LionForests: Local Interpretation of Random Forests through Path Selection

Ioannis Mollas, Grigorios Tsoumakas, Nick Bassiliades[1]

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 sequence of actions 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 explanation 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. However, 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 therefore 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 explanation of black box models [1, 12, 13, 20, 21, 26, 37].

Methods for explaining machine learning models are categorized, 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 explanation 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 US [2].

Another important dimension along which IML methods are categorized 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 models. Methods of the former category have wider applicability, but they inevitably only approximately explain the models they are applied to [1].

This work focuses on the latter category of methods and in particular on tree ensembles [5, 7], which are very effective in several applications involving tabular and time-series data [43]. Inside the black box of a tree ensemble hide a number of transparent decision trees. We hypothesize that using smart techniques, we could infer explanations for the decisions of tree ensembles.

iForest [41], a global and local explanation system of random forests, provides insights for a decision through a visualisation tool. Such an approach lacks the ability to reveal the rationale behind the decision without a complex visual explanation, presenting inaccessible explanations to non-expert users. In the same time, the system demands user interaction in order to construct the local explanations. Another instance-level explanation technique for random forests proposed by Moore et al. [31], which is more approachable to users than iForest, produces an explanation in the form of a list of features with their ranges, accompanied by an influence metric. However, if the list of features is extensive, and the ranges are very narrow, the explanations can be characterised as unreliable and untrustworthy. Finally, both methods are handling categorical data improperly, providing none human-intelligible information about them.

This paper introduces a local-based model-specific approach for explaining individual predictions of a random forest via a rule. Unsupervised techniques like association rules [2] and k-medoids clustering [24] using a path-oriented similarity metric are the tools for path and feature selection. The ultimate goal is to reduce the number of features and broaden the feature-ranges producing more robust and indisputable explanations. Additionally, the categorical features are handled in an elegant way, providing intelligible information about them throughout the rules. Finally, the constructed rule will be presented as the explanation, if its length is acceptable to be comprehensible. Otherwise, additional processing will be held to form an acceptable explanation. We call this technique "LionForests" (Local Interpretation Of raNdom FORESts through paTh Selection) and we use its path and feature selection ability to process the explanations in order to make them more comprehensible.

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

[2] ECOA 15 U.S. Code Âğ1691 et seq.

2 RELATED WORK

If it is preferable to understand the core processes of a tree ensemble model, due to its complicated nature, an approximation will be used to display its structure. The single-tree approximation is a global-based model-specific method, which is highly studied by many researchers [10, 11, 23, 42] to interpret tree ensembles, but this method, as its name implies, approximates the performance of the model it seeks to explain. However, this approach is extremely problematic and criticised, because it is not feasible to summarise a complex model like tree ensembles to a single simple tree, as supported by Stefan Th. Gries [19].

Another interpretation techniques family about black-box models, as well as 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 model-agnostic techniques. 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].

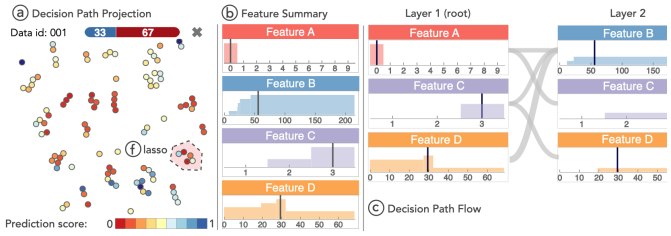

Figure 1. Template of visualisation tool of iForest [41]

An additional approach on interpreting tree ensembles focuses on clustering the trees of an ensemble using a tree dissimilarity metric [8]. This work is very close to the idea in iForest [41], where they use a path distance metric to project an instance's paths to a two-dimensional space using TSNE [29]. The path distance metric they propose considers distant two paths in two cases. If a feature exists in only one out of the two paths the distance between those two paths is increasing. Moreover, if a feature exists in both paths, the distance between them is increasing according to the non-common ranges of the feature on the paths divided in 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. In iForest, 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 explanation.

Lastly, Moore et al. [31], they attempt more accurately than iForest, to interpret tree ensembles in instance-level (local-based technique) providing as explanation [Figure 2] a set of features with their

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. Template of explanation of Moore et al. [31]

ranges, ranked based on their contribution. Thus, the interpretation process consists of two parts. Firstly, they calculate the influence of a feature j on the prediction for a given instance x, this influence later will be used for the ranking process. To achieve this, a node per tree monitoring process is applied to find the aggregated influence of all features for a specific instance's prediction. The second step is to find the narrowest range across all trees for every feature.

3 OUR APPROACH

Our objective is to provide local explanations of random forest 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:

$$f(x_i) = \frac{1}{T}\sum_{t=0}^{T} f_t(x_i) \tag{1}$$

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 is assigned to class $c_j \in C$.

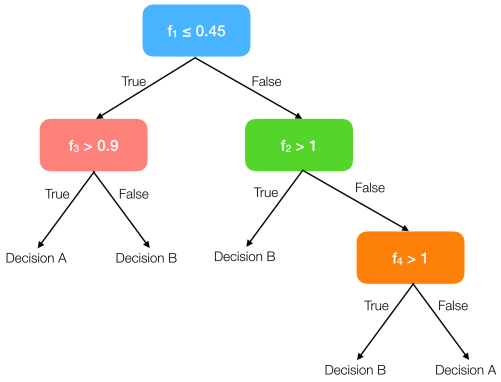

Figure 3. A simple decision tree classifier with 4 features

Each decision tree $t \in T$ has nodes and leaves, and by degrading its structure, we are able to derive a set P of paths. Therefore, every instance can be classified with one of these paths. A path $p \in P$ is a conjunction of conditions, and the conditions are features and values with relations $\leq$ and $>$. For example, a path from the simple tree on Figure 3 can be expressed like this: 'if $f_1 \leq 0.45$ and $f_3 > 0.9$ then Decision A'. Thus, each path p is expressed as a set:

$$p = \{f_i \boxtimes v_j | f_i \in F, v_j \in \Re, \boxtimes \in \{\leq, >\}\} \tag{2}$$

By extracting the paths by the majority of trees voted for an instance's prediction, we have our primary source of information to build an explanation. For example, suppose we have a random forests

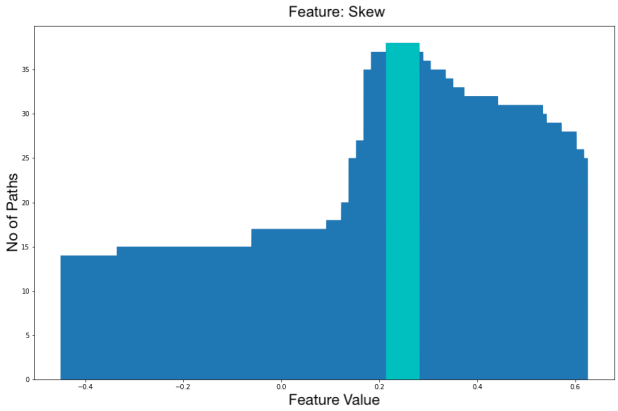

Figure 4. Stacked area plot of all paths containing 'skew' feature from Banknote Dataset

model with N trees. For a specific instance x, M of those trees vote for class A of a binary problem of two classes (class A and B). Then, for these M trees, we can extract their decision paths P. Each path contains conditions about some features. Collecting all the conditions from all the paths for one feature at a time, we can draw the stacked area and bars plots (Figures 4 and 6). These plots are showcasing that all paths containing the feature displayed have at least one conjunction about this feature with a range equal or larger than the intersection of ranges of all paths (cyan/light grey section in Figure 4). 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

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 this intersection range, the decision paths are not going to change. Summarising the aforementioned, an explanation can have this shape:

'if $0.47 \leq f_1 \leq 0.6$ and $-0.52 \leq f_2 \leq -0.5$ and ... then Class_A'

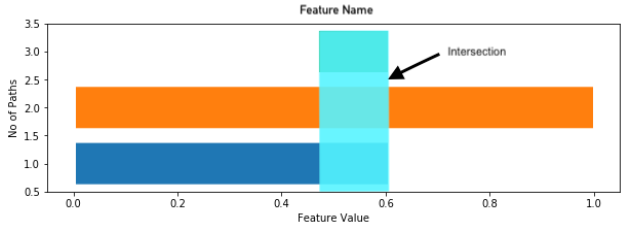

Figure 5. Bars plot of the simple example

But having as many paths as possible voting to class A, we are facing the following two problems:

1. A lot of paths will lead to more features, and by extension to an unintelligible explanation and a dissatisfied user.

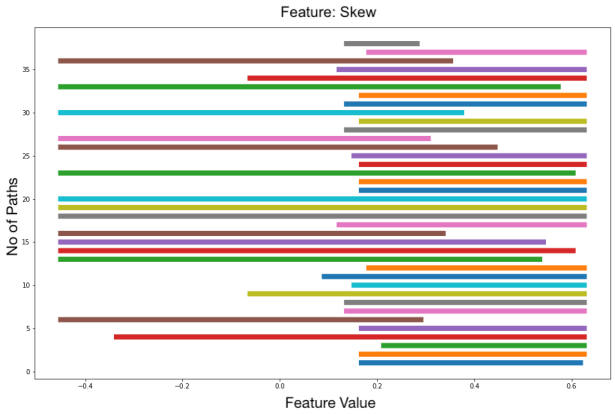

Figure 6. Bars plot of all paths containing 'skew' feature from Banknote Dataset

2. A lot of paths will lead to a small, strict and very specific range. For example, f_1 instance's value was 0.5 and the intersection range of all paths for this feature happens to be $0.47 \leq f_1 \leq 0.6$, while the feature range is [-1,1]. Such strict range will lead to a negative impression about the model, which will be considered as unstable and unreliable. Thus, a wider range will be less refutable.

We propose LionForests, a framework for interpreting random forests models in the instance level. The algorithm is a series of actions: reduction through association rules, clustering and random based, which limits the number of paths keeping not less than the quorum of the total number of trees, in order to maintain the same prediction about an instance.

> *quorum |ˈkwɔːrəm|: 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].*

3.1 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 $I = \{i_1, i_2, \ldots, i_n\}$. Each dataset contains sets of items, called itemsets $T = \{t_1, t_2, \ldots, t_m\}$, 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 dataset. The dataset T 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 how many paths can be true with these features. If there

are 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, most probably will 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:

$$p = \{f_i \boxtimes v_j | f_i \in F', v_j \in \Re, \boxtimes \in \{\leq, >\}\}. \quad (3)$$

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 fulfilled 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.

Algorithm 1: Path similarity metric

input : Paths p_i, p_j, $feature\_names$, $min\_max\_feature\_values$
return: Similarity s_{ij}
$s_{ij} \leftarrow 0$
for *f in* $feature\_names$ **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
      $inter \leftarrow min(u_i, u_j) - max(l_i, l_j)$
      $union \leftarrow max(u_i, u_j) - min(l_i, l_j)$
      if $union \neq 0$ **then**
        $s_{ij} \leftarrow s_{ij} + inter/union$
      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 $s_{ij}/len(feature\_names)$

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.

3.2 Reduction through Clustering

By applying association rules to reduce the feature set, and consequentially the number of paths, failure is probable. In that case, we apply a second reduction technique based on clustering. Clustering is another set of unsupervised techniques in machine learning, apart from 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. Analysing this algorithm, if both paths do not contain any conjunction about a feature, that makes them similar, thus the similarity is increasing. When, both paths contain conjunctions about a feature, then it increases the similarity depending on the intersection of the two ranges normalised by the union of the two ranges.

In Algorithm 2, we calculate the k medoids and their groups 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 groups. Then, we collect paths from the larger groups into a list, until we acquire at least a quorum. By summing larger groups first, the possibility of feature reduction is increasing, because the paths inside a group tend to be more similar between them, hence they may have fewer irrelevant features.

Algorithm 2: Paths reduction through k-medoids clustering

input : Similarity Matrix $similarity\_matrix$, No of Estimators $no\_of\_estimators$, Paths $paths$
return: Paths $paths$
$quorum\_of\_estimators \leftarrow no\_of\_estimators/2 + 1$
if $no\_of\_estimators < 5$ **then**
  $no\_of\_medoids \leftarrow no\_of\_estimators$
end
if $no\_of\_estimators \geq 100$ **then**
  $no\_of\_medoids \leftarrow \lceil quorum\_of\_estimators * 3/22 \rceil$
end
$m \leftarrow kmedoids(similarity\_matrix, no\_of\_medoids)$
$sorted\_m \leftarrow sort\_by\_key(m, descending = True)$
$count \leftarrow 0, size \leftarrow 0, reduced\_paths \leftarrow []$
while $size < quorum\_of\_estimators$ *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 $size \geq quorum\_of\_estimators$ **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 association rules method, which may not accomplish to reduce the features, clustering is going to significantly limit the number of paths. By the end of the reduction process through clustering, random sampling is applied to the paths to obtain the acceptable minimum number of paths, in case reduction via clustering did not reach the quorum.

3.3 Handling Categorical Features

It is possible, even expected, a dataset to contain categorical features. Of course, in order to make good use of these data, a transformation through OneHot or Ordinal [40] encoding is applied. Then this transformed kind of information will be acceptable from the machine learning systems. *But is there any harm of explainability by the use of encoding methods?* Suddenly, yes! Using ordinal encoding will transform a feature like "$country$" $=$ ["GR", "UK", "US", ...] 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 overlength and incomprehensible explanations by transforming the feature "$country$" to "$country\_GR$" $=$ $[0, 1]$, "$country\_UK$" $=$ $[0, 1]$, and so on. Due to the fact that the encoding transformations are part of feature engineering, and it is not invariable, we can not create an entirely automated process to inverse transform the features to human interpretable shapes inside the explanations.

However, LionForests provide two automated encoding processes using either OneHot or Ordinal encoding and their inverse transformation for the explanation extraction. Feature-ranges of Ordinal encoded data transform like $(1 \leq country \leq 2) \rightarrow (country =$ ["UK", "US"]), while feature-ranges of OneHot encoded data $(0.5 \leq country\_UK \leq 1) \rightarrow (country =$ "UK") and feature-ranges like $(0 \leq country\_US \leq 0.499)$ are removed. The removed OneHot encoded features will appear to the user as possible alternative values for the categorical feature. If one OneHot encoded feature is reduced through the feature reduction process, then it will not appear to the list of alternative values for the feature of the user. For this reason, the categorical features will appear in the explanations with a notation 'c' like '$categorical\_feature^c = value$'. Depending on the application and the explanation's representation, a user will be able to ask for the list of the alternative values or he will be able to just hover above the feature to reveal the list. Section 4.4 is showcasing transformations of OneHot encoded features, as well as one example of a OneHot encoded feature's alternative values list.

3.4 Explanation Extraction

The above processes are part of LionForests technique, which is finally producing the explanation in the form of a feature-range rule. Lastly, before the algorithm generates the explanation, it ranks the appearance of the features in the rule based on the feature importance computed through either SHAP TreeExplainer [27], when dealing with small datasets, or the feature importance build-in attribute of Scikit's [33] random forests model, when dealing with larger datasets. A notable example of an explanation is the following:

'if $0 \leq f_1 \leq 0.5$ and $-0.5 \leq f_3 \leq 0.15$ then class A'.

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 explanations are comprehensible and human readable. Thus, if we manage to keep them in small number of features, 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 ranking process. In the same time, users will have the ability to expand their rules to explore all the feature-ranges. Such example is showcased in Section 4.4.

4 EXPERIMENTS

We conducted a set of experiments using LionForests with three different tabular datasets to assess the validity of this research. LionForests' code and evaluation experiments are available at GitHub repository "LionLearn"[3].

4.1 Setup

For this set of experiments, we used the RandomForestClassifier from Scikit-learn [33] and the MinMaxScaler with feature range $[-1, 1]$. For each experiment, a 10-fold cross-validation grid search was executed with the following set of parameters:

- max_depth: 1, 5, 7 or 10
- max_features: 'sqrt', 'log2', 75% or None[4]
- min_samples_leaf: 1, 2, 5, 10 or 10%
- bootstrap: True or False
- n_estimators: 10, 100, 500 or 1000

The scoring metric of the grid search was the f1-score. By finding the best classifier for each dataset, we trained the model to the whole dataset, and we computed the mean average feature and path reduction throughout all instances using our method and some variations.

As already mentioned, three different datasets were utilised to examine this technique. Below, short descriptions of them are provided:

1. Banknote Authentication [14]: This dataset contains representations of real or fake banknotes. It has 4 features, 1372 instances and 2 classes (real or fake banknote).
2. Statlog (Heart) [14]: This dataset describes a heart disease and it contains 13 features, 270 cases and 2 classes (absence or presence of disease).
3. Adult Census [25]: Adult Census dataset holds 14 features (6 numerical and 8 categorical features), 48842 instances and 2 classes (income over or under 50K per year). However, after feature engineering we got 80 numerical features.

4.2 Banknote Dataset

The result of the grid search on the banknote dataset, found the best classifier with f1-score 99.43% and the below set of parameters:

- max_depth: 10
- max_features: 0.75
- min_samples_leaf: 1
- bootstrap: True
- n_estimators: 500

[3] `https://github.com/iamollas/LionLearn`
[4] None = all features

With this classifier, we computed the feature and path reduction ratios [Table 1] using LionForests and its variations. Applying clustering and/or random based reduction methods without association rules, we are unable to reduce significantly the number of features.

Reduction Technique			Reduction %	
Association Rules	**Clustering**	**Random Based**	**Feature**	**Path**
✓	✓	✓	**30.85%**	**49.47%**
-	✓	✓	2.06%	**49.47%**
✓	-	✓	30.70%	**49.47%**
✓	✓	-	**30.85%**	48,57%
✓	-	-	30.70%	27.02%
-	✓	-	2.06%	47,59%
-	-	✓	0%	**49.47%**

Table 1. Feature and path reduction ratios on banknote dataset

Here is an example of a pair of explanations (1) without and (2) with LionForests:

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

The reduced rule has two features less than the original. Moreover, the feature "curtosis" has a broader range. The instance has the value 1.92 for the "curtosis" feature, and in the original rule this value is marginal on the very narrow range $1.82 \leq curtosis \leq 2.13$ suggesting that a small change may lead to different results, but this is not the case for the reduced rule too. Additionally, changing feature's "skew" value from -2.64 to -4, which is out of the feature's range in the original rule, will not change the prediction and will produce the same reduced feature-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 Heart Disease

Executing the grid search on the heart disease dataset, the best classifier in the terms of f1-score using 10-fold cross-validation, which was 81.89%, had the following set of parameters:

- max_depth: 5
- max_features: 'sqrt'
- min_samples_leaf: 5
- bootstrap: False
- n_estimators: 500

Reduction Technique			Reduction %	
Association Rules	**Clustering**	**Random Based**	**Feature**	**Path**
✓	✓	✓	**21.37%**	**43.41%**
-	✓	✓	0.02%	**43.41%**
✓	-	✓	21.36%	**43.41%**
✓	✓	-	**21.37%**	42.23%
✓	-	-	21.36%	32.67%
-	✓	-	0.02%	41.93%
-	-	✓	0%	**43.41%**

Table 2. Feature and path reduction ratios on heart disease dataset

Like before, the feature and path reduction ratios are computed [Table 2]. Once again with LionForests, we achieve both the higher feature and path reduction ratios. In this specific dataset, there are thirteen features available for the model. Thus, the explanation can have a maximum of thirteen features. Hence, feature reduction is more than necessary to provide comprehensible explanations. Once again, we choose an example and we present the original rule (1) and the reduced rule produced by LionForests (2):

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

The reduced rule (2) is four features smaller, than the original. We observe that the feature "maximum heart rate achieved" has more broad ranges. Changing the "sex" value from 'Male' (1) to 'Female' (0) did not change the reduced rule at all. We tweak "age" from 35 to 15 and once again the reduced rule remains the same. Thus, features like "age", "sex", "resting electrocardiographic results" and "fast blood sugar", they can not influence the prediction.

4.4 Adult Census

Running the 10-fold cross-validation grid search on Adult Census dataset the f1-score was 88.71%. For the best RandomForestClassifier we acquired the following set of parameters of :

- max_depth: 10
- max_features: 'sqrt'
- min_samples_leaf: 1
- bootstrap: False
- n_estimators: 100

Reduction Technique			Reduction %	
Association Rules	**Clustering**	**Random Based**	**Feature**	**Path**
✓	✓	✓	**10.02%**	**44.18%**
-	✓	✓	**10.02%**	**44.18%**
✓	-	✓	9.48%	**44.18%**
✓	✓	-	7.82%	36.25%
✓	-	-	0.00%	0.00%
-	✓	-	7.82%	36.25%
-	-	✓	9.48%	**44.18%**

Table 3. Feature and path reduction ratios on adult census dataset

In Table 3, we can see that having less estimators and plenty of features renders reduction through association rules useless. However, LionForests method is an ensemble of different techniques rather than feature and path reduction through association rules because it utilises reduction via clustering and random based selection. We

present a pair of explanations for an instance of this dataset without (1) and with (2) LionForests:

1. 'if $marital\_status^c = Married$ and $sex^c = Female$ and $education^c = HS\_grad$ and $workclass^c = Private$ and $94721 \leq fnlwgt \leq 161182$ and $47 \leq age \leq 53$ and $15 \leq hours\_per\_week \leq 25$ and $native\_country^c = Jamaica$ and [other 2 feature-ranges] then income >50K'
2. 'if $marital\_status^c = Married$ and $sex^c = Female$ and $education^c = HS\_grad$ and $workclass^c = Private$ and $87337 \leq fnlwgt \leq 382719$ and $47 \leq age \leq 63$ and $15 \leq hours\_per\_week \leq 99$ and $native\_country^c = Jamaica$ and [other 2 feature-ranges] then income >50K'

The reduced rule is thirteen features smaller than the original. But this is not visible because some OneHot categorical features are not presented. For example, we present only the valid category, as described in section 3.3, for features like "marital_status_Married" and "marital_status_Separated". Despite this, we observe that feature-ranges like "age", "fnlwft" and "hours_per_week" have broader ranges. Specifically, "age" range from $[47, 53]$ increases to $[47, 63]$, while "hours_per_week" range from $[15, 25]$ expands to $[15, 99]$. Furthermore, we can explore the categorical feature's "native_country" alternative values. In Table 4, the first list concerns the values that may change the instance's prediction, while the second list displays the values that they can not affect the prediction.

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', 'Trinadad&Tobago', 'Honduras'

Table 4. List of values affecting or not the classification of an instance

5 RESULTS & DISCUSSION

In the first two datasets, we observe that only association rules can reduce the features, while random based and clustering are reducing the paths more effectively. But in the third dataset, association rules are useless, achieving zero feature and path reduction, while random based and clustering perform both feature and path reduction. LionForests is an ensemble of techniques and through these experiments we revealed the necessity of each component. We can conclude that the LionForests technique is considerably effective on both feature and path reduction. It creates such stable and robust rules, which are more indisputable from other explanations since they are more compact, they have wider feature-ranges, while in the same time, they present categorical data in a human-comprehensible form. Besides, these experiments revealed how random forests classifiers can be interpreted optimally. These results go beyond previous reports, which are either visualising random forests structure [41] or creating a list of features with their ranges sorted by their influence [31].

Specifically, in iForest the explanations are generated through user input. A user has to draw a lasso in the decision path projection [Figure 1a] to obtain the feature summary and the decision path flow. Choosing a wrong set of paths or a non-representative set of paths will lead to a faulty explanation, which may misguide the user.

On the other hand, in Moore et al. [31], the explanations are lists of features with their ranges ranked by their influence [Figure 2]. However, they do not attempt to widen these ranges. Also, they assume that their influence metric will assign zero influence to some features, and by extension removing them, they could offer more compact explanations. Despite this, they do not know by keeping only features with non-zero influence, that these features will be at least present and true in the quorum of the trees responsible for the instance's prediction. Additionally, they are not handling the categorical features with such elegant way as LionForests.

Nevertheless, using LionForests is not a complete solution either because it does not apply for model inspection tasks. For instance, if a researcher is working to develop a robust and stable model, with the highest performance, he may need a visualisation tool like iForest. Thus, this method is a proposed framework to connect the user with his explanation most optimally and easily. One last negative effect of using our approach is that by decreasing the number of paths to the quorum to reduce the features and in the same time to broaden their ranges, will result to a discounted probability of the instance's classification, which raises doubts about the reliability of the prediction. This can be counter-attacked by adding a threshold parameter to the reduction effect, to force the algorithm to keep at least a specific percentage of the paths.

6 CONCLUSION

Providing helpful explanations is a challenging task. Providing comprehensible and undeniable explanations is even tougher. Seeking to investigate every black-box model agnostically will not lead to the desired outcome because model-agnostic methods produce approximations, namely the nearest optimal explanations, but not ***the*** optimal explanations. In this work, we introduced a model-specific local-based approach for obtaining optimal interpretations for 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 be unfamiliar with visualisation provided by iForest [41]. Moreover, an explanation containing a lot of features [31] with narrow ranges, it may lead to large and untrustworthy rules. The proposed technique, which we call "LionForests", can provide to users small rules as explanations in natural language form, while at the same time the feature-ranges will be broadened making the explanations more stable and trustworthy. In order to achieve feature and path reduction we used classic unsupervised techniques like association rules and k-medoids clustering.

Future research will investigate the effect of models' parameters tuning to feature and path reduction. Furthermore, we could examine different tree ensemble models, rather than random forests, as well as different datasets and data types. FPGrowth [22], and its variant FP-Max [18], will be tested against the Apriori algorithm. Additionally, we can explore the possibility of applying LionForests to other tasks like multi-class or multi-label classification, as well as regression. Also, we will study the possibility of using LionForests' explanations to provide descriptive narratives through counter-examples. Finally, we are going to further analyse this promising approach to prove its comprehensibility through human-oriented evaluation.

ACKNOWLEDGEMENTS

This paper is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619, AI4EU Project[5].

[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. IbmâĂŹs 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 microelectronics (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/oeGYCx5`, 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] Tameru Hailesilassie, '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, 'Lionets: Local interpretation of neural networks through penultimate layer decoding', in *ECML PKDD 2019 AIMLAI XKDD Workshop. WÃijrzburg, 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).