Towards Democratized Machine Learning: A Semantic Web Approach

Antonis Klironomos
antonis.klironomos@de.bosch.com
Bosch Center for AI, Renningen, Germany & University of Mannheim, Mannheim, Germany
Supervised by Mohamed H. Gad-Elrab, Evgeny Kharlamov, and Heiko Paulheim

Abstract

The rapid growth of machine learning (ML) research has produced a vast and expanding collection of algorithms, datasets, and pipelines available on the Web. However, fragmented and dispersed documentation of these resources hampers accessibility, transparency, and effective use, posing challenges for users seeking to understand, adapt, and create ML pipelines. To address these challenges, we leverage Knowledge Graphs (KGs) and ontologies to represent ML pipelines as executable KGs (ExeKGs). This approach fosters an intuitive understanding of pipeline components and their relationships while defined constraints streamline the creation of valid and efficient pipelines. Furthermore, the structure of our KGs enables intelligent exploration and discovery of relevant ML artifacts, including pipelines and datasets. By incorporating KG-based ML techniques, we enhance the discovery and reuse of these artifacts. To consolidate these functionalities and provide users with an intuitive interface, we are developing ExeKGLab, a GUI-based platform for interacting with ExeKGs. This thesis explores the potential of KGs to democratize the ML landscape. We present our ongoing efforts to build a KG for ML, emphasizing its role in simplifying pipeline design, enhancing comprehension, and enabling smart exploration. By creating a structured and interconnected framework, our approach seeks to bridge gaps in accessibility and foster a more collaborative ML ecosystem. We invite discussion and feedback to advance this promising direction for future ML research.

CCS Concepts

• **Computing methodologies** → **Knowledge representation and reasoning**; **Machine learning**; • **Information systems**;

Keywords

Knowledge graphs, Machine learning, Semantics, Similarity

ACM Reference Format:
Antonis Klironomos. 2025. Towards Democratized Machine Learning: A Semantic Web Approach. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3701716.3715280

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3715280

1 Introduction

Machine learning (ML) is indispensable in various domains, driving advancements in healthcare, manufacturing, and finance. A vast and growing collection of ML algorithms forms the foundation for a multitude of pipelines designed to address diverse tasks.

However, descriptions and implementations of these pipelines are often scattered across the Web, residing mainly in research papers and code repositories. This fragmentation, coupled with the increasing popularity and complexity of ML, poses significant challenges to accessibility and transparency. Researchers and practitioners alike struggle to efficiently locate, understand, and adapt existing pipelines to their specific needs. This challenge extends to domain experts, a trend we also observe amongst those at Bosch.

We leverage Knowledge Graphs (KGs) to address these challenges. We utilize KGs to represent ML pipelines in a standardized format, facilitating sharing and interoperability. This structured representation, coupled with the inherent visualizability of KGs, enhances the understandability of complex pipelines and empowers users with varying levels of expertise to create and modify them. Also, applying KG-based ML techniques on top of our executable KGs (ExeKGs) enables intelligent exploration of ML artifacts.

The PhD thesis and this paper explore the use of Semantic Web technologies, particularly KGs and ontologies, to democratize access to and creation of ML pipelines. We present our ongoing work on ExeKGLib, a Python library for representing, creating, and exploring ML pipelines represented as ExeKGs; ExeKGLab, a user-friendly GUI-based application that utilizes ExeKGLib to enable users from diverse backgrounds to engage with ML; and ML artifact recommendation by learning from a KG consisting of ExeKGs.

2 Problem Statement

In this thesis, we examine the democratization of ML by invistigating the following research questions:

- **RQ1**: How can KGs and Semantic Web technologies be leveraged to create a standardized representation of ML pipelines?
- **RQ2**: How can ontologies and KGs aid the creation and configuration of ML pipelines for users with varying levels of expertise?
- **RQ3**: How can KG-based ML techniques enhance the exploration and discovery of ML datasets and pipelines?

3 Related Work

Semantic Web for ML. Ontologies have been developed to describe datasets (*e.g.*, DCAT [11]), ML models (*e.g.*, ML-Schema [13]), and ML experiments (*e.g.*, EXO). However, these efforts often focus on specific aspects of the ML lifecycle and lack a holistic representation of pipelines, including data flow and high-level data science concepts. Some works have utilized Semantic Web technologies to represent or annotate ML workflows. RapidMiner [9], for instance, uses semantic annotations for pipeline validation. However, these approaches often fall short of capturing the full complexity and expressiveness of ML pipelines.

ML Pipeline Representation and Management. AutoML platforms like Auto-sklearn and TPOT focus on pipeline optimization but often lack explicit knowledge representation for explainability and reusability. Some AutoML works, such as KGpip [8] and DORIAN [14], utilize graphs to represent pipelines, but these representations are often custom-designed and lack adherence to common standards. ML democratization applications like KNIME [2] and RapidMiner [12] frequently rely on formats like YAML and JSON for ML pipeline representation. These formats lack the rich semantic descriptions offered by KGs, limiting interoperability and hindering advanced reasoning capabilities over ML pipelines.

GUI-based Tools for ML. GUI-based ML applications have evolved to democratize machine learning. Early IML tools like Crayons and ReGroup focused on interactive data labeling and model correction [6]. Platforms like Alpine Meadow streamlined pipeline creation for domain experts [16]. Research also highlighted the need for novice-friendly programming tools with integrated debugging [5]. This led to domain-specific applications like ilastik for bio-image analysis [1], Wekinator for music performance [7], and Apolo for network data exploration [4]. General-purpose platforms like KNIME, Google AutoML, Azure ML Studio, and RapidMiner offer simplified workflow creation through drag-and-drop interfaces and automation, broadening access to ML. However, these tools often lack representations of pipeline semantics, limiting their ability to automatically validate pipelines and generate tailored UIs.

KG Embeddings. KG embeddings play a crucial role in enabling effective reasoning and search over large KGs. By representing entities and relationships within our ML pipeline KG as dense, low-dimensional vectors, we can leverage semantic similarity measures to identify related pipelines and datasets. While various KG embedding techniques exist, including complex deep learning models like graph neural networks (GNNs) and triple-based methods like TransE [3], these can struggle with the scale and heterogeneity of our ML pipeline KG. Therefore, we anticipate exploring more scalable and efficient approaches like walk-based methods such as RDF2Vec [15], which are better suited for handling the complexity and potential sparsity of our graph data. These embeddings will be instrumental in enabling intelligent exploration and recommendation of ML pipelines within ExeKGLab.

Gaps and Opportunities. While Semantic Web technologies have been employed for representing and validating ML pipelines, their use for creating, executing, and learning from them is limited. Existing AI democratization platforms often do not comply with Semantic Web principles, hindering interoperability and knowledge sharing. There is a need for a more comprehensive framework that combines the strengths of Semantic Web technologies and KGs to democratize ML pipelines, making them more accessible, understandable, and reusable. This framework should provide a holistic view of ML pipelines, including their constituent components, data flow, and relationships with datasets.

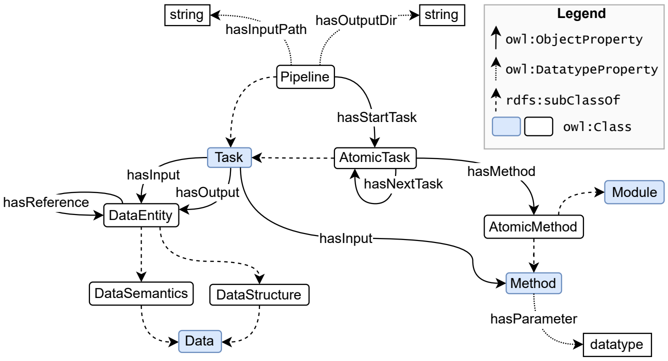

Figure 1: Top-level ExeKG Ontology: *Data Science*

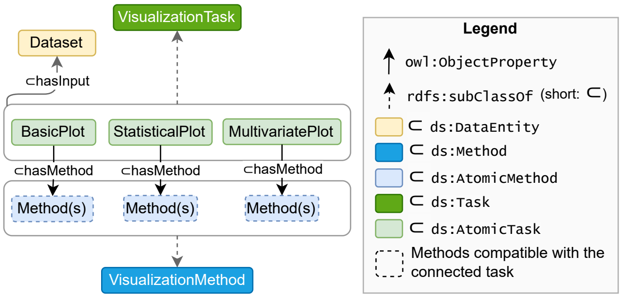

Figure 2: A Bottom-level ExeKG Ontology: *Visualization*

4 Proposed Approach

To address the challenges of accessibility and fragmentation in the ML landscape, we propose a novel approach that leverages KGs, ontologies, and KG-based ML techniques and presented below.

4.1 Ontologies

We propose a publicly available ontology structure [1], extending the work presented in [17]. This structure consists of a top-level *Data Science* ontology and three specialized bottom-level ontologies for *Machine Learning*, *Visualization* (Fig. 2), and *Statistics*. The top-level ontology, referred to as the Data Science (DS) Ontology (Fig. 1), defines general concepts such as *Data*, *Method*, and *Task*.

The bottom-level ontologies, such as the ML ontology, are specialized extensions of the DS ontology. They contain subclasses of *AtomicTask* and *AtomicMethod*, each representing a specific task type solvable by a group of methods. For instance, *Classification* tasks can be solved using methods implementing algorithms like k-NN. The Statistics and Visualization ontologies follow the same structure but with content tailored to their respective domains.

We use a semi-automatic process to generate parts of our ontologies by leveraging popular data science Python libraries like `scikit-learn`, `matplotlib`, and `numpy`. Our conversion tool extracts information from these libraries and converts it into KG components, ensuring the ontologies are up-to-date and consistent. This tool also generates SHACL constraints to maintain the validity of the generated ExeKGs.

[1] https://github.com/nsai-uio/ExeKGOntology

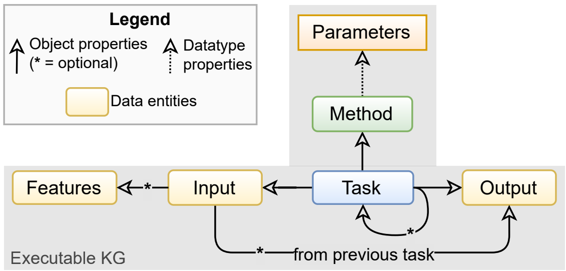

Figure 3: High-level Representation of an ExeKG

4.2 Tools

Building upon the foundation established in [10], we are developing two tools that utilize these ontologies.

ExeKGLib: A Python Library for Creating and Managing ExeKGs. ExeKGLib is our publicly available Python library [2] that enables the creation and manipulation of ML pipelines as ExeKGs (Fig. 3). It provides a programmatic interface for defining pipeline components (e.g., a "Task" that is fulfilled by a "Method") and their interconnections, representing them as nodes and edges within the KG. ExeKGLib uses predefined ontologies to ensure semantic consistency and enables validation of pipeline structures against these ontologies. Furthermore, it offers functionality for serializing and deserializing ExeKGs in standard formats like RDF, facilitating storage, sharing, and interoperability with other Semantic Web tools. ExeKGLib’s core purpose is to bridge the gap between high-level pipeline descriptions and executable code, enabling automated reasoning, validation, and generation of user interfaces tailored to the semantics of ML pipelines.

ExeKGLab: A GUI-based Platform for Interacting with ExeKGs. ExeKGLab is a user-friendly GUI-based platform (Fig. 4) built on top of ExeKGLib. It provides an intuitive visual interface for interacting with ExeKGs, making it easier for users to understand, create, and modify ML pipelines. ExeKGLab incorporates an LLM-assisted interface for no-code ML, facilitating pipeline creation and enhancing pipeline understandability for users with varying levels of expertise. Its functionalities include visualizing, editing, and executing ExeKGs, as well as exploring relevant ML artifacts. For example, users can visually construct pipelines by dragging and dropping components from a palette, connecting them to define the data flow, and setting hyperparameters through interactive widgets.

4.3 Dataset and Pipeline Recommendation

We leverage our tools and ontologies to support two key tasks.
Dataset Recommendation. Given a chosen dataset, we recommend similar datasets based on a multifaceted similarity measure. This measure considers inherent dataset properties (e.g., number of features, data types) and the pipelines previously applied to the dataset. We consider the structure of pipelines used with the dataset, coupled with the pipelines’ performance ranking on that dataset. Datasets exhibiting similar pipeline structures and comparable performance rankings on those pipelines are deemed more similar. This approach helps users find datasets with similar characteristics and performance profiles when used with related pipelines.

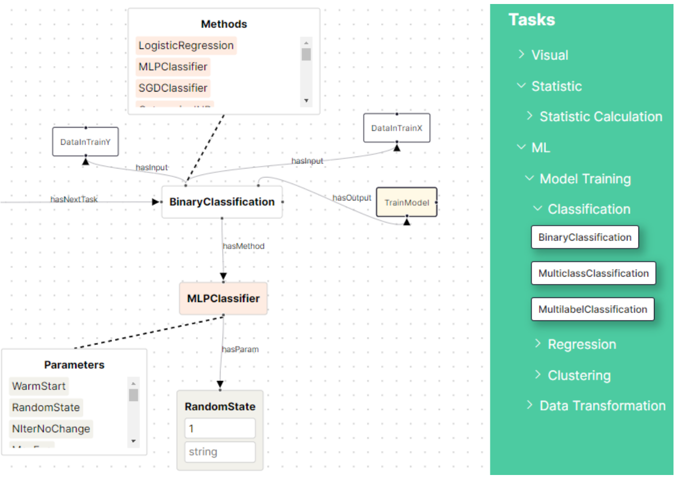

Figure 4: The ExeKGLab GUI. Users can construct ML pipelines by dragging elements (right-hand side) and dropping and configuring them on the canvas (left-hand side).

Pipeline Recommendation. Given a new dataset, we identify promising pipelines regarding expected performance. This task leverages the knowledge embedded in our ExeKGs to recommend pipelines based on the new dataset’s characteristics and the performance of the pipelines on similar datasets.
Methodology. We first convert OpenML’s datasets and pipelines into ExeKGs. This involves extracting the relevant information from OpenML, such as the dataset metadata, the pipeline structure, and the algorithm hyperparameters, and representing it as ExeKGs using our ontologies and ExeKGLib. We then employ KG embedding techniques to learn vector representations of datasets and pipelines, enabling the calculation of KGE-based similarity measures, such as cosine similarity. These embeddings capture the semantic relationships between datasets and pipelines, allowing us to compare them based on their underlying characteristics and relationships.

4.4 Proposed Evaluation

We evaluate our approach via automated and user-centric methods.

For **dataset recommendation**, we evaluate our KGE-based similarity measures by comparing them to existing dataset similarity measures. This involves analyzing the correlations between our measures and baselines when applied to a large set of dataset pairs. Strong correlations with existing measures will indicate that our approach captures relevant notions of similarity. To further validate our approach, we conduct user studies to verify that our similarity measures capture nuanced similarities beyond simple text matching. In these studies, users are presented with dataset pairs exhibiting varying degrees of similarity (high, medium, low) and asked to provide binary judgments (similar or not similar). This user feedback will help us assess the alignment between our computational measures and human perception of dataset similarity.

For **pipeline recommendation**, we perform automated evaluation by comparing our approach to existing AutoML solutions. This involves comparing our recommended pipelines’ performance to those generated by state-of-the-art AutoML systems on benchmark datasets. By comparing the performance of our recommendations

[2] https://github.com/boschresearch/ExeKGLib

Table 1: KG Statistics

# ExeKGs	# Entities	# Relations	# Triples
3,165	2,660,337	187	7,916,692

against established AutoML techniques, we can assess the effectiveness of our approach in identifying high-performing pipelines.

4.5 Use Case: KNIME Workflow Suggestion

KNIME [2] is an open-source platform for creating visual data science workflows with a drag-and-drop GUI. It supports a variety of data mining tasks. Its wide use within Bosch motivated our choice. To demonstrate the practical applicability of our approach, we present a use case focused on recommending KNIME workflows.

We first convert a collection of KNIME workflows into ExeKGs, capturing their structure and components in a semantically rich representation. Then, we train a KGE model on these ExeKGs to learn vector embeddings for each workflow. These embeddings capture the semantic relationships between different workflows, enabling us to calculate their similarity. By leveraging these similarity measures, we can provide recommendations to users, helping them explore relevant KNIME workflows for their specific needs.

To evaluate the effectiveness of our workflow recommendations, we can conduct a user study with KNIME users. In this study, users would be presented with a selection of workflows, each accompanied by a set of recommended workflows generated by our system. Users would then be asked to provide feedback on the relevance and usefulness of these recommendations. To simplify the feedback process, users could provide binary judgments (e.g., "useful" or "not useful") on each recommended workflow, indicating whether they perceive the recommendation as helpful for their needs and tasks.

5 Preliminary Experiments

To conduct our preliminary evaluation, we constructed a KG from OpenML pipelines and datasets.

Data Source. We focused on sklearn-based pipelines from OpenML, selecting the best-performing pipeline based on F1-score (classification) or RMSE (regression) for each dataset. Each pipeline was then transformed into an ExeKG.

ExeKGs Construction. We construct ExeKGs by extracting information from OpenML pipelines and leveraging ExeKGLib [10] for the conversion process. This involves representing each dataset's features as `DataEntity` nodes and analyzing the pipeline's sequence of operations. For each operation, we create corresponding `Task` and `Method` nodes, linking each `Method` node to its parameters. These operations span various stages of the ML pipeline, including data preprocessing, feature engineering, and learning algorithms.

KG Statistics. The resulting KG comprises 3,165 ExeKGs across 312 datasets, with 2,660,337 entities, 187 relations, and 7,916,692 triples. Table 1 provides a summary of the KG statistics.

Dataset Pairs. From the 312 datasets, we generated 48,516 possible dataset pairs. Due to computational constraints, we randomly sampled 10,000 pairs for our analysis, encompassing binary and multiclass classification and regression tasks. We are currently evaluating the performance of our dataset and pipeline recommendation approaches on this dataset, comparing our KGE-based similarity measures to existing baselines and conducting user studies to assess the quality and relevance of our recommendations.

6 Conclusion and Future Work

By leveraging KGs, our research aims to empower users of all backgrounds to effectively locate, understand, and adapt ML pipelines to their specific needs. This approach promotes accessibility, collaboration, and ultimately wider adoption of ML techniques.

Future work includes expanding ExeKGLab's support for ML algorithms and preprocessing, requiring ontology enrichment and robust metadata integration. We will investigate advanced reasoning techniques, including semantic similarity, for sophisticated pipeline discovery, moving beyond keyword search. User studies will evaluate ExeKGLab's usability and effectiveness in supporting pipeline design and collaboration. Finally, we will explore ExeKGs for AutoML via graph-based optimization.

Acknowledgments

The work was partially supported by EU Projects Graph Massivizer (GA 101093202), Dome 4.0 (GA 953163), enRichMyData (GA 101070284), and SMARTY (GA 101140087).

References

[1] S. Berg, D. Kutra, T. Kroeger, C.N. Straehle, B.X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, et al. 2019. Ilastik: interactive machine learning for (bio) image analysis. *Nature methods* 16, 12 (2019), 1226–1232.

[2] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. 2007. KNIME: The Konstanz Information Miner. In *GfKL 2007*. Springer.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *NeurIPS*, Vol. 26. Curran Associates, Inc.

[4] D.H. Chau, A. Kittur, J.I. Hong, and C. Faloutsos. 2011. Making sense of large network data by combining rich user interaction and ML. In *SIGCHI*. 167–176.

[5] Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'Connell, Terrance Gray, F. Maxwell Harper, and Haiyi Zhu. 2019. Explaining Decision-Making Algorithms through UI. In *CHI* (Glasgow, Scotland Uk). ACM, New York, NY, USA, 1–12.

[6] John J Dudley and Per Ola Kristensson. 2018. A review of user interface design for interactive machine learning. *TiiS* 8, 2 (2018), 1–37.

[7] Rebecca Fiebrink and Perry R Cook. 2010. The Wekinator: a system for real-time, interactive machine learning in music. In *ISMIR 2010*, Vol. 3. Citeseer, 2–1.

[8] M. Helali, E. Mansour, I. Abdelaziz, J. Dolby, and K. Srinivas. 2022. A scalable AutoML approach based on graph neural networks. *VLDB* 15, 11 (2022), 2428–2436.

[9] Jörg-Uwe Kietz, Floarea Serban, Simon Fischer, and Abraham Bernstein. 2014. "Semantics Inside!" But Let's Not Tell the Data Miners: Intelligent Support for Data Mining. In *ESWC*. Springer International Publishing, Cham, 706–720.

[10] Antonis Klironomos, Baifan Zhou, Zhipeng Tan, Zhuoxun Zheng, Gad-Elrab Mohamed, Heiko Paulheim, and Evgeny Kharlamov. 2023. ExeKGLib: knowledge graphs-empowered machine learning analytics. In *ESWC*. Springer, 123–127.

[11] Fadi Maali and John Erickson. 2014. *Data Catalog Vocabulary (DCAT)*. Recommendation. W3C.

[12] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. 2006. YALE: Rapid Prototyping for Complex Data Mining Tasks. In *ACM SIGKDD (KDD '06)*. Association for Computing Machinery, New York, NY, USA, 935–940.

[13] G.C. Publio, D. Esteves, A. Ławrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren, and H. Zafar. 2018. ML-Schema: Exposing the Semantics of Machine Learning with Schemas and Ontologies. arXiv:1807.05351 [cs, stat]

[14] Sergey Redyuk, Zoi Kaoudi, Sebastian Schelter, and Volker Markl. 2024. Assisted design of data science pipelines. *The VLDB Journal* (2024), 1–25.

[15] Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International semantic web conference*. Springer, 498–514.

[16] Z. Shang, E. Zgraggen, B. Buratti, F. Kossmann, P. Eichmann, Y. Chung, C. Binnig, E. Upfal, and T. Kraska. 2019. Democratizing data science through interactive curation of ml pipelines. In *SIGMOD*. 1171–1188.

[17] Zhuoxun Zheng, Baifan Zhou, Dongzhuoran Zhou, Xianda Zheng, Gong Cheng, Ahmet Soylu, and Evgeny Kharlamov. 2022. Executable knowledge graphs for machine learning: a Bosch case of welding monitoring. In *ISWC*. 791–809.