Decision Tree Induction Through Meta-learning

Caique Augusto Ferreira, Adriano Henrique Cantão, José Augusto Baranauskas

▶ **To cite this version:**

Caique Augusto Ferreira, Adriano Henrique Cantão, José Augusto Baranauskas. Decision Tree Induction Through Meta-learning. 18th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2022, Hersonissos, Greece. pp.101-111, 10.1007/978-3-031-08337-2_9 . hal-04668637

HAL Id: hal-04668637
https://inria.hal.science/hal-04668637v1

Submitted on 7 Aug 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Decision Tree Induction through Meta-Learning

Caique Augusto Ferreira[1 [0000−0001−9871−7832]], Adriano Henrique Cantão[1 [0000−0003−0939−4870]], and José Augusto Baranauskas[1 [0000−0002−7501−7187]]

Department of Computer Science and Mathematics, Faculty of Philosophy, Sciences and Letters at Ribeirao Preto, University of Sao Paulo. Bandeirantes Avenue, 3900, 14040-901, Ribeirao Preto, SP, Brazil
{caiqueaugustoferreira, cantao, augusto}@usp.br
https://www.usp.br/

Abstract. Symbolic or explainable learning models stand out within the Machine Learning area because they are self-explanatory, making the decision process easier to be interpreted by humans. However, these models are overly responsive to the training set used. Thus, even tiny variations in training sets can result in much worse precision. In this research we propose a meta-learning approach that transforms a Random Forest into a single Decision Tree. Experiments were performed on classification datasets from different domains. Our approach using precision (positive reliability) performs as good as a Random Forest with no statistically significant differences. Yet, its advantage is the interpretability provived by a single decision tree. Results indicate that it is possible to obtain a resulting model which is easier to interpret than a Random Forest, still with higher precision than a standard Decision Tree.

Keywords: Meta-Learning · Model Combination · Random Forest · Meta-Decision Tree.

1 Introduction

Machine Learning (ML) algorithms can be categorized as symbolic and non-symbolic. The symbolic category, also known as interpretable or Explanatory Artificial Intelligence (XAI), is characterized by representations of knowledge that can be easily interpreted by humans on a scale of understanding that can range from the common sense level to the expert level. The non-symbolic category, also known as the black-box, is characterized by representations that are not easily interpreted by humans. For this category, the algorithm develops its own knowledge representation, which generally does not provide any clarification, thus making it difficult to understand [8]. Symbolic learning algorithms contribute a lot to the understanding of induced knowledge (model) needed in many applications [12].

For non-critical applications such as movie, product and digital content recommendations, not understanding how the model achieves the result does not offer significant risks or impacts if the result obtained is not good. However,

there are areas, such as medicine and healthcare, where the impact of a wrong prediction can cause great harm [10, 17]. Even though the resulting model is used only to support the decision process, the fact of not understanding how the result was obtained is a factor that makes its use unfeasible [23]. The spread of Machine Learning has also led to the emergence of new regulatory laws to control its use towards XAI. For instance, the European Union created the General Data Protection Regulation (GDPR) and its Article 22 defines the right of explanation, and guarantees that anyone affected by the decision of an algorithm has the right to know how that decision was made [6]. Interpretable models are important for human experts and to ensure the model work as expected [4].

The machine learning literature is recently trying to produce interpretable models from black-box models, with new algorithms emerging [20]. LIME (Local Interpretable Model-Agnostic Explanations) is an agnostic and locally linear; it finds a linear model in the neighborhood of the instance to be explained using black-box model decision boundaries [24]. LORE (Local Rule-based Explanations), also an agnostic and local algorithm, tries to explain the decision of a black-box for a given instance by generating a symbolic surrogate model (a Decision Tree) [13]. A Decision Tree is an inherently interpretable model. Each path from the root to a leaf of the decision tree can be easily converted into a rule. Detailed surveys on the explainability of models can be found in [3, 14, 19].

However, symbolic models are generally less accurate than non-symbolic ones.The ensemble model combination strategy is an alternative to improve the precision and stability of models [2, 7, 22, 1]. Although ensembles, in general, improve individual model precision, for symbolic algorithms, the resulting ensemble model is not symbolic anymore: even considering that each individual model is interpretable, the process of interpreting the resulting ensemble model becomes humanly difficult or infeasible, even for domain experts.

Combining a set of models resulting from the ensemble strategy into a single model is an alternative to minimize the difficulty of interpretation. In this study, we present an algorithm to combine decision trees generated by the Random Forest algorithm into a single decision tree using meta-learning.

The remaining of this work is organized as follows: In Sect. 2 we describe our methodological approach to generate a Meta Decision Tree from a Random Forest. Sect. 3 shows the empirical setup used to evaluate the proposed algorithm; Sect. 4 shows the experiments and discusses the results; finally, Sect. 5 shows the leaf weighting metrics with better performance of this study, the main contributions and some possible approaches for future work.

2 A Meta Decision Tree Algorithm

The proposed meta-learning approach is represented by Algorithms 1 and 2, where:

– Each attribute in the original set of instances corresponds to a column in the decision table.

- Each leaf in the decision tree corresponds to a single decision table row.
- In the representation of a leaf, the decision table columns assume the values contained in the branches of the subtrees.
- For attributes that are not in the leaf representation, the respective columns assume the value '?', representing the absence of a value or that the test on this attribute is unnecessary.
- A direct way to represent the weight of each leaf is to use the number of instances that reached the respective leaf. However, in this work, were used the metrics described in Sect. 2.1 as leaf weight.
  - If the tree is a single leaf, then the table contains a single row. In the decision table, the attribute columns assume the value '?', the class column assumes the same class as the leaf.
  - If the tree has multiple leaves, for each leaf, a row is generated in the decision table based on the subtree branches from the root to the respective leaf.

According to Algorithm 1, initially, a Random Forest is generated containing decision trees based on the set of instances D (line 2). The number of decision trees was set to $\mathcal{L} = 128$ [21].

Then, each decision tree in the Random Forest is transformed into a decision table (lines 4–7). The Algorithm 2 is responsible for this transformation, it identifies each leaf contained in the decision tree, considering the following information: the subtree branches from the root to the leaf, the class and the number of instances contained in the respective leaf (lines 4–12). For each leaf identified in the decision tree, a new row r is created to store this information (lines 6 - 10). Thus r represents a row in the resulting decision table R (line 11). In this way, each leaf contained in the decision tree corresponds to a single row in the resulting decision table.

Considering that all $\mathcal{L}$ decision trees contained in the Random Forest were converted to decision tables and merged into Z, the `Table2Instances` method (line 8 of the Algorithm 1) transforms Z into a new training set D_{new}, which will be used as input to the Meta Decision Tree inducer. In this step, the meta-learning happens, where the learning acquired by the decision trees contained in the Random Forest is used as a training set for the induction of a Meta Decision Tree, transforming all decision trees into a single decision tree. The `Table2Instances` method is responsible for formatting the rows contained in the decision table Z to the format expected by the algorithm used to induce the Meta Decision Tree.

2.1 Leaf-weighting Metrics

Each path in the decision tree from the root to a leaf corresponds to a rule, which can be seen as having two components $L \rightarrow R$, where L represents the conditions (attribute tests) until reaching that leaf, and R is the class present in that leaf.

Algorithm 1 Meta Decision Tree Induction algorithm

Input: D (set of n classified instances $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, n\}$ containing m attributes $\{X_1, X_2, \ldots, X_m\}$), $\mathcal{L}$ (number of decision trees in the Random Forest, where $\mathcal{L} = 128$)
Output: Meta Decision Tree Model

function MetaDecisionTreeInduction(D, $\mathcal{L}$)
F = buildRandomForest(D, $\mathcal{L}$)
$Z = []$
for $i \in \{1, 2, \ldots, \mathcal{L}\}$ **do**
  t = Tree2Table(F_i, $\mathcal{L}$, n)
  $Z = Z \cup t$
end for
D_{new} = Table2Instances(Z)
T = buildDecisionTree(D_{new})
return T

Algorithm 2 Tree to Table algorithm

Input: T (decision tree), n (number of instances), $\mathcal{L}$ (number of decision trees in the Random Forest)
Output: Decision Table

function Tree2Table(T, $\mathcal{L}$)
$R = []$
$W = \sum_i w_i$
for each leaf $l \in T$ with class C and weight w **do**
  $r = []$ {Structure of values $[X_1, \ldots, X_m, Y, \text{Weight}]$}
  for each attribute X_j with threshold O_j from root to leaf l in T **do**
    $r[X_j] = O_j$
  end for
  $r[Y] = C$
  $r[\text{Weight}] = \frac{nw}{W\mathcal{L}}$
  $R = R \cup r$
end for
return R

From this point of view, it is possible to define the corresponding contingency matrix of a leaf (or a rule), shown in Table 1 [18]. In this table, L denotes the set of instances for which the rule condition is true (instances is covered by the rule) and its complement $\bar{L}$ denotes the set of examples for which the rule condition is false (instances is not covered by the rule), and analogously for R and $\bar{R}$. LR denotes the set of instances $L \cap R$ in which L and R are both true (the rule correctly classifies the instances), $L\bar{R}$ denotes the set of instances $L \cap \bar{R}$ where L is true and R is false (the rule misclassifies instances) and so on.

The cardinality of a set A is denoted as $a = |A|$. Thus, l denotes the number of instances in the set L, that is, $l = |L|$, r denotes the number of instances in

Table 1. Contingency matrix for a leaf (rule) $L \rightarrow R$

	L	$\bar{L}$	
R	lr	$\bar{l}r$	r
$\bar{R}$	$l\bar{r}$	$\bar{l}\bar{r}$	$\bar{r}$
	l	$\bar{l}$	n

the set R, that is, $r = |R|$, lr denotes the number of instances in the LR set with $lr = |LR|$ and so on; $n = l + \bar{l} = r + \bar{r}$ indicates the total number of instances.

The relative frequency $|A|/n = a/n$ associated with the subset A is denoted by $p(A)$, where A is a subset of the n instances. In this way, the relative frequency is used as a probability estimate. The notation $p(A|B)$ follows its usual definition in probability, given by (1), where A and B are both subsets of the n instances.

$$p(A|B) = \frac{p(A \cap B)}{p(B)} = \frac{p(AB)}{p(B)} = \frac{\frac{|AB|}{n}}{\frac{|B|}{n}} = \frac{\frac{ab}{n}}{\frac{b}{n}} = \frac{ab}{b} \tag{1}$$

Many measures can be used to evaluate the performance of a leaf. Precision (positive reliability) is the most common. However, with new problems to be dealt with, new measures such as novelty, simplicity and ease of human understanding may be interesting [25]. Based on the contingency matrix, it is possible to define most measures about rules. Of special interest in this work will be used the positive reliability metrics *prel* (2), novelty *nov* (3), satisfaction *sat* (5) and Laplace precision *lacc* (6).

Positive reliability corresponds to the ratio between the number of instances correctly classified by the rule and the total number of instances covered by the rule. Assumes values in the range $[0, 1]$.

$$prel(L \rightarrow R) = p(R|L) = \frac{lr}{l} \tag{2}$$

$$nov(L \rightarrow R) = p(LR) - p(L)p(R) = \frac{lr}{n} - \frac{l \cdot r}{n^2} \tag{3}$$

$$nov4(L \rightarrow R) = 4 \times nov(L \rightarrow R) \tag{4}$$

$$sat(L \rightarrow R) = \frac{p(\bar{R}) - p(\bar{R}|L)}{p(\bar{R})} = 1 - \frac{n \cdot l\bar{r}}{l \cdot \bar{r}} \tag{5}$$

$$lacc(L \rightarrow R) = \frac{lr + 1}{l + k} \tag{6}$$

Considering L and R, the novelty is defined by checking whether LR is independent of them. This can be obtained by comparing the observed result lr against the expected value under the independence consideration $\frac{l \cdot r}{n}$. The more the observed value differs from the expected value, greater the probability that there is a true and unexpected association between L and R. This metric takes values in the range $[-0.25, 0.25]$. It can be shown that the higher a positive value

(close to 0.25), the stronger the association between L and R, while the smaller a negative value (close to -0.25), the stronger the association between L and $\bar{R}$. In this work, the value of the novelty metric was multiplied by four to in order to place the metric in the range $[-1, +1]$, which leads to (4).

Satisfaction is the relative increase in precision between the rule $L \rightarrow \mathit{true}$ and the rule $L \rightarrow R$. According to [18], this measure, whose values vary in the range $[-1, +1]$, is suitable for tasks aimed at discovering knowledge, being able to promote a balance between rules with different conditions and conclusions.

As can be seen, the novelty and satisfaction metrics can take on negative values. As the purpose of this work is to represent a decision tree containing rules whose conclusion is the class (and not its complement, that is, all other classes), rules with negative values for these two metrics will not be considered.

Laplace's precision does not fit directly into the frequency/probability notation proposed by [18] but fixes the problem of rules with few errors covering many examples of positive reliability [5]. In (6), k represents the number of classes in the training set. This metric takes values in the range $(0, 1)$.

2.2 Tree leaves weights normalization

In the induction process of the Meta Decision Tree, it is expected that such tree reflects the number of examples provided in the training set, in a way analogous to the generation of a single tree without the use of meta-learning. The approach adopted for this in this research is described below.

Let T_i be a Random Forest Tree in which the leaves' weights without normalization are $\{w_{i1}, w_{i2}, \ldots\}$. Let the sum of weights of a tree T_i given by $W_i = \sum_j w_{ij}$. The weights for the tree T_i must be adjusted to sum 1, given by $\{\frac{w_{i1}}{W_i}, \frac{w_{i2}}{W_i}, \ldots\}$. Now, for the tree T_i to represent the total number of instances n of the training set, the bootstrap sample size n_i to generate each tree is equal to the number of instances n in the training set. The weights of each tree are given by $\{n\frac{w_{i1}}{W_i}, n\frac{w_{i2}}{W_i}, \ldots\}$. Considering that there are $\mathcal{L}$ trees in the forest $\{T_1, T_2, \ldots, T_{\mathcal{L}}\}$, it is necessary to adjust the weight of the meta decision tree as being $\{n\frac{w_{i1}}{\mathcal{L}W_i}, n\frac{w_{i2}}{\mathcal{L}W_i}, \ldots\}$ for each decision tree, such as $\{\{n\frac{w_{11}}{\mathcal{L}W_1}, n\frac{w_{12}}{\mathcal{L}W_1}, \ldots\}, \ldots, \{n\frac{w_{\mathcal{L}1}}{\mathcal{L}W_{\mathcal{L}}}, n\frac{w_{\mathcal{L}2}}{\mathcal{L}W_{\mathcal{L}}}, \ldots\}\}$

2.3 Example

Fig. 1 presents a simple example of the implementation our algorithm using data from Table 2. The number of instances in each leaf was used to represent the weight of the respective row in the decision table for ease of reading. To start the process, consider a Random Forest that contains only two trees. For the creation of this Random Forest consider the set of instances represented in the Table 2. The process (1) is responsible for transforming decision trees into decision tables (Algorithm 2). The process (2) performs the union of decision tables, thus forming, after the necessary formatting, a new set of instances. The process (3) performs the induction of the Meta Decision Tree based on the new set of instances.

Table 2. Toy dataset containing instances about friendly and enemy robots described by four attributes (Head, Body, Hold and Smile) and two classes (friend and enemy).

Instance	Head	Body	Hold	Smile	Class
z_1	round	square	flag	no	enemy
z_2	triangular	triangular	balloon	yes	friend
z_3	round	round	flag	yes	friend
z_4	square	triangular	sword	no	enemy
z_5	square	square	balloon	yes	friend
z_6	triangular	round	sword	yes	enemy

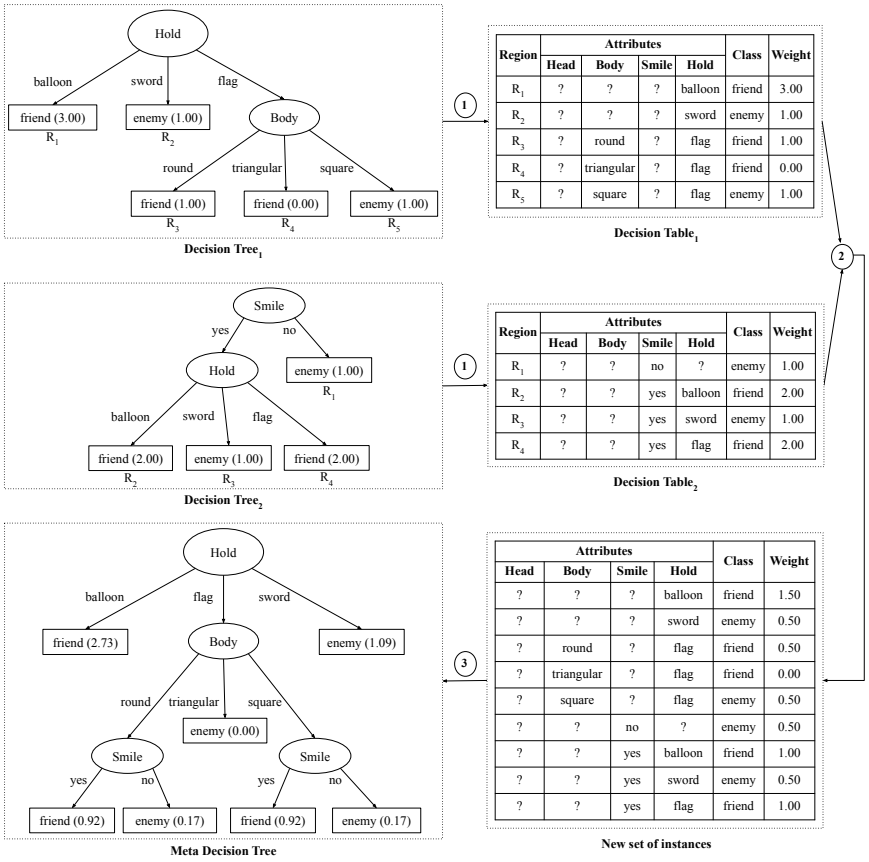

Fig. 1. Example of our meta induction tree algorithm on dataset described in Table 2. In this example, two decision trees are induced (first two trees/tables) that are then combined into a resulting Meta Decision Tree (last tree/table) in figure.

3 Experimental Setup

To analyze the efficiency of the proposed algorithm, it was evaluated through an experimental study as described below. The experimental study compared a single Decision Tree, a Random Forest and the Meta Decision Tree, considering the predictive performance criterion for evaluation.

Predictive performance was evaluated using the multiclass extension of the Area Under the Curve (AUC) measure, which aggregates the AUC values over each pair of classes [16].

Friedman's test [11] was used for pairwise multiple comparisons, which assume that the difference in the data is by chance as the null hypothesis, considering a confidence level of 95%. The null hypothesis assumes all algorithms have equal performance. The Friedman did reject the null hypothesis, and the Bonferroni-Dunn [9] post-hoc test was employed to detect any significant difference among algorithms, also using a confidence level of 95%.

In conducting experiments, all algorithms were evaluated by 10-fold cross-validation. The models analyzed were: a single Decision Tree (DT), a Random Forest (RF with $\mathcal{L} = 128$ trees) and a Meta Decision Tree (MDT). The Meta Decision Tree was evaluated with four weights: MDT-Precision, MDT-Laplace, MDT-Novelty, and MDT-Satisfaction, using Eqs. (2), (6), (4), and (5), respectively. The Weka machine library was used to run all experiments [15].

Twenty-nine datasets from different domains were selected for the experiment. The datasets were obtained from OpenML [26]. An important consideration is that all datasets used contain only categorical (nominal) attributes.

4 Results and Discussion

Table 3 shows the AUC mean and standard deviation values for each dataset and each algorithm. Fig. 2 shows the critical difference diagram for each algorithm considering all datasets shown in Table 3.

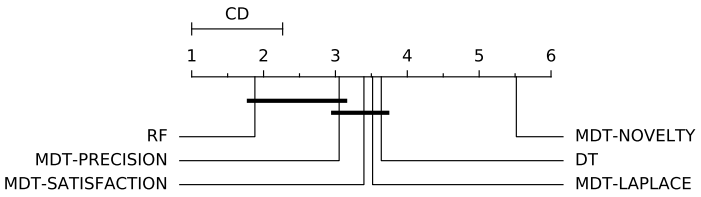

Fig. 2. Critical difference diagram for each algorithm, and all datasets using Bonferroni-Dunn post-hoc test.

Table 3. AUC Mean and standard deviation values from 10-fold cross-validation. The highest figures on a row are in boldface. Green-shaded cells correspond to values that do not have a significant difference (p-value> 0.05) when compared to the best value(s) in the row by the Bonferroni-Dunn post-hoc test.

Dataset	MDT-Prec.	MDT-Lapl.	MDT-Nov.	MDT-Satisf.	RF	DT
audiology	92.03±2.94	81.74±4.80	77.14±0.95	92.11±2.82	**97.42±1.73**	93.22±2.61
blogger	77.71±18.46	76.76±20.21	59.46±20.76	74.35±18.32	**90.12±9.12**	69.4±20.94
boxing1	84.93±15.22	85.00±15.03	50.00±0.00	84.93±15.22	**89.24±8.71**	87.83±12.47
boxing2	**85.69±8.93**	85.66±8.99	50.00±0.00	**85.69±8.93**	85.06±10.61	84.29±10.47
breast-cancer	62.29±13.83	61.48±14.57	50.00±0.00	61.94±12.10	**65.14±13.89**	62.81±10.02
car-df	98.96±0.52	98.09±0.73	50.00±0.00	98.96±0.52	**99.45±0.23**	97.62±0.64
dbworld-subjects	85.83±19.61	68.19±18.64	50.00±0.00	85.83±19.61	**95.14±7.15**	75.00±10.39
dmft	54.31±2.69	**54.95±1.86**	53.12±2.18	53.84±2.27	52.74±2.75	54.57±2.10
dna	97.39±0.75	97.01±1.27	82.97±1.66	97.39±0.75	**99.30±0.20**	94.67±1.51
donner	**60.00±41.16**	58.33±39.67	50.00±0.00	58.33±39.67	20.00±34.96	50.00±0.00
fraud	71.67±24.91	73.33±23.83	73.33±21.08	71.67±24.91	**78.33±22.29**	69.17±24.23
king-and-rook	91.61±0.23	88.53±0.29	50.00±0.00	91.61±0.23	**96.20±0.16**	87.84±0.39
kr-vs-kp	99.47±0.43	99.50±0.42	93.76±1.16	99.47±0.43	**99.93±0.11**	99.88±0.19
lung-cancer	52.50±14.72	50.00±0.00	50.00±0.00	52.50±14.72	**68.33±28.81**	67.71±22.97
marketing	52.17±6.86	52.17±6.86	50.00±0.00	52.17±6.86	**66.56±5.37**	61.50±10.49
monks-problems	58.99±7.18	59.55±8.27	50.00±0.00	59.10±7.34	**79.61±8.05**	54.06±6.89
mushroom	99.90±0.09	99.84±0.19	99.69±0.35	99.90±0.09	**100.00±0.00**	**100.00±0.00**
nursery	99.86±0.05	99.79±0.06	96.57±0.44	99.86±0.05	**99.97±0.01**	99.54±0.14
phishing	98.26±0.31	98.22±0.30	95.18±0.39	98.22±0.29	**99.59±0.06**	98.43±0.49
po-patient	40.06±10.42	46.61±16.91	**50.00±0.00**	44.87±11.66	44.81±18.61	49.29±2.26
primary-tumor	79.43±3.06	72.98±3.47	60.91±2.00	79.24±4.02	**80.17±2.51**	70.38±3.47
reviewer	65.33±8.94	65.08±7.63	50.00±0.00	64.89±8.95	**68.92±7.54**	64.78±7.54
servo	97.63±3.20	97.53±3.32	95.38±4.73	97.63±3.20	**98.91±1.91**	95.38±4.73
solar-flare-1	90.03±3.85	**90.21±2.59**	78.12±2.80	89.83±3.42	89.54±4.46	88.93±3.27
solar-flare-2	92.13±1.42	**92.18±1.50**	92.08±1.48	92.11±1.54	91.98±1.47	91.96±0.73
soybean	96.83±0.79	94.71±2.78	79.22±2.17	96.84±0.80	**99.70±0.26**	98.42±0.60
spect	78.86±10.79	78.26±11.36	71.21±14.54	77.63±10.91	79.15±11.45	**80.64±10.70**
splice	98.34±0.50	98.02±0.77	58.24±13.28	98.34±0.50	**99.45±0.26**	96.44±0.82
vote	97.79±1.90	97.82±1.89	96.54±3.25	97.66±2.13	**99.11±0.96**	97.96±2.24
Mean	81.37±17.81	80.05±17.40	67.68±18.81	81.27±17.60	**83.92±19.35**	80.74±16.88
Average Rank	3.05	3.52	5.52	3.40	**1.88**	3.64

As expected, the RF algorithm outperformed significantly the DT and MDT-Satisfaction, MDT-Laplace, DT, and MDT-Novelty. However, the results for both RF and MDT-Precision are not statistically significant different.

The DT algorithm only significantly outperformed the MDT-Novelty algorithm. For the MDT-Precision, MDT-Satisfaction and MDT-Laplace algorithms, DT obtained a lower performance, but, not significant. When compared to the RF algorithm, DT had a significantly lower performance. Regarding the performance of our algorithm and leaf weights, the ones that obtained the best performance were MDT-Precision and MDT-Satisfaction, showing an interesting result concerning precision and satisfaction metrics.

In summary, our approach MDT-Precision performs as good as a Random Forest with no statistically significant differences. Yet, its advantage is the interpretability provived by a single Decision Tree.

5 Conclusions

In this study, we have used meta-learning to transform trees from a Random Forest into a unique decision tree, a more human-interpretable model. The main contribution of our work was to show that it is possible to obtain a single tree with a performance statistically similar to that of a Random Forest using MDT-Precision. Continuing this work, we are analyzing how to handle datasets containing numeric attributes. Some initial ideas would be to represent the average value of the test performed on the attribute in the decision table; another possibility would be to represent the attribute limits with two lines in the decision table, one for the lower limit and one for the upper one.

References

1. Ampomah, E.K., Qin, Z., Nyame, G.: Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. Information **11**(6) (2020). https://doi.org/10.3390/info11060332, https://www.mdpi.com/2078-2489/11/6/332
2. Breiman, L.: The heuristics on instability in model selection. Tech. rep., Statistics Department, University of California (1996)
3. Burkart, N., Huber, M.F.: A survey on the explainability of supervised machine learning. Journal of Artificial Intelligence Research **70**, 1–74 (2021)
4. Carvalho, D.V., Pereira, E.M., Cardoso, J.S.: Machine learning interpretability: A survey on methods and metrics. Electronics **8**(8) (2019)
5. Clark, P., Boswell, R.: Rule induction with cn2: Some recent improvements. In: Kodratoff, Y. (ed.) Proceedings of the 5th European Conference (EWSL 91). pp. 151–163. SV (1991)
6. COUNCIL OF EUROPEAN UNION: Council regulation (eu) no 279/2016 - official website of the european union (2016), https://eur-lex.europa.eu/eli/reg/2016/679/oj
7. Dietterich, T.G.: Machine learning research: Four current directions (may 1997)
8. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)
9. Dunn, O.J.: Multiple comparisons among means. Journal of the American Statistical Association **56**(293), 52–64 (1961). https://doi.org/10.1080/01621459.1961.10482090
10. ElShawi, R., Sherif, Y., Al-Mallah, M., Sakr, S.: Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. Computational Intelligence **37**(4), 1633–1650 (2021)
11. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics **11**(1), 86–92 (1940). https://doi.org/10.1214/aoms/1177731944
12. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. In: 2018 IEEE 5th Int. Conf. on Data Science and Advanced Analytics (DSAA). pp. 80–89 (2018)
13. Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems **34**(6), 14–23 (2019)

14. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) **51**(5), 1–42 (2018)
15. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter **11**(1), 10–18 (2009)
16. Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Mach. Learn. **45**(2), 171–186 (Oct 2001). https://doi.org/10.1023/A:1010920819831, https://doi.org/10.1023/A:1010920819831
17. Lakhani, P., Prater, A.B., Hutson, R.K., Andriole, K.P., Dreyer, K.J., Morey, J., Prevedello, L.M., Clark, T.J., Geis, J.R., Itri, J.N., et al.: Machine learning in radiology: applications beyond image interpretation. Journal of the American College of Radiology **15**(2), 350–359 (2018)
18. Lavrač, N., Flach, P., Zupan, R.: Rule evaluation measures: A unifying view. In: Dzeroski, S., Flach, P. (eds.) Proceedings of the Ninth International Workshop on Inductive Logic Programming (ILP-99). vol. 1634, pp. 74–185. SV (jun 1999), lNAI
19. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable ai: A review of machine learning interpretability methods. Entropy **23**(1) (2021)
20. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proc. of the 31st Int. Conf. on Neural Information Processing Systems. p. 4768–4777. NIPS'17, Curran Associates Inc. (2017)
21. Oshiro, T.M., Perez, P.S., Baranauskas, J.A.: How many trees in a random forest? In: Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM 2012, Lecture Notes in Computer Science, ISBN 978-3-642-31536-7. vol. 7376, pp. 154–168. Berlin, Germany (July 13-20 2012), http://dx.doi.org/10.1007/978-3-642-31537-4_13
22. Pham, K., Kim, D., Park, S., Choi, H.: Ensemble learning-based classification models for slope stability analysis. CATENA **196**, 104886 (2021). https://doi.org/https://doi.org/10.1016/j.catena.2020.104886, https://www.sciencedirect.com/science/article/pii/S0341816220304367
23. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
24. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proc. of the 22nd ACM SIGKDD Int. Conf. on knowledge discovery and data mining. pp. 1135–1144 (2016)
25. Todorovski, L., Flach, P., Lavrač, N.: Predictive performance of weighted relative accuracy. In: Zighed, D.A., Komorowski, J., Zytkow, J. (eds.) 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000). pp. 255–264. SV (sep 2000)
26. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: Openml: networked science in machine learning. SIGKDD Explorations **15**(2), 49–60 (2013). https://doi.org/10.1145/2641190.2641198, http://doi.acm.org/10.1145/2641190.264119