i3 LAB

Systematic Reproducibility and Replicability of Data Science Projects

A MDE approach

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

AGENCIA ESTATAL DE INVESTIGACIÓN

Software Engineering Group
QUERCUS
UNIVERSIDAD DE EXTREMADURA

UNIVERSIDAD DE EXTREMADURA

Associate Professor
Universidad de Extremadura

Quercus Software Engineering Group (i3 lab)

2020 - Co-founder and Research Advisor of the spin-off **metrikamedia**

2021 - Head of the **Applied Informatic Technologies Institute**

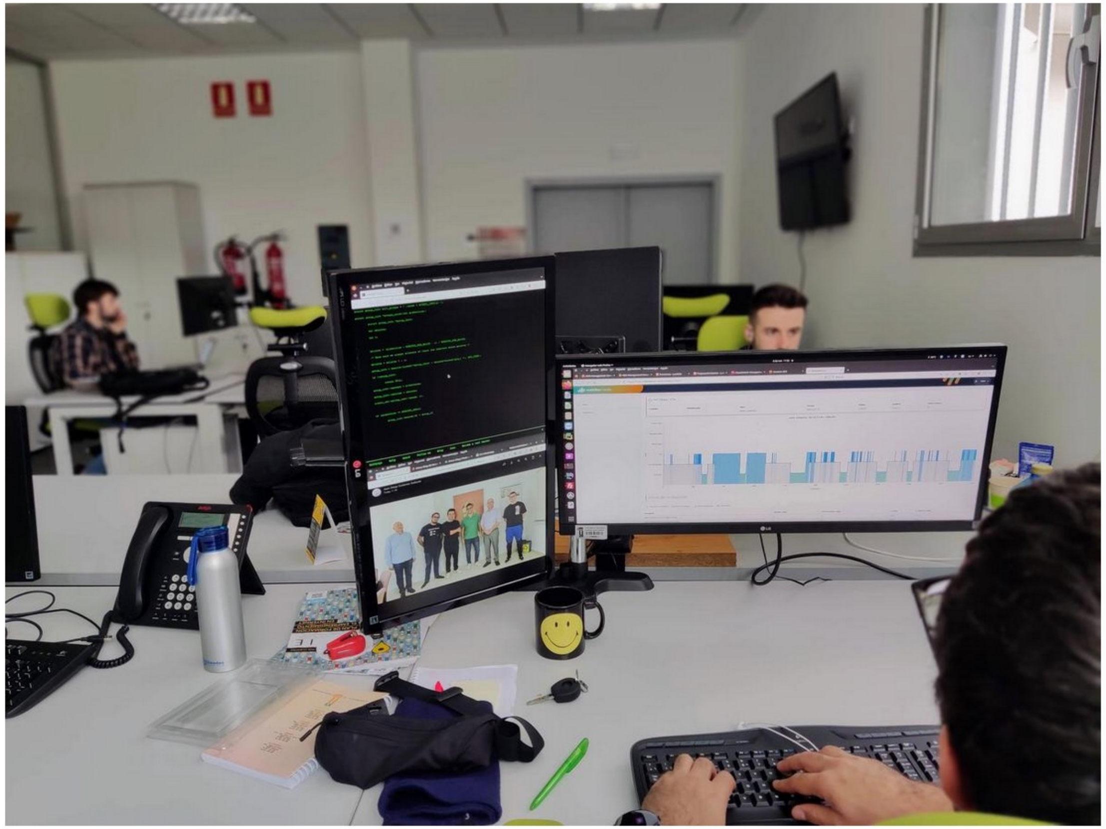
Let me introduce you…

my research lab

i3
LAB

i3 lab by QSEG

i3 lab is a multidisciplinar group of researchers, educators, and students, organized in a workspace managed by QSEG@UEx. Our three standing stones are research, innovation, and imagination. We are interested in a wide range of subjects, focusing on AI's impact on how people interact with technology, the world, and each other. Interested? Let's talk!



https://i3lab.unex.es

Principal Investigators

Fernando Sánchez-Figueroa

Associate Professor

Juan Carlos Preciado

Associate Professor

José M. Conejero

Associate Professor

Roberto Rodriguez-Echeverria

Associate Professor

Alvaro E. Prieto

Associate Professor

Researchers

Emilio Delgado

Researcher

Juan D. Gutiérrez

Researcher

Jorge Perianez

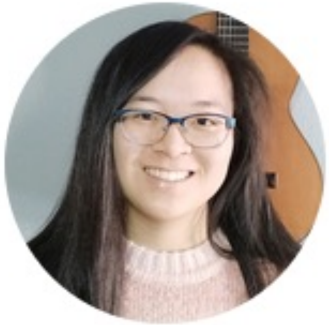
Researcher

Engineers

Sergio Fernández Rincón

Software Engineer

Alumni



Sara Guillén Torrado

Student

Sergio Tores Mora

Student

Former Members

Antonio Jesús Fernández-García

Researcher

Enrique Vílchez

Researcher

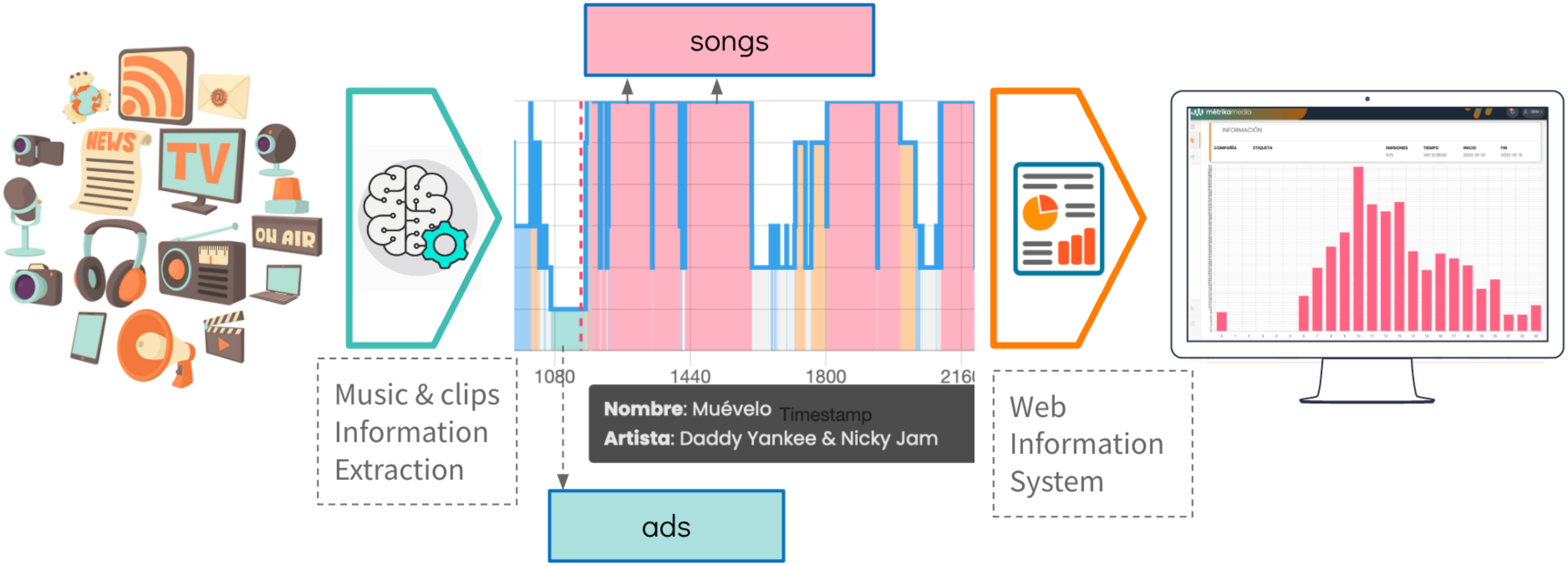
Fran Melchor

Researcher

Alejandro Fernández Camello

Researcher

Spin-off company

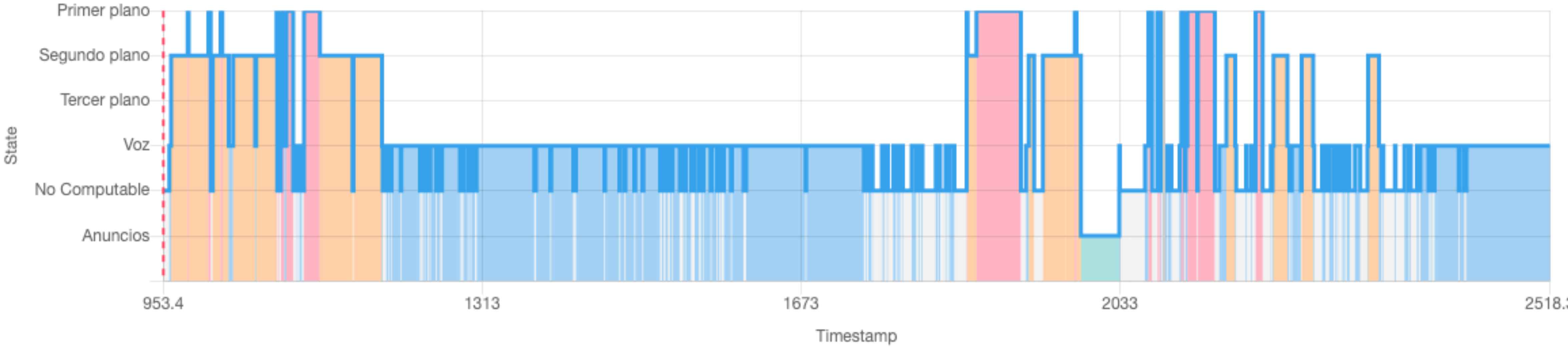


https://metrikamedia.com

métrikamedia

I3RM

Línea temporal: 00:00:00.00 (0.00)



Datos de la Medición

Resumen medición | Línea de tiempo | Obras Musicales | Publicidad

Categoría	Tiempo (seg)	Desconocido (seg)	% total	% desconocido
OBRAS MUSICALES EN PRIMER PLANO	283	10	7.85	0.28
OBRAS MUSICALES EN SEGUNDO PLANO	360	12	9.99	0.33
OBRAS MUSICALES EN TERCER PLANO	0	0	0.00	0.00
OBRAS MUSICALES NO COMPUTABLES	740		20.54	-

i3 LAB

Main projects

Software Engineering

Call	PID
Name	MDE4DSP
Start	01/01/2023
Duration	36 meses
Research team	9
Staff	2
Grant	**100K€**

José M. Conejero
Associate Professor

Alvaro E. Prieto
Associate Professor

Fran Melchor
Researcher

Deep Learning

Call	pending

Call	CPP
Name	MUSICGENIA
Start	01/01/2023
Duration	36 meses
Research team	4
Staff	2 (mtkm) + 2 (i3lab)
Grant	**400K€**

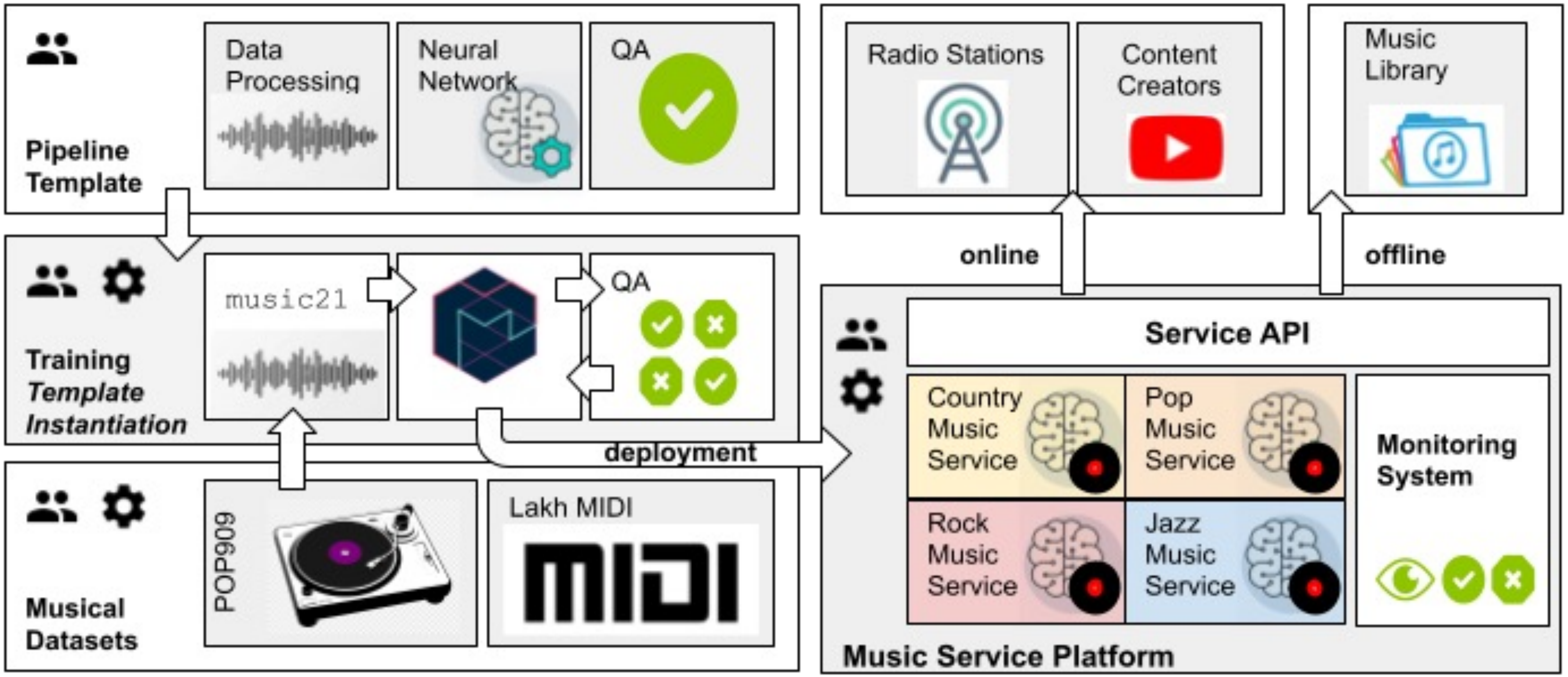
Juan D. Gutiérrez
Researcher

Emilio Delgado
Researcher

Roberto Rodriguez-Echeverria
Associate Professor

Fernando Sánchez-Figueroa
Associate Professor

Jorge Perianez
Researcher



Pipeline Template
Data Processing
Neural Network
QA
Training Template Instantiation
music21
QA
Musical Datasets
POP909
Lakh MIDI
MIDI
deployment
Radio Stations
Content Creators
Music Library
online
offline
Service API
Country Music Service
Pop Music Service
Rock Music Service
Jazz Music Service
Monitoring System
Music Service Platform

musicgenia

Let’s go back to the main topic…

A MDE approach for the Systematic Reproducibility and Replicability of Data Science Projects

Systematic Reproducibility and Replicability of Data Science Projects

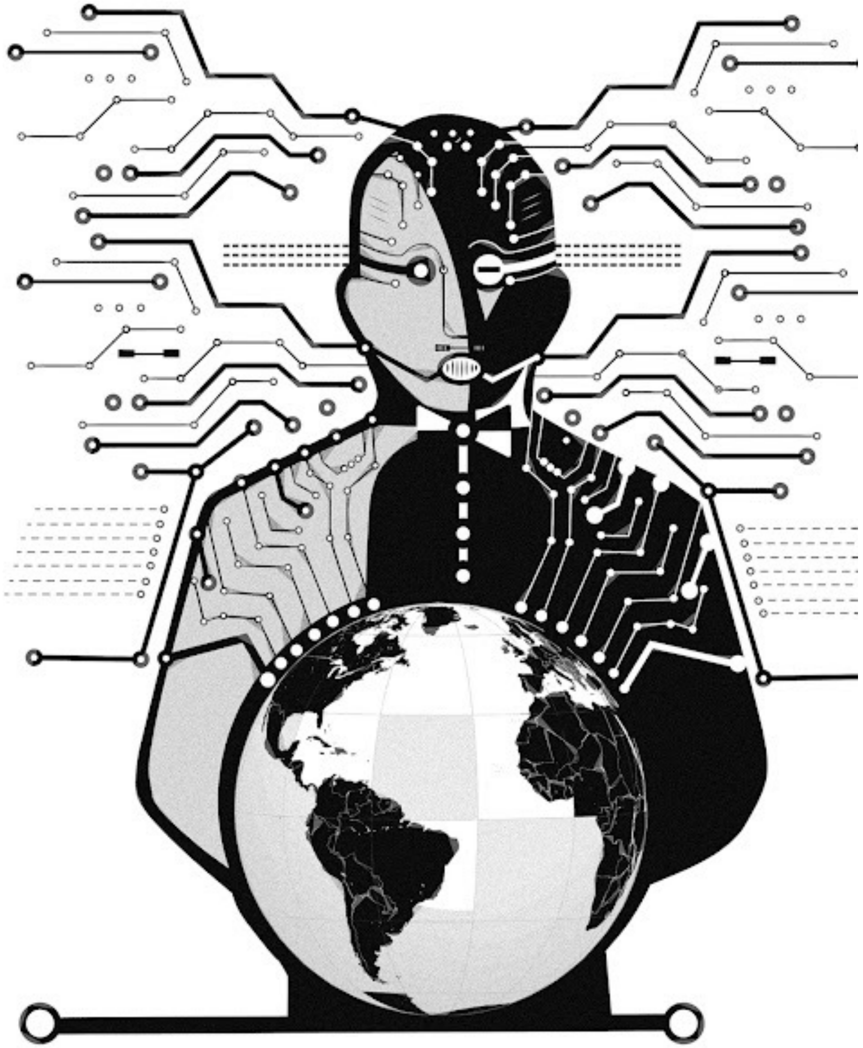
A MDE approach

Fran Melchor
Researcher

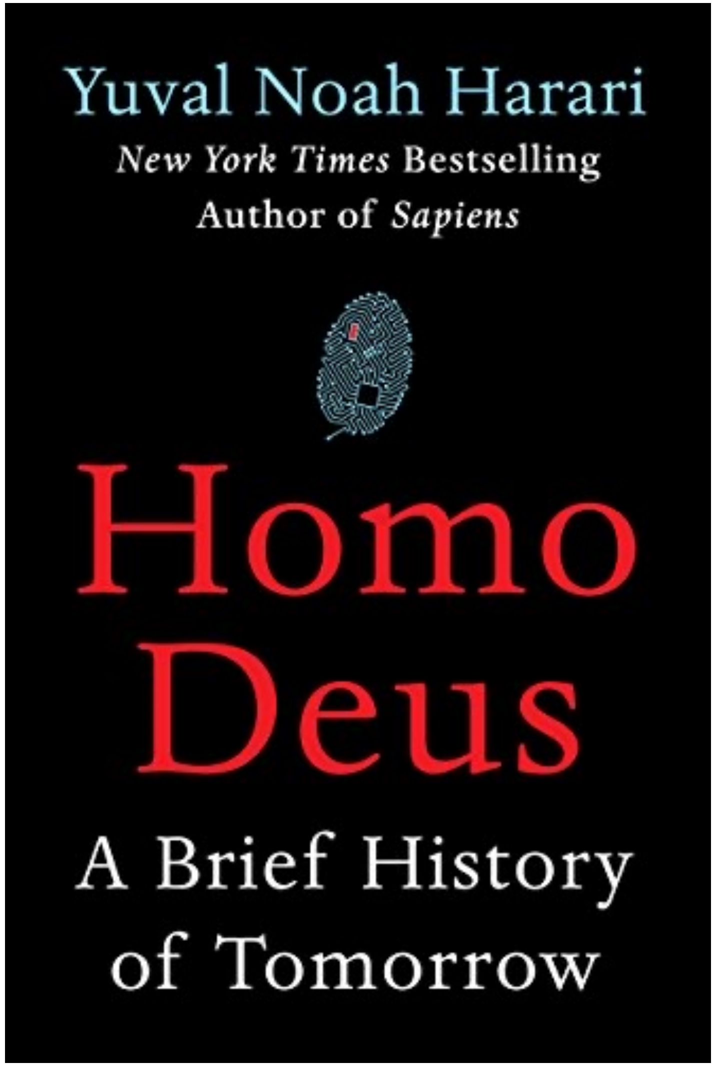
José M. Conejero
Associate Professor

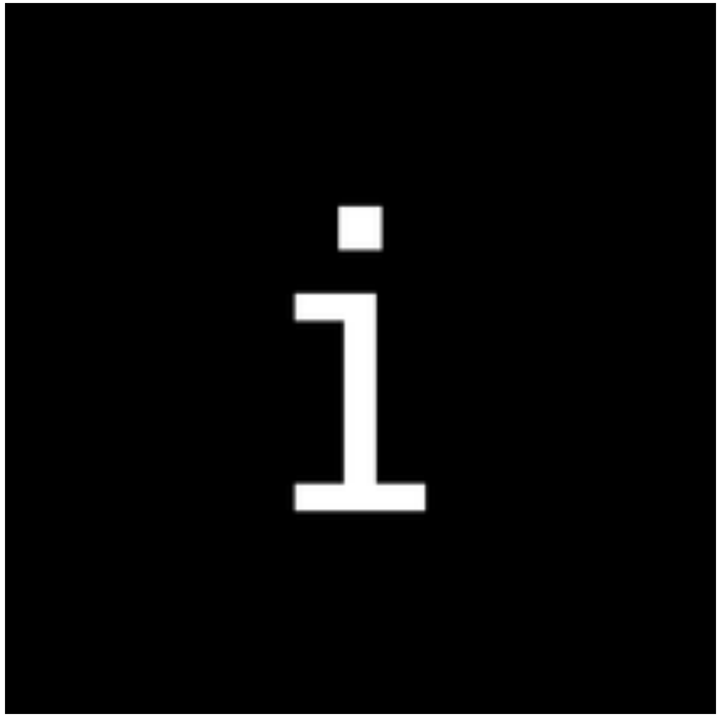
Alvaro E. Prieto
Associate Professor

All your (digital) **life** is **data**



All your (digital) **life** is **data**

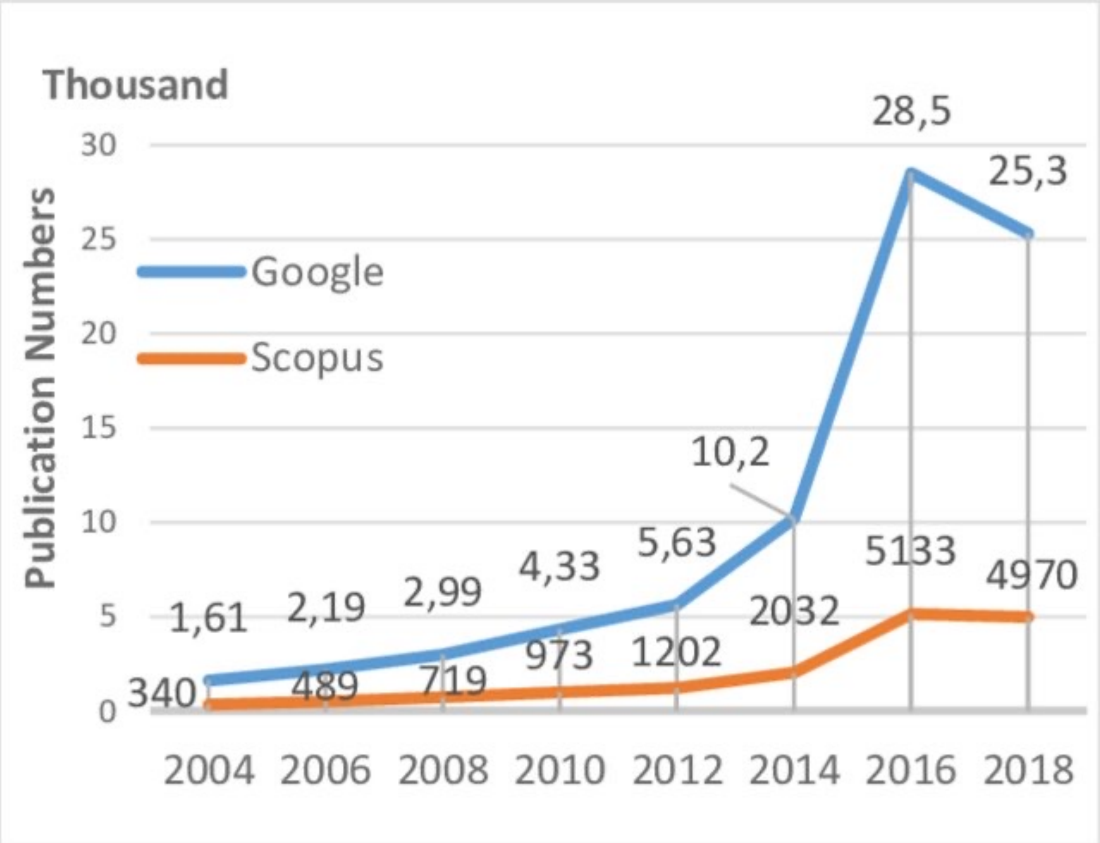




Data abundance & analysis tools as commodities

All your (digital) **life** is **data**

Lots of papers and experiments



Yapıcı, Mutlu & Tekerek, Adem & Topaloglu, Nurettin. (2019). Literature Review of Deep Learning Research Areas. 5. 188-215. 10.30855/gmbd.2019.03.01.

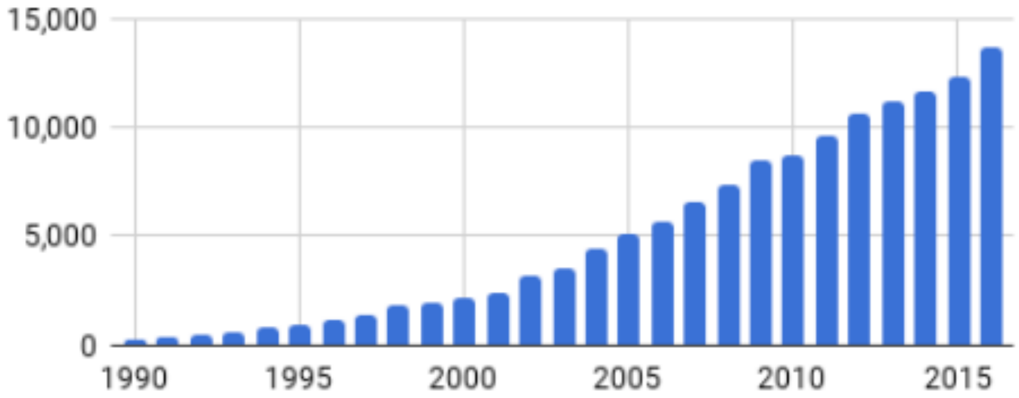


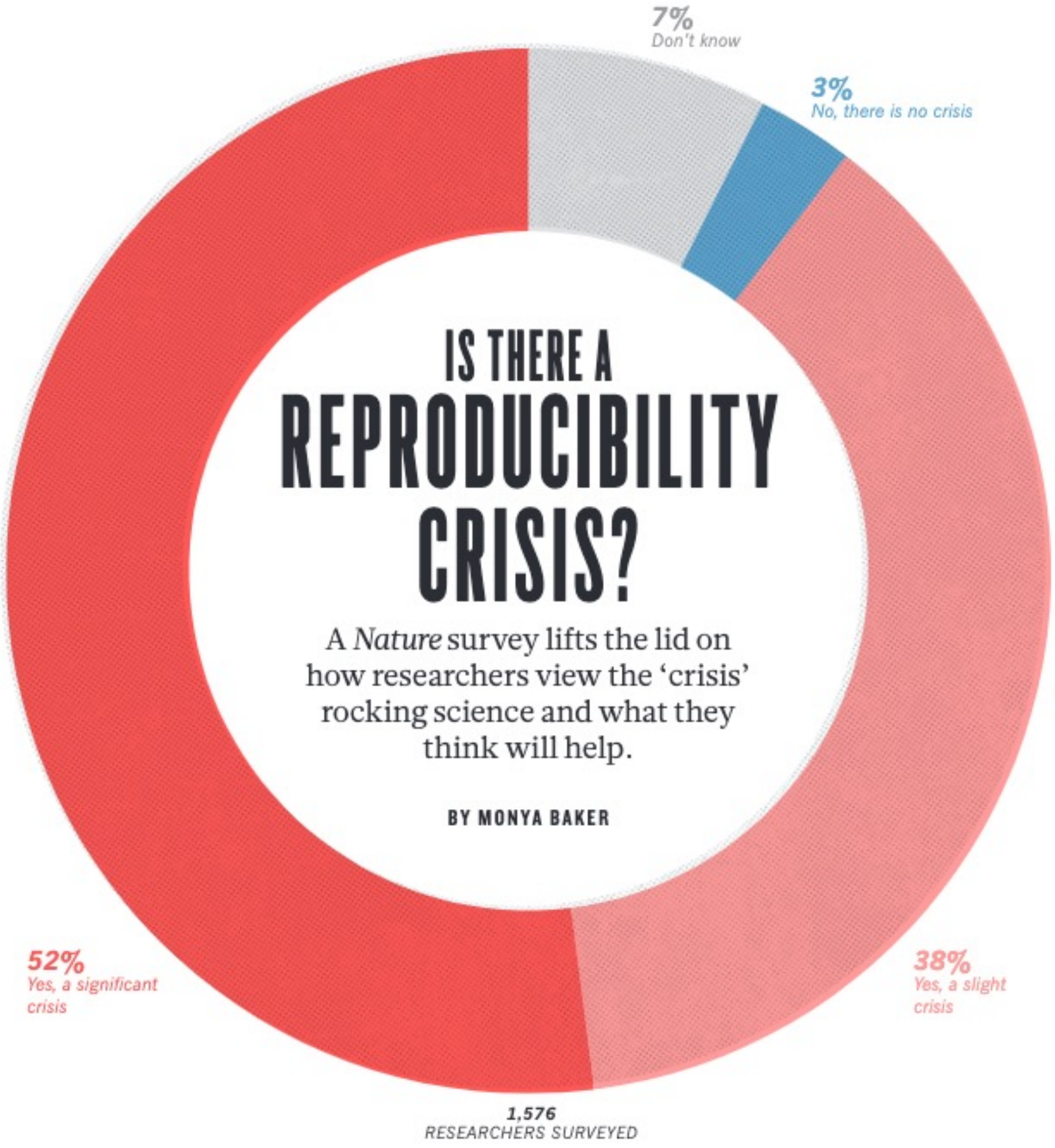
Figure 1: Growth of published reinforcement learning papers. Shown are the number of RL-related publications (y-axis) per year (x-axis) scraped from Google Scholar searches.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18). AAAI Press, Article 392, 3207–3214.

IS THERE A REPRODUCIBILITY CRISIS?

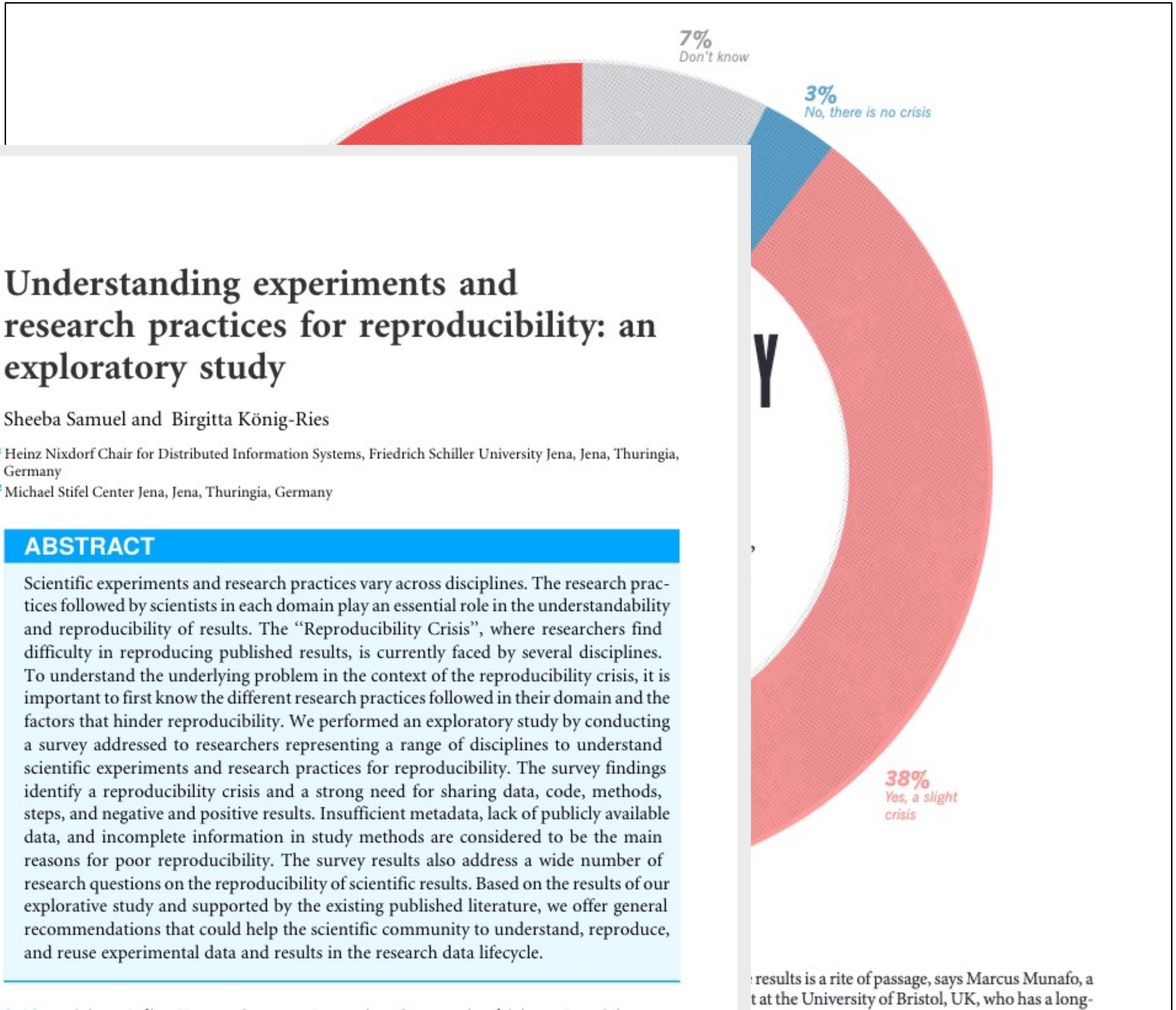
A *Nature* survey lifts the lid on how researchers view the 'crisis' rocking science and what they think will help.

BY MONYA BAKER



More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more

Failing to reproduce results is a rite of passage, says Marcus Munafo, a biological psychologist at the University of Bristol, UK, who has a long-



PeerJ

Understanding experiments and research practices for reproducibility: an exploratory study

Sheeba Samuel and Birgitta König-Ries

[1] Heinz Nixdorf Chair for Distributed Information Systems, Friedrich Schiller University Jena, Jena, Thuringia, Germany
[2] Michael Stifel Center Jena, Jena, Thuringia, Germany

ABSTRACT

Scientific experiments and research practices vary across disciplines. The research practices followed by scientists in each domain play an essential role in the understandability and reproducibility of results. The "Reproducibility Crisis", where researchers find difficulty in reproducing published results, is currently faced by several disciplines. To understand the underlying problem in the context of the reproducibility crisis, it is important to first know the different research practices followed in their domain and the factors that hinder reproducibility. We performed an exploratory study by conducting a survey addressed to researchers representing a range of disciplines to understand scientific experiments and research practices for reproducibility. The survey findings identify a reproducibility crisis and a strong need for sharing data, code, methods, steps, and negative and positive results. Insufficient metadata, lack of publicly available data, and incomplete information in study methods are considered to be the main reasons for poor reproducibility. The survey results also address a wide number of research questions on the reproducibility of scientific results. Based on the results of our explorative study and supported by the existing published literature, we offer general recommendations that could help the scientific community to understand, reproduce, and reuse experimental data and results in the research data lifecycle.

Submitted 9 September 2020

PeerJ

Understan…
research p…
explorator…

Sheeba Samuel and …

[1] Heinz Nixdorf Chair for D…
Germany
[2] Michael Stifel Center Jena, …

ABSTRACT

Scientific experime… tices followed by sci… and reproducibility… difficulty in reprod… To understand the … important to first k… factors that hinder … a survey addressed … scientific experime… identify a reprodu… steps, and negative … data, and incompl… reasons for poor r… research questions … explorative study and supported by the existing published literature, we offer general recommendations that could help the scientific community to understand, reproduce, and reuse experimental data and results in the research data lifecycle.

Submitted 9 September 2020

Matters arising

Transparency and reproducibility in artificial intelligence

https://doi.org/10.1038/s41586-020-2766-y

Received: 1 February 2020

Accepted: 10 August 2020

Published online: 14 October 2020

Check for updates

Benjamin Haibe-Kains[1,2,3,4,5], George Alexandru Adam[3,5], Ahmed Hosny[6,7], Farnoosh Khodakarami[1,2], Massive Analysis Quality Control (MAQC) Society Board of Directors*, Levi Waldron[8], Bo Wang[2,3,5,9,10], Chris McIntosh[2,5,9], Anna Goldenberg[3,5,11,12], Anshul Kundaje[13,14], Casey S. Greene[15,16], Tamara Broderick[17], Michael M. Hoffman[1,2,3,5], Jeffrey T. Leek[18], Keegan Korthauer[19,20], Wolfgang Huber[21], Alvis Brazma[22], Joelle Pineau[23,24], Robert Tibshirani[25,26], Trevor Hastie[25,26], John P. A. Ioannidis[25,26,27,28,29], John Quackenbush[30,31,32] & Hugo J. W. L. Aerts[6,7,33,34]

ARISING FROM S. M. McKinney et al. *Nature* https://doi.org/10.1038/s41586-019-1799-6 (2020)

Breakthroughs in artificial intelligence (AI) hold enormous potential as it can automate complex tasks and go even beyond human performance. In their study, McKinney et al.[1] showed the high potential of AI for breast cancer screening. However, the lack of details of the methods and algorithm code undermines its scientific value. Here, we identify obstacles that hinder transparent and reproducible AI research as faced by McKinney et al.[1], and provide solutions to these obstacles with implications for the broader field.

The work by McKinney et al.[1] demonstrates the potential of AI in medical imaging, while highlighting the challenges of making such work reproducible. The authors assert that their system improves the …

reporting-standards). Publication of insufficiently documented research does not meet the core requirements underlying scientific discovery[2,3]. Merely textual descriptions of deep-learning models can hide their high level of complexity. Nuances in the computer code may have marked effects on the training and evaluation of results[4], potentially leading to unintended consequences[5]. Therefore, transparency in the form of the actual computer code used to train a model and arrive at its final set of parameters is essential for research reproducibility. McKinney et al.[1] stated that the code used for training the models has "a large number of dependencies on internal tooling, infrastructure and hardware", and claimed that the release of the code was there-…

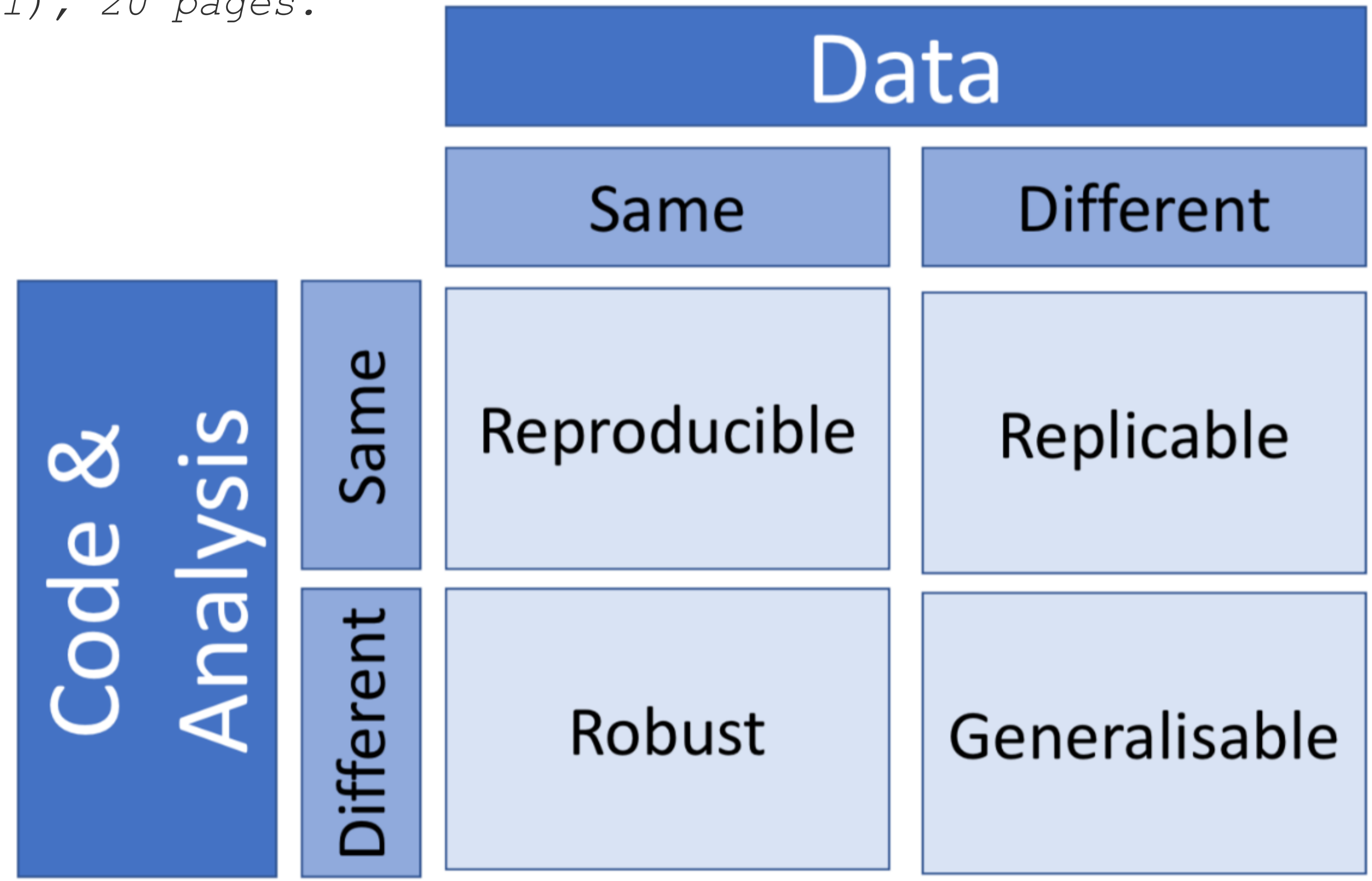
…results is a rite of passage, says Marcus Munafo, a …t at the University of Bristol, UK, who has a long-

1. Lack of access to the same training data
2. Misspecification or under-specification of the model or training procedure
3. Code availability or code with errors
4. Infrastructure complexity:

 ”a large number of dependencies on internal tooling, infrastructure and hardware”

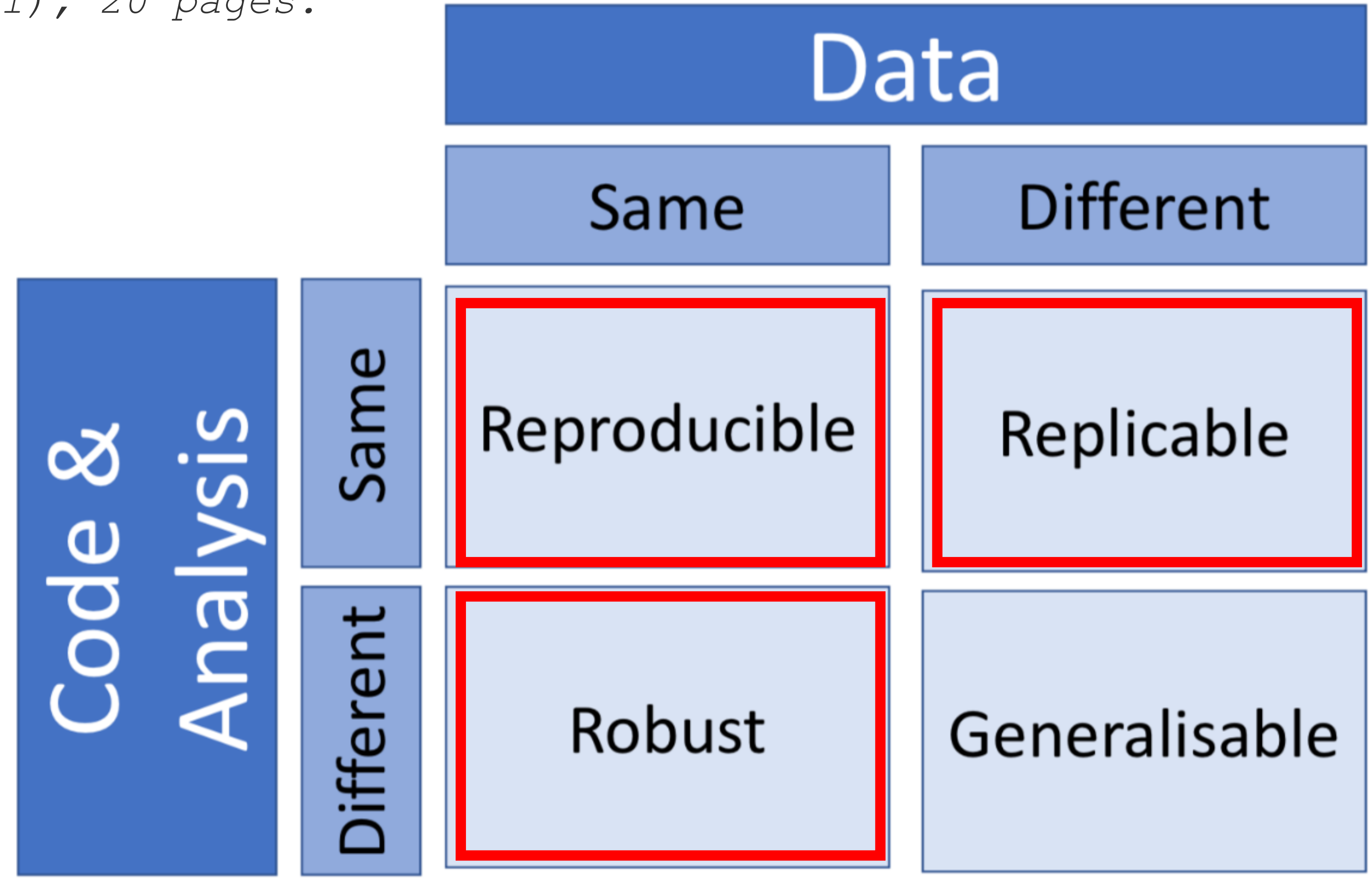
Although sharing of code and data are widely seen as a crucial part of scientific research, the adoption varies across fields.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché-Buc, Emily Fox, and Hugo Larochelle. 2021. ***Improving reproducibility in machine learning research*** *(a report from the NeurIPS 2019 reproducibility program). J. Mach. Learn. Res. 22, 1, Article 164 (January 2021), 20 pages.*



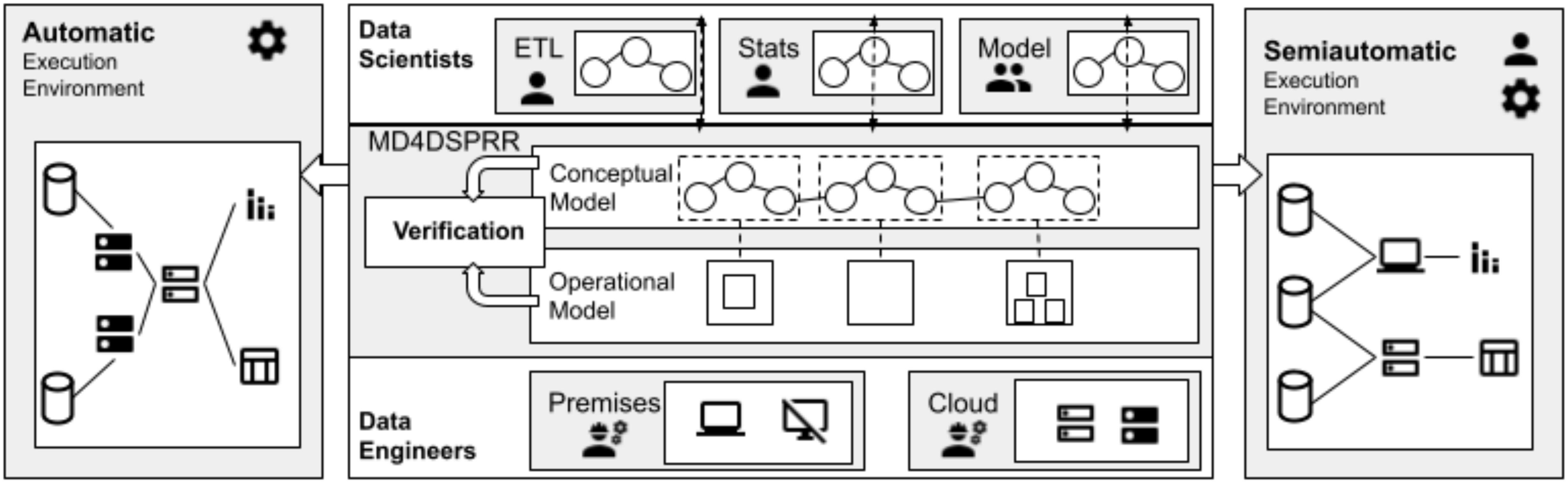
https://github.com/WhitakerLab/ReproducibleResearch

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché-Buc, Emily Fox, and Hugo Larochelle. 2021. ***Improving reproducibility in machine learning research*** *(a report from the NeurIPS 2019 reproducibility program). J. Mach. Learn. Res. 22, 1, Article 164 (January 2021), 20 pages.*

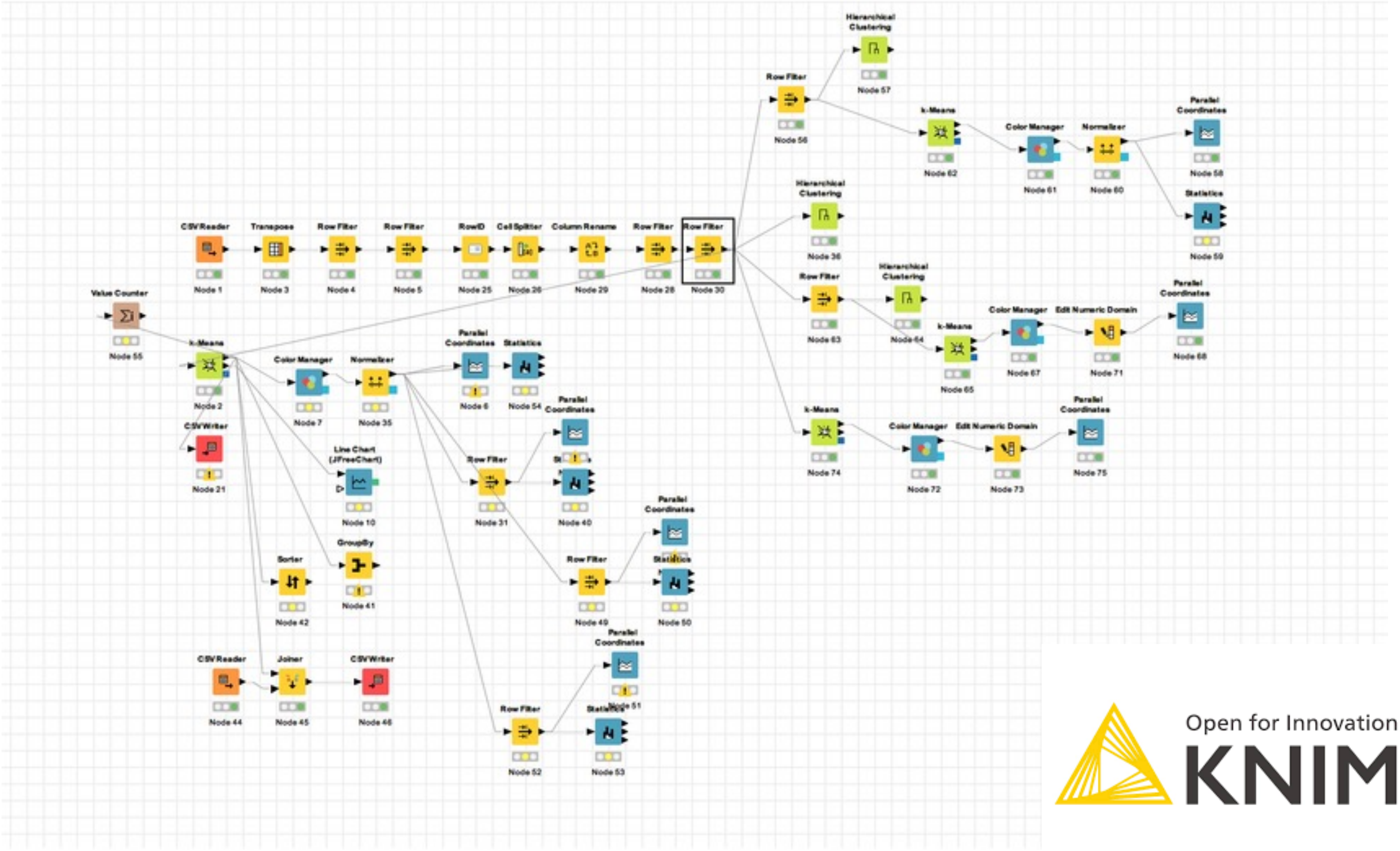


https://github.com/WhitakerLab/ReproducibleResearch

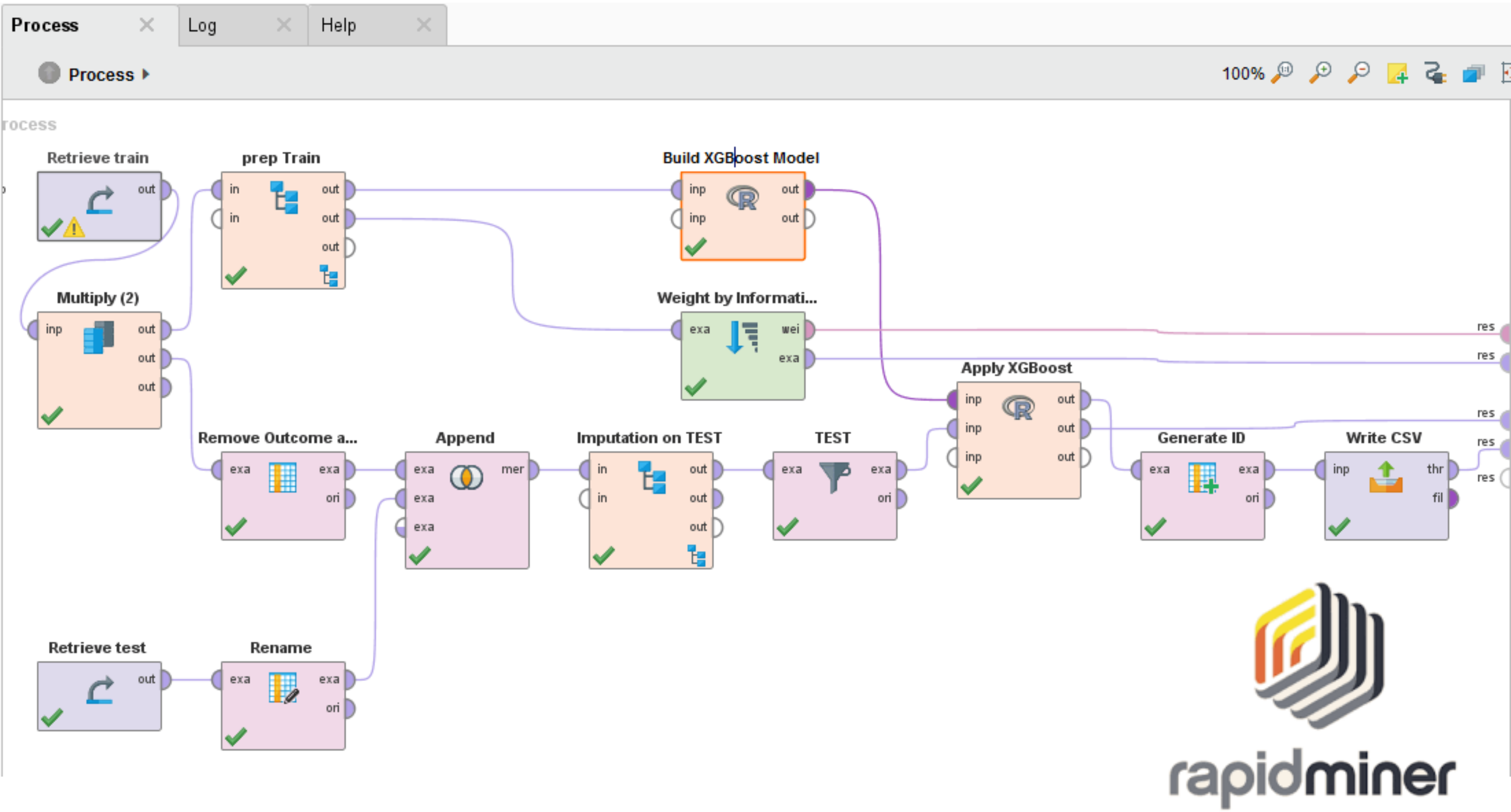
MDE framework

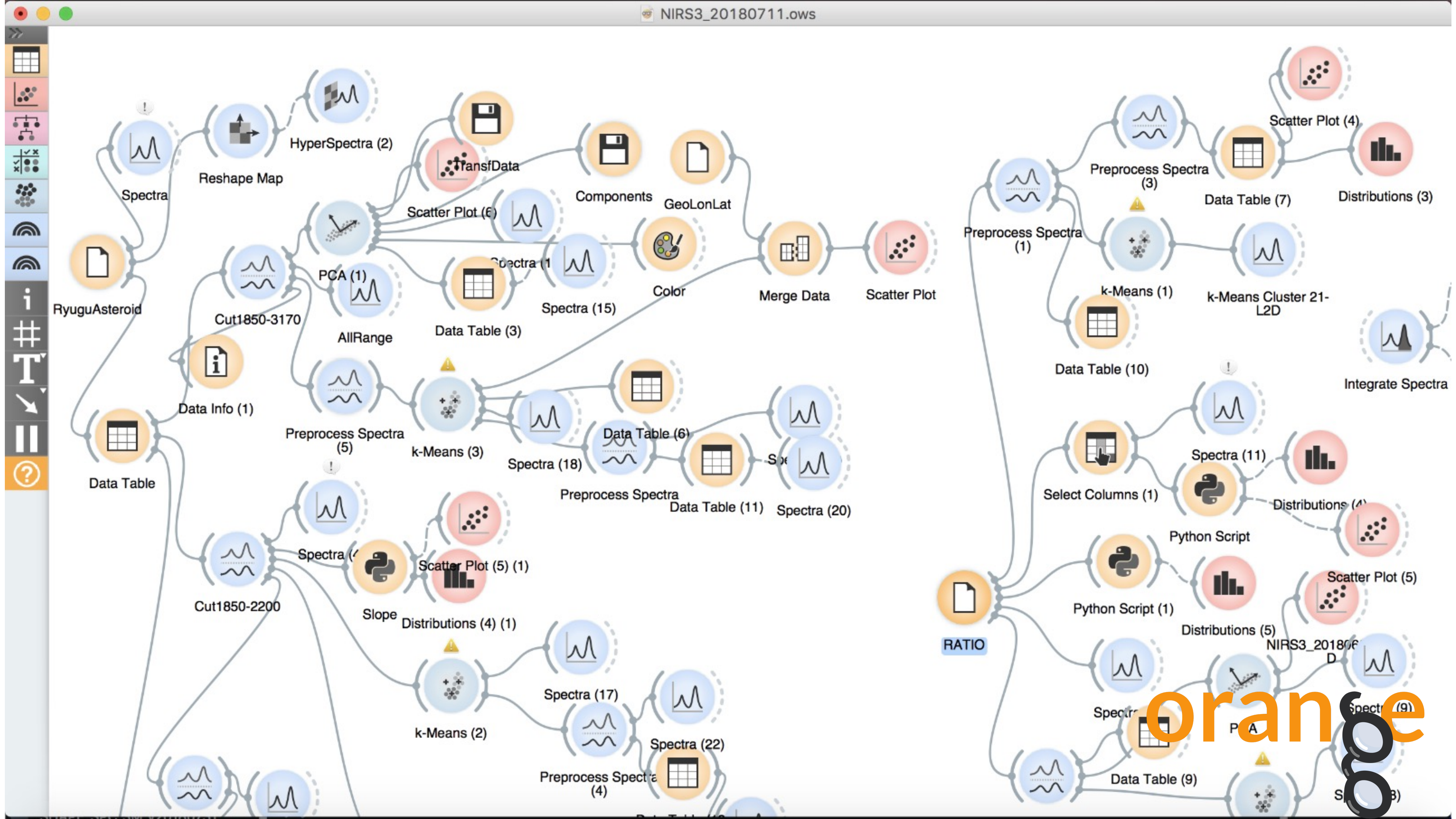


Conceptual level



Open for Innovation
KNIME


Process
Log
Help
Process
100%
Retrieve train
prep Train
Build XGBoost Model
Multiply (2)
Weight by Informati...
Apply XGBoost
Remove Outcome a...
Append
Imputation on TEST
TEST
Generate ID
Write CSV
Retrieve test
Rename
rapidminer


NIRS3_20180711.ows
HyperSpectra (2)
Reshape Map
Spectra
RyuguAsteroid
TransfData
Scatter Plot (6)
Components
GeoLonLat
PCA (1)
Cut1850-3170
AllRange
Data Table (3)
Spectra (15)
Color
Merge Data
Scatter Plot
Data Info (1)
Preprocess Spectra (5)
k-Means (3)
Spectra (18)
Data Table (6)
Preprocess Spectra
Data Table (11)
Spectra (20)
Data Table
Cut1850-2200
Slope
Scatter Plot (5) (1)
Distributions (4) (1)
k-Means (2)
Spectra (17)
Spectra (22)
Preprocess Spectra (4)
Preprocess Spectra (3)
Scatter Plot (4)
Data Table (7)
Distributions (3)
Preprocess Spectra (1)
k-Means (1)
k-Means Cluster 21-L2D
Data Table (10)
Integrate Spectra
Select Columns (1)
Spectra (11)
Python Script
Python Script (1)
Distributions (5)
Scatter Plot (5)
RATIO
Data Table (9)
orange

FourthWeekAssignmentFinal.py

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Dec  1 13:41:23 2015

@author: Ossaio
"""

# Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker

# Library used to manipulate output files in excel
import openpyxl

# Read in the marscrater csv dataset
data = pd.read_csv('marscrater_pds.csv', low_memory=False)

# Bug fix for display formats to avoid run time errors. Sets outputs to 2 decimal places
pd.set_option('display.float_format', lambda x:'%.2f'%x)

# Setting variables in the working dataset to numeric
data['NUMBER_LAYERS'] = pd.to_numeric(data['NUMBER_LAYERS'], errors='coerce')
data['DIAM_CIRCLE_IMAGE'] = pd.to_numeric(data['DIAM_CIRCLE_IMAGE'], errors='coerce')
data['DEPTH_RIMFLOOR_TOPOG'] = pd.to_numeric(data['DEPTH_RIMFLOOR_TOPOG'], errors='coerce')
data['LATITUDE_CIRCLE_IMAGE'] = pd.to_numeric(data['LATITUDE_CIRCLE_IMAGE'], errors='coerce')
data['LONGITUDE_CIRCLE_IMAGE'] = pd.to_numeric(data['LONGITUDE_CIRCLE_IMAGE'], errors='coerce')

```

Predictive Model Markup Language (PMML)

PMML 4.4 - General Structure

For Time Series models, please see the Notice of Essential Claims.

PMML uses XML to represent mining models. The structure of the models is described by an XML Schema. One or more mining models can be contained in a PMML document. A PMML document is an XML document with a root element of type PMML. The general structure of a PMML document is:

```
<?xml version="1.0"?>
<PMML version="4.4"
  xmlns="https://www.dmg.org/PMML-4_4"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Header copyright="Example.com"/>
  <DataDictionary> ... </DataDictionary>

  ... a model ...

</PMML>
```

The namespaces in the PMML Schema itself are defined as:

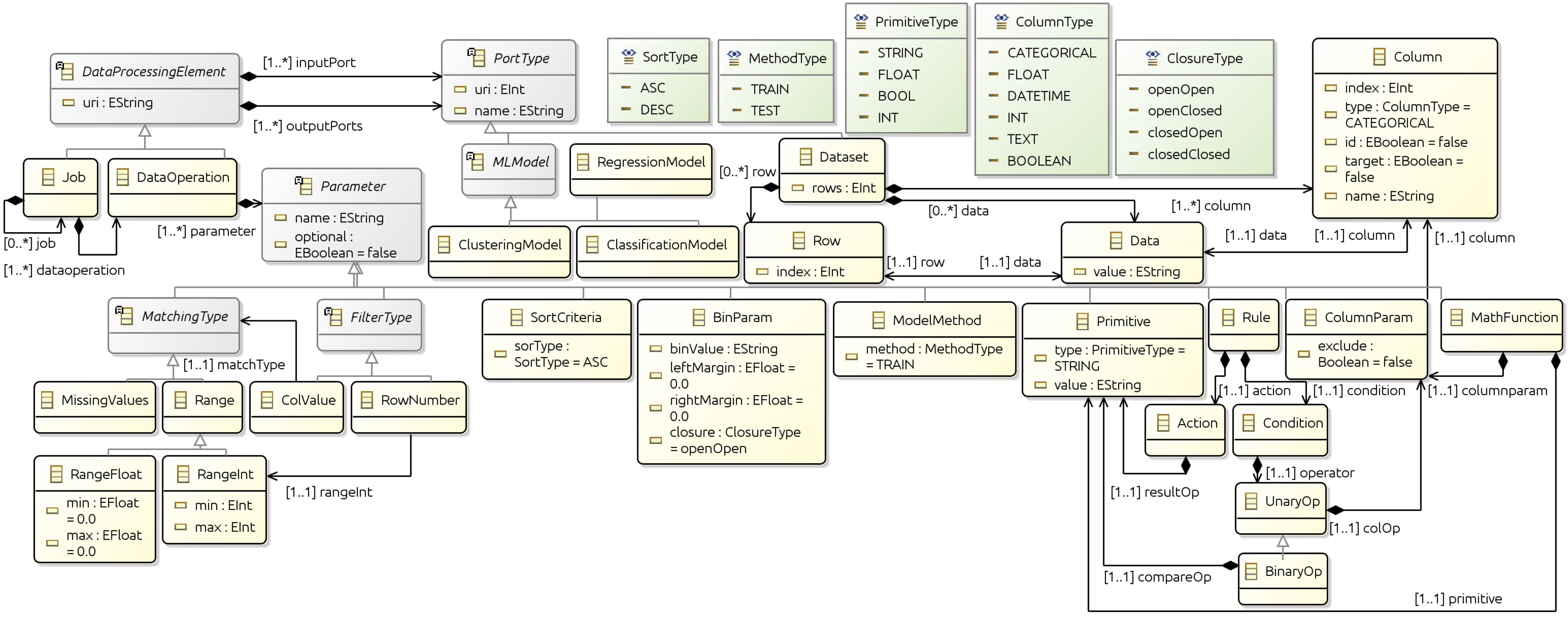
PMML4.4 Menu

Home

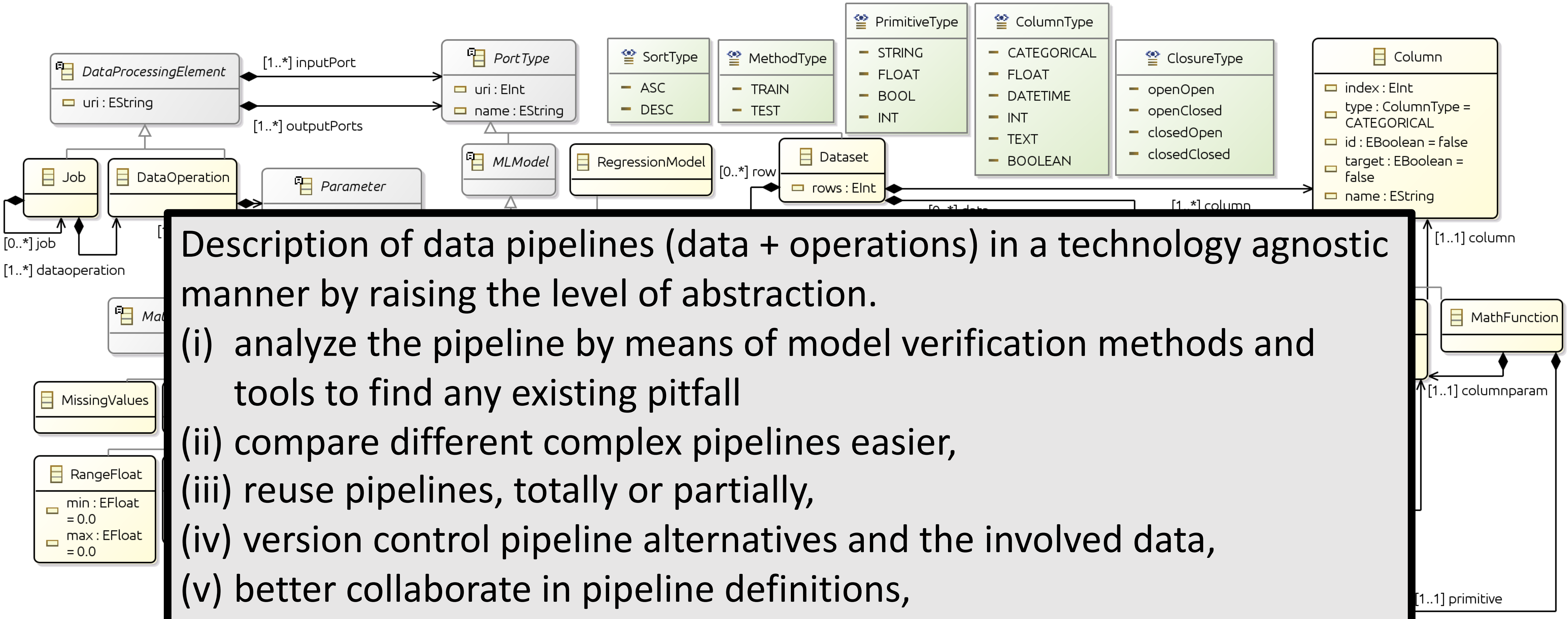
Changes
XML Schema

Conformance
Interoperability
General Structure
Field Scope
Header
Data Dictionary
Mining Schema
Transformations
Statistics
Taxomony
Targets

Conceptual metamodel



Conceptual metamodel



Description of data pipelines (data + operations) in a technology agnostic manner by raising the level of abstraction.

(i) analyze the pipeline by means of model verification methods and tools to find any existing pitfall

(ii) compare different complex pipelines easier,

(iii) reuse pipelines, totally or partially,

(iv) version control pipeline alternatives and the involved data,

(v) better collaborate in pipeline definitions,

(vi) import/export from/to different tools: common language

2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)

Pitfalls Analyzer: Quality Control for Model-Driven Data Science Pipelines

Gopi Krishnan Rajbahadur, Queen's University, Kingston, Canada, krishnan@cs.queensu.ca
Gustavo Ansaldi Oliva, Queen's University, Kingston, Canada, gustavo@cs.queensu.ca
Ahmed E. Hassan, Queen's University, Kingston, Canada, ahmed@cs.queensu.ca
Juergen Dingel, Queen's University, Kingston, Canada, dingel@cs.queensu.ca

***Abstract*—Data science pipelines are a sequence of data processing steps that aim to derive knowledge and insights from raw data. Data science pipeline tools simplify the creation and automation of data science pipelines by providing reusable building blocks that users can drag and drop into their pipelines. Such a graphical, model-driven approach enables users with limited data science expertise to create complex pipelines. However, recent studies show that there exist several data science pitfalls that can yield spurious results and, consequently, misleading insights. Yet, none of the popular pipeline tools have built-in quality control measures to detect these pitfalls. Therefore, in this paper, we propose an approach called Pitfalls Analyzer to detect common pitfalls in data science pipelines. As a proof-of-concept, we implemented a prototype of the Pitfalls Analyzer for KNIME, which is one of the most popular data science pipeline tools. Our prototype is model-driven, since the detection of pitfalls is accomplished using pipelines that were created with KNIME building blocks. To showcase the effectiveness of our approach, we run our prototype on 11 pipelines that were created by KNIME experts for 3 Internet-of-Things (IoT) projects. The results indicate that our prototype flags all and only those instances of the pitfalls that we were able to flag while manually inspecting the pipelines.**

***Index Terms*—Data science pipelines, model-driven engineering, quality control, data science pitfalls**

I. Introduction

Data science is the science of extracting knowledge and insights from the data. Data science pipelines are sequences of processing and analytic steps that are applied on data to extract such knowledge and insights. These data science pipelines are being widely used in various industries [1, 2, 3, 4] towards diverse use-cases. For instance, GE [5], SAP [6], Bosch [2], and Siemens [7] use a variety of data science pipelines to ad- tools include Microsoft Azure Machine Learning Studio [11], IBM ThingWorx [12], Verizon ThingSpace [13], KNIME [14], Weka [15], and RapidMiner Studio [16]. Such pipeline tools leverage domain-specific graphical modeling languages (DSL) to enable the specification of data science pipelines. From a practical perspective, users specify pipelines by interconnecting building-blocks using graphical components that are provided by the tool. An example of a data science pipeline that is designed in KNIME is shown in Figure 3. Once the specification of the pipeline is completed, the pipeline tool automatically generates the low-level code that enables the execution of the pipeline. In other words, pipeline tools transform a user-specified data science pipeline into executable code (Figure 1). Ultimately, these tools enable users with limited data science and programming expertise to implement their data science pipelines with ease [17].

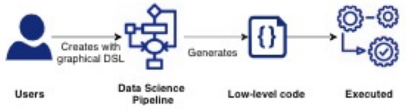
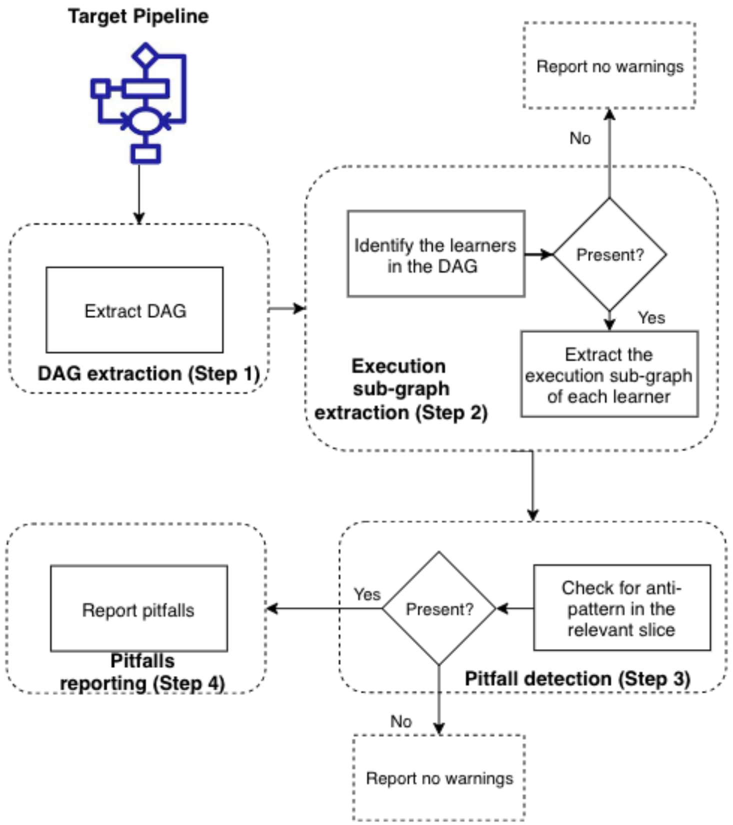


Fig. 1. Code generation in data science pipeline tools.

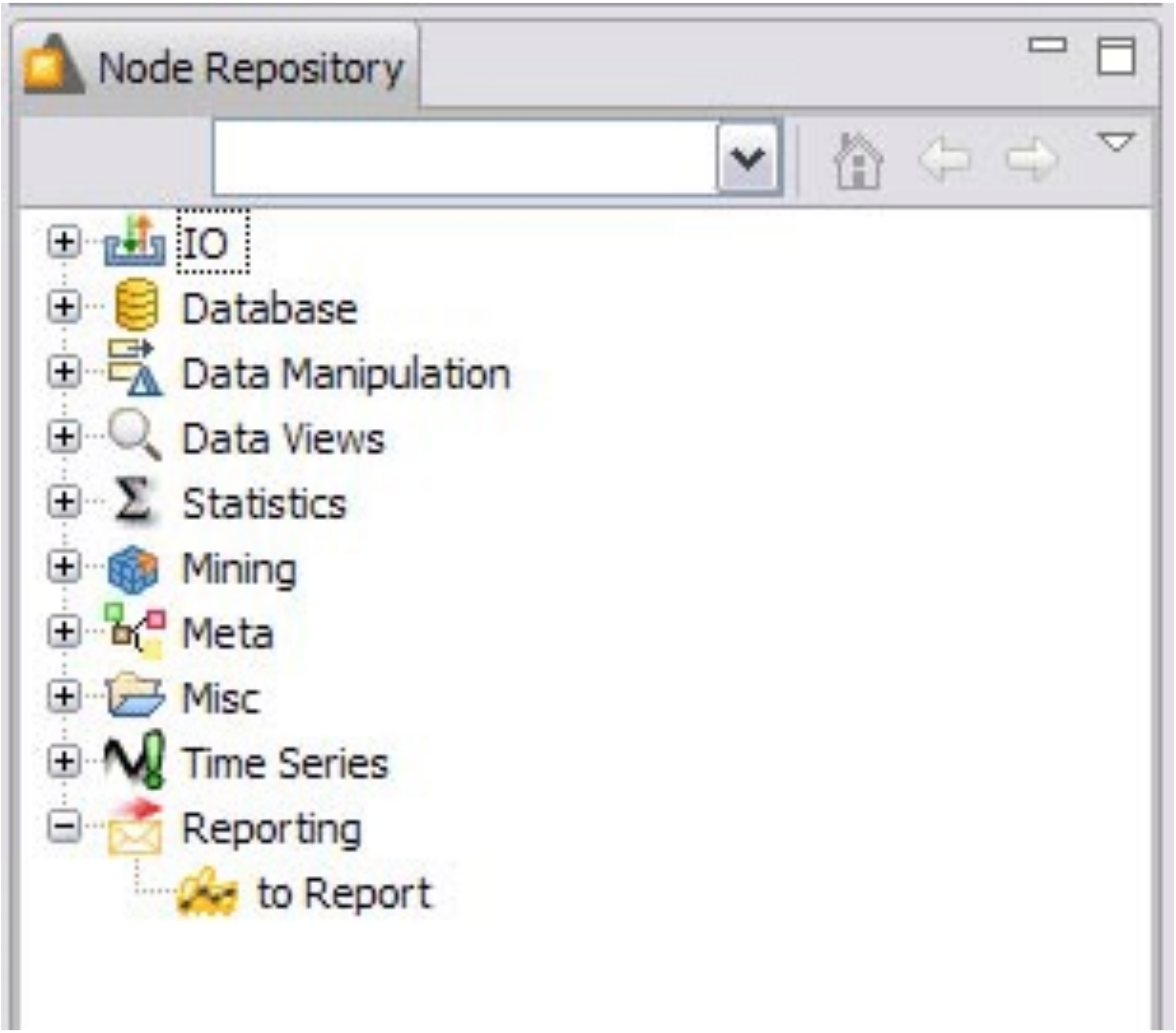
Unfortunately, data science pipeline tools encapsulate a serious problem. While they enable users with limited knowledge to create and automate pipelines easily, these tools do not offer ways to ensure the quality of the created pipelines and the generated results. Yet, prior research has shown that users



Pitfall	Anti-pattern
P1	Absence of control metrics in the dataset used
P2	Absence of nodes to removing correlated variables
P3	Presence of class rebalancer nodes and subsequent use of threshold dependent measure computation nodes
P4	Absence of parameter optimization nodes
P5	Absence of threshold independent performance measure nodes and presence of threshold dependent performance measure computation nodes
P6	Presence of cross validation nodes and absence of bootstrap validation nodes
P7	Presence of Type-1 ANOVA computation nodes being fed by the learner's coefficients
P8	Extraction of coefficients of a learner node

Reusability and extensibility

Data operations library



Reusability and extensibility

Data operations library

Data operations mapping

Data operations definition

Data operations mapping

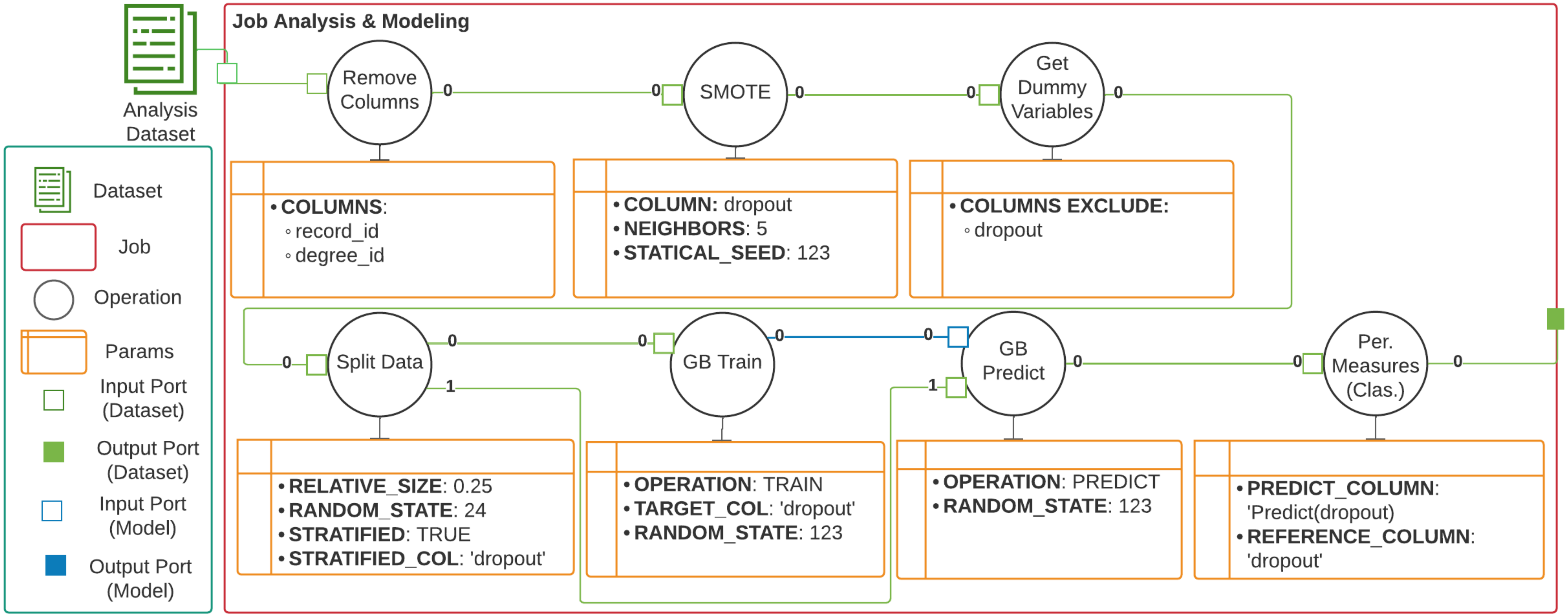
> 400

MD4DSPRR	KNIME	Orange	Python	PMML	RapidMiner
RemoveColumns	Column Filter	Select Columns	Drop(pandas)	Yes	Remove Attribute Range
ImbalancedLearn	SMOTE	–	SMOTE(imblearn)	No	SMOTE Upsampling
OneHot	One to Many	Continuize	Get_dummies(pandas)	Yes	Nominal to Numerical
Split	Partitioning	Data Sampler	Train_test_split(sklearn)	No	Split Data
StringToNumber	String to Number	–	To_numeric(pandas)	No	Parse Numbers
RowFilter	Row Filter	Filter	Filter(pandas)	No	Filter Examples
ConditionalFunction	Rule Engine	–	Apply (pandas)	Yes	–
Concatenate	Concatenate	Concatenate	Concat (pandas)	No	Append

Data operations definition

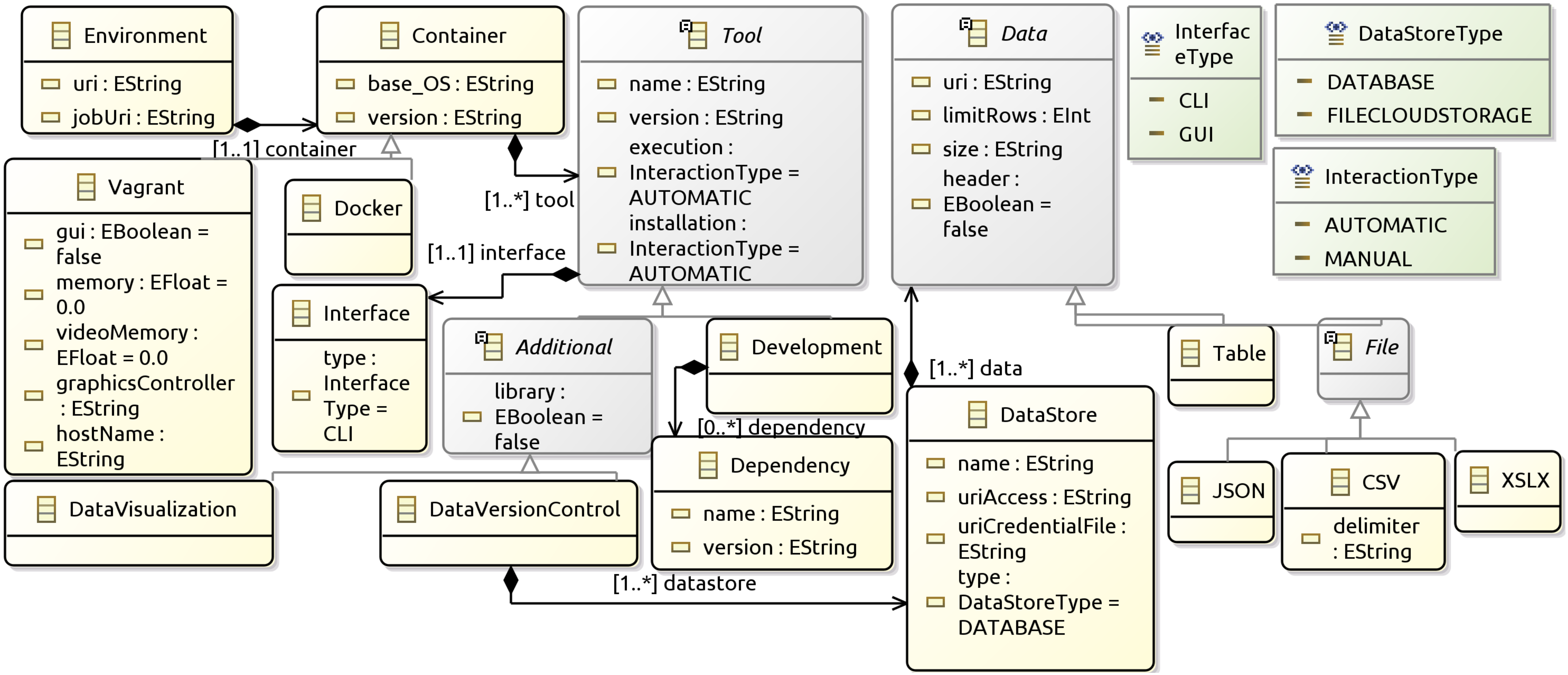
Data Operation	Input Ports	Parameters	Output Ports
RemoveColumns	Dataset	cols:Set(ColumnParam)	Dataset
ImbalancedLearn	Dataset	col:ColumnParam	Dataset
OneHot	Dataset	cols:Set(ColumnParam)	Dataset
Split	Dataset	size: Primitive(Float), stratified: Primitive(Bool), col:ColumnParam	Dataset 1, Dataset 2
StringToNumber	Dataset	col:ColumnParam	Dataset
RowFilter	Dataset	filterType:FilterType, col:ColumnParam, matchingType:MatchingType, include:Primitive(Bool)	Dataset
ConditionalFunction	Dataset	rules:Set(Rule)	Dataset
Concatenate	Dataset 1 Dataset 2	–	Dataset

Pipeline example

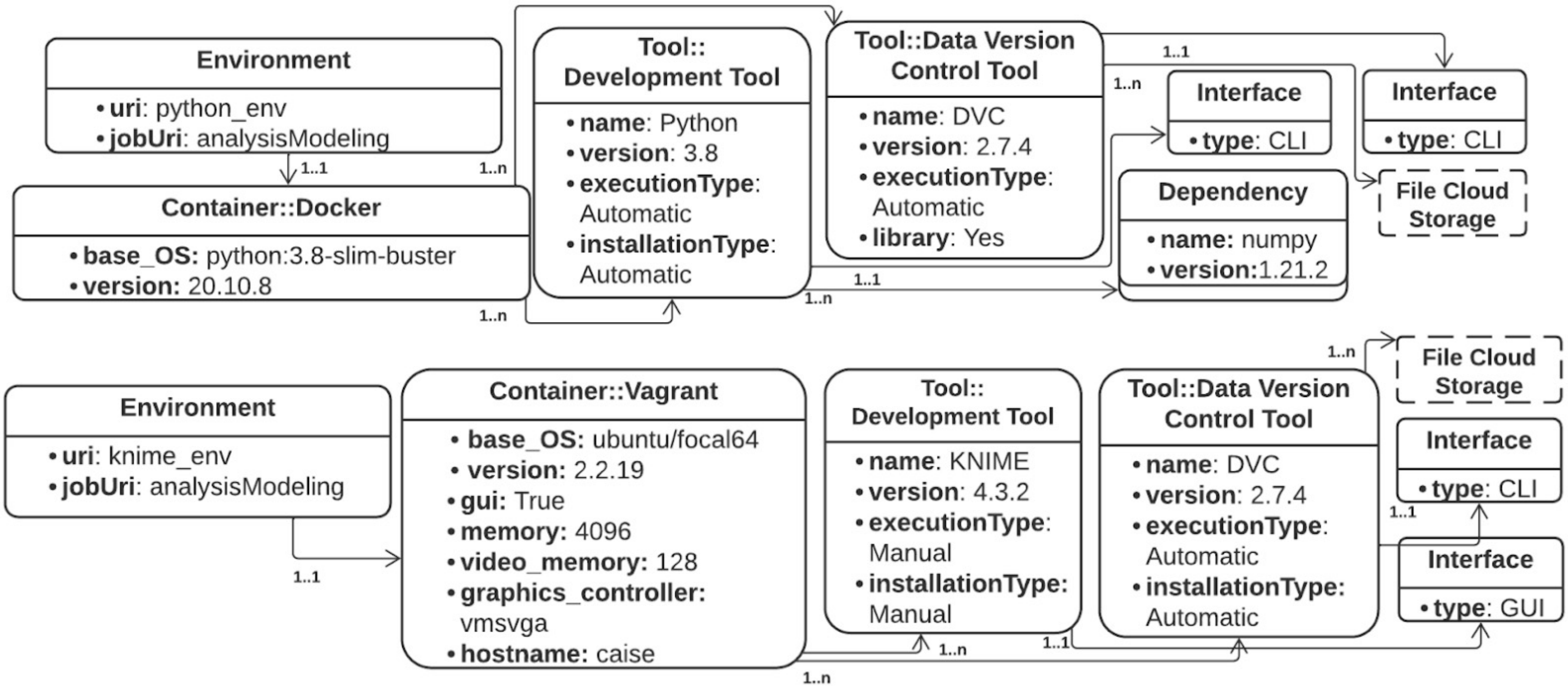


Operational level

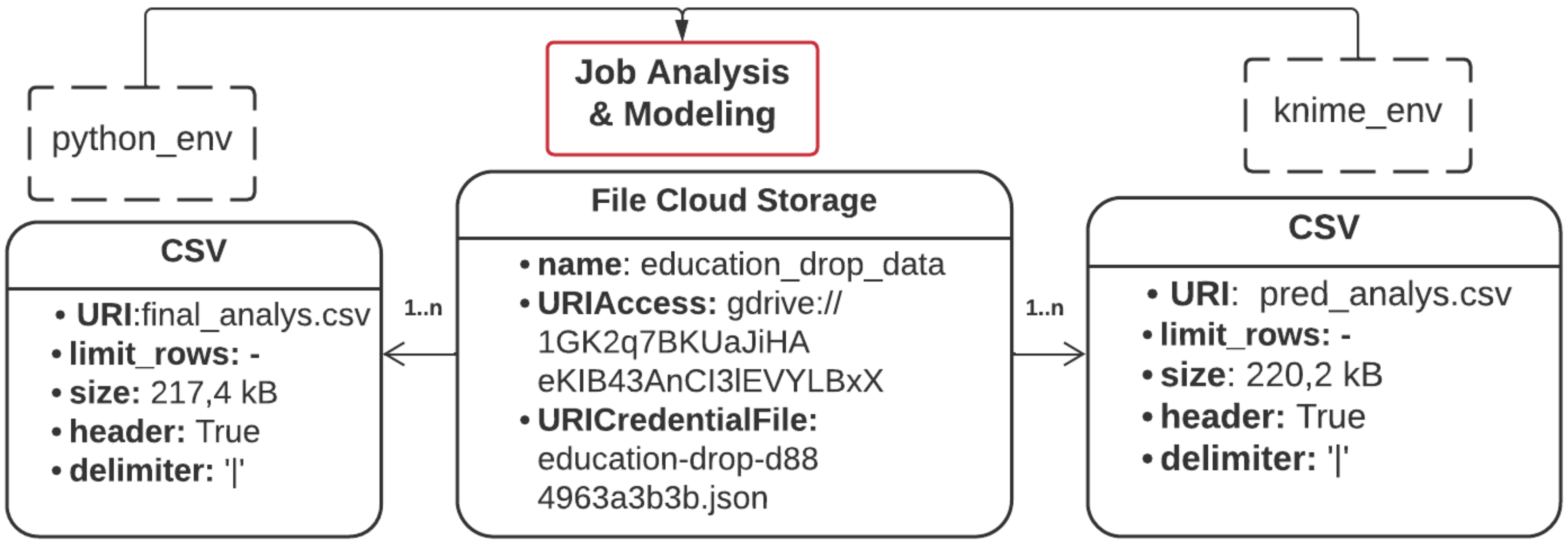
Operational metamodel



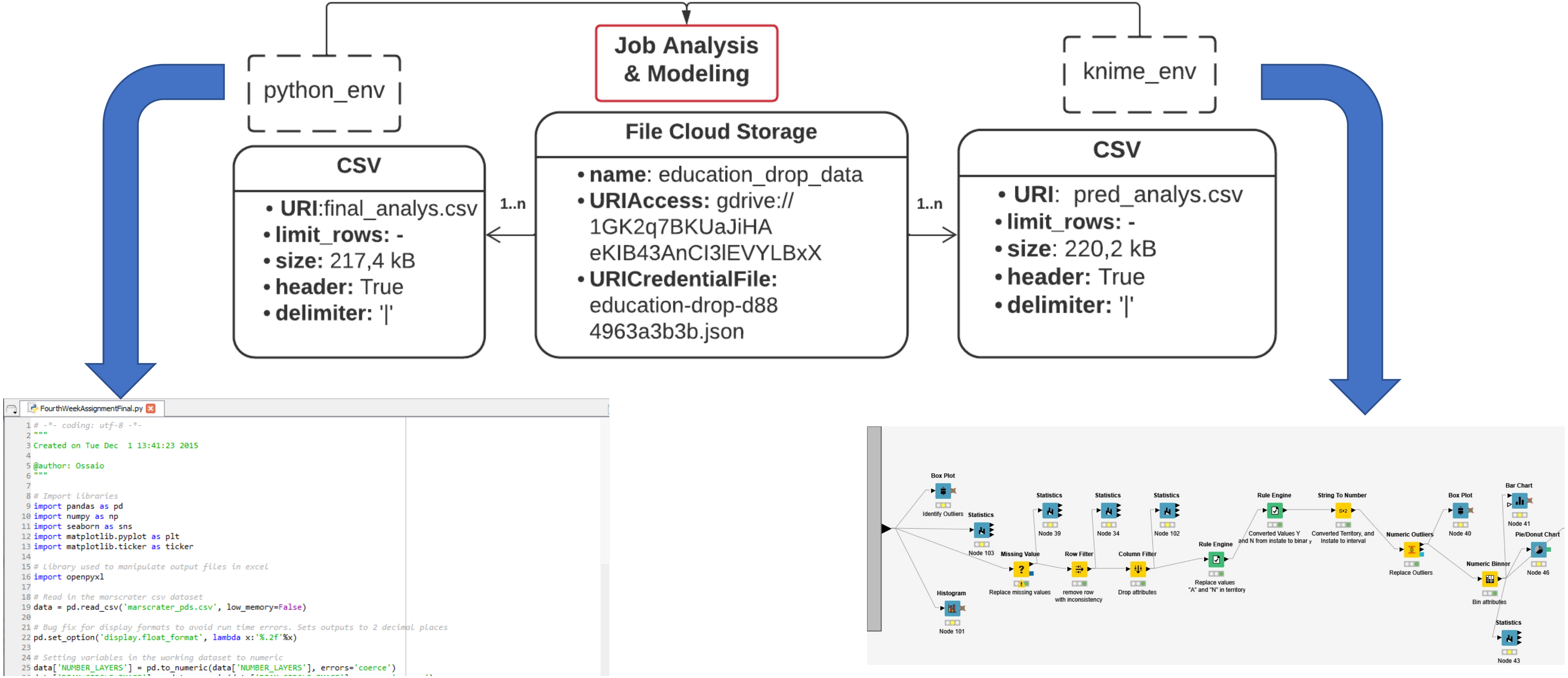
Environments



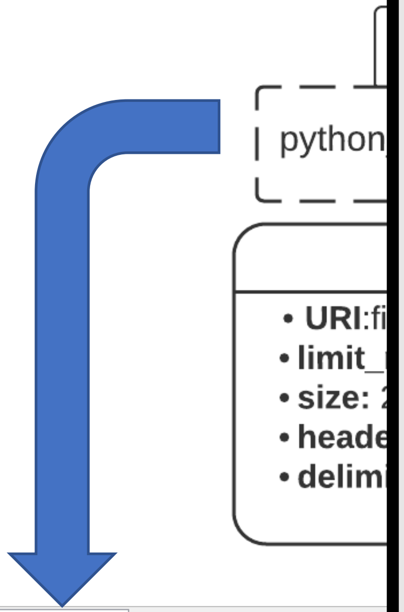
Data storage



M2T Transformation: code generation

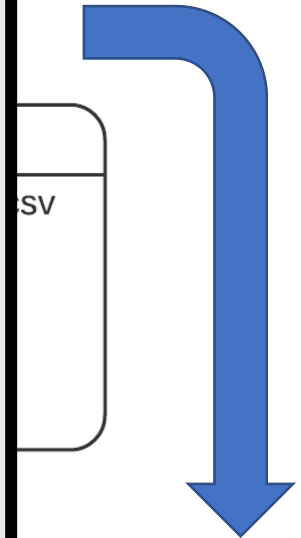


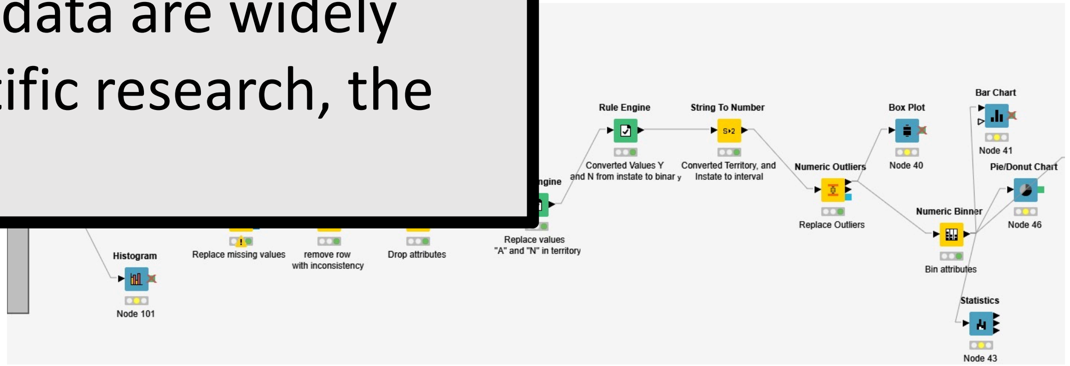
M2T Transformation: code generation



1. Lack of access to the same training data
2. Misspecification or under-specification of the model or training procedure
3. Code availability or code with errors
4. Infrastructure complexity:
 ”a large number of dependencies on internal tooling, infrastructure and hardware”

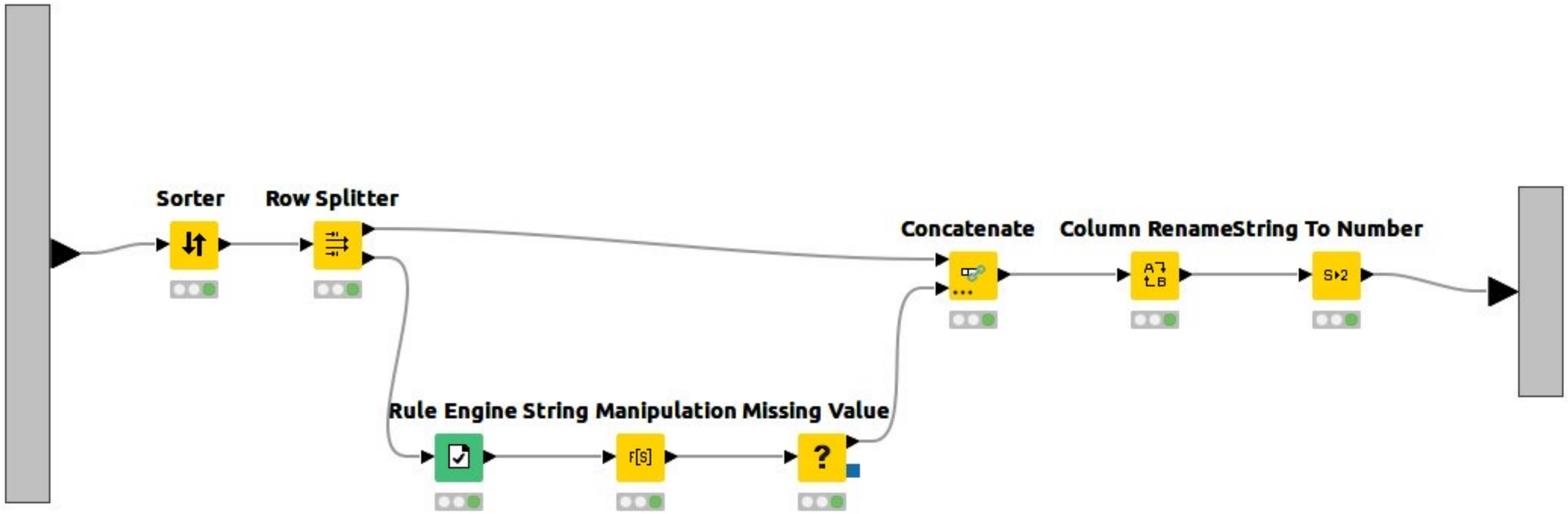
Although sharing of code and data are widely seen as a crucial part of scientific research, the adoption varies across fields.





Runtime verification

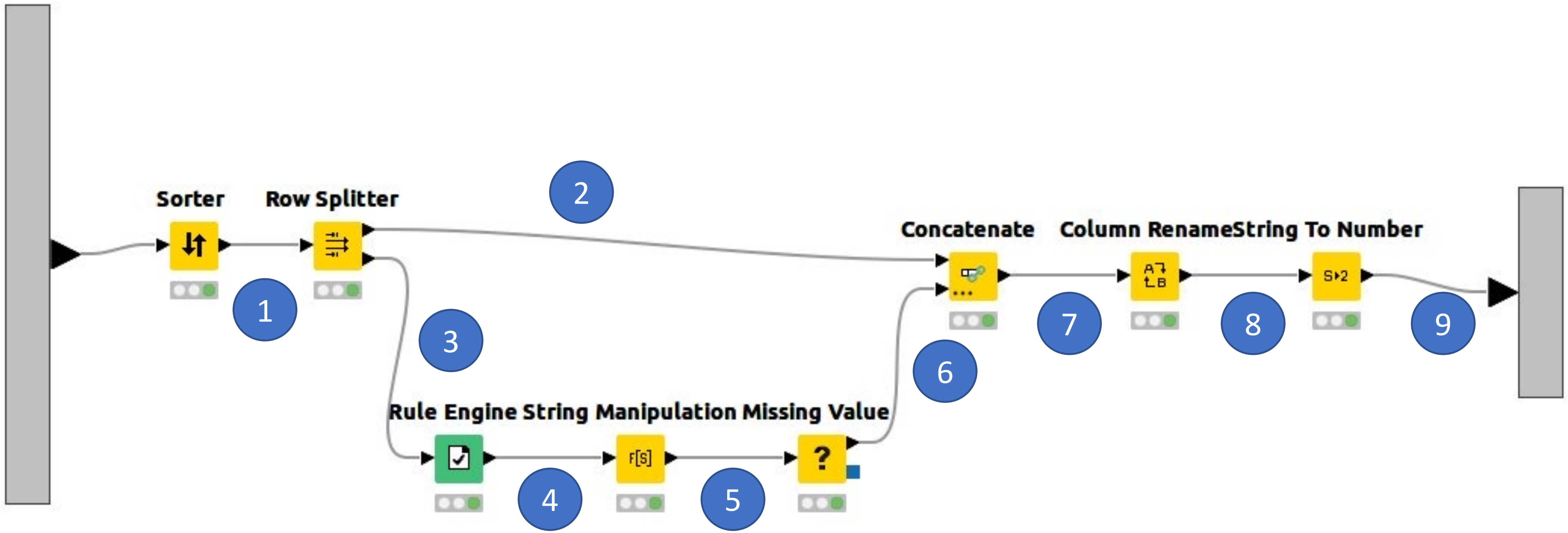
Data consistency



Contracts for data operations

Data Operation	Preconditions	Postconditions	Invariants
RemoveColumns	**dsCol**: The columns specified as parameters must be part of the input dataset.	**noDsCol**: The columns specified as parameters must not be part of the output dataset.	
ImbalancedLearn	**dsCol**: The column specified as parameter must be part of the input dataset. **colCategorical**: The column specified as parameter must be of categorical or boolean type.	**dsCol**: The column specified as parameter must appear in the output dataset. **colCategorical**: The modified column must be of categorical or boolean type.	**dataBalanced**: All values must have the same frequency in the modified column.

Data consistency



KNIME Operation	Frequency	Our Framework	PRE-Conditions Satisfied	POST-Conditions Satisfied	INVariants Satisfied
Column Filter	3	Remove Columns	3/3	3/3	-
String To Number	6	StringToNumber	12/12	12/12	6/6
Missing Value	4	ConditionalFunction	-	-	8/8
Auto-Binner	2	Discretize	4/4	2/2	2/2
Rule-Based Row Filter	2	RuleBasedRowFilter	-	-	2/2
Row Filter	2	RowFilter	2/2	-	2/2
Row Splitter	7	RowSplitter	7/7	-	7/7
Math Formula	4	MathFormula	-	-	4/4
Concatenate	7	Concatenate	-	-	14/14
Constant Value Column	2	ConstantValueColumn	2/2	2/2	2/2
Rule Engine	5	ConditionalFunction	-	-	7/10
Sorter	3	Sorter	3/3	3/3	3/3

Publications

- Rodriguez-Echeverria, R., Conejero, J. M., Melchor González, F. J., Gutiérrez Gallardo, J. D., & Prieto, A. E. (2021). *Towards a Conceptual Framework for the Specification of Reproducible and Replicable Data Analysis Projects.* In E. Insfran, F. González, S. Abrahão, M. Fernández, C. Barry, H. Linger, M. Lang, & C. Schneider (Eds.), *Information Systems Development: Crossing Boundaries between Development and Operations (DevOps) in **Information Systems (ISD2021** Proceedings).* Valencia, Spain: Universitat Politècnica de València.
- Melchor, F., Rodriguez-Echeverria, R., Conejero, J.M., Prieto, Á.E., Gutiérrez, J.D. (2022). *A Model-Driven Approach for Systematic Reproducibility and Replicability of Data Science Projects*. In: Franch, X., Poels, G., Gailly, F., Snoeck, M. (eds) Advanced Information Systems Engineering. **CAiSE 2022**. Lecture Notes in Computer Science, vol 13295. Springer, Cham.