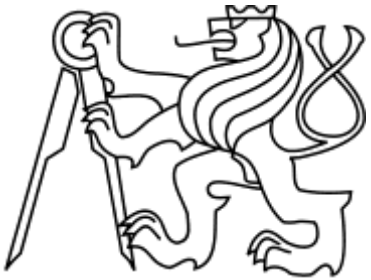
Testování modelů a jejich výsledků

Jak moc můžeme věřit tomu, co jsme se naučili?



Nature Inspired Technologies Group

Osnova

- Úvod
- Trénovací, Testovací a Validační datové soubory
  - Práce s nebalancovanými daty; ladění parametrů
- Křížová validace (Cross-validation)
- Porovnání různých schémat pro dobývání znalostí

Úvod

- Jak dobře předpovídá (klasifikaci) model, který jsme vytvořili?
- Chyba, s jakou model klasifikuje na trénovacích datech *není* dobrým odhadem pro chování modelu na dosud neznámých datech
  - ***Q: Proč?***
  - A: Nová data **nebudou přesně stejná** jako ta použitá pro učení!
- **Přeučení** (overfitting) – na trénovacích datech můžeme vytvořit model s libovolně malou chybou! Ale testování takového modelu obvykle dává špatné výsledky!

Jak se hodnotí vzniklý model?

- Míry, které se obvykle k hodnocení modelů používají:
  - **Klasifikační přesnost** (Classification Accuracy)
  - **Celková cena chyby** – v případě, že cena různých typů chyb je různá (použití „drahé" vyšetřovací metody)
  - **Zdvih** (lift): *kolikrát se zvýší spolehlivost oproti „průměru*"
  - **ROC křivky**
  - Chyba u predikce numerických hodnot (při regresi) – např. součet čtverců (nebo abs. hodnot) odchylek od skutečné hodnoty
- Jak moc se můžeme spolehnout na výsledky predikované modelem ?

Relativní chyba klasifikátoru

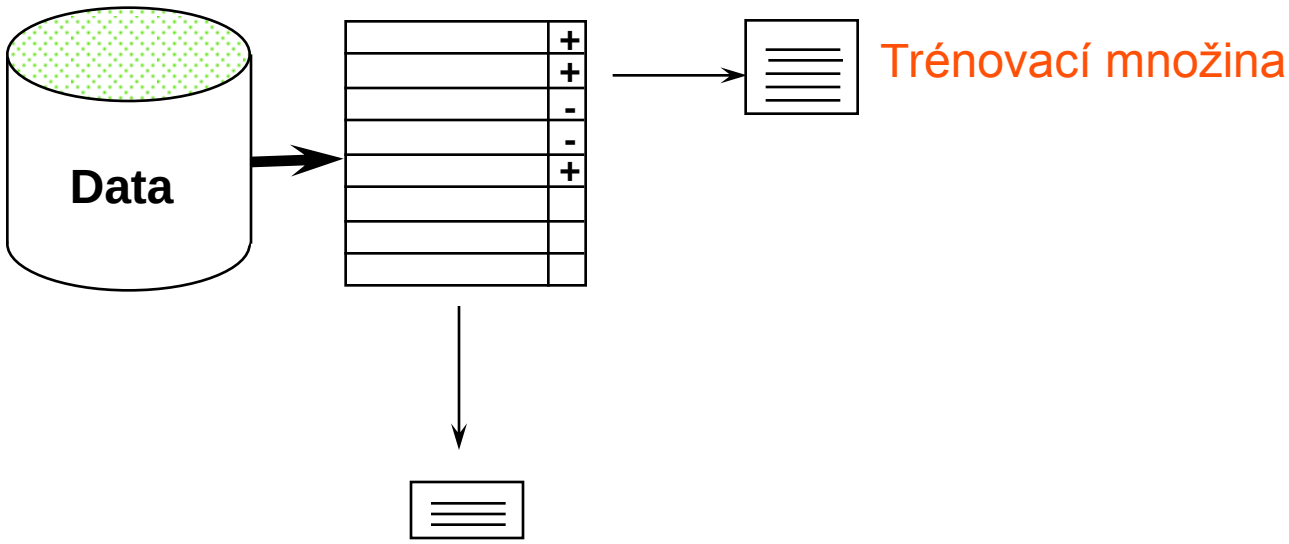
- Velmi přirozenou mírou je relativní chyba (*error rate*) vypočtená přes všechny uvažované instance :
  - *Úspěch (success)* : model pro danou instanci model určí správnou třídu
  - *Chyba* : model pro instanci určí třídu špatně
  - **Relativní chyba**: procentuální podíl chybných instancí vůči mohutnosti celé uvažované množiny instancí
- *Chyba na trénovacích datech* je příliš optimistický odhad!
  - I náhodně vygenerovaný konečný soubor dat lze totiž popsat nějakým modelem (třeba samotnou výchozí tabulkou)

Hodnocení pro "ROZSÁHLÁ" data

- Máme-li hodně dat (tisíce instancí), které obsahují pro každou třídu dostatek vzorků (stovky instancí), pak stačí provést jednoduché testování:
  - Rozděl výchozí data náhodně do 2 množin: **trénovací** (asi 2/3 dat) a **testovací** (zbytek, tedy asi 1/3 dat)
  - Vytvoř klasifikační model nad *trénovací množinou* a proveď hodnocení (např. pomocí relativní chyby) na *testovací množině*.

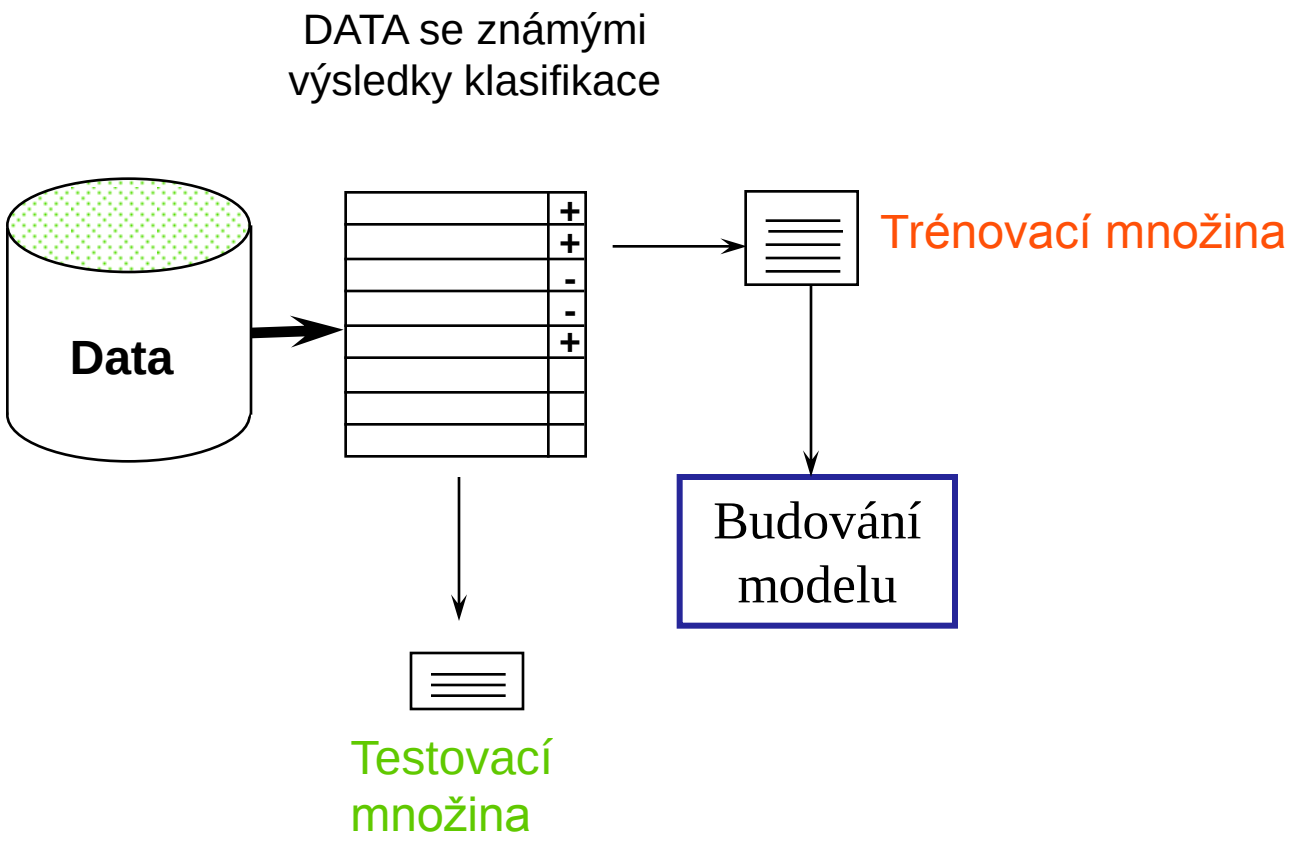
Klasifikace - krok 1: Rozděl data na trénovací a testovací množinu

DATA se známými výsledky klasifikace



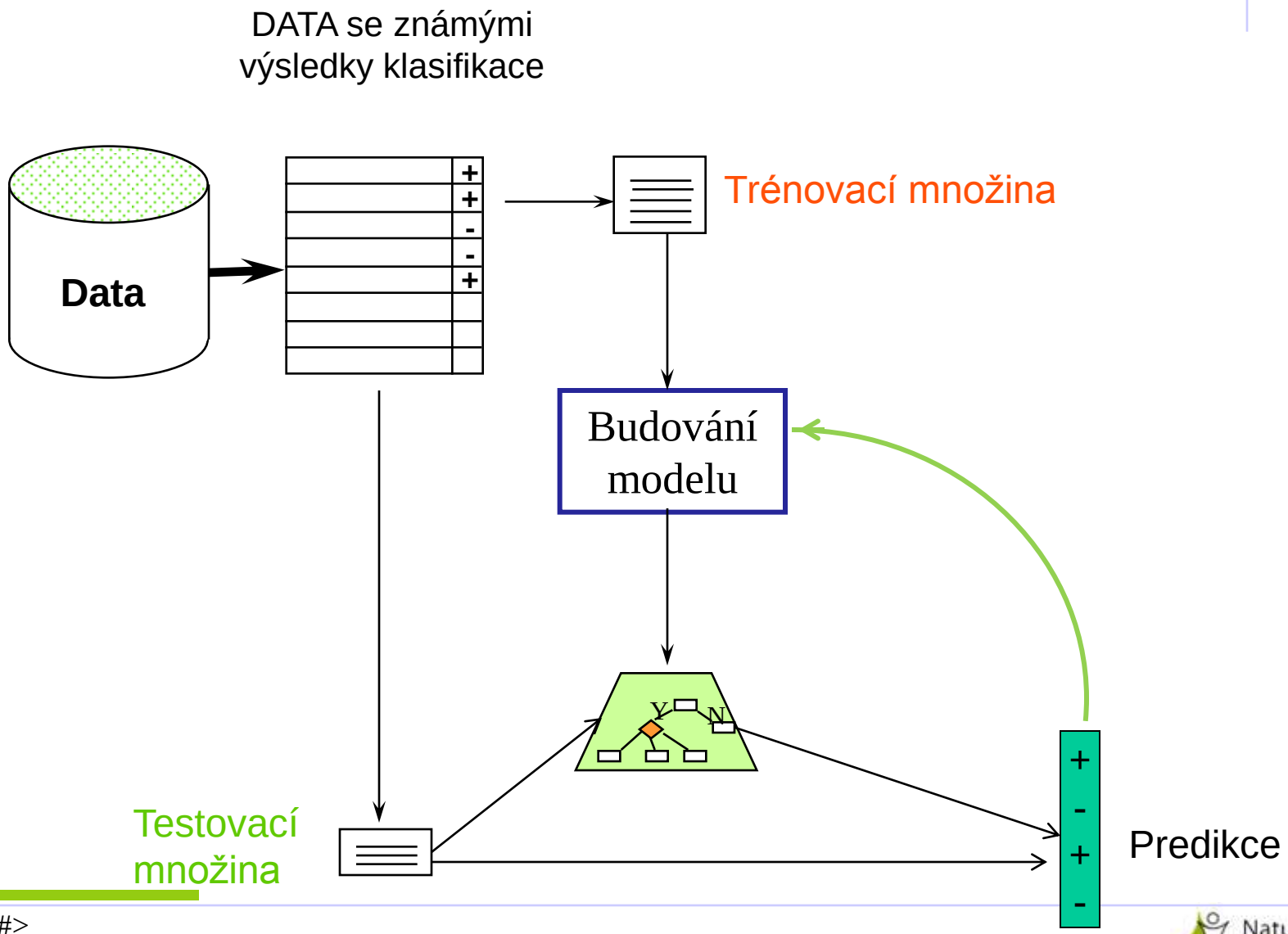
Nature Inspired Technologies Group

Klasifikace - krok 2: Vytvoř model na trénovacích datech



Nature Inspired Technologies Group

Klasifikace - krok 3: Otestuj model na test. datech (a případně zkus vytvořit jiný)



Nature Inspired Technologies Group

Co s nerovnoměrným zastoupením tříd (*unbalanced data*)?

- Často máme data s nerovnoměrným zastoupením tříd
  - Únik zákazníků: 97% zůstane, 3% odcházejí (za měsíc)
  - Lék.diagnóza: 90% zdravých, 10% nemocných
  - eCommerce: 99% nekoupí, 1% koupí
  - Bezpečnost: >99.99% cestujících nejsou teroristé
- Obdobně v případě klasifikace do více tříd.
- Model klasifikující do **majoritní třídy** bude dávat nízkou relativní chybu. Ale **není vůbec** užitečný!

Vyvážení nevyvážených dat

- Klasifikace do 2 tříd: vytvoř **vyvážené** (*BALANCED*) **soubory dat na trénování** (vytvoření modelu) i na **testování**.
  - Vyber náhodně potřebný počet instancí klasifikovaných do minoritní třídy
  - a doplň je stejným množstvím náhodně vybraných instancí z majoritní třídy
- Zobecnění postupu "vyvážení" pro více tříd
  - Je nutné zajistit, aby v trénovací i testovací množině byl **počet instancí** pro každou třídu **zhruba vyrovnaný**

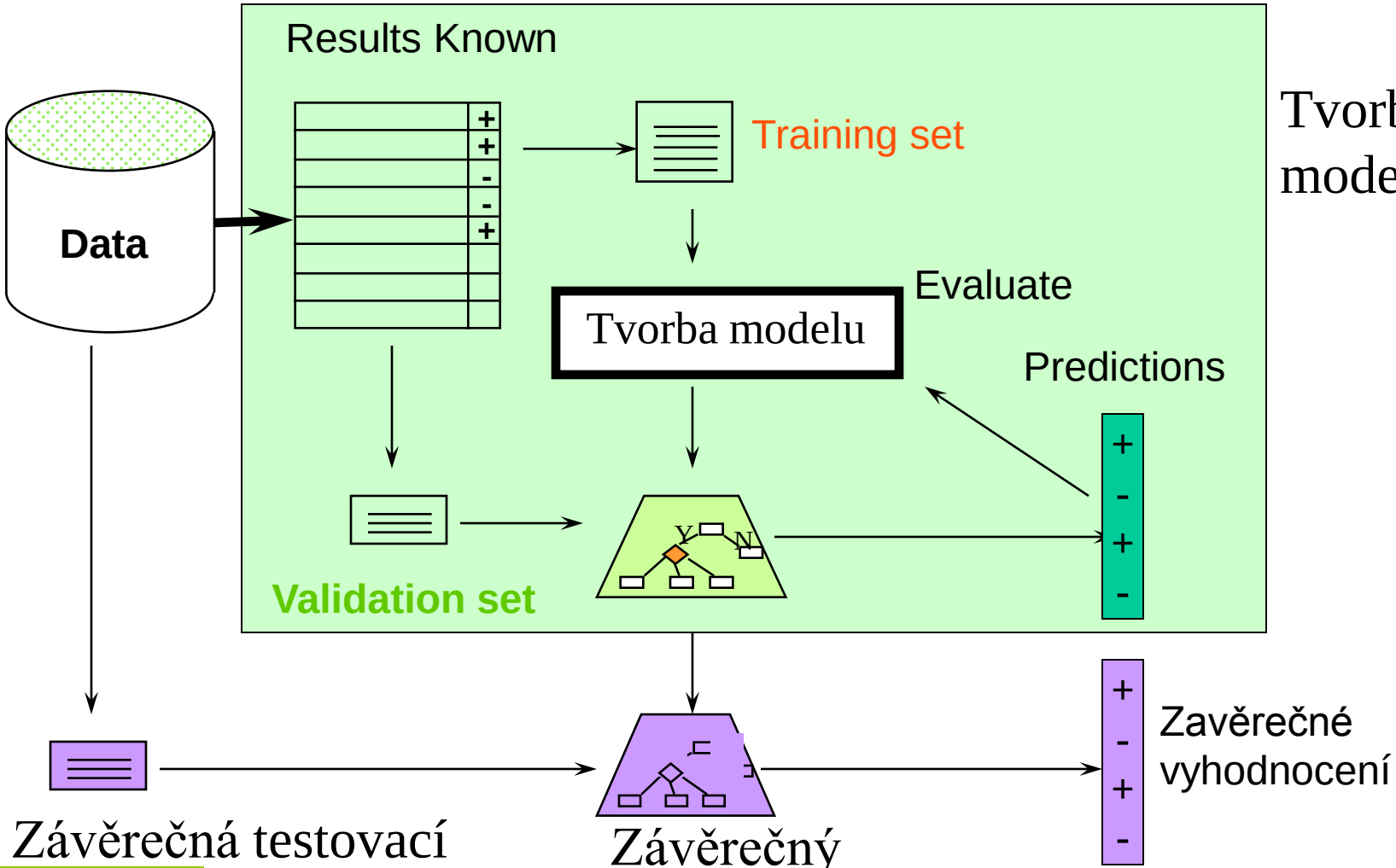
Poznámka k ladění parametrů

- Někdy učení modelu postupuje ve 2 krocích:
  - **Krok 1**: navrhne základní strukturu (např. rozhodovací strom)
  - **krok 2**: optimalizuje parametry zvolené struktury (který pak prořeže)
- Testovací model musí vzniknout tak, že *nijak nejsou použita* trénovací data! A to ani pro ladění parametrů!
- V tomto případě by korektní procedura měla používat 3 nezávislé množiny dat: **data trénovací, validační a testovací**
  - Validační data slouží k optimalizaci parametrů (např. prořezání stromu)

Jak co nejlépe využít dostupná data ?

- Po ukončení evaluace je možné použít *VŠECHNA DATA* pro budování výsledného klasifikátoru
- **Obecně:**
  - Čím větší je trénovací množina, tím je lepší klasifikátor (úměra však není lineární)
  - Čím větší je testovací množina, tím kvalitnější je odhad průměrné chyby.

Klasifikace: dělení na množiny trénovací, validační a testovací



Odhad budoucího výkonu modelu

- Nechť je odhad relativní chyby (na testovacích datech) roven 25%. Jak to bude při reálném testování?
  - Záleží na množství testovacích dat.
- Predikce na test.datech je podobná házení „*cinknutou*“ mincí (!).
  - "**Hlava**" je "*shoda mezi třídou skutečnou a tou, která je předpovězena modelem*", "**znak**" je "*neshoda …*"
- Statistika nazývá takovou posloupnost nezávislých jevů ***Bernoulliho proces,*** pro který statistická teorie nabízí **konfidenční intervaly**, které odhadnou odpovídající skutečnou hodnotu chyby (= hodnocení ***p***, jak je mince *cinknutá*) !

Konfidenční intervaly

- **Význam**: Úspěšnost klasifikace ***p*** leží uvnitř nějakého specifického intervalu s určitou mírou důvěry.
- **Příklad 1**: *S*=750 správně klasifikovaných příkladů pro *N*=1000 instancí
  - Odhad relativní úspěšnosti klasifikace: 75%
  - Jak spolehlivý je tento odhad relativní úspěšnosti klasifikace ***p***?
    - Odpověď: S pravděpodobností 80% je ***p*** $\in$[73.2,76.7]
- **Příklad 2**: *S*=75 a *N*=100
  - Odhad relativní úspěšnosti klasifikace: 75%
  - S pravděpodobností 80% je ***p*** $\in$[69.1,80.1]

Řešení pro "malé soubory" dat

- Metoda zádrže (*holdout*) si ponechá určitou část dat pro testování a zbytek použije na trénování
  - Obvykle: 1/3 na testování, zbytek tvoří trénovací množinu

- Výsledné trénovací a testovací množiny nemusí být dostatečně reprezentativní v případě malých či nevyvážených souboru data.
  - Např. máme-li jen málo (nebo žádné) instance některé třídy - to řeší vytvoření **vyváženého** (*ballanced)* **vzorku**

Vyhodnocení na "stratifikovaném vzorku"

- **Stratifikovaný náhodný vzorek** musí respektovat i zastoupení vrstev, které jsou relevantní pro studovanou úlohu. *Např.* při studiu tělocvičných aktivit je třeba zohlednit např. věk, pohlaví a sociální status. Náhodný vzorek pak musí vznikat v každé příslušné vrstvě zvlášť !

- ***Stratifikovaný vzorek**: pokročilá verze vyvažování dat*

Metoda opakované zádrže (repeated holdout)

- Odhad pomocí zádrže může být upřesněn tím, že se proces vícekrát opakuje s různými vzorky
- Metoda opakované zádrže:
  - V každé iteraci je určitá část dat náhodně vybrána jako trénovací (s využitím stratifikace, je-li třeba)
  - Relativní chyby všech iterací se zprůměrují - výsledek je **celková relativní chyba**
- Ale pozor: různé testovací množiny se mohou překrývat.

Lze se tomu vyhnout?

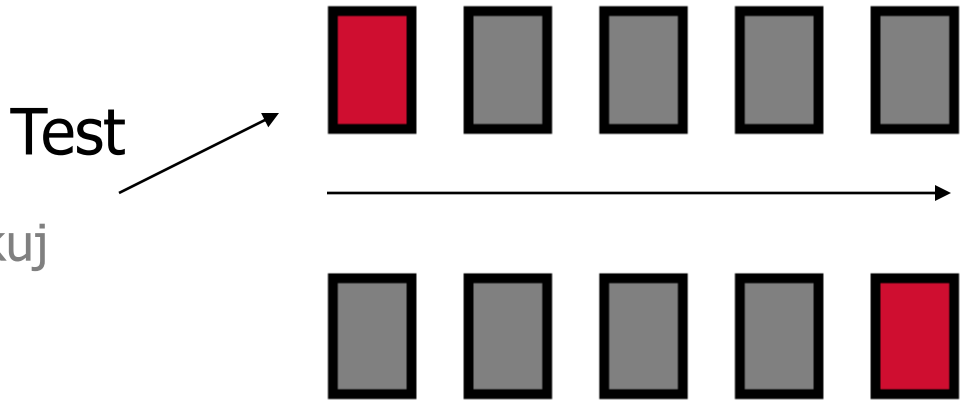
Křížová validace (cross-validation)

- *k-ární křížová validace* zamezuje překrývání testovacích množin
  - Krok 1: data jsou rozdělena do *k* disjunktních podmnožin stejné velikosti
  - Krok 2: Každá podmnožina je použita právě jednou pro testování modelu vzniklého ze zbylých dat
- Často se ještě předem jednotlivé podmnožiny stratifikují
- Odhad chyby zvoleného modelu se pak získá jako průměr chyb pro jednotlivé testovací množiny

Příklad na křížovou validaci:

- Rozděl data data do skupin (folds) stejné velikosti

- Zadrž jednu skupinu na testování a zbytek použij pro trénování



- Opakuj

Ještě o křížové validaci

- Standardní postup vyhodnocení**: stratifikovaná 10-násobná** (ten-fold) **křížová validace**
- Proč 10? Empirická zkušenost ověřená na řadě experimentů: odhady z této volby jsou velmi dobré!
- Stratifikace pak ještě zmenšuje variabilitu odhadu
- **Další vylepšení**: opakovaná stratifikovaná křížová validace
  - Např. opakuj 10x křížovou validaci se základem 10 (10-násobná KV) a zprůměruj výsledky

Křížová validace „vynech 1“

- **„vynech 1“**:
  (zvláštní případ křížové validace):
  - Nechť počet skupin = počet výchozích dat
  - T.j., pro *n* výchozích instancí, vytvoř *n* X klasifikátor (z trénovacích dat o rozsahu *n* -1)

- **Vlastnosti:**
  - Optimální využití dat (důležité pro malé soubory)
  - Nepoužívá náhodné vzorkování
  - **Nevýhody:**
    - **výpočetně náročné** (výjímkou jsou některé neuronové sítě)
    - **Stratifikace:** nelze žádným způsobem zajistit!

KV „Vynech 1“ a stratifikace

- **Velmi hrubý odhad chyby** - viz následující *extrémní příklad*: Mějme výchozí soubor dat se sudým počtem instancí rozdělení NÁHODNĚ do dvou stejně velkých tříd
  - Na každé trénovací skupině nejlepší model bude ten, který predikuje majoritní třídu
  - Úspěšnost tohoto modelu na úplně nových datech bude 50%
  - Ovšem výsledek KV „Vynech 1“ bude 100% chyba!

Metoda „bootstrapping“

- KV pracuje se vzorkováním *bez navracení (without replacement)*
  - Je-li určitá instance jednou vybrána do jedné skupiny, nemůže být vybrána podruhé do jiné
- *Bootstrapping* vytváří trénovací množinu postupem s *navracením* . Mějme na začáku *n* instancí dat

- Vytvoř **skupinu *n* instancí** tak, že budeš vybírat z výchozí množiny dat *n* krát *s navracením*
- Právě vybraná skupina se stane trénovací množinou.
- Data z původní množiny, která nejsou v trénovací množině, tvoří **testovací** množinu.

Jiný název: „0.632 bootstrap“

- Zdůvodnění pro výchozí množinu s *n* instancemi
  - **Konkrétní instance** má pravděpodobnost (1–1/*n*), že *nebude* vybrána do trénovací množiny
  - Tedy provděpodobnost, že tato konkrétní instance se dostane do testovací množiny je:

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} = 0.368$$

  - Z toho vyplývá, že trénovací data budou obsahovat asi 63.2% instancí

Odhad chyby a bootstrapping

- Odhad chyby z testovacích dat, kterých je jen 36,8 %, je velmi pesimistický!
- Proto se doporučuje tento odhad upřesnit tím, že se kombinuje s odhadem z trénování

$$err = 0.632 \cdot e_{\text{test instances}} + 0.368 \cdot e_{\text{training instances}}$$

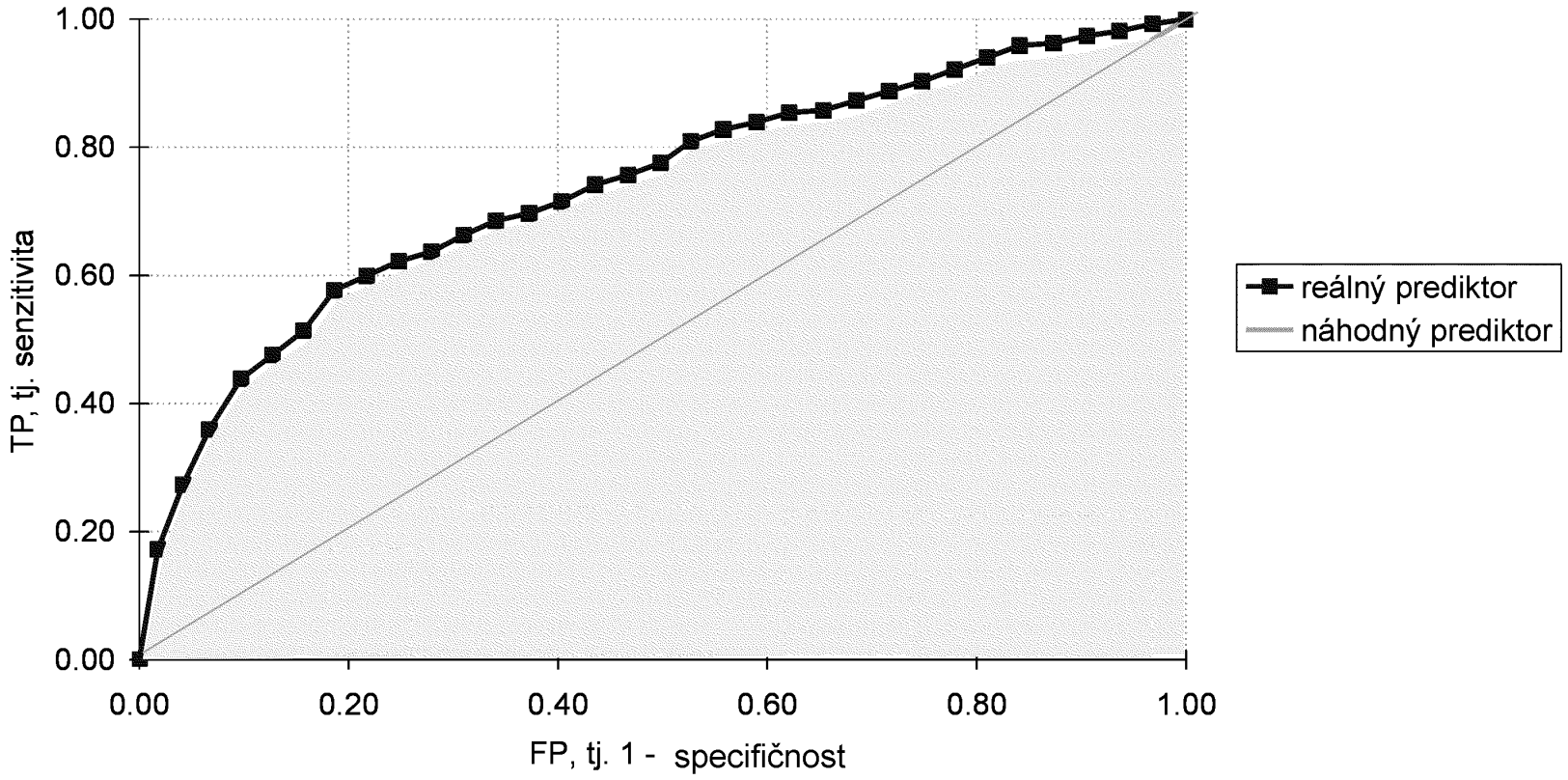
s tím, že resubstituční chyba (z trénovacích dat) má nižší váhu než chyba na testovacích datech!

- **Další upřesnění:**

Celý proces „bootstrapping" se několikrát opakuje a výsledky zprůměrní

Křivka ROC (Receiver Operating Char.)

	Předvídaný pozitivní	Předvídaný negativní	
Skutečně pozitivní	a	b	***TP*** $= a/(a+b)$ **senzitivita** $FN = b/(a+b)$
Skutečně negativní	c	d	***TN*** $= d/(c+d)$ **specifičnost** $FP = c/(c+d)$



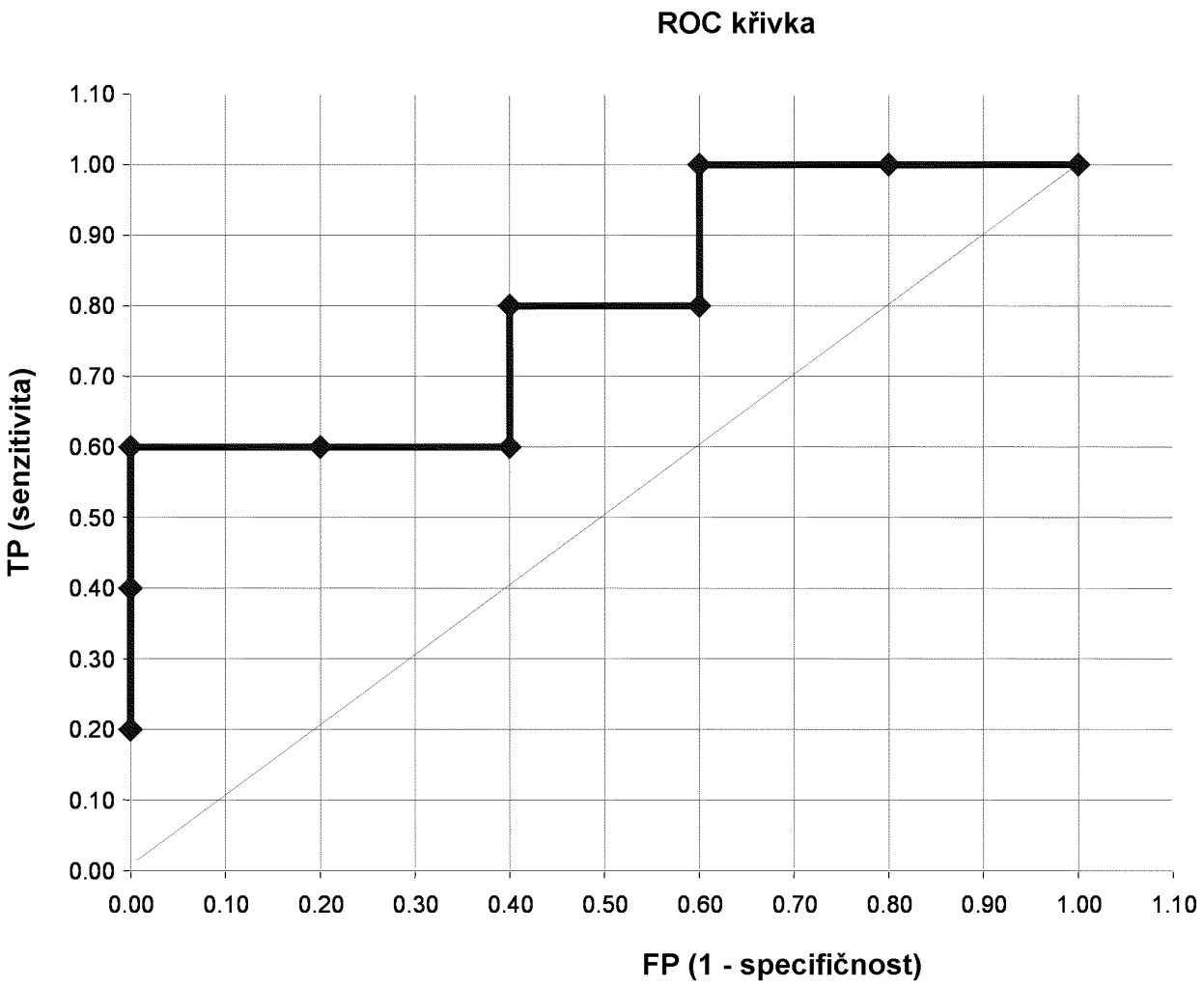
Křivka ROC (Receiver Operating Char.)

- Srovnání modelů neprovádíme v jediném bodě
- Vhodné u modelů, které přímo neklasifikují, ale odhadují pravděpodobnost příslušnosti k některé ze tříd.
-
- Např. mějme klasifikační algoritmus, který predikuje pravděpodobnost příslušnosti k pozitivní třídě $p(\boldsymbol{x}(i))$. O diskrétní klasifikaci rozhoduje parametru θ pro hodnotu prahu:
  - Je-li pro objekt i hodnota $p(\boldsymbol{x}(i)) > \theta$, zařadíme objekt popsaný vektorem nezávislých veličin $\boldsymbol{x}(i)$ mezi pozitivní příklady,
  - je-li menší, označíme jej jako negativní.

Příklad konstrukce ROC křivky

Predikce	0	0,05	0,2	0,22	0,3	0,31	0,4	0,42	0,7	0,8
Skutečná třída	0	0	1	0	1	0	0	1	1	1

Křivka ROC vznikne vypočtením hodnot *TP* a *FP* pro **všechny různé prahy** θ: 0,025; 0,125; ..; 0,75



* Studentův párový t-test

Princip vychází z následující úvahy pro výsledky $x_1\ x_2 \ldots x_k$ a $y_1\ y_2 \ldots y_k$ získané při testování dvou modelů pomocí k násobné KV

- Máme-li dostatek vzorků, pak by průměr měl mít normální distribuci
  - m_x a m_y jsou příslušné průměry
  - Odhad pro rozptyl průměrů je σ_x^2/k a σ_y^2/k

$$\frac{m_x - \mu}{\sqrt{\sigma_x^2 / k}}$$

- Kdyby μ_x a μ_y byly skutečné hodnoty průměrů nad oběma modely, mělo by jít o **normální rozdělení** s průměrem 0 a rozptylem 1

$$\frac{m_x - \mu_x}{\sqrt{\sigma_x^2 / k}} \qquad \frac{m_y - \mu_y}{\sqrt{\sigma_y^2 / k}}$$

William Gosset, Born:1876 in Canterbury; Died: 1937 in Beaconsfield, England

Obtained a post as a chemist in the Guinness brewery in Dublin in 1899. Invented the t-test to handle small samples for quality control in brewing. Wrote under the name "Student".

*Studentovo rozložení

- Pro malé vzorky o ***k*** prvcích ($k < 100$) má průměr *Studentovo rozložení o* ***k − 1*** *stupních volnosti*
- Meze spolehlivosti:

9 stupňů volnosti

Pr[$X \geq z$]	Z
0.1%	4.30
0.5%	3.25
1%	2.82
5%	1.83
10%	1.38
20%	0.88

normalní rozložení

Pr[$X \geq z$]	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84

* Distribuce rozdílů

- Nechť $m_d = m_x - m_y$
- Rozdíl průměrů (m_d) má rovněž Studentovo rozložení s (k–1) stupni volnosti
- Nechť σ_d^2 je rozptyl rozdílů
- Standardizovaná verze m_d se nazývá t-statistika:

$$t = \frac{m_d}{\sqrt{\sigma_d^2 / k}}$$

- Veličina t se používá pro realizaci t-testu

*Průběh testu

1. Zvol hladinu významnosti α
   - Je-li rozdíl signifikantní na hladině α%, pak s pravděpodobností (100-α)% lze rozdíl prohlásit za významný
2. Sniž hladinu významnosti na polovinu (protože test je párový „2-tailed“)
   - Tj. skutečný rozdíl může být **+ve** nebo **−ve**
3. Najdi hodnotu z odpovídající $\alpha/2$
4. Když $t \leq -z$ nebo $t \geq z$, můžeme rozdíl prohlásit za významný
   - Tj. Nulovou hypotézu lze zamítnout!

* Nepárová pozorování

- Pokud odhady KV jsou získány z různých randomizací, nepovažují se za párové!

  (stačí např. když se pro jeden model používá ***k*** – násobná KV a pro druhý ***j*** -násobná KV)

- V takovém případě používáme *nepárový* t-test s **min(*k* , *j*) − 1** stupni volnosti

- Výsledná *t*-statistika

$$t = \frac{m_d}{\sqrt{\sigma_d^2 / k}} \quad \Rightarrow \quad t = \frac{m_x - m_y}{\sqrt{\frac{\sigma_x^2}{k} + \frac{\sigma_y^2}{j}}}$$

*Interpretace výsledků

- Všechny naše odhady z KV vycházejí z výsledků získaných na témže souboru dat
- Tedy test říká pouze, jestli existuje rozdíl pro *úplnou* (*complete*) *k-násobnou* KV na tomto souboru dat
  - Úplná *k*-násobná KV generuje všechna možná disjunktní pokrytí dat vedoucí ke *k* skupinám a průměruje získané výsledky
- Ideálně by bylo nejlépe používat různá data pro získání každého *k*-násobného KV odhadu v testu

- **t-statistika je pro DM velmi užitečná**!!!

Shrnutí:

- Jsou-li k dispozici ROZSÁHLÁ data, rozdělí se na disjunktní **trénovací**, **testovací** a **validační podmnožiny**
- Nevyvážená data je nutné vhodně upravit
- Křížová validace je zvlášť vhodná pro MALÉ objemy dat
- Je nutné dbát, aby testovací data NEBYLA použita pro ladění parametrů metody – k tomu slouží data validační

- **Především je třeba se vyhnout přeučení** (*overfitting*) **!**

Dopručená literatura

- Petr Berka: ***Dobývání znalostí z databází***, Academia, Praha 2003
- Kap.11 *Strojové učení v dobývání znalostí* (F. Železný, J. Kléma, O. Štěpánková) v ***UI*** (4)
- V. Mařík, O. Štěpánková, J. Lažanský: ***Umělá inteligence*** (4), Academia, Praha 2003
- I.H. Witten, E. Frank, M.A. Hall: ***Data Mining: Practical Machine Learning Tools and Techniques*** (Third Edition), 3rd edition, Morgan Kaufmann 2011