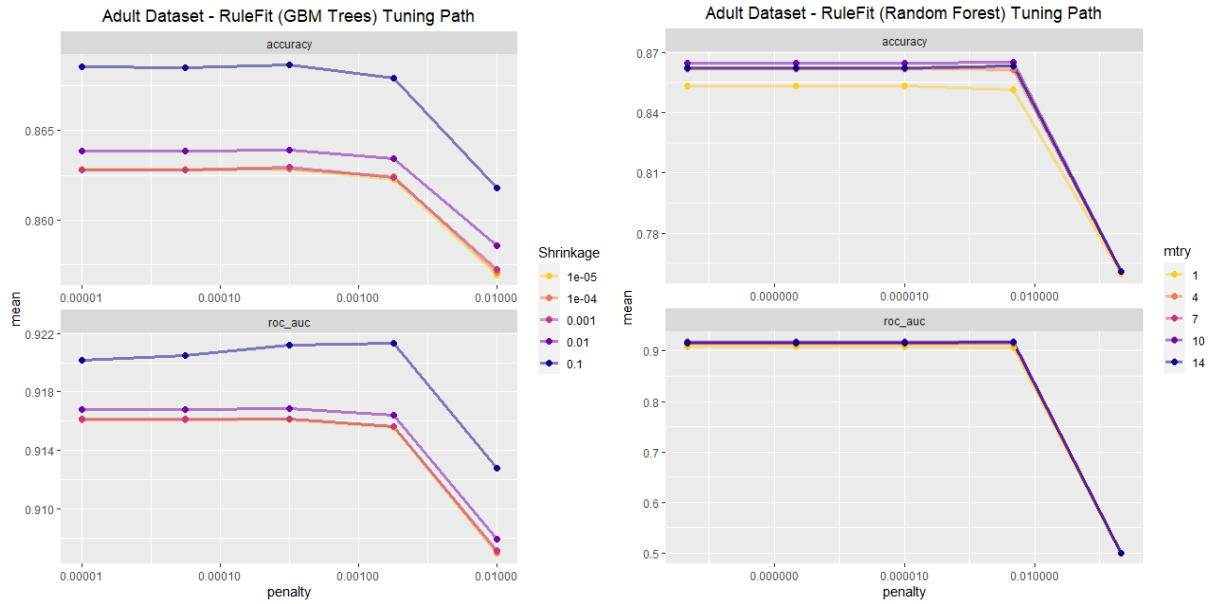


Figure A.6 – RuleFit Cross-Validation Performance (Accuracy & AUC) path for the Adult dataset. Left side shows the parameters of penalty and shrinkage for the GBM version. Right side shows the parameters of penalty and mtry for the RandomForest version (made using tidymodels package)

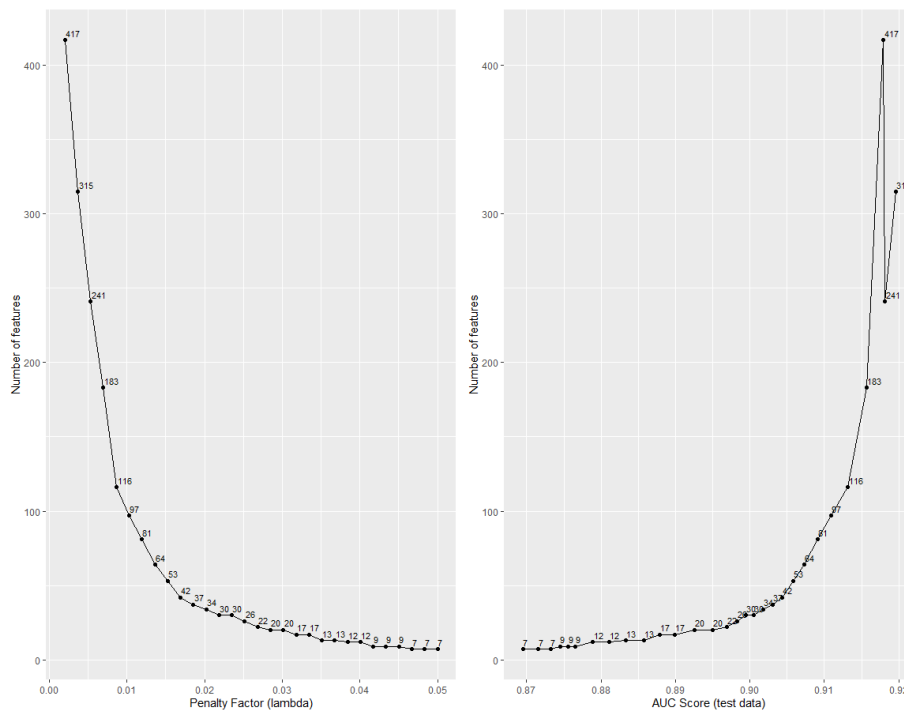


Figure A.7 – RuleFit (Random Forest based) for Bank dataset: The number of features for different penalty values on the training data (left plot) and the respective AUC scores when predicting the test data (right plot). (made in R using ggplot package)

References

1. *Adult*. (1996). UCI Machine Learning Repository. <https://doi.org/10.24432/C5XW20>
2. Angwin, J. A., Larson, J. L., Kirchner, L. K., & Mattu, S. M. (2020, February 29). Machine Bias. *ProPublica*. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>
3. Bénard, C., Biau, G., Da Veiga, S., & Scornet, E. (2021). SIRUS: Stable and Interpretable RULe Set for classification. *Electronic Journal of Statistics*, 15(1). <https://doi.org/10.1214/20-ejs1792>
4. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/a:1010933404324>
5. Bogen, M. (2021, August 30). *All the Ways Hiring Algorithms Can Introduce Bias*. Harvard Business Review. <https://hbr.org/2019/05/all-the-ways-hiring-algorithms-can-introduce-bias>
6. Danuser, Y., & Kendzia, M. J. (2019). Technological Advances and the Changing Nature of Work: Deriving a Future Skills Set. *Advances in Applied Sociology*, 09(10), 463–477. <https://doi.org/10.4236/aasoci.2019.910034>
7. Diakopoulos, N. (2014). Algorithmic Accountability Reporting: On the Investigation of Black Boxes. *Tow Center for Digital Journalism, Columbia University*. <https://doi.org/10.7916/d8zk5tw2>
8. Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *Multiple Classifier Systems*, 1–15. https://doi.org/10.1007/3-540-45014-9_1
9. Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. *Cornell University - ArXiv*. <https://doi.org/10.48550/arxiv.1702.08608>

-
10. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
 11. Efron, B. (2020). Prediction, Estimation, and Attribution. *Journal of the American Statistical Association*, 115(530), 636–655.
<https://doi.org/10.1080/01621459.2020.1762613>
 12. Fanaee-T, H., & Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2–3), 113–127.
<https://doi.org/10.1007/s13748-013-0040-3>
 13. Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1), 3133–3181.
 14. Fokkema, M. (2020). Fitting Prediction Rule Ensembles with R Package pre. *Journal of Statistical Software*, 92(12). <https://doi.org/10.18637/jss.v092.i12>
 15. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5). <https://doi.org/10.1214/aos/1013203451>
 16. Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3). <https://doi.org/10.1214/07-aos148>
 17. Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1), 44–65.
<https://doi.org/10.1080/10618600.2014.907095>
 18. Hara, S., & Hayashi, K. (2016). Making Tree Ensembles Interpretable. *ArXiv: Machine Learning*. <https://arxiv.org/pdf/1606.05390>
 19. Holub, K. (2022). *xrf: eXtreme RuleFit*. <https://CRAN.R-project.org/package=xrf>

-
20. Hothorn, T., Leisch, F., Zeileis, A., & Hornik, K. (2005). The Design and Analysis of Benchmark Experiments. *Journal of Computational and Graphical Statistics*, 14(3), 675–699. <https://doi.org/10.1198/106186005x59630>
 21. Huguenin, N. (2020). Thinking inside the box: exploratory research on factors influencing attainment of trustworthy algorithms. *Business Information Management*. Retrieved from <http://hdl.handle.net/2105/54520>
 22. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R*. Springer Publishing.
 23. Klippenstein, K. (2023, May 16). Klippenstein, K. (2023b, June 2). Exclusive: Surveillance Footage of Tesla Crash on SF's Bay Bridge Hours After Elon Musk Announces "Self-Driving" Feature. *The Intercept*. <https://theintercept.com/2023/01/10/tesla-crash-footage-autopilot/>
 24. Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. *Knowledge Discovery and Data Mining*, 202–207. <http://dblp.uni-trier.de/db/conf/kdd/kdd96.html#Kohavi96>
 25. Kundu, R., Das, R., Geem, Z. W., Han, G. T., & Sarkar, R. (2021). Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLOS ONE*, 16(9), e0256630. <https://doi.org/10.1371/journal.pone.0256630>
 26. Lipton, Z. C. (2016). The Mythos of Model Interpretability. *Cornell University - ArXiv*. <https://doi.org/10.48550/arxiv.1606.03490>
 27. Miller, T. (2017). Explanation in Artificial Intelligence: Insights from the Social Sciences. *ArXiv (Cornell University)*. <https://arxiv.org/pdf/1706.07269.pdf>

-
28. Mita, G., Papotti, P., Filippone, M., & Michiardi, P. (2019). LIBRE: Learning Interpretable Boolean Rule Ensembles. *ArXiv (Cornell University)*.
<http://export.arxiv.org/pdf/1911.06537>
29. Molnar, C., Casalicchio, G., & Bischl, B. (2020). Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges. *Communications in Computer and Information Science*, 417–431. https://doi.org/10.1007/978-3-030-65965-3_28
30. Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Christoph Molnar.
31. Molnar, C., König, G., Herbringer, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., Casalicchio, G., Grosse-Wentrup, M., & Bischl, B. (2022). General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models. *Lecture Notes in Computer Science*, 39–68. https://doi.org/10.1007/978-3-031-04083-2_4
32. Nalenz, M., & Villani, M. (2018). Tree ensembles with rule structured horseshoe regularization. *The Annals of Applied Statistics*, 12(4). <https://doi.org/10.1214/18-aos1157>
33. Nalenz, M., & Augustin, T. (2022, May). Compressed Rule Ensemble Learning. In International Conference on Artificial Intelligence and Statistics (pp. 9998-10014). PMLR.
34. O’Neil, C. (2016). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. <https://ci.nii.ac.jp/ncid/BB22310261>
35. Pruet, W. A., & Hester, R. L. (2016). The Creation of Surrogate Models for Fast Estimation of Complex Model Outcomes. *PLOS ONE*, 11(6), e0156574.
<https://doi.org/10.1371/journal.pone.0156574>
36. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. <https://doi.org/10.18653/v1/n16-3020>

-
37. Singh, C., Nasser, K., Tan, Y. S., Tang, T., & Yu, B. (2021). imodels: a python package for fitting interpretable models. In *Journal of Open Source Software* (Vol. 6, p. 3192). The Open Journal. <https://doi.org/10.21105/joss.03192>
38. Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling LIME and SHAP. <https://doi.org/10.1145/3375627.3375830>
39. Smiley, L. (2022, March 8). 'I'm the Operator': The Aftermath of a Self-Driving Tragedy. *WIRED*. <https://www.wired.com/story/uber-self-driving-car-fatal-crash/>
40. *UCI Machine Learning Repository: Data Sets*. (2023). Accessed on March 2023 <https://archive.ics.uci.edu/ml/datasets.php>
41. Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. *International Conference on Machine Learning*. <https://doi.org/10.1145/1273496.1273614>
42. Watson, D. I. (2022). Conceptual challenges for interpretable machine learning. *Synthese*, 200(2). <https://doi.org/10.1007/s11229-022-03485-5>
43. Wei, D., Dash, S., Gao, T., & Günlük, O. (2019). Generalized Linear Rule Models. *International Conference on Machine Learning*, 6687–6696. <http://proceedings.mlr.press/v97/wei19a/wei19a.pdf>