Predviđanje biološke aktivnosti malih molekula modelom strojnog učenja

Tomasić, Jakov

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Prijediplomski studij računarstva

Završni rad

Predviđanje biološke aktivnosti malih molekula modelom strojnog učenja

Rijeka, rujan 2023.

Jakov Tomasić
0069089376

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Prijediplomski studij računarstva

Završni rad

Predviđanje biološke aktivnosti malih molekula modelom strojnog učenja

Mentor: doc. dr. sc. Goran Mauša

Rijeka, rujan 2023.

Jakov Tomasić
0069089376

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 13. ožujka 2023.

Zavod: **Zavod za računarstvo**
Predmet: **Programsko inženjerstvo**
Grana: **2.09.04 umjetna inteligencija**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Jakov Tomasić (0069089376)**
Studij: Sveučilišni prijediplomski studij računarstva

Zadatak: **Predviđanje biološke aktivnosti malih molekula modelom strojnog učenja // Machine learning-based biological activity prediction for small molecules**

Opis zadatka:

Analizirati podatke biološke aktivnosti malih molekula sadržanih u bazi otvorenog pristupa CO-ADD. Provesti predobradu podataka s ciljem njihove pripreme za primjenu strojnog učenja. Izraditi model predviđanja biološke aktivnosti tehnikom klasifikacije s višestrukim oznakama te ju usporediti s tehnikom binarne klasifikacije. Provesti analizu uzroka netočnih predviđanja i predložiti tehnike koje bi mogle poboljšati uspješnost predviđanja.

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

Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor:

Doc. Goran Mauša, dipl. ing.

Predsjednik povjerenstva za završni ispit:

Prof. dr. sc. Miroslav Joler

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2023.

_____
Jakov Tomasić

Zahvala

Zahvaljujem mentoru doc. dr. sc. Goranu Mauši i asistentu Eriku Otoviću, mag. ing. comp. na podršci i svom prenesenom znanju tijekom pisanja ovoga rada. Zahvaljujem i svim prethodnim mentorima, kolegama i obitelji na kvalitetnim iskustvima, prenesenom znanju i podršci.

Sadržaj

Popis slika

Popis tablica

Poglavlje 1

Uvod

Velika problematika u medicini, a pogotovo kod razvoja antibiotika i ostalih farmaceutskih proizvoda, svodi se na istraživanje razine biološke aktivnosti pojedinih molekula na različite mikroorganizme. Cilj je postići što veću reaktivnost, odnosno negativno utjecati na štetne mikroorganizme, a pritom izbjeći negativne učinke na korisne mikroorganizme. Takva su istraživanja, tj. testiranja, veoma skupa i dugotrajna. Zbog toga se što veći dio problematike pokušava prebaciti u domenu računarstva i umjetne inteligencije s ciljem bržeg i jeftinijeg predviđanja reakcija molekula i mikroorganizama.

Ovaj završni rad izrađen je u okviru projekta „Dizajn katalitički aktivnih peptida i peptidnih nanostruktura" s oznakom UIP-2019-04-7999 u kojem se istražuje antimikrobna aktivnost malih molekula koje karakterizira veća stabilnost i šira terapeutska primjena u odnosu na peptide. Cilj rada je za danu malu molekulu predvidjeti s kojima od podržanih mikroorganizama ona reagira. Ovaj završni rad će također istražiti klasifikaciju s višestrukim oznakama s ciljem bržeg treniranja i predviđanja kao i poboljšanja konačnih rezultata u usporedbi s jednostavnijom binarnom klasifikacijom. Predviđanje reakcije molekula i mikroorganizama je moderan problem bez općeprihvaćenih i univerzalnih rješenja. Zbog toga će se jedan dio rada usredotočiti na pokušaje optimizacije modela predviđanja, obrade podataka i primjene različitih tehnika iz domene strojnog učenja i znanosti o podacima, a sve s ciljem povećanja kvalitete rješenja.

Poglavlje 2

Analiza podataka

Podaci uzeti za treniranje, testiranje i validaciju su iz *dose response data* Comma-separated values (CSV) datoteke iz Co-add baze podataka [1]. Ta CSV tablica sadrži 42210 zapisa što je relativno puno za ovu domenu. To je baza podataka o malim molekulama s kojima je napravljena sustavna analiza aktivnosti raznih molekula za nekoliko mikroorganizama.

Svaki redak u tablici predstavlja mjerenja reakcije jedne male molekule nad jednim organizmom. Molekula je predstavljena pomoću *string* zapisa u formatu simplified molecular-input line-entry system (SMILES).

Uspoređujući prikaz rezultata molekula na stranici CO-ADD baze podataka [1] i spomenute CSV datoteke vidljivo je da stupac `DRVAL_MEDIAN` binarno označuje je li u testu došlo do reakcije molekule i organizma. Stupac `DRVAL_MEDIAN` označava količinu molekula potreban za reakciju. Svaka molekula imat će reakciju sa svakim organizmom ako osiguramo dovoljnu količinu. Cilj je predvidjeti reakciju praktično malih količina molekula kako bi se mogli iz toga napraviti razni proizvodi. Ako je X proizvoljan realan broj, mogući zapisi u `DRVAL_MEDIAN` stupcu i njihova značenja su sljedeća:

- „X” - dolazi do reakcije kada je koncentracija molekula veća ili jednaka X
- „<X” - dolazi do reakcije čak i u slučajevima kada je koncentracija molekula manja od X, a pogotovo kada je ona veća ili jednaka X

- „>X" - ne dolazi do reakcije kada je koncentracija molekula manja ili jednaka X, a X predstavlja najveću prihvatljivu koncentraciju malih molekula

Za potrebe binarne klasifikacije s mogućim izlazima postojanja i nepostojanja biološke reakcije zapisi tipa „X" i „<X" označuju postojanje reakcije dok zapis tipa „>X" označuje ne postojanje reakcije malih molekula. Za regresijske pristupe može se koristiti `DMAX_AVE` stupac.

2.1 Kategorije aktivnosti malih molekula

Podaci sadrže i molekule sa samo nekoliko zapisa odnosno provedenih mjerenja reakcije. Takve podatke sam morao pročistiti na samo najzastupljenije kategorije aktivnosti:

- Staphylococcus aureus (SA)
- Escherichia coli (EC)
- Klebsiella pneumoniae (KP)
- Pseudomonas aeruginosa soj ATCC 27853 (PA)
- Pseudomonas aeruginosa soj PAO397, PAO1 (PA5Δ) - zanemareno
- Acinetobacter baumannii (AB)
- Candida albicans (CA)
- Cryptococcus neoformans (CN)
- Embrionalne stanice bubrega čovjeka HEK 293 (HEK)
- Ljudska crvena krvna zrnca (engl. *Human red blood cell*, hRBC) - zanemareno

Iako neke od ovih kategorija aktivnosti nisu organizmi (npr. HEK i hRBC su stanice), u nastavku rada ću se referencirati na sve kategorije aktivnosti malih molekula kao organizmi odnosno mikroorganizmi zbog jednostavnosti. Cilj ovoga rada je naći reakcije molekula s mikroorganizmima i ostalim štetnim tijelima, a paralelno predvidjeti hoće li te male molekule reagirati s korisnim mikroorganizmima i stanicama, poput ljudskih crvenih krvnih zrnaca. Zato ima smisla razmišljati o izlaznim kategorijama

aktivnosti kao mikroorganizmima.

Raspodjela oznaka vidljiva je u tablici 2.1 u kojoj su značenja vrijednost prvog stupca sljedeća:

- 0 - zapis da nema reakcije molekule s određenim mikroorganizmom (negativan zapis)
- 1 - zapis da molekula reagira s određenim mikroorganizmom (pozitivan zapis)
- 0/1 - ukupni broj postojećih zapisa za mikroorganizam (s ili bez reakcije)
- ? - broj molekula koji nemaju zapis o reakciji s određenim mikroorganizmom

Zapažen je iznimno malen broj zapisa za hRBC i PA5Δ organizme. Također je lako očitati da podaci nisu ujednačeni odnosno da pozitivnih zapisa ima više nego negativnih.

Tablica 2.1 Raspodjela poznatih i nepoznatih oznaka po mikroorganizmu

	SA	EC	KP	PA	PA5Δ	AB	CA	CN	HEK	hRBC
0	3362	4385	4502	4709	826	4492	3573	3251	3260	2501
1	1200	178	56	44	99	266	988	1310	1321	100
0/1	4562	4563	4558	4753	925	4758	4561	4561	4581	2601
?	223	222	227	32	3860	27	224	224	204	2184

2.2 Priprema podataka za strojno učenje

Dani podaci nisu obrađeni za strojno učenje te ih treba kvalitetno pročistiti i pripremiti za isto. Ovaj korak u procesu strojnog učenja je izuzetno važan jer je teško dobiti dobre rezultate od loših podataka. Svi daljnji rezultati uvelike će ovisiti o kvaliteti obrade podataka.

2.2.1 Filtriranje podataka

Dani zapis gdje svaki red predstavlja jedan rezultat nije optimalan te je problematičan za strojno učenje. Zbog toga sam ga preradio u novu jednostavniju CSV datoteku. Svaki redak u novom zapisu pokazuje rezultate za jednu molekulu odnosno sadrži SMILES zapis te molekule te reagira li ona s mikroorganizmom. Točnije, za svaki mikroorganizam je zabilježeno reagira li s molekulom (oznaka 1), ne reagira (oznaka 0) ili je podatak nepoznat (oznaka ?). Svaka zapis nove CSV datoteke sadrži podatke iz deset zapisa stare. Tako je broj novih zapisa znatno manji, samo 4786.

Navedeni nepoznati podaci predstavljaju velik problem za klasifikaciju s višestrukim oznakama (engl. *Multi-task learning (MTL)*) pošto je treniranje nad podacima, gdje je očekivana vrijednost nepoznata, mnogo složenije i izvan dosega ovoga rada. Radi toga je bilo potrebno izbaciti sve zapise koji imaju nepoznat zapis odnosno sve molekule čija reakcija s nekim od odabranih organizama nije poznata. Nakon te filtracije broj zapisa je samo 723.

Za povećanje broja validnih zapisa mogu se zanemariti neki organizmi koji imaju velik broj nedostajućih vrijednosti. Analizom podataka i testiranjem svih opcija izbacivanja utvrđeno je da je najbolje izbaciti dva organizma - Human red blood cell (hRBC) i Pseudomonas aeruginosa soj PAO397, PAO1 (PA5Δ). Nakon toga krajnji broj zapisa jednak je 4558. Ovim filtriranjem podataka je broj podržanih mikroorganizama pao s deset na osam. Da je ova kombinacija optimalna potvrđuje činjenica da se izbacivanjem još jednog zapisa dobiveno još samo ukupno dva zapisa. To je potvrđeno računanjem broja validnih zapisa za svaki od $2^10 = 1024$ podskupova oznaka.

Treba napomenuti da izbacivanje molekula s nepoznatim zapisom nije potrebno za klasični pristup binarne klasifikacije. Međutim, oni su izbačeni kako bi se isti podaci mogli koristiti za oba tipa strojnog učenja, radi pravednije usporedbe s identičnim početnim uvjetima.

2.2.2 Formatiranje ulaznih podataka

Model strojnog učenja za svaku molekulu treba predvidjeti s kojim od osam organizama ona reagira. To znači da je ulaz u model molekula, a izlaz binarna vrijednost za svaki organizam, tj. ima li (1) ili nema (0) reakcije. Molekula je predstavljena SMILES zapisom koji je u suštini znakovni niz. Zapis znakovnog niza je daleko od optimalnoga što je potvrđeno rezultatima provedenog testiranja kod kojeg je preciznost predviđanja bila niža od 1% , a druge metrike su dale slično loše rezultate. To dovodi do zaključka kako je potrebna obrada podataka u drugi zapis.

To je riješeno pomoću knjižnica Python *rdkit*[2] i *mordred*[3] pomoću kojih se SMILES zapis pretvorio u molekularni zapis te iz njega generirao popis od oko 1600 značajki za svaku molekulu iz skupa podataka. Nakon pretvorbe uklonjene su sve značajke koji imaju nepoznatu vrijednost za neku molekulu ili koji imaju jednaku vrijednost za sve molekule. Nakon tog čišćenja, broj značajki svake molekule iznosio je 801. Zahvaljujući ovom rješenju, model strojnog učenja na ulaz prima 801 realnih brojeva (*float*) iz kojih može lakše zaključiti svojstva molekula te ih upotrijebiti za predviđanje reakcije sa zadanim organizmima.

2.3 Predobrada podataka

2.3.1 Skaliranje i centriranje ulaznih podataka

Spomenute značajke molekula su realni brojevi različitih vrijednosti Literatura je pokazala[4] kako modeli dubokih neuronskih mreža daju bolje rezultate nakon skaliranja i centriranja ulaznih podataka. Za to sam koristio *StandardScaler*[5] iz knjižnice *sklearn.preprocessing* koji centrira podatke tako da konačni medijan bude jednak 0 te skalira ulazne podatke tako da je njihova standardna devijacija jednaka 1.

Testiranje modela prije i nakon uvođenja skaliranja pokazalo je da skaliranja u ovome slučaju ne mijenja rezultate modela. Svejedno, podaci će ostati skalirani zato što je takva preporuka iz navedene literature.

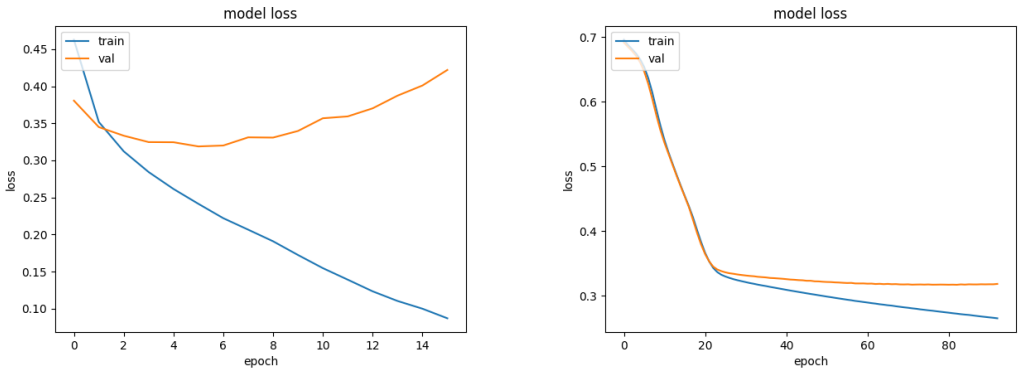
2.3.2 Redukcija dimenzionalnosti

Model strojnog učenja je opterećen velikim ulazom od čak 801 podatka. Zbog toga je učenje otežano te model ne može ustanoviti pravilnosti između značajki promatranog skupa podataka. On zato pokušava učiti podatke *napamet* umjesto ispravnog treniranja, odnosno učenja poveznica i zaključivanja rezultata. Drugim riječima, zbog prevelike dimenzionalnosti ulaznih podataka dolazi do fenomena koji nazivamo *pretreniranje* (engl. *overfitting*).

Potencijalno rješenje ovog problema je PCA koji je popularna tehnika sažimanja broja podataka. Ona analizira sve ulazne podatke s ciljem sažimanja istih u nove varijable bez gubitka neke količine podataka[6, 7]. Ta količina podataka se u programskoj knjižnici Python *sklearn.decomposition*[8] može lako definirati prije pokretanja ove tehnike. Točnije, ta knjižnica prima minimalnu željenu varijancu koju rezultirajući skup podataka mora opisivati. Veća varijanca inače znači da su podaci više raširenі[9]. U ovom slučaju, veća varijanca znači manji broj izgubljenih podataka u odnosu na početni skup podataka. Zato treba naći ravnotežu između što veće varijance i što manjeg skupa izlaznih podataka. Idealan slučaj bi bio kada bi svi početni podaci mogli bili predstavljeni samo jednim zapisom bez gubitka podataka. Nakon korištenja PCA tehnike s varijancom od 95% broj podataka na ulazu smanjio se na 81.

Grafovi 2.1 pokazuju kako korištenje tehnike PCA (graf 2.1b) rješava problem *pretreniranja* koji je bez PCA prisutan (graf 2.1a).

Prilikom treninga modela neuronskih mreža *pretreniranje* se očitava naglim rastom vrijednosti funkcije gubitka (engl. *loss*) validacijskog skupa podataka za vrijeme pada funkcije gubitka skupa podataka za treniranje (graf 2.1a). U našem slučaju automatskog zaustavljanja rastom funkcije gubitka, *pretreniranje* također rezultira manjim brojem epoha (engl. *epoch*) treniranja. Kao što se može vidjeti na grafu 2.1, broj epoha bez primjerne PCA tehnike je 15 dok treniranje modela nakon primjene PCA tehnike sadrži oko 90 epoha. Poželjan je veći broj *epoha* unutar kojih se obje loss funkcije smanjuju. Također, manji broj ulaza zahtijeva i time manji model što smanjuje potrebno vrijeme treniranja i predviđanja pomoću modela. Literatura preporučuje skaliranje i centriranje ulaznih podataka (kao i ostali *pre-processing*) *prije*



(a) Bez PCA (b) Sa PCA

Slika 2.1 Usporedba loss funkcije uz korištenje PCA tehnike

izračuna PCA[10] što je poštovano u izradi ovoga rada.

2.4 Binarna izlazna vrijednost

Zbog kasnijih uporaba metrika bitno je odrediti koji će podatak biti predstavljen s pozitivnom oznakom (1), a koji s negativnom oznakom (0). Teoretsko razlikovanje značenja tih oznaka je bitno jer metrikama u izračunu nije jednako radi li se o npr. true positive (TP) ili true negative (TN), a ono ponajviše ovisi o primjeni programskog rješenja. U ovome slučaju, pozitivna oznaka (1) označuje reakciju, a negativna oznaka (0) nedostatak iste. Mi tražimo oznake reaktivnosti u svrhu detekcije koje na parazitske organizme molekula utječe odnosno koje molekule negativno utječu na ljudski organizam (Embrionalne stanice bubrega čovjeka HEK 293 (HEK)) kako bi njih mogli izbjeći. To bi nam omogućilo olakšan razvoj lijekova protiv željenih organizama, ali bez negativnih posljedica na tijelo. Dakle, željeni rezultat je postojanje reakcije te ćemo njega označavati kao pozitivan (1).

Treniranje te testiranje modela s obrnutom reprezentacijom pozitivnih i negativnih rezultata daje nerealno velike rezultate. To je potvrda dobrog odabira prikaza pozitivnih i negativnih rezultata, ali i potvrda da krivi odabir može uvelike promijeniti sliku rezultata modela.

2.5 Kontradiktorni podaci

Prilikom pripreme podataka za strojno učenje, različite značajke molekula se generiraju iz SMILES zapisa molekula iz početnog skupa podataka. Detaljnijom analizom odnosa ulaznih podataka te njihovih razlikovanja otkriveno je da postoje pet parova podataka s identičnim ulazom (značajkama), a različitim očekivanim izlazom (mikrobiološkim reakcijama).

Ova saznanja uvelike dovode u upit odabrano rješenje za dani problem te mogućnost predviđanja mikrobiološke reakcije s mikroorganizmima samo iz SMILES zapisa molekule. Točnije, postojanje istih ulaza s jednakim izlazima izravno govori da korištenje programske knjižnice Python mordred[3] nije primjereno odnosno optimalno.

Spomenuta knjižnica za izračun značajki iz SMILES zapisa malih molekula izračunava jednake skupove značajki za različite molekule s potpuno različitim svoj-

stvima reakcija s promatranim mikroorganizmima (u slučaju sličnih SMILES zapisa). Jedan primjer dvaju različitih molekula s jednakim generiranim značajkama su *Doxorubicin hydrochloride* (a) i *Epirubicin hydrochloride* (b). Najvjerojatniji uzrok generiranja identičnih značajki je izrazito velika sličnost njihovog SMILES zapisa:

a) COc1cccc2C(=O)c3c(O)c4C[C@](O)(C[C@H](O[C@H]5C[C@H](N)[C@H](O)[C@H](C)O5)c4c(O)c3C(=O)c12)C(=O)CO.Cl

b) COc1cccc2C(=O)c3c(O)c4C[C@](O)(C[C@H](O[C@H]5C[C@H](N)[C@ @H](O)[C@H](C)O5)c4c(O)c3C(=O)c12)C(=O)CO.Cl

gdje sam plavom pozadinom istaknuo jedini znak različit u dva SMILES zapisa.

U korištenom skupu podataka otkriveno je čak 13 parova molekula s gotovo istim značajkama, a različitim reakcijama s promatranim mikroorganizmima. Ovaj broj dobiven je primjenom algoritma traženja najbližih parova po euklidskoj udaljenosti s graničnom vrijednosti od samo 0.05. Algoritam će kasnije biti detaljnije objašnjen u poglavlju 4.5.

Nakon uklanjanja ovih 13 konfliktnih podataka iz cijelog skupa podataka rezultati treniranja modela se nisu značajno promijenili.Iako taj mali broj konfliktnih podataka nema veliki učinak na cjelokupan rezultat treba uzeti u obzir da sama tehnika predobrade podataka i pristup rješenja ovog problema zahtijevaju poboljšanja. Također postoji mogućnost kako, zbog korištenja podataka izvedenih samo iz zapisa SMILES, model strojnog učenja nema dovoljno informacija za ostvarenje zavidnih rezultata.

Poglavlje 3

Izrada i vrednovanje modela

Treniranje i vrednovanje rezultata modela vrše se pet puta za svaki model. To je implementirano pomoću pet iteracija *K-Fold*[11] funkcije koja za svaku iteraciju podijeli sve podatke na skup za treniranje i testiranje u omjeru 4:1. Podaci u većem skupu za treniranje nakon toga se dijele na podatke za treniranje i validaciju u omjeru 7:3. Podaci za testiranje su različiti za svaku iteraciju što znači da će kroz svih pet iteracija svaki podatak biti točno jednom unutar skupa podataka za testiranje. Od svih pet rezultata testiranja kao završnu metriku uzimamo srednje vrijednosti (engl. *mean*) svih metrika te njihove standardne devijacije. Programski alat *K-Fold* tako nam omogućava vrednovanje modela nad cijelim skupom podataka, ali zadržava željeno svojstvo da se model nikada ne smije testirati nad istim podacima nad kojima je treniran ili validiran. Vrednovanje podataka koristi više metrika navedenih u nastavku.

3.1 Korištene metrike

Odabir odgovarajućih metrika ključan je korak pri izradi rješenja. Neodgovarajuće metrike mogle bi davati odličnu sliku za model koji je u stvarnosti jako loš i obratno. Za pouzdanu usporedbu različitih parametara modela i pristupa optimizacija rezultata korištene su mnoge osnove metrike opisane u nastavku.

Mi ćemo računati metrike za svaki mikroorganizam zasebno. Mogući izlazi mo-

dela su 0 (nema reakcije) ili 1 (reakcija postoji). U našem slučaju vrste svih navedenih metrika će biti klasifikacijske metrike koje primaju četiri vrijednost:

- TP (*engl. True Positive*) - broj točno predviđenih slučajeva 1 (reakcija postoji)
- TN (*engl. True Negative*) - broj točno predviđenih slučajeva 0 (nema reakcije)
- FP (*engl. False Positive*) - broj netočno predviđenih slučajeva 1 (reakcija postoji) dok je očekivana vrijednost bila 0 (nema reakcije)
- FN (*engl. False Negative*) - broj netočno predviđenih slučajeva 0 (nema reakcije) dok je očekivana vrijednost bila 1 (reakcija postoji)

3.1.1 Točnost

Točnost (*engl. accuracy*) generalno predstavlja odstupanje rezultata mjerenja od prave vrijednosti mjerene veličine[12]. Točnost binarnog klasificiranja se izražava kao omjer točnih predviđanja i ukupnog broja predviđanja:[13]

$$\text{Točnost} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Točnost će se kasnije pokazati kao loša metrika za naš slučaj zbog neujednačene distribucije klasa unutar skupa podataka. Visoki rezultati točnosti će pokazati da odabir krive metrike može davati netočnu sliku o kvaliteti rješenja. Vrijednost točnosti može biti u rasponu između 0 i 1.0 (uključujući), gdje je 0 najlošiji, a 1.0 najbolji rezultat.

3.1.2 Preciznost

Preciznost (*engl. precision*) odgovara na sljedeće pitanje: "Koji omjer pozitivnih predviđanja (oznaka 1) su bili točno predviđeni?"[14]. Preciznost se izražava kao omjer točnih pozitivnih previđanja i ukupnog broja izlaza u kojima je model predvidio pozitivan ishod:

$$\text{Preciznost} = \frac{TP}{TP + FP} \tag{3.2}$$

Vrijednost preciznosti može biti u rasponu između 0 i 1.0 (uključujući), gdje je 0 najlošiji, a 1.0 najbolji rezultat.

3.1.3 Opoziv

Opoziv (*engl. recall*) odgovara na sljedeće pitanje: "Koji omjer stvarno pozitivnih rezultata (oznaka 1) su bili ispravno predviđeni?"[14]. Preciznost se izražava kao omjer točnih pozitivnih previđanja i ukupnog broja stvarno pozitivnih rezultata:

$$\text{Opoziv} = \frac{TP}{TP + FN} \tag{3.3}$$

Vrijednost opoziva može biti u rasponu između 0 i 1.0 (uključujući), gdje je 0 najlošiji, a 1.0 najbolji rezultat.

3.1.4 F1 parametar

Preciznost i opoziv bolje su metrike za probleme s neujednačenom distribucijom klasa. Potrebno je istovremeno pratiti preciznost i opoziv. No, problem je što su te dvije metrike često u tenzijama[14]. Rast jedne metrike često će pratiti pad druge i obrnuto. Iz tog razloga razvijene su razne metrike koje pokušavaju kombinirati preciznost i opoziv u jedan, lako usporedivi, broj.

F1 parametar jedna je od tih metrika. F1 je harmonijska srednja vrijednost preciznosti i opoziva koja jednim brojem simetrično i ravnopravno predstavlja obje metrike[15]. Očekivano, F1 parametar računa se iz preciznosti i opoziva:[16]

$$F_1 = \frac{1}{\text{preciznost}^{-1} + \text{opoziv}^{-1}} = 2 * \frac{\text{preciznost} * \text{opoziv}}{\text{preciznost} + \text{opoziv}} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.4}$$

F1 metrika pokazala se najboljom za problem ovoga rada te ćemo ju u nastavku koristiti kao glavnu metriku. To će nam omogućiti jednostavne usporedbe različitih rezultata gledajući samo jedan broj. Treba napomenuti da za detaljniju analizu treba promatrati i ostale navedene metrike kako bi se dobila bolja slika o kvaliteti i razlogu rezultata.

Vrijednost F1 parametra može biti u rasponu između 0 i 1.0 (uključujući), gdje je 0 najlošiji, a 1.0 najbolji rezultat.

3.1.5 Površina ispod ROC krivulje

Površina ispod ROC krivulje (*engl. Area Under The ROC Curve (AUC)*) mjeri kvalitetu rezultata kroz sve moguće granične vrijednosti klasifikacijskog problema[17]. Ove metrika se pomoću integrala računa kao ukupna površina ispod dvodimenzionalne *engl. receiver operating characteristic curve (ROC)* krivulje. Ta krivulja proteže se između brojeva 0 i 1.0 na obje osi te time i njen integral može poprimiti vrijednosti između 0 i 1.0, uključujući.

3.2 Postavke modela

Radi pouzdanosti usporedba rezultata bitno je umanjiti utjecaj nasumičnosti u postupku vrednovanja performansi modela predviđanja. To se postiglo korištenjem iste vrijednosti varijable *random seed* u svakom treniranju. Tako je osigurano konzistentno okruženje za sva testiranja, što je jako važno za pravednu usporedbu različitih modela i tehnika optimizacija.

Jedno bitno pitanje u implementaciji svakog strojnog učenja je koliko dugo treba trenirati model. I nedovoljno i prekomjerno treniranje negativno utječu na performanse modela. Za osiguranje najboljih performansi, prilikom treniranja modela program prati funkciju gubitka nad validacijskim podacima te prekida treniranje kada funkcija gubitka počne rasti zbog *pretreniranja* ili kada funkcija počne stagnirati odnosno model prestane učiti. Nakon zaustavljanja treniranja, program vrati model nekoliko koraka ranije na trenutak u kojem je imao najbolje rezultate. Na ovaj način osiguravamo da će se model ispravno istrenirati. Ponašanja ovog mehanizma zaustavljanja mogu se najbolje vidjeti na slici 2.1 gdje graf 2.1a prikazuje prekid treniranja zbog porasta loss funkcije, a graf 2.1b prikazuje zaustavljanje zbog stagnacije rezultata nad validacijskim podacima.

3.3 Prag izlaza klasifikacije

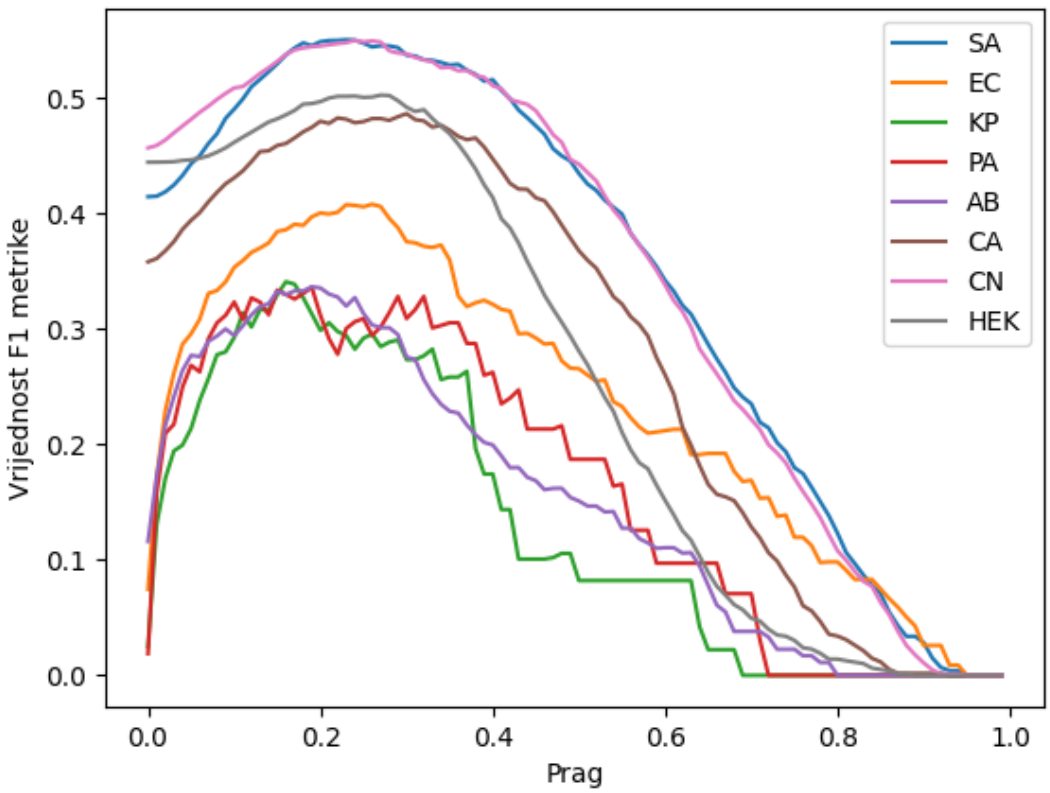
Prag (*engl. threshold*) je vrijednost za pretvorbu decimalnog broja u rasponu `[0, 1]`, što je izlaz modela za reakciju sa svakom molekulom, u konkretne vrijednosti (0 ili 1) koje označuju ima li reakcije ili ne. Prag radi tako da se sve vrijednosti ispod praga pretvore u 0, a sve iznad praga pretvore u 1.

Odabir različitih pragova daje različite rezultate modela. Za odabir najboljega praga iterira se kroz moguće pragove između 0% i 100% s korakom iteracije jednakim 1%. U toj iteraciji je svaki prag primijenjen na rezultate modela te je izračunat *F1* parametar. Konačan prag koji se odabere je onaj čija je *F1* vrijednost najveća. Ovaj proces odabira izvršava se na svakom izlazu modela zasebno, tj. zasebno za svaki organizam. Naravno, biranje praga radi se na validacijskom skupu podataka kako bi bio neovisan o treniranju, ali i kasnijem testiranju. Nakon toga se kvaliteta modela testira nad test skupom podataka.

U grafu na slici 3.1 mogu se vidjeti rezultati prikazani u *F1* metrici ovisno o različitim pragovima. Odabir pogrešnog praga može rezultirati puno lošijim modelom. Najbolji prag za odabrati je onaj čija je vrijednost F1 metrike najveća. Primjer odabranih pragova sa slike 3.1 je u tablici 3.1.

Tablica 3.1 Primjer odabranih pragova

SA	EC	KP	PA	AB	CA	CN	HEK
0.21	0.26	0.18	0.29	0.20	0.37	0.32	0.21



Slika 3.1 Usporedba različitih pragova sa F1 parametrom

3.4 Binarna klasifikacija

Sve tehnike i postupci u ovom poglavlju opisani su za modele klasifikacije s višestrukim oznakama koji kao izlaz imaju osam različitih vrijednosti. Njihova sposobnost je istovremeno predviđanje aktivnosti u odnosu na sve mikroorganizme za svaku malu molekulu koja se stavi kao ulazni podatak.

Tehnika binarne klasifikacije ima samo jedan binarni izlaz (postoji li reakcija ili ne), odnosno cijeli model trenira se samo za jedan mikroorganizam. Zbog toga je potrebno imati osam različitih modela binarne klasifikacije kako bi se postigla ista funkcionalnost kao što to ima jedan model s višestrukim oznakama. Cjelokupni

postupak i tehnike isti su kao za opisane klasifikacije s višestrukim oznakama, ali se sve ponavlja osam puta, po jednom za svaki mikroorganizam.

3.5 Modeli

Odabrana vrsta modela je neuralna mreža dubokog učenja koristeći Tensorflow biblioteku pomoću Keras programatskog sučelja. Po topologiji, model je duboka neuralna mreža sekvencijskog tipa. Svaki sloj je gusti odnosno svaki je njegov čvor povezan sa svakim čvorom prethodnog sloja.

Radi izbjegavanja pretreniravanja modela korišten su i *Dropout* slojevi u sekvencijskom modelu. Oni spomenutim gustim slojevima određen postotak težina automatski postave na 0[18].

Testiran je velik broj konfiguracija modela mijenjajući broj njihovih slojeva i čvorova u mreži. Parametri modela su mijenjani ručno metodom pokušaj-pogreške vođene ljudskom intuicijom. Testirani modeli su sadržavali između tri i osam slojeva, a broj čvorova u svakom sloju između 8 i 200. Najčešće su bile mreže koje postepeno smanjuju broj čvorova od ulaznog sloja do izlaznog. Rezultati najboljih modela će biti navedeni u odjeljku 4.1. Sljedeći logičan korak bio pokušaj optimizacije modela odnosno dobivanje boljih rezultata.

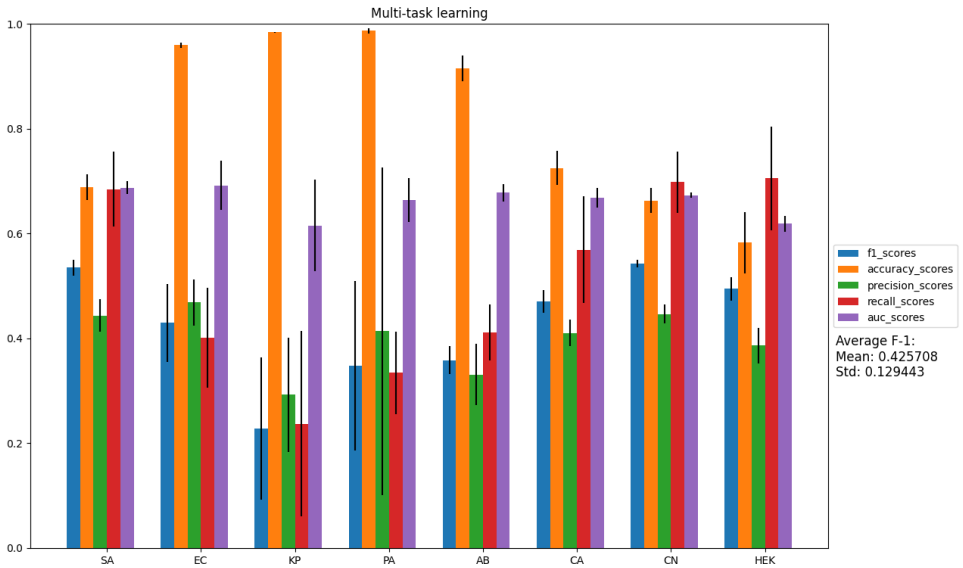
Poglavlje 4

Optimizacije rezultata

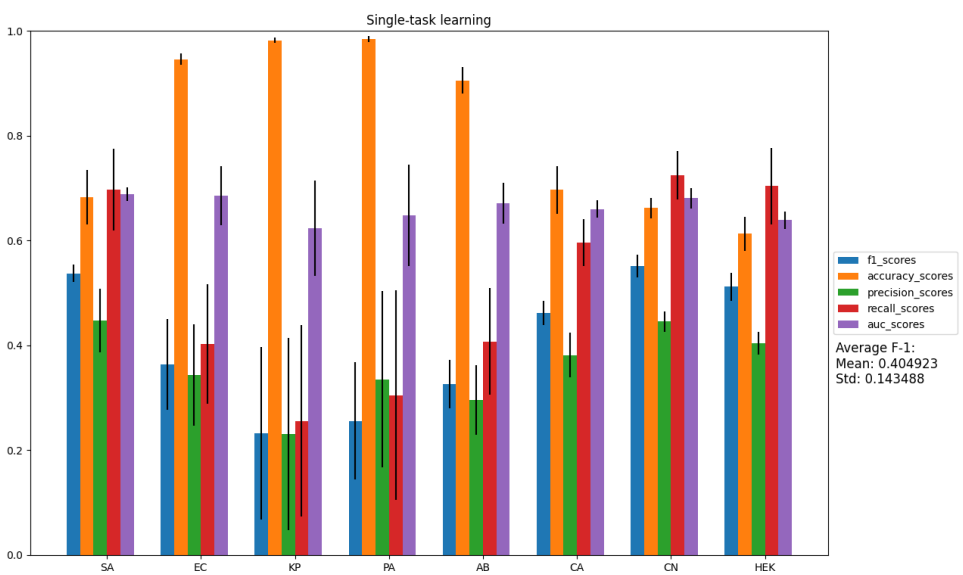
Predviđanje reakcije molekula i mikroorganizama je veoma složen problem. Zbog toga, i manjeg broja dostupnih podataka, rezultati predviđanja nisu bili zavidni. Pokušao sam primijeniti brojne popularne tehnike poboljšanja modela dubokog strojnog učenja. Sve tehnike optimizacije provedene su nad modelima klasifikacije s višestrukim oznakama i modelima binarne klasifikacije, a rezultati optimizacija su većinom bili slični. Radi jednostavnije usporedbe rezultata, najveća pažnja se posvećuje ukupnoj F1 metrici jer se ona istaknula kao najbolji pokazatelj kvalitete modela.

4.1 Rezultati prije optimizacije

Radi lakše usporedbe, rezultati modela klasifikacije s višestrukim oznakama prije primjene sljedećih optimizacija može se vidjeti na slici 4.1. Rezultati modela binarne klasifikacije prije primjene optimizacija je na slici 4.2.



Slika 4.1 Referentni rezultat klasifikacije s višestrukim oznakama prije optimizacija



Slika 4.2 Referentni rezultat binarne klasifikacije prije optimizacija

4.2 Neuravnoteženi podaci

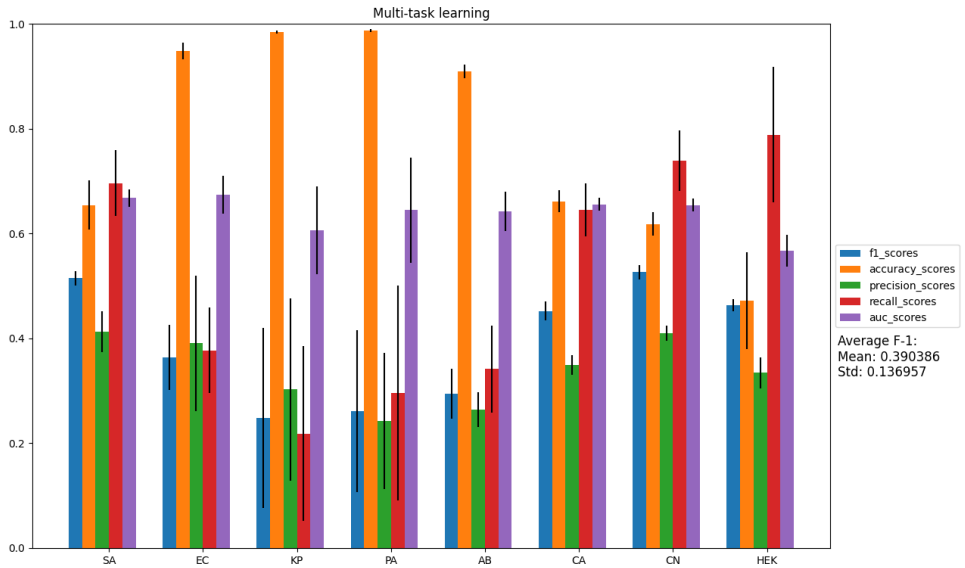
Velik problem donosi neuravnotežena priroda skupa podataka za treniranje. Broj zapisa bez reakcije je višestruko veći od broja zapisa s reakcijom kao što se može vidjeti u tablici 2.1. Izlazna klasa HEK predstavlja najbolji slučaj sa omjerom dviju klasa 1:2.468. Izlazna klasa Pseudomonas aeruginosa soj ATCC 27853 (PA) predstavlja najgori slučaj sa omjerom dviju klasa 1:107.

4.2.1 Brisanje podataka bez reakcije

Za rješavanje ovoga problema ne možemo koristiti tehniku redundantnog otipkavanja (engl. *oversampling*) jer naš model koristi tehniku klasifikacije s višestrukim oznakama. Tehnikom redundantnog otipkavanja željenih zapisa s reakcijom automatski bi dupliciralo i ostale zapise bez reakcije za ostale organizme. Ove tehnike se mogu primijeniti u klasičnom modelu binarne klasifikacije što je jedna od prednosti toga pristupa.

S druge strane, moguće je ukloniti podatke svih molekula koje ne reagiraju niti s jednim organizmom i tako smanjiti broj više zastupljenih zapisa. Ovo nije idealno jer se uklanjaju neki potencijalno bitni slučajevi iz baze podataka za treniranje modela. Od ukupno 4557 zapisa, takvih zapisa bez niti jedne reakcije je čak 1708 što znači da je broj zapisa nakon brisanja onih bez reakcije jednak 2849, samo 62.52% ukupnog broja podataka. Ovaj postupak je znatno smanjio dostupne podatke, ali time više uravnotežio podatke. Dobiveni rezultat je gori nego onaj bez brisanja podataka bez reakcije sa srednjom vrijednosti F1 parametra 0.39, a standardnom devijacijom od 0.14 kao što se to može vidjeti na slici 4.3.

Treba napomenuti da su se podaci micali samo iz skupa podatak za treniranje te su validacijski i test podaci ostali nepromijenjeni. To je jako bitno jer izmjenom testnih podataka neopravdano utječemo i na procjenu kvalitete modela. Kao dokaz tome, na slici 4.4) možemo vidjeti da je srednja vrijednost F1 metrika puno bolja, čak 51.1% sa standardnom devijacijom od 17.4%. Ovaj rezultat ne možemo gledati kao uspješnu optimizaciju modela zbog prethodno spomenute nedozvoljene izmjene test podataka. Međutim, on je dobar pokazatelj da neuravnoteženost podataka jest



Slika 4.3 Rezultat modela nakon uklanjanja zapisa bez reakcija samo u podacima za treniranje modela

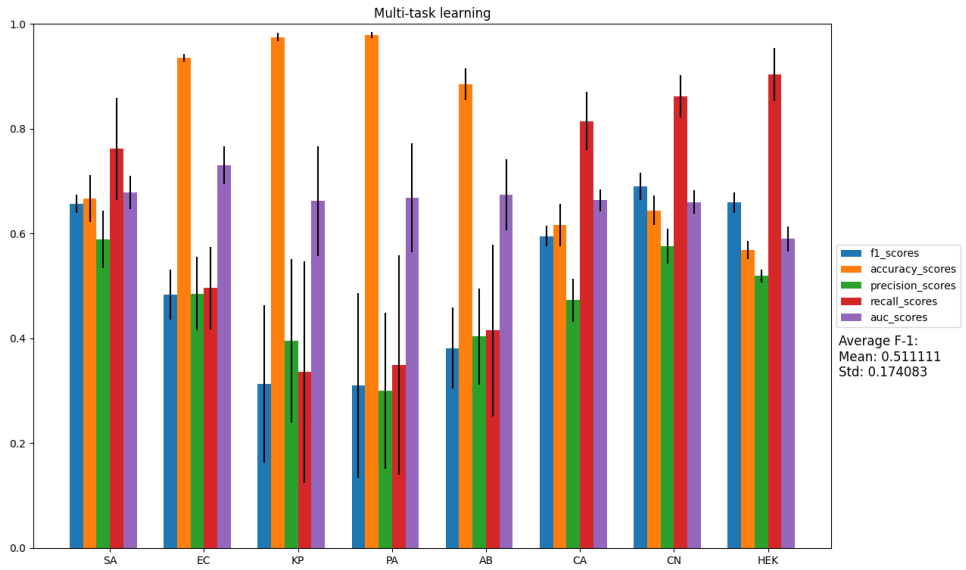
problem te da bi neko vezano rješenje moglo donijeti bolje rezultate.

4.2.2 Dupliciranje rijetkih slučajeva

Najjednostavnije rješenje neuravnoteženosti podataka je tehniku redundantnog otipkavanja. Ova tehnika moguća je samo za klasične modele s jednim izlazom, a nije moguća za modele klasifikacije s višestrukim oznakama zbog već objašnjene uske povezanosti izlaznih podataka.

Korišteno rješenje za ovaj problem je tehnika sinteze uzoraka redundantnim otipkavanjem manjinske klase (engl. *Synthetic Minority Oversampling Technique*, SMOTE) [19]. To je algoritam za predobradu podataka koji se smatra *de facto* standardom za rad s neuravnoteženim podacima [20].

Implementacija u programskom jeziku Python je jednostavna zbog postojeće standardne knjižnice `imblearn.over_sampling.SMOTE`[21, 22]. U suštini, implemen-



Slika 4.4 Rezultat modela nakon uklanjanja svih zapisa bez reakcija

tacija se svodi samo na odluku strategije uzorkovanja odnosno jednog broja koji označuje željeni omjer broja uzoraka iz manje zastupljene klase naprema broju uzoraka iz zastupljenije klase. Točnije, moguće je odabrati željeni omjer α za koji vrijedi:[21]

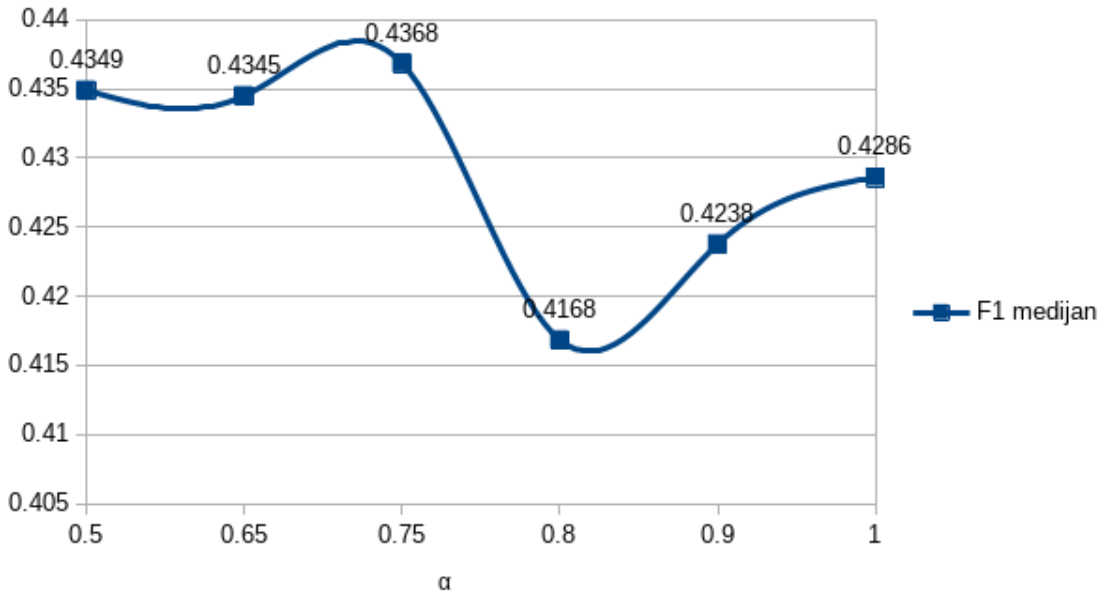
$$\alpha = N_{rm}/N_M \tag{4.1}$$

gdje je:

- N_{rm} broj instanca manje zastupljene klase nakon proširivanja broja podataka
- N_M broj instanca zastupljenijih klasa

SMOTE se računao zasebno za svaki od osam mikroorganizama, ali uvijek s istim željenim omjerom zastupljenosti klasa. Naravno, SMOTE tehnika vrši se samo nad skupom podataka za treniranje. Srednje vrijednost F1 rezultata za svaku testiranu α vrijednost mogu se vidjeti na slici 4.5.

Primjena SMOTE za $\alpha = 0.2$ i manje rezultira greškom jer je to manji omjer nego

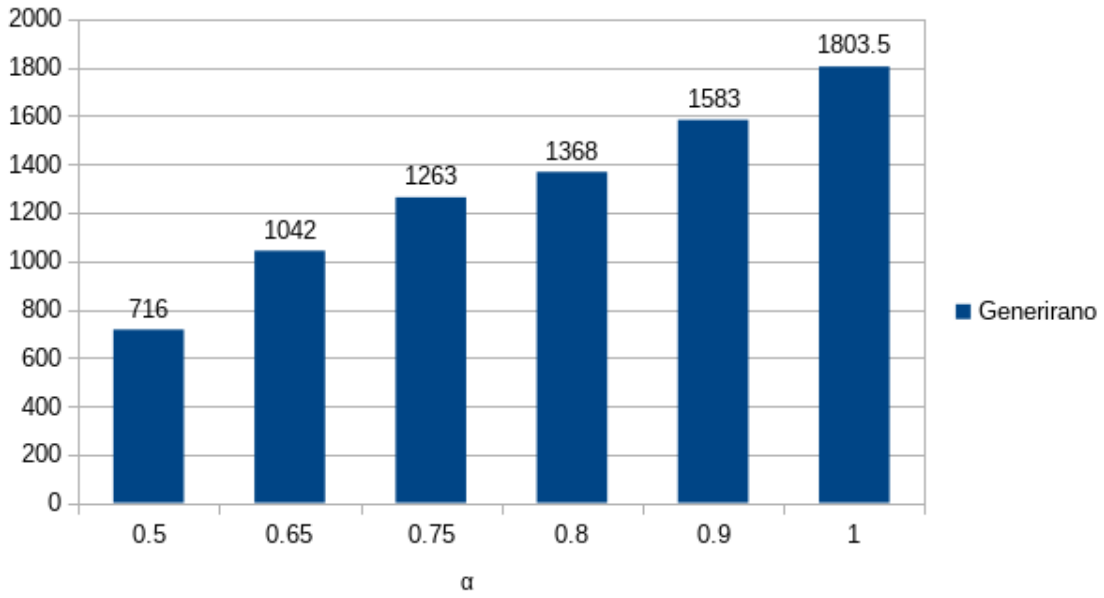


Slika 4.5 Rezultati modela za različite α vrijednosti

prije primjene tehnike. To bi značilo efektivno brisanje manje zastupljenih podataka što nije cilj ove tehnike.

Prije ovog povećanja, ima ukupno 2551 podataka za svaku molekulu. Na slici 4.6 prikazan je prosječan broj zapisa koje je dodan za svaku od α vrijednost. Logično je da veća α vrijednost znači i veći broj dodanih zapisa.

Iz prikazanih rezultata vidljivo je da SMOTE tehnika poboljšava rezultate predviđanja za dani skup podataka.Vrijednost varijable $\alpha = 0.75$ daje najbolji pronađeni rezultat za SMOTE tehniku s F1 parametrom jednakim 0.4368 što je bolje nego rezultat modela prije primjene tehnike s F1 parametrom jednakim 0.4049. Iako su povećanja metrika rezultata relativno malena, tehnika SMOTE je, u usporedbi s drugim tehnikama unaprijeđenja performansi predviđanja, dala najveći napredak rezultata. Mana ove tehnike je nemogućnost primjene na model klasifikaciju s višestrukim oznakama (engl. *Multi-task learning*, MTL).



Slika 4.6 Broj generiranih zapisa za različite α vrijednosti

4.2.3 Pretvorba problema označavanja u problem klasifikacije

Složenost problema riješenim tehnikom klasifikacije s višestrukim oznakama raste povećanjem broja klasa na izlazu modela. Sva rješenja do sada istražena u ovome radu koja koriste ovu tehniku imaju na izlazu osam različitih oznaka od kojih svaka nezavisno može poprimiti dva moguća stanja. To znači da naš model podržava $2^8 = 256$ različitih izlaznih stanja. Neki od tih stanja su potencijalno nemogući, tj. ne postoje u skupu podataka. Ti nemogući izlazi su mogući izvor šuma u modelu te mogući uzrok lošijih rezultata u slučaju da izlaz modela odgovara jednom od nepostojećih stanja.

Ideja iza ove optimizacije je pronaći sva moguća stanja izlaza te ih pretvoriti u zasebne klase. Zatim možemo dosadašnji model označavanja zamijeniti novim modelom tipa klasifikacije. Neuronska mreža klasifikacije ima onoliko izlaza koliko postoji klasa. Ona kao rezultat predviđanja za svaku od klasa, odnosno na svakom izlazu postavi decimalni broj između 0 i 1 koji predstavlja predviđenu vjerojatnost da je dani ulaz ispravno klasificirati kao tu određenu klasu. Svi izlazi modela su tako raspoređeni da je njihov zbroj uvijek jednak jedan za što je zadužena aktivacijska

funkcija `softmax`. Za odabir klase se odabere ona s najvećom predviđenom vjerojatnosti. U nastavku je opisan postupak pretvorbe problema označavanja u problem klasifikacije kao i dobiveni rezultati.

Predobrada podataka

Ideja je niz od osam rezultata predviđanja pretvoriti u jednu klasu. Svaka kombinacija rezultata bit će označena različitom klasom. Popis klasa sortiranih po broju zapisa koji pripadaju toj klasi može se vidjeti u tablicama 4.1 i 4.2. U tablicama su za svaku klasu prikazane značajke te klase odnosno binarne oznake postoji li kemijska reakcija molekula pripadne klase s navedenim mikroorganizmima.

Izračun u koju klasu pripada određen rezultat odnosno skup kemijskih reakcija osmišljen je tako da se osam mikroorganizama gleda kao osam bitova jednog binarnog zapisa. Klasa je zatim jednostavna pretvorba osam bitnog binarnog broja u dekadski zapis kao što se može vidjeti u već spomenutim tablicama 4.1 i 4.2.

Iz prethodnog opisa izračuna klase slijedi da je maksimalan mogući broj klasa za osam mikroorganizama jednak $2^8 = 256$ dok je ukupan broj stvarnih klasa jednak 78, od čega svega 29 klasa ima deset ili više pojavljivanja u skupu podataka. Iz toga možemo zaključiti da binarni rezultati bioloških reakcija nisu nasumični već usko povezani. Primjerice, ako znamo da određena mala molekula ima kemijsku reakciju s Cryptococcus neoformans (CN) onda postoji osjetno veća vjerojatnost da ona također ima reakciju s Candida albicans (CA) mikroorganizmom. Upravo ova povezanost dovela je do ideje da klasifikacija s višestrukim oznakama može poboljšati rezultate modela zbog mogućnosti iskorištavanja zajedničkog znanja o više mikroorganizama.

Rezultati modela

Prije izračuna kvalitete predviđanja bioloških reakcija metodom klasifikacije potrebno je model za klasifikaciju s višestrukim oznakama izmijeniti u model binarne klasifikacije. Potrebno je funkciju gubitka postaviti na `categorical_crossentropy`, aktivacijsku funkciju izlaznog sloja modela na `softmax` te broj čvorova izlaznog sloja postaviti da bude jednak broju klasa. Zatim, nakon dobivanja rezultata modela po-

trebno je izlaze zadnjeg sloja modela pretvoriti u zapis gdje se jedino čvor s najvećom vrijednosti pretvara u 1, a svi ostali u 0. Tako će model uvijek predvidjeti samo jednu klasu za dani ulaz.

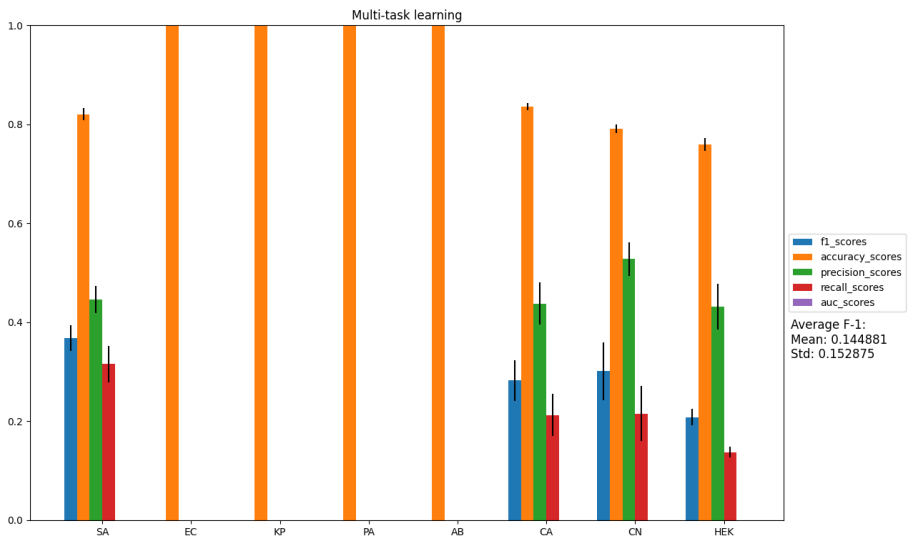
Što je veći broj klasa na izlazu time se složenost modela povećava, a vjerojatnost točnog predviđanja reakcije a priori smanjuje. Iz tog razloga te da bi se testiranju ove tehnike dale sve mogućnosti većeg rezultata odabrano je samo 10 najzastupljenijih klasa za treniranje. Podaci za validaciju i testiranje će i dalje sadržavati sve podatke nevezano o klasi kako bi rezultati bili reprezentativniji.

Treba biti oprezan pri usporedbi rezultata ovog modela s ostalim modelima jer se može dogoditi da koriste potpuno druga rješenja za predobrade podataka, treniranja i izračuna rezultata. Taj problem sam izbjegao tako da se nakon rezultata modela broj klase pretvori u značajke kao kod drugih pristupa te se rezultati modela računaju na potpuno isti način kao u ostalim pristupima. Štoviše, programski kod za izračun rezultata nije uopće mijenjan.

Rezultati modela nakon treniranja sa samo 10 najzastupljenijih klasa mogu se vidjeti na slici 4.7. Očito je da su oni puno gori nego druge metode. Preciznije, srednje vrijednosti te standardne devijacije svake metrike su u tablici 4.3. Uzimajući standardnu devijaciju u obzir, za ove rezultate se može reći da su zanemarivi odnosno praktički 0.0.

Treba napomenuti da je Area Under The ROC Curve (AUC) metrika uvijek jednaka nuli. Razlog tome je greška računanja AUC metrike zbog nedostatka nekih značajki u podskupu podataka za testiranje. Ako pogledamo 10 najzastupljenijih klasa te njihovu reaktivnost s pojedinim mikroorganizmima (tablica 4.1) možemo vidjeti da mikroorganizmi Escherichia coli (EC), Klebsiella pneumoniae (KP), PA i Acinetobacter baumannii (AB) nikada nemaju reakcije s niti jednom od klasa. Već ova informacija je mogla biti indikacija da pristup pretvaranja problema u klasifikaciju te testiranje nad najzastupljenijim značajkama nije kvalitetno rješenje.

Ovaj eksperiment proveden je i s 20 najzastupljenijih klasa te sa svim klasama. Rezultati su također niski te je odustano od primjene ove tehnike.



Slika 4.7 Rezultat modela nakon treniranje nad 10 najzastupljenijih klasa

Tablica 4.1 Najzastupljenije klase i popis njihovi značajki (prvih 39 od 78)

	Klasa	Broj	SA	EC	KP	PA	AB	CA	CN	HEK
1	0	1708	0	0	0	0	0	0	0	0
2	1	455	0	0	0	0	0	0	0	1
3	128	389	1	0	0	0	0	0	0	0
4	2	335	0	0	0	0	0	0	1	0
5	6	301	0	0	0	0	0	1	1	0
6	129	259	1	0	0	0	0	0	0	1
7	4	153	0	0	0	0	0	1	0	0
8	7	136	0	0	0	0	0	1	1	1
9	135	111	1	0	0	0	0	1	1	1
10	3	88	0	0	0	0	0	0	1	1
11	134	81	1	0	0	0	0	1	1	0
12	131	58	1	0	0	0	0	0	1	1
13	130	56	1	0	0	0	0	0	1	0
14	5	44	0	0	0	0	0	1	0	1
15	8	43	0	0	0	0	1	0	0	0
16	132	21	1	0	0	0	0	1	0	0
17	64	20	0	1	0	0	0	0	0	0
18	133	20	1	0	0	0	0	1	0	1
19	143	18	1	0	0	0	1	1	1	1
20	207	16	1	1	0	0	1	1	1	1
21	14	14	0	0	0	0	1	1	1	0
22	137	14	1	0	0	0	1	0	0	1
23	248	14	1	1	1	1	1	0	0	0
24	9	13	0	0	0	0	1	0	0	1
25	12	12	0	0	0	0	1	1	0	0
26	200	12	1	1	0	0	1	0	0	0
27	136	11	1	0	0	0	1	0	0	0
28	199	11	1	1	0	0	0	1	1	1
29	192	10	1	1	0	0	0	0	0	0
30	10	9	0	0	0	0	1	0	1	0
31	239	9	1	1	1	0	1	1	1	1
32	193	8	1	1	0	0	0	0	0	1
33	255	7	1	1	1	1	1	1	1	1
34	201	6	1	1	0	0	1	0	0	1
35	232	6	1	1	1	0	1	0	0	0
36	11	5	0	0	0	0	1	0	1	1
37	15	5	0	0	0	0	1	1	1	1
38	203	5	1	1	0	0	1	0	1	1
39	139	4	1	0	0	0	1	0	1	1

Tablica 4.2 Najzastupljenije klase i popis njihovi značajki (drugih 39 od 78)

	Klasa	Broj	SA	EC	KP	PA	AB	CA	CN	HEK
40	142	4	1	0	0	0	1	1	1	0
41	147	4	1	0	0	1	0	0	1	1
42	195	4	1	1	0	0	0	0	1	1
43	206	4	1	1	0	0	1	1	1	0
44	216	4	1	1	0	1	1	0	0	0
45	65	3	0	1	0	0	0	0	0	1
46	194	3	1	1	0	0	0	0	1	0
47	238	3	1	1	1	0	1	1	1	0
48	240	3	1	1	1	1	0	0	0	0
49	70	2	0	1	0	0	0	1	1	0
50	71	2	0	1	0	0	0	1	1	1
51	78	2	0	1	0	0	1	1	1	0
52	79	2	0	1	0	0	1	1	1	1
53	196	2	1	1	0	0	0	1	0	0
54	224	2	1	1	1	0	0	0	0	0
55	233	2	1	1	1	0	1	0	0	1
56	249	2	1	1	1	1	1	0	0	1
57	31	1	0	0	0	1	1	1	1	1
58	32	1	0	0	1	0	0	0	0	0
59	37	1	0	0	1	0	0	1	0	1
60	56	1	0	0	1	1	1	0	0	0
61	72	1	0	1	0	0	1	0	0	0
62	75	1	0	1	0	0	1	0	1	1
63	80	1	0	1	0	1	0	0	0	0
64	88	1	0	1	0	1	1	0	0	0
65	96	1	0	1	1	0	0	0	0	0
66	138	1	1	0	0	0	1	0	1	0
67	145	1	1	0	0	1	0	0	0	1
68	146	1	1	0	0	1	0	0	1	0
69	175	1	1	0	1	0	1	1	1	1
70	197	1	1	1	0	0	0	1	0	1
71	198	1	1	1	0	0	0	1	1	0
72	202	1	1	1	0	0	1	0	1	0
73	204	1	1	1	0	0	1	1	0	0
74	208	1	1	1	0	1	0	0	0	0
75	209	1	1	1	0	1	0	0	0	1
76	225	1	1	1	1	0	0	0	0	1
77	250	1	1	1	1	1	1	0	1	0
78	254	1	1	1	1	1	1	1	1	0

Tablica 4.3 Rezultati modela nakon treniranja nad 10 najzastupljenijih klasa

	F1	Točnost	Preciznost	Opoziv	AUC
Srednja vrijednost	0.145	0.901	0.230	0.110	0.0
Standardna devijacija	0.153	0.102	0.234	0.122	0.0

4.3 Hiperparametri

Uspješnost modela uvelike ovisi o odabranim hiperparametrima. Hiperparametri su parametri o čije vrijednosti utječu na proces strojnog učenja, a time i kasnijih rezultata[23]. Neki od hiperparemetara su broj slojeva duboke neuronske mreže, veličina svakog od tih slojeva, stopa učenja, duljina učenja (npr. maksimalan broj epoha) i sl. Ovaj rad pridonosi najveći fokus broju slojeva duboke neuronske mreže, njihovim veličinama te stopa učenja.

Pri početnoj izradi modela, navedeni hiperparametri određeni su metodom pokušaja i pogreške te korištenjem "zdrave logike". Ova ručna metoda doprinijela je boljim rezultatima, ali ona nije idealna. Uvođenjem automatske metode odredbe optimalnih hiperparametara mogli bismo poboljšati rezultate modela.

Keras tuner pogodan je razvojni okvir (engl. *framework*) za automatsko pretraživanje hiperparametara[24]. On nudi brojne algoritme pretraživanja od kojih ćemo se fokusirati na jednostavnije nasumično pretraživanje (engl. *random search*)[25] te složeniji *hyperband*[26, 27].

Implementacija[28] koristi sljedeće parametre te njihove raspone: - broj slojeva duboke neuronske mreže u rasponu [4, 8] - zasebni brojevi čvorova pojedinog sloja neuronske mreže u rasponu [8, 512] - faktor *dropout* sloja neuronske mreže u rasponu [0%, 70%] - stopa učenja u rasponu [1e-5, 1e-2]

Algoritmi pretraživanja hiperparametara u suštini rade tako da odaberu jedan skup hiperparametara, evaluiraju njegovu kvalitetu nad podacima, iz toga izračunaju sljedeći skup hiperparametara te ponavljaju spomenuti proces. Razvojni okvir automatski pamti procijenjeno najbolji skup hiperparametara do tog trenutka.

Rezultati pretrage *hyperband* algoritma odnosno najbolji pronađeni hiperparametri navedeni su u tablici 4.4. Tablica sadrži tri stupca rezultata. Prvi stupac (*Točnost)* predstavlja rezultate algoritma pri traženju najbolje moguće točnosti. Drugi stupac rezultata (TOP_1) predstavlja najbolje pronađene hiperparametre pri traženju najbolje kombinacije točnosti, preciznosti i opoziva s jednakom važnosti za sve tri metrike. Dva prethodna pretraživanja najboljih hiperparametara izvršena su nad cijelim skupom podataka. Podešavanje modela na dane hiperparametre te trenira-

nje stalno je davalo gore rezultate nego što je to algoritam pretraživanja predvidio. Razlog te nekonzistentnosti bio je skup podataka nad kojima se vršio odabrani algoritam pretraživanja. Kao ulaz *hyperband* algoritmu dan je cijeli skup podataka dok za kasnije treniranje modela to nije slučaj. Za treniranje pravog modela potrebno je podijeliti podatke na skup podataka za treniranje, validaciju i testiranje. U treći stupac u tablici (TOP_2) zapisani su rezultati algoritma pretraživanja pri traženju najbolje kombinacije točnosti, preciznosti i opoziva (kao i TOP_1), ali s razlikom da su za stupac TOP_2 podaci raspodijeljeni na identičan način kao i za treniranje pravoga modela.

U nastavku ovog poglavlja navodim rezultate modela s navedenih tri skupa hiperparametara.

Tablica 4.4 Najbolji rezultati hyperband *algoritma pretrage najbolje točnosti, preciznosti i opoziva*

	Točnost	TPO_1	TPO_2
Stopa učenja	0.0008267	0.000029643	0.000021407
Broj skrivenih slojeva	8	8	4
Veličina skrivenog sloja 1	128	352	416
Dropout skrivenih slojeva 1-2	0.5	0.0	0.1
Veličina skrivenog sloja 2	80	200	312
Dropout skrivenih slojeva 2-3	0.2	0.1	0.6
Veličina skrivenog sloja 3	432	152	200
Dropout skrivenih slojeva 3-4	0	0.6	0.1
Veličina skrivenog sloja 4	80	504	104
Dropout skrivenih slojeva 4-5	0.2	144	/
Veličina skrivenog sloja 5	144	248	/
Dropout skrivenih slojeva 5-6	0.1	0.6	/
Veličina skrivenog sloja 6	240	328	/
Dropout skrivenih slojeva 6-7	0.6	0.4	/
Veličina skrivenog sloja 7	48	8	/
Dropout skrivenih slojeva 7-8	0.1	0.2	/
Veličina skrivenog sloja 8	16	296	/

Predviđena točnost za hiperparametre *Točnost* u tablici 4.4 je 0.741228, dok je stvarna točnost nakon treniranja nad podacima jednaka 0.775401. Za ovu razliku

vjerojatno je zaslužena različitost nasumične distribucije podataka u pretraživanju optimalnih hiperparametara i treniranju stvarnog modela. Svi rezultati ovoga modela nakon treniranja prikazani su u tablici 4.5. Možemo primijetiti da prema F1 metrici ovaj model nije bolji od dosadašnjeg. Razlog toga je traženje najveće točnosti umjesto F1 metrike. Kako je F1 metrika kombinacija preciznosti i opoziva, sljedeći korak je pokrenuti algoritam pretraživanja hiperparametara koji uključuje i te metrike.

Tablica 4.5 Rezultati modela s hiperparametrima za najbolju točnost

	F1	Točnost	Preciznost	Opoziv	AUC
Srednja vrijednost	0.396	0.775	0.352	0.536	0.647
Standardna devijacija	0.127	0.204	0.104	0.246	0.078

Tablica 4.6 pokazuje rezultate modela s najboljim pronađenim hiperparametrima u potrazi za najboljom kombinacijom točnosti, preciznosti i opoziva (TOP_1). Rezultati su puno gori od očekivanih, tj. referentnog modela. Moguće je da predviđanje hiperparametara nije realno jer se ono odvija nad različito raspoređenim podacima nego samo treniranje. Pretraživanje koristi sve podatke, a treniranje samo dio zbog potrebe boljeg vrednovanja istreniranog modela.

Tablica 4.6 Rezultati modela s hiperparametrima za najbolju točnost, preciznost i opoziv

	F1	Točnost	Preciznost	Opoziv	AUC
Srednja vrijednost	0.336	0.731	0.294	0.501	0.596
Standardna devijacija	0.155	0.235	0.145	0.297	0.065

Iz tog razloga sam pokrenuo pretraživanje hiperparametara nad jednakim dijelom podataka kao i samo treniranje (TOP_2). Njegovi rezultati nalaze se u tablici 4.7.

Ovi rezultati su puno bliže referentnom modelu dobivenog pomoću metode pokušaj-pogreške praćenom ljudskom intuicijom. Pronađeni hiperparametri nisu doprinijeli poboljšanju aktualnog modela, ali služe kao dobra potvrda da su odabrani hiperparametri adekvatni za promatrani problem. Ulaganjem mnogo više vremena

Tablica 4.7 Rezultati modela s hiperparametrima za najbolju točnost, preciznost i opoziv s jednakom raspodjelom podataka

	F1	Točnost	Preciznost	Opoziv	AUC
Srednja vrijednost	0.404	0.813	0.380	0.485	0.654
Standardna devijacija	0.153	0.153	0.162	0.212	0.057

i procesorske moći u pretraživanje optimalnih hiperparametra moglo bi dovesti do boljih rezultata, ali procijenjeno je kako unaprjeđenje ne bi bilo značajno.

4.4 Analiza rezultata

Sljedeći korak u pokušavanju poboljšanja rezultata našeg rješenja je analiza rezultata s ciljem boljeg shvaćanja razloga točnih i netočnih predviđanja. Rezultati analize dat će nam bolju sliku o tome koje sljedeće tehnike optimizacije se isplati pokušati, tj. od kojih tehnika se može očekivati najbolji napredak.

4.4.1 Euklidske udaljenosti ulaznih podataka

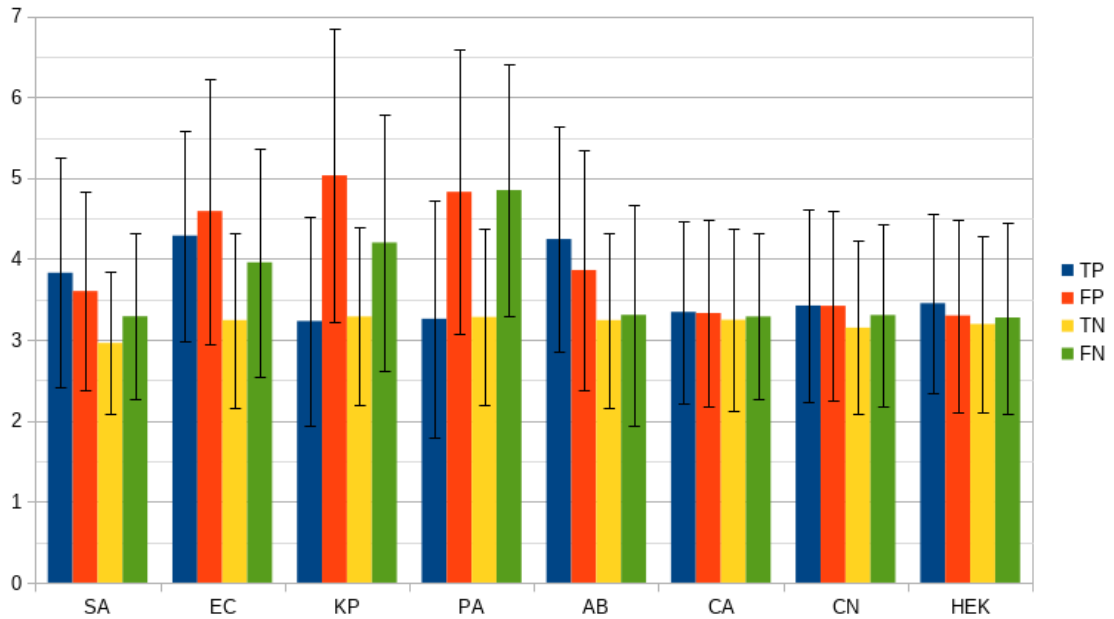
Ova analiza provjerava hipotezu da su značajke na ulazu modela jako slične za one podatke koji imaju jednak izlaz. Drugim riječima, ako gledamo ulaz u model kao vektor s 81 dimenzijom onda je euklidska udaljenost dvaju molekula koje reagiraju s istim mikroorganizmom mala, a euklidska udaljenost tih molekula s molekulama koje s tim mikroorganizmom ne reagiraju značajno veća.

Prvo sam rezultate predviđanja modela podijelio na četiri skupine: true positive (TP), false positive (FP), true negative (TN) i false negative (FN). Zatim sam za svaku skupinu zasebno izračunao sve moguće parove podataka te njihove euklidske udaljenosti. Rezultati na slikama 4.8 i 4.9 predstavlja srednje vrijednosti 81-dimenzionalnih euklidskih udaljenosti svih parova zasebnih skupina i njihovu standardnu devijaciju. Može se vidjeti da su sve srednje vrijednosti od prilike jednake te unutar standardne devijacije ostalih. To znači da nema većih odstupanja između ove četiri kategorije - rezultat modela ne ovisi o unutarnjoj udaljenosti točaka raznih skupina odnosno ta udaljenost ne utječe na rezultat modela.

Također se može primijetiti da su standardne devijacije relativno velike. Velika standardna devijacija inače ukazuje na visoku raspršenost podataka, ali u ovom slučaju to ima smisla zbog prirode mjerenja udaljenosti svih parova.

Ono što nije toliko vidljivo iz navedenih slika, ali je svakako prisutno, je povezanost između broja elemenata svake skupine s veličinom srednjih vrijednosti i standardnih devijacija. Što je broj elemenata skupine veći, to je generalno srednja vrijednost i standardna devijacija manja. Moguć i vjerojatan uzrok tome može biti nasumična raspodjela višedimenzionalnih ulaznih točaka za model. Veći broj nasumičnih točaka u nekom prostoru značit će i veći broj bližih točaka što će smanjiti

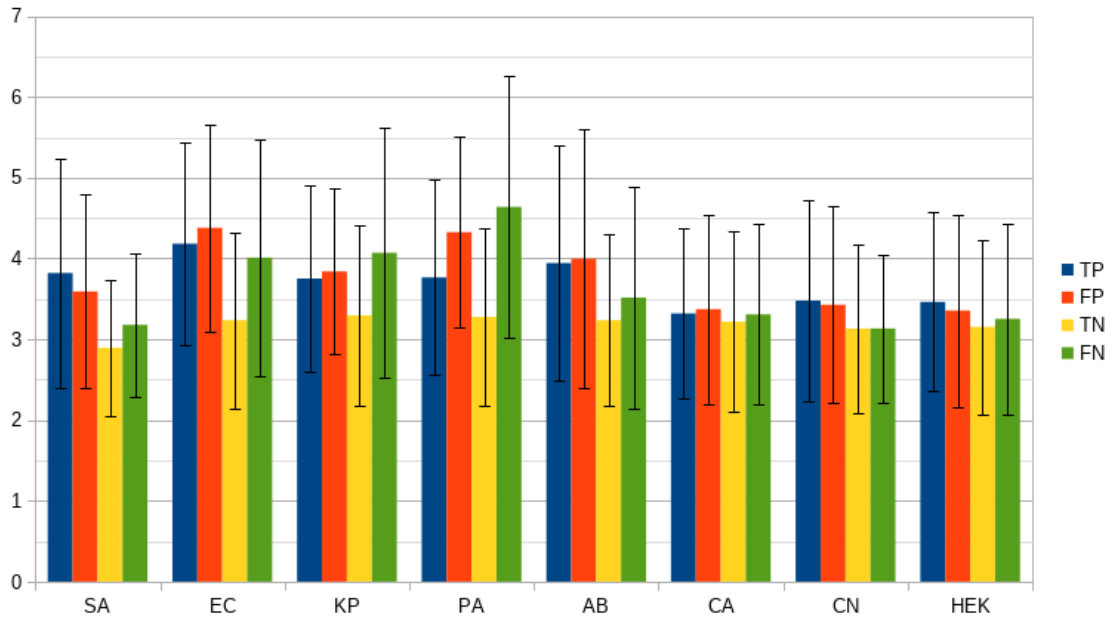
srednju vrijednost njihovih udaljenosti.



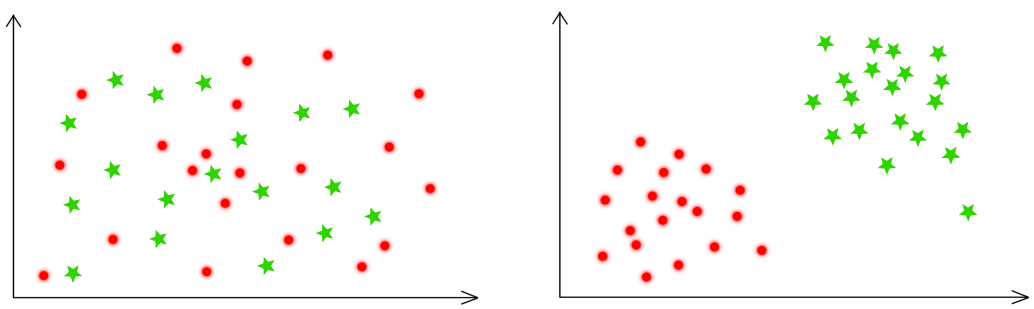
Slika 4.8 Srednje vrijednosti udaljenosti unutar skupina nakon treniranja modela višestruke klasifikacije

Sada znamo da su svi od četiri skupa približno jednako nasumično raspodijeljeni, ali ne znamo jesu li skupine zasebno grupirane. Dva trenutno moguća slučajeva prikazana su na slici 4.10 gdje su dva različita skupa podataka prikazana crvenim krugom odnosno zelenom zvijezdom. Na slici 4.10b svaka je skupina zasebno grupirana te se problem predviđanja reakcije svodi na odabir najbliže skupine za danu točku. Ovakva raspodjela uvijek je poželjnija jer olakšava predviđanje skupina. U slučaju slike 4.10a skupine su nasumično isprepletene te za danu točku nije moguće odrediti kojoj skupini ona pripada.

Kako bismo odredili na koji od dva navedena načina su skupine raspoređene, za svaku sam skupinu izračunao prosječnu točku u 81-dimenzionalnom prostoru te zatim izračunao sve udaljenosti između različitih prosječnih točaka. Prosječne matrice udaljenosti svih mikroorganizama se može vidjeti u tablicama 4.8 i 4.9. Iz matricama udaljenosti skupova i navedenih udaljenosti točaka unutar svakog skupa može se zaključiti da su ulazni podaci u odnosu na izlazne raspodijeljeni na nasumičan



Slika 4.9 Srednje vrijednosti udaljenosti unutar skupina nakon treniranja modela binarne klasifikacije



(a) Nasumični podaci (b) Euklidski povezani skupovi

Slika 4.10 Usporedba mogućih raspodjela podataka

način kao što je to prikazano na slici 4.10a. Svaka srednja vrijednost međusobnih udaljenosti svake skupine je veća od 3.1, a svaka udaljenost različitih skupina iz matrice udaljenosti manja je od 0.6. Iz te usporedbe možemo zaključiti da se različite

skupine značajno preklapaju. Zato za danu točku nije moguće trivijalno odrediti kojoj od skupina ona pripada.

Tablica 4.8 Prosječna matrica udaljenosti značajki između svih skupina rezultata predviđanja za klasifikaciju s višestrukim oznakama

	TP	FP	TN	FN
TP	0	0.169	0.528	0.449
FP	0.169	0	0.501	0.415
TN	0.528	0.501	0	0.188
FN	0.449	0.415	0.188	0

Tablica 4.9 Prosječna matrica udaljenosti značajki između svih skupina rezultata predviđanja za binarnu klasifikaciju

	TP	FP	TN	FN
TP	0	0.152	0.508	0.408
FP	0.152	0	0.485	0.353
TN	0.508	0.485	0	0.197
FN	0.408	0.353	0.197	0

4.4.2 Usporedba modela za predviđanje višestrukih i jednostrukih oznaka

Također korisna informacija je griješe li model klasifikacije s višestrukim oznakama na istim podacima kao i model binarne klasifikacije. Analiza je napravljena nad cijelim skupom podataka, ali uvijek nad test skupom podataka. Implementacija pomoću *K-Fold*[11] funkcije podijeli sve podatke na skup za treniranje i testiranje u omjeru 4:1. Takva podjela napravi se pet puta uvijek s različitim podacima. Nakon svih pet iteracija svaki je podatak točno jednom bio u skupu podataka za testiranje.

Raspodjela točnosti predviđanja za dva različita modela mogu se vidjeti na tablicama 4.10 i 4.11. Točnosti ovise o mikroorganizmu, a kumulativno je puno veći broj točnih nego netočnih predviđanja.

Tablica 4.10 Raspodjela točnosti predviđanja za model klasifikacije s višestrukim oznakama

	SA	EC	KP	PA	AB	CA	CN	HEK	$\sum$
točno	3196	4360	4482	4491	4189	3355	3011	2606	29690
netočno	1361	197	75	66	368	1202	1546	1951	6766

Tablica 4.11 Raspodjela točnosti predviđanja za model binarne klasifikacije

	SA	EC	KP	PA	AB	CA	CN	HEK	$\sum$
točno	3052	4334	4475	4484	4190	3136	2979	2692	29342
netočno	1505	223	82	73	367	1421	1578	1865	7114

Točnosti su veoma slične, ali se postavlja pitanje koji postotak točnih i netočnih predviđanja su jednaki za oba modela, a koliko se modeli razlikuju. Iz tablice 4.12 može se vidjeti da je broj jednakih predviđanja dvaju modela puno veći nego broj različitih predviđanja (izlaza). Uzimajući samo krive rezultate modela može se vidjeti da u 56.15% slučajeva oba modela naprave istu grešku, dok se u nezanemarivoj manjini slučajeva oni razlikuju. Detaljniju distribuciju tih slučajeva kada je samo jedan od modela netočno predvidio rješenje može se vidjeti na tablici 4.13. Tu se može primijetiti da je model klasifikacije s višestrukim oznakama nešto precizniji s 45.54% netočnih predviđanja.

Tablica 4.12 Distribucije jednakih i različitih predviđanja dvaju modela

	SA	EC	KP	PA	AB	CA	CN	HEK	$\sum$
oba točna	2730	4280	4444	4454	4040	2841	2610	2168	27567
točan/netočan	788	134	69	67	299	809	770	962	3898
oba netočna	1039	143	44	36	218	907	1177	1427	4991

Tablica 4.13 Usporedba modela klasifikacije s višestrukim oznakama (MTL) i modela binarne klasifikacije (STL) za netočne i različita predviđanja

	SA	EC	KP	PA	AB	CA	CN	HEK	$\sum$
MTL netočan	322	54	31	30	150	295	369	524	1775
STL netočan	466	80	38	37	149	514	401	438	2123

4.5 Uklanjanje kontradiktornih podataka

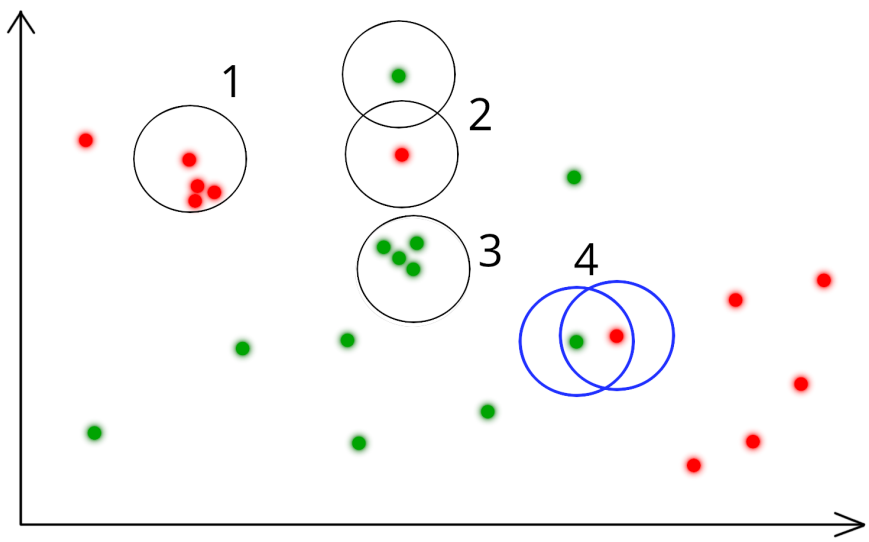
Ako ulazne značajke molekula gledamo kao vektor odnosno točku u 81 dimenziji onda možemo izmjeriti udaljenost različitih podataka. Neuralne mreže općenito teže tome da podaci s jako bliskom euklidskom udaljenosti imaju jednake rezultate na izlazu modela. Ideja iza ove optimizacije je izbaciti iz podataka za treniranje one točke koje jesu jako blizu, ali očekuju različit rezultat na izlazu kako bi manje mogle "zbuniti" model pri treniranju.

Te točke micat ćemo samo iz podataka za treniranje te će podaci za validaciju i testiranje ostati isti što je bitno radi kvalitetnije usporedbe rješenja. Algoritam uklanjanja točaka je sljedeći:

1. izračunati udaljenosti između svih točaka u podskupu podataka za treniranje
2. izdvojiti sve parove točaka koji su bliži od neke granične vrijednosti
3. ako su rezultati obje točke para jednaki, zanemari taj par i nastavi dalje
4. ako su rezultati obje točke para različiti, makni onu točku koja nema reakciju zato što je takvih više te su podaci s reakcijom manje zastupljeni i time bitniji

Slika 4.11 shematski prikazuje karakteristične primjere važne za algoritam za uklanjanje susjednih kontradiktornih točaka. Primjer je radi vizualizacije sveden na samo dvije dimenzije za razliku od stvarnih 81 dimenzija. Crvene točke na slici predstavljaju podatke s reakcijom dok zelene točke predstavljaju podatke bez reakcije molekule i mikroorganizma. Kružnice nacrtane oko nekih točaka predstavljaju raspon granične vrijednosti unutar koje se brišu sve bliže točke. Te kružnice odnosno granične vrijednosti u stvarnosti postoje oko svih točaka. U kružnicama s oznakama 1 i 3 niti jedna točka neće biti uklonjena jer su sve točke unutar kružnica jednakih

vrijednosti. Primjer s oznakom 2 prikazuje dvije točke odnosno kružnice. Iako se na primjeru 2 kružnice preklapaju, suprotne točke se ne nalaze unutar kružnica odnosno one su izvan graničnih vrijednosti pa se ne brišu nikoje točke. Kružnice primjera 4 istaknute su debljom plavom linijom jer će se jedino u tom primjeru jedna od točaka brisati iz skupa podataka. Možemo vidjeti da su obje točke unutar kružnica druge točke odnosno njihova udaljenost je unutar graničnih vrijednosti te da su one konfliktne (različitog tipa). Zelena točka (podatak bez reakcije) bit će izbrisana zato što crvenih točaka (podataka s reakcijom) ima manje.



Slika 4.11 Ilustracija uklanjanja obližnjih konfliktnih točaka

Ovaj algoritam pokretao se pet puta za pet različitih *K-Fold* iteracija. Za svaki od navedenih graničnih vrijednosti napisano je koliko je podataka izbrisano iz u podskupa za treniranje za tu *K-Fold* iteraciju.

Za graničnu vrijednost 0.75 algoritam je prosječno uklonio 251 podataka od dostupnih 2551. Za slučaj da su bilo koji od osam rezultata na udaljenosti manjoj od granične, brišu se svi rezultati za tu malu molekulu (njih osam). To je tako napravljeno za pristup klasifikacije s višestrukim oznakama jer su izlazni podaci usko

povezani. Rezultati modela nakon tog uklanjanja su u tablici 4.14.

Tablica 4.14 Rezultati modela klasifikacije s višestrukim oznakama nakon uklanjanja točaka bližih od 0.75

	F1	Točnost	Preciznost	Opoziv	AUC
Srednja vrijednost	0.414	0.804	0.366	0.518	0.662
Standardna devijacija	0.146	0.163	0.128	0.207	0.061

Za klasičan pristup binarne klasifikacije algoritam uklanjanja izvršava se za svaki mikroorganizam zasebno. Tablica 4.15 pokazuje prosječan broj obrisanih podataka za svaki mikroorganizam s graničnom vrijednosti od 0.75. Treba naglasiti da izvršavanje ovog algoritma zasebno nad osam različitih skupova podataka traje višestruko duže nego prethodna tehnika. Ovaj proces uklanjanja je na standardnom osobnom računalu trajao 28 minuta što uvelike otežava pretraživanje optimalne granične vrijednosti. Rezultati modela nakon tog uklanjanja su u tablici 4.16.

Tablica 4.15 Prosječan broj obrisanih podataka (bližih od 0.75) za svaki mikroorganizam

SA	EC	KP	PA	AB	CA	CN	HEK
82	11	3	2	22	95	101	121

Tablica 4.16 Rezultati modela binarne klasifikacije nakon uklanjanja točaka bližih od 0.75

	F1	Točnost	Preciznost	Opoziv	AUC
Srednja vrijednost	0.402	0.807	0.379	0.497	0.653
Standardna devijacija	0.135	0.159	0.102	0.222	0.056

Rezultati su malo gori od početnih što nije očekivano. Pretpostavka koju imamo kao uvjet da bi ova tehnika poboljšala rezultate je da si euklidski bliski podaci s različitim izlazima *smetaju*. Kao provjeru te pretpostavke možemo ukloniti sve bliske točke po navedenom algoritmu, ali nad cijelim skupom podataka uključujući one za treniranje, validiranje i testiranje. Ta izmjena svih podataka umjesto samo onih

za treniranje je loša te krši više dobrih praksi iz domene strojnog učenja. Zato se ovaj pristup neće koristiti kao optimizacija postojećeg modela već samo kao provjera spomenute pretpostavke. Rezultati opisanog pristupa su u tablici 4.17.

Tablica 4.17 Rezultati modela klasifikacije s višestrukim oznakama nakon uklanjanja obližnjih točaka

Prag	Broj uklanjanja	F1	Točnost	Preciznost	Opoziv	AUC
0.25	42	0.423	0.814	0.406	0.498	0.662
0.50	212	0.433	0.798	0.398	0.538	0.666
0.75	657	0.433	0.800	0.432	0.531	0.666
1.00	1356	0.410	0.784	0.407	0.507	0.643

Iako smo uočili mali porast rezultata zaključak je da ova tehnika ne može poboljšati rezultate. Naime, porast je minimalan i to tek nakon kršenja dobrih praksi iz domene strojnog učenja. Testirao sam više graničnih vrijednosti (pragova) nego što je tu navedeno, a sve zasebno bez i sa skaliranja podataka pomoću *StandardScaler*[5] programskog alata. Nikoji od pokušanih pristupa nije značajno povećao rezultate, a najbolji rezultati već su navedeni u tablici 4.17. Prethodni pokušaj uz praćenje svih dobrih praksi nije rezultirao poboljšanjem rezultata te su oni bili malo gori od početnih. Uklanjanje bliskih točaka zanemarivo utječe na rezultat (dokle god to uklanjanje ne utječe značajni na veličinu skupa podataka).

4.6 Treniranje više puta

Sljedeća ideja optimizacije rezultata modela jest trenirati model više puta. Nakon treniranja modela te računanja metrika nad podacima za treniranje rezultati nisu dobri odnosno model često griješi nad istim podacima nad kojima je treniran.

Ideja optimizacije je umjesto jednog treniranja napraviti neki broj n iteracija sljedećeg algoritma treniranja:

1. izvršiti treniranje modela nad podacima za treniranje
2. s tim modelom predvidi rezultate nad istim podacima za treniranje

3. izbriši sve točno predviđene rezultate, a ostavi one krivo predviđene
4. smanji stopu učenja za sljedeću iteraciju treniranja - podijeli stopu učenja s konstantom M
5. ako nema više podataka za treniranje (svi su izbrisani) onda prekini algoritam, inače izvrši sljedeću iteraciju

S ovim algoritmom svaka sljedeća iteracija trenira model samo nad podacima koje je model krivo predvidio u prethodnoj iteraciji algoritma.

Hiperparametri za podešavanje ovog algoritma su broj iteracija n te djelitelj stope učenja u svakoj iteraciji M. Broj iteracija bitan je zato što nakon nekog broja iteracija ovaj algoritam počne biti kontraproduktivan. Dijeljenje stope učenja brojem M omogućuje davanje veće važnosti ranijim iteracijama. Ranije iteracije nose veću važnost zbog toga što treniraju model nad većem i kompletnijem skupom podataka dok svaka sljedeća iteracija služi samo kao manje podešavanje modela za slučajeve netočnog predviđanja. Iz sličnog razloga granične vrijednosti za pretvaranje decimalnog broja predviđanja između 0 i 1 u binarnu vrijednost (0 ili 1) o postojanju biološke reakcije molekule s mikroorganizmom se računaju samo nakon prve iteracije (na isti način kao u referentnom rješenju bez ove optimizacije). Kasnija treniranja neće mijenjati graničnu vrijednost kako ju ne bi pogoršali.

Rezultati točnije broj preostalih netočnih predviđanja nakon prve iteracije su motivacija za ovu optimizaciju. Tablica 4.18 za svaku *K-Fold* kombinaciju modela klasifikacije s višestrukim oznakama ulaznih podataka pokazuje srednju vrijednost rezultata izračunatih u F1 metrici (zasebno za podatke za testiranje i podatke za treniranje), njihovu standardnu devijaciju te broj točnih i netočnih predviđanja na modelu za treniranje nakon prve iteracije.

Jedan od razloga jako malenog broja točnih i velikog broja netočnih predviđanja na modelu za treniranje nakon prve iteracije jest način određivanja koji podatak je točan, a koji netočan. Naime, zbog navedenog ograničenja povezanosti izlaznih podataka u klasifikaciji s višestrukim oznakama nije moguće brisati samo netočne izlaze. Svaki podatak sadrži osam izlaza za reakcije s odgovarajućih osam mikroorganizama te se taj podatak može nakon iteracije algoritma ostaviti ili izbrisati. Brišu se samo ona točna potpuno predviđanja što znači da model mora točno predvidjeti reakciju

Tablica 4.18 Rezultati nakon prve iteracije za svaku K-Fold *kombinaciju podataka modela klasifikacije s višestrukim oznakama*

K-Fold #	train F1	train std	test F1	test std	točno	netočno
1	0.603	0.063	0.486	0.081	721	1830
2	0.590	0.035	0.441	0.128	765	1786
3	0.603	0.032	0.405	0.143	757	1795
4	0.616	0.033	0.406	0.128	853	1699
5	0.575	0.100	0.375	0.150	821	1731

sa svih osam mikroorganizama. Samo jedno krivo predviđanje znači da će algoritam treniranja s n iteracija označiti podatak kao netočno predviđen te ga ostaviti za sljedeću iteraciju.

Treba naglasiti da za tablicu 4.18 vrijedi $n = 1$ odnosno nema dodatnih iteracija što znači da ona opisuje stanje osnovnog referentnog slučaja prije primjene optimizacija. Dakle, broj točnih predviđanja nad podacima za treniranje je uvijek značajno manji od broja netočnih predviđanja. To ponovno ističe težinu ovoga problema te manjkavosti odabranog rješenja. Rezultati izraženi F1 metrikom bolji su nad podacima za treniranje nego podacima za testiranje što je očekivano.

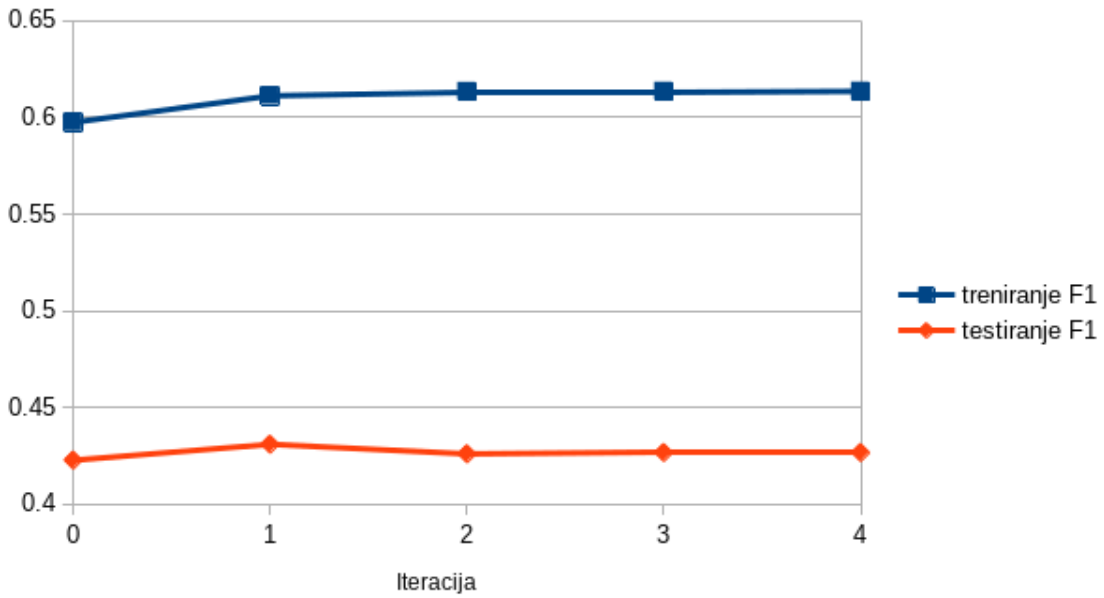
4.6.1 Traženje optimalnih postavki

Slika 4.12 pokazuje rezultate nakon svake iteracije treniranja nad podacima za treniranje i za testiranje izražene u F1 metrici. Rezultati sa slike računati su s pet iteracija ($n = 5$) s konstantom dijeljenja stope učenja nakon svake iteracije od $M = 5$ (početna stopa učenja, ona za prvu iteraciju, jednaka je referentnoj prije optimizacije).

Slika 4.13 pokazuje broj preostalih podataka nakon svake iteracije treniranja modela klasifikacije s višestrukim oznakama za isti slučaj kao slika 4.12. Vidljivo je da se nakon prve iteracije izbriše velik broj podatak (one koje je model točno "naučio" za vrijeme te iteracije). Međutim, svaka sljedeća iteracija ne pridonosi puno smanjenju podataka.

Isti se trend može vidjeti u F1 metrikama na spomenutoj slici 4.12. Prva iteracija vidljivo poveća model dok svaka sljedeća utječe neutralno ili čak negativno na daljnje

rezultate.



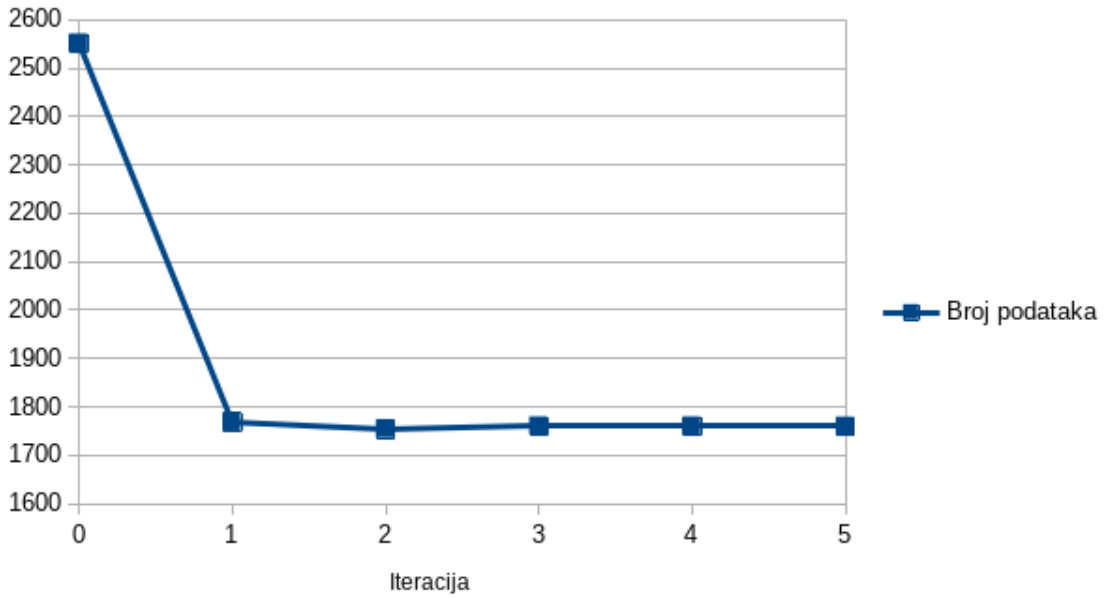
Slika 4.12 F1 rezultati nakon svake iteracije treniranja modela klasifikacije s višestrukim oznakama, $M = 5$

Iz ovih podataka možemo vidjeti da je najbolji rezultat nad podacima za testiranje ostvaren nakon druge iteracije. Metrike nad podacima za treniranje samo su informativni bez većeg značaja o kvaliteti modela.

Standardna devijacija F1 metrika kroz iteracije je uglavnom nepromijenjena te slična onoj iz referentnog rezultata. To znači da sljedeće iteracije konzistentno poboljšavaju, pogoršavaju ili ne mijenjaju rezultate.

Pogoršanje F1 rezultata nakon druge iteracije dešava se zbog premalene izabrane konstante dijeljenja stop učenja M. To potvrđuju rezultati za slučaj $M = 15$ gdje F1 metrika podataka za testiranje nema *koljeno* kao za $M = 5$. Rezultati F1 metrika za slučaj $M = 15$ kao i linija broja preostalih podataka nakon svake iteracije mogu se vidjeti na slikama 4.14 i 4.15. Sve ostale postavke i hiperparametri ostali su nepromijenjeni u usporedbi s prethodnim rezultatima ($M = 5$).

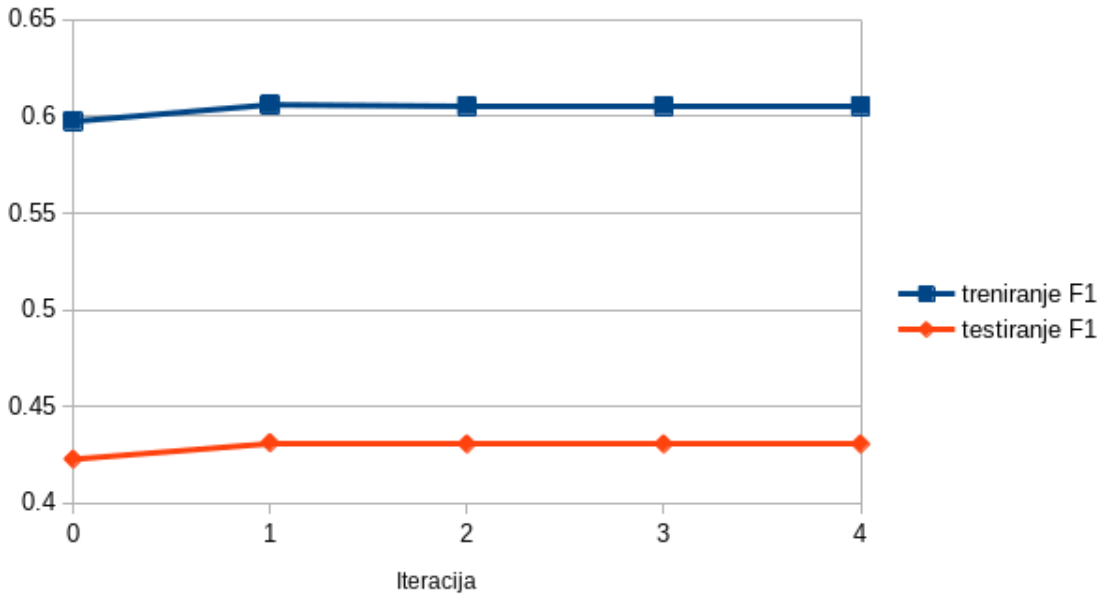
Treba naglasiti da sam u duhu eksperimentiranja pokušavao optimizirati ove re-



Slika 4.13 Broj preostalih podataka nakon svake iteracije treniranja modela klasifikacije s višestrukim oznakama, $M = 5$

zultate na više načina te su napisani samo najbolji rezultati. Neki pokušaji uključivali su: - veće konstante M - eksponencijalno smanjenje stope učenja tako da se broj M poveća nakon svake iteracije - mijenjanje funkcije smanjenja stope učenja (umjesto korištenja konstante M) - mijenjanje početne stope učenja i sl.

Svi ovi pokušaji rezultirali su gorim ili sličnim rezultatima te ih u svrhu jednostavnosti ne navodim u ovom radu. Mana ovoga pristupa je ručna priroda izmjene funkcije stope učenja (pomoću broja M ili na druge načine) od koje svaki pokušaj ima svoje nezanemarivo trajanje (model se mora iz početka trenirati i to veći broj puta zbog više iteracija). Zbog toga je moguće da sam prilikom eksperimentiranja preskočio neke bolje postavke koje bi donijele bolje rezultate. Međutim, zbog teškoće ovog problema i čestih neuspjeha u povećanju rezultata sumnjam da bi ti rezultati bili puno veći nego dobiveni.

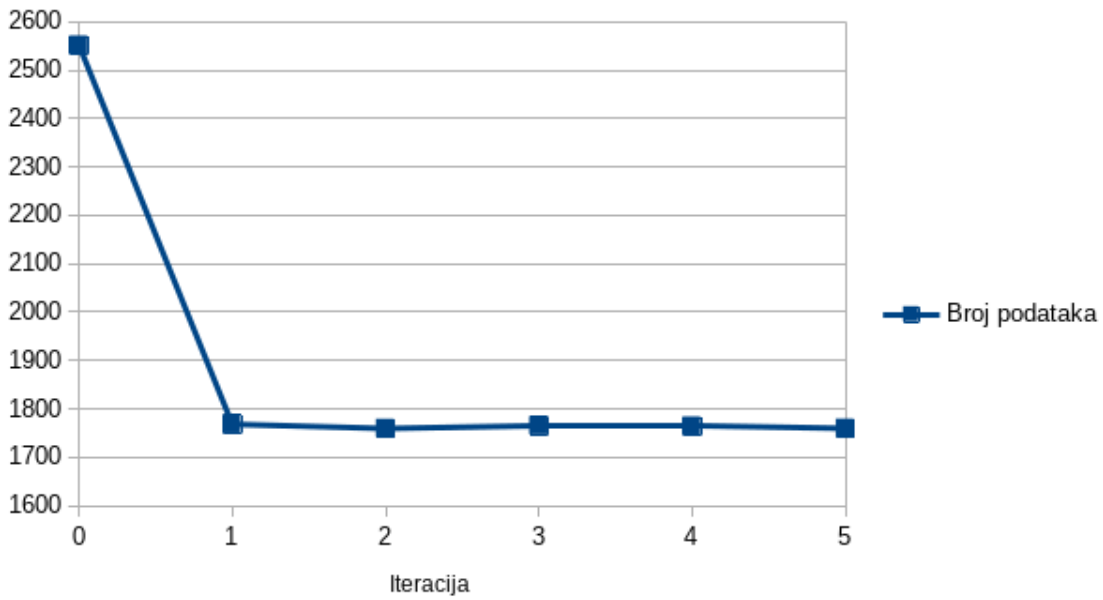


Slika 4.14 F1 rezultati nakon svake iteracije treniranja modela klasifikacije s višestrukim oznakama, $M = 15$

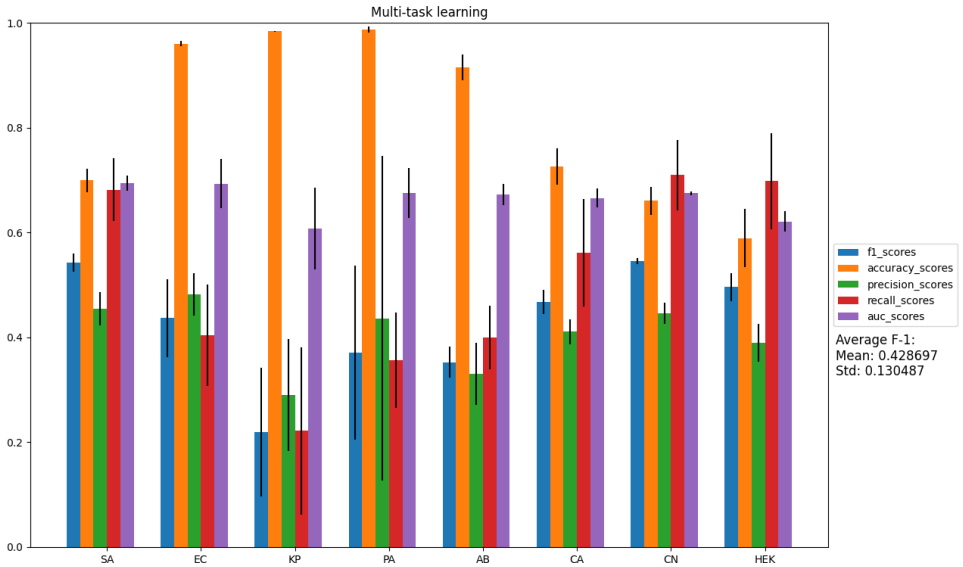
4.6.2 Najbolji rezultati

Iz prethodno predstavljenih rezultata možemo vidjeti da najbolje pronađene rezultate dobivamo nakon druge iteracije ($n = 2$) te s konstantnim faktorom smanjenja stope učenja $M = 15$. Točnije, model je u tom slučaju prvo trenira sa stopom učenja 0.0003, izbrišu se svi točno predviđeni podaci iz podataka za treniranje prema prethodno navedenom algoritmi iteracija te se nad preostalim podacima model još jednom trenira sa stopom učenja $0.0003/15 = 0.00002$.

Ukupni rezultat u F1 metrici za te postavke iznosi 0.428697 što je malo više nego referentni rezultat od 0.425708. Ostala raspodjela rezultata prikazana je na grafu 4.16. Iako se napredak od niti pola postotka čini malen njega nikako ne treba zanemariti zato što je ovo jedini uspješan napredak od svih probanih tehnika optimizacije rezultata.



Slika 4.15 Broj preostalih podataka nakon svake iteracije treniranja modela klasifikacije s višestrukim oznakama, $M = 15$

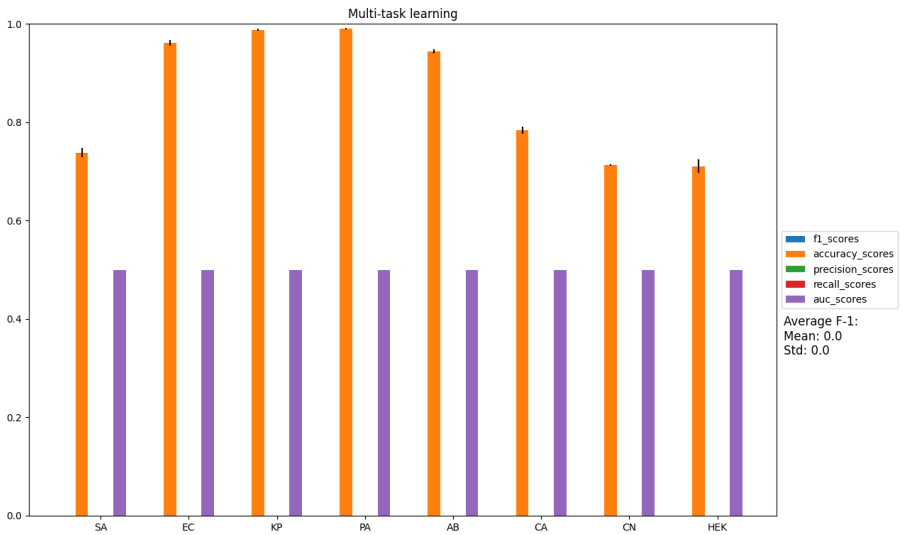


Slika 4.16 Rezultati za vrijednost varijabli ponovnog treniranja $n = 2$ *i* $M = 15$

4.7 Validacija metrika

Dani skup podataka je jako neujednačen, tj. ima puno više zapisa bez reakcije (0) nego s reakcijom (1). To znači da se može postići jako velika točnost ako model uvijek predviđa da nema reakcije. Iz tog razloga se razmatraju i metrike preciznosti i odaziva te se koristi F1 kao primarna metrika.

Zapaženo je da nakon treniranja model nikada ne odabire tu strategiju stalnog predviđanja da nema reakcija. Ručno sam zamijenio model onime koji uvijek predviđa nepostojanje reakcije (0) te izračunao metrike. Dobiveni rezultati se mogu vidjeti na slici 4.17.



Slika 4.17 Rezultati modela koji uvijek predviđa nepostojanje reakcije (0)

Samo metrike točnosti i površine ispod krivulje (AUC) nisu jednake nula. Točnost je dosta visoka s prosjekom od 0.853385 zbog neujednačene distribucije klasa unutar skupa podataka. AUC vrijednost je očekivanih 0.5 zato što je pola vrsta odgovora uvijek točno, a pola uvijek pogrešno predviđeno. Vrijednost AUC metrike jednaka 0.5 može značiti da model daje nasumične rezultate što znači da je AUC rezultat

ovog modela jednak nasumičnom modelu. Vrijednosti svih ostalih metrika jednaki su nula. Razlog tome je formula računanja pojedinih modela u kojoj se ne gledaju samo točna predviđanja kao u točnosti već i netočna predviđanja smanjuju ukupan rezultat.

Provedeni je isti postupak nad modelima koji imaju 1%, 5% te čak 10% nasumične vjerojatnosti predviđanja reakcije (oznake 1) kako ne bi svi podaci bili 0 kako bi se izbjegli potencijalni rubni slučajevi u računanju metrika. Rezultati tih pokušaja bili su jednaki prethodno prikazanom - vrijednost svih metrika osim dviju navedenih jednak je nula. To također ima smisla zato što je mala vjerojatnost da će se rezultat 1 nasumično odabrati baš za onaj podatak kojemu je to točan odgovor jer su ti podaci u velikoj manjini. Nadalje, rezultat 1 vjerojatnije će se nasumično odabrati za podatak kojemu je točan odgovor 0 te će time dodatno negativno utjecati na rezultat metrika.

Ovaj mali eksperiment nad metrikama je potvrdio da nije moguće zlouporabiti veći broj podataka bez postojanja reakcije kako bi se povećao rezultat kao što bi to bio slučaj da smo razmatrali samo metriku točnosti. To znači da model prilikom treniranja mora dolaziti do pravih zaključka te naučiti predviđati reakcije. Model na ovaj način ne može "prevariti sustav" te dobiti naizgled kvalitetne rezultate. Dodatna potvrda da model donosi neke zaključke u treniranju je vrijednost AUC metrike koja je veća od ovdje dobivenih 0.5 što znači da model ne daje nasumične odgovore. U nastavku je kao dodatnu provjeru istinitosti te tvrdnje validiran cijeli postupak od skupa podataka do dobivanja rezultata modela.

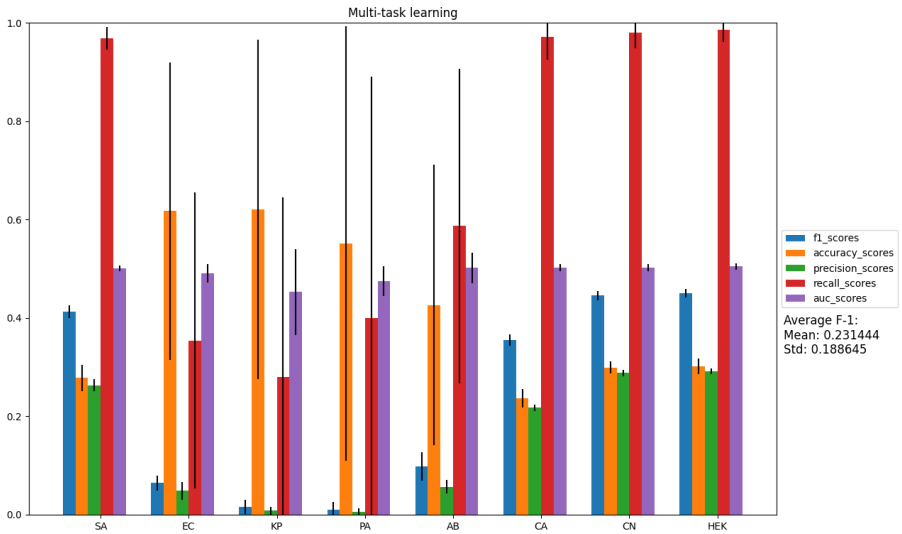
4.8 Validacija postupka

Izrada ovoga rada uključivala je više ručno pisanih algoritama za obrade podataka, metrika te povezanog programskog koda. Samo jedna manja pogreška često bi rezultirala krivom obradom podataka, tj. podaci bi bili praktički nasumični.

Kao provjeru smislenosti podataka možemo trenirati i testirati model nad nasumičnim podacima te vidjeti hoće li dobiveni rezultati biti slični. Nasumični rezultati također će provjeriti može li se išta zaključiti iz danih podataka ili su do sada dobi-

veni rezultati bili samo sreća. Te podatke nisam generirao potpuno nasumično već sam samo pomiješao redove izlaza odnosno pridružio svaki ulaz modela s nasumičnim izlazom.

Rezultati modela klasifikacije s višestrukim oznakama nad nasumičnim podacima vidljivi su na slici 4.18. Rezultati su puno gori od referentnih prema vrijednostima (srednja vrijednost F1 jednaka je 0.231444) te također prema konzistentnosti (standardna devijacija metrike F1 jest 0.188645).



Slika 4.18 Rezultati modela nakon miješanja podataka

Ovi rezultati pokazuju da referentno rješenje nije samo slučajnost nego da model zaista nauči neke poveznice između značajki malih molekula i njihovim reakcijama s mikroorganizmima. Osim boljih rezultata, smislenost podataka podržava i velika konzistentnost referentnog rješenja (niske standardne devijacije) za razliku od jako ne konzistentnih rezultata modela nad nasumičnim podacima.

Poglavlje 5

Rezultati najboljih modela

U ovom poglavlju navest ću najbolje dobivene rezultate za model klasifikacije s višestrukim oznakama te model binarne klasifikacije. Usporedba prednosti i mana ove dvije vrste modela već je spomenuta na više mjesta u ovome radu. U nastavku ću navesti podsjetnik na razlike koje smatram najbitnijima.

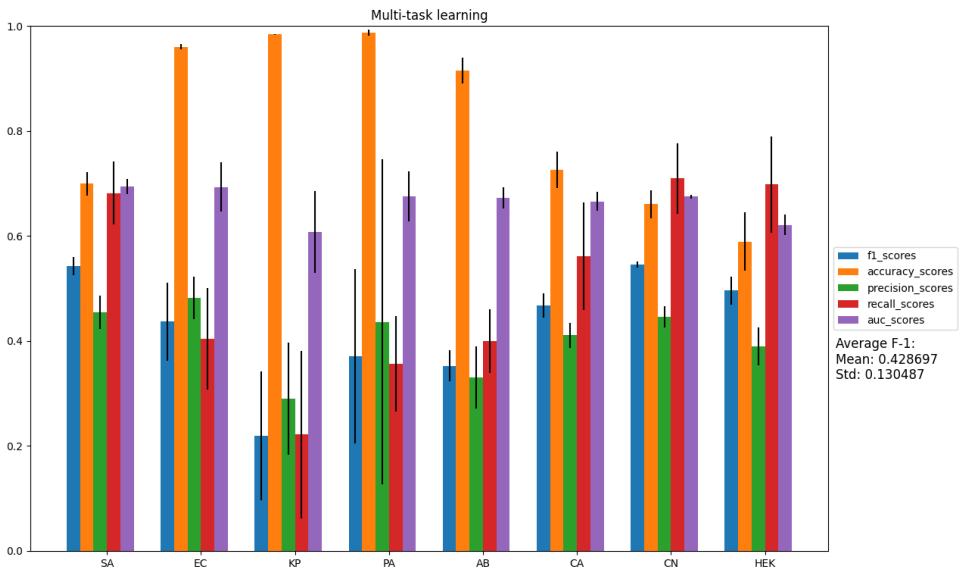
5.1 Klasifikacije s višestrukim oznakama

Najbolji pronađeni rezultat modela klasifikacije s višestrukim oznakama donosi onaj sa slojevima definiranima redom:

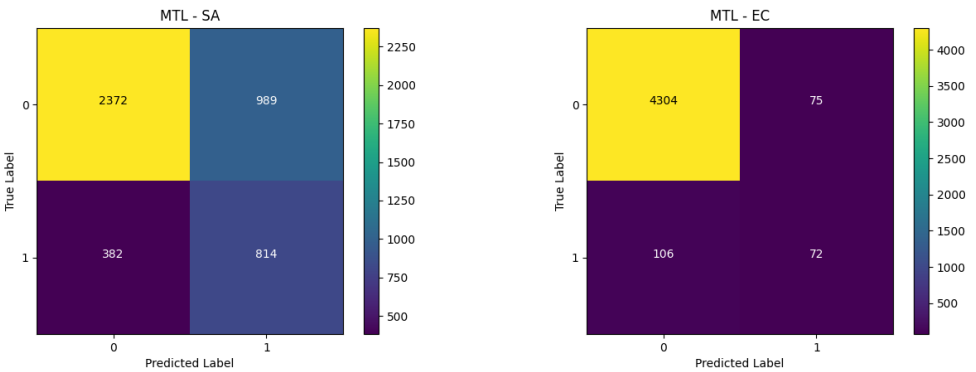
- 81 ulaznih čvorova - obavezno toliko jer to odgovara broju značajki nakon PCA
- gusti sloj sa 120 čvorova
- *dropout* sloj sa stopom poništavanja izlaza od 25%
- gusti sloj s 90 čvorova
- *dropout* sloj - 15%
- gusti sloj s 60 čvorova
- *dropout* sloj - 10%
- gusti sloj s 40 čvorova

- gusti sloj s 8 čvorova na izlazu

Osim tih parametra, za najbolje rezultate možemo koristiti optimizaciju ponovnog treniranja nad krivo predviđenim modelima objašnjenu u prethodnom poglavlju. Kao što je tamo spomenuto, najbolje pronađene rezultate dobivamo nakon druge iteracije ($n = 2$) te s konstantnim faktorom smanjenja stope učenja $M = 15$. Rezultati tog modela mogu se vidjeti na slici 5.1 gdje je navedena srednja vrijednost F1 metrike od 0.428697. Njegove matrice zabune mogu se vidjeti na slikama 5.2 i 5.3.

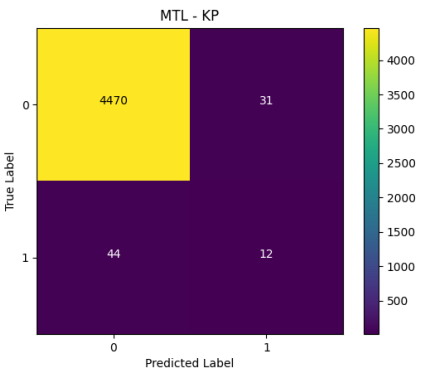


Slika 5.1 Najbolji rezultati modela klasifikacije s višestrukim oznakama

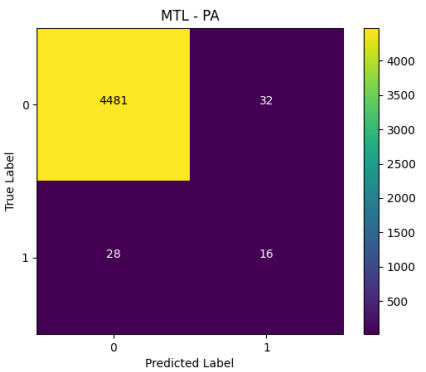


(a) SA

(b) EC

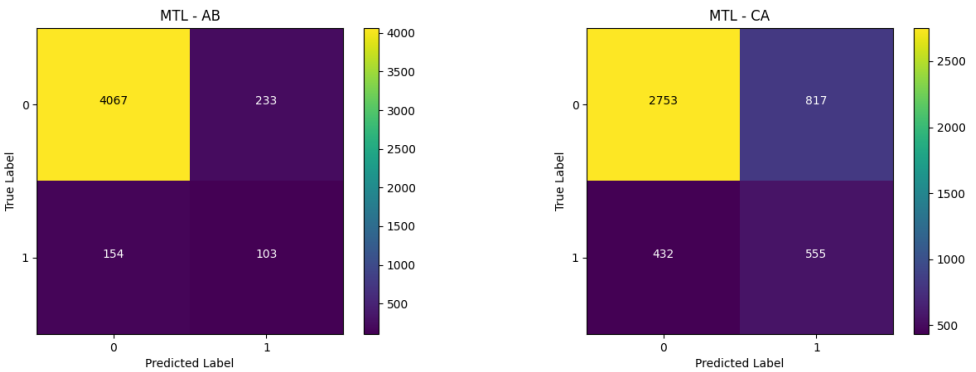


(c) KP



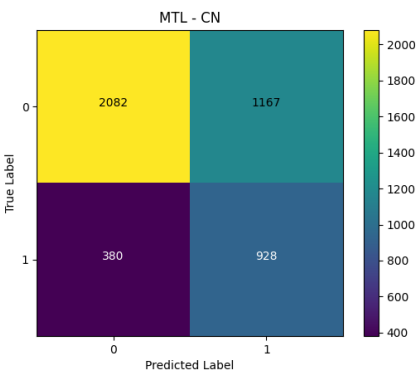
(d) PA

Slika 5.2 Matrice zabune najboljih rezultata klasifikacije s višestrukim oznakama - prvi dio

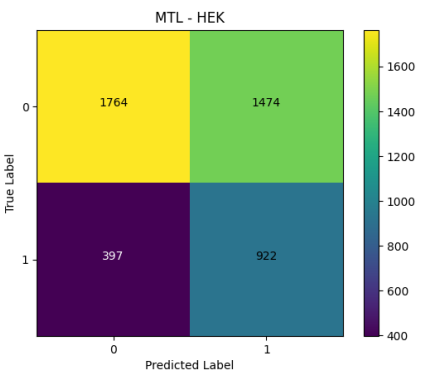


(a) AB

(b) CA



(c) CN



(d) HEK

Slika 5.3 Matrice zabune najboljih rezultata klasifikacije s višestrukim oznakama - drugi dio

5.2 Binarna klasifikacija

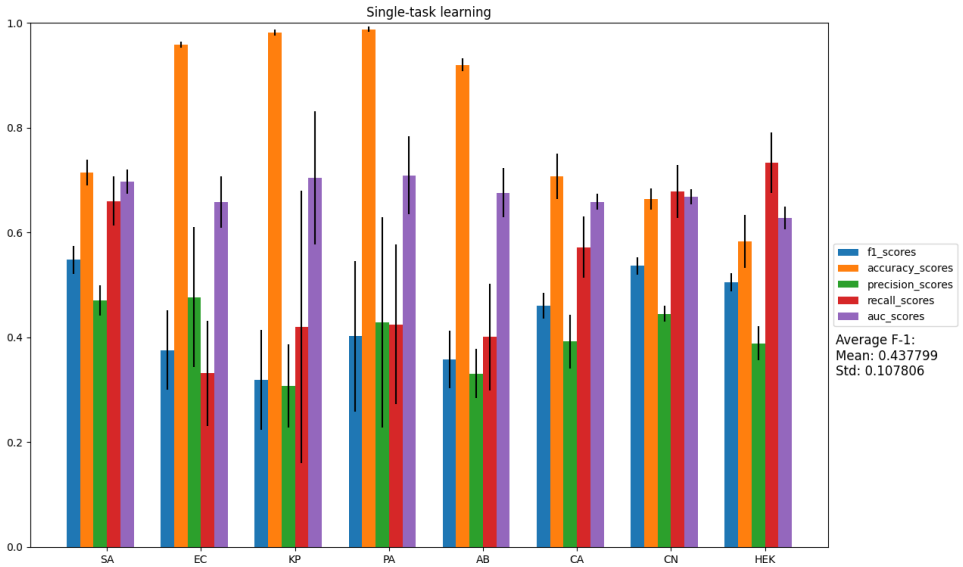
Za razliku od modela klasifikacije s višestrukim oznakama, za *klasičan* pristup potrebno je napraviti više različitih modela, po jedan za svaki mikroorganizam, te trenirati i testirati ih zasebno. To također znači da treniranje traje višestruko duže.

Najbolji pronađeni rezultat donosi model binarne klasifikacije sa slojevima definiranima redom:

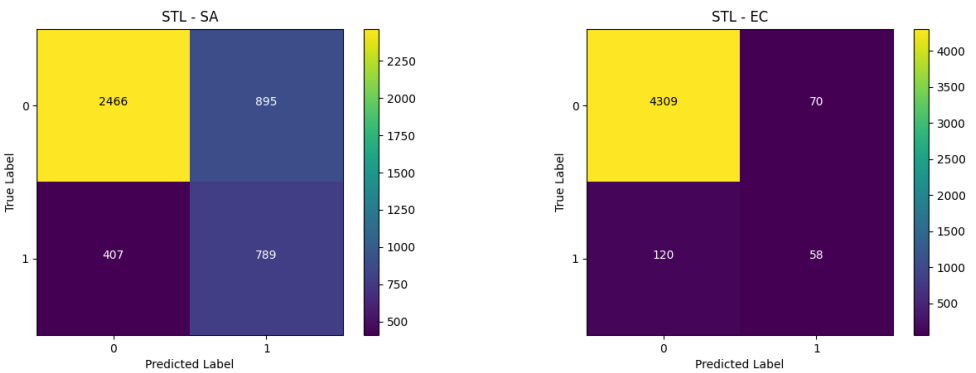
- 81 ulaznih čvorova - obavezno toliko jer to odgovara broju značajki nakon PCA
- gusti sloj sa 100 čvorova
- *dropout* sloj sa stopom poništavanja izlaza od 30%
- gusti sloj s 80 čvorova
- *dropout* sloj - 25%
- gusti sloj s 60 čvorova
- *dropout* sloj - 20%
- gusti sloj s 40 čvorova
- *dropout* sloj - 10%
- gusti sloj s 10 čvorova
- jedan izlazni čvor

Osim tih parametra, za najbolje rezultate korišten je i SMOTE (engl. *Synthetic Minority Oversampling Technique*) algoritam za augmentaciju podataka s već spomenutom konstantom $\alpha = 0.75$. Rezultati tog modela mogu se vidjeti na slici 5.4 gdje je navedena srednja vrijednost F1 metrike od 0.437799. Njegove matrice zabune mogu se vidjeti na slikama 5.5 i 5.6.

Treba napomenuti da je puno više vremena utrošeno u podešavanje parametara za model klasifikacije s višestrukim oznakama nego za model binarne klasifikacije te da se ovi rezultati vjerojatno mogu još malo optimizirati. Svejedno, puno veća poboljšanja rezultata nisu očekivana.

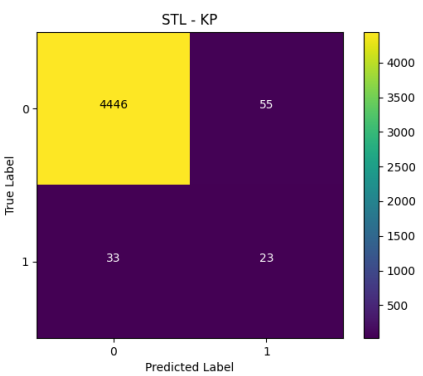


Slika 5.4 Najbolji rezultati modela klasifikacije jedne oznake

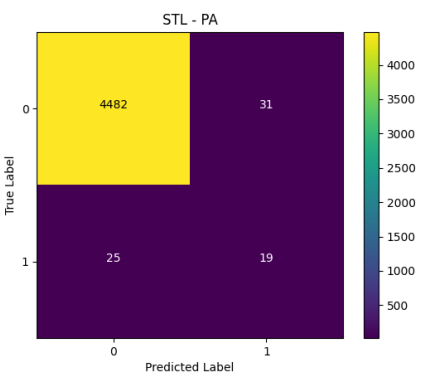


(a) SA

(b) EC

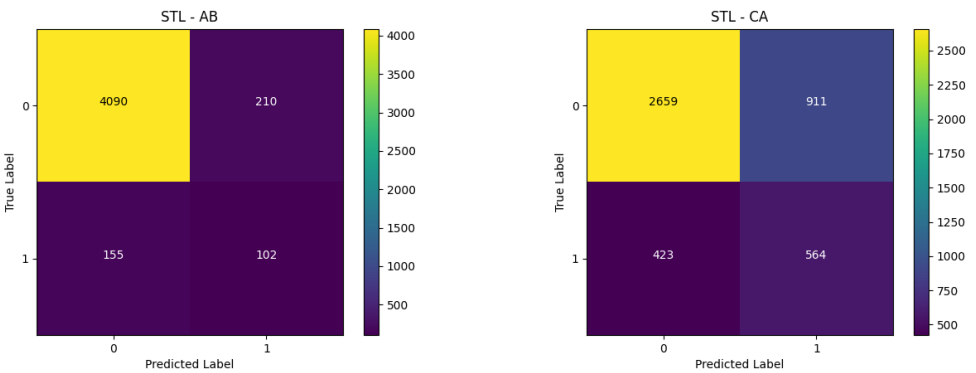


(c) KP



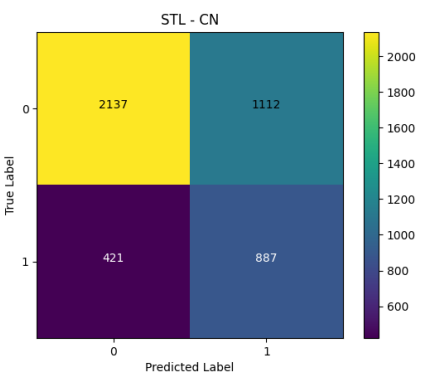
(d) PA

Slika 5.5 Matrice zabune najboljih rezultata klasifikacije jedne oznake - prvi dio

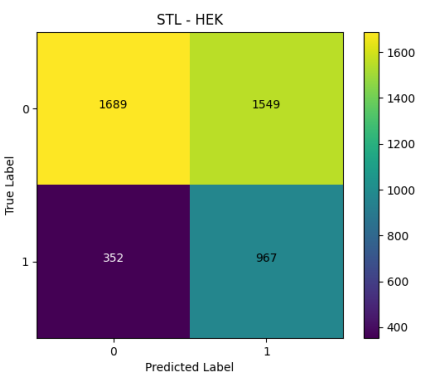


(a) AB

(b) CA



(c) CN



(d) HEK

Slika 5.6 Matrice zabune najboljih rezultata klasifikacije jedne oznake - drugi dio

5.3 Diskusija

Model klasifikacije višestrukih oznaka ima mnoga ograničenja od kojih su najosjetnija ona za predobradu (engl. *pre-processing*) podataka. Podaci su usko vezani te se ne može manipulirati njima. Točnije, ako je ulaz u model jedna molekula, a izlaz osam binarnih predviđanja o postojanju reakcije s molekulom, tada su spomenutih osam izlaza usko povezani te nije moguće brisati niti duplicirati jedan od njih bez brisanja i dupliciranja ostalih. To je bila velika prepreka u rješavanju neuravnoteženog skupa podataka za koji je najčešće rješenje dupliciranje nedovoljno zastupljenih podataka što iz spomenutih razlika nije bilo moguće.

Model binarne klasifikacije nema navedena ograničenja jer se za svaki izlaz model trenira potpuno nezavisno. Primjena spomenutih optimizacija na modelu binarne klasifikacije u ovom radu nije zavidno povećala rezultat. Međutim, tehnika povećanja broja željenih i smanjenja broja neželjenih podataka jako je korisna te bi u drugim radovima mogla nositi veliku važnost.

Uska povezanost izlaznih podataka modela klasifikacije višestrukih oznaka također znači da svaka molekula mora imati sve izlazne podatke iz čega slijedi da broj podataka o reakciji za svaki mikroorganizam mora biti jednak. U danom skupu podataka to nije bio slučaj te je bilo nužno zanemariti bilo koju molekulu koja nema podatak o reakciji s barem jednim mikroorganizmom. Pošto je takvih molekula bilo previše, rezultat je bio potpuno zanemarivanje dva mikroorganizma kako bi model imao dovoljno podataka za treniranje.

Model binarne klasifikacije mogao bi predviđati i ta dva izbačena mikroorganizma, iako s vjerojatno manjom točnosti zbog jednostavno manjeg broja podataka za treniranje. Rješenja ta dva mikroorganizma u ovom radu nisu istražena jer se ti podaci ne bi mogli usporediti s modela klasifikacije višestrukih oznaka koji ih ne podržava. Za daljnja istraživanja može se model klasifikacije višestrukih oznaka istrenirati nad podacima svih deset mikroorganizama te usporediti s rezultatima modela binarne klasifikacije.

S druge strane, glavna uočena mana modela klasifikacije jedne oznake jest višestruko sporiji postupak treniranja i testiranja zbog postojanja više različitih modela. Brže treniranje modela klasifikacije višestrukih oznaka omogućilo je veći broj poku-

šaja poboljšanja rezultata s jednakim brojem potrošenih resursa. Treniranje, testirane i formatiranje rezultata za model klasifikacije višestrukih oznaka traje 3 minute i 23 sekunde dok isti postupak za klasifikaciju jedne oznake traje čak 11 minuta i 22 sekunde što je 8 minuta, tj. 3.4 puta duže. Također, primjena istreniranog modela za predviđanje rezultata je višestruko brža i zahtjeva manje procesorske moći.

Uspoređujući rezultate dvaju pristupa možemo vidjeti da je model binarne klasifikacije malo bolji zbog većeg rezultata F1 metrike. Boljem rezultatu je najviše pridonijela augmentacija podataka odnosno SMOTE algoritam koji nije bio moguć za model klasifikacije višestrukih oznaka. To ne znači da je model binarne klasifikacije jednostavno bolji izbor jer treba uzeti u obzir i vremensku složenost treniranja kao i kasnijeg predviđanja. S klasifikacijom višestrukih oznaka bilo je puno lakše raditi zbog njene brzine, dokle god nisu bile potrebne veće manipulacije podataka koje ta tehnika ne podržava. Prije odabira jedne od ovih tehnika potrebna su daljnja istraživanja literature. Oba rješenja će se bitno drugačije ponašati ovisno o odabranom problemu i skupu podataka. Skup podataka u ovome radu veoma je težak za predviđanja te su različiti izlazi potencijalno nedovoljno povezani da bi uživali u prednostima klasifikacije višestrukih oznaka.

Poglavlje 6

Zaključak

Predviđanje reakcija malih molekula s mikroorganizmima moderan je i veoma složen problem. Iako bez donošenja većeg napretka u toj domeni, ovaj rad pokazao je brojne prepreke za razvoj pouzdanog rješenja. Osim toga, testirane su brojne tehnike poboljšanja rezultata od kojih su neke bile uspješne, a neke neuspješne.

Također je nenadano pronađena mana u ideji rješenja - više molekula s potpuno istim značajkama različito reagiraju na razne mikroorganizme što čini optimalno rješenje za predviđanje reakcija nemogućim bez dodatnih podataka o svakoj maloj molekuli.

Dobiveni rezultati potvrđuju već poznate probleme neuravnoteženih skupova podataka, pogotovo za strojna učenja bazirana na neuralnim mrežama. Tehnika sinteze uzoraka redundantnim otipkavanjem manjinske klase (SMOTE) pokazala se efektivnom u poboljšanju rezultata za skupove podataka s neujednačenim distribucijama klasa.

Tehnika klasifikacije s višestrukim oznakama pokazala je 3.4 puta brže ukupno izvršavanje, poslužiši pritom kao odlična optimizacija treniranja i predviđanja. Najveća mana te tehnike je uska povezanost izlaznih podataka zbog čega optimizacija SMOTE nije moguća. Iz navedenog razloga, vrijednost njezine F1 metrike iznosila je 0.429, što je nešto lošiji rezultat u usporedbi s 0.438 za tehniku binarne klasifikacije. Tehnika binarne klasifikacije je sporija, ali zato u ovom slučaju daje malo bolje rezultate. Iako u određenim situacijama tehnika klasifikacije s višestrukim oz-

nakama rezultira modelom boljih performansi, u ovom završnom radu rezultati su upućivali na suprotno što potvrđuje nužnost analize i usporedbe pojedinih tehnika na predmetnom problemu.

Literatura

[1] "Co-add baza podataka," s Interneta, https://db.co-add.org/downloads/, 15. veljače 2023.

[2] "programska knjižnica python rdkit," s Interneta, https://www.rdkit.org/, 4. kolovoza 2023.

[3] "programska knjižnica mordred," s Interneta, https://github.com/mordred-descriptor/mordred, 4. kolovoza 2023.

[4] L. Huang, X. Liu, Y. Liu, B. Lang, and D. Tao, "Centered weight normalization in accelerating training of deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[5] "sklearn.preprocessing.standardscaler documentation," s Interneta, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, 15. veljače 2023.

[6] H. Abdi and L. J. Williams, "Principal component analysis," *WIREs Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010. , s Interneta, https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101

[7] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philos Trans A Math Phys Eng Sci*, vol. 374, no. 2065, p. 20150202, Apr. 2016.

[8] "*sklearn.decomposition.pca*," s Interneta, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html, 6. rujna 2023.

[9] "How to calculate variance," s Interneta, https://www.scribbr.com/statistics/variance/, 6. rujna 2023.

[10] A. Parente and J. C. Sutherland, "Principal component analysis of turbulent combustion data: Data pre-processing and manifold sensitivity,"

Combustion and Flame, vol. 160, no. 2, pp. 340–350, 2013. , s Interneta, https://www.sciencedirect.com/science/article/pii/S0010218012002775

[11] "$sklearn.model_selection.kfold$," s Interneta, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html, 23. kolovoza 2023.

[12] "Mjerna točnost," s Interneta, https://tl.lzmk.hr/clanak/mjerna-tocnost, 5. kolovoza 2023.

[13] "Classification: Accuracy," s Interneta, https://developers.google.com/machine-learning/crash-course/classification/accuracy, 5. kolovoza 2023.

[14] "Classification: Precision and recall," s Interneta, https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall, 5. kolovoza 2023.

[15] "F-score," s Interneta, https://en.wikipedia.org/wiki/F-score, 5. kolovoza 2023.

[16] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," *BMC Med. Imaging*, vol. 15, no. 1, p. 29, Aug. 2015.

[17] "Classification: Roc curve and auc," s Interneta, https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc, 5. kolovoza 2023.

[18] "Keras dropout layer documentation," s Interneta, https://keras.io/api/layers/regularization_layers/dropout/, 15. veljače 2023.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, p. 321–357, jun 2002.

[20] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Int. Res.*, vol. 61, no. 1, p. 863–905, jan 2018.

[21] "`imblearn.over_sampling.smote` dokuemntacija," s Interneta, https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html, 16. lipnja 2023.

[22] "Dealing with class imbalance with smote," s Interneta, https://www.kaggle.com/code/theoviel/dealing-with-class-imbalance-with-smote, 16. lipnja 2023.

[23] "Parameters and hyperparameters in machine learning and deep learning," s Interneta, https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac, 23. lipnja 2023.

[24] "The tuner classes in kerastuner," s Interneta, https://keras.io/api/keras_tuner/tuners/, 23. lipnja 2023.

[25] "Randomsearch tuner," s Interneta, https://keras.io/api/keras_tuner/tuners/random/#randomsearch-class, 23. lipnja 2023.

[26] "Hyperband tuner," s Interneta, https://keras.io/api/keras_tuner/tuners/hyperband/#hyperband-class, 23. lipnja 2023.

[27] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018. , s Interneta, http://jmlr.org/papers/v18/16-558.html

[28] "Hyperparameter tuning with keras tuner," s Interneta, https://blog.tensorflow.org/2020/01/hyperparameter-tuning-with-keras-tuner.html, 17. ožujka 2023.

Pojmovnik

AB Acinetobacter baumannii. 3, 4, 15, 26, 42, 57, 61

AUC Area Under The ROC Curve. 14, 26

CA Candida albicans. 3, 4, 15, 25, 42, 57, 61

CN Cryptococcus neoformans. 3, 4, 15, 25, 42, 57, 61

CSV Comma-separated values. 2, 5

EC Escherichia coli. 3, 4, 15, 26, 42, 56, 60

FN false negative. 35, 38

FP false positive. 35, 38

HEK Embrionalne stanice bubrega čovjeka HEK 293. 3, 4, 9, 15, 20, 42, 57, 61

hRBC Human red blood cell. 3–5

KP Klebsiella pneumoniae. 3, 4, 15, 26, 42, 56, 60

MTL Multi-task learning. 5, 23

PA Pseudomonas aeruginosa soj ATCC 27853. 3, 4, 15, 20, 26, 42, 56, 60

PA5Δ Pseudomonas aeruginosa soj PAO397, PAO1. 3–5

PCA Principal component analysis. ix, 7, 8, 54, 58

ROC receiver operating characteristic curve. 14

SA Staphylococcus aureus. 3, 4, 15, 42, 56, 60

SMILES simplified molecular-input line-entry system. 2, 5, 6, 9, 10

SMOTE Synthetic Minority Oversampling Technique. 21, 58

TN true negative. 9, 35, 38

TP true positive. 9, 35, 38

Sažetak

Bitan dio moderne medicine je predviđanje reakcija raznih molekula i mikroorganizama. Ovaj rad istražuje rješenje predviđanja reakcije malih molekula i mikroorganizama iz baze otvorenog pristupa CO-ADD. Korištene su razne tehnike za poboljšanje rezultata: skaliranje i centriranje ulaznih podataka, brisanje više zastupljenih podataka, dupliciranje rijetkih slučajeva, pretvorba problema označavanja u problem klasifikacije, algoritamska pretraga najboljih hiperparametara modela, uklanjanje kontradiktornih podataka te treniranje više puta. Čak i nakon primjene svih ovih tehnika dobiveni rezultati su daleko od optimalnih. Korištena baza podataka u kombinaciji s knjižnicom Mordred za pretvaranje SMILES zapisa molekula u njihove značajke nema dovoljno podataka za kvalitetno rješenje ovog problema. Uočena je velika prednost u performansama tehnike klasifikacije s višestrukim oznakama u odnosu na češće korištenu tehniku binarne klasifikacije. Ta prednost nosi i nedostatke poput teže obrade podataka i malo lošijih rezultata. Više istraživanja potrebno je za kvalitetniju usporedbu ove dvije tehnike. Najkorisnija tehnika za predobradu podataka bila je *principal component analysis* (PCA) koja je smanjila broj ulaznih podataka i time omogućila rezultate sa srednjom vrijednosti F1 metrike jednakom 0.429 za klasifikaciju s višestrukim oznakama, odnosno 0.438 za binarnu klasifikaciju. Ovaj rad utvrđuje korisnost tehnike višestruke klasifikacije s ciljem manje potrošnje resursa, ali i pokazuje težinu problema predviđanja biološke reakcije malih molekula i mikroorganizama.

Ključne riječi — **strojno učenje, duboka neuralna mreža, klasifikacija s višestrukim oznakama, binarna klasifikacija, SMILES zapis, molekula, mikroorganizam**

Abstract

One of the major parts of modern medicine research is detecting reactions of various molecules and microorganisms. This paper explores a solution for predicting the reactions of small molecules and microorganisms from the open-access CO-ADD database. Various techniques for improving the results were used: scaling and centering

the input data, removing overrepresented data, duplicating rare cases, transforming the labeling problem into a classification problem, algorithmic search for the best model hyperparameters, removing contradictory data, and training multiple times. Even after applying all of these techniques, the obtained results are far from optimal. The data from the used database and the Mordred toolkit for converting SMILES representations of molecules into their features isn't sufficient for a high-quality solution to this problem. This paper shows a significant advantage in the performance of the multi-label classification technique compared to the more conventional binary classification technique. This advantage is followed by its drawbacks like difficulties with data processing and slightly poorer results. Further research is needed for a more robust comparison of these two techniques. The most useful data preprocessing technique was principal component analysis (PCA), which reduced the number of data inputs and thus enabled results with a median F1 score of 0.429 for multi-label classification, and 0.438 for binary classification. This paper confirms the usefulness of multi-task learning with the goal of lower resource usage while also demonstrating the complexity of predicting the biological reactions of small molecules and microorganisms.

Keywords **— machine learning, deep neural network, multi-task learning, binary classification, SMILES notation, molecule, microorganism**