

Příklady otázek – Strojové učení

Reálné otázky ze SZZ 2022–2026 (S3)

Jakub Klesa

Pokryté předměty: NAIL121 (Seminář dobývání znalostí), NPFL054 (Úvod do SU v R),
NPFL129 (Úvod do SU v Pythonu), NPGR035 (SU v počítačovém vidění)

Obsah

1	Logistická regrese	2
2	Evaluační binárního klasifikátoru	3
3	Shlukování (K-means)	3
4	Přeučení a regularizace	4
5	Rozhodovací stromy	5
6	k nejbližších sousedů	6
7	Učení pomocí kolektivních prediktorů	6
8	Evaluační binárního klasifikátoru	7
9	Naivní Bayesovský klasifikátor	7
10	Testování binárního klasifikátoru	8
11	Rozhodovací stromy	9
12	Shlukování	10
13	chi-kvadrát test (test dobré shody)	10
14	Kombinace více modelů	11
15	Evaluační metriky	12
16	Statistické testy – t-test	12
17	Lineární regrese a regularizace	13
18	Shlukování	14
19	PCA analýza	15
20	Metoda podpůrných vektorů (SVM)	16

21 Metoda nejmenších čtverců	17
22 Logistická regrese	17
23 Evaluace binárního klasifikátoru	18
24 Rozhodovací strom	19
25 k -nejbližších sousedů	20
26 Logistická regrese pro binární klasifikaci	21
27 Lineární regrese	22
28 Shlukování	23
29 EM algoritmus	25
30 Evaluace binárního klasifikátoru	26
31 Predikce pomocí tabulárních dat	27
32 Lineární regrese, L2 regularizace	28
33 Shlukování pomocí K -means	29
34 Kombinace modelů	30
35 Rozhodovací strom	31

Poznámka. Tento dokument obsahuje reálné otázky z minulých termínů SZZ (2022–2026) zařazené do okruhu S3. U otázek s obrázkem je grafika oříznutá z originálního PDF (viz odkaz na zdroj). Text, číselná data a tabulky jsou přepsané věrně. Zadání i oficiální náčrty řešení pocházejí ze sad zveřejněných na webu MFF UK:

<https://www.mff.cuni.cz/cs/studenti/bakalarske-studium/statni-zaverecne-zkousky/bakalarske-statni-zkousky-studijniho-programu-informatika>

Řešení označená „V PDF bez náčrtu řešení; vlastní vzorové řešení“ jsou vlastní, oficiální sada k dané otázce náčrt neobsahuje. Vlastní doplnění k oficiálnímu náčrtu jsou zřetelně oddělena. Pojmy a značení odpovídají přednáškám předmětu NPFL129 *Úvod do strojového učení v Pythonu* (M. Straka, J. Libovický, <https://ufal.mff.cuni.cz/courses/npfl129>); statistické testy navazují na předmět NMAI059 *Pravděpodobnost a statistika 1*.

1 Logistická regrese

(Reálná otázka: léto 2022, č. 10 – Logistická regrese)

Zadání.

1. Napište funkci, kterou predikuje model logistické regrese.
2. Jakých hodnot tato funkce nabývá a jaká je jejich interpretace?

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení.)

1. Model nejprve spočítá lineární kombinaci vstupních příznaků $z = \mathbf{w}^T \mathbf{x} + b$ a na ni aplikuje logistickou (sigmoidní) funkci $\sigma(z) = \frac{1}{1 + e^{-z}}$. Predikuje tedy

$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}.$$

2. Hodnoty leží v intervalu $(0, 1)$ a interpretují se jako pravděpodobnost příslušnosti k pozitivní třídě, $P(y=1 | \mathbf{x})$. Vlastní klasifikaci získáme prahováním (typicky práh 0,5: je-li $\hat{y} \geq 0,5$, predikujeme třídu 1, jinak 0).

2 Evaluace binárního klasifikátoru

(Reálná otázka: léto 2022, č. 11 – Evaluace binárního klasifikátoru)

Zadání.

Pro evaluaci binárního klasifikátoru byla použita množina 1000 testovacích příkladů, ve které je 50 % příkladů negativních. Klasifikátor má citlivost (sensitivity) 60 % a úspěšnost (accuracy) 70 %.

1. Napište kompletní matici konfuze (confusion matrix).
2. Vypočtěte specifičnost (specificity) klasifikátoru.

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.)

Z rozdělení je $P = 500$ pozitivních a $N = 500$ negativních příkladů. Citlivost (sensitivity, recall) je $TP/P = 0,6 \Rightarrow TP = 300$, $FN = 200$. Úspěšnost (accuracy) je $(TP + TN)/1000 = 0,7 \Rightarrow TP + TN = 700$, odtud $TN = 400$ a $FP = N - TN = 500 - 400 = 100$.

1. Matice konfuze (řádky = skutečná třída, sloupce = predikovaná třída):

	predikováno pozitivní	predikováno negativní
skutečně pozitivní	TP = 300	FN = 200
skutečně negativní	FP = 100	TN = 400

2. Specifičnost (specificity) $= \frac{TN}{N} = \frac{TN}{TN + FP} = \frac{400}{500} = 0,8 = 80 \%$.

3 Shlukování (K-means)

(Reálná otázka: léto 2022, č. 12 – Shlukování (K-means))

Zadání.

Ručně provedeme první iteraci algoritmu K-means pro $K = 2$ na množině šesti příkladů s příznaky A_1, A_2 :

příklad	A_1	A_2
1	2	4
2	1	3
3	0	4
4	5	2
5	6	2
6	4	0

Náhodně jsme příklady přiřadili do dvou shluků: $C_1 = \{2, 3, 4\}$, $C_2 = \{1, 5, 6\}$.

1. Spočítejte pro každý shluk jeho centroid.
2. Přiřaďte každý příklad k nejbližšímu centroidu (euklidovská vzdálenost) a uveďte prvky nově vzniklých shluků.

Řešení. (*V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.*)

1. Centroid je průměr bodů shluku:

$$c_1 = \left(\frac{1+0+5}{3}, \frac{3+4+2}{3} \right) = (2, 3), \quad c_2 = \left(\frac{2+6+4}{3}, \frac{4+2+0}{3} \right) = (4, 2).$$

2. Pro každý bod porovnáme čtverce vzdáleností k $c_1 = (2, 3)$ a $c_2 = (4, 2)$:

bod	(A_1, A_2)	$\ \cdot - c_1\ ^2$	$\ \cdot - c_2\ ^2$	shluk
1	(2, 4)	1	8	C_1
2	(1, 3)	1	10	C_1
3	(0, 4)	5	20	C_1
4	(5, 2)	10	1	C_2
5	(6, 2)	17	4	C_2
6	(4, 0)	13	4	C_2

Nově vzniklé shluky: $C_1 = \{1, 2, 3\}$, $C_2 = \{4, 5, 6\}$.

4 Přeučení a regularizace

(*Reálná otázka: podzim 2022, č. 18 – Přeučení a regularizace*)

Zadání.

Předpokládejme, že chceme trénovat lineární model pro regresi.

1. Definujte střední čtvercovou chybu (MSE). Definujte upravenou chybovou funkci používanou při L2 regularizaci.
2. Mějme dva lineární modely M_1 , M_2 pro data se čtyřmi atributy:

$$M_1(x) = x_1 - 2x_2 + 4x_3 - x_4 + 3, \quad M_2(x) = x_2 - 5x_3 + x_4 - 1.$$

Oba modely mají pro daná data stejnou MSE. Který má menší hodnotu upravené chybové funkce pro L2 regularizaci?

3. Popište alespoň jeden další způsob pro zlepšení generalizačních schopností modelů strojového učení.

Řešení. (*V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.*)

1. Střední čtvercová chyba je

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$

kde $\hat{y}_i$ je predikce modelu pro i -tý vstup a y_i skutečná hodnota. Při L2 regularizaci se k chybě přidá penalizace za velikost vah:

$$\text{MSE} + \lambda \sum_j w_j^2, \quad \lambda > 0,$$

kde w_j jsou váhy modelu a λ je síla regularizace. Absolutní člen (bias / intercept) se obvykle neregularizuje.

2. Regularizační člen závisí jen na vahách (bez biasu). Pro M_1 jsou váhy $(1, -2, 4, -1)$, takže $\sum w_j^2 = 1 + 4 + 16 + 1 = 22$. Pro M_2 jsou váhy $(0, 1, -5, 1)$, takže $\sum w_j^2 = 0 + 1 + 25 + 1 = 27$. Při stejné MSE má proto *menší* regularizovanou chybu model M_1 ($22 < 27$).
Poznámka: kdyby se regularizoval i absolutní člen, vyšlo by M_1 : $22 + 3^2 = 31$ a M_2 : $27 + (-1)^2 = 28$, a vyhrál by M_2 . Právě proto je podstatné, že se bias neregularizuje.
3. Generalizaci lze zlepšit například získáním více trénovacích dat, výběrem či redukcí příznaků (méně parametrů), předčasným zastavením (early stopping), dropoutem u neuronových sítí, kombinováním modelů (ensembling) nebo jinou regularizací (L1). Vhodné hyperparametry (např. λ) se volí křížovou validací.

5 Rozhodovací stromy

(Reálná otázka: podzim 2022, č. 19 – Rozhodovací stromy)

Zadání.

1. Definujte rozhodovací strom. Popište algoritmus pro trénování rozhodovacího stromu.
2. Definujte kritéria pro výběr atributu pro větvení pro klasifikaci a pro regresi (alespoň jedno pro klasifikaci a alespoň jedno pro regresi).
3. Mějme data s kategoričnými atributy A_1 , A_2 a cílovou hodnotou C (tabulka níže). Podle kterého atributu budeme data dělit v kořeni rozhodovacího stromu pro *regresi* a proč?

A_1	A_2	C
0	0	5
0	0	3
0	1	4
1	0	6
1	1	2
1	1	4

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.)

1. Rozhodovací strom je stromový model, jehož vnitřní uzly testují atribut, větve odpovídají jeho hodnotám a listy nesou predikci (třidu nebo číselnou hodnotu). Trénuje se hladově shora dolů (rozděl a panuj): v každém uzlu se vybere atribut podle zvoleného kritéria, data se podle něj rozdělí a rekurzivně se pokračuje, dokud nejsou listy „čisté“, nedojdou atributy nebo není splněna zastavovací podmínka.
2. Pro *klasifikaci*: maximalizace informačního zisku (pokles entropie) nebo minimalizace Gini indexu. Pro *regresi*: minimalizace vážené sumy čtvercových odchylek (SSE) v dětech, ekvivalentně maximalizace redukce rozptylu.
3. Pro regresi počítáme redukci rozptylu (resp. SSE). Celkový průměr je $\bar{C} = 4$ a SSE rodiče = 10.
 - Dělení podle A_1 : $A_1=0 \rightarrow \{5, 3, 4\}$, průměr 4, SSE 2; $A_1=1 \rightarrow \{6, 2, 4\}$, průměr 4, SSE 8. Celkem SSE = 10, redukce = 0.
 - Dělení podle A_2 : $A_2=0 \rightarrow \{5, 3, 6\}$, průměr $\approx 4,67$, SSE $\approx 4,67$; $A_2=1 \rightarrow \{4, 2, 4\}$, průměr $\approx 3,33$, SSE $\approx 2,67$. Celkem SSE $\approx 7,33$, redukce $\approx 2,67$.

V kořeni proto dělíme podle A_2 (větší redukce rozptylu). Atribut A_1 nedává žádnou redukci, protože obě jeho větve mají stejný průměr 4 jako rodič.

6 k nejbližších sousedů

(Reálná otázka: podzim 2022, č. 20 – k nejbližších sousedů)

Zadání.

1. Popište obecně fungování techniky k nejbližších sousedů (k -NN) pro klasifikaci i pro regresi: Jak funguje trénování modelu? Jak probíhá předpovídání?
2. Mějme data se dvěma příznaky A_1 , A_2 a dvěma třídami $C \in \{0, 1\}$ (tabulka níže). Rozhodněte, jak bude pomocí k -NN pro $k = 3$ ohodnocen nový vstup $(5, 5)$. Používejte Manhattanskou (L1) vzdálenost.
3. Změní se odpověď při změně k ? Uveďte alespoň jeden příklad k , pro které je odpověď jiná (pokud existuje).

A_1	A_2	C
0	3	0
2	2	0
3	0	0
4	2	0
6	6	1
4	5	1
1	1	1

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.)

1. k -NN je „líný“ (instance-based) model: „trénování“ pouze uloží trénovací data. Predikce pro nový bod: najdi k nejbližších trénovacích bodů podle zvolené metriky; pro klasifikaci vrať většinovou třídu mezi nimi, pro regresi vrať průměr (případně vážený) jejich cílových hodnot.
2. Manhattanské vzdálenosti bodu $(5, 5)$ od dat:

bod	L1 vzdálenost
$(4, 5)$	1
$(6, 6)$	2
$(4, 2)$	4
$(2, 2)$	6
$(0, 3)$	7
$(3, 0)$	7
$(1, 1)$	8

Tři nejbližší jsou $(4, 5)$ s $C=1$, $(6, 6)$ s $C=1$ a $(4, 2)$ s $C=0$, tedy třídy $\{1, 1, 0\}$, většina 1. Pro $k = 3$ je $(5, 5)$ klasifikován jako třída **1**.

3. Ano, odpověď se může změnit. Pro $k = 5$ je pět nejbližších bodů $(4, 5):1$, $(6, 6):1$, $(4, 2):0$, $(2, 2):0$, $(0, 3):0$, tedy třídy $\{1, 1, 0, 0, 0\}$, většina 0. Pro $k = 5$ (a stejně tak $k = 7$) tedy vyjde třída 0. (Na pátém místě je remíza vzdáleností: $(0, 3)$ i $(3, 0)$ mají vzdálenost 7, oba ale patří do třídy 0, takže na výsledku nezáleží.)

7 Učení pomocí kolektivních prediktorů

(Reálná otázka: jaro 2023, č. 6 – Učení pomocí kolektivních prediktorů)

Zadání.

1. Vysvětlete princip trénování metodami *bagging* a *boosting*.
2. Popište trénovací algoritmus *nahodilých lesů*.
3. Uveďte konkrétní příklad algoritmu typu *boosting*. Stručně vysvětlete hlavní myšlenku.

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení.)

1. *Bagging* (bootstrap aggregating): natrénujeme M kopií modelu, každou na vlastním *bootstrapovém* vzorku (losování N příkladů s opakováním), a jejich predikce agregujeme – u klasifikace hlasováním, u regrese průměrem; snižuje rozptyl. *Boosting*: modely trénujeme *sekvenčně*, každý další klade větší váhu na příklady, které dosavadní kombinace klasifikuje špatně; výsledkem je vážený součet slabých klasifikátorů.
2. *Náhodný les* (random forest): bagging rozhodovacích stromů s jednou přidanou náhodností – v každém uzlu se nejlepší řez hledá jen z náhodné podmnožiny příznaků (typicky $\sqrt{D}$). Stromy se nechávají přerůst (neprořezané); predikce hlasováním (klasifikace) nebo průměrem (regrese).
3. Konkrétní boosting: *AdaBoost*. Začne s rovnoměrnými váhami $w_i = 1/N$; v kole m natrénuje slabý klasifikátor h_m , spočte jeho váženou chybu ε_m , hlas $\alpha_m = \frac{1}{2} \ln \frac{1-\varepsilon_m}{\varepsilon_m}$ a zvýší váhy špatně klasifikovaných příkladů ($w_i \propto w_i e^{-\alpha_m t_i h_m(x_i)}$). Finální predikce je $\text{sign}(\sum_m \alpha_m h_m)$.

8 Evaluace binárního klasifikátoru

(Reálná otázka: jaro 2023, č. 7 – Evaluace binárního klasifikátoru)

Zadání.

Předpokládejme, že binární klasifikátor vykazuje sensitivitu 0.8 a specifitu 0.9. Při jeho testování bylo použito 10000 příkladů, z nichž 500 bylo pozitivních.

Poznámka: Sensitivita bývá též nazývána *recall* nebo *true positive rate* a specifita se někdy nazývá též *selectivity* neboli *true negative rate*.

1. Na základě uvedených údajů spočítejte celou matici konfuze.
2. Uvažujme nahodile zvolený testovací příklad, který je predikován jako pozitivní. Určete pravděpodobnost, že tento příklad je skutečně pozitivní.
3. Jaká je přesnost (*accuracy*) daného klasifikátoru?

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.)

1. Pozitivních je 500, negativních 9500. Sensitivita (*recall*) = $TP/500 = 0,8 \Rightarrow TP = 400$, $FN = 100$; specifita = $TN/9500 = 0,9 \Rightarrow TN = 8550$, $FP = 950$.
2. Hledaná pravděpodobnost je *precision*: $\frac{TP}{TP + FP} = \frac{400}{1350} \approx 0,296$.
3. $accuracy = \frac{TP + TN}{10000} = \frac{8950}{10000} = 0,895$.

9 Naivní Bayesovský klasifikátor

(Reálná otázka: jaro 2023, č. 16 – Naivní Bayesovský klasifikátor)

Zadání.

1. Popište princip naivního Bayesovského klasifikátoru. Na základě čeho (vzorec) přiřazuje objekty do tříd?

2. Popište význam jednotlivých pravděpodobností v Bayesově vzorci. V čem spočívá naivnost klasifikátoru?
3. Použitím Bayesovského klasifikátoru určete, zda student půjde do školy, jestliže vstane pozdě, nemá oblíbenou přednášku a prší. Pravděpodobnost určete na základě následujících pozorování.

Den	Vstanu	Je oblíbená přednáška?	Počasí	Jdu do školy
1	Brzo	Ne	Slunce	Ne
2	Normálně	Ne	Déšť	Ne
3	Brzo	Ne	Slunce	Ano
4	Pozdě	Ne	Slunce	Ano
5	Pozdě	Ano	Slunce	Ano
6	Pozdě	Ano	Déšť	Ne
7	Brzo	Ano	Déšť	Ano
8	Normálně	Ne	Slunce	Ne
9	Normálně	Ano	Slunce	Ano
10	Pozdě	Ano	Slunce	Ano
11	Normálně	Ano	Déšť	Ano
12	Brzo	Ne	Déšť	Ano
13	Brzo	Ano	Slunce	Ano

V originálním PDF je hodnota psána „Déšť“, správně „Děšť“; tabulka je přepsána věrně.

Řešení. (*V PDF bez náčrtu řešení; vlastní vzorové řešení, četnosti a skóre ověřeny numericky.*)

1. Klasifikátor přiřadí třídu maximalizující $P(c) \prod_j P(x_j | c)$ – Bayesovo pravidlo s *naivním* předpokladem podmíněné nezávislosti příznaků. Pravděpodobnosti se odhadnou relativními četnostmi z tabulky.
2. $P(c)$ jsou apriorní pravděpodobnosti tříd, $P(x_j | c)$ věrohodnosti jednotlivých příznaků, jmenovatel $P(x_1, \dots, x_D)$ je normalizace (stejná pro všechny třídy, pro výběr maxima ji lze vynechat). „Naivnost“ je v předpokladu, že příznaky jsou při dané třídě navzájem nezávislé.
3. Z 13 dnů jde do školy $9\times$, nejde $4\times$: $P(\text{Ano}) = \frac{9}{13}$, $P(\text{Ne}) = \frac{4}{13}$. Pro dotaz (Vstanu=Pozdě, oblíbená=Ne, Počasí=Děšť):

$$\text{Ano} : \frac{9}{13} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} = \frac{1}{39} \approx 0,026, \quad \text{Ne} : \frac{4}{13} \cdot \frac{1}{4} \cdot \frac{3}{4} \cdot \frac{2}{4} = \frac{3}{104} \approx 0,029.$$

Větší je skóre třídy **Ne**, takže klasifikátor predikuje, že student *do školy nepůjde* (po normalizaci $P(\text{Ne} | \text{data}) \approx 53\%$).

10 Testování binárního klasifikátoru

(*Reálná otázka: léto 2023, č. 6 – Testování binárního klasifikátoru*)

Zadání.

Předpokládejme, že k testování binárního klasifikátoru byla použita testovací sada ve složení 100 pozitivních a 400 negativních příkladů.

- a) Ukázalo se, že ROC křivka prochází bodem $\text{TPR} = \text{FPR} = 20\%$. Vypočítejte hodnotu precision v tomto bodě.
- b) Jiný klasifikátor vykazuje konstantní hodnotu precision = 50 %. Znázorněte odpovídající ROC křivku.

Použité zkratky: ROC — Receiver Operating Characteristic curve; TPR — true-positive rate (též: sensitivity neboli recall); FPR — false-positive rate (též: fall-out).

Řešení. (*V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.*)

a) $\text{TPR} = 0,2 \Rightarrow \text{TP} = 0,2 \cdot 100 = 20$; $\text{FPR} = 0,2 \Rightarrow \text{FP} = 0,2 \cdot 400 = 80$. Precision

$$= \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{20}{100} = 0,2.$$

b) Konstantní precision $\frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{1}{2}$ znamená $\text{FP} = \text{TP}$ při každém prahu. Protože $\text{TPR} = \text{TP}/100$ a $\text{FPR} = \text{FP}/400 = \text{TP}/400$, platí $\text{TPR} = 4\text{FPR}$. ROC křivka je tedy úsečka z $(0, 0)$ do bodu $(\text{FPR}, \text{TPR}) = (0,25, 1)$ (leží nad diagonálou náhodného hádání).

11 Rozhodovací stromy

(*Reálná otázka: léto 2023, č. 7 – Rozhodovací stromy*)

Zadání.

Chceme trénovat rozhodovací strom na datech v tabulce níže, která obsahují dva kategorické atributy x_1 a x_2 . Cílem je předpovědět kategorickou hodnotu y .

Nápověda: Při řešení můžete potřebovat hodnoty některých logaritmů. Můžete použít následující aproximace: $\log_2 \frac{1}{3} = -1,5$, $\log_2 \frac{2}{3} = -0,6$.

x_1	x_2	y
0	0	0
1	0	1
0	1	1
2	1	0
0	0	1
1	1	1
2	0	0
0	1	0

1. Popište stručně algoritmus trénování rozhodovacího stromu a kritéria pro větvení.
2. Určete, podle jakého atributu se bude strom větvit v kořeni stromu. Proč?
3. Je možné zadaná trénovací data klasifikovat se 100 % přesností (accuracy)? Vysvětlete.

Řešení. (*V PDF bez náčrtu řešení; vlastní vzorové řešení, zisky ověřeny numericky.*)

1. Strom se staví hladově shora dolů (rozděl a panuj): v každém uzlu se zvolí atribut s největším *informačním ziskem* (největším poklesem entropie), příklady se podle něj rozdělí a rekurzivně se pokračuje, dokud nejsou listy čisté (nebo nedojdou atributy).
2. Entropie kořene je $B(4, 4) = 1$ bit. Dělení podle x_1 : $x_1=0$ dává poměr $(2, 2)$, $x_1=1$ poměr $(0, 2)$, $x_1=2$ poměr $(2, 0)$; vážená entropie $\frac{4}{8} \cdot 1 + \frac{2}{8} \cdot 0 + \frac{2}{8} \cdot 0 = 0,5$, tedy zisk 0,5. Dělení podle x_2 : obě hodnoty dají poměr $(2, 2)$, zisk 0. V kořeni se proto větví podle x_1 .
3. Ne. Stejný vstup se v datech opakuje s *různým* cílem: $(x_1, x_2) = (0, 0)$ má jednu $y=0$ a jednu $y=1$, podobně $(0, 1)$. Tyto příklady nelze odlišit žádným stromem, takže lze dosáhnout nejvýše $6/8 = 75\%$ přesnosti.

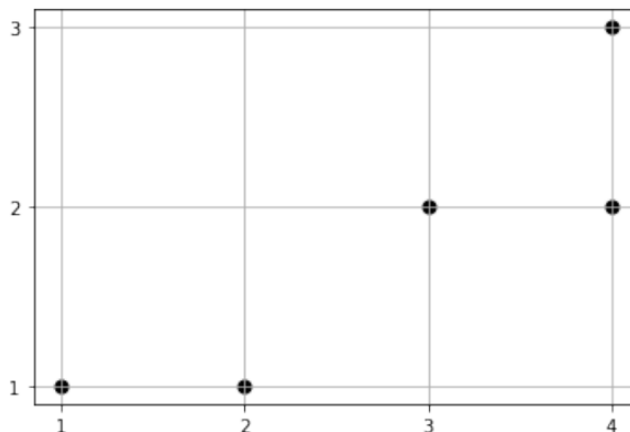
12 Shlukování

(Reálná otázka: léto 2023, č. 8 – Shlukování)

Zadání.

Uvažujme data zadaná v tabulce níže a zobrazená v obrázku. Budeme na ně chtít aplikovat shlukovací algoritmy.

x_1	x_2
1	1
2	1
3	2
4	3
4	2



1. Nejprve chceme aplikovat algoritmus k -means pro $k = 2$. Na počátku jsme náhodně zvolili jako středy shluků body $(1, 1)$ a $(3, 2)$. Jaké budou nové středy shluků po jedné iteraci algoritmu k -means?
2. Jak se změní polohy středů v dalších iteracích tohoto algoritmu?
3. Popište některý z algoritmů pro hierarchické shlukování. Jaký by byl jeho výsledek na uvedených datech?

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení, iterace ověřena numericky.)

1. *Přiřazení* (porovnáme vzdálenosti k oběma středům): $(1, 1) \rightarrow \mu_1$, $(2, 1) \rightarrow \mu_1$ (vzdálenost $1 < \sqrt{2}$), $(3, 2) \rightarrow \mu_2$, $(4, 3) \rightarrow \mu_2$, $(4, 2) \rightarrow \mu_2$. *Přepočít* (průměry shluků): $\mu_1 = (1,5, 1)$ a $\mu_2 = (\frac{3+4+4}{3}, \frac{2+3+2}{3}) = (\frac{11}{3}, \frac{7}{3})$.
2. Opakované přiřazení k novým středům dá totéž rozdělení, takže se středy už nemění – algoritmus zkonvergoval (po jediné iteraci).
3. *Aglomerativní* hierarchické shlukování: začneme s pěti jednobodovými shluky a opakovaně slučujeme dvojici nejbližších (single linkage = vzdálenost nejbližších bodů). Vzdálenost 1 mají tři dvojice: $(1, 1)$ – $(2, 1)$, $(3, 2)$ – $(4, 2)$ a $(4, 2)$ – $(4, 3)$; ve výšce 1 tak vzniknou skupiny $\{(1, 1), (2, 1)\}$ a $\{(3, 2), (4, 2), (4, 3)\}$, které se nakonec spojí ve výšce $\sqrt{2}$ (vzdálenost $(2, 1)$ – $(3, 2)$). Řez dendrogramu na dva shluky dá totéž rozdělení jako k -means: $\{(1, 1), (2, 1)\}$ a $\{(3, 2), (4, 2), (4, 3)\}$.

13 chí-kvadrát test (test dobré shody)

(Reálná otázka: podzim 2023, č. 29 – chí-kvadrát test (test dobré shody))

Zadání.

1. Popište χ^2 test (test dobré shody). Jaká je jeho nulová a alternativní hypotéza? Jak se spočítá hodnota testové statistiky? Jaké má tato testová statistika pravděpodobnostní rozdělení?

2. Majitel obchodu měl během pěti pracovních dnů jednoho týdne celkem 100 zákazníků. Počty zákazníků v jednotlivých dnech (pondělí až pátek) byly postupně 25, 30, 15, 14 a 16. Pomocí χ^2 testu rozhodněte, zda se počty zákazníků v jednotlivých dnech statisticky významně liší. *Nápověda:* Kritické hodnoty testové statistiky na hladině významnosti $\alpha = 0.05$ pro 3–6 stupňů volnosti (df) jsou (postupně) 7,8 (df = 3), 9,5 (df = 4), 11,1 (df = 5) a 12,6 (df = 6).

Náčrt řešení.

1. Nulová hypotéza je, že pozorované četnosti jednotlivých kategorií odpovídají jejich očekávaným četnostem. Alternativní hypotéza je, že neodpovídají. Testová statistika se vypočítá jako

$$\sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i},$$

kde O_i jsou pozorované četnosti a E_i jsou očekávané četnosti. Pokud nulová hypotéza platí, má testová statistika χ^2 rozdělení s $n - 1$ stupni volnosti, kde n je počet různých kategorií.

2. Zajímá nás, jestli počet zákazníků je každý den stejný. Očekávané četnosti jsou tedy $E_i = 20$. Dosazením do vzorce dostaneme hodnotu testové statistiky

$$\begin{aligned} & \frac{(25 - 20)^2}{20} + \frac{(30 - 20)^2}{20} + \frac{(15 - 20)^2}{20} + \frac{(14 - 20)^2}{20} + \frac{(16 - 20)^2}{20} \\ &= \frac{25 + 100 + 25 + 36 + 16}{20} = \frac{202}{20} = 10,1. \end{aligned}$$

Máme 5 kategorií, kritická hodnota testové statistiky se 4 stupni volnosti je 9,5, rozdíl v počtu zákazníků tedy je statisticky významný.

14 Kombinace více modelů

(Reálná otázka: podzim 2023, č. 30 – Kombinace více modelů)

Zadání.

1. Popište hlavní myšlenky technik bagging a boosting. Popište podrobněji jeden vybraný konkrétní algoritmus pro boosting.
2. Jak funguje klasifikátor založený na náhodných lesech (random forest)? Jakým způsobem se trénuje?
3. Porovnejte rozhodovací stromy a náhodné lesy. Jaké jsou výhody a nevýhody těchto modelů?

Náčrt řešení.

1. Podstatou baggingu je vytvoření více modelů trénovaných na různých podmnožinách (bootstrap)¹ trénovací množiny. Tyto podmnožiny jsou typicky sámkovány s opakováním. Hlavním myšlenkou boostingu je trénování několika modelů postupně s tím, že další modely jsou trénovány na vážených datech, kde instance nesprávně klasifikované aktuálním klasifikátorem mají větší váhu. Příkladem může být např. algoritmus AdaBoost.
2. Náhodné lesy obsahují množinu mnoha rozhodovacích stromů, které hlasují o celkovém výsledku klasifikace/regrese. Jednotlivé rozhodovací stromy jsou trénované na podmnožině instancí trénovací množiny a na podmnožině příznaků.

¹V originálním PDF je zde překlep: „bootrap“; zde uvedeno korektně.

3. Výhodou rozhodovacích stromů je jejich interpretabilita a rychlost trénování. Výhodou náhodných lesů je jejich (typicky) větší přesnost a schopnost generalizace.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Bootstrapový vzorek má stejnou velikost N jako trénovací množina a losuje se s opakováním (asi 37 % příkladů v něm chybí). Bagging snižuje hlavně rozptyl, boosting hlavně vychýlení. AdaBoost dá slabému klasifikátoru s váženou chybou ε_m hlas $\alpha_m = \frac{1}{2} \ln \frac{1-\varepsilon_m}{\varepsilon_m}$ a váhy chybně klasifikovaných příkladů vynásobí e^{α_m} , správně klasifikovaných $e^{-\alpha_m}$ (pak normalizuje). V náhodném lese se podmnožina příznaků losuje *v každém uzlu* znovu (typicky $\sqrt{D}$ příznaků) a stromy se neprořezávají; to dekoreluje jejich chyby. Nevýhody lesa oproti jednomu stromu: ztráta interpretovatelnosti a vyšší nároky na čas i paměť při trénování i predikci; nevýhodou jednoho stromu je vysoký rozptyl a sklon k přeučení.

15 Evaluační metriky

(Reálná otázka: podzim 2023, č. 31 – Evaluační metriky)

Zadání.

1. Definujte základní evaluační metriky pro binární klasifikátor – úspěšnost (accuracy), citlivost (sensitivity, recall), přesnost (precision), F1 skóre.
2. Chceme vytvořit systém pro detekci podvodných plateb kreditní kartou. Máme data, kde jsou miliony plateb, z nichž jen několik stovek jsou platby podvodné. Jaké metriky použijete pro vyhodnocení takového systému? Jaké metriky naopak nebudou moc vhodné? Odpověď zdůvodněte.
3. Pro výše zmíněnou úlohu jsme vytvořili dva systémy. Jeden z nich má citlivost 0,99 a přesnost 0,2. Druhý má citlivost 0,9 a přesnost 0,95. Vysvětlete, co tyto hodnoty prakticky znamenají. Který ze systémů byste použili? Své rozhodnutí zdůvodněte.

Náčrt řešení.

1. Definujme TP/TN/FP/FN jako počty true/false positive/negative. Správnost potom je $(TP + TN)/(TP + FP + TN + FN)$, citlivost je $TP/(TP + FN)$, přesnost je $TP/(TP + FP)$ a F1 skóre je harmonický průměr citlivosti a přesnosti $= 2TP/(2TP + FP + FN)$.
2. Pro vyhodnocení systému je vhodné použít dvojici citlivost a přesnost. Dá se také použít F1 skóre nebo AUC. Nevhodná naopak bude správnost vzhledem k výrazně nevyváženému množství instancí v jednotlivých třídách.
3. První model odhalí 99 % podvodných plateb, na druhou stranu ale jen 20 % z ohlášených podezřelých plateb budou platby podvodné. Druhý model sice odhalí jen 90 % podvodných plateb, ale 95 % z podezřelých plateb opravdu podvody budou. V případě použití prvního modelu tedy bude potřeba prošetřit cca 5krát více případných podvodů, což může vést ke zbytečnému obtěžování zákazníků a i vyšším nákladům na prošetřování. Může tedy být vhodnější použít model druhý. Na druhou stranu, pokud bychom chtěli mít větší jistotu, že podvody odhalíme a výše zmíněné nevýhody nám nevadí, je vhodnější model první. Neexistuje tu jednoznačně správná odpověď, důležitá je především argumentace.

16 Statistické testy – t-test

(Reálná otázka: jaro 2024, č. 26 – Statistické testy – t-test)

Zadání.

Výrobce sušenek vyrábí sušenky s deklarovanou hmotností 100 gramů. Při kontrole kvality bylo náhodně vybráno a převáženo 100 sušenek, které měly průměrnou hmotnost 102 gramů se směrodatnou odchylkou 2 gramy. Pomocí t-testu chceme statisticky vyhodnotit, jestli se průměrná hmotnost sušenek liší od deklarované hmotnosti.

1. Jaká je nulová a alternativní hypotéza jednovýběrového t-testu?
2. Spočítejte hodnotu testové statistiky pro data zadaná výše. Jaké má tato testová statistika pravděpodobnostní rozdělení?
3. Liší se statisticky významně hmotnost sušenek od deklarované hmotnosti? *Nápověda:* Kritická hodnota testové statistiky na hladině významnosti $\alpha = 0.05$ je cca 2.

Náčrt řešení.

1. Nulová hypotéza je $H_0 : \bar{X} = \mu$, kde $\bar{X}$ je průměrná hmotnost sušenek a μ je testovaná střední hodnota. Alternativní hypotéza je $H_1 : \bar{X} \neq \mu$.
2. $t = \frac{102 - 100}{2/\sqrt{100}} = 10$, má t rozdělení s 99 stupni volnosti.
3. Spočítaná t-hodnota je větší než kritická hodnota, zamítáme tedy nulovou hypotézu. Průměrná hmotnost sušenek se liší od deklarované hmotnosti.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Hypotézy se formulují pro střední hodnotu μ rozdělení hmotnosti, ne pro výběrový průměr $\bar{X}$ (ten je statistika spočtená z dat): $H_0 : \mu = 100$, $H_1 : \mu \neq 100$ (oboustranný test). Statistika $t = (\bar{X} - \mu_0)/(s/\sqrt{n})$ má za H_0 Studentovo rozdělení s $n - 1 = 99$ stupni volnosti; přesná kritická hodnota je $t_{99;0,975} \approx 1,98$. Protože $|t| = 10$ leží hluboko v kritickém oboru, H_0 na hladině 0,05 zamítáme (p-hodnota je prakticky nulová).

17 Lineární regrese a regularizace

(Reálná otázka: jaro 2024, č. 27 – Lineární regrese a regularizace)

Zadání.

Chceme natrénovat lineární model tvaru $y = \beta_1 x_1 + \beta_2 x_2 + \beta_0$ na datech níže. Při trénování chceme použít gradientní metodu a L2 regularizaci. Použijeme učicí konstantu $\alpha = 0.1$ a regularizační konstantu $\lambda = 0.5$.

x_1	x_2	y
0	2	3
2	3	9
1	1	2
2	0	2

1. Definujte chybovou funkci pro model zadaný podle parametrů nahoře (součet čtverců s L2 regularizačním členem).
2. Předpokládejme, že aktuální odhad koeficientů modelu je $\beta_1 = 1$, $\beta_2 = 3$, $\beta_0 = -1$. Proveďte jeden krok gradientní metody pro nastavení koeficientů. Jaké jsou jejich nové hodnoty?

Náčrt řešení.

1. Chybová funkce je typicky $RSS = \sum_{i=1}^N (y_i - \hat{y}_i)^2$, pro L2 regularizaci přičteme ještě člen $\lambda \sum_{i=1}^m \beta_i^2$ (N je počet trénovacích vzorů, $m+1$ je počet koeficientů modelu, $\hat{y}_i$ je výstup modelu pro i -tý vstup). Koeficient β_0 se většinou v regularizaci neuvažuje.
2. Je potřeba spočítat gradient chybové funkce podle jednotlivých parametrů. Spočítáme ho pro jeden trénovací vzor, pro všechny vzory to bude jen součet. Derivace $\frac{dRSS}{d\beta_0} = -2(y_i - \hat{y}_i)$, derivace $\frac{dRSS}{d\beta_i} = -2(y_i - \hat{y}_i)x_i$. Nesmíme zapomenout také na derivaci regularizačního členu podle β_i – ta je $\lambda 2\beta_i = \beta_i$. Dosazením dostaneme, že derivace RSS podle β_0 je $4+2+2+(-2) = 6$, derivace podle β_1 je 2 a derivace podle β_2 je 16. Přičtením regularizačního členu dostaneme, že derivace chybové funkce podle β_1 je 3 a podle β_2 je 19. Hodnoty v dalším kroku tedy jsou $\beta_0 = \beta_0 - \alpha \cdot 6 = -1.6$, $\beta_1 = \beta_1 - \alpha \cdot 3 = 0.7$ a $\beta_2 = \beta_2 - \alpha \cdot 19 = 1.1$.

18 Shlukování

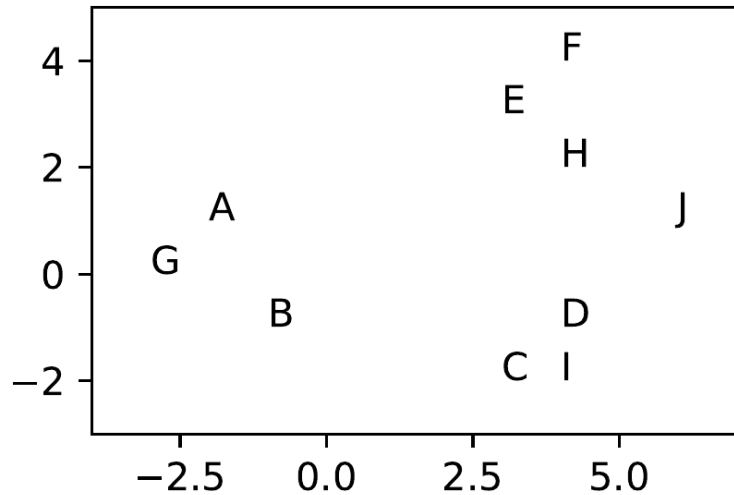
(Reálná otázka: jaro 2024, č. 28 – Shlukování)

Zadání.

Máme data zadaná v tabulce níže a graficky znázorněná v obrázku. Chtěli bychom je rozdělit do třech shluků pomocí algoritmu k -means.

Poznámka: Jména nejsou součástí dat – slouží jen pro vizualizaci a pro zjednodušení formulace odpovědí.

Jméno	x	y
A	-2	1
B	-1	-1
C	3	-2
D	4	-1
E	3	3
F	4	4
G	-3	0
H	4	2
I	4	-2
J	6	1



1. Popište stručně hlavní kroky algoritmu k -means.
2. Předpokládejme, že v první iteraci jsme náhodně vybrali body A , F a J jako počáteční středy shluků. Jaké bude rozdělení bodů do shluků po dokončení první iterace? Jaké budou středy těchto shluků? Při výpočtech používejte Euklidovskou vzdálenost.
3. Změní se nějak středy shluků v dalších iteracích algoritmu?

Náčrt řešení.

1. Algoritmus napřed náhodně zvolí počáteční středy shluků. Následně opakuje dva kroky: přiřazení každého bodu k nejbližšímu středu shluku a přepočítání polohy středu (jako průměr bodů patřících do daného shluku).
2. V prvním shluku budou body $\{A, G, B\}$, ve druhém shluku budou body $\{E, F, H\}$, ve třetím budou body $\{J, D, I, C\}$. Nové středy budou průměry těchto bodů $(-2, 0)$, $(\frac{11}{3}, 3)$ a $(\frac{17}{4}, -1)$.

3. Přiřazení do shluků už se v další iteraci nezmění, středy shluků tedy také ne.

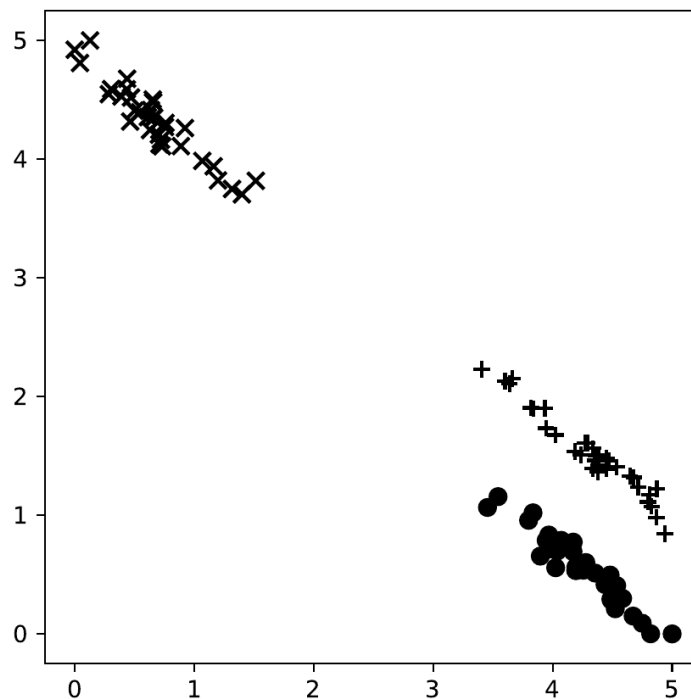
Doplňující vysvětlení (nad rámec oficiálního náčrtu): Hraniční body: $C(3, -2)$ má ke středům A, F, J vzdálenosti $\sqrt{34}, \sqrt{37}, \sqrt{18}$, patří tedy k J ; $D(4, -1)$ má vzdálenosti $\sqrt{40}, 5, \sqrt{8}$, také J ; $H(4, 2)$ má k F vzdálenost 2 a k J jen $\sqrt{5}$, patří k F . Ve druhé iteraci ke středům $(-2, 0), (\frac{11}{3}, 3), (\frac{17}{4}, -1)$ vyjde stejné přiřazení (ověřeno výpočtem), takže algoritmus končí.

19 PCA analýza

(Reálná otázka: léto 2024, č. 29 – PCA analýza)

Zadání.

Mějme data se dvěma atributy a ze třech tříd zobrazená na obrázku níže.



1. Vysvětlete, jak funguje PCA analýza. Jakým způsobem můžeme spočítat hlavní komponenty?
2. Bez počítání odhadněte, jaké budou směry dvou hlavních komponent pro data na obrázku výše. Zakreslete je do obrázku.
3. Uvažujme klasifikátor založený na metodě k nejbližších sousedů pro vhodné k a předpokládejme, že jsme v rámci předzpracování dat použili PCA analýzu pro snížení počtu dimenzí na 1. Jaká bude přesnost tohoto klasifikátoru v porovnání s přesností stejného klasifikátoru trénovaného na původních datech? Vysvětlete.

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení.)

1. PCA: data se vycentrují (odečte se průměr), spočte se *kovarianční matice* a její *vlastní vektory* – to jsou hlavní komponenty, seřazené sestupně podle vlastních čísel (každé udává rozptyl zachycený v daném směru). Redukce dimenze = projekce na prvních pár komponent.
2. První hlavní komponenta míří ve směru *největšího rozptylu* dat (podél protáhlého mraku, zhruba po antidiagonále), druhá kolmo na ni.

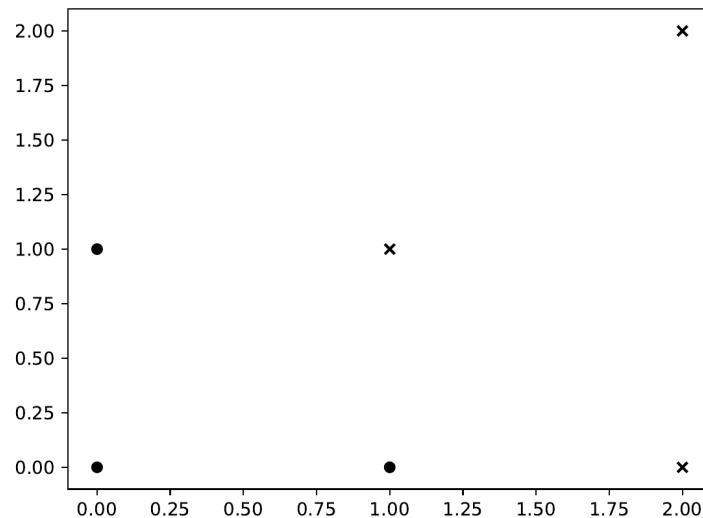
3. Přesnost se spíš *zhorší*. PCA je metoda *bez učitele* – maximalizuje rozptyl, nikoli oddělitelnost tříd. Na těchto datech leží rozdíl mezi třídami převážně ve směru *kolmém* na 1. komponentu, takže projekce na 1D třídy překryje a k -NN ztratí informaci potřebnou k jejich rozlišení.

20 Metoda podpůrných vektorů (SVM)

(Reálná otázka: léto 2024, č. 30 – Metoda podpůrných vektorů (SVM))

Zadání.

Uvažujme data se dvěma příznaky, která chceme klasifikovat do dvou tříd. Data pro třídu A jsou na pozicích (1, 1), (2, 2) a (2, 0), data pro třídu B jsou na pozicích (0, 0), (1, 0) a (0, 1). Data jsou také zobrazena na obrázku níže. Třída A je označena jako křížek (×), třída B jako kolečko (•).



1. Popište, jak fungují a jak se trénují lineární support vector machines (metoda podpůrných vektorů) v případě, že předpokládáme, že data jsou lineárně separabilní.
2. Na základě dat výše zakreslete přibližnou rozhodovací hranici lineárního SVM, která odděluje tyto dvě třídy. Nemusíte nic počítat.
3. Anž byste cokoliv počítali, vysvětlete, jak přidání nového bodu (3, 3) do třídy A ovlivní rozhodovací hranici lineárního SVM.
4. Vylepšilo by na datech uvedených výše použití RBF jádra klasifikaci? Zakreslete nějaký dataset, kde by RBF jádra pomohla.

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení, separátor ověřen řešičem SVM (sklearn).)

1. Lineární SVM hledá oddělující přímku (obecně nadrovinu) s *maximálním marginem* – největším odstupem od nejbližších bodů obou tříd. Řeší úlohu kvadratického programování $\min \frac{1}{2} \|\mathbf{w}\|^2$ za podmínek $t_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$. Hranici určují *podpůrné vektory* (body s nenulovým multiplikátorem); ty leží přesně na okraji marginu, obráceně to platit nemusí.
2. Nejbližší body třídy A leží na přímce $x_1 + x_2 = 2$ ((1, 1) a (2, 0)), nejbližší body třídy B na $x_1 + x_2 = 1$ ((1, 0) a (0, 1)). Maximálně marginová hranice je rovnoběžka uprostřed, $x_1 + x_2 = 1,5$: bod (1, 1) má od úsečky (1, 0)–(0, 1) (leží v konvexním obalu třídy B) vzdálenost $1/\sqrt{2}$, takže žádná oddělující přímka nemá větší margin než polovinu, a tato přímka ho dosahuje (kanonicky $\mathbf{w} = (2, 2)$, $b = -3$, margin $1/(2\sqrt{2}) \approx 0,354$). Podpůrné

- vektory (s nenulovým multiplikátorem) vrací řešič $(1, 1)$ ze třídy A a $(1, 0), (0, 1)$ ze třídy B; bod $(2, 0)$ leží na marginu také.
3. Bod $(3, 3)$ leží daleko od hranice na správné straně, není tedy podpurným vektorem – rozhodovací hranice se *nezmění*.
 4. Data jsou už lineárně separabilní, takže RBF jádro klasifikaci nezlepší (spíš by hrozilo přeučení). RBF pomáhá u *nelineárně* oddělitelných dat – např. jedna třída tvoří shluk uvnitř a druhá prstenec kolem něj (soustředné kružnice).

21 Metoda nejmenších čtverců

(Reálná otázka: léto 2024, č. 31 – Metoda nejmenších čtverců)

Zadání.

Mějme data s jedním číselným atributem x_1 a číselnou cílovou hodnotou y zadaná v tabulce níže. Pro tato data chceme najít lineární model metodou nejmenších čtverců.

x_1	y
1	1
2	3
3	2

1. Popište lineární model a metodu nejmenších čtverců. Jakým způsobem se touto metodou odhadují koeficienty modelu?
2. Odhadněte koeficienty modelu pro data zadaná v tabulce výše.
3. Spočítejte střední kvadratickou chybu získaného modelu.

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení, výpočet ověřen numericky.)

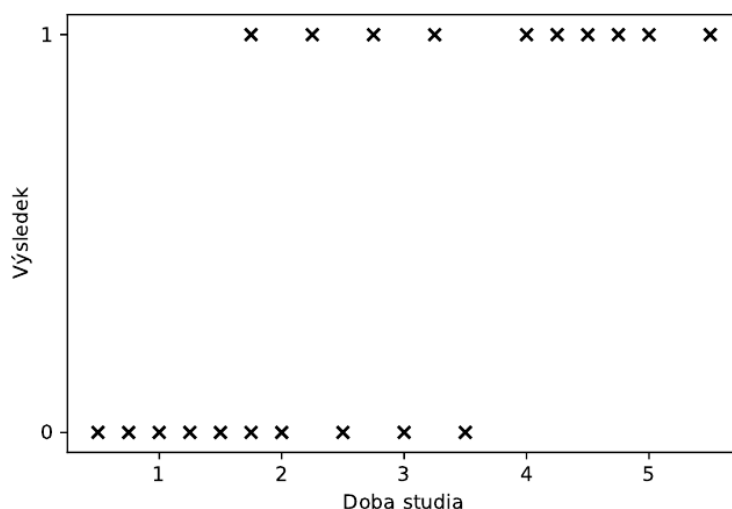
1. Lineární model $\hat{y} = \beta_0 + \beta_1 x$; metoda nejmenších čtverců volí koeficienty tak, aby *součet čtverců reziduí* $\sum_i (\hat{y}_i - y_i)^2$ byl nejmenší (řeší se normální rovnice $\beta = (X^\top X)^{-1} X^\top y$, pro jednu proměnnou stačí uzavřené vzorce).
2. S průměry $\bar{x} = 2, \bar{y} = 2$: $\beta_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = \frac{1}{2}, \beta_0 = \bar{y} - \beta_1 \bar{x} = 1$. Tedy $\hat{y} = 1 + \frac{1}{2}x$.
3. Predikce $(1, 5; 2; 2, 5)$, rezidua $(-0, 5; 1; -0, 5)$, $MSE = \frac{1}{3}(0, 25 + 1 + 0, 25) = 0, 5$.

22 Logistická regrese

(Reálná otázka: podzim 2024, č. 29 – Logistická regrese)

Zadání.

Chceme použít logistickou regresi k vytvoření modelu, který bude předpovídat pravděpodobnost úspěchu studentů u zkoušky v závislosti na době učení. Pro získání dat jsme se po zkoušce ptali studentů, jak dlouho se učili. Jejich odpovědi společně s informací, jestli zkouška byla úspěšná, jsou v obrázku níže, který zobrazuje úspěch (1) a neúspěch (0) v závislosti na době učení.



1. Popište model logistické regrese. Jak se spočítá jeho výstup? Jakým způsobem se trénují jeho parametry? Jaká se používá ztrátová funkce?
2. Do obrázku výše zakreslete přibližně výstup modelu natrénovaného na zadaných datech. Nemusíte nic počítat.
3. Jakým způsobem by se model lišil, pokud bychom klasifikovali do více než dvou tříd (např. pokud bychom chtěli předpovídat i známku)?

Řešení. (V PDF bez náčrtu řešení; vlastní vzorové řešení.)

1. Logistická regrese modeluje pravděpodobnost úspěchu $P(\text{úspěch} \mid x) = \sigma(wx + b)$ se sigmoidou $\sigma(z) = 1/(1 + e^{-z})$; predikuje úspěch, je-li výstup $\geq 0,5$. Parametry se učí maximalizací věrohodnosti, tj. minimalizací *křížové entropie* (negative log-likelihood); v SGD se používá gradient $(\sigma(wx + b) - t)x$.
2. Výstup je rostoucí *sigmoidní* křivka: u krátké doby učení blízko 0, u dlouhé blízko 1, s přechodem (hodnota 0,5) v oblasti, kde se mísí úspěchy a neúspěchy.
3. Pro více než dvě třídy (např. predikce známky) se sigmoida nahradí *softmaxem*: model má sadu vah pro každou třídu, $P(C_k \mid x) = e^{w_k^\top x} / \sum_j e^{w_j^\top x}$, a trénuje se kategoričnou křížovou entropií.

23 Evaluace binárního klasifikátoru

(Reálná otázka: podzim 2024, č. 30 – Evaluace binárního klasifikátoru)

Zadání.

Testovací množina obsahuje 1000 příkladů. Binární klasifikátor z nich 400 klasifikoval do pozitivní třídy a 600 do negativní třídy s úspěšností (accuracy) 70 % a specificitou (specificity) 80 %.

1. Napište kompletní matici konfuze (confusion matrix).
2. Vypočtěte přesnost (precision) klasifikátoru.
3. Vypočtěte úplnost (recall) klasifikátoru.
4. Je možné na základě uvedených dat vypočítat F-score? Pokud ano, jak?

Náčrt řešení.

1. $TN = 400, FN = 200, TP = 300, FP = 100.$
2. $P = TP/(TP + FP) = 0.75, R = TP/(TP + FN) = 0.6.$
3. Ano. $F1 = 2 \cdot P \cdot R/(P + R).$

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Soustava: $TP + FP = 400, TN + FN = 600,$
 $TP + TN = 700$ (accuracy) a $TN/(TN + FP) = 0,8$ (specificity). Dosazením $FP = 400 - TP$
a $TN = 700 - TP$ do poslední rovnice vyjde $TP = 300$, odtud ostatní hodnoty. F-skóre spočítat
lze: $F1 = \frac{2 \cdot 0,75 \cdot 0,6}{0,75 + 0,6} = \frac{2}{3} \approx 0,67.$

24 Rozhodovací strom

(Reálná otázka: podzim 2024, č. 31 – Rozhodovací strom)

Zadání.

Máme data se třemi atributy A, B a C a cílovou třídou Y uvedená v tabulkách níže.

1. Na trénovacích datech uvedených v tabulce popište algoritmus tvoření rozhodovacího stromu a strom postavte (plný, bez omezení na počet dat v listech a při dělení).
2. Pro atribut vybraný do kořene popište výpočet kritéria, na základě kterého jste daný atribut vybrali. (Potřebujete-li logaritmy, stačí přibližně, některé aproximace jsou v tabulce.)

x	1/5	1/3	2/5	3/5	2/3	4/5
$\approx \log_2(x)$	-2,32	-1,58	-1,32	-0,74	-0,58	-0,32

3. Dále popište některou standardní metodu prořezávání. Pokud metoda využívá validační data, použijte validační data v tabulce.

Trénovací data:

	A	B	C	Y
0	1	2	6	n
1	1	2	6	n
2	3	2	8	y
3	3	4	6	y
4	3	4	8	y
5	3	4	8	n
6	3	4	8	n
7	3	4	6	y

Validační data:

	A	B	C	Y
8	1	2	8	n
9	3	4	6	y
10	3	4	6	y
11	3	4	8	y

Náčrt řešení.

Pro výběr atributu se užívá entropie nebo gini index. Entropie pro dvouhodnotovou pravděpodobnost je definovaná

$$H(a, b) = -\frac{a}{a+b} \log_2\left(\frac{a}{a+b}\right) - \frac{b}{a+b} \log_2\left(\frac{b}{a+b}\right)$$

např.

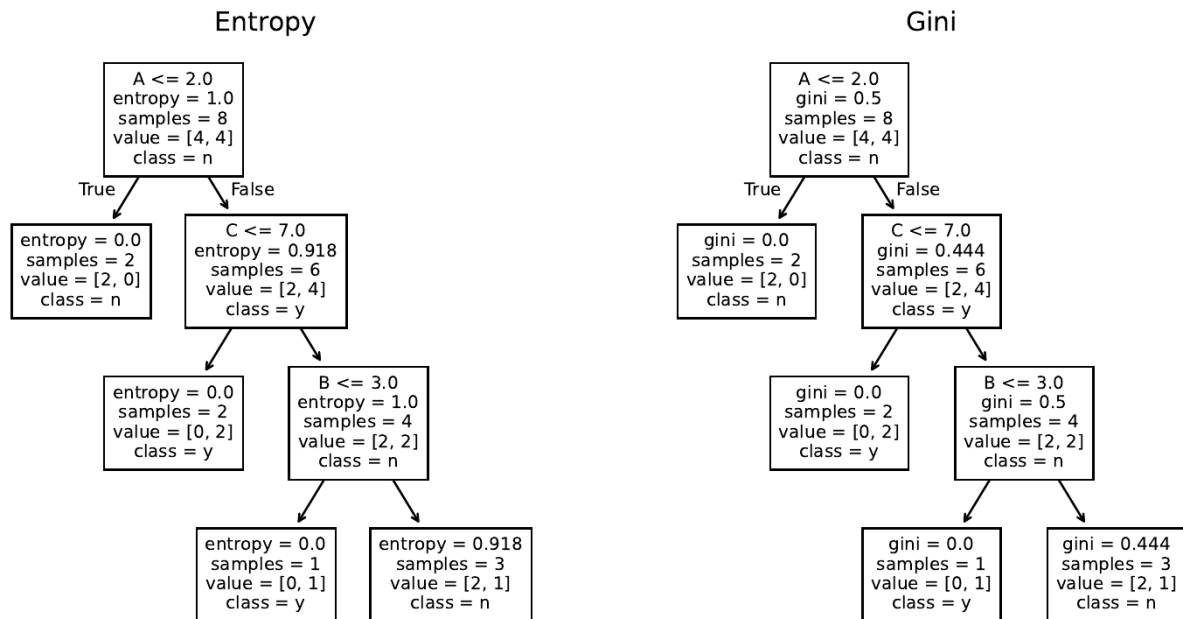
$$H(2/6, 4/6) = -\frac{1}{3} \log_2\left(\frac{1}{3}\right) - \frac{2}{3} \log_2\left(\frac{2}{3}\right) \approx \frac{-1,58}{3} - \frac{2 \cdot (-0,58)}{3} = \frac{2,74}{3} \approx 0,91.$$

Entropický zisk je rozdíl entropie před dělením a vážené entropie po dělení; entropický zisk dělení dle A v kořeni je:

$$H(4, 4) - \frac{2}{8} \cdot 0 - \frac{6}{8} H(2, 4) \approx 1 - \frac{6}{8} \cdot 0,91 \approx 0,32.$$

Postavený strom vyjde stejný na základě entropie i gini, viz obrázek. Většinou je atribut k dělení zřejmý, např. dělení podle B v kořeni by vedlo k $[2, 1]$, $[2, 3]$, což má nižší entropický zisk kvůli nenulové entropii v obou listech a více špatně klasifikovaných příkladů.

Prořezávání na základě validačních dat vyhodnotí, že je vhodné nahradit pravý uzel C listem, protože sníží validační chybu. Alternativou může být použití cost complexity pruning.

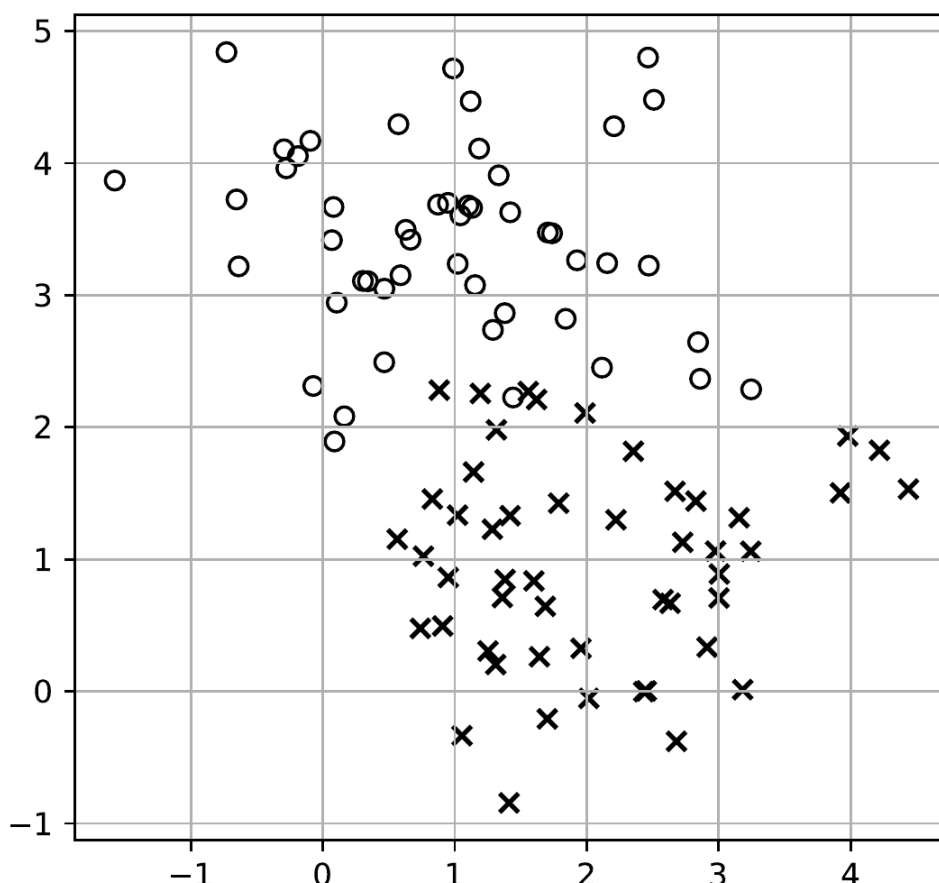


25 k -nejbližších sousedů

(Reálná otázka: jaro 2025, č. 28 – k -nejbližších sousedů)

Zadání.

1. Popište metodu k -nejbližších sousedů pro klasifikaci i pro regresi. Jak probíhá trénování a předpovídání?
2. Mějme data se dvěma atributy, která chceme klasifikovat do dvou tříd ($\circ$, $\times$) znázorněná na obrázku níže. Uvažujme $k = 3$ (tedy používáme tři nejbližší sousedy). Jak bude klasifikován nový vstup $(3, 2)$? Změnil by se výsledek klasifikace pokud změníme hodnotu k ?
3. Popište, jakým způsobem můžeme nastavit vhodnou hodnotu k pro naše data.



Náčrt řešení.

1. Při trénování se pouze uloží trénovací data. Při klasifikaci/regresi se vyhledá k nejbližších sousedů. Pro klasifikaci se použije nejčastější třída těchto sousedů, pro regresi se použije průměr hodnot nejbližších sousedů.
2. Bod bude klasifikován jako $\circ$, neboť většina ze 3 nejbližších sousedů patří do této třídy. Pokud zvýšíme k , mnoho bodů s třídou $\times$ se dostane mezi nejbližší sousedy a tím se změní klasifikace (např. pro $k = 7$).
3. Vhodnou hodnotu k je možné nastavit například pomocí křížové validace.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Podle souřadnic odečtených z obrázku jsou dva nejbližší sousedé dotazu $(3, 2)$ kroužky (vzdálenosti $\approx 0,38$ a $0,40$), třetím je křížek ($\approx 0,58$; druhý křížek je prakticky stejně daleko). Pro $k = 3$ tedy hlasování 2:1 dává $\circ$. Pro $k = 5$ je pátý a šestý soused (kroužek a křížek) ve stejné vzdálenosti $\approx 0,67$ v rámci přesnosti obrázku, výsledek je tedy na hraně. Pro $k = 7$ je poměr jednoznačně 3:4 ve prospěch $\times$. Při volbě k křížovou validací se pro dvě třídy volí liché k , aby nevznikaly remízy.

26 Logistická regrese pro binární klasifikaci

(Reálná otázka: jaro 2025, č. 29 – Logistická regrese pro binární klasifikaci)

Zadání.

1. Popište, jakým způsobem můžeme použít logistickou regresi pro binární klasifikaci. Máme k dispozici dataset se vstupními příznaky $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ a s cílovými hodnotami $\mathbf{t} = (t_1, \dots, t_N)$.

Jak vypadá predikce modelu? Jak budeme postupovat, abychom mohli nezávisle určit očekávanou chybu modelu na neviděných datech?

2. Jak je definována ztrátová funkce? Odvoďte její derivaci a ukažte, jakým způsobem lze použít gradientní metodu (stochastic gradient descent) pro trénování.
3. Jakým způsobem zabráníte přeučení při stochastic gradient descentu?

Náčrt řešení.

1. Model: $p(C_1) = \sigma(\mathbf{x}^T \mathbf{w})$, predikce: if $(y > 0.5)$ 1 else 0.

Postup pro odhad chyby: Rozdělíme si data na trénovací, validační a testovací část. Validační data použijeme pro ladění hyperparametrů, testovací pro reportování výsledku na neviděných datech.

2. Chybová funkce: negative log-likelihood $E(\mathbf{w}) = \frac{1}{N} \sum_i -\log(p(C_{t_i} | \mathbf{x}_i; \mathbf{w}))$, dále rozepíšeme podle Bernoulliho distribuce a můžeme derivovat podle $\mathbf{w}$.

Derivace vyjde $\nabla_{\mathbf{w}} E(\mathbf{w}) = (\sigma(\mathbf{x}^T \mathbf{w}) - t)\mathbf{x}$.

SGD: Inicializujeme váhy (buď náhodně nebo nulami), iterujeme přes dataset, v každém kroku odečteme $\alpha \cdot \nabla E(\mathbf{w})$.

3. Přeučení můžeme bránit: L^2 regularizací, pomocí early stopping (sledujeme validační chybu a přestaneme s trénováním, když začne růst), vhodným learning rate schedule.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Odvození derivace pro jeden příklad, $z = \mathbf{x}^T \mathbf{w}$, $y = \sigma(z)$: Bernoulliho model dává $p(t | \mathbf{x}; \mathbf{w}) = y^t(1 - y)^{1-t}$, tedy $E = -t \log \sigma(z) - (1 - t) \log(1 - \sigma(z))$. Derivace sigmoidy je $\sigma'(z) = \sigma(z)(1 - \sigma(z))$, takže

$$\frac{\partial E}{\partial z} = -t \frac{\sigma'(z)}{\sigma(z)} + (1 - t) \frac{\sigma'(z)}{1 - \sigma(z)} = -t(1 - \sigma(z)) + (1 - t)\sigma(z) = \sigma(z) - t,$$

a protože $\nabla_{\mathbf{w}} z = \mathbf{x}$, je $\nabla_{\mathbf{w}} E = (\sigma(\mathbf{x}^T \mathbf{w}) - t)\mathbf{x}$. Krok SGD na minibatchi $\mathbb{B}$: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{1}{|\mathbb{B}|} \sum_{i \in \mathbb{B}} (\sigma(\mathbf{x}_i^T \mathbf{w}) - t_i) \mathbf{x}_i$. S L^2 regularizací $\frac{\lambda}{2} \|\mathbf{w}\|^2$ přibude ke gradientu člen $\lambda \mathbf{w}$.

27 Lineární regrese

(Reálná otázka: léto 2025, č. 30 – Lineární regrese)

Zadání.

1. Pro trénovací dataset X_{train} a cílové hodnoty t_{train} zformulujte model lineární regrese a chybovou funkci, kterou v lineární regresi minimalizujeme.
2. Ukažte, jakým způsobem lze explicitně získat optimální řešení na trénovacích datech.
3. Zformulujte optimalizaci pomocí stochastic gradient descent.
4. Okomentujte výhody a nevýhody obou přístupů, především z hlediska možnosti kontroly chyby na případných validačních datech.

Náčrt řešení.

Definice modelu a podrobnější odvození jsou ve slidech k prvním dvěma přednáškám z NPFL129.

Je třeba spočítat derivaci chyby, tu pak položit rovnou nule a vyřešit buď s využitím inverzní matice nebo s pomocí SVD dekompozice. Nevýhoda je, že invertování matice je pomalé a matice je velká. Optimální řešení na trénovací množině nemusí být to, co chceme, protože může dojít k přeučení. SGD bude méně náročné na paměť, navíc můžeme kontrolovat loss na validační množině a udělat early stopping a vyhnout se přeučení.

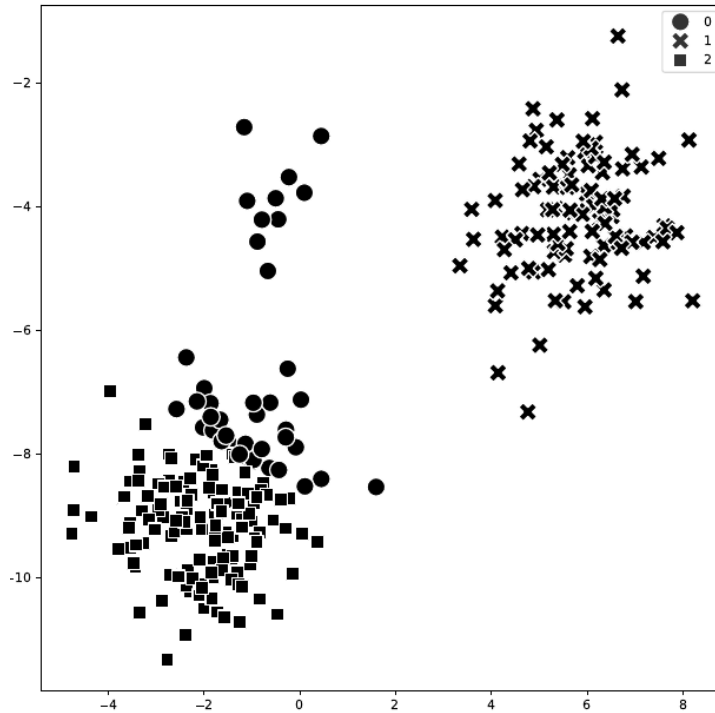
Doplňující vysvětlení (nad rámec oficiálního náčrtu): Model $y(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$ (bias jako váha u přidaného konstantního příznaku 1), chyba $E(\mathbf{w}) = \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{t}\|^2$ (případně MSE, liší se jen kladnou konstantou). Gradient $\nabla_{\mathbf{w}} E = \mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{t})$ položený roven nule dává normální rovnice $\mathbf{X}^T \mathbf{X} \mathbf{w} = \mathbf{X}^T \mathbf{t}$, tedy $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$, je-li $\mathbf{X}^T \mathbf{X}$ regulární (jinak pseudoinverze přes SVD). Minimum je globální, protože E je konvexní (Hessián $\mathbf{X}^T \mathbf{X}$ je pozitivně semidefinitní). SGD: pro náhodnou minibatch $\mathbb{B}$ opakuj $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{1}{|\mathbb{B}|} \sum_{i \in \mathbb{B}} (\mathbf{x}_i^T \mathbf{w} - t_i) \mathbf{x}_i$. Explicitní řešení stojí $\mathcal{O}(ND^2 + D^3)$ a vrací jen konečné optimum; SGD dává průběžně posloupnost modelů, takže lze po každé epoše měřit validační chybu a zastavit v jejím minimu (early stopping).

28 Shlukování

(Reálná otázka: léto 2025, č. 31 – Shlukování (*k-means*))

Zadání.

1. Popište stručně algoritmus *k-means*. Jakým způsobem se volí počáteční polohy středů shluků (centroidů)? Jak se jejich polohy v průběhu algoritmu upravují? Kdy algoritmus končí?
2. Jaké kritérium algoritmus *k-means* optimalizuje?
3. Algoritmus *k-means* pro $k = 3$ jsme spustili na data se třemi shluky z obrázku níže. Jednotlivé značky ($\times$, $\bullet$, $\blacksquare$) odlišují nalezené shluky. Proč algoritmus na těchto datech nenašel optimální shluky podle kritéria v předchozím bodě? Jak se dá zvýšit šance, že je algoritmus najde?



Náčrt řešení.

1. Počáteční centroidy se mohou volit náhodně z dat. V průběhu algoritmu se každý bod z dat přiřadí k nejbližšímu centroidu, ten se potom přepočítá jako střed bodů k němu přiřazených. Tyto kroky se opakují dokud se mění přiřazení bodů ke shlukům, nebo po daný počet iterací.

2. Minimalizuje se

$$\sum_{i=1}^N \sum_{k=1}^K z_{i,k} \|x_i - \mu_k\|^2,$$

kde N je počet bodů, K je počet středů a $z_{i,k}$ vyjadřuje, že bod x_i patří do shluku se středem μ_k ($z_{i,k} = 1$ pokud tam patří, jinak 0).

3. Prostřední shluk, který byl přiřazen k části levého, je mnohem menší než ostatní dva shluky. Šance na lepší výsledek se dá zlepšit opakováním náhodné inicializace počátečních centroidů a výběrem nejlepšího výsledku podle funkce v předchozím bodě.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Na obrázku jsou tři přirozené shluky: velký dole vlevo, malý nahoře uprostřed a jeden vpravo. Nalezené řešení rozdělilo velký levý shluk mezi dva středy (■ a ●) a malý horní shluk připojilo ke středu ●. Takové rozdělení je *lokálním* minimem kritéria: žádný bod nemá blíž k jinému středu a průměry se nemění, takže se algoritmus zastaví, ačkoli rozdělení podle přirozených shluků má menší součet čtverců. k -means konverguje jen k lokálnímu minimu a výsledek závisí na počáteční volbě středů. Kromě opakovaných běhů s výběrem nejmenšího kritéria pomáhá inicializace **k-means++**, která rozprostře počáteční středy.

29 EM algoritmus

(Reálná otázka: léto 2025, č. 32 – EM algoritmus)

Zadání.

1. Popište EM algoritmus.
2. Uvažujme pravděpodobnostní model definovaný bayesovskou sítí s veličinou *Cluster* s hodnotami $\{1, 2\}$ jakožto rodičem veličin *Color* a *Holes*, které nabývají hodnot $\{blue, green\}$ resp. $\{yes, no\}$. Veličiny *Color* a *Holes* nejsou spojeny hranou.
Určete parametry modelu a inicializujte je hodnotami z množiny $\{0.4, 0.6\}$, aby tvořily korektní model a podmíněné pravděpodobnosti pro podmínku $Cluster = 1$ nebyly identické s těmi pro $Cluster = 2$.
3. Zvolte jeden z parametrů z předchozího bodu a vypočtěte pro něj jednu aktualizaci dle EM algoritmu. Uvažujte následující data, kde „?“ značí chybějící hodnotu.

Color	Holes	Cluster
blue	no	?
blue	yes	?
blue	no	?
green	yes	?
green	yes	?

Při výpočtu můžete použít následující aproximace: $0.6^0 * 0.4^3 \approx 0.064$, $0.6^1 * 0.4^2 \approx 0.096$, $0.6^2 * 0.4^1 \approx 0.144$, $0.6^3 * 0.4^0 \approx 0.216$.

Náčrt řešení.

1. EM algoritmus umožňuje odhadnout parametry modelu s nepozorovanými proměnnými. Inicializuje model a iteruje kroky E a M, dokud parametry modelu nezkonvergují.

E Spočte očekávané hodnoty chybějících dat.

M Spočte maximálně věrohodné odhady parametrů na datech s doplněnými chybějícími hodnotami.

2. Parametry modelu (s jejich inicializací pro následující bod):

parametr	inicializace	značení
$P(Cluster = 1)$	0.6	θ
$P(Color = blue \mid Cluster = 1)$	0.6	θ_{B1}
$P(Color = blue \mid Cluster = 2)$	0.4	θ_{B2}
$P(Holes = yes \mid Cluster = 1)$	0.6	θ_{H1}
$P(Holes = yes \mid Cluster = 2)$	0.4	θ_{H2}

3. Nejjednodušší aktualizace je pro parametr θ . Přes datové řádky $[color_j, holes_j]$ spočteme $\sum_j P(Cluster = 1 \mid color_j, holes_j)$ a vydělíme počtem příkladů.

Color	Holes	$P(Cluster = 1 \mid color_j, holes_j)$
blue	no	$\frac{\theta \cdot \theta_{B1} \cdot (1 - \theta_{H1})}{\theta \cdot \theta_{B1} \cdot (1 - \theta_{H1}) + (1 - \theta) \cdot \theta_{B2} \cdot (1 - \theta_{H2})} = \frac{0.6 * 0.6 * 0.4}{0.6 * 0.6 * 0.4 + 0.4 * 0.4 * 0.6} = \frac{0.6 * 0.6 * 0.4}{0.4 * 0.6} = 0.6$ ²
blue	yes	$\frac{\theta \cdot \theta_{B1} \cdot \theta_{H1}}{\theta \cdot \theta_{B1} \cdot \theta_{H1} + (1 - \theta) \cdot \theta_{B2} \cdot \theta_{H2}} = \frac{0.6 * 0.6 * 0.6}{0.6 * 0.6 * 0.6 + 0.4 * 0.4 * 0.4} = \frac{0.216}{0.216 + 0.064} = 0.77$
blue	no	jako výše, 0.6
green	yes	$\frac{\theta \cdot (1 - \theta_{B1}) \cdot \theta_{H1}}{\theta \cdot (1 - \theta_{B1}) \cdot \theta_{H1} + (1 - \theta) \cdot (1 - \theta_{B2}) \cdot \theta_{H2}} = \frac{0.6 * 0.4 * 0.6}{0.6 * 0.4 * 0.6 + 0.4 * 0.6 * 0.4} = 0.6$
green	yes	jako výše, 0.6

Pro námi zvolené počáteční parametry je nová hodnota θ rovna $\frac{4 \cdot 0.6 + 0.77}{5} = \frac{3.17}{5} = 0.634$.

30 Evaluace binárního klasifikátoru

(Reálná otázka: podzim 2025, č. 28 – Evaluace binárního klasifikátoru)

Zadání.

Představte si, že vaším úkolem je vybrat nástroj pro analýzu počítačových logů a vyhledávání anomálií v nich (ty mohou být způsobené například selháním hardware, kybernetickým útokem, apod.). Máte možnost si vybrat ze dvou nástrojů. Z dostupných marketingových materiálů jste se o prvním nástroji dozvěděli, že má úspěšnost (accuracy) 99,5 % a umí správně odhalit 90 % anomálií. U druhého nástroje se uvádí, že odhalí 81 % anomálií a jen 10 % nahlášených anomálií jsou falešně pozitivní.

Předpokládejte, že oba nástroje byly vyhodnoceny na stejných datech obsahujících 10 000 instancí, z nichž pouze 1 % jsou anomálie.

1. Spočítejte chybovou matici (matice konfuze, confusion matrix) obou nástrojů a tyto metriky pro binární klasifikaci: úspěšnost (accuracy), přesnost (precision), úplnost (recall), F1-skóre.
2. Popište výhody a nevýhody jednotlivých nástrojů při praktickém nasazení. Který z nich byste použili? Zdůvodněte. (Zde není nutné jediná správná odpověď.)

Náčrt řešení.

O datech víme, že obsahují celkem 9 900 negativních instancí (nejdou anomálie) a 100 pozitivních instancí.

1. První nástroj správně detekuje 90 ze 100 anomálií. Tedy $TP = 90$, z toho plyne, že $FN = 10$. Tento nástroj také správně určí 9 950 z 10 000 instancí. Tedy musí správně určit $9 950 - TP = 9 860$ negativních instancí. Tedy $TN = 9 860$ a $FP = 40$. Recall je 0.9 (ze zadání), precision je $TP/(TP + FP) = 90/130$ (cca 0.69). F1-skóre je 0.78, accuracy (ze zadání) 0.995.

Druhý nástroj správně detekuje 80 ze 100 anomálií.³ Tedy $TP = 81$, a $FN = 19$. Ze zadání jen 10 % nahlášených jsou FP, tedy precision je $0.9 = TP/(TP + FP)$ a tedy $FN = 81/9 = 9$.

²V originálním PDF je ve jmenovateli druhého členu napsáno $(1 - \theta_{C2})$; podle parametrizace i číselného dosazení jde o $(1 - \theta_{H2})$; zde uvedeno korektně.

³V originálním PDF je zde uvedeno „ $TP = 81$, a $FN = 19$ “, dále „ $FN = 81/9 = 9$ “ a accuracy 0.997; tyto hodnoty jsou v PDF nekonzistentní (recall 0.81 při 100 pozitivních dává $TP = 81$, ne 80). Také je v PDF u precision uveden vzorec $TP/(TP + FN)$, správně má být $TP/(TP + FP)$. Zde přepsáno věrně dle PDF až na opravu vzorce precision; případné aritmetické nekonzistence v originálu ponechány.

Zbytek instancí (9 891) je TN. Recall je tedy 0.81 (ze zadání), precision je 0.9 (ze zadání), F1-skóre je 0.85, accuracy 0.997.

2. Druhý nástroj je lepší ve všech metrikách (kromě recall) než nástroj první. Jeho výhodou může být to, že hlásí mnohem méně falešně pozitivních výsledků, nezaměstnává tedy správce systému zbytečnými hlášeními. Na druhou stranu, pokud jsou anomálie kritické, může být lepší použít první nástroj, který jich detekuje více za cenu toho, že okolo 30 % detekcí jsou falešně pozitivní.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Konzistentní hodnoty pro druhý nástroj: recall 0,81 při 100 anomáliích dává TP = 81, FN = 19; precision 0,9 dává FP = 81/9 = 9 (v náčrtu omylem označeno FN) a TN = 9900 – 9 = 9891. Accuracy je $(81 + 9891)/10\,000 = 0,9972$ a $F_1 = \frac{2 \cdot 0,9 \cdot 0,81}{0,9 + 0,81} \approx 0,85$. U prvního nástroje je $F_1 = \frac{2 \cdot (90/130) \cdot 0,9}{90/130 + 0,9} \approx 0,78$ a podíl falešných poplachů mezi hlášeními $40/130 \approx 31\%$.

31 Predikce pomocí tabulárních dat

(Reálná otázka: podzim 2025, č. 29 – Predikce pomocí tabulárních dat)

Zadání.

Máte k dispozici následující zemědělský dataset pro predikci výnosu plodin a na základě tohoto datasetu chcete natrénovat model strojového učení.

pH půdy	Srážky (mm)	Průměr. teplota (°C)	Typ hnojiva	Odrůda osiva	Velikost farmy (ha)	Zavlažování	Nadm. výška (m)	Výnos (t/ha)
6,8	450	22	Organické	Hybrid-A	48,6	Kapkové	150	4,2
7,2	380	25	Chemické	Standard	34,4	Postřikovač	200	3,8
6,5	520	18	Organické	Odolná	121,4	Dešťová	850	3,2
7,0	320	28	Bio-hnojivo	Hybrid-B	60,7	Kapkové	75	5,1
6,9	680	20	Chemické	Vysoký výnos	25,9	Záplavové	300	6,8

Na základě ukázkového datasetu odpovězte na následující otázky:

1. Jaký typ úlohy strojového učení to je?
2. Navrhněte vhodný model strojového učení pro tuto úlohu a stručně vysvětlete, proč by byl vhodný.
3. Pokud to navržený model vyžaduje, popište kroky předpracování, které byste na tento dataset aplikovali před trénováním modelu. Jak byste zacházeli s různými typy příznaků (features)?
4. Jaké evaluační metriky byste použili k posouzení výkonnosti modelu?

Náčrt řešení.

Jedná se o *regresi* a *učení s učitelem*, protože cílová proměnná (Výnos) je spojitá číselná hodnota a máme k dispozici trénovací data s cílovými hodnotami.

Nejllepší volba modelu: *Random Forests* nebo *Gradient Boosting (XGBoost/LightGBM)*, vhodné pro tabulární data. *Lineární regrese* nebo *MLP* by také mohly fungovat s pečlivým předpracováním příznaků.

Kroky předpracování:

- *Kategorické příznaky:* Aplikovat one-hot enkódování pro typ hnojiva, odrůdu osiva a metodu zavlažování.

- *Číselné příznaky*: Zkontrolovat odlehle hodnoty a aplikovat standardizaci/normalizaci (např. StandardScaler), pokud používáme algoritmy citlivé na škálu. V datasetu: pH půdy, srážky, teplota, velikost farmy, nadmořská výška.
- (volitelně) *Chybějící hodnoty*: Dodat chybějící hodnoty (průměr/medián pro číselné, modus pro kategorické).
- (volitelně) *Feature engineering*: Zvážit vytvoření kombinovaných příznaků (např. srážky vs. teplota) nebo polynomiálních příznaků, pokud doménové znalosti naznačují nelineární vztahy.

Evaluační metriky: Mean Absolute Error, (Root) Mean Square Error.

32 Lineární regrese, L2 regularizace

(Reálná otázka: podzim 2025, č. 30 – Lineární regrese, L2 regularizace)

Zadání.

Mějme data s nezávislou proměnnou x a predikovanou proměnnou y zadaná v tabulce níže. Budeme pro ně hledat lineární model bez regularizace a s L2 regularizací.

x	y
-1	1
0	3
1	2

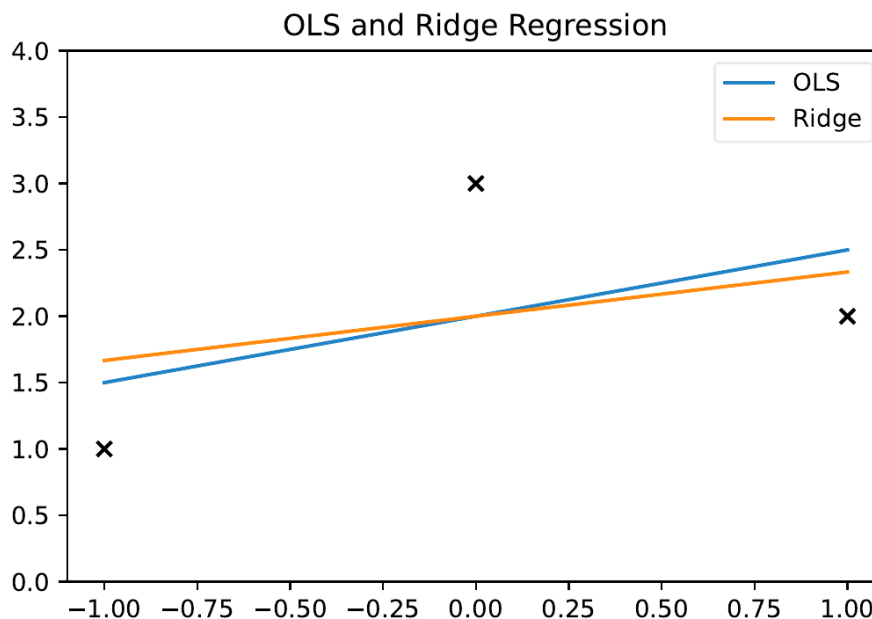
1. Popište lineární model a metodu nejmenších čtverců. Jakým způsobem se odhadují koeficienty modelu?
2. Uvažujte model bez regularizace (OLS). Vypočítejte koeficienty modelu pro data zadaná v tabulce a do grafu zanešte predikované hodnoty pro $x \in \langle -1, 1 \rangle$.
3. Popište optimalizovanou funkci s L2 regularizací. Jak se změní koeficienty při odhadu modelu s L2 regularizací? Stačí určit směr změny koeficientů u x oproti OLS a přibližně zaneš do grafu predikované hodnoty pro $x \in \langle -1, 1 \rangle$.

Náčrt řešení.

1. Lineární model s jednou nezávislou proměnnou odhaduje dva parametry β_0, β_1 modelu $f(x) = \beta_0 + \beta_1 \cdot x$, minimalizujeme funkci $\sum_{i=1,2,3} (f(x_i) - y_i)^2$. Pro odhad parametrů k tabulce přidáme sloupec jedniček a řešíme $\beta = (X^T X)^{-1} X^T y$.
2. Díky jediné nezávislé proměnné a centrovaným datům lze i snadněji, zde uvádíme obecné řešení:

$$\begin{aligned} \beta &= (X^T X)^{-1} X^T y \\ &= \left(\begin{bmatrix} 1, 1, 1 \\ -1, 0, 1 \end{bmatrix} \cdot \begin{bmatrix} 1, -1 \\ 1, 0 \\ 1, 1 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1, 1, 1 \\ -1, 0, 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} = \left(\begin{bmatrix} 3, 0 \\ 0, 2 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 6 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{6}{3} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 2 \\ \frac{1}{2} \end{bmatrix}. \end{aligned}$$

3. S L2 regularizací přidáváme do cílové funkce součet kvadrátů β_j bez β_0 , tj. minimalizujeme $\sum_{i=1,2,3} (f(x_i) - y_i)^2 + \alpha \beta_1^2$, $\alpha > 0$. Koeficient β_1 bude menší než v předchozím případě, ale zachová si znaménko. Intercept β_0 není penalizován, v tomto případě se nezmění, obecně se změnit může. (V grafu je R2 regularizace nazvaná Ridge.)



Doplňující vysvětlení (nad rámec oficiálního náčrtu): Pro tato data lze L^2 řešení spočítat přesně: penalizace $\alpha\beta_1^2$ přičte α jen k prvku matice $X^T X$ u β_1 , takže soustava je $\begin{pmatrix} 3 & 0 \\ 0 & 2+\alpha \end{pmatrix} \beta = \begin{pmatrix} 6 \\ 1 \end{pmatrix}$ a vyjde $\beta_0 = 2$, $\beta_1 = \frac{1}{2+\alpha} < \frac{1}{2}$. Intercept se nemění, protože data jsou centrovaná ($\bar{x} = 0$, matice $X^T X$ je diagonální). Přímka se tedy zploští kolem bodu $(0, 2)$. Označení „R2“ v poslední větě náčrtu je zřejmě překlep za $L2$.

33 Shlukování pomocí K -means

(Reálná otázka: jaro 2026, č. 28 – Shlukování pomocí K -means)

Zadání.

1. Popište algoritmus K -means. K čemu se používá? Popište podrobně jeho kroky a jak je možné algoritmus inicializovat.
2. Jakou ztrátovou funkci algoritmus optimalizuje?
3. Konverguje algoritmus? Zformulujte argument nebo důkaz, proč tomu tak je. Jaké to má praktické důsledky?

Náčrt řešení.

- K -means je algoritmus učení bez učitele (neřízené učení) používaný pro shlukování dat do K skupin podle (geometrické) podobnosti jejich rysů.
- Algoritmus:
 1. Inicializace K centroidů (náhodně nebo pomocí metod jako `kmeans++`).
 2. Algoritmus: Opakuj, dokud nedojde ke konvergenci (žádná změna přiřazení nebo centroidů):
 - (a) Přiřaď každý datový bod k nejbližšímu centroidu.
 - (b) Aktualizuj centroidy výpočtem průměru všech bodů v každém shluku.

- **Kmeans++** inicializace: začni s náhodným centroidem, poté iterativně vybírej nové centroidy s pravděpodobností úměrnou čtverci vzdálenosti od nejbližšího již existujícího centroidu.
- Ztrátová funkce, kterou algoritmus optimalizuje, je součet čtverců vzdáleností mezi datovými body a jejich přiřazenými centroidy.
- Algoritmus konverguje, protože každá iterace snižuje hodnotu ztrátové funkce, která je omezená zdola nulou. Zaručuje pouze konvergenci k lokálnímu minimu, nikoli nutně ke globálnímu minimu.
- Praktické důsledky: Volba počátečních centroidů může ovlivnit konečný výsledek shlukování, a proto může být nutné provést více běhů s různými inicializacemi, aby se našlo lepší řešení.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Samotné klesání zdola omezené funkce ještě nezaručuje, že se algoritmus po konečně mnoha krocích zastaví. Úplný argument: krok přiřazení (každý bod k nejbližšímu centroidu) ani krok přepočtu (centroid jako průměr, který minimalizuje součet čtverců ve shluku) ztrátu J nezvýší. Hodnota J po přepočtu je určena jen přiřazením bodů do shluků a různých přiřazení je konečně mnoho (nejvýše K^N). Při každé skutečné změně přiřazení J ostře klesne, takže se žádné přiřazení nemůže zopakovat, a algoritmus se proto po konečně mnoha iteracích zastaví, v lokálním minimu J .

34 Kombinace modelů

(Reálná otázka: jaro 2026, č. 29 – Kombinace modelů)

Zadání.

1. Stručně popište základní myšlenky technik bagging a boosting. Uveďte příklad konkrétních modelů, které tyto techniky používají.
2. Uvažujme tři různé modely M_1 , M_2 a M_3 pro binární klasifikaci. Přesnost (accuracy) modelu M_1 je 0,70, přesnost modelu M_2 je 0,75 a přesnost modelu M_3 je 0,80. Z těchto modelů vytvoříme ensemble tak, že modely necháme hlasovat (vzorek se předloží všem modelům a jako finální odpověď se použije ta, která je častější). Za jakých podmínek bude přesnost výsledného ensemble lepší, než přesnost nejlepšího modelu? Jaká je nejlepší dosažitelná přesnost ensemble? Za jakých podmínek bude naopak přesnost ensemble horší než přesnost nejlepšího modelu? *Nápověda: Uvažujte předpovědi každého z modelů na jednotlivých vzorcích dat.*

Náčrt řešení.

1. Bagging trénuje modely na různých náhodných podmnožinách trénovací množiny (bootstrap). Výstupy modelů se následně agregují (např. hlasováním) pro finální predikci ensemble. Při boostingu se trénuje posloupnost modelů tak, že instance špatně předpovězené předchozím modelem mají větší váhu. Příkladem baggingu může být random forest, příkladem boostingu AdaBoost nebo XGBoost.
2. Pokud každý z modelů chybně předpovídá jiná data (množiny chybně předpovězených instancí jsou navzájem disjunktní), výsledný ensemble bude mít accuracy 100 %. Pokud by naopak množina instancí chybně klasifikovaná modelem M_2 byla podmnožinou množiny chybně klasifikované modelem M_1 , bude accuracy přinejlepším 0,75. Pokud by navíc chybně klasifikované vzorky modelu M_3 pokryly zbytek chybně klasifikovaných modelem M_1 , bude accuracy přinejlepším 0,7.

Doplňující vysvětlení (nad rámec oficiálního náčrtu): Označme E_i množinu vzorků, na nichž chybí M_i ($|E_1| = 30\%$, $|E_2| = 25\%$, $|E_3| = 20\%$). Ensemble se mýlí právě na vzorcích, které leží aspoň ve dvou množinách E_i . Je lepší než nejlepší model, právě když takových vzorků je méně než 20% , tj. když se chybové množiny překrývají málo (chyby modelů jsou málo korelované). Při nezávislých chybách by vyšla accuracy $1 - (0,14 + 0,015) = 0,845$. Horší než nejlepší model je ensemble při velkém překryvu; nejhorší možná accuracy je $1 - \frac{0,30+0,25+0,20}{2} = 0,625$ (každý chybný vzorek ensmbly spotřebuje aspoň dvě chyby modelů), např. při $|E_1 \cap E_2| = 17,5\%$, $|E_1 \cap E_3| = 12,5\%$, $|E_2 \cap E_3| = 7,5\%$ a prázdném trojném průniku.

35 Rozhodovací strom

(Reálná otázka: jaro 2026, č. 30 – Rozhodovací strom)

Zadání.

- a) Popište algoritmus tvoření rozhodovacího stromu. Postavte strom z trénovacích dat uvedených dále s tím, že do kořene zvolte atribut D . Všechny další volby atributů zdůvodněte, pro pravý podstrom D spočítejte hodnoty kritéria pro všechny atributy.

Cílový atribut je Y . Sloupec „id“ nesmí být použit k učení, slouží jen pro snazší identifikaci řádku.

Potřebujete-li logaritmy, stačí přibližně. Například $\log_2(1/3) \approx -1,58$, $\log_2(2/3) \approx -0,58$, $\log_2(3/4) \approx -0,42$.

- b) Vysvětlete pojem prořezávání stromu a uveďte neprořezaný strom do maximální hloubky a prořezaný strom. Uveďte kritérium pro prořezávání.

Trénovací data:					
id	A	B	C	D	Y
0	3	4	8	6	0
1	1	2	8	6	0
2	1	2	8	3	1
3	3	2	8	6	1
4	1	4	6	6	1
5	3	2	8	6	0
6	3	4	6	3	0
7	3	4	8	3	1

Náčrt řešení.

- a) Pro výběr atributu se užívá entropie nebo gini index. Entropie pro dvouhodnotovou pravděpodobnost je definovaná

$$H(a, b) = -\frac{a}{a+b} \log_2\left(\frac{a}{a+b}\right) - \frac{b}{a+b} \log_2\left(\frac{b}{a+b}\right)$$

např.

$$H(1/3, 2/3) = -\frac{1}{3} \log_2\left(\frac{1}{3}\right) - \frac{2}{3} \log_2\left(\frac{2}{3}\right) \approx \frac{-1,58}{3} - \frac{2 \cdot (-0,58)}{3} = \frac{2,74}{3} \approx 0,91.$$

Entropický zisk je rozdíl entropie před dělením a vážené entropie po dělení; vážená entropie po dělení dle A , B v pravém podstromu je:

$$\frac{3}{5} \cdot H(1, 2) + \frac{2}{5} H(1, 1) \approx \frac{3}{5} \cdot 0,91 - \frac{2}{5} \cdot 1 \approx 0,95.$$

vážená entropie po dělení dle C v pravém podstromu je:

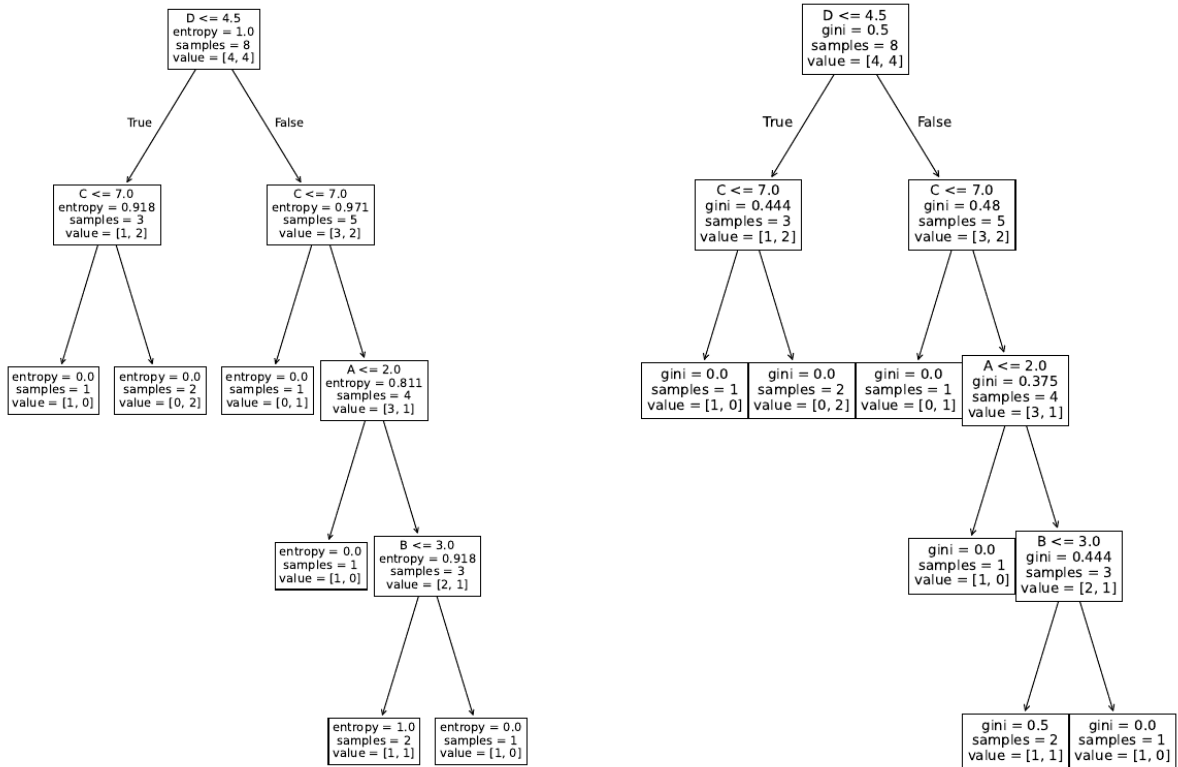
$$\frac{1}{5} \cdot H(1, 0) + \frac{4}{5} H(3, 1) \approx \frac{4}{5} \cdot 0,811 \approx 0,65,$$

proto má dělení podle C vyšší entropický zisk. Postavený strom vyjde stejný na základě entropie i gini, viz obrázek.

- b) Prořezáváním se snaží odstranit přečunění, kdy chyba na trénovacích datech je sice malá, ale na testovacích větší než u jednoduššího stromu.

V našem případě ve stromě na obrázku nahradíme uzel $A \leq 2.0$ listem, protože chybu klasifikace nezhoršíme a strom bude mít menší počet listů.

Je mnoho různých metod prořezávání, např. cost complexity pruning definuje $R(T_t)$ jakožto celkovou chybu klasifikace (pod)stromu T_t s kořenem v uzlu t , $R(t)$ když t bude listem, $|T|$ počet listů stromu, a iterativně vybírá uzel t s nejmenším $\alpha_{eff} = \frac{R(t) - R(T_t)}{|T_t| - 1}$ dokud je α_{eff} menší než zvolený práh.



Doplňující vysvětlení (nad rámec oficiálního náčrtu): Ve výpočtu vážené entropie pro A a B má být součet: $\frac{3}{5} \cdot 0,91 + \frac{2}{5} \cdot 1 \approx 0,95$ (v náčrtu je omylem znaménko minus). Entropie pravého podstromu $D=6$ s poměrem (3, 2) je $B(\frac{2}{5}) \approx 0,97$, takže zisk atributu C je $\approx 0,32$ a u A i B jen $\approx 0,02$. V levém podstromu $D=3$ (id 2, 6, 7) oddělí C čistě id 6 (list 0) od id 2, 7 (list 1). V uzlu $C=8$ pravého podstromu (id 0, 1, 3, 5) dávají A i B stejný zisk; obrázek volí A . Nahrazení uzlu $A \leq 2$ listem 0 trénovací chybu nezmění (v obou případech zůstane chybně klasifikován jeden z příkladů id 3 a 5, které mají shodné atributy a různé Y), strom se ale zmenší o dva listy.