Statističko učenje 23./24.
Četvrta domaća zadaća

Teorijski dio

Zadatak 1 (LDA i QDA)

Na Slici 1 prikazani su podaci $\{(x^{(i)}, z_i) : i = 1, \dots, n\}$ pri čemu je $n = 200$, te su z i jedina kovarijata neprekidne varijable. Možete pretpostaviti da su podaci u tri "klastera" genererirani približno iz uniformne razdiobe, te da u srednjem "klasteru" ima otprilike dva puta više podataka nego u svakom od ostala dva. Problem ćemo pretvoriti u problem klasifikacije tako što ćemo definirati

$$y_i = \begin{cases} 1, & z_i \in (-\infty, 2.5] \\ 2, & z_i \in (2.5, 3.5] \\ 3, & z_i \in (3.5, \infty) \, . \end{cases}$$

Pretpostavimo da smo primijenili QDA metodu:

(a) Ugrubo procijenite $\mu_k = \mathbb{E}[X \mid Y = k]$ i $\sigma_k = \sqrt{\mathrm{Var}(X \mid Y = k)}$ za $k = 1, 2, 3$,[1] te skicirajte procijenjene gustoće za sve tri klase (na istom grafu).

(b) Kako bi QDA klasificirala $x = -5$ i $x = 7$? Objasnite kako ste došli do rezultata.

(c) Objasnite kako bi se procjene gustoća u (a) promijenile kada bi primijenili LDA metodu? Mislite li da je LDA primjerenija u ovom primjeru? Mislite li da su općenito LDA ili QDA primjerene ovdje?

Uputa: Možete koristiti i `R` za račune i/ili grafove u (a) i (b) dijelu.

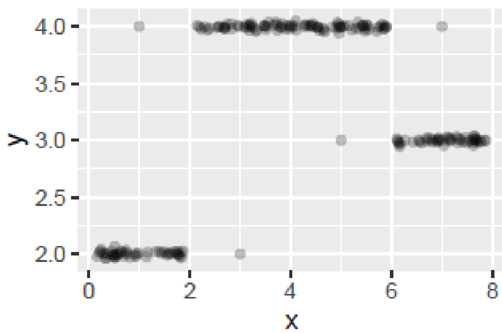
Zadatak 2 (CART)

(a) Nalazimo se u slučaju regresije uz $p = 1$ kovarijatu, te je skup za učenje veličine $n = 5$ dan s

$$\tau = \{(1, 1), (2, 1), (7, 0.5), (10, 10), (20, 11)\} \, .$$

Pretpostavimo da ćemo u klasičnom CART algoritmu za regresiju (dakle, uz L_2-gubitak) napraviti samo jedno dijeljenje – odredite procjenitelj $\hat{f} : \mathbb{R} \to \mathbb{R}$ baziran na rezultirajućem stablu.[2] Ukoliko želite, račune možete provesti i u `R`-u. Ponovite sve, ali ako prvo transformiramo kovarijate, tj. gledamo skup za učenje $\tau' = \{(\log(x^{(i)}), y_i) : i = 1, \dots, n\}$. Trebate li zapravo išta dodatno računati?

[1] Pogledajte koliko je standardna devijacija uniformne razdiobe na $[a, b]$.
[2] Stablo sa samo dva lista nekad se naziva *panj*, engl. *stump*.



Slika 1: Podaci za teorijski Zadatak 1.

(b) Nalazimo se u slučaju klasifikacije s K klasa, tj. $Y \in S = \{0, 1, \ldots, K-1\}$, te s $p \in \mathbb{N}$ kovarijata. Pretpostavimo da je zadano stablo T te neka je za svaki čvor $t \in T$, $p_k(t) \in [0,1]$ postotak $x^{(i)}$-eva iz t za koje je $y_i = k$, tj.

$$p_k(t) = \frac{\sum_{i=1}^{n} 1_{\{x^{(i)} \in t,\, y_i = k\}}}{\sum_{i=1}^{n} 1_{\{x^{(i)} \in t\}}} =: \frac{n_k(t)}{n(t)}\,,\; k \in S\,.$$

Promatramo randomiziran klasifikator koji svakom $x \in t$ pridružuje klasu $\hat{Y}(x)$ koja je slučajna varijabla s distribucijom $\mathbb{P}(\hat{Y}(x) = k) = p_k(t)$, $k \in S$. Pokažite da je Ginijeva mjera nečistoće čvora t (oznaka $i(t)$) upravo greška ovakvog klasifikatora na $\tau \cap t$, tj. da vrijedi

$$i(t) := \sum_{k \in S} p_k(t)(1 - p_k(t)) = \mathbb{E}\Big[\frac{1}{n(t)} \sum_{x^{(i)} \in t} 1_{\{y_i \neq \hat{Y}(x^{(i)})\}}\Big]\,.$$

Praktični dio

Zadatak 1

Cilj zadatka je ilustritati kako funkcionira Gradient boosting (GB) algoritam u slučaju binarne klasifikacije uz Bernoullijev gubitak, te koliko dobro procjenjuje vjerojatnosti (a ne samo klase).

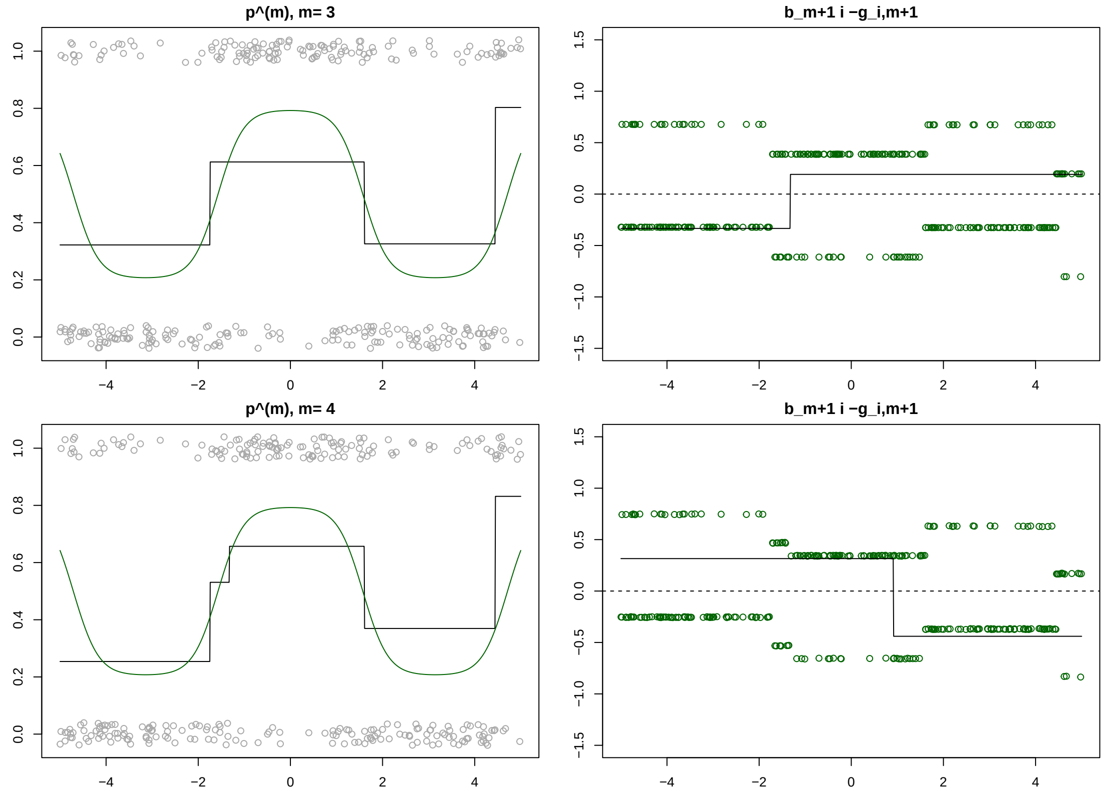
Distribucija para (X, Y)**:** Imamo samo jednu kovarijatu $X := X_1$ koja ima uniformnu razdiobu na $[-5, 5]$, te neka je $Z := \cos(X) + \epsilon$ pri čemu je $\epsilon \sim N(0, \sigma^2)$ za $\sigma = 1/5$ (funkcija `rnorm` prima σ, a ne σ^2). Zatim stavimo $Y' := 1_{\{Z \geq 0\}} \in \{0, 1\}$ te konačno definiramo da je odziv $Y \in \{0, 1\}$ s vjerojatnosti $1 - q$ jednak Y', a s vjerojatnosti q jednak $1 - Y'$ (tj. zamijenimo klasu) za $q = 0.1$.

(a) Generirajte uzorak duljine $n = 300$ iz (X, Y) (koristite `set.seed` kako biste mogli reproducirati rezultate) te grafički prikažite podatke. Napišite formulom kako izgleda regresijska funkcija $p^*(x) = \mathbb{P}(Y = 1 \mid X = x)$ te dodajte i njen graf. (*Uputa:* Radi boljeg grafičkog prikaza podataka $(x^{(i)}, y_i)$ koristite npr. naredbu `jitter`.)

(b) Generirajte niz funkcija $f^{(m)}, m = 0, 1, \dots, M$ za $M = 200$, koristeći GB algoritam uz Bernoullijevu funkciju gubitka uz `shrinkage`, `interaction.depth` i `bag.fraction` sve jednake 1. Prisjetimo se, u tom slučaju za svaki $m = 1, \dots, M$, $f^{(m)} = f^{(m-1)} + b_m$, pri čemu je b_m regresijsko stablo s dva lista koje prilagođavamo pseudo-rezidualima $-g_{i,m}$, $i = 1, \dots, n$, a procjena za $p^*(x)$ je $p^{(m)}(x) := \text{logit}(f^{(m)}(x)) = (1 + \exp\{-f^{(m)}(x)\})^{-1}$ (vidi predavanja).

(c) Za razne vrijednosti broja iteracija $m = 0, 1 \dots, M$, prikažite usporedno na dva grafa (i) podatke $(x^{(i)}, y_i)$ zajedno s grafom funkcija $p^{(m)}$ i p^*, te (ii) pseudo-reziduale $(x^{(i)}, -g_{i,m+1})$ zajedno s grafom funkcije b_{m+1}; vidi Sliku 2 dolje kao primjer. Koliki broj iteracija se čini optimalan za procjenu funkcije p^*, tj. kada se algoritam počinje previše prilagođavati podaciima? Zašto se to događa? (*Uputa:* Za graf u (ii) možete npr. koristiti funkciju `predict.gbm` uz opciju `single.tree`=TRUE. Za računanje inicijalne procjene $f^{(0)}$, odnosno $p^{(0)}$ vidi ovdje.)

(d) Generirajte nezavisan testni skup veličine $N = 5000$ te za svaki broj iteracija $m = 1, \dots, M$ procijenite testnu grešku (s obzirom na 0-1 gubitak) klasifikatora $\hat{f}^{(m)}(x) := 1_{\{f^{(m)}(x) \geq 0\}}$ iz GB algoritma. Nadalje, procijenite testnu grešku Bayesovog klasifikatora $x \mapsto 1_{\{p^*(x) \geq 1/2\}}$. Sve prikažite grafički i komentirajte rezulate. Jesu li zaključci slični kao u (c) dijelu?



Slika 2: Ilustracija za Zadatak 1.

Zadatak 2

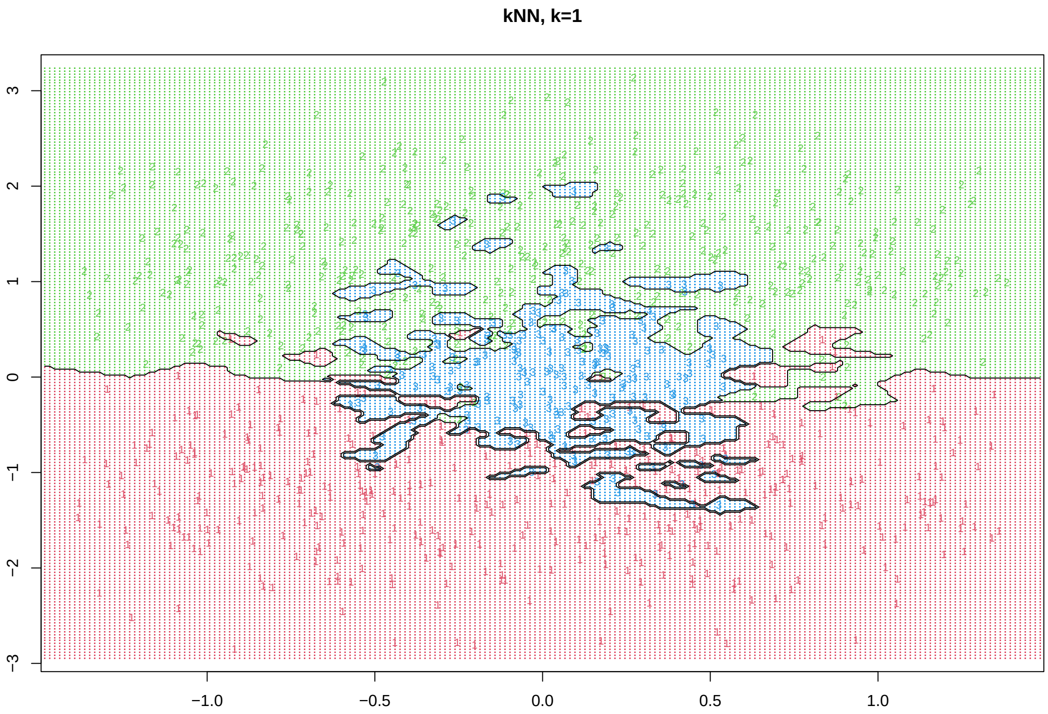
Nalazimo se u problemu klasifikacije s tri klase $Y \in \{1, 2, 3\}$ i dvije kovarijate $X = (X_1, X_2)$, a cilj zadatka je na konkretnom uzorku vizualizirati granice odluke za različite klasifikatore. Skup za učenje `data` duljine $n = 1000$ učitaje pomoću naredbe `load("zad2.Rda")`, a prikazati ga možete koristeći naredbe:

```
plot(data$x1,data$x2, pch = as.character(data$y),
     col= as.integer(data$y)+1, cex=0.5,
     xlab = "", ylab = "")
```

Za svaku od klasifikacijskih metoda koju smo radili (multinomijalna logistička regresija, LDA, QDA, naivni Bayes, kNN, CART, bagging, slučajne šume) zajedno s podacima prikažite i njihove granice odluke.

Upute:

- Jedan način kako možete riješiti zadatak je da izračunate predikcije svake od metoda na fiksnoj gustoj ekvidistantnoj mreži (npr. 200×200) područja u $\mathbb{R}^2$ koji sadrži sve $x^{(i)}$-eve i zatim te predikcije prikažete u odgovarajućoj boji koristeći npr. `pch="."`, te na kraju dodate i granice odluke koristeći funkciju `contour` (kao `levels` koristite vektor $(1.5, 2.5)$); vidi Sliku 3 kao primjer. Kod možete skratiti tako da napišete generičku funkciju koja će za dane predikcije na gustoj mreži i ime metode crtati graf poput onoga na Slici 3.
- U slučaju kNN metode, koristite $k = 1, 20, 200$.
- Za CART, bagging i slučajne šume hiperparametre izaberite proizvoljno.
- Multinomijalna log. regresija implementirana je npr. u paketu `nnet` pod funkcijom `multinom`.



Slika 3: Ilustracija za Zadatak 3.