A Soft-Computing Approach to Knowledge Flow Synthesis and Optimization

Tomáš Řehořek Pavel Kordík

Computational Intelligence Group (CIG),
Faculty of Information Technology (FIT),
Czech Technical University (CTU) in Prague

September 5, 2012

Common tasks in Predictive Data Analysis:

1. **Data Preprocessing**
   - Feature Selection,
   - Transformation of Input Space (e.g. PCA)
   - Methods can be put into a Chain
2. **Model Selection**
   - **Selection of Algorithms**
     - k-NN, Naive Bayes, Decision Tree, Neural Networks, . . . ,
     - *Ensemble Techniques* (Majority Vote, Bagging, Stacking. . .)
   - **Parameter Tuning**
     - Distance Measure and k in k-NN,
     - Minimal Gain for Split in Decision Tree

Introduction: Knowledge Flows in Predictive Analysis

Dependence on Data

The optimal choice of Preprocessing and Modeling methods is **data-dependent**:

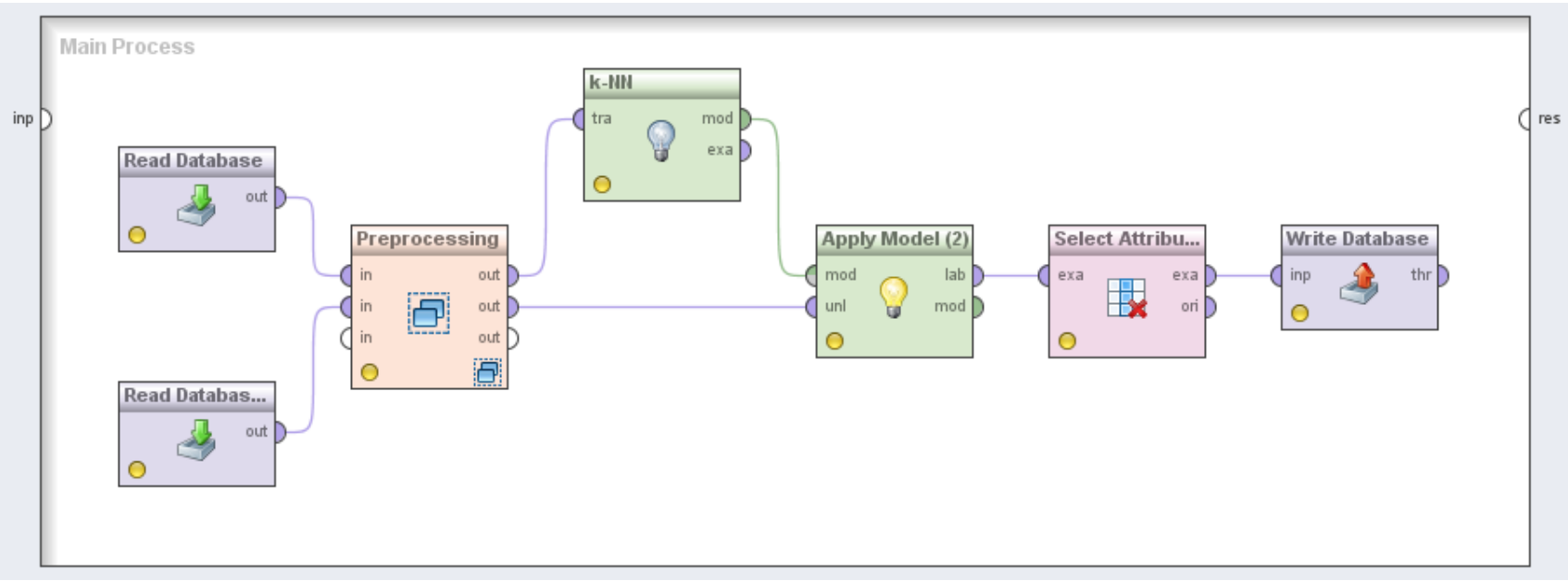
- Different Datasets require different preprocessing and are suitable for different modeling algorithms

Traditional approach: Explore the data in **trial-and-error** manner

- Even data mining experts follow this scenario

Modern approach to express the whole process: **Knowledge Flows (KFs)**

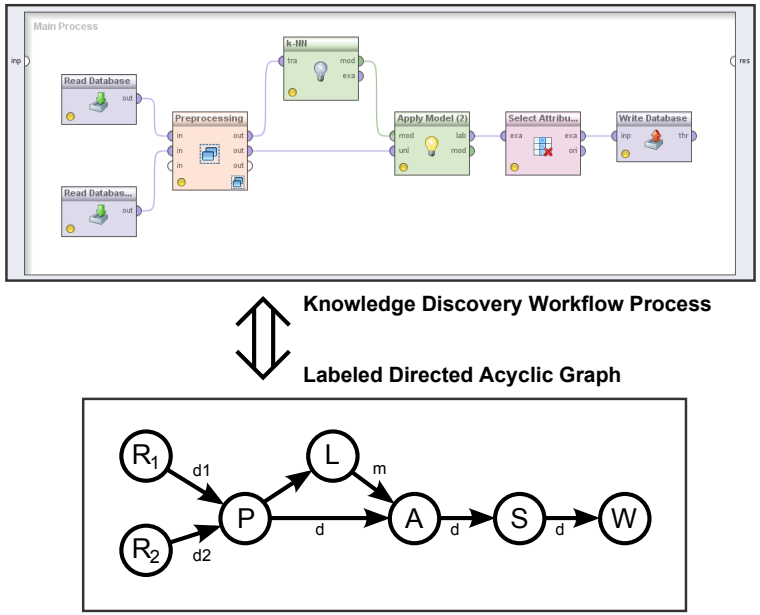
- *Directed Graphs* of interconnected, properly configured **Actions**
- Example: KFs in *RapidMiner 5* learning environment:



Optimization of Knowledge Flows

Evolving Good Graphs

Our Approach: KFs are viewed as **Directed Acyclic Graphs (DAGs) with labeled nodes**.



These graphs are **subject to optimization** by means of **evolutionary computation**.

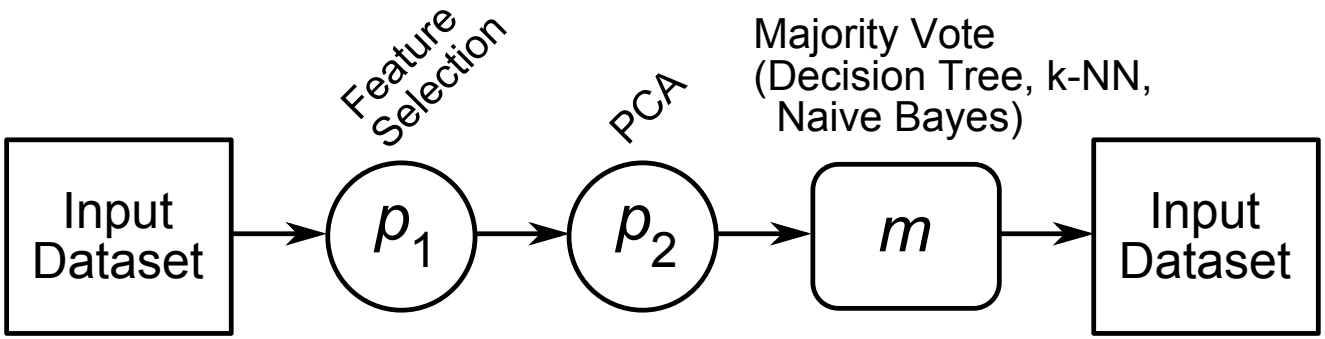
Problem Statement: Finding Optimal Predictive Analysis KF for a given Dataset

Given a Dataset **D** of examples from $\mathbb{R}^n \times \mathcal{L}$, find:

1. Sequence $p_1, \ldots, p_k$ of properly configured **preprocessing actions**,
2. Properly configured **learning algorithm** (or possible hierarchical ensemble of such algorithms) m

such that model obtained as $m\left(p_k\left(\ldots p_1\left(\mathbf{D}\right)\ldots\right)\right)$ has **minimal generalization error**.

Example:



Methodology

Embryonic STGP [Koza1997]

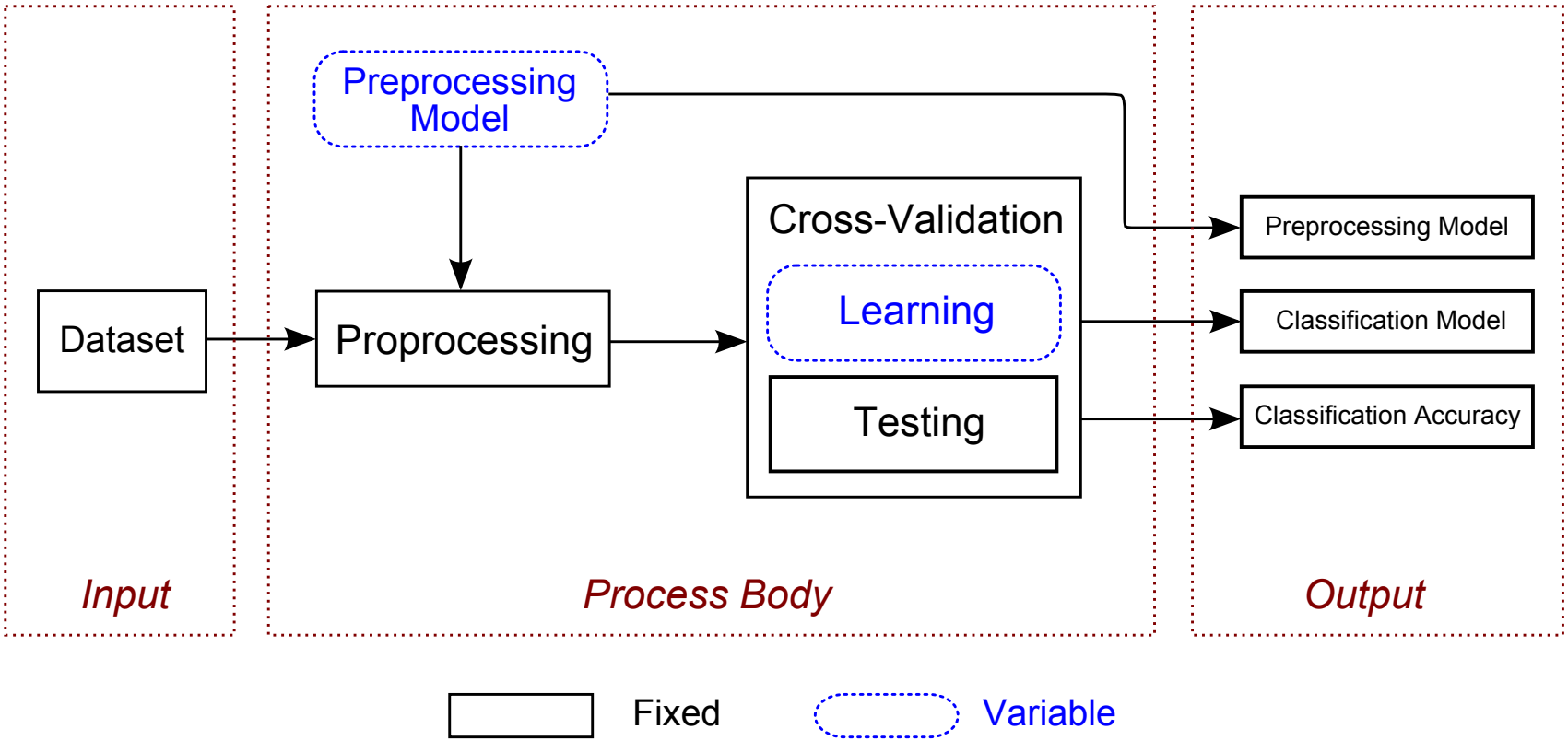
We use **Embryonic Strongly Typed Genetic Programming**

- Proposed by J.R.Koza for evolving Analog Electrical Circuits [Koza1997]
- Encodes the graph in form of **rooted tree**
  - The tree codes a plan for developing a complete graph from a simple graph referred to as the **embryo**
- Fitness function: **classification accuracy** $\frac{1}{|T|} \sum_{(\mathbf{x},\ell) \in T} \begin{cases} 0, & m(\mathbf{x}) \neq \ell \\ 1, & m(\mathbf{x}) = \ell \end{cases}$

Methodology

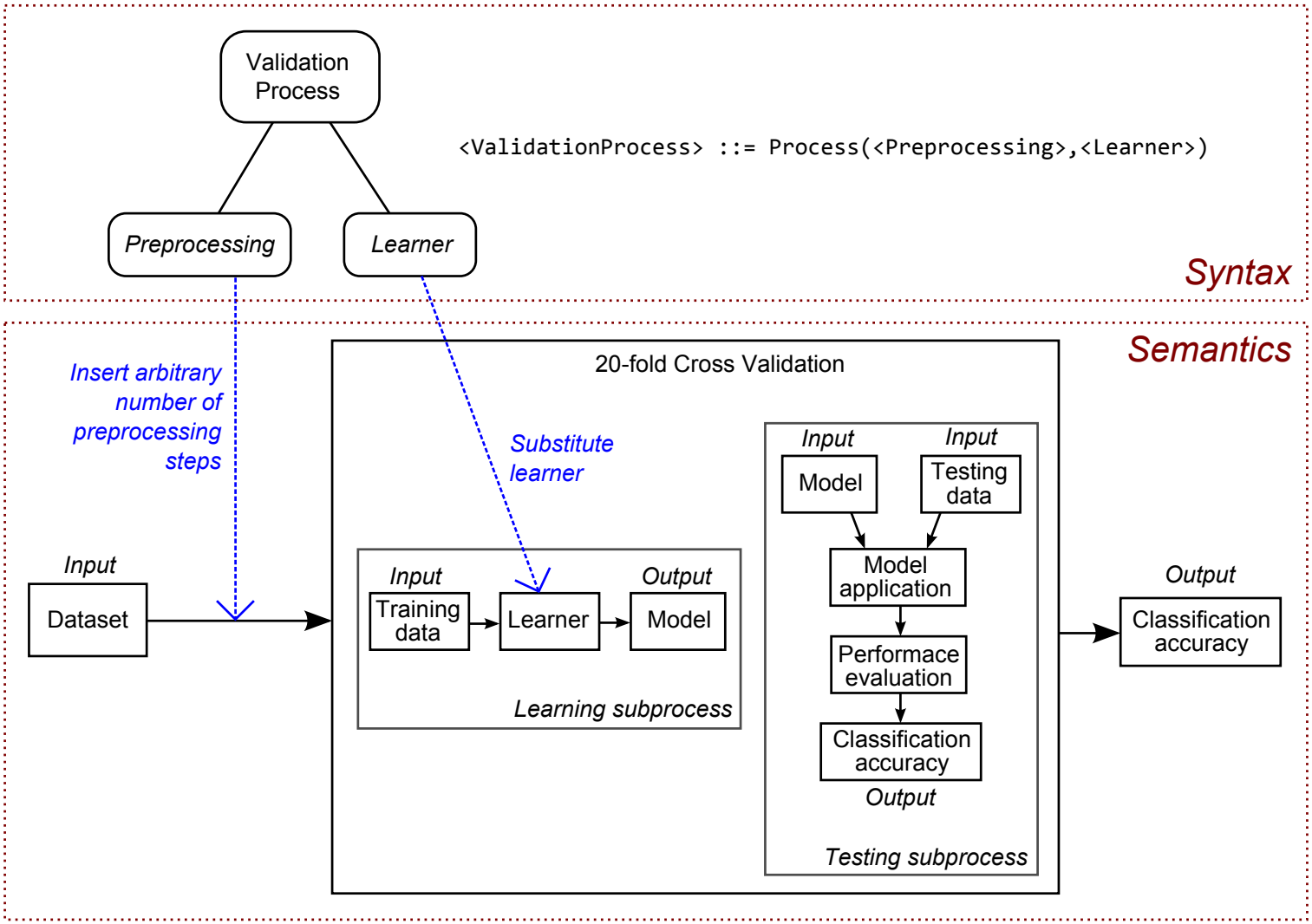
Embryonic KF in RapidMiner 5

- We use the RapidMiner 5 software to measure fitness
- Embryonic Knowledge Flow for our experiment:



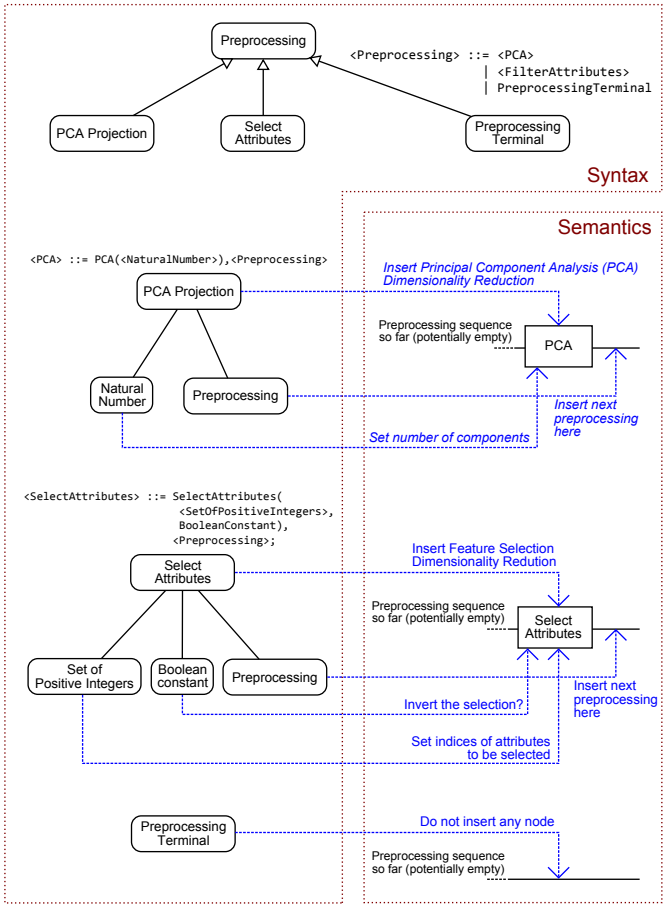
Methodology

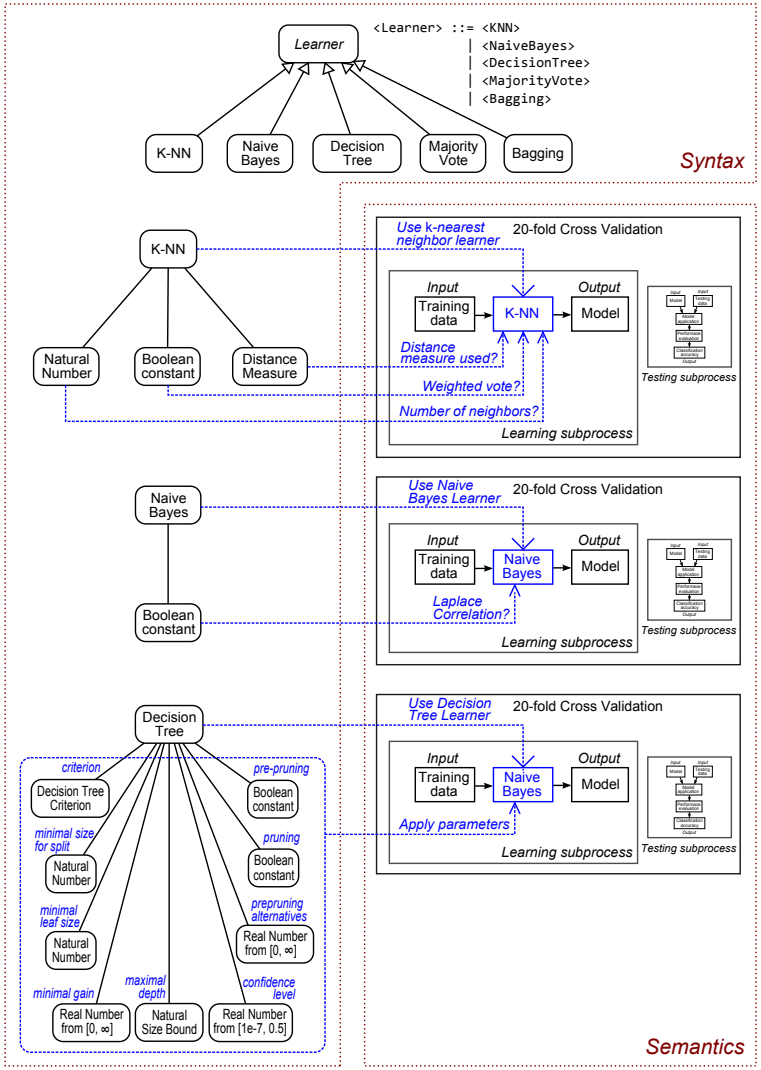
STGP Grammar: Validaton Process



Methodology

STGP Grammar: Preprocessing, Modeling

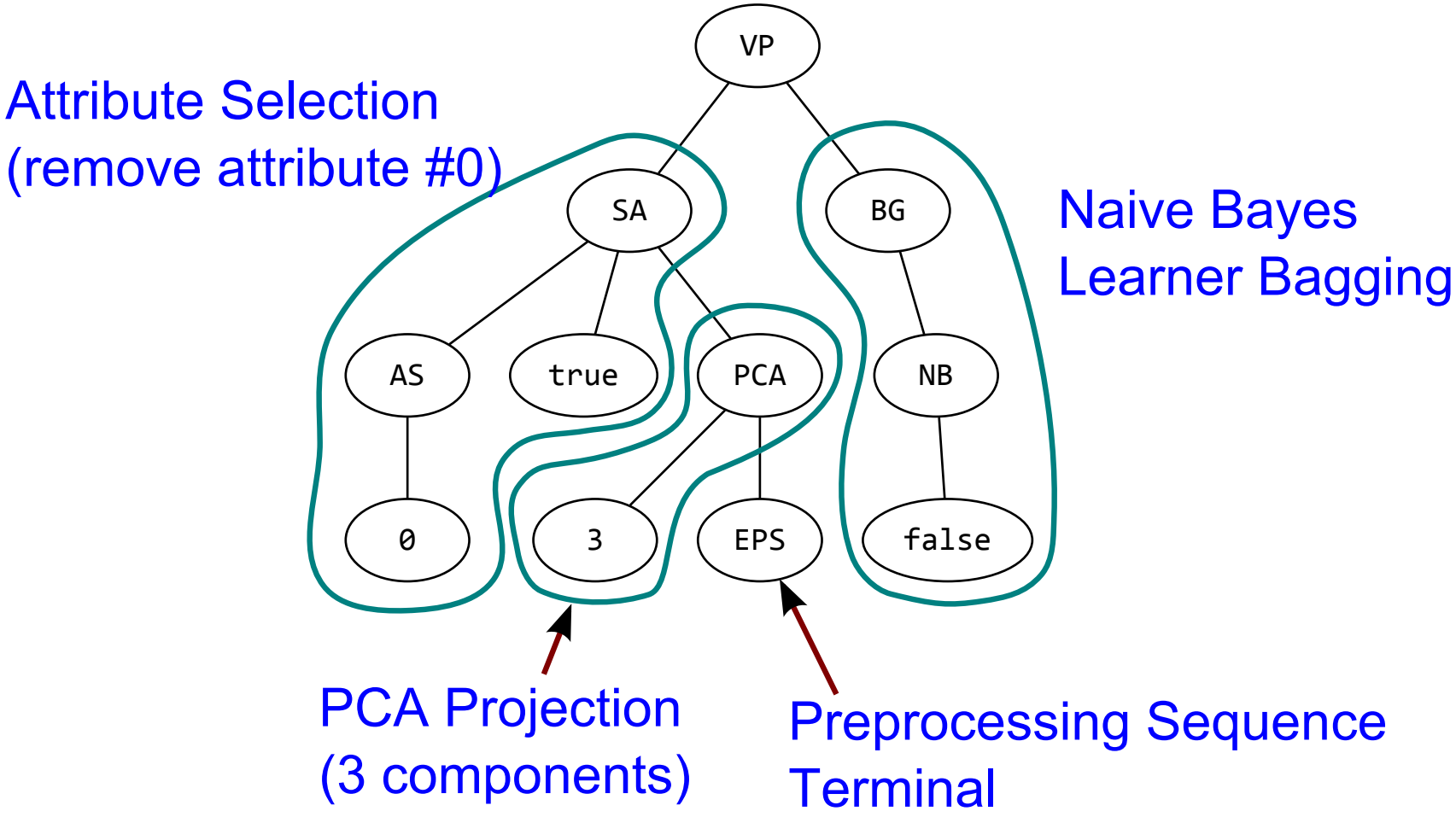




Preliminary Results

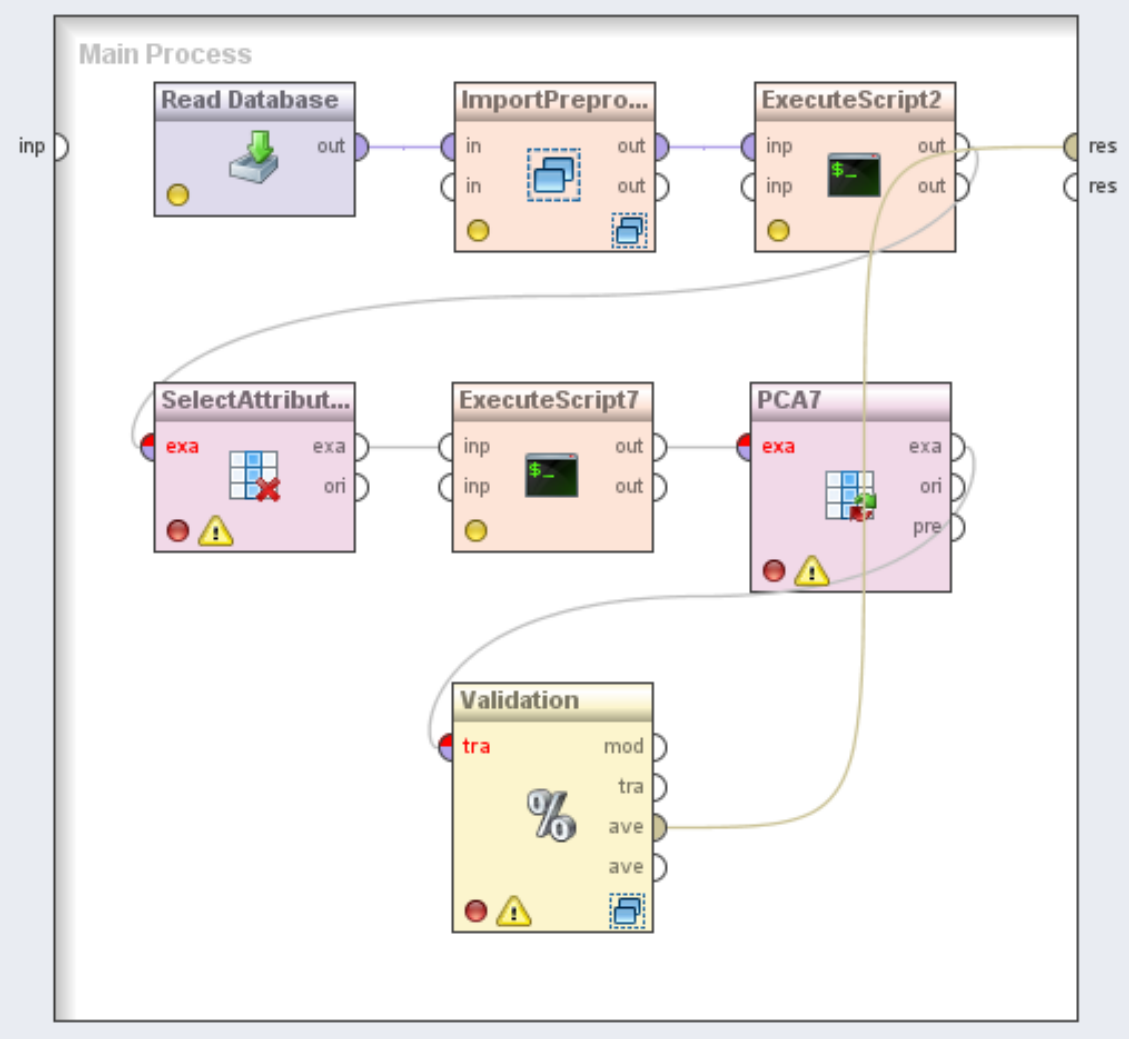
Ecoli Dataset: Sample Tree Evolved

Sample tree evolved on the Ecoli datatset



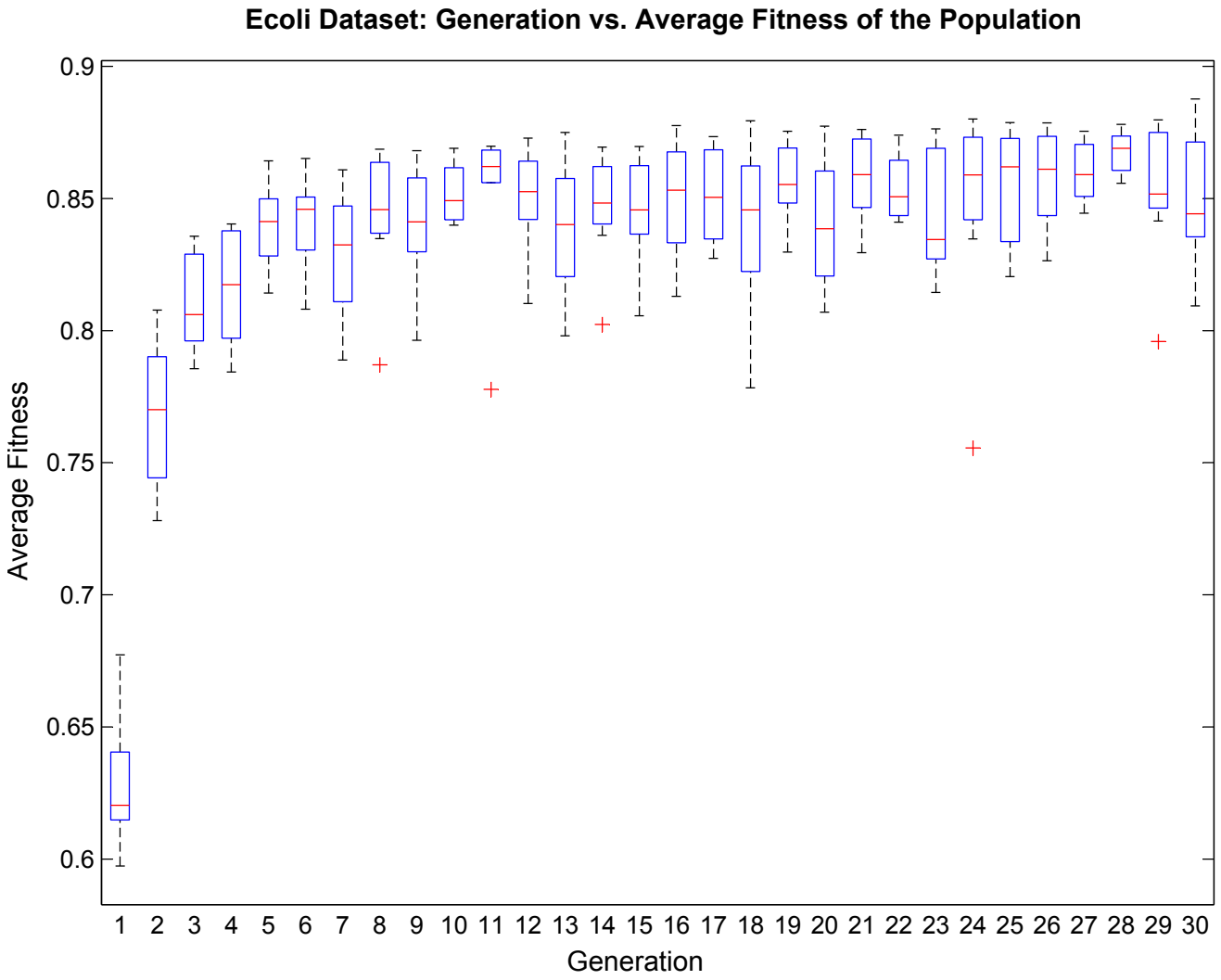
Preliminary Results

Ecoli Dataset: Evolved KF in RapidMiner 5



Preliminary Results

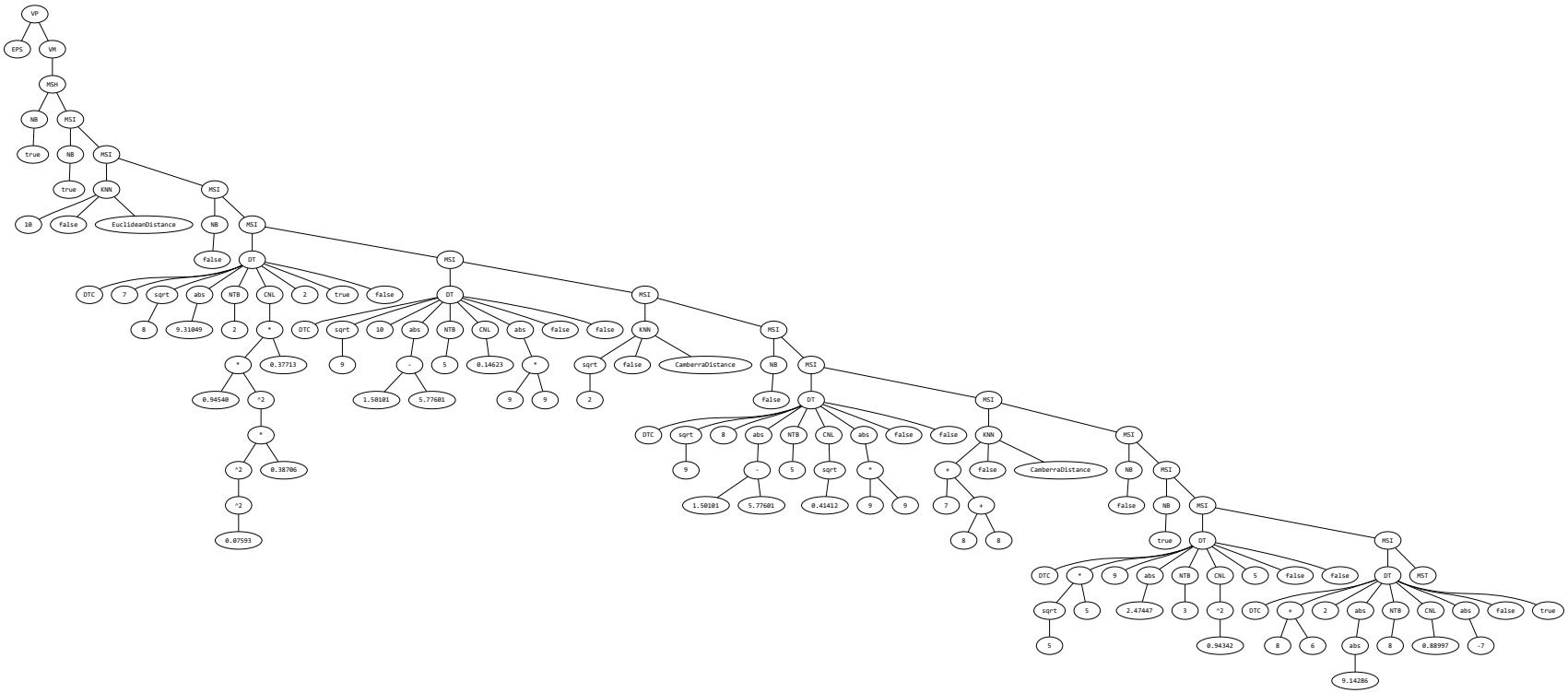
Ecoli Dataset: Generation vs. Average Fitness of the Population



Preliminary Results

Very Complex Tree Evolved of the Vehicle Dataset

A very complex tree evolved on the Vehicle dataset



Questions & Discussion

Thank you for you attention!

Tomáš Řehořek
tomas.rehorek@fit.cvut.cz