Arg-XAI: a Tool for Explaining Machine Learning Results

Stefano Bistarelli*
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
stefano.bistarelli@unipg.it

Alessio Mancinelli
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
alessio.mancinelli@unipg.it

Francesco Santini*
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
francesco.santini@unipg.it

Carlo Taticchi*
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
carlo.taticchi@unipg.it

2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3503-9744-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTAI56018.2022.00037

***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 be used to represent and analyse logical relations between pieces of information, serving as a basis for constructing human tailored rational explanations to a given problem. In particular, we use extension-based semantics to find the rationale behind a class prediction.**

***Index Terms*—Computational Argumentation, Machine Learning, Explainability**

I. 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 [28]: 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 have a direct impact on 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 [19]. Another reason for the strong interest in understanding the processes behind ML algorithms is the increase in public sensitivity towards privacy [21].

Among the various approaches to explanation, argumentative models play a fundamental role both in the literature relating to AI and in the social sciences, given their dialectical nature which allows linking applications to the human beings who develop and use them [11]. The use of 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) [13] allow for specifying arguments and dialectical relations between them; a different paradigm can also be used to represent both conflict and support relations [9]. AFs are also endowed with a set of semantics for evaluating the acceptability of arguments themselves. Therefore, there are two main advantages to use 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 the development and provision of new explanation techniques [2].

In this paper, which extends and supersedes the preliminary work of [4], we provide an argumentative interpretation of both 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) [9] 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

This work has been partially supported by: GNCS-INdAM, CUP E55F22000270001; Project RACRA - funded by Ricerca di Base 2018-2019, Univeristy of Perugia; Project BLOCKCHAIN4FOODCHAIN: funded by Ricerca di Base 2020, Univeristy of Perugia; Project DopUP - REGIONE UMBRIA PSR 2014-2020; Project GIUSTIZIA AGILE, CUP: J89J22000900005.

*The author is a member of the INdAM Research group GNCS and of Consorzio CINI.

2375-0197/22/$31.00 ©2022 IEEE
DOI 10.1109/ICTAI56018.2022.00037

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.

Differently from other methodologies in the literature [11], our proposal aims to offer explanations as reasoning processes, rather than evidences that make outputs of ML algorithms credible. Moreover, it only depends on the dataset and not on the ML technique used for the classification task. Finally, since we make assumptions neither on the dataset nor on the algorithm used, the procedure we present can be applied on existing models without the need for further adjustments.

II. Preliminaries

Machine Learning is a branch of Artificial Intelligence devoted to the automation of analytical model building. In particular, ML offers techniques capable of making predictions or decisions without the need to explicitly write an ad hoc program. An ML algorithm [26] 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 itself. There are three main approaches to ML, namely supervised, unsupervised and reinforced learning. In this paper, we focus 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 new input. Classification algorithms, then, are a type of 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 human natural fashion of reasoning, allowing one to deal with uncertainty in non-monotonic (defeasible) reasoning, and it is used to give a qualitative, logical evaluation to sets of interacting arguments, called extensions. An Abstract Argumentation Framework [13] 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** [9], 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. Different kinds of extension-based semantics (e.g., admissible, complete, stable, semi-stable, preferred, and grounded) have been introduced [3], [13] 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 [3]. A labelling semantics [3], [8] 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 $undec$.

III. Explanation through Argumentation

The proposed approach consists in giving an argumentative interpretation of both training phase and 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 steps listed below.

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, so as to obtain a more appropriate and concise explanation.
2) **BAF Generation.** We build a BAF based on the correlation matrix computed among the features.
3) **Breaking BAF's Complete Symmetry.** Given the correlation matrix, we apply a procedure that removes symmetric edges from the BAF to establish a causal relationship between features.
4) **Computing Extensions and Explanation Trees.** We compute the semi-stable extensions of the previously obtained framework and we build the explanation tree for the selected class.

In the following, we provide a detailed description for each of these steps. We begin by selecting a dataset to analyse, together with the problem's class and a list of categorical and numerical features.

A. Dataset Clustering

To improve efficiency, we binarise categorical features: for each possible value of a categorical feature a new column is generated in the clustered dataset. 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, instead, for numerical features. In this case, we use a methodology based on entropy [14] 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. Various techniques were tested, including the subdivision of features through the *Silhouette* coefficient [16], but the number of splits generated,

and therefore of arguments with a different numerical range, became too high to be represented in an understandable fashion within the AF. The generation of new records with values of either 1 or 0, occurs in the same way as described for categorical features. Note that entropy-based splits cannot be found when there is no dependence between the analysed feature and the assigned class. In this case, the split can still be forced, and the number of intervals will always be two, i.e., the minimum possible.

B. BAF Generation

Following the features splitting *phase*, a BAF is generated starting from a subset of arguments, chosen from the list of features, that will be used for the explanation. The correlation matrix can be computed by using rank coefficients as the Kendall [17], Pearson [18] and Spearman [12] ones.

We start building the BAF by adding an attack with weight equal to -1 between all the categorical and numerical features generated from the splitting of a same feature. For example, if the features B, C and D are created by splitting the feature A, then we add symmetrical attacks with weight -1 between arguments B, C and D. In this way, we prevent arguments coming from the same feature to be in the same extension. Then, to determine what kind of relation (between support and attack) exists between two arguments X and Y, we look at their correlation value. If such a value is negative, we add an attack between X and Y. If it is positive, we add a support relation. In both cases, the weight of the relation is equal to the correlation value between X and Y. At this stage, all the relations in such an assembled BAF, regardless of their type, are symmetrical.

C. Breaking BAF's Complete Symmetry

Breaking the symmetry of the obtained framework is crucial, as we want to detect causality between arguments in the BAF, that is we want to know which of two related features implies the other. Such a causal relation cannot be studied only relying on the correlation matrix (which is symmetrical by construction), hence we consider the conditional probability [7] 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. [27]) 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 the 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*.

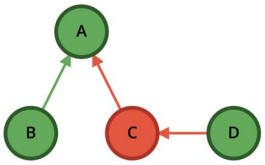

Fig. 1: Example of explanation tree for argument A.

D. Attack and Support Relations Closure

To obtain the list of semi-stable extensions and compute their probability of being also admissible, we first translate the considered BAF into a classical AF where only attack relations are allowed [6]. 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 $supp(A, C) = x * y$. Note that the support from A to C is only added if its weight is greater than the selected threshold. The next step is the **attack relations closure** and comprises two distinct phases. 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 [6].

E. Computing Extensions and Explanation Trees

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 II. 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 of being 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 IV 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).

IV. Experiments and Validation

For our validation, we use two datasets: "Titanic" and "Heart Attack Analysis & Prediction"[1]. The former contains records relating to people involved in the Titanic disaster (see Table I for the list of features). The class to predict is *Survived*, which determines whether a person survived the disaster (value 1) or not (value 0). In the Heart Attack Analysis & Prediction dataset, instead, each record represents a person who may have a heart attack. The features are shown in Table II. In this case, the class to predict is *Heart Attack*, whose possible values are: 0 (the person in question is less likely to have a heart attack) and 1 (the person in question is more likely to have a heart attack).

We now present an example of explanation obtained through the procedure described in Section III using records from the Titanic dataset. First, we select the problem class (for which we want to build an explanation), the categorical features and the numerical ones. We want to find an explanation for the class *Survived=1* using all the dataset features except *Survived=0* (it is also possible to use a different subsets of the available features).

Then, we compute the correlation matrix for the selected features (note that, due to the structure of the dataset, the Kendall, Pearson and Spearman coefficients all produce the same correlation matrix) 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 in choosing the correct correlation threshold, we implemented a procedure that finds the minimum values which guarantees the graph to remain connected (in fact, we want the explanation to be 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 that have $A = 1$ 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. To give an example, 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 choose the minimum values possible that keep the graph connected, which is, in this case, a removal percentage of 30%. 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 is that representing the assignment of the class *Survived=1*.

To identify the set of arguments which are more likely to be accepted, we compute the semi-stable extensions and then we use the tool described in [5] to find, for each of them, 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 III-D, obtaining an AF with 265 attack relations. We can then proceed to 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*.

$$\begin{array}{c} \textit{Fare}{\geq}\textit{10.4812},\ \textit{Age}{<}\textit{0.96},\ \textbf{\textit{Survived=1}}, \\ \textit{Embarked=C},\ \textit{Sex=0},\ \textit{Parch=1},\ \textit{SibSp=1},\ \textit{Pclass=1} \end{array} \quad (1)$$

Extension (1) represents a good explanations 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*.

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≥10.4812*) and

[1]Both datasets are taken from https://www.kaggle.com.

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

TABLE I: Titanic dataset features.

Feature	Values	Type	Description
sex	$0, 1$	categorical	patient gender
cp	$0, 1, 2, 3$	categorical	chest pain type
fbs	$0, 1$	categorical	fasting blood sugar level
restecg	$0, 1, 2$	categorical	electrocardiographic results
exng	$0, 1$	categorical	exercise-induced angina
caa	$0, 1, 2, 3, 4$	categorical	# of major blood vessels
thall	$0, 1, 2, 3$	categorical	Thalium Stress Test result
Heart Attack	$0, 1$	categorical	patient heart attack
age	$29 - 77$	numerical	patient age
trtbps	$94 - 200$	numerical	resting blood pressure
chol	$126 - 564$	numerical	cholesterol
thalach	$71 - 202$	numerical	maximum heart rate
oldpeak	$0 - 6.2$	numerical	previous peak reached

TABLE II: Heart Attack dataset features.

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

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 three different methods: decision trees [24], rule based classifiers [22] and LIME [25].

A. Validation via Decision Tree

Decision trees represent a classification technique that involves walking a tree from the root to its leafs. 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, which corresponds to the class to be assigned. 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* [23], that consists of performing an exhaustive search of optimised parameters (in a selected range). The performance of the classifier are 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.

For the class *Survived=0*, on the other hand, we can find the following semi-stable extension which is also admissible with probability 1.

Fare<10.4812, Age≥0.96, ***Survived=0****, Embarked=Q, Sex=1, Parch=0, SibSp=0, Pclass=3*

An explanation tree against survival is shown in Figure 4. In this case, argument *Embarked=Q* is not considered in the tree, since it is not in the same connected component as *Survived=0*. Walking again the decision tree of Figure 3 we verify that a leaf with class *Survived=0* can be reached.

B. Validation via Rule-Based Classifier

Rule-based classifiers are used to solve classification problems through the construction of condition-action rules. A class is assigned to a certain record by comparing the features of that record with the premises of each rule. The set of rules can be obtained following different methods. We use the RIPPER algorithm [10] to derive a set of rules from the Heart Attack Analysis & Prediction dataset.

The rules generated by RIPPER for the class *Heart Attack=1* are the following.

thall=2 $\wedge$ *caa=0* $\wedge$ *slp=2*
exng=0 $\wedge$ *caa=0* $\wedge$ *sex=0*
exng=0 $\wedge$ *thall=2* $\wedge$ *cp=2*
caa=0 $\wedge$ *thall=2* $\wedge$ *sex=1*
trtbps=130.0-138.0 $\wedge$ *chol=187.0-207.0*

We compare the features in the rules with the arguments belonging to the following semi-stable extension, obtained through the procedure of Section III.

oldpeak<1.7, thalach≥147.5, chol<245.5, trtbps<107, age<54.5, ***Heart Attack=1****, thall=2, caa=0, slp=2, exng=0, restecg=1, fbs=1, cp=2, sex=0*

Such extension contains the argument *Heart Attack=1* and it is also admissible with probability 1. As we can notice, almost all the arguments belonging to the extension are also part of the rules found by RIPPER, with the only exception of $sex = 1$.

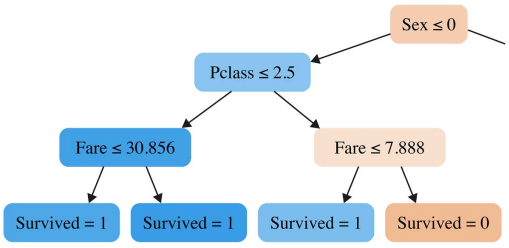

Fig. 3: Decision tree model trained on the original Titanic dataset. Only the left branch is shown.

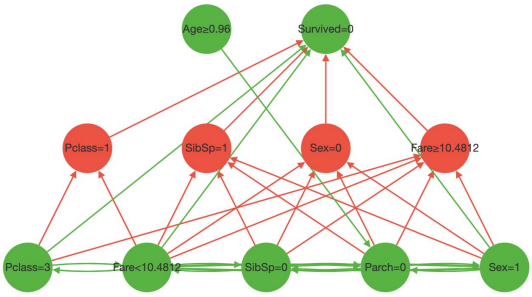

Fig. 4: An explanation tree for the class *Survived=0* of the Titanic dataset. To enhance the presentation, weights on edges are not displayed.

In the opposite case, that is when the predicted class is *Heart Attack=0*, we found the following rules.

cp=0 $\wedge$ *thall=3*
slp=1 $\wedge$ *exng=1* $\wedge$ *trtbps=155.0-180.0*
sex=1 $\wedge$ *slp=1* $\wedge$ *caa=1*
cp=0 $\wedge$ *chol=274.0-300.0*
caa=1 $\wedge$ *cp=0*

The most probable admissible extension (with a probability of 0.737364) among the semi-stable ones for the class *Heart Attack=0* is as follows.

oldpeak$\geq$*1.7, thalach*$<$*147.5, chol*$\geq$*245.5, trtbps*$\geq$*107, age*$\geq$*54.5,* ***Heart_Attack=0****, thall=3, slp=1, exng=1, fbs=1, cp=0, sex=1*

Arguments *chol*$\geq$*245.5*, *trtbps*$\geq$*107*, *thall=3*, *slp=1*, *exng=1*, *cp=0* and *sex=1* are also features used in RIPPER rules to identify records with class *Heart Attack=0*.

C. 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) and 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 5. We can see that the most relevant features, i.e., *Sex=0* and *Pclass=1*, are also arguments belonging to Extension (1).

We provide one last example using a record from the Titanic dataset with the features *Pclass=3, Age=56, SibSp=3, Parch=5, Fare=10, Sex=1 and Embarked=Q*. This time we want to find an explanation for the class *Survived=0*. As we can see from Figure 6, LIME detects *Sex=0, Pclass=3* and *Age=24* as the first three features that have most influenced the classification of the record. Those three features are also arguments of the following extension, which is semi-stable and also admissible with probability equal to 1.

Fare$<$*10.4812, Age*$\geq$*0.96, Survived*=0*, Embarked=Q,*
Sex=1*, Parch*=0*, SibSp*=0*, Pclass*=3

V. Web Interface

In addition to the Python code used to implement the steps described above, a user interface[2] was created to facilitate the use of the proposed method. In this section we describe how this interface works.

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 into a multiple-choice select for enabling the user to select the problem class, the

[2]Tool web interface: http://arg-xai.dmi.unipg.it.

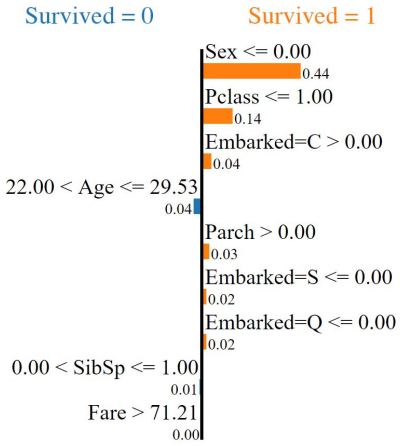

Fig. 5: A LIME explanation for the class *Survived=1*.

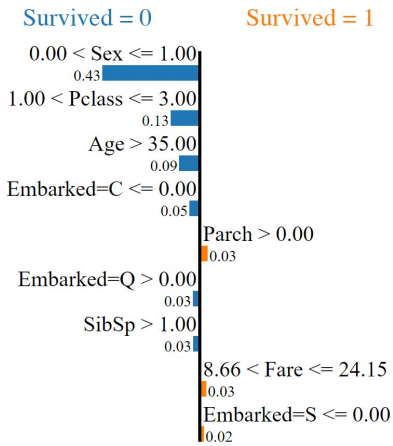

Fig. 6: A LIME explanation for the class *Survived=0*.

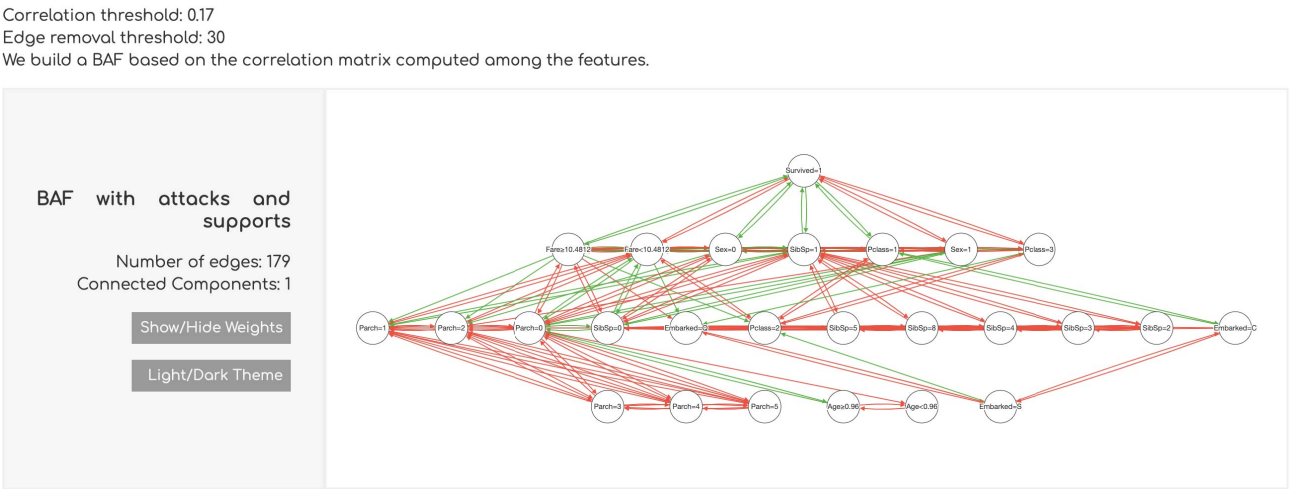

Fig. 7: Visualisation of the BAF obtained from the user input.

categorical features and the numerical ones. Subsequently the user can also choose whether or not to force the split of the numerical features through the entropy index, and select the correlation index to use. Once those choice are made, the user clicks the "Submit" button, and proceeds to the second step.

Then, the user is asked to select the set of features to be used for the explanation. Indeed, it is possible to use a representative subset of features, and not always all of them. Furthermore, the user has to select the main feature for which he wants to obtain the explanation: it will be inserted as the root node of the generated BAF. Once all the parameters have been selected, the minimum correlation threshold is calculated to keep the graph connected, including the minimum and maximum value within the correlation matrix. The example in Figure 7 shows a minimum connection threshold of 0.17, a minimum correlation threshold of -1 and a maximum correlation threshold of $+1$. Setting the edge removal percentage to 30% (the lowest possible value to keep the graph connected) and the correlation threshold to 0.17, a total of 179 edges within a single connected component are obtained. A representation of the obtained BAF is also shown.

The next step concerns the visualisation of the BAF obtained after the transitive closure of attack and support relations. From this screen the user can see the resulting BAF, including the updated number of edges and connected components. In our example, the number of edges increases from 179 to 315. Then, an AF obtained from the BAF after the removal of the supports. On this AF, we compute the semi-stable extensions and, for each of them, its probability of being admissible.

In the last panel, the user can see the BAF before the closure phase, together with the list of semi-stable extensions and their probability computed through ConArg[3]. Each extension in the list is accompanied by a "Highlight" button which produces an explanation tree for the analysed extension.

VI. Conclusion and Future Work

We provide an argumentative interpretation of the answers provided by Machine Learning models, proposing AFs to obtain a dialectical explanation. To this end, we devised a procedure which allows the construction of a BAF and an explanation tree for each computed semi-stable extension. Such a tree represents the *dialectical reasoning* among the features of the analysed problem, and together with the produced BAF, explains why a certain class is assigned. The extensions are calculated through ConArg, a tool based on programming with constraints capable of solving various problems related to the AF, including the calculation of semi-stable semantics, which allowed both to always guarantee an answer, given the obligation of existence, and constraints that can provide an acceptable explanation. The list of extensions we obtain can be seen as a set of possible values that a record must take to be assigned a certain class. In particular, to specify the value taken within the extensions the single value that the analyzed topic can assume, the clustering phase of the dataset was necessary, while for the visualization of the correlation calculation between the analyzed topics a BAF was built within the proposed site. To then find a causal relationship between arguments, and not a simple correlation, it was necessary to break the complete symmetry of the BAF created in the previous step. Subsequently, in order to calculate the semi-stable extensions through ConArg, it was necessary to transitively close the arcs and then eliminate all the supports, obtaining an AF composed of only attachments equivalent to the original BAF. Finally, again through ConArg, it was possible to associate a probability to each calculated extension, to assign a level of *trust* to the answers provided. A web interface has been also developed: for the communication between the site and the backend, reference was made to Flask[4], while the Cytoscape.js

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

[4]Flask documentation available at https://flask.palletsprojects.com.

library[5] was used for the display and manipulation of the AF and explanation tree.

As future work, alternative techniques could be applied to break the symmetry of the graph to obtain 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 serve as an explanation for 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 [15] 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, and F. Herrera, "Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: https://doi.org/10.1016/j.inffus.2019.12.012

[2] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, and S. Villata, "Towards artificial argumentation," *AI Mag.*, vol. 38, no. 3, pp. 25–36, 2017. [Online]. Available: https://doi.org/10.1609/aimag.v38i3.2704

[3] P. Baroni, M. Caminada, and M. Giacomin, "An introduction to argumentation semantics," *Knowl. Eng. Rev.*, vol. 26, no. 4, pp. 365–410, 2011. [Online]. Available: https://doi.org/10.1017/S0269888911000166

[4] S. Bistarelli, A. Mancinelli, F. Santini, and C. Taticchi, "An argumentative explanation of machine learning outcomes," in *Computational Models of Argument - Proceedings of COMMA 2022, Cardiff, Wales, United Kingdom, 14th - 16th September 2022*, ser. Frontiers in Artificial Intelligence and Applications, F. Toni, S. Polberg, R. Booth, M. Caminada, and H. Kido, Eds., vol. 353. IOS Press, 2022, pp. 347–348.

[5] S. Bistarelli, T. Mantadelis, F. Santini, and C. Taticchi, "Using metaproblog and conarg to compute probabilistic argumentation frameworks," in *Proceedings of the 2nd Workshop on Advances In Argumentation In Artificial Intelligence*, ser. CEUR Workshop Proceedings, vol. 2296. CEUR-WS.org, 2018, pp. 6–10. [Online]. Available: http://ceur-ws.org/Vol-2296/AI3-2018_paper_2.pdf

[6] G. Boella, D. M. Gabbay, L. W. N. van der Torre, and S. Villata, "Support in abstract argumentation," in *Computational Models of Argument: Proceedings of COMMA*, ser. FAIA, vol. 216. IOS Press, 2010, pp. 111–122. [Online]. Available: https://doi.org/10.3233/978-1-60750-619-5-111

[7] M. Borovcnik, "Conditional probability – a review of mathematical, philosophical, and educational perspectives," in *Proceedings of ICME 12, Topic Study Group 11 "Teaching and Learning Probability"*, 2012.

[8] M. Caminada, "On the Issue of Reinstatement in Argumentation," in *Logics in Artificial Intelligence, 10th European Conference, JELIA*, ser. LNCS, vol. 4160. Springer, 2006, pp. 111–123.

[9] C. Cayrol and M. Lagasquie-Schiex, "On the acceptability of arguments in bipolar argumentation frameworks," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 8th European Conference, ECSQARU*, ser. LNCS, vol. 3571. Springer, 2005, pp. 378–389. [Online]. Available: https://doi.org/10.1007/11518655_33

[10] W. W. Cohen, "Fast effective rule induction," in *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 115–123. [Online]. Available: https://doi.org/10.1016/b978-1-55860-377-6.50023-2

[11] K. Cyras, A. Rago, E. Albini, P. Baroni, and F. Toni, "Argumentative XAI: A survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI*. ijcai.org, 2021, pp. 4392–4399. [Online]. Available: https://doi.org/10.24963/ijcai.2021/600

[12] Y. Dodge, *Spearman Rank Correlation Coefficient*. New York, NY: Springer New York, 2008, pp. 502–505. [Online]. Available: https://doi.org/10.1007/978-0-387-32833-1_379

[13] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–357, Sep. 1995.

[14] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1993, pp. 1022–1029. [Online]. Available: http://ijcai.org/Proceedings/93-2/Papers/022.pdf

[15] A. Jøsang, "Conditional reasoning with subjective logic," *J. Multiple Valued Log. Soft Comput.*, vol. 15, no. 1, pp. 5–38, 2009. [Online]. Available: http://www.oldcitypublishing.com/journals/mvlsc-home/mvlsc-issue-contents/mvlsc-volume-15-number-1-2009/mvlsc-15-1-p-5-38/

[16] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990. [Online]. Available: https://doi.org/10.1002/9780470316801

[17] W. B. Kendall, "A new algorithm for computing correlations," *IEEE Trans. Computers*, vol. 23, no. 1, pp. 88–90, 1974. [Online]. Available: https://doi.org/10.1109/T-C.1974.223783

[18] W. Kirch, Ed., *Pearson's Correlation Coefficient*. Dordrecht: Springer Netherlands, 2008, pp. 1090–1091. [Online]. Available: https://doi.org/10.1007/978-1-4020-5614-7_2569

[19] T. Kulesza, M. M. Burnett, W. Wong, and S. Stumpf, "Principles of explanatory debugging to personalize interactive machine learning," in *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 2015, pp. 126–137. [Online]. Available: https://doi.org/10.1145/2678025.2701399

[20] H. Li, N. Oren, and T. J. Norman, "Probabilistic argumentation frameworks," in *Theorie and Applications of Formal Argumentation - First International Workshop, TAFA 2011, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 7132. Springer, 2011, pp. 1–16. [Online]. Available: https://doi.org/10.1007/978-3-642-29184-5_1

[21] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 31:1–31:36, 2021. [Online]. Available: https://doi.org/10.1145/3436755

[22] H. Liu, A. E. Gegov, and F. T. Stahl, "Categorization and construction of rule based systems," in *Engineering Applications of Neural Networks - 15th International Conference, EANN*, ser. Communications in Computer and Information Science, vol. 459. Springer, 2014, pp. 183–194. [Online]. Available: https://doi.org/10.1007/978-3-319-11071-4_18

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986. [Online]. Available: https://doi.org/10.1023/A:1022643204877

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD*. ACM, 2016, pp. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[26] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020. [Online]. Available: http://aima.cs.berkeley.edu/

[27] S. T. Timmer, J. C. Meyer, H. Prakken, S. Renooij, and B. Verheij, "A two-phase method for extracting explanatory arguments from bayesian networks," *Int. J. Approx. Reason.*, vol. 80, pp. 475–494, 2017. [Online]. Available: https://doi.org/10.1016/j.ijar.2016.09.002

[28] W. J. von Eschenbach, "Transparency and the black box problem: Why we do not trust ai," *Philosophy & Technology*, vol. 34, no. 4, pp. 1607–1622, Dec 2021. [Online]. Available: https://doi.org/10.1007/s13347-021-00477-0

[5]Cytoscape.js documentation available at https://js.cytoscape.org.