SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2742

ODABIR ZNAČAJKI SKUPA PODATAKA UZ POMOĆ EVOLUCIJSKIH ALGORITAMA

Laura Majer

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2742

ODABIR ZNAČAJKI SKUPA PODATAKA UZ POMOĆ EVOLUCIJSKIH ALGORITAMA

Laura Majer

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 11. ožujka 2022.

DIPLOMSKI ZADATAK br. 2742

Pristupnica:	**Laura Majer (0036506475)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Marin Golub

Zadatak: **Odabir značajki skupa podataka uz pomoć evolucijskih algoritama**

Opis zadatka:

Obrazložiti zašto je važno odabrati informativan skup značajki nekog skupa podataka te navesti pregled metoda odabira značajki. Navesti taksonomiju i pregled evolucijskih algoritama. Detaljno opisati evolucijske algoritme pogodne za rješavanje takvog optimizacijskog problema. Programski ostvariti evolucijski algoritam za odabir značajki koristeći javno dostupna radna okruženja i programske knjižnice. Analizirati nekoliko slobodno dostupnih skupova podataka. Odabrati one skupove podataka koji su prikladni za rješavanje problema odabira značajki te nad njima ispitati djelotvornost ostvarenog algoritma za odabir značajki.

Rok za predaju rada: 27. lipnja 2022.

SADRŽAJ

1. Uvod

Obilje informacija u suvremenim skupovima podataka dvosjekli je mač. Iako omogućuje izgradnju dovoljno ekspresivnih i složenih modela, dovodi do redundancije koja može rezultirati u manje djelotvornim te teže objašnjivim modelima.

U svrhu rješavanja tih nedostataka razvijene su brojne metode prilagodbe skupa podataka. Metode za redukciju dimenzionalnosti, primjerice PCA (*Principal Components Analysis*) i MDS (*Multidimensional Scaling*), modificiraju originalni skup značajki. Za razliku od transformacije, selekcijom značajki zadržavaju se njihova svojstva i time interpretabilnost - podskup značajki odabran algoritmom ne modificira se ni na koji način.

Metode selekcije značajki sastoje se od dvije komponente - pretraživanja prostora stanja te evaluacije kvalitete značajki u potencijalnom skupu, korak koji se ponavlja iterativno dok nije postignut kriterij zaustavljanja. Stoga je osim metode evaluacije značajki potrebno odabrati i strategiju pretraživanja prostora stanja. Osim jednostavnih strategija, poput unaprijednog i unatražnog odabira, koriste se i kompliciranije heuristike. S obzirom da se radi o NP-teškom problemu, primjenjive su heuristike isprobane i korištene u rješavanju ekvivalentnih problema.

U području algoritama insipiriranih prirodnim pojavama svojim utjecajem i primjenom ističu se evolucijski algoritmi. Koristeći teoriju biološke evolucije kao temelj algoritma - od nasljeđivanja genoma, preko mutacije i rekombinacije pa sve do selekcije najpogodnijih jedinki - u evolucijskim algoritmima prostor stanja pretražuje se usmjereno uz minimiziranje vjerojatnosti zapinjanja u lokalnom optimumu.

U nastavku rada bit će opisana motivacija, teorija i praksa vezana uz odabir značajki te evolucijski algoritmi kao odabrana metoda pretraživanja prostora stanja. Nakon toga, analizirati će se i usporediti rezultati implementacije nekoliko opisanih metoda.

2. Odabir značajki skupa podataka

Strukturirani skupovi podataka, koji su zbog svojih karakteristika prikladni za rukovanje i analizu, najčešći su tip skupova korištenih u strojnom učenju. Sastoje se od značajki (atributa) i primjera (uzoraka). Značajke su preoblikovane izvorne vrijednosti korištene na ulazu metode strojnog učenja, dok su primjeri skup vrijednosti za svaku od značajki u skupu podataka. Promatranjem skupa podataka kao tablice, značajke se predstavljaju kao stupci, gdje je oznaka klase za klasifikacijske ili numerička vrijednost za regresijske probleme jedan od stupaca. S druge strane, primjeri se predstavljaju kao retci u tablici.

Važnost korištenja značajki može se sumirati u tri točke - sažetost, informativnost i interpretabilnost. Sažetost je postignuta korištenjem značajki umjesto sirovih podataka, što omogućuje veću učinkovitost modela strojnog i dubokog učenja. Nadalje, informativnost je karakteristika koja definira da se prava informacija krije u odnosima između značajki, ne u samim vrijednostima. Konačno, interpretabilnost je veća za značajke nego sirove vrijednosti jer su ljudima lakše razumljive u kontekstu modela i analize njegovih performansi.

No, nisu sve značajke u skupu podataka jednako važne - odabirom ispravnog podskupa značajki može se poboljšati svaka od tri pozitivne karakteristike značajki. Sažetost se smanjivanjem podskupa pojačava direktno, kao i interpretabilnost, pogotovo u slučajevima gdje se utvrdi jasna veza između oznake klase i nekih značajki. Informativnost je također povećana s ispravno odabranim podskupom, zbog činjenice da nevažne značajke doprinose smanjenoj preciznosti ili prenaučenosti modela. Osim toga, vrijeme izgradnje modela smanjuje se za manji skup značajki.

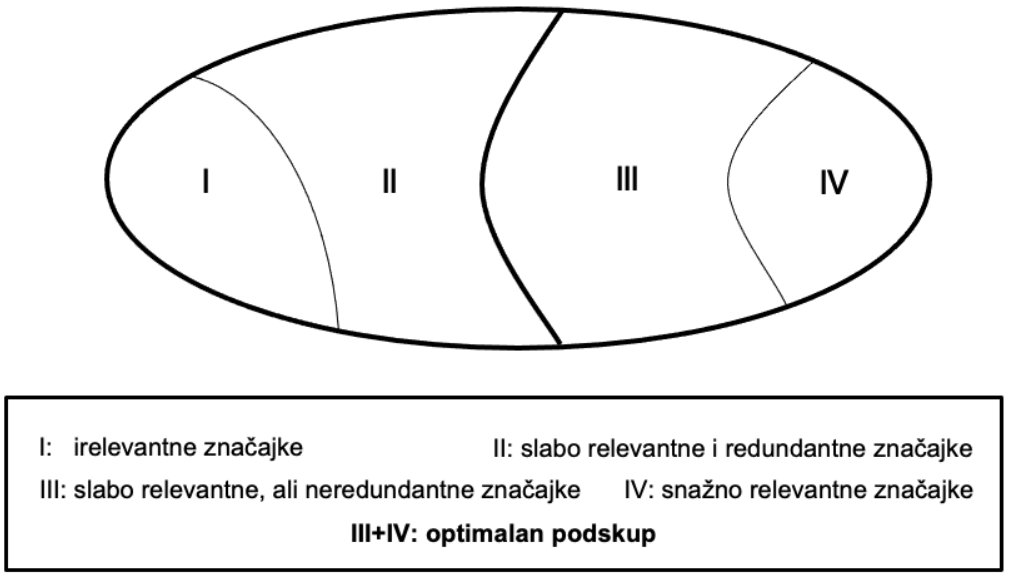
Proces prilagodbe značajki (eng. *feature engineering*) gotovo je neizostavan korak u obradi velikih skupova podataka. Područja u kojem je odabir značajki najviše primjenjiv uključuju analizu teksta i analizu gena [1], zbog spomenutog svojstva malog broja uzoraka uz veliki skup značajki, slučajeve gdje se iz sirovih podataka izluči velik broj značajki, poput analize slika ili vremenskih nizova [12] te područja gdje je nužno ubrzati vrijeme izgradnje i korištenja modela.

2.1. Raščlamba značajki

Prije opisa postupka odabira značajki skupa podataka, važno je definirati na koji se način značajke u teoriji definiraju. S obzirom na to koliko su informativne, značajke se mogu kategorizirati u:

- irelevantne značajke
- slabo relevantne i redundantne značajke
- slabo redundantne, ali relevantne značajke
- snažno relevantne značajke

Relevantnost značajke određuje se individualno, dok se redundancija značajki određuje u odnosu na ostale značajke u skupu podataka. Optimalan podskup tada je definiran unijom slabo redundantnih, ali relevantnih značajki te snažno relevantnih značajki. Podjela je grafički prikazana na 2.1. Cilj odabira značajki maksimizacija je relevantnosti uz istovremenu minimizaciju redundancije.



Slika 2.1: Četiri ključna koraka odabira značajki [13]

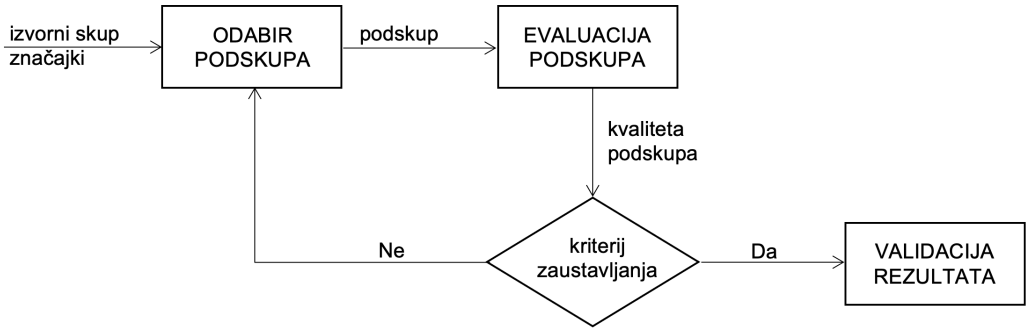
2.2. Definicija problema

Formalna definicija problema odabira značajki odabir je podskupa izvornih značajki, gdje se optimalnost podskupa procjenjuje nekim evaluacijskim kriterijem. Ovisno

o broju značajki m, broj mogućih podskupova je $2^m - 1$, gdje se prazan skup zanemaruje. Porastom broja značajki m, eksponencijalno raste i broj mogućih podskupova, što problem odabira značajki čini nerješivim iscrpljujućom pretragom. U nastavku su prikazani tijek i komponente raznih pristupa odabiru značajki. Bitno je napomenuti da se u obzir uzimaju isključivo metode koje ne modificiraju ulazne podatke ni na koji način, već samo odabiru njihov podskup.

2.3. Algoritam odabira značajki skupa podataka

Iako se metode odabira značajki međusobno razlikuju po svojstvima i načinima odabira, moguće je definirati općeniti tijek koji slijedi većina metoda [1]. Kategorizacija i svojstva različitih vrsta algoritma opisani su u sljedećem poglavlju, a na slici 2.2 prikazan je generalizirani tijek algoritma [13].



Slika 2.2: Četiri ključna koraka odabira značajki [13]

Prvi je korak odabir podskupa, koji je vođen određenom strategijom pretraživanja. Nakon toga, odabrani se podskup evaluira i uspoređuje s dosadašnjim rezultatima. Ta se dva koraka ponavljaju sve dok nije postignut kriterij zaustavljanja. U nastavku su detaljnije opisane navedene komponente u procesu odabira značajki.

2.4. Odabir podskupa - pretraživanje prostora stanja

Odabir podskupa skupa podataka zapravo je problem pretraživanja prostora stanja, u kojem stanje predstavlja potencijalni podskup značajki. Veličina prostora je $2^m - 1$

[1] Iznimka su univarijatne metode, koje evaluiraju cijeli skup značajki (stoga je redundantan korak odabira podskupa) te ugrađene metode koje slijede vlastite principe odabira značajki, često ugrađenih u logiku algoritma.

(zanemaruje se prazan skup). Primjenjive su sve metode pretraživanja korištene u ekvivalentnim problemima te se gotovo svi pristupi mogu razdijeliti na sljedeći način:

1. unaprijedni odabir, gdje se značajke slijedno dodaju na početni prazni skup;
2. unatražni odabir, gdje se značajke slijedno oduzimaju od cijelog skupa značajki;
3. dvosmjerni odabir, gdje se značajke simultano unaprijedno dodaju i unatražno oduzimaju;
4. heuristički odabir, gdje se značajke istražuju usmjerene određenom heurističkom strategijom.

U ovom je radu kao strategija pretraživanja odabran genetski algoritam. Korišten je u radovima [21] [18] [19], gdje se nekad uz odabir značajki paralelno optimiraju parametri klasifikatora, kao što je slučaj u [19].

Ipak, ključna komponenta algoritama za odabir značajki skupa podataka nije pretraživanje već evaluacija podskupa. Kao i u ostalim područjima rada s podacima, ni u odabiru značajki ne postoji univerzalna metoda - jedna koja se jednako uspješno može primijeniti na svim instancama i za različite probleme strojnog učenja. U nastavku su prikazani različiti načini evaluacije podskupova značajki, čija procjena u konačnici dovodi do odabira finalnog podskupa. Svaka se strategija pretraživanja prostora stanja može koristiti za navedene metode evaluacije podskupova te se algoritmi za odabir značajki prvenstveno razvrstavaju po ovim metodama, gdje je strategija pretraživanja samo sekundarno svojstvo.

2.5. Metode filtera

Metode filtera koriste se neovisno o modelu za čije se korištenje odabiru značajke, što znači da se mogu smatrati univerzalnijim od kasnije objašnjenih metoda omotača. Za razliku od preciznosti modela, metode filtera definiraju mjeru koliko su određena značajka ili skup značajki bitni za opis ciljne varijable. Neovisnost od modela ujedno je prednost i mana - iako se time ubrzava izvođenje, oslanjanje isključivo na analitičke metode smanjuje efikasnost metoda filtera.

Jedna od podjela metoda filtera zasniva se na temelju cilja metrike, tj. načina na koji se značajke uspoređuju. Metode se tako mogu razvrstati u one koje se temelje na informaciji, udaljenosti, sličnosti, konzistentnosti te statističkim mjerama.

S druge strane, s obzirom na broj značajki koje evaluiraju, metode filtera dijele se na univarijantne i multivarijatne. Univarijatne metode ocjenjuju značajke pojedinačno, za razliku od multivarijatnih metoda, gdje se evaluira podskup značajki.

2.5.1. Univarijantne metode

Univarijatne metode koriste se za evaluaciju cijelog skupa značajki odjednom. Tako se za svaku značajku određuje vrijednost metrike te se na temelju toga odabire određeni broj ili postotak značajki iz originalnog skupa. Univarijatne metode se, kao što je već objašnjeno, mogu razvrstati na metode temeljene na informaciji (uključujući informacijsku dobit (eng. *Information gain*) i simetričnu nesigurnost (eng. *Symmetrical uncertainty*)), metode temeljene na statističkim procjenama (primjerice korelacija i Fischer vrijednost), metode temeljene na sličnosti (spektralni odabir značajki *SPEC* i Laplace-vrijednost *LS*) te metode temeljene na udaljenosti (*Relief* obitelj metoda). U nastavku su detaljnije opisane tri metode korištene u praktičnom radu.

Simetrična nesigurnost

Simetrična nesigurnost metrika je temeljena na informaciji, a u kontekstu odabira značajki promatra se odnos između prediktivne značajke X i ciljne značajke Y. Radi se o prilagodbi metrike informacijske dobiti,

$$IG(X|Y) = H(X) - H(X|Y) \tag{2.1}$$

gdje $H(X)$ označava entropiju varijable X,

$$H(X) = -\sum_i p(x_i) log_2 p(x_i) \tag{2.2}$$

a $H(X|Y)$ uvjetnu entropiju varijable X uz poznavanje Y

$$H(X|Y) = -\sum_j p(x_j) \sum_i p(x_i|y_j) log_2 p(x_i|y_j) \tag{2.3}$$

S obzirom da informacijska dobit favorizira značajke s većim brojem vrijednosti, simetrična nesigurnost eliminira tu prednost i ograničava vrijednost metrike na interval [0,1], gdje vrijednost 1 označava potpunu prediktibilnost vrijednosti jedne varijable na temelju druge, a vrijednost 0 neovisnost jedne varijable o drugoj.

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X) + H(Y)}\right] \tag{2.4}$$

Relief metoda

Kao jedna od metoda u *Relief* skupini metrika, Relief metoda temeljena je na udaljenosti, točnije na konceptu najbližih susjeda (eng. *nearest neighbours*). Za razliku od

većine drugih filterskih metoda za pojedinačne značajke, detektira ovisnost među prediktivnim značajkama. Ima veću složenost od većine metrika ($O(n^2m)$), no uglavnom daje kvalitetnije rezultate. [20]

Fischer vrijednost

Cilj metrike Fischer vrijednosti pronalazak je podskupa značajki takvog da je u prostoru odabranih značajki udaljenost između uzoraka različitih klasa maksimalna, dok je udaljenost između uzoraka unutar klase minimalna. Izračunava se na temelju matrica raspršenosti. Ne daje informaciju o međusobnom odnosu značajki. [5]

2.5.2. Multivarijatne metode

Multivarijatne metode nastoje definirati koji je međusoban odnos značajki u podskupu s ciljem eliminiranja redundantnih značajki. Odabir podskupa za evaluaciju provodi se metodama opisanim u 2.4. Primjeri multivarijatnih metoda uključuju mRMR metodu (eng. *Minimum-Redundancy Maximum-Relevance*) [16], metodu temeljenu na korelaciji značajki (eng. *Correlation-based feature selection*) [8] te kriterij nekonzistentnosti (eng. *Inconsistency criterion*) [14]. U ovom će se radu koristiti univarijatne metode za redukciju originalnog skupa značajki te metoda omotača uz pretraživanje s genetskim algoritmom za konačan odabir podskupa značajki što isključuje korištenje multivarijatne metode filtera.

2.6. Metode omotača

Za razliku od filter metoda, metode omotača uzimaju performanse algoritma strojnog učenja u obzir kod odabira značajki. Metrike korištene pri procjeni podskupa značajki najčešće su točnost modela te mjera prikladnosti modela (eng. *goodness of fit*). Prikladnost se odnosi na korelaciju rezultata dobivenog modela i stvarne raspodjele podataka.

Glavni nedostatak korištenja omotača velika je računalna složenost. Svaki je potencijalni podskup potrebno evaluirati treniranjem te testiranjem modela, što je moguće ubrzati isključivo pohranjivanjem vrijednosti za već viđene podskupove. Iako je strategija pretraživanja važna za metode filtriranja bitna i za metode filtriranja, upravo je veća računalna složenost razlog zašto za metode omotača strategija pretraživanja ima još veću važnost. Osim toga, pristranost korištenom modelu neizbježna je posljedica

metoda omotača. Iz tog razloga se u validaciji rezultata često koristi model različit od onog korištenog za evaluaciju.

Unatoč nedostacima, metode omotača u osnovi su djelotvornije od filter metoda jer se za testiranje podskupa koristi stvarni model, ne samo analitičke metode. [2]

Iako se u osnovni svaka kombinacija pretraživanja i modela može koristiti u metodi omotača, najčešće se radi o jednostavnim strategijama te brzim modelima, poput Naivnog Bayesovog klasifikatora te stroja s potpornim vektorima [9]. Iz tog će se razloga u ovom radu kao model koristiti upravo navedeni.

2.7. Hibridne metode

S ciljem kombiniranja pozitivnih strana obiju metoda, hibridne metode objedinjuju tijek filter metode i metode omotača. Metoda filtriranja koristi se za nalaženje potencijalnih podskupova značajki određene kardinalnosti ili se originalan podskup reducira za određeni faktor. Nakon toga, optimalan podskup odabire se korištenjem metode omotača.

2.8. Ugrađene metode

U nekim se modelima odabir značajki implicitno provodi tijekom izvođenja, gdje je ugrađen kao normalna ili dodatna funkcionalnost. Primjeri takvih metoda uključuju CART, C4.5 te slučajne šume. Osim toga, svaki model koji u svom izvođenju koristi L1-regularizaciju također se implicitno bavi odabirom značajki. Prilikom L1 (*Lasso*) regularizacije, koeficijenti značajki koje manje doprinose modelu pritežu se na vrijednost 0, što u konačnici rezultira u modelu koji sadržava samo bitne značajke.

S obzirom na to da se unutar ugrađenih metoda eksplicitno ne provodi odabir značajki, neće biti detaljnije prikazane u okviru ovog rada.

2.9. Cilj i kriterij zaustavljanja algoritma

Prilikom implementacije odabira značajki ključno je definirati krajnji cilj. Primjerice, fokus može biti na maksimalnoj redukciji broja značajki pri čemu se traži najmanji podskup s približno jednakim rezultatima kao i početni. S druge strane, može se tražiti najmanji podskup značajki koji nadmašuje rezultate početnog skupa ili u slučaju

vremenski osjetljivih problema prvi pronađeni podskup koji nadmašuje rezultate početnog.

U skladu s ciljem, bitno je odabrati kriterij zaustavljanja pretrage. Mogući kriteriji uključuju:

1. dosegnuto je određeno ograničenje, gdje ograničenje može biti minimalan broj odabranih značajki ili maksimalan broj iteracija;
2. slijedno dodavanje ili oduzimanje značajki nije unaprijedilo rezultat;
3. pronađen je dovoljno dobar podskup, npr. pogreška klasifikacije modela koji koristi podskup je ispod definiranog praga.

3. Genetski algoritam

3.1. Definicija

Evolucija (lat. *evolutio*: razvoj, razvitak) je naziv za proces uslijed kojeg dolazi do promjena nasljednih osobina bioloških populacija tijekom velikog broja uzastopnih generacija u svrhu prilagodbe na uvjete u okolini. Teorija evolucije uspostavljena je sredinom 19. stoljeća, a kao inspiracija u računarstvu koristi se zadnjih pedesetak godina. Razlog tome je to što se proces evolucije može interpretirati i kao proces pretraživanja prostora stanja.

Genetski algoritam dio je grane genetskog računarstva. Spada u heurističke metode optimiranja i svojim karakteristikama imitira prirodni evolucijski proces. Jedinka u biološkoj populaciji predstavlja jedno moguće rješenje problema, dok je usmjeravanje prema kvalitetnijim rješenjima ostvareno pomoću procjene kvalitete jedinki i određenih selekcijskih algoritama. U nastavku je prikazana analogija između prirodnog evolucijskog procesa i genetskog algoritma koji se temelji na njemu. [4]

evolucija	**genetski algoritam**
jedinka	jedno rješenje problema
populacija	skup rješenja
genom jedinke	genotip
mutacija genoma jedinke	unarni operatori
reprodukcija jedinki	operatori višeg reda
šanse za preživljavanje	dobrota

Tablica 3.1: Analogija između procesa evolucije i genetskih algoritama

3.2. Komponente algoritma

Prilikom odabira genetskog algoritma kao metode rješavanja problema bitno je uočiti da određene implementacijske odluke uvelike utječu na konačnu kvalitetu rezultata. Ključne komponente genetskog algoritma su sljedeće:

1. genotip, tj. struktura podataka koja će predstavljati rješenje
2. inicijalizacija početne populacije
3. funkcija dobrote, koja služi za evaluaciju kvalitete rješenja
4. operatori pomoću kojih se stvaraju nove jedinke (operatori križanja i mutacije)
5. postupak selekcije jedinki koje prelaze u sljedeću generaciju
6. uvjet zaustavljanja koji određuje koliko dugo se rješenje pretražuje

3.2.1. Genotip

Kvaliteta rješenja genetskog algoritma uvelike ovisi o odabiru strukture podataka koja predstavlja genom jedinke. Taj se odabir prvenstveno temelji na zahtjevima zadatka. Radni okviri i programi za GA nude već implementirane genotipe, gdje je su za svaki definirani i operatori križanja i mutacije.
Uobičajene strukture uključuju binarni kod, realan broj (koji se interpretira kao broj s pomičnom točkom s jednostrukom ili dvostrukom preciznošću), permutaciju cijelih brojeva te stablo.

3.2.2. Inicijalizacija populacije

Smjer u kojem će krenuti genetski algoritam uvelike određuje početna populacija jedinki. Ako je za problem moguće generirati mnoga nemoguća rješenja, prisutnost takvih u početnoj populaciji može kompromitirati konačno rješenje.

Tri su uobičajena načina za stvaranje početne populacije: jedinke se mogu generirati u odnosu na neko otprije poznato rješenje, generiranje može biti u potpunosti slučajno (u okviru restrikcija odabranog genotipa) ili se može koristiti pseudo-slučajno generiranje koje u obzir uzima poželjne karakteristike jedinke i omogućuje početak s *boljim* jedinkama.

Osim toga, bitno je odrediti veličinu populacije, koja u većini algoritama ostaje konstantna do završetka izvođenja.

3.2.3. Funkcija dobrote

U biološkom smislu *bolja* jedinka je ona bolje prilagođena na okolinu ili s velikim brojem poželjnih karakteristika. U kontekstu GA, jedinka (rješenje) je dobra ako zadovoljava zahtjeve problema te je svojim svojstvima bliska optimumu.

U optimizacijskom problemu može se tražiti minimalna ili maksimalna vrijednost funkcije cilja te je u skladu s tim jedinka *bolja* što je rezultat funkcije dobrote za tu jedinku manji ili veći, ovisno o definiciji problema. Funkcija dobrote izravno ovisi o cilju optimizacije. Primjerice, pri optimizaciji vrijednosti parametara neke polinomijalne funkcije, funkcija dobrote može biti upravo vrijednost ciljne funkcije. No, česta je praksa unutar funkcije dobrote ugraditi i dodatne karakteristike ili regularizaciju, s obzirom na to da je vrijednost dobrote glavni način usmjeravanja tijeka pretraživanja.

3.2.4. Operatori križanja i mutacije

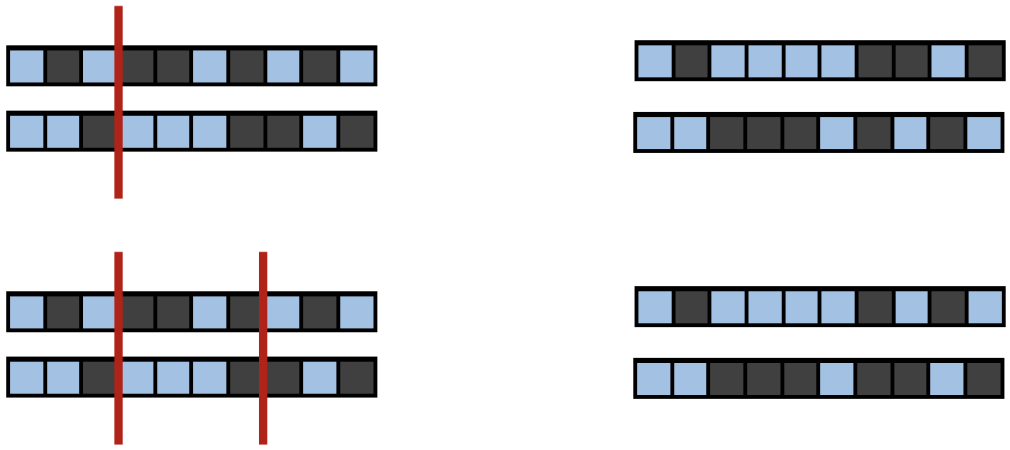
Potomci dobiveni spolnim razmnožavanjem uvijek su genetski različiti od roditeljskih jedinki te se i međusobno razlikuju. Razlog tome je rekombinacija gena.

U kontekstu algoritma, promjene u genotipu omogućuju kretanje do rješenja. Preko relativno malih te relativno slučajnih legalnih promjena unutar odabranih jedinki, genetski materijal se iz generacije u generaciju mijenja te se bliži optimumu (ako su korišteni parametri i operatori ispravni i prilagođeni problemu). Operatori se dijele na unarne, koji stvaraju novu jedinku mijenjajući manji dio genetskog materijala (operatori mutacije) te operatore višeg reda, koji kreiraju nove jedinke kombinirajući osobine dvaju jedinki (operatori križanja). Najčešće se unutar parametara algoritma određuje koja je vjerojatnost za mutaciju, a učestalost križanja i odabir jedinki koje ulaze u križanje ovisi o korištenom algoritmu selekcije. U nastavku su prikazani primjeri križanja i mutacije za binarni genotip.

Primjeri

Jedna od učestalih operatora križanja je križanje s jednom točkom prekida. Na početku postupka slučajnim se odabirom selektira točka presjeka unutar genotipa. Nakon toga, dobiveni novi genotip nastat će od odlomka prvog roditelja ispred presjeka te drugog roditelja nakon presjeka te obrnuto za drugu jedinku koja se analogno generira. Prednost ove metode, osim njene jednostavnosti, činjenica je da se njenim korištenjem istovremeno mogu generiraju dva različita potomka. Ekvivalentno, sličnih karakteristika je križanje s dvije točke prekida, gdje se ekvivalentno odabiru dva presjeka u

genotipu te se potomak generira iz naizmjeničnih odlomaka. Primjer križanja za binarni genotip duljine 10 prikazan je na slici 3.1. Nakon odabranih točaka presjeka, kombinacijom materijala iz roditeljskih jedinki generiraju se dva potomka s različitim genomom[1]. Prilikom korištenja ovog operatora križanja, kao i svih ostalih operatora, potrebno je obratiti pažnju na restrikcije genotipa i samog problema.



Slika 3.1: Ilustrativni prikaz križanja s jednom i dvije točke prekida

Mutacija znatno ovisi o korištenom genotipu. Korištenje binarnog koda omogućuje promjenu u jednom bitu bez razmatranja ostalih bitova, što nije slučaj kod nekih učestalih genotipa. Jednostavna mutacija se na binarnom kodu provodi uz parametar vjerojatnosti p_m koja predstavlja vjerojatnost mutacije jednog bita. Pripadajuća vrijednost mutacije kromosoma jednaka je $p_M = 1-(1-p_m)^n$, gdje n predstavlja dimenziju kromosoma. Za svaki bit u genotipu, ako za slučajno generiran broj x u intervalu $[0, 1]$ vrijedi $x < p_m$, vrijednost tog bita se komplementira. Na slici 3.2 prikazan je primjer mutacije u binarnom kodu. Strelicom je prikazana slučajno odabrana točka mutacije. Desno od početne jedinke prikazana je mutirana jedinka s promijenjenim 8. bitom.

Slika 3.2: Ilustrativni prikaz mutacije u binarnom kodu

[1] Ako je genetski materijal prije ili nakon presjeka u obje roditeljske jedinke jednak, generiraju se dva jednaka potomka. Ova se posljedica može spriječiti dodatnim provjerama.

3.2.5. Selekcija

Prelaskom iz generacije u generaciju potrebno je odabrati jedinke koje preživljavaju, tj. odraditi postupak selekcije. Svrha selekcije čuvanje je te prenošenje dobrih svojstava na sljedeću generaciju jedinki. No, pohlepan odabir isključivo najboljih jedinki u generaciji mogao bi rezultirati zapinjanjem u lokalnom optimumu. U nastavku su opisani neki uobičajeni selekcijski algoritmi za GA.

Jednostavna selekcija (eng. *Roulette wheel selection*) je metoda u kojoj je vjerojatnost odabira jedinke proporcionalna s njenom dobrotom. Iako bolje jedinke imaju veću šansu za preživljavanje, njihov opstanak nije siguran kao ni eliminacija slabijih jedinki. Rezultat toga je selekcija koja omogućuje određenu dozu raznolikosti u odabranoj novoj generaciji.

Eliminacijska selekcija djeluje suprotno od jednostavne selekcije. Na svakoj se generaciji obavlja selekcija, vjerojatnost odabira jedinke obrnuto je proporcionalna njenoj dobroti te se odabrane jedinke brišu iz populacije.

Turnirska selekcija je metoda korištena u ovoj implementaciji. Nakon odabira k jedinki (najčešće 3), najgora od njih mijenja se djetetom ostalih jedinki. Nova jedinka se zatim mutira te u tom obliku ubacuje u populaciju.

3.2.6. Uvjet zaustavljanja

Prilikom pokretanja genetskog algoritma bitno je odrediti kad će završiti s izvođenjem. Moguće je ograničiti broj evaluacija, koji definira koliko će se puta pokrenuti algoritam selekcije. Osim toga, može se zadati maksimalan broj uzastopnih generacija za koji se dopušta stagnacija najbolje vrijednosti funkcije dobrote, što omogućuje zaustavljanje algoritma koji je zapeo. Manje uobičajen uvjet zaustavljanja je vremensko ograničenje izvođenja.

3.3. Tijek izvođenja algoritma

Na početku izvođenja algoritma potrebno je inicijalizirati početnu populaciju. Nakon toga, slijedi proces koji se ponavlja sve dok nije ispunjen jedan od postavljenih uvjeta zaustavljanja. Proces se sastoji od djelovanja genetskih operatora selekcije, križanja i mutacije nad populacijom jedinki. Tijekom selekcije loše jedinke odumiru dok bolje opstaju te se u sljedećem koraku križaju, što je ekvivalent razmnožavanju. Križanjem se prenose određena svojstva roditelja na djecu, a mutacijom se slučajnom promjenom gena mijenjaju svojstva jedinke. Ponavljanjem tog postupka iz generacije u generaciju

postiže se sve veća prosječna dobrota. U nastavku je prikazan pseudokod opisanog tijeka algoritma.

```
Genetski algoritam {

      t = 0
      generiraj početnu populaciju potencijalnih rješenja P(0);
      sve dok (! zadovoljen uvjet završetka)
      {
            t = t + 1;
            selektiraj P'(t) iz P(t-1);
            križaj jedinke iz P'(t) i djecu spremi u P(t);
            mutiraj jedinke iz P(t);
      }
      ispiši rješenje;
}
```

Slika 3.3: Pseudokod genetskog algoritma

4. Praktičan rad

U svrhu proučavanja primjenjivosti evolucijskih algoritama kao strategije za pretraživanje prostora stanja značajki skupa podataka, u okviru ovog rada programski je ostvarena hibridna metoda odabira značajki. U nastavku su opisani detalji praktičnog rada te predstavljeni kvantitativni rezultati.

4.1. Skupovi podataka

Dimenzija skupa podataka može se definirati kao $N * m$, gdje N označava broj uzoraka (*retci* u skupu podataka), a m broj značajki (*stupci* u skupu podataka). Ipak, pojam dimenzionalnosti redovito se odnosi isključivo na broj značajki, s obzirom na analogiju preslikavanja u m dimenzionalni prostor. Iako se na natjecanjima za odabir značajki često koriste skupovi koji sadrže od nekoliko stotina do nekoliko tisuća značajki kako bi se uspješno replicirale dimenzionalnosti skupova iz područja bioinformatike, farmakologije, obrade teksta ili raspoznavanja uzoraka [6], testiranje metoda odabira značajki na daleko manjim skupovima je također smisleno, makar se češće koristi isključivo kao alat za optimizaciju modela.

Već za skupove umjerene veličine ($m > 20$) *brute-force* pretraživanje, koje bi u ovom kontekstu uključivalo treniranje ciljanog modela na svakom od $2^m - 1$ podskupova, nije izvedivo te se ne primjenjuje u praksi [13]. U nastavku su opisani skupovi podataka koji su za ovaj rad odabrani zbog različitih dimenzija, karakteristika i kompleksnosti.

4.1.1. Skup podataka *Wisconsin Diagnostic Breast Cancer*

Skup podataka *Wisconsin Diagnostic Breast Cancer*, kraće *WDBC*, dostupan je na [3]. Predstavlja problem binarne klasifikacije tumora dojke, gdje uzorak može predstavljati benignu ili malignu stanicu tumora. Sastoji se od 30 značajki realnih vrijednosti, gdje svaka od značajki predstavlja jednu od karakteristika analiziranih nakupina

stanica tumora. S obzirom na svoju dostupnost i važnost sadržaja, ovaj je skup podataka često korišten u istraživanjima za optimizaciju modela, hiperparametara te odabira značajki. Iz tog su razloga u literaturi dostupni mnogi prijašnji rezultati, prikladni za usporedbu uspješnosti odabira značajki.

4.1.2. Skup podataka *MADELON*

Skup podataka *MADELON* jedan je od pet skupova podataka korištenih na NIPS natjecanju za odabir značajki održanom 2003. godine [6] [7]. Sastoji se od 500 značajki s cjelobrojnim vrijednostima. Specifičnost ovog skupa podataka nije nužno u njegovoj dimenziji (ostala četiri skupa predstavljena u sklopu natjecanja sastojali su se od više stotina ili tisuća značajki), već u tome da je u potpunosti umjetno generiran te ne predstavlja nikakav stvarni problem. Skupovi točaka generirani normalnom raspodjelom raspoređeni su na vrhove kocke u 5-dimenzionalnom prostoru, te im je slučajnim odabirom dodijeljena jedna od dvije oznake klase. Od pet značajki koje označavaju dimenzije generirano je dodatnih 15, koje su informativne ali redundantne, te je osim toga ubačeno 480 značajki koje ne koreliraju s oznakom klase [6]. Svrha ovih značajki je ometanje modela, a natjecateljima je predstavljen izazov selekcije ispravnih značajki.

Ovaj je skup podataka odabran zbog svoje kompleksnosti i šire margine rezultata zbog svoje multilinearnosti i velikog broja značajki.

4.2. Korišteni programski paketi

U nastavku su prikazani programski okviri korišteni u radu. Za čitavu je implementaciju korišten programski jezik Python. Osim detaljnije opisanih, korišteni su i radni okviri *Pandas*, za učitavanje i obradu skupa podataka te *Matplotlib* za izradu grafičkih prikaza.

4.2.1. Radni okvir *DEAP*

Distributed evolutionary algorithms in Python, skraćeno *DEAP*, radni je okvir za programski jezik Python koji služi za implementaciju algoritama evolucijskog računarstva [17]. Unutar okvira dostupni su unaprijed implementirani algoritmi spremni za korištenje, poput jednostavnog genetskog algoritma, ali je omogućena i zasebna konfiguracija svake komponente algoritma. Stvaranje vlastitog genotipa, operatora te zapisa

rezultata je lako izvedivo, što *DEAP* čini prikladnim odabirom za implementaciju evolucijskih algoritama.

U ovom radu, genetski algoritam ostvaren je u radnom okruženju *DEAP*, a troturnirska selekcija implementirana je na temelju ugrađenih klasa genotipa te prikladnih operatora križanja i mutacije.

4.2.2. Radni okvir *ITMO_FS*

ITMO Feature Selection, skraćeno *ITMO_FS*, radni je okvir koji nudi implementaciju uobičajenih metoda za odabir značajki. Razvijen je na Sveučilištu ITMO u Sankt-Peterburgu. U sklopu ovog rada korištene su tri filter metode - simetrična nesigurnost, *Relief* te Fischer vrijednost, detaljnije objašnjene u nastavku.

4.2.3. Radni okvir *Scikit-learn*

Scikit-learn dobro je poznat, testiran i često korišten radni okvir za strojno učenje. U okviru ovog rada preuzeti su modeli te evaluacija modela korištenjem k-struke unakrsne provjere (eng. *k-fold validation*).

4.3. Komponente algoritma

Na slici 4.1 prikazan je apstraktan tok algoritma. Prvi je korak provođenje tri različite univarijatne metode filtera na čitavim skupovima podataka te se na temelju tri metrike (Relief, Fischer vrijednost te simetrična nesigurnost) u metodu omotača prosljeđuju podskupovi s poželjnim značajkama. Tada se pomoću genetskog algoritma pretražuje prostor stanja i evaluira podskupove na temelju modela.

Slika 4.1: Komponente implementiranog algoritma

4.3.1. Metode filtera

Kao korak koji prethodi genetskom algoritmu izvedene su tri filter metode temeljene na različitim karakteristikama - **simetrična nesigurnost** (eng. *Symmetrical uncertainty*), kao metrika temeljena na informaciji, **Relief**, kao metoda temeljena na udaljenosti te **Fischer vrijednost** kao statistička mjera. Za svaku od metrika proveden je jednak postupak koji se sastoji od sljedećih koraka:

1. za skup značajki i oznaku klase izračunata je ciljana metrika, gdje je svakoj značajki pridružena vrijednost metrike
2. značajke se sortiraju silazno po vrijednosti metrike
3. odabire se podskup značajki koji iznosi određeni postotak sortiranog skupa, s korakom od 5 posto, te se koristi za treniranje modela
4. podskup za koji model postiže najveću preciznost odabire se kao najbolji za određenu metriku

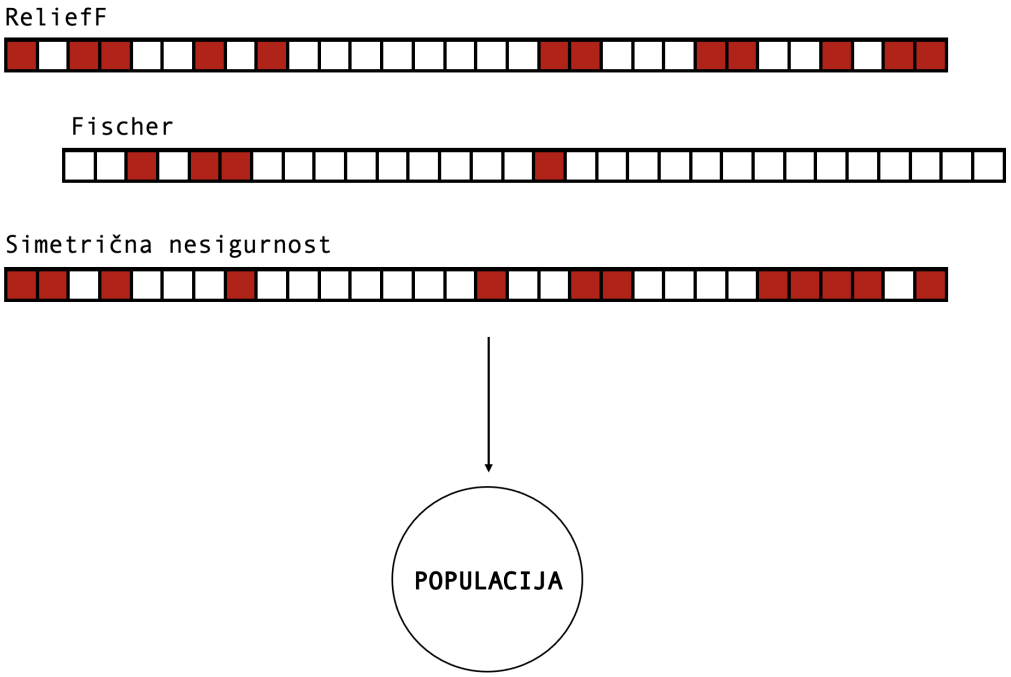
Nakon što se za svaku od tri metrike gornjim postupkom utvrdi najbolji podskup značajki, ova se informacija sprema, prebacuje u format genotipa te prosljeđuje dalje na sljedeći korak - inicijalizaciju populacije genetskog algoritma.

4.3.2. Inicijalizacija početne populacije

Most između metode filtera i genetskog algoritma čini inicijalizacija početne populacije. Radi se o pseudo-slučajnoj inicijalizaciji - osim slučajno generiranih genotipova na temelju uniformne razdiobe, dodatno se u svrhu usmjeravanja pretrage koriste informacije dobivene iz filter metode. Radi se o uobičajenom postupku u evolucijskim algoritmima (eng. *seeding*). S obzirom na karakteristike skupova podataka, rezultati filtera na različit se način ubacuju u populaciju.

Skup podataka WDBC

Za *WDBC* skup podataka koji sadrži 30 značajki, informacija o najboljim rezultatima filter metoda ubacuje se u genetski algoritam izravno preko jedinki. Za svaku od tri metrike utvrđen je najbolji podskup značajki za koji se onda generira pripadajući genotip (1 na mjestu odabrane značajke, 0 na mjestu značajke koja nije uključena u podskup, ukupne duljine 30). S pretpostavkom da su filter metode na ispravan način identificirale relevantne značajke, ubacivanje kvalitetnih jedinki u početnu populaciju može usmjeriti pretragu u povoljno područje što u konačnici može dati bolje rezultate.



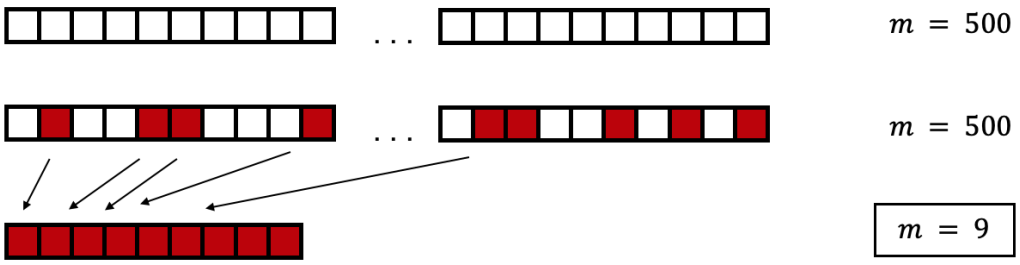
Slika 4.2: Ubacivanje povoljnih jedinki za *WDBC* skup podataka

Skup podataka MADELON

Za *MADELON* skup podataka incijalizacija populacije nije ostvarena na jednak način. S obzirom na veliku dimenziju skupa, što posljedično rezultira u velikom prostoru za pretraživanje ($2^{500} - 1$), ubacivanje jedinki dobivenih filter metodom nije optimalno rješenje za *seeding*. Velika dimenzija skupa utječe i na vrijeme treniranja i testiranja modela koji se koristi u funkciji dobrote, što algoritam čini i računalno zahtjevnijim.

Umjesto toga, reducira se dimenzija genotipa. Ograničiti genotip na dimenziju rezultata filtera omogućuje manji prostor pretraživanja, lokalizaciju od početka algoritma i potencijalno kvalitetnije krajnje rezultate. Ograničavanjem dimenzije genotipa, ograničavaju se i značajke - s obzirom da je unutar skupa podataka prisutan velik broj sintetičkih značajki koje ometaju model, drastična redukcija u prvom koraku može eliminirati većinu lažnih značajki, uz pretpostavku da je filter kvalitetno prepoznao važnost značajki.

Rezultati filter metoda ne mogu se istovremeno koristiti u incijalizaciji, već se koriste u odvojenim pokretanjima algoritma, zbog toga što se skup značajki reducira na one odabrane jednom od filter metoda. Na slici 4.3 prikazan je proces redukcije genotipa za skup podataka *MADELON*.



Slika 4.3: Redukcija veličine genotipa za *MADELON* skup podataka

4.3.3. Odabir modela

Kao što je već spomenuto u poglavlju 2.6, prilikom implementacije metode omotača najčešće se koriste jednostavni modeli, oni s brzim vremenom treniranja te evaluacije. Sofisticiraniji modeli koji su potrebni za rješavanje problema klasifikacije i regresije na skupovima koji su visoko nekorelirani te multilinearni zbog svog vremena treniranja i izvođenja rjeđe su korišteni, iako postižu bolju preciznost.

Iako u ovom radu nije fokus na postizanju maksimalne preciznosti modela za specifičan skup podataka, nego na primjenjivosti evolucijskih algoritama kao načina pretraživanja prostora stanja za dani problem, korištenje nedovoljno ekspresivnog modela moglo bi rezultirati u ograničavanju evolucijskog algoritma i ne bi u potpunosti prikazalo njegove mogućnosti. Stoga su odabrana dva različita modela u skladu sa složenošću skupova podataka - Naivan Bayesov klasifikator za *WDBC* i stroj potpornih vektora s mekom marginom za *MADELON*. U nastavku su ukratko opisani modeli.

Klasifikator Naivan Bayes

Klasifikator Naivan Bayes, čija je programska izvedba preuzeta s [10], temelji se na primjeni Bayesovog teorema o uvjetnoj nezavisnosti značajki skupa podataka u odnosu na oznaku klase. Radi se o jednostavnom modelu s brzim vremenom treniranja i izvođenja.

Stroj potpornih vektora

Stroj potpornih vektora (eng. *Support vector machine*, kraće *SVM*), čija je programska izvedba preuzeta s [11], često je korišten i konfigurabilan model za klasifikaciju. Klasifikacija radi na principu rješavanja problema maksimalne margine, gdje se traži granica između klasa koja je maksimalno udaljena od najbližeg uzorka iz svake klase.

Jednostavna inačica modela s *tvrdom* marginom pretpostavlja linearnu odvojivost. Uz pomoć *meke* margine omogućuje se rješavanje nelinearnih problema i povećava složenost modela. Mekoća margine kontrolira se hiperparametrom C, koji posljedično utječe na složenost modela.

Za primjenu modela na skupu podataka *MADELON* odabrana je vrijednost $C = 20$, dobivena k-strukom unakrsnom provjerom. Osim toga, koristi se RBF jezgra (funkcija preslikavanja linearnog produkta vektora je radijalna bazna funkcija), što smanjuje vrijeme izvođenja u odnosu na linearnu jezgru.

4.3.4. Metoda omotača - genetski algoritam

Genotip

Odabran je binarni genotip, gdje je dimenzija genotipa jednaka broju značajki skupa podataka (30 za *WDBC*, 500 za *MADELON*). Vrijednost 1 u genotipu označava da je značajka na tom indeksu uključena u reducirani podskup koji genotip predstavlja, dok 0 označava da određena značajka nije uključena. Tako je ukupan kardinalitet podskupa jednak broju vrijednosti 1 u genotipu.

Funkcija dobrote

Funkcija dobrote (eng. *fitness function*) jedna je od ključnih komponenti pri definiranju evolucijskog algoritma, ako ne i temeljna. Kao što je opisano u 3.2.3, funkcija dobrote se u potpunosti može preklapati s funkcijom cilja, no to nije nužno. U ovom je radu korištena sljedeća funkcija dobrote:

$$dobrota(C) = P(X_C) - penalizacija \tag{4.1}$$

gdje X_c označava podskup značajki predstavljen genotipom C, $P(X_C)$ preciznost modela treniranom na tom podskupu dobivena k-strukom unakrsnom provjerom s $k = 3$, a penalizacija predstavljena kao

$$penalizacija = w * |Xc| \tag{4.2}$$

gdje parametar w određuje važnost manjeg broja značajki - s porastom w raste i penalizacija, zbog čega pretraživanje favorizira podskupove s ne nužno najvećim rezultatom preciznosti modela ali s manjim brojem značajki.

Penalizacija je odabrana kao sredstvo za dodatno smanjenje broja značajki, uz prioritet preciznosti klasifikatora kao glavnog mjerila dobrote genotipa [15]. Iz tog razloga je bilo nužno podesiti iznos faktora penalizacije kao jednog od hiperparametara algoritma te je u analizi rezultata u nastavku rada razmatrana uloga penalizacije u krajnjim rješenjima.

Operatori križanja i mutacije

U algoritmu korišteni su operatori križanja i mutacije uobičajeni za binarni genotip, opisani u 3.2.4. Pri svakoj reprodukciji slučajno je odabran jedan od tri operatora:

- križanje s jednom točkom prekida
- križanje s dvije točke prekida
- uniformno križanje

Kao operator mutacije korištena je jednostavna mutacija, gdje vjerojatnost p_m odgovara vrijednosti mutacije jednog bita, te je pripadajuća vrijednost mutacije kromosoma jednaka $p_M = 1 - (1 - p_m)^n$, gdje n predstavlja dimenziju kromosoma (30 za *WDBC*, 500 za *MADELON*).

Selekcija

Kao metoda selekcije odabrana je troturnirska selekcija, često korištena u genetskim algoritmima. Iz populacije su slučajno odabrane tri jedinke koje se zatim silazno sortiraju po dobroti. Između dvije bolje jedinke obavlja se križanje i mutacija te se nastala jedinka ubacuje u populaciju umjesto najgore jedinke u turniru. Ovaj proces selekcije osigurava elitizam bez dodatnih mehanizama. Postupak se ponavlja ovisno o uvjetu zaustavljanja te se veličina populacije održava konstantnom kroz čitav algoritam.

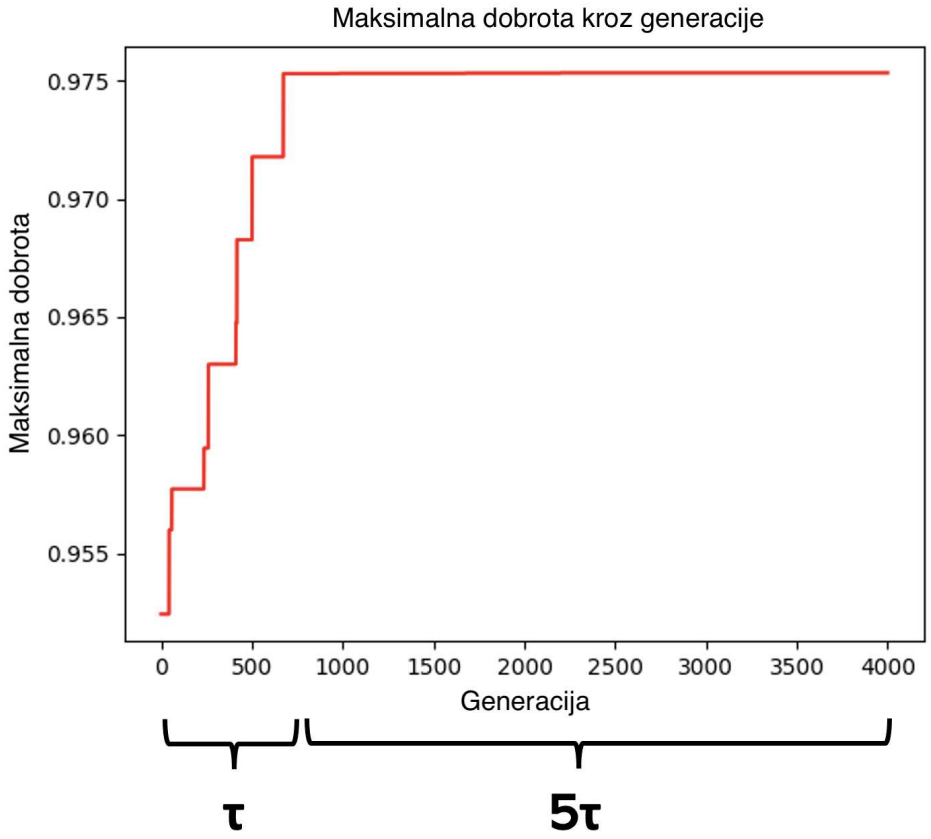
Odabir hiperparametara

Rezultati optimiranja mogu se znatno poboljšati finim podešavanjem parametara. Broj parametara ovisi o svojstvima algoritma i vrsti selekcije, za genetski algoritam s troturnirskom eliminacijskom selekcijom parametri koji se mogu varirati su veličina populacije, vjerojatnost mutacije i broj iteracija. Parametri se u pravilu određuju eksperimentalno. Veličina populacije i broj iteracija su indirektno vezani - što je populacija veća, rješenje je potencijalno kvalitetnije, ali je potreban veći broj iteracija da bi se stiglo do rješenja. Broj iteracija najčešće je ograničen vremenom izvođenja, osim toga može se zadati putem stagnacije u najboljem rješenju.

U ovom će se radu prvenstveno varirati dva parametra - faktor penalizacije broja značajki w i dodatak ili nedostatak informacija dobivenih iz filtera u početnu populaciju genetskog algoritma. Uvjet zaustavljanja definiran je u sljedećem poglavlju, dok su na temelju izvođenja uz fiksirane parametre odabrane vrijednosti 30 za veličinu populacije te 5% za vjerojatnost mutacije jednog bita genotipa.

Uvjet zaustavljanja

Za uvjet zaustavljanja odabran je broj iteracija (koji se za troturnirsku selekciju poklapa s brojem generacija) koji je utvrđen eksperimentalno na temelju krivulje maksimalne vrijednosti dobrote u populaciji po generacijama. Na slici 4.4 prikazan je primjer krivulje, gdje je period rasta vrijednosti maksimalne dobrote označen s τ. Poželjno trajanje perioda stagnacije vrijednosti za genetske algoritme iznosi 5τ, što je upravo slučaj na 4.4. Stoga je za broj generacija izabrana vrijednost od 4000.



Slika 4.4: Prikaz broja iteracija algoritma

5. Rezultati

U nastavku su prikazani rezultati algoritma, grupirani po komponentama. Analiziraju se rezultati metode filtera te genetskog algoritma, koji se pokreće s uključenom inicijalizacijom početnih rješenja ili bez te s drugačijom vrijednosti faktora penalizacije.

5.1. Metode filtera

Postupkom opisanim u 4.3.1, za svaku od tri korištene metrike (simetrična nesigurnost, Relief i Fischer vrijednost) utvrđen je optimalan podskup značajki za *WDBC* i *MADELON* skupove podataka. Postotak i broj odabranih značajki te preciznost modela prikazani su u tablicama 5.1, 5.2 i 5.3. Maksimalna vrijednost preciznosti dodatno je istaknuta.

Skup podataka WDBC

Rezultati metode filtera za skup podataka *WDBC* prikazani su u tablici 5.1. Preciznost je dobivena k-strukom unakrsnom provjerom preciznosti s $k = 3$ za Naivan Bayesov klasifikator.

metrika	**postotak značajki [%]**	**broj značajki**	**preciznost**
bez odabira značajki	100	30	**0.942**
simetrična nesigurnost	95	28	0.940
Fischer vrijednost	80	24	**0.942**
ReliefF	95	28	0.940

Tablica 5.1: Rezultati filtera za *WDBC* skup podataka

Preciznost klasifikatora na punom skupu značajki iznosi 0.942. Korištenjem metrika postignut je jednak ili neznatno slabiji rezultat. Odabrani broj značajki iznosi najmanje 80%.

Skup podataka MADELON

Rezultati metode filtera za skup podataka *MADELON* i model naivnog Bayesovog klasifikatora prikazani su u tablici 5.2. Preciznost je dobivena k-strukom unakrsnom provjerom preciznosti s $k = 3$. Vrijednost preciznosti klasifikatora na punom skupu značajki iznosi niskih 0.596, što pokazuje da model Bayesovog klasifikatora nije dovoljno složen za ovakav skup podataka. Rezultati filtera ne nadmašuju preciznost u velikoj mjeri, najbolja vrijednost postiže se Relief metrikom, gdje je dosegnuta preciznost od 0.625 korištenjem podskupa od 125 značajki.

metrika	**postotak značajki [%]**	**broj značajki**	**preciznost**
bez odabira značajki	100	500	0.596
simetrična nesigurnost	95	475	0.577
Fischer vrijednost	55	275	0.608
ReliefF	25	125	**0.625**

Tablica 5.2: Rezultati filtera za skup podataka *MADELON* i Naivni Bayesov klasifikator

Iako ovi rezultati mogu ukazati i na slabu primjenjivost filtera, s obzirom na karakteristike skupa podataka vjerojatnije je da je problem u modelu. Stoga je osim Naivnog Bayesovog klasifikatora isproban i stroj potpornih vektora s parametrom $C = 20$. Rezultati su prikazani u tablici 5.3. Preciznost klasifikatora na punom skupu značajki iznosi 0.669, što pokazuje složenost klasifikacije na ovom skupu podataka, čak i uz korištenje složenijeg modela.

metrika	**postotak značajki [%]**	**broj značajki**	**preciznost**
bez odabira značajki	100	500	0.669
simetrična nesigurnost	85	425	0.583
Fischer vrijednost	5	25	0.838
ReliefF	5	25	**0.868**

Tablica 5.3: Rezultati filtera za skup podataka *MADELON* i stroj potpornih vektora

U ovom su slučaju podskupovi odabrani metrikama u dva slučaja znatno nadmašili rezultat modela na punom skupu podataka. Za Fischer vrijednost i ReliefF odabrano je 25 značajki od polaznih 500 te je postignuta preciznost veća od 0.85. Korištenjem simetrične nesigurnosti odabrano je 85% značajki te je postignuta preciznost od 0.583, manja od rezultata na punom skupu podataka, što ukazuje na to da simetrična nesigurnost nije dobar filter za skup podataka s takvim karakteristikama. Iz tog razloga se

u sljedećem koraku algoritma neće razmatrati podskup odabran simetričnom nesigurnosti.

Činjenica da je korištenjem Relief i Fischer metrika odabran minimalan podskup od 25 značajki ukazuje na to da metrike potencijalno ispravno filtriraju ključnih 20 značajki (5 informativnih i 15 redundantnih, objašnjeno u 4.1.2). Osim toga, od 25 odabranih značajki, 14 značajki se poklapa u obje metrike, što može biti dodatan indikator da su identificirane ispravne značajke.

5.2. Varijacija penalizacije

U svrhu postizanja cilja pronalaska podskupa značajki skupa podataka koji postiže što veću točnost modela uz što manji kardinalitet, u funkciju dobrote genetskog algoritma ubačen je faktor penalizacije broja značajki w. Funkcija dobrote predstavljena je formulom 4.1.

U cilju pronalaska optimalnog iznosa penalizacije izvedeno je testiranje fiksiranjem ostalih hiperparametara. Faktor je testiran u rasponu [0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}].

Nakon odabranog faktora penalizacije provodi se sljedeći korak ispitivanja - inicijalizacija populacije uz pomoć rezultata filtera.

Skup podataka WDBC

U tablici 5.4 prikazani su rezultati maksimalne dobrote, pripadajuće preciznosti klasifikatora i broja značajki odabranih nakon 10 pokretanja genetskog algoritma s 4000 generacija i modelom Naivnog Bayesovog klasifikatora.

penalizacija	**dobrota**	**preciznost**	**broj značajki**
0	0.97540	0.97540	10
10^{-6}	0.97532	0.97540	8
10^{-5}	0.97460	0.97540	8
10^{-4}	0.96740	0.97540	8
10^{-3}	0.94540	0.96310	3

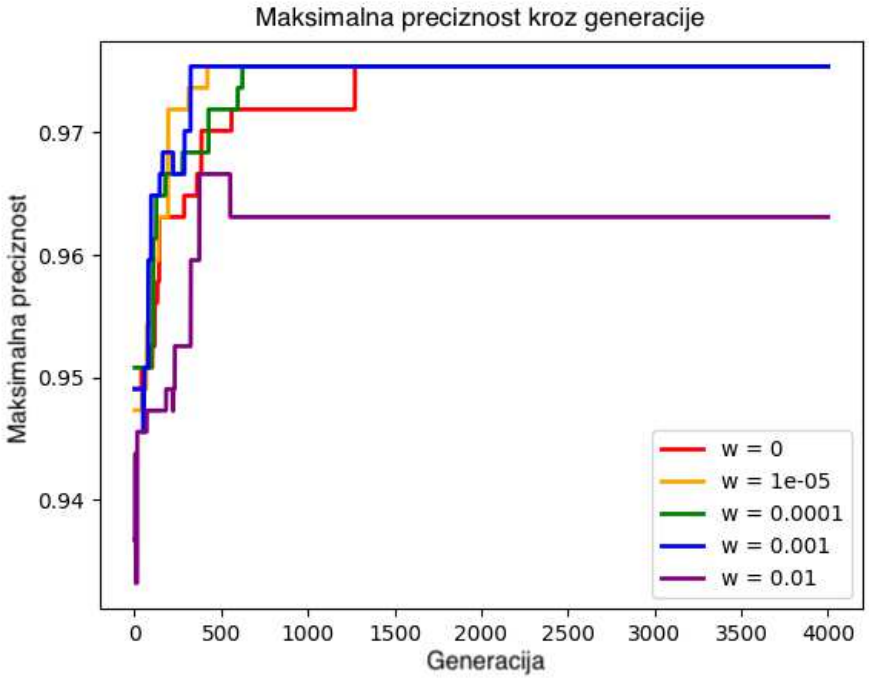
Tablica 5.4: Različit faktor penalizacije za skup podataka *WDBC*

U 10 iteracija algoritma bez korištenja penalizacije, gdje je dobrota jedinke jednaka srednjoj vrijednosti preciznosti klasifikatora s 3-strukom unakrsnom provjerom, najbolja jedinka postigla je preciznost od 0.97540 s 10 značajki. Četiri različita fak-

tora penalizacije postigla su jednaku preciznost klasifikatora (10^{-6}, 10^{-5}, 10^{-4}), gdje je odabrano istih 8 značajki u svakom od slučaja, što ukazuje na to da su u slučaju bez penalizacije uključene i 2 redundantne značajke. Najbolja jedinka za faktor penalizacije od 10^{-2} sastoji se od 3 značajke te postiže visoku preciznost od 0.96310. Te su tri značajke uključene i u podskupove ostalih rezultata.

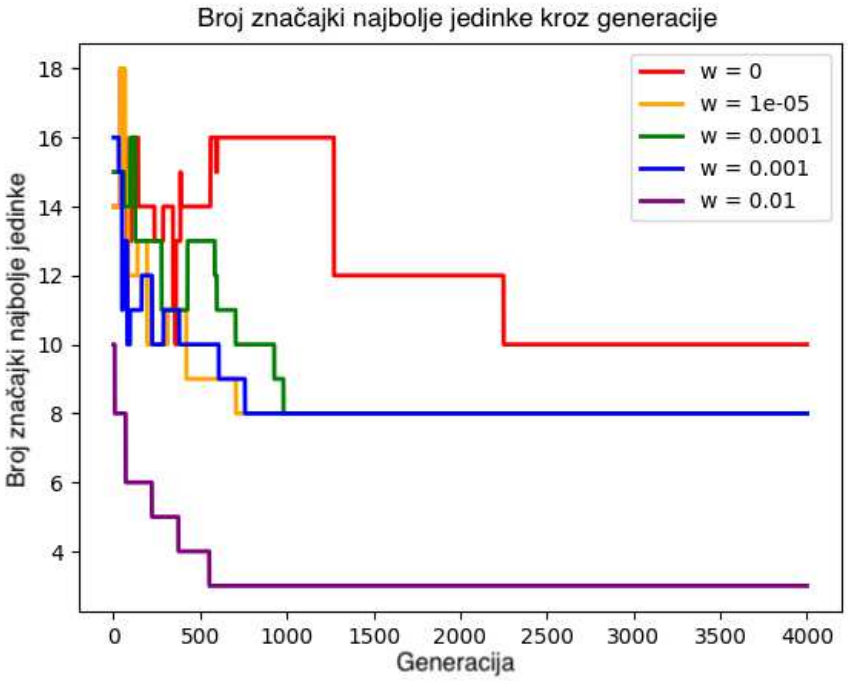
S obzirom na rezultate može se zaključiti da je potencijalna maksimalna preciznost koju Naivan Bayesov klasifikator može dostići za ovaj skup podataka 0.975 te da je skup od 8 značajki optimalan za tu kombinaciju parametara. Radi se o poboljšanju od 4% u odnosu na rezultat klasifikatora bez odabira značajki, što je znatno poboljšanje za preciznost veću od 90%. Uz varijaciju modela te hiperparametara algoritma vrlo je lako moguće da se preciznost može dodatno poboljšati.

Na slici 5.1 prikazano je kretanje maksimalne preciznosti klasifikatora kroz generacije najboljih pokretanja za svaku vrijednost penalizacije. Vidljiv je jednak trend rasta preciznosti za sve vrijednosti w, uz iznimku vrijednosti $w = 0.01$, zbog povremenog favoriziranja podskupa za koji klasifikator postiže slabiju preciznost, ali je odabran manji broj vrijednosti. Ovo dokazuje da je parametar penalizacije broja značajki moguće koristiti za regulaciju cilja odabira značajki - ako je cilj maksimalna redukcija uz zadržavanje određene preciznosti klasifikatora, moguće je povećati vrijednost w.



Slika 5.1: Kretanje preciznosti ovisno o penalizaciji za skup podataka *WDBC*

Na slici 5.2 prikazano je kretanje broja odabranih značajki genotipa s najvećom dobrotom u populaciji. Vidljivo je da se kroz generacije postupno odabiru sve manji podskupovi značajki.



Slika 5.2: Kretanje dimenzije podskupa ovisno o penalizaciji za skup podataka *WDBC*

Skup podataka MADELON

U tablici 5.5 prikazani su rezultati za skup podataka *MADELON* i stroj potpornih vektora s parametrom $C = 20$. Faktor penalizacije je u rasponu [0, 10^{-6}, 10^{-5}, 10^{-4}]. Faktor $w = 10^{-3}$ nije uključen zbog izbjegavanja negativne dobrote jedinki i gubitka prioriteta preciznosti klasifikatora u tom slučaju.

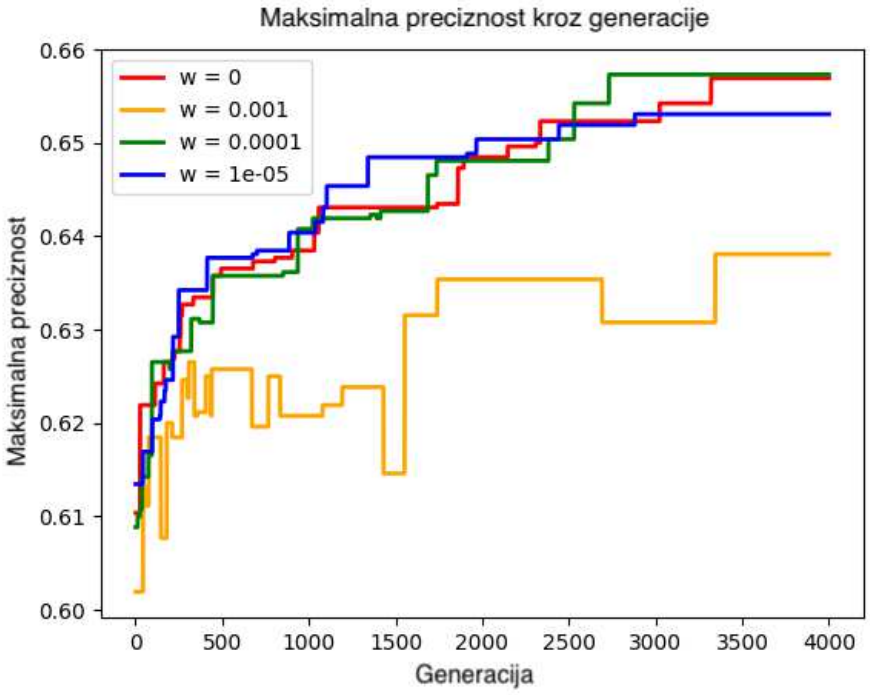
penalizacija	**dobrota**	**preciznost**	**broj značajki**
0	0.65692	**0.65692**	250
10^{-6}	0.65079	0.65307	228
10^{-5}	0.63582	0.65732	215
10^{-4}	0.48508	0.63808	153

Tablica 5.5: Različit faktor penalizacije za skup podataka *MADELON*

Kao i u slučaju skupa podataka *WDBC*, genetski algoritam je za skup podataka

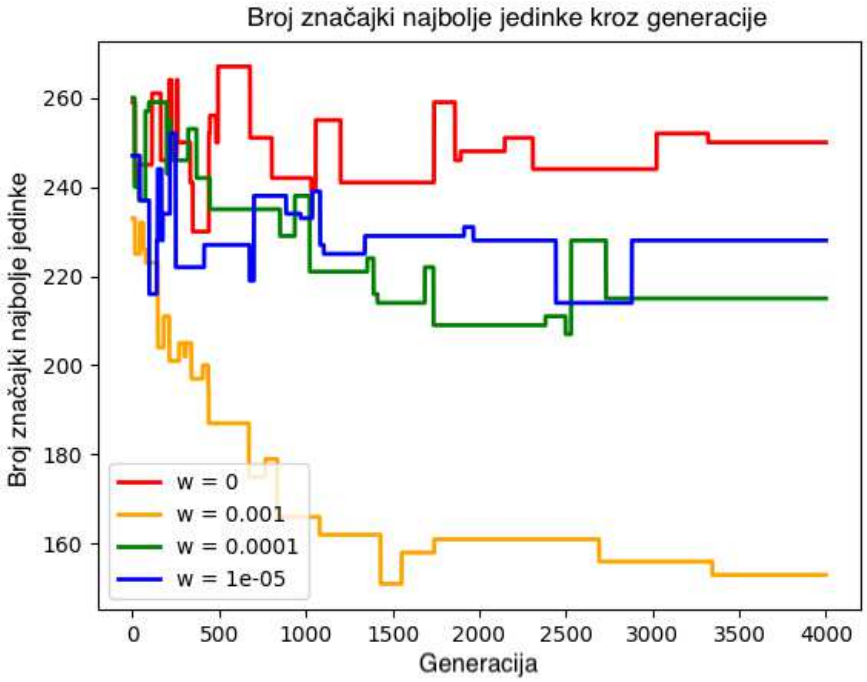
MADELON uz svaki stupanj penalizacije pronašao podskup koji daje veću preciznost za model nego bez odabira značajki ili korištenjem neke metrike filtera. Maksimalna preciznost od 0.65692 postignuta je bez korištenja penalizacije, no uz najveći podskup od 250 značajki, što je točno polovica od punog skupa značajki. U ostalim se rezultatima uočava pravilnost, gdje se uz veću penalizaciju smanjuje preciznost klasifikatora, ali i broj značajki.

Na slici 5.3 prikazano je kretanje maksimalne dobrote u populaciji kroz generacije. Radi se o krivuljama sličnog oblika kao i za skup podataka *WDBC*, no uz kraći period stagnacije. Zbog veličine prostora stanja za ovaj je skup podataka potreban veći broj iteracija GA za postizanje boljih rezultata, no to nije provedeno u ovom radu zbog neprikladnosti modela za optimizaciju ovog skupa podataka. Za skup koji je izrazito nelinearan i sadrži velik broj lažnih značajki potreban je složeniji model, te se u praksi definitivno ne bi koristio Naivni Bayesov klasifikator. Stoga je za daljnje eksperimente na skupu podataka korištena redukcija dimenzije genotipa i time prostora stanja koji se pretražuje genetskim algoritmom. Rezultati su prikazani u sljedećem poglavlju.



Slika 5.3: Kretanje preciznosti ovisno o penalizaciji za skup podataka *MADELON*

Na slici 5.4 prikazano je kretanje dimenzija odabranog podskupa kojeg predstavlja najbolja jedinka u populaciji po generacijama.



Slika 5.4: Kretanje dimenzije podskupa ovisno o penalizaciji za skup podataka *MADELON*

5.3. Inicijalizacija populacije uz filter

Skup podataka WDBC

Uz pseudo-slučajnu inicijalizaciju populacije za WDBC skup podataka, gdje se osim slučajno generiranih genotipova u početnu populaciju ubacuju tri jedinke koje predstavljaju najbolje rezultate filter metoda, nije postignut napredak u preciznosti klasifikatora ili manjem broju odabranih značajki. S 8 odabranih značajki, dosegnuta je preciznost od 0.97540, jednaka kao i bez inicijalizacije uz jedinke iz filtera.

Razlog tome je preciznost klasifikatora za odabrane genotipove koja je niža od najboljeg genotipa nađenog s GA. Slabije jedinke iz tog razloga nemaju utjecaj na konačnu kvalitetu rješenja.

inicijalizacija	maksimalna preciznost	broj značajki
slučajno	0.97540	8
pseudo-slučajno (*seeding*)	0.97540	8

Tablica 5.6: Preciznost za različite načine inicijalizacije, *WDBC*

Skup podataka MADELON

Kao što je objašnjeno u 4.3.2, za *MADELON* nije korišten jednak način inicijalizacije populacije. S obzirom na dimenziju od 500 značajki, ispravna redukcija prostora značajki uvelike može poboljšati performanse genetskog algoritma. Stoga su rezultati najuspješnije filter metode *Relief* korišteni za ograničavanje značajki i time duljine genotipa. Koristeći znatno manji skup značajki ubrzalo je vrijeme treniranja i testiranja modela, što je GA učinilo dovoljno brzim za korištenje - iako je prihvatljivo vrijeme izvođenja relativno, za stroj potpornih vektora s RBF jezgrom vrijeme izvođenja jednog pokretanja GA na punom skupu značajki iznosilo je 2 sata, za razliku od reduciranog skupa čije je trajanje iznosilo 20 minuta.

U tablici 5.7 prikazan je najbolji rezultat preciznosti modela nakon 10 pokretanja GA s reduciranim prostorom značajki dimenzije 25. Odabrano je 11 značajki te je postignuta preciznost od 0.891, što je poboljšanje u odnosu na rezultat *Relief* filtera i drastično poboljšanje u odnosu na rezultat modela bez odabira značajki.

metoda	maksimalna preciznost	broj značajki
Relief filter	0.868	25
Relief inicijalizacija	**0.891**	11

Tablica 5.7: Preciznost za različite načine inicijalizacije, *MADELON*

6. Zaključak

Trend porasta dimenzija skupova podataka definitivno se neće zaustaviti u budućnosti, s porastom izvora i broja informacija i jačanjem procesorske snage. Stoga će i važnost problema odabira značajki isključivo rasti, što povećava značaj istraživanja najboljih metoda za njegovo rješavanje.

Optimalna metoda za odabir značajki, kao i optimalan model, ne postoje (ili još nisu otkriveni). Osim toga, ciljevi odabira značajki mogu biti različiti - ovisno o tome je li u fokusu maksimalna preciznost modela, drastična redukcija značajki ili brzo vrijeme izvođenja potrebno je odabrati prikladnu metodu.

Jedna od mogućnosti je korištenje metode filtera, koje uz pozitivnu stranu brzog vremena izvođenja ipak generalno postižu slabije rezultate od metoda omotača. U praktičnom dijelu ovog rada korištene su tri univarijatne filter metode, svaka temeljena na drugoj metrici - simetričnoj nesigurnosti, Fischer vrijednosti i *Relief* metodi. Između tri testirane metode najgori rezultati postignuti su korištenjem metrike simetrične nesigurnosti, dok su najbolji rezultati postignuti *Relief* metrikom, čija je prednost detektiranja međusobnog odnosa između značajki.

S druge strane, metode omotača uz mane sporijeg vremena izvođenja i pristranosti modelu, postižu kvalitetnije rezultate. U praktičnom dijelu ovog rada metoda omotača uz korištenje genetskog algoritma za pretraživanje prostora stanja nadmašila je rezultate svih metoda filtera na oba skupa podataka. Brži od unaprijednog ili unatražnog odabira te provodeći samo djelić evaluacija u usporedbi s *brute-force* pretraživanjem, genetski algoritmi prikladno su rješenje za ovu primjenu. Ipak, nedostatak ovog pristupa uključuje nedeterminizam genetskih algoritama i dulje vrijeme izvođenja za kompleksnije modele i veće skupove podataka. Osim toga, kao i u ostalim inačicama metode omotača, uspješnost odabira značajki poglavito se temelji na modelu, stoga bi za unaprjeđenje rezultata praktičnog rada i njegovu dodatnu validaciju bilo nužno koristiti veći skup modela te druge modele uklopiti u provjeru konačnih rezultata.

Bez obzira na definirani cilj ili odabranu metodu, odabir značajki korak je u stvaranju sustava strojnog učenja koji ne bi smio izostati.

LITERATURA

[1] Heba Abusamra. A comparative study of feature selection and classification methods for gene expression data of glioma. *Procedia Computer Science*, 23:5–14, 2013.

[2] Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. U *Icml*, svezak 1, stranice 74–81. Citeseer, 2001.

[3] Dheeru Dua i Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[4] Marin Golub et al. *Genetski algoritam - Prvi dio*. Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, 2004.

[5] Quanquan Gu, Zhenhui Li, i Jiawei Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

[6] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, i Gideon Dror. Result analysis of the nips 2003 feature selection challenge. *Advances in neural information processing systems*, 17, 2004.

[7] Isabelle Guyon, Jiwen Li, Theodor Mader, Patrick A Pletscher, Georg Schneider, i Markus Uhr. Feature selection with the clop package. *Technical Report*, 2006.

[8] Mark A Hall i Lloyd A Smith. Practical feature subset selection for machine learning. 1998.

[9] Alan Jović, Karla Brkić, i Nikola Bogunović. A review of feature selection methods with applications. U *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, stranice 1200–1205. Ieee, 2015.

[10] Scikit learn razvojni tim. *Naive Bayes*, . URL `https://scikit-learn.org/stable/modules/naive_bayes.html`.

[11] Scikit learn razvojni tim. *SVC*, . URL https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[12] Chao Liu, Dongxiang Jiang, i Wenguang Yang. Global geometric similarity scheme for feature selection in fault diagnosis. *Expert Systems with Applications*, 41(8):3585–3595, 2014. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2013.11.037. URL https://www.sciencedirect.com/science/article/pii/S0957417413009573.

[13] Huan Liu i Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005. doi: 10.1109/TKDE.2005.66.

[14] Huan Liu, Rudy Setiono, et al. A probabilistic approach to feature selection-a filter solution. U *ICML*, svezak 96, stranice 319–327. Citeseer, 1996.

[15] Il-Seok Oh, Jin-Seon Lee, i Byung-Ro Moon. Hybrid genetic algorithms for feature selection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11):1424–1437, 2004.

[16] Hanchuan Peng, Fuhui Long, i Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[17] DEAP razvojni tim. Deap framework. URL https://deap.readthedocs.io/en/master/.

[18] Dongkoo Shon, Kichang Im, Jeong-Ho Park, Dong-Sun Lim, Byungtae Jang, i Jong-Myon Kim. Emotional stress state detection using genetic algorithm-based feature selection on eeg signals. *International Journal of environmental research and public health*, 15(11):2461, 2018.

[19] Zhou Tao, Lu Huiling, Wang Wenwen, i Yong Xia. Ga-svm based feature selection and parameter optimization in hospitalization expense modeling. *Applied Soft Computing*, 75:323–332, 2019. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2018.11.001. URL https://www.sciencedirect.com/science/article/pii/S1568494618306264.

[20] Ryan J Urbanowicz, Melissa Meeker, William La Cava, Randal S Olson, i Jason H Moore. Relief-based feature selection: Introduction and review. *Journal of biomedical informatics*, 85:189–203, 2018.

[21] Betty Wutzl, Kenji Leibnitz, Frank Rattay, Martin Kronbichler, Masayuki Murata, i Stefan Martin Golaszewski. Genetic algorithms for feature selection when classifying severe chronic disorders of consciousness. *PLOS ONE*, 14(7):1–16, 07 2019. doi: 10.1371/journal.pone.0219683. URL https://doi.org/10.1371/journal.pone.0219683.

Odabir značajki skupa podataka uz pomoć evolucijskih algoritama

Sažetak

U ovom je radu obrađen problem odabira značajki skupa podataka korištenjem evolucijskih algoritama kao metode pretraživanja prostora stanja. Predstavljena je teorijska osnova odabira značajki i evolucijskih algoritama te je programski izveden algoritam koji kombinira metode filtera i metodu omotača, koristeći simetričnu nesigurnost, Relief i Fischer metriku te stroj potpornih vektora i klasifikator Naivan Bayes. Kvaliteta algoritma ispitana je na dva skupa podataka s različitim karakteristikama, *Wisconsin Diagnostic Breast Cancer* i *MADELON*.

Ključne riječi: odabir značajki, genetski algoritam, optimizacija, model strojnog učenja

Data set feature selection using evolutionary algorithms

Abstract

In this paper feature selection using evolutionary algorithms was discussed. The theoretical basis of feature selection and evolutionary algorithms was given, as well as a potential solution combining filter and wrapper methods, including the Relief, Fischer and symmetrical uncertainty measures as well as Naive Bayes and SVM models. The algorithm's quality was assessed on two datasets with differing characteristics, *Wisconsin Diagnostic Breast Cancer* and *MADELON*.

Keywords: feature selection, genetic algorithm, optimisation, machine-learning model