Journal of Information and Data Management, 2024, 15:1, doi: 10.5753/jidm.2024.3337

Machine Learning Model Explainability supported by Data Explainability: a Provenance-Based Approach

Rosana Leandro de Oliveira [**Instituto Militar de Engenharia** | ***rosanasleandro@ime.eb.br***] [**Centro de Análises de Sistemas Navais** | ***rosana.leandro@marinha.mil.br***]
Julio Cesar Duarte [**Instituto Militar de Engenharia** | ***duarte@ime.eb.br***]
Kelli de Faria Cordeiro [**Instituto Militar de Engenharia** | ***kelli@ime.eb.br***] [**Ministério da Defesa** | ***kelli.cordeiro@defesa.gov.br***]

Programa de Pós-Graduação em Sistemas e Computação – Instituto Militar de Engenharia (IME), Praça Gen. Tibúrcio, 80 - Urca, Rio de Janeiro - RJ, 22290-270 - Brazil.

Received: 20 April 2023 • **Published:** 21 February 2024

Abstract The task of explaining the result of Machine Learning (ML) predictive models has become critically important nowadays, given the necessity to improve the results' reliability. Several techniques have been used to explain the prediction of ML models, and some research works explore the use of data provenance in ML cycle phases. However, there is a gap in relating the provenance data with model explainability provided by Explainable Artificial Intelligence (XAI) techniques. To address this issue, this work presents an approach to capture provenance data, mainly in the pre-processing phase, and relate it to the results of explainability techniques. To support that, a relational data model was also proposed and is the basis for our concept of data explainability. Furthermore, a graphic visualization was developed to better present the improved technique. The experiments' results showed that the improvement of the ML explainability techniques was reached mainly by the understanding of the attributes' derivation, which built the model, enabled by data explainability.

Keywords: Data Pre-processing, Machine Learning, Data Provenance, Explainability

1 Introduction

In recent years, Machine Learning (ML) algorithms have contributed to the knowledge discovery process. With the growth of data volume and computational power, these algorithms also increased their performance, allowing impressive results, close to human response, and, for some complex tasks, even better. On the other hand, these new solutions, which can be seen as black boxes, made the results less understandable to human understanding.

The difficulty in understanding the prediction of an algorithm can limit its use, since it makes its results less reliable, a fundamental component mainly in areas whose prediction can affect human life. Since ML-derived systems are generally involved in increasingly sensitive processes, the search for greater interpretability of the model became a critical factor in AI. Explainable AI (XAI), which is a relatively new area of AI, has emerged in a way that allows responses to be justified while increasing confidence in the model results. Several XAI studies and techniques have been conducted in this area, showing its benefits [Barredo Arrieta *et al.*, 2020].

In the same way, input data for an ML model need to be treated, either for adaptation to the training algorithm or for improving its quality, generating a model with better performance. In this sense, capturing operations performed in the pre-processing phase helps to understand the treatment performed on data and the possible influence that each operation generated on the model result. For this purpose, a set of operators is used to process and improve data quality.

Data provenance provides a description of the data origin and its deriving process. In ML, capturing data provenance can help interpret the results, through knowledge of the performed operations on data and the attributes used in the model derivation. According to Freire *et al.* [2008], this captured provenance can be either prospective or retrospective. Prospective provenance refers to the process of capturing the task specification, while retrospective provenance refers to the capturing of the steps that were performed and the execution environment information.

Some authors have explored the need for explainability in AI systems to be more comprehensive, affecting other phases of the ML cycle, and not just the model training [Tsakalakis *et al.*, 2021; Scherzinger *et al.*, 2019; Jentzsch and Hochgeschwender, 2019; Jaigirdar *et al.*, 2020]. Thus, in this work, we consider that ML explainability is provided by two distinct but related areas: (i) data explainability, supplied by data provenance, mainly in the pre-processing phase of the ML life-cycle, and (ii) model explainability, supplied by XAI techniques.

Some recent work handle capturing data provenance for ML and focus on the pre-processing phase [Chapman *et al.*, 2022; Namaki *et al.*, 2020; Moura *et al.*, 2021]. Others emphasize the learning phase [Hartley and Olsson, 2020], and some concentrate on gathering the whole ML cycle [Souza *et al.*, 2019]. However, few studies have been found so far regarding data explainability from the pre-processing provenance and none of them relates data explainability with model explainability.

Therefore, our motivation is to present an approach called *Explainable Machine Learning Model supported by Pre-processing Provenance* (xMML-PPP) which aims to capture and recover provenance information, especially in the pre-processing phase of the ML cycle, to relate the pre-processing phase provenance with model explainability.

It is important to highlight that the previously mentioned studies use approaches to capture provenance in an ML context, each one addressing a single phase or its entire cycle. They also have different objectives, mostly aimed at the experiment's reproducibility or explainability. However, our proposal is not simply another way of capturing the provenance in the ML environment, but an approach that improves the understanding of the ML models' explainability, adding, for this, data provenance, with a focus on the pre-processing phase, which is called here data explainability.

Two databases were used for the application of the approach: the traditional Titanic dataset and a dataset containing demographic information of patients tested with the SARS-CoV-2 virus. The experiments that received more pre-processing operations showed a performance improvement, and capturing the provenance of this information helped understand the derivation of the data, especially for the derived attributes.

To achieve its purpose, this article is organized according to the following structure. In section 2, some research on ML provenance is presented, while in section 3, the xMML-PPP approach is proposed. In section 4, two case studies are carried out that use this proposed approach, and finally, in section 5, we conclude our arguments, presenting possible future work.

2 Related Work

According to Herschel *et al.* [2017], provenance refers to any information that describes the production process of a final product, which can be any artifact. from a piece of information to a physical object. For scientific experiments, provenance aids in interpreting and understanding results, making it possible to examine the steps' sequence that contributed to a particular result, to verify input data, and to reproduce a result [Davidson and Freire, 2008].

Several provenance studies targeting different phases of the ML life cycle have been proposed. Chapman *et al.* [2020] implements a Python library for provenance capture, where annotations are associated with relational algebra operators that describe their effects on individual data elements. In recent work, Chapman *et al.* [2022] presents Data Provenance for Data Science (DPDS), a tool that helps data specialists to collect, store and investigate the provenance of each individual element in a dataset. Moura *et al.* [2021] presents an assistant that helps a non-specialist user in the selection of data pre-processing operators, in addition to capturing these operators. The tool is built based on a proposed domain reference ontology. In Namaki *et al.* [2020], the proposed provenance is also related to the pre-processing phase. The proposed idea is to track which columns in a dataset are used to derive the attributes of a specific model.

Rupprecht *et al.* [2020] proposes a provenance collection system for data science environments, focusing on the entire ML life-cycle. The methodology of this work is to capture provenance, integrating with the runtime to automatically track static and runtime configuration parameters. Finally, Souza *et al.* [2019] proposes a data provenance representation for workflows in the ML life-cycle, in large-scale projects, considering data transformation, from data curation of raw data to the generation of trained models.

Table 1 presents a comparison that classifies the related work, according to the following criteria:

- **1:** Indicates what type of provenance the work captures. R – Retrospective, P – Prospective, or B – Both;
- **2:** Indicates whether provenance encompasses capturing life-cycle pre-processing operators;
- **3:** Indicates whether the work captures model explainability (XAI technique);
- **4:** Indicates whether provenance capture is based on the W3C PROV model.
- **5:** Indicates whether the approach captures other phases of the ML life-cycle.
- **6:** Indicates whether the provenance scope refers to a simple (S) or complex (C) workflow. A simple workflow is one where a single workflow is considered to run, and a complex refers to systems where multiple workflows can interconnect.

Table 1. Related Word Comparison Features Summary

	Criteria					
Work	**1**	**2**	**3**	**4**	**5**	**6**
Chapman *et al.* [2020]	R	✓		✓		S
Chapman *et al.* [2022]	R	✓		✓		S
Moura *et al.* [2021]	R	✓				S
Namaki *et al.* [2020]	R	✓				S
Rupprecht *et al.* [2020]	R	✓			✓	S
Souza *et al.* [2019]	B	✓		✓		C
This work	B	✓	✓	✓	✓	S

The present work mainly contemplates the phases of the model pre-processing and explanation. As far as it is possible to investigate, this proposal differs from the related work since the cited works here do not aim to unite data explainability (data provenance) with model explainability (XAI technique).

3 The xMML-PPP Approach

The goal of the xMML-PPP approach is to contribute to ML explainability and, therefore, encompasses the entire ML life-cycle, from data collection to post-training, where the model explainability occurs. To achieve this goal, data from each phase, that can contribute to explainability, are collected. However, data explainability occurs mainly in the data pre-processing phase, making it possible to know the treatment that data received before being used in training.

As the provenance model, we adopted a subset of the PROV-DM [W3C, 2013] from W3C, which is a widely accepted ontology that defines provenance documents and supports RDF and other serialization formats for better interoperability. In the PROV model, an entity is defined as "a physical, digital, conceptual, or other kind of thing with some fixed aspects". In our context, these aspects correspond to the attributes in the dataset. On the other hand, an activity is defined as "something that occurs over a period of time and acts upon or with entities". In our context, activities represent any preprocessing or data manipulation performed on the attributes.

We define data explainability as a technique that captures data provenance from the pre-processing phase while improving the model explainability understanding by creating relations between them to increase the reliability of ML results. Thus, we differ data explainability from data provenance due to its inner relation with ML model explainability.

Additionally, the approach also considers the storage of performance data and the result of the contribution of each attribute to the model, thus making it possible to relate the configuration and performance of the model with the pre-processing treatment, in addition to the respective attributes' contributions. In Figure 1, a simplified view of the activities of the xMML-PPP approach can be observed.

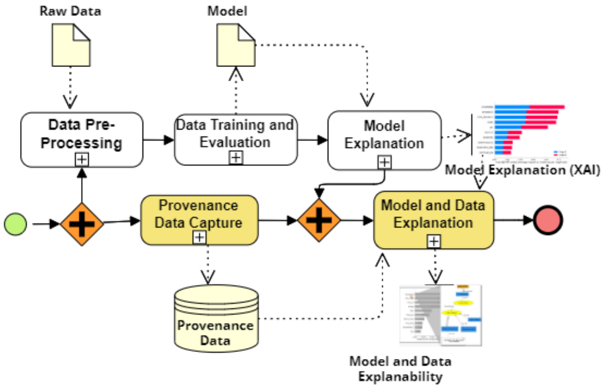

Figure 1. xMML-PPP Process – Overview – The processes highlighted in yellow are the major contribution of the approach.

The "Data Pre-Processing" sub-process starts with the activities of dataset loading and visualization, which are processed throughout the workflow. Also, in this sub-process, the data pre-processing activities are carried out. In this step, dataset information, such as name, number of rows and columns, size and location, as well as the pre-processing operations information, for each operation performed in this phase, is sent to the provenance data repository by the parallel activities of the "Provenance Data Capture" sub-process. In addition to that, activities are performed to relate and store the attribute description with the attribute information, such as labels and data types. The attribute description refers to the meaning of each attribute. For instance, using the "Parch" attribute from the Titanic dataset, the attribute description value is "of parents and children aboard the Titanic". On the other hand, the attribute information for this attribute is "Parch" and "integer" values for the label and data type information, respectively.

In the "Data Training and Evaluation" sub-process, the pre-processed data are trained, after choosing the algorithm and its parameters. In this step, the training information data such as algorithm, parameters, and values used, in addition to information about the trained model performance is sent to the provenance data repository by the parallel activities of the "Provenance Data Capture" sub-process.

In the "Model Explanation" sub-process, activities are carried out to configure and generate model explainability using XAI tools. In this step, information such as the method and dataset (training or testing) used, and contribution values for each attribute is sent to the provenance data repository by the parallel activities of the "Provenance Data Capture" sub-process.

Finally, in the "Model and Data Explanation", after the result of the model explanation and the contribution of each attribute, data stored in the provenance repository is retrieved. Through queries, it obtains information about the treatment that each attribute received for a better understanding of the operations that contributed to the result.

It is worth mentioning that model explainability is carried out after the model is ready and evaluated, precisely to understand its result.

3.1 xMML-PPP - Data Model

To meet its objective, the xMML-PPP approach specifies the model (Figure 2) for data representation that is related to the structure of the data flow and corresponds to the prospective provenance and the execution flow of the pre-processing operations, which corresponds to the retrospective provenance.

In Figure 2, entities highlighted in green have basic information related to the workflow, the dataset, and its original attribute information. The entities highlighted in gray refer to the activities of the pre-processing operations performed on the dataset. The entities highlighted in yellow refer to the information of the experiments carried out in the workflow, to store the provenance of each experiment configuration and its respective results. Finally, the entities highlighted in blue refer to the XAI configuration and their result information.

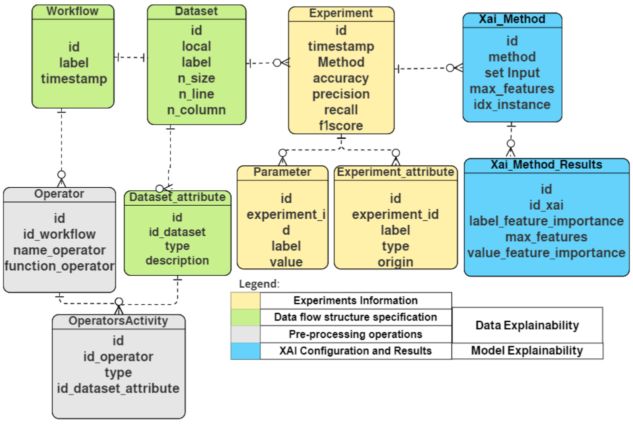

Figure 2. The Data Model displays various entities highlighted in different colors within a workflow: green, gray, yellow, and blue, which represent essential workflow information, pre-processing activities, experiment information, and configuration/results of XAI, respectively.

3.2 xMML-PPP - Tool

The "xMML-PPP" tool was developed according to the objectives and specificities of the xMML-PPP approach for its validation, and, thus, it implements the activities described in Figure 1. It was developed in Python[van Rossum, 1995] using the *Streamlit*[1] framework. Streamlit is a Python library for creating web applications without the need to code its front end. The data repository was implemented using the PostgreSQL[Stonebraker *et al.*, 1990] DBMS and codified with the object-relational mapping library SQLAlchemy.[2]

4 Application and Evaluation of xMML-PPP

To evaluate our approach, experiments were carried out with two different datasets. The first one is the traditional base for the Titanic [Kaggle, 2022], a ship that sank in 1912 on its first voyage. This database was used as a didactic example to illustrate the change in model performance derived from the pre-processing operations.

The second dataset is derived from a contemporary problem. Due to the recent COVID-19 pandemic, many studies were conducted to apply ML techniques to the solution of related problems. Some of them aimed to understand the disease risk factors, to predict possible contamination or even a possible mortality rate due to the virus. Demographic data and pre-existing health conditions allow predictive models to be customized to predict whether a patient will be hospitalized, die, or require intensive health care [Wollenstein-Betech *et al.*, 2020].

The original dataset was produced and made available by the Mexican Government [de Salud, 2020], however, in the present article, a modified version of the same dataset was used, derived from the work of Muhammad *et al.* [2021], whose dictionary has already been translated into English and refers to the observation period of April 12th 2020 to March 3rd 2020. This dataset is available in Franklin [2020].

Thus, this second dataset is a set of epidemiological data from patients in Mexico with suspected contamination by the COVID-19 virus and who underwent a PCR-RT test. This dataset was chosen since it is a use case from the health area, with recent data, that better reflects the current mandatory use of explainability.

4.1 Experiments Settings and Results

In both datasets, two derived experiments are performed. Each first experiment is always carried out with the minimum amount of pre-processing necessary just for the application of the algorithms. In the second experiment, on the other hand, the pre-processing operations are expanded to improve the model's results. To simplify the comparison between the approaches and considering that the objective of the study is not to establish a new benchmark for datasets with or without pre-processing, we opted to conduct a single holdout experiment for each dataset instead of a full cross-validation approach.

In all experiments, the Random Forest (RF) [Breiman, 2001] algorithm was used, implemented using the scikit-learn library [Pedregosa *et al.*, 2011]. RF is a supervised learning algorithm that can be used for classification and regression tasks and uses several decision trees in its implementation, where each tree is trained with a different view of the dataset, either in terms of the number of available attributes or even by randomly subsampling the examples. This algorithm does not allow training with missing values, and the categorical attributes need to be encoded for training.

RF was selected since it has already been used in other studies on COVID-19, obtaining good results, while also presenting good results in other studies with the Titanic dataset. For all experiments, the following hyperparameters are used: *estimators*, number of trees; *max_features*, number of features to consider for the best split; *min_samples_split*, minimum number of samples needed to split an internal node; *min_samples_leaf*, minimum number of samples needed to form a leaf node; e *max_depth*, maximum depth of a tree. The values for the parameters are presented in the configuration of the experiments on each dataset. All the other parameters are set to their default values;

To compare the models' results, the metrics of precision, recall, and f1-measure are used. Precision evaluates how many positive predictions are actually positive, while recall is the proportion of true positives among all predictions that are actually positive. Finally, F1 is simply the result of the harmonic mean between precision and recall. The averaging for the classes used in all experiments metrics was 'weighted'.[3]

4.1.1 Titanic Dataset

The Titanic dataset contains 12 columns and 891 instances and provides the following attributes: *PassengerId, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Survived*, and *Embarked*. The goal of this study is to create a predictive model that answers the question: "Which characteristics in people determined their likelihood of survival?". To answer this question, the available passenger info is used, such as age, sex, economy class, etc. For the first experiment, only the following basic pre-processing operations are performed, so that the RF could be executed: filling in null values and transforming and deleting categorical attributes.

Thus, the nominal attributes *Sex* and *Embarked* are encoded using the *OneHotEncoder* operator. To fill in the null values for the Embarked attribute, its mode was used and for the *Age* attribute, its average was used. The attributes *PassengerId*, *Ticket* and *Cabin* were removed because they represent only possible passenger identification. For the second experiment, the same operations as the first experiment were performed. Additionally, on this dataset, a previous exploratory analysis of data was carried out, where some characteristics were verified, such as females have survived more than males, and people from class 1 have survived more than people from classes 2 and 3, among others.

Thus, the following pre-processing operations are performed for this experiment: the *Title* attribute is created by extracting the title and passenger name. For this particular

[1] https://https://streamlit.io/

[2] https://www.sqlalchemy.org/

[3] weighted calculate metrics for each label and find the average weighted by the support (the number of true instances for each label). This alters 'macro' to account for label imbalance, and it can result in an F1 that is not between precision and recall.

attribute, only the *Master*, *Miss*, *Mr.*, *Mrs.* values are kept since they represented the most amount of examples. Instances whose values differed from those have their values for this attribute replaced by *Others*. A *Lastname* attribute is also created, by extracting the last name from the passenger name and contributing to the engineering of another derived attribute, *Groupsize*, which refers to the number of women or children with the same last name and, possibly, from the same family. Another created attribute is the *Family_Size*, which represents the sum of all members of a family on board the Titanic, which is derived from the *Parch* and *SibSp* attributes of each instance.

In addition, the categorical attributes (*Embarked*, *Title*, and *Sex*) are encoded with the *OneHotEncoder* operator. The Name, Ticket, and Lastname attributes are then deleted, and the attribute values are normalized using the *MinMaxScaler* operator. The values of the parameters used in the RF configuration are max_features: sqrt[4]; *min_samples_split*: 2; *min_samples_leaf*: 2; *max_depth*: 4; e *num_estimators*: 100.

Table 2 summarizes the RF performance in the two experiments with the Titanic database.

Table 2. Titanic Dataset Experiments – Results

Exp	Accuracy	Precision	Recall	F1-measure
1	81.72	82.48	81.72	81.17
2	84.33	84.32	84.33	84.20

It can be observed that, even with few pre-processing operations performed in the first experiment, RF performs well in the task. In the second experiment, with other pre-processing operations, including feature engineering, an even more significant improvement in its performance can be seen.

4.1.2 COVID-19 Dataset

The used COVID-19 Mexico dataset has 41 columns and 263,007 instances containing demographic data such as age, gender, nationality, and immigrant status, in addition to clinical data, pre-existing diseases of the patient such as diabetes, asthma, hypertension, and obesity, among other diseases, pregnancy, smoking, temporal information, such as date of symptoms onset, date of health facility admission, and possible date of death, as well as the result of the RT-PCR test for the disease. The objective of the experiments in this study is to classify the mortality of confirmed cases of COVID-19 through the result of a PCR test. For this dataset, it is necessary some preparation before it can be inserted into the xMML-PPP tool through some operations. In this article, the original attribute names for this dataset have been translated into English for better understanding purposes.

The target attribute, DEAD, is created, which is extracted from the original attribute *death_date* that represents the date of death. This new attribute receives the value 0 to indicate the negative class, and 1 to indicate the positive class. Next, the attributes *id*, *id_registry*, and *death_date* are removed. Instances with unknown pre-existing disease information indicated in the dataset by identifiers *98* and *99* are also removed. The instances whose PCR-RT test result is positive are selected since the study objective is to classify whether patients who tested positive for COVID-19 died. After this selection, the attribute that indicated the exam result is removed, and the dataset is finally split. The final dataset used in the experiments contains 38 columns and 12.874 examples.

For the first experiment, as little pre-processing as possible is performed, only what is necessary for the dataset to be able to be trained by RF. This phase activates the simplified activity diagram "Receive and prepare data" phase. In this case, the existing categorical attributes in the dataset, such as dates and city names, are removed for the first experiment. The same operations for removing categorical attributes existing in the data set performed in the first experiment are initially performed for the second experiment. Additionally, some attributes with regional information are also removed. A previous exploratory analysis of the data is also carried out, identifying that patients over 40 years of age who were hospitalized had a higher incidence of death. Thus, three attributes *(age_group, total_disease, and has_highrisk)* are created.

The created *age_group* receives the value 0 when the patient age is less than 40 years and 1 for patients aged equal to or greater than 40. The created attribute *total_disease* indicates the number of comorbidities presented by the patient infected by the virus. The attribute *has_highrisk* received 1, in case the patient is aged 40 years or older and was hospitalized. Thus, the creation of this attribute is carried out as an attempt to help the model identify a more significant number of examples of the minority class. The model is then trained with a total of 30 attributes.

Finally, a model capable of correctly predicting the target attribute is trained with the following RF hyperparameters: *max_features*: sqrt; *min_samples_split*: 2; *min_samples_leaf*: 1; *max_depth*: 16; e *num_estimators*: 200. Table 3 shows the performance result for the two carried-out experiments.

Table 3. COVID-19 Mexico Experiments - Results

Exp	Accuracy	Precision	Recall	F1-measure
1	84.29	84.29	83.49	83.63
2	84.52	83.81	84.52	83.97

It is possible to notice that even with few examples, the model of experiment 2 can classify patients in class 1 better than the model of experiment 1.

4.2 Explainability of experiments' results

The explainability of the model is performed by applying XAI techniques. There are several XAI techniques, which can be used according to the explanation scope, data types, and techniques approach. For this work, the global scope is considered, and the approach of the technique selected is the relevance of attributes. To explain the model in this work, the SHAP technique [Lundberg and Lee, 2017] is used in all experiments. SHAP is a game theory-based method that improves explainability by calculating importance values for each attribute using individual predictions. It was chosen among other existing explainability techniques because it is

[4] sqrt indicates that the square root of the number of attributes is used.

one of the most comprehensive methods for explaining black box methods, helping visualize interactions and attributes importance [Linardatos *et al*., 2021].

XAI techniques are used to help understand the model output, however, the treatment information carried out on data that made up the model and influenced the result is unknown with XAI. The provenance data thus helps in knowing how these data were treated. In this section, explainability graphs are created using the SHAP technique, to explain the contribution of the attributes in each experiment. We also present queries that contribute to the understanding of data derived from the model. Figure 3 refers to the SHAP graph derived from experiment 1, the Titanic dataset, and Figure 4 refers to the SHAP graph from its second experiment.

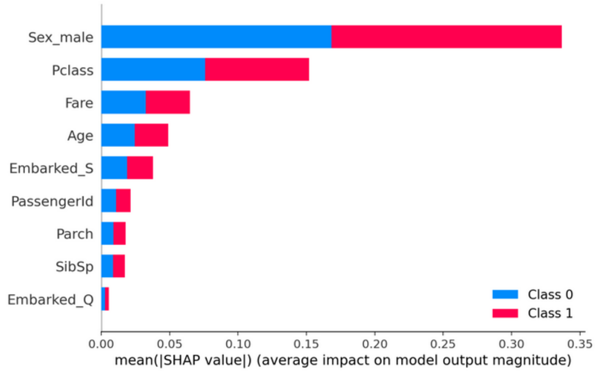

Figure 3. SHAP Chart for the 1st experiment of Titanic dataset – Illustrates the significant contribution of the 'Sex_male' attribute to this experiment.

For the SHAP Chart illustrated in Figure 3, it can be seen that the attributes *Sex_Male, Pclass*, and *Fare* are the attributes that most contribute to the model result.

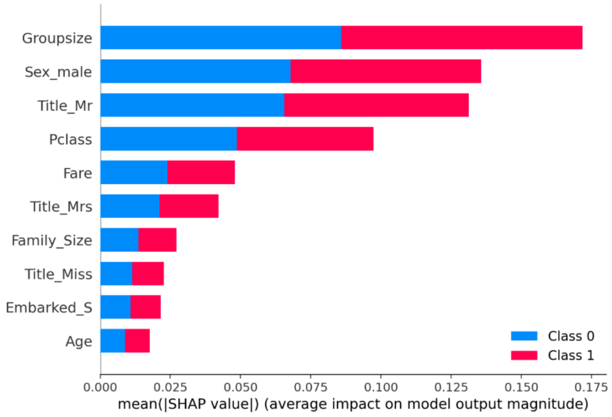

Figure 4. SHAP Chart for the 2nd experiment of Titanic – Illustrates the significant contribution of the constructed attribute Groupsize to the model.

However, according to Figure 4, it can be seen that the derived attribute *Groupsize*, and the attributes *Sex_Male* and *Title_Mr* are the attributes that most contribute to this model result. Thus, it is clear that the derived attribute is the most important for the model result. It is also possible to obtain more information regarding the attributes that are trained and appeared in the SHAP graph through queries to the data repository. Using the *Groupsize* attribute as an example, which appears as the attribute with the highest contribution, queries can be carried out to obtain the following attribute information: creation source, pre-processing operations, and model contribution value. The result of the information from these queries can be seen in Figures 5, 6, and 7.

The figures illustrating the retrieval of information stored in the data repository are divided into two parts. The first part always contains the SQL command that retrieves the desired information, while the second part always presents a table with the resulting retrieved information. Figure 5 shows, in the first part, the SQL command to retrieve the information regarding the origin of the derived attribute Groupsize, and, then, the query's result.

```
SELECT Experiment_attribute.label,
experiment_attribute.origin
FROM  experiment_attribute,
experiment, dataset, workflow
WHERE experiment_attribute.experimentid=
experiment.id
AND  experiment.datasetid=dataset.id
AND dataset.workflowid = workflow.id
AND  workflow.id = 152
AND experiment.attribute.label
LIKE 'Groupsize';
```

	attribute	source
0	Groupsize	df['Groupsize'] = df['Lastname'].apply(lambda x: df.loc[(df['Sex']=='female') | (df['Title']=='Master')].loc[df.loc[(df['Sex']=='female') | (df['Title']=='Master')] ['Lastname']==x]['Survived'].count())

Figure 5. Provenance information regarding the origin (construction) of the derived attribute Groupsize.

```
SELECT "operatorsActivity".name,
"operatorsActivity".function,
"operatorsActivity".label_attribute
FROM  "operatorsActivity", workflow
WHERE "operatorsActivity".workflowid=
workflow.id AND  workflow.id = 152
AND "operatorsActivity".label_attribute
LIKE 'Groupsize';
```

	name	function	label
0	IncludeColumn	Attribute Construction	Groupsize
1	MinMaxScaler	Data Normalization	Groupsize
1	TrainTestSplit	Data Partition	Groupsize

Figure 6. Information regarding the preprocessing provenance of the *Groupsize* attribute in Experiment 2.

The purpose of these queries is to bring explainability of the data to the model, that is, information additional to that provided by the XAI charts. The figure 8 refers to an example of a modified interface of the SHAP chart that enables the visualization functionality of PROV provenance information for the*Groupsize* attribute, presented upon selecting this attribute. Figures 9 and 10 show the SHAP graphs generated from the models of experiments 1 and 2, respectively, referring to the Mexican epidemiological dataset.

```
SELECT "xai_Results".label_feature_importance,
"xai_Results".value_feature_importance,
FROM  "xai_Results",xai, experiment,
dataset, workflow
WHERE "xai_Results"."int"
AND xai.experimentid=experiment.id
AND experiment.datasetid=dataset.id
AND dataset.workdlowid=workflow.id
AND workflow.id = 152
AND "xai_Results".label_feature_importance
LIKE 'Groupsize';
```

	label_feature_importance	value_feature_importance
0	Groupsize	53.4932

Figure 7. Information about the contribution value of the constructed attribute Groupsize in Experiment 2.

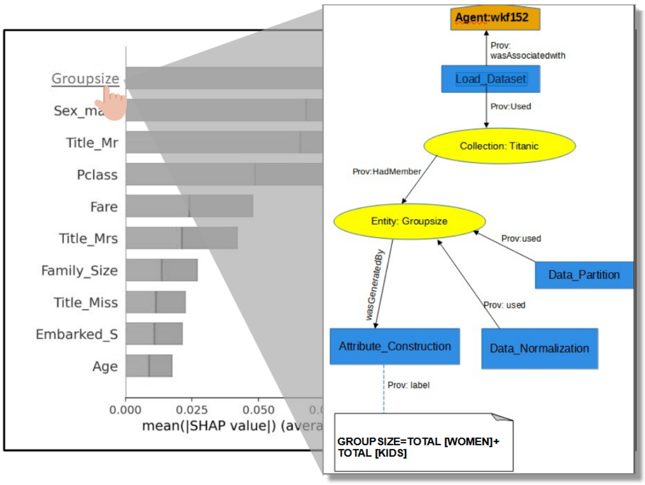

Figure 8. This is an example of enhancing the SHAP chart by visualizing the operations performed (activities) on the attribute (entity) Groupsize.

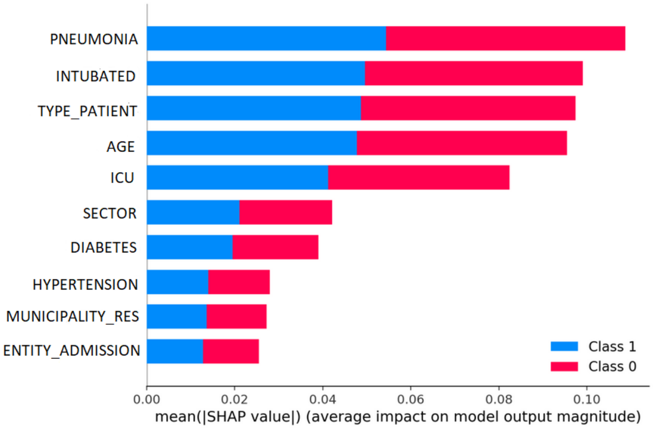

Figure 9. SHAP plot for the 1st experiment of the Covid-19 dataset – Illustrates the highest contribution of the PNEUMONIA attribute. The *intubated*, *type_patient*, and *AGE* attributes had a very similar contribution to this experiment.

The xMML-PPP tool already has the main queries available for retrieving information in the data repository without the need to write SQL codes. However, it is essential to be able to build free queries that can better respond to specific questions. In this way, SQL codes can also be accepted. In order to get an idea of important query questions that provide data explainability, some examples are presented in Table 4. The *workflow* referring to experiment 2 of the dataset of patients with COVID-19 from Mexico was used as an example to answer these queries.

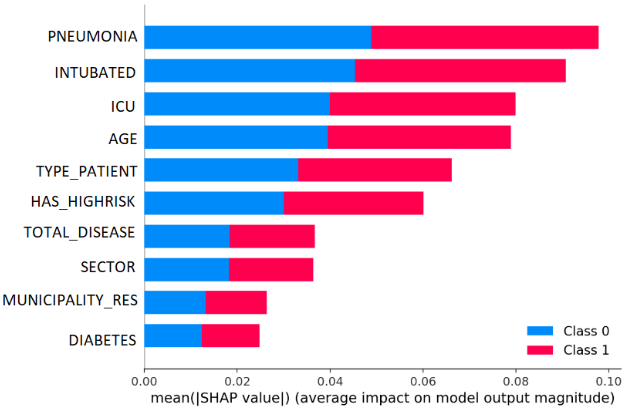

Figure 10. SHAP plot for the 2nd experiment of the Covid-19 data set – Illustrates the largest contribution of the *Pneumonia* attribute, followed by the *intubated* attribute, and the *icu* and *age* attributes appear with a very similar contribution for this experiment.

Table 4. Examples of Provenance Query Questions

Q1	"Given a trained model, what attributes derived the training set of a specific experiment?"
Q2	"Given a trained model, what were the constructed attributes and the construction origin of these attributes?"
Q3	"Given a trained model from a specific workflow, what were the values for all the parameters and the evaluation metric values associated with the model?"
Q4	"Given a trained model, what is the meaning (attribute description) of specific attributes that most contributed to the result of the model? "
Q5	"Given a trained model, what were the preprocessing operations performed on the first three attributes that most contributed to the result of this model? "

The answer to Q1 is presented in Figure 11, which makes it possible to find the attributes that derive a given trained model. Knowing which attributes are derived from a model is important to know which features available in the dataset are considered to better train the model.

The answer to Q2 Figure 12 informs which of the attributes that generated the model are not original to the data set; that is, they are derived from other existing attributes and what the origin of the construction of these attributes is.

Query Q3 has been divided into two parts, for better visualization of the results. Figures 13 and 14 answer about the performance measures and parameters used in a given trained model, respectively. This information helps analyze the model performance and configuration.

Q4 (Figure 15) helps to retrieve the attributes' descriptions. Keeping the original attributes' description in the data repository can contribute to understanding the result of an XAI graph since, in some datasets, only the name of the attribute may not be enough to understand the meaning of the

```
SELECT experiment_attribute_label,
experiment_attribute_origin
FROM  experiment_attribute, experiment,
dataset, workflow
WHERE experiment_attribute.experiment.id=
experiment.id
AND experimentid.datasetid=dataset.id
AND dataset.workflowid=workflow.id
AND workflow.id = 146
```

	label	origin
0	entity_admission	dataset original attribute
1	entity_res	dataset original attribute
		- - -
26	icu	dataset original attribute
27	dead	dataset original attribute
28	total_ disease	df['total_disease'] = df[df.columns[22:31]]. sum(axis=1)
29	age_group	df['age_group'] = df.apply(lambda df: (0 if df['age'] <= 39 else 1), axis=1)
30	has_highrisk	df['has_highrisk'] = df.apply(lambda df: (1 if (df['age_group']==1 and df['type_patient']==2) else 0), axis=1)

Figure 11. Query Q1 and an extract of its results for Experiment 2 of the Covid-19 dataset – relative to the attributes that made up the training dataset.

```
SELECT experiment_attribute.label,
experiment_attribute.origin
FROM experiment_attribute, experiment,
dataset,workflow
WHERE experiment_attribute.experimentid=
experiment.id AND experiment.datasetid=
dataset.id AND dataset.workflowid=
workflow.id AND workflow.id=146
and experiment_attribute.label IN
(SELECT "operatorsActivity".label_attribute
AS Attribute
FROM "operatorsActivity",workflow
WHERE  "operatorsActivity".workflowid=
workflow.id AND workflow.id=146
AND "operatorsActivity".function=
'Attribute␣Construction')
```

	label	origin
0	total_ disease	df['total_disease'] = df[df.columns[22:31]].sum(axis=1)
1	age_ group	df['age_group'] = df.apply(lambda df: (0 if df['age'] <= 39 else 1), axis=1)
2	has_highrisk	df['has_highrisk'] = df.apply(lambda df: (1 if (df['age_group']==1 and df['type_patient']==2) else 0), axis=1)

Figure 12. Query Q2 and the extract concerning provenance information about the origin (construction) of the derived attributes that composed the training dataset.

attribute.

```
SELECT experiment.accuracy, experiment.recall,
    experiment.precision, experiment.f1score FROM
      experiment,
dataset,workflow
WHERE experiment.datasetid=
dataset.id AND dataset.workflowid= workflow.id
AND workflow.id = 146
```

	accuracy	recall	precision	f1score
0	0.8452	0.8452	0.8381	0.8397

Figure 13. Query Q3A and the result of the model evaluation measures.

```
SELECT parameter.label, parameter.value
FROM parameter,  experiment, dataset, workflow
WHERE parameter.experimentid=experiment.id AND
    experiment.datasetid=dataset.id AND dataset.
    workflowid = workflow.id AND workflow.id =
    146
GROUP BY experiment.id, parameter.label,
    parameter.value
```

	label	value
0	max_depth	16
1	max_features	sqrt
2	min_samples_leaf	1
3	min_samples_split	2
4	n_estimators	200
5	random_state	42

Figure 14. Query Q3B and the parameters and respective values used to train the model.

```
SELECT dataset_attribute.label,
dataset_attribute.description
FROM  dataset_attribute,dataset,workflow
WHERE dataset_attribute.datasetid=
dataset.id
AND dataset.workflowid=workflow.id
AND workflow.id=146
AND dataset_attribute.label IN
('pneumonia', 'intubated','icu', 'sector')
```

	label	description
0	sector	Identifies the type of institution of the National Health System that provided the care.
1	intubated	Identifies if the patient required intubation.
2	pneumonia	Identifies if the patient was diagnosed with pneumonia.
3	icu	Identifies if the patient was admitted to Intensive Care Unit.

Figure 15. Query Q4 and the description of specific attributes to better understand their meanings.

Finally, Q5 (Figure 16) answers the main objective of the xMML-PPP approach. After the model is trained, it is possible to know which pre-processing operations were carried out for each attribute that participated in the model, including those that contributed the most to the model performance.

Since such operations influence the model result, knowledge of this information can help to complement the understanding of the model behavior. For example, in the graph depicted in Figure 10, the attributes *pneumonia*, *intubated*, and *icu* appear with the most significant contribution to the model of experiment 2. By applying Q5, we can obtain information on which pre-processing operations were performed

```
SELECT "operatorsActivity".name, "
    operatorsActivity".function, "
    operatorsActivity".label_attribute
FROM "operatorsActivity",workflow
WHERE "operatorsActivity".
workflowid=workflow.id
AND workflow.id=146
AND "operatorsActivity".label_attribute
IN (SELECT  "xai_Results".
label_feature_importance
FROM "xai_Results",xai,experiment,
dataset,workflow
WHERE"xai_Results"."int"=xai.id
AND xai.experimentid = experiment.id
AND experiment.datasetid=dataset.id
AND dataset.workflowid=workflow.id
AND workflow.id= 146 limit 7)
```

	Name	Function	label_attribute
0	IncludeColumn	Attribute-Construction	total_disease
1	IncludeColumn	Attribute-Construction	has_highrisk
2	TrainTestSplit	DataPartition	type_patient
3	TrainTestSplit	DataPartition	intubated
4	TrainTestSplit	DataPartition	pneumonia
5	TrainTestSplit	DataPartition	age
6	TrainTestSplit	DataPartition	icu
7	TrainTestSplit	DataPartition	total_disease
8	TrainTestSplit	DataPartition	has_highrisk

Figure 16. Query Q5 and the preprocessing operations that were performed on the attributes with the highest contribution.

on these attributes, an example of data explainability, provided by the result of query Q5, supporting the model explainability, provided by the SHAP graph. It is also possible to complement the result analysis in an experiment, by uniting the result of searches Q1, Q3, and Q5, for example, to obtain information about the model configuration, performance, and pre-processing operations that the attributes that derived the model received, in addition to each attribute contribution importance, according to these configurations.

5 Conclusion

XAI techniques emerged to contribute to the so-called black-box models' understanding, which have low interpretability. However, the difficulty in understanding the predictions of these algorithms can limit their use, as it makes their results less reliable, which is not desirable, especially in areas whose prediction can affect human life.

In this article, we present xMML-PPP, a new approach that captures provenance data from the pre-processing phase for ML classification tasks. The purpose of this approach is to use data provenance, mainly from the pre-processing phase, which is called here data explainability, to complement model explainability, provided by XAI techniques. The approach presented a data model that allows relating data related to the pre-processing phase with model explainability data, in addition to the model's configuration and performance. This data is stored and captured by the tool presented in the approach, as seen in section 3.

The evaluation of the approach was carried out by conducting four experiments using two datasets, the Titanic dataset and an epidemiological dataset for COVID-19 cases in Mexico. It was observed that the information about the preprocessing operations, mainly for feature engineering, can contribute to the understanding of the derivation of the attributes that composes the model, mainly when these attributes have a significant contribution to the model performance. Additionally, it was verified that it is also possible to perform data analyses through queries that relate the preprocessing operations with the performance and result, thus enhancing model explainability.

The main limitation of the approach lies in complexity and interpretability: the preprocessing operations can involve complex transformations. The provided provenance may not always capture the full complexity of the preprocessing steps, making it difficult to interpret their impact on the final results. Future work includes implementing an interactive version of the SHAP chart in the approach tool which would allow the captured provenance information to be visualized graphically by selecting desired attributes, as demonstrated in the article's interface prototype. Another possibility is to replace the relational database with a graph database. This would allow more flexibility, especially for the inclusion of new entities, such as for storing runtime training information in a neural network, without worrying about changes to the database schema.

Funding

This work was partially supported by national funds through FINEP, Financiadora de Estudos e Projetos and FAPEB, Fundação de Apoio à Pesquisa, Desenvolvimento e Inovação do Exército Brasileiro, under project "Sistema de Sistemas de Comando e Controle" with reference nº 2904/20 under contract nº 01.20.0272.00.

Authors' Contributions

All authors contributed to the conception and writing of this study. Rosana Leandro developed the experiment's source code. All authors read and approved the final manuscript.

Availability of data and materials

The modified dataset, the built tool and the data model of this work are available in http://github.com/RosanaLeandro/ppm-ml

References

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115. DOI: https://doi.org/10.1016/j.inffus.2019.12.012.

Breiman, L. (2001). Machine learning, volume 45, number 1 - springerlink. *Machine Learning*, 45:5–32. DOI: 10.1023/A:1010933404324.

Chapman, A., Lauro, L., Missier, P., and Torlone, R. (2022). Dpds: Assisting data science with data provenance. *Proc. VLDB Endow.*, 15(12):3614–3617. DOI: 10.14778/3554821.3554857.

Chapman, A., Missier, P., Simonelli, G., and Torlone, R. (2020). Capturing and querying fine-grained provenance of preprocessing pipelines in data science. *Proc. VLDB Endow.*, 14(4):507–520. DOI: 10.14778/3436905.3436911.

Davidson, S. and Freire, J. (2008). Provenance and scientific workflows: Challenges and opportunities. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1345–1350. DOI: 10.1145/1376616.1376772.

de Salud, S. (2020). Datos abiertos dirección general de epidemiología. `https://www.gob.mx/salud/documentos/datos-abiertos-152127`. (Acessado em 04/08/2021).

Franklin, M. R. (2020). ''Kaggle: Mexico COVID-19 clinical data''. `https://www.kaggle.com/marianarfranklin/mexico-COVID19-clinical-data`. (Acessado em 02/10/2021).

Freire, J., Koop, D., Santos, E., and Silva, C. T. (2008). Provenance for computational tasks: A survey. *Computing in Science Engineering*, 10(3):11–21. DOI: 10.1109/MCSE.2008.79.

Hartley, M. and Olsson, T. S. (2020). dtoolai: Reproducibility for deep learning. *Patterns*, 1(5):100073. DOI: https://doi.org/10.1016/j.patter.2020.100073.

Herschel, M., Diestelkämper, R., and Ben Lahmar, H. (2017). A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906. DOI: 10.1007/s00778-017-0486-1.

Jaigirdar, F. T., Rudolph, C., Oliver, G., Watts, D., and Bain, C. (2020). What information is required for explainable ai? : A provenance-based research agenda and future challenges. In *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, pages 177–183. DOI: 10.1109/CIC50333.2020.00030.

Jentzsch, S. F. and Hochgeschwender, N. (2019). Don't forget your roots! using provenance data for transparent and explainable development of machine learning models. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)*, pages 37–40. DOI: 10.1109/ASEW.2019.00025.

Kaggle (2022). Titanic - machine learning from disaster. `https://www.kaggle.com/c/titanic/`. Acesso em: 26 de fevereiro 2022.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1). DOI: 10.3390/e23010018.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.. DOI: 10.5555/3295222.3295230.

Moura, L. d. A. L., da Silva, M. A. A., Cordeiro, K. d. F., and Cavalcanti, M. C. R. (2021). A well-founded ontology to support the preparation of training and test datasets. In *Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS 2021*, pages 99–110. SCITEPRESS. DOI: 10.5220/0010460000990110.

Muhammad, L., Algehyne, E. A., Usman, S. S., Ahmad, A., Chakraborty, C., and Mohammed, I. A. (2021). Supervised machine learning models for prediction of covid-19 infection using epidemiology dataset. *SN computer science*, 2(1):1–13. DOI: 10.1007/s42979-020-00394-7.

Namaki, M. H., Floratou, A., Psallidas, F., Krishnan, S., Agrawal, A., Wu, Y., Zhu, Y., and Weimer, M. (2020). Vamsa: Automated provenance tracking in data science scripts. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. DOI: 10.1145/3394486.3403205.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.* (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830. DOI: 10.5555/1953048.2078195.

Rupprecht, L., Davis, J. C., Arnold, C., Gur, Y., and Bhagwat, D. (2020). Improving reproducibility of data science pipelines through transparent provenance capture. *Proc. VLDB Endow.*, 13(12):3354–3368. DOI: 10.14778/3415478.3415556.

Scherzinger, S., Seifert, C., and Wiese, L. (2019). The best of both worlds: Challenges in linking provenance and explainability in distributed machine learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1620–1629. DOI: 10.1109/ICDCS.2019.00161.

Souza, R., Azevedo, L., Lourenço, V., Soares, E., Thiago, R., Brandão, R., Civitarese, D., Brazil, E., Moreno, M., Valduriez, P., *et al.* (2019). Provenance data in the machine learning lifecycle in computational science and engineering. In *2019 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*, pages 1–10. IEEE. DOI: 10.1109/WORKS49585.2019.00006.

Stonebraker, M., Rowe, L., and Hirohama, M. (1990). The implementation of postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):125–142. DOI: 10.1109/69.50912.

Tsakalakis, N., Stalla-Bourdillon, S., Carmichael, L., Huynh, T. D., Moreau, L., and Helal, A. (2021). The dual function of explanations: Why it is useful to compute explanations. *Computer Law & Security Review*, 41:105527. DOI: 10.1016/j.clsr.2020.105527.

van Rossum, G. (1995). Python reference manual.

W3C (2013). The prov data model. `https://www.w3.org/TR/PROV-DM`. (Acessado em 01/11/2021).

Wollenstein-Betech, S., Cassandras, C. G., and Paschalidis, I. C. (2020). Personalized predictive models for symptomatic covid-19 patients using basic preconditions: Hospitalizations, mortality, and the need for an icu or ventilator. *International Journal of Medical Informatics*, 142:104258. DOI: https://doi.org/10.1016/j.ijmedinf.2020.104258.