Primjena logističke regresije u inženjerstvu

Gašljević, Raul Ivan

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij strojarstva

Završni rad

PRIMJENA LOGISTIČKE REGRESIJE U INŽENJERSTVU

Rijeka, srpanj 2023.

Raul Ivan Gašljević
0069089360

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij strojarstva

Završni rad

PRIMJENA LOGISTIČKE REGRESIJE U INŽENJERSTVU

Mentorica: Izv. prof. dr. sc. Loredana Simčić

Komentorica: Prof. dr. sc. Nelida Črnjarić

Rijeka, srpanj 2023.

Raul Ivan Gašljević
0069089360

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 9. ožujka 2023.

Zavod: **Zavod za matematiku, fiziku i strane jezike**
Predmet: **Inženjerska statistika**
Grana: **1.01.07 primijenjena matematika i matematičko modeliranje**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Raul Ivan Gašljević (0069089360)**
Studij: Sveučilišni prijediplomski studij strojarstva

Zadatak: **Primjena logističke regresije u inženjerstvu / Application of logistic regression in engineering**

Opis zadatka:

U mnogim se inženjerskim problemima na temelju ulaznih podataka, odnosno poznatih značajki, trebaju donijeti zaključci o pripadnosti određenoj kategoriji ili pak procijeniti (predvidjeti) vjerojatnost pojave određenog ishoda. Jedan od mogućih alata koji se može koristiti za takvu svrhu je logistička regresija, koja daje vezu između predikcijskih varijabli i klasifikacijskih vjerojatnosti koju je lako interpretirati. Logistička se regresija može smatrati matematičkim, odnosno statističkim modelom u području strojnog učenja.
Zadatak ovog završnog rada je opisati model binarne logističke regresije i višestruke logističke regresije. Potrebno je objasniti postupak učenja modela s osvrtom na implementaciju u programskom jeziku Python, te značenje parametara koji se pojavljuju. Objasniti postupak procjene točnosti dobivenog modela.
Zatim je potrebno navesti moguće primjene u inženjerstvu, te za odabrane skupove podataka odrediti u programskom
jeziku Python model logističke regresije i procijeniti njezinu točnost.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Raul Ivan Gašljević

Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor:

Loredana Simčić

Doc. dr. sc. Loredana Simčić

Prof. dr. sc. Nelida Črnjarić-Žic (komentor)

Predsjednik povjerenstva za završni ispit:

Izv. prof. dr. sc. Samir Žic

IZJAVA

Izjavljujem da sam završni rad „Primjena logističke regresije u inženjerstvu“ izradio samostalno, uz znanje stečeno tijekom studiranja, te uz korištenje navedene literature i pod stručnim vodstvom mentorice izv. prof. dr. sc. Loredane Simčić i komentorice prof. dr. sc. Nelide Črnjarić.

Rijeka, srpanj 2023.

_____

Raul Ivan Gašljević
0069089360

ZAHVALA

Zahvaljujem se mentorici izv. prof. dr. sc. Loredani Simčić i komentorici prof. dr. sc. Nelidi Črnjarić na savjetima i utrošenom vremenu tijekom izrade ovog rada. Zahvaljem se i svim profesorima na stečenom znanju tijekom studiranja na Tehničkom fakultetu u Rijeci. Ponajviše, zahvaljujem se obitelji i prijateljima na omogućenom studiranju i potpori tijekom studija.

Sadržaj

1. UVOD

U mnogim se inženjerskim problemima na temelju ulaznih podataka, odnosno poznatih značajki, trebaju donijeti zaključci o pripadnosti određenoj kategoriji ili pak procijeniti (predvidjeti) vjerojatnost pojave određenog ishoda. Jedan od mogućih alata koji se može koristiti za takvu svrhu je logistička regresija, koja daje vezu između predikcijskih varijabli i klasifikacijskih vjerojatnosti koju je lako interpretirati. Logistička se regresija može smatrati matematičkim, odnosno statističkim modelom u području strojnog učenja.

Logistička regresija predstavlja jedan od osnovnih alata u strojnom učenju za binarnu klasifikaciju, odnosno klasifikaciju u dvije kategorije. Primjena ovog modela je vrlo široka, a primjeri obuhvaćaju područja kao što su medicina, financije, inženjerstvo, ekonomija i mnoga druga područja. Ovaj model se često koristi za rješavanje problema kao što su procjena rizika, klasifikacija klijenata u financijskom sektoru ili kategorizacija klijenata u marketinške svrhe.

Jednostavnost i lakoća korištenja su glavni razlozi zašto se ovaj algoritam često koristi u praksi. Logistička regresija je posebno učinkovita u situacijama kada se radi o binarnoj klasifikaciji, tj. kada se želi razvrstati podatke u jednu od dvije klase. Metoda je vrlo prilagodljiva i može se lako primijeniti na različite vrste podataka, bilo da se radi o malim ili velikim skupovima podataka.

Međutim, značajno je napomenuti da logistička regresija nije uvijek najbolji izbor za složenije probleme klasifikacije. U takvim je situacijama često potrebno koristiti složenije algoritme koji mogu primijeniti složenije obrasce u podacima. Tako su na primjer, kod klasifikacije slika ili govora, složeniji algoritmi kao što su neuronske mreže, konvolucijske neuronske mreže ili rekurentne neuronske mreže obično bolji izbor. Ti algoritmi mogu obraditi velike količine podataka i izvući složenije značajke iz podataka, što omogućuje precizniju klasifikaciju. Dakle, uz sve prednosti logističke regresije, postoje i neki izazovi u njezinoj primjeni. Jedan od glavnih izazova je kvaliteta podataka, jer neispravni ili nedostajući podaci mogu dovesti do netočnih predviđanja. Drugi izazov je interpretacija rezultata, jer je potrebno pažljivo analizirati dobivene rezultate i razumijeti njihovu stvarnu primjenu. U konačnici, odabir algoritma za klasifikaciju ovisi o specifičnim zahtjevima problema koji se rješava, pa je važno dobro razumjeti karakteristike različitih algoritama i njihovu primjenu u različitim situacijama kako bi se donijela najbolja odluka.

U ovom radu će se istražiti primjena logističke regresije u inženjerstvu. Cilj rada je prikazati kako se ovaj model može primijeniti u inženjerskim problemima, te kako se interpretiraju dobiveni rezultati. U prvom dijelu rada bit će objašnjeni osnovni koncepti logističke regresije, kao što su logistička funkcija i parametri modela. Nakon toga, bit će prikazane različite primjene binarne logističke regresije i višestruke logističke regresije. Objasnit će se postupak učenja modela s osvrtom na implementaciju u programskom jeziku Python, te značenje parametara koji se pojavljuju.

U sklopu programskog jezika Python koristit će se niz alata, uključujući Pandas, NumPy, Matplotlib, Scikit-learn, Statsmodels, Wntr te drugi potrebni za analizu i vizualizaciju podataka. Također, cilj ovog rada je primijeniti pristup strojnog učenja za analizu podataka i implementirati logističku regresiju pomoću Scikit-learn biblioteke. Bit će prikazani postupci za evaluaciju performansi modela i odabir optimalnih parametara. Konkretno, logistička regresija će se primijeniti

na skup podataka kako bi se predvidio određeni ishod na temelju značajki. Ovaj rad će također pokazati kako koristiti alate za vizualizaciju podataka kako bi se bolje razumjeli podaci, a isto tako će se obraditi i problemi poput testiranja modela kako bi se osigurala njegova pouzdanost i primjenjivost u stvarnim situacijama.

2. LOGISTIČKA REGRESIJA

U statistici, logistički model je statistički model kojim se modelira vjerojatnost p realizacije događaja uz pretpostavku da postoji jedna ili više nezavisnih varijabli koje utječu na promatrani događaj. Pritom se pretpostavlja da je logaritam izgleda realizacije promatranog događaja linearna kombinacija tih nezavisnih varijabli. Izgled nekog događaja definira se kao kvocijent vjerojatnosti da se taj događaj dogodi i vjerojatnosti da se ne dogodi tj. $\frac{p}{1-p}$. Logaritam izgleda (eng. *log–odds*) potom je jednak $\ln \frac{p}{1-p}$. Iz toga će slijediti da je vjerojatnost događaja opisana logističkom krivuljom, koja je sigmoidnog oblika. U regresijskoj analizi, logistička regresija odnosi se na procjenu parametara logističkog modela. Parametri te krivulje određuju se na osnovu empirijskih podataka.

Logistička regresija također se naziva binomna logistička regresija ili binarna logistička regresija, jer se njome modelira binarna (Bernoullijeva, indikatorska) slučajna varijabla Y koja poprima vrijednosti "0" ili "1". Vrijednost logističke funkcije predstavljat će procjenu vjerojatnosti događaja $p = P(Y = 1)$, pa se zapravo model može koristiti za klasifikaciju u dvije klase (binarnu klasifikaciju). Ako postoje više od dvije klase tada se koristi poopćenje logističke regresije, koje se naziva višekategorijska logistička regresija.

2.1. Model logističke regresije

Standardna logistička funkcija je sigmoidna funkcija σ: $\mathbf{R} \to$ (0,1) definirana s:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}. \tag{2.1}$$

2.1.1. Jednostavna logistička regresija

Kod modela jednostavne logističke regresije pretpostavlja se da vjerojatnost realizacije nekog promatranog događaja ovisi o vrijednosti nezavisne varijable X. Precizinije rečeno, pretpostavlja se da su za neke vrijednosti nezavisne varijable $x_1, \ldots, x_N$ poznate uvjetne vjerojatnosti

$$p(x_i) = P(Y = 1|x_i), \quad i = 1, \ldots, N. \tag{2.2}$$

Zadatak logističke regresije je određivanje logističke funkcije oblika

$$p(x; \beta_0, \beta_1) = \sigma(\beta_0 + \beta_1 x),$$

odnosno parametara β_0 i β_1 u funkciji:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}, \tag{2.3}$$

na način da funkcija $p(x)$ u točkama $x_i,\ i = 1, \ldots, N$ "najbolje" opisuje empirijske podatke. Pritom se u (2.3) zbog jednostavnosti na lijevoj strani zanemarilo pisanje ovisnosti funkcije p o parametrima β_0 i β_1. Može se uočiti da se ovdje pretpostavilo da je varijabla t logističke funkcije zapravo linearna funkcija varijable x, tj. $t = \beta_0 + \beta_1 x$.

Iz izraza (2.3) lako se dobiva

$$\ln \frac{p(x)}{1 - p(x)} = \beta_0 + \beta_1 x, \tag{2.4}$$

pa s obzirom da je prije spomenuto da je logaritam izgleda događaja s vjerojatnošću p jednak

$$\ln \frac{p}{1 - p},$$

iz (2.4) slijedi da u logističkom modelu vrijedi linearna veza između nezavisne varijable i logaritma izgleda. Izgled događaja za tako definirani model jednak je

$$\frac{p(x)}{1 - p(x)} = \mathrm{e}^{\beta_0 + \beta_1 x}. \tag{2.5}$$

Iz prethodnih se izraza može zaključiti da parametar β_1 u modelu određuje koliko se promjeni mjera izgleda realizacije događaja $(Y = 1)$ kada se nezavisna varijabla x poveća za 1.

2.1.2. Složena (višestruka) logistička regresija

U slučaju kada ishod (zavisna varijabla) ovisi o više nezavisnih varijabli $X_1, \ldots, X_n$, govori se o višestrukoj ili složenoj logističkoj regresiji. Kao i kod jednostavne logističke regresije, u modelu se pretpostavlja da je varijabla t u logističkoj funkciji linearno povezana s nezavisnim (prediktorskim) varijablama tj. $t = \beta_0 + \beta_1 x_1 + \cdots, \beta_n x_n$. Dobiveni logistički model ima oblik:

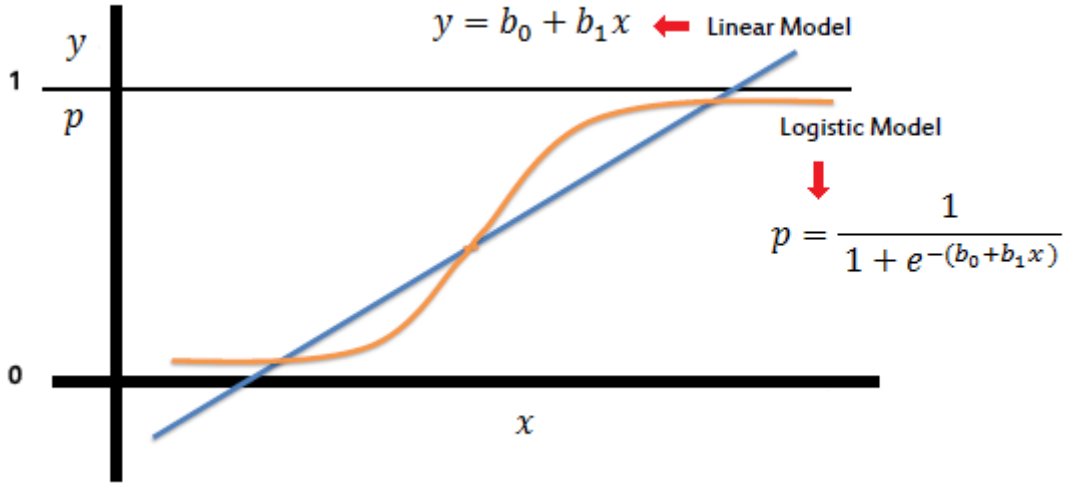
$$p(x_1, ..., x_m) = \frac{1}{1 + \mathrm{e}^{-(\beta_0 + \beta_1 x_1 + ... + \beta_n x_n)}}. \tag{2.6}$$

U ovom je slučaju zadatak logističke regresije odrediti parametre $\beta_0, \beta_1, \ldots, \beta_n$ za koje će funkcija maksimalne vjerojatnosti dosegnuti maksimalnu vrijednost.

2.2. Usporedba linearne i logističke regresije

Na Slici 2.1 dana je usporedba grafičkog prikaza linearnog i logističkog modela, kao i izrazi kojima se ti modeli opisuju. Linearni model prikazan grafički predstavlja pravac, dok je graf logističke regresije, logistička ili logit krivulja koja poprima vrijednosti između 0 i 1.

Kod problema linearne regresije pretpostavlja se da postoji linearna funkcijska veza između zavisne i nezavisne varijable. Zadatak linearne regresije je odrediti linearnu funkciju koja "najbolje" opisuje dane podatke. Pritom se kod modela linearne regresije obično pretpostavlja da je zavisna varijabla kontinuirana. U slučaju kada je slučajna varijabla diskretna ili kategorijska, model linearne regresije neće biti pogodan, pa će se u tom slučaju biti pogodnije koristiti logističku

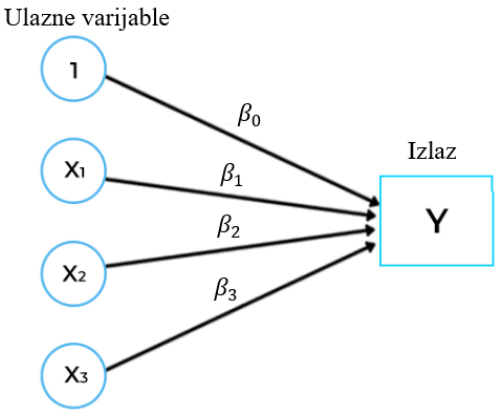


Slika 2.1. Usporedba linearnog i logističkog modela [4]

regresiju.

U najjednostavnijem slučaju binarne logističke regresije promatra se zavisna slučajna varijabla koja može imati samo dvije vrijednosti, 0 ili 1. Model logističke regresije omogućit će određivanje vjerojatnosti promatranog događaja, odnosno temeljem dobivene vjerojatnosti zaključivanje o vrijednosti promatrane slučajne varijable.

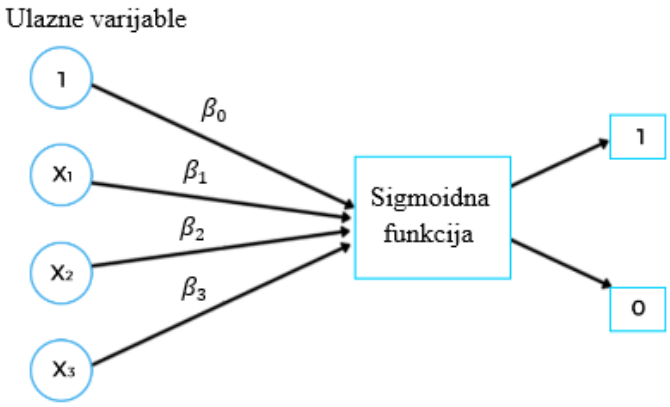
Nezavisna varijabla može biti samo jedna ili može postojati više nezavisnih slučajnih varijabli, pa će u tom slučaju u problemu s više nezavisnih varijabli (ulaznih značajki) model linearne regresije procjenjivati vrijednost zavisne varijable uzimajući u obzir linearnu kombinaciju svih ulaznih značajki. Na Slici 2.2 shematski su prikazane ulazne varijable, označene s x_1, x_2 i x_3 i izlazna varijabla Y. Parametri modela označeni su s $\beta_i,\ i = 0, 1, 2, 3$.



Slika 2.2. Ulaz i izlaz kod linearnog modela

Kod modela logističke regresije, prikazanom shematski na Slici 2.3 uzima se također linearna kombinacija ulaznih varijabli, ali se na rezultat primjenjuje logistička (sigmoidna) funkcija. Time se dobiva nelinearna veza između nezavisnih ulaznih varijabli i vjerojatnosti ishoda promatranog događaja. Temeljem te se vjerojatnosti dobiva binarni izlaz u obliku 1 ili 0 (ili pak "točno" ili

"netočno").



Slika 2.3. Ulaz i izlaz kod logističkog modela

Metode za određivanje parametara linearne, odnosno logističke regresije razlikuju se. Kod problema linearne regresije koristi se metoda najmanjih kvadrata za određivanje regresijske funkcije koja najbolje odgovara danom problemu. Prema metodi, regresijski koeficijenti trebaju biti odabrani tako da smanje zbroj kvadrata udaljenosti svake varijable odgovora na prilagođenu vrijednost. Rezultat se može odrediti direktno rješavanjem linearnog sustava. Logistička regresija pak koristi metodu najveće vjerojatnosti, gdje se regresijski koeficijenti biraju tako da maksimiziraju vjerojatnost ishoda y za dani x. Kod rješavanja problema logističke regresije dobiva se nelinerani sustav koji se rješava iterativnim numeričkim metodama koje su bitno zahtjevnije, te stoga i sporije od rješavanja lineranog sustava.

2.3. Procjena parametara modela

U nastavku će najprije biti općenito opisana metoda maksimalne vjerojatnosti. Zatim će se opisati njezina primjena na binarnoj slučajnoj varijabli, te na kraju za određivanje parametara logističke funkcije.

2.3.1. Metoda maksimalne vjerojatnosti

Metoda maksimalne vjerojatnosti (eng. *maximum likelihood method*) polazi od pretpostavke da su poznati ishodi nekog slučajnog pokusa koji se mogu opisati slučajnom varijablom Y, te da funkcija distribucije te slučajne varijable nije u potpunosti poznata, odnosno da nisu poznati svi parametre te funkcije. Zadatak metode je odrediti takve parametre da vjerojatnost pojavljivanja dobivenih ishoda bude najveća moguća.

Postupak je sljedeći. Za zadanu funkciju gustoće $f(y;\theta)$, pri čemu su $\theta = (\theta^1, \ldots, \theta^p)$ nepoznati parametri, i dobiveni uzorak $y_1, ..., y_N$ vrijednosti slučajne varijable Y, funkcija maksimalne

vjerojatnosti (eng. *maximum likelihood function*) definira se s:

$$L(y_1, ..., y_N; \theta) = f(y_1; \theta)...f(y_N; \theta) = \prod_{i=1}^{N} f(y_i; \theta) \tag{2.7}$$

Zadatak je odrediti parametre $\hat{\theta}$ za koji ta funkcija dosegne maksimalnu vrijednost tj.

$$\hat{\theta} = \arg\max_{\theta} L(y_1, ..., y_N; \theta). \tag{2.8}$$

Ovdje je funkcijom $\arg\max$ označena vrijednost argumenta u kojem funkcija dosegne maksimum. S obzirom da je logaritamska funkcija $\ln$ rastuća, logaritmiranjem funkcije $L(y_1, ..., y_N; \theta)$ dobivamo funkciju

$$l(y_1, ..., y_N; \theta) = \sum_{i=1}^{N} \ln f(y_1, ..., y_N; \theta) \tag{2.9}$$

za koju je maksimalna vrijednost dosegnuta u istoj točki tj. vrijedi

$$\hat{\theta} = \arg\max_{\theta} l(y_1, ..., y_N; \theta).$$

2.3.2. Metoda maksimalne vjerojatnosti za binarnu slučajnu varijablu

U nastavku će se metoda maksimalne vjerojatnosti primijeniti na binarnu (Bernoulijevu) slučajnu varijablu. Pretpostavlja se da je slučajna varijabla dana distribucijom

$$Y \sim Ber(p) = \begin{pmatrix} 1 & 0 \\ p & 1-p \end{pmatrix}, \tag{2.10}$$

pri čemu je p nepoznati parametar kojeg treba procijeniti temeljem empirijskih vrijednosti $y_1, \ldots, y_N$. Funkcija gustoće za razdiobu (2.10) može se zapisati na sljedeći način:

$$f(y; p) = \begin{cases} p, y = 1 \\ 1 - p, y = 0 \end{cases} = p^y(1-p)^{1-y}. \tag{2.11}$$

Funkcija maksimalne vjerojatnosti i pripadajući logaritam potom su jednaki:

$$L(y_1, ..., y_N; p) = \prod_{i=1}^{N} p^{y_i}(1-p)^{1-y_i}$$

i

$$\begin{aligned} l(y_1, ..., y_N; p) &= \sum_{i=1}^{N} y_i \log p + (1 - y_i) \log(1-p) \\ &= \big(\sum_{i=1}^{N} y_i\big) \log p + \big(N - \sum_{i=1}^{N} y_i\big) \log(1-p). \end{aligned} \tag{2.12}$$

Za određivanje maksimalne vrijednosti funkcije l, izraz (2.12) je potrebno derivirati i izjednačiti s nulom, pa se dobiva:

$$\begin{aligned}\frac{\partial l}{\partial p} &= \big(\sum_{i=1}^{N} y_i\big)\frac{1}{p} + \big(N - \sum_{i=1}^{N} y_i\big)\frac{1}{1-p}(-1) = 0 \\ &\Longrightarrow \sum_{i=1}^{N} y_i(1-p) - (N - \sum_{i=1}^{N} y_i)p = 0 \\ &\Longrightarrow \sum_{i=1}^{N} y_i - \sum_{i=1}^{N} y_i p - Np + \sum_{i=1}^{N} y_i p = 0 \\ &\Longrightarrow \hat{p} = \frac{1}{N}\sum_{i=1}^{N} y_i = \overline{y},\end{aligned}$$

gdje je s $\overline{y}$ označena aritmetička sredina uzorka $\{y_1, \ldots, y_N\}$. Dobiveni izraz $\hat{p}$ naziva se točkastim procjeniteljem parametra p prema metodi maksimalne vjerojatnosti.

2.3.3. Procjena koeficijenata logističke regresije

Opisana metodologija primijenjuje se kod određivanja parametara logističke regresije. U modelu logističke regresije se pretpostavlja da parametar p koji se pojavljuje u binarnom modelu, dodatno ovisi o vrijednostima nezavisne slučajne varijable X. Kao što je u potpoglavlju 2.1 opisano, za poznate vrijednosti nezavisne slučajne varijable $x_1, \ldots, x_N$, vrijedi:

$$P(Y = 1 \mid x_i) = p(x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}}$$

i

$$P(Y = 0 \mid x_i) = 1 - p(x_i) = \frac{e^{-(\beta_0 + \beta_1 x_i)}}{1 + e^{-(\beta_0 + \beta_1 x_i)}},$$

te

$$f(y; p(x_i)) = p(x_i)^y (1 - p(x_i))^{1-y}, \quad i = 1, \ldots, N.$$

Funkcija maksimalne vjerojatnosti je sada jednaka

$$L(y_1, ..., y_N; \beta_0, \beta_1) = \sum_{i=1}^{N} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i},$$

a njezin logaritam

$$\begin{aligned} l(y_1, ..., y_N; \beta_0, \beta_1) &= \sum_{i=1}^{N} y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \\ &= \sum_{i=1}^{N} y_i \log \frac{p(x_i)}{1 - p(x_i)} + \log(1 - p(x_i)) \\ &= \sum_{i=1}^{N} y_i(\beta_0 + \beta_1 x_i) - (\beta_0 + \beta_1 x_i) - \log(1 + e^{-(\beta_0 + \beta_1 x_i)}). \end{aligned} \tag{2.13}$$

S obzirom da je cilj odrediti parametre β_0 i β_1 tako da funkcija l bude maksimalna, izraz (2.13) je potrebno derivirati po β_0 i β_1. Dobiva se

$$\begin{aligned} \frac{\partial l}{\partial \beta_0} &= \sum_{i=1}^{N} y_i - 1 - \frac{e^{-(\beta_0 + \beta_1 x_i)}}{1 + e^{-(\beta_0 + \beta_1 x_i)}}(-1) \\ &= \sum_{i=1}^{N} y_i - \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}} \\ &= \sum_{i=1}^{N} y_i - p(x_i) \end{aligned} \tag{2.14}$$

$$\begin{aligned} \frac{\partial l}{\partial \beta_1} &= \sum_{i=1}^{N} x_i y_i - \frac{x_i}{1 + e^{-(\beta_0 + \beta_1 x_i)}} \\ &= \sum_{i=1}^{N} x_i y_i - x_i p(x_i) \end{aligned} \tag{2.15}$$

Izjednačavanjem dobivenih izraza (2.14) i (2.15) s nulom dobiva se nelinearni sustav jednadžbi za nepoznate parametre β_0 i β_1:

$$\begin{cases} \sum\limits_{i=1}^{N} y_i - p(x_i; \beta_0, \beta_1) = 0 \\ \sum\limits_{i=1}^{N} x_i y_i - x_i p(x_i; \beta_0, \beta_1) = 0 \end{cases} \tag{2.16}$$

koji je potrebno riješiti. Sustav se tipično rješava nekom od metoda za rješavanje nelinearnih sustava, kao npr. Newton–Raphsonovom metodom.

Procjena parametara kod višestruke logističke regresije dobit će se sličnim postupkom kao i kod jednostavne logističke regresije. Razlika je u tome da će se u tom slučaju, za n prediktorskih varijabli, dobiti nelinearni sustav $n + 1$ jednadžbi s $n + 1$ nepoznanica koji je potrebno riješiti.

S obzirom na dostupnost brojnih statističkih paketa, u ovom se radu koeficijenti logističke regresije neće određivati korištenjem numeričkih metoda za rješavanje sustava (2.16), već će se koeficijenti, kao i pripadajući model izračunati koristeći gotove funkcije u programskom jeziku *Python*.

3. PRIMJENA JEDNOSTAVNE LOGISTIČKE REGRESIJE

U idućim će se primjerima odrediti model logističke regresije koristeći programski jezik *Python*. Opisat će se korištenje različitih modula i funkcija potrebnih za izgradnju modela. Dobiveni model logističke regresije koristit će se za klasifikaciju. Promatrat će se ovisnost modela o nekim odabranim parametrima.

3.1. Primjena binarne logističke regresije za klasifikaciju tumora

U ovom primjeru primjenit će se logistička regresija za predviđanje hoće li tumor biti kancerogen na temelju njegove veličine.

Pretpostavlja se da su poznati ulazni podaci oblika $(x_i, y_i), i = 1, ..., N$, pri čemu se x_i odnosi na veličinu tumora, a $y_i \in \{0, 1\}$, pri čemu $(Y = 1)$ znači da je tumor kancerogen, a $(Y = 0)$ da to nije slučaj.

Da bi u *Pythonu* odredili model logističke regresije, najprije je potrebno pozvati određene module. U ovome su radu korišteni sljedeći moduli (Slika 3.1):

- *NumPy* - koristi se za rad s matricama i u domeni linearne algebre
- *Matplotlib* - služi za vizualizaciju podataka
- *sklearn* - sadržava alate za analizu podataka i predviđanje
- *statsmodels* - sadržava funkcije za statističku analizu i izgradnju statističkih modela.

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
import statsmodels.api as sm
```

Slika 3.1. Moduli korišteni u Pythonu

Podaci u promatranom primjeru prikazani su na Slici 3.2. Podaci u vektoru X predstavljaju nezavisne slučajne varijable koja označava veličinu tumora u centimetrima, dok podaci u vektoru y predstavljaju empirijski utvrđene izlazne vrijednosti, čije su vrijednosti binarne i odnose se na činjenicu je li tumor bio kancerogen ili nije.

```
X = np.array([1.98,0.49,0.25,0.08,3.78,3.18,2.44,2.09,0.14,1.72,1.65,4.92,4.37,
              4.96,4.52,1.69,5.88,6.01,4.58,4.97,3.67,5.25]).reshape(-1,1)
y = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

Slika 3.2. Ulazni podaci

Formiranje modela logističke regresije u *Pythonu* prikazano je na Slici 3.3. Pritom se naredbom `linear_model.LogisticRegression()` definira instanca klase koja predstavlja objekt

```
logr = linear_model.LogisticRegression()
logr.fit(X,y)

print('Koeficijenti su jednaki:', logr.coef_[0], logr.intercept_)
```

```
Koeficijenti su jednaki - β1 iznosi [1.25983762] ,a slobodni član β0 iznosi: [-4.04132893]
```

Slika 3.3. Formiranje modela logističke regresije u Pythonu

kojim se opisuje model logističke regresije. Funkcija `fit()` tog objekta optimizira parametre logističke regresije temeljem danih podataka X i y (Slika 3.3). U nastavku su za odabrani primjer ispisani koficijenti dobivenog modela logističke regresije.

Dobiveni model može se iskoristiti za predviđanje hoće li tumor biti kancerogen ako je njegova veličina npr. 3,46 centimetra. Postupak i izlazni rezultat u *Pythonu* prikazani su na Sici 3.4.

```
xp = np.array([3.46]).reshape(-1,1)
predicted = logr.predict(xp)
if predicted==1:
    print('Previđeno:', predicted,"- što znači da je tumor kancerogen")
else:
    print('Previđeno:', predicted,"- što znači da nije tumor kancerogen")
```

```
Previđeno: [1] - što znaći da je tumor kancerogen
```

Slika 3.4. Predviđanje pomoću dobivenog modela logističke regresije

Iz dobivenog se modela mogu odrediti i izgledi, pa je za prikazani slučaj to prikazano u nastavku na Slici 3.5. Dobivena vrijednost izgleda je 3.52, što znači da se povećanjem veličine tumora za 1 centimetar izgledi da se radi o tumoru povećavaju za 3.52 puta.

```
logr = linear_model.LogisticRegression()
logr.fit(X,y)

log_odds = logr.coef_
odds = np.exp(log_odds)

print(odds)
```

```
izgledi iznose:  [3.52484907]
```

Slika 3.5. Izgledi (eng. odds)

Vrijednost dobivene logističke funkcije predstavlja vjerojatnost da je za pojedine veličine tumora, isti kacerogen. Te se vjerojatnosti mogu odrediti funkcijom `ogr.predict_proba(), pri čemu je` označen dobiveni regresijski model. Na Slici 3.6 je prikazana funkcija u kojoj je implementirana vrijednost logističke funkcije "po definiciji" prema izrazu (2.3).

```
def logit2prob(logr, X):
  log_odds = logr.coef_[0] * X + logr.intercept_
  odds = np.exp(log_odds)
  probability = odds / (1 + odds)
  return probability

print(logit2prob(logr, X))
```

Slika 3.6. Vjerojatnosti dobivene iz modela logističke regresije

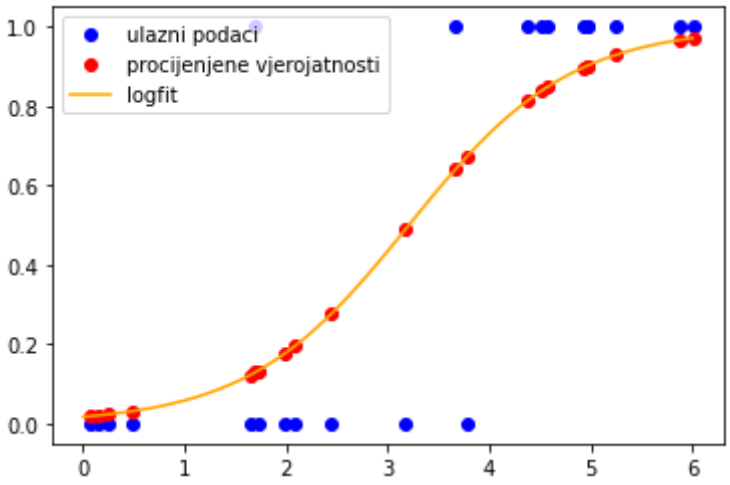
```
[0.17554163 0.03155351 0.02351388 0.01906706 0.67279577 0.49123957
 0.27541398 0.19650733 0.02053347 0.13303248 0.12318628 0.8963278
 0.81216833 0.90091793 0.83931459 0.12873352 0.96664154 0.97153758
 0.84924956 0.90203685 0.64159075 0.92909161]
```

Slika 3.7. Vjerojatnosti procijenjene pomoću modela logističke regresije za zadane ulazne podatke X

Za zadane ulazne podatke X prikazane na Slici 3.2, dobivene vjerojatnosti prikazane su na Slici 3.7 i mogu se interpretirati na sljedeći način:

- Procijenjena vjerojatnost da je tumor veličine 1.98 cm kancerogen je 17.5%.
- Procijenjena vjerojatnost da je tumor veličine 0.49 cm kancerogen je 3.1%.
- Procijenjena vjerojatnost da je tumor veličine 0.25 cm kancerogen je 2.3%.
- Procijenjena vjerojatnost da je tumor veličine 5.25 cm kancerogen je 92.9%.
- ...

Na Slici 3.8 su vjerojatnosti dobivene iz logističkog modela prikazane crvenim točkicama.



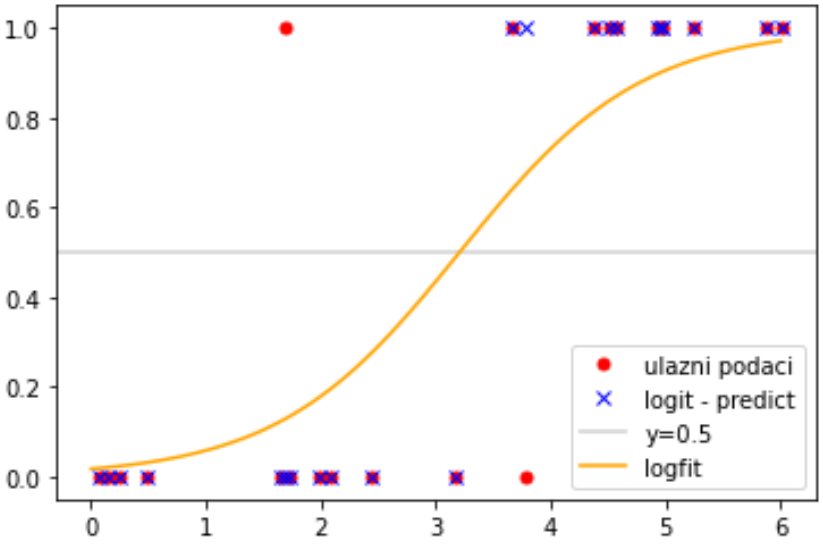
Slika 3.8. Grafički prikaz procijenjenih vjerojatnosti dobivenih modelom logističke regresije.

Dobivena logistička regresija se sada može koristiti za kategorizaciju na sljedeći način. Uzme li se kao granica vjerojatnost 0.5, za $p(x) < 0.5$ procijenjena vrijednost slučajne varijable je $(Y = 0)$, a za $p(x) > 0.5$ će procjena dati $(Y = 1)$. U promatranom primjeru je kod koji ispisuje vrijednosti

```
xp = np.array([X]).reshape(-1,1)
predicted = logr.predict(xp)
print('Previđeno:', predicted)
print('Score:', logr.score(X,y))
```

```
Previđeno: [0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 1]
```

Slika 3.9. Predviđanje ulaznih podataka pomoću dobivenog modela logističke regresije.



Slika 3.10. Usporedba ulaznih podataka i procijenjenih vrijednosti

slučajne varijable Y, odnosno kategorije, procijenjene modelom logističke regresije prikazan na Slici 3.9, dok je na Slici 3.10 prikazana usporedba ulaznih podataka i procijenjenih vrijednosti.

Za promatrane podatke predviđene su vrijednosti za Y prikazane crvenim točkama, dok su plavim križićem prikazani ulazni podaci. Točke u kojima se za iste vrijednosti x, procijenjene i zadane empirijske vrijednosti ne poklapaju su točke u kojima je napravljena pogrešna klasifikacija. U ovom slučaju podaci koji su pogrešno procijenjeni prikazani su na Slici 3.10 križićem (ulazni podaci) koji nije prekriven crvenom točkom (podaci dobiveni predviđanjem), a odnose se na tumor veličine 3.78 cm koji nije kancerogen i tumor veličine 1.69 cm koji je kancerogen.

Utjecaj parametara optimizacije na procjenu logističkog modela

Kod određivanja parametara logističke regresije provodi se optimizacijski postupak koji minimizira funkciju cilja koja je u ovom slučaju jednaka logaritmu funkcije maksimalne vjerojatnosti i naziva se funkcija gubitka. Funkcija cilja se u nekim slučajevima može i zamijeniti na način da se osim funkcije gubitka uzima u obzir i funkcija "kazne" koja se u funkciju cilja uključuje s nekim regularizacijskim parametrom. U *Pythonu* je taj regularizacijski parametar u funkciji `LogisticRegression()` označen s C. U ovom se primjeru promatra utjecaj koeficijenta regularizacije C na dobiveni model logističke regresije. Na Slici 3.11 prikazan je kod za izgradnju modela logističke regresije za različite vrijednosti regularizacijskog parametra, te dobivene vrijednosti parametara logističke regresije za odabrane primjere (Slika 3.12).

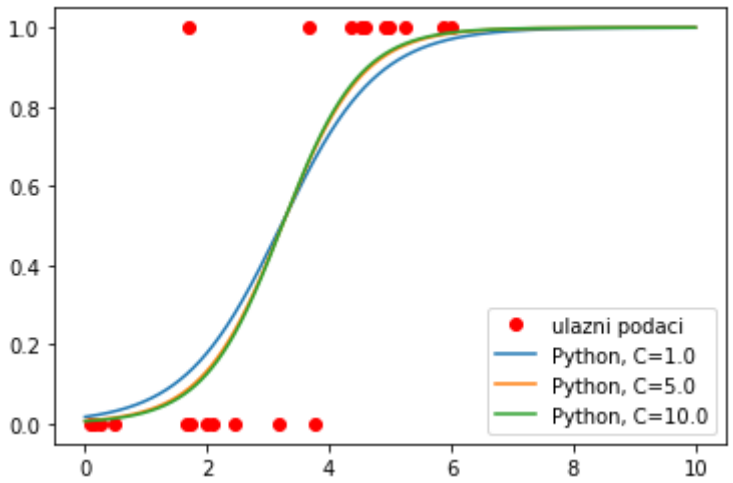
```
c=[1.,5.,10.]
for ci in c:
    logr = linear_model.LogisticRegression(C=ci)
    logr.fit(X,y)
    log_odds = logr.coef_
    odds = np.exp(log_odds)
    print("C=",ci)
    print('odds:',odds[0].flatten())
    print('Parametri: ',"\u03B21", logr.coef_[0],"\u03B20", logr.intercept_)
```

Slika 3.11. Formiranje modela logističke regresije korištenjem različitih koeficijenata regularizacije i vrijednosti parametara dobivenih modela.

Funkcije vjerojatnosti dobivenih logističkih modela prikazane su na Slici 3.13

```
C= 1.0
odds: [3.52484907]
Parametri:  β1 [1.25983762] β0 [-4.04132893]
C= 5.0
odds: [4.57998174]
Parametri:  β1 [1.52169501] β0 [-4.90955846]
C= 10.0
odds: [4.82116392]
Parametri:  β1 [1.57301538] β0 [-5.08034311]
Optimization terminated successfully.
         Current function value: 0.280806
         Iterations 7
```

Slika 3.12. Vrijednosti parametara i izgleda dobivenih modela logističke regresije za odabrane koeficijente regularizacije.



Slika 3.13. Funkcije vjerojatnosti za različite koeficijente regularizacije

Logistička analiza u modulu statsmodels

Određivanje koeficijenata logističke regresije u Pythonu moguće je i korištenjem klasa koje se nalaze u modulu `statsmodels`. Klasa logističke regresije nalazi se u modulu `statsmodels.api`, a model klase se dobiva naredbom `Logit()`. Metoda `fit()` te klase optimizirat će parametre modela na isti način kao što je opisano u potpoglavlju 2.3.3.

```
x1 = sm.add_constant(X)
model = sm.Logit(y, x1).fit()
print(model.params)
print(model.pred_table())
print(model.summary())
```

Slika 3.14. Formiranje modela logističke regresije korištenjem modula statsmodels (sm).

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                   22
Model:                          Logit   Df Residuals:                       20
Method:                           MLE   Df Model:                            1
Date:                Fri, 13 Jan 2023   Pseudo R-squ.:                  0.5949
Time:                        09:53:37   Log-Likelihood:                -6.1777
converged:                       True   LL-Null:                       -15.249
Covariance Type:            nonrobust   LLR p-value:                 2.049e-05
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -5.2788      2.224     -2.374      0.018      -9.638      -0.920
x1             1.6326      0.629      2.595      0.009       0.400       2.865
==============================================================================
```

Slika 3.15. Rezultati regresijske analize dobiveni pomoću modula statsmodels.

Na Slici 3.14 prikazan je kod kojim se formira model logističke regresije, te se ispisuju parametri modela i tablica za analizu statističkog modela (Slika 3.15).

Iz tablice se mogu očitati vrijednosti regresijskih koeficijenata: $\beta_0 = -5.28$, a $\beta_1 = 1.63$, te njihova značajnost u modelu. Kako je p-vrijednost za oba koeficijenta manja od 0.05 (označeno crvenim), to znači da su oba koeficijenta značajna u ovom modelu.

4. PRIMJENA VIŠESTRUKE LOGISTIČKE REGRESIJE

Kao što je opisano u potpoglavlju 2.2.2., višestruka logistička regresija podrazumijeva da na izlazne vrijednosti binomne varijable utječe više prediktorskih varijabli. Izgradnja modela višestruke logističke regresije u *Pythonu* ne razlikuje se od postupka izgradnje jednostavne logističke regresije. Python moduli koji će se koristiti u ovom primjeru jednaki su onima iz prethodnog primjera.

4.1. Višestruka logistička regresija za klasifikaciju prolaska kolegija

U ovom primjeru primijenit će se logistička regresija za predviđanje hoće li studenti položiti kolegij ili ne, na osnovu podataka o ostvarenim bodovima na testovima, broju pristupa on-line kolegiju, bodovima ostvarenima na srodnom kolegiju itd. Dakle, u ovom se slučaju pretpostavlja da ishod (zavisna varijabla) ovisi o više nezavisnih varijabli.

Kao nezavisne varijable u promatranom primjeru koriste se sljedeći podaci:

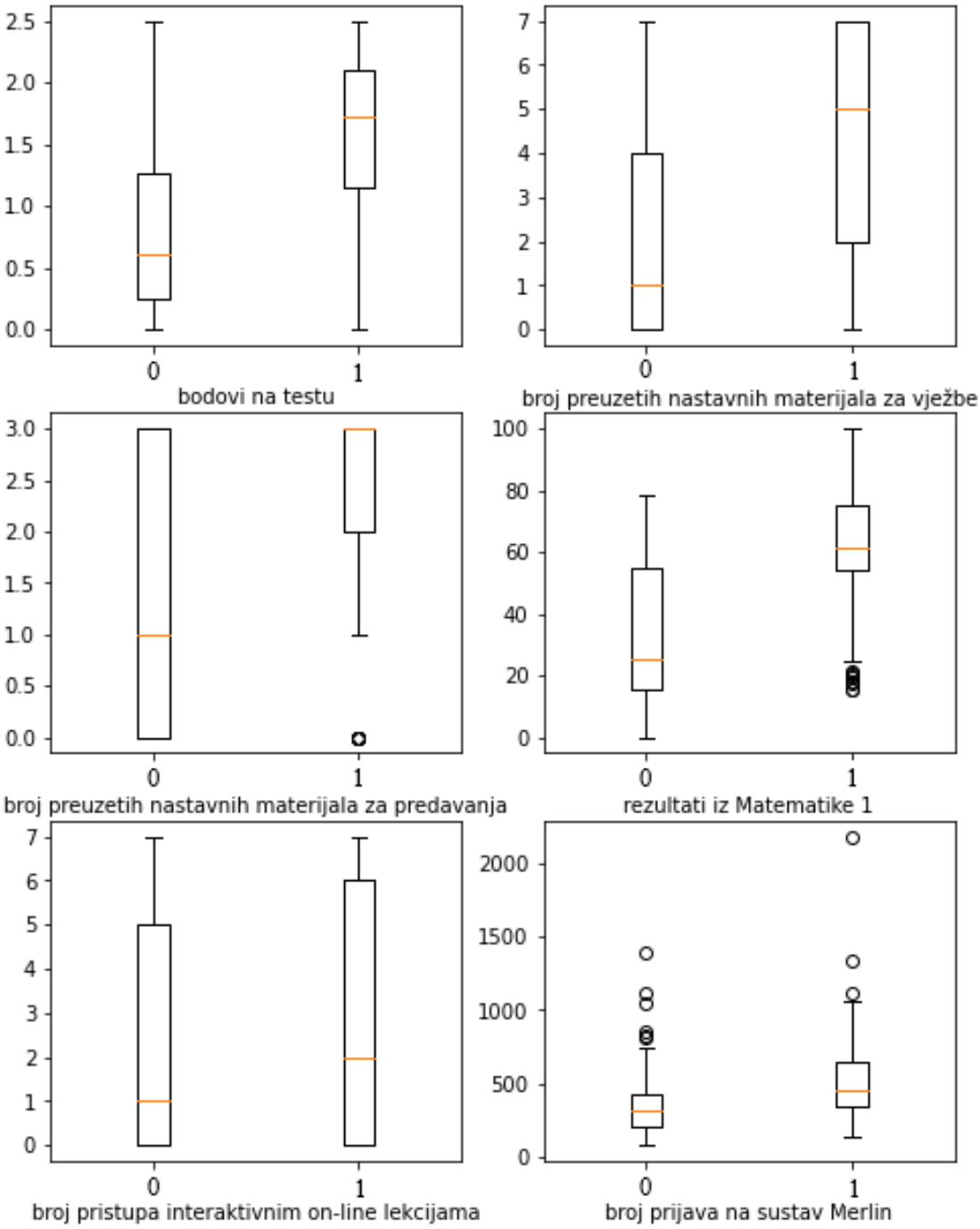
- bodovi na testu (oznaka: tests)
- broj pristupa interaktivnim on-line lekcijama (oznaka: lessons)
- broj "download"-anih nastavnih materijala za predavanja (oznaka: lectures)
- broj "download"-anih nastavnih materijala za vježbe (oznaka: exercises)
- rezultati iz Matematike 1 (oznaka: math1)
- broj prijava na sustav Merlin (oznaka: lms)

Zavisna varijabla, odnosno konačni rezultat, u ovom primjeru odnosi se na prolaz ili pad kolegija. Postupak unosa podataka u matricu X u kojoj su podaci nezavisnih varijabli, te matricu y izlaznih vrijednosti prikazan je na Slici 4.1. Kako bi se mogla izvršiti analiza kvalitete logističke regresije uneseni podaci se moraju podijeliti na podatke za učenje modela i podatke za validaciju, odnosno analizu kvalitete modela.

```
y=pd.read_excel('Sevilla_logr.xlsx')
test=y["tests"]
lessons=y["lessons"]
lectures=y["lectures"]
exercises=y["exercises"]
math1=y["math1"]
lms=y["lms"]
Y=y["result"]
X=y[["tests","lessons","lectures","exercises","math1","lms"]]
X_train, X_test, Y_train, Y_test = train_test_split
(y[["tests","lessons","lectures","exercises","math1","lms"]]
 ,y["result"], test_size=0.3,random_state=0)
```

Slika 4.1. Unos podataka o kolegiju Matematika 2

Grafički prikaz ulaznih podataka prikazan je na Slici 4.2 pomoću box-plot dijagrama. Box-plot dijagram objedinjuje pet karakterističnih vrijednosti statističkog skupa: minimalnu vrijednost, prvi kvartil, medijan (drugi kvartil), treći kvartil i maksimalnu vrijednost.



Slika 4.2. Vrijednosti ulaznih parametara za kolegij Matematika 1

Kao i u prethodnom primjeru, radi se o binarnoj logističkoj regresiji koja se sastoji samo od dvije klase. Postupak izgradnje modela prikazan je na Slici 4.3.

```
logr = linear_model.LogisticRegression(max_iter=1200)
logr.fit(X_train,Y_train)
```

Slika 4.3. Izgradnja modela logističke regresije.

Dobivena logistička regresija je ovom slučaju oblika:

$$p(x_1, .., x_6) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + ... + \beta_6 x_6)}}.$$

Vrijednosti parametara dobivene u *Pythonu* prikazane su na Slici 4.4

```
Koeficijenti su jednaki - β1,...,β6 iznose: [ 0.9518686   0.17990137  0.47559742
0.09037327  0.04810381 -0.00096929] β0 - slobodni član iznosi: [-4.7267608]
```

Slika 4.4. Koeficijenti dobiveni pomoću modula sklearn.

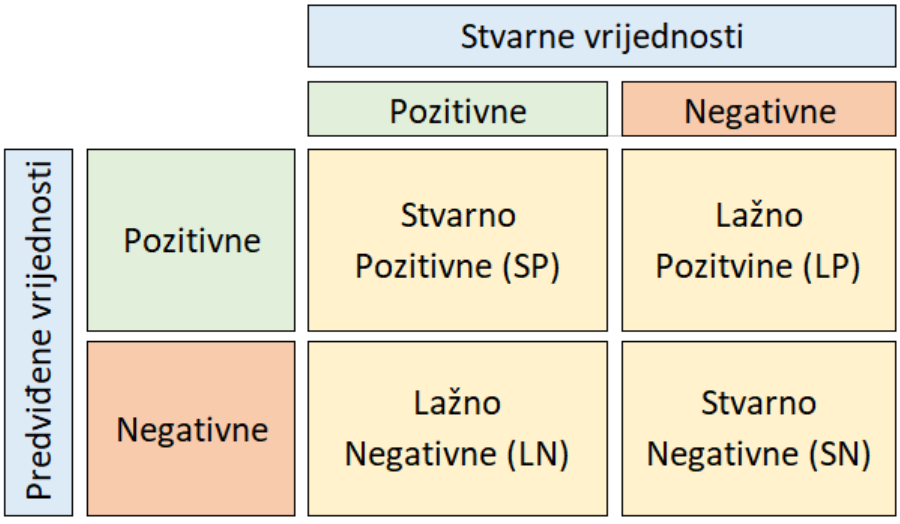
Analiza kvalitete i točnosti predviđanja pomoću logističke regresije

Na ovom će se primjeru dodatno opisati postupak za analizu točnosti predviđanja logističke regresije u postupku klasifikacije. Analiza se tipično provodi na podacima koji su namijenjeni za testiranje i koji su prethodno izdvojeni iz osnovnog skupa podataka, a temelji se na matrici konfuzije (eng. *confusion matrix*).

Da bi se formirala matrica konfuzije, potrebno je testirati klasifikator na poznatom skupu podataka koji ima poznate oznake klasa. Klasifikator će klasificirati svaki unos iz testnog skupa podataka, a zatim ćete usporediti dobivenu klasifikaciju s stvarnom oznakom klase kako bi se utvrdila točnost klasifikacije.

Matrica konfuzije je tablica koja prikazuje rezultate klasifikacije za svaku klasu, pri čemu se promatra odnos između stvarnih i predviđenih vrijednosti. Promatraju se četiri klase odnosno broj elemenata u svakoj od njih: broj uzoraka koji su ispravno klasificirani kao pozitivni (stvarno pozitivni), broj uzoraka koji su ispravno klasificirani kao negativni (stvarno negativni), broj uzoraka koji su lažno klasificirani kao pozitivni (lažno pozitivni) i broj uzoraka koji su lažno klasificirani kao negativni (lažno negativni).

Matrica konfuzije sljedećeg je oblika:



Slika 4.5. Matrica konfuzije

Pored matrice konfuzije, logistička regresija ima još nekoliko važnih značajki koje se koriste za evaluaciju i interpretaciju njenih rezultata. Neke od tih značajki su točnost (eng. *accuracy*), preciznost (eng. *precision*) i odziv (eng. *recall*). Ove značajke su direktno vezane za matricu konfuzije i iz nje se određuju.

Točnost se koristi za pronalaženje udjela ispravno klasificiranih vrijednosti. Govori nam koliko je često promatrani klasifikator u pravu. To je zbroj svih stvarnih vrijednosti podijeljen ukupnim vrijednostima.

$$\text{Točnost} = \frac{SP + SN}{SP + LP + LN + SN} \tag{4.1}$$

Preciznost se koristi za izračunavanje sposobnosti modela da ispravno klasificira pozitivne vrijednosti. To su stvarni pozitivni rezultati podijeljeni s ukupnim brojem predviđenih pozitivnih vrijednosti.

$$\text{Preciznost} = \frac{SP}{SP + LP} \tag{4.2}$$

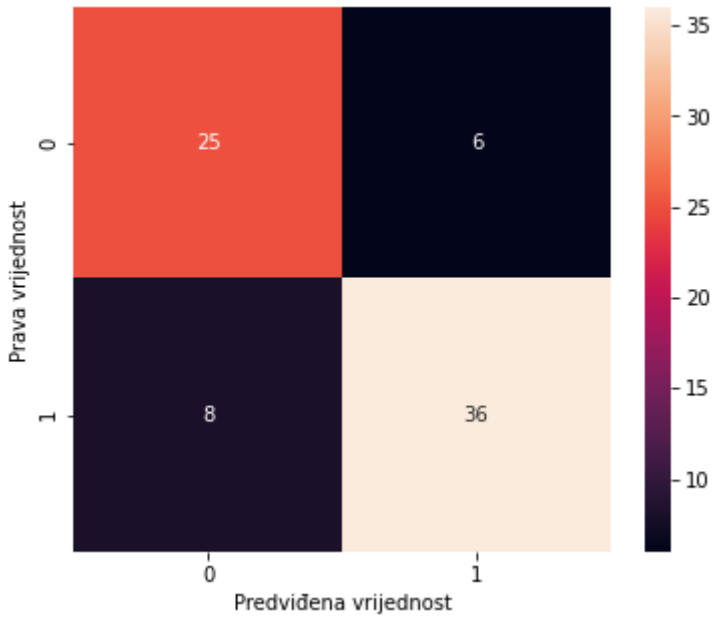
Odziv ili osjetljivost se koristi za izračun sposobnosti modela da predvidi pozitivne vrijednosti. To su pravi pozitivni rezultati podijeljeni s ukupnim brojem stvarnih pozitivnih vrijednosti.

$$\text{Osjetljivost} = \frac{SP}{SP + LN} \tag{4.3}$$

Nakon definiranih osnovnih pojmova vezanih za matricu konfuzije može se pristupiti određivanje same matrice konfuzije za zadani skup podataka.

```
predicted_output = logr.predict(X_test)
cm = confusion_matrix(Y_test, predicted_output)
print(cm)

plt.figure(figsize = (6,5))
sn.heatmap(cm, annot=True)
plt.xlabel('Predviđena vrijednost')
plt.ylabel('Prava vrijednost')
```

Slika 4.6. Dobivanje matrice konfuzije i njezin prikaz



Slika 4.7. Matrica konfuzije za klasifikaciju pomoću dobivenog modela logističke regresije.

Matrica konfuzije se u *Pythonu* dobiva korištenjem funkcije `confusion_matrix`(Y_test, $predicted\_output$) iz modula `sklearn.metrics`. Ulazni podaci te funkcije su empirijski podaci izdvojeni za testiranje, označeni ovdje s Y_test te podaci klasifikacije predviđeni pomoću dobivenog modela logističke regresije s $predicted\_output$ (Slika 4.6). Vizualni prikaz dobivenih vrijednosti u matrici konfuzije dobiven korištenjem funkcije `heatmap` iz modula `seaborn` dan je na Slici 4.7. Kako bi se protumačila dobivena matrica konfuzije, prvo se moraju pogledati vrijednosti na dijagonali. Broj 25 koji govori koliko je stvarnih pozitivnih vrijednosti previđeno, dok broj 36 govori koliko je stvarnih negativnih vrijednosti predviđeno. Brojevi koji se nalaze van dijagonale, odnose se na lažna predviđanja, tako u ovom slučaju broj 6 znači da je za 6 podataka predviđen pozitivan ishod, iako je on zapravo negativan. Za 8 ulaznih podataka predviđeni ishod je negativan, a on je zapravo pozitivan. Dakle, nakon dobivene matrice konfuzije može se odrediti točnost binarne logističke regresije. Računanje točnosti modela i vrijednosti za dobiveni primjer prikazanu su na Slici

Jedna od mogućih mjera kvalitete logističke regresije može se odnositi na značajnost pojedinih parametara u modelu. Na prethodnom je primjeru prikazano da model logističke regresije u modulu `statmodels` ima mogućnost ispisa izvještaja koji između ostalog daje značajnost svake pojedine varijable u modelu. Postupak prilagodbe modela podacima i ispis izvještaja u *Pythonu* prikazan je na Slici 4.9.

Izvještaj je prikazan u nastavku:

Od svih koeficijenata, jedini značajni su oni koji imaju p vrijednost manju od 0.05. U ovom primjeru koeficijenti koji su značajni se odnose na "math1" i "tests", odnosno na broj bodova iz Matematike 1 i broj bodova na testu. Može se zaključiti kako prethodno usvojeno znanje iz predmeta Matematika 1, odnosno na testovima iz Matematike 2 bitno utječu na to hoće li student položiti kolegij.

Stoga će se u nastavku odrediti logistička regresija koja uključuje samo te dvije varijable.

```
a = cm.shape
corrPred = 0
falsePred = 0
for row in range(a[0]):
    for c in range(a[1]):
        if row == c:
            corrPred +=cm[row,c]
        else:
            falsePred += cm[row,c]

print('Ispravna predviđanja: ', corrPred)
print('Neispravna prediđanja:', falsePred)
print ('Točnost binarne logističke klasifikacije je: ', corrPred/(cm.sum()))
```

```
Ispravna predviđanja:  60
Neispravna prediđanja: 15
Točnost binarne logističke klasifikacije je:  0.8
```

Slika 4.8. Kod za točnost binarne logističke regresije

```
x1 = sm.add_constant(X, prepend=False)
logit_mod = sm.Logit(Y, X)
logit_fit = logit_mod.fit()
print (logit_fit.summary())
```

Slika 4.9. Izgradnja modela logističke regresije u Pythonu i ispis izvještaja.

Logit Regression Results

Dep. Variable:	result	No. Observations:	249
Model:	Logit	Df Residuals:	243
Method:	MLE	Df Model:	5
Date:	Fri, 13 Jan 2023	Pseudo R-squ.:	0.1073
Time:	13:02:57	Log-Likelihood:	-153.43
converged:	True	LL-Null:	-171.87
Covariance Type:	nonrobust	LLR p-value:	6.317e-07

	coef	std err	z	P>|z|	[0.025	0.975]
tests	0.4764	0.229	2.078	0.038	0.027	0.926
lessons	0.0111	0.064	0.175	0.861	-0.114	0.136
lectures	-0.1247	0.136	-0.916	0.359	-0.391	0.142
exercises	0.0935	0.066	1.419	0.156	-0.036	0.223
math1	0.0124	0.006	2.081	0.037	0.001	0.024
lms	-0.0015	0.001	-1.900	0.057	-0.003	4.79e-05

Logistička regresija u ovom slučaju glasi:

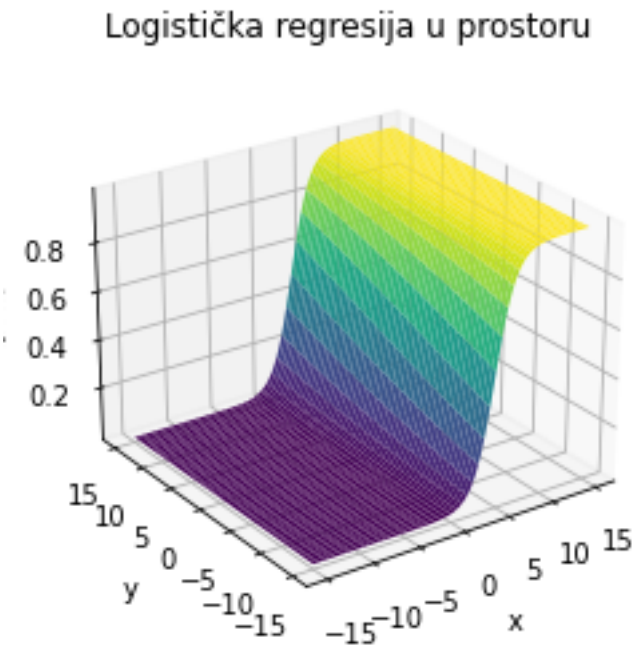
$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}},$$

pri čemu su vrijednosti parametara izračunate u *Pythonu* jednake:

$$\beta_0 = -3.5, \beta_1 = 0.91 \text{ i } \beta_2 = 0.05.$$

Ova se logistička regresija može prikazati i grafički, kao ploha u prostoru (Slika 4.10.).

Za primjer, korištenjem dobivenog modela, može se procijeniti kolika je vjerojatnost da će

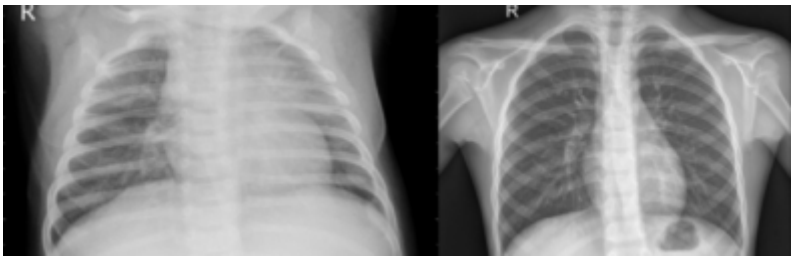


Slika 4.10. Funkcija vjerojatnosti u 3D

student koji je imao 4 boda na testu i koji je imao 65 bodova iz matematike 1 položiti kolegij matematika 2. Tražena procijenjena vjerojatnost iznosi 0.967 i dobiva se jednostavno uvršavanjem vrijednosti u prethodni izraz.

4.2. Klasifikacija rendgenskih snimaka pluća pomoću logističke regresije

U ovom poglavlju obrađuje se primjena binarne logističke regresije za klasifikaciju slika koje prikazuju rendgenski snimak pluća. U području medicine, rendgenski snimci često se koriste za dijagnosticiranje bolesti pluća, poput upale pluća. Upala pluća je ozbiljna infekcija koja može biti uzrokovana virusima, bakterijama i drugim mikroorganizmima, a može dovesti do ozbiljnih komplikacija ako se ne liječi pravilno.



Slika 4.11. Rendgenski snimak upaljenih pluća (lijevo) i normalnih pluća (desno)

Cilj ovog poglavlja je razviti model koji može automatski klasificirati rendgenske snimke pluća kao normalna ili upaljena (Slika 4.11). Pretpostavlja se da se slike mogu pripisati samo jednoj od dvije klase, tj. klasi normalnih ili upaljenih pluća, pa se radi o binarnoj klasifikaciji.

Binarna logistička regresija je jedan od najčešćih pristupa klasifikaciji slika i pomoću nje će se procijeniti vjerojatnost da slika pripada jednoj od dvije klase, a na temelju značajki koje se izlučuju iz slike. Ove značajke se mogu odnositi na različite karakteristike pluća, kao što su gustoća tkiva,

veličina pluća i drugi faktori koji mogu biti relevantni za dijagnosticiranje upale pluća. U ovom primjeru korišteno je 234 slika normalnih pluća i 390 slika upaljenih pluća (za treniranje modela).

```
import os
import numpy as np
import cv2
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
import seaborn as sn
import matplotlib.image as mpimg
from sklearn.metrics import roc_curve, auc, roc_auc_score
from sklearn.metrics import classification_report
```

Slika 4.12. Moduli potrebni u Pythonu za ovaj primjer

```
def load_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename))
        if img is not None:
            img = cv2.resize(img, (IMG_SIRINA, IMG_VISINA))
            images.append(img)
    return np.array(images)
```

Slika 4.13. Funkcija za učitavanje slika.

Nakon učitavanja odgovarajućih modula (Slika 4.12) učitavaju se slike iz mape u kojoj su one spremljene. Funkcija za učitavanje slika prikazana je na Slici 4.13. Funkcija uzima putanju mape kao ulazni argument. Zatim se u funkciji "load_images_from_folder" kreira prazna lista "images" koja će služiti za pohranjivanje učitanih slika. Petljom "for" prolazi se kroz sve datoteke u mapi koja je zadata putanjom "folder". Za svaku datoteku se koristi funkcija "cv2.imread" za čitanje slike, a ako je slika pročitana bez greške, skalira se na veličinu "IMG_SIRINA" x "IMG_VISINA" korištenjem funkcije "cv2.resize". Visina i širina slika jednake su 100.

```
#učitavaju se slike normalnih pluća
normal_images = load_images_from_folder(normal_dir)
normal_labels = np.zeros(len(normal_images))
#učitavaju se slike upaljenih pluća
pneumonia_images = load_images_from_folder(pneumonia_dir)
pneumonia_labels = np.ones(len(pneumonia_images))
#spajam slike i labele u jednu matricu (np.concatenate spaja dva niza u jedan)
X = np.concatenate((normal_images, pneumonia_images), axis=0)
y = np.concatenate((normal_labels, pneumonia_labels), axis=0)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
#izrada i treniranje modela logističke regresije
model = LogisticRegression(max_iter=1200)
model.fit(X_train.reshape(len(X_train), -1), y_train)
```

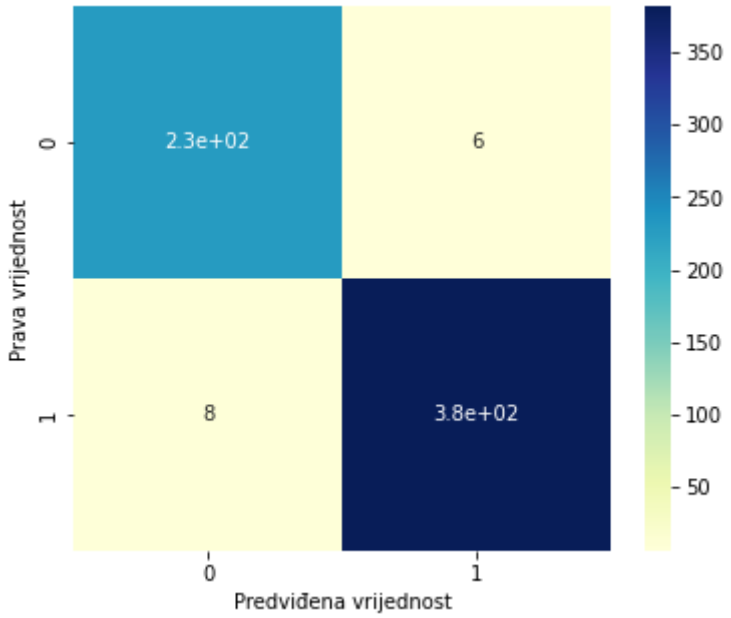
Slika 4.14. Formiranje modela logističke regresije

Nakon učitavanja slika, podaci se podijele na skup za treniranje i skup za testiranje, te se zatim formira model koristeći već prethodno opisanu metodologiju (Slika 4.14). Funkcija `concatenate`

koristi se za spajanje dva niza te se ovdje koristi se za spajanje slika i njihovih odgovarajućih oznaka, tako da se mogu koristiti kao ulaz za logistički regresijski model. Ulančavanjem nizova, slike i oznake raspoređuju se istim redosljedom, pri čemu se svakoj slici pridružuje se odgovarajuća oznaka. Nakon što su podaci spojeni i podijeljeni u skupove za treniranje i testiranje, model `LogisticRegression` iz scikit-learn koristi se za treniranje klasifikatora binarne logističke regresije na podacima za treniranje. Dobiveni se model zatim koristi za predviđanje na podacima za testiranje.

Analiza točnosti klasifikatora i predviđanja pomoću ROC krivulje

Usporedbom predviđanja i stvarnih podataka za testiranje može se analizirati točnost klasifikatora. Broj elemenata koji pripadaju određenoj kategoriji u matrici konfuzije prikazan je na Slici 4.15.



Slika 4.15. Matrica konfuzije modela binarne logističke regresije

Zbog velikog broja elemenata na dijagonali matrice konfuzije dobiveni model ima vrlo visoku točnost od 0.978. Povećanje točnosti rezultata može se povećati učitavanjem znatno većeg broja rendgenskih snimaka.

Na temelju predviđenih i stvarnih podataka za testiranje određuje se i ROC krivulja. ROC (eng. *Receiver Operating Characteristic*) krivulja je grafikon koji se koristi za evaluaciju performansi klasifikacijskih modela. Na os apcisa nanosi se lažno pozitivna stopa (FPR) (eng. *False Positive Rate*), koja se računa formulom

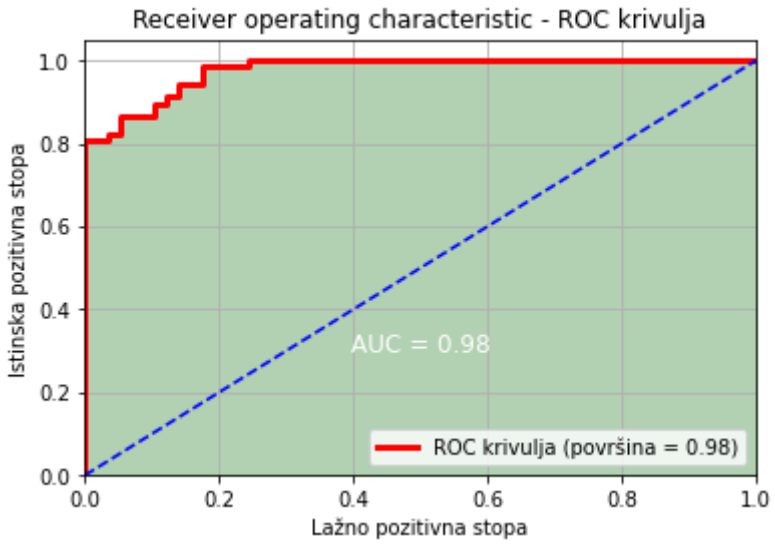
$$\text{FPR} = \frac{LP}{LP + SN}. \tag{4.4}$$

Na os ordinata nanose se pripadajuće istinske pozitivne stope (TPR) (eng. *True Positive Rate*), što je drugi naziv za osjetljivost. Model koji klasificira u potpunosti slučajno ima ROC krivulju koja je jednaka pravcu $y = x$. Kao mjera kvalitete klasifikatora može se računati površina ispod ROC

```
y_proba = model.predict_proba(X_test.reshape(X_test.shape[0], -1))
y_proba = y_proba[:, 1]

fpr, tpr, thresholds = roc_curve(y_test, y_proba)
roc_auc = auc(fpr, tpr)

plt.plot(fpr, tpr, color='red', lw=3, label='ROC krivulja (površina = %0.2f)' % roc_auc)
plt.grid()
plt.plot([0, 1], [0, 1], color='blue', lw=1.5, linestyle='--') #Plavi pravac
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Lažno pozitivna stopa')
plt.ylabel('Istinska pozitivna stopa')
plt.title('Receiver operating characteristic - ROC krivulja')
plt.legend(loc="lower right")
plt.fill_between(fpr, tpr, color='darkgreen', alpha=0.3)
plt.text(0.5, 0.3, 'AUC = %0.2f' % roc_auc, fontsize=12, ha='center', color='white')
plt.show()
plt.figure()

auc = roc_auc_score(y_test, y_proba)
print('Površina ispod ROC krivulje: {:.2f}'.format(auc))
```

Slika 4.16. Određivanje ROC krivulje u Pythonu

krivulje, a za tu se površinu uobičajeno koristi oznaka AUC, od engleskog naziva *Area Under Curve*. Za AUC vrijedi: $0 \leq \text{AUC} \leq 1$. Veća površina ispod ROC krivulje, odnosno veći AUC ukazuje na bolju performansu modela. Što je AUC bliže 1, to je model bolji u razlikovanju između pozitivnih i negativnih primjera. ROC krivulja za promatrani primjer prikazana je na Slici 4.17.



Slika 4.17. ROC krivulja

Ovdje je AUC $= 0.98$ što sugerira da je model vrlo dobar u razlikovanju pozitivnih i negativnih primjera i da ima visoku osjetljivost i specifičnost. Koliko točno iznose specifičnost i osjetljivost ovog modela može se provjeriti pomoću naredbe `classification_report`(y_test, y_pred), čime se dobiva sljedeći izvještaj:

	precision	recall	f1-score	support
0.0	0.88	0.86	0.87	49
1.0	0.91	0.92	0.92	76
accuracy			0.90	125
macro avg	0.89	0.89	0.89	125
weighted avg	0.90	0.90	0.90	125

Ovi rezultati ukazuju na to da je model relativno dobro klasificirao slike upaljenih i normalnih pluća, s visokom sensitivnošću i pristojnom preciznošću.

5. VIŠEKATEGORIJSKA REGRESIJA I NJEZINA PRIMJENA

5.1. Višekategorijska logistička regresija

Kao što je u Poglavlju 2 opisano, cilj logističke regresije je dobivanje modela koji će omogućiti procjenu vjerojatnosti nekog promatranog događaja, koji se može i ne mora dogoditi, uz pretpostavku da ishod ovisi o jednoj ili više prediktorskih varijabli. Pritom se kao model koristi logistička funkcija. Pokazalo se je da je veza između logaritma odziva i nezavisnih varijabli linearna i dana izrazom (2.4). Dobiveni se model potom koristi za određivanje pripadnosti jednoj od kategorija binarne slučajne varijable.

Višekategorijska logistička regresija predstavlja poopćenje jednostavne logističke regresije i koristi se za predviđanje vjerojatnosti pripadnosti nekoj od više mogućih kategorija, odnosno klasifikaciju u više kategorija. Na primjer, može se primijeniti na prepoznavanje rukom pisanih slova ili brojeva, za klasifikaciju slika životinja u različite vrste ili prepoznavanje različitih objekata u vozilima. U medicinskoj dijagnostici može se koristiti za predviđanje dijagnoze ili prognozu bolesti na temelju različitih kliničkih karakteristika. Općenito, ovaj model može se primijeniti u mnogim područjima gdje je potrebno klasificirati podatke u više kategorija na temelju različitih ulaznih varijabli.

Dakle, u situaciji kada se promatra problem u kojem je vrijednost slučajne varijable jedna od mogućih kategorija, odnosno diskretna kategorijska slučajna varijabla Y s funkcijom distribucije

$$Y \sim \begin{pmatrix} 1 & 2 & \cdots & K-1 & K \\ p_1 & p_2 & \cdots & p_{K-1} & p_K \end{pmatrix} \tag{5.1}$$

koja ovisi o jednoj ili više prediktorskih (nezavisnih) varijabli $X_1, \ldots, X_n$, koristit će se višekategorijska regresija. Ona će omogućiti procjenu vjerojatnosti pripadnosti nekoj od kategorija zavisne varijable Y u ovisnosti o empirijskim vrijednostima nezavisnih varijabli. Pritom će kao i kod jednostavne ili složene logističke regresije vrijediti da je veza između logaritma izgleda i nezavisnih varijabli linearna. Zbog jednostavnosti, u nastavku će biti napisani izrazi za slučaj kada postoji samo jedna nezavisna varijabla, ali se jednostavno to može proširiti i na situaciju s više nezavisnih varijabli. Znači, slično kao u (2.4), pretpostavit će se da vrijedi:

$$\begin{aligned} \log \frac{P(Y=1|x)}{P(Y=K|x)} &= \beta_{01} + \beta_{11}x \\ \log \frac{P(Y=2|x)}{P(Y=K|x)} &= \beta_{02} + \beta_{12}x \\ &\vdots \\ \log \frac{P(Y=K-1|x)}{P(Y=K|x)} &= \beta_{0(K-1)} + \beta_{1(K-1)}x \end{aligned} \tag{5.2}$$

Uvede li se za pripadnost kategoriji k oznaka

$$p_k(x) = P(Y = k \mid X = x)$$

iz (5.2) slijedi:

$$\begin{aligned} p_1(x) = P(Y=1|x) &= \frac{e^{\beta_{01}+\beta_{11}x}}{1+\sum_{l=1}^{K-1} e^{\beta_{0l}+\beta_{1l}x}} \\ p_2(x) = P(Y=2|x) &= \frac{e^{\beta_{02}+\beta_{12}x}}{1+\sum_{l=1}^{K-1} e^{\beta_{0l}+\beta_{1l}x}} \\ &\vdots \\ p_{K-1}(x) = P(Y=K-1|x) &= \frac{e^{\beta_{0(K-1)}+\beta_{1(K-1)}x}}{1+\sum_{l=1}^{K-1} e^{\beta_{0l}+\beta_{1l}x}} \\ p_K(x) = P(Y=K|x) &= \frac{1}{1+\sum_{l=1}^{K-1} e^{\beta_{0l}+\beta_{1l}x}} \end{aligned} \tag{5.3}$$

Zadatak višekategorijske regresije je određivanje parametara $\beta_{01}, \beta_{11}, \beta_{02}, \beta_{12}, \ldots, \beta_{0(K-1)}, \beta_{1(K-1)}$ korištenjem metode maksimalne vjerojatnosti, na sličan način kao što je opisano u potpoglavlju 2.3. Time će se dobiti model, odnosno funkcije oblika $p_k(x)$ koje će predstavljati procjenu vjerojatnosti da uz poznatu vrijednost prediktorske varijable x, zavisna varijabla pripada određenoj kategoriji, tj. da je $(Y = k)$. Na tome se temelji klasifikacija u više kategorija, odnosno promatrani objekt se klasificira u određenu kategoriju u skladu s najvećom vjerojatnošću.

Provođenje višekategorijske logističke regresija u *Pythonu* gotovo je identično postupku njezinog određivanja u binarnom slučaju. Stoga se postupak neće posebno detaljno opisivati.

Višekategorijska regresija u sljedećim će potpoglavljima biti primijenjena na primjeru klasifikacije vrste cvijeća temeljem više karakteristika, te na klasifikaciju slika. Osim toga, bit će opisani postupci za procjenu kvalitete dobivenog modela.

5.2. Primjena višekategorijske logističke regresije za klasifikaciju vrste cvijeća

Ovaj se primjer odnosi na višekategorijsku logističku regresiju koja će se provesti na skupu podataka poznatom u literaturi kao "Iris data set".

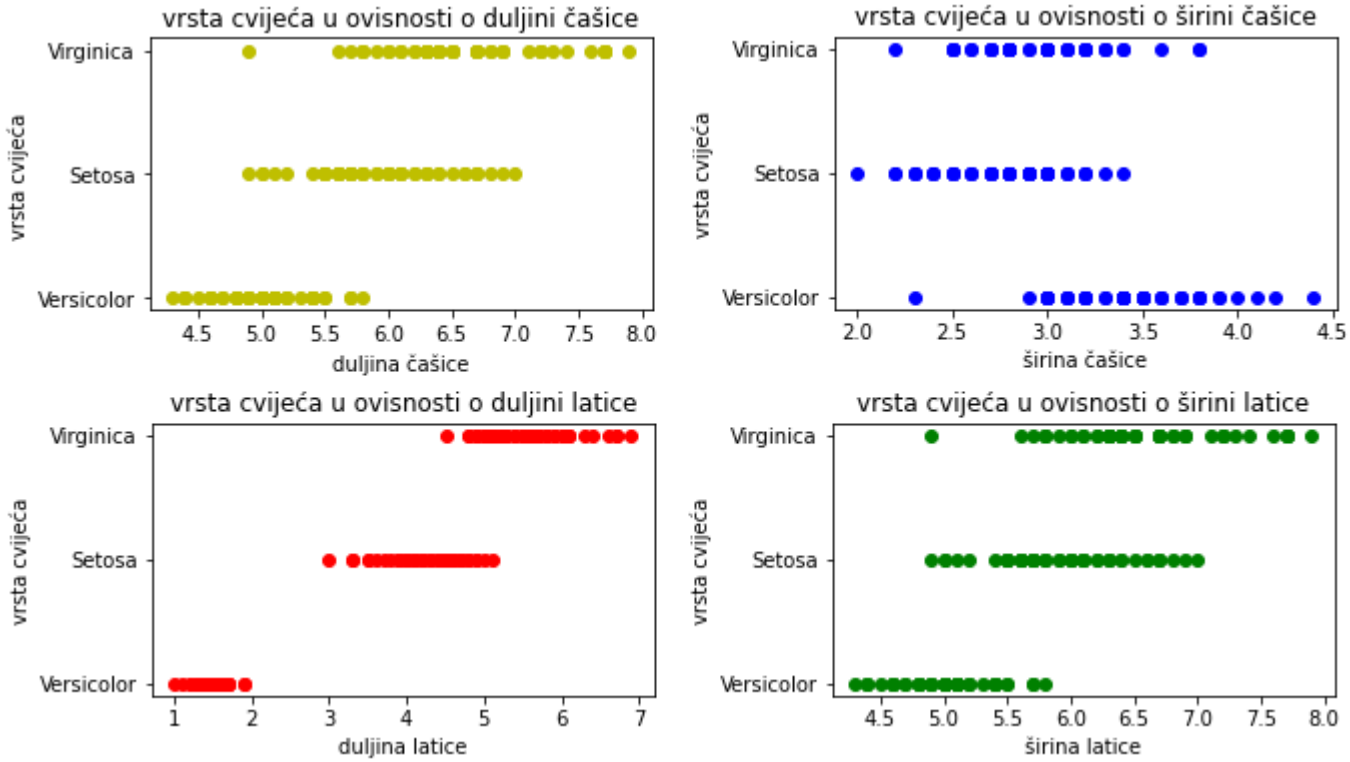
```
data = pd.read_csv('IRIS.csv')
y=data["variety"]
Y=data["variety"].replace({'Setosa':1, 'Versicolor':2, 'Virginica':3})
X=data[['sepal.length','sepal.width','petal.length','petal.width']]
X_train, X_test, Y_train, Y_test = train_test_split
(data[['sepal.length','sepal.width','petal.length','petal.width']]
,data['variety'], test_size=0.1,random_state=0)
```

Slika 5.1. Treniranje ulaznih podataka

Podaci u datoteci *IRIS.csv* sadržavaju podatke o duljini i širini čašice i latica, te se nalaze u stupcima "sepal length" i "petal length", "sepal width" i "petal width". Duljine i širine latica, odnosno čašica u ovom primjeru predstavljaju nezavisne varijable. Na Slici 5.1 prikazan je *Python* kod za učitavanje podataka i definiranje varijabli. Zavisne varijable, odnosno konačni rezultati u ovom primjeru, odnose se na vrstu cvijeća "variety", pri čemu su imena tih vrsta: Versicolor, Setosa i Virginica (Slika 5.2). Za potrebe provođenja logističke regresije, u modelu su se pojedinim kategorijama pridružile numeričke vrijednosti 1, 2 i 3.

Slika 5.2. Cvijeće Iris

Kako će se na ovom primjeru opisati i procjena kvalitete dobivene logističke regresije kao klasifikatora, učitani se podaci pomoću funkcije `train_test_split` u modulu *sklearn* podijele u dva podskupa: podatke za treniranje i podatke za testiranje.



Slika 5.3. Vrsta Iris cvijeća u ovisnosti o promatranim značajkama

Grafički prikaz podataka pomoću grafova prikazanih na Slici 5.3 ukazuje da postoji ovisnost kategorija o pojedinim značajkama, odnosno ulaznim varijablama.

```
logr = linear_model.LogisticRegression(max_iter=1200)
logr.fit(X_train,Y_train)
```

```
Koeficijenti su jednaki:
 β1,..,β4 - [-0.39747832  0.83329068 -2.28960456 -0.9785686 ] β0 - [8.99997302]
 β1,..,β4 - [ 0.54392309 -0.29016433 -0.23240701 -0.65809366] β0 - [1.54470238]
 β1,..,β4 - [-0.14644477 -0.54312635  2.52201157  1.63666226] β0 - [-10.5446754]
```

Slika 5.4. Prilagođavanje modela i ispis koeficijenata

Na Slici 5.4 prikazan je postupak formiranja modela u *Pythonu* i on je isti kao i u slučaju

jednostavne logističke regresije. Razlika dobivenog modela je u broju koeficijenata koji se pojavljuju, a koji ovisi o broju kategorija K, te broju nezavisnih varijabli n. Metoda dobivenog modela

```
predicted=logr.predict(np.array([[1,2,3,4]])).reshape(-1,1)
print('Previđeno:', predicted.flatten())
```

```
Previđeno: ['Virginica']
```

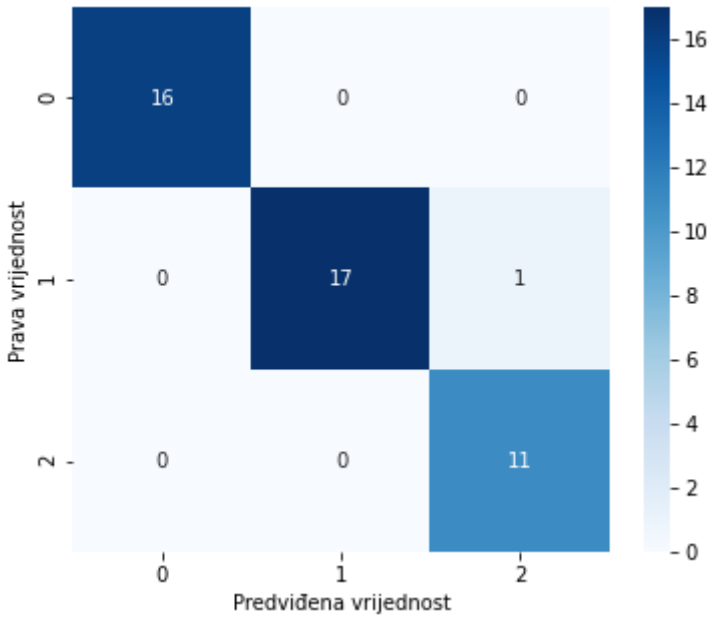
Slika 5.5. Procjena pomoću dobivenog modela

`predict` koristi se za predviđanje pomoću dobivenog modela. Na Slici 5.5 dan je primjer procjene vrste cvijeta u slučaju kada su varijable redom jednake: "sepal lenght" - 1, "sepal width" - 2, "petal lenght" - 3, a "petal width" - 4. Procijenjeno je da se radi o vrsti "Virginica".

Analiza kvalitete i točnosti logističke regresije za klasifikaciju

U nastavku će se promotriti konfuzijska matrica za ovaj primjer. S obzirom da su u primjeru 3 kategorije (zavisne varijable), dimenzija matrice konfuzije bit će dimenzije 3x3. Na osnovu vrijednosti dobivenih u konfuzijskoj matrici, a koji se odnose na broj ispravno i pogrešno klasificiranih vrijednosti, određuju se specifičnost i preciznost. Pritom se ne mogu koristiti formule definirane u potpoglavlju 4.1., ali se ovdje njihovo računanje neće posebno razmatrati.

```
cm = confusion_matrix(Y_test, predicted_output)
print(cm)

plt.figure(figsize = (6,5))
sn.heatmap(cm, annot=True,cmap="Blues")
plt.xlabel('Predviđena vrijednost')
plt.ylabel('Prava vrijednost')
```



Slika 5.6. Određivanje i prikaz matrice konfuzije

Kao što je već prethodno pojašnjeno, matrica konfuzije u *Pythonu* se određuje pomoću funkcije `confusion_matrix` i provodi se na testnom skupu. Grafički prikaz dobivene matrice korištenjem funkcije `heatmap` u modulu *seaborn* dan je na Slici 5.6. Broj ispravno klasificiranih podataka uvijek se nalazi na glavnoj dijagonali, a broj podataka izvan glavne dijagonale ukazuje na broj pogrešno klasificiranih podataka. Kako je u ovom primjeru samo jedna vrijednost izvan dijagonale, može se smatrati da je kvaliteta ove logističke regresije visoka.

```
predictions=logr.predict(X_test)
print(classification_report(Y_test,predictions))
print(accuracy_score(Y_test,predictions))
plt.figure()
```

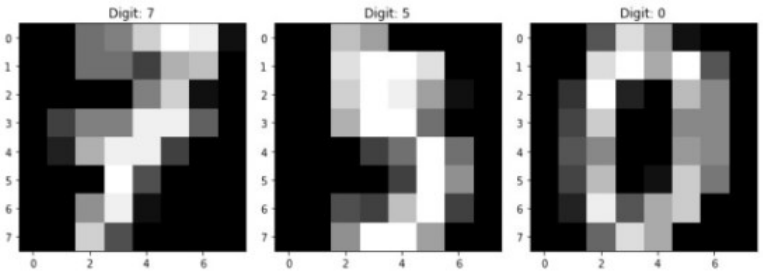
```
              precision    recall  f1-score   support

      Setosa       1.00      1.00      1.00        16
  Versicolor       1.00      0.94      0.97        18
   Virginica       0.92      1.00      0.96        11

    accuracy                           0.98        45
   macro avg       0.97      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

Slika 5.7. Mjere točnosti klasifikacije dobivenog modela.

Točnost predviđanja daje funkcija `classification_report`, a dobiva se, kao i konfuzijska matrica usporedbom testnih podataka s ishodima predviđanja. Funkcija će za svaku od klasa, dati vrijednosti za preciznost, specifičnost i f1-mjeru ("precision", "recall", "f1-score"), te za točnost predviđanja ("accuracy"). F1-mjera definira se kao harmonijska sredina preciznosti i specifičnosti. Rezultati su prikazani na Slici 5.7.

5.3. Višekategorijska logistička regresija za klasifikaciju slika

Ovo poglavlje fokusira se na primjenu višekategorijske logističke regresije za klasifikaciju slika rukom pisanih brojeva od 0 do 9. Dakle, postoji 10 klasa koje predstavljaju različite brojeve od 0 do 9, a cilj klasifikacije je dodijeliti ispravnu klasu svakoj slici. U ovom će se primjeru, osim uz pomoć logističke regresije, rezultati odrediti korištenjem nekih drugih algoritama strojnog učenja, te usporediti.



Slika 5.8. Primjer nekih od korištenih slika

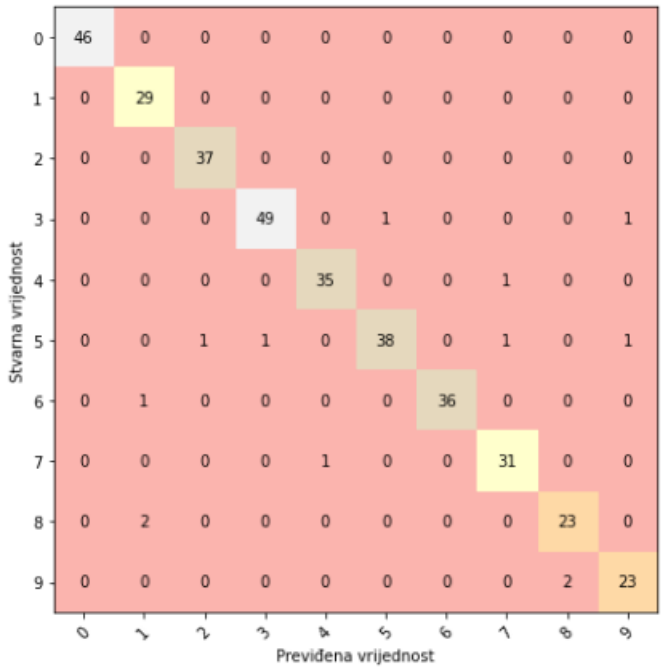
Skup podataka može se naći u biblioteci modula *sklearn* te sadrži 1797 slika formata 8x8, što znači da je skup ulaznih podataka *digits.data* matrica dimenzija 1797 x 64. Primjeri nekih slika

prikazani su na Slici 5.8. Ciljane vrijednosti, tj. brojke prikazane na pojedinim slikama, spremljene su u varijabli *digits.target* .

```
x_train, x_test, y_train, y_test = train_test_split(digits.data, digits.target, test_size=0.2)
logisticRegr = LogisticRegression(max_iter=1000)
logisticRegr.fit(x_train, y_train)
logisticRegr.predict(x_test[0].reshape(1,-1))
logisticRegr.predict(x_test[0:10])
```

Slika 5.9. Model logističke regresije na temelju ulaznih podataka

Cilj je trenirati logističku regresiju kako bi se predvidjeli brojevi na slikama. Kao i u prethodnom primjeru, skup podataka podijeli se na skupove za treniranje i testiranje. U ovom slučaju, funkcija dijeli podatke u omjeru 80:20, pri čemu se 80% podataka koristi za treniranje, a 20% za testiranje. Zatim se kreira objekt logisticRegr klase LogisticRegression s maksimalnim brojem iteracija 1000. Nakon toga, poziva se metoda fit nad logisticRegr kako bi se trenirao model na skupu za treniranje (x_train i y_train). Postupak je prikazan na Slici 5.9. Matrica konfuzije prikazana



Slika 5.10. Matrica konfuzije modela višestruke logističke regresije

je na Slici 5.10. Točnost doobivenog logističkog modela je 0.9638.

Klasifikacija korištenjem drugih metoda strojnog učenja

Stroj potpornih vektora (eng. *Support Vector Machine*) (SVM) je jedan od popularnih algoritama za klasifikaciju i regresiju koji se često koristi u strojnom učenju. SVM se temelji na ideji pronalaženja hiperravnine koja najbolje razdvaja različite klase podataka. U *Pythonu*, SVM je implementiran također u modulu scikit-learn (kao i logistička regresija), te sadrži različite funkcije i klase za rad sa SVM-om. Za klasifikaciju se obično koristi klasa SVC (*Support Vector Classification*), dok se za regresiju koristi klasa SVR (*Support Vector Regression*). Kod za izgradnju SVC modela prikazan je na Slici 5.11.

```
train_data, test_data, train_labels, test_labels = train_test_split(digits.data,
                                    digits.target, test_size=0.2, random_state=42)
svm = SVC(kernel='linear')
svm.fit(train_data.reshape(train_data.shape[0], -1), train_labels)
predictions = svm.predict(test_data.reshape(test_data.shape[0], -1))
accuracy = accuracy_score(test_labels, predictions)
print(f"Točnost SVR modela: {accuracy}")
```

Slika 5.11. Izgradnja SVC modela u Pythonu.

Stablo odluka (eng. *Decision Tree*) je algoritam koji izgradi stablo s pitanjima i pretpostavkama koja vode do klasifikacije ili predviđanja. U *Pythonu* postoji nekoliko modula koje pružaju implementaciju tog algoritma, a najčešće se koristi *scikit-learn*.

```
rf = RandomForestClassifier(n_estimators=100, random_state=0)
rf.fit(x_train, y_train)
y_pred = rf.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print('Točnost klasifikacije pomoću RandomForesta:', accuracy)
```

Slika 5.12. Izgradnja Random Forest modela u Pythonu.

Slučajna šuma (eng. *Random Forest*) je još jedan popularan algoritam strojnog učenja koji se koristi za klasifikaciju, regresiju i ostale zadatke. Radi se o algoritmu koji spaja više stabala odluka kako bi se stvorio jak model koji može predvidjeti izlazne vrijednosti na temelju ulaznih podataka. Svako stablo se trenira na dijelu podataka i koristi se kako bi se donijela odluka o klasi novog primjera. Nakon što su sva stabla izgrađena, klasa koju svako stablo predviđa se kombinira kako bi se donijela konačna odluka.

```
x_train = x_train.reshape(x_train.shape[0], 8, 8, 1)
x_test = x_test.reshape(x_test.shape[0], 8, 8, 1)
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
model = Sequential()
model.add(Conv2D(64, kernel_size=3, activation='relu', input_shape=(8,8,1)))
model.add(Conv2D(32, kernel_size=3, activation='relu'))
model.add(Flatten())
model.add(Dense(10, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=10)

accuracy = model.evaluate(x_test, y_test)[1]
print(f"Točnost pomoću CNNs: {accuracy}")
```

Slika 5.13. Izgradnja konvolucijske neuronske mreže u Pythonu.

Konvolucijske neuronske mreže (eng. *Convolutional Neural Networks*) (CNN) su vrsta neuronskih mreža koja se koristi u strojnom učenju za obradu i analizu slika, videa i zvukova. One su posebno dizajnirane za izdvajanje značajki iz ulaznih podataka korištenjem procesa konvolucije. U *Pythonu* postoji nekoliko biblioteka koje omogućuju izgradnju CNN-a, a najčešće se koriste *TensorFlow*, *Keras*, *PyTorch* i *scikit-learn*.

Ono što može biti argument za proučavanje logističke krivulje je činjenica da se kod neuronskih mreža vrlo često upravo logistička funkcija koristi kao aktivacijska funkcija, odnosno logistička

regresija se može promatrati kao najjednostavnija neuronska mreža s jednim slojem.

```
x_train, x_test, y_train, y_test = train_test_split(digits.data,
                    digits.target, test_size=0.25, random_state=0)
nb = GaussianNB()
nb.fit(x_train, y_train)
y_pred = nb.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print("Točnost pomoću Naive Bayes:", accuracy)
```

Slika 5.14. Izgradnja Naive Bayes modela u Pythonu.

Naivni Bayesov algoritam temelji se na Bayesovom teoremu i pretpostavlja da su značajke nezavisne jedna od druge. Naivni Bayesov algoritam se često koristi za klasifikaciju teksta, gdje se može primijeniti na različite probleme kao što su detekcija spam poruka, kategorizacija dokumenata ili za klasifikaciju slika, iako u manjoj mjeri.

```
clf = DecisionTreeClassifier()
clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)
print("Točnost pomoću Decision Tree", accuracy)
```

Slika 5.15. Programiranje modela pomoću algoritma Decision Tree u Pythonu.

Tablica 5.1. Usporedba točnosti korištenih algoritama za klasifikaciju slika.

Algoritam	**Točnost modela**
Višestruka logistička regresija	93.6%
SVM	97.7%
Slučajna šuma	97.2%
CNN	96.6%
Naivni Bayesov algoritam	83.3%
Stabla odluka	84.6%

Dobivene točnosti za pojedine algoritme prikazane su u tablici. Za klasifikaciju slika brojeva od 0 do 9 najveća točnost dobivena je pomoću algoritma SVR, iako je točnost algoritma slučajne šume jako blizu te vrijednosti. Točnost modela na nekim drugim skupovima podataka ne mora uvijek biti najveća s SVR algoritmom, već točnost u velikoj mjeri ovisi o ulaznim podacima.

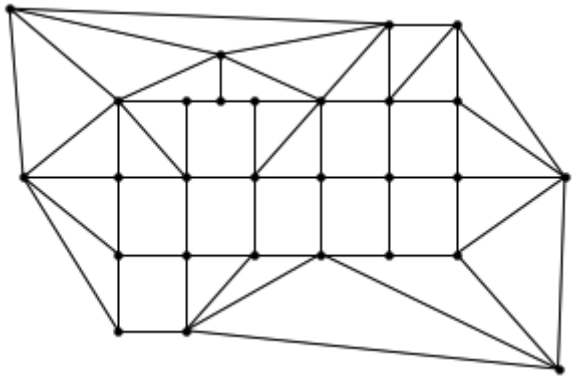
6. PRIMJENA LOGISTIČKE REGRESIJE U MEHANICI FLUIDA

U ovom poglavlju završnog rada primijenit će se *Water network tool for resilience (Wntr)* modul u programskom jeziku *Python*, kako bi se korištenjem simulacija dobili relevantni podaci za analizu vodoopskrbnog sustava. Glavni cilj ovog poglavlja jest provesti logističku regresiju na skupovima podataka dobivenih iteriranjem tlakova u vodoopskrbnom sustavu. Nakon provedene regresije, bit će moguće odrediti optimalne pozicije za postavljanje senzora u mreži, čime se može povećati učinkovitost i poboljšati praćenje rada sustava. Analiza ovakvog tipa uz pomoć metoda strojnog učenja u analizi vodoopskrbnih sustava u budućnosti bi mogla značajno utjecati na podizanje kvalitete vodoopskrbe i smanjenje gubitaka u distribucijskoj mreži.

Wntr je Python modul koji se koristi za modeliranje i simulaciju vodovodnih mreža. Ovaj modul služi za stvaranje i analizu vodovodnih mreža, kao i za provođenje simulacija u cjevovodima. Da bi se stvorila mreža cjevovoda u Wntr modulu, najprije je potrebno definirati topologiju mreže. To uključuje određivanje pozicija čvorova, cjevovoda i spojeva, kao i njihovih međusobnih veza i karakteristika.

Osnovne naredbe za definiranje mreže u Wntr modulu:

- wntr.network.WaterNetworkModel(): Stvara novu praznu mrežu cjevovoda.
- add_tank(): Dodaje spremnik u mrežu cjevovoda.
- add_pipe(): Dodaje cjevovod u mrežu cjevovoda.
- add_junction(): Dodaje spoj u mrežu cjevovoda.
- add_pump(): Dodaje crpku u mrežu cjevovoda.
- add_valve(): Dodaje ventil u mrežu cjevovoda.
- add_reservoir(): Dodaje ventil u mrežu cjevovoda.



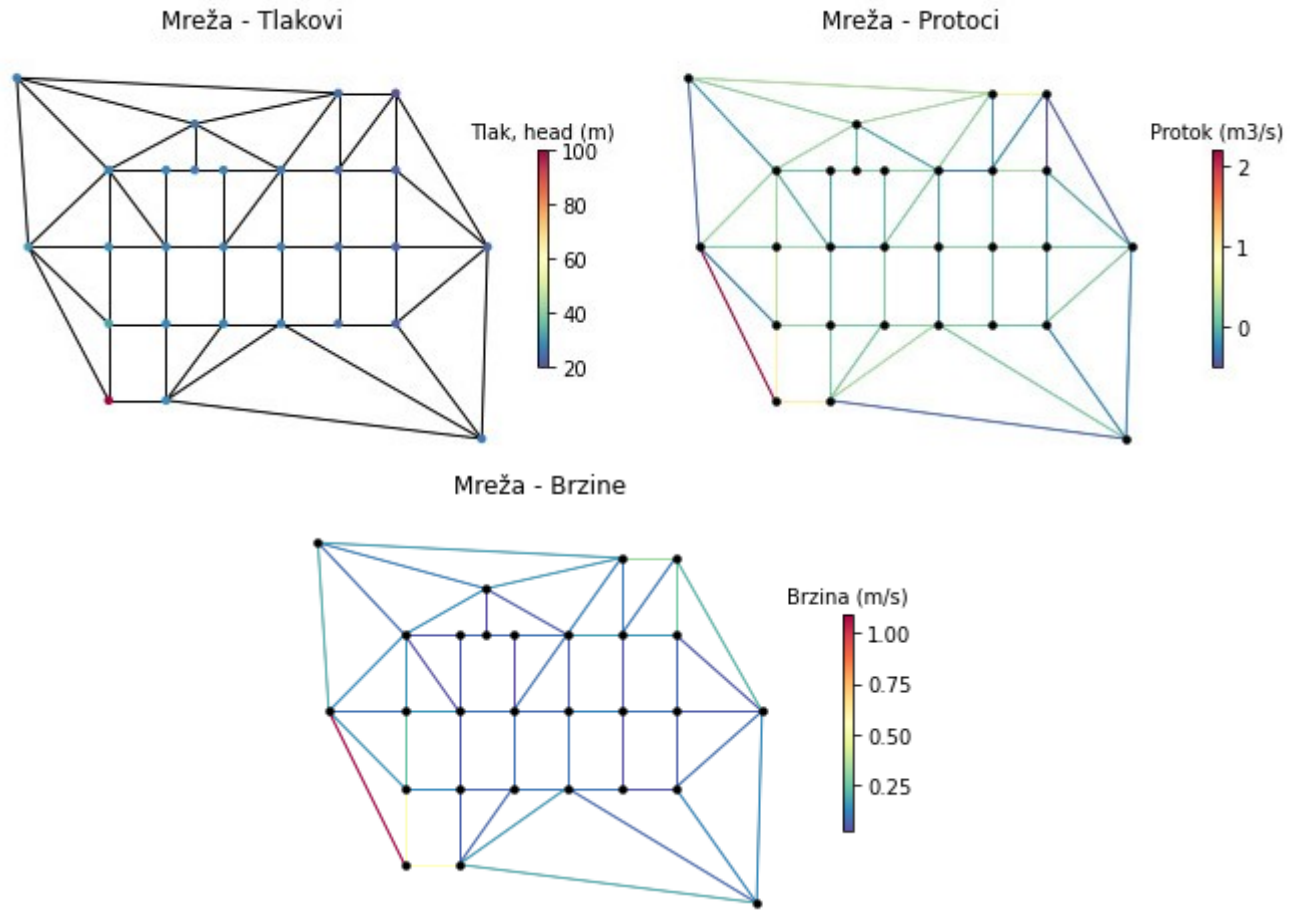
Slika 6.1. Kreirana vodoopskrbna mreža

Nakon stvaranja prazne mreže cjevovoda, proizvoljno će se kreirati vodoopskrbna mreža koja će se sastojati od 3 rezervoara, 26 spojeva i 60 cijevi. Tako kreirana vodoopskrbna mreža shematski

```
sim = wntr.sim.WNTRSimulator(wn)
results = sim.run_sim()

node_keys = results.node.keys()
link_keys = results.link.keys()
```

Slika 6.2. Pokretanje simulacije i rezultati u spojevima i cijevima

je prikazana na Slici 6.1. Nakon što je kreirana vodoopskrbna mreža može se pokrenuti simulacija se različitim potražnjama (base_demand) u svim čvorovima koje su jednake onim uobičajenim potražnjama (do 1 $[l/s]$).



Slika 6.3. Raspodjela tlakova, protoka i brzina u cjevovodima

Dobiveni rezultati mogu se prikazati i grafički, ovisno o kojoj se karakteristici radi (tlak, protok, brzina, visina,...), kao što je prikazano na Slici 6.3.

U ovom slučaju bila je provedena jedna simulacija. Naime, da bi se mogla koristiti logistička regresija potrebno je iterativno provesti simulacije i razmatrati dva slučaja. Prvi slučaj kada je potražnja u svim čvorovima u normalnom rasponu (do 1 $[l/s]$) i drugi slučaj kada se u određenom čvoru javlja curenje (potražnja raste znatno iznad 1 $[l/s]$). Provest će se veliki broj iteracija (2500 za svaki slučaj), te će se tlakovi za svaku iteraciju spremiti u zasebnu matricu koja će se potom koristiti za formiranje modela logističke regresije kojim će se moći odrediti u kojim je čvorovima najbolje postaviti senzore. Što se tiče samog modela logističke regresije tlakovi u pojedinom čvoru predstavljati će nezavisnu varijablu, odnosno ulaz kod modela logističke regresije. Izlaz modela

logističke regresije, odnosno zavisna varijabla ovisi o tome postoji li u mreži curenje (1) ili ne (0).

```
results = []
for i in range(N):
    neg_p = True
    print(f'sim {i}: ', end='')
    while neg_p:
        for node_name, node in wn.junctions():
            node.demand_timeseries_list[0].base_value = random.uniform(q_out[0], q_out[1])
        sim = wntr.sim.WNTRSimulator(wn)
        result = sim.run_sim()
        neg_p = np.any(result.node['pressure'] < 0)
        print('.', end='')

    print('done')
    results.append(result.node['pressure'])
df = pd.concat(results, axis=0)
```

Slika 6.4. Proračun za normalne uvjete u svim čvorovima

U kodu prikazanom na Slici 6.4 provodi se simulacija protoka u vodoopskrbnoj mreži N (1000) puta uz različite slučajne vrijednosti potražnje u izlaznom čvoru (sve su do 1 $[l/s]$), što predstavlja normalnu potražnju u pojedinom čvoru. Zatim se postavljaju slučajne vrijednosti za potražnju u svim čvorovima, pokreće se simulacija protoka vodovodne mreže i sprema se vrijednost tlakova u svim čvororvima u matricu "result". Dobivena matrica sadrži 1000 redaka sa podacima o tlakovima u pojedinim čvorovima pri normlanim uvjetima u vodoopskrbnoj mreži. Ovoj matrici nadodati će se još jedan stupac u kojoj će sve vrijednosti biti 0 što će kasnije predstavljati varijablu y kojom se formira model logističke regresije.

```
for i in range(N):
    neg_p = True
    print(f'sim {i}: ', end='')
    while neg_p:
        for node_name, node in wn.junctions():
            node.demand_timeseries_list[0].base_value = random.uniform(q_out[0], q_out[1])
        random_node_name = random.choice(list(wn.junction_name_list))
        wn.get_node(random_node_name).demand_timeseries_list[0].base_value = random.uniform
        (q_leak[0], q_leak[1])
        sim = wntr.sim.WNTRSimulator(wn)
        result = sim.run_sim()
        neg_p = np.any(result.node['pressure'] < 0)
        print('.', end='')

    print('done')
    results.append(result.node['pressure'])
```

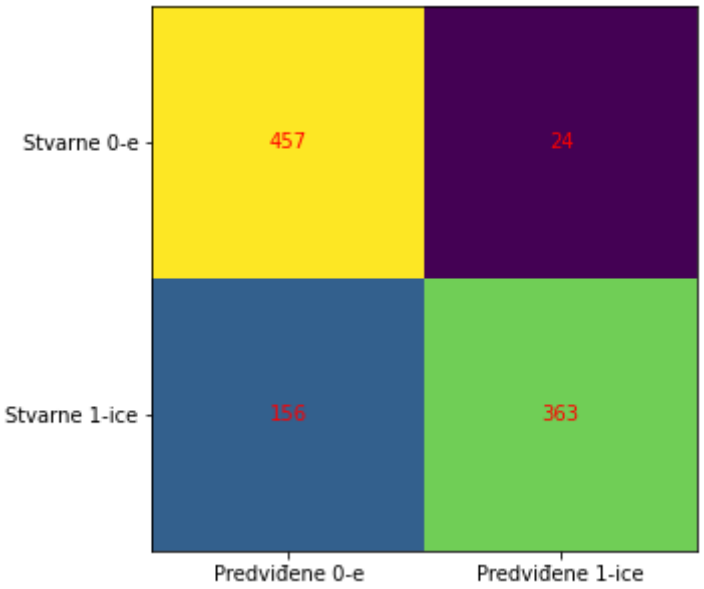
Slika 6.5. Proračun za situaciju s curenjem u slučajno odabranom čvoru

Kod na Slici 6.5 sličaj je prethodnome, samo što se ovdje uzima jedan nasumičan čvor i na njemu se definira puno veća potražnja od normalne (3-4 $[l/s]$), čime se simulira curenje u odabranom čvoru. U ostalim čvorovima potražnja je i dalje normalna (do 1 $[l/s]$). Ovime je dobivena matrica koja sadrži tlakove u pojedinim čvorovima u slučaju curenja u vodoopskrbnoj mreži. Na dobivenu matricu dodati će se vrijednost 1 što će predstavljati oznaku da u vodopskrbnoj mreži postoji curenje. Dobivena se matrica spaja sa matricom iz prethodnog koda i time su dobiveni

```
X = df.to_numpy()[:, :]
y = np.zeros(2 * N)
y[N:] = 1

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = LogisticRegression(max_iter=1200)
model.fit(X_train, y_train)
probas = model.predict_proba(X_test)
```

Slika 6.6. Formiranje modela logističke regresije

```
cm = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots(figsize=(6, 5))
ax.imshow(cm)
ax.grid(False)
ax.xaxis.set(ticks=(0, 1), ticklabels=('Predviđene 0-e', 'Predviđene 1-ice'))
ax.yaxis.set(ticks=(0, 1), ticklabels=('Stvarne 0-e', 'Stvarne 1-ice'))
ax.set_ylim(1.5, -0.5)
for i in range(2):
    for j in range(2):
        ax.text(j, i, cm[i, j], ha='center', va='center', color='red')
plt.show()
```



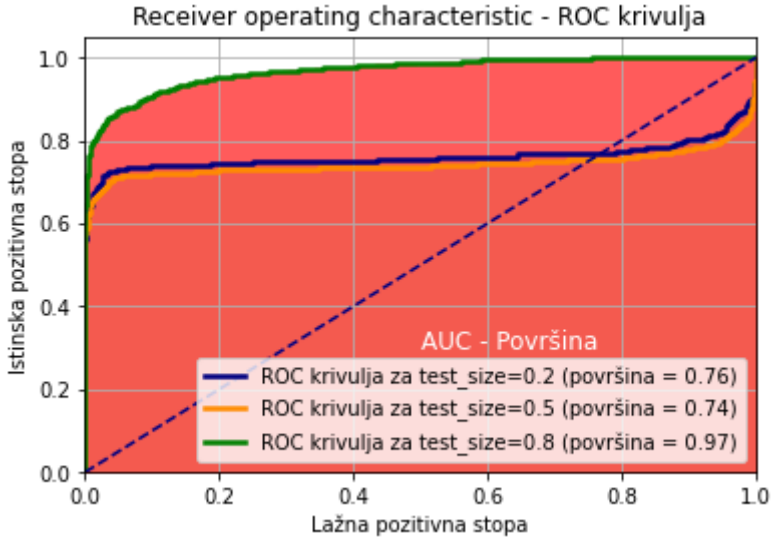
Slika 6.7. Definiranje matrice konfuzije i njezin prikaz

podaci potrebni za formiranje modela logističke regresije. Nakon što je formiran model logističke regresije (Slika 6.6) za tlakove u vodoopskrbnoj mreži može se odrediti matrica konfuzije (Slika 6.7) . Osim matrice konfuzije može se odrediti i točnost modela binarne logističke regresije (model

	precision	recall	f1-score	support
0.0	0.75	0.96	0.84	481
1.0	0.95	0.70	0.81	519
accuracy			0.83	1000
macro avg	0.85	0.83	0.83	1000
weighted avg	0.86	0.83	0.83	1000

Slika 6.8. Podaci o točnosti modela logističke regresije

je binaran jer ima 2 zavisne varijable, odnosno 0 predstavlja normalnu raspodjelu tlakova u mreži cjevovoda, dok 1 predstavlja da u mreži cjevovoda postoji curenje). Točnost u promatranom slučaju iznoci 0.85. Osim točnosti za ovaj model se, pomoću naredbe `classification_report`, određuju odziv i osjetljivost (Slika 6.8). Do sada su bile određene temeljne karakteristike kojima se obično određuje točnost modela logističke regresije. Preostaje još odrediti ROC krivulju. Osim same površine ispod ROC krivulje koja procjenjuje točnost izvedbe modela binarne klasifikacije, ona će biti potrebna kasnije za određivanje lokacije senzora. ROC krivulje u promatranom primjeru



Slika 6.9. ROC krivulja za različite testne vrijednosti

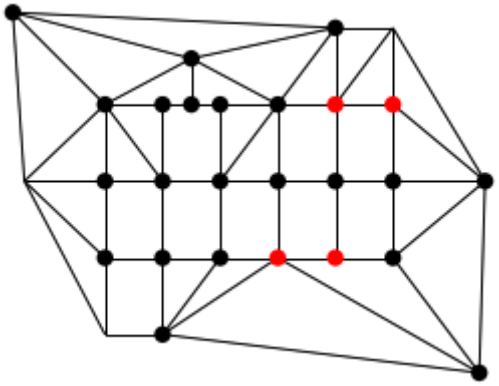
dobivene korištenjem različite veličine skupova za testiranje prikazane su na Slici 6.9.

U nastavku se opisuje postupak određivanja najpovoljnijih lokacija za postavljanje senzora. Odredit će se četiri najpovoljnije lokacije u vodoopskrbnoj mreži u kojima će se postaviti senzori. To će se učiniti na način da će se za svaki čvor (ukupno 26) definira model logističke regresije koristeći podatke samo u tom čvoru i pripadajuće izlazne vrijednosti. Takav će se postupak provesti za sve čvorove (Slika 6.10). Nakon dobivanja modela logističke regresije, kao mjera točnosti za pojedini čvor koristit će se površina ispod ROC krivulje. Četiri najpovoljnije lokacije bit će oni čvorovi kojima pripada logistički model s najvećom površinom ispod ROC krivulje. Naime, može se smatrati se da će podaci najbolje biti klasificirani korištenjem logističke regresije koja je bazirana na proračunima u tim čvorovima. To može značiti da je utjecaj tlakova u tim čvorovima na vodoopskrbnu mrežu najveći, pa ima smisla upravo u njima postaviti senzore. Čvorovi u kojima se nalaze senzori tlaka označeni su crvenom bojom (Slika 6.11). Cjelokupna mreža s pripadajućim AUC vrijednostima modela logističke regresije za svaki čvor prikazana je na Slici 6.12.

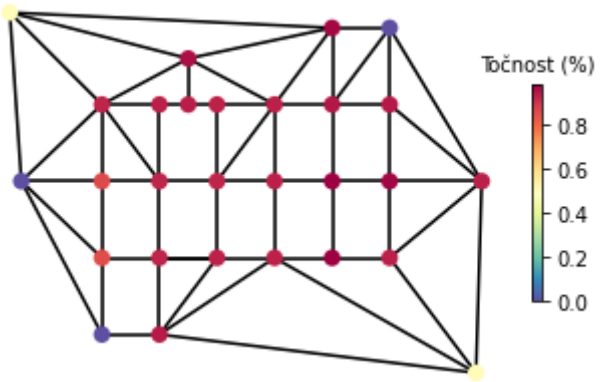
U modelima logističke regresije u ovome primjeru mogli su se umjesto tlaka u čvorovima, kao nezavisne varijable mogli koristiti protoci u cijevima. U tom slučaju bi umjesto 26 nezavisnih varijabli (jer je 26 čvorova), bilo 60 nezavisnih varijabli s obzirom da u promatranoj mreži postoji 60 cijevi. Koristeći prethodno opisanu metodologiju, odredile su se najpovoljnije pozicije za postavljanje senzora protoka, te su iste prikazane na Slici 6.13.

```
roc_x1 = []
for i in range(X.shape[1]):
    X1 = X[:, i].reshape(-1, 1)
    X_train1, X_test1, y_train1, y_test1 = train_test_split(X1, y, test_size=0.2)
    logreg_1 = LogisticRegression(max_iter=1200)
    logreg_1.fit(X_train1, y_train1)
    y_pred = logreg_1.predict(X_test1)
    y_score_1 = logreg_1.decision_function(X_test1)
    fpr_1, tpr_1, _ = roc_curve(y_test1, y_score_1)
    roc_auc_1 = auc(fpr_1, tpr_1)
    auc_score = roc_auc_score(y_test1, y_score_1)
    roc_x1.append(auc_score)
roc_x1 = np.concatenate((roc_x1, np.zeros((3,))))
indeksi = np.argsort(roc_x1)

senzor1 = np.argmax(roc_x1)+1
senzor2 = indeksi[-2]+1
senzor3 = indeksi[-3]+1
senzor4 = indeksi[-4]+1
```

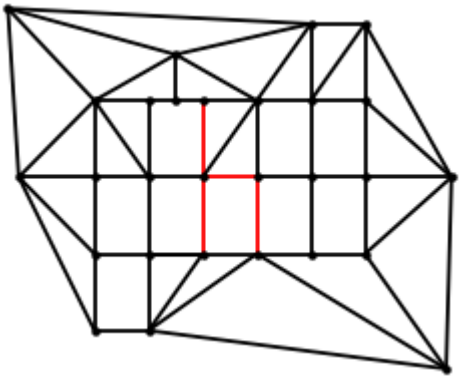
Slika 6.10. Računanje površina ispod ROC krivulje za svaku logističku regresiju



Slika 6.11. Grafički prikaz čvorova u kojima se predlaže postavljanje senzora tlaka



Slika 6.12. Točnosti čvorova temeljena na ROC krivuljama



Slika 6.13. Grafički prikaz cijevi u kojima se predlaže postavljanje senzora protoka

7. Zaključak

U ovom završnom radu provedena je analiza primjene logističke regresije u inženjerstvu s ciljem proučavanja njenih mogučnosti i prednosti u različitim inženjerskim područjima. Logistička regresija je statistička metoda koja se koristi za modeliranje ovisnosti između kategorijske ovisne varijable i jedne ili više nezavisnih varijabli. U ovom radu svi primjeri izvedeni su pomoću programskog jezika *Python*.

U radu su prikazani osnovni koncepti logističke regresije i opisani različiti pristupi evaluaciji performansi logističke regresije, kao što su matrica konfuzije, točnost modela, preciznost, osjetljivost i ROC krivulja.

Prikazana je usporedba logističke regresije u odnosu na druge algoritme u stojnom učenju kao što su stabla odluka, SVR, slučajna šuma, konvolucijske neuronske mreže i naivni Bayesov algoritam.

Također, istaknute su i prednosti i ograničenja logističke regresije. Prednosti uključuju jednostavnost implementacije, interpretabilnost rezultata i mogučnost rada s kategorijskim varijablama. Prednost je i to što se mogu koristiti različiti tipovi podataka i slike, pa se logistička regresija može primijenjivati u raznim inženjerskim podruèjima.

Ipak, postoje i neka ograničenja logističke regresije. Ova metoda pretpostavlja linearnu vezu između ulaznih varijabli i logaritma izgleda vjerojatnosti, što može biti problematično u slučajevima kada je ta veza zapravo nelinearna. Također, logistička regresija može biti osjetljiva na prisutnost ekstremnih vrijednosti u podacima (*outliersa*). Stoga, pri primjeni logističke regresije važno je provesti odgovarajuću analizu podataka, uključujući prethodnu obradu i odabir relevantnih varijabli.

Ukratko, ovaj završni rad pruža uvid u primjenu logističke regresije u inženjerstvu. Metoda se pokazala korisnom za rješavanje različitih problema u tom području, ali je važno pažljivo prikupljati i obraditi podatke kako bi rezultati bili pouzdani. Ovo istraživanje može poslužiti kao temelj za daljnje istraživanje i primjenu logističke regresije u inženjerskim disciplinama.

8. Literatura

[1] https://en.wikipedia.org/wiki/Logistic_regression, 10. ožujak 2023.

[2] https://realpython.com/logistic-regression-python/, 10. ožujak 2023.

[3] https://towardsdatascience.com/machine-learning-basics-logistic-regression-890ef5e3a272, 10. ožujak 2023.

[4] https://www.turing.com/kb/auc-roc-curves-and-their-usage-for-classification-in-python, 10. ožujak 2023.

[5] https://www.javatpoint.com/linear-regression-vs-logistic-regression-in-machine-learning, 10. ožujak 2023.

[6] https://www.simplilearn.com/tutorials/machine-learning-tutorial/linear-regression-vs-logistic-regression, 23. ožujak 2023.

[7] https://www.kdnuggets.com/2022/03/linear-logistic-regression-succinct-explanation.html, 23. ožujak 2023.

[8] https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, 3. travanj 2023.

[9] https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488, 3. travanj 2023.

[10] https://en.wikipedia.org/wiki/Precision_and_recall, 11. travanj 2023.

[11] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html, 11. travanj 2023.

[12] Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning - Second edition", 20. travanj 2023.

[13] https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/, 20. travanj 2023.

[14] https://scikit-learn.org/stable/modules/tree.html, 28. travanj 2023.

[15] https://en.wikipedia.org/wiki/Random_forest, 2. svibanj 2023.

[16] https://en.wikipedia.org/wiki/Convolutional_neural_network, 17. svibanj 2023.

[17] https://en.wikipedia.org/wiki/Naive_Bayes_classifier, 17. svibanj 2023.

[18] https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia, 17. svibanj 2023.

[19] https://wntr.readthedocs.io/en/latest/overview.html, 13. lipanj 2023.

[20] https://github.com/USEPA/WNTR, 13. lipanj 2023.

[21] https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5, 13. lipanj 2023.

[22] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc, 13. lipanj 2023.

POPIS SLIKA

SAŽETAK I KLJUČNE RIJEČI

Sažetak

U ovom radu logistička regresija primjenjena je u sklopu programskog jezika Python na razne skupove podataka. Također je opisana binarna logistička regresija i višekategorijska logistička regresija. Objašnjen je postupak učenja modela s osvrtom na implementaciju u programskom jeziku Python, te osnovna značenja parametara. Jedan od ključnih aspekata rada je analiza točnosti modela temeljena na ulaznim podacima. U tu svrhu su objašnjene osnovne karakteristike koje se koriste za opisivanje točnosti modela, kao što su matrica konfuzije, preciznost, osjetljivost i ROC-krivulja. Ove metrike omogučuju procjenu performansi logističke regresije u detekciji i klasifikaciji problema. Nadalje, provedena je usporedba logističke regresije s drugim algoritmima koji se koriste u strojnom učenju. Ova usporedba ima za cilj prikazati prednosti i nedostatke logistièke regresije u odnosu na druge algoritme, te ukazati na situacije u kojima je logistička regresija najpogodnija za primjenu u inženjerskim problemima. U zaključku rada ističe se kako je logistička regresija jako močan alat za inženjere. Njena primjena može pomoći u rješavanju različitih inženjerskih problema, poput predviđanja kvarova u nekom vodoopskrbnom sustavu.

Abstract

In this work, logistic regression was applied within the Python programming language to various data sets. Binary logistic regression and multicategory logistic regression are also described. The learning process of the model is explained with reference to the implementation in the Python programming language, and the basic meaning of the parameters. One of the key aspects of the work is the analysis of model accuracy based on input data. For this purpose, the basic characteristics used to describe the accuracy of the model are explained, such as the matrix of confusion, precision, sensitivity and ROC-curve. These metrics allow evaluating the performance of logistic regression in problem detection and classification. Furthermore, logistic regression was compared with other algorithms used in machine learning. This comparison aims to show the advantages and disadvantages of logistic regression compared to other algorithms, and to point out the situations in which logistic regression is most suitable for application in engineering problems. In the conclusion of the paper, it is emphasized that logistic regression is a very powerful tool for engineers. Its application can help solve various engineering problems, such as predicting failures in a water supply system.

Ključne riječi

Logistička regresija, binarna logistička regresija, višestruka logistička regresija, Python, matrica konfuzije, ROC krivulja, precision, recall, klasifikacija, strojno učenje.

Keywords

Logistic regression, binary logistic regression, multiple logistic regression, Python, confusion matrix, ROC curve, precision, recall, classification, machine learning.