Dijagnostika karcinoma pluća korištenjem metoda umjetne inteligencije

Režek, Anja

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:190:787882

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2025-04-02**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Engineering

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

DIJAGNOSTIKA KARCINOMA PLUĆA KORIŠTENJEM METODA UMJETNE INTELIGENCIJE

Rijeka, studeni 2022.

Anja Režek
0069077305

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

DIJAGNOSTIKA KARCINOMA PLUĆA KORIŠTENJEM METODA UMJETNE INTELIGENCIJE

Mentor: prof. dr. sc. Zlatan Car

Rijeka, studeni 2020.

Anja Režek
0069077305

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za automatiku i elektroniku**
Predmet: **Primjena umjetne inteligencije**
Grana: **2.03.06 automatizacija i robotika**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Anja Režek (0069077305)**
Studij: Diplomski sveučilišni studij elektrotehnike
Modul: Automatika

Zadatak: **Dijagnostika karcinoma pluća korištenjem metoda umjetne inteligencije/Lung cancer diagnostics using artificial intelligence methods**

Opis zadatka:

Izraditi pregled literature u području dijagnostike bolesti, posebice karcinoma i plućnih oboljenja primjenom umjetne inteligencije. Dati opis korištenog seta podataka. Primijeniti metode augmentacije za kategorijske setova podataka. Usporediti performanse augmentiranih i neaugmentiranih setova podataka. Komentirati utjecaj hiperparametara na modele.

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

Režek

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Prof. dr. sc. Zlatan Car

Predsjednik povjerenstva za diplomski ispit:

Prof. dr. sc. Viktor Sučić

IZJAVA

Sukladno članku 8. Pravilnika o diplomskom radu, diplomskom ispitu i završetku diplomskih sveučilišnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 31. siječnja 2020., izjavljujem da sam samostalno izradio/izradila diplomski rad prema zadatku preuzetom dana 21.ožujka 2022.

Rijeka, studeni 2022.

Anja Režek

Ovom prilikom želim se zahvaliti prof. dr. sc. Zlatanu Caru na prihvaćanju mentorstva za izradu rada i asistentu Sandiju Baressi Šegoti, mag. ing. comp. koji mi je svojom strpljivošću, znanjem i nesebičnom suradnjom pomogao u pisanju diplomskog rada.

Hvala svim mojim kolegama i prijateljima na pruženoj podršci tokom cijelog studiranja i koji su mi bitno olakšali isto.

Najveću zahvalu upućujem svojoj obitelji bez čije potpore i vjere cilj nebi bio ostvariv, koji su me bodrili u najtežim trenucima studiranja i davali mi snagu za dalje.

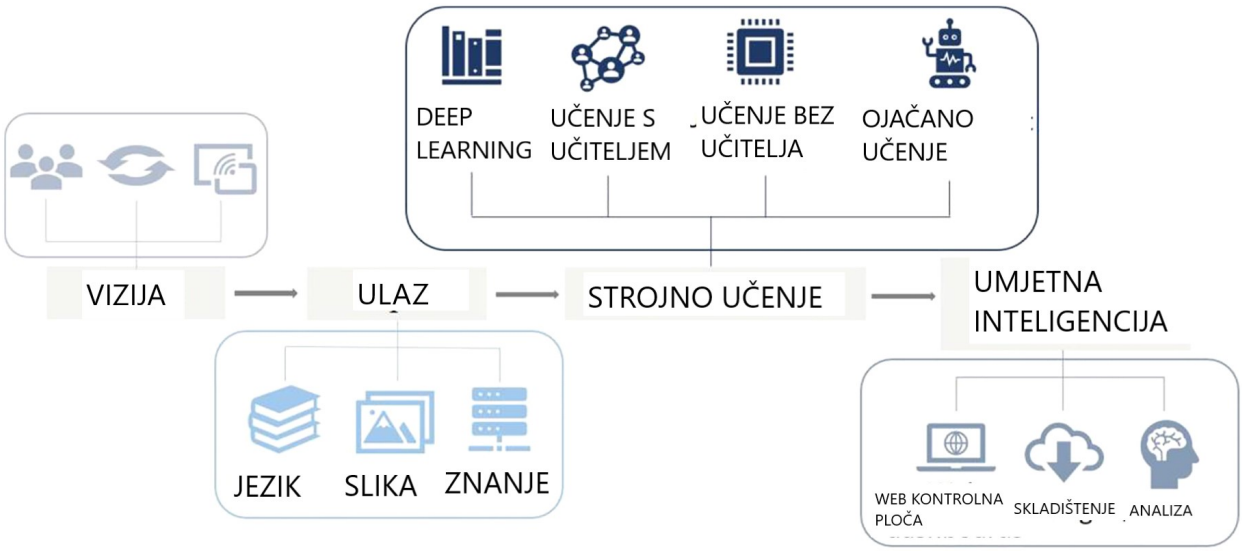
Sadržaj

1. Uvod

Rak pluća jedan je od najčešćih malignih tumora s najbržim porastom morbiditeta i mortaliteta. Preživljenje pacijenata s rakom pluća 5 godina nakon postavljanja dijagnoze je samo 10-20% u većini zemalja zbog činjenice da je većina dijagnosticiranih karcinoma pluća u srednjem i kasnom stadiju bolesti i da su metode liječenja ograničene. Rak pluća klinički se klasificira uglavnom prema histopatologiji, koja se može podijeliti na rak pluća nemalih stanica i rak pluća malih stanica. Većina karcinoma pluća klasificira se kao rak pluća nemalih stanica, što čini oko 85-90%, uključujući karcinom velikih stanica. Međunarodna udruga za istraživanje raka pluća (IASLC) određuje stadij raka pluća prema kriterijima promjera tumora, metastaza u limfnim čvorovima i udaljenih metastaza, a rak pluća dijeli na stadije I-IV od kojih je stadij I-II rani stadij, a stadij III-IV je uznapredovali rak pluća. Većina karcinoma pluća obično se dijagnosticira u uznapredovalom stadiju i može biti povezana s lošom prognozom. Osim toga, ograničenja odabira liječenja i procjene prognoze također su donijela izazove.

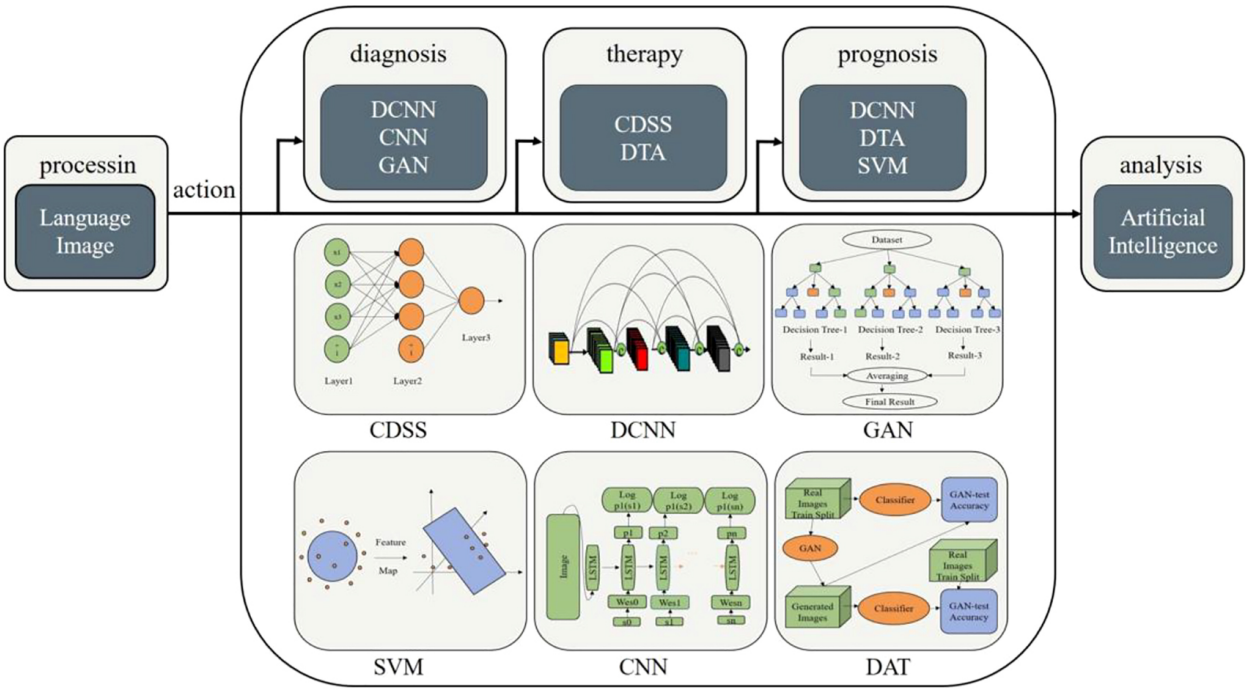
Do sada se dijagnoza raka pluća uglavnom oslanja na kompjutoriziranu tomografiju (CT) i biopsiju tkiva. Obje dijagnostičke metode imaju mana - CT-om je lako napraviti pogrešnu dijagnozu, a biopsija tkiva je invazivna metoda U međuvremenu, potrebno je poboljšati osjetljivost i specifičnost neinvazivnih biomarkera za rak pluća. Štoviše, lokacija tumora, patološki tip, prisutnost metastaza i komplikacija otežavaju dijagnozu što rezultira time da više od polovice bolesnika s rakom pluća ima metastaze u vrijeme dijagnoze. Uobičajeno liječenje raka pluća je kirurška resekcija i kemoterapija. Klinički, liječenje se uglavnom odabire prema histopatološkoj klasifikaciji. Međutim, uz trenutna tehnička sredstva, kirurška trauma je prevelika, a ciljna funkcija kemoterapijskih lijekova nije idealna. Nadalje, vrijeme dijagnoze raka pluća, trauma kirurškog liječenja, otpornost na lijekove za kemoterapiju, metastaze i drugi čimbenici čine prognozu bolesnika teškom za procjenu. Za smanjenje smrtnosti oboljelih od karcinoma pluća vrlo je važna točna rana dijagnoza i pravodobno liječenje.

Umjetna inteligencija (eng. artificial intelligence - AI) je novo područje studija koje razvija teorije, metode, tehnologije i aplikacijske sustave za simulaciju i proširenje ljudske inteligencije. Tehnički sustav umjetne inteligencije može se sažeti u četiri modula: obrada prirodnog jezika, prepoznavanje slike, interakcija između čovjeka i računala i strojno učenje kao što je prikazano na slici 1.1.



Slika 1.1. Klasifikacija AI

Obrada prirodnog jezika integrira lingvistiku, informatiku, matematiku i druge discipline i uglavnom proučava računalne sustave koji mogu implementirati komunikaciju prirodnim jezikom, što uključuje pronalaženje informacija, ekstrakciju informacija, označavanje dijela govora, sintaktičku analizu, prepoznavanje govora, gramatičku analizu, prijevod jezika i dalje. Tehnologija obrade slike uključuje prikupljanje slike, filtriranje i podešavanje slike, ekstrakciju značajki. Interaktivna tehnologija čovjek-računalo pretvorba je obrade prirodnog jezika i prepoznavanja slika. To prvenstveno uključuje računalnu grafiku, interaktivni dizajn sučelja, proširenu stvarnost i tako dalje. Strojno učenje uglavnom uključuje nadzirano učenje (zadatak klasifikacije i regresije), nenadzirano učenje, prijenosno učenje, učenje s potkrepljenjem i integrirano učenje. Njegovi reprezentativni algoritmi obuhvaćaju duboko učenje, umjetnu neuronsku mrežu, stablo odlučivanja, algoritam poboljšanja itd. Sve u svemu, točna dijagnoza i prognoza ključni su u odabiru i planiranju liječenja raka pluća. S brzim napretkom tehnologije medicinskog oslikavanja, oslikavanje cijelog dijapozitiva (WSI) u patologiji postaje rutinski klinički postupak. Nedavno je umjetna inteligencija, posebice duboko učenje, pokazala veliki potencijal u zadacima analize patoloških slika kao što su identifikacija regije tumora, predviđanje prognoze, karakterizacija mikrookruženja tumora i otkrivanje metastaza. Na slici 1.2 prikazan je dijagram funkcija umjetne inteligencije u dijagnostici, liječenju i prognozi raka pluća.



Slika 1.2. Dijagram funkcije AI u dijagnostici, liječenju i prognozi raka pluća [1]

2. Set podataka

Korišteni set podataka dan je u Tablici 2.1.

Tablica 2.1. Set podataka

Karakteristike skupa podataka:	Multivarijantni
Broj slučajeva:	32
Područje:	Život
Karakteristike atributa:	Cijeli broj
Broj atributa:	56
Datum doniranja:	1992-05-01
Povezani zadaci:	Klasifikacija
Nedostaju vrijednosti?	Da
Broj posjeta webu:	388896

Ove su podatke upotrijebili Hong i Young kako bi ilustrirali snagu optimalne diskriminantne ravnine čak i u loše postavljenim postavkama. Primjena KNN metode u rezultirajućoj ravnini dala je 77% točnosti. Međutim, ti su rezultati jako pristrani. Podaci su opisali 3 tipa patoloških karcinoma pluća. Autori ne daju podatke o pojedinačnim varijablama niti o tome gdje su podaci izvorno korišteni.

Koristi se prepoznavanje patoloških tipova raka pluća kao primjer za testiranje teorije. Eksperimentalni sadržaji uključuju perturbaciju rješenja i ocjenu učinka klasifikatora. Prepoznavanje patoloških tipova raka pluća problem je prepoznavanja uzoraka od tri klase i 56 dimenzija. Postoji 56 parametara značajki izdvojenih iz kliničkih podataka i podataka rendgenskih snimaka, itd. Trideset i dva uzorka dana su u Tablici 2.2. Treba napomenuti da je broj danih uzoraka daleko manji od dimenzionalnosti prostora značajki. Uzimaju se 32 uzorka kao uzorci za treniranje. Nedefiniranih vrijednosti ima 5.

Pojedinačne varijable izdvojene su korištenjem prilagođene metode obrade značajki koju su izveli autori. Svaka od varijabli u skupu podataka pojedinačna je značajka izvedena iz algoritma koji se temelji na SIFT i SURF obradi slika radiografskih snimaka pluća, napravljenih na pacijentima s dijagnosticiranim rakom pluća, u neoptimalnim postavkama radioloških varijabli, kao što su intenzitet i kontrast. Značajke su normalizirane na raspone dane u skupu podataka. Glavna svrha skupa podataka je utvrditi mogućnost korištenja metoda umjetne inteligencije u slučajevima kada su korištene loše postavljene postavke.

Tablica 2.2. Skup uzoraka raka pluća [2]

	Klasa I									Klasa II													Klasa III									
No./ Xi	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	3	3	3	2	3	3	3	2	3	2	2	2	2	2	3	3	3	2	2	1	2	3	2	2	3	2	2	2	3	2	2	2
3	0	3	3	3	2	3	2	2	1	3	2	3	1	2	0	2	2	1	0	2	0	3	3	3	3	3	2	2	2	1	3	3
4	-1	1	2	2	1	2	1	1	1	2	0	2	1	1	-1	2	2	1	-1	1	-1	2	1	1	1	2	2	1	2	1	2	1
5	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0
6	2	3	3	3	3	3	3	3	3	2	3	1	1	2	1	2	1	2	2	3	3	2	2	1	3	1	2	2	2	2	2	2
7	2	1	3	3	3	3	3	1	1	2	2	2	2	3	2	2	1	1	3	3	3	1	2	2	3	2	1	2	2	2	2	3
8	2	3	3	3	3	3	3	3	3	2	3	1	2	3	2	2	3	3	3	3	3	3	1	1	1	2	2	2	2	1	3	3
9	1	1	1	1	2	1	1	3	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	2	2	2	2	3	1	2	1	2	2	1	1	1	1	1	2	2	2	1	1	1	1	2	1	1	1	1	1	1
11	1	1	1	1	2	2	1	3	1	2	3	1	1	1	1	2	1	1	1	2	1	3	1	2	2	1	1	2	2	1	2	1
12	1	1	0	0	1	0	2	2	3	2	1	0	1	1	1	1	0	0	0	1	1	3	1	1	1	2	0	0	0	0	1	1
13	3	1	3	3	1	3	3	1	2	2	3	1	2	2	2	2	1	3	2	1	3	3	1	1	1	1	2	2	2	1	2	2
14	2	3	3	3	2	3	3	3	3	2	1	2	2	2	1	3	3	1	2	3	3	3	2	1	2	1	2	2	2	1	2	2
15	2	3	3	1	2	3	3	3	3	2	1	2	2	2	1	3	3	3	1	3	3	3	2	2	2	1	3	2	2	1	2	2
16	1	1	1	1	2	2	3	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	1	2	2	2	1	1	1	2	1	1
17	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	1	2	1	2	2	2	1	2	2
18	2	2	1	2	2	1	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2
19	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	2	1	2	2	2	2	2	0	0	0
20	2	0	0	0	0	0	0	0	0	2	2	2	2	1	2	1	2	0	0	0	0	0	2	2	2	1	2	2	2	2	2	2
21	2	2	2	2	2	1	2	1	2	2	1	2	2	2	2	2	2	1	2	2	2	2	1	1	1	2	2	2	2	1	2	2
22	2	2	2	2	2	2	2	1	2	2	1	2	2	2	2	2	2	2	1	2	2	2	2	2	1	1	2	2	2	1	2	2
23	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	2	2	2	1	1	1	2
24	1	1	1	2	1	2	2	1	1	3	1	1	1	1	1	1	1	3	1	1	2	1	1	1	1	2	3	1	1	1	1	1
25	2	2	2	1	1	1	1	2	2	2	1	2	2	2	2	2	1	3	2	2	1	2	1	2	1	1	3	2	2	1	2	2
26	2	1	2	3	2	3	3	1	1	3	2	1	1	1	2	3	2	3	3	1	1	2	1	2	2	3	3	1	2	2	2	1
27	2	3	3	2	3	3	2	3	2	3	2	2	3	2	2	2	2	2	2	3	2	3	2	2	2	2	2	3	2	2	2	2
28	3	2	2	3	2	1	2	2	1	3	2	2	2	2	2	2	2	2	2	2	3	3	2	2	2	1	2	3	2	1	2	2
29	2	3	3	3	2	2	2	1	1	3	2	3	2	2	2	1	2	2	3	3	2	3	3	2	2	1	1	3	2	2	2	2
30	1	1	1	1	1	2	2	1	1	3	1	1	1	1	3	2	1	1	1	1	1	3	1	1	3	2	2	1	2	1	1	1
31	1	1	3	3	1	3	3	3	3	3	3	3	1	1	3	2	3	3	3	1	1	3	3	3	3	2	2	3	1	3	2	1
32	1	1	3	3	1	3	3	3	3	3	3	3	1	1	3	2	3	3	3	1	1	3	3	3	3	2	2	3	1	3	2	1
33	3	3	3	3	3	3	3	3	3	3	3	3	1	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	1	3	3	1
34	3	3	1	1	2	2	2	2	3	2	1	1	1	2	1	1	3	1	2	1	1	2	1	2	2	1	3	2	2	1	2	1
35	2	2	2	1	1	1	1	2	2	1	2	2	2	1	1	3	3	1	1	1	3	2	1	2	1	1	1	3	1	1	2	2
36	2	2	2	1	2	2	1	1	2	2	2	2	2	1	2	2	1	1	1	2	1	2	1	2	2	2	2	1	1	3	2	2
37	1	2	1	1	2	2	2	2	1	1	1	1	1	2	1	1	2	2	2	2	1	1	3	3	2	2	2	2	3	3	1	1
38	2	1	2	2	1	1	2	1	2	2	2	2	2	1	2	2	2	1	1	1	2	2	1	3	2	4	2	1	1	3	2	2
39	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	1	2	2	2	2	1	3	1	2	2	2	3	3	2	2	2
40	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	1	2	1	1	2	3	3	2	2
41	1	2	1	2	2	2	1	1	1	2	1	2	1	2	2	1	1	2	2	2	1	1	3	2	2	1	2	2	3	1	1	1
42	2	1	2	1	1	1	2	2	2	2	2	1	2	1	1	2	2	1	3	1	2	2	3	2	2	1	1	3	2	2	2	2
43	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	2	2	1	2	3	2	2	2
44	2	2	1	2	2	2	2	2	1	2	2	2	2	2	2	1	2	1	2	2	1	2	3	2	2	2	2	2	2	3	2	1
45	2	1	2	2	2	2	2	1	2	2	2	2	2	1	2	2	2	2	2	2	2	2	3	2	2	2	2	2	2	3	2	2
46	2	2	2	1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2	2	1	1	3	2	2	2	2
47	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2	2	2	2	3	2	2	2	2
48	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2	2	2	2	3	2	2	2	2
49	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	1	3	2	2	2	2	2	2	3	2	2
50	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	3	2	2	1	2	2	3	2	2	2
51	2	2	2	2	2	2	1	2	2	3	2	2	2	2	2	2	2	2	2	2	1	2	1	2	3	3	2	3	1	2	2	2
52	1	2	2	2	2	2	2	2	2	2	2	1	2	1	2	2	2	1	2	1	2	2	3	2	2	1	2	2	2	2	2	2
53	1	2	2	2	2	2	2	2	2	1	2	1	2	1	2	2	2	1	2	1	2	2	3	2	2	2	2	2	2	2	2	2
54	1	1	2	2	1	2	2	1	1	1	2	2	1	1	1	2	1	1	2	2	1	2	2	2	2	2	1	2	2	2	1	2
55	2	2	1	2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	1	2	2	1	2	2	2	1	2	2	2	2	2	2
56	2	2	2	2	2	2	2	2	2	2	2	1	2	2	1	2	2	2	2	1	2	2	1	1	1	2	1	1	2	1	2	2

2.1. Histogram atributa

Na slici 2.1 prikazani su histogrami svih 55 atributa te će u nastavku svaki pojedini histogram biti objašnjen.

Slika 2.1. Histogram atributa

Na histogramu atributa 1 postoje tri klase od kojih najviše atributa ima u klasi dva, nešto manje u klasi tri koju onda prati klasa jedan. Histogram atributa 2 ima dvije klase te prevladava prva klasa, dok u drugoj klasi nema mnogo atributa. Nadalje, histogram atributa 3 ima tri klase te prevladava druga klasa, slijedi ju treća klasa pa prva. Histogram atributa 4 pak ima četiri klase i najviše atributa je prisutno u klasi četiri, slijedi klasa tri, a u klasi jedan i dva je podjednak broj atributa. U histogramu atributa 6 su dvije klase od kojih dominira prva klasa. Histogram atributa 7 ima tri klase te prevladava druga klasa koju slijedi treća klasa, a u prvoj klasi je najmanje atributa.

Isti slučaj je i za histogram atributa 8. Za razliku od histograma atributa 7 i 8, kod histograma atributa 9 prevladava treća klasa te ju prati druga pa prva klasa. Histogram atributa 10 ima tri klase te prevladava prva klasa, a u drugoj i trećoj klasi je vrlo malo atributa. Kod histograma atributa 11 i 12 pojavljuju se tri klase te prva klasa dominira slijede ju druga klasa te na kraju treća klasa. Što se tiče histograma atributa 13 dominira druga klasa, nakon druge klase najviše atributa ima u prvoj klasi te slijede treća i četvrta klasa. Histogrami atributa 14, 15, 16 i 17 sve imaju tri klase, jedino u histogramu atributa 17 prevladava prva klasa, a u preostalima druga klasa. Kod histograma atributa 18 i 19 vrlo je slična situacija, postoje dvije klase te prevladava klasa dva. Histogram atributa 20 ima tri klase te prevladava prva klasa, dok u drugoj klasi ima najmanje atributa. Nadalje, histogram atributa 21 ima tri klase, dominira treća klasa, slijedi ju prva klasa te na kraju druga klasa. U histogramima atributa 22, 23 i 24 samo su dvije klase i kod svih je dominantna druga klasa. Histogrami atributa 25, 26 i 27 imaju svaki po tri klase, u histogramima 26 i 27 prevladava druga klasa, dok kod histograma 25 dominira prva klasa. Kod histograma atributa 28 samo su dvije klase i dominantna je prva klasa. Što se tiče histograma atributa 29 i 30, kod oboje dominira druga klasa iza koje slijede redom treća pa prva. Histogram atributa 31 ima tri klase, najmanje atributa ima u klasi tri iza čega slijedi druga klasa, a prva klasa je najdominantnija. Histogrami atributa 32 i 33 su skoro pa identični, prevladava treća klasa, a u drugoj klasi ima najmanje atributa. Na histogramu 34 javljaju se tri klase od kojih treća dominira, a prva i druga imaju podjednak broj atributa. Histogrami 35 i 36 imaju tri klase najmanje atributa je u klasi tri, a najviše u klasi jedan. Od tri klase u histogramu 37 najdominantnija je druga klasa, prati ju prva te na kraju treća klasa. U histogramu atributa 38 od triju klasa podjednako prevladavaju klase jedan i dva. Histogrami 40 i 41 imaju tri klase te prevladava druga klasa. U histogramu atributa 42 podjednako prevladavaju prva i druga klasa,a treća klasa ima minimalan broj atributa. Od tri postojeće klase u histogramu atributa 43 dominira druga klasa, slijedi ju prva klasa te na kraju treća klasa. Kod histograma atributa 44 od tri klase najdominantnija je druga klasa, podosta manje atributa ima u trećoj klasi, a u prvoj ih ima najmanje. Slično kao i kod histograma atributa 43 je i u histogramima 45, 46 i 47 s nešto manjim brojem atributa. Histogrami atributa 48 i 49 imaju samo dvije klase od kojih je dominantnija prva klasa. U histogramima 50, 51 i 52 postoje tri klase, od kojih je najdominantnija druga klasa, a prva i treća klasa su podjednake, s nešto manje atributa u slučaju 50 i 51. Histogrami atributa 53 i 54 su identični, imaju tri klase prevladava druga klasa, iza nje slijedi prva klasa te na kraju treća klasa.Posljednje, histogrami atributa 55, 56 i 57 imaju samo dvije klase te je dominantna klasa dva.

U tablici 2.3 dane su deskriptivne statistike seta podataka.

Tablica 2.3. Tablica deskriptivnih statistika

	0	1	2	3	4	5	6	7	8
UNIQUE	3	2	3	4	4	2	3	3	3
MIN	1	0	1	0	1	0	1	1	1
MAX	3	1	3	3	2	1	3	3	3
MEDIAN	2	0	2	2	1	0	2	2	3
AVERAGE	2,03125	0,03125	2,375	2,03125	1,09375	0,28125	2,1875	2,125	2,40625
VARIANCE	0,6119	0,03125	0,30645	1,06351	0,92641	0,20867	0,54435	0,56452	0,57157
STD	0,78224	0,17678	0,55358	1,03127	0,9625	0,4568	0,7378	0,75134	0,75602
COUNT	32	32	32	32	28	32	32	32	32
	9	10	11	12	13	14	15	16	17
UNIQUE	3	3	3	4	3	3	3	3	2
MIN	1	1	1	0	1	1	1	1	1
MAX	3	3	3	3	3	3	3	3	2
MEDIAN	1	1	1	1	2	2	2	1	2
AVERAGE	1,15625	1,40625	1,5	0,90625	1,9375	2,1875	2,21875	1,3125	1,875
VARIANCE	0,26512	0,31351	0,45161	0,73286	0,64113	0,54435	0,56351	0,28629	0,1129
STD	0,5149	0,55992	0,67202	0,85607	0,80071	0,7378	0,75067	0,53506	0,33601
COUNT	32	32	32	32	32	32	32	32	32
	18	19	20	21	22	23	24	25	26
UNIQUE	2	3	3	2	2	2	3	3	3
MIN	1	0	0	1	1	1	1	1	1
MAX	2	2	2	2	2	2	3	3	3
MEDIAN	2	0	1,5	2	2	2	1	2	2
AVERAGE	1,875	0,46875	1,09375	1,75	1,8125	1,75	1,34375	1,71875	1,9375
VARIANCE	0,1129	0,70867	0,92641	0,19355	0,15726	0,19355	0,42641	0,3377	0,64113
STD	0,33601	0,84183	0,9625	0,43994	0,39656	0,43994	0,653	0,58112	0,80071
COUNT	32	32	32	32	32	32	32	32	32
	27	28	29	30	31	32	33	34	35
UNIQUE	2	3	3	3	3	3	3	3	3
MIN	2	1	1	1	1	1	1	1	1
MAX	3	3	3	3	3	3	3	3	3
MEDIAN	2	2	2	1	3	3	3	2	1,5
AVERAGE	2,3125	2,0625	2,15625	1,4375	2,3125	2,3125	2,71875	1,71875	1,625
VARIANCE	0,22177	0,31855	0,45867	0,5121	0,80242	0,80242	0,40222	0,53125	0,5
STD	0,47093	0,5644	0,67725	0,71561	0,89578	0,89578	0,63421	0,72887	0,70711
COUNT	32	32	32	32	32	32	32	32	32
	36	37	38	39	40	41	42	43	44
UNIQUE	3	3	4	3	3	3	3	3	3
MIN	1	1	1	1	1	1	1	1	1
MAX	3	3	3	3	3	3	3	3	3
MEDIAN	2	2	2	2	2	2	2	2	2
AVERAGE	1,6875	1,6875	1,625	1,96875	2	1,59375	1,75	2,0625	1,875
VARIANCE	0,28629	0,47984	0,56452	0,2248	0,19355	0,37802	0,3871	0,125	0,24194
STD	0,53506	0,6927	0,75134	0,47413	0,43994	0,61484	0,62217	0,35355	0,49187
COUNT	32	32	31	32	32	32	32	32	32
	45	46	47	48	49	50	51	52	53
UNIQUE	3	3	2	2	3	3	3	3	3
MIN	1	1	2	2	1	1	1	1	1
MAX	3	3	3	3	3	3	3	3	3
MEDIAN	2	2	2	2	2	2	2	2	2
AVERAGE	1,96875	1,9375	2,0625	2,0625	2	2	2	1,84375	1,84375
VARIANCE	0,16028	0,18952	0,06048	0,06048	0,12903	0,12903	0,25806	0,2006	0,2006
STD	0,40035	0,43533	0,24593	0,24593	0,35921	0,35921	0,508	0,44789	0,44789
COUNT	32	32	32	32	32	32	32	32	32
	54	55	56						
UNIQUE	2	2	2						
MIN	1	1	1						
MAX	2	2	2						
MEDIAN	2	2	2						
AVERAGE	1,5625	1,8125	1,71875						
VARIANCE	0,25403	0,15726	0,20867						
STD	0,50402	0,39656	0,4568						
COUNT	32	32	32						

Varijabla 0 ima 3 jedinstvene vrijednosti, minimalna vrijednost je 1, maksimalna vrijednost je 3 te je srednja vrijednost 2. U ovom slučaju, prosječna vrijednost je 2.03125 i varijanca 0.6119 sa standardnom devijacijom 0.78224. Nadalje, varijabla 1 ima dvije jedinstvene vrijednosti, minimalna i srednja vrijednost su 0, dok je maksimalna vrijednost 1. Standardna devijacija i varijanca imaju vrijednosti 0.17678 i 0.03125, te je prosjek 0.03125. Varijabla 2 ima prosječnu vrijednost od 2.375, sa standardnom devijacijom 0.55358, varijancom 0.30645. Ova varijabla ima 3 jedinstvene vrijednosti, maksimalna je 3, minimalna 1 i srednja vrijednost iznosi 2. Varijabla 3 sa standardnom devijacijom 1.03127 i varijancom 1.06351 ima prosječnu vrijednost od 2.03125 te sadrži 4 jedinstvene vrijednosti. Minimalna vrijednost je 0, maksimalna je 3 i srednja vrijednost je 2. Varijabla 4, kao i varijabla 3, sadrži 4 jedinstvene vrijednosti, minimalna i srednja vrijednost su u ovom slučaju 1, a maksimalna vrijednost je 2. Postignuta prosječna vrijednost je 1.09375, dok su standardna devijacija i varijanca 0.9625 i 0.92641. Nadalje, varijabla 5 ima dvije jedinstvene vrijednosti, minimalna i srednja vrijednost su 0 te je maksimalna 1. Za ovaj slučaj standardna devijacija je 0.4568, varijanca iznosi 0.20867 te je prosječna vrijednost 0.28125. Varijable 6 i 7 imaju po 3 jedinstvene vrijednosti, kod obje varijable maksimalna vrijednost je 3, srednja vrijednost 2 i minimalna 1. Kod varijable 6 varijanca i standardna devijacija imaju vrijednosti 0.54435 i 0.7378, a prosječna vrijednost je 2.1875, dok je kod varijable 7 prosječna vrijednost 2.125, a standardna devijacija i varijanca iznose 0.75134 i 0.56452. Varijabla 8 ima standardnu devijaciju 0.75602, prosječna vrijednost i varijanca imaju iznose 2.40625 i 0.57157, te ova varijabla ima 3 jedinstvene vrijednosti od kojih su maksimalna i srednja vrijednost 3, a minimalna vrijednost je 1. Varijable 9, 10 i 11 imaju po 3 jedinstvene vrijednosti, minimalne i srednje vrijednosti su 1, dok je maksimalna vrijednost kod svih tri varijabli 3. Varijabla 9 ima prosječnu vrijednost 1,15625, varijancu 0.26512 i standardnu devijaciju 0.5149. Varijabla 10 pak ima standardnu devijaciju 0.55992, varijancu 0.31351 i prosjek je 1.40625. Varijabla 11 ima varijancu 0.45161, prosjek 1.5 i standardnu devijaciju 0.67202. Nadalje, varijabla 12 sa standardnom devijacijom 0.85607, varijancom 0.73286 i prosječnom vrijednosti 0.90625 ima 4 jedinstvenih vrijednosti. Minimalna vrijednost je 0, maksimalna 3 i srednja vrijednost iznosi 1. Varijable 13, 14 i 15 imaju 3 jedinstvene vrijednosti, minimalne vrijednosti su 1, maksimalne su 3 i srednje vrijednosti imaju iznos od 2. Što se tiče varijable 13 prosječna vrijednost iznosi 1.9375, varijanca i standardna devijacija imaju vrijednosti 0.64113 i 0.80071. Varijabla 14 pak ima prosječnu vrijednost 2.1875, varijanca i standardna devijacija su 0.54435 i 0.7378. Kod varijable 15 prosječna vrijednost iznosi 2.21875, varijanca je 0.56351 i standardna devijacija je 0.75067. Varijabla 16 s prosječnom vrijednosti 1.3125, varijancom 0.28629 i standardnom devijacijom 0.53506, ima 3 jedinstvene vrijednosti. Minimalna i srednja vrijednost su 1, a maksimalna je 3. Varijabla 17 ima maksimalnu i srednju vrijednost 2, minimalna je 1, a varijabla ima dvije jedinstvene vrijednosti. Kod ove varijable prosječna vrijednost je 1.875, varijanca i standardna devijacija su 0.1129 i 0.33601. Varijabla 18 ima dvije jedinstvene vrijednosti, minimalna vrijednost je 1, maksimalna vrijednost je 2 te je srednja vrijednost 2. U ovom slučaju, prosječna vrijednost je 1.875 i varijanca 0.1129 sa standardnom devijacijom 0.33601. Nadalje, varijabla 19 ima 3 jedinstvene vrijednosti, minimalna i srednja vrijednost su

0, dok je maksimalna vrijednost 2. Standardna devijacija i varijanca imaju vrijednosti 0.84183 i 0.70867, te je prosjek 0.46875. Varijabla 20 ima prosječnu vrijednost 1.09375, sa standardnom devijacijom 0.9625, varijancom 0.92641. Ova varijabla ima 3 jedinstvene vrijednosti, maksimalna je 2, minimalna 0 i srednja vrijednost iznosi 1.5. Varijabla 21 sa standardnom devijacijom 0.43994 i varijancom 0.19355 ima prosječnu vrijednost od 1.75 te sadrži dvije jedinstvene vrijednosti. Minimalna vrijednost je 1, maksimalna je 2 i srednja vrijednost je 2. Varijabla 22, kao i varijabla 23, sadrži dvije jedinstvene vrijednosti, maksimalna i srednja vrijednost su u ovom slučaju 2, a minimalna vrijednost je 1. Postignuta prosječna vrijednost varijable 22 je 1.8125, dok su standardna devijacija i varijanca 0.39656 i 0.15726. Kod varijable 23 je prosječna vrijednost 1.75, varijanca 0.19355, sa standardnom devijacijom 0.43994. Nadalje, varijabla 24 ima 3 jedinstvene vrijednosti, minimalna i srednja vrijednost su 1 te je maksimalna 3. Za ovaj slučaj standardna devijacija je 0.653, varijanca iznosi 0.42641 te je prosječna vrijednost 1.34375. Varijable 25 i 26 imaju po 3 jedinstvene vrijednosti, kod obje varijable maksimalna vrijednost je 3, srednja vrijednost 2 i minimalna 1. Kod varijable 25 varijanca i standardna devijacija imaju vrijednosti 0.3377 i 0.58112, a prosječna vrijednost je 1.71875, dok je kod varijable 26 prosječna vrijednost 1.9375, a standardna devijacija i varijanca iznose 0.80071 i 0.64113. Varijabla 27 ima standardnu devijaciju 0.47093, prosječna vrijednost i varijanca imaju iznose 2.3125 i 0.22177, te ova varijabla ima dvije jedinstvene vrijednosti od kojih su minimalna i srednja vrijednost 2, a maksimalna vrijednost je 3. Varijable 28 i 29 imaju 3 jedinstvene vrijednosti, minimalne vrijednosti su 1, srednje 2 i maksimalne vrijednosti su 3. Prosječna vrijednost i varijanca varijable 28 iznose 2.0625 i 0.31855, sa standardnom devijacijom 0.5644. Dok je, kod varijable 29, varijanca 0.45867, prosjek 2.15625 i standardna devijacija 0.67725. Varijabla 30 ima prosjek 1.4375, varijancu 0.5121 te standardnu devijaciju 0.71561. Broj jedinstvenih vrijednosti je 3, minimalna i srednja vrijednost su 1, a maksimalna je 3. Varijable 31, 32 i 33 imaju po 3 jedinstvene vrijednosti, maksimalne i srednje vrijednosti su 1, dok je minimalna vrijednost kod svih tri varijabli 1. Varijabla 31 ima prosječnu vrijednost 2.3125, varijancu 0.80242 i standardnu devijaciju 0.89578. Varijabla 32 pak ima standardnu devijaciju 0.89578, varijancu 0.80242 i prosjek je 2.3125. Varijabla 33 ima varijancu 0.40222, prosjek 2.71875 i standardnu devijaciju 0.63421. Nadalje, varijabla 34 sa standardnom devijacijom 0.72887, varijancom 0.53125 i prosječnom vrijednosti 1.71875 ima 3 jedinstvene vrijednosti. Minimalna vrijednost je 1, maksimalna 3 i srednja vrijednost iznosi 2. Varijabla 35 ima 3 jedinstvene vrijednosti, maksimalna vrijednost je 3, minimalna 1 i srednja vrijednost 1.5. U ovom slučaju prosjek je 1.625, varijanca iznosi 0.5 te je standardna devijacija 0.70711. Varijable 36 i 37 imaju 3 jedinstvene vrijednosti, minimalne vrijednosti su 1, maksimalne su 3 i srednje vrijednosti imaju iznos od 2. Što se tiče varijable 36 prosječna vrijednost iznosi 1.6875, varijanca i standardna devijacija imaju vrijednosti 0.28629 i 0.53506. Varijabla 37 pak ima prosječnu vrijednost 1.6875, varijanca i standardna devijacija su 0.47984 i 0.6927. Kod varijable 38 prosječna vrijednost iznosi 1.625, varijanca je 0.56452 i standardna devijacija je 0.75134. Sadrži 4 jedinstvenih vrijednosti, minimalna vrijednost je 1, srednja vrijednost 2 i maksimalna vrijednost je 3. Varijabla 39 s prosječnom vrijednosti 1.96875, varijancom 0.2248 i standardnom devijacijom 0.47413,

ima 3 jedinstvene vrijednosti. Minimalna vrijednost je 1, srednja vrijednost 2, a maksimalna je 3. Varijabla 40 ima maksimalnu vrijednost 3, srednju vrijednost 2 te minimalnu vrijednost 1, a varijabla ima 3 jedinstvene vrijednosti. Kod ove varijable prosječna vrijednost je 2, varijanca i standardna devijacija su 0.19355 i 0.43994. Varijable 41, 42, 43, 44, 45 i 46 sve imaju po 3 jedinstvene vrijednosti, minimalne vrijednosti su 1, srednje 2 i maksimalne vrijednosti su 3. Varijabla 41 ima prosjek 1.59375, varijancu 0.37802 i standardnu devijaciju 0.61484. Sljedeća varijabla, tj. varijabla 42 sa standardnom devijacijom 0.62217 ima prosjek 1.75 i varijancu 0.3871. Varijabla 43 pak ima prosjek 2.0625, standardnu devijaciju 0.35355 i varijancu 0.125. Nadalje, varijabla 44 s varijancom 0.24194 i standardnom devijacijom 0.49187 ima prosjek 1.875. Varijabla 45 s prosjekom 1.96875 ima standardnu devijaciju 0.40035 i varijancu 0.16028. Te posljednja varijabla iz ovog niza, tj. varijabla 46 ima standardnu devijaciju 0.43533, prosjek 1.9375 i varijancu 0.18952. Varijabla 47 ima dvije jedinstvene vrijednosti, minimalna vrijednost je 2, maksimalna vrijednost je 3 te je srednja vrijednost 2. U ovom slučaju, prosječna vrijednost je 2.0625 i varijanca 0.06048 sa standardnom devijacijom 0.24593. Nadalje, varijabla 48 ima dvije jedinstvene vrijednosti, minimalna i srednja vrijednost su 2, dok je maksimalna vrijednost 3. Standardna devijacija i varijanca imaju vrijednosti 0.24593 i 0.06048, te je prosjek 2.0625. Varijabla 49 ima prosječnu vrijednost od 2, sa standardnom devijacijom 0.35921, varijancom 0.12903. Ova varijabla ima 3 jedinstvene vrijednosti, maksimalna je 3, minimalna 1 i srednja vrijednost iznosi 2. Varijabla 50 sa standardnom devijacijom 0.35921 i varijancom 0.12903 ima prosječnu vrijednost od 2 te sadrži 3 jedinstvene vrijednosti. Minimalna vrijednost je 1, maksimalna je 3 i srednja vrijednost je 2. Varijable 51, 52 i 53 imaju po 3 jedinstvene vrijednosti, minimalne vrijednosti su 1, srednje vrijednosti su 2, dok je maksimalna vrijednost kod svih tri varijabli 3. Varijabla 51 ima prosječnu vrijednost 2, varijancu 0.25806 i standardnu devijaciju 0.508. Varijable 52 i 53 pak imaju standardnu devijaciju 0.44789, varijancu 0.2006 i prosjek je 1.84375. Varijable 54, 55 i 56 imaju po dvije jedinstvene vrijednosti, minimalne vrijednosti su 1, srednje i maksimalne vrijednosti iznose 2. Varijabla 54 ima prosječnu vrijednost 1.5625, varijancu 0.25403 i standardnu devijaciju 0.50402. Varijabla 55 pak ima standardnu devijaciju 0.39656, varijancu 0.15726 i prosjek je 1.8125. Posljednja varijabla tj. varijabla 56 ima varijancu 0.20867, prosjek 1.71875 i standardnu devijaciju 0.4568.

3. Metodologija

U ovom poglavlju ćemo predstaviti što su neuronske mreže kao metoda strojnog učenja. Također će se predstaviti najkorišteniji model u neuronskim mrežama tzv. višeslojni perceptron. Tradicionalna metoda optimizacije hiperparametara tj. pretraživanje mreže.

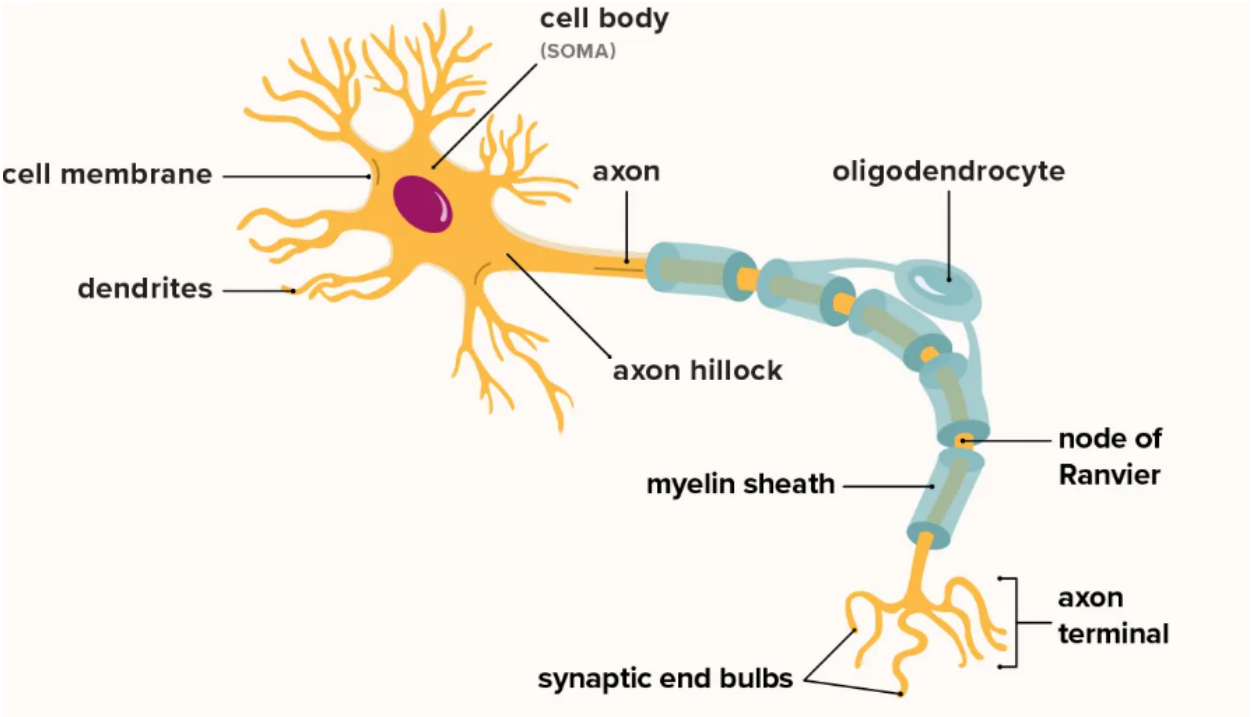
3.1. Višeslojni perceptron

U tradicionalnom strojnom učenju svatko tko želi graditi model mora biti stručnjak za područje problema na kojem radi. Bez stručnog znanja, značajke projektiranja i inženjeringa postaju sve teži izazov. Kvaliteta modela strojnog učenja ovisi o kvaliteti skupa podataka, ali i o tome koliko dobro značajke kodiraju uzorke u podacima. Algoritmi dubokog učenja koriste umjetne neuronske mreže kao svoju glavnu strukturu. Ono što ih razlikuje od drugih algoritama je to što ne zahtijevaju stručne podatke tijekom faze dizajna značajki i inženjeringa. Neuronske mreže mogu naučiti karakteristike podataka. Algoritmi dubokog učenja uzimaju skup podataka i uče njegove obrasce, uče kako predstaviti podatke sa značajkama koje sami izdvajaju. Zatim kombiniraju različite prikaze skupa podataka, od kojih svaki identificira određeni obrazac ili karakteristiku u apstraktniji prikaz skupa podataka na visokoj razini. Ovaj praktični pristup omogućuje algoritmima da se puno brže prilagode podacima koji su pri ruci. Neuronske mreže inspirirane su strukturom mozga, ali ne točnim modelom zbog, još uvijek, nedovoljnog znanja o mozgu i njegovom funkcioniranju. Ljudski mozak služi kao inspiracija u mnogim znanstvenim područjima zbog svoje sposobnosti razvijanja inteligencije. Postoje neuronske mreže koje su stvorene s jedinom svrhom razumijevanja rada mozga, duboko učenje kakvo danas poznajemo nije namijenjeno repliciranju načina na koji mozak funkcionira. Umjesto toga, duboko učenje fokusira se na omogućavanje sustava koji uče višestruke razine sastava uzoraka. I, kao i sa svakim znanstvenim napretkom, duboko učenje nije započelo sa složenim strukturama i široko rasprostranjenim primjenama, već je počelo s osnovnom strukturom, onom koja podsjeća na moždani neuron. Početkom 1940-ih Warren McCulloch, neurofiziolog, udružio se s logičarom Walterom Pittsom kako bi stvorili model funkcioniranja mozga. Bio je to jednostavan linearni model koji je proizveo pozitivan ili negativan izlaz, s obzirom na skup ulaza i težine kao što je prikazano u formuli (3.1).

$$f(x, w) = x_1 w_1 + \cdots + x_n w_n, \tag{3.1}$$

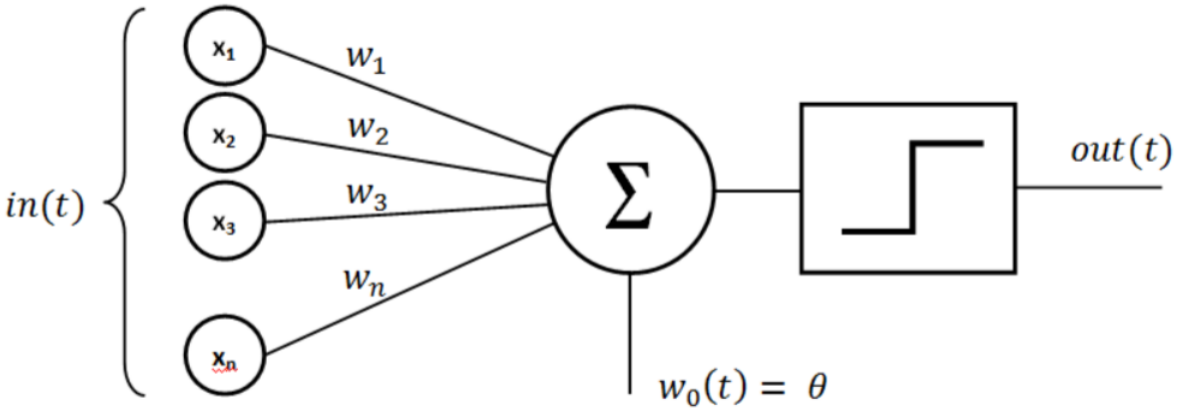
gdje je:
$f(x, w)$ - izlaz
x_n - ulazi
w_n - težine.

Ovaj model računanja nazvan je neuron, jer je pokušao oponašati način na koji radi temeljni blok mozga. Kao što moždani neuroni primaju električne signale, McCullochov i Pittsov neuron je primao ulazne podatke i ako su ti signali bili dovoljno jaki, prosljeđivao ih drugim neuronima.



Slika 3.1. Struktura neurona [23]

Prva primjena neurona replicirala je logička vrata, gdje postoji jedan ili dva binarna ulaza i booleova funkciju koja se aktivira samo ako su dati pravi ulazi i težine. Međutim, ovaj model je imao problem. Jedini način da se dobije željeni učinak je da su težine prethodno postavljene. Tek desetljeće kasnije je Frank Rosenblatt proširio ovaj model i stvorio algoritam koji je mogao naučiti težine kako bi se generirao izlaz. Na osnovu McCullochovog i Pittovog neurona, Rosenblatt je razvio perceptron. Kod Rosenblattovog modela ulazi se kombiniraju u ponderirani zbroj i ako taj zbroj premaši unaprijed definirani prag, neuron se aktivira i proizvodi izlaz.



Slika 3.2. Umjetni neuron koji koristi perceptron [24]

$$out\,(t) = \left\{ \begin{array}{c} 1, ako \sum w_i x_i - T > 0 \\ 0, inace \end{array} \right\}, \tag{3.2}$$

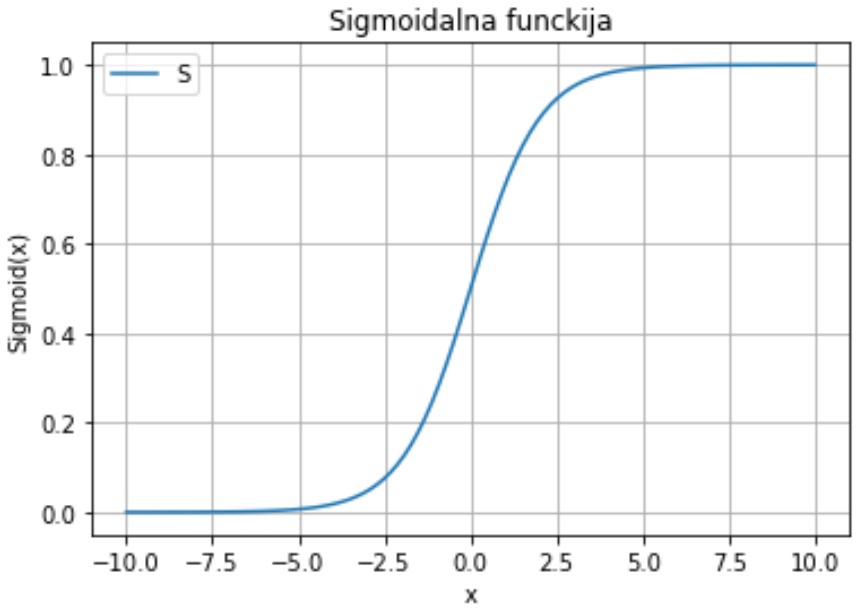
gdje T predstavlja aktivacijsku funkciju. Ako je ponderirani zbroj ulaza veći od nule, neuron daje vrijednost 1, inače je izlazna vrijednost nula.

S diskretnim izlazom i kontroliranom aktivacijskom funkcijom, perceptron se može koristiti kao binarni klasifikacijski model, definirajući linearnu granicu odlučivanja. Pronalazi razdvajajuću hiperravninu koja smanjuje udaljenost između pogrešno klasificiranih točaka i granice odluke kao što je opisano jednadžbom (3.3).

$$D\,(w, c) = -\sum_{i \in M} y_i\,(x_i w_i + c)\,, \tag{3.3}$$

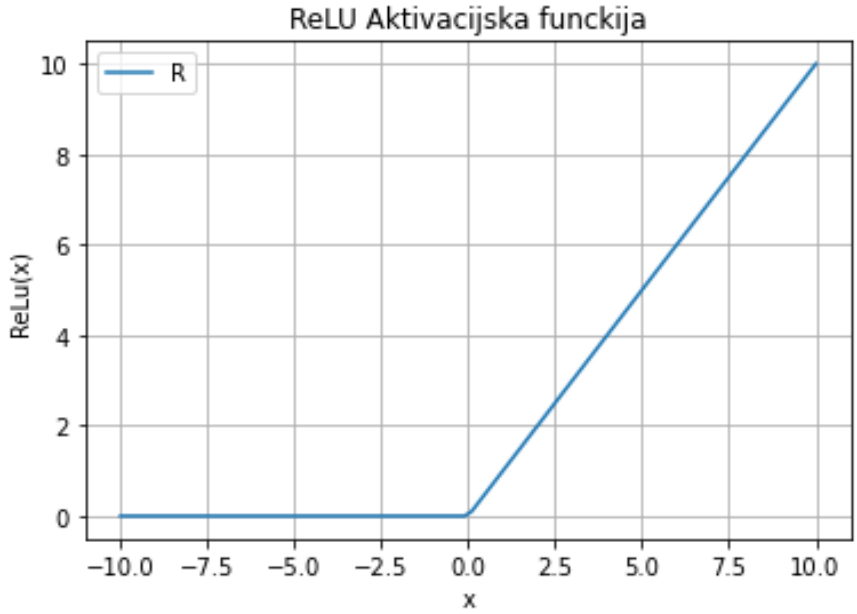
gdje je,
$D\,(w, c)$ - udaljenost
M - pogrešno klasificirano opažanje
y_i - izlaz.

Kako bi se smanjila udaljenost, perceptron koristi stohastički gradijentni pad kao funkciju optimizacije. Ako su podaci linearno odvojivi, stohastički gradijentni pad će konvergirati u konačnom broju koraka. Posljednji dio koji je potreban perceptronu je funkcija aktivacije. Funkcija aktivacije određuje hoće li se neuron aktivirati ili ne. Početni modeli perceptrona koristili su sigmoidalnu funkciju. Ona preslikava svaki stvarni ulaz u vrijednost koja je 0 ili 1 i kodira nelinearnu funkciju. Što znači da neuron može primiti negativne brojeve kao ulaz, ali će i dalje moći proizvesti izlaz koji je 0 ili 1. Na slici 3.3 prikazana je sigmoidalna funkcija.



Slika 3.3. Sigmoidalna funkcija

Ako se pogledaju algoritmi dubokog učenja iz prošlog desetljeća, može se vidjeti da većina njih koristi rektificiranu linearnu (ReLu) funkciju kao funkciju aktivacije neurona. Na slici 3.4 prikaza je ReLu aktivacijska funkcija.



Slika 3.4. ReLu aktivacijska funkcija

Razlog zašto je ReLu postao sve prihvaćeniji je taj što omogućuje bolju optimizaciju korištenjem stohastičkog gradijentnog pada, učinkovitije izračunavanje te nepromjenjiv omjer, što znači da na njegove karakteristike ne utječe razmjer ulaza.

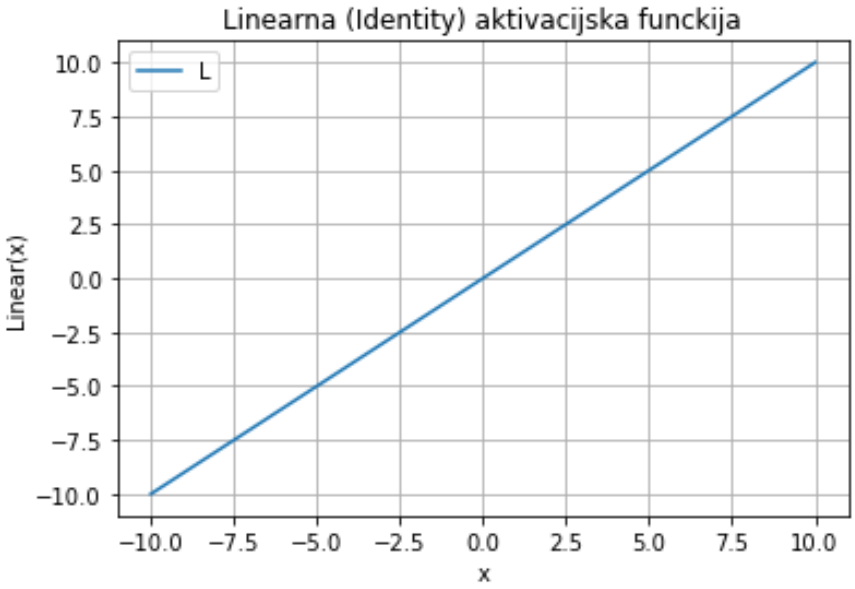
Neuron prima ulazne podatke i nasumično bira početni skup težina. Dalje se oni kombiniraju u ponderirani zbroj, a zatim ReLu funkcija određuje vrijednost izlaza.

Aktivacijska funkcija koja se također koristi kod dubokog učenja je tangens hiperbolna (Tanh) aktivacijska funkcija. Raspon izlaza funkcije tanh je (-1, 1) i predstavlja slično ponašanje kao i sigmoidna funkcija. Glavna razlika je činjenica da funkcija tanh gura ulazne vrijednosti na 1 i -1 umjesto na 1 i 0. Na slici 3.5 prikazana je Tanh funkcija: $f\left(x\right) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

Slika 3.5. Tanh aktivacijska funkcija

Povijesno gledano, funkcija tanh postala je preferirana u odnosu na sigmoidalnu funkciju jer je davala bolje performanse za višeslojne neuronske mreže. Ali to nije riješilo problem nestajanja gradijenta, koji je učinkovitije riješen uvođenjem ReLU aktivacijske funkcije.

Na slici 3.6 prikazana je najjednostavnija od svih aktivacijskih funkcija koja se koristi kod dubokog učenja, a to je linearna (identity) aktivacijska funkcija. Uzima ulaze, pomnožene s težinama za svaki neuron i stvara izlazni signal proporcionalan ulazu $f\left(x\right) = x$. Nedostatak ove funkcije je taj što propagacija unatrag nije moguća - izvod funkcije je konstanta i nema veze s ulazom. Dakle, nije moguće vratiti se i razumjeti koje težine u ulaznim neuronima mogu dati bolje predviđanje. Također, bez obzira koliko slojeva u neuronskoj mreži ima, posljednji sloj bit će linearna funkcija prvog sloja. Domena linearne funkcije je od $-\infty$ do ∞.



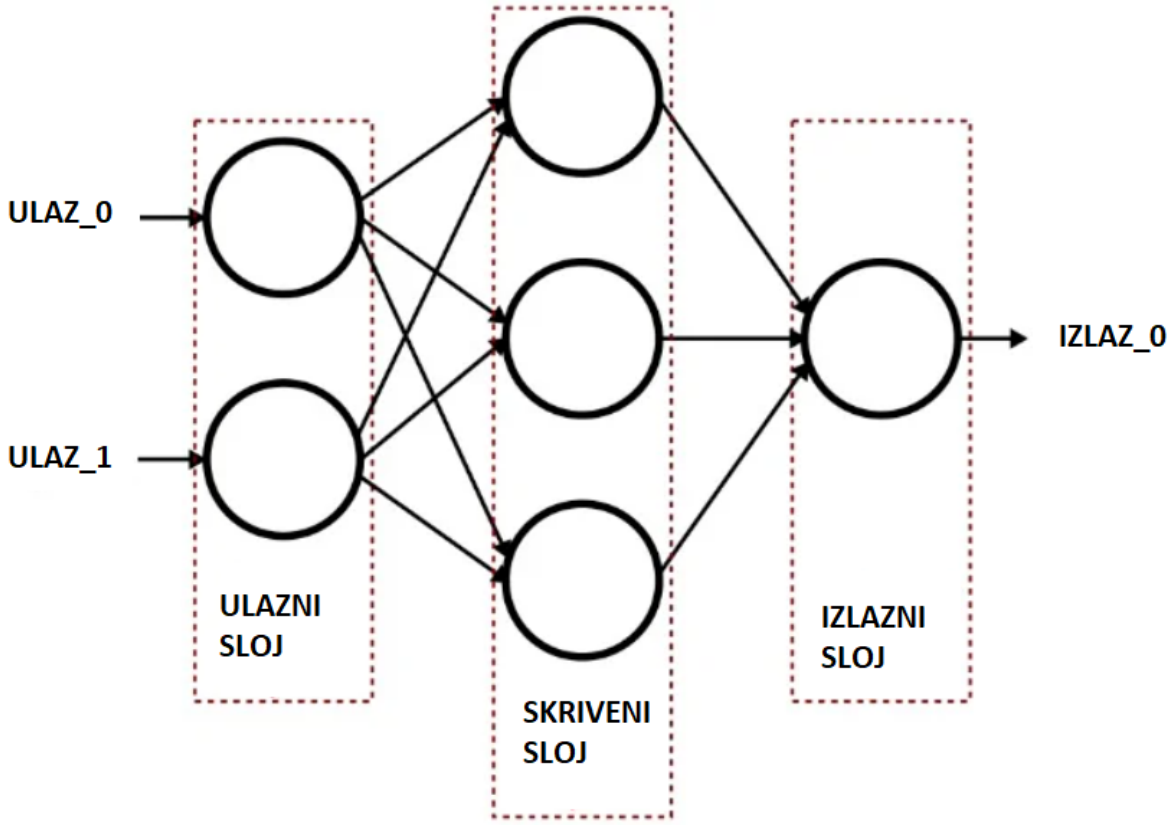
Slika 3.6. Linearna (Identity) aktivacijska funkcija

Perceptron koristi stohastički gradijentni pad kako bi pronašao tj. naučio skup težine koji minimizira udaljenost između pogrešno klasificiranih točaka i granice odluke. Jednom kada stohastički gradijentni pad konvergira, skup podataka je odvojen u dva područja linearnom hiperravninom. Iako je rečeno da perceptron može predstavljati bilo koji sklop i logiku, najveća kritika bila je da ne može predstavljati XOR vrata, isključivi OR, gdje vrata vraćaju samo 1 ako su ulazi različiti. Ovo su gotovo desetljeće kasnije dokazali Minsky i Papert, 1969. i naglašavaju činjenicu da se perceptron, sa samo jednim neuronom, ne može primijeniti na nelinearne podatke.

Višeslojni perceptron (eng. Multilayer Perceptron - MLP) razvijen je kako bi se uhvatio u koštac s ovim ograničenjem. To je neuronska mreža gdje je preslikavanje između ulaza i izlaza nelinearno.

Višeslojni perceptron su najkorišteniji modeli u neuronskim mrežama koje koriste algoritam unazadne propagacije. Također, višeslojni perceptron je specifična vrsta višeslojne mreže koja završava s jednim neuronom. Definicija arhitekture umjetne neuronske mreže je relevantna budući da nedostatak veza može učiniti mrežu nesposobnom za rješavanje problema nedovoljnih podesivih parametara, dok višak veza može uzrokovati prekomjerno uklapanje podataka za treniranje, osobito u slučaju kada se koristi veliki broj slojeva i neurona. Višeslojni perceptron je varijanta originalnog modela mreže koju je predložio Rosenblatt 1950. godine [7]. On ima jedan ili više skrivenih slojeva između svojih ulaznih i izlaznih slojeva, u kojima su organizirani neuroni, veze su uvijek usmjerene od donjih prema gornjim slojevima te kod višeslojnog perceptrona neuroni u istom sloju nisu nikad povezani. Broj neurona u ulaznom sloju jednak je dimenzijama ulaznog vektora za svaku podatkovnu točku. Za mrežu se kaže da je potpuno povezana ako je svaki neuron u svakom od slojeva povezan na sve neurone koji se nalaze u sljedećem sloju. Mreža je djelomično povezana ako neke od veza nedostaju. Na slici 3.7 prikazan je primjer jednog višeslojnog

perceptrona.



Slika 3.7. Višeslojni perceptron [5]

Višeslojni perceptron spada u kategoriju feedforward algoritma, jer se ulazi kombiniraju s početnim težinama u težinski zbroj i podvrgavaju funkciji aktivacije, baš kao u perceptronu. Ali razlika je u tome što se svaka linearna kombinacija prenosi na sljedeći sloj. Svaki sloj hrani sljedeći s rezultatom svog izračuna, njihovim internim prikazom podataka. Ovo ide skroz kroz skrivene slojeve do izlaznog sloja. Kad bi algoritam samo izračunao ponderirane zbrojeve u svakom neuronu, propagirao rezultate u izlazni sloj i tu stao, ne bi mogao naučiti pondere koji minimiziraju funkciju troška. Kad bi algoritam izračunao samo jednu iteraciju, stvarnog učenja ne bi bilo. Ovo je sada mjesto gdje nastupa unazadna propagacija (eng. Backpropagation). Unazadna propagacija je mehanizam učenja koji omogućuje višeslojnom perceptronu da iterativno prilagođava težine u mreži, s ciljem minimiziranja troškovne funkcije. Postoji jedan težak zahtjev za pravilno funkcioniranje unazadne propagacije. Funkcija koja kombinira ulaze i težine u neuronu, na primjer ponderirani zbroj i funkcija praga, na primjer ReLU, moraju biti diferencijabilne. Ove funkcije moraju imati ograničenu derivaciju, jer je gradijentni spust obično optimizacijska funkcija koja se koristi u višeslojnom perceptronu. U svakoj iteraciji, nakon što se ponderirani iznosi proslijede kroz sve slojeve, gradijent srednje kvadratne pogreške izračunava se preko svih ulaznih i izlaznih parova. Zatim, kako bi se proširio natrag, težine prvog skrivenog sloja ažuriraju se vrijednošću gradijenta. Tako se težine propagiraju natrag do početne točke neuronske mreže prema jednadžbi (3.4).

$$\Delta_w(t) = -\epsilon \frac{dE}{dw(t)} + \alpha \Delta_w(t-1), \tag{3.4}$$

gdje je
$\Delta_w(t)$ - gradijent trenutne iteracije
ϵ - pristranost
dE - greška
$dw(t)$ - vektor težine
α - stopa učenja
$\Delta_w(t-1)$ - gradijent prethodne iteracije

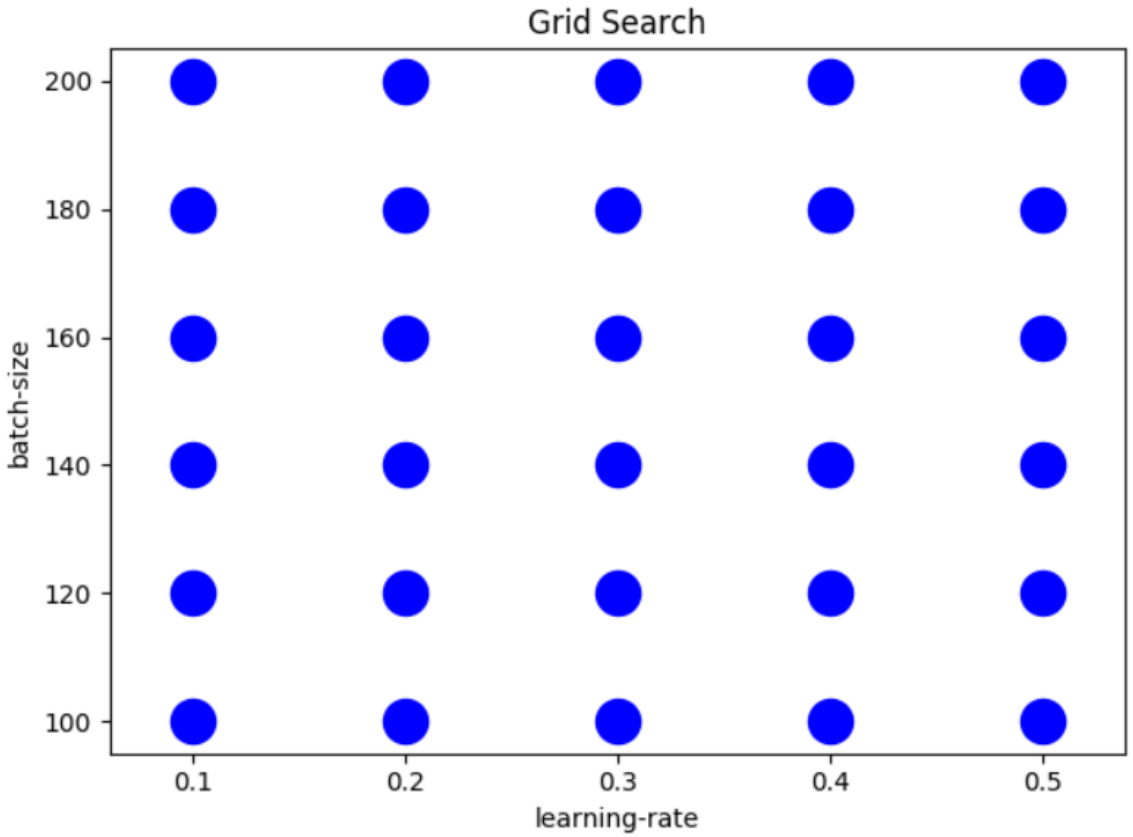
Ovaj se proces nastavlja sve dok gradijent za svaki ulazno-izlazni par ne konvergira, što znači da se novoizračunati gradijent nije promijenio više od navedenog praga konvergencije, u usporedbi s prethodnom iteracijom.

3.2. Pretraživanje mreže arhitektura

Umjetne neuronske mreže (eng. artificial neural networks - ANN) su potkategorija strojnog učenja gdje se proces učenja provodi pomoću velikog broja međusobno povezanih procesnih elemenata koji se nazivaju neuroni ili čvorovi. Ovaj način obrade podataka inspiriran je biološkim živčanim sustavima npr. prijenos električnih signala kroz neurološke mreže u čovjekovom mozgu. Dokazale su se brojne prednosti ANN-a, njihov potencijal u različitim medicinskim granama kao što su onkologija, neurologija, radiologija, ortopedija, kardiologija, pulmologija i slične biomedicinske primjene. Uzimajući u obzir medicinsko oslikavanje i radijacijsku onkologiju, duboko učenje (eng. deep learning - DL) pokazuje sposobnost automatskog i adaptivnog izdvajanja značajki slike iz uzoraka niske na visoku razinu, ovisno o zadatku računalnog vida. Višeslojna perceptron umjetna neuronska mreža (MLP-ANN) pokazala se puno boljim izborom u odnosu na tradicionalne nadzirane algoritme strojnog učenja u postizanju visoke točnosti dijagnoze.

Tradicionalna metoda optimizacije hiperparametara je pretraživanje mreže (eng. Grid search), koji jednostavno čini potpunu pretragu danog podskupa prostora hiperparametara treniranog algoritma. Na slici 3.8 prikazana je metoda pretraživanja prostora. Budući da prostor parametra algoritma strojnog učenja može uključivati prostore sa stvarnim ili neograničenim vrijednostima za neke parametre, moguće je da trebamo navesti granicu za primjenu pretraživanja mreže. Pretraživanje mreže pati od visokih dimenzionalnih prostora, ali se često mogu lako paralelizirati, budući da su vrijednosti hiperparametara s kojima algoritam radi obično neovisne jedni o drugima. U osnovi, domenu hiperparametara dijelimo na diskretnu mrežu. Zatim isprobavamo svaku kombinaciju vrijednosti ove mreže, izračunavajući neke metrike izvedbe koristeći unakrsnu provjeru. Točka rešetke koja maksimizira prosječnu vrijednost u unakrsnoj provjeri je optimalna kombinacija vrijednosti za hiperparametre. Pretraživanje rešetke je iscrpan algoritam koji obuhvaća sve

kombinacije, tako da zapravo može pronaći najbolju točku u domeni. Veliki nedostatak je što je jako spor. Provjera svake kombinacije prostora zahtijeva dosta vremena. Svaka točka u mreži treba k-struku unakrsnu provjeru, što zahtijeva k koraka obuke. Dakle, podešavanje hiperparametara modela na ovaj način može biti prilično složeno i skupo. Međutim, ako se traži najbolja kombinacija vrijednosti hiperparametara, pretraživanje mreže je vrlo dobra metoda.



Slika 3.8. Grid search [8]

Svaki od obučenih modela trebat će imati prilagođene hiperparametre kako bi se postigla kvalitetna izvedba regresije. Hiperparametri su vrijednosti koje opisuju opću arhitekturu neuronske mreže koja se koristi za treniranje modela. Hiperparametri MLP-a moraju se mijenjati kako bi se postigli najbolji regresijski modeli za svaki slučaj. Jedan od različitih hiperparametara je funkcija aktivacije neurona skrivenog sloja. Daljnji hiperparametri uključuju broj skrivenih slojeva i broj neurona po skrivenom sloju izražen kao (k_1, k_2, k_n) u kojem je ukupan broj slojeva n, a k_i predstavlja broj neurona u sloju i. Algoritam koji se koristi za izračunavanje vrijednosti težine tijekom procesa treninga, nazvan solver, također je jedan od različitih hiperparametara. Dodatni različiti hiperparametri su brzina učenja, koja prilagođava brzinu prilagodbe težine tijekom procesa povratnog širenja, kao i vrstu stope učenja—hoće li njegova vrijednost ostati konstantna ili će se skalirati ovisno o broju ponavljanja. Konačno, hiperparametar je $L2$ parametar regulacije, koji, ako je visok, kažnjava ulaze koji imaju veliki pojedinačni utjecaj na izlaznu vrijednost MLP-a—što može rezultirati nedovoljno opremljenim modelima. Kako bi se pronašao optimalni skup

hiperparametara, može se koristiti algoritam pretraživanja mreže. Pretraživanje mreže funkcionira tako da izračunava sve moguće kombinacije hiperparametara. Zatim se neuronska mreža trenira sa svakom od ovih kombinacija hiperparametara. Na ovaj način se može ispitati širok raspon hiperparametara. Iako algoritam možda neće pronaći najbolju moguću kombinaciju hiperparametara, s dovoljno hiperparametara može pronaći kombinaciju hiperparametara koja je blizu najbolje. Ako je potrebno, na primjer, ako nijedan od dobivenih modela ne daje zadovoljavajuće rezultate performanse, moguće vrijednosti hiperparametara mogu se proširiti ili dodatno poboljšati, oko kombinacije hiperparametara koja daje najbolje rezultate.

U tablici 3.1 prikazani su mogući hiperparametri. Broj varijanti dimenzija skrivenih slojeva je 50, aktivacijskih funkcija 4 te je broj mogućih vrijednosti rješavača 2. Što se tiče vrste stope učenja broj mogućih varijanta je 3. Početna stopa učenja i L2 regularizacija imaju po 4 mogućih vrijednosti. Množenjem navedenih varijanti hiperparametara dobije se ukupni broj kombinacija Grid search-a koji iznosi 19200.

Tablica 3.1. Tablica hiperparametara

Hiperparametar	**Moguće vrijednosti**	**Broj mogućih vrijednosti**
Dimenzije skrivenih slojeva	(1),(1,1),(1,1,1),(1,1,1,1),(1,1,1,1,1), (2),(2,2),(2,2,2),(2,2,2,2),(2,2,2,2,2), (4),(4,4),(4,4,4),(4,4,4,4),(4,4,4,4,4), (8),(8,8),(8,8,8),(8,8,8,8),(8,8,8,8,8), (16),(16,16),(16,16,16),(16,16,16,16),(16,16,16,16,16), (32),(32,32),(32,32,32),(32,32,32,32),(32,32,32,32,32), (64),(64,64),(64,64,64),(64,64,64,64),(64,64,64,64,64), (128),(128,128),(128,128,128),(128,128,128,128),(128,128,128,128,128), (256),(256,256),(256,256,256),(256,256,256,256),(256,256,256,256,256), (512),(512,512),(512,512,512),(512,512,512,512),(512,512,512,512,512)	50
Aktivacija	ReLU, Identitet, Sigmoid, TanH	4
Rješavač	Adam, LBFGS	2
Vrsta stope učenja	Konstantna, Adaptivna, inverzno skalirajuća	3
Početna stopa učenja	0.5, 0.1, 0.01, 0.00001	4
L2 regularizacija (alfa)	0.1, 0.01, 0.001, 0.0001	4
Ukupno		19200

3.3. SMOTE

U 1990-ima, kako je sve više podataka i aplikacija strojnog učenja i rudarenja podataka počelo prevladavati, pojavio se važan izazov: kako postići željenu točnost klasifikacije kada se radi s podacima koji su imali značajno iskrivljenu distribuciju klasa. Autori iz nekoliko disciplina uočili su neočekivano ponašanje standardnih algoritama klasifikacije nad skupovima podataka s neravnomjernom distribucijom klasa (Anand, Mehrotra, Mohan i Ranka, 1993.; Bruzzone i Serpico, 1997.; Kubat, Holte i Matwin, 1998.). U mnogim slučajevima je lokalna točnost, na primjerima većinske klase, nadjačala onu postignutu na manjinskim. Značaj ovog područja istraživanja nastavlja rasti uglavnom potaknut izazovnim izjavama problema iz različitih područja primjene (kao što je prepoznavanje lica, softversko inženjerstvo, društveni mediji, društvene mreže i medicinska dijagnoza), pružajući novi i suvremeni niz izazova istraživačima strojnog učenja i znanosti o podacima (Krawczyk, 2016.; Haixiang et al., 2017.; Maua i Galinac Grbac, 2017.; Zhang et. al., 2017.; Zuo i sur., 2016.; Lichtenwalter i sur., 2010.; Krawczyk i sur., 2016.; Bach i sur., 2017.; Cao i sur., 2017.a). Sveobuhvatno pitanje koje su istraživači pokušavali riješiti je: kako pomaknuti granice

predviđanja prema podzastupljenim ili manjinskim klasama, a istovremeno upravljati kompromisom s lažno pozitivnim rezultatima? Kao rezultat toga, istraživači su razvili metode prekomjernog uzorkovanja koje možda neće dovesti do smanjenja primjera većinske klase i rješavanje problema neravnoteže klasa repliciranjem primjera manjinske klase. Međutim, primjena nasumičnog prekomjernog uzorkovanja samo implicira veću težinu ili trošak za manjinske instance.

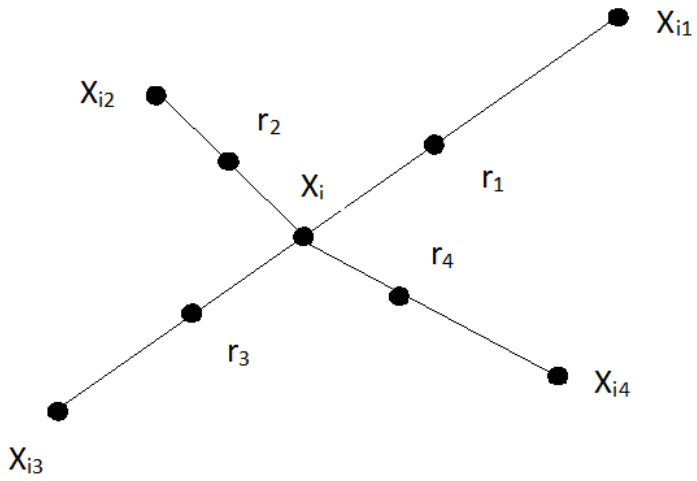
Godine 2002. Chawla, Bowyer, Hall i Kegelmeyer (2002.) predložili su novi pristup kao alternativu standardnom nasumičnom prekomjernom uzorkovanju. Ideja je bila prevladati prekomjerno uzorkovanje replikacijom i pomoći klasifikatoru da poboljša svoju generalizaciju podataka testiranja. Umjesto "ponderiranja" podatkovnih točaka, temelj ove nove tehnike predobrade podataka bilo je stvaranje novih manjinskih instanci. Ova tehnika je nazvana Tehnika sintetskog manjinskog prekomjernog uzorkovanja, sada naširoko poznata kao SMOTE. Osnova postupka SMOTE bila je provođenje interpolacije među instancama susjednih manjinskih klasa. Kao takav, sposoban je povećati broj primjera manjinske klase uvođenjem novih primjera manjinske klase u susjedstvu, čime pomaže klasifikatorima da poboljšaju svoju sposobnost generalizacije. Tehnika pretprocesiranja SMOTE postala je pionir za istraživačku zajednicu u neuravnoteženoj klasifikaciji. Od izdavanja predložena su mnoga proširenja i alternative za poboljšanje performansi u različitim scenarijima. Zbog svoje popularnosti i utjecaja, SMOTE se smatra jednim od najpopularnijih značajnih algoritama za pretprocesiranje/uzorkovanje podataka u strojnom učenju i rudarenju podataka. Neki pristupi kombiniraju SMOTE s tehnikama čišćenja podataka. Drugi se autori usredotočuju na unutarnju proceduru modificirajući neke njezine komponente poput odabira instanci za generiranje novih podataka ili vrsti interpolacije.

Chawla je radio na razvoju algoritma klasifikacije za učenje i predviđanje kancerogenih piksela mamografskih podataka. Osnovni klasifikator stabla odlučivanja dao mu je točnost od oko 97%. Njegova prva reakcija bila je slavlje, jer je postigao preko 97% točnosti na problemu koji mu je predstavljen kao izazov. Brzo je shvatio da bi samim pogađanjem većinske klase postigao točnost od 97,68% (što je bila distribucija većinske klase u izvornim podacima). Dakle, on je zapravo prošao lošije od klasifikatora nagađanja većinske klase. Štoviše, klasifikator stabla odlučivanja imao je loše rezultate u važnom zadatku ispravnog predviđanja kalkulacija. Popratni izazov bila je niska tolerancija na lažno pozitivne, tj. primjere većinske klase identificirane kao manjinske. Odnosno, trebalo je postići odgovarajući kompromis između istinski i lažno pozitivnih rezultata, a ne samo biti pretjerano agresivan u predviđanju manjinske klase (kancerogeni pikseli) kako bi se kompenzirala distribucija od 2,32%. To je bilo zato što su postojali troškovi povezani s pogreškama - svaki lažno negativan rezultat nosio je teret pogrešnog klasificiranja raka kao ne-raka, a svaki lažno pozitivan nosio je trošak dodatnih testova pogrešnom klasifikacijom ne-raka kao raka. Pogreške očito nisu bile iste vrste. Tijekom daljnjeg istraživanja primijetio je izazov koji proizlazi iz pretjeranog postavljanja instanci manjinske klase zbog prevelikog uzorkovanja. Ovo zapažanje dovelo je do pitanja: kako poboljšati sposobnost generalizacije osnovnog razreda? I tako je stvoren SMOTE za sintetsko generiranje novih instanci za pružanje novih informacija algoritmu učenja kako bi se poboljšala njegova predvidljivost o instancama manjinske klase. SMOTE je pru-

žio statistički značajno bolju izvedbu na mamografskim podacima, kao i nekoliko drugih, čime je postavio temelje za učenje iz neuravnoteženih skupova podataka. Naravno, SMOTE, kao i drugi pristupi uzorkovanju, suočava se s izazovom količine uzorkovanja, koji su Chawla i njegovi kolege također pokušali ublažiti razvojem okvira omotača, sličnog odabiru značajki.

Algoritam SMOTE provodi pristup prekomjernog uzorkovanja kako bi ponovno uravnotežio izvorni set za obuku. Umjesto primjene jednostavne replikacije instanci manjinske klase, ključna ideja SMOTE-a je uvođenje sintetičkih primjera. Ovi novi podaci nastaju interpolacijom između nekoliko instanci manjinske klase koje su unutar definiranog susjedstva. Iz tog razloga, kaže se da je postupak usmjeren na "prostor značajki", a ne na "prostor podataka". Drugim riječima, algoritam se temelji na vrijednostima značajki i njihovom odnosu, umjesto razmatranja podatkovnih točaka u cjelini. To je također dovelo do proučavanja teorijskog odnosa između izvornih i sintetičkih instanci koje treba dubinski analizirati, uključujući dimenzionalnost podataka. Moraju se uzeti u obzir neka svojstva kao što su varijanca i korelacija u prostoru podataka i značajki, kao i odnos između distribucije primjera obuke i testa.

Jednostavan primjer SMOTE-a prikazan je slici 3.9 . Instanca x_i manjinske klase odabrana je kao osnova za stvaranje novih sintetičkih podataka. Na temelju metrike udaljenosti, nekoliko najbližih susjeda iste klase (točke x_{i1} do x_{i4}) bira se iz skupa za obuku. Na kraju se provodi nasumična interpolacija kako bi se dobile nove instance r_1 do r_4.



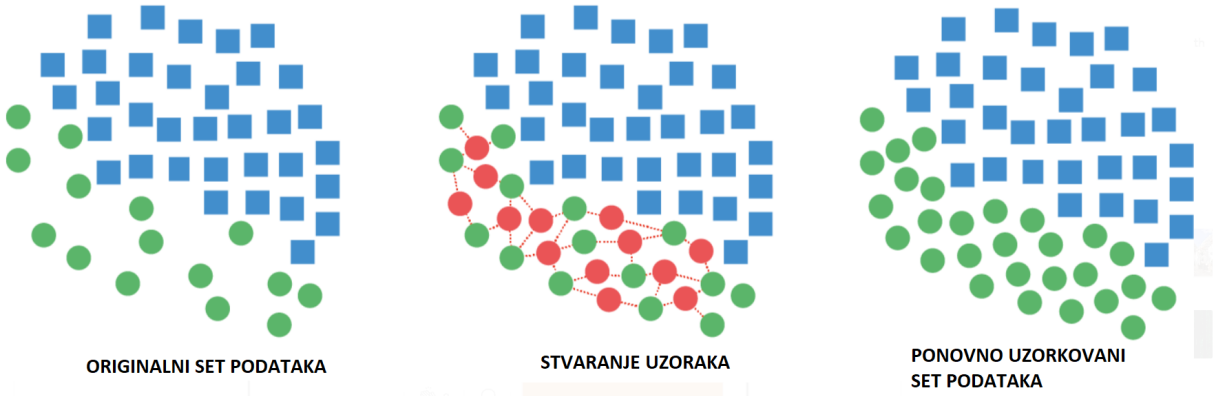
Slika 3.9. Ilustracija kako stvoriti sintetičke podatkovne točke u algoritmu SMOTE

Formalni postupak funkcionira na sljedeći način. Prvo se postavlja ukupna količina prekomjernog uzorkovanja N (cjelobrojna vrijednost), koja se može postaviti tako da se dobije približna distribucija klase 1:1 ili otkriti putem procesa omotača. Zatim se provodi iterativni proces koji se sastoji od nekoliko koraka. Najprije se iz skupa za obuku nasumično odabire instanca manjinske klase. Zatim se dobiva njegovih K najbližih susjeda (5 prema zadanim postavkama). Konačno, Nod ovih K instanci nasumično se odabire za izračun novih instanci interpolacijom. Da bi se to učinilo, uzima se razlika između vektora obilježja (uzorka) koji se razmatra i svakog od odabranih

susjeda. Ta se razlika množi nasumičnim brojem izvučenim između 0 i 1, a zatim se dodaje prethodnom vektoru obilježja. To uzrokuje odabir nasumične točke duž "linije" između značajki. U slučaju nominalnih atributa, jedna od dvije vrijednosti odabire se slučajno.

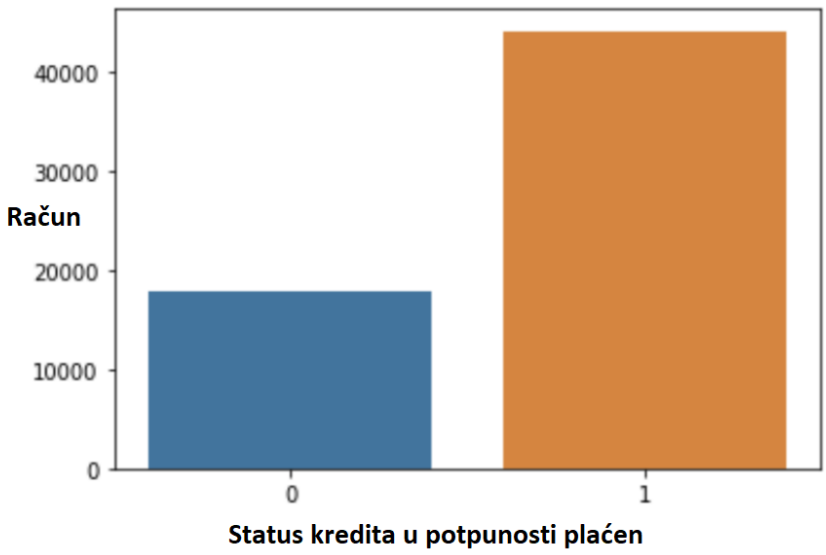
Algoritam rada SMOTE-a

Na slici 3.10 prikazan je algoritam rada SMOTE-a. To je jedan od načina ispravljanja neravnoteže podataka stvaranjem sintetičkih podataka sličnih izvornim podacima, drugim riječima prekomjerno uzorkovanje.



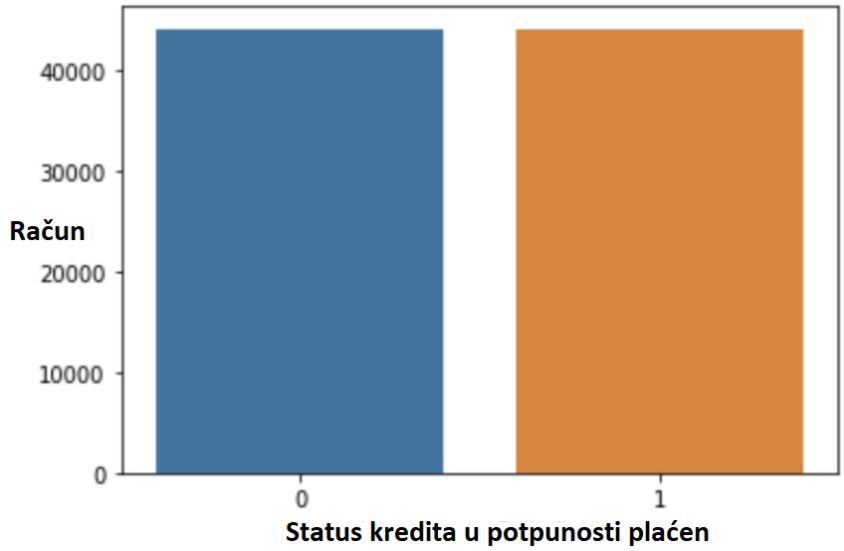
Slika 3.10. Ilustracija kako stvoriti sintetičke podatkovne točke u algoritmu SMOTE

Na slici 3.11 prikazan je neuravnotežen set podataka. Neuravnoteženost se može uočiti po tome što postoji više vrijednosti u jednoj klasi nego u drugoj. Tako se može uzrokovati problem visoke pristranosti, stvarajući više pogrešaka tipa 1 i/ili pogrešaka tipa 2.



Slika 3.11. Prikaz neuravnoteženih podataka

SMOTE će omogućiti da se pravilno uravnoteže podaci kako bi se tako pomoglo u budućim algoritmima strojnog učenja, kao što je prikazano na slici 3.12



Slika 3.12. Prikaz podataka nakon korištenja SMOTE

Korištenjem SMOTE dobiva se uklapanje podataka tj. sada se dobije jednak broj vrijednosti u obje klase.

3.4. Unakrsna validacija

Unakrsna validacija (eng. Cross-validation) je bilo koja od različitih tehnika validacije modela za procjenu kako će se rezultati statističke analize generalizirati na neovisni skup podataka. Ono je metoda ponovnog uzorkovanja kod koje se koriste različiti dijelovi podataka za testiranje i treniranje modela u različitim iteracijama. Uglavnom se koristi za predviđanje, odnosno ondje gdje se želi procijeniti kako će se predviđeni model ponašati u praksi. Kod predviđanja, modelu se obično daje skup poznatih podataka na kojima se provodi treniranje i skup nepoznatih podataka tj. prvi put viđeni podaci, prema kojima se model testira. Cilj unakrsne validacije je testiranje sposobnosti modela da predvidi neke nove podatke koji nisu korišteni pri procjeni te kako bi se tako označili problemi kao npr. prekomjerno prilagođavanje ili pristranost odabira i kako bi se dobio uvid kako će se model generalizirati na neovisnom skupu podataka. Unakrsna validacija uključuje dijeljenje podataka u komplementarne podskupove, nadalje, izvođenje analize na jednom podskupu (skup za treniranje) te provjeru valjanosti analize na drugom podskupu (skup za testiranje). Kako bi se smanjila promjenjivost, u većini metoda, višestruki krugovi unakrsne validacije izvode se korištenjem različitih podjela te se rezultati validacije kombiniraju (npr. prosjek) kako bi se dobila procjena predviđene izvedbe modela.

Tipičan primjer iz bioinformatike je skup podataka o ekspresiji gena temeljen na podacima mikronizova DNK, gdje svaki slučaj predstavlja jedan označeni uzorak tumora opisan profilom

ekspresije gena. Jedan od uobičajenih izazova se odnosi na razvoj klasifikatora koji može pouzdano predvidjeti klasu novog, neviđenog uzorka tumora na temelju njihovih profila ekspresije.

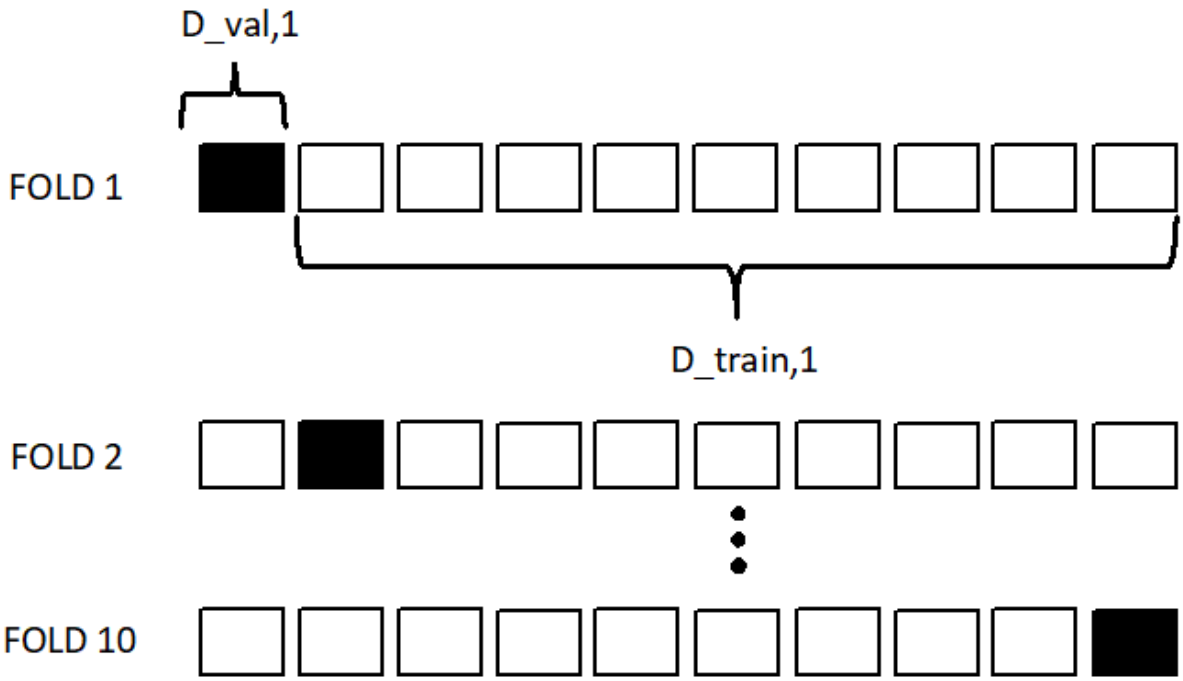
3.4.1. K-struka unakrsna validacija

U k-strukoj unakrsnoj validaciji (eng. K-fold cross-validation), izvorni uzorak se nasumično dijeli na k disjunktnih poduzoraka jednake veličine. Od k poduzoraka, jedan poduzorak se zadržava kao validacijski podatak za testiranje modela, a preostali $k-1$ poduzorci se koriste kao podaci za treniranje. Proces unakrsne validacije zatim se ponavlja k puta, pri čemu se svaki od k poduzoraka koristi točno jednom kao validacijski podatak. Prosjek performansi k mjerenja na k setovima za provjeru validacije je unakrsno validirana izvedba. Slika 3.13 ilustrira ovaj proces za $k = 10$, tj. 10-struka unakrsna provjera valjanosti gdje je skup podataka nasumično podijeljen u deset disjunktnih podskupova, od kojih svaki sadrži (približno) 10% podataka. Model se uvježbava na skupu za vježbanje i zatim primjenjuje na skup za validaciju.

U prvom krugu, prvi podskup služi kao validacijski skup $D_{val,1}$, a preostalih devet podskupova služi kao skup za obuku $D_{train,1}$. U drugom krugu, drugi podskup je validacijski skup, a preostali podskupovi su skup za obuku itd. Unakrsno validirana točnost, na primjer, prosjek je svih deset točnosti postignutih na validacijskim setovima. Općenitije, neka f_{-k} označava model koji je treniran na svim osim k^{th} podskupu iz skupa za učenje. Vrijednost $y_i = f_{-k}(x_i)$ je predviđena ili procijenjena vrijednost za oznaku stvarne klase, y_i, slučaja x_i, koji je element k^{th} podskupa. Unakrsno validirana procjena pogreške predviđanja,$\in_{cv}$, tada se daje kao

$$\in_{cv} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\left(y_i, f_{-k}\left(x_i\right)\right) \tag{3.5}$$

Unakrsna provjera valjanosti često uključuje slojevito slučajno uzorkovanje, što znači da se uzorkovanje provodi na takav način da omjeri razreda u pojedinačnim podskupovima odražavaju omjere u skupu za učenje.



Slika 3.13. 10-struka unakrsna provjera validacije. [10]

Na primjer, pretpostavimo da skup za učenje sadrži $n = 100$ slučajeva dviju klasa, pozitivne i negativne klase, s $n_+ = 80$ i $n_- = 20$. Ako se nasumično uzorkovanje provodi bez raslojavanja, tada je sasvim moguće da neki validacijski skupovi sadrže samo pozitivne slučajeve (ili samo negativne slučajeve). S raslojavanjem, svaki validacijski set u 10-strukoj unakrsnoj validaciji zajamčeno sadrži oko osam pozitivnih slučajeva i dva negativna slučaja, odražavajući tako omjer klasa u skupu za učenje. Temeljni razlog za slojevito uzorkovanje je sljedeće. Udio uzorka je nepristrana procjena udjela stanovništva. Skup za učenje predstavlja uzorak iz populacije od interesa, tako da je omjer klasa u skupu za učenje najbolja procjena omjera klasa u populaciji. Kako bi se izbjegla pristrana procjena, podskupovi podataka koji se koriste za procjenu modela bi također trebali odražavati ovaj omjer klasa.

Prednost ove metode je u tome što se sva promatranja koriste i za treniranje i za validaciju, a svako se promatranje koristi za validaciju točno jednom. Obično se koristi deseterostruka unakrsna provjera validacije, ali općenito k ostaje nefiksan parametar.

Na primjer, postavljanje $k = 2$ rezultira dvostrukom unakrsnom provjerom validacije. U dvostrukoj unakrsnoj provjeri, nasumično miješamo skup podataka u dva skupa d_0 i d_1, tako da su oba skupa jednake veličine. Zatim treniramo na d_0 i validiramo na d_1 , nakon čega slijedi treniranje na d_1 i validacija na d_0 .

3.5. Evaluacija

Evaluacijske metrike (eng. Evaluation metrics) koriste se za mjerenje kvalitete statističkog modela ili modela strojnog učenja. Procjena modela ili algoritama strojnog učenja ključna je za svaki projekt. Postoji mnogo različitih vrsta mjernih podataka za procjenu dostupnih za testiranje modela. To uključuje točnost klasifikacije, logaritamski gubitak, matricu konfuzije i druge. Ideja izgradnje modela strojnog učenja funkcionira na principu konstruktivne povratne informacije. Izrađuje se model, dobivaju se povratne informacije iz metrike, radi se poboljšanje i nastavlja se dok se ne postigne željena točnost. Mjerila evaluacije objašnjavaju izvedbu modela. Važan aspekt metrike procjene je njihova sposobnost razlikovanja rezultata modela.

3.5.1. Točnost

Točnost (eng. accuracy) je metrika procjene koja omogućuje mjerenje ukupnog broja predviđanja koja model dobiva točnima tj. ono je udio istinitih rezultata u ukupnom broju ispitanih slučajeva. Točnost se koristi za probleme klasifikacije koji su dobro uravnoteženi i nisu iskrivljeni ili nemaju neravnotežu klase.

Postoje četiri varijante pogrešaka:
- Lažno negativni (eng. false negative - FN) koji označavaju one podatke za koje je model netočno predvidio negativnu klasu.
- Lažno pozitivni (eng. false positive - FP) koji označavaju one podatke za koje je model netočno predvidio pozitivnu klasu.
- Istinski negativni (eng. true negative - TN) koji označavaju one podatke za koje je model točno predvidio negativnu klasu.
- Istinski pozitivni (eng. true positive - TP) koji označavaju one podatke za koje je model točno predvidio pozitivnu klasu.

Iz prethodno navedenog slijedi formula za točnost:

$$Acc = \frac{TN + TP}{TP + FP + TN + FN} \tag{3.6}$$

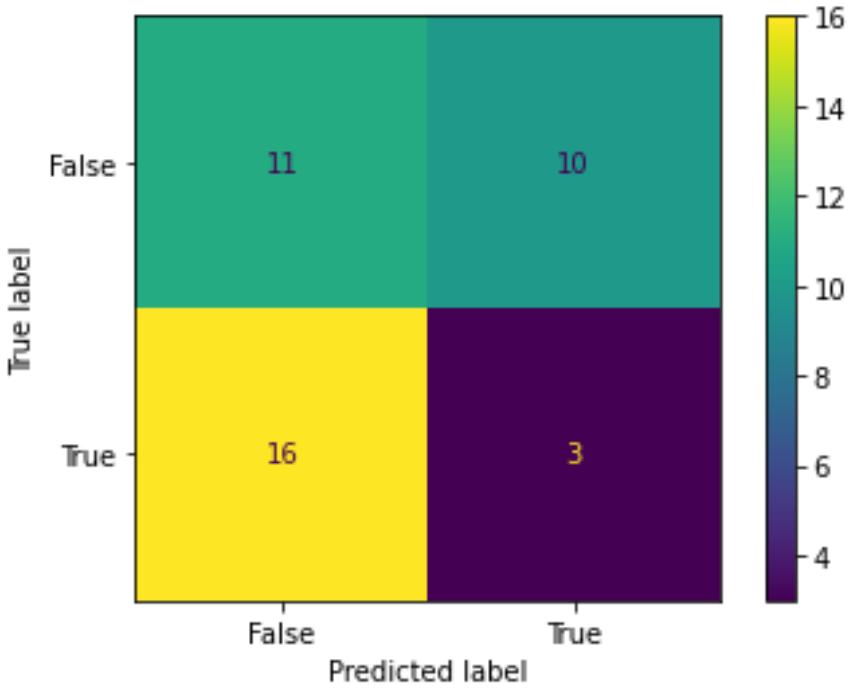
Točnost odgovara na pitanje koji je postotak predviđanja modela bio točan. Ono gleda na istinski pozitivne i istinske negativne strane. Tablica 3.5 prikazuje strukturu matrice konfuzije.

Tablica 3.2. Struktura matrice konfuzije

		STVARNO	
		POZITIVNO	NEGATIVNO
PREDVIĐENO	POZITIVNO	Istinski pozitivno	Lažno pozitivno
	NEGATIVNO	Lažno negativno	Istinski negativno

Na primjer matrica konfuzije prikazuje broj istinski pozitivnih, lažno pozitivnih, istinski negativnih i lažno negativnih rezultata koje je proizveo model. Pomoću matrice konfuzije može se

dobiti vrijednosti potrebne za izračunavanje točnosti modela. Gledajući matricu konfuzije na slici 3.14, i korištenjem prethodno navedene formule može se izračunati točnost.

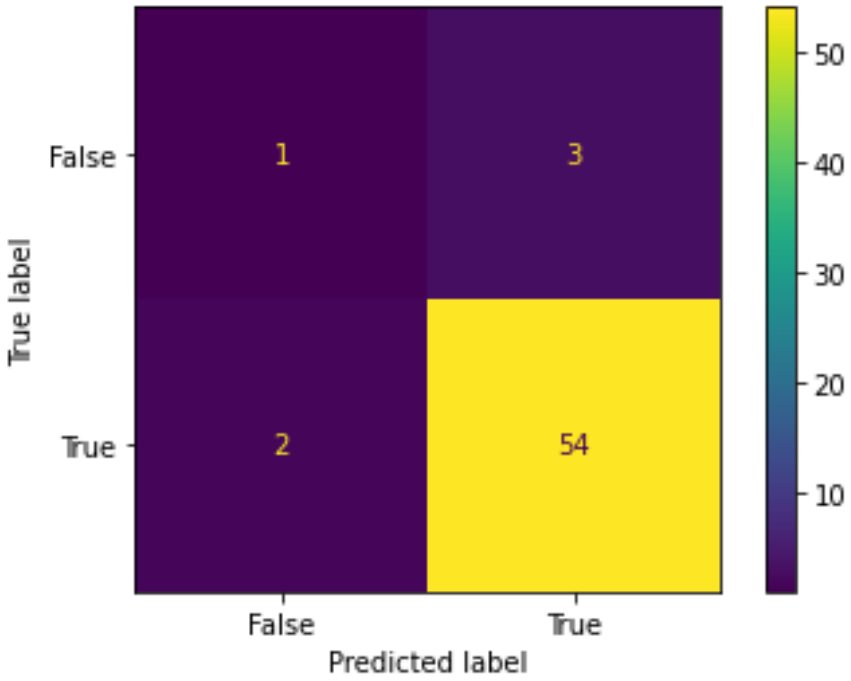


Slika 3.14. Matrica konfuzije

$$Acc = \frac{11+3}{3+10+11+16} = \frac{14}{40} = 0.35 \tag{3.7}$$

Točnost od 0.35 prilično je niska, što znači da model ne predviđa točno mnoge istinske pozitivne ili istinske negativne rezultate.

Nadalje, gledajući matricu konfuzije na slici 3.15 i korištenjem formule za točnost dobiju se sljedeći rezultati.



Slika 3.15. Matrica konfuzije

$$Acc = \frac{54+1}{54+3+1+2} = \frac{55}{60} = 0.91666 \tag{3.8}$$

Točnost od 0.91666 je vrlo visoka. Može se doći u iskušenje da se odabere model koji ima visoku točnost, ali treba razmisliti o tome. Možda je izgrađen model koji predviđa je li netko bio bolestan ili ne. Ako većina ljudi nema bolest i model svaki put predviđa negativno kao u gornjem primjeru, postići će se visoka točnost. Međutim, ovaj model nikada neće prepoznati bolest ni kod koga i stoga bi bio beskoristan u predviđanju ima li netko bolest. Kada točnost nije dobra metrika za procjenu modela koriste se druge metrike.

3.5.2. F1-rezultat

F1-rezultat kombinira preciznost i opoziv klasifikatora u jednu metriku uzimajući njihovu harmonijsku sredinu. Prvenstveno se koristi za usporedbu performansi dvaju klasifikatora. Pretpostavimo da klasifikator A ima veće pamćenje, a klasifikator B ima veću preciznost. U ovom slučaju, F1-rezultati za oba klasifikatora mogu se koristiti za određivanje koji daje bolje rezultate.

F1-rezultat klasifikacijskog modela izračunava se na sljedeći način:

$$\frac{2\,(P \cdot R)}{P+R}, \tag{3.9}$$

gdje je:
P-preciznost
R- opoziv na model klasifikacije.

Mikro i makro prosjeci predstavljaju dva načina tumačenja matrica konfuzije u postavkama više klasa. Potrebno je izračunati matricu konfuzije za svaku klasu $g_i \in G = \{1, \ldots, K\}$ tako da i-ta matrica konfuzije smatra klasu g_i pozitivnom klasom, a sve ostale klase g_j s $j \neq i$ kao negativne klase. Budući da svaka matrica konfuzije objedinjuje sva opažanja označena klasom koja nije g_i kao negativnu klasu, ovaj pristup dovodi do povećanja broja pravih negativnosti, osobito ako postoji mnogo klasa.

Da bi se objasnilo zašto je porast istinski negativnih rezultata problematičan, zamislit će se da postoji 10 klasa s po 10 opažanja. Tada matrica konfuzije za jednu od klasa može imati strukturu kao što je prikazano u tablici 3.3:

Tablica 3.3. Struktura matrice konfuzije

Predviđanje/ Referenca	Klasa 1	Druge klase
Klasa 1	8	10
Druge klase	2	80

Na temelju matrice 3.3 specifičnost bi bila $\frac{80}{80+10} = 88.9\%$ iako je klasa 1 bila točno predviđena samo u 8 od 18 slučajeva (preciznost 44,4%). Stoga, budući da je negativna klasa dominantna, specifičnost postaje napuhana. Stoga su mikro i makro prosjeci definirani samo za F1 rezultat, a ne za uravnoteženu točnost, koja se oslanja na stvarnu negativnu stopu. U nastavku će se koristiti TP_i, FP_i i FN_i za označavanje istinski pozitivnih, lažno pozitivnih i lažno negativnih rezultata u matrici zabune povezanoj s i-tom klasom. Neka je u ovom primjeru preciznost označena sa P, a opoziv sa R.

Mikroprosjek je dobio ime po činjenici da objedinjuje izvedbu na najmanjoj mogućoj jedinici (tj. na svim uzorcima). U nastavku su prikazane jednadžbe za preciznost (3.10) i opoziv (3.11).

$$P_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FP_i} \tag{3.10}$$

$$R_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FN_i} \tag{3.11}$$

Iz mirkoprosječne preciznosti (3.10) i opoziva (3.11) dolazi se do mikro F1-rezultata kao što je prikazano u jednadžbi (3.12).

$$F1_{micro} = \frac{2\left(P_{micro} \cdot R_{micro}\right)}{P_{micro} + R_{micro}} \tag{3.12}$$

Ako klasifikator dobije veliki $F1_{micro}$, to znači da općenito radi dobro. Mikroprosjek nije osjetljiv na prediktivni učinak za pojedinačne klase. Kao posljedica toga, mikroprosjek može biti posebno pogrešan kada je distribucija klase neuravnotežena.

S druge strane, makroprosjek je dobio ime po činjenici da iznosi prosjek za veće grupe, odnosno za učinak za pojedinačne klase, a ne zapažanja. U nastavku su napisane jednadžbe za makroprosječnu preciznost (3.13) i opoziv (3.14).

$$P_{macro} = \frac{1}{|G|} \cdot \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FP_i} = \frac{\sum_{i=1}^{|G|} P_i}{|G|} \tag{3.13}$$

$$R_{macro} = \frac{1}{|G|} \cdot \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FP_i} = \frac{\sum_{i=1}^{|G|} R_i}{|G|} \tag{3.14}$$

Iz markoprosječne preciznosti (3.13) i opoziva (3.14) dolazi se do makro F1-rezultata kao što je prikazano u jednadžbi (3.15).

$$F1_{macro} = \frac{2\,(P_{macro} \cdot R_{macro})}{P_{macro} + R_{macro}} \tag{3.15}$$

Ako $F1_{macro}$ ima veliku vrijednost, to znači da klasifikator radi dobro za svaku pojedinačnu klasu. Makroprosjek je stoga prikladniji za podatke s neuravnoteženom distribucijom klasa.

Zaključno, ako su podaci savršeno uravnoteženi, makro i mikro prosjek će rezultirati istim rezultatom. Nadalje, mikroprosječna preciznost i mikroprosječni opoziv jednaki su točnosti kada je svaka podatkovna točka dodijeljena točno jednoj klasi. Mikroprosječne metrike razlikuju se od ukupne točnosti kada su klasifikacije višestruko označene ili kada su neke klase isključene u slučaju više klasa. Budući da velika klasa radi bolje od malih, očekivalo bi se da će mikro prosjek biti viši od makro prosjeka. Mikroprosjek je poželjniji ako je klasa neuravnotežena. Ovisno o tome koji je cilj odabire se pogodniji. Ako se ne preferira niti jedna klasa, a bitni su ukupni podaci, mikro je sasvim u redu. No, recimo, klasa A je rijetka, ali je jako važna, makro bi trebao biti bolji izbor jer tretira svaku klasu jednako. 'Mikro' je s druge strane bolji ako je bitnija ukupna točnost. Mikro je bliži točnosti, dok je makro malo drugačiji kada njime ne dominira prevladavajuća klasa. U postavci klasifikacije s više klasa, mikroprosjek je poželjniji ako postoji sumnja da bi mogla postojati neravnoteža klasa.

3.5.3. AUC

Područje ispod ROC krivulje, ili jednostavno AUC, agregira ponašanje modela za sve moguće pragove odluke. Može se procijeniti pod parametarskim, poluparametarskim i neparametarskim pretpostavkama. Neparametarska procjena AUC-a široko se koristi kod strojnog učenja i rudarenja podataka. To je zbroj površina trapeza nastalih povezivanjem točaka na ROC krivulji i predstavlja vjerojatnost da će nasumično odabrana pozitivna instanca postići veći rezultat nego nasumično odabrana negativna instanca. Ekvivalentan je Wilcoxon-Mann-Whitneyevom (WMW) [14] statističkom testu rangova. Huang i Ling [15] također teoretski i empirijski pokazuju da je AUC bolja mjera za evaluacija modela nego točnost.

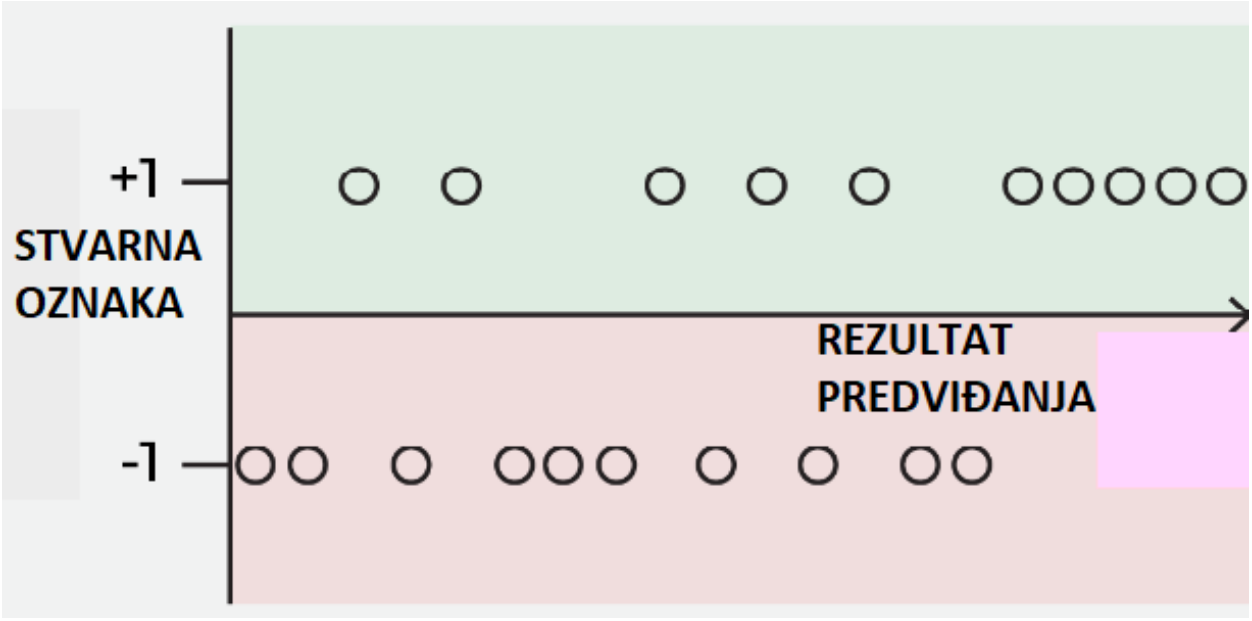
Neparametarska procjena AUC izračunava se na temelju rangova predviđenih rezultata. Njegova prednost je u tome što ne ovisi ni o kakvoj pretpostavci o distribuciji koja se obično zahtijeva u parametarskoj statistici. Njegov nedostatak je što se predviđeni rezultati koriste samo za rangiranje instanci, a inače se zanemaruju. AUC, procijenjen jednostavno iz redova predviđenih rezultata, može ostati nepromijenjen čak i kada se predviđeni rezultati promijene. To može dovesti do gubitka korisnih informacija i stoga može proizvesti rezultate koji nisu optimalni.

Jedno područje gdje AUC može biti osobito koristan je kada točnost nije dovoljna. Na primjer, ako model uvijek ne predviđa postojanje raka kada pokušava dijagnosticirati rijetku vrstu raka, model bi imao visoku točnost jer je gotovo uvijek točan (tj. gotovo nitko zapravo nema rak). S druge strane, model bi imao AUC vrijednost od 0,5 – što znači da je potpuno beskoristan (vrijednost od 0,5 proizlazi iz činjenice da bi takav model dao isti rezultat predviđanja za sve podatkovne točke).

Da bi se izračunao AUC, potreban je skup podataka s dva stupca: rezultat predviđanja i stvarna oznaka kao što je prikazano u 3.4. Budući da je stvarna oznaka binarna u ovom slučaju, koristi se +1 i -1 za označavanje pozitivne odnosno negativne klase. Ovdje je važno napomenuti da su stvarne oznake određene stvarnim svijetom, dok su rezultati predviđanja samo neki brojevi do kojih se dolazi, obično s modelom.

Tablica 3.4. Podaci

Rezultat predviđanja	Stvarna oznaka
7.412	-1
6.027	-1
4.916	+1
3.093	+1
7.594	-1
...	...



Slika 3.16. Vizualizacija podataka u Tablici 3.4

Kako bi se jasno vidio izračun AUC-a, prvo se vizualizira tablica 3.4 kao slika 3.16, gdje je svaka podatkovna točka iscrtana sa svojim rezultatom predviđanja na x-osi i stvarnom oznakom na y-osi. To se čini kako bi se podijelile podatkovne točke u dvije skupine identificirane njihovim stvarnim oznakama, jer je sljedeći korak generiranje predviđajućih oznaka i izračun dviju veličina prikazanih u tablici 3.5, čiji su nazivnici upravo brojevi podatkovnih točaka u ove dvije skupine.

Tablica 3.5. Količine za izračunavanje

Opis	Naziv	Statistički žargon
% pozitivnih stvarnih oznaka koje se slažu s oznakama predviđanja	Broj podatkovnih točaka s pozitivnim stvarnim oznakama	Istinski pozitivna stopa
% negativnih stvarnih oznaka koje se ne slažu s oznakama predviđanja	Broj podatkovnih točaka s negativnim stvarnim oznakama	Lažno pozitivna stopa

Da bi se generirale oznake predviđanja, odabire se proizvoljni granični iznos za rezultat predviđanja.

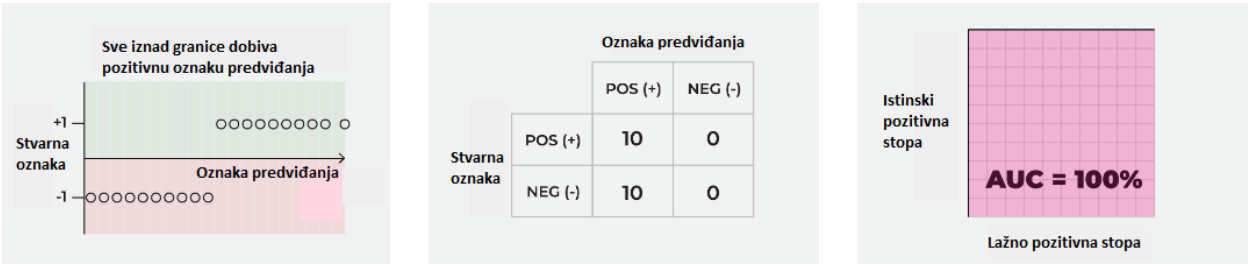
Podatkovne točke s rezultatima predviđanja iznad granice dobivaju pozitivne oznake predviđanja. Podatkovne točke s rezultatima predviđanja ispod granice dobivaju negativne oznake predviđanja. ROC krivulja je popis svih takvih pragova. Svaka točka na ROC krivulji odgovara jednoj od dvije veličine u tablici 3.5 koje možemo izračunati na temelju svake granične vrijednosti.

Tablica 3.6. Kako stvarna oznaka svake podatkovne točke utječe na kretanje ROC krivulje

Kada se granična vrijednost pomakne iznad podatkovne točke sa stvarnom oznakom koja je...	Pomak ROC krivulje
Pozitivna	Prema gore
Negativna	Udesno

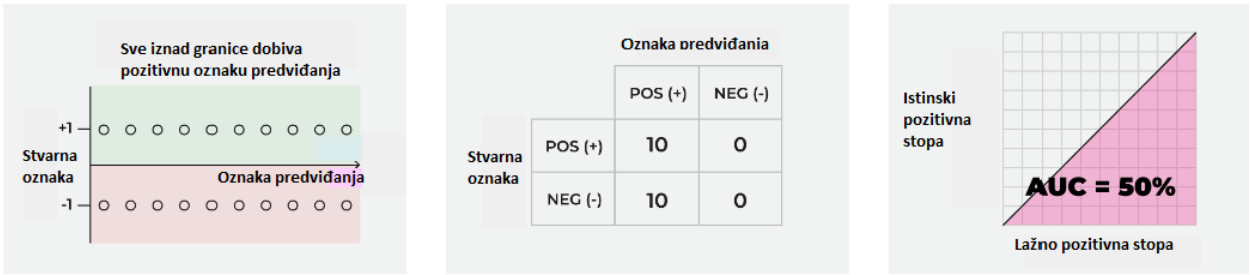
Ključni zaključak ovdje je da AUC mjeri stupanj razdvajanja između ove dvije grupe podatkovnih točaka – identificiranih njihovim stvarnim oznakama – kada su njihovi rezultati predviđanja ucrtani na os x. Tablica 3.6 ukratko prikazuje kako kretanje na ROC krivulji odgovara stvarnoj oznaci svake podatkovne točke, a slike 3.17 i 3.18 pokazuju kako AUC može biti 1 odnosno 0,5.Radi jednostavnosti, AUC se predstavlja kao površina ispod krivulje nalik na stepenice. U praksi se često primjenjuje dodatni korak zaglađivanja prije nego što se izračuna površina ispod njega.

Ako su dvije skupine savršeno odvojene svojim rezultatima predviđanja, tada je AUC = 1 i rezultat modela obavlja savršen posao razlikovanja pozitivnih stvarnih vrijednosti od negativnih stvarnih vrijednosti kao što je prikazano na slici 3.17.



Slika 3.17. Slika procesa izračuna AUC-a kada je AUC=1

Ako su dvije skupine savršeno izmiješane, tada je AUC = 0,5 i rezultati modela ne rade dobar posao razlikovanja pozitivnih stvarnih vrijednosti od negativnih stvarnih vrijednosti kao što je prikazano na slici 3.18.



Slika 3.18. Slika procesa izračuna AUC-a kada je AUC=0.5

AUC kod više klase računa se na isti način kao što je prethodno objašnjeno za mikro i makro-prosječne vrijednosti kod F1-rezultata.

4. Rezultati i diskusija

U ovom dijelu dan je pregled postignutih rezultata. Ovi rezultati postignuti su pomoću metodologije opisane u prethodnom odjeljku.

4.1. Ostvareni rezultati MLP algoritma bez SMOTE (micro)

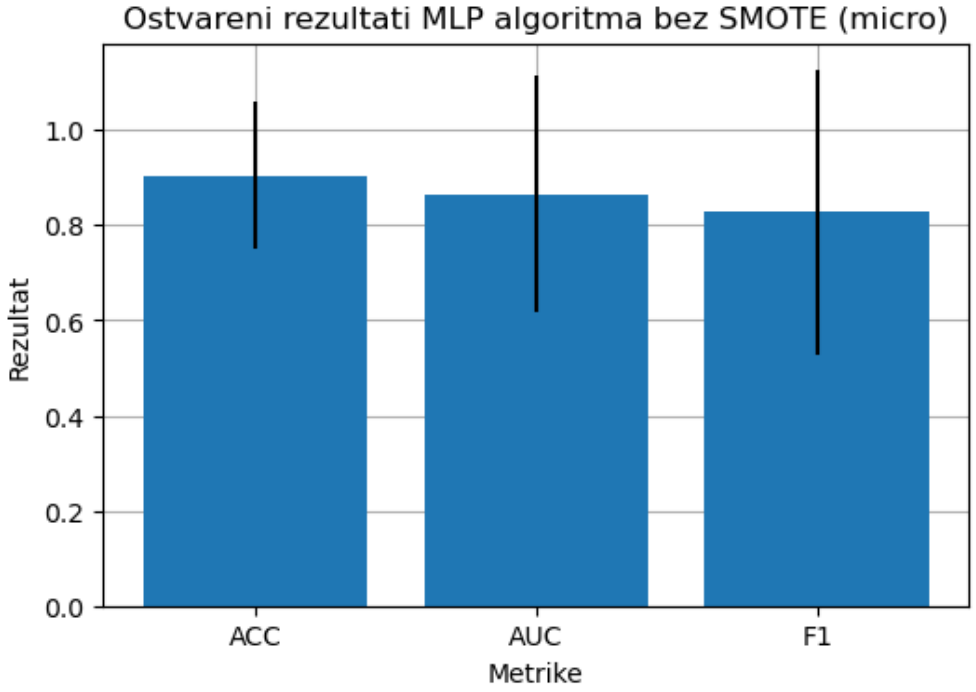
U tablici 4.1 prikazano je sedam konfiguracija koje postižu najbolje rezultate MLP algoritma bez SMOTE (micro). Kod prvog modela, točnost koja je postignuta je 0.905 sa standardnom devijacijom od 0.156. AUC i F1 metrike pokazuju vrijednost od 0.865 i 0.827 sa standardnim devijacijama 0.248 i 0.299. Ovi rezultati su ostvareni s aktivacijskom funkcijom tanh čija je stopa učenja 0.01, s 5 skrivenih slojeva od kojih svaki sadrži 32 neurona. Vrsta stope učenja je invscaling, a njena početna stopa iznosi 1e-05. Rješavač je Adam. Drugi model koji ima aktivacijsku funkciju logistic, sa stopom učenja 0.0001, kod kojeg postoje 3 skrivena sloja od kojih svaki sadrži 16 neurona. Kod ovog modela vrsta stope učenja je adaptive s početnom stopom od 0.1. Rješavač koji se koristi je LBFGS. Tako je postignuta točnost od 0.881 sa standardnom devijacijom od 0.217. AUC i F1 pokazuju vrijednost od 0.875 I 0.767 sa standardnim devijacijama 0.232 i 0.4. Kod trećeg modela metrike AUC i F1 pokazuju vrijednosti od 0.835 i 0.76 sa standardnim devijacijama 0.346 i 0.261 te je postignuta točnost od 0.876 sa standardnom devijacijom od 0.216. Ovi rezultati ostvareni su aktivacijskom funkcijom ReLu sa stopom učenja 0.01. Kod ovog modela postoje 4 skrivena sloja te se u svakom sloju nalazi 64 neurona. Vrsta stope učenja je invscaling sa stopom učenja 0.01, a rješavač je Adam. Kod četvrtog modela ostvarena je točnost od 0.876 sa standardnom devijacijom 0.216. Metrika AUC pokazuje vrijednost od 0.83 sa standardnom devijacijom 0.377, dok metrika F1 pokazuje vrijednost od 0.76 sa standardnom devijacijom 0.261. Kao aktivacijska funkcija, za dobivanje rezultata, korištena je ReLu funkcija čija je stopa učenja 0.01. Broj skrivenih slojeva je 5, a u svakom sloju nalazi se 64 neurona. Vrsta stope učenja je invscaling sa stopom učenja 1e-05, a korišteni rješavač je Adam. Peti model, za dobivanje rezultata, koriti ReLu aktivacijsku funkciju sa stopom učenja 0.01, broj skrivenih slojeva je 1, a svaki sloj ima 128 neurona. Korištena vrsta stope učenja, kod ovog modela, je constant s početnom stopom koja iznosi 1e-05. Korišteni rješavač je Adam. Ovim modelom postignuta je točnost od 0.876 sa standardnom devijacijom 0.245. Metrike F1 i AUC postigle su vrijednost od 0.667 i 0.895 sa standardnim devijacijama 0.730 i 0.310. Kod šestog modela postignuta je vrijednost točnosti 0.876 sa standardnom devijacijom 0.245. Metrike AUC i F1 ostvarile su vrijednost od 0.88 i 0.667 sa standardnim devijacijama 0.388 i 0.730. Rezultati su postignuti korištenjem ReLu aktivacijske funkcije sa stopom učenja 0.001. Postoje 5 skrivenih slojeva od kojih svaki ima 8 neurona. Kod ovog modela vrsta stope učenja je constant čija je početna stopa učenja 0.1. Rješavač je LBFGS. Sedmi model ostvaruje rezultate korištenjem ReLu aktivacijske funkcije čija je stopa učenja

0.0001. Broj skrivenih slojeva je 2 od kojih svaki sadrži po 256 neurona. Vrsta stope učenja koju koristi ovaj model je adaptive s početnom stopom učenja 1e-05, a rješavač koji se koristi je Adam. Na ovaj način postignuta je točnost od 0.876 sa standardnom devijacijom 0.125. Metrika AUC dostigla je vrijednost 0.785 sa standardnom devijacijom 0.279, a metrika F1 0.76 sa standardnom devijacijom 0.261.

Tablica 4.1. Ostvareni rezultati MLP algoritma bez SMOTE (micro)

Točnost	σ_{ACC}	**AUC**	σ_{AUC}	**F1**	σ_{F1}	**Hiperparametri**	
0.905	0.156	0.865	0.248	0.827	0.299	**Aktivacija**	Tanh
						Regularizacija	0.01
						Skriveni slojevi	5
						Neurona po sloju	32
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.881	0.217	0.875	0.232	0.767	0.4	**Aktivacija**	Logistic
						Regularizacija	0.0001
						Skriveni slojevi	3
						Neurona po sloju	16
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	0.1
						Rješavač	LBFGS
0.876	0.216	0.835	0.346	0.76	0.261	**Aktivacija**	ReLu
						Regularizacija	0.01
						Skriveni slojevi	4
						Neurona po sloju	64
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	0.01
						Rješavač	Adam
0.876	0.126	0.83	0.377	0.76	0.261	**Aktivacija**	ReLu
						Regularizacija	0.01
						Skriveni slojevi	5
						Neurona po sloju	64
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.876	0.245	0.895	0.310	0.667	0.730	**Aktivacija**	ReLu
						Regularizacija	0.01
						Skriveni slojevi	1
						Neurona po sloju	128
						Vrsta stope učenja	Constant
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.876	0.245	0.88	0.388	0.667	0.730	**Aktivacija**	ReLu
						Regularizacija	0.001
						Skriveni slojevi	5
						Neurona po sloju	8
						Vrsta stope učenja	Constant
						Početna stopa učenja	0.1
						Rješavač	LBFGS
0.876	0.125	0.785	0.279	0.76	0.261	**Aktivacija**	ReLu
						Regularizacija	0.0001
						Skriveni slojevi	2
						Neurona po sloju	256
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	1e-05
						Rješavač	Adam

Na slici 4.1 prikazan je graf koji predstavlja najbolji rezultat od svih modela prikazanih u tablici. Najbolji ostvareni rezultat MLP algoritma bez SMOTE (micro) je onaj s točnosti od 0.905 i standardnom devijacijom 0.156, tj. prvi opisani model iz ranije navedene tablice.



Slika 4.1. Najbolji ostvareni rezultati MLP algoritma bez SMOTE (micro)

4.2. Ostvareni rezultati MLP algoritma bez SMOTE (macro)

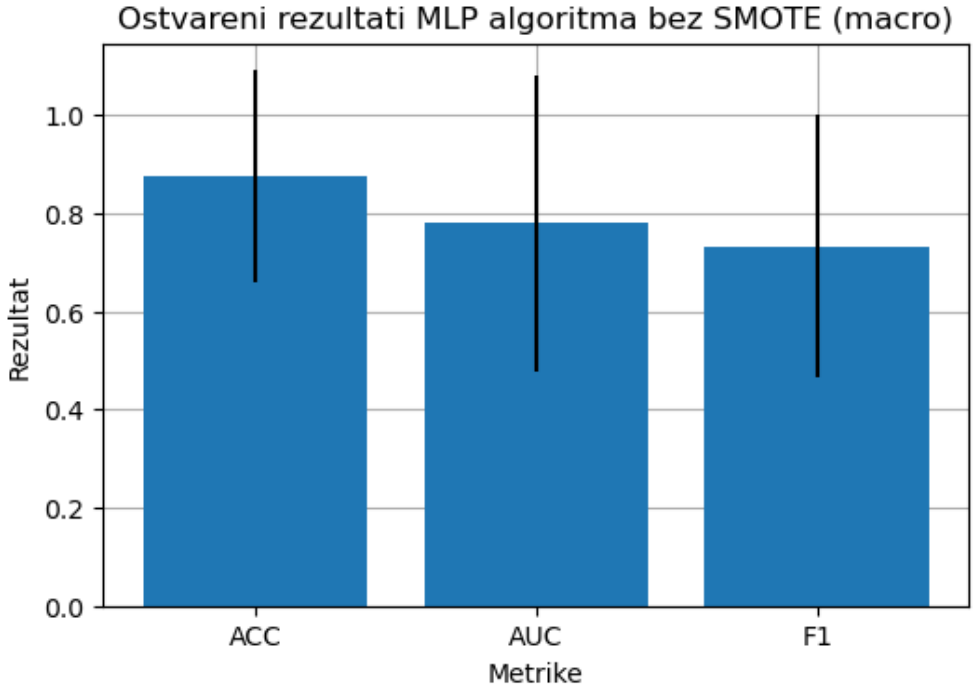
U tablici 4.2 prikazano je sedam konfiguracija koje postižu najbolje rezultate MLP algoritma bez SMOTE (macro). Kod prvog modela, točnost koja je postignuta je 0.876 sa standardnom devijacijom od 0.126. AUC i F1 metrike pokazuju vrijednost od 0.78 i 0.733 sa standardnim devijacijama 0.302 i 0.267. Ovi rezultati su ostvareni s aktivacijskom funkcijom ReLu čija je stopa učenja 0.001, s 2 skrivena sloja od kojih svaki sadrži 16 neurona. Vrsta stope učenja je adaptive, a njena početna stopa iznosi 0.01. Rješavač je LBFGS. Kod drugog modela, koji ima aktivacijsku funkciju ReLu sa stopom učenja 0.001 i koji ima 2 skrivena sloja od kojih svaki ima 64 neurona, postignuta je točnost od 0.876 sa standardnom devijacijom 0.126. Metrike AUC i F1 postigle su vrijednost od 0.825 i 0.76 sa standardnim devijacijama 0.374 i 0.261. Kod ovog modela vrsta stope učenja je invscaling s početnom stopom 1e-05, a rješavač je Adam. Treći model kao aktivacijsku funkciju koristi ReLu funkciju sa stopom učenja 0.001. Broj skrivenih slojeva je 2 i svaki sloj ima 128 neurona. Vrsta stope učenja kod ovog modela je constant s početnom stopom učenja 1e-5. Rješavač koji se koristi je Adam. Uz postavljene hiperparametre dobivena je točnost od 0.876 sa standardnom devijacijom 0.126. Metrika AUC poprima vrijednost 0.805 a standardnom devijacijom 0.344, dok metrika F1 postiže vrijednost 0.733 sa standardnom devijacijom 0.267. Četvrti model postigao je točnost od 0.876 sa standardnom devijacijom 0.126, dok su metrike

AUC i F1 postigle vrijednost od 0.825 i 0.76 sa standardnim devijacijama 0.276 i 0.261. Rezultati četvrtog modela postignuti su korištenjem ReLu aktivacijske funkcije čija je stopa učenja 0.001, a broj skrivenih slojeva je 1 od kojih svaki sadrži 512 neurona. Vrsta stope učenja je invscaling s početnom stopom učenja 1e-05 te je rješavač Adam. Peti model za postizanje rezultata koristi ReLu aktivacijsku funkciju sa stopom učenja 0.001. Broj skrivenih slojeva, u ovom slučaju, je 4, a svaki sloj sadrži 215 neurona. Korištena vrsta stope učenja je invscaling s početnom stopom učenja 1e-05 i rješavačem Adam. Metrika AUC dostignula je vrijednost od 0.8 sa standardnom devijacijom 0.318, dok je metrika F1 postigla vrijednost 0.76 sa standardnom devijacijom 0.261. Postignuta točnost petog modela je 0.876 sa standardnom devijacijom 0.216. Kod šestog modela postignuta je vrijednost točnosti 0.876 sa standardnom devijacijom 0.126. Metrike AUC i F1 ostvarile su vrijednost od 0.805 i 0.76 sa standardnim devijacijama 0.344 i 0.261. Rezultati su postignuti korištenjem logistic aktivacijske funkcije sa stopom učenja 0.0001. Postoje 3 skrivena sloja od kojih svaki ima 16 neurona. Kod ovog modela vrsta stope učenja je invscaling čija je početna stopa učenja 0.5. Rješavač je LBFGS. Sedmi model ostvaruje rezultate korištenjem tanh aktivacijske funkcije čija je stopa učenja 0.1. Broj skrivenih slojeva je 3 od kojih svaki sadrži po 64 neurona. Vrsta stope učenja koju koristi ovaj model je adaptive s početnom stopom učenja 1e-05, a rješavač koji se koristi je Adam. Na ovaj način postignuta je točnost od 0.876 sa standardnom devijacijom 0.126. Metrika AUC dostigla je vrijednost 0.8 sa standardnom devijacijom 0.318, a metrika F1 0.76 sa standardnom devijacijom 0.261.

Tablica 4.2. Ostvareni rezultati MLP algoritma bez SMOTE (macro)

Točnost	σ_{ACC}	**AUC**	σ_{AUC}	**F1**	σ_{F1}	**Hiperparametri**	
0.876	0.126	0.78	0.302	0.733	0.267	**Aktivacija**	ReLu
						Regularizacija	0.001
						Skriveni slojevi	2
						Neurona po sloju	16
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	0.01
						Rješavač	LBFGS
0.876	0.126	0.825	0.374	0.76	0.261	**Aktivacija**	ReLu
						Regularizacija	0.001
						Skriveni slojevi	2
						Neurona po sloju	64
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.876	0.126	0.805	0.344	0.733	0.267	**Aktivacija**	ReLu
						Regularizacija	0.001
						Skriveni slojevi	2
						Neurona po sloju	128
						Vrsta stope učenja	Constant
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.876	0.126	0.825	0.276	0.76	0.261	**Aktivacija**	ReLu
						Regularizacija	0.001
						Skriveni slojevi	1
						Neurona po sloju	512
						Vrsta stope učenja	Invscaling
						Početna stopa rate	1e-05
						Rješavač	Adam
0.876	0.126	0.8	0.318	0.76	0.261	**Aktivacija**	ReLu
						Regularizacija	0.001
						Skriveni slojevi	4
						Neurona po sloju	215
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.876	0.126	0.805	0.344	0.76	0.261	**Aktivacija**	Logistic
						Regularizacija	0.0001
						Skriveni slojevi	3
						Neurona po sloju	16
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	0.5
						Rješavač	LBFGS
0.876	0.126	0.8	0.318	0.76	0.261	**Aktivacija**	Tanh
						Regularizacija	0.1
						Skriveni slojevi	3
						Neurona po sloju	64
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	1e-05
						Rješavač	Adam

Na slici 4.2 prikazan je graf koji predstavlja najbolji rezultat od svih modela prikazanih u tablici. Najbolji ostvareni rezultat MLP algoritma bez SMOTE (macro) je onaj s točnosti od 0.876 i standardnom devijacijom 0.126, tj. prvi opisani model iz ranije navedene tablice.



Slika 4.2. Najbolji ostvareni rezultati MLP algoritma bez SMOTE (macro)

4.3. Ostvareni rezultati MLP algoritma sa SMOTE (micro)

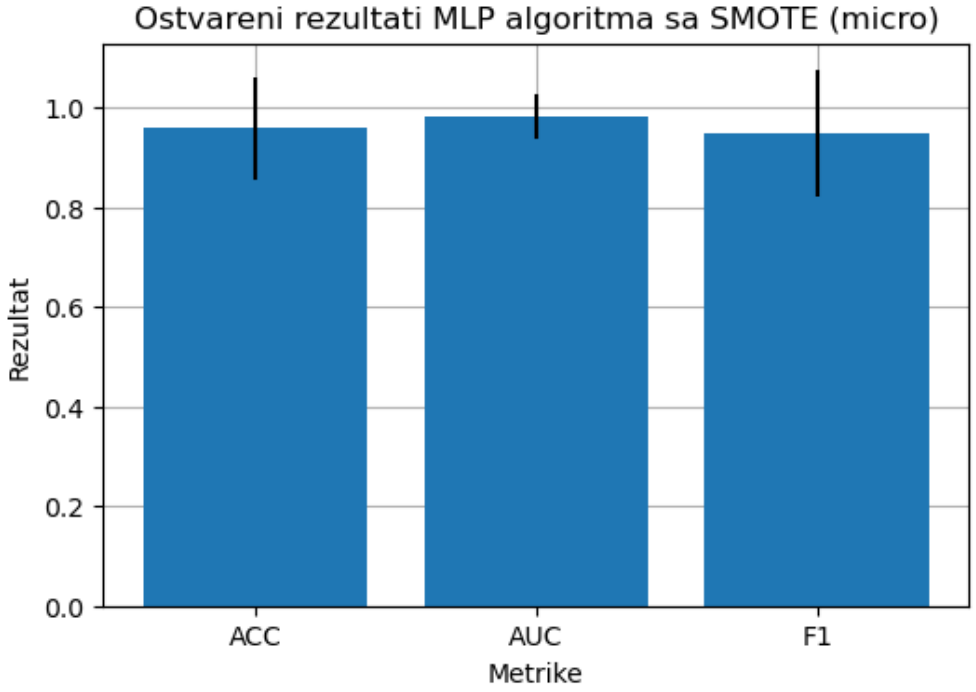
U tablici 4.3 prikazano je sedam konfiguracija koje postižu najbolje rezultate MLP algoritma sa SMOTE (micro). Kod prvog modela, točnost koja je postignuta je 0.958 sa standardnom devijacijom od 0.104. AUC i F1 metrike pokazuju vrijednost od 0.982 i 0.949 sa standardnim devijacijama 0.045 i 0.126. Ovi rezultati su ostvareni s aktivacijskom funkcijom tanh čija je stopa učenja 0.001, s 5 skrivenih slojeva od kojih svaki sadrži 64 neurona. Vrsta stope učenja je constant, a njena početna stopa iznosi 1e-05. Rješavač je Adam. Drugi model za ostvarivanje rezultata koristi identity aktivacijsku funkciju čija je stopa učenja 0.001, s 4 skrivena sloja od kojih svaki sadrži 1 neuron. Vrsta stope učenja je adaptive čija je početna stopa učenja 0.01, a rješavač je Adam. Treći model postiže točnost od 0.936 sa standardnom devijacijom 0.176. AUC metrika postiže vrijednost od 0.982 sa standardnom devijacijom 0.045, dok metrika F1 postiže vrijednost 0.944 sa standardnom devijacijom 0.141. Kod trećeg modela metrike AUC i F1 postignule su vrijednost od 0.974 i 0.944 sa standardnim devijacijama 0.066 i 0.141.Točnost je 0.936 sa standardnom devijacijom 0.176. Rezultati za treći model postignuti su identity aktivacijskom funkcijom čija je stopa učenja 0.0001. Broj skrivenih slojeva je 3, a svaki od njih sadrži po 16 neurona. Vrsta stope učenja je constant s početnom stopom učenja 0.01 i rješavačem Adam. Kod četvrtog modela kod metrika AUC i F1 ostvarene su vrijednosti od 0.952 i 0.944 sa standardnim devijacijama 0.155 i 0.141. Postignuta

točnost ovog modela je 0.936 sa standardnom devijacijom 0.176. Rezultati ovog modela ostvareni su identity aktivacijskom funkcijom sa stopom učenja 0.0001. Broj skrivenih slojeva je 5 od kojih svaki ima 64 neurona. Vrsta stope učenja je invscaling s početnom stopom učenja 0.41 i rješavač koji se koristi je LBFGS. Posljednji, tj. peti model ostvaruje točnost od 0.936 sa standardnom devijacijom 0.176, dok su metrike AUC i F1 postigle vrijednosti 0.98 i 0.944 sa standardnim devijacijama 0.08 i 0.141. Za dobivanje ovih rezultata korištena je identity aktivacijska funkcija koja ima stopu učenja 0.0001. Broj skrivenih slojeva je 4 i svaki od tih slojeva ima 128 neurona. Vrsta stope učenja kod ovog modela je adaptive s početnom stopom učenja 1e-05. Rješavač je Adam. Kod šestog modela postignuta je vrijednost točnosti 0.936 sa standardnom devijacijom 0.176. Metrike AUC i F1 ostvarile su vrijednost od 0.98 i 0.944 sa standardnim devijacijama 0.08 i 0.141. Rezultati su postignuti korištenjem logistic aktivacijske funkcije sa stopom učenja 0.0001. Postoji 1 skriveni sloj koji ima 16 neurona. Kod ovog modela vrsta stope učenja je invscaling čija je početna stopa učenja 1e-05. Rješavač je LBFGS. Sedmi model ostvaruje rezultate korištenjem tanh aktivacijske funkcije čija je stopa učenja 0.001. Broj skrivenih slojeva je 2 od kojih svaki sadrži po 64 neurona. Vrsta stope učenja koju koristi ovaj model je adaptive s početnom stopom učenja 1e-05, a rješavač koji se koristi je Adam. Na ovaj način postignuta je točnost od 0.936 sa standardnom devijacijom 0.176. Metrika AUC dostigla je vrijednost 0.97 sa standardnom devijacijom 0.12, a metrika F1 0.944 sa standardnom devijacijom 0.141.

Tablica 4.3. Ostvareni rezultati MLP algoritma sa SMOTE (micro)

Točnost	σ_{ACC}	**AUC**	σ_{AUC}	**F1**	σ_{F1}	**Hiperparametri**	
0.958	0.104	0.982	0.045	0.949	0.126	**Aktivacija**	Tanh
						Regularizacija	0.001
						Skriveni slojevi	5
						Neurona po sloju	64
						Vrsta stope učenja	Constant
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.936	0.176	0.982	0.045	0.944	0.141	**Aktivacija**	Identity
						Regularizacija	0.001
						Skriveni slojevi	4
						Neurona po sloju	1
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	0.01
						Rješavač	Adam
0.936	0.176	0.974	0.066	0.944	0.141	**Aktivacija**	Identity
						Regularizacija	0.0001
						Skriveni slojevi	3
						Neurona po sloju	16
						Vrsta stope učenja	Constant
						Početna stopa učenja	0.01
						Rješavač	Adam
0.936	0.176	0.952	0.155	0.944	0.141	**Aktivacija**	Identity
						Regularizacija	0.0001
						Skriveni slojevi	5
						Neurona po sloju	64
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	0.1
						Rješavač	LBFGS
0.936	0.176	0.98	0.08	0.944	0.141	**Aktivacija**	Identity
						Regularizacija	0.0001
						Skriveni slojevi	4
						Neurona po sloju	128
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.936	0.176	0.98	0.08	0.944	0.141	**Aktivacija**	Logistic
						Regularizacija	0.0001
						Skriveni slojevi	1
						Neurona po sloju	2
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	1e-05
						Rješavač	LBFGS
0.936	0.176	0.97	0.12	0.944	0.141	**Aktivacija**	Tanh
						Regularizacija	0.001
						Skriveni slojevi	2
						Neurona po sloju	64
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	1e-05
						Rješavač	Adam

Na slici 4.3 prikazan je graf koji predstavlja najbolji rezultat od svih modela prikazanih u tablici. Najbolji ostvareni rezultat MLP algoritma sa SMOTE (micro) je onaj s točnosti od 0.958 i standardnom devijacijom 0.104, tj. prvi opisani model iz ranije navedene tablice.



Slika 4.3. Najbolji ostvareni rezultati MLP algoritma sa SMOTE (micro)

4.4. Ostvareni rezultati MLP algoritma sa SMOTE (macro)

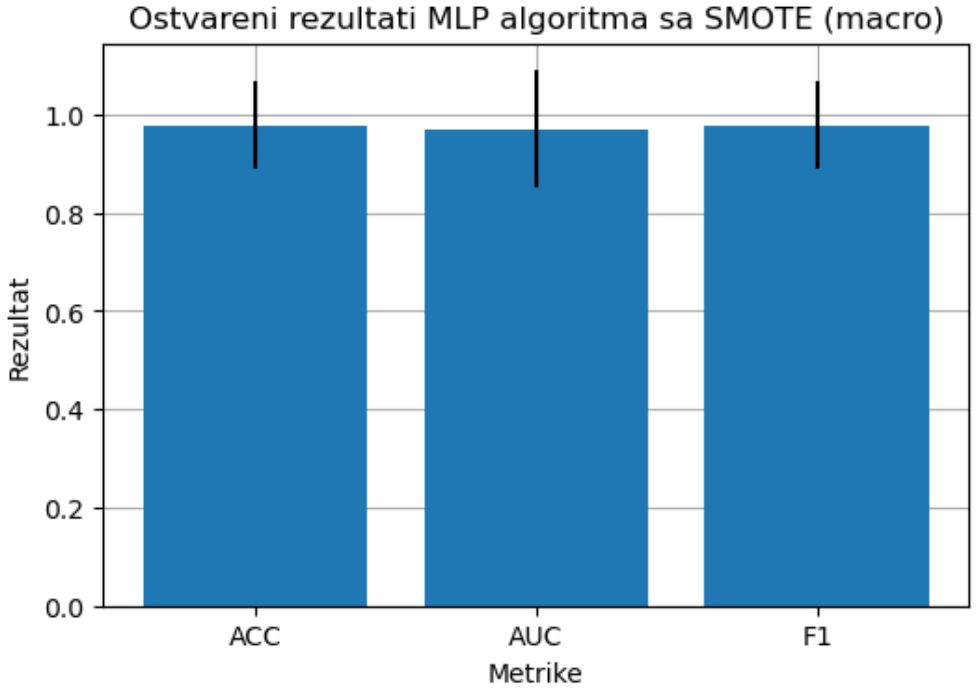
U tablici 4.4 prikazano je pet konfiguracija koje postižu najbolje rezultate MLP algoritma sa SMOTE (macro). Kod prvog modela, točnost koja je postignuta je 0.978 sa standardnom devijacijom od 0.089. AUC i F1 metrike pokazuju vrijednost od 0.97 i 0.978 sa standardnim devijacijama 0.12 i 0.089. Ovi rezultati su ostvareni s aktivacijskom funkcijom identity čija je stopa učenja 0.01, s 2 skrivena sloja od kojih svaki sadrži 512 neurona. Vrsta stope učenja je adaptive, a njena početna stopa iznosi 0.5. Rješavač je LBFGS. Drugi model koji ima aktivacijsku funkciju ReLu, sa stopom učenja 0.1, kod kojeg postoje 5 skrivenih slojeva od kojih svaki sadrži 128 neurona. Kod ovog modela vrsta stope učenja je adaptive s početnom stopom od 1e-05. Rješavač koji se koristi je Adam. Tako je postignuta točnost od 0.956 sa standardnom devijacijom od 0.178. AUC i F1 pokazuju vrijednost od 0.99 I 0.967 sa standardnim devijacijama 0.04 i 0.133. Kod trećeg modela metrike AUC i F1 pokazuju vrijednosti od 0.95 i 0.967 sa standardnim devijacijama 0.2 i 0.133 te je postignuta točnost od 0.956 sa standardnom devijacijom od 0.178. Ovi rezultati ostvareni su aktivacijskom funkcijom identity sa stopom učenja 0.001. Kod ovog modela postoje 4 skrivena sloja te se u svakom sloju nalazi 32 neurona. Vrsta stope učenja je adaptive sa stopom učenja 1e-05, a rješavač je Adam. Kod četvrtog modela ostvarena je točnost od 0.956 sa standardnom devijacijom 0.178. Metrika AUC pokazuje vrijednost od 0.93 sa standardnom devijacijom 0.28,

dok metrika F1 pokazuje vrijednost od 0.967 sa standardnom devijacijom 0.133. Kao aktivacijska funkcija, za dobivanje rezultata, korištena je identity funkcija čija je stopa učenja 0.0001. Broj skrivenih slojeva je 5, a u svakom sloju nalazi se po 2 neurona. Vrsta stope učenja je invscaling sa stopom učenja 0.5, a korišteni rješavač je LBFGS. Posljednji model, za dobivanje rezultata, koriti identity aktivacijsku funkciju sa stopom učenja 0.0001, broj skrivenih slojeva je 3, a svaki sloj ima 32 neurona. Korištena vrsta stope učenja, kod ovog modela, je invscaling s početnom stopom koja iznosi 0.1. Korišteni rješavač je LBFGS. Ovim modelom postignuta je točnost od 0.956 sa standardnom devijacijom 0.178. Metrike F1 i AUC postigle su vrijednost od 0.94 i 0.967 sa standardnim devijacijama 0.24 i 0.133. Kod šestog modela postignuta je vrijednost točnosti 0.956 sa standardnom devijacijom 0.178. Metrike AUC i F1 ostvarile su vrijednost od 0.98 i 0.967 sa standardnim devijacijama 0.08 i 0.133. Rezultati su postignuti korištenjem tanh aktivacijske funkcije sa stopom učenja 0.01. Postoje 3 skrivena sloja od kojih svaki ima 32 neurona. Kod ovog modela vrsta stope učenja je constant čija je početna stopa učenja 1e-05. Rješavač je Adam. Sedmi model ostvaruje rezultate korištenjem tanh aktivacijske funkcije čija je stopa učenja 0.1. Broj skrivenih slojeva je 4 od kojih svaki sadrži po 32 neurona. Vrsta stope učenja koju koristi ovaj model je constant s početnom stopom učenja 1e-05, a rješavač koji se koristi je Adam. Na ovaj način postignuta je točnost od 0.956 sa standardnom devijacijom 0.178. Metrika AUC dostigla je vrijednost 0.96 sa standardnom devijacijom 0.16, a metrika F1 0.967 sa standardnom devijacijom 0.133.

Tablica 4.4. Ostvareni rezultati MLP algoritma sa SMOTE (macro)

Točnost	σ_{ACC}	**AUC**	σ_{AUC}	**F1**	σ_{F1}	**Hiperparametri**	
0.978	0.089	0.97	0.12	0.978	0.089	**Aktivacija**	Identity
						Regularizacija	0.01
						Skriveni slojevi	2
						Neurona po sloju	512
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	0.5
						Rješavač	LBFGS
0.956	0.178	0.99	0.04	0.967	0.133	**Aktivacija**	ReLu
						Regularizacija	0.1
						Skriveni slojevi	5
						Neurona po sloju	128
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.956	0.178	0.95	0.2	0.967	0.133	**Aktivacija**	Identity
						Regularizacija	0.001
						Skriveni slojevi	4
						Neurona po sloju	32
						Vrsta stope učenja	Adaptive
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.956	0.178	0.93	0.28	0.967	0.133	**Aktivacija**	Identity
						Regularizacija	0.0001
						Skriveni slojevi	5
						Neurona po sloju	2
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	0.5
						Rješavač	LBFGS
0.956	0.178	0.94	0.24	0.967	0.133	**Aktivacija**	Identity
						Regularizacija	0.0001
						Skriveni slojevi	3
						Neurona po sloju	32
						Vrsta stope učenja	Invscaling
						Početna stopa učenja	0.1
						Rješavač	LBFGS
0.956	0.178	0.98	0.08	0.967	0.133	**Aktivacija**	Tanh
						Regularizacija	0.01
						Skriveni slojevi	3
						Neurona po sloju	32
						Vrsta stope učenja	Constant
						Početna stopa učenja	1e-05
						Rješavač	Adam
0.956	0.178	0.96	0.16	0.967	0.133	**Aktivacija**	Tanh
						Regularizacija	0.1
						Skriveni slojevi	4
						Neurona po sloju	32
						Vrsta stope učenja	Constant
						Početna stopa učenja	1e-05
						Rješavač	Adam

Na slici 4.4 prikazan je graf koji predstavlja najbolji rezultat od svih modela prikazanih u tablici. Najbolji ostvareni rezultat MLP algoritma sa SMOTE (macro) je onaj s točnosti od 0.956 i standardnom devijacijom 0.178, tj. prvi opisani model iz ranije navedene tablice.



Slika 4.4. Najbolji ostvareni rezultati MLP algoritma sa SMOTE (macro)

Korištenjem standardne varijante s MLP tj.korištenjem metode NORMAL nisu dobiveni nužno najbolji rezultati. Idealno bi bilo kada bi njihov iznos bio preko 0.95 što u prikazanim tablicama 4.1 i 4.2 nije slučaj. Stoga je pokrenuta još jedna varijantna tj. metoda predobrade SMOTE, koja se koristi za balansiranje seta podataka. Tom metodom značajno su poboljšani rezultati kao što je vidljivo u tablicama 4.3 i 4.4.

5. Zaključak

Rak pluća jedan je od najčešćih malignih tumora od kojih je većina u srednjem i kasnom stadiju, što rezultira visokim mortalitetom. Umjetna inteligencija (kao relativno nova tehnologija) ima tendenciju postati od velike pomoći i značaja u dijagnostici uz standardne postupke kao što su CT i biopsija. Potencijal AI leži upravo u patološkoj analizi – identifikaciji regije tumora, predviđanju prognoze, karakterizaciji mikrookruženja tumora, otkrivanju metastaza, a na kraju i u predviđanju samog ishoda bolesti. Za potrebe analize uzet je Hongov i Youngov set podataka, a testiranje teorije temelji na prepoznavanju patoloških tipova tumora pluća. Navedeni testni podaci svrstavaju se u odgovarajuće klase, a sam cilj je otkriti optimalnu diskriminantnu ravninu čak i uz loše postavljene početne postavke. Kod metodologije najviše se ističe višeslojna umjetna neuronska mreža s perceptronom, kao potkategorija strojnog učenja. Same neuronske mreže temelje na kompleksnim biološkim živčanim sustavima, što znači obradu velike količine podataka koji se ne mogu lako generalizirati. Višeslojna perceptron umjetna neuronska mreža (MLP ANN) pokazala se kao dobar izbor u aplikacijama bitnima u dijagnostici, kao što su: prepoznavanje uzoraka, optimizacija, memorija s adresiranjem sadržaja, interakcija između čovjeka i računala i slično. Za prevladavanje poteškoća prekomjernog uzorkovanja, a samim time za poboljšanje učinka uvedena je tehnika sintetskog manjinskog prekomjernog uzorkovanja (SMOTE). SMOTE pruža nove informacije algoritmu učenja i rezultira poboljšanjem njegove predvidljivosti, a samim time i boljom izvedbom. Kod evaluacijskih metrika najčešće se ističu točnost, F1-rezultat i AUC. Nakon prikupljanja i obrade podataka slijedi njihova validacija u svrhu procjene kako će se predviđeni model ponašati u praksi. Uz navedeno, postignuti su rezultati točnosti analize od 94% do 98%, prema čemu je logičan zaključak da spomenuta dijagnostika uz AI metode daje iznimno zadovoljavajuće rezultate i nadu čovječanstvu za smanjenje oboljenja i mortaliteta od karcinoma pluća.

Bibliografija

[1] "Artificial intelligence in clinical applications for lung cancer: diagnosis, treatment and prognosis", s Interneta, https://www.degruyter.com/document/doi/10.1515/cclm-2022-0291/html, 06.08.2022.

[2] Hong, Z.Q. and Yang, J.Y. "Optimal Discriminant Plane for a Small Number of Samples and Design Method of Classifier on the Plane", Pattern Recognition, Vol. 24, No. 4, pp. 317-324, 1991.

[3] "Artificial Intelligence Tools for Refining Lung Cancer Screening", s Interneta, https://www.mdpi.com/2077-0383/9/12/3860, 06.08.2022.

[4] "Role of artificial intelligence in the care of patients with nonsmall cell lung cancer", s Interneta, https://onlinelibrary.wiley.com/doi/abs/10.1111/eci.12901, 06.08.2022.

[5] "How to Train a Multilayer Perceptron Neural Network",s Interneta,https://www.allaboutcircuits.com/technical-articles/how-to-train-a-multilayer-perceptron-neural-network/, 15.10.2022.

[6] Hassan Ramchoun, Mohammed Amine Janati Idrissi, Youssef Ghanou, Mohamed Ettaouil. "Multilayer Perceptron: Architecture Optimization and Training", Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, University Sidi Mohammed Ben Abdellah, Fez, Morocco

[7] Rosenblatt, “The Perceptron: A Theory of Statistical Separability in Cognitive Systems”, Cornell Aeronautical Laboratory, Report No. VG1196-G-1, January, 1958

[8] "Hyperparameter tuning for TensorFlow using Katib and Kubeflow", s Interneta, https://tfworldkatib.github.io/tutorial/katib/grid.html, 15.10.2022.

[9] "Hyperparameter optimization", s Interneta, https://en.m.wikipedia.org/wiki/File:Hyperparameter_Optimization_using_Grid_Search.svg, 15.10.2022.

[10] "Cross-validation",Daniel Berrar,Data Science Laboratory, Tokyo Institute of Technology 2-12-1-S3-70 Ookayama, Meguro-ku, Tokyo 152-8550, Japan, s Interneta,https://www.researchgate.net/profile/Daniel-Berrar/publication/324701535_Cross-Validation/links/5cb4209c92851c8d22ec4349/Cross-Validation.pdf, 15.10.2022.

[11] "The 5 Classification Evaluation metrics every Data Scientist must know", s Interneta, https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226, 16.10.2022.

[12] "Accuracy, Recall and Precision", s Interneta, https://medium.com/@erika.dauria/accuracy-recall-precision-80a5b6cbd28d, 16.10.2022.

[13] Hercules Dalianis. "Clinical Text Mining Secondary Use of Electronic Patient Records", DSV-Stockholm University Kista, Sweden, 2018.

[14]] J.A. Hanley and B.J. McNeil, “The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve,” Radiology, vol. 143, pp. 29-36, 1982.

[15] J. Huang and C.X. Ling “Using AUC and Accuray in Evaluating Learing Algorithms”, IEEE Transactions on Knowledge and Data Engineering vol. 17, no. 3, pp. 299-310, 2005.

[16] "A scored AUC Metric for Classifier Evaluation and Selection", s Interneta, http://dmip.webs.upv.es/ROCML2005/papers/wuCRC.pdf, 17.10.2022.

[17] Hossin, M.;Sulaiman, M.N.,"A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS",International Journal of Data Mining & Knowledge Management Process (IJDKP), Vol.5, No.2, 2015.

[18] "What Is AUC?", s Interneta, https://arize.com/blog/what-is-auc/, 17.10.2022.

[19] Baressi Šegota,S; i dr. “Improvement of Marine Steam Turbine Conventional Exergy Analysis by Neural Network Application”, Journal of Marine Science and Engineering, 37, Faculty of Engineering, University of Rijeka, Vukovarska 58, 2020.

[20] Grandin,M ; i dr. “METRICS FOR MULTI-CLASS CLASSIFICATION: AN OVERVIEW”, A WHITE PAPER,Department of Computer Science,University of Bologna, 2020.

[21] Fernandez,A ; i dr. “SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary”, Journal of Artifcial Intelligence Research 61 (2018) 863-905,2018.

[22] V. Chawla,N ; i dr. SMOTE: Synthetic Minority Over-sampling Technique”, Journal of Artificial Intelligence Research 16 (2002) 321–357,2002.

[23] "An Easy Guide to Neuron Anatomy with Diagrams", s Interneta, https://www.healthline.com/health/neurons, 4.11.2022.

[24] "Rosenblatt’s perceptron, the first modern neural network", s Interneta, https://towardsdatascience.com/rosenblatts-perceptron-the-very-first-neural-network-37a3ec09038a, 4.11.2022.

Popis slika

Popis tablica

Sažetak i ključne riječi

U radu je izrađen pregled literature u području dijagnostike bolesti, posebice karcinoma i plućnih oboljenja primjenom metoda umjetne inteligencije, u svrhu što ranije dijagnoze. Također, dan je opis Hongovog i Youngovog seta podataka koji se koristio za potrebe analize. Opisana je višeslojna perceptron neuronska mreža koja se pokazala kao dobar izbor u aplikacijama bitnim u dijagnostici. Nadalje, opisana je tehnika sintetskog manjinskog prekomjernog uzorkovanja SMOTE koji rezultira boljom izvedbom rezultata. Što se tiče evaluacijskih metrika, dani su opisi točnosti, F1-rezultata te AUC. Nakon prikupa i obrade podataka dana je njihova validacija u svrhu procjene kako će se model ponašati u praksi.

Ključne riječi: karcinom pluća, dijagnostika, umjetna inteligencija, višeslojna perceptron neuronska mreža, SMOTE, točnost, F1-rezultat, AUC

Summary and key words

The paper contains a review of the literature in the field of disease diagnostics, especially cancer and lung diseases, using artificial intelligence methods, for the purpose of early diagnosis. Also, a description of the Hong and Young data set used for analysis is provided. A multilayer perceptron neural network is described, which proved to be a good choice in applications important in diagnostics. Furthermore, a SMOTE synthetic minority oversampling technique is described which results in better performance of the results. Regarding evaluation metrics, descriptions of accuracy, F1 score and AUC are given. After data collection and processing, their validation was given in order to evaluate how the model will behave in practice.

Keywords: lung cancer, diagnosis, artificial intelligence, multilayer perceptron neural network, SMOTE, accuracy, F1-score, AUC

A Python kod za matricu konfuzije

```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

actual = numpy.random.binomial(1,.9,size = 1000)
predicted = numpy.random.binomial(1,.9,size = 1000)

confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
```

B Python kod za modeliranje

```
import numpy as np
import os
import warnings
from sklearn.model_selection import KFold, GridSearchCV
from sklearn.neural_network import MLPClassifier
from imblearn.over_sampling import SMOTE
import pickle
import uuid

#Adjust to True/False to use/not use SMOTE
USE_SMOTE = True

#TODO
#CHANGE DATA
DATA = np.loadtxt("./lung-cancer.data", skiprows=0, delimiter=',')
X = DATA[:, 1:56]
Y = DATA[:, 0]


if USE_SMOTE:
sm = SMOTE()
X_res, Y_res = sm.fit_resample(X,Y)
else:
X_res = X
Y_res = Y
print(Y)
descriptor="mlp"


params_dict = [{'hidden_layer_sizes': [(1),(1,1),(1,1,1),(1,1,1,1),(1,1,1,1,1),
(2),(2,2),(2,2,2),(2,2,2,2),(2,2,2,2,2),
(4),(4,4),(4,4,4),(4,4,4,4),(4,4,4,4,4),
(8),(8,8),(8,8,8),(8,8,8,8),(8,8,8,8,8),
(16),(16,16),(16,16,16),(16,16,16,16),(16,16,16,16,16),
(32),(32,32),(32,32,32),(32,32,32,32),(32,32,32,32,32),
(64),(64,64),(64,64,64),(64,64,64,64),(64,64,64,64,64),
(128), (128,128), (128,128,128), (128,128,128,128), (128,128,128,128,128),
(256), (256,256), (256,256,256), (256,256,256,256), (256,256,256,256,256),
(512), (512,512), (512,512,512),(512,512,512,512),(512,512,512,512,512)],
'activation': ['relu','identity','logistic','tanh'],
'solver': ['adam', 'lbfgs'],
```

```
'learning_rate':['constant','adaptive','invscaling'],
'learning_rate_init': [0.1,0.01,0.5, 0.00001],
'alpha': [0.01,0.1,0.001, 0.0001],
'max_iter': [10000]}]


scores = ['accuracy', 'roc_auc', 'f1']
model = GridSearchCV(MLPClassifier(), params_dict, cv=5, n_jobs=-1, scoring=scores,
 verbose=10, refit=False)
model.fit(X_res,Y_res)

print(50*"*"+"\n"+50*"*"+"\n"+50*"*"+"\n"+"Fitting␣DONE\n"+50*"*"
+"\n"+50*"*"+"\n"+50*"*"+"\n")

means1 = model.cv_results_['mean_test_accuracy_micro']
stds1 = model.cv_results_['std_test_accuracy_micro']
means2 = model.cv_results_['mean_test_roc_auc_micro']
stds2 = model.cv_results_['std_test_roc_auc_micro']
means3 = model.cv_results_['mean_test_f1_micro']
stds3 = model.cv_results_['std_test_f1_micro']

uuid_=uuid.uuid4()
file = open(descriptor+"-"+str(uuid_)+"-results.txt", 'w')
file.write("ACC;␣STD;␣AUC;␣STD;␣F1;␣STD;␣PARAMS\n")
for mean1, std1,mean2, std2, mean3, std3, params in zip(means1, stds1,
means2, stds2, means3, stds3, model.cv_results_['params']):
file.write("%0.20f;␣%0.020f;␣%0.20f;␣%0.020f;␣%0.20f;␣%0.020f;
␣%r" % (mean1, std1*2, mean2, std2*2, mean3, std3*2, params) +"\n")
file.close()
model_name = descriptor+"-"+str(uuid_)+".pickle"

print(50*"*"+"\n"+50*"*"+"\n"+50*"*"+"\n"+"DONE\n"+50*"*"
+"\n"+50*"*"+"\n"+50*"*"+"\n")
```

C Python kod za plotanje grafova

```
from matplotlib import pyplot as plt

import numpy as np
#ACC$  #AUC  #F1
VALUES =     [0.978, 0.97, 0.978]
STDS =       [0.089, 0.12, 0.089]
title = r"Ostvareni rezultati MLP algoritma sa/bez SMOTE (micro/macro)"

plt.figure(dpi=100)
plt.rc('axes', axisbelow=True)
plt.grid()
plt.bar([0,1,2], VALUES, yerr=STDS)

plt.title(title)
plt.xticks([0,1,2],["ACC", "AUC", "F1"])
plt.ylabel("Rezultat")
plt.xlabel("Metrike")
plt.show()
```

D Python kod za sigmoidalnu aktivacijsku funkciju

```
import numpy as np
import matplotlib.pyplot as plt

# Sigmoid Activation Function
def sigmoid(x):
return 1/(1+np.exp(-x))


# Generating data to plot
x_data = np.linspace(-10,10,100)
y_data = sigmoid(x_data)


# Plotting
plt.plot(x_data, y_data)
plt.title('Sigmoidalna funckija')
plt.legend('Sigmoid')
plt.xlabel('x')
plt.ylabel('Sigmoid(x)')
plt.grid()
plt.show()
```

E Python kod za rektificiranu linearnu (ReLu) aktivacijsku funkciju

```
import numpy as np
import matplotlib.pyplot as plt

# Rectified Linear Unit (ReLU)
def ReLU(x):
data = [max(0,value) for value in x]
return np.array(data, dtype=float)


# Generating data for Graph
x_data = np.linspace(-10,10,100)
y_data = ReLU(x_data)

# Graph
plt.plot(x_data, y_data)
plt.title('ReLU Aktivacijska funckija')
plt.legend('ReLU funkcija: max(0,x)')
plt.xlabel('x')
plt.ylabel('ReLu(x)')
plt.grid()
plt.show()
```

F Python kod za tangens hiperbolnu (Tanh) aktivacijsku funkciju

```
import math
import numpy as np
import matplotlib.pyplot as plt

in_array = np.linspace(-10, np.pi**2, 30)

out_array = []

for i in range(len(in_array)):
out_array.append(math.tanh(in_array[i]))
i += 1

print("Input_Array : \n", in_array)
print("\nOutput_Array : \n", out_array)


plt.plot(in_array, out_array)
plt.title("Tanh aktivacijska funckija")
plt.xlabel("x")
plt.ylabel("Tanh(x)")
plt.grid()
plt.legend('Tanh')
plt.show()
```

G Python kod za linearnu (Identity) aktivacijsku funkciju

```
import numpy as np
import matplotlib.pyplot as plt
import numpy as np

def linear(x):
return x

x = np.linspace(-10,10)
plt.plot(x, linear(x))
plt.axis('tight')
plt.xlabel('x')
plt.ylabel('Linear(x)')
plt.title('Linearna (Identity) aktivacijska funckija')
plt.grid()
plt.legend('L')
plt.show()
```