

STROJOVÉ UČENÍ



Kurz: **Datové struktury a algoritmy**

Lektor: Doc. Ing. Radim Burget, Ph.D.

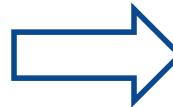
Autor: Doc. Ing. Radim Burget, Ph.D.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Vytvoření této videopřednášky bylo podpořeno projektem č. CZ.1.07/2.2.00/28.0098
Evropského sociálního fondu a státním rozpočtem České republiky.

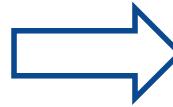
Souvislost s předchozími přednáškami



Paměť počítače

Znalost /
myšlenka

Seznamy
pole, stromy,
grafy, ...



Znalost

Uložená data

Jak z uložených
dat získat znalost?

„Od počátku naší civilizace až do roku 2003 bylo vytvořeno kolem 5 exabajtů dat. Podobné množství dat nyní vyprodukuje lidstvo každé dva dny. Přitom se tempo růstu množství dat neustále zvyšuje.“

Eric Schmidt, výkonný ředitel Google, 2010

Poznámka

- Tento kurz je primárně v JAVA
- Datová věda se téměř ve všech případech točí okolo Pythonu, v případě pracovního pohovoru očekávat práci v tomto prostředí
 - TensorFlow
 - NumPy
 - SciPy
 - Pandas
 - Matplotlib
 - Keras
 - SciKit-Learn
 - PyTorch

Cíl přednášky

1. Dolování znalostí (Data Mining)

- Motivace, Historie, Souvislosti
- Základy dolování znalostí
- Typické úlohy dolování
- Učení bez učitele
 - K-means
 - K-medoids
- Vstupní data
 - Základní pojmy
 - Sběr dat (struktura)
 - Předzpracování dat
 - Zpracování dat => ML

2. Strojové učení (Machine Learning)

- S učitelem
 - Typy dat, X-Validace, Srovnávání
 - Lineární a Logistická regrese
 - Rozhodovací stromy
 - Systém podpůrných vektorů
 - Neuronové sítě
 - Bayesovské sítě
 - k -NN

Motivace

- **Lidská mysl je zahlcena informacemi => Strojové učení**
 - Denně vzniká několik peta bajtů (10^{15} B)
 - Vzrůstající tlak na dolování znalostí z báze dat
 - Nutnost činit na základě znalostí dat určitá rozhodnutí
 - Potřeba rozhodovat se na základě faktů, nikoliv intuice
 - Sociální a ekonomické dopady na společnost i jedince
 - Dolování znalostí z dat textových, zvukových, obrazových

Historie

- **1960s: (Data Fishing, Data Dredging)**
 - Datové kolekce, Souborové databáze, DBMS
- **1970s:**
 - Relační databázový model, Relační implementace DBMS
- **1980s:**
 - Objektové databáze, Deduktivní databáze, Aplikačně orientované DBMS
- **1990s: (Data Mining, Knowledge Data Discovering)**
 - Dolování dat, Datové sklady, Multimediální a webové databáze
- **2000s: (Business Intell., Info. Harvesting, Knowl. Extraction...)**
 - Zpracování proudu dat, Globální informační systémy

Výuka: Bottom-up vs. Top-down

- Ze spodu nahoru
 - Známé základy
 - Postupně se nabízí podrobnosti
 - Dlouhou dobu není jasné, k čemu to vlastně celé je

- Ze shora dolů
 - Je zřejmý cíl a proč to dělám
 - Uniká spousta detailů

- Detekce útočníka v síti
 - Rozpoznání poruchy serveru
 - Detekce chyb v elektrické síti
 - ...
 - Rozpoznání druhu ovoce s pomocí základních parametrů

DEMO: Histogram

- Mějme 413 měření, kde levnými senzory je ke každému ovoci zjištěno: velikost (výška, šířka), váha a barva. Druhy ovoce jsou jablka, mandarinky, pomeranče a citrony
- Úkol: Zobrazte histogram, který ukáže, kolik měření pro jednotlivé druhy ovoce máme k dispozici



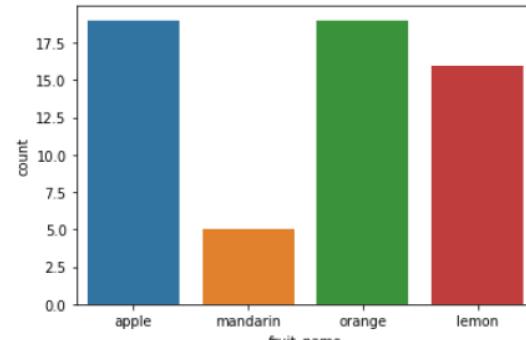
Na základě o předpovědět o

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79

DEMO: <https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>

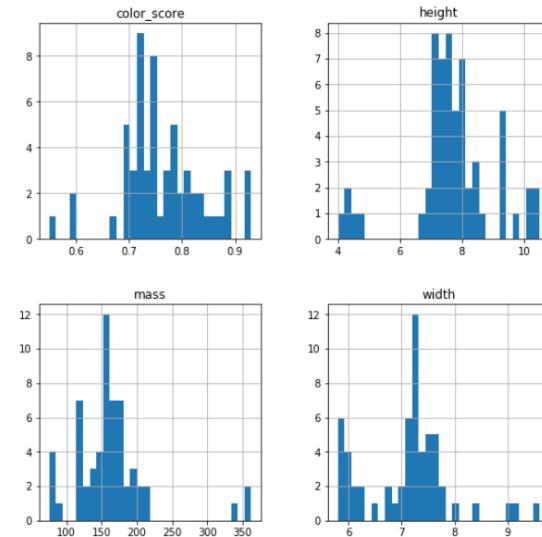
DEMO: Histogram

```
import seaborn as sns
sns.countplot(fruits['fruit_name'],label="Count")
plt.show()
```



Histogram for each numeric input variable

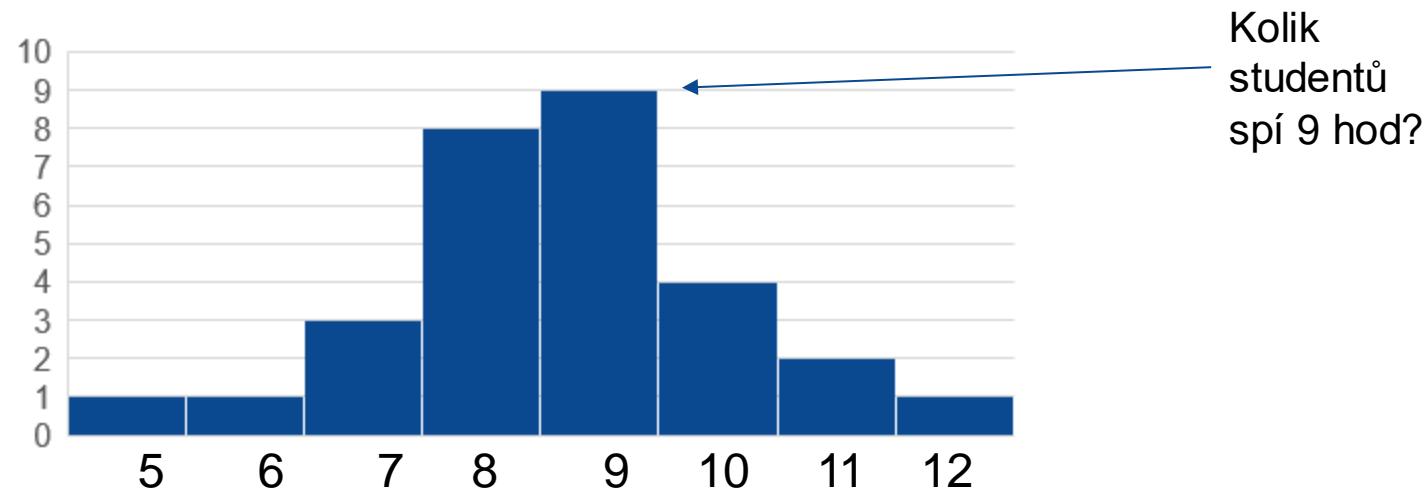
```
import pylab as pl
fruits.drop('fruit_label' ,axis=1)
    .hist(bins=30, figsize=(9,9))
pl.suptitle("Histogram for each numeric input variable")
plt.savefig('fruits_hist')
plt.show()
```



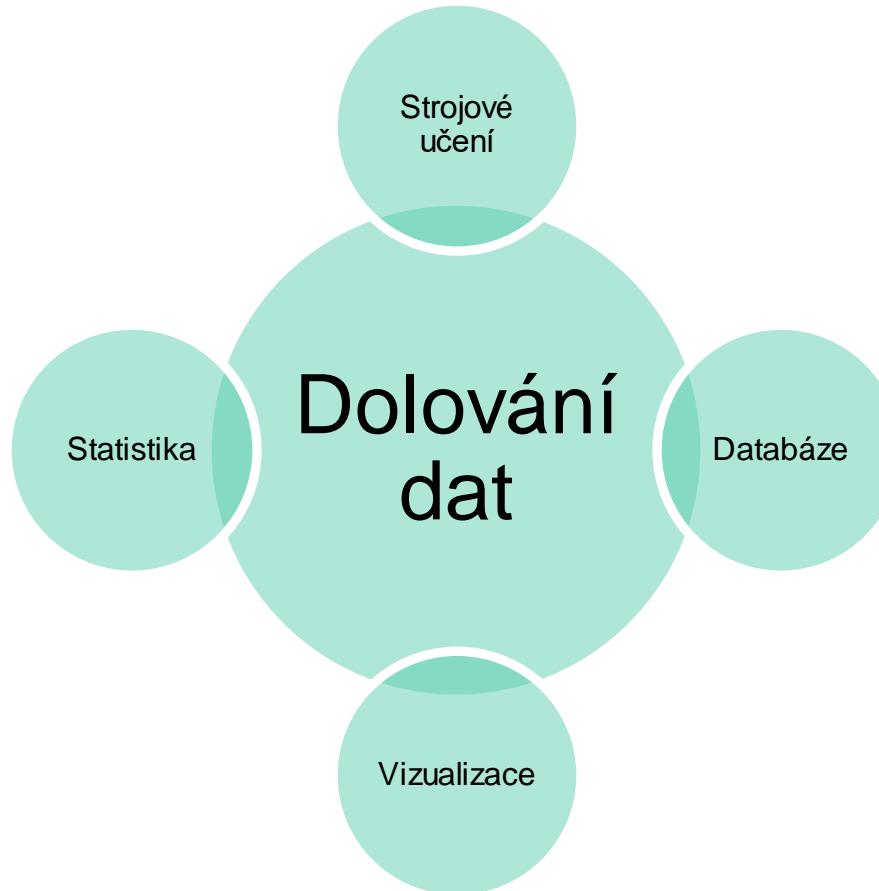
DEMO: <https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>

Histogramy - příklad

- Na základě oslovení studentů, kolik hodin spí před konáním výuky jsme obdrželi tyto odpovědi (dohromady 29)
 - 7 8 5 9 9 9 7 7 10 10 11 9 8 8 8 12 6 11 10 8 8 9 9 9 8 10 9 9 8
- S pomocí histogramu zobrazte:

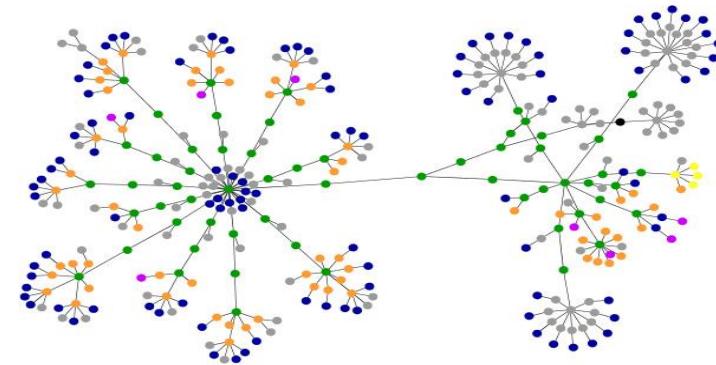
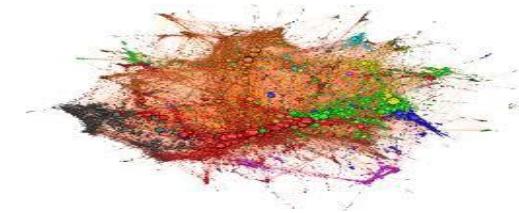


Souvislosti



Souvislosti

- **Databáze**
 - data, sběr, čištění, předzpracování, zpracování
- **Statistika**
 - přístupy zaměřené na teorii a
 - založené na hypotézách
- **Strojové učení**
 - přístupy zaměřené na algoritmy
 - založené na heuristikách (samočinné řešení problému)
- **Vizualizace**
 - převedení reprezentace problému do jednoduše pochopitelné podoby



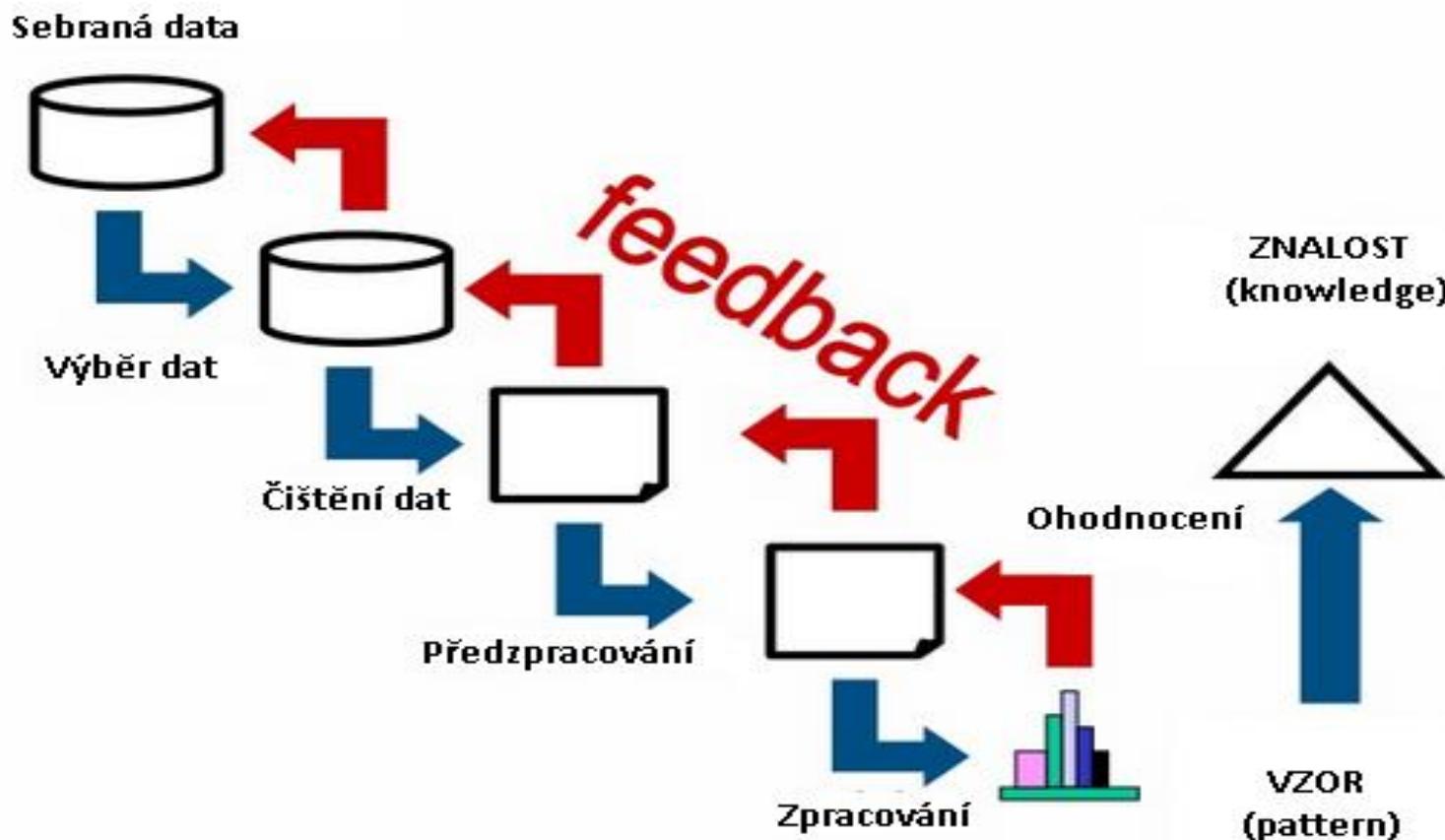
Dolování znalostí

- **Co není dolování znalostí?**
 - Vyhledání čísla v telefonním seznamu
 - Vyhledání hesla v internetovém vyhledávači
- **Co je dolování znalostí?**
 - Která příjmení jsou více používaná v určitých lokalitách
 - Předpověď ceny bytu podle jeho nejrůznějších parametrů
 - Binární klasifikace problému (ano/ne) na základě znalostí dat
 - Shlukování dokumentů na základě společných charakteristik

Souvislosti – Strojové učení vs. Statistika

- **Statistika**
 - Nejprve hypotéza, potom sběr dat a následně analýza
 - Často modelově orientované (silně parametrické modely)
- **Strojové učení**
 - Malá (či žádná) počáteční hypotéza
 - Data jsou již zpravidla získána dopředu
 - Analýza je řízena daty, ne hypotézou
 - Často algoritmicky orientované než modelově orientované
- **Statistické myšlenky jsou velmi užitečné i v dolování znalostí, např., při validaci zdali objevená znalost je použitelná**

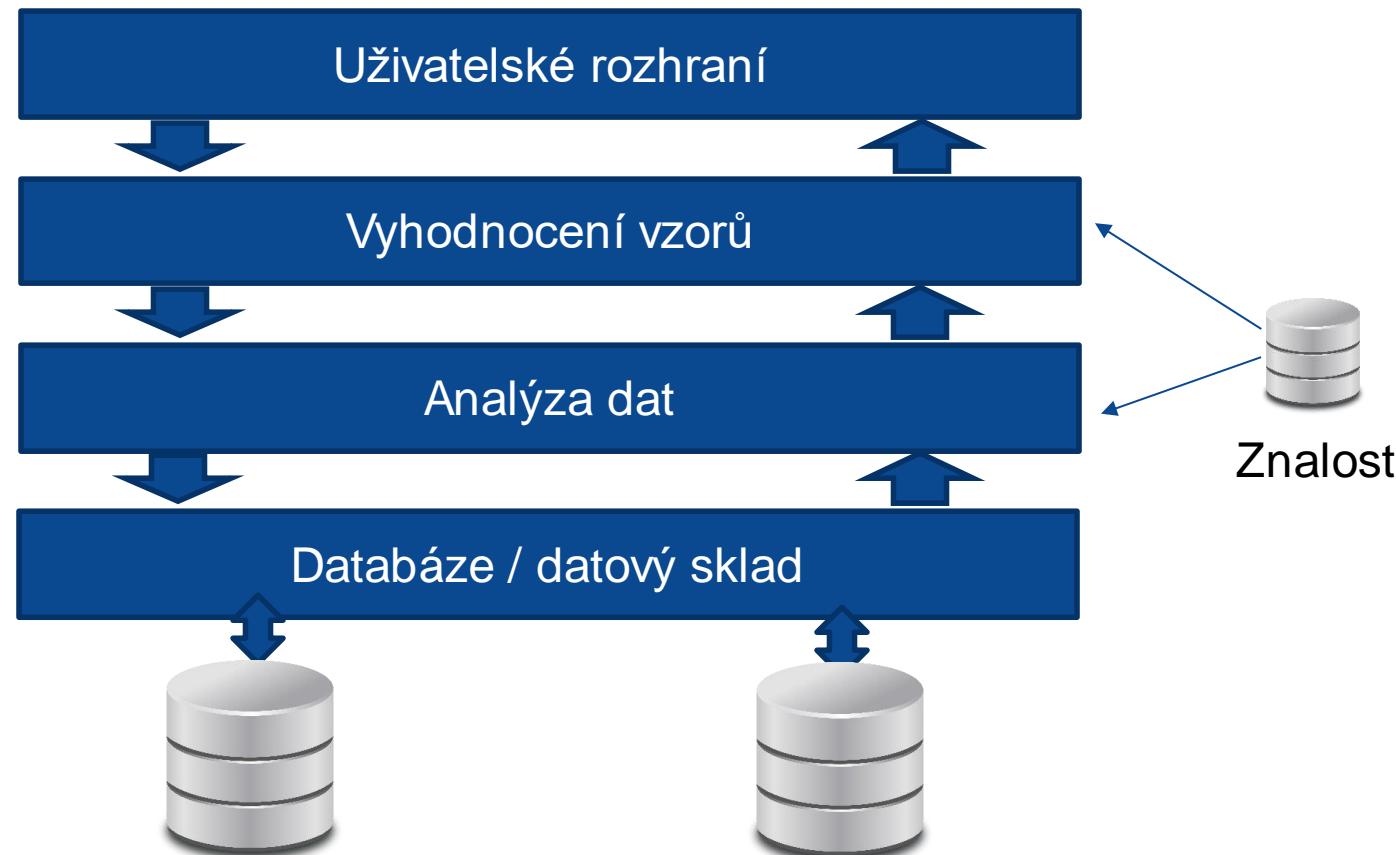
Dolování znalostí – Základní model



Dolování znalostí ve firemní praxi



Dolování znalostí a architektura aplikace



Metody prozkoumání dat

1. Prohlédnout si jednotlivá měření, ve většině případů nemožné (stovky a více měření nelze takto procházet)
2. Statistické metody (schopny poskytnout náhled do libovolného počtu vzorků)
 - Histogramy
 - Box plot
 - Scatter matice
 - Mnoho dalších

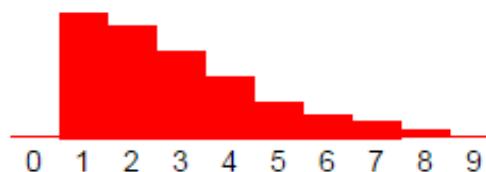
Histogramy

- Grafické zobrazení dat s použitím sloupců různých výšek.
- Každý sloupec vyjadřuje počet výskytů události v jednotlivých intervalech
- Histogramy se používají pro analýzu velkých datových množin kvantitativních dat.

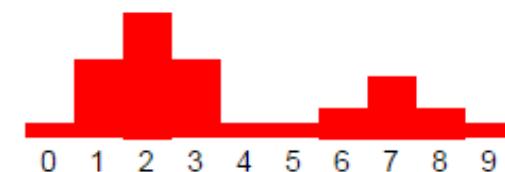
Histogram – typické tvary



Symetrický (zvobnovitý)
normální rozdělení



Šikmý doprava

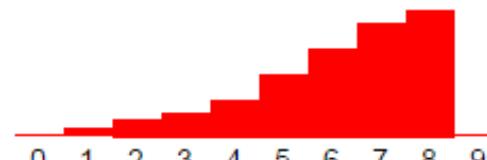


Bimodální – asymetrický
Naznačuje, že data pochází ze dvou zdrojů
a měly by být analyzovány odděleně

Případy jsou navíc nevyvážené, jedna třída
převáží druhý případ



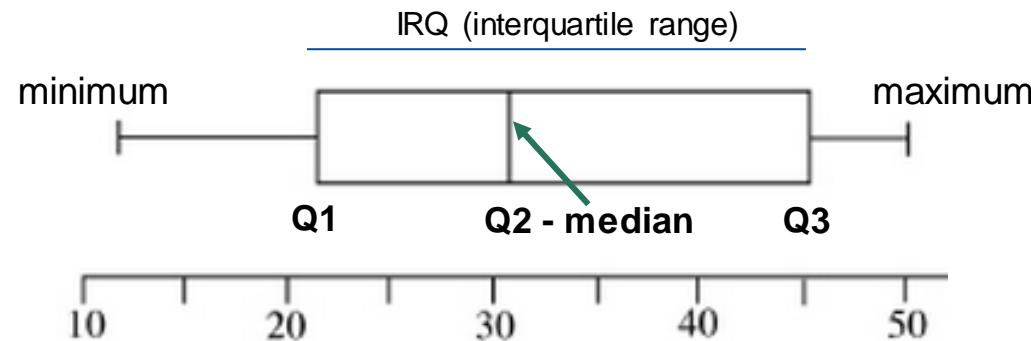
Uniformní
(dává nejméně informace)



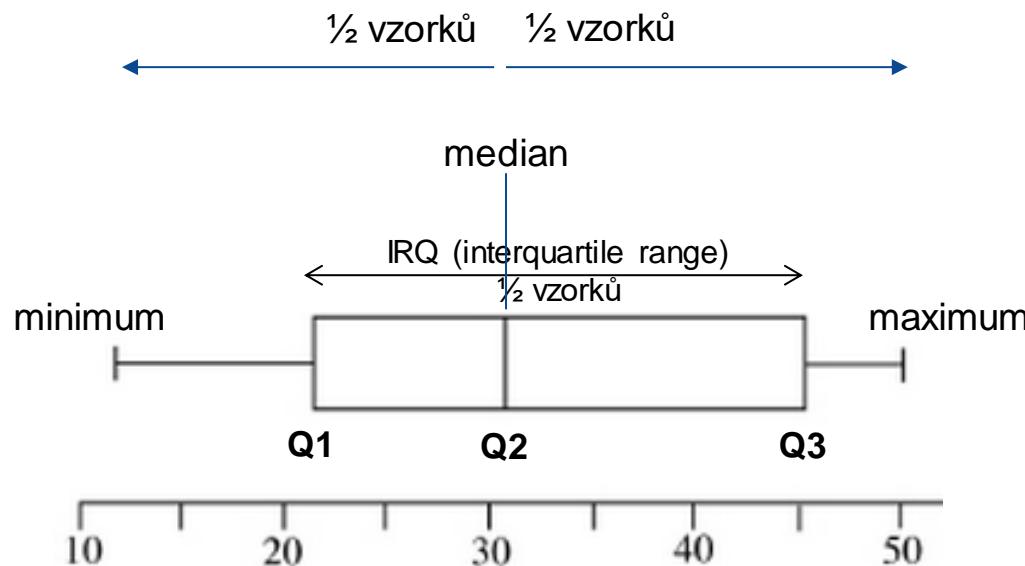
Šikmý doleva

Box Plot (krabicový graf/diagram)

- Rozděluje data do kvartilů
- Kvartil je $\frac{1}{4}$ či 25% celkových dat
- Tělo grafu je tvořeno “krabici” která začíná prvním kvartilem (Q1) každým třetím kvartilem (Q3).
- Uvnitř „krabice“ je v místě Q2 (medián) zakreslena svislá čára
- Dvě vodorovné čáry potom spojují data od minima k Q1 a od Q3 k maximu



Box Plot (krabicový graf/diagram) - význam



Normální rozdělení



Pozitivní zkosení
(např. platy, je jen málo milionářů)

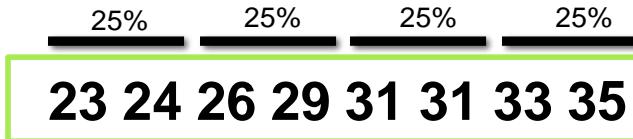


Negativní zkosení

Box Plot (krabicový graf/diagram)

- Zakreslete krabicový graf pro následující data:
31, 23, 33, 35, 26, 24, 31, 29

Krok 1: seřadit data:



Krok 2: určit medián Q2:

$$\text{Median} = (29 + 31) \div 2 = 30$$

Krok 3: Q1

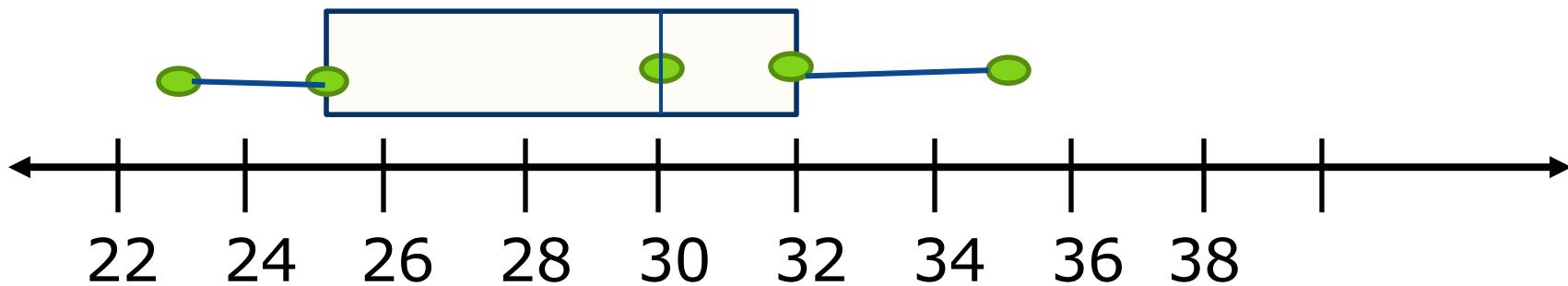
$$Q1 = (24 + 26) \div 2 = 25$$

Krok 4: Q3

$$Q3 = (31 + 33) \div 2 = 32$$

Box Plot (krabicový graf/diagram)

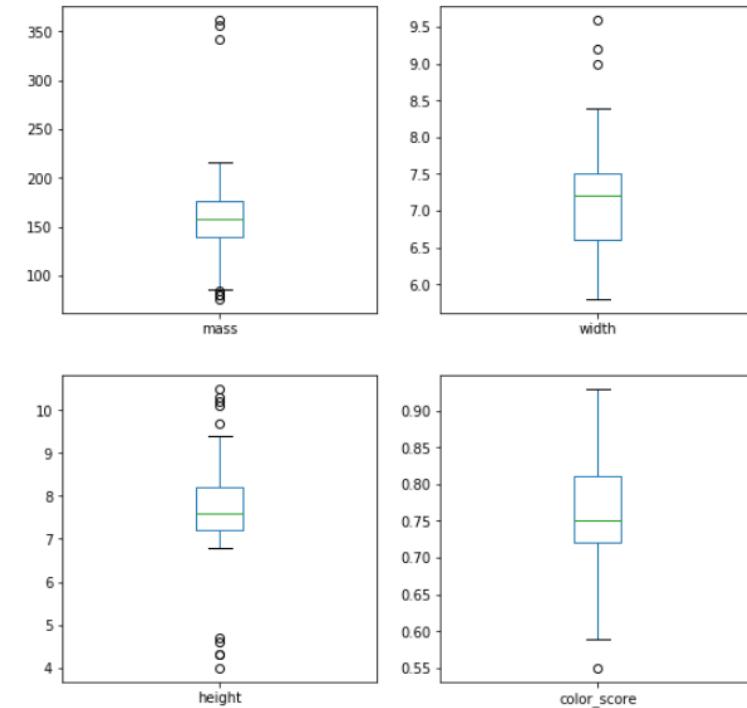
- Zakreslete krabicový graf pro následující data:
31, 23, 33, 35, 26, 24, 31, 29



Demo: Box Plot

```
fruits.drop('fruit_label', axis=1).plot(  
    kind='box', subplots=True,  
    layout=(2,2), sharex=False,  
    sharey=False, figsize=(9,9),  
    title='Box Plot for each input variable')  
  
plt.savefig('fruits_box')  
plt.show()
```

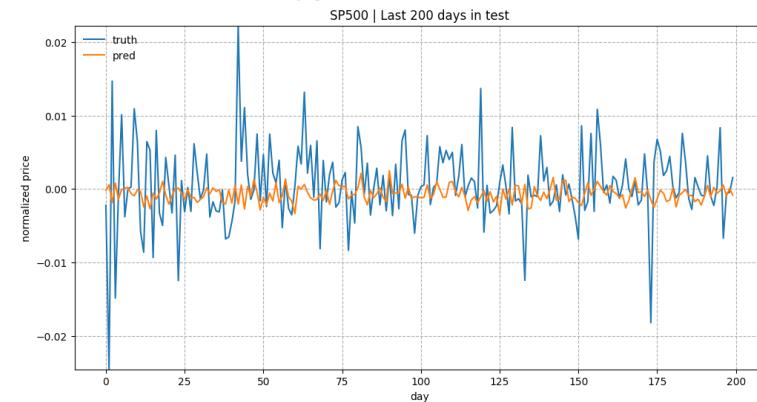
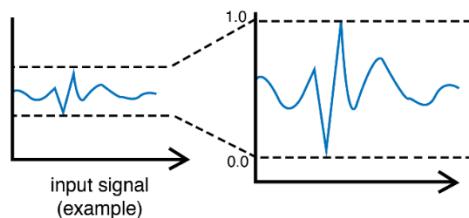
Box Plot for each input variable



DEMO: <https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>

Normalizace dat

- Motivace: mějme problém předpovídání ceny akcií na burze. Je pro algoritmus snazší určit absolutní hodnotu akcie anebo určit směr vývoje (nákup/prodej)

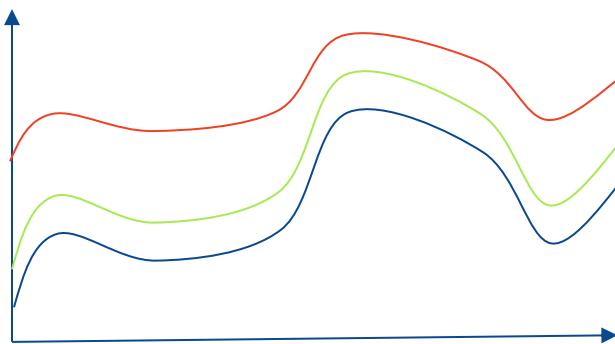


Obdobně se používá pro všechny typy dat,
vč. audia, obrazu, časových řad, ...

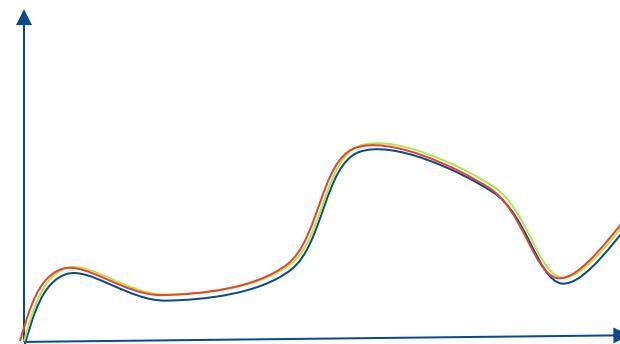
Normalizace dat

- Abychom usnadnili následujícím algoritmům práci, obvykle se provádí řada operací, která data transformují do podoby, aby usnadnila následné zpracování
 - Odhalí lepší vlastnosti, předejdou odhalení nesmyslů (zejména při nedostatku dat)
- V mnohých případech vytváříme nové atributy
 - Příklad: datum: 12. 11. 2019 pro předpověď hustoty dopravy:
 - -> úterý, není státní svátek, možnost sněhu, možnost mlhy, ...

Normalizace dat



VS.



Normalizace dat

Min-Max Normalizace

- min-max normalizace škáluje veškerá data v číselného attributu A do zadaného rozsahu
[nová hodnota – $minA$, nová hodnota – $maxA$].
- Transformace hodnoty v do nové hodnoty v' :

$$v' = \frac{v - min_A}{max_A - min_A} (new - max_A - new - min_A) + new - min_A$$

Normalizace dat

Z-score Normalizace

- Pokud minimum či maximum hodnoty **A není známo** (máme velmi málo hodnot měření), či data obsahují šum, *min-max* normalizace není vhodná
- Alternativa: normalizuj data attributu A, abychom obdrželi rozdělení se střední hodnotou $\bar{A} = 0$ and std. odchylkou σ_A rovnu 1

$$v' = \frac{v - \bar{A}}{\sigma_A}.$$

Z-score Normalizace

- Pokud minimum či maximum hodnoty **A není známo** (máme velmi málo hodnot měření), či data obsahují šum, *min-max* normalizace není vhodná
- Alternativa: normalizuj data attributu A, abychom obdrželi rozdělení se střední hodnotou $\bar{A} = 0$ and std. odchylkou σ_A rovnu 1

Příklad: normalizace 1/2

- Normalizujte atribut hodinová mzda:

Průměr: $\bar{A} = \frac{120+320+1500+170+200}{5} = 462$

- Standardní odchylka:

$$\begin{aligned}\sigma_A &= \sqrt{\frac{(120 - \bar{A})^2 + (320 - \bar{A})^2 + (1500 - \bar{A})^2 + (170 - \bar{A})^2 + (200 - \bar{A})^2}{5}} \\ &= \sqrt{\frac{(-342)^2 + (-142)^2 + (1038)^2 + (-292)^2 + (-262)^2}{5}} \\ &= 1169,820499\end{aligned}$$

Hodinová mzda

120
320
1500
170
200

Počet vzorků n=5

Více info: <https://scikit-learn.org/stable/modules/preprocessing.html>

Příklad: normalizace 1/2

- Normalizujte atribut hodinová mzda:

- Průměr: $\bar{A} = 462$
- Std. odchylka: $\sigma_A = 1169,82$
- $Z_{\text{SCORE}} = \frac{x-\bar{A}}{\sigma_A} = \frac{120-462}{1169,82} = -0,29235$

Hodinová mzda	Hodinová sazba (Z-score)
120	-0,29235
320	-0,12139
1500	0,887316
170	-0,24961
200	-0,22397

Více info: <https://scikit-learn.org/stable/modules/preprocessing.html>

Demo: Normalizace dat Min-Max

Vstup:

fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score	
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79



Výstup:

	0	1	2	3
0	0.188406	0.000000	0.709677	0.486486
1	0.289855	0.352941	0.500000	0.702703
2	0.268116	0.441176	0.483871	0.648649
3	0.130435	0.029412	0.612903	0.486486
4	0.297101	0.500000	0.451613	0.756757

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

<https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>
Bod 4)

Rozdělení strojového učení

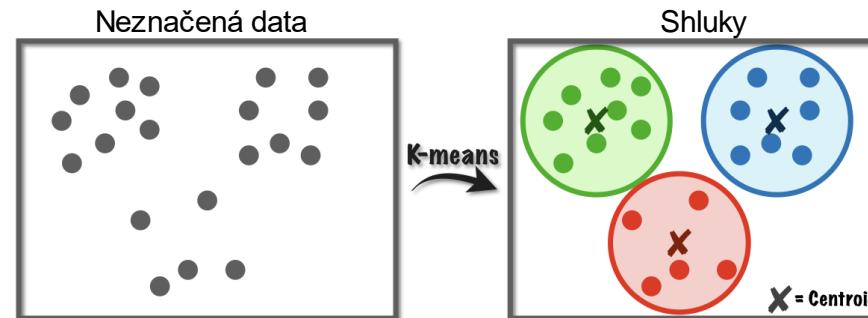
	Učení s učitelem	Učení bez učitele
Diskrétní	Klasifikace	Hledání shluků (např. k-means) Detekce anomálií
Spojité	Regresi	Snížení dimensionality

* Reinforcement learning

Dolování znalostí – Základní model

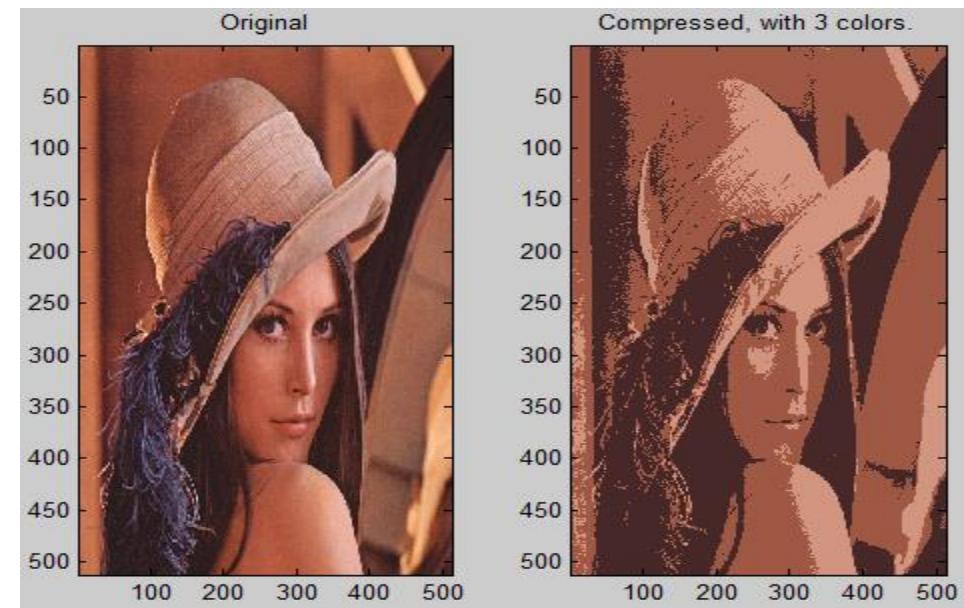
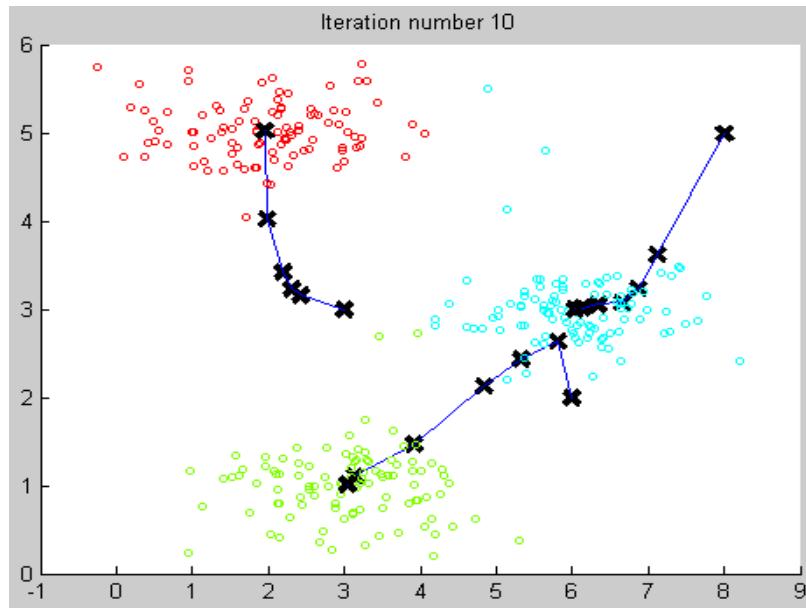
	Učení s učitelem	Učení bez učitele
Diskrétní	Klasifikace	Hledání shluků (např. k-means)
Spojité	Regresy	Snížení dimensionality

Data nemusí být značená
= „bez učitele“



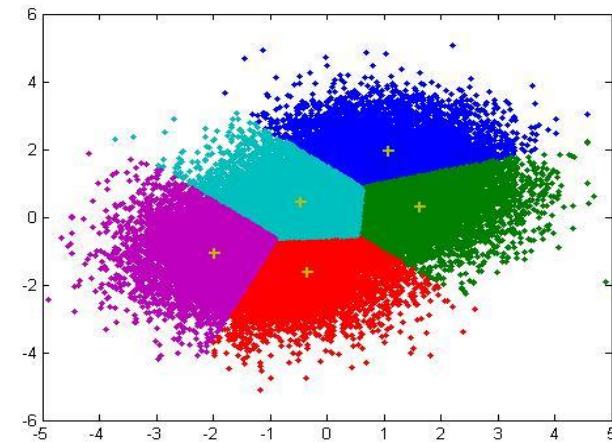
Typické úlohy dolování – Shlukování

- Kategorizace podobných dokumentů
- Na základě shluků je možné objevit (dříve neznámý) trend v datech -> určit budoucí vývoj či odvodit různá pravidla atd.



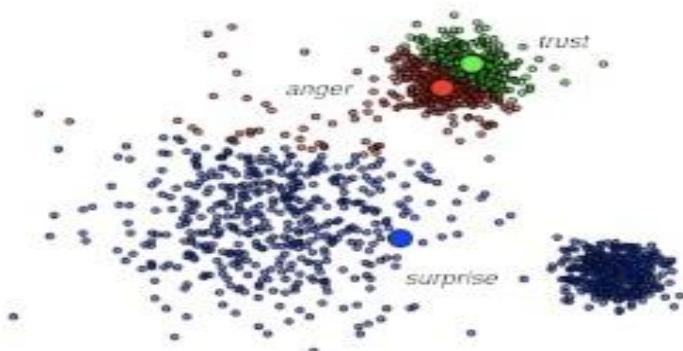
Strojové učení bez učitele (k -means)

- *Měří podobnost prvků ve shluku na základě vzdálenosti vůči středu shluku (centroid, center of gravity).*
- Středy jsou nejprve náhodně zvoleny a poté jsou v následných iteracích upravovány na základě sumy čtverců vzdáleností mezi středem a jednotlivými body ve shluku.
- Nevhodné pro shluky velmi odlišné velikosti.
- Citlivé na šum a na odchylky (outliers).



Strojové učení bez učitele (k -medoids)

- *Snaha eliminovat citlivost na šum a výrazné odchylky.*
- Místo výpočtu středu shluku se nalezne prvek, jenž daný shluk reprezentuje nejlépe.
- Ostatní prvky se zařadí do shluku, se kterým vykazují největší podobnost.
- Iteracemi znova postupně hledáme nevhodnější reprezentanty shluku.



k-means - algoritmus

Vstup: Stanovíme počet požadovaných shluků k + datová množina.

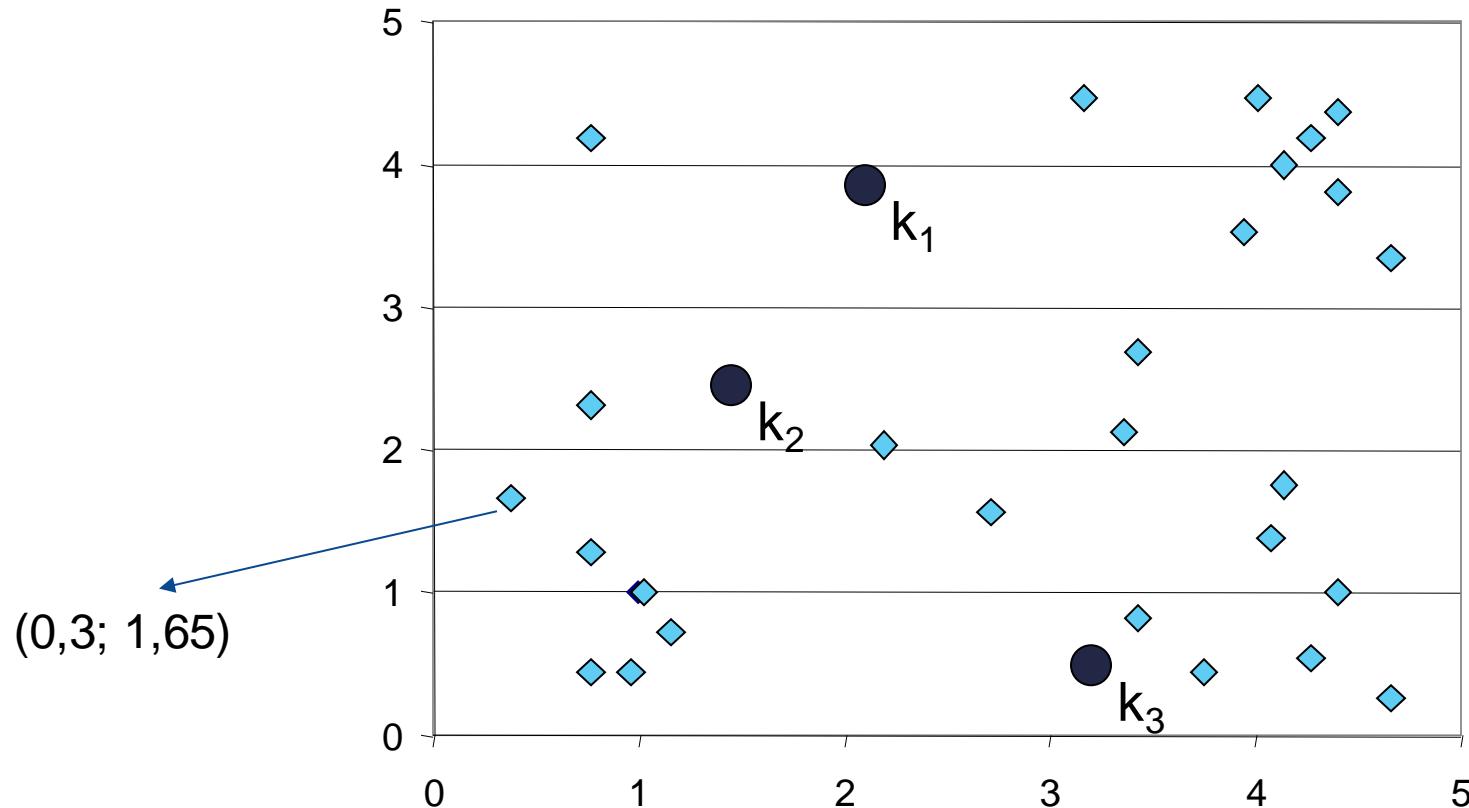
1. Inicializujeme polohu pro všech k centroidů (náhodně, pokud je potřeba).
2. Určíme členství každého vzorku ke centroidu
3. Aktualizujeme středy shluků, předpokládáme, že shluky jsou přiřazeny správně
4. Pokud žádná změna, končíme, jinak opakujeme od kroku 3



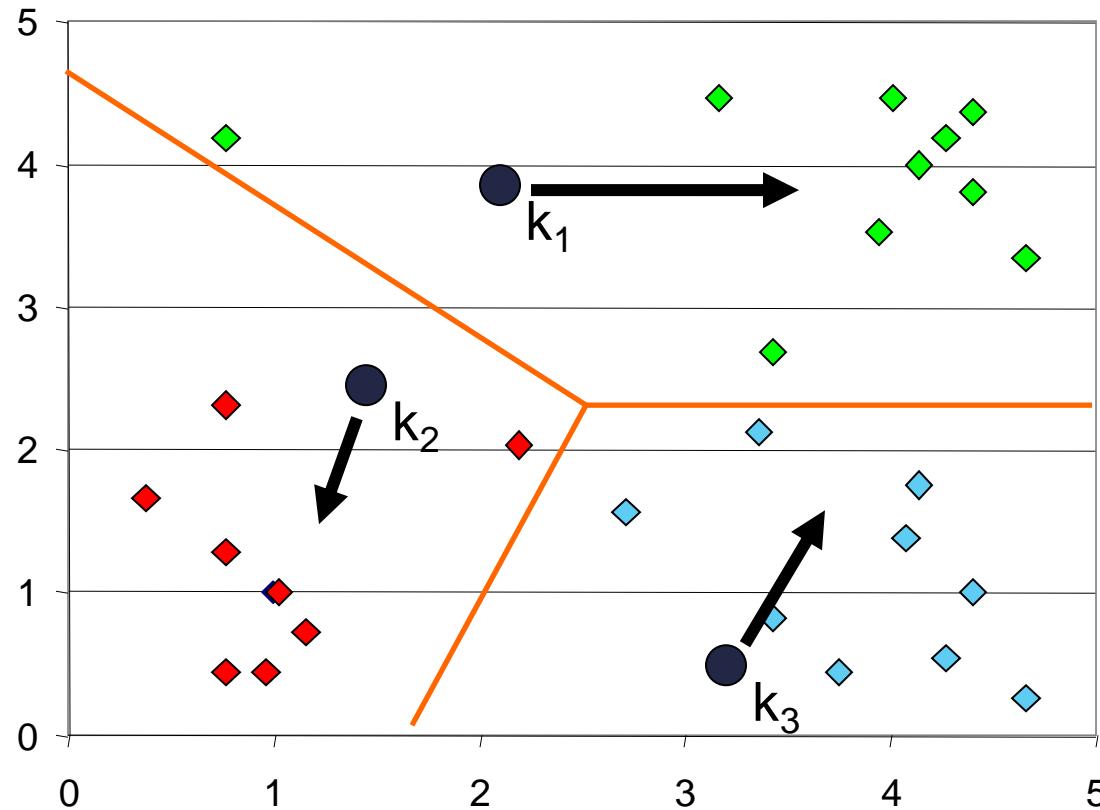
Výstup: souřadnice k centroidů (náležitost příkladů z datové množiny k nejbližšímu centroidu)

k-means : Krok 1

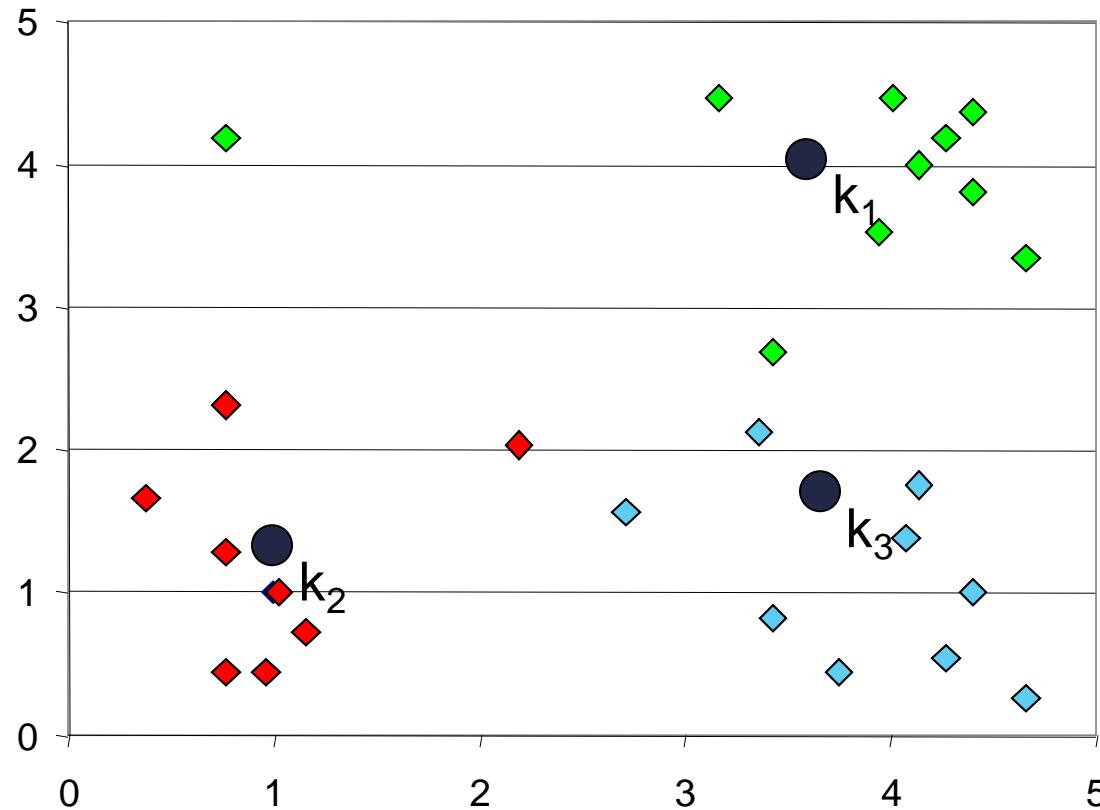
Centroidy: k_1 , k_2 , k_3 (pozice stanovena náhodně)
◆ příklad z datové množiny (vektor délky 2)



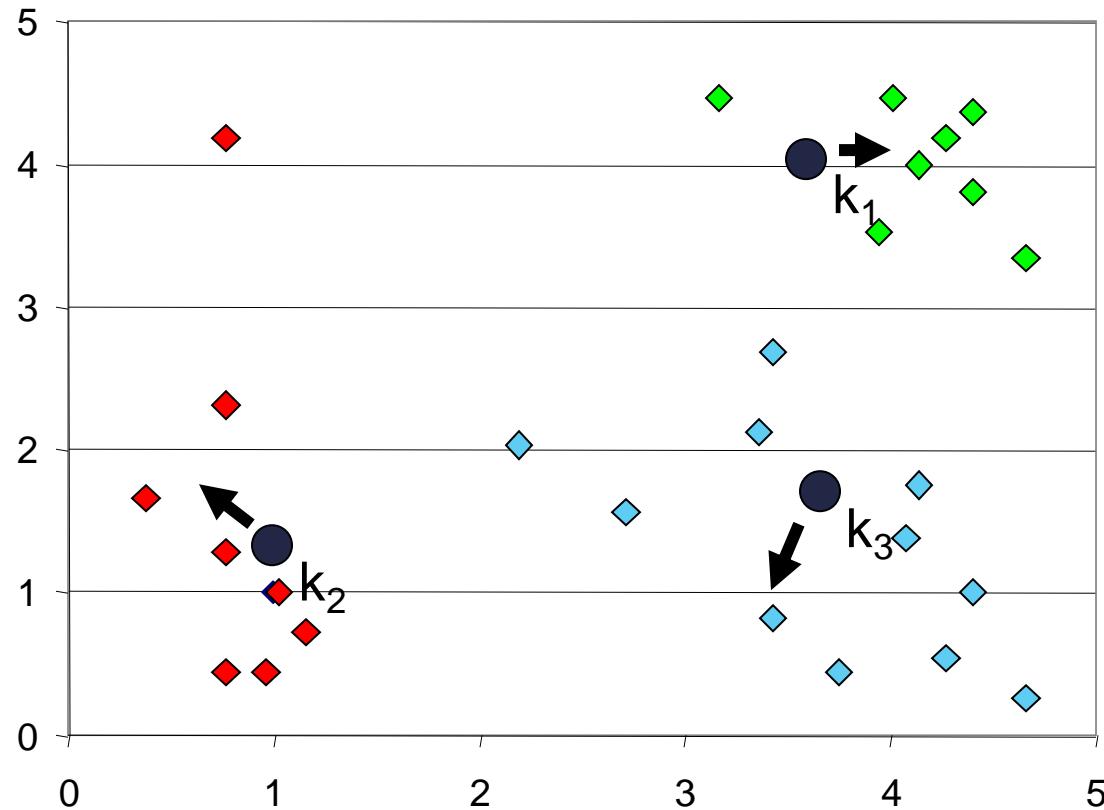
k-means : Krok 2



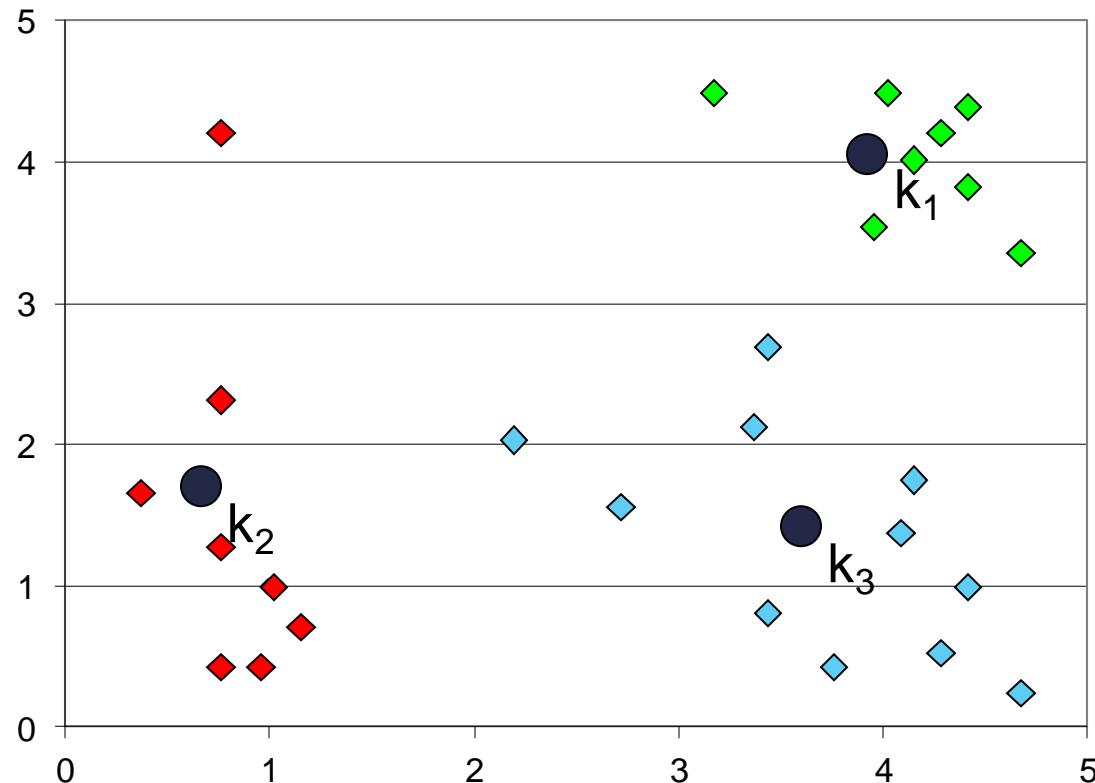
k -means : Krok 3



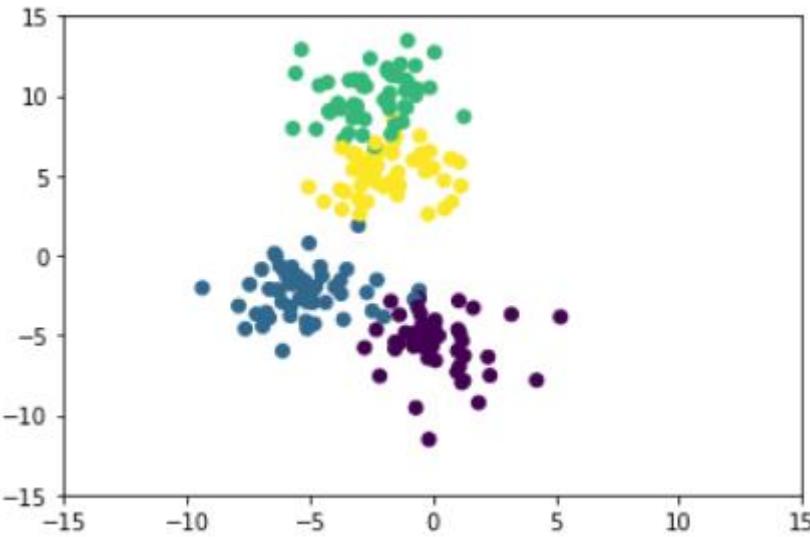
k -means : Krok 4



k -means : Krok 5



Demo: Shlukování

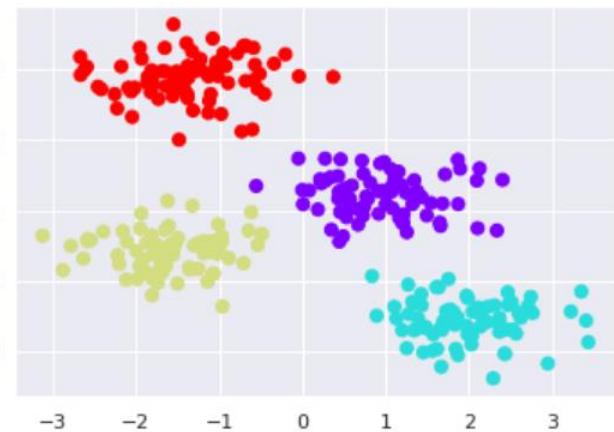
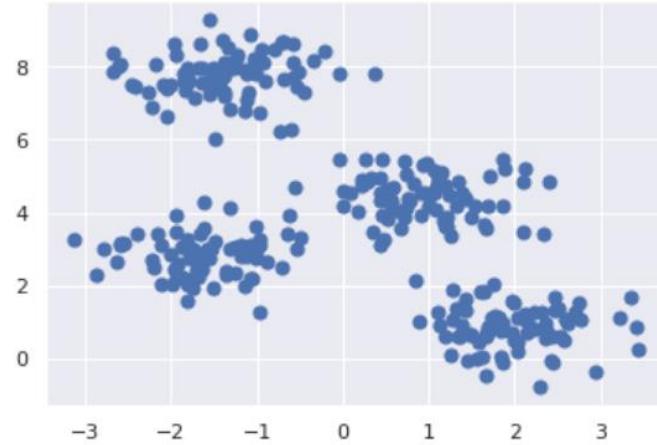


<https://colab.research.google.com/drive/1gffAVAj0LPJR211h1SUCb-kU50niNGmE>

Demo: k-means

- Najděte v datech 4 shluky

```
from sklearn.cluster import KMeans  
est = KMeans(4) # 4 clusters  
est.fit(X)  
y_kmeans = est.predict(X)
```

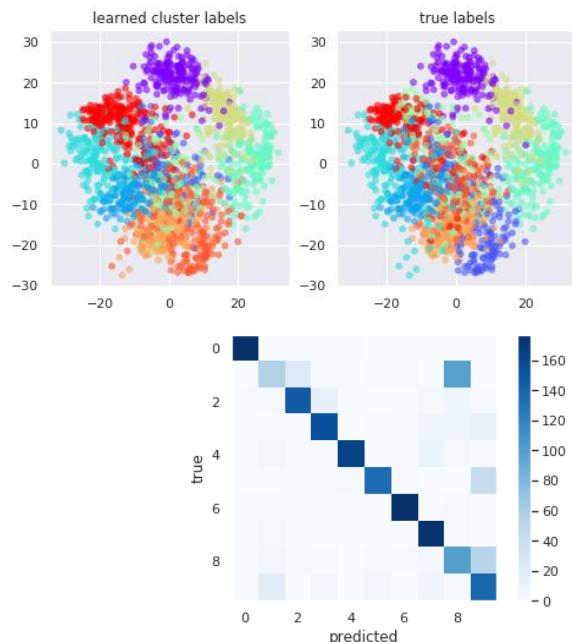
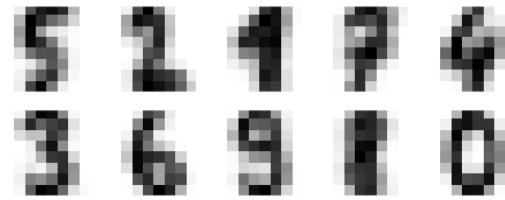


Demo:

https://drive.google.com/file/d/1wwJJ5GNlZ0_BA7n5Q_m4JzFied74JLke/view?usp=sharing

Demo: *k*-means

- Mějme ručně psaná čísla 0 až 9, u kterých nikdo zatím ručně neoznačil, o jaká čísla se jedná
- Jak tato čísla rozdělit do 10 kategorií?
 - Jaký je rozdíl oproti učení s učitelem?
 - Je přesnější s učitelem nebo bez učitele?
 - Jak drahé je ručně označit trénovací množinu? (např. MNIST 10 tis. obr.)



Demo:

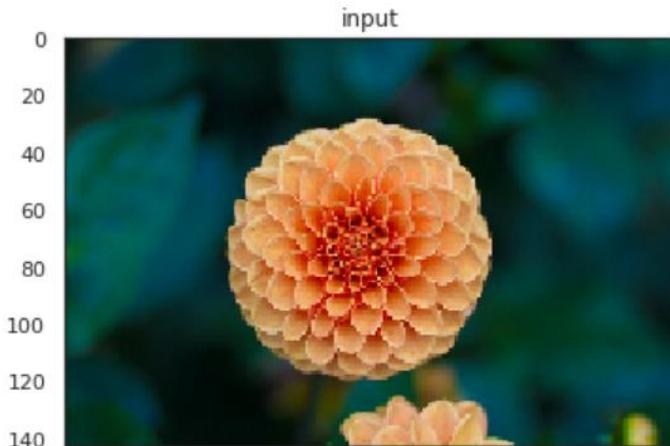
https://drive.google.com/file/d/1wwJJ5GNlz0_BA7n5Q_m4JzFied74JLke/view?usp=sharing

Demo: k-means

- Vstupní obrázek: $256^3 = 16\ 777\ 216$ barev
- Zkomprimujte reprezentaci obrázku s použitím pouhých 64 barev

```
model = KMeans(n_colors)
labels = model.fit_predict(X)
colors = model.cluster_centers_
```

```
new_image = colors[labels].reshape(image.shape)
new_image = (255 * new_image).astype(np.uint8)
```



Demo:

https://drive.google.com/file/d/1wwJJ5GNlz0_BA7n5Q_m4JzFied74JLke/view?usp=sharing

Dolování znalostí – Základní model

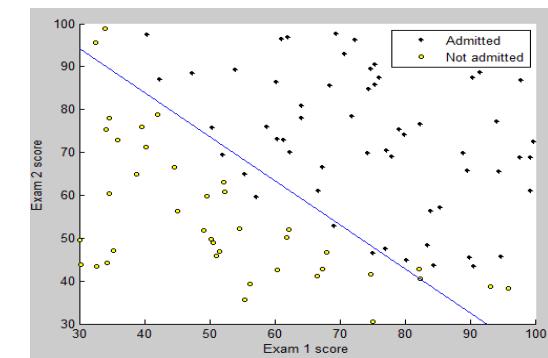
	Učení s učitelem	Učení bez učitele
Diskrétní	Klasifikace	Hledání shluků (např. k-means)
Spojité	Regresy	Snížení dimensionality

Abychom mohli model naučit, musíme označit trénovací vzorky
= „s učitelem“

$f(\text{apple}) = \text{"jablko"}$

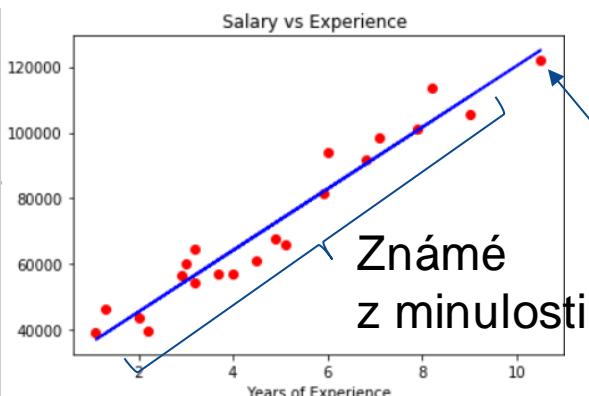
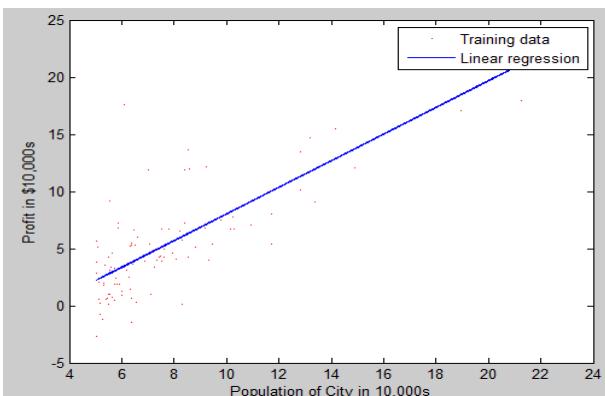
$f(\text{orange}) = \text{"rajče"}$

$f(\text{cow}) = \text{"kráva"}$



Dolování znalostí – Základní model

	Učení s učitelem	Učení bez učitele
Diskrétní	Klasifikace	Hledání shluků (např. k-means)
Spojité	Regres	Snížení dimensionality



Jakou lze očekávat
odměnu v 11 roce?

Dolování znalostí – Základní model

	Učení s učitelem	Učení bez učitele
Diskrétní	Klasifikace	Hledání shluků (např. k-means)
Spojité	Regresy	Snížení dimensionality

Jak rozpoznat síťového útočníka:

Kompletní záznam logů

vs.

Počet scanovaných portů

...

Typické úlohy dolování – Asociační pravidla

- Druhy zboží nakupované společně -> nabídka e-shopu nebo rozmístění zboží v supermarketu, tak aby zákazník utratil co nejvíce a na nic nezapomněl



Typické úlohy dolování – Sekvenční pravidla

- Analýza logů na serveru pro napadení / pokusu o napadení

```

128.195.36.195, -, 3/22/00, 10:35:11, W3SVC, SRVR1, 128.200.39.181, 781, 363, 875, 200, 0, GET, /top.html, -
128.195.36.195, -, 3/22/00, 10:35:16, W3SVC, SRVR1, 128.200.39.181, 5288, 524, 414, 200, 0, POST, /spt/main.html, -
128.195.36.195, -, 3/22/00, 10:35:17, W3SVC, SRVR1, 128.200.39.181, 30, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -
128.195.36.101, -, 3/22/00, 16:18:50, W3SVC, SRVR1, 128.200.39.181, 60, 425, 72, 304, 0, GET, /top.html, -
128.195.36.101, -, 3/22/00, 16:18:58, W3SVC, SRVR1, 128.200.39.181, 8322, 527, 414, 200, 0, POST, /spt/main.html, -
128.195.36.101, -, 3/22/00, 16:18:59, W3SVC, SRVR1, 128.200.39.181, 0, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -
128.200.39.17, -, 3/22/00, 20:54:37, W3SVC, SRVR1, 128.200.39.181, 140, 199, 875, 200, 0, GET, /top.html, -
128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 17766, 365, 414, 200, 0, POST, /spt/main.html, -
128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -
128.200.39.17, -, 3/22/00, 20:55:07, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -
128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 1061, 382, 414, 200, 0, POST, /spt/main.html, -
128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -
128.200.39.17, -, 3/22/00, 20:55:39, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -
128.200.39.17, -, 3/22/00, 20:56:03, W3SVC, SRVR1, 128.200.39.181, 1081, 382, 414, 200, 0, POST, /spt/main.html, -
128.200.39.17, -, 3/22/00, 20:56:04, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -
128.200.39.17, -, 3/22/00, 20:56:33, W3SVC, SRVR1, 128.200.39.181, 0, 262, 72, 304, 0, GET, /top.html, -
128.200.39.17, -, 3/22/00, 20:56:52, W3SVC, SRVR1, 128.200.39.181, 19598, 382, 414, 200, 0, POST, /spt/main.html, -

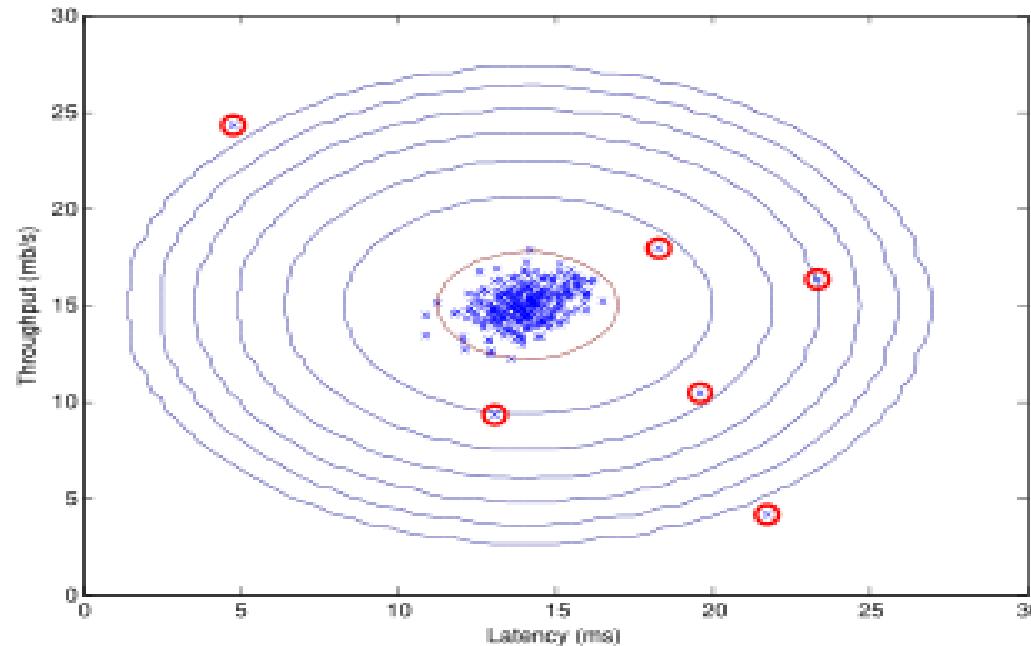
```

User 1	2	3	2	2	3	3	3	1	1	1	1	3	1	3	3	3	3
User 2	3	3	3	1	1	1											
User 3	7	7	7	7	7	7	7	7	7								
User 4	1	5	1	1	1	5	1	5	1	1	1	1	1	1	1	1	1
User 5	5	1	1	5													
...	...																



Typické úlohy dolování – Detekce anomálií

- Detekce anomálií v síti pomocí propustnosti a prodlevy



Dolování znalostí a typické úlohy

- **Dolování zn. se využívá téměř všude, kde se pracuje s informacemi**
- Zpracování textu
 - Průzkumy veřejného mínění
 - Analýzy finančních trhů
 - Určování a předpověď trendů
 - Reklamní průmysl, marketing
 - Datová žurnalistika
- Zpracování řeči (klasifikace emocí)
- Zpracování obrazu (např. klasifikace objektů, rozpoznávání objektů)
- Optimalizace procesů ve výrobě, veřejné dopravě, ve skladech
- Boj s Internetovou kriminalitou
- Vizualizace dat



Učení s učitelem

$f(\text{apple}) = \text{"jablko"}$
 $f(\text{orange}) = \text{"rajče"}$
 $f(\text{cow}) = \text{"kráva"}$

$$y = f(x)$$



- **Trénování:** je dána trénovací databáze značených vzorků $\{(x_1, y_1), \dots, (x_N, y_N)\}$, odhadujeme prediktivní funkci f s pomocí minimalizace chyby na této množině
- **Testování:** používáme f na dosud neviděné vzorky x předpovídáme jejich značku $y = f(x)$

Vstupní data – Sběr dat (struktura)

- **Strukturovaná**

- Mají definovanou strukturu (datový model)
- DB, DW, CSV, XML

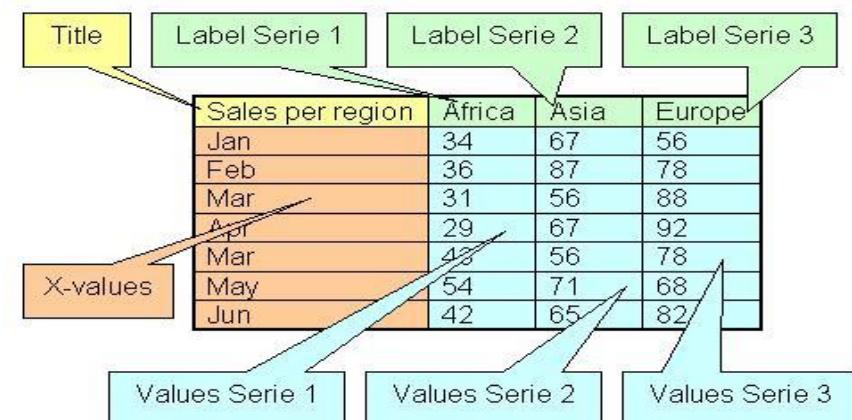
- **Nestrukturovaná**

- Nemají definovanou strukturu
- TXT, Zvuky, Obrazy, Videa

- **Semi-strukturovaná**

- HTML kód

	student	math	arts	philosophy
□	Benjamin	75	90	95
□	Leonardo	65	100	95
□	Galileo	93	75	85
□	Johannes	78	85	85
□	Codex	95	90	49
□	Albert	95	57	70
□	Isaac	95	62	55
□	Marie	61	75	76
□	Thomas	75	80	85
□	Michaelangelo	51	99	80
□	Socrates	55	75	100



Vstupní data – základní pojmy

m ... počet vzorků (entities)

X ... vstupy (features, příznaky...) ... i až n

y ... výstupy

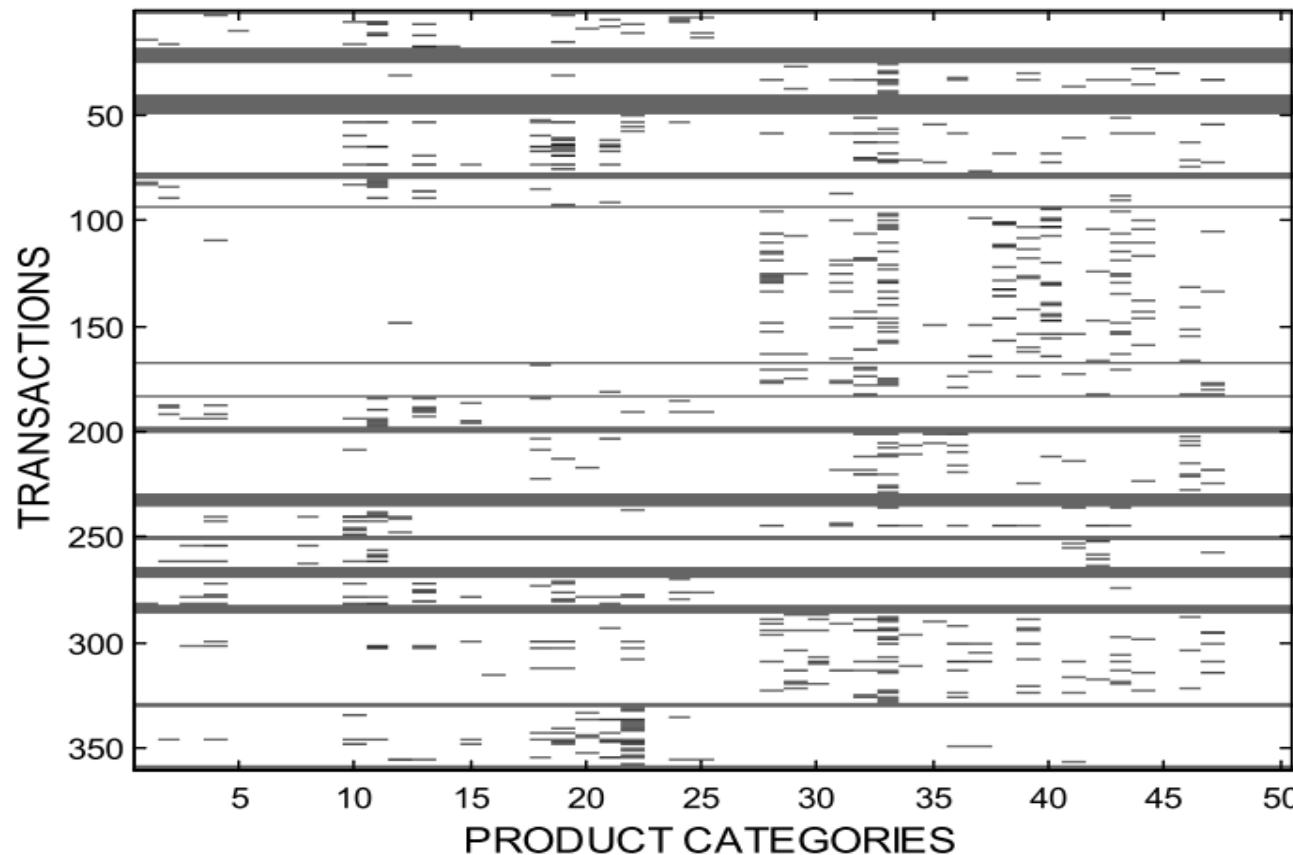
(x^i, y^i) ... trénovací vzorek

Measurements

Entities	ID	Income	Age	...	Monthly Debt	Good Risk?
	18276	65,000	55	2200	Yes
	72514	28,000	19	1500	No
	28163	120,000	62	1800	Yes
	17265	90,000	35	4500	No

	61524	35,000	22	900	Yes

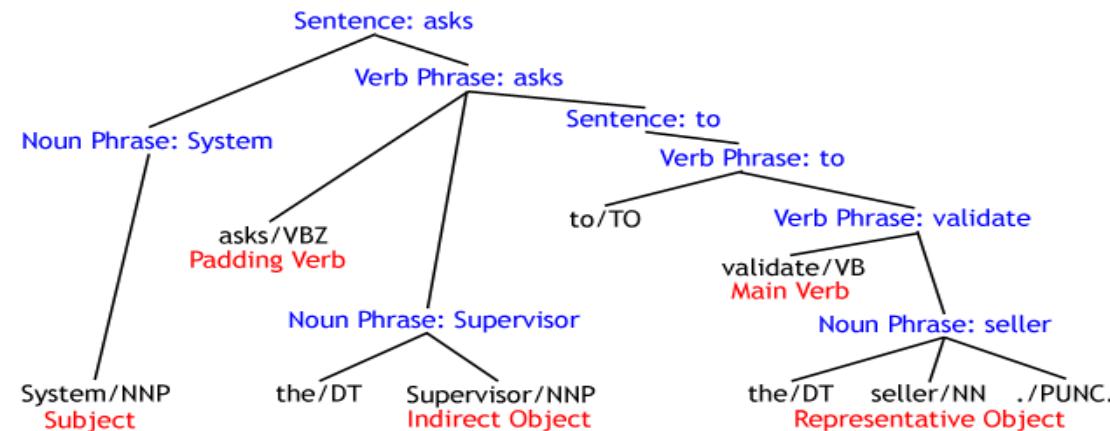
Vstupní data – Příklad dat



Vstupní data – Předzpracování dat

- Konkrétní kroky jsou závislé na povaze dat.
 - Strukturovaná x Nestrukturovaná podoba
 - Typ dat: obraz, řeč, text, data v relační databázi
 - Nestrukturovaná data je nutné při předzpracování transformovat do strukturované podoby.

„System asks the
Supervisor to
validate the seller.“



Vstupní data – Předzpracování dat – DB

- Nesmyslné nebo chybějící hodnoty u záznamů mohou negativním způsobem ovlivnit výsledky → čištění dat.
- Způsoby zacházení s chybějícími položkami:
 - Odstranit záznamy s chybějícími položkami
 - Odhad chybějících hodnot na základě známých hodnot
 - Použití algoritmů, které se dokážou vypořádat s chybějícími položkami
 - Náhodné hodnoty s využitím distribučního rozložení zbytku dat



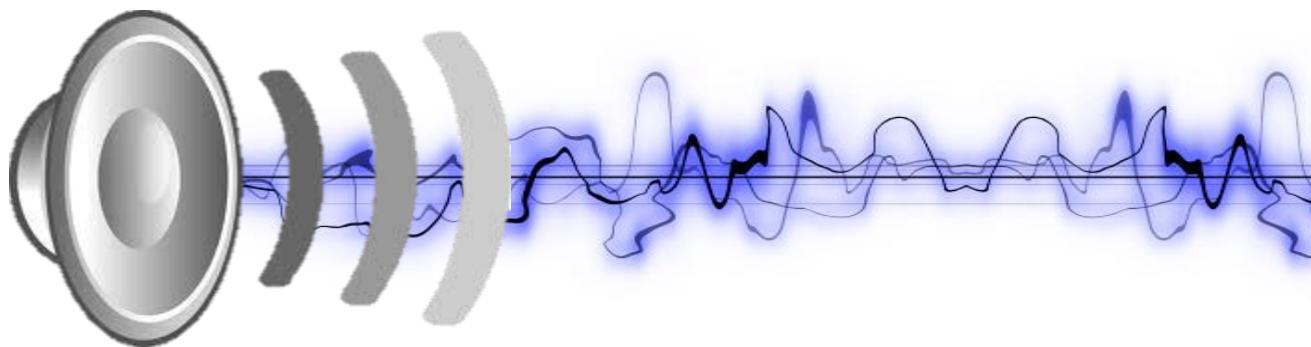
.. data cleaning ..

Vstupní data – Předzpracování dat – DB

- Vlivem slučování z několika zdrojů dochází k duplikaci záznamů → nutné tuto redundanci odstranit.
- Odstranění hodnot ležících mimo převážnou část zbylých hodnot → odstranění „outliers“
- Sumarizace dat:
 - Mean, Medián, Modus, Strmost, Šikmost...
 - Rozptyl, Směrodatná odchylka
- Odstranění redundance:
 - Např. atribut „Plat“: konkrétní částka může být převedena na platonou třídu (Integer→Byte).
 - Odstranění jednoho z atributů na základě korelačního koeficientu.

Vstupní data – Předzpracování dat – Řeč

- Filtrace HP/DP, Segmentace,..
- Extrakce příznaků: kmitočet základního tónu, počet průchodu nulou, Melovské spektrální/kepstrální koeficienty.
- **Příklady:** Identifikace řečníka, Speech to Text, Emoce

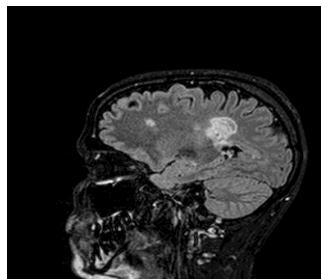


Poznámka – analýza řeči

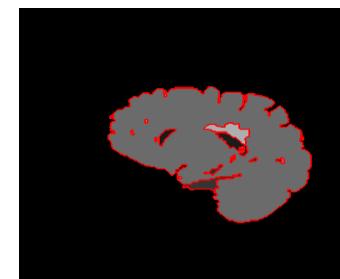
- Dřívější přístupy vyžadovaly matematické metody pro extrakci příznaků
- Dnešní stav vědy a techniky používá tzv. konvoluční neuronové sítě

Vstupní data – Předzpracování dat – Obraz

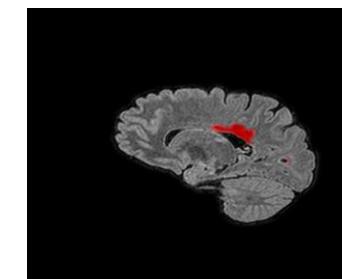
- Zvýraznění hran, zostření/rozostření detailů



Originál



Segmentace
&
Selekce



Rozpoznaná
nemoc



Ontologická
databáze

3D model

Vstupní data – Předzpracování dat – Obraz

- Rozpoznané objekty:
- Osoba
- Strom
- Země
- Zvíře

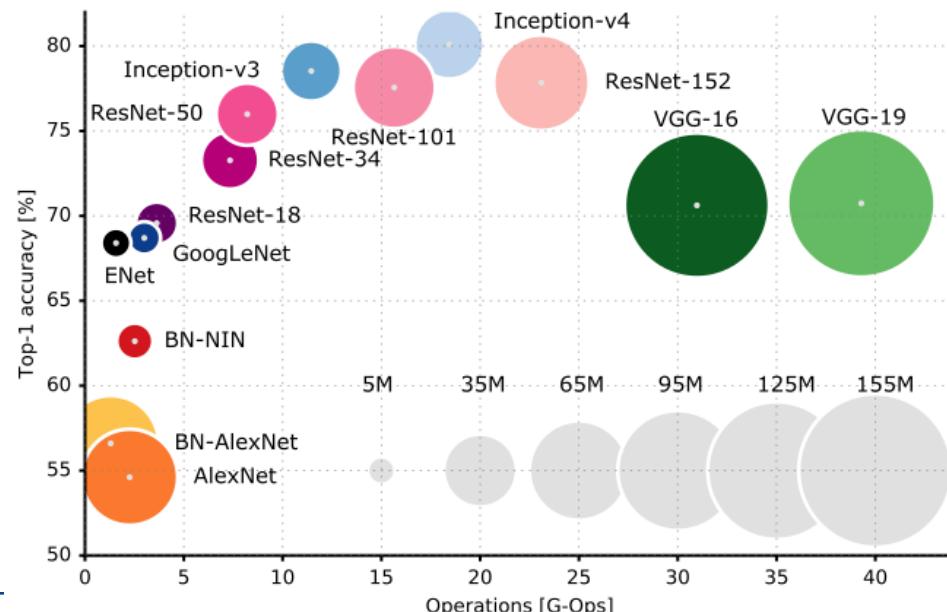
Kontextová znalost:

- Člověk ubližuje ještěrce



Poznámky – klasifikace obrazu

- V současné době jsou nejpoužívanější konvoluční neuronové sítě.
- Existují předtrénované sítě (2017):
 - VGG16
 - Resnet
 - LeNet
 - Xception
 - AlexNet
 - Inception v1, v2



Poznámky – detekce objektů v obrazu

- opět je základem konvoluční neuronová síť'
- mAP = Mean Average Precision, pro klasifikaci nad více třídami
- Nejlepší výsledky s pomocí sítí (2017):
- Výrazně přesnější nežli HOG, Viola Jones

Detection Framework	mAP	FPS
Faster RCNN - <u>VGG16</u>	73.2	7
Faster RCNN - ResNet	76.4	5
YOLO	63.4	45
SSD 500	76.8	19
YOLO v2 (<u>416x416</u> image size)	76.8	67
YOLO v2 (480x480 image size)	77.8	59

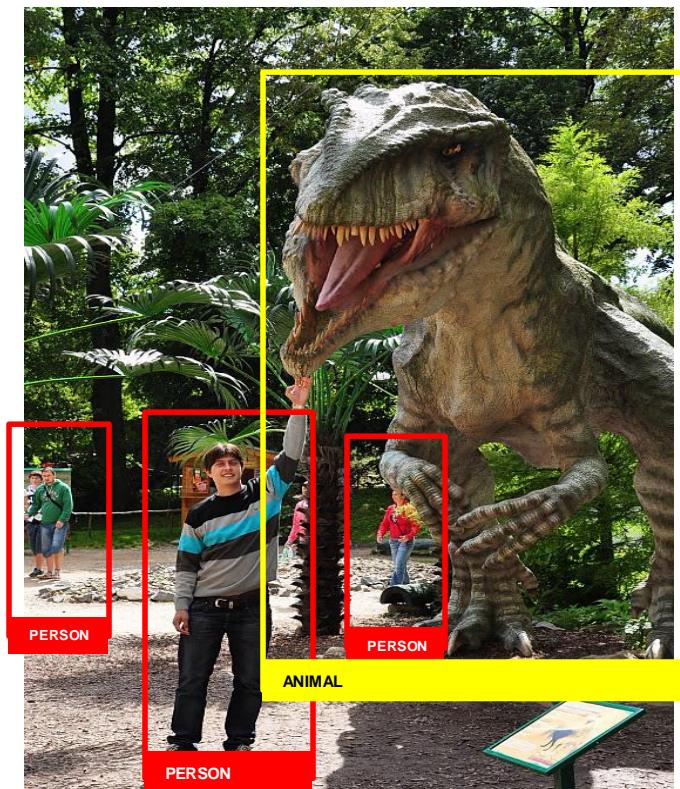
Vstupní data – Předzpracování dat – Video

Rozpoznané objekty a akce:

- Osoba – natahuje ruku
- Strom
- Země
- Zvíře – křik, útěk

Kontextová znalost:

- Člověk ubližuje ještěrce



Málo dat? – nutné ruční předzpracování

Nestrukturovaná data



Strukturovaná data

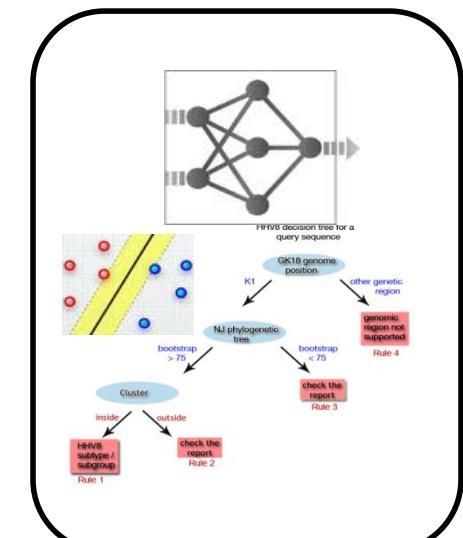


Datová
analýza

Učící se algoritmy

Dobré
výsledky

(relativně)



Cíl přednášky

1. Dolování znalostí (Data Mining)

- Motivace, Historie, Souvislosti
- Základy dolování znalostí
- Typické úlohy dolování
- Učení bez učitele
 - K-means
 - K-medoids
- Vstupní data
 - Základní pojmy
 - Sběr dat (struktura)
 - Předzpracování dat
 - Zpracování dat => ML

2. Strojové učení (Machine Learning)

- S učitelem
 - Typy dat, X-Validace, Srovnávání
 - Lineární a Logistická regrese
 - Rozhodovací stromy
 - Systém podpůrných vektorů
 - Neuronové sítě
 - Bayesovské sítě
 - k -NN

Motivace

- Předpovězte chování, abyste si udrželi zákazníky. Je možné předpovídat, který ze zákazníků pravděpodobně odejde?
- Soubor dat obsahuje informace o:
 - Zákazníci (7044x), kteří odešli během posledního měsíce (sloupec se nazývá Churn)
 - Služby, které si každý zákazník zaregistroval – telefon, více linek, internet, online zabezpečení, online zálohování, ochrana zařízení, technická podpora a streamování TV a filmů
 - Informace o zákaznickém účtu – jak dlouho je zákazníkem, smlouva, způsob platby, bezpapírová fakturace, měsíční poplatky a celkové poplatky
 - Demografické informace o zákaznících – pohlaví, věkové rozmezí a zda mají partnery a závislé osoby

Jaká je přesnost daného modelu? S jakou přesností?

(79,64%)

Jaké vlastnosti jsou charakteristické?

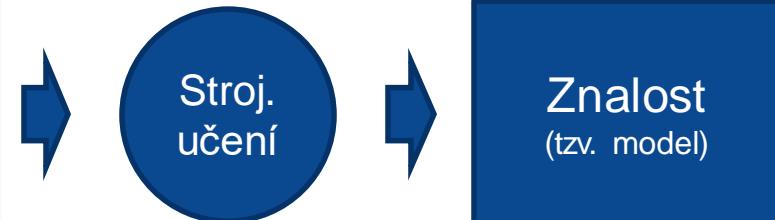
Jsou nějaké charakteristické skupiny uživatelů, kterým můžeme nabídnout speciální tarif?

- https://raw.githubusercontent.com/Saad20x/Telco-Customer-Churn/main/WA_Fn-UseC_-Telco-Customer-Churn.csv

Motivace

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79

Data
(mohou být i texty, audio, video, obraz, ...)



Algoritmy

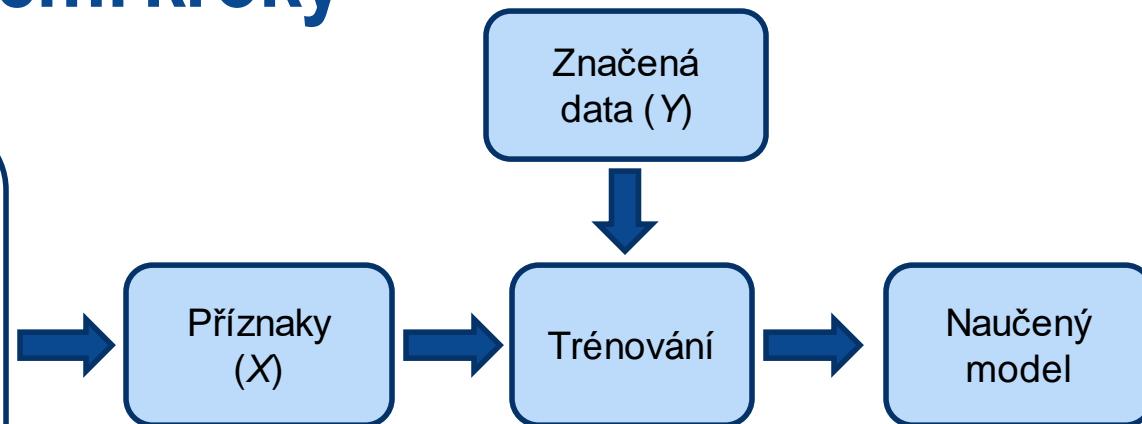
Znalost
reprezentovaná
datovými
strukturami

Jakými algoritmy vytvořit
z dat znalost?

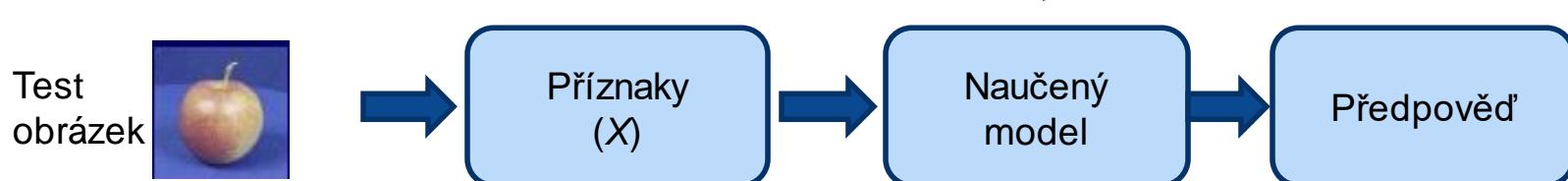
Jak reprezentovat znalost?
(nejen lineární, strom., graf, ale jak konkrétně mají vypadat)

Učení s učitelem: kroky

Trénování



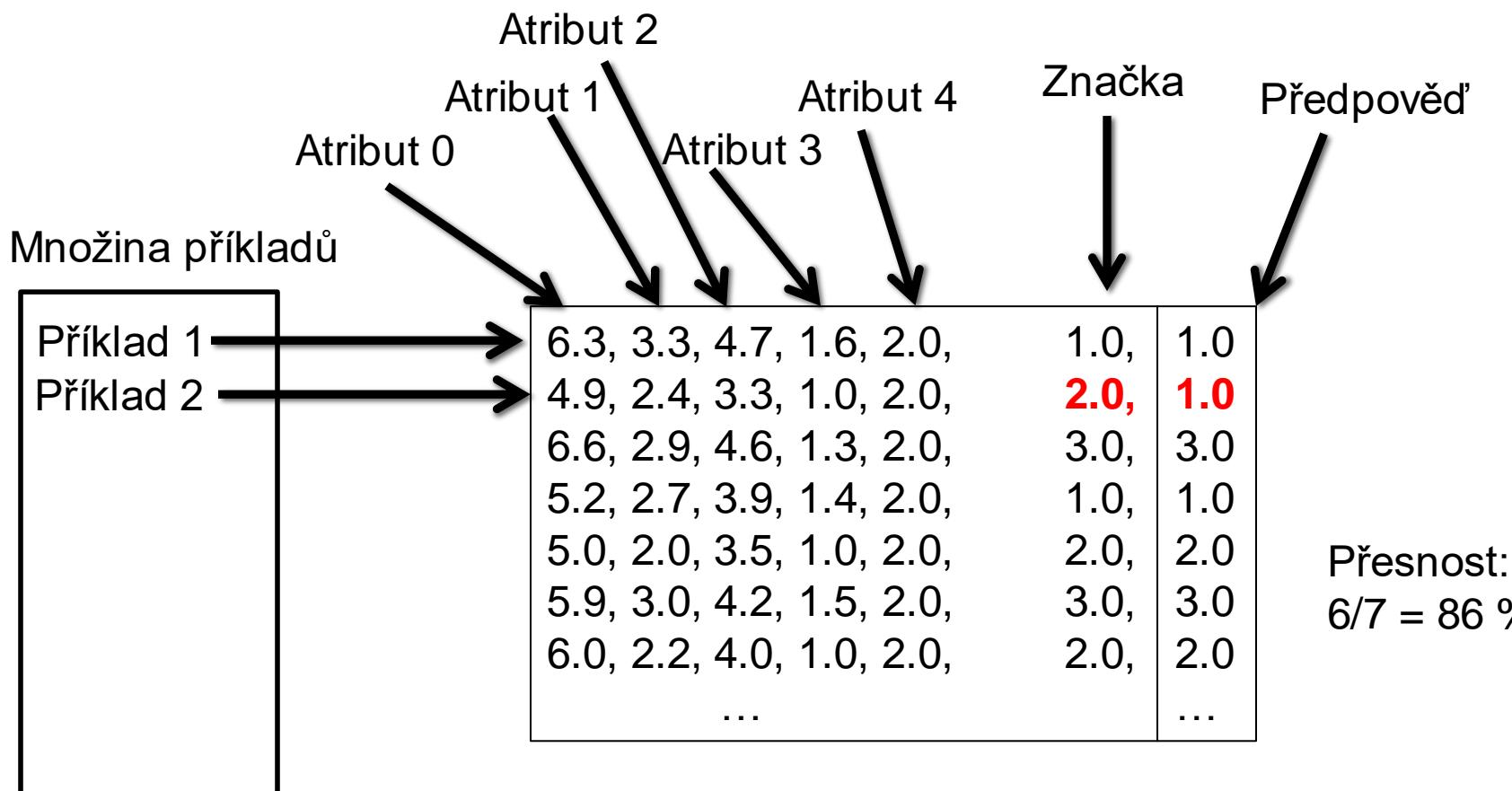
Testing



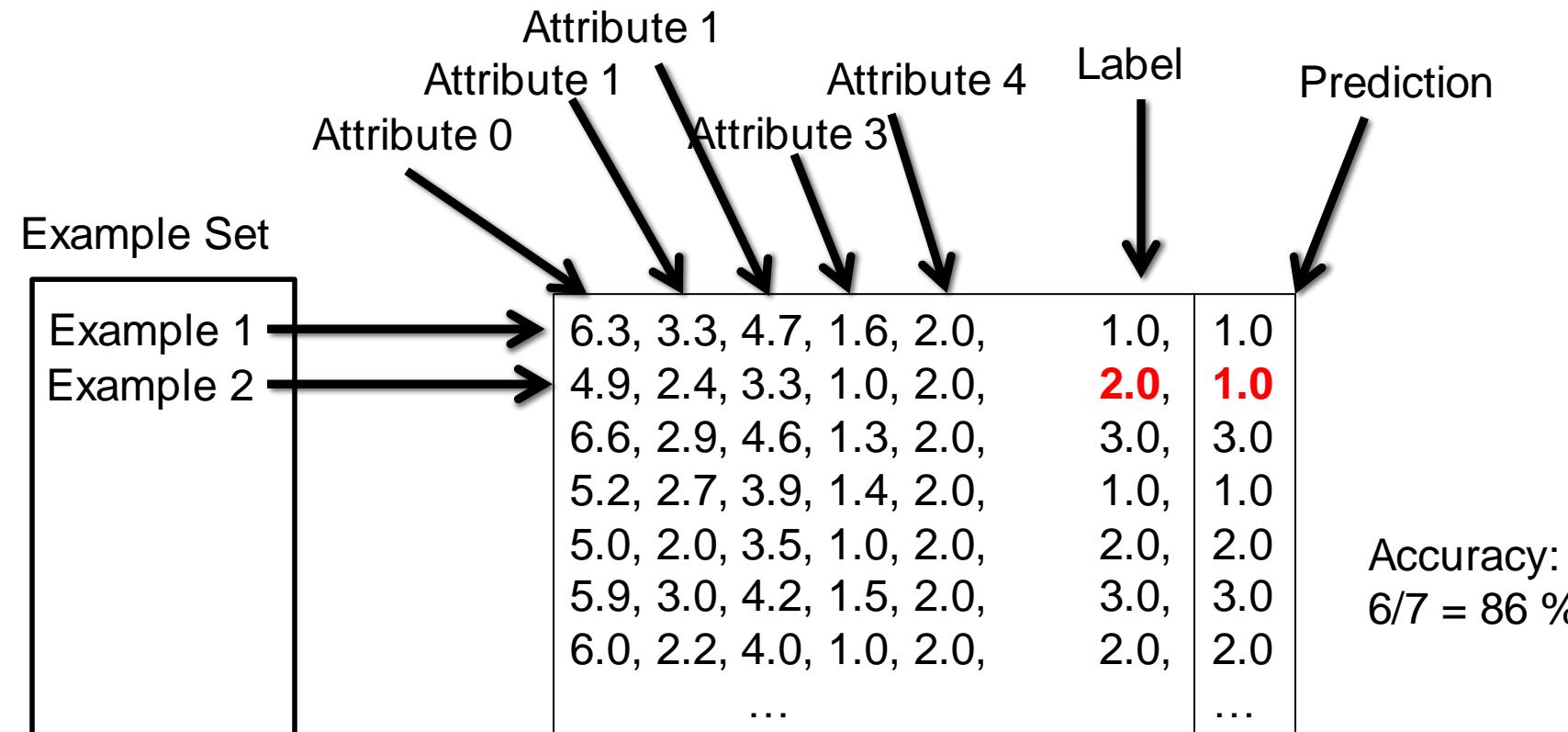
Strojové učení

- **Učení s učitelem (supervised learning)**
regrese, klasifikace
- Učení bez učitele (unsupervised learning)
shlukování
- Kombinované učení (semi-supervised learning)
- Zpětnovazební učení (reinforcement learning)
(nebude probíráno)

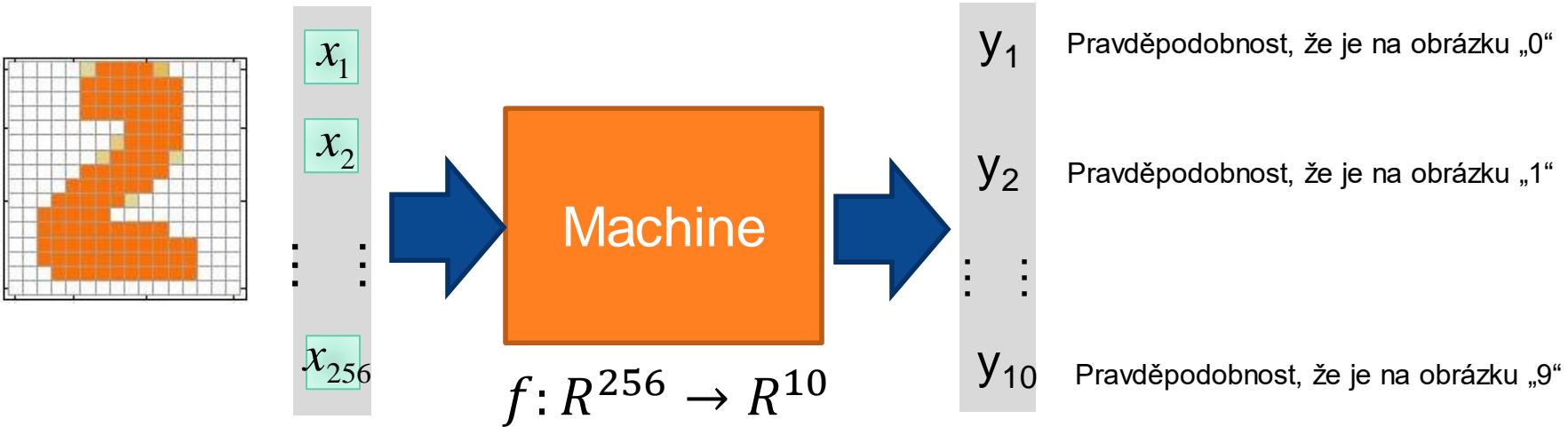
Základní terminologie



Terminologie – Anglicky

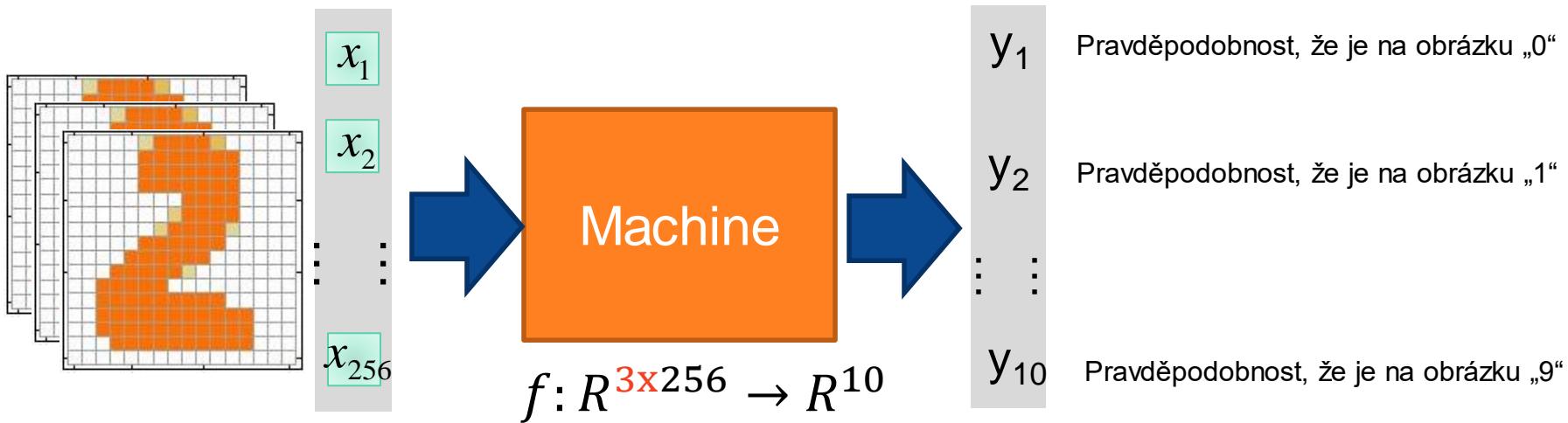


Příklad: Klasifikace obrazu



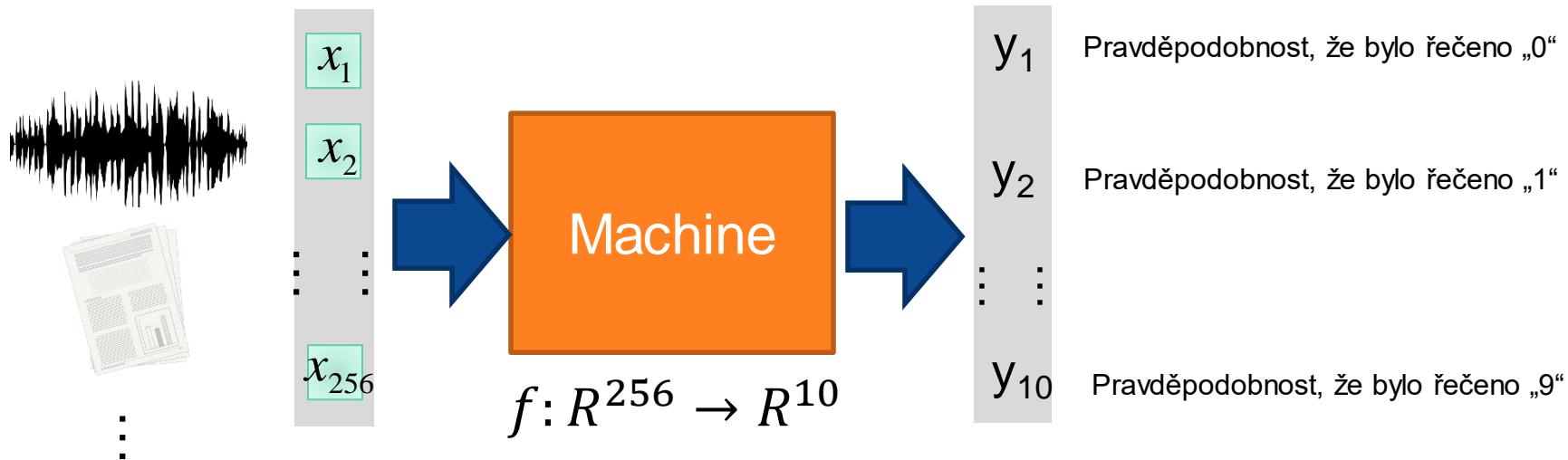
Pozn.: Jestliže je odezva funkce $[0, 0.9, 0.1, 0, 0.2, 0, 0.1, 0, 0, 0]$, jaké je předpovězeno číslo?

Příklad: Klasifikace videa



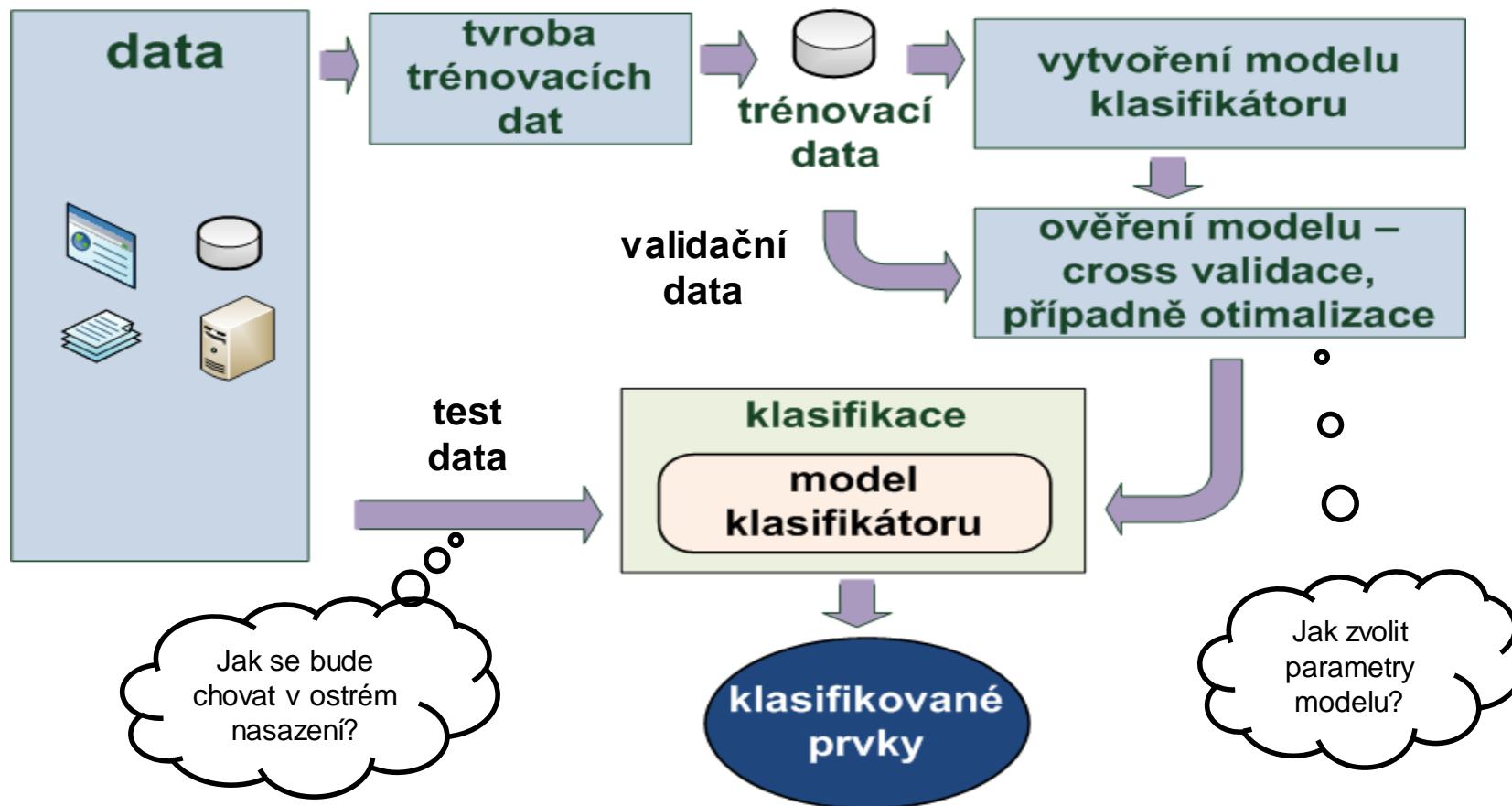
Pozn.: Jestliže je odezva funkce $[0, 0.9, 0.1, 0, 0.2, 0, 0.1, 0, 0, 0]$, jaké je předpovězeno číslo?

Příklad: Klasifikace audia



Pozn.: Jestliže je odezva funkce $[0, 0.9, 0.1, 0, 0.2, 0, 0.1, 0, 0, 0]$, jaké je předpovězeno číslo?

Strojové učení s učitelem



Demo: rozdělení dat

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=0)
```

Testovací množina: 33%, z toho vyplývá, že trénovací 67%



Strojové učení s učitelem – Typy dat

- **Trénovací data**
 - Slouží pro vytvoření („nátrénování“) modelu klasifikátoru
 - Způsob tvorby modelu je závislý na konkrétním algoritmu strojového učení (SVM, k-NN apod.)
- **Validační data**
 - Slouží pro ověření modelu klasifikátoru a případnou optimalizaci (doladění parametrů).
 - Mohou být vytvořeny z trénovacích dat (ověření pomocí Cross-validation) nebo mohou být zcela nezávislá.
- **Testovací data**
 - Pro získání skutečné výkonosti modelu (reálné chování).
 - **Nesmí mít žádný podíl na tvorbě modelu.**

Učení s učitelem: ověření



Training set (labels known)



Test set (labels unknown)

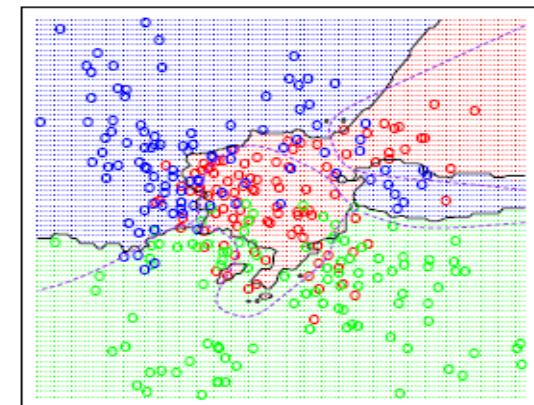
- Jak je model schopen zobecňovat naučená pravidla?
Vždy je nutné testovat na dosud neviděných vzorcích!

Strojové učení s učitelem – ověření přesnosti

Ověření kvality
naučeného modelu musí
být provedeno na
datech nezávislých na
trénovací množině!

Strojové učení s učitelem – Trénovací data

- Klíčová součást metod strojového učení s učitelem
- Kvalita modelu = kvalita trénovacích dat
- Parametry „ideálních“ trénovacích dat jsou závislé na povaze konkrétního problému.
- Rozhodující faktory:
 - Množství
 - Míra šumu
 - Rozměr (dimenze)
 - Vzorky zcela mimo (outliers)
 - Poměrné zastoupení jednotlivých tříd



Křížová validace (Cross-validation)

- Pro ohodnocení vytvořeného modelu
- Data určená pro trénování jsou současně daty testovacími
- Rozdělení do n-skupin (n-1 pro trénovaní, 1 pro testování).
- Opakujeme n-krát (s postupnou výměnou skupiny pro testování), výsledky jednotlivých průměrujeme



Příklad: Rozdělte pomocí 3-složkové křížové validace

ID	Příjmy	Závazky	Půjčit
1	32t	11t	ANO
2	24t	17t	NE
3	44t	15t	ANO
4	36t	30t	NE
5	19t	0t	ANO
6	66t	49t	NE
7	51t	6t	ANO
8	37t	38t	NE
9	22t	6t	ANO



ID	Příjmy	Závazky	Půjčit
7	51t	6t	ANO
8	37t	38t	NE
9	22t	6t	ANO

ID	Příjmy	Závazky	Půjčit
1	32t	11t	ANO
2	24t	17t	NE
3	44t	15t	ANO

ID	Příjmy	Závazky	Půjčit
4	36t	30t	NE
5	19t	0t	ANO
6	66t	49t	NE

Krok 1:
rozděl na 3
složky

(zpravidla
náhodně)

Příklad: Rozdělte pomocí 3-složkové křížové validace

ID	Příjmy	Závazky	Půjčit
1	32t	11t	ANO
2	24t	17t	NE
3	44t	15t	ANO
4	36t	30t	NE
5	19t	0t	ANO
6	66t	49t	NE



ID	Příjmy	Závazky	Půjčit
1	32t	11t	ANO
2	24t	17t	NE
3	44t	15t	ANO
7	51t	6t	ANO
8	37t	38t	NE
9	22t	6t	ANO



ID	Příjmy	Závazky	Půjčit
4	36t	30t	NE
5	19t	0t	ANO
6	66t	49t	NE
7	51t	6t	ANO
8	37t	38t	NE
9	22t	6t	ANO



ID	Příjmy	Závazky	Půjčit
7	51t	6t	ANO
8	37t	38t	NE
9	22t	6t	ANO

ID	Příjmy	Závazky	Půjčit
4	36t	30t	NE
5	19t	0t	ANO
6	66t	49t	NE

ID	Příjmy	Závazky	Půjčit
1	32t	11t	ANO
2	24t	17t	NE
3	44t	15t	ANO

Trénování

Testování

Příklad: Rozdělte pomocí 3-složkové křížové validace

Střední hodnota

 \bar{x}

Směrodatná odchylka

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Výsledná přesnost = $90 \pm 4.0825\%$

Přesnost1 = 90%

Přesnost2 = 95%

Přesnost3 = 85%

Model č. 1

validace

Model č. 2

validace

Model č. 3

validace

ID	Příjmy	Závazky	Půjčit
7	51t	6t	ANO
8	37t	38t	NE
9	22t	6t	ANO

ID	Příjmy	Závazky	Půjčit
4	36t	30t	NE
5	19t	0t	ANO
6	66t	49t	NE

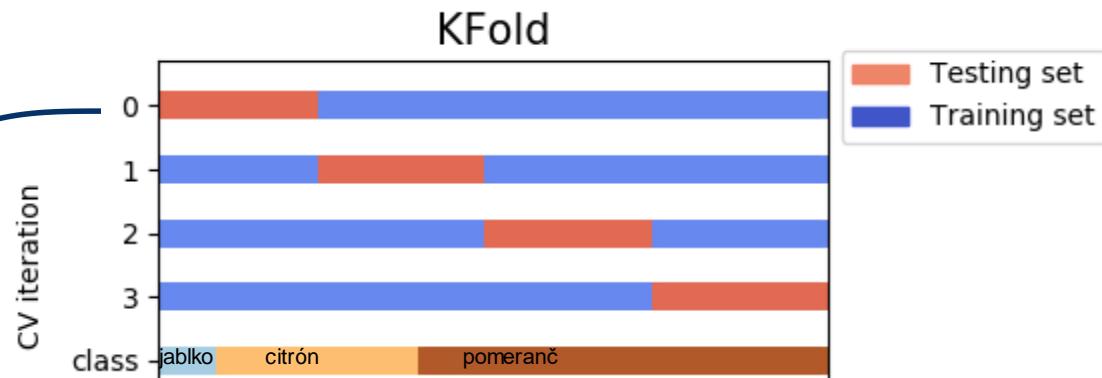
ID	Příjmy	Závazky	Půjčit
1	32t	11t	ANO
2	24t	17t	NE
3	44t	15t	ANO

Testování

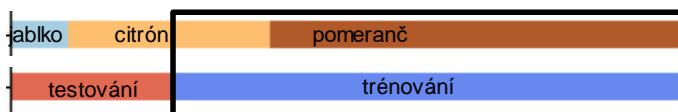
Otázky

- K čemu mi jsou tři modely, nestačí jeden?
 - K ničemu, u křížové validace mi jde o předpověď přesnosti
- Jaká je nevýhoda?
 - Čas strávený se násobí počtem složek (např. 10-složková = 10x čas trénování)
- Bez použití křížové validace, bylo by ověření jen na 3 vzorcích dostatečně vypovídající?
- Jaká je výhoda?
 - Pokud mám nedostatek dat, tak z nich dokážu předpovědět spolehlivost výrazně přesněji, použita jsou veškerá data, za cenu vyšší výpočetní náročnosti

Křížová validace: možný problém



Zdroj: https://scikit-learn.org/stable/modules/cross_validation.html



Může se z 1. iterace naučit rozpoznávat jablka?

Promítně se do hodnocení schopnost klasifikovat pomeranče?

Řešení: znáhodnění pořadí vzorků
(shuffling)

Zdroj: <https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>
kapitola 5

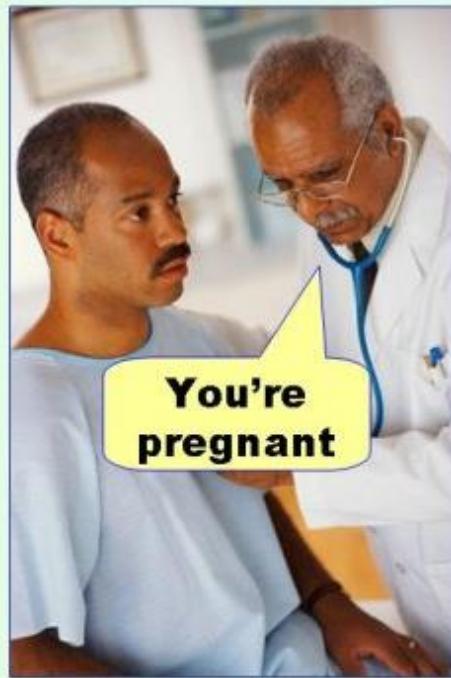
Strojové učení – Srovnání učících se algoritmů

- Matice záměn (confusion matrix).
A: TP (true positive), B: FN (false negative)
C: FP (false positive), D: TN (true negative)

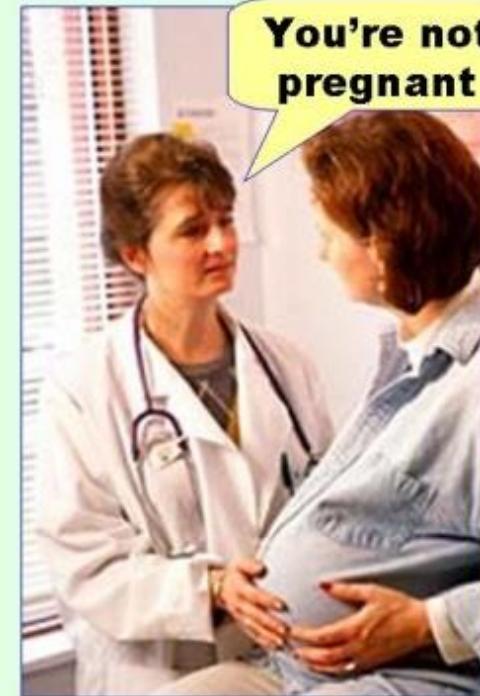
		Předpověděná třída	
Skutečná třída		Class=Ano	Class=Ne
Skutečná třída	Class=Ano	A (TP)	B (FN)
	Class=Ne	C (FP)	D (TN)

Příklad: FN, FP

(false positive)



(false negative)



Strojové učení – Srovnání učících se algoritmů

Z matice záměn je potom možné vyjádřit mnoho parametrů.

$$\textbf{Přesnost} (\text{accuracy}, A) = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\textbf{Specifičnost} (\text{specificity}, S) = \frac{TN}{FP+TN}$$

$$\textbf{Senzitivita} (\text{recall}, R) = \frac{TP}{TP+FN}$$

Strojové učení – Srovnání učících se algoritmů

Youden:

$$Y = \text{Senzitivita} + \text{Specifičnost} - 1$$

Pozitivní předpovídající hodnota:

$$PPV = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

Negativní předpovídající hodnota:

$$NPV = \frac{N_{TN}}{N_{TN} + N_{FN}}$$

Strojové učení – Srovnání učících se algoritmů

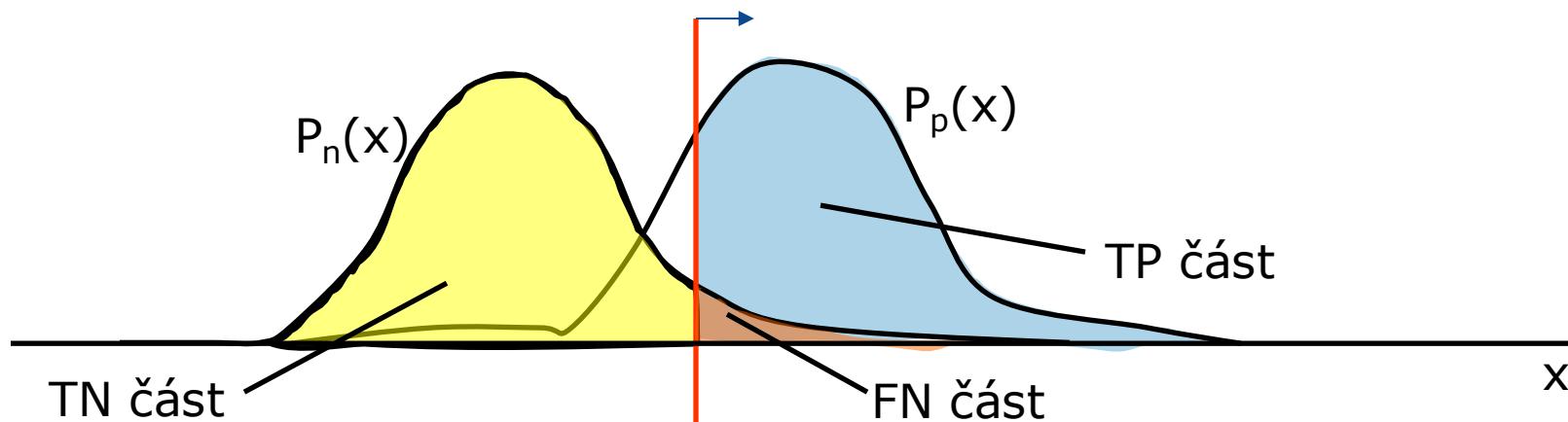
F1 – míra (F-measure, F-score):

$$F_{\text{SCORE}} = \frac{2a}{2a+b+c} \quad (\text{kompromis mezi precision a recall})$$

Mathews korelační koeficient:

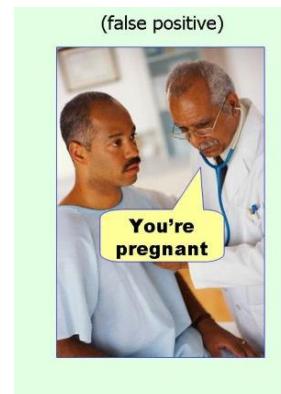
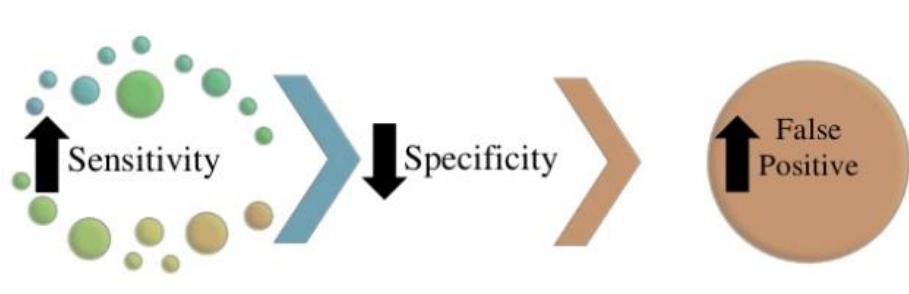
$$MCC = \frac{N_{\text{TP}} \times N_{\text{TN}} - N_{\text{FP}} \times N_{\text{FN}}}{\sqrt{(N_{\text{TP}} + N_{\text{FP}})(N_{\text{TP}} + N_{\text{FN}})(N_{\text{TN}} + N_{\text{FP}})(N_{\text{TN}} + N_{\text{FN}})}}$$

Sensitivita vs. specificita

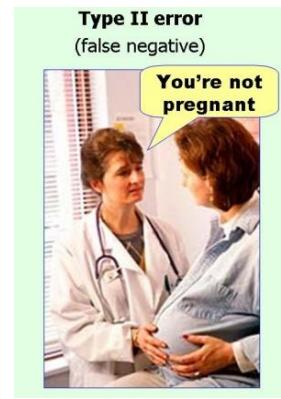
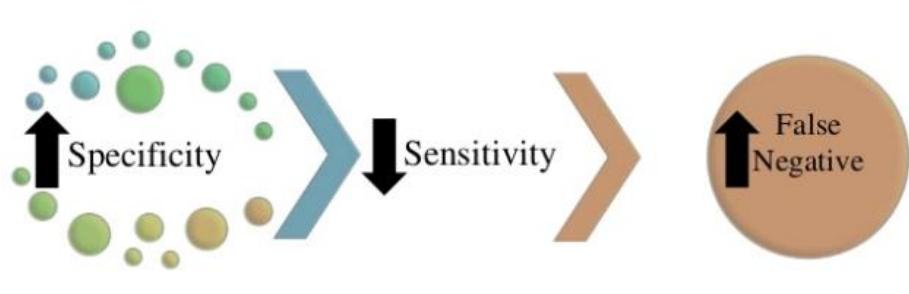


- V mnoha případech není snadné separovat
- Posunutím prahu vpravo zvýšíme specifičnost, tj. zachytíme více TN příkladů („netěhotných žen“), na druhou stranu snížíme specificitu, tj. ve více případech se bude jednat falešně pozitivní (FP = „těhotných mužů“) předpovědi.

Příklad: Sensitivita a specificita jdou proti sobě



Na druhou stranu bude méně FN, tj. těhotných, nerozpoznaných



Na druhou stranu bude méně FP, tj. těhotných, nesprávně označených

Příklad

- S pomocí neuronové sítě byl natrénován model, který předpověděl ve 133 případech, že se jedná o pacienta s onemocněním, z toho ve 33 případech chybně. Dále ve 105, že nejedná o onemocnění, z toho ve 12 případech chybně.
- Vytvořte matici záměn a spočtěte parametry sensitivita, specificita, přesnost, f1-míra a další

Řešení:

	Skutečnost	
Předpověď	Ano	Ne
Ano	100	12
Ne	33	93

Measure	Value	Derivations
Sensitivity	0.7519	$TPR = TP / (TP + FN)$
Specificity	0.8857	$SPC = TN / (FP + TN)$
Precision	0.8929	$PPV = TP / (TP + FP)$
Negative Predictive Value	0.7381	$NPV = TN / (TN + FN)$
False Positive Rate	0.1143	$FPR = FP / (FP + TN)$
False Discovery Rate	0.1071	$FDR = FP / (FP + TP)$
False Negative Rate	0.2481	$FNR = FN / (FN + TP)$
Accuracy	0.8109	$ACC = (TP + TN) / (P + N)$
F1 Score	0.8163	$F1 = 2TP / (2TP + FP + FN)$
Matthews Correlation Coefficient	0.6343	$TP \cdot TN - FP \cdot FN / \sqrt{((TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN))}$

Příklad pokračování

- Byl vytvořen nový model založený na rozhodovacích stromech, který nemoc správně rozpoznal v 108 případech, v 90 případech určil, že se jedná o zdravé pacienty. Naopak v 25 případech nebyla nemoc u pacienta rozpoznána a 15 bylo nesprávně označeno, že jsou nemocní.
- Který model je lepší? Který parametr porovnání použijeme, chceme-li minimalizovat případy, kde není nemoc rozpoznána?

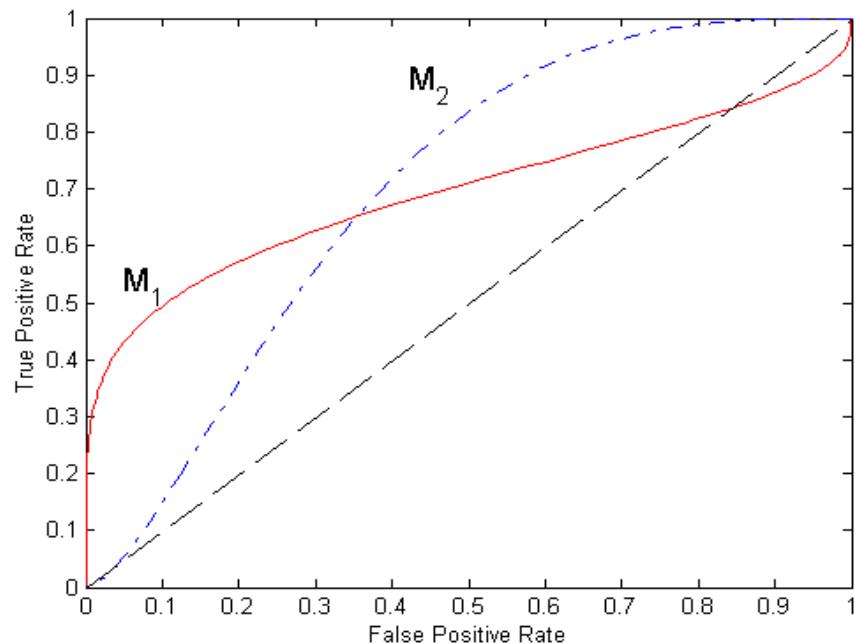
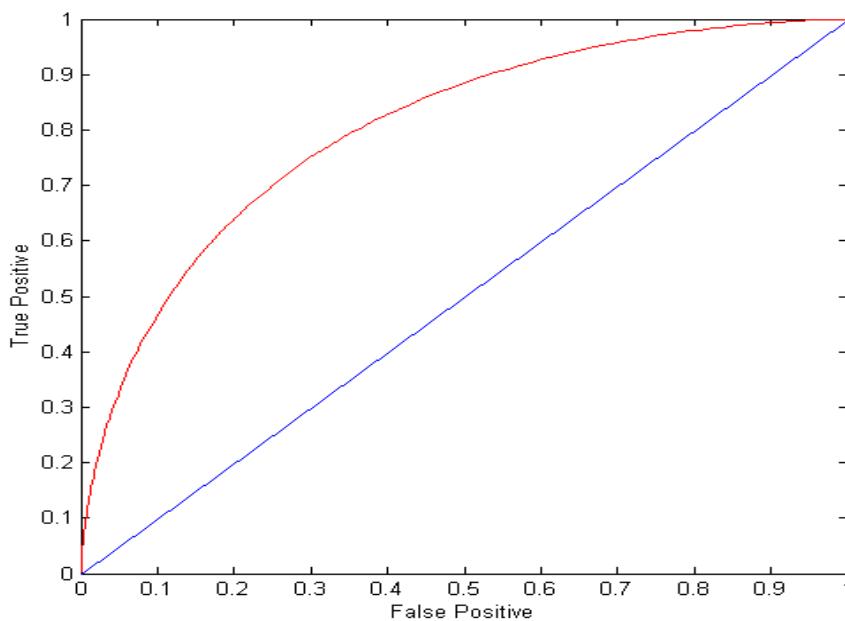
		Skutečnost	
		Ano	Ne
Předpověď	Ano	108	15
	Ne	25	90

	Skutečnost	
Předpověď	Ano	Ne
Ano	108 TP	15 FN
Ne	25 FP	90 TN

Measure	Value	Derivations	Value
Sensitivity	0.8120	$TPR = TP / (TP + FN)$	0.7519
Specificity	0.8571	$SPC = TN / (FP + TN)$	0.8857
Precision	0.8780	$PPV = TP / (TP + FP)$	0.8929
Negative Predictive Value	0.7826	$NPV = TN / (TN + FN)$	0.7381
False Positive Rate	0.1429	$FPR = FP / (FP + TN)$	0.1143
False Discovery Rate	0.1220	$FDR = FP / (FP + TP)$	0.1071
False Negative Rate	0.1880	$FNR = FN / (FN + TP)$	0.2481
Accuracy	0.8319	$ACC = (TP + TN) / (P + N)$	0.8109
F1 Score	0.8438	$F1 = 2TP / (2TP + FP + FN)$	0.8163
Matthews Correlation Coefficient	0.6649	$TP \cdot TN - FP \cdot FN / \sqrt{((TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN))}$	0.6343

Strojové učení – Srovnání učících se algoritmů

- ROC křivky – poměr mezi TP a FP (Receiver Operating Characteristic)



Algoritmy strojového učení

Učení s učitelem

Strojové učení

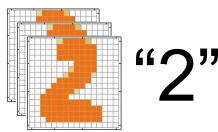
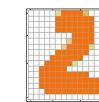
- Jedná se o poskupinu oblasti s názvem „umělá inteligence“
- Zabýváme se skupinou, která je mapující funkcí:

$$f(R^m) \rightarrow R^n$$

R^8								R^2	
# těhot.	Glukóz.	Tlak	Triceps	Inzulín	BMI	Dědič.	Věk	Ano	Ne
3	74	68	28	45	29.7	0.293	23	0,9	0,1

Základní koncept

ID	Příjmy	Závazky	Půjčit
7	51t	6t	ANO
8	37t	38t	NE
9	22t	6t	ANO

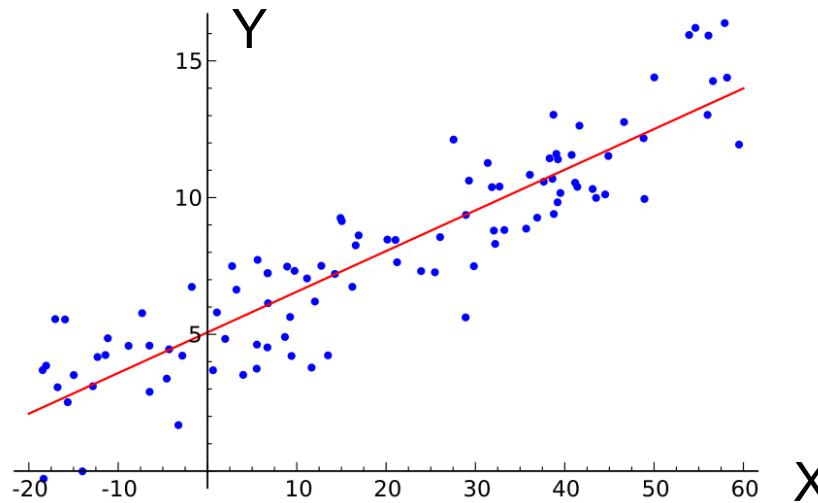


- Rychlé
- 24/7

Značení dat je:

- Zdlouhavé
- Finančně nákladné

Jednoduchá lineární regrese



Předpověď
modelu (skalár) Vstupní
proměnná (skalár)

Lineární model: $Y = mX + b$

úhel
(konst., skalár) odstup (bias)
(konst. ,skalár)

Lineární model (zobecnění pro více dimenzí)

Skalární
odezva

$$y = \theta^T x + \epsilon$$

Vektor
parametrů

Vektor
proměnných

Skalární
šum

+ b

Lineární kombinace
vstupních proměnných

$$\sum_{i=1}^p \theta_i x_i$$

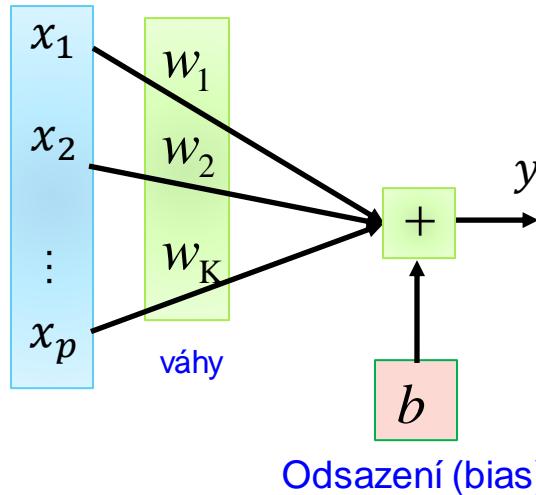
A co odsazení (bias)?

Model šumu:

$$\epsilon \sim N(0, \sigma^2)$$

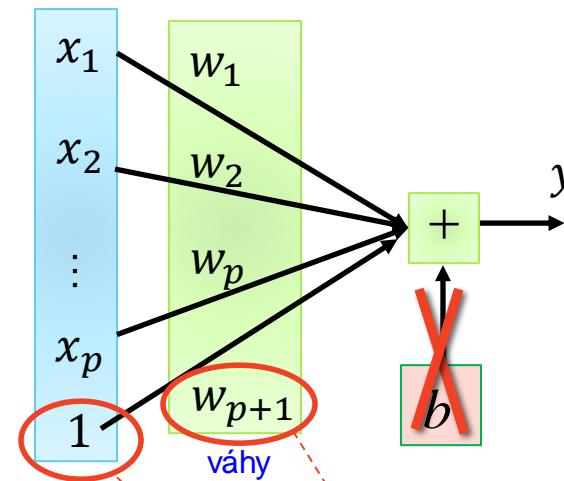
Raději definujme $x_{p+1} = 1$
A p nastavme na $p = p + 1$

Intuitivní metoda výpočtu



$$y = \theta^T x + b$$

Používaná (zjednodušená)



$$y = \theta'^T x'$$

POZN.: Proč je b potřeba? Mohlo by bez něj ze vstupu $[0,0,\dots,0]$ vzniknout něco jiného než 0?

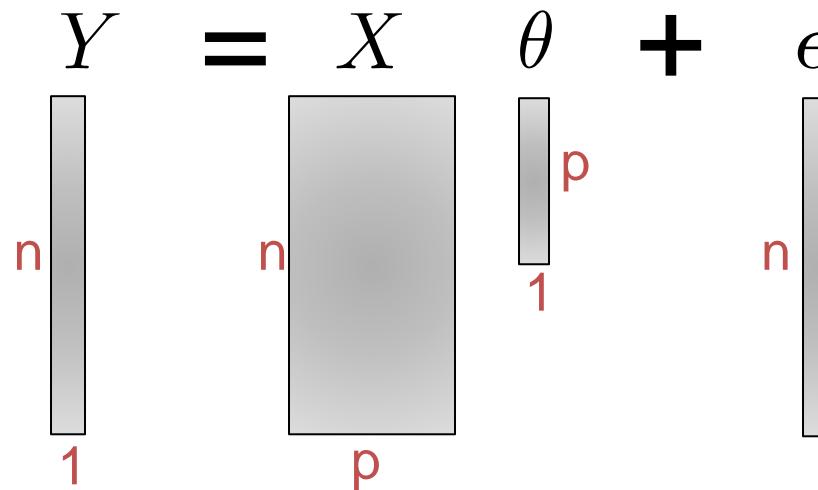
Lineární regrese s pomocí matic



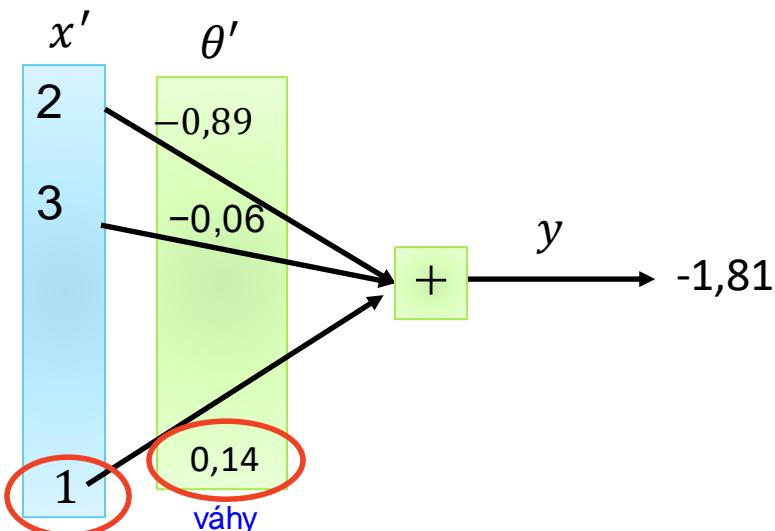
- Předchozí operaci zapišme s pomocí násobení matic:

$$Y = X\theta + \epsilon$$

p...počet dimenzí příznaku
 n...počet vzorků
 na vstupu



Příklad



$$y = \theta'^T x' = x'^T \theta' =$$

x_1	x_2	b
2,00	3,00	1,00

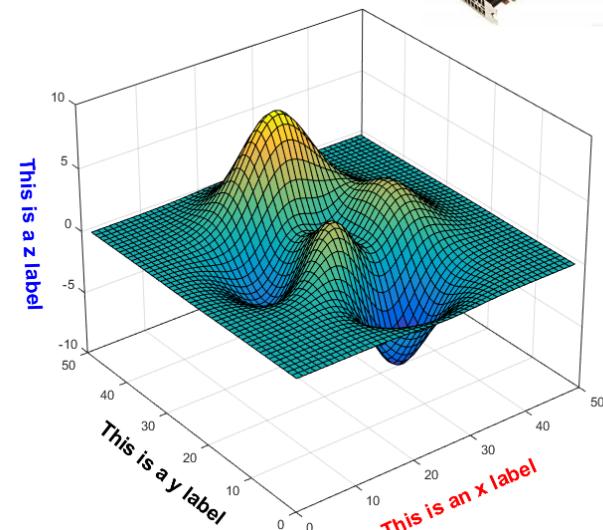
w_1	-0,89
w_2	-0,06
w_3	0,14

$$= (-1,78 - 0,17 + 0,14) = -1,81$$

$$x'^T \quad \theta'$$

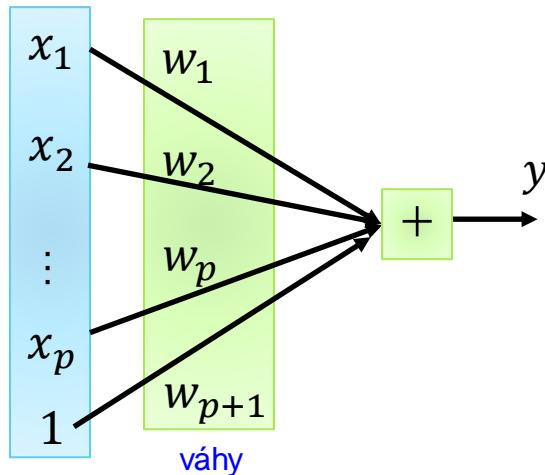
Pozn: Které parametry jsou při trénování lineární regrese upravovány?

Odpověď: θ na základě trénovačích dat x



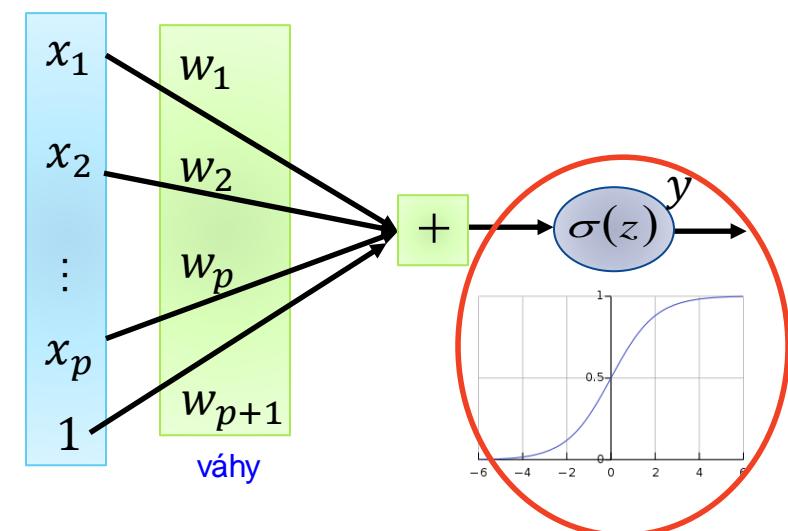
Strojové učení – Logistická regrese

Lineární regrese



regrese

Logistická regrese



klasifikace

Strojové učení – Lineární regrese

- $h_{\Theta}(x)$ = hypotéza, Θ = parametry

$$h_{\Theta}(x) = \Theta_0 + \Theta_1 x$$

- Θ se volí tak, aby $h_{\Theta}(x)$ bylo co nejblíže y pro trénovací vzorek (x, y)

$$J(\Theta_0, \Theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\Theta}(x^i) - y^i)^2$$

- Co nejblíže = Jde o minimalizační problém

$$\min_{\Theta_0, \Theta_1} J(\Theta_0, \Theta_1)$$

Strojové učení – Lineární regrese

Gradient Descent algoritmus pro minimalizaci Θ parametrů

- Začneme s libovolnými Θ parametry a redukujeme je skrze $J(\Theta_0, \Theta_1)$ dokud není dosaženo minima funkce

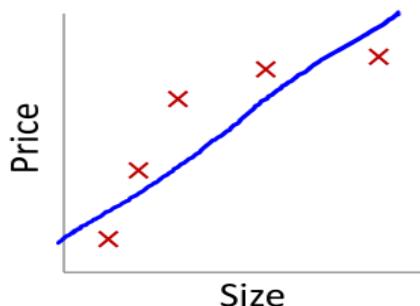
$$\Theta_j = \Theta_j - \alpha \frac{\delta}{\delta \Theta_j} J(\Theta_0, \Theta_1)$$

Derivační členy jsou: (MUSÍ JÍT O SIMULTÁNNÍ AKTUALIZACI)

$$\Theta_0 = \frac{1}{m} \sum_1^m (h_\Theta(x^i) - y^i)$$

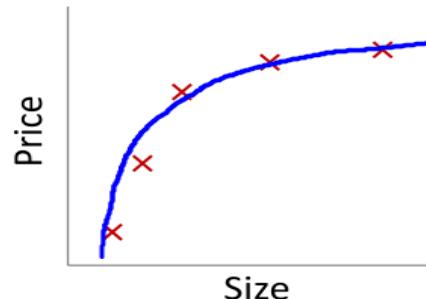
$$\Theta_1 = \frac{1}{m} \sum_1^m (h_\Theta(x^i) - y^i) x^i$$

Strojové učení – Lineární regrese



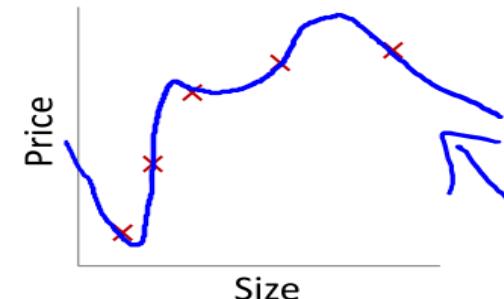
$$\rightarrow \theta_0 + \theta_1 x$$

"Underfit" "High bias"



$$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$$

"Just right"



$$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

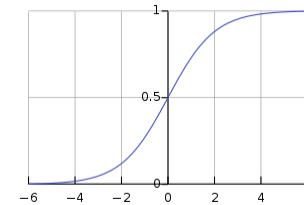
"Overfit" "High variance"

Strojové učení – Logistická regrese

- Nejjednodušší forma: binární klasifikace (ano vs. ne)
- Chceme: $0 < h_{\Theta}(x) < 1$

$$h_{\Theta}(x) = g(\theta_0 + \theta_1 x) \text{ a } g(z) = \frac{1}{1+e^{-z}}$$

$$J(\Theta_0, \Theta_1) = \frac{1}{m} \sum_1^m cost \left((h_{\Theta}^i), y^i \right)$$



Pokud $y = 1$ potom $cost \left((h_{\Theta}^i), y^i \right) = -\log(h_{\Theta}(x))$

Pokud $y = 0$ potom $cost \left((h_{\Theta}^i), y^i \right) = -\log(1 - h_{\Theta}(x))$

$$cost \left((h_{\Theta}^i), y^i \right) = (y) * \left(-\log \left(h_{\Theta}(x) \right) \right) + (1 - y) * \left(-\log(1 - h_{\Theta}(x)) \right)$$

Logistická Regrese – demo

fruit_name	fruit_subtype	mass	width	height	color_score
apple	granny_smith	192	8.4	7.3	0.55
apple	granny_smith	180	8.0	6.8	0.59
apple	granny_smith	176	7.4	7.2	0.60
mandarin	mandarin	86	6.2	4.7	0.80

Trénovací data



```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

Učení

O jaké se jedná ovoce?
(znám rozměry, barvu a váhu)

fruit_name	fruit_subtype	mass	width	height	color_score
????		84	6.0	4.6	0.79

<https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>
Kapitola 6.1)

MTIN I: cz.vutbr.feec.mtin.w08.logistic

Rozhodovací stromy

- Snadná interpretovatelnost člověkem, dle čeho se rozhoduje
- Často nižší přesnost než jiné přístupy
- Rozdělení:
 - Rozhodovací stromy
 - Rozhodovací lesy (= kombinace několika stromů a hlasování o výsledném stanovisku)

Klasifikační problém - příklad

Název	Teplota těla	Pokrytí těla	Živorodí	Vodní	Létaví	Má nohy	Hibernuje	Druh
Člověk	Teplo	srst	ano	ne	ne	ano	ne	Savec
Krajta	Studeno	scales	ne	ne	ne	ne	ano	Plaz
Losos	Studeno	scales	ne	ano	ne	ne	ne	Ryba
Velryba	Teplo	hair	ano	ano	ne	ne	ne	Savec
Žába	Studeno	none	ne	semi	ne	ano	ano	Obojživelník
Komodo	Studeno	scales	ne	ne	ne	ano	ne	Plaz
Netopýr	Teplo	hair	ano	ne	ano	ano	ano	Savec
Holub	Teplo	feathers	ne	ne	ano	ano	ne	Pták
Kočka	Teplo	fur	ano	ne	ne	ano	ne	Savec
Leopard	Studeno	scales	ano	ano	ne	ne	ne	Ryba
Želva	Studeno	scales	ne	semi	ne	ano	ne	Plaz
Tučňák	Teplo	feathers	ne	semi	ne	ano	ne	Pták
Dikobraz	Teplo	quills	ano	ne	ne	ano	ano	Savec
Úhoř	Studeno	scales	ne	ano	ne	ne	ne	Ryba
Mlok	Studeno	none	ne	semi	ne	ano	ano	Obojživelník

Kam by dle těchto pravidel patřil například **žralok** anebo **drak**?

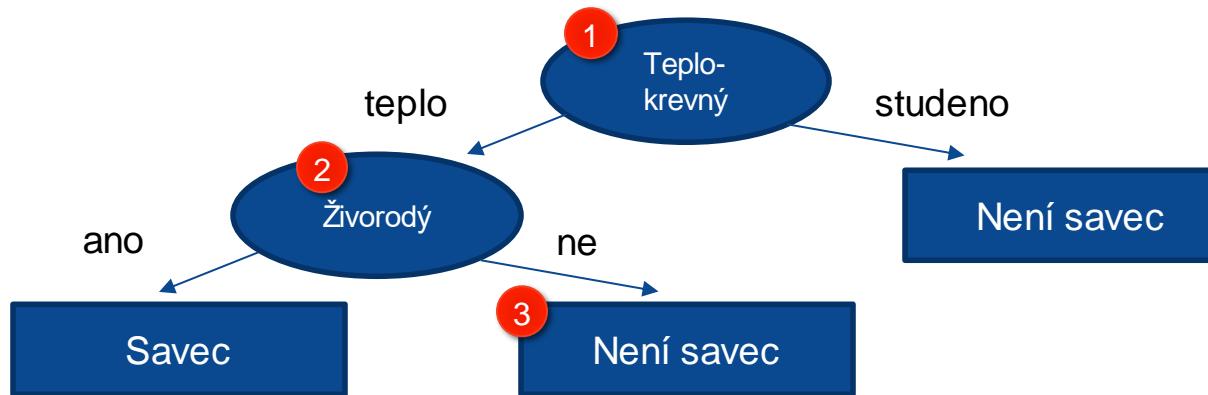
Příklad rozhodovacího stromu

Klasifikace obratlovců

Předpokládejme nově objevený druh:

Název	Teplota těla	Pokrytí těla	Živoro dí	Vodní	Létaví	Má nohy	Hibernuje	Druh
Nový druh	Studeno-krevný	roste	ne	ne	ne	ano	ano	?

- Rozhodovací strom z dat předchozího slide může vypadat následovně:



Definice: rozhodovací strom

- Mějme datovou sadu $D = \{t_1, t_2, \dots, t_n\}$, kde t_i představuje dvojici, které jsou představovány množinou atributů $A = \{A_1, A_2, \dots, A_m\}$. A množinou tříd $C = \{c_1, c_2, \dots, c_k\}$.
- Rozhodovací strom T je **strom** asociovaný s D , který má *následující vlastnosti*:
 - Každý vnitřní uzel je značen atributem A_i ,
 - Každá hrana je označena s předpokladem, který může být aplikován na asociovaný atribut s jeho rodičovským uzlem
 - Každý list stromu je označen s třídou c_j

Algoritmus tvorby rozhodovacího stromu

Algoritmus VytvořRS

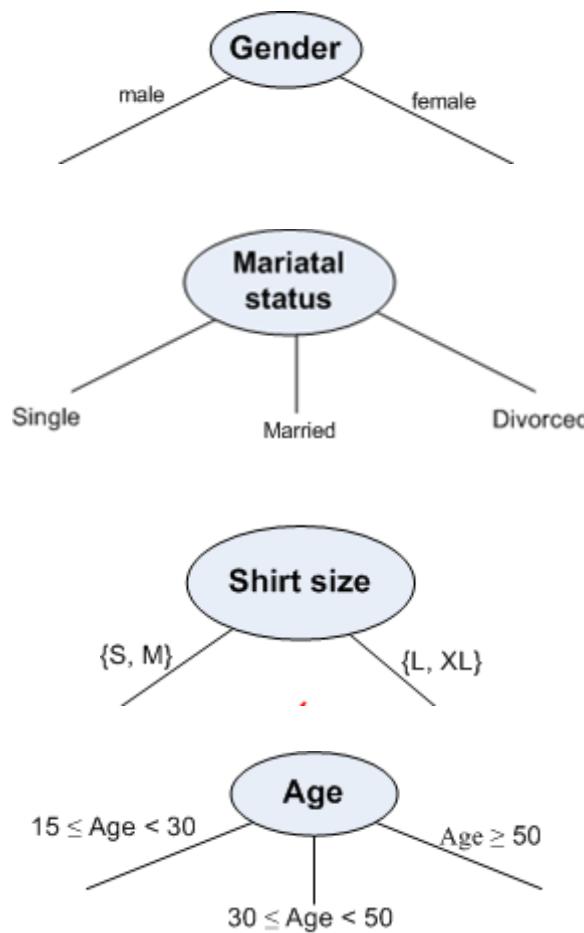
- Vstup: D : Trénovací datová množina
- Výstup: T : rozhodovací strom

Kroky

1. Pokud všechny příklady D patří do stejné třídy C_j
 Přidej list stromu a přiřaď mu třídu C_j
 Return // ukončení
2. Vyber atribut A_i (tak aby nebyl zvolen vícekrát pro každou z větví)
3. Rozděl $D = \{D_1, D_2, \dots, D_p\}$ na základě p a různých hodnotách A_i v D
4. For each $D_k \in D$
 Vytvoř uzel a přidej hranu mezi D a D_k s značkou A_i a hodnotou atributu v D_k
5. For each $D_k \in D$
 VytvořRS (D_k) // rekurzivní volání
6. Stop

Rozdělování uzlů

- Binární atribut
- Nominální atribut
- Ordinální atribut
- Číselný atribut



Algoritmus vytvoření uzlu

- Rozdělování uzlů
 - Entropie
 - Informační zisk
 - GINI index
- Algoritmy:
 - Hrubou silou (nejjednodušší, nepoužívá se)
 - ID3
 - C4.5
 - CART

Algoritmus ID3

- ID3 definuje míru pro dělení uzlu nazývanou *informační zisk* (Information Gain) který posuzuje vhodnost dělení
 - Volí se atribut s nejvyšší hodnotou informačního zisku pro rozdělení uzlu a
 - Rozdělí trénovací sadu rodičovského uzlu na dvě menší podsady na základě dělícího atributu

Strojové učení – Rozhodovací stromy

GINI index ($p(j | t)$ je relativní frekvence třídy j v uzlu t)

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

Popřípadě (CART, SLIQ, SPRINT) (n je počet všech záznamů v rodičovském uzlu p , n_i je počet záznamů spadající pod část stávajícího uzlu i):

$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

Entropie (míra neuspořádanosti), ($p(j | t)$ je relativní frekvence třídy j v uzlu t)

$$Entropy(t) = -\sum_j p(j | t) \log p(j | t)$$

Strojové učení – Rozhodovací stromy

Informační zisk (ID3, C4.5):

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right)$$

Klasifikační chybě ($p(j | t)$) je relativní frekvence třídy j v uzlu t .)

$$Error(t) = 1 - \max_i P(i | t)$$

Výpočet entropie

- Pokud se vyskytuje m objektů s frekvencí výskytů p_1, p_2, \dots, p_m , potom průměrný počet bitů (tj. otázek) které jsou nezbytné pro získání informace (tj. entropie) je četnost výskytů i – té hodnoty násobené počtem bitů, které musejí být zjištěny, pro všechna i od 1 do m .

Výpočet entropie

Pokud p_i značí četnost výskytu m rozdílných úkazů, potom entropie E je:

$$E = \sum_{i=1}^m p_i \log(1/p_i) \text{ and } \sum_{i=1}^m p_i = 1$$

Poznámka:

- Pokud jsou všechny stejně vhodné, potom $p_i = \frac{1}{m}$ a $E = \log_2 m$; jedná se o speciální případ.

Definice informačního zisku

- Uvažujme následující symboly a terminologii pro definici informačního zisku, který značíme α .
- $D \equiv$ trénovací sada
- $|D| \equiv$ počet příkladů v trénovací sadě D
- $E(D) \equiv$ entropie trénovací sady D
- Entropie trénovací množiny D

$$E(D) = -\sum_{i=1}^k p_i \log_2(p_i)$$

- Kde trénovací množina D má c_1, c_2, \dots, c_k , tj. k různých tříd a
- $p_i, 0 < p_i \leq 1$ je pravděpodobnost, že libovolný příklad D náleží třídě c_i ($i = 1, 2, \dots, k$).

Strojové učení – Rozhodovací stromy

- **Minimální velikost rozdělení** – minimální velikost uzlu, který může být ještě rozdělen.
- **Minimální velikost listu stromu** – minimální velikost všech listů stromu.
- **Minimální zisk** – minimální hodnota zisku, aby bylo povoleno rozdělení.
- **Maximální hloubka** – maximální povolená hloubka stromu.
- **Důvěryhodnost** – úroveň důvěryhodnosti použitý pro rozhodnutí o prořezávání stromu (pruning).
- **Minimální velikost rozdělení** – minimální velikost uzlu, který může být ještě rozdělen.

Rozhodovací stromy: demo

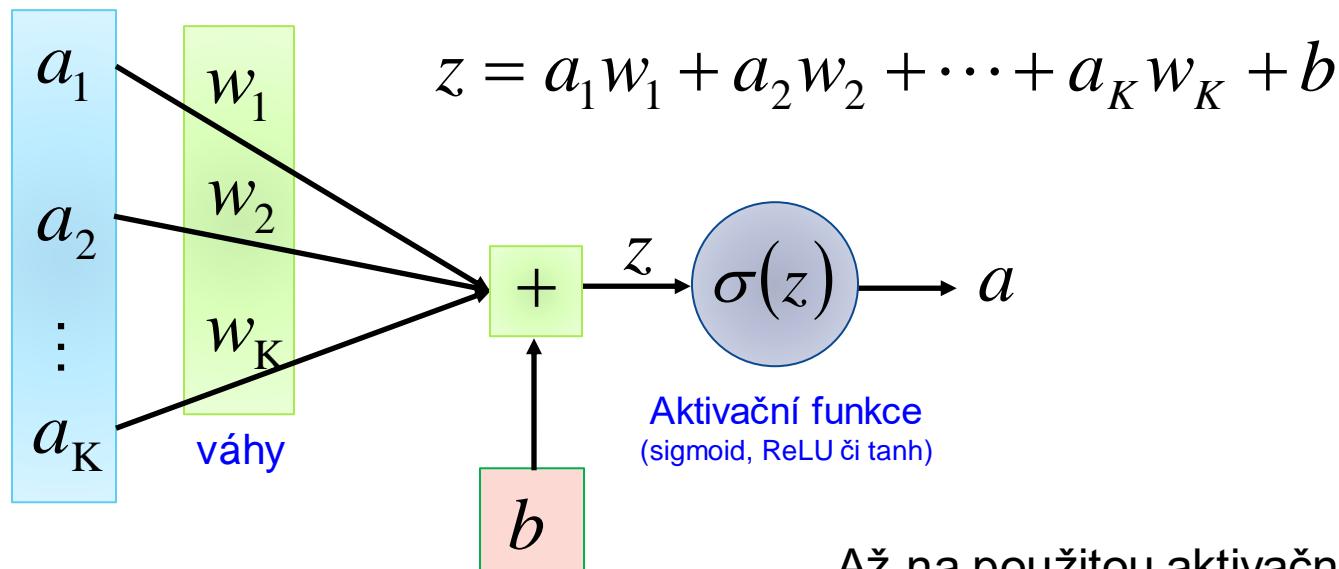
```
from sklearn.tree import DecisionTreeClassifier  
clf = DecisionTreeClassifier().fit(X_train, y_train)
```

<https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>
https://drive.google.com/file/d/1uooclccCfi1MD18p4_7FUqBzrM-O6hlf6/view?usp=sharing
Kapitola 6.2)

MTIN I: cz.vutbr.feec.mtin.w08.logistic

Část neuronové sítě

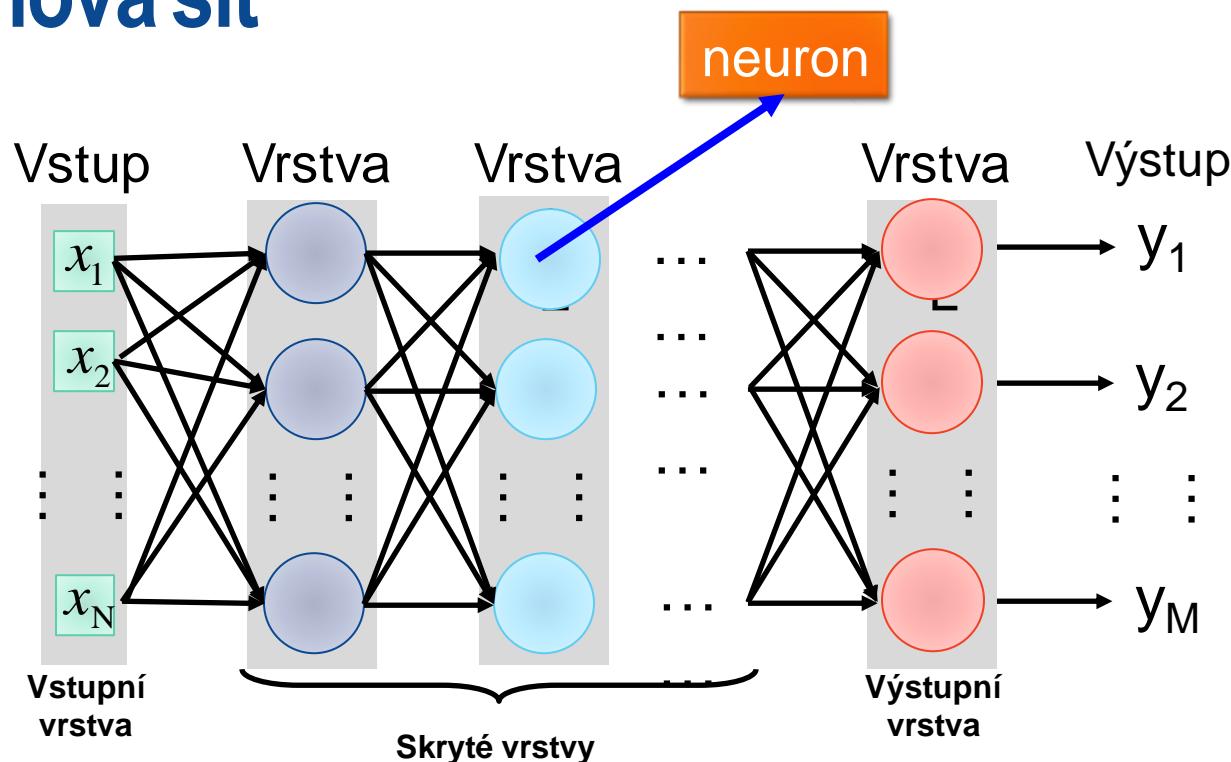
Umělý neuron $f: R^K \rightarrow R$



Pozn.: Biologický a umělý neuron jsou výrazně odlišné záležitosti

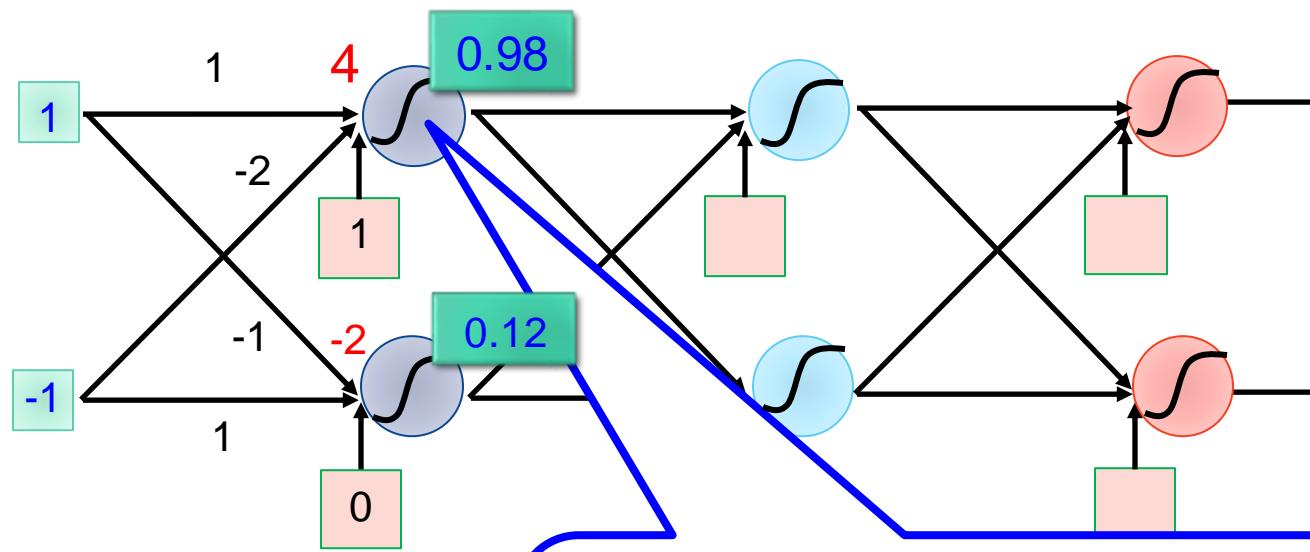
Až na použitou aktivační funkci, velmi podobné logistické regresi

Neuronová síť'



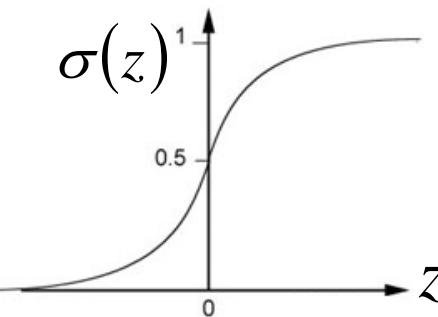
„Hluboké učení“ znamená mnoho skrytých

Příklad dopředné neuronové sítě

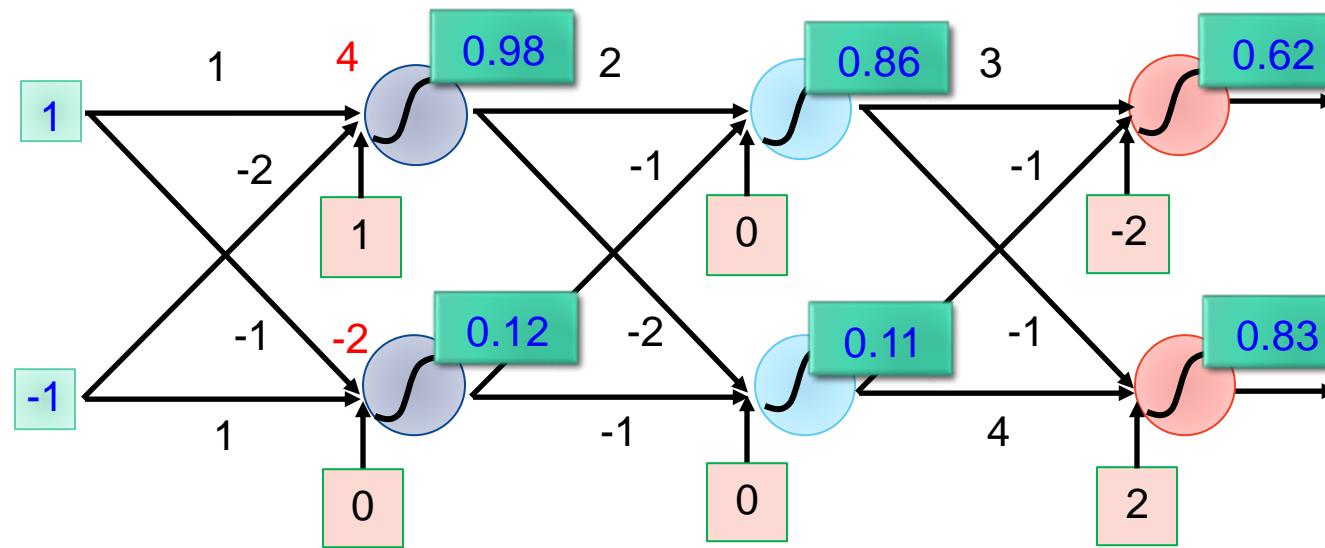


Sigmoid Function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

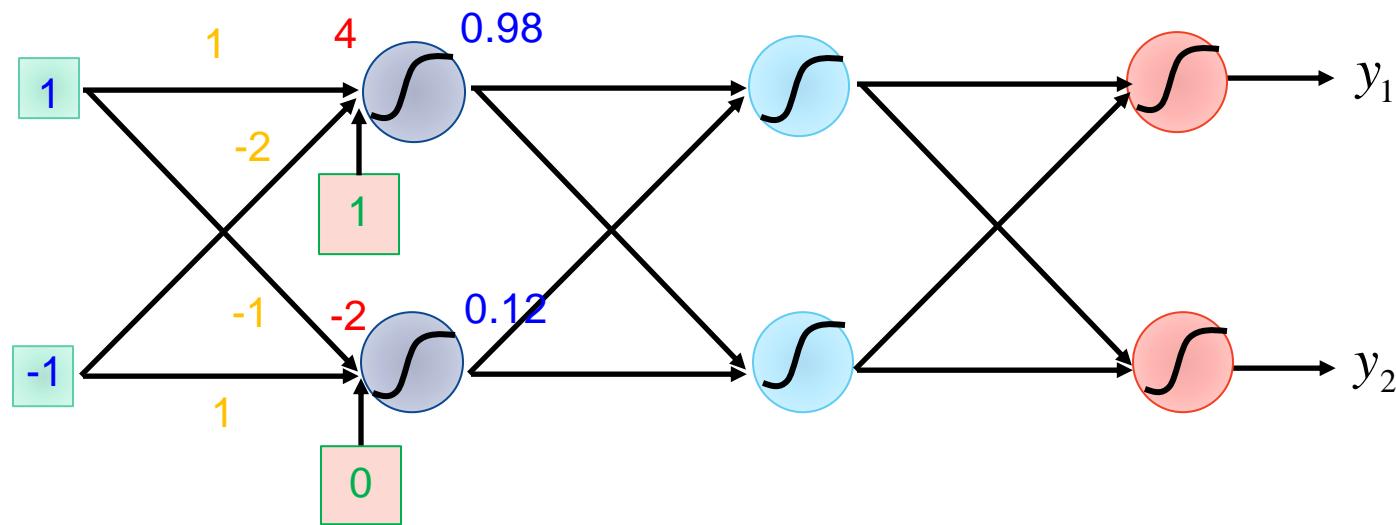


Příklad dopředné neuronové sítě





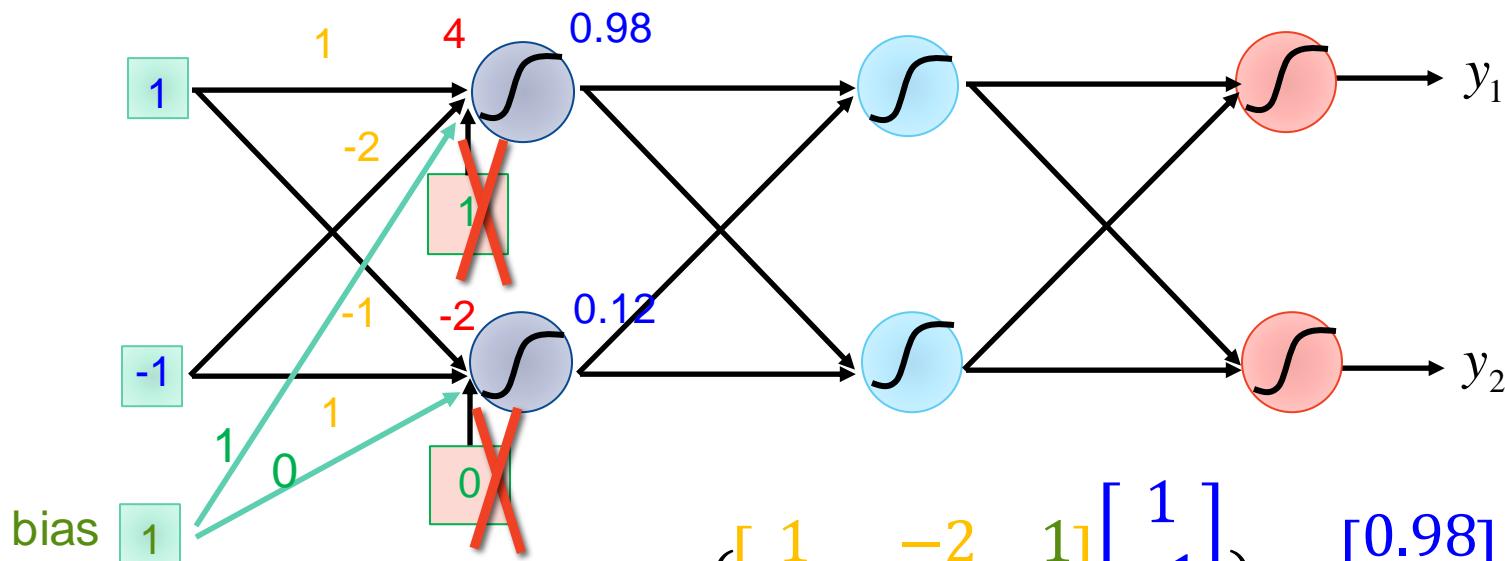
Maticové operace



$$\sigma \left(\underbrace{\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}} \right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$



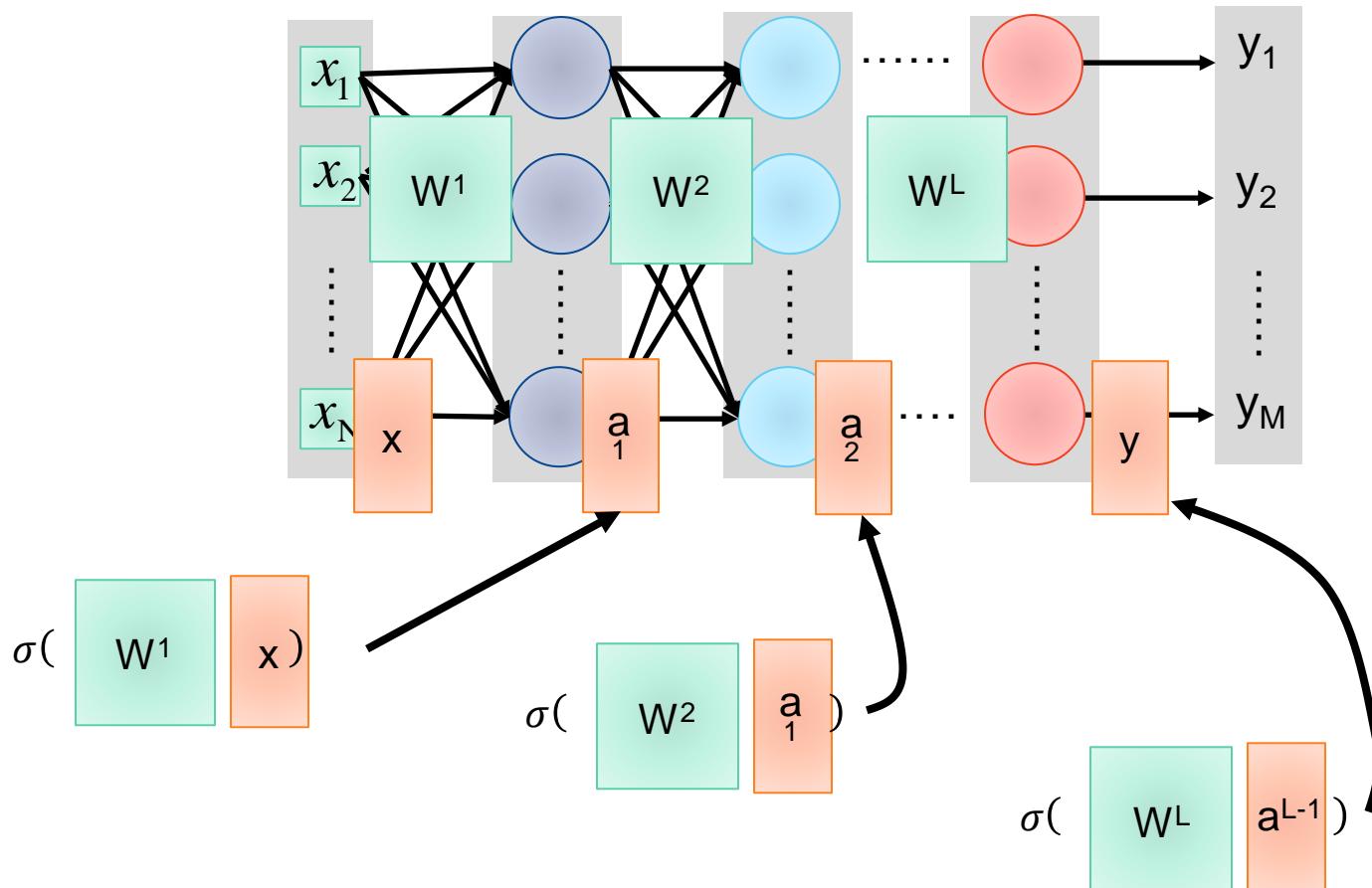
Maticové operace



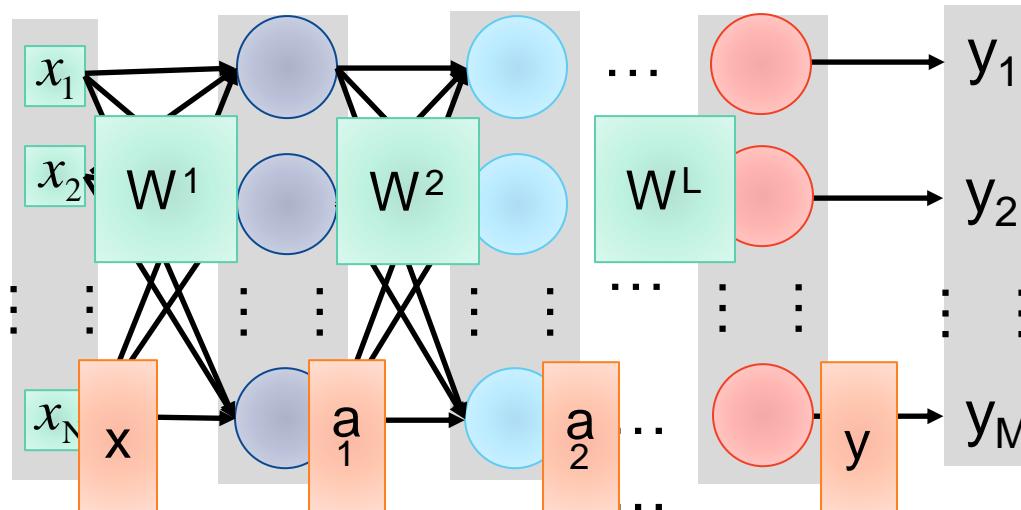
Podobně jako u lineární regrese se v praxi výpočty s odsazením (bias) zjednodušuje přímo s výpočty v matici.

$$\sigma \left(\underbrace{\begin{bmatrix} 1 & -2 & 1 \\ -1 & 1 & 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

Neuronová síť'



Neuronová síť'



$$y = f(x)$$

$$= \sigma(W^L \dots \sigma(W^2 \sigma(W^1 x) \dots))$$

Softmax

- Softmax vrstva jako výstupní vrstva

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

Obyčejná vrstva



$$z_1 \rightarrow \sigma \rightarrow y_1 = \sigma(z_1)$$



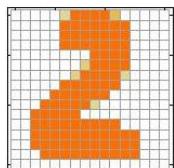
$$z_2 \rightarrow \sigma \rightarrow y_2 = \sigma(z_2)$$



$$z_3 \rightarrow \sigma \rightarrow y_3 = \sigma(z_3)$$

Jak interpretovat
výstup?

[0, 0.21, 0.01, 0, 0.2, 0, 0.1, 0, 0, 0]



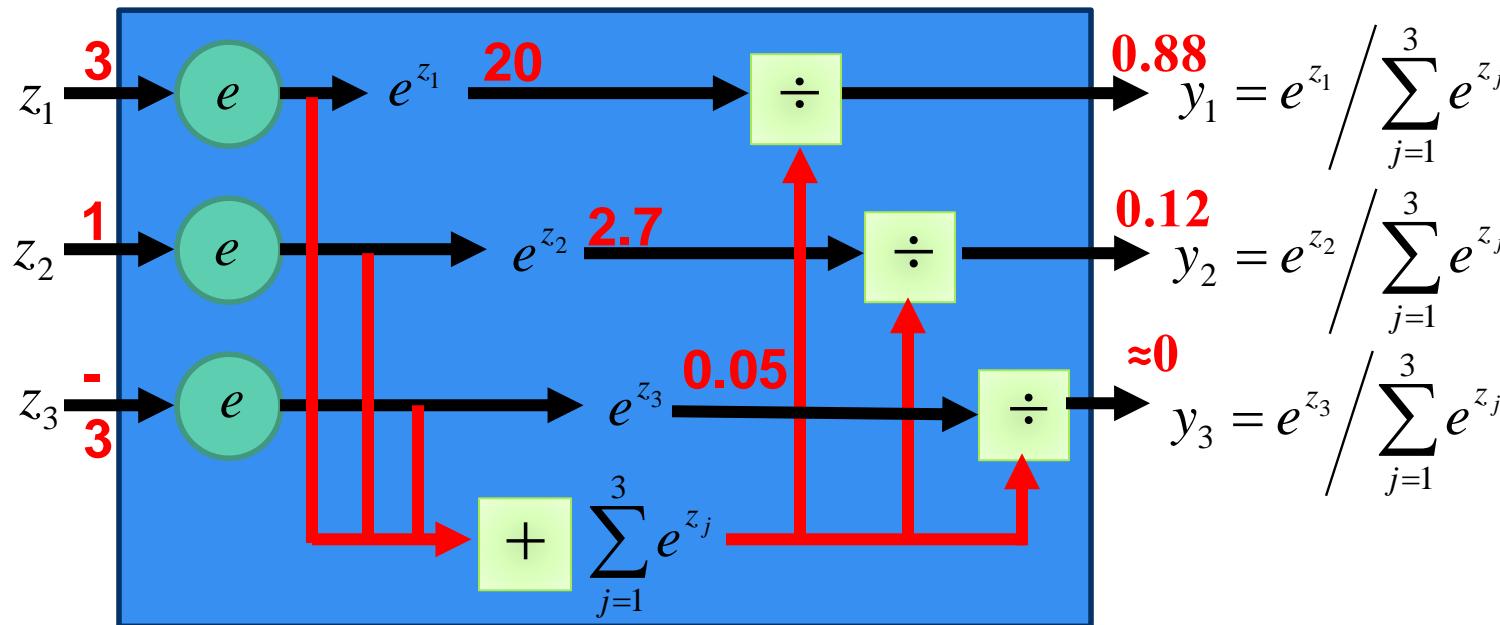
Softmax

- Softmax layer as the output layer

Pravděpodobnost:

- $1 > y_i > 0$
- $\sum_i y_i = 1$

Softmax vrstva



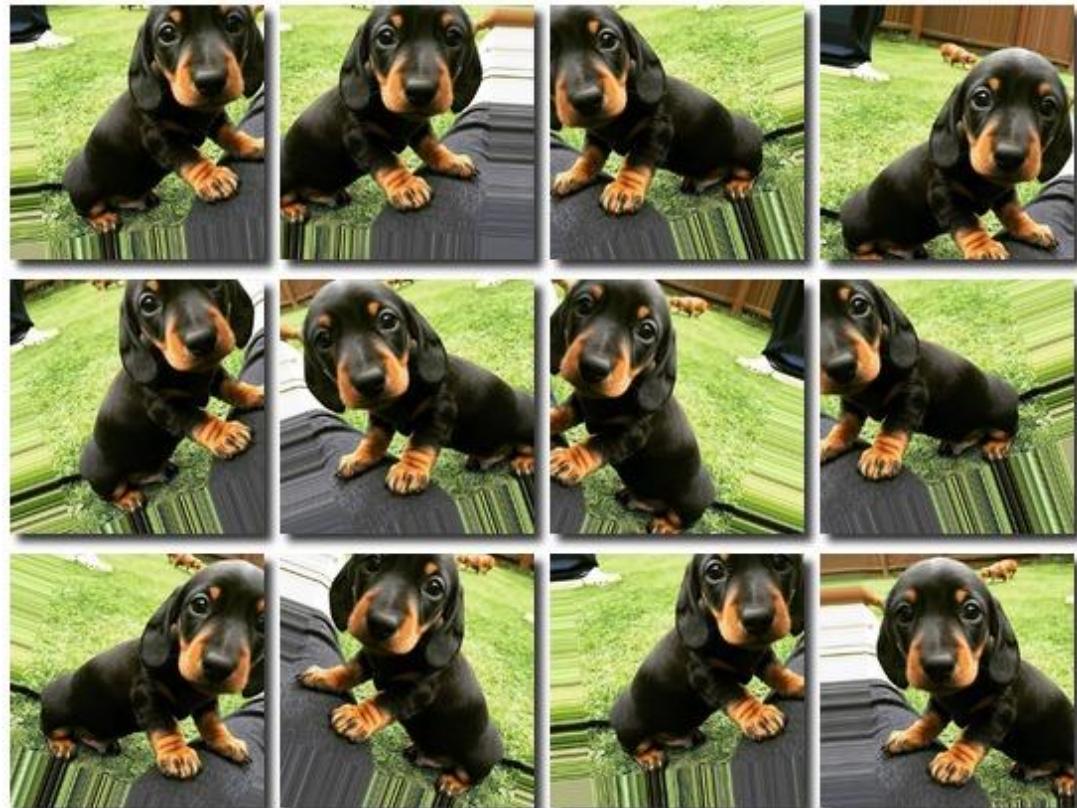
K zamyšlení

- Dává smysl použití softmax jinde než u klasifikace – například regrese? NE

Typy vrstev

- Plně propojená vrstva (viz předchozí slide)
- Konvoluční neuronová vrstva:
 - 1D ... audio
 - 2D ... obraz
 - 3D ... 3D modely, např. magnetická rezonance, poč. tomografie
- Sjednocující vrstva (Merge)
- Regularizace (L1, či L2)
- Max Pooling
- Zahazování (Dropout)

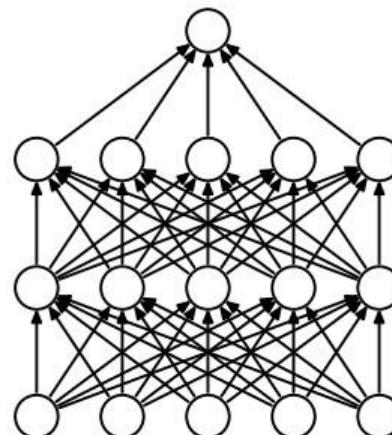
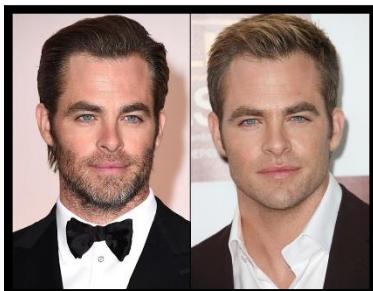
Rozšíření dat



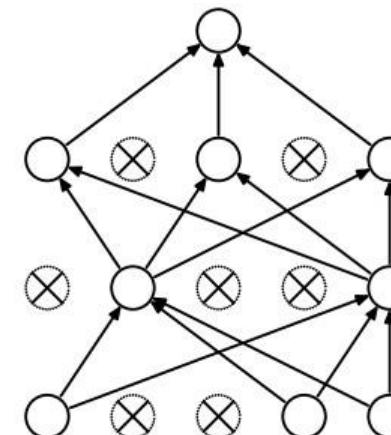
Vzorek je při každém načtení
trochu odlišný

Netřeba obrovské trénovací množiny

Zahazování (Dropout)



(a) Standard Neural Net

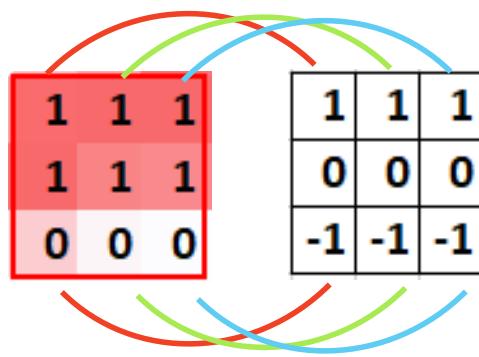


(b) After applying dropout.

Jak přesvědčit
NN, aby
používala
všechny možné
vstupy?

Náhodně nastavuje některé vstupy na „0“
jen během trénování !

Výpočet - detail



$$\begin{aligned}
 & 1*1 + 1*1 + 1*1 + \\
 & 1*0 + 1*0 + 1*0 + \\
 & 0*-1 + 0*-1 + 0*-1
 \end{aligned} = 3$$

Pozn: matematicky správně by se měla nazývat cross-corelační vrstva. Pojem „konvoluční vrstva“ se nicméně natolik zažil, že by vám asi nikdo nerozuměl.

Max Pooling

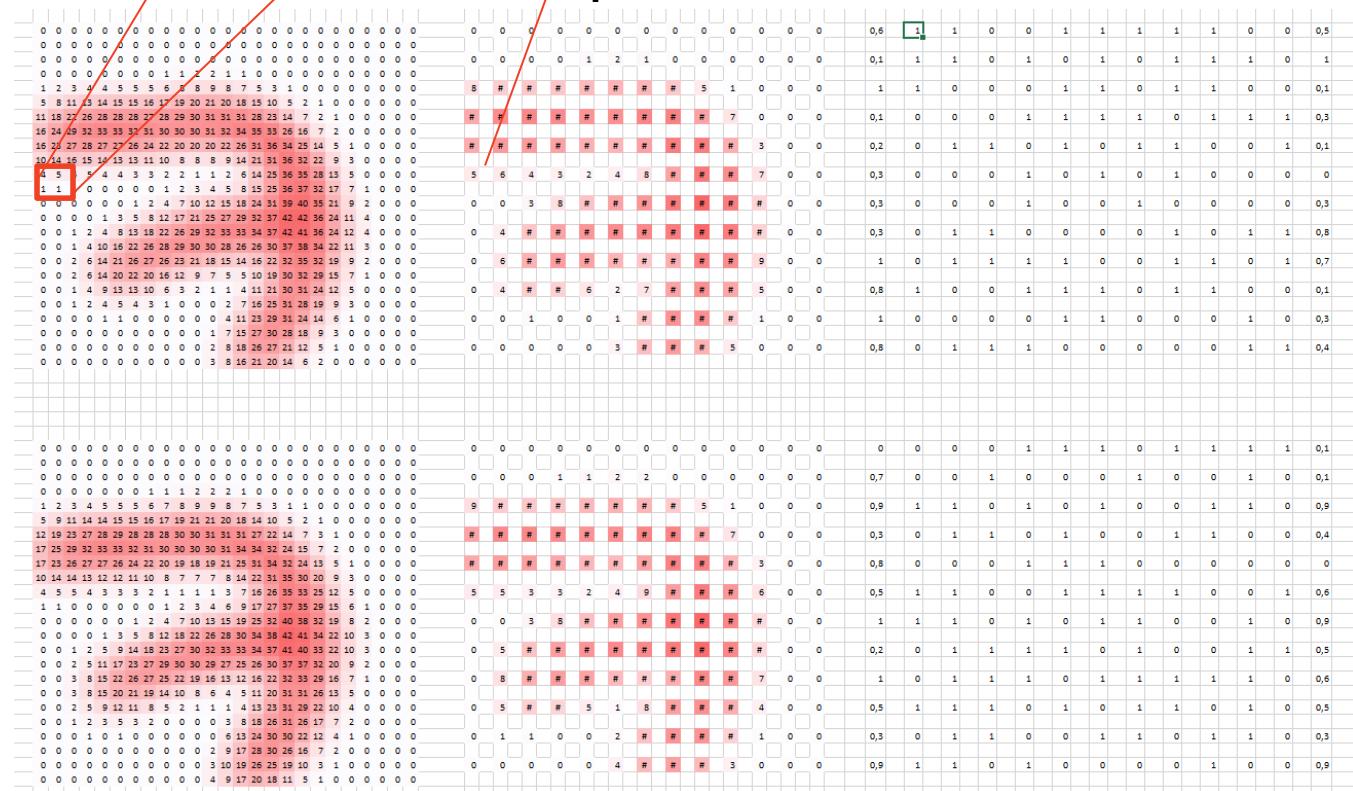
4	5
1	1

$$\max(4, 5, 1, 1) = 5$$



- Odolnost proti otáčení, zmenšení paměťové náročnosti

- Nakonec zpravidla následuje plně propojená neuronová vrstva

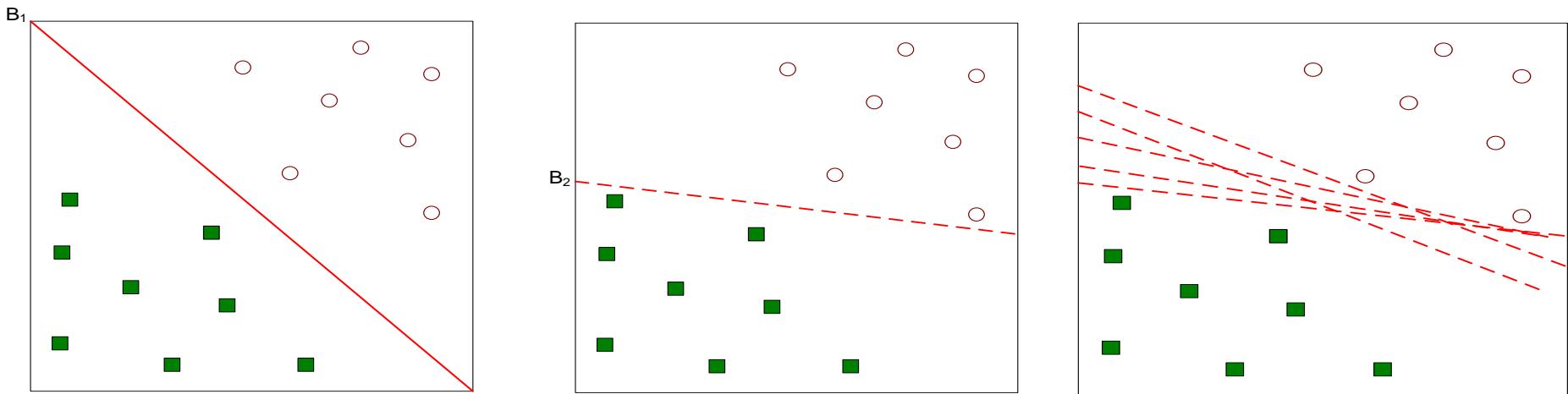


Strojové učení – Podpůrné vektory (SVM)

Algoritmus dokáže separovat i data, která původně nejsou lineárně separabilní

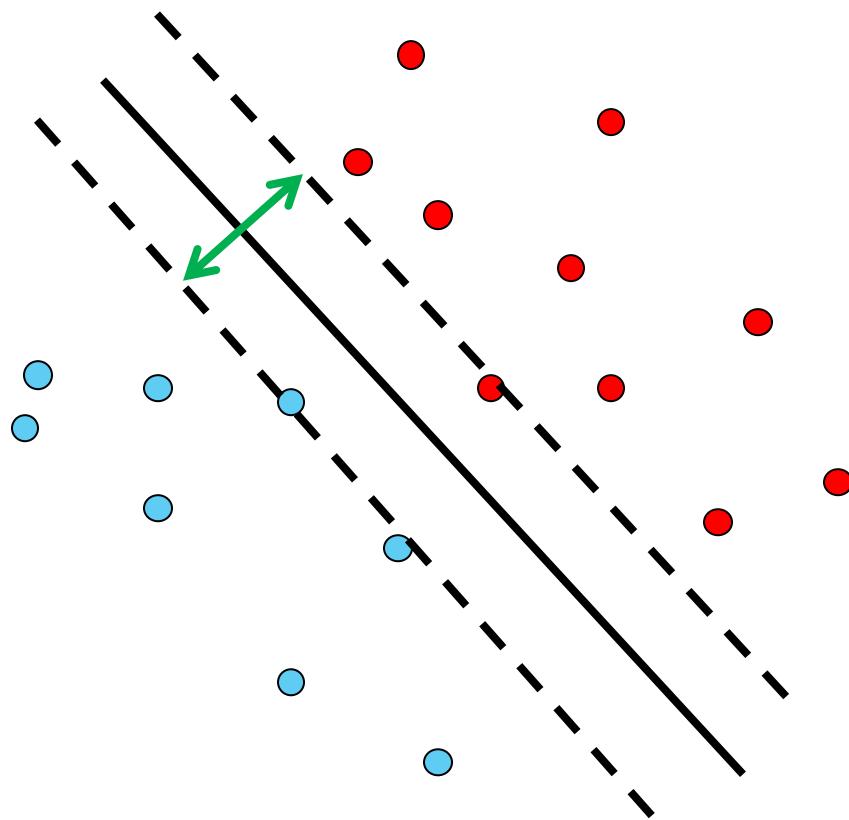
- Zjednodušený postup:
 - Nelineární mapování (pomocí vektorového modelu): převod do prostoru s vyšší dimenzí
 - Hledání optimální *nadroviny* (rozhodovací hranice)
 - Ideální oddelení jednotlivých tříd
 - Maximální vzdálenost mezi *nadrovinou* a jednotlivými prvky třídy (large margin classifiers)

Strojové učení – Podpůrné vektory (SVM)



$$f(\bar{x}_i) = \begin{cases} 1 & \text{if } \bar{w} \cdot \bar{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \bar{w} \cdot \bar{x}_i + b \leq -1 + \xi_i \end{cases}$$

SVM



- Diskriminativní klasifikátor pro optimální dělící přímku (*pro 2D případ*)
- Maximualizuje *okraj* (*margin*) mezi pozitivními a negativními trén. vzorky

SVM

- Trénování:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i \quad (\text{pro každý podpůrný vektor } i)$$

$$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

- Klasifikační funkce:

$$f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

$$= \text{sign}\left(\sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b\right)$$

*If $f(x) < 0$, klasifikuj jako negativní,
if $f(x) > 0$, klasifikuj jako pozitivní*

Strojové učení – Podpůrné vektory (SVM)

Ve výše popsaném případě vycházíme z **lineárního jádra** SVM algoritmu.

Další varianty jsou:

- Radiální
- Bodové
- Neuronové
- Anova, Gausovské,....

Použití: klasifikace textových dokumentů (Text Mining), zpracování obrazů, zpracování řeči,....

Strojové učení – Podpůrné vektory (SVM)

V rámci optimalizace se snažíme vhodně nastavit především parametry C a ξ

Jejich hodnoty nelze stanovit obecně – závisí na konkrétní trénovací množině.

Parametr C – kompromis mezi složitostí modelu (jeho hladkostí) a mírou odchylek větších než, které jsou tolerovány optimalizační rovnici.

Parametr ξ (ksí) – ovlivňuje šířku tzv. – necitlivé oblasti, která se používá k nastavení vzorků trénovacích dat.



Systémy podpůrných vektorů: demo

```
from sklearn.svm import SVC  
svm = SVC()  
svm.fit(X_train, y_train)
```

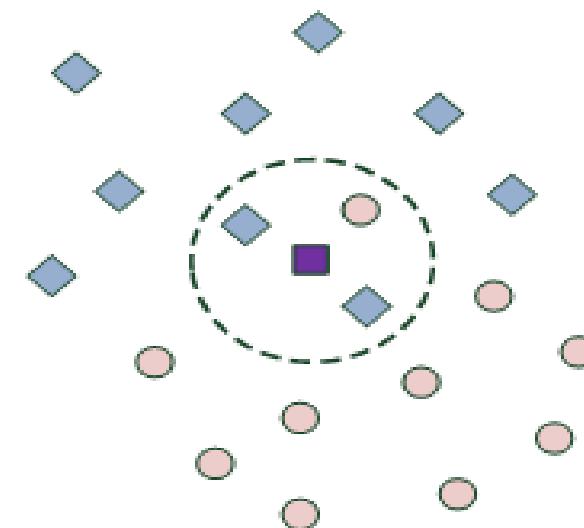
<https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>
Kapitola 6.4)

MTIN I: cz.vutbr.feec.mtin.w08.logistic

Strojové učení – k -nejbližších sousedů

- Model klasifikátoru je vytvářen až ve fázi klasifikace.
- Pro klasifikaci neznámého prvku je vyhledáno k nejbližších sousedů – nejvíce zastoupená třída je přiřazena neznámému prvku.
- K – liché číslo

$$dist(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2}$$



Příklad k-NN

Euklidovská vzdálenost dvou vektorů o délce n :
$$\sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Trénovací množina:

1	1	1	ANO
0	1	0	ANO
0	1	1	NE
0	0	0	NE

Do které kategorie spadá tento vzorek?

0.1	0.9	0.3	???
-----	-----	-----	-----

Např. $k=1$, kde nejpodobnější je dle Euklidovské vzdálenosti

Předpověď:

0.1	0.9	0.3	ANO
-----	-----	-----	-----

Jak by vypadal výpočet, pokud $k=3$?

K-nejbližších sousedů: demo

```
from sklearn.neighbors import KNeighborsClassifier  
knn = KNeighborsClassifier()  
knn.fit(X_train, y_train)
```

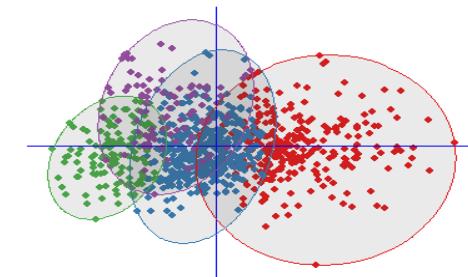
<https://colab.research.google.com/drive/1gSKX7LCzNoZDsx5Y2hmunbD6AyOTHQqw>
Kapitola 6.3)

Algoritmy strojového učení

Učení bez učitele

Strojové učení bez učitele

- **Shlukování dat** – proces seskupování dat do tříd (shluků), tak aby atributy prvků v rámci dané třídy byly co nejpodobnější a zároveň co nejodlišnější vůči atributům prvků jiných tříd.
- **Díky shlukování můžeme objevit i dosud neznámé skupiny, třídy, trendy v datech apod.**



Strojové učení bez učitele

Shluková analýza obecně

- *Klasifikace*: v některých případech může být klasifikace založená na strojovém učení nerealizovatelná – především z důvodu časové náročnosti manuální tvorby trénovací množiny.
- *Komprese dat*: s jednotlivými shluky může být zacházeno jednotně (jako se skupinou) – data ve shluku vykazují velkou podobnost.
- *Detekce odchylek (outliers)*
- *Adaptace na změny dat*: dynamicky je možné měnit zařazení do skupin v závislosti na změně dat (nemožné u manuálního zařazení prvků).

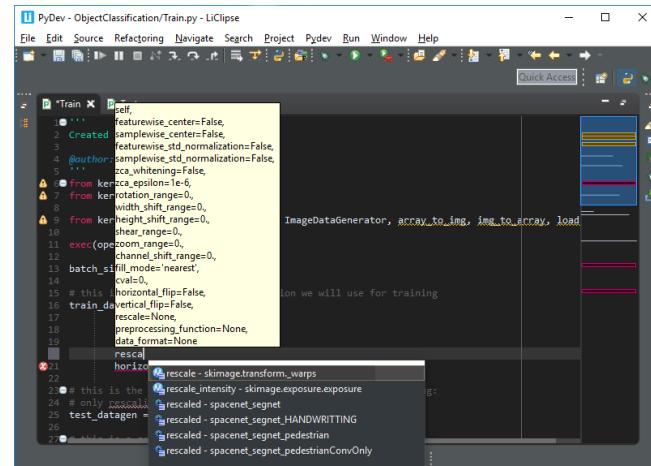
Strojové učení bez učitele

Shluková analýza konkrétně

- *Marketing*: identifikace skupin zákazníků, kteří nakupují podobné zboží.
- *Biologie*: odvození taxonomie (klasifikace) rostlin a živočichů.
- *Genetika*: kategorizace genů se stejnou funkcí.
- *Geografie*: vyhledávání oblastí v geo. datech se stejnou strukturou země.
- *Identifikace odchylek*: pojistné podvody, detekce útoku na server

Doporučené prostředí

- Tradiční strojové učení: běžné PC
- Hluboké učení:
 - Google collaboration: zdarma
 - Python: Tensorflow + PyTorch / Keras
 - Moderní herní GPU, cca \$1000
 - Tesla, cca. \$ 10 000, mírně vyšší výkon
 - Návrh: Python, Produkční nasazení: JAVA



The screenshot shows the Eclipse IDE interface with a Python file named 'Train.py' open. The code is for data augmentation using the `ImageDataGenerator` class. A tooltip is displayed over the `rescale` parameter, showing its documentation: `rescale - skimage.transform._warp.rescale_intensity - skimage.exposure.exposure.rescale_intensity`. Other options listed in the tooltip include `spacenet_segnet`, `HANDWRITING`, `pedestrian`, and `pedestrianConvOnly`.

```
PyDev - ObjectClassification/Train.py - LiClipse
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
Quick Access
*Train X Datasource
1 self
2     featurewise_center=False,
3     samplewise_center=False,
4     featurewise_std_normalization=False,
5     samplewise_std_normalization=False,
6     ...
7     @author: sambor
8     from keras import backend as K
9     from keras import rotation_range=0,
10    width_shift_range=0,
11    shear_range=0.,
12    zoom_range=0.,
13    channel_shift_range=0.,
14    batch_size=batch_size,
15    rescale=None,
16    preprocessing_function=None,
17    data_format=None
18
19    rescale=1./255
20    # this is the critical line
21    # rescale - skimage.transform._warp.rescale_intensity - skimage.exposure.exposure.rescale_intensity
22    # # only rescale
23    # rescaled - spacenet_segnet
24    # rescaled - spacenet_segnet.HANDWRITING
25    # rescaled - spacenet_segnet.pedestrian
26    # rescaled - spacenet_segnet.pedestrianConvOnly
27
```



Děkuji za pozornost