İSTANBUL UNIVERSITY PRESS

Journal of Data Analytics and Artificial Intelligence Applications, D3AI 1, 1 (January 2025): 14–35

https://doi.org/10.26650/d3ai.1606958

Submitted: 24.12.2024
Revision Requested: 03.01.2025
Last Revision Received: 20.01.2025
Accepted: 20.01.2025
Published Online 23.01.2025

Journal of Data Analytics and Artificial Intelligence Applications

Research Article

Open Access

# Children of the Tree: Optimised Rule Extraction from Machine Learning Models

Hilal Meydan [1] & Mert Bal [1]

[1] Yıldız Technical University, Department of Mathematical Engineering, İstanbul, Türkiye

**Abstract**

The *"Children of the Tree"* algorithm provides a strong understanding of how the imbalanced dataset is classified by extracting rules from each tree of the Random Forest (RF) model. Basically, it converts the divisions created at each node of the trees into "if-then" rules and extracts individual rules for each tree by differentiating the general "community model" perception in the RF. Thus, the algorithm finds the "Children of the Tree" by converting the forest into a rule set. This study, developed on the "German Credit Data Set", which is one of the banking data sets on which many studies have been conducted in the literature; determines the rules that cause to fall into that class(class good or class bad) for candidate customers. In this way, the bank would see the rules for potential customers belonging to the risky class and have the chance to recommend the alternative plans/products that are suitable for their risk strategy to their potential customers. The study evaluates rule validity and reliability using association rule mining metrics—support, confidence, lift, leverage, conviction - calculates "Minimum Description Length" (MDL), and ranks rules by "support" and "MDL cost" to extract the simplest rules for each class. It addresses risk management in banking and marketing needs, using MDL cost and SMOTE to handle imbalanced datasets, setting it apart from other algorithms.

**Keywords** Children of the Tree · Machine Learning · Rule Extraction · Random Forest · Minimum Description Length

Citation: Hilal Meydan, and Mert Bal. 2025. Children of the Tree: Optimised Rule Extraction from Machine Learning Models. *Journal of Data Analytics and Artificial Intelligence Applications* 1, 1 (January 2025), 14-35. https://doi.org/10.26650/d3ai.1606958



Corresponding author: Hilal Meydan hilalmydan@gmail.com

# 1. Introduction

This article will discuss the “Children of the Tree” algorithm, which we introduce as a novel algorithm to the Machine Learning (ML) literature. *“Children of the Tree”* provides rule extraction for candidate bank customers, which we explain through Random Forest (RF) as the tree-based machine learning model. This study helps banks and non-bank financial institutions accurately detect potential customer risks and make decisions aligned with their risk strategies. The rules needed are extracted accurately and in a less complex manner. The algorithm can work on imbalanced data sets.

In this study, we examine how the *“Children of the Tree”* algorithm can efficiently generate rules for customer classification in the banking sector. This approach helps us understand the dataset classification by extracting decisions from each node of the tree-based models. These decisions are then converted into rules. It provides a powerful method for analysing the risk status of ongoing customers and gaining meaningful insights for candidate customers. In this way, the factors causing customers or groups to fall into certain classes are revealed. This enables banks to make strategic decisions in marketing, risk management, and customer relations while developing better strategies for potential customers.

There are studies in the literature on the use of rule extraction and tree-based models in customer classification. However, these studies often overlook the simultaneous evaluation of the reliability, simplicity, and validity of the extracted rules. Additionally, studies that focus on these aspects do not incorporate 'Minimum Description Length' (MDL)-based simplicity research. The *“Children of the Tree”* approach addresses this gap by measuring the confidence and validity of each rule using quantitative metrics for interestingness. Additionally, it selects simpler rules by performing an MDL cost analysis. Thus, it reduces complexity and ensures that the obtained rules are based on a more meaningful and simple basis. This is essential in high-risk sectors such as banking because the accuracy of the rules used in customer classification and risk analysis can directly affect the business strategies of banks.

In addition, generating insights for candidate customers by seeing the current risk status of the bank through existing customers will be a meaningful and valid analysis. In this respect, the *“Children of the Tree”* algorithm will provide a unique solution in customer classification and risk analysis in the banking industry and will make significant contributions. This study effectively addresses imbalanced datasets by incorporating the SMOTE technique, demonstrating its strength in this area.

In this study, Chapter 1 reviews the existing rule extraction algorithms. Mathematical programming-based algorithms and machine learning-based algorithms were investigated and verbally compared with our algorithm "Children of the Tree". In addition, Chapter 1 briefly touches on the background of the association rule mining metrics and the Minimum Description Length (MDL) method, which is used to evaluate the inferred rules that are the basis of our algorithm. In addition, in this section, the SMOTE method is briefly explained and the dataset used in the study is introduced. Chapter 2 explains the basic principles, working steps, and mathematical background of our algorithm. In addition, it is discussed in detail how the metrics used in the evaluation of the rules extracted in our algorithm are calculated. Chapter 3 covers the applications of our algorithm to the dataset. In this section, the practical results and performance of our algorithm are conveyed. Chapter 4 is the discussion and conclusion section. In this section, our algorithm is compared with RuleFit [10] and Anchors [18], which are widely known in the literature and can be used as open source, on the German Credit Data dataset. According to the benchmark results, our algorithm generally produces a higher F1 score than these rule extraction algorithms. In addition, our algorithm could provide high and close F1 scores for both classes, considering the minority class as well as the majority class. This shows that our study exhibits superior performance on unbalanced datasets.

According to the benchmark results, the Children of the Tree algorithm demonstrates superior performance, achieving an overall F1 score of 0.80 compared to 0.74 for RuleFit. Specifically, the F1 score for Class 1 (good class) is 0.80 for Children of the Tree and 0.73 for RuleFit, while for Class 2 (bad class), the F1 score is 0.81 for Children of the Tree and 0.75 for RuleFit. Additionally, when compared to Anchors, the Children of the Tree algorithm stands out by evaluating rules across broader metrics. Anchors provide high precision values, but these are often associated with very small subsets of the dataset, as seen in its low coverage values (e.g., 0.0104 and 0.0147 for Class 1 and Class 2, respectively). In contrast, Children of the Tree achieved confidence values analogous to precision, such as 0.78 for Class 2 and 0.69 for Class 1, while maintaining significantly higher support values (up to 0.48 for Class 2 and 0.40 for Class 1). This balance between confidence and support ensures that the generated rules are both meaningful and scalable. By addressing the challenges of imbalanced datasets and generating reliable, interpretable rules, Children of the Tree proves to be a robust and practical alternative to existing methods like RuleFit and Anchors.

### 1.1. Rule Extraction in Literature

The issue of rule extraction from ML models has been discussed and used in many areas (e.g., finance, healthcare, etc.). Until now, many researchers have conducted studies ranging from the interpretation of models, explainable/interpretable models, to the detection of model tendencies through rule extraction, and even the evolution of these rules into use by business units/owners. Rule extraction from tree-based machine learning models is also an important approach to make the decision processes of the model more interpretable. In ensemble models such as Random Forest, the analysis of rules from multiple decision trees is used to increase the explainability of the model and to obtain reliable decisions [1]. In this context, various algorithms that work in the form of "if-then" rules stand out in the literature.

*inTrees* Framework [2] ranks the rules extracted from the RF according to metrics such as frequency, error rate, and length. It reduces the complexity of the rules and makes them simpler and more understandable. However, it does not use the MDL technique, which is the way to reduce the complexity and provide the simplicity by analysing the model's error and rule length. In addition, it has some limitations in the area of rule overlap and explainability. *ExtractingRuleRF* [3] extracts and ranks the rules from RF using a greedy algorithm. This method prioritises predictive accuracy while limiting the interpretability coverage. *SIRUS* [4] is a method derived from RF and based on the rule frequency. It focuses on creating shorter and more stable rules. *RF+HC* [5] reduces the number of rules extracted from RF using the Hill-Climbing algorithm. This method applies optimisation to create small and meaningful rule sets. *defragTrees* [6] simplifies RF rules using a Bayesian model selection algorithm and optimises predictive performance. It preserves explainability while reducing complexity. *ForEx++* [7] generates high-quality rule ensembles from decision forests. It presents a framework based on average metrics and focuses on predictive performance. *MIRCO* [8] uses mathematical programming to minimize rule heterogeneity and complexity. This method produces rules that better represent the minority class.

*OptExplain* [9] extracts rules using logical inference, sampling and optimisation techniques. It offers the ability to explain particularly large data coverage. *RuleFit* [10] combines rules extracted from RF nodes with sparse linear regression. However, it may experience instability when working with highly correlated rules. *Node Harvest* [11] uses the rules extracted from RF as a weighted prediction model. It works with non-negative weights and provides more explainable models. *Forest-ORE* (Optimal Rule Ensemble) [12] is a method that generates an explainable rule ensemble from Random Forest models. This method uses Mixed-Integer Programming (MIP) to optimise the balance between the predictive performance, rule coverage, and rule complexity.

Our algorithm ***"Children of the Tree"*** optimises rules using MDL (Minimum Description Length) [13], which offers a different evaluation mechanism than most other algorithms in the literature. The inTrees algorithm also evaluates rules in terms of length calculation to simplify rule sets, but it does not conduct an MDL-based optimisation study. For example, while algorithms such as *CN2* and *RIPPER* usually focus on metrics such as support and confidence, our approach evaluates the complexity and accuracy of the rules together [14,15]. This distinct evaluation approach makes direct comparison with algorithms like CN2 and RIPPER challenging, as they do not incorporate complexity into their assessments. In addition, MDL-based optimisation allows the rules to be more compact and explainable. This unique feature places our algorithm in a distinct category, emphasising explainability and balance between complexity and accuracy. The fact that our algorithm can successfully work on imbalanced data sets is a significant advantage. Classical rule extraction algorithms such as *CN2* and *RIPPER* generally tend to overfit the majority class in such data sets [14,15].

On the other hand, our algorithm can effectively target the minority class in imbalanced datasets and extract meaningful rules for this class. This makes it unnecessary to compare our algorithm in the same context with others.

These studies have essentially set us a benchmark. Although they did not directly use the MDL principle in terms of rule extraction from the model, some of these studies aimed to balance model fit and complexity, which indirectly resembled the basic ideas of MDL.

Table 1 includes the basic properties, advantages, and disadvantages of mathematical programming-based algorithms in the literature that create benchmarks, and the comparison of the application domain in which they are suited.

**Table 1.** Mathematical programming-based algorithms

| Algorithm Name | Basic Properties | Advantages | Disadvantages | Application Domain/ Fields |
|---|---|---|---|---|
| Forest-ORE [12] | It extracts rule sets from RF models that can be explained by the mixed-integer optimisation. | It strikes a balance between predictive performance and explainability. | It has high computational cost and is time consuming on large datasets. | Global model explainability and minority classes. |
| MIRCO [8] | It minimizes the total rule complexity and heterogeneity using mathematical programming. | Creates rules that better represent the minority class. | The computational cost is high. | Risk analysis and data mining. |
| OptExplain [9] | It creates rules through logical inference, sampling and optimisation techniques. | Explains the broad scope of data. | Logical operations and optimisation processes can be complex. | Optimisation and logical inference. |
| defragTrees [6] | It simplifies and optimises RF rules through Bayesian model selection. | Predictive simplifies the rules while maintaining accuracy. | Bayesian processes can be slow on large datasets. | Bayesian modelling, finance, and healthcare. |

On the other hand, Table 2 includes the comparison of the basic properties, advantages, disadvantages and application domain of rule extraction algorithms that can work on machine learning models in the literature that creates benchmarks for us.

**Table 2.** Machine Learning-Based Algorithms

| Algorithm Name | Basic Properties | Advantages | Disadvantages | Application Domain/Fields |
|---|---|---|---|---|
| RuleFit [10] | It derives if-then based rules, learns rules from complex models and combines them with linear models. | L1 regularisation selects important rules and provides a balance between accuracy and explainability. | Requires SMOTE or optimisation on unbalanced datasets. | Classification and regression, financial analysis. |
| RIPPER [15] | It is if-then based and uses incremental pruning to reduce errors in rules. | Creates simple, fast, and explainable rules. | Performance may degrade on large data sets. | Medicine, finance, and small data sets. |
| CN2 [14] | It is if-then based and produces rules with an implicit (covering) algorithm. | It is powerful in unbalanced data sets and generates meaningful rules. | Accuracy may decrease in complex data sets. | Biology, medicine, classification. |
| PART [16] | It is if-then based and extracts partial rules from the decision trees. | It creates explainable and simple rules and is effective in multi-class problems. | Performance in complex relationships is limited. | Education, classification. |
| Bayesian Rule Lists (BRL) [17] | It is if-then based, sorts and optimises the rules according to the Bayesian probability model. | It offers a balance of explainability and accuracy and is robust on small datasets. | The computational cost is high for large data sets. | Healthcare/Medicine, law, finance. |
| Anchors [18] | It is if-then based, creating local rules that explain each predicted situation. | It is powerful in making sense of complex patterns and produces explanatory and intuitive rules. | Scalability may be limited to large datasets. | Model explainability and engineering. |
| C4.5 ve CART [19,20] | It is if-then based and creates rules with the decision tree algorithm. | Easy to apply, fast and explainable. | Is prone to overfitting. | Classification and regression, training. |
| Slipper [21] | It is if-then based and increases the accuracy of the rules with boosting. | It improves performance and can be effective on imbalanced datasets. | The computational cost may increase due to boosting. | Binary classification. |
| Scalable Rule-Based Learner (SRL) [22] | It is if-then based and creates scalable rules on large datasets. | It is fast, explainable and optimisable on large datasets. | May produce oversimplified results on small data sets. | Large data sets and real-time applications. |
| Interpretable Decision Sets (IDS) [23] | It is if-then based and produces non-overlapping and low-complexity rules. | Explainability is at the forefront, and the overlap between rules is minimized. | The computational cost is high for large data sets. | Healthcare, law, and sectors requiring high reliability. |

| Algorithm Name | Basic Properties | Advantages | Disadvantages | Application Domain/Fields |
|---|---|---|---|---|
| EBM (Explainable Boosting Machines) [24] | It is based on Gradient Boosting and creates explainable rules by modelling each feature independently. | Near-Gradient Boosting accuracy, meaningful explanations. | Performance in complex relationships may be limited. | Healthcare, finance, and critical decision-making processes. |
| TE2Rules [25] | It optimises the rules extracted from the tree ensemble models with a balance of fidelity and explainability. | High balance of fidelity and explainability; covers all decision paths. | It has high computational cost and is time consuming on large datasets. | Machine learning, explainable models. |
| SIRUS [4] | Creates stable and explainable rule sets; derived from RF models. | It creates short and decisive rules and provides stability. | May overlook rare but important rules. | Regression and classification, stable models. |
| inTrees [2] | It derives rules from all the decision paths in the RF and optimises these rules. | Optimises by considering the frequency of rules. | The rules are highly expressive, but complexity can increase in large data sets. | Machine learning, predictive models. |
| ExtractingRuleRF [3] | It extracts rules from the RF and weights them with the greedy algorithm. | Optimises the accuracy and coverage of the rules. | Predictive accuracy is prioritised over explainability. | Predictive performance, financial analysis. |
| RF + HC (Hill-Climbing) [5] | It uses hill-climbing to optimise the rules within the RF. | Creates small and meaningful rule sets. | The optimisation process can be lengthy. | Optimised small datasets. |
| Node Harvest [11] | It combines the rules obtained from the RF nodes with a weighted prediction model. | Creates a simple rule set with non-negative weights. | Predictive performance may be limited. | Machine learning, rule-based analysis. |
| ForEx++ [7] | Generates high-quality rule populations, improving the predictive performance. | Optimises predictive performance. | Optimised rule size may limit explainability. | Risk analysis and data mining. |
| ***Children of the Tree***[1] | It is if-then based, uses RF models, is suitable for working on imbalanced data sets, creates rules by balancing with SMOTE and ranks according to MDL cost. | It achieves high accuracy and F1 scores in both classes and finds the least cost rules. | Since computational costs can be high in numerous data sets, feature-based filtering should be added for such data sets. | Healthcare, finance, data-intensive sectors, imbalanced data sets, Machine Learning classification problems. |

[1]This article describes an algorithm for extracting rules from a new model. The algorithm details are available in Section 2. The Application results are available in Section 3.

## 1.2. Quantitative Association Rule Mining Measures in the Literature

Association rule mining is a technique frequently used in data mining to discover dependencies and patterns between elements in large data sets. It is also referred to in the literature as “interestingness metrics”. In particular, metrics such as "support", "confidence", and "lift" are among the most used metrics in association rule mining. The development of these metrics provides valuable information to the user by determining meaningful relationships between elements in the data. This approach enables the analysis of past associations to inform future studies and decision-making processes [26, 27].

*Support* and *confidence* metrics, first introduced by Agrawal and Srikant[2], form the basis of association rule mining and express the probability of the co-occurrence of elements in a dataset [26]. Later, additional metrics such as *lift* were developed to help determine the degree of dependency of the rules, expressing positive or negative dependencies. Interestingness metrics such as *the certainty factor* and *netconf* provide more meaningful results, especially by eliminating misleading or independent rules[3] [27].

## 1.3. Minimum Description Length in the Literature

The principle of Minimum Description Length (MDL) was developed through a series of papers, primarily by Jorma Rissanen [28,30,31]. Its roots lie in the Kolmogorov or algorithmic complexity theory developed by Solomonoff, Kolmogorov, and Chaitin in the 1960s[4] [32].

The Minimum Description Length (MDL) principle is a formalisation of Occam’s Razor in machine learning and statistics. In model selection, MDL seeks to balance model complexity and data adaptability [29].

### 1.3.1. Concept of MDL

MDL suggests that the best model for a dataset is the one that compresses the data most effectively. This approach consists of two main components [13,33]:

- **Model Complexity ($\mathbf{L(h)}$):** Refers to the definition length of the hypothesis or model, i.e., the number of bits necessary to represent the model. A simple model usually has a shorter definition length.
- **Data Adaptation Cost ($\mathbf{L(D \mid h)}$):** Refers to the length required to describe the data based on the model or hypothesis. A well-fitting model requires fewer bits to encode its errors or deviations from the data.

### 1.3.2. MDL Formulation

Mathematically, the total definition length L(D, h) is given by [33]:

$$L(D, h) = L(h) + L(D \mid h) \tag{1}$$

Here,

$L(h)$: It is the definition length (complexity) of the model itself,

$L(D \mid h)$: The definition length of the model relative to the data (error or redundancy encoding).

In MDL, the goal is to minimize this total definition length. This strikes a balance between model simplicity and data fidelity.

[2]See: References Section, source number 26.

[3]For a detailed analysis, please see the “Measures” section under the title “QUANTITATIVE ASSOCIATION RULES” in the 2nd Chapter of the References Section, reference number 27.

[4]For details, the source numbered 32 in the Bibliography Section can be examined.

## 1.4. SMOTE Method

SMOTE (Synthetic Minority Oversampling Technique) was introduced by Chawla et al. in 2002. It aims to enhance classification models by increasing the number of minority class samples in imbalanced datasets, enabling better predictions for the minority class. SMOTE produces synthetic data points by interpolating between an example in the minority class and one of its k-nearest neighbours, providing a wider decision boundary for the minority class. The effectiveness of SMOTE is usually evaluated by metrics such as AUC [34].

## 1.5. Introduction of the Dataset

In this study, we used the Statlog (German Credit Data)[5] dataset, a widely recognised resource in the literature for credit risk analysis. This dataset effectively captures the characteristics of bank customers and is suitable for extracting rules related to customer risk levels. This dataset is used to determine the risk level of prospective customers in the bank's marketing and risk management departments. This dataset is valuable because it combines the demographic, financial, and behavioural characteristics of the applicants. In assessing credit risk, multidimensional data such as a customer's age, employment status, past credit payments, credit period, and requested credit amount provide detailed insights into potential risks. In such imbalanced datasets, it is critical to develop models with high predictive accuracy and derive statistically significant rules to ensure effective decision-making.

Talking about the nature of the dataset is important to understand the area that the study serves. The dataset includes 20 features and 1 target variable, which are used to evaluate loan applications. These features include demographic, financial, and behavioural information about each applicant. Age indicates the age of the applicant, while Personal Status and Sex refers to the applicant's marital status and gender. Housing represents the ownership status of the applicant's residence and is classified as "own house," "rent," or "free accommodation". Number of Dependents represents how many people rely on the applicant financially [35].

Among the financial characteristics in the data set, the loan amount (Credit Amount) refers to the amount of credit requested; Duration in Months indicates the repayment period of the loan in months. The amount of deposits (Savings) refers to the amount of the applicant's savings and is categorically divided into different ranges from low to high. The Existing Property attribute classifies the type and value of properties owned by the applicant. Other Installment Plans indicate whether the applicant has additional loan agreements. The Other Debtors attribute indicates whether the applicant has a guarantor or other debtors in the loan application. Employment status and occupation (Job) is a characteristic that categorically expresses the employment status and occupation of the applicant. Employment Duration is the time worked in the current workplace, and the length of this period can be a criterion for measuring financial stability.

Among behavioural characteristics, the number of existing credits that the applicant has is an important factor in evaluating the loan application. On the other hand, the purpose of the loan indicates the purpose for which the loan is taken and is divided into categories such as "car", "furniture", "education". Credit History provides a summary of the applicant's past loan payments, and regularity in payments plays a fundamental role in understanding credit risk.

Telephone ownership indicates whether the applicant has a phone, which is particularly important for communication. Foreign Worker indicates whether the applicant is a foreign employee. It is predicted that

[5]The dataset is sourced from the UCI Machine Learning Repository; It classifies people defined by a set of characteristics as having good or bad credit risks. For details, the source numbered 35 in the Bibliography Section can be examined.

these two features may offer indirect effects to the model in evaluating a person's loan application. The Credit Risk field, on the other hand, refers to our target variable, which consists of two different classes. Class 1 refers to applications in a good (no risk) condition, while class 2 refers to applications in a poor (risky) condition.

While all these features come together in different aspects in the credit risk analysis and provide information about the loan repayment capacity of the applicant, in our study, it was estimated whether the person was risky in terms of granting loans on the classification model and it was tried to provide meaningful rules for bank prospective customers based on the situation of the customers in the bank with different rule extractions according to this risk class.

Table 3 shows the features in the dataset can be summarised in tabular form as follows.

**Table 3.** Original German Loan Dataset Specifications and Descriptions[6]

| Features Name | Data type | Demographics | Description |
|---|---|---|---|
| Checking Account | Categorical | | Existing "checking account" information |
| Duration | Numeric | | Loan Term (Term) |
| Credit History | Categorical | | Credit History |
| Purpose | Categorical | | Purpose of Obtaining Loans |
| Credit Amount | Numeric | | Loan Amount |
| Saving Account | Categorical | | Saving account information |
| Employment Duration | Categorical | Other | Length of work in the employee's current job (time interval) |
| Installment rate | Numeric | | Installment rate as a percentage of disposable income |
| Personal Status and Sex | Categorical | Marital status | Marital status and gender |
| Other Debtors | Categorical | | Other debtors/guarantors |
| Present Residence | Numeric | | Where he currently resides |
| Property | Categorical | | Properties |
| Age | Numeric | Age | Age |
| Other Installment Plans | Categorical | | Other payment plans |
| Housing | Categorical | Other | Housing |
| Number of Credits | Numeric | | Number of other loans in this bank |
| Job | Categorical | Profession | Work |
| Dependents | Numeric | | Number of people responsible for providing care |
| Telephone | Binary | | Phone |
| Foreign Worker | Binary | Other | Foreign Employee |
| Credit Risk | Binary | Target | Risk class (good/bad) |

The input of the dataset to our model is as shown in Figure 1 below before the label encoder is made.

[6] For the details of the data set, source number 35 in the Bibliography Section can be examined.

| checking | duration | credit_hi | purpose | credit_an | savings_ | employm | installme | personal_ | other_de | present_ | property | age | other_ins | housing | number_ | job | depende | telephon | foreign_v |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A11 | 6 | A34 | A43 | 1169 | A65 | A75 | 4 | A93 | A101 | 4 | A121 | 67 | A143 | A152 | 2 | A173 | 1 | A192 | A201 |
| A12 | 48 | A32 | A43 | 5951 | A61 | A73 | 2 | A92 | A101 | 2 | A121 | 22 | A143 | A152 | 1 | A173 | 1 | A191 | A201 |
| A14 | 12 | A34 | A46 | 2096 | A61 | A74 | 2 | A93 | A101 | 3 | A121 | 49 | A143 | A152 | 1 | A172 | 2 | A191 | A201 |
| A11 | 42 | A32 | A42 | 7882 | A61 | A74 | 2 | A93 | A103 | 4 | A122 | 45 | A143 | A153 | 1 | A173 | 2 | A191 | A201 |
| A11 | 24 | A33 | A40 | 4870 | A61 | A73 | 3 | A93 | A101 | 4 | A124 | 53 | A143 | A153 | 2 | A173 | 2 | A191 | A201 |

**Figure 1.** German Credit Data-Input Data

The digitisation result of the value contained in each categorical variable is shown in Table 4 below.

**Table 4.** Encoding values based on decision variables and categories

| Columns | Category | Descriptions/Values | Encoded |
|---|---|---|---|
| checking_account | A11 | X < 0 DM (Deutche Mark) | 0 |
| checking_account | A12 | 0 <= X < 200 DM | 1 |
| checking_account | A13 | X >= 200 DM/salary assignments for at least 1 year | 2 |
| checking_account | A14 | no checking account | 3 |
| credit_history | A30 | no credits taken/all credits paid back duly | 0 |
| credit_history | A31 | all credits at this bank paid back duly | 1 |
| credit_history | A32 | existing credits paid back duly till now | 2 |
| credit_history | A33 | delay in paying off in the past | 3 |
| credit_history | A34 | critical account/other credits existing (not at this bank) | 4 |
| purpose | A40 | Car (new) | 0 |
| purpose | A41 | Car (used) | 1 |
| purpose | A410 | others | 2 |
| purpose | A42 | furniture/equipment | 3 |
| purpose | A43 | radio/television | 4 |
| purpose | A44 | domestic appliances | 5 |
| purpose | A45 | repairs | 6 |
| purpose | A46 | education | 7 |
| purpose | A48 | retraining | 8 |
| purpose | A49 | business | 9 |
| savings_account | A61 | X < 100 DM | 0 |
| savings_account | A62 | 100 <= X < 500 DM | 1 |
| savings_account | A63 | 500 <= X < 1000 DM | 2 |
| savings_account | A64 | X >= 1000 DM | 3 |
| savings_account | A65 | unknown/no saving account | 4 |
| employment_duration | A71 | unemployed | 0 |
| employment_duration | A72 | X < 1 year | 1 |
| employment_duration | A73 | 1 <= X < 4 years | 2 |
| employment_duration | A74 | 4 <= X < 7 years | 3 |
| employment_duration | A75 | X >= 7 years | 4 |
| personal_status_sex | A91 | Male: divorced/separated | 0 |
| personal_status_sex | A92 | female : divorced/separated/married | 1 |
| personal_status_sex | A93 | male: single | 2 |
| personal_status_sex | A94 | Male: married/widowed | 3 |
| other_debtors | A101 | none | 0 |

| Columns | Category | Descriptions/Values | Encoded |
|---|---|---|---|
| other_debtors | A102 | co-applicant | 1 |
| other_debtors | A103 | guarantor | 2 |
| property | A121 | real estate | 0 |
| property | A122 | Building a society savings agreement/life insurance | 1 |
| property | A123 | car or other, not in attribute 6 | 2 |
| property | A124 | unknown/no property | 3 |
| other_installment_plans | A141 | bank | 0 |
| other_installment_plans | A142 | stores | 1 |
| other_installment_plans | A143 | none | 2 |
| housing | A151 | rent | 0 |
| housing | A152 | own | 1 |
| housing | A153 | for free | 2 |
| job | A171 | unemployed/ unskilled - non-resident | 0 |
| job | A172 | unskilled-resident | 1 |
| job | A173 | skilled employee/official | 2 |
| job | A174 | management/ self-employed/highly qualified employee/officer | 3 |
| telephone | A191 | none | 0 |
| telephone | A192 | yes, registered under the customer's name | 1 |
| foreign_worker | A201 | yes | 0 |
| foreign_worker | A202 | no | 1 |
| credit_risk | 1 | Good | 1 |
| credit_risk | 0 | Bad | 0 |

## 2. Children of the Tree

This article introduces a new algorithm for rule extraction from tree-based classification models such as Random Forest. In domains where rule-based predictions are crucial, such as banking, meaningful rules for prospective customers play a key role in the development of the algorithm. This algorithm is expected to provide a simple yet effective solution for the risk management and marketing departments of banks compared to existing methods in the literature.

Thanks to the "Minimum Description Length (MDL)" method, the algorithm reduces the rules that lead to unnecessary complexity and thus allows the creation of a more understandable and optimised rule set.

### 2.1. The Foundation of the *"Children of the Tree"*

A random forest model is an ensemble learning method that consists of multiple decision trees. Each decision tree classifies or predicts the $T_i$ (i = 1,2,...,N) dataset through specific rules.

Decision trees typically start with a root node (r). This root node represents the starting point of the dataset.

If a node d is a leaf node, then no distinction is made on d, and that node represents a class label or estimated value.

Leaf node metrics, M(d), are calculated after all rule extraction is complete and the significance and explainability of each rule are evaluated. These metrics include criteria such as support, confidence, and MDL (Minimum Description Length).

M(d) = f(d)

*Here, f(d) represents the metrics that indicate the significance of the rule inferred at the leaf node d.*

In our study, these metrics are support, confidence, lift, leverage, conviction and mdl values.

If the inner node is d, it can have two child nodes (dL and dR), and this node differentiates on the data using a property Xj and a threshold value θj.

The decision rule is created by using the property and threshold value on the inner node. Mathematically, the rule for the inner node d φ(d) is expressed as follows:

$$\varphi(d) = \begin{cases} X_J \leq \theta_J & \text{If d switches to the left child node} \\ X_J > \theta_J & \text{If d switches to the right child node} \end{cases}$$

*This rule splits the pieces of data that are separated from the inner node into two. The left and right child nodes represent the subsets that this rule creates.*

## 2.2. Algorithm Iterations

Below are the steps of "Children of the Tree", our rule-extraction algorithm from a tree-based machine learning model.

ALGORITHM 1: Children of the Tree Algorithm Iterations

```
current_node root
current_node_type internal
current_node is not null
while current_node is not null, do
  if current_node is leaf, do
  add rule for current_node to rule_list
else
  identify the left_child and right_child of the current_node
  split_data using feature Xj and threshold θj
  Generate rules:
    if (Xj ≤ θj) move to the left_child
    if (Xj > θj) move to the right_child
end
Move to the next node in the tree (left_child or right_child)
end
```

- **Beginning:**

The decision tree is started from the root node (r).

- **Decision Function:**

Any node d in a decision tree uses a decision function f(d) to classify the dataset. This function determines how the separation in the node and the results are obtained. In general, the decision function is defined as

$$f(d) = \begin{cases} \text{Class C1} & \text{If d is the leaf node} \\ \text{Decision } (X_j \leq \theta_j \text{ ve } d_L) \text{ veya } (X_j > \theta_J \text{ ve } d_R) & \text{If d is the inner node} \end{cases}$$

*Here, C1 is the class label on the leaf node, or it represents the predicted value. In the inner node, the decision function represents the rules that divide the dataset into two.*

- **Leaf Node Control:**

If node d is a leaf node, the extracted rule is added to the list of rules:

Rule List ← Rule List ∪ {(D)}

- **Internal Node Processing:**

If node d is the inner node:

The left (dL) and right (dR) child nodes are passed.

Using the property Xj and the threshold value θj,

Rules(d) ←{(Xj ≤ θj and dL),(Xj > θj and dR)}

*Rules are created and the child nodes are passed.*

Switching to the left and right child nodes allows similar operations to be performed on each child node. After the rules are issued, metrics such as support, confidence, lift, leverage, and conviction are calculated.

The MDL (Minimum Description Length) value is calculated by evaluating the complexity and error costs of each rule. These metrics are evaluated after the entire rule extraction process is completed and the optimal rules are selected.

### 2.3. Evaluation of the Rules Metrics

*"Children of the Tree"* uses support, confidence, lift, leverage, and conviction metrics to determine how meaningful a rule or association is in the dataset and how interesting it is.

**Support:** Indicates how often the rule is passed in the dataset.

Support = $\frac{Nrule}{Ntotal}$

*Nrule: The number of instances to which the rule applies.*

*Ntotal: Total number of instances.*

**Confidence:** Confidence measures the probability that a rule is true. It refers to how often the rule is true, especially when given a property or condition.

Confidence = $\frac{Ncorrect}{Nrule}$

*Ncorrect: The number of instances where the rule is true.*

*Nrule: The number of instances to which the rule applies.*

**Lift**: Measures how well the rule is relative to the expected accuracy.

The upgrade shows how good the rule is compared to the expected accuracy rate.

Lift = $\frac{Confidence}{\frac{Nclass}{Ntotal}}$

*Confidence: The confidence value of the rule.*

$\frac{Nclass}{Ntotal}$ *: The proportion of the class that exists as a result of the rule in the dataset.*

**Leverage**: Subtracts the support value of the rule from its expected support in the case of independence.

Leverage = $Support - \frac{Nantecedent}{Ntotal} x \frac{Nclass}{Ntotal}$

*Support: The support value of the rule.*

$\frac{Nantecedent}{Ntotal}$*: The expected accuracy of the feature (or condition).*

$\frac{Nclass}{Ntotal}$ *: The proportion of the class that exists as a result of the rule in the dataset.*

**Conviction**: It measures how persuasive the rule is to ensure its accuracy.

Conviction = $\frac{1-\frac{Nclass}{Ntotal}}{1-\ Confidence}$

*Confidence: The confidence value of the rule.*

$\frac{Nclass}{Ntotal}$ *: The* proportion of the class that exists as a result of the rule in the dataset.

## 2.4. Application of MDL in Rule Extraction

In the context of rule extraction, MDL can be used to evaluate the "cost" of a set of rules. Section 1.3 “Minimum Description Length in Literature” describes the way MDL is calculated in the literature. Accordingly, a rule that is too complex (with too many conditions) will have a high L(h) value, while a rule that adapts poorly to the data will have a high L(D|h) value. The best rule set would be the one that minimizes this unified definition length. Our algorithm implements MDL calculations based on the principles outlined in the literature.

1. Sample Calculation for MDL Cost

In our rule extraction algorithm, the MDL value is calculated as follows, alongside other metrics commonly used in the literature:

Rule Complexity (L(h)): If a rule r has k conditions, each condition contributes to the complexity. For example, if we assume that each condition requires a certain number of bits, the total complexity of the rule can be expressed approximately as follows:

$$L(h) = k \cdot \log 2(N) \tag{2}$$

*Here, N refers to the number of samples in the dataset and can be thought of as a "resolution" that determines the complexity of the rule.*

Cost of Error (L(D | h)): This represents the number of samples that were misclassified or incorrectly estimated by the rule. Let E be the number of misclassifications:

$$L(D \mid h) = E \cdot log2(N) \tag{3}$$

Total MDL Cost: By combining the two components, the MDL cost of an r rule can be written as:

$$MDL(r) = k \cdot log2(N) + E \cdot log2(N) \tag{4}$$

*This cost function promotes simple rules (low* k*) and correct rules (low* E*).*

MDL's Interpretation

- A low MDL value indicates a good balance between model complexity and accuracy.
- A high MDL value indicates that the rule is either too complex (high k) or contains too many errors (high E), which makes it less preferable.

The interpretation of low or high MDL values should be considered relative to all other calculated MDL values in the dataset.

## 3. Application

This section presents the application of our algorithm. The functionality of *"Children of the Tree"* is explained in detail in Chapter 2. In this section, the *"Children of the Tree"* algorithm was applied on the dataset introduced in Section 1.5 “Introduction of the Dataset” and the results were obtained as shown in Table 5.

**Table 5.** Children of the Tree Algorithm Results: Rules, Metrics, and Related Classes

| Rule | Class | Support | Confidence | Lift | Leverage | Conviction | MDL Cost |
|---|---|---|---|---|---|---|---|
| duration > 6.50 and savings_account <= 2.50 and checking_account <= 2.50 and other_debtors <= 1.50 and credit_history <= 3.50. | 2 | 0,48 | 0,78 | 1,56 | 0,24 | 2,28 | 9176,16 |
| savings_account <= 2.50 and other_debtors <= 1.50 and checking_account <= 2.50 and credit_history <= 3.50 and duration > 10.50. | 2 | 0,45 | 0,79 | 1,59 | 0,22 | 2,42 | 9479,25 |
| savings_account <= 1.50 and credit_history <= 3.50 and checking_account <= 2.50 and other_debtors <= 1.50 and personal_status_sex <= 2.50. | 2 | 0,49 | 0,79 | 1,57 | 0,22 | 2,34 | 9521,05 |
| present_residence <= 3.50 and checking_account <= 2.50 and foreign_worker <= 0.50 and other_debtors <= 1.50 and purpose <= 8.50. | 2 | 0,43 | 0,76 | 1,52 | 0,22 | 2,07 | 9865,94 |
| checking_account <= 2.50 and other_debtors <= 1.50 and credit_history <= 3.50 and duration > 12.50 and savings_account <= 3.50. | 2 | 0,39 | 0,80 | 1,62 | 0,19 | 2,62 | 10116,77 |
| checking_account <= 2.50 and other_debtors <= 1.50 and credit_amount > 1044.50 and number_credits <= 1.50 and credit_amount > 1173.00. | 2 | 0,43 | 0,70 | 1,41 | 0,22 | 1,70 | 10189,93 |
| credit_history <= 3.50 and checking_account <= 2.50 and duration > 7.50 and credit_history > 1.50 and other_debtors <= 1.50. | 2 | 0,43 | 0,70 | 1,39 | 0,21 | 1,65 | 10325,80 |
| credit_amount <= 10918.00 and telephone <= 0.50 and checking_account <= 2.50 and employment_duration <= 3.50 and dependents <= 1.50 | 2 | 0,40 | 0,74 | 1,48 | 0,20 | 1,92 | 10398,96 |
| checking_account <= 2.50 and duration > 10.50 and | 2 | 0,37 | 0,78 | 1,56 | 0,19 | 2,29 | 10409,41 |

| Rule | Class | Support | Confidence | Lift | Leverage | Conviction | MDL Cost |
|---|---|---|---|---|---|---|---|
| other_debtors <= 1.50 and employment_duration <= 3.50 and telephone <= 0.50. | | | | | | | |
| employment_duration > 1.50 and purpose <= 8.50 and duration <= 25.50 and credit_history > 1.50 and credit_amount <= 7452.00. | 1 | 0,40 | 0,69 | 1,38 | 0,20 | 1,61 | 10628,88 |
| duration <= 27.50 and housing > 0.50 and employment_duration > 1.50 and credit_history > 1.50 and other_installment_plans > 1.50. | 1 | 0,29 | 0,78 | 1,55 | 0,15 | 2,24 | 11339,56 |
| duration <= 24.50 and savings_account <= 3.50 and employment_duration > 1.50 and credit_amount <= 7881.00 and duration > 7.00. | 1 | 0,39 | 0,59 | 1,17 | 0,19 | 1,21 | 11381,37 |
| other_installment_plans > 1.50 and employment_duration > 1.50 and credit_history > 1.50 and credit_amount <= 3897.50 and housing > 0.50. | 1 | 0,28 | 0,79 | 1,58 | 0,14 | 2,37 | 11444,08 |
| personal_status_sex > 1.50 and duration > 8.50 and duration <= 24.50 and credit_amount > 1081.00 and credit_amount <= 7521.00. | 1 | 0,3 | 0,66 | 1,33 | 0,15 | 1,49 | 11768,06 |
| credit_history > 1.50 and foreign_worker <= 0.50 and other_installment_plans > 1.50 and credit_amount <= 3916.00 and number_credits <= 1.50. | 1 | 0,33 | 0,59 | 1,18 | 0,17 | 1,23 | 11788,97 |
| savings_account <= 2.50 and duration <= 24.50 and employment_duration > 1.50 and duration > 7.50 and credit_history > 1.50 | 1 | 0,31 | 0,63 | 1,27 | 0,15 | 1,37 | 11830,77 |
| checking_account > 2.50 and age > 24.50 and duration <= 45.00 and credit_amount <= 9569.00 and other_installment_plans > 1.50. | 1 | 0,20 | 0,93 | 1,86 | 0,10 | 7,13 | 11914,38 |
| other_installment_plans > 1.50 and personal_status_sex | 1 | 0,27 | 0,70 | 1,40 | 0,13 | 1,66 | 11966,64 |

| Rule | Class | Support | Confidence | Lift | Leverage | Conviction | MDL Cost |
|---|---|---|---|---|---|---|---|
| > 1.50 and credit_history > 1.50 and property <= 2.50 and savings_account <= 3.50. | | | | | | | |

The *"Children of the Tree"* algorithm was produced with the original ideas of the authors, since no similar solution was found in the literature. In particular, the fact that the MDL interpretation is used in the step of rule extraction from machine learning models makes our algorithm valuable in terms of making a unique contribution to the literature.

The *"Children of the Tree"* algorithm is an optimisation algorithm for extracting rules from the machine learning model, based on Random Forest. In the study, the SMOTE technique was used to make the algorithm work on unbalanced data sets.

The study focuses on two main components. First, the imbalanced dataset was balanced using the SMOTE technique. Various hyperparameter optimizations were applied to the Random Forest model, achieving an overall F1 score of approximately 0.81. The F1 scores were 0.80 for Class 1 (good class) and 0.81 for Class 2 (bad class). Based on these results, the rule extraction mechanism detailed in Section 2 of the "Children of the Tree" algorithm was implemented on the resulting model. After extracting the rules, metrics such as support, confidence, lift, leverage, and conviction were calculated. The MDL value was determined by assessing the complexity and error costs of each rule. After that, all these metrics were evaluated. The optimal rules were selected. Table 5 presents the optimal rules, the classes they represent, and the corresponding metric values, including validity and prevalence. Additionally, the table highlights the MDL cost values, reflecting the simplicity of these rules.

In Table 5 above, our *"Children of the Tree"* algorithm is applied to "German Credit Data" to extract rules for potential customers through bank customers. The first column lists the extracted rules, while the second column indicates the classes to which the rules belong. In other words, if a rule applies to the bank's prospective customer, it indicates whether the customer is potentially risky or has a low risk level. The support values in the table indicate the frequency of each rule in the dataset, while the confidence values represent their validity. As is known, the MDL Cost field is a reflection of the mathematical calculation of the simplicity value of the rule in the table.

MDL costs are problem-specific and may vary significantly across different datasets and studies, sometimes being higher or lower than the values observed here. However, the lowest of the costs given in this study was calculated for class 2 and this cost belongs to the rule "duration > 6.50 and savings_account <= 2.50 and checking_account <= 2.50 and other_debtors <= 1.50 and credit_history <= 3.50" with a value of 9176.16[7] . The rule with the lowest MDL cost for Class 1 is "employment_duration > 1.50 and purpose <= 8.50 and duration <= 25.50 and credit_history > 1.50 and credit_amount <= 7452.00" with an MDL Cost value of 10628.88[8].

[7] The decision limit determined for each variable in these rules can be read with the numerical equivalents of the values described in Section 1.5 Introduction to the Dataset in Table 4 Table of Encoding Values Based on Decision Variables and Categories.

[8] Example rule reading: Decision class 1 for loans of 25.50 months or less when the employment_duration is A73, A74 or A75, and for those with a credit history of less than 7452, the credit history is A32, A33 or A34. See Section 1.5 for a variable-based explanation of each categorical statement.

## 4. Conclusion of the Article and Future Works

In our study, the MDL principle is used to enhance the explainability and simplicity of the rules. MDL is useful in minimizing the definition length of the rules by considering the complexity of the rule (number of conditions) and the cost of error (number of misclassifications). This encourages rules that are not only true but also relatively less complex and explainable with fewer conditions. Our study introduces a new and effective algorithm that prioritises less complex rules, evaluates simplicity using MDL cost, and generates more general rules by accounting for the minority class in imbalanced datasets. This approach distinguishes it from alternative methods in rule optimisation.

In the future, we aim to enhance the algorithm's ability to generate more stable rules, similar to its competitors, while producing more general rules with higher accuracy and simplicity. Additionally, calculations can be performed based on the extraction and importance of the rules for the selected features. This will enable the generation of more focused rules for specific classes. In future studies, it is aimed to enable users to identify the most important features and obtain rules focused on them.

Below, Table 6 presents a comparison of our algorithm with RuleFit, as discussed in Section 1.1 "Rule Extraction in Literature" in terms of model performance. The results for RuleFit were obtained by applying it within its framework, with the tree_generator parameter set to random_forest_classifier. The same preprocessing steps were performed for both algorithms, and the same features were used as input to the models. Additionally, both datasets were balanced using the SMOTE technique and both random forest models were configured with the same hyperparameter values.

**Table 6.** Comparison of Model Performances of RuleFit and Children of the Tree

| Algorithm | Accuracy | F1 Score for Class 2 | F1 Score for Class 1 |
|---|---|---|---|
| Rule Fit | 0.74 | 0.75 | 0.73 |
| Children of the Tree | 0.80 | 0.81 | 0.80 |

The results for "Children of the Tree" stand out against RuleFit for Class 2 (bad) and Class 1 (good). The superiority of our algorithm is evident from Table 6, as it enables the model to make more reliable predictions and better distinguish between classes, which contributes to the extraction of more trustworthy rules being extracted. The reason for comparing model performance is that RuleFit, like our algorithm, can train on a random forest model within its framework. Therefore, comparing the performance of the two algorithms on datasets processed with identical preprocessing steps and feature selection allows for a fairer evaluation of their classification success.

Let us continue our comparison with another algorithm, Anchors, and this time analyse the rules extracted by the Anchors algorithm when it is applied to the same random forest model used with Children of the Tree. Table 7 presents some rules extracted by Anchors. Anchors evaluate the rules it extracts based on precision and coverage. Precision reflects the accuracy of a rule, while coverage represents the proportion of the dataset to which the rule applies.

**Table 7.** Anchor Results Based on the Highest Precision Values

| Class | Anchors | Precision | Coverage |
|---|---|---|---|
| 2 | checking_account = 0 AND personal_status_sex = 1 AND duration > 28.00 | 1.0 | 0.04 |
| 2 | checking_account = 1 AND credit_amount > 4176.25 AND telephone = 0 AND property = 2 AND other_installment_plans = 2 | 1.0 | 0.01 |

| Class | Anchors | Precision | Coverage |
|---|---|---|---|
| 1 | employment_duration = 4 AND duration <= 19.00 AND age > 39.25 AND other_installment_plans = 2 AND 1361.75 < credit_amount <= 2319.50 | 1.0 | 0.01 |
| 1 | duration <= 12.00 AND present_residence > 3.00 AND housing = 1 AND property = 0 AND credit_amount <= 2319.50 | 0.99 | 0.03 |

In Table 7, rules with these rules with high precision apply to negligible subsets of the data. In contrast, the Children of the Tree algorithm evaluates rules across broader metrics. While they may not be fully comparable, the coverage metric in Anchors, which indicates how much of the dataset a rule covers, can be compared to the support values of Children of the Tree. Similarly, Anchors' precision, which indicates how often a rule is correct, can be compared to the confidence values of Children of the Tree.

When analysed by class, the highest support and confidence values observed in Children of the Tree were 0.48 and 0.78 for Class 2 and 0.40 and 0.69 for Class 1, respectively. For Anchors, when we examine a rule for Class 1 that is comparable in length to the rules extracted by Children of the Tree, specifically the rule "employment_duration = 4 AND duration <= 19.00 AND age > 39.25 AND other_installment_plans = 2 AND 1361.75 < credit_amount <= 2319.50," its precision is 1, but its coverage is only 0.0104. This indicates that its applicability across the dataset is extremely low. Similarly, for Class 2, the rule in Anchors that is most similar in length and has the best combination of precision and coverage, "checking_account = 1 AND credit_amount > 4176.25 AND telephone = 0 AND property = 2 AND other_installment_plans = 2," has a precision of 1 but a coverage of just 0.0147.

Table 8 presents the results of our Children of the Tree algorithm, ranked by the highest confidence values. When compared to Table 7 above, if we consider confidence as analogous to precision and support as analogous to coverage, it is clear that the Children of the Tree algorithm stands out compared to Anchors, especially when evaluated based on similar rules.

**Table 8.** Children of the Tree Algorithm's Results Ranked by the Highest Confidence

| Rule | Class | Confidence | Support |
|---|---|---|---|
| age > 33.50 and purpose <= 4.50 and credit_amount <= 4814.00 and checking_account > 2.50 and duration <= 16.50 | 1 | 1.0 | 0.05 |
| checking_account > 2.50 and duration <= 16.50 and other_installment_plans > 1.50 and credit_history > 3.50 | 1 | 1.0 | 0.04 |
| housing <= 0.50 and checking_account <= 2.50 and present_residence <= 3.50 and dependents <= 1.50 and other_installment_plans <= 1.50 | 2 | 1.0 | 0.04 |
| credit_history <= 3.50 and personal_status_sex <= 1.50 and credit_history <= 1.50 and savings_account <= 1.50 and installment_rate > 2.50 | 2 | 1.0 | 0.04 |

The comparison of the Children of the Tree algorithm with both Anchors and RuleFit highlights the strengths of the proposed approach. While Anchors provides highly precise rules, these rules often apply to extremely small subsets of the dataset, limiting their practical utility. On the other hand, RuleFit exhibited lower model performance under the same hyperparameters and preprocessing steps, resulting in reduced classification success compared to Children of the Tree. The Children of the Tree algorithm effectively balances confidence (analogous to precision) and support (analogous to coverage) while achieving higher F1 scores for both classes. This makes Children of the Tree a superior alternative, particularly for unbalanced datasets, where generating meaningful and scalable rules is critical for decision-making processes, such as credit risk analysis. By leveraging metrics such as MDL cost and support, Children of the Tree demonstrates its ability to produce rules that are both interpretable and practical, distinguishing itself as a robust and effective alternative to existing algorithms like Anchors and RuleFit.

## 4.1. Limitations of the Study

The "Children of the Tree" algorithm presents notable contributions to rule extraction, yet certain limitations remain that can guide future enhancements. While the algorithm demonstrates strong performance on the German Credit Data dataset, its generalizability to datasets with different characteristics has not been fully explored. Additionally, the current approach lacks the ability to prioritise rules based on user-specified features or domain knowledge, which could increase its applicability in real-world scenarios. Finally, although comparisons were made with Anchors and RuleFit, future work evaluating the algorithm against other state-of-the-art rule extraction and explainable AI techniques would provide deeper insights into its relative advantages and areas for improvement.

Peer Review: Externally peer-reviewed.

Author Contributions: Conception/Design of Study- H.M., M.B.; Data Acquisition – H.M.; Data Analysis/Interpretation- H.M., M.B.; Drafting Manuscript- H.M., M.B.; Critical Revision of Manuscript- M.B.; Final Approval and Accountability- H.M., M.B.; Technical or Material Support- H.M., M.B.; Supervision- M.B.

Acknowledgements: We acknowledge GTech for their support, as the preliminary version of this study was used in the “AI-Powered Decision Support and Early Warning System in the Finance and Banking Sector” project. This project provided the foundation and inspiration for the development of the current study. GTech’s domain expertise and contributions in machine learning applications enriched the study by guiding feature selection.

Conflict of Interest: The authors have no conflict of interest to declare

Grant Support: The authors declared that this study has received no financial support.

Author Details

Hilal Meydan

[1] Yıldız Technical University, Department of Mathematical Engineering, İstanbul, Türkiye

0009-0000-6145-4418

Mert Bal

[1] Yıldız Technical University, Department of Mathematical Engineering, İstanbul, Türkiye

0000-0001-6250-929X

# References

[1] Leo Breiman. 2001. Random Forests. Machine Learning. 45, 1 (Oct. 2001), 5-32.https://doi.org/10.1023/A:1010933404324.

[2] Houtao Deng. 2019. Interpreting Tree Ensembles with inTrees. International Journal of Data Science and Analytics. 7, 4 (Dec. 2019), 277–287. https://doi.org/10.1007/s41060-018-0144-8.

[3] Kim Phung Lu Thi, Ngoc Chau Vo Thi, and Nguyen Hua Phung. 2015. Extracting Rule RF in Educational Data Classification: From a Random Forest to Interpretable Refined Rules. In Proceedings of the 2015 International Conference on Advanced Computing and Applications (ACOMP). 20–27. https://doi.org/10.1109/ACOMP.2015.13.

[4] Clément Bénard, Gérard Biau, Sébastien da Veiga, and Erwan Scornet. 2019. SIRUS: Stable and Interpretable Rule Set for Classification. arXiv:1908.06852. Retrieved from https://arxiv.org/abs/1908.06852.

[5] Morteza Mashayekhi and Robin Gras. 2015. Rule Extraction from Random Forest: The C Methods. In Kanade, T., Kittler, J., Kleinberg, J.M., et al. (Eds.). Advances in Artificial Intelligence. Vol. 3060. Springer Berlin Heidelberg, Berlin, Heidelberg, 223–237. 10.1007/978-3-319-18356-5_20

[6] Satoshi Hara and Kohei Hayashi. 2017. Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach.arXiv:1606.09066. Retrieved from https://arxiv.org/abs/1606.09066.

[7] Md Nasim Adnan and Md Zahidul Islam. 2017. For A New Framework for Knowledge Discovery from Decision Forests. Australasian Journal of Information Systems. 21, (Nov. 2017). https://doi.org/10.3127/ajis.v21i0.1539.

[8] S. Ilker Birbil, Mert Edali, and Birol Yuceoglu. 2020. Rule Covering for Interpretation and Boosting. arXiv:2007.06379.Retrieved from https://arxiv.org/abs/2007.06379.

[9] Gelin Zhang, Zhe Hou, Yanhong Huang, Jianqi Shi, Hadrien Bride, Jin Song Dong, and Yongsheng Gao. 2021. Extracting Optimal Explanations for Ensemble Trees via Logical Reasoning. arXiv:2103.02191. Retrieved from https://arxiv.org/abs/2103.02191.

[10] Jerome H. Friedman and Bogdan E. Popescu. 2008. Predictive Learning via Rule Ensembles. The Annals of Applied Statistics. 2, 3 (September 2008), 916–954. https://doi.org/10.1214/07-AOAS148.

[11] Nicolai Meinshausen. 2010. Node Harvest. The Annals of Applied Statistics. 4, 4 (December 2010), 2049–2072. DOI:https://doi.org/10.1214/10-AOAS367. arXiv:0910.2145. Retrieved from https://arxiv.org/abs/0910.2145.

[12] Haddouchi Maissae and Berrado Abdelaziz. 2024. Forest-ORE: Mining Optimal Rule Ensemble to Interpret Random Forest Models. arXiv:2403.17588. Retrieved from https://doi.org/10.48550/arXiv.2403.17588.

[13] Peter D. Grünwald. 2007. The Minimum Description Length Principle. Adaptive Computation and Machine Learning series. The MIT Press. https://doi.org/10.7551/mitpress/4643.001.0001.

[14] Peter Clark and Tim Niblett. 1989. The CN2 Induction Algorithm. Machine Learning. 3, (1989), 261–283. https://doi.org/10.1007/BF00116835.

[15] Mlungisi Duma, Bhekisipho Twala, and Tshilidzi Marwala. Improving the Performance of the RIPPER in Insurance Risk Classification: A Comparative Study Using Feature Selection. arXiv:1108.4551. Retrieved from https://doi.org/10.48550/arXiv.1108.4551.

[16] Eibe Frank and Ian H. Witten. 1998. Generating Accurate Rule Sets Without Global Optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*. 144–151. Published: 24 July 1998.

[17] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. 2015. Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model. The Annals of Applied Statistics. 9, 3 (2015), 1350–1371. https://doi.org/10.1214/15-AOAS848.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: high-precision model-agnostic explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 187, 1527–1535.

[19] Steven L. Salzberg. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Mach Learn 16, 235–240 (1994). https://doi.org/10.1007/BF00993309.

[20] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. 1984. Classification and regression trees. Wadsworth International Group. https://doi.org/10.1201/9781315139470.

[21] William W. Cohen and Yoram Singer. 1999. A simple, fast, and effective rule learner. In Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence (AAAI '99/IAAI '99). American Association for Artificial Intelligence, USA, 335–342.

[22] Zhuo Wang, Wei Zhang, Ning Liu, Jianyong Wang. 2021. Scalable Rule-Based Representation Learning for Interpretable Classification. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[23] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. 2016. Interpretable Decision Sets: A Joint Framework for Description and Prediction. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 1675–1684. https://doi.org/10.1145/2939672.2939874

[24] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. Interpretml: A Unified Framework for Machine Learning Interpretability. arXiv:1909.09223. Retrieved from https://doi.org/10.48550/arXiv.1909.09223.

[25] G Roshan Lal, Xiaotong Chen, and Varun Mithal. 2022. TE2Rules: Explaining Tree Ensembles using Rules. arXiv:2206.14359. Retrieved from https://doi.org/10.48550/arXiv.2206.14359.

[26] Agrawal, R., Imieliński, T., and Swami, A. 1993. Mining Association Rules Between Sets of Items in Large Databases. *ACM SIGMOD Record*. 22, 2 (June 1993), 207–216. https://doi.org/10.1145/170036.170072.

[27] Elif VAROL ALTAY and BİLAL ALATAŞ. 2020. Nicel Birliktelik Kural Madenciliği İçin Baskın Olmayan Sıralama Genetik Algoritma-II'nin Duyarlılık Analizi. BİLİŞİM TEKNOLOJİLERİ DERGİSİ, Cilt 13, (Ocak 2020).

[28] Jorma Rissanen, Modeling by the shortest data description, Automatica 14 (1978) 465–471.

[29] Teemu Roos. 2017. Minimum Description Length Principle. In Encyclopedia of Machine Learning and Data Mining(editors:Sammut, C., Webb, G.I.). Springer, 823–827. https://doi.org/10.1007/978-1-4899-7687-1_894.

[30] Jorma Rissanen. 1989. Stochastic Complexity and Statistical Inquiry. World Scientific.

[31] Andrew R. Barron, Jorma Rissanen and Bin Yu. 1998. The minimum description length principle in coding and modeling. IEEE Trans. Inform. Theory 44(6), 2743–2760. https://doi.org/10.1109/18.720554.

[32] Ming Li and Paul Vitányi. 2019. An Introduction to Kolmogorov Complexity and Its Applications. Springer. ISBN: 978-3-030-11297-4.

[33] Peter Grünwald and Teemu Roos. 2019. Minimum Description Length Revisited. *International Journal of Mathematics for Industry*. 11, 01 (2019), 1930001. https://doi.org/10.1142/S2661335219300018.

[34] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research 16* (2002) 321–357. https://doi.org/10.48550/arXiv.1106.1813

[35] Hans Hofmann. 1994. Statlog (German Credit Data) [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5NC77.