Rukovanje problemom neuravnoteženosti klasa putem preuzorkovanja

Zmeškal, Ivan

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:125535

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-12-26**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Preddiplomski studij računarstva

RUKOVANJE PROBLEMOM NEURAVNOTEŽENOSTI KLASA PUTEM PREUZORKOVANJA

Završni rad

Ivan Zmeškal

Osijek, 2020.

SADRŽAJ

1.UVOD

Kada se govori u općenitom smislu, klasifikacija označuje nešto što se može razvrstavati po određenim poznatim svojstvima, grupiranje stvari u kategorije. Jednostavan primjer običnog razvrstavanja, klasifikacije, bio bi grupiranje ljudi ovisno o tome jesu li bolesni ili ne, odnosno pripadaju li grupi s oboljelima ili grupi koji nisu oboljeli. Kada u strojnom učenju nešto, neki uzorak, ne pripada nijednoj klasi, koristi, se klasifikator za određivanje i prepoznavanje svojstva tog uzorka te ga na temelju prepoznatog dodijeli nekoj klasi. Algoritam proizvodi klasifikator koji mapira svojstva ovih primjera izraženih u obliku parova atributa i vrijednosti, oznakama klase. Algoritmi kojima se rješavala problematika u ovome radu su algoritam *k*-najbliž susjeda te algoritam SMOTE. Algoritam SMOTE (engl. *Synthetic Minority Oversampling Technique*) je jedna od najčešće korištenih metoda preuzorkovanja za rješavanje problema neuravnoteženosti, kojoj je cilj uravnotežiti izvornu raspodjelu klasa. Neuravnoteženost klasa primjer je problema kod kojega je raspodjela klasa asimetrična. Kao rješenje za neuravnoteženost skupova postoje tehnike ponovnog uzorkovanja, koje su empirijski ispitane te ne ovise o klasifikatoru. Tehnike ponovnog uzorkovanja obuhvaćaju tehnike poduzorkovanja, tehnike preuzorkovanja i hibridne tehnike.

U drugom poglavlju opisano je što predstavlja pojam klasifikacije i kako dolazi do problema neuravnoteženosti klasa. U poglavlju su opisani algoritam *k*-najbližih susjeda i algoritam SMOTE te njegove interpretacije. U trećem poglavlju opisan je način rada ostvarenog programskog rješenja te način na koji se koristi. Četvrto poglavlje je eksperimentalna analiza ostvarena pomoću programskog rješenja.

1.1. Zadatak završnog rada

Zadatak završnog rada je opisati problem neuravnoteženosti klasa i kako utječe na učinkovitost klasifikatora. Objasniti SMOTE algoritam kao popularni i učinkoviti pristup preuzorkovanju koji se koristi dostupnim podacima. U radu prikazati programsko rješenje koje omogućuje preuzorkovanje danog skupa podataka pomoću mogućih interpretacija algoritma SMOTE te izvršiti eksperimentalnu analizu koristeći programsko rješenje.

2. PROBLEM NEURAVNOTEŽENOSTI KLASA I ALGORITAM SMOTE

U strojnom učenju izraz klasifikacija najčešće se povezuje s određenom vrstom učenja, u kojem su primjeri jedne ili više klasa, označeni nazivom klase, dani algoritmu učenja. Algoritam proizvodi klasifikator koji mapira svojstva ovih primjera, obično izraženih u obliku parova atributa i vrijednosti, oznakama klase. U primjeru gdje je klasa nepoznata, klasificira se kad klasifikator dodijeli oznaku klase na temelju njegovih svojstava.

Kao jedan od često korištenih klasifikatora ističe se algoritam *k*-najbližih susjeda (engl. *k-nearest neighbors*, *k*-NN). U prepoznavanju uzorka, algoritam *k*-NN je neparametarska metoda, koja se prilagođava samim podacima umjesto da se temelji na njihovoj prethodno definiranoj razdiobi, te se koristi za klasifikaciju i regresiju. U oba se slučaja ulaz sastoji od *k* najbližih uzoraka u prostoru značajki. Izlaz u *k*-NN klasifikaciji je oznaka klase kojoj pripada nepoznat uzorak. Objekt je klasificiran višestrukim glasanjem svojih susjeda, pri čemu je objekt dodijeljen klasi koja je najčešća među njegovim *k* najbližim susjedima (*k* je pozitivan cijeli broj, u literaturi su 1, 3 i 5 najčešće vrijednosti). Ako je $k = 1$, objekt je jednostavno dodijeljen klasi tog jednog najbližeg susjeda. Algoritam *k*-NN je vrsta lijenog učenja, gdje se funkcija samo aproksimira lokalno, a sva se izračunavanja odgađaju do provjere funkcije.

2.1. Klasifikacija i problem neuravnoteženosti klasa

Klasifikacija kao pojam u strojnom učenju predstavlja problem identificiranja kojoj od skupa kategorija pripada novo promatranje, na temelju niza podataka koji sadrže promatranja čija je pripadnost kategorijama poznata.

Problem neuravnoteženosti klasa je primjer problema s klasifikacijom, gdje je raspodjela klasa asimetrična. Raspodjela može varirati od neznatne asimetrije do itekako značajne neravnoteže gdje je u manjinskoj klasi jedan primjer u odnosu na stotine, tisuće ili milijune primjera u većinskoj klasi ili klasama [1]. Neuravnotežene klase predstavljaju izazov za prediktikvno modeliranje jer je većina algoritama strojnog učenja korištenih za klasifikaciju oblikovana oko pretpostavke jednakog broja primjera za svaku klasu. Navedeno rezultira modelima sa slabim prediktivnim karakteristikama, osobito kod manjinskih klasa. To je problem, zato što je uglavnom manjinska klasa važnija klasa. To je jedan od nekoliko razloga zbog kojih dolazi do veće osjetljivosti na pogreške u klasifikaciji manjinske klase. Drugi razlog za povećanje osjetljivosti na pogreške u klasifikaciji, koji može rezultirati davanjem prednosti većinskoj klasi, nastaje prilikom korištenja globalnih mjera uspješnosti za vođenje procesa

učenja. Još jedna situacija koja šteti efikasnoj klasifikaciji nastaje kada se mali skupovi primjera manjinskih klasa prepoznaju kao šum pa ih klasifikator pogrešno raspodjeli [1].

Kao rješenje neuravnoteženih skupova podataka se nude tehnike ponovnog uzorkovanja. Te tehnike su prema [1] empirijski ispitane i dokazano je da su uglavnom korisne te im je najveća prednost što ne ovise o klasifikatoru. Tehnike ponovnog uzorkovanja je moguće podijeliti u tri grupe obitelji:

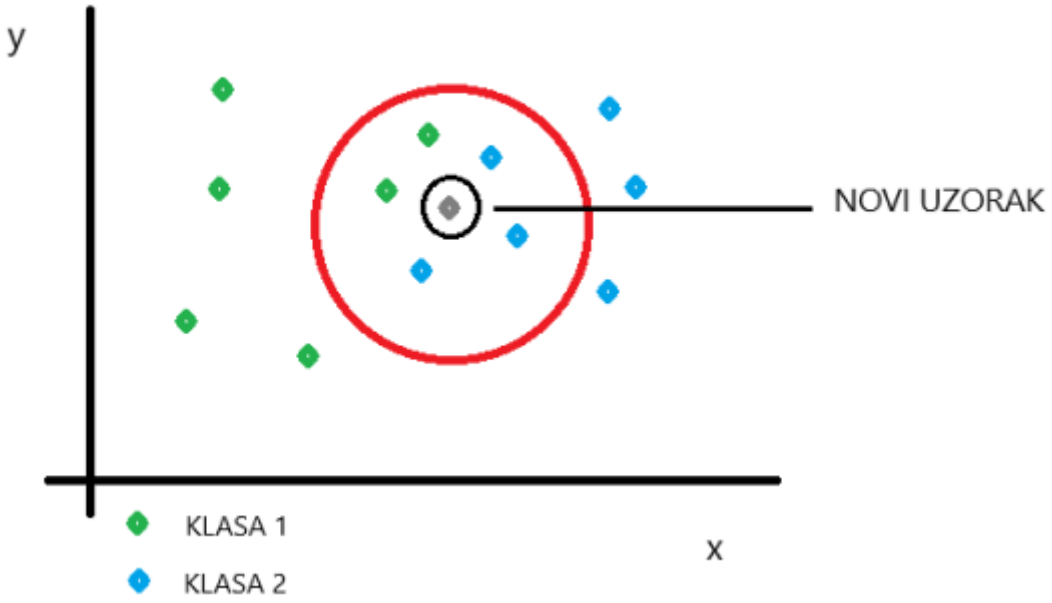
- tehnike poduzorkovanja: stvaraju podskup originalnog skupa eliminiranjem uzoraka, obično se radi o uzorcima većinske klase
- tehnike preuzorkovanja: stvaraju novi skup od originalnog skupa kopiranjem uzoraka originalnog skupa ili stvaraju nove uzorke od postojećih
- hibridne tehnike: kombiniranje tehnika poduzorkovanja i preuzorkovanja.

Najjednostavnije tehnike iz navedenih obitelji su one koje nisu optimizacijske, kao što su slučajno preuzorkovanje *(*engl. *random oversampling)* i slučajno poduzorkovanje *(*engl. *random undersampling)*. Međutim, kod slučajnog preuzorkovanja dolazi do stvaranja točnih kopija već postojećih uzoraka, dok je glavni nedostatak slučajnog poduzorkovanja potencijalno odbacivanje podataka koji mogu biti korisni, odnosno važni za proces učenja [1].

2.1.1. Algoritam *k*-najbližih susjeda

Algoritam *k*-NN je jedan od najjednostavnijih algoritama za klasifikaciju i jedan od najčešće korištenih. Algoritam koji koristi postojeće podatke te na temelju njihovih udaljenosti od novog nepoznatog uzorka odabire odgovarajuću klasu novom uzorku. Algoritam *k*-najbližih susjeda je neparametarski algoritam, što znači da ne radi nikakve pretpostavke o razdiobi podataka, struktura modela se određuje iz podataka. Iz tog razloga se algoritam *k*-najbližih susjeda nameće kao jedan od prvih izbora za vršenje klasifikacije kada o distribuciji podataka ima malo ili nema uopće prethodnih saznanja. Kada se *k*-najbližih susjeda koristi za klasifikaciju podataka, izlaz je pripadnost klasi (predviđa oznaku). Pri odabiru vrijednosti *k* bitno je da ne bude paran broj, ako se radi o problemu gdje postoje samo dvije klase. U literaturi su često korištene vrijednosti 1, 3 i 5, ali odabir vrijednosti parametra *k* se vrši ovisno o skupu podataka i na temelju ispitivanja učinkovitosti za različite vrijednosti [2].

Objekt je klasificiran većinom glasova svojih susjeda, pri čemu je objekt dodijeljen klasi koja je najčešća među njegovih k najbližih susjeda, primjer klasificiranja objekta prikazan na slici 2.1.



Slika 2.1. *Klasificiranje novog uzorka u slučaju k=5.*

U slučaju kada se koristi $k = 1$ za uzorke x_i gdje je $i = 1,2,...,M$, onda se definira l – dimenzionalan prostor koji se dijeli na M područja R_i gdje je R_i definiran formulom:

$$Ri = \{x: d(x, x_i) < d(x, x_j), i \neq j\} \quad (2\text{-}1)$$

gdje R_i sadrži sve točke u prostoru bliže uzorku x_i od ostalih uzoraka s obzirom na udaljenost d. Takva raspodjela se naziva Voronijev dijagram [2]**.**

U nastavku na slici 2.2. slijedi pseudo-kod algoritma k-najbližih susjeda:

```
1: Razvrstaj (X,x)
2:  ZA i=1 DO m RADI
3:       Izračunaj udaljenost d(X_i , x)
4:  KRAJ
5: Sortiraj udaljenosti uzlazno za k najbližih susjeda
6: Dodijeliti x najčešćoj klasi
```

Slika 2.2. *Algoritam k-najbližih susjeda*

U opisu algoritma na slici 2.2 je: *m* broj uzoraka, skup podataka *X* za treniranje i uzorak *x* kojeg treba svrstati su ulazni podaci, a izlazni podatak je klasa u koju je *x* svrstan.

Navedeni pseudo-kod sadrži funkciju za računanje udaljenosti između pojedinih uzoraka, odnosno izračunava udaljenost između novog uzorka i svih njegovih susjeda. Postoji nekoliko različitih funkcija koje se koriste za računanje udaljenosti, kao što su: Mahalanobisova i Minkowski te specijalni slučajevi Minkowskijeve udaljenosti: Euklidska i Manhattan udaljenost od kojih je Euklidska najčešće korištena. Za Euklidsku udaljenost vrijedi: neka su A i B vektori gdje je $A = (x_1, x_2, ..., x_n)$, a $B = (y_1, y_2, ..., y_n)$ tada se udaljenost računa na sljedeći način:

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2-2}$$

gdje n predstavlja broj atributa.

Neke od prednosti algoritma su da nema pretpostavki o podacima te je algoritam jednostavan za razumjeti i interpretirati. Također ima i relativno visoku točnost, smatra se svestranim algoritmom, može biti koristan za klasifikaciju i regresiju. Kao najveći nedostaci se ističu potreba za velikim količinama memorije i spora faza predviđanja ako postoji velik broj susjeda [2].

2.1.2. Vrednovanje učinkovitosti algoritama klasifikacije

Kao što postoji nekoliko različitih funkcija za izračun udaljenosti, isto tako postoji više načina za ocjenjivanje učinkovitosti algoritma klasifikacije. Najzastupljenija je matrica zbunjenosti (engl. *confusion matrix*). Nakon što su podaci za treniranje provedeni kroz klasifikator, matrica zbunjenosti prikazuje koliko je uspješno algoritam obavio klasificiranje, uz napomenu da je sama matrica definirana u ovisnosti o algoritmu korištenog klasifikatora. Uobičajena oznaka za matricu zbunjenosti dimenzija $l \times l$ je C te vrijedi njezin općeniti zapis:

$$C = \{c_{ij}\}, i, j \in \{1, 2, \ldots, l\} \tag{2-3}$$

gdje su i i j oznake indeksa retka, odnosno stupca [3]. Općenito govoreći, postoje dvije bitne stvari: matricu je moguće ostvariti tako da uključuje podatke za izvedbu više od jednog algoritma. Druga stvar se odnosi na izlazne vrijednosti klasifikatora, one ovise o podacima za treniranje te algoritam uči u skladu sa zadanim podacima. Iz tog razloga se vrijednosti matrice zbunjenosti, kao i mjere izvedene iz matrice, definiraju ovisno o klasifikatoru.

Kada je u pitanju binarna klasifikacija, gdje je manjinska klasa pozitiv, a većinska negativ, radi se o matrici gdje redovi prikazuju stvarnu klasu testnog uzorka, a stupci prikazuju klasu koju je klasifikator predvidio. Vrijednosti koje matrica zbunjenosti može imati su: broj lažno pozitivnih

(engl. *false positive*, FP), lažno negativnih (engl. *false negative*, FN), stvarno pozitivnih (engl. *true positive*, TP) i stvarno negativnih (engl. *true negatives*, TN). Vrijednosti stvarno pozitivnih i stvarno negativnih predstavljaju broj uzoraka za koje je klasifikator ispravno predvidio klasu (pripadaju li većinskoj ili manjinskoj klasi), analogno tome vrijedi i za lažno pozitivne te lažno negativne vrijednosti [3]. Opća matrica je prikazana na slici 2.3.

	PREDVIĐENO NEGATIVNI	**PREDVIĐENO POZITIVNI**
STVARNO NEGATIVNI	**STVARNO NEGATIVNI (TN)**	**LAŽNO POZITIVNI (FP)**
STVARNO POZITIVNI	**LAŽNO NEGATIVNI (FN)**	**STVARNO POZITIVNI (TP)**

Slika 2.3. *Općeniti prikaz matrice zbunjenosti za slučaj binarne klasifikacije.*

Prve mjere za vrednovanje učinkovitosti algoritma klasifikacije izvedene iz matrice zbunjenosti su preciznost klasifikacije te njezin ekvivalent stopa pogreške. Preciznost klasifikacije (engl. *accuracy*) mjeri udio uzoraka iz testnog skupa koji su ispravno svrstani algoritmom klasifikatora te obuhvaća uzorke svih klasa u skupu. Analogno preciznosti klasifikatora, stopa pogrešaka mjeri udio pogrešno svrstanih uzoraka. Preciznost klasifikatora (A_{CC}) i stopu pogreške (R_T) je, u ovisnosti o matrici zbunjenosti, moguće odrediti pomoću izraza:

$$Acc_T(f) = \frac{c_{11}(f)+c_{22}(f)}{c_{11}(f)+c_{12}(f)+c_{21}(f)+c_{22}(f)} = \frac{TP+TN}{P+N} \tag{2-4}$$

$$R_T(f) = 1 - Acc_T(f) = \frac{c_{12}(f)+c_{21}(f)}{c_{11}(f)+c_{12}(f)+c_{21}(f)+c_{22}(f)} = \frac{FN+FP}{P+N} \tag{2-5}$$

Ove dvije mjere su korisne i pouzdane kada su u pitanju uravnotežene klase. Kada se radi o klasama s različitim brojem uzoraka, dolazi do neuravnoteženosti klasa, kada je jedna od klasa značajno veća (u smislu broja uzoraka) od druge. Tada preciznost klasifikacije i stopa pogreške ne mogu biti vjerodostojni pokazatelji kvalitete algoritma klasifikacije. Ulazni podaci matrice zbunjenosti sami po sebi mogu nositi određene informacije, ali iz istog razloga svaki podatak onda može i dovesti do zablude [3]. Zato se općenito pokušava uzeti u obzir informacije koje ti podaci prenose u ovisnosti na druge mjerodavne ulazne podatke matrice. Budući da se preciznost klasifikacije i stopa pogreške oslanjaju na podatke na dijagonali matrice zbunjenosti kako bi dobili ukupnu ocjenu klasa, nisu pogodne za neuravnotežene klase te se uparuje s drugim,

prikladnijim mjerama za takav slučaj. Jedna od tih mjera je F-1 mjera (engl. *F-1 score*) koja sjedinjuje druge dvije mjere: stopa stvarno pozitivnih (engl. *true-positive rate, TRP*) ili odziv (engl. *recall*) te pozitivno prediktivnu vrijednost (engl. *positive predictive value, PPV*) ili preciznost (engl. *precision*). F-1 mjera je ponderirana harmonijska sredina navedenih i određena je izrazom [3]:

$$F_a = \frac{(1+\alpha)[Prec(f)*Rec(f)]}{\{[\alpha*Prec(f)]+Rec(f)\}} \tag{2-6}$$

gdje je α bilo koji realan broj veći od nule.

Sljedeća mjera se naziva geometrijska sredina istina (engl. *geometric mean of trues, G_{mean}*). Ona uzima u obzir i pozitivnu i negativnu klasu. Geometrijska sredina istina je određena izrazom [3]:

$$G_{mean1}(f) = \sqrt{TPR(f) * TNR(f)} \tag{2-7}$$

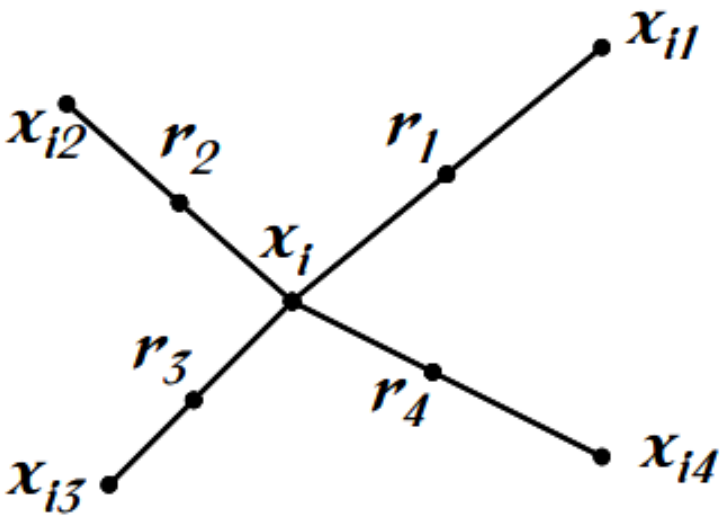
Postoji mogućnost da G_{mean} poprimi vrijednost 1 u slučaju kada su stopa stvarno pozitivnih i stopa stvarno negativnih jednakog iznosa, odnosno *TPR(f)=TNR(f)=1.* Druga varijanta geometrijske sredine istina može za izračun koristiti prethodno spomenutu preciznost klasifikatora, *Prec(f),* tada je izraz [3]:

$$G_{mean2}(f) = \sqrt{TPR(f) * Prec(f)} \tag{2-8}$$

Stoga G_{mean2} uzima u obzir udio stvarno pozitivnih svrstanih kao pozitivni i udio uzoraka koji su svrstani kao pozitivni, a koji su uistinu pozitivni [3].

2.2. Algoritam SMOTE

Algoritam SMOTE je jedna od najčešće korištenih metoda preuzorkovanja za rješavanje problema neuravnoteženosti. Cilj mu je uravnotežiti izvornu raspodjelu klasa. Za razliku od nasumičnog preuzorkovanja, koje primjenjuje jednostavno kopiranje uzoraka manjinske klase, SMOTE stvara nove sintetičke uzorke. Primjer funkcioniranja SMOTE algoritma prikazan je na jednostavnom primjeru na slici 2.4.



Slika 2.4. *Jednostavan primjer funkcioniranja SMOTE algoritma [4, str.867].*

Neka su prema slici 2.4. x_i, x_{i1}, x_{i2}, x_{i3} i x_{i4} uzorci manjinske klase koje su odabrane kao polazna točka za stvaranje novih sintetičkih točaka podataka, odnosno uzoraka [4]. Koristeći neku funkciju za izračun udaljenosti, odabiru se najbliži susjedi točke x_i, na slici 2.4. označeni indeksima od 1 do 4. Zatim se svaka udaljenost između x_i i pojedinog susjeda množi slučajnim brojem od 0 do 1, što rezultira odabirom slučajne točke gdje se kreira novi sintetički uzorak, na slici 2.4. označene r_1, r_2, r_3 i r_4 . U nastavku na slici 2.5. slijedi pseudo-kod SMOTE algoritma:

SMOTE (*T*,*N*,*k*)

1: **AKO** je *N* veći od 100 **ONDA**

2: Nasumice poredati *T* uzoraka manjinske klase

3: *T* = *(N* / 100*)* * *T*

4: *N* = 100

5: **KRAJ**

6: *N* = *N* / 100 //pretpostavlja se da je *N* višekratnik broja 100

7: **ZA** *i* = 1 **DO** *T* **RADI**

8: Odredi *k* najbližih susjeda za *x_i*, spremi indekse u *NN_niz*

9: Popuni (*N*, *i*, *NN_niz*)

10: **KRAJ**

Slika 2.5. *Algoritam SMOTE.*

U opisu algoritma na slici 2.5. ulazni podaci su broj uzoraka manjinske klase *T*, iznos preuzorkovanja *N*, broj najbližih susjeda *k*, izlazni podaci su (*N* / 100) * *T* sintetičkih uzoraka manjinske klase, a dodatne varijable su: niz za originalne uzorke manjinske klase *Original_niz*, cjelobrojna varijabla koja broji nove uzorke *novi_indeks* i niz za spremanje novih uzoraka *Sintetički_niz*.

U nastavku na slici 2.6. slijedi pseudo-kod funkcije Popuni spomenute u prethodnom pseudo-kodu.

```
POPUNI (N, i , NN_niz)
 1: DOK N ≠ 0 RADI
 2:     nn = slučajan_odabir(1,k)
 3:     ZA atribut = 1 DO broj_atributa RADI
 4:         Izračunaj udaljenost između najbližeg susjeda i trenutnog uzorka
 5:         Odredi nasumičan broj od 0 do 1, pomnoži sa udaljenosti
 6:         Popuni Sintetički_niz[novi_indeks][atribut] sa sintetičnim uzorcima
 7:     KRAJ
 8:     novi_indeks ++
 9:     N = N - 1
10: KRAJ
```

Slika 2.6. *Algoritam funkcije Popuni*.

U opisu algoritma na slici 2.6. ulazni podaci su: broj novostvorenih uzoraka *N*, indeks originalnog uzorka *i*, niz najbližih susjeda *NN_niz*, a izlazni podaci koji predastavljaju *N* novih sintetičkih uzoraka spremljenih u *Sintetički_niz* [4].

Mogućnost povećavanja manjinske klase stvaranjem novih uzoraka rezultira pozitivnim učinkom u sprječavanju, odnosno smanjivanju rizika prekomjernog prilagođavanja klasifikatora podskupu za treniranje (engl. *overfitting*) te dovodi do poboljšanja kvalitete samog klasifikatora. Da bi bila općenito primjenjiva, metoda sinteze ne bi smjela zahtijevati ponovno stvaranje novih uzoraka. Umjesto toga bi trebalo biti moguće stvaranje sintetičkih uzoraka na temelju postojećih podataka u skupu za treniranje ili nekih drugih, od ranije poznatih, uputa ili svojstava o raspodjeli podataka. Očekuje se da stvaranje novih uzoraka manjinske klase uzrokuje „širenje“ prostora manjinske klase koje poboljšava učinak klasifikatora [5].

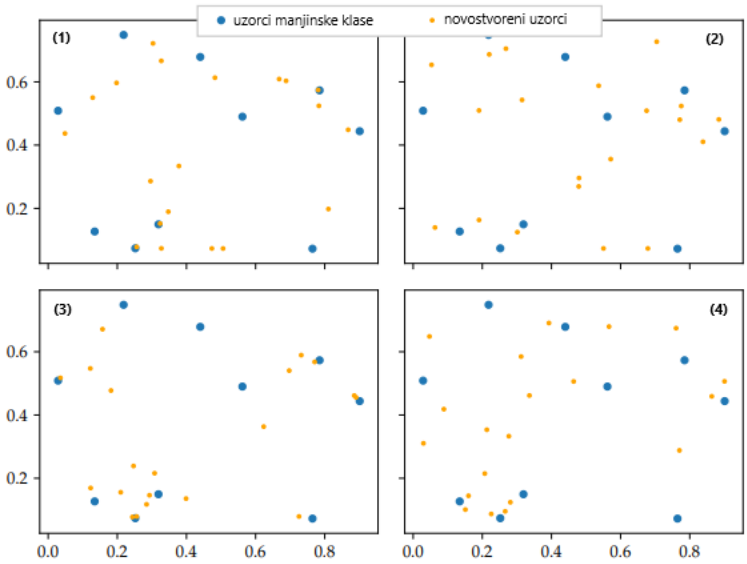
Međutim, prilikom sintetiziranja novih uzoraka može doći do povećanja varijance i pojačanja šumova. Do povećanja varijance dolazi zbog zanemarivanja raspodjela klasa tijekom sintetiziranja novih uzoraka u okolini slučajnog najbližeg susjeda [5]. U slučaju kada u skupu postoje šumovi onda se oni pojačaju samim kreiranjem novih uzoraka što rezultira smanjivanjem kvalitete klasifikatora. Ako postoje dva potkoncepta manjinske klase odvojena koceptom većinske klase, SMOTE algoritam sintetizira nove primjerke u većinskom prostoru duž rubova koji povezuju uzorke manjinske klase u različitim potkonceptima [5].

Budući da je SMOTE algoritam poprilično jednostavan, u literaturi postoji velik broj različitih interpretacija osmišljenih s ciljem da se smanje neki njegovi nedostaci. Neke od unaprijeđenih varijanti su *borderline-SMOTE, random-SMOTE* i *weighted-SMOTE* algoritam. Algoritam *borderline-SMOTE* je specifičan po izboru uzoraka koje koristi za uzorkovanje: odabire samo one blizu granice kako bi se dodatno naglasila razlika između manjinske i većinske klase. Postoje dvije interpretacije *borderline-SMOTE* algoritma, *borderline-SMOTE1* i *borderline-SMOTE2*, koje se razlikuju po odabiru susjeda korištenih za stvaranje novih uzoraka - prva interpretacija koristi susjede iz samo manjinske klase, dok druga koristi uzorke i manjinske i većinske klase [6]. Algoritam *random-SMOTE* stvara nove uzorke pomoću dva slučajno odabrana susjeda s kojima čini trokut od tri podatkovne točke unutar kojeg se stvara određen broj novih uzoraka [7]. U interpretaciji *weighted-SMOTE* dodijeljene su težine kojima se određuje količina novih uzoraka SMOTE algoritma stvorenih u odnosu na pojedinačni uzorak manjinske klase. Težina predstavlja Euklidsku udaljenost od pojedinog uzorka do svih ostalih uzoraka manjinske klase [8].

2.2.1. Moguće interpretacije

Za navođenje različitih interpretacija SMOTE algoritma, bitno je uočiti razliku između pristupa stvaranja novih uzoraka. Prva situacija predstavlja stvaranje novih uzoraka na zamišljenoj dužini između uzoraka manjinske klase, algoritam se oslanja na prostor podataka (engl. *data space*). Druga situacija nastaje kada se algoritam oslanja na prostor značajki (engl. *feature space*). Kombinacijom navedenih načina stvaranja novih uzoraka i ovisno o tome jesu li uzorci manjinske klase nasumično odabrani ili je odabran svaki uzorak, mogu se izvesti četiri interpretacije SMOTE algoritma [7]. Prva interpretacija bi bio SMOTE algoritam koji se oslanja na prostor podataka i koristi svaki uzorak manjinske klase skupa za treniranje da bi stvorio nove uzorke, na slici 2.7. označen brojem (1). Sljedeća interpretacija, označena brojem (3) na slici

2.7., se također oslanja na prostor podataka, ali ne koristi svaki manjinski uzorak za preuzorkovanje, već ih odabire nasumično. Analogno tim interpretacijama, interpretacije označena brojem (2) i (4) na slici 2.7., koriste svaki uzorak manjinske klase, odnosno nasumično odabire uzorak, ali se algoritam oslanja na prostor značajki. Razlike između navedenih interpretacija su prikazane na slici 2.7. [7].



Slika 2.7. *Prikaz razlika između interpretacija SMOTE algoritma [7].*

3. OSTVARENO PROGRAMSKO RJEŠENJE

Programsko rješenje je ostvareno u razvojnom okruženju *JetBrains PyCharm Community*, u programskom jeziku python. Program omogućuje učitavanje podatkovnih datoteka u CSV (engl. *comma-separated values*) formatu koje je prije samog korištenja moguće podijeliti na dva podskupa: podskup za treniranje i podskup za testiranje. U programu je implementiran algoritam k-najbližih susjeda koji je korišten za klasifikaciju. Osim algoritma za klasifikaciju, implementiran je algoritam za slučajno preuzorkovanje te interpretacije algoritma SMOTE.

3.1. Način rada programskog rješenja

U programu se nalaze funkcije za klasifikaciju k-NN algoritmom bez ikakve dorade nad podacima za treniranje, zatim k-NN gdje se vrši nasumično preuzorkovanje. Nakon toga dolazi do preuzorkovanja algortimom SMOTE prikazano na slici 3.1. Prva interpretacija koristi prve uzorke i između njih stvara novi nasumičan uzorak dok druga interpretacija bira uzorke nasumično. Ove dvije interpretacije se oslanjaju na podatkovni prostor te u skladu s tim stvaraju nove uzorke. Kod druge dvije interpretacije također jedna bira svaki uzorak, a jedna nasumično, ali se oslanjaju na prostor značajki umjesto podatkovnog prostora. Osim broja uzoraka manjinske i većinske klase, program ispisuje matricu zbunjenosti, geometrijsku sredinu istina te F-1 mjeru kao što je prikazano prema slici 3.2.

```
def smote_algoritam_resample_random(train_X_manji, len_veci, list_matrix_index_i, list_matrix_index_j):
    train_X_manji_pd = pd.DataFrame(train_X_manji).to_numpy()
    start_data_num = train_X_manji.shape[0]
    end_data_num = len_veci
    train_X_reshape = train_X_manji_pd
    br = start_data_num
    for temp in range(1, len(list_matrix_index_i) - 1):
        k = random.random()
        n = random.randint(1, len(list_matrix_index_i) - 1)
        prvi_dio = (k * train_X_manji_pd[list_matrix_index_i[n]])
        drugi_dio = ((1.0 - k) * train_X_manji_pd[list_matrix_index_j[n]])
        new_row = []
        for (num1, num2) in zip(prvi_dio, drugi_dio):
            new_row.append(num1 + num2)
        train_X_reshape = pd.np.vstack((train_X_reshape, new_row))
        br = br + 1
        if br == end_data_num:
            break
    return train_X_reshape
```

Slika 3.1. *Implementacija algoritma SMOTE.*

```
[[264  11]
 [ 12  15]]              precision    recall  f1-score

    negative       0.96      0.96      0.96       275
    positive       0.58      0.56      0.57        27

    accuracy                           0.92       302
   macro avg       0.77      0.76      0.76       302
weighted avg       0.92      0.92      0.92       302
 0.9238410596026491
```

Slika 3.2. *Primjer ispisa.*

3.2. Prikaz i način upotrebe programskog rješenja

Da bi program počeo s radom, potrebno je prvo unijeti ulazne podatke. Prvi potreban podatak je putanja do određene CSV datoteke, unos broja stupaca klase, a zatim podatak koji govori ima li datoteka zaglavlje ili nema. Nakon odabira ulaznih vrijednosti, program koristeći pandas modul učitava CSV datoteku te odstranjujemo moguće nedefinirane (engl. *not-a-number*) vrijednosti što je prikazano na slici 3.4. Zatim definiramo broj klasa nad kojima se izvršava k-NN te slijedi daljnja manipulacija podacima te ispitivanje.

```
def read_data(file_path):
    dataset = pd.read_csv(file_path, header=header_value)
    dataset.dropna()
    return dataset
```

Slika 3.4. *Učitavanje CSV datoteke*

Zatim slijedi podjela podataka na podskupove za treniranje i testiranje, izvršena po kodu prikazanom na slici 3.5. Dijeljenje se izvršava u omjeru 70:30.

```
def split_dataset(dataset):
    train, test = train_test_split(dataset, test_size=0.3)
    return test,train
```

Slika 3.5. *Podjela na podskupove za treniranje i testiranje.*

Nakon podjele podataka dolazi do preuzorkovanja ulaznim podacima, izvršava se nasumično preuzorkovanje ili preuzorkovanje nekom od interpretacija SMOTE algoritma te se vrši analiza i ocjenjivanje dobivenih rezultata.

4. EKPERIMENTALNA ANALIZA

U eksperimentalnoj analizi izvršena je usporedba ranije navedenih interpretacija algoritma SMOTE. Podatkovni skupovi nad kojima je izvršen eksperiment su neuravnoteženi skupovi podataka preuzeti sa KEEL repozitorija [9] i prikazani su u tablici 4.1. U eksperimentu je korišten klasifikator k-najbližih susjeda s vrijednosti $k = 3$ nad kojima se najprije izvršava podjela na podskupove za treniranje i testiranje. Posljednji stupac u tablici 4.1. odnosi se na omjer broja uzoraka većinske i manjinske klase (engl. *imbalance ratio, IR*).

Tablica 4.1. *Korišteni skupovi podataka.*

Naziv	Broj stupaca	Broj uzoraka	IR omjer
yeast-0-2-5-6_vs_3-7-8-9	8	1004	9.14
Pima	8	768	1.87
yeast1	8	1484	2.46
yeast-1-2-8-9_vs_7	8	947	30.57
yeast-2_vs_8	8	482	23.1

4.1. Postavke eksperimenta

Preuzeti skupovi su prvotno bili zapisani u tekstualnom DAT formatu te su naknadno formatirani u CSV format koristeći program *Microsoft Excel*. Svaki skup je podijeljen u podskup za treniranje i testiranje u omjeru 70:30 te su uklonjene nedefinirane vrijednosti. Nakon podjele, nad svakim skupom je izvršeno preuzorkovanje svakim od prethodno navedenih interpretacija preuzorkovanja. Taj postupak se ponavlja 30 puta pri čemu je svaki put nova podjela skupa podataka izvršena. Parametri za sve interpretacije algoritma SMOTE su jednaki: parametar k iznosi $k = 3$, a parametar N za iznos preuzorkovanja je $N = 100$. Parametar N govori koliko se želi „povećati“ manjinsku klasu, ovdje se želi udvostručiti podatke manjinske klase pa je odabran $N = 100$. Konačni rezultati su određeni kao aritmetička sredina svakog ponavljanja uz istaknute minimalne i maksimalne vrijednosti te standardnu devijaciju kod mjera korištenih za ocjenjivanje klasifikatora.

4.2. Rezultati

U tablici 4.2. i tablici 4.3. su prikazani odnosi većinske (u tablicama označena brojem 1) i manjinske (u tablicama označena brojem 0) klase podskupova za treniranje prije i poslije izvršene klasifikacije nad skupom obrađenim algoritmom za preuzorkovanje. Nazivi stupaca u tablici 4.3. ozačavaju izvršeno nasumično preuzorkovanje i preuzorkovanje bilo kojom od interpretacija algoritma SMOTE.

Tablica 4.2. *Raspodjela klasa podskupova za treniranje prije preuzorkovanja.* .

Naziv	**1**	**0**
yeast-0-2-5-6_vs_3-7-8-9	624	71
Pima	351	188
yeast1	922	113
yeast-1-2-8-9_vs_7	464	23
yeast-2_vs_8	322	14

Tablica 4.3. *Raspodjela klasa nakon preuzorkovanja*

Naziv	Nasumično		SMOTE	
	1	**0**	**1**	**0**
yeast-0-2-5-6_vs_3-7-8-9	624	624	624	171
Pima	351	348	351	345
yeast1	922	948	922	288
yeast-1-2-8-9_vs_7	464	463	464	54
yeast-2_vs_8	322	323	322	39

.

Sljedeća tablica sadržava matrice zbunjenosti za svaki skup pojedinačno i za svaku interpretaciju. Nazivi stupaca SMOTE1, SMOTE2, SMOTE3 i SMOTE4 su oznake korištene interpretacije. Pod oznakama SMOTE1 i SMOTE3 su interpretacije gdje se algoritam oslanja na prostor podataka, razlika je što se u SMOTE1 koristi svaki uzorak manjinske klase skupa za treniranje da bi stvorio nove uzorke, a u SMOTE3 se uzorci nasumično biraju. Pod oznakama SMOTE2 i SMOTE4 su interpretacije koje se oslanjaju na prostor značajki. Odabir uzoraka manjinske klase skupa za treniranje je analogan interpretacijama SMOTE1 i SMOTE3, interpretacija SMOTE2 koristi svaki uzorak, a SMOTE4 ih nasumično bira. Budući da je riječ o

30 puta ponovljenom postupku, za svaki skup je u tablici 4.4. prikazana po jedna matrica zbunjenosti ispunjena prosječnim vrijednostima.

Tablica 4.4. *Matrice zbunjenosti*

Naziv	Nasumično	SMOTE1	SMOTE2	SMOTE3	SMOTE4
yeast-0-2-5-6_vs_3-7-8-9	**0**: 137, 3 **1**: 3, 3	**0**: 134, 3 **1**: 4, 4	**0**: 137, 3 **1**: 2, 3	**0**: 135, 4 **1**: 1, 5	**0**: 134, 4 **1**: 3, 4
Pima	**0**: 124, 25 **1**: 49, 36	**0**: 112, 47 **1**: 23, 49	**0**: 109, 44 **1**: 31, 47	**0**: 96, 58 **1**: 20, 57	**0**: 97, 54 **1**: 30, 50
yeast1	**0**: 391, 14 **1**: 16, 29	**0**: 375, 32 **1**: 35, 34	**0**: 368, 20 **1**: 15, 43	**0**: 387, 20 **1**: 9, 30	**0**: 387, 20 **1**: 9, 30
yeast-1-2-8-9_vs_7	**0**: 192, 6 **1**: 6, 1	**0**: 266, 11 **1**: 5, 3	**0**: 275, 6 **1**: 4, 0	**0**: 271, 10 **1**: 3, 1	**0**: 271, 10 **1**: 3, 1
yeast-2_vs_8	**0**: 137, 3 **1**: 3, 3	**0**: 134, 3 **1**: 4, 4	**0**: 137, 3 **1**: 2, 3	**0**: 135, 4 **1**: 1, 5	**0**: 134, 4 **1**: 3, 4

.

Po prikazanim matricama zbunjenosti se može primijetiti prednost SMOTE algoritma u odnosu na slučajno preuzorkovanje, u većini slučajeva ima veći broj točno predviđenih uzoraka. U tablici 4.5. se nalaze F-1 mjera i geometrijska sredina istina. Kao i u prethodnoj tablici, vrijednosti F-1 mjere i geometrijske sredine istina su njihove aritmetičke sredine, ali su istaknute i njihove minimalne i maksimalne vrijednosti, kao i standardna devijacija.

Tablica 4.5. *Prikaz vrijednosti F-1 mjere i geometrijske sredine istina*

SLUČAJNO PREUZORKOVANJE						
Naziv	F-1 mjera	Geometrijska sredina istina	F-1 min/max	G_{mean} min/max	Standardna devijacija F-1	Standarnda devijacija G_{mean}
yeast-0-2-5-6_vs_3-7-8-9	0,96	0,96	0,95/0,98	0,95/0,97	0,01	0,01
Pima	0,74	0,81	0,71/,78	079/0,84	0,03	0,02
yeast1	0,96	0,97	0,94/0,98	0,96/0,98	0,03	0,01
yeast-1-2-8-9_vs_7	0,97	0,96	0,96/0,98	0,95/0,98	0,02	0,01
yeast-2_vs_8	0,97	0,97	0,95/98	0,95/0,99	0,02	0,02
SMOTE1						
yeast-0-2-5-6_vs_3-7-8-9	0,95	0,90	0,93/0,96	0,814/0,934	0,01	0,034
Pima	0,72	0,676	0,66/0,78	0,613/0,722	0,06	0,104
yeast1	0,96	0,908	0,95/0,97	0,903/0,944	0,01	0,036
yeast-1-2-8-9_vs_7	0,95	0,913	0,94/0,97	0,903/0,938	0,02	0,025
yeast-2_vs_8	0,84	0,83	0,78/0,87	0,80/0,92	0,03	0,09
SMOTE2						
yeast-0-2-5-6_vs_3-7-8-9	0,94	0,89	0,92/0,96	0,864/0,920	0,02	0,012
Pima	0,71	0,66	0,66/0,75	0,597/0,712	0,045	0,059
yeast1	0,96	0,93	0,95/0,97	0,899/0,953	0,027	0,006
yeast-1-2-8-9_vs_7	0,94	0,902	0,92/0,96	0,869/0,933	0,02	0,008
yeast-2_vs_8	0,98	0,96	0,96/0,99	0,937/0,986	0,031	0,002
SMOTE3						
yeast-0-2-5-6_vs_3-7-8-9	0,95	0,91	0,93/0,97	0,877/0,934	0,01	0,013
Pima	0,73	0,67	0,68/0,79	0,619/0,732	0,05	0,296
yeast1	0,96	0,93	0,95/0,97	0,899/0,953	0,01	0,025
yeast-1-2-8-9_vs_7	0,956	0,925	0,93/0,97	0,889/0,956	0,021	0,296
yeast-2_vs_8	0,98	0,968	0,97/0,99	0,945/0,986	0,01	0,017
SMOTE4						
yeast-0-2-5-6_vs_3-7-8-9	0,931	0,888	0,92/0,95	0,86/0,913	0,016	0,009
Pima	0,72	0,665	0,66/0,77	0,610/0,724	0,089	0,057
yeast1	0,96	0,929	0,92/0,98	0,904/0,961	0,032	0,029
yeast-1-2-8-9_vs_7	0,946	0,901	0,92/0,97	0,879/0,942	0,025	0,033
yeast-2_vs_8	0,98	0,963	0,95/0,99	0,910/0,986	0,022	0,041

Prema rezultatima prikazanima u tablici 4.5., vrijednosti koje su postigle interpretacije SMOTE algoritma su tek malo bolje nego što se postiglo slučajnim preuzorkovanjem, ali najveća prednost nad slučajnim preuzorkovanjem je povećanje prostora i raznolikosti manjinske klase stvaranjem novih uzoraka, što klasifikatoru poboljšava učinak.

5. ZAKLJUČAK

U ovom radu cilj je bio opisati način kako rukovati problemom neuravnoteženih klasa, to je problem koji može lako nastati, ali postoje načini njegova rješavanja. U radu su opisani algoritam SMOTE i još četiri interpretacije algoritma izvedene iz algoritma SMOTE te klasifikator k-najbližih susjeda. Opisane su sličnosti i razlike između originalnog algoritma i svake pojedine interpretacije. U programskom rješenju su implemenirane te interpretacije, uz algoritam slučajnog preuzorkovanja koji je korišten za usporedbu. U eksperimentalnom djelu su interpretacije algoritma SMOTE uspoređene sa slučajnim preuzorkovanjem i iako slučajno preuzorkovanje naizgled odrađuje preuzorkovanje gotovo jednako, u pogledu izjednačavanja broja uzoraka klasa, no u poboljšanju učinkovitosti klasifikacije je ipak SMOTE uspješniji. Dobiveni rezultati u obliku F-1 mjere, geometrijske sredine istina i matrice zbunjenosti su po brojčanim vrijednostima podjednaki, ali zbog navedenih razloga slučajno preuzorkovanje ipak nije bolja opcija od SMOTE algoritma.

U literaturi postoji velik broj interpretacija SMOTE algoritma, što pruža nove mogućnosti provođenja eksperimenata koji bi se mogli nadovezati na ovaj rad, ali s više različitih varijanti, uz znatno veće skupove podataka. Moguća ideja za daljnje istraživanje je implementacija s nekim drugim klasifikatorom umjesto k-najbližeg susjeda ili nekim drugim načinom uzorkovanja, možda kombinacija preuzorkovanja i poduzorkovanja gdje bi se iz većinske klase poduzorkovanjem uklanjali šumovi, nepotrebni podaci, a istovremeno preuzorkovanjem stvarali nove uzorke manjinske klase.

LITERATURA

[1] V. Lopez, A. Fernandez, S. Garcıa, V. Palade i F. Herrera, An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Information Sciences., broj sveska 250, str. 113–141, 2013.

[2] S. Theodoridis i K. Koutroumbas, Pattern Recognition, Fourth Edition. Academic Press, 4. izdanje., SAD, 2008.

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

[4] A. Fernandez, S. Garcıa, F. Herrera i N. V. Chawla, SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary, Journal Artificial Intelligence Research., broj časopisa 61, sv. 61, str. 863–905, travanj 2018.

[5] C. Bellinger, C. Drummond i N. Japkowicz, Beyond the boundaries of SMOTE, na konferenciji Proc. ECML PKDD, str. 248–263, Kanada, 2016.

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

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

[8] M. R. Prusty, T. Jayanthi, i K. Velusamy, „Weighted-SMOTE: A modification to SMOTE for event classification in sodium cooled fast reactors", Progress in Nuclear Energy, sv. 100, str. 355–364, 2017.

[9] Knowledge Extraction based on Evolutionary Learning, dostupno na: https://sci2s.ugr.es/keel/imbalanced.php [zadnja posjeta:19.9.2020]

SAŽETAK

Pojam klasifikacija u strojnom učenju označava nešto što se može razvrstati po određenim poznatim svojstvima, odnosno grupiranje stvari u kategorije. Kada određeni uzorak u strojnom učenju ne pripada niti jednoj klasi, tada se koristi klasifikator, koji ga na temelju prepoznatog dodijeli određenoj klasi; u ovom slučaju algoritam k-najbližih susjeda i algoritam SMOTE uz njegove interpretacije. Problem neuravnoteženosti klasa je primjer problema s klasifikacijom gdje je raspodjela klasa asimetrična, a može varirati od neznatne asimetrije do itekako značajne neravnoteže. Nakon objašnjenja problema neuravnoteženosti klasa i povezivanja neuravnoteženosti s algoritmima za preuzorkovanje, odrađen je eksperiment gdje se može uvidjeti učinkovitost SMOTE algoritma nad skupovima različitih omjera neuravnoteženosti.

Ključne riječi: algoritam k-najbližih susjeda, algoritam SMOTE, klasifikacija, preuzorkovanje, problem neuravnoteženosti klasa.

ABSTRACT

Handling class imbalance problem with oversampling

The term classification in machine learning means something that can be classified to certain known properties, i.e. grouping things into categories. When a particular instance does not belong to any class, classifier is used to assign it to a particular class considering recognized propotiers; in this case k-nearest neighbors classifier was used along SMOTE algorithm and his interpretations. After explaining the problem of class imbalance and linking the imbalance with the oversampling algorithms, an experiment was performed where the efficiency of the SMOTE algorithm over sets of various imbalance ratios can be seen. The results of experiment showed that SMOTE increases the accuracy of classifiers for minority classes and provides a new approach to the oversampling imbalanced classes.

Keywords: k-nearest neighbors algorithm, SMOTE algorithm, classification, oversampling, class imbalance problem.

ŽIVOTOPIS

Ivan Zmeškal rođen je 23. travnja 1996. godine u Požegi. Svoje srednjoškolsko obrazovanje započeo je upisom Tehničke škole u Požegi, smjer tehničar za računalstvo, u vremenskom periodu 2011.-2015. godine. Interes za računala i tehnologiju 2015. godine bio je motivacija za upis preddiplomskog studija smjera računarstvo na Fakultet elektrotehnike, računarstva i informacijske tehnologije Osijek.

PRILOZI /na CD-u/

1. Završni rad u DOCX i PDF formatu
2. Ostvareno programsko rješenje
3. Skupovi korišteni u analizi