December 2020

Metadata Service to Enable Display of Rich Artifacts in Machine Learning Pipelines

Ajay Alfred

Pavel Dournov

Jingxiao Wu

Hongye Sun

Apoorv Verma (AP)

See next page for additional authors

Inventor(s)
Ajay Alfred, Pavel Dournov, Jingxiao Wu, Hongye Sun, Apoorv Verma (AP), Ajay Gopinathan, Hui Miao, and Neoklis Polyzotis

Metadata Service to Enable Display of Rich Artifacts in Machine Learning Pipelines

ABSTRACT

Cloud-based machine learning (ML) platforms enable ML practitioners to build, rebuild, and serve multiple machine learning models in production environments. Proper ML metadata tracking and management is important to enable large-scale experimentation and to provide traceability and verifiability for modern production ML. This disclosure describes a ML metadata service to manage the lifecycle of metadata consumed and produced by ML pipelines. The ML metadata service enables logging detailed metadata as artifacts, capturing metadata as typed artifacts, and capturing a ML pipeline in an intuitive workflow graph. The metadata service enables provision of a ML dashboard that displays visualizations of a ML workflow along with the relevant metadata for each type of entity and supports queries for models and/or datasets that meet specific criteria.

KEYWORDS

- Machine learning
- ML metadata
- ML pipeline
- Workflow graph
- Data visualization
- Model provenance
- Post-hoc analysis

BACKGROUND

Cloud-based machine learning (ML) platforms enable ML practitioners to build, rebuild, and serve multiple machine learning models in production environments. Production grade ML platforms typically provide a pipeline comprising the following functionalities: data storage; data preprocessing and feature engineering; model training; model management (including model performance evaluation and model selection); and model deployment and serving.

A crucial aspect that often gets overlooked in traditional ML platforms is the systematic management of ML metadata. ML metadata can include information about data sets, executions, features, models, and other artifacts from a machine learning pipeline. Proper ML metadata tracking and management on a ML platform is important to enable large-scale experimentation and to provide traceability and verifiability for modern production ML.

DESCRIPTION

This disclosure describes a ML metadata service to manage the lifecycle of metadata consumed and produced by machine learning (ML) pipelines. The ML metadata service addresses the following primary use cases:

- Logging detailed metadata (as artifacts) from production ML pipelines to enable data aware orchestration and post-hoc analysis.
- Capturing metadata as typed artifacts such as models, datasets, metrics during ML pipeline execution phase to aid in reasoning about performance and analysis.
- Capturing the ML pipeline in an intuitive workflow graph driven user interface (UI) with rich visual representations of data artifacts (datasets, models, metrics, etc.) for easy consumption and awareness of the information passing through the pipeline.

The ML metadata service described herein can be used to automatically capture the metadata produced by data-driven ML pipelines and store it in a centralized registry. The centralized registry can be used to potentially power a ML dashboard, where users can see visualizations of the various pieces of metadata representing their ML workflows as well as the relationships between them. As shown in Fig. 1, the ML metadata service can be used to handle both managed artifacts (e.g., API resources on ML Platform) as well as unmanaged ones (e.g., models and/or datasets stored in users' local and cloud storage, accessed with permission). In both cases, the centralized registry references the artifacts and the executions that produced/consumed them, as well as their relationships.

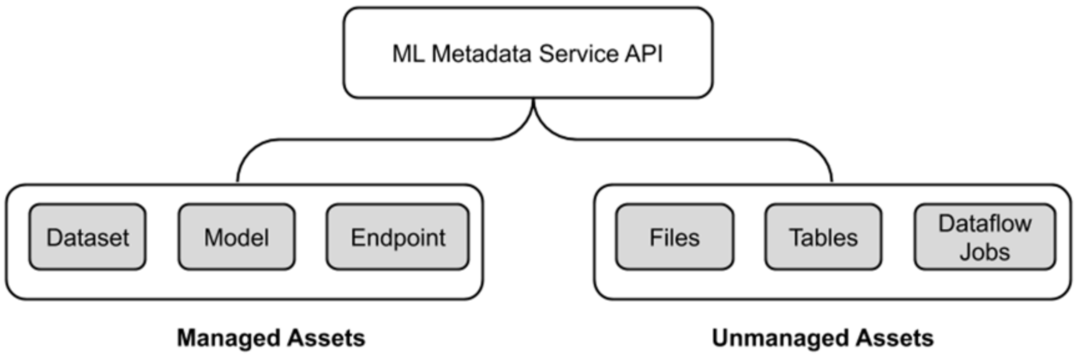

Fig. 1: Types of assets handled by ML metadata service

This functionality is enabled by using a set of application programming interfaces (APIs) that record the metadata for machine-learning *artifacts* and *executions*, with support for *events* that record workflow data provenance and *contexts* for arbitrary but typed grouping of metadata. Fig. 2 illustrates a graph-like data model that is used to represent artifacts and their relationships in ML workflows.

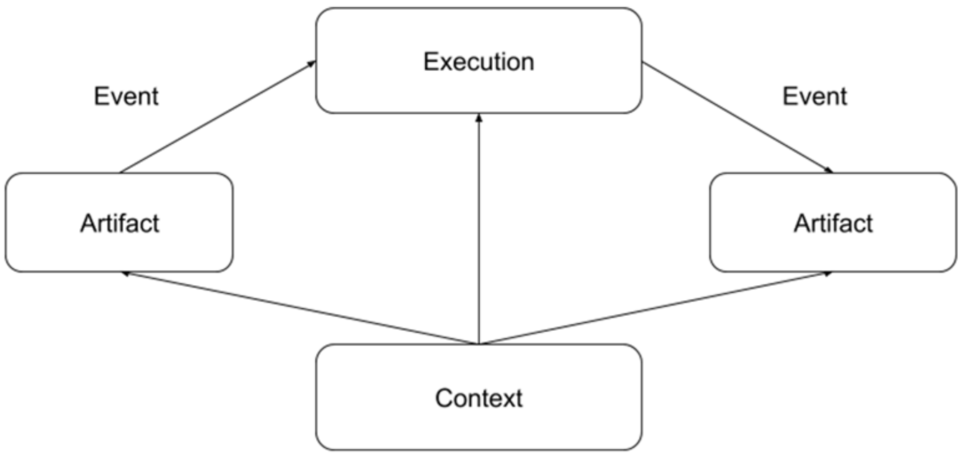

Fig. 2: Data model used for ML metadata API

Broadly speaking, the main types of concepts represented in the data model of the ML metadata service are *artifacts*, *executions*, *contexts*, and *events*.

- **Artifacts:** Artifacts are used to capture individual pieces of data produced and consumed by a machine-learning computation step. Some examples of artifacts include input files, transformed datasets, models, training logs, etc. A dataset represents a container of data that was either consumed or produced by a ML workflow step. A model represents a trained machine-learning model that can be deployed to a serving infrastructure. Metadata for a model captures information such as framework type and version used to produce the model. Metrics represent evaluation metrics for the specific application, and may include either or both of simple summary metrics (e.g., average accuracy of the model) as well as complex metrics (usually for a time series).
- **Executions:** Executions are used to describe an individual machine-learning workflow step in terms of its runtime parameters, input and output artifacts, etc.
- **Events:** Events are used to capture the relationship between artifacts and executions, where each artifact can be produced and/or consumed by executions. Events enable users to determine the provenance in their ML workflows, for example, the particular dataset used to produce a model.
- **Contexts:** Contexts are used to group artifacts and executions together under a single, queryable and typed category. For example, contexts can be used to represent one or more runs of the same machine-learning pipeline, a single notebook, or a user-specified named experiment.

Artifacts, executions and contexts can be ascribed with key-value pairs that describe their properties. For example, a model artifact can have properties that describe the framework and/or algorithm used to produce it, as well as the performance of the model in terms of metrics such as accuracy, precision, recall etc. Additionally, the ML metadata service can enable users to create custom user-defined types.

Fig. 3 illustrates an example of a ML dashboard, where users can see visualizations of their ML workflow along with the relevant metadata for each type of entity. The Artifacts/Executions as illustrated in Fig. 3 are scoped to the context of execution of a specific ML pipeline.

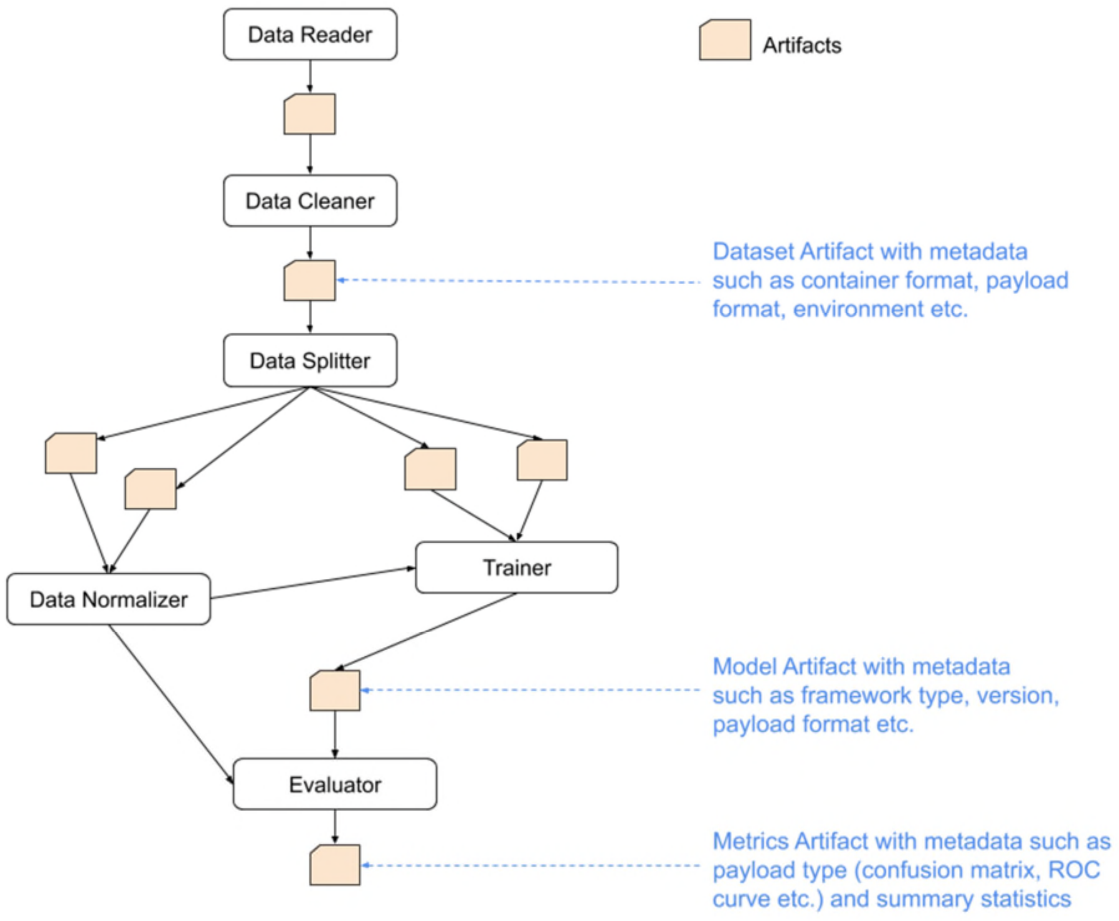
Data Reader
Artifacts
Data Cleaner
Dataset Artifact with metadata such as container format, payload format, environment etc.
Data Splitter
Data Normalizer
Trainer
Model Artifact with metadata such as framework type, version, payload format etc.
Evaluator
Metrics Artifact with metadata such as payload type (confusion matrix, ROC curve etc.) and summary statistics

Fig. 3: Example of a ML dashboard

Fig 4. illustrates an example use case for the ML metadata service. In this example, the user is running a simple model training step and uses the API to record pertinent metadata describing the workflow. The recorded metadata can be easily queried at a later time.

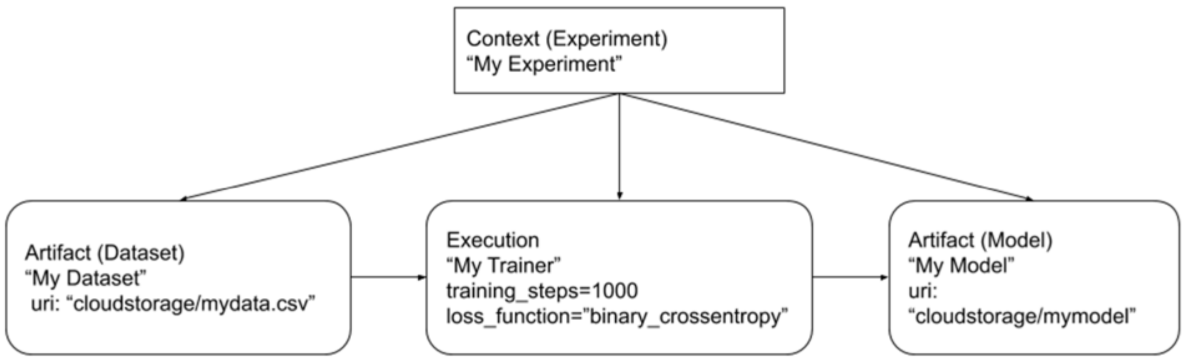

Fig. 4: Using the ML metadata service

In the example of Fig. 4,

- *Dataset* and *Model* are system-defined types
- *MyTrainer* is a user-defined training step that executes in the user's notebook
- *Experiment* is a system-defined grouping concept for organizing metadata produced by workflows within some experiment

The user first defines a file to describe their custom types, and stores it in a cloud location accessible to the ML metadata service API. After this is done, the user can use the ML metadata service to record metadata that describes the workflow. The user begins by recording the system-defined *Model* and *Dataset* artifacts. Next, the user registers their *Training* step. Since this uses a user-defined schema, the user specifies the location to the file containing the type definition(s). In the next step, the user constructs the lineage graph connecting *Dataset-> My Trainer->*

Model. After this is done, the user groups the entire lineage graph under an *Experiment* to organize the workflow.

The metadata service can be used to answer queries such as, for example:

- `Find all Models produced under the experiment MyExperiment with a recorded model accuracy > 0.9`
- `Find all Models produced using a specific Dataset ‘artifact/my_dataset_1’`
- `Find all Datasets that were used as input when producing the Model ‘artifacts/my_model_1’`

CONCLUSION

This disclosure describes a ML metadata service to manage the lifecycle of metadata consumed and produced by ML pipelines. The ML metadata service enables logging detailed metadata as artifacts, capturing metadata as typed artifacts, and capturing a ML pipeline in an intuitive workflow graph. The metadata service enables provision of a ML dashboard that displays visualizations of a ML workflow along with the relevant metadata for each type of entity and supports queries for models and/or datasets that meet specific criteria.

REFERENCES

1. https://github.com/google/ml-metadata