XMLPO: An Ontology for Explainable Machine Learning Pipeline

Donika XHANI [a,1], João Luiz REBELO MOREIRA [b], Marten VAN SINDEREN [b] and Luís FERREIRA PIRES [b]

[a] *IEBIS group, BMS Faculty, University of Twente*

[b] *SCS group, EEMCS Faculty, University of Twente*

ORCiD ID: Donika Xhani https://orcid.org/0000-0003-2536-0574, João Luiz Rebelo Moreira https://orcid.org/0000-0002-4547-7000, Marten van Sinderen https://orcid.org/0000-0001-7118-1353, Luís Ferreira Pires https://orcid.org/0000-0001-7432-7653

Abstract. Machine Learning (ML) models often operate as black-boxes, lacking transparency in their decision-making processes. Explainable Artificial Intelligence (XAI) aims to address the rationale behind these decisions, thereby enhancing the trustworthiness of ML models. In this paper, we propose an extension of the Explainable ML Workflows ontology, which was designed as a reference ontology with OntoUML, and implemented as an operational ontology with OWL. The Explainable ML workflows ontology reuses ML-Schema, which is a core ontology for representing ML algorithms. We have identified four main issues in the conceptualization of this ontology, namely the lack of feature categorization, the lack of data pre-processing methods, the shallow description of metadata related to training and testing, and the lack of detailed representation of XAI approaches and metrics. We addressed these four issues in the so-called Explainable ML Pipeline Ontology (XMLPO), which aims to provide a comprehensive description of the ML pipeline for XAI. XMLPO offers a deeper understanding of the entire ML pipeline, encompassing data input, pre-processing, model training and testing, and explanation processes. XMLPO was validated through a case study on the prediction of specific performance indicators in a manufacturing company, and the results of this validation showed that the ontology helps data scientists to better comprehend a ML pipeline and the features that influence the ML prediction model the most.

Keywords. Ontology, Explainable AI (XAI), Machine Learning pipeline, Semantics

1. Introduction

Organizations are increasingly turning to Machine Learning (ML) models to develop predictive or classification models for their data [1]. However, especially when these models are based on complex structures like neural networks or deep learning they are commonly referred to as black-boxes due to their inherent opacity. The lack of transparency in black-box ML models makes it difficult for their users to understand why the model predicts or classifies certain outcomes, and to correct erroneous feature selections, cre-

[1] Corresponding Author: Donika Xhani, d.xhani@utwente.nl

ating uncertainty about the reliability of the predictions and classifications [2,3]. Moreover, the complexity inherent in these black-box models poses significant challenges for humans to comprehend the reasoning and methods behind their outcomes [5,6].

Explainable Artificial Intelligence (XAI) encompasses methods and techniques designed to make ML models explicable, interpretable, transparent, and understandable for humans [7]. Understanding ML models is crucial as it enables users to recognize necessary adjustments in data pre-processing, model training, and testing procedures when the output is incorrect [8,9]. As stated by Adadi and Berrada [10]: "*It is not enough to just explain the model, the user has to understand it*".

An ontology can serve as an effective tool for precisely and clearly establishing the semantics of the essential components of a ML pipeline and explanation processes [11]. In the Explainable ML workflows ontology [13], which is based on an ontological analysis of ML-Schema [14], we described the ML pipeline and model explanation for the post-hoc approach, in which ML results are explained. However, from the lessons learned in the first version reported in [13], we have identified four main issues: (1) the ontology categorizes data as either input or pre-processed without revealing characteristics such as whether features are numerical or categorical, dependent or independent; (2) it focuses solely on pre-processed data, ignoring details on data pre-processing methods such as data cleaning, feature scaling, and encoding; (3) the 'Specific Module' [13], which refers to the implementation of ML models, is too simple, covering only metadata related to training and testing without considering other ML aspects like specific models, libraries, and evaluation techniques; (4) the 'Explanation Module' [13] lacks a categorization of XAI approaches, as well as the usage of metrics for evaluating the generated explanation from the XAI models.

The goal of this paper is to address these limitations with the Explainable ML Pipeline Ontology (XMLPO), as an extension of the Explainable ML workflows ontology [13]. In XMLPO, we improved the description of data input and pre-processing, of the characteristics of the ML model, of the usage of ML model evaluation techniques, and of the categorization of the XAI approaches and metrics, providing a comprehensive description of the entire ML pipeline and XAI techniques. We validated XMPLO through a case study to predict the performance indicators of an automotive manufacturing company, in particular the prediction of specific attributes of a product line. The validation results showed that the ontology properly addressed its requirements, and achieved its goals. By leveraging this ontology, users can gain a holistic insight into the decision-making process of the ML model, fostering trust and transparency.

This paper is structured as follows: Section 2 provides background information on XAI. Section 3 discusses related work. Section 4 presents the ML pipeline architecture, the ontology specification and the conceptual model as a reference ontology. Section 5 describes the implementation of the operational ontology, and the metadata attributes. Section 6 describes the validation of the ontology, and finally, Section 7 discusses the contributions, limitations, and future work.

2. Background

Explainable Artificial Intelligence methods can be categorized into three main categories based on their scope, implementation, and forms of explanations [2]. The scope of explainability is either global or local. Global explanations encompass the whole model by showing which features affect the model output the most, whereas local explanations only apply to specific instances [2]. Explanations in XAI can be achieved through ante-hoc or post-hoc approaches. The ante-hoc approach consists of constructing an intrinsic explainable model before the ML training process, facilitating an understanding of outcomes through semantic resources [2,15]. While ante-hoc solutions ensure model transparency, they may not detail every step leading to the outcome, and are model-specific, as explainability is integrated into the model architecture [2,12]. In contrast, the post-hoc approach applies semantic resources to predictions generated from black-box ML models after the training process [13]. Post-hoc solutions offer the advantage of separating explanations from the ML model, making them model-agnostic and applicable to various ML models [2]. Some popular model-agnostic models are Local Interpretable Model-Agnostic Explanations (LIME), and SHapley Additive exPlanations (SHAP). LIME provides explanations by manipulating the input data of the model, constructing a surrogate model, and analyzing changes in predictions to identify the most influential features [2]. In contrast, SHAP uses an approach based on game theory for explaining the output of a ML model by assigning a weight, known as the Shapley value, to each feature of a trained model [2].

Forms of explanations in XAI typically vary based on user needs and concerns [2]. However, four common forms of explanation are numerical, visual, rule-based, and textual explanations [3]. Numerical explanations consist of values, vectors, or matrices elucidating the model's outcome, visual explanations are graphical (e.g., heatmaps) [2], textual explanations are often used for individual predictions (local scope), and rule-based explanations employ if-then rules or trees with and/or operators to explain model predictions (global scope) [3].

Understanding the ML pipeline can also give some insight into ML models and results, by showing its various influencing factors, including dataset preparation, preprocessing steps, variable configurations, and dataset split for training and testing, all of which influence the behavior and outcomes of ML models. The processes of a ML pipeline include several stages that are essential for developing and deploying a ML model. According to the CRISP-DM (Cross-Industry Standard for Data Mining) methodology [4], the ML pipeline undergoes six phases: (1) Business Understanding, (2) Data Understanding, (3) Data Preparation, (4) Modelling, (5) Evaluation, and (6) Deployment. Moreover, insights from the ML pipeline are crucial for users to comprehend how explanations were derived. This encompasses whether explanations are provided for specific instances (local explanations) or the entire dataset (global explanations), and whether they accurately reflect the rationale behind the ML model, thereby building trust among users.

3. Related Work

In a literature review, we identified some ontologies related to XMLPO that cover elements regarding Machine Learning algorithms. The OntoDM-core [16] ontology aims to provide semantic annotations for Data Mining (DM) algorithms and to describe the fundamental elements of data mining and their characteristics, such as dataset and datatype, data mining tasks, and data mining algorithms. The Expose [17] ontology focuses on ML experiments and provides support to the analysis of ML algorithms and the exchange of DM experiments and workflows. The primary emphasis of this ontology lies in applying supervised ML techniques on propositional datasets (i.e., datasets with true or false logical propositions that are done on a relational dataset).

ML-Schema [14] is a top-level ontology that provides a comprehensive set of classes, properties, and constraints for representing machine learning algorithms, including details about their inputs, outputs, primary steps, dependencies, implementations, and executions. ML-Schema is built upon existing ontologies like Expose and OntoDM-core, by providing a better coverage of provenance metadata for both data and ML models.

However, none of the existing ontologies fully covers all the aspects of a ML pipeline and explanation approaches, as these ontologies only cover the semantics of data mining algorithms, ML experiments, or ML workflows and algorithms. The XMLPO ontology aims to provide semantic representation also for data input, data preparation and pre-processing, ML approaches, and XAI techniques.

4. XMLPO Conceptual Model

This section describes the processes of an ML pipeline through an architecture defined with the ArchiMate language and introduces the ontology requirements along with the XMLPO modules.

4.1. ML pipeline

In this research, we developed XMLPO to represent an ML pipeline consisting of the following four phases (1) setting goals, (2) data preparation, (3) data pre-processing, and (4) training & testing of the ML model. Considering the CRISP-DM methodology [4], we combined data preparation and modeling into one phase, and we did not consider model deployment in the production environment. Our ML pipeline consists of these phases:

Figure 1 represents the main processes of the ML pipeline. In the first phase ('Setting Goals') the project goals are defined. The second phase ('Data Preparation') includes feature selection, the execution of queries for extracting data, data selection, and data merging (i.e., in case data is selected from different databases). The third phase ('Data Pre-processing') includes data cleaning by dropping rows with many missing values or by using imputation techniques to replace missing values, feature encoding (i.e, a technique used to transform the categorical features into numerical ones so that they can be used in regression models), and feature scaling (i.e., a technique used to normalize the range of the independent features).

The fourth phase ('Train & Test Split') includes the optimization of the used hyper-parameters (i.e., external configuration variables used in model training) in the ML models, K-fold cross-validation, training, and testing of the ML models. Hyper-parameter optimization refers to the process of selecting the optimal values for the hyper-parameters of a ML model to maximize its performance on a given dataset. K-fold cross-validation is used to assess predictive models, where the dataset is partitioned into k subsets or folds. The post-hoc XAI approach implies the application of XAI models on top of the black-box models that were generated from the third phase of the ML pipeline. The last step (process) includes the evaluation of the explainability by using XAI metrics.

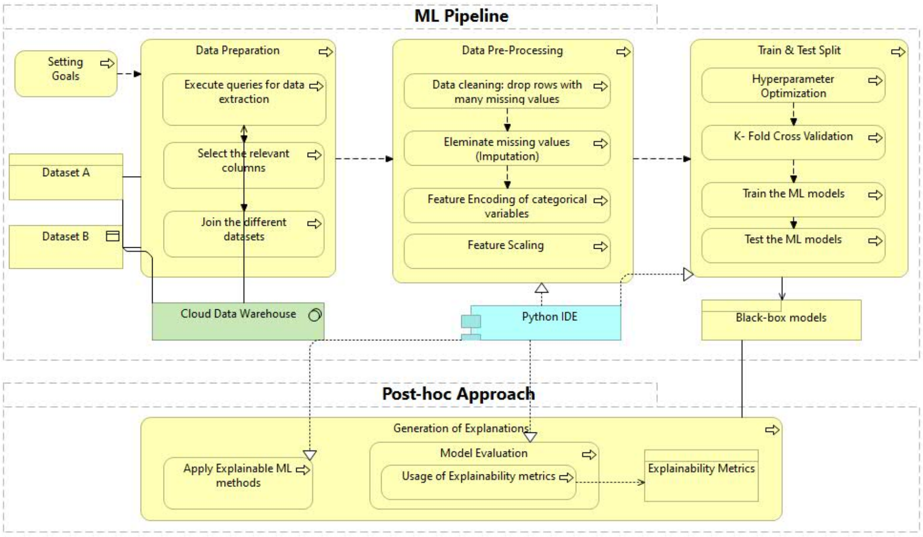

ML Pipeline
Setting Goals
Data Preparation
Execute queries for data extraction
Select the relevant columns
Join the different datasets
Dataset A
Dataset B
Cloud Data Warehouse
Data Pre-Processing
Data cleaning: drop rows with many missing values
Eleminate missing values (Imputation)
Feature Encoding of categorical variables
Feature Scaling
Python IDE
Train & Test Split
Hyperparameter Optimization
K- Fold Cross Validation
Train the ML models
Test the ML models
Black-box models
Post-hoc Approach
Generation of Explanations
Apply Explainable ML methods
Model Evaluation
Usage of Explainability metrics
Explainability Metrics

Figure 1. ML pipeline and post-hoc explanation

4.2. XMLPO Requirements

The Explainable ML Pipeline Ontology (XMLPO) has been designed by following the Systematic Approach for Building Ontology (SABiO) methodology [18], which consists of five phases: (1) purpose identification and requirements elicitation, (2) ontology capture and formalization, (3) design, (4) implementation, and (5) testing. The first phase of SABiO consists of identifying the purpose of the ontology and the ontology requirements, which are divided into non-functional requirements as qualitative aspects or characteristics of the ontology, and functional requirements, which are expressed as competency questions (CQ). The purpose of this ontology is to provide information and metadata on how the ML pipeline is built, including the input data, data preparation and pre-processing, ML models, ML model evaluations, XAI models, and XAI model evaluations. The intended users of the ontology are data scientists. **Table 1** shows the Competency Questions (CQ) of the XMLPO ontology which are categorized as data input, data pre-processing, ML model and ML model evaluation, and XAI model and XAI model evaluation. These competency questions are used to address the four issues that we identified with the Explainable ML workflows ontology [13] that we identified before.

4.3. XMLPO Modules

The second phase of SABiO ('Ontology Capture and Formalization Phase'), focuses on the development of the conceptual model, for example, by employing the OntoUML language for representing concepts based on the Unified Foundation Ontology (UFO). **Figure 2** shows that XMLPO is structured in four modules (General Module, Data Input and Pre-Processing, ML model, and the Explanation module), in accordance with the processes shown in **Figure 1**.

The General Module describes the main concepts that are needed to represent algorithms and experiments based on the ML-Schema vocabulary [14,13]. Every *Experiment*

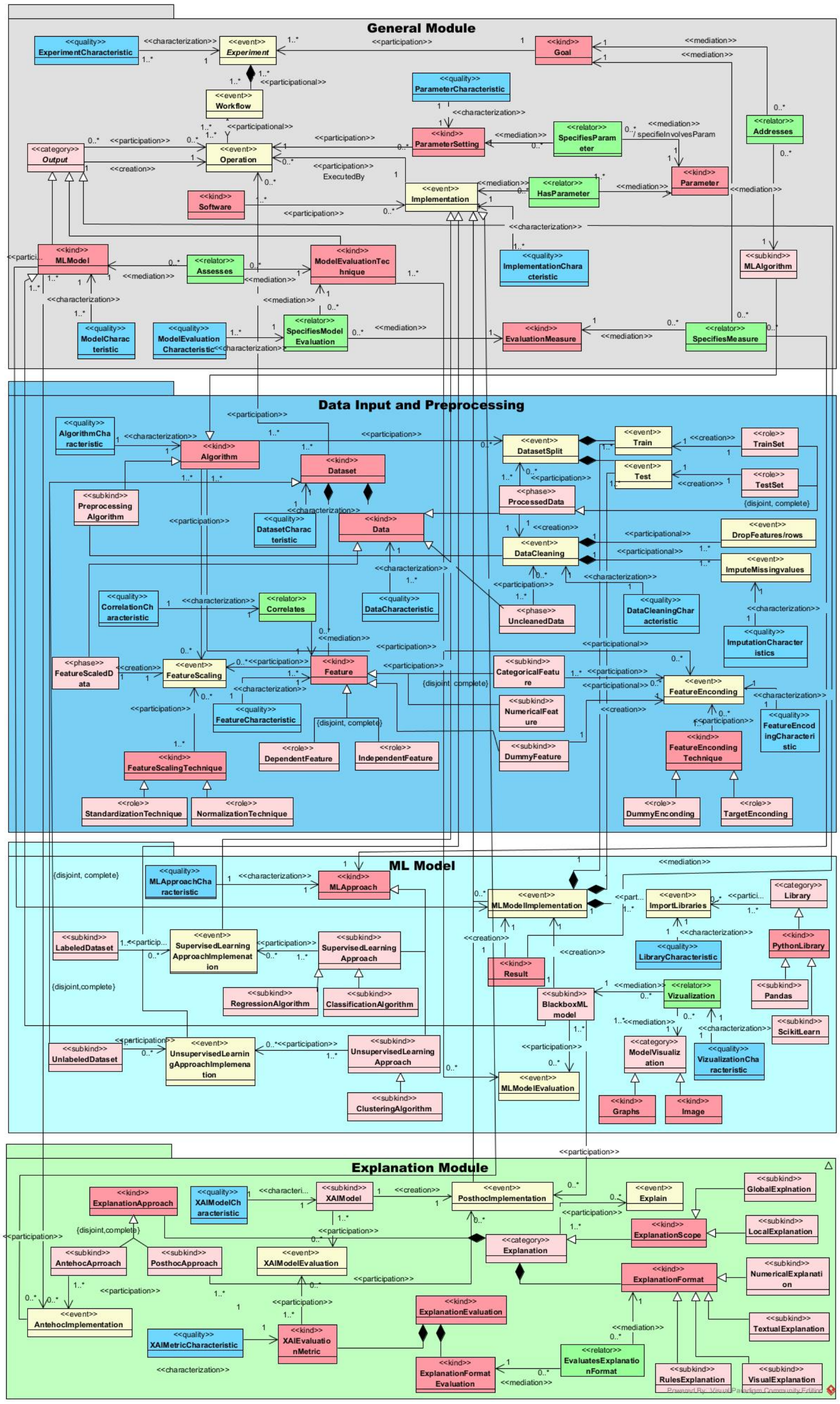

General Module
<<quality>> ExperimentCharacteristic
<<event>> Experiment
<<kind>> Goal
<<event>> Workflow
<<quality>> ParameterCharacteristic
<<relator>> Addresses
<<category>> Output
<<event>> Operation
<<kind>> ParameterSetting
<<relator>> SpecifiesParameter
/ specifieInvolvesParam
<<kind>> Parameter
<<kind>> Software
<<event>> Implementation
<<relator>> HasParameter
ExecutedBy
<<kind>> MLModel
<<relator>> Assesses
<<kind>> ModelEvaluationTechnique
<<quality>> ImplementationCharacteristic
<<subkind>> MLAlgorithm
<<quality>> ModelCharacteristic
<<quality>> ModelEvaluationCharacteristic
<<relator>> SpecifiesModelEvaluation
<<kind>> EvaluationMeasure
<<relator>> SpecifiesMeasure
Data Input and Preprocessing
<<quality>> AlgorithmCharacteristic
<<kind>> Algorithm
<<kind>> Dataset
<<event>> DatasetSplit
<<event>> Train
<<role>> TrainSet
<<event>> Test
<<role>> TestSet
{disjoint, complete}
<<subkind>> PreprocessingAlgorithm
<<quality>> DatasetCharacteristic
<<kind>> Data
<<phase>> ProcessedData
<<event>> DataCleaning
<<event>> DropFeatures/rows
<<event>> ImputeMissingvalues
<<quality>> CorrelationCharacteristic
<<relator>> Correlates
<<quality>> DataCharacteristic
<<phase>> UncleanedData
<<quality>> DataCleaningCharacteristic
<<quality>> ImputationCharacteristics
<<phase>> FeatureScaledData
<<event>> FeatureScaling
<<kind>> Feature
<<subkind>> CategoricalFeature
{disjoint, complete}
<<event>> FeatureEnconding
<<quality>> FeatureCharacteristic
<<subkind>> NumericalFeature
<<quality>> FeatureEncodingCharacteristic
{disjoint, complete}
<<subkind>> DummyFeature
<<kind>> FeatureEncondingTechnique
<<role>> DependentFeature
<<role>> IndependentFeature
<<kind>> FeatureScalingTechnique
<<role>> StandardizationTechnique
<<role>> NormalizationTechnique
<<role>> DummyEnconding
<<role>> TargetEnconding
ML Model
{disjoint, complete}
<<quality>> MLApproachCharacteristic
<<kind>> MLApproach
<<event>> MLModelImplementation
<<event>> ImportLibraries
<<category>> Library
<<subkind>> LabeledDataset
<<event>> SupervisedLearningApproachImplemenation
<<subkind>> SupervisedLearningApproach
<<kind>> Result
<<quality>> LibraryCharacteristic
<<kind>> PythonLibrary
{disjoint,complete}
<<subkind>> RegressionAlgorithm
<<subkind>> ClassificationAlgorithm
<<subkind>> BlackboxMLmodel
<<relator>> Vizualization
<<subkind>> Pandas
<<subkind>> ScikitLearn
<<subkind>> UnlabeledDataset
<<event>> UnsupervisedLearningApproachImplemenation
<<subkind>> UnsupervisedLearningApproach
<<category>> ModelVisualization
<<quality>> VizualizationCharacteristic
<<event>> MLModelEvaluation
<<subkind>> ClusteringAlgorithm
<<kind>> Graphs
<<kind>> Image
Explanation Module
<<kind>> ExplanationApproach
<<quality>> XAIModelCharacteristic
<<subkind>> XAIModel
<<event>> PosthocImplementation
<<event>> Explain
<<subkind>> GlobalExplnation
{disjoint,complete}
<<kind>> ExplanationScope
<<subkind>> LocalExplanation
<<category>> Explanation
<<subkind>> AntehocAprroach
<<subkind>> PosthocApproach
<<event>> XAIModelEvaluation
<<kind>> ExplanationFormat
<<subkind>> NumericalExplanation
<<event>> AntehocImplementation
<<kind>> ExplanationEvaluation
<<subkind>> TextualExplanation
<<quality>> XAIMetricCharacteristic
<<kind>> XAIEvaluationMetric
<<relator>> EvaluatesExplanationFormat
<<kind>> ExplanationFormatEvaluation
<<subkind>> RulesExplanation
<<subkind>> VisualExplanation
<<characterization>>
<<participation>>
<<participational>>
<<mediation>>
<<creation>>
Powered By: Visual Paradigm Community Edition

Figure 2. Conceptual Model of the XMLPO ontology

Table 1. Competency questions of the XMLPO ontology

- Data Input
  - CQ1: Which are the independent (input) features?
  - CQ2: Which are the dependent (output) features?
  - CQ3: Which are the categorical features?
  - CQ4: Which are the numerical features?
- Data Pre-processing
  - CQ5: How many values are missing?
  - CQ6: How is data cleaning done?
  - CQ7: How is the train and/or test split done?
  - CQ8: What feature scaling and/or feature encoding is done?
- ML model and evaluation
  - CQ9: Which libraries are being used (for a specific model)?
  - CQ10: Which ML model is used?
  - CQ11: What visualizations are performed?
  - CQ12: What are the parameters of a model?
  - CQ13: How is the ML model evaluated?
- XAI model and model evaluation
  - CQ14: Which XAI model is used?
  - CQ15: If a local explanation is applied, for which instance is the model tested, and what is the predicted value?
  - CQ16: Which features have a positive or negative impact?
  - CQ17: Which XAI metrics are used for evaluating the XAI model?

has some characteristics and is used to realize a specific goal (e.g., prediction of target feature). The *Goal* is addressed by the *MLAlgorithm* (e.g., Regression models are used for predicting target features). An *Operation* is an activity (also known as *Task* in ML-Schema) that refers to the execution of an *Implementation*. There are different kinds of implementations, such as the supervised ML model approach implementation, unsupervised ML model approach implementation, ML model implementation, etc. An *operation* generates an *output* which could be a *ML Model*, a *Model Evaluation Technique*, or a *Result* from the implementation of the ML model.

According to the Expose ontology [17], model evaluation techniques are classified into categories such as *Predictive Model Evaluation Measures* and *ClusteringEvaluation Measures*, which are related to the used ML model (as shown in **Figure 3**). *Predictive Model Evaluation Measures* include numerical prediction, graphical, or class prediction evaluation measures. Some *Numerical Prediction Evaluations* are *Error-Based Evaluations* (for example, Mean Absolute Error, Mean Squared Error, Root Mean Squared Error), *Information Criteria*, and *Correlation Coefficients*.

The Data Input and Pre-Processing module describes data and its features, data cleaning, feature encoding techniques, etc. A dataset comprises data that undergoes various phases, including *Uncleaned Data* and *Processed Data*, where the latter represents the data on which data cleaning procedures have been performed. *Data cleaning* is defined as an event as it involves actions such as removing features or rows with substantial missing values or employing imputation techniques to address missing feature values. The *Dataset Split* is defined as an event as it occurs within the *Processed Data* phase and

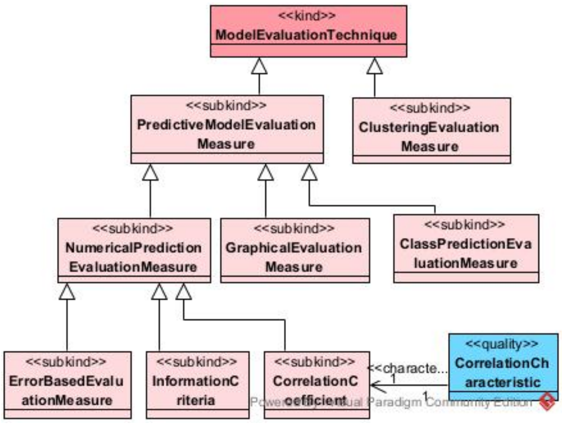

<<kind>>
ModelEvaluationTechnique
<<subkind>>
PredictiveModelEvaluation Measure
<<subkind>>
ClusteringEvaluation Measure
<<subkind>>
NumericalPrediction EvaluationMeasure
<<subkind>>
GraphicalEvaluation Measure
<<subkind>>
ClassPredictionEva luationMeasure
<<subkind>>
ErrorBasedEvalu ationMeasure
<<subkind>>
InformationC riteria
<<subkind>>
CorrelationC oefficient
<<characte...
1
1
<<quality>>
CorrelationCh aracteristic
Powered By Visual Paradigm Community Edition

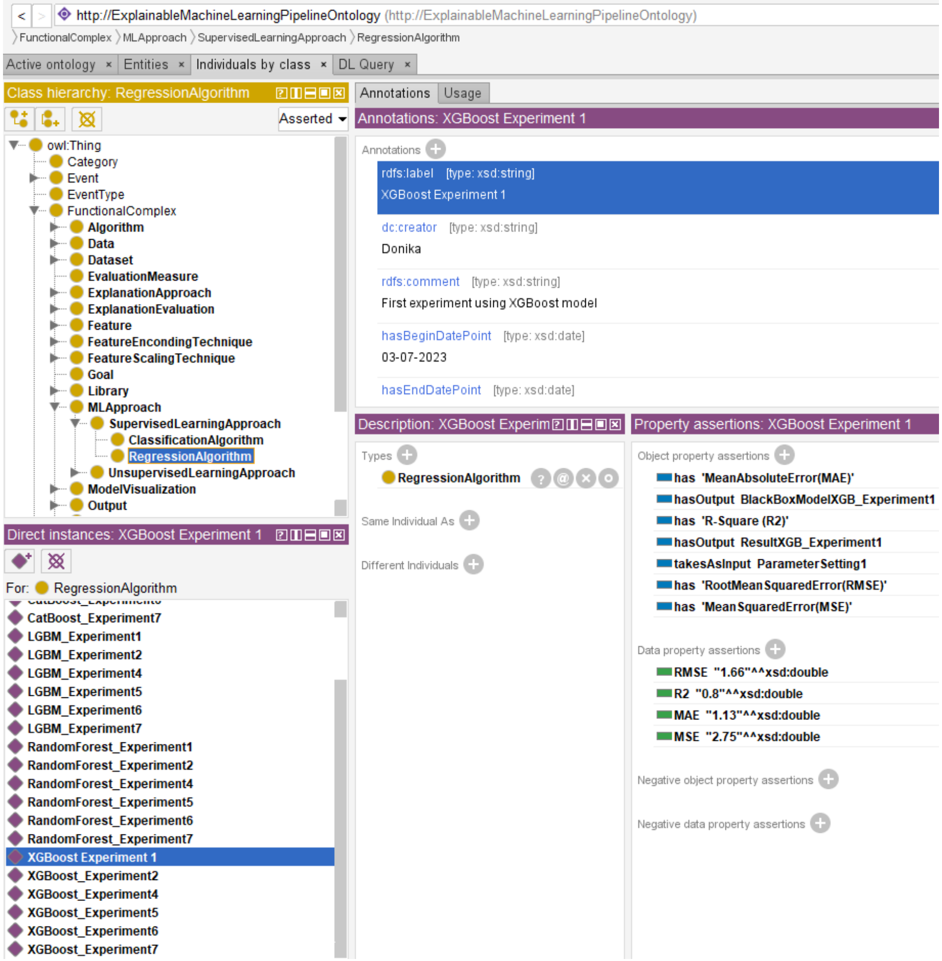
Figure 3. Conceptual Model on Model Evaluation Techniques

includes the split of the dataset into a training set and a test set. These sets are specifically used for training and testing ML models.

A dataset encompasses features categorized as either *Categorical Features*, with string data types, or numerical features, taking on integer or double data types. In experiments using regression models, where all features must be numerical, data scientists employ feature encoding methods like target encoding or dummy encoding to convert categorical values into numerical ones. Depending on the experiment, a feature may function as an independent (input) or dependent (output) variable, the latter being influenced by the former. Typically, data scientists apply feature scaling techniques to the independent features to restrict their values within a consistent range, and harmonize their influence on the results. Commonly used techniques for feature scaling are standardization and normalization methods.

The ML Model module describes the main characteristics of implementing the ML model, which takes *Processed Data* as input. *ML Approaches* are categorized as a *supervised Learning Approach* (e.g., regression or classification algorithms) which uses a labeled dataset, and an *Unsupervised Learning Approach* (e.g., clustering algorithm) which uses an unlabeled dataset. Multiple code libraries (e.g., Python libraries), are used during the implementation of the ML model such as the panda library used for working with the dataset, and the sci-kit learn library used for implementing the ML models. The output of an ML model implementation like the Cat boost model implementation is a *Black-box ML model*, which can be evaluated by using *ML model Evaluation Techniques*.

The Explanation Module describes the explanation techniques that can be used. The explanation is composed of an *Explanation Approach*, which is categorized as *Ante-hoc Approach* and *Post-hoc Approach*, and *Explanation Format*, which is categorized as numerical, textual, visual, and rules explanation. The *Post-hoc Implementation* takes as input the ML black box model from the previous module and gives as output the XAI model (e.g., the SHapley Additive exPlanations model), which is evaluated by using *XAI Evaluation Metrics*. The *Explanation Scope* refers to the scope of the generated explanation and is categorized as *Local Scope* (explanation for a specific instance of the model) or *Global Scope* (explanation at the model level).


< >
http://ExplainableMachineLearningPipelineOntology (http://ExplainableMachineLearningPipelineOntology)
FunctionalComplex 〉MLApproach 〉SupervisedLearningApproach 〉RegressionAlgorithm
Active ontology ×
Entities ×
Individuals by class ×
DL Query ×
Class hierarchy: RegressionAlgorithm
Asserted
Annotations
Usage
Direct instances: XGBoost Experiment 1
For: RegressionAlgorithm
Description: XGBoost Experim
Property assertions: XGBoost Experiment 1
Types
Same Individual As
Different Individuals
Object property assertions
Data property assertions
Negative object property assertions
Negative data property assertions

owl:Thing
- Category
- Event
- EventType
- FunctionalComplex
  - Algorithm
  - Data
  - Dataset
  - EvaluationMeasure
  - ExplanationApproach
  - ExplanationEvaluation
  - Feature
  - FeatureEnconding Technique
  - FeatureScalingTechnique
  - Goal
  - Library
  - MLApproach
    - SupervisedLearningApproach
      - ClassificationAlgorithm
      - RegressionAlgorithm
    - UnsupervisedLearningApproach
  - ModelVisualization
  - Output

CatBoost_Experiment7
LGBM_Experiment1
LGBM_Experiment2
LGBM_Experiment4
LGBM_Experiment5
LGBM_Experiment6
LGBM_Experiment7
RandomForest_Experiment1
RandomForest_Experiment2
RandomForest_Experiment4
RandomForest_Experiment5
RandomForest_Experiment6
RandomForest_Experiment7
XGBoost Experiment 1
XGBoost_Experiment2
XGBoost_Experiment4
XGBoost_Experiment5
XGBoost_Experiment6
XGBoost_Experiment7

Annotations: XGBoost Experiment 1

rdfs:label [type: xsd:string]
XGBoost Experiment 1

dc:creator [type: xsd:string]
Donika

rdfs:comment [type: xsd:string]
First experiment using XGBoost model

hasBeginDatePoint [type: xsd:date]
03-07-2023

hasEndDatePoint [type: xsd:date]

Types: RegressionAlgorithm

Object property assertions:
- has 'MeanAbsoluteError(MAE)'
- hasOutput BlackBoxModelXGB_Experiment1
- has 'R-Square (R2)'
- hasOutput ResultXGB_Experiment1
- takesAsInput ParameterSetting1
- has 'RootMeanSquaredError(RMSE)'
- has 'MeanSquaredError(MSE)'

Data property assertions:
- RMSE "1.66"^^xsd:double
- R2 "0.8"^^xsd:double
- MAE "1.13"^^xsd:double
- MSE "2.75"^^xsd:double

Figure 4. Individuals of the 'Regression Algorithm' class in XMPLO in Protégé

5. XMLPO Implementation and Metadata

5.1. Operational Ontology Implementation

The aim of the Ontology Design (phase 3 of SABiO) and Ontology Implementation (phase 4 of SABiO) is to generate an operational version of the ontology. **Figure 2** shows the XMLPO conceptual model obtained after performing the two first steps of SABiO. During the Ontology Design phase, we specified the technical details of the ontology and its implementation environment using OWL.

The conceptual models shown in **Figure 2** and in **Figure 3** were designed using OntoUML. These models were exported hen to gUFO (a lightweight implementation of UFO in OWL) by using the OntoUML plugin, and imported in the Protégé ontology development tool (in OWL). This plugin offers an automated transformation that maps OntoUML kind and subkind elements onto OWL classes and subclasses respectively, OntoUML associations onto OWL object properties, and quality elements onto OWL quality characteristics.

The XMLPO ontology was implemented in a manufacturing company whose goal was to understand the behavior of the ML models and its generated output. Therefore, we illustrated the usage of XMLPO here with a case study that aimed to predict specific performance indicators, in particular, to predict specific attributes of a product line. First, we conducted some ML experiments by using the CatBoost model, Extreme Gradient Boosting (XGBoost), Random Forest, and Light Gradient Boosting Model (LGBM)) in Jupyter Notebook in Python language, and applied XAI models such as SHAP on the results of the black-box models.

After that, we defined instances (individuals in Protégé) of the XMLPO ontology concepts to represent the steps taken and the outputs from the processes of the ML pipeline and explanation approach. **Figure 4** shows the individuals of the *RegressionAlgorithm* class, which is a subclass of the *SupervisedLearningApproach* class. For each of the implemented regression algorithms, we checked the evaluation scores for these evaluation methods, namely Root Mean Square Error (RMSE), R-squared (R^2), Mean Absolute Error (MAE) and Mean Squared Error (MSE). Lower values of RMSE, MAE, and MSE indicate better model performance, while values closer to one of R^2 indicate that a larger proportion of the variance in the dependent (output) feature is explained by the model. **Figure 4** shows the evaluation scores of the XGBoost_Experiment1 (RMSE=1.66; R^2=0.8; MAE=1.13 and MSE=2.75). These values indicate that the model performs reasonably well, with relatively low error metrics (RMSE, MAE, and MSE) and a high R^2 value, which indicates that 80% of the variance in the dependent feature is explained by the model, indicating a good fit.

Figure 5 shows the individuals of the '*Characteristics of XAI model1*' class. In this model, we used the SHAP values for identifying the feature importance (i.e., which are the features that had a positive or negative impact on the target feature). The features with positive impact contribute positively to the prediction (i.e., if the value of these features increases then also the predicted feature will have a higher value) while the features with negative contribute negatively to the prediction (i.e., if the value of these features increases then the predicted feature will have a lower value).

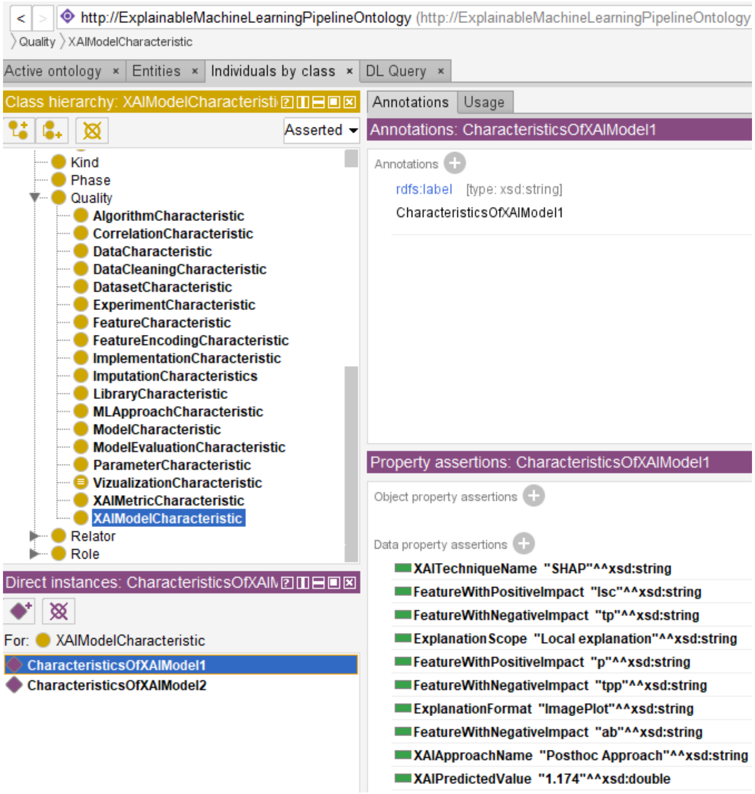
5.2. XMLPO Metadata Attributes

In the XMLPO ontology, we defined many metadata attributes for providing semantic information about the elements (i.e., concrete individuals such as kind, and subkind), which are represented as quality attributes in the conceptual model of the ontology (see **Figure 2**).

Firstly, metadata attributes have been defined for all the individuals by complying with *rdfs*[2] schema, and *Dublin Core*[3] metadata schema. For example, every concrete individual has a *rdfs:comment* attribute, which is used to describe the attribute, and a *rdfs:label* attribute, which is its human-readable name. All the event elements have these metadata attributes: 1) *dc: creator*, which provides information about the name of the creator, 2) a begin date point (*hasBeginDatePoint*), which refers to the start date of the

[2] https://www.w3.org/TR/rdf12-schema/#ch_classes.

[3] https://www.w3.org/wiki/DublinCore.


http://ExplainableMachineLearningPipelineOntology (http://ExplainableMachineLearningPipelineOntology
Quality 〉 XAIModelCharacteristic
Active ontology
Entities
Individuals by class
DL Query
Class hierarchy: XAIModelCharacteristi
Asserted
Kind
Phase
Quality
AlgorithmCharacteristic
CorrelationCharacteristic
DataCharacteristic
DataCleaningCharacteristic
DatasetCharacteristic
ExperimentCharacteristic
FeatureCharacteristic
FeatureEncodingCharacteristic
ImplementationCharacteristic
ImputationCharacteristics
LibraryCharacteristic
MLApproachCharacteristic
ModelCharacteristic
ModelEvaluationCharacteristic
ParameterCharacteristic
VizualizationCharacteristic
XAIMetricCharacteristic
XAIModelCharacteristic
Relator
Role
Direct instances: CharacteristicsOfXAIM
For: XAIModelCharacteristic
CharacteristicsOfXAIModel1
CharacteristicsOfXAIModel2
Annotations
Usage

Annotations: CharacteristicsOfXAIModel1

Annotations

rdfs:label [type: xsd:string]
CharacteristicsOfXAIModel1

Property assertions: CharacteristicsOfXAIModel1

Object property assertions

Data property assertions

XAITechniqueName "SHAP"^^xsd:string
FeatureWithPositiveImpact "lsc"^^xsd:string
FeatureWithNegativeImpact "tp"^^xsd:string
ExplanationScope "Local explanation"^^xsd:string
FeatureWithPositiveImpact "p"^^xsd:string
FeatureWithNegativeImpact "tpp"^^xsd:string
ExplanationFormat "ImagePlot"^^xsd:string
FeatureWithNegativeImpact "ab"^^xsd:string
XAIApproachName "Posthoc Approach"^^xsd:string
XAIPredictedValue "1.174"^^xsd:double

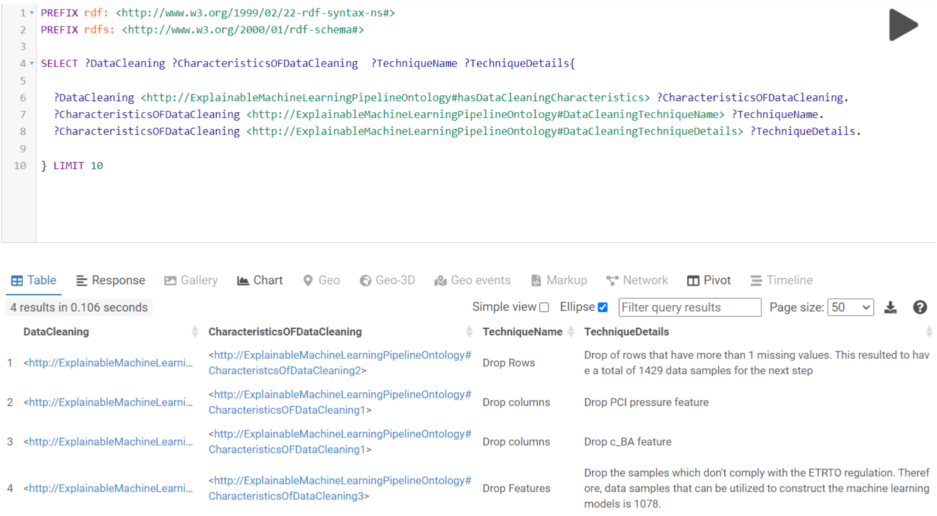
Figure 5. Individuals of the 'Characteristics of XAI model' class in the XMPLO

event, and 3) an end date point (*hasEndDatePoint*), which refers to the end date of the event, as shown in the Annotations tab in **Figure 4**.

Secondly, for each individual defined in Protégé, we provided information on its name, description, and technique name (e.g., ML technique name, data cleaning technique name, etc.). Besides the dataset name, the metadata about the *Dataset Characteristic* is enriched with the download description, which refers to the URL of the website where the dataset was found or the description of the database where it was found), access time, which refers to the date of retrieval, number of features and instance in the dataset.

6. Validation

XMLPO was validated by checking whether it fulfills the functional requirements shown in **Table 1**. To validate whether the XMLPO ontology can answer the competency questions of **Table 1** we tested the ontology with SPARQL. For example, to answer CQ6, which refers to how data cleaning is done, we wrote the SPARQL code shown in **Figure 6**. This query shows that the data cleaning consisted of dropping some columns and rows that had many missing values and dropping the samples that do not comply with

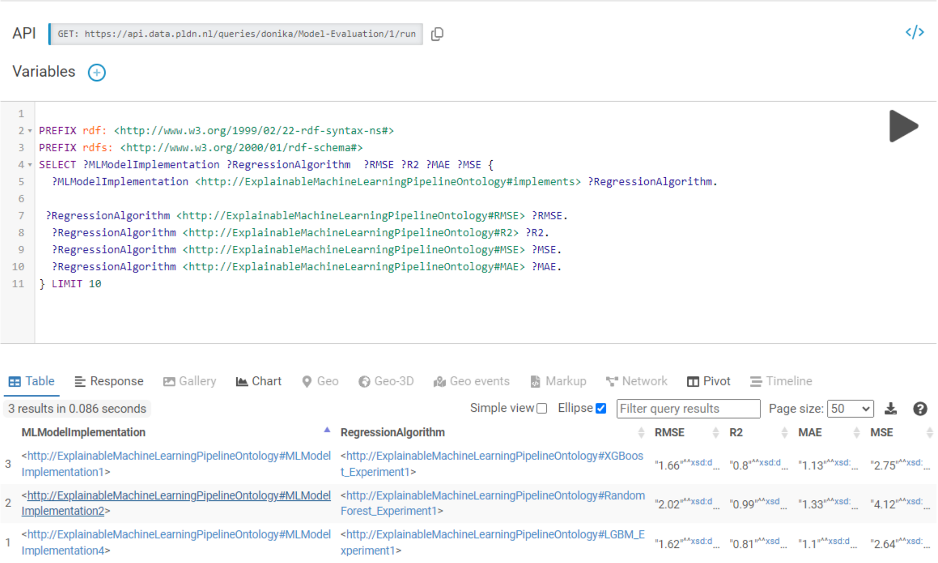


```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?DataCleaning ?CharacteristicsOFDataCleaning  ?TechniqueName ?TechniqueDetails{

  ?DataCleaning <http://ExplainableMachineLearningPipelineOntology#hasDataCleaningCharacteristics> ?CharacteristicsOFDataCleaning.
  ?CharacteristicsOFDataCleaning <http://ExplainableMachineLearningPipelineOntology#DataCleaningTechniqueName> ?TechniqueName.
  ?CharacteristicsOFDataCleaning <http://ExplainableMachineLearningPipelineOntology#DataCleaningTechniqueDetails> ?TechniqueDetails.

} LIMIT 10
```

4 results in 0.106 seconds

	DataCleaning	CharacteristicsOFDataCleaning	TechniqueName	TechniqueDetails
1	<http://ExplainableMachineLearni...	<http://ExplainableMachineLearningPipelineOntology#CharacteristcsOfDataCleaning2>	Drop Rows	Drop of rows that have more than 1 missing values. This resulted to have a total of 1429 data samples for the next step
2	<http://ExplainableMachineLearni...	<http://ExplainableMachineLearningPipelineOntology#CharacteristicsOFDataCleaning1>	Drop columns	Drop PCI pressure feature
3	<http://ExplainableMachineLearni...	<http://ExplainableMachineLearningPipelineOntology#CharacteristicsOFDataCleaning1>	Drop columns	Drop c_BA feature
4	<http://ExplainableMachineLearni...	<http://ExplainableMachineLearningPipelineOntology#CharacteristicsOfDataCleaning3>	Drop Features	Drop the samples which don't comply with the ETRTO regulation. Therefore, data samples that can be utilized to construct the machine learning models is 1078.

Figure 6. SPARQL results for obtaining the characteristics of data cleaning



```
GET: https://api.data.pldn.nl/queries/donika/Model-Evaluation/1/run
```

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?MLModelImplementation ?RegressionAlgorithm  ?RMSE ?R2 ?MAE ?MSE {
  ?MLModelImplementation <http://ExplainableMachineLearningPipelineOntology#implements> ?RegressionAlgorithm.

 ?RegressionAlgorithm <http://ExplainableMachineLearningPipelineOntology#RMSE> ?RMSE.
  ?RegressionAlgorithm <http://ExplainableMachineLearningPipelineOntology#R2> ?R2.
  ?RegressionAlgorithm <http://ExplainableMachineLearningPipelineOntology#MSE> ?MSE.
  ?RegressionAlgorithm <http://ExplainableMachineLearningPipelineOntology#MAE> ?MAE.
} LIMIT 10
```

3 results in 0.086 seconds

	MLModelImplementation	RegressionAlgorithm	RMSE	R2	MAE	MSE
3	<http://ExplainableMachineLearningPipelineOntology#MLModelImplementation1>	<http://ExplainableMachineLearningPipelineOntology#XGBoost_Experiment1>	"1.66"^^xsd:d...	"0.8"^^xsd:d...	"1.13"^^xsd:...	"2.75"^^xsd:...
2	<http://ExplainableMachineLearningPipelineOntology#MLModelImplementation2>	<http://ExplainableMachineLearningPipelineOntology#RandomForest_Experiment1>	"2.02"^^xsd:d...	"0.99"^^xsd...	"1.33"^^xsd:...	"4.12"^^xsd:...
1	<http://ExplainableMachineLearningPipelineOntology#MLModelImplementation4>	<http://ExplainableMachineLearningPipelineOntology#LGBM_Experiment1>	"1.62"^^xsd:d...	"0.81"^^xsd...	"1.1"^^xsd:d...	"2.64"^^xsd:...

Figure 7. SPARQL results for obtaining the evaluation scores

the ETRTO regulation (i.e., a regulation used in manufacturing). As a result, the number of data samples that were utilized to construct the ML models in this case study is 1078.

To give another example, for answering CQ13, which refers to how the ML model is evaluated, we formulated the SPARQL code shown in **Figure 7**. This query shows that the implemented ML models were evaluated by using different evaluation measures such as Root Mean Square Error, R^2, Mean Absolute Error, and Mean Squared Error. **Figure 7** shows some of the evaluation scores of the models.

Moreover, XMLPO was validated in a case study in a manufacturing company. The goals of this case study were (1) to provide understandable insights into the processes of the ML pipeline so that the produced models can be reused and changed in the future if needed (i.e., it would be easy for other data scientists to understand how the current models were built by considering data input, pre-processing, and model training and testing), (2) to provide appropriate prediction models, and (3) to show the features that affected the predictions the most. To demonstrate this, we presented the results from the SPARQL queries to a group of stakeholders in the company. The stakeholders confirmed that this ontology gives a better understanding of the decision process within the ML pipeline, and that the top features identified with the SHAP model that affected the predictions the most align with their intuitive understanding of the domain. The second goal of providing good prediction models is fulfilled by checking the evaluation scores of the models (**Figure 7**), which indicate that the models perform well, with relatively low error metrics (RMSE, MAE, and MSE) and high R^2 values (close to 1).

7. Conclusion

In this paper, we introduce the Explainable ML Pipeline Ontology (XMLPO), which extends the Explainable ML workflows ontology [13]. In this research, we have improved the representation of a Machine Learning (ML) pipeline and explanation approaches, by addressing the four gaps that we identified regarding the information and provenance metadata about data input, and pre-processing, characteristics of ML approaches and categorization of XAI.

XMLPO is designed as an OntoUML conceptual model, and implemented as an OWL-based operational ontology based on gUFO. The ontology covers main metadata attributes about various aspects of data components in the ML pipeline. This operational ontology was implemented with the OntoUML plug-in and modified in Protégé by asserting object properties and data properties. We validated the operational ontology by analyzing how the competency questions are answered through a case study with a ML pipeline, which generates a model that predicts specific performance indicators in a manufacturing context.

Due to confidentiality constraints imposed by the company in which the case study was performed, we are not allowed to give more details about the case study. In addition, we were not able to check the evaluation of the XAI models by using XAI metrics (refer to CQ17), as in our case study we implemented the SHAP model on top of the ML models. The output generated from the SHAP model indicated the top features that have the strongest relation with the prediction of the performance indicators. However, we presented the outputs to the stakeholders and they confirmed that the top features align with their intuitive understanding (i.e., this refers to context explainability). In other cases, it might be necessary to evaluate the explanations by using metrics regarding the XAI evaluation properties.

As future work, this ontology can be extended to cover the XAI evaluation properties, by providing a more comprehensive evaluation of explainability. The Co-12 recipe [19] classifies these properties into 3 categories based on: (1) content (i.e., checking the correctness, completeness, consistency, continuity, and covariate complexity), (2) presentation form (i.e., checking the compactness, composition, and confidence), and (3) end-users (i.e, checking the context, controllability, and coherence). In addition, the reusability of this ontology can be assessed and possibly improved by applying it to other case studies with different scopes such as the usage of unsupervised ML approaches and/or ante-hoc approaches. The ontology can also be extended to also cover the concepts of deploying the ML models in a production environment, which is the last phase of the ML pipeline in the CRISM-DM methodology [4].

References

[1] Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. Science. 2015 Jul 17;349(6245):255-60.

[2] Viswan V, Shaffi N, Mahmud M, Subramanian K, Hajamohideen F. Explainable artificial intelligence in Alzheimer's disease classification: A systematic review. Cognitive Computation. 2024 Jan;16(1):1-44.

[3] Yang G, Ye Q, Xia J. Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond. Information Fusion. 2022 Jan 1;77:29-52.

[4] Schröer C, Kruse F, Gómez JM. A systematic literature review on applying CRISP-DM process model. Procedia Computer Science. 2021 Jan 1;181:526-34.

[5] Gonçalves RS, Musen MA. The variable quality of metadata about biological samples used in biomedical experiments. Scientific data. 2019 Feb 19;6(1):1-5.

[6] Fill HG, Cabot J, Maass W, Van Sinderen M. AI-Driven Software Engineering–The Role of Conceptual Modeling. Enterprise Modelling and Information Systems Architectures (EMISAJ). 2024 Jan 24;19.

[7] Seeliger A, Pfaff M, Krcmar H. Semantic web technologies for explainable machine learning models: A literature review. PROFILES/SEMEX@ ISWC. 2019 Oct 27;2465:1-6.

[8] Ribeiro MT, Singh S, Guestrin C. " Why should i trust you?" Explaining the predictions of any classifier. InProceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining 2016 Aug 13 (pp. 1135-1144).

[9] Gunning D, Aha D. DARPA's explainable artificial intelligence (XAI) program. AI magazine. 2019 Jun 24;40(2):44-58.

[10] Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access. 2018 Sep 16;6:52155.

[11] Guarino N, editor. Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy. IOS press; 1998.

[12] Longo L, Goebel R, Lecue F, Kieseberg P, Holzinger A. Explainable artificial intelligence: Concepts, applications, research challenges and visions. In International cross-domain conference for machine learning and knowledge extraction 2020 Aug 18 (pp. 1-16). Cham: Springer International Publishing.

[13] Nakagawa PI, Pires LF, Moreira JL, Bonino da Silva Santos LO, Bukhsh F. Semantic Description of Explainable Machine Learning Workflows for Improving Trust. Applied Sciences. 2021 Nov 16;11(22):10804.

[14] Publio GC, Esteves D, Ławrynowicz A, Panov P, Soldatova L, Soru T, Vanschoren J, Zafar H. ML-schema: exposing the semantics of machine learning with schemas and ontologies. arXiv preprint arXiv:1807.05351. 2018 Jul 14.

[15] Srihari SN. Explainable Artificial Intelligence. Journal of the Washington Academy of Sciences. 2020 Dec 1;106(4):9-38.

[16] Panov P, Soldatova L, Džeroski S. Ontology of core data mining entities. Data Mining and Knowledge Discovery. 2014 Sep;28:1222-65.

[17] Vanschoren J, Soldatova L. Exposé: An ontology for data mining experiments. InInternational workshop on third generation data mining: Towards service-oriented knowledge discovery (SoKD-2010) 2010 (pp. 31-46).

[18] de Almeida Falbo R. SABiO: Systematic Approach for Building Ontologies. Onto. Com/odise@ FOIS. 2014 Sep;1301.

[19] Nauta M, Seifert C. The co-12 recipe for evaluating interpretable part-prototype image classifiers. In-World Conference on Explainable Artificial Intelligence 2023 Jul 26 (pp. 397-420). Cham: Springer Nature Switzerland.