■ *Izvirni znanstveni članek*

Mirza Tupkušić, Rok Blagus

Preoptimistične ocene točnosti napovednih modelov: ilustracija na primeru skupne uporabe tehnik vzorčenja in navzkrižnega preverjanja

Povzetek. Napovedni modeli uporabljajo različne statistične metode za gradnjo pravil za uvrščanje enot v posamezno skupino na podlagi učnih podatkov. Podatki v praksi običajno niso primerni za postopek gradnje pravila, pač pa jih je potrebno predprocesirati. Tak primer so neuravnoteženi podatki, kjer dobimo slabo napovedno točnost za manjši razred, če se razvrščanja lotimo naivno. Z različnimi popravki podatkov se da izboljšati točnost napovednega modela. Toda pri tem je treba paziti, da delovanje razvrščevalca oziroma njegovo točnost pravilno ovrednotimo, saj v primeru napačnega ovrednotenja lahko pride do preoptimistične ocene točnosti napovednega modela. Ta problem podrobno razložimo in prikažemo dejavnike, ki vplivajo na preoptimizem pri ocenjevanju točnosti napovednih modelov. Rezultate ilustriramo na različnih primerih, kjer uporabljamo različne mere napovedne točnosti, različne metode za uravnoteženje podatkov ter različne načine navzkrižnega preverjanja. Rezultati lahko pomagajo razvijalcem napovednih modelov pri pravilnem ovrednotenju dejanske napovedne moči modela oziroma pri razumevanju in kritičnemu ovrednotenju, ali je bila ocena napovedne moči modela izvedena pravilno ali pa so rezultati zaradi napačne izvedbe preoptimistični.

Ključne besede: napovedni model; neuravnoteženi podatki; navzkrižno preverjanje; preprileganje.

Over-optimistic Assessment of the Performance of Prediction Models: An Illustration Based on the Joint Use of Sampling Techniques and Cross-Validation

Abstract. Prediction models use various statistical methods for building classification rules to classify units into pre-specified groups based on the learning data. In practice, the data are often not suitable for the chosen procedure and they need to be pre-processed before training the classifier. An important example are imbalanced data where the naïve approach can lead to poor accuracy for the minority class. Many data augmentation approaches have been developed to alleviate this issue. However, when using these techniques, one needs to be careful to correctly evaluate the performance of the classifier in terms of its predictive accuracy, because incorrect evaluation can lead to an overly optimistic estimate of the classifier's performance. We explain in detail why this happens and showcase the different contributing factors. The results are illustrated using various performance measures, various data augmentation techniques, and various cross-validation techniques. Our results can help the developers of prediction models to correctly evaluate predictive ability of the derived model, as well as to understand and critically appraise whether the predictive ability of the model was correctly estimated or the evaluation was too optimistic.

Key words: prediction models; cross-validation; rare events; overfitting.

■ **Infor Med Slov** 2022; 27(1-2): 1-13

Institucije avtorjev / Authors' institutions: Medicinska fakulteta, Univerza v Ljubljani (MT, RB); Fakulteta za šport, Univerza v Ljubljani (RB); FAMNIT, Univerza na Primorskem, Koper (RB).

Kontaktna oseba / Contact person: izr. prof. dr. Rok Blagus, MF, IBMI, Vrazov trg 2, 1000 Ljubljana. E-pošta / E-mail: rok.blagus@mf.uni-lj.si.

Prispelo / Received: 28. 11. 2022. Sprejeto / Accepted: 24. 12. 2022.

Uvod

Napovedovanje lahko definiramo kot problem ocenjevanja in odločanja na podlagi znanih podatkov.[1] V vsakdanjem življenju se vseskozi srečujemo z nalogami ali vprašanji, na katera želimo odgovoriti čim bolj pravilno. Človeški odgovori so subjektivni, zato so lahko povsem napačni. Zaradi tega postajajo računalniško izdelani napovedni modeli (angl. *prediction models*) vse bolj priljubljeni, še zlasti na področju medicine,[2-7] pogosto pa se uporabljajo tudi na drugih področjih, npr. v trženju in strojništvu.[8] V kliničnih raziskavah nas pogosto zanima verjetnost ali napoved, da bo pacient zbolel za določeno boleznijo, kako se bo odzval na zdravljenje ipd. Podobno velja na drugih področjih, npr. za odliv strank iz podjetja ali čas do okvare stroja. V medicini so napovedni modeli posebej pomembni v okviru presejalnih programov za zgodnje odkrivanje določene bolezni,[9] na primer raka. Gradnjo oziroma razvoj napovednih modelov razdelimo v tri faze:

- faza 1: priprava podatkov;
- faza 2: gradnja/učenje razvrščevalca;
- faza 3: preverjanje točnosti razvrščevalca.

V prvi fazi pripravimo podatke za izgradnjo modela oziroma za učenje razvrščevalca (angl. *classifier*). Gradnjo razvrščevalca pogosto otežuje narava zbranih podatkov: v podatkih se lahko pojavljajo manjkajoče vrednosti, napake, osamelci, veliko število spremenljivk itd. V tem članku se bomo osredotočili na pogost problem, ko so podatki v dveh razredih neuravnoteženi (angl. *unbalanced data*),[10-12] se pa podobne težave pojavijo tudi v primeru nadomeščanja manjkajočih podatkov, izločanja osamelcev iz podatkov, izbire spremenljivk za analizo ipd. O neuravnoteženih podatkih govorimo, ko se število enot med razredoma razlikuje. Na področju medicine je običajno število pacientov z določeno boleznijo veliko manjše kot število zdravih ljudi; podobno je število strank, ki ostanejo v podjetju, praviloma veliko večje od števila strank, ki podjetje zapustijo. Razred z večjim številom enot imenujemo večinski razred (angl. *majority class*), razred z manjšim številom enot pa manjšinski razred (angl. *minority class*).

Gradnja napovednega modela na podlagi neuravnoteženih podatkov je problematična predvsem zaradi slabe napovedne točnosti v manjšinskem razredu.[12] Preprosto povedano, do tega pride, ker se razvrščevalcu, ki želi minimizirati celotno napako, izplača osredotočiti na večinski razred, posledica pa je slaba točnost za manjšinski razred. Mogoča rešitev tega problema, ki dokazano deluje dobro, so različne metode za uravnoteženje razredov.[10,13-17] Tovrstne metode izboljšajo napovedano točnost modela v manjšinskem razredu tako, da zmanjšajo neravnotežje v podatkih ali pa da celo izenačijo število enot v večjem in manjšem razredu.[13] K temu lahko pristopimo na več načinov. Razreda lahko uravnotežimo z večanjem števila enot v manjšinskem razredu (angl. *oversampling*), zmanjševanjem števila enot v večinskem razredu (angl. *undersampling*) ali kombinacijo teh dveh pristopov.[10,13] Ko smo končali prvo fazo uravnoteženja podatkov, lahko pristopimo k fazi učenja. Enote, za katere poznamo pripadnost razredu, uporabimo za izgradnjo modela ali razvrščevalca, na podlagi katerega bomo uvrščali nove enote.[1] Obstaja množica različnih razvrščevalcev.[18] V ilustraciji bomo uporabili grebensko regresijo (angl. *ridge regression*),[19-23] so pa ugotovitve splošne in v podobni meri veljajo tudi za druge razvrščevalce. Ko razvrščevalec izgradimo in s tem končamo drugo fazo, bi seveda radi ovrednotili njegovo točnost oziroma ocenili njegovo napako. Na voljo so različne mere točnosti.[24] Pri izbiri ustrezne mere točnosti moramo biti previdni, še posebej, ko imamo opraviti z neuravnoteženimi podatki.[25] V ilustraciji bomo uporabljali ploščino pod krivuljo ROC[26] (mero AUC),[24,27] točnost za manjšinski in večinski razred, njuno geometrijsko sredino (G-povprečje)[28] ter mero F_1, ki se pogosto uporabljajo v tem kontekstu. Idealno bi se točnost razvrščevalca ovrednotila na (veliki) neodvisni tesni množici,[1,18] ki pa v praksi pogosto ni dostopna. Za preverjanje točnosti razvrščevalca se zato pogosto uporabi navzkrižno preverjanje s k pregibi (angl. *k-fold cross-validation* – CV) oziroma njegova različica navzkrižno preverjanje z izpustitvijo ene enote (angl. *leave-one-out CV* – LOOCV), za katero velja $k = u$, kjer je u velikost učne množice. Pisali bomo $u = m + v$, kjer je m število enot v manjšinskem razredu, v število enot v večinskem razredu in velja $m < v$.

Problem napačnega ovrednotenja točnosti napovednih modelov v različnih kontekstih (npr. v kontekstu izbire spremenljivk v prvi fazi) je znan.[29,30] Raziskave kažejo na nujnost pravilnega ovrednotenja točnosti delovanja razvrščevalcev: v primeru napačnega ovrednotenja je delovanje napovednega modela lahko slabše ali boljše, kot je predstavljeno. V članku bomo ilustrirali, kakšen je vpliv napačne uporabe navzkrižnega preverjanja na oceno točnosti napovednega modela ob uporabi različnih pristopov za uravnoteženje podatkov. Pokazali bomo, da napačna uporaba navzkrižnega preverjanja vodi do precenjenih mer točnosti, in prikazali različne dejavnike, ki na to vplivajo. Rezultati so pomembni, ker je bilo doslej objavljenih precej člankov, kjer je

bilo navzkrižno preverjanje izvedeno napačno (npr. v kombinaciji s prevzorčenjem[31-33]), objavljene mere točnosti pa so posledično preoptimistične. Podobno tematiko smo že obravnavali,[34] s to razliko, da se tokrat bolj osredotočamo na oris in pomembnost posameznih dejavnikov, ki vplivajo na preoptimizem zaradi napačne uporabe navzkrižnega preverjanja, manj pa na pojasnjevanje razlogov, zakaj do tega pride. V pričujočem članku obravnavamo tudi različne mere točnosti, ki jih v prvotnem[34] nismo.

V nadaljevanju najprej predstavimo uporabljene metode, kjer na kratko orišemo različne pristope za uravnoteženje podatkov, uporabljeni razvrščevalec in mere točnosti. Sledi ilustracija, kjer prikažemo vpliv različnih dejavnikov na precenjenost ocene točnosti napovednega modela. Članek zaključimo s kratkim povzetkom ključnih ugotovitev.

Metodologija

V nadaljevanju bolj podrobno predstavljamo metode, ki jih kasneje v ilustraciji uporabljamo v posameznih fazah razvoja napovednega modela.

Metode za uravnoteženje razredov

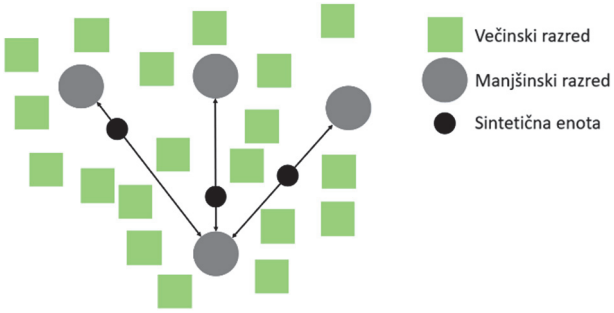
V ilustraciji bomo uporabili tri različne metode uravnoteženja razredov. Pri naključnem prevzorčenju (angl. *random oversampling*) naključno s ponavljanjem izberemo $n \leq v - m$ enot iz manjšega razreda, izbrane enote kopiramo in jih dodamo v nabor podatkov.[35] Tako se manjšinski razred poveča za n neinformativnih enot, popolnih kopij prvotnih enot iz manjšinskega razreda. Metoda prevzorčenja torej uravnoteži razrede z znanimi enotami, zato uravnoteženi podatki ne nosijo nobene dodatne informacije kot izvorni, so le (umetno) uravnoteženi. Posledično so lahko ob uporabi napačnega pristopa navzkrižnega preverjanja iste enote uporabljene v fazi učenja in preverjanja točnosti razvrščevalca in zato zaradi problema preprileganja (angl. *overfitting*[18]) dobimo preoptimistično oceno točnosti. O preprileganju na primer govorimo, ko je v fazi preverjanja točnosti razvrščevalca vrednost AUC velika, a je uspešnost razvrščevalca na neznanih (novih) podatkih mnogo slabša.

Pri naključnem podvzorčenju (angl. *random undersampling*) naključno (običajno brez ponavljanja) izberemo $n \leq m$ enot iz večinskega razreda.[12] Izbrane enote večinskega razreda združimo z enotami manjšinskega razreda v novi podatkovni okvir. Na tak način ostane število enot v manjšinskem razredu nespremenjeno, število enot v večinskem razredu pa je za $v - n$ manjše. Posledično v fazi učenja razvrščevalca lahko izpustimo pomembno informacijo, ki se nanaša na večinski razred, kar se lahko odrazi v slabši napovedni točnosti v večinskem razredu. Izgubo informacije se lahko omili z večkratnim naključnim podvzorčenjem, kar lahko bistveno poveča točnost razvrščevalca,[12] a za namen naše analize to ni zelo pomembno, zato tega ne bomo podrobneje obravnavali. Metoda podvzorčenja uravnoteži razrede z izgubo informacije, zato so uravnoteženi podatki manj informativni kot izvorni. Toda ker nobena enota ni podvojena, do problema preprileganja, ki nastopi pri naključnem prevzorčenju, pri naključnem podvzorčenju ni. Bi pa do podobnega problema vseeno prišlo, če so enote, ki se jih obdrži v večinskem razredu, izbere sistematično[36,37] (s tem se podrobneje ne bomo ukvarjali).

SMOTE (angl. *Synthetic Minority Oversampling Technique*) je metoda kjer se hkrati podvzorči in prevzorči, pri čemer se pri prevzorčenju tvorijo sintetični podatki za manjšinski razred (v primarni definiciji je metoda SMOTE vezana le na sintetično prevzorčenje, vendar obstaja več izvedb, med katerimi se bomo osredotočili na kombinacijo prevzorčenja in podvzorčenja).[13] S tvorjenjem sintetičnih enot metoda SMOTE pomaga pri premagovanju problema preprileganja, a ga ne odpravi povsem. V manjšinskem razredu metoda naključno izbere eno enoto x_r (angl. *random minority*), nato pa poišče njenih g najbližjih sosedov x_{gNN} (angl. g*-nearest neighbours*[38]). Nato izračuna razdaljo med izbrano enoto in g najbližjimi sosedi, na kateri naključno tvori eno ali več sintetičnih enot x_{new_i},

$$x_{new_i} = x_{r_i} + rand(0,1)\left(x_{gNN_i} - x_{r_i}\right) \; i = 1, \dots, l \quad (1)$$

kjer $rand(0,1)$ označuje naključno vrednost iz enakomerne porazdelitve na intervalu (0,1). Tako nove enote niso identične obstoječim, pač pa so njihove linearne kombinacije (slika 1**Error! Reference source not found.**). Med tvorbo novih sintetičnih enot metoda SMOTE lahko izvaja podvzorčenje.[13] Postopek se konča, ko dosežemo želeno (ne)ravnotežje števila enot v manjšem in večjem razredu (običajno podatke povsem uravnotežimo). Ker nove (sintetične) enote niso popolnoma neodvisne od osnovnih enot (saj so tvorjene z uporabo informacij o osnovnih enotah), lahko seveda pride do problema preprileganja; o tem več kasneje.

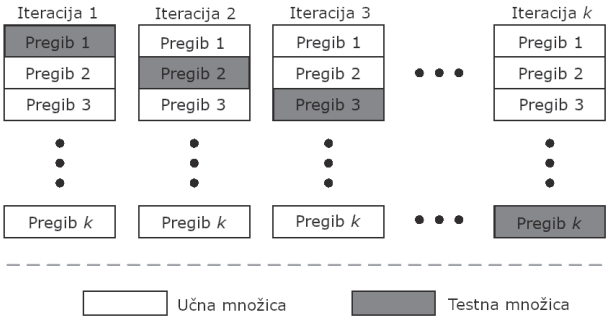


Slika 1 Načelo delovanja metode SMOTE (angl. *Synthetic Minority Oversampling Technique*).

Navzkrižno preverjanje s k pregibi

Navzkrižno preverjanje s k pregibi je ena izmed metod, ki jih lahko uporabimo za oceno razvrstitvene točnosti.[18] V navzkrižnem preverjanju s k pregibi je podatkovni okvir razdeljen na k podmnožic (angl. *folds*): $k-1$ podmnožic uporabljamo za gradnjo razvrščevalca, eno podmnožica pa za oceno njegove točnosti. Podmnožice ustvarimo tako, da je število enot v vsaki podmnožici enako in je delež enot manjšinskega in večinskega razreda v vsaki podmnožici enak kot v osnovni množici.

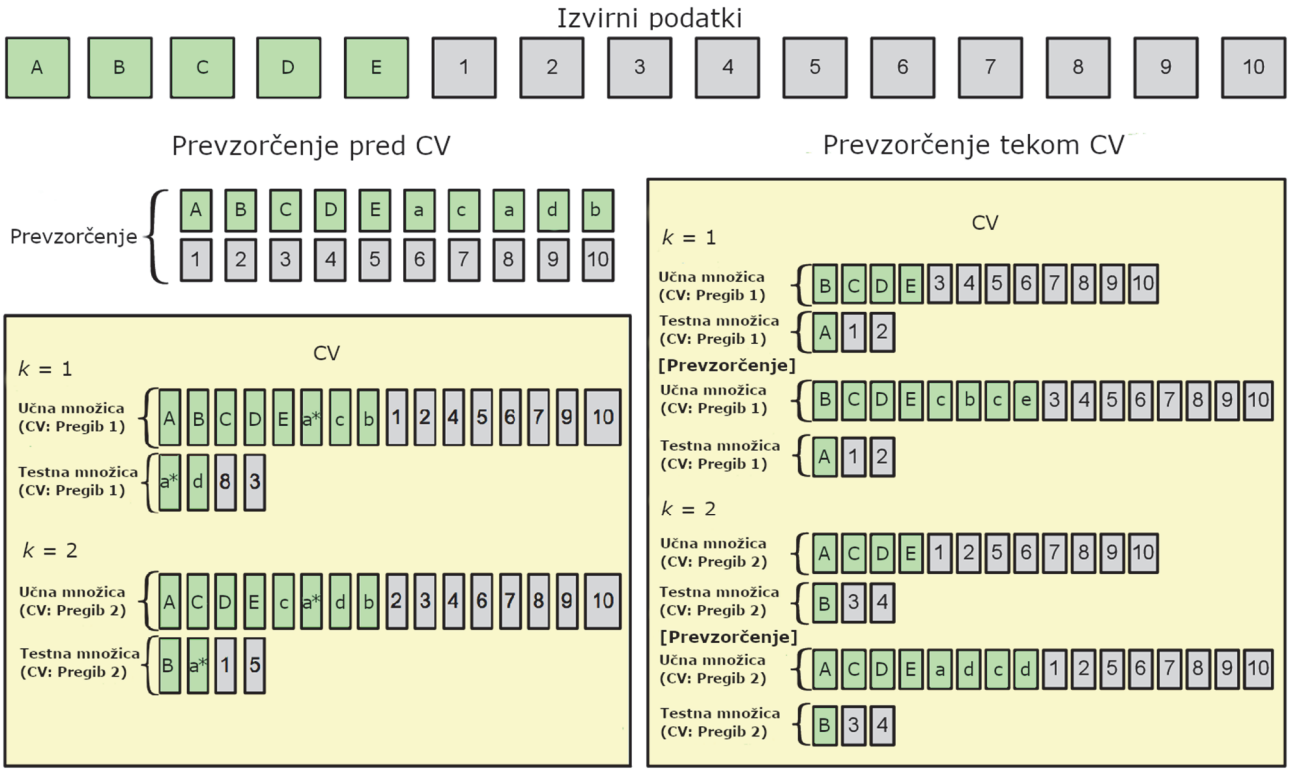
Iterativni postopek ponovimo k-krat, tako je vsaka izmed k podmnožic enkrat uporabljena kot testna množica (slika 2).[24] Navzkrižno preverjanje z izpustitvijo ene enote je skrajna različica navzkrižnega preverjanja s k pregibi: $u-1$ enot uporabljamo za gradnjo razvrščevalca, eno enoto pa uporabimo za preverjanje njegove točnosti. Iterativni postopek ponovimo u-krat, tako je vsaka enota enkrat uporabljena kot testna množica. Izvedba z izpustitvijo ene enote je seveda računsko in časovno najbolj zahtevna.



Slika 2 Navzkrižno preverjanje s k pregibi.

Pri uporabi navzkrižnega preverjanja imamo dve možnosti, kako izračunati neko mero točnosti. Prva možnost je, da točnost izračunamo za vsak pregib posebej in potem povprečimo k tako dobljenih ocen. Druga možnost je, da vse napovedi združimo in mero točnosti izračunamo zgolj enkrat. Katera izbira je pravilna je odvisno med drugim tudi od uporabljene mere točnosti in je še vedno predmet razprave.[39] Zaradi primerljivosti med različnimi oblikami navzkrižnega preverjanja bomo uporabili drugo možnost (ki je v primeru LOOCV edina možnost, če želimo oceniti AUC), za katero je sicer znano, da vodi do pristranske ocene AUC in pravilne ocene mere F_1;[39] s podrobno primerjavo obeh pristopov se ne bomo ukvarjali.

Pri skupni izvedbi navzkrižnega preverjanja in ene izmed metod uravnoteženja razredov moramo paziti, da oba postopka izvedemo pravilno. Če najprej uravnotežimo podatke, potem pa uporabimo navzkrižno preverjanje, smo slednje izvedli napačno (slika 3). Navzkrižno preverjanje je pravilno, če proces uravnoteženja podatkov izvedemo znotraj postopka navzkrižnega preverjanja. V pravilni izvedbi navzkrižnega preverjanja metode uravnoteženja razredov uporabljamo samo na učni množici, kar pomeni, da moramo uravnoteženje razredov k-krat (oziroma v primeru LOOCV u-krat) ponoviti (slika 3).



Slika 3 Napačna (levo) in pravilna (desno) izvedba navzkrižnega preverjanja s k pregibi in naključnega prevzorčenja.

Razvrščevalec

Grebenska regresija[19] je statistična metoda, s katero lahko izboljšamo točnost napovedi z zmanjšanjem ocen parametrov (t. i. krčenjem, angl. *shrinkage*).[23] Z dodajanjem penalizacijske funkcije (angl. *penalising function*) spreminja oziroma zmanjša ocenjeno vrednost regresijskega koeficienta, s čimer poskušamo zmanjšati problem preprileganja. Splošni regresijski model lahko zapišemo v matrični obliki kot $\boldsymbol{Y} = \boldsymbol{\beta X} + \boldsymbol{e}$, kjer so $\boldsymbol{Y}$ izidi, $\boldsymbol{X}$ napovedne spremenljivke, $\boldsymbol{\beta}$ regresijski koeficienti, $\boldsymbol{e}$ pa naključne napake.[18] Regresijske koeficiente z grebensko regresijo dobimo tako, da rešimo optimizacijski problem

$$\beta^{ridge} = argmin_{\beta}\left[\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2\right] \quad (2),$$

kjer je λ uglaševalski parameter. Opazimo, da za $\lambda = 0$ dobimo enako rešitev, kot če uporabljamo standardna orodja (denimo metodo največjega verjeta[40]), medtem ko za $\lambda = \infty$ vse ocene postavimo na nič. Parameter λ se običajno določi s navzkrižnim preverjanjem.[41] Če želimo zgornji model uporabiti za (binarno) razvrščanje, moramo enotam določiti vrednost izidov, $\boldsymbol{Y}$. V našem primeru bomo enotam iz manjšinskega razreda določili vrednost 0, enotam iz večinskega razreda pa vrednost 1 (lahko bi uporabili tudi obratno definicijo, rezultati pa bili enaki). Ko izberemo parameter λ in pridobimo ocene regresijskih koeficientov, lahko na podlagi teh ocen izračunamo verjetnost dogodka, ki jo označimo s $\hat{p}$. Za izračun nekaterih mer točnosti (npr. AUC) lahko $\hat{p}$ uporabimo neposredno, medtem, ko moramo za izračun drugih mer (npr. napovedne točnosti) verjetnostno napoved spremeniti v napoved vrednosti 0 ali 1 (ki jo označimo z $\hat{y}$), za kar lahko uporabimo pravilo

$$\hat{y} = \begin{cases} 1 \text{ če } \hat{p} > \tau \\ 0 \text{ če } \hat{p} < \tau \end{cases} \quad (3),$$

kjer je τ prag za uvrščanje. Če velja $\hat{p} = \tau$, enoto naključno uvrstimo v enega izmed razredov. Ker (podobno kot v običajni logistični regresiji) velja, da so ocenjene verjetnosti zgoščene okrog neravnotežja v učni množici, naivna uporaba $\tau = 0{,}5$ za neuravnotežene podatke praviloma ni ustrezna.[12] V ilustraciji bomo zato kot prag za uvrščanje uporabljali delež dogodkov na (uravnoteženi) učni množici.

Mere razvrstitvene točnosti

Za oceno točnosti razvrščanja bomo izračunali ploščino pod krivuljo ROC (angl. *area under the curve* – *AUC*),[24] G-povprečje (angl. G-*mean*)

$$G = \sqrt{\left(\frac{TP}{TP+FN}\right)\left(\frac{TN}{TN+FP}\right)} = \sqrt{PA_1 PA_2} \quad (4).$$

kjer je TP število pravilno uvrščenih enot iz manjšinskega razreda, FN število napačno uvrščenih enot iz večinskega razreda, TN število pravilno uvrščenih enot iz večinskega razreda, FP število napačno uvrščenih enot iz manjšinskega razreda, $PA_1 = \frac{TP}{TP+FN}$ in $PA_2 = \frac{TN}{TN+FP}$ pa sta točnost za manjšinski in večinski razred, ter mero F_1

$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (5).$$

Pri izračunu AUC bomo uporabljali verjetnostno napoved $\hat{p}$, za izračun ostalih mer pa bomo uporabljali $\hat{y}$, ki ga dobimo, kot je pojasnjeno zgoraj.

Implementacija metod v programskem jeziku R

Za izvedbo metode naključnega prevzorčenja uporabljamo funkcijo **upSample(x, y, list = FALSE, yname = class)** iz paketa **caret**, kjer je **x** matrika ali podatkovni okvir vrednosti enot za vsako spremenljivko, **y** indikatorska spremenljivka, ki določi pripadnost, in argument **yname** določi ime spremenljivke, ki nam pove pripadnost posamezne enote v izhodu funkcije. Funkcija dela enako, kot je opisano v razdelku o metodah za uravnoteženje razredov: razreda uravnotežimo z naključnim dodajanjem enot manjšinskega razreda s ponavljanjem v nabor podatkov. Za izvedbo naključnega podvzorčenja uporabljamo funkcijo **downSample(x, y, list = FALSE, yname = class)**, iz paketa **caret**. Funkcija ima enake argumente kot funkcija za naključno prevzorčenje. Tudi ta funkcija deluje enako, kot je opisano zgoraj. Za izvedbo metode SMOTE uporabljamo funkcijo **SMOTE(formula, data, perc.over = 100, k = 5, perc.under = 200, …)** iz paketa **DMwR**. Z argumentom **formula** zapišemo napovedni model, z argumentom **data** podamo orginalni neuravnoteženi podatkovni okvir, z argumentom **perc.over** definiramo število dodanih sintetičnih enot, z argumentom **k** definiramo število najbližjih sosedov, z argumentom **perc.under** pa definiramo število izbrisanih enot v večinskem razredu. Funkcija vrednosti **perc.over** in **perc.under** deli s 100, dobljeni vrednosti pa določita, koliko novih enot v vsaki ponovitvi dodamo in odstranimo. Za vpogled v ostale parametre, ki jih lahko nastavimo v funkciji, priporočamo pregled dokumentacije paketa **DMwR**. Za ogled izvorne kode priporočamo ogled funkcije **SMOTE(form, data, perc.over = 200, k = 5, perc.under = 200, learner = NULL, …)** in **smote.exs(data, tgt, N, k)** na spletu (https://rdrr.io/cran/DMwR/src/R/smote.R). Funkcijo smo uporabili na dva načina: pri prvem načinu smo uporabili **perc.over = 100**, **perc.under = 200**; pri drugem pa **perc.over = 400**, **perc.under = 100**.

Za učenje razvrščevalca, ki smo ga predstavili istoimenskem razdelku, uporabljamo funkcijo **glmnet(x, y, alpha = 0, lambda, …)** iz paketa **glmnet**. Z argumentom **x** definiramo matriko neodvisnih spremenljivk, z **y** definiramo odzivno (indikatorsko) spremenljivko in z argumentom **lambda** nastavimo vrednost parametra λ. Optimalno vrednost parametra λ določimo s pomočjo funkcije **cv.glmnet(x, y, alpha = 0, nfolds = 10, …)**, ki določi optimalno vrednost na podlagi navzkrižnega preverjanja z 10 pregibi; ostali vhodni argumenti, **x**, **y** in **alpha**, so enaki kot pri funkciji **glmnet**. Za izračun napovedi uporabimo funkcijo **predict(ovject, s, newx, type, …)** iz paketa **stats**. Z argumentom **object** določimo model, za katerega želimo izračunati napovedi, argument **s** določa optimalno vrednost λ, argument **newx** določa vrednosti napovednih spremenljivk in z argumentom **type** določimo tip izhoda, ki ga vrne funkcija (v našem primeru je to ocenjena verjetnost dogodka). AUC izračunamo s pomočjo funkcije **auc()** iz paketa **pROC**, meri G in F_1 mero pa izračunamo po zgoraj predstavljeni definiciji. Navzkrižno preverjanje s k pregibi in z izpustitvijo ene enote smo sprogramirali sami, kot je opisano v razdelku o navzkrižnem preverjanju.

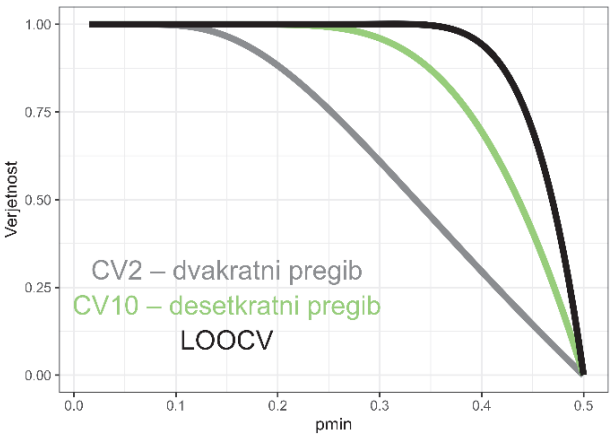
Ilustracija

Za primer naključnega prevzorčenja lahko izračunamo verjetnost, da je enota iz manjšega razreda hkrati vključena v učno in testno množico, če navzkrižno preverjanje izvedemo napačno (če navzkrižno preverjanje izvedemo pravilno, je ta verjetnost seveda nič, enako pa velja tudi za primer, ko uporabimo naključno podvzorčenje, tudi če navzkrižno preverjanje izvedemo napačno, kar smo že pojasnili). Verjetnost, da je ista enota vključena v učno in testno množico, je odvisna od števila enot v podatkovnem okvirju, deleža enot, vključenih v testno množico p_{test}, in deleža enot v manjšinski množici $p_{min} = m/u$:

$$P = 1 - \frac{\binom{u - v/m}{up_{test} - v/m}}{\binom{u-1}{up_{test}-1}} \quad (6).$$

Z manjšanjem deleža enot v manjši množici p_{min} se verjetnost povečuje (slika 4). Če imamo opravka z

neuravnoteženimi podatki (npr. $p_{min} = 0.1$), metoda prevzorčenja večkrat v podatkovni okvir doda veliko obstoječih enot, zaradi česar se bolj pogosto zgodi, da imamo pri napačnem navzkrižnem preverjanju v učni in testni množici vključene iste enote. Če je delež enot v manjši množici blizu vrednosti 0,5, metoda prevzorčenja v podatkovni okvir doda manjše število podvojenih enot, posledično se redkeje zgodi, da je v primeru napačnega navzkrižnega preverjanja ena enota vključena v učno in testno množico hkrati. Z manjšanjem deleža enot v testni množici p_{test} se verjetnost povečuje (slika 4). To pomeni, da se verjetnost povečuje z večanjem števila podmnožic k pri navzkrižnem preverjanju. Posledično ima navzkrižno preverjanje z izpustitvijo ene enote, ki predstavlja skrajni primer navzkrižnega preverjanja s k pregibi $(k = u)$, pri vsakem številu enot n in p_{min} vedno največjo verjetnost. Na omenjeno verjetnost lahko vplivamo tudi s številom enot v podatkih, pri čemer se z večanjem števila enot se verjetnost zmanjšuje. Ko za prevzorčenje uporabljamo metodo SMOTE, je verjetnost, da bo ista enota vključena v testni in učni množici, seveda enaka nič, vendar pa so lahko v primeru napačne izvedbe navzkrižnega preverjanja v testni množici vključene *podobne* enote kot v učni. Spomnimo, da z metodo SMOTE ne ustvarjamo kopij enot iz manjšinskega razreda, temveč njihove linearne kombinacije, te linearne kombinacije (novi sintetični podatki) pa vsebujejo tudi informacijo, ki je vključena v osnovnih podatkih, zato ti novi podatki nikakor niso neodvisni od prvotnih.



Slika 4 Verjetnost, da je vsaj ena enota vključena v učno in testno množico, v odvisnosti od deleža enot v manjšinskem razredu (p_{min}).

Čeprav sintetične enote niso identične prvotnim, je torej v primeru napačne izvedbe navzkrižnega preverjanja v testni množici prisotna informacija, ki smo jo dobili neposredno iz učne množice, kar lahko vodi do preprileganja in preoptimistične ocene. V nadaljevanju podrobneje ilustriramo, kakšen je vpliv napačne izvedbe navzkrižnega preverjanja na (pre)optimistično oceno različnih mer točnosti.

V ilustraciji uporabljamo podatke, ki smo jih simulirali neodvisno iz standardne normalne porazdelitve za vse enote iz učne množice; odločitev o uporabi konkretne porazdelitve ni bistvena, podobne ugotovitve bi veljale tudi za druge porazdelitve. V simulaciji smo spreminjali število neodvisno generiranih spremenljivk p, število enot N, delež enot v manjšem razredu p_{min} in delež enot v testni množici p_{test} (preko različne izbire števila pregibov $k = 2, 10, u$ v navzkrižnem preverjanju); podatke smo simulirali stokrat in rezultati, o katerih poročamo, so povprečeni čez 100 ponovitev. Naj poudarimo, da simuliramo na način, da med razredoma dejansko ni razlike: točna vrednost AUC je enaka 0,5, $PA_1 + PA_2 = 1$ in zato $G = \sqrt{PA_1(1 - PA_1)} = \sqrt{PA_2(1 - PA_2)}$ in $F_1 = \frac{2PA_1 p_{min}}{PA_1 + p_{min}} = \frac{2(1-PA_2)p_{min}}{1-PA_2+p_{min}}$. Če dobimo vrednosti, ki odstopajo od pravih, točnosti napovednega modela nismo pravilno ovrednotili: če so ocene večje od pravih, smo delovanje napovednega modela precenili, če so manjše, pa podcenili. Če bi med razredoma obstajale razlike, bi bili zaključki podobni predstavljenim.

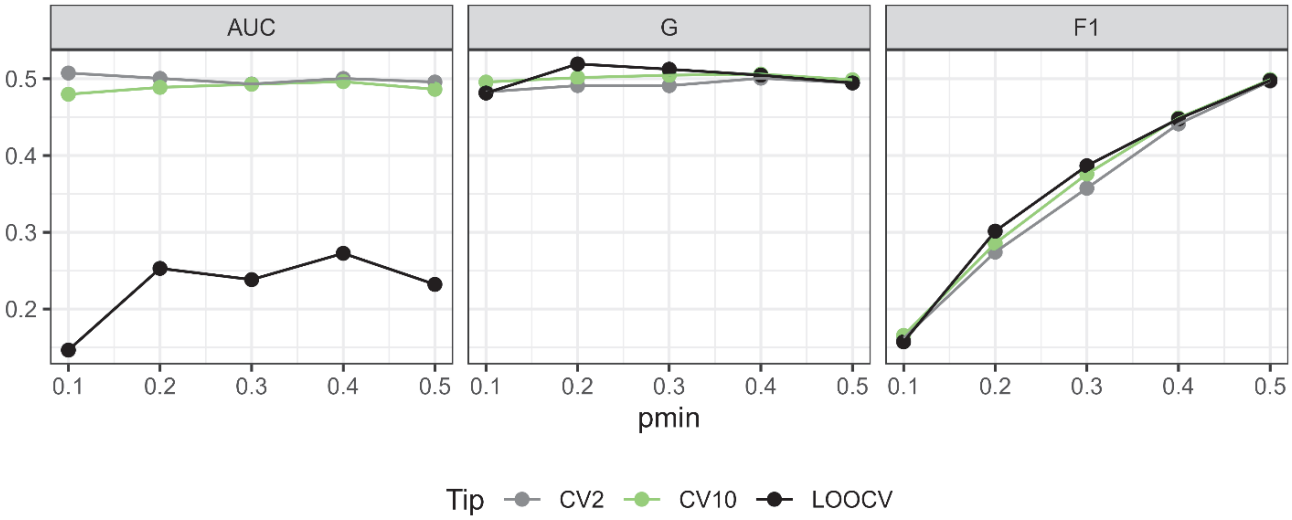
Naj na tem mestu opomnimo, da smo pri izračunu pravih mer točnosti za našo ilustracijo predpostavljali zgolj, da je razvrščevalec neinformativen, torej tak, za katerega velja $PA_1 + PA_2 = 1$. To je (malenkost) bolj splošna zahteva, kot če bi bil razvrščevalec naključen, torej tak, za katerega velja $PA_1 = PA_2 = 1/2$. Opazimo lahko, da je vsak naključen razvrščevalec tudi neinformativen, ni pa vsak neinformativen razvrščevalec tudi naključen. Ilustrirajmo to na primeru, ko se o razredu odločimo glede na met kovanca. V prvem primeru denimo, da je kovanec pošten (verjetnost grba je $1/2$), v drugem pa, da je verjetnost grba enaka $\pi \neq 1/2$. V prvem primeru bo seveda v povprečju (!) veljalo (kot vemo iz osnov verjetnosti) $PA_1 = PA_2 = 1/2$, v drugem pa $PA_1 = (\pi m)/m = \pi$ in $PA_2 = (1 - \pi)v/v = 1 - \pi$. V obeh primerih gre za neinformativen razvrščevalec, vendar pa je zgolj prvi razvrščevalec tudi naključen.

Ignoriranje problema neuravnoteženih razredov

Najprej bomo prikazali, kaj se zgodi, ko zanemarimo problem neuravnoteženih razredov, torej izpustimo prvo fazo gradnje napovednega modela. Ilustracija se nanaša na primer, ko spreminjamo delež enot v manjšem razredu: $p_{min} = 0{,}1, 0{,}2, \ldots, 0{,}5$. Ostali

parametri so nastavljeni na $N = 300$ in $p = 1000$. Rezultati so prikazani na sliki 5.

Vrednosti G-povprečja in mere F_1 so enake pravim vrednostim, saj v tretji fazi nismo naredili nobene napake, zaradi katere bi prišlo do preoptimističnih ocen. Za razliko od G-povprečja in mereF_1 pa so ocenjene vrednosti AUC ob uporabi LOOCV premajhne, do česar pride zaradi napake, ki smo jo naredili, ko smo združevali ocene različnih pregibov. Ko smo združili napovedane verjetnosti v posameznih pregibih, smo namreč združili nezdružljive ocene: združili smo ocene, ki so bile pridobljene na učnih množicah z različnimi neravnotežji (neravnotežje je seveda drugačno, ko izpustimo enoto iz manjšinskega oziroma večinskega razreda). Ko uporabljamo $k = 2$ in $k = 10$, do tega problema seveda ne pride, ker pregibe ustvarjamo tako, da je neravnotežje ves čas enako. Če bi pri izračunu AUC uporabljali napovedani razred ($\hat{y}$) in ne ocenjene verjetnosti ($\hat{p}$), bi bila tudi ob uporabi LOOCV za vsak p_{min} AUC pravilno ovrednotena (bila bi enaka 0.5). To je tudi razlog, zakaj sta G-povprečje in mera F_1 pravilno ocenjena tudi, če uporabimo LOOCV.

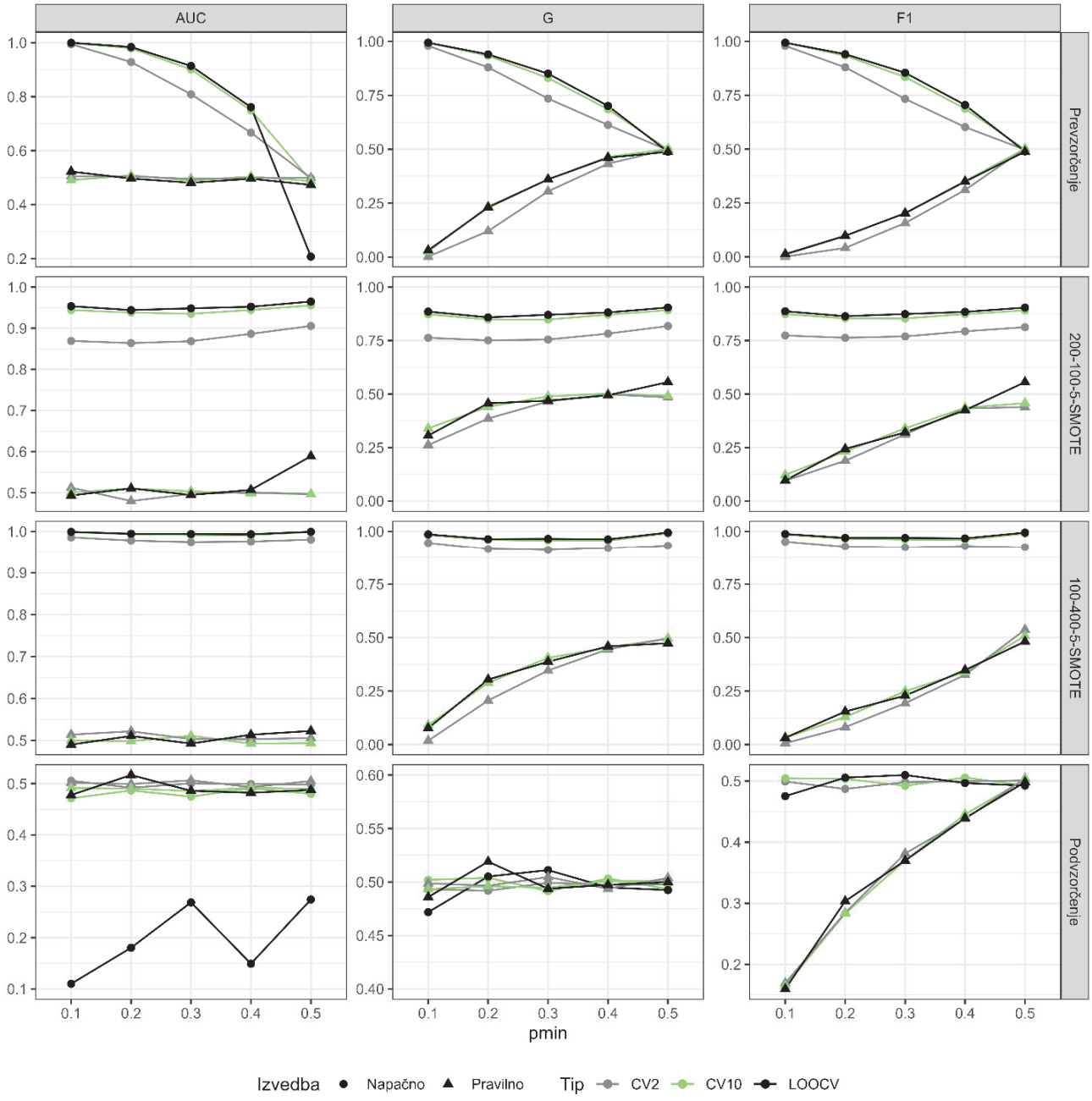


Slika 5 Navzkrižno preverjana točnost razvrščevalca za različna neravnotežja v podatkih (p_{min}).

Pojasnimo ta problem bolj podrobno na primeru, ko velja $\lambda = \infty$ (dejansko so bile pri nas ocenjene vrednosti za λ zelo velike, kar je pričakovano, saj to pomeni, da model pravilno ugotovi, da spremenljivke niso pomembne za pojasnjevanje izida). V tem primeru je namreč ocenjena verjetnost točno enaka deležu dogodkov v učni množici.[23] To pomeni, da je enaka $\hat{p}_m = v/(v-1)$ za vse enote iz manjšinskega razreda (ki ga kodiramo z vrednostjo 0 – nedogodek) in $\hat{p}_v = (v-1)/(u-1)$ za vse enote iz večinskega razreda (ki ga kodiramo z 1 – dogodek). Opazimo, da velja $\hat{p}_m > \hat{p}_v$: vse enote iz manjšega razreda so rangirane višje od enot iz večjega razreda (imajo večjo verjetnost, da spadajo v večinski razred), zato je AUC enaka nič (spomnimo se interpretacije AUC: to je verjetnost, da bo razvrščevalec naključno izbran dogodek rangiral višje kot nedogodek[24]). Če ocenjeno verjetnost spremenimo v razred, opazimo, da vsako enoto popolnoma naključno uvrstimo v enega izmed razredov (enačba 3), posledično je AUC (v povprečju!) enaka 0,5 in do problema podcenjene AUC ne pride.

Odvisnost preoptimistične ocene od deleža enot v manjšem razredu

Vsi parametri se enaki kot v prejšnjem primeru $(N = 300, p = 1000, p_{min} = 0{,}1, 0{,}2, \ldots, 0{,}5)$, le da tu uporabimo eno od treh predstavljenih metod za uravnoteženje podatkov ter primerjamo rezultate pravilne in napačne uporabe navzkrižnega preverjanja. Na sliki 6 smo prikazali razliko med pravilno in napačno izvedbo navzkrižnega preverjanja s k pregibi ($k = 2, 10$ in u), ob uporabi različnih metod za uravnoteženje podatkov, pri različni vrednosti deleža enot v manjšinskem razredu. V primeru pravilne izvedbe navzkrižnega preverjanja so vse mere pravilno ocenjene. Zanimivo, opazimo, da do podcenjenega AUC v primeru uporabe LOOCV v tem primeru ne pride. V kolikor uporabimo napačno navzkrižno preverjanje v kombinaciji s podvzorčenjem, potem sta AUC in G-povprečje ocenjena pravilno; izjema je AUC ob uporabi LOOCV, o razlogih za to pa smo govorili že v prejšnjem primeru.

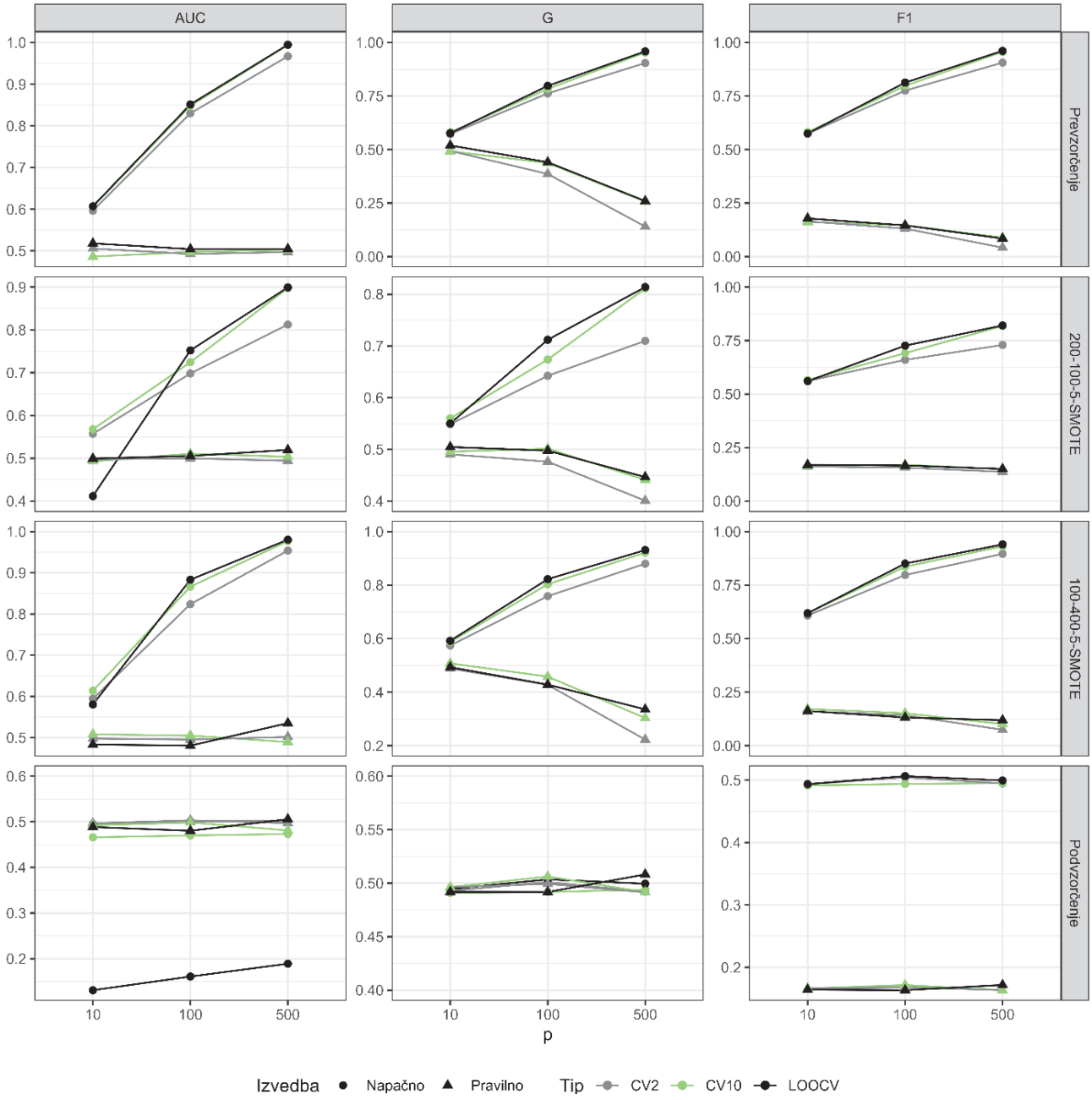


Slika 6 Navzkrižno preverjana točnost razvrščevalca v visokorazsežnem prostoru za različne velikosti manjšinskega razreda ob skupni uporabi različnih metod za uravnoteženje podatkov in različnih izvedb navzkrižnega preverjanja.

Spomnimo, s podvzorčenjem v testno množico ne uvajamo nobene informacije iz učne množice, zato je ta rezultat popolnoma pričakovan. Kljub temu, pa je F_1 mera precenjena. Pri vseh ostalih popravkih za uravnoteženje podatkov so v primeru napačne uporabe navzkrižnega preverjanja (izrazito) precenjene, še posebej, ko je neravnotežje v podatkih večje.

Odvisnost preoptimistične ocene od števila spremenljivk

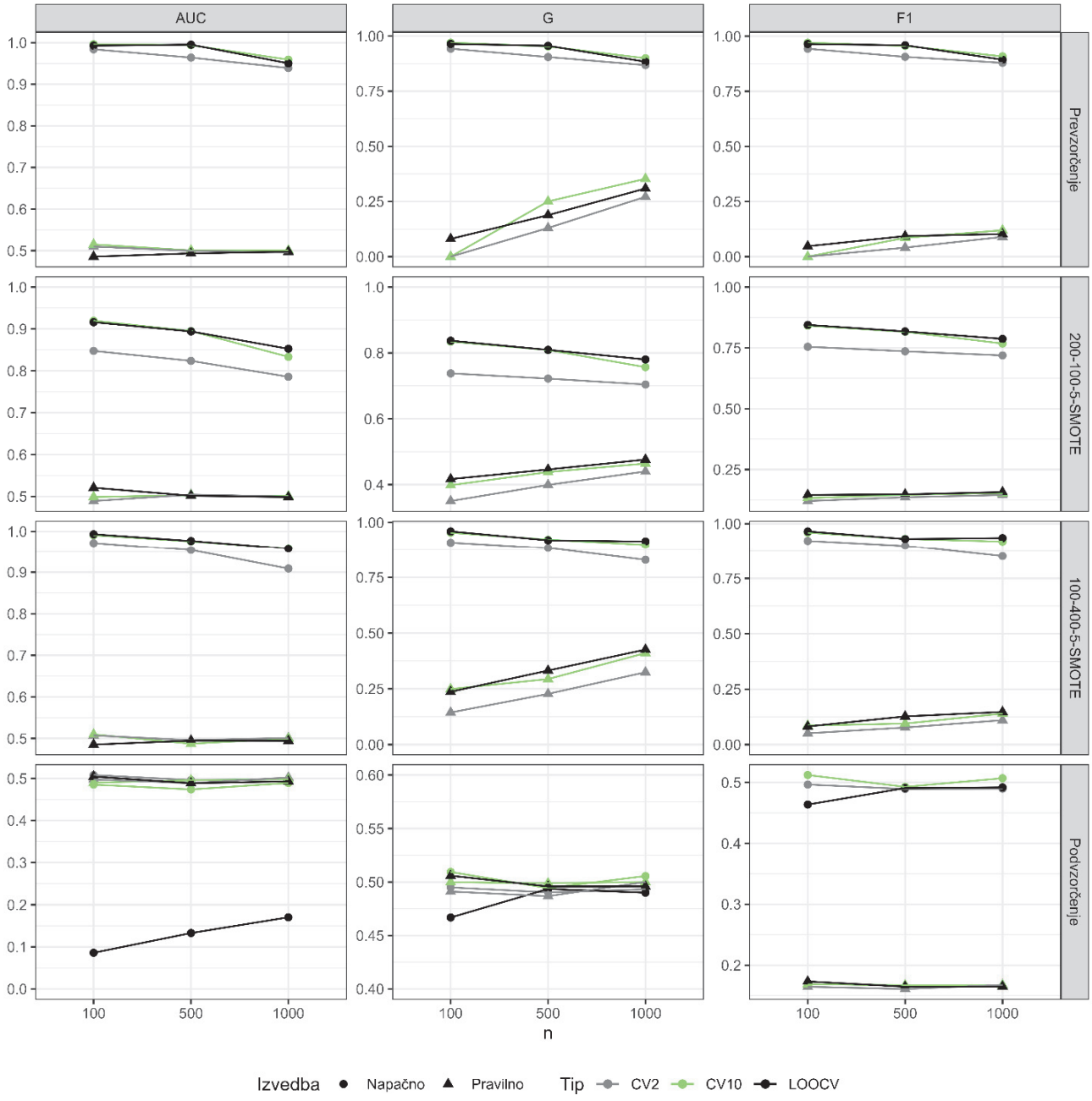
V tem delu spreminjamo število spremenljivk $p = 10, 100, 500$, ostali parametri pa so $N = 500$ in $p_{min} = 0.1$, rezultati so prikazani na sliki 7. Rezultati so zelo podobni kot v prejšnjem primeru, opazimo pa, da z večanjem števila spremenljivk ocene postajajo vedno bolj precenjene. Ko se število spremenljivk povečuje, postaja problem preprileganja bolj izrazit, kar se v primeru napačne izvedbe navzkrižnega preverjanja bolj pozna na preoptimističnih oceni točnosti delovanja razvrščevalca.



Slika 7 Navzkrižno preverjana točnost razvrščevalca za različne velikosti manjšinskega razreda ob skupni uporabi različnih metod za uravnoteženje podatkov in različnih izvedb navzkrižnega preverjanja, ko je število enot večje od števila spremenljivk.

Odvisnost preoptimistične ocene od števila enot

Tu spreminjamo število $N = 300, 500, 1000$, ostala parametra pa sta $p = 500$ in $p_{min} = 0{,}1$. Rezultati so prikazani na sliki 8. Podobno kot v prejšnjem primeru opazimo, da ob manjšanju števila enot v primeru napačne izvedbe navzkrižnega preverjanja ocene postajajo vedno bolj precenjene. Razlogi so enaki kot v prejšnjem primeru: ko se velikost učne množice zmanjšuje, se problem preprileganja povečuje, kar vodi do precenjenih ocen.



Slika 8 Navzkrižno preverjana točnost razvrščevalca ob skupni uporabi različnih metod za uravnoteženje podatkov in različnih izvedb navzkrižnega preverjanja pri različnem številu enot.

Zaključek

Ocenjevanje točnosti napovednih modelov je pomemben, če ne kar najpomembnejši korak pri razvoju napovednih modelov. Pokazali smo, da v primeru napačne uporabe navzkrižnega preverjanja v kombinaciji z uporabo metod za uravnoteženje podatkov precenimo točnost napovednega modela. Naše ocene tedaj nakazujejo, da gre za (zelo) dober napovedni model, dejansko pa je njegovo delovanje zelo slabo. Pojasnili smo razloge za to in predstavili dejavnike, ki vplivajo na preoptimizem: delež enot v manjšinski množici (preoptimizem se povečuje, ko se delež enot v manjšinski množici zmanjšuje), število spremenljivk (z večanjem števila spremenljivk se preoptimizem povečuje) in število enot (preoptimizem narašča z manjšanjem števila enot). Prvi dejavnik je neposredna posledica uvajanja informacije iz učne množico v testno: pri naključnem prevzorčenju v učni in testni množici nastopajo iste enote, ki jih je zaradi preprileganja precej lažje pravilno uvrstiti kot neke enote, ki jih med učenjem razvrščevalca nismo vključili v učno množico. Problem preprileganja je seveda bolj izrazit, ko je

število spremenljivk veliko in ko je število enot majhno, kar pojasnjuje druga dva dejavnika.

Pravilna izvedba navzkrižnega preverjanja je torej ključna, da se izognemo preoprimističnim ocenam in pravilno ovrednotimo moč napovednega modela. Zelo pomembno je, da so vse faze izgradnje modela (najsi gre za uravnoteženje podatkov, izbiro spremenljivk, izbiro najboljšega razvrščevalca ali nadomeščanje manjkajočih vrednosti) del navzkrižnega preverjanja. V nasprotnem primeru lahko v testni množici napačno upoštevamo informacijo iz učne množice in zato preoptimistično ocenimo točnost napovednega modela.

Reference

1. Bishop CM. Pattern recognition and machine learning (information science and statistics). New York 2007: Springer.
2. Ramaswamy S, Ross KN, Lander ES, Golub TR. A molecular signature of metastasis in primary solid tumors. *Nat Genet* 2003; 33(1): 49-54. https://doi.org/10.1038/ng1060
3. Shipp MA, Ross KN, Tamayo P, et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nat Med 2002; 8(1): 68-74. https://doi.org/10.1038/nm0102-68 (15. 10. 2022)
4. Iizuka N, Oka M, Yamada-Okabe H, et al. Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocellular carcinoma after curative resection. *Lancet* 2003; 361(9361): 923-929. https://doi.org/10.1016/S0140-6736(03)12775-4 (19. 11. 2022)
5. Sotiriou C, Neo SY, McShane LM, et al. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proc Natl Acad Sci USA* 2003; 100(18): 10393-10398. https://doi.org/10.1073/pnas.1732912100 (12. 10. 2022)
6. Wang Y, Klijn JG, Zhang Y, et al. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet* 2005; 365(9460): 671-679. https://doi.org/10.1016/S0140-6736(05)17947-1 (12. 10. 2022)
7. Shen R, Ghosh D, Chinnaiyan A, Meng Z. Eigengene-based linear discriminant model for tumor classification using gene expression microarray data. *Bioinformatics* 2006; 22(21): 2635-2642. https://doi.org/10.1093/bioinformatics/btl442 (10. 9. 2022)
8. Jimeno-Yepes AJ, Plaza L, Mork JG, Aronson AR, Díaz A. MeSH indexing based on automatically generated summaries. *BMC Bioinformatics* 2013; 14: 208. https://doi.org/10.1186/1471-2105-14-208 (10. 9. 2022)
9. Štotl I, Blagus R, Urbančič-Rovan V. Individualised screening of diabetic foot: creation of a prediction model based on penalised regression and assessment of theoretical efficacy. *Diabetologia* 2022; 65(2): 291-300. https://doi.org/10.1007/s00125-021-05604-2 (19. 11. 2022)
10. Tao D, Tang X, Li X, Wu X. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans Pattern Anal Mach Intell* 2006; 28(7): 1088-1099. https://doi.org/10.1109/TPAMI.2006.134 (19. 11. 2022)
11. He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 2009; 21(9):, 1263-1284. https://doi.org/10.1109/TKDE.2008.239 (5. 9. 2022)
12. Blagus R, Lusa L. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics*. 2010; 11: 523. https://doi.org/10.1186/1471-2105-11-523 (13. 8. 2022)
13. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 2002; 16: 341-378. https://doi.org/10.1613/jair.953 (5. 9. 2022)
14. Liu XY, Wu J, Zhou ZH. Exploratory undersampling for class-imbalance learning. *IEEE Trans Syst Man Cybern B Cybern* 2009; 39(2): 539-550. https://doi.org/10.1109/TSMCB.2008.2007853 (10. 9. 2022)
15. Lin WJ, Chen JJ. Class-imbalanced classifiers for high-dimensional data. *Brief Bioinform* 2013; 14(1): 13-26. https://doi.org/10.1093/bib/bbs006 (14. 10. 2022)
16. Galar M, Fernandez A, Barrenechea E, Bustince, H, Herrera F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern, Part C Appl Rev* 2012 42(4): 463-484. https://doi.org/10.1109/TSMCC.2011.2161285 (20. 10. 2022)
17. Blagus R, Lusa L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics* 2013; 14: 106. https://doi.org/10.1186/1471-2105-14-106 (17. 11. 2022)
18. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. New York 2003: Springer.
19. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 1970; 12(1): 55-67. https://doi.org/10.1080/00401706.1970.10488634 (20. 10. 2022)
20. Schaefer RL, Roi LD, Wolfe RA. A ridge logistic estimator. *Commun Stat Theory Methods* 1984; 13(1): 99-113. https://doi.org/10.1080/03610928408828664 (3. 11. 2022)
21. Zhu J, Hastie T. Classification of gene microarrays by penalized logistic regression. *Biostatistics* 2004; 5(3): 427-443. https://doi.org/10.1093/biostatistics/5.3.427 (18. 11. 2022)
22. Goeman J, Meijer R, Chaturvedi N, Lueder M. *L1 (lasso and fused lasso) and L2 (ridge) penalized estimation in GLMs and in the Cox model.* 2014. http://CRAN.R-project.org/package=penalized (20. 10. 2022)
23. Blagus R, Goeman JJ. Mean squared error of ridge estimators in logistic regression. *Stat Neerl* 2020; 74(2):

159-191. https://doi.org/10.1111/stan.12201 (10. 9. 2022)
24. Pepe MS. The statistical evaluation of medical tests for classification and prediction. New York 2003: Oxford University Press.
25. Blagus R, Goeman JJ. What (not) to expect when classifying rare events. *Brief Bioinform* 2018; 19(2): 341-349. https://doi.org/10.1093/bib/bbw107 (20. 10. 2022)
26. Fawcett T. An introduction to ROC analysis. *Pattern Recognit Lett* 2006; 27(8): 861-874. https://doi.org/10.1016/j.patrec.2005.10.010 (10. 9. 2022)
27. Perme MP, Manevski D. Confidence intervals for the Mann-Whitney test. *Stat Methods Med Res* 2019; 28(12): 3755-3768. https://doi.org/10.1177/0962280218814556 (18. 11. 2022)
28. Blagus R, Lusa L. Improved shrunken centroid classifiers for high-dimensional class-imbalanced data. *BMC Bioinformatics* 2013; 14: 64. https://doi.org/10.1186/1471-2105-14-64 (3. 11. 2022)
29. Simon R, Radmacher MD, Dobbin K, McShane LM. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *J Natl Cancer Inst* 2003; 95(1): 14-18. https://doi.org/10.1093/jnci/95.1.14 (20. 10. 2022)
30. Ambroise C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci U S A* 2002; 99(10) :6562-6566. https://doi.org/10.1073/pnas.102102699 (6. 10. 2022)
31. Taft LM, Evans RS, Shyu CR, et al. Countering imbalanced datasets to improve adverse drug event predictive models in labor and delivery. *J Biomed Inform* 2009; 42(2): 356-364. https://doi.org/10.1016/j.jbi.2008.09.001 (8. 10. 2022)
32. López-de-Uralde J, Ruiz I, Santos I, et al. Automatic morphological categorisation of carbon black nano-aggregates. In: Bringas PG, Hameurlain A, Quirchmayr G (eds). *Database and Expert Systems Applications*. DEXA 2010. Lecture Notes in Computer Science, vol 6262. Berlin, Heidelberg 2010: Springer: 185-193. https://doi.org/10.1007/978-3-642-15251-1_15 (3. 11. 2022)
33. Naseriparsa M, Kashani MM. Combination of PCA with SMOTE resampling to boost the prediction rate in lung cancer dataset. *Int J Comput Appl* 2013; 77(3): 33-38. https://doi.org/10.5120/13376-0987 (16. 11. 2022)
34. Blagus R, Lusa L. Joint use of over- and under-sampling techniques and cross-validation for the development and assessment of prediction models. *BMC Bioinformatics* 2015; 16: 363. https://doi.org/10.1186/s12859-015-0784-9 (3. 11. 2022)
35. Japkowicz N. The Class Imbalance Problem: Significance and Strategies. In: *Proceedings of the 2000 International Conference on Artificial Intelligence ICAI*, 2000.
36. Rahman MM, Davis D. Cluster based under-sampling for unbalanced cardiovascular data. In: *Proceedings of the World Congress on Engineering*, vol. 3, London 2013: 3-5. https://www.iaeng.org/publication/WCE2013/WCE2013_pp1480-1485.pdf (3. 11. 2022)
37. Zhang JP, Mani I. KNN Approach to unbalanced data distributions: a case study involving information extraction. In: *Proceeding of International Conference on Machine Learning (ICML 2003), Workshop on Learning from Imbalanced Data Sets*. Washington 2003: 1-7.
38. Cover TM, Hart PE. Nearest neighbor pattern classification. *IEEE Trans Inf Theory* 1967 13(1): 21-27. 10.1109/TIT.1967.1053964 (20. 10. 2022)
39. Forman G, Scholz M. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *SIGKDD Explor Newsl* 2010; 12(1): 49-57. https://doi.org/10.1145/1882471.1882479 (10. 9. 2022)
40. Cox DR, Hinkley DV. *Theoretical statistics*. New York 1979: CRC Press.
41. Le Cessie S, Van Houwelingen JC. Ridge estimators in logistic regression. *J R Stat Soc Ser C Appl Stat* 1992; 41(1): 191-201. https://doi.org/10.2307/2347628 (3. 11. 2022)