Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

28. siječnja 2026.

Strojno učenje 1 – Domaća zadaća 6

*Domaća zadaća sadrži **6 pitanja** i ukupno nosi najviše 6 bodova (skalirano na 2 boda na predmetu). Točan odgovor nosi 1 bod. Za razliku od bodovanja ispita, netočni odgovori ne nose negativne bodove. Prije nego što krenete rješavati ove zadatke preporučujemo da riješite sve zadatke iz dijela "Zadatci za učenje" za sve nastavne cjeline obuhvaćene ovom zadaćom.*

***Upozorenje:** Za svaki zadatak na koji odgovorite, morate predati i ručno ispisan postupak. Ako to ne učinite čak i za samo jedan zadatak, gubite sve bodove prikupljene kroz aktivnost na domaćim zadaćama.*

***Napomena:** Ova domaća zadaća je personalizirana. Svaki student dobiva jedinstvenu varijantu zadataka.*

19. Grupiranje

19.1 (P) Raspolažemo sljedećim neoznačenim podatcima u dvodimenzijskome ulaznom prostoru:

$$\mathcal{D} = \big\{(1,1),(1,2),(1,3),(3,8),(4,2),(5,4),(6,2)\big\}$$

Podatke grupiramo algoritmima K-sredina i K-medoida u $K = 3$ grupe. Kao početna središta odnosno medoide koristimo vektore iz $\mathcal{D}$. Za algoritam K-medoida kao funkciju udaljenosti koristimo L_1-normu. Razmotrite najbolje moguće grupiranje koje možemo dobiti algoritmom K-sredina odnosno algoritmom K-medoida s L_1-normom. Ako ima više takvih grupiranja, razmotrite ih sve. Zanima nas koliko medoidi mogu najviše odstupati od centroida. Za svaku grupu izračunajte euklidsku udaljenost između njezinog centroida i njezinog medoida. **Koliko iznosi najveća moguća udaljenost između centroida i medoida grupe?**

A $\frac{3\sqrt{2}}{2}$ B 1 C $\frac{4}{3}$ D $\frac{\sqrt{13}}{3}$

19.2 (N) Raspolažemo sljedećim neoznačenim skupom primjera:

$$\mathcal{D} = \{(\mathbf{x}^{(i)})\}_i = \{(1,1),(1,2),(2,2),(2,3),(3,3)\}$$

Primjere grupiramo algoritmom K-sredina sa $K = 2$ grupe. Za početna središta odabrali smo primjere $\mathbf{x}^{(2)} = (1,2)$ i $\mathbf{x}^{(5)} = (3,3)$. Provedite prvu iteraciju algoritma K-sredina. **Koliko iznosi vrijednost kriterijske funkcije J nakon ažuriranja centroida?**

A 4.25 B 6.66 C 3.00 D 1.83

20. Grupiranje II

20.1 (N) Algoritmom HAC grupiramo riječi engleskog jezika. Neoznačeni skup podataka sastoji se od sljedećih riječi:

$$\mathcal{D} = \{\text{“water”}, \text{“waterloo”}, \text{“moon”}, \text{“air”}\}$$

Kao mjeru sličnosti između primjera koristimo jezgrenu funkciju nad znakovnim nizovima, definiranu kao $\kappa(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 \cap \mathbf{x}_2| / |\mathbf{x}_1 \cup \mathbf{x}_2|$, gdje su operacije unije i presjeka definirane nad skupovima slova od kojih se riječi sastoje. Npr., $\kappa(\text{“water”}, \text{“waterloo”}) = 5/7 = 0.714$. Provedite prve dvije iteracije grupiranja algoritmom HAC uz prosječno povezivanje. **Na kojoj se razini sličnosti spajaju grupe u drugoj iteraciji algoritma HAC?**

A 0.292 B 0.535 C 0.354 D 0.583

20.2 (P) Za grupiranje skupa primjera $\mathcal{D}$ koristimo algoritam GMM. Koristimo nekoliko varijanti tog modela:

$\mathcal{H}_1$: Model sa $K = 50$ središta inicijaliziranima algoritmom k-sredina
$\mathcal{H}_2$: Model sa $K = 50$ središta inicijaliziranima algoritmom k-sredina i dijeljenom kovarijacijskom matricom
$\mathcal{H}_3$: Model sa $K = 50$ slučajno inicijaliziranim središtima i dijeljenom kovarijacijskom matricom
$\mathcal{H}_4$: Model sa $K = 10$ središta inicijaliziranima algoritmom k-sredina i dijeljenom kovarijacijskom matricom

Sa svakim modelom grupiranje ponavljamo 1000 puta i zatim za svaki model crtamo graf funkcije log-izglednosti kroz iteracije EM-algoritma, uprosječen kroz svih 1000 ponavljanja. Neka je LL_α^0 prosječna log-izglednost za model $\mathcal{H}_\alpha$ na početku izvođenja EM-algoritma, a neka je LL_α^* prosječna log-izglednost za taj model na kraju izvođenja EM-algoritma. **Što možemo unaprijed zaključiti o ovim log-izglednostima?**

A $LL_3^0 \geq LL_4^0$, $LL_1^* \geq LL_3^* \geq LL_4^*$ C $LL_2^0 \leq LL_4^0$, $LL_2^* \leq LL_1^* \geq LL_3^*$
B $LL_2^0 \geq LL_4^0$, $LL_1^* \geq LL_2^* \geq LL_3^*$ D $LL_2^0 \geq LL_4^0 \geq LL_3^0$, $LL_1^* \geq LL_2^*$

21. Vrednovanje modela

21.1 (N) Logističku regresiju vrednujemo na ispitnome skupu od $N = 10$ primjera. Stvarne oznake primjera $y^{(i)}$ i vjerojatnosne predikcije klasifikatora $h(\mathbf{x}^{(i)}) = p(y = 1|\mathbf{x}^{(i)})$ na tom skupu su sljedeće:

$$\left\{(y^{(i)}, h(\mathbf{x}^{(i)}))\right\}_{i=1}^{10} = \left\{(1, 0.6), (0, 0.2), (1, 0.2), (0, 0.6), (1, 0.8), (0, 0.8), (1, 0.8), (0, 0.2), (0, 0.2), (1, 0.8)\right\}$$

Na temelju ovog uzorka želimo procijeniti mjeru AUC (površinu ispod krivulje ROC). Prisjetite se da krivulja ROC opisuje TPR (odziv) kao funkciju od FPR (stopa lažnog alarma). Skicirajte krivulju ROC, linearno interpolirajući između točaka koje odgovaraju opaženim vjerojatnosnim izlazima klasifikatora. **Koliko je ovaj klasifikator prema mjeri AUC bolji od nasumičnog klasifikatora?**

A 0 B 0.24 C 0.35 D 0.16

21.2 (P) Raspolažemo sa 1000 označenih primjera. Na tom skupu treniramo i evaluiramo algoritam SVM. Pritom razmatramo tri hiperparametra: jezgra (linearna ili RBF), regularizacijski faktor C i preciznost RBF jezgre γ. Posljednja dva hiperparametra optimiramo rešetkastim pretraživanjem u rasponima $C \in \{2^{-15}, 2^{-14}, \ldots, 2^{15}\}$ i $\gamma \in \{2^{-5}, 2^{-4}, \ldots, 2^{5}\}$. Naravno, ako ne koristimo RBF-jezgru, onda hiperparametar γ ne optimiramo. Za treniranje i evaluaciju modela koristimo ugniježđenu unakrsnu provjeru s 5 ponavljanja u vanjskoj petlji i 3 ponavljanja u unutarnjoj petlji. **Koliko će puta svaki primjer biti iskorišten za treniranje modela?**

A 5585 B 2980 C 4096 D 2820