FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

ODHAD PŘESNOSTI MODELU

Autor textu:
Ing. Petr Honzík, Ph.D.

Květen 2014

Komplexní inovace studijních programů a zvyšování kvality výuky na FEKT VUT v Brně
OP VK CZ.1.07/2.2.00/28.0193

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Obsah přednášky

Jaká asi bude chyba modelu na nových datech?

- Chyba modelu
- Bootstrap
- Cross Validation
- Vapnik-Chervonenkisova dimenze

Chyba skutečná a trénovací

- Máme 30 záznamů, rozhodli jsme se na jejich základě navrhnout rozhodovací strom. Jaká bude jeho skutečná chyba na nových datech?
- Z dostupných dat strom vytvoříme a na těchto datech zjistíme, s jakou chybou strom klasifikuje – tímto postupem získáme tzv. **trénovací chybu** Err_{train}. V tabulce jest Err_{train} ze 3 různých datových souborů.

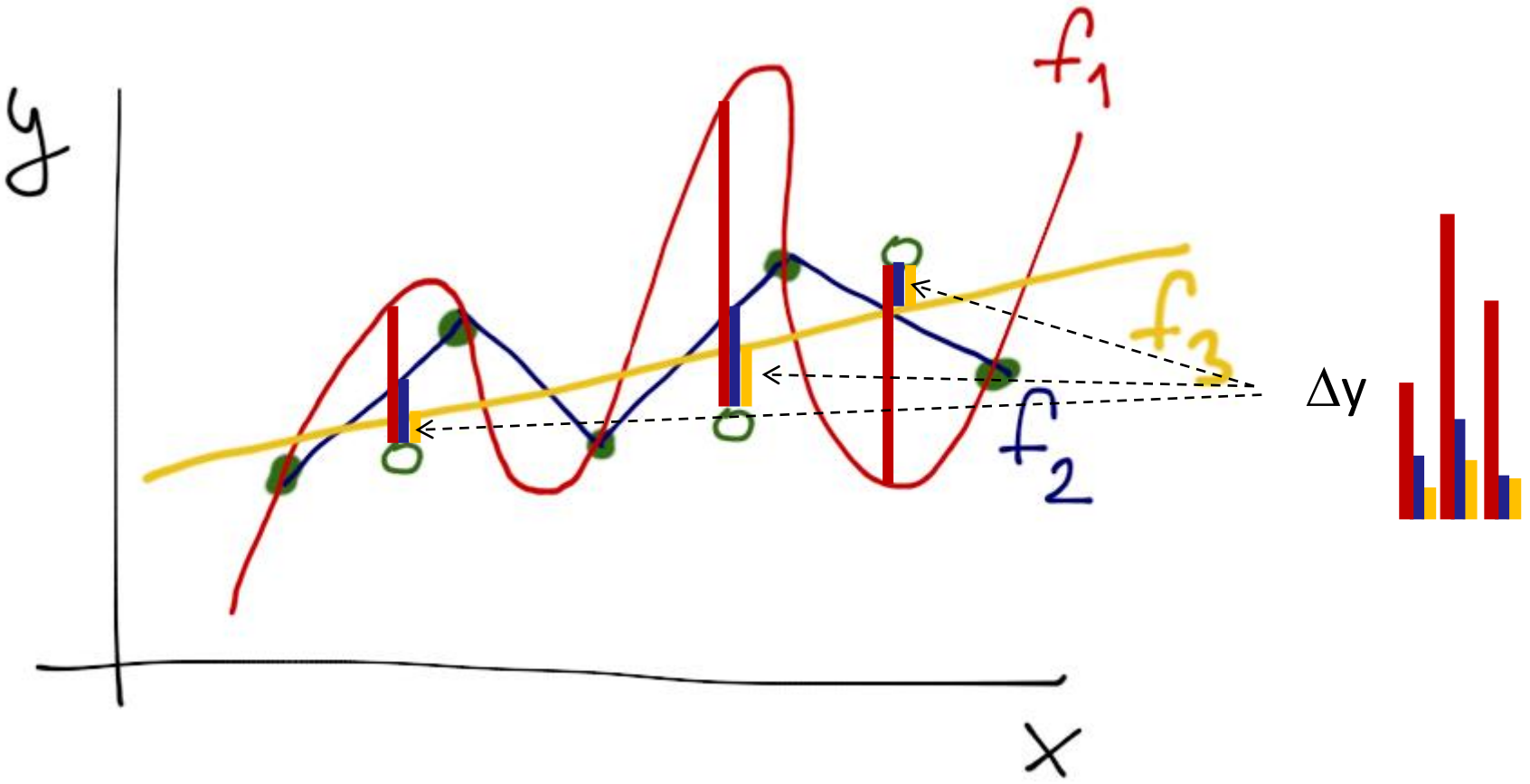
Err_{train}	0,210	0,100	0,300

- Bude Err_{train} odpovídat Err_{REAL}, tedy **chybě skutečné**? Ano? Ne? Bude větší? Menší? Proč?

Kde je problém?

- Mějme množinu dat *D*, každý záznam definován dvojicí $(\boldsymbol{x}, y)$
- Hledám funkční závislost *f* takovou, že $f(\boldsymbol{x})=y$
- Můžeme vytvořit neomezeně mnoho modelů. Jaká funkce (model) *f* je optimální?
- Rozdělme data na trénovací $(\boldsymbol{x}_R, y_R)$ a testovací $(\boldsymbol{x}_S, y_S)$
- Pak pro každou *f* existuje f^* takové, že $f(\boldsymbol{x}_R, y_R)=f^*(\boldsymbol{x}_R, y_R)$ a přitom $f(\boldsymbol{x}_S, y_S) \neq f^*(\boldsymbol{x}_S, y_S)$
- Uvažme, že máme pouze trénovací data. Jaká funkce je potom správná?

Příklad



Odhad chyby modelu z trénovacích dat

- Jak **rozdělit data**, aby byla chyba modelu co nejmenší (vhodný výběr trénovacích dat)?
- Jak potom **odhadnout skutečnou chybu** modelu z trénovacích dat?

- Obecný princip: opakované rozdělení použitých dat na trénovací a testovací data a určení průměrné chyby
  - máme málo dat
  - máme dostatek dat
  - známe kapacitu modelu (specifické)

Nejjednodušší způsob odhadu chyby

- Proto **rozdělíme data** (metoda hold out)
  - dělení *trénovací : testovací* např. *2/3 : 1/3*
  - dělení *trénovací : validační : testovací* přibližně *½ : ¼ : ¼*
- **Nevýhody** takového postupu
  - máme-li málo dat, není možné realizovat
  - existuje riziko, že rozdělení vytvoří neodpovídající rozložení (např. outliers pouze v trénovacích nebo pouze v testovacích datech) => velký rozptyl odhadnuté chyby v závislosti na rozdělení dat
  - s rostoucím počtem dat roste šance na přesnější model
- **Resubstituční chyba** – chyba zjištěná na datech použitých na trénování – vede k podhodnocení skutečné chyby

Vysvětli pojem resubstituční chyba?

Metoda bootstrap – když je dat málo…

- Bootstrap je **postup**, jak
  - rozdělit data na trénovací a testovací
  - získat odhad skutečné chyby modelu vytvořeného ze všech dat
- základní princip spočívá v tom, že je vygenerován velký počet trénovacích souborů B_i o četnosti N prvků **výběrem s opakováním** ze základního souboru všech N dostupných dat
- doporučovaný počet těchto B_i souborů je 50 až 2000, může jich být však i řádově více
- soubory B_i budou opakovaně použity jako trénovací
- ve statistice slouží bootstrap k robustnímu určení intervalů spolehlivosti základních charakteristik vzorku (průměr, rozptyl, medián, korelační koeficient atd.)
- při testování modelů je typické jeho použití v případech, kdy máme **nedostatek dostupných dat** (např. N<30)

Bootstrap - výpočet

- datový soubor o *N* záznamech, B_i výběrů s opakováním
- tradiční **bootstrap**:

$$\hat{Err}_{boot} = \frac{1}{N}\frac{1}{|B|}\sum_{j=1}^{|B|}\sum_{i=1}^{N} LF\left(y_i, \tilde{f}^{B_j}(x_i)\right)$$

kde $\tilde{f}^{B_j}$ je model naučený na B_j-tý výběr, testuje se na původním souboru

- existuje přesnější varianta, tzv. **0,632 bootstrap**
  - prvky, které nebyly vybrány, budou použity jako testovací data
  - pravděpodobnost nevybrání jednoho vzorku je $(1-1/N)^N \approx e^{-1}=0{,}368$

$$\hat{Err}_{Btest} = \frac{1}{|B|}\sum_{j=1}^{|B|}\frac{1}{|C_j|}\sum_{i=1}^{|C_j|} LF\left(y_i, \tilde{f}^{B_j}(x_{C_{ji}})\right)$$

kde C_i je množina všech prvků neobsažených ve výběru B_i

  - odhad celkové chyby modelu metodou 0,632 bootstrap je:

$$Err = 0{,}632 \,.\, Err_{Btest} + 0{,}368 \,.\, Err_{boot}$$

Příklady - pokračování

- Předpokládáme tedy (teoreticky) následující pořadí ve velikosti odhadu chyby:

$$Err_{Boots} < Err_{632} < \boldsymbol{Err_{REAL}} < Err_{Btest}$$

- V našich příkladech jsme získali následující výsledky:

	Err_{Boots} <	Err_{632} <	Err_{Btest}
gen1	0,237	0,284	0,311
gen2	0,112	0,156	0,174
Titanic	0,349	0,408	0,442

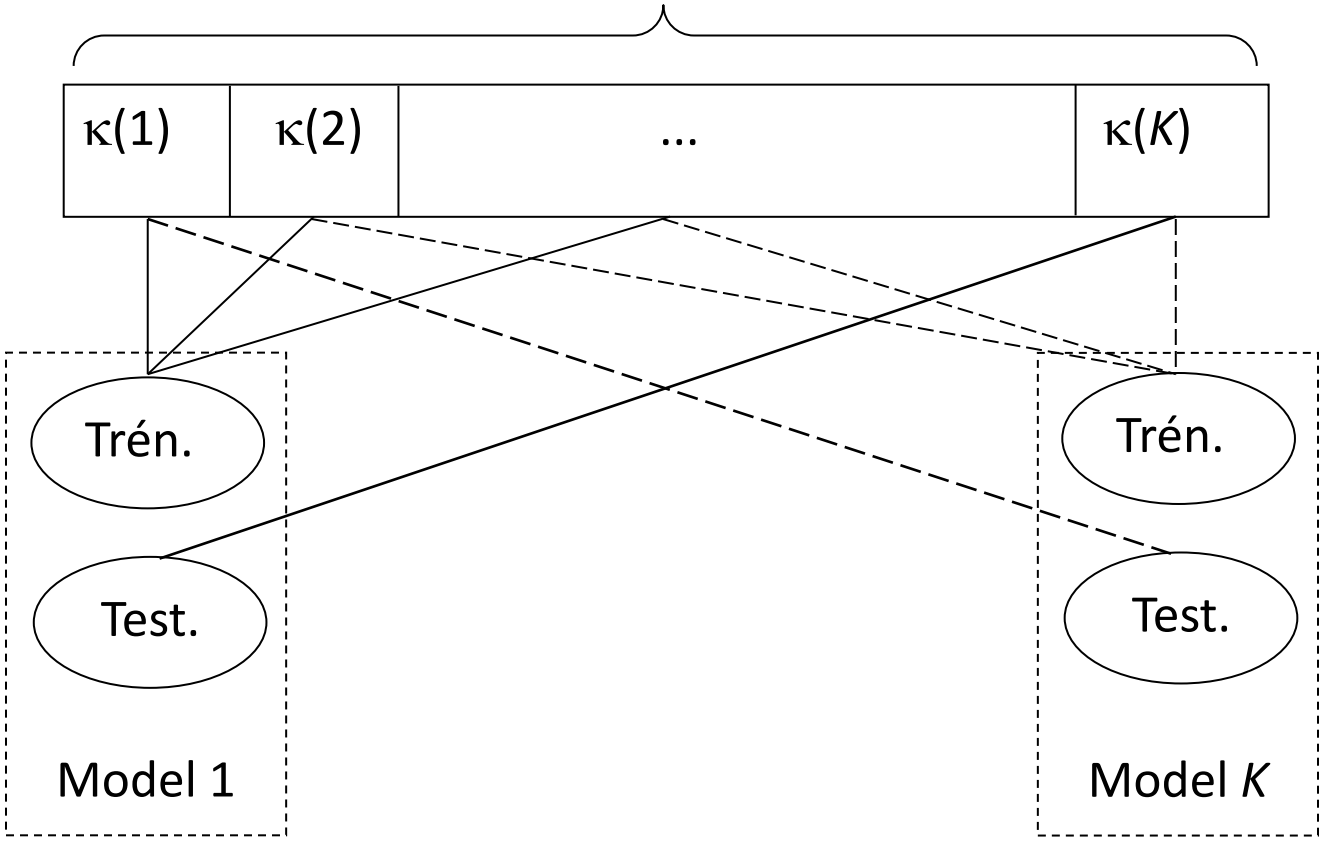
Zkuste zpřesnit odhad **skutečné chyby** pro jednotlivé datové soubory

Cross Validation - princip

- CV je metoda sloužící k odhadu skutečné chyby modelu, tedy k posouzení hypotézy, do jaké míry data odpovídají danému modelu.
- Princip spočívá v tom, že je datový soubor rozdělen na určitý počet pokud možno stejně velkých disjunktních množin *K*. Na základě tohoto dělení je *K*-krát nastaven a vyhodnocen model tak, že je postupně vždy jedna množina použita jako testovací a sjednocení ostatních množin jako trénovací soubor dat. Je tak získáno *K* různě nastavených modelů.
- Součet všech vypočtených odchylek slouží k určení skutečné chyby modelu vytvořeného na základě použitých dat.

Cross Validation - schéma

Datový soubor rozdělený na *K* podmnožin



Cross Validation – typické nastavení *K*=10

- typické dělení je ***K*=10,** tzv. **Tenfold Cross Validation** nebo *K*=5, zdůvodnění tohoto nastavení je experimentální zkušenost…
- Celková chyba modelu metodou Err_{CV} je dána průměrem chyby Err všech dílčích modelů:

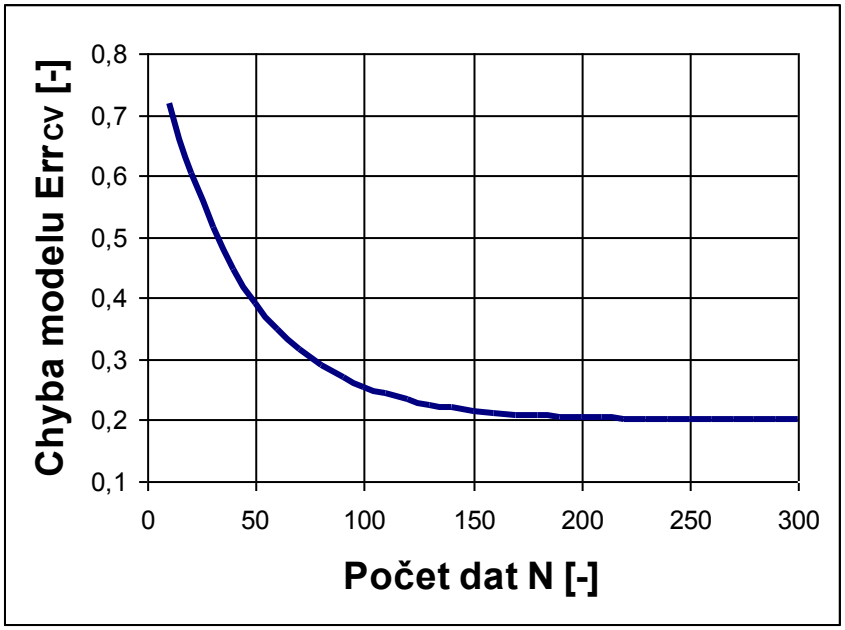
$$Err_{CV} = \frac{1}{K}\sum_{i=1}^{K} Err\left(y_{\kappa(i)}, \tilde{f}^{-\kappa(i)}\left(x_{\kappa(i)}\right)\right)$$

kde *K* je počet podmnožin vytvořených z úplného datového souboru , $\kappa(i)$ je *i*-tá podmnožina, $y_{\kappa(i)}$ a $x_{\kappa(i)}$ jsou výstupní a vstupní data obsažená v podmnožině $\kappa(i)$ a $\tilde{f}^{-\kappa(i)}$ je model nastavený bez použití podmnožiny $\kappa(i)$.

? vysvětlete princip Cross Validation

Cross Validation – přecenění chyby

- graf znázorňuje hypotetickou křivku chyby modelu v závislosti na počtu použitých trénovacích dat (velikost trénovací množiny na ose *x* jen hypotetická – ve skutečnosti je individuální pro každý případ)
- důsledkem je přecenění chyby metodou CV při menším počtu dat, což lze řešit zvětšením *K,* tedy počtu skupin, na které data rozdělíme



? co je to přecenění chyby a proč k němu dochází u metody CV

Cross Validation – Leave-one-out (K=N)

- metoda Cross Validation s dělením odpovídajícím počtu samotných prvků
- naučíme model na N-1 prvků a na posledním ověříme správnost klasifikace; to zopakujeme N-krát pro všechny prvky
- nejpřesnější odhad chyby modelu
- chyba však vykazuje velký rozptyl
- časově velice náročné

$$Err_{CV} = \frac{1}{N}\sum_{i=1}^{N} L\left(y_{\kappa(i)}, \tilde{f}^{-\kappa(i)}\left(x_{\kappa(i)}\right)\right)$$

? co je to Leave-one-out

Vliv počtu dělení CV na odhad přesnosti 1/2

- Mějme model se známou přesností 80% na 100 záznamech
- Rozdělme data na 5, 10 a 100 foldů (prům. přesnost vždy 80%)
- Podívejme se na vývoj směrodatné odchylky

5 fold CV

průměr	80,0	80,0
směr. odch.	40,0	4,0
fold 1	0	78
fold 2	100	85
fold 3	100	84
fold 4	100	74
fold 5	100	79

10 fold CV

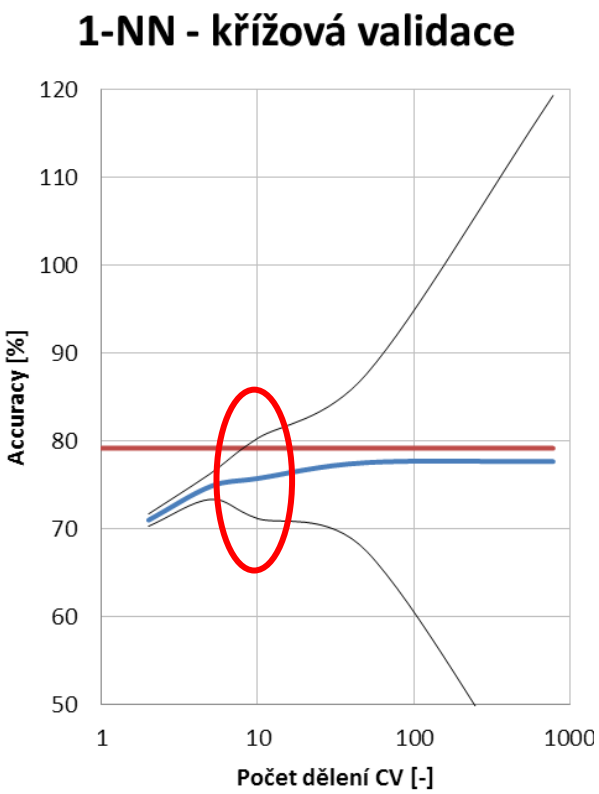
průměr	80,0	80,0
směr. odch.	40,0	5,4
fold 1	0	85
fold 2	0	88
fold 3	100	71
fold 4	100	76
fold 5	100	79
fold 6	100	78
fold 7	100	82
fold 8	100	81
fold 9	100	73
fold 10	100	87

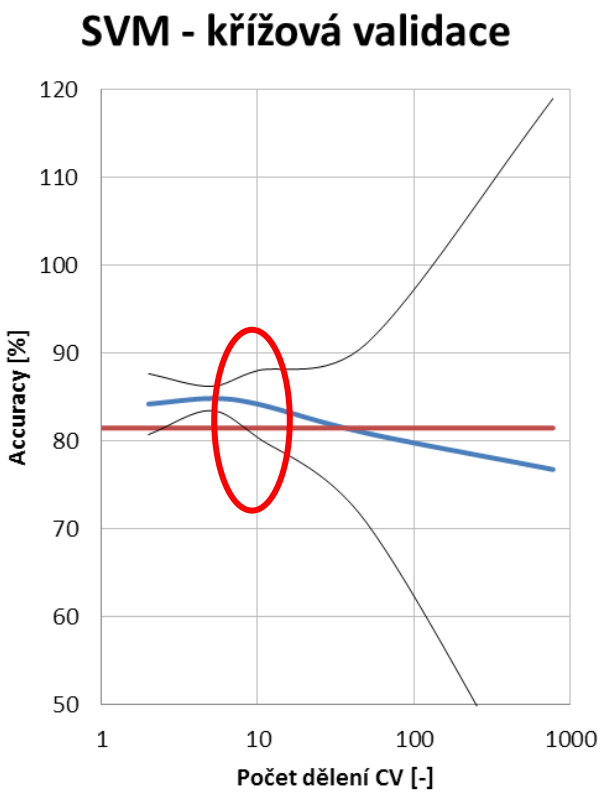
100 fold CV

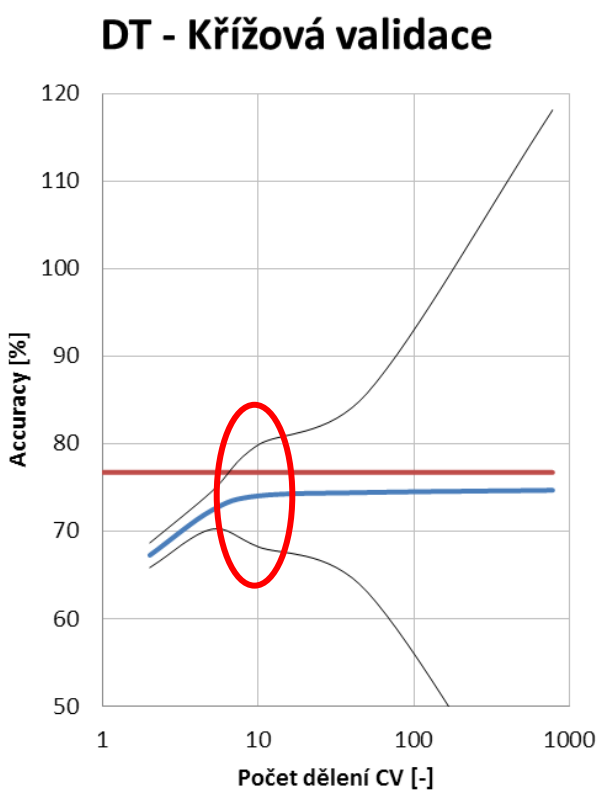
průměr	80,0
směr. odch.	40,0
fold 1	0
fold 2	0
fold 3	0
fold 4	0
fold 5	0
fold 6	0
...	...
fold 95	100
fold 96	100
fold 97	100
fold 98	100
fold 99	100
fold 100	100

Vliv počtu dělení CV na odhad přesnosti 2/2

- experiment na jednom datovém souboru
- 779 trénovacích záznamů, 26 tříd
- **skutečná přesnost**, **odhad pomocí CV**, směrodatná odchylka







Cross Validation – použití, vlastnosti

- použití
  - srovnání více přístupů, výběr nejlepšího (různé typy modelů, různá nastavení jednoho typu modelu)
  - stanovení předpokládané přesnosti modelu (průměrem parametrů jednotlivých modelů, použití stejné metodiky pro všechna dostupná data)
- vlastnosti
  - výhody: vyšší přesnost, kompromis *k-fold* má výhody přesnosti (one-leave-out) a zároveň rozumné výpočetní náročnosti
  - nadhodnocení chyby, u one-leave-out velký rozptyl odhadu, časová náročnost

Příklady - pokračování

- Předpokládáme tedy (teoreticky) následující pořadí ve velikosti odhadu chyby:

$$Err_{REAL} < Err_{CV}$$

- V našich příkladech jsme získali následující výsledky:

	Err_{CV}
gen1	0,312
gen2	0,133
Titanic	0,433

Zkuste zpřesnit odhad **skutečné chyby** pro jednotlivé datové soubory

Základní srovnání CV a Bootsrap

	Cross Validation	Bootstrap
Pozitivní	• nepřekrývání testovacích dat • jednoduché • K=10 – snížená výpočetní náročnost	• lze použít při malém počtu dat • statisticky zajímavé pro intervalové odhady charak-teristik datových souborů
Negativní	• nejednoznačné stanovení velikosti K • požaduje více dat	• chyba překrytím trénovacích a testovacích dat (resubstituční chyba) • výpočetně náročné

? Která metoda (CV,Bootstrap) chybu přeceňuje a která podceňuje, proč?

Očekávaní a experimenty: CV vs. Bootstrap

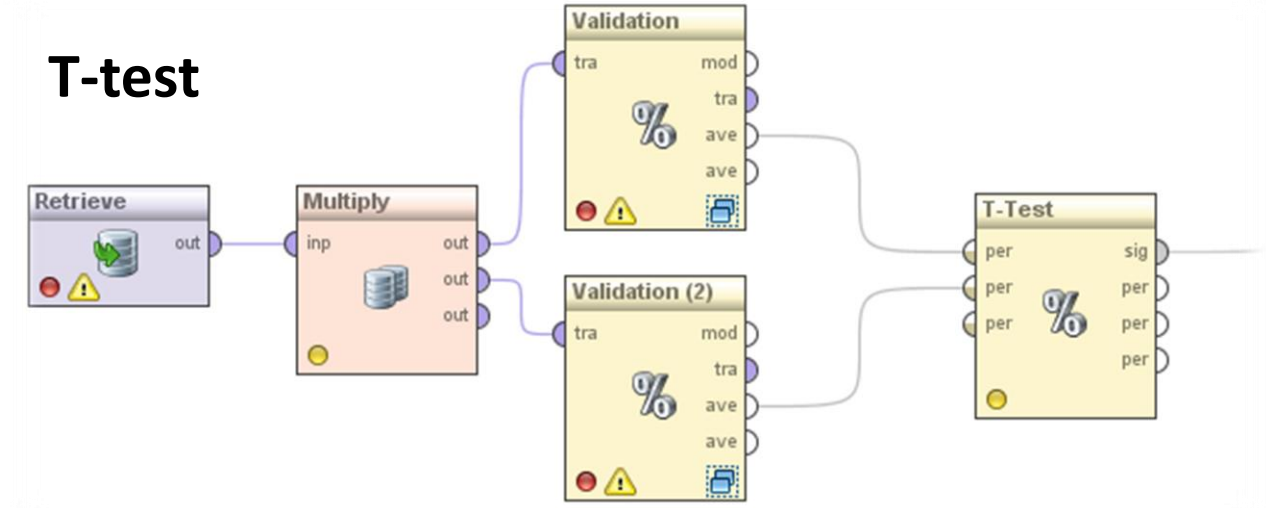
	$Err_{train} < Err_{Boots} < Err_{632} < \mathbf{Err_{REAL}} < Err_{CV} < Err_{Btest}$					
gen1	0,210	0,237	0,284	**0,287**	0,312	0,311
gen2	0,100	0,112	0,156	**0,343**	0,133	0,174
Titanic	0,300	0,349	0,408	**0,302**	0,433	0,442

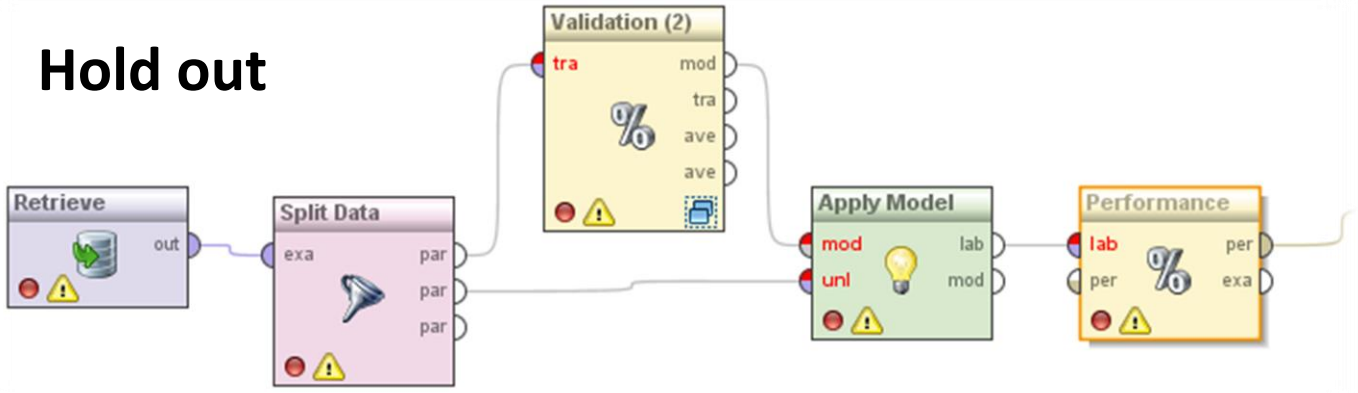
Z uvedených experimentů je zřejmé, že k odhadům chyby je třeba přistupovat obezřetně, jejich **přesnost je velice citlivá** na kvalitu výběru (z chybných dat korektním postupem dobrý model nikdy nezískáte!). V příkladech je pracováno s výběrem 30 záznamů, takže zkreslení je velké. Použití CV není s tímto výběrem smysluplné (použito pro ukázku růstu odhadu chyby).

Cross Validation – výběr modelu

- Typické použití – při stejném dělení do foldů model s největší průměrnou přesností
- Možné testování H0: chyba obou modelů je stejná (tedy rozdíl průměrů je roven nule)
  - Možné použít párový t-test, při větším počtu dělení (>30) nemusí mít normální rozdělení
  - Data by měla mít stejný rozptyl (F-test)
  - Ideální použití alternativního neparametrického testu –**Two-sample Wilcoxon Signed-Rank Test**
- Riziko „přeučení CV“
  - mějme 100 záznamů, 1000 příznaků
  - opakovaným hledáním pomocí CV lze najít modely s nulovou chybou – takové modely však lze vytvořit i na datech s náhodnou klasifikací
- Přeučení se lze bránit buď
  - snížením kapacity modelu nebo regularizací
  - Použitím „hold out“ metody, nechat si pro selekci bokem data, která se neúčastní CV

Cross Validation v Rapid Mineru





Vapnik-Chervonenkisova dimenze - princip

- hlavní myšlenka spočívá v tom, že každá funkce má svoji **kapacitu *h***, jejíž hodnota odpovídá počtu prvků, které je funkce schopna rozlišit (*h* je závislá na modelu, ne na datech)
- tato metoda umožňuje určení testovací chyby z chyby trénovací a kapacity modelu *h*
- mějme *N* prvků binárně klasifikovaných; pak existuje právě 2^N kombinací jejich klasifikací
- funkce má kapacitu *N*, pokud existuje taková množina prvků, které lze rozlišit ve všech možných 2^N kombinacích (přímka, RS)
- existuje více algoritmů pro vyjádření kapacity funkce, její určení je zásadní problém

? co je to kapacita funkce

Vapnik - Chervonenkisova dimenze – odhad chyby modelu

- trénovací chyba Err_{train}

$$Err_{train} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left|f(\boldsymbol{x}_i) - y_i\right|$$

- hodnota skutečné chyby Err_{REAL}:

$$Err_{REAL} \leq Err_{train} + \Phi(h, N, \delta)$$

- výpočet intervalu spolehlivosti Φ:

$$\Phi(h, N, \delta) = \sqrt{\frac{1}{N}\left(h\left(\ln\frac{2N}{h}+1\right)+\ln\frac{4}{\delta}\right)}$$

kde h je kapacita funkce, N je počet prvků a δ je pravděpodobnost, že chyba bude větší než ve výše uvedené nerovnici