USPOREDBA ALGORITAMA PREUZORKOVANJA ZASNOVANIH NA SMOTE ZA RJEŠAVANJE PROBLEMA NEURAVNOTEŽENOSTI KLASA

Rojnić, Domagoj

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Preddiplomski sveučilišni studij

USPOREDBA ALGORITAMA PREUZORKOVANJA ZASNOVANIH NA SMOTE ZA RJEŠAVANJE PROBLEMA NEURAVNOTEŽENOSTI KLASA

Završni rad

Domagoj Rojnić

Osijek, 2020.

SADRŽAJ

1. UVOD

U današnje vrijeme svaki aspekt života pruža određene podatke. Ti podatci prikupljaju se u digitalnom obliku u velikim količinama te ih je potrebno obraditi u svrhu dobivanja informacija. U tu svrhu se nude mnogi postupci strojnog učenja, a algoritmi klasifikacije ili klasifikatori često su prikladni za mnoge primjene i probleme. Temeljem prethodno točno označenih uzoraka ovisno o njihovim značajkama, zadaća algoritma je nove uzorke razvrstati u skupine definirane oznakama. Problem se javlja ukoliko su uzorci neuravnoteženo raspoređeni, odnosno ukoliko su uzorci jedne klase u znatno većem broju u odnosu na brojnost uzoraka druge klase, što je poznato kao problem neuravnoteženosti klasa. Manji broj uzoraka pripada takozvanoj manjinskoj klasi, a često je upravo ona od interesa. Međutim, prilikom suočavanja s neuravnoteženim podatcima, klasifikator, odnosno uspostavljeni model pristran je većinskoj klasi. Kako bi se ublažio ovaj problem, moguće je donekle izjednačiti brojnost većinskih i manjinskih uzoraka. Dva jednostavna načina su metode preuzorkovanja i metode poduzorkovanja kojima je moguće postići ravnotežu brojnosti primjeraka pojedine klase temeljem dostupnih podataka. Preuzorkovanjem se povećava brojnost uzoraka manjinske klase, dok se poduzorkovanjem briše određen broj uzoraka većinske klase. Međutim, ove metode mogu dovesti do različitih problema poput mogućeg brisanja važnih uzoraka prilikom poduzorkovanja ili dupliciranja podataka pri prostom preuzorkovanju. S obzirom da je prethodno navedena mogućnost gubljenja bitnih primjeraka većinske klase značajnija, sigurnije je korištenje metoda preuzorkovanja. Jedan od popularnih i učinkovitih metoda preuzorkovanja je takozvana tehnika sintetičkog preuzorkovanja manjinskih uzoraka, odnosno algoritam SMOTE (*Synthetic Minority Over-sampling Technique*) kojim su sintetički uzorci stvoreni temeljem dostupnih manjinskih uzoraka. Algoritam se sastoji od nasumičnog odabira susjeda na temelju kojeg postojeći manjinski uzorci stvaraju nove, sintetičke uzorke. Nedostatak algoritma leži u odabiru svih uzoraka manjinske klase prilikom čega su zanemareni većinski uzorci. Ovakav način odabira može dovesti do povećanog preklapanja uzoraka različitih klasa i povećanog utjecaja šumnih uzoraka. Kako bi se ublažili određeni nedostatci, predložena su brojna proširenja i unaprijeđenja koja povećavaju učinkovitost izgrađenih klasifikatora. Međutim, sva proširenja nisu ista te imaju različite utjecaje na stvaranje sintetičkih podataka.

Drugo poglavlje daje detaljniji uvid u algoritam klasifikacije i problem neuravnoteženosti klasa, detaljno su obrađeni algoritam *k*-najbližih susjeda te je dan opis različitih mjera kojima se vrednuje učinkovitost algoritama klasifikacije. Uz to, opisan je algoritam SMOTE sa svojim

prednostima i nedostatcima te su opisane neke unaprijeđene varijante koje teže povećanju učinkovitosti originalnog algoritma kroz njegova različita proširenja. Treće poglavlje daje uvid u ostvareno programsko rješenje te način rada i uporabe istog. U četvrtom poglavlju dane su postavke provedenih eksperimenta te su prikazani i objašnjeni dobiveni rezultati.

1.1. Zadatak završnog rada

Zadatak je opisati problem klasifikacije. Opisati što predstavlja neuravnoteženosti klasa i kako utječe zadatak klasifikacije uzoraka. Opisati algoritam SMOTE kao popularni i učinkoviti pristup preuzorkovanju koji se koristi samo dostupnim podacima te nekoliko njegovih unaprijeđenih varijanti. U praktičnom dijelu rada nužno je razviti programsko rješenje koje omogućuje preuzorkovanje danog skupa podataka pomoću algoritma SMOTE i barem dvije unaprijeđene varijante iz literature. Također, potrebno je napraviti eksperimentalnu analizu na nekoliko različitih skupova podataka.

2. PROBLEM NEURAVNOTEŽENOSTI KLASA I ALGORITAM SMOTE

Općenito gledajući, klasifikacijom [1] se razvrstavaju podatci u određene klase temeljem prethodno označenih podataka, odnosno uzoraka i njihovih značajki. Ukoliko je brojnost uzoraka jedne klase uvelike manja od brojnosti druge, dolazi do problema poznatog kao problem neuravnoteženosti klasa. Podatci su neravnotežno raspoređeni u klase što rezultira pristranosti klasifikatora većinskoj klasi. Kako bi se ublažio ovaj problem, predloženi su različiti algoritmi preuzorkovanja i poduzorkovanja [5], gdje je preuzorkovanje vjerojatno više zastupljeno u literaturi, a čiji je najpoznatiji predstavnik algoritam SMOTE [10].

2.1. Klasifikacija i neuravnoteženost klasa

Klasifikacija podataka [1, 2, 3] podrazumijeva proces određivanja oznake neviđenim podatcima temeljem već poznatog konačnog skupa oznaka klasa. Tih klasa može biti dvije ili više, no u ovom radu će se fokusirati na klasifikaciju u dvije klase što se naziva binarnom klasifikacijom. Prema [1], primjenu klasifikacije moguće je pronaći u različitim područjima poput medicine, gdje se na temelju podataka o pacijentovom zdravstvenom stanju utvrđuje ima li određenu bolest ili ne, u bankovnim sustavima, gdje se određuje je li klijentu adekvatno i profitabilno izdati kredit, u sustavima elektroničke pošte, gdje se određuje spada li nova pošta u neželjeni sadržaj ili ne i slično.

Klasifikacija se uobičajeno sastoji od dvije faze, trening faze i testne faze. Podatci su oblika (x,y) gdje x predstavlja ulaz, a y izlaz odnosno oznaku. Dakle, dostupno je n ulaznih uzoraka $x_1,...,x_n$ gdje je svaki uzorak opisan s m značajki te su poznate vrijednosti izlaza kao skup diskretnih vrijednosti $y = \{y_1, y_2,...,y_k\}$. Algoritam klasifikacije je općenito funkcija $f(x) = y$. Grubo rečeno, prilikom treninga se pokušava odrediti f(x) kako bi se u testnoj fazi za nove ulaze x mogao odrediti izlaz y. Prema tome, klasifikator određuje kojoj klasi pripada, odnosno koju bi oznaku imao novi neviđeni podatak. Važno je napomenuti kako ne postoji klasifikator koji je najbolji za sve skupove podataka (ovo je također poznato kao takozvani Nema-Besplatnog-Ručka teorem[4]). Preporučljivo je uspostaviti više klasifikatora te utvrditi koji najbolje rješava dani problem i koji najbolje odgovara određenom skupu podataka.

Prilikom klasifikacije skupa podataka može se javiti problem ukoliko jedna klasa ima značajno veći broj uzoraka od druge. U većini slučajeva klasa s manje uzoraka je ona od većeg interesa, jer upravo ta klasa otkriva anomalije i odstupanja od većinskih vrijednosti kao na primjer otkrivanje prijevare u bankovnim transakcijama ili otkrivanje određene bolesti kod pacijenta.

Krivo klasificiranje ovakvih podataka može uzorkovati određene posljedice ili gubitke. Ovo je poznato kao problem neuravnoteženosti klasa (engl. *class imbalance problem*) [5]. Klasa s više uzoraka naziva se većinska klasa (engl. *majority class*) dok se klasa s manje uzoraka naziva manjinska klasa (engl. *minority class*).

Postoje različiti načini kako se nositi s ovim problemom. Često su korišteni pristupi predobrade skupa podataka neovisni o odabranom klasifikatoru. Te pristupe predstavljaju preuzorkovanje (engl. *oversampling*) i poduzorkovanje (engl. *undersampling*) [5, 6]. Preuzorkovanje podrazumijeva stvaranje novih uzoraka koji pripadaju manjinskoj klasi dupliciranjem ili stvaranjem sintetičkih na temelju već postojećih podataka, dok poduzorkovanje predstavlja uklanjanje određenih uzoraka većinske klase. Obje metode imaju svoje prednosti i nedostatke i ne bi bilo valjano tvrditi da je jedna bolja od druge, no važna činjenica je ta da kod poduzorkovanja može doći do gubljenja bitnih uzoraka što je posebno izraženo u slučaju skupova podataka koji ispoljavaju visoku razinu neuravnoteženosti [6, 7].

2.1.1. Algoritam *k*-najbližih susjeda

Jedan od popularnijih i često učinkovitih algoritama klasifikacije je algoritam *k*-najbližih susjeda (engl. *k-nearest neighbours*, *k*-NN) [2, 3] zbog čega je i odabran u ovome radu. Jednostavnost algoritma leži u činjenici da ne treba imati nikakva saznanja o podatcima i njihovoj razdiobi. Glavna pretpostavka kojom se algoritam vodi je ta da se slični uzorci nalaze u bliskom okruženju. Drugim riječima, slični uzorci, odnosno oni koji imaju istu oznaku, nalaze se blizu jedni drugih.

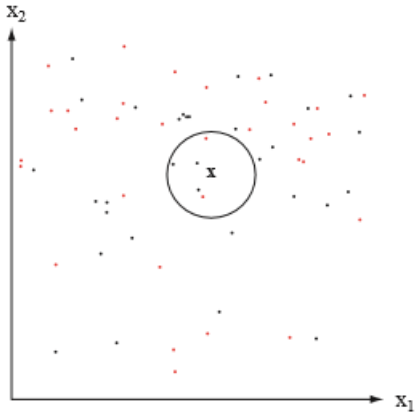
Funkcionalnost algoritma zasniva se na računanju Euklidske udaljenosti između ulaznog neviđenog uzorka i svakog uzorka iz podskupa podataka za trening. Neka je x_i ulazni uzorak s m značajki (x_{i1}, x_{i2},...,x_{im}). Tada je Euklidska udaljenost između testnog uzorka x_i i jednog trening uzorka x_l

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \cdots + (x_{im} - x_{lm})^2} \tag{2-1}$$

Osim računanja pomoću Euklidske udaljenosti, moguće je odrediti udaljenost koristeći druge funkcije poput *Manhattan* udaljenosti ili Chebyshevljeve udaljenosti.

Na ponašanje algoritma je moguće utjecati definiranjem veličine susjedstava koje se promatra za dani ulazni uzorak. Shodno tome, parametar k predstavlja broj koji govori koliko susjeda će se uzimati u obzir prilikom klasifikacije ulaznog uzorka. Jednostavnije rečeno, ulazni uzorak će pripasti onoj klasi koja je zastupljenija među k najbližih trening uzoraka. Dakako, potrebno je

pripaziti pri odabiru vrijednosti tog parametra. Kako bi se odabrao pravilan k za određen skup podataka, algoritam *k*-NN se testira s različitim vrijednostima k te se odabire ona vrijednost koja smanjuje broj pogrešaka uz održavanje sposobnosti precizne klasifikacije testnih uzoraka. Parametar k uobičajeno je neparan broj kako ne bi došlo do izjednačenog broja susjeda različitih klasa, ali odabir najprikladnije vrijednosti ovisi o samom skupu podataka i potrebno ga je odabrati sukladno ispitivanju učinkovitosti različitih vrijednosti. Često se u literaturi koriste vrijednosti
k = 1, 3 i 5 [8, 9].



Slika 2.1. Gledajući u dvodimenzionalnom prostoru uzoraka, testni uzorak se ubacuje u prostor kreirajući sferno područje oko sebe koje se širi sve dok ne obuhvati k trening uzoraka.
Izvor: [2, str 211.]

Prednost algoritma leži u njegovoj jednostavnosti prilikom čega nije potrebna izgradnja modela, podešavanje različitih parametara kao ni postavljanje pretpostavki o podatcima. Algoritam je univerzalan te se koristi u svrhe klasifikacije i regresije. Nedostatak je taj što postaje znatno sporiji povećavanjem skupa uzoraka ili parametra k. Pseudokod algoritma dan je slikom 2.2.

2.1.2. Vrednovanje učinkovitosti algoritama klasifikacije

Nakon odabira algoritma klasifikacije i dobivanjem rezultata u obliku kojoj klasi pripada pojedini uzorak, potrebno je ustanoviti koliko je odabrani algoritam efikasan za zadani problem, odnosno za zadani skup podataka. Različite mjere učinkovitosti se primjenjuju za različite algoritme strojnog učenja. Izbor mjera ima veliku važnost, jer uvelike utječu na mjerenje i usporedbu performansi samog algoritma. U ovom radu obrađena je matrica zbunjenosti i izračun različitih mjera ovisno o vrijednostima upisanima u nju [1, 6, 10].

Matrica zbunjenosti (engl. *confusion matrix*) [1] jedna je od jednostavnijih i razumljivijih mjera korištenih pri određivanju točnosti i preciznosti klasifikatora. U navedenim primjerima izlaz je ograničen na dvije moguće klase, kao što je već prethodno navedeno radi analize algoritama

ulaz: podskup trening uzoraka niz[], veličina podskupa trening uzoraka n, broj susjeda k, testni uzorak p

izlaz: klasa kojoj testni uzorak pripada

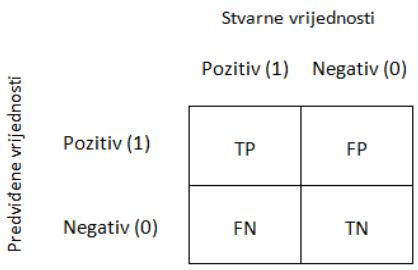
```
1. za svaki i = 1 do n čini
2.       izračunaj udaljenost i-tog i testnog uzorka, spremi kao vrijednost niz[i].udaljenost
3. sort(niz)                                        (-sortiranje uzlazno po udaljenosti-)
4. frekvencija1 = 0                                 (-frekvencija pojavljivanja uzoraka prve klase-)
5. frekvencija2 = 0                                 (-frekvencija pojavljivanja uzoraka druge klase-)
6. za svaki i = 1 do k čini
7.       ako je niz[i].vrijednost = 0
8.            povećaj frekvencija1 za 1
9.       inače
10.           povećaj frekvencija2 za 1
11. ako je frekvencija1 > frekvencija2
12.      vrati 1
13. inače
14.      vrati 2
15. kraj funkcije
```

Slika 2.2. Pseudokod algoritma *k*-NN.

binarne klasifikacije. Matrica zbunjenosti sama po sebi ne određuje učinkovitost klasifikatora, ali gotovo se sve mjere učinkovitosti zasnivaju na matrici i vrijednostima upisanim u njoj.

Matrica se sastoji od dva stupca i dva retka, gdje stupci predstavljaju stvarne vrijednosti, a retci predviđene vrijednosti uzoraka. Oblik matrice vidljiv je na slici 2.3.

Pozitiv (*Positive*(1)) predstavlja uzorke manjinske klase, dok negativ (*Negative*(0)) predstavlja uzorke većinske klase. Iz slike je vidljivo postajanje četiri kategorije:



Slika 2.3. Matrica zbunjenosti.

1. Istiniti pozitivi (engl. *True Positives, TP*) – broj uzoraka koji su pozitivi i ispravno klasificirani kao pozitivi
2. Istiniti negativi (engl. *True Negatives, TN*) - broj uzoraka koji su negativi i ispravno klasificirani kao negativi
3. Lažni pozitivi (engl. *False Positives, FP*) – broj uzoraka koji su negativi, ali su krivo klasificirani kao pozitivi
4. Lažni negativi (engl. *False Negatives, FN*) – broj uzoraka koji su pozitivi, ali su krivo klasificirani kao negativi

U idealnom slučaju, FP i FN bi bili jednaki 0, no to nije tako u stvarnoj primjeni, jer ni jedan algoritam klasifikacije (u većini slučajeva) nije sto posto precizan. Iz tog razloga, potrebno je odabrati koju je vrijednost važnije smanjiti, lažne pozitive (FP) ili lažne negative (FN). Odabir ovisi o problemu koji se rješava i temeljem toga se smanjuje jedna ili druga vrijednost. Primjerice, ukoliko algoritam otkriva rak, smanjenje lažnih negativa bila bi pametnija odluka, jer klasificiranje osobe koja ima rak negativnom dovodi do velikih posljedica. U drugom slučaju, primjerice prilikom detekcije elektroničke pošte kao neželjena ili ne, bila bi pogreška vrlo važnu poštu klasificirati kao neželjenu stoga je potrebnije napraviti određena podešavanja radi smanjenja lažnih pozitiva.

Ukupna točnost klasifikatora mjeri se pomoću točnosti klasifikacije (engl. *classification accuracy*) [1]. Točnost predstavlja omjer broja točno klasificiranih primjera i broja ukupnih klasifikacija. Formula temeljem matrice zbunjenosti je

$$\text{Točnost} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{2-2}$$

Točnost kao mjeru učinkovitosti dobro je koristiti kada su klase u skupu podataka donekle uravnotežene. No u slučaju neuravnoteženih podataka, na što se ovaj rad i fokusira, točnost se nikada ne bi trebala koristiti. U prethodno korištenom primjeru otkrivanja raka, gdje je skup podataka za treniranje neuravnotežen, 95 ljudi od 100 nema rak, dok njih 5 ima. Ukoliko se pretpostavi da je model loš i da svaki novi uzorak klasificira sa „nema rak“, 95 ljudi će točno klasificirati. Usprkos činjenici da je model beskoristan, njegova točnost bi bila 95%. Ova situacija je toliko česta da je dobila i vlastito ime, paradoks točnosti (engl. *accuracy paradox*) [11]. Izračun točnosti putem formule je ispravan, no neispravno je pretpostaviti kako točnost ispravno opisuje učinkovitost klasifikatora u svim situacijama, odnosno na svim problemima.

Preciznost (engl. *precision*) [1], dana formulom (2-3), je omjer broja točno klasificiranih pozitivnih primjera i ukupnog broja pozitivnih klasifikacija. Njome se dobiva informacija o uspješnosti klasifikatora u odnosu na vrijednost lažnih pozitiva, odnosno govori koliko uzoraka je točno klasificirano.

$$\text{Preciznost} = \frac{TP}{TP+FP} \tag{2-3}$$

Drugi naziv ove mjere je također i pozitivno predviđene vrijednosti (engl. *positive predictive values, PPV*). Suprotno tomu, moguće je izračunati omjer broja točno klasificiranih negativnih primjera i ukupnog broja negativnih klasifikacija (engl. *negative predictive values, NPV*) prema

$$\text{NPV} = \frac{TN}{TN+FN}. \tag{2-4}$$

Odziv ili osjetljivost (engl. *recall*) [1] mjera je koja govori koji udio stvarno pozitivnih vrijednosti je algoritam klasificirao kao pozitivne. Odziv daje informaciju o uspješnosti klasifikatora u odnosu na lažne negative, odnosno govori koliko je uzoraka klasifikator krivo klasificirao. Ova mjera je još poznata i pod nazivima stopa pogotka (engl. *hit rate*) i stopa istinitih pozitiva (engl. *true positive rate, TPR*) te se računa kao

$$\text{Odziv} = \frac{TP}{TP+FN}. \tag{2-5}$$

Kao što je ranije navedeno, ponekad je potrebno odabrati vrijednost koju je važnije svesti na minimalnu, lažne pozitive ili lažne negative. Preciznost i odziv utječu na variranje tih vrijednosti. Ukoliko se lažni negativi žele svesti na što manju moguću vrijednost, odziv bi trebao biti bliže 1 jer njegova važnost leži u detekciji pozitivnih primjera i pozitivnom klasificiranju istih. Ako je potrebno smanjiti lažne pozitive, tada bi fokus bio preciznost dovesti što bliže 1.

Mjera suprotna odzivu je specifičnost (engl. *specificity*) [1], odnosno mjera koja prikazuje koliko je negativnih primjera algoritam klasificirao kao negativne. Mjera je također poznata i pod nazivom stopa istinitih negativa (engl. *true negative rate, TNR*) te je dana formulom

$$\text{Specifičnost} = \frac{TN}{TN+FN}. \tag{2-6}$$

Kako ne bi bilo potrebno računati preciznost i odziv za svaki klasifikacijski problem, moguće je obje formule objediniti u jednu i njome predstaviti i preciznost i odziv. Jedna od tih formula je jednostavno uzimanje aritmetičke sredine, odnosno zbroj preciznosti i odziva, i dijeljenje dobivene vrijednosti s dva. U slučaju neuravnoteženih klasa nije primjereno uzeti ovu mjeru zbog pristranosti klasifikatora većinskoj klasi. Potrebno je uvesti mjeru koja uvodi ravnotežu, a to je harmonijska sredina. Harmonijska sredina je otprilike jednaka aritmetičkoj kada su preciznost i odziv jednaki. No, u slučaju neuravnoteženih klasa oni su drastično različiti i tu je harmonijska sredina bliža manjem broju. Harmonijska sredina preciznosti i odziva još se i naziva F-mjera [1], čija je formula

$$\text{F} = \frac{2*preciznost*odziv}{preciznost+odziv}. \tag{2-7}$$

U općem slučaju je

$$\text{F}_\beta = \frac{(1+\beta^2)*preciznost*odziv}{\beta^2*preciznost+odziv}, \tag{2-8}$$

gdje se parametrom β upravlja s doprinosom preciznosti i odziva. Podjednaka važnost računa se pri β = 1, dok se pri β = 2 pridaje dvostruko veća važnost odzivu. Pri β = 0,5 preciznost je dvostruko veće važnosti od odziva. Mijenjanje parametra β utječe na ranije spomenuto smanjivanje jedne od dvije vrijednosti na minimalnu, lažne pozitive ili lažne negative.

2.2. Algoritam SMOTE

Problem neuravnoteženosti klasa treba shvatiti ozbiljno, jer može algoritam klasifikacije učiniti potpuno beskorisnim i krivo interpretirati nove podatke. Kao što je već spomenuto, potrebno je napraviti predobradu podataka kako bi se uzorci većinske i manjinske klase uravnotežili i bili približno jednako brojni. To se može postići preuzorkovanjem različitim algoritmima koji su predloženi u literaturi [5].

Najjednostavnija metoda preuzorkovanja je nasumično preuzorkovanje (engl. *random oversampling, RO*) [5]. Uzorci manjinske klase se na slučajan način odabiru i dupliciraju u skupu trening podataka. Postupak se ponavlja sve dok se ne ostvari željena brojnost podataka. Nasumično preuzorkovanje je također poznato pod nazivom naivno uzorkovanje, jer nije potrebno postavljanje pretpostavki o podatcima. Radi toga je algoritam brz i jednostavan za implementiranje. Međutim, nasumično uzorkovanje podrazumijeva duplikaciju već postojećih uzoraka manjinske klase što može otežati klasifikaciju algoritmu u slučajevima preklapanja podataka, gdje uzorci različitih klasa imaju slične karakteristike.

U [12] predložena je alternativa standardnom nasumičnom preuzorkovanju. Umjesto dupliciranja već postojećih manjinskih uzoraka, predstavljen je način kojim se stvaraju novi, sintetički uzorci manjinske klase. Algoritam je dobio ime tehnika sintetičkog preuzorkovanja manjinskih uzoraka. Osnovna zadaća algoritma je vršenje interpolacije među susjednim elementima uzoraka, odnosno na temelju poznatih uzoraka stvoriti nove u njihovom susjedstvu kako bi se u konačnici poboljšala generalizacija samog klasifikatora koji će se trenirati na tim podacima. Algoritam se ispostavio vrlo učinkovitim i primjenjivim u velikom broju područja radi čega se danas smatra jednim od najutjecajnijih algoritama za predobradu podataka u području strojnog učenja i rudarenja podataka [9, 12].

Formalno, procedura algoritma je sljedeća. Prvo se postavlja cjelobrojna vrijednost N. Nadalje, vrši se iterativni proces koji se sastoji od nekoliko koraka. Prvo, redom se uzimaju svi uzorci iz podskupa podataka za treniranje. Zatim se za svaki pojedini uzorak uzima u obzir njegovih k najbližih susjeda, gdje je parametar k po zadanoj vrijednosti jednak 5 [12]. Naposlijetku, $\frac{N}{100}$ puta se nasumično odabere susjed i stvara se sintetički uzorak kao konveksna kombinacija odabranog uzorka za preuzorkovanje i njegovog odabranog susjeda. Novi uzorci generiraju se na dužini koja povezuje početni uzorak i njegov k-ti susjed. Stvaranje $\frac{N}{100}$ sintetičkih uzoraka prema [12, 13] opisano je sa

$$s^i := x + U_i(0,1)\cdot(x^{r(i)} - x) , \quad i = 1,...,\frac{N}{100} , \tag{2-9}$$

gdje je $U_i(0,1)$ uniformna slučajna varijabla u intervalu (0,1), x je uzorak manjinske klase dok je $x^{r(i)}$ njegov odabrani susjed. Važno je napomenuti kako se vrijednost N ($N \geq 100$) zadaje u postotcima. Taj broj se dijeli sa 100 kako bi se odredilo koliko susjeda od k mogućih se uzima u obzir, odnosno koliko se novih sintetičkih uzoraka stvara, po jedan u smjeru svakog odabranog susjeda. Primjerice, vidljivo je kako će se pri N = 100% broj uzoraka udvostručiti, a pri N = 200% utrostručiti i tako dalje.

Algoritam, čiji je pseudokod prikazan na slici 2.4, ima tri parametra, T, N i k, čije vrijednosti znatno utječu na stvaranje sintetičkih uzoraka [14], te shodno tome na uspješnost i performanse samog klasifikatora kroz njegovo treniranje na tim podacima.

2.2.1. Prednosti i nedostatci

Metoda preuzorkovanja stvaranjem sintetičkih uzoraka nudi povećanje ravnoteže podskupa podataka za trening bez zanemarivanja korisnih uzoraka većinske klase. Nadalje, suprotno nasumičnom preuzorkovanju, algoritam ne duplicira postojeće uzorke već stvara nove temeljene na uzorcima manjinske klase. Ovime se smanjuje rizik stvaranja modela koji odgovara jedinstvenom skupu podataka te se efektivno povećava prostor manjinskih uzoraka što može dovesti do veće razine generalizacije klasifikatora [12].

Međutim, pojavljuju se dva problema prilikom sintetičkog preuzorkovanja [15, 16]. Prvi je taj da SMOTE u nekim slučajevima generira uzorke unutar većinske klase, dok drugi problem predstavlja generiranje uzoraka u smjeru najbližih uzoraka manjinske klase ne uzimajući u obzir odnos s bliskim uzorcima većinske klase. Manjinski uzorci u vrlo malom broju dovode do njihove raspršenosti u prostoru podataka što uzrokuje veće pogreške pri interpolaciji sintetičkih uzoraka između njih.

2.2.2. Unaprijeđene i proširene varijante

Kako bi se prethodno navedeni problemi ublažili, predložene su, primjerice, izmjene algoritma koje brišu uzorke manjinske klase generirane u prostoru većinske, ali su također i predložene izmjene koje nove sintetičke uzorke smještaju bliže prostoru većinskih uzoraka [16].

U svrhu ostvarivanja veće učinkovitosti, većina klasifikacijskih algoritama pokušava što bolje odrediti granicu razdvajanja različitih klasa. Uzorci na granici (engl. *borderline*) i u njenoj

neposrednoj blizini skloniji su krivoj klasifikaciji nego oni uzorci koji su daleko od granice te su time važniji za sam postupak klasifikacije.

U [17] predložena su dva algoritma koja se bave problemom graničnih uzoraka, Borderline-SMOTE1 i Borderline-SMOTE2. Važnost algoritama leži u pronalasku graničnih uzoraka čija je

ulaz: broj uzoraka manjinske klase T; postotak preuzorkovanja N; broj najbližih susjeda k

izlaz: $\frac{N}{100}$ * T sintetičkih uzoraka manjinske klase

1. N = (int)(N/100)
2. k = broj najbližih susjeda
3. br_atr = broj atributa
4. orig_niz[][] = niz uzoraka manjinske klase
5. novi_index = 0 (prati broj generiranih sintetičkih uzoraka)
6. sint_niz[][] = niz sintetičkih uzoraka
7. za svaki i = 1 do T čini
8. odredi k najbližih susjeda za i te spremi njihove indekse u ns_niz
9. generiraj_sin_uzorak (N, i, ns_niz)
10. kraj petlje

generiraj_sin_uzorak (N, i, ns_niz) (-funkcija koja generira sintetičke uzorke-)

11. sve dok je N≠0 čini
12. nn = nasumični broj od 1 do k (-odabir jednog od k najbližih susjeda)
13. za svaki atribut = 1 do m čini
14. razlika = orig_niz [ns_niz[nn]] [atribut] – orig_niz [i] [atribut]
15. raskorak = nasumični broj između 0 i 1
16. sint_niz [novi_index] [atribut] = orig_niz [i] [atribut] + raskorak*razlika
17. kraj petlje
18. povećaj novi_index za 1
19. smanji N za 1
20. kraj petlje
21. kraj funkcije

Slika 2.4. Pseudokod algoritma SMOTE.

vjerojatnost krive klasifikacije velika. Algoritmom Borderline-SMOTE1 svakom uzorku manjinske klase određuje se m najbližih susjeda iz cijelog skupa podataka. Broj uzoraka većinske klase od m najbližih susjeda se definira kao vrijednost varijable m'. Ukoliko vrijedi da je m = m', tada svi susjedi pripadaju većinskoj klasi te se taj manjinski uzorak smatra šumom, no ukoliko je $\frac{m}{2} < m' \leq m$, manjinski uzorak može lako biti krivo klasificiran te ga se stavlja u skup „opasnih“ podataka. Ukoliko vrijedi $m \leq m' < \frac{m}{2}$ uzorak nije granični te se smatra „sigurnim“ za klasifikaciju. Nadalje, na „opasnom“ skupu uzoraka primjenjuje se osnovni SMOTE algoritam kojim se za svaki pojedini uzorak radi od 1 do k sintetičkih uzoraka.

Borderline-SMOTE2 ne samo da generira sintetičke uzorke od svakog uzorka iz „opasnog“ skupa do svakog pojedinog pozitivnog uzorka, nego to isto čini i između najbližeg susjeda većinske klase. Razlika između uzorka iz „opasnog“ skupa i najbližeg susjeda manjinske klase množi se s nasumičnim brojem između 0 i 0,5 što rezultira novo generiranim uzorcima smještanje bliže manjinskoj klasi.

Algoritam Safe-Level-SMOTE [18] također uzorke dijeli na „sigurne“ i „nesigurne“. Algoritam svakom pozitivnom uzorku dodjeljuje razinu sigurnosti (engl. *safe level*) prije generiranja sintetičkih uzoraka. Ukoliko je razina sigurnosti odabranog uzorka blizu 0, uzorak se smatra šumom dok se uzorak razine sigurnosti blizu k smatra „sigurnim“. Svaki novonastali sintetički uzorak smješten je što bliže najvećem području „sigurne“ razine čime se ostvaruje da su svi novi uzorci generirani samo u „sigurnim“ regijama. Razina sigurnosti dana je kao

$$\text{razina sigurnosti (rs)} = \text{broj pozitivnih uzoraka od k najbližih susjeda.} \tag{2-10}$$

Računanjem razine sigurnosti odabranog pozitivnog uzorka i njegovog susjeda, koji je također pozitivan uzorak, određuje se omjer razine sigurnosti dan jednadžnom (2-11) kojime se novi, sintetički uzorak smješta bliže onom uzorku koji ima veću razinu sigurnosti.

$$\text{omjer razine sigurnosti} = \frac{razina\ sigurnosti\ pozitivnog\ uzorka}{razina\ sigurnosti\ najbližeg\ pozitivnog\ susjeda} \tag{2-11}$$

Nadalje, u [19] predloženo je i poboljšanje algoritma nazvano Random-SMOTE čiji fokus leži u

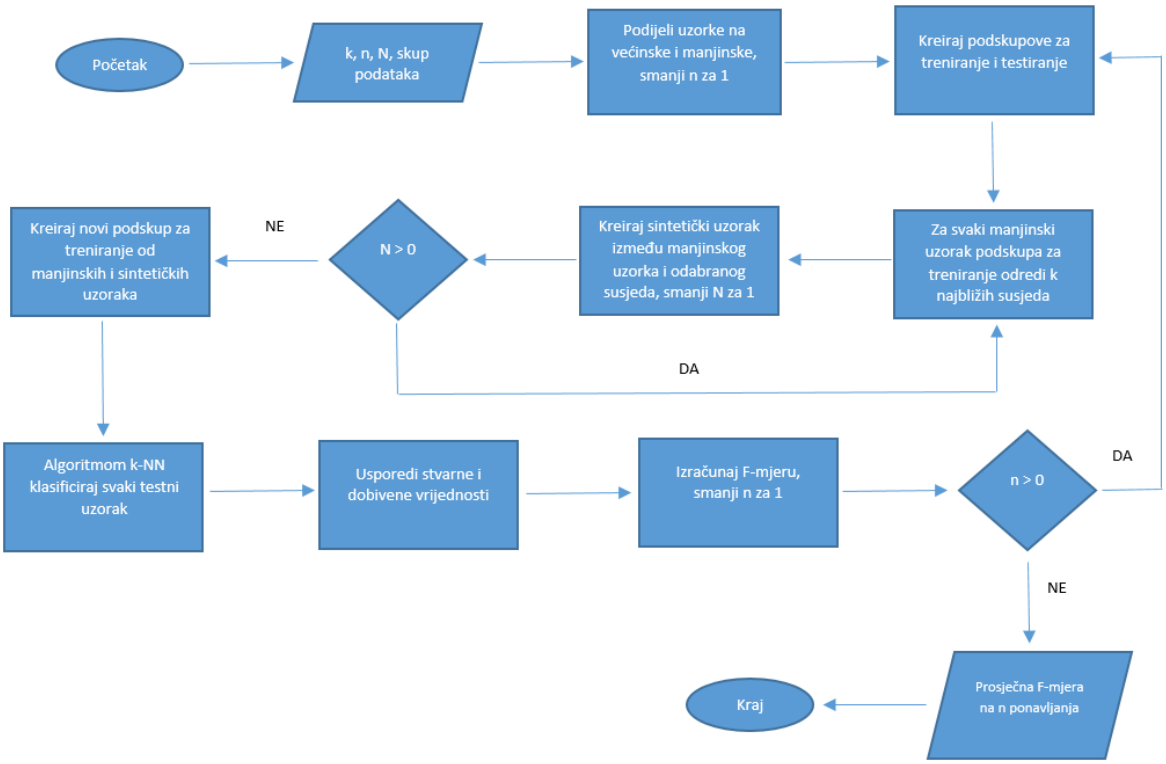
samom stvaranju sintetičkih uzoraka. Umjesto korištenja samo jednog susjeda, uzimaju se dva nasumična susjeda kako bi se kreiralo $\frac{N}{100}$ sintetičkih uzoraka svakog manjinskog uzorka unutar trokuta definiranog s tri podatkovne točke. Ovime se novi uzorci stvaraju u rjeđim područjima manjinske klase čime se uvodi ravnoteža raspršenosti uzoraka.

3. OSTVARENO PROGRAMSKO RJEŠENJE

Programsko rješenje ostvareno je u .NET okruženju koristeći C# programski jezik. Ono omogućuje učitavanje skupa podataka u CSV (engl. *comma separated values*) formatu nad kojim se vrši preuzorkovanje jednim od četiri dostupnih algoritama preuzorkovanja. Temeljem ispitivanja uspješnosti odabranog algoritma, računa se izlaz koji daje informaciju o njegovoj učinkovitosti.

3.1. Način rada programskog rješenja

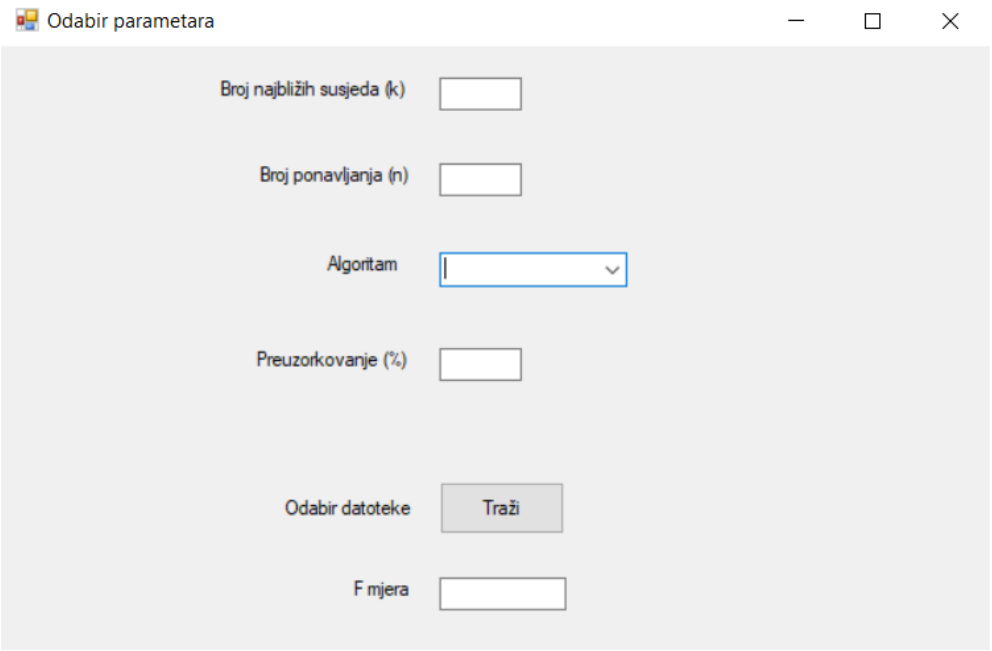
Ulaz programskog rješenja predstavljaju postavke poput broja ponavljanja izvođenja odabranog algoritma, broj najbližih susjeda k, postotak preuzorkovanja te datoteka u CSV formatu u kojoj se nalaze podatci. Korištenjem apstrakcije, moguća je izmjena koda kako bi se promijenio format ulaznog skupa podataka. Ulazni podatci se nadalje dijele na manjinske i većinske uzorke te se od njih formira podskup podataka za treniranje i za testiranje. Manjinski uzorci podskupa za treniranje preuzorkuju se jednim od četiri algoritama preuzorkovanja: nasumično preuzorkovanje, SMOTE, Borderline-SMOTE ili Safe-level-SMOTE. Nadalje, temeljem preuzorkovanih manjinskih uzoraka i početno zadanih većinskih uzoraka podskupa za treniranje, stvara se konačni podskup podataka za treniranje. Sljedećim korakom podatcima, koji se nalaze u podskupu za testiranje, određuju se oznake, odnosno uzorci se klasificiraju pomoću algoritma *k*-NN. Svaki testni uzorak bit će klasificiran onom klasom koja je zastupljenija u već spomenutom susjedstvu. Skup vrijednosti koje predstavljaju klasifikacije testnih uzoraka uspoređuju se sa stvarnim vrijednostima pripadnosti uzorka određenoj klasi. Temeljem usporedbe računaju se vrijednosti TP, TN, FP, FN koje su opisane u potpoglavlju 2.1.2. Njima se određuje izlaz u obliku prosječne F-mjere. U budućnosti bilo bi jednostavno postaviti neku drugu mjeru učinkovitosti za izlaz jer se sve računaju pomoću prethodno navedene četiri vrijednosti matrice zbunjenosti. Slikom 3.1. prikazan je dijagram toka izvođenja cjelokupnog programskog rješenja od ulaza do dobivenog izlaza.



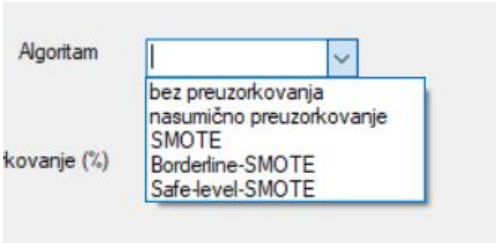
Slika 3.1. Dijagram toka izvođenja cjelokupnog programskog rješenja.

3.2. Prikaz i način uporabe programskog rješenja

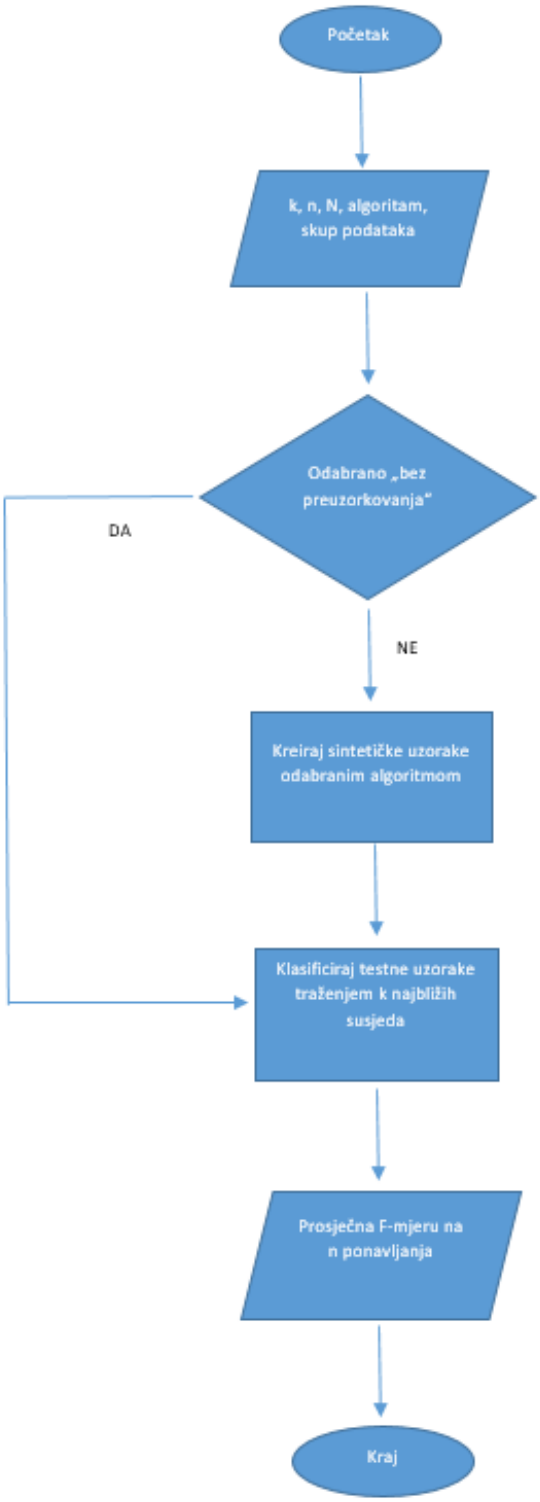
Pokretanjem programa otvara se grafičko sučelje koje omogućuje postavljanje parametara eksperimenta. Temeljem unesenih vrijednosti, odabranog algoritma i odabrane datoteke CSV formata, izlaz se računa kao prosjek F-mjere nad odabranim podatcima. Prikaz početnog grafičkog sučelja dan je slikom 3.2., mogućnost odabira algoritma prikazan je na slici 3.3, dok je dijagram toka samog programskog rješenja dan slikom 3.4.



Slika 3.2. Grafičko sučelje programa.



Slika 3.3. Mogućnosti pri odabiru algoritma preuzorkovanja.



Slika 3.4. Dijagram toga programskog rješenja.

4. EKSPERIMENTALNA ANALIZA

Glavni zadatak ovoga rada je usporedba različitih algoritama preuzorkovanja zasnovanih na SMOTE i prikaz njihovog utjecaja na rješavanje problema neuravnoteženosti klasa. Odabrani algoritmi dani su u tablici 4.1. Kao algoritmu SMOTE i njegovim inačicama, također je u analizu uključen algoritam nasumičnog preuzorkovanja te slučaj kada se ne primjenjuje preuzorkovanje koji služe kao referentne točke u usporedbi.

Kako bi se dao uvid u različitosti izvedbi i rezultata različitih algoritama, eksperimentalna analiza provedena je na nekoliko neuravnoteženih skupova podataka. Skupovi su preuzeti s Keel [20] i UCl [21] repozitorija za strojno učenje te predstavljaju stvarne podatke. Karakteristike odabranih skupova podataka prikazani su u tablici 4.2.

Klasifikator korišten u ovom eksperimentu je klasifikator *k*-najbližih susjeda [2, 3], odabran zbog svoje jednostavnosti te čestog korištenja u sličnim eksperimentima. Međutim, mogli su biti korišteni i drugi klasifikatori, poput stabla odluke (engl. *decision trees*) ili umjetnih neuronskih mreža (engl. *artificial neural networks*). Uz algoritam *k*-NN, ovi klasifikatori su često korišteni u literaturi prilikom usporedbe algoritama preuzorkovanja [7, 14, 17, 19].

Tablica 4.1. Korišteni algoritmi i njihove oznake.

Ime algoritma	Oznaka algoritma
Nasumično preuzorkovanje	RO
SMOTE	SMOTE
Borderline-SMOTE	BSMOTE
Safe-level-SMOTE	SLSMOTE

Tablica 4.2. Karakteristike korištenih skupova.

Ime skupa podataka	Oznaka	Izvor	#uzoraka	#značajki	IR
Ionosphere	D1	UCI	351	34	1,79
Parkinsons	D2	UCI	195	22	3,06
Yeast2v4	D3	Keel	514	8	9,08
Climate	D4	UCI	540	18	10,74
Shuttle6v23	D5	Keel	230	9	22
Poker9v7	D6	Keel	244	10	29,5

4.1. Postavke eksperimenta

Skupovi podataka podijeljeni su prema standardnom 0,8 : 0,2 omjeru na podskup za treniranje i podskup za testiranje. Istim omjerom je održan broj većinskih i manjinskih uzoraka unutar svakog podskupa. Svaki algoritam zasebno je testiran na pojedinom skupu podataka. Ovo je izvršeno 30 puta nad svim skupovima podataka, gdje je prilikom svakog novog ponavljanja pojedini skup podijeljen na nove podskupove za treniranje i testiranje uz održavanje istog omjera podjele. Svaki skup je predobrađen u svrhu smanjivanja utjecaja različitih raspona vrijednosti značajki. Predobrada je izvršena postupkom normalizacije prilikom kojeg svaka značajka poprima vrijednost unutar intervala [0, 1]. Veličina susjedstva predstavljena brojem k postavljena je na jednu od često korištenih vrijednosti, k = 1, 3 ili 5 [8, 9]. Ova vrijednost se zadaje algoritmu SMOTE unutar kojeg se prosljeđuje algoritmu *k*-NN. Količina preuzorkovanja u postotcima (N) u svim ponavljanjima postavljena je na vrijednost koja ostvaruje približnu ravnotežu skupa podataka. Ova vrijednost izračunala se prilikom izvođenja programa te se zasniva na omjeru broja većinskih i broja manjinskih uzoraka pojedinog skupa.

4.2. Rezultati

Kako bi se odredio najučinkovitiji pristup preuzorkovanju, eksperimentalnom analizom utvrđene su prosječne vrijednosti F-mjere svih kombinacija algoritama i skupova podataka. Prosječne vrijednosti F-mjere eksperimenta za vrijednost k = 5 prikazani su tablicom 4.3 dok je tablicom 4.4 prikazana minimalna i maksimalna ostvarena vrijednost F-mjere od 30 ponavljanja svakog algoritma nad svakim skupom podataka. Uz prosječne vrijednosti F-mjere također je prikazana i standardna devijacija. Važno je napomenuti kako oznaka NO predstavlja slučaj kada nije primijenjeno preuzorkovanje (engl. *no oversampling*).

Tablica 4.3. Prosječna F-mjera eksperimenta uz k = 5.

	NO	RO	SMOTE	BSMOTE	SLSMOTE
D1	0,735 ± 0,063	0,787 ± 0,049	**0,818** ± 0,071	0,808 ± 0,059	0,782 ± 0,062
D2	0,795 ± 0,109	**0,804** ± 0,098	0,753 ± 0,111	0,796 ± 0,072	0,803 ± 0,072
D3	**0,756** ± 0,106	0,742 ± 0,062	0,729 ± 0,094	0,717 ± 0,081	0,732 ± 0,068
D4	0,391 ± 0,153	0,405 ± 0,073	0,342 ± 0,027	0,377 ± 0,074	**0,431** ± 0,087
D5	0,919 ± 0,166	0,944 ± 0,133	**1** ± 0	0,956 ± 0,113	0,933 ± 0,133
D6	0,667 ± 0	0,824 ± 0,160	0,569 ± 0,148	**0,880** ± 0,163	0,868 ± 0,161

Tablica 4.4. Minimalne i maksimalne ostvarene vrijednosti F-mjere uz k = 5.

	NO	RO	SMOTE	BSMOTE	SLSMOTE
D1	0,595 - 0,833	0,667 - 0,880	0,585 - 0,939	0,651 - 0,923	0,698 - 0,920
D2	0,428 - 0,952	0,571 - 1	0,428 - 0,909	0,608 - 0,909	0,571 - 0,952
D3	0,5 - 0,952	0,56 - 0,833	0,526 - 0,909	0,556 - 0,909	0,570 - 0,869
D4	0,14 - 0,571	0,285 - 0,538	0,2667 - 0,4	0,25 - 0,551	0,2 - 0,615
D5	0,667 - 1	0,667 - 1	1 - 1	0,667 - 1	0,667 - 1
D6	0,667 - 0,667	0,4 - 1	0,333 - 1	0,4 - 1	0,5 - 1

Podebljano označene vrijednosti predstavljaju najbolji rezultat koji je postignut za odgovarajući skup podataka. Dani rezultati dobiveni su postavljanjem postavki algoritama kako bi se ostvarila ravnoteža većinskih i manjinskih uzoraka. Analiza nije provedena na istim podskupovima podataka za treniranje i testiranje, no osigurano je 30 ponavljanja svake kombinacija skupa i algoritma preuzorkovanja uz održavanje istog omjera podjele i istih parametara eksperimenta. Vidljivo je da je u većini slučajeva održana konzistentnost rezultata neovisno o algoritmu. Slučaj koji odskače je primjena algoritma SMOTE na skup D6, gdje su ostvareni lošiji rezultati u odnosu na druge algoritme preuzorkovanja. Moguće objašnjenje je prethodno navedeno odabiranje postavki algoritma kojima se postiže ravnoteža, gdje prilikom velike odabrane vrijednosti N može doći do prezasićenosti pojedinih regija prostora podataka. Suprotno tomu, vidljivo je kako je SMOTE jedini postigao najveći mogući rezultat za skup D5. Vidljivo je kako nijedan algoritam nije superioran, odnosno nijedan algoritam nije ostvario najveće vrijednosti prosječne F-mjere za sve skupove podataka. Algoritam SMOTE je, gledajući u odnosu na NO i RO, na dva skupa (D1, D5) ostvario bolje rezultate, dok je na ostale skupove (D2, D3, D4, D6) ostvario lošije. Algoritmi BSMOTE i SLSMOTE većinom ostvaruju bolje rezultate u odnosu na NO i RO, dok su u odnosu na SMOTE bolji izbor za skupove D2, D4 i D6. Razlika među algoritmima po pojedinom skupu nije znatna osim u slučaju skupa D6, u kojem SMOTE i izostanak preuzorkovanja (NO) ostvaruju znatno lošije rezultate. Najveći učinak algoritma BSMOTE vidljiv je u skupu D6, gdje ostvaruje znatno bolje rezultate u odnosu na NO i SMOTE. Time se može zaključiti kako u skupu ima velik broj podataka koji se lako krivo klasificiraju, odnosno graničnih podataka i onih podataka koji se nalaze izvan prostora podataka klase, a upravo te podatke preuzorkuju BSMOTE i SLSMOTE čime ostvaruju dobre rezultate klasifikacije. Izostanak preuzorkovanja (NO) pokazuje lošije rezultate u odnosu na RO, što je i očekivano zbog neuravnoteženosti podataka te je očitije pri većim omjerima neuravnoteženosti (IR).

Promatrajući tablicu 4.4 uočljivo je kako je NO za skup D6 uvijek dobio isti rezultat F-mjere bez obzira na podjelu, što je također i vidljivo iz tablice 4.3 gledajući standardnu devijaciju. Svi pristupi na skupu D5 su u barem jednom ponavljanju ostvarili F-mjeru jednaku 1, dok ju je jedino SMOTE ostvario u svakom od ponavljanja. Također je NO za skup D6 ostvario u svakom ponavljanju istu vrijednost F-mjere. Za skup D5 su za sve pristupe ostvarene iste minimalne i maksimalne vrijednosti osim za SMOTE koji ima obje vrijednosti jednake. Najveći raspon podataka uočljiv je pri primjeni SMOTE algoritma nad skupom D6.

Tablicom 4.5 dan je prosjek F-mjere eksperimenta uz uporabu vrijednost k = 3. Minimalne i maksimalne vrijednosti F-mjere od 30 ponavljanja svakog algoritma nad svakim skupom podataka prikazane su tablicom 4.6. Proučavajući rezultate moguće je primijetiti kako su ostvareni bolji rezultati klasifikacije osim u slučaju skupova D4 i D5. Ovo se može objasniti mogućnošću da su pojedini uzorci smješteni van prostora podataka određene klase i stoga krivo klasificirani. U većini slučajeva, razlika između vrijednosti klasifikacije uz smanjenje parametra k je manja od 5%, dok je najveća razlika zabilježila približno 8%. Svi pristupi nad skupovima D2, D5 i D6, osim primjene SMOTE na skup D2, ostvarili su F-mjeru jednaku 1 u barem jednom od svih ponavljanja. Uspoređujući s tablicom 4.4, niti jedan pristup nije ostvario maksimalnu vrijednost F-mjere za sva ponavljanja. Svi algoritmi su nad skupom D5 ostvarili iste minimalne i maksimalne vrijednosti.

Tablica 4.7 prikazuje prosječnu F-mjeru uz korištenje vrijednosti k = 1 dok su tablicom 4.8 prikazane minimalne i maksimalne vrijednosti F-mjere od 30 ponavljanja svakog algoritma nad svakim skupom podataka. Algoritam SMOTE je , u odnosu na njegovu primjenu za k = 5 i 3, za sve skupove osim D3 ostvario bolje rezultate, te je opet, kao i uz korištenje vrijednosti k = 5, postigao maksimalnu F-mjeru za skup D5. Algoritmi BSMOTE i SLSMOTE nisu pokazali značajne promjene u odnosu na rezultate ostvarene uz uporabu k = 1 ili 3 te su, uspoređujući s primjenom SMOTE algoritma, za skupove D1 i D5 ostvarili lošije, a za ostale skupove bolje rezultate. Najveća promjena vidljiva je u slučaju izostanka preuzorkovanja za skup D6, gdje se F-mjera povećala za nešto manje od 30% u odnosu na vrijednost pri uporabi parametra k =5 ili 3. Razlog tomu je korištenje jednake vrijednosti parametra k za *k*-NN i SMOTE pri čemu je 1-NN bolji od ostalih klasifikatora za taj skup podataka. Također, tako veliku promjenu podupiru i rezultati primjene RO te SMOTE koji donekle ravnomjerno preuzorkuju manjinske uzorke u odnosu na razmatrane dvije unaprijeđene inačice (BSMOTE i SLSMOTE).

U slučaju korištenja k = 1, baš kao i pri vrijednosti k = 3, skupovi D2, D5 i D6 većinom su ostvarili u barem jednom ponavljanju vrijednost F-mjere jednaku 1, pri čemu je, isto kao i za k = 5, SMOTE u svakom ponavljanju nad skupom D5 ostvario maksimalnu vrijednost.

Uspoređujući rezultate tablica 4.4, 4.6 i 4.8, vidljivo je kako su skupovi D5 i D6, neovisno o podjeli podataka, ostvarili maksimalnu vrijednost F-mjere u barem jednom od ponavljanja. U većini slučajeva radi se o značajnijim odstupanjima dobivenih vrijednost F-mjere što može biti posljedica podijele skupova radi čega je važno imati više ponavljanja kako bi se dobio uvid u performanse i ponašanje pojedinih pristupa. Također, vidljivo je kako nema značajnih promjena minimalnih i maksimalnih vrijednosti uz različite vrijednosti parametra k pri čemu se može zaključiti kako nema značajan utjecaj na raspon F-mjere.

Tablica 4.5. Prosječna F-mjera uz k = 3.

	NO	RO	SMOTE	BSMOTE	SLSMOTE
D1	0,747 ± 0,048	0,814 ± 0,053	**0,824 ± 0,052**	0,812 ± 0,056	0,808 ± 0,063
D2	0,842 ± 0,144	**0,882 ± 0,068**	0,802 ± 0,089	0,842 ± 0,082	0,860 ± 0,061
D3	**0,798 ± 0,085**	0,750 ± 0,084	0,727 ± 0,063	0,750 ± 0,082	0,735 ± 0,088
D4	0,354 ± 0,156	**0,372 ± 0,104**	0,346 ± 0,033	0,353 ± 0,096	0,352 ± 0,095
D5	0,877 ± 0,155	0,922 ± 0,147	**0,989 ± 0,059**	0,967 ± 0,113	0,933 ± 0,140
D6	0,667 ± 0,167	0,891 ± 0,160	0,649 ± 0,134	0,894 ± 0,185	**0,922 ± 0,142**

Tablica 4.6. Minimalne i maksimalne ostvarene vrijednosti F-mjere uz k = 3.

	NO	RO	SMOTE	BSMOTE	SLSMOTE
D1	0,632 - 0,844	0,667 - 0,880	0,698 - 0,894	0,634 - 0,898	0,680 - 0,938
D2	0,375 - 1	0,64 - 1	0,5 - 0,909	0,667 - 1	0,714 - 1
D3	0,526 - 0,857	0,455 - 0,857	0,608 - 0,869	0,5 - 0,869	0,583 - 0,956
D4	0,153 - 0,7	0,09 - 0,538	0,28 - 0,425	0,222 - 0,583	0,095 - 0,538
D5	0,667 - 1	0,667 - 1	0,667 - 1	0,667 - 1	0,667 - 1
D6	0,667 - 1	0,5 - 1	0,4 - 1	0,5 - 1	0,5 - 1

Tablica 4.7. Prosječna F-mjera uz k = 1.

	NO	RO	SMOTE	BSMOTE	SLSMOTE
D1	0,777 ± 0,067	0,794 ± 0,050	**0,832 ± 0,054**	0,788 ± 0,052	0,793 ± 0,052
D2	0,854 ± 0,075	**0,887 ± 0,075**	0,837 ± 0,072	0,884 ± 0,065	0,884 ± 0,067
D3	0,753 ± 0,122	**0,764 ± 0,094**	0,705 ± 0,082	0,729 ± 0,081	0,727 ± 0,073
D4	0,286 ± 0,114	0,303 ± 0,120	**0,353 ± 0,064**	0,304 ± 0,101	0,283 ± 0,089
D5	0,911 ± 0,140	0,978 ± 0,083	**1 ± 0**	0,978 ± 0,099	0,933 ± 0,124
D6	0,910 ± 0,140	**0,932 ± 0,122**	0,695 ± 0,200	0,878 ± 0,147	0,916 ± 0,154

Tablica 4.8. Minimalne i maksimalne ostvarene vrijednosti F-mjere uz k = 1.

	NO	RO	SMOTE	BSMOTE	SLSMOTE
D1	0,578 - 0,851	0,697 - 0,893	0,681 - 0,901	0,7 - 0,897	0,681 - 0,901
D2	0,667 - 1	0,667 - 1	0,631 - 0,952	0,705 - 1	0,736 - 1
D3	0,333 - 0,909	0,526 - 0,9	0,571 - 0,956	0,631 - 0,952	0,6 - 0,857
D4	0,117 - 0,526	0,117 - 0,556	0,238 - 0,487	0,153 - 0,5	0,105 - 0,444
D5	0,667 - 1	0,667 - 1	1 - 1	0,667 - 1	0,667 - 1
D6	0,667 - 1	0,5 - 1	0,25 - 1	0,667 - 1	0,5 - 1

5. ZAKLJUČAK

Važnost primjene SMOTE algoritma kao metode preuzorkovanja jasno je vidljiva u literaturi u kojoj je čest odabir prilikom rješavanja problema neuravnoteženosti klasa. Usprkos efikasnoj primjeni u različitim područjima, način odabira uzoraka i tehnike interpolacije ne dovode do željenih rezultata. Zadatak ovoga rada bio je usporediti SMOTE i algoritme zasnovane na njemu prilikom rješavanja problema neuravnoteženosti klasa.

Eksperimentalnom analizom primijenjeni su različiti algoritmi preuzorkovanja nad često korištenim stvarnim skupovima podataka. Iz prikazanih rezultata vidljivo je kako niti jedan pristup nije pokazao superiornu mjeru učinkovitosti uspoređujući ga s ostalim, sličnim algoritmima. Međutim, iz usporedbe je moguće donijeti zaključke o pojedinim algoritmima i njihovoj efikasnosti. Analizom je utvrđeno kako jedno ponavljanje primjene algoritma nad skupom podataka ne daje stvarni prikaz njegove učinkovitosti. Potrebno je provesti više ponavljanja s različitim podskupovima kako bi se dobio precizniji uvid. Također, rezultati pokazuju kako odabir uzoraka koji se lako krivo klasificiraju ima veći utjecaj pri većim omjerima neuravnoteženosti, prilikom čega su Borderline-SMOTE i Safe-level-SMOTE ostvarili bolje rezultate. Gledajući ostvarenu vrijednost F-mjere, jedino je SMOTE u 2 slučaja ostvario najveću moguću. Nadalje, mijenjanje parametra k nije imalo velik utjecaj na rezultate, iako prilikom smanjenja k na vrijednost 1, Borderline-SMOTE i Safe-level-SMOTE ostvarili su lošije rezultate u usporedbi s ostalim algoritmima preuzorkovanja.

U daljnjem radu moguće je u usporedbu uključiti više algoritma preuzorkovanja te njihove kombinacije pri rješavanju navedenog problema. Također, bilo bi od koristi uključiti i algoritme poduzorkovanja u svrhu ostvarivanja detaljnijih rezultata usporedbe kao i povećanja učinkovitosti. Analiza različitih funkcija predobrade podataka kao i različitih vrijednosti parametara k također bi omogućila stvaranje šireg pogleda na sam problem neuravnoteženosti kao i na utjecaj algoritama na većinske i manjinske uzorke.

LITERATURA

[1] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, 2011.

[2] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley-Interscience, 2000.

[3] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Fourth Edition, Academic Press, 4th ed., 2008.

[4] E. Alpaydin, Introduction to Machine Learning, Second Edition, The MIT Press, 2010.

[5] N. Japkowicz, The class imbalance problem: Significance and strategies, in Proc. IC-AI'2000, 2000.

[6] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Inf. Sci., vol. 250, pp. 113–141, 2013.

[7] D. Bajer, B. Zorić, M. Dudjak, and G. Martinović, Performance analysis of smotebased oversampling techniques when dealing with data imbalance, in 2019 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 265–271, 2019.

[8] N. García-Pedrajas, J.A. Romero del Castillo, G. Cerruela-García, A proposal for local k values for k-nearest neighbor rule, IEEE Trans. Neural Netw. Learn. Syst.28(2), 470–475 (2017), 2017.

[9] A. Fernández, S. García, F. Herrera, N. V. Chawla, SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary, J. Artif. Intell. Res., vol. 61, pp. 863–905, 2018.

[10] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: Review of methods and applications, Expert Syst. Appl., vol. 73, no. C, pp. 220–239, 2017.

[11] R. Lao, Machine Learning | Accuracy Paradox, linkedin.com, 2017, dostupno na: https://www.linkedin.com/pulse/machine-learning-accuracy-paradox-randy-lao [25.9.2020.]

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[13] M. Dudjak, G. Martinović, In-depth performance analysis of smote-based oversampling algorithms in binary classification, International Journal of Electrical and Computer Engineering Systems, vol. 11, no. 1, pp. 13–23, 2020.

[14] B. Zorić, D. Bajer, G. Martinović, Employing different optimisation approaches for SMOTE parameter tuning, in Proc. SST'16, pp. 191–196, 2016.

[15] J. Stefanowski, S. Wilk, Improving rule based classifiers induced by modlem by selective pre-processing of imbalanced data, in Proc. RSKD Workshop at ECML/PKDD, pp. 54–65, 2007.

[16] C. Bellinger, C. Drummond, N. Japkowicz, Beyond the boundaries of SMOTE, in Proc. ECML PKDD'16, pp. 248–263, 2016.

[17] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, in Proc. ICIC'05, pp. 878–887, 2005.

[18] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-SMOTE: Safelevel-synthetic minority over-sampling technique for handling the class imbalanced problem, in Proc. PAKDD'09, pp. 475–482, 2009.

[19] Y. Dong, X. Wang, A new over-sampling approach: Random-SMOTE for learning from imbalanced data sets, in Proc. KSEM'11, pp. 343–352, 2011.

[20] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, J. Mult.Valued Log. Soft Comput., vol. 17, no. 11, pp. 255–287, 2010.

[21] K. Bache, M. Lichman, UCI machine learning repository, archive.ics.uci.edu, 2013, dostupno na: http://archive.ics.uci.edu/ml [25.9.2020.]

SAŽETAK

U radu je napravljena usporedba algoritama preuzorkovanja zasnovanih na SMOTE za rješavanje problema neuravnoteženosti klasa. Rad daje detaljan opis pojma klasifikacije podataka, problema neuravnoteženosti klasa, algoritma *k*-najbližih susjeda kao i algoritma SMOTE i njegovih inačica, Borderline-SMOTE i Safe-level-SMOTE. Također su opisane različite mjere učinkovitosti uspoređivanih algoritama. Za potrebe rada ostvareno je programsko rješenje pomoću kojeg se određuje prosječna F-mjera primjene algoritma preuzorkovanja nad određenim stvarnim skupom podataka. Programsko rješenje ostvareno je u .NET okruženju koristeći programski jezik C#. Usporedba četiri algoritma preuzorkovanja uz klasifikaciju bez primjene preuzorkovanja provedena je nad šest skupova podataka različite brojnosti i omjera neuravnoteženosti. Rezultatima je dana prosječna F-mjera iz koje je vidljivo održavanje konzistentnosti rezultata neovisno o algoritmu prilikom čega je SMOTE jedini postigao maksimalnu F-mjeru u dva slučaja, dok njegove inačice nisu pokazale znatno bolje niti znatno lošije rezultate u odnosu na originalni SMOTE.

Ključne riječi: algoritam *k*-najbližih susjeda, F-mjera, klasifikacija, programski jezik C#, SMOTE

ABSTRACT

Comparison of oversampling algorithms based on SMOTE for dealing with the class imbalance problem.

This paper provides comparison of SMOTE-based oversampling algorithms for solving class imbalance problem. Paper gives detail description of terms data classification, class imbalance problem, k-nearest neighbors algorithm and SMOTE algorithm with its improved versions, Borderline-SMOTE and Safe-level-SMOTE. For the purpose of this paper, software was implemented to calculate average F-measure of oversampling algorithm over real-world datasets. Software was implemented in .NET framework using C# programming lanuage. Comparison of four oversampling algorithms added with method of no oversampling was conducted over six datasets with different number of samples and different imbalance ratios. Results provide average F-measure, which shows maintaned consistency of the results independent of the algorithm, whereby only SMOTE achieved maximum F-measure in two cases while its improvements did not show significantly better or worse results compared to the original SMOTE.

Keywords: *k*-nearest neighbors algorithm, F-measure, classification, C# programming language, SMOTE

ŽIVOTOPIS

Domagoj Rojnić rođen je u Osijeku 27. lipnja 1998. Osnovnu školu u Višnjevcu završava 2013. godine čime počinje pohađati Prirodoslovno-matematičku gimnaziju u Osijeku. Završetkom srednje škole i polaganjem državne mature, 2017. godine upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek u Osijeku.

PRILOZI /na CD-u/

1. Završni rad u doc, docx i PDF formatu
2. Projekt programskog rješenja
3. Podaci korišteni za eksperimentalnu analizu