Revista Brasileira de Computação Aplicada, April, 2020

DOI: 10.5335/rbca.v12i1.10247
Vol. 12, Nº 1, pp. 113–121
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

ORIGINAL PAPER

A study about Explainable Artificial Intelligence: using decision tree to explain SVM

Carla Piazzon Ramos Vieira [1] and Luciano Antonio Digiampietri [1]

[1]University of São Paulo

*carla.piazzon.vieira@usp.br; digiampietri@usp.br

Received: 2019-11-16. **Revised:** 2020-01-08. **Accepted:** 2020-03-23.

Abstract

The technologies supporting Artificial Intelligence (AI) have advanced rapidly over the past few years and AI is becoming a commonplace in every aspect of life like the future of self-driving cars or earlier health diagnosis. For this to occur shortly, the entire community stands in front of the barrier of explainability, an inherent problem of latest models (e.g. Deep Neural Networks) that were not present in the previous hype of AI (linear and rule-based models). Most of these recent models are used as black-boxes without understanding partially or even completely how different features influence the model prediction avoiding algorithmic transparency. In this paper, we focus on how much we can understand the decisions made by an SVM Classifier in a post-hoc model agnostic approach. Furthermore, we train a tree-based model (inherently interpretable) using labels from the SVM, called secondary training data to provide explanations and compare permutation importance method to the more commonly used measures such as accuracy and show that our methods are both more reliable and meaningful techniques to use. We also outline the main challenges for such methods and conclude that model-agnostic interpretability is a key component in making machine learning more trustworthy.

Keywords: black-box; explainable artificial intelligence; interpretability; explainability; transparency

Resumo

As tecnologias baseadas em Inteligência Artificial (IA) avançaram rapidamente nos últimos anos e a IA está se tornando comum em todos os aspectos da vida, como o futuro dos carros autônomos ou agilidade em diagnósticos médicos. Para que isso ocorra, toda a comunidade está diante da barreira da explicabilidade, um problema inerente às mais recentes técnicas trazidas por modelos mais complexos de aprendizado máquina (por exemplo, redes neurais profundas) que não estavam presentes na última onda de IA (modelos lineares ou baseados em regras). A maioria desses modelos mais recentes é usada como caixa-preta, sem entender parcialmente ou até completamente como diferentes características influenciam nas predições do modelo, evitando a transparência algorítmica. Este artigo foca na identificação da melhor maneira de entender as decisões tomadas por um classificador SVM em uma abordagem agnóstica post-hoc. Além disso, treinamos um modelo baseado em árvore de decisão (inerentemente interpretável) usando rótulos do SVM, chamado de dado secundário de treinamento, para fornecer explicações e comparar a importância das características por meio do método de permutação com as métricas mais usadas, como acurácia, e mostrar que nossos métodos e técnicas são mais significativos. Também delineamos os principais desafios de tais métodos e concluímos que a interpretabilidade post-hoc é um componente essencial para tornar o aprendizado de máquina mais confiável.

Palavras-Chave: caixa preta; inteligência artificial explicável; interpretabilidade; explicabilidade; transparência

1 Introduction

As intelligent systems become more widely applied (robots, automobiles, medical and legal decision-making), users and the general public are becoming increasingly concerned with issues of explainability and trust. These considerations in the public discourse are partly responsible for the establishment of projects such as DARPA's Explainable AI Project (Gunning, 2017), European response to the General Data Protection Regulation (Goodman and Flaxman, 2017), and the recent series of Explainable Artificial Intelligence (XAI) Workshops and Talks at major AI conferences such as South by Southwest (SXSW) and Google I/O.

Explainability is not a new issue for AI systems. But it has grown along with the success and adoption of deep learning, which has given rise both to more diverse and advanced applications and to more opaqueness. The current generation of Intelligent Systems based on machine learning seems to be what we call black-box models. Even when inputs and outputs are known, these systems can suggest answers, but not say the "why" behind their decisions. This makes it difficult for a company or the government to explain its decision-making process to clients, board members, and other stakeholders, or for doctors to have confidence about the results produced by some algorithms.

"*If the designers and end-users of a learning system are to be confident in the performance of the system, they must understand how it arrives at its decisions. Learning systems may also play an important role in the process of scientific discovery.*" (Craven and Shavlik, 1995)

It is hard to imagine a person who would feel comfortable in blindly agreeing with a system's decision in such highly consequential and ethical situations without a deep understanding of the decision making rationale of the system. To overcome this dangerous practice, it is prudent for an AI to provide not only an output, but also a human-understandable explanation that expresses the rationale of the machine. Analysts can turn to such explanations to evaluate if a decision is reached by rational arguments and does not incorporate reasoning steps conflicting with ethical or legal norms.

"*The current generation of AI systems offer tremendous benefits, but their effectiveness will be limited by the machine's inability to explain its decisions and actions to users*" (Gunning, 2017).

But what constitutes an explanation? The Oxford English Dictionary has no entry for the term 'explainable', but has one for *explanation*: "*A statement or account that makes something clear; a reason or justification given for an action or belief*". Do present systems that claim to make 'explainable' decisions really provide explanations?

Explanation is closely related to the concept of interpretability: systems are interpretable if their operations can be understood by a human. In the case of machine learning models, explanation is often a difficult task since most models are not readily interpretable. In order for humans to trust black-box methods, we need explainability - models that are able to summarize the reasons for their behavior, gain the trust of users or produce insights about the causes of their decisions.

The prevailing solution to this explanation problem is to use the so-called "interpretable" models, such as decision trees, rules or linear models. Instead of supporting models that are functionally black-boxes, such as neural networks or random forests with thousands of trees, these approaches use models in which there is the possibility of meaningfully inspecting model components directly - e.g. a path in a decision tree, a single rule, or the weight of a specific feature in a linear model. As long as the model is accurate for the task, and uses a reasonably restricted number of internal components (i.e. paths, rules, or features), such approaches provide extremely useful insights. But these models are in general relatively simple and thus inadequate for capturing the complexity of some real-world problems.

An alternative approach to interpretability in machine learning is to be *model-agnostic*, i.e. to extract *post-hoc* explanations by treating the original model as a black-box. It might be possible to hybridize an inherently explainable modeling approach with a complex black-box method to devise a high-performance and explainable model.

In this article, we make a preliminary attempt to answer the question: what can we infer and interpret from an already trained model that can help to understand its decisions on test data, i.e., post-hoc interpretability. However, in this article, we do not focus on the explanations themselves. Instead, we focus on learning an interpretable model from a SVM model given we have a reasonable domain understanding and user expertise.

To this extent, we first present a set of definitions and review several approaches towards interpretability of AI systems. In Section 2, we define key terms including "explanation", "interpretability", and "explainability". In Section 3, we provide a summary of related work papers highlighting differences between explainable AI approaches for SVM. In Section 4, we propose a *post-hoc* approach using decision trees to extract rules from a SVM model (black-box). In Section 5, we present and discuss the results. We conclude, in Section 6, with a discussion addressing open questions and recommend a path to the development and adoption of explainable methods for artificial intelligence applications.

2 Background and Basic Concepts

In this section, we provide background information about the key concepts of explainability and interpretability, and describe the meaningful differences between them.

2.1 Interpretability

There is no mathematical definition of interpretability. A (non-mathematical) definition by Miller (2017) is: "*Interpretability is the degree to which a human can understand the cause of a decision*". The higher the interpretability of a machine learning model, the easier it is for someone to comprehend why certain decisions or predictions have been made. A model is better interpretable than another model if its decisions are easier for a human to comprehend. In this paper, we will use both the terms interpretable and explainable interchangeably. Like Miller (2017) said, it makes sense to distinguish between the terms interpretability/explainability and explanation. We will use "explanation" for explanations of individual predictions.

2.2 What Is an Explanation?

"*An explanation is the answer to a why-question*" (Miller, 2017).

- Why did not the treatment work on the patient?
- Why was my loan rejected?

2.2.1 *What Is a Good Explanation?*

This section further condenses Miller's summary on "good" explanations and includes concrete implications for interpretable machine learning.

Explanations are contrastive (Lipton, 1990). Humans usually do not ask why a certain prediction was made, but why this prediction was made instead of another prediction. We tend to think in counterfactual cases, i.e. "How would the prediction have been if input X had been different?". For a house price prediction, the house owner might be interested in why the predicted price was high compared to the lower price they had expected. If my loan application is rejected, I do not care to hear all the factors that generally speak for or against a rejection. I am interested in the factors in my application that would need to change to get the loan. The recognition that contrasting explanations matter is an important finding for explainable machine learning.

From most interpretable models, you can extract an explanation that implicitly contrasts a prediction of an instance with the prediction of an artificial data instance or an average of instances.

Physicians might ask: "Why did the drug not work for my patient?". And they might want an explanation that contrasts their patient with a patient for whom the drug worked and who is similar to the non-responding patient. Contrastive explanations are easier to understand than complete explanations. A complete explanation of the physician's question of why the drug does not work might include: The patient has had the disease for 10 years, the patient's body is very quick in breaking the drug down into ineffective chemicals, etc. A contrastive explanation might be much simpler: In contrast to the responding patient, the non-responding patient has a certain combination of genes that make the drug less effective. The best explanation is the one that highlights the greatest difference between the object of interest and the reference object.

Humans do not want a complete explanation for a prediction, but want to compare what the differences were to another instance's prediction (can be an artificial one). Creating contrastive explanations is application-dependent because it requires a point of reference for comparison. And this may depend on the data point to be explained, but also on the user receiving the explanation.

Explanations are selected. People do not expect explanations that cover the complete list of causes of an event. We are used to selecting one or two causes from a variety of possible causes as the explanation. Make the explanation very short, give only 1 to 3 reasons, even if the world is more complex.

2.3 Model-Agnostic Methods

The easiest way to achieve interpretability is to use only a subset of algorithms that create interpretable models. Linear regression, logistic regression, and the decision tree are commonly classified as interpretable models. However, the big disadvantage is that predictive performance is lost compared to other machine learning models and you limit yourself to a few types of models. The other alternative is to use model-specific interpretation methods. The disadvantage of this is that it binds you to one model type and it will be difficult to switch to something else.

As an alternative for these approaches is separating the explanations from the machine learning model (model-agnostic interpretation methods) (Ribeiro et al., 2016a).

The great advantage of model-agnostic interpretation methods over model-specific ones is their flexibility. Machine learning developers are free to use any machine learning model they like as the interpretation methods can be applied to any model. Anything that builds on an interpretation of a machine learning model, such as a graphic or user interface, also becomes independent of the underlying machine learning model. Typically, not just one, but many types of machine learning models are evaluated to solve a task, and when comparing models in terms of interpretability, it is easier to work with model-agnostic explanations, because the same method can be used for any type of model.

As you can see in Fig. 1 (Molnar, 2017), the world can be captured by collecting data and it is further abstracted by learning to predict the data with a machine learning model. Interpretability is another layer on top that helps humans understand.

The first layers represent the World that contains everything that can be observed, such as the biology of the human body and how it reacts to medications.

The second layer is the Data extracted from observations of the World and formatted to make it processable by computers.

By fitting machine learning models based on the Data layer, we get the Black-Box Model layer. Machine

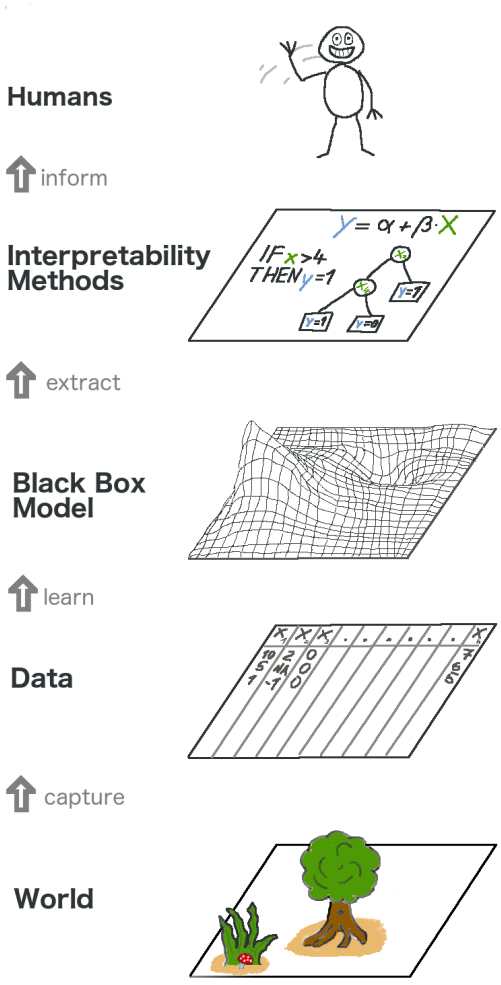

Figure 1: The big picture of explainable machine learning. The real world goes through many layers before it reaches the human in the form of explanation (Molnar, 2017).

learning algorithms learn with data from the real world to make predictions or find structures.

Above the Black-Box Model layer is the Interpretability Methods layer, which can help to deal with the opacity of machine learning models. What were the most important features for a particular diagnosis? Why was a financial transaction classified as a fraud?

The last layer is occupied by a Human. Humans are ultimately the consumers of the explanations.

Of course, this graphic does not capture everything: Data could come from simulations. Black-box models also output predictions that might not even reach humans, but only supply other machines, and so on. But overall it is a useful abstraction to understand how interpretability becomes this new layer on top of machine learning models.

2.4 Methods to open black-box systems

A recent survey on methods for explaining black-box models (Guidotti et al., 2018) outlined a taxonomy to provide classifications of the main problems with opaque algorithms. Their taxonomy is detailed, distinguishing small differing components in explanation approaches (e.g. Decision tree vs. single tree, SVM, etc.) Their classification examines four features for each explanation method:

i. The type of problem faced.
ii. The explanatory capability used to open the black-box.
iii. The type of black-box model that can be explained.
iv. The type of input data provided to the black-box model.

They primarily divide the explanation methods according to the types of the problem faced, and identify four groups of explanation methods as you can see in Fig. 2 (Guidotti et al., 2018).

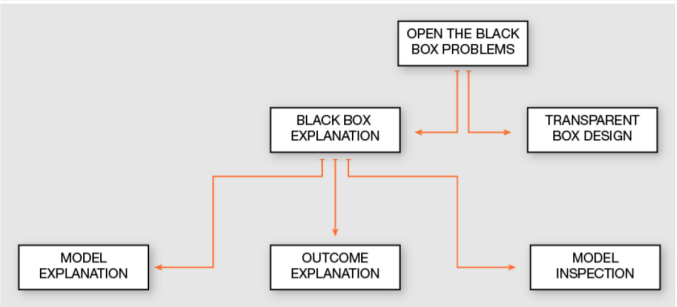

Figure 2: Open the black-box problems taxonomy (Guidotti et al., 2018).

The black-box explanation problem consists in providing a global explanation of the black-box model through an interpretable and transparent model. Given a black-box and an input instance, the outcome explanation problem consists in providing an explanation for the outcome of the black-box on that instance. The model inspection problem consists in providing a representation (visual or textual) for understanding some specific property of the black-box model or of its predictions. And, finally, the transparent box design problem consists in directly providing a model that is locally or globally interpretable.

We find this work a meaningful contribution that is useful for exploring the design space of explanation methods and helped to understand different approaches that are being used to achieve explainability that we used in this article.

3 Related Work

Due to the growing number of subfields, as well as the policy and legal ramifications of black-box systems, the volume of research in interpretability is quickly expanding. In the previous sections we reviewed different methods for explainability, so, in this section, we focus on papers related to post-hoc explanations.

There are two cases when considering the post-hoc interpretability of decisions made by the machine learning models - the first, where we are aware of the procedure (linear models, neural networks or others) used to train the model and second where the model is used as black-box. In the latter model agnostic scenario, it is harder to acquire a deeper understanding of the model's behavior and how different features influence the model's predictions.

During our search for related works, we found out that explainability is not a new area of study and research as it may seem. Single-tree approximations for Neural Networks were first presented in 1995 by Craven and Shavlik (1995). They presented an algorithm, TREPAN, that induces decision trees by querying trained neural networks. Their algorithm represented a promising advance towards the goal of general methods for understanding black-box algorithms predictions.

Recent work by Ribeiro et al. (2016b) trains a shadow interpretable model to match the results of a given classifier. They suggest the use of LIME[1] - an algorithm that can explain the predictions of any classifier in a faithful way.

Recently, the rule extraction with the SVM becomes a mainstream of machine learning. Each of the techniques to achieve explanation differentiates from others by how deep they go into the algorithm inner structure. First, some authors propose techniques to build rule-based models only from the support vectors of a trained model. This is the approach of Barakat and Bradley (2007), which proposes a method that extracts rules directly from the support vectors of a trained SVM using a modified sequential covering algorithm. In another work, Barakat and Bradley (2006) propose eclectic rule extraction, still considering only the support vectors of a trained model.

4 Experiments, Methods and Analysis

In this work, we focus on approximating and understanding the global decision surface of SVM model by rule extraction. It is important to note that our work shares some similarities with Barakat and Bradley (2006), but it differs in some aspects: (i) we do not use the area under the ROC curve (AUC) to compare the performance and (ii) we use decision tree to compare accuracy and to explain models while they use the eclectic rule extraction approach.

[1]The lime package code in Python is available at https://github.com/marcotcr/lime, accessed on 11-06-2019

4.1 Algorithms used

In this section, we provide an overview of the models used in this work considering their complexity and interpretability.

4.1.1 *Decision Trees*

Decision trees are an example of a model that can easily fulfill every constraint for transparency. Decision trees are hierarchical structures for decision making used to support regression and classification problems. However, their poor generalization properties in comparison with other models make this model family less interesting for their application to scenarios where a balance between predictive performance is a design driver of importance.

4.1.2 *Support Vector Machines*

SVM is a Machine Learning Model with a historical presence in literature. SVM models are more complex than decision trees, with a much opaquer structure.

As reviewed in Section 3, many implementations of post-hoc explainability techniques have been proposed to relate what is mathematically described internally in these models, to what different authors considered explanations about the problem at hand. Technically, an SVM constructs a hyperplane of a set of hyperplanes in a high or infinite-dimensional space, which can be used for classification and regression. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general, the larger the margin, the lower the generalization error of the classifier. SVMs are among the most used Machine Learning models due to their excellent prediction and generalization. But SVM has the following disadvantages: (1) the opaqueness of the decision procedure makes it difficult to be applied for health and finance applications; (2) SVM originally is a binary classification method and needs further improvements in the application of the multi-class problems.

4.2 Rule Extraction from SVMs

In this article, we use a rule extraction approach. This approach uses a labeled data set to train an SVM and obtain the SVM classification, which produced acceptable accuracy, precision, and recall. Next, a new data set composed of the original patterns is constructed with the target label for these patterns replaced by the label predicted by the SVM. Rules representing the concepts learned by SVM are then extracted from this new data set using a decision tree. The flow diagram steps of this work can be seen in Fig. 3.

Thus, the proposed approach uses three classification models and results. One produced directly by the decision tree on the dataset, one produced by the SVM classifier and one produced by the decision tree applied to the output of the SVM classifier. The first result works as the baseline of an explainable algorithm. The second represents the

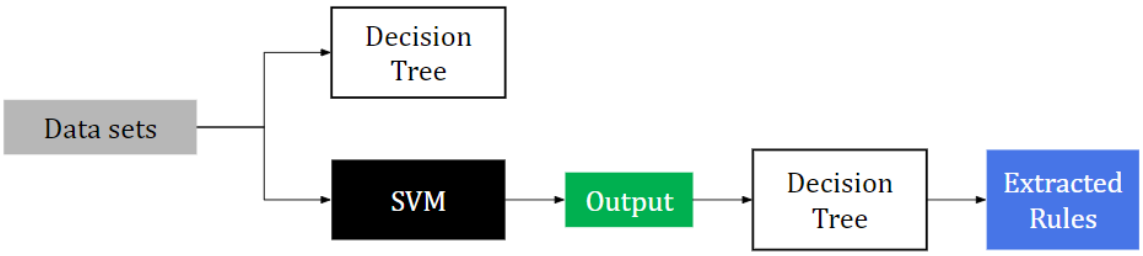

Figure 3: Methodology flowchart

results of a 'black-box' algorithm and, finally, the third is used to verify the ability of the decision tree to explain the SVM's results. In this context, the first result is used mainly as a measure for performance comparison with that of the SVM algorithm, and the produced decision tree is compared with the decision tree produced in the third result.

Experiments were performed using two benchmark data sets from UCI Machine Learning to assess the quality of the rules extracted using the rule-extraction approach. The details are presented as follows.

Pima Indians Diabetes[2] a sample of 534 patterns were used from the original data set, after removing all patterns with a zero value for the attributes glucose, diastolic blood pressure or triceps skin thickness which are clinically insignificant.

Heart Disease:[3] the reduced Cleveland heart diseases data set was used. All 303 patterns were used.

In our experimental design, each of the data sets was split into disjoint training and test sets, as shown in Table 1.

Table 1: Data sets

Data set	Nº Features	Training set	Test set
Pima Indians	8	373	161
Heart Disease	13	212	91

In the next subsections, we explain how we constructed the classifiers used in this work. For all classifiers, we used Python 3.6 and Jupyter Notebook. All the code used to produce this work is available on a public GitHub repository (https://github.com/carlaprv/tcc-classification-xai).

4.2.1 *Classifiers*

For both SVM and Decision Tree, we used the classifiers available on Scikit Learn (Buitinck et al., 2013).

As we intended to create a simple and explainable model, we trained our Decision Tree using all features and setting the depth to three. We used entropy to split nodes. Entropy is the measure of Randomness in the system.

For the SVM Classifier, we used all features and specify the linear kernel. It is one of the most common kernels to be used and is faster than any other kernel.

In order to achieve the primary goal of this work, i.e., to construct a rule-based classifier that can explain the classifications made by an SVM, we run the same Decision Tree Classifier built over the new data set considering the SVM Classifier output for the original data sets.

5 Results

In this phase, after running all experiments, we compare the metrics of each classifier as well the generated rules for each data set.

The first question we intend to answer is that if we used exactly the same features used for training SVM Classifier how close can we get to the classification induced by it.

Accuracy is the first factor for evaluating. Accuracy is the percentage of the correct classification. It can be seen that the Extracted Rules has the highest accuracy.

Table 2: Accuracy

Data set	Decision Tree	SVM	Extracted Rules
Pima Indians	0.75	0.80	0.88
Heart Disease	0.74	0.77	0.80

The precision is the ratio $tp/(tp + fp)$ where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0.

For both data sets, Extracted Rules has a better precision than the other classifiers. It means that the ability of the produced Decision Tree to explain the labels produced by the SVM classifier is better than the ability of both classifiers (Decision Trees and SVM) to represent the original input data.

Table 3: Precision

Data set	Decision Tree	SVM	Extracted Rules
Pima Indians	0.74	0.80	0.87
Heart Disease	0.73	0.77	0.80

[2]Data set available at https://www.kaggle.com/uciml/pima-indians-diabetes-database

[3]Data set available at https://www.kaggle.com/ronitf/heart-disease-uci

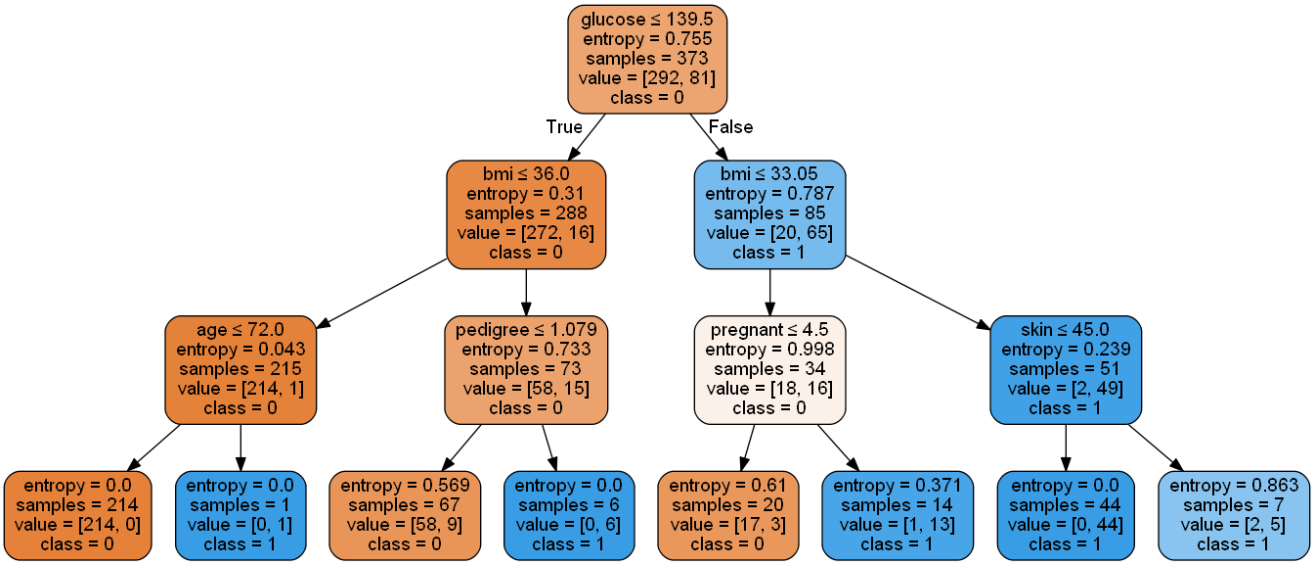

Figure 4: Extracted Rules from SVM for pima indians data set

The recall is the ratio $tp/(tp + fn)$ where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. The best value is 1 and the worst value is 0.

Table 4: Recall

Data set	Decision Tree	SVM	Extracted Rules
Pima Indians	0.75	0.76	0.84
Heart Disease	0.74	0.76	0.79

For a better understanding of recall and precision metrics, we computed the F1 score, also known as balanced F-score or F-measure. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$F1 = 2 * (precision * recall)/(precision + recall)$$

In this case, for both data sets, the F1 Score for Extracted Rules is greater than for the other classifiers.

Table 5: F1 Score

Data set	Decision Tree	SVM	Extracted Rules
Pima Indians	0.74	0.77	0.85
Heart Disease	0.73	0.77	0.79

The higher accuracy, precision, and recall for the Extracted Rules answers our first question and shows that this model is a good representation of the SVM classifier for both data sets, as we intended to create.

Figs. 4 and 5 provide a better understanding of the extracted rules from SVM Classifier. These figures were produced using the module Graphviz from Scikit Learn. In this visualization, the more samples of the first class (0), the darker the orange color of the node; the more samples of the second class (1), the darker the blue.

Although the tree representations for SVM Classifier provides an explanation for the model, we used ELI5, a Python library, to extract feature importance by measuring how score decreases when a feature is not available; the method is also known as "permutation importance" and the results are presented in Figs. 6 and 7.

5.1 Pima Indians

From Figs. 4 and 6, it is possible to identify that Glucose level, Body mass index (BMI) and diabetes pedigree (a function which scores likelihood of diabetes based on family history) have a significant influence on the model, especially glucose level and BMI. It is worth highlighting the produced machine learning model match what most doctors say about the subject.

We can read the tree visualization in Fig. 4 as follows. In the beginning, there were 373 samples (instances), 292 of class 0 and 81 of class 1. The entropy of the initial state was E=0.755. Then, the first partition of the samples into 2 groups was made by comparing the value of glucose with 139.5. With that, the entropy of the left group decreased and of the right group increased. The process continues up to depth 3.

5.2 Heart Disease

From Figs. 5 and 7, we can see that the most important features are chest pain type, thalessemia result of 'reversable defect' and ST depression. All these features make sense considering articles about the subject.

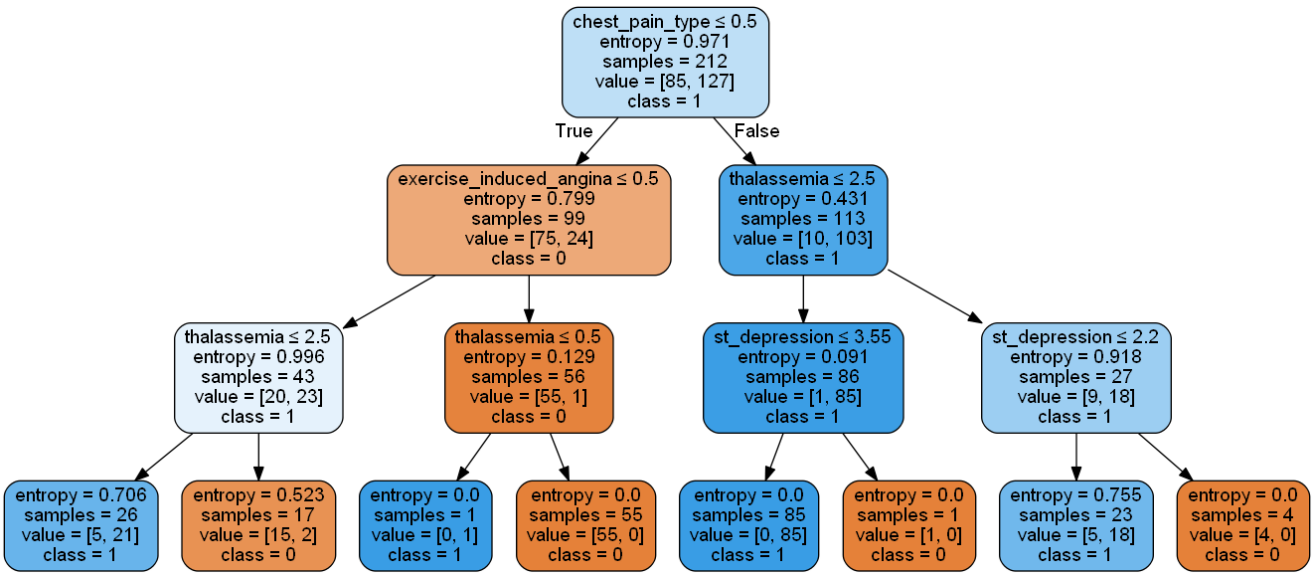

Figure 5: Extracted Rules from SVM for heart disease data set

Weight	Feature
0.2062 ± 0.0506	glucose
0.0323 ± 0.0253	bmi
0.0025 ± 0.0061	pedigree
0 ± 0.0000	age
0 ± 0.0000	insulin
0 ± 0.0000	skin
0 ± 0.0000	bp
-0.0037 ± 0.0290	pregnant

Figure 6: Feature weights for extracted rules from SVM for Pima Indians data set

Weight	Feature
0.0615 ± 0.0473	chest_pain_type
0.0396 ± 0.0493	thalassemia
0.0264 ± 0.0408	st_depression
0.0000 ± 0.0278	exercise_induced_angina
0 ± 0.0000	num_major_vessels
0 ± 0.0000	st_slope
0 ± 0.0000	max_heart_rate_achieved
0 ± 0.0000	rest_ecg
0 ± 0.0000	fasting_blood_sugar
0 ± 0.0000	cholesterol
0 ± 0.0000	resting_blood_pressure
0 ± 0.0000	sex
0 ± 0.0000	age

Figure 7: Feature weights for extracted rules from SVM for Heart Disease data set

We can read the tree visualization in Fig. 5 as follows. In the beginning, there were 212 samples (instances), 85 of class 0 and 127 of class 1. The entropy of the initial state was E=0.971. Then, the first partition of the samples into 2 groups was made by comparing the value of Chest Pain Type with 0.5. With that, the entropy of the left group increased and of the right group decreased. The process continues up to depth 3.

For both trees, it is important to note that some of the leaf nodes are overfitted and the entropy is E=0. Ideally, we want the leaf nodes to be as little randomized as possible for high accuracy and less overfitting. But, in this case, as we intended to use the decision tree to explain the SVM classifier, this overfitting can be considered a good result.

6 Conclusions

Although interpretable models provide crucial insight into why predictions are made, they impose some limitations. We argued that model-agnostic explanation systems provide a generic framework for interpretability that allows for flexibility in the choice of models and representations. In this article, we described the use of Decision Trees to generate rules from an SVM. We have shown that Decision Trees and permutation feature importance provide a reliable measure for assessing the quality of the extracted rules than the commonly used measures of accuracy. In addition, we have shown that, on the data sets, the extracted rules have at least an equivalent performance to the original SVM.

Once we develop better techniques to learn interpretable SVM Classifiers, we envisage the following important scenarios that require the system to provide an explanation: (i) *Explain pairs* – Given a chosen pair of different diagnosis from the data sets,

why is one item is classified as positive and the other as negative? (ii) *Explain item vs rest* - Given a diagnosis of interest, why is it classified as positive or negative. In the future, we would also like to address the problem of explainability for end-users.

In our point of view, as the community learns to advance its work collaboratively by combining ideas from different fields, the overall state of system explanation will improve dramatically, resulting in methods that build trust in machine learning systems and provide useful insight into black-box models operations enabling system behavior understanding and improvement.

References

Barakat, N. and Bradley, A. P. (2006). Rule extraction from support vector machines: Measuring the explanation capability using the area under the roc curve, *18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 2, pp. 812–815. https://doi.org/10.1109/ICPR.2006.1021.

Barakat, N. H. and Bradley, A. P. (2007). Rule extraction from support vector machines: A sequential covering approach, *IEEE Transactions on Knowledge and Data Engineering* **19**(6): 729–741. https://doi.org/10.1109/TKDE.2007.190610.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B. and Varoquaux, G. (2013). API scikit-learn project, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122.

Craven, M. W. and Shavlik, J. W. (1995). Extracting tree-structured representations of trained networks, *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, MIT Press, Cambridge, MA, USA, pp. 24–30. Available at http://dl.acm.org/citation.cfm?id=2998828.2998832.

Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation", *AI Magazine* **38**(3): 50–57. https://doi.org/10.1609/aimag.v38i3.2741.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Pedreschi, D. and Giannotti, F. (2018). A survey of methods for explaining black box models. Available at https://arxiv.org/abs/1802.01933.

Gunning, D. (2017). *Explainable Artificial Intelligence (XAI)*. Available at https://www.darpa.mil/attachments/XAIProgramUpdate.pdf.

Lipton, P. (1990). Contrastive explanation, *Royal Institute of Philosophy Supplement* **27**: 247–266. Available at https://philpapers.org/rec/LIPCE.

Miller, T. (2017). Explanation in artificial intelligence: Insights from the social sciences, *Cornell University* . Available at https://arxiv.org/abs/1706.07269.

Molnar, C. (2017). Interpretable machine learning. Available at https://christophm.github.io/interpretable-ml-book/.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016a). Model-agnostic interpretability of machine learning, *Cornell University* . Available at https://arxiv.org/abs/1606.05386.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016b). "why should I trust you?": Explaining the predictions of any classifier, *CoRR* **abs/1602.04938**. Available at https://dblp.org/rec/bib/journals/corr/RibeiroSG16.