



MACHINE LEARNING AS A SERVICE FOR THE EARTH SCIENCES

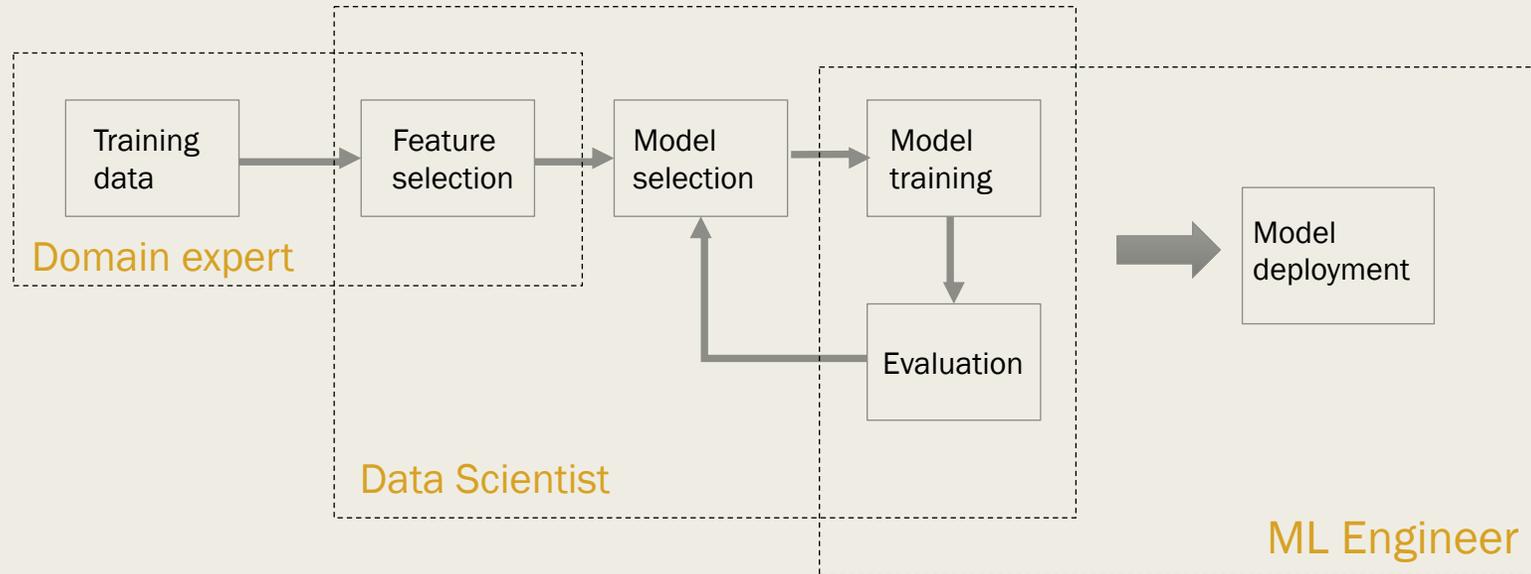
Presenter – Sujen Shah (NASA JPL/Caltech)



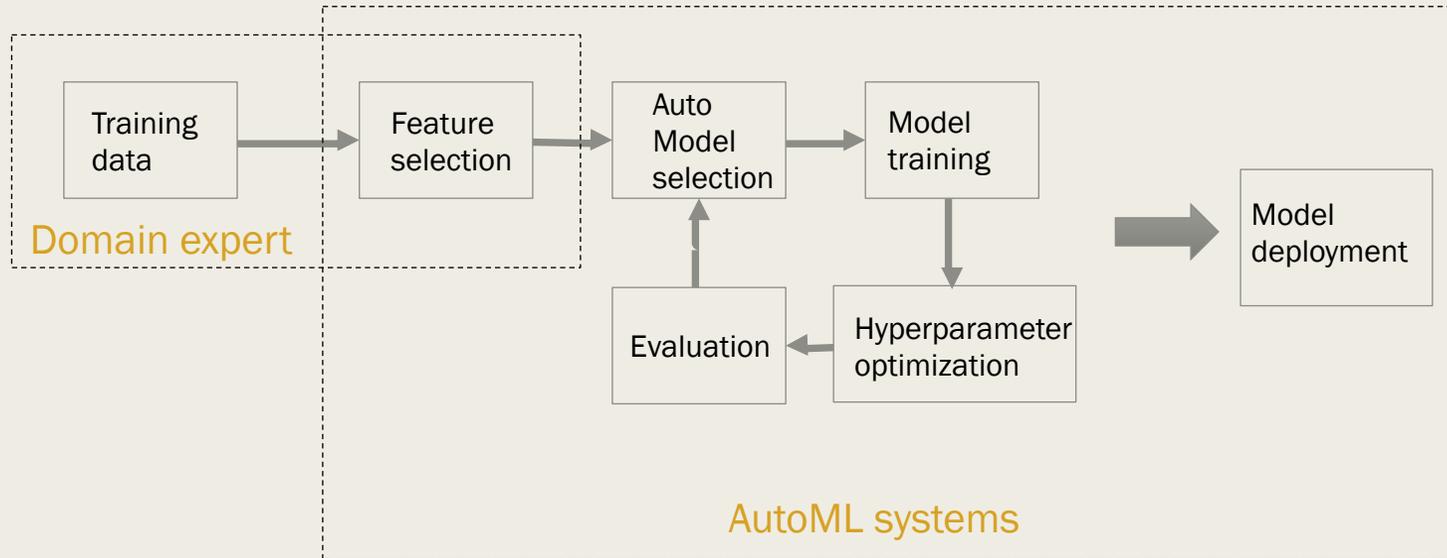
Overview

- (Automated) Machine Learning Workflow
- Typical D3M System
- D3M Open Source Resources
- Demo

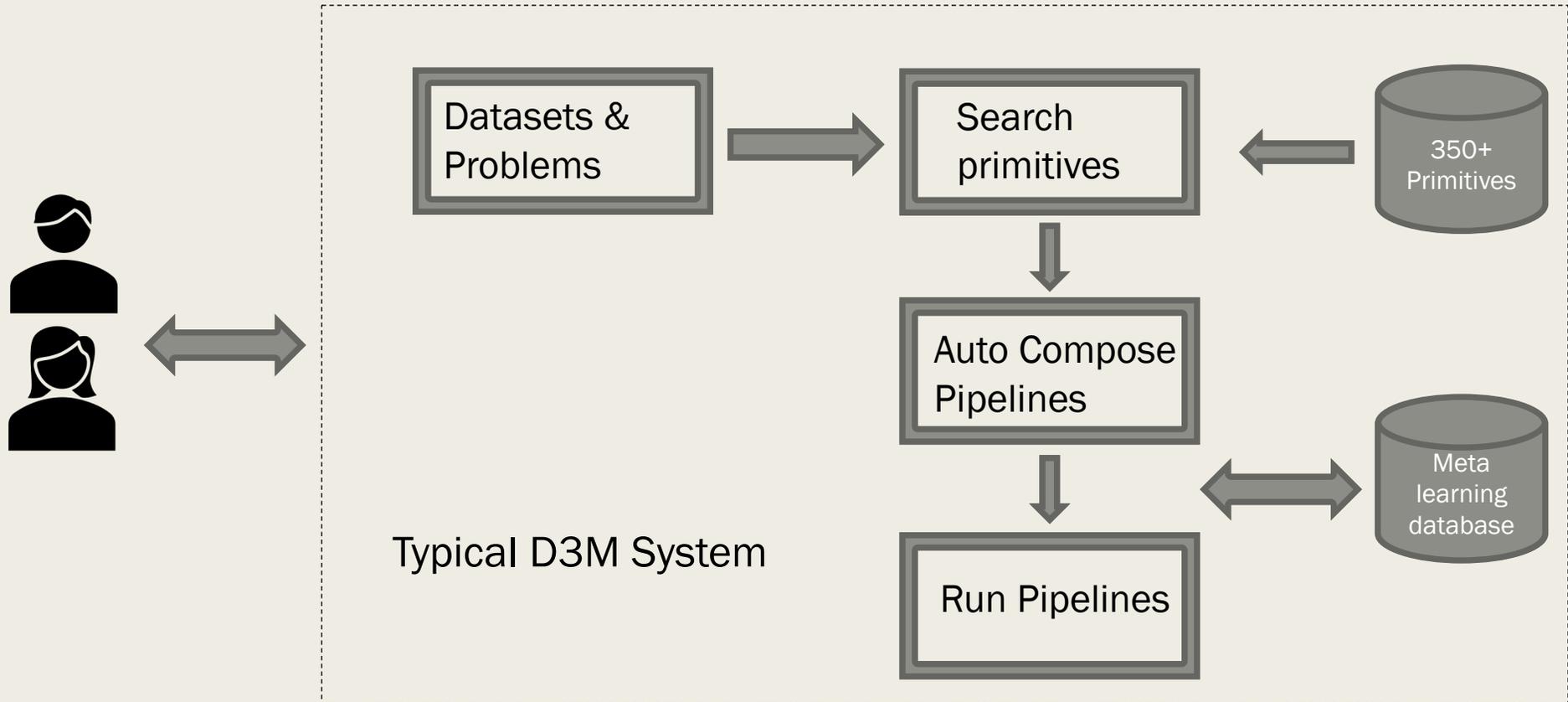
Machine learning workflow



Automated Machine learning workflow



Data Science Workflow using D3M Open Source Resources



D3M Open Source Resources

- D3M Core library
- Dataset schema
- Problem schema
- Curated Library of ML/DL Primitives
- Pipeline and Reference Runtime
- Metalearning Database

D3M Core Library

- Open source - <https://gitlab.com/datadrivendiscovery/d3m>.
- Easy install – pip install d3m.
- Multiple interfaces for different type of primitives
 - *clustering*
 - *distance*
 - *featurization*
 - *generator*
 - *supervised_learning*
 - *transformer*
 - *unsupervised_learning*

D3M Dataset Schema

- Dataset schema provides a specification of an abstract dataset.
- Open - https://gitlab.datadrivendiscovery.org/MIT-LL/d3m_data_supply/blob/shared/documentation/datasetSchema.md
- Program has curated over 400 datasets.

D3M Problem Schema

- Problem schema specifies a dataset in three sections: about, inputs, and expected Outputs.
- Open - https://gitlab.datadrivendiscovery.org/MIT-LL/d3m_data_supply/blob/shared/documentation/problemSchema.md

Curated Library of ML/DL Primitives

- Primitives integrated into the d3m core library via interfaces.
- Metadata generated and stored to allow search over the primitives library.
- Open - Primitive schema ([json](#))
<https://metadata.datadrivendiscovery.org/v2019.6.7/?primitive>
- Over 350 curated primitives.
- Marvin - <https://marvin-beta.datadrivendiscovery.org/primitives>

MARVIN – UI to browse datasets, problems, primitives and pipelines

Marvin | [Home](#) | [Datasets](#) | [Problems](#) | **</> Primitives** | [Pipelines](#) | [Docker Registry](#) | [Metalearning](#)

2496 results found in 11ms

Interface version

- 2019.6.7 (351)
- 2019.5.8 (301)
- 2019.2.18 (273)
- 2018.7.10 (262)
- 2019.1.21 (262)
- 2019.4.4 (261)
- 2019.2.12 (241)

Algorithm Type

- DATA_CONVERSION (309)
- NEURAL_NETWORK_BACKPROPAGATION (97)
- DECISION_TREE (80)
- IMPUTATION (76)
- MINIMUM_REDUNDANCY_FEATURE_SELECTION (76)
- FEATURE_SCALING (75)
- RANDOM_FOREST (69)

Team

- JPL (611)

Random Sampling Imputer

Team Name: byu-dml **Interface version:** 2019.1.21 **# of Pipelines:** 1

Python Path: d3m.primitives.data_preprocessing.random_sampling_imputer.BYU

Description A primitive which takes a DataFrame with "NaN" for all missing values, and imputes them for each column by randomly sampling from the existing values of that column. If a column has no existing values (aka a completely empty column), the column is ignored and remains in the dataset unimputed"

URIs <https://github.com/byu-dml/d3m-primitives>

Contact bjschoenfeld@gmail.com

Documentation [Show Hyperparameter Documentation](#) [Show Fit method Documentation](#) [Show Produce method Documentation](#)

Dataset Metafeature Extraction

Team Name: byu-dml **Interface version:** 2019.1.21 **# of Pipelines:** 1

Python Path: d3m.primitives.metafeature_extraction.metafeature_extractor.BYU

Description A primitive which takes a DataFrame and computes metafeatures on the data. Target column is identified by being labeled with 'https://metadata.datadrivendiscovery.org/types/TrueTarget' in 'semantic_types' metadata. Otherwise primitive assumes there is no target column and only metafeatures that do not involve targets are returned. If DataFrame metadata does not include semantic type labels for each column, columns will be classified as CATEGORICAL or NUMERIC according to their dtype: int and float are NUMERIC, all others are CATEGORICAL. Metafeatures are stored in the metadata object of the DataFrame, and the DataFrame itself is returned unchanged

URIs <https://github.com/byu-dml/d3m-primitives>

Contact bjschoenfeld@gmail.com

Pipeline and Reference Runtime

- Declarative Pipelines in JSON Schema
- Automated runs in 3 Phases: fit, predict, and metric evaluation
- Pipeline schema ([json](#))
 - <https://metadata.datadrivendiscovery.org/devel/?pipeline>
- Reference runtime - <https://gitlab.com/datadrivendiscovery/d3m/blob/devel/d3m/runtime.py>
- Collect multiple runs in a database

Metalearning Database

- Enables
 - *Learning across 200+ problems and 8000+ pipeline runs*
 - *Search / compose Pipelines*
 - *Hyperparameter tuning sets by problem type & estimator*
- Metalearning api open source -
<https://gitlab.com/datadrivendiscovery/metalearning/tree/database>

Demo

- Write a solution pipeline for the 185 Baseball dataset - https://gitlab.datadrivendiscovery.org/d3m/datasets/tree/master/seed_datasets_current/185_baseball
- Generate pipeline
- Use reference pipeline to run problem
- Submit to metalearning database

Acknowledgements

- DARPA D3M Program and performers <https://www.darpa.mil/program/data-driven-discovery-of-models>
- Chris Mattmann (PI) and then JPL D3M team
- ESIP Machine learning cluster



Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov

© 2019 California Institute of Technology.
Government sponsorship acknowledged