Gradivo za državno tekmovanje iz umetne inteligence (DTUI)

Programski odbor izbirnega tekmovanja iz umetne inteligence

1 Priprava podatkov

1.1 Naloge z rešitvami

1. **Standardizacija**

   Dana je značilka z vrednostmi

   $$[2, 4, 6, 8, 10]. \tag{1}$$

   Izračunajte njeno povprečje in standardni odklon ter nato vse vrednosti standardizirajte.

   Rešitev.

   Povprečje je

   $$\mu = \frac{2 + 4 + 6 + 8 + 10}{5} = \frac{30}{5} = 6. \tag{2}$$

   Varianca je

   $$\sigma^2 = \frac{(2-6)^2 + (4-6)^2 + (6-6)^2 + (8-6)^2 + (10-6)^2}{5} = \frac{40}{5} = 8. \tag{3}$$

   Standardni odklon je zato

   $$\sigma = \sqrt{8} \approx 2{,}83. \tag{4}$$

   Vrednosti standardiziramo po enačbi

   $$\tilde{x}_i = \frac{x_i - \mu}{\sigma}. \tag{5}$$

   Dobimo približno

   $$[-1{,}41, -0{,}71, 0, 0{,}71, 1{,}41]. \tag{6}$$

2. **Enično kodiranje**

   Kategorična značilka *velikost* ima tri možne vrednosti:

$$\{\text{majhna}, \text{srednja}, \text{velika}\}. \tag{7}$$

Koliko stolpcev nastane pri eničnem kodiranju (angl. *one-hot encoding*)? Zapišite kodiranje vrednosti *srednja*.

Rešitev.

Pri eničnem kodiranju nastanejo trije stolpci, po eden za vsako kategorijo.

Če uporabimo vrstni red

$$[\text{majhna}, \text{srednja}, \text{velika}], \tag{8}$$

je vrednost *srednja* kodirana kot

$$[0, 1, 0]. \tag{9}$$

3. **Delitev podatkov**

Podatkovna množica vsebuje 5000 primerov. Razdeliti jo želimo v razmerju

$$70\% : 15\% : 15\% \tag{10}$$

na učno, validacijsko in testno množico.

Koliko primerov vsebuje posamezna množica? Kakšna je vloga vsake izmed njih?

Rešitev.

Učna množica vsebuje

$$5000 \cdot 0{,}70 = 3500 \tag{11}$$

primerov.

Validacijska množica vsebuje

$$5000 \cdot 0{,}15 = 750 \tag{12}$$

primerov.

Testna množica prav tako vsebuje

$$5000 \cdot 0{,}15 = 750 \tag{13}$$

primerov.

Torej:

- učna množica: 3500 primerov,
- validacijska množica: 750 primerov,

- testna množica: 750 primerov.

Učna množica se uporablja za učenje parametrov modela. Validacijska množica se uporablja med razvojem za izbiro modela in hiperparametrov. Testna množica se uporabi na koncu za oceno uspešnosti izbranega modela na novih podatkih.

4. **Manjkajoče vrednosti**

   V podatkovni množici manjka 10 % vrednosti v stolpcu *dohodek*.

   Predlagajte tri različne načine obravnave manjkajočih vrednosti. Za vsak način navedite:

   (a) primer, ko bi bil primeren;

   (b) primer, ko bi lahko povzročil težave.

   Rešitev.

   Uporabimo lahko na primer naslednje tri pristope:

   (a) **Nadomeščanje s povprečjem ali mediano.**

   Mediana je primerna predvsem takrat, ko vrednosti manjkajo približno naključno in želimo ohraniti vse primere. Poleg tega je manj občutljiva na ekstremne vrednosti kot povprečje.

   Težava nastane, če vrednosti ne manjkajo naključno. Če na primer predvsem osebe z zelo visokimi dohodki ne želijo navesti dohodka, lahko nadomeščanje z mediano sistematično podceni njihove dohodke.

   (b) **Odstranitev vrstic z manjkajočimi vrednostmi.**

   Ta pristop je lahko primeren, kadar imamo veliko podatkov in izguba 10 % primerov ni problematična.

   Težava nastane, če manjkajoče vrednosti niso naključne. Z odstranitvijo takih vrstic lahko dobimo pristransko podatkovno množico.

   (c) **Napoved manjkajoče vrednosti.**

   Dohodek lahko ocenimo iz drugih značilk, na primer starosti, izobrazbe ali poklica. Ta pristop je primeren, kadar druge značilke dobro napovedujejo dohodek.

   Težava je, da ima tudi model za napoved dohodka napake, ki se nato prenesejo v nadaljnji postopek.

5. **Skaliranje značilk**

   Zakaj bi pretvorba cene hiš v logaritemsko skalo lahko pomagala pri uspešnosti napovedi?

   Rešitev.

   Cene hiš imajo pogosto desno asimetrično porazdelitev: veliko je hiš z zmernimi cenami, malo pa je izjemno dragih nepremičnin.

   Logaritemska transformacija

$$x' = \log(x) \tag{14}$$

   stisne zgornji del razpona. Porazdelitev tako postane bolj simetrična, ekstremno visoke cene pa imajo manjši vpliv na funkcijo izgube modela.

To lahko modelu olajša učenje in izboljša napovedovanje, predvsem kadar bi nekaj zelo visokih cen sicer nesorazmerno močno vplivalo na rezultat/funkcijo izgube modela.

6. **Gradnja značilk**

   Na voljo imamo podatke o nakupih uporabnikov s stolpci:

   `timestamp`, `user_id`, `purchase_amount`, `product_category`.

   Želimo napovedati, ali bo uporabnik v naslednjem tednu opravil nakup.

   Predlagajte vsaj tri nove značilke, ki bi jih lahko ustvarili iz podanih podatkov. Pri vsaki na kratko pojasnite, zakaj bi lahko bila uporabna.

   Rešitev.

   Ustvarimo lahko na primer:

   - **Število dni od zadnjega nakupa.** (angl. *recency*) Ta značilka meri, kako nedavno je uporabnik opravil nakup.
   - **Povprečni znesek nakupov v zadnjih 30 dneh.** (angl. *monetary trend*) Značilka opisuje uporabnikovo nedavno potrošnjo in morebitne spremembe v nakupovalnem vedenju.
   - **Število nakupov po kategorijah izdelkov.** (angl. *preference signal*) Tako dobimo informacijo o uporabnikovih preferencah.
   - **Dan v tednu in ura nakupa.** (angl. *temporal behavior*) Iz teh podatkov lahko zaznamo časovne vzorce nakupovanja.
   - **Razmerje med trenutnim oziroma zadnjim nakupom in uporabnikovim zgodovinskim povprečjem.** (angl. *anomaly signal*) Veliko odstopanje lahko kaže na spremembo nakupovalnega vedenja.

7. **Skaliranje značilk**

   Podatkovna množica vsebuje značilke:

   - starost: 0–100,
   - letni dohodek: 0–1, 000, 000,
   - število otrok: 0–10.

   Odgovorite:

   (a) Kaj se lahko zgodi, če na takih podatkih brez skaliranja uporabimo algoritem, ki temelji na razdaljah?

   (b) Zakaj bi lahko značilka *dohodek* prevladala nad ostalima?

   (c) Ali pričakujemo enako težavo pri odločitvenem drevesu? Utemeljite.

   Rešitev.

   (a) Pri algoritmih, ki temeljijo na razdaljah, kot je k-NN, bi bile razdalje med primeri skoraj v celoti določene z dohodkom. Najbližji sosedi bi bili zato izbrani predvsem glede na podobnost v dohodku, medtem ko bi imeli starost in število otrok zelo majhen vpliv.

   (b) Razlog je v zelo različnih merilih značilk. Dohodek ima razpon od 0 do 1 000 000, starost od 0 do 100, število otrok pa od 0 do 10. Razlike v dohodku so zato numerično mnogo večje od razlik pri drugih dveh značilkah in lahko prevladajo pri izračunu razdalje.

(c) Pri odločitvenem drevesu praviloma ne pričakujemo te težave. Odločitveno drevo obravnava vsako značilko posebej in išče pragove, po katerih razdeli primere. Skaliranje posamezne značilke zato ne spremeni vrstnega reda njenih vrednosti in posledično ne vpliva na možne razdelitve drevesa.

Odločitvena drevesa so zato neobčutljiva na merilo značilk (angl. *scale-invariant*), medtem ko so metode, ki temeljijo na razdaljah, kot so k-NN, SVM in k-means, na merilo značilk občutljive.

8. **Iskanje ciljnega uhajanja**

Želimo napovedati, ali bo uporabnik odpovedal naročnino. Na voljo imamo naslednje značilke:

(a) število dni od zadnje prijave;

(b) skupno število nakupov;

(c) razlog za odpoved naročnine (tekstovno polje);

(d) povprečno trajanje seje;

(e) število zahtevkov za podporo.

Katera značilka lahko predstavlja **ciljno uhajanje** (angl. *target leakage*)?

Pojasnite, zakaj. Kako bi v praksi preverili, ali je značilka ob času napovedovanja sploh na voljo?

Rešitev.

Problematična značilka je **razlog za odpoved naročnine**.

Ta informacija je praviloma znana šele po tem, ko je uporabnik naročnino že odpovedal, zato neposredno razkriva informacijo o ciljni spremenljivki. Če želimo odpoved napovedati vnaprej, ta značilka v trenutku napovedi še ni na voljo.

Model, ki bi uporabljal to značilko, bi lahko med učenjem in evalvacijo dosegal zelo visoko uspešnost, vendar bi bil tak rezultat zavajajoč. V produkciji bi bilo namreč polje z razlogom za odpoved pri aktivnih uporabnikih, za katere želimo napovedati prihodnjo odpoved, prazno oziroma še ne bi obstajalo.

V praksi moramo zato preveriti, kdaj se posamezna značilka ustvari oziroma zapiše v podatkovno zbirko. Pri podatkovnem procesu preverimo, ali je bila vrednost značilke dejansko na voljo že v trenutku, ko bi model moral podati napoved, in ne šele po nastanku ciljnega dogodka.

Preostale značilke so lahko znane že pred odpovedjo naročnine in zato ne predstavljajo ciljnega uhajanja, če so izračunane samo iz podatkov, ki so bili na voljo do trenutka napovedi.

9. **Močna povezanost še ni uhajanje**

Pri napovedovanju cene hiše imamo značilke:

- površina hiše,
- število spalnic,
- leto izgradnje,
- zemljepisna širina in dolžina.

Nekdo predlaga, da značilke odstranimo, ker so močno povezane s ceno in zato predstavljajo uhajanje podatkov.

Razložite, zakaj ta sklep ni pravilen.

Pojasnite razliko med legitimno napovedno značilko in dejanskim uhajanjem podatkov.

Rešitev.

Močna povezanost s ciljno spremenljivko sama po sebi ni uhajanje podatkov. Pravzaprav želimo, da dobre napovedne značilke vsebujejo informacije, ki pomagajo napovedovati cilj.

Površina, število spalnic, leto izgradnje in lokacija opisujejo hišo in so znani že pred prodajo. Zato so legitimne napovedne značilke.

Pri uhajanju podatkov značilka vsebuje informacijo iz prihodnosti ali informacijo, ki neposredno izhaja iz ciljne spremenljivke in v trenutku napovedi ne bi bila na voljo.

Ključna razlika torej ni v moči povezave s ciljem, temveč v tem, *od kod informacija prihaja in kdaj je na voljo.*

10. **Vrste uhajanja podatkov**

Za vsak primer določite, ali gre za:

- ciljno uhajanje (angl. *target leakage*),
- uhajanje med učenjem in testiranjem (angl. *train-test leakage*),
- ali pa sploh ne gre za uhajanje.

Odgovor utemeljite.

(a) Napovedujemo rezultat izpita in uporabimo število ur učenja, izmerjeno pred izpitom.

(b) Napovedujemo rezultat izpita in uporabimo zadovoljstvo učenca z oceno, izmerjeno po objavi rezultatov.

(c) Manjkajoče vrednosti nadomestimo s povprečjem, izračunanim na celotni podatkovni množici, nato pa podatke razdelimo na učno in testno množico.

(d) Napovedujemo ponovni sprejem pacienta v bolnišnico in uporabimo dolžino njegovega začetnega bivanja, ki je znana ob odpustu.

Rešitev.

(a) **Ni uhajanja.** Število ur učenja je izmerjeno pred izpitom in je legitimna napovedna značilka.

(b) **Ciljno uhajanje.** Zadovoljstvo z oceno je izmerjeno po objavi rezultata in zato vsebuje informacijo o izidu, ki ga želimo napovedati.

(c) **Uhajanje med učenjem in testiranjem.** Pri izračunu povprečja smo uporabili tudi testne podatke. Informacija iz testne množice je tako vplivala na pripravo učnih podatkov.

(d) **Ni uhajanja.** Če napoved naredimo ob odpustu, je dolžina začetnega bivanja takrat že znana, ponovni sprejem pa se zgodi kasneje.

11. **Pristranskost vzorčenja**

Za vsak primer ugotovite, zakaj podatkovna množica morda ni reprezentativna. Kjer je mogoče, poimenujte vrsto pristranskosti vzorčenja.

(a) Sistem za samodejno vožnjo je naučen skoraj izključno na posnetkih vožnje iz Kalifornije.

(b) Model za analizo mnenj uporabnikov se uči samo iz javno objavljenih ocen izdelkov.

(c) V raziskavi o zdravilu 30 % udeležencev pred zaključkom preneha sodelovati.

(d) Model za napoved uspešnosti zagonskih podjetij je naučen samo na podatkih podjetij, ki so uspešno preživela prvih pet let.

Pri vsakem primeru navedite tudi, katera skupina ali vrsta podatkov je premalo zastopana.

Rešitev.

(a) Gre za **geografsko pristranskost** (angl. *geographic bias*). Premalo so zastopane razmere zunaj Kalifornije, na primer vožnja v snegu in močnem dežju, drugačna urbana okolja ter države z drugačno prometno infrastrukturo.

(b) Gre za **izbirno pristranskost** (angl. *seelction bias*). Javne ocene pogosteje pišejo ljudje z izrazito pozitivnimi ali negativnimi izkušnjami. Uporabniki z zmernimi oziroma nevtralnimi izkušnjami so zato lahko premalo zastopani.

(c) Gre za **pristranskost zaradi osipa** (angl. *survivorship/attrition bias*). Udeleženci, ki odstopijo, se lahko sistematično razlikujejo od tistih, ki ostanejo. Če na primer pogosteje odstopijo bolniki s stranskimi učinki, bodo ti v končnih podatkih premalo zastopani.

(d) Gre za **pristranskost preživelih** (angl. *survivorship bias*). Podatkovna množica izključuje podjetja, ki so propadla v prvih petih letih. Neuspešna zagonska podjetja so zato povsem premalo zastopana.

12. **Premik porazdelitve**

Model za napovedovanje povpraševanja po taksijih v New Yorku je bil naučen na podatkih iz leta 2019 in uporabljen marca 2020.

(a) Zakaj lahko pričakujemo, da porazdelitev podatkov ob uporabi ni več enaka učni porazdelitvi?

(b) Ali se je lahko spremenila porazdelitev vhodov $P(X)$, ciljnih vrednosti $P(Y)$ ali povezava $P(Y \mid X)$?

(c) Kako bi lahko takšno spremembo zaznavali po uvedbi modela?

Rešitev.

(a) Marca 2020 je pandemija močno spremenila način gibanja ljudi. Vzorci voženj, delovni čas, potovanja na letališča in obisk poslovnih središč zato niso bili več podobni tistim iz leta 2019.

Spremenila se je torej lahko porazdelitev vhodnih podatkov, kar imenujemo **premik porazdelitve vhodov** (angl. *covariate shift*). Hkrati se je lahko spremenila tudi povezava med značilkami in povpraševanjem, na primer povezava med časom dneva in številom voženj. Takšno spremembo imenujemo **sprememba koncepta** (angl. *concept drift*).

(b) Označimo z P_{train} porazdelitev podatkov, na katerih je bil model naučen, in s P_{deploy} porazdelitev podatkov ob dejanski uporabi modela.

Spremeni se lahko porazdelitev vhodnih značilk:

$$P_{\text{train}}(X) \neq P_{\text{deploy}}(X). \tag{15}$$

To predstavlja **premik porazdelitve vhodov** (angl. *covariate shift*). Na primer, spremenijo se lahko običajni časi voženj, lokacije prevzemov ali pogostost poti na letališča.

Spremeni se lahko tudi porazdelitev ciljnih vrednosti:

$$P_{\text{train}}(Y) \neq P_{\text{deploy}}(Y), \tag{16}$$

saj se lahko spremeni skupno povpraševanje po taksijih.

Če se spremeni tudi povezava med vhodnimi značilkami in ciljno vrednostjo, lahko velja

$$P_{\text{train}}(Y \mid X) \neq P_{\text{deploy}}(Y \mid X). \tag{17}$$

To predstavlja **spremembo koncepta** (angl. *concept drift*). Na primer, čas dneva, ki je leta 2019 običajno pomenil veliko povpraševanje, marca 2020 ni več nujno pomenil enako velikega povpraševanja.

(c) Po uvedbi modela lahko spremljamo porazdelitve novih vhodnih značilk in jih primerjamo s porazdelitvami v učnih podatkih. Tako lahko zaznamo premik porazdelitve vhodov.

Ko postanejo znane dejanske ciljne vrednosti, lahko spremljamo tudi napake modela in preverjamo, ali njegova uspešnost sčasoma pada. Poleg tega lahko nastavimo opozorila, če začne zaupanje modela sistematično upadati.

13. **Veljavnost testne množice**

Model za prepoznavanje kožnega raka je naučen na slikah iz dermatoloških klinik v severni Evropi. Testno množico sestavlja naključnih 15 % slik iz istih klinik in model na njej doseže visoko uspešnost.

Model nato uporabimo v klinikah v podsaharski Afriki.

(a) Zakaj se lahko uspešnost modela spremeni?

(b) Ali je prvotni rezultat na testni množici zato napačen?

(c) Kaj rezultat dejansko pove in česa ne?

(d) Kako bi oblikovali testno množico, če vnaprej vemo, kje bo model uporabljen?

Rešitev.

(a) Uspešnost se lahko spremeni, ker se podatki iz novega okolja razlikujejo od učnih podatkov. Razlikujejo se lahko odtenki kože, razširjenost bolezni, osvetlitev, uporabljene kamere in drugi pogoji zajema slik.

(b) Prvotni rezultat zato ni nujno napačen. Pravilno meri uspešnost na podatkih, ki prihajajo iz podobne porazdelitve kot učni podatki.

(c) Rezultat pove, kako dobro model deluje na novih slikah iz podobnih severnoevropskih klinik. Ne pove pa zanesljivo, kako dobro bo deloval na drugih populacijah in v drugačnih pogojih.

(d) Če vnaprej vemo, da bo model uporabljen v različnih geografskih okoljih, mora testna množica vključevati podatke iz teh ciljnih okolij. Vsebovati mora različne klinike, odtenke kože, geografska območja ter različne pogoje in naprave za zajem slik.

14. **Časovna delitev podatkov**

Imamo podatke o prodaji izdelka od leta 2020 do leta 2025 in želimo napovedovati prodajo v prihodnosti.

Nekdo predlaga, da vse primere naključno premešamo in 20 % podatkov uporabimo za testiranje.

Zakaj je lahko takšna delitev problematična? Predlagajte ustreznejši način delitve podatkov.

Rešitev.

Pri naključni delitvi lahko pride do tega, da se model uči na novejših podatkih in testira na starejših. Tako bi lahko na primer uporabili podatke iz leta 2025 za napovedovanje primerov iz leta 2022.

To ne ustreza dejanski uporabi modela, kjer imamo ob napovedovanju prihodnosti na voljo samo podatke iz preteklosti.

Primer primernejše delitve je

$$\begin{aligned} \text{učna množica:} &\quad 2020 - -2023, \\ \text{validacijska množica:} &\quad 2024, \\ \text{testna množica:} &\quad 2025. \end{aligned} \tag{18}$$

Na ta način model vedno uporabljamo v isti časovni smeri kot pri dejanski uporabi: učimo ga na preteklosti in ocenjujemo na prihodnosti.

15. **Ciklično kodiranje**

Mesec v letu je predstavljen s številom

$$x \in \{1, 2, \ldots, 12\}. \tag{19}$$

Zakaj je neposredna numerična predstavitev lahko problematična?

Izračunajte ciklično predstavitev za januar ($x = 1$) in december ($x = 12$):

$$x_{\sin} = \sin\left(2\pi \frac{x}{12}\right), \qquad x_{\cos} = \cos\left(2\pi \frac{x}{12}\right). \tag{20}$$

Na podlagi dobljenih vrednosti pojasnite, kako ciklično kodiranje bolje izrazi bližino januarja in decembra.

Rešitev.

Pri neposrednem numeričnem kodiranju sta januar in december navidezno zelo oddaljena:

$$|12 - 1| = 11, \tag{21}$$

čeprav sta v koledarju sosednja meseca.

Za januar ($x = 1$) dobimo

$$x_{\sin} = \sin\left(\frac{2\pi}{12}\right) = \sin\left(\frac{\pi}{6}\right) = \frac{1}{2} = 0{,}5, \tag{22}$$

in

$$x_{\cos} = \cos\left(\frac{\pi}{6}\right) = \frac{\sqrt{3}}{2} \approx 0{,}866. \tag{23}$$

Januar je torej predstavljen s točko

$$(0{,}5, 0{,}866). \tag{24}$$

Za december ($x = 12$) dobimo

$$x_{\sin} = \sin(2\pi) = 0, \tag{25}$$

in

$$x_{\cos} = \cos(2\pi) = 1. \tag{26}$$

December je torej predstavljen s točko

$$(0, 1). \tag{27}$$

Evklidska razdalja med cikličnima predstavitvama januarja in decembra je

$$\sqrt{(0{,}5 - 0)^2 + (0{,}866 - 1)^2} \approx 0{,}518. \tag{28}$$

Namesto velike numerične razlike 11 sta tako meseca predstavljena z bližnjima točkama na enotski krožnici. Ciklično kodiranje zato pravilno zajame dejstvo, da decembru neposredno sledi januar.

16. **Celoten postopek priprave podatkov**

Prejeli ste podatkovno množico, v kateri so:

- manjkajoče numerične vrednosti,
- kategorične značilke,
- numerične značilke na zelo različnih skalah,
- časovni žig,
- podvojene vrstice.

Model boste na koncu ocenjevali na ločeni testni množici.

Opišite smiseln vrstni red priprave podatkov. Pri vsakem postopku, ki iz podatkov izračuna kakšno statistiko ali drugo nastavitev, posebej navedite, na kateri množici ga smemo prilagoditi.

Rešitev.

Smiseln postopek je naslednji:

(a) **Osnovno čiščenje in odstranjevanje podvojenih vrstic.**

Preverimo podatke in odstranimo dejanske podvojene zapise. Pri tem moramo paziti, da ne odstranimo dveh legitimnih primerov, ki imata slučajno enake vrednosti.

(b) **Delitev podatkov.**

Podatke razdelimo na učno, validacijsko in testno množico, preden iz njih izračunamo statistike, ki se bodo uporabljale pri predobdelavi.

Testno množico nato pustimo nedotaknjeno do končne evalvacije.

(c) **Obravnava manjkajočih vrednosti.**

Če na primer manjkajoče numerične vrednosti nadomeščamo z mediano, mediano izračunamo samo na učni množici:

$$m_{\text{train}} = \text{median}(X_{\text{train}}). \tag{29}$$

Isto vrednost nato uporabimo pri validacijski in testni množici.

(d) **Kodiranje kategoričnih značilk.**

Kodirnik prilagodimo na učni množici. Na njej določimo kategorije in nato z istim kodirnikom transformiramo validacijsko in testno množico.

Pri tem moramo predvideti tudi možnost kategorij, ki jih med učenjem nismo videli.

(e) **Obdelava časovnega žiga.**

Iz časovnega žiga lahko ustvarimo značilke, kot so ura, dan v tednu, mesec ali čas od zadnjega dogodka.

Pri tem moramo paziti, da uporabljamo samo informacije, ki bi bile znane v trenutku napovedi.

(f) **Skaliranje numeričnih značilk.**

Parametre skaliranja izračunamo samo na učni množici. Pri standardizaciji na primer določimo

$$\mu_{\text{train}} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{30}$$

in

$$\sigma_{\text{train}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{\text{train}})^2}. \tag{31}$$

Nato učno, validacijsko in testno množico transformiramo z istima vrednostma:

$$z = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}}. \tag{32}$$

(g) **Učenje in izbira modela.**

Model učimo na učni množici. Validacijsko množico uporabljamo za izbiro hiperparametrov, značilk in modela.

(h) **Končna evalvacija.**

Ko so model, hiperparametri in celoten postopek predobdelave dokončno določeni, model ocenimo na testni množici.

Testne množice med razvojem ne uporabljamo za izbiro modela ali spreminjanje postopka predobdelave.

Ključno pravilo je:

Vsak postopek, ki se mora iz podatkov nekaj naučiti, prilagodimo samo na učni množici.

To vključuje na primer izračun povprečja, mediane in standardnega odklona, določanje kategorij ter druge parametre predobdelave. Tako preprečimo, da bi informacije iz validacijske ali testne množice vplivale na učenje modela.

2 Nadzorovano učenje

2.1 Naloge z rešitvami

1. **Linearna regresija**

   Model linearne regresije napoveduje ceno hiše po enačbi

   $$h(x) = 150\,000 + 5000x, \tag{33}$$

   kjer je x površina hiše v kvadratnih metrih.

   Kakšno ceno model napove za hišo s površino 160 kvadratnih metrov? Kako interpretiramo oba koeficienta modela?

   Rešitev.

   V enačbo vstavimo $x = 160$:

   $$h(160) = 150\,000 + 5000 \cdot 160 = 150\,000 + 800\,000 = 950\,000. \tag{34}$$

   Model torej napove ceno

   $$\boxed{950\,000}. \tag{35}$$

   Koeficient 150 000 je prestreznik oziroma osnovna napoved modela pri $x = 0$. V tem primeru bi pomenil napovedano ceno hiše s površino $0\,\mathrm{m}^2$, zato njegova praktična interpretacija ni nujno smiselna.

   Koeficient 5000 je naklon premice. Pomeni, da model za vsak dodatni kvadratni meter površine napovedano ceno poveča za 5000 denarnih enot.

2. **Logistična regresija**

   Model logistične regresije vrne

   $$h_\theta(x) = 0{,}73. \tag{36}$$

   Kateri razred model napove pri pragu 0,5? Kaj predstavlja vrednost 0,73?

   Rešitev.

   Ker velja

   $$0{,}73 > 0{,}5, \tag{37}$$

   model napove pozitivni razred oziroma razred 1.

   Vrednost 0,73 interpretiramo kot ocenjeno verjetnost modela, da primer pripada pozitivnemu razredu:

   $$P(y = 1 \mid x) \approx 0{,}73. \tag{38}$$

   Model torej ocenjuje približno 73 % verjetnost, da primer pripada razredu 1.

3. **Pridobljena informacija**

Pri gradnji odločitvenega drevesa razdelitev po značilki A ustvari dve veji, ki primere razdeli v dve podmnožici:

$$(4 \text{ pozitivni}, 0 \text{ negativnih}) \qquad \text{in} \qquad (1 \text{ pozitiven}, 5 \text{ negativnih}). \tag{39}$$

Začetna entropija je enaka 1,0.

Izračunajte **informacijski dobiček** (angl. *information gain*) te razdelitve.

Rešitev.

Prva veja vsebuje samo pozitivne primere,

$$p_+ = \frac{4}{4} = 1 \tag{40}$$

zato je njena entropija

$$H_1 = 0 \tag{41}$$

Druga veja vsebuje en pozitiven in pet negativnih primerov. Verjetnosti so

$$p_+ = \frac{1}{6}, \qquad p_- = \frac{5}{6}. \tag{42}$$

Njena entropija je

$$H_2 = -\frac{1}{6}\log_2\left(\frac{1}{6}\right) - \frac{5}{6}\log_2\left(\frac{5}{6}\right) \approx 0{,}65. \tag{43}$$

Skupaj imamo 10 primerov. Utežena entropija po razdelitvi je

$$H_{\text{po}} = \frac{4}{10}\cdot 0 + \frac{6}{10}\cdot 0{,}65 = 0{,}39. \tag{44}$$

Informacijski dobiček je zato

$$IG = H_{\text{pred}} - H_{\text{po}} = 1{,}0 - 0{,}39 = \boxed{0{,}61}. \tag{45}$$

4. **SVM in rob**

Pojasnite, zakaj SVM imenujemo **klasifikator z velikim robom** (angl. *large margin classifier*).

Kako parameter C vpliva na velikost roba in dovoljevanje napačnih razvrstitev?

Rešitev.

SVM išče ločitveno hiperravnino, ki maksimizira najmanjšo razdaljo med mejo odločanja in najbližjimi učnimi primeri. Ta razdalja se imenuje **rob** (angl. *margin*).

Zato SVM ne išče zgolj meje, ki pravilno loči učne primere, temveč med možnimi mejami išče takšno z največjim robom.

Parameter C določa kompromis med velikostjo roba in kaznovanjem napačnih razvrstitev.

- **Velik C:** napake so močno kaznovane. Model poskuša pravilno razvrstiti skoraj vse učne primere, zato je rob praviloma ožji. To lahko poveča nevarnost prekomernega prileganja.
- **Majhen C:** posamezne napake oziroma kršitve roba so manj kaznovane. Model lahko izbere širši rob in dopušča več napačno razvrščenih učnih primerov. Če je C premajhen, lahko pride do prešibkega prileganja.

5. **Regularizacija**

   Primerjajte regularizaciji L1 in L2.

   Kako vplivata na uteži modela? V katerih primerih bi dali prednost eni oziroma drugi?

   Rešitev.

   Regularizacija L1 doda funkciji izgube kazen

   $$\lambda \sum_j |w_j|. \tag{46}$$

   L1 lahko nekatere uteži zmanjša natančno na 0. Zato hkrati izvaja tudi implicitno izbiro značilk. Primerna je predvsem takrat, ko pričakujemo, da je veliko značilk nepomembnih in želimo redek model.

   Regularizacija L2 uporablja kazen

   $$\lambda \sum_j w_j^2. \tag{47}$$

   L2 uteži zmanjšuje proti 0, vendar jih praviloma ne postavi natančno na 0. Primerna je, kadar pričakujemo, da je večina značilk vsaj nekoliko pomembna in želimo predvsem omejiti velikost uteži.

   Na kratko:

   - L1: redkejši model in izbira značilk;
   - L2: bolj enakomerno zmanjševanje vseh uteži.

6. **IAIO-slog: algoritem, ki ga še niste videli**

   Sodelavec predlaga **uteženi klasifikator sosedov** (angl. *Weighted Neighbor Classifier*, WNC).

   Za testni primer x algoritem:

   (a) izračuna razdaljo d_i do vsakega učnega primera x_i;

   (b) vsakemu učnemu primeru dodeli utež

   $$w_i = \exp\left(-\frac{d_i^2}{\sigma^2}\right); \tag{48}$$

   (c) napove

   $$\hat{y} = \operatorname{sign}\left(\sum_i w_i y_i\right), \qquad y_i \in \{-1, +1\}. \tag{49}$$

   Odgovorite:

(a) Kaj se zgodi, ko $\sigma \to 0$? Kaj se zgodi, ko $\sigma \to \infty$?

(b) Kako je ta algoritem povezan s k-NN in jedrskimi metodami (angl. *kernel methods*)?

(c) Kakšne oblike je njegova meja odločanja? Ali je nujno linearna?

(d) Na kakšne načine lahko algoritem odpove?

(e) Predlagajte spremembo, s katero bi bolje obravnavali neuravnotežene razrede (angl. *class imbalance*).

Rešitev.

(a) Ko $\sigma \to 0$, uteži zelo hitro padajo z razdaljo. Pomemben postane praktično samo najbližji učni primer. WNC se zato približa klasifikatorju 1-NN.

Ko $\sigma \to \infty$, velja približno

$$w_i \approx 1 \tag{50}$$

za vse učne primere. Napoved tako postane glasovanje vseh učnih primerov in praviloma zmaga večinski razred.

(b) WNC je gladka, utežena različica k-NN. Namesto da bi izbrali samo k najbližjih sosedov, upoštevamo vse učne primere, vendar bližjim pripišemo večjo težo.

Funkcija

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{\sigma^2}\right) \tag{51}$$

je Gaussovo oziroma RBF-jedro. WNC lahko zato interpretiramo tudi kot jedrski klasifikator.

(c) Meja odločanja nastane tam, kjer je uteženi glas obeh razredov izenačen:

$$\sum_i w_i y_i = 0. \tag{52}$$

Ta meja na splošno ni linearna. Lahko se ukrivlja okoli skupin učnih primerov.

(d) Algoritem lahko odpove na več načinov:

- premajhen σ povzroči prekomerno prileganje;
- prevelik σ povzroči prešibko prileganje;
- pri vsaki napovedi je treba izračunati razdaljo do vseh učnih primerov, zato je napovedovanje počasno;
- pri neuravnoteženih razredih lahko večinski razred prevlada v uteženi vsoti.

(e) Prispevek vsakega razreda lahko normaliziramo glede na njegovo velikost:

$$\hat{y} = \operatorname{sign}\left(\frac{1}{n_+}\sum_{i:y_i=+1} w_i - \frac{1}{n_-}\sum_{i:y_i=-1} w_i\right), \tag{53}$$

kjer sta n_+ in n_- števili primerov pozitivnega in negativnega razreda.

Tako vsak razred prispeva primerljivejšo skupno težo ne glede na število učnih primerov.

7. **Problem XOR**

Obravnavajte točke

$$(0,0), \quad (0,1), \quad (1,0), \quad (1,1) \tag{54}$$

z oznakami

$$0, \quad 1, \quad 1, \quad 0. \tag{55}$$

(a) Dokažite, da problema ne more rešiti noben linearni klasifikator.

(b) Pokažite, da z dodajanjem značilke

$$x_3 = x_1 x_2 \tag{56}$$

problem postane linearno ločljiv.

(c) Narišite odločitveno drevo, ki pravilno reši problem XOR.

(d) Kako je ta primer povezan z **jedrskim trikom** (angl. *kernel trick*)?

Rešitev.

(a) Predpostavimo, da obstaja linearna meja

$$w_1 x_1 + w_2 x_2 + b = 0. \tag{57}$$

Za pravilno klasifikacijo morajo veljati pogoji

$$b < 0 \tag{58}$$

za $(0,0) -> 0$,

$$w_2 + b > 0 \tag{59}$$

za $(0,1) -> 1$,

$$w_1 + b > 0 \tag{60}$$

za $(1,0) -> 1$ in

$$w_1 + w_2 + b < 0 \tag{61}$$

za $(1,1) -> 0$.

Iz srednjih dveh neenačb dobimo

$$w_1 + w_2 + 2b > 0. \tag{62}$$

Po drugi strani iz

$$b < 0 \tag{63}$$

in

$$w_1 + w_2 + b < 0 \tag{64}$$

sledi

$$w_1 + w_2 + 2b < 0. \tag{65}$$

Dobimo protislovje. Linearna ločitev zato ne obstaja.

(b) Z novo značilko

$$x_3 = x_1 x_2 \tag{66}$$

dobimo točke

$$(0,0,0), \quad (0,1,0), \quad (1,0,0), \quad (1,1,1). \tag{67}$$

Ravnina

$$x_1 + x_2 - 2x_3 = 0{,}5 \tag{68}$$

ločuje primera z oznako 1 od primerov z oznako 0.

Problem je torej v razširjenem prostoru linearno ločljiv.

(c) Odločitveno drevo lahko najprej razdeli po x_1:

- če je $x_1 = 0$, je izhod enak x_2;
- če je $x_1 = 1$, je izhod enak $1 - x_2$.

Torej:

x_1	x_2	$\hat{y}$
0	0	0
0	1	1
1	0	1
1	1	0

(d) Jedrski trik omogoča, da implicitno delamo v višjedimenzionalnem prostoru, ne da bi vse nove značilke eksplicitno izračunali.

Polinomsko jedro druge stopnje na primer implicitno vsebuje tudi navzkrižni člen

$$x_1 x_2, \tag{69}$$

ki v tem primeru omogoči linearno ločitev XOR.

8. **Tehnično-etični primer**

   Logistična regresija, ki jo uporablja banka, vsebuje značilko *število let na trenutnem naslovu*. Njen koeficient je negativen, kar pomeni, da daljše bivanje na istem naslovu model povezuje z nižjim tveganjem neplačila.

   Ta značilka je hkrati povezana s starostjo.

   (a) Zakaj bi lahko uporaba te značilke povzročila starostno diskriminacijo, čeprav starost sama ni vhodna značilka?

   (b) Po odstranitvi značilke AUC pade z 0,82 na 0,79. Opišite nastali kompromis.

   (c) Katere metrike pravičnosti (angl. *fairness metrics*) bi izračunali za dodatno analizo?

   Rešitev.

   (a) Število let na trenutnem naslovu je povezano s starostjo, saj starejši ljudje v povprečju lahko dlje živijo na istem naslovu. Značilka zato lahko deluje kot **posredna oziroma nadomestna značilka** za starost.

   Model lahko tako posredno diskriminira mlajše prosilce, čeprav starost ni eksplicitno uporabljena.

   (b) AUC se zmanjša z 0,82 na 0,79, kar pomeni padec za 0,03 oziroma tri odstotne točke AUC.

   Odstranitev značilke lahko zmanjša napovedno uspešnost, hkrati pa zmanjša nevarnost posredne starostne diskriminacije.

   Gre torej za kompromis med napovedno uspešnostjo in pravičnostjo. Ali je padec sprejemljiv, je odvisno od konteksta, zahtev glede pravičnosti in minimalne zahtevane uspešnosti modela.

   (c) Izračunali bi lahko na primer:

   - **demografsko pariteto**: ali imajo različne starostne skupine podobne stopnje odobritve;
   - **equalized odds**: ali so TPR in FPR podobni med starostnimi skupinami;
   - **80 % rule**: razmerje stopenj pozitivnih odločitev med skupinami;
   - **kalibracijo po skupinah**: ali enaka napovedana verjetnost pomeni podobno dejansko tveganje v različnih starostnih skupinah.

9. **IAIO-slog: še en algoritem, ki ga niste videli**

   Obravnavajte **prototipni klasifikator** (angl. *Prototype Classifier*).

   Za vsak razred c najprej izračunamo središče oziroma povprečje vseh učnih primerov tega razreda:

$$\mu_c = \frac{1}{|C_c|} \sum_{x \in C_c} x. \tag{70}$$

   Nov primer x nato razvrstimo v razred, katerega središče mu je najbližje:

$$\hat{y} = \arg\min_c \|x - \mu_c\|. \tag{71}$$

   Odgovorite:

   (a) Kakšne oblike je meja odločanja med dvema razredoma? Pomagajte si s pogojem

$$\|x - \mu_1\| = \|x - \mu_2\|. \tag{72}$$

(b) Kako se ta metoda primerja s k-NN, če bi pri k-NN upoštevali vse učne primere?

(c) Kakšne lastnosti podatkov bi morale veljati, da bi ta klasifikator dobro deloval?

(d) Sestavite preprost dvodimenzionalni primer podatkov, pri katerem prototipni klasifikator zelo slabo deluje. Katera lastnost podatkov povzroči težavo?

(e) Koliko parametrov mora klasifikator shraniti? Primerjajte ga s k-NN in logistično regresijo.

Rešitev.

(a) Meja odločanja je množica točk, ki so enako oddaljene od obeh centroidov:

$$\|x - \mu_1\| = \|x - \mu_2\|. \tag{73}$$

Če obe strani kvadriramo, dobimo

$$\|x - \mu_1\|^2 = \|x - \mu_2\|^2. \tag{74}$$

Po razširitvi se člen $x^\top x$ na obeh straneh izniči, zato ostane linearna enačba v x.

Meja je torej **hiperravnina**, geometrijsko pa pravokotna simetrala daljice med μ_1 in μ_2.

(b) Če pri k-NN vzamemo k enak številu vseh učnih primerov, klasifikator izvede večinsko glasovanje vseh primerov in zato za vsak x praviloma napove večinski razred.

Prototipni klasifikator namesto tega vsak razred predstavi z njegovim centroidom in izbere najbližji centroid. Metodi zato na splošno nista enaki.

(c) Metoda dobro deluje, kadar vsak razred tvori približno eno kompaktno, približno sferično skupino, katere povprečje dobro predstavlja razred.

(d) Naj bo razred A

$$C_A = \{(0, 0), (10, 10)\} \tag{75}$$

in razred B

$$C_B = \{(5, 5)\}. \tag{76}$$

Centroid razreda A je

$$\mu_A = \left(\frac{0 + 10}{2}, \frac{0 + 10}{2}\right) = (5, 5), \tag{77}$$

centroid razreda B pa je prav tako

$$\mu_B = (5, 5). \tag{78}$$

Centroid razreda A tako leži na mestu drugega razreda in sploh ne predstavlja dejanske lokacije njegovih primerov.

Težava nastane zaradi večmodalne oziroma močno razpršene porazdelitve razreda.

(e) Če imamo C razredov in d značilk, mora prototipni klasifikator shraniti

$$C \cdot d \tag{79}$$

vrednosti.

k-NN mora shraniti vse učne primere, torej približno

$$n \cdot d \tag{80}$$

vrednosti.

Logistična regresija mora pri večrazredni različici shraniti približno

$$(d+1)C \tag{81}$$

parametrov.

Prototipni klasifikator je zato pomnilniško zelo kompakten.

10. **En korak gradientnega spusta**

Za model

$$h_\theta(x) = \theta_0 + \theta_1 x \tag{82}$$

naj bo trenutno

$$\theta_0 = 1, \qquad \theta_1 = 2. \tag{83}$$

Gradient funkcije izgube pri trenutnih parametrih je

$$\nabla J(\theta) = \begin{pmatrix} 2 \\ -4 \end{pmatrix}, \tag{84}$$

stopnja učenja pa je

$$\alpha = 0{,}1. \tag{85}$$

Izvedite en korak gradientnega spusta in izračunajte novi vrednosti θ_0 in θ_1.

Rešitev.

Gradientni spust uporablja posodobitev

$$\theta^{\text{novo}} = \theta^{\text{staro}} - \alpha \nabla J(\theta). \tag{86}$$

Za θ_0 dobimo

$$\theta_0^{\text{novo}} = 1 - 0{,}1 \cdot 2 = 0{,}8. \tag{87}$$

Za θ_1 dobimo

$$\theta_1^{\text{novo}} = 2 - 0{,}1 \cdot (-4) = 2 + 0{,}4 = 2{,}4. \tag{88}$$

Po enem koraku sta parametra

$$\boxed{\theta_0 = 0{,}8, \qquad \theta_1 = 2{,}4}. \tag{89}$$

11. **Več značilk in polinomska regresija**

Model ima obliko

$$h_\theta(x) = 3 + 2x_1 - 4x_2. \tag{90}$$

(a) Izračunajte napoved za $x_1 = 5$ in $x_2 = 2$.

(b) Kako bi model razširili, če bi želeli zajeti kvadratno odvisnost od x_1?

(c) Zakaj model, ki vsebuje člen x_1^2, še vedno imenujemo linearen glede na parametre?

(d) Kakšna težava se lahko pojavi, če dodamo zelo veliko polinomskih členov visoke stopnje?

Rešitev.

(a) Vstavimo vrednosti:

$$h_\theta(5, 2) = 3 + 2 \cdot 5 - 4 \cdot 2 = 3 + 10 - 8 = \boxed{5}. \tag{91}$$

(b) Dodamo lahko novo polinomsko značilko x_1^2:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2. \tag{92}$$

(c) Model je še vedno linearen *glede na parametre*, saj se parametri

$$\theta_0, \theta_1, \theta_2, \theta_3 \tag{93}$$

v enačbi pojavljajo samo linearno. Nelinearna je transformacija vhodne značilke x_1.

(d) Z dodajanjem velikega števila členov visoke stopnje postane model zelo prilagodljiv in lahko začne opisovati tudi šum v učnih podatkih. Pride lahko do **prekomernega prileganja** oziroma overfittinga.

12. **Logistična regresija in logaritemska izguba**

Model za binarno klasifikacijo pravemu razredu $y = 1$ pripiše verjetnost

$$\hat{p} = 0{,}9. \tag{94}$$

Drug model istemu primeru pripiše

$$\hat{p} = 0{,}1. \tag{95}$$

Za oba izračunajte binarno navzkrižno entropijo

$$L(y, \hat{p}) = -\left[y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})\right]. \tag{96}$$

Kateri model je močneje kaznovan in zakaj?

Kaj bi se zgodilo z izgubo, če bi model pravemu razredu $y = 1$ pripisal verjetnost, ki se približuje 0?

Rešitev.

Ker je $y = 1$, se izraz poenostavi v

$$L(1, \hat{p}) = -\log(\hat{p}). \tag{97}$$

Za prvi model:

$$L(1, 0{,}9) = -\log(0{,}9) \approx \boxed{0{,}105}. \tag{98}$$

Za drugi model:

$$L(1, 0{,}1) = -\log(0{,}1) \approx \boxed{2{,}303}. \tag{99}$$

Drugi model je zato veliko močneje kaznovan. Pravemu razredu je pripisal samo 10 % verjetnost, kar pomeni, da je bil zelo samozavesten v napačno smer.

Ko se

$$\hat{p} \to 0, \tag{100}$$

velja

$$-\log(\hat{p}) \to \infty. \tag{101}$$

Logaritemska izguba zato zelo močno kaznuje samozavestne napačne napovedi.

13. **Logistična regresija in meja odločanja**

Podan je model

$$h_\theta(x) = \sigma(-4 + 2x_1 + x_2). \tag{102}$$

Pri pragu 0,5:

(a) zapišite enačbo meje odločanja;

(b) določite, na kateri strani meje model napove razred 1;

(c) določite napoved za točko $(x_1, x_2) = (1, 1)$;

(d) določite napoved za točko $(x_1, x_2) = (3, 1)$.

Rešitev.

Sigmoidna funkcija ima vrednost 0,5 natanko pri vhodu 0. Zato meja odločanja ustreza pogoju

$$-4 + 2x_1 + x_2 = 0. \tag{103}$$

Enačbo lahko zapišemo tudi kot

$$\boxed{x_2 = 4 - 2x_1}. \tag{104}$$

Model napove razred 1, kadar je argument sigmoidne funkcije pozitiven:

$$-4 + 2x_1 + x_2 > 0. \tag{105}$$

Za $(1, 1)$ dobimo

$$-4 + 2 \cdot 1 + 1 = -1 < 0. \tag{106}$$

Zato je

$$h_\theta(1, 1) < 0{,}5 \tag{107}$$

in model napove

$$\boxed{0}. \tag{108}$$

Za $(3, 1)$ dobimo

$$-4 + 2 \cdot 3 + 1 = 3 > 0. \tag{109}$$

Zato je

$$h_\theta(3, 1) > 0{,}5 \tag{110}$$

in model napove

$$\boxed{1}. \tag{111}$$

14. **Naivni Bayes**

Želimo razvrstiti elektronsko sporočilo kot neželeno (S) ali običajno (N).

Poznamo:

$$P(S) = 0{,}2, \qquad P(N) = 0{,}8. \tag{112}$$

V sporočilu se pojavita besedi *free* in *meeting*. Ocenjene pogojne verjetnosti so

$$P(\text{free} \mid S) = 0{,}7, \qquad P(\text{meeting} \mid S) = 0{,}1, \tag{113}$$

ter

$$P(\text{free} \mid N) = 0{,}05, \qquad P(\text{meeting} \mid N) = 0{,}4. \tag{114}$$

Ob predpostavki pogojne neodvisnosti izračunajte količini

$$P(S)P(\text{free} \mid S)P(\text{meeting} \mid S) \tag{115}$$

in

$$P(N)P(\text{free} \mid N)P(\text{meeting} \mid N). \tag{116}$$

V kateri razred bi Naivni Bayes razvrstil sporočilo?

Zakaj za samo izbiro razreda ni treba izračunati normalizacijskega imenovalca iz Bayesovega izreka?

Rešitev.

Za neželeno sporočilo dobimo nenormalizirano posteriorno vrednost

$$P(S)P(\text{free} \mid S)P(\text{meeting} \mid S) = 0{,}2 \cdot 0{,}7 \cdot 0{,}1 = \boxed{0{,}014}. \tag{117}$$

Za običajno sporočilo dobimo

$$P(N)P(\text{free} \mid N)P(\text{meeting} \mid N) = 0{,}8 \cdot 0{,}05 \cdot 0{,}4 = \boxed{0{,}016}. \tag{118}$$

Ker velja

$$0{,}016 > 0{,}014, \tag{119}$$

Naivni Bayes sporočilo razvrsti kot

$$\boxed{N}, \tag{120}$$

torej kot običajno sporočilo.

Po Bayesovem izreku bi za posteriorno verjetnost sicer delili z verjetnostjo opaženih besed. Ta imenovalec je pri primerjavi obeh razredov enak:

$$P(\text{free}, \text{meeting}). \tag{121}$$

Ker oba izraza delimo z isto pozitivno vrednostjo, se njun vrstni red ne spremeni. Za izbiro razreda zato zadostuje primerjava nenormaliziranih vrednosti.

15. **Izbira algoritma**

Za vsak spodnji problem izberite enega ali več smiselnih algoritmov iz tega poglavja in izbiro utemeljite.

(a) Želimo preprost in razložljiv model za napovedovanje zvezne vrednosti.

(b) Imamo visokodimenzionalne redke podatke iz besedila.

(c) Imamo nelinearno klasifikacijsko mejo in srednje veliko podatkovno množico.

(d) Potrebujemo model, katerega odločitve lahko predstavimo kot zaporedje preprostih pravil.

(e) Potrebujemo verjetnostni izhod za binarno klasifikacijo in pričakujemo približno linearno mejo odločanja.

Pri odgovoru navedite tudi eno pomembno omejitev izbranega algoritma.

Rešitev.

(a) **Linearna regresija.**

Primerna je za napovedovanje zvezne vrednosti in je enostavno razložljiva: vsak koeficient pove, kako sprememba posamezne značilke vpliva na napoved.

Pomembna omejitev je predpostavka približno linearne povezave med značilkami in ciljno spremenljivko. Močno nelinearnih odnosov brez dodatnih značilk ne more dobro modelirati.

(b) **Naivni Bayes ali linearni SVM.**

Oba algoritma sta primerna za visokodimenzionalne redke podatke, kakršni nastanejo pri predstavitvi besedila.

Naivni Bayes je zelo hiter, vendar uporablja močno predpostavko pogojne neodvisnosti značilk.

Linearni SVM dobro deluje v visokih dimenzijah, vendar njegov osnovni izhod ni neposredno kalibrirana verjetnost.

(c) **SVM z RBF-jedrom.**

RBF-jedro omogoča nelinearno mejo odločanja in je lahko dobra izbira za srednje velike podatkovne množice.

Omejitev je občutljivost na izbiro hiperparametrov, predvsem C in parametra jedra. Učenje je lahko tudi računsko zahtevno pri zelo velikih podatkovnih množicah.

(d) **Odločitveno drevo.**

Odločitev lahko predstavimo kot zaporedje pravil oblike

če velja pogoj, pojdi levo; sicer pojdi desno.

Zato je drevo zelo razložljivo.

Pomembna omejitev je, da se globoka drevesa hitro prekomerno prilegajo učnim podatkom in so lahko nestabilna glede na majhne spremembe podatkov.

(e) **Logistična regresija.**

Logistična regresija neposredno vrne vrednost med 0 in 1, ki jo interpretiramo kot ocenjeno verjetnost pozitivnega razreda. Njena meja odločanja je v osnovnem prostoru značilk linearna.

Pomembna omejitev je, da brez transformacije značilk ne more dobro predstaviti močno nelinearne meje odločanja.

3 Spodbujevano učenje

3.1 Naloge z rešitvami

1. **Bellmanove enačbe**

   Obravnavajte Markov proces odločanja z dvema stanjema A in B ter faktorjem znižanja $\gamma = 0{,}9$.

   Nagrade za posamezne prehode so:

   - $A \to A$: $+1$,
   - $A \to B$: 0,
   - $B \to A$: $+2$,
   - $B \to B$: $+1$.

   Zapišite Bellmanovi enačbi za optimalni vrednostni funkciji $V^*(A)$ in $V^*(B)$, pri čemer predpostavite, da agent vedno izbere optimalno akcijo.

   Rešitev.

   Bellmanova enačba optimalnosti pove, da je vrednost stanja enaka največji možni vsoti takojšnje nagrade in znižane vrednosti naslednjega stanja.

   Za stanje A lahko agent ostane v A in prejme nagrado $+1$ ali preide v B in prejme nagrado 0. Zato velja

   $$V^*(A) = \max\left\{1 + 0{,}9V^*(A),\ 0 + 0{,}9V^*(B)\right\}. \tag{122}$$

   Za stanje B lahko agent preide v A in prejme nagrado $+2$ ali ostane v B in prejme nagrado $+1$:

   $$V^*(B) = \max\left\{2 + 0{,}9V^*(A),\ 1 + 0{,}9V^*(B)\right\}. \tag{123}$$

   Preverimo strategijo, pri kateri iz A ostanemo v A, iz B pa preidemo v A.

   Za stanje A dobimo

   $$V^*(A) = 1 + 0{,}9V^*(A). \tag{124}$$

   Od tod

   $$0{,}1V^*(A) = 1, \tag{125}$$

   zato je

   $$\boxed{V^*(A) = 10}. \tag{126}$$

   Za stanje B potem dobimo

   $$V^*(B) = 2 + 0{,}9 \cdot 10 = \boxed{11}. \tag{127}$$

   Preverimo še možnost ostanka v B:

$$1 + 0{,}9V^*(B) = 1 + 0{,}9 \cdot 11 = 10{,}9. \tag{128}$$

Ker velja

$$11 > 10{,}9, \tag{129}$$

je iz stanja B res optimalno preiti v A.

Optimalna strategija je torej:

- v stanju A ostanemo v A;
- v stanju B preidemo v A.

2. **Posodobitev pri Q-učenju**

Pri Q-učenju naj bodo $\alpha = 0{,}1$, $\gamma = 0{,}9$ in $Q(s, a) = 5$.

Agent izvede akcijo a, prejme nagrado $r = 2$ in preide v stanje s', za katerega velja

$$\max_{a'} Q(s', a') = 8. \tag{130}$$

Izračunajte novo vrednost $Q(s, a)$.

Rešitev.

Pravilo za posodobitev pri Q-učenju je

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]. \tag{131}$$

Vstavimo podane vrednosti:

$$Q(s, a) \leftarrow 5 + 0{,}1 \left[2 + 0{,}9 \cdot 8 - 5 \right]. \tag{132}$$

Najprej izračunamo ciljno vrednost:

$$2 + 0{,}9 \cdot 8 = 2 + 7{,}2 = 9{,}2. \tag{133}$$

Napaka časovne razlike je

$$9{,}2 - 5 = 4{,}2. \tag{134}$$

Zato dobimo

$$Q(s, a) = 5 + 0{,}1 \cdot 4{,}2 = 5 + 0{,}42 = \boxed{5{,}42}. \tag{135}$$

3. **Raziskovanje in izkoriščanje**

   Pojasnite kompromis med **raziskovanjem** (angl. *exploration*) in **izkoriščanjem** (angl. *exploitation*).

   Zakaj agent ne more vedno preprosto izbrati trenutno najboljše znane akcije?

   Rešitev.

   Pri **izkoriščanju** agent uporabi svoje trenutno znanje in izbere akcijo, za katero trenutno ocenjuje, da bo prinesla največjo nagrado.

   Pri **raziskovanju** agent preizkuša tudi druge, manj raziskane akcije, da bi pridobil nove informacije o okolju.

   Če bi agent vedno izbral samo trenutno najboljšo znano akcijo, bi lahko spregledal boljšo možnost, ki je še nikoli ni preizkusil. Ocene akcij so predvsem na začetku učenja lahko nenatančne.

   Akcija, ki se trenutno zdi slabša, je lahko v resnici boljša, vendar agent tega brez raziskovanja ne more ugotoviti.

   Raziskovanje je zato potrebno, da agent pridobiva nove informacije in se izogne temu, da bi se prezgodaj ustalil pri dobri, vendar ne optimalni strategiji.

4. **Vpliv faktorja znižanja**

   Kaj se zgodi z vedenjem optimalne strategije, ko se γ približuje 0?

   Kaj pa, ko se γ približuje 1?

   Pojasnite, kako faktor znižanja vpliva na pomen takojšnjih in prihodnjih nagrad ter s tem na skupno nagrado.

   Rešitev.

   Skupna znižana nagrada je

   $$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots . \tag{136}$$

   Ko velja

   $$\gamma \to 0, \tag{137}$$

   imajo prihodnje nagrade zelo majhen vpliv. Pomembna je skoraj samo takojšnja nagrada. Agent zato postane **kratkoviden** in izbira predvsem akcije z največjo neposredno nagrado.

   Ko velja

   $$\gamma \to 1, \tag{138}$$

   se prihodnje nagrade zmanjšujejo vedno manj. Agent zato bolj upošteva dolgoročne posledice svojih odločitev in načrtuje bolj dolgoročno.

   Pri γ blizu 1 so torej prihodnje nagrade skoraj enako pomembne kot takojšnje. Pri neskončnem časovnem horizontu pa lahko to povzroči težave: če nagrade niso omejene, lahko vrednostne funkcije divergirajo, konvergenca učnih algoritmov pa je lahko počasnejša.

5. **Izpeljava Bellmanove enačbe**

Začnite z enačbo

$$Q^{\pi}(s,a) = \mathbb{E}\left[R_{t+1} + \gamma Q^{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]. \tag{139}$$

Iz nje izpeljite **Bellmanovo enačbo optimalnosti** (angl. *Bellman optimality equation*) za Q^*.

Rešitev.

Za poljubno strategijo π naslednjo akcijo A_{t+1} določa strategija π.

Pri optimalni strategiji π^* pa v naslednjem stanju izberemo akcijo z največjo možno optimalno Q-vrednostjo:

$$\max_{a'} Q^*(S_{t+1}, a'). \tag{140}$$

Zato v začetni enačbi izraz

$$Q^{\pi}(S_{t+1}, A_{t+1}) \tag{141}$$

nadomestimo z

$$\max_{a'} Q^*(S_{t+1}, a'). \tag{142}$$

Dobimo

$$\boxed{Q^*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} Q^*(S_{t+1}, a') \mid S_t = s, A_t = a\right]}. \tag{143}$$

Če pričakovano vrednost eksplicitno zapišemo kot vsoto po možnih naslednjih stanjih, dobimo

$$\boxed{Q^*(s,a) = \sum_{s'} P(s' \mid s, a)\left[R(s,a,s') + \gamma \max_{a'} Q^*(s', a')\right]}. \tag{144}$$

To je Bellmanova enačba optimalnosti za Q^*.

6. **IAIO-slog: ko požrešna intuicija odpove**

Obravnavajte Markov proces odločanja, pri katerem ima agent v začetnem stanju s_0 tri možne akcije:

- **Akcija A:** takojšnja nagrada +10, nato se epizoda konča;
- **Akcija B:** takojšnja nagrada 0, nato agent preide v stanje X;
- **Akcija C:** takojšnja nagrada 0, nato agent preide v stanje Y.

Iz stanja X je na voljo samo ena akcija, ki prinese nagrado +5 in konča epizodo.

Iz stanja Y je na voljo samo ena akcija, ki prinese nagrado +20 in konča epizodo.

Naj bo faktor znižanja $\gamma = 0{,}9$.

Odgovorite:

(a) Izračunajte $Q^*(s_0, A)$, $Q^*(s_0, B)$ in $Q^*(s_0, C)$.

(b) Požrešen agent (angl. *greedy agent*) je do sedaj preizkusil samo akcijo A in zanjo prejel nagrado +10. Zakaj bo tak agent brez raziskovanja vedno znova izbiral akcijo A?

(c) Agent uporablja ε-požrešno strategijo (angl. *ε-greedy strategy*) z $\varepsilon = 0{,}1$.

Po 1000 epizodah približno kolikokrat bo preizkusil posamezno akcijo? Ali bo odkril optimalno strategijo?

(d) Predlagajte preprosto spremembo, zaradi katere bi požrešno vedenje hitreje odkrilo optimalno strategijo.

Rešitev.

(a) Akcija A prinese nagrado +10 in takoj konča epizodo, zato je

$$\boxed{Q^*(s_0, A) = 10}. \tag{145}$$

Pri akciji B agent najprej prejme 0, nato pa v naslednjem koraku +5:

$$Q^*(s_0, B) = 0 + 0{,}9 \cdot 5 = \boxed{4{,}5}. \tag{146}$$

Pri akciji C agent najprej prejme 0, nato pa +20:

$$Q^*(s_0, C) = 0 + 0{,}9 \cdot 20 = \boxed{18}. \tag{147}$$

Ker velja

$$18 > 10 > 4{,}5, \tag{148}$$

je optimalna akcija

$$\boxed{C}. \tag{149}$$

(b) Požrešen agent vedno izbere akcijo z največjo trenutno znano Q-vrednostjo.

Če je preizkusil samo akcijo A, ima na primer ocene

$$Q(s_0, A) = 10, \qquad Q(s_0, B) = 0, \qquad Q(s_0, C) = 0. \tag{150}$$

Ker je

$$10 > 0, \tag{151}$$

bo vedno znova izbral A. Ker B in C nikoli ne preizkusi, tudi nikoli ne odkrije, da je C dejansko boljša.

(c) Pri $\varepsilon = 0{,}1$ agent v približno 10 % epizod raziskuje, v približno 90 % pa izbere trenutno požrešno akcijo.

V 1000 epizodah to pomeni približno

$$1000 \cdot 0{,}1 = 100 \tag{152}$$

raziskovalnih epizod.

Ker pri raziskovanju naključno izbira med tremi akcijami, bo vsako izmed njih raziskoval približno

$$\frac{100}{3} \approx 33 \tag{153}$$

krat.

Če predpostavimo, da je A na začetku požrešna akcija, dobimo približno

$$A \approx 900 + 33 = 933, \tag{154}$$

$$B \approx 33, \tag{155}$$

$$C \approx 33. \tag{156}$$

Torej približno

$$\boxed{A \approx 933, \qquad B \approx 33, \qquad C \approx 33}. \tag{157}$$

Po približno 33 preizkusih akcije C bo agent ugotovil, da ima ta akcija vrednost 18, zato bo C postala nova požrešna akcija.

Agent bo torej sčasoma odkril optimalno strategijo.

(d) Uporabimo lahko **optimistično inicializacijo Q-vrednosti**. Na primer:

$$Q_0(s_0, A) = Q_0(s_0, B) = Q_0(s_0, C) = 100. \tag{158}$$

Ko agent preizkusi A in ugotovi, da je njena dejanska vrednost le 10, ostaneta B in C še vedno ocenjeni z večjo vrednostjo. Požrešen agent bo zato naravno preizkusil tudi njiju.

Ko preizkusi vse tri akcije, hitro odkrije, da ima C največjo vrednost:

$$Q^*(s_0, C) = 18. \tag{159}$$

Optimistična inicializacija tako spodbuja raziskovanje tudi pri sicer požrešnem agentu.

7. **Robot na mreži**

Robot se premika po mreži velikosti 4×4.

Cilj je v polju $(4, 4)$, nevarno polje pa v $(2, 2)$. Nagrade so:

- $+10$ za prihod na cilj;
- -10 za prihod v nevarno polje;
- -1 za vsak izvedeni korak.

Odgovorite:

(a) Koliko stanj vsebuje okolje? Koliko različnih deterministčnih strategij je možnih?

(b) Pri faktorju znižanja $\gamma = 0{,}9$ bo robot raje izbral dolgo varno pot ali krajšo, vendar bolj tvegano pot? Odgovor utemeljite.

(c) Zapišite posodobitev pri Q-učenju za stanje $(1, 1)$ in akcijo *desno*, če agent pristane v stanju $(2, 1)$ in prejme nagrado $r = -1$.

Rešitev.

(a) Mreža 4×4 vsebuje

$$4 \cdot 4 = \boxed{16} \tag{160}$$

stanj.

Polje $(2, 2)$ je terminalno nevarno stanje, polje $(4, 4)$ pa terminalni cilj. Odločitve zato sprejemamo v

$$16 - 2 = 14 \tag{161}$$

neterminalnih stanjih.

Če ima vsako neterminalno stanje največ štiri možne akcije, je zgornja meja števila deterministčnih strategij

$$4^{14} = \boxed{268\,435\,456}. \tag{162}$$

To je zgornja meja, saj imajo robna in kotna polja dejansko manj veljavnih premikov.

(b) Ker je $\gamma = 0{,}9$, agent upošteva tako takojšnje kot prihodnje nagrade.

Za ilustracijo lahko primerjamo dolgo varno pot s šestimi koraki in kratko tvegano pot s tremi koraki.

Približna vrednost varne poti je

$$-6 + 0{,}9^6 \cdot 10 \approx -6 + 5{,}3 = -0{,}7. \tag{163}$$

Če kratka pot uspe in se robot izogne nevarnemu polju, je približna vrednost

$$-3 + 0{,}9^3 \cdot 10 \approx -3 + 7{,}3 = 4{,}3. \tag{164}$$

Če pa robot pri tvegani poti pade v nevarno polje, je vrednost približno

$$-3 + 0{,}9^2(-10) = -3 - 8{,}1 = -11{,}1. \tag{165}$$

Zato robot ne bo nujno izbral niti najdaljše popolnoma varne poti niti najbolj tvegane najkrajše poti. Optimalna strategija bo praviloma izbrala **zmerno varno pot**, ki se izogne nevarnemu polju, vendar hkrati ne naredi nepotrebno dolgega obvoza.

(c) Splošno pravilo za Q-učenje je

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]. \tag{166}$$

V našem primeru je

$$s = (1,1), \qquad a = \text{desno}, \qquad s' = (2,1), \qquad r = -1. \tag{167}$$

Zato dobimo

$$\boxed{Q((1,1),\text{desno}) \leftarrow Q((1,1),\text{desno}) + \alpha \left[-1 + \gamma \max_{a'} Q((2,1),a') - Q((1,1),\text{desno}) \right]}. \tag{168}$$

Ker je v nalogi

$$\gamma = 0{,}9, \tag{169}$$

lahko zapišemo tudi

$$Q((1,1),\text{desno}) \leftarrow Q((1,1),\text{desno}) + \alpha \left[-1 + 0{,}9 \max_{a'} Q((2,1),a') - Q((1,1),\text{desno}) \right]. \tag{170}$$

Brez podane vrednosti α in trenutnih Q-vrednosti izraza ne moremo dodatno numerično izračunati.

4 Evalvacija modelov strojnega učenja

4.1 Naloge z rešitvami

1. **Delitev podatkov**

   Imamo 10 000 podatkovnih primerov.

   Opišite, kako bi podatke razdelili na učno, validacijsko in testno množico, če želimo izbirati hiperparametre. Kolikšen delež podatkov bi namenili posamezni množici?

   Rešitev.

   Primer smiselne delitve je

   $$70\,\% : 15\,\% : 15\,\%. \tag{171}$$

   Pri 10 000 primerih to pomeni:

   - učna množica: 7000 primerov,

- validacijska množica: 1500 primerov,
- testna množica: 1500 primerov.

Učno množico uporabljamo za učenje parametrov modela. Validacijsko množico uporabljamo med razvojem za izbiro hiperparametrov in drugih nastavitev modela.

Testna množica mora ostati ločena od razvoja modela. Uporabimo jo šele na koncu za končno, nepristransko oceno uspešnosti.

2. **Pristranskost in varianca**

   Model doseže 99 % točnost na učni množici, vendar samo 65 % točnost na validacijski množici.

   Diagnosticirajte težavo z vidika pristranskosti in variance. Kaj se najverjetneje dogaja?

   Rešitev.

   Razlika med učno in validacijsko uspešnostjo je

   $$99\,\% - 65\,\% = 34 \tag{172}$$

   odstotnih točk.

   Zelo visoka učna uspešnost kaže, da se model zelo dobro prilagodi učnim podatkom. Slaba validacijska uspešnost pa kaže, da tega znanja ne zna dobro prenesti na nove primere.

   Gre za **prekomerno prileganje** (angl. *overfitting*), oziroma za režim

   $$\boxed{\text{nizka pristranskost, visoka varianca}}. \tag{173}$$

   Model si je učne podatke preveč natančno zapomnil in slabo posplošuje.

   Možne rešitve so:

   - uporaba preprostejšega modela,
   - močnejša regularizacija,
   - več učnih podatkov,
   - pri ustreznih modelih tudi zgodnje ustavljanje ali dropout.

3. **Matrika zamenjav**

   Dana je naslednja matrika zamenjav:

		Napoved	
		+	−
Dejansko	+	450	50
	−	100	400

   Izračunajte:

   (a) natančnost (angl. *precision*);

   (b) priklic (angl. *recall*);

   (c) mero F1 (angl. *F1 score*).

Rešitev.

Iz matrike preberemo

$$TP = 450, \qquad FP = 100, \qquad FN = 50, \qquad TN = 400. \tag{174}$$

Natančnost je

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{450}{450 + 100} = \frac{450}{550} \approx \boxed{81{,}8\,\%}. \tag{175}$$

Priklic je

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{450}{450 + 50} = \frac{450}{500} = \boxed{90\,\%}. \tag{176}$$

Mera F1 je harmonično povprečje natančnosti in priklica:

$$F1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{177}$$

Zato

$$F1 = 2\frac{0{,}818 \cdot 0{,}90}{0{,}818 + 0{,}90} \approx \boxed{85{,}7\,\%}. \tag{178}$$

4. **Natančnost ali priklic**

   V katerih primerih bi želeli optimizirati **natančnost** (angl. *precision*) namesto **priklica** (angl. *recall*)?

   Navedite primer iz resničnega sveta in pojasnite, katera vrsta napake je v njem posebej problematična.

   Rešitev.

   Natančnost želimo posebej optimizirati takrat, ko so **napačno pozitivne napovedi** (angl. *false positives*) zelo drage ali nevarne.

   Dober primer je filter neželene pošte.

   Če sistem neželeno sporočilo označi kot legitimno, gre za napačno negativen primer in uporabnik prejme nekaj neželene pošte.

   Če pa sistem legitimno in pomembno sporočilo napačno označi kot neželeno, gre za napačno pozitiven primer in uporabnik lahko pomembno sporočilo spregleda ali izgubi.

   V takem primeru smo pripravljeni dovoliti nekoliko nižji priklic, če s tem dosežemo visoko natančnost in zmanjšamo število legitimnih sporočil, ki jih sistem napačno označi kot neželena.

5. **Zakaj deluje prečno preverjanje**

   Pri k-kratnem prečnem preverjanju (angl. *k-fold cross-validation*) naučimo k modelov. Vsak model se uči na deležu $(k-1)/k$ podatkov, preostala guba pa služi za validacijo. Končno oceno dobimo s povprečenjem validacijskih rezultatov.

   Odgovorite:

(a) Intuitivno pojasnite, zakaj povprečenje rezultatov čez k gub daje zanesljivejšo oceno kot ena sama delitev na učno in validacijsko množico.

(b) Ko se k približuje številu primerov n, dobimo prečno preverjanje z izpuščanjem enega primera (angl. *leave-one-out cross-validation*). Kaj se pri tem zgodi z:

- pristranskostjo ocene napake;
- varianco ocene napake;
- računsko zahtevnostjo?

(c) Zakaj sta $k = 5$ in $k = 10$ pogosto smiselna kompromisa?

(d) Predpostavimo, da imajo podatki naravne skupine, na primer več meritev istega pacienta. Zakaj lahko običajno k-kratno prečno preverjanje poda preveč optimistično oceno? Kako bi morali podatke razdeliti?

Rešitev.

(a) Pri eni sami delitvi je rezultat močno odvisen od tega, kateri konkretni primeri pristanejo v učni in kateri v validacijski množici. Zaradi tega ima lahko ocena visoko varianco.

Pri k-kratnem prečnem preverjanju vsak primer enkrat nastopa v validacijski množici in večkrat v učni množici. Povprečenje rezultatov čez različne gube zmanjša odvisnost od ene same naključne delitve in zato poda stabilnejšo oceno.

(b) Ko se

$$k \to n, \tag{179}$$

dobimo prečno preverjanje z izpuščanjem enega primera.

- **Pristranskost se zmanjša**, ker se vsak model uči na skoraj celotni podatkovni množici.
- **Varianca ocene se poveča**, ker se učne množice med posameznimi iteracijami skoraj popolnoma prekrivajo. Rezultati posameznih gub so zato močno korelirani.
- **Računska zahtevnost se močno poveča**, saj moramo model naučiti n-krat.

(c) Pri $k = 5$ se vsak model uči na približno 80 % podatkov, pri $k = 10$ pa na približno 90 % podatkov.

To predstavlja dober kompromis med:

- pristranskostjo,
- varianco ocene,
- računsko zahtevnostjo.

Potrebujemo samo 5 oziroma 10 učenj modela, hkrati pa dobimo dovolj stabilno oceno.

(d) Če imamo več meritev istega pacienta, lahko običajna naključna delitev nekatere meritve iste osebe postavi v učno, druge pa v validacijsko množico.

Model lahko zato izkoristi značilnosti konkretnega pacienta in na validaciji doseže nerealno visoko uspešnost.

Uporabiti moramo **skupinsko prečno preverjanje** (angl. *grouped k-fold cross-validation*), pri katerem vse meritve istega pacienta vedno ostanejo v isti gubi.

6. **Razcep pristranskosti in variance**

Pri regresiji lahko pričakovano kvadratno napako zapišemo kot

$$\mathbb{E}\left[(f(x) - \hat{f}(x))^2\right] = \text{Bias}^2 + \text{Variance} + \text{Noise}. \tag{180}$$

Odgovorite:

(a) Definirajte pristranskost s pomočjo $\mathbb{E}[\hat{f}(x)]$ in $f(x)$.

(b) Zakaj ima konstantni model

$$\hat{f}(x) = c \tag{181}$$

običajno visoko pristranskost?

(c) Zakaj ima model, ki popolnoma interpolira vse učne primere, pogosto visoko varianco?

(d) Regularizacija običajno poveča pristranskost modela. Zakaj lahko kljub temu zmanjša napako na novih podatkih?

Rešitev.

(a) Pristranskost pri točki x je

$$\boxed{\text{Bias}(x) = \mathbb{E}[\hat{f}(x)] - f(x)}. \tag{182}$$

Predstavlja sistematično razliko med povprečno napovedjo modela in pravo funkcijo.

(b) Konstantni model

$$\hat{f}(x) = c \tag{183}$$

popolnoma ignorira vhodne značilke.

Če se prava funkcija $f(x)$ spreminja z x, konstantni model teh vzorcev ne more zajeti. Povsod, kjer velja

$$f(x) \neq c, \tag{184}$$

bo imel sistematično napako in zato visoko pristranskost.

(c) Model, ki popolnoma interpolira učne primere, se lahko prilagodi tudi naključnemu šumu.

Če nekoliko spremenimo učno množico, se lahko naučena funkcija močno spremeni. Različne učne množice zato vodijo do zelo različnih napovedi pri istem x, kar pomeni visoko varianco.

(d) Regularizacija omeji fleksibilnost modela. S tem lahko poveča pristranskost, vendar hkrati zmanjša varianco.

Če je zmanjšanje variance večje od povečanja kvadrata pristranskosti, se skupna pričakovana napaka

$$\text{Bias}^2 + \text{Variance} + \text{Noise} \tag{185}$$

zmanjša.

Zato lahko nekoliko bolj pristranski, vendar stabilnejši model bolje posplošuje na nove podatke.

7. **Ko točnost zavaja**

Presejalni test za bolezen, ki je prisotna pri približno 1 % populacije, da naslednjo matriko zamenjav:

		Test	
		+	−
Bolezen	prisotna	90	10
	ni prisotna	495	9405

Odgovorite:

(a) Izračunajte točnost (angl. *accuracy*), natančnost (angl. *precision*), priklic (angl. *recall*) in mero F1.

(b) Zakaj je točnost 94,95 % v tem primeru zavajajoča?

(c) Izračunajte **pozitivno napovedno vrednost** (angl. *positive predictive value*, PPV). Zakaj se tako močno razlikuje od priklica?

Rešitev.

Iz matrike dobimo

$$TP = 90, \qquad FN = 10, \qquad FP = 495, \qquad TN = 9405. \tag{186}$$

Skupaj je

$$90 + 10 + 495 + 9405 = 10\,000 \tag{187}$$

primerov.

(a) Točnost je

$$\text{Accuracy} = \frac{TP + TN}{N} = \frac{90 + 9405}{10\,000} = \boxed{94{,}95\,\%}. \tag{188}$$

Natančnost je

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{90}{90 + 495} = \frac{90}{585} \approx \boxed{15{,}4\,\%}. \tag{189}$$

Priklic je

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{90}{100} = \boxed{90\,\%}. \tag{190}$$

Mera F1 je

$$F1 = 2\frac{0{,}154 \cdot 0{,}90}{0{,}154 + 0{,}90} \approx \boxed{26{,}3\,\%}. \tag{191}$$

(b) Točnost je zavajajoča zaradi zelo neuravnoteženih razredov.

Ker ima bolezen samo približno 1 % ljudi, bi trivialni model, ki bi za *vsakega* človeka napovedal, da bolezni nima, dosegel

$$\boxed{99\,\%} \tag{192}$$

točnost.

To je celo več kot 94,95 % danega modela. Visoka točnost zato sama po sebi ne pomeni, da je model pri odkrivanju bolezni dober.

(c) Pozitivna napovedna vrednost je enaka natančnosti:

$$PPV = \frac{TP}{TP + FP} = \frac{90}{585} \approx \boxed{15{,}4\,\%}. \tag{193}$$

To pomeni, da je samo približno eden od šestih pozitivnih testov dejansko pozitiven.

Priklic pa je

$$90\,\%. \tag{194}$$

Razlika nastane, ker priklic odgovarja na vprašanje:

Kolikšen delež dejansko bolnih smo odkrili?

PPV pa odgovarja na vprašanje:

Kolikšen delež pozitivnih rezultatov je dejansko bolan?

Pri zelo redki bolezni lahko že razmeroma majhna stopnja lažno pozitivnih rezultatov ustvari veliko število napačnih alarmov.

8. **Odkrivanje nepravilne evalvacije**

Sodelavec predstavi model, ki doseže $AUC = 0{,}99$.

Navedite tri načine, na katere bi lahko med razvojem modela nenamerno povzročil, da je rezultat pretirano optimističen.

Rešitev.

Tako visok rezultat bi morali skrbno preveriti. Med možnimi vzroki so:

(a) **Uhajanje informacij iz testne množice.**

Na primer, normalizacijo ali drugo predobdelavo smo prilagodili na celotni podatkovni množici namesto samo na učni množici. Tako so informacije iz testne množice vplivale na učenje.

(b) **Ciljno uhajanje.**

Ena izmed vhodnih značilk neposredno ali posredno razkriva ciljno vrednost.

Primer bi bil uporaba diagnoze ob odpustu za napovedovanje izida ob sprejemu v bolnišnico.

(c) **Časovno uhajanje.**

Model uporablja informacijo iz prihodnosti, ki v trenutku dejanske napovedi še ne bi bila znana.

Rezultat, kot je $AUC = 0{,}99$, je na realnem problemu lahko resničen, vendar je dovolj visok, da je smiselno posebej skrbno preveriti podatkovni proces in možnost uhajanja informacij.

9. **Kateri model bolje posplošuje?**

   Model A doseže 100 % točnost na učni množici in 60 % na testni množici.

   Model B doseže 70 % točnost na učni množici in 65 % na testni množici.

   Primerjajte modela in pojasnite, kaj njuni rezultati povedo o pristranskosti, varianci in posploševanju.

   Rešitev.

   Pri modelu A je razlika med učno in testno uspešnostjo

   $$100\,\% - 60\,\% = 40 \tag{195}$$

   odstotnih točk.

   To je znak zelo močnega prekomernega prileganja. Model A ima zelo nizko učno napako, vendar visoko varianco in slabo posplošuje.

   Pri modelu B je razlika

   $$70\,\% - 65\,\% = 5 \tag{196}$$

   odstotnih točk.

   Njegova učna uspešnost je sicer nižja, vendar je razlika med učenjem in testiranjem bistveno manjša. Model je stabilnejši in bolje posplošuje.

   Poleg tega ima model B tudi višjo testno uspešnost:

   $$65\,\% > 60\,\%. \tag{197}$$

   Zato bi izbrali

   $$\boxed{\text{model B}}. \tag{198}$$

   Zanesljivih 65 % na novih podatkih je bolj uporabnih kot popolna učna uspešnost modela, ki na novih podatkih doseže samo 60 %.

4.1.1 Dodatne naloge z rešitvami

10. **Prag odločanja**

    Binarni klasifikator za posamezne primere vrača verjetnost pozitivnega razreda.

    (a) Kaj se običajno zgodi z natančnostjo in priklicem, če prag odločanja zvišamo z 0,5 na 0,8?

    (b) Kaj se običajno zgodi, če prag znižamo z 0,5 na 0,2?

    (c) Pri presejalnem testu za nevarno bolezen bi želeli zmanjšati število napačno negativnih primerov. Bi prag praviloma zvišali ali znižali? Utemeljite.

    Rešitev.

(a) Če prag zvišamo z 0,5 na 0,8, je težje napovedati pozitivni razred. Manj primerov bo označenih kot pozitivnih.

Običajno zato:

- **natančnost naraste**,
- **priklic pade**.

(b) Če prag znižamo z 0,5 na 0,2, bo več primerov označenih kot pozitivnih.

Običajno zato:

- **priklic naraste**,
- **natančnost pade**.

(c) Pri nevarni bolezni želimo zmanjšati število napačno negativnih primerov, zato želimo povečati priklic.

Prag bi zato praviloma

$$\boxed{\text{znižali}}. \tag{199}$$

S tem bo model več primerov označil kot pozitivne. Odkrili bomo več bolnih oseb, vendar bomo praviloma dobili tudi več lažno pozitivnih rezultatov.

11. **ROC in AUC**

Model A ima $AUC = 0{,}92$, model B pa $AUC = 0{,}71$.

(a) Kaj vrednost AUC meri?

(b) Kako lahko interpretiramo $AUC = 0{,}92$ z vidika naključno izbranega pozitivnega in negativnega primera?

(c) Ali iz višjega AUC neposredno sledi, da bo model pri izbranem pragu imel višjo natančnost in višji priklic? Pojasnite.

Rešitev.

(a) ROC-krivulja prikazuje razmerje med stopnjo resnično pozitivnih primerov

$$TPR = \frac{TP}{TP + FN} \tag{200}$$

in stopnjo lažno pozitivnih primerov

$$FPR = \frac{FP}{FP + TN} \tag{201}$$

pri različnih pragih odločanja.

AUC je ploščina pod ROC-krivuljo in povzema sposobnost modela, da pozitivne primere razvršča višje od negativnih.

(b) $AUC = 0{,}92$ lahko interpretiramo tako, da obstaja približno

$$\boxed{92\,\%} \tag{202}$$

verjetnost, da bo model naključno izbranemu pozitivnemu primeru pripisal višji rezultat kot naključno izbranemu negativnemu primeru.

(c) Ne.

AUC ocenjuje kakovost razvrščanja čez *vse* možne pragove. Natančnost in priklic pa sta odvisna od konkretno izbranega praga.

Model z višjim AUC zato nima nujno hkrati višje natančnosti in višjega priklica pri določenem pragu.

12. **Kalibracija**

Model A med vsemi primeri, ki jim pripiše približno 80 % verjetnost pozitivnega izida, dejansko opazi pozitivni izid le v 45 % primerov.

(a) Ali je model dobro kalibriran? Zakaj?

(b) Ali lahko ima tak model kljub temu visok AUC?

(c) Navedite primer uporabe, kjer bi bila slaba kalibracija posebej problematična.

Rešitev.

(a) Model ni dobro kalibriran.

Pri dobro kalibriranem modelu bi med primeri, ki jim model pripiše približno

$$80\,\% \tag{203}$$

verjetnost pozitivnega izida, pričakovali približno

$$80\,\% \tag{204}$$

dejansko pozitivnih primerov.

Tukaj jih je samo

$$45\,\%. \tag{205}$$

Model je torej preveč samozavesten.

(b) Da. Model ima lahko visok AUC in hkrati slabo kalibracijo.

AUC meri predvsem, ali model pravilno *razvršča* pozitivne primere nad negativne. Ne zahteva, da so napovedane verjetnosti numerično pravilne.

Model lahko zato zelo dobro razvršča primere po tveganju, vendar so njegove vrednosti 0,8, 0,6, 0,4 itd. slabo kalibrirane.

(c) Slaba kalibracija je posebej problematična, kadar sama napovedana verjetnost neposredno vpliva na odločitev.

Primer je ocena finančnega tveganja. Če model napove

$$P(\text{neplačilo}) = 0{,}8, \tag{206}$$

banka pa na podlagi te verjetnosti določi obrestno mero ali višino kapitala za kritje tveganja, mora vrednost 0,8 dejansko predstavljati približno 80 % tveganje.

13. **Regresijske metrike**

Za štiri primere so prave vrednosti

$$y = [3, 5, 2, 10] \tag{207}$$

in napovedi

$$\hat{y} = [2, 5, 4, 7]. \tag{208}$$

Izračunajte:

(a) povprečno absolutno napako MAE;

(b) povprečno kvadratno napako MSE;

(c) koren povprečne kvadratne napake RMSE.

Katera od teh metrik močneje poudari velike posamezne napake?

Rešitev.

Napake so

$$y - \hat{y} = [1, 0, -2, 3]. \tag{209}$$

Absolutne napake so

$$[1, 0, 2, 3]. \tag{210}$$

Zato je

$$MAE = \frac{1 + 0 + 2 + 3}{4} = \frac{6}{4} = \boxed{1{,}5}. \tag{211}$$

Kvadrati napak so

$$[1, 0, 4, 9]. \tag{212}$$

Zato je

$$MSE = \frac{1 + 0 + 4 + 9}{4} = \frac{14}{4} = \boxed{3{,}5}. \tag{213}$$

RMSE je

$$RMSE = \sqrt{MSE} = \sqrt{3{,}5} \approx \boxed{1{,}87}. \tag{214}$$

MSE in RMSE zaradi kvadriranja napak močneje kaznujeta velike posamezne napake kot MAE.

MAE vse napake obravnava linearno, zato je manj občutljiv na velike odklone.

14. **Testna množica ni razvojna množica**

Ekipa razvije deset različnih modelov. Vsakega oceni na isti testni množici in nato izbere model z najboljšim testnim rezultatom.

(a) Kaj je narobe s tem postopkom?

(b) Kakšno vlogo je testna množica dejansko prevzela?

(c) Kako bi postopek spremenili, da bi na koncu še vedno imeli nepristransko končno oceno?

Rešitev.

(a) Z izbiro najboljšega modela na podlagi testnih rezultatov smo informacije iz testne množice uporabili za odločanje med razvojem.

Čeprav modelov neposredno nismo učili na testnih podatkih, smo se njihovim rezultatom na testni množici posredno prilagodili.

Končni testni rezultat zato ni več nepristranska ocena uspešnosti na popolnoma novih podatkih.

(b) Testna množica je dejansko prevzela vlogo

$$\boxed{\text{validacijske množice}}. \tag{215}$$

Uporabili smo jo za izbiro med različnimi modeli.

(c) Podatke bi morali razdeliti na:

- učno množico,
- validacijsko množico,
- testno množico.

Deset modelov primerjamo na validacijski množici in na njej izberemo najboljšega.

Testno množico pustimo popolnoma nedotaknjeno. Ko je model dokončno izbran, ga na testni množici ocenimo samo na koncu.

Tako dobimo nepristransko končno oceno.

15. **Izbira primerne metrike**

Za vsak primer izberite metriko oziroma metrike, ki bi bile posebej smiselne, in odgovor utemeljite.

(a) Odkrivanje zelo redke nevarne bolezni, pri kateri je spregledan primer posebej nevaren.

(b) Filter neželene pošte, pri katerem želimo predvsem preprečiti, da bi legitimno pošto označili kot neželeno.

(c) Uravnotežena klasifikacijska naloga, pri kateri so vse vrste napak približno enako pomembne.

(d) Sistem, pri katerem napovedana verjetnost neposredno določa višino finančnega tveganja.

(e) Primerjava sposobnosti več klasifikatorjev za razvrščanje pozitivnih primerov nad negativne, ne da bi vnaprej določili prag.

Rešitev.

(a) **Priklic oziroma občutljivost.**

Pri nevarni bolezni je posebej problematičen napačno negativen primer: bolan človek, ki ga model ne odkrije.

Zato želimo visok

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{216}$$

Pri zelo redki bolezni je smiselno poleg priklica spremljati tudi natančnost oziroma PPV, saj lahko veliko lažno pozitivnih rezultatov povzroči veliko nepotrebnih nadaljnjih pregledov.

(b) **Natančnost.**

Pozitivni razred naj predstavlja neželeno pošto. Napačno pozitiven primer je potem legitimno sporočilo, ki ga filter označi kot neželeno.

Ker želimo predvsem zmanjšati takšne napake, je pomembna visoka natančnost:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{217}$$

(c) **Točnost** je smiselna, ker so razredi uravnoteženi in imajo različne vrste napak približno enako ceno.

Uporabimo lahko tudi mero F1, če želimo hkrati povzeti natančnost in priklic.

(d) Potrebujemo dobro **kalibracijo verjetnosti**.

Če napovedana verjetnost neposredno določa finančno tveganje, mora na primer napoved

$$P(y = 1) = 0{,}20 \tag{218}$$

dejansko ustrezati približno 20 % pogostosti pozitivnega izida med takšnimi primeri.

Za numerično ocenjevanje verjetnostnih napovedi sta primerni na primer **logaritemska izguba** ali **Brierjeva mera**, poleg tega pa preverimo kalibracijsko krivuljo.

(e) **ROC-AUC.**

AUC meri sposobnost modela, da pozitivnim primerom pripiše višje rezultate kot negativnim, in za primerjavo ne zahteva vnaprej določenega praga.

Zato je primerna za primerjavo sposobnosti razvrščanja različnih klasifikatorjev.