Vytěžování dat, přednáška 11: Testování modelů

Miroslav Čepek

Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Fakulta elektrotechnická, ČVUT

Chyba modelu

- Jak zjistíme jestli je vytvořený model dobrý a případně jak moc je dobrý?
- Musíme jej vyzkoušet.
- Zkusíme aplikovat model na data a podíváme se, jak model funguje...
- Ale na jaká data?
- Ideálně na všechny vstupní vzory, které se kdy mohou objevit (a ideálně i se stejnou distribucí).
- Ale takových je nekonečně mnoho. Ale máme (snad) reprezentativní vzorek – trénovací množinu.

Chyba na trénovacích datech

- Můžeme zjistit chybu modelu na trénovacích datech?
- No jasně, můžeme.
- A je to dobrý odhad chyby modelu na neznámých datech?
- Ne, není! Proč?
- Vůbec totiž neříká, jak se bude model chovat pro neznámá data.

Špatný model

- Představme si klasifikační metodu, která si jen zapamatuje vstupní vzory a pokud přijde na vstup zapamatovaný vzor, odpoví zapamatovanou třídou. Jinak odpoví – `NIL`.

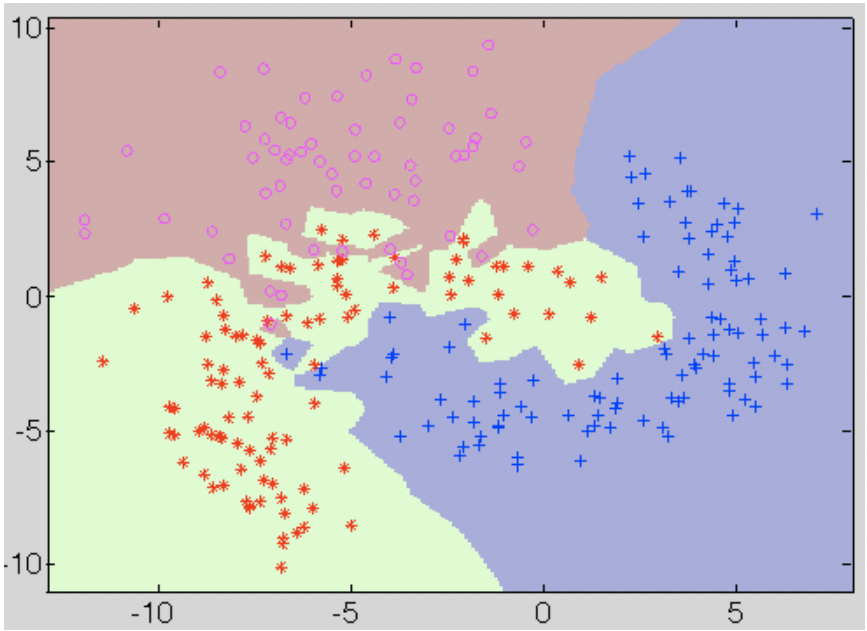
Trénovací data:

x_1	x_2	x_3	x_4	y
1.0	1.0	1.5	1.8	"A"
1.0	1.1	1.4	1.7	"A"
2.0	2.0	1.9	0.9	"B"
2.0	0.0	1.8	1.1	"B"

- Jakou má chybu na trénovacích datech?
- Nulovou!
- A jakou chybu bude mít na neznámém vzoru? Třeba $(1.1, 1.05, 1.55, 1.85)$.
- 100%. **Čili nedokáže generalizovat!** Neumí zobecnit vlastnosti, které jsou schované v trénovacích datech.

Přeučení

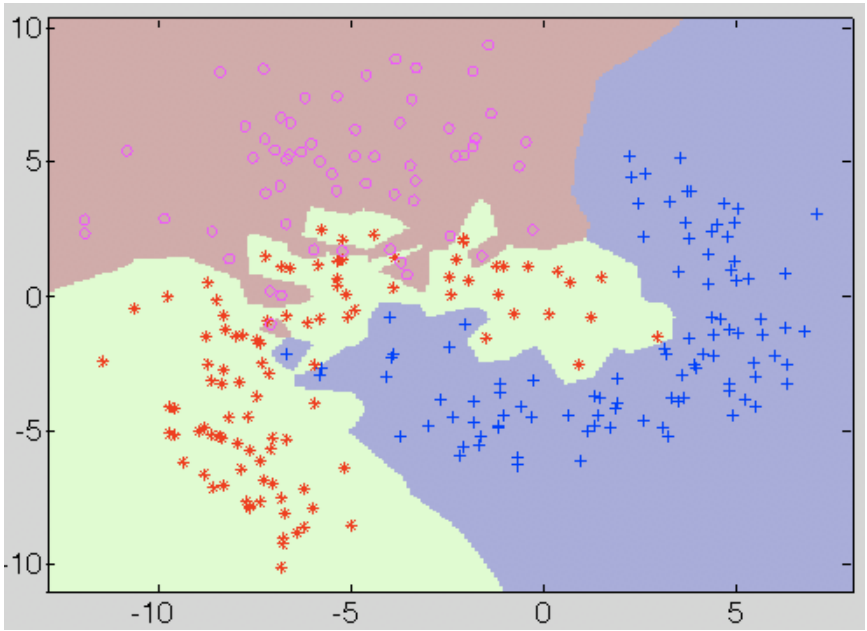
- Druhým a neméně důležitým problémem je přeučení.
- Jde o to, že model se naučí i závyslosti, které v datech nejsou.
- Představte si, že se snažím predikovat, zda bude pršet, svítit sluníčko nebo bude zataženo, podle toho, jaká je teplota a vlhkost vzduchu.
- Při měření se ale občas přehlédnu a zapíši špatný údaj.



http://perclass.com/doc/guide/classifiers.html

Trénovací data

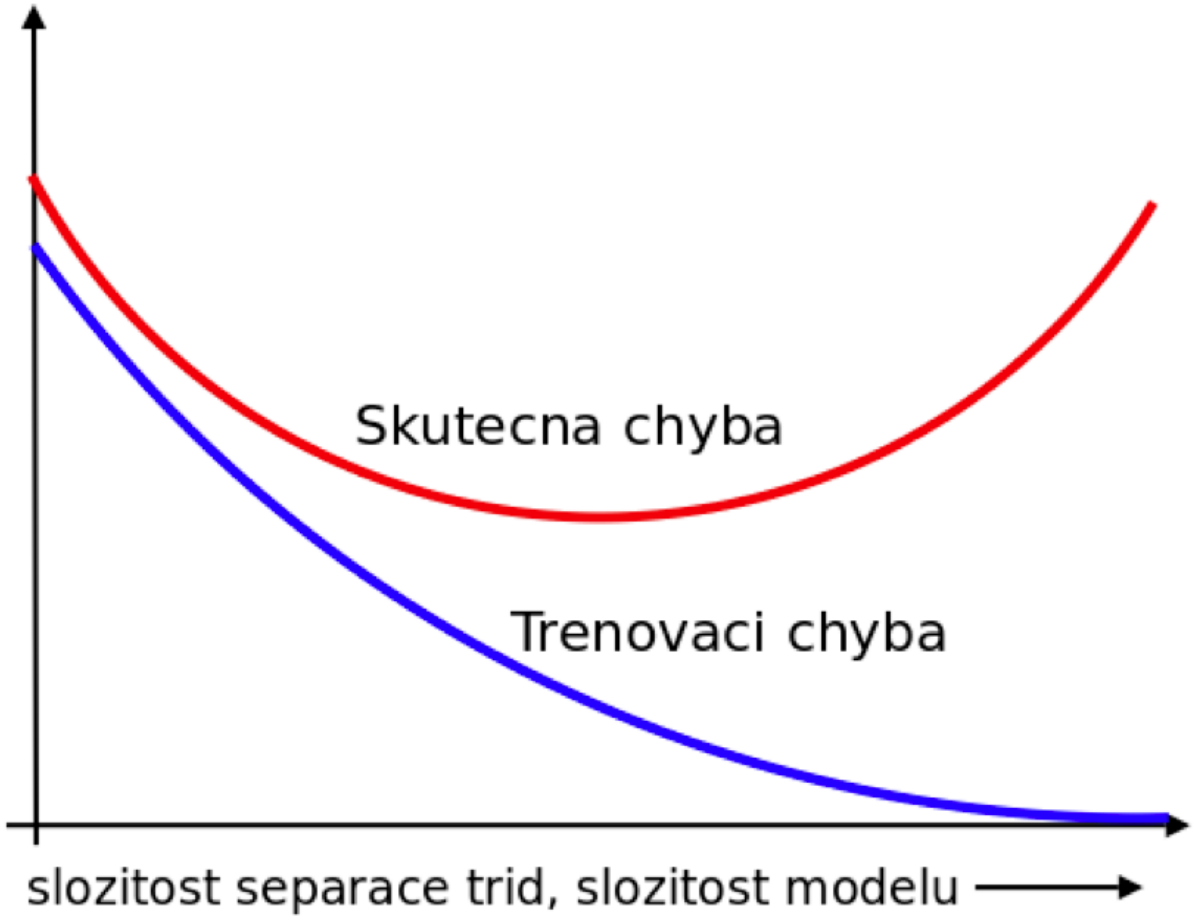
- Ale tyto nechci, aby se tyto chyby model naučil.
- Naopak chci, aby ostatní (správná) data tyto chyby překryla.



Testovací data

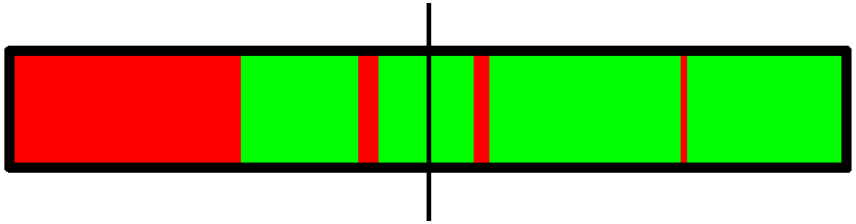
- Jak se tedy vyzrát na tento problém?
- Respektive, jak zjistit, jestli je model opravdu dobrý?
- Zkusím jej na datech, které model neviděl při učení!
- Tímto získám **nevychýlený** odchad chyby modelu na skutečných (doposud neznámých) datech.

Chyba na trénovacích a Testovacích datech



Trénovací a testovací data

- Jak získat trénovací a testovací množinu?
- Už jsme na to narazili – rozdělím náhodně data na trénovací a testovací část.
- Proč náhodně? Nemůžu jen vzít první a druhou polovinu instancí?



Chyba modelu při jiných testovacích datech

- Když zkusím spočítat chybu modelu na jiných testovacích datech, získám stejnou chybu?
- (nejen) Chybovost modelu je vlastně náhodná veličina s (většinou) normálním rozdělením.
- Tím, že spočítám chybu na testovací množině, získám jednu realizaci náhodné proměnné.
- Když budu počítat chybu na různých testovacích množinách, získám několik realizací chybové náhodné proměnné a můžu spočítat průměrnou chybu a získat tak představu, jak moc je jeden konkrétní model dobrý/špatný.

Chyba modelu při jiných trénovacích datech

- Když znovu spustím učení modelu, vznikne mi vždy naprosto stejný model?
- Teď už vím, jak moc je špatný jeden model. Ale co když mám smůlu a tento model se naučil výrazně hůře/lépe než jiný model vytvořený stejnou metodou.
- Vím jak moc je dobrá jedna BP neuronová síť, ale jak moc jsou dobré všechny BP neuronové sítě? A jsou lepší než naivní bayesovská síť?
- Zase parametry chyby jedné realizace modelu jsou jen náhodnými proměnnými všech modelů naučených na tato data.

Statistika chyb

- Takže průměrná chyba jednoho modelu je zase jen jedna z množných realizací náhodné veličiny *hodnota průměru všech modelů této modelovací metody*.
- Čili, pokud vytvořím a spočítám průměrnou chybu pro jeden model, nemusí to nic znamenat o jiných modelech naučených stejnou technikou.

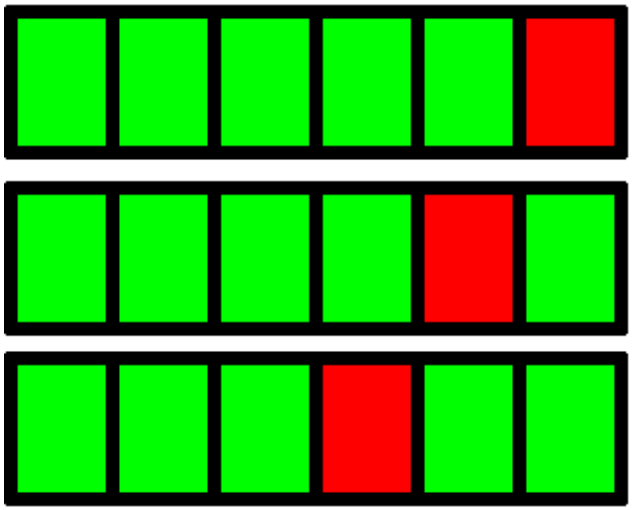
!POMOC!

X-fold cross validace (1)

- Řešením je opakovaně rozdělovat data na trénovací a testovací množinu.
- A to buď náhodně a nebo nějak systematicky.
- A uznávaný systematický přístup je křížová validace.
- Ta funguje tak, že data rozdělím do N částí.
- Zlatý standard je poružít 10 částí. Pak se mluví o 10 cross fold validation.
- A to buď náhodně nebo i podle pořadí.

X-fold cross validace (2)

- A model pak postavím na N-1 částech (foldech) a na poslední model otestuji.
- Tím získám jeden odhad chyby.
- Posunu se o jedna doprava a zase postavím model na (N-1) částech a na zbylém otestuji.



X-fold cross validace (3)

- Takto získám N odhadů chyby.
- Z toho již dokáži spočítat statistiku – například průměrnou chybu klasifikátoru a získat tak poměrně přesný (uvěřitelný) odhad chyby dané klasifikační metody na předložených datech.
- Navíc s těmito N odhady mohu provádět další statistické testy a vizualizace (boxploty, t-testy, ...)
- Navíc každý vstupní vzor bude v testovací množině právě jednou. Čili získám představu, jak klasifikátor bude fungovat pro tento konkrétní vzor.

Použití křížové validace

- Možné příklady použití křížové validace:
- odhad přesnosti modelu na datech,
  - Provedu křížovou validaci a průměr chyb z každé z N validací je nevychýleným odhadem chyby modelu.
- výběr vhodných parametrů modelu,
  - Vytvořím modely s různými parametry a na každý z nich spustím křížovou validaci. A opět spočítám pro každé nastavení modelu průměrnou chybu z křížové validace a vyberu tu konfiguraci (ty parametry), které mají nejmenší průměrnou chybu.
- porovnání modelovacích metod.
  - Pro každou modelovací metodu spočítám průměrnou chybu pomocí křížové validace a vyberu tu metodu, která má nejmenší průměrnou chybu.

Validační množina

- Zejména při použití křížové validace pro určení parametrů se ještě používá tzv. validační množina.
- Jde o to, že před začátkem křížové validace z dat odeberu část – validační množinu.
- Na zbytku spustím křížovou validaci a najdu optimální parametry.
- Pak naučím model s těmito optimálními parametry na celé datové množině, kterou jsem předtím použil pro křížovou validaci a abych měl jistotu, že učení modelu dopadlo dobře, naučený model nechám oklasifikovat validační množinu a spočítám validační chybu.

Nevýhoda trénovacích/testovacích/validačních chyb

- Představme si datovou množinu se dvěma třídami – *zdraví pacienti* a *nemocní pacienti*.
- Zdravých pacientů je 95% dat a nemocných je zbývajících 5% pacientů.
- Jakou chybu na testovacích datech (vybraných jako podmnožinu z tohoto datasetu) bude mít klasifikátor, který bude předpovídat, že všichni pacienti jsou v pořádku?
- 95% – to je super klasifikátor! Ale dělá něco užitečného?
- NE! Takový klasifikátor je k ničemu.
- Dokáži zjistit z testovací chyby, že klasifikátor provádí něco takového?

Matice záměn

- Řešením je matice záměn.

	true 1	true 2	true 3	true 5	true 6	true 7	class precision
pred. 1	69	0	0	0	0	0	100.00%
pred. 2	1	74	4	0	0	0	93.67%
pred. 3	0	1	13	9	2	0	52.00%
pred. 5	0	0	0	2	4	0	33.33%
pred. 6	0	0	0	0	0	0	0.00%
pred. 7	0	1	0	2	3	29	82.86%
class recall	98.57%	97.37%	76.47%	15.38%	0.00%	100.00%	

Specificita a senzitivita (1)

- Pokud mám binární klasifikátor (tj klasifikátor, který zařazuje do dvou tříd), mohu čísla v matici záměn kvantifikovat číslem.
- Často se používá specificita a senzitivita.
- Abychom je dokázali spočítat, musíme se nejdřív zamyslet nad zavést pojmy:
  - Positive examples – jedna z tříd binárního klasifikátoru (v našem příkladě lidé mající nemoc).
  - Negative examples – druhá z tříd (v našem příkladě zdraví lidé).

Specificita a senzitivita (2)

- True positives (TP) – vzory, které model správně označil jako pozitivní (tj lidé, kteří jsou ve skutečnosti jsou nemocní a model je také označil za nemocné).
- True negatives (TN) – vzory, které model správně označil jako negativní (tj lidé, kteří jsou ve skutečnosti jsou zdraví a model je také označil za zdravé).
- False positives (FP) – vzory, které model mylně označil jako pozitivní (tj lidé, kterí jsou ve skutečnosti zdraví, ale model je označil za nemocné).
- False negatives (FN) – vzory, které model mylně označil jako negativní (tj lidé, kterí jsou ve skutečnosti nemocní, ale model je označil za zdravé).

Specificita a senzitivita (2)

- Když se podívám do matice záměn, můžu přímo zjistit počty vzorů spadající do jednotlivých škatulek (TP, TN, FP, FN).

	Actually positive	Actually negative
Predicted positive	#True positives	#False positives
Predicted negative	#False negatives	#True negatives

Specificita a senzitivita (3)

- Teď můžu konečně spočítat specificitu a senzitivitu.

$$\text{specificita} = \frac{\#\text{True negatives}}{\#\text{True negatives} + \#\text{False negatives}}$$

$$\text{sensitivita} = \frac{\#\text{True positives}}{\#\text{True positives} + \#\text{False positives}}$$

- Specificita je tedy procento správných "negatives" ze všech vzorů, které byly označeny za negatives (Procento skutečně zdravých lidí mezi všemi, kteří byli modelem označeni za zdravé).
- Senzitivita je tedy procento správných "positives" ze všech vzorů, které byly označeny za positives (Procento skutečně nemocných lidí mezi všemi, kteří byli modelem označeni za nemocné).

Specificita a senzitivita (4)

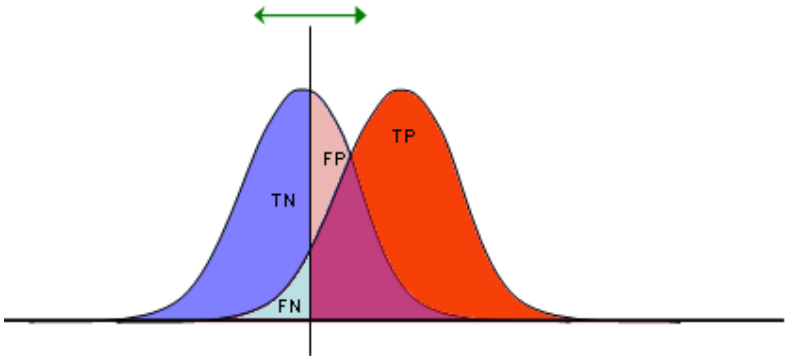
- Když se vrátím k příkladu s klasifikátorem, který klasifikuje všechny lidi, jako zdravé. Jaká bude specificita a senzitivita?
- Specificita $= 1.0$
- Senzitivita $= 0.0$

ROC křivka (1)

- Specificita a senzitivita jsou jen jednou z možností, jak vyhodnocovat možnosti a vlastnosti binárního klasifikátoru.
- Další hojně využívanou možností je ROC křivka.
- Typickým výstupem binárního klasifikátoru není přímo hodnota *Positive*/*Negative*, ale většinou klasifikátor vrátí číselnou hodnotu a pomocí prahu prozhodnu, kam aktuální vzor zařadím.
- Typicky se práh volí 0.5, ale jak se změní chyba klasifikátoru, když změní práh?

ROC křivka (2)

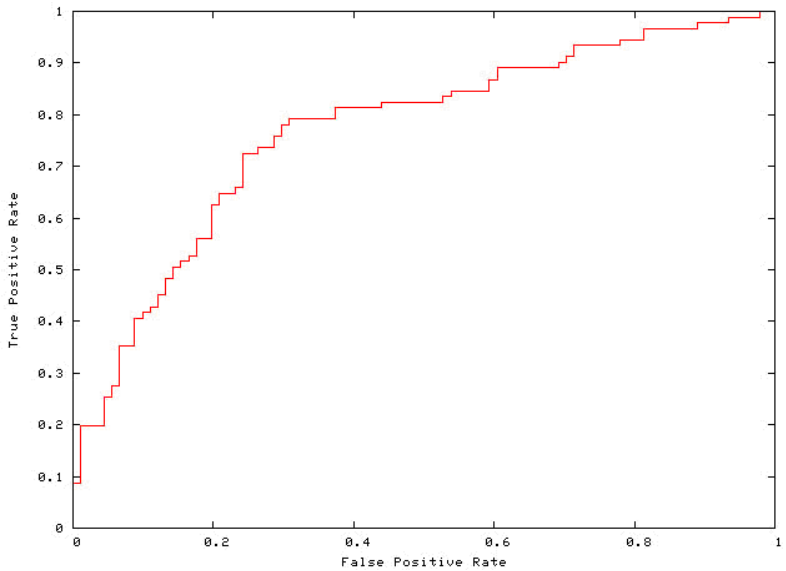
- Mějme klasifikátor, který na tělesné teploty klasifikuje, zda se jedná o zdravého nebo nemocného. Pokud má člověk teplotu menší než práh, jedná se o zdravého člověka. Pokud větší, jedná se o nemocného.



- Pokud prahem posunu doprava, klasifikuji správně více zdravých lidí, ale (z principu) se mezi ně připletou i nemocní.
- A obráceně při posunu prahu doleva.

ROC křivka (3)

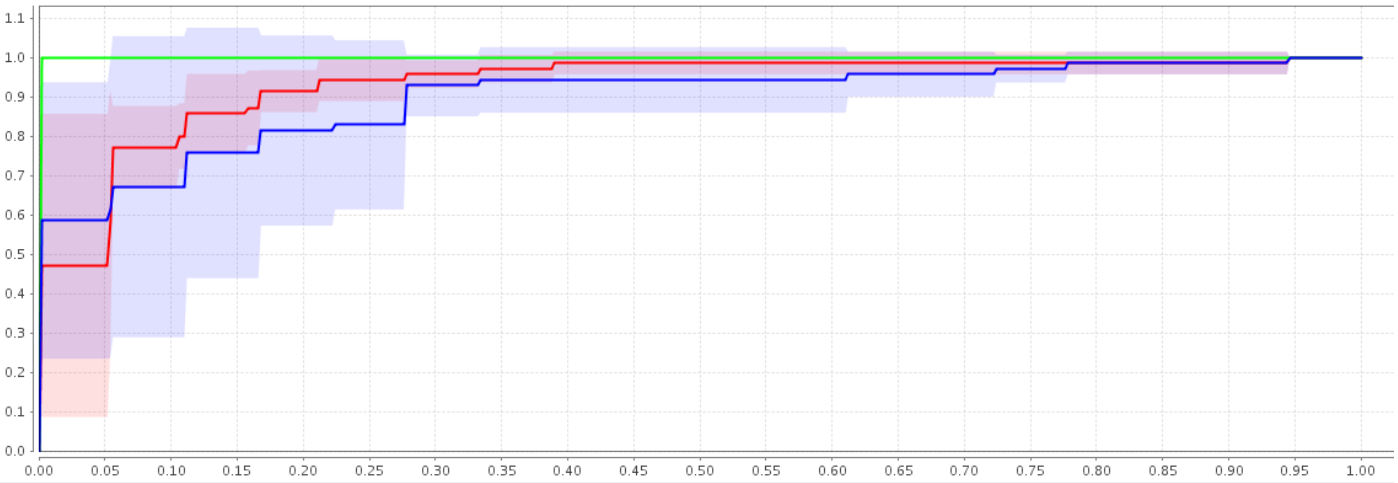
- Pak můžu nakreslit graf, kde na
  - na ose Y je počet true positives,
  - na ose X je počet false positives.



http://research.cens.ucla.edu/projects/2006/Multiscaled_Actuated_Sensing/Classification_Minirhizotron/default.htm

ROC křivka (4)

- Jak bude vypadat ROC křivka ideálního klasifikátoru (který bezchybně rozděluje obě třídy)?
- Jak bude vypadat ROC křivka náhodného klasifikátoru (který má chybu 50%)?



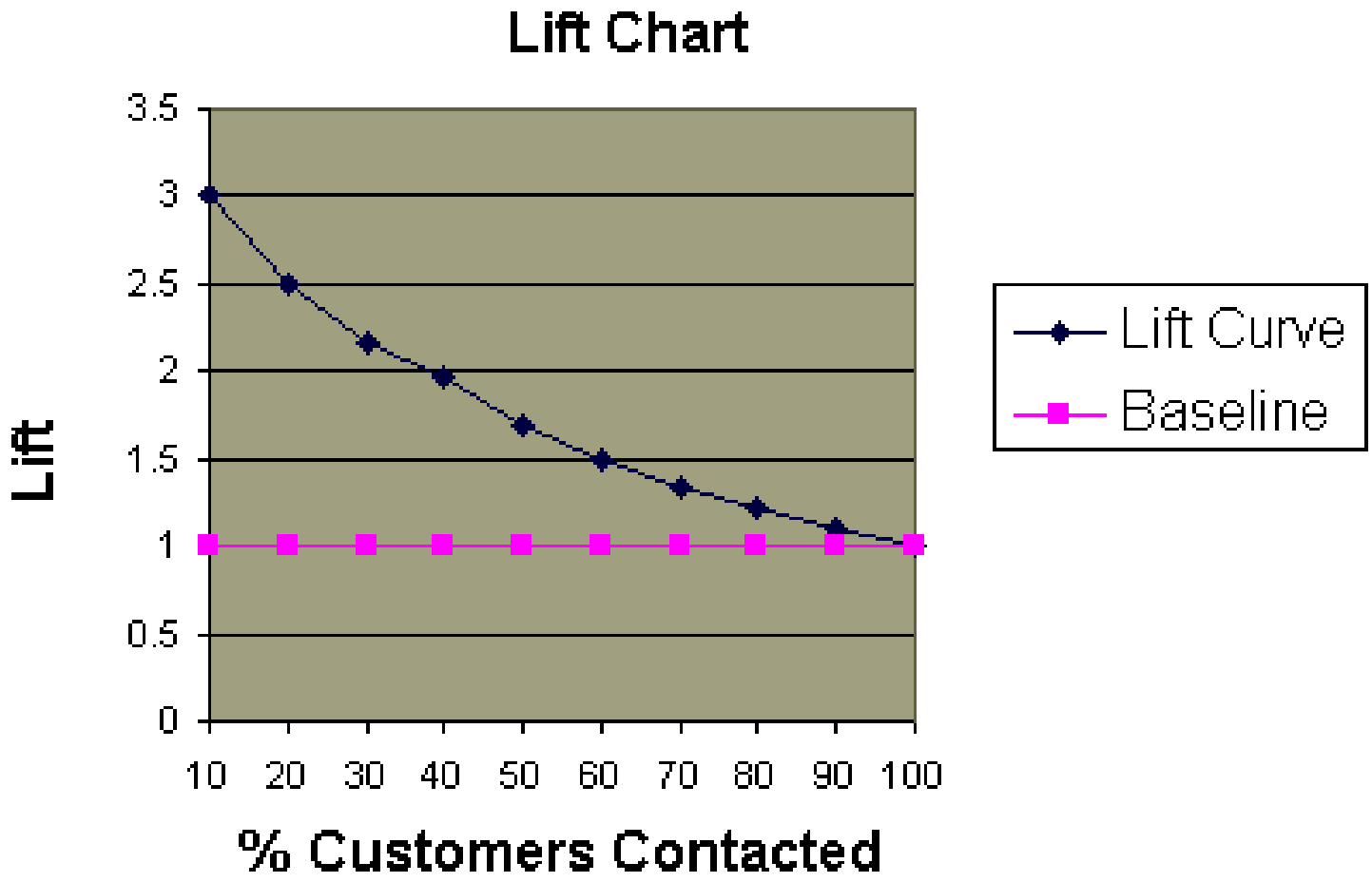
- Pro lepší posouzení kvality ROC křivek můžu použít plochu, kterou shora ohraničuje ROC křivka (Area under curve).

Lift

- Lift je další způsob, jak měřit kvalitu klasifikátoru ve specifických úlohách.
- Mám klasifikátor, který identifikuje zákazníky, kteří by mohli kladně odpovědět na marketingovou nabídku.
- A ptám se, když oslovím 10% všech mých zákazníků, které model identifikoval jako nejnadějnější, kolik procent zákazníků, kteří by skutečně odpověděli jsem oslovím.
- $\mathit{lift} = \frac{\text{procento oslovených zákazníků, kteří budou kladně reagovat}}{\text{procento oslovených zákazníků}}$
- Například oslovím 10% zákazníků T-Mobilu a mezi nimi je 50% těch, kteří si skutečně zaplatí nový internet do mobilu, mám $\mathit{lift} = \frac{50\%}{10\%} = 5$.

Lift křivka

- Když vynesu do grafu lift pro různé počty oslovených zákazníků, získám lift chart, který vypadá takto:



Zdroje a další čtení

- http://gim.unmc.edu/dxtests/ROC3.htm
- http://en.wikipedia.org/wiki/Receiver_operating_characteristic
- http://en.wikipedia.org/wiki/Sensitivity_and_specificity
- http://www2.cs.uregina.ca/~dbd/cs831/notes/lift_chart/lift_chart.html
- http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html