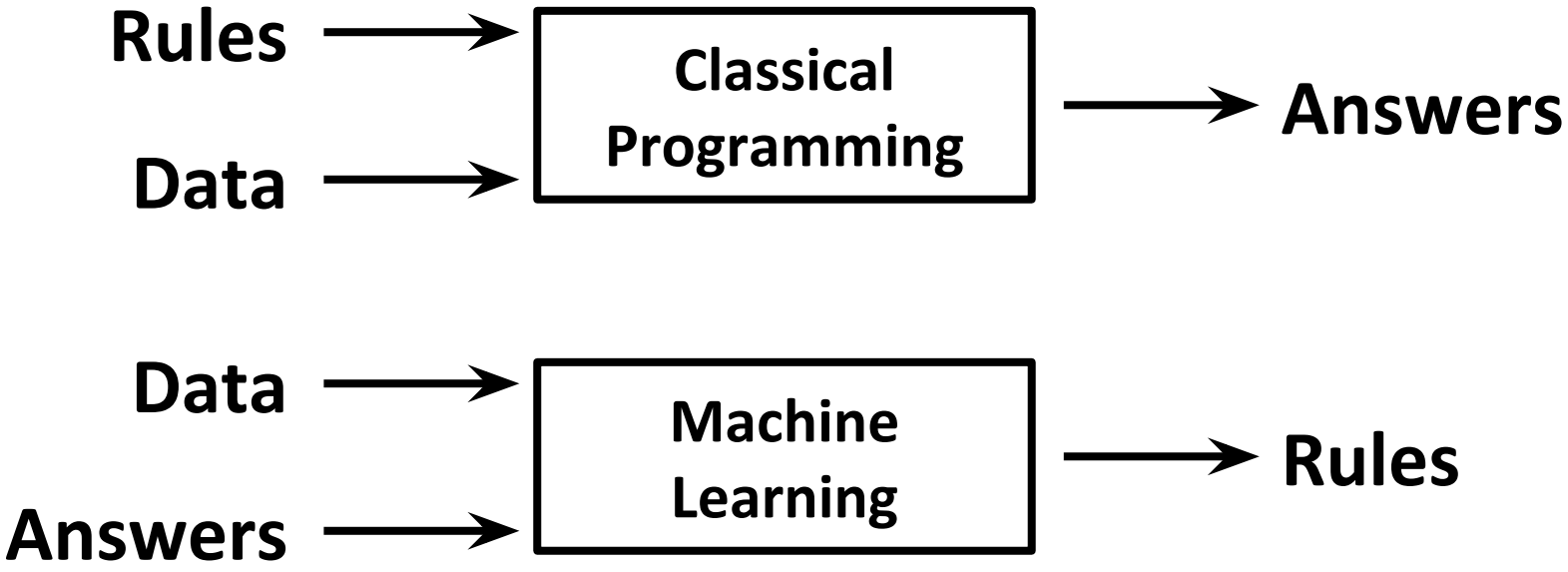
The case for (J)PMML

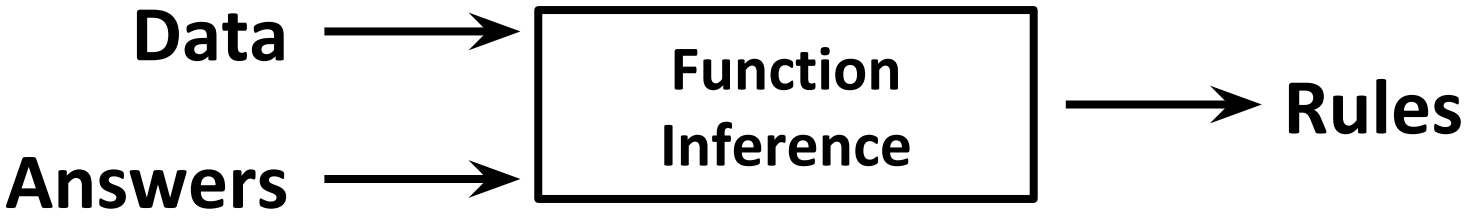
Villu Ruusmann
Openscoring OÜ

"Machines that learn"



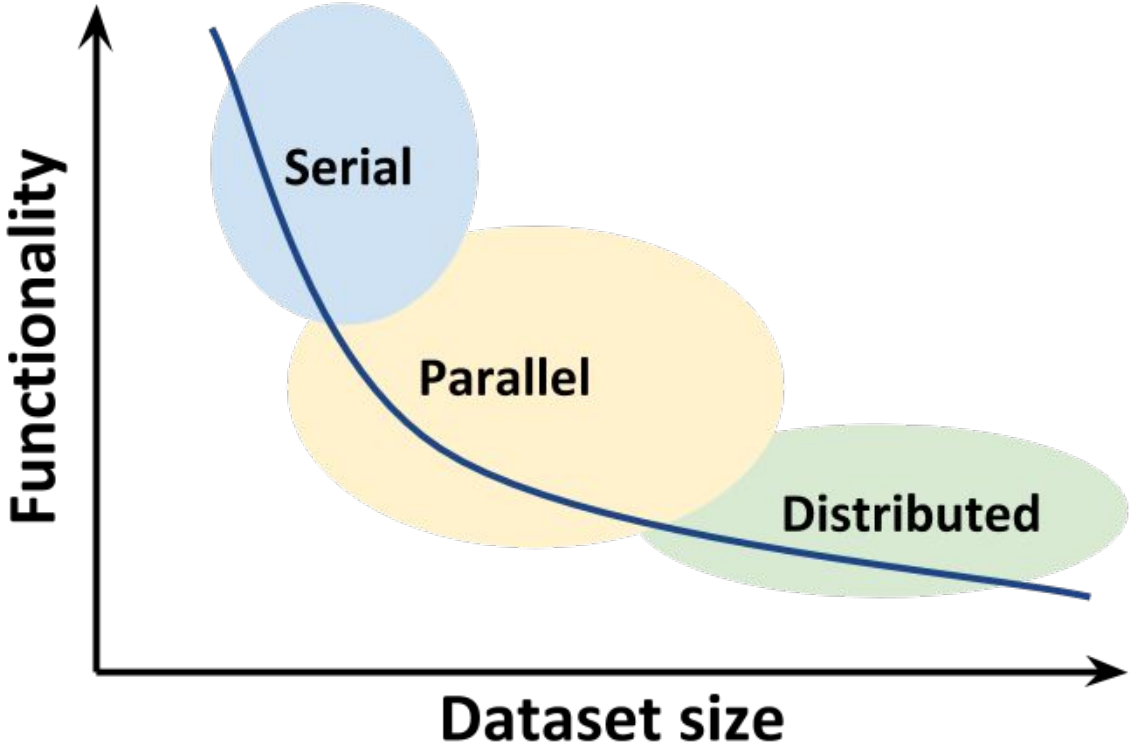
https://twitter.com/chrisalbon/status/889987842675429376

Data science

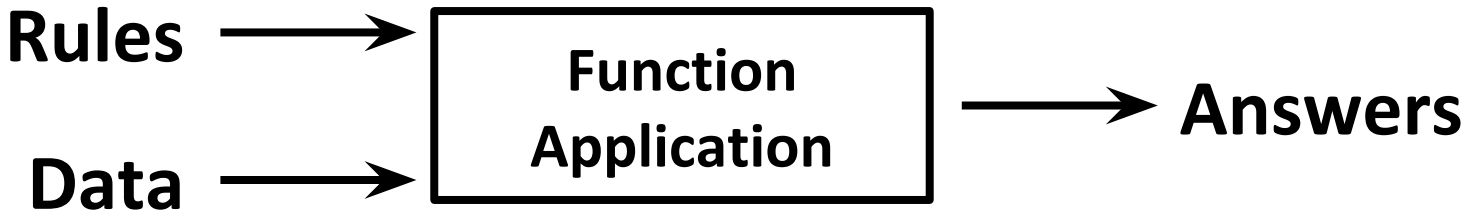


- Business design
- All existing data records in batch mode
- Mathematical optimization problem

ML algorithms

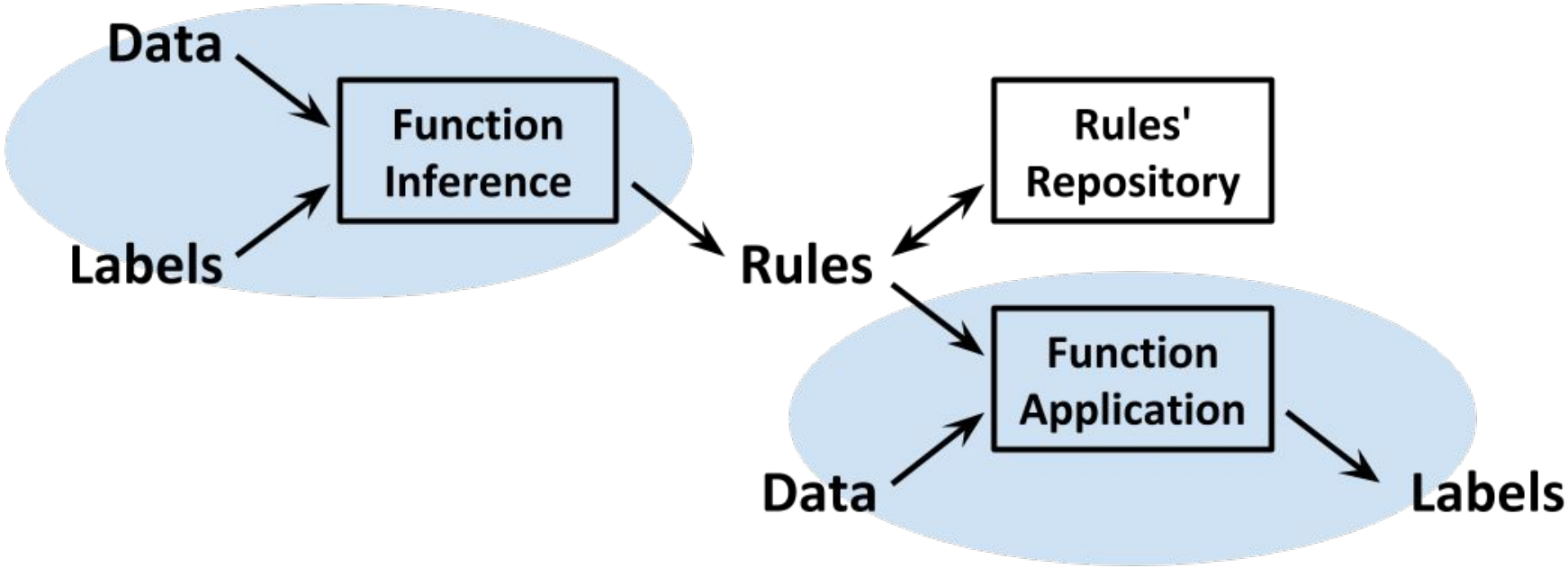


DevOps



- Business implementation
- Single new data records in real-time mode, multiple new data records in streaming and batch modes
- Ease-of-use and robustness typically preferable to raw performance

Workflow


Data
Function Inference
Labels
Rules
Rules' Repository
Function Application
Data
Labels

Data and Labels

- Unstructured data:
  - Low-level
  - Binary (video, images, voice)
- Structured data:
  - High-level
  - Conceptual, relational (SQL); possibly with sequential and/or temporal dimensions
  - Metadata

Rules

- "Black box" (aka "Artificial Intelligence") models:
  - Deep and wide neural networks
- "Grey box" (aka "Machine Learning") models:
  - Shallow and narrow neural networks
  - Ensemble models
- "White box" (aka "Statistical") models:
  - Linear and logistic regression
  - Decision tree

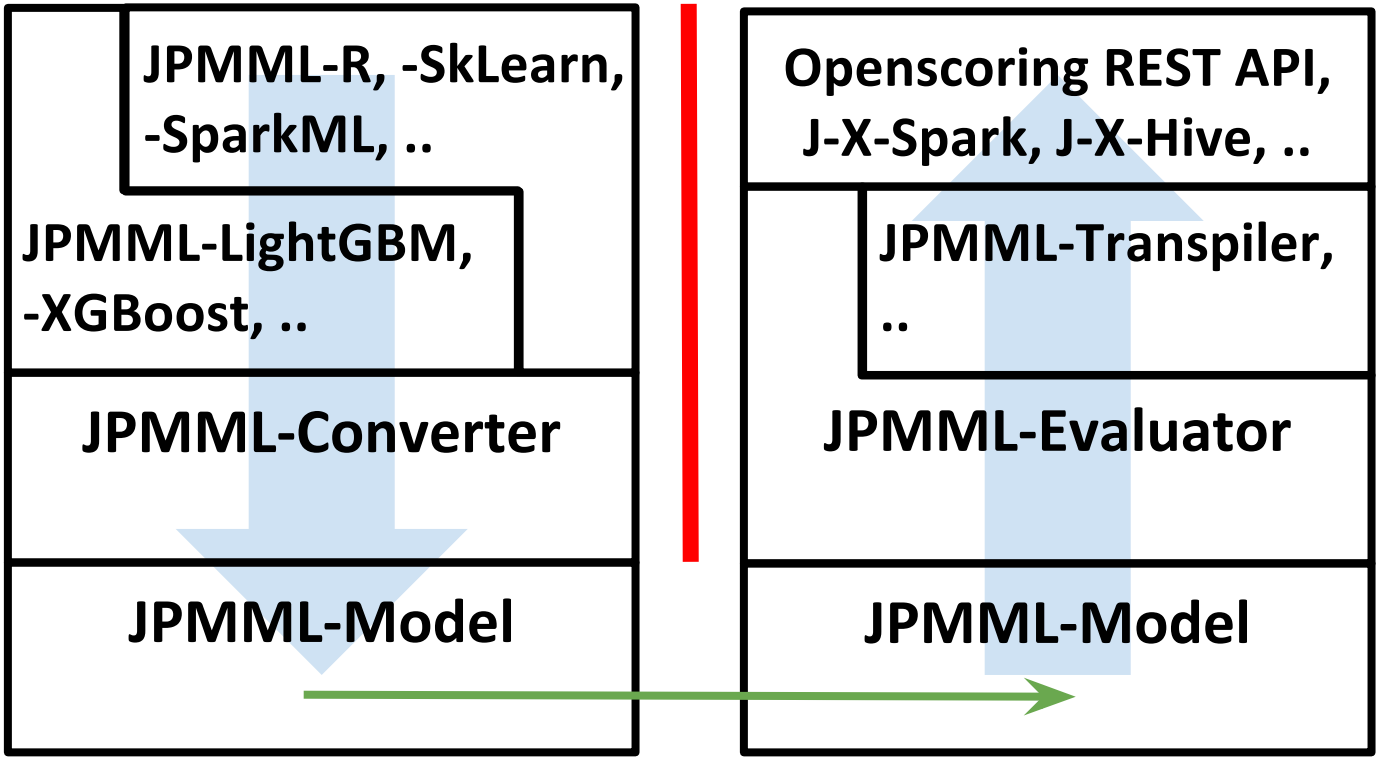
Predictive Model Markup Language (PMML)

- A DMG.org effort since 1999
- Representing rules (functions) in terms of **standardized data structures**:
  - "Conventions over configuration"
  - Backward- and forward-compatibility
  - Vendor extensions
- Platform-, language- and framework-agnostic
- Human- and machine readable, executable, writable

JPMML software stack (1/2)

- "The (Java-) API is the product"
  - PMML producer (aka model conversion) vertical
  - PMML consumer (aka model scoring) vertical
  - ML-framework vs. PMML integration testing suite
- Layered library approach
  - Lower layer(s): BSD 3-Clause License
  - Higher layers: Affero GPLv3

JPMML software stack (2/2)



Q&A

villu@openscoring.io

https://github.com/jpmml
https://github.com/openscoring
https://groups.google.com/forum/#!forum/jpmml