Explainable Intelligent Environments

Davide Carneiro[1,3], Fábio Silva[1,2], Miguel Guimarães[1], Daniel Sousa[1], and Paulo Novais[2]

[1] CIICESI, Escola Superior de Tecnologia e Gestão, Instituto Politécnico do Porto Felgueiras, Portugal
{dcarneiro,fas,8150520,8160334}@estg.ipp.pt
[2] Algoritmi Centre/Department of Informatics, Universidade do Minho, Portugal
pjon@di.uminho.pt

Abstract. The main focus of an Intelligent environment, as with other applications of Artificial Intelligence, is generally on the provision of *good* decisions towards the management of the environment or the support of human decision-making processes. The quality of the system is often measured in terms of accuracy or other performance metrics, calculated on labeled data. Other equally important aspects are usually disregarded, such as the ability to produce an intelligible explanation for the user of the environment. That is, asides from proposing an action, prediction, or decision, the system should also propose an explanation that would allow the user to understand the rationale behind the output. This is becoming increasingly important in a time in which algorithms gain increasing importance in our lives and start to take decisions that significantly impact them. So much so that the EU recently regulated on the issue of a "right to explanation". In this paper we propose a Human-centric intelligent environment that takes into consideration the domain of the problem and the mental model of the Human expert, to provide intelligible explanations that can improve the efficiency and quality of the decision-making processes.

Keywords: Intelligent Environments, Explainable AI, Fraud Detection

1 Introduction

Artificial Intelligence is nowadays used in virtually all aspects of our lives, controlling our routines in pervasive and transparent ways, but nonetheless taking decisions with significant influence. These applications range from innocuous ones such as image or speech classification, used in our smartphones and virtual assistants [1], to critical ones such as autonomous vehicle driving, health diagnostics, or crime/re-incidence risk assessment [2].

Generally, the more complex the problem/domain is, the more complex the models learned are. Consequently, they are also harder to understand. This poses an interpretability problem: we often get a decision from a model, but we lack the information to properly judge and evaluate the decision. How good is it? How good are neighboring decisions? What is the rationale behind it?

There are domains in which the lack of an explanation is not relevant. However, in domains in which the lives of people are significantly affected, explanations are of the utmost importance. For instance, an individual should not be sent to jail or a credit card should not be denied with a simple "yes or no" answer. Such decisions should come with a proper explanation, that would allow the interested parties to *understand* the reasons behind the decision. Indeed, we often fail to understand how these complex models work. This is not a problem while models work as expected. However, when there is the need to debug them, we often learn that we do not understand their inner workings.

One of the best arguments in favor of the need for explanations, even when a model is apparently working appropriately, is given by [3]. The authors conducted an experiment whose task was to classify pictures containing either wolfs or huskies. While the model performed fairly well, the use of saliency maps showed that the model was not deciding based on the pixels that constituted the animal, but was actually using the background of the picture which contained mostly snow in the case of wolves, and grass in the case of huskies. If we were to provide the model with an image of a wolf standing on a grassy background, it would probably get it wrong and we would have no idea why.

The need for explanations in AI is thus evident, much more so in critical applications. Indeed, the EU recently regulated on the "right to explanation"[4], ensuring that any decision uttered by an automated algorithm that has critical and binding decisions must be accompanied by an intelligible explanation.

In line with this view, in this paper we propose a human-in-the-loop system, that combines Human experts and Machine Learning. The system continuously learns from the interaction with the Human experts, and the efficiency of this process is improved through elements of explainable AI such as interpretability, interactivity or counterfactual analyses. The system is also developed bearing in mind the mental model of the Human expert and the specific domain of fraud detection. However, the approach is general enough to be used in other domains.

2 Explanations and Human Factors

The concept of Explainable Artificial Intelligence (xAI) is related with the ability of a Human to *understand* the decision process of algorithms. In this context it is important to first make the distinction between two important terms: explanation and interpretability.

Indeed, one can explain a decision process without actually understanding the model which generated such decision, or the intricate relationships between cause and effect in the decision process [5]. Thus, the ability to understand how a decision algorithm behaves when its inputs are sightly altered relates to the interpretability of the model. In other words, the ability to predict how changes in the input change the decision output. On the other hand, explainability is related to how the human cognition can understand the mechanics of the decision from their natural perception. The subtle difference is that to explain a decision we do not need to understand how a decision could be altered if inputs were different.

An explanation can also vary according to its degree of completeness, which is the extent to which it allows a complete understanding of all the domains for each attribute in the decision-making process [6].

Explanation is naturally easier on some models, namely statistical or rule-based algorithms. It is much harder and less intuitive in ensemble models or under the umbrella of the connectionist methods, namely with algorithms such as Recorrent Neural Networks (RNNs). Indeed, explanations and interpretability are particularly difficult in these so-called "black-box" models, that are characterized by high complexity and abstraction levels. Nonetheless, many different approaches are being undertaken in both explainable and black-box models, which are reviewed in the next section.

2.1 Approaches to enhance explainability and interpretability

The research community has developed several approaches to improve explanations and interpretability in Machine Learning (ML) models. These approaches are sometimes specific to a given algorithm, or generic and applicable to a broad range of them.

One of the most interesting examples is the use of counterfactuals or evidence based on the interpertability of the model. These require a deep understanding of the machine learning model being used and how changes in the input may alter the decision outcome [7]. These decisions are categorized by the complete categorization of a specific decision and or how the decision would be altered given some changes in the input.

This is a generic idea which may have different implementations depending on the algorithm being studied. In the literature we can find this approach in linear classification algorithms [8] where a linear machine learning algorithm is exploited to find how changes in coefficients or inputs change the final decision.

Black box models, such as mutlilayer perceptrons, can also embed this approach. In [9], a genetic algorithm is used to search an output domain to provide suggestions for credit risk assessment, which can be perceived as an approach to interpret and explain a neural network decision process. This approach is similar to a technique known as LIME: Local Interpretable Model-Agnostic Explanations [10], which develops an approximation of the model by testing what happens when certain aspects within the input of the model are changed. It is about trying to recreate the outputs through a process of experimentation.

Still in the domain of credit scoring, there are also examples of ensemble explanation, implemented through layers of interpertability of machine learning models [11]. In this approach, the decision making process is explained in different steps by an expert rule based system.

In the case of black box models, there are techniques to recreate the decision process through the analysis of the internals of such models. In the case of neural networks and deep learning models, there is a technique called Deep Lift [12]. It works by taking the output and attempting to interpret the neurons that are significant to the original output. In short, it performs a sort of feature selection to explain the decision process based on the activated neurons. A similar

approach to Deep Lift is the layer-wise relevance propagation technique [13]. It also works backwards from the output, identifying the most relevant neurons within the neural network.

The general perception is that all models can be explained to some extent, some more than other. Moreover, some are easy to explain (generally those under a symbolic approach to AI) while other are more challenging (generally the connectionist models). However, explanations should also consider the mental model of the user and the domain of the application. In this paper we describe an intelligent environment for the domain of fraud detection, that incorporates a series of concepts from explainable systems, and that is built to integrate with the work of a Human auditor.

3 An Explainable Intelligent Environment for Tax Fraud Detection

The importance of explaining decisions in an Intelligent Environment has already been addressed in Section 2. However, nowadays, explanations are not only desirable from a perspective of interpretability but are starting to become a legal requirement. In the context of the GDPR, the EU recently regulated on algorithmic decision-making and, specifically, addressed the issue of a "right to explanation"[4]. There are particularly sensitive domains in which algorithmic decisions significantly affect one's life, such as credit scoring, sentencing, or fraud detection.

In this paper we present one such environment, in the domain of financial fraud detection, in the context of the Neurat funded project (31/SI/2017 - 39900). This environment is being built as a cooperative system in which Machine Learning tools and Human experts work together to increase the efficiency of tax audits.

However, the use of Machine Learning, and in particular of supervised methods, requires vast amounts of labeled data. The problem is that data can only be labeled by Human experts (auditors) and, in this case, it comes at a high cost: auditors must undergo extensive training and their time is very limited. As a consequence, they are able to review but a small portion of the transactions of a company, usually by sampling, and thus provide a small amount of labeled data.

An Active Learning (AL) approach is being followed to implement this environment [14]. Generally, AL approaches aim to make ML less expensive by reducing the need for labeled data. To achieve this, a so-called *Oracle*, which may be a Human expert or some automated artifact, is included in an cycle in which a ML model is continuously improved by training on a growing pool of labeled data. The key element in this approach is the selection strategy for unlabeled data, which will optimize selection queries so that learning occurs faster. Different data selection strategies may be implemented. However, the goal is the same: to cover the search space as quickly as possible, minimizing the necessary labeled data. ML accuracy is maintained while reducing the training set size.

However, we introduce two major changes to the "traditional" AL scheme (Figure 1). First, we consider a pool of models rather than a single model [15]. New models are trained and added to the pool, which constitute a voting/averaging ensemble whose weights are continuously optimized by a Genetic Algorithm. Over time, models with a smaller weight are removed from the ensemble. This allows the system to converge while using relatively simple models, trained with partial data, instead of a very large and complex one.

Secondly, we add another input to the Oracle, which in this case is the Human auditor. The auditor has access to the selected instance i, which is now accompanied by a prediction p and an explanation e. Both are provided by the ensemble f and are a result of $f(i)$, that is, of asking the current ensemble to classify a specific instance. Now, when the auditor receives the instance to label (that is, when the auditor performs an audit action), he also receives the label proposed by the system as well as an intelligible explanation for it, tailored for this specific domain.

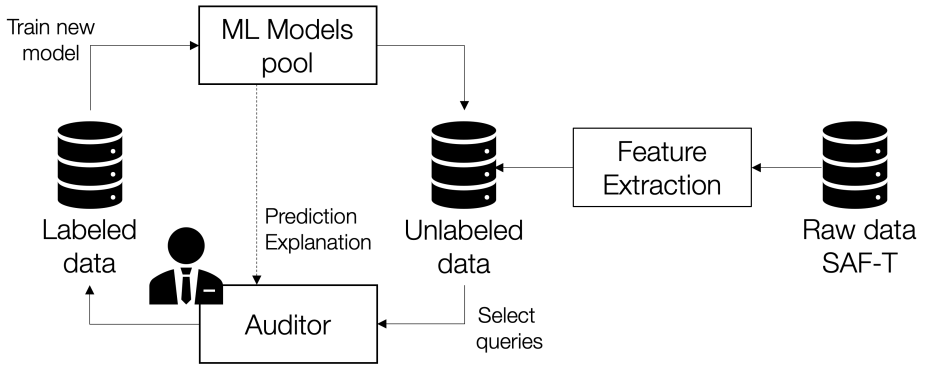

Fig. 1. Overview of the main elements of the proposed environment for fraud detection.

To achieve this, we are using a modified version of the CART algorithm[16]. This algorithm allows to build a Decision Tree from a group of observations. Each node of the tree contains boolean rules about the observations (e.g. value of variable x is greater than y) and each leaf contains the result of the prediction for a given path in the tree. While the tree is being built, the training set is increasingly split at each node, leading to smaller sub-sets of the data. This splitting process ends when one or more stopping criteria are met, which may include a minimum size of the split or a minimum degree of variance/purity.

Variance denotes how much the values for the dependent variable of a split are spread around their mean value (in regression tasks), while purity considers the relative frequency of classes: if all classes have roughly the same frequency the node is deemed "impure". The Gini index is used in the CART algorithm to measure impurity [17].

Formula 1, as proposed by [18], describes the relationship between the outcome y and features x. Each instance of the training set is attributed to a single leaf node (subset R_m). $I\{x \in R_m\}$ is a function that returns 1 if x is in the subset R_m or 0 otherwise. In a regression problem the predicted outcome $\widehat{y} = c_l$ of a leaf node R_1 is given by the average value of the instances in that same node.

$$\widehat{y} = \widehat{f}(x) = \sum_{m=1}^{M} c_m I\{x \in R_m\} \quad (1)$$

While the algorithm can be used for both classification and regression tasks, in this work we use a regression tree, as the task is to assign a value between 0 and 10 which represents the degree of certainty of a given instance to constitute fraud.

3.1 Generating interpretable explanations

A Decision Tree is, in itself, an explainable model: it can be analyzed visually to understand which variables and values are used at each level to take a decision. However, this may be difficult for example if the tree is too large. There is also additional information that can be provided that is not explicitly in the tree's structure. In this section we detail the explainable elements that are generated by the system, to support the Human auditor in decision-making.

When the tree is being built and each split generated, additional information is stored in the node which includes: the boolean rule that generates the split (mentioning the variable and the value interval), the prediction $\widehat{y}$ based on that split (i.e. the average or most frequent value, depending on the problem), measures of dispersion or purity (variance, standard deviation and Gini index), and the indexes of the instances in the split.

These values are then used to provide a notion of *confidence* and *support* to the decision-maker. Confidence is given by dispersion and purity measures: the lower the dispersion or the higher the purity, the higher the confidence on the decision is. Support is given by the number of instances in the split: the higher the number of instances, the higher the support is.

This information on the nodes allows to incorporate a group of explainable elements in the user interface. Figure 2 shows a prototype of the graphical user interface that is used to provide explanations. When an auditor wants to analyze a specific instance she/he selects that instance and is redirected to this interface, which receives the data of the instance, the prediction, and an explanation. The user interface has three main areas, marked in the Figure as (a) - Explore, (b) - Decision path and (c) - Last results.

Area (a) allows the user to explore the search space and analyze each feature according to their relative importance. Features and values are collected from the internal nodes when traversing the tree to make a prediction. In this context, feature relevance is based on how much that split/feature decreases dispersion/purity. For each feature that the interface shows the following elements

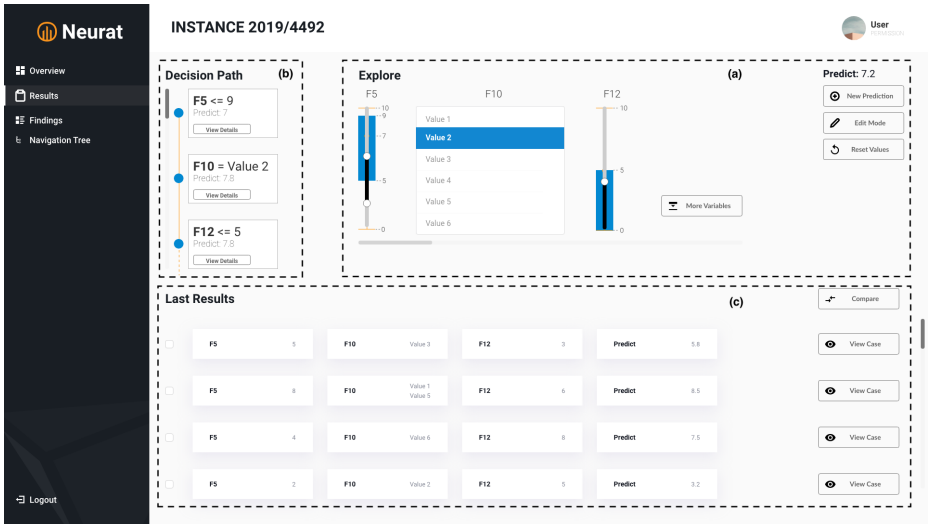

Fig. 2. Prototype of the main screen of the application, with some of the explainable elements created, and three main areas highlighted: Explore (a), Decision Path (b) and Last Results (c).

(depending on whether the variable is numeric or nominal): the domain of the feature (range/enumeration of possible values), the interval/values for which the prediction holds (blue bar or values highlighted in blue), and the value of the feature in the instance being audited (gray dot).

This allows the auditor to gain a sense of how *risky* the decision is. If the value of a given feature is very close to the upper or lower limits of the blue bar, it indicates that a slight change of this feature towards the limit would significantly alter the prediction of the tree. Likewise, the size of the blue bar is also related to this sense of risk: the shorter the bar the more risky the decision is. In the case of a nominal feature, multiple values can be highlighted to show for which values of the enumeration the prediction holds. The risk of the decision grows with fewer highlighted values.

In Figure 2, the graphical interface is shown in "Edit Mode". This means that the user may change the values of the variables to perform a counterfactual analysis. That is, what would be the prediction if the value of a feature had been v_2 instead of v_1. These "what-if" scenarios allow the auditor to interact with the tree and to understand how predictions would change under different scenarios. This contributes significantly to the interpretability and interactivity of the explanation, as addressed in Section 2. The user does this by changing the value of the features by means of a slider, or by selecting a value from a list. The scenarios created by the user can be added to area (c), to be compared. The user can also reset area (a), returning all the values and the associated prediction to the initial state of the instance being audited.

There is also a pagination mechanism that controls the amount of information provided to the user, to avoid overload. Indeed, depending on the training set, the number of levels/nodes/features on a tree may be too large to be efficiently analyzed by a Human. In that sense, in this interface we show only the n most relevant features. The user can then choose to request additional features (and the associated prediction) by clicking on the "More variables" button. These are gradually added upon request by decreasing relevance.

In the left side of the interface there is the area marked as (b). This area shows the path followed through the tree to make the prediction. Like in (a), this area may not show the whole path as it implements the same pagination mechanism: when features are added to (a) they are also added to (b). This element allows the user to understand (part of) the reasons for a given prediction: "because feature f_1 is smaller or equal than v_1 and feature f_2 equals v_2".

In this area the user may also click on a specific node to see its details (Figure 3). The details show, in the left side, the information for the feature that is also visible on (a). On the center and right, the "details" modal provides information regrading the *confidence* and *support* of the prediction. The graphical representation shows the prediction (blue dot) and the interval given by the standard deviation. A smaller interval indicates an increased confidence as instances in this split are more closely distributed around the mean, and vice-versa.

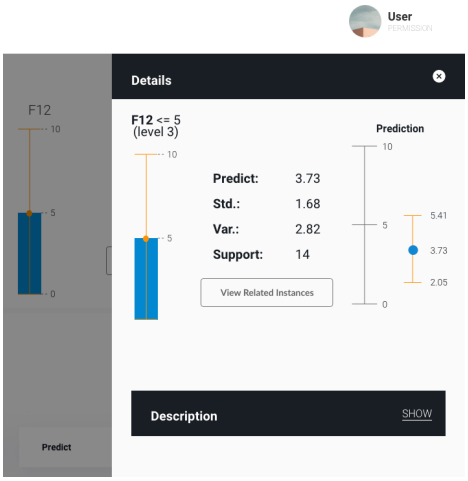

Fig. 3. Details of a split node, with confidence and support measures.

The central part of the modal shows values which include the support (number of instances in this split) and a button that allows the user to access the instances that fall into this split. The user may thus visualize the instances, which are shown sorted by similarity to the current instance in descending order. Similarity is calculated based on a weighted sum of differences, given by the

euclidean distance for numerical variables and by the cosine similarity for the vector of nominal data (if any). While visualizing specific instances the user may add them to a list for comparison (area (c)).

As the user moves down the path, splits become smaller but confidence increases. It is up to the user to decide how far down to travel: an early stop may lead to a more general decision (with high support and potential low confidence), while going further down will lead to low support but high confidence. Finally, in area (c) the user has access to a list of previous prediction results (the scenarios that were simulated) and/or to actual instances that were visualized by the user and added for comparison. This allows to more easily compare a group of scenarios or real cases and their results.

4 Conclusions and Future Work

With the growing use of AI models in our daily lives and the impact of their decisions, their inner workings must be more closely scrutinized. More and more we require not only a decision or a prediction, but also an intelligible explanation that we can use to judge the quality of the decision. However, the vast majority of existing AI systems do not consider this kind of elements. In this paper we presented an adapted version of a human-in-the-loop system, based on Active Learning. We expand the "traditional" process flow with the provision of predictions and corresponding explanations for the unlabeled data that is presented to the Human expert. We believe that the provision of these explanations will contribute to the efficiency of the interaction between the Human and the system, as well as to the quality of the decisions made by the Human. The quantification of such improvements will be carried out in future work. Among other aspects, the proposed system considers elements such as interactivity, counterfactual explanations, simulation, and rule-based explanations. The approach was developed taking into consideration the mental model of the auditor. Nonetheless, it is generic enough to be used in other domains, thus contributing to an increased awareness of users towards the Machine Learning models that they interact with.

5 Acknowledgments

This work was supported by the Northern Regional Operational Program, Portugal 2020 and European Union, trough European Regional Development Fund (ERDF) in the scope of project number 39900 - 31/SI/2017, and by FCT - Fundação para a Ciência e a Tecnologia, through projects UIDB/04728/2020 and UID/CEC/00319/2019.

References

1. Ververidis, D., Kotropoulos, C., Pitas, I.: Automatic emotional speech classification. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume 1., IEEE (2004) I–593

2. Crawford, K.: Artificial intelligence's white guy problem. The New York Times **25** (2016)
3. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. (2016) 1135–1144
4. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI magazine **38**(3) (2017) 50–57
5. Dosilovic, F.K., Brcic, M., Hlupic, N.: Explainable artificial intelligence: A survey. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings (May) (2018) 210–215
6. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018 (2019) 80–89
7. Sokol, K., Flach, P.: Desiderata for interpretability: Explaining decision tree predictions with counterfactuals. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 10035–10036
8. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency. (2019) 10–19
9. Silva, F., Analide, C.: Information asset analysis: credit scoring and credit suggestion. International Journal of Electronic Business **9**(3) (2011) 203
10. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should i trust you?" Explaining the predictions of any classifier. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining **13-17-August-2016** (2016) 1135–1144
11. Chen, C., Lin, K., Rudin, C., Shaposhnik, Y., Wang, S., Wang, T.: An Interpretable Model with Globally Consistent Explanations for Credit Risk. arXiv preprint arXiv:1811.12615 (nov 2018) 1–10
12. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. 34th International Conference on Machine Learning, ICML 2017 **7** (2017) 4844–4866
13. Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R., Samek, W.: Layer-wise relevance propagation for neural networks with local renormalization layers. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **9887 LNCS** (2016) 63–71
14. Settles, B.: From theories to queries: Active learning in practice. In: Active Learning and Experimental Design workshop In conjunction with AISTATS 2010. (2011) 1–18
15. Ramos, D., Carneiro, D., Novais, P.: evorf: An evolutionary approach to random forests. In: International Symposium on Intelligent and Distributed Computing, Springer (2019) 102–107
16. Singh, S., Gupta, P.: Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. International Journal of Advanced Information Science and Technology (IJAIST) **27**(27) (2014) 97–103
17. Lerman, R.I., Yitzhaki, S.: A note on the calculation and interpretation of the gini index. Economics Letters **15**(3-4) (1984) 363–368
18. Molnar, C.: Interpretable machine learning. Lulu. com (2019)