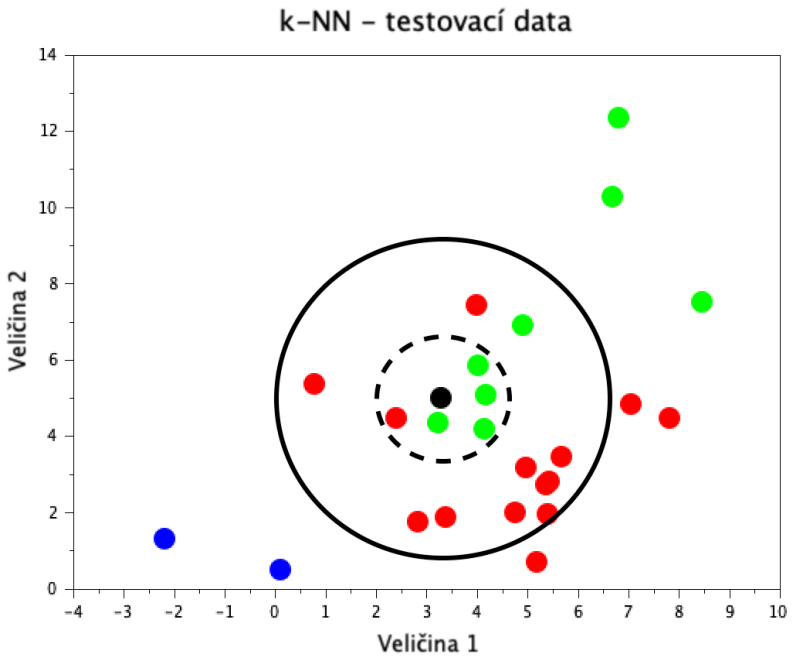
Algoritmus k-nejbližších sousedů (k-nearest neighbors, k-NN)

- máme data $\{y_i\}_{i=1}^{n_d}$ se známými shluky
- naměříme nový datový bod y_{new}

Princip:

1. Spočteme vzdálenost bodu y_{new} ode všech bodů
2. Určíme k nejbližších bodů
3. Přiřadíme y_{new} do shluku, kam patří většina nejbližších bodů



Příklad: věk dětí ve třídách

$k = 5$ • $\in \{\bullet, \ldots, \bullet\}$

$k = 15$ • $\in \{\bullet, \ldots, \bullet\}$

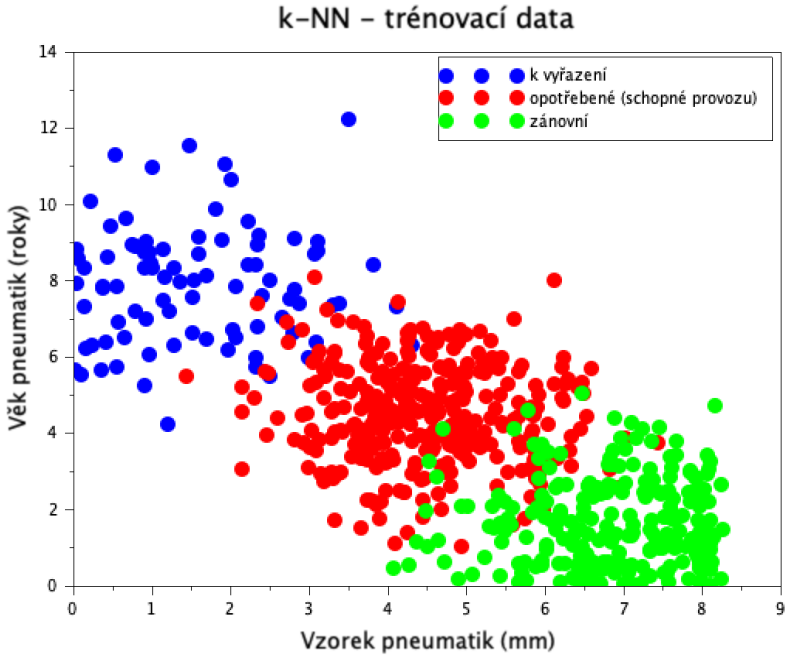
Volba k – empiricky

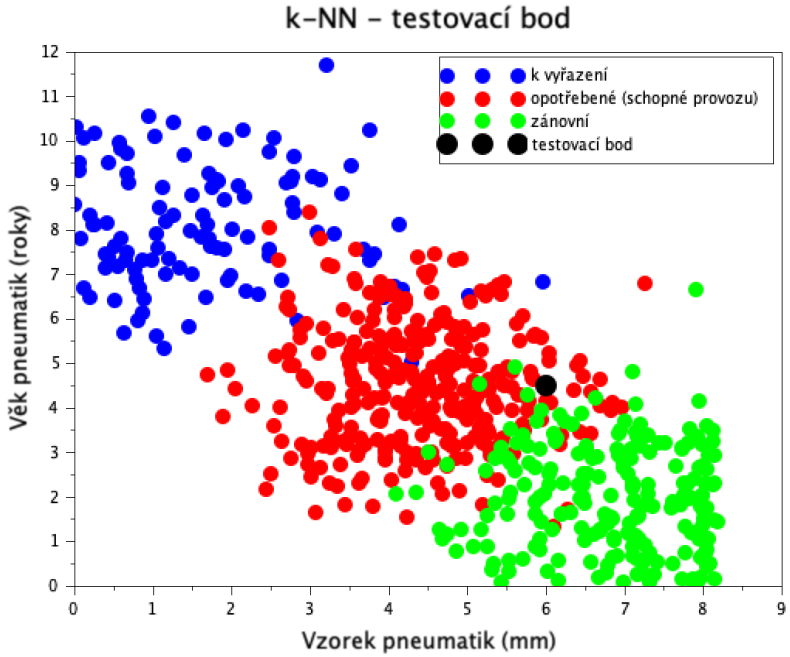
- Křížová validace – přesnost pro různé k
- liché k
- $k = \sqrt{n_d}$, $k = \ln(n_d)$
- expertně

Příklad: Klasifikace stavu pneumatik v autoservisu

Postup:

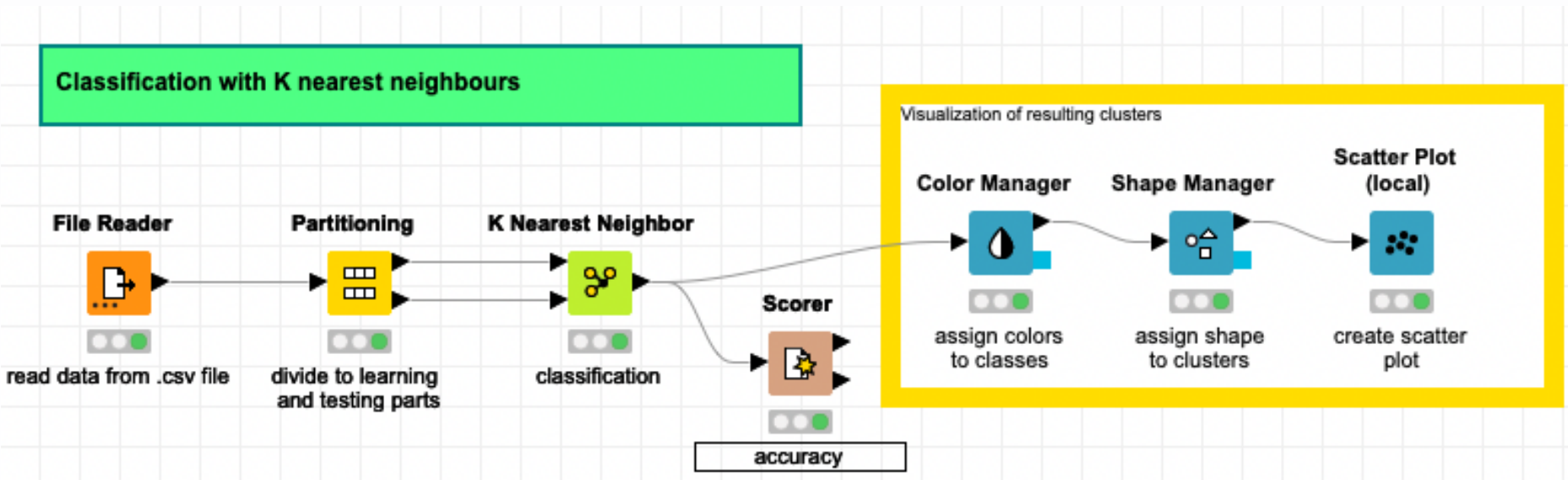
- Trénovací data – databáze servisu (vzorek, věk, stav pneumatik)





- Testovací data
  1. Seřadíme vzdálenosti (každý bod – trénovací data)
  2. Vezmeme indexy k nejkratších vzdáleností
  3. Které shluky to byly?
  4. Kterých bylo nejvíce?
- Chybová matice, přesnost

Program v KNIME



Correct classified: 146	Wrong classified: 17
Accuracy: 89,571%	Error: 10,429%
Cohen's kappa (κ): 0,838%	

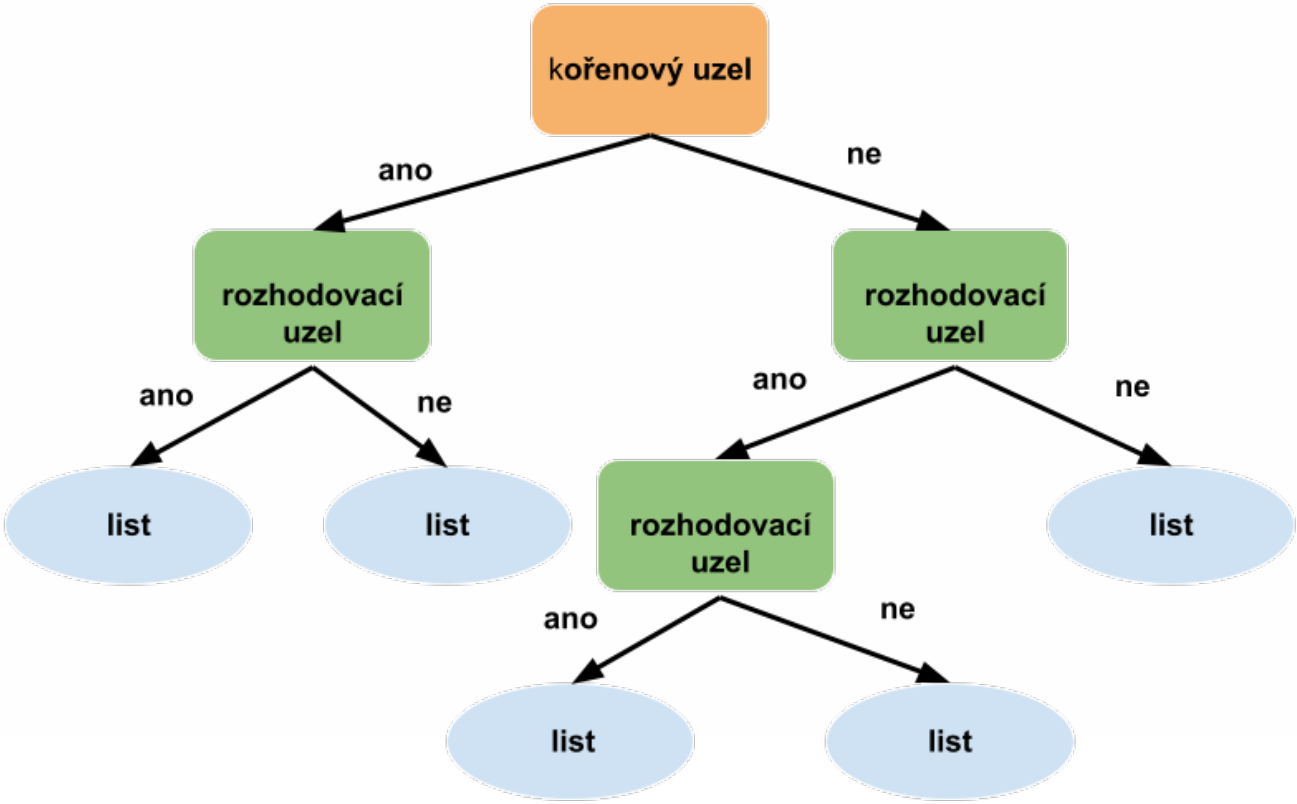
Poznámky – zvýšení přesnosti

- vážení dat (podle vzdálenosti)
- normování dat
- volba k

Rozhodovací stromy

Princip, definice:

- kořenový uzel, větve



- rozhodovací uzly, listy (finální klasifikace)

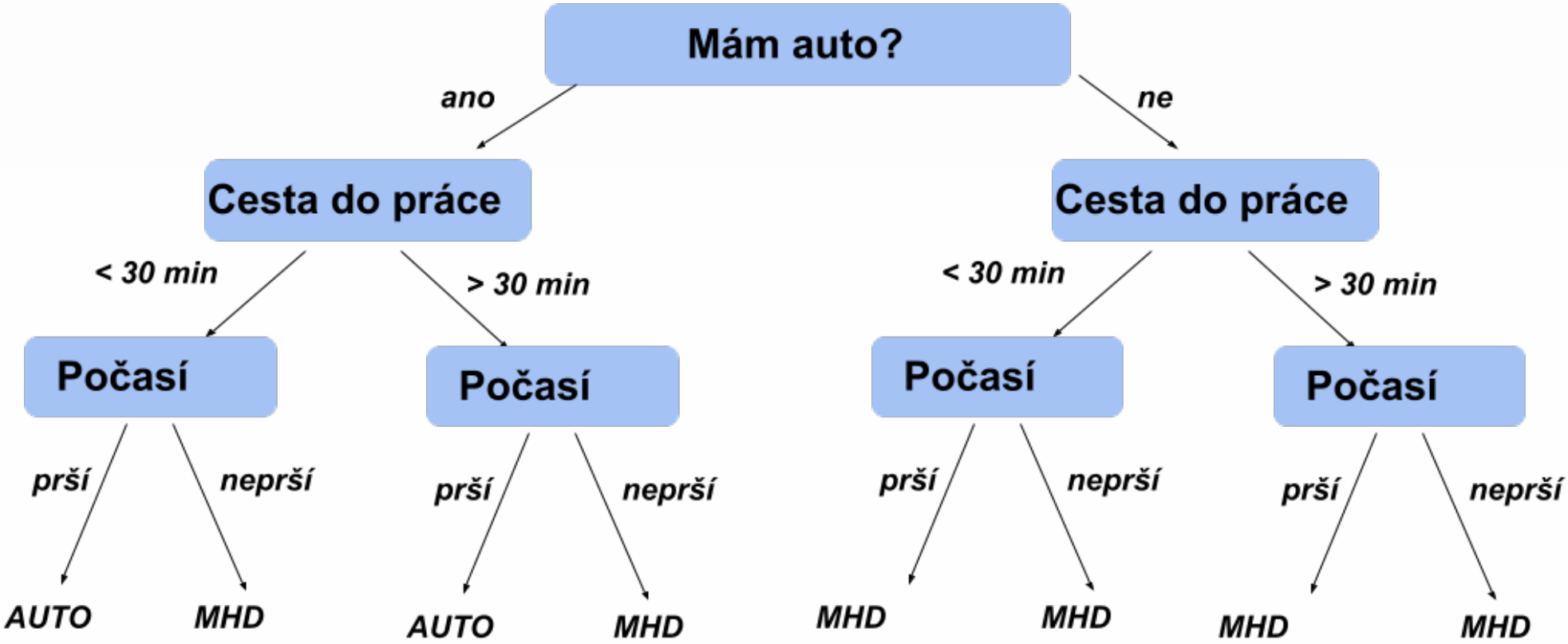
Příklad: Klasifikace volby dopravního prostředku

Data

- mám auto $\in \{0 = \text{ne}, 1 = \text{ano}\}$
- cesta do práce $\in \{0 = \text{kratší něž 30 min}, 1 = \text{delší něž 30min}\}$
- počasí $\in \{0 = \text{neprší}, 1 = \text{prší}\}$
- cíl: volba dopravního prostředku $\in \{1 = \text{auto}, 2 = \text{MHD}\}$

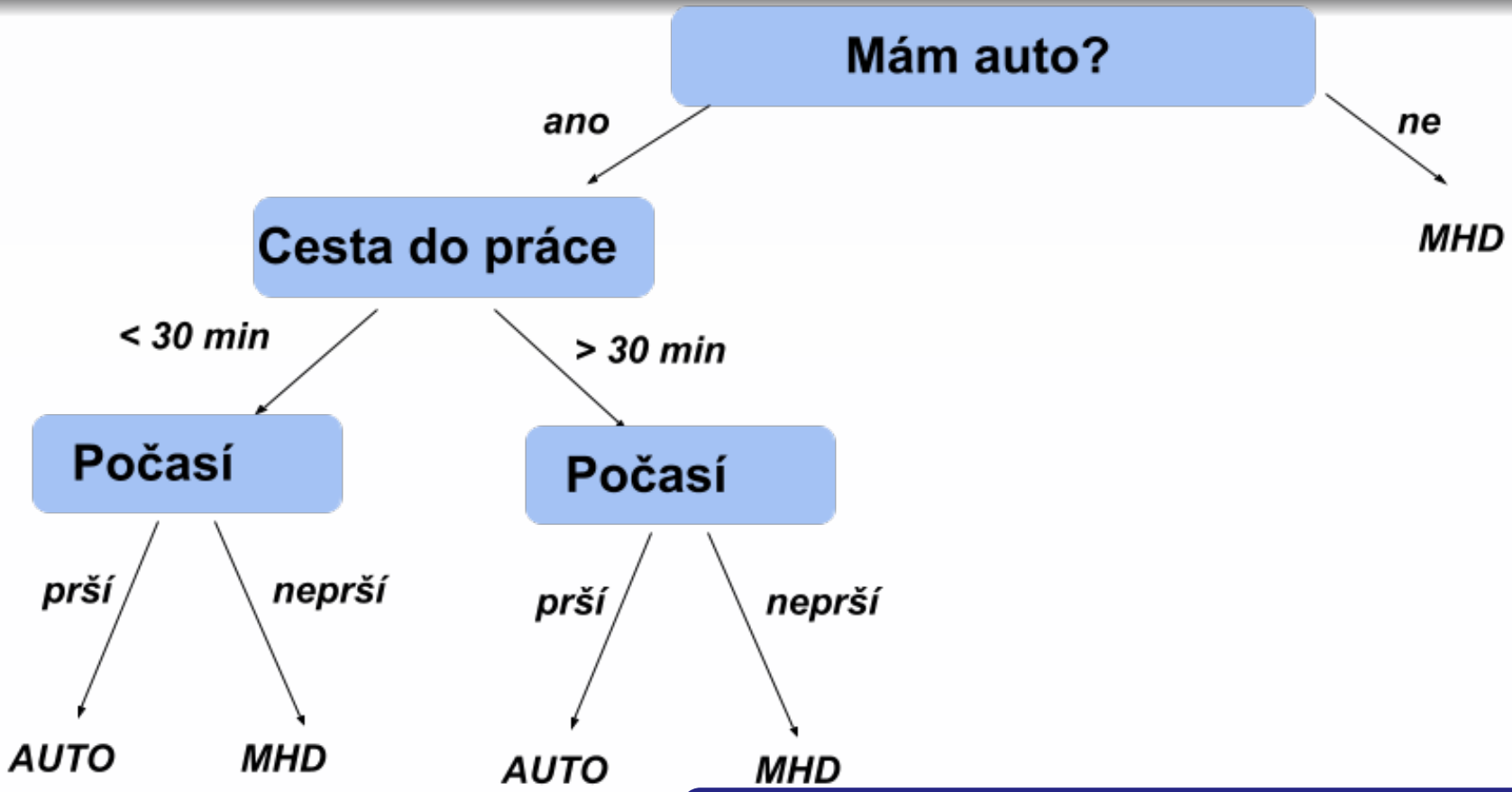
mám auto	cesta do práce	počasí	dopravní prostředek
ano	$<$ 30 min	prší	auto
ano	$<$ 30 min	neprší	MHD
ano	$>$ 30 min	prší	auto
ano	$>$ 30 min	neprší	MHD
ne	$<$ 30 min	prší	MHD
ne	$<$ 30 min	neprší	MHD
ne	$>$ 30 min	prší	MHD
ne	$>$ 30 min	neprší	MHD

Rozhodovací strom ručně – 1



- zbytečně hluboký strom

Rozhodovací strom ručně – 2



- jiný kořenový uzel – počasí, cesta

Nejlepší strom – dělí data jednoznačně

- 100% – 0% vs 50% – 50%
- informační zisk IG $\in (0, 1)$ (ideální IG $\to 1$)
- čistota (purity) každého rozkladu
- čistý rozklad – všechny datové body patří do stejné třídy (všichni jedou MHD)

Jak najít nejlepší rozhodovací strom?

Giniho index (Gini impurity) $\in (0, 0.5)$

- 0 – zcela čistý (100% – 0%)
- 0.5 – zcela nečistý (50% – 50%)

$$\text{Gini} = 1 - \sum_{i=1}^{K} p_i^2$$

- p_i – pravděpodobnosti tříd
- menší Gini = lepší rozklad

Entropie $\mathcal{E} \in (0, 1)$

- 0 – zcela čistý (100% – 0%)
- 1 – zcela nečistý (50% – 50%)

$$\mathcal{E} = -\sum_{i=1}^{K} p_i \log_2 p_i$$

- menší $\mathcal{E}$ = lepší rozklad

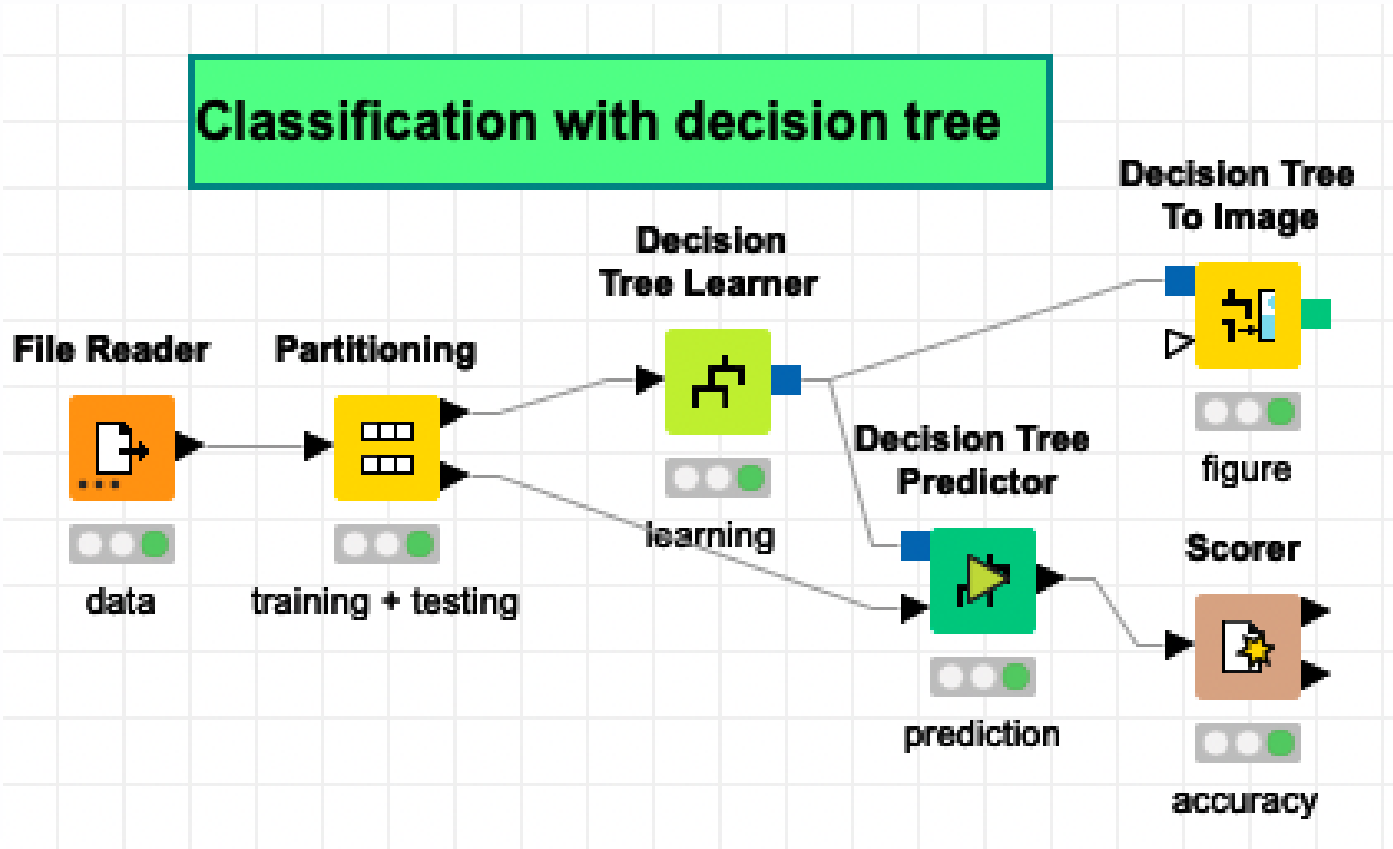
- binární klasifikace – Gini nebo $\mathcal{E}$
- více tříd – $\mathcal{E}$

Informační zisk

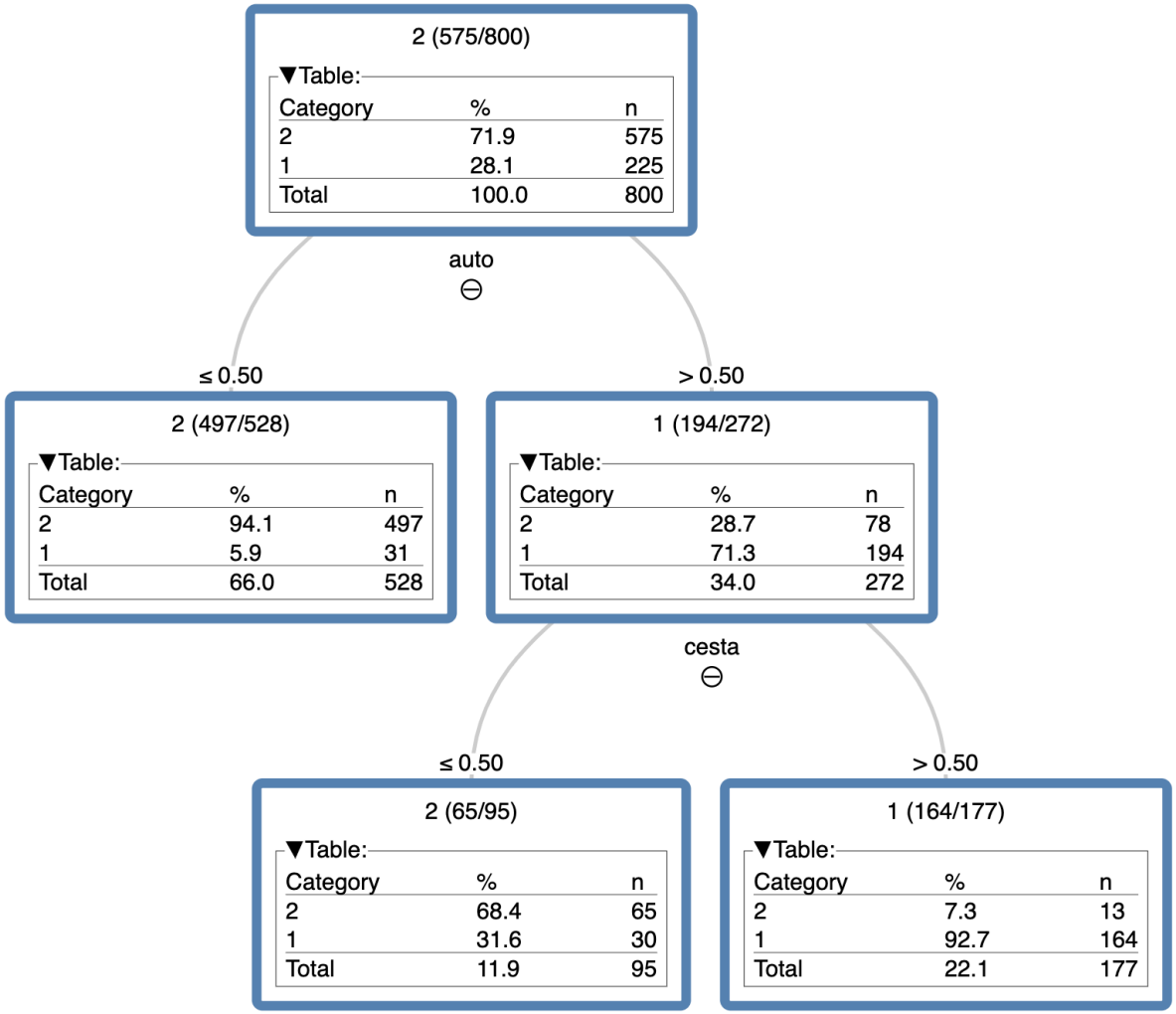
$$\text{IG} = M(y) - \frac{|y_1|}{|y|} M(y_1) - \frac{|y_2|}{|y|} M(y_2) - \ldots - \frac{|y_K|}{|y|} M(y_K)$$

- M = Gini nebo $\mathcal{E}$
- y – data, $y_1, \ldots, y_K$ – data po rozkladu (mají/nemají auto atd)
- $|y|$ – počet prvků
- větší IG - lepší rozklad

Program v KNIME



Rozhodovací strom v KNIME



Correct classified: 180	Wrong classified: 20
Accuracy: 90%	Error: 10%
Cohen's kappa (κ): 0,751%	

- 800+200 dat
- rozklad: MHD ($c = 2$) | auto, cesta
- výsledky – predikce, chyba predikce

Nedostatky a výhody rozhodovacích stromů

Nedostatky

- náchylné k přeučení (trénování – dobře, testování – špatně)
  - co pomáhá: pre-pruning (prořezávání stromu):
    - max depth – maximální hloubka
    - leaf size – velikost listu (minimální počet vzorků)
    - number of leaf nodes – počet listů
- výpočetní náročnost trénování (IG pro každý rozklad) spojitá data

Výhody

- predikce – výpočetně velmi nenáročná
- dobrá interpretovatelnost

- jak snížit chybu predikce

Náhodné lesy

Konstrukce náhodného lesu – metoda Bootstrap Aggregation

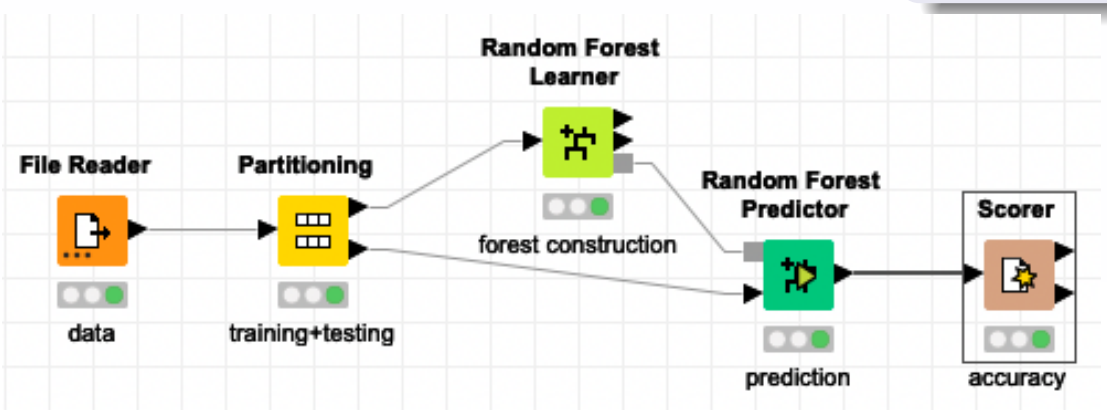
- berou se 10 – 100 náhodných vzorků z trénovacích dat
- náhodný les = 10 – 100 rozhodovacích stromů
- každý strom v lese se učí
- kombinace výsledků stromů – agregace (podle většiny) nebo průměr

Výhody

- fungují dobře bez ladění

Nedostatky

- špatná interpretovatelnost
- pomalé sestavení lesa
- pomalejší predikce

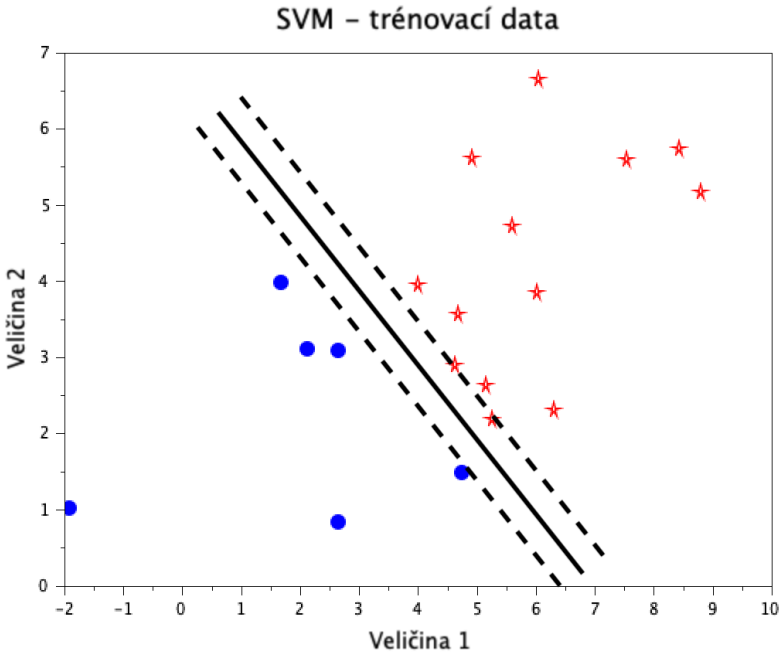


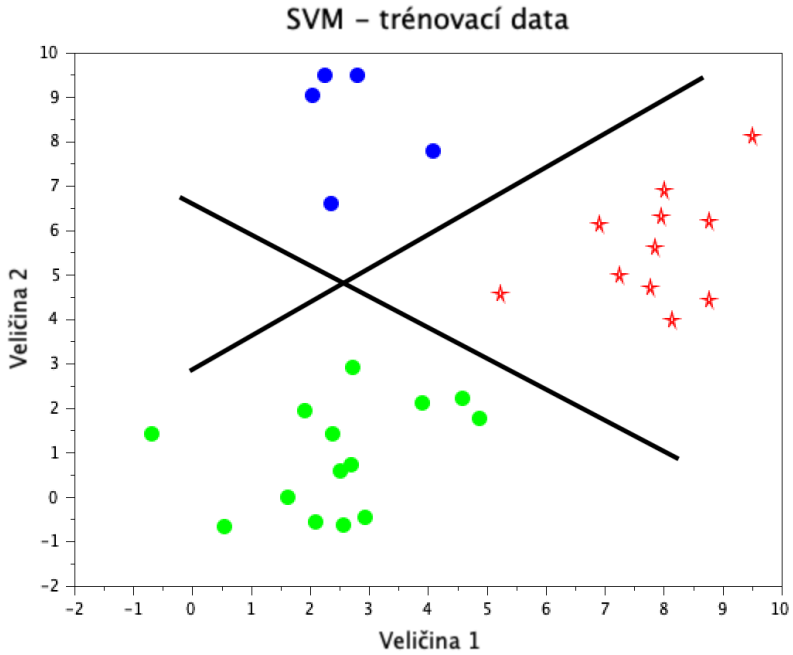
Correct classified: 189	Wrong classified: 11
Accuracy: 94,5%	Error: 5,5%
Cohen's kappa (κ): 0,85%	

Podpůrné vektorové stroje (SVM)

Lineární klasifikace – princip:

- každý datový bod – vektor např. $y_t = [y_{1;t;}, y_{2;t}]'$
- potřebujeme body oddělit pomocí nadroviny
- optimální nadrovina – maximální vzdáleností bodů

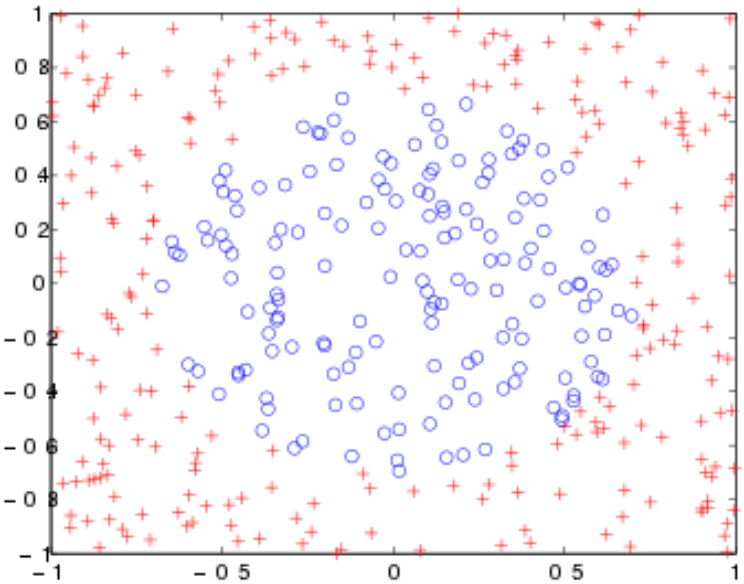


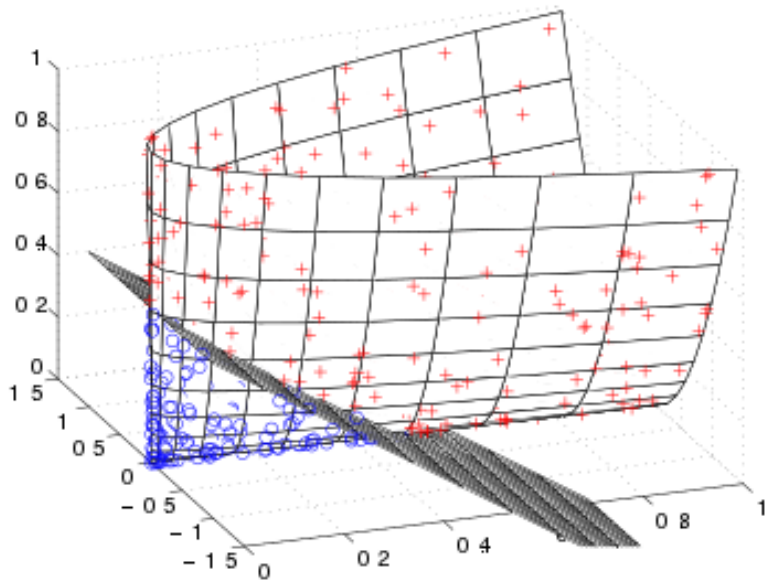


- hraniční pásmo, podpůrné vektory – body na hranici pásma
- $\omega y - b = 0$ – optimální nadrovina, ω, b – optimalizace

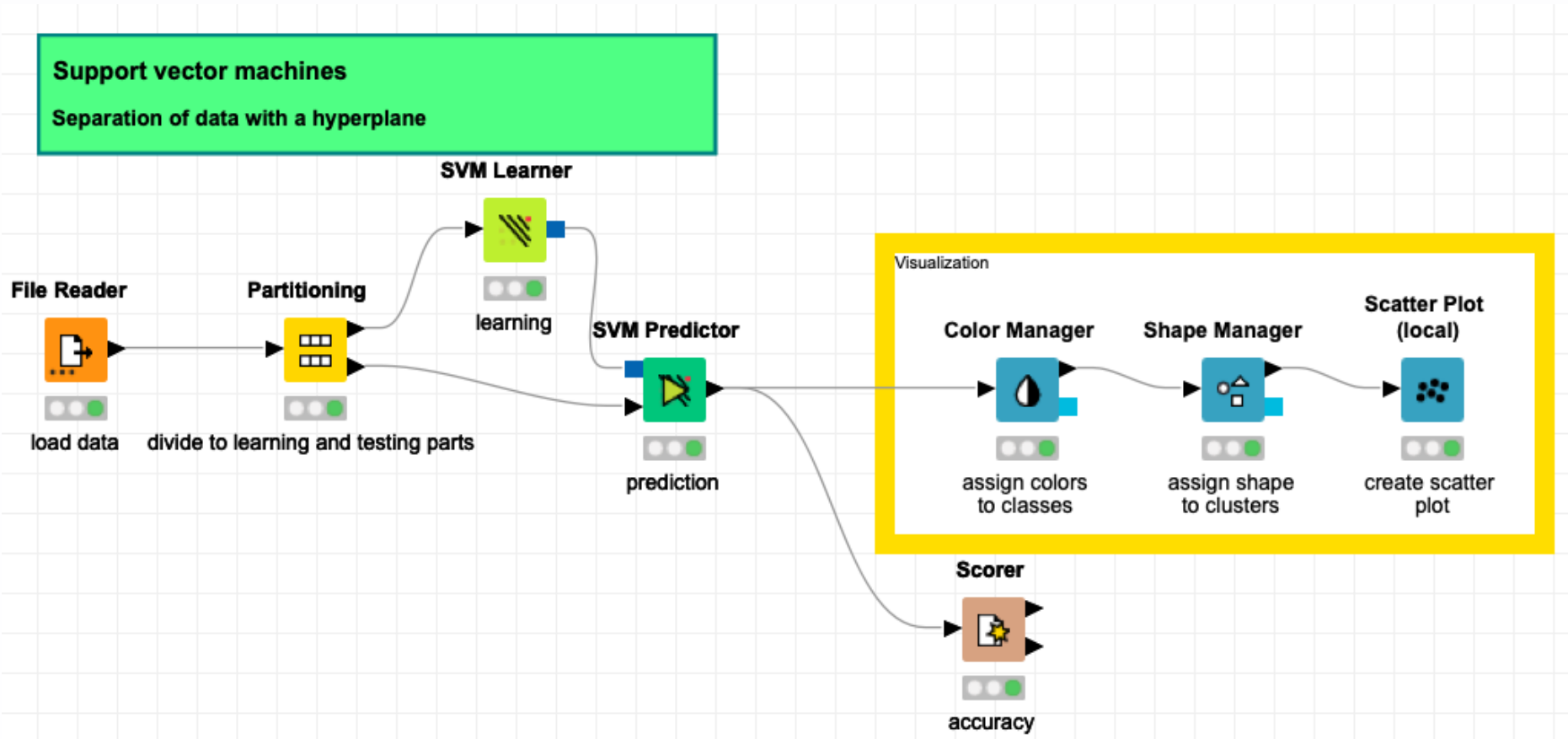
Nelineární případ

- data nejsou lineárně separovatelná (např., zašuměná)
- transformace dat do prostoru vyšší dimenze, ve kterém již jsou lineárně separabilní
- jádrová funkce (např., polynom) – numerická optimalizace



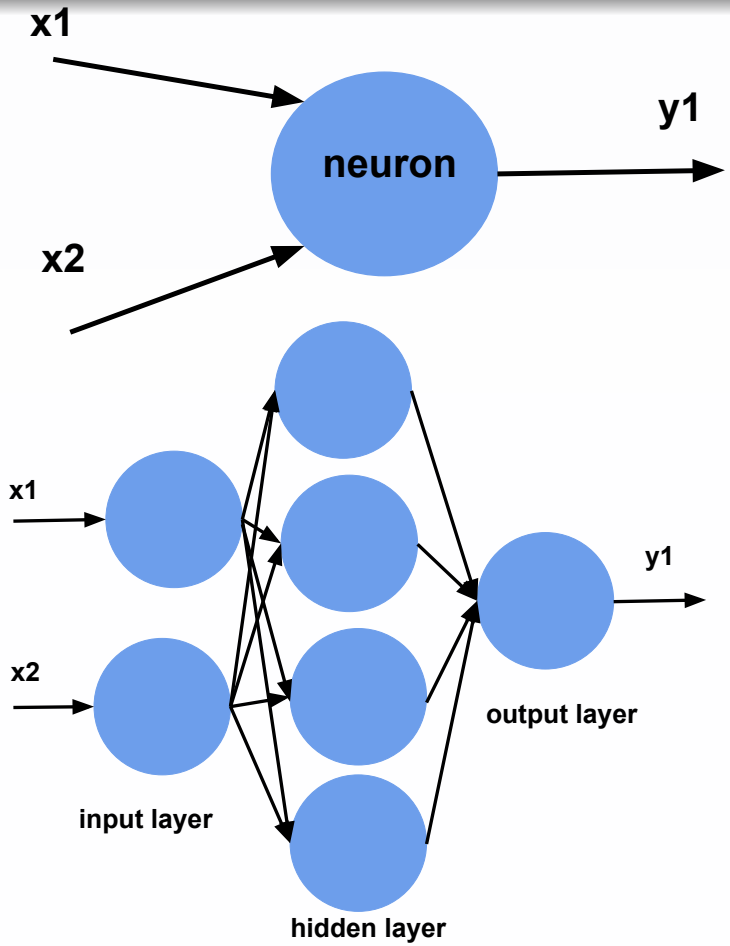


Program v KNIME



Correct classified: 184	Wrong classified: 16
Accuracy: 92%	Error: 8%
Cohen's kappa (κ): 0,79%	

Neuronové sítě



- výpočet v neuronu – aktivační funkce

$$y = f\left(\sum_{i=1}^{K} w_i x_i + b\right)$$

- w_i, b – parametry (váhy)

Tři běžně používané aktivační funkce

$$\text{sigmoid}(A) = \frac{1}{1+e^{-A}}$$

$$\tanh(A) = \frac{\sinh(A)}{\cosh(A)} = \frac{e^{A}-e^{-A}}{e^{A}+e^{-A}}$$

$$\text{ReLU}(A) = \begin{cases} 0 & \text{if } A \leq 0 \\ A & \text{if } A > 0 \end{cases}$$

$$A = \sum_{i=1}^{K} w_i x_i + b$$

- skryté vrstvy – vícevrstvé perceptrony (multi-layer perceptron MLP)

Trénování

- update parametrů – zpět od výstupu ke vstupu na základě chyby predikce

Zpětné šíření chyby (backpropagation)

$$\sum_{j=1}^{n_d} \sum_{i=1}^{K} (\hat{y}_j - y_t)^2 = \sum_{j=1}^{n_d} e_j \to \min$$

$$\Delta w = -\alpha \frac{\partial e_j}{\partial \omega}$$

Gradientní sestup (hledání minima)

Δw – přírůstky vah, α – velikost gradientního kroku

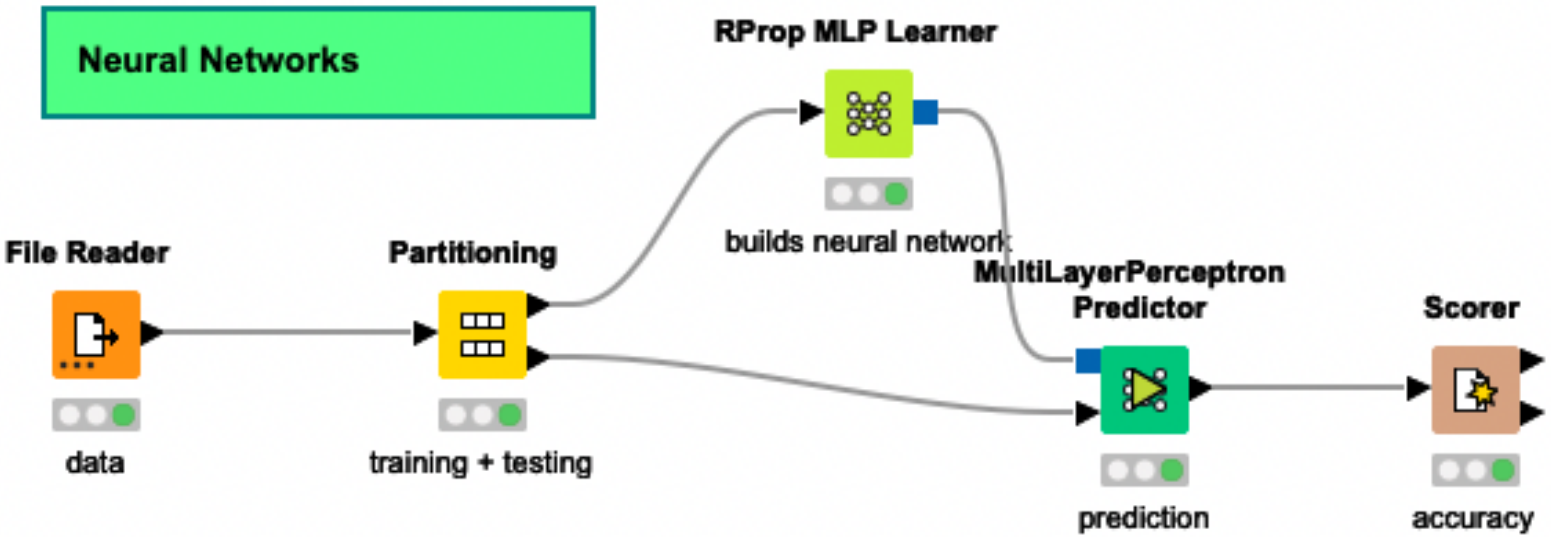
Program v KNIME

Výhody

- snadné rozšíření na více tříd
- vyšší výkonnost (ladění parametrů)
- vyšší výkonnost s rozsáhlými daty
- rozšíření – deep learning
  (3 a více skryté vrstvy)

Nedostatky

- špatná interpretovatelnost
- vyšší výpočetní náročnost



Correct classified: 179	Wrong classified: 21
Accuracy: 89,5%	Error: 10,5%
Cohen's kappa (κ): 0,758%	