# Model predviđanja antimikrobnih peptida

**Erjavac, Ivan**

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Preddiplomski sveučilišni studij računarstva

Završni rad

# MODEL PREDVIĐANJA ANTIMIKROBNIH PEPTIDA

Ivan Erjavac

0069078568

Rijeka, srpanj 2020.

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Preddiplomski sveučilišni studij računarstva

Završni rad

# MODEL PREDVIĐANJA ANTIMIKROBNIH PEPTIDA

Mentor: doc. dr. sc. Goran Mauša

Ivan Erjavac

0069078568

Rijeka, srpanj 2020.

## Izjava o samostalnosti

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku preddiplomskih sveučilišnih studija, izjavljujem da sam ovaj rad izradio samostalno primjenjujući stečeno znanje, uz stručnu pomoć mentora doc. dr. sc. Goran Mauša.

Rijeka, rujan 2020.

_____

Ivan Erjavac

# Zahvala

Zahvaljujem se svom mentoru doc. dr. sc. Goranu Mauši na savjetima oko izrade ovog završnog rada, kao i na svoj pomoći oko istraživanja teme strojnog učenja.
Zahvaljujem se Anamariji Ljutić na lektoriranju ovog završnog rada.

# Sadržaj

# 1. Uvod

Antiobiotici u današnjem svijetu medicine igraju značajnu ulogu. Jedna su od najvažnijih vrsta lijekova koji se koriste u današnjoj modernoj medicini [1]. Koriste se za liječenje, ali i sprječavanje širokog spektra bakterijskih infekcija. Čak i neke relativno bezopasne svakodnevne pojave, poput porezotine ili ugriza psa, uslijed infekcije mogu potencijalno biti smrtonosne, ukoliko se ne liječe antibioticima [2]. Također, veliki se napori ulažu u sprječavanje takozvanih intrahospitalnih infekcija, vrste infekcija do kojih može doći u bolnicama tijekom ili nakon operativnih zahvata [3]. Nadalje, bakterije koje su uzrok ove vrste infekcija, postaju sve otpornije na današnje antibiotike, što današnju medicinu čini još motiviranijom za otkrivanje novih, potentnijih antibiotika. Stoga se može zaključiti da bi ljudsko zdravlje bilo kudikamo ugroženije bez postojanja antibiotika.

Rad na antimikrobnim peptidima počeo je sredinom 20. stoljeća, a od tada su otkrivene mnoge primjene ove podvrste antibiotika [4]. Neke od ovih primjena su antibakterijske, antigljivične i antivirusne aktivnosti, no koriste se i kao protuupalna sredstva, a pokazuju i citotoksičnu aktivnost prema stanicama malignih tumora.

Budući da su antimikrobni peptidi prirodni proizvodi imunoloških sustava bića iz svih životinjskih carstava, kao i nekih biljaka, sama je količina antimikrobnih peptida koji se pojavljuju u prirodi enormna. Kako bi saznali koji bi od tih antimikrobnih peptida mogli biti primjenjivi u medicini, potrebno je testirati svaki od tih antimikrobnih peptida te zaključiti imaju li oni svoju primjenu. Od velikog broja antimikrobnih peptida koji se pojavljuju u prirodi, samo je mali dio njih iskoristiv u medicini, što dovodi do problema njihova pronalaska konvencionalnim putem.

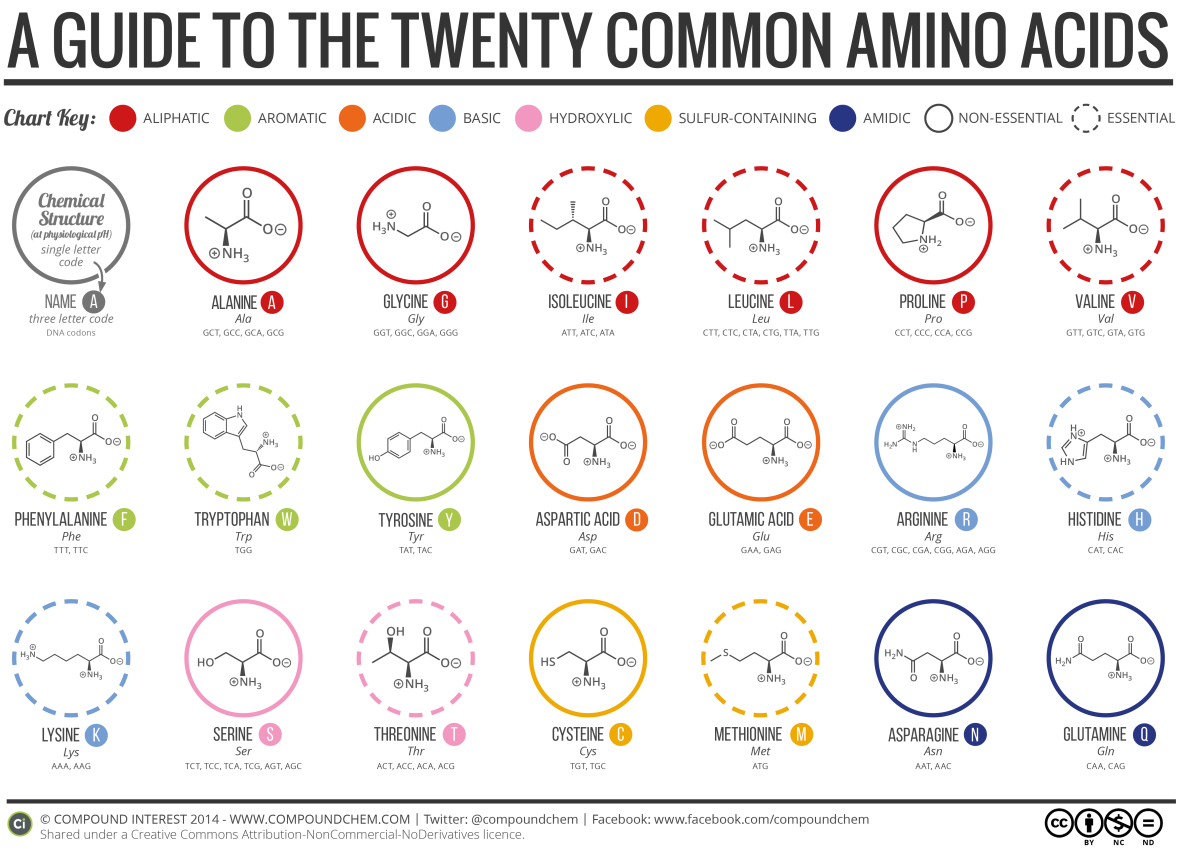
Upravo se u tom segmentu istraživanja krije temeljna motivacija upotrebe strojnog učenja na području antimikrobnih peptida, a to je ubrzanje cijelog procesa testiranja i otkrivanja primjena antimikrobnih peptida, smanjenje količine testiranja, te u konačnici, povećanje broja antimikrobnih peptida sa primjenom u medicini.

U ovome se radu predstavlja izgradnja modela koji vrši predviđanje antimikrobnog svojstva peptida. Objasniti će se proces izgradnje seta podataka, odabir modela i podešavanje njegovih hiperparametara, te će se na kraju prezentirati rezultati i zaključci. No prije toga, potrebno je definirati pojam antimikrobnih peptida i staviti problem u kontekst strojnog učenja.

# 2. Antimikrobni peptidi

## 2.1. Aminokiseline i proteini

Kako bismo definirali pojam peptida, prvo moramo definirati aminokiseline. Aminokiseline su organske molekule koje djeluju kao osnovna građevna jedinica peptida i proteina [5]. Lanac koji se sastoji od 2-50 aminokiselina tvori jedan peptid, dok lanac od 51 i više aminokiselina nazivamo proteinom. Što se tiče kemijskog sastava aminokiselina, sastavljeni su od 4 kemijska elementa: vodika, dušika, kisika i ugljika. U prirodi se može pronaći preko 500 različitih aminokiselina, no za ljudski organizam i genetski kod važno je samo njih 20. Njihove nazive, kratice i kemijske strukture prikazuje *Slika 2.1.*




*Slika 2.1. 20 osnovnih aminokiselina (preuzeto iz [6])*

U kontekstu ljudskog organizma, glavna je podjela ovih 20 osnovnih aminokiselina na esencijalne i neesencijalne aminokiseline [7]. Neesencijalne aminokiseline (alanin, aspargin, asparginska kiselina, cistein, glicin, glutamin, glutaminska kiselina, prolin, serin, tirozin) su one koje naše tijelo može sintetizirati prirodnim putem, dok esencijalne aminokiseline (arginin,

fenilalanin, histidin, izoleucin, leucin, lizin, metionin, treonin, triptofan, valin) ne može, te se moraju unositi u organizam drugim putem (prehranom, lijekovima, itd.).

Aminokiseline u ljudsko tijelo ulaze u obliku proteina i peptida, koje organizam razgrađuje na aminokiseline. Organizam zatim koristi te novostečene aminokiseline i povezuje ih u nove proteine, odnosno peptide. U konačnici, ti lanci aminokiselina grade ljudsko tijelo i povezuju se u krv, tkivo, neurotransmitere, hormone, enzime, kao i mnoge druge sastavne jedinice ljudskog tijela. Također, neki od tih proteina sudjeluju u razgradnji hrane, rastu i zacijeljivanju tkiva, a služe i kao izvor energije.

Od svih proteina i peptida koje naše tijelo sintetizira, za ovaj rad su najvažniji oni peptidi koji imaju antimikrobnu aktivnost, dakle, antimikrobni peptidi.

## 2.2. Izvori i podjela antimikrobnih peptida

Budući da su antimikrobni peptidi produkti imunoloških sustava živih bića, nije toliko teško pronaći proizvođače antimikrobnih peptida. U vrijeme pisanja ovog rada, identificirano je oko 140 ljudskih antimikrobnih peptida [8]. To znači da ih ljudsko tijelo prirodno sintetizira kako bi se obranilo od raznih mikroba. Pronađeni su u raznim dijelovima tijela, kao što su koža, oči, uši, želudac, crijeva, te u imunološkim, živčanim i urinarnim sustavima [9].

Svi organizmi koji proizvode antimikrobne peptide rade to na isti ili sličan način kao i ljudski organizam, što je opisano u poglavlju 2.1, dakle, rastavljenjem proteina na aminokiseline, te ponovno sastavljanje tih aminokiselina u nove proteine i peptide.

Jedna od podjela antimikrobnih peptida je po njihovom porijeklu. Prema referentnoj bazi antimikrobnih peptida za ovaj projekt *The Antimicrobial Peptide Database* (APD), antimikrobni peptidi su raspoređeni na sljedećih 6 carstava: bakterije, arheje, protisti, gljive, biljke i životinje [8, 10]. Prema APD, ukupan broj antimikrobnih peptida sa dokazanom djelotvornošću u vrijeme pisanja ovog rada iznosi 3198, a njihov broj po carstvima prikazuje *Tablica 2.1.*

*Tablica 2.1. Broj AMP-ova po carstvima (preuzeto iz [8])*

| Carstvo | Broj antimikrobnih peptida |
|---|---|
| Bakterije | 357 |
| Arheje | 5 |
| Protisti | 8 |
| Gljive | 20 |
| Biljke | 352 |
| Životinje | 2374 |

Preostali antimikrobni peptidi, njih 82, otkriveni su sintetičkim putem te se ne pojavljuju u prirodi [8].

## 2.3. Primjene antimikrobnih peptida

Još jedna podjela antimikrobnih peptida je prema njihovim upotrebama. Najraširenija upotreba antimikrobnih peptida se može iščitati iz njihovog naziva. To je borba protiv mikroba, dakle, antibiotska primjena. Neki od tih mikroba uključuju viruse, gljivice, gram-pozitivne i gram-negativne bakterije.

Jedan od takvih antimikrobnih peptida jest skvalamin [11]. Godine 1993. primjećeno je kako kostelj (*Squalus acanthias*), vrsta morskog psa, čak i u nehigijenskim uvjetima nikada ne podliježe infekcijama. Nakon daljnjeg istraživanja, iz tkiva kostelja izoliran je peptid koji je nazvan skvalamin, nakon čega je uspješno reproduciran sintetskim putem u laboratoriju. Raznim testiranjima dokazano je da je upravo skvalamin zaslužan za kosteljevu uspješnu antibakterijsku bitku protiv infekcija.

U modernoj medicini se skvalamin koristi kao antibiotik u borbi protiv bakterijskih infekcija [12]. Također, postoje indikacije da bi se skvalamin mogao koristiti kao pomoć kod očnih problema koji se javljaju kao posljedica dijabetesa, te u borbi protiv malignih tumora, što nas dovodi do sljedeće važne primjene antimikrobnih peptida.

Primjena antimikrobnih peptida možda može riješiti, ili barem ublažiti, jedan od vodećih zdravstvenih problema u svijetu - rak. Svaka šesta smrt u svijetu posljedica je nekog oblika ove zloćudne bolesti, što čini ovu bolest drugim vodećim uzrokom smrti na svijetu (prvo mjesto zauzimaju kardiovaskularne bolesti) [13]. Stoga nije teško povjerovati da se lijek za rak smatra jednim od glavnih ciljeva u modernoj medicini.

Podvrstu antimikrobnih peptida koji pokazuju aktivnost protiv stanica raka nazivamo antikancerogeni peptidi. Jedni od pripadnika antikancerogenih peptida su aureini [14]. Aureini su kratki peptidi koje izlučuju zrnate dorzalne žlijezde žaba *Litoria aurea* i *Litoria raniformis*, autohtone vrste Australije i Novog Zelanda. Iako su aureini primarno antibakterijski peptidi, razna testiranja dokazala su citotoksičnu aktivnost prema stanicama raka. U jednom od prvih istraživanja o aureinima, australski znantvenici otkrili su 22 različita aureina, od kojih je 13 pokazalo široku antibiotsku i antikancerogenu aktivnost. Još jedna od prednosti aureina, kao i svih ostalih antikancerogenih peptida, jest slaba citotoksičnost prema zdravim stanicama.

Uz antibiotsku i antikancerogenu aktivnost, antimikrobni peptidi imaju svojstva zacijeljivanja rana, antioksidantska svojstva, pokazuju antiparazitsku aktivnost, dok se neki AMP-ovi koriste i kao pesticidi u poljoprivredi.

Zbog svoje široke primjene i učinkovitosti, lako je zaključiti zašto su antimikrobni peptidi važna tema u modernoj medicini i biologiji, a uz daljnja istraživanja i napredak znanosti, područje antimikrobnih peptida će rasti, a antimikrobni peptidi će igrati sve veću ulogu u našim životima.

# 3. Strojno učenje

## 3.1. Definicija strojnog učenja

Strojno učenje (*eng. Machine Learning,* ML) je područje računarstva koje uz pomoć matematičkih modela i statističkih metoda omogućava učenje računalnim programima i sustavima na temelju prošlih iskustava čime poboljšavaju svoje performanse. Iako se o strojnom učenju najčešće priča u kontekstu računarstva, za izgradnju uspješnog ML modela potrebno je upotrijebiti znanja i iz drugih znanstvenih disciplina, kao što su matematika, statistika i analiza podataka.

Matematičke i statističke metode na kojima se temelji moderno strojno učenje postoje već stoljećima. Iako su igrale veliku ulogu u znanosti tijekom stoljeća svojih postojanja, izumom i razvojem modernog računala tijekom druge polovice 20. stoljeća, njihova je upotreba dobila potpuno novu dimenziju. Od izuma ENIAC-a, prvog elektronskog računala za opću upotrebu, pa sve do IBM-ovih Summita i Sierre, najbržih superračunala izumljenih do danas, razvoj discipline strojnog učenja pratio je razvoj modernog računala [15]. Arthur Samuel, IBM-ov znanstvenik koji je prvi definirao strojno učenje, 1952. je godine osmislio prvi računalni program koji uči, a svrha mu je bila naučiti igru dame [16]. Od tada, dizajnirana je prva neuralna mreža, osmišljen je algoritam „najbližih susjeda", a 1990-ih godina prelazi se sa pristupa koji se temelji na znanju na pristup koji se temelji na količini podataka [17]. Taj je pomak u pristupu doveo do razvijanja nove metodologije u strojnom učenju koje se naziva duboko učenje (eng. *Deep Learning*), koje se temelji na korištenju neuralnih mreža i velikih količina podataka.

Strojno učenje je našlo svoju primjenu na raznim područjima današnjeg znanstvenog, ali i svakodnevnog života. Neke od najčešćih primjena strojnog učenja su prepoznavanje slika, obrada prirodnog jezika, medicinske dijagnoze, predviđanje prometa, preporuke proizvoda, kao i široka primjena strojnog učenja u biokemiji, o čemu se više priča u nastavku rada.

Budući da većina definicija strojnog učenja zvuči prilično apstraktno, uobičajeno je da se strojno učenje definira i opiše kroz svakodnevne primjere.

U ovom ćemo primjeru biti kardiovaskularni doktori i postavljati ćemo dijagnoze za svoje pacijente, to jest postoji li mogućnost za srčane probleme kod pacijenta koji je ušao u našu ordinaciju. U obzir ćemo uzeti neke relevantne faktore, kao na primjer dob i spol pacijenta, težinu i visinu, krvni tlak, kolesterol, konzumira li pacijent alkohol i/ili cigarete i niz drugih faktora. Naravno, ovo nam nije prvi pacijent. Kroz naše dugogodišnje iskustvo, skupili smo stotine, ako ne i tisuće različitih pacijenata. Za svakog od tih pacijenata znali smo već navedene relevantne faktore, kao i njihovu dijagnozu. Sada, na temelju svih tih pacijenata i naših saznanja

o njima postavljamo dijagnozu našem novom pacijentu. Tijekom godina naučili smo puno toga te smo danas uspješni i cijenjeni doktori koji poboljšavaju zdravlje svojih pacijenata.

Strojno učenje funkcionira na sličan način. U ovom primjeru, naše iskustvo skupljeno putem pacijenata predstavlja podatke, koji su temelj svakog dobrog ML modela. Mi kao doktori predstavljamo matematičke modele koji na temelju iskustva, to jest podataka, oblikuju svoje hiperparametre putem raznih optimizacijskih algoritama. Nakon tog postupka učenja putem podataka, koji se u kontekstu strojnog učenja naziva treniranje, možemo predviđati buduće ishode i rješenja za određene probleme. Primjer jednog seta podataka prikazuje *Slika 3.1.*

| age | gender | height | weight | ap_hi | ap_lo | cholesterol | gluc | smoke | alco | active | cardio |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 18,393 | 2 | 168 | 620 | 110 | 80 | 1 | 1 | 0 | 0 | 1 | 0 |
| 20,228 | 1 | 156 | 850 | 140 | 90 | 3 | 1 | 0 | 0 | 1 | 1 |
| 18,857 | 1 | 165 | 640 | 130 | 70 | 3 | 1 | 0 | 0 | 0 | 1 |
| 17,623 | 2 | 169 | 820 | 150 | 100 | 1 | 1 | 0 | 0 | 1 | 1 |
| 17,474 | 1 | 156 | 560 | 100 | 60 | 1 | 1 | 0 | 0 | 0 | 0 |
| 21,914 | 1 | 151 | 670 | 120 | 80 | 2 | 2 | 0 | 0 | 0 | 0 |
| 22,113 | 1 | 157 | 930 | 130 | 80 | 3 | 1 | 0 | 0 | 1 | 0 |
| 22,584 | 2 | 178 | 950 | 130 | 90 | 3 | 3 | 0 | 0 | 1 | 1 |
| 17,668 | 1 | 158 | 710 | 110 | 70 | 1 | 1 | 0 | 0 | 1 | 0 |
| 19,834 | 1 | 164 | 680 | 110 | 60 | 1 | 1 | 0 | 0 | 0 | 0 |
| 22,530 | 1 | 169 | 800 | 120 | 80 | 1 | 1 | 0 | 0 | 1 | 0 |
| 18,815 | 2 | 173 | 600 | 120 | 80 | 1 | 1 | 0 | 0 | 1 | 0 |
| 14,791 | 2 | 165 | 600 | 120 | 80 | 1 | 1 | 0 | 0 | 0 | 0 |

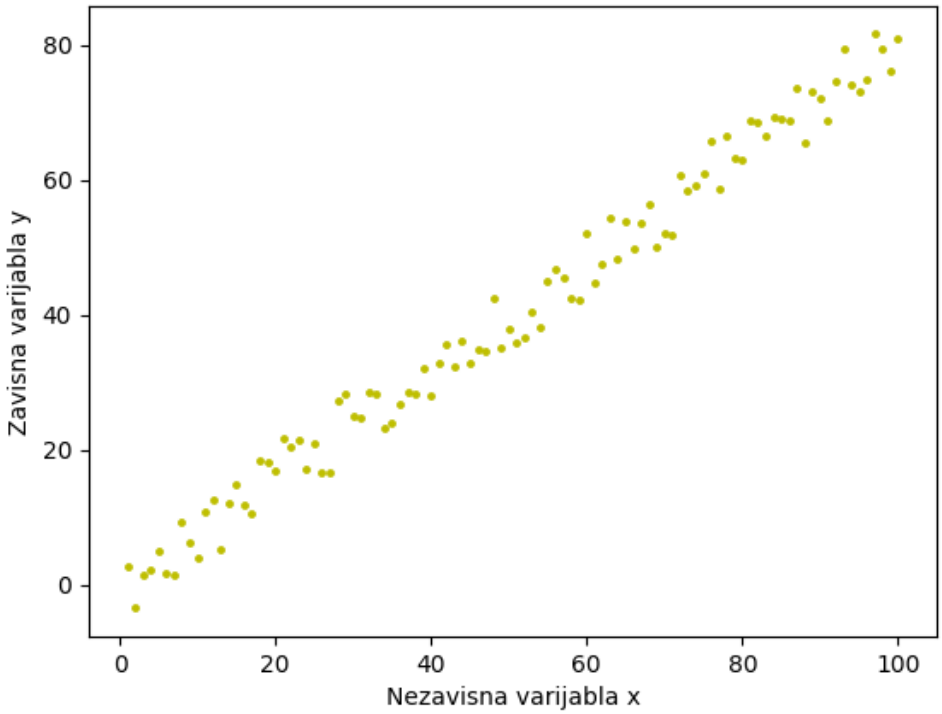
*Slika 3.1. Dio seta podataka o kardiovaskularnim bolestima (preuzeto iz [18])*

U kontekstu setova podataka, stupci se nazivaju značajke, dok se reci nazivaju primjeri ili instance. Veza između značajki i instanci jest ta da značajke opisuju instance. Dakle, u navedenom primjeru, značajke (dob, spol, visina, težina, itd.) opisuju pacijente, odnosno instance.

Osnovna svrha strojnog učenja je predviđanje. To predviđanje odrađuju modeli koji se treniraju uz pomoć podataka. U nastavku se opisuje postupak treniranja jednog iznimno čestog, a u isto vrijeme prilično jednostavnog i moćnog modela, a to je linearna regresija [19].

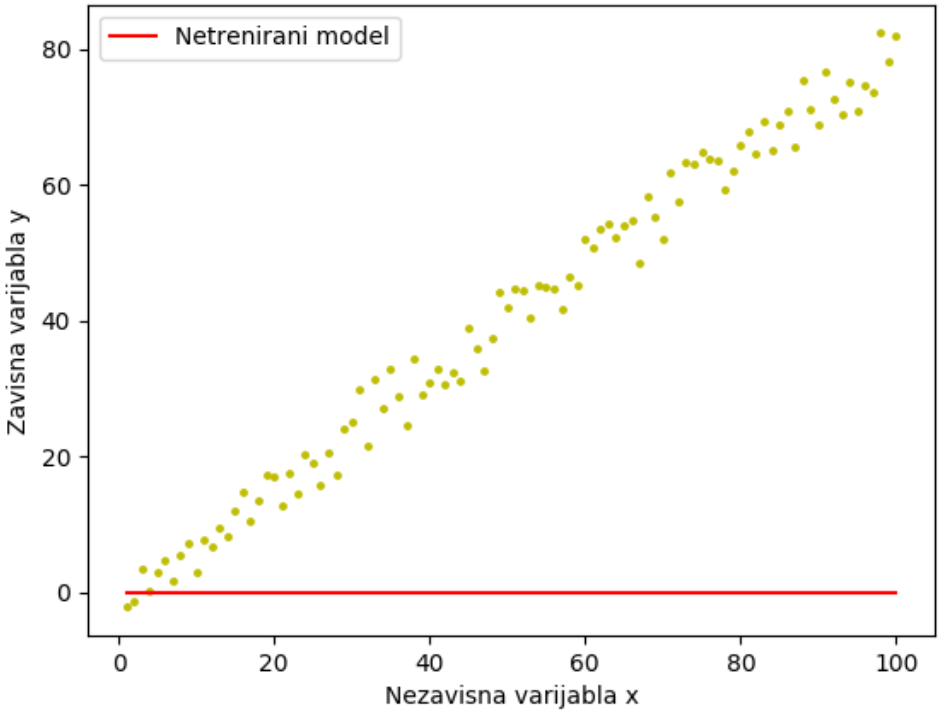
Osnovni cilj linearne regresije jest pronaći linearnu jednadžbu koja opisuje neke podatke. Zatim, na temelju te linearne jednadžbe, mogu se vršiti predviđanja.

Recimo da imamo neki set podataka koji je opisan kroz jednu značajku $x$ koja se naziva nezavisna varijabla. Ciljna varijabla, to jest varijabla koja se predviđa je varijabla $y$ koja se naziva zavisnom varijablom. Varijabla $y$ ovisi o varijabli $x$, zbog čega se naziva i zavisnom varijablom. Iz tog razloga se varijabla $y$ može vizualizirati kao funkcija varijable $x$.




*Slika 3.2. Zavisna varijabla y ovisi o nezavisnoj varijabli x*

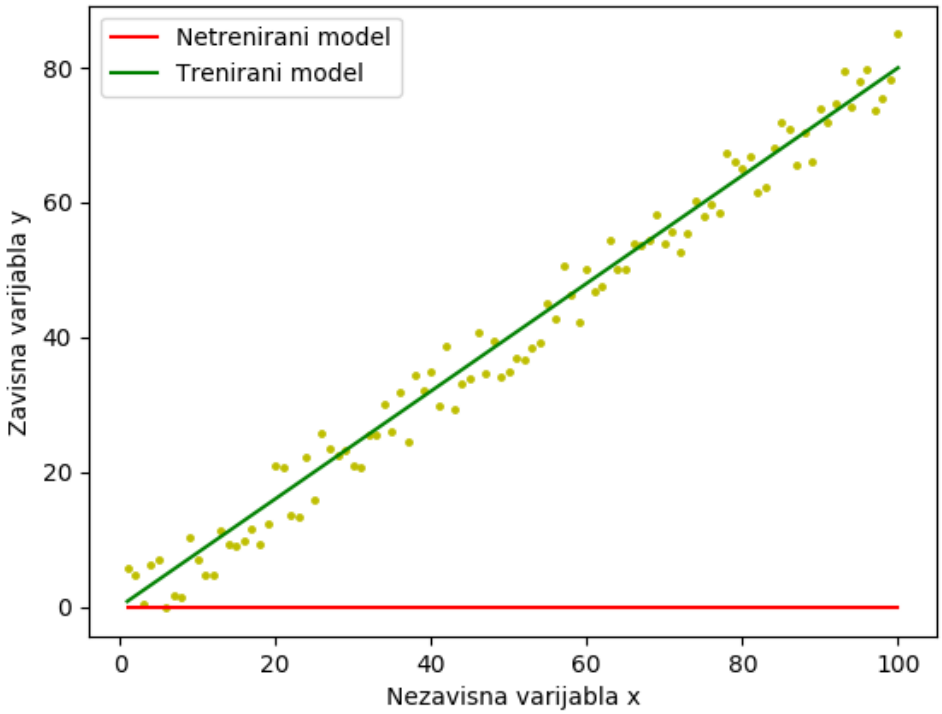
Cilj je pronaći linearnu jednadžbu $y = ax + b$ koja vjerno opisuje ovisnost varijable $y$ o varijabli $x$. Da bi se pronašli parametri $a$ i $b$, prvo se moraju inicijalizirati. Obično se parametri $a$ i $b$ postavljaju na nulu, nakon čega pravac linearne regresije izgleda kao onaj na *Slici 3.3.*




*Slika 3.3. Pravac linearne regresije za a = 0 i b = 0*

Sljedeći je korak u treniranju modela pronaći takve parametre *a* i *b* da bi linearna funkcija $y = ax + b$ dobro opisivala set podataka. To znači da bi pravac linearne regresije trebao prolaziti "kroz sredinu" vizualiziranih podataka.

Kako bismo imali metriku koja opisuje performanse modela, definira se cjenovna funkcija (eng. *Cost Function*). U kontekstu *Slike 3.3*, cjenovna funkcija mjeri udaljenost svake točke od pravca linearne regresije i zbraja te udaljenosti. Cilj je pronaći krivulju koja se nalazi što bliže svakoj od točaka, to jest minimizirati cjenovnu funkciju. To se postiže primjenom nekog minimizacijskog algoritma, kao što je gradijentni spust (eng. *Gradient Descent*) [20]. Gradijentni spust pronalazi parametre *a* i *b* za koje je cjenovna funkcija najniža. Primjenom gradijentnog spusta dobivaju se konačni paramteri, a time i istrenirani model, što prikazuje *Slika 3.4.*




*Slika 3.4. Krivulja istreniranog modela linearne regresije*

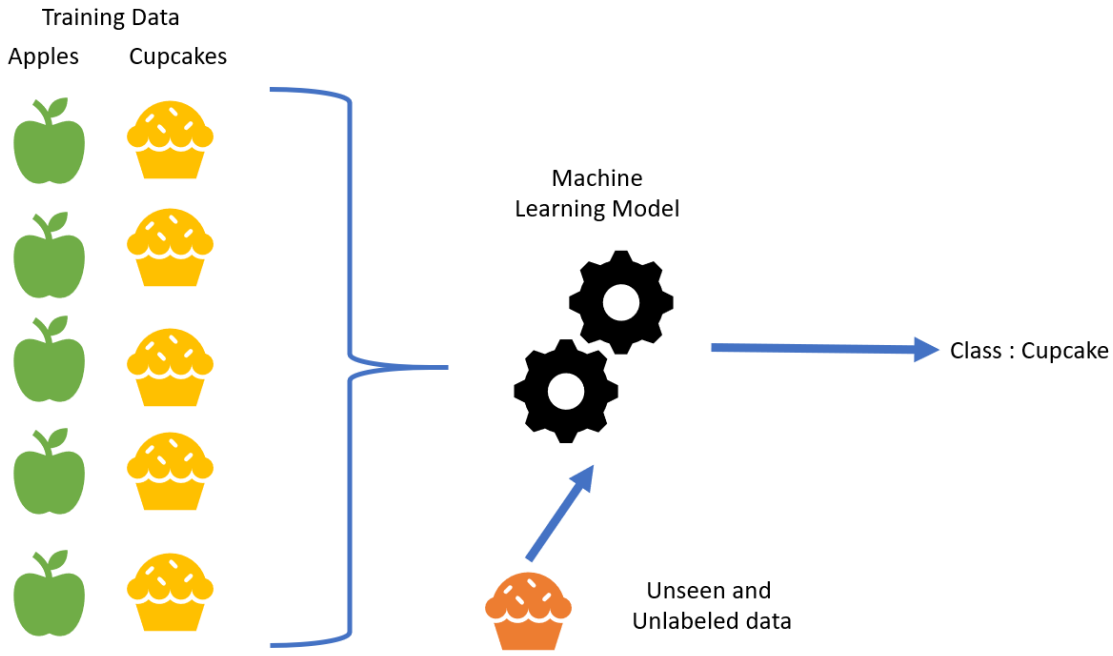
Gradijentni spust pronašao je da je cjenovna funkcija najniža za $a = 0.8$ i $b = 0$, što znači da jednadžba linearne funkcije glasi $y = 0.8x$. Sa ovim istreniranim modelom mogu se obavljati predviđanja na način da se za bilo koju vrijednost $x$ pronađe odgovarajuća vrijednost $y$ na pravcu.

Stvarna primjena strojnog učenja naravno nije toliko jednostavna. Značajke uglavnom nisu linearno zavisne, a može ih biti i na tisuće. Također, za naprednija predviđanja koriste se puno kompliciraniji modeli.

Linearna regresija model je koji rješava problem nadziranog učenja, jednog od područja strojnog učenja, što je tema sljedećeg poglavlja.

## 3.2. Vrste strojnog učenja

Strojno učenje primarno se dijeli na tri veće podvrste. Linearna regresija, model opisan u prethodnom poglavlju, pripada podvrsti strojnog učenja koje se naziva nadzirano strojno učenje (eng. *Supervised Learning*) [20]. Cilj je nadziranog strojnog učenja da se na temelju ulaznih podataka istrenira ML model koji će se kasnije koristiti za vršenje predviđanja. Ulazni se podaci sastoje od značajki, to jest nezavisnih varijabli, te zavisne, ciljne varijable. Veza između nezavisnih i zavisnih varijabli nalazi se u tome da zavisne varijable "označuju" nezavisne varijable. To znači da model na temelju trening podataka nauči kako označavati podatke koje prethodno nije vidio, to jest nezavisne varijable koje nisu označene pripadajućim zavisnim varijablama. Vizualnu reprezentaciju nadziranog učenja prikazuje je *Slika 3.5.*



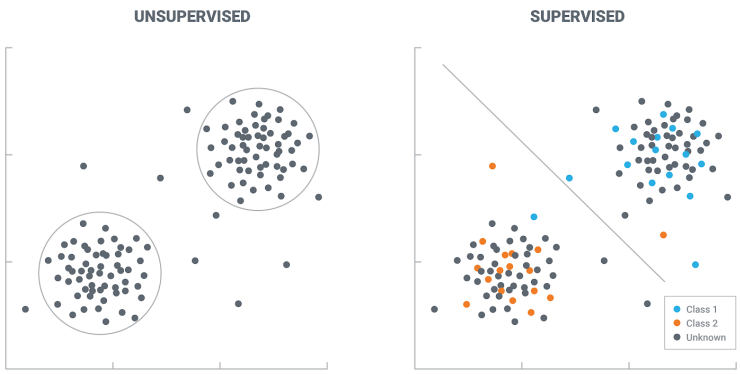

*Slika 3.5. Primjer modela nadziranog strojnog učenja (preuzeto iz [21])*

U primjeru na *Slici 3.5*, modelu se pokazuju primjeri jabuka i kolačića. Model zatim nauči razlikovati jabuke od kolačića i u mogućnosti je utvrditi da je ovaj novi, dosad neviđeni predmet zapravo kolačić. Nadalje, razlikovanje jabuka i kolačića primjer je klasifikacije, tipa nadziranog učenja u kojem su zavisne varijable kategoričkog tipa. Još neki primjeri klasifikacije su detekcija *spam* e-mail poruka, detekcija srčanih mana, prepoznavanje slika, i mnogi drugi primjeri.

Drugi tip nadziranog strojnog učenja naziva se regresija. Regresijski modeli predviđaju zavisnu varijablu koja se pojavljuje u obliku kontinuiranog broja koji može poprimiti bilo koju vrijednost, obično u nekom smislenom rasponu. Neki od primjera regresijskih zadataka su predviđanje cijene dionica, predviđanje internetskog prometa, te predviđanje tržišne cijene nekretnina.

Sljedeća podvrsta strojnog učenja bavi se podacima koji nisu označeni, to jest nemaju zavisnu varijablu koja se predviđa. Ovakav tip strojnog učenja naziva se nenadzirano strojno učenje (eng. *Unsupervised Learning*) [22]. Postoji nekoliko tipova nenadziranog strojnog učenja, kao što su grupiranje, detekcija anomalija, traženje asocijacija i redukcija dimenzionalnosti. Ono što je zajedničko svim tim podvrstama jest to da se bave nestrukturiranim podacima na temelju kojih modeli izvlače neke strukture, pravila ili zakonitosti. Glavna razlika između nadziranog i nenadziranog učenja nalazi se u tome da se kod nadziranog učenja modelu pokazuju primjeri iz kojih on uči i kasnije vrši predviđanja, dok kod nenadziranog učenja model sam uči na podacima i pronalazi uzorke. Nenadzirano učenje primjenjivo je kod, na primjer, grupiranja korisnika po određenim preferencijama, kreiranja asocijacija u marketingu (npr. ako kupac kupi kruh i mlijeko, velika je vjerojatnost da će kupiti i sir), kao i smanjivanja dimenzionalnosti visokodimenzionalnih setova podataka.

Razliku između nadziranog i nenadziranog učenja prikazuje *Slika 3.6.*




*Slika 3.6. Nenadzirano učenje (lijevo) naspram nadziranog učenja (desno) (preuzeto iz [23])*

Na prikazanim podacima, iz *Slike 3.6* je vidljivo da model nenadziranog učenja pokušava pronaći neku pravilnost, to jest grupirati podatke u neke cjeline. Na desnoj strani *Slike 3.6* je prikazan model nadziranog učenja, točnije klasifikacijski model. On će u istim tim podacima

pokušati pronaći neke razlike, to jest smisleno ih podijeliti u različite klase, kako bi kasnije mogao vršiti predviđanja.

Posljednja podvrsta strojnog učenja naziva se polunadzirano učenje (eng. *Semi-supervised Learning*). Polunadzirano učenje je svojevrsna kombinacija nadziranog i nenadziranog strojnog učenja. Ova se podvrsta strojnog učenja koristi kada na raspolaganju imamo set podataka kod kojeg je manji dio podataka označen, dok je većina podataka neoznačena. Pomoću metoda polunadziranog učenja, moguće je označiti veći dio podataka koji je neoznačen, a moguće je i pronaći funkciju koja opisuje odnos između nezavisnih varijabli i zavisne varijable koju predviđamo.

### 3.3. Primjena strojnog učenja u biokemiji

Kao što je napomenuto u prijašnjim poglavljima, strojno učenje primjenjivo je u mnogim aspektima današnjeg života. Jedna od tih primjena upravo je na području biokemije. Tako su, na primjer, američki znanstvenici 2008. godine razvili sustav koji korištenjem modificirane verzije modela „naivni Bayesov klasifikator" konstruira nove, dosad nepoznate supstrate peptida za enzime [24].

U malo novije doba, znanstvenici sa MIT-a (*Massachusetts Institute of Technology*) u veljači 2020. objavili su otkriće novog antibiotskog spoja korištenjem strojnog učenja [25]. Ovo je otkriće posebno značajno, budući da novootkriveni antibiotik ima sposobnost ubijanja nekih od najproblematičnijih bakterija u svijetu današnjice. Također, proteklih je desetljeća otkriveno vrlo malo novih antibiotika, a većina njih je naprosto varijacija na već postojeće antibiotike.

Kao što je i za pretpostaviti, strojno učenje našlo je i svoju primjenu na području antimikrobnih peptida. U jednoj od prvih takvih primjena, indijski su znanstvenici 2006. godine korištenjem različitih modela uspješno predviđali mogućnost upotrebe različitih peptida u antibakterijske svrhe [26]. Razvijanjem strojnog učenja, kao i povećanja dostupnih podataka, došlo je do razvijanja preciznijih i naprednijh modela. Kineski su znanstvenici 2019. razvili sustav koji putem dva klasifikatora nasumičnih šuma (eng. *Random Forest Classifier*) identificira funkcionalnosti različitih antimikrobnih peptida [27]. Prvi klasifikator predviđa ima li pojedini peptid antimikrobna svojstva. Ukoliko ima, proslijeđuje ga drugom klasifikatoru koji zatim predviđa koja su to svojstva, kao na primjer antibakterijska, antiparazitska ili antikancerogena svojstva.

U ovim su poglavljima definirani antimikrobni peptidi te od čega se sastoje, a definirani su i njihovi izvori. Također, definiran je i pojam strojnog učenja, objašnjen je postupak treniranja

modela, a predstavljeni su i pristupi koji postoje u disciplini strojnog učenja. Sada je vrijeme za glavni dio ovog rada, a to je predstavljanje modela predviđanja antimikrobnih peptida.

# 4. Studija slučaja

## 4.1. Razvojno okruženje

U tijeku razvoja ovog projekta, korištena su dva programska jezika, kao i nekoliko knjižnica koje ti jezici podržavaju. U sljedećim se poglavljima opisuju programski jezici R i Python, te korištene knjižnice.

### 4.1.1. Programski jezik R

R je interpretacijski, multiparadigmatski programski jezik razvijen 1990-ih godina na Sveučilištu u Aucklandu kao implementacija programskog jezika S, statističkog programskog jezika razvijenog 1976. godine [28, 29]. Od objavljivanja prve verzije 29. 2. 2000., R postaje sve popularniji jezik u zajednici podatkovnih znanstvenika, upravo zbog svojih moćnih mogućnosti i raznolikih knjižnica. Stoga nije iznenađujuće da R danas kotira kao osmi najpopularniji programski jezik prema TIOBE indeksu [30].

Unatoč svim prednostima jezika R, u ovom radu korišten je samo za stvaranje setova podataka. Za R je dostupna knjižnica pod nazivom Peptides pomoću koje je moguće izračunati čitav niz fiziokemijskih proteinskih deskriptora [31]. Korištenje ove knjižnice uvelike je olakšalo postupak selekcije značajki i proces stvaranja setova podataka, a moglo bi se i reći da su dobri rezultati konačnog modela posljedica korištenja knjižnice Peptides.

### 4.1.2. Programski jezik Python

Poput R-a, Python je interpretacijski programski jezik koji podržava više programskih paradigmi. Razlikuje se od R-a po tome što njegov fokus nije samo na statističkoj upotrebi, već je opće namjene, što argumentira njegova opširna standardna knjižnica. Činjenica da je Python proglašen najpopularnijim programskim jezikom 2007., 2010. i 2018. godine, svjedoči o tome da je Python već duže vrijeme u užem izboru programera diljem svijeta [32]. Trenutno je Python treći najpopularniji programski jezik [30].

Glavni razlog zbog kojeg je korišten Python u ovom projektu su njegove moćne i učinkovite knjižnice za strojno učenje, te manipulaciju, analizu i vizualizaciju podataka. Konkretno, Python je korišten za izgradnju i testiranje modela, kao i validaciju njegovih rezultata. Ovi postupci ostvareni su korištenjem knjižnica koje su opisane u nastavku.

Knjižnica Pandas primarno se koristi za rad sa podacima. Sastoji se od temeljne strukture podataka koja se naziva okvir podataka (eng. *Dataframe)*, dvodimenzionalne tablične strukture koja pohranjuje podatke u retke i stupce. Također, Pandas sadrži i dugi niz metoda za manipulaciju okvirima podataka.
Scikit-learn je iznimno opširna knjižnica koja sadrži sve alate potrebne za strojno učenje. U njoj se može pronaći velik broj gotovih ML modela, a manipulacijom njihovih hiperparametara moguće je izgraditi iznimno optimizirane modele. Također, u Scikit-learn je implementirano i mnogo tehnika za validaciju modela, prilagođavanje hiperparametara, računanje metrika i mnogo drugih metodologija strojnog učenja.

## 4.2. Podaci

Kvaliteta i kvantiteta podataka iznimno je važna za stvaranje svakog uspješnog modela strojnog učenja. U sljedećim se poglavljima opisuju podaci korišteni za stvaranje modela predviđanja antimikrobnih peptida.

### 4.2.1. Izvori podataka

Godine istraživanja na području antimikrobnih peptida dovele su do stvaranja raznih baza podataka punih informacija o antimikrobnim peptidima. Jedna od tih baza podataka je *The Antimicrobial Peptide Database*, skraćeno APD [8, 10]. U vrijeme provođenja projekta, APD sadrži informacije o 3167 peptida sa dokazanom antimikrobnom aktivnošću. Većina tih peptida je prirodnog porijekla, dok je jedan manji dio otkriven sintetskim putem. APD sadrži osnovne informacije o svakom peptidu, kao što su naziv peptida, njegovo porijeklo i sekvenca aminokiselina, kao i neke osnovne fiziokemijske deskriptore. Primjer antimikrobnog peptida opisanog u APD prikazuje *Slika 4.1.*

**Antimicrobial Peptide AP00001**

| | |
|---|---|
| APD ID: | AP00001 |
| Name/Class: | Dermaseptin-B2 (XXA, DRS-B2, Dermaseptin B2, DRS B2, DS bII, ADENOREGULIN; UCLL1c; frog, amphibians, animals) |
| Source: | Giant leaf frog, *Phyllomedusa bicolor*, South America |
| Sequence: | GLWSKIKEVGKEAAKAAAKAAGKAALGAVSEAV |
| Length: | 33 |
| Net charge: | 4 |
| Hydrophobic residue%: | 54% |
| Boman Index: | 0.23 kcal/mol |
| 3D Structure: | Helix |
| Method: | NMR |
| SwissProt ID: | SwissProt ID: P31107 Go to SwissProt |
| Activity: | Anti-Gram+ & Gram-, Antifungal, candidacidal, Anticancer |
| Crucial residues: | |
| Additional info: | A frog used for "hunting magic" by several groups of Panoan-speaking Indians in the borderline between Brazil and Peru is identified as Phyllomedusa bicolor. This natural peptide, isolated from that frog skin, may contain a D amino acid residue, since it is not identical in chromatographic properties to the synthetic peptide (Proc Natl Acad Sci U S A. 1992 Nov 15;89(22):10960-3). Synthetic adenoregulin enhanced the binding of agonists to several G-protein-coupled receptors in rat brain membranes. Active against M. canis IP 1194, T. rubrum IP 1400-82, A. simii IP 1063-74, A. caviae IP 67-16 P, E. coli IP 76-24, P. aeruginosa (MIC 3.1 uM), S. aureus ATCC 25923 (MIC 0.7 uM), N. brasiliensis IP 16-80, C. neoformans IP 960-67, C. neoformans IP 962-67, and C. albicans (MIC 10-60 ug/ml). A helix-hinge-helix structural motif (helix 1: 1-8; helix 2: 11-31) was found in complex with SDS2003 micelles. The N-terminal segment residues 1-11 is critical for antibacterial activity (Lequin O et al. 2003 Biochemistry 42: 10311-23). Of note is that the structure in TFE is quite different. APD Updated 10/2008; 5/2014; 7/2017 GW. |
| Title: | Isolation and structure of novel defensive peptides from frog skin. |
| Author: | Mor, A., Nicolas, P.1994 |
| Reference: | Eur J Biochem 1994, 219 (1-2):145-54. PubMed. |

*Slika 4.1. Dermaseptin-B2 opisan u APD (preuzeto iz [8])*

Kako bi se poštivala statistička uravnoteženost duljina peptida, što je u konačnici dovelo do boljih performansi modela, za trening set su korišteni peptidi duljine od 15 do 45 aminokiselina. Ova odluka temelji se na činjenici da medijan svih duljina peptida iz APD baze iznosi 29. Budući da su neki peptidi iz APD iznimno kratki, a neki iznimno dugački, korišten je medijan, a ne aritmetička sredina. Za razliku od aritmetičke sredine, medijan nije osjetljiv na ekstremne vrijednosti.

Nakon ovog filtriranja, iz APD baze se za pozitivne primjere uzima 1746 peptida. Naravno, za izgradnju uspješnog modela, potrebno je koristiti i negativne primjere. Kao izvor negativnih primjera, to jest peptida koji nemaju antimikrobnu aktivnost, korištena je UniProt baza podataka [33]. UniProt sadrži zapise o milijunima proteina, a većina njih je doznačena računalnom analizom bez ljudskog utjecaja. UniProt baza je podijeljena na nekoliko manjih baza, a jedna od tih baza naziva se UniProtKB/Swiss-Prot. Ova baza sadrži proteine koji su ručno pregledani i validirani od strane stručnih recenzenata, a informacije o tim proteinima temelje se na raznoj znanstvenoj literaturi.

Upravo je UniProtKB/Swiss-Prot izvor negativnih primjera za trening set. Uz pomoć službene UniProtove službe za podršku, kreiran je upit bazi podataka koji vraća peptide koji dokazano nemaju antimikrobnu aktivnost, što je polučilo 8813 peptida.

Budući da odabrani omjer pozitivnih i negativnih primjera iznosi 50:50, iz UniProtKB/Swiss-Prot iskorišteno je 1746 peptida duljine od 15 do 45 aminokiselina.

### 4.2.2. Značajke

Pri konstruiranju setova podataka, ponekad je važno staviti naglasak na kvalitetu, a ne kvantitetu podataka. Ovakav je pristup odabran u razvoju ovog modela. Veliku ulogu u procesu selekcije značajki imala je knjižnica Peptides za programski jezik R. Proučavanjem dostupne dokumentacije, dobiven je pregled svih fiziokemijskih deskriptora koje Peptides knjižnica nudi. Daljnjim istraživanjem i proučavanjem razne znanstvene literature, izdvojeni su obećavajući proteinski deskriptori. Konačno, provedena su testiranja učinkovitosti modela treniranih na setovima podataka dizajniranih korištenjem različitih kombinacija deskriptora. Ova testiranja dovela su do odabira finalnih deskriptora, čije su izlazne vrijednosti iskorištene kao značajke u završnom setu podataka. Popis ovih deskriptora prikazuje *Tablica 4.1.*

*Tablica 4.1. Fiziokemijski deskriptori u završnom setu podataka*

| | **Naziv** | **Tip deskriptora** | **Peptides funkcija** | **Broj komponenata** |
|---|---|---|---|---|
| 1. | BLOSUM Indices | Fiziokemijski | *blosumIndices()* | 10 |
| 2. | Cruciani Properties | Fiziokemijski | *crucianiProperties()* | 3 |
| 3. | Fasgai Vectors | Fiziokemijski | *fasgaiVectors()* | 6 |
| 4. | Kidera Factors | Fiziokemijski | *kideraFactors()* | 10 |
| 5. | MSWHIM Scores | Topološki | *mswhimScores()* | 3 |
| 6. | ProtFP | Fiziokemijski | *protFP()* | 8 |
| 7. | ST-Scales | Topološki | *stScales()* | 8 |
| 8. | T-Scales | Topološki | *tScales()* | 5 |
| 9. | VHSE Scales | Fiziokemijski | *vhseScales()* | 8 |
| 10. | Z-Scales | Fiziokemijski | *zScales()* | 5 |

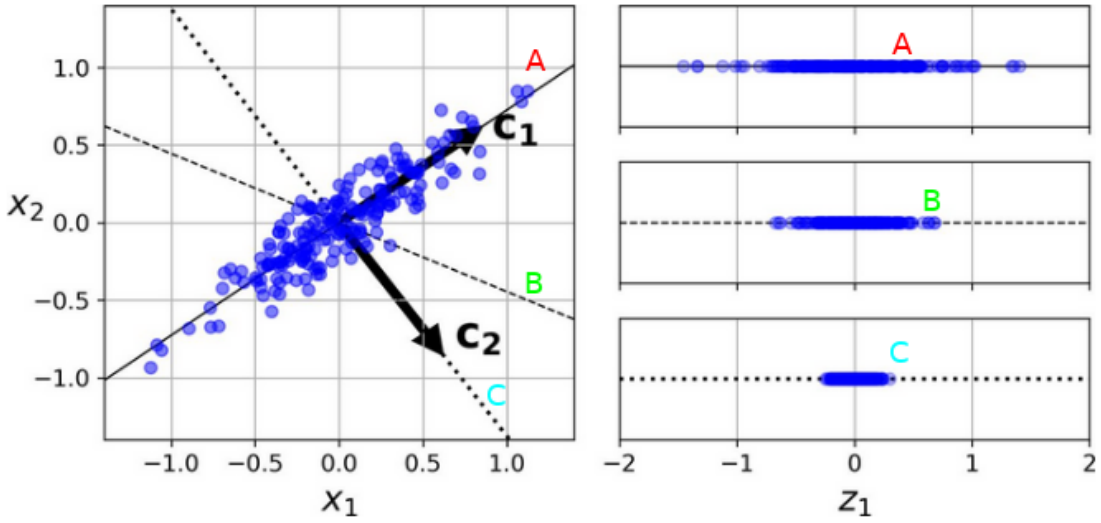
Stupac „tip deskriptora” odnosi se na svojstva koja pojedini deskriptor opisuje. Fiziokemijska svojstva su ona koja opisuju fizička svojstva, svojstva topljivosti i molekularna svojstva koja određuju intrizičnu kemijsku reaktivnost. Takva svojstva mogu biti, na primjer, hidrofobne, elektronegativne, lipofilne ili steričke prirode.

S druge strane, topološka svojstva odnose se na strukturalna svojstva peptida, kao što su raspored i povezanost aminokiselina ili molekularna struktura aminokiselina.

Stupac „Peptides funkcija” odnosi se na funkciju koja se poziva iz Peptides knjižnice za izračun pojedinog deskriptora.

Stavku „broj komponenata” najbolje je objasniti na primjeru procesa nastanka jednog od deskriptora, a taj primjer će biti VHSE deskriptor. VHSE (*Vectors of hydrophobic, steric, and electronic properties*) deskriptor dobiva se na način da se prvo izračunaju neka fiziokemijska svojstva peptida. Konkretno, računa se 18 hidrofobnih, 17 steričkih i 15 elektronskih svojstava, što ukupno iznosi 50 fiziokemijskih svojstava. Zatim ta svojstva prolaze kroz analizu principalnih komponenata, ili PCA (eng. *Principal Component Analysis*).

PCA je tehnika za reduciranje dimenzionalnosti uz istovremeni minimalni gubitak informacija. Reduciranje dimenzionalnosti ostvaruje se na način da se podaci dimenzije *n* projiciraju na hiper-ravninu dimenzije *n*-1. Ovu projekciju podataka prikazuje *Slika 4.2.*




*Slika 4.2. Projekcija dvodimenzionalnih podataka na jednodimenzionalni pravac (preuzeto iz [34])*

Na *Slici 4.2*, desno vidimo projekciju nekih podataka opisanih kroz dvije varijable, $x_1$ i $x_2$, na tri različita pravca. Očito je da pravac označen slovom „A” minimizira gubitak informacija, odnosno maksimizira varijanciju. Pravac „A” se naziva principalna komponenta. Ovime smo smanjili dimenzionalnost podataka uz minimiziranje gubitka informacija.

Isti postupak ponavlja se pri dobivanju VHSE deskriptora. Počinjemo sa 50 izračunatih svojstava. To znači da imamo 50-dimenzionalni prostor značajki. Provođenjem analize principalnih komponenata, smanjujemo ovaj prostor na 8 dimenzija, to jest 8 komponenata. Ukupan broj komponenata svih korištenih deskriptora iznosi 66, što znači da je svaki peptid u setu podataka opisan kroz 66 značajki.

Glavna motivacija za korištenje ove kombinacije deskriptora leži u znanstvenom radu koji je plod suradnje britanskih, belgijskih i nizozemskih znanstvenika [35]. U radu se opisuje i uspoređuje 8 različitih deskriptora, od kojih su neki isprobani u više varijanti, čime se broj testiranih deskriptora penje na 13. U pratećem radu, ti deskriptori korišteni su za stvaranje setova podataka na kojima su trenirani regresijski i klasifikacijski modeli nasumičnih šuma [36]. Jedan od problema koje su ti modeli rješavali bio je, na primjer, predviđanje bioaktivnosti enzimskih inhibitora protiv enzimskih mutanata HIV-a. U tom je znanstvenom radu dokazano kako modeli trenirani na setovima podataka koji se sastoje od ovih 13 deskriptora, unatoč tome što svaki deskriptor opisuje proteine na malo drugačiji način, pokazuju vrlo slične performanse. Nadalje, kombiniranjem određenih deskriptora u jedan set podataka dolazi se do poboljšanja performansi modela. Koristeći ovu smjernicu, počeo je proces stvaranja seta podataka.

Za stvaranje seta podataka testirani su deskriptori koji su dostupni u Peptides knjižnici, a koji su također testirani u navedenom istraživanju. To su *BLOSUM*, *FASGAI*, *MSHWIM*, *ProtFP*, *ST-scales*, *T-scales*, *VHSE* i *Z-scales*. Ne oslanjajući se na znanstvena istraživanja, u kombinaciju su ubačena i dva dodatna deskriptora, *Kidera Factors* i *Cruciani Properties*.

Testiranjem raznih kombinacija navedenih značajki, pokazalo se da model predviđanja antimikrobnih peptida pokazuje najbolje performanse kada se trenira na setu podataka koji je opisan koristeći sve navedene deskriptore.

## 4.3. Odabir modela

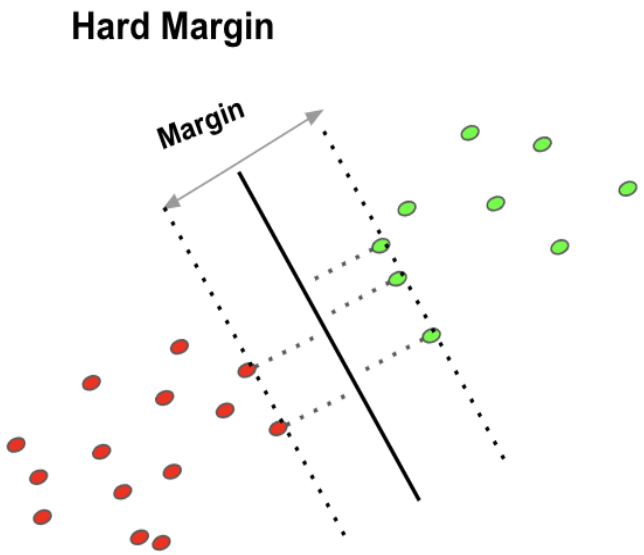
Budući da različiti modeli pokazuju različite performanse na istim podacima, važno je odabrati pravi model za problem koji se rješava. Stoga, u ovom poglavlju se opisuje model koji se pokazao kao najbolje rješenje za dani problem, a to je SVM model. Također, definiraju se i metrike koje su korištene za procjenu performansi modela.

### 4.3.1. Osnove SVM modela

Strojevi vektora podrške (eng. *Support Vector Machines,* SVM) je model strojnog učenja čija je prvotna verzija razvijena 1960-ih godina u Bell Labs, znanstveno-istraživačkoj tvrtki koja je zaslužna za mnoge velike tehnološke inovacije, poput tranzistora, lasera, Unix operacijskog sustava, te programskih jezika C i C++ [37]. Tijekom sljedeća 3 desetljeća, SVM model je postupno dorađivan, a na petoj ACM-ovoj konferenciji o teoriji računalnog učenja 1992. godine predstavljen je SVM model kakav se koristi danas [38].

SVM je model nadziranog učenja koji je sposoban za rješavanje klasifikacijskih i regresijskih problema. Unatoč tome što je regresija pomoću SVM modela vrlo slična klasifikaciji, ovo će poglavlje biti orijentirano prema opisivanju SVM klasifikatora.

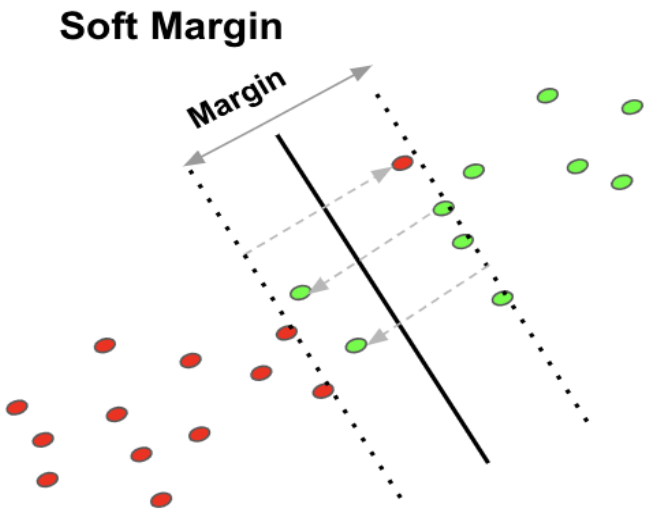
Osnovna ideja SVM modela je da se pronađe hiper-ravnina koja smisleno odjeljuje podatke, kako bi se na taj način mogla obavljati predviđanja. Jednu takvu hiper-ravninu prikazuje *Slika 4.3*.




*Slika 4.3. Klasifikacija tvrdom marginom (preuzeto iz [39])*

Ravna puna linija na slici predstavlja hiper-ravninu, u ovom slučaju je to pravac, koji smisleno odvaja instance dviju različitih klasa. U kontekstu SVM-a, hiper-ravnina se naziva granica odluke (eng. *Decision Boundary*). Na *Slici 4.3* je vidljivo da granica odluke prolazi sredinom prostora između dvije klase instanci, a taj prostor zove se margina. Također, vidljivo je da je margina određena sa dvije isprekidane linije. Svaka od ovih isprekidanih linija prolazi kroz instance koje su najbliže instancama suprotne klase. Ove se instance nazivaju vektori podrške (eng. *Support Vectors*). Budući da je cilj maksimizirati marginu između vektora podrške, logično je da će margina, a time i granica odluke, biti određena vektorima podrške, te da unutar margine neće biti niti jedne instance. Sada, želimo li klasificirati neku novu instancu, to možemo učiniti korištenjem granice odluke. Prema *Slici 4.3*, ukoliko se nova instanca nalazi lijevo od granice odluke, bliže crvenim instancama, reći ćemo da instanca pripada crvenoj klasi. U suprotnom slučaju reći ćemo da pripada zelenoj klasi. Ovakva metoda klasificiranja naziva se klasifikacija tvrdom marginom (eng. *Hard Margin Classification*).

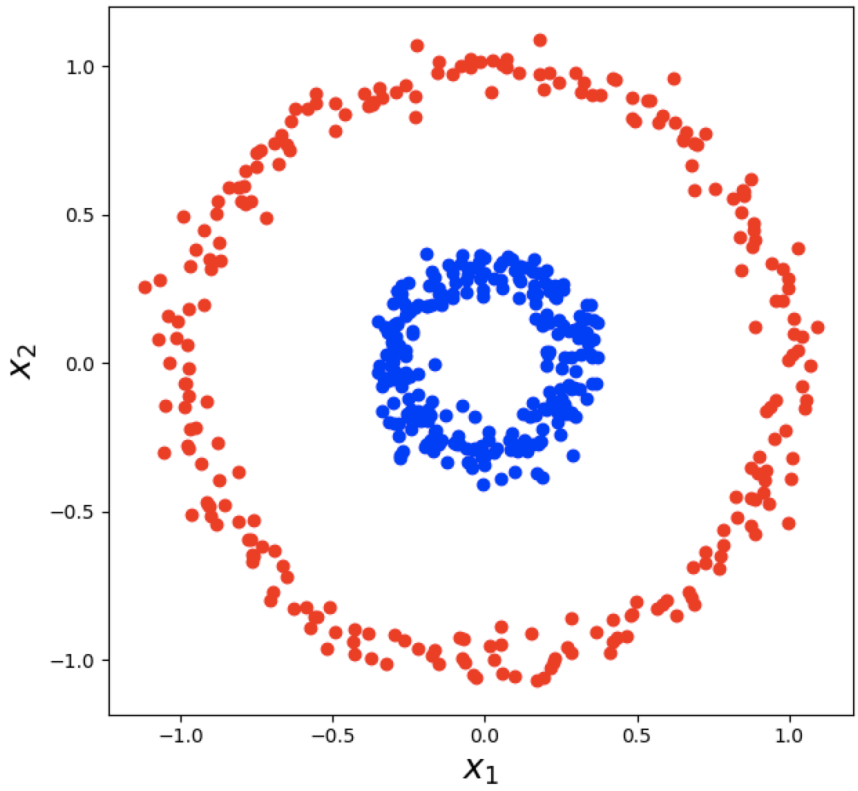
No metoda klasifikacije tvrdom marginom ima svoje nedostatke. Recimo da imamo podatke kao na *Slici 4.4.*




*Slika 4.4. Klasifikacija mekanom marginom (preuzeto iz [39])*

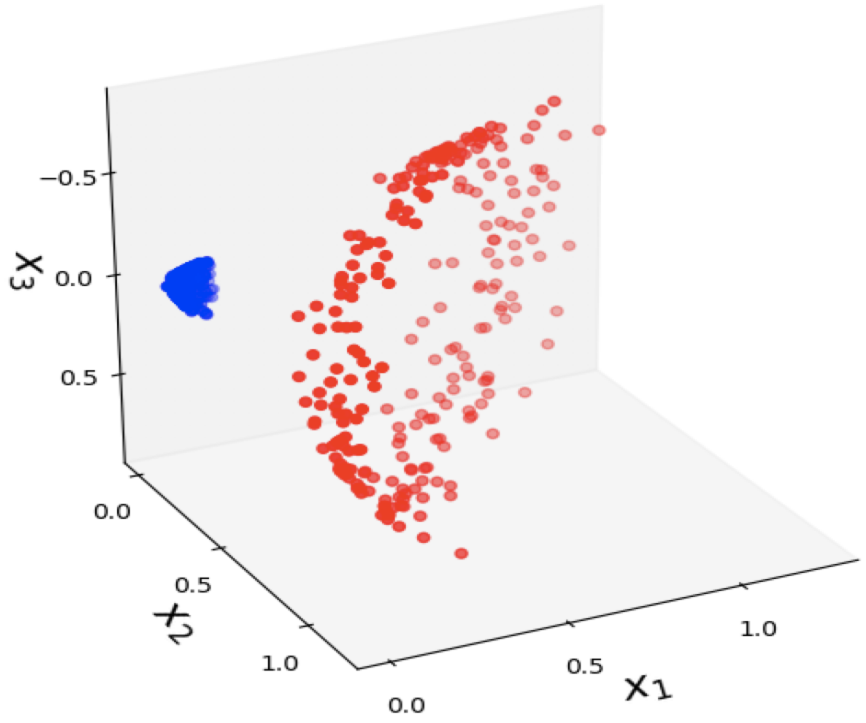
Na *Slici 4.4* vidimo da su neke instance različitih klasa „pomiješane", zbog čega metodom tvrde margine nebi bilo moguće odrediti vektore podrške, čime nebi mogli odrediti marginu, a u konačnici ni granicu odluke. Zbog ovakvih situacija je korištenje klasifikacije mekanom marginom standard modernog SVM modela. Kod klasifikacije mekanom marginom instance unutar margine su dopuštene, pa čak i sa „krive" strane granice odluke. Sa ovom promjenom dobiva se regularizirana granica odluke koja će na bolji i precizniji način odjeljivati različite klase.

Prava vrijednost SVM modela leži u njegovoj interakciji sa podacima i mijenjanju dimenzionalnosti podataka. Recimo da imamo podatke kao na *Slici 4.5.*




*Slika 4.5. Dvodimenzionalni set podataka (preuzeto iz [40])*

Očito je da ovaj set podataka nije linearno djeljiv, to jest ne može se odrediti hiper-ravninu koja će dijeliti crvenu klasu od plave klase. Ono što SVM model radi jest da provodi transformacije podataka i povećava dimenzionalnost podataka kako bi pronašao granicu odluke više dimenzionalnosti koja će moći razdijeliti crvenu klasu od plave. U kontekstu podataka sa *Slike 4.5*, takvu promjenu dimenzionalnosti predstavlja *Slika 4.6.*




*Slika 4.6. Promjena dimenzionalnosti podataka (preuzeto iz [40])*

Na *Slici 4.6* vidimo da je sada moguće pronaći granicu odluke, pomoću koje možemo obavljati predikcije na novim, neviđenim podacima. Kako bi transformirao podatke i mapirao ih u više dimenzije, SVM koristi matematičke funkcije koje se nazivaju jezgrene funkcije (eng. *Kernel Functions*). Budući da su ove funkcije prilično matematički kompleksne, neće se detaljno predstavljati, nego će se istaknuti samo ono najvažnije. Jezgrene funkcije računaju odnose između svih instanci u svakoj pojedinoj dimenziji *d*. Ovi se izračunati odnosi zatim koriste kako bi se pronašla granica odluke. U primjeru opisanom na *Slici 4.5* i *Slici 4.6* korištena je polinomska jezgrena funkcija, kod koje je moguće predefinirati ciljnu dimenzionalnost podataka. Postoji mnogo različitih jezgrenih funkcija, a uz polinomsku, neke od najkorištenijih jezgrenih funkcija su Gaussova, sigmoidna, te RBF jezgrena funkcija, koja se pokazala kao najbolja za model korišten u ovom projektu. Ono zbog čega su jezgrene funkcije perjanica SVM modela je činjenica da su računski iznimno učinkovite. Jezgrene funkcije računaju odnose između instanci na način da se ponašaju prema instancama kao da su u višoj dimenziji, bez da ih zapravo i transformiraju u te više dimenzije. Ovo iznimno učinkovito, ali i poprilično apstraktno matematičko rješenje zove se jezgreni trik (eng. *The Kernel Trick*), te omogućuje pronalaženje granice odluke u beskonačno mnogo dimenzija.

U strojnom učenju, kao i u bilo kojem radu sa podacima, uvijek je dobro imati podatke koji su opisani sa jednom od uobičajenih distribucija jer je takve podatke lakše opisati i analizirati. Budući da je u ovom projektu korišten SVM model koji ne pretpostavlja nikakvu distribuciju podataka, nije bilo potrebe za provođenjem statističkih testova kako bi se ispitala distribucija podataka, niti su provođene statističke transformacije podataka.

### 4.3.2. SVM model iz knjižnice Scikit-learn

Kao što je već navedeno, za postupak razvijanja modela korištena je knjižnica Scikit-learn za programski jezik Python. Za instanciranje modela korištena je klasa *sklearn.svm.SVC*. Postoji niz hiperparametara koje je moguće prilagoditi kako bi model imao što bolje performanse, no kako odabrati vrijednosti tih hiperparametara? Pri podešavanju hiperparametara, korištena je metoda pretraživanja (eng. *Grid Search*). Kod ove se metode model trenira više puta, svaki put sa drugačijim hiperparametrima. Performanse svakog istreniranog modela se ispituju unakrsnom validacijom. Produkt metode pretraživanja su hiperparametri modela koji je pokazao najbolje performanse. Najvažniji hiperparametri koje je metoda pretraživanja pomogla optimizirati su odabir jezgrene funkcije i vrijednost regularizacijskog hiperparametra. Metoda pretraživanja ostvarena je pomoću klase *sklearn.model_selection.GridSearchCV*.

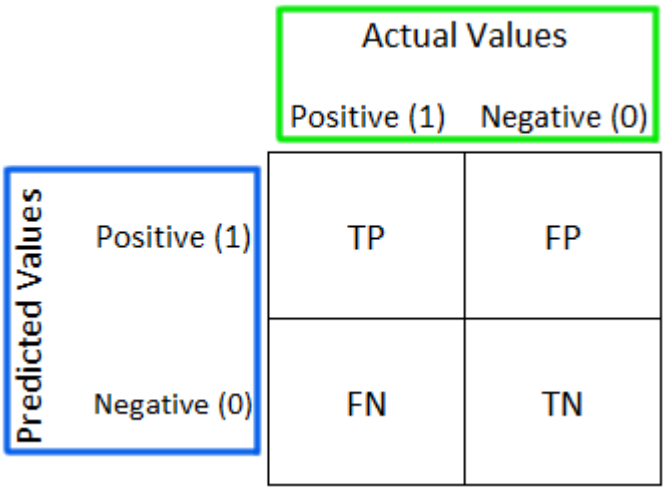
Sa pronađenim optimalnim hiperparametrima, može se krenuti u treniranje modela. No prije toga, budući da na optimalnu hiper-ravninu utječe raspon podataka s kojima se radi, pri korištenju SVM modela dobra je praksa standardizirati podatke prije treniranja modela. Standardizacija podataka provodi se na sljedeći način: za svaku značajku izračuna se aritmetička sredina i standardna devijacija. Od svake značajke svake instance oduzme se aritmetička sredina te značajke, a ta se vrijednost podijeli sa standardnom devijacijom te značajke. Formula za standardizaciju podataka je sljedeća:

$$x' = \frac{x - u}{s}$$

Za provođenje standardizacije korištena je klasa *sklearn.preprocessing.StandardScaler*.

### 4.3.3. Evaulacija modela predviđanja

Metrike koje su korištene za vrednovanje modela su one standardne koje se koriste za klasifikacijske probleme u strojnom učenju. Prije opisivanja samih metrika, potrebno je definirati matricu konfuzije (eng. *Confusion Matrix*). Vizualno, matricu konfuzije prikazuje *Slika 4.7.*




*Slika 4.7. Matrica konfuzije (preuzeto iz [41])*

Reci označeni plavim kvadratom predstavljaju predviđane vrijednosti, dok stupci označeni zelenim kvadratom predstavljaju stvarne, istinite vrijednosti. Ćelije označene sa TP, FP, FN i TN znače objašnjava *Tablica 4.2.*

*Tablica 4.2. Definicija ćelija matrice konfuzije*

| **Kratica** | **Značenje** | **Prijevod** | **Objašnjenje** |
|---|---|---|---|
| TP | *True Positive* | Točan pozitiv | Pozitivna instanca točno identificirana kao pozitivna. |
| FP | *False Positive* | Netočan pozitiv | Negativna instanca netočno identificirana kao pozitivna. |
| FN | *False Negative* | Netočan negativ | Pozitivna instanca netočno identificirana kao negativna. |
| TN | *True Negative* | Točan negativ | Negativna instanca točno identificirana kao negativna. |

Prije same evaulacije performansi modela, set podataka se dijeli na trening set i testni set podataka. Omjer trening i testnih podataka generalno ovisi o dostupnoj količini podataka i o računskoj kompleksnosti modela, no obično se 80% podataka koristi kao za trening, dok se preostalih 20% podataka koristi za testiranje. Također, poželjno je kada bi trening i testni set podataka oboje bili stratificirani. Na primjer, ukoliko je omjer pozitivnih i negativnih primjera 50:50, poželjno je kada bi omjer pozitivnih i negativnih primjera u trening setu i testnom setu također bio 50:50.

Model se trenira na trening setu podataka, a zatim se koristi testni set kako bi se dobile brojke koje si definirane u *Tablici 4.2. Z*brajaju se svi točni pozitivi, netočni pozitivi, netočni negativi i točni negativi te se pomoću ovih vrijednosti mogu izračunati metrike za validaciju modela.

Prva korištena metrika, a ujedno i ona koja najbolje opisuje performanse modela, je točnost (eng. *Accuracy*). Točnost mjeri postotak ukupnih instanci koje je model točno identificirao. Računa po sljedećoj formuli:

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

Sljedeća korištena metrika je preciznost (eng. *Precision*). Ova metrika otkriva koliki je postotak instanci, od svih koje su identificirane kao pozitivne, uistinu pozitivan. Računa se korištenjem sljedeće formule:

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

Još jedna korištena metrika je osjetljivost (eng. *Sensitivity, Recall*). Osjetljivost odgovara na sljedeće pitanje: od svih instanci koje su uistinu pozitivne, koliki ih je postotak točno identificirano kao pozitivne? Osjetljivost se naziva i omjer točnih pozitiva (eng. *True Positive Rate*, TPR). Računa se pomoću formule:

$$\text{Osjetljivost} = \frac{TP}{TP + FN}$$

Analogno osjetljivosti, postoji i metrika koja računa postotak točnih negativa od svih instanci identificiranih kao negativne. Ova se metrika naziva specifičnost (eng. *Specificity*) ili omjer točnih negativa (eng. *True Negative Rate*, TNR). Računa se po formuli:

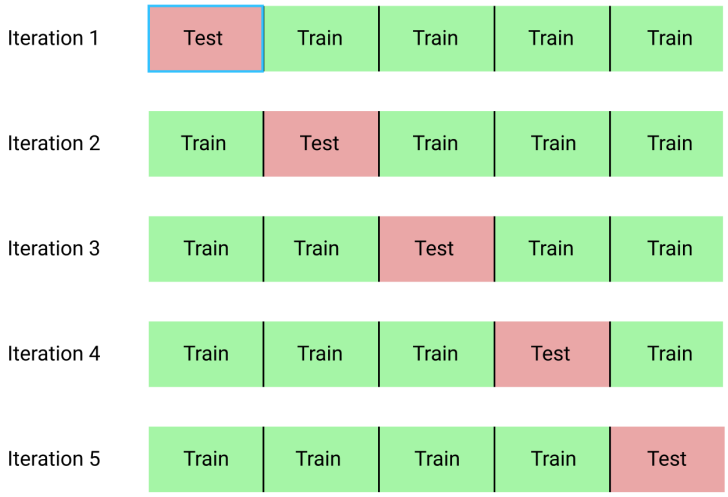
$$\text{Specifičnost} = \frac{TN}{TN + FP}$$

Posljednja metrika korištena za evaluaciju modela je $F_1$ rezultat (eng. *$F_1$ Score*). Koristi se za evaluaciju ukupne točnosti modela. $F_1$ rezultat računa se kao harmonijska sredina preciznosti i osjetljivosti, te zbog toga daje veću težinu niskim vrijednostima. Posljedično tome, $F_1$ rezultat će biti visok samo ako su metrike preciznosti i osjetljivosti također visoke. Zbog toga je ova metrika vrlo pogodna za procjenu performansi modela, kao i za brzu usporedbu performansi dvaju ili više modela. $F_1$ rezultat računa se po sljedećoj formuli:

$$F_1 \text{ rezultat} = \frac{2 \cdot Preciznost \cdot Osjetljivost}{Preciznost + Osjetljivost}$$

Ove su metrike neizostavan element pri evaluaciji performansi nekog modela. Korištenjem jedne metrike brzo se dobiva uvid u performanse testiranog modela. Korištenjem većeg broja metrika dobiva se još dublji uvid u performanse modela, čime se pronalaze nedostaci modela i otkrivaju se novi načini za poboljšanje. Sve su metrike izračunate pomoću funkcija koje se nalaze u *sklearn.metrics* modulu iz Scikit-learn knjižnice.

Kako bi se dodatno povećala preciznost informacija koje nam metrike performansi pružaju, koristi se metoda unakrsne validacije (eng. *Cross-validation*). Ova metoda dijeli podatke na *k* podjednakih dijelova. Jedan od tih dijelova koristi se za testiranje, dok se preostalih *k*-1 dijelova koristi za treniranje modela. Postupak treniranja i testiranja ponavlja se *n* puta kako bi svaki od *k* dijelova bio korišten za testiranje točno jednom. Kao i kod obične podijele na trening i testni set, poželjno je kada bi svaki od dijelova *k* dijelova bio stratificiran. Metoda unakrsne validacije vizualno je prikazana putem *Slike 4.8*.




*Slika 4.8. Petodijelna unakrsna validacija (preuzeto iz [42])*

U ovom je projektu korištena desetodijelna unakrsna validacija (eng. *10-fold cross-validation*). Korištenje 10 dijelova znači da u svakoj od 10 iteracija omjer trening i testnih podataka iznosi 90:10. Za implementaciju metode unakrsne validacije korištena je funkcija *sklearn.model_selection.cross_val_predict*. Također, važno je napomenuti kako ova funkcija ima mogućnost stratifikacije svakog od *k* dijelova. To znači da pri unakrsnoj validaciji modela predviđanja antimikrobnih peptida u svakom od 10 dijelova omjer pozitivnih i negativnih primjera iznosi 50:50.

# 5. Rezultati

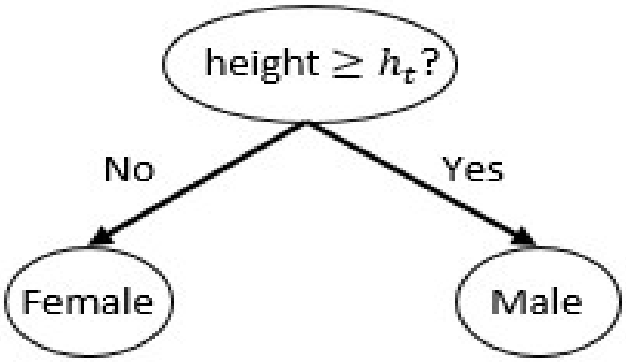
U ovom će se poglavlju predstaviti rezultati glavnog SVM modela koji su evaluirani putem metrika definiranih u poglavlju 4.3.3. Prije toga, predstaviti će se i modeli koji su prethodili SVM modelu, te njihovi rezultati.

## 5.1. Usporedbe rezultata različitih modela

Pri odabiru modela, dobra praksa je isprobati više modela te odabrati onaj model koji pokazuje najbolje performanse na podacima kojima raspolažemo. U ovom se poglavlju predstavljaju rezultati tri različita modela.

Prvi testirani model je logistička regresija, klasifikacijska verzija linearne regresije, modela koji je ugrubo predstavljen u poglavlju 3.1. Unatoč tome što je logistička regresija poprilično jednostavan, ali moćan model, rezultati ovog modela su bili u skladu sa očekivanjima - poprilično prosječni, što je prikazano u *Tablici 5.1*. Ovo je vjerojatno posljedica visokog stupnja dimenzionalnosti seta podataka. Također, valja napomenuti da je testirano više oblika logističke regresije, poput *ridge* regresije, *lasso* regresije, kao i *elasticnet* regresije, a isprobano je i nekoliko različitih optimizacijskih algoritama u procesu treniranja modela. Svaki od ovih modela je postigao vrlo slične rezultate, što je točnost od 82%. Za stvaranje modela, korištena je klasa *sklearn.linear_model.LogisticRegression* iz knjižnice Scikit-learn za Python.

Sljedeći testirani model je AdaBoost (*Adaptive Boosting*). AdaBoost je ansambl metoda koja kombinira veći broj slabijih modela od kojih svaki daje svoju predikciju za pojedini primjer, a na kraju se predikcije svih tih slabijih modela kombiniraju u konačan rezultat. Kod AdaBoost-a, ti manji i slabiji modeli su stabla odluke (eng. *Descision Trees*) čija dubina iznosi jedan. Takva se stabla odluke nazivaju i panjevi odluke (eng. *Decision Stumps*). Primjer jednog panja odluke prikazuje *Slika 5.1.*




*Slika 5.1. Panj odluke koji predviđa spol osobe (preuzeto iz [43])*

Panj odluke na slici predviđa spol osobe na temelju visine. Naravno, potrebno je puno više faktora kako bi se odredio spol osobe, a to je posao drugih panjeva odluke. Upravo u tome leži smisao ansambl metoda - kombiniranjem stotina ili tisuća prediktora slabijih performansi dobivamo jedan prediktor visokih performansi. Budući da je AdaBoost napredniji model od logističke regresije, očekivale su se bolje performanse. Ta su se očekivanja i ispunila, budući da je AdaBoost model postigao točnost od 88%. U kontekstu knjižnice Scikit-learn za Python, za stvaranje modela korištena je klasa *sklearn.ensemble.AdaBoostClassifier*.

Posljednji testirani model pokazao je najveću točnost, a zbog toga je i odabran kao konačni model koji će se koristiti u projektu - SVM model. Bolje performanse SVM modela naspram performansi logističke regresije i AdaBoost-a mogu se pripisati činjenici da SVM model iznimno dobro reagira na visokodimenzionalne podatke. Metrike performansi SVM modela prikazane su u *Tablici 5.1.* SVM model je detaljnije opisan u poglavlju 4.3.1.

Rezultati ova tri testirana modela uspoređuju se u *Tablici 5.1.* Kao što je već spomenuto u poglavlju 4.3.3, svi modeli su validirani putem desetodijelne unakrsne validacije.

*Tablica 5.1. Izmjerene metrike testiranih modela*

| | **Model** | **Točnost** | **Preciznost** | **Osjetljivost** | **Specifičnost** | **$F_1$ rezultat** |
|---|---|---|---|---|---|---|
| 1. | Logistička regresija | 82.7% | 83.1% | 82.1% | 82.7% | 82.7% |
| 2. | AdaBoost | 88.4% | 88% | 89% | 88.4% | 88.4% |
| 3. | SVM | 91.5% | 90% | 92.5% | 91.5% | 91.5% |

# 6. Zaključak

Ljudsko je zdravlje oduvijek bilo i uvijek će biti iznimno važna tema kojoj se posvećuje puno vremena, rada i truda s ciljem njegovog konstantnog poboljšanja. Nema te znanstvene discipline koja se nije upotrijebila u svrhe unaprijeđivanja medicine, pa su tako svoj danak dali biokemija i računarstvo. Kombiniranje metodologija biokemije i strojnog učenja dokazano funkcionira u svrhe unaprijeđivanja medicine i poboljšanja ljudskog zdravlja. Uz pomoć strojnog učenja, moguće je ubrzati postupak otkrivanja novih antimikrobnih peptida, što dovodi do povećanja ukupnog broja poznatih antimikrobnih peptida sa sposobnošću otklanjanja manjih i svakodnevnih zdravstvenih poteškoća, ali i trajnih i ozbiljnih problema globalnih razmjera.

Zaključno, na temelju svog vremena potrošenog na istraživanje, truda uloženog u razvijanje modela, te zadovoljavajućih konačnih rezultata, drago mi je što mogu zaključiti da primjena strojnog učenja u svrhe predviđanja antimikrobnih peptida uistinu može doprinijeti dobrobiti ljudskog zdravlja.

# 7. Literatura

[1] https://www.medicalnewstoday.com/articles/10278, 2.6.2020.

[2] https://www.nhs.uk/conditions/antibiotics/uses/, 2.6.2020.

[3] https://www.healthline.com/health/hospital-acquired-nosocomial-infections, 2.6.2020.

[4] Kang H. i dr.: "The therapeutic applications of antimicrobial peptides (AMPs): a patent review", Journal of Microbiology, 2017.

[5] https://medlineplus.gov/ency/article/002222.htm, 4.6.2020.

[6] https://www.compoundchem.com/2014/09/16/aminoacids/, 4.6.2020.

[7] http://www.biology.arizona.edu/biochemistry/problem_sets/aa/aa.html, 4.6.2020.

[8] http://aps.unmc.edu/AP/main.php, 8.6.2020.

[9] Wang G.: "Human Antimicrobial Peptides and Proteins", Pharmaceuticals, 2014.

[10] Wang G., Li X., Wang Z., "APD3: the antimicrobial peptide database as a tool for research and education", Nucleic Acids Research, 2016.

[11] Moore, K. S. i dr.: "Squalamine: an aminosterol antibiotic from the shark." Proceedings of the National Academy of Sciences of the United States of Americ, 1993.

[12] https://www.medicinenet.com/squalamine/supplements-vitamins.htm, 11.6.2020.

[13] https://ourworldindata.org/cancer, 11.6.2020.

[14] Rozek, T. i dr.: "The antibiotic and anticancer active aurein peptides from the Australian Bell Frogs Litoria aurea and Litoria raniformis", The FEBS Journal, 2000.

[15] https://www.ibm.com/thought-leadership/summit-supercomputer/, 17.8.2020.

[16] https://www.sciencedirect.com/topics/psychology/machine-learning, 17.8.2020.

[17] https://www.linkedin.com/pulse/machine-learning-what-milestones-everyone-should-know-bernard-marr, 17.8.2020.

[18] https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

[19] http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm, 12.8.2020.

[20] https://towardsdatascience.com/minimizing-the-cost-function-gradient-descent-a5dd6b5350e1, 12.8.2020.

[20] https://www.sciencedirect.com/topics/computer-science/supervised-learning, 13.8.2020.

[21] https://towardsdatascience.com/unsupervised-learning-clustering-60f13b4c27f1

[22] https://www.datarobot.com/wiki/unsupervised-machine-learning/, 13.8.2020.

[23] https://lawtomated.com/tag/supervised-learning/

[24] Tallorin L.i dr.: "Discovering de novo peptide substrates for enzymes using machine learning.", Nature, 2018.

[25] http://news.mit.edu/2020/artificial-intelligence-identifies-new-antibiotic-0220, 18.8.2020.
[26] Lata S.i dr.: "Analysis and prediction of antibacterial peptides.", BMC Bioinformatics 8, 2007.
[27] Lin Y.i dr.: "An advanced approach to identify antimicrobial peptides and their function types for penaeus through machine learning strategies.", BMC Bioinformatics 20, 2019.
[28] https://www.r-project.org/about.html, 20.8.2020.
[29] https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120512S.pdf, 20.8.2020.
[30] https://www.tiobe.com/tiobe-index/, 20.8.2020.
[31] https://www.rdocumentation.org/packages/Peptides/versions/2.4.2, 20.8.2020.
[32] https://www.tiobe.com/tiobe-index/python/, 20.8.2020.
[33] https://www.uniprot.org/, 20.8.2020.
[34] Aurélien Géron: "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", O'Reilly Media, 2013.
[35] van Westen, G.J.: "Benchmarking of protein descriptor sets in proteochemometric modeling (part 1): comparative study of 13 amino acid descriptor sets.", Journal of cheminformatics, 2013.
[36] van Westen, G.J.: "Benchmarking of protein descriptor sets in proteochemometric modeling (part 2): modeling performance of 13 amino acid descriptor sets.", Journal of cheminformatics, 2013.
[37] https://www.britannica.com/topic/Bell-Laboratories, 1.9.2020.
[38] https://www.svms.org/history.html, 1.9.2020.
[39] https://www.vebuso.com/2020/02/a-top-machine-learning-algorithm-explained-support-vector-machines-svms/, 1.9.2020.
[40] http://gregorygundersen.com/blog/2019/12/10/kernel-trick/, 1.9.2020.
[41] https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, 13.9.2020.
[42] https://stackoverflow.com/questions/58821599/splitting-a-data-set-for-k-fold-cross-validation-in-sci-kit-learn, 13.9.2020.
[43] https://openclassrooms.com/fr/courses/4470521-modelisez-vos-donnees-avec-les-methodes-ensemblistes/4664692-initiez-vous-aux-methodes-sequentielles-et-au-boosting, 13.9.2020.

## Sažetak

U ovom je radu predstavljeno svo znanje potrebno za razumijevanje teme strojnog učenja, te kako je strojno učenje primjenjivo u svrhe predviđanja antimikrobnih peptida. Također, opisan je postupak konstruiranja dosad nepostojećeg seta podataka putem istraživanja teme antimikrobnih peptida, pronalaska izvora podataka i značajki koje opisuju te podatke. Korištenjem standarnih alata, programskih jezika i knjižnica koje se upotrebljavaju na području strojnog učenja, stvoren je model koji sa točnošću od 91.5% predviđa ima li zadani peptid antimikrobnu aktivnost ili nema.

**Ključne riječi:** strojno učenje, antimikrobni peptidi, SVM model, Scikit-learn.

## Abstract

This paper defines all the knowledge needed to understand the topic of machine learning, and how machine can be leveraged for the purpose of predicting antimicrobial peptides. This paper also describes the construction of a dataset from scratch, through research of the topic of antimicrobial peptides, finding data sources, and discovering the features that describe the data. Using standardized tools, programming languages and machine learning libraries, a model was developed that predicts antimicrobial activity in peptides with an accuracy of 91.5%.

**Keywords:** machine learning, antimicrobial peptides, SVM model, Scikit-learn.