

LionForests: Local Interpretation of Random Forests

Ioannis Mollas, Nick Bassiliades, Ioannis Vlahavas, Grigorios Tsoumakas¹

Abstract. Towards a future where machine learning systems will integrate into every aspect of people’s lives, researching methods to interpret such systems is necessary, instead of focusing exclusively on enhancing their performance. Enriching the trust between these systems and people will accelerate this integration process. Many medical and retail banking/finance applications use state-of-the-art machine learning techniques to predict certain aspects of new instances. Tree ensembles, like random forests, are widely acceptable solutions on these tasks, while at the same time they are avoided due to their black-box uninterpretable nature, creating an unreasonable paradox. In this paper, we provide a methodology for shedding light on the predictions of the misjudged family of tree ensemble algorithms. Using classic unsupervised learning techniques and an enhanced similarity metric, to wander among transparent trees inside a forest *following breadcrumbs*, the interpretable essence of tree ensembles arises. An interpretation provided by these systems using our approach, which we call “LionForests”, can be a simple, comprehensive rule.

1 INTRODUCTION

Machine learning models are becoming pervasive in our society and everyday life. Such models may contain errors, or may be subject to manipulation from an adversary. In addition, they may be mirroring the biases that exist in the data from which they were induced. For example, Apple’s new credit card is being recently investigated over claims it gives women lower credit [15], IBM Watson Health was accused of suggesting unsafe treatments for patients [6] and state-of-the-art object detection model YOLOv2 is easily tricked by specially designed patches [9, 39]. Being able to understand how a machine learning model operates and why it predicts a particular outcome is important for engineering safe and unbiased intelligent systems.

Unfortunately, many families of highly accurate (and thus popular) models, such as deep neural networks and tree ensembles, are opaque: humans cannot understand the inner workings of such models and/or the reasons underpinning their predictions. This has recently motivated the development of a large body of research on *interpretable machine learning* (IML), concerned with the interpretation of black box models [1, 12, 13, 20, 21, 26, 37].

Methods for interpreting machine learning models are categorised, among other dimensions [20], into *global* ones that uncover the whole logic and structure of a model and *local* ones that aim to interpret a single prediction, such as “Why has this patient to be immediately hospitalized?”. This work focuses on the latter category. Besides their utility in uncovering errors and biases, local interpretation methods are in certain domains a prerequisite due to legal frameworks, such as the General Data Protection Regulation (GDPR) [34]

of the EU and the Equal Credit Opportunity Act of the US².

Another important dimension, which IML methods can be categorised, concerns the type of machine learning model that they are interpreting [1]. *Model-agnostic* methods [28, 35, 36] can be applied to any type of model, while *model-specific* methods [4, 27, 30, 31] are engineered for a specific type of model. Methods of the former category have wider applicability, but they just approximately explain the models they are applied to [1]. This work focuses on the latter category of methods, proposing a technique specified for tree ensembles [5, 7], which are very effective in several applications involving tabular and time-series data [43].

Past work on model-specific interpretation techniques, about tree ensembles, is limited [31, 41]. iForest [41], a global and local interpretation system of random forests, provides insights for a decision through a visualisation tool. Notwithstanding, such visual explanation tool is very complex for non-expert users, while at the same time requires user interaction in order to construct the local interpretations. Another instance-level interpretation technique for random forests [31], produces an interpretation in the form of a list of features with their ranges, accompanied by an influence metric. If the list of features is extensive, and the ranges are very narrow, the interpretation can be considered as unreliable and untrustworthy, because small changes in the features will render the interpretation useless. Finally, both methods do not handle categorical data appropriately.

To address the above problems, we introduce a local-based model-specific approach for interpreting an individual prediction of a random forest through a single rule, in natural language form. The ultimate goal is mainly to reduce the number of features and secondly to broaden the feature-ranges producing more robust, indisputable and intuitive interpretations. Additionally, the categorical features are handled properly, providing intelligible information about them throughout the rules. The constructed rule will be presented as the interpretation, if its length is acceptable to be comprehensible. Otherwise, additional processing will be held to form an acceptable interpretation. We call this technique “LionForests” (Local Interpretation Of raNdom FORESTS) and we use its path and feature selection ability, based on unsupervised techniques like association rules [2] and *k*-medoids clustering [24] to process the interpretations in order to make them more comprehensible.

The rest of this work is structured the following way. First, Section 2 introduces the related work, presenting approaches of interpretation techniques applicable to random forest models. In Section 3, we present the methodology of our approach to the interpretation of random forest models. In section 4, we perform the quantitative and qualitative analysis. Finally, we conclude and present possible future directions of our strategy in Section 5.

¹ Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece, email: iamollas,nbassili,vlahavas,greg@csd.auth.gr

² ECOA 15 U.S. Code Â§1691 et seq.: <https://www.law.cornell.edu/uscode/text/15/1691>

2 RELATED WORK

A number of works concerning the tree ensembles interpretation problem are either model-agnostic or model-specific solutions, with a global or local scope, as presented with a similar taxonomy in a recent survey [?].

A family of model-agnostic interpretation techniques about black-box models, including tree ensembles, concerns the efficient calculation of feature importance. These are variations of feature permutation methods [16], partial dependence plots [?] and individual conditional expectation [17], which are global-based. SHAP [28] is an alternative method to compute feature importance for both global and local aspects of any black-box model.

Specifically, on tree ensembles, the most common techniques include the processes of extracting, measuring, pruning and selecting rules from the trees to compute the feature importance [10, 38]. A highly studied by many researchers [10, 11, 23, 42] technique attempts to globally interpret tree ensembles using single-tree approximations. But this method, as its name implies, approximates the performance of the model it seeks to explain. Thus, this approach is extremely problematic and criticised, because it is not feasible to summarise a complex model like tree ensembles to a single tree [19]. An additional approach on interpreting tree ensembles focuses on clustering the trees of an ensemble using a tree dissimilarity metric [8].

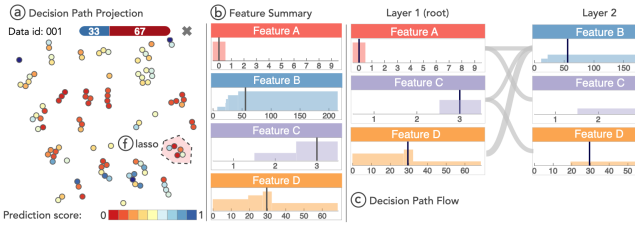


Figure 1. A sample example from the visualisation tool of iForest adapted from [41]

iForest [41], a model-specific global and local approach, utilises a path distance metric to project an instance’s paths to a two-dimensional space via t-Distributed Stochastic Neighbour Embedding (t-SNE) [29]. The path distance metric they propose considers distant two paths in two cases: a) if a feature exists in only one out of the two paths the distance between those two paths is increasing, b) if a feature exists in both paths, the distance is increasing according to the non-common ranges of the feature on the paths divided by half. The total distance, which is the aggregation of those cases for all the features, is finally divided by the total number of features appearing at least in one out of the two paths. Except the projection of the paths (Figure 1a), they provide feature summary (Figure 1b) and decision path flow (Figure 1c), which is a paths overview. In feature summary, a stacked area plot visualises every path’s range for a specific feature, while decision path flow plot visualises the paths themselves. However, they do not provide this information automatically. The user has to draw a lasso (Figure 1f) around some points-paths in the paths projection plot in order to get the feature summary and paths overview. But requiring the user to select the appropriate paths is critical, simply because the user can easily choose wrong paths, a small set of paths, or even paths entirely different to the paths being responsible for his prediction. That may lead to incorrect feature summary and paths overview, thus to a faulty interpretation.

Lastly, one technique [31] interprets tree ensembles in instance-level (local-based technique) providing as interpretation a set of fea-

Class A (95.1%)

Rank	Feature	Influence	Min	Max
1	Feature D	+74.9	7073.5	7074
2	Feature A	+22.0	12.5	∞
3	Feature C	+19.3	0.5	1
1	Feature E	-12.27	$-\infty$	0.5
2	Feature B	-0.73	0.25	0.5
3	Feature I	-0.60	$-\infty$	174

Figure 2. A sample explanation from [31]

tures with their ranges, ranked based on their contribution (see Figure 2). Thus, the interpretation process consists of two parts. Firstly, they calculate the influence of a feature by monitoring the changes of the activated nodes for a specific instance’s prediction. This influence later will be used for the ranking process. The second step is to find the narrowest range across all trees for every feature. However, they do not attempt to expand these ranges, while they also claim that their influence metric assigns zero influence to some features, and by extension removing them, they could offer more compact explanations. In spite of this, they do not know, by keeping only features with a non-zero influence, that these features will at least be present in half plus one paths to preserve the same prediction of an instance. Finally, they do not manage categorical features properly.

3 OUR APPROACH

Our objective is to provide local interpretation of random forest binary classifiers. In random forests, a set of techniques like data and feature sampling, is used in order to train a collection of T weak trees. Then, these trained trees vote for an instance’s prediction:

$$f(x_i) = \frac{1}{|T|} \sum_{t \in T} f_t(x_i) \quad (1)$$

where $f_t(x_i)$ is the vote cast from the tree $t \in T$ for the instance $x_i \in X$, representing the probability $P(C = c_j | X = x_i)$ of x_i to be assigned to class $c_j \in C = \{0, 1\}$.

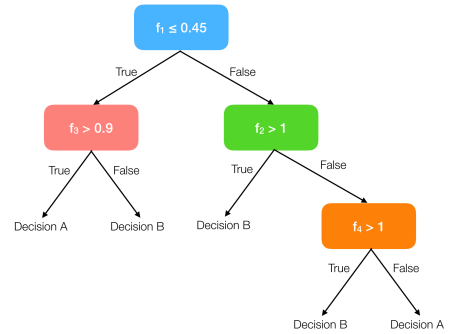


Figure 3. A decision tree binary classifier with 4 features

Each decision tree $t \in T$ is a directed graph, and by deconstructing its structure, we are able to derive a set P_t of paths from the root to the leaves. Therefore, every instance can be classified with one of these paths. A path $p \in P_t$ is a conjunction of conditions, and the conditions are features and values with relations \leq and $>$. For example, a path from the tree on Figure 3 could be the following: ‘if $f_1 \leq 0.45$ and $f_3 > 0.9$ then Decision A’. Thus, each path p is expressed as a set:

$$p = \{f_i \boxtimes v_j | f_i \in F, v_j \in \mathbb{R}, \boxtimes \in \{\leq, >\}\} \quad (2)$$

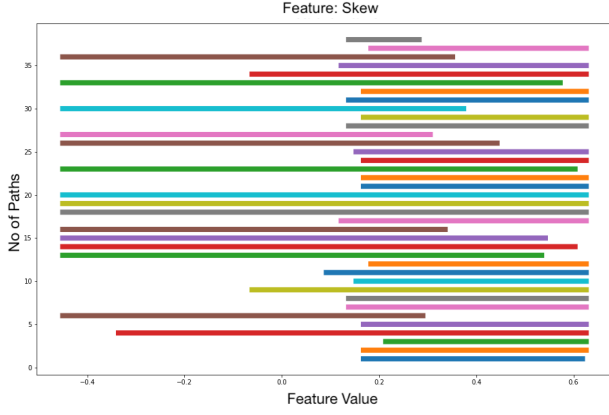


Figure 4. Bars plot of all paths containing 'skew' feature from Banknote Dataset

We are presenting LionForests, a framework for interpreting random forests models in the instance level. LionForests is a pipeline of actions: a) feature-ranges extraction, reduction through b₁) association rules, b₂) clustering and b₃) random selection, c) categorical feature handling, and, finally, d) interpretation composition.

3.1 Feature-Ranges Extraction

Our approach delivers local explanations for random forests. Assume we have a random forest of N trees and an instance x , which is classified by the forest as $c_j \in C$. We focus on the M trees of the forest that classify x as c_j . For each feature, we compute from each of the M trees that it appears in, its values range as imposed from the conditions involving this feature in the path from the root to the leaf. Figure 4 shows an example of these ranges for feature 'skew' with range $-1 \dots 1$ from the Banknote dataset [14] for a particular random forest of N trees and a particular instance whose value for the skew feature is Y . Figure 6 shows the corresponding stacked area plot. The highlighted (cyan/light grey) area represents the intersection of these ranges, which they will always contain the instance's value for the specific feature. Moreover, no matter how much the feature value is going to change, as long as it stays within this intersection range, the instance will not follow a different decision path in these trees.

In order to give a brief example, we have the following three paths:

- p_1 if $f_1 \leq 0.6$ and ... then Class_A
- p_2 if $f_1 \leq 0.6$ and $f_1 > 0.469$ and ... then Class_A
- p_3 if $f_1 > 0$ and $f_1 \leq 1$ and ... then Class_A

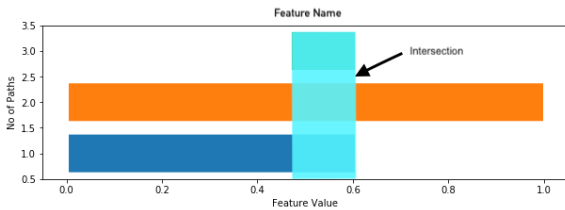


Figure 5. Bars plot of the simple example

The intersection of this three paths is the cyan/light grey area in Figure 5, which we can infer that the f_1 feature's range can be: $0.47 \leq f_1 \leq 0.6$. This intersection range will always contain the instance's value for the specific feature. Moreover, no matter how much the feature value is going to change, as long as it stays within

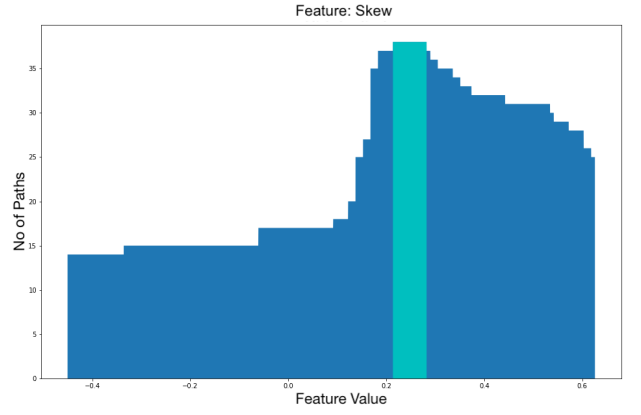


Figure 6. Stacked area plot of all paths containing 'skew' feature from Banknote Dataset

this intersection range, the decision paths are not going to change. For example, if the value of the instance for the feature f_1 was 0.5, if the value will change to 0.52, each tree will take its decision through the same path. Summarising the aforementioned, an interpretation can have this shape:

$$\text{'if } 0.47 \leq f_1 \leq 0.6 \text{ and } \dots \text{ then Class_A'}$$

But with as many paths as possible to vote to class A , we face the following two issues:

1. A lot of paths can lead to an explanation with many features, by extension to an unintelligible understanding and a frustrated user.
2. A lot of paths will lead to a small, strict and very specific feature range. For example, f_1 instance's value was 0.5 and the intersection range of all paths for this feature occurs to be $0.47 \leq f_1 \leq 0.6$, while the feature range is $[-1, 1]$. A narrow range, like the aforementioned, would result in a negative impression of the model, which will be considered unstable and unreliable. Consequently, a broader range will be less refutable.

Consequently, we formulate the optimisation problem (Eq. 3) to minimise the number of features that satisfy the paths of a subset of the original trees, thereby retaining the same classification result with the original set of trees and making the size of the total number of trees equal to or greater than the quorum.

quorum ['kwɔːrəm]: the minimum number of members of an assembly or society that must be present at any of its meetings to make the proceedings of that meeting valid [32].

$$\begin{aligned} & \underset{F' \subseteq F}{\text{minimise}} && |F'| \\ & \text{subject to} && p = \{f_i \boxtimes v_j | f_i \in F'\}, p \in P_t \forall t \in T', \\ & && \lfloor \frac{1}{|T'|} \sum_{t \in T'} f_t(x_i) + \frac{1}{2} \rfloor = \lfloor \frac{1}{|T|} \sum_{t \in T} f_t(x_i) + \frac{1}{2} \rfloor, \\ & && |T'| \geq \text{quorum} \end{aligned} \tag{3}$$

3.2 Reduction through Association Rules

The first step of the reduction process begins by using association rules. Association rules [2] mining is an unsupervised technique,

which is used as a tool to extract knowledge from large datasets and explore relations between attributes. In association rules, the attributes are called items $I = \{i_1, i_2, \dots, i_n\}$. Each dataset contains sets of items, called itemsets $T = \{t_1, t_2, \dots, t_m\}$, where $t_i = \{i_j | i_j \in I\}$. Using all possible items of a dataset, we can find all the rules $X \Rightarrow Y$, where $X, Y \subseteq T$. X is called antecedent, while Y is called consequent. In association rules the goal is to calculate the support and confidence of each rule in order to find useful relations. A simple observation is that X is independent from Y when the confidence is critically low. Furthermore, we can say that X with high support, means it is probably very important.

But how can we use association rules in random forests? We are going to do this in the path-level. The items I will contain the features F of our original dataset. The dataset T , which we are going to use to mine the association rules, will contain sets of features that represent each path $t_i = \{i_j | i_j = f_j \in p_i, p_i \in P\}$. Then, it is feasible to apply association rules techniques, like apriori algorithm [3].

The next step is to sort the association rules extracted by the apriori algorithm based on the ascending confidence score of each rule. For the rule $X \Rightarrow Y$, with the lowest confidence, we will take the X and will add its items to the list of features. Afterwards, we are calculating the number of paths containing conjunctions, which are satisfied with the new feature set. If the amount of paths is at least half plus one paths of the total number of trees, we have found the reduced set of paths. *We have a quorum!* Otherwise, we iterate and add more features from the next antecedent of the following rule. By using this technique, we reduce the number of features and we have the new feature set $F' \subseteq F$. Reducing the features, can lead to a reduced set of paths too, because paths containing conjunctions with the redundant features will no longer be valid. Thus, for every path p we have the following representation:

$$p = \{f_i \boxtimes v_j | f_i \in F', v_j \in \mathbb{R}, \boxtimes \in \{\leq, >\}\}. \quad (4)$$

Illustrating this, for a toy dataset of four features $F = [f_1, f_2, f_3, f_4]$ and a random forests model with five estimators $T = [t_1, t_2, t_3, t_4, t_5]$, for every instance x , from each $t_i \in T$ we can extract a path p_i . Supposing that for the instance x , we have five paths:

- p_1 if f_1 and f_2 and f_4 then Class_A
- p_2 if f_1 and f_3 and f_4 then Class_A
- p_3 if f_1 and f_2 and f_4 then Class_A
- p_4 if f_3 and f_4 then Class_A
- p_5 if f_4 then Class_A

Then, we can compute the association rules using apriori. Our objective is to create a set of features $F' \subset F$. We take the first rule $f_4 \Rightarrow (f_3, f_1)$, the rule with the lowest confidence. This rule informs us that f_4 , which has the highest support value, exist in 80% of the paths without (f_1, f_3) . Thus, the first thing we add to our feature list is the antecedent of this rule, f_4 . By adding the feature, we are counting how many paths can be satisfied with the features of $F' = [f_4]$. Only one path is valid (p_5), and is not enough because we need a quorum. Skipping all the association rules having the chosen features at their antecedents, the next rule we have is $f_1 \Rightarrow f_3$. f_1 has 0.6 support value, and the rule has 0.33 confidence. This means that in 66.6% of paths containing f_1 , the f_3 is absent. We add f_1 to the feature list and now the $F' = [f_1, f_4]$. With this feature list only the p_5 is activated again. Hence, we need another feature. The next rule we have is $f_3 \Rightarrow (f_1, f_4)$ with 0.4 support of f_3 and confidence 0.2. Adding f_3 now the paths p_2, p_4 and p_5 are valid.

In the aforementioned example, we achieved to reduce the features from four to three and the paths from five to three, as well. However,

applying this method to datasets with plenty of features and models with more estimators, the reduction effect can be observed. Section 4 is seeking to explore that effect, through a set of experiments.

3.3 Reduction through Clustering and Random Selection

However, association rules may not be able to reduce the number of features and, consequently, the number of paths. In that case, a second reduction technique based on clustering is applied. Clustering is yet another group of unsupervised machine learning techniques, aside from the association rules. k -medoids [24] is a well-known clustering algorithm, which considers as cluster's centre an existing element from the dataset. This element is called medoid. k -medoids, like other clustering techniques, needs a distance or a dissimilarity metric to find the optimum clusters. Thus, performing clustering to paths will require a path specific distance or dissimilarity metric.

We designed a path similarity metric in Algorithm 1. This similarity metric is close to the distance metric introduced in iForest [41], but eliminates some minor problems of the aforementioned. Parsing this algorithm, if a feature is absent from both paths, the similarity of these paths increases by 1. When a feature is present in both paths, the similarity increases by a value between 0 and 1, which is the intersection of the two ranges normalised by the union of the two ranges. The similarity metric is biased towards the absence of a feature from both paths because it assigns always one similarity point. However, this is a desirable feature for our goal of minimising the feature set.

Algorithm 1: Path similarity metric

```

input :  $p_i, p_j, \text{feature\_names}, \text{min\_max\_feature\_values}$ 
return :  $\text{similarity}_{ij}$ 

 $s_{ij} \leftarrow 0$ 
for  $f$  in  $\text{feature\_names}$  do
    if  $f$  in  $p_i$  and  $f$  in  $p_j$  then
        find  $l_i, u_i, l_j, u_j$  lower and upper bounds
        if  $u_i \leq l_j$  or  $u_j \leq l_i$  then
            |  $s_{ij} \leftarrow s_{ij}$ 
        else
             $\text{inter} \leftarrow \min(u_i, u_j) - \max(l_i, l_j)$ 
             $\text{union} \leftarrow \max(u_i, u_j) - \min(l_i, l_j)$ 
            if  $\text{union} \neq 0$  then
                |  $s_{ij} \leftarrow s_{ij} + \text{inter}/\text{union}$ 
            end
        end
    else if  $f$  not in  $p_i$  and  $f$  not in  $p_j$  then
        |  $s_{ij} \leftarrow s_{ij} + 1$ 
    end
end
return  $s_{ij}/\text{len}(\text{feature\_names})$ 

```

In Algorithm 2, we calculate the k medoids and their clusters using the similarity metric of Algorithm 1. Afterwards, we perform an ordering of the medoids based on the number of paths they cover in their clusters. Then, we collect paths from the larger clusters into a list, until we acquire at least a quorum. By summing larger clusters first, the possibility of feature reduction is increasing, because the paths inside a cluster tend to be more similar among them. Additionally, the biased similarity metric would cluster paths with less irrelevant features, leading to a subset of paths that are satisfied with a smaller set of features.

Algorithm 2: Paths reduction through k -medoids clustering

```
input :  $similarity\_matrix, no\_of\_estimators,$   
         $paths, no\_of\_medoids$   
return:  $paths$   
 $quorum \leftarrow no\_of\_estimators/2 + 1$   
 $m \leftarrow kmedoids(similarity\_matrix, no\_of\_medoids)$   
 $sorted\_m \leftarrow sort\_by\_key(m, descending = True)$   
 $count \leftarrow 0, size \leftarrow 0, reduced\_paths \leftarrow []$   
while  $size < quorum$  and  $count < len(sorted\_m)$  do  
    for  $j$  in  $m[sorted\_m[count]]$  do  
         $reduced\_paths.append(paths[j])$   
    end  
     $count \leftarrow count + 1, size \leftarrow len(reduced\_paths)$   
end  
if  $size \geq quorum$  then  
     $paths \leftarrow reduced\_paths$   
end  
return  $paths$ 
```

Performing clustering does not guarantee feature reduction, but there is a probability of an unanticipated reduction of the feature set. This procedure attempts to minimise the number of paths at least at the quorum. Unlike the association rules method, which may not accomplish to reduce features, clustering will significantly reduce the number of paths. By the end of the reduction process through clustering, random selection is applied to the paths to obtain the acceptable minimum number of paths, in case of reduction via clustering has not reached the quorum.

3.4 Handling Categorical Features

It is possible, even expected, to deal with a dataset containing categorical features. Of course, a transformation through OneHot or Ordinal [40] encoding is used to make good use of these data. This transformed type of information will then be acceptable from machine learning systems. *But is there any harm to interpretability caused by the use of encoding methods?* Sadly, yes! Using ordinal encoding will transform a feature like $country=[GR, UK, \dots]$ to $country=[0, 1, 2, \dots]$. As a result we lose the intelligibility of the feature. On the other hand, using OneHot encoding will increase dramatically the amount of features leading to over-length and incomprehensible interpretations by transforming the feature $country$ to $country_GR=[0, 1]$, $country_UK=[0, 1]$, and so forth. Since the encoding transformations are part of the feature engineering and are not invariable, we cannot construct a fully automated process to inverse transform the features into human interpretable forms within the interpretations.

Nevertheless, LionForests provides two automated encoding processes using either OneHot or Ordinal encoding and their inverse transformation for the interpretation extraction. Feature-ranges of Ordinal encoded data transform like $(1 \leq country \leq 2) \rightarrow (country=[UK, FR])$, while feature-ranges of OneHot encoded data $(0.5 \leq country_UK \leq 1) \rightarrow (country=UK)$ and feature-ranges like $(0 \leq country_GR \leq 0.49)$ are removed. The excluded OneHot encoded features for the categorical feature should appear to the user as possible alternative values. If a feature reduction method reduces one of the encoded OneHot features, it will not appear in the feature's list of alternative values, but in the list of values that do not influence the prediction. For this reason, the categorical features will appear in the interpretations with a

notation 'c' like ' $categorical_feature^c = value$ '. Depending on the application and the interpretation, the user will be able to request a list of alternative values or will be able to simply hover over the feature to reveal the list. Section 4.3.3 is showcasing transformations of OneHot encoded features, as well as one example of a OneHot encoded feature's alternative values list.

3.5 Interpretation Composition

These processes are part of the LionForests technique, which, in the end, produces an interpretation in the form of a feature-range rule. Lastly, LionForests combines the ranges of the features in the reduced feature set to a single natural language rule. The order of appearance of the feature ranges in the rule is determined by using a global interpretation method, such as the SHAP TreeExplainer [27] or the Scikit [33] random forest model's built-in *feature importance* attribute, for smaller and larger datasets, respectively. One notable example of an interpretation is the following:

$$\text{'if } 0 \leq f_1 \leq 0.5 \text{ and } -0.5 \leq f_3 \leq 0.15 \text{ then class_A'}. \quad (5)$$

We are able to interpret this feature-range rule like that: "As long as the value of the f_1 is between the ranges 0 and 0.5, and the value of f_3 is between the ranges -0.5 and 0.15, the system will classify this instance to class A. If the value of f_1 , f_3 or both, surpass the limits of their ranges then the prediction may change. Note that the features are ranked through their influence". This type of interpretation is comprehensible and human readable. Thus, if we manage to keep them shorter, then they could be an ideal way to explain a random forests model. A way to encounter an over-length rule could be to hide the last n feature-ranges, which they will be the least important due to the ordering process. At the same time, users will have the ability to expand their rules to explore the feature-ranges. However, we do not completely exclude those features, but we are only hiding them because otherwise, we would effect the correctness of both the explanation and the prediction. Such example is showcased in Section 4.3.3.

4 EXPERIMENTAL RESULTS

We first discuss the setup of our experiments. Then, we present quantitative results, followed by qualitative results for the explanation of particular instances.

4.1 Experimental Setup

The implementation of LionForests as well as of all the experiments in this paper is available in the LionLearn repository at GitHub³.

Our experiments were conducted on the following three tabular binary classification datasets: Banknote authentication [14], Heart (Statlog) [14] and Adult Census [25]. The top part of Table 1 shows the number of instances and features of each dataset. Particularly, Adult Census contains 6 numerical and 8 categorical features. The eight categorical features are transformed through OneHot encoding into 74 numerical, resulting in a total of 80 features.

We used the RandomForestClassifier from Scikit-learn [33] and the MinMaxScaler with feature range $[-1, 1]$. In order to work with optimised models, a 10-fold cross-validation grid search was carried out on each dataset, using the following set of parameters and values: max_depth {1, 5, 7, 10}, $max_features$ {'sqrt', 'log2', '75%',

³ <https://github.com/intelligence-csd-auth-gr/LionLearn>

None⁴}, min_samples_leaf {1, 2, 5, 10, 10%}, bootstrap {True, False}, n_estimators {10, 100, 500, 1000}. The parameters' values that achieved the best F_1 score, along with the score itself are shown in the middle and bottom part of Table 1, respectively.

	Banknote	Heart (Statlog)	Adult Census
instances	1372	270	48842
features	4	13	14 (80)
max depth	10	5	10
max features	0.75	'sqrt'	'sqrt'
min sample leaf	1	5	1
bootstrap	True	False	False
estimators	500	500	100
F_1 %	99.43	81.89	88.71

Table 1. For each of the three datasets: main statistics (top), optimal parameter values of the Random Forest (middle), corresponding F_1 score (bottom)

4.2 Quantitative Results

For each dataset, we train one random forest model, using all data, with the parameters shown in Table 1. Then we apply LionForests to all instances of each dataset and report the mean feature and path reduction in Table 2.

In terms of path reduction, we notice that among the three reduction techniques, random selection consistently leads to the best results across all datasets, achieving an average reduction of 45.69%. Clustering is the second-best method with only slightly worse results in the first two datasets, but much worse results in the Adult Census dataset, achieving an average of 41.94% reduction. Association rules is the worst technique, achieving zero reduction in the Adult Census dataset and an average of 19.90% reduction. Combining random selection with other techniques does not lead to improved results.

With respect to feature reduction, we find that the association rules strategy leads to the best results in the first two datasets, reaching an average reduction of 26.04%. The other two techniques achieve negligible reduction in these two datasets. In banknote authentication combining the three methods improves slightly the reduction from 30.70% to 30.85%. In Adult Census on the other hand, association rules achieve zero reduction of features similarly to paths. The best result in this dataset is achieved by random selection, followed closely by clustering. Combining these two techniques improves slightly the reduction to 10%.

The weak performance of association rules in Adult Census is related to the small number of estimators (100) and the huge number of features (80). In particular, each of the one hundred estimators can have a maximum of eight features, as resulted from the grid search (max features = 'sqrt'). Thus, the overlapping features among the

different paths are fewer, and by extension, techniques relying on feature selection, like association rules, cannot perform the feature reduction, as well as path reduction. However, LionForests is an ensemble of techniques, and we have revealed with these quantitative experiments the importance of each part. We may infer that the LionForests strategy is considerably effective in reducing both features and paths.

4.3 Qualitative Results

LionForests technique provides consistent and robust rules which are more indisputable from other interpretations because they are more compact, have broader ranges for the features, while at the same time present categorical data in a human-comprehensible form. In this section, we provide an example from each dataset to demonstrate how random forests models can be interpreted efficiently.

4.3.1 Banknote Authentication

We observe that the complete methodology, incorporating association rules, clustering and random selection reduction, achieves the highest performance on both feature and path reduction for the first dataset, Banknote Authentication. We also provide an example of a pair of explanations (1) without and (2) with LionForests:

1. 'if $2.4 \leq \text{variance} \leq 6.83$ and $-3.13 \leq \text{skew} \leq -2.30$ and $1.82 \leq \text{curtosis} \leq 2.13$ and $-0.64 \leq \text{entropy} \leq 0.73$ then fake banknote'
2. 'if $2.4 \leq \text{variance} \leq 6.83$ and $-1.60 \leq \text{curtosis} \leq 17.93$ then fake banknote'

The reduced rule (2) is smaller than the original by two features. In addition, the "curtosis" feature has a wider range. The instance has a value of 1.92 for the "curtosis" feature, and in the original rule this value is marginal for the very narrow range $1.82 \leq \text{curtosis} \leq 2.13$ indicating that a small change may lead to a different outcome, but this is not the case for the reduced rule as well. In addition, changing the skew value from -2.64 to -4 , which is outside the range of the feature in the original rule, will not change the prediction and will produce the same reduced range rule. We observe the same result when we change the value of the "entropy" feature, as well as when we tweak both "skew" and "entropy".

4.3.2 Heart Disease

Again, with LionForests we achieve both higher feature and path reduction ratios. There are thirteen features included in this particular dataset and, as a result, the interpretations may have a total of thirteen features. In fact, in this case the reduction of features is essential to provide comprehensible interpretations. We choose an example and we present the original rule (1) and the reduced rule (2):

⁴ None = all features

Reduction Technique			Reduction %						Mean Reduction %	
Association Rules	Clustering	Random Based	Feature	Path	Feature	Path	Feature	Path	Feature	Path
✓	✓	✓	$30.85 \pm 5.1e^{-2}$	$49.47 \pm 5.5e^{-15}$	21.37 ± 0.0	43.41 ± 0.0	$10.03 \pm 1.5e^{-2}$	$44.18 \pm 5.5e^{-15}$	20.75	45.69
-	✓	✓	$2.15 \pm 1.8e^{-1}$	$49.47 \pm 5.5e^{-15}$	$8.6e^{-3} \pm 1.3e^{-2}$	43.41 ± 0.0	$10.03 \pm 1.5e^{-2}$	$44.18 \pm 5.5e^{-15}$	4.06	45.69
✓	-	✓	30.70 ± 0.0	$49.47 \pm 5.5e^{-15}$	21.37 ± 0.0	43.41 ± 0.0	$9.11 \pm 1.4e^{-2}$	$44.18 \pm 5.5e^{-15}$	20.39	45.69
✓	✓	-	$30.84 \pm 4.4e^{-2}$	$48.56 \pm 2.4e^{-2}$	21.37 ± 0.0	$42.88 \pm 2.2e^{-2}$	$7.82 \pm 1.4e^{-2}$	$36.26 \pm 2.6e^{-2}$	20.01	42.57
✓	-	-	30.70 ± 0.0	27.02 ± 0.0	21.37 ± 0.0	32.67 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	17.36	19.90
-	✓	-	$2.12 \pm 1.3e^{-1}$	$47.57 \pm 3.6e^{-2}$	$5.7e^{-3} \pm 1.1e^{-2}$	$41.98 \pm 7.4e^{-2}$	$7.82 \pm 1.4e^{-2}$	$36.26 \pm 2.6e^{-2}$	3.32	41.94
-	-	✓	0.0 ± 0.0	$49.47 \pm 5.5e^{-15}$	$2.9e^{-3} \pm 8.6e^{-3}$	43.41 ± 0.0	$9.11 \pm 1.4e^{-2}$	$44.18 \pm 5.5e^{-15}$	3.04	45.69
			Banknote Authentication		Heart Disease		Adult Census			

Table 2. Feature and path reduction ratios on the three datasets. ✓ (#) the order of the applied techniques

1. 'if $6.5 \leq \text{reversible defect} \leq 7.0$ and $3.5 \leq \text{chest pain} \leq 4.0$ and $0.0 \leq \text{number of major vessels} \leq 0.5$ and $1.55 \leq \text{oldpeak} \leq 1.7$ and $0.5 \leq \text{exercise induced angina} \leq 1.0$ and $128.005 \leq \text{maximum heart rate achieved} \leq 130.998$ and $1.5 \leq \text{the slope of the peak exercise} \leq 2.5$ and $\text{sex}^c = \text{Male}$ and $184.999 \leq \text{serum cholestoral} \leq 199.496$ and $29.002 \leq \text{age} \leq 41.497$ and $0.0 \leq \text{resting electrocardiographic results} \leq 0.5$ and $119.0 \leq \text{resting blood pressure} \leq 121.491$ and $0.0 \leq \text{fasting blood sugar} \leq 0.5$ then presence'
2. 'if $6.5 \leq \text{reversible defect} \leq 7.0$ and $3.5 \leq \text{chest pain} \leq 4.0$ and $0.0 \leq \text{number of major vessels} \leq 0.5$ and $1.55 \leq \text{oldpeak} \leq 1.7$ and $0.5 \leq \text{exercise induced angina} \leq 1.0$ and $128.005 \leq \text{maximum heart rate achieved} \leq 133.494$ and $1.5 \leq \text{the slope of the peak exercise} \leq 2.5$ and $184.999 \leq \text{serum cholestoral} \leq 199.496$ and $119.0 \leq \text{resting blood pressure} \leq 121.491$ then presence'

The reduced rule is shorter than the original rule by four features. In addition, we observe that the "maximum heart rate achieved" feature has a wider range in the reduced rule. Changing the "sex" value from 'Male' (1) to 'Female' (0) did not change the reduced rule at all. We tweak "age" value from 35 to 15 and again the reduced rule remains the same. Thus, features like "age", "sex", "resting electrocardiographic results" and "fast blood sugar", cannot influence the prediction.

4.3.3 Adult Census

Finally, we present a pair of explanations produced by LionForests for an instance of Adult Census, without (1) and with (2) reduction:

1. 'if $\text{marital status}^c = \text{Married}$ and $\text{sex}^c = \text{Female}$ and $\text{education}^c = \text{HS grad}$ and $\text{workclass}^c = \text{Private}$ and $94721 \leq \text{fnlwt} \leq 161182$ and $47 \leq \text{age} \leq 53$ and $15 \leq \text{hours per week} \leq 25$ and $\text{native country}^c = \text{Jamaica}$ and [other 2 feature-ranges] then income >50K'
2. 'if $\text{marital status}^c = \text{Married}$ and $\text{sex}^c = \text{Female}$ and $\text{education}^c = \text{HS grad}$ and $\text{workclass}^c = \text{Private}$ and $87337 \leq \text{fnlwt} \leq 382719$ and $47 \leq \text{age} \leq 63$ and $15 \leq \text{hours per week} \leq 99$ and $\text{native country}^c = \text{Jamaica}$ and [other 2 feature-ranges] then income >50K'

The reduced rule is thirteen features smaller than the original rule. This is not directly obvious because some of the OneHot categorical features are not presented. For example, only valid features such as "marital_status_Married" and "education_HS_Grad" are presented as described in Section 3.4. Furthermore, we observe that the ranges of "age", "fnlwt" and "hours per week" are broader. Specifically, "age" range from [47, 53] increased to [47, 63], while "hours per week" range from [15, 25] expanded to [15, 99]. Moreover, we can explore the categorical feature's "native country" alternative values. In Table 3, the first list refers to the values that may change the prediction of an instance, while the second list shows the values that cannot affect the prediction. In the original rule this type of information was not available because the non-affecting values (second list) were present.

5 CONCLUSION

Providing helpful explanations is a challenging task. Providing comprehensible and undeniable explanations is even tougher. Since

Possible values of "native_country", which they may affect the prediction	preserve the prediction
'Mexico', 'United-States', 'Canada', 'Philippines', 'England', 'Thailand', 'Japan', 'China', 'Dominican-Republic', 'Germany', 'South', 'Columbia', 'Italy', 'Puerto-Rico', 'Vietnam', 'Cambodia', 'Ireland', 'Taiwan', 'Portugal', 'Laos', 'Yugoslavia', 'Nicaragua', 'Scotland'	'India', 'France', 'El-Salvador', 'Iran', 'Cuba', 'Haiti', 'Guatemala', 'Peru', 'Trinidad&Tobago', 'Honduras'

Table 3. List of values affecting or not the classification of an instance

model-agnostic approaches produce approximations, i.e. the nearest optimal ones but not the optimal explanations, the attempt to indifferently interpret each black-box model will not lead to the desired outcome. In this work, we introduced a model-specific local-based approach for obtaining real interpretations of random forests predictions. Other works [31, 41] attempt to provide explanations of this form, but they do not try to make them more comprehensible, either indisputable. A user may not be familiar with iForest's [41] visualisation tool. Besides, an interpretation containing a lot of features with narrow ranges [31] may lead to incomprehensible and untrustworthy rules. The proposed technique, which we call LionForests, will provide users with small rules as interpretations in natural language, which by widening the feature ranges will be more reliable and trustworthy as well. We use classic unsupervised methods such as association rules and k -medoids clustering to achieve feature and path reduction.

Nevertheless, LionForests is not a complete solution either, since it is not appropriate for tasks of model inspection. For example, if a researcher is working to build a reliable and stable model aiming for the highest performance, a visualisation tool like iForest may be preferred. This approach is the best and easiest way of providing an interpretation to a non-expert user. Lastly, by reducing the number of paths to the quorum to minimise the features and at the same time to increase the features' ranges, the outcome would be a discounted probability of the classification of the instance, which poses questions about the prediction's reliability. This can be counter-attacked by introducing a threshold parameter to the reduction effect, requiring the algorithm to retain at least a specific percentage of the paths.

Future research will explore the impact of tuning parameters, such as the number of estimators or the maximum number of features, on the reduction of features and paths. We also aim to apply LionForests to different models of tree ensembles, rather than random forests, as well as to various datasets and data types. FPGrowth [22], and its variant FPMMax [18], will be tested against the Apriori algorithm. In addition, we will consider the option of adapting LionForests to other tasks such as multi-class or multi-label classification as well as regression. We will also look into the possibility of using the interpretations from LionForests to provide descriptive narratives through counter-examples. Ultimately, by qualitative, human-oriented analysis, we will further examine this promising method to prove its intelligibility.

ACKNOWLEDGEMENTS

This paper is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619, AI4EU Project⁵.

⁵ <https://www.ai4eu.eu>

REFERENCES

- [1] Amina Adadi and Mohammed Berrada, ‘Peeking inside the black-box: A survey on explainable artificial intelligence (xai)’, *IEEE Access*, **6**, 52138–52160, (2018).
- [2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami, ‘Mining association rules between sets of items in large databases’, in *Acm sigmod record*, volume 22, pp. 207–216. ACM, (1993).
- [3] Rakesh Agrawal, Ramakrishnan Srikant, et al., ‘Fast algorithms for mining association rules’, in *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pp. 487–499, (1994).
- [4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek, ‘On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation’, *PloS one*, **10**(7), e0130140, (2015).
- [5] Leo Breiman, ‘Random forests’, *Machine learning*, **45**(1), 5–32, (2001).
- [6] Angela Chen. IBM’s Watson gave unsafe recommendations for treating cancer. <https://cutt.ly/keHQDma>, 2018. Accessed: 2019-11-18.
- [7] Tianqi Chen and Carlos Guestrin, ‘Xgboost: A scalable tree boosting system’, in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM, (2016).
- [8] HA Chipman, EI George, and RE McCulloh, ‘Making sense of a forest of trees’, *Computing Science and Statistics*, 84–92, (1998).
- [9] Samantha Cole. This trippy t-shirt makes you invisible to ai. <https://cutt.ly/FeHQHAA>, 2019. Accessed: 2019-11-18.
- [10] Houtao Deng, ‘Interpreting tree ensembles with intrees’, *International Journal of Data Science and Analytics*, **7**(4), 277–287, (2019).
- [11] Pedro Domingos, ‘Knowledge discovery via multiple models’, *Intelligent Data Analysis*, **2**(1-4), 187–202, (1998).
- [12] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić, ‘Explainable artificial intelligence: A survey’, in *2018 41st International convention on information and communication technology, electronics and micro-electronics (MIPRO)*, pp. 0210–0215. IEEE, (2018).
- [13] Mengnan Du, Ninghao Liu, and Xia Hu, ‘Techniques for interpretable machine learning’, *arXiv preprint arXiv:1808.00033*, (2018).
- [14] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [15] Niall Firth. Apple card is being investigated over claims it gives women lower credit limits. <https://cutt.ly/oeGYC5>, 2019. Accessed: 2019-11-18.
- [16] Aaron Fisher, Cynthia Rudin, and Francesca Dominici, ‘All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance’, *arXiv preprint arXiv:1801.01489*, (2018).
- [17] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin, ‘Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation’, *Journal of Computational and Graphical Statistics*, **24**(1), 44–65, (2015).
- [18] Gösta Grahne and Jianfei Zhu, ‘Efficiently using prefix-trees in mining frequent itemsets’, in *FIMI*, volume 90, (2003).
- [19] Stefan Th Gries, ‘On classification trees and random forests in corpus linguistics: Some words of caution and suggestions for improvement’, *Corpus Linguistics and Linguistic Theory*, (2019).
- [20] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi, ‘A survey of methods for explaining black box models’, *ACM computing surveys (CSUR)*, **51**(5), 93, (2019).
- [21] Tamer Hailasilassie, ‘Rule extraction algorithm for deep neural networks: A review’, *arXiv preprint arXiv:1610.05267*, (2016).
- [22] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao, ‘Mining frequent patterns without candidate generation: A frequent-pattern tree approach’, *Data mining and knowledge discovery*, **8**(1), 53–87, (2004).
- [23] Satoshi Hara and Kohei Hayashi, ‘Making tree ensembles interpretable: A bayesian model selection approach’, in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, eds., Amos Storkey and Fernando Perez-Cruz, volume 84 of *Proceedings of Machine Learning Research*, pp. 77–85, Playa Blanca, Lanzarote, Canary Islands, (09–11 Apr 2018). PMLR.
- [24] Leonard Kaufman and Peter J Rousseeuw, ‘Clustering by means of medoids. statistical data analysis based on the l1 norm’, *Y. Dodge, Ed.*, 405–416, (1987).
- [25] Ron Kohavi, ‘Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid’, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, p. to appear, (1996).
- [26] Zachary C. Lipton, ‘The mythos of model interpretability’, *Commun. ACM*, **61**(10), 36–43, (September 2018).
- [27] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee, ‘Explainable ai for trees: From local explanations to global understanding’, *arXiv preprint arXiv:1905.04610*, (2019).
- [28] Scott M Lundberg and Su-In Lee, ‘A unified approach to interpreting model predictions’, in *Advances in Neural Information Processing Systems 30*, eds., I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 4765–4774, Curran Associates, Inc., (2017).
- [29] Laurens van der Maaten and Geoffrey Hinton, ‘Visualizing data using t-sne’, *Journal of machine learning research*, **9**(Nov), 2579–2605, (2008).
- [30] Ioannis Mollas, Nikolaos Bassiliades, and Grigorios Tsoumakas, ‘Li-onets: Local interpretation of neural networks through penultimate layer decoding’, in *ECML PKDD 2019 AIMLAI XKDD Workshop. Wäijrzburg, Germany*, (2019).
- [31] Alexander Moore, Vanessa Murdock, Yaxiong Cai, and Kristine Jones, ‘Transparent tree ensembles’, in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1241–1244. ACM, (2018).
- [32] Oxford University Press (OUP). Lexico.com, 2019.
- [33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ‘Scikit-learn: Machine learning in Python’, *Journal of Machine Learning Research*, **12**, 2825–2830, (2011).
- [34] General Data Protection Regulation, ‘Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46’, *Official Journal of the European Union (OJ)*, **59**(1-88), 294, (2016).
- [35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, ‘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. ACM, (2016).
- [36] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, ‘Anchors: High-Precision Model-Agnostic Explanations’, in *Thirty-Second AAAI Conference on Artificial Intelligence*, (2018).
- [37] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller, ‘Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models’, *arXiv preprint arXiv:1708.08296*, (2017).
- [38] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas, ‘Interpretable predictions of tree-based ensembles via actionable feature tweaking’, in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 465–474. ACM, (2017).
- [39] James Vincent. This colorful printed patch makes you pretty much invisible to ai. <https://cutt.ly/TeHQJHU>, 2019. Accessed: 2019-11-18.
- [40] Alexander Von Eye and Clifford C Clogg, *Categorical variables in developmental research: Methods of analysis*, Elsevier, 1996.
- [41] Xun Zhao, Yanhong Wu, Dik Lun Lee, and Weiwei Cui, ‘iforest: Interpreting random forests via visual analytics’, *IEEE transactions on visualization and computer graphics*, **25**(1), 407–416, (2018).
- [42] Yichen Zhou and Giles Hooker, ‘Interpreting models via single tree approximation’, *arXiv preprint arXiv:1610.09036*, (2016).
- [43] Andreas Ziegler and Inke R König, ‘Mining data with random forests: current options for real-world applications’, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **4**(1), 55–63, (2014).