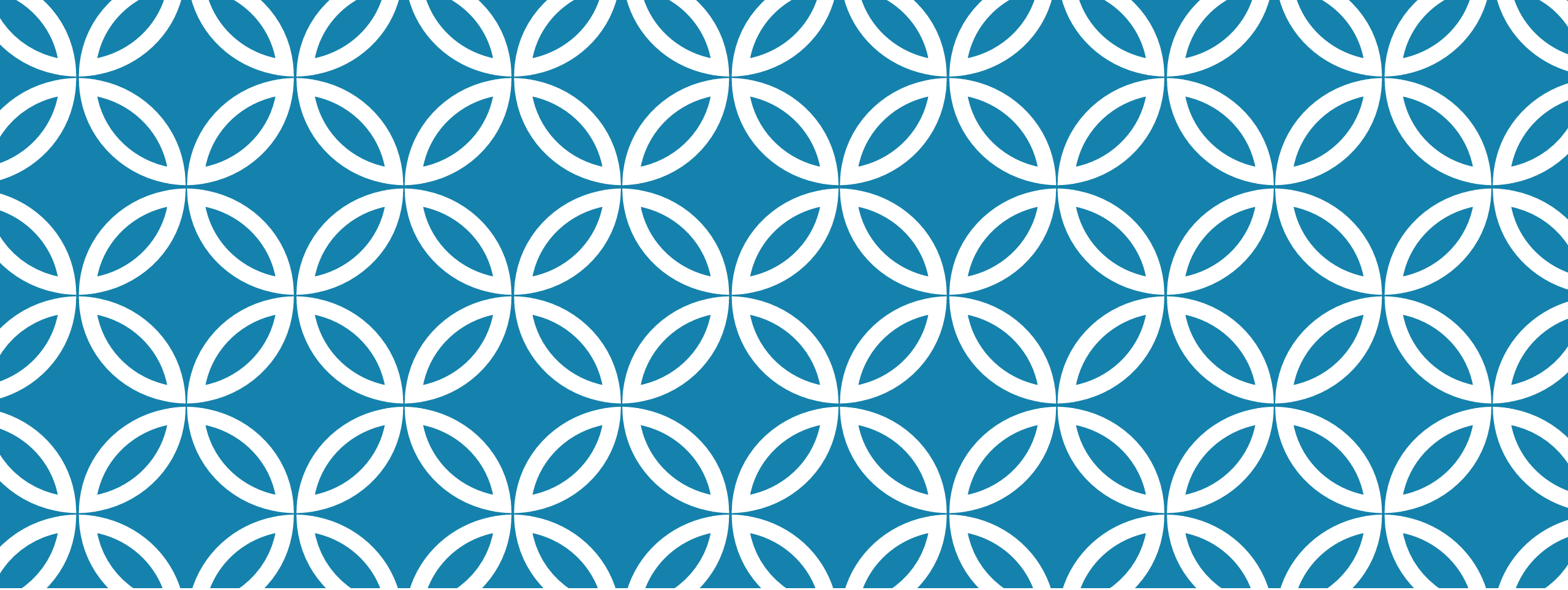


STROJOVÉ UČENIE

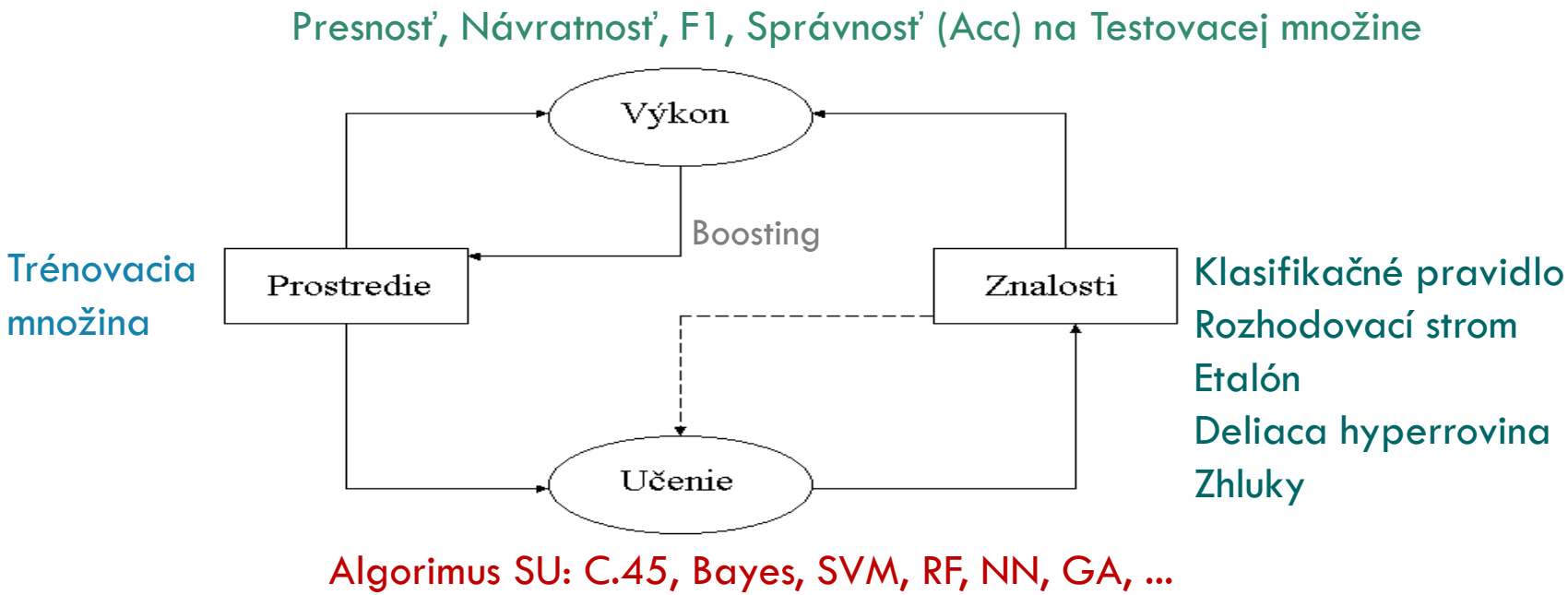
prof. Ing. Kristína Machová, PhD.

OSNOVA

- Strojové učenie
- Metódy podporných vektorov
- Rozhodovacie stromy
- Náhodné stromy
- Úlohy NLP
- Hlboké učenie (RNN, LSTM, GRU, Transformer)

ČO JE TO STROJOVÉ UČENIE?

- Človek učením nadobúda schopnosť rozlišovať stále jemnejšie rozdiely (kvet - motýľ, cestovanie, štúdium, ...)
- Počítačový program je schopný učiť sa zo skúsenosti (trénovacie príklady), ak sa jeho výkon zvýšil (Presnosť, Návratnosť, F1, Správnosť (Acc)) pri riešení triedy úloh (klasifikačná, regresná, zhlukovanie) vďaka danej skúsenosti.



PARADIGMY SU - REPREZENTÁCIA VÝSTUPOV – TYPY MODELOV

Symbolické učenie a indukcia pravidiel

- Klasifikačné pravidlá
- Rozhodovacie stromy a zoznamy

Pravdepodobnostné a štatistické modely

- Deliaca hyper-rovina (Lineárna prahová jednotka, Metóda podporných vektorov)
- Etalón – typický reprezentant
- Pravdepodobnostný popis (Naivný Bayes klasifikátor)

Hlboké učenie

- Umelé neurónové siete (CNN, RNN, LSTM, GRU, Transformer)

Učenie súborom metód

- Množina RS resp. iných modelov (bagging, boosting, XGBoost)
- Množina de-korelovaných stromov RS - (náhodný les – RS (Random Forest))

Federované učenie

- Učenie viacerých modelov na separovaných dátových podmnožinách a ich následná agregácia v centre

Aktívne učenie

- Zhluky kombinované napr. extenzionálnym modelom (lenivé učenie (K-NN))
- Kombinácia zhlukovacích techník a klasifikátorov

DELENIE STROJOVÉHO UČENIA

Kontrolované (supervised)

trénovacie príklady pozostávajú zo vstupno/výstupných vzorov. Cieľom učiaceho algoritmu je predikovať výstupnú hodnotu nových príkladov na základe ich vstupnej hodnoty

Nekontrolované (unsupervised)

algoritmus má k dispozícii iba vstupné hodnoty na objavenie zhlukov, zmysluplných asociácií alebo vzorov

Posilňované (reinforcement)

prostredie v ktorom sa algoritmus učí nie je reprezentované trénovacími príkladmi ale stavovým priestorom

Tri základné prístupy:

- Klasické strojové učenie
- Učenie súborom metód
- Hlboké učenie

PRÍKLADY TRÉNOVACEJ MNOŽINY – REPREZENTÁCIA VSTUPOV

Počasie	Teplota[°C]	Vlhkosť [%]	Vietor	Záver/Trieda
slnečno (1)	27	90	áno (1)	Kniha
slnečno (1)	21	70	nie (2)	Golf
zamračené (2)	28	78	nie (2)	Golf
daždivo (3)	18	70	áno (1)	Kniha
daždivo (3)	24	80	nie (2)	Golf
. . .				

Trénovacia množina obsahuje trénovacie príklady TP

TP predstavujú pozorovania vo forme vstupno/výstupných vzorov (x_i, y_i)
x_i ... reprezentuje hodnoty atribútov
y_i ... reprezentuje triedu – anotáciu (labelling) od experta, supervizora – to umožňuje učenie

Atribúty (binárne, nominálne, numerické, ordinárne, hierarchické)

RELÁCIA MEDZI VSTUPOM A VÝSTUPOM

Počasie	Teplota[°C]	Vlhkosť [%]	Vietor	Záver/Trieda
x1				y1
x2				y2
x3				y3
x4				y4
x5				y5
. . .				

Hľadáme vzťah medzi vstupom a výstupom v tvare
funkcie $y_i = f(x_i)$
ak ide o regresnú úlohu (y_i a x_i sú numerické hodnoty)

RELÁCIA MEDZI VSTUPOM A VÝSTUPOM

Počasie	Teplota[°C]	Vlhkosť [%]	Vietor	Záver/Trieda
x1				c(x1)
x2				c(x2)
x3				c(x3)
x4				c(x4)
x5				c(x5)
. . .				

Hľadáme vzťah medzi vstupom $\mathbf{x_i}$ a výstupom v tvare $\mathbf{c(x_i)}$
ak ide o klasifikačnú úlohu

- vstupy $\mathbf{x_i}$ sú reprezentované nominálnymi aj numerickými hodnotami
- $\mathbf{c(x_i)}$ sú kategórie (Category) resp. triedy trénovacích príkladov
- trénovacie príklady sú dvojice $\mathbf{[x_i, c(x_i)]}$

Konzistentná hypotéza h(x) je taká hypotéza, pre ktorú platí h(x)=c(x) nad trénovacou množinou, teda príkladmi [x, c(x)]

PRÍKLADY TRÉNOVACEJ MNOŽINY – ANOTÁCIA (LABELLING)

1. Only lies, utter tosh. Name one fact!!
2. Never believe politics.
3. So nice, thank you.
4. You should resign immediately for that balderdash.
5. It really drives me mad.

Post	lies	tosh	never	nice	thank	resign	balderdash	really	mad	Class
1	W_{11}	W_{12}	0	0	0	0	0	0	0	Toxic
2	0	0	W_{23}	0	0	0	0	0	0	Toxic
3	0	0	0	W_{34}	W_{35}	0	0	0	0	Nontoxic
4	0	0	0	0	0	W_{46}	W_{47}	0	0	Toxic
5	0	0	0	0	0	0	0	W_{58}	W_{59}	Toxic

Class	Class	Class	Class	Class	Class
Positive	3	Hapiness	Toxic	Authority	Non-troll
Negative	2	Sadness	Toxic	Non-authority	Troll
Positive	1	Surprise	Nontoxic	Authority	Non-troll
Positive	-1	Fear	Toxic	Non-authority	Troll
Negative	-2	Disgustt	Toxic	Non-authority	Troll
Negative	-3	Anger	Nontoxic	Authority	Non-troll

PROCES STROJOVÉHO UČENIA

1. **Fáza strojového učenia modelu** z trénovacej množiny TM a následného **testovania modelu** na testovacej množine TestM (časť TM, z ktorej sa model neučil)
2. **Fáza** automatického rozhodovania – **predikcie**

Predikovať môžeme

Triedu, kategóriu – **klasifikácia**

- doména publikácie, typ emócie, polarity názoru (pozitívny/negatívny), prosperujúci podnik, chyba plechu je zavalcovateľná, falošné správy, falošné komentáre, toxické posty, predpoveď počasia (slnečno/polooblačno/dážď)

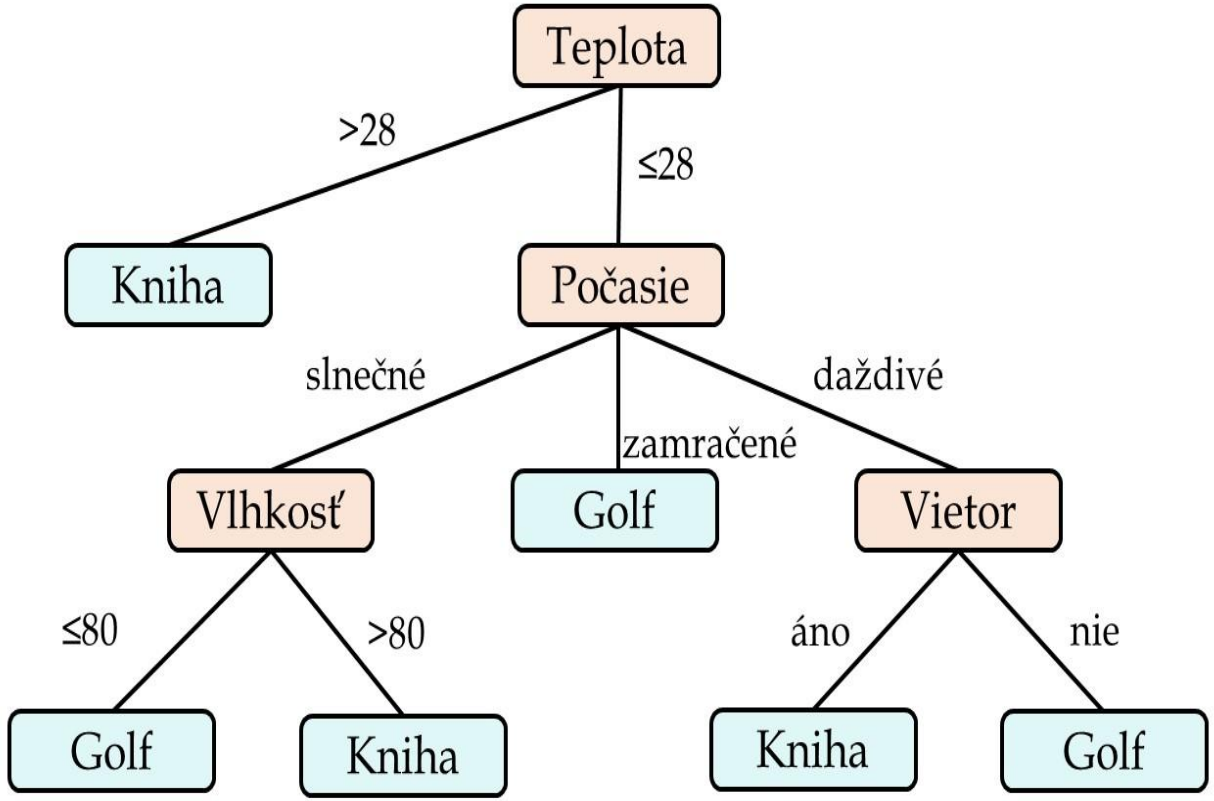
Presnú hodnotu - **regresia**

- odber elektrickej energie, vody, predpoveď teploty, odhad dôveryhodnosti reviewera (trol/netrol)

FÁZA UČENIA modelu vo forme ROZHODOVACIEHO STROMU

Počasie	Teplota	Vlhkosť	Vietor	Trieda
slnečno	27	90	áno	Kniha
slnečno	21	70	nie	Golf
zamračené	28	78	nie	Golf
daždivo	18	70	áno	Kniha
daždivo	24	80	nie	Golf

Strojové učenie sa algoritmom na generovanie Rozhodovacích stromov

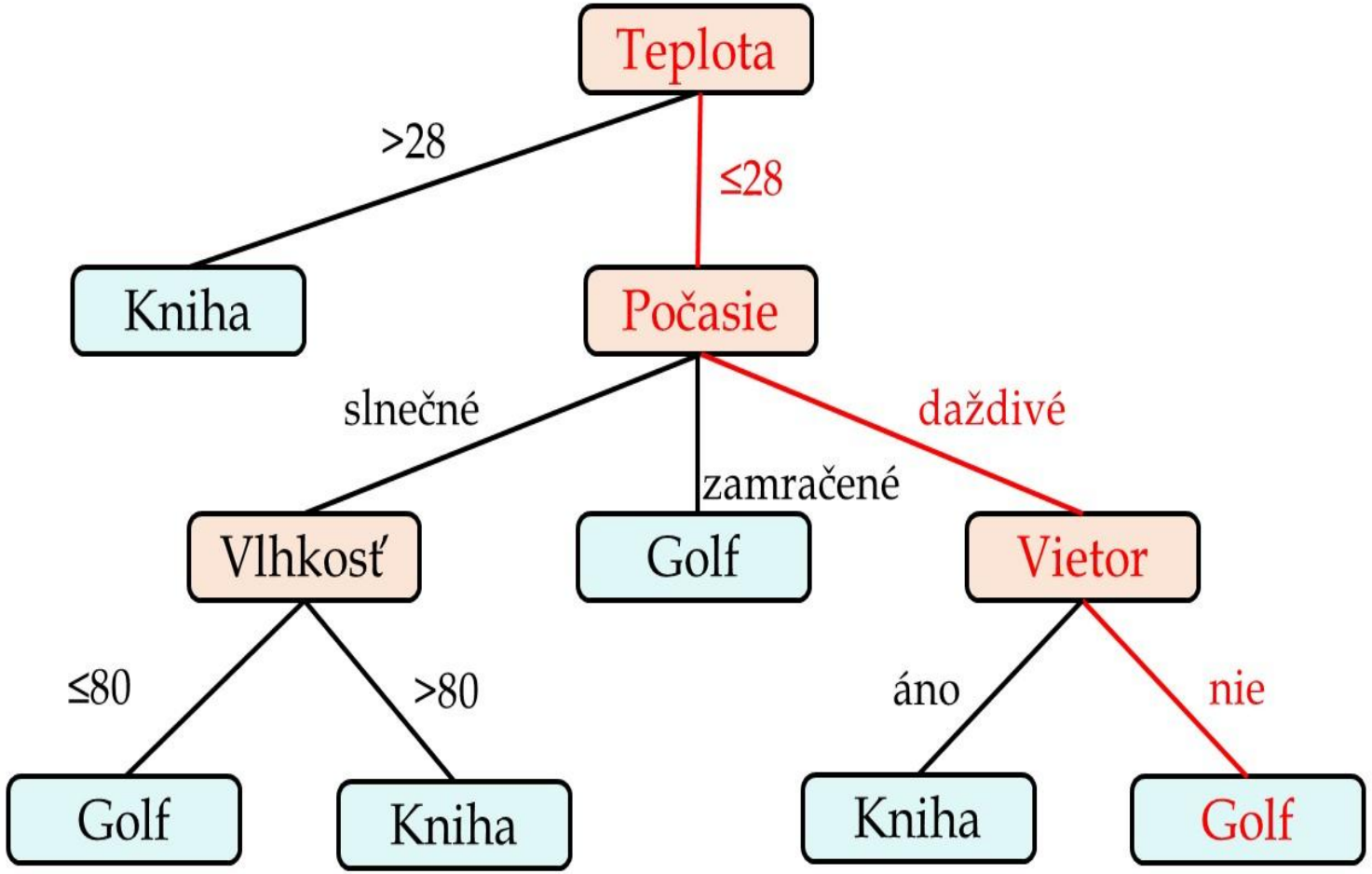


FÁZA PREDIKCIE

Strojové rozhodovanie
(klasifikácia ku triede)

Model reprezentuje
Rozhodovaciu
Procedúru - RS

PRÍKLAD
Dnes je:
26 stupňov
poprcháva
vietor nefúka



PREDIKCIA
Počasie je vhodné na golf

CHYBA KLASIFIKÁCIE

$$chyba_D(h) = P_{x \in M}[c(x) \neq h(x)]$$

C je množina cieľových pojmov

H je množina hypotéz, ktoré predstavujú aproximácie pojmov z **C**

x je náhodne vybratý príklad z M

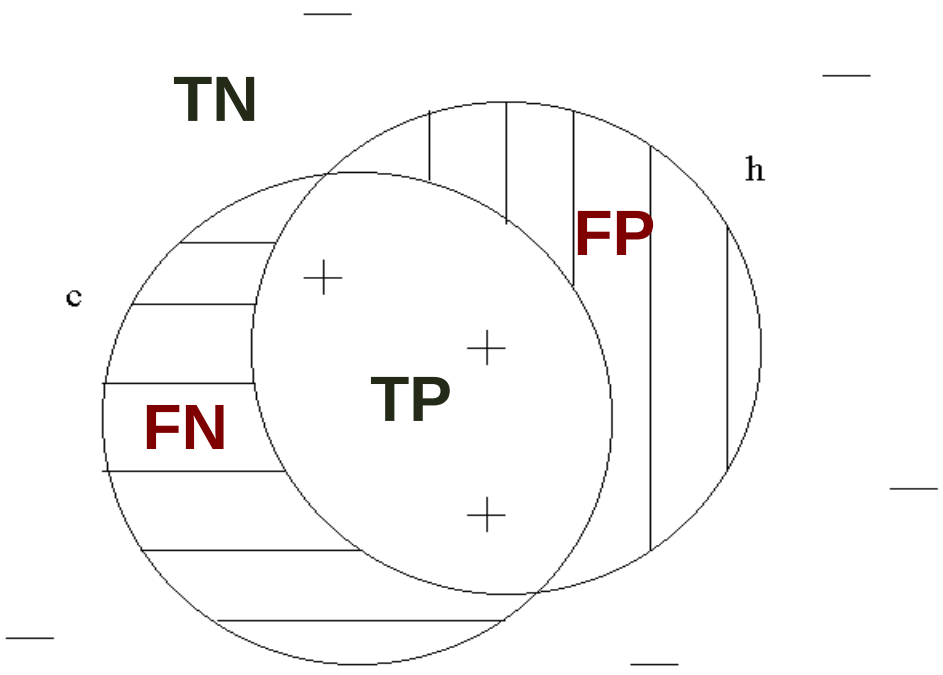
M je množina všetkých trénovacích príkladov,

TP (True Positive) – počet pozitívnych príkladov, ktoré sú **správne** klasifikované ako **pozitívne**

FP (False Positive) – počet negatívnych príkladov, ktoré sú **nesprávne** klasifikované ako **pozitívne**

FN (False Negative) – počet pozitívnych príkladov, ktoré sú **nesprávne** klasifikované ako **negatívne**

TN (True Negative) – počet negatívnych príkladov, ktoré sú **správne** klasifikované ako **negatívne**



MIERY EFEKTÍVNOSTI KLASIFIKÁCIE

	Expert (labelling) = áno	*Expert (labelling) = nie*
Predikcia systému = áno	**TP**	**FP**
Predikcia systému = nie	**FN**	**TN**

Najčastejšie používané: Presnosť (*Precision*), Návratnosť (*Recall*, resp. *Sensitivity*), *F1* a Správnosť (Accuracy - *Acc*)

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad F1 = 2\frac{Precision * Recall}{Precision + Recall}$$

Presnosť - počet správnych výsledkov pozitívnej klasifikácie ku všetkým pozitívnym klasifikáciám (reflektuje FP chyby – falošný poplach)
Návratnosť - počet správnych výsledkov pozitívnej klasifikácie ku všetkým pozitívnym príkladom (reflektuje FN chyby – neviem o probléme)
F1 – harmonický priemer Presnosti a Návratnosti
Správnosť (*Acc*) – počet správnych klasifikácií k celkovému počtu TP (*n*)

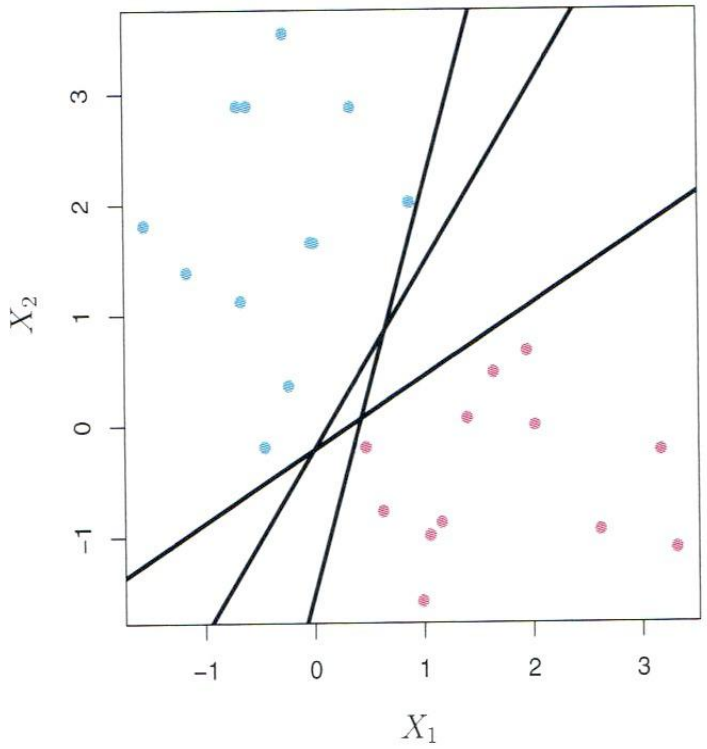
$$Acc = \frac{TP + TN}{n}$$

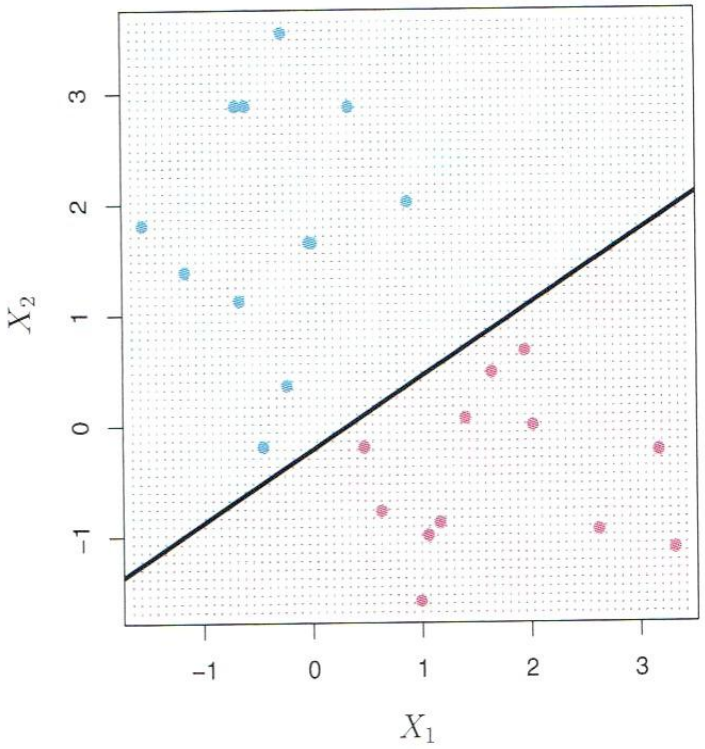
SVM – SUPPORT VECTOR MACHINES – KLASICKÉ SU

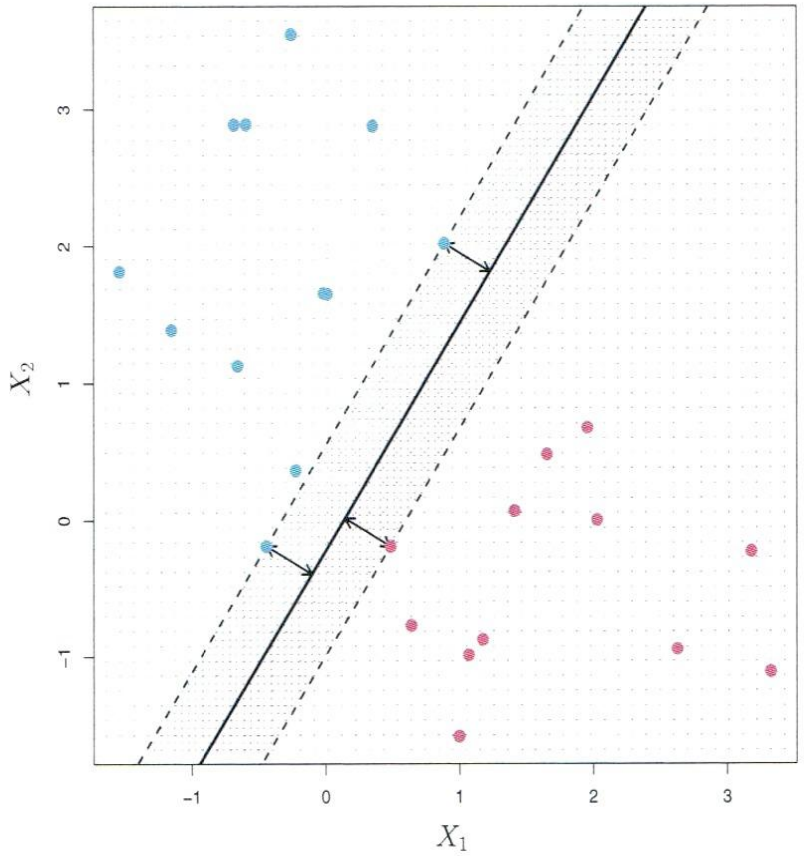
- SVM sa objavila v roku 1990
- Populárna – pri riešení širokej škály problémov dáva najlepšie výsledky
- Vhodná na spracovanie textov
- Vznikla na základe:
  - Klasifikátora maximálneho rozpätia, ktorý je takzvaný lineárny SVM a je vhodný pre lineárne separovateľné dáta
  - Jeho rozšírenie je Klasifikátor mäkkých okrajov, ktorý je možné aplikovať aj na dáta, ktoré nemusia byť prísne lineárne separovateľné
  - Ďalšie rozšírenie predstavujú **Stroje podporných vektorov SVMs,** ktoré dokážu generovať nelineárne hranice použitím rôznych nelineárnych rovníc (nelineárne separovateľné dáta)
  - Ku SVMs patrí aj **Stroj podporných vektorov SVM** založený na používaní **kernelových funkcií** (neseparovateľné dáta)

SVM – KLASIFIKÁTOR MAXIMÁLNEHO ROZPATIA

- Vo všeobecnosti existuje nekonečné množstvo takýchto hyper-rovín na perfektné separovanie jednotlivých tried od seba pre ich jasnú klasifikáciu
- Musíme definovať hodnotiacu funkciu na výber optimálneho riešenia.
- Tento problém rieši **Klasifikátor maximálneho rozpätia**

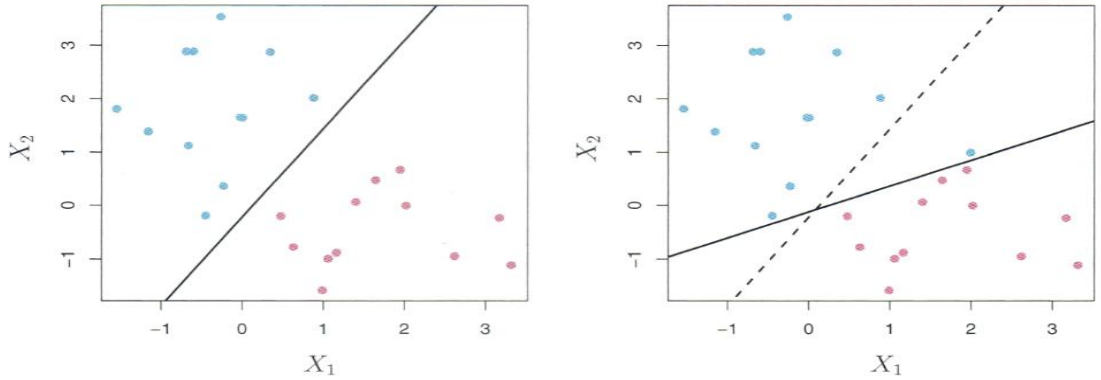


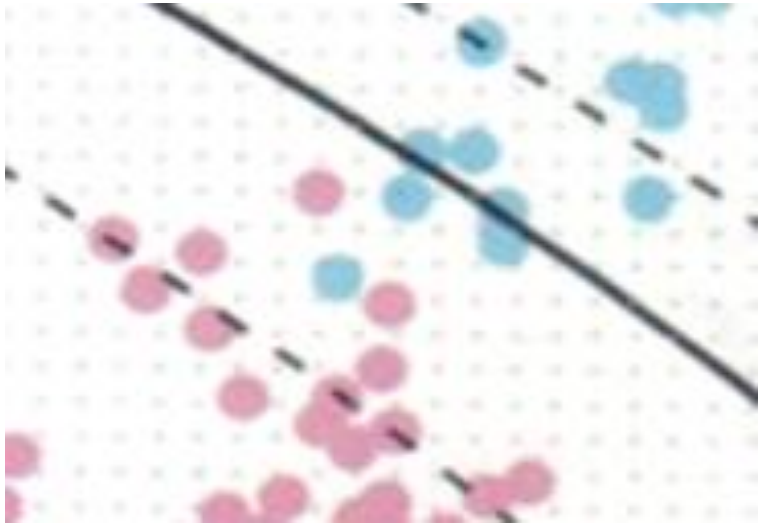




KLASIFIKÁTOR MÄKKÝCH OKRAJOV

- Klasifikátor maximálneho rozpätia - stačí pridať jeden nový TP a poloha hyper-plochy sa pootočí o takmer 40 stupňov a má tenšie rozpätie.
- Príliš tenké rozpätie je problém!
- Dostatočná vzdialenosť klasifikovaného príkladu od deliacej hyper-plochy je zárukou toho, že nové príklady (na ktorých nebol SVM trénovaný) budú správne klasifikované.
- **Riešenie**:
- Obetujeme chybnú klasifikáciu zopár trénovacích príkladov za lepšiu klasifikáciu zvyšnej väčšiny, teda použijeme takzvané **mäkké okraje (soft margins).**
- Vznikne **Klasifikátor mäkkých okrajov - Soft Margin Classifier.**
- Riešením je ďalšia optimalizácia.

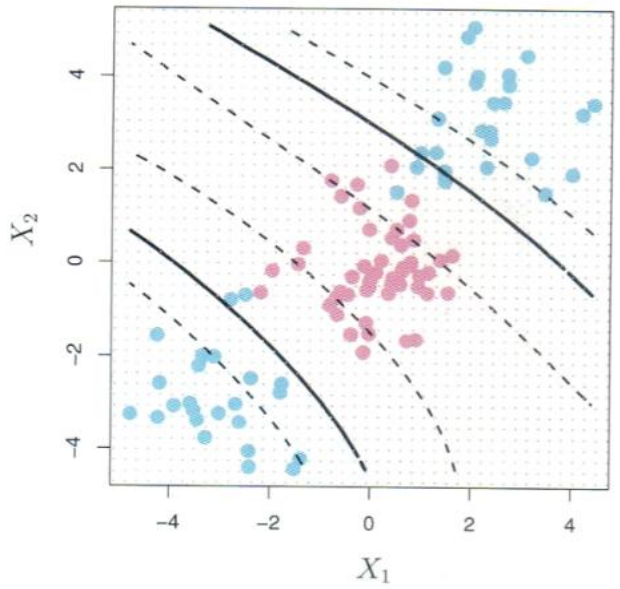


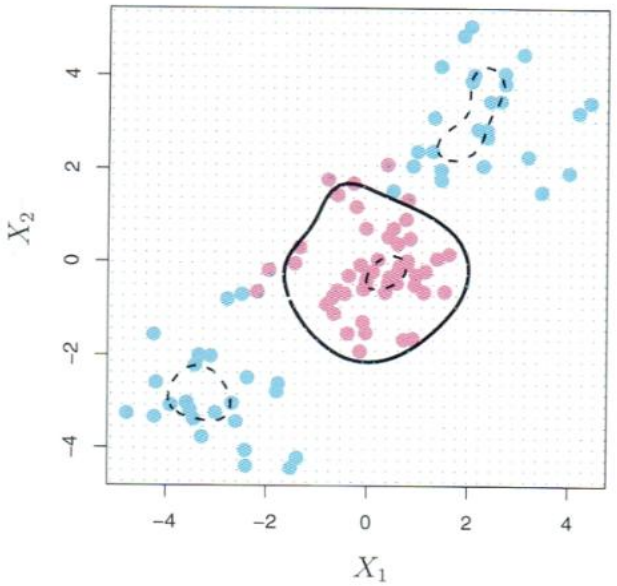


KLASIFIKÁTOR S NELINEÁRNOU ROZHODOVACOU HRANICOU

- V praxi máme často k dispozícii dáta, ktoré vyžadujú skôr hranicu simulovanú nelineárnou funkciou
  - Kvadratickou, kubickou
  - polynomiálnou funkciou vyššieho rádu
- Klasifikátor s nelineárnou rozhodovacou hranicou môže viesť k nezvládnuteľnej výpočtovej zložitosti.
- V takom prípade lepšie použiť **kernelove funkcie** a **Stroj podporných vektorov** (jeden) SVM.

Polynomiálny Kernel stupňa 3





Radiálny Kernel

KERNELOV TRIK

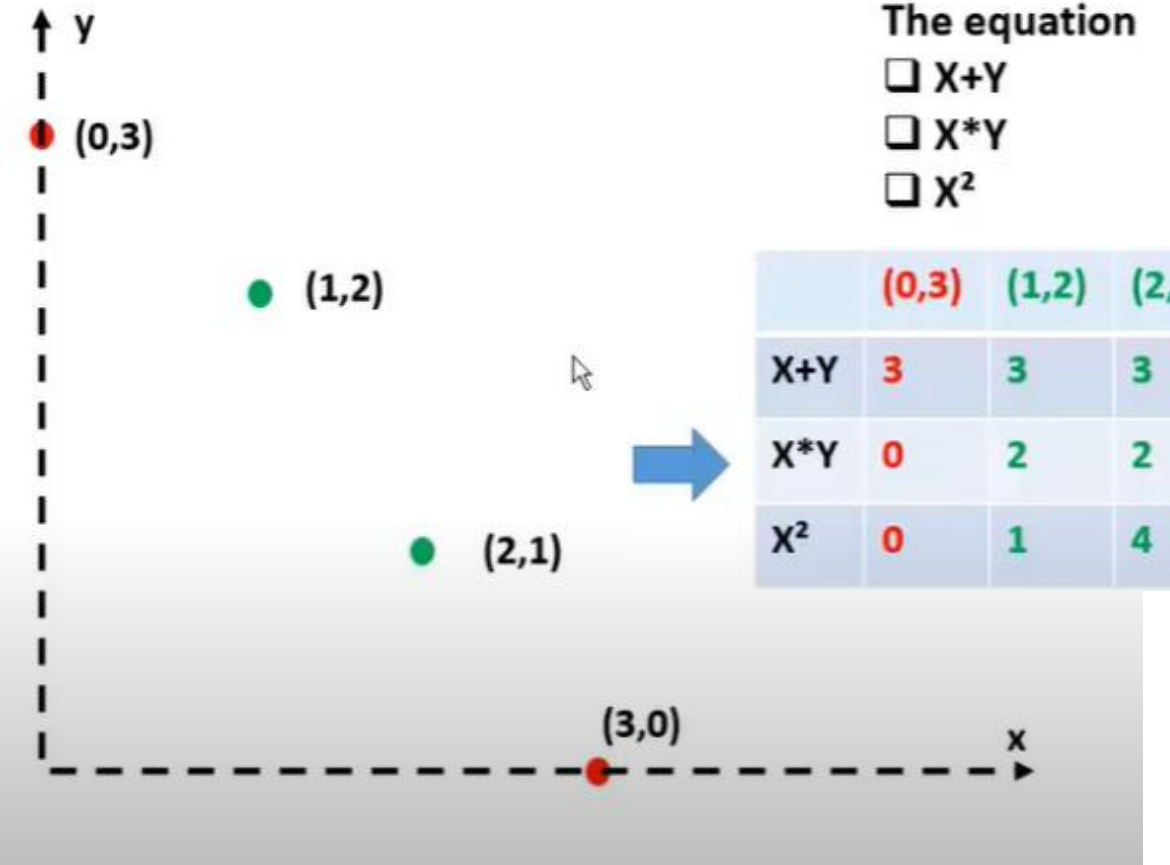
(x,y) → (x,y,xy)

(0,3) → (0,3,0)

(1,2) → (1,2,2)

(2,1) → (2,1,2)

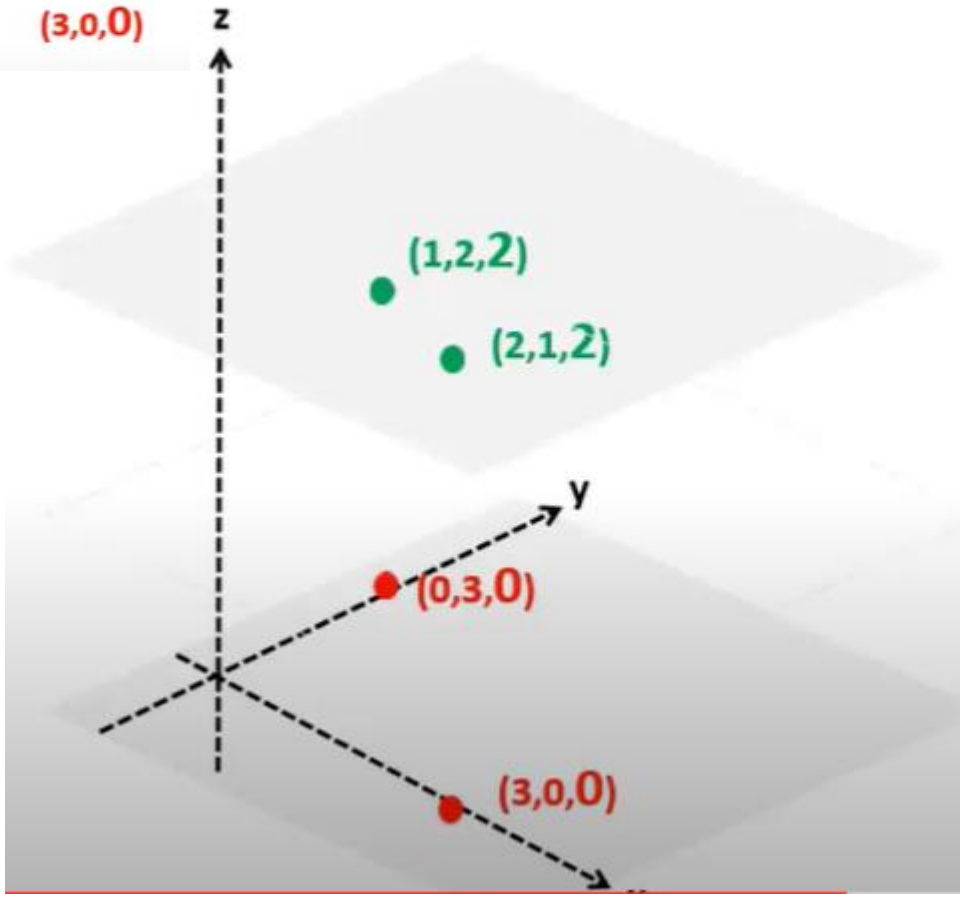
(3,0) → (3,0,0)



The equation

- ❑ X+Y
- ❑ X*Y
- ❑ X^2

	(0,3)	(1,2)	(2,1)	(3,0)
X+Y	3	3	3	3
X*Y	0	2	2	0
X^2	0	1	4	9



ROZHODOVACIE STROMY – ZÁKLADNÉ INFORMÁCIE

- Rozhodovací strom (RS) má veľkú výhodu vo svojej ilustratívnosti a zrozumiteľnosti aj pre laikov – zadávateľov riešenia
- RS reprezentuje rozhodovaciu procedúru vo forme acyklickej grafovej štruktúry - stromu
- Uzly stromu reprezentujú triedu (listové) alebo test (ostatné)
- Test má formu testovací atribút TA = hodnota
- Hrany reprezentujú hodnoty zvolených testovacích atribútov
- Použitie pri hľadaní výstupu pre nový príklad

  Novému príkladu vyhovujúcemu testom na ceste od koreňového po listový uzol je priradený výstup relevantný danému listovému uzlu (trieda alebo numerická hodnota)

- Ak **trieda** potom ide o **klasifikačný rozhodovací strom**

  - ID3, C4.5

- Ak **numerická hodnota** potom ide o **regresný rozhodovací strom**

  - CART

INDUKCIA KLASIFIKAČNÝCH ROZHODOVACÍCH STROMOV

- Algoritmy, ktoré generujú na výstupe Rozhodovací Strom (RS)pracujú väčšinou neinkrementálne
- Uplatňujú princíp „rozdeľuj a panuj“ resp. princíp „delenia priestoru príkladov na pod-priestory“
- Ukončovacia podmienka (UK)
  - Pre perfektnú klasifikáciu: každý pod-priestor obsahuje iba príklady jednej a tej istej triedy
  - Pre nie perfektnú klasifikáciu: každý pod-priestor obsahuje aspoň dané hraničné percento príkladov jednej triedy (napr. 90%) – generovanie RS nevykoľají prítomnosť zašumených dát

Všeobecný popis algoritmu na generovanie RS:

Ak je pre každý pod-priestor splnená UK → Potom KONIEC

Inak

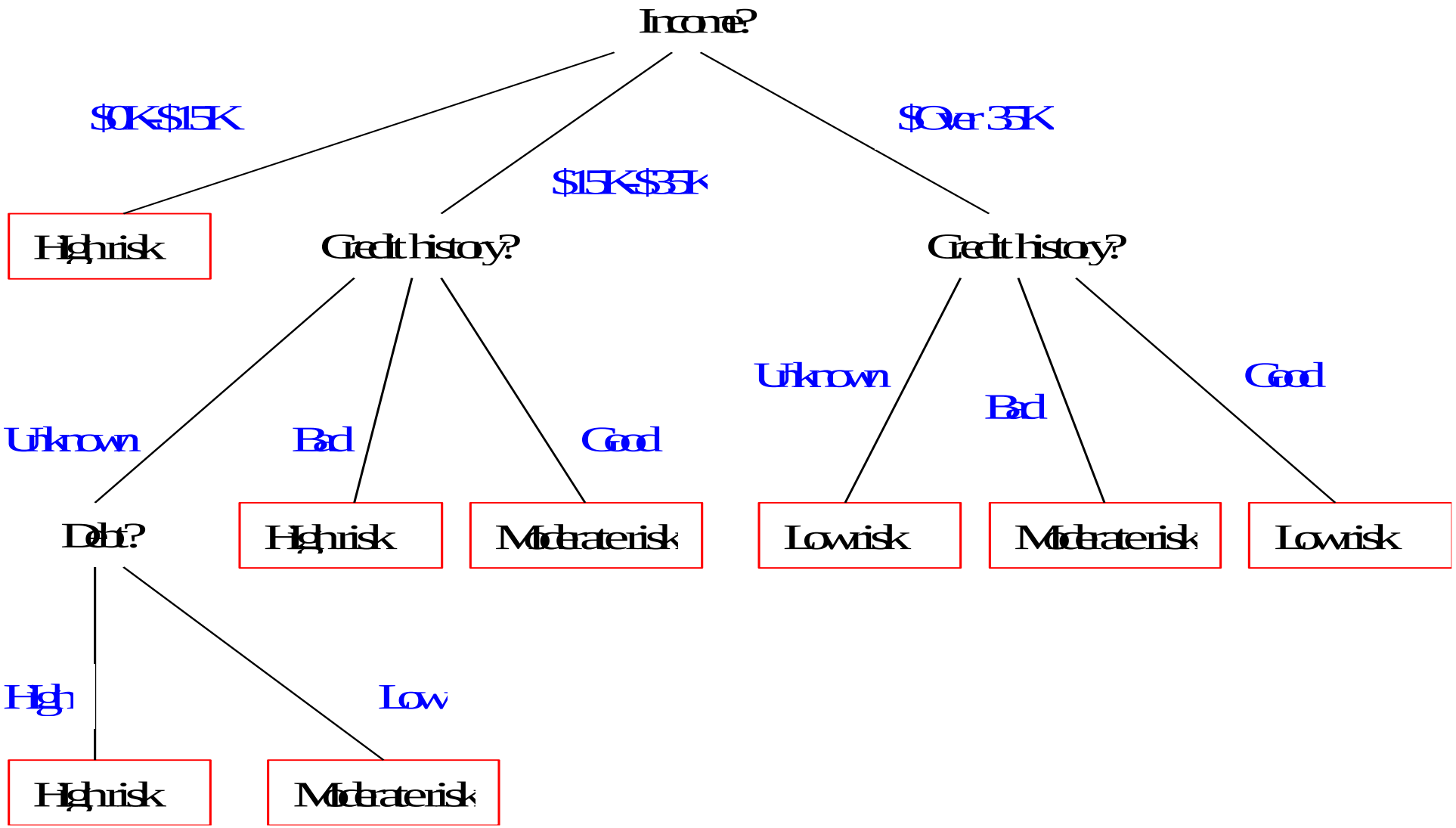
1. Zvoľ pod-priestor obsahujúci príklady rôznych tried
2. Zvoľ preň ešte nepoužitý testovací atribút (TA)
3. Rozdeľ ho na ďalšie pod-priestory podľa hodnôt TA

INDUKCIA KLASIFIKAČNÝCH RS – reálne dáta

No.	*Risk (Classification)*	*Credit History*	*Debt*	*Collateral*	*Income*
1	High	Bad	High	None	$0-15k
2	High	Unknown	High	None	$15-35k
3	Moderate	Unknown	Low	None	$15-35k
4	High	Unknown	Low	None	$0-15k
5	Low	Unknown	Low	None	Over $35k
6	Low	Unknown	Low	Adequate	Over $35k
7	High	Bad	Low	None	$0-15k
8	Moderate	Bad	Low	Adequate	Over $35k
9	Low	Good	Low	None	Over $35k
10	Low	Good	High	Adequate	Over $35k
11	High	Good	High	None	$0-15k
12	Moderate	Good	High	None	$15-35k
13	Low	Good	High	None	Over $35k
14	High	Bad	High	None	$15-35k

Collateral - záruka

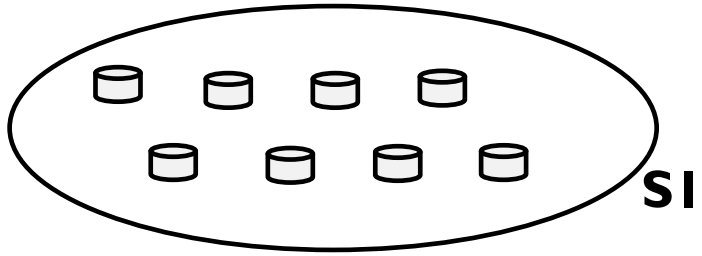
INDUKCIA KLASIFIKAČNÝCH RS – reálne dáta

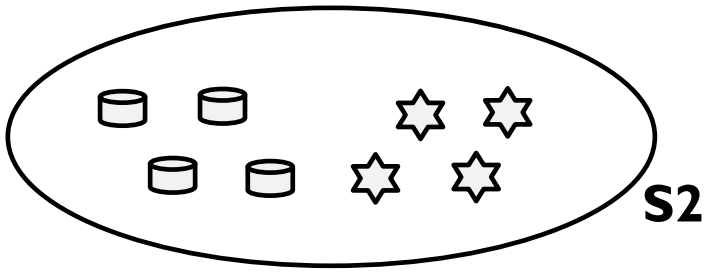


ALGORITMUS ID3 – KATEGORICKÉ DÁTA

- Ross Quinlan, 1979, je **neinkrementálny** algoritmus
- Generuje **klasifikačný model** - uskutočňuje **perfektnú klasifikáciu**, lebo algoritmus používa UK, kým pod-priestory neobsahujú iba TP jednej triedy
- Generuje model v tvare minimálneho stromu, lebo vyberá TA pomocou **Shannonovej teórie informácie**
- Základom je výpočet **entrópie**, teda **neurčitosti priestoru príkladov** – vzorec pre dve triedy: **H(S) ϵ <0,1>**

$$H(S) = -\sum_{j=1}^{2} \frac{n_j}{n_1 + n_2} \log_2 \frac{n_j}{n_1 + n_2}$$





$$H(S1) = -\frac{8}{8} * \log_2 \frac{8}{8} - \frac{0}{8} * \log_2 \frac{0}{8}$$

$$H(S1) = -1 * \log_2 1 - 0 = \mathbf{0}$$

$$H(S2) = -\frac{4}{8} * \log_2 \frac{4}{8} - \frac{4}{8} * \log_2 \frac{4}{8}$$

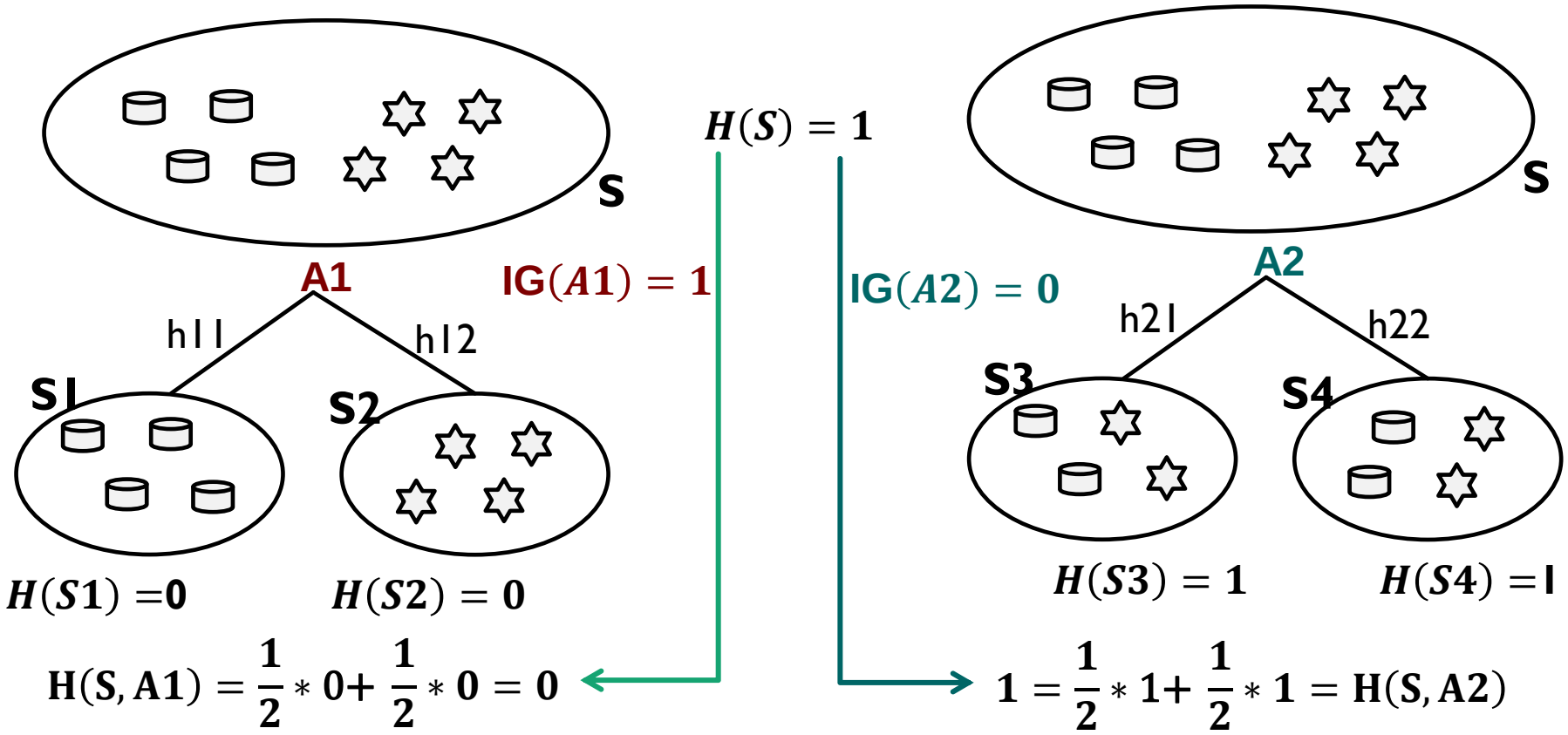
$$H(S2) = [-\frac{1}{2} * \log_2 \frac{1}{2}] * 2 = -\frac{1}{2}(-1) * 2 = \mathbf{1}$$

ALGORITMUS ID3

Shannonová teória informácie

Výpočet **zmeny entrópie**, po rozdelení priestoru príkladov podľa hodnôt TA, kde H je počet hodnôt atribútu – **H(S,A) je váhovaná entrópia dcérskych uzlov**

$$H(S,A)=\sum_{i=1}^{H} p(S_i)H(S_i) \qquad IG(A) = H(S) - H(S,A)$$



ALGORITMUS C4.5

- Ross Quinlan, 1993, **neinkrementálny** alg., generuje **klasifikačný model**
- Je modifikáciou ID3 v nasledovnom:
  - umožňuje spracovať numerické, reálne atribúty – **spojité atribúty**
  - nominálne atribúty označuje ako **diskrétne atribúty**
  - zavádza **pomerové kritérium zisku**
  - dokáže spracovať dáta s neznámymi chýbajúcimi hodnotami atribútov

- Na rozdiel od ID3 - C4.5 neuprednostňuje TA s väčším počtom hodnôt
- Normalizuje informačný zisk pomerovou entrópiou

$$IG_p(S,A) = \frac{IG(S,A)}{H_p(S,A)}$$

$$H_p(S,A) = -\sum_{j=1}^{m} p(a_j)\log_2(p(a_j))$$

- Pomerová entrópia:
  - narastá s rastúcim počtom vetiev → vyššia penalizácia (pôvodný informačný zisk IG je delený vyššou hodnotou)
  - vytvára novú prehľadávaciu preferenciu
  - vytvára rozmerovo menší RS (do šírky)

$$H_p(S,A1) = -2 * \left[\frac{1}{2}\log_2\frac{1}{2}\right] = 1$$

$$H_p(S,A2) = -8 * \left[\frac{1}{8}\log_2\frac{1}{8}\right] = 3$$

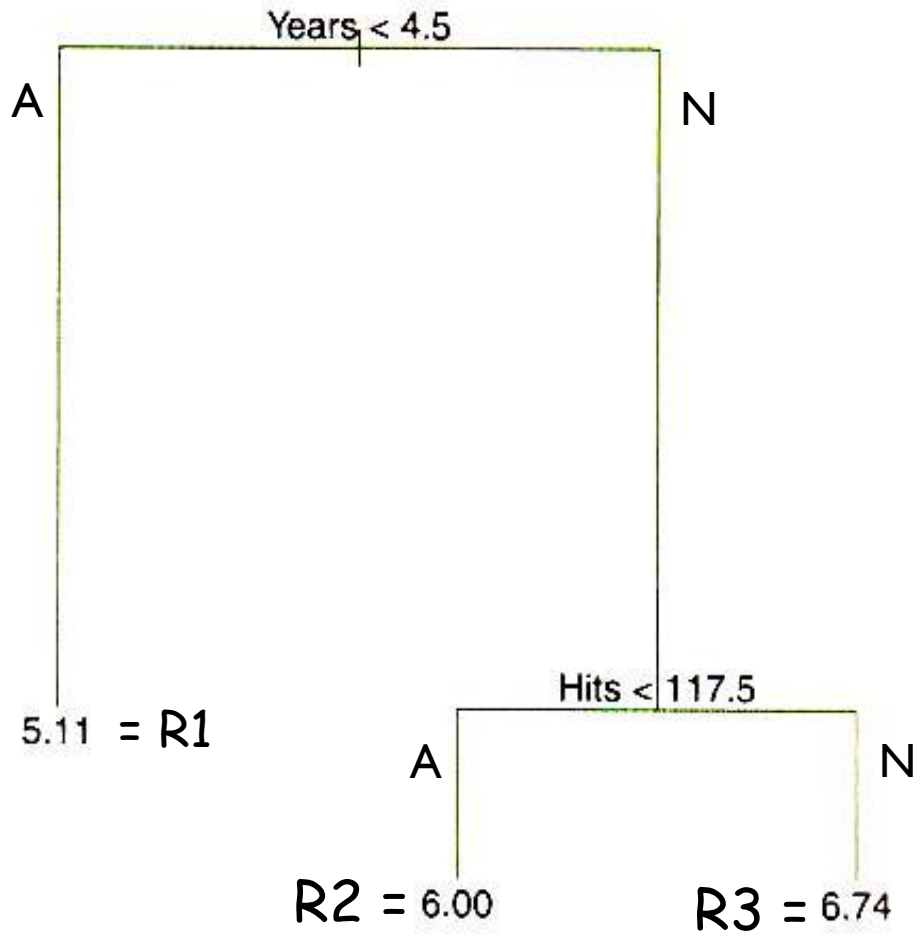
INDUKCIA REGRESNÝCH ROZHODOVACÍCH STROMOV

- Algoritmy na generovanie RS sa môžu aplikovať aj na regresnú úlohu. Potom generujú regresný strom
- Regresný RS tiež reprezentuje rozhodovaciu procedúru
- Aj tu sa uplatňuje princíp „rozdeľuj a panuj“ resp. princíp „delenia priestoru príkladov na pod-priestory“
- Aplikuje sa na **numerické dáta** (nie kategorické)
- **Segmentuje priestor príznakov, atribútov** (predictors space) na určitý počet jednoduchých oblastí
- Regresný strom predikuje numerickú hodnotu pre nové pozorovanie z ktorého sa neučil

Použitie:

Novému TP je predikovaná presná hodnota oblasti v priestore príznakov do ktorej TP patrí na základe jeho hodnôt atribútov

REGRESNÝ ROZHODOVACÍ STROM



Years ... počet rokov hry basketbalu

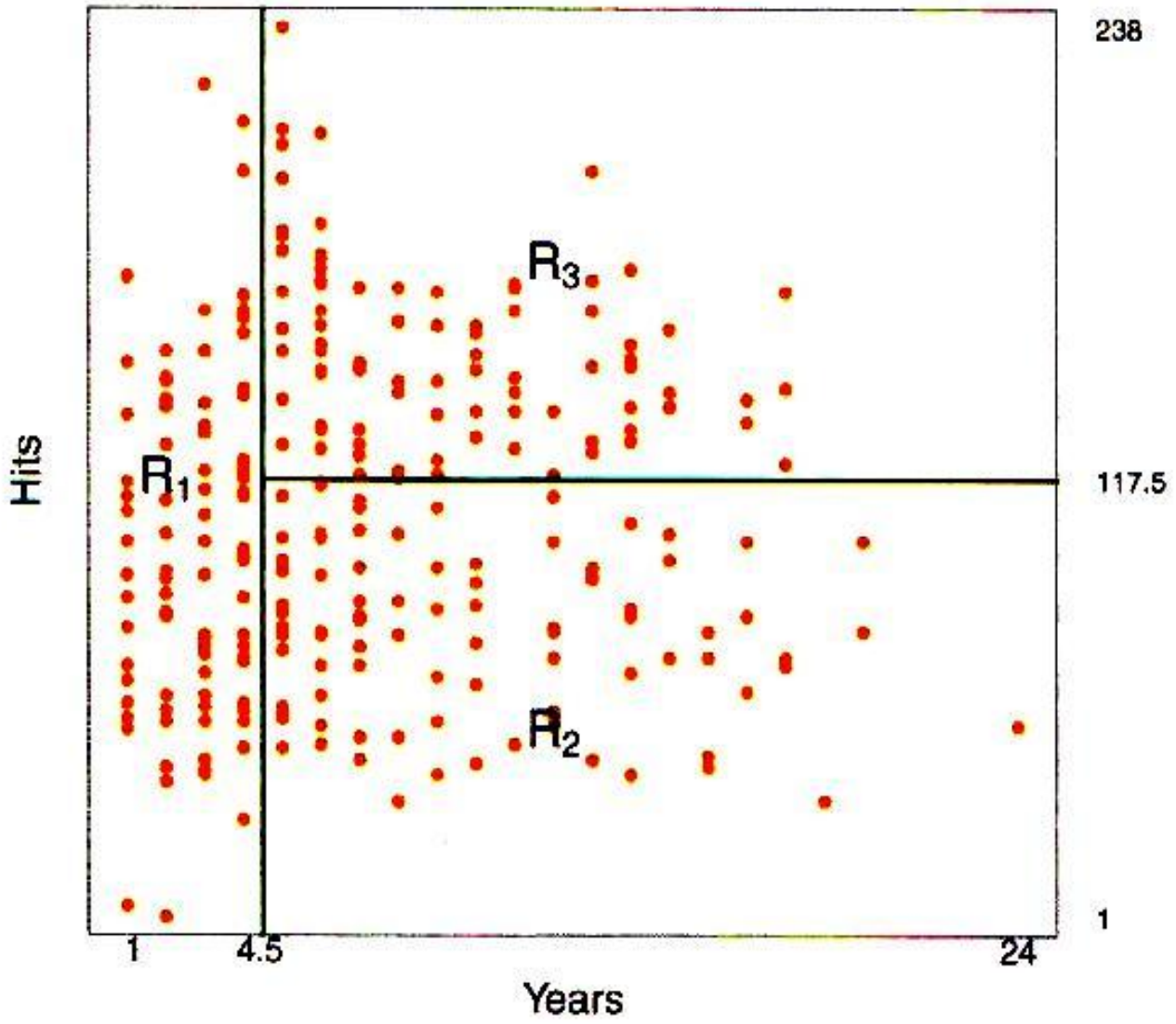
Hits ... počet košov v predchádzajúcom roku

Predikuje sa **hodnota platu**

Platové hladiny:

R1, R2 a R3

REGRESNÝ ROZHODOVACÍ STROM



Years ... počet rokov hry basketbalu

Hits ... počet košov v predchádzajúcom roku

Predikuje sa **hodnota platu**

Platové hladiny:

R1, R2 a R3

v tvare log Salary = 5.11

plat v dolároch je R1 = $e^{5.11}$ = 165,174

STROMOVÉ VERZUS LINEÁRNE MODELY

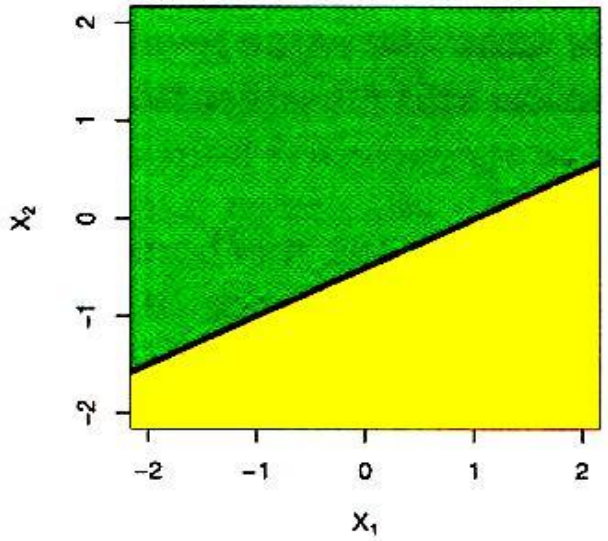
Vhodnosť použitia závisí na probléme teda na charaktere dát

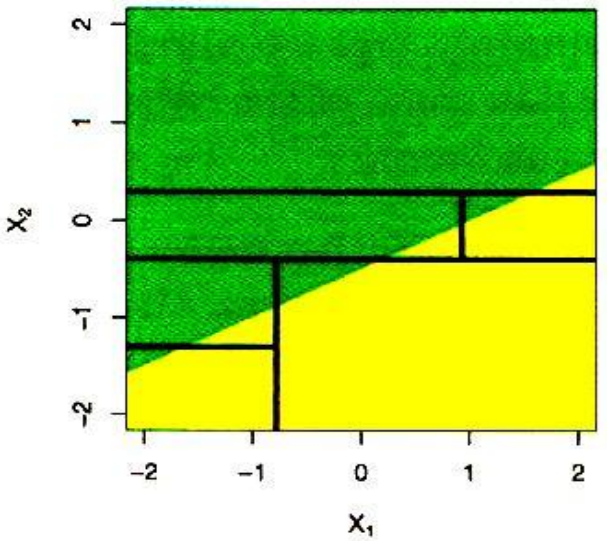
- Ak závislosť výstupu (odpovede, predikcie) na hodnotách atribútov (prediktorov) má lineárny charakter, teda máme lineárne dáta, potom sú vhodné lineárne modely:
  - Lineárna prahová jednotka,
  - Lineárne SVM
  - Lineárna regresia

- Ak je táto závislosť komplexná a vysoko nelineárna, teda máme nelineárne dáta, potom sú vhodné stromové modely:
  - Klasifikačný rozhodovací strom
  - Regresný strom
  - Náhodné lesy (Random Forests)

STROMY VERZUS LINEÁRNE MODELY

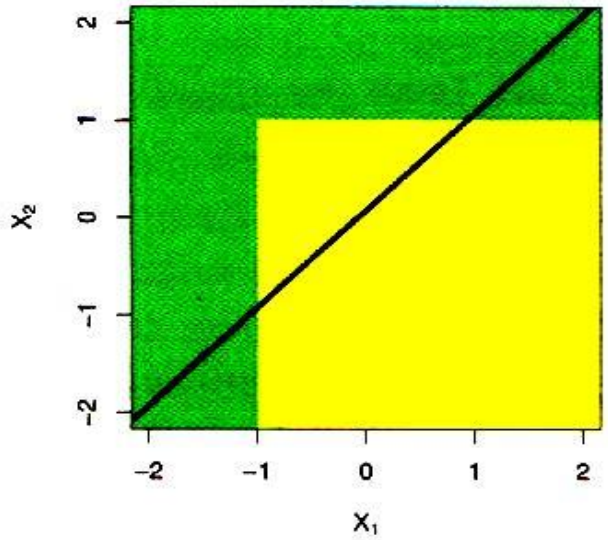
Lineárny model na lineárnych dátach

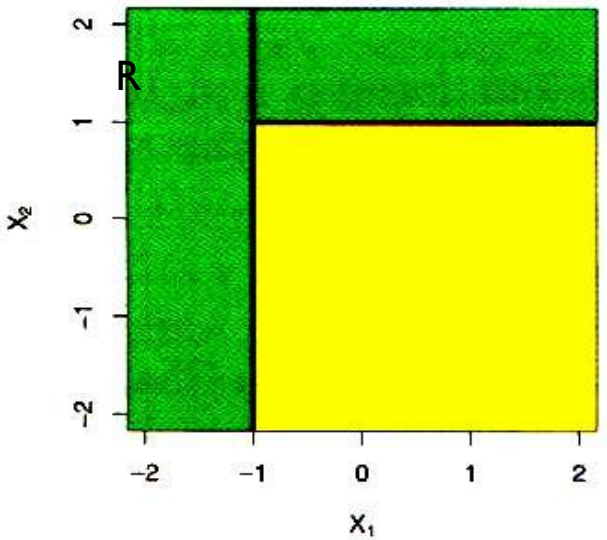




Nelineárny model (RS) na lineárnych dátach (vysoká chyba RSS (Residual Sum of Squares))

Lineárny Model na nelineárnych dátach (vysoká chyba RSS)



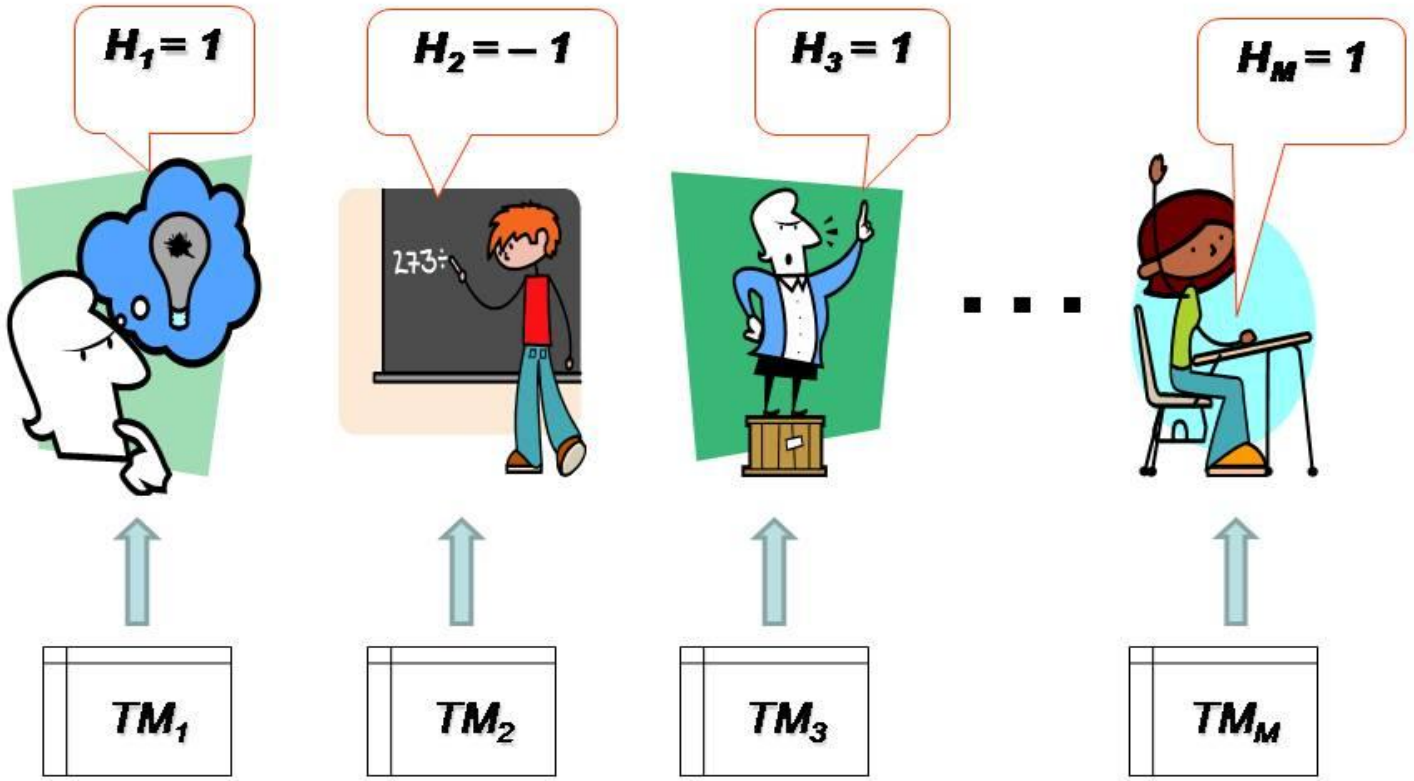


Nelineárny model na nelineárnych dátach

UČENIE SÚBOROM METÓD

- Učenie súborom metód rieši problém slabých klasifikátorov (dosahujú nízku efektívnosť – Presnosť, Návratnosť, F1, Acc., ...)
- Nízka efektívnosť je často zapríčinená malou alebo nekvalitnou TM.
- Preto sa tento prístup sústreďuje na formovanie rôznych výberov z pôvodnej TM.
- Nad každým výberom sa trénuje, učí slabý - partikulárny klasifikátor.
- Výsledok klasifikácie je určený hlasovaním všetkých slabých klasifikátorov. Preto učenie súborom metód - modelov.
- Pri regresnej úlohe sa výsledok určí ako priemer hodnôt výsledkov všetkých partikulárnych klasifikátorov.
- Najznámejšie prístupy:
  - Stacking (partikulárne klasifikátory - rôzne metódy SU),
  - Bagging (partikulárne klasifikátory – rovnaké metódy SU),
  - Boosting (partikulárne klasifikátory - rovnaké metódy SU),
  - Náhodné lesy (partikulárne klasifikátory – výhradne rozhodovacie stromy).

HLASOVANIE PATIKULÁRNYCH KLASIFIKÁTOROV



Pre klasifikáciu

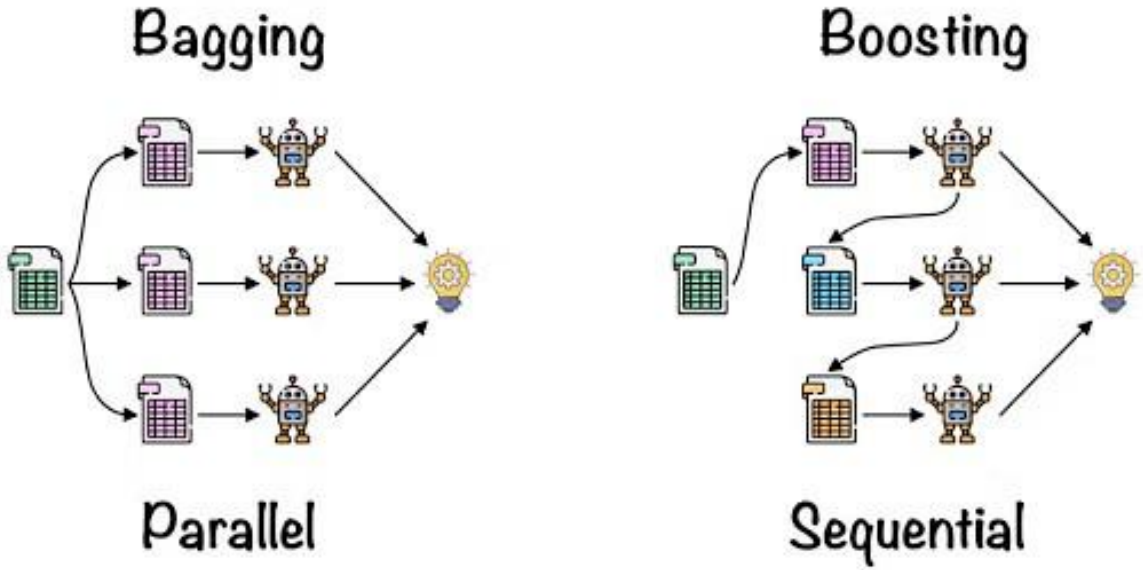
$$H(d_i, c_j) = sign\left(\sum_{m=1}^{M} \alpha_m H_m(d_i, c_j)\right)$$

Pre regresiu

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

POROVNANIE BAGGING A BOOSTING

- Zvyšujú presnosť klasifikácie slabých klasifikátorov (silný klasifikátor sa nepotrebuje radiť)
- Nevýhodou je strata jednoduchosti, prehľadnosti, ilustratívnosti a zvýšená výpočtová zložitosť.
- Dosiahnuté výsledky preferujú boosting pred baggingom.
- Riadenie chybou klasifikácie pri boostingu nielen spresňuje ale aj zrýchľuje proces učenia.
- Boosting – váhy nesprávne klasifikovaných príkladov sa zvyšujú
- Bagging – paralelné učenie
- Boosting – sekvenčné učenie

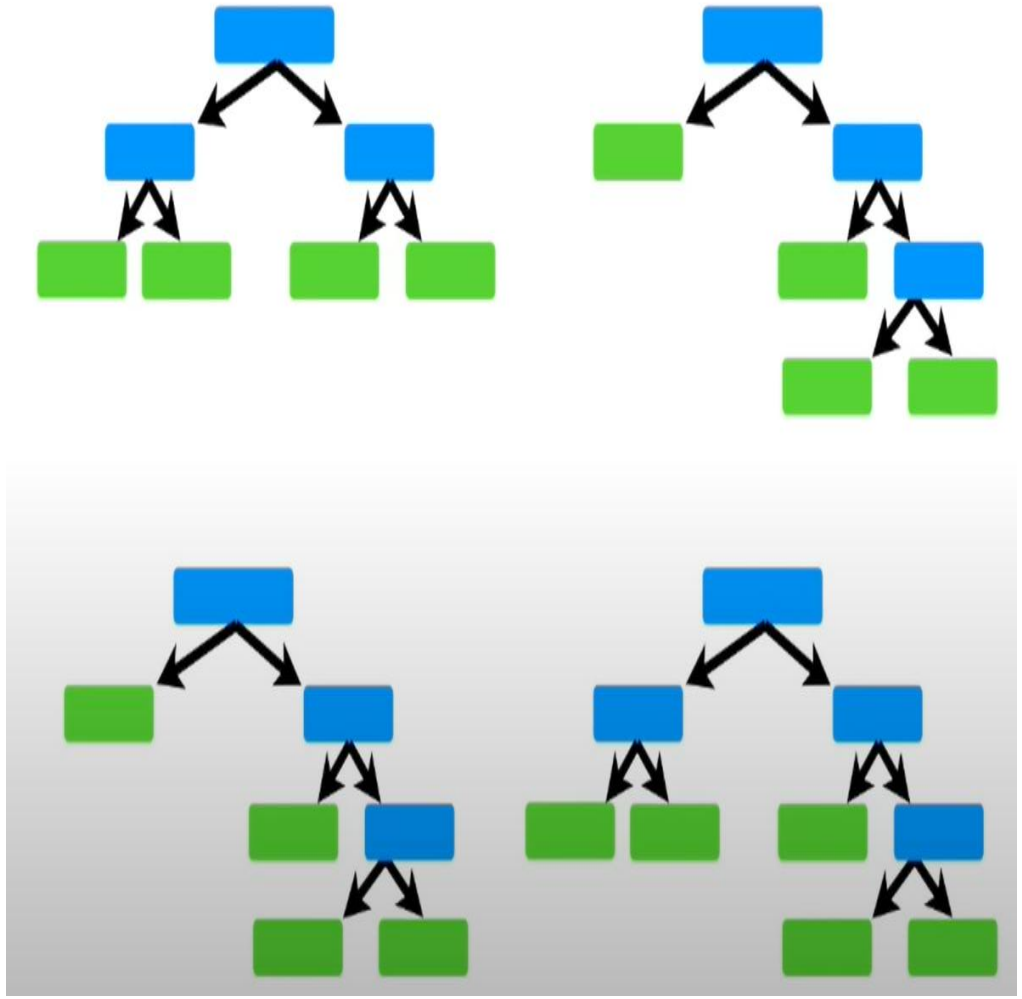


NÁHODNÉ LESY – RANDOM FORESTS

- Random Forests [Breiman, 2001] - používa výhradne rozhodovací strom.
- Je modifikácia baggingu - buduje súbor de-korelovaných stromov.
- Zložený klasifikátor skladá výsledky viacerých rozhodovacích stromov.
- Výsledok je získaný hlasovaním alebo spriemerňovaním.
- Jednotlivé stromové modely majú byť nezávislé, dekorelované – preto sa používa náhodný výber atribútov pre každý strom.
- Náhodný výber trénovacej množiny každého stromu umožňuje ho validovať na dátach, ktoré neboli vybraté na jeho trénovanie, čo uľahčuje validáciu.
- Keďže sú stromy nezávislé, je možné a výhodné ako aj jednoduché ich generovať paralelne.
- Potrebuje zadať niektoré parametre:
  - Počet rozhodovacích stromov v modeli
  - Počet náhodne zvolených atribútov v každom strome.
  - Pri výbere testovacieho atribútu sa berie do úvahy ***p*** atribútov z celkového počtu ***n***.

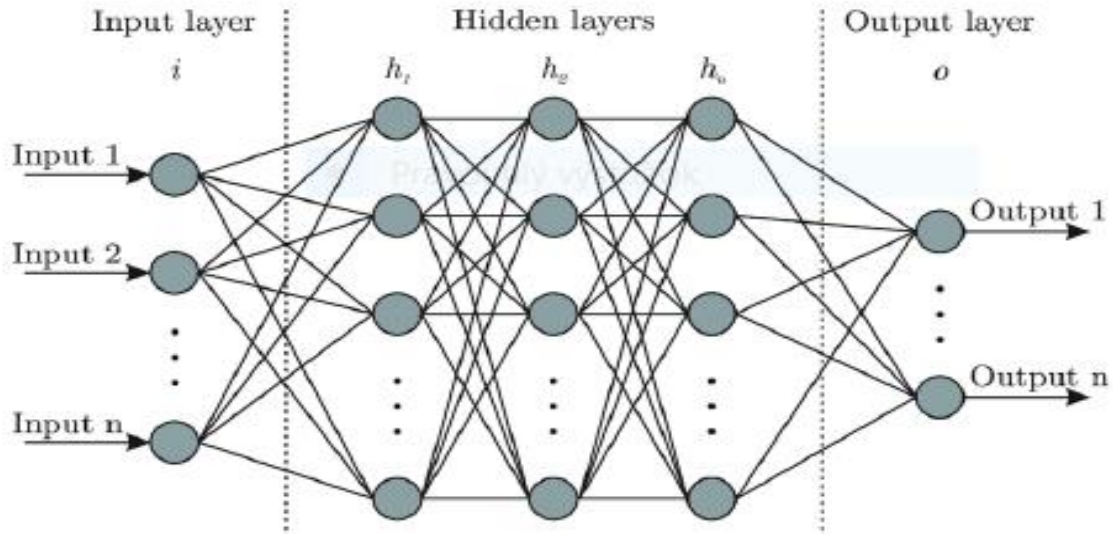
NÁHODNÉ LESY

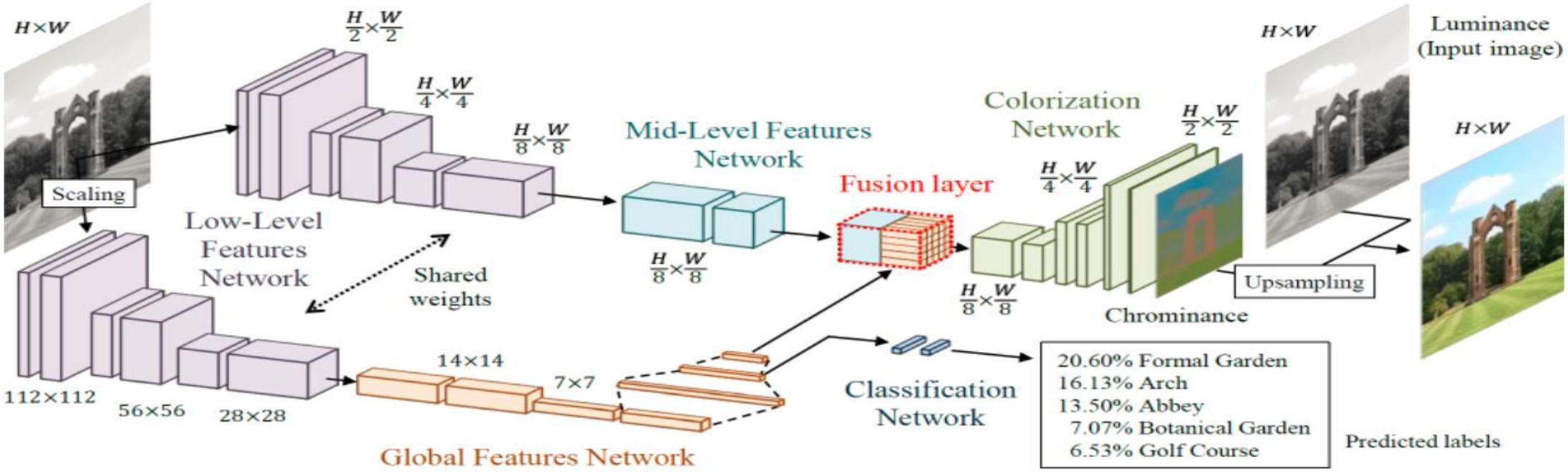
- Pri každom delení stromu nie je dovolené uvažovať väčšinu atribútov – prediktorov. Neracionálne?
- Predpokladajme jeden veľmi silný prediktor v rámci skupiny mierne silných prediktorov. Potom väčšina partikulárnych stromov ho použije ako testovací atribút.
- Jednotlivé partikulárne stromy by sa dosť podobali - silná korelácia stromov (nežiadúce).
- Spriemerňovanie lesa vysoko korelovaných stromov - neprinesie výhodu v porovnaní s použitím jedného stromu.
- Princíp de-korelácie stromov - pravdepodobnosť silného prediktora bude iba ***(n-p)/n*** a teda aj iné prediktory dostanú viac šance.



NEURÓNOVÉ SIETE –KLASIFIKAČNÝ PROBLÉM

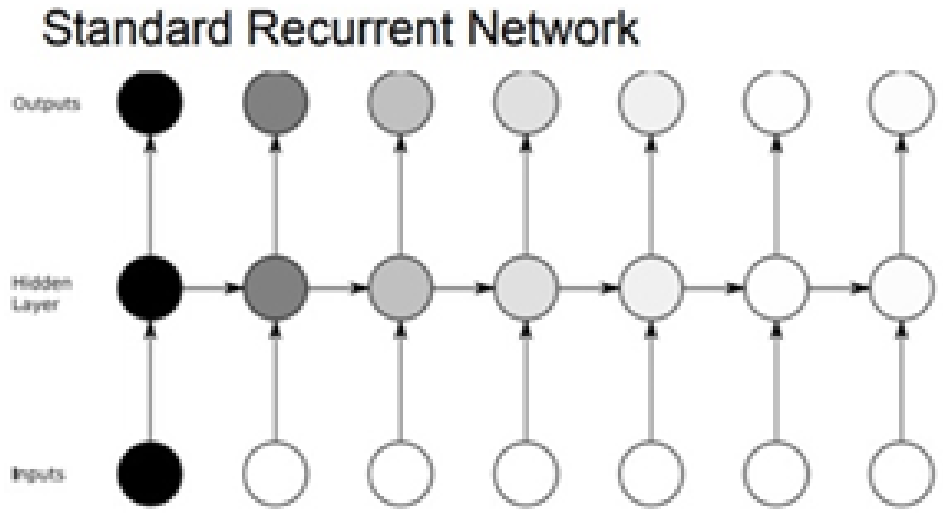
- Pre najbežnejšiu klasifikáciu do dvoch tried potrebujeme dva výstupy
- Vstupná vrstva, skrytá vrstva a výstupná vrstva s dvoma neurónmi
- Hlboké neurónové siete majú veľké množstvo skrytých vrstiev (aj 30)

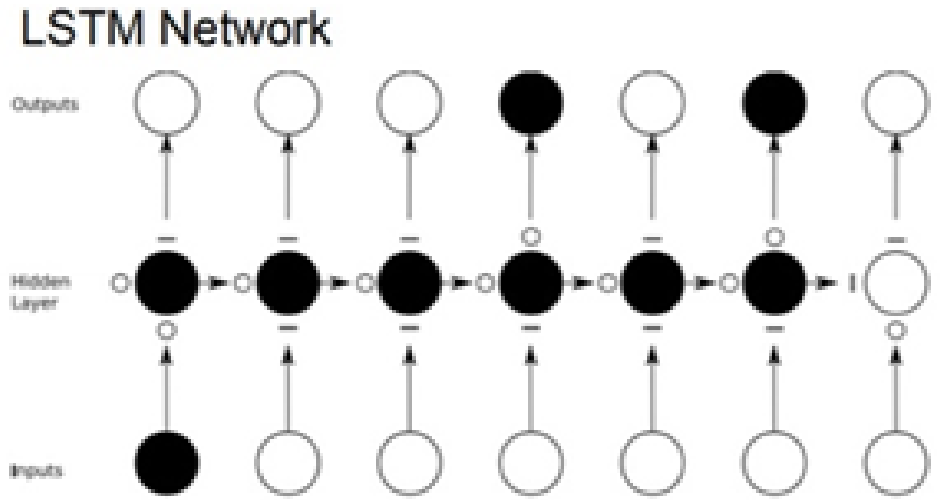




REKURENTNÉ NS – RIEŠENIE PROBLÉMU S PAMÄŤOU

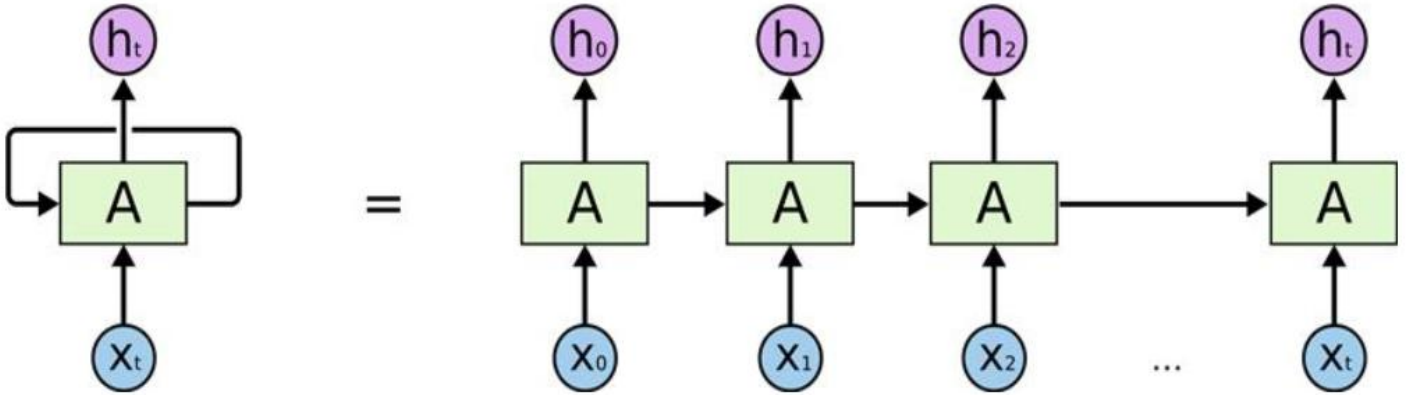
- LSTM(Long-short-term memory) - tok informácií regulujú 3 brány
- GRU (Gated recurrent unit) – zjednodušená LSTM – obsahuje iba 2 brány
- Brány sú schopné si zapamätať, ktoré informácie sú dostatočne dôležité na zachovanie
- LSTM a GRU siete sú bežne využívané pri spracovaní textov, rozpoznávaní reči, generácii textu alebo syntéze reči
- Obojsmerné modifikácie - BiLSTM (Bidirectional LSTM) a BiGRU (Bidirectional GRU)



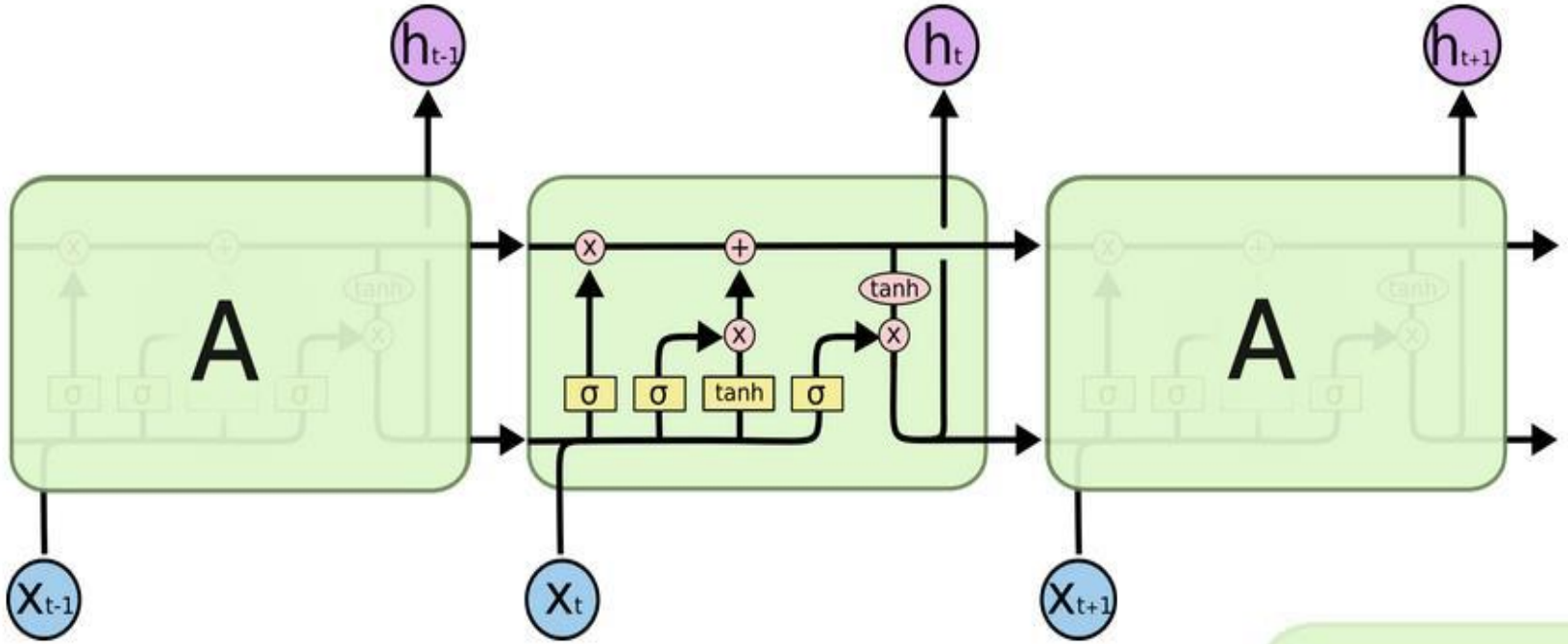


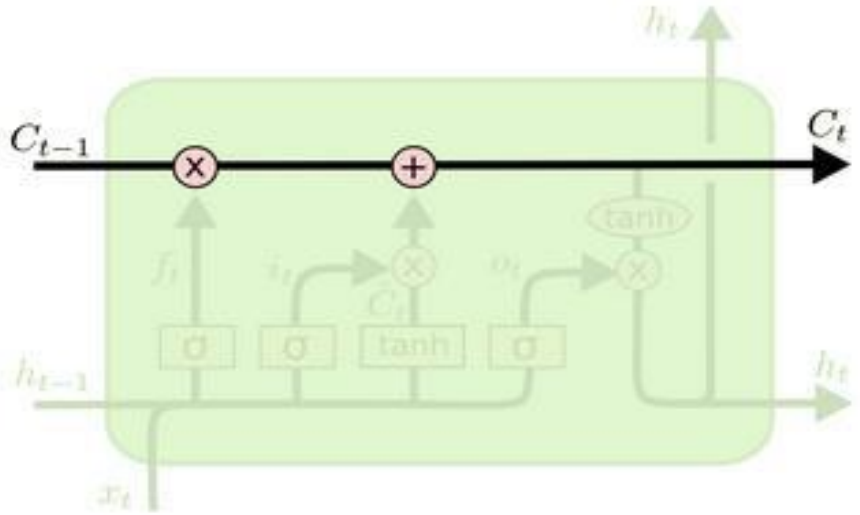
LSTM – MODELOVANIE VZŤAHOV MEDZI SLOVAMI

- V LSTM (Long Short Memory) každá bunka obsahuje bránu pre vstup (Input gate), výstup (Output gate) a zabudnutie (Forget gate) – naviac oproti bežným rekurentným sieťam
- Tieto brány využívajú sigmoidálnu funkciu
- Do tejto funkcie prídu informácie ohľadom minulého a súčasného stavu, následne funkcia vráti hodnotu medzi 0 a 1 (0 - zabudnutie, 1 – zapamätanie)
- Forget gate rozhoduje, co je dostatocne relevantné pre zachovanie z minulého kroku
- Input gate rozhoduje čo je relevantné zo súčasného kroku
- Output gate určuje ďalší skrytý stav, ktorý predstavuje pamäť NS



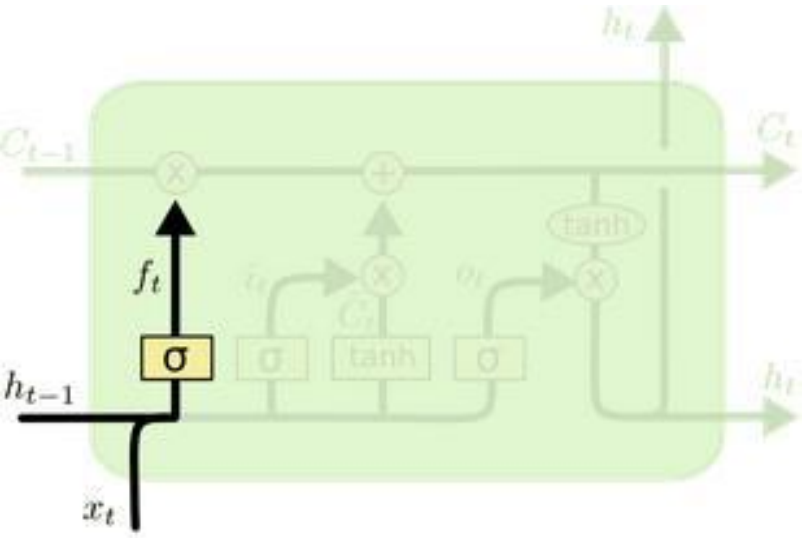
LSTM – SPRACOVANIE TEXTOVÝCH DÁT



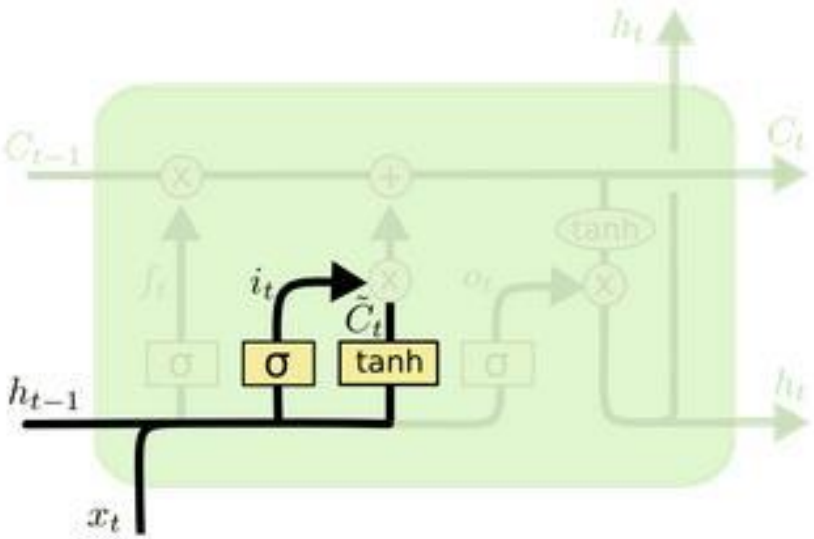


Základ LSTM tvorí horizontálna línia ktorou prechádza vektor medzi jednotlivými blokmi, kde dochádza iba k malým lineárnym operáciám. Pomocou tejto línie prechádza informácia cez celú štruktúru LSTM siete.

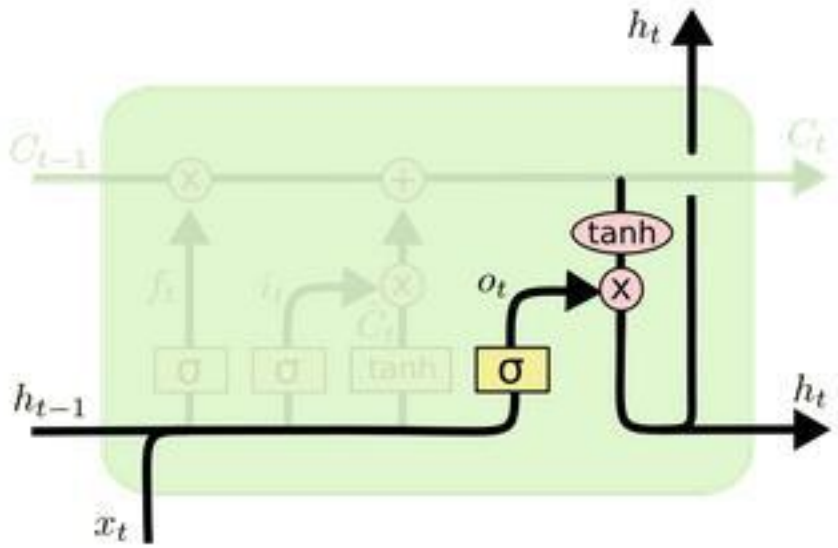
LSTM – BRÁNY



Prvá zabúdacia brána (forget gate) rozhoduje o tom, aká informácia sa má zabudnúť, respektíve aká jej časť



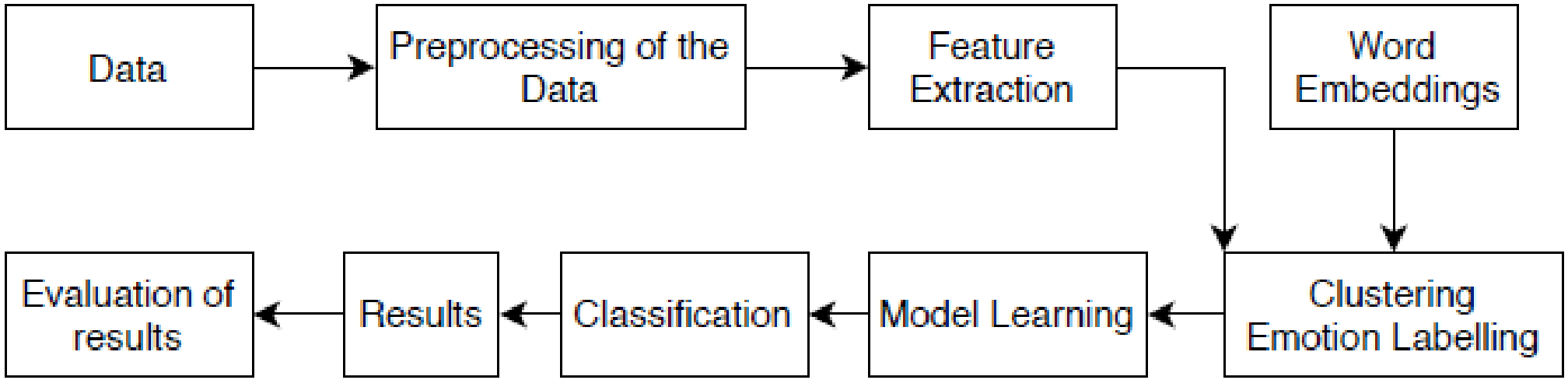
Druhá vstupná brána (ang. input gate) - rozhoduje o tom ktorá časť informácie bude aktualizovaná.
Druhý je neurón s aktivačnou funkciou hyperbolický tangens (vytvára kandidátov na novú hodnotu *Ct,*, o ktorú môže byť aktualizovaný stav bloku)



Tretia výstupná brána (*output gate*) rozhoduje o tom, aká časť informácie bude posunutá ďalej. Aktuálny stav prejde cez funkciu hyperbolický tangens, ktorú LSTM využíva ako aktivačnú funkciu. Potom je vynásobená táto hodnotu s hodnotou z výstupnej brány

ANALÝZA TEXTU – KLASIFIKÁCIA EMÓCIÍ

- Vektorová reprezentácia (jedna veta – jeden vektor, TF-IDF)
  - Klasické strojové učenie (SVM, LR, Naive Byes)
  - Učenie súborom metód (RF, XGBoost, Bagging)
- Embedding (jedno slovo – jeden vektor, Word2Vec, GloVe, FastText)
  - Rekurentné neurónové siete (LSTM, GRU)
  - Konvolučné neurónové siete (CNN)
  - Transformers (mT5, byT5) + GPT



PREDSPRACOVANIE TEXTU

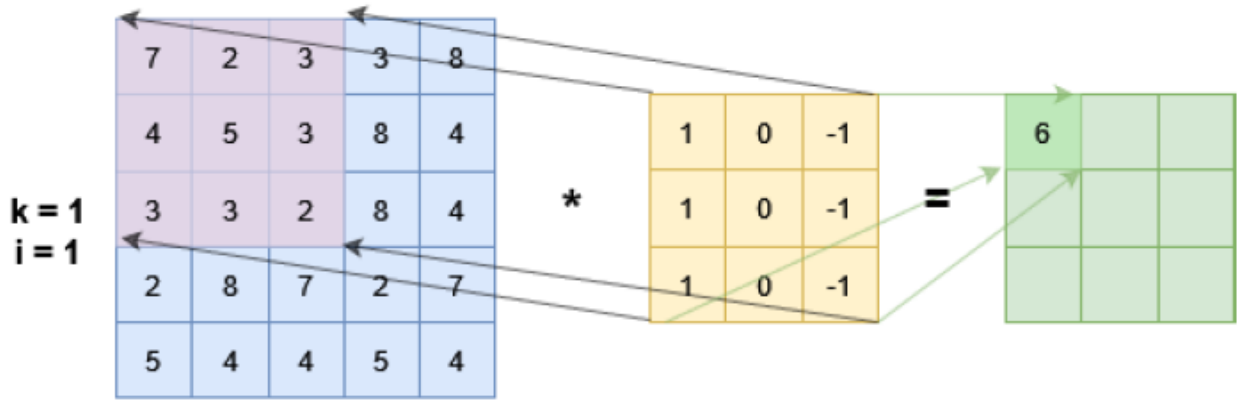
Embedding W2V – text transformujeme na vektory pre použitie na trénovanie NN

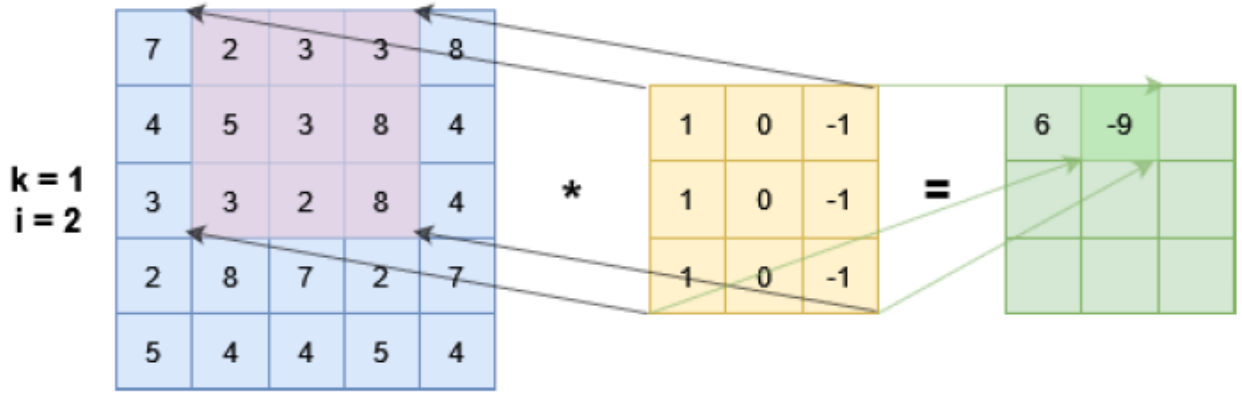
I	Really	Like	Very	Dark	Coffee

I	0,1	0,6	-0,3	0,3
Really	0,2	0,4	-0,2	0,8
Like	0,3	-0,7	0,1	0,1
Very	0,4	0,8	0,2	0,5
Dark	-0,2	0,9	0,2	-0,3
Coffee	0,4	0,5	-0,1	-0,2

I	0,1	0,6	-0,3	0,3
Really	0,2	0,4	-0,2	0,8
Like	0,3	-0,7	0,1	0,1
Very	0,4	0,8	0,2	0,5
Dark	-0,2	0,9	0,2	-0,3
Coffee	0,4	0,5	-0,1	-0,2

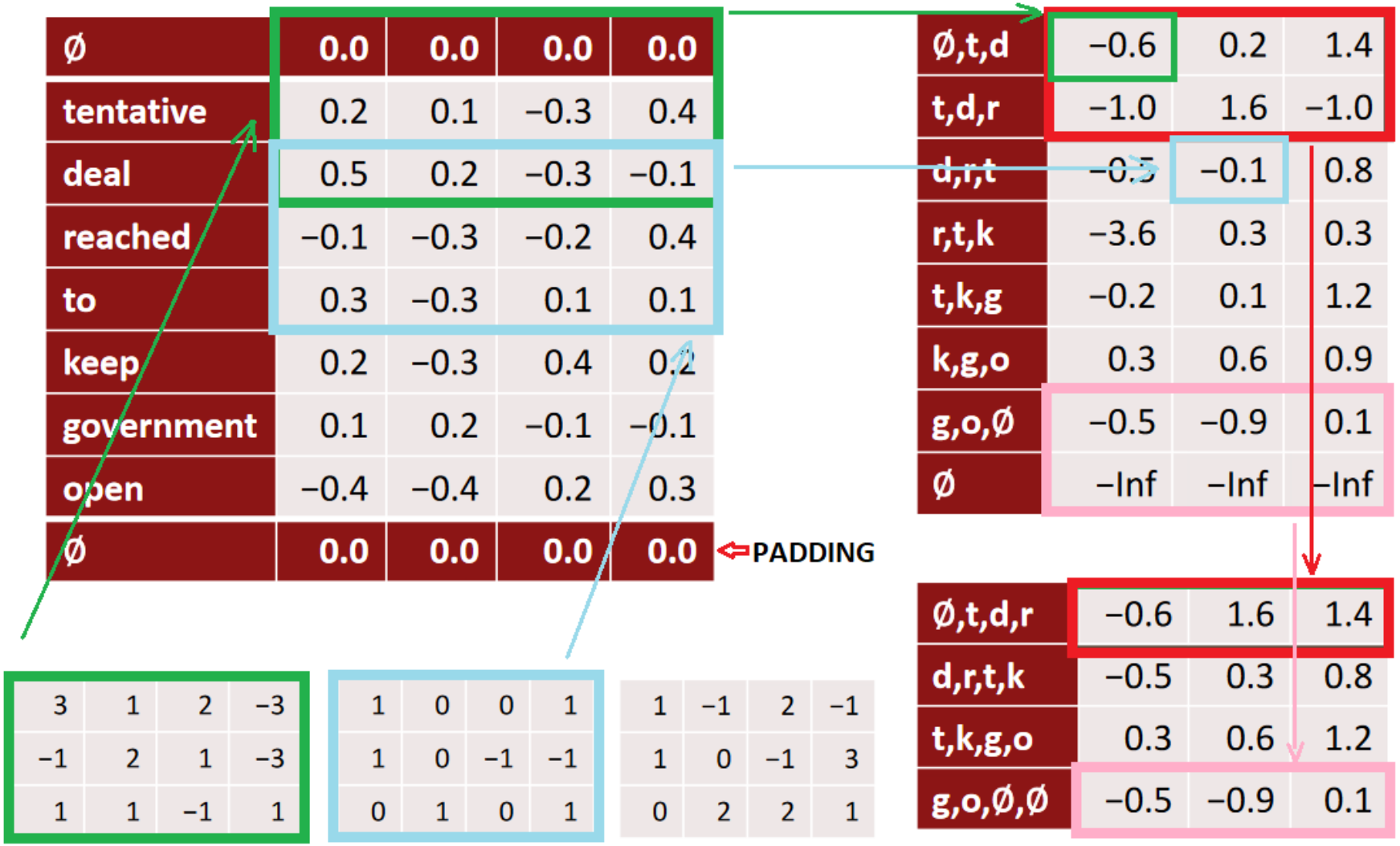
Podobne ako pri obrazových dátach - konvolúcia





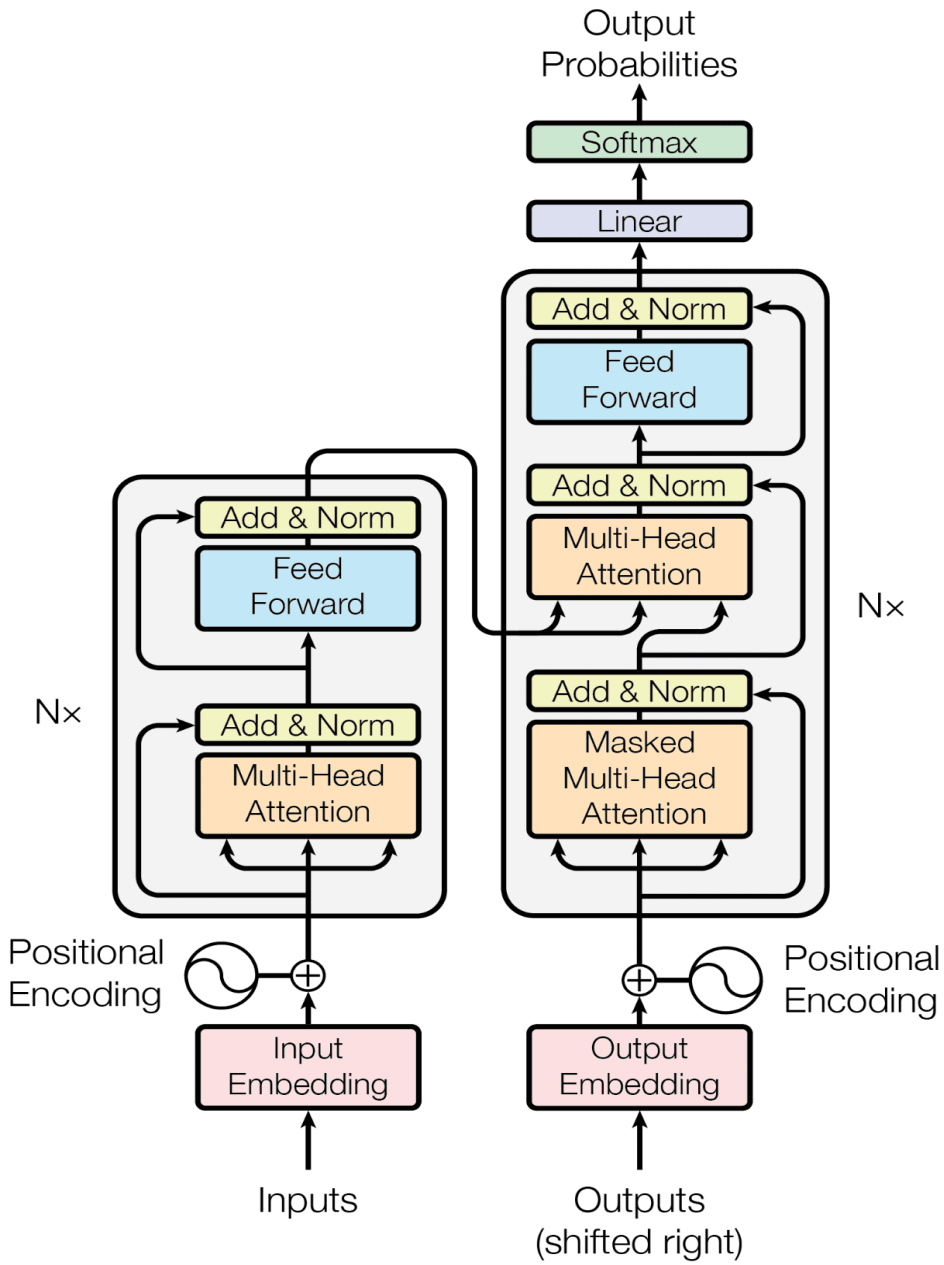
EMBEDDINGS Z TEXTU

1D convolúcia – tri filtre a max pooling – združovacia vrstva založené na funkcii max



TRANSFORMERY

- Výrazný posun v oblasti NLP (Natural Language Processing)
- Inovácia spočíva vo využití mechanizmu pozornosti, ktorý umožňuje súčasné zohľadnenie všetkých pozícií v rámci sekvencie vstupu a výstupu a efektívne zachytenie závislostí s dlhým dosahom
- Maskovaná vrstva „Attention“ ďalej zvyšuje schopnosť modelu rozoznávať rozmanité vzťahy vo vstupných údajoch



TRANSFORMERY

Modely s architektúrou Transformers - mBERT, SlovakBERT, RoBERTa, DistilBERT, mT5, byT5 a GPT

- Model Transformer využíva architektúru kodéra a dekodéra, ktorá obsahuje viacero vrstiev mechanizmu vlastnej pozornosti a vrstvy Feed Forward v oboch komponentoch
- Mechanizmus pozornosti (z angl. Attention Mechanism, AM) je transformačný koncept, ktorý umožňuje modelu efektívnejšie spracovať zložité, dlhé vstupné vety
- Mechanizmus pozornosti zlepšuje proces spracovania textu tým, že dynamicky upravuje zameranie modelu na každé cieľové slovo
- Priraďuje skrytým stavom kodéra rôzne váhy - sa používajú na výpočet kontextového vektora (váženého súčtu skrytých stavov), ktorý obsahuje relevantné informácie zo zdrojovej vety pre aktuálne cieľové slovo
- Tento kontextový vektor sa potom zlúči so vstupom dekodéra, aby pomohol predpovedať nasledujúce slovo v cieľovej sekvencii.
- Počas procesu trénovania sa model učí upravovať tieto váhy pozornosti

TRANSFORMERY - BERT

- *BERT (z angl. Bidirectional Encoder Representations from Transformers)* je algoritmus od Google AI Language, ktorý priniesol prevratné výsledky v oblasti NLP.
- BERT využíva obojsmerné trénovanie populárnej architektúry Transformer, čo mu umožňuje chápať kontext slov na základe celého okolia.
- Hlavná inovácia BERT spočíva v technike jazykového modelu s maskovaním slov (z angl. Masked Language Model, MLM), kde 15% slov v trénovacích sekvenciách je nahradených maskami.
- Používa stratégiu predikcie ďalšej vety (z angl. Next Sentence Prediction, NSP), kde sa model učí rozpoznávať, či dve vety nasledujú po sebe.
- BERT používa iba kodér mechanizmu Transformer, ktorý číta celú postupnosť naraz, čím získava obojsmerný kontext.
- Pri trénovaní BERT sa kombinuje MLM a NSP s cieľom minimalizovať kombinovanú funkciu straty oboch stratégií.
- Ak model dokáže predpovedať ďalšie slovo ktoré nasleduje vo vete, potom môže zovšeobecniť syntaktické a sémantické pravidlá jazyka.
- BERT bol vyvinutý na prácu so stratégiou veľmi podobnou GPT (z angl. Generative Pre-trained Transformer)

ĎAKUJEM ZA POZORNOSŤ