Predviđanje aktivnosti peptida modelom strojnog učenja temeljenog na podskupu značajki iz formata SMILES

Negovetić, Mario

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski sveučilišni studij računarstva

Završni rad

Predviđanje aktivnosti peptida modelom strojnog učenja temeljenog na podskupu značajki iz formata SMILES

Rijeka, rujan 2022.

Mario Negovetić
0081156960

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski sveučilišni studij računarstva

Završni rad

Predviđanje aktivnosti peptida modelom strojnog učenja temeljenog na podskupu značajki iz formata SMILES

Mentor: doc. dr. sc. Goran Mauša

Rijeka, rujan 2022.

Mario Negovetić
0081156960

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 12. srpnja 2022.

Zavod: **Zavod za računarstvo**
Predmet: **Uvod u objektno orijentirano programiranje**
Grana: **2.09.04 umjetna inteligencija**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Mario Negovetić (0081156960)**
Studij: Preddiplomski sveučilišni studij računarstva

Zadatak: **Predviđanje aktivnosti peptida modelom strojnog učenja temeljenog na podskupu značajki iz formata SMILE / Machine learning-based peptide activity prediction using a subset of features from the SMILE format**

Opis zadatka:

Izraditi programsko rješenje za predviđanje peptidne aktivnosti temeljeno na strojnom učenju. Analizirati dostupne programske knjižnice kojima se u fazi pripreme podataka peptidni zapis pretvara u format SMILE te kojima se potom računa što veći niz značajki. U fazi predobrade podataka primijeniti postupke čišćenja podataka i odabira reprezentativnog podskupa značajki. Provesti usporedbu primjerene filtarske tehnike i tehnike omotača sa stanovišta računalnog vremena i performansi završnog modela predviđanja.

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

Mario Negovetić

Zadatak uručen pristupniku: 12. srpnja 2022.

Mentor:

Doc. Goran Mauša, dipl. ing.

Predsjednik povjerenstva za završni ispit:

Prof. dr. sc. Kristijan Lenac

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2022.

_____

Mario Negovetić

Zahvala

Zahvaljujem mentoru doc. dr. sc. Goranu Mauši na podršci tijekom pisanja ovoga rada, korisnim raspravama te prenesenom znanju iz umjetne inteligencije koje će mi pomoći u daljnjem školovanju. Također, zahvaljujem svojoj obitelji i prijateljima na podršci i razumijevanju tijekom studiranja.

Sadržaj

Popis slika **ix**

Popis tablica **xi**

1 Uvod **1**

2 Teorijska analiza peptida **3**

2.1 Opća definicija peptida 3

2.1.1 Katalitički peptidi 4

2.1.2 Antimikrobni peptidi 4

2.2 Vrste zapisa 5

2.2.1 Format SMILES 7

3 Predobrada podataka **9**

3.1 Čišćenje podataka 9

3.1.1 Vrijednosti izvan raspona 10

3.1.2 Obrada nedostajućih vrijednosti 11

3.2 Normalizacija podataka 12

3.3 Teorijska analiza tehnika odabira značajki 12

3.3.1 Pretraga značajki 14

3.4 Tehnika filtara 15

3.4.1 Statistička metoda Kendall 16
3.5 Tehnika omotača 18

4 Strojno učenje **20**
4.1 Definicija i svrha strojnog učenja 20
4.2 Nadzirano učenje 22
4.3 Nenadzirano učenje 22
4.4 Unakrsna validacija u fazi učenja 23
4.5 Metrike vrednovanja modela 24
4.5.1 Matrica zabune 25
4.5.2 ROC-AUC mjera 27
4.6 Algoritam Stablo odluke 27
4.7 Algoritam Slučajna šuma 29
4.8 Algoritam naivni Bayesov klasifikator 31

5 Studija slučaja **33**
5.1 Ulazni podaci 33
5.1.1 Analiza knjižnica za izračun značajki 34
5.1.2 Predobrada podataka 36
5.2 Metodologija 39
5.2.1 Algoritamska primjena filtar tehnike 40
5.2.2 Algoritamska primjena omotač tehnike 41
5.2.3 Implementacija strojnog učenja 43

6 Rezultati **45**
6.1 Usporedba tehnika odabira značajki 45
6.2 Usporedba performansi strojnog modela 52

7 Zaključak **67**

Literatura **69**

Pojmovnik **76**

Sažetak **77**

Popis slika

2.1 Usporedba prostornog i formata SMILES zapisa peptida 8

3.1 Opći prikaz funkcije za odabir značajki 15
3.2 Smjernice odabira statističkih metoda 16
3.3 Shematski prikaz tehnike omotač 19

4.1 Proces strojnog učenja 22
4.2 Unakrsna validacija strojnog modela 24
4.3 Matrica zabune u binarnoj klasifikaciji 25
4.4 Shematski prikaz algoritma Stablo odluke 28
4.5 Shematski prikaz algoritma Slučajna šuma 30

5.1 Primjer ulazne CSV datoteke prije izračuna značajki 34
5.2 Grafički prikaz mjere spljoštenosti i asimetrije 37
5.3 Shematski prikaz programskog projekta 39
5.4 Hodogram tehnike filtar 41
5.5 Dijagram slijeda forward tehnike 42
5.6 Dijagram slijeda backward tehnike 43

6.1 Forward tehnika kod pretrage katalitičkih peptida 47
6.2 Forward tehnika kod pretrage antimikrobnih peptida 48
6.3 Backward tehnika kod pretrage katalitičkih peptida 50

6.4 Backward tehnika kod pretrage antimikrobnih peptida 51
6.5 Matrica zabune strojnog modela kod kataličkih peptida 54
6.6 Grafički prikaz ROC-AUC vrijednosti strojnog modela kod katalitičkih peptida 55
6.7 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod katalitičkih peptida - 1. dio 57
6.7 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod katalitičkih peptida - 2. dio 58
6.8 Matrica zabune strojnog modela kod antimikrobnih peptida 61
6.9 Grafički prikaz ROC-AUC vrijednosti strojnog modela kod antimikrobnih peptida 62
6.10 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod katalitičkih peptida - 1. dio 64
6.10 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod antimikrobnih peptida - 2. dio 65

Popis tablica

2.1 Prikaz aminokiselina i njezinog skraćenog zapisa 6

5.1 Broj značajki koje su uklonjene ili transformirane u predobradi podataka 39

6.1 Mjerni rezultati strojnog učenja korištenjem katalitičkih peptida . . 53

6.2 Mjerni rezultati strojnog učenja korištenjem antimikrobnih peptida . 60

Poglavlje 1

Uvod

Ovaj završni rad izrađen je kao dio uspostavnog istraživačkog projekta Hrvatske zaklade za znanost „Dizajn katalitički aktivnih peptida i peptidnih nanostruktura" s oznakom UIP-2019-04-7999 i ERASMUS+ projekta „Promoting Sustainability as a Fundamental Driver in Software development Training and Education" s oznakom 2020-1-PT01-KA203-078646. Tema rada je predviđanje aktivnosti peptida modelom strojnog učenja temeljenog na podskupu značajki iz formata SMILES (eng. *Simplified Molecular Input Line Entry System*). Obrađuje se teorijska podloga peptida kao i njihova važnost u području bioinformatike. Koristeći dostupne knjižnice za programski jezik Python, omogućen je programski izračun velikog broja značajki koje su središnji dio analize u daljnjem radu. Nadalje, upotrebom modela strojnog učenja, pokušat će se dobiti model koji ima visoku preciznost u predviđanju aktivnosti katalitičkih i antimikrobnih peptida, sa stajališta izračunatih vrijednosti ROC-AUC (eng. *Receiver Operating Characteristic - Area Under Curve*). Dva osnovna postupka pomoću kojih se odabiru reprezentativni podskupovi značajki su filtarske tehnike (eng. *filter*) i tehnike omotača (eng. *wrapper*). Glavni je cilj istražiti koja od navedenih algoritamskih tehnika je brža te ima kvalitetnije predviđanje izlazne klasifikacije pošto su računalni resursi često ograničeni.

Napretkom znanosti i modernih tehnologija bioinformatika postaje područje sve većeg interesa mnogih znanstvenika. Samim time, peptidi i proteini, postali su izazovan predmet istraživanja zbog svoje molekulske građe kao i kemijskih reakcija koje iz-

azivaju. Peptide općenito smatramo polimerima koji nastaju povezivanjem kratkih lanaca aminokiselina. Njihova struktura utječe na sastav lijekova te se često primjenjuje s funkcijom aktivne supstance koja ubrzava kemijske reakcije i/ili pomaže u djelovanju drugih kemijskih tvari. Osim toga, imaju veliku ulogu u prehrambenoj industriji, kozmetičkoj industriji te u raznim područjima biološkog istraživanja poput regeneracije tkiva.

Laboratorijski pokusi često su financijski i vremenski iznimno skupi, zbog čega je predviđanje aktivnosti peptida pomoću računalnih programa danas izuzetno prihvatljiva, štoviše i preporučljiva metoda. Stoga, razvojem umjetne inteligencije i napretkom tehnologija poput strojnog učenja i dubokog učenja, danas postoje raznovrsne računarske tehnike koje se primjenjuju kao priprema laboratorijskih pokusa. Velike baze podataka kao i znanja o samim kemijskih elementima, u kombinaciji s računalnim znanjem pružaju široki raspon djelovanja. Stoga se u ovom radu primjenjuju samo neke od računalnih tehnika u predviđanju aktivnosti peptida.

Rad se sastoji od sedam poglavlja. Prvi dio je *Uvod*. U drugom poglavlju *Teorijska analiza peptida*, pojašnjeni su osnovni pojmovi o peptidima te njihov standardan zapis. U trećem poglavlju *Predobrada podataka* prikazuju se osnovni procesi čišćenja podataka koji su korišteni u ovom radu te postupci odabira značajki. Četvrti dio nosi naslov *Strojno učenje*, u kojem se analiziraju glavni algoritmi korišteni u procesu strojnog učenja, a zatim je objašnjen pojam strojnog učenja. Peti dio nosi naslov *Studija slučaja* gdje se detaljno objašnjava programsko rješenje ovoga rada. U šestom poglavlju *Rezultati*, radi se usporedna analiza različitih načina odabira značajki sa stajališta vremena i performansi modela predviđanja. Posljednji dio rada je *Zaključak*, u kojem se objedinjuje cijeli rad i donosi zaključak provedenog istraživanja.

Poglavlje 2

Teorijska analiza peptida

U ovom poglavlju objašnjena je teorijska podloga peptida sa stajališta fizikalnih, kemijskih svojstva te općenitog djelovanja. Također, pojašnjene su vrste računalnog zapisa kemijskih struktura koji se koriste u nastavku rada.

2.1 Opća definicija peptida

„Peptidi i proteini su biomolekule sastavljene iz prirodnih L-aminokiselina, te je njihova kemijska aktivnost u uskoj vezi zavisna o definiranoj trodimenzijskoj strukturi" [1]. Često se pojam peptida i proteina koristi kao istoznačnice, međutim peptidi nastaju prekidanjem dugih lanaca proteina kemijskim djelovanjem enzima. Peptidi su polimeri nastalim povezivanjem kratkih lancima aminokiselina, najčešće između dvije do sto, pri čemu je broj većinom proizvoljno odabran [2] odnosno postoji dogovorena granica. Bitno je napomenuti da neki autori za gornju granicu uzimaju pedeset aminokiselina. Danas službeno postoji dvadeset prirodnih aminokiselina, te su slične po svojoj fizikalnoj strukturi, a samim time i po kemijskom djelovanju. Sastoje se od atoma ugljika koji je povezan s atomom vodika, amino skupinom, karboksilnom skupinom i R skupinom bočnih lanaca [3].

Peptide općenito dijelimo u dvije grupe, a to su prirodni i umjetno laboratorijski proizvedeni peptidi koje još nazivamo i sintetski. Prirodni peptidi poput antimikrobnih

često su proučavani zbog svojih antivirusnih i imunosupresivnih svojstava, osobito u djelovanju na B i T bijele krvne stanice. Zbog ovih se vrijednih svojstava smatraju pionirima u obrani ljudskog organizma koji djeluju putem direktnog kontakta s patogenim uzročnicima [4]. S druge strane, sintetski peptidi danas omogućuju razvoj raznovrsnih lijekova u farmaceutskoj industriji te omogućuju efikasno djelovanje protiv novih bolesti. Osobito su popularna cjepiva temeljena na peptidnim svojstvima.

2.1.1 Katalitički peptidi

Katalitički peptidi su oligopeptidi koji različitom prostornom kombinacijom aminokiselina potiču katalitičke reakcije različite snage. Reakcije koje nastaju slične su enzimskim katalizama[1]. Međutim, problem predstavljaju teškoće u postizanju reakcija iste jakosti poput onih koje nastaju prirodnim enzimskim djelovanjem. Također, peptidi koji imaju svojstvo samosastavljanja omogućuju izazivanje katalitičkih aktivnosti, ali se mogu spajati s drugim molekulama. Upravo su zbog toga čest predmet istraživanja u bioinformatici [6]. Iz tog je razloga ovaj pristup istraživanja katalitičkih reakcija znatno vremenski i novčano isplativiji od rekreiranja umjetnih enzima.

Značajna je prednost primijećena kod peptida prilikom laboratorijskih pokusa u odnosu na umjetno napravljene enzime. Oligopeptidi su gradivni blokovi enzima te samim time omogućuju aktiviranje sličnih kemijskih reakcija. Naime, njihova veličina omogućuje tehnikom kombiniranja iz knjižnice peptida ispitati sve reakcije koje je moguće kreirati. Nadalje, još jedna od fizikalnih prednosti je mogućnost spajanja sa drugim peptidima tvoreći tako veće strukture koje bi u budućnosti mogle dovesti do jednakih katalitičkih reakcija kao i prirodni enzimi [6].

2.1.2 Antimikrobni peptidi

Antimikrobni peptidi (eng. *Antimicrobial peptides*, AMP) su peptidi topljivi u vodi i sastavljeni od malih lanaca aminokiselina. Prirodni su proizvod višestaničnih organizama i njihova primarna svrha je suzbijanje djelovanja različitih virusa, bakterija,

[1] „Enzimski katalizatori su biološki katalizatori koji obavljaju bitne uloge u ubrzavanju raznih bioloških reakcija te prijenosu energije unutar stanica kao i transkripciju i translaciju genetskih informacija" [5].

gljivica te tumorskih stanica [7, 8]. S obzirom da su AMP-i većinom pozitivno nabijeni, to im svojstvo omogućuje direktno djelovanje na membrane ciljanih mikroorganizama. Postoji više načina na koji djeluju, a jedan od učestalijih je prolazak kroz negativno nabijenu citoplazmatsku membranu nakon čega istiskuju lipide. Time se stvaraju stanične pore, oštećuje stanična membrana i omogućava se kemijsko djelovanje AMP-a na ciljanog uzročnika [9]. Također, posjedovanjem hidrofobnih aminokiselina, omogućava se direktno spajanje na negativno nabijene bakterije što dovodi do nesposobnosti vezanja na površinu stanica domaćina. Posljedica je nemogućnost replikacije RNA. Navedenom akcijom negativni uzročnici se suzbijaju u daljnjem prodoru u organizam.

Razumijevanje kemijske aktivnosti AMP, danas omogućava učinkovito razumijevanje načina djelovanja bakterija. Ljudsko je tijelo svakodnevno prirodno izloženo različitim patogenima, te se evolucijski AMP smjestio na svim mjestima prvog fizičkog kontakta, a to su su koža, pluća, oči, probavni trakt i oralna sluznica. Bitno je napomenuti da nemaju svi organizmi iste peptide, a najčešća grupa peptida koji djeluju u ljudskom tijelu su α-obrambeni peptidi. Svakako je poželjno da organizam ima što veću koncentraciju AMP-a iako njegova prevelika koncentracija bez svrsishodnog djelovanja na mikroorganizme može dovesti do razvoja autoimunih bolesti nauršavajući zdravlje domaćina. Jedan od primjera takve bolesti je psorijaza [8]. Trenutno nije zabilježena niti jedna bakterija koja ima rezistenciju na ljudske AMP-ove zbog opće strukture lipidnog dvosloja membrane. Stoga, ovi su peptidi od visokog interesa jer im se rezistencija na antibiotike povećala.

2.2 Vrste zapisa

Molekule se u biokemiji najčešće prikazuju u obliku prostorne atomske strukture. Takva vrsta zapisa pruža uvid u strukturu neke kemijske sekvence. Međutim, takav zapis nije pogodan za računalo, i nema praktičnu primjenu u bioinformatici. Stoga se za područje istraživanja peptida najčešće koriste biološki i kemijski jezici. Kemijski se kod koristi kod zapisa kojima se opisuju velike molekulske strukture sastavljene od ponavljajućih uzoraka [10]. U takvom zapisu razlikuje se kod od jednog slova

(eng. *single letter code*) i kod od više slova (eng. *multi-letter code*) [11], pri čemu odabir zapisa prvenstveno ovisi o namjeni istraživanja. U slučaju primjene koda s jednim slovom, atomske se sekvence zapisuju sa sekvencama aminokiselina čije se oznake nalaze u tablici 2.1. Takav zapis se naziva FASTA te je ono čest odabir zbog jednostavnosti. S druge strane, zapis s više slova koristi se kada se peptid želi prikazati pomoću oznaka za neproteinske ili neprirodne ostatke aminokiselina [11]. Biološki jezici, s druge strane, koriste se kada je cilj prikazati pojedinačne atome unutar molekulske strukture. Minkiewicz, Iwaniak i Darewicz [11] navode da su najčešći zapisi SMILES, InChI i InChIKey format. U daljem radu će se koristiti format FASTA i SMILES.

Tablica 2.1 Prikaz aminokiselina i njezinog skraćenog zapisa

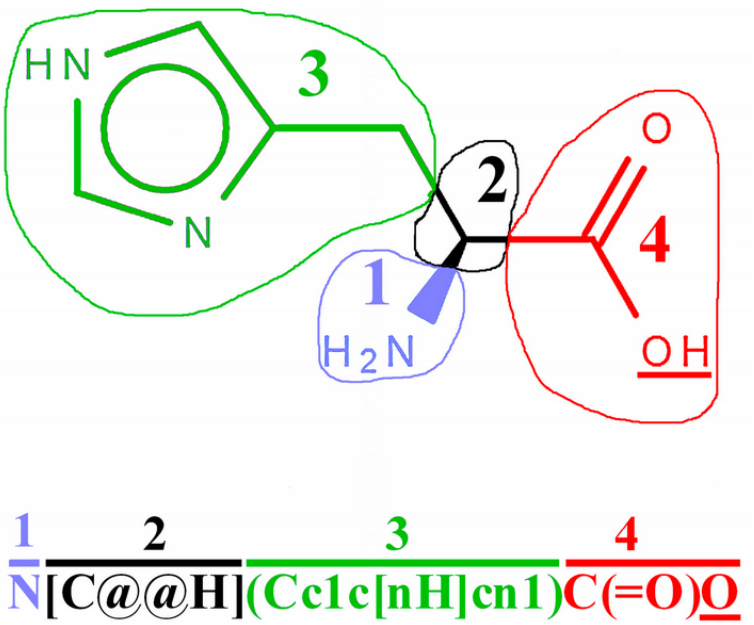
Izvor: [12]

Aminokiselina	**Zapisa s tri slova**	**Zapis jednim slovom**
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Histidine	His	H
Isoleucine	Ile	I
Glutamine	Gln	Q
Glutamate	Glu	E
Glycine	Gly	G
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

2.2.1 Format SMILES

Linijski sustav pojednostavljenog molekularnog unosa (eng. *Simplified Molecular Input Line Entry System*, SMILES) jedan je od najpopularnijih zapisa peptidnih struktura zbog svoje jednostavnosti i zadržavanja prostornog zapisa molekula. Karakterizira ga jednostavnost zapisa pri čemu je moguće iščitavanje odnosa među atomima. Nadalje, lako ga je prebaciti u druge formate korištenjem dostupnih računalnih programa, kao što je format FASTA. Na slici 2.1 prikazan je usporedni zapis peptida koji je zapisan u prostornom molekularnom obliku te ekvivalentnom formatu SMILES. Svi organski spojevi lako se mogu zapisati pomoću sljedeća četiri pravila [13, 14]:

- Organske atome navodi se jednim velikim slovom (B, C, N, 0, P, S , F, Cl, Br, I), dok se za anorganske atome mora navesti naboj i broj vodikovih atoma.
- Veze između atoma mogu biti jednostruke, dvostruke, četverostruke.
- Prostorno grananje atoma prikazuje se pomoću zagrada i nema ograničenja u prikazu dubina.
- Prstenovi se zapisuju tako da se razbijaju jednostruke veze pri čemu prvo slovo označava atom koji otvara prsten, a zadnje slovo atom koji zatvara prsten.



Slika 2.1 Usporedba prostornog i formata SMILES zapisa peptida

Izvor: [11]

Poglavlje 3

Predobrada podataka

U strojnom učenju, faza predobrade podataka jedna je od glavnih aktivnosti koja doprinosi točnijem donošenju odluka. Primjena algoritama na neobrađenim podacima često može uzrokovati netočne rezultate zbog različitih šumova unutar podataka. Glavni procesi koji se provode su: čišćenje, integracija, transformacija te smanjenje podataka [15]. U ovom poglavlju bit će prikazani neki od načina preobrade podataka s fokusom na tehnike za odabir podskupa značajki. S obzirom da postoje razne vrste zapisa peptida, samim time se dobivaju i različite značajke koje je potrebno obraditi kako bi strojno učenje bilo efikasno u smislu rezultata i vremenskog izvođenja.

3.1 Čišćenje podataka

Čišćenje podataka (eng. *data cleaning*) postupak je primjene različitih tehnika obrada podataka nad neuređenim, nepotpunim ili netočnim skupom podataka. Trenutno se smatra da je rudarenje podacima najbolja tehnika zbog fokusiranja na bitne informacije [16]. Osnovni je cilj reducirati sve šumove kako bi podaci mogli dobiti svoju svrhu i ispravno tumačenje. Stoga, da bi se neki podaci mogli proglasiti kvalitetnima je potrebno zadovoljiti sljedeća svojstva prema Kumaru, Khsola [17] : dosljednost, cjelovitost, točnost, relevantnost te postojanje. Ukoliko je potrebno integrirati podatke iz više različitih izvora, tada potrebe za zadovoljavanjem kvalitete višestruko rastu kako bi se izbjegli krivi zaključci zbog unosa dodatnog šuma. U

tim je slučajevima preporuka koristiti alate ETL (eng. *Extraction, Transformation, Loading*) kako bi se osigurao kvalitetan proces. Ipak, ponekad je potrebno provesti ručna ispravljanja zbog potrebnog znanja o domeni [18].

3.1.1 Vrijednosti izvan raspona

Jedno od učestalijih problema vrijednosti su izvan definirane domene zbog čega je potrebno rano provesti otkrivanje takvih vrijednosti (eng. *Outlier detection*) [16, 19]. S obzirom na problem koji se rješava, ponekad je dobro takve podatke u potpunosti izbrisati iz sustava ako su potencijalna prijetnja za krive zaključke. U suprotnom se slučaju mogu ostaviti i adekvatno prilagoditi. Tehnike otkrivanja se općenito dijele na statističke, tehnike modela i tehnike udaljenosti [19]. Svaka od ovih tehnika ima svoje prednosti i nedostatke pa je samim time nemoguće naći univerzalno rješenje.

Statistička metoda generira model na temelju poznatih treniranih podataka te pomoću statističkog testa određuje pripada li novi ulazni podatak u anomalije. Odabir statističkog testa koji će se provesti ovisit će prvenstveno primjenjuju li se parametarske tehnike ili neparametarske tehnike, pri čemu je razlika u korištenju funkcije gustoće to jest distribucije podataka. Parametarske tehnike su: *Gaussian Model, Regression Model, Mixture of Parametric Distributions*, a neparametarske tehnike su: *Histogram, Kernel Function* [20]. Prednosti ovih tehnika očituju se u velikoj brzini obrade podataka obzirom da se sva ispitivanja rade na jednom izgrađenom modelu. Ukoliko je distribucija podataka točna, tada se sa visokom sigurnošću može tvrditi da su dobiveni rezultati točni i opravdani. Valja primijetiti da je ova prednost ujedno i nedostatak. Ne preporuča se korištenje parametarske tehnike zbog potrebnog znanja o distribuciji podataka, što u stvarnosti nije uvijek dostupno. Također, određivanje granica u odnosu na distribuciju podataka često rezultira krivom odlukom o anomalijama. Dodatno, multivarijantni podaci u ovoj tehnici predstavljaju problem zbog velikih računskih zadataka pri testiranju hipoteza [21].

Sljedeći pristup ogleda se kod korištenja modela. Metode koje se ovdje često koriste pripadaju aktivnom učenju i dubokom učenju. Aktivno učenje je polunadzirano uče-

nje u kojem se prostor značajki vektorski dijeli na više prostora projekcije koji su definirani granicama odluke. Klasifikator uči na temelju treniranih podataka gdje u svakoj iteraciji postavlja novu funkcijsku hipotezu. Novi zapis bit će proglašen kao vrijednost izvan raspona ukoliko se nalazi izvan granica odluke. Proces se iterativno ponavlja i u njemu računalo zahtijeva dodatne informacije odnosno označene podatke od korisnika kako bi suzio krug generaliziranja [22]. S obzirom da je ovo i dalje novi pristup, češće se ipak koristi duboko učenje. Prednosti tog pristupa očituje se u boljem definiranju granica između vrijednosti izvan raspona i točnih vrijednosti, što je osobito uočljivo u velikom skupu podataka. Također, nema ograničenja u vrsti učenja što znači da ovisno o problemu podaci mogu biti označeni i neoznačeni [21].

Tehnike zasnovane na udaljenosti koriste neparametarski pristup za izračun udaljenosti između podataka. U pravilu ako n-ti podatak ima manje od k susjeda na definiranoj udaljenosti ili je prosječna udaljenost do k susjeda najveća onda ga se može nazvati vrijednost izvan raspona [23]. Tri najčešće metode koje se koriste su k-NN, metode rezanja i metode tokova podataka. Iako su u implementaciji jednostavne, u velikim skupovima podataka uzrokuju visoku računsku cijenu zbog neučinkovite skalabilnosti. Ovaj nedostatak eksponencijalno raste s povećanjem dimenzije podataka [21].

3.1.2 Obrada nedostajućih vrijednosti

Nedostajuće vrijednosti su čest slučaj prilikom prikupljanja podataka u stvarnom svijetu. Taj slučaj događa se zbog nepostojanja određenih informacija u datom trenutku ili automatskog izbacivanja vrijednosti tijekom pohrane zbog kršenja pravila u sustavima za upravljanje podacima. Takva situacija ne utječe bitno na donošenje budućih zaključaka ukoliko se radi od jednoj vrijednosti, no ako se radi o više takvih vrijednosti ili značajki onda to predstavlja veliki problem. Problemi se očituju u gubitcima informacija, poteškoćama koje se javljaju prilikom analize podatka kao i računanja odnosno nemogućnosti izračuna rezultata visoke preciznosti [24, 25].

Postoji više pristupa za rukovanje s nedostajućim vrijednostima. Najjednostavniji

način je ukloniti instancu odnosno zapis koji ima nedostajuće vrijednosti. Ovaj pristup zasigurno nije dobar zbog posljedičnog smanjenja baze podataka, odnosno broja zapisa čime se gube informacije koje su možda bile bitne. Nadalje, moguće je koristiti globalne konstante poput NULL, izbjegavajući time potencijalne probleme u računskim izrazima. Sljedeća mogućnost je koristiti statističke metode poput *mode, median, mean*. Operacije se koriste tako da se uzimaju vrijednosti od svih zapisa određenog atributa odnosno značajke te se primjenjuje odabrana metoda. Ovo su česti izbori kada se koriste diskretizirane numeričke vrijednosti. Međutim na taj se način vrijednosti mogu previše generalizirati. Zadnja je opcija korištenje algoritama poput metode najbližeg susjeda (eng. *Nearest Neighbor Method*, k-NN), linearne regresije, osnovne Kernel regresije te neuronske mreže. Zasigurno najkorišteniji je k-NN Imputer algoritam [15, 25] zbog svoje jednostavnosti i visoke efikasnosti. To je algoritam koji koristi sve dostupne podatke te nedostajuću vrijednost zamjenjuje sa srednjom vrijednosti k susjeda.

3.2 Normalizacija podataka

Podaci se rijetko nalaze u uniformnom obliku. U jednom skupu podataka najčešće postoje kontinuirane vrijednosti, diskretizirane vrijednosti te kategoričke vrijednosti. Uz to, podaci imaju i različiti raspon vrijednosti što stvara problem generalizacije. Stoga, kako se model strojnog učenja ne bi zbunjivao potrebno je provesti normalizaciju podataka i to nakon izbacivanja vrijednosti izvan raspona. Ovim postupkom se skup podataka ograničava u istu domenu u rasponu od nula do jedan ili od minus jedan do plus jedan. Metode koje se primjenjuju su *Min-Max*, *Decimal scaling*, *Norm*, *Z-Score*, *Mean-Mad*, i *Median-Mad* [26].

3.3 Teorijska analiza tehnika odabira značajki

Odabir značajki (eng. *Feature selection*) postupak je obrade podataka prije primjene algoritama za strojno učenje gdje se uklanjaju suvišne odnosno irelevantne informacije što posljedično smanjuje dimenzionalnost i kompleksnost skupa podataka.

Trenutno postoji puno tehnika za odabir značajki, ali sve se više pažnje posvećuje evolucijskom računarstvu zbog boljih optimizacijskih implementacija [27]. No, iako evolucijski algoritmi poput genetskog algoritma daju bolje rezultate nego druge tradicionalne tehnike, njihova uspješna implementacija zahtijeva visoku razinu znanja iz domene stohastike i programiranja [28]. Kada je riječ o optimizaciji, radi se o memorijskoj i vremenskoj učinkovitosti zbog efikasnijeg pretraživanja prostora značajki stoga je taj pristup česti odabir kao dodatak omotač tehnikama koje su računski zahtjevnije. Najgore je iscrpno pretraživanje čija je vremenska složenost $O(2^n)$.

U ovom kontekstu bitno je naglasiti razlike između relevantnih (bitnih), irelevantnih (nebitnih) te redundantnih (suvišnih) podataka. Uklanjanjem nebitnih značajki smanjuje se mogućnost zbunjivanja modela zbog unosa dodatnih informacija. Također, smanjenjem broja takvih podataka postiže se veća memorijska učinkovitost u pogledu korištenja RAM-a i brža izvedba algoritama [29]. Funkcijski to znači da dvije značajke međusobno mogu biti zamijenjene negacijom druge, a da pritom neće doći do promjene vjerojatnosti na izlazu funkcije [30]. S druge strane, redundantne značajke također uzrokuju zauzimanje dodatne memorije. Dvije značajke su redundantne ako jedna podrazumijeva drugu te se uklanjanjem odnosno dodavanjem jedne od njih, ne doprinosi poboljšavanju modela zbog doprinosa istih informacija [29].

Algoritme za odabir značajki svrstavamo u skupine u ovisnosti radi li se o zadacima rudarenja podataka, metode evaluacije podskupova ili metode generiranja podskupova [31]. Kada je riječ o tehnikama rudarenja podataka, temeljni pristupi koji se koriste u pretragama su nadzirano (eng. *Supervised*) i nenadzirano (eng. *Unsupervised*) učenje te njihov odabir ovisi o problemu koji se rješava. Nadzirani odabir značajki za vrednovanje podskupa koristit će tehnike poput Euklidske udaljenosti, heuristike ili statističke korelacije promatrajući odnos značajki i ciljane varijable. Ovaj se pristup koristi kod klasifikacijskih i regresijskih problema te je glavno obilježje da se znaju cilj i imena značajki. S druge strane ako su problemi vezani za grupiranje tada je bolje koristiti nenadzirano učenje. U ovom pristupu ne postoje informacije o ciljanoj varijabli, to jest radi se s neoznačenim podacima zbog čega se uzimaju sve instance odnosno zapisi za odabir relevantnog podskupa [32, 33, 34].

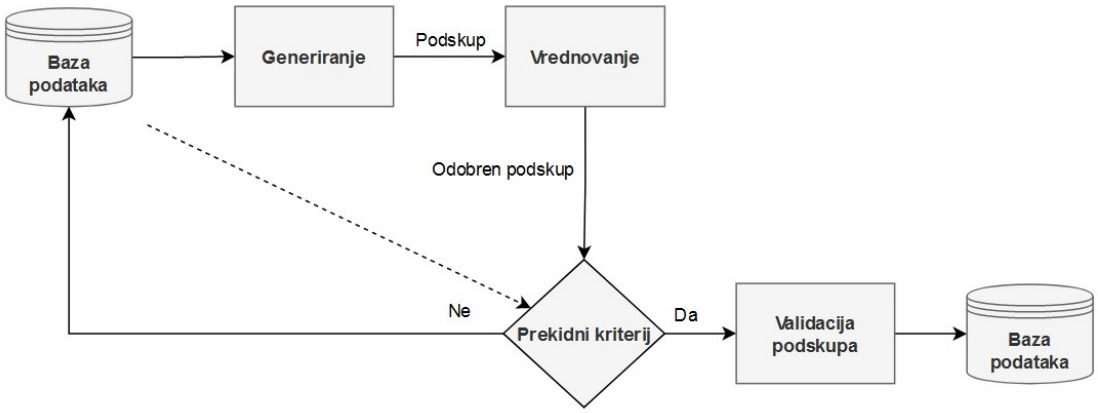
Algoritmi koji se koriste u problematici generiranja podskupova dijele se u: potpune pretrage, sekvencijalne pretrage, slučajne pretrage i integralno ponderiranje [31]. S druge strane algoritmi za evaluaciju podskupa odnosno odabir značajki svrstavaju se u sljedeće kategorije [27]:

- Omotač (eng. *Wrapper*) metode
- Filtar (eng. *Filter*) metode
- Ugradbene (eng. *Embedded*) metode[2]

3.3.1 Pretraga značajki

Svaka pretraga podskupa značajki započinje s prikupljenim podacima za koje su izračunate numeričke značajke. Iz ulaznih podataka se temeljem odabrane tehnike odabire podskup koji će se proučavati međutim, u nekim slučajevima se započinje sa praznim skupom podataka. Definirani smjer pretraživanja može biti: unaprijedno (eng. *forward search*), unazadno (eng. *backward search*) ili nasumično (eng. *random search*). Nakon odabira smjera definira se i strategija pretraživanja te se pokreće algoritam. U svakom koraku procjenjuje se koliko je kvalitetan podskup značajki koji je odabran. Metode koje se ovdje koriste su različite statističke metode, Euklidska udaljenost, informacijska mjerila ili algoritmi strojnog učenja. Ukoliko je odabrani podskup kvalitetniji od prijašnjeg podskupa onda se primjenjuje zamjena skupova podataka. Potom se ulazi u prekidni kriterij gdje se ispituje je li se dostigao određeni broj iteracija, broj značajki ili istražila cijela baza podataka odnosno pronašao najbolji mogući podskup. Ukoliko je uvjet zadovoljen izlazi se iz metode te je podskup moguće koristiti za daljnje strojno učenje [35, 32, 36]. Zadnji je korak vrednovanje kvalitete izabranog podskupa. Mjerila koja se koriste zavise o tome radi li se s označenim ili neoznačenim podacima. U poglavlju 4.5.1. opisane su klasifikacije metrike koje su korištene i u ovom radu. Cijeli navedeni proces prikazan je na slici 3.1.

[2]Ovu metodu naziva se još i hibridnom, a njezina funkcionalnost se temelji na objedinjavanju dobrih strana filtara i metoda omotača



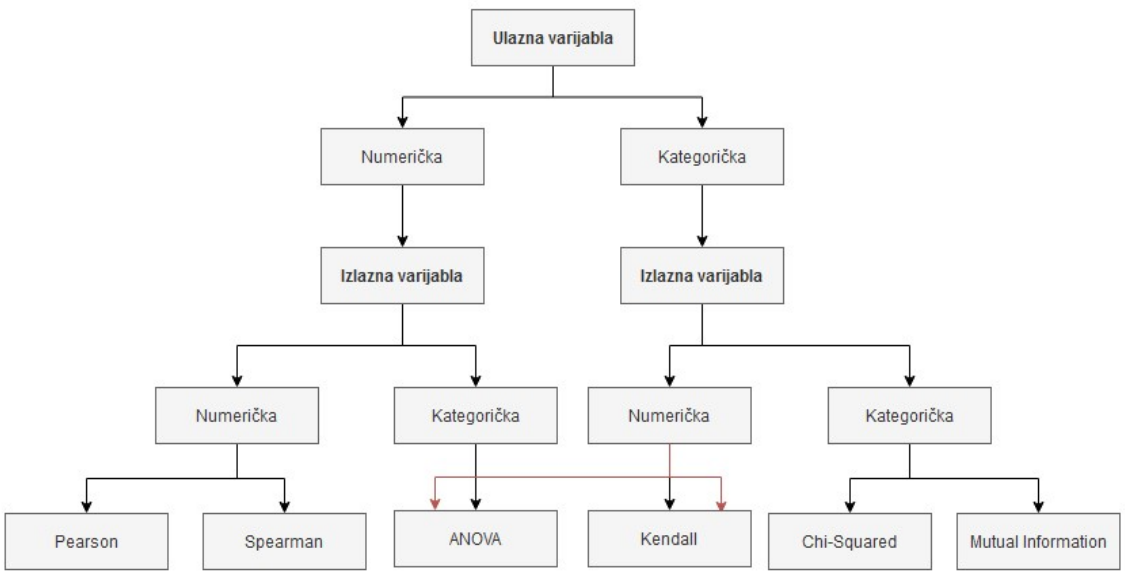
Slika 3.1 Opći prikaz funkcije za odabir značajki
Izvor: [35]

3.4 Tehnika filtara

Tehnika filtara jedna je od osnovnih tehnika odabira značajki u kojoj se relevantnost značajki ocjenjuje na temelju njihovih podataka pri čemu se ne gleda međusobna zavisnost značajki. Nakon vrednovanja svake značajke primjenjuje se rangiranje te se one s najnižom ocjenom uklanjaju iz skupa podataka. Bitno je napomenuti da se u tehnici ne primjenjuju algoritmi strojnog učenja, već se vrednovanje donosi pomoću različitih primjena statističkih metoda poput *V-rezultat, Fischerov rezultat, Relief, Hi-kvadrat, Minimalna-redundancija-Maksimalna-relevantnost* i *Dobitak informacija* [34, 37, 38].

Nadalje, razvojem ove tehnike kao i potreba za optimizacijom, razvili su se algoritmi za ponderiranje značajki i algoritmi za pretraživanje podskupa. Jedan od učestalijih algoritama koji se koristi ponderiranim težinama je *Relief* zbog dobrog pronalaženja relevantnih značajki [39]. On na temelju razine praga i Euklidske udaljenosti između zapisa *Near-hit* i *Near-miss*, otkriva koje su značajke s obzirom na ciljanu varijablu značajne. Svakako treba imati na umu da se primjenjuje u klasifikacijskim problemima s dvije klase te jednako kvalitetno radi s diskretiziranim i kontinuiranim brojevima. Međutim, primjenom heuristike ne uklanjaju se suvišne značajke

zbog čega mogu postojati situacije u kojima se dimenzije podataka ne smanjuju. U ovakvim situacijama je polinomna vremenska složenost prevelika cijena [40]. S druge strane, algoritmi pretraživanja podskupa strategiju temelje na pronalasku optimalnog podskupa koristeći se evaluacijskim mjerama poput statističkih testova, iscrpne i nasumične pretrage ili heuristikom. Odabir statističkih metoda primarno se donosi na temelju ulaznih odnosno izlaznih varijabli kao što je prikazano na slici 3.2. Za dvije odabrane značajke može se reći da su relevantne ukoliko samo između njih postoji visoka korelacija to jest, međusobna zavisnost pri čemu se najčešće se primjenjuje linearna korelacijska funkcija [39].



Slika 3.2 Smjernice odabira statističkih metoda

Izvor: [41]

3.4.1 Statistička metoda Kendall

Kendall τ je neparametarska statistička mjera korelacije između dvije slučajne promatrane varijable koja se koristi ukoliko ne postoji pravilna distribucija datih podataka [42, 43], iako to nije glavni uvjet za njezino korištenje. Ona je u izračunu slična Spermanovom koeficijentu korelacije, no Kendall daje funkcijsku vjerojatnost

uspoređujući nasumične parove dviju listi odnosno varijable. Gledano iz aspekta količine podataka, τ zasigurno daje bolju procjenu korelacije pri malim veličinama podataka te bolje procjenjuje odbacivanje *nulte hipoteze*. S obzirom da u svojoj definiciji koristi rangove unutar skupa podataka, efikasnije se koriste vrijednosti izvan raspona te nelinearni odnosi. To u pravilu znači da se promatra odnos kretanja dviju visoko koreliranih varijabli kroz monotonu zavisnost gdje ukoliko jedna varijabla raste, pita se hoće li druga varijabla također rasti ili padati po vrijednosti [44, 45].

Problem ove metode je taj što se svi parovi između dviju listi jednako promatraju bez korištenja težinskih faktora što je i jedan od temeljnih problema svih tehnika koje koriste rangove. Također, problem predstavlja i nezavisno statističko promatranje zamjene dviju listi što dovodi do krivih zaključaka [46].

Prilikom izračuna koristi se izraz 3.1 iz čijeg rezultata se dobije neparametarski koeficijent korelacije u rasponu između minus jedan i plus jedan [47]. Drugi način izračuna je korištenjem izraza 3.2 gdje se u odnos stavlja broj skladnih parova i broj neskladnih parova značajki [46]. Ova dva izraza računaju τ-a koji se najčešće računa pomoću programskih knjižnica. Postoji još τ-b i τ-c.

Ne postoji generalno definirano tumačenje τ vrijednosti, već se mogu tumačiti samo njezine krajnje vrijednosti. Ukoliko je vrijednost između dviju varijabli jednaka plus jedan, to znači da promatrane varijable poredaju skup podataka na isti način, što posljedično pokazuje da su visoko korelirane. S druge strane, ako je vrijednost minus jedan, to znači da varijable skup podataka poredaju na suprotan način. Postoji i *nulta hipoteza*, odnosno neovisnost dviju varijabli što je često predmet ispitivanja [48]. Mnogi autori uzimaju 0.9 kao referentnu vrijednost visoko koreliranih značajki.

$$\tau = 1 - \frac{4N}{n(n-1)} \tag{3.1}$$

gdje je:

- N - broj neskladnih parova značajki,

- n - broj različitih značajki.

$$\tau = \frac{c - d}{c + d} \tag{3.2}$$

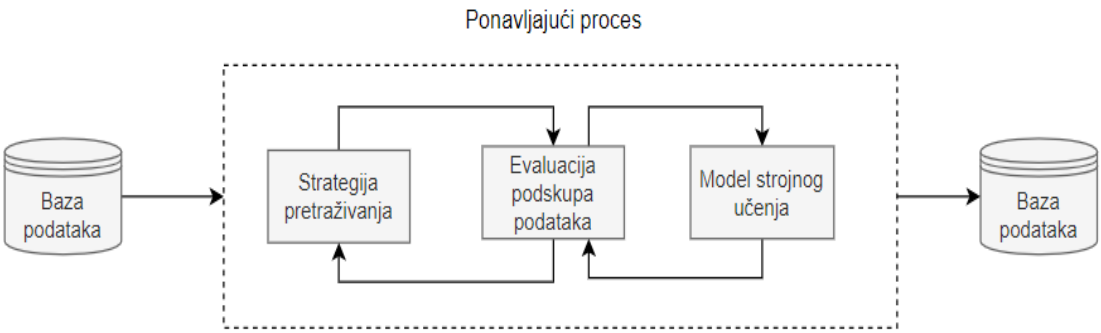
gdje je:

- c - broj skladnih parova,
- d - broj neskladnih parova.

3.5 Tehnika omotača

Tehnika omotača (eng. *Wrapper*) drugi je najčešći odabir kod redukcije značajki. Ono što ga razlikuje od filtar tehnike je to što se koristi tehnika strojnog učenja, odnosno predefinirani algoritam te uzima u obzir međusobnu povezanost podataka zbog čega će biti odabran najbolji mogući podskup [29]. Obzirom da ga karakterizira visoka računska cijena, treba biti pažljiv prilikom odabira algoritma jer mu vremenska složenost može biti i $O(n^2)$. Drugim riječima, veličina podataka uvelike utječe na računsku cijenu, ali i kvalitetu odabira značajki. Ako je skup podataka malen, tada se može dogoditi pretreniranost što nije poželjna situacija. U suprotnom slučaju, ako je skup podatak velik, tada je računska cijena visoka zbog unakrsne validacije koju je potrebno primijeniti nakon svake završene faze strojnog učenja [49].

Glavne strategije pretraživanja koje se koriste u tehnikama omotača su iterativne, sekvencijalne i inspirirane prirodom [50]. Njima se odabire koje značajke će se vrednovati u sljedećem koraku te ih razlikuje procesorska vremenska složenost. Osim toga, izabire se algoritam strojnog učenja pomoću kojeg se trenira strojni model kojem se po principu 'crne kutije' predaje cijeli ili prazan skup podataka. Ovisno o problemu koriste se pretrage unaprijedno (eng. *forward*), unazadno (eng. *backward*) pri čemu je unazadno računalno izuzetno skupa operacija ukoliko oba pristupa koriste parametarska ograničenja. Bez obzira koja se tehnika odabere, krajnji cilj je imati najbolji podskup dobiven strojnim vrednovanjem [51]. Cijeli proces prikazan je na slici 3.3 te detaljno opisan u poglavlju 5.2.2.



Slika 3.3 Shematski prikaz tehnike omotač

Sekvencijalni pristup koristi tehniku iscrpnog pretraživanja kod kojeg se započinje s praznim ili punim skupom podataka. U svakom koraku dodaje se ili odbacuje jedna značajka vrednujući je li se model poboljšao ili je ostao jednakih performansi. Takvim pristupom može se garantirati da će se pronaći najmanji podskup zbog čega je česti odabir kod pretraga značajki. Kod iterativnog pristupa koristi se filtar tehnika pomoću koje se izvrši ocjenjivanje svake značajke sa ciljanom klasom te ih se rangira. Potom se izvrši unaprijedni odabir značajki. Iako ovakvo pretraživanje uveliko smanjuje vremensku kompleksnost na O(n), i dalje je moguća pojava suvišnih značajki [52, 53]. U tehnikama inspiriranim prirodom koriste se evolucijski algoritmi koji se temelje na slučajnom pretraživanju [50].

Poglavlje 4

Strojno učenje

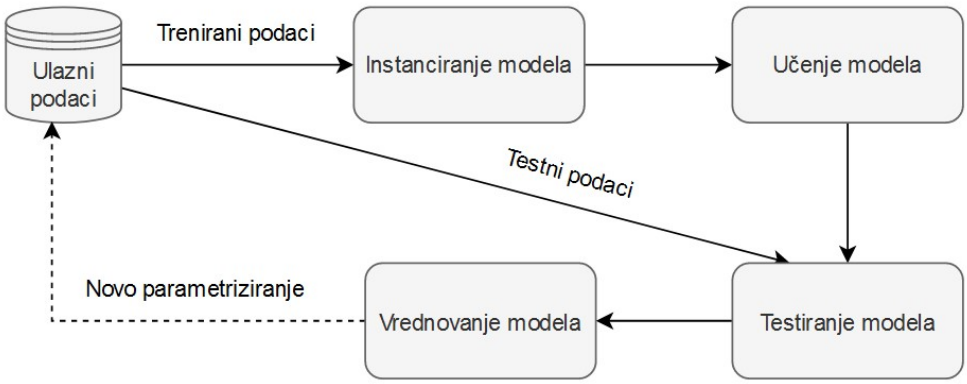
Razvojem tehnologije omogućeno je korištenje računala za rješavanje kompleksnih zadataka. Ponekad zadaci dostižu razinu kompleksnosti koju ljudi ne mogu riješiti ili zadaci imaju kratak vremenski rok za rješavanje. Upravo zato se pojavila potreba za razvijanjem dodatne grane umjetne inteligencije nazvane strojno učenje (eng. *Machine learning*). U ovom poglavlju bit će objašnjeni značenje i svrha strojnog učenja uz osvrt na metode vrednovanja modela to jest algoritma s fokusom na klasifikaciju. Potom će biti riječ o osnovnim algoritmima koji su korišteni u radu.

4.1 Definicija i svrha strojnog učenja

Mnogi autori znanstvenih knjiga i članaka imaju svoju definiciju strojnog učenja. U pravilu, sve definicije u općenitom obliku zagovaraju da je strojno učenje proces u kojem se stroj to jest računalo trenira da temeljem podataka i uzoraka u podacima nauči samostalno izvršavati zadatke. U širem smislu, strojno učenje je spona između umjetne inteligencije i dubokog učenja. Spajanjem ta tri područja omogućeno je korištenje robota u svakodnevnom životu. Ideja strojnog učenja je da imitiranjem ljudskog mozga i rudarenjem podataka računalo bude dovoljno sposobno samostalno donositi odluke koristeći prethodno stečeno znanje. Algoritmi koji se koriste u strojnom učenju temelje se na iterativnom izvršavanju te se u svakom koraku poboljšavaju izvođenje programa bez da se eksplicitno mijenja programski kod, a sve postavke i

generalizirani uzorci među podacima spremaju se u strojni model u obliku težinskih vrijednosti [54]. Strojni model se može koristiti po principu crne kutije koja se postepeno trenira temeljem iskustva. Osnovna svrha korištenja ovakvog pristupa je taj što se može provoditi neograničeno učenje to jest treniranje stroja naspram čovjeka te ga se u konačnici može koristiti za izvršavanje kompleksnih zadataka. Čovjek često nije sposoban izvršiti neke zadatke obzirom da razina za potrebnim znanjem svakim danom sve više raste. Samim time ne postoje granice za stroj, a on se može koristiti u raznim područjima od zahtjevnih medicinskih dijagnostika do rutinskih poslova [54].

Proces strojnog učenja započinje pretvaranjem ulaznih podataka razumljivim računalu. Podaci se obično dijele na podskup za treniranje i za testiranje kojim se provjeravaju performanse izgrađenog modela. Potom se definira određeni algoritam iz skupine nadziranog učenja, nenadziranog učenja, polunadziranog učenja, ojačanog učenja neuronskih mreža, metode ansambla, višezadaćno učenje ili učenje temeljeno na primjerima [55]. Bitno je napomenuti da određeni tipovi algoritama ne zahtijevaju treniranje pa im samim time nisu potrebni trenirani podaci. Nadalje, nakon što se definira algoritam u odnosu na problem koji se rješava, potrebno je instancirati model, koji je glavni dio strojnog učenja. Uzimajući u obzir algoritam, pokreće se proces učenja modela nad treniranim podacima. U svakoj iteraciji model će se vrednovati i bilježiti koliko je napredovao. Potom se model vrednuje nad testnim podacima gdje se definiranim mjerama dobiva koliko je dobro predviđanje. Ukoliko korisnik nije zadovoljan postignutim rezultatima, tada se koriste novi parametri za definiranje algoritma kako bi se dobili bolji rezultati. U suprotnom slučaju završava se proces strojnog učenja te se model može koristi. Ovaj proces prikazan je na slici 4.1.



Slika 4.1 Proces strojnog učenja

4.2 Nadzirano učenje

U klasifikacijskim problemima često se koriste algoritmi koji spadaju u domenu nadziranog učenja (eng. *Supervised learning*). Prije primjene algoritma potrebno je podatke razdvojiti na testne i trenirane. Trenirani podaci u postupku učenja će se koristiti kao uređeni parovi X,Y gdje X označava ulazne značajke na kojima se zasniva predviđanje vrijednosti Y. Glavna osobina svim algoritmima iz ove skupine je ta što model napravi predviđanje izlazne vrijednosti te uspoređuje izlaznu vrijednost s pravom vrijednosti koja je takozvana označena vrijednost. Na temelju te usporedbe model će učiti iz pogrešaka odnosno, krivih predviđanja [56]. Nedostatak ovog pristupa je u tome što mogu postojati podaci koji nisu označeni to jest neki ulazni podaci nemaju svoju izlaznu vrijednost. U tom slučaju treba ponoviti postupak označavanja koji se često provodi ručno i zahtjeva stručno znanje te samim time iziskuje veliku količinu vremena.

4.3 Nenadzirano učenje

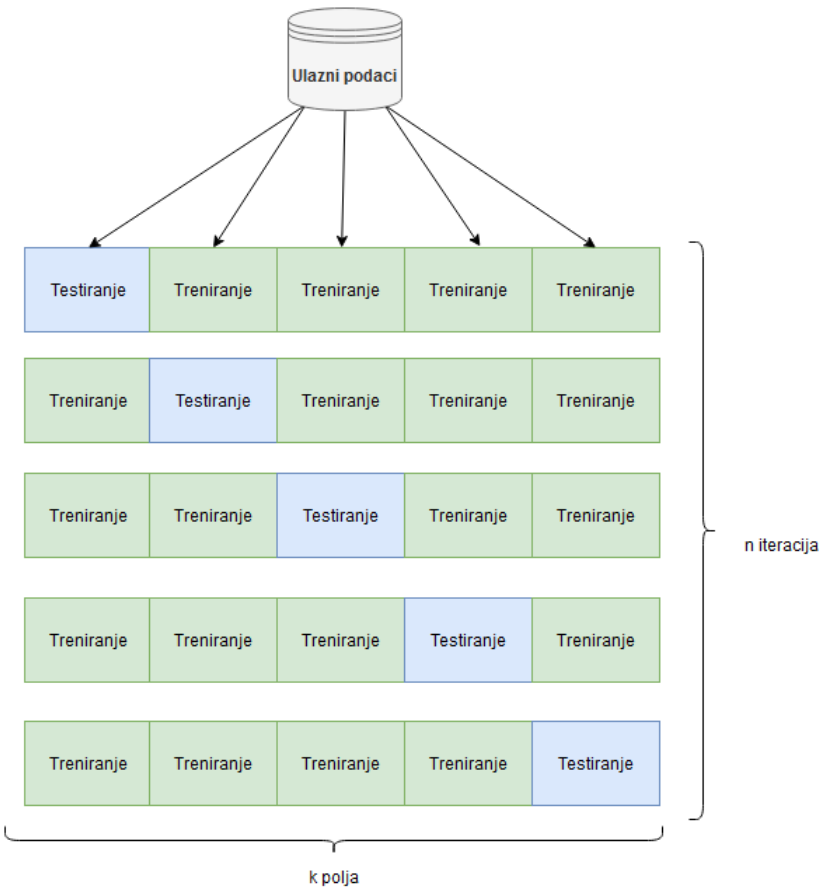
Nenadzirano učenje je vrsta strojnog učenja u kojem ulazni podaci nemaju pridruženu izlaznu vrijednost, to jest kaže se da su podaci neoznačeni. Algoritmi koji se nalaze u ovoj grupi funkcioniraju na principu traženja skrivene povezanosti među podacima promatrajući njihovu strukturu. Pritom samostalno odrede koja je izlazna

vrijednost grupirajući ih po sličnim obilježjima koje model sam prepozna. Na taj se način podaci generaliziraju.

4.4 Unakrsna validacija u fazi učenja

Potencijalna opasnost u modeliranju rješenja nekog problema je pretreniranost. To naizgled zvuči kontradiktorno jer je cilj naučiti model da što bolje prepoznaje uzorak problema. Međutim, moguće je da model radi ispravno predviđanje samo za podskup podataka koji je korišten u fazi treniranja te u trenutku kada dobije testne podatke dolazi do neispravnog predviđanja. Navedeni problem se naziva pretreniranost (eng. *overfit*), a postupak kojim se provjerava ako dolazi do pretreniranosti kao i izbjegavanje da do njega dođe je korištenje unakrsne validacija (eng. *Cross-validation*).

Unakrsna validacija u k-preklopa (eng. *K-fold cross-validation*) je postupak kod kojeg se iz cijelog skupa podataka generira testni i trenirani podskup podataka. Način na koji se dobiva je da se skup podataka od n zapisa nasumično podjeli u k jednakih grupa. Nakon podjele će se k-1 skup podataka koristiti za treniranje modela, dok će se preostali skup koristiti za testiranje modela. Karakteristika ovakvog pristupa je što se ovaj iterativni proces podjele izvršava k puta, i to najčešće deset. Odluka o tom parametru ovisi o veličini uzoraka koji će se koristiti. U slučaju malog skupa podataka, korist će se *Leave-One-Out* gdje je samo jedan uzorak testni podatak [57]. Općeniti prikaz postupka je na slici 4.2. Konačna validacija modela dobiva se kao srednja vrijednosti dobivenih mjerenja. Valja primijetiti da će se ovaj rezultat mijenjati zbog nasumičnog uzorkovanja podskupa podataka i na taj način će se prikazati koliko je model stabilan u svojim performansama. Međutim, unatoč ovom pravilu ne treba ponavljati cijeli postupak validacije jer se varijanca procjene modela postepeno smanjuje ukoliko se postupak nanovo pokrene [58]. Ukoliko se uspoređuju performanse više modela predviđanja, potrebno je postići jednaku podjelu podataka radi kvalitetnije usporedbe. To je moguće na način da se parametarski odrediti broj nasumičnog miješanja podataka, a time će se uvijek dobiti isti odabrani indeksi na izlazu.



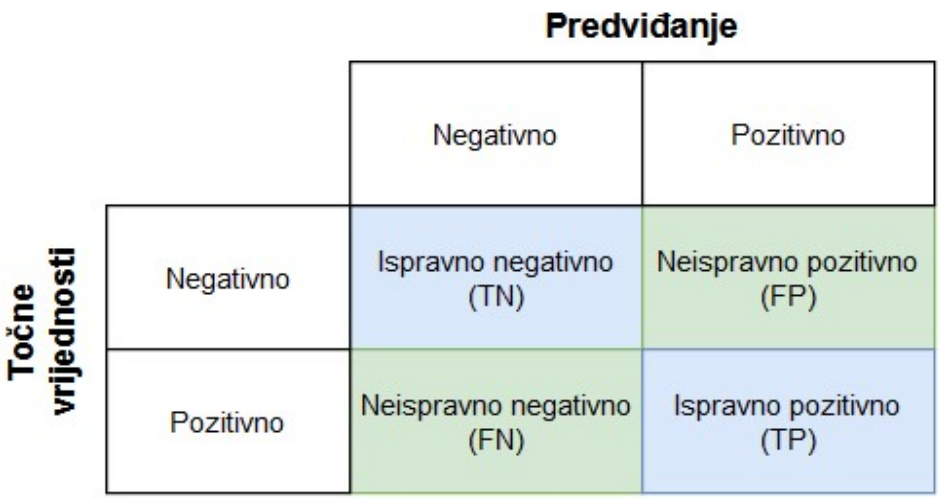
Slika 4.2 Unakrsna validacija strojnog modela

4.5 Metrike vrednovanja modela

U strojnom učenju bitno je nakon primjene algoritama te korištenja testnih i treniranih podataka primijeniti tehnike vrednovanja modela. Cilj vrednovanja je utvrditi koliko je predviđanje dobro odnosno loše te je li potrebno provesti ponovno treniranje modela. Mjere za klasifikaciju i regresiju se međusobno razlikuju zbog različitih predviđanja, no u nastavku su prikazane klasifikacijske mjere kao glavni fokus rada. Prema Hossin, Sulaiman [59] mjere koje se koriste su točnost (eng. *Accuracy*), stopa pogreške (eng. *Error rate*), osjetljivost (eng. *Sensitivity*), specifičnost (eng. *Specificity*), preciznost (eng. *Precision*), opoziv (eng. *Recall*), F-mjera (eng. *F-Measurel*), geometrijska sredina (eng. *Geometric mean*) te njihove prosječne vrijednosti.

4.5.1 Matrica zabune

Matrica zabune (eng. *Confusion matrix*) je statistički prikaz ispravnih i neispravnih predviđanja u odnosu na ciljanu varijablu. U binarnoj klasifikaciji, veličine je 2x2, no veličine mogu biti bilo kojeg raspona NxN pri čemu N označava broj klasa [60]. Vrijednosti se mogu podijeliti u stupce i retke matričnog polja u kojem se detaljnije dobije uvid koje klase su imale krivo predviđanje te koliki broj. Uobičajena binarna klasifikacija prikazana je na slici 4.3. Ispravno negativno (eng. *True Negative*, TN) označava broj točnih predviđanja negativnih klasa, neispravno pozitivno (eng. *False Positive*, FP) označava broj krivih predviđanja pozitivnih klasa, neispravno negativno (eng. *False Negative*, FN) označava broj krivih predviđanja negativnih klasa te ispravno pozitivno (eng. *True Positive*, TP) označava broj točnih predviđanja pozitivnih klasa [61]. Na temelju ove četiri vrijednosti moguće je izračunati ostale vrijednosti u vrednovanju modela.



Slika 4.3 Matrica zabune u binarnoj klasifikaciji

Točnost (eng. *Accuracy*, ACC) je jedna od najčešćih mjera koja se promatra u strojnom učenju. Temeljem izraza 4.1 procjenjuje se postotak koliko je točno model predvidio klasu temeljem ulaznih podataka u odnosu na cijeli skup podataka. Ovo često nije relevantna metrika ukoliko se radi o neuravnoteženom skupu podataka i

postoji prevelika dominacija jedne klase. To se posebice vidi u binarnoj klasifikaciji [62].

$$ACC = \frac{TN + TP}{TN + FP + FN + TP} \tag{4.1}$$

Preciznost (eng. *Precision*, Pr) prikazana izrazom 4.2 je sljedeća mjera koja se često koristi. Izračunom se dobiva postotak točno predviđenih klasa u skupu pozitivnih klasa.

$$Pr = \frac{TP}{FP + TP} \tag{4.2}$$

Opoziv (eng. *Recall*, TPR) odnosno osjetljivost je postotak točnih predviđanja prikazan izrazom 4.3. Teži se da ovaj postotak bude što veći jer je ključan za kvalitetan model.

$$TPR = \frac{TP}{TP + FN} \tag{4.3}$$

F1 mjera je harmonijska srednja vrijednost prikazana izrazom 4.4. Predstavlja odnos opoziva i preciznosti i cilj je da rezultat bude što bliže jedan, a idealno jedan.

$$F1 = 2 * \frac{TPR * Pr}{TPR + Pr} \tag{4.4}$$

Srednja geometrijska vrijednost (eng. *Geometric mean value*, G-mean) je mjera koja se dobiva kao drugi korijen umnoška opoziva i specifičnost prikazana izrazom 4.5. Osjetljiva je na distribuciju podataka stoga je dobar indikator ako postoji previše zapisa iz samo jedne klase što može rezultirati većinskim predviđanjem te klase [63].

$$G - mean = \sqrt{TPR * S} \tag{4.5}$$

gdje je:

- TPR - Opoziv,
- S - Specifičnost.

4.5.2 ROC-AUC mjera

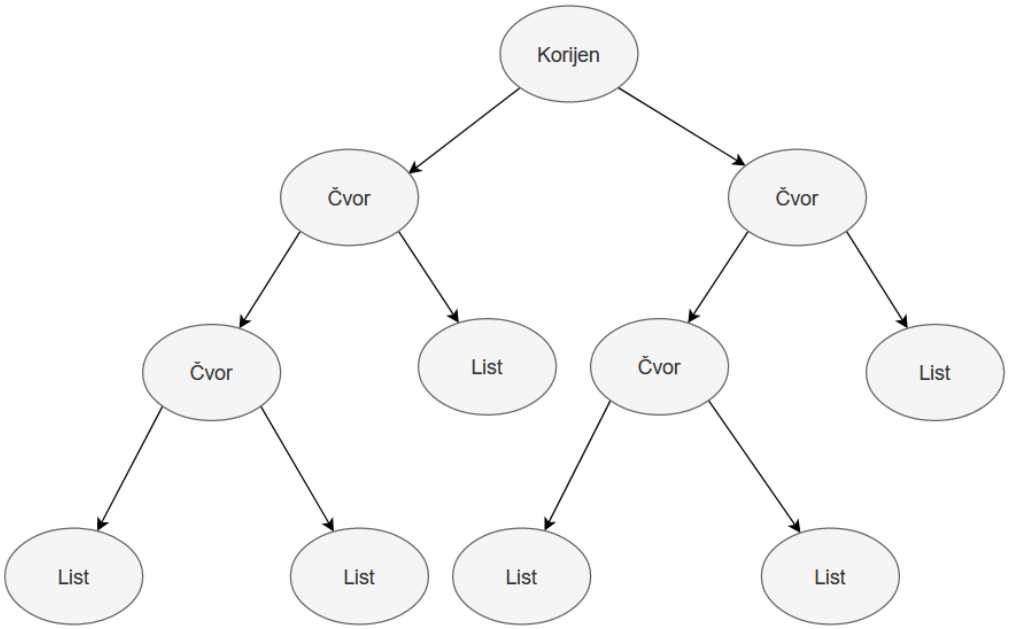
Radna karakteristika prijemnika (eng. *Receiver Operating Characteristic*, ROC) je grafički prikaz odnosa ispravno pozitivnih stopa i neispravno pozitivnih stopa. Vrijednosti se unutar grafa prikazuju u rasponu vrijednosti između nula i jedan po x i y osi pri čemu je model bolji ukoliko se krivulja nalazi na lijevoj strani u odnosu na glavnu dijagonalu odnosno u točki (TPR=1, FPR=0) [64]. Ta činjenica govori da model dobro klasificira ulazne podatke odnosno prepoznaje razliku između nula ili jedan. Ovim prikazom se procjenjuje koliko je dobar algoritam, odnosno model.

Često se uz grafički prikaz ROC krivulje računa površina ispod krivulje (eng. *Area under curve*, AUC). Prikazuje općenitiju mjeru vjerojatnost predviđanja pozitivne klase pri čemu model ne bi smio imati vjerojatnost manju od 0,5 koliko iznosi AUC nasumičnog pogađanja zbog površinog izračuna i odnosa pozitivnih i negativnih klasifikacija [65]. Raspon vrijednosti se kreće između nula i jedan te se u računskom postupku za svaku vrijednost ROC dobije numerička aproksimacija integrala koristeći trapezoidno pravilo za svaku vrijednost odaziva na x osi i preciznosti na y osi [66]. U nastavku rada ova vrijednost je označena oznakom ROC-AUC.

4.6 Algoritam Stablo odluke

Algoritam strojnog učenja Stablo odluke (eng. *Decision Tree*) jedan je od najčešćih izbora kod rješavanja klasifikacijskih problema u nadziranom učenju. Njime se uspješno rješava klasifikacijske probleme gdje određuje kojoj kategoriji odnosno klasi pripada ulazna instanca, a može biti upotrijebljen i za regresiju. Pritom ulazni podaci mogu biti diskretizirane vrijednosti, kontinuirane vrijednosti, a može postojati i određeni udio nedostajućih vrijednosti [67]. Također, karakterizira ga hijerarhijska stablasta struktura koja se sastoji od korijena (eng. *root*), čvorova (eng. *node*) u kojima se donosi odluka o grananju te listova (eng. *leaf*) u kojima se donosi odluka predviđanja. Cijela ova struktura, prikazana na slici 4.4 vrlo je jednostavna za razumijevanje zbog imitiranja programske uvjetne sintakse *if-then* [68].

Izgradnja stabla rekurzivno počinje od korijenskog čvora u kojem se nalaze ulazni podaci. Potom se podaci dijele na dodatna podstabla na način da se traži čvor u kojem je mogući informacijski dobitak određene značajke najveći na globalnoj razini cijele strukture. Taj proces se ponavlja za svaku značajku te se svaka iteracija zasebno pamti. Ukoliko se dogodi da u nekom čvoru postoji velika dominacija jedne klase, tada se ovaj postupak ne provodi te se taj čvor automatski proglašava listom [69].



Slika 4.4 Shematski prikaz algoritma Stablo odluke

Vremenska složenost ovog algoritma je logaritamska, međutim povećavanjem veličine podataka računska cijena se uvelike povećava, stoga je potrebno algoritam optimizirati različitim tehnikama i parametrima. Kad je riječ o parametrima, potrebno je definirati prekidni kriterij u vidu dubine stabla, broja zapisa u listu, broja zapisa potrebnih za grananja, broja odabranih značajki te kriterija grananja. Ako se parametri ne definiraju, stablo će se rekurzivno granati dokle god ne dođe do kriterija da u jednom čvoru bude dominantna jedna klasa [69].

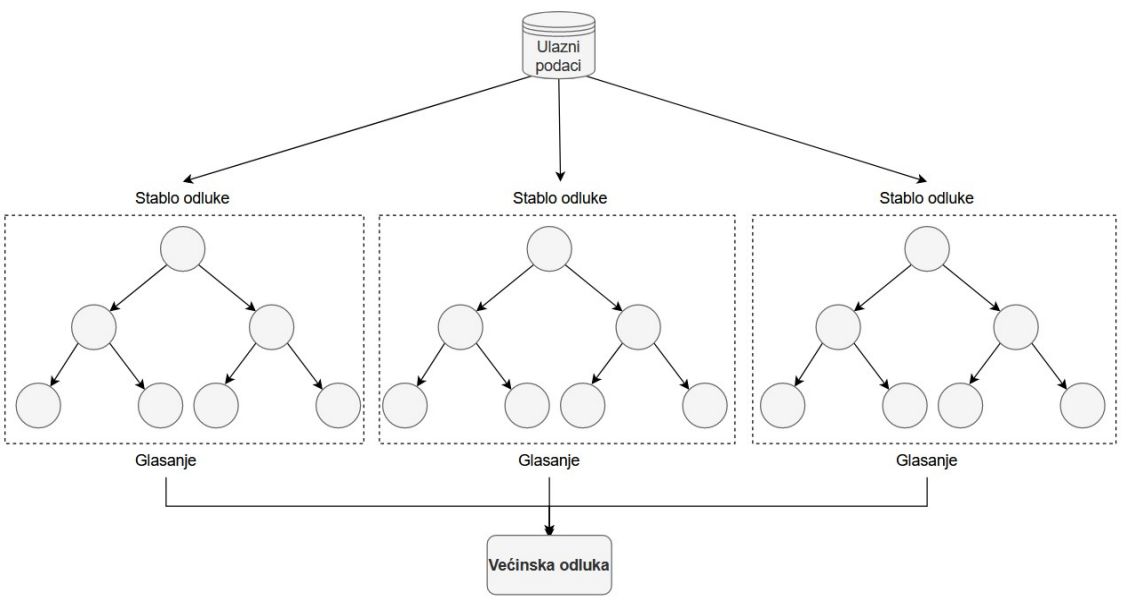
Uz parametre, dvije glavne tehnike koje se koriste su rezanje (eng. *pruning*) te razdvajanje (eng. *splitting*). Kod modeliranja će se razdvajanje izvršavati dok se ne pronađe najbolji čvor za određenu značajku. Cilj je ostvarenje visokog stupanja čistoće to jest smanjenje entropije, koja će doprinijeti predviđanju izlazne klase. Mjere koje se koriste u čvoru za vrednovanje su entropija, Ginijev indeks, pogreška klasifikacije te informacijski dobitak. Nakon što se izgradi stablo, primjenjuje se rezanje podstabla koje donosi suvišne informacije u odlučivanju pri čemu rezanje može biti unaprijed ili unatrag. Rezanje unatrag je bolji izbor zbog izbjegavanja pretreniranja jer ne ograničava rast stabla u procesu treniranja. [68, 70, 71]. Ukoliko se provodi rezanje, tada mora postojati podjela podataka na testne i trenirane kako bi se vrednovanjem moglo odrediti mjesto rezanja.

4.7 Algoritam Slučajna šuma

Algoritam Slučajna šuma (eng. *Random forest*) je algoritam koji je stvorio Leo Breiman. Sastoji se od više definiranih stabala koja predstavljaju Stablo odluke. Ovaj algoritam se često primjenjuje u rješavanja problema iz područja bioinformatike, a jednako kvalitetno rješava klasifikacije i regresijske probleme pa nema ograničenja u ulaznim podacima. Štoviše, njegova snaga ne očituje se samo u tome već i u učinkovitom korištenju skupa podataka koji su kategorički brojčano neuravnoteženi. Također, nema problema ni s različitim šumovima u podacima kao što su suvišni podaci i nedostajuće vrijednosti. Međutim, obradom takvih podataka povećava se vremenska složenost. S druge strane, najveći nedostatak koji se javlja je mogućnost pretreniranja, zbog čega treba koristiti parametarsku optimizaciju pri čemu treba također paziti jer preveliki broj klasifikatora može imati negativan učinak [72]. Osim toga, postoje problemi i sa korištenjem multiklasa zbog binarne stablaste strukture.

Izgradnja modela započinje na način da se ulazni podaci nasumično podijele na trenirane i testne podatke. Temeljem treniranih podataka izgradit će se n stabala odluke s m ulaznih značajki. Ova dva parametra korisno je računski ograničiti zbog računske kompleksnosti, no u suprotnom slučaju će se koristiti preddefinirane vrijednosti zadane u programskoj knjižnici. Svako podstablo se potom rekurzivno izgrađuje neo-

visno od drugih podstabala. U svakom čvoru jednog podstabla temeljem značajke koja pridonosi najveći informacijski dobitak ili Gini indeks, čvor će se podijeliti na desni i lijevi čvor odnosno podstablo. Temeljem podataka u *out-of-bag* testirat će se pojedinačno Stablo odluke te se dobiva prosječna pogreška koja pomaže algoritmu u boljoj raspodjeli značajki pomoću težinskih faktora [73]. *Out-of-bag* je glavna značajka koja omogućava kvalitetno izvođenje algoritma. Temeljem tog svojstva se slučajnim odabirom generira podskup podataka iz treniranih podataka te se individualno u svakoj iteraciji ispituje svako Stablo odluke [74]. Testni podaci se zatim koriste kako bi se odredilo kojoj klasi pripada, ako se instanca pusti kroz određeno Stablo odluke. Nakon što se odluči klasa za svako podstablo, temeljem grupnih glasova većinski se odlučuje o konačnom predviđanju ulaznog zapisa. Opisana struktura shematski je prikazana na slici 4.5.



Slika 4.5 Shematski prikaz algoritma Slučajna šuma

Postupak vrednovanja značajki bitan je koncept u ovom algoritmu. Pristupi koji se korite su *Gini importance* koji je prethodno spomenut te postoji i permutacijska varijanta. U slučaju korištenja permutacije, ona se zbiva u pozadini korištenjem *Out-of-bag* nakon čega se provodi tehnika odabira značajki bez da to korisnik programski zatraži. Najprije će se generirati podskup podataka kojim će se procijeniti stopa

pogreške u izgrađenom stablu. Zatim se izvrši permutacija tog podskupa na način da se vrijednosti jedne značajke permutiraju dok druge ostaju fiksne. Nakon ovog postupka ponovno se procjeni stopa pogreške. Ovim postupkom se svakoj značajki pridodaje važnost te se time može predvidjeti hoće li se greška predviđanja koristeći testne podatke smanjiti ili povećati [75].

4.8 Algoritam naivni Bayesov klasifikator

Algoritam naivni Bayesov klasifikator (eng. *Naive Bayes Classifier*) smatra se jednim od najstarijih algoritama nastalih na području umjetne inteligencije i predstavlja temeljno pravilo u vjerojatnosti i statistici. Međutim, i dalje je sveprisutan zbog svoje jednostavnosti i brzine izvršavanja obzirom da ne zahtijeva veliki skup podataka u procesu treniranja [76]. Brzina se očituje u unaprijed izračunatim vjerojatnostima koje se potom koriste ovisno o potrebnoj značajki. Izlazno predviđanje ovisit će o klasi koja ima najveću *a posteriori* vjerojatnost.

Glavna logika programskog izvršavanja zasnovana je na temeljnom Bayesovom teoremu prikazanim izrazom 4.6. U ovoj jednadžbi korištena je uvjetna vjerojatnost, no ponekad je ta vjerojatnost teško dostupna pa se koristi združena razdioba. Prednost, ali i nedostatak ovog pristupa je zanemarivanje povezanosti između svih značajki i ciljane klase uslijed korištenja uvjetne nezavisnosti klase. To predstavlja problem kod modeliranja stvarnih situacija. Unatoč tome, ističe ga fleksibilnost u korištenju visokodimenzionalnih podataka ne zahtjevajući redukcijsko smanjenje [77].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4.6}$$

gdje je:

- *P(A|B)* - vjerojatnost A ukoliko se događaj B dogodio (eng. *a posteriori*),
- *P(B|A)* - vjerojatnost B ukoliko se događaj A dogodio (eng. *likelihood*),
- *P(A)* - vjerojatnost A (eng. *a priori*),
- *P(B)* - vjerojatnost B (eng. *evidence*).

Jedan od optimizacijskih algoritama koji se ujedno koristi u ovom radu za potrebe tehnike omotač je *Gaussian Naive Bayes*. Uz njega, postoji još *Multinomial* i *Bernoulli*. On se koristi kada podaci slijede Gaussovu distribuciju. Pokazuje dobre računske rezultate kod kontinuiranih i velikih skupova podataka iako i dalje postoji problem uvjetne nezavisnosti to jest zanemarivanja povezanosti značajki. U računskom izrazu koristi se srednja vrijednost i standardna devijacija za svaku izlaznu klasu [78], što je prednost u odnosu na standardnu implementaciju. Ono šta ga čini optimizacijskim je nekorištenje kovarijance između promatranih podataka.

Poglavlje 5

Studija slučaja

Programski kod za strojno učenje i odabir značajki napisan je u programskom jeziku Python verzije 3.7. i algoritmi za strojno učenje su preuzeti iz knjižnice Scikit-learn. Kako bi se olakšao izračun značajki za sve peptide, korištena je programska knjižnica otvorenog koda opisana u nastavku. U ovom poglavlju bit će opisana cijela metodologija izrade programskog koda te njezina logika.

5.1 Ulazni podaci

Podaci korišteni u ovom radu dio su HRZZ uspostavnog istraživačkog projekta UIP-2019-04-7999 kojeg provodi istraživačka skupina DeShPet. Podaci se sastoje od dvije odvojene skupine, a to su katalitički i antimikrobni peptidi. Na obje skupine su primjenjivane iste tehnike navedene u radu i rezultati svake skupine su zasebno promatrani. Svaka skupina zapisana je u zarezom odvojene vrijednosti (eng. *Comma-separated values*, CSV) datoteke u kojima se na početku programa nalaze peptidi zapisani u formatu FASTA te zasebna odgovarajuća izlazna vrijednost u obliku nula ili jedan. Nula označava da peptid nije antimikroban odnosno ne izaziva katalitičke reakcije. Jedan označava da je antimikroban to jest izaziva katalitičke reakcije. Zapisa antimikrobnih peptida ima 10341 od čega je 5701 negativno klasificirano, a 4640 pozitivno klasificirano. Katalitičkih peptida ima 76 od čega je 27 negativno klasificirano, a 49 pozitivno klasificirano.

Prije procesa predobrade podataka, svaki peptid je zapisan iz format FASTA u format SMILES korištenjem metode *MolFromFASTA()* te potom *MolToSmiles()*. Primjer izgleda CSV datoteke u datom trenutku prikazan je na slici 5.1. Nakon odgovarajućeg zapisa, primijenjena je klasa *Calculator* u kojoj svaka metoda izračunava jednu značajku pri čemu je ulazni zapis ponovno transformiran metodom *MolFromSmiles()*. Parametrom *ignore_3D=True* je definirano da se izostave značajke koje su 3D obzirom da se zahtjeva drugačiji ulazni format. Nakon izračuna svih značajki, podaci su ponovno spremljeni u CSV datoteke. Pomoću Python knjižnice Mordred izračunate su sve značajke, a korištenjem knjižnice RDKit napravljen je peptidni zapis u format SMILES.

FASTA form	SMILES form	result
YVDVDVSV	CC(C)[C@H](NC(=O)[C@H](CO)NC(=O)[C@@H](NC(=O)[C@H](CC(=O)O)NC(=O)[C@@H](NC(=O)[C@H](CC(=O)O)NC(=O)[C@@H](NC(=O)[C@@H](N)Cc1ccc(O)cc1)C(C)C)C(C)C)C(C)C)C(=O)O	0
YVHVSVSV	CC(C)[C@H](NC(=O)[C@H](CO)NC(=O)[C@@H](NC(=O)[C@H](CO)NC(=O)[C@@H](NC(=O)[C@H](Cc1c[nH]cn1)NC(=O)[C@@H](NC(=O)[C@@H](N)Cc1ccc(O)cc1)C(C)C)C(C)C)C(C)C)C(=O)O	0
IAIHIRI	CC[C@H](C)[C@H](N)C(=O)N[C@@H](C)C(=O)N[C@H](C(=O)N[C@@H](Cc1c[nH]cn1)C(=O)N[C@H](C(=O)N[C@@H](CCCNC(=N)N)C(=O)N[C@H](C(=O)O)[C@@H](C)CC)[C@@H](C)CC)[C@@H](C)CC	1

Slika 5.1 Primjer ulazne CSV datoteke prije izračuna značajki

5.1.1 Analiza knjižnica za izračun značajki

Za usporednu analizu programskih knjižnica u programskog jeziku Python, a za izračun značajki peptida korištene su: PaDELPy [79] i Mordred [80]. Međutim, u radu je odlučeno da se koristi Mordred knjižnica zbog boljih računskih performansi.

Programska knjižnica Mordred napisana je izvorno u programskom jeziku Python te je trenutno dostupna verzija 1.2.0. Izračunava sveukupno 1826 značajki, od čega 1613 značajki zahtijeva ulaz u formatu 2D, a 213 u 3D formatu. Osnovne klase su *Descriptor* i *Calculator*. Pomoću klase *Calculator* moguće je izračunati sve značajke koje su dostupne u podmodulu kojih je 50. Svaki podmodul se zasebno sastoji

od određeng broja metoda koje izračunavaju specifične značajke definirane za taj podmodul poput broja prstenova, atoma, različitih indeksa i ostalog. Knjižnica je otvorenog koda te sadrži automatizirane testove koji omogućuju provjeru ispravnosti izvršene instalacije. Također, može se koristiti pomoću naredbenog retka ili u web sučelju. Omogućuje izračun svih značajki ili specifične značajke koju korisnik odredi. Bitna je karakteristika postojanje ugrađenih mehanizama za rukovanje s greškama kao što su primjerice nedostajuće vrijednosti [81].

S druge strane, **programska knjižica PaDELPy** izvorno je napisana u programskom jeziku Javi, ali je za potrebe korištenja u Pythonu napravljen ovaj oblik koji dolazi od knjižnice PaDEL. Također je otvorenog koda, a zadnja verzija je 0.1.11. Omogućava izračun za 1875 značajki od čega su 1444 za 2D, a 431 za 3D [82]. Podržava više dretveni rad na različitim operacijskih sustavima.

U širem smislu ove programske knjižnice slične su zbog karakteristika poput broja izračuna značajki i otvorenog koda. Međutim, unutarnje performanse im se drastično razlikuju zbog čega je i odlučeno da će se koristiti Mordred. U PaDELPy-ju postoje veliki problemi s brzinom izračuna, kao i veličinom ulaznog zapisa. Iako programski jezik Java jest brži od Pythona, zbog dodatnog omotača oko izvornog koda, se stvara usporavanje. Također, postoji problem kod izračuna dugačkih peptida što se pokazalo kao veliki problem kod izračuna AMP-a, gdje većina peptida nije mogla biti izračunata. Nadalje, veliki problemi postoje i s rukovanjem iznimkama. U slučaju da neku značajku nije moguće izračunati, zaustavit će se kod tog izračuna i neće nastaviti dalje. Osim toga, posjeduje vremensko ograničenje koje određuje koliko se dugo programska petlja odnosno računanje može izvršavati, a nakon toga se prekida izvršavanje.

Nasuprot tome, Mordred ima odlične vremenske performanse i nema ograničenje u dužini peptida. Posjeduje i mehanizam za rukovanje sa svim vrstama grešaka te obavještava korisnika ukoliko se dogodi *overflow* ili *underflow*. Međutim, unatoč velikom broju značajki koje izračunava, problem se javlja kod metoda koje očekuju dodatne ulazne parametre. Primjerice, parametri mogu biti tip operacije koja se

želi primijeniti u izračunu značajki, konstante u obliku broja veza u molekulskim prstenovima, duljina najvećeg prstena ili broj ugljikovih veza.

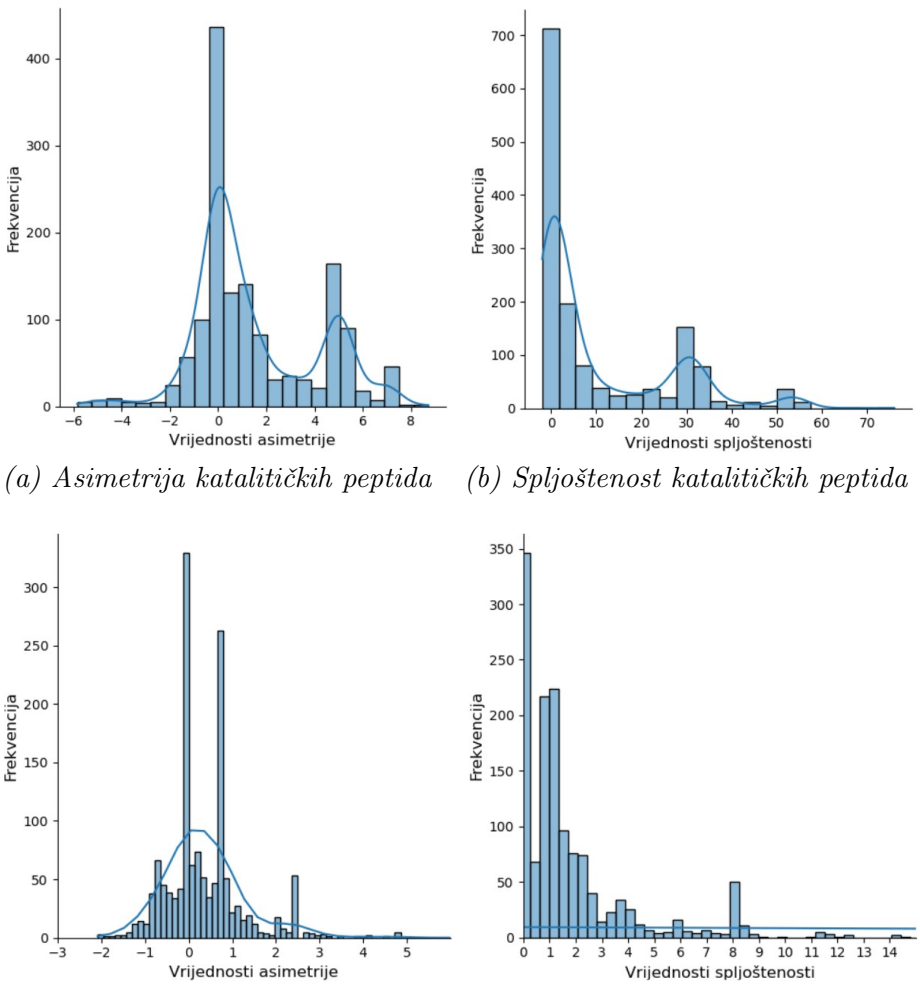
5.1.2 Predobrada podataka

U procesu predobrade podataka najviše pažnje posvećeno je čišćenju neodgovarajućih vrijednosti. Međutim, prvotno je napravljena analiza svih podataka prikazana na slici 5.2 kako bi se dobio bolji uvid. Raspodjela podataka definira njihovu međusobnu ovisnost te se promatra mjera spljoštenosti (eng. *Kurtosis*) i mjera asimetrije (eng. *Skewness*). Kod katalitičkih peptida distribucija podataka donekle prati ravnomjernu raspodjelu, od čega je 49,69% vrijednosti koeficijenata u rasponu između plus i minus jedan pri čemu je minimalna vrijednost asimetrije -5,8502, a maksimalna vrijednost 8,7178. Distribucija podataka je blago plosnata zbog 56,50% vrijednosti koeficijenta zaobljenosti koji su manji od tri. Minimalna vrijednost je -1,9769, a maksimalna 75,9999. Kod AMP-a distribucija podataka je također ravnomjerna odnosno simetrična zbog 82,70% vrijednosti u rasponu između plus i minus jedan pri čemu je maksimalna vrijednost 58,6941, a minimalna -2,0864. Distribucija podataka je plosnatija od normalne zbog 81,87% vrijednosti koeficijenta zaobljenosti koji su manji od tri. Minimalna vrijednost je -1,9320, a maksimalna 3443,6657.

Na početku predobrade podataka sve dobivene vrijednosti su dodatno analizirane pomoću knjižnice NumPy. Ukoliko je primijećena vrijednost koja pripada vrijednosti izvan raspona, odnosno po svojoj vrijednosti je minus ili plus beskonačno, onda je promijenjena u konstantu NaN. Time se osiguralo da ne dolazi do pogrešaka u usporedbi. U katalitičkom skupu pronađeno je 0 vrijednosti, a u AMP-u 69.

S obzirom da se značajke računaju tako da se za jedan peptid računaju sve moguće značajke iz definirane knjižnice, primijećeno je da se neke vrijednosti ne mogu izračunati. Na tim mjestima se postavlja string *None* koji se potom zamjenjuje s NULL odnosno NaN vrijednost. Također, neke vrijednosti su kompleksnim izračunom dovedene do *overflowa* ili *underflowa* zbog čega je bilo potrebno ručno pregledati sve CSV datoteke kako ne bi postojale krive unesene vrijednosti u strojno učenje. Analizom

podataka primijećeno je da u katalitičkom i antimikrobnom skupu podataka postoji deset značajki koje imaju problem s *overflow* te su one uklonjene iz skupa podataka. S druge strane, vrijednosti koje su *underflow*, ostavljene su u skupu podataka jer generalno ne predstavljaju problem u strojnom učenju te su neke od njih zabilježene kao nula.



(a) Asimetrija katalitičkih peptida *(b) Spljoštenost katalitičkih peptida*

(c) Asimetrija antimikrobnih peptida *(d) Spljoštenost antimikrobnih peptida*

Slika 5.2 Grafički prikaz mjere spljoštenosti i asimetrije

Glavno čišćenje podataka ostvareno je metodom u kojoj se izbacuju sve značajke koje u svim zapisima imaju konstantu ili NULL vrijednost. Takve značajke donose šum u podacima te ih je potrebno maknuti. Katalitički peptidi imaju 408 takvih značajki, dok ih AMP ima 407. Radi se o istim značajkama, osim jedne, u oba skupa podataka.

Sljedeća metoda izvršava provjeru postoje li u nekim značajkama pojedine vrijednosti koje imaju NULL. U slučaju da postoje, te vrijednosti su zamijenjene korištenjem algoritma k-NN točnije, *KNNImputer()* iz knjižnice sklearn.impute. Metoda je parametrizirana upotrebom pet najbližih susjeda i Euklidskom udaljenosti. Na taj se način definira nova vrijednost pri čemu sve susjedne vrijednosti imaju jednaku težinu u odluci. Ovakvih podataka nije bilo puno. U katalitičkom skupu pronađeno je 30 značajki, a u AMP-u 35.

Obzirom da se računaju različite značajke čiji izlazi nisu istog tipa i raspona vrijednosti, bitno je napraviti normalizaciju podataka. Koristeći metodu *MinMaxScaler()* iz knjižnice sklearn.preprocessing, sve vrijednosti su transformirane u rasponu između nula i jedan. Nova vrijednost je zapisana u obliku decimalnog broja ograničenog na šest decimalnih mjesta.

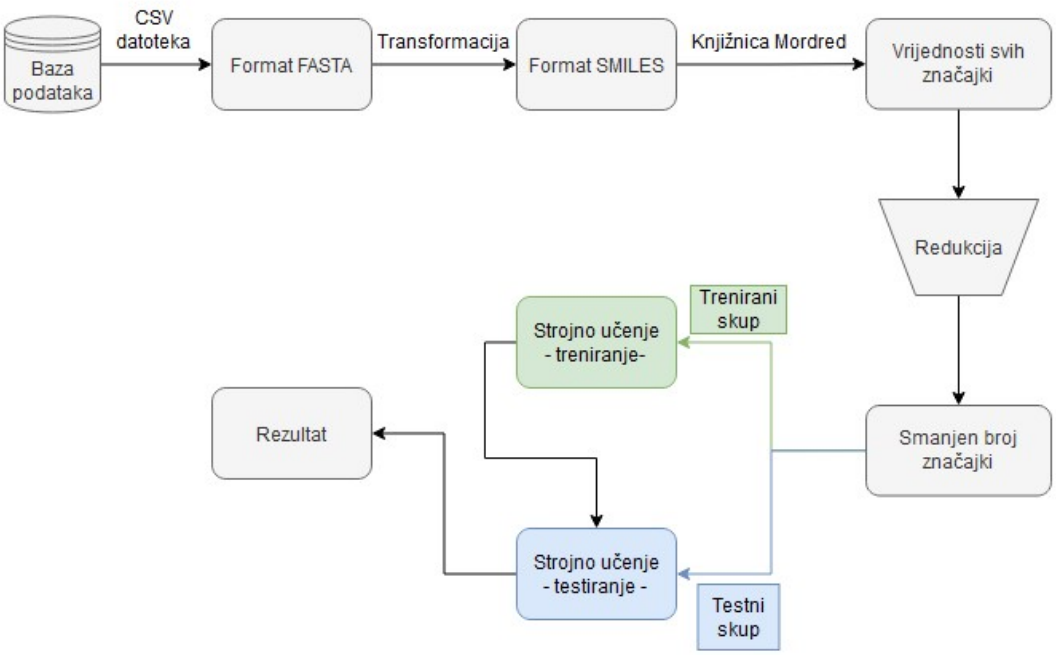
Zadnji korak kojim se osigurava da su suvišne informacije izbačene, jest provjera postotne jedinstvenosti vrijednosti u stupcima. Analizom podataka primijećeno je da nema duplih zapisa peptida, zbog čega je odlučeno da se izbace sve značajke koje imaju manje od 10% jedinstvenih vrijednosti. Broj takvih stupaca u katalitičkim je 47, a u AMP-u 111. Svi navedeni rezultati dostupni su u tablici 5.1.

Tablica 5.1 Broj značajki koje su uklonjene ili transformirane u predobradi podataka

Vrsta peptida	Katalitički	Antimikrobni
Vrijednosti izvan raspona	0	69
Značajke s konstantama i NULL	408	407
Značajke s nasumičnim NULL	30	35
Značajke sa slabim jedinstvenim vrijednostima	47	111
Duplicirani zapisi**	0	0
***Overflow* podaci**	10	10

5.2 Metodologija

Nakon izračunatih značajki i predobrade podataka, primjenjuju se filtar i omotač tehnike. Za svaku od tih tehnika koristi se odvojeni proces kako bi se moglo usporediti performanse. Svaki proces se sastoji od odabira značajki i strojnog učenja korištenjem algoritma Slučajna šuma. U nastavku je zasebno opisan svaki proces te cijeli hodogram prikazan je na slici 5.3.

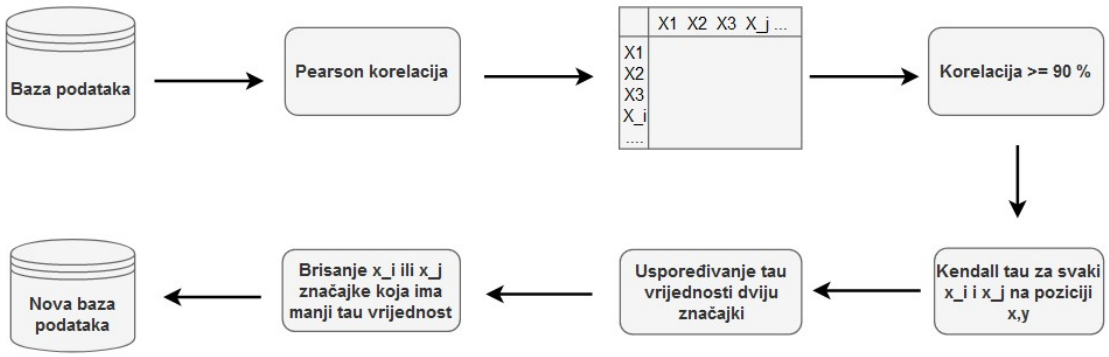


Slika 5.3 Shematski prikaz programskog projekta

5.2.1 Algoritamska primjena filtar tehnike

Filtar tehnika primjenjuje se na cijelom skupu značajki koja je prije toga očišćena od problematičnih vrijednosti i suvišnih značajki. U ovom procesu radi se redukcija cijelog skupa nakon čega se dobiva smanjen broj značajki. Jedini parametar kojim se utječe na broj odabranih značajki je vrijednost korelacije definirana kao konstanta između nula i jedan. Dobiveni novi skup podataka potom se dijeli na trenirani i testni skup podataka korištenjem unakrsne validacije. Nakon iteracijskog ponavljanja treniranja i testiranja, dobiveni su rezultati modela prikazani slikama te matricom zabune.

Obzirom da je tijekom analize skupa podataka uočena visoka korelacija između podataka, odlučeno je da će biti primijenjena dodatna korelacija kako se ne bi reducirao prevelik broj značajki. Najprije se koristi Pearson korelacija definirana u knjižici NumPy. Kad se dobije matrica NxN, s ciljem optimizacije se uzimaju samo vrijednosti iznad gornje glavne dijagonale korelacijske matrice koja se privremeno pohranjuje u jednu varijablu. Iterativno se provjerava za svaku vrijednost na poziciji x,y ima li trenutna vrijednost korelaciju veću ili jednaku od 90%. Bitno je napomenuti da je ova vrijednost eksperimentalno odabrana zbog prethodne analize podataka. Ukoliko je tvrdnja točna, tada se uzima značajka na x_i i x_j poziciji te se primjenjuje Kendall tau metoda preuzeta iz SciPy knjižnice. Korištenjem metode programski se dobije *tau* i *p* vrijednost. Međutim, *tau* je vrijednost koja se definicijski gleda kao ključna vrijednost. Značajka koja ima veći *tau* ostavlja se u skupu podataka a druga se izbacuje. Veća vrijednost *tau* govori da je promatrana značajka jače korelacijski ovisna o izlaznim vrijednostima i da ima veću važnost u definiranju klasifikacije peptida. Ovaj postupak ponavlja se za svaku vrijednost u korelacijskoj matrici. Navedeni postupak prikazan je na slici 5.4.



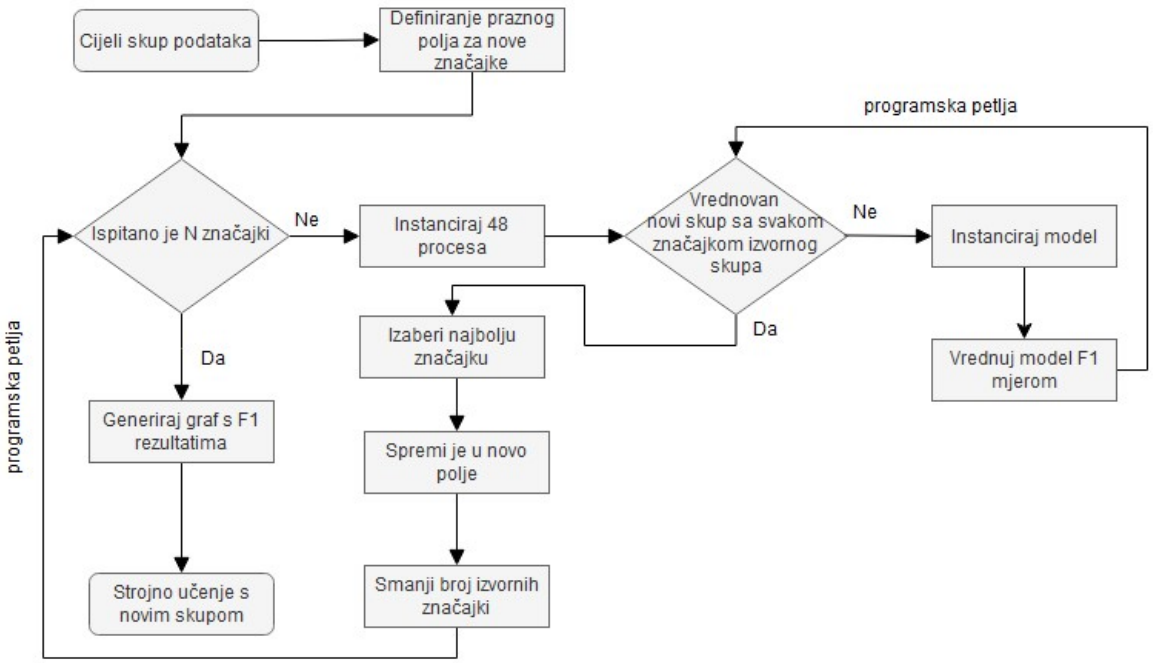
Slika 5.4 Hodogram tehnike filtar

5.2.2 Algoritamska primjena omotač tehnike

U sklopu ovog rada za algoritamsko izvođenje tehnike omotača korišteno je Superračunalo Bura Sveučilišta u Rijeci zbog zahtjevnih algoritamskih performansi. U nastavku rada prikazana je tehnika *forward* i *backward* kako bi se usporedile performanse i kvaliteta.

Forward tehnika započinje definiranjem praznog polja. U svakoj iteraciji se jedna značajka proglasi najboljom te se F1 mjera dobivena vrednovanjem modela i ime nove dodane značajke spremaju u dva odvojena polja. Metoda se sastoji od dvije programske petlje. Glavni uvjet koji se koristi u vanjskoj petlji je da se algoritam izvršava onoliko dugo koliko je potrebno da se ispitaju sve značajke. Zatim se u svakoj iteraciji instanciraju 48 procesa korištenjem klase *Pool*. Ovaj način korištenja multiprocesora ubrzava vremensko izvođenje. Nakon instanciranja se pokreće nova unutarnja petlja metodom *starmap()* u kojoj se uzima jedna značajka iz originalnog skupa. Postavlja se u privremeno polje u kojem se nalaze do sada odabrane najbolje značajke s kojima je model imao najbolji rezultat. Nadalje, definira se model temeljem algoritma strojnog učenja *Gaussian Naive Bayes* u kojem su svi parametri ostavljeni u predefiniranom obliku. Kako bi se izbjeglo pretreniranje, koriste se četiri preklopa. Mjera temeljem koje se odlučuje o kvaliteti značajke je srednja F1 vrijednost dobivena u četiri iteracije strojnog učenja.

Nakon vrednovanja svake značajke unutarnje iteracije, odabire se značajka sa kojom model ima najbolji rezultat te se ona sprema u novo polje. Kako se težilo algoritamskoj optimizaciji, odabrana značajka se uklanja iz izvornog skupa. Nakon završetka provjere cijele baze podataka, definiran je novi podskup značajki nad kojim će se izvršiti strojno učenje. Cijeli proces odabira značajki putem ove tehnike prikazan je na slici 5.5.

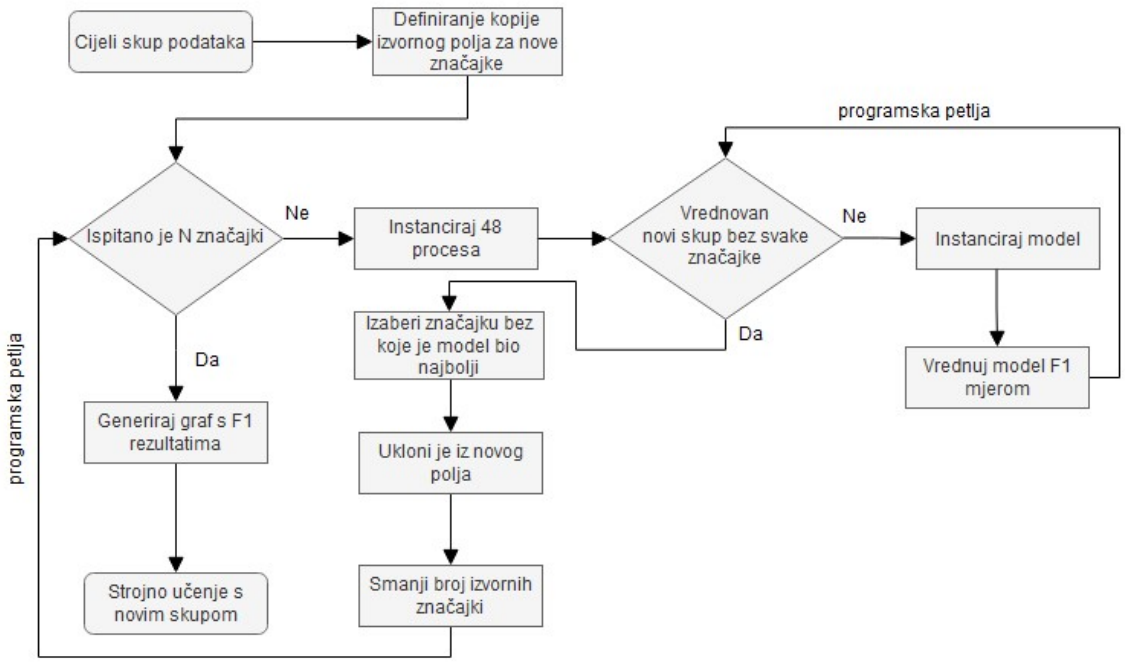


Slika 5.5 Dijagram slijeda forward tehnike

Backward tehnika je slična *forward*, a razlika je u početnom definiranom polju i smjeru pretrage. Kod ove tehnike definira se polje koje sadrži sve značajke izvornog skupa podataka. Metoda se također sastoji od dvije programske petlje. Izvršavanje vanjske petlje traje onoliko dugo koliko ima značajki. Koristi se 48 procesa iz klase *Pool* u svakom ispitivanju kako bi se ubrzalo vremensko izvođenje. Međutim, u unutarnjoj petlji, korištenjem metode *starmap()*, uklanja se jedna značajka iz novog skupa te se provjerava koliko je njezino uklanjanje doprinijelo modelu. Za definiranje modela koristi se algoritam strojnog učenja Stablo odluke čija je dubina ograničena na razinu pet kod korištenja AMP-a, dok je kod kataličkih peptida razina ograničena

na osam. Kao i u prijašnjoj tehnici za treniranje i testiranje modela, koristi se unakrsna validacija s četiri preklopa te mjera koja se računa unutar svake validacije je srednja F1 vrijednost.

Nakon što se u unutarnjoj petlji izračunaju F1 mjere, određuje se značajka bez koje je model imao najbolje performanse. Ta se značajka uklanja iz novog polja značajki kao i izvornog da bi se ostvarile bolje performanse. U dva odvojena polja sprema se ime izbačene značajke i ostvaren F1 rezultat. Također, dijagram slijeda ove tehnike prikazan je na slici 5.6.



Slika 5.6 Dijagram slijeda backward tehnike

5.2.3 Implementacija strojnog učenja

Katalitički peptidi i AMP koriste istu metodu kojom je implementirano strojno učenje. Nakon što se reduciraju značajke i ponovno spreme u CSV datoteke, one se koriste za nanovo čitanje podataka u obliku *DataFrame*. Pritom se uklanjaju zapisi

peptida FASTA, SMILES te se rezultante klasifikacije spremaju u posebnu varijablu.

Za kreiranje modela koristi se algoritam strojnog učenja Slučajna šuma definiranom metodom *RandomForestClassifier()*. Obzirom da postoje razni parametri koji se mogu postaviti, eksperimentalno su odabrana četiri parametra. Definirano je da se koristi 600 stabala za procjenu modela, a maksimalni broj značajki određen je korijenskim brojem značajki. Minimalni broj značajki potrebnih u nekom čvoru da se proglasi list je šest, a određeno je da se pedeset puta nasumično izmiješa cijeli skup podataka.

Obzirom da se koriste dvije vrste ulaznih podataka, potrebno je koristiti i dvije vrste unkarsne validacije zbog različitog broja zapisa u datoteci. Za katalitičke peptide koristi se *LeaveOneOut()*, dok se za AMP koristi *StratifiedKFold()* s definiranim parametrom deset za broj preklopa. Podaci se ne dijele na standardan način na teste i trenirane, već se pomoću unakrsne validacije nasumičnim odabirom izabiru indeksi zapisa koji će biti testni to jest trenirani podaci. Model se prvo trenira s treniram podacima, nakon čega se radi testno predviđanje. U zasebno se polje spremaju vrijednosti dobivenih predviđanja, postotak točnih predviđanja te točne klasifikacije.

Zadnji dio obuhvaća korištenje varijabli u kojima su spremljeni predviđeni i točni rezultati. Pomoću njih se radi matrica zabune, generira ROC krivulja i graf za prikaz važnosti značajki. Također, računa se vrijeme trajanja unakrsne validacije mjereno u sekundama.

Poglavlje 6

Rezultati

Za vrednovanje rezultata korištene su ROC-AUC vrijednosti, vrijeme potrebno za izračun značajki kao i vrijeme potrebno za strojno učenje, izračun matrice zabune te grafički izračun važnosti značajki za svaki model. Dodatno izračunate mjere su točnost, preciznost, opoziv, F1 rezultat i srednja geometrijska vrijednost. Najveća razlika primijećena je u vremenu[3] izvršavanja mjerenom u sekundama, te u ROC-AUC vrijednostima.

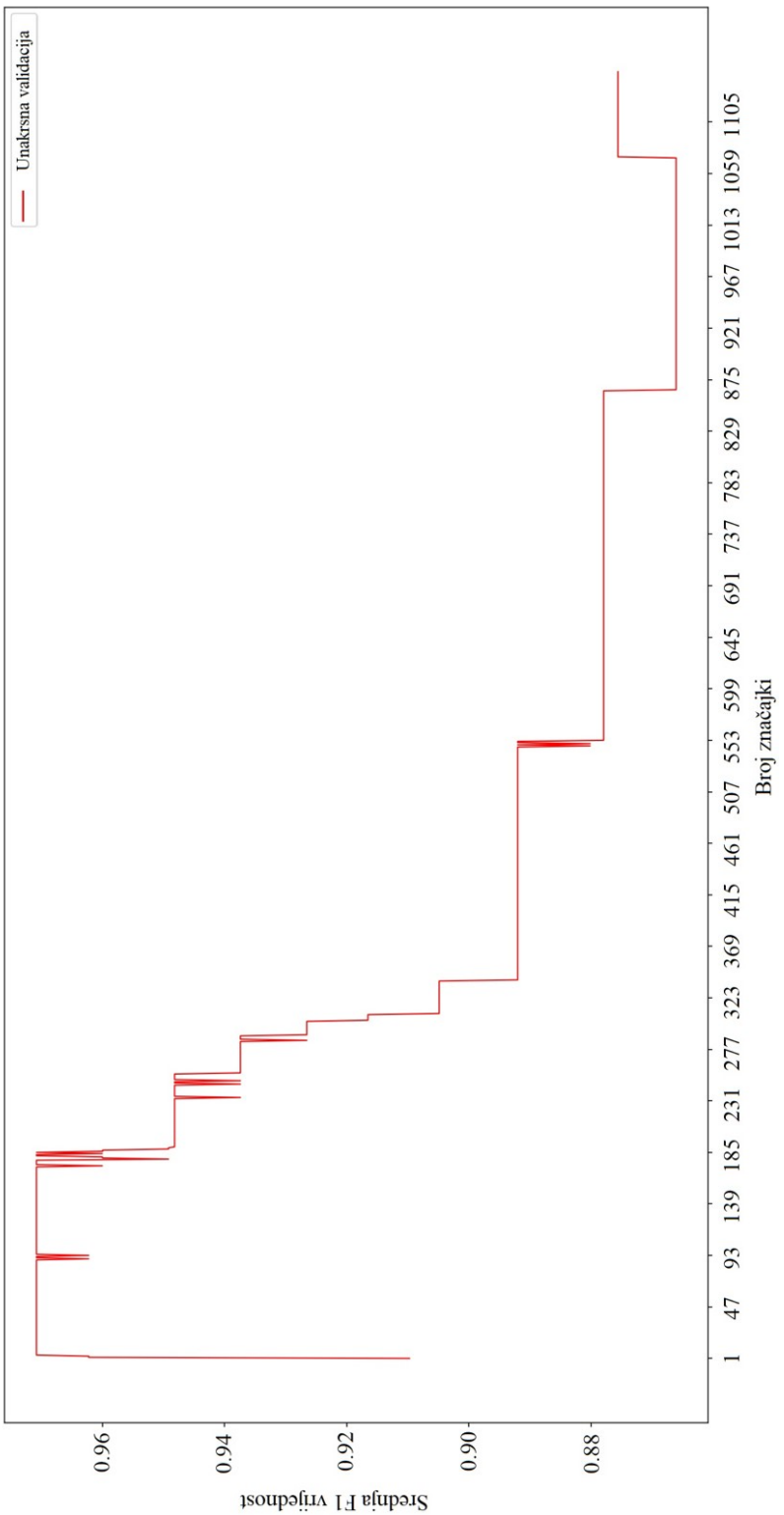
6.1 Usporedba tehnika odabira značajki

Filtar tehnika je vrsta tehnike koju karakterizira velika brzina izvršavanja. Njezina primjena u katalitičkim peptidima rezultira brzinom pretraživanja od 00:00:38,64 sekundi. Ulazni podaci su dimenzije 76 redova i 1150 stupaca odnosno značajki. Korištenjem Kendall tau uz Pearson korelaciju pronađeno je ukupno 906 značajki koje su statistički proglašene manje relevantnim za zavisnu varijablu. Kod AMP-a je situacija drugačija zbog većeg broja ulaznih zapisa. Podaci se strukturno sastoje od 10341 reda i 1087 značajki. Izvršavanje filtar tehnike traje 00:03:33,88 sekundi nakon čega se 796 značajki proglasi manje relevantnima.

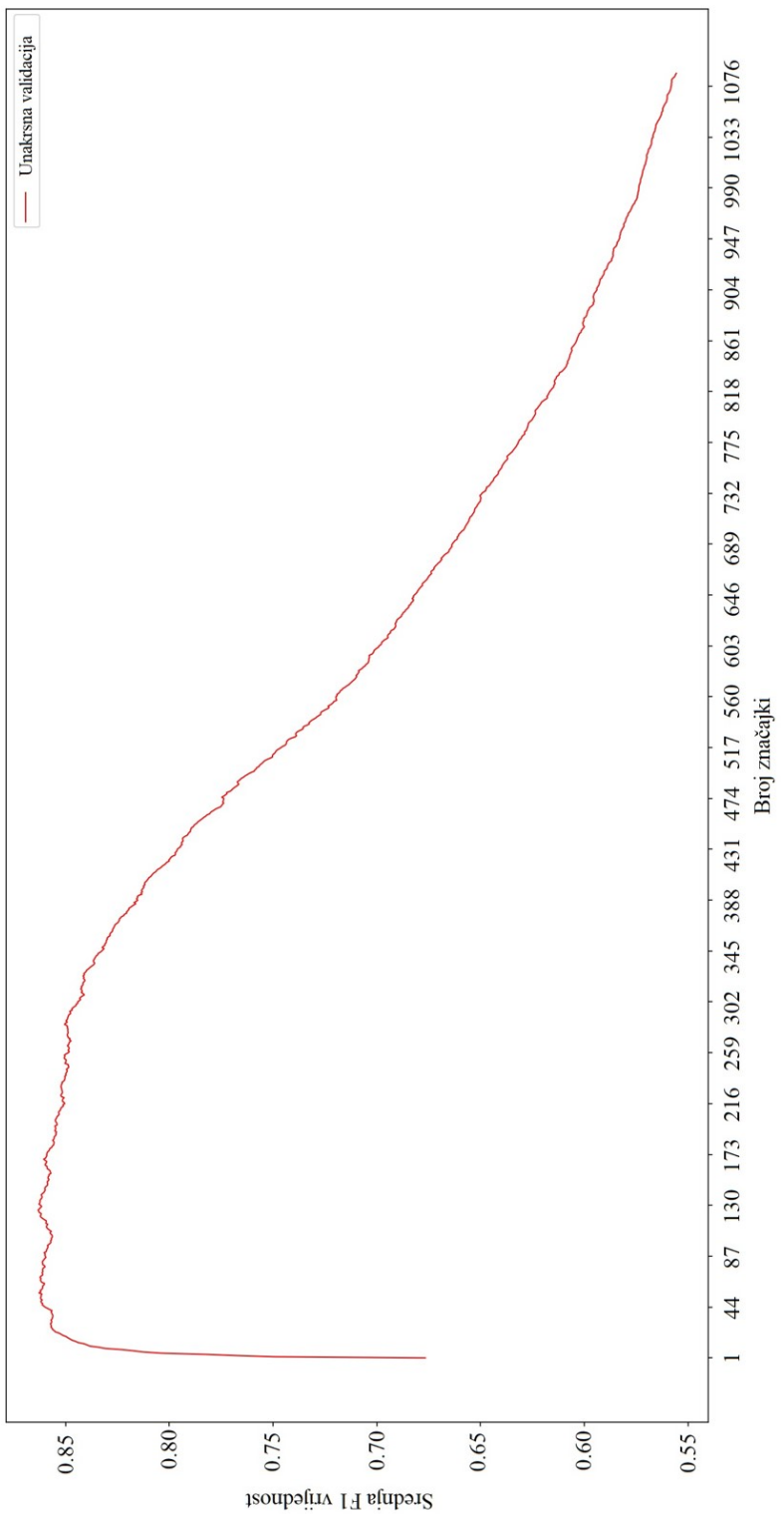
Omotač tehnika ima karakteristiku dugog vremenskog izvršavanja što je ovim ra-

[3]Zapis vremena prikazan je u obliku hh:mm:ss zbog jednostavnije čitljivosti.

dom i potvrđeno. *Forward* je generalno brža tehnika zbog dodavanja značajki u novom skupu te je u ovom radu korišten klasifikator *Gaussian Naive Bayes*. Potrebno vrijeme za pronalazak relevantnih značajki u skupu katalitičkih peptida iznosi 00:18:03,06 sekundi. Na slici 6.1 prikazane su sve F1 mjere modela dobivene izračunom. Analizom rezultata, odabrane su četiri značajke obzirom na to da je koristeći te značajke ostvarena F1 vrijednost modela 0,9708 te se dalje vrijednost ne povećava, već ostaje ista ili opada tijekom vremena s dodavanjem novih značajki. S druge strane, AMP se sastoji od većeg broja zapisa pa je i dulje vrijeme pretrage, a iznosi 04:42:17,79 sekundi nakon čega je odabrano 126 značajki. Tim skupom, model je ostvario najbolji rezultat pri čemu F1 mjera iznosi 0,8632. Naknadnim dodavanjem značajki vrijednosti opadaju kao i kod katalitičkih peptida. Na slici 6.2 prikazane su sve F1 vrijednosti modela s dodavanjem značajki.



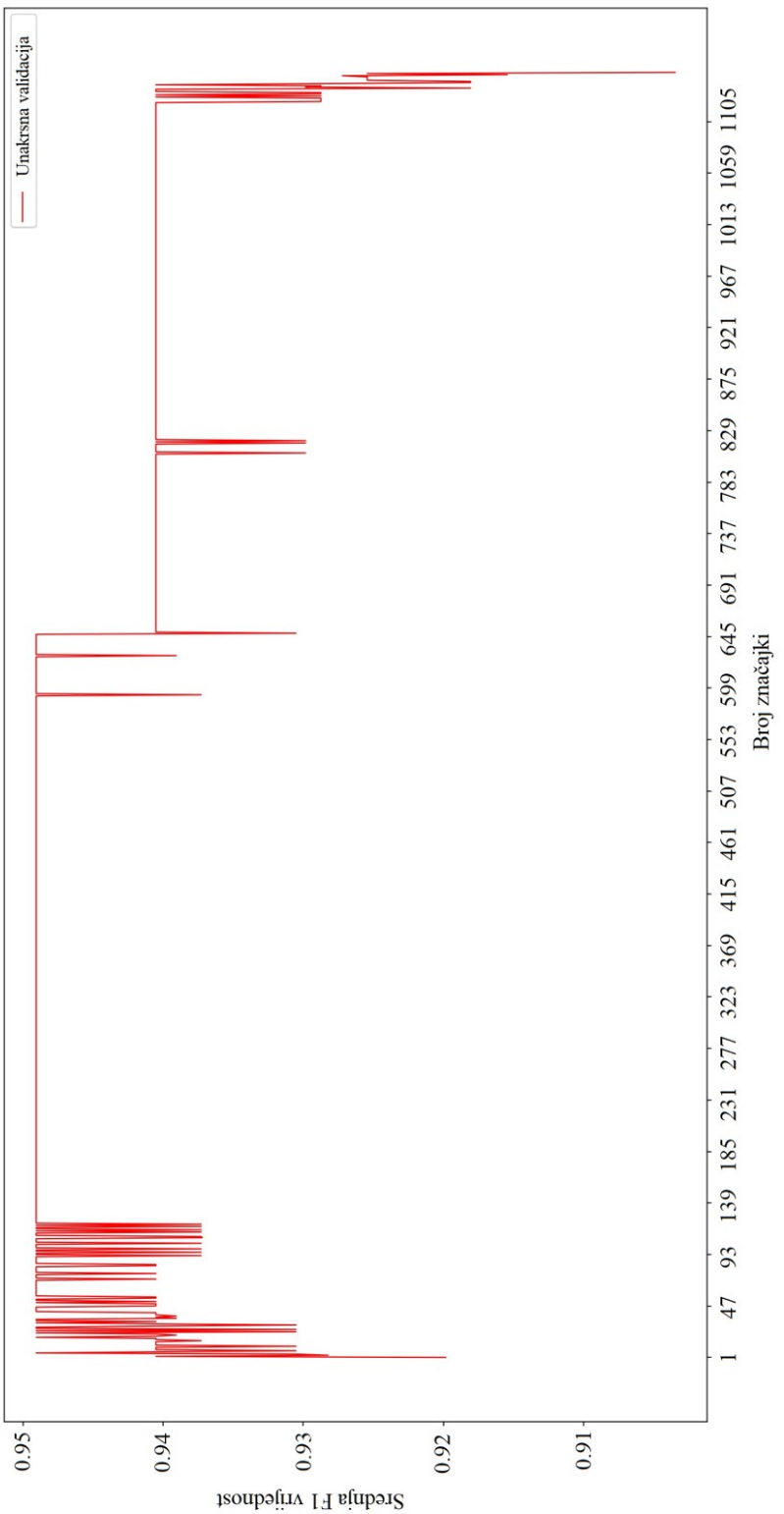
Slika 6.1 Forward tehnika kod pretrage katalitičkih peptida



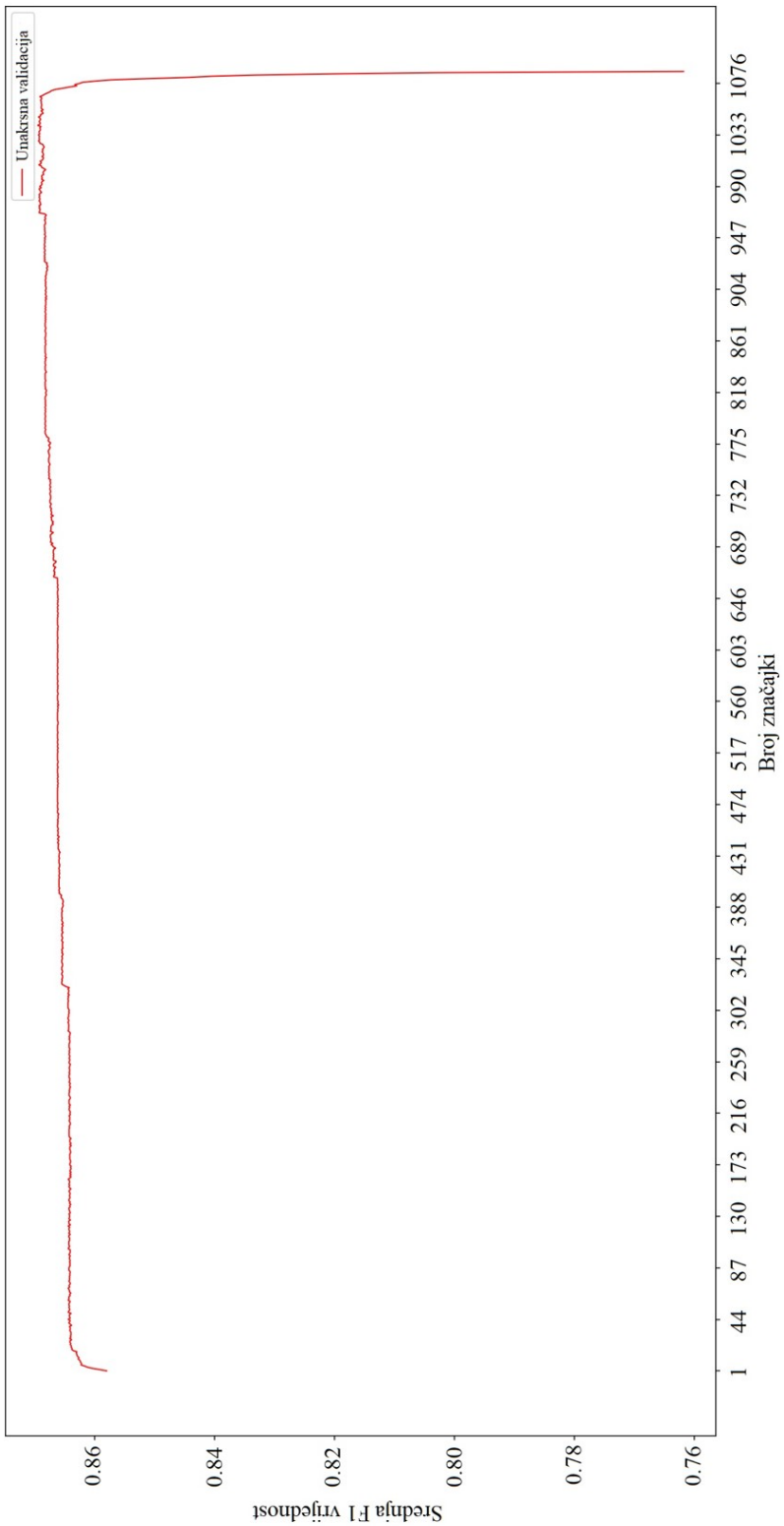
Slika 6.2 Forward tehnika kod pretrage antimikrobnih peptida

Backward tehnika imala je očekivano sporije performanse izvođenja pri čemu je ulazni skup identičan kao i kod *forward* tehnike te se koristi klasifikator Stabla odluke. Kod katalitičkih peptida vrijeme pretrage traje 00:34:42,99 sekundi nakon čega se odabiru zadnje 503 značajki te je model imao najbolju F1 vrijednost u iznosu 0,9490. Na slici 6.3 prikazano je postepeno izbacivanje značajki i opadajuće vrijednosti kod vrednovanja modela. S druge strane, pretraga i odabir značajki kod AMP-a vremenski je trajala 2 dana, 10:46:08,57 sekundi. Uzrok ovakve pretrage je velika složenost pri usporedbi svake značajke. Odabrano je zadnjih 45 značajki pri čijem je vrednovanju model imao F1 vrijednost u iznosu 0,8695. Na slici 6.4 prikazane su sve vrijednosti vrednovanja modela u postupku odabira značajki.

Iz priloženih rezultata jasno je vidljivo da su omotač tehnike znatno skuplje u smislu korištenja računalnih resursa. Izvođenje ovih algoritama, zbog ograničenja programske knjižnice, moguće je samo na procesoru što dodatno iziskuje veliku snagu i veliki broj jezgri procesora. Korištenje algoritma *Gaussian Naive Bayes* u *forward* tehnici ubrzava izvođenje algoritma, no u *backward* slučaju performanse su približno jednake kao i kod korištenja algoritma Stablo odluke. Najefikasniji odabir značajki ostvaren je filtar tehnikom.



Slika 6.3 Backward tehnika kod pretrage katalitičkih peptida



Slika 6.4 Backward tehnika kod pretrage antimikrobnih peptida

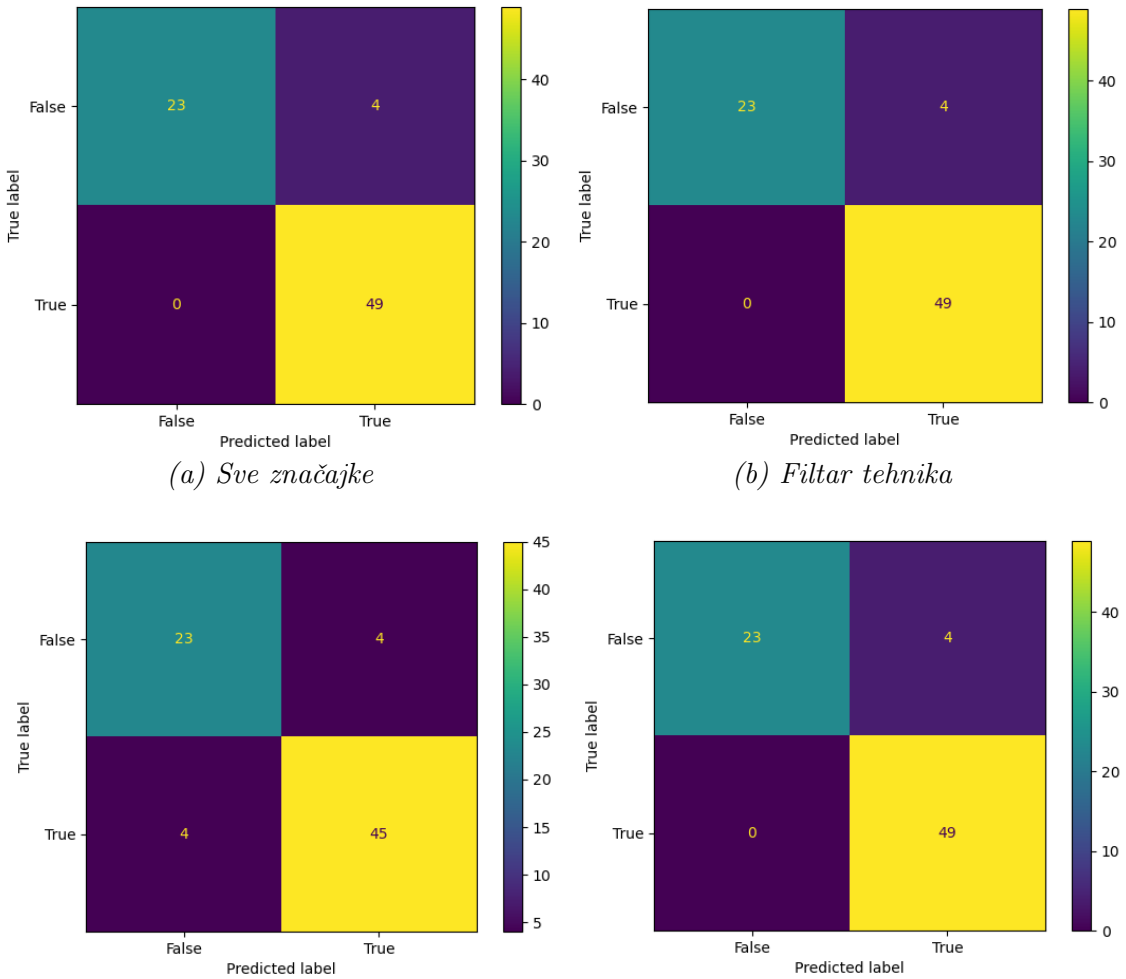
6.2 Usporedba performansi strojnog modela

Strojno učenje primjenjuje se nad modelima koji se zasebno treniraju i testiraju temeljem svih značajki ili odabranog skupa ovisno o definiranoj tehnici. U tablici 6.1 prikazani su rezultati svih primijenjenih tehnika korištenjem skupa katalitičkih peptida. Usporedbom svih relevantnih mjera, kao i vremenom za odabir značajki za svaku tehniku, pokazalo se da strojni model najbolje rezultate ostvaruje nakon korištenja filtar tehnike odabirom 244 značajke. Odlični rezultati dobiveni su u kratkom vremenu unakrsne validacije u trajanju od 65,25 sekundi i odnose se na točno predviđanje od 94,7% , F1 rezultat od 96,1%, srednju geometrijsku vrijednost od 0,996, opoziv od 100%, te preciznost od 92,5%. Od ovih rezultata se posebno ističe srednja geometrijska vrijednost koja je veća za 0,034 od rezultata dobivenih s drugim modelima i za 0,078 kod modela nakon primjene tehnike omotač *forward*. U korist zasigurno ide činjenica da su značajke visoko korelirane, zbog čega model jednostavnije nauči prepoznati uzorak u katalitičkim peptidima. Modeli nakon primjene omotač *backward* i filtar tehnika imaju slične rezultate, osim u vremenu potrebnom za treniranje modela, ROC-AUC vrijednosti i srednje geometrijske vrijednosti. Vrijeme unakrsne validacije traje 71 sekundu zbog korištenja 503 značajki, iako je unatoč tome dobivena slabija srednja geometrijska vrijednost u iznosu od 0,962. Iz toga proizlazi da je postojao veći broj peptida s izlaznom klasom nula, što je dovelo do prednosti te klase u odnosu na cijeli skup podataka. Nakon primjene omotač *forward* tehnike, dobiveni su najslabiji rezultati korištenjem četiri značajke u vremenu unakrsne validacije 63,68 sekundi koji iznose točnost 89,5%, srednja geometrijska vrijednost 0,918, opoziv 91,8%, preciznost 91,8% i F1 rezultat 91,8%. Ove vrijednosti su najmanje naspram rezultata drugih modela pri čemu se posebno ističu razlike u točnosti od 0,052 odnosno 5,2% i u opozivu od 0,082 odnosno 8,2%. Usporedbom rezultata svih modela s modelom koji je koristio sve značajke, vidljivo je da su vrijeme validacije i ostale mjere približne jednake.

Tablica 6.1 Mjerni rezultati strojnog učenja korištenjem katalitičkih peptida

Odabir značajki	Nema	Filtar	Omotač *forward*	Omotač *backward*
Broj značajki	1150	244	4	503
Vrijeme odabira	00:00:00	00:00:38,64	00:18:03,06	00:34:42,99
Vrijeme validacije	00:00:58,38	00:01:05,25	00:01:03,68	00:01:11,00
Točnost	0,947	0,947	0,895	0,947
Preciznost	0,925	0,925	0,918	0,925
Opoziv	1,0	1,0	0,918	1,0
F1 rezultat	0,961	0,961	0,918	0,961
G-mean	0,962	0,996	0,918	0,962
ROC-AUC	0,919	0,939	0,915	0,883

Vrijednosti dobivene matricom zabune na slici 6.5 prikazuju da svi strojni modeli imaju veliku točnost u predviđanju katalitičke klasifikacije. Također, svi imaju krivo predviđanje za četiri peptida čija je izlazna klasifikacija negativna. No, to ne predstavlja značajan problem zbog visoke točnosti u ostalim mjerama. Iznimka je da model nakon *forward* tehnike pokazuje slabije performanse kod predviđanja četiri peptida. Model je predvidio nulu odnosno odsustvo katalitičkih reakcija korištenjem tih peptida dok je točna klasifikacija jedan to jest izazivanje katalitičkih reakcija. Ostali modeli nisu imali krivo predviđanje za peptide koji izazivaju katalitičke reakcije. Za 23 peptida čija je klasifikacija i predviđanje nula, svi modeli imaju istu točnost. Nadalje, za 49 peptida čija je klasifikacija i predviđanje jedan, modeli nakon tehnika filtar i *backward* te model sa svim značajkama imaju istu točnost.



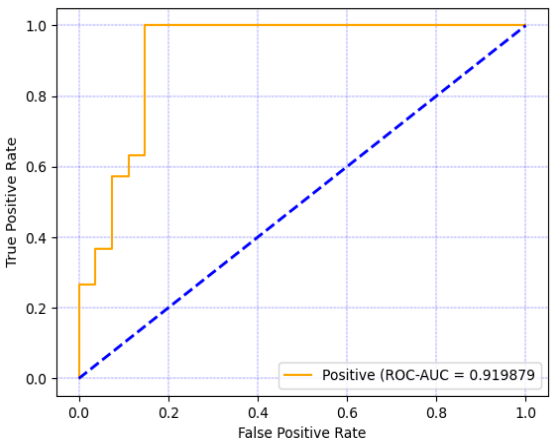
(a) Sve značajke (b) Filtar tehnika

(c) Omotač forward (d) Omotač backward

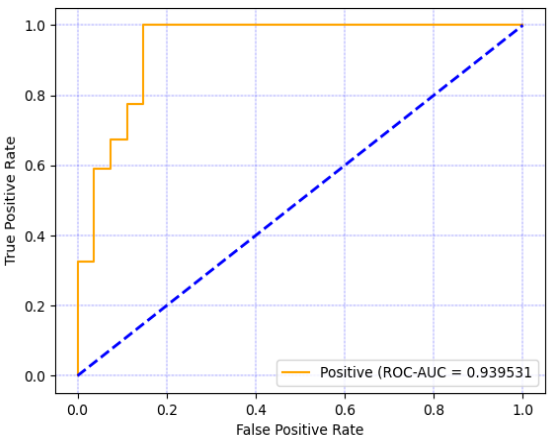
Slika 6.5 Matrica zabune strojnog modela kod kataličkih peptida

Ključna mjera vrednovanja ROC-AUC prikazana je na slici 6.6 za katalitičke peptide. Iako svi modeli imaju visoke vrijednosti, cijena koja se plaća za intenzivno korištenje procesora je ipak prevelika u nekim tehnikama. *Backward* tehnika koja ima najveću potrošnju procesora, ima najslabiji rezultat od 88,28%. Razlog tomu može biti činjenica da je algoritam implementiran tako da ukoliko postoji više značajki čijim se izbacivanjem ostvaruje ista F1 vrijednost modela, onda se izbacuje prvi po abecedi. Buduće istraživanje bi zahtijevalo redizajniranje ovog algoritma. Nadalje, najbolji rezultat ostvaren je korištenjem filtar tehnike u iznosu 93,95% . Model koji je koristio

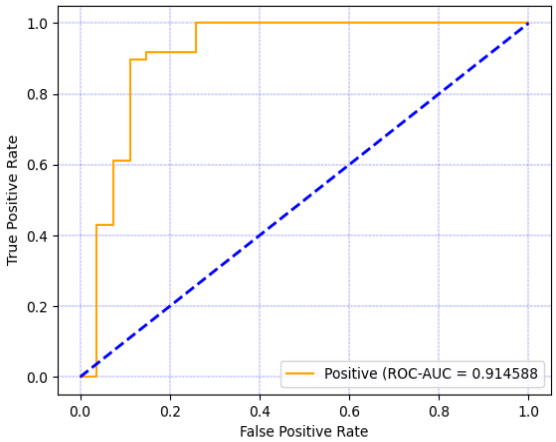
sve značajke i model koji je koristio značajke nakon omotač *forward* imaju približno jednake dobre rezultate u klasifikaciji koji iznose 91,99% i 91,46%.



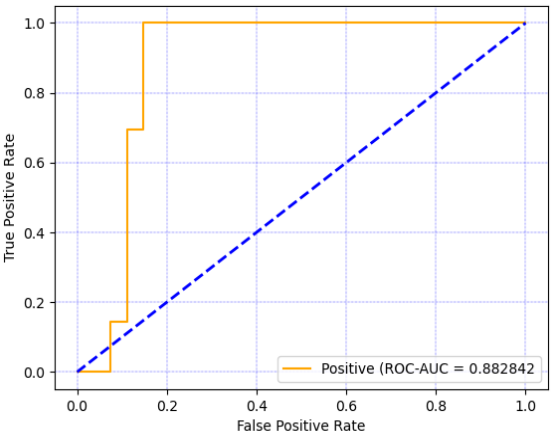
(a) Sve značajke



(b) Filtar tehnika



(c) Omotač forward

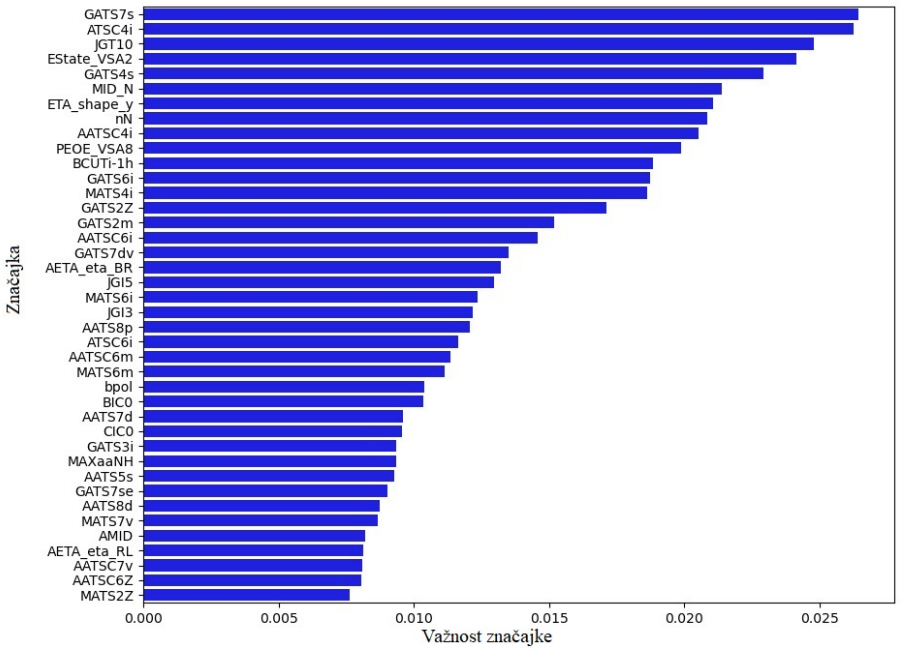


(d) Omotač backward

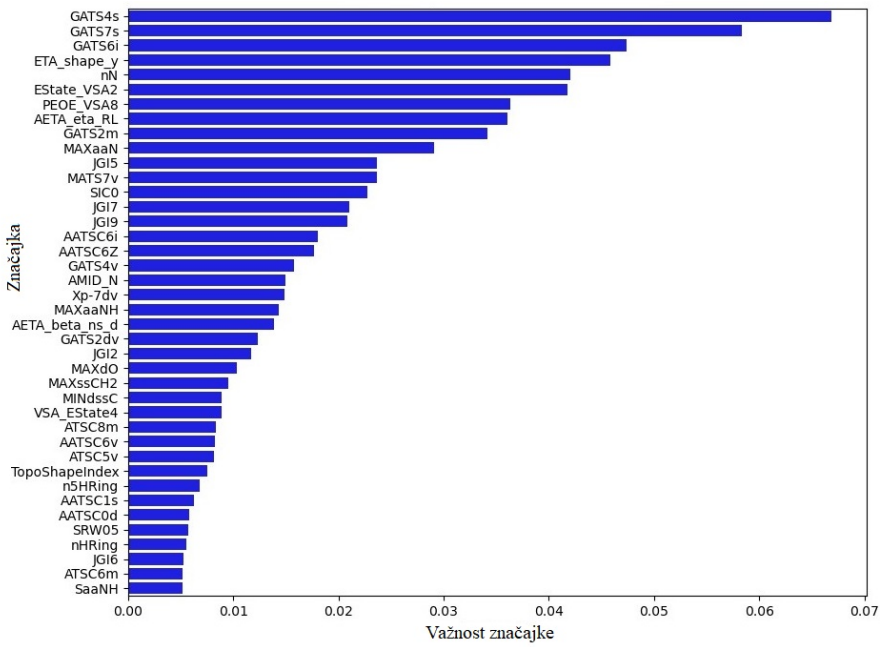
Slika 6.6 Grafički prikaz ROC-AUC vrijednosti strojnog modela kod katalitičkih peptida

Važnost značajki je informacija koja govori koliko neka značajka smanjuje Gini nečistoću u postupku dijeljenja čvora, pri čemu je ova relativna mjera srednja vrijednost za sva stabla odluke u šumi. Na slici 6.7 prikazano je prvih četrdeset značajki u skupu katalitičkih peptida koje su najrelevantnije od strane modela. Ne postoji preklapanje odabranih značajki po važnosti između različitih tehnika. Jedan od mogućih razloga

je nasumično miješanje cijelog skupa podataka prije treniranja s ciljem oponašanja stvarnih uvjeta. Drugi mogući razlog je da se ova metoda zasniva na korelacijskom promatranju, što znači da model sam bira koja je značajka relevantna. Međutim, svakako valja primijetiti da je najveća važnost značajki dobivena nakon primjene filtar tehnike ako se uzme u obzir i broj značajki. Drugim riječima, tom tehnikom model kod velikog broja stabala postavlja istu značajku za dijeljenje čvora na podstabla. Također, postoji pojavljivanje pojedinih istih značajki kao i kod modela koji koristi sve značajke samo drugim poretkom. Model nakon primjene *forward* tehnike također ima značajke sa visokom važnosti, no u prilog tomu ide što su korištene četiri značajke. Nakon primjene omotač *backward* tehnike, model je poredao značajke po važnosti drugačijim redoslijedom i drugačijim vrijednostima. Velik utjecaj ima broj značajki koji je veći nego kod ostalih dviju tehnika.

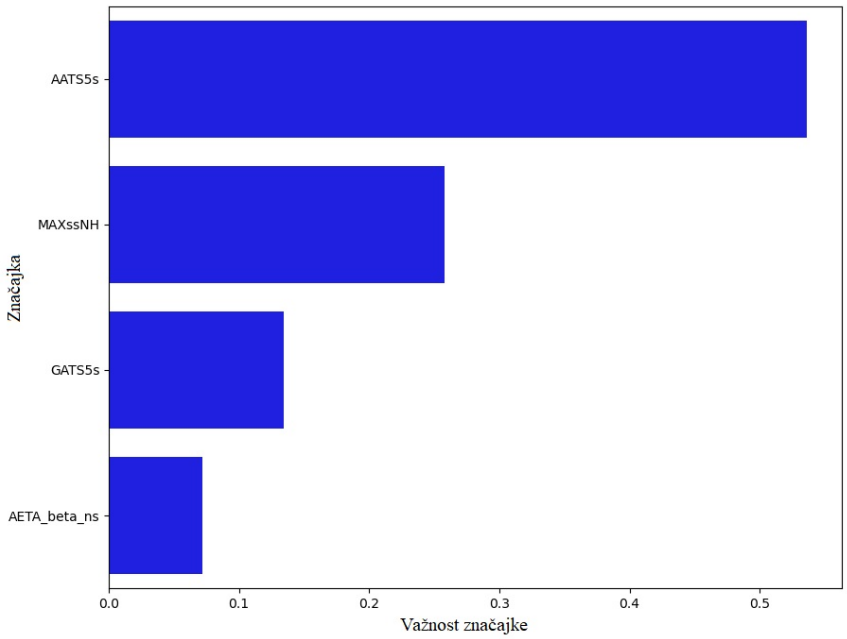


(a) Sve značajke

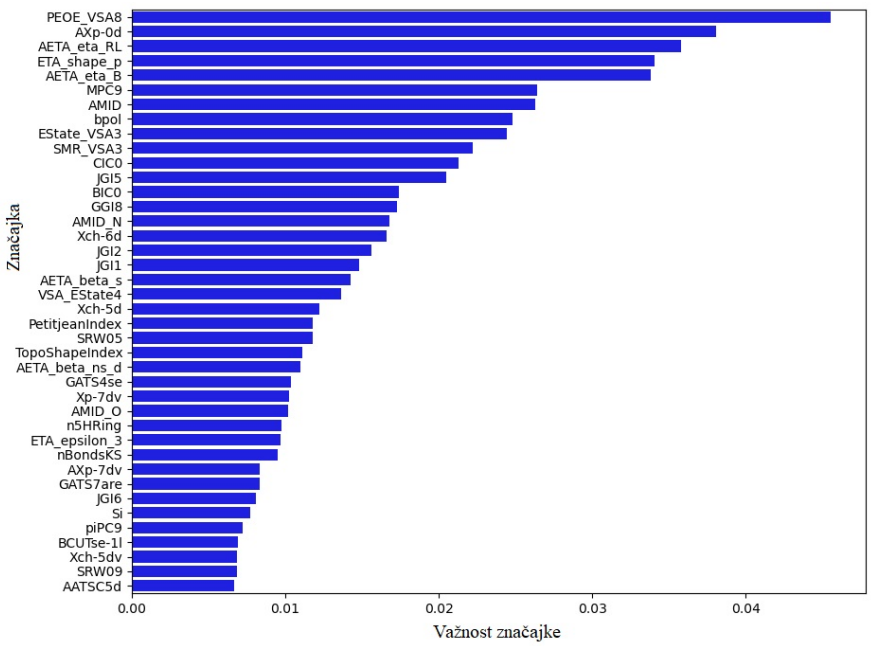


(b) Filtar tehnika

Slika 6.7 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod katalitičkih peptida - 1. dio



(c) Omotač forward



(d) Omotač backward

Slika 6.7 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod katalitičkih peptida - 2. dio

Analiza dobivenih rezultata kod katalitičkih peptida dovodi do jasnog zaključka da strojni model ima najbolje performanse nakon redukcije značajki pomoću filtar tehnike osobito zbog visoke ROC-AUC vrijednosti i F1 rezultata. Druga opcija koja se može razmatrati je korištenje svih značajki u modelu jer se tada dobiva isti broj točnih predviđanja, F1 rezultat od 96,1%, točnost od 94,7% te je ROC-AUC minimalno lošiji u razlici za 0.02. Svakako, ova opcija nije uvijek učinkovita i model se može zbuniti zbog redundantnih informacija. Model nakon primjene omotač *forward* ostvaruje najslabiju točnost od 89,5% što je žrtva zbog korištenja samo četiri značajke. Nakon primjene omotač *backward* je ostvarena bolja točnost i F1 rezultat, ali ROC-AUC ima slabiji rezultat unatoč velikom broju značajki te najduljem vremenu validacije.

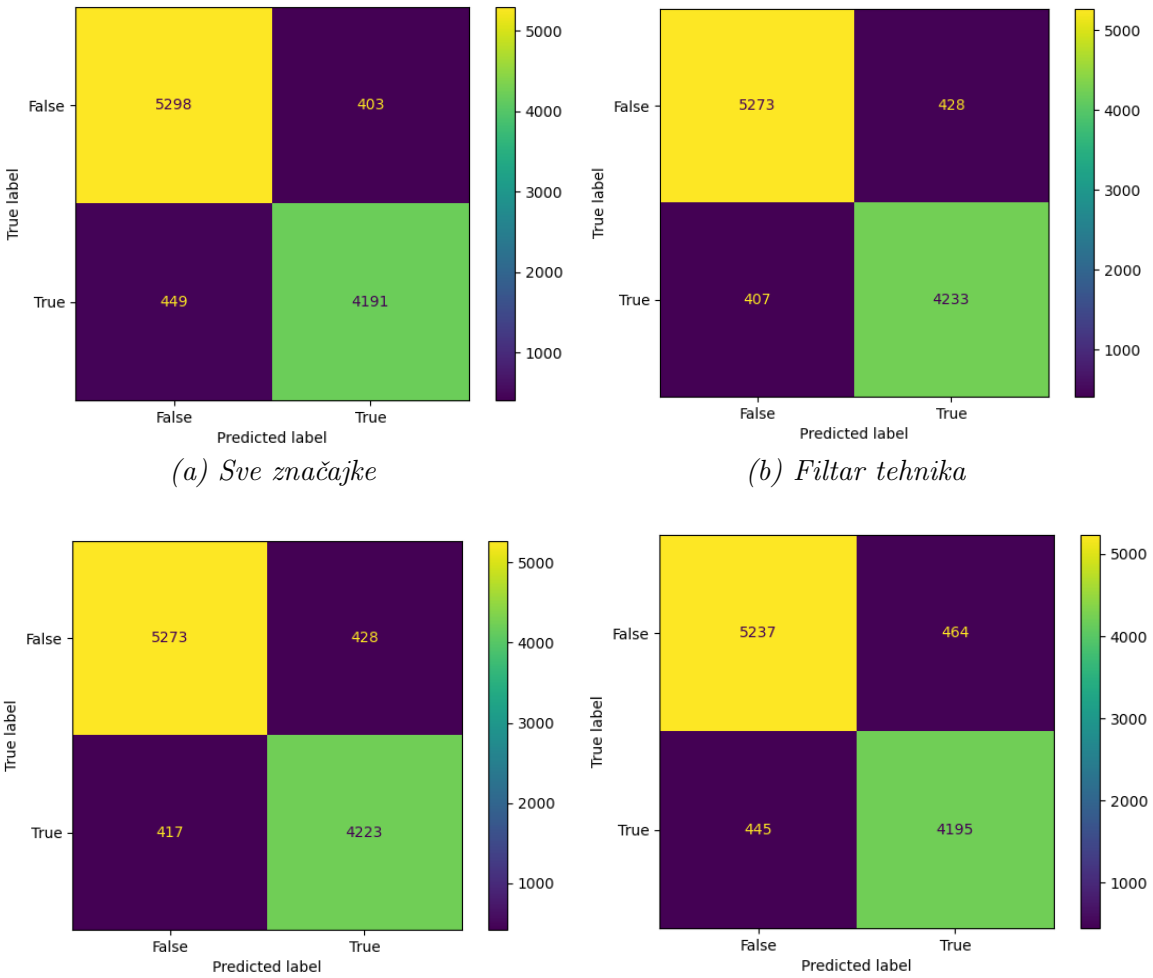
Vrijednosti dobivenih metrika strojnog modela korištenjem AMP-a prikazane su u tablici 6.2. Najbolje vrijednosti modela dobivene su korištenjem filtar tehnike, međutim u usporedbi s *forward* tehnikom ostvareni su slični rezultati. Korištenjem 291 značajke u vremenu unakrsne validacije od 579,78 sekundi nakon filtar tehnike, ostvarena je točnost od 91,9%, F1 rezultat od 91%, srednja geometrijska vrijednost od 0.91, preciznost od 90,8% te opoziv od 91,2%. Velika razlika očituje se u vremenu unakrsne validacije i to u razlici od 244,5 sekundi. Pritom, u obzir treba uzeti i vrijeme koje je korišteno u tim tehnikama za odabir značajki. *Forward* tehnika troši znatno više računalnih resursa pa je duže vrijeme unakrsne validacije nakon filtar tehnike ipak efikasnije. Rezultati modela nakon *forwarda* korištenjem 126 značajki u vremenu unakrsne validacije od 335,28 sekundi su točnost 91,8%, F1 rezultat 90,9%, srednja geometrijska vrijednost 0,909, preciznost 90,8% i opoziv 91 %. Ove vrijednosti razlikuju se naspram filtar tehnike za 0,001 te se jedino opoziv razlikuje za 0,002. Korištenjem svih značajki ostvaruje se minimalno lošiji rezultati u odnosu na model nakon filtar tehnike pri čemu je preciznost bolja za 0,004, ali vrijeme unakrsne validacije traje 1054,82 sekunde, što ga čini ne efikasnim. S druge strane, i dalje je efikasniji od modela koji koriste tehnike za odabir značajki. Najslabiji rezultati dobiveni su nakon *backward* tehnike korištenjem samo 45 značajki u vremenu unakrsne validacije u trajanju od 187,53 sekunde te iznose točnost 91,2%, F1 rezultat 90,2%, srednja geometrijska vrijednost 0,902, preciznost 90% i opoziv 90,4%. Razlika modela

u odnosu na model nakon primjene filtar tehnike je također u trećoj decimali koja iznosi 0,008 te je jedino točnost manja za 0,007. Sve vrijednosti se većinom razlikuju u trećoj decimali što je manje od jedan posto zbog čega te razlike ne predstavljaju veliki problem.

Tablica 6.2 Mjerni rezultati strojnog učenja korištenjem antimikrobnih peptida

Odabir značajki	Nema	Filtar	Omotač *forward*	Omotač *backward*
Broj značajki	1087	291	126	45
Vrijeme odabira	00:00:00	00:03:33,88	04:42:17,79	2 D, 10:46:08,57
Vrijeme validacije	00:17:34,82	00:09:39,78	00:05:35,28	00:03:07,53
Točnost	0,918	0,919	0,918	0,912
Preciznost	0,912	0,908	0,908	0,900
Opoziv	0,903	0,912	0,910	0,904
F1 rezultat	0,908	0,910	0,909	0,902
G-mean	0,908	0,910	0,909	0,902
ROC-AUC	0,975	0,977	0,975	0,974

Točna predviđanje na slici 6.8 prikazuju da modeli, kao i kod katalitičkih peptida, imaju visoku stopu točnih predviđanja AMP-a. Sva četiri modela imaju približno sličan postotak pogrešnih predviđanja koji je manji od deset posto, što je i dalje zadovoljavajuće ako se uzme u obzir veličina ulaznog skupa podataka. Model nakon filtar tehnike, napravio je točno predviđanje za 9506, a pogriješio za 835 zapisa. U prilog ovoj činjenici ide i to što algoritam Slučajne šume ima vrlo dobro predviđanje kod klasifikacijskih problema. Model nakon primjene *forward* tehnike ima približno točno predviđanje kao i filtar. Iznimka je krivo napravljena za 417 peptida čija je točna klasifikacija jedan odnosno aktivno stanje, a model previdio nula to jest neaktivano stanje. Razlika iznosi 10 peptida u odnosu na filtar tehniku. Najlošiji rezultati ostvareni su korištenjem *backward* tehnike, kod koje je pogrešno klasificirano ukupno 909 peptida. Za 445 peptida klasificiranih kao aktivni, model je napravio predviđanje da su neaktivni. S druge strane, za 464 peptida klasificiranih kao neaktivno stanje, model je proglasio aktivnim. Nadalje, napravljeno je točno predviđanje za 9432 peptida. Usporedno gledajući rezultate svih modela nakon primjene tehnika za odabir značajki i modela koji je koristio sve značajke, vidljivo je da je ostvaren manji broj točnih predviđanja za 4191 aktivnih peptida, a veći broj točnih predviđanja za 5298 neaktivnih peptida.

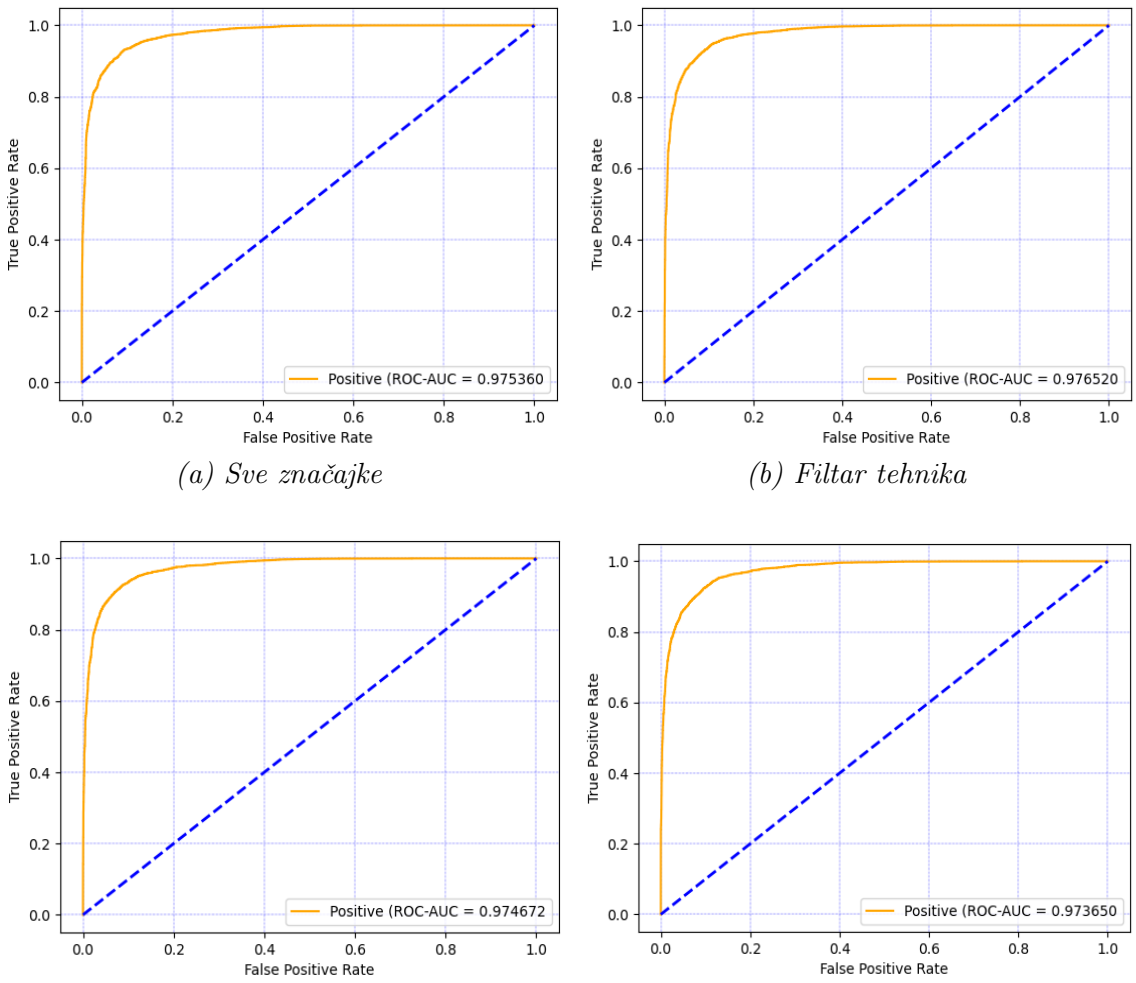


(a) Sve značajke *(b) Filtar tehnika*

(c) Omotač forward *(d) Omotač backward*

Slika 6.8 Matrica zabune strojnog modela kod antimikrobnih peptida

ROC-AUC vrijednosti za AMP prikazane su na slici 6.9. Sve vrijednosti prikazuju vrlo dobro predviđanje strojnog modela u iznosu od 97% te se neznatne razlike očituju u trećoj decimali. Temeljem tih vrijednosti, sve tehnike imaju isti efekt na model. Jedino se može izdvojiti filtar tehnika s iznosom od 97,65%.



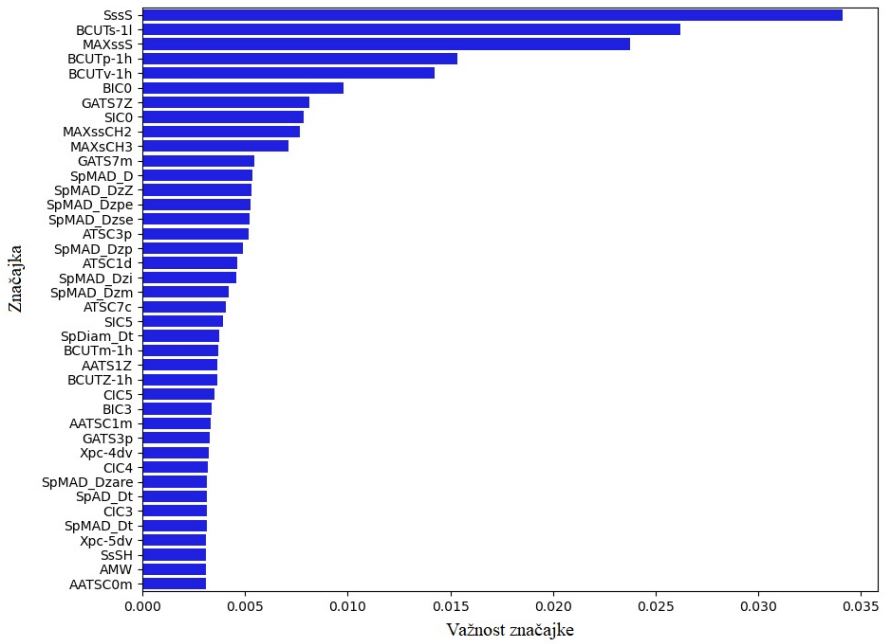
(a) Sve značajke *(b) Filtar tehnika*

(c) Omotač forward *(d) Omotač backward*

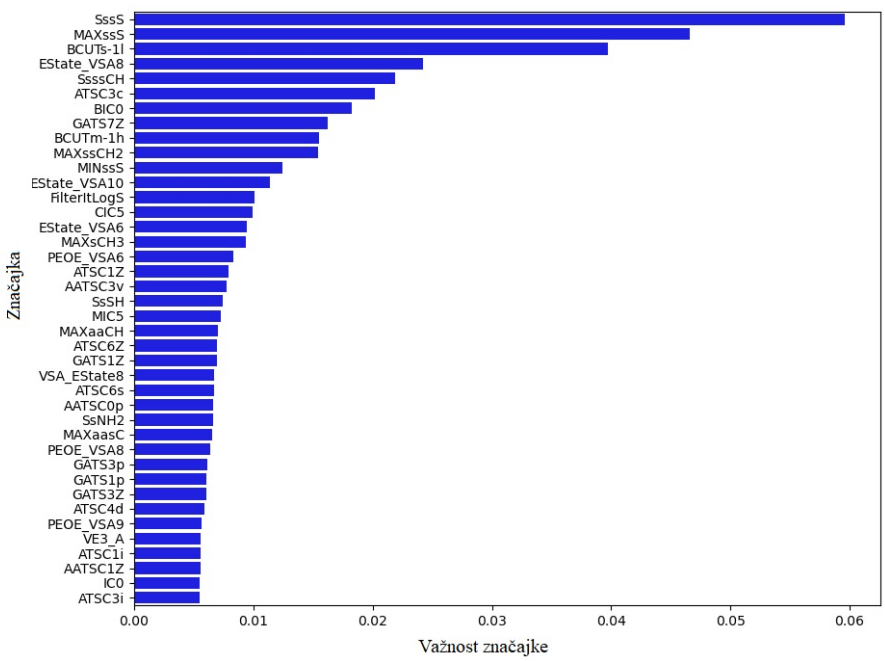
Slika 6.9 Grafički prikaz ROC-AUC vrijednosti strojnog modela kod antimikrobnih peptida

Na slici 6.10 prikazano je četrdeset najrelevantnijih značajki i njihove vrijednosti Gini nečistoće koje je algoritam Slučajne šume sam odredio za AMP. Postoje preklapanja u nekim značajkama poput SssS, BCTUs-1l, MAXssS međutim, to ne može garantirati da su te značajke presudne za predviđanje klasifikacije na izlazu odnosno ostvarivanje maksimalne homogenosti u čvoru. Međutim, svakako treba značajke koje se ponavljaju u prvih četrdeset gledati kao potencijalno važne informacije. Valja primijetiti da je prvih pet značajki jako bitno za ostvarivanje visokog stupnja

homogenosti u čvoru kojom se utječe i na visoku točnost predviđanja, a što se može zaključiti temeljem vrijednosti na x osi. Nakon tih značajki, vrijednosti za ostale značajke su većinom približno iste. Najveća vrijednost značajki dobivena je kod modela nakon korištenja omotač *backward* pri čemu veliki značaj ima korištenje najmanjeg broja značajki u odnosu na ostale modele.

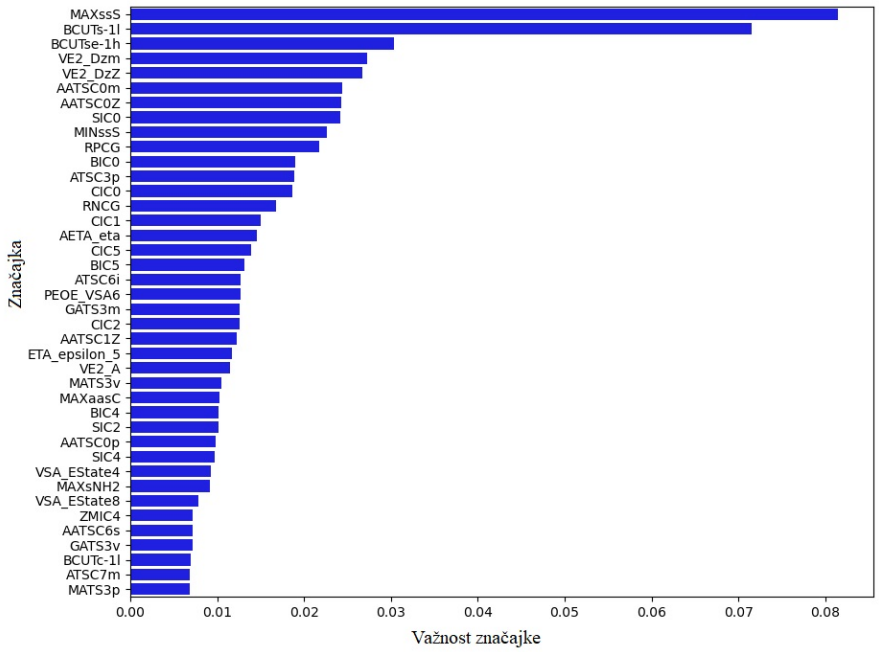


(a) Sve značajke

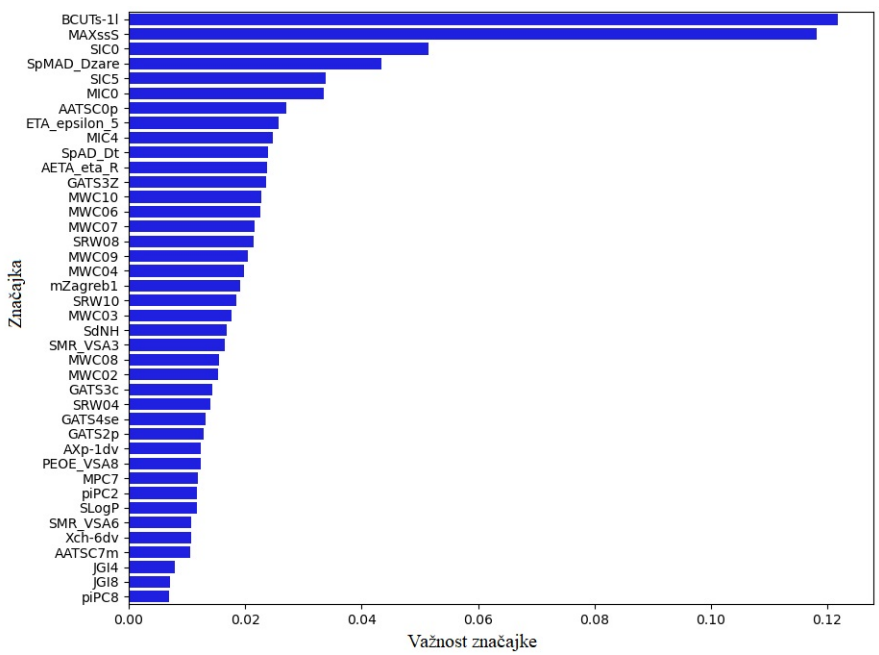


(b) Filtar tehnika

Slika 6.10 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod katalitičkih peptida - 1. dio



(c) Omotač forward



(d) Omotač backward

Slika 6.10 Grafički prikaz važnosti značajki određenih od strane strojnog modela kod antimikrobnih peptida - 2. dio

Uzevši u obzir dobivene rezultate i vrijeme za odabir značajki, filtar tehnika je najbolji izbor i u skupu antimikrobnih peptida. Činjenica da se ovaj skup podataka sastoji od velikog broja zapisa pokazuje da su statističke tehnike zasigurno efikasniji odabir s gledišta računalnih resursa. Točnost, F1 rezultat i ROC-AUC vrijednost prikazuju da model nakon ove tehnike ostvaruje najbolje rezultate u točnom predviđanju iako vrijeme unakrsne validacije traje duže nego kod omotač tehnika. Također, ima najveći broj točnih predviđanja u odnosu na druge modele. Model nakon primjene *forward* i *backward* ima kraće trajanje unakrsne validacije te su rezultati neznatno lošiji. S gledišta broja krivih predviđanja, model nakon primjene *backward* tehnike je ipak najlošiji izbor. Ukoliko se koriste sve značajke, model se drastično usporava u procesu unakrsne validacije, iako u konačnici rezultira sličnim kvalitetama u predviđanju. Zaključno, svi rezultati su podjednako dobri što činjenično pokazuje da algoritam Slučajne šume odlično radi predviđanje na malom i na velikom skupu podataka. Vrijednosti se neznatno razlikuju u trećoj decimali što s gledišta točnosti u predviđanju ne igra veliku ulogu.

Poglavlje 7

Zaključak

Upotreba strojnog učenja u bioinformatici sve je zastupljenija metoda. Razlog tome su brže otkrivanje novih kemijskih i spoznaja bioloških procesa. Definiranje peptida u formatu SMILES inovativan je pristup koji omogućuje otkrivanje novih zavisnosti između značajki. U ovom projektu analizirani su podaci antimikrobnih i katalitičkih peptida odnosno njihovih značajki. Problem koji se pri tome javlja je preveliki broj značajki koje su često višak zbog kojeg je potrebno napraviti predobradu podataka.

Čišćenje podataka brisanjem krivih vrijednosti ili NULL vrijednosti osnovni je pristup. Broj uklonjenih značajki koje su imale konstante ili nedostajuće vrijednosti u katalitičkom skupu iznosi 408 dok u antimikrobnom skupu iznosi 407. Također, uklonjene su i značajke koje su imale *overflow* vrijednosti te ih je u skupu katalitičkih i antimikrobnih peptida bilo 10. Nakon prvotnog brisanja značajki primjenjuje se redukcija značajki. Najčešće se koriste omotač i filtar tehnika. Međusobno ih razlikuje način implementacije. Omotač tehnike koriste strojni model kod vrendovanja, a filtar tehnike ga ne koriste. Najbitnija je razlika u vremenskoj složenosti koja se drastično primjećuje povećanjem skupa podataka. Strojni se modeli ovakvom vrstom predobrade podataka oslobađaju dodatnih informacija i potencijalnih krivih zaključaka.

U ovom radu analizirana je usporedba performansi odabira značajki, vrednovanje

modela pomoću svih značajki te redukcija filtar tehnikom, omotač *forward* i omotač *backward* tehnikom. Cijena korištenja *backward* tehnike pokazala se izuzetno neisplativa metoda zbog velikog troška koji predstavljaju računalni resursi. Naime, vremensko izvođenje kod AMP-a traje 2 dana, 10:46:08,57 sekundi, a kod katalitičkih iznosi 00:34:42,99 sekundi. Model koji je dobio novi skup značajki imao je približno dobre performanse kao i filtar tehnika koja je vremenski manje zahtjevna. Analizom svih metrika, filtar tehnika pokazala se najboljim izborom zbog bržeg odabira značajki te preciznijeg vršenja klasifikacije. U katalitičkim podacima bile su potrebne 00:00:38,64 sekunde za odabir relevantnih značajki, dok su AMP skupu podataka bile potrebne 00:03:33,88 sekundi. *Forward* tehnike pokazale su se također vrlo dobrim izborom. Vremenski su manje zahtjevne operacije iako i dalje traju duže od filtar tehnika. Nadalje, dobar pristup je i korištenje svih značajki, no ono nije preporučljivo zbog šuma u podacima.

Algoritam Slučajne šume pokazuje odlične performanse u klasifikacijskim problemima. Svakako, parametriziranje ima visok učinak u dobrom predviđanju. Nakon svih tehnika odabira značajki, primijećen je odličan F1 rezultat koji je uvijek iznad 0.90. Također, ROC-AUC pokazuje izvrsne rezultate, gdje je nakon primjene filtar tehnike nad katalitičkim peptidima ostvarena vrijednost od 93,9%, a kod AMP-a čak 97,7%. Modeli nakon primjene *forward* i *backward* odabira značajki ostvaruju slabije rezultate. Što se tiče algoritamske implementacije modela koja se koristi kod omotač tehnika, *Gaussian Naive Bayes* u *forwardu* se pokazao kao vremenski brz algoritam. S druge strane, *backward* tehnika upotrebom Stabla odluke pokazuje veću vremensku složenost. Gledajući rezultate i vrijeme, efikasnija je *forward* pretraga u omotač tehnici.

U budućnosti je potrebno empirijski istražiti kemijsku povezanost značajki kod filtar tehnike kako bi se dobio bolji uvid u odluke strojnog modela. Naime, značajke uglavnom nemaju preklapanje u važnosti definiranoj od strane algoritma Slučajne šume. Unatoč tome, strojno učenje pokazuje odlično predviđanje pri čemu je nužno uzeti u obzir žrtvu računalnih resursa.

Literatura

[1] Rapič, V.; Kovačević, M.: „III. Organometalna i bioorganometalna kemija - ferocenski peptidi", Kemija u industriji : časopis kemičara i tehnologa Hrvatske, Vol. 61, No. 2, pp. 71–120, 2012.

[2] Hu, G.: „Understanding the fundamentals of peptides and proteins", BioProcessing Journal, Vol. 10, No. 1, pp. 12–14, 2011.

[3] Zhao, X.; Pan, F.; Lu, J. R.: „Recent development of peptide self-assembly", Progress in Natural Science, Vol. 18, No. 6, pp. 653–660, 2008.

[4] Al-Azzam, S; Ding, Y.; Liu J.; Pandya P.; Ting, J. P.; Afshar, S.: „Peptides to combat viral infectious diseases", Peptides, Vol. 134, pp. 1-15, 2020.

[5] Kamerlin, S. C.; Warshel A.: „At the dawn of the 21st century: Is dynamics the missing link for understanding enzyme catalysis?", Proteins: Structure, Function, and Bioinformatics, Vol. 78, No. 6, pp. 1339–1375, 2010.

[6] Maeda, Y.; Makhlynets, O. V.; Matsui, H.; Korendovych, I. V.: „Design of catalytic peptides and proteins through rational and combinatorial approaches", Annual review of biomedical engineering, Vol. 18, No. 1, pp. 311-328, 2016.

[7] Izadpanah, A.; Gallo, R. L.: „Antimicrobial peptides", Journal of the American Academy of Dermatology, Vol. 52, No. 3, pp. 381–390, 2005.

[8] Zhang, L.-J.; Gallo, R. L.: „Antimicrobial peptides", Current Biology, Vol. 26, No. 1, pp. R14–R19, 2016.

[9] Fjell, C. D.; Hiss, J. A.; Hancock, R. E.; Schneider, G.: „Designing antimicrobial peptides: form follows function", Nature reviews Drug discovery, Vol. 11, No. 1, pp. 37–51, 2012.

[10] Minkiewicz, P.; Darewicz, M.; Iwaniak, A.; Turło, M.: „Proposal of the annotation of phosphorylated amino acids and peptides using biological and chemical codes", Molecules, Vol. 26, No. 3, pp. 712-726, 2021.

[11] Minkiewicz, P.; Iwaniak, A.; Darewicz, M.: „Annotation of peptide structures using smiles and other chemical codes–practical solutions", Molecules, Vol. 22, No. 12, pp. 2075-2091, 2017.

[12] „The Twenty Amino Acids", s Interneta, http://www.cryst.bbk.ac.uk/education/AminoAcid/the_twenty.html, 10. kolovoza 2022.

[13] Weininger, D.: „Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules", Journal of chemical information and computer sciences, Vol. 28, No. 1, pp. 31–36, 1988.

[14] „OpenSMILES Home Page", s Interneta, http://opensmiles.org/, 21. srpnja 2022.

[15] Sessa, J.; Syed, D.: „Techniques to deal with missing data", IEEE 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), pp. 1–4, 2016.

[16] Singh, S. K.; Dwivedi, R. K.: „Data mining: dirty data and data cleaning", International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications (ICAISC-2020), pp. 1-5, 2020.

[17] Kumar, V.; Khosla, C.: „Data cleaning-A thorough analysis and survey on unstructured data", 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 305–309, 2018.

[18] Rahm, E.; Do, H. H.: „Data cleaning: Problems and current approaches", IEEE Data Eng. Bull., Vol. 23, No. 4, pp. 3–13, 2000.

[19] Chu, X.; Ilyas, I. F.: „Data Cleaning", ACM/Association for Computing Machinery, New York, 2019.

[20] Chandola, V.; Banerjee, A.; Kumar, V.: „Anomaly detection: A survey", ACM Computing Surveys, Vol. 41, No. 3, pp. 1–58, 2009.

[21] Wang, H.; Bah, M. J.; Hammad, M.: „Progress in outlier detection techniques:A survey", IEEE Access, Vol. 7, pp. 107 964–108 000, 2019.

[22] Trittenbach, H.: „User-centric active learning for outlier detection", Ph.D. disertacija, Karlsruhe Institute of Technology, Njemačka, 2020, s Interneta, https://publikationen.bibliothek.kit.edu/1000117443, 20. kolovoza 2022.

[23] Orair, G. H.; Teixeira, C. H.; Meira Jr, W.; Wang, Y.; Parthasarathy, S.: „Distance-based outlier detection: consolidation and renewed bearing", Proceedings of the VLDB Endowment, Vol. 3, No. 1-2, pp. 1469–1480, 2010.

[24] Osborne, J. W.: „Best practices in quantitative methods", Sage, Thousands Oaks, Kalifornija, 2008.

[25] Fatima, A.; Nazir, N.; Khan, M. G.: „Data cleaning in data warehouse: A survey of data pre-processing techniques and tools", International Journal of Information Technology and Computer Science, Vol. 9, No. 3, pp. 50–61, 2017.

[26] Eesa, A. S.; Arabo, W. K.: „A Normalization Methods for Backpropagation: A Comparative Study", Science Journal of University of Zakho, Vol. 5, No. 4, pp. 319–323, 2017.

[27] Xue, B.; Zhang, M.; Browne, W. N.; Yao, X.: „A Survey on Evolutionary Computation Approaches to Feature Selection", IEEE Transactions on Evolutionary Computation, Vol. 20, No. 4, pp. 606–626, 2016.

[28] Hancer, E.; Xue, B.; Zhang, M.: „Differential evolution for filter feature selection based on information theory and feature ranking", Knowledge-Based Systems, Vol. 140, No. C, pp. 103–119, 2018.

[29] Wang, S.; Tang, J.; Liu, H.: „Feature selection", Encyclopedia of Machine Learning and Data Mining, Webb, G. I.; Sammut, C., Berlin, Njemačka, Springer, 2017, pp. 1-9.

[30] John, G. H.; Kohavi, R.; Pfleger, K.: „Irrelevant features and the subset selection problem", Machine Learning: Proceedings of the Eleventh International Conference, pp. 121–129, 1994.

[31] Mlambo, N.; Cheruiyot, W. K.; Kimwele, M. W.: „A survey and comparative study of filter and wrapper feature selection techniques", The International Journal Of Engineering And Science, Vol. 5, No. 8, pp. 57–67, 2016.

[32] Cai, J.; Luo, J.; Wang, S.; Yang, S.: „Feature selection in machine learning: A new perspective", Neurocomputing, Vol. 300, pp. 70–79, 2018.

[33] Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; Liu, H.: „Feature selection: A data perspective", ACM Computing Surveys (CSUR), Vol. 50, No. 6, pp. 1–45, 2017.

[34] Tang, J.; Alelyani, S.; Liu, H.: „Feature selection for classification: A review", Data classification: Algorithms and applications, Aggarwal, C. C., New York, CRC press, 2013, ch. 2, pp. 37-64.

[35] Dash, M.; Liu, H.: „Feature selection for classification", Intelligent data analysis, Vol. 1, No. 1-4, pp. 131–156, 1997.

[36] Venkatesh, B.; Anuradha, J.: „A review of feature selection and its methods", Cybernetics and Information Technologies, Vol. 19, No. 1, pp. 3–26, 2019.

[37] Saeys, Y.; Inza, I.; Larranaga, P.: „A review of feature selection techniques in bioinformatics", Bioinformatics, Vol. 23, No. 19, pp. 2507–2517, 2007.

[38] Cherrington, M.; Thabtah, F.; Lu, J.; Xu, Q.: „Feature selection: filter methods performance challenges", 2019 International Conference on Computer and Information Sciences (ICCIS), pp. 1–4, 2019.

[39] Yu, L.; Liu, H.: „Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution", Proceedings of the 20th international conference on machine learning (ICML-03), Vol. 2, pp. 856–863, 2003.

[40] Kira, K.; Rendell, L. A.; et al.: „The feature selection problem: Traditional methods and a new algorithm", AAAI-92: Proceedings of the tenth national conference on Artificial intelligence, Vol. 2, pp. 129–134, 1992.

[41] Brownlee, J.: „Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python", Machine Learning Mastery, 2020.

[42] Jinyuan, L.; Wan, T.; Guanqin, C.; Yin, L.; Changyong, F.: „Correlation and agreement: overview and clarification of competing concepts and measures", Shanghai archives of psychiatry, Vol. 28, No. 2, pp. 115-120, 2016.

[43] Hamed, K. H.: „The distribution of Kendall's tau for testing the significance of cross-correlation in persistent data", Hydrological Sciences Journal, Vol. 56, No. 5, pp.841–853, 2011.

[44] Dehling, H.; Vogel, D.; Wendler, M.; Wied, D.: „Testing for changes in Kendall's tau", Econometric Theory, Vol. 33, No. 6, pp. 1352–1386, 2017.

[45] Long, J. D.; Cliff, N.: „Confidence intervals for kendall's tau", British Journal of Mathematical and Statistical Psychology, Vol. 50, No. 1, pp. 31–41, 1997.

[46] Carterette, B.: „On rank correlation and the distance between rankings", Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09), pp. 436–443, 2009.

[47] Lapata, M.: „Automatic evaluation of information ordering: Kendall's tau", Computational Linguistics, Vol. 32, No. 4, pp. 471–484, 2006.

[48] Brossart, D. F.; Laird, V. C.; Armstrong, T. W.: „Interpreting Kendall's Tau and Tau-U for single-case experimental designs", Cogent Psychology, Vol. 5, No. 1, pp. 1-26, 2018.

[49] Das, S.: „Filters, wrappers and a boosting-based hybrid for feature selection", ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning, Vol. 1, pp. 74–81, 2001.

[50] Solorio-Fernández, S.; Carrasco-Ochoa, J. A.; Martínez-Trinidad, J. F.: „A review of unsupervised feature selection methods", Artificial Intelligence Review, Vol. 53, No. 2, pp. 907–948, 2020.

[51] Kohavi, R.; John, G. H.: „Wrappers for feature subset selection", Artificial intelligence, Vol. 97, No. 1-2, pp. 273–324, 1997.

[52] Wang, A.; An, N.; Chen, G.; Yang, J.; Li, L.; Alterovitz, G.: „Incremental wrapper based gene selection with Markov blanket", 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 74–79, 2014.

[53] Wang, A.; An, N.; Chen, G.; Li, L.; Alterovitz, G.: „Accelerating wrapper-based feature selection with K-nearest-neighbor", Knowledge-Based Systems, Vol. 83, pp. 81–91, 2015.

[54] El Naqa, I.; Murphy, M. J.: „What is machine learning?", Machine Learning in Radiation Oncology: Theory and Applications, Naqa I. E., Li R., Murphy M. J., Švicarska, Springer, 2015, pp. 3–11.

[55] Mahesh, B.: „Machine learning algorithms-a review", International Journal of Science and Research (IJSR), Vol. 9, No. 1 pp. 381–386, 2020.

[56] Nasteski, V. : „An overview of the supervised machine learning methods", Horizons. b, Vol. 4, pp. 51–62, 2017.

[57] Zhou, Z.-H.: „Machine learning", Springer, Singapur, 2021.

[58] Vanwinckelen, G.; Blockeel, H.: „On estimating model accuracy with repeated cross-validation", BeneLearn 2012 : proceedings of the 21st Belgian-Dutch conference on machine learning, pp. 39–44, 2012.

[59] Hossin, M.; Sulaiman, M. N.: „A review on evaluation metrics for data classification evaluations", International journal of data mining & knowledge management process, Vol. 5, No. 2, pp. 1-11, 2015.

[60] Visa, S.; Ramsay, B.; Ralescu, A. L.; Van Der Knaap, E.: „Confusion matrix-based feature selection", Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference 2011, Vol. 710, No. 1, pp. 120–127, 2011.

[61] Novaković, J. D.; Veljović, A.; Ilić, S. S.; Papić, Ž.; Milica, T.: „Evaluation of classification models in machine learning", Theory and Applications of Mathematics & Computer Science, Vol. 7, No. 1, pp. 39–46, 2017.

[62] Japkowicz, N.: „Why question machine learning evaluation methods", AAAI workshop on evaluation methods for machine learning, pp. 6-11, 2006.

[63] Akosa, J. S.: „Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data", SAS Global Forum 2017, Vol. 942, pp. 1-11, 2017.

[64] Vujović, Ž. Đ.: „Classification model evaluation metrics", International Journal of Advanced Computer Science and Applications, Vol. 12, No. 6, pp. 599–606, 2021.

[65] Fawcett, T.: „ROC Graphs: Notes and Practical Considerations for Researchers", Machine learning, Vol. 31, No. 1, pp. 1-38, 2004.

[66] Davis, J.; Goadrich, M.: „The relationship between Precision-Recall and ROC curves", Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240, 2006.

[67] Gupta, B.; Rawat, A.; Jain, A.; Arora, A.; Dhami, N.: „Analysis of various decision tree algorithms for classification in data mining", International Journal of Computer Applications, Vol. 163, No. 8, pp. 15–19, 2017.

[68] Song, Y.-Y.; Ying, L.: „Decision tree methods: applications for classification and prediction", Shanghai archives of psychiatry, Vol. 27, No. 2, pp. 130-135, 2015, s Interneta, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/, 02. kolovoza 2022.

[69] Suthaharan, S.: „Decision tree learning", Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning,Suthaharan, S, Springer, Boston, Massachusetts, SAD, 2016, pp. 237–269.

[70] Kotsiantis, S. B.: „Decision trees: a recent overview", Artificial Intelligence Review, Vol. 39, No. 4, pp. 261–283, 2013.

[71] Myles, A. J.; Feudale, R. N.; Liu, Y.; Woody, N. A.; Brown, S. D.: „An introduction to decision tree modeling", Journal of Chemometrics, Vol. 18, No. 6, pp. 275–285, 2004.

[72] Liu, Y.: „Random forest algorithm in big data environment", Computer modelling & new technologies, Vol. 18, No. 12A, pp. 147–151, 2014.

[73] Livingston, F.: „Implementation of Breiman's random forest machine learning algorithm", s Interneta, https://datajobs.com/data-science-repo/Random-Forest-[Frederick-Livingston].pdf, 10. kolovoza 2022.

[74] Rogers, J.; Gunn, S.: „Identifying feature relevance using a random forest", SLSFS'05: Proceedings of the 2005 international conference on Subspace, Latent Structure and Feature Selection, pp. 173–184, 2005.

[75] Zhang, C.; Ma, Y.: „Ensemble machine learning: methods and applications", Springer, New York, SAD, 2012.

[76] Stern, M.; Beck, J.; Woolf, B. P.: „Naive Bayes classifiers for user modeling", Center for Knowledge Communication, Computer Science Department, University of Massachusetts, SAD, 1999.

[77] Larsen, K.: „Generalized Naive Bayes Classifiers", ACM SIGKDD Explorations Newsletter, Vol. 7, No. 1, pp. 76–81, 2005.

[78] Jahromi, A. H.; Taheri, M.: „A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features", 2017 Artificial intelligence and signal processing conference (AISP), pp. 209–212, 2017.

[79] „ECRL/padelpy: A python wrapper for Padel-Descriptor Software", s Interneta, https://github.com/ecrl/padelpy, 03.kolovoza, 2022.

[80] „mordred-descriptor/mordred: Mordred", s Interneta, https://github.com/mordred-descriptor/mordred, 03. kolovoza 2022.

[81] Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T.: „Mordred: a molecular descriptor calculator", Journal of Cheminformatics, Vol. 10, No. 1, pp. 1–14, 2018.

[82] „PaDEL-Descriptor", s Interneta, http://www.yapcwsoft.com/dd/padeldescriptor/, 03. kolovoza 2022.

Pojmovnik

ACC Accuracy (Točnost). 25

AMP Antimicrobial peptides (Antimikrobni peptidi). 4

AUC Area under curve (Površina ispod krivulje). 1

CSV Comma-separated values (Zarezom odvojene vrijednosti). 33

FN False Negative (Neispravno negativno). 25

FP False Positive (Neispravno pozitivno). 25

G-mean Geometric mean value (Srednja geometrijska vrijednost). 26

k-NN k-Nearest Neighbor (k-najbližih susjeda). 12

Pr Precision (Preciznost). 26

ROC Receiver Operating Characteristic (Radna karakteristika prijemnika). 1

SMILES Smplified Molecular Input Line Entry System (Linijski sustav pojednostavljenog molekularnog unosa). 1

TN True Negative (Ispravno negativno). 25

TP True Positive (Ispravno pozitivno). 25

TPR Recall (Opoziv). 26

Sažetak

U današnje vrijeme dostupne su velike količine informacija koje je potrebno smanjiti i odabrati samo one koje su bitne za donošenje zaključaka ili predviđanje budućeg stanja. U ovom radu analiziraju se teorijski i implementacijski glavne tehnike u predobradi podataka te postupci strojnog učenja u primjeni za predviđanje katalitičke i antimikrobne aktivnosti peptida. Skup podataka dobiven je pretvorbom zapisa peptida iz formata FASTA u format SMILES, nakon čega su izračunate značajke kemijske strukture. Napravljena je komparativna analiza filtar tehnika i tehnika omotač s posebnim osvrtom na *forward* i *backward*, pritom uspoređujući vremenske performanse i broj odabranih značajki. Model strojnog učenja temeljem algoritma Slučajna šuma treniran je raznim skupovima podataka pri čemu najbolje metrike ostvaruje nakon filtar tehnike uzevši u obzir ROC-AUC i F1 rezultate. U katalitičkom skupu podataka one iznose 0,939 i 0,961, a u antimikrobnom skupu iznose 0,977 i 0,910. Ovim radom demonstrirana je učinkovitost tehnike filtra sa stajališta trošenja računalnih resursa.

Ključne riječi **— filtar tehnika, tehnika omotač, strojno učenje, Slučajna šuma, peptidi, format SMILES**

Abstract

Nowadays, large amounts of information are available and there is a growing need to reduce and sort through them in order to make valid conclusions or predict future states. This thesis analyses the main theoretical and implementational techniques in data preprocessing and machine learning procedures applied in predicting the catalytic and antimicrobial activity of peptides. The data set was obtained by converting peptides from the FASTA to the SMILES format, upon which chemical structure features were calculated. A comparative analysis of the filter and wrapper techniques was made with special reference to forward and backward, whilst comparing time performance and selected feature numbers. The machine learning model based on the Random Forest algorithm was then trained with various data sets whereby the best metrics was achieved after the filter technique, taking into

consideration the ROC-AUC and F1 results. In the catalytic data set they amount to 0,939 and 0,961, and in the antimicrobial set they amount to 0,977 and 0,910. This thesis demonstrates the efficiency of the filter technique from the perspective of computer resource spending.

Keywords **— filter technique, wrapper technique, machine learning, Random Forest, peptides, SMILES format**