Enhancing Interpretability of Machine Learning Models over Knowledge Graphs

Yashrajsinh Chudasama[1,2,†], Disha Purohit[1,2], Philipp D. Rohde[1,2,3] and Maria-Esther Vidal[1,2,3]

[1]*Leibniz University Hannover, Germany*
[2]*TIB Leibniz Information Centre for Science and Technology, Hannover, Germany*
[3]*L3S Research Center, Hannover, Germanny*

Abstract

Artificial Intelligence (AI) plays a critical role in data-driven decision-making frameworks. However, the lack of transparency in some machine learning (ML) models hampers their trustworthiness, especially in domains like healthcare. This demonstration aims to showcase the potential of Semantic Web technologies in enhancing the interpretability of AI. By incorporating an interpretability layer, ML models can become more reliable, providing decision-makers with deeper insights into the model's decision-making process. InterpretME effectively documents the execution of an ML pipeline using factual statements within the InterpretME knowledge graph (KG). Consequently, crucial metadata such as hyperparameters, decision trees, and local ML interpretations are presented in both human- and machine-readable formats, facilitating symbolic reasoning on a model's outcomes. Following the Linked Data principles, InterpretME establishes connections between entities in the InterpretME KG and their counterparts in existing KGs, thus, enhancing contextual information of the InterpretME KG entities. A video demonstrating InterpretME is available online[1], and a Jupyter notebook[2] for a live demo is published in GitHub[3].

1. Introduction

The advent of machine learning (ML) has highlighted the importance of *interpretability* in comprehending the decisions made by computational frameworks. While these frameworks often provide highly accurate outcomes, understanding the reasoning behind their decisions can be challenging. Although interpretable tools like LIME [1] exist to interpret the predictions of ML models, they fall short in translating the captured knowledge, such as model insights, into the application domain. In contrast, knowledge graphs (KGs) represent real-world knowledge through

[1]https://www.youtube.com/watch?v=Bu4lROnY4xg
[2]https://mybinder.org/v2/gh/SDM-TIB/InterpretME_Demo/main?labpath=InterpretME_Demo.ipynb
[3]https://github.com/SDM-TIB/InterpretME_Demo

SEMANTICS 2023 EU: 19^{th} *International Conference on Semantic Systems, September 20–22, 2023, Leipzig, Germany*

[†] All authors contributed equally. This work has been supported by "Leibniz Best Minds: Programme for Women Professors", project TrustKG-Transforming Data in Trustable Insights with grant P99/2020 and Federal Ministry for Economic Affairs and Energy of Germany (BMWK) in the project CoyPu (project number 01MK21007[A-L]).

yashrajsinh.chudasama@tib.eu (Y. Chudasama); disha.purohit@tib.eu (D. Purohit); philipp.rohde@tib.eu (P. D. Rohde); maria.vidal@tib.eu (M. Vidal)
0000-0003-3422-366X (Y. Chudasama); 0000-0002-1442-335X (D. Purohit); 0000-0002-9835-4354 (P. D. Rohde); 0000-0003-1160-8727 (M. Vidal)

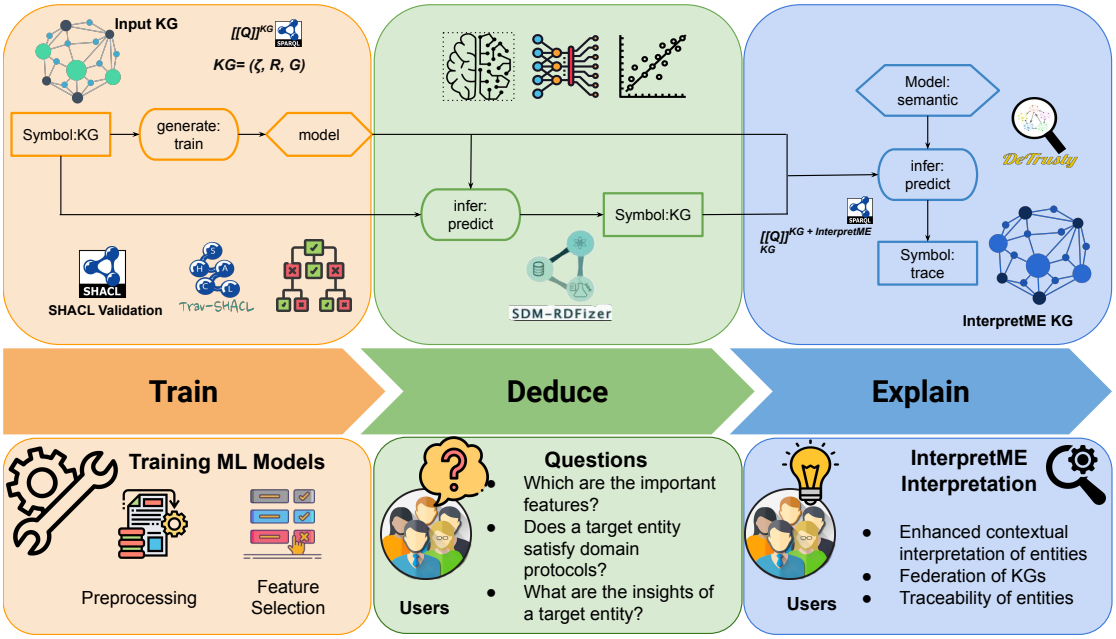

Figure 1: InterpretME comprises the *Train*, *Deduce* and *Explain* steps. *Train* resorts to supervised ML models, and AutoML, for model training and hyperparameter optimization. *Deduce* resorts to LIME and SHACL engines to explain feature importance and integrity constraints' satisfaction. An InterpretME KG comprises metadata traced during *Train* and *Deduce*. *Explain* provides enhanced statements of an ML model's outcomes. RML and Federated SPARQL engines create and explore the InterpretME KG.

domain ontologies, using entities (e.g., `dbr:Louis_XIV`) and relations (e.g., `dbo:spouse`). KGs have garnered significant attention for representing ML models (e.g., ML schema [2]) and enhancing our understanding of predictive model characteristics. In this demonstration, attendees will witness InterpretME's ability to interpret predictive model decisions based on the French Royalty KG e.g., feature selection, prediction probabilities, and SHACL validation reports. Entailment regimes of `owl:sameAs` will enable to deduce new insights. The French Royalty KG [3] is a fully curated KG representing factual statements about the French royal families; it includes class `dbo:Person` and its relationships, e.g., `dbo:spouse` and `dbo:child`.

2. InterpretME

The pipeline implemented by InterpretME, as illustrated in Figure 1, follows a hybrid design pattern [4] and consists of three layers: *Train*, *Deduce*, and *Explain*. InterpretME accepts input in the form of KGs or datasets (CSV or JSON format). The user configuration, provided in JSON format, specifies the independent variables – features selected by the user for analyzing ML model predictions, e.g., child – dependent variables (features that change as the independent variables vary, e.g., spouse), the path to a dataset or SPARQL endpoint, and a target class definition. Application data is retrieved from the input KGs using *SPARQL queries*. The *Train* component utilizes standard supervised ML models (e.g., decision trees) to train on the provided

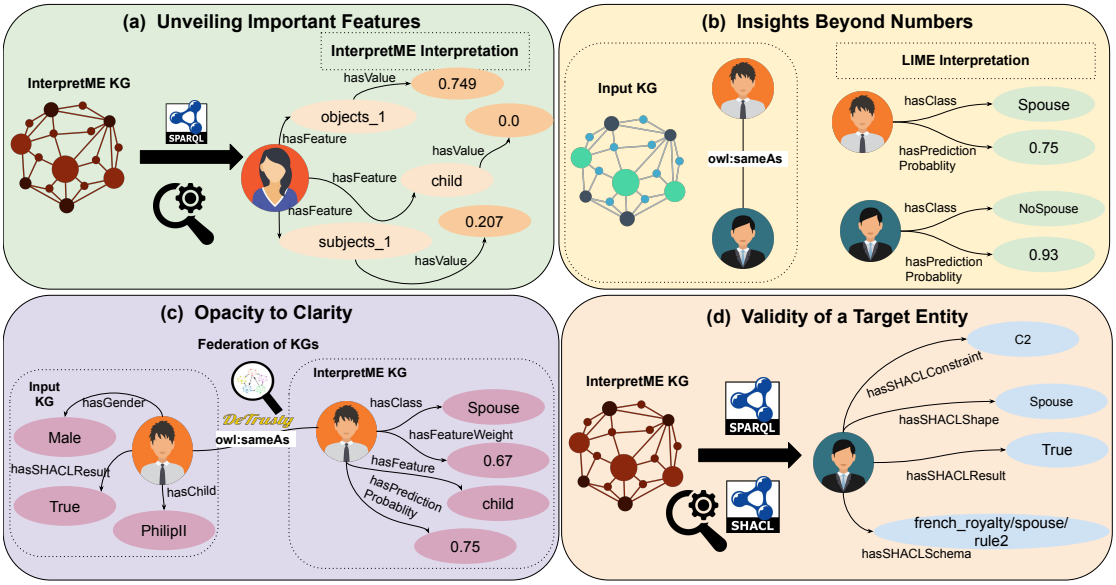

Figure 2: InterpretME Use Cases. (a) Illustrates the key features. (b) Interpretable tools like LIME lack the ability to consider semantics, such as `owl:sameAs`, when generating interpretations for individual entities. (c) Leveraging the federation of KGs enables the discovery of contextual insights about entities from both the input KG and the InterpretME KG. (d) SHACL validation is employed to validate the target entity, ensuring the quality of input data and enhancing the interpretability of ML models.

input data. The input data is preprocessed into the necessary format for training the predictive model, and AutoML [5] optimizes hyperparameters. The trained model performs a binary classification task, such as *"Predicting whether a French royal person has a spouse"*.

The *Deduce* component utilizes the predictions of the trained ML model to enhance interpretability. Interpretable tools like decision trees and LIME [1] are obtained to understand the predictive model's decisions. Here a symbolic reasoning system, i.e., SHACL validation is used to check whether a predicted entity satisfies domain protocols, e.g., *"If two persons are having the same child then they are married"*. SHACL validation generates a post-hoc justification for entities that violate the protocols while simultaneously ensuring the accuracy of the input RDF data.

InterpretME leverages RML mappings [6], as utilized by the SDM-RDFizer [7], to define the captured metadata from the trained predictive model, which is then incorporated into the InterpretME KG. These RML mappings provide a declarative specification of the classes and properties in the InterpretME ontology and ML schema, ensuring that the collected entities are accurately represented. Consequently, the InterpretME KG contains factual statements that are both human-readable and machine-readable, effectively documenting the behavior of the predictive models. The *Explain* module offers users more comprehensive interpretations of the predictive models' decisions. Users can perform statistical analysis on specific target entities using *SPARQL queries*, gaining deeper insights into the decision-making process of the ML models. Federated Query Processing is employed to query both the input KGs and the InterpretME KG. This module enables users to interpret the characteristics of a target entity within the predictive task and understand its context within the input KGs.

3. Demonstration of Use Cases

InterpretME is demonstrated over the French Royalty KG [3]. Attendees will execute ML pipelines on KGs and run *SPARQL queries* to retrieve traced statements. The use cases facilitate a comprehensive understanding of the characteristics of a target entity within the predictive model and its context within the input KGs. We will focus on a predictive task involving the French Royalty KG, where an ML model assigns a classification to a particular target entity, leaving the decision-making process puzzling. The following use cases will be demonstrated:

Unveiling Important Features. Feature significance is an important part of understanding the underlying mechanics and context in which ML models work. The predictive models utilize input features to detect patterns and representations of a target entity. Figure 2 illustrates an exemplar target entity represented in the InterpretME KG with all the ML model characteristics and reveals the most important feature with its contribution in the predictive task. Attendees will be able to study and discover which crucial features contribute the most to the decisions of the ML models. Thus, the contextual knowledge provided by the InterpretME KG assists attendees in understanding the features that influence the training of predictive models.

Insights Beyond Numbers. Aside from statistical and numerical data, another critical factor is the underlying logic of ML models. Quantitative metrics, e.g., accuracy and precision reflect the overall performance of the ML models, while interpretable tools, such as LIME, provide interpretations of a target entity without considering the semantic meaning of an entity. For instance, *"Is dbr:Louis_XIV (i.e., target entity) linked to another entity in the input KG?". SPARQL queries* over the InterpretME KG reveal that LIME creates interpretation for target entities, `dbr:Louis_XIV` and `dbr:Philip_III_of_Spain`, despite the fact that both entities represent the same concept in the input KG. Thus, InterpretME helps attendees in understanding the importance of considering the semantic properties of an entity within the ML model.

From Opacity to Clarity. Although interpretable tools like LIME offer interpretations, they are human-readable only and unclear for which entity the interpretation is generated. InterpretME overcomes this limitation and provides human- and machine-readable interpretations. Figure 2 illustrates the enhanced interpretation of the target entity, for instance, `dbr:Louis_XIV` with all the characteristics from the trained predictive model and the properties from the input KG. These entities are specified in the InterpretME KG in terms of metadata obtained by the *Train* and *Deduce* layers. InterpretME enhances the contextual description of a target entity by annotating the behavior of the person (e.g., the models' characteristics) in the predictive model, thus, providing more insights into the ML model's decision. InterpretME follows the FAIR principles and the vocabulary that describes the captured metadata of ML models' by InterpretME is publicly available as an instance of VoCoL[1]. InterpretME stores independent and dependent variables as metadata allowing the user to trace back the target entity defined in the input KGs. InterpretME resorts to the federated query engine DeTrusty [8] for KG traversing. Attendees will execute *SPARQL queries* to retrieve data from the input KG, the InterpretME KG, or both. Based on these findings, attendees will trace back the characteristics of a target entity.

Validity of a Target Entity. SHACL validation is essential for ensuring the quality of the input data and enhancing the interpretability of ML models. SHACL allows the user to define domain

[1] http://ontology.tib.eu/InterpretME/

protocols or rules for the characteristics of a target entity. InterpretME utilizes Trav-SHACL [9] to perform validation of a target entity of an ML model; validation reports are generated and illustrate whether the entity follows the defined protocols. For instance, in the French Royalty KG, to ensure the validity of entities, the protocol *"If a person has a father and a mother then the father has a spouse"* is defined. InterpretME captures the validation report (i.e., SHACL shapes, constraints, and validation results) of the entities in the input KG and aligns it with the ML model characteristics of a target entity in the InterpretME KG. Attendees can explore the validation results of a particular target entity using *SPARQL queries*, and check if the classification of ML models is based on entities that invalidate SHACL constraints. Thus, SHACL validation enhances the interpretability of a target entity, allowing to assess domain protocol adherence, interpret ML model predictions in the light of faithfulness, and deduce meaningful insights.

4. Conclusions

We illustrate how the ML models over KGs can be interpreted using InterpretME. InterpretME enhances the interpretation of a target entity by adding contextual knowledge collected from the trained predictive model. In our demonstration, we show how our approach can be applied to data-driven frameworks, which is a crucial trait in the Semantic Web community. Moreover, attendees will recognize the significance of capturing the predictive pipeline's metadata.

References

[1] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: ACM SIGKDD, 2016. doi:10.1145/2939672.2939778.

[2] D. Esteves, A. Ławrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren, ML Schema Core Specification, W3C Submission, 2016. URL: http://www.w3.org/2016/10/mls/.

[3] N. Halliwell, F. Gandon, F. Lecue, User Scored Evaluation of Non-Unique Explanations for Relational Graph Convolutional Network Link Prediction on Knowledge Graphs, in: K-CAP, ACM, 2021.

[4] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, A. ten Teije, Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases, Appl. Intell. (2021). doi:10.1007/s10489-021-02394-3.

[5] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: KDD, 2019. URL: https://doi.org/10.1145/3292500.3330701.

[6] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, R. Van de Walle, RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data, in: 7th Workshop on Linked Data on the Web, CEUR-WS.org, 2014.

[7] E. Iglesias, S. Jozashoori, D. Chaves-Fraga, D. Collarana, M.-E. Vidal, SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs, in: CIKM, 2020.

[8] P. D. Rohde, M. Bechara, Avellino, DeTrusty v0.12.3, 2023. doi:10.5281/zenodo.8095810.

[9] M. Figuera, P. D. Rohde, M.-E. Vidal, Trav-SHACL: Efficiently Validating Networks of SHACL Constraints, in: The Web Conference, ACM, 2021. doi:10.1145/3442381.3449877.