Rozpoznávanie obrazcov
šk.r. 2019-20

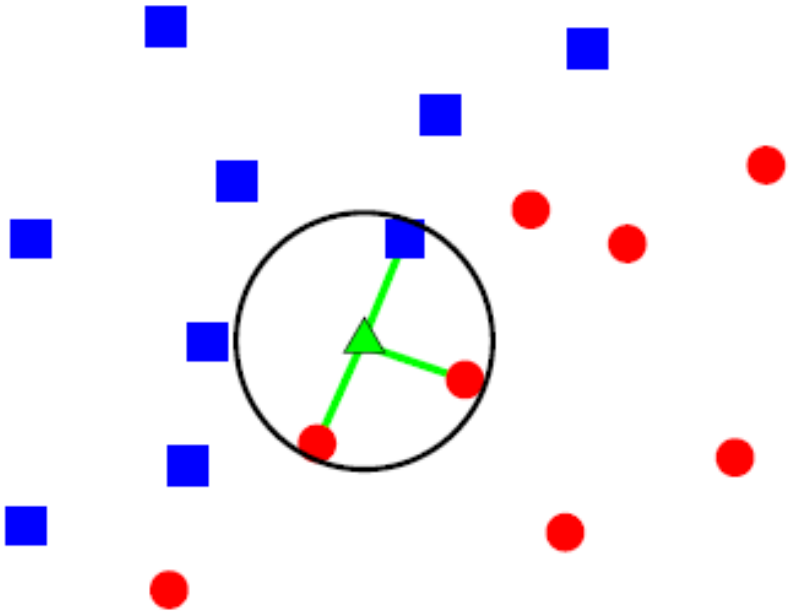
kNN a rozhodovacie stromy

Doc. RNDr. Milan Ftáčnik, CSc.

K najbližších susedov

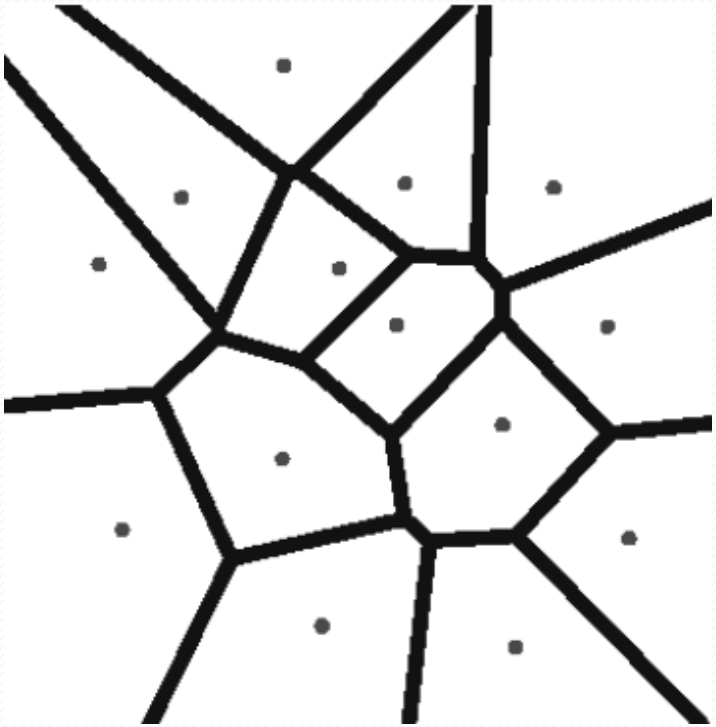
- Algoritmus:
  - Pre každú testovanú vzorku $\boldsymbol{x}$, nájdi *K* najbližších susedov z trénovacej množiny
  - Klasifikuj $\boldsymbol{x}$ podľa príslušnosti väčšiny z týchto susedov

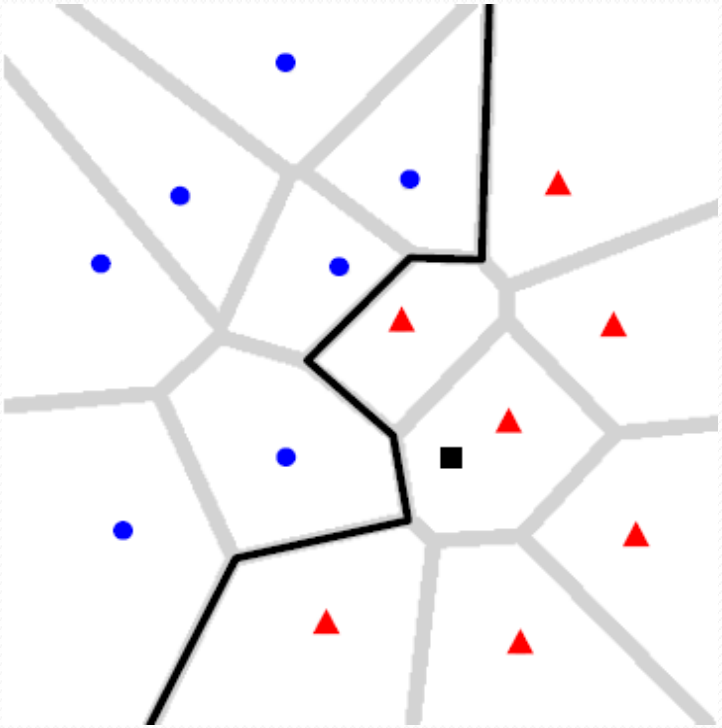
$K = 3$



K najbližších susedov II

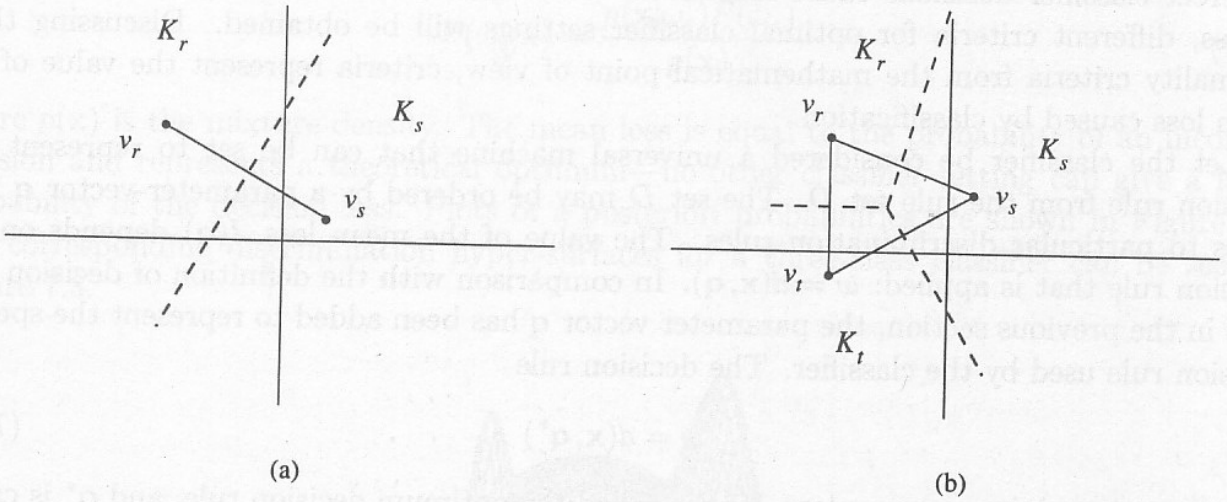
- Pre *K* = 1 dostaneme Voronoiov diagram





Špecifický prípad najbližšieho suseda

- Ak vyberieme za každú z *R* tried jedného typického predstaviteľa (etalón) a *K* = 1, tak dostaneme lineárny klasifikátor
- Etalóny sa obvykle vyberajú ako centroidy známych vektorov z tried



K najbližších susedov III

- Rozhodovacie pravidlo:

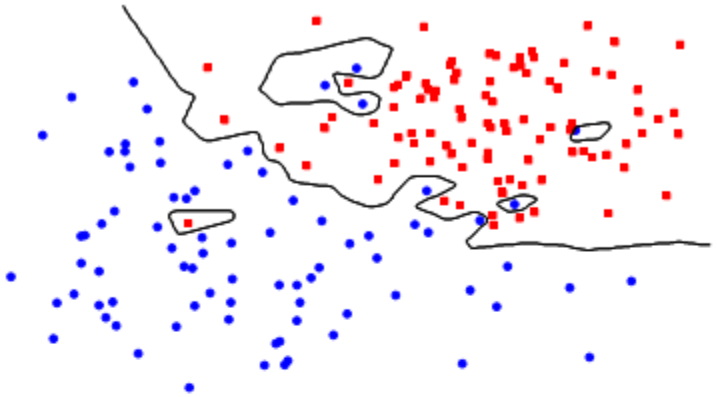
$$d(\boldsymbol{x}) = \omega_i \leftrightarrow \text{ak väčšina z } K \text{ najbližších susedov } \boldsymbol{x} \text{ patrí do } \omega_i$$

- Aká je zložitosť klasifikátora? Trénovanie je $O(1)$, klasifikácia je $O(N)$
- To je zle: chceme klasifikátor, ktorý je **rýchly** pri klasifikácii, ak je **pomalý** pri trénovaní, tak to nevadí

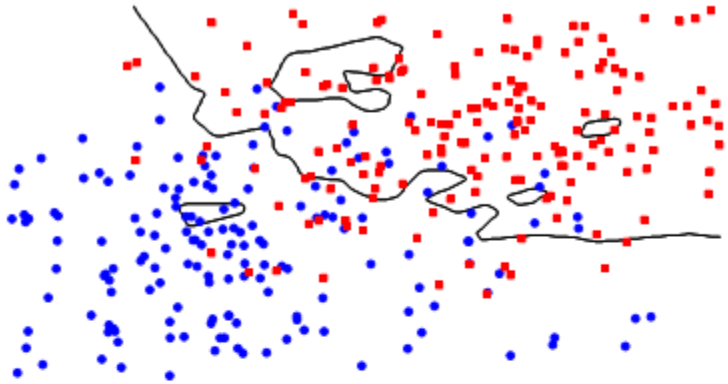
K najbližších susedov IV

- Ide o neparametrickú metódu, ktorá nemá žiadne vnútorné parametre, čiže nemá ani fázu učenia klasifikátora
- Hyperparametrami sú počet najbližších susedov K a metrika, ktorá meria vzdialenosť
- Ako máme správne vybrať hyperparametre: nestačí, že fungujú na trénovacej množine, validujeme ich na validačnej množine

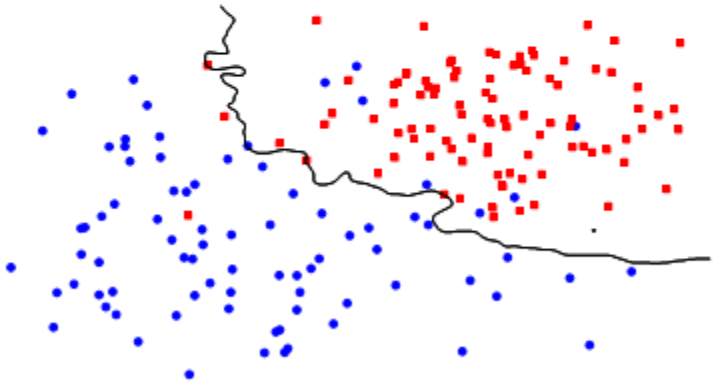
K najbližších susedov V



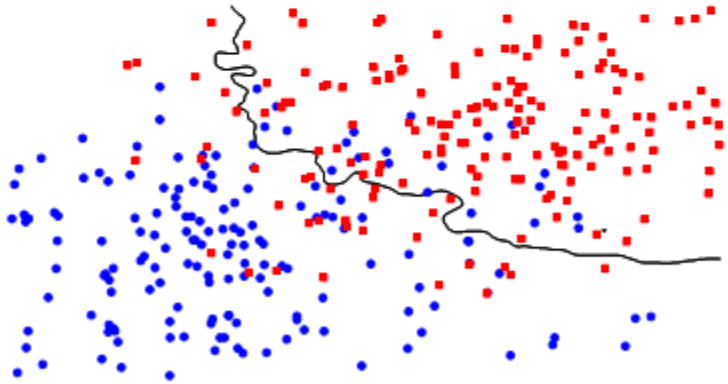
(a) $K = 1$, tréning, dosiahnutá chyba 0



(b) $K = 1$, testovanie, dosiahnutá chyba 0,1667

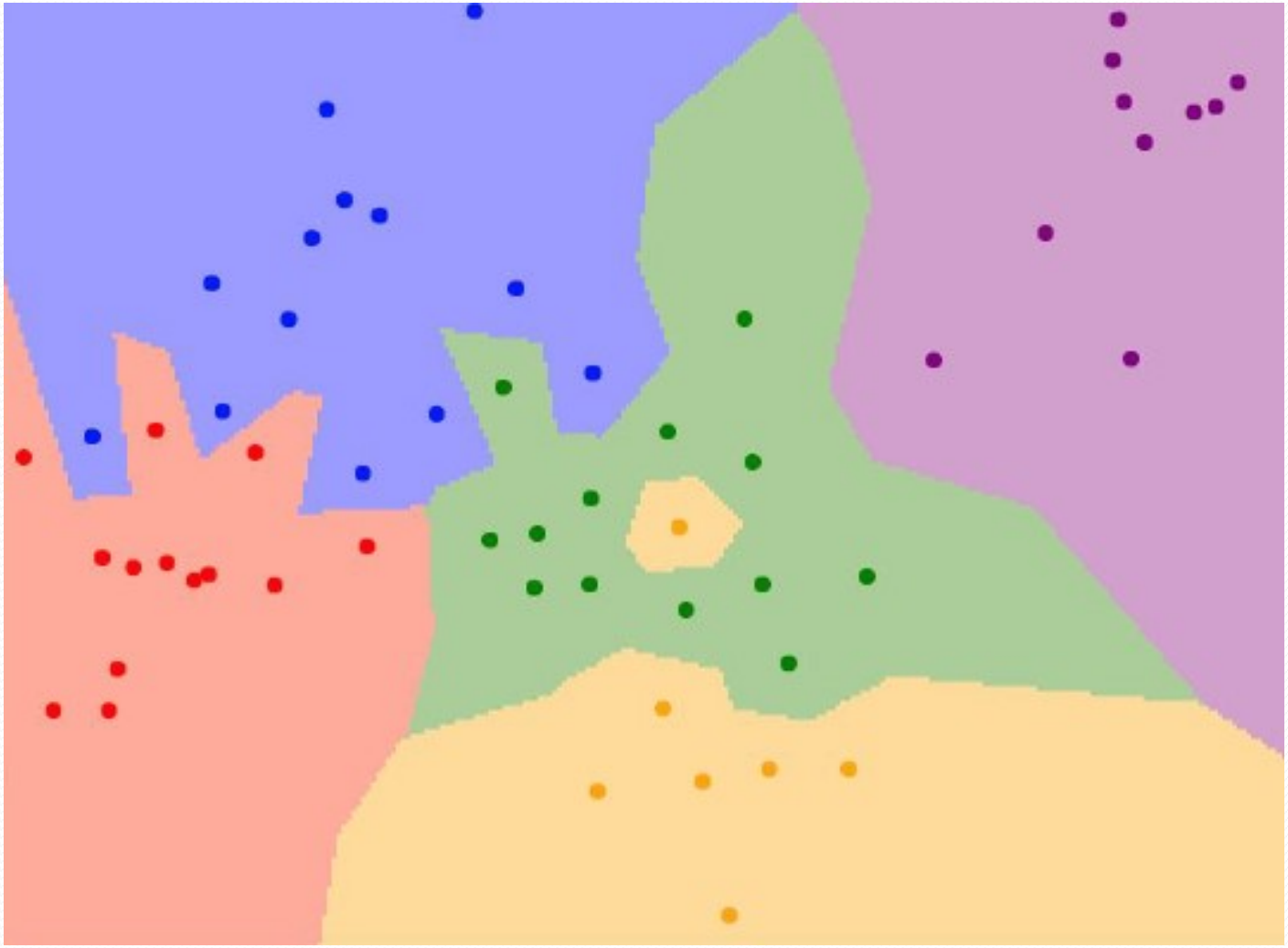


(c) $K = 7$, tréning, dosiahnutá chyba 0,0565



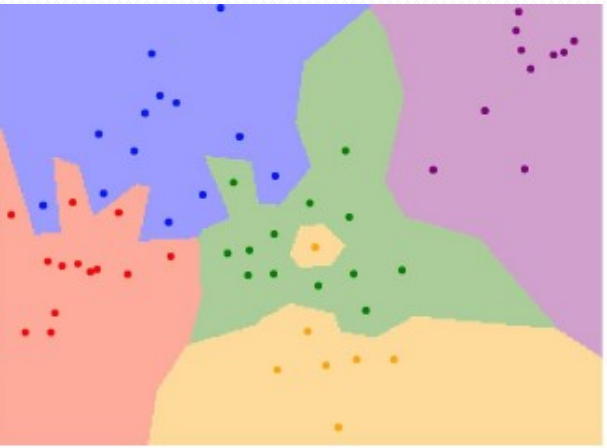
(d) $K = 7$, testovanie, dosiahnutá chyba 0,1637

K najbližších susedov pre *K* = 1

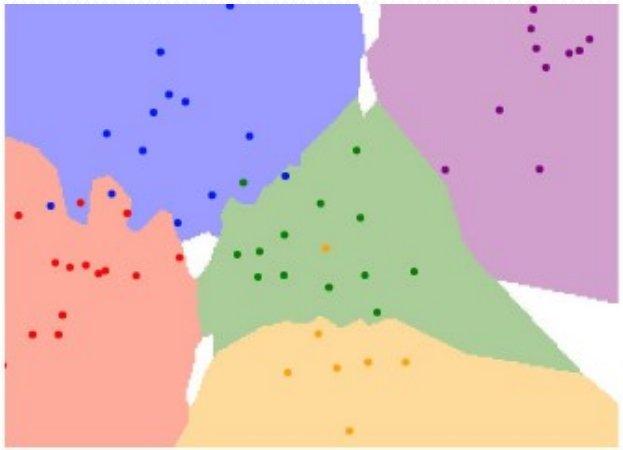


K najbližších susedov VI

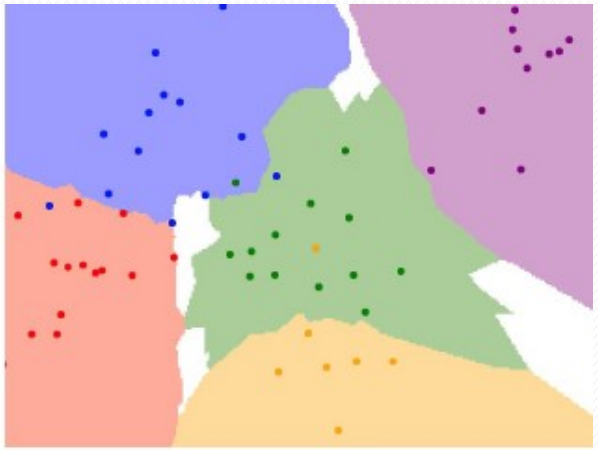
- Pri zvyšovaní *K* výsledok klasifikácie určí **väčšinové hlasovanie** z *K* najbližších bodov (biele plochy, reprezentujú remízu)



K = 1



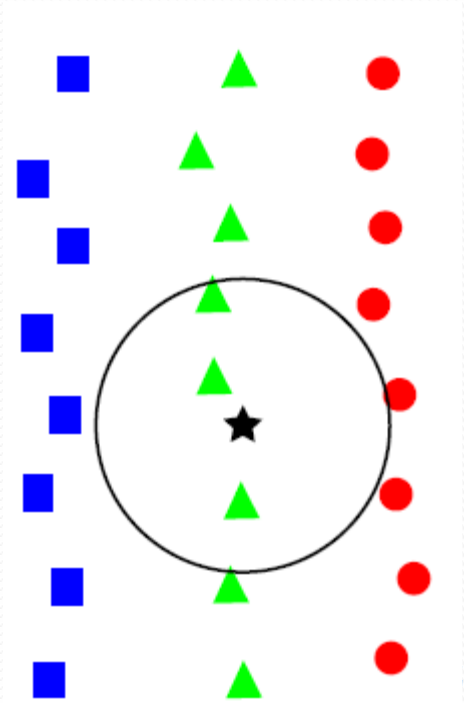
K = 3

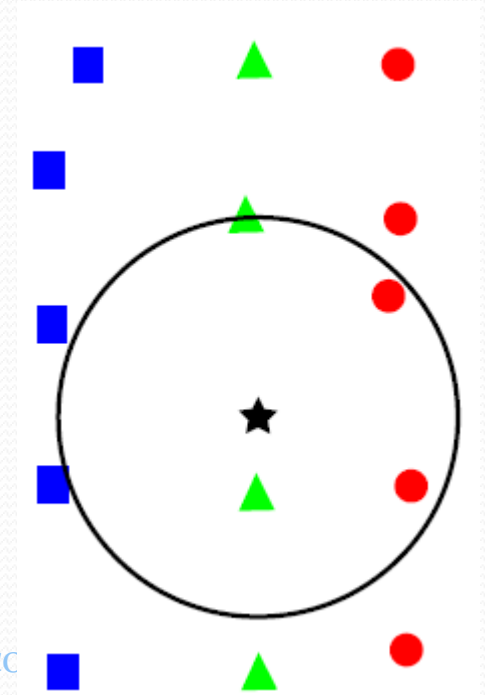


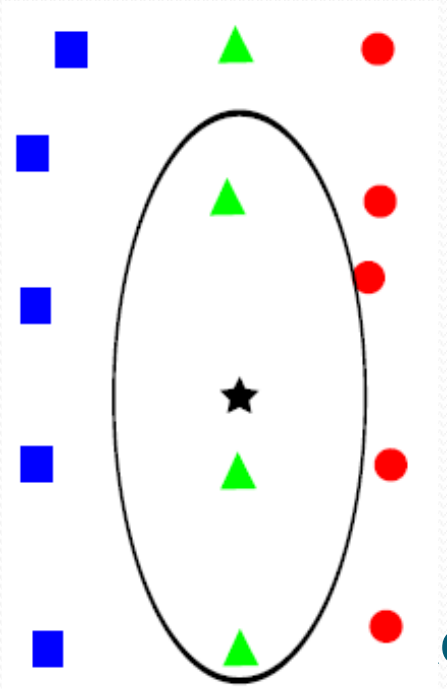
K = 5

Metrika pre *k*NN

- Čo znamená najbližší? Veľký vplyv má metrika
- Vždy musíme prehľadať celú trénovaciu množinu na určenie susedov



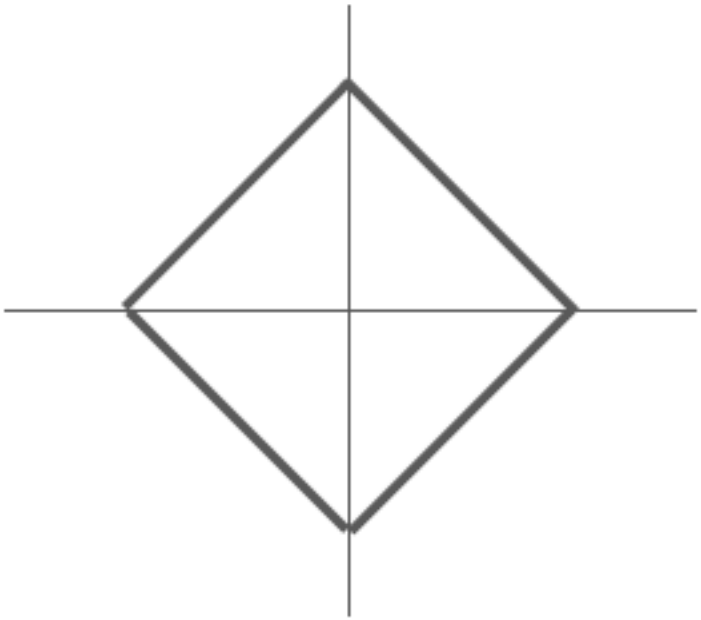




Metrika pre kNN II

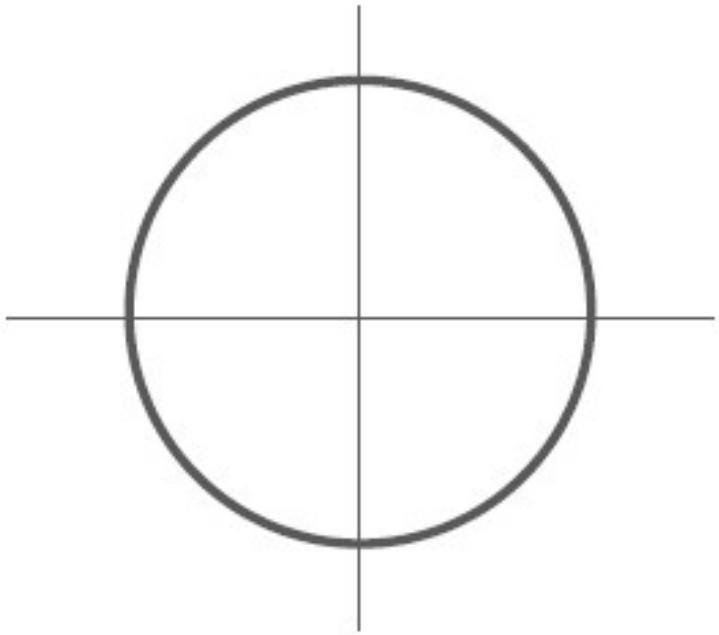
L1 (Manhattanská) metrika

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



L2 (Euklidovská) metrika

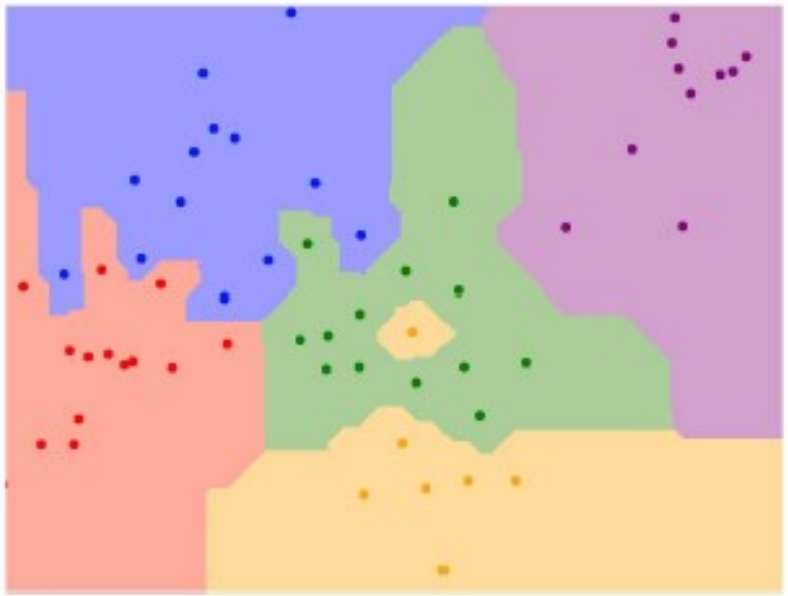
$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



Metrika pre *k*NN III

L1 (Manhattanská) metrika

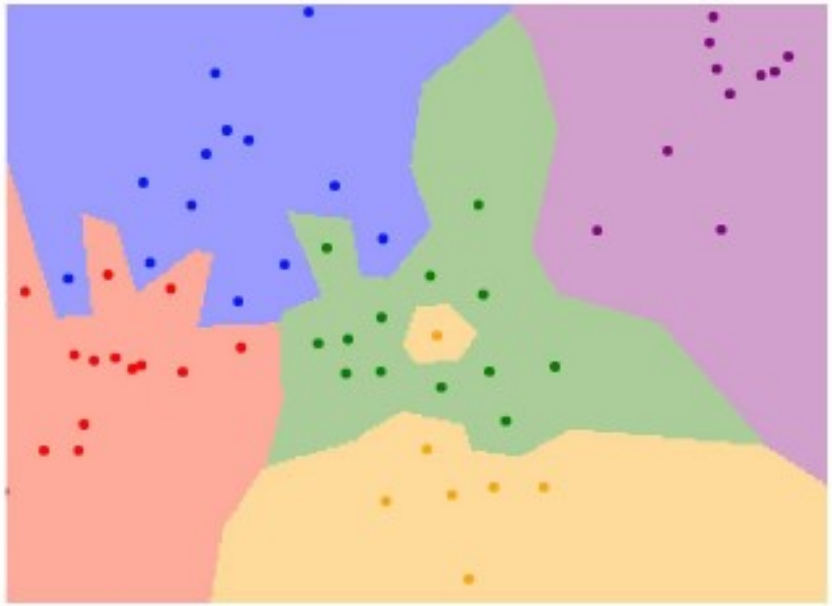
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



K = 1

L2 (Euklidovská) metrika

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



K = 1

K najbližších susedov VII

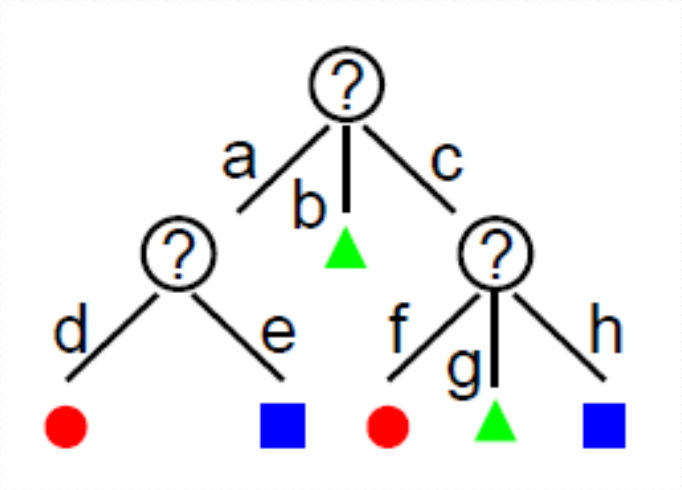
- Zvyšovanie *K:*
  - Vyhladzuje sa hranica
  - Zvyšuje/znižuje sa klasifikačná chyba
  - Rastie výpočtová náročnosť
- Vzájomná validácia na určenie *K:*
  - Trénovacia/validačná množina
  - Priemer chyby na trénovacích množinách

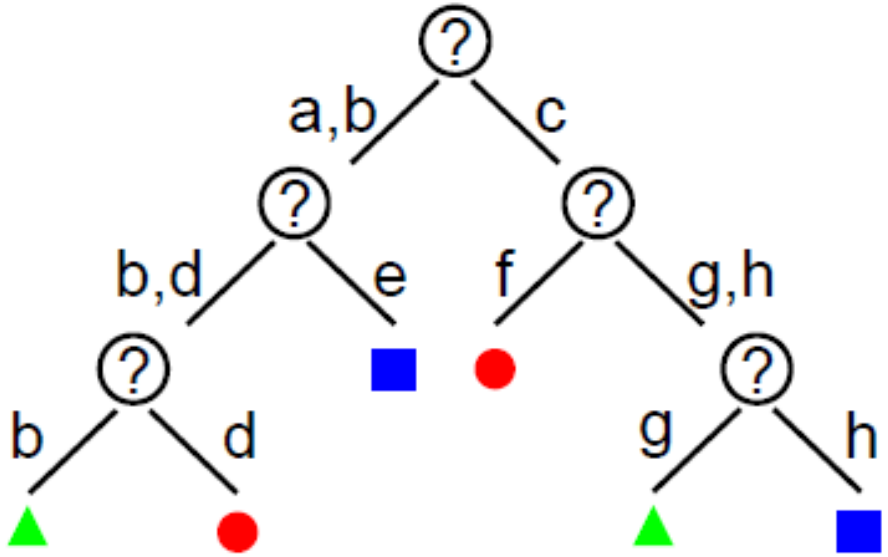
Rozhodovacie stromy

- Používajú sa na nominálne dáta, kde pojem vzdialenosti stráca zmysel
- Strom:
  - Uzly sú testy, ktoré majú viacero možných výsledkov
  - Hrany zodpovedajú možným výsledkom testu
  - List = klasifikačná trieda

Rozhodovacie stromy II

- Vytvárame binárne alebo *n*-árne stromy
- Binárne stromy môžeme vyrobiť aj z príznakov s viacerými hodnotami





Rozhodovacie porovnanie

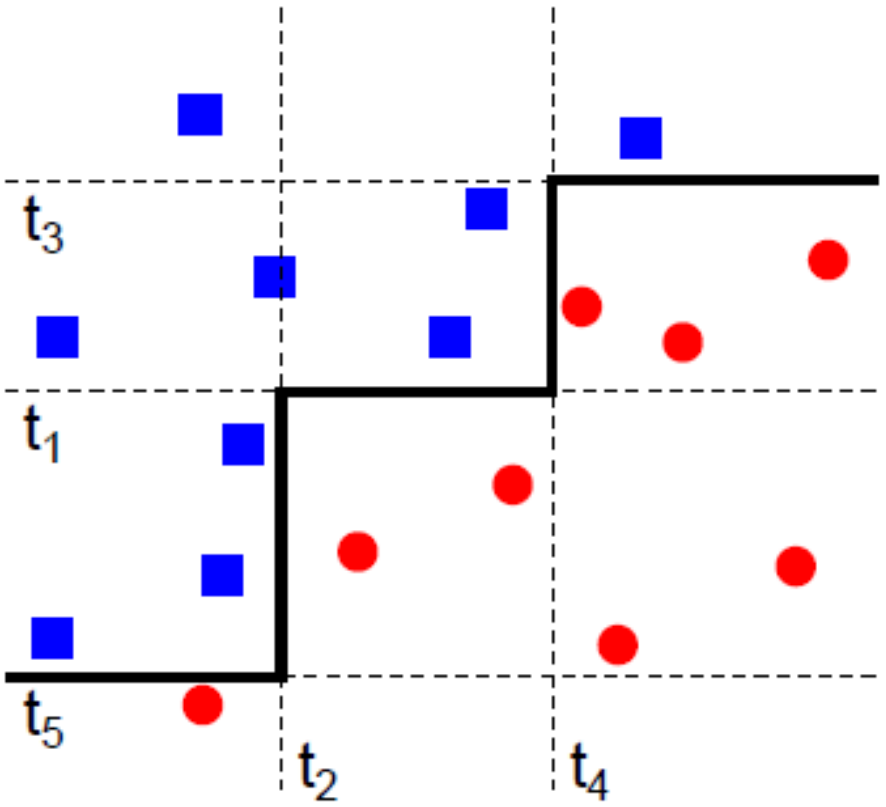
- V každom uzle sa pýtame, aký je vzťah hodnoty zvoleného príznaku k vybranej prahovej hodnote
- Koreň stromu skúma celú trénovaciu množinu
- Uzol t skúma podmnožinu X_t trénovacej množiny a rozdeľuje ju na dva podmnožiny X_{tA}, X_{tN}

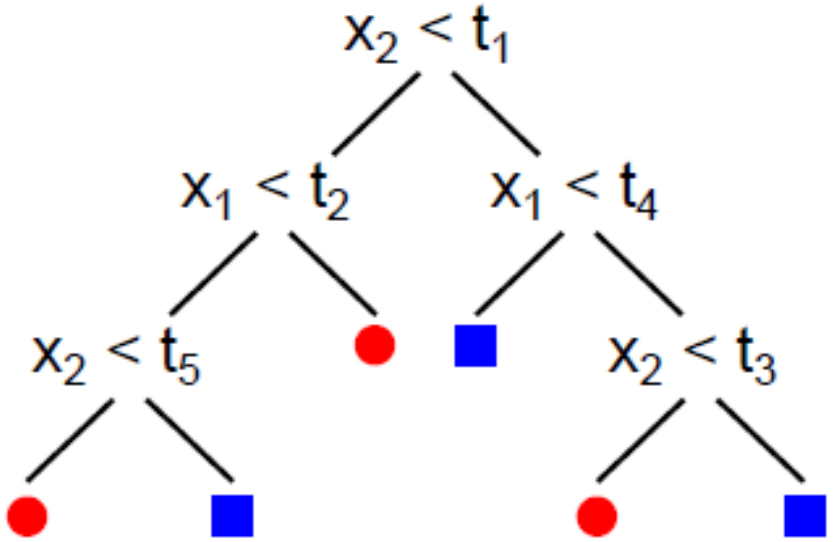
$$X_{tA} \bigcap X_{tN} = 0,$$
$$X_{tA} \bigcup X_{tN} = X_t$$

Rozhodovacie porovnanie II

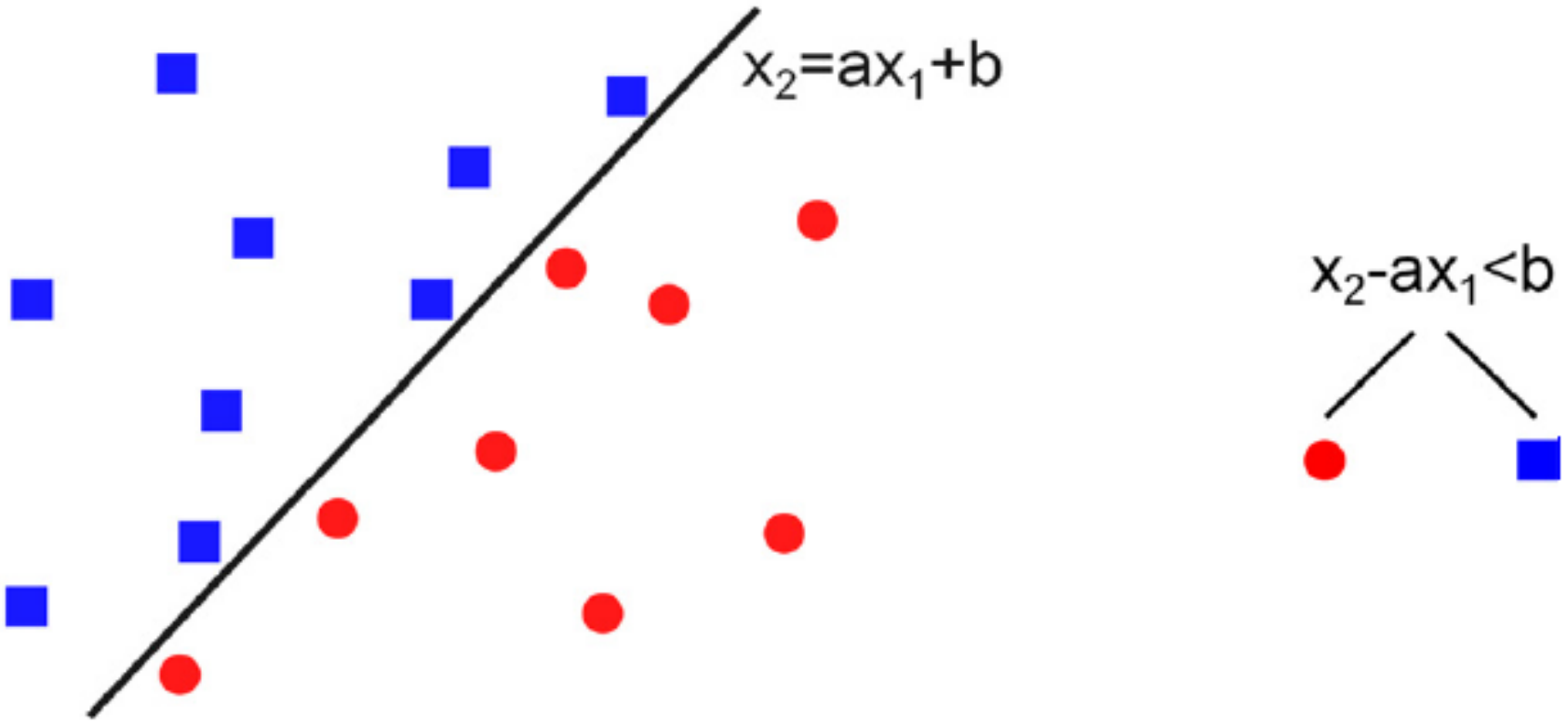
- Rozhodovacie porovnanie – pýtame sa na jeden príznak alebo viac (napr. $p_j > 7$, chuť = sladká apod.)
- Do testov môže vstupovať aj viacero príznakov spoločne, pričom väčšina viacnásobných testov je lineárnou kombináciou príznakov
- Ale väčšinou sa pýtame na jeden príznak

Rozhodovacie porovnanie III





Rozhodovacie porovnanie IV



Voľba parametrov

- **Rozdeľujúce kritérium**
  - Určuje prahovú hodnotu a príznak, ktoré budú vystupovať v rozhodovacom porovnaní
- **Ukončujúce kritérium**
  - Riadi rast stromu
- **Pravidlá**
  - Určujú klasifikačnú triedu v listoch

Rozdeľujúce kritérium II

- Je zvolené buď ako entropia (neurčitosť) príznaku alebo ako vzájomná informácia
- Čím je entropia príznaku nižšia, tým je vyšší jeho informačný prínos
- Metóda ID3 (Iterative Dichotomiser 3) – je založená na minimalizácii entropie príznakov
- Metóda C4.5 – zložená na maximalizácii vzájomnej informácie

Metóda ID3

- 1. Vypočítaj entropiu každého príznaku v trénovacej množine *S*
- 2. Vyber príznak, pre ktorý entropia je minimálna (alebo informačný prínos je maximálny) a rozdeľ *S* na podmnožiny podľa tohto príznaku
- 3. Vytvor vrchol rozhodovacieho stromu s týmto príznakom
- 4. Rekurzívne opakuj na podmnožinách pre ostatné príznaky

Metóda C4.5

- Je rozšírením algoritmu ID3, čiže postup je podobný
- Pracuje s normalizovaným informačným prínosom príznaku v podobe maximalizácie vzájomnej informácie medzi príznakom a skúmanou trénovacou (pod)množinou *S*
- Pracuje s kategorickými aj spojitými, aj chýbajúcimi dátami

Ukončujúce kritérium

- Všetky objekty v skúmanej množine patria do jednej klasifikačnej triedy
- Strom dosiahol maximálne stanovenú hĺbku
- Počet objektov klasifikovaných v danom uzle je menší ako stanovený prah
- Ohodnotenie najlepšieho príznaku je nižšie ako stanovený prah

Postup pri tvorbe stromu

1. vezmi všetky doteraz nepoužité príznaky, ohodnoť ich
2. do nového uzla vezmi príznak s najlepším ohodnotením
3. pre každý výsledok porovnania hodnoty príznaku vytvor podmnožinu dát

- Postup pri tvorbe binárneho stromu, bez orezávania je nasledovný:

Postup pri tvorbe binárneho stromu

```
PROC RozhStrom (X)
  vytvor strom T s jediným vrcholom V
  IF je splnené ukončovacie kritérium THEN
     označ V ako list a zaraď do určenej triedy
  ELSE
     nájdi príznak P a prahovú hodnotu H pomocou
        rezdeľujúceho kritéria
     vyhodnoť rozhodovacie porovnanie
     urči množiny XtA, XtN
     TA=RozhStrom(XtA)
     TN=RozhStrom(XtN)
     spoj uzol V s TA hranou označenou A
     spoj uzol V s TN hranou označenou N
  END IF
  RETURN T
```

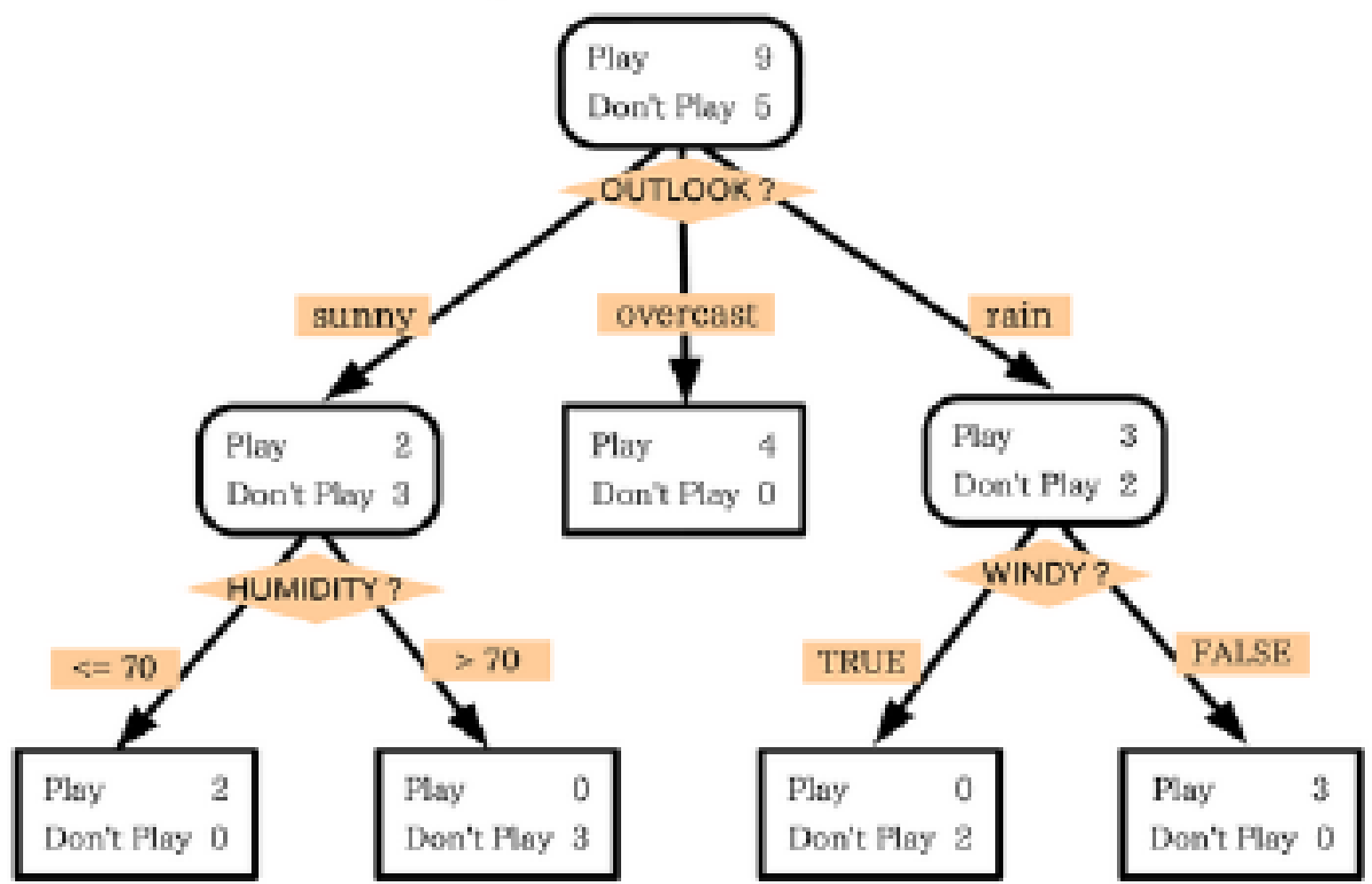
Príklad C4.5

- Maximalizácia MI – aké otázky si môžeme položiť?

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

Príklad C4.5 II

- Ideálny výsledok

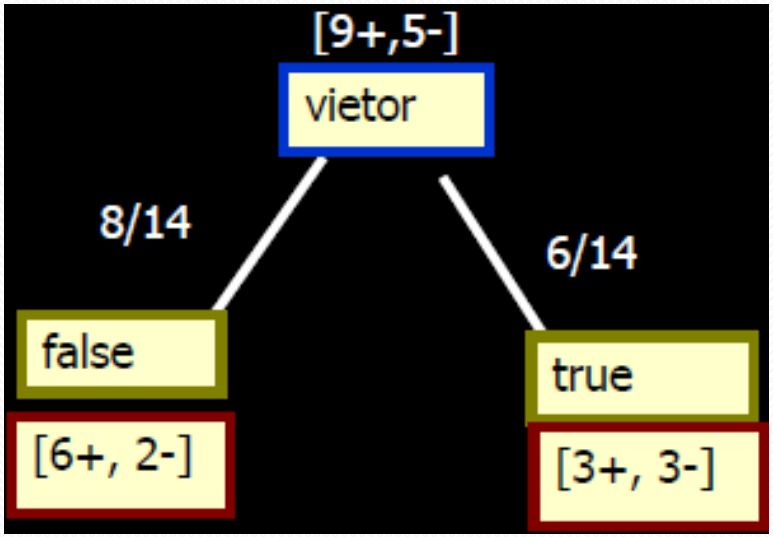


Príklad C4.5 III

			Windy	
sunny	69	70	FALSE	Play
overcast	81	75	FALSE	Play
overcast	83	78	FALSE	Play
rain	68	80	FALSE	Play
rain	75	80	FALSE	Play
sunny	85	85	FALSE	Don't Play
sunny	72	95	FALSE	Don't Play
rain	70	96	FALSE	Play
overcast	64	65	TRUE	Play
rain	65	70	TRUE	Don't Play
sunny	75	70	TRUE	Play
rain	71	80	TRUE	Don't Play
overcast	72	90	TRUE	Play
sunny	80	90	TRUE	Don't Play

Y – trieda

X – príznak



I(S;vietor)=0,940-(8/14)*0,811-(6/14)*1.0 =0,048

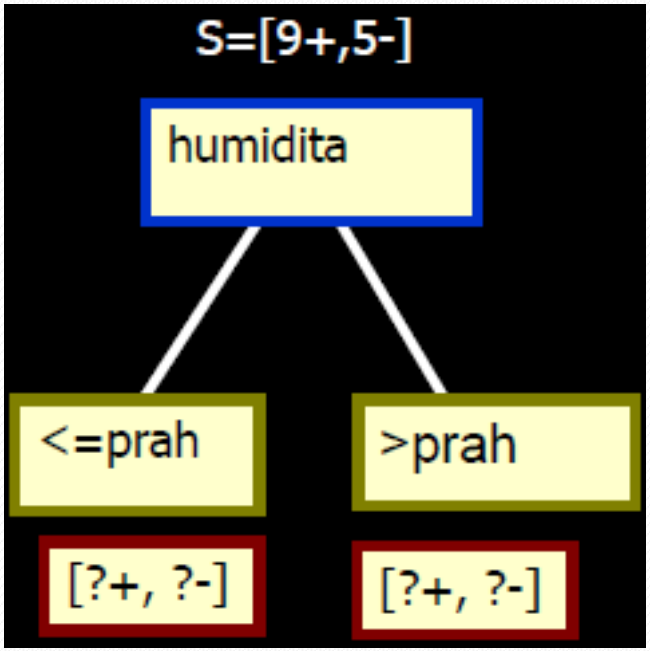
$$I(Y;X) = H(Y) - H(Y|X) = -\sum p_i.log_2(p_i) - \sum P(X \in \omega)H(Y|X \in \omega)$$

Príklad C4.5 IV

- Prah $h = \frac{v_i + v_{i+1}}{2}$ pre rozdelenie

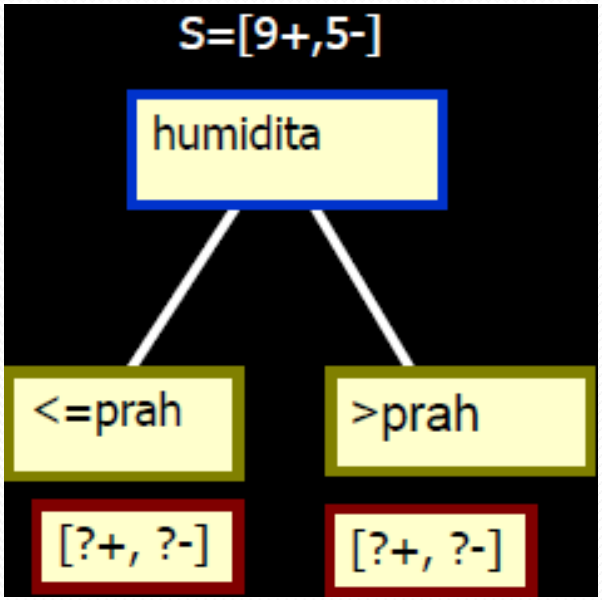
$\{v_1, v_2, \ldots, v_i\}$ a $\{v_{i+1}, v_{i+2}, \ldots, v_k\}$

- Takýchto možných rozdelení je $k - 1$
- Hodnotiaca funkcia rozdelenia
- Vyberieme rozdelenie s maximálnym ohodnotením



Príklad C4.5 V

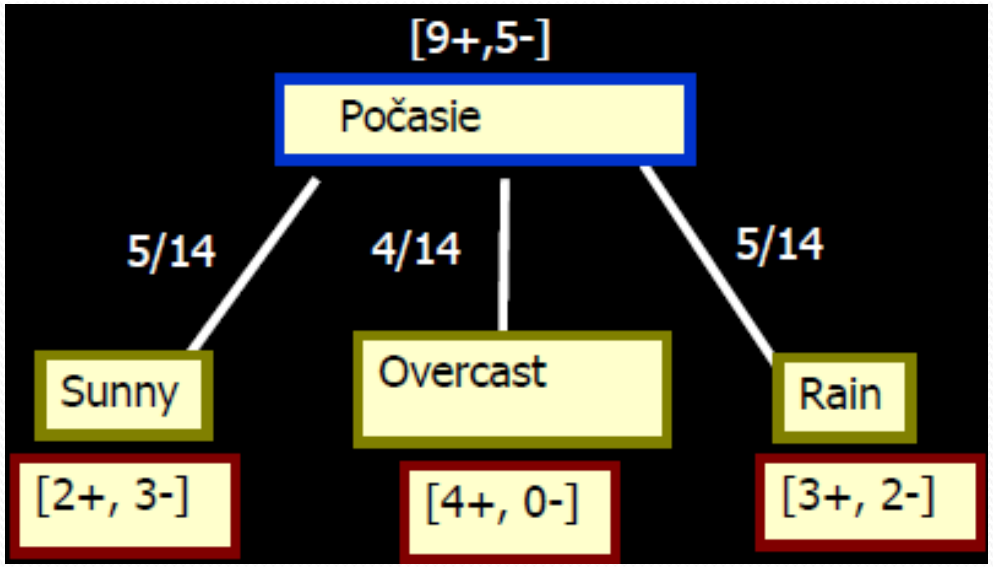
Outlook	Temp	Humidity	Windy	Play (positive) / Don't Play (negative)
overcast	64	65	TRUE	Play
rain	65	70	TRUE	Don't Play
sunny	69	70	FALSE	Play
sunny	75	70	TRUE	Play
overcast	81	75	FALSE	Play
overcast	83	78	FALSE	Play
rain	68	80	FALSE	Play
rain	71	80	TRUE	Don't Play
rain	75	80	FALSE	Play
sunny	85	85	FALSE	Don't Play
overcast	72	90	TRUE	Play
sunny	80	90	TRUE	Don't Play
sunny	72	95	FALSE	Don't Play
rain	70	96	FALSE	Play



- 8 možných prahov I(S;Humidity) = 0,102
- Najlepší 82,5
- Obdobne teplota: 70,5 I(S;Teplota) = 0,045

Príklad C4.5 VI

Outlook	Temp	Humid	Windy	Play (positive) / Don't Play (negative)
overcast	64	65	TRUE	Play
overcast	81	75	FALSE	Play
overcast	83	78	FALSE	Play
overcast	72	90	TRUE	Play
rain	65	70	TRUE	Don't Play
rain	68	80	FALSE	Play
rain	71	80	TRUE	Don't Play
rain	75	80	FALSE	Play
rain	70	96	FALSE	Play
sunny	69	70	FALSE	Play
sunny	75	70	TRUE	Play
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
sunny	72	95	FALSE	Don't Play



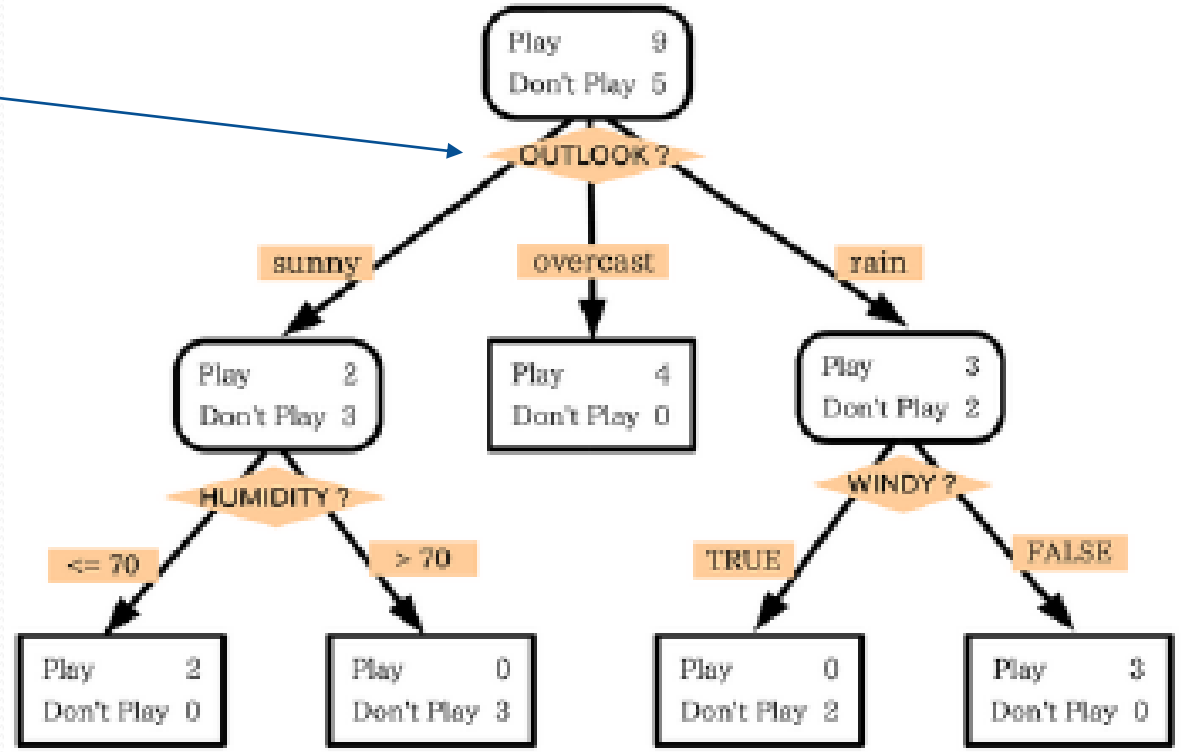
I(S;Počasie) =

= 0,940 – (5/14)*0,971-(4/14)*0-(5/14)*0,971 =

= 0,247

Príklad C4.5 VII

- I(S;Počasie) = 0,247
- I(S;Vietor) = 0,048
- I(S;Humidity) = 0,102
- I(S;Teplota) = 0,045



sunny	69	70	FALSE	Play
sunny	75	70	TRUE	Play
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
sunny	72	95	FALSE	Don't Play

rain	65	70	TRUE	Don't Play
rain	68	80	FALSE	Play
rain	71	80	TRUE	Don't Play
rain	75	80	FALSE	Play
rain	70	96	FALSE	Play

Orezávanie rozhodovacieho stromu

- **Počas generovania rozhodovacieho stromu**
  - Ak hodnota zvolenej miery významnosti pre skúmaný uzol nepresiahne stanovený prah, ďalšie vetvenie sa zastaví
- **Po vygenerovaní stromu**
  - Odstraňuje vetvy stromu takým spôsobom, že porovnáva veľkosť očakávanej chyby klasifikácie pre daný podstrom a jeho náhradu listovým uzlom. Ak sa chyba po náhrade nezväčší, podstrom sa odreže

Rozhodovacie stromy

- **Výhody**
  - Jednoduchá interpretácia
  - Dáta netreba predspracovať
  - Numerické aj kategorické dáta
  - Biela skrinka (booleovská logika)
  - Nízka výpočtová náročnosť
- **Nevýhody**
  - Optimálny strom – NP úplný problém
  - Heuristiky nezaručujú optimálny strom
  - Možnosť preučenia

Náhodné lesy (random forests)

- Ide o množinu rozhodovacích stromov, z kto-rých každý je náhodne obmedzený tak, aby bol citlivý iba na vybranú podmnožinu príznakov
- To im umožní pri raste stromu zvyšovať presnosť, ale znižuje možnosti preučenia
- To, či dokážu mať nízku chybu na testovacej množine, závisí od sily jednotlivých stromov a ich korelácie

Náhodné lesy II

- Vo všeobecnosti rozhodovacie stromy, ktoré sú príliš hlboké, sa vedia naučiť aj veľmi nepravidelné obrazce a majú tendenciu k preučeniu
- Náhodné lesy sú cestou priemerovania viacerých hlbokých rozhodovacích stromov trénovaných na rôznych častiach trénovacej množiny
- Znižuje sa tým vysoký rozptyl rozhodovacích stromov a zlepšuje sa výkon finálneho modelu

Náhodné lesy III

- Obmena trénovacej množiny pri tvorbe stromu:
- Pre každé $b = 1, \ldots, B$ náhodne vyberieme N príznakových vektorov (s opakovaním) a túto množinu označíme X_b, ich zaradenie do tried označíme ako Y_b
- Vytvoríme rozhodovací strom f_b na tejto trénovacej množine

Náhodné lesy IV

- Potom rozhodnutie o zaradení neznámej vzorky robíme na základe väčšinového rozhodnutia vytvorených stromov $f_1, \ldots, f_B$
- To znižuje rozptyl výsledného modelu, lebo jeden rozhodovací strom môže byť ovplyvnený šumom v trénovacej množine
- Viackrát trénovať strom na tej istej množine by viedlo k tomu istému stromu, preto obmena

Náhodné lesy V

- Číslo B sa volí ako hyperparameter, typicky od zopár sto po niekoľko tisíc
- Pri náhodných stromoch používame upravený všeobecný algoritmus, ktorý vyberie pri každej obmene trénovacej množiny aj náhodnú podmnožinu príznakov
- Typicky pre D príznakov vyberá podmnožinu s $\sqrt{D}$ príznakmi – to zníži koreláciu príznakov