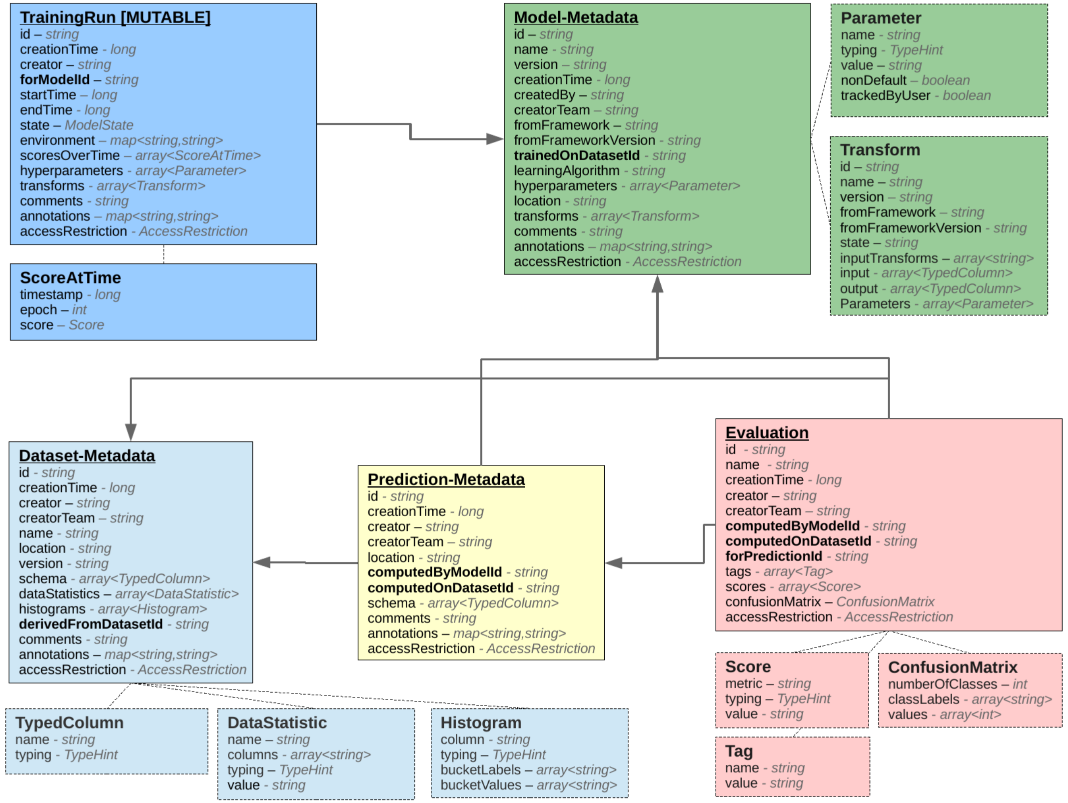
Optimization of Machine Learning Workloads with Experiment Databases

Behrouz Derakhshan

Behrouz.derakhshan@dfki.de

Experiment Database

- Contains meta-data of machine learning experiments
- Based on Schelter, et al., [1] database schema for storing experiments’ meta-data
- Their meta-data extraction system collects information about the datasets, models, transformations, and evaluations



Database schema for storing declarative description of machine learning experiments [1]

Machine Learning Workloads

Next paper

Future Work

1. Interactive batch workload

- Multi-user (from tens to thousands of users)
- Primarily consists of repeated data preprocessing and data transformations
- Aggressive Hyper-parameter tuning and model evaluation
- Use case: OpenML[2], Kaggle[3]
- Opportunities for Reuse, materialization, multi-query optimization, and warm-starting

2. Incremental Training Workload

- Multi-user (tens of users)
- Incremental improvement of existing data processing pipelines and ML models
- Retrain models and pipelines (typically daily)
- Use case : Industry
- Speed up in data preprocessing, decreasing the search space of hyper-parameters, multi-model training, and warm-starting

3. Continuous Training Workload

- Real-time (or near real-time) data processing
- Online ML models
- Use case : Industry
- Speed up in data preprocessing, multi-model training, and fast detection of model quality degradation

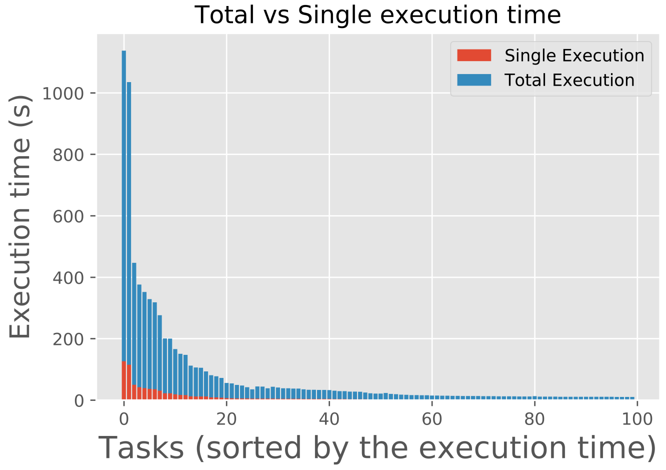
The green color indicates how Experiment Database can help in optimizing the specific type of workload

Plan for next Paper

- VLDB 2018 (Deadline: 1st of March)
- Focus on Workload 1 (Interactive batch)
- Plan:
  - Develop a simple database based on [1]
  - Experiment:
    - Focus only on a few frameworks (scikit-learn, R mlr package*)
    - Popular Kaggle competitions, OpenML datasets and tasks
    - Report the reduction in the processing time and the development time

- I currently have a paper under review for SIGMOD2018. Depending on the result of the review, this work may have to be pushed back to SIGMOD2019/VLDB2018

- Early Results:
  - The Most popular pipeline in OpenML (scikit-learn) consists of:
    - Missing Value Imputer
    - Dimensionality reduction using PCA
    - Random Forest Classifier
  - It is repeatedly executed on 100 different tasks (average of 9 times on each task)
  - A simple reuse can save around 2 hours of processing time



This figure may include executions performed by bots, therefore, we cannot reach a conclusion about the reduction on the development time

References

[1] *Sebastian Schelter, Joos-Hendrik Böse, Johannes Kirschnick, Thoralf Klein, Stephan Seufert*, **Automatically Tracking Metadata and Provenance of Machine Learning Experiments**, Machine Learning Systems workshop at the conference on Neural Information Processing Systems (NIPS) 2017

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

[3] https://www.kaggle.com/