

Pokročilé neparametrické metody



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



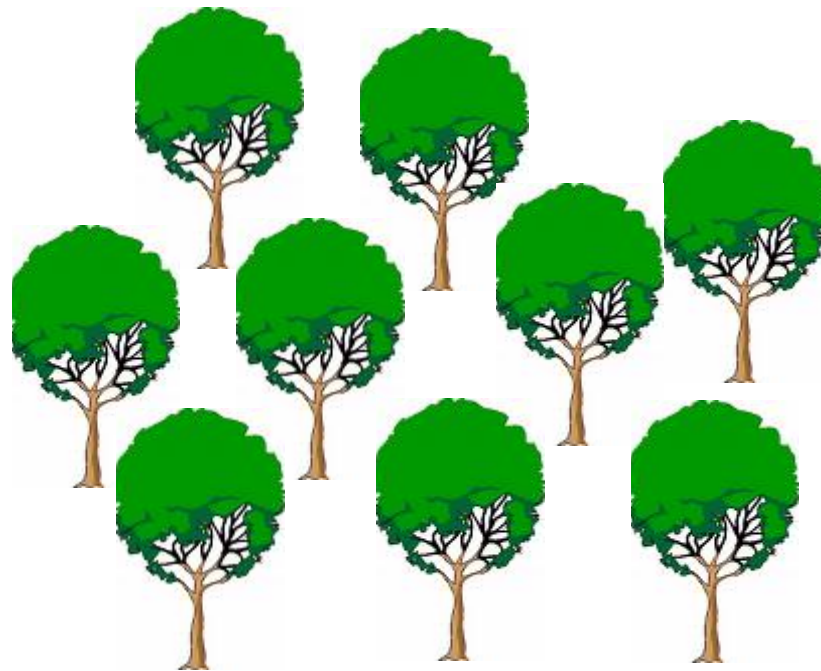
MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Klasifikační a regresní lesy

Klasifikační a regresní lesy

Klasifikační les

- Klasifikační les je klasifikační model vytvořený kombinací určitého počtu klasifikačních stromů.
- Každý strom přiřazuje hodnotě vektoru prediktorů nějakou třídu.
- Výsledná klasifikace je dána hlasováním nebo jako průměr pravděpodobnosti (zastoupení kategorie v terminálním uzlu)

Regresní les

- Je tvořen několika regresními stromy.
- Výsledná regresní funkce je definována jako vážený průměr regresních funkcí několika stromů.

Rozhodovací lesy

- James Surowiecki, 2004

„skupinový úsudek je daleko inteligentnější a přesnější než úsudek jednotlivce, v případech, kdy jde o hodnocení faktů“

- každý příslušník davu musí činit svůj úsudek na základě vlastních, nezávislých informací
- Výsledek je dán hlasováním
- **Stromy nejsou nezávislé**

Metody vytváření lesů

- Random forests (L. Breiman, 2001)
- Bagging – bootstrap aggregating (L. Breiman, 1996)
- Boosting (Y. Freund & R. E. Schapire, 1997)

Do lesa není vidět !



Bagging – bootstrap aggregating

- Leo Breiman (1996)
- Technika pro zvýšení stability a přesnosti u klasifikačních a regresních metod
- nejčastěji používaná pro rozhodovací stromy, ale použitelná i pro ostatní modely
- Z trénovacího souboru L (o velikosti n) se náhodným výběrem s vrácením vytvoří k souborů L_1, \dots, L_k o velikosti $n' \leq n$ (vytvoříme k bootstrapových výběrů)
- Každý z výběrů se použije na tvorbu klasifikačního (regresního) stromu
- Výsledný klasifikační les je dán většinovým hlasováním se stejnými váhami nebo jako průměr pravděpodobnosti, u regresních lesů pak aritmetickým průměrem dílčích regresních funkcí.

Boosting (to boost = zesilovat) algoritmus AdaBoost (adaptive boosting)

- Mějme klasifikační strom T vytvořený na základě trénovacích dat, kdy každému případu z trénovací množiny je přiřazena určitá váha.
- Konstruujeme posloupnost rozdílných stromů T_1, \dots, T_K tak, že podle předchozích výsledků postupně upravujeme váhy případů.
- V 1. kroku je zadán vektor vah (např. rovnoměrné váhy.)
- V dalších krocích: pro konstrukci T_i se váhy případů, které byly špatně klasifikovány stromem T_{i-1} zvýší, váhy správně klasifikovaných případů se sníží
- pozornost je soustředěna na případy, které jsou špatně klasifikovány
- je-li v datech obsaženo malé množství informace, vede boosting k tomu, že se učíme opakovat chyby v datech

Random Forests pro regresi a klasifikaci

“ looking inside the black box is necessary”

Leo Breiman



Použití RF

- 😊 měření významnosti proměnných
- 😊 efekt proměnných na predikci
- 😊 shlukování
- 😊 detekce odlehlých hodnot
- 😊 klasifikace
- 😊 predikce

Random Forests I.

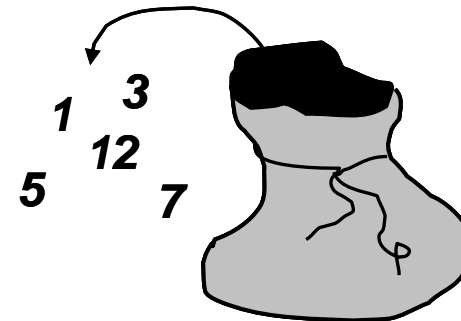
- Náhodný les se skládá ze souboru stromů T_1, \dots, T_N , jejichž klasifikační nebo regresní funkce lze vyjádřit jako

$$h(X, \Theta_1), \dots, h(X, \Theta_N),$$

- kde h je funkce, X je prediktor a $\Theta_1, \dots, \Theta_N$ jsou nezávislé stejně rozdělené náhodné vektory.
- Pro metodu Random Forests se používají binární stromy typu CART.
- Podobně jako při tvorbě jednotlivých stromů se i zde používá rozdělení na testovací a trénovací soubor.

bootstrapové výběry

- Trénovací soubory pro jednotlivé stromy T_i jsou tzv. **bootstrapové výběry** z datového souboru L .
- Bootstrapové výběry jsou náhodnými **výběry s opakováním** o velikosti n .



- Takovýto výběr nesnižuje postupně velikost množiny původních pozorování → lze rozdělit i velmi malé soubory na velký počet trénovacích a testovacích souborů.

bootstrapové výběry

- jednotlivé bootstrapové výběry nejsou nezávislé jako například u krosvalidace → do bootstrapového výběru jsou některá pozorování vybrána opakovaně a některá naopak vůbec.
- počet pozorování, která se do bootstrapového výběru nedostanou je přibližně 37%
- pozorování, která jsou v i -tém bootstrapovém výběru L_i , se použijí při tvorbě stromu T_i → trénovací soubor
- pozorování, která se do toho výběru nedostala jsou použita k odhadu jeho chyby → testovací soubor
- odhady chyby na testovacím souboru se nazývají *oob* (*out-of-bag*, *out of bootstrap sample*) odhady
- celkový počet *oob* pozorování tvoří 1/3 datového souboru

Přiřazení hodnoty v RF

- Klasifikace $\hat{C}_{rf} = \text{vetsinove_hlasovani} \{ \hat{C}_i(x) \}_1^N$

- kde $\hat{C}_i(x)$ je výsledek klasifikace i -tého stromu

- Regrese $\hat{f}_{rf}(x) = \frac{1}{N} \left(\sum_{i=1}^N T_i(x) \right)$

- predikce je průměrem ze všech stromů

- náhodný les zvyšuje přesnost (**snižuje zkreslení**) →
pěstováním se velkých stromů, které se neprořezávají →
cílem je snížit chybu celého lesa nikoli jednotlivého stromu

- zároveň udržuje „rozumnou“ **varianci** kombinováním výsledků
jednotlivých stromů (většinové hlasování/průměrování)

Stromy nejsou nezávislé – co s tím?

- Jak zajistit také nízkou korelaci mezi jednotlivými stromy ?
- Pokud tvoříme výběry s opakováním, které nejsou navzájem nezávislé, budou výsledné stromy korelované → může vést k nadhodnoceným výsledkům klasifikace či predikce!
- Snížení korelace mezi stromy se dosáhne náhodným výběrem pouze **určitého počtu prediktorů**.
- Pro každý strom se tak nejlepší větvení pro daný uzel hledá pouze z m prediktorů X_1, \dots, X_M .
- **Random Forests používá jak náhodný výběr pozorování, tak náhodný výběr prediktorů.**

Algoritmus tvorby lesa

- **Krok1:** Vytvoř bootstrapový podsoubor L_i o velikosti N - trénovací soubor.
- **Krok2:** Vyber náhodně m prediktorů.
- **Krok3:** Vytvoř strom T_i na bootstrapovém souboru L_i **pouze s použitím m náhodně vybraných prediktorů** (stejně jak bylo popsáno v metodě CART, hledáme nejlepší rozdělení daného uzlu mezi prediktory na dva dceřiné uzly). Růst stromu se zastaví, až strom dosáhne minimální hodnoty velikosti uzlu.
- **Krok 4:** Zařaď *oob* pozorování (testovací soubor) vytvořeným stromem a urči výslednou klasifikační třídu (kategorii) nebo predikci všech *oob* pozorování.

Krok 1-4 se opakuje do konečného počtu stromů v lese.

- Spočítej celkový výsledek klasifikace/predikce celého lesa většinovým hlasováním/průměrováním.

Výběr m a n_{tree}

- je potřeba vybrat správný počet proměnných (m) pro náhodný výběr a počet stromů (n_{tree}) v lese.
- Určení těchto parametrů je experimentální
- otestování parametrů → spuštění RF několikrát s různým nastavením parametrů → snažíme se získat les, který má nejmenší celkovou chybovost.
- Počet stromů n_{tree} - po určitém čase začínají stromy konvergovat ke správné hodnotě *oob* odhadu.
- Minimální velikost lesa lze určit jako počet stromů, kdy se chyba *oob* odhadu s přibývajícím stromy již nemění.

Počet prediktorů

Pro náhodné lesy je doporučeno následující nastavení:

- klasifikace - hodnota $m = \sqrt{p}$ a minimální velikost uzlu je jedna;
- regrese - hodnota $m = p/3$ a minimální velikost koncového uzlu je pět

- Výše uvedené hodnoty slouží jako defaultní nastavení ve většině softwarů.

- V praxi však určení počtu prediktorů závisí na řešeném problému a parametr m je vhodný zvolit podle výsledků testování modelů s různým nastavením.

- Vybereme takové m , při němž má výsledný les nejmenší chybovost.
- Vzhledem k tomu, že stromy nelze přetrénovat, je počet prediktorů nejdůležitější hodnotou, kterou musíme zvolit, neboť počet stromů nás omezuje pouze časově.

Měření významnosti proměnných (*importance*)

- Metoda přínosná u problémů s velkým množstvím prediktorů, které mnohdy obsahují málo informací o závisle proměnné

Náhodné lesy vracejí několik měření významnosti (*importance*) proměnné

- Dva typy měření významnosti z RF:
 - založená na poklesu klasifikační přesnosti (*misclassification rate*), kdy jsou hodnoty prediktoru náhodně permutovány
 - významnost lze také spočítat pomocí Gini indexu.

Významnost - *misclassification rate*

- nejčastější měření *importance*:

měření založené na poklesu klasifikační přesnosti, když hodnoty proměnné v uzlu stromu jsou permutovány náhodně

Procedura se dá popsat následovně:

- vytvoříme bootstrapové výběry
- z M - prediktorů vybereme m_0
- po vytvoření jednotlivých stromů, jsou hodnoty m -té proměnné z OOB (pozorování, která zůstala mimo výběr) vzorku náhodně permutovány

Random forest – *misclassification rate*

- u každého vzorku z OOB výběru je pomocí příslušného klasifikačního stromu zjištěn výsledek klasifikace, opakováno pro $m = 1, \dots, M$
- Na konci jsou srovnány výsledky pozorování u m -tého prediktoru zatíženého šumem (randomizovaného) se správnou klasifikací těchto pozorování.
- Pokles v přesnosti predikce stromu, který nastane po randomizaci pozorování, je zprůměrován přes všechny stromy → hodnoty MR (*misclassification rate*) pro každý prediktor, které určují jeho významnost
- Toto měření je často vyjádřeno jako procento a je standardizováno na maximální hodnotu MR nejvýznamnějšího prediktoru (nejvýznamnější prediktor $MR = 100\%$).
- Popsaný proces se opakuje pro všechny prediktory.

Lokální významnost

- Pro každé pozorování můžeme navíc spočítat **lokální významnost** m -tého prediktoru.
- Je zjištěno procento správné klasifikace n -tého pozorování do správné kategorie přes všechny stromy, kdy bylo v *oob* výběru a hodnoty prediktoru X byly permutovány.
- Tato hodnota je odečtena od procenta správné klasifikace pro *oob* pozorování prediktoru bez randomizace.
- Měření významnosti založené na randomizaci proměnných se objevuje v různých variantách.

Významnost založená na Gini indexu

- Při rozdělení uzlu na dva dceřiné uzly prediktorem, ke kterému je použit Gini index, dochází k poklesu tohoto indexu.
- Součet poklesu v G v jednotlivých stromech pro každý prediktor udává jeho významnost.
- hledáme optimální sadu prediktorů - máme tuto informaci z významnosti proměnných?
- Sada významných proměnných nemusí být počet proměnných použitých (či nutných) pro klasifikaci/predikci, je však sadou maximální.
- Ke zjištění nejmenší možné sady parametrů při zachování celkové přesnosti lesa je nutné testovat různé kombinace proměnných.
- Můžeme dojít k více modelům o stejné přesnosti, ovšem s jinou kombinací proměnných.
- V případě velkého počtu proměnných je les na začátku spuštěn jednou se všemi proměnnými a potom znovu s použitím pouze významných proměnných, čímž se výrazně šetří čas při testování optimálního lesa.

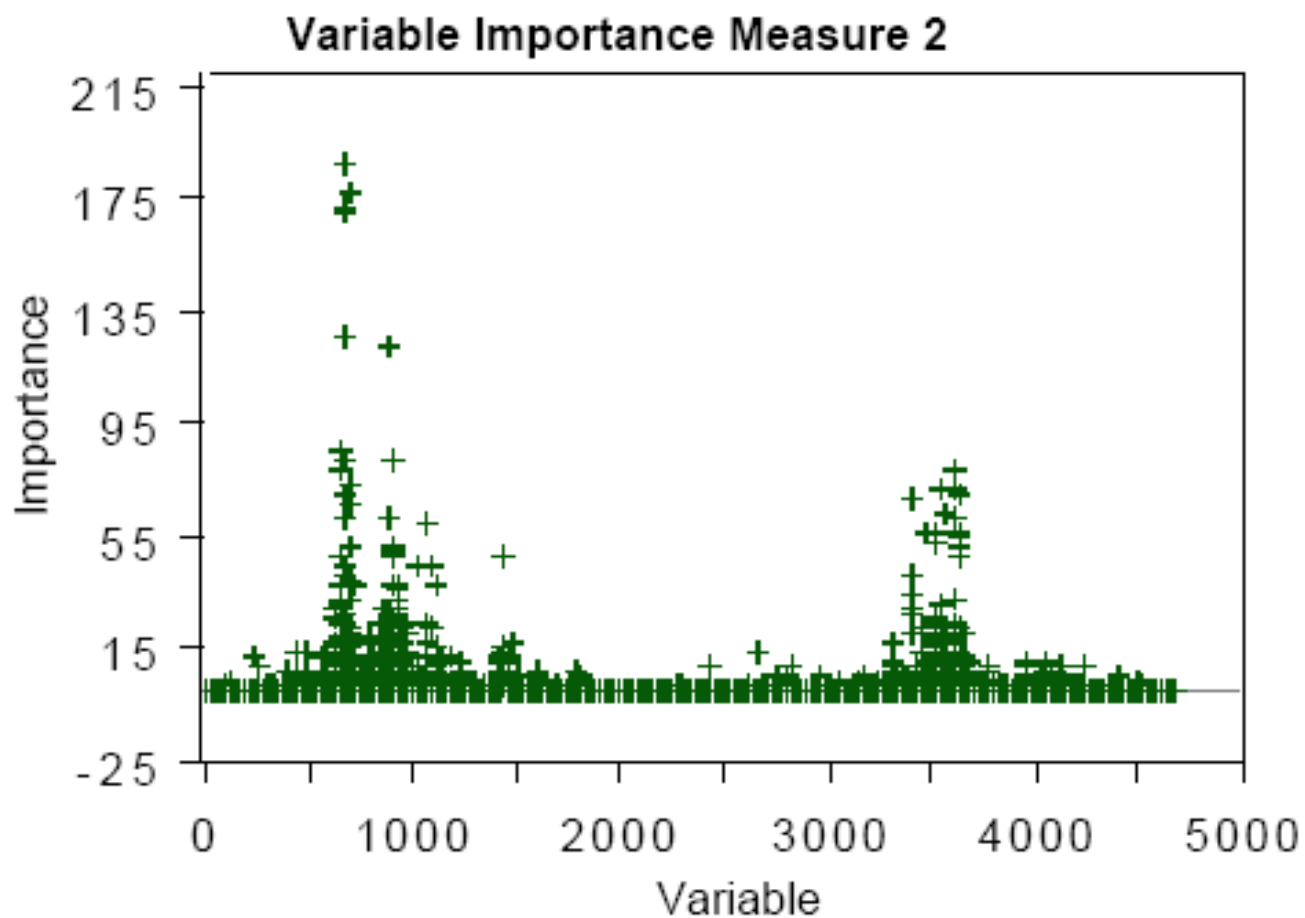
Příklad I - genetika



- RF byl použit na microarray datový soubor, celkem 81 pacientů (s leukémií) bylo rozděleno do tří tříd.
- Prediktory - 4682 genů bez jakéhokoli předešlého výběru.
- Chybovost výsledného lesa byla nízká (1.2%).
- Cílem nebyla pouze predikce nových pacientů do těchto tří skupin, ale rovněž zjištění, které geny byly pro rozdělení nejvýznamnější



Příklad I - genetika



Příklad II: Kriminalistika

- Studie různých typů klasifikací skla byla motivována vyšetřováním trestných činů. Sklo z místa činu může být použito jako důkaz...jestliže je správně identifikováno!
- Datový soubor obsahuje 214 vzorků šesti druhů různého skla a devět prediktorů – různé chemické parametry skla.
- typy skla: (class)
 - A okenní sklo (z domu)1
 - B okenní sklo (z domu)2
 - C okenní sklo (z auta)1
 - D okenní sklo (z auta)2
 - E obalové sklo
 - F sklo z nádobí
 - G reflektorové sklo
- Prediktory: index lomu (RI), Na, Mg, Al, Si, K, Ca, Ba, Fe





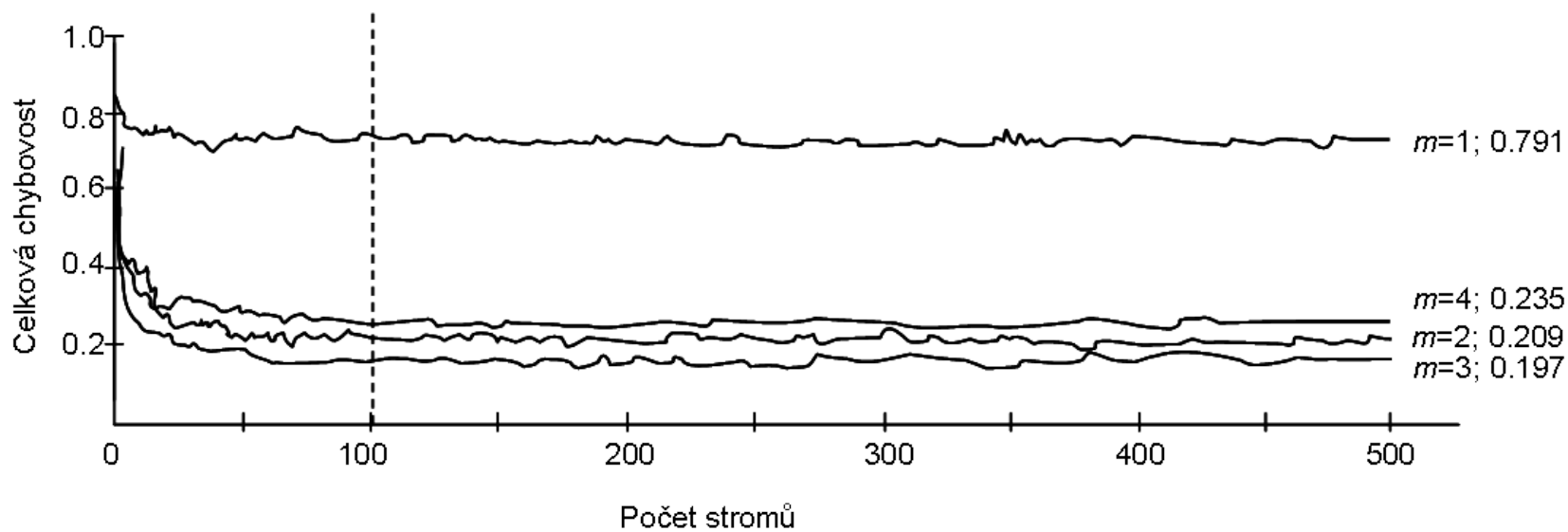
Příklad II: Parametry, které je potřeba testovat

- *n*tree – počet stromů v lese
- *m*try – počet (náhodně) vybraných prediktorů
- *n*node – počet vzorků v terminálním uzlu
- % vzorků pro testování (OOB výběr)

Výsledky testovací chyby pro různé *m*try

<i>m</i> try = 1	OOB error = 23.36%
<i>m</i> try = 2	OOB error = 25.56%
<i>m</i> try = 3	OOB error = 21.3%
<i>m</i> try = 4	OOB error = 22.9%
<i>m</i> try = 5	OOB error = 23.36%
<i>m</i> try = 6	OOB error = 26.17%
<i>m</i> try = 7	OOB error = 23.36%
<i>m</i> try = 8	OOB error = 23.36%

Příklad II: parametry $ntree$ a m





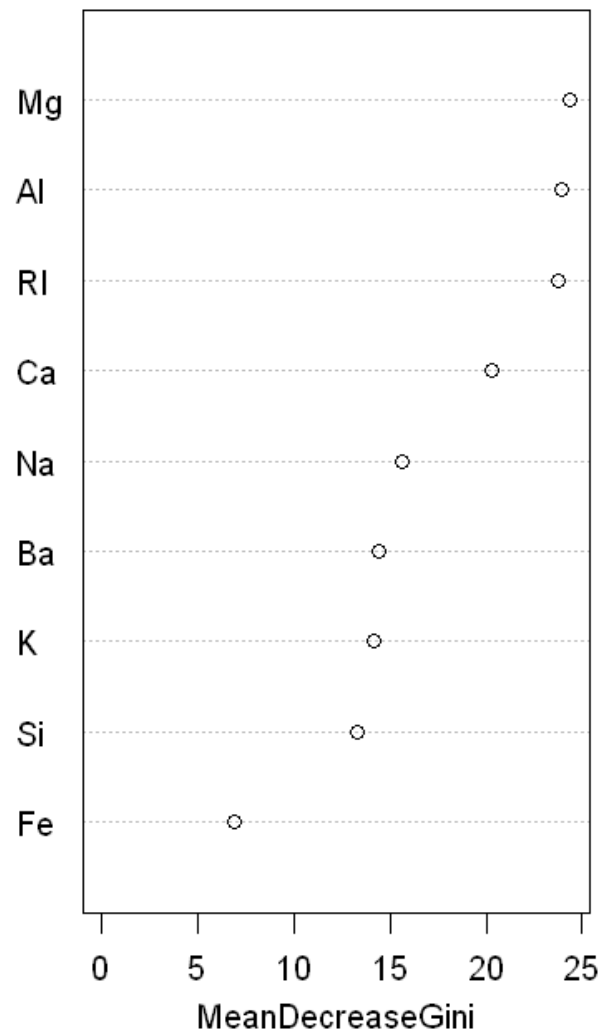
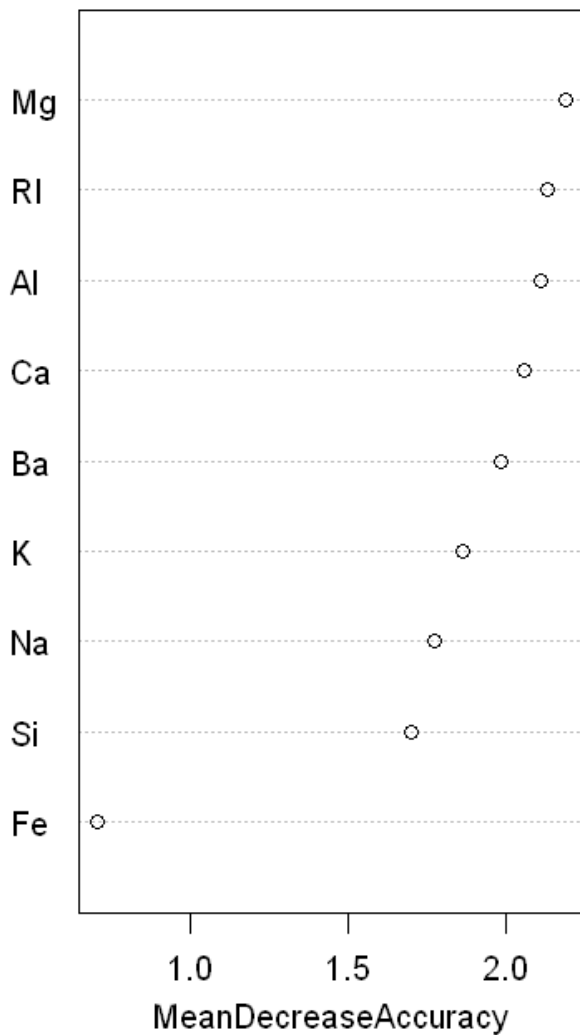
Příklad II: Výsledek klasifikace

- R-project, knihovna randomForest
- randomForest(formula = class ~ ., data = glass, importance = TRUE, proximity = TRUE)
 - Type of random forest: classification
 - Number of trees: 500
 - No. of variables tried at each split: 3
 - OOB estimate of error rate: 21.03%

Confusion matrix:

	A	B	C	E	F	G	class.error	n
A	60	8	2	0	0	0	0.1428571	70
B	10	60	1	3	1	1	0.2105263	76
C	6	4	7	0	0	0	0.5882353	17
E	0	2	0	10	0	1	0.2307692	13
F	0	2	0	0	7	0	0.2222222	9
G	1	3	0	0	0	25	0.1379310	29

Příklad II: Významnost proměnných



Efekt proměnných na predikci I

- Můžeme zjistit které kategorie nebo hodnoty prediktoru jsou významné
- Pro případy z *oob* výběru známe kategorii, do které bylo pozorování zařazeno, zjistíme podíl klasifikace pozorování do jednotlivých kategorií neboli *cpv* (*class proportion vote*)
- příklad pro čtyři kategorie A, B, C, D - pozorování při klasifikaci je ze 100 stromů zařazeno:
 - 10 stromy jako A,
 - 50 stromy jako B
 - 40 stromy jako C
 - žádným stromem jako D
- Hodnoty *cpv* pro jednotlivé kategorie budou:
 - $cpv_A=0,1$, $cpv_B = 0,5$, $cpv_C = 0,4$ a $cpv_D = 0$.

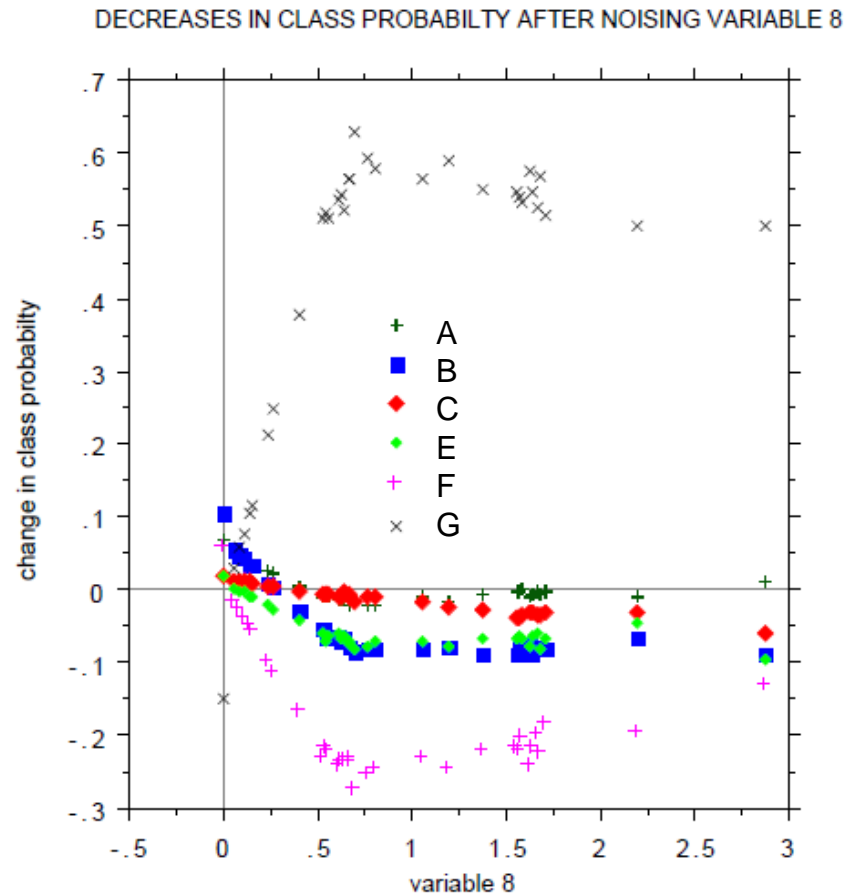
Efekt proměnných na predikci II

- Pro každou kategorii j a každou proměnnou m spočítáme pravděpodobnost zařazení každého vzorku do kategorie j , kdy je m -tá proměnná X randomizována.
- Rozdíl mezi cpv_{Ran} pro randomizovanou proměnnou a proměnnou bez šumu udává velikost změny pravděpodobnosti zp_{cpv} pro každou kategorii u všech vzorků:

- $$zp_{cpv} = cpv - cpv_{Ran}$$

- Myšlenka je tedy stejná jako při určení významnosti proměnné.
- Pokud vyneseme hodnoty této změny do grafu proti hodnotám proměnné, získáme graf efektu proměnné na predikci.

Příklad III: Pokles v pravděpodobnosti kategorií po randomizaci proměnné Fe



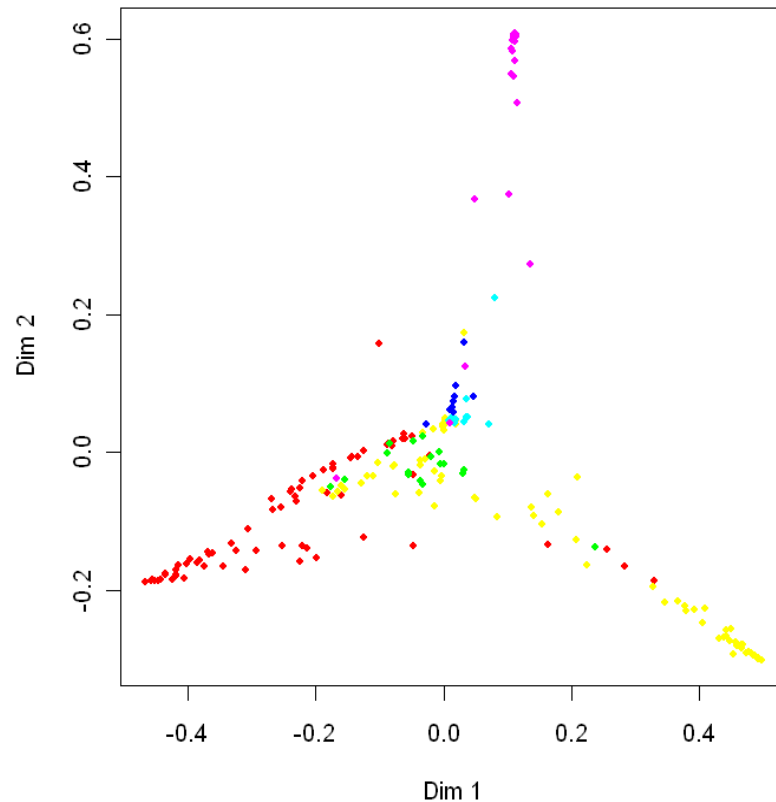
Matice těsnosti (*proximity*)

- neprořezané stromy – konečný počet uzlů obsahuje pouze malý počet vzorků
- všechny vzorky z trénovacího souboru necháme zařadit stromem- jestliže vzorky leží v jednom uzlu, zvýší se *těsnost* (proximity) mezi nimi na jedna; $\text{prox}(n,k)$
- po proběhnutí procedury podělíme matici *proximities* celkovým počtem stromů → získáme **matice těsnosti** $M_{\text{prox}} = \{\text{prox}(n, k)\}$.
- Velikost matice je $N \times N$ - je symetrická, pozitivně definitní a nabývá hodnot mezi 0-1 s prvky na diagonále rovnými jedné.
- Můžeme převést na matici vzdáleností $1 - M_{\text{prox}}$
- Používá se v různých shlukovacích a ordinačních metodách.
- Nejčastější využití je pro výpočet faktorových os ve vícerozměrném škálování, které umožňuje projekci vzorků do prostoru s méně dimenzemi, přičemž zachovává vzdálenosti mezi objekty.
- užitečné pro vizualizaci výsledků klasifikace nebo k hodnocení překryvu jednotlivých skupin

Překryv kategorií - *proximity a probability heat map*

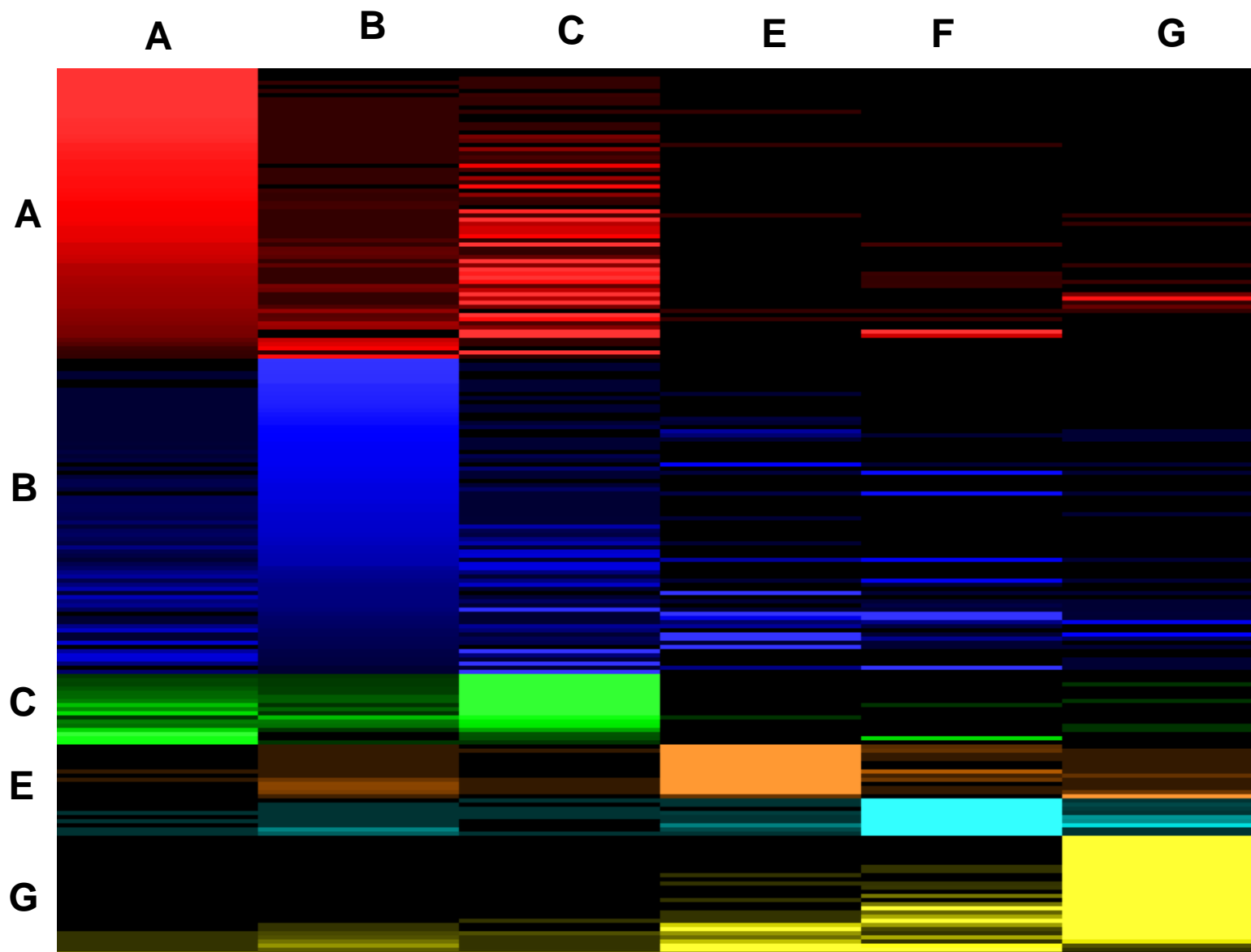
- K určení překryvu jednotlivých kategorií můžeme použít zobrazení mapy těsnosti (*proximity heat map*).
- Intenzita barev v mapě je dána vzrůstající mírou těsností jednotlivých vzorků.
- Stejně lze použít k vytvoření mapy i matici pravděpodobností (*probability heat map*).

Příklad II: Shlukování



- nemetrické mnohorozměrné škálování probíhá na matici těsnosti (proximities)
- matici těsnosti je možné použít pro všechny typy ordinačních a shlukových analýz, neboť má všech vlastnosti matice vzdáleností

Příklad II - Pravděpodobnostní mapa – překryv tříd skla



Prototypy kategorií – pomocí těsnosti

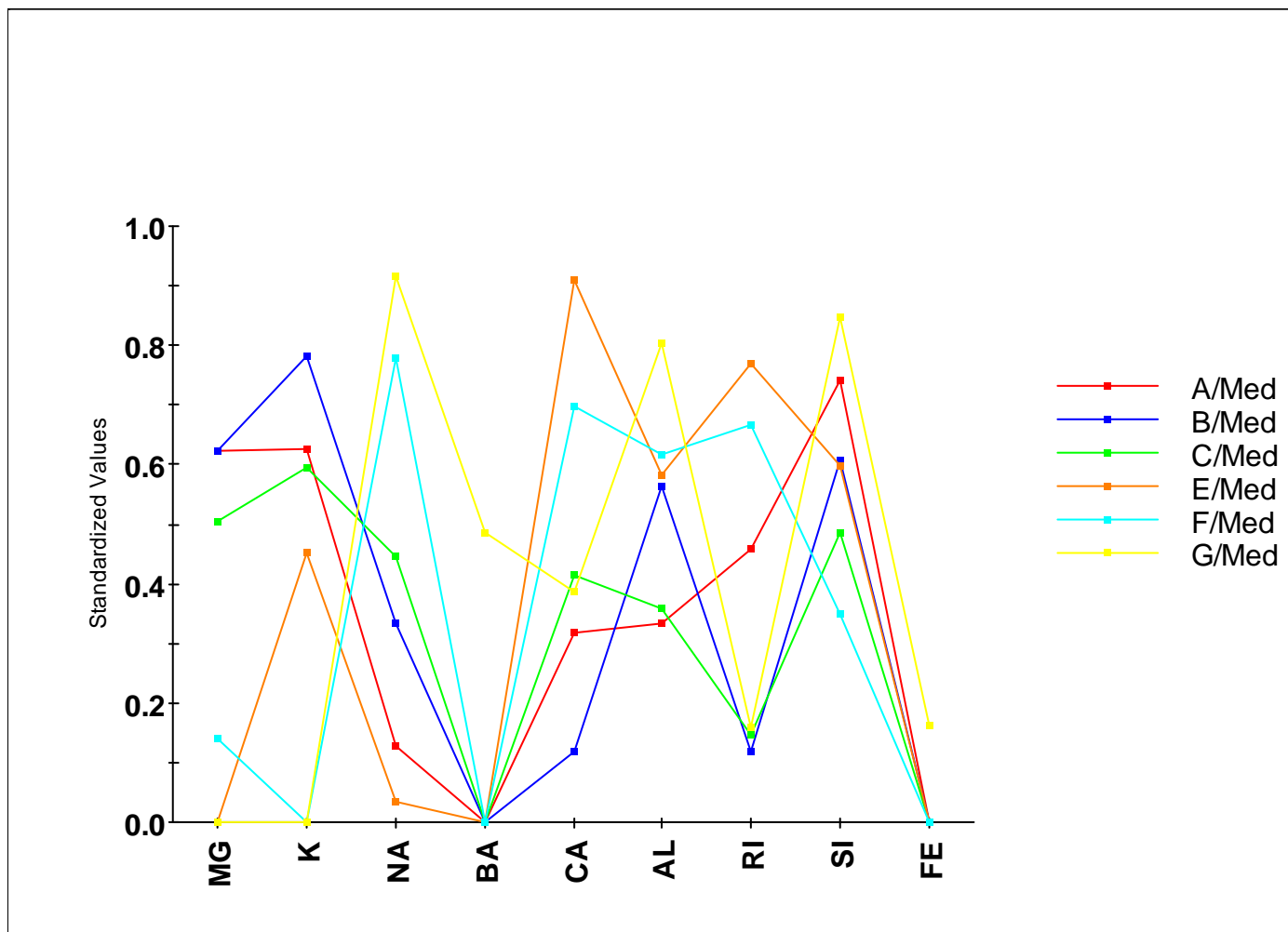
- Dalším příkladem použití měření těsnosti jsou prototypy kategorií.
- Pro každou kategorii j najdeme pozorování, u něhož je největší počet pozorování z téže kategorie mezi jeho k nejbližšími sousedy, podle měření těsnosti.
- Mezi těmito k -pozorováními zjistíme medián a kvartily pro každý prediktor.
- Mediány jsou prototypy dané kategorie a kvartily nám dávají odhad jejich stability.
- Hodnoty jsou standardizovány (odečtením 5. percentilu a podělením rozsahem mezi 5. a 95. percentilem).
- Pro kategoriální proměnné je prototypem kategorie s největší frekvencí. Druhý prototyp můžeme získat zopakováním této procedury, ale již bez pozorování, která byla obsažena v prvním prototypu.

Prototypy kategorií – pomocí pravděpodobnosti

- místo měření těsnosti použít také pravděpodobnost zařazení do správné kategorie.
- Zvolíme si hodnotu této pravděpodobnosti například větší než 0,5 (pozorování bylo s více než 50% zařazeno správně) nebo 1 (pozorování bylo všemi stromy klasifikováno do správné kategorie).
- Z pozorování, která mají pravděpodobnost zařazení do správné kategorie větší než zadaná hodnota, můžeme opět spočítat **medián a kvartily** (v případě spojitého prediktoru) nebo **procentuální zastoupení** (pro kategoriální prediktor).
- Prototypy tak udávají celkovou představu o tom, jaký vztah mají prediktory ke klasifikaci/predikci a tvoří „jádro“ dané kategorie.



Příklad II – pokračování : Prototypy kategorií



Prediktory jsou seřazeny podle významnosti

Detekce odlehlých hodnot

- vzorky, které mají nejmenší *proximities* ke všem ostatním vzorkům
- definován pouze pro třídu, do které patří
- definujme průměrnou těsnost pozorování n od všech pozorování k ve stejné kategorii j jako:

$$\bar{P}(n) = \sum_{c(k)=j} prox^2(n, k)$$

- Míra odlehlosti pro pozorování n je definována jako:

$$out(n) = N_j / \bar{P}(n)$$

kde N_j je počet pozorování v kategorii j

- Pro všechna pozorování v dané kategorii spočítáme medián ze všech $out(n_j)$ a jejich absolutní odchylku od mediánu. Odečtením mediánu od hodnoty $out(n)$ a podělením její absolutní odchylkou získáme finální měření odlehlosti pozorování n .
- $out(n) < 0$ jsou převedeny na nulu
- $out(n) > 10$ je považováno za odlehlou hodnotu

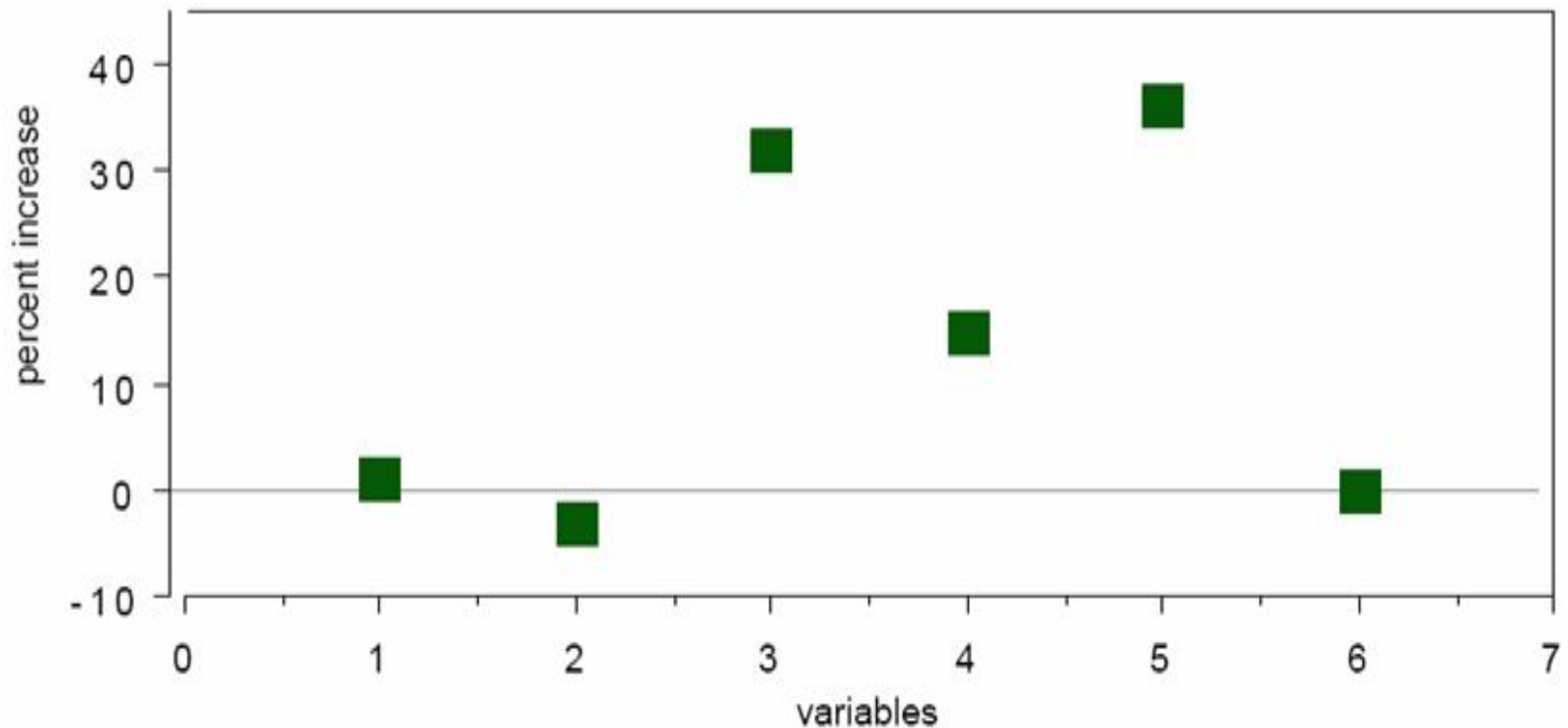
Příklad III: Biomedicína



- Soubor dat se skládá se z šesti proměnných. Prediktory jsou výsledky krevních testů, které mají vztah k fungování jater a poslední proměnnou je spotřeba alkoholu na den. Celkem 345 pacientů, bylo zařazeno do dvou tříd podle závažnosti poruchy funkce jater.
- Procento správně klasifikovaných vzorků z RF bylo 28%



Příklad III: Významnost proměnných

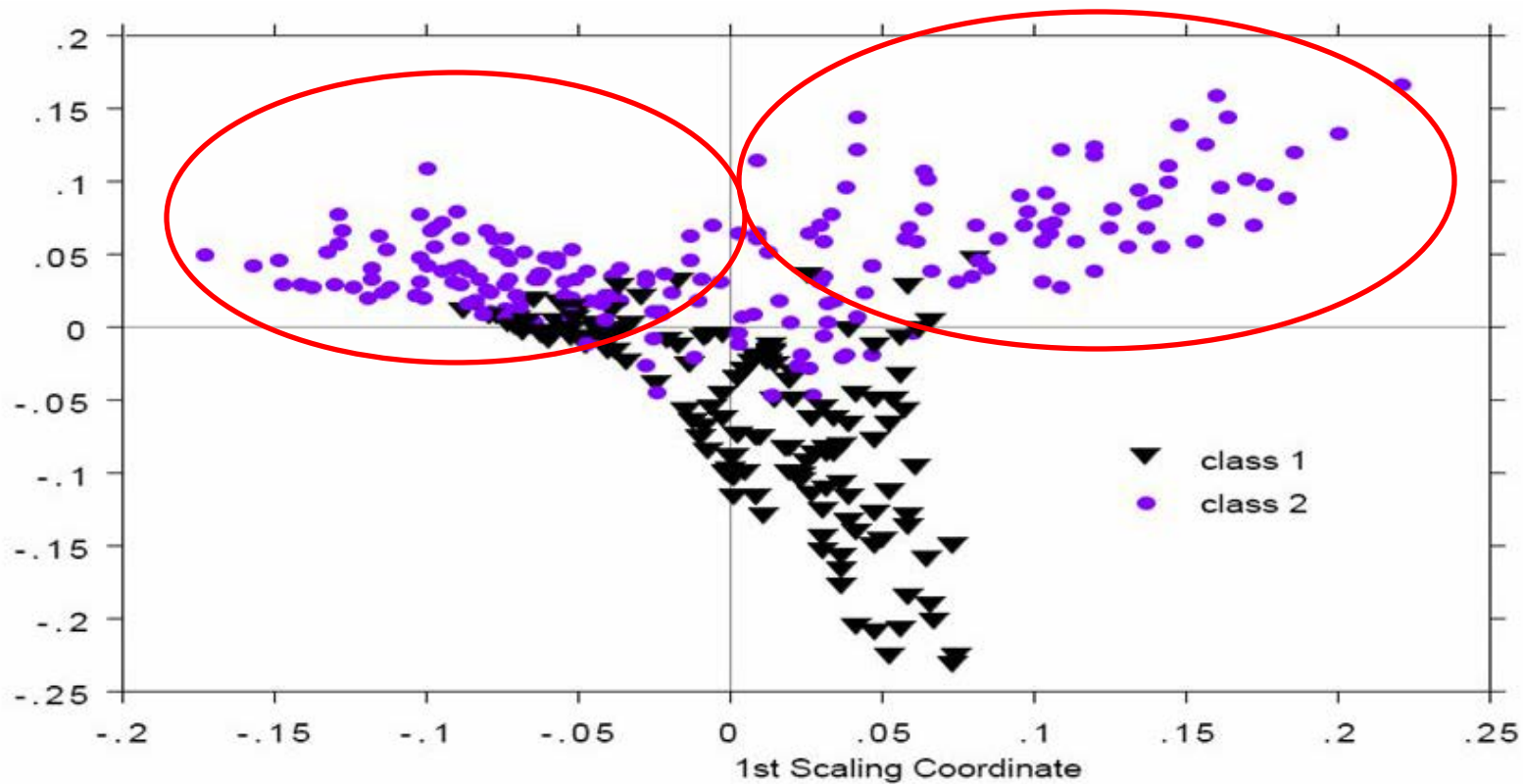


Proměnné 1-5 krevní testy; 6 spotřeba alkoholu na den

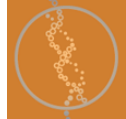
? Alkohol nemá vliv ?



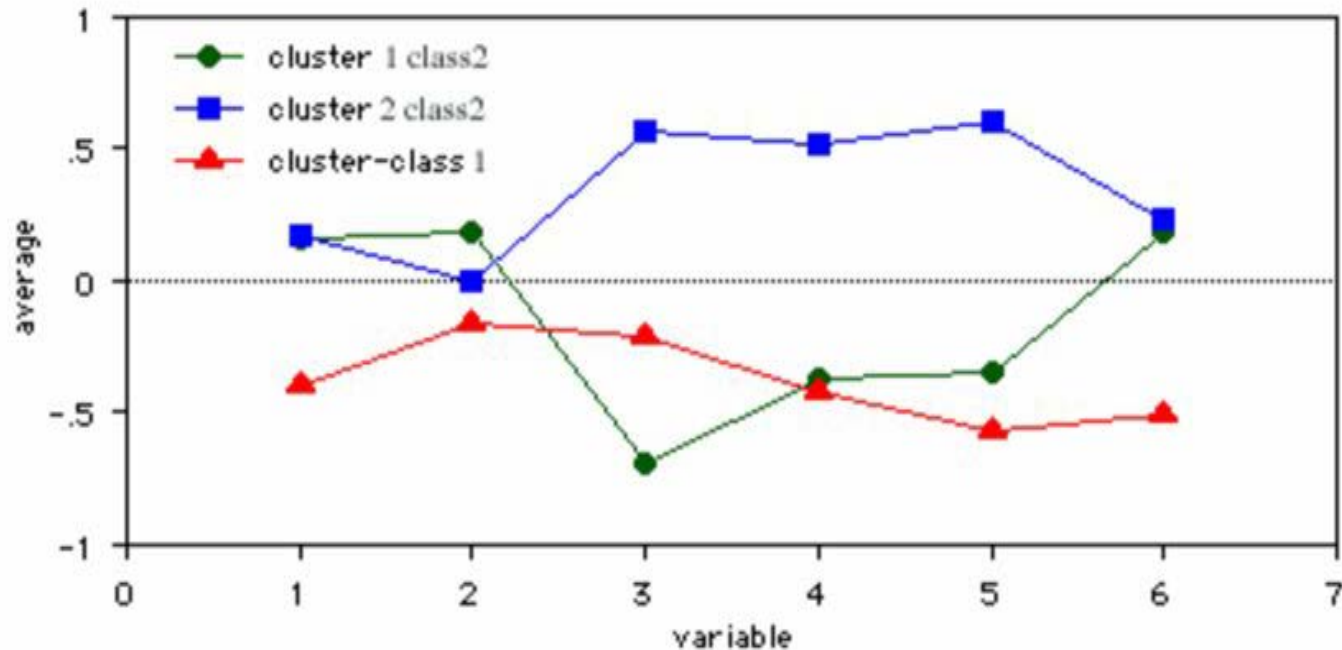
Příklad III: metrické mnohorozměrné škálování – matice těsnosti



Třídu 2 je možné rozdělit na další dvě kategorie



Příklad III: Významnost proměnných na predikci II



Spotřeba alkoholu na den již významný prediktor !

Chybějící hodnoty

- dvě možnosti, jak les provádí náhradu chybějících hodnot
- 1. varianta
 - **Spojité proměnné** - nahrazení chybějící hodnoty x_n mediánem hodnot m -té proměnné v kategorii j závisle proměnné
 - **Kategoriální proměnné** - k doplnění je použita hodnota kategorie s nejvyšší frekvencí opět pouze v příslušné kategorii závisle proměnné
 - jednodušší, rychlejší, ale málo přesná
- 2. varianta
 - využití měření těsnosti:
 - **Spojité proměnné** - chybějící hodnota je nahrazena váženým průměrem pozorování x_k , jejichž hodnoty byly vyplněné → jako váha je použito měření těsnosti $prox(x_i, x_k)$.
 - **Kategoriální proměnné** - chybějící hodnota je nahrazena nejvíce frekventovanou hodnotou, která je opět vážená těsností.
 - Nahrazené hodnoty jsou použity v další iteraci lesa a jsou spočítány nové těsnosti. Tento proces se zastaví, pokud již nedochází k žádnému zlepšení, nebo např. po pěti iteracích (počet iterací je rovněž možné zvolit).
 - Tato varianta je poměrně přesná a vhodná pro datové soubory, které obsahují velké množství chybějících hodnot, nicméně může značně zvýšit výpočetní čas, protože jde o iterativní proces.

Literatura

- Breiman L. (1996) Bagging predictors. Machine Learning 24, pp.123-140.
- Breiman L. (2001) Random forests. Machine Learning 45, pp. 5-32.
- Breiman L. (July 2002), LOOKING INSIDE THE BLACK BOX, Wald II
- Using Random Forest to Learn Imbalanced Data, Chao Chen, Andy Liaw & Leo Breiman, July 2004
- The Elements of. Statistical Learning: Data Mining, Inference, and Prediction. Second Edition. February 2009. T.Hastie, R.Tibshirani, J. Friedman
- Yoav Freund & Robert E. Schapire (1997); A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting, Journal of Computer and System Sciences, 55(1):119-139
- Robert E. Schapire & Yoram Singer (1999); Improved Boosting Algorithms Using Confidence-Rated Predictors, Machine Learning, 37(3):297-336