Univerzitet u Beogradu
Matematički fakultet

Master Rad

Predviđanje funkcija proteina metodama binarne klasifikacije

Autor:
Anja Bukurov

Mentor:
dr Jovana Kovačević

Članovi komisije:

dr Jovana Kovačević
prof. dr Gordana Pavlović-Lažetić
dr Mladen Nikolić

Beograd, 2019

UNIVERZITET U BEOGRADU

Sažetak

Matematički fakultet
Katedra za Računarstvo i informatiku

Predviđanje funkcija proteina metodama binarne klasifikacije

Anja Bukurov

Proteini su makromolekuli koji igraju važnu ulogu u organizmu svakog živog bića. To su biološki najaktivniji molekuli neophodni za izgradnju i funkcionisanje ćelija i imaju veliki broj esencijalnih funkcija. Struktura proteina zavisi od rasporeda aminokiselina i utiče na njegovu funkciju. Primarna struktura obuhvata sekvencu aminokiselina koje izgrađuju protein. Ona je osnovni izvor informacija o proteinu i njegovoj funkciji. Poznato je da proteini sa sličnim primarnim strukturama teže da obavljaju iste funkcije.

Svake godine sekvencira se veliki broj novih genoma čime raste broj novootkrivenih proteina. Funkcija proteina određuje se eksperimentalno što je skup i spor proces zbog čega se ulaže trud u razvoj računarskih metoda koje mogu da predvide funkciju proteina. Jedan pristup rešavanju ovog problema jeste predviđanje funkcije proteina. Ono može koristiti različite izvore informacija o proteinima poput interakcija između proteina, evolucione povezanosti i stukture proteina. U ovom radu razvijen je alat za određivanje funkcije proteina na osnovu njegove sekvence pomoću metoda binarne klasifikacije sa ciljem ispitivanja performansi jednostavnih metoda mašinskog ucenja na ovom kompleksnom problemu.

Sadržaj

Spisak slika

Spisak tabela

Glava 1

Uvod

Funkcionalna anotacija proteina može pomoći u dizajnu lekova za savremene bolesti jer mnoge od njih nastaju kao posledica izmene funkcije proteina usled mutacija [1]. Zbog toga, predviđanje funkcije proteina predstavlja jedan od najbitnijih zadataka u bioinformatici.

Metode za eksperimentalno određivanje funkcije proteina su spore u odnosu na brzinu sekvencioniranja genoma koje uvećava broj novih sekvenci. Do sada je za veoma mali broj proteina eksperimentalno određena funkcija zbog cene i trajanja tog procesa. Zbog toga se radi na razvijanju i unapređenju računarskih metoda za određivanje funkcije proteina. Proteini mogu imati više funkcija što omogućava sagledavanje problema predviđanja funkcije proteina kao problema višestruke klasifikacije.

Brzi razvoj računarskih metoda za predviđanje funkcije proteina doveo je do potrebe za njihovim poređenjem nad proteinima sa novoutvrđenim funkcijama. Zbog toga je kreiran eksperiment *Critical Assessment of Function Annotation* (CAFA) [2] koji se održava svake dve godine i gde autori prediktora šalju rezultate za veliki skup proteina za koje je funkcija nepoznata, pri čemu se za deo tog skupa funkcija određuje eksperimentalno pre evaluacije rezultata. Različiti algoritmi se evaluiraju prema sposobnosti da predvide koje funkcije obavljaju proteini. Preko 90% metoda poređenih u CAFA takmičenju koriste informacije o sekvenci proteina na neki način [3]. Jedan pristup je računanje sličnosti proteinskih sekvenci [4, 5, 6] ili nekih drugih osobina sekvence, poput učestalosti pojavljivanja k-grama [4] i obogaćivanje određenih podsekvenci u proteinima koje obavljaju određenu funkciju [7].

U ovom radu prikazan je razvoj alata za predviđanje funkcije proteina na osnovu njihove primarne strukture. Korišćene su metode binarne klasifikacije i to: metod potpornih vektora, logistička regresija i slučajne šume. Alat je razvijan u programskom jeziku Python.

U poglavlju 2 uvedeni su biološki pojmovi neophodni za razumevanje rada. U poglavlju 3 prikazan je način predstavljanja bioloških podataka u računaru, a onda su ukratko predstavljene metode binarne klasifikacije korišćene za razvoj prediktora. Zatim, u poglavlju 4, analizirani su podaci o proteinima i njihovim funkcijama iz CAFA preporučenog trening skupa, nakon čega je opisana implementacija alata za predviđanje funkcije proteina. Na kraju, u poglavlju 5 sumirani su rezultati koje su

prediktori dali na izdvojenom skupu proteina za testiranje, a zatim i poređenje sa aktuelnim rezultatima CAFA takmičenja.

Glava 2

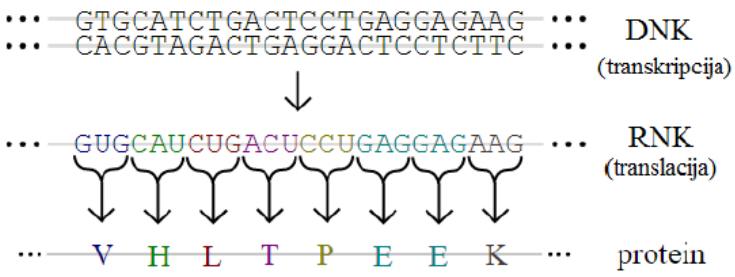
Proteini

Sva živa bića sastoje se iz ćelija. U ćelijama se neprestano odvijaju različiti procesi u kojima učestvuju nukleinske kiseline (dezoksiribonukleinka kiselina - DNK i ribonukleinska kiselina - RNK) i proteini. Unutar molekula DNK šifrovan je genetski materijal koji sadrži uputstva za sintezu proteina.

Proteini su makromolekuli koji igraju mnoge kritične uloge u organizmu. Sačinjavaju više od 50% suvog dela ćelije i važni su za njenu izgradnju i funkcionisanje. Kontrakcija mišića, strukturna podrška, ubrzavanje i usporavanje hemijskih reakcija, odbrana od virusa i bakterija samo su neke od mnogobrojnih uloga koje proteini obavljaju [8, 9].

2.1 Sinteza proteina

DNK sadrži informacije koje su neophodne ćeliji za izgradnju veoma važnog tipa molekula - proteina. Proteini se sintetišu prilikom genske ekspresije i to u dva koraka: transkripcija i translacija (slika 2.1). Prvi korak je dekodiranje genske poruke, prilikom čega se od DNK sekvence dobija RNK sekvenca. U sastav obe nukleinske kiseline ulazi 4 nukleotida i oni su prikazane u tabeli 2.1. S obzirom da su tri nukleotidne baze iste, proces transkripcije sastoji se iz zamene svakog molekula T molekulom U.

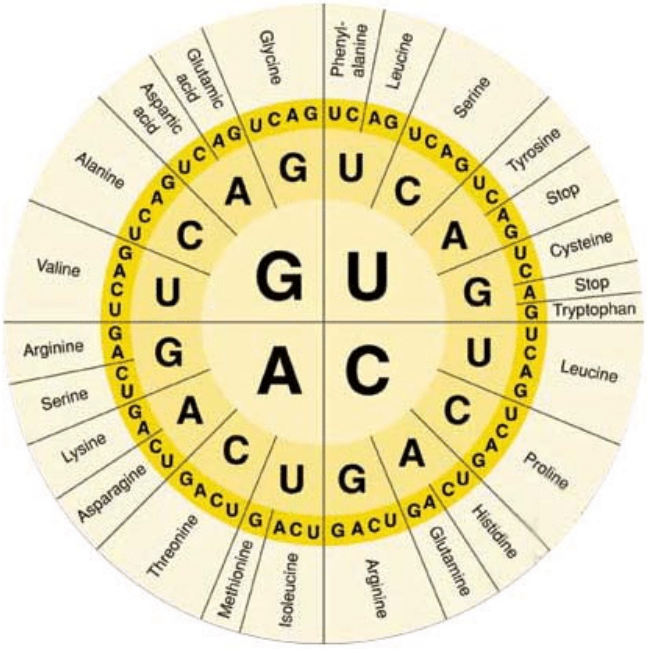


Slika 2.1: Prikaz procesa sinteze proteina [9]

DNK	adenin (A)	guanin (G)	citozin (C)	timin (T)
RNK	adenin (A)	guanin (G)	citozin (C)	uracil (U)

Tabela 2.1: Prikaz nukleinskih kiselina sa nukeotidima koji ih grade

Sledeći korak, proces translacije, jeste grupisanje aminokiselina kako bi se dobio protein. Genetski kod se čita u grupama od 3 nukleotida koje nazivamo *kodoni*. Svaki kodon odgovara tačno jednoj aminokiselini ili služi da označi kraj sekvence (stop kodon). Na primer, kodon **GUA** kodira aminokiselinu valin, dok kodon **UAG** označava kraj sekvence. Na slici 2.2 je dat šematski prikaz svih kodona i odgovarajućih aminokiselina. Jedan po jedan, kodoni se prevode u odgovarajuće aminokiseline čime se dobija sekvenca aminokiselina koja čini protein [10, 11].



Slika 2.2: Prikaz kodona i odgovarajućih aminokiselina

2.2 Aminokiseline

Aminokiseline su organska jedinjenja koja se sastoje od karboksilne grupe ($COOH$), aminogrupe (NH_2) i bočnog lanca (R-grupa) koji je vezan za α-ugljenikov atom i karakterističan je za svaku aminokiselinu. Postoji 20 standardnih aminokiselina i one su prikazane u tabeli 2.2. Pod standardnim aminokiselinama podrazumevaju se one aminokiseline za koje postoji najmanje jedan specifičan kodon u genetskom kodu [12, 13, 14].

Aminokiselina	Oznaka	Simbol	Aminokiselina	Oznaka	Simbol
Alanin	ALA	A	Arginin	ARG	R
Asparagin	ASN	N	Asparaginska kiselina	ASP	D
Cistein	CYS	C	Glutamin	GLN	Q
Glutaminska kiselina	GLU	E	Glicin	GLY	G
Histidin	HIS	H	Izoleucin	ILE	I
Leucin	LEU	L	Lisin	LYS	K
Metionin	MET	M	Fenilalanin	PHE	F
Prolin	PRO	P	Serin	SER	S
Treonin	THR	T	Triptofan	TRP	W
Tirosin	TYR	Y	Valin	VAL	V

TABELA 2.2: Prikaz standardnih aminokiselina sa oznakama i simbolima

Aminokiseline možemo podeliti u nekoliko grupa prema osobinama bočnog lanca [12, 13, 15]:

1. **aminokiseline sa nepolarnim bočnim lancem**
   Bočni lanac ovih aminokiselina ne može da otpušta niti da vezuje protone, kao ni da učestvuje u vodoničnim ili jonskim vezama. Zbog svoje nepolarnosti, one su hidrofobne i obično popunjavaju praznine u unutrašnjosti proteina čime doprinose oblikovanju njegove strukture. U ovu grupu ubrajamo 7 standardnih aminokiselina: alanin, valin, leucin, izoleucin, metionin, fenilalanin, triptofan.

2. **aminokiseline sa nenaelektrisanim polarnim bočnim lancem**
   R-grupa aminokiselina iz ove grupe može da gradi vodonične veze sa molekulima vode što ih čini rastvorljivijim u odnosu na aminokiseline iz prethodne grupe. Zbog polarnosti, ove aminokiseline se obično nalaze na spoljašosti proteina. Ova grupa obuhvata 6 standardnih aminokiselina i to: serin, treonin, tirozin, asparagin, glutamin i cistein.

3. **aminokiseline sa naelektrisanim polarnim bočnim lancem**
   U ovu grupu spadaju veoma hidrofilne aminokiseline zbog čega se one nalaze na površini proteina. Dodatno ih možemo podeliti na kisele i bazne aminokiseline. Kisele imaju jednu karboksilnu grupu više i imaju negativno naelektrisanje, dok su bazne aminokiseline pozitivno naelektrisane. Asapraginska i glutaminska kiselina su kisele aminokiseline, a lizin, histidin i arginin spadaju u bazne aminokiseline.

4. **konformaciono važne aminkiseline**
   Preostale dve standardne aminokiseline, glicin i prolin, se po svojoj strukuturi razlikuju od ostalih. Glicin nema bočni lanac i može da se prilagođava konformacijama koje su nedostupne drugim aminokiselinama. Prolin sadrži jedan heterociklički prsten i u svojoj strukuturi sadrži sekundarnu amino grupu.

Bilo koje dve aminokiseline mogu izgraditi veći molekul, dipeptid, formiranjem peptidne veze između njih. Peptidna veza se ostvaruje između atoma ugljenika iz

karboksilne grupe i atoma azota iz amino grupe. Peptidne veze omogućavaju stvaranje lanaca aminokiselina, tzv. polipeptida. Peptidna veza nastaje reakcijom dve aminokiseline pri čemu se spajaju karboksilna grupa jedne sa amino grupom druge aminokiseline uz izdvajanje vode. Prilikom tog vezivanja pojavljuje se niz koji se zove kičma polipeptidnog lanca koji čine ugljenikov atom karboksilne grupe, atom azota aminogrupe i α-ugljenikov atom. To je osnovni niz i isti je za sve proteine, a oni se međusobno razlikuju po bočnim lancima aminokiselina [13, 15].

Peptide možemo podeliti prema broju aminokiselina koje sadrže i to na oligopeptide i polipeptide. Oligopeptidi su sačinjeni od najviše 10 aminokiselina, dok polipeptidi sadrže do 100 aminokiselina. Jedinjenja sa više od 100 aminokiselina u lancu spadaju u proteine [13].

2.3 Struktura proteina

U sastav proteina ulazi 20 standardnih aminokiselina. Sekvenca aminokiselina, koja se formira peptidnim vezama, specifična je za svaki protein. Ona je primarni izvor informacija o proteinu i njegovoj funkciji. Složenost proteinske strukture najbolje se analizira kroz četiri nivoa: primarna, sekundarna, tercijerna i kvaterna struktura [12].

Primarna strktura Jedinstveni redosled aminokiselina koje su povezane peptidnom vezom kako bi formirale protein čini primarnu strukturu proteina. Proteini koji imaju slične sekvence često imaju i slične osobine i funkcije. Zbog toga je poređenje sekvenci prvi korak u izučavanju proteina. Razumevanje primarne sekvence je bitno zbog mnogih genetskih bolesti koje za posledicu imaju proteine sa neispravnim sekvencama što vodi do pogrešnog savijanja i nefunkcionalnog proteina [12, 13, 15].

Sekundarna struktura Polipeptidni lanac ne zauzima bilo kakav oblik u prostoru već ima opšti raspored aminokiselina koje se u lancu nalaze jedna blizu druge. Taj raspored označava sekundarnu strukturu proteina i podrazumeva savijanje ili uvijanje polipeptidnog lanca. Lanac može da uzme oblik α-heliksa (engl. α-helix), β-traka (engl. β-sheet) ili β-okreta (engl. β-turn). α-heliks je periodična struktura u kojoj se kičma proteina spiralno uvrće, a bočni lanci aminokiselina izviruju izvan nje. β-trake formiraju se kao parovi lanaca aminokiselina koji se uzdužno vezuju vodoničnim vezama. β-okret menja pravac polipeptidnog lanca čime mu pomaže da dobije kompaktan, loptast oblik [12, 15, 16].

Tercijarna struktura Prostorna struktura čitavog molekula proteina predstavlja tercijarnu strukturu. Hidrofobni bočni lanci nepolarnih aminokiselina teže da budu unutar molekula proteina zaštićeni od vode, dok se kisele i bazne aminokiseline obično nalaze na površini proteina pošto su hidrofilne. α-heliksi i β-trake služe da obezbede maksimalan broj vodoničnih veza u unutrašnjosti molekula, čime sprečavaju da se molekuli vode vežu za hidrofilne grupe i time naruše integritet proteina [12, 13].

Kvaternarna struktura Mnogi proteini su formirani grupisanjem više savijenih polipeptidnih lanaca. Pojedinačnu komponentu nazivamo podjedinica. One mogu biti međusobno različite ili potpuno iste. Raspored ovih podjednica predstavlja kvaternarnu strukturu. U kvaternarnoj strukturi podjedinice se međusobno drže zajedno nekovalentnim interakcijama i kovalentnim vezama [9, 12, 16].

2.4 Uloga proteina

Proteini su najbrojniji i funkcionalno najrazličitiji molekuli u živom svetu. Svaki od njh ima veoma važnu ulogu u organizmu. Na primer:

- Enzimi su proteini koji olakšavaju hemijske reakcije. Učestvuju u skoro svim reakcijama u ćelijama i pomažu u izgradnji novih molekula.
- Antitela su proteini koje proizvodi imuni sistem da bi pomogli u odstranjivanju stranih supstanci i kako bi se borile protiv infekcija. Oni se vezuju za nepoznate čestice, poput bakterija i virusa čime brane telo.
- Kontrakcijski proteini učestvuju u kontrakcijama mišića i kretanju.
- Strukturni proteini su vlaknasti i obezbeđuju strukturu i podršku ćelijama. Učestvuju u izdgradnji kose, noktiju, kože, kostiju, itd.
- Transportni proteini prenose molekule kroz telo.
- Hormonski proteini prenose signale kako bi upravljali biološkim procesima među ćelijama, tkivima i organima.
- Skladišni proteini čuvaju aminokiseline za kasniju upotrebu [17, 18, 19].

Glava 3

Podaci i metode binarne klasifikacije

U ovom poglavlju biće opisani korišćeni podaci i način njihovog predstavljanja u računaru. Zatim će ukratko biti opisane metode binarne klasifikacije koje su korišćene za predviđanje funkcija proteina.

3.1 Podaci

Podaci o proteinima mogu se pronaći u biomedicinskim bazama podataka, a neke od njih prikazane su u tabeli 3.1.

Baza podataka	URL	Opis
UniProtKB	uniprot.org	Proteinske sekvence i funkcije proteina
PFAM	pfam.xfam.org	Proteinske familije
PDB	wwpdb.org	Eksperimentalno utvrđene strukture
ModBase	modbase.compbio.ucsf.edu	Strukture utvrđene predviđanjem
I2D	ophid.utoronto.ca	Interakcije između proteina
GEO	www.ncbi.nlm.nih.gov/geo	Podaci o genskoj ekspresiji
PRIDE	www.ebi.ac.uk/pride	Podaci dobijeni masenom sprektometrijom

Tabela 3.1: Prikaz nekih javno dostupnih biomedicinskih baza podataka [8, 9]

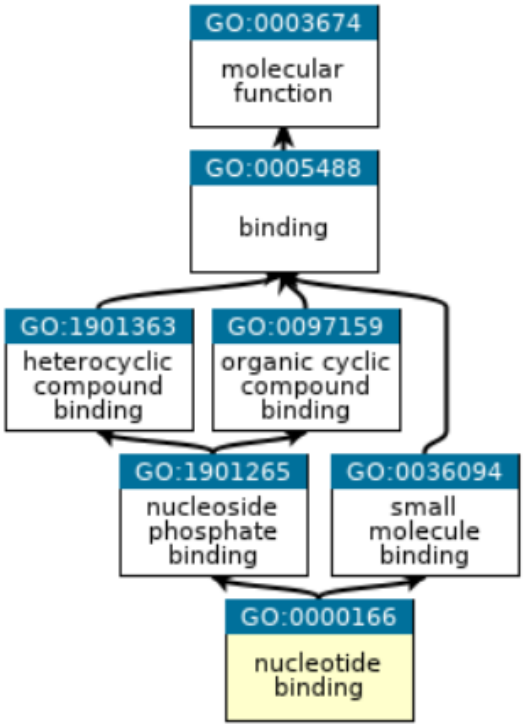
3.1.1 Predstavljanje proteina

Kao što je već rečeno, proteini su izgrađeni od 20 različitih aminokiselina, a svaka aminokiselina ima jedinstveni simbol (tabela 2.2). Najjednostavniji način za predstavljanje proteina u računaru jeste kao niska karaktera nad azbukom $\Sigma = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. Nad ovako predstavljenim proteinima mogu su koristiti algoritmi za rad sa tekstom kao što je poravnanje sekvenci [8].

3.1.2 Predstavljanje funkcije proteina

Da bi predviđanje funkcija proteina bilo moguće neophodno je da postoje dobro definisani odnosi između funkcija. Sistem za predstavljanje funkcije proteina koji se trenutno najviše koristi je *Gene Ontology*. Ovaj sistem deli funkcije proteina na tri ontologije: biološki procesi (BPO), molekulske funkcije (MFO) i ćelijske komponente (CCO).

Svaka ontologija predstavljena je kao usmereni aciklički graf gde su čvorovima pridruženi nazivi funkcija, a grane koje ih povezuju definišu relaciju ,,is_a". Hijerarhijska organizacija obezbeđuje da svaki čvor ima specifičniju funkciju od roditeljskog čvora. U ovoj hijerarhiji jedan čvor može imati više roditeljskih čvorova što je prikazano na slici 3.1 [9, 20]. U korenu svake ontologije nalazi se funkcija sa nazivom te ontologije, a u listovima su najspecifičnije fukcije.



Slika 3.1: Prikaz svih predaka lista označenog funkcijom ,,nucleotid binding" u ontologiji molekulskih funkcija [21]

3.2 Nebalansirani skupovi podataka

Nebalansirani skupovi podataka podrazumevaju skupove u kojima nije svaka klasa predstavljena jednako tj. nema jednak ili približan broj instanci za svaku klasu. Nebalansirani podaci ne moraju biti loša stvar - u praksi su skupovi često nebalansirani do nekog stepena. Međutim, problem nastaje kada je stepen nebalansiranosti veliki. Na primer, u detekciji prevara ili donošenju medicinskih dijagnoza, nebalansiranost je

velika. Većina transakcija koja se obavlja neće biti prevare, a postoje i slučajevi kada jeste prevara, iako je retko [22, 23].

Algoritmi mašinskog učenja napravljeni su tako da minimizuju grešku. S obzirom da će, u slučaju velike nebalansiranosti podataka, mnogo veća šansa biti da instanca pripada većinskoj klasi, algoritmi će češće klasifikovati podatke baš u tu klasu.

Jedan od pristupa ovakvom problemu jeste pravljenje novog skupa podataka od postojećeg u cilju balansiranja klasa. Postoje dva popularna načina da se ovo učini. Jedan podrazumeva dodavanje novih instanci, a drugi uklanjanje postojećih instanci. Prvi pristup uvećava broj instanci klase u manjini u trening skupu. Najbolje rešenje jeste dodavanje novih instanci klase u manjini kako bi se nebalansiranost ublažila ili potpuno odstranila. Međutim, najčešće novih instanci nema pa se prave veštačke instance na osnovu postojećih. Time se ne gube informacije iz originalnog trening skupa, ali je pristup sklon preprilagođavanju. Sa druge strane, uklanjanje postojećih instanci podrazumeva smanjenje skupa većinske klase čime se dobija izbalansiran skup, ali može se desiti da se izgube bitne informacije [23].

Prilikom predviđanja funkcije proteina, problem nebalansiranih skupova podataka javlja se prilikom podele proteina na one koji vrše neku funkciju iz ontologije i na one koji je ne vrše. Proteina koji se svrstavaju u pozitivnu klasu ima mnogo manje od onih iz negativne klase i broj opada što se dublje ide kroz ontologiju. U glavi 4 su detaljnije opisani podaci korišćeni u radu i preciznije je prikazana podela podataka.

3.3 Binarni klasifikatori

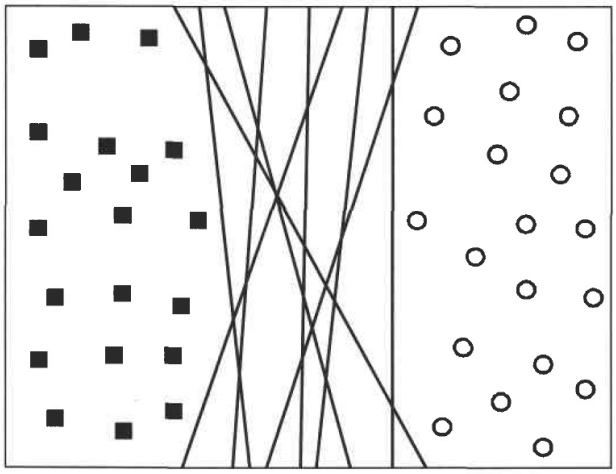
Klasifikacija, odnosno, zadatak dodeljivanja objekata jednoj od više predefinisanih kategorija, rasprostranjen je problem koji obuhvata mnoštvo različitih primena. Primeri uključuju otkrivanje spam poruka na osnovu zaglavlja poruke i njenog sadržaja, kategorisanje ćelija kao malignih ili beningnih na osnovu rezultata magnetne rezonance, klasifikaciju galaksija na osnovu njihovog oblika, itd. Binarna klasifikacija je slučaj klasifikacije u kojoj postoje tačno dve predefinisane kategorije u koje treba razvrstati date objekte. Obično se za jednu kategoriju kaže da je to pozitivna klasa, a za drugu da je negativna.

Svaki klasifikator upotrebljava algoritam za učenje kako bi odredio model koji najbolje odgovara vezi između skupa atributa i klase ulaznih podataka. Model koji algoritam generiše trebalo bi da odgovara ulaznim podacima kao i da tačno predviđa klasu slogova koje ranije nije video.

3.3.1 Metod potpornih vektora

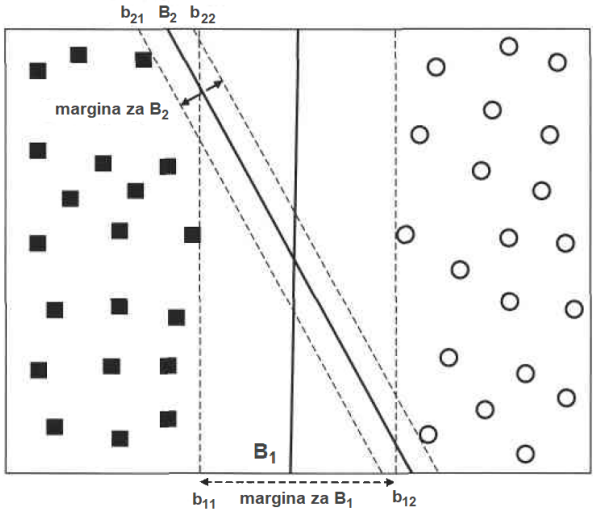
Metod potpornih vektora (engl. *support vector machine*) je tehnika binarne klasifikacije bazirana na pronalasku razdvajajuće hiperravni. Ideja je pronaći takvu hiperravan da su instance iste klase sa iste strane ravni. Sa tako postavljenim uslovom,

razdvajajućih hiperravni može biti više od jedne što je prikazano na slici 3.2. Iako svaka od ravni razdvaja podatke bez greške, nema garancije da će se podjednako dobro ponašati sa novim podacima koje je potrebno klasifikovati.



Slika 3.2: Primeri razdvajajućih hiperravni [24]

Na slici 3.3 izdvojene su dve hiperravni B_1 i B_2 i za svaku su dodate dve pomoćne hiperavni b_{i1} i b_{i2}. Pomoćne ravni paralelne su glavnoj i pomerene u levu ili desnu stranu do najbliže instance jedne klase. Rastojanje između pomoćnih ravni odnosno rastojanje između najbližih instanci iz obe klase u odnosu na hiperravan naziva se *margina*, a instance oslonjene na hiperravni su *potporni vektori*. Cilj je pronaći hiperravan koja maksimizuje veličinu margine. Sa slike 3.3 jasno se vidi da je bolja hiperravan B_1. Jednačina optimalne hiperravni predstavlja klasifikacioni model. Korak klasifikacije nepoznate instance sastoji se iz izračunavanja njenog rastojanja od hiperravni na osnovu čega se određuje klasa kojoj instanca pripada [24, 25].



Slika 3.3: Margine razdvajajućih hiperravni [24]

Jednačina hiperravni je

$$w \cdot x + w_0 = 0$$

gde je w_0 slobodan član. Na osnovu jednačine rastojanja tačke od hiperravni

$$\frac{|w \cdot x + w_0|}{||w||_2}$$

i činjenice da za svaku od tačaka sa ovih hiperravni važi $w \cdot x + w_0 = 1$, dobija se da je ukupno rastojanje između klasa, u pravcu normalnom u odnosu na optimalnu hiperravan $\frac{2}{|w|}$. Tako se optimalna hiperravan dobija pronalaženjem koeficijenata koji maksimizuju ovaj izraz pod uslovima da su sve tačke sa pravih strana te hiperravni odnosno da su podaci linearno razdvojivi. Optimizacioni problem može da se zapiše i kao problem minimizacije i glasi:

$$\min_{w,w_0} \frac{||w||_2}{2}$$
$$y_i(w \cdot x_i + w_0) \geq 1 \qquad i = 1, \ldots, N$$

Dodatni uslov će obezbediti da sve tačke budu na većem rastojanju od hiperravni u odnosu na potporne vektore [26].

S obzirom da je čest slučaj da podaci nisu linearno razdvojivi potrebno je prihvatiti greške tj. dozvoliti da se neka instanca nađe sa pogrešne strane razdvajajuće hiperravni. U tu svrhu uvode se nove promenljive, ξ_i koje mere koliko je svaka pogrešno klasifikovana instanca udaljena od hiperravni. Taj metod nazivamo metod potpornih vektora sa *mekom marginom*. Optimizacioni problem se menja:

$$\min_{w,w_0} \frac{||w||_2}{2} + C\sum_{i=0}^{N} \xi_i$$
$$y_i(w \cdot x_i + w_0) \geq 1 - \xi_i \qquad i = 1, \ldots, N$$
$$\xi_i \geq 0 \qquad i = 1, \ldots, N$$

Metaparametar C kontroliše koliki značaj imaju greške. Ukoliko je vrednost jednaka nuli, onda greške ne igraju nikakvu ulogu, a ako je vrednost velika, onda su greške veoma važne, a pravac hiperravni i širina pojasa nisu bitne [26].

3.3.2 Logistička regresija

Logistička regresija (engl. *logistic regression*) je statistički zasnovan metod za analizu skupa podataka u kom jedna ili više nezavisnih promenljivih određuju ishod. Osnovna pretpostvaka je Bernulijeva rasporedela ciljne promenljive y pri datim vrednostima atributa x odnosno, za date vrednosti atributa x, postoji parametar $\mu \in [0, 1]$ tako da važi:

$$p(y = 1|x) = \mu$$

odakle je $p(y = 0|x)$ jednoznačno određeno. Zadatak je sličan kao u prethodnoj metodi, potrebno je pronaći razdvajajuću hiperravan koja deli podatke tako da sa jedne strane budu instance iste klase. Najjednostavniji je linearan model:

$$f(x) = w \cdot x$$

S obzirom da funkcija uzima vrednosti iz intervala $[-\infty, \infty]$, a parametar μ mora imati vrednost iz intervala $[0, 1]$ da bi verovatnoća bila ispravno definisana, ovakav model nije prihvatljiv. Zbog toga se vrednost linearnog modela transformiše monotonom funkcijom u interval $[0, 1]$. U te svrhe, najčešće se koristi sigmoidna funkcija:

$$\sigma(t) = \frac{1}{1 + exp(-t)}$$

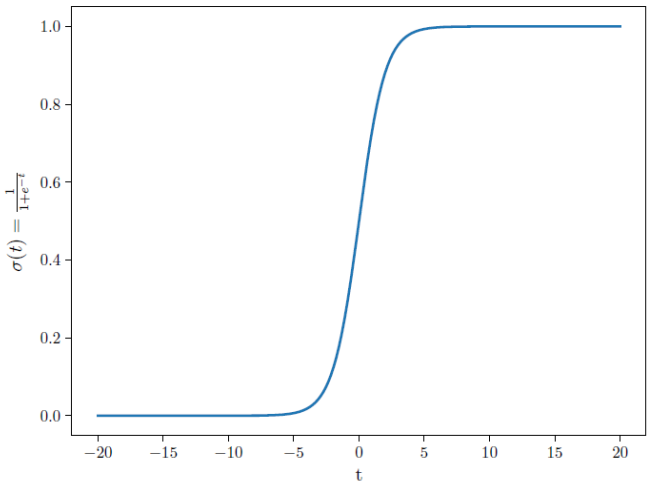
čiji je grafik prikazan na slici 3.4. Nakon transformacije vrednosti linearnog modela sigmoidnom funkcijom, model logističke regresije određen je relacijom:

$$p_w(y = 1|x) = \sigma(w \cdot x)$$

Iz toga se može odrediti i puna specifikacija problema:

$$p_w(y|x) = \sigma(w \cdot x)^y (1 - \sigma(w \cdot x))^{1-y}$$

Verovatnoća da instanca pripada jednoj klasi je veća što je instanca dalje od hiperravni sa odgovarajuće strane [26].



Slika 3.4: Grafik sigmoidne funkcije [26]

Ocena parametara ovog modela zasniva se na principu maksimalne verodostojnosti. Uz pretpostavku nezavisnosti instanci, funkcija verodostojnosti zadata je izrazom:

$$\mathcal{L}(w) = \prod_{i=1}^{N} p_w(y_i|x_i)$$

i potrebno je rešiti problem:

$$\max_{w} \mathcal{L}(w)$$

Ukoliko se pređe na negativan logaritam verodostojnosti optimizacioni problem postaje:

$$\min_{w} -\sum_{i=0}^{N}[y_i \log f_w(x_i) + (1-y_i)\log(1-f_w(x_i))]$$

3.3.3 Slučajne šume

Metod slučajnih šuma (engl. *random forests*) spada u grupu metoda specijalno dizajnirane za stabla odlučivanja. Ona kombinuje predviđanja više različitih stabala, gde je svako stablo generisano na osnovu vrednosti nezavisnog skupa slučajno odabranih vektora.

Stabla odlučivanja

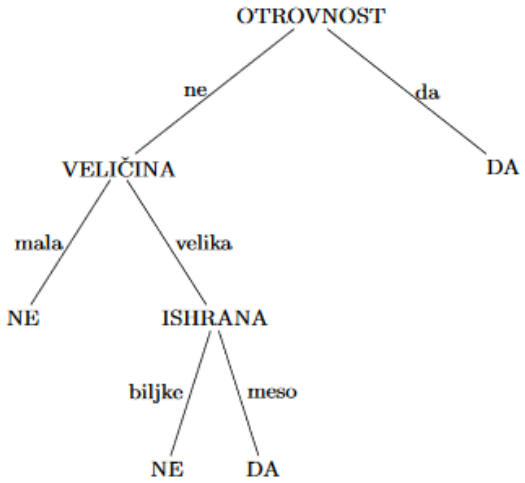
Problem klasifikacije rešava se postavljanjem pažljivo sastavljenih pitanja o atributima podataka. Pitanja se postavljaju sve dok nije moguće zaključiti klasu date instance. Niz pitanja i odgovora organizovan je u hijerarhijsku strukturu koja se sastoji iz čvorova i direktinih grana. Na slici 3.5 je prikazano stablo odlučivanja koje određuje da li je životinja opasna ili ne na osnovu podataka o otrovnosti, veličini i ishrani.

Svaki list u stablu ima dodeljenu klasu, a unutrašnji čvorovi i koren sadrže uslove koji razdvajaju instance sa različitim karakteristikama. Grane predstavljaju odgovor na pitanje čvora iz kog izlaze. Jednom kada je stablo konstruisano, klasifikacija je jednosmerna. Kreće se od korenog čvora i za konkretnu instancu odgovara se na pitanja koja se nalaze u čvorovima praćenjem odgovarajućih grana sve do listova koje sadrže konačan odgovor. Klasa koja je pridružena listu dodeljuje se instanci [24].

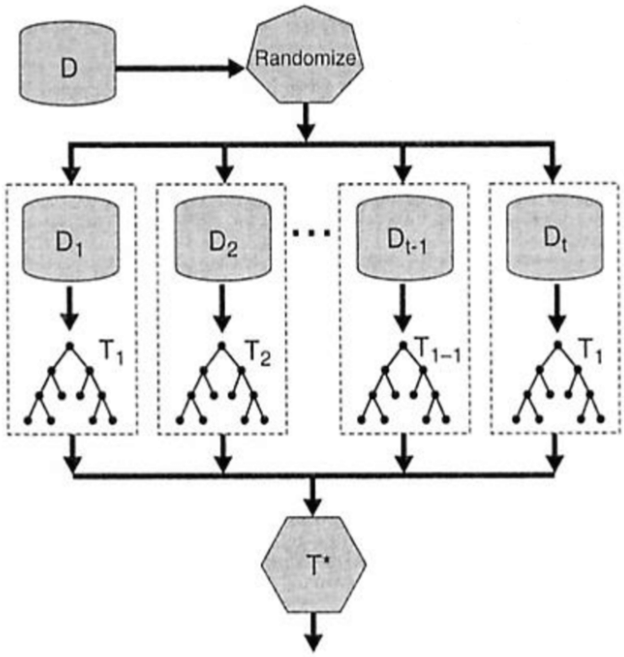
Predviđanje korišćenjem slučajnih šuma

Osnovna ideja je kombinovanje više stabala odlučivanja u jedan model. Stabla koja se kombinuju treniraju se nad nezavisnim, slučajno odabranim podskupovima podataka, a može se koristiti i slučajno odabran podskup atributa. Obučavaju se nad različitim skupovima kako bi njihove greške bile što slabije korelisane [26].

Instanca se klasifikuje glasanjem. Svako od konstruisanih stabala klasifikuje instancu u jednu od dve klase, a zatim se svi odgovori broje i odgovor klasifikatora slučajne šume je ona klasa koja ima više glasova tj. ona klasa koju je više stabala predvidelo. Slika 3.6 ilustruje proces treniranja i klasifikacije.



Slika 3.5: Primer stabla odlučivanja koje određuje da li je životinja opasna ili ne [27]



Slika 3.6: Primer slučajne šume [24]. Prvo se iz početnog skupa slučajno odabiraju instance i formira se podskup za svako stablo. Zatim se nad odgovarajućim podskupovima treniraju stabla. Svako stablo klasifikuje nepoznatu instancu i odgovori se kombinuju u jedan, konačan, odgovor modela slučajnih šuma.

3.4 Evaluacija modela binarne klasifikacije

Metode binarne klasifikacije za test instancu daju jedan od dva moguća odgovora, na primer, ,,da" ili ,,ne". Pošto se jedna klasa obično posmatra kao pozitivna a druga kao negativna, neka u ovom primeru ,,da" bude pozitivna klasa, a ,,ne" neka bude

negativna klasa. Prilikom ocenjivanja kvaliteta klasifikacionog modela od značaja su 4 veličine:

- tp - broj instanci za koju je predviđena pozitivna klasa i čija je stvarna klasa pozitivna
- tn - broj instanci za koju je predviđena negativna klasa i čija je stvarna klasa negativna
- fp - broj instanci za koju je predviđena pozitivna klasa, a čija je stvarna klasa negativna
- fn - broj instanci za koju je predviđena negativna klasa, a čija je stvarna klasa pozitivna.

Kada su definisane ove veličine, mogu se odrediti i neke mere kvaliteta kao što je, na primer, tačnost modela. Tačnost (*engl. accuracy*) određuje koliko je instanci tačno klasifikovano u odnosu na ukupan broj instanci i definiše se formulom:

$$\text{accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

Ova metrika se često koristi u mašinskom učenju, međutim, ona ne daje uvek dobru ocenu metoda. U slučaju nebalansiranih klasa[1] model može da daje visoku vrednost za tačnost, a da ipak loše predviđa. Razlog je to što često loši modeli predviđaju skoro uvek samo jednu, dominantnu klasu, a pošto je instanci dominantne klase značajno više, većina instanci će biti ispravno klasifikovana. Međutim, cilj je napraviti model koji će biti uspešan u klasifikovanju obe klase, a ne samo jedne [28].

Zbog toga se definišu još neke mere kvaliteta modela. Prva je preciznost (*engl. precision*), a druga je odziv (*engl. recall*) i definišu se formulama:

$$\text{precision} = \frac{tp}{tp + fp} \qquad \text{recall} = \frac{tp}{tp + fn}$$

Preciznost određuje koliko je pozitivnih instanci ispravno klasifikovano u odnosu na ukupan broj instanci koje su klasifikovane kao pozitivne. Sa druge strane, odziv određuje udeo ispravno klasifikovanih pozitivnih instanci u odnosu na ukupan broj pozitivnih instanci u skupu.

Preciznost i odziv pojedinačno nisu korisne [22]:

- visok odziv i preciznost: model dobro klasifikuje podatke iz određene klase
- nizak odziv i visoka preciznost: model retko klasifikuje instance u određenu klasu, ali kad to uradi može mu se verovati da je ispravno
- visok odziv i niska preciznost: model često klasifikuje instance u određenu klasu, ali uključiće i instance drugih klasa pa mu se ne može mnogo verovati

[1] Pod nebalansiranim klasama podrazumeva se da u skupu podataka postoji mnogo više instanci koji pripadaju jednoj klasi u odnosu na broj instanci koje pripadaju drugoj klasi.

- nizak odziv i preciznost: model loše klasfikuje instance u određenu klasu.

Ukoliko su sve instance klasifikovane kao pozitivne, odziv će biti maksimalan, međutim preciznost će biti veoma loša, dok sa druge strane, ukoliko su sve instance klasifikovane kao negativne model ne greši i preciznost je maksimalna, ali odziv je veoma loš. Stoga ima smisla posmatrati ih zajedno što se često radi određivanjem njihove harmonijske sredine koja je nazvana f_1-mera [26]:

$$f_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Još jedna mera kvaliteta koja se koristi prilikom evaluacija klasifikacionih modela je površina ispod ROC krive (*engl. area under the curve*). Pretpostavlja se da klasifikator počiva na nekom modelu koji različitim instancama pridružuje neke skorove. Obično se takav skor prevodi u klasu nekom vrstom zaokruživanja u odnosu na neki prag [26]. ROC kriva je grafik koji prikazuje performanse klasifikacionog modela pri svim pragovima. Ona prikazuje dve parametra: meru stvarno pozitivnih (*engl. true postive rate*) i meru lažno pozitivnih instanci (*engl. false positive rate*) koje se računaju po formulama [29]:

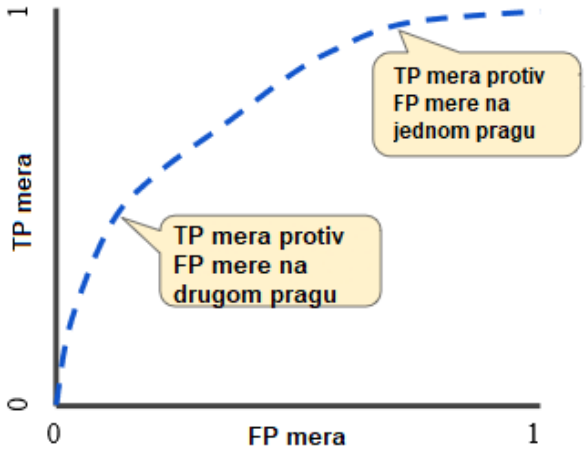
$$\text{tpr} = \frac{tp}{tp + fn} \qquad \text{fpr} = \frac{fp}{fp + tn}$$

Smanjenje praga rezultuje povećanjem pozitivno klasifikovanih instanci čime rastu i veličine tp i fp. Slika 3.7 prikazuje primer ROC krive. Površina ispod ROC krive meri dvodimenzionalni prostor ispod krive od tačke $(0, 0)$ do $(1, 1)$ i daje uprosečenu meru performansi modela za sve pragove. Može se interpretirati kao verovatnoća da model daje viši skor pozitivnoj instanci nego negativnoj. [29].

Površina ispod ROC krive govori koliko je model uspešan u razdvajanju klasa. Što je veća vrednost, model bolje klasifikuje negativne instance kao negativne, a pozitivne kao pozitivne. Model za koji je ova mera dala vrednost blisku 1 je odličan model koji dobro razdvaja klase. Sa druge strane, vrednost bliska 0 označava slab model koji loše razdvaja klase i to uglavnom klasifikuje instance u suprotnu klasu. Ukoliko je vrednost ove mere za neki model jednaka 0.5 to znači da model nije naučio da razdvaja klase [30].

3.5 Analiza glavnih komponenti

Ideja analize glavnih komponenti je smanjenje dimenzionalnosti skupa podataka koji su korelisani, a da se pritom varijabilnost u skupu što bolje očuva. Dimenzionalnost se definiše kao broj atributa koji su prisutni u podacima. Korelacija pokazuje koliko su dve promenljive slične. Varijacija promenljivih nosi više infomacija, ali koreliralnost atributa ukazuje na to da nose manje informacija nego što njihov broj sugeriše. Potrebno je pronaći skup vektora takvih da je varijansa projekcija podataka na prostor koji razapinju najveća, a da kovarijacije između projekcija podataka



Slika 3.7: ROC kriva [29]

nema. Dodatno se odbacuju vektori duž kojih je varijacija najmanja, a odabrani vektori se nazivaju *glavne komponente*. Kako bi projekcije bile nekorelirane, konstruišu se ortogonalni vektori. Ortogonalnost označava da promenljive nisu korelirane tj. da je korelacija između svaka dva para promenljivih jednaka 0. Može se reći da analiza glavnih komponenti (*engl. Principal component analysis*) pruža projekciju ili senku objekta posmatranog iz njegove najbogatije perspektive [24, 31, 32].

Matrica kovarijacije je matrica koja se sastoji od kovarijacija između parova promenljivih. Element σ_{ij} označava kovarijansu između i-te i j-te promenljive. Sopstveni vektor je nenula vektor v kvadratne matrice A ukoliko važi:

$$Av = \lambda v$$

Prva komponenta se bira tako da nosi najviše informacija odnosno, da održi najveću varijansu. Vektori koji se biraju su spostveni vektori matrice kovarijacije zbog čega su ortogonalni.

Glavne komponente se predstavljaju kao linearne kombinacije posmatranih promenljivih. Izlaz algoritma su glavne komponente kojih ima manje ili jednako početnom broju promenljivih. Ukoliko želimo da smanjimo dimenzionalnost taj broj će svakako biti manji [32].

U ovom radu, proteini su predstavljeni nizovima dimenzije 20^3, što je opisano u sekciji 4.1.1. Veliki broj atributa za svaki protein uticao je na performanse metode potpornih vektora zbog čega je primena ovog algoritma bila neophodna. Više reči o tome biće u poglavlju 4.3.

Glava 4

Implementacija predviđanja funkcije proteina

Predviđanje funkcije proteina vršeno je metodama binarne klasifikacije i to metodom potpornih vektora, slučajnim šumama i logističkom regresijom. Trenirani su binarni klasifikatori za pojedinačne funkcije ontologije molekulskih funkcija. Ulaz predstavljaju sekvence proteina za koje je eksperimentalno određeno da li obavljaju ili ne obavljaju konkretnu funkciju. Odgovor koji svaki od klasifikatora daje je da li zadati protein izvršava odgovarajuću funkciju ili ne.

Kada su svi modeli za jednu od metoda istrenirani, ujedinjeni su u jedinstveni izlaz za jedan prediktor. Prilikom predviđanja funkcije jednog proteina, prediktor testira proteinsku sekvencu nad svakim binarnim klasifikatorom, a odgovore koje dobija spaja u konačan odgovor - podgraf ontologije koji predstavlja funkciju zadatog proteina. Na taj način dobijena su tri prediktora, po jedan za svaku navedenu metodu.

4.1 Podaci

Podaci o proteinima korišćeni u ovom radu preuzeti su sa adrese https://biofunctionprediction.org/cafa-targets/CAFA3_training_data.tgz. Oni su podeljeni u dve datoteke:

1. **uniprot_sprot_exp.fasta** - proteini i njihove sekvence,
2. **uniprot_sprot_exp.txt** - proteini i eksperimentalno utvrđene funkcije koje obavljaju.

Dodatne informacije o organizmima iz kojih proteini potiču preuzete su sa `https://www.uniprot.org/` i to za organizme:

- čovek (human)
- miš (mouse)
- pacov (rat)
- ešerihija koli (ecoli)

- arabidopsis (arath).

Informacije o ontologijama preuzete su sa http://geneontology.org/docs/download-ontology/ u OBO formatu.

Preuzeti podaci nisu bili u pogodnom obliku za ulaz klasifikatora zbog čega je bilo neophodno njihovo parsiranje.

Ontologija Datoteka *go.obo* sadrži funkcije iz sve tri ontologije. Njenim parsiranjem izdvojena je ontologija molekulskih funkcija (MFO). Ona se sastoji iz približno 12000 čvorova, međutim, neki čvorovi su zastareli (*engl. obsolete*) zbog čega su izbačeni iz grafa. Pored toga, postoje čvorovi koji predstavljaju alternativni identifikator nekog drugog čvora te su takvi čvorovi ujedinjeni u jedan. Time je broj čvorova smanjen na 11078 molekulskih funkcija. Broj je dodatno umanjen zbog prirode podataka. Pre svega, približno 5500 funkcija se uopšte ne pojavljuje u trening skupu što znači da za njih nema pozitivnih instanci odnosno proteina koji ih izvršavaju pa su one izbačene. Zatim, oko 5000 funkcija se pojavljuje manje od 100 puta u trening skupu. Pokušaji treninga klasifikatora za takve funkcije su bili neuspešni te su i one izbačene iz skupa. Nakon svih redukcija ostalo je 399 funkcija sa 100 ili više pojavljivanja u trening skupu za koje su trenirani modeli.

Proteini i funkcije Parsiranjem *uniprot_sprot_exp.txt* izdvojeno je više informacija - proteini sa funkcijama koje obavljaju kao i funkcije sa proteinima za koje je utvrđeno da ih obavljaju. Prvi skup podataka je obogaćen podacima iz ontologije s obzirom da su zadati samo krajnji čvorovi, a ne i svi preci, kako bi se dobio ceo podgraf ontologije koji predstavlja funkciju proteina. Drugi skup poslužio je za prebrojavanje pojavljivanja funkcije u trening skupu kao i za kasnije formiranje skupa pozitivnih i negativnih instanci.

Sekvence proteina U datoteci *uniprot_sprot_exp.fasta* nalazi se 66817 proteina. Među njima se nalaze i proteini koji ne obavljaju neku od funkcija iz MFO. Pored toga, postoje proteini čije sekvence nisu validne u smislu aminokiselina koje sadrže. Pod validnim sekvencama podrazumevaju se samo one koje se sastoje isključivo iz 20 standardnih aminokiselina. Nakon eliminacije ovakvih proteina preostaje 34785 onih koji obavljaju bar jednu molekulsku funkciju. Nakon redukcije broja funkcija na 399 smanjio se i skup proteina. Naime, izbačeni su svi proteini za koje je utvrđeno da vrše neku od eliminisanih funkcija. Nakon svih redukcija, veličina trening skupa je 20960.

4.1.1 Predstavljanje proteina

Mnoge metode mašinskog učenja koriste vektore kao ulaz zbog čega je pogodno da se niska aminokiselina prepiše u niz. Jedan pogodan način za to jeste prebrojavanjem pojavljivanja svakog mogućeg trigrama nad azbukom 20 standardnih aminokiselina. Dimenzija jednog niza je samim tim 20^3, a jedan element sadrži broj pojavljivanja odgovarajućeg trigrama u niski aminokiselina. Ono što je neophodno jeste da za svaki

trigram postoji jedinstveno određen redni broj u nizu. U te svrhe, prvo je potrebno odrediti brojeve pojedinačnih aminokiselina, a šema korišćena u ovoj implementaciji prikazana je u tabeli 4.1.

A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Tabela 4.1: Preslikavanje aminokiselina u broj

Naredni segment koda definiše preslikavanje funkcije koja preslikava aminokiseline u brojeve:

```
def amino_to_number(aa):
    amino_acids = ['A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K',
    'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'Y']
    return amino_acids.index(aa)
```

Sada, za svaki trigram može da se odredi njegov jedinstveni broj koji predstavlja poziciju u nizu i to formulom:

$$kmer\_index = aa_1 * 20^2 + aa_2 * 20 + aa_3$$

Pridruživanje broja trigramu implementirano je rekurzivnom funkcijom `kmer_to_number`. Funkcija je napisana tako da može da radi za različite vrednosti broja k, a podrazumevana je vrednost 3.

```
def kmer_to_number(kmer, k=3):
    if k == 1:
        return amino_to_number(kmer)

    return kmer_to_number(kmer[:-1], k - 1) * number_of_aa +
    amino_to_number(kmer[-1])
```

Sa ovakvim preslikavanjem trigrama u brojeve, jednostavnim prolaskom kroz nisku sa korakom od 3 karaktera dobija se odgovarajući niz. Funkcija `form_array` pravi niz dimenzije 20^k, za parametarski zadato k. Povratna vrednost je niz broja pojavljivanja svakog trigrama na odgovarajućim pozicijama koje se određuju prethodnom funckijom.

```
def form_array(protein, k=3):
    n = len(protein)
    array = np.zeros(number_of_aa ** k)

    for i in range(0, n - k):
        kmer = protein[i:i + k]
        position = kmer_to_number(kmer, k)
        array[position] += 1

    return array
```

4.2 Podela podataka na trening, validacioni i test skup

Početni skup od 20960 proteina podeljen je na dve grupe - *trening* i *test* skup. Trening skup je korišćen za obučavanje pojedinačnih binarnih modela i njihovu evaluaciju, a test skup je primenjen prikilom evaluacije svakog od prediktora koji ujedinjuju pojedinačne modele. U tu svrhu je izdvojeno 100 proteina, a preostalih 20860 korišćeno je za obučavanje. Trening skup je potrebno dodatno podeliti na deo za obučavanje i deo za odabir i validaciju modela, koji se još naziva *validacioni* skup.

Svaka funkcija ima različit skup proteina koji je vrše. Zato je za svaku potrebno posebno obeležiti proteine koji obavljaju zadatu funkciju kao pozitivne instance i one koji je ne obavljaju kao negativne. Nakon toga moguće je izvršiti i podelu trening skupa na skup za obučavanje i validacioni skup. 25% skupa predviđeno je za odabir modela, dok je ostalih 75% korišćeno za obučavanje modela sa različitim parametrima.

Podela na pozitivne i negativne instance implementirano je u funkciji `pos_neg_data`. Prvi parametar funkcije je mapa koja preslikava oznake proteina u sekvencu. Drugi parametar je funkcija za koju se pravi podela. Zatim slede parametri koji označavaju putanje do datoteka koje sadrži podatke o funkcijama sa proteinima iz trening skupa koji vrše tu funkciju i listu svih proteina iz trening skupa. Iz prve datoteke čitaju se podaci o funkciji koja je prosleđena čime se dobijaju pozitivne instance. Negativne instance izdvajaju se iz liste svih proteina preskakanjem prethodno izdvojenih pozitivnih.

```
def pos_neg_data(all_sequences, function, f_path, p_path, k=3):
    positive_proteins = read.read_map_file(f_path)[function]
    negative_proteins = read.read_proteins(p_path, positive_proteins)
    size = len(all_sequences)

    x = np.zeros((size, 20 ** k))
    y = np.ones(size, dtype=int)
    i = 0

    for protein in all_sequences:
        if protein in positive_proteins:
            x[i] = make_array(all_sequences[protein], k)
        if protein in negative_proteins:
            x[i] = make_array(all_sequences[protein], k)
            y[i] = -1

        i += 1

    return x, y
```

Funkcija `read_map_file(file)` predviđena je za čitanje podataka u obliku ključ→vrednosti, pri čemu se ključ koristi kao ključ mape, a vrednosti se spajaju u listu i pridružuju ključu u mapi. Funkcija `read_proteins(file, skip_proteins=None)` služi za čitanje datoteke u koju su zapisane oznake proteina, svaka u zasebnom redu.

Parametar `skip_proteins` određuje koje proteine treba preskočiti u slučaju određivanja negativnih instanci, a kada je vrednost `None`, što je podrazumevana vrednost, čitaju se svi proteini i smeštaju u listu koja se vraća iz funkcije.

Funkcija `make_array(sequences, k=3)` prima nisku koja je ranije formirana pomoću `form_array` za određeni protein. U nisku su upisani parovi indeks:broj, gde je indeks redni broj u nizu kojim se protein predstavlja, a broj je broj pojavljivanja određenog k-grama u proteinskoj sekvenci. Funkcija vraća niz dimenzije 20^k kojim je predstavljen protein.

Podela podataka izvršena je pomoću funkcije `train_test_split` iz *sklearn* biblioteke. Ona deli podatke na deo za obučavanje i deo za evaluaciju u zadatoj razmeri. Postavljena je vrednost za parametar `random_state` kako bi svi modeli bili trenirani na istom podskupu.

4.3 Treniranje modela

Program je pisan u programskom jeziku Python i korišćene su implementacije metoda binarne klasifikacije iz Python-ove biblioteke *sklearn*. Trenirano je 398 modela za svaki metod pojedinačno. Za koren ontologije (funckija GO:0003674) nije bilo moguće napraviti model zbog trening skupa u kom se nalaze samo proteini koji vrše molekulske funkcije. Drugim rečima, za koren nije bilo negativnih instanci u skupu.

Početni skup proteina podeljen je na trening i test skup u razmeri 3:1. Tokom treniranja izvršen je i odabir modela na validacionom skupu koji je izdvojen iz trening skupa u istoj razmeri. Odabir najboljeg modela izvršen je na osnovu f_1-mere.

Nakon što je obučavanje jednog modela završeno, on je sačuvan u posebnoj datoteci sa nazivom koji odgovara identifikatoru funkcije za koju je model treniran i to korišćenjem još jedne Python-ove biblioteke - *pickle*. Ova biblioteka omogućava čuvanje i kasnije čitanje modela mašinskog učenja u pogodnom obliku, tako da nema potrebe za obučavanjem ispočetka već su modeli odmah spremni za predviđanje.

Za odabir najboljeg modela korišćena je funkcija `best_classifier(x_train, y_train, x_test, y_test)` kojoj je potrebno proslediti pripremljene podatke za obučavanje i testiranje. Najbolji model se bira na osnovu f_1-mere koja se računa za svaki od modela, a funkcija vraća model i vrednosti za mere kvaliteta najboljeg modela i to: f_1-meru, tačnost, preciznost, odziv i površinu ispod krive.

Metod potpornih vektora Prilikom odabira modela birana je vrednost za parametar C i to iz skupa $\{0.01, 0.1, 1, 10\}$. Pored toga, odabiran je bolji od dva kernela, linearan i gausov. Svakom modelu je postavljen i parametar `class_weight` na vrednost *balanced* kako bi se svakoj klasi pridružila težina obrnuto proporcionalna frekvenciji pojavljivanja u trening skupu. Zbog dugačkog treniranja jednog modela i velikog broja modela koje je trebalo obučiti, svim nizovima redukovana je dimenzionalnost na 1000. U te svrhe korišćena je Python-ova implementacija algoritma analize glavnih komponenti iz biblioteke *sklearn*.

Logistička regresija Prilikom odabira modela birana je vrednost za parametar C i to iz skupa $\{0.0001, 0.001, 0.01, 0.1, 1\}$. Svakom modelu je postavljen i parametar `class_weight` na vrednost *balanced*. Neki modeli nisu uspevali da nauče ništa iz podataka i njihova f_1-mera bila je jednaka 0. Za takve modele izvršen je dodatan trening na proširenom skupu podataka. Proširenje skupa se odnosi na generisanje sintetičkih instanci kako bi se ublažila nebalansiranost pozitivnih i negativnih instanci. Za proširivanje skupa korišćena je Python-ova biblioteka *imblearn*, a skup je obogaćen tako da odnos pozitivnih i negativnih instanci bude 1:2.

Slučajne šume Kod modela slučajnih šuma trenirani su modeli sa različitim brojem stabala iz skupa $\{100, 400, 700, 1000\}$. Svakom modelu je postavljen i parametar `class_weight` na vrednost *balanced*. Slično kao kod logističke regresije, za modele čija je f_1-mera bila 0 izvršen je dodatan trening sa dodatnim pozitivnim instancama.

4.4 Objedinjavanje modela

Nakon što su svi modeli za odabrani metod obučeni prelazi se na testiranje. Izdvojen je skup od 100 proteina nad kojim je testiran prediktor. Prediktor je formiran na osnovu 398 prethodno obučenih modela za svaku od 398 funkcija koje se pojavljuju u trening skupu. Prilikom predviđanja funkcije jednog proteina, protein se prosleđuje kao ulaz svakom od binarnih klasifikatora koji daju vrednosti 0 ili 1. Ujedinjavanjem svih odgovora dobija se konačan odgovor. Sve funkcije za koje je odgovarajući klasifikator dao 1 kao odgovor predstavljaju čvor podgrafa.

Funkcija `all_predictions(protein, true_functions, pca)` kao parametre prima jedan test protein, zatim funkcije za koje je eksperimentalno utvrđeno da ih izvršava i indikator da li treba koristiti smanjenje dimenzionalnosti (u slučaju metode potpornih vektora prosleđuje se vrednost `True`). Za svaku od 399 funkcija proverava se da li je protein izvršava ili ne primenom odgovarajućeg klasifikatora i rezultat se upisuje u niz - 0 ako ne izvršava odnosno 1 ako izvršava. Izuzetno u slučaju korena ontologije se dodeljuje vrednost 1. Nakon što je svaki klasifikator dao svoj odgovor, određuju se vrednosti za mere kvaliteta koje su postignute za dati protein, što je opisano u narednoj sekciji. Funkcija vraća podgraf ontologije koji predstavlja funkciju proteina i vrednost za sve izračunate mere kvaliteta modela.

4.5 Evaluacija modela

Kao mera kvaliteta pojedinačnih modela korišćena je f_1 mera. U okviru biblioteke *sklearn* implementirana je funkcija koja određuje ovu vrednost na osnovu pravih i predviđenih klasa instanci iz test skupa.

Ista mera korišćena je za evaluaciju konačnog prediktora koji ujedinjuje sve odgovore. S obzirom da prediktor daje strukturu kao odgovor (usmereni aciklički graf) treba preciznije definisati kako se ova mera određuje. Pretpostavimo da je datoj test

instanci pridružen izlazni vektor $y = [0, 1, 1, 0, 1, 1]$, a da je prediktor dao odgovor $y' = [0, 0, 1, 1, 0, 1]$ za istu test instancu. Poređenjem dva vektora može se lako utvrditi koje su klase ispravno određene, a koje pogrešno odnosno mogu se odrediti veličine tp, tn, fp i fn opisane u sekciji 3.4:

$$y' = [\underset{\in tn}{0}, \underset{\in fn}{0}, \underset{\in tp}{1}, \underset{\in fp}{1}, \underset{\in fn}{0}, \underset{\in tp}{1}]$$

Na osnovu ovih veličina dalje se mogu odrediti sve metrike opisane u poglavlju 3.4.

Nakon što su izvršena sva predviđanja, formirana su dva niza - y_true i y_predicted. Svaki od njih sastoji se iz nula i jedinica na odgovarajućim pozicijama, gde nula označava da protein ne izvršava funkciju, a 1 znači da je izvršava. Ovako formirani nizovi mogu se koristiti kao parametri funkcija za određivanje vrednosti za mere kvaliteta iz biblioteke *sklearn*.

```
def all_predictions(protein, true_functions, pca):
    predicted_functions = []
    functions = read_files.read_functions("molecular_functions.txt")

    n = len(functions)
    y_true = np.zeros(n)
    y_predicted = np.zeros(n)
    i = 0

    sequence = train_test_data.make_array(protein, 3)

    for function in functions:

        if function == "GO:0003674":
            predicted_functions.append(function)
            y_true[i] = 1
            y_predicted[i] = 1
            continue

        if pca:
            pca_model = read_model(function, "PCA_models/")
            sequence = pca_model.transform([sequence])[0]

        predicted = prediction([sequence], function)

        if function in true_functions:
            y_true[i] = 1

        if predicted == 1:
            y_predicted[i] = 1
            predicted_functions.append(function)

        i += 1

    f1 = metrics.f1_score(y_true, y_predicted)
```

```
36      acc = metrics.accuracy_score(y_true, y_predicted)
37      pre = metrics.precision_score(y_true, y_predicted)
38      rec = metrics.recall_score(y_true, y_predicted)
39      auc = metrics.roc_auc_score(y_true, y_predicted)
40
41      return predicted_functions, f1, acc, pre, rec, auc
```

Glava 5

Rezultati

Za svaki od tri prethodno opisana metoda binarne klasifikacije trenirano je po 398 modela na celom trening skupu koji su kasnije ujedinjeni u 3 prediktora za predviđanje funkcije proteina. Implementiran je još jedan jednostovaniji klasifikator koji je poslužio kao osnovna metoda za poređenje rezultata. U pitanju je naivni klasifikator koji svakom čvoru dodeljuje vrednost koja odgovara njegovoj frekvenciji pojavljivanja u trening skupu i tako formirani graf pridružuje svakom test primeru [9].

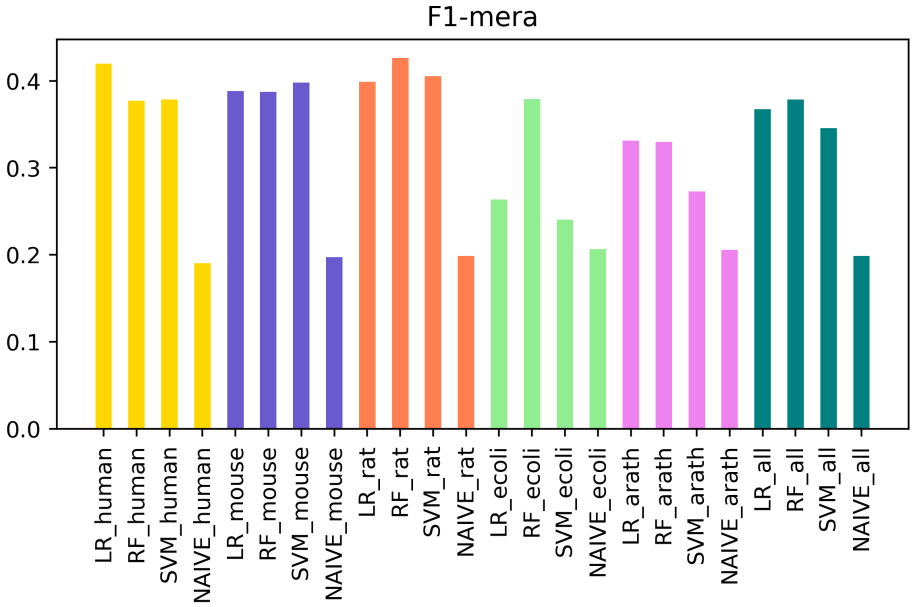
Naivni klasifikator je testiran na istom test skupu kao i 3 prediktora pri čemu su određene prosečne vrednosti za 5 mera kvaliteta modela i to f_1-mera, tačnost, preciznost, odziv i površina ispod ROC krive. Prilikom svakog testiranja računata je prosečna vrednost svake mere kvaliteta i to na nivou pojedinačnih organizama, kao i na nivou celog skupa.

Sa slike 5.1a se može videti da je svaki od tri prediktora bolji od naivnog klasifikatora prema $f1$-meri. Pored toga, sva 3 prediktora na celom skupu imaju približnu $f1$-meru. Nešto slabiji rezultat metode potpornih vektora mogu se objasniti primenom analize glavnih komponenti na trening skup čime je izgubljen deo informacija.

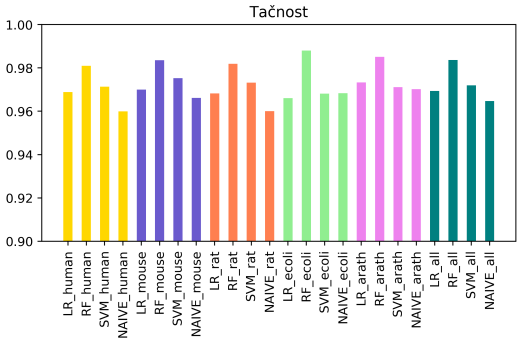
Poređenje tačnosti, prikazano na slici 5.1b, pokazuje da svi modeli imaju sličnu tačnost, što može biti posledica nebalansiranog skupa podataka jer je visoka tačnost postignuta zbog velikog broja ispravno klasifikovanih negativnih instanci, a pošto je skup nebalansiran u korist negativne klase, većina instanci je dobro klasifikovana. Prema preciznosti (slika 5.1c) se ističu modeli metode slučajne šume, dok su prema odzivu (slika 5.1d) bliski klasifikatori metode potpornih vektora i logističke regresije. Sa svih grafika na slici 5.1 vidi se da su rezultati konzistentni za sve organizme. Na primer, predviđanje funkcija proteina čoveka ne daje bolje rezultate nego predviđanje funkcija proteina za ostale organizme.

U tabelama 5.1, 5.2 i 5.3 za 20 najboljih[1] pojedinačnih klasifikatora za svaku upotrebljenu metodu binarne klasifikacije prikazane su vrednosti za f_1-meru, tačnost, preciznost, odziv, površinu ispod krive, kao i nivo na kom se čvor nalazi u grafu, broj i procenat pojavljivanja funkcije u trening skupu. Pored toga, prikazan je i broj pojavljivanja u trening skupu za svaku funkciju, kao i udeo broja pojavljivanja funkcije u celom trening skupu. Vrednosti za sve klasifikatore prikazane su na adresi `http://poincare.matf.bg.ac.rs/~anja_bukurov/master/`.

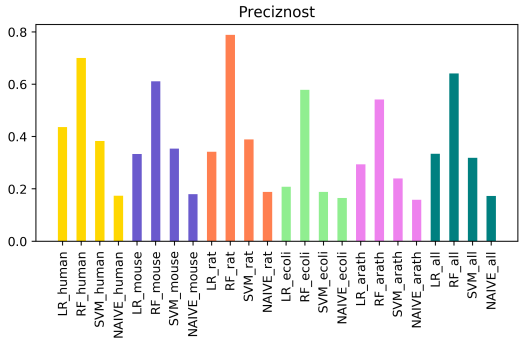
[1] Najboljih prema vrednosti za $f1$-meru.



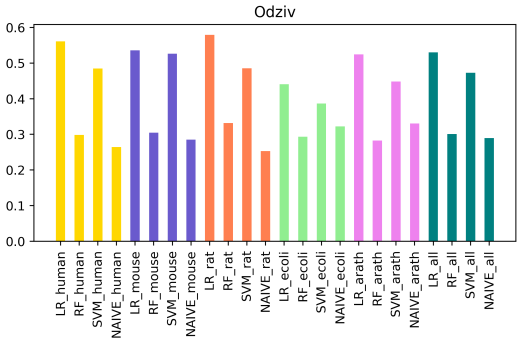
(A) Poređenje prediktora i naivnog klasifikatora prema f1-meri.



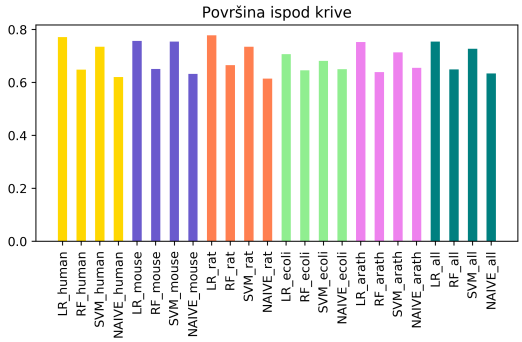
(B) Poređenje prema tačnosti.



(C) Poređenje prema preciznosti.



(D) Poređenje prema odzivu.



(E) Poređenje prema površini ispod ROC krive.

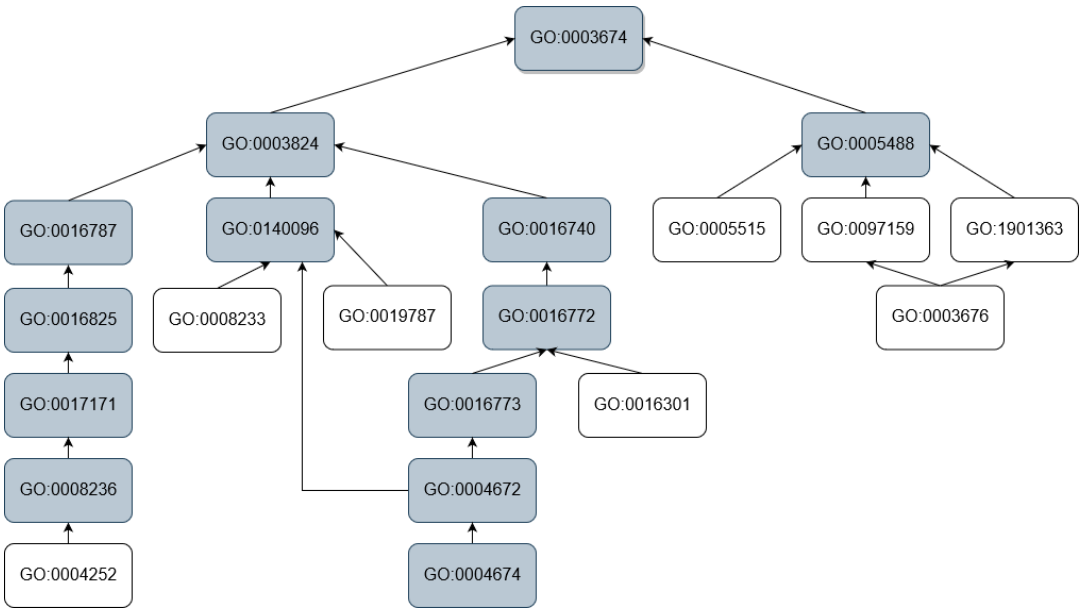
Slika 5.1: Poređenje mera kvaliteta prediktora i naivnog klasifikatora. Rezultati jednog organizma prikazani su istom bojom.

Klasifikatori se odlikuju visokom preciznošću, ali niskim odzivom što znači da većinu instanci klasifikuju kao negativne, ali one koje su klasifikovane kao pozitivne su uglavnom ispravno klasifikovane. Tačnost ovih modela je visoka (oko 0.9 i više)

Funkcija	F1	Acc	Pre	Rec	AUC	Nivo	Br. p.	Pr. p.
GO:0005488	0.84	0.731	0.74	0.97	0.54	1	15092	72.3%
GO:0017171	0.57	0.991	0.91	0.42	0.71	4	354	1.7%
GO:0016825	0.56	0.991	0.91	0.4	0.7	3	354	1.7%
GO:0008236	0.54	0.991	0.93	0.38	0.69	5	349	1.7%
GO:0004672	0.53	0.961	0.92	0.37	0.69	3	1257	6.0%
GO:0003824	0.52	0.738	0.84	0.37	0.67	1	7843	37.6%
GO:0016773	0.51	0.955	0.94	0.35	0.67	4	1377	6.6%
GO:0016301	0.47	0.95	0.96	0.31	0.65	4	1452	7.0%
GO:0016772	0.46	0.943	0.96	0.31	0.65	3	1609	7.7%
GO:0004674	0.44	0.974	0.8	0.3	0.65	4	776	3.7%
GO:0140096	0.42	0.88	0.89	0.27	0.63	2	3337	16.0%
GO:0016740	0.36	0.881	0.92	0.23	0.61	2	3069	14.7%
GO:0030594	0.32	0.998	0.75	0.2	0.6	3	86	0.4%
GO:1901363	0.3	0.735	0.73	0.19	0.58	2	6236	29.9%
GO:0016787	0.23	0.864	0.77	0.13	0.56	2	3202	15.3%
GO:0019199	0.17	0.996	0.67	0.1	0.55	4	78	0.4%
GO:0060089	0.16	0.957	0.72	0.09	0.54	1	976	4.7%
GO:0038023	0.11	0.959	0.72	0.06	0.53	2	909	4.4%
GO:0004888	0.11	0.971	0.75	0.06	0.53	3	626	3.0%
GO:0004930	0.09	0.984	0.8	0.05	0.52	4	312	1.5%

TABELA 5.1: Prikaz mera kvaliteta za pojedinačne klasifikatore metode slučajne šume za 20 čvorova sa najboljom $f1$-merom

za funckije sa manjim brojem pozitivnih instanci (ispod 10%), dok je za funkcije sa većim brojem instanci nešto tačnost nešto manja (ispod 0.8). Slika 5.2 ilustruje podgraf ontologije koji sadrži funkcije ove tabele.

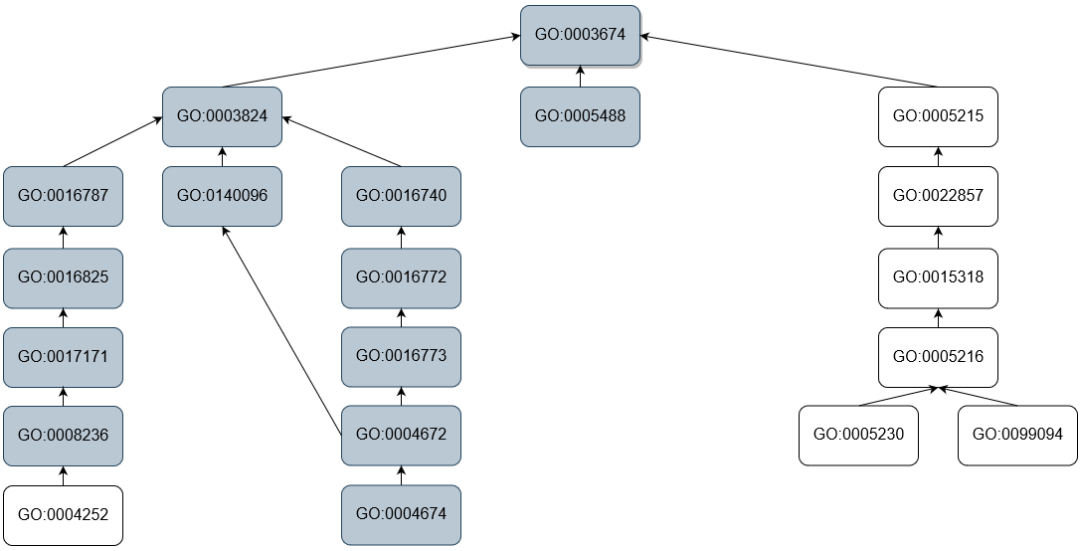


SLIKA 5.2: Prikaz podgrafa ontologije koji sadrži funkcije iz tabele 5.1

Funkcija	F1	Acc	Pre	Rec	AUC	Nivo	Br. p.	Pr. p.
GO:0004672	0.82	0.979	0.82	0.81	0.9	3	1257	6.0%
GO:0004930	0.79	0.993	0.84	0.74	0.87	4	312	1.5%
GO:0016773	0.77	0.97	0.78	0.75	0.87	4	1377	6.6%
GO:0005488	0.76	0.673	0.8	0.73	0.63	1	15092	72.3%
GO:0004674	0.74	0.98	0.65	0.87	0.93	4	776	3.7%
GO:0016301	0.73	0.961	0.72	0.74	0.86	4	1452	7.0%
GO:0003824	0.71	0.756	0.65	0.77	0.76	1	7843	37.6%
GO:0008236	0.7	0.992	0.71	0.69	0.84	5	349	1.7%
GO:0016825	0.69	0.991	0.68	0.71	0.85	3	354	1.7%
GO:0017171	0.69	0.991	0.69	0.68	0.84	4	354	1.7%
GO:0016772	0.67	0.948	0.69	0.64	0.81	3	1609	7.7%
GO:0030594	0.63	0.997	0.52	0.8	0.9	3	86	0.4%
GO:0004888	0.61	0.972	0.52	0.75	0.87	3	626	3.0%
GO:0019199	0.61	0.996	0.52	0.75	0.87	4	78	0.4%
GO:0140096	0.59	0.858	0.54	0.65	0.78	2	3337	16.0%
GO:0038023	0.58	0.964	0.58	0.59	0.78	2	909	4.4%
GO:0060089	0.58	0.96	0.55	0.61	0.79	1	976	4.7%
GO:1901363	0.56	0.703	0.5	0.64	0.69	2	6236	29.9%
GO:0016740	0.53	0.844	0.49	0.58	0.74	2	3069	14.7%
GO:0016787	0.46	0.808	0.4	0.55	0.7	2	3202	15.3%

Tabela 5.2: Prikaz mera kvaliteta za pojedinačne klasifikatore metode logistička regresija za 20 čvorova sa najboljom $f1$-merom

Najbolji modeli logističke regresije odlikuju se približnim merama za preciznost i odziv. Tačnost modela je veća kod modela sa manjim brojem pozitivnih instanci kao i kod metode slučajnih šuma.

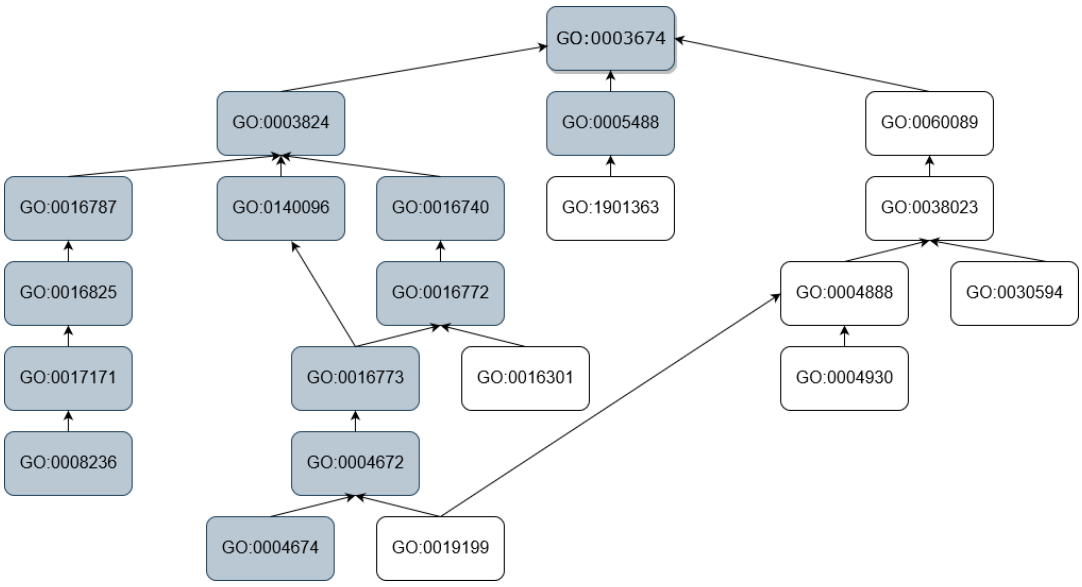


Slika 5.3: Prikaz podgrafa ontologije koji sadrži funkcije iz tabele 5.2

Funkcija	F1	Acc	Pre	Rec	AUC	Nivo	Br. p.	Pr. p.
GO:0004672	0.81	0.978	0.84	0.78	0.88	3	1257	6.0%
GO:0030594	0.79	0.999	0.85	0.73	0.87	3	86	0.4%
GO:0004930	0.78	0.992	0.73	0.84	0.92	4	312	1.5%
GO:0005488	0.78	0.696	0.81	0.75	0.65	1	15092	72.3%
GO:0016773	0.76	0.97	0.81	0.72	0.85	4	1377	6.6%
GO:0004674	0.75	0.981	0.67	0.86	0.92	4	776	3.7%
GO:0016301	0.75	0.967	0.82	0.68	0.84	4	1452	7.0%
GO:0003824	0.72	0.786	0.71	0.74	0.78	1	7843	37.6%
GO:0019199	0.72	0.998	0.74	0.7	0.85	4	78	0.4%
GO:0017171	0.71	0.991	0.65	0.78	0.89	4	354	1.7%
GO:0016825	0.71	0.992	0.75	0.67	0.83	3	354	1.7%
GO:0008236	0.7	0.991	0.64	0.76	0.88	5	349	1.7%
GO:0016772	0.67	0.951	0.73	0.62	0.8	3	1609	7.7%
GO:0004888	0.64	0.98	0.68	0.6	0.8	3	626	3.0%
GO:0038023	0.63	0.971	0.69	0.57	0.78	2	909	4.4%
GO:0060089	0.62	0.967	0.66	0.59	0.79	1	976	4.7%
GO:0140096	0.59	0.883	0.66	0.54	0.75	2	3337	16.0%
GO:1901363	0.57	0.728	0.54	0.61	0.69	2	6236	29.9%
GO:0016740	0.55	0.873	0.59	0.51	0.72	2	3069	14.7%
GO:0016787	0.48	0.842	0.48	0.49	0.7	2	3202	15.3%

TABELA 5.3: Prikaz mera kvaliteta za pojedinačne klasifikatore metode potpornih vektora za 20 čvorova sa najboljom $f1$-merom

Najbolji modeli metode potpornih vektora pokazuju slično ponašanje kao modeli logističke regresije.



SLIKA 5.4: Prikaz podgrafa ontologije koji sadrži funkcije iz tabele 5.3

Prikazani podgrafi sadrže zajedničke čvorove odnosno, sva tri prediktora su za nekoliko istih funkcija dali najbolje rezultate. Nazivi ovih funckija navedeni su u tabeli 5.4. Poređenje mera kvaliteta klasifikatora ovih funkcija prikazano je u tabeli 5.5, a na slikama su zajednički čvorovi označeni sivom bojom.

Oznaka funkcije	Naziv funkcije
GO:0003674	molecular_function
GO:0003824	catalytic activity
GO:0005488	binding
GO:0016787	hydrolase activity
GO:0140096	catalytic activity, acting on a protein
GO:0016740	transferase activity
GO:0016825	hydrolase activity, acting on acid phosphorus-nitrogen bonds
GO:0016772	transferase activity, transferring phosphorus-containing groups
GO:0017171	serine hydrolase activity
GO:0016773	phosphotransferase activity, alcohol group as acceptor
GO:0008236	serine-type peptidase activity
GO:0004672	protein kinase activity
GO:0004674	protein serine/threonine kinase activity

Tabela 5.4: Oznake i nazivi funkcija za zajedničke čvorove podgrafa sa slika 5.2, 5.3 i 5.4

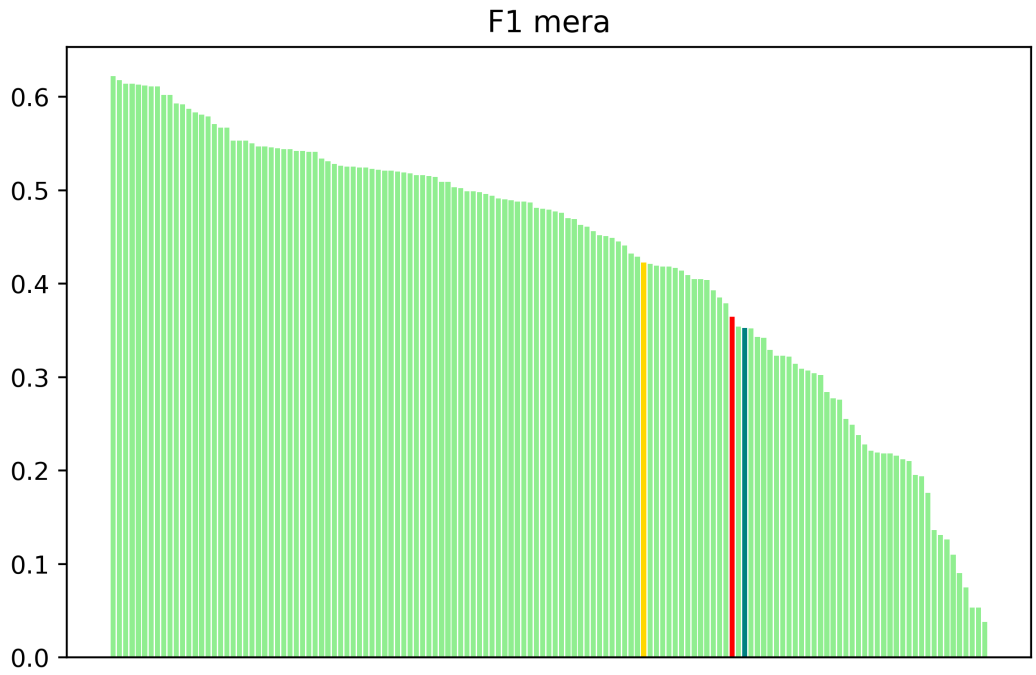
Modeli metode slučajnih šuma za iste funkcije daju nešto slabiju $f1$-meru u poređenju sa ostalim metodama, ali zato je preciznost skoro svih modela veća u odnosu na modele drugih metoda. Tačnost svih modela su prilično bliske, dok se prema odzivu najviše ističu modeli linearne regresije. Površina ispod ROC krive je slična za modele logističke regresije i modela potpornih vektora, a slučajne šume su dale nešto slabije rezultate.

Oznaka funkcije	F1-mera			Tačnost			Preciznost			Odziv			Površina ispod ROC krive		
	LR	RF	SVM	LR	RF	SVM	LR	RF	SVM	LR	RF	SVM	LR	RF	SVM
GO:0003824	0.71	0.52	**0.72**	0.76	0.74	**0.79**	0.65	**0.84**	0.71	**0.77**	0.37	0.74	0.76	0.67	**0.78**
GO:0005488	0.76	**0.84**	0.78	0.67	**0.73**	0.7	0.8	0.74	**0.81**	0.73	**0.97**	0.75	0.63	0.54	**0.65**
GO:0016787	0.46	0.23	**0.48**	0.81	**0.86**	0.84	0.4	**0.77**	0.48	**0.55**	0.13	0.49	**0.7**	0.56	**0.7**
GO:0140096	**0.59**	0.42	**0.59**	0.86	**0.88**	**0.88**	0.54	**0.89**	0.66	**0.65**	0.27	0.54	**0.78**	0.63	0.75
GO:0016740	0.53	0.36	**0.55**	0.84	**0.88**	0.87	0.49	**0.92**	0.59	**0.58**	0.23	0.51	**0.74**	0.61	0.72
GO:0016825	0.69	0.56	**0.71**	**0.99**	**0.99**	**0.99**	0.68	**0.91**	0.75	**0.71**	0.4	0.67	**0.85**	0.7	0.83
GO:0016772	**0.67**	0.46	**0.67**	**0.95**	0.94	**0.95**	0.69	**0.96**	0.73	**0.64**	0.31	0.62	**0.81**	0.65	0.8
GO:0017171	0.69	0.57	**0.71**	**0.99**	**0.99**	**0.99**	0.69	**0.91**	0.65	0.68	0.42	**0.78**	0.84	0.71	**0.89**
GO:0016773	**0.77**	0.51	0.76	**0.97**	0.96	**0.97**	0.78	**0.94**	0.81	**0.75**	0.35	0.72	**0.87**	0.67	0.85
GO:0008236	**0.7**	0.54	**0.7**	**0.99**	**0.99**	**0.99**	0.71	**0.93**	0.64	0.69	0.38	**0.76**	0.84	0.69	**0.88**
GO:0004672	**0.82**	0.53	0.81	**0.98**	0.96	**0.98**	0.82	**0.92**	0.84	**0.81**	0.37	0.78	**0.9**	0.69	0.88
GO:0004674	0.74	0.44	**0.75**	**0.98**	0.97	**0.98**	0.65	**0.8**	0.67	**0.87**	0.3	0.86	**0.93**	0.65	0.92

Tabela 5.5: Poređenje mera kvaliteta za zajedničke čvorove podgrafa sa slika 5.2, 5.3 i 5.4

Predikotri su dodatno testirani na *benchmark* skupu proteina koji je korišćen u okviru CAFA3 takmičenja [2]. Skup sadrži 453 proteina različitih organizama. Izračunata je prosečna vrednost f_1-mera za svaki prediktor, a dobijeni rezultati su dodatno upoređeni sa rezultatima učesnika takmičenja [33].

Prediktor za metodu slučajnih šuma dao je najbolji rezultat od tri prediktora, čime je među učesnicima ovog takmičenja zauzeo 85. mesto među 136 tačkmičara. Poređenje sva tri prediktora sa svim učesnicima prikazano je na slici 5.5.



Slika 5.5: Poređenje f_1-mera tri prediktora sa rezultatima učesnika CAFA3 takmičenja. Crvenom bojom označen je prediktor linearne regresije, žutom prediktor slučajnih šuma, plavom prediktor metode potpornih vektora. Svetlo zelenom bojom prikazani su rezultati postignuti na CAFA3 takmičenju.

Glava 6

Zaključak

U ovom radu prikazan je razvoj tri prediktora za predviđanje funkcije proteina. Računarski metodi za određivanje funkcija razvijaju se godinama, ali i dalje nema metoda koji može da odredi funkciju proteina preciznije od eksperimentalnog metoda. Kao što je već pomenuto, eksperimentalno utvrđivanje funkcije proteina je skup i spor proces zbog čega je ovaj problem i dalje aktuelan i od velikog značaja.

Iako su se obučeni prediktori pokazali bolje od naivnog klasifikatora, nemaju približnu moć predviđanja u poređenju sa aktivnim rezultatima prikazanim na poslednjem CAFA takmičenju, najrelevantnijem takmičenju u ovoj oblasti. Planovi za unapređivanje prediktora obuhvataju:

- treniranje pojedinačnih modela i prediktora za svaki od organizama - cilj je utvrditi da li će se prediktori bolje ponašati za određeni organizam ukoliko se obučavaju na proteinima koji potiču isključivo iz tog organizma,
- povećanje trening skupa - zbog malog broja pozitivnih instanci, za određene funkcije nije pravljen klasifikator,
- promenu ulaznih podataka - ideja je povećati vrednost parametra k koji određuje dužinu podniski, a samim tim i niza kojim se predstavlja protein,
- korišćenje raznovrsnijih metoda binarne klasifikacije - na primer, neuronskih mreža.

Literatura

[1] Carlos E. Pedreira Juliana S. Bernardes. *A Review of Protein Function Prediction Under Machine Learning Perspective.*

[2] *The CAFA challenge.* on-line na: https://www.biofunctionprediction.org/cafa/.

[3] Jiang Y. et al. (2016). *An expanded evaluation of protein function prediction methods shows an improvement in accuracy.*

[4] Cozzetto D. et al. (2013). *Protein function prediction by massive integration of evolutionary analyses and multiple data sources.*

[5] Gong Q. et al. (2016). *GoFDR: a sequence alignment based method for predicting protein functions.*

[6] Lan L. et al. (2013). *MS-kNN: protein function prediction by integrating multiple data sources.*

[7] Cheng J Cao R. *Integrated protein function prediction by mining function associations, sequences, and protein-protein and gene-gene interaction networks.*

[8] Predrag Radivojac. *A (not so) Quick Introduction to Protein Function Prediction.* 2013.

[9] Jovana Kovačević. *Strukturna predikcija funkcije proteina i odnos funckionalnih kategorija i neuređenosti.* 2015.

[10] Rick Ricer Michael A. Lieberman. *Biochemistry, Molecular Biology, and Genetics.* New Science Press Ltd, 2004.

[11] OpenStax. *Anatomy and Phisiology.* 2013.

[12] Denise R. Ferrier Richard A. Harvey. *Biochemistry Fifth Edition.* Lippincott Williams & Wilkins, a Wolters Kluwer business, 2011. ISBN: 978-1-60831-412-6.

[13] Vesna Spasojević-Kalimanovska Slavica Spasić Zorana Jelić-Ivanović. *Opšta biohemija.* 2002.

[14] Dubravka Cvorišćec Ivana Čepelak. *Štrausova medicinska biokemija.* Medicinska naklada, 2009.

[15] Marek Kimmel Andrzej Polanski. *Bioinformatics.* Springer-Verlag, 2007. ISBN: 978-3-540-24166-9.

[16] Dagmar Ringe Georgy A Pesko. *Protein Structure and Function.* New Science Press Ltd, 2004.

[17] Regina Bailey. *The Function and Structure of Proteins*. on-line na: https://www.thoughtco.com/protein-function-373550. 2019.

[18] *Role of proteins in the body*. on-line na: https://www.sciencelearn.org.nz/resources/209-role-of-proteins-in-the-body. 2011.

[19] *What are proteins and what do they do?* on-line na: https://ghr.nlm.nih.gov/primer/howgeneswork/protein. 2019.

[20] *Gene Ontology*. http://geneontology.org/.

[21] *QuickGO*. https://www.ebi.ac.uk/QuickGO/.

[22] Baptiste Rocca. *Handling imbalanced datasets in machine learning*. on-line na: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28. 2019.

[23] Hoang Minh. *How to Handle Imbalanced Data in Classification Problems*. on-line na: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28. 2018.

[24] Kumar Vipin Tan Pang-Ning Steinbach Michael. *Introduction to Data Mining*. Pearson Education, 2006.

[25] Jelena Graovac. *Prilog metodama klasifikacije teksta: matematički modeli i primene*. 2014.

[26] Anđelka Zečević Mladen Nikolić. *Mašinsko učenje*. on-line na: http://ml.matf.bg.ac.rs/readings/ml.pdf.

[27] Mladen Nikolić Predrag Janičić. *Veštačka inteligencija*. on-line na: http://poincare.matf.bg.ac.rs/~janicic/courses/vi.pdf.

[28] Ishaan Dey. *Evaluating Classification Models*. on-line na: https://towardsdatascience.com/hackcvilleds-4636c6c1ba53. 2019.

[29] *Classification: ROC Curve and AUC*. on-line na: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[30] Sarang Narkhede. *Understanding AUC-ROC curve*. on-line na: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[31] Anđelka Zečević Mladen Nikolić. *Naučno izračunavanje*. on-line na: http://ni.matf.bg.ac.rs/materijali/ni.pdf.

[32] DeZyre. *Principal Component Analysis Tutorial*. on-line na: https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial. 2019.

[33] Zhou et al. *The CAFA challenge reports improved protein function predictionand new functional annotations for hundreds of genes throughexperimental screens*. on-line na: https://www.biorxiv.org/content/biorxiv/early/2019/05/29/653105.full.pdf.