Arg-XAI: a Tool for Explaining Machine Learning Results

Stefano Bistarelli[1], Alessio Mancinelli[1], Francesco Santini[1] and Carlo Taticchi[1,*]

[1]Dipartimento di Matematica e Informatica - Università degli Studi di Perugia, Via Luigi Vanvitelli, 1, 06123 Perugia, Italia

Abstract
The requirement of explainability is gaining more and more importance in Artificial Intelligence applications based on Machine Learning techniques, especially in those contexts where critical decisions are entrusted to software systems (think, for example, of financial and medical consultancy). In this paper, we propose an Argumentation-based methodology for explaining the results predicted by Machine Learning models. Argumentation provides frameworks that can represent and analyse logical relations between pieces of information to construct human-tailored rational explanations for a given problem. In particular, we use extension-based semantics to find the rationale behind a class prediction.

Keywords
Computational Argumentation, Machine Learning, Explainability

1. Introduction

The term *Explainable Artificial Intelligence* (*XAI*) refers to the principle by which the operating procedures and the results offered by intelligent computer systems are made understandable to human users [1]. The black box model used in Machine Learning (ML) is considered one of the major problems in the application of Artificial Intelligence (AI) techniques [2]: it makes machine decisions non-transparent and often incomprehensible even to experts or developers themselves, which reduces trust in ML and AI in general.

The need for explainability is exacerbated in critical contexts where the decisions made directly impact people's lives (e.g., financial plans or disease treatments). Understanding the choices made by AI algorithms is therefore of fundamental importance, not only to increase trust in AI but also to provide insights into the model itself and to carry out debugging operations [3]. Another reason for the strong interest in understanding the processes behind ML algorithms is the increase in public sensitivity towards privacy [4].

Among the various approaches to explanation, argumentative models play a fundamental role in the literature relating to AI and the social sciences, given their dialectical nature, which allows linking applications to the human beings who develop and use them [5]. Argumentation in XAI is supported by the solid foundation and flexibility provided by the wide variety of frameworks offered in the literature. For instance, Abstract Argumentation Frameworks (AFs) [6] allow for specifying arguments and dialectical relations between them; a different paradigm can also be used to represent both conflict and support relations [7]. Furthermore, AFs are also endowed with semantics for evaluating arguments' acceptability. Therefore, there are two main advantages to using an argumentative approach for understanding the behaviour of black box models. First, it allows for explanations that can be assimilated and evaluated following the natural declination of human reasoning. Indeed, arguing is a primary means by which people reason about decisions to be made in real life (we can argue both with others and with ourselves), and Argumentation paradigms mimic the human way of thinking. Then, regarding implementation aspects, many tools and formal models are provided by Computational Argumentation that are already predisposed for automation and can therefore serve as a basis for developing and providing new explanation techniques [8].

This paper describes Arg-XAI [9, 10], a tool for the argumentative interpretation of the training process and the results predicted by Machine Learning models.

We take in input a dataset characterised by a certain number of features, one of which represents the class of the record, and we build a Bipolar Argumentation Framework (BAF) [7] whose arguments consist of a subset of selected features related to each other (with supports/attacks) in accordance with their correlation value. We then use Argumentation semantics to evaluate the acceptability of the arguments in the BAF: starting from justified arguments, we can build an explanation tree that shows motivations behind the attribution of a certain class to a given record. Our proposal aims to offer explanations as reasoning processes, and since we make assumptions neither on the dataset nor the algorithm used, the procedure can be applied to existing models without further adjustments.

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
*Corresponding author.
stefano.bisterelli@unipg.it (S. Bistarelli); alessio.mancinelli@unipg.it (A. Mancinelli); francesco.santini@unipg.it (F. Santini); carlo.tatichi@unipg.it (C. Taticchi)
0000-0001-7411-9678 (S. Bistarelli); 0000-0002-3935-4696 (F. Santini); 0000-0003-1260-4672 (C. Taticchi)

2. Preliminaries

Machine Learning is a branch of Artificial Intelligence devoted to automating analytical model building. In particular, ML offers techniques capable of making predictions or decisions without explicitly writing an ad hoc program. For example, an ML algorithm [11] can recognise new and never seen samples by extracting patterns from a given dataset and using them to approximate a function that generalises the data. There are three primary ML approaches: supervised, unsupervised and reinforced learning. This paper focuses on supervised learning, in which the training data (a set of examples used to train the model) contains both the input and the desired outputs. The model can thus be learned by optimising a function that predicts the output associated with the new input. Classification algorithms are supervised learning algorithms that address the problem of associating each input with the class it belongs to. The model able to correctly classify an input is learned during the training phase and consists of a function $m : F \to C$ where F is the set of features (measurable properties of a record) and C the possible classes.

On the other hand, Argumentation is an interdisciplinary field that aims to understand and model the natural human fashion of reasoning, allowing one to deal with uncertainty in non-monotonic (defeasible) reasoning. It gives a qualitative, logical evaluation of sets of interacting arguments called extensions. An Abstract Argumentation Framework [6] is a pair $\langle Arg, R\rangle$ where Arg is a set of arguments and R is a binary attack relation on Arg. For two arguments $a, b \in Arg$, $(a, b) \in R$ represents an attack from a directed to b. A generalisation of AFs is provided by Bipolar Argumentation Frameworks [7], which admit two different types of relations between arguments: attack and support relations.

Given an AF, we are interested in establishing which are the acceptable arguments according to a certain semantics, namely a selection criterion. Non-accepted arguments are rejected. Extension-based semantics (e.g., admissible, complete, stable, semi-stable, preferred, and grounded) have been introduced [6, 12] that reflect qualities which are likely to be desirable for "good" subsets of arguments. In particular, the semi-stable semantics has properties that make it suitable for constructing explanations: it always exists (contrary, for example, to the stable semantics, which may not admit any extension), and it provides a solid justification for accepted arguments since it expresses a definite opinion on the largest possible set of arguments [12]. A labelling semantics [12] can be used to increase expressiveness by assigning a label (between in, out and $undec$) to the arguments: an argument is labelled in if all its attackers are labelled out, and it is labelled out if at least an in node attacks it; in all other cases, the argument is labelled $undec$.

3. Explanation Procedure

The proposed approach gives an argumentative interpretation of the training phase and the answers provided by machine learning models for classification. We start from an input dataset composed of n records, each with a number of features also including the class it belongs to. The goal is to build a BAF showing the dialectical reasoning behind the assignment of a certain class to a given record. The used procedure consists of the following tasks: *Dataset Clustering*, *BAF Generation*, *Breaking Symmetries*, *Attack and Support Closure* and *Explanation Tree Construction*.

3.1. Dataset Clustering

Starting from the input dataset, we create a new clustered dataset in which numerical features are split into categories that group ranges of values to obtain a more appropriate and concise explanation. To improve efficiency, we binarise categorical features: a new column is generated in the clustered dataset for each possible categorical feature value. For instance, if the feature A can take three values 0, 1 and 2, we add three new columns, respectively for *A=0*, *A=1* and *A=2*. If in a certain record of the original dataset, the feature A takes value 0, then the corresponding record *A=0* in the clustered dataset is set to 1, while *A=1* and *A=2* are both set to 0.

Generating a new column for every possible value is not feasible for numerical features. In this case, we use a methodology based on entropy [13] to find the best *split*. Following this approach, the data is partitioned into subsets on which the class entropy (amount of information needed to specify the classes in the partition) is computed. The best split is the one that minimizes entropy.

3.2. BAF Generation

The BAF is generated starting from a subset of arguments chosen from the list of features that will be used for the explanation. To prevent arguments deriving from the same feature from being in the same extension, we add an attack with a weight equal to -1 between all the arguments generated after the splitting of a feature. Then, we compute the correlation matrix among the features to determine what kind of relation (support or attack) exists between the arguments. For this task we use rank coefficients as the Kendall [14], Pearson [15] and Spearman [16] ones. If the correlation value between two arguments is negative, we add an attack between them; if it is positive, we add a support relation. In both cases, the weight of the relation is equal to the correlation value. All the attacks and supports in such an assembled BAF are symmetrical at this stage.

3.3. Breaking Symmetries

Given the correlation matrix, we apply a procedure that removes symmetric edges from the BAF. Breaking the symmetry of the obtained framework is crucial, as we want to detect causality between arguments in the BAF. Such a causal relation cannot be studied only relying on the correlation matrix (which is symmetrical by construction); hence we consider the conditional probability [17] between the features, which expresses how likely an event is to happen given that another event has already happened. This kind of probabilistic reasoning has already been successfully adopted in the literature (e.g., in a paper by Timmer et al. [18]) to extract probabilistically supported arguments from a Bayesian network. Given two features A and B, we consider the conditional probability of A given B (written $P(A|B)$) and the conditional probability of B given A ($P(B|A)$). If $P(A|B) > P(B|A)$, then the (attack or support) relation from B to A is removed since A is more probable to happen. In practice, we also consider a threshold not to remove features with similar conditional probabilities. In the opposite case, when $P(B|A) > P(A|B)$, we would have removed the relation from A to B. Note that we only remove relations between arguments that do not come from the same feature. Indeed, we do not want to remove the symmetrical attack with weight -1 we added in the previous step between arguments obtained through a *split*.

3.4. Attack and Support Closure

We translate the considered BAF into a classical AF with attack relations only to obtain the list of semi-stable extensions and compute their probability of being admissible. Indeed a tool for computing such a probability directly on BAFs is not currently available in the literature. The translation phase begins with the support relations closure: given three arguments A, B and C, if $supp(A, B) = x$ and $supp(B, C) = y$, we add a support relation from A to C such that $supp(A, C) = x * y$. The next step is the attack relations closure. First, we look for triples of arguments A, B and C with $supp(A, B) = x$ and $att(B, C) = y$, and we add an attack relation from A to C such that $att(A, C) = x*y$. Then, for all arguments A, B and C with $att(A, B) = x$ and $supp(C, B) = y$, we add an attack $att(C, A) = -y$. At this point, we delete all the support relations from the modified BAF, thus obtaining a classical AF [19].

3.5. Explanation Tree Construction

We are now able to compute the set of acceptable arguments. The choice of the semantics falls on the semi-stable one for the reasons mentioned in Section 2. In our BAF, each relation between two features A and B is endowed with a probability corresponding to the value in the correlation matrix between A and B. Such probability represents uncertainty over the topology of the graph. We use the constellation approach [20] to compute the probability of a set of arguments to be a semi-stable extension: this gives an idea of the plausibility of each possible explanation. Due to computational issues, we use a workaround to decrease the complexity of the operation (see Section 4 for implementation details).

Finally, we build the explanation tree. We show an example in Figure 1, in which the root node A represents the attribution of the class we want to explain. We can add to the tree nodes that attack/support the root: the explanation can be produced by showing features that either support the class attribution or are against it or both. In Figure 1, the root has a supporting feature B, which must be an accepted argument belonging to an extension found in the previous step. Argument C, instead, is attacking A and must be defeated by another argument supporting the root (D in our case).

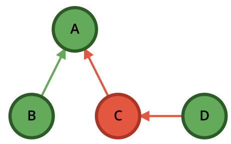

Figure 1: Example of explanation tree for the argument A.

4. Experiments and Validation

To validate the proposed approach, we use the "Titanic" dataset[1] containing records about people involved in the Titanic disaster (see Table 1 for the list of features).

Table 1
Titanic dataset features.

Feature	Values	Description
Pclass	$1, 2, 3$	Ticket class
sex	$0, 1$	passenger gender
SibSp	$0 - 8$	# of siblings/spouses
Parch	$0 - 6$	# of parents/children
Embarked:	C, Q, S	port of embarkation
Survived:	$0, 1$	passenger survival
Age	$0.17 - 76$	passenger age
Fare	$0 - 512$	passenger fare

The class to predict is *Survived*, which determines whether a person survived the disaster (value 1) or not (value 0). In this example, we want to find an explanation for the class *Survived=1*. First, we compute the

[1]The Titanic dataset is taken from https://www.kaggle.com.

correlation matrix for the selected features and obtain a BAF with only symmetrical relations. The symmetry is broken through the conditional probability computed for arguments which attack/support each other. When the difference in conditional probability is minimal, however, we want to maintain both relations because there is not enough confidence in determining which feature implies the other. Hence we specify a correlation threshold which must be reached in order to remove one of the two symmetrical relations. If not, both relations remain in the BAF.

To help the user choose the correct correlation threshold, we implemented a procedure that finds the minimum values that guarantee the graph to remain connected (in fact, we want the explanation dependent on all the selected features). In this example, we use a correlation threshold of 0.17. A percentage threshold is also used to manipulate the number of attacks and supports to remove. Let m and n be the number of records with $A = 1$ and $B = 1$, respectively. With a threshold of $x\%$, the relation from B to A is removed only if the condition $\frac{m \cdot x}{100} > n$ is satisfied. We choose the minimum values possible that keep the graph connected, which is, in this case, a removal percentage of 30%.

Suppose to have the following data:

- Number of records set to 1 for the feature A: 507
- Number of records set to 1 for the feature B: 117
- Relation removal percentage: 30%

We first compute 30% of 507, that is 152. Since $152 > 117$, the relation from B to A is removed. We obtain a BAF with 26 arguments and 179 relations (131 attacks and 48 supports) within a single connected component. The "main" argument visualised at the top of the BAF represents the assignment of the class *Survived=1*.

To identify the set of arguments more likely to be accepted, we compute the semi-stable extensions through the ConArg solver[2] and then use the tool described in [21] to find, for each semi-stable extension, its probability of being admissible. Since we cannot compute classical semantics directly on a BAF, we first translate it into an AF by applying the transitive closure and the removal of supports of Section 3.4, obtaining an AF with 265 attack relations. We can then compute the set of semi-stable extensions and, for each of them, the probability of being admissible.

For example, Extension (1) is a semi-stable extension of the generated AF, which is also an admissible extension with probability 1 (the highest possible) and contains the argument *Survived=1*.

$$\begin{gathered}\textit{Fare}{\geq}\textit{10.4812},\ \textit{Age}{<}\textit{0.96},\ \boldsymbol{\textit{Survived=1}},\\ \textit{Embarked=C},\ \textit{Sex=0},\ \textit{Parch=1},\ \textit{SibSp=1},\ \textit{Pclass=1}\end{gathered} \tag{1}$$

[2]ConArg website: https://conarg.dmi.unipg.it.

Extension (1) represents a good explanation of why the individual survives since, being semi-stable, it provides the maximal number of arguments justifying the class *Survived=1*. Finally, starting from arguments of the selected extension, we produce the explanation tree of Figure 2, where accepted arguments are labelled in and highlighted in green, while rejected ones are labelled out and highlighted in red. Possible $undec$ arguments (not present in our example) would have been removed as they are not helpful in the explanation. Note that argument *Age*$<$*0.96* is not used in Figure 2 since it is not in the same connected component as *Survived=1*.

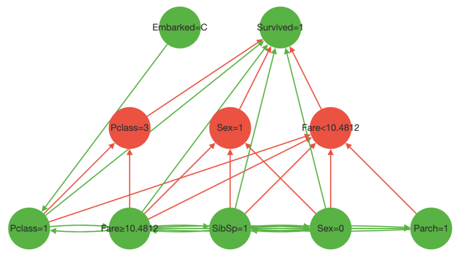

Figure 2: An explanation tree for the class *Survived=1* of the Titanic dataset. To enhance the presentation, weights on edges are not displayed.

Looking at the obtained explanation, we can conclude, for instance, that the person in question survived because "she is a woman (*Sex=0*), with a paid ticket (*Fare*$\geq$*10.4812*) and travelling first class (*Pclass=1*)". Indeed, arguments representing those features in Figure 2 attack other arguments that are against the assignment of the class *Survived=1*, standing in turn for being male (*Sex=1*) and having a third-class ticket (*Pclass=3*) with a low fare (*Fare*$<$*10.4812*). From Figure 2, we could also assume that most of the first class passengers boarded from Cherbourg (*Embarked=C*), indeed argument *Embarked=C* supports *Pclass=1*.

To validate the proposed explanation technique, we conducted experiments using decision trees [22] and LIME [23].

4.1. Validation via Decision Tree

Decision trees represent a classification technique that involves walking a tree from the root to its leaves. A learning algorithm decides the shape of the tree and assigns splitting features to individual nodes. If the test on a node is true, then we move to the left branch, otherwise to the right. Given a certain record, we follow the path from the root to a leaf node corresponding to the assigned class. Since we want to minimise the number of arguments used for the explanation, we use entropy to split the numerical features. It follows that our split will not exactly coincide with the numerical range found by the decision tree. The trees are built using *grid search* [24], which consists of performing an exhaustive search of optimised parameters (in a selected range). The classifier's performance is evaluated for each combination of parameters, making this process expensive in computational terms but able to guarantee good results. We want to check whether arguments corresponding to the leaves of the decision tree (that is, the assigned classes) belong to a semi-stable extension. For example, if the class *Survived = 1* is a leaf in the decision tree, then we expect the argument corresponding to that class to be accepted in our BAF with respect to the semi-stable semantics.

To validate the explanation provided by Extension (1), we refer to the decision tree of Figure 3 trained on the Titanic dataset. Starting from the root, we proceed to the left subtree since the feature *sex* is set to 0 in Extension (1). Then we check the feature *Pclass*, which has value 1 in our extension, and we proceed again to the left branch. The last feature to analyse is *Fare* and, regardless of whether or not the split condition occurs, the class in the leaves is *Survived = 1*, which also belongs to the extension.

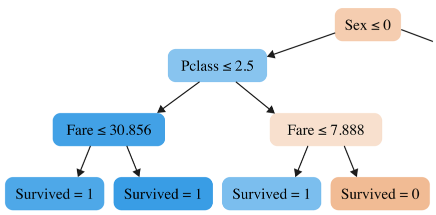

Figure 3: Left branch of the decision tree model trained on the original Titanic dataset.

4.2. Validation via LIME

The LIME algorithm is used for explaining the predictions of any classifier by approximating it with a model that can be easily interpreted. LIME generates new records using small variations of the instance taken as input. On this new dataset, LIME trains an interpretable model (logistic regression in our case). The new records are then labelled using the original classifier, and the similarity distance between the original predictions and the new ones is computed to explain the local behaviour of the analysed black box.

Taking into account the Titanic dataset, consider first a record consisting of the following features: *Pclass=1*, *Age=24*, *SibSp=1*, *Parch=1*, *Fare=100*, *Sex=0* and *Embarked=C*. The explanation provided by LIME for passenger survival (*Survived=1*) is shown in Figure 4. We can see that the most relevant features, i.e., *Sex=0* and *Pclass=1*, are also arguments belonging to Extension (1).

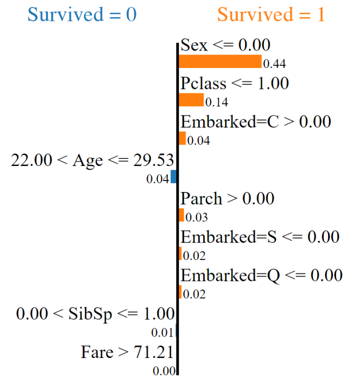

Figure 4: A LIME explanation for the class *Survived=1*.

5. Web Interface

This section describes the web interface[3] we developed to facilitate the use of the proposed method. The user is first required to specify a dataset to be analysed. Clicking on the "Load Dataset" button, all the features contained in the dataset are presented in a multiple-choice select to enable specifying the class to justify, and the categorical and numerical features. Then, the user chooses whether to force the split of the numerical features and selects the correlation index to use. Clicking on the "Submit" button leads to the second step.

At this point, the user selects the features to use for the explanation and the interface displays the BAF obtained after the transitive closure of attack and support relations. The user can see the resulting BAF, including the updated number of edges and connected components. Then, the AF obtained from the BAF after removing the supports is visualised.

In the last panel, the user can see the BAF before the closure phase, the list of semi-stable extensions, and their probability. Each extension in the list is accompanied by a "Highlight" button which produces an explanation tree for the analysed extension.

6. Conclusion and Future Work

We introduce a tool for an argumentative interpretation of the answers provided by Machine Learning models, proposing AFs to obtain a dialectical explanation. To this end, we devise a procedure which allows the construction of an explanation tree representing the *dialectical reasoning* among the features of the analysed problem and explaining why a particular class is assigned. We also provide a web interface to ease the use of the tool.

[3] Arg-XAI web interface: http://arg-xai.dmi.unipg.it.

In future work, alternative techniques could be applied to break the symmetry of the graph and obtain a causal relationship between arguments. It would also be interesting to conduct studies with more complex datasets (e.g., containing categorical features with a large number of possible values). In this case, the produced BAF may be too large to explain the assignment of a certain class, and particular attention should be paid to simplifying the final explanation tree. Notions of symmetry and interchangeability between arguments, as well as NLP-generated textual explanations, could be used for this purpose. We could also apply Subjective Logic models [25] and use the weights on the BAF's edges to obtain a better explanation instead of just using them for computing the probability of the extensions. Finally, we plan to implement other extension-based semantics in addition to the semi-stable one. A qualitative/quantitative comparison could also be made between promising semantics.

References

[1] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[2] W. J. von Eschenbach, Transparency and the black box problem: Why we do not trust ai, Philosophy & Technology 34 (2021) 1607–1622.

[3] T. Kulesza, M. M. Burnett, W. Wong, S. Stumpf, Principles of explanatory debugging to personalize interactive machine learning, in: IUI, ACM, 2015, pp. 126–137.

[4] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, Z. Lin, When machine learning meets privacy: A survey and outlook, ACM Comput. Surv. 54 (2022) 31:1–31:36.

[5] K. Cyras, A. Rago, E. Albini, P. Baroni, F. Toni, Argumentative XAI: A survey, in: IJCAI, ijcai.org, 2021, pp. 4392–4399.

[6] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (1995) 321–358.

[7] C. Cayrol, M. Lagasquie-Schiex, On the acceptability of arguments in bipolar argumentation frameworks, in: ECSQARU, Springer, 2005, pp. 378–389.

[8] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, S. Villata, Towards artificial argumentation, AI Mag. 38 (2017) 25–36.

[9] S. Bistarelli, A. Mancinelli, F. Santini, C. Taticchi, An argumentative explanation of machine learning outcomes, in: COMMA, IOS Press, 2022, pp. 347–348.

[10] S. Bistarelli, A. Mancinelli, F. Santini, C. Taticchi, Arg-XAI: a tool for explaining machine learning results, in: ICTAI, IEEE, 2022. To appear.

[11] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2020.

[12] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, Knowl. Eng. Rev. 26 (2011) 365–410.

[13] U. M. Fayyad, K. B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: IJCAI, Morgan Kaufmann, 1993, pp. 1022–1029.

[14] W. B. Kendall, A new algorithm for computing correlations, IEEE Trans. Computers 23 (1974) 88–90.

[15] W. Kirch (Ed.), Pearson's Correlation Coefficient, Springer Netherlands, Dordrecht, 2008, pp. 1090–1091.

[16] Y. Dodge, Spearman Rank Correlation Coefficient, Springer New York, New York, NY, 2008, pp. 502–505.

[17] M. Borovcnik, Conditional probability–a review of mathematical, philosophical, and educational perspectives, Topic Study Group 11 (2012).

[18] S. T. Timmer, J. C. Meyer, H. Prakken, S. Renooij, B. Verheij, A two-phase method for extracting explanatory arguments from bayesian networks, Int. J. Approx. Reason. 80 (2017) 475–494.

[19] G. Boella, D. M. Gabbay, L. W. N. van der Torre, S. Villata, Support in abstract argumentation, in: COMMA, IOS Press, 2010, pp. 111–122.

[20] H. Li, N. Oren, T. J. Norman, Probabilistic argumentation frameworks, in: TAFA, Springer, 2011, pp. 1–16.

[21] S. Bistarelli, T. Mantadelis, F. Santini, C. Taticchi, Using metaproblog and conarg to compute probabilistic argumentation frameworks, in: AI^3@AI*IA, CEUR-WS.org, 2018, pp. 6–10.

[22] J. R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.

[23] M. T. Ribeiro, S. Singh, C. Guestrin, "why should I trust you?": Explaining the predictions of any classifier, in: KDD, ACM, 2016, pp. 1135–1144.

[24] 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, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[25] A. Jøsang, Conditional reasoning with subjective logic, J. Multiple Valued Log. Soft Comput. 15 (2009) 5–38.