Predviđanje ulaska poduzeća u Hrvatskoj u postupak predstečajne nagodbe

- Projektni prijedlog -

David Bojanić, Domagoj Demeterfi

04/2018

Sadržaj

Predviđanje ulaska poduzeća u Hrvatskoj u postupak predstečajne nagodbe

1 Sažetak

Predviđanje stečaja je područje od velikog interesa u ekonomiji. Cilj projekta je postaviti prediktivni model koji koristeći ekonometrijske mjere poduzeća u Hrvatskoj predviđa ulazak tog poduzeća u postupak predstečajne nagodbe. Uz to cilj je dobiveni model interpretirati s ekonomskog stajališta. Ovaj projektni prijedlog sadrži opis problema, motivaciju njegovog rješavanja, konkretne ciljeve projekta i detaljan prijedlog rješenja.

2 Uvod

Predviđanje stečaja poduzeća od iznimne je važnosti pri donošenju ekonomskih odluka. Stanje poduzeća, bilo malog ili velikog, od interesa je za lokalnu zajednicu, sudionike u industriji i investitore, ali i za zakonodavce te globalnu ekonomiju. Stoga ne čudi što taj problem već dugo privlači pozornost istraživača.

U ovom projektu razvijamo model za predviđanje ulaska poduzeća u postupak predstečajne nagodbe. Projekt je inspiriran radom [1] u kojem su autori razvili okvir za predviđanje stečaja poduzeća temeljen na modelu ansambla boosted stabala koji je treniran tehnikom Extreme Gradient Boosting. Osnova našeg rada je također spomenuti model (detaljnije o modelu u Metodologija) no umjesto predviđanja stečaja mi pokušavamo predvidjeti ulazak poduzeća u postupak predstečajne nagodbe. U [1] autori su svoj model testirali na podacima za poljska poduzeća u sektoru proizvodnje, a mi ćemo svoj testirati na podacima o hrvatskim poduzećima u građevinskom sektoru.

Osim samog razvoja modela predviđanja ulaska u postupak predstečajne nagodbe u projektu bismo veliki naglasak stavili i na interpretaciju dobivenog modela s ekonomskog stajališta.

2.1 Ciljevi projekta

Obično se u literaturi koja se bavi predviđanjem propadanja poduzeća predviđa ulazak poduzeća u stečaj. Mi smo kroz razgovor sa stručnjakinjom iz ovog područja i zbog dostupnih podataka odlučili da bi za hrvatski slučaj bilo zanimljivo predviđati ulazak u postupak predstečajne nagodbe. Naime, zbog zakonske regulative specifične za Hrvatsku poduzeća u Hrvatskoj su zapravo niz godina prije samog ulaska u stečaj u velikim problemima.

Osnovni cilj ovog projekta je dobiti model za predviđanje ulaska poduzeća u postupak predstečajne nagodbe. Unutar modela poduzeća su opisana značajkama koje su dobivene iz njihovih bilanci iz određene godine. Više o značajkama nalazi se u dijelu Podatkovni skup. Želimo dobiti model koji će na temelju tih značajki predviđati hoće li poduzeće u nekom budućem trenutku ući u postupak predstečajne nagodbe. Potrebno je specificirati u kojem periodu predviđamo. Po uzoru na [1] zapravo ćemo rješavati pet zadataka binarne klasifikacije u ovisnosti o periodu predviđanja: poduzeće je ušlo u postupak predstečajne nagodbe nakon jedne, dvije, tri, četiri ili pet godina. Potencijalne mjere uspješnosti ovog dijela projekta objašnjene su u poglavlju Metode evaluacije dok je konkretan način njihove

primjene opisan u poglavlju Evaluacija modela.

Kao što smo već spomenuli osim same prilagodbe modela u ovom projektu bavit ćemo se i njegovom interpretacijom. Detalji o načinu interpretacije nalaze se u poglavlju Interpretacija. Također se nadamo da ćemo imati priliku o rezultatima iz ovog dijela projekta raspraviti sa stručnjakom iz ovog područja.

Zanimljivo bi u daljnjem istraživanju bilo umjesto predviđanja samo binarnog ishoda hoće li poduzeće ući u postupak predstečajne nagodbe predviđati i trajanje predstečajne nagodbe i/ili iznos tražbine nakon prijeboja u predstečajnoj nagodbi. Također proširenje istraživanja je moguće u vidu dodavanja novih značajki koje će opisivati makroekonomske uvjete, izloženost pojedinog poduzeća drugim poduzećima i slično. Više o ovome u poglavlju Dodatni rad.

2.2 Dosadašnja istraživanja

Prvi pokušaji formalnog predviđanja stečaja javljaju se početkom 20. stoljeća kada su predloženi prvi ekonometrijski indikatori za opis mogućnosti predviđanja propasti poduzeća. Šezdesete godine predstavljaju prekretnicu u istraživanju ranog otkrivanja uzroka propasti poduzeća - počinju se primjenjivati statistički modeli u svrhu predviđanja stečaja (veliki naglasak stavljen je na generalizirane linearne modele). U doba kada su velike količine podataka postale dostupne ispostavilo se da tradicionalno korišteni linearni modeli ne mogu odraziti netrivijalne veze među ekonomskim pokazateljima. Od devedesetih godina 20. stoljeća umjetna inteligencija i strojno učenje postali su značajan smjer istraživanja predviđanja stečaja. Među najpopularniji pristupima su metoda potpornih vektora, neuronske mreže te u zadnje vrijeme ansambli klasifikatora. U nastavku slijedi par izdvojenih radova koji se bave sličnom problematikom kao i mi.

Zieba et al. su u [1] vrlo uspješno primijenili ansambl boosted stabala u svrhu predviđanja stečaja poljskih poduzeća. Svoj model su eksperimentom usporedili s pristupima uobičajenim u području predviđanja stečaja te dobili značajno bolje rezultate. Kao opis poduzeća oni koriste 64 financijska omjera korištena u integriranim financijskim modelima i financijskoj analizi, a izračunati su iz podataka koji se nalaze u financijskim izvještajima poduzeća (popis značajki se može naći u Table 2 iz [1]). Oni u svom radu predlažu korištenje dodatnih sintetičkih značajki u svrhu poboljšanja predikcijskih svojstava ansambla stabala. Sintetičke značajke računaju se u svakom koraku boostinga kombinirajući postojeće značajke nekom

od osnovnih aritmetičkih operacija $(+, -, *, /)$ (detalji o sintetičkim značajkama su u Sintetičke značajke). Evaluacijom važnosti značajki navode kao korisne u predviđanju stečaja poduzeća prilagođen udio kapitala u financiranju imovine, koeficijent tekuće likvidnosti i koeficijent obrtaja obveza. Također zaključuju da bi korisni mogli biti i koeficijenti profitabilnosti, koeficijent financijske poluge te neki drugi. Analizom važnosti sintetičkih značajki zaključuju da su važne operativne performanse, profitabilnost firme te financijska poluga.

Alfaro et al. u [2] uspoređuju uspješnost primjene AdaBoost strategije kombiniranja stabala u ansambl i neuronske mreže za problem predviđanja stečaja. Njihov eksperiment pokazao je superiornost ansambla stabala nad individualnom neuronskom mrežom. Kao prediktivne varijable koriste 13 uobičajenih financijskih omjera temeljenih na računovodstvenim podacima, ali i neke kvalitativne varijable kao što su veličina firme, sektor i vlasnička struktura (za razliku od pristupa gdje se poduzeća podijele u grupe po npr. sektoru kao u [1] oni sektor u kojem poduzeće posluje uzimaju kao prediktivnu varijablu). Točan popis tih varijabli može se naći u Table 2 iz [2]. U okviru neuronske mreže analizirali su relativni doprinos svake input varijable globalnim svojstvima pomoću analize osjetljivosti. Prema toj analizi došli su do zaključka da su najvažnije značajke redom po važnosti efikasnost (prodaja kroz ukupna imovina), sektor, varijabla koja procjenjuje veličinu firme, vlasnička struktura te razina zaduženosti. Analizom modela dobivenog AdaBoost strategijom zaključuju da su najvažnije značajke redom po važnosti ekonomska profitabilnost (pokazuje uspješnost firme u korištenju imovine), razina zaduženosti, efikasnost i varijabla koja procjenjuje veličinu firme. Zadnje četiri značajke pojavljuju se i kod neuronske mreže no u drugom redoslijedu po važnosti.

Rujoub et al. u [3] ocijenjuju korisnost varijabli vezanih uz novčani tok u predviđanju stečaja. Multrivarijatnom diskriminantnom analizom isputuju 3 hipoteze: diskriminantna sposobnost podataka o novčanom toku (u obliku financijskih omjera) za predviđanje stečaja statistički je značajna, točnost klasifikacije temeljene na informacijama o novčanom toku veća je od klasifikacijske točnosti temeljene na konvencionalno korištenim podacima iz računovodstva (kao što su koeficijenti profitabilnosti, zaduženosti i slično) te korištenje podataka o novčanom toku zajedno s tradicionalnim podacima iz računovodstva može poboljšati točnost klasifikacije za predviđanje stečaja. U testiranju tih hipoteza došli su do zaključaka da korištenje podataka o novčanom toku bolje predviđa stečaj nego tradicionalno korišteni računovodstveni podaci te da korištenje podataka o novčanom toku zajedno s tradicionalnim računovodstvenim podacima poboljšava moć predviđanja računovodstvenih podataka

korištenih u prethodnim istraživanjima.

Liang et al. u [4] ispituju diskriminatornu moć dobivenu kombiniranjem različitih kategorija financijskih koeficijenata (FR) i indikatora korporativnog upravljanja (CGI). Preciznije, uzeli su u obzir sedam kategorija FR-a (koeficijenti profitabilnosti, kapitalne strukture, obrtaja, novčanog toka te rasta) i pet kategorija CGI-a (struktura upravnog odbora, vlasnička struktura, prava od novčanih tokova te zadržavanje ključnog osoblja). Da bi odredili najbolju kombinaciju FR-a i CGI-a koristili su podatke o poduzećima u Tajvanu. Njihovi rezultati pokazuju da kombinacija FR-a i CGI-a može poboljšati performanse modela kada se usporede s modelom koji koristi samo FR-ove. Nadalje, zaključuju da su najvažnije značajke za predviđanje stečaja FR kategorije solventnost i profitabilnost te CGI kategorije struktura upravnog odbora i vlasnička struktura. No, korisnost CGI značajki ovisi o tržištu. Proveli su analognu analizu za kinesko tržište te došli do zaključka da prediktivne performanse kombinacije FR-a i CGI-a na tom tržištu nisu bolje od korištenja samo FR-a. Oni kao razloge tome navode drugačije definicije stečaja na tim tržištima te da je opseg u kojemu su CGI-ovi povezani s karakteristikama kompanije ovisan o tržištu.

3 Metodologija

3.1 Logistička regresija

Logistička regresija modelira posteriori vjerojatnosti klasa linearnim funkcijama u x (p-dimenzionalni vektor značajki).

3.1.1 Klasa modela

Model logističke regresije za dvije klase može se zapisati u formi

$$\log \frac{\Pr(Y=1|X=x)}{\Pr(Y=0|X=x)} = \beta_0 + \beta_1^T x,$$

gdje su β_0 i β_1^T parametri koje trebamo naučiti. Stavimo $\beta = (\beta_0, \beta_1^T)$ te $p(x;\beta) = \Pr(Y = 1|X = x;\beta)$. Jednostavnim računom vidimo da je tada

$$p(x;\beta) = \frac{\exp(\beta_0 + \beta_1^T x)}{1 + \exp(\beta_0 + \beta_1^T x)}.$$

3.1.2 Algoritam učenja

Modeli logističke regresije se obično uče metodom maksimalne vjerodostojnosti. Log-vjerodostojnost za N opažanja dana je s

$$l(\beta) = \sum_{i=1}^{N} [y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta))]$$

$$= \sum_{i=1}^{N} \left[y_i \beta^T x_i - \log(1+e^{\beta^T x_i}) \right],$$

gdje pretpostavljamo da vektor značajki x_i na prvom mjestu ima dodanu jedinicu koja odgovara slobodnom koeficijentu β_0.

Da bi maksimizirali log-vjerodostojnost izjednačavamo njene parcijalne derivacije s 0

$$\frac{\delta l(\beta)}{\delta \beta} = \sum_{i=1}^{N} x_i (y_i - p(x_i;\beta)) = 0$$

i dobivamo $p+1$ jednadžbi nelinearnih u β.

Da bi riješili gornju jednadžbu koristimo Newton-Raphsonov algoritam za koji nam je potrebna Hesseova matrica

$$\frac{\delta^2 l(\beta)}{\delta \beta \delta \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i;\beta)(1-p(x_i;\beta)).$$

Iteracija Newton-Raphsonove metode sada je dana s

$$\beta^{new} = \beta^{old} - \left(\frac{\delta^2 l(\beta)}{\delta \beta \delta \beta^T} \right)^{-1} \frac{\delta l(\beta)}{\delta \beta},$$

gdje su derivacije izračunate u β^{old}. Obično je $\beta = 0$ dobra početna vrijednost iako konvergencija nije garantirana. Tipično algoritam konvergira jer je log-vjerodostojnost konkavna funkcija, ali ponekad može doći do premašivanja (tj. overshootinga).

3.2 Metode stabla

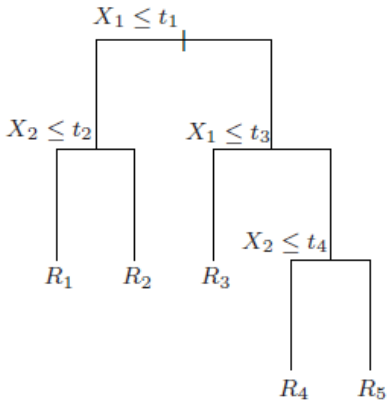
3.2.1 Klasa modela

Modeli stabla su jednostavni i interpretabilni modeli koji se mogu vizualizirati strukturom stabla. Oni obično imaju ograničene sposobnosti predviđanja pa se često kombiniraju u ansamble.

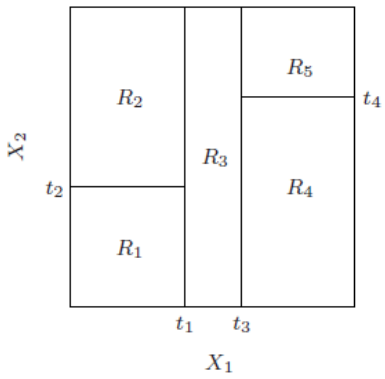
Modeli stabla dijele prostor značajki χ u skup od T pravokutnih, nepreklapajućih dijelova $R_1, ..., R_T$ te na svakom od tih dijelova prilagođavaju jednostavan model poput konstante (npr. srednju vrijednost varijable odziva primjera koji su u tom dijelu). Na slici 1 prikazan je primjer stabla, particije inducirane njime te vizualizacija dobivenog modela. Kada se koriste konstante (po dijelovima) model stabla se može zapisati u obliku

$$f(x) = \sum_{j=1}^{T} w_j I(x \in R_j).$$

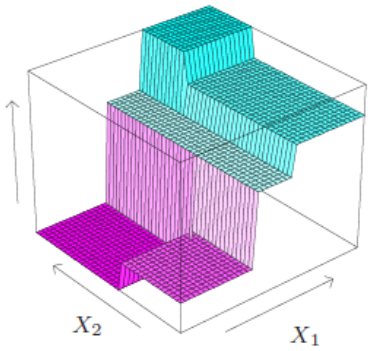
Dakle, model stabla je zapravo po dijelovima konstantna funkcija.



(a) Primjer stabla



(b) Particija dvodimenzionalnog prostora značajki koja odgovara tom stablu



(c) Vizualizacija modela

Slika 1: Slike preuzete iz [5] (Figure 9.2)

3.2.2 Algoritam učenja

Zbog jednostavnosti ovdje ćemo proučavati učenje stabla bez penalizacije. Tada je funkcija cilja jednostavno

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \sum_{j=1}^{T} w_j I(x \in R_j)).$$

Kada je poznata struktura stabla, tj. pravokutnici $R_1, ..., R_T$ optimalne težine $w_1, ..., w_T$ lako je naučiti. No vrlo je teško naći optimalne pravokutnike. Stoga problem pojednostavljujemo tako da tražimo približno rješenje. Postoje mnogi algoritmi učenja za učenje modela stabla. U nastavku je opis CART (Classification And Regression Trees) modela učenja (XGBoost implementacija Extreme Gradient Boostinga opisanog u 3.4.2 koristi algoritam vrlo blizak CART-u).

CART stvara stablo pohlepno "odozgo prema dolje" koristeći binarne podjele. Stvaranje počinje u korijenu stabla. Razmatraju se sve podjele paralelne s koordinatnim osima, tj. u obzir ulaze sve podjele određene s $x_i \leq s$ za neki $s \in \mathbb{R}$ i neki $i \in \{1, 2, ..., n\}$, ako je $(x_1, ..., x_n)$ vektor značajki. U sljedećem koraku razmatraju se sve podjele paralelne s koordinatnim osima unutar trenutno postojećih regija te se u svakom odabire najbolja. Taj proces se ponavlja sve do nekog kriterija zaustavljanja.

Za dan pravokutnik R_j odrediti težinu w_j obično nije problem. Označimo s I_j skup indeksa koje pripadaju pravokutniku R_j, tj. za $i \in I_j$ je $x_i \in R_j$. Tada je procjena težine

$$\hat{w_j} = \arg\min_{w} \sum_{i \in I_j} L(y_i, w).$$

Npr. ako za funkciju gubitka (L) uzmemo kvadrat greške tada će procjena težine biti prosjek odziva u tom području (tj. pravokutniku). Ako za funkciju gubitka uzmemo apsolutnu vrijednost greške tada će procjena biti medijan odziva.

Za model stabla $\hat{f}$ funkciju cilja možemo zapisati kao

$$\hat{R}(\hat{f}) = \sum_{j=1}^{T} \sum_{i \in I_j} L(y_i, \hat{w_j}) = \sum_{j=1}^{T} \hat{L}_j,$$

gdje smo s $\hat{L}_j$ označili ukupni gubitak na čvoru j. Pretpostavimo sada da smo u procesu učenja stabla te označimo trenutni model s $\hat{f}_{before}$. Nadalje označimo s $\hat{f}_{after}$ model stabla nakon što se razmatrana podjela u čvoru k na lijevi čvor L i desni čvor R ostvarila. Sada funkcije cilja ovih modela možemo zapisati kao

$$\hat{R}(\hat{f}_{before}) = \sum_{j \neq k} \hat{L}_j + \hat{L}_k$$

i

$$\hat{R}(\hat{f}_{after}) = \sum_{j \neq k} \hat{L}_j + \hat{L}_L + \hat{L}_R.$$

Gain ("dobit", za koliko smo tom podjelom smanjili ili povećali funkciju cilja koju minimiziramo) razmatrane podjele definira se kao

$$Gain = \hat{R}(\hat{f}_{before}) - \hat{R}(\hat{f}_{after}) = \hat{L}_k - (\hat{L}_L + \hat{L}_R).$$

Gain se računa za sve moguće podjele na svim čvorovima te se odabire podjela s najvećim gainom.

3.2.3 Regularizacija

Regularizacija za modele stabla se obično postiže ograničavanjem ili penaliziranjem kompleksnosti stabla. Postoje različiti načini za definiciju kompleksnosti modela stabla. Za kompleksnost možemo reći da ovisi o dubini stabla ili broju terminalnih čvorova stabla.

Kompleksnost općenito ovisi i o veličini lokalnih susjedstva, tj. u slučaju stabala o veličini pravokutnika $R_1, ..., R_T$. Stabla s manjim pravokutnicima mogu bolje opisati lokalnu strukturu bolje pa su stoga i kompleksniji.

Za kompleksnost možemo reći da ovisi i o relativnoj razlici težina listova $w_1, ..., w_T$. Zašto? Pretpostavimo da su sve težine stabla iste. Tada je taj model globalno konstantan. S druge strane, ako su težine jako različite model možemo smatrati kompleksnijim. Definiranje kompleksnosti na ovaj način nije uobičajeno za modele individualnih stabala, ali se koristi u XGBoost-u prilikom prilagođavanja aditivnih modela stabla.

Da bi kontrolirali fleksibilnost prilikom prilagođavanja modela stabla možemo ograničiti kompleksnost stabla. Postoje različiti načini kako ograničiti kompleksnost stabla. Jedan očiti i često korišten način za to je ograničiti broj terminalnih čvorova stabla (T_{max}). Drugi mogući način ograničenja kompleksnosti je ograničiti najmanji broj primjeraka dopuštenih u svakom terminalnom čvoru (n_{min}). To će nametnuti ogradu koliko mala susjedstva R_j mogu biti, a to pak direktno ograničava varijancu modela jer će više primjeraka biti potrebno da bi se procijenile težine listova.

Najčešći način penaliziranja kompleksnosti stabla je penalizirati broj terminalnih čvorova T. Funkcija cilja koju tada treba minimizirati dana je cost-complexity kriterijem

$$J(f) = \hat{R}(f) + \Omega(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \sum_{j=1}^{T} w_j I(x \in R_j)) + \gamma T,$$

gdje je γ hiperparametar poznat kao parametar kompleksnosti. Drugi način penalizacije kompleksnosti stabla je uvesti penalizaciju težina listova $w_1, ..., w_T$. Ovaj oblik penalizacije koristi se u XGBoost-u, ali se obično ne koristi u modelima s jednim stablom.

3.3 Metoda slučajne šume

Bagging (bootsrap aggregation) je tehnika smanjivanja varijance procijenjene funkcije predviđanja. Ta tehnika radi vrlo dobro za modele s visokom varijancom i niskim biasom, kao što su stabla. Za regresiju jednostavno prilagođavamo isto regresijsko stablo bootstrap uzorku iz trening skupa pa usrednjimo rezultate. Za klasifikaciju "odbor" stabala glasa za predviđenu klasu.

Slučajne šume su modifikacija bagging metode koja stvara veliki skup "dekoreliranih" stabala pa ih usrednjuje.

3.3.1 Opis modela

Osnovna ideja bagginga je usrednjiti mnogo modela s visokom varijancom, ali niskim biasom i tako smanjiti varijancu. Stabla su idealni kandidati za bagging jer ona mogu "uhvatiti" kompleksne strukture u podacima te, ako su dovoljno duboka, imaju relativno niski bias. Stabla imaju vrlo visoku varijancu pa imaju mnogo koristi od usrednjavanja. Nadalje, pošto je svako stablo generirano baggingom jednako distribuirano očekivanje srednje vrijednosti B takvih stabala je jednako očekivanju bilo kojeg od njih. To znači da je bias bagged stabala jednak biasu individualnih stabala pa je jedina nada za poboljšanje smanjenje varijance. Za razliku od bagginga u boostingu se stabla stvaraju na adaptivan način da bi se uklonio bias pa nisu jednako distribuirana.

Aritmetička sredina B nezavisnih jednako distribuiranih slučajnih varijabli od kojih svaka ima varijancu σ^2 ima varijancu $\frac{1}{B}\sigma^2$. Ako su varijable samo jednako distribuirane, ali ne nužno i nezavisne, s pozitivnom korelacijom ρ po parovima varijanca prosjeka je

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$$

Odavde vidimo da kako broj stabala B raste drugi član ovog izraza nestaje, ali prvi ostaje pa zaključujemo da veličina korelacije parova stabala utječe na koristi usrednjavanja. Glavna ideja slučajnih šuma je poboljšati smanjenje varijance bagginga smanjenjem korelacije među

stablima bez prevelikog povećanja varijance. To se postiže tako da se u procesu stvaranja stabala nasumično izabiru input varijable s kojima ćemo trenirati. Detaljnije o tome u sljedećem dijelu.

3.3.2 Algoritam učenja

Model iterativno učimo tako u svakom koraku učenja b prvo iz skupa za učenje izabiremo bootstrap uzorak $\mathbf{Z}^*$ veličine N. Na tom uzorku učimo stablo T_b tako za svaki terminalni čvor rekurzivno primjenjujemo sljedeće korake: 1) nasumično izabiremo m od p varijabli poticaja, 2) odabiremo najbolju varijablu (tj. podjelu) među tih m, 3) podijelimo čvor na dva čvora potomka. Na kraju ovog postupka imamo ansambl stabala $\{T_b\}_{b=1}^{B}$.

Predviđanje za novi primjerak x za regresiju je

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x),$$

a za klasifikaciju

$$\hat{C}(x) = \textit{većinski glas}\ \left\{\hat{C}_b\right\}_{b=1}^{B},$$

gdje je $\hat{C}_b(x)$ klasa predviđena b-tim stablom slučajne šume.

3.4 Metode boostinga stabala

3.4.1 Klasa modela

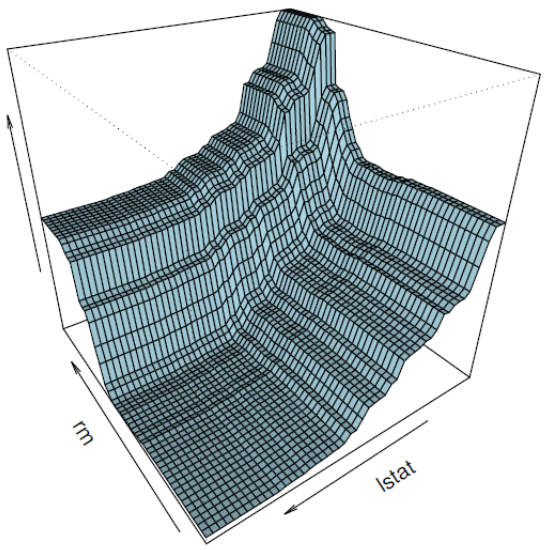
Model boosted stabala je zbroj stabala f_m opisanih u 3.2,

$$f(x) = \sum_{m=1}^{M} f_m(x).$$

Modeli boosted stabala se stoga ponekad nazivaju i ansambli stabala ili modeli aditivnih stabala. Na slici 2 prikazan je model aditivnih stabala prilagođen podacima o nekretninama u Bostonu.

3.4.2 Algoritam učenja

Boosting metode za stabla sekvencijalno u svakoj iteraciji dodaju trenutnom modelu stablo koje poboljšava fit cjelokupnog modela. Pretpostavimo za sada zbog jednostavnosti da nema



Slika 2: Vizualizacija modela aditivnih stabala (Slika preuzeta iz [6], Figure 6.1)

penalizacije. U svakoj iteraciji m minimiziramo kriterij

$$J_m(f_m) = \sum_{i=1}^{n} L(y_i, f^{(m-1)}(x_i) + f_m(x_i)),$$

gdje je $f^{(m-1)}$ trenutni model, a $f_m(x) = \sum_{j=1}^{T} w_{jm} I(x \in R_{jm})$ traženo stablo koje će biti dodano ansamblu. Račun u nastavku ovog dijela proveden je za općenitu funkciju L, no mi u ovom projektu za problem binarne klasifikacije ($y_i \in \{0,1\}$) koristimo logistički gubitak, tj. $L(y_i, f^{(m)}(x_i)) = y_i \log(1+\exp(-f^{(m)}(x_i))) + (1-y_i)\log(1+\exp(f^{(m)}(x_i)))$.

Postoji nekoliko boosting algoritama, a mi ćemo u nastavku opisati metodu koja se koristi u XGBoost algoritmu. Ta metoda poznata je kao Extreme Gradient Boosting. U toj metodi gornji kriterij aproksimira se s (Taylorov razvoj L oko $f^{(m-1)}(x_i)$ te zanemarivanje konstanti)

$$J_m(f_m) = \sum_{i=1}^{n} \left[g_m(x_i) f_m(x_i) + \frac{1}{2} h_m(x_i) f_m(x_i)^2 \right],$$

gdje je

$$g_m(x_i) = \frac{\delta L(y_i, f^{(m-1)}(x_i))}{\delta f^{(m-1)}(x_i)}$$

te

$$h_m(x_i) = \frac{\delta^2 L(y_i, f^{(m-1)}(x_i))}{\delta f^{(m-1)}(x_i)^2}.$$

Zapišimo gornju aproksimaciju kriterija uzimajući u obzir da je f_m stablo

$$J_m(f_m) = \sum_{i=1}^{n} \left[g_m(x_i) \sum_{j=1}^{T} w_{jm} I(x_i \in R_{jm}) + \frac{1}{2} h_m(x_i) \left(\sum_{j=1}^{T} w_{jm} I(x_i \in R_{jm}) \right)^2 \right].$$

Pravokutnici $R_{jm}, j = 1, ..., T$ su disjunktni pa dalje imamo

$$J_m(f_m) = \sum_{i=1}^{n} \left[g_m(x_i) \sum_{j=1}^{T} w_{jm} I(x_i \in R_{jm}) + \frac{1}{2} h_m(x_i) \sum_{j=1}^{T} w_{jm}^2 I(x_i \in R_{jm}) \right]$$

$$= \sum_{j=1}^{T} \sum_{i \in I_{jm}} \left[g_m(x_i) w_{jm} + \frac{1}{2} h_m(x_i) w_{jm}^2 \right],$$

gdje je I_{jm} skup indeksa i za koje $x_i \in R_{jm}$. Stavimo $G_{jm} = \sum_{i \in I_{jm}} g_m(x_i)$ i $H_{jm} = \sum_{i \in I_{jm}} h_m(x_i)$ pa imamo

$$J_m(f_m) = \sum_{j=1}^{T} \left[G_{jm} w_{jm} + \frac{1}{2} H_{jm} w_{jm}^2 \right].$$

Dakle, uz poznatu (fiksnu) strukturu stabla (tj. pravokutnike $R_{jm}, j = 1, ..., T$) težine su dane s

$$\tilde{w}_{jm} = -\frac{G_{jm}}{H_{jm}},$$

a vrijednost kriterija za te težine tada je

$$J_m(f_m) = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_{jm}^2}{H_{jm}}.$$

Kao što smo vidjeli u 3.2.2 učenje strukture stabla zapravo je traženje podjela. Tražimo podjelu koja minimizira gornju funkciju gubitka ili ekvivalentno maksimizira gain (smanjenje funkcije gubitka odabranom podjelom)

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G_{jm}^2}{H_{jm}} \right].$$

Ukratko, algoritam učenja ansambla se može opisati na sljedeći način. U svakoj iteraciji algoritma m odredimo strukturu stabla kojeg dodajemo ($\hat{f}_m$) postojećem modelu ($\hat{f}^{(m-1)}$) $\hat{R}_{jm}, j = 1, ..., T$ te zatim odredimo težine $\hat{w}_{jm}, j = 1, ..., T$ za tu strukturu. Stavimo $\hat{f}_m(x) = \eta \sum_{j=1}^{T} w_{jm} I(x \in \hat{R}_{jm})$ te $\hat{f}^{(m)}(x) = \hat{f}^{(m-1)}(x) + \hat{f}_m(x)$, gdje je $0 < \eta \leq 1$ hiperparametar, tzv. parametar sažimanja ili stopa učenja.

3.4.3 Regularizacija

Regularizacija modela aditivnih stabala može se postići na mnogo načina. Možemo regularizirati ekspanziju baznih funkcija (tj. broj stabala), svaku od baznih funkcija (tj. individualna stabla) te možemo uvesti regularizaciju kroz tzv. subsampling (”poduzorkovanje”).

Kompleksnost modela aditivnih stabala ovisi o nekoliko parametara. Kao prvo, broj stabala u ansamblu je očito povezan s kompleksnošću modela. Također, u obzir treba uzeti i kompleksnost individualnih stabala u ansamblu.

Stopa učenja ili parametar sažimanja η će smanjiti težine listova individualnog stabla naučenog u svakoj iteraciji. Ako postavimo parametar η previsoko naučit ćemo model puno strukture podataka u ranim iteracijama i time vrlo brzo povećavati varijancu modela. Stopa učenja povezana je s brojem stabala u ansamblu, smanjenjem parametra η potreban je veći broj stabala. Korištenje većeg broja stabala povećava reprezentacijsku sposobnost modela. Stoga smanjenjem parametra sažimanja modelu možemo dodati veći broj stabla (pa mu i povećati reprezentacijsku sposobnost) prije nego što počnemo overfitati podatke.

Ograničiti kompleksnost modela aditivnih stabala možemo tako da ograničimo broj stabala. Nadalje, možemo ograničiti broj terminalnih čvororva svakog pojedinačnog stabla ili možemo ograničiti minimalni broj primjeraka koji upadaju u neki terminalni čvor.

XGBoost nudi i mogućnost penalizacije kompleksnosti stabala (uz ograničavanje) te je to jedno od bitnijih poboljšanja XGBoosta u odnosu na prijašnje algoritme korištene u ovu svrhu. Prije XGBoosta penalizacija kompleksnosti nije se često koristila u modelima aditivnih stabala. Članovi funkcije cilja vezani uz penalizaciju mogu se zapisati kao

$$\Omega(f) = \sum_{m=1}^{M} \left[\gamma T_m + \frac{1}{2}\lambda \left\| w_m \right\|_2^2 + \alpha \left\| w_m \right\|_1 \right].$$

Dakle, penalizacija je zbroj penalizacija kompleksnosti pojedinačnih stabala. Ona uključuje penalizaciju broja terminalnih čvorova (parametar γ kontrolira količinu te penalizacije) te dodatno $l2$ i $l1$ regularizaciju težina čvorova (gdje parametri λ i α kontroliraju jačinu odgovarajućih penalizacija). To je i implementirano u XGBoost algoritmu. $l2$ i $l1$ regularizacija smanjuju težine listova te tako smanjuju varijancu u njihovoj procjeni. To je slično učinku stope učenja η no za razliku od η parametri λ i α varijarjući smanjuju težine (dok η smanjuje jednako sve težine). Najveća razlika je zapravo to što λ i α osim na same težine utječu i na strukturu stabla (kroz sudjelovanje u funkciji cilja $J(\theta) = L(\theta) + \Omega(\theta)$).

Mogućnost dodatne regularizacije je kroz subsampling, redaka ili stupaca. Subsampling

redaka realizira se tako da se u svakoj iteraciji boostinga model prilagođava nasumično izabranom poduzorku podataka. Tada ta frakcija (udio poduzorka u cijelom podatkovnom skupu) $0 < \omega_r \leq 1$ postaje hiperparametar boosting procedure. Slično, subsampling stupaca je nasumično biranje prediktora (stupaca) te i ovdje dobivamo hiperparametar frakcije izabranih stupaca $0 < \omega_c \leq 1$. U XGBoost algoritmu uključeni su subsampling i redaka i stupaca.

3.4.4 Ansambl boosted stabala za predviđanje stečaja

Procjenitelji ekonomskih indikatora koji opisuju kompanije karakterizirani su visokom varijancom uzrokovanom relativno malim uzorkom. U praksi to znači da je većina vrijednosti akumulirana u nekom uskom segmentu no postoje kompanije koje su opisane relativno visokim/niskim vrijednostima tih značajki. Stoga primjena gradijentnih modela kao što su neuronske mreže ili logistička regresija vodi do problema s treniranjem pa i lošim predviđanjem. Taj problem teško je riješiti i kada se podaci normaliziraju ili standardiziraju. Nasuprot ovim pristupima modeli temeljeni na ansamblu stabala uzimaju u obzir red vrijednosti značajki, a ne same vrijednosti. Stoga su otporni na ogromne vrijednosti ekonomskih indikatora i ne zahtijevaju preprocesiranje podataka.

Drugi značajan problem kod primjene metoda strojnog učenja na predviđanja stečaja je i fenomen neuravnoteženosti podataka - obično postoji puno više uspješnih kompanija nego onih u stečaju. XGBoost model nije osjetljiv na ovaj problem jer dopušta evaluaciju AUC mjerom te primorava pravilan poredak neuravnoteženih podataka.

Nedavno je pokazano da se ansambl klasifikator može uspješno primijeniti za predviđanje stečaja ([7]) te da je značajno bolji od drugih metoda ([2]).

3.5 Sintetičke značajke

Modeli ansambla stabala mogu se učinkovito naučiti na podacima koji su opisani s mnogo značajki. Zieba et al. u [1] predlažu korištenje dodatnih sintetičkih značajki u svrhu poboljšanja predikcijskih svojstava ansambla stabala. Sintetičke značajke računaju se u svakom koraku boostinga kombinirajući postojeće značajke nekom od osnovnih aritmetičkih operacija $(+, -, *, /)$.

Na svaku sintetičku značajku možemo gledati kao na jedan regresijski model. Drugi pogled na sintetičke značajke je da su one na neki način analogon skrivenih jedinica u neuronskim mrežama, ali način na koji se izvlače je potpuno drukčiji.

Svrha sintetičkih značajki je kombinirati ekonomske indikatore predložene od stručnjaka u kompleksne značajke. Te kompleksne značajke mogu imati bolji utjecaj na predviđanje nego tipični ekonomski faktori.

Sintetičke značajke generirane su slučajnim odabirom dvije postojeće značajke te slučajnim odabirom aritmetičke operacije koja će na njima biti izvršena. Vjerojatnost odabira neke značajke ovisi o popularnosti (broju pojavljivanja) te značajke u već postojećim stablima, što veća popularnost veća je vjerojatnost odabira te značajke. Konkretnije, definiramo kategorijsku distribuciju $\boldsymbol{\theta} = \left(\theta^{(1)}, ..., \theta^{(d)}, ..., \theta^{(D)}\right)$, tj. vektor vjerojatnosti odabira pojedine značajke, gdje

$$\theta^{(d)} = \frac{m_d}{\sum_{i=1}^{D} m_i},$$

a m_d je broj pojavljivanja d-te značajke u stablima ansambla. Na to možemo gledati kao na evolucijski pristup koji odabire "najsnažnije" roditelje značajke za potomak značajku. Operacija se uniformno bira iz skupa $\{+, -, *, /\}$.

3.6 Algoritam učenja modela kojim pristupamo problemu

Kao u [1] model kojim pristupamo problemu predviđanja ulaska poduzeća u postupka predstečajne nagodbe je ansambl boosted stabala treniran tehnikom Extreme Graditent Boosting (model i tehnika opisani u 3.4) uz korištenje sintetičkih značajki (opisanih u 3.5).

Postupak stvaranja ansambla opisan je s Algoritmom 1. U svakoj iteraciji učenja t postojećem modelu $\left\{f^{(1)}, ..., f^{(t-1)}\right\}$ dodajemo stablo $f^{(t)}$ koje učimo tehnikom Extreme Gradient Boosting koristeći trening skup $\mathscr{D}$.

U ovisnosti o važnostima značajki m_d, $d = 1, ..., D$ izračunatim iz dosad poznatog modela $\left\{f^{(1)}, ..., f^{(t)}\right\}$ nastavljamo učenje samo s onim značajkama za koje je m_d veće od praga ε. Dosad naučen model dalje koristimo da bi odredili popularnost značajki, tj. da procijenimo distribuciju $\boldsymbol{\theta}$.

Sintetičke značajke generiraju se na sljedeći način. Biramo dvije značajke f_1 i f_2 iz distribucije $\boldsymbol{\theta}$ te uniformno biramo operaciju $\circ$ iz skupa $\{+, -, *, /\}$. Nakon toga vrijednost nove značajke $f_{new} = f_1 \circ f_2$ računa se za sve primjere u trening skupu $\mathscr{D}$. Postupak stvaranja sintetičkih značajki nastavlja se sve dok ne postignemo željeni broj sintetičkih značajki D_{new}. Nakon toga proširujemo trening skup s dobivenim sintetičkim značajkama i na njemu dalje učimo stablo $f^{(t+1)}$.

Input: trening skup $\mathscr{D}$, broj sintetičkih značajki D_{new}, broj stabala u ansamblu T, prag prihvaćanja značajki ε

Output: skup stabala koja su u ansamblu $\left\{f^{(1)},...,f^{(T)}\right\}$

for $t = 1,...,T$ **do**

  Nauči $f^{(t)}$ koristeći $\mathscr{D}$ tehnikom Extreme Gradient Boosting;

  Ukloni iz $\mathscr{D}$ značajke za koje je $m_d < \varepsilon$;

  Procijeni θ iz ansambla $\left\{f^{(1)},...,f^{(t)}\right\}$;

  for $d = 1,...,D_{new}$ **do**

    Izaberi značajke f_1 i f_2 iz distribucije $\boldsymbol{\theta}$;

    Izaberi operaciju $\circ$ uniformno iz skupa $\{+,-,*,/\}$;

    Generiraj novu značajku $f_{new} = f_1 \circ f_2$;

    Proširi $\mathscr{D}$ s f_{new};

  end

end

return $\left\{f^{(1)},...,f^{(T)}\right\}$;

Algoritam 1: Učenje ansambla boosted stabala sa sintetičkim značajkama (Zieba et al. [1])

3.7 Metode evaluacije

3.7.1 Tehnike probira

Metode evaluacije su metode kojima procjenjujemo prediktivnu moć algoritma. Među empirijskim procjenama greške algoritma postoje razne tehnike probira (en. resampling) kojima dijelimo skup podataka na manje skupove na kojima onda procjenjujemo grešku. Tim tehnikama pripadaju Train & Test metoda, unakrsna validacija (en. Cross Validation) i LOOCV (en. Leave-One-Out-Cross-Validation).

Train & Test je najprihvaćenija metoda u strojnom učenju. Ta metoda dijeli podatkovni skup na dio na kojemu učimo model te na dio na kojem ga evaluiramo računajući koliko predikcije modela odstupaju od stvarnih podataka.

Kod k-struke unakrsne validacije, dijelimo skup podataka na k jednakih dijelova te model iterativno treniramo na podacima iz njih $k-1$, a validiramo na preostalom skupu. Na kraju izračunamo prosječnu grešku za svih k modela. Ovakva metoda je bolja od procjene samih

reziduala modela jer iz reziduala ne možemo procijeniti koliko dobro će model predviđati na novim podacima.

Slično unakrsnoj validaciji, LOOCV iterativno koristi samo jedan podatak za validiranje algoritma, a ostatak podataka za učenje algoritma te se na kraju opet računa prosječna greška.

Napomenimo da je konvencija koristiti Train & Test metodu te unakrsnu validaciju zajedno. Konkretnije, najprije podijelimo podatke na train i test skupove. Zatim koristimo unakrsnu validaciju na skupu za treniranje na kojem učimo potrebne parametre modela, a onda validiramo cjelokupni algoritam na testnom skupu.

3.7.2 Mjere uspješnosti učenja modela za klasifikaciju

Postoje razne mjere kojima dobivamo uvid u kvalitetu naučenog modela. U nastavku ćemo opisati neke od tih mjera.

Krivulja učenja

Krivulja učenja nam prikazuje napredak (smanjenje greške) s povećanjem broja primjera za učenje. Uz krivulju učenja korisno je prikazati i krivulju koja prikazuje promjenu greške na skupu za testiranje. Na slici 3 prikazan je primjer krivulje učenja (plavo).

Slika 3: Slika preuzeta iz [8]

Matrica konfuzije

Matrica konfuzije je tablica koja se koristi za evaluaciju učinkovitosti klasifikacijskog modela.

		Stvarna klasa: Pozitivni	Stvarna klasa: Negativni
Predviđeno modelom	Pozitivni	TP	FP
	Negativni	FN	TN

TP - true positives (broj stvarno pozitivnih primjera, točno predviđenih od strane modela)
FP - false positives (broj stvarno negativnih primjera, koji su netočno predviđeni od strane modela kao pozitivni)
TN - true negatives (broj stvarno negativnih primjera, koji su točno predviđeni od strane modela kao negativni)
FN - false negatives (broj stvarno pozitivnih primjera, koji su netočno predviđeni od strane modela kao negativni)

Osnovne evaluacijske mjere koje proizlaze iz matrice konfuzije
Iz matrice konfuzije proizlaze mnoge mjere koje nam daju bolju intuiciju o prediktivnoj moći algoritma.

Omjer točno klasificiranih primjera u odnosu na ukupan broj primjera

$$\text{Točnost (en. Accuracy)} = \frac{TP+TN}{TP+FP+TN+FN} \times 100$$

nam daje postotak točnih predviđanja našeg algoritma. To je vrlo česta i uobičajena mjera no ne uvijek i ono što nam treba. Točnost daje istu težinu (važnost) pozitivnim i negativnim primjerima, a nekada su pozitivni ili negativni primjeri važniji. U tom slučaju informativnije su neke druge mjere, npr. osjetljivotsti i specifičnost.

Omjer pozitivnih primjera koje je model prepoznao kao pozitivne, od ukupnog broja pozitivnih primjera

$$\text{Osjetljivost (en. Sensitivity/Recall/True positive rate)} = \frac{TP}{TP+FN}$$

nam govori koliko je algoritam dobar u pronalasku pozitivnih primjera.

Omjer dobro prepoznatih negativnih primjera, od ukupnog broja negativnih primjera

$$\text{Specifičnost (en. Specificity/True negative rate)} = \frac{TN}{FP+TN}$$

nam govori koliko dobro raspoznajemo negativne primjere.

Osjetljivost i specifičnost nam opisuju koliko dobro algoritam razlikuje, odnosno diskriminira između pozitivne i negativne klase.

Omjer stvarno pozitivnih primjera koje je model prepoznao kao pozitivne, od ukupnog broja pozitivno predviđenih primjera

$$\text{Preciznost (en. Precision)} = \frac{TP}{TP+FP}$$

nam govori koliko je izgledno da je primjer klasificiran kao pozitivan stvarno pozitivan.

F-mjera

Obično preciznost pada kako osjetljivost raste. Htjeli bismo imati ravnotežu među njima pa stoga uvodimo tzv. $F-$mjeru

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R} = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}},$$

gdje je P preciznost, R osjetljivost algoritma, a β relativna važnost preciznosti u odnosu na osjetljivost. Druga jednakost slijedi za $\alpha = \frac{1}{1+\beta^2}$ i iz nje vidimo da je F_β težinska harmonijska sredina P i R. Još jedan razlog uvođenja ovakve mjere je sažimanje mjere preciznosti i osjetljivosti u jedan broj s ciljem lakše usporedbe raznih modela.

Obično se koristi mjera

$$F_1 = \frac{2PR}{P+R}$$

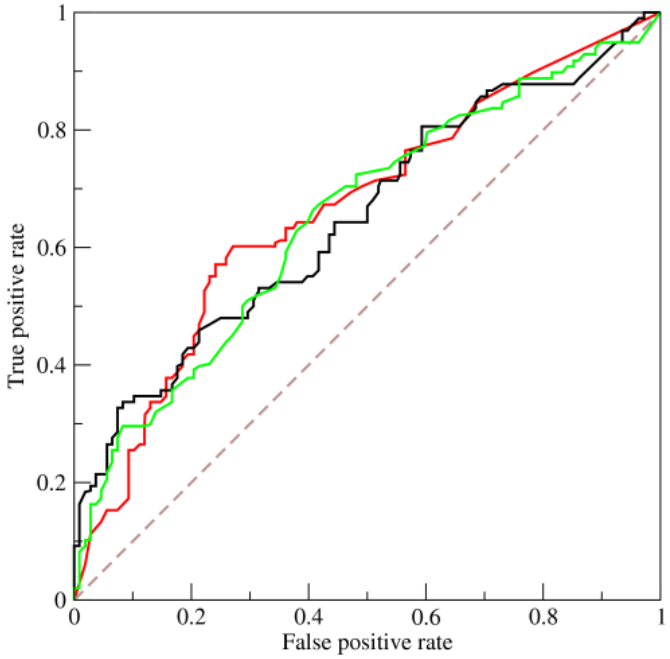
koja predstavlja harmonijsku sredinu P i R i ponaša se kao aritmetička sredina kada su P i R relativno blizu. Kada P i R nisu blizu, tada je harmonijska sredina manja od aritmetičke što odražava činjenicu da želimo dobru uravnoteženost između preciznosti i osjetljivosti. Želimo da ta mjera bude što bliža 1 jer za te vrijednosti imamo vrlo dobru i preciznost i osjetljivost algoritma.

ROC krivulja

Krivulja koja prikazuje odnos TPR (True positive rate) odnosno broj korektnih klasifikacija u (pozitivnoj) klasi u odnosu na ukupan broj pozitivnih primjera i FPR (False positive rate) odnosno broj krivih klasifikacija u (pozitivnoj) klasi u odnosu na ukupan broj negativnih primjera zove se ROC (en. Reciever Operating Characteristic) krivulja. Stoga možemo reći da ROC krivulja prikazuje relativni trade-off između koristi (TP) i troškova (FP).

$$TPR = \text{Osjetljivost} = \frac{TP}{TP+FN}$$

$$FPR = 1 - \text{Specifičnost} = \frac{FP}{FP + TN}$$



Slika 4: Primjer 3 ROC krivulje

Kako mijenjamo prag klasifikacije između pozitivnih i negativnih primjera za algoritam tako dobivamo različite omjere TPR i FPR koje možemo prikazati na grafu. Najbolja moguća predikcija bi se nalazila u gornje ljevom kutu gdje bi imali savršen klasifikator s osjetljivosti i specifičnosti jednakima jedan. Simetrala kvadranta predstavlja beznačajan algoritam (slučajni klasifikator). ROC krivulja pokazuje sposobnost rangiranja pozitivnih primjera relativno u odnosu na negativne.

Površina ispod ROC krivulje

Za uspoređivanje klasifikatora željeli bismo sažeti ROC performansu u jedan broj. Često korištena metoda je površina ispod ROC krivulje (en. Area Under ROC curve, AUROC, AUC). Ona nam daje vjerojatnost da algoritam nasumično izabran pozitivan primjer rangira iznad (smatramo da ima veću vjerojatnost da bude klasificiran kao pozitivan) od nasumično izabranog negativnog primjera. Ta površina korisna mjera jer je invarijantna obzirom na prag klasifikacije algoritma. Nasumično pogađanje stvara ROC krivulju koja je linija između točaka $(0,0)$ i $(1,1)$ što odgovara površini 0.5. Stoga nijedan smisleni klasifikator ne bi trebao imati AUC manji od 0.5.

4 Eksperiment

Garcia et al. u [13] rade pregled više od 140 radova u području kreditnog rizika i predviđanja propadanja poduzeća s naglaskom na dizajn eksperimenta. Oni ističu kako dizajn eksperimenta ima važnu ulogu u validaciji performansi (iako dosta istraživača ne posvećuje dovoljno pažnje tome) pa ga stoga treba oprezno izvesti da bi dobiveni rezultati bili značajni. U tome radu navode podatkovni skup, metode podijele podataka, mjere evaluacije performansi te statističke testove značajnosti kao četiri ključne komponente eksperimenta. U ovom području dizajn eksperimenta posebno je važan, ali i vrlo izazovan zbog posebnosti u odnosu na druga područja kao što su neuravnoteženost podataka, relativno mali podatkovni skupovi, asimetrični trošak FP i FN grešaka te uzimanje u obzir više mjera uspješnosti koje se možda suprotstavljaju. Stoga te posebnosti treba uzeti u obzir pri dizajnu eksperimenta jer one mogu značajno utjecati na njegove rezultate.

4.1 Podatkovni skup

S ciljem evaluacije kvalitete našeg pristupa prikupili smo podatke o hrvatskim poduzećima. Podatke smo prikupili iz financijskih izvještaja poduzeća u Hrvatskoj. Financijska izvješća se sastoje od bilance, računa dobiti i gubitka te novčanog toka. Uz to za svako poduzeće dostupni su razni opisni parametri kao što su sektor u kojem poduzeće posluje, županija poduzeća, veličina poduzća i drugi no njih za sada ne koristimo u treniranju. Uzorak poduzeća korišten u ovom radu sastoji se od poduzeća koja su završila u predstečajnoj nagodbi te onih koji nisu. Kao što je već napomenuto radi se o neuravnoteženom uzorku jer postoji puno više poduzeća koja nisu završila u postupku predstečajne nagodbe nego onih koji jesu.

Proces selekcije podataka koje ćemo koristiti za učenje i evaluaciju sastoji se od specificiranja sektora, perioda u kojem promatramo poduzeća, broja poduzeća te broja financijskih i drugih indikatora koje ćemo uzeti u obzir.

Na temelju prikupljenih podataka, kao što smo napomenuli u Uvodu, razlikujemo 5 različitih problema klasifikacije u ovisnosti o 5 različitih perioda predviđanja. Na primjer, za klasifikacijski problem predviđanja ulaska u postupak predstečajne nagodbe nakon 5 godina uzeli smo značajke o poduzeću iz financijskih izvješća iz određene godine i odgovarajuću oznaku koja nam govori je li poduzeće završilo u predstečaju nakon 5 godina. Konkretnije, s trenutno dostupnim podacima za ovaj problem gledali smo financijske izvještaje iz 2007., 2008. i 2009. godine te je li poduzeće završilo u predstečaju u 2012., 2013. i 2014. godini,

respektivno.

4.2 Eksperimentalni setup

Cilj eksperimenta je evaluirati našu metodu (model treniran Algoritmom 1) na podacima o hrvatskim poduzećima opisanim u prethodnom dijelu. Također, usporediti ćemo ju s klasifikacijskim metodama logističke regresije (LR), stabla odluke (DT) i metodom slučajne šume (RF) koje su često korištene u području predviđanja propadanja poduzeća.

Način evaluacije i usporedbe opisan je u sljedećem dijelu.

4.3 Evaluacija modela

Kao što je već objašnjeno u poglavlju Podatkovni skup, podaci su neuravnoteženi, tj. postoji mnogo više poduzeća koja nisu završila u predstečajnoj nagodbi. Stoga ako želimo evaluirati modele nad ovakvim podacima, potrebna je mjera koja nije osjetljiva na neuravnoteženost podataka.

ROC krivulje imaju takvo svojstvo - nisu osjetljive na promjenu u distribuciji klasa. Ako se proporcija pozitivnih i negativnih primjera u testnom skupu promijeni ROC krivulja se neće promijeniti. Da bi vidjeli zašto je to tako pogledajmo ponovno matricu konfuzije. Uočimo da je distribucija klasa (omjer pozitivnih i negativnih primjera) zapravo odnos lijevog i desnog stupca matrice konfuzije. Stoga će svaka mjera uspješnosti koja koristi vrijednosti iz oba stupca biti osjetljiva na neuravnoteženost klasa. Takve mjere su npr. točnost, preciznost i F-mjera pa će promjena u distribuciji klasa promijeniti i vrijednosti tih mjera iako se fundamentalna sposobnost predviđanja ne mijenja. ROC krivulje se temelje na TPR i FPR koji proizlaze iz jednog stupca pa stoga ne ovise o distribuciji klasa. Konkretnije, iz izraza za TPR vidimo da ako povećamo pozitivne primjere za neki faktor, povećali bi se i TP i FN, što znači da ne bi došlo do promjene TPR-a. Analogno, iz izraza za FPR vidimo da ako povećamo negativne primjere za neki faktor, FPR se ne bi mjenjao. Iz ovoga možemo zaključiti da ni površina ispod ROC krivulje nije osjetljiva na neuravnotežene podatke.

AUC mjeru također možemo smatrati primjerenom za ovo područje u kojemu različite vrste greške (FP greške i FN greške) imaju različite troškove ili posljedice.

Opišimo detaljnije kako koristimo tehnike probira i AUROC mjeru. Prvi korak je iskorostiti Train & Test metodu, tj. podijeliti podatke na skup za treniranje i skup za testiranje. Zatim koristimo unakrsnu validaciju na skupu za treniranje. Konkretnije, koristimo desete-

rostruku unakrsnu validaciju s AUROC mjerom. To znači da podijelimo skup za treniranje na 10 jednakih dijelova te iterativno treniramo na 9 skupova dok na preostalom skupu validiramo algoritam. Za tu validaciju koristimo AUROC mjeru. To znači da na kraju postupka unakrsne validacije imamo 10 AUROC mjera koje ćemo uprosječiti i tako dobiti konačnu ocjenu. Nakon toga provest ćemo i dodatnu ocjenu (također AUROC mjerom) na spomenutom izoliranom testnom skupu.

4.4 Interpretacija

Nakon prilagodbe modela razmatranjem njegove strukture možemo uvidjeti koje ekonomske mjere bitno sudjeluju pri donošenju predviđanja unutar modela. U [1] gledali su učestalost pojavljivanja određene značajke u stablima ansambla kao indikator te važnosti i tako došli do zaključka na koje ekonomske indikatore bi bilo dobro obratiti pozornost pri analizi poduzeća u svrhu predviđanja propadanja. Planiramo na taj način analizirati dobiveni model te usporediti zaključke s "tradicionalnim" ekonomskim znanjem. Zanimljivo bi bilo čuti mišljenje ekonomskih stručnjaka i ljudi koji se u praksi bave ovim problemom o dobivenim rezultatima.

5 Dodatni rad

U ovom projektu za početak se bavimo osnovnim problemima koji su opisani u ovom prijedlogu. Zadaci se mogu na nekoliko načina proširiti kako bi se naš pristup poboljšao te da se riješe neki dodatni zanimljivi problemi. Nakon ostvarenja ciljeva postavljenih ovdje planiramo poboljšati i proširiti pristup na nekoliko načina opisanih u nastavku.

Kao što smo već spomenili u Uvodu klasifikacijski zadatak može se redefinirati tako da umjesto binarnog ishoda poduzeće je/nije ušlo u postupak predstečajne nagodbe pokušavamo predvidjeti duljinu trajanja predstečajne nagodbe ili iznos tražbine nakon prijeboja u predstečajnoj nagodbi (kao regresijski ili klasifikacijski problem).

Druga mogućnost proširenja, uz dostupnost potrebnih podataka, je dodavanje dodatnih značajki kao što su npr. neki makroekonomski faktori ili značajke iz mreže vjerovnika o međusobnim dugovanjima među promatranim poduzećima i slično.

Što se tiče evaluacije uspješnosti našeg pristupa bilo bi korisno više pažnje posvetiti usporedbi našeg pristupa s s ostalim pristupima korištenim u području. Mogli bismo uzeti u obzir i neke druge mjere uspješnosti uz AUROC. Također, važno je uzeti u obzir da je supe-

riornost nekog predikcijskog modela temeljena na nekoj mjeri uspješnosti na testnom skupu naivan rezultat koji nije dovoljan da bi garantirao da taj model sigurno ima bolje performanse od drugih modela. Za potpunu evaluaciju performansi važno je napraviti neku vrstu testiranja hipoteza kako bi utvrdili da su uočene razlike u performansama statistički značajne, a ne samo slučajna posljedica ekperimentalnog setupa ([13]). Npr. u [12] koriste Friedmanov test za rangiranje algoritama za mjere uspješnosti točnost i AUROC u području credit-scoringa.

Kao što smo vidjeli u 3.4.3 u modelu s kojim pristupamo problemu u ovom projektu pojavljuje se značajan broj hiperparametara pa bi u budućem istraživanju trebalo posvetiti više vremena odabiru modela, tj. optimizaciji hiperparametara. Postoji nekoliko tehnika kojima se može pristupiti ovom problemu. Xia et al. u [10] kao i mi primjenjuju XGBoost metodu, ali za problem credit-scoringa. U svome radu oni su proveli optimizaciju hiperparametara s nekoliko metoda te došli do rezultata da Bayesovska optimizacija hiperparametra radi bolje nego metode slučajnog pretraživanja, pretraživanja rešetke i ručnog pretraživanja.

Također, bilo bi dobro posvetiti više vremena interpretaciji modela. Individualna stabla vrlo su interpretabilna no nemaju veliku sposobnost predviđanja dok ansambli imaju puno bolje predikcijske sposobnosti, ali cijena toga je veliki gubitak interpretabilnosti. Postoje načini poboljšanja interpretabilnosti modela stabala. Hara et al. u [9] predlažu metodu postprocesiranja koja poboljšava interpretabilnost ansambla stabala. Metoda se sastoji od toga da se nakon što je naučen kompleksan model ansambla on aprokisimira jednostavnijim modelom koji je interpretabilan.

6 Literatura

[1] Maciej Zieba, Sebastian K. Tomczak, Jakub M. Tomczak: *Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction*, Expert Systems with Applications, str: 93-101

[2] Esteban Alfaro, Noelia Garcia, Matias Gamez, David Elizondo: *Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks*, Decision Support Systems, str: 110-122

[3] Mohamed A. Rujoub, Doris M. Cook, Leon E. Hay: *Using Cash Flow Ratios To Predict Business Failures*, Journal of Managerial Issues, str: 75-90

[4] Deron Liang, Chia-Chi Lub, Chih-Fong Tsaic, Guan-An Shiha: *Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study*, European Journal of Operational Research, str: 561-572

[5] Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie: *The Elements of Statistical Learning*, Springer Series in Statistics

[6] Didrik Nielsen: *Tree Boosting With XGBoost*

[7] Loris Nanni, Alessandra Lumini: *An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring*, Expert Systems with Applications, str: 3028-3033

[8] Tomislav Šmuc: *Evaluacija modela*, Predavanja iz kolegija Strojno učenje

[9] Satoshi Hara, Kohei Hayashi: *Making Tree Ensembles Interpretable*

[10] Yufei Xia, Chuanzhe Liu, YuYing Li, Nana Liu: *A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring*, Expert Systems with Applications, str: 225-241

[11] Tom Fawcett: *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*

[12] Joaquín Abellan, Javier G. Castellano: *A comparative study on base classifiers in ensemble methods for credit scoring*, Expert Systems with Applications, str: 1-10

[13] Vicente Garcia, Ana I. Marques, J. Salvador Sanchez: *An insight into the experimental design for credit risk and corporate bankruptcy prediction systems*, Journal of Intelligent Information Systems, str: 159-189