BI-VZD přednáška 1

Karel Klouda

FIT ČVUT

23. 9. 2021

Autoři: Karel Klouda, Juan Pablo Maldonado Lopez, Daniel Vašata.
Problémy, návrhy apod. hlaste v GitLabu.
Verze souboru: 29. září 2021 20:07.

Co bude v dnešní přednášce

- základní informace, úvodní příklad
- problém klasifikace
- jak fungují rozhodovací stromy
- jak se budují rozhodovací stromy

Příklad: prenatální (1/4)

- S prvním *datovým modelem* jste se pravděpodobně setkali ještě v prenatálním období Vašeho života.
- Při odhadování porodní váhy plodu pomocí ultrazvuku (tzv. SONO) se používá mj. následující model:

Model pro porodní váhu (Shephard et al.)

$$\log_{10}(eFW) = -1.749 + 0.017 \cdot BPD + 0.005 \cdot AC - \frac{2.646}{1000} \cdot (BPD \cdot AC),$$

eFW = odhadovaná hmotnost při narození, BPD = biparietal diameter (příčný průměr hlavy), AC = abdominal circumference (obvod břicha).

- Pro nás to bude **lineární regresní model** (7. přednáška), kde je
  - porodní váha eFW tzv. **vysvětlovaná proměnná** (angl. **target variable**),
  - BPD, AC a $BPD \cdot AC$ jsou pak tři tzv. **příznaky** (anlg. a často i česky **features**),
  - čísla $1.749, 0.017$ atd. jsou takzvané **regresní koeficienty** (příp. váhy), obecně **parametry modelu**.

Příklad: prenatální (2/4)

Model pro porodní váhu (Shephard et al.)

$$\log_{10}(eFW) = -1.749 + 0.017 \cdot BPD + 0.005 \cdot AC - \frac{2.646}{1000} \cdot (BPD \cdot AC),$$

- **Jak se tento model používá?**
  1. Na ultrazvuku se změříte veličiny BPD a AC,
  2. získaná čísla dosadíte do pravé strany vzorce,
  3. výsledek je odhad hodnoty $\log_{10}(eFW)$ a tedy i kýžené porodní váhy eFW.

- **Kde se vzala ta divná čísla jako 2.646 apod.?**
  To se právě budeme učit v tomto kurzu: Jsou to parametry zvoleného modelu a ty se vždy nějakým způsobem odhadují na základě dostupných dat, to je tzv. **učení modelu**.

Příklad: prenatální (3/4)

Jak asi pan Shephard et al. došli právě k tomuto modelu:

1. Z nějakého důvodu věřili, že příznaky BPD a AC jsou pro porodní váhu důležité. Nejspíše ale zkoušeli i jiné, ale ty buď nepřinášely velké zlepšení nebo se daly BPD a AC plně nahradit.

2. Pomocí různých testovacích procedur si jako model zvolili lineární regresní model s třemi příznaky a čtyřmi koeficienty:

$$\log_{10}(eFW) = w_0 + w_1 \cdot BPD + w_2 \cdot AC + w_3 \cdot (BPD \cdot AC).$$

3. Předchozí fázi, kdy hledáme „tvar“ modelu, se říká **ladění hyperparametrů** (angl. **hyperparameter tuning**).

4. Koeficienty w_i pak odhadli z dat o již narozených dětech, u kterých měli přesnou porodní váhu i prenatálně naměřené hodnoty BPD a AC.

Příklad: prenatální (4/4)

Zmíněná data, na kterých se model učil (a testoval), mohla vypadat nějak takto:

`kid_id`	FW	BCD	AC
1	číslo	číslo	číslo
2	číslo	číslo	číslo
⋮	⋮	⋮	⋮

Pro zajímavost (detaily v 7. přednášce):

- Označme sloupec pod FW jako vektor $\boldsymbol{Y}$
- a vytvořme matici $\mathbf{X}$ o čtyřech sloupcích, kde v každém řádku je vektor $(1, BCD, AC, BCD \cdot AC)$.
- Nejpoužívanější metoda pro výpočet koeficientů w_i je **metoda nejmenších čtverců**, ta nám říká, že

$$(w_0, w_1, w_2, w_3)^T = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{Y}.$$

- Tento vzorec jsme získali vyřešením jistého optimalizačního problému (a.k.a. *hledání extrémů*), což je velice častý případ: **učení modelu = optimalizace**!

Supervizované vs. nesupervizované učení

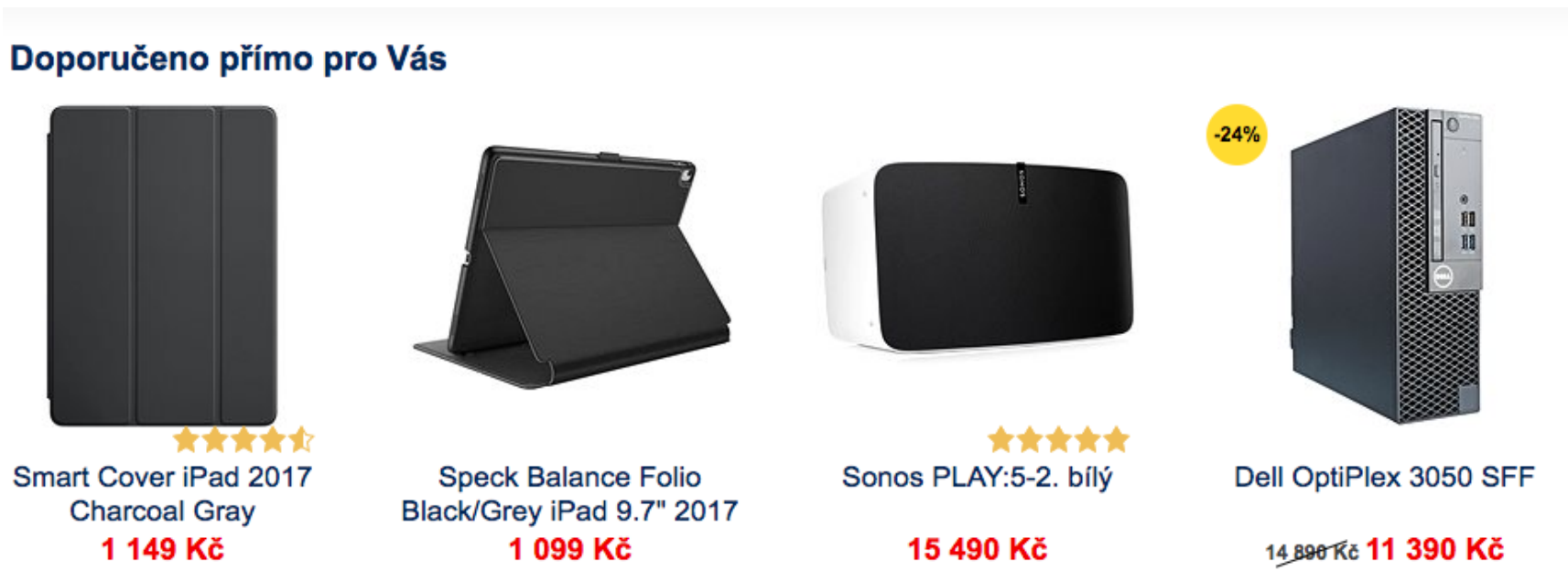
- Předchozí *prenatální* příklad je typickou ukázkou **supervizovaného učení** (učení s učitelem, angl. **supervized learning**).
- Tím „učitelem“ jsou zde známé hodnoty porodních vah u dětí, což je veličina, kterou se snažíme pomocí modelu *predikovat* resp. pochopit, na čem závisí.
- Někdy takovou veličinu ale ani nemáme a prostě se v datech pokoušíme nějak vyznat a najít jejich skrytou strukturu.
- Takovým problémům se říká **nesupervizované učení** (učení bez učitele) a typickým příkladem je **clusterování** dat (téma 4. přednášky).

Příklady nesupervizovaného učení

- Problém clusterování je velice obvyklý v praxi.
- Pokud máte například e-shop (nebo banku, nebo telefonního operátora), chcete se vyznat ve svých zákaznících, o kterých máte nasbíraná různá data (tzv. *customer segmentation*).
- Můžete tak hledat např. podmnožinu „nejlepších“ zákazníků, kterým má cenu věnovat speciální péči. Nebo naopak skupinu, která potřebuje k polepšení pomoci nějakou reklamní akcí (cílení reklamy je velký byznys).
- Do nesupervizovaného učení také (obvykle) spadá i **detekce anomálií** (angl. **anomaly detection**).
- Např. banka se snaží najít podezřelé transakce (fraud detection, ochrana proti zneužití karty, atp.).

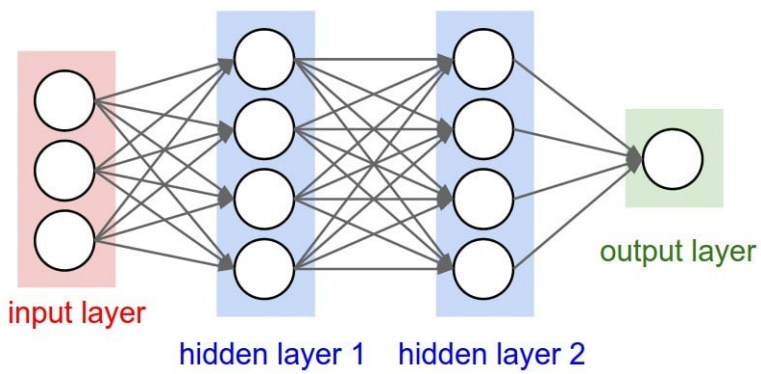
Další příklady: doporučovací systémy

- Dalším příkladem problému řešeného pomocí zkoumání dat je tzv. **doporučování** (angl. **recommendation**).
- Například: vlastníte-li e-shop (příp. internetový časopis, iTunes, Netflix atp.), snažíte se na základě dat o zákaznících a zejména zákazníkovi, který právě prohlíží Vaše stránky, odhadnout, co by si tak mohl ještě chtít koupit (přečíst, podívat, poslechnout) a to mu ukázat.



Další příklady: data bez jasných příznaků

- Data ale vždy nemusí mít formu tabulky. Může se jednat o obrázky, videa, časové řady, dlouhé texty atp., ze kterých je těžké získat pro modely příznaky.
- V takovém případě si musíte dát práci a nějaké příznaky z dat vydolovat (tzv. **feature extraction**).
- Nebo použijete algoritmy a metody, které si příznaky vytvářejí samy automaticky.
- Mezi takové metody patří (čím dál populárnější) umělé **neuronové sítě** (angl. artificial **neural networks**, ANN)
- O ANN budeme mluvit v posledních přednáškách. Používají se k všemožným úkolům (překlady, detekce objektů v obrázku videu, hraní GO, clusterování, detekci anomálií, . . .).



Co je problém klasifikace

- Supervizované učení: Snažíme se zjistit, jak vysvětlovanou proměnnou Y ovlivňují příznaky $X_0, X_1, \ldots, X_{p-1}$, hledáme tedy nějaký funkční vztah tak, aby „co nejvíce platilo“

$$Y \approx f(X_0, X_1, \ldots, X_{p-1}).$$

- Funkce f nemusí být nutně podobná funkcím, které znáte z analýzy. Např. v této přednášce to bude strom ϟ.
- Tvar hledané funkce často ovlivňuje to, jakých hodnot může nabývat vysvětlovaná proměnná Y:
  - Může-li nabývat jen několik málo hodnot, mluvíme o problému **klasifikace** (angl. **classification**). Sem spadá např. určení, jestli pacient má/nemá nemoc, jaké písmeno je (ručně) napsáno na obrázku, atp.
  - Může-li nabývat tolika hodnot, že je rozumnější ji považovat za *spojitou*, mluvíme o problému **regrese** (angl. **regression**).
- **Rozhodovací stromy** (angl. **decision trees**) lze použít pro oba typy problému: my začneme tím klasifikačním.

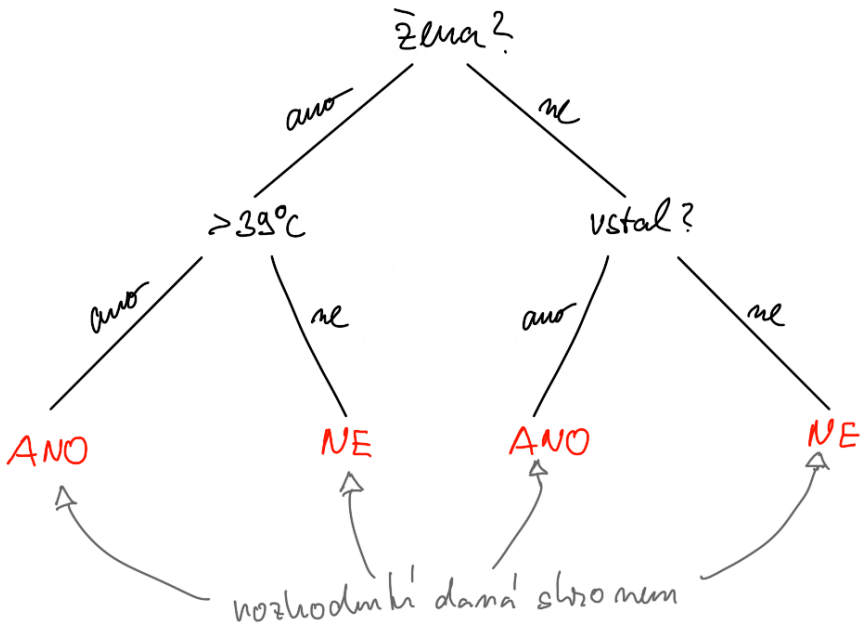
Ukázka použití rozhodovacího stromu (1/6)

- Velmi často je klasifikační problém **binární**, kdy proměnná Y může mít jen dvě hodnoty.
- My si použití stromu ukážeme na (vymyšlených) datech a problému určování, jestli pacient má či nemá závažnou nemoc známou jako „rýmička“.
- Příznaky budou pro jednoduchost také binární: Pohlaví (žena/muž), horečka ($> 39°\mathrm{C}/\leq 39°\mathrm{C}$) a to, jestli daný člověk zvládl/nezvládl vstát z postele.
- Ukážeme si dva rozhodovací stromy a porovnáme si, jak je který z nich dobrým modelem následujících dat:

rýmička	pohlaví	$> 39°\mathrm{C}$	vstal(a)?
ano	muž	ne	ne
ne	žena	ano	ano
ne	muž	ne	ano
ano	žena	ano	ne

Ukázka použití rozhodovacího stromu (2/6)

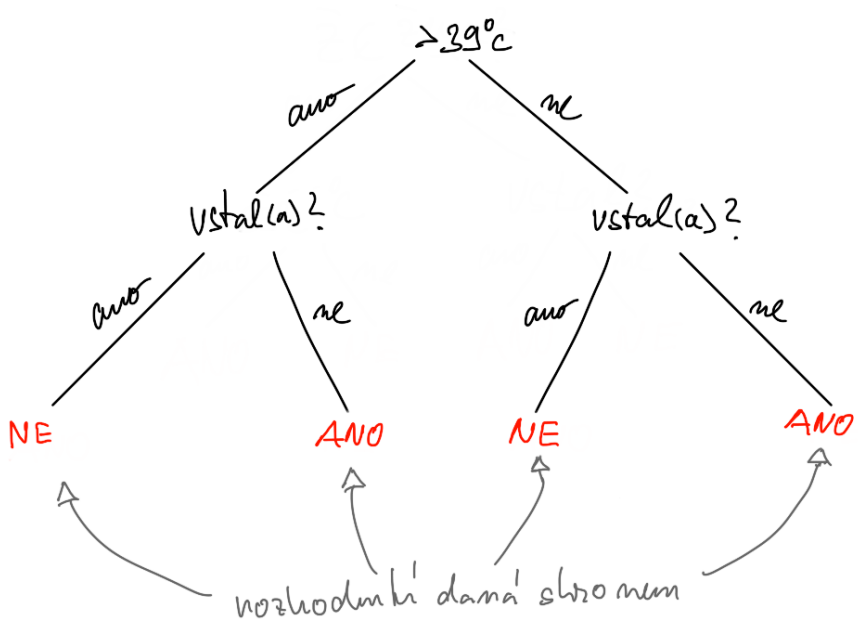
Strom 1:



rýmička	pohlaví	$> 39°C$	vstal(a)?	co říká strom
ano	muž	ne	ne	ano
ne	žena	ano	ano	ano
ne	muž	ne	ano	ne
ano	žena	ano	ne	ano

Ukázka použití rozhodovacího stromu (3/6)

Strom 2:



rýmička	pohlaví	$> 39°\mathrm{C}$	vstal(a)?	co říká strom
ano	muž	ne	ne	ano
ne	žena	ano	ano	ne
ne	muž	ne	ano	ne
ano	žena	ano	ne	ano

Ukázka použití rozhodovacího stromu (4/6)

- Strom 1 dává špatný výsledek pro druhý řádek, strom 2 dává správné výsledky pro všechny řádky.
- Strom 2 je tedy, zdá se, lepší. Je však toto dostatečné zdůvodnění?
- My ve skutečnosti chceme vědět, jak často se strom trefí **pro všechna možná data**.
- Což je trochu smůla, protože všechna možná data nikdy nemáme. Máme většinou jen „jednu tabulku“ dat a ta nám musí stačit jak pro vytvoření (neboli naučení) stromu, tak i pro ověření toho, jak je dobrý.
- Jak se to dělá, si ukážeme později.

Ukázka použití rozhodovacího stromu (5/6)

- Strom 1 i strom 2 jsou stromy hloubky 2 a mají tedy 4 listy.
- Kdybychom tedy vytvořili strom hloubky 3, měl by 8 listů. Přesně tolik je ale taky možných kombinací hodnot tří příznaků! **To už není model, ale index!**
- Strom hloubky tři by se tedy mýlil pouze v případě, že by hodnoty všech příznaků byly stejné, ale hodnota vysvětlované proměnné by byla jiná (např. kdyby jeden pacient byl chlapík s angínou):

rýmička	pohlaví	$> 39°\mathrm{C}$	vstal(a)?
⋮	⋮	⋮	⋮
ano	muž	ano	ne
ne	muž	ano	ne
⋮	⋮	⋮	⋮

- V takovém případě by se mýlil libovolný strom a dokonce i jakákoli funkce příznaků (viz definici funkce).

Ukázka použití rozhodovacího stromu (6/6)

- Skutečným model rýmičky je, jak známo, toto: rýmička nastává právě když

$$(\text{žena} \ \wedge (> 39°) \wedge \ \text{nevstala}) \vee (\text{muž} \ \wedge ((> 39°) \vee \ \text{nevstal}))$$

- Takový model ale není postižitelný stromem hloubky dva! Vzpomeňme BI-MLO a minimalizaci formulí v disjunktivním normálním tvaru ...

Konstrukce stromu: základní úkoly

Postupně si ukážeme, jak se řeší následující problémy:

- Máme-li data s příznaky i s hodnotami vysvětlované proměnné, jak zkonstruuji rozhodovací strom, který data co nejlépe modeluje?
- Jak poznám, že můj vytvořený strom není jen dobrým modelem dat, která mám, ale bude dobře fungovat i pro data jiná?
- Jak poznám, jakou mám zvolit hloubku stromu příp. jiné jeho parametry?
- Jak si poradit s nebinárními, nebo dokonce se spojitými parametry?

Konstrukce stromu: formulace úlohy

- Na vstupu máme N řádkovou tabulku s hodnotami pro binární vysvětlovanou proměnnou a p binárních příznaků $X_0, X_1, \ldots, X_{p-1}$.
- Cílem je vytvořit strom zadané hloubky k, který správně přiřadí hodnotu Y co nejvíce řádkům z tabulky.
- Jak to vyřešit? Vyzkoušejme všechny stromy a pro každý změříme podíl správně určených a je hotovo!
- Nebo je to snad problém?
- Stromů hloubky 1 je p, stromů hloubky 2 je $p \cdot (p-1)^2$, stromů hloubky 3 je „fakt hodně“, ...
- Konstrukce hrubou silou je neprůchozí kvůli počtu možných stromů, ve skutečnosti je konstrukce optimálního stromu **NP-úplný** problém (viz [Hyafil, Rivest, (1976)]).

Konstrukce stromu: hladový ID3 algoritmus

- Pro konstrukci stromů se používá **hladový algoritmus** označovaný jako **ID3** (resp. jeho rafinovanější verze **C4.5** a **C5** vše od **Johna Rosse Quinlana**).
- Tyto algoritmy pro danou množinu dat vybírají jeden (ze zatím nepoužitých) příznaků, který rozdělí data na dvě části tak, že vzniklé rozdělení maximalizuje vybrané kritérium (viz dále).
- Daná množina dat je tak rozdělena na dvě části a na každou zvlášť je pak aplikován stejný postup, jehož výsledkem jsou další dva příznaky, použité jako kritérium a rozdělení na čtyři podmnožiny dat.
- Takto se postupuje, dokud nenastane nějaké zastavovací kritérium (maximální hloubka stromu, na listech stromu už je málo dat, atp.).
- Jak zvolit to kritérium? Používají se dvě, o obou si řekneme později.

Konstrukce stromu: ukázková data

Zkusme zkonstruovat strom pro následující binární data s třemi příznaky:

id	Y	X_0	X_1	X_2
0	1	1	0	0
1	1	0	1	1
2	1	1	0	0
3	1	1	1	1
4	0	0	0	1
5	0	0	1	0
6	0	0	0	1
7	0	1	1	0

? Jaký příznak použít jako první k rozdělení dat?

- Příznak X_0 rozdělí data na dvě části[1] $\{1_0, 1_2, 1_3, 0_7\}$ a $\{1_1, 0_4, 0_5, 0_6\}$.
- Příznak X_1 rozdělí data na dvě části $\{1_1, 1_3, 0_5, 0_7\}$ a $\{1_0, 1_2, 0_4, 0_6\}$.
- Příznak X_2 rozdělí data na dvě části $\{1_1, 1_3, 0_4, 0_6\}$ a $\{1_0, 1_2, 0_5, 0_7\}$.

[1] Uvádíme hodnotu Y a jako dolní index `id` přísl. řádku.

Konstrukce stromu: kritérium volby příznaku

- Příznak X_0 rozdělí data na dvě části $\{1_0, 1_2, 1_3, 0_7\}$ a $\{1_1, 0_4, 0_5, 0_6\}$.
- Příznak X_1 rozdělí data na dvě části $\{1_1, 1_3, 0_5, 0_7\}$ a $\{1_0, 1_2, 0_4, 0_6\}$.
- Příznak X_2 rozdělí data na dvě části $\{1_1, 1_3, 0_4, 0_6\}$ a $\{1_0, 1_2, 0_5, 0_7\}$.

? Který příznak je lepší a jak to změřit?

- Na vstupu jsou data $\{1_0, 1_1, 1_2, 1_3, 0_4, 0_5, 0_6, 0_7\}$, kde jsou hodnoty 0 a 1 zastoupeny rovnoměrně.
- Ideální by byl příznak, který data rozdělí na dvě skupiny, jednu se samými 1 a druhou s 0. Takový ale nemáme k dispozici.
- Příznaky X_1 a X_2 jsou ale opačný extrém: Data rozdělí na dva kusy, kde jsou opět 0 a 1 zastoupeny přesně napůl.
- Příznak X_0 pak představuje zřejmě nejlepší volbu, neb data alespoň trochu více „uspořádá“: z rovnoměrného zastoupení k poměru 1 ku 3.
- Jak tuto uspořádanost resp. neuspořádanost měřit?

Míra **ne**uspořádanosti

- Máme množinu $\mathcal{D}$ nul a jedniček (nebo i více hodnot) a chceme nějak změřit, jak moc je uspořádaná.
- Co by taková míra měla splňovat? Označme p_0 a p_1 poměry počtu 0 resp. 1 v množině (tj. $p_0 + p_1 = 1$).
  1. Míra by měla být nezáporná (z technických důvodů).
  2. Pokud jsou v množině např. samé nuly (tj. $p_0 = 1$), měla by být neuspořádanost nulová.
  3. Měla by být maximální, pokud jsou počty nul a jedniček stejné, tj. když $p_0 = p_1 = \frac{1}{2}$.
  4. Měla by to být rostoucí funkce p_0 na intervalu $[0, \frac{1}{2}]$ a klesající na intervalu $[\frac{1}{2}, 1]$.
- Taková funkce **měřící neuspořádanost** existuje a říká se jí **Entropie**:

$$H(\mathcal{D}) = -p_0 \log p_0 - p_1 \log p_1 = -p_0 \log p_0 - (1 - p_0) \log(1 - p_0).$$

- Pro nebinární případ s k různými hodnotami je to analogické:

$$H(\mathcal{D}) = -\sum_{i=0}^{k-1} p_i \log p_i.$$

Entropie: poznámky

- Formálněji se budete entropii věnovat v předmětu *NI-VSM: Vybrané statistické metody*, kde si zavedete entropii *náhodné veličiny*. Nám ale stačí předchozí intuitivní zavedení[2].
- Ve vzorci pro entropii budeme používat dvojkový logaritmus. V takovém případě se jednotce entropie říká **bit**.
- Entropie množiny našich vstupních dat $\mathcal{D} = \{1_0, 1_1, 1_2, 1_3, 0_4, 0_5, 0_6, 0_7\}$ je rovna 1, neboť $p_0 = \frac{1}{2}$, a tedy

$$H(\mathcal{D}) = -\frac{1}{2}\log\frac{1}{2} - \left(1-\frac{1}{2}\right)\log\left(1-\frac{1}{2}\right) = -2\frac{1}{2}\log\frac{1}{2} = -\log\frac{1}{2} = 1.$$

- Entropie množiny dat $\mathcal{D}_1 = \{1_0, 1_1, 1_3, 0_7\}$ je rovna ($p_0 = \frac{1}{4}$)

$$H(\mathcal{D}_1) = -\frac{1}{4}\log\frac{1}{4} - \left(1-\frac{1}{4}\right)\log\left(1-\frac{1}{4}\right) = 0.811278124459133 28\ldots$$

[2] To, co jsme zavedli, není přesně řečeno entropie, ale její odhad na základě dat. Skutečné pravděpodobnosti p_i totiž vlastně neznáme a používáme jen jejich odhad.

Konstrukce stromu: informační zisk

Vraťme se k otázce, který příznak použít pro rozdělení množiny dat $\mathcal{D} = \{1_0, 1_1, 1_2, 1_3, 0_4, 0_5, 0_6, 0_7\}$:

- Příznak X_0: $\mathcal{D}_1 = \{1_0, 1_2, 1_3, 0_7\}$ a $\mathcal{D}_0 = \{1_1, 0_4, 0_5, 0_6\}$.
- Příznak X_1: $\mathcal{D}_1 = \{1_1, 1_3, 0_5, 0_7\}$ a $\mathcal{D}_0 = \{1_0, 1_2, 0_4, 0_6\}$.
- Příznak X_2: $\mathcal{D}_1 = \{1_1, 1_3, 0_4, 0_6\}$ a $\mathcal{D}_0 = \{1_0, 1_2, 0_5, 0_7\}$.

Chceme vybrat příznak, který rozdělením dat nejvíce sníží neuspořádanost! Toto snížení se určuje tzv. **informačním ziskem** (angl. **information gain**), který je definován jaké „entropie $\mathcal{D}$ mínus vážený součet entropií $\mathcal{D}_0$ a $\mathcal{D}_1$“.

Formálně:

$$IG(\mathcal{D}, X_i) = H(\mathcal{D}) - t_0 H(\mathcal{D}_0) - t_1 H(\mathcal{D}_1)$$

kde $\mathcal{D}_0$ a $\mathcal{D}_1$ jsou podmnožiny dat $\mathcal{D}$, pro které $X_i = 0$ resp. $X_i = 1$, a t_i je podíl počtu prvků v $\mathcal{D}_i$ a $\mathcal{D}$, neboli $t_i = \frac{\#\mathcal{D}_i}{\#\mathcal{D}}$.

Konstrukce stromu pro ukázková data (1/4)

Spočítejme informační zisk pro naše tři příznaky:

- Příznak X_0: $\mathcal{D}_1 = \{1_0, 1_2, 1_3, 0_7\}$ a $\mathcal{D}_0 = \{1_1, 0_4, 0_5, 0_6\}$.
- Příznak X_1: $\mathcal{D}_1 = \{1_1, 1_3, 0_5, 0_7\}$ a $\mathcal{D}_0 = \{1_0, 1_2, 0_4, 0_6\}$.
- Příznak X_2: $\mathcal{D}_1 = \{1_1, 1_3, 0_4, 0_6\}$ a $\mathcal{D}_0 = \{1_0, 1_2, 0_5, 0_7\}$.
- Už víme, že $H(\mathcal{D}) = 1$.
- Pro X_0 dostáváme $H(\mathcal{D}_0) = H(\mathcal{D}_1) = 0.811278124459\,1328$ a tedy

$$IG(\mathcal{D}, X_0) = H(\mathcal{D}) - \frac{1}{2}H(\mathcal{D}_0) - \frac{1}{2}H(\mathcal{D}_1) = 1 - 0.811 = 0.189.$$

- Pro X_1 s X_2 postupujeme podobně. Pro ně platí $H(\mathcal{D}_0) = H(\mathcal{D}_1) = 1$ a tedy

$$IG(\mathcal{D}, X_1) = IG(\mathcal{D}, X_2) = H(\mathcal{D}) - \frac{1}{2}H(\mathcal{D}_0) - \frac{1}{2}H(\mathcal{D}_1) = 1 - 1 = 0.$$

Vítězem hladového souboje je tedy příznak X_0.

Konstrukce stromu pro ukázková data (2/4)

- Příznak X_0: $\mathcal{D}_1 = \{1_0, 1_2, 1_3, 0_7\}$ a $\mathcal{D}_0 = \{1_1, 0_4, 0_5, 0_6\}$.
- Nyní aplikujeme stejný postup dvakrát: na $\mathcal{D}_1$ a na $\mathcal{D}_0$. Nyní už ale nemá smysl dělit data podle X_0, takže budeme zkoušet jen X_1 a X_2.
- Data $\mathcal{D}_1$ příznak X_1 rozdělí na $\mathcal{D}_{11} = \{1_3, 0_7\}$ a $\mathcal{D}_{10} = \{1_0, 1_2\}$, informační zisk je tedy

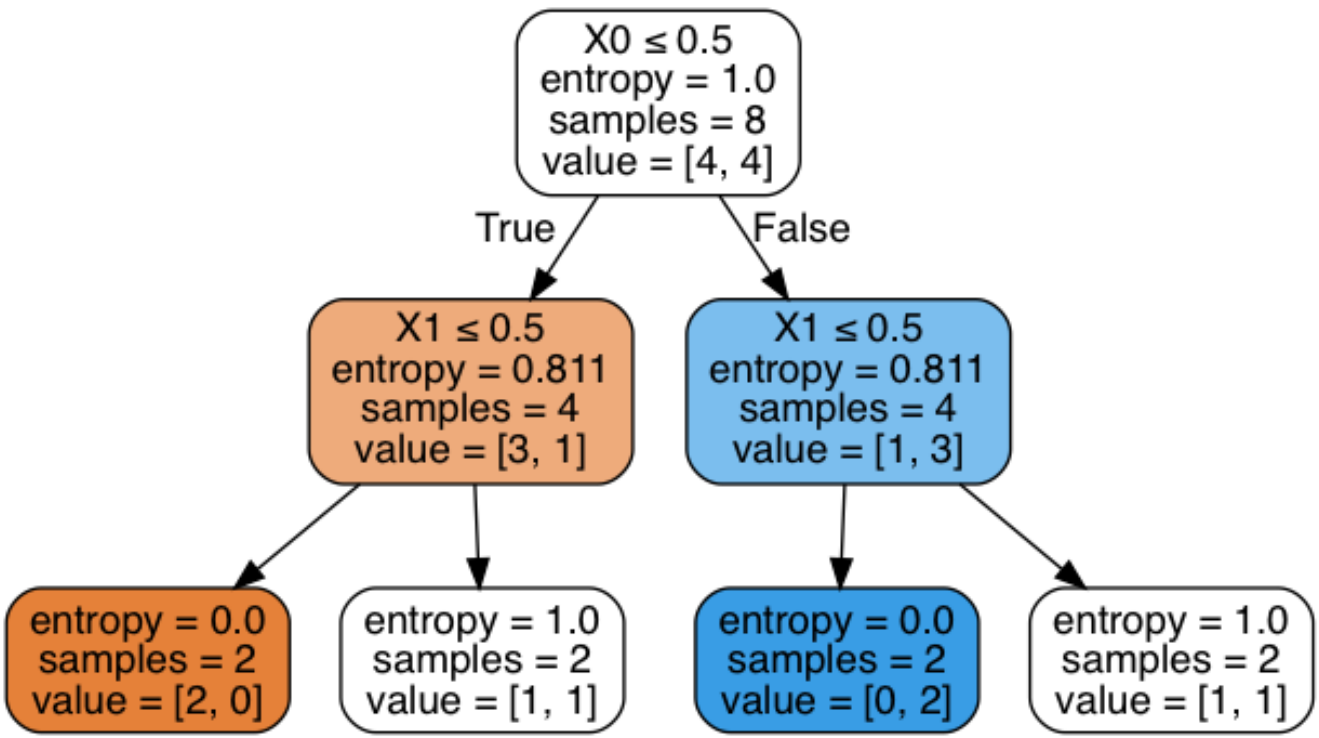
$$IG(\mathcal{D}_1, X_1) = H(\mathcal{D}_1) - \frac{1}{2}H(\mathcal{D}_{11}) - \frac{1}{2}H(\mathcal{D}_{10}) = 0.811 - \frac{1}{2}1 - \frac{1}{2}0 = 0.311\,.$$

- Pro příznak X_2 vyjde informační zisk nižší (zhruba 0.123, spočítejte si), takže hladový souboj vyhrává příznak X_1.

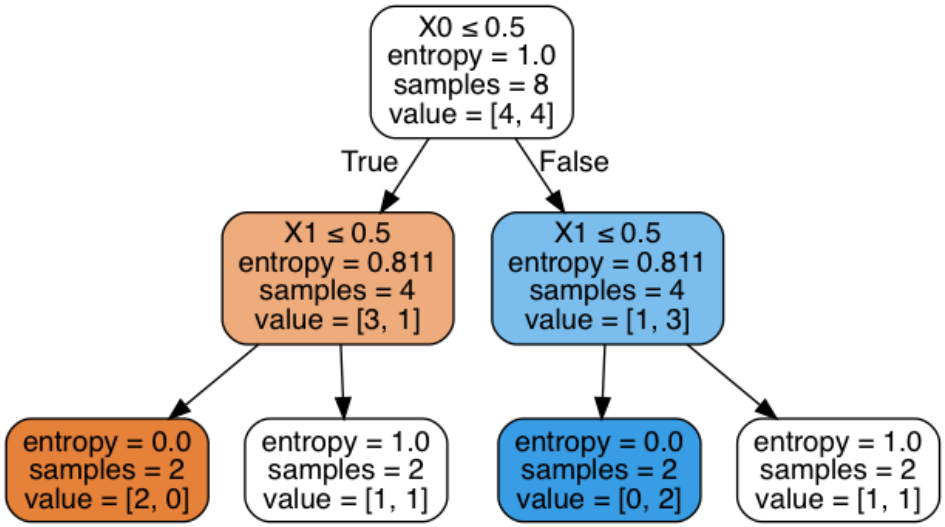
Konstrukce stromu pro ukázková data (3/4)

Takto přesně postupuje i implementace konstruování rozhodovacích stromů v knihovně `sklearn`:

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(criterion='entropy', max_depth=2)
dt.fit(X,Y)
```



Konstrukce stromu pro ukázková data (4/4)



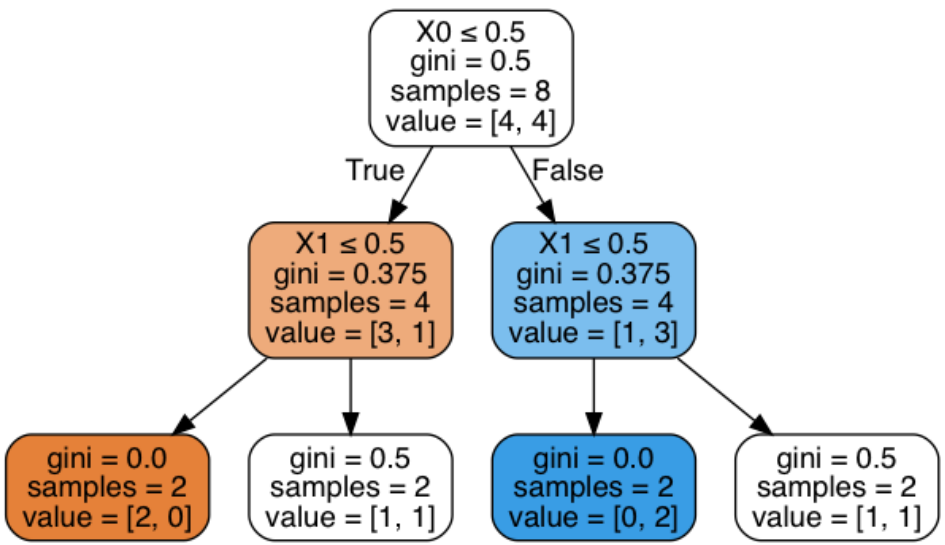
- Strom jsme zkonstruovali, ale ještě jsme k jeho jednotlivým listům nepřiřadili rozhodnutí, jestli pro daný list má být výsledek $Y = 1$ nebo $Y = 0$.
- To se dělá prostým hlasováním daty, které do daného listu „propadnou“: převažují-li nuly (první list zleva), je rozhodnutí $Y = 0$, převažují-li jedničky (třetí list), je to $Y = 1$ a při shodě (druhý a čtvrtý) přiřadíme rozhodnutí náhodně.

Gini index

- Namísto entropie se také používá **Gini index** (angl. **Gini impurity**), pro množinu $\mathcal{D}$ s k různými hodnotami:

$$GI(\mathcal{D}) = 1 - \sum_{0}^{k-1} p_i^2 = \sum_{0}^{k-1} p_i(1 - p_i).$$

- Gini index má podobné vlastnosti jako Entropie. Je to jakási míra toho, že nově přidaný prvek bude špatně klasifikován.
- Jinak ale vše funguje stejně, jen se nahradí $H(\mathcal{D})$ výrazem $GI(\mathcal{D})$.
- Pro naše data to při použití Gini indexu dopadne stejně (blbě):



Hladový $\neq$ optimální

- Získaný rozhodovací strom nebyl bezchybným modelem, neboť vždy u dvou dat rozhodl špatně.
- Jak jsme viděli dříve, je to někdy nevyhnutelná situace, řešitelná pouze tím, že se použije strom s větší hloubkou.
- V tomto případě to tak ale není, neboť existuje strom hloubky dva, který daných osm dat modeluje perfektně.
- **Tento optimální strom ale není dosažitelný hladovým algoritmem!** Přitom je celkem očividně daný podmínkou $Y = 1 \Leftrightarrow (X_1 = X_2)$.

id	Y	X_0	X_1	X_2
0	1	1	0	0
1	1	0	1	1
2	1	1	0	0
3	1	1	1	1
4	0	0	0	1
5	0	0	1	0
6	0	0	0	1
7	0	1	1	0

Konstrukce rozhodovacího stromu: shrnutí

- Konstrukce optimálního stromu je NP-úplný problém, a proto se v praxi používají hladové strategie (algoritmy ID3, C4.5 a C5 od Johna Rosse Quinlana), které ale často najdou *suboptimální* řešení.

- Hladový algoritmus funguje rekurzivně: Pro danou množinu dat najde příznak, který tuto množinu rozdělí tak, aby bylo dosaženo maximálního možného *informačního zisku* (příp. Gini indexu). Stejný postup se pak opakovaně aplikuje na podmnožiny vzniklé tímto rozdělením.

- Takto se data dělí na menší a menší podmnožiny, dokud nenastane **ukončovací podmínka**, kterou si uživatel zvolil (max. hloubka stromu, minimální počet dat v množině, minimální nutná hodnota informačního zisku, atp., viz cvičení).