SVEUČILIŠTE U RIJECI

ODJEL ZA INFORMATIKU

Nola Čumlievski

Ivan Šimičić

PROJEKT IZ KOLEGIJA OTKRIVANJE ZNANJA U PODATCIMA

Zadatak 9.

Rijeka, 2020.

Sadržaj

1. Uvod....1
2. Rješavanje problema nebalansiranosti podataka....2
3. Usporedba kvalitete modela dobivenih postupcima strojnog učenja....6
4. Objašnjenje modela....12
   4.1. Break Down plot....12
   4.2. Shapley metoda....14
   4.3. Mjere važnosti varijabli....16
5. Zaključak....17
Popis literature....18

1. Uvod

U ovom projektnom zadatku ćemo obraditi zadatke zadane temom pod brojem 9. nad skupom podataka Peaches_100_ALL_1 koristeći SMOTE metodu za nebalansiranu klasifikaciju te logističku regresiju i algoritam podizanja gradijenta za usporedbu kvalitete modela dobivenih postupcima strojnog učenja. Za postupak objašnjena modela uzeli smo Break Down, Shapley i metodu mjere važnosti varijabli budući da smo iste obradili ranije na seminarskim zadatcima.

U projektnom zadatku su korištene biblioteke dplyr, arules, ggplot2, plyr, DMWR, corrplot, lares, caret, xgboost, glmnet, pROC te DALEX.

2. Rješavanje problema nebalansiranosti podataka

Problem nebalansiranosti podataka je učestala pojava u strojnom učenju, a do njega dolazi kada u skupu podataka imamo neproporcionalan omjer zapažanja unutar svake klase. Nebalansiranost najčešće nalazimo kada radimo s podatcima vezanim uz zdravstvene dijagnoze, filtriranje spam poruka ili otkrivanje prevara kod bankarskog poslovanja.

U ovom projektnom zadatku koristit ćemo metodu SMOTE, a ideja je da moramo preuzorkovati vrijednosti unutar nebalansirane klase.

SMOTE radi na način da traži primjere koji su bliski u prostoru značajki nakon čega povlači liniju između tih primjera te zatim kreira nove uzorke na toj liniji.

Rješavanje problema započinje učitavanjem podataka te vezanjem istih za workspace. Nakon što smo učitali podatke, možemo primijetiti kako imamo redove s nepostojećim (NA) vrijednostima te iste moramo izbaciti. Također moramo izbaciti prvi stupac budući da nam redni broj opažanja nije važan za analizu. Kod možemo vidjeti u nastavku:

```
breskve = peaches_1
attach(breskve)

breskve[ breskve == "NULL"] <- NA
breskve = breskve[ complete.cases(breskve),]

breskve = subset(breskve, select=-num)
```

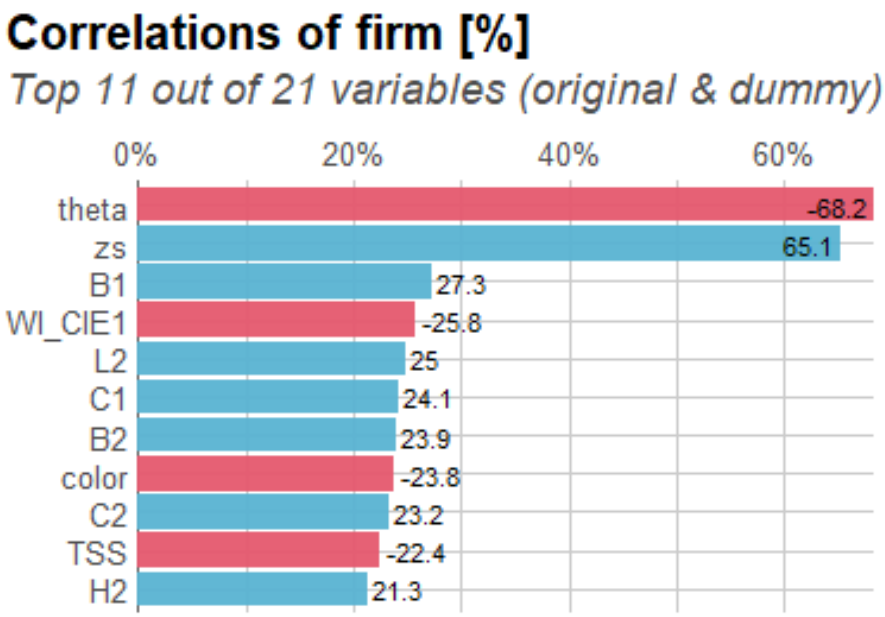
Kako bi uspješno „očistili“ skup podataka, potrebno je pronaći korelaciju između varijabli. Varijabla *AE* (subjektivna procjena postotka prekrivenosti ploda bojom) je kategorička te ju je potrebno faktorirati (pretvoriti klase u brojčane klase) prije nego što koristimo funkciju *cor* uz pomoć koje dobijemo izračun korelacija između varijabli (kako bi izračunali korelaciju između vrijednosti, sve varijable trebaju biti numeričke).

```
korelacije = cor(breskve)
korelacije = korelacije[, "firm"]
korelacije
```

Pokretanjem prethodnog koda dobijemo pregled korelacija nezavisnih varijabli s obzirom na ciljnu varijablu *firm* koja predstavlja tvrdoću breskve:

```
     volume        mass        firm         TSS          TA          zs
-0.04330500  0.07237648  1.00000000 -0.22449360 -0.10878050  0.65133929
      theta          AE       color      dE2000          L1          A1
-0.68229694  0.11678034 -0.23824811  0.15820424  0.19072398  0.17776399
         B1          C1          H1     WI_CIE1          L2          A2
 0.27311508  0.24153209  0.07967059 -0.25786020  0.24938657 -0.16466426
         B2          C2          H2     WI_CIE2
 0.23897835  0.23187327  0.21284008 -0.17158786
```

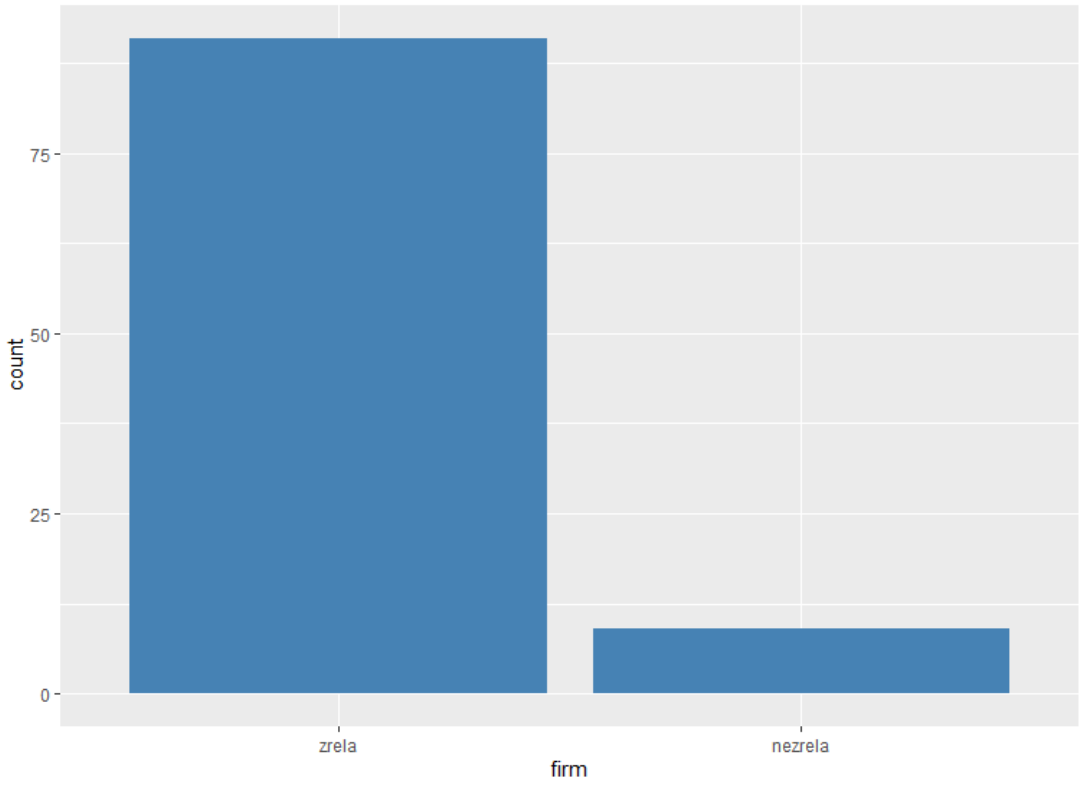
Za potrebe kreiranja modela koristit ćemo varijable koje imaju korelaciju veću od 0.2, a to su varijable *TSS, zs, theta, color, B1, C1, WI_CIE1, L2, B2, C2* te *H2*.



Slika 1. Vizualizacija korelacije

Prije nego što krenemo sa samom SMOTE metodom, potrebno je diskretizirati ciljnu varijablu, a za to ćemo koristiti kategorije „zrela“ i „nezrela“. Važno je za napomenuti da je važan redoslijed kojim navodimo nazive klasa, s obzirom da vrijednosti *firm* varijable s manjom vrijednošću označavaju zrelije breskve, potrebno je navedeni naziv navesti prvi.

```
breskve$firm=discretize(breskve$firm, method = "interval", breaks = 2,labels =
c("zrela", "nezrela"))
```



Slika 2. Odnos između kategorija varijable firm

Uz pomoć funkcije *prop.table* možemo vidjeti točne proporcije varijable *firm*, nakon čije primjene možemo uočiti jasnu razliku između broja opservacija unutar navedenih klasa:

```
  zrela nezrela
   0.91    0.09
```

Za uporabu SMOTE metode možemo koristiti SMOTE funkciju unutar koje unosimo varijablu za koju moramo riješiti problem nebalansiranosti te zadajemo željene postotke:

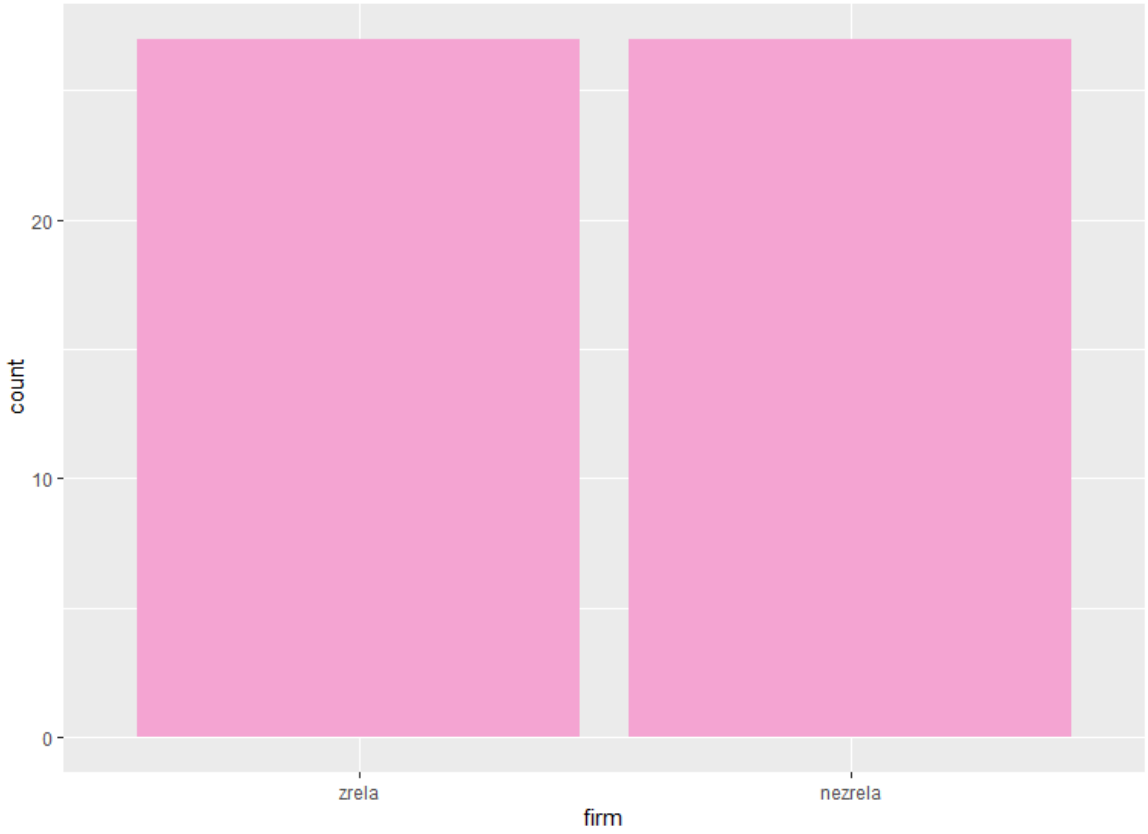
```
breskve_df <- as.data.frame(breskve)
breskve_novi <- SMOTE(firm ~ ., breskve_df, perc.over = 200, perc.under=150)
```

Proučavanjem dane literature, metodu SMOTE moguće je primijeniti i unutar funkcije *trainControl* koja služi za postavljanje parametara koji će se koristiti prilikom treniranja modela. No, pokušajem izvođenja oba načina primjene, primjena SMOTE funkcije se u ovom slučaju pokazala učinkovitijom. Postavljamo parametar *perc.over* na 200 kako bi smanjili broj

opservacija klase „zrela“, te parametar *perc.under* na 150 kako bi povećali broj opservacija s klasom „nezrela“.

Slično kao i prije uporabe SMOTE metode, uz pomoć funkcije *prop.table* ćemo saznati nove proporcije kategorija varijable firm:

```
zrela nezrela
  0.5     0.5
```



Slika 3. Odnos kategorija varijable firm nakon SMOTE

Možemo vidjeti kako smo nakon SMOTE postigli uravnoteženu raspodjelu kategorija varijable *firm*, a to će uvelike poboljšati kvalitetu modela na način da, iako smo smanjili broj opservacija, skup podataka je kvalitetniji te balansiranjem klasa sprečavamo „overfitting“ modela većinskoj klasi.

3. Usporedba kvalitete modela dobivenih postupcima strojnog učenja

Nakon što smo riješili problem nebalansiranosti podataka, možemo kreirati modele uz pomoć strojnog učenja te iste usporediti kako bi vidjeli koji od njih je kvalitetniji i bolje opisuje dobiveni i procesirani skup podataka.

```
set.seed(42)
index = createDataPartition(breskve_novi$firm, p = 0.7, list = FALSE)
treniranje = breskve_novi[index, ]
testiranje = breskve_novi[-index, ]

gbmGrid =  expand.grid(interaction.depth = c(1, 5, 9), n.trees = 150,
                       shrinkage = 0.1, n.minobsinnode = 5)
```

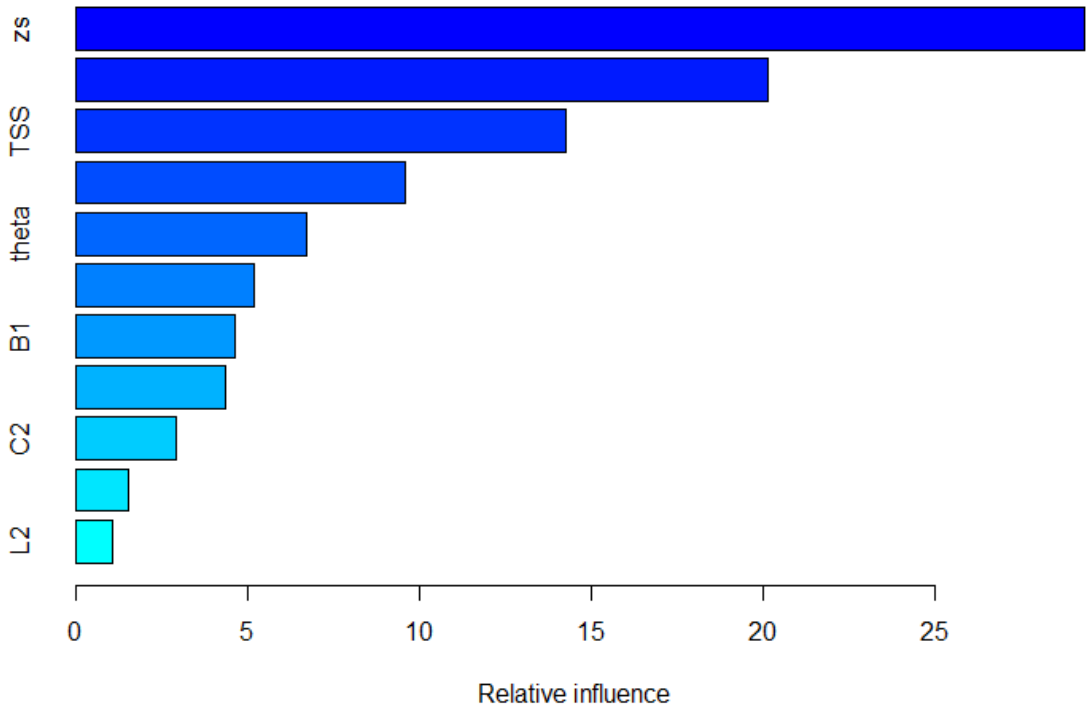
Potrebno je odrediti „seed“ kako bi iste rezultate mogli reproducirati, a zatim možemo kreirati skupove za treniranje i testiranje. Kod ovog modela postojao je problem da, parametar *n.minobsinnode* koji je po defaultu 10, sprječava izvršavanje koda te je bilo potrebno izraditi custom grid gdje navodimo manji broj opservacija po čvoru (ovaj problem se javlja jer je skup podataka malen). Kod većih skupova podataka broj opservacija po čvoru može biti zadani - 10 ili više.

Uz pomoć funkcije *train* i metode „gbm“ kreiramo prvi model:

```
gradient_boosting = train(firm ~ theta + zs + B1 + WI_CIE1+ L2 + C1 + B2 + col
or + C2 + TSS + H2, data = treniranje,
                          method = "gbm", tuneGrid = gbmGrid)

summary(gradient_boosting)
```

```
            var   rel.inf
zs           zs 29.378620
color     color 20.163795
TSS         TSS 14.281152
C1           C1  9.607290
theta     theta  6.732960
H2           H2  5.207636
B1           B1  4.669315
B2           B2  4.387800
C2           C2  2.921351
WI_CIE1 WI_CIE1  1.547064
L2           L2  1.103016
```

Funkcija *summary* nam omogućava uvid u sažetak informacija o kreiranom modelu, a u ovom slučaju možemo vidjeti relativan utjecaj nezavisnih varijabli na ciljnu varijablu.



Slika 4. prikaz utjecaja varijabli na ciljanu varijablu

Nakon što smo dobili uvid u utjecaj varijabli s obzirom na ciljnu varijablu, možemo kreirati predikciju uz pomoć funkcije *predict*:

```
predikcija = predict(gradient_boosting, newdata = testiranje)
```

Na temelju kreirane predikcije možemo napraviti konfuzijsku matricu i ispisati statistike:

```
          Reference
Prediction zrela nezrela
   zrela       7       1
   nezrela     1       7

               Accuracy : 0.875
                 95% CI : (0.6165, 0.9845)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.00209
```

```
                Kappa : 0.75

 'Positive' Class : zrela
```

Možemo vidjeti kako je model ostvario točnost od 87.5% uz Kappa koeficijent od 0.75 u slučaju kada je pozitivna vrijednost klase „zrela“. Na slici 5 to možemo vidjeti u grafičkom prikazu.

Slika 5. konfuzijska matrica za GBM model

Nakon što smo kreirali prvi model i napravili predikciju, možemo krenuti s izradom drugog modela, a pri tome ćemo također koristiti funkciju *train* uz parametar „glm“ budući da se radi o logističkom modelu:

```
logisticki_model = train(firm ~ theta + zs + B1 , data = treniranje,
                         method = "glm", verbose = FALSE, metric = "ROC", fa
mily = "binomial", trControl = kontrola, control = list(maxit = 150))
```

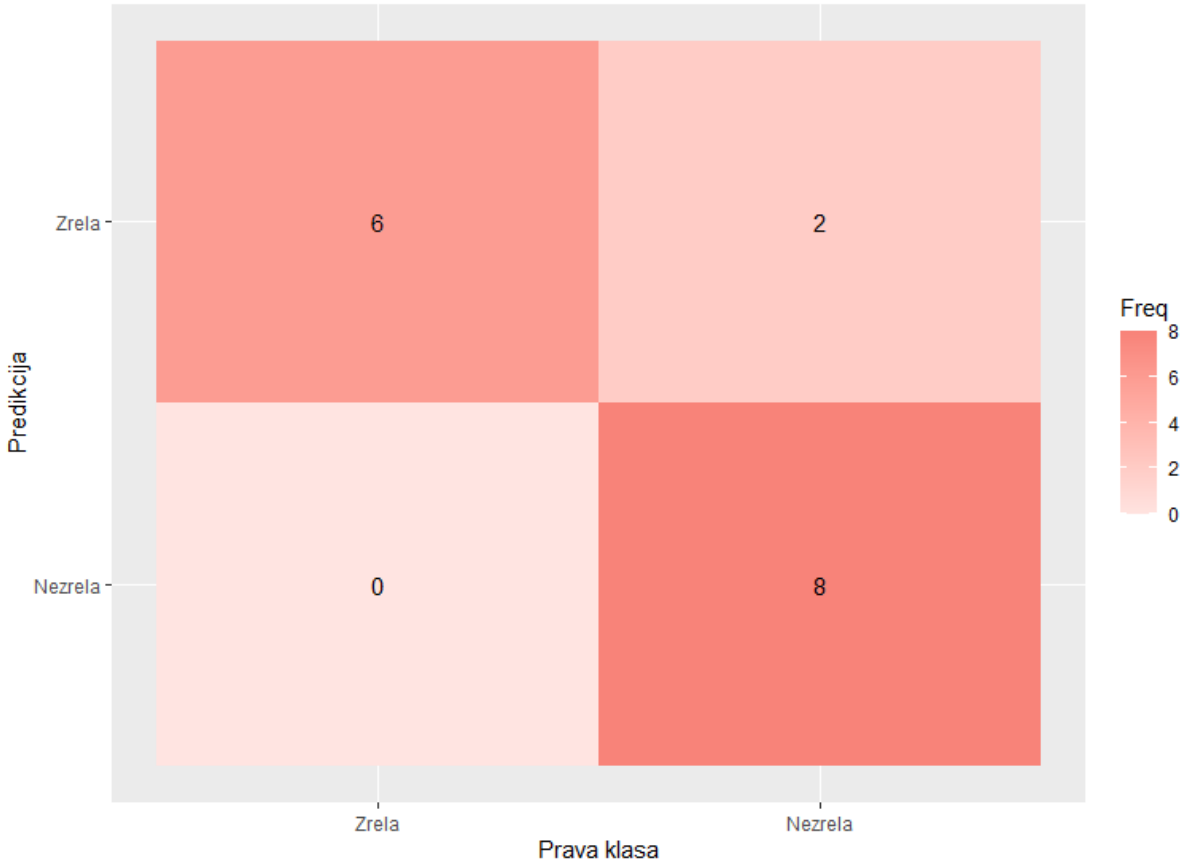
```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3435  -0.5699   0.2575   0.7475   1.3517

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.47043    4.82856  -0.926  0.35453
theta        0.24498    0.20306   1.206  0.22765
zs          11.61212    3.58294   3.241  0.00119 **
B1           0.09941    0.05242   1.897  0.05789 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 181.65  on 132  degrees of freedom
Residual deviance: 121.35  on 129  degrees of freedom
AIC: 129.35

Number of Fisher Scoring iterations: 5
```

Kod logističkog modela, javljao se problem kod navođenja svih 11 „relevantnih“ nezavisnih varijabli unutar formule modela. Prilikom navođenja svih navedenih varijabli, dobili bi model gdje bi *p* vrijednost svih varijabli iznosila 1 i *z* vrijednosti 0. Ove vrijednosti u sažetku modela označavaju overfit modela gdje model ne predviđa vrijednosti u smislu vjerojatnosti te je bilo potrebno ukloniti određen broj varijabli kako bi dobili validan model.

Uz pomoć funkcije predict ćemo kreirati predikciju, a zatim ćemo ispisati informacije i statistiku uz pomoć konfuzijskih matrica:

```
Confusion Matrix and Statistics

          Reference
Prediction zrela nezrela
   zrela       6       0
   nezrela     2       8

               Accuracy : 0.875
                 95% CI : (0.6165, 0.9845)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.00209

                  Kappa : 0.75

       'Positive' Class : zrela
```

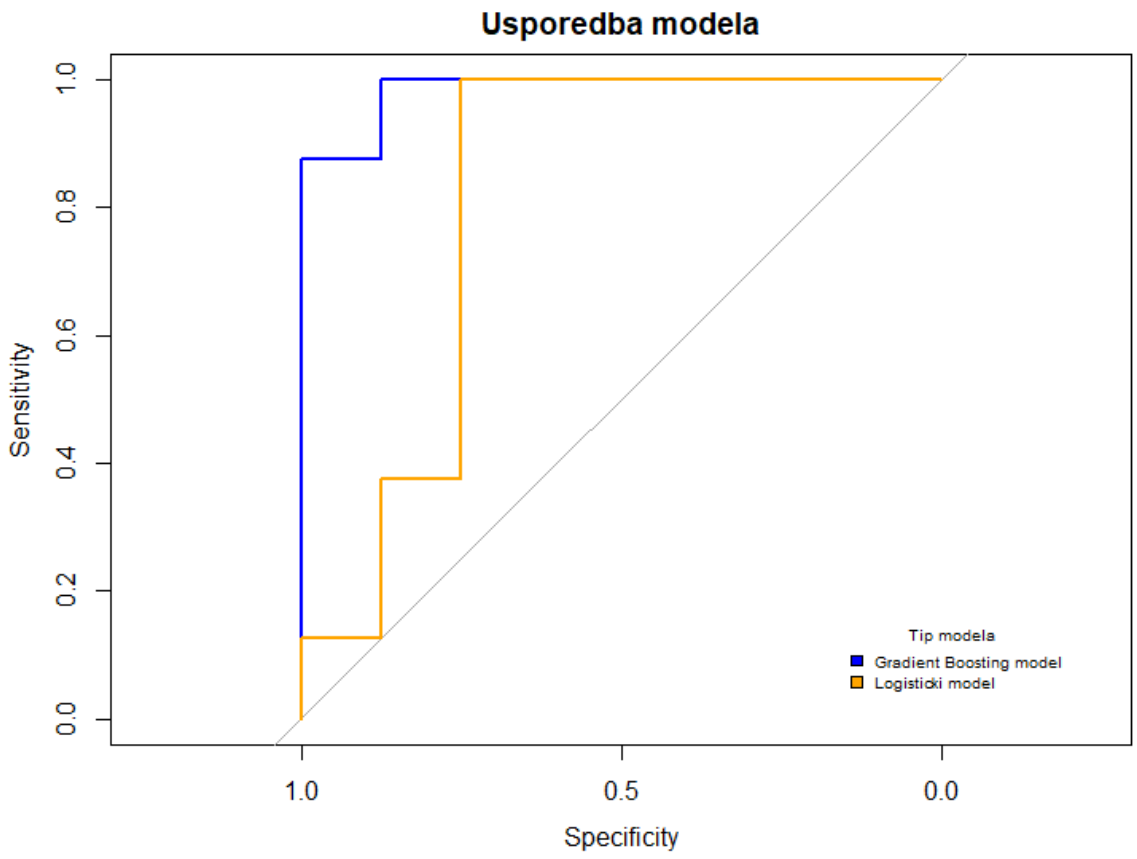
Možemo vidjeti kako logistički model ima jednaku točnost kao i prethodni, a ona iznosi 87.5%, ali također ima i jednak Kappa koeficijent koji iznosi 0.75. Važno je uočiti da modeli imaju različitu distribuciju klasa unutar rezultata predikcije.



Slika 6. konfuzijska matrica za logistički model

Kako bi usporedili modele, moramo izračunati prostor ispod krivulje, a to ćemo napraviti uz pomoć funkcija *roc* i *auc*. Usporedbu ćemo također prikazati grafički.

Prostor ispod krivulje za Gradient Boosting model iznosi 0.9844, a za logistički model 0.8125, a grafički prikaz je vidljiv na slici 7.



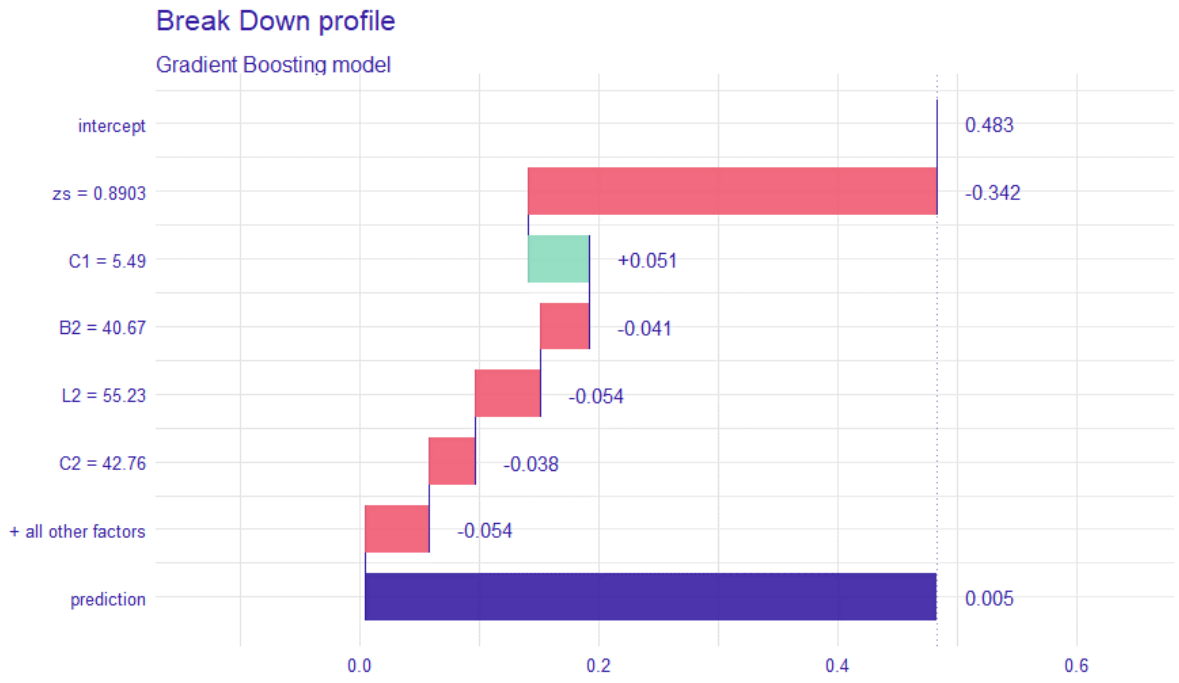
Slika 7. Usporedba prostora ispod krivulje

4. Objašnjenje modela

Unutar ovog poglavlja ukratko su opisana i primjenjena 3 postupka za objašnjenje modela, a to su Break Down, Shapley i mjere važnosti varijabli (engl. *Variable-importance measures*).

4.1. Break Down plot

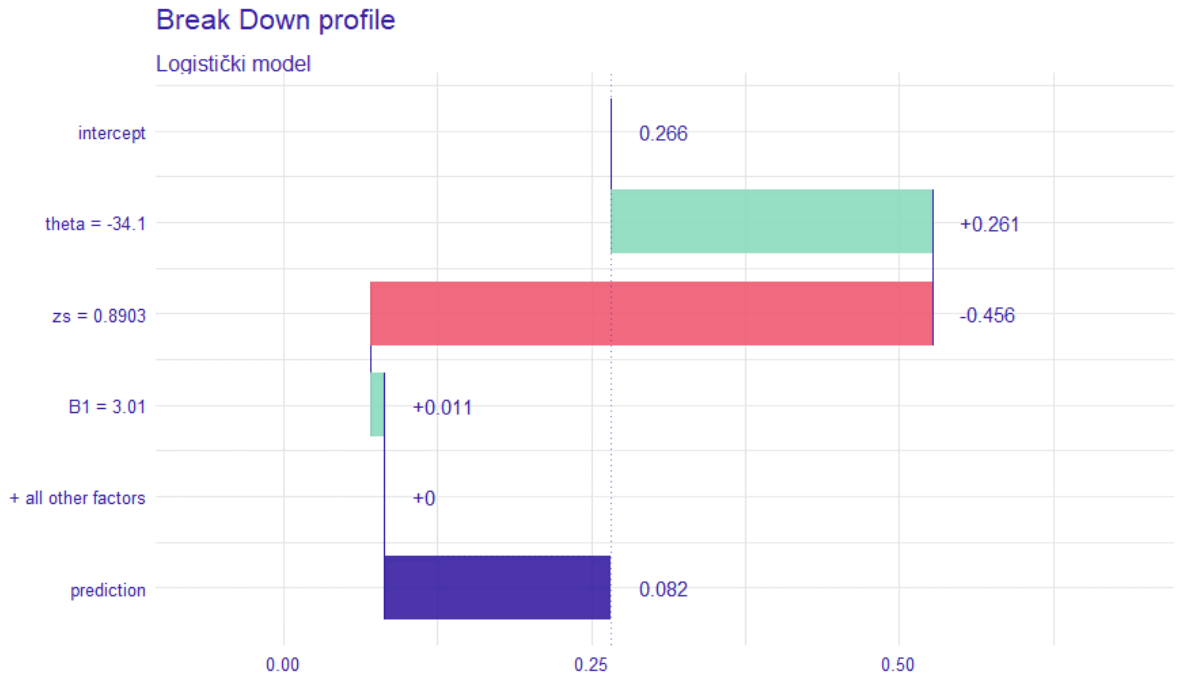
Break Down plot pokazuje nam kako doprinosi vezani uz pojedinu nezavisnu varijablu utječu na predikciju modela, formirajući na kraju stvarnu predikciju određene instance (opservacije). Za objašnjenje modela koristit ćemo funkciju *explain* uz parametre „break_down“ i „shap“ koji predstavljaju različite metode objašnjavanja modela. Parametar „break_down“ nam služi prilikom izrade Break Down grafova na kojima su vidljive interakcije između varijabli, a parametar shap nam služi za Shapley metodu.



Slika 8. Break down plot za GBM model

Na slici 8 možemo vidjeti break down plot za Gradient Boosting model. Za testiranje je korišten prvi red skupa podataka za testiranje. Zeleni i crveni stupci nam ukazuju na negativan i pozitivan utjecaj varijabli na vrijednosti predikcije modela. Red po nazivom *„intercept“* (0.483) prikazuje srednju vrijednosti predikcije za cijeli skup podataka. Svi naredni redovi prikazuju nam promjene u srednjoj vrijednosti predikcije (promjene u predikciji su navedene desno od reda) prouzročene vrijednostima pridruženim varijablama (vrijednosti lijevo, uz imena varijabli),

npr. početna predikcija od 0.483, smanjena je za 0.342 pod utjecajem *zs* varijable i tako do kraja redova. Zadnji red, *„prediction“*, označava konačnu predviđenu vrijednost zrelosti instance (~0.5), što znači da model sa vjerojatnosti od ~0.5 zaključuje da je prva breskva (instanca) skupa zrela. Broj desno od reda predikcije označava koliko se predikcija povećala ili smanjila u odnosu na početnu predikciju. Gledajući skup podataka, model je točan.

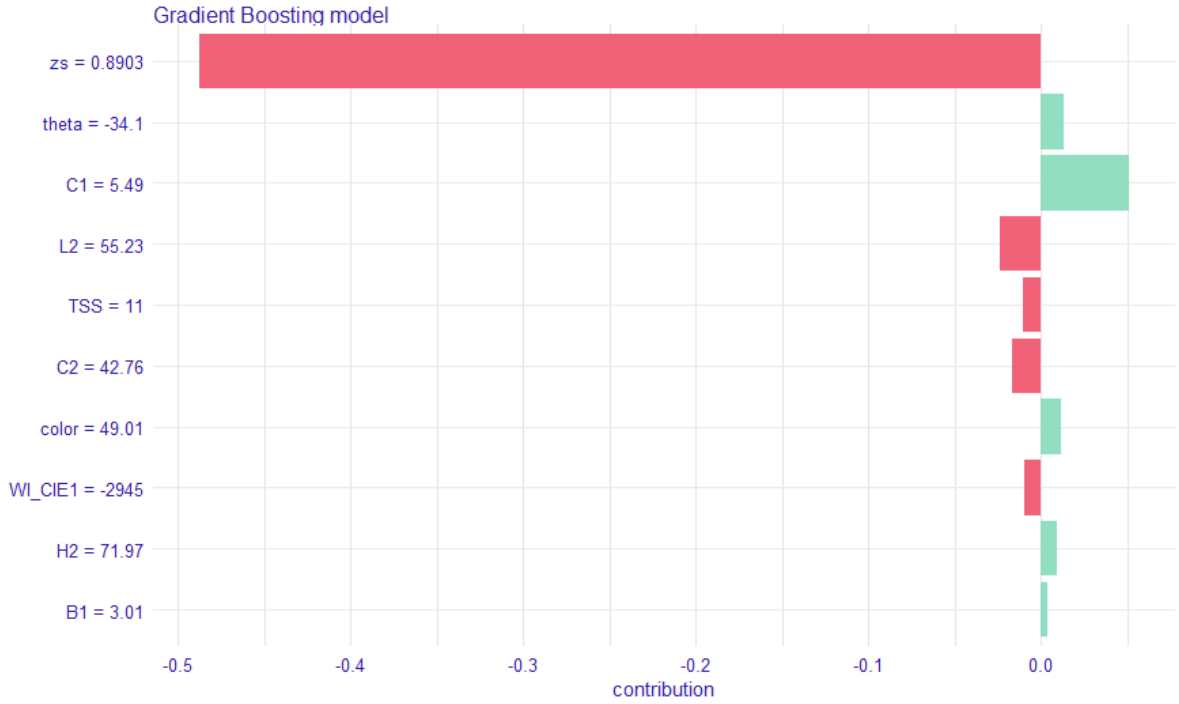


Slika 9. Break down plot za logistički model

Na slici 9 možemo vidjeti Break Down plot za logistički model. Za testiranje je uzeta treća instanca skupa za testiranje, također označena pod „zrela“. Vidimo da početna predikcija modela iznosi 0.266 – vjerojatnost da je breskva zrela. Zatim *theta* vrijednost pridodaje vrijednost od 0.261 navedenoj vjerojatnosti no *zs* vrijednost znatno smanji vjerojatnost instance. Konačna predikcija modela iznosi ~0.3 vjerojatnost da je breskva zrela, što predstavlja netočnu predikciju.

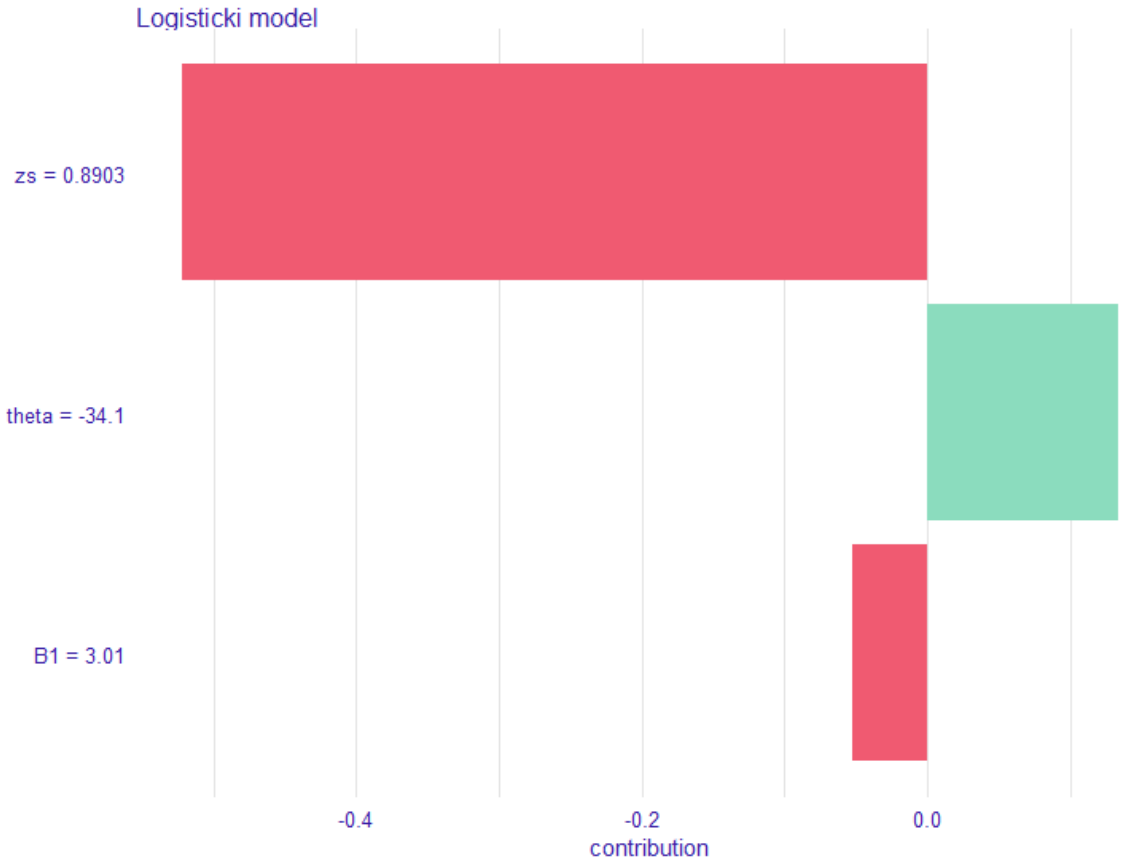
4.2. Shapley metoda

Shapley metoda koristi se za objašnjavanje mjere koliko svaka varijabla unutar modela, pozitivno ili negativno, doprinosi rezultatu modela. Ova metoda, kao i metoda „Break Down“, bavi se izračunom doprinosa nezavisnih varijabli prilikom predikcije modela. No, za razliku od metode „Break Down“, pomoću Shapley metode obavlja se izračun srednje vrijednosti doprinosa varijabli za sve nasumične poretke, što znači da poredak varijabli prilikom primjene Shapley metode nema utjecaj na konačni rezultat, s obzirom da se kao konačan rezultat uzima prosjek svih prethodnih rezultata.



Slika 10. Plot Shapley metode za Gradient Boosting Model

Iz priložene slike možemo primijetiti Shapley vrijednosti pojedine varijable unutar Gradient Boosting modela. Vidimo da daleko najveći utjecaj na predikciju Gradient Boosting modela ima varijabla *zs* (realni dio impendancije breskve) sa negativnim utjecajem od ~ -0.5 Shapley vrijednosti, te nakon navedene varijable redom varijable *C1* (CIELCh komponenta najtamnijeg djela breskve) i *L2* (CIELAB komponenta najsvjetlijeg djela breskve) sa Shapley vrijednostima redom ~0.05 i -0.025. Vrijednosti navedene uz imena varijabli predstavljaju numeričke vrijednosti navedenih varijabli koje označavaju karakteristike zrele breskve, npr. možemo reći, gledamo li varijable s najvećim utjecajem, da ukoliko breskva ima realni dio impendancije veličine 0.8903, C1 komponentu najtamnijeg djela breskve veličine 5.49 te L2 komponentu najsvjetlijeg djela breskve veličine 55.23 može se smatrati zrelom breskvom – prema Gradient Boosting modelu.

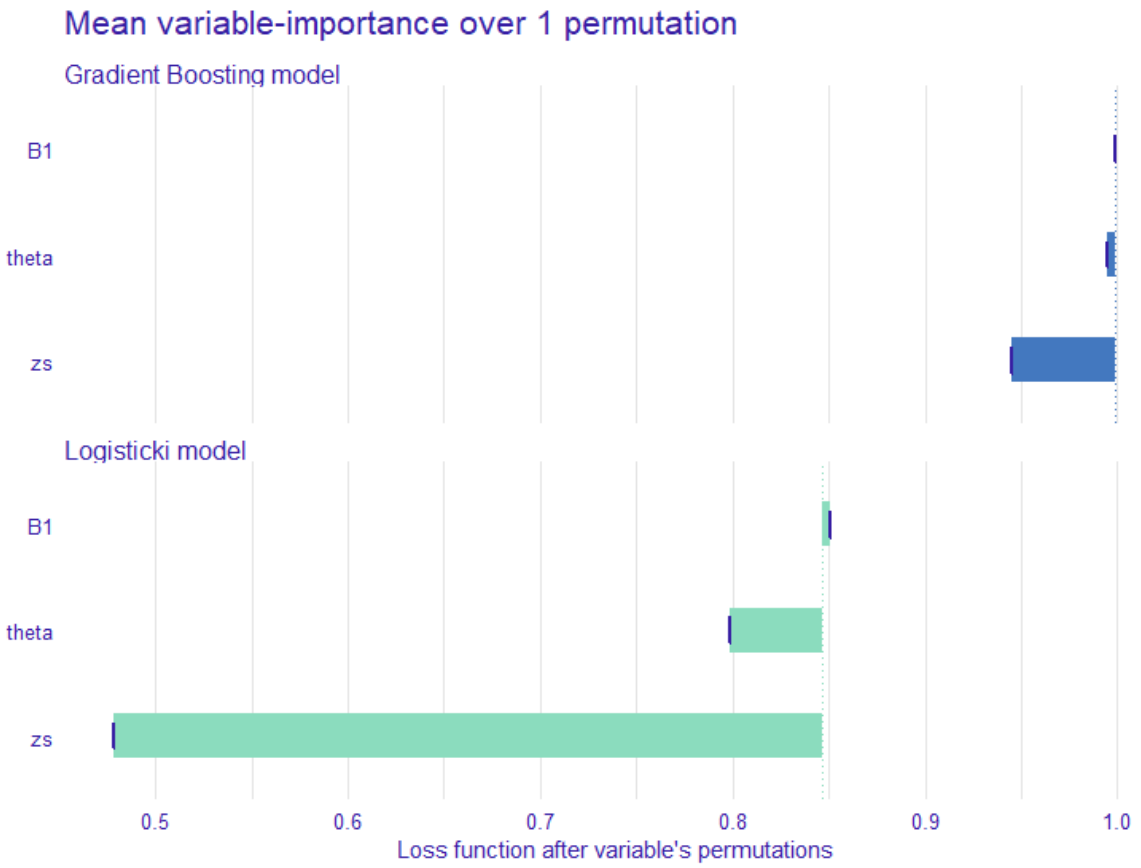


Slika 11. Graf za shapley metodu

Na slici 11 možemo vidjeti plot Shapley vrijednosti za logistički model. Vidimo da, od korištene 3 varijable, najveći utjecaj ima realni dio impedancije na zrelost breskve sa Shapley vrijednosti od ~-0.5, te zatim varijabla *theta* sa Shapley vrijednosti od ~ 0.13 te zadnja varijabla *B1* (CIELAB komponenta najtamnijeg dijela breskve) sa Shapley vrijednosti od ~-0.05.

4.3. Mjere važnosti varijabli

Glavna ideja ove metode jest izmjeriti koliko se mijenja performansa modela ukoliko se određena nezavisna varijabla ukloni. Ako je varijabla važna za predikciju modela, očekuje se da će se performanse modela, nakon permutacije vrijednosti varijable, značajno pogoršati. Podrazumijeva se da, što je veća razlika u performansi modela, to je nezavisna varijabla značajnija za predikciju.



Slika 12. Graf središnje varijabilne važnosti

Na slici 12 možemo vidjeti graf mjere važnosti varijabli za oba modela (iz modela gradient boosting uzete su iste varijable koje su korištene za izradu logističkog modela kako bi se mogle usporediti performanse). Iz priloženog vidimo, da se performanse modela najviše pogoršavaju uklanjanjem *zs* varijable, što bi značilo da navedena varijabla ima najveći utjecaj na konačnu predikciju modela. Zatim, druga najviše značajna varijabla je *theta* a zatim *B1.* Možemo primijetiti i da varijabla *zs* dosta više utječe na performanse logističkog modela s manje varijabli te da, iako je također najznačajnija varijabla modela Gradient Boosting, isključivanje varijable iz Gradient Boosting modela imalo bi dosta manji značaj nego uklanjanje iste iz logističkog modela (gdje bi performanse modela značajno pale, ali nije ni čudno s obzirom da se u modelu koriste 3 varijable).

5. Zaključak

U ovom projektnom zadatku smo primijenili ranije naučene metode poput gradient boosting i logističkih modela u kombinaciji s metodama koje smo naučili ove godine, kao što metode break down, shapley metoda i mjere važnosti varijabli. Projekt upotpunjuje i utvrđuje sva stečena znanja na kolegijima vezanim uz strojno učenje na diplomskom studiju te je bio vrlo zanimljiv i koristan za buduća istraživanja i daljnje učenje i napredovanje. Kao daljne istraživanje, moguće je ukloniti više značajne varijable iz modela te vidjeti kako uklananje iste utječe na točnost modela.

Popis literature

1. Explanatory Model Analysis, dostupno na: http://ema.drwhy.ai/
2. SMOTE for Imbalanced Classification with Python, dostupno na: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
3. SMOTE - Supersampling Rare Events in R, dostupno na: http://amunategui.github.io/smote/