Rule extraction for process mining based on machine learning techniques

Tomás Benavides-Álvarez[a,*]

[a]*Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain*

Abstract

Process mining is a discipline that has been gaining importance by offering a set of techniques that allow extracting knowledge from the event logs in which the information generated in the execution of processes is stored. One of the main objectives in process mining is to understand what has happened during the execution of a process. Typically, this goal is achieved by manually exploring the actual model, describing the behaviour of the process and temporal and frequency analytics on its variants and business indicators. In this paper, an innovative approach based on decision trees is presented that allows the automatic classification of certain behaviours that occur during a process based on the information generated during its executions and the variables associated with them, so that process stakeholders can have a better understanding of what is going on and thus improve decision making. This technique has been validated on a medical process, the Aortic Stenosis Integrated Care Process (AS ICP) implemented in the Cardiology Department of the University Hospital of Santiago de Compostela. On this process, the waiting times of patients have been tackled in order to extract those patient profiles susceptible to delays or to be prioritised.

Keywords: Process mining, Healthcare processes, Declarative processes, Decision trees

1. Introduction

The large amount of information generated individually by people in their daily lives is currently one of the main assets of many companies and organisations, which seek to extract knowledge from it in order to improve the results they obtain from their activity. Business processes consist of a collection of activities or tasks that, following a certain sequence, provide a product or a service to their customers or users [1]. These business processes are present in many aspects of our daily lives, one example being healthcare organisations. Typical processes in this sector, known as healthcare processes, consist of a series of activities focused on the detection, treatment or prevention of a disease, in which patient information is also collected to guide decision-making throughout the process.

Concern for improving this type of processes and making them more efficient has led to an approach to process mining, a very recent discipline that bridges the gap between data mining and traditional business process management [2]. To this end, it offers a range of techniques that allow knowledge to be extracted from the event logs in which the information generated by information systems during the execution of processes is stored [3]. With these techniques, the real behaviour of the processes can be discovered, using this information for the detection and correction of performance problems, i.e. bottlenecks, which would otherwise be practically impossible to address.

In the medical field, waiting times are a recurrent problem that has been tried to be solved by different means [4], as it is one of the main reasons for dissatisfaction and directly harms patients by postponing the benefits of the application of their treatments [5]. An example of this type of process could be the preparation and execution of a surgery, where, following traditional methodologies, a detailed manual analysis would be necessary to detect and optimise delays affecting patients, which is costly and time consuming. Moreover, this analysis may be conditioned by the heterogeneity of opinions among those involved in the process (physiscians, managers, etc.), who tend to consider one ideal scenario out of the many possible ones, which may lead to discussions among stakeholders with discordant points of view.

To solve these problems, an objective analysis by exploiting the data stored in the event logs is possible thanks to the techniques offered by process mining. This, in a medical process associated with a healthcare centre, can be a great opportunity to improve the management of patients, detecting those groups prone to delays or which patients tend to be prioritised.

However, this is no easy task, as working with healthcare processes is complex in a way that cannot be compared to other business process organisations, creating a number of challenges that need to be addressed to achieve satisfactory results [6]. The main reason behind this is the heterogeneity of patients, diseases and treatments, in addition to the multidisciplinary nature of healthcare centres and the ad hoc decision making that takes place at the time. All this generates dynamic and complex processes in which deviations from the theoretical model are very common, even among patients with the same disease and the same treatment, something that does not occur in other domains in which the same sequence of activities always takes

*Work supervised by Manuel Lama and Manuel Mucientes

Email addresses: tomas.benavides.alvarez@usc.es (Tomás Benavides-Álvarez), manuel.lama@usc.es (Manuel Lama), manuel.mucientes@usc.es (Manuel Mucientes)

place for each task. Another characteristic of the medical field is the transparency of decision making and the understandability of the models [7], which considerably limits the number of options and poses a challenge when designing and implementing a system based on artificial intelligence.

To address all these problems that have just been introduced, this study presents an approach focused on the extraction of knowledge through decision trees, which will allow the extraction of those rules present in the event logs that store the information generated during the execution of the processes. For this, a declarative approach will be followed to model the relationships between activities in a clear and understandable way, taking into account not only the workflow perspective of the process, i.e. the relationships between activities, but also the temporal and data perspective of the process. This system will be validated on a real medical process, specifically on the Aortic Stenosis Integrated Care Process (AS ICP) implemented in the Cardiology Department of the University Hospital of Santiago de Compostela. This process covers the path followed by patients from the detection of the disease until their intervention, discharge and subsequent follow-up. On this basis, an attempt will be made to extract the profiles of patients who suffer a delay in their interventions or who are correctly prioritised, which will be defined by the variables used by the decision trees when classifying them.

The following sections are structured as follows, Section 2 will introduce the current knowledge of the field to be addressed, with a special emphasis on declarative process modelling; Section 3 will focus on describing the two main proposals related to the approach adopted in this paper, illustrating their similarities and differences; Section 4 will present the methodology proposed to solve the problem, addressing its different phases; Section 5 will describe the real process on which this proposal has been validated, the AS ICP implemented in the Cardiology Department of the University Hospital of Santiago de Compostela; the experimentation carried out and its results will be shown in Section 6; finally, Section 7 will present the main conclusions reached during this process and some of the points for improvement that have been detected.

2. Background

2.1. Process mining

Process mining is a relatively young discipline that began to gain importance in the first years of the 2010s, when the most relevant proposals in the field started to appear [3]. As already mentioned, this discipline allows extracting the knowledge that is stored in the event logs during the execution of processes. These logs store the different instances of a given process, which are called cases. Each case has a unique identifier that differentiates it from the others, and is formed by the sequence of activities that define it, the timestamps of its execution and, optionally, variables related to it, such as the resource in charge of carrying them out or their result. Even trace variables can be found, which are those related to the case and not to a specific activity, such as the final state of the case.

To extract the knowledge stored in these event logs, process mining offers three main types of techniques illustrated in Figure 1: process discovery, process conformance and process enhancement [8]. Process discovery focuses on obtaining the model that describes the behaviour recorded during the execution of a process from its event log. In contrast, process conformance techniques aim to detect differences between the recorded behaviour and the theoretical model in order to detect possible deviations from the latter. Finally, process enhancement methods seek to improve the theoretical model with the information extracted from the logs, so that critical points that present performance problems, such as bottlenecks, can not only being detected, but also corrected.

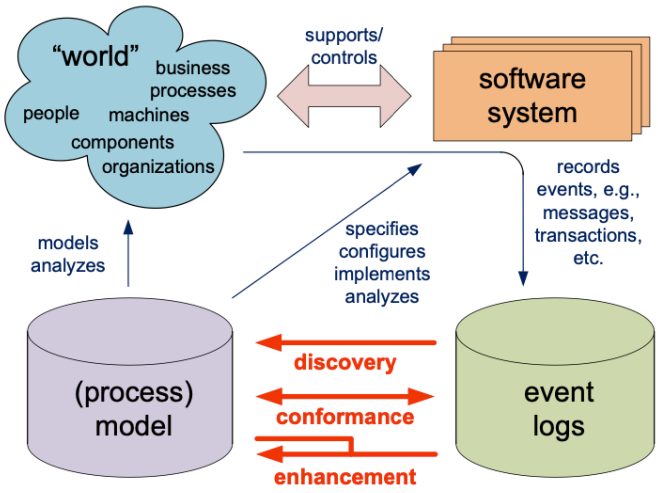

Figure 1: The three basic types of process mining: discovery, conformance and enhancement. [8]

In the healthcare domain, process mining has been used in different case studies, with promising results. The new techniques offered by this paradigm make it possible to deal with the great complexity and variability of medical processes, making easier to understand them and providing an efficient analysis, which improves the quality of the services offered and their management. Among the possibilities offered by the application of process mining in this type of processes are a better identification of the real behaviour of the process, the extraction of suggestions for improvement, the analysis of performance to reduce waiting times, the prediction of patients based on previous cases or the identification of the rules that manage the course of the process [9].

2.2. Declarative processes

Focusing on process discovery, the techniques that have received the most attention are focused on generating a model that represents the explicit relationships between the activities in the process, known as procedural or imperative modelling [10]. This type of modelling provides good results when working with structured processes, i.e. processes in which there is little variability and the same chain of events is normally followed. However, when dealing with unstructured processes uncertainty and high variability increase the number of relation-

ships that can be established between activities, giving rise to models such as the one shown in Figure 2, known as spaghetti processes. Unlike structured processes, where variability is low and the same chain of events is usually followed, in these it is not possible to explicitly represent each of the relationships between activities, as this leads to incomprehensible processes such as the one seen, so other approaches are advisable.

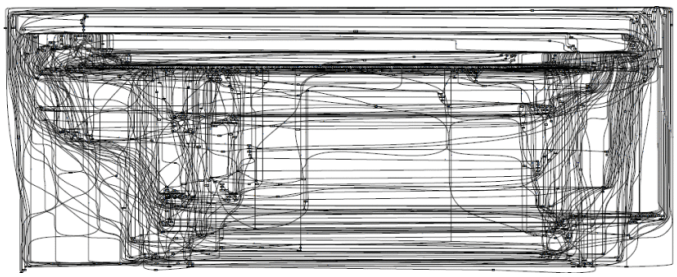

Figure 2: Spaghetti process describing the diagnosis and treatment of 2765 patients in a Dutch hospital. [11]

To solve this, the DECLARE language appeared in 2007 [12], giving rise to the declarative paradigm, in which instead of explicitly representing all the possible relationships between each pair of activities, a set of restrictions that must be satisfied throughout the execution of the process are described. The models obtained with this type of techniques therefore offer a greater degree of flexibility, describing more clearly the relationships established between the activities of a process in which uncertainty is high, as would be a case study focused on the medical domain [13]. These constraints, which can be seen as rules to be complied with during the execution of the process, limit or prohibit certain behaviours rather than representing the explicitly allowed one. The difference between the two approaches is ilustrated in the example shown in Figure 3, where there are two activities *A* and *B* that cannot occur simultaneously.

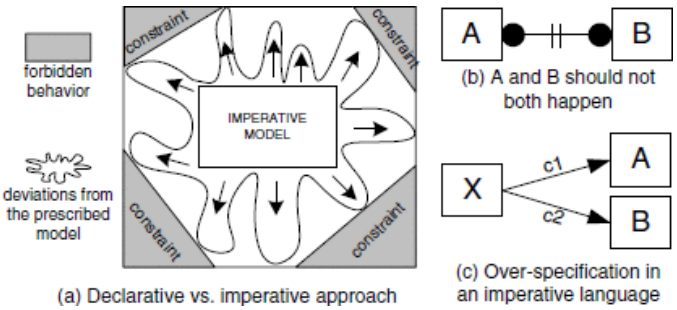

Figure 3: Declarative vs Imperative approach. [12]

This versatility offered by DECLARE is given by the Linear Temporal Logic (LTL) [14], which is the one that allows defining the constraints on which the language is based. This type of logic allows to represent formulae about the future of a path following a temporal approach, i.e. basing its conditions on the succession of events over time. Table 1 shows the constraints offered by DECLARE, along with their formulation in LTL logic, which restrict the behaviour of a process without explicitly representing a relationship between two activities.

At this point, it can be see that one of the problems detected in the unstructured processes typical of the medical domain has been solved. Despite this, the declarative models represented through DECLARE have a major limitation, which is that they only take into account the workflow of the process, that is, the succession of activities that take place in each of its executions, leaving aside all the remaining information that is stored in the event logs, such as the time or data perspective. In a problem such as the one at hand, in which a lot of clinical information about the patient, that is also relevant to decision-making and the course of events, is available, it is essential to take advantage of it to achieve good results. As a solution, MP-DECLARE [15] has emerged, an augmented version of DECLARE itself that offers a multiperspective view of the process, making it possible to obtain models in which all the information just mentioned is taken into account, as can be seen in Figure 4.

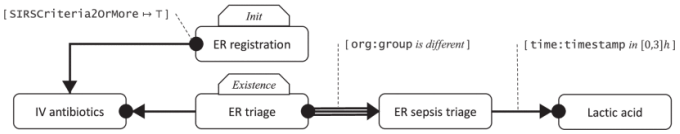

Figure 4: Example of a declarative multi-perspective model. [16]

With the emergence of this new paradigm, new tools such as RuM emerge [16], which allows, among other things, the discovery of declarative models from a multiperspective view. Even so, all these types of approaches present in the state of the art of the problem focus on discovering those restrictions whose support and confidence satisfy a certain threshold, giving a general vision of the behaviour of the process, as shown in Figure 4. On the other hand, in this case the aim is to use all this expressiveness offered by MP-DECLARE to describe the rules to be extracted, being this extraction guided by certain behaviours that occur in the process, such as delays. In this way, those rules that explain and describe them will be extracted so as to identify which correlations are present in those cases.

3. Related work

As mentioned above, the approach presented here is based on a knowledge extraction system that uses decision trees for rule extraction. These rules show the correlation between a set of attributes of the process and a class that will depend on the analysis to be performed. In this way, those rules that describe certain behaviours that occur in the process will be extracted to provide its stakeholders with information that is useful for them when it comes to its understanding and improvement.

In terms of process mining work focused on extracting correlations from event logs, two main approaches have been identified. The first is the one proposed by de Leoni et al. [17]. In it, a development framework is presented that proposes a general solution to analyse a process from multiple perspectives, such as control, data or temporal. This is very similar to the approach presented here, as it encompasses the multi-perspective view of

Constraint	LTL Expression	Description
Existence	$\Diamond a$	a must be executed at least once.
Init	a	a should be the first activity executed.
Last	$\Diamond(a \wedge \bigcirc \neg T)$	a should be the last activity executed.
Choice	$\Diamond a \vee \Diamond b$	a or b must be executed.
Exclusive Choice	$(\Diamond a \vee \Diamond b) \wedge \neg(\Diamond a \wedge \Diamond b)$	a or b must be executed, but never both.
Responded Existence	$\Diamond a \rightarrow \Diamond b$	Si a es ejecutada, b también debe ejecutarse.
CoExistence	$(\Diamond a \rightarrow \Diamond b) \wedge (\Diamond b \rightarrow \Diamond a)$	a and b are both executed, or neither is.
Response	$\Box(a \rightarrow \Diamond b)$	Each time a is executed, b must be executed subsequently.
Precedence	$\neg b \mathcal{W} a$	b can be executed only if a has been executed before.
Sucession	$\Box(a \rightarrow \Diamond b) \wedge (\neg b \mathcal{W} a)$	Combination of *Response* and *Precedence*.
Alternate Response	$\Box(a \rightarrow \bigcirc(\neg a \mathcal{U} b))$	Each execution of a must be followed by an execution of b, with no further execution of a in between.
Alternate Precedence	$(\neg b \mathcal{W} a) \wedge \Box(b \rightarrow \bigcirc(\neg b \mathcal{W} a))$	Each execution of b must be preceded by an execution of a, with no further execution of b in between.
Alternate Sucession	$\Box(a \rightarrow \bigcirc(\neg a \mathcal{U} b)) \wedge (\neg b \mathcal{W} a) \wedge \Box(b \rightarrow \bigcirc(\neg b \mathcal{W} a))$	Combination of *Alternate Response* and *Alternate Precedence*.
Chain Response	$\Box(a \rightarrow \bigcirc b)$	Each time a is executed, b must be executed immediately thereafter.
Chain Precedence	$\Box(\bigcirc b \rightarrow a)$	b can only be executed immediately following a.
Chain Sucession	$\Box(a \equiv \bigcirc b)$	Combination of *Chain Response* and *Chain Precedence*.
Not CoExistence	$\neg(\Diamond a \wedge \Diamond b)$	a and b are never both executed.
Not Sucession	$\Box(a \rightarrow \neg \Diamond b$	a must not be followed by b, and b must not be preceded by a.
Not Chain Sucession	$\Box(a \equiv \bigcirc \neg b)$	a and b must not be executed in succession.

Table 1: DECLARE templates for the constraints taken into account in the approach along with their expression in LTL logic.

the process discussed above. However, there is one major difference, which lies in the use of LTL logic and the constraints offered by DECLARE based on it.

If the pipeline of the proposal is observed, it is not very different from the one that will be shown in the following section, being key the manipulation and enrichment of the initial event log. This preprocessing phase is in charge of adding the features to be taken into account in the knowledge extraction, which will be generated from the information stored in the log, and of adapting its format for further processing. Finally, with regard to the input characteristics of this approach, it has already been said that the main difference in favour of the proposal presented in this article is the presence of declarative constraints, which are key when working with unstructured processes. On the other hand, variables related to process conformance are also offered by de Leoni et al., allowing the discrimination of cases on the basis of their compliance with the theoretical model of the process, i.e. their expected behaviour, something that is beyond the scope of our proposal.

The second approach is the one proposed by Leno et al. [18]. This proposal aims to address the discovery of declarative correlations by also including the perspective of the data. Therefore, it is similar to the solution of this article in the declarative approach when representing the behaviour of the process, although from a completely different focus. In this case, instead of extracting those rules that classify the cases of the process, the correlations between the attributes of the activities that satisfy a declarative constraint previously specified by the user are extracted. To do so, two different techniques are used, one based on clustering and the other based on trees, more specifically on redescription mining.

As can be seen, although there are proposals with similar characteristics, none of them allows the discovery of rules from a multiperspective and declarative point of view in a guided way. The most similar is the one presented by de Leoni et al. [17], although it offers an imperative approach in which the relationships between activities are explicitly represented, so it is not suitable for environments with high variability such as the medical domain. This is the gap that the approach presented below aims to fill, providing an innovative and appropriate solution for the problem being addressed.

4. Methodology

The solution proposed in this work is based on decision trees [19]. The use of this technique is mainly due to the ease it offers to extract intuitively the set of if-then rules that follow each other in its decision nodes, encouraging the explainability and interpretability of the results, which is of great interest in the medical field. In addition to this, a previous preprocessing of the data will be necessary to generate the information that will be used to train these trees. This will generate a large number of variables which, in problems with few instances, may negatively affect the training of a single tree. Therefore, to solve this problem, a feature selection process will be carried out using tree ensembles, specifically XGBoost [20], to improve the

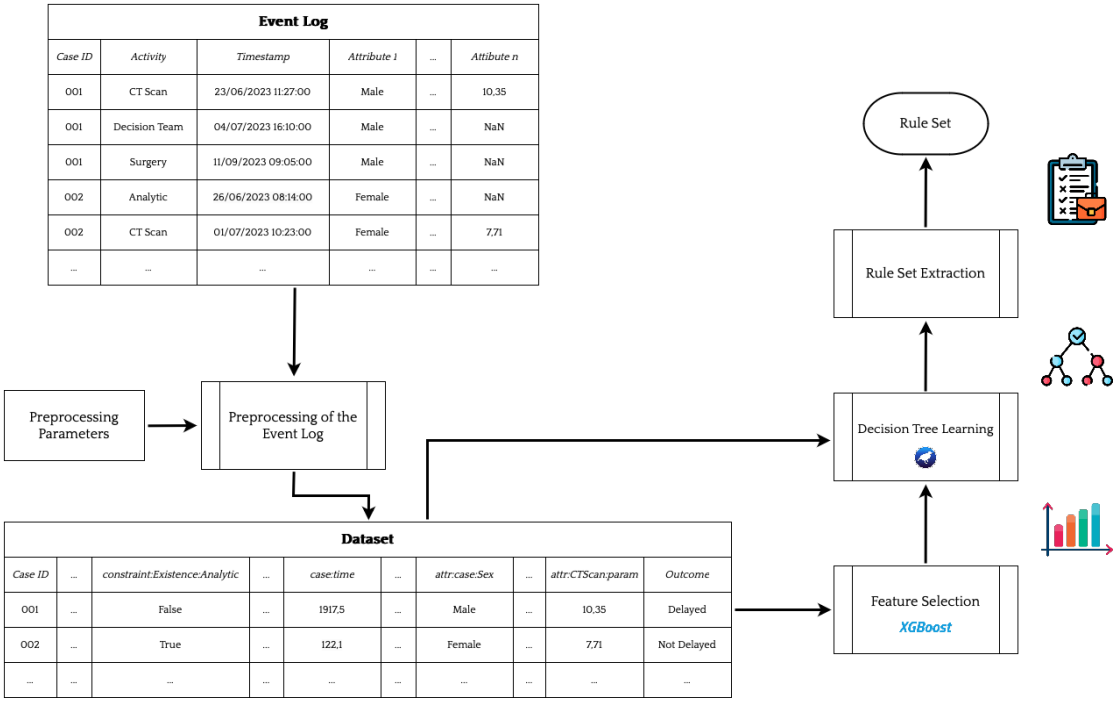

Figure 5: Pipeline of the proposal, in which the phases of preprocessing, feature selection, decision tree training and rule extraction are identified.

robustness of the variable selection that is already done by the decision trees. Figure 5 shows the general pipeline of the proposal, where all these stages can be identified, which will be detailed below.

4.1. Preprocessing of the event log

The preprocessing of the event log is one of the most important parts of making rule extraction possible. Unlike traditional datasets, where each row represents an instance of the problem, in event logs each row stores information about the specific execution of an activity belonging to a case. In this way, each case spans several rows, as can be seen in the original event log in Figure 5. In order to be able to process such datasets using traditional learning algorithms, the dimensionality of the event log must be reduced so that each execution of the process corresponds to a single row. During this process, not only must the dataset be resized, but the variables that provide the necessary information for the subsequent classification process must also be generated, including the class that will guide this process.

In the case of the process workflow, which will be the only one that will always be present after preprocessing, reducing the dimension of each case to a single row raises a problem with the distribution over time of the different activities that make up each execution of the process. Unlike other techniques such as recurrent neural networks [21], decision trees cannot deal with an arbitrarily large number of variables that represent the succession of activities that have taken place throughout each of the cases. Through the specification offered by DECLARE, these explicit sequences that would be impossible to handle can be converted into a set of bounded variables that describe the different behaviours that occur in the process. To do this, what is done is to process each trace or case with a set of symbolic automata, which are defined from the expressions in LTL logic of each of the constraints offered by the language, already shown in Table 1.

These automata, generated through the FLLOAT library [22], make it possible to detect the fulfilment of each of the constraints that have been specified for the sequence of activities that make up each instance, regardless of their length. In this way, instead of explicitly specifying each arc between activities, a series of boolean variables are generated that indicate, for each instance, whether or not a declarative restriction on one or two activities is fulfilled. An example could be that activity B is executed after the occurrence of activity A, whose automaton is illustrated in Figure 6. Additionally, variables related to the redundancy of the events will also be generated, which will indicate the number of times each of them has been executed, something that could be of importance for loop detection. In this case, as introduced in previous sections, the aim is to provide a multiperspective approach that looks beyond the workflow of the process, taking advantage of the temporal and data information stored in the event logs. To this end, during preprocessing, in addition to generating the above variables, also the following will be produced.

- *Time*. The duration of the activities or the time elapsed between each particular pair of activities is very useful information when discriminating one process execution from another. Here, both these two values and the time elapsed between the start or end of the execution and the completion of each task are considered. Regarding these time variables, it must be taken into account that it is possible that there is more than one value for each of them, since the

repetition of activities is not uncommon in processes. To solve this, aggregation of the times has been considered, usually by choosing the maximum or minimum value of the variables.

- *Resources*. Interactions between different members of an organisation can be of great relevance when analysing the performance of a process. Synergies when collaborating between two colleagues or the clash of two completely different methodologies can make a big difference in the final result. With the resource perspective, information is stored on which resource or resources perform which activity throughout the process, in order to detect performance problems related to the specific execution of an activity by one resource or poor coordination between two of them.

- *Attributes*. In event logs, as mentioned above, two types of attributes are distinguished: trace attributes and event attributes. The former apply to the whole case, while the latter are related to the execution of a specific activity and would have no meaning without it. Thus, variables related to both will be generated, containing for the second scenario also information about the event with which the attribute is associated. This perspective will be very useful, especially in the domain to which this approach is applied, since attributes are fundamental for decision making in the medical field.

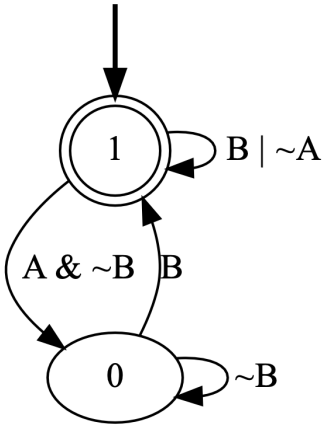

Figure 6: Symbolic automata generated by *FLLOAT* for the template *Response(A, B)*. This constraint implies that if activity *A* occurs, activity *B* must occur afterwards.

These would be the parameters considered by the proposal, which offer a multiperspective view covering all levels of the process for a detailed and customised analysis. While the workflow is always taken into account, the other aspects are optional and will depend on the problem to be addressed. On the other hand, given the high complexity of some processes, it should be noted that the number of variables can be very high and can exponentially increase, especially in terms of declarative constraints. This is due to the three types of interactions that can be seen between the activities involved.

- *Simples*. Constraints formed by a single activity, such as *Existence(A)*, i.e. activity *A* occurs in the process. In this case, only one activity is involved in the constraint, so we would have a maximum of *n* variations, where *n* is the total number of different activities in the process.

- *Combinations*. Constraints composed of two activities where the order does not matter, such as the *Choice(A, B)* constraint, i.e. activities *A* or *B* occur in the process. Here, the maximum number of variants of the same constraint would be limited by the formula

$$C_n^p = \binom{n}{k} = \frac{n!}{(n-p)!\,p!}$$

where *n* is equal to the number of different activities in the process and *p* is equal to 2.

- *Variations*. Constraints composed of two activities where order matters, such as the *Response(A, B)* constraint previously seen, which would have a different meaning from *Response(B, A)*. For this type of constraint, the maximum number of variants would be limited by

$$V_n^p = \frac{n!}{(n-p)!}$$

where again *n* is equal to the number of activities in the process and *p* is equal to 2.

As can be seen, as the number of unique activities of a process increases, so does its complexity, and with it the number of variables generated in the preprocessing, which can explode.

4.2. Selection of features

The preprocessing phase that has just been introduced entails the generation of a large number of variables that grows with the complexity of the process to be treated. This is added to the scarcity of data inherent to process mining, which studies very specific cases such as the one under discussion. Additionally, it should be taken into account that some of the declarative constraints considered, which constitute the majority of the variables in the problem, represent very similar behaviours. An example can be seen with the variables *Response(A, B)* and *ChainResponse(A, B)*, where the only difference is that the latter implies that activity *B* must occur immediately following activity *A*, if *A* occurs, rather than at any time thereafter. In some processes, the variables could be identical, since the behaviour expressed by the latter is contained in that of the former, and the same is true for other sets of constraints. Because of this, a variable selection process is needed to reduce the dimensionality of the problem so that it is addressable in the rule extraction process.

For this purpose, XGBoost [20], one of the most widely used and effective methods in the field of machine learning, has been chosen. This algorithm is based on ensemble methods, which rely on the combination of models to obtain results of higher quality and robustness than those achieved by a single model. Within ensembles, this is a boosting method, characterised by

the sequential combination of models so that, in each iteration, the new model focuses on the most complicated data, i.e. those where classification has failed.

For all these reasons, this algorithm has been used to implement an embedded feature selection method. This type of method is based on including the selection of variables in the learning process itself, for which XGBoost will be applied to the dataset obtained after preprocessing and, subsequently, the information on the importance of the attributes used will be extracted. In this learning process, some of the main features offered by the algorithm will be used, such as column and row subsampling, also used by random forests [23], which is another of the main methods used for the embedded selection of variables. Furthermore, small trees will be used, offering low variance and high bias, since it will be the sequential combination of the models in the ensemble that will reduce the latter.

Once the models have been trained, the information related to the feature importance of each of the attributes that have been used for their creation can be extracted. For this purpose, XGBoost offers several evaluation criteria. Of these options, it will be used the average information gain offered by the attribute in each of the splits in which it participates, i.e., how much the purity of a tree node improves when its instances are split by that attribute. This is the most balanced criterion and the one that best suits our purpose, favouring those attributes that most correlate with the target used to guide the classification. In contrast, other criteria, such as the number of times a feature is used to split the data across trees, may favour certain types of attributes over others, such as numerical attributes over binary ones, as this can only occur once per branch for the latter, while in the former can happen at different levels of the same branch. In this case, most attributes are boolean, i.e. they store a true or false value, so using such methods could give biased results that do not favour the final classification process.

With this criterion, the average gain values of each of the attributes used by the ensemble will be extracted, on which a normalisation process will be carried out to facilitate their comparison. Additionally, a filtering will be applied to the total number of attributes used by the trees based on the amount of information they represent, i.e., attributes that accumulate a user-specified percentage of the total information gain will be selected, sorted by importance. The final result will be a graph showing, in order of importance, the different attributes used during the learning of the algorithm along with the threshold that delimits which of these attributes will be selected for the training process of the final decision tree, from which the rules will be extracted.

This information on its own may already be of great relevance to the stakeholders of the problem. However, there is no way of knowing from this information how the extracted features affect the process, i.e. whether they contribute to the delay or favour the agility of the assistance. Moreover, the combination of trees in the ensemble hinders explainability, making the extraction of rules from them very complex to analyse. Therefore, an additional stage is necessary in which the variables extracted in this step are used to train a single tree, from which a set of rules will be extracted to show the correlations of the attributes with the final output of the process to be explained. In this way, the information extracted from this phase will be complemented with knowledge that will allow measures to be taken to improve the process or help to better understand what is happening in it.

4.3. Generation of decision trees

Decision trees are one of the best techniques for discovering key patterns when discriminating instances of a dataset based on a class, i.e. in supervised learning problems. Moreover, their simplicity and intuitiveness when it comes to interpreting their reasoning make them one of the main techniques in the exploratory analysis of data, as will be seen later in the rule extraction phase. In this case, regarding the input features of the model, those selected in the previous feature selection process will be used. These variables have been generated in the preprocessing phase, by which the initial event log is converted into a dataset with the input features of the problem to be analysed, taking into account the different perspectives of the process. Therefore, this will be the final phase that allows the extraction of knowledge from the event log being worked on. To this end, this process will be guided by the assignment of classes that has been carried out previously, so that the final result will be the rules that show which combinations of variables correlate with each one of them.

To implement this solution, it has been chosen the implementation of the CART algorithm [24] offered by the Weka tool [25], one of the main open source machine learning and data mining software developed by the University of Waikato. This algorithm is characterised by offering a binary decision tree in which each decision node is divided into two different branches based on a condition, following a greedy approach known as recursive binary splitting. To do this, at each of these nodes, a certain cost function is evaluated on the different values of each one of the attributes that make up the dataset, selecting the splitting that offers the lowest cost.

As cost function, this algorithm uses the Gini index or Gini impurity. This criterion measures how often a randomly chosen element of a set would be incorrectly labelled if it were labelled according to the distribution of labels in the set, i.e. it gives an insight into how pure a node is based on the instances it contains. The following equation is used for its calculation

$$GI = \sum_{i=0}^{c} P_i(1 - P_i) = 1 - \sum_{i=0}^{c} P_i^2$$

where P_i refers to the proportion of instances of class i in the node under consideration, representing the probability that an instance of that class is randomly selected in the node. With this in mind, the best scenario in a binary case would be the one in which a node only contains instances of one class, $GI = 0$, while the worst scenario would be that in which the classes are equally distributed, yielding a $GI = 0.5$.

Taking this into account, to evaluate a split in a decision node, what is done is to weight the Gini index of each of the nodes resulting from the split by the number of instances of the parent

node, which would give its Gini score

$$GS = GI_{left} * \frac{n_{left}}{n} + GI_{right} * \frac{n_{right}}{n}$$

where GI_{left} and GI_{right} correspond to the Gini index of the child nodes generated by the split and n_{left} and n_{right} to the number of instances of each of them. Thus, all the cut-off points for each of the attributes under consideration are tested in order to select the one with the lowest GS. This process is repeated iteratively until one of the stopping conditions specified in the form of hyperparameters to the algorithm is met, such as the maximum depth of the tree, the minimum number of examples that a leaf node must contain or the minimum number of examples that a decision node must have to be able to split again.

In addition, there are other relevant characteristics for the operation of the algorithm that have been taken into account when selecting it, choosing those that best adapt to the nature of the data with which this kind of problems work.

4.3.1. Missing values

Missing values are one of the main problems faced by machine learning algorithms. In this case, as far as declarative variables are concerned, they do not pose any problem, since there is always a true or false value depending on whether or not each constraint is met. On the other hand, when taking into account the variables related to the time and data perspective of the process, it may be the case that null values are present, especially in the first case. This is because, when generating the variables related to the temporal aspect of the process, it may be possible that an activity does not occur in a specific execution, which would make the variable in charge of storing its duration empty, as well as any other attribute in which this activity is involved, such as the time elapsed between it and another given activity.

Because of this, for some variables in the dataset the presence of missing values can be a problem. To solve it, some method of imputation by the mean, or even replacing these values by zero, could be used. But in this particular problem, this would only introduce bias, since the fact that an activity has not been performed should not be confused with the fact that this activity has been completed in a given duration, be it zero or any other value considered.

Another approach that can be adopted in this case, taking advantage of the characteristics of decision trees, is that of fractional instances, proposed by Quinland in another of the great decision tree algorithms, C4.5 [26]. This is the technique used by Weka in its implementation of CART, and is based on assigning a probability to each of the possible values of the attribute under consideration. For this, all the instances that present a valid value for the attribute on which the split is being performed are taken into account. From them, the probability that the attribute takes each of these values is extracted, and the instances with missing values are partitioned on the basis of these probabilities.

Consider an example for a given node to be partitioned over the numerical attribute T using a threshold x. In this node, we have ten instances with valid values, of which six are less than x and four are greater, i.e., the probability that $T \leq x$ is 0.6, while the probability that $T > x$ is 0.4. Taking this into account, if in this node there is an instance that has a null value for the attribute T, what will be done is to distribute a fractional instance with a weight of 0.6 for the corresponding branch with $T \leq x$, and a fractional instance with a weight of 0.4 for the other branch. In this way, it will be these fractional instances that will be used when calculating the Gini index of each node and that will be taken into account in the following partitions.

This method allows the missing values problem to be solved in an elegant manner, taking advantage of the characteristics of the decision trees, both for the learning and the prediction processes, although in this case it is the former that is of interest. Moreover, it is considered to be the most appropriate strategy considering the nature of the missing values used in this type of problem, so it was one of the main points in favour when selecting the algorithm.

4.3.2. Categorical values

Another key issue when working with decision trees are categorical or nominal variables, which are related to qualitative characteristics of the data, offering a limited number of different values. The most common approach is to perform their discretisation using one hot encoding. With this, each variable is split into n different variables, where n is the number of different values the original variable has. In this way, one of these new attributes will take the value 1 and the rest will take the value 0, indicating the value of that instance for the original variable.

This is one of the possible approaches in decision trees. In fact, for a binary tree, it is not necessary to apply one hot encoding, as this discretisation will be done naturally by the algorithm when selecting the corresponding threshold for the variable when splitting. However, splitting the data based only on a specific value of a categorical attribute may not be optimal, since in the case that this categorical attribute has several different values that share the same correlation with the class that guides the learning process, several levels in the tree will be needed to represent it, which may even make it difficult to reach this conclusion.

However, if instead of this approach, the possible subsets for this type of variables are considered, the sparsity of the trees obtained could be reduced and the quality of the models could be improved. Therefore, it has been decided to use this strategy, offered by the implementation under consideration, as it can be beneficial for the models and rules obtained when such variables are present.

4.4. Rule extraction

Once the characteristics of the decision trees that will be used following the presented methodology have been introduced, it remains to explain how the extraction of rules from them will be performed. These decision trees will be formed by decision nodes and leave nodes. The former contain the variables used to discriminate the data based on the classification that guides the learning process, while the latter are the terminal nodes of each branch that represent the class assigned to the instances that

reach them. In this way, from the root node to each leaf node of the tree there is a path formed by a conjunction of decision nodes, which indicate the conditions that must occur in order to reach that terminal node. Each of these paths can be understood as a rule, so that a tree is nothing more than a set of decision rules, which have certain support and confidence values based on the instances they cover.

- *Support*. It refers to the coverage of the rule, i.e. the percentage of instances to which the condition of a rule applies.
- *Confidence*. It refers to the accuracy of the rule, i.e. how accurate the rule is in predicting the correct class for the instances to which the condition of the rule applies.

Thanks to the use of decision trees, this process just introduced is extremely intuitive and trivial, as can be seen in the example shown in Figure 7. However, in order to make it even more systematic, support and confidence thresholds will be established, so that only those rules that satisfy these restrictions on the class of interest for the stakeholders of the process will be extracted. That is, only those rules will be extracted whose antecedent affects at least a pre-specified minimum number of cases, i.e. rules with a higher support than a given threshold, and whose consequent is also true for at least a given percentage of these cases, i.e. rules with a higher confidence than a give threshold. With this, the rule extraction process, which is a manual process, will become systematic. Finally, following these steps, a set of rules about the class or classes of interest will be obtained whose antecedents will be given by the variables that form the decision nodes, and whose consequent will be related to the class of the leaf node to which its succession leads.

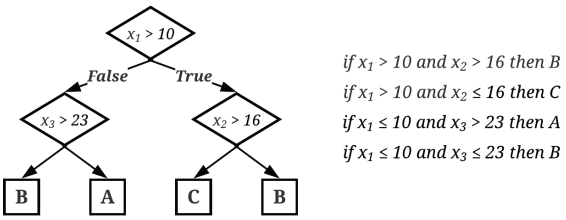

Figure 7: Example of the rule extraction process from a decision tree.

5. The Aortic Stenosis Integrated Care Process

This proposal works on a medical process, as mentioned above, more specifically on the Aortic Stenosis Integrated Care Process (AS ICP) implemented in the Cardiology Department of the University Hospital of Santiago de Compostela.

Aortic stenosis (AS) is a chronic progressive disease that affects the aortic valve, limiting blood flow from the left ventricle when this valve does not open properly. In fact, severe AS is the most prevalent valve disease in elderly patients [27], which is expected to worsen in the coming decades as the population ages. Mortality in this disease is strongly related to the development of symptoms, with a poor prognosis in the absence of intervention. However, the negative point when an intervention is indicated is that it is not performed immediately, but patients may suffer delays of even years [28]. During this waiting time, complications can occur that may hinder the prognosis, which is why the correct evaluation of the patient and the analysis of his or her conditions is key.

The paradigm of care for patients with AS is a process involving different professionals who provide fragmented care at different times and places, with little continuity of care and a high risk of lack of coordination that can hinder patient management. In order to solve this, Integrated Care Processes (ICPs) have emerged, which seek to ensure the effectiveness of clinical actions through greater coordination and a guarantee of continuity in care [29]. In January 2018, the ICP for severe symptomatic AS was implemented in the Cardiology Department of the University Hospital of Santiago de Compostela, with the aim of managing the care of patients requiring intervention and guaranteeing the correct and coordinated functioning of each and every one of its stages [30].

5.1. Definition of the process

The AS ICP covers all diagnostic and therapeutic events, from the indication for intervention at the heart team meeting (formal inclusion in the process), to the return of the patient to normal activity after completion of the post-intervention rehabilitation (exit of the patient from the process). The following stages can be distinguished:

- *Stage 0*. Stage prior to formal inclusion of the patient in the AS IPC, where the necessary tests will be requested and evaluated for presentation at the heart team meeting. The aim of this phase is to identify patients with severe AS who require intervention and to accelerate appropriate complementary tests to avoid delays and loss to follow-up.
- *Stage 1*. From the decision to intervene at the heart team meeting (formal entry into the process) to the valve replacement procedure, including prehabilitation of the patient. The aim of this phase is to optimise the situation of the patient prior to the intervention, to speed up pending tests and the resolution of conditions, and to identify possible alerts or decompensation early on, so as to avoid unexpected events on the waiting list or loss of follow-up.
- *Stage 2*. Valve replacement surgery, surgical or percutaneous. The aim of this stage is to guarantee a protocolised and homogeneous management of patients during their hospitalisation for the intervention.
- *Stage 3*. Rehabilitation and follow-up after valve surgery. This facilitates the recovery of the patient after the operation and the return to normal life.

To monitor the results of the AS ICP, a patient log is established in which the clinical information relevant to the process

is stored. This information includes the performance of diagnostic tests or patient monitoring and management events, which make up the activities of the process. In addition, information is also available on patient variables, such as variables describing the baseline characteristics of the sample, variables related to the patient management decisions, those referring to the previous preparation and rehabilitation after the intervention, adverse events and alerts prior to the intervention, variables referring to the final management of the patient and the intervention and, finally, those focusing on the follow-up after the exit from the process.

All of this makes up the process event log, which contains information for all patients of legal age with AS who have presented at a heart team session between 2018 and 2021 to decide on the management of their valvular heart disease. On them, there are a series of clinical questions or analyses that the medical team of stakeholders of the process wishes to address. Here, the application of process mining can provide a great advantage by facilitating the management of the large variability present in the process, assisting the healthcare professionals involved in the AS ICP.

5.2. Analysis of the problem

As mentioned in Section 1, one of the most recurrent problems in the medical field is that of delays. This problem is very difficult to deal with in a care process such as the one considered given the heterogeneity of patients and the multidisciplinary nature of these processes, so process mining can provide a new approach to improve current solutions. In this case, the analysis to be performed by the AS ICP managers is that of delays to intervention, i.e. the time that elapses between the last heart team meeting and the intervention. In this way, taking advantage of the information stored on the process, the aim is to extract those patient profiles that are more likely to suffer a longer delay than usual until their intervention, or to detect those profiles that are being prioritised by those involved in the process, in order to check whether priority is being given to the groups that require it.

The key process indicator (KPI) to be taken into account for this will be compliance with the stipulated waiting times between the heart team meeting and the intervention of the patient (Stages 1 and 2). When performing this analysis, the different pathways that patients may follow depending on their origin and the type of intervention they undergo must be taken into account. These two factors will determine the type of tests performed or the priority they will receive during their monitoring, which will be of great relevance in determining the standard waiting time for each of them. Therefore, a total of six groups will be defined, which will present different workflows and, therefore, different needs.

- *Outpatients undergoing surgery*. This is the case for patients whose disease is diagnosed during an outpatient consultation and whose indicated intervention is traditional (surgical) aortic valve surgery. In this case the expected time until surgery is 90 days.
- *Outpatients undergoing TAVI*. This is the case for patients whose disease is diagnosed during an outpatient consultation and whose indicated intervention is Transcatheter Aortic Valve Implantation (TAVI). For this type of patient, the stipulated time between the heart team meeting and the intervention is 30 days.
- *Inpatients undergoing surgery during admission*. These are patients who are diagnosed with AS during a symptomatic admission and are intervened during that admission following the surgical approach. For them, the ideal time is 15 days.
- *Inpatients undergoing TAVI during admission*. These are patients who are diagnosed with AS during a symptomatic admission and are intervened during that admission using the percutaneous approach. Again, the stipulated time for this group of patients is 15 days.
- *Inpatients deferred undergoing surgery*. Patients who are diagnosed with AS during an admission for symptoms and are discharged, with a subsequent admission for intervention, in this case surgery. For these patients, 30 days are stipulated from the meeting of the cardiac surgical team and their intervention.
- *Inpatients deferred undergoing TAVI*. Patients who are diagnosed with AS during an admission for symptoms and are discharged, with a subsequent admission for intervention, in this case TAVI. As with the previous group, the expected time for this type of patient is 30 days.

Following the estimated waiting times for each of these groups, which have been provided by the experts in the process, patients have been classified into two different classes according to whether or not they present a longer delay than expected. Table 2 shows the distribution of these classes, where the imbalance of two of the groups, outpatients undergoing TAVI and inpatients undergoing TAVI during admission, is striking. In these, the proportion of delayed cases, which are of most interest, is 13% and 8%, respectively. Since the patients indicated for TAVI are the least delayed and do not represent a significant number of problematic cases, it was decided to discard these two groups and to work henceforth with the remaining four in the rule extraction phase.

Group	Delayed	Not delayed	Total
Outpatients undergoing surgery	89	90	**179**
Outpatients undergoing TAVI	23	150	**173**
Inpatients undergoing surgery during admission	29	58	**87**
Inpatients undergoing TAVI during admission	8	93	**101**
Inpatients deferred undergoing surgery	30	21	**51**
Inpatients deferred undergoing TAVI	24	38	**62**

Table 2: Distribution of waiting time delays for the patients of the defined groups.

6. Experiments and results

The experiments carried out on the data corresponding to the AS ICP, for which the methodology presented in Chapter 4 will be applied, are presented below. The Notebooks and the source code on which the experimentation has been carried out can be found in the following repository.

6.1. Data preparation

The first and most important step is the preparation of the data. Here, a data cleaning process has been carried out to remove those cases that do not fit into this analysis. The problem to be addressed is focused on the waiting times from the heart team meeting to the intervention of the patient, so all those patients who have not yet been intervened will be discarded. This results in 653 patients, which have been divided into the corresponding groups based on their origin and type of intervention, as shown in Table 2. Once this has been done, the part of the process to be analysed will be extracted, in this case from its start to intervention, since the activities that happen afterwards should have no influence on the waiting times.

On this data, the preprocessing explained in Section 4.1 will be applied. In this case, the workflow and data perspectives will be taken into account, not the time dimension. This is because the classification is going to be performed on a variable derived from the time between two activities, as the type delay or no delay will depend on whether the time between the last heart team meeting and the intervention of the patient is higher than the threshold specified for each group previously, in Section 5.2. Therefore, the addition of variables on the time perspective of the process causes numerous undesired dependencies and correlations that return very good results without providing real information on the process that is useful for decision-making, overshadowing the other variables.

Finally, in terms of data perspective, two different approaches were followed. The first consisted of using domain knowledge to select the most relevant patient variables for the analysis of waiting times. For this purpose, experts in the process were consulted, obtaining a list of the variables that could most influence waiting times or that were of most interest for this analysis, made up of a total of 18 variables. In the second approach, the 123 patient variables present in the initial event log were used. The aim of this is to try to extract information that physicians miss or that, based on current domain knowledge, is unexpected, so that new relationships between variables can be discovered and new knowledge can be generated. In Table 3 we can see the number of variables resulting from applying the preprocessing and variable generation stage from these data for the first approach, which will be the one focused on from here on, as the latter is still pending validation.

6.2. Feature selection

When observing the Tables 2 and 3, one can see a big problem with the dimensionality of the data, with at least twice as many attributes as instances, reaching up to seven times more in some cases, such as the inpatients deferred undergoing surgery. This can make it difficult for the learning algorithms to work properly. For this reason, a feature selection phase prior to the application of decision trees has been considered necessary. For this, the methodology introduced in Section 4.2 will be followed, in which XGBoost will be used to extract the feature importance of the attributes used by the ensemble models. In this case, 20 iterations of the algorithm will be used, in which trees will be generated with a maximum depth of three by applying a subsample of the 80% in both rows and columns.

From the attributes used to generate the decision trees of the model, their feature importance will be extracted based on the average information gain and those that represent 95% of the total information gain will be selected. The result of this feature selection process is a graph as shown in Figure 8 for each group. In this case, it is the inpatients deferred undergoing surgery. The graph shows all the attributes that have participated in the generation of the ensemble models together with the relative value of information gain they have contributed to these models. In addition, a vertical line representing the cutoff appears, indicating which of all the variables have been selected taking into account the specified threshold.

From this outcome, very useful information can be extracted with regard to the variables that most influence waiting times. It can be seen, for example, that the fact that a concomitant mitral intervention is necessary is what most affects these times, providing a gain in information of 18.4%. The fact that an echocardiogram is performed before the heart team meeting, or that the anaesthetic assessment is performed after the patient has been placed on the waiting list, are also relevant. Even the number of hospitalisations of the patient, associated with the number of unscheduled admissions, directly affects their priority in the process. Moreover, Table 3 also shows the results of applying this process for each of the groups considered, which can be compared with the situation prior to its application.

6.3. Rule extraction

The information obtained in the previous step is very useful for the analysis of the problem, but it does not allow to know how each of the selected attributes affects the waiting times, i.e. if they increase or decrease the delay. For this purpose, guided rule extraction will now be performed by creating a decision tree using only the attributes that have just been selected. As

Group	Generated features	Selected features
Outpatients undergoing surgery	**439** (423 + 16)	**28** (16 + 12)
Outpatients undergoing TAVI	**355** (341 + 14)	**20** (11 + 9)
Inpatients undergoing surgery during admission	**334** (317 + 17)	**23** (15 + 8)
Inpatients undergoing TAVI during admission	**208** (193 + 15)	**14** (9 + 5)
Inpatients deferred undergoing surgery	**380** (363 + 17)	**17** (8 + 9)
Inpatients deferred undergoing TAVI	**174** (160 + 14)	**14** (4 + 9)

Table 3: Features generated in the preprocessing of the original event log (workflow + data) and features obtained after the feature selection stage for the domain based approach, where up to 18 patient variables are considered for analysis.

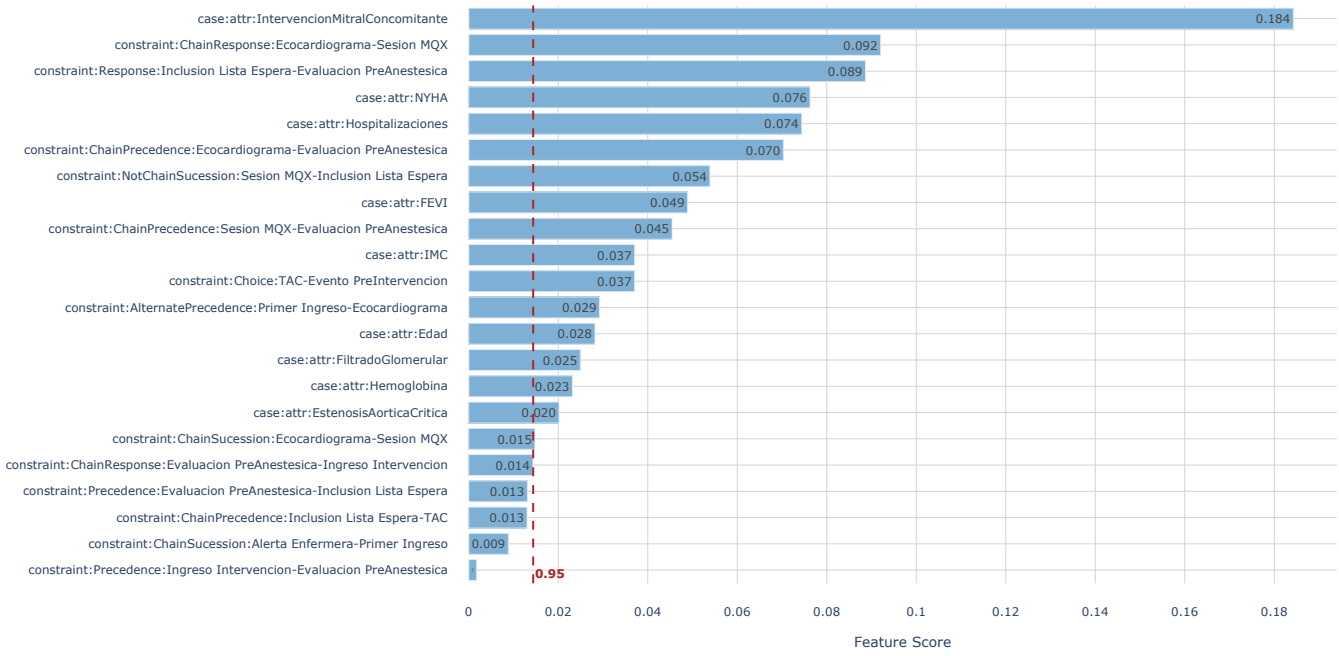

Figure 8: Graph obtained from the feature selection process using XGBoost attribute importance for inpatients deferred undergoing surgery.

indicated in Section 4.3, CART will be used for this, obtaining trees that will be limited to a minimum number of instances of four at the leaf nodes in order to maintain acceptable support for the rules obtained.

	Class	Precision	Recall	F-Score
Group A	*Delayed*	0.85	0.89	0.87
	Not Delayed	0.88	0.84	0.86
	Weighted Avg.	0.87	0.87	0.87
Group B	*Delayed*	0.73	0.93	0.82
	Not Delayed	0.96	0.83	0.89
	Weighted Avg.	0.88	0.86	0.87
Group C	*Delayed*	0.81	1.0	0.90
	Not Delayed	1.0	0.67	0.80
	Weighted Avg.	0.89	0.86	0.86
Group D	*Delayed*	0.72	0.75	0.74
	Not Delayed	0.84	0.82	0.83
	Weighted Avg.	0.79	0.79	0.79

Table 4: Performance metrics obtained from the decision trees for outpatients undergoing surgery (A), inpatients undergoing surgery during admission (B), inpatients deferred undergoing surgery (C) and inpatients deferred undergoing TAVI (D), respectively.

Table 4 shows the performance metrics obtained for the four groups considered in the rule extraction process (remember that in Section 5.2, two of the initial groups were discarded for this process), both for each class separately and for the complete problem. From these results, the implemented solution can be positively validated from a quantitative point of view. As can be seen, most of the delays are identified with good levels of precision, ranging from 70 – 85% for this class. In addition, very good results are also achieved for patients with no delay. All this translates into F-Score values exceeding 85% in most cases, which in a real medical problem can be considered as a very positive result.

From the resulting trees, the rules have been extracted as indicated in Section 4.4, establishing support and confidence thresholds of 4% and 70% respectively. Table 5 shows some of the most relevant rules extracted for some of the groups analysed, explained below.

- *Rule 1*. This rule indicates that most outpatients undergoing surgery who require an intervention on the aorta suffer a longer delay than usual.
- *Rule 2*. When outpatients undergoing surgery do not require this additional intervention and follow a normal flow of activities, with reasonable haemoglobin values, they suffer a delay as long as a nurse alert is given during the process or a CT scan is required, as these are activities that are not expected in patients undergoing surgery.
- *Rule 3*. Those outpatients undergoing surgery who follow a normal process and suffer syncope are treated with the appropiate priority, i.e. without delay, in most cases.
- *Rule 4*. This rule represents the typical non-risk patient profile for outpatients undergoing surgery, whose intervention is often postponed to give priority to other patients, casusing many to be delayed longer than desired.
- *Rule 5* and *Rule 6*. This pair of rules represents inpatients undergoing surgery during admission who follow the usual

flow of activities, i.e. they are admitted for the procedure, then the heart team meeting takes place and, once they are put on the waiting list, they undergo a preanaesthesia assessment immediately afterwards. The difference between the two rules is whether the AS is critical or not. This makes it clear that patients are prioritised properly, with the vast majority of those with a critical AS undergoing surgery without delay, while many of those who are not at risk take longer than the stipulated 15 days to receive their surgery.

- *Rule 7* and *Rule 8*. These rules for inpatients deferred undergoing surgery that suffer from delay do not include any type of declarative restriction, but focus only on those patient variables that describe the delays. This information is useful for patient monitoring, providing knowledge to detect future complications during the process if this combination of values is detected, for instance.

- *Rule 9* and *Rule 10*. From them, it can be extracted that in inpatients deferred undergoing TAVI, priority is being given to elderly women, who have waiting times within the optimal limits for the process. However, a problem can be detected for the male sex, where many delays are occurring. Unlike the other rules, which can be explained by applying medical arguments about the domain of the problem, this last one has been unexpected, leading to a more intensive analysis to discover the real reason behind this situation.

6.4. Validation by experts

The entire process shown above has undergone qualitative validation by the physicians in charge of managing and operating the AS ICP. In this process, the extracted rules have been discussed to check both their validity and their potential for application in extracting knowledge about the process. As a result, a positive validation has been received which has allowed the stakeholders of the process to understand some of the patient profiles based on their waiting times. This has given rise to a future comparison between the results obtained using this novel methodology and the traditional techniques used in the medical field to analyse care processes, with the aim of improving the current state of the field in this aspect.

	Rule	Antecedent	Consecuent	Support	Confidence
Group A	*1*	Intervención sobre aorta → **True**	Delayed	0.09	0.88
	2	Intervención sobre aorta → **False** Precedence: Evaluación Preanestésica - Ingreso Intervención → **True** Response: Sesión MQX - Evaluación Preanestésica → **True** Sincope → **False** Hemoglobina ≥ **13.95** Choice: Alerta Enfermera - TAC → **True**	Delayed	0.07	1.0
	3	Intervención sobre aorta → **False** Precedence: Evaluación Preanestésica - Ingreso Intervención → **True** Response: Sesión MQX - Evaluación Preanestésica → **True** Sincope → **True**	Not Delayed	0.04	0.86
	4	Intervención sobre aorta → **False** Precedence: Evaluación Preanestésica - Ingreso Intervención → **True** Response: Sesión MQX - Evaluación Preanestésica → **True** Sincope → **False** Hemoglobina ≥ **13.95** Choice: Alerta Enfermera - TAC → **False** IMC ≥ **25.23** Filtrado Glomerular ≥ **85.735**	Delayed	0.13	0.78
Group B	*5*	Response: Ingreso Intervención - Sesión MQX → **True** Not Chain Sucession: Inclusión Lista Espera - Evaluación Preanestésica → **False** Estenosis Aórtica Crítica → **False**	Delayed	0.13	0.73
	6	Response: Ingreso Intervención - Sesión MQX → **True** Not Chain Sucession: Inclusión Lista Espera - Evaluación Preanestésica → **False** Estenosis Aórtica Crítica → **True**	Not Delayed	0.07	0.83
Group C	*7*	FEVI ≥ **42.5** IMC ≥ **30.15** Edad ≥ **67**	Delayed	0.29	1.0
	8	FEVI ≥ **42.5** IMC < **24.93**	Delayed	0.12	1.0
Group D	*9*	Chain Response: Inclusión Lista de Espera - Ingreso Intervencion → **False** Intervencion Coronaria Percutanea → **False** Sexo → **Hombre**	Delayed	0.21	0.85
	10	Chain Response: Inclusión Lista de Espera - Ingreso Intervencion → **False** Intervencion Coronaria Percutanea → **False** Sexo → **Mujer** Edad ≥ **83.5**	Not Delayed	0.11	0.86

Table 5: Some of the rules extracted for outpatients undergoing surgery (A), inpatients undergoing surgery during admission (B), inpatients deferred undergoing surgery (C) and inpatients deferred undergoing TAVI (D), respectively.

7. Conclusions and future work

In this paper, a novel approach has been presented that addresses the extraction of knowledge, in a guided way, from the event logs in which the information generated during the execution of the processes is stored. This proposal covers all the necessary stages to achieve the desired results from a set of raw data. From the initial preprocessing of the event log, by which all the variables necessary for a correct analysis are generated, to the selection of features and the extraction of rules, which provide useful information on the performance of the process. All this from a multiperspective point of view that offers a complete vision of the process.

The result of this is a system that makes it possible to extract the knowledge that explains certain behaviours or outcomes that occur within the processes, which provides those in charge with a better understanding that favours decision-making with a view to the application of improvements. Furthermore, the proposal has been validated on a real use case such as the Aortic Stenosis Integrated Care Process of the University Hospital of Santiago de Compostela. The methodology presented has been applied to this process to extract patient profiles based on their waiting times. To this end, close collaboration with a group of cardiologists specialised in the process has been stablished to orient the analysis to their needs. Finally, the experimental results have been offered for discussion, which have allowed to identify unknown trends in the management of patients or to confirm behaviours that were already expected from the process. Therefore, a positive evaluation has been achieved that will have a real impact on the improvement of the process.

As future work, ways of improving the current proposal have been identified that would facilitate its implementation at a more general level. These would mainly focus on the presentation of rules in an easy-to-understand natural language, with a special focus on the description of declarative constraints, which can be complex especially when more than one restriction is combined in the same rule. This would reduce the need for a person specialised in the tool while facilitating the understanding of the extracted knowledge by those interested in the process, who do not necessarily need to be familiar with the terms used to describe it.

References

[1] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, et al., Fundamentals of business process management, Vol. 2, Springer, 2018.

[2] W. M. P. Van Der Aalst, Process mining, 2016.

[3] W. Van Der Aalst, A. Adriansyah, A. K. A. De Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. Van Den Brand, R. Brandtjen, J. Buijs, et al., Process mining manifesto, in: Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I 9, Springer, 2012, pp. 169–194.

[4] OECD, Waiting Time Policies in the Health Sector, 2013.

[5] R. S. Mans, W. M. P. Van Der Aalst, R. J. B. Vanwersch, Process mining in healthcare, 2015.

[6] P. Homayounfar, Process mining challenges in hospital information systems, in: 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), 2012, pp. 1135–1140.

[7] J. Munoz-Gama, N. Martin, C. Fernandez-Llatas, O. A. Johnson, M. Sepúlveda, E. Helm, V. Galvez-Yanjari, E. Rojas, A. Martinez-Millana, D. Aloini, I. A. Amantea, R. Andrews, M. Arias, I. Beerepoot, E. Benevento, A. Burattin, D. Capurro, J. Carmona, M. Comuzzi, B. Dalmas, R. de la Fuente, C. Di Francescomarino, C. Di Ciccio, R. Gatta, C. Ghidini, F. Gonzalez-Lopez, G. Ibanez-Sanchez, H. B. Klasky, A. Prima Kurniati, X. Lu, F. Mannhardt, R. Mans, M. Marcos, R. Medeiros de Carvalho, M. Pegoraro, S. K. Poon, L. Pufahl, H. A. Reijers, S. Remy, S. Rinderle-Ma, L. Sacchi, F. Seoane, M. Song, A. Stefanini, E. Sulis, A. H. ter Hofstede, P. J. Toussaint, V. Traver, Z. Valero-Ramon, I. van de Weerd, W. M. van der Aalst, R. Vanwersch, M. Weske, M. T. Wynn, F. Zerbato, Process mining for healthcare: Characteristics and challenges, Journal of Biomedical Informatics 127 (2022) 103994.

[8] W. M. P. Van Der Aalst, Process mining, 2011.

[9] E. Rojas, J. Munoz-Gama, M. Sepúlveda, D. Capurro, Process mining in healthcare: A literature review, Journal of Biomedical Informatics 61 (2016) 224–236.

[10] A. Augusto, R. Conforti, M. Dumas, M. L. Rosa, F. M. Maggi, A. Marrella, M. Mecella, A. Soo, Automated discovery of process models from event logs: Review and benchmark, IEEE Transactions on Knowledge and Data Engineering 31 (4) (2019) 686–705.

[11] W. M. van der Aalst, Process mining: discovering and improving spaghetti and lasagna processes, in: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2011, pp. 1–7.

[12] M. Pesic, H. Schonenberg, W. M. van der Aalst, Declare: Full support for loosely-structured processes, in: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 2007, pp. 287–287.

[13] M. Rovani, F. M. Maggi, M. de Leoni, W. M. van der Aalst, Declarative process mining in healthcare, Expert Systems with Applications 42 (23) (2015) 9236–9251.

[14] A. Pnueli, The temporal logic of programs, in: 18th Annual Symposium on Foundations of Computer Science (sfcs 1977), 1977, pp. 46–57.

[15] A. Burattin, F. M. Maggi, A. Sperduti, Conformance checking based on multi-perspective declarative process models, Expert Systems with Applications 65 (2016) 194–211.

[16] A. Alman, C. D. Ciccio, D. Haas, F. M. Maggi, A. Nolte, Rule mining with rum, in: 2020 2nd International Conference on Process Mining (ICPM), 2020, pp. 121–128.

[17] M. de Leoni, W. M. van der Aalst, M. Dees, A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs, Information Systems 56 (2016) 235–257.

[18] V. Leno, M. Dumas, F. M. Maggi, M. La Rosa, A. Polyvyanyy, Automated discovery of declarative process models with correlated data conditions, Information Systems 89 (2020) 101482.

[19] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, S. D. Brown, An introduction to decision tree modeling, Journal of Chemometrics 18 (6) (2004) 275–285.

[20] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 785–794.

[21] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, Tech. rep. (9 1985).

[22] Whitemech, GitHub - Whitemech/flloat: From LTLF/LDLF to automata.

[23] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[24] L. Breiman, Classification and regression trees, Routledge, 2017.

[25] I. H. Witten, E. Frank, Data mining: practical machine learning tools and techniques with java implementations, Acm Sigmod Record 31 (1) (2002) 76–77.

[26] J. R. Quinlan, C4.5: Programs for Machine learning, 1992.

[27] G. W. Eveborn, H. Schirmer, G. Heggelund, P. Lunde, K. Rasmussen, The evolving epidemiology of valvular aortic Stenosis. The Tromsø Study, Heart 99 (6) (2012) 396–400.

[28] H. G. Saldívar, L. V. Alaminos, C. Rodríguez-Pascual, G. De La Morena, C. Fernández-Golfín, C. Amorós, M. B. Alonso, L. M. Dolz, A. A. Solé, G. Guzmán-Martínez, J. J. Gómez-Doblas, A. A. Jiménez, M. E. Fuentes, M. R. Ortiz, P. Avanzas, E. Abu-Assi, T. Ripoll-Vera, O. Díaz-Castro, E. P. Osinalde, E. Bernal, M. Martínez-Sellés, Prognosis of patients with severe aortic stenosis after the decision to perform an intervention, Revista Española de Cardiología (English Edition) 72 (5) (2019) 392–397.

[29] R. F. Gomis, M. Mata-Cases, D. Mauricio, S. A. Menéndez, J. E. Muñoz, J. J. M. Bravo, C. M. Fernández-Santos, D. Orozco-Beltrán, L. Rodríguez-Mañas, C. Villalba, J. A. Martínez, Aspectos metodológicos de los Procesos Asistenciales Integrados (PAI), Revista De Calidad Asistencial 32 (4) (2017) 234–239.

[30] V. González, C. Peña, C. Neiro, D. López, Proceso asistencial integrado de estenosis aórtica, Servivio de Cardiología del Complejo Hospitalario Clínico Universitario de Santiago (2021).