

5. Klasifikacija primjenom logističke regresije.

5.1 Cilj Vježbe

Upoznati se s problemom klasifikacije te načinom njegova rješavanja pomoću logističke regresije.
Upoznati se s načinom vrednovanja klasifikacijskih modela.

5.2 Teorijska pozadina

5.2.1 Nadzirano učenje. Klasifikacijski problem

U ovoj vježbi razmatra se problem **nadziranog učenja** (engl. supervised learning) gdje je cilj odrediti nepoznatu funkcionalnu ovisnost između m ulaznih veličina $X = [x_1, x_2, \dots, x_m]$ i izlazne veličine y na temelju podatkovnih primjera. Podatkovni primjeri mogu se predstaviti kao parovi koji se sastoje od vektora ulaznih veličina i odgovarajućih vrijednosti izlazne veličine. Stoga, i -ti podatkovni primjer se može prikazati kao uređeni par $(x^{(i)}, y^{(i)})$ pri čemu je vektor ulaznih jednak:

$$x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_m^{(i)}]^\top. \quad (5.1)$$

Podatkovni skup koji se sastoji od n raspoloživih primjera $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})$ može se zapisati u matričnom obliku:

$$X = \begin{bmatrix} x_1^{(1)}, x_2^{(1)}, \dots, x_m^{(1)} \\ x_1^{(2)}, x_2^{(2)}, \dots, x_m^{(2)} \\ \vdots \\ x_1^{(n)}, x_2^{(n)}, \dots, x_m^{(n)} \end{bmatrix}, y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} \quad (5.2)$$

U ovoj vježbi promatra se slučaj kada je izlazna veličina y diskretna ili nebrojčana veličina, tj. razmatra se problem **klasifikacije**. Velik je broj primjera klasifikacije u praksi, npr. klasifikacija rukom pisanih brojeva, odvajanje neželjene pošte (spam), i sl.

Ako izlazna veličina ima samo dvije moguće vrijednosti (klase), npr. $y^{(i)} \in \{0, 1\}$, tada se problem naziva **binarna klasifikacija** (engl. *binary classification*). Primjeri koji imaju $y^{(i)} = 1$ nazivaju se **pozitivni** primjeri, dok se primjeri koji imaju $y^{(i)} = 0$ nazivaju **negativni** primjeri.

U slučaju kada izlazna veličina može poprimiti više od dvije vrijednosti, tada se problem naziva **višeklasna klasifikacija** (engl. *multiclass classification*). U slučaju višeklasne klasifikacije najčešće se koristi 1-od-K označavanje gdje je K broj klasa. U tom se slučaju vrijednost primjera izlazne veličine $y^{(i)}$ kodira u vektor $y^{(i)}$ koji ima K mogućih vrijednosti, ovisno kojoj klasi primjer pripada:

$$y^{(i)} \in \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\}. \quad (5.3)$$

5.2.2 Binarna klasifikacija pomoću modela logističke regresije

Model logističke regresije je jedan od osnovnih algoritama za klasifikaciju. U slučaju binarne klasifikacije (kada je $y^{(i)} \in \{0, 1\}$) model logističke regresije je oblika:

$$h(x; \theta) = g(\theta^\top x) = \frac{1}{1 + e^{-\theta^\top x}} \quad (5.4)$$

pri čemu je θ vektor parametara, a g je logistička funkcija koja je prikazana na slici 5.1. Uočite kako je izlaz modela 5.4 ograničen na interval $(0, 1)$ i on se može interpretirati kao vjerojatnost da je primjer x označen s "1":

$$p(y = 1 | x) = h(x; \theta) \quad (5.5)$$

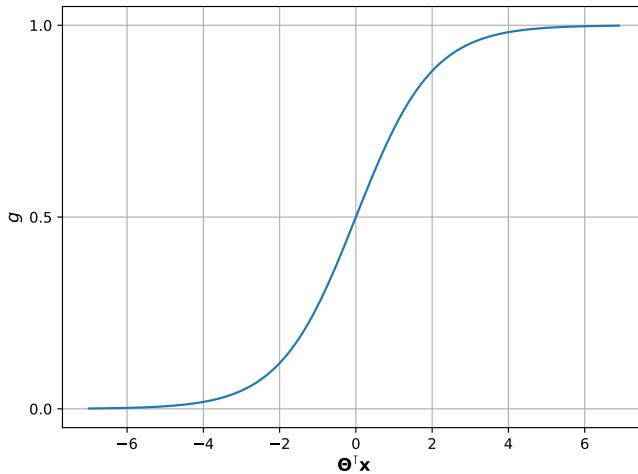
Stoga, granica između dviju klasa dobiva se za $h(x) = 0.5$ i to je hiperravnina u ulaznom prostoru za koju vrijedi:

$$\theta^\top x = 0. \quad (5.6)$$

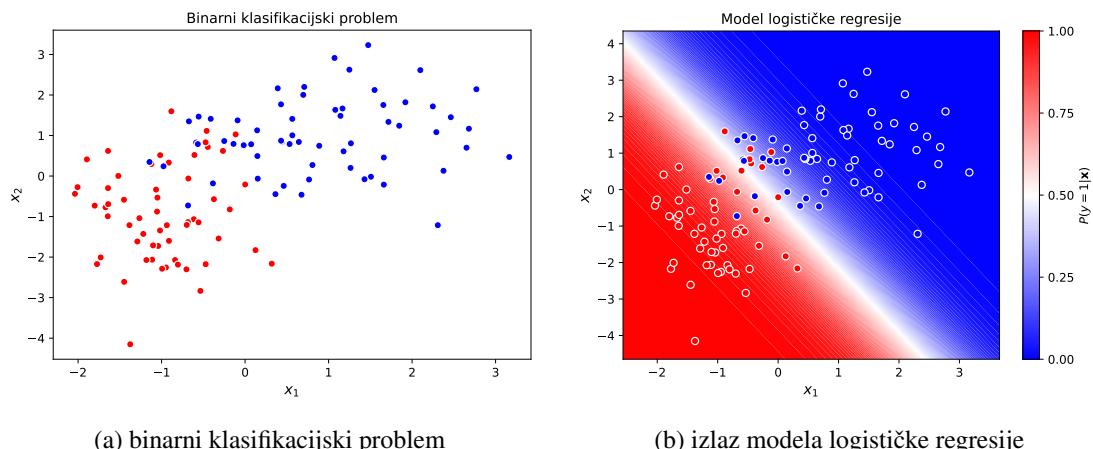
Ovu hiperravninu nazivamo granica odluke. Slika 5.2a prikazuje primjer binarnog klasifikacijskog problema s dvije ulazne veličine ($m = 2$) i sto dvadeset podatkovnih primjera ($n = 120$). Slika 5.2b prikazuje izlaz modela logističke regresije 5.4 u obliku boje pozadine. Primijetite kako je granica odluke pravac prikazan bijelom bojom. Što su primjeri x „dalje“ od granice odluke, to je vrijednost modela 5.4 „bliže“ vrijednosti 0 odnosno vrijednosti 1 ovisno s koje strane se nalazi primjer.

U konačnici klasifikacijski model treba dodijeliti svakom podatkovnom primjeru ili klasu „0“ ili klasu „1“, ovisno s koje granice odluke se nalazi podatkovni primjer:

$$\hat{y}(x) = \begin{cases} 1, & \text{ako je } \theta^\top x \geq 0 \\ 0, & \text{ako je } \theta^\top x < 0 \end{cases} \quad (5.7)$$



Slika 5.1: Logistička funkcija (sigmoidna funkcija)



Slika 5.2: Binarna klasifikacija pomoću modela logističke regresije

Vrijednosti parametara modela logističke regresije 5.4 određuju se na minimizacijom kriterijske funkcije uz dane podatke za učenje. Do kriterijske funkcije moguće je doći metodom maksimalnevjerojatnosti uz pretpostavku da su podaci za učenje nezavisni i jednoliko distribuirani. Kriterijska funkcija u tom slučaju glasi:

$$J(\theta) = \frac{-1}{n} \sum_{i=1}^n \{y^{(i)} \ln h(x^{(i)}; \theta) + (1 - y^{(i)}) \ln(1 - h(x^{(i)}; \theta))\}, \quad \theta^* = \operatorname{argmin}(J(\theta)) \quad (5.8)$$

Rješenje ovog problema ne postoji u zatvorenoј форми па се moraju koristiti iterativni numeričки поступци за optimizaciju.

5.2.3 Višeklasna klasifikacija pomoću modela logističke regresije

Model logističke regresije se može koristiti i za rješavanje problema višeklasne klasifikacije. Neka je broj klasa jednak K . Moguća su dva pristupa:

1. Izgradnja više binarnih klasifikatora. Pri tome se mogu koristiti dvije strategije:
 - Jedan naspram jedan (engl. *One-vs-One* – OvO). U ovom slučaju izrađuje se $K(K - 1)/2$ binarnih klasifikatora pri čemu svaki klasifikator modelira odnos između primjera koji pripadaju dvjema različitim klasama. Za dani testni primjer svaki binarni klasifikator daje predikciju klase prema 5.7 te je konačan rezultat klasa s najviše predikcija.
 - Jedan naspram ostalih (engl. *One-vs-Rest* – OvR). U ovom slučaju izrađuje se K binarnih klasifikatora. Svaki klasifikator modelira odnos između primjera jedne klase u odnosu na sve ostale primjere koji pripadaju ostalim klasama. Za dani testni primjer predikcija klase određuje se prema klasifikatoru koji ima maksimalnu vrijednost 5.4.
2. Multinomijalna logistička regresija (softmax regresija) koja kao predikciju izravno daje vjerojatnost pojedine klase:

$$h(x; \theta) = \begin{bmatrix} p(y = 1 \vee x) \\ p(y = 2 \vee x) \\ \vdots \\ p(y = K \vee x) \end{bmatrix} = \frac{1}{\sum_{k=1}^K \exp(\theta_k^\top x)} \begin{bmatrix} \exp(\theta_1^\top x) \\ \exp(\theta_2^\top x) \\ \vdots \\ \exp(\theta_K^\top x) \end{bmatrix}. \quad (5.9)$$

pri čemu su parametri modela pohranjeni u matricu θ . Ova matrica je dimenzija $m \times K$ i sadrži vektore parametara za svaku klasu u svojim stupcima:

$$\theta = [\theta_1 \theta_2 \dots \theta_K] \quad (5.10)$$

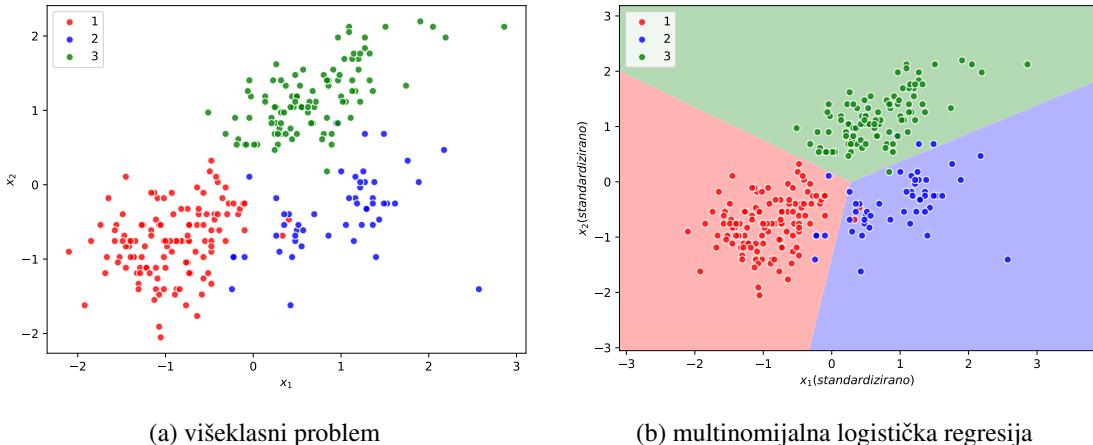
Prema 5.9, izlaz multinomijalne logističke regresije za dani primjer \mathbf{x} je vektor od K elemenata pri čemu je svaki element manji od 1, a suma svih elemenata jednaka je 1 uslijed softmax aktivacijske funkcije. Vrijednosti parametara modela multinomijalne logističke regresije 5.9 određuju se minimizacijom kriterijske funkcije uz dane podatke za učenje:

$$J(\theta) = \frac{-1}{n} \sum_{i=1}^n \sum_{k=1}^K y_k^{(i)} \ln(h_k(x^{(i)}; \theta)) \quad (5.11)$$

Na slici 5.3a prikazan je primjer višeklasnog klasifikacijskog problema s dvije ulazne veličine. Na slici 5.3b prikazan je rezultat koji se dobije pomoću multinomijalne logističke regresije za dani problem pri čemu su ulazne veličine standardizirane. Boja pozadine definira kojoj klasi pripada pojedino područje ulaznog prostora prema izgrađenom modelu.

5.2.4 Vrednovanje klasifikacijskih modela

Testiranje predikcijskih sposobnosti odnosno sposobnosti generalizacije izgrađenog klasifikacijskog modela (klasifikatora) potrebno je provesti na zasebnom skupu podataka koji se naziva skup za testiranje. U slučaju binarne klasifikacije podatkovni primjer $(x^{(i)}, y^{(i)})$ može biti pozitivan $y^{(i)} = 1$ ili negativan $y^{(i)} = 0$. Stoga, moguća su četiri slučaja prilikom uspoređivanja rezultata koje daje klasifikator 5.7 sa stvarnom vrijednosti:



Slika 5.3: Višeklasna klasifikacija pomoću modela logističke regresije

- istinito pozitivan rezultat (engl. *true positive* – TP) – pozitivni primjer kojeg je klasifikator klasificirao kao pozitivan primjer,
- istinito negativan rezultat (eng. *true negative* – TN) – negativni primjer kojeg je klasifikator klasificirao kao negativan primjer,
- lažno pozitivan rezultat (eng. *false positive* – FP) – negativan primjer kojeg je klasifikator klasificirao kao pozitivan primjer, i
- lažno negativan rezultat (eng. *false negative* – FN) – pozitivan primjer kojeg je klasifikator klasificirao kao negativan primjer.

Očito da su TP i TN primjeri koje klasifikator točno klasificira dok u su FP i FN primjeri koje klasifikator pogrešno klasificira. Vrlo je korisno prikazati ove rezultate u obliku matrice zabune (engl. *confusion matrix*). Ova matrica pokazuje ukupan broj rezultata za sva četiri slučaja i dana je u tablici 5.1.

Tablica 5.1: Matrica zabune

Matrica zabune		Predviđeno klasifikatorom	
		Klase „1“	Klase „0“
Stvarna klasa	Klase „1“	TP	FN
	Klase „0“	FP	TN

Za vrednovanje klasifikacijskog modela se osim matrice zabune mogu koristiti sljedeće metrike:

- točnost (engl. *accuracy*) - predstavlja udio točno klasificiranih primjera:

$$točnost = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.12)$$

- preciznost (engl. *precision*) - predstavlja udio točno klasificiranih primjera u skupu koje model klasificira kao pozitivne primjere:

$$preciznost = \frac{TP}{TP + FP} \quad (5.13)$$

- odziv (engl. *recall*) - predstavlja udio točno klasificiranih primjera u skupu pozitivnih primjera:

$$odziv = \frac{TP}{TP + FN} \quad (5.14)$$

- F1 mjera (engl. *F1-score*) - predstavlja kombinaciju preciznosti i odziva:

$$F1 = 2 \frac{preciznost \cdot odziv}{preciznost + odziv} \quad (5.15)$$

Iznosi ovih metrika su u intervalu od 0 do 1 pri čemu je 1 najbolja vrijednost.

Primjer

Izgrađen je klasifikator koji na temelju podataka o osobi (pacijentu) zaključuje ima li osoba određenu bolest ili ne. Testni skup sastoji se od 20 uzoraka u kojima je u 12 slučajeva bila prisutna bolest, a u preostalih 8 ne. Rezultat klasifikacije ovog skupa se može prikazati matricom zabune.

Matrica zabune		Predviđeno klasifikatorom		$točnost = (9+7)/(20) = 80\%$
		DA	NE	
Stvarna klasa	DA	9	3	$preciznost = 9/10 = 90\%$
	NE	1	7	$odziv = 9/12 = 75\%$

Zbroj brojeva u tablici je 20. Vidljivo je kako je od 12 bolesnih osoba model točno klasificirao njih 9 dok je 3 označio kao zdrave. U slučaju zdravih osoba, 7 je točno klasificirao dok je jednu osobu klasificirao kao bolesnu.

5.3 Klasifikacija pomoću scikit-learn biblioteke

Model logističke regresije u scikit-learn biblioteci implementiran je u obliku klase:

```
class sklearn.linear_model.LogisticRegression (penalty='l2', *,
dual=False, tol=0.0001, C=1.0, fit_intercept=True,
intercept_scaling=1, class_weight=None, random_state=None,
solver='lbfgs', max_iter=100, multi_class='auto', verbose=0,
warm_start=False, n_jobs=None, l1_ratio=None)
```

Najvažniji parametri su:

- C – konstanta koja definira jačinu regularizacije* (manja vrijednost jača regularizacija)
decimalna vrijednost, default=1.0
- tol – tolerancija za kriterij zaustavljanja
- solver – algoritam koji se koristi prilikom optimizacije
{'lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga'},
default='lbfgs'
- max_iter – broj iteracija tijekom optimizacije
cjelobrojna vrijednost, default=100
- multi_class – način rješavanja problema višeklasne klasifikacije (jedan naspram ostalih ili multinomijalna logistička regresija)
{'auto', 'ovr', 'multinomial'}, default='auto'

Najvažnije metode ove klase su:

- .fit(X,y) za procjenu parametara modela na temelju podataka za učenje,
- .predict(X) za izračunavanje izlaza modela na temelju ulaznih vrijednosti ulaznih veličina,
- .predict_proba(X) za izračunavanje vjerojatnosti klase na temelju ulaznih vrijednosti ulaznih veličina

Primjer 5.1 prikazuje isječak koda koji inicijalizira model logističke regresije te se zatim procjenjuju parametri modela na temelju podataka za učenje. Izgrađeni model se onda koristi za predikciju izlazne veličine na skupu podataka za testiranje.

■ Primjer 5.1

```
from sklearn.linear_model import LogisticRegression

# inicijalizacija i ucenje modela logistickog regresije
LogRegression_model = LogisticRegression()
LogRegression_model.fit(X_train, y_train)

# predikcija na skupu podataka za testiranje
y_test_p = LogRegression_model.predict(X_test)
```

U scikit-learn biblioteci dostupne su funkcije za evaluaciju klasifikacijskih modela. U primjeru 5.2 koristi se točnost klasifikacije i matrica zabune za evaluaciju izgrađenog modela. Izračunata matrica zabune prikazuje se u obliku slike. Pomoću funkcije `classification_report` moguće je izračunati četiri glavne metrike (točnost, preciznost, odziv i F1 mjeru).

■ Primjer 5.2

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# stvarna vrijednost izlazne velicine i predikcija
y_true = np.array([1, 1, 1, 0, 1, 0, 1, 0, 1])
y_pred = np.array([0, 1, 1, 1, 1, 0, 1, 0, 0])

# točnost
print("Točnost: ", accuracy_score(y_true, y_pred))

# matrica zabune
cm = confusion_matrix(y_true, y_pred)
print("Matrica zabune: ", cm)
disp = ConfusionMatrixDisplay(confusion_matrix(y_true, y_pred))
disp.plot()
plt.show()

# report
print(classification_report(y_true, y_pred))
```

5.4 Priprema za vježbu

1. Proučite poglavlje 5.2.

2. Po potrebi dodatno proučite dokumentaciju:
 - a. Model logističke regresije u scikit-learn biblioteci
 - b. Metrike za evaluaciju klasifikacijskih modela u scikit-learn biblioteci

5.5 Rad na vježbi

1. Isprobajte Python primjere iz poglavlja 5.3 u Visual Studio Code IDE.
2. Riješite dane zadatke.

Zadatak 5.5.1 Skripta `zadatak_1.py` generira umjetni binarni klasifikacijski problem s dvije ulazne veličine. Podaci su podijeljeni na skup za učenje i skup za testiranje modela.

- a) Prikažite podatke za učenje u $x_1 - x_2$ ravnini matplotlib biblioteke pri čemu podatke obojite s obzirom na klasu. Prikažite i podatke iz skupa za testiranje, ali za njih koristite drugi marker (npr. 'x'). Koristite funkciju `scatter` koja osim podataka prima i parametre c i $cmap$ kojima je moguće definirati boju svake klase.
- b) Izgradite model logističke regresije pomoću scikit-learn biblioteke na temelju skupa podataka za učenje.
- c) Pronađite u atributima izgrađenog modela parametre modela. Prikažite granicu odluke naučenog modela u ravnini $x_1 - x_2$ zajedno s podacima za učenje. Napomena: granica odluke u ravnini $x_1 - x_2$ definirana je kao krivulja: $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$.
- d) Provedite klasifikaciju skupa podataka za testiranje pomoću izgrađenog modela logističke regresije. Izračunajte i prikažite matricu zabune na testnim podacima. Izračunate točnost, preciznost i odziv na skupu podataka za testiranje.
- e) Prikažite skup za testiranje u ravnini $x_1 - x_2$. Zelenom bojom označite dobro klasificirane primjere dok pogrešno klasificirane primjere označite crnom bojom.

Zadatak 5.5.2 Skripta `zadatak_2.py` učitava podatkovni skup Palmer Penguins [1]. Ovaj podatkovni skup sadrži mjerenja provedena na tri različite vrste pingvina ('Adelie', 'Chinstrap', 'Gentoo') na tri različita otoka u području Palmer Station, Antarktika. Vrsta pingvina odabrana je kao izlazna veličina i pri tome su klase označene s cjelobrojnim vrijednostima 0, 1 i 2. Ulazne veličine su duljina kljuna ('bill_length_mm') i duljina peraje u mm ('flipper_length_mm'). Za vizualizaciju podatkovnih primjera i granice odluke u skripti je dostupna funkcija `plot_decision_region`.

- a) Pomoću stupčastog dijagrama prikažite koliko primjera postoji za svaku klasu (vrstu pingvina) u skupu podataka za učenje i skupu podataka za testiranje. Koristite numpy funkciju `unique`.
- b) Izgradite model logističke regresije pomoću scikit-learn biblioteke na temelju skupa podataka za učenje.
- c) Pronađite u atributima izgrađenog modela parametre modela. Koja je razlika u odnosu na binarni klasifikacijski problem iz prvog zadatka?
- d) Pozovite funkciju `plot_decision_region` pri čemu joj predajte podatke za učenje i izgrađeni model logističke regresije. Kako komentirate dobivene rezultate?
- e) Provedite klasifikaciju skupa podataka za testiranje pomoću izgrađenog modela logističke regresije. Izračunajte i prikažite matricu zabune na testnim podacima. Izračunajte točnost. Pomoću `classification_report` funkcije izračunajte vrijednost četiri glavne metrike

- na skupu podataka za testiranje.
- f) Dodajte u model još ulaznih veličina. Što se događa s rezultatima klasifikacije na skupu podataka za testiranje?

5.6 Izvještaj s vježbe

Kao izvještaj s vježbe prihvaća se web link na repozitorij pod nazivom OSU_LV.

Literatura

[1] <https://github.com/allisonhorst/palmerpenguins>