Biostat 202: Opportunities and Challenges of "Big Data".

Machine Learning 3: Supervised learning, contd. [classification]

Aaron Wolfe Scheffler

Division of Biostatistics

Department of Epidemiology and Biostatistics

Announcements

Assignment 3 due Thursday

Project Component 2 due 8/15 @ midnight

Overview of this lecture - more classification!

- K-nearest neighbors
  - Parameter tuning
  - Training-validation-test
- Classification trees (recursive partitioning)
- Support vector machines
- Artificial neural networks
- Comparison of classifiers

Ongoing case study – Heart data

- Data:
  - n = 303
  - 14 attributes
- **Target:**
  - **AHD: diagnosis of heart disease (1 = yes, 0 = no)**

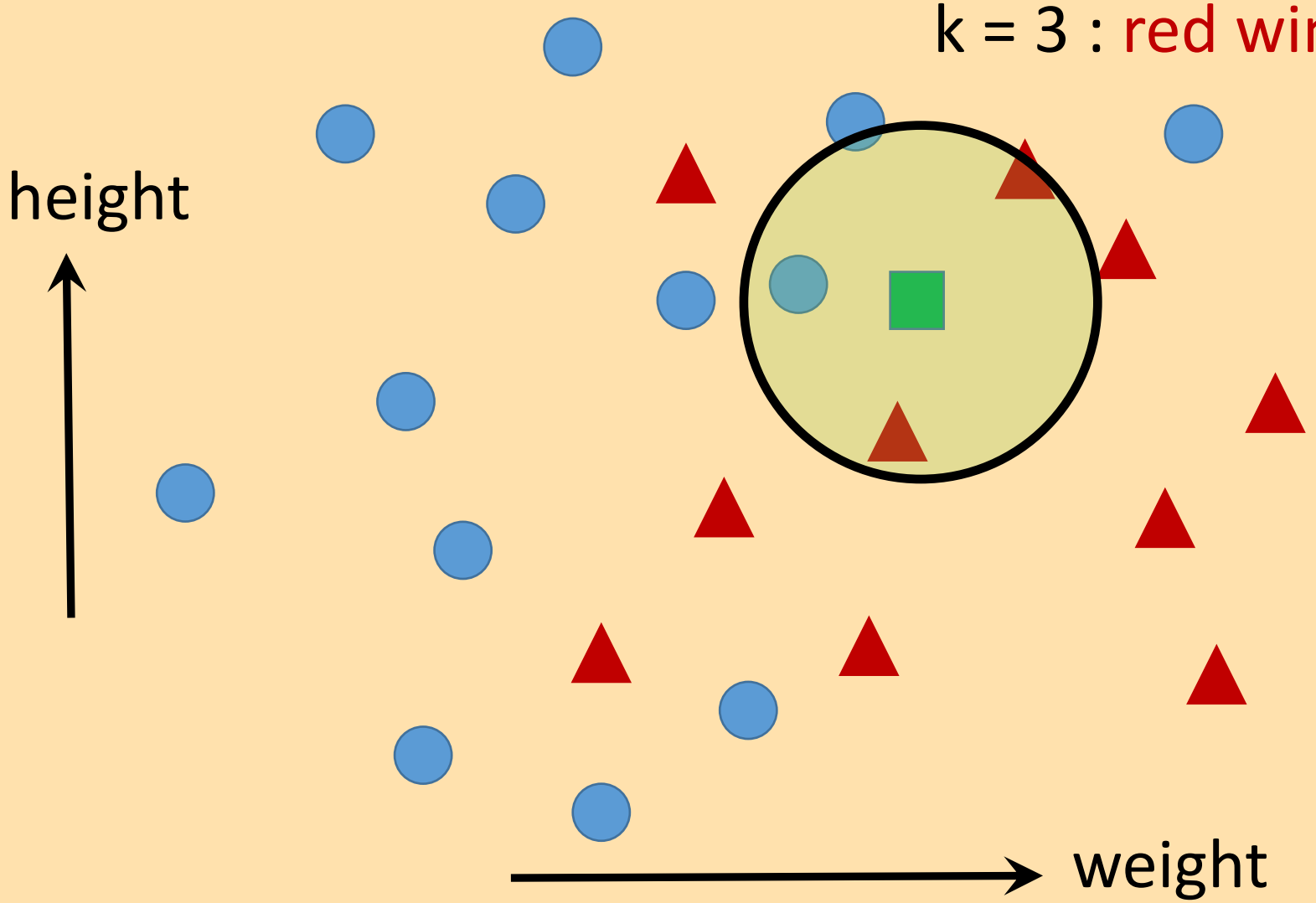
Predictors:

- Age: age (years)
- Sex: gender
- ChestPain: chest pain type
- Chol: cholesterol
- Fbs: fasting blood sugar >120 mg/dL
- restecg: resting electrocardiographic results
- MaxHR: maximum heart rate achieved
- ExAng: exercise induced angina
- Oldpeak: ST depression induced by exercise
- Slope: slope of peak exercise ST segment
- Ca: # of major vessels colored by flouroscopy
- Thal: Thallium stress test
- SysBp: systolic blood pressure

Goal: Predict (probability) of heart disease using different classifiers.

K-nearest neighbors classification

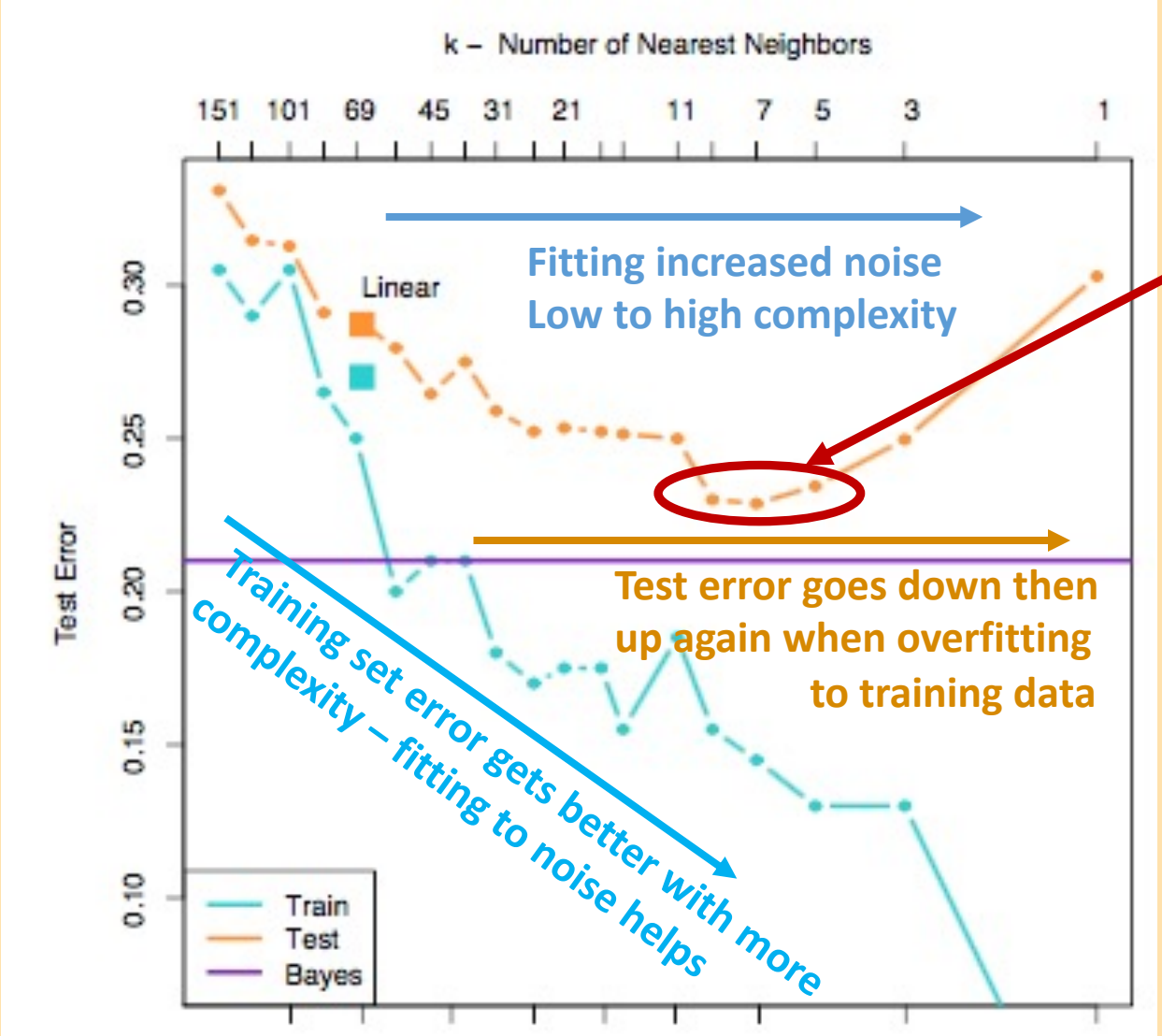
K-nearest neighbors classification



Parameter tuning

- Parameter tuning is the process of adjusting model parameters in order to find the “best” model
- For example, we tune the K-nearest neighbors by adjusting the parameter k, the number of neighbors
- Analogy: tuning the strings of a guitar to play the proper note is like parameter tuning to get the best prediction

Choice of k


k – Number of Nearest Neighbors
151 101 69 45 31 21 11 7 5 3 1
Test Error
0.10 0.15 0.20 0.25 0.30
Linear
Fitting increased noise
Low to high complexity
Best predictive models (k = 7)
Test error goes down then up again when overfitting to training data
Training set error gets better with more complexity – fitting to noise helps
Train
Test
Bayes

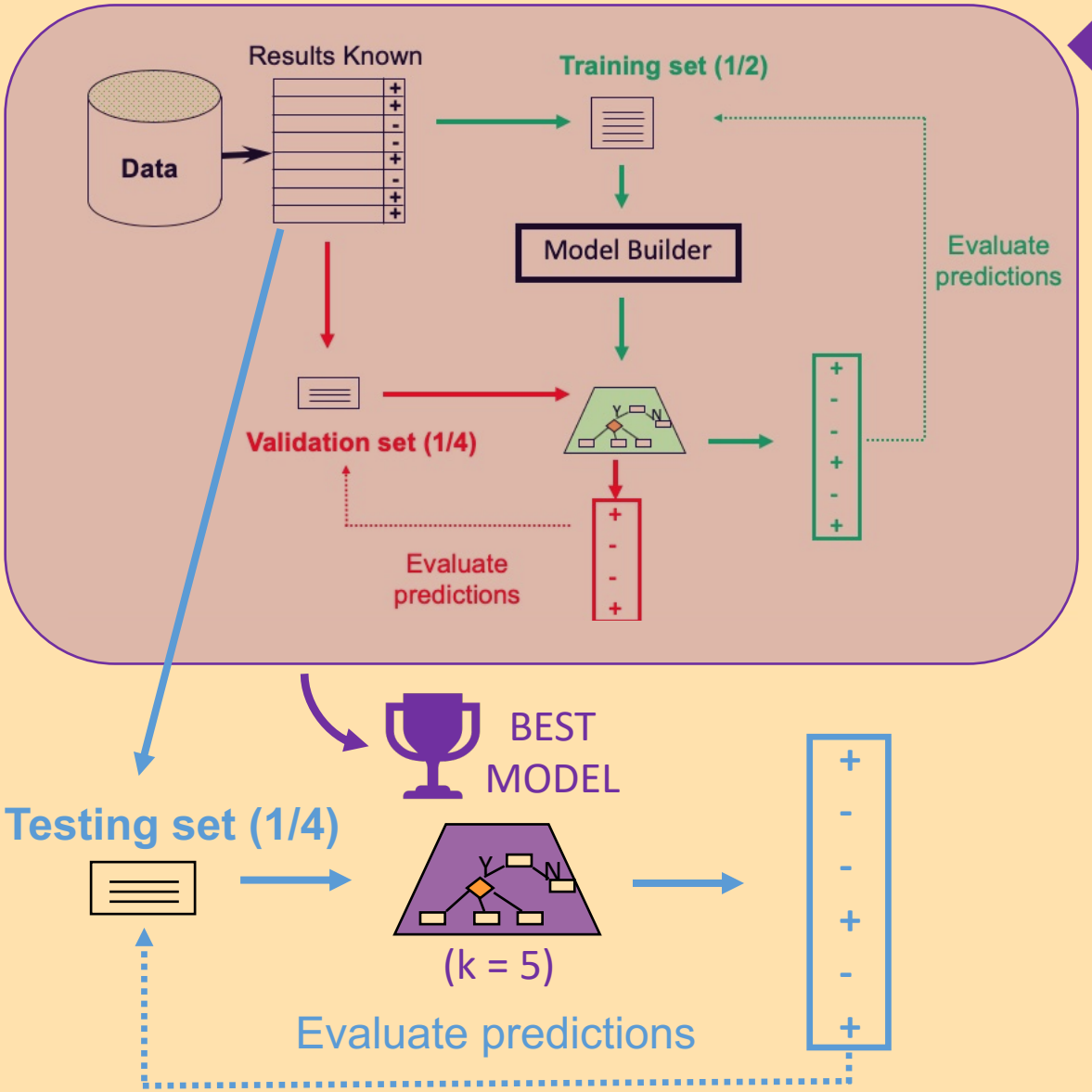
Train/Validate/Test

k = 3, 4, 5, 6

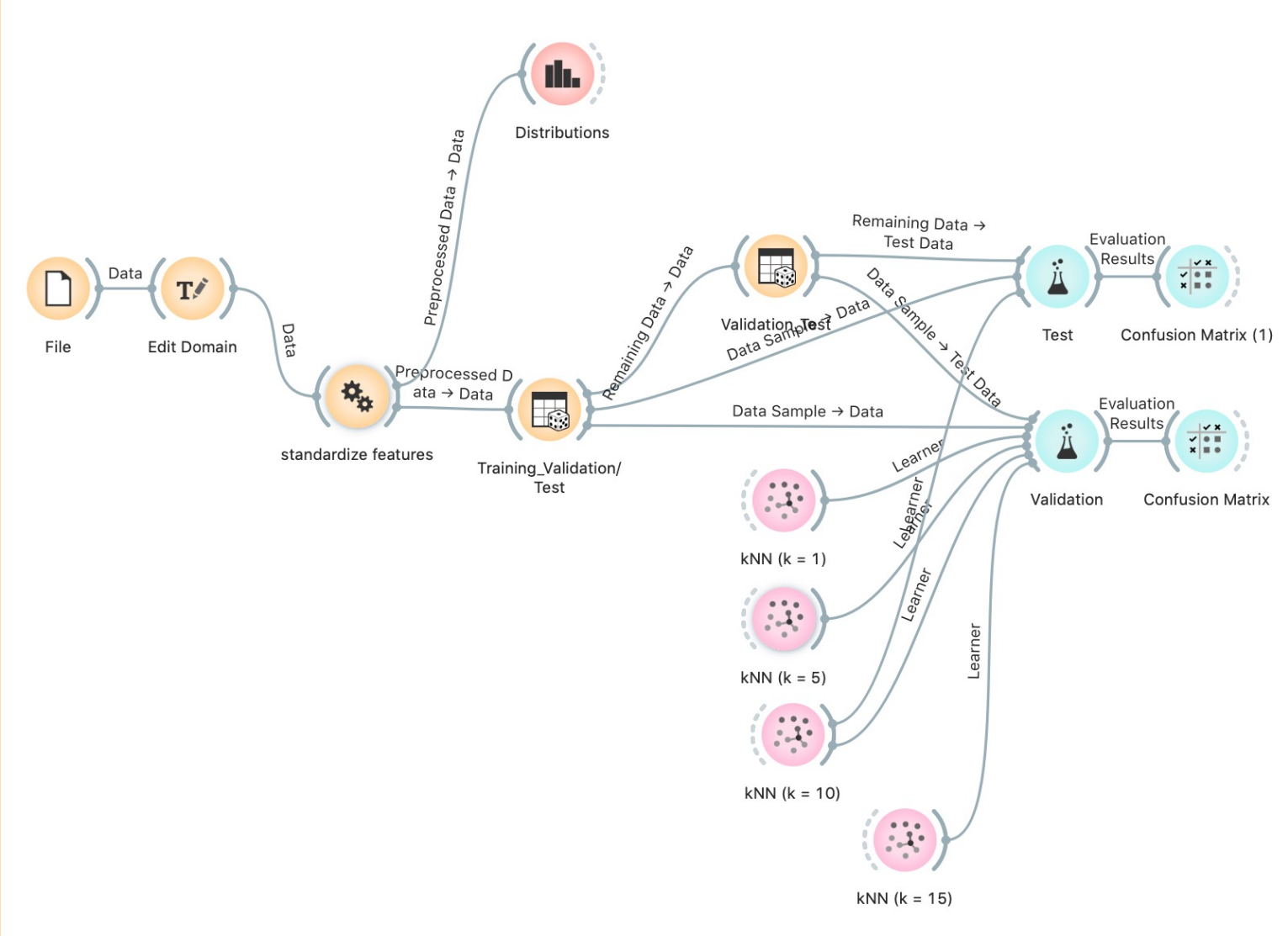
1. Train: repeatedly fit model to training data for multiple values of the procedures parameter(s), e.g. k = 3, 4, 5, 6 for K-nearest neighbors

2. Validate: each model based on training set has potentially been overfit… so utilize a validation set to determine best fitting model

3. Test: we are choosing from many models so we may have over-fit…utilize a test set to determine predictive performance of the best fitting model from the purple box

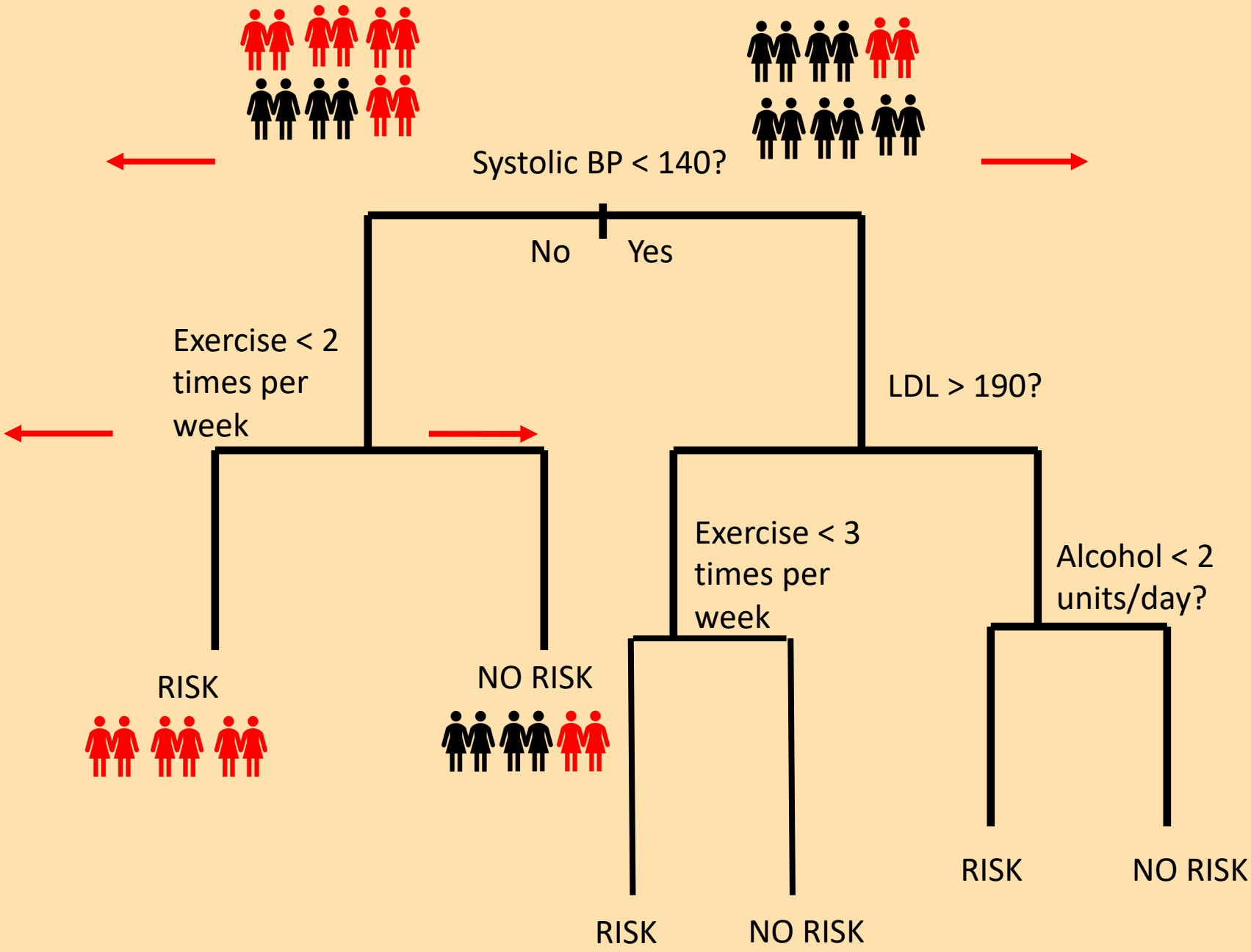


K-Nearest Neighbors workflow – available on CLE

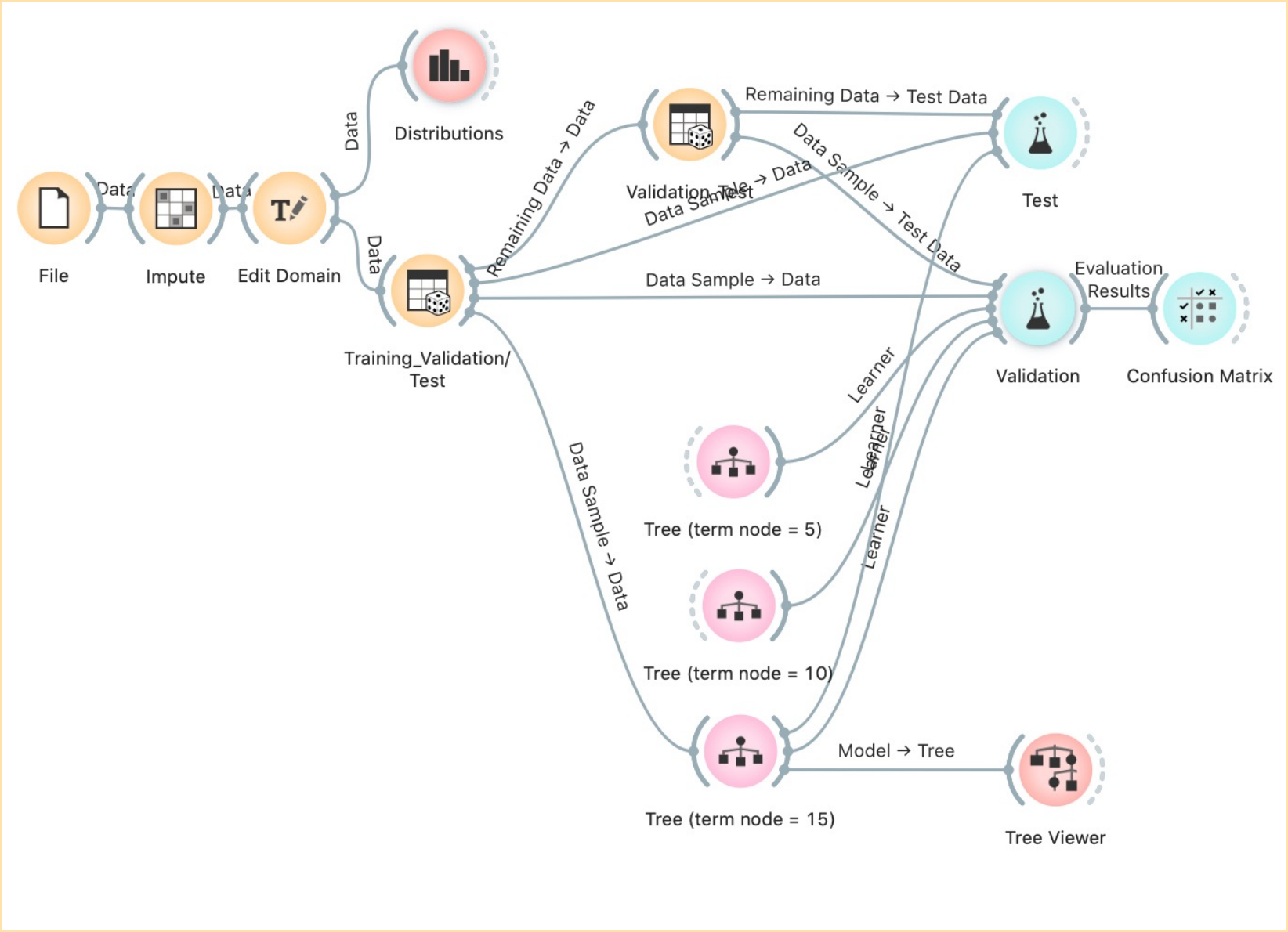


Decision Trees
(Recursive Partitioning)

At risk for heart disease? RISK vs NO RISK

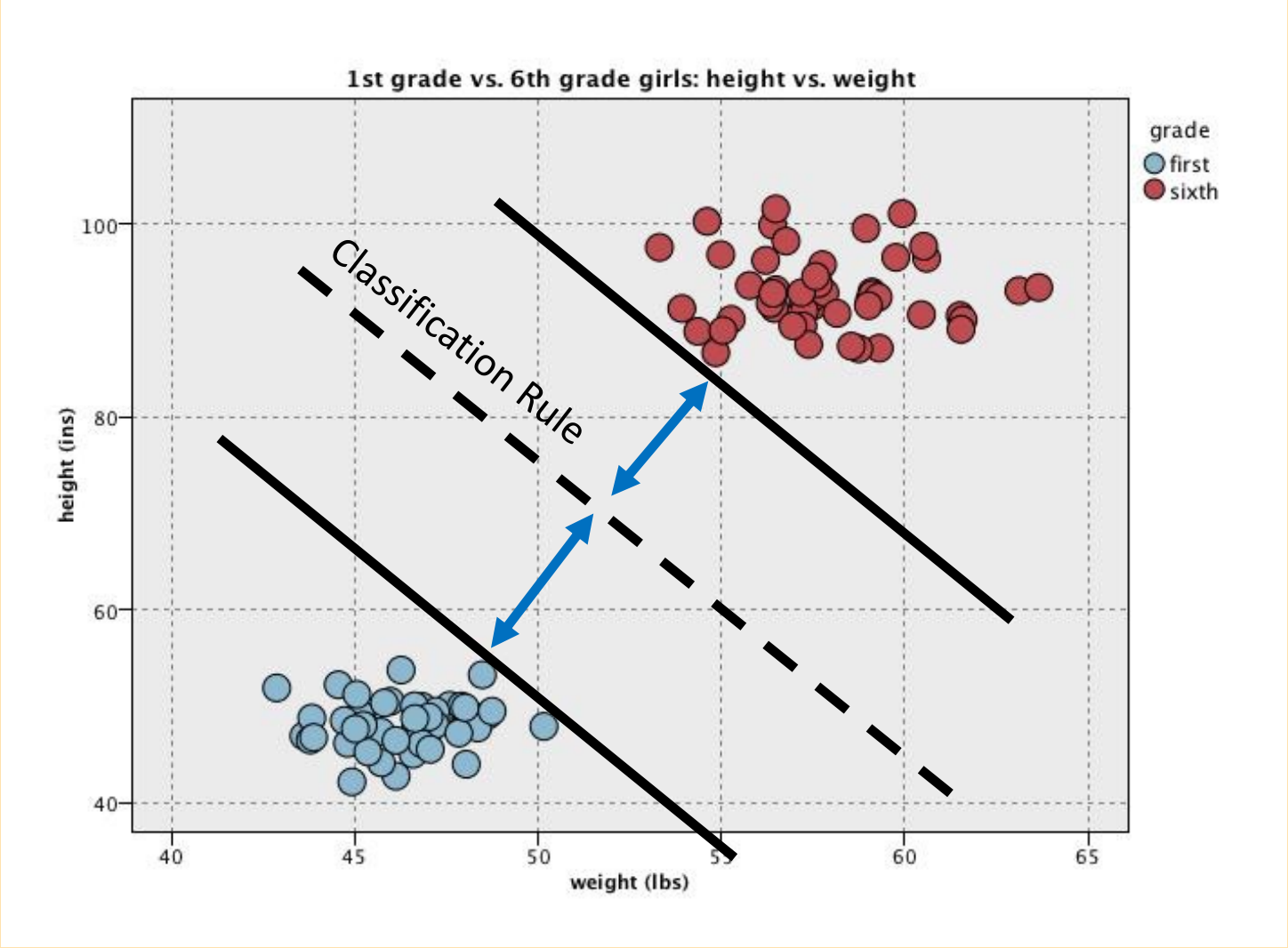


Tree workflow – available on CLE

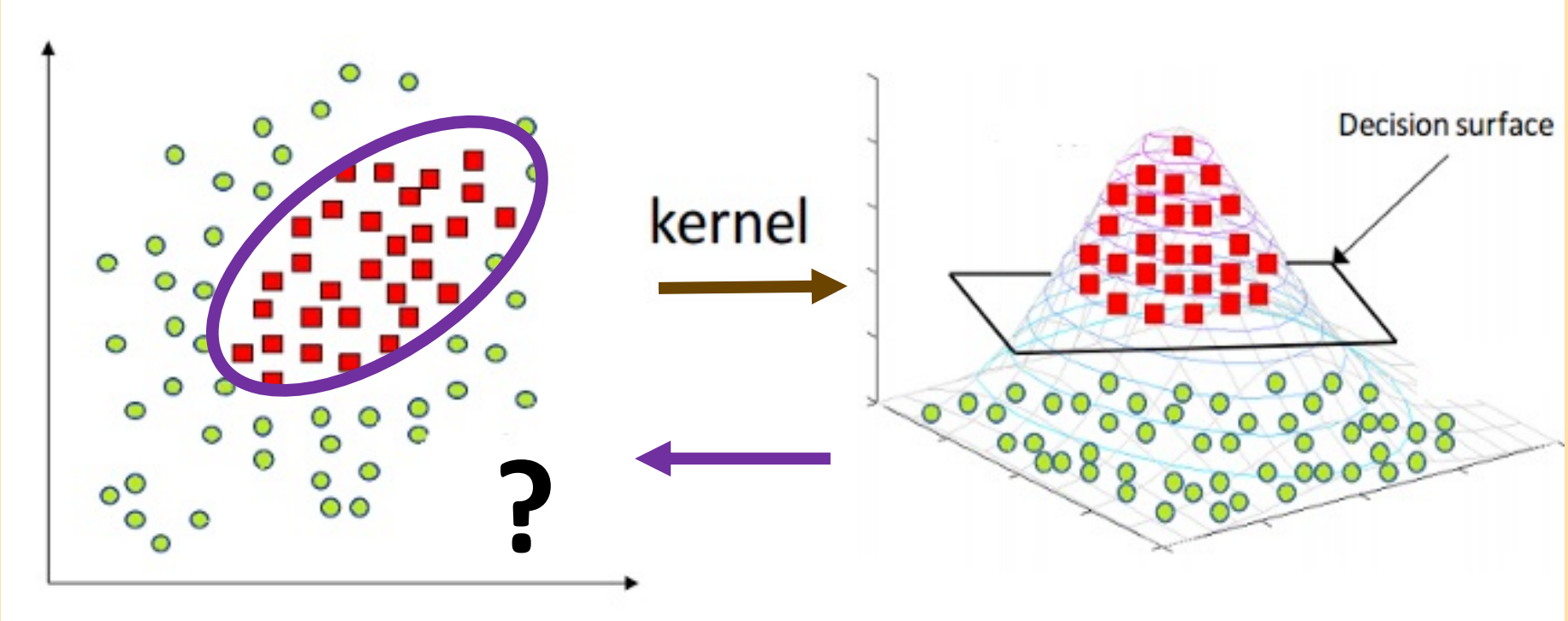


Support Vector Machines (SVM)

Maximizing margins

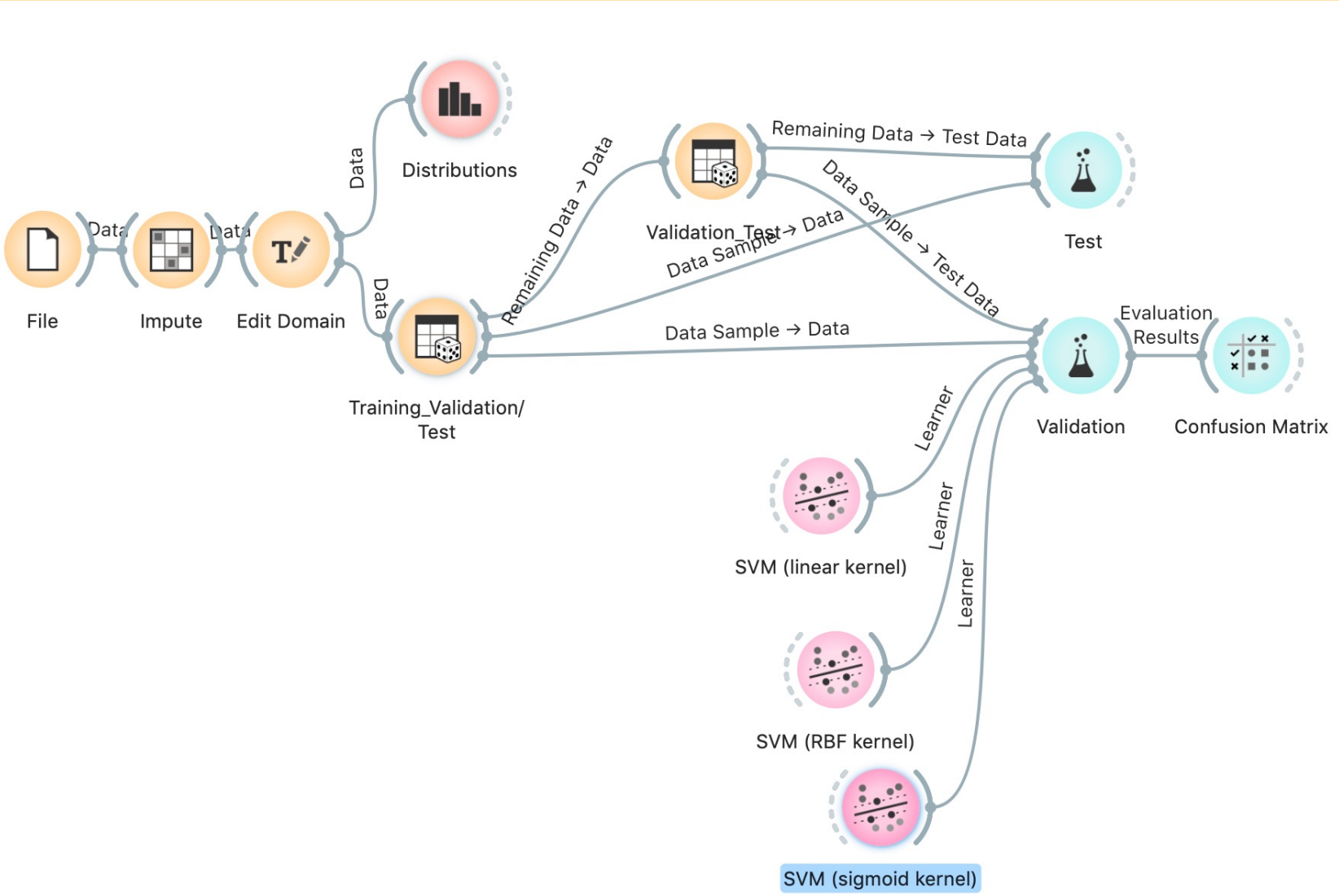


Linearity can be limited, may need "non-linear" decision boundary

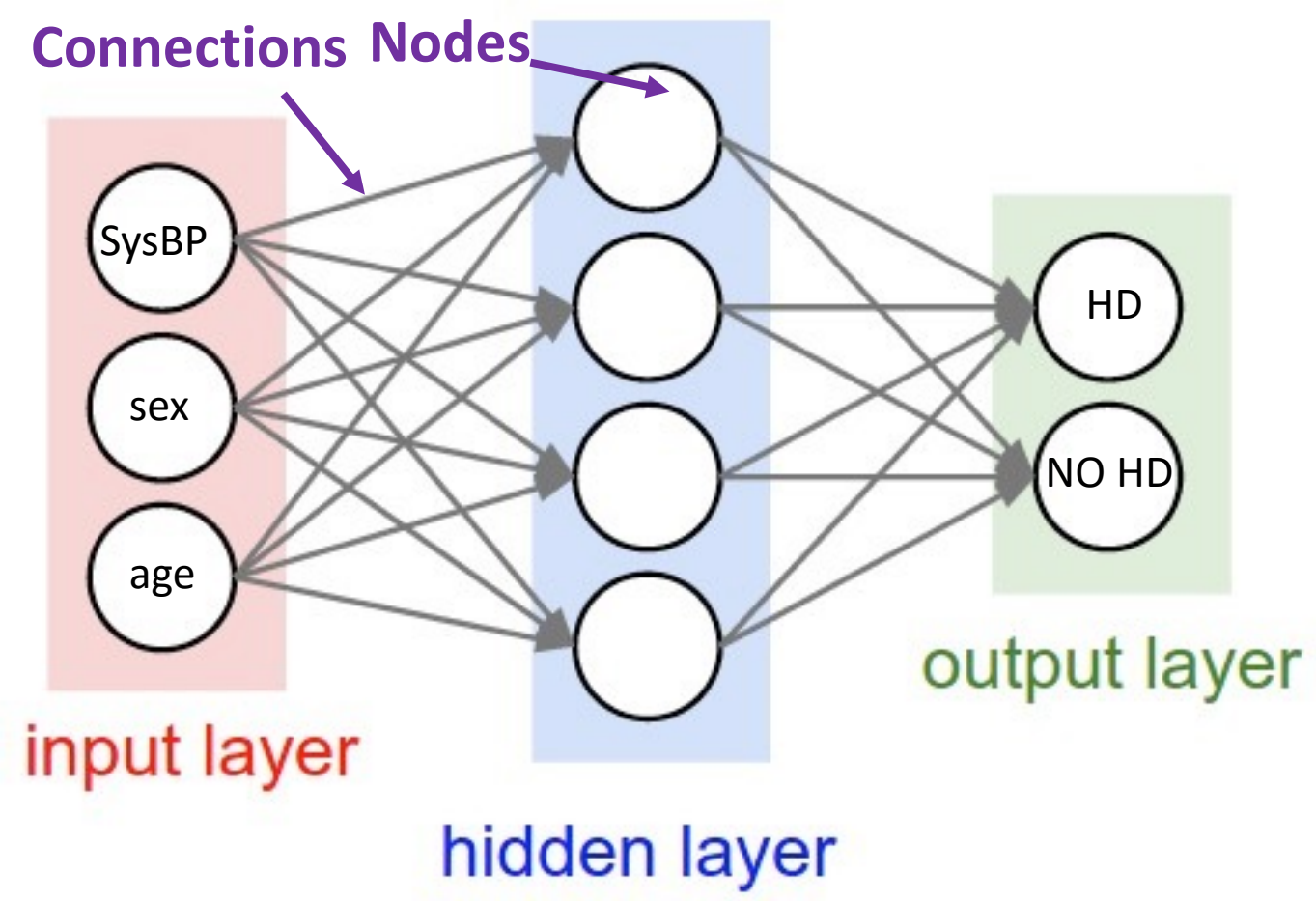


The trick of SVM is not to have non-linear models for the separation boundaries directly, but rather to **warp the space** that the data is in. That way, linear boundaries in the warped space become **non-linear** back in the original space.

SVM stream – available on CLE



Artificial neural networks

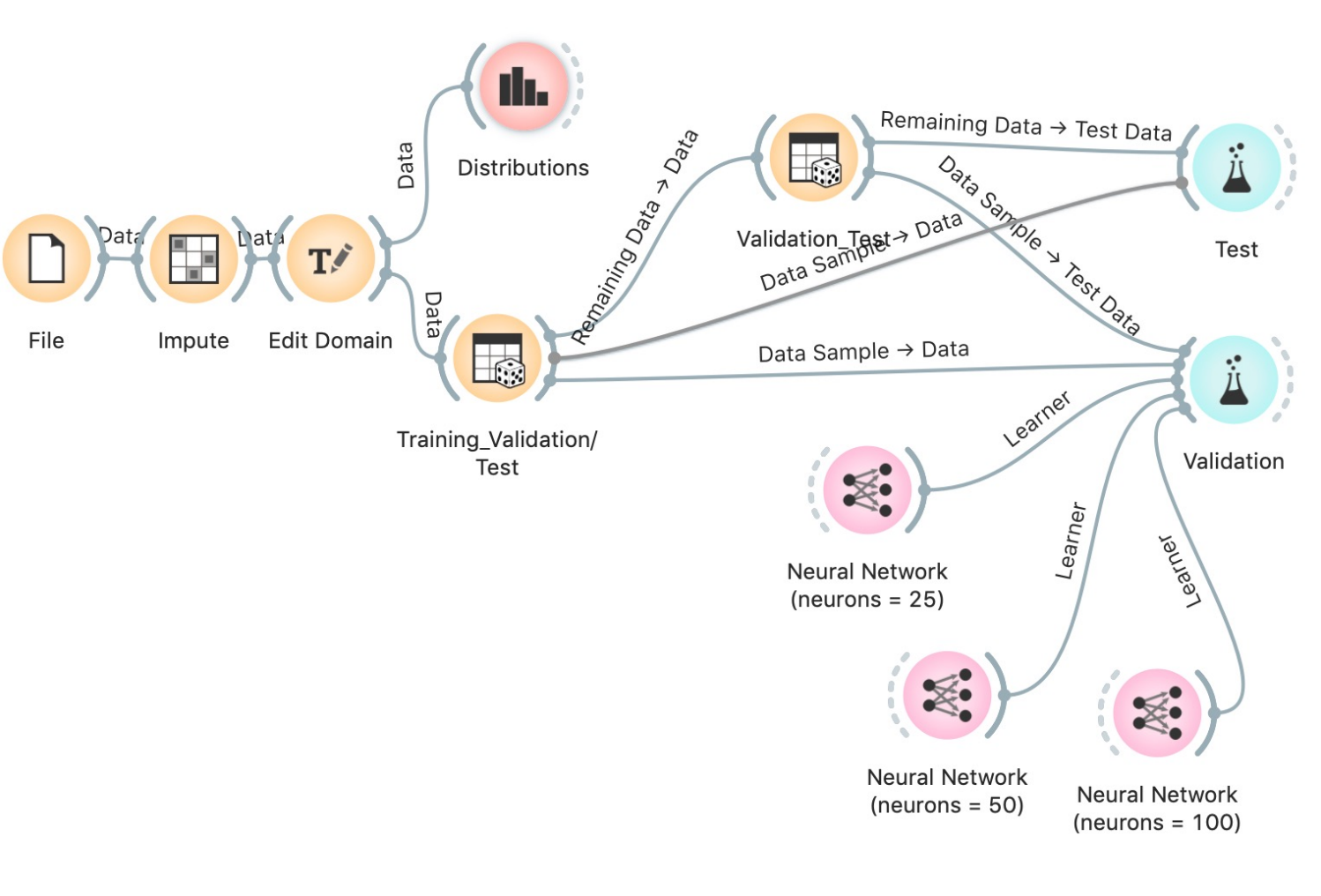


Underlying math is complex, but every node is a complex function of it's inputs, and the parameters/weights (connections) of the function are learned from the training data. To expand model complexity, add more hidden layers.

Artificial neural network - visualization

https://www.youtube.com/watch?v=3JQ3hYko51Y

Artificial Neural Network stream – available on CLE



Many of the models discussed today can be used for regression!

- With some adaptation, the techniques we have discussed today can be applied to target continuous outcomes, i.e. regression problems
- We will not discuss but Orange offers regression versions of the following algorithms:
  - K-nearest neighbors
  - Regression trees (recursive partitioning)
  - Support vector machines
  - Artificial neural networks

Next Thursday—supervised learning fin

- Cross validation
- Ensemble methods
  - Bagging
  - Boosting
  - Stacking
- Parameter tuning
- Feature importance