Klasifikacijska stabla

Damiš, Anja

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Anja Damiš

KLASIFIKACIJSKA STABLA

Diplomski rad

Voditelj rada:
prof. dr. sc. Anamarija Jazbec

Zagreb, rujan, 2016.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

Statistika je neprestano izazvana problemima u mnogim područjima znanosti i industrije. Generiranje ogromnih količina podataka beskorisno je ako iz njih nešto ne možemo naučiti i dati im smisao. Učenje iz podataka je izdvajanje važnih obrazaca i trendova kao i razumijevanje onoga što podaci govore. Problemi u učenju koje razmatramo mogu se ugrubo kategorizirati u nadzirane probleme i probleme bez nadzora. Kod učenja bez nadziranja nema izlaznog mjerenja i cilj je opisati kako su podaci organizirani i grupirani. U nadziranom učenju cilj je predvidjeti vrijednost mjerenja ishoda na temelju broja ulaznih mjerenja. Zovu se nadzirani jer je prisutna izlazna varijabla koja vodi proces učenja. Na temelju prikupljenih podataka možemo istrenirati neku metodu kao i testirati njezinu preciznost. Podaci se sastoje od mjernih značajki za skup objekata (npr. ljudi) i opaženih ishoda. Korištenjem ovih podataka izrađujemo prediktivni model koji će nam omogućiti predviđanje ishoda za nove objekte. Dobar prediktivni model je onaj koji precizno predviđa ishod. U tipičnom scenariju imamo kvantitativna (npr. cijena dionica) ili kategorijska (npr. ima infarkt/nema infarkt) izlazna mjerenja, koja želimo predvidjeti na temelju skupa značajki (npr. dijeta i klinička mjerenja). Slijedi nekoliko primjera problema učenja:

- Identificiranje neželjene pošte na temelju frekvencije nekih riječi i znakova u samoj poruci.
- Predviđanje potrebe navodnjavanja poljoprivrednog zemljišta na temelju klimatskih, oborinskih podataka i podataka o kulturi koja se uzgaja.
- Predviđanje hoće li pacijent, hospitaliziran radi srčanog udara, imati drugi srčani udar na temelju demografskih obilježja, dijete i kliničkih mjerenja pacijenta.
- Identificiranje brojeva u ručnom zapisu ZIP koda s digitalne fotografije.

Navedeni primjeri predstavljaju nadzirane probleme. Razlika u tipovima izlaznih varijabli dovodi nas do konvencije u nazivlju vezane uz prediktivne zadatke: regresija kada predviđamo kvantitativni ishod i klasifikacija kada predviđamo kvalitativni ishod. Jedan od mogućih pristupa modeliranja klasifikacijskog problema je metoda klasifikacijskih stabla. To je konceptualno jednostavna metoda, osobito za interpretaciju i vizualizaciju rezultata,

temeljena na binarnim stablima. Problem je u pronalaženju uvijeta na podatke prema kojima kreiramo samo stablo i u postizanju optimalnog stabla, tj. onog koje će proizvoditi minimalnu grešku na novim podacima koji nisu korišteni prilikom njegove izrade.

Poglavlje 1

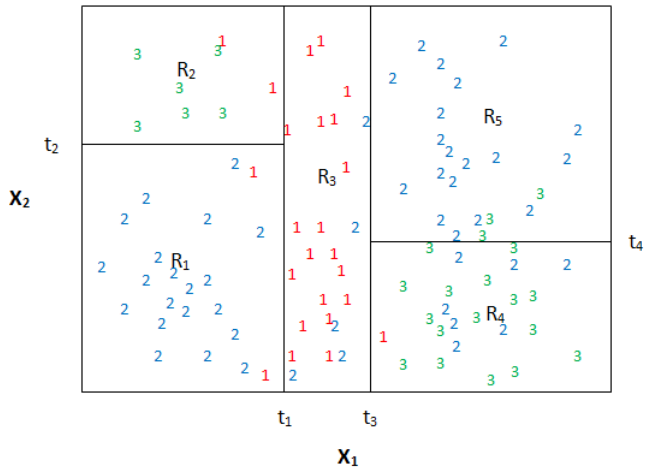
Metode bazirane na stablu

Razmotrimo klasifikacijski problem s tri razreda i ulaznim varijablama X_1 i X_2, a prostor značajki je skup svih mogućih vrijednosti od X_1 i X_2. Particioniramo ga rekurzivnim binarnim dijeljenjem u skup pravokutnika. Tada prilagođavamo jednostavni model (poput konstante) u svaki od pravokutnika. Prvim cijepanjem dobivamo 2 područja i modeliramo odziv u svakom području najučestalijom kategorijom koja se pojavljuje, za razliku od regresijskog stabla gdje bi odziv modelirali srednjom vrijednosti varijable odziva. Odabiremo varijablu i točku cijepanja tako da ostvarimo najbolju prilagodbu podataka. Tada su jedno ili oba područja podijeljena u još dva područja i taj proces se nastavlja dok se ne primijeni neko pravilo zaustavljanja. U primjeru ilustriranom na slici 1.1 prvo cijepamo u $X_1 = t_1$. Tada smo područje gdje je $X_1 \leq t_1$ podijelili u $X_2 = t_2$ i područje $X_1 > t_1$ je podijeljeno u $X_1 = t_3$. I konačno je područje gdje je $X_1 > t_3$ podijeljeno u $X_2 = t_4$. Rezultat ovog procesa je particija na 5 područja $R_1, R_2, ..., R_5$.

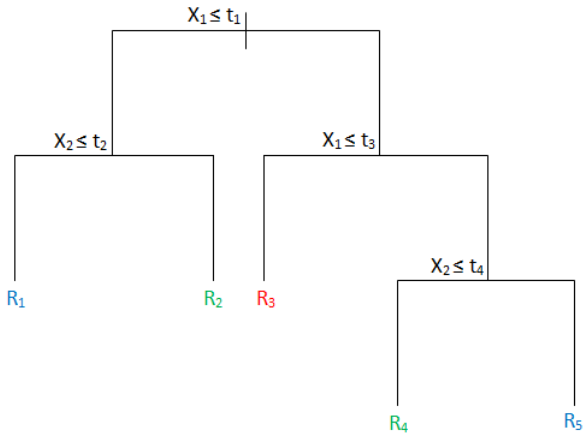
Odgovarajući klasifikacijski model predviđa kategorijsku varijablu Y s konstantom c_m u području R_m:

$$\hat{f}(X_1, X_2) = \sum_{m=1}^{5} c_m \mathbb{1}_{\{(X_1,X_2)\in R_m\}}. \tag{1.1}$$

Isti model možemo prezentirati binarnim stablom. Ukupni skup podataka leži u korijenu stabla. Opservacije koje zadovoljavaju uvijet pri svakom čvoru dodijeljuju se lijevoj grani, a preostale desnoj grani. Krajnji čvorovi ili listovi stabla predstavljaju područja $R_1, R_2, ..., R_5$ za koja modeliramo odzive najučestalijim kategorijama u tim područjima (slika 1.2). Particija prostora značajki u potpunosti je opisana jednim stablom čak i kada imamo više od dvije ulazne varijable. Umjesto da cijepamo svaki čvor u samo dvije grupe u svakoj fazi, mogli bi razmotriti mnogostruko cijepanje u više od dvije grupe. Dok to ponekad može biti korisno, nije dobra općenita strategija. Problem je u tome što višestruka cijepanja fragmentiraju podatke prebrzo, ostavljajući nedovoljno podataka na sljedećoj razini ispod. Osnovna prednost rekurzivnih binarnih stabala, koja još zovemo i stabla odlučivanja



Slika 1.1: Particija prostora značajki na područja $R_1, R_2, ..., R_5$ u kojima modeliramo odzive redom s $c_1 = 2$, $c_2 = 3$, $c_3 = 1$, $c_4 = 3$ i $c_5 = 2$.



Slika 1.2: Klasifikacijsko stablo za dvije ulazne varijable X_1 i X_2 s tri razreda 1, 2 i 3 čija je učestalost za svako područje prikazana odgovarajućom bojom.

je u interpretativnosti. Stabla su sličnija uobičajenom procesu donošenja odluka od nekih drugih pristupa. Klasifikacijsko stablo izrađeno je na temelju dostupnih podataka kako bi za nove vrijednosti ulaznih varijabli mogli predvidjeti kategoriju propuštanjem podataka kroz stablo. Implementacija popularne metode za klasifikaciju i regresiju bazirana na stablu je opisana u sljedećem poglavlju.

Poglavlje 2

Implementacija klasifikacijskih stabla

Algoritam za implementaciju klasifikacijskih stabla temelji se na CART (engl. *classification and regression tree*) metodi. Skup podataka za učenje se sastoji od p ulaznih varijabli $X_1, X_2, ..., X_p$ i kategorijskog odziva Y koji poprima vrijednosti $1, 2, ..., K$ za svaku od n opservacija. Podijelimo taj skup na skup podataka za trening, nad njim kreiramo stablo, i na skup podataka za testiranje na kojem opažamo grešku dobivenog stabla. Izrada klasifikacijskog stabla odgovara problemu particioniranja prostora značajki (skupa svih mogućih vrijednosti za ulazne varijable) na l različitih nepreklapajućih područja. Iako bi u teoriji ta područja mogla imati bilo kakav oblik, zbog jednostavnosti i interpretacije rezultata dijelimo prostor značajki na višedimenzionalne pravokutnike $R_1, R_2, ..., R_l$. Algoritam mora automatski odlučiti koje su to varijable cijepanja i točke cijepanja kao i kojeg oblika stablo treba biti.

Koraci algoritma

1. Izrada velikog stabla rekurzivnim binarnim dijeljenjem na skupu podataka za trening zaustavljajući se tek kad svaki krajnji čvor ima manje od zadanog broja opservacija.

2. Podrezivanje velikog stabla kako bismo dobili niz najboljih podstabala.

3. a) Odabir podstabla koje proizvodi najmanju grešku na skupu podataka za testiranje.

   b) Ukoliko nemamo dovoljno podataka da ih podijelimo na podatke za treniranje i testiranje koristimo krosvalidaciju za procjenu greške dobivenih podstabala na temelju čega odaberemo finalno stablo.

2.1 Kreiranje klasifikacijskog stabla

Kod kreiranja stabla primijenjujemo pristup odozgo prema dolje. Počinjemo na vrhu stabla kada sve opservacije pripadaju jednom području, a svakim cijepanjem nastaju 2 čvora djeteta ispod čvora cijepanja. U svakom koraku odabire se najbolje cijepanje bez obzira na to što će to cijepanje prouzročiti u sljedećem koraku, dakle radi se o pohlepnom algoritmu (engl. *greedy algorithm*) kojeg zovemo rekurzivno binarno dijeljenje. Najbolje cijepanje znači da smo proizveli čvorove koji sadrže opservacije iz istog razreda. Skup mogućih cijepanja koja uzimamo u obzir definiran je na sljedeći način:

- Svako cijepanje ovisi o vrijednosti samo jedne ulazne varijable.
- Cijepanje varijable X_j, $j \in \{1, 2, ..., p\}$ je tipa $X_j \leq s$, za neki $s \in (-\infty, \infty)$.

Lako je vidjeti da postoji konačan broj različitih cijepanja podataka. Ako su vrijednosti varijable X_j poredane po veličini, imamo najviše n različitih vrijednosti pa imamo najviše $n - 1$ različitih mogućnosti za odabir mjesta cijepanja. Točka cijepanja s jednaka je srednjoj vrijednosti dviju uzastopnih vrijednosti varijable X_j između kojih je odabrano mjesto cijepanja. Prostor značajki se time podijeli na dvije poluravnine:

$$R_1(j, s) = \{X; X_j \leq s\} \quad \text{i} \quad R_2(j, s) = \{X; X_j > s\}, \tag{2.1}$$

koje predstavljaju dva nastala čvora djeteta. Ponavljamo proces cijepanja na svako od ta dva područja. Tada se proces ponavlja na sva nastala područja. Skeniranjem kroz sve ulazne varijable lako je pronaći najbolji par varijable i točke cijepanja koristeći jedan od sljedećih kriterija za utvrđivanje nečistoće čvora, a pomoću njega i kvalitetu pojedinog cijepanja.

Mjere nečistoće čvorova i kvaliteta cijepanja čvora

Neka (x_i, y_i) predstavlja opservaciju za $i = 1, 2, ..., n$ s time da je $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$. Označimo s $\hat{p}_{mk}$ relativnu frekvenciju opservacija koje pripadaju razredu k u čvoru m:

$$\hat{p}_{mk} = \frac{1}{n_m} \sum_{x_i \in R_m} \mathbb{1}_{\{y_i = k\}}(x_i), \tag{2.2}$$

gdje je R_m područje koje predstavlja čvor m, a n_m broj opservacija u tom čvoru. Kriterij koji koristimo prilikom odabira mjesta cijepanja je mjera nečistoće čvora koju označavamo s $Q(m)$. To je funkcija relativnih frekvencija u čvoru m, $Q(m) = \phi(\hat{p}_{m1}, \hat{p}_{m2}, ..., \hat{p}_{mk})$ koja mora zadovoljiti sljedeće zahtjeve:

1. ϕ poprima maksimalnu vrijednost kada su opservacije u čvoru m jednako zastupljene, tj. $\hat{p}_{mk} = \frac{1}{n_m}, \forall k$.

2. ϕ poprima minimalnu vrijednost kada sve opservacije u čvoru m pripadaju istom razredu k, tj. $\hat{p}_{mk} = 1$ i $\hat{p}_{mk'} = 0, \forall k' \neq k$.

3. ϕ je simetrična funkcija relativnih frekvencija $\hat{p}_{m1}, \hat{p}_{m2}, ..., \hat{p}_{mk}$.

Najzastupljeniji razred u čvoru m definiramo na sljedeći način:

$$k(m) = arg \max_k \hat{p}_{mk}. \tag{2.3}$$

Bez obzira na to koju mjeru nečistoće čvora m koristimo, prirodno je definirati **kvalitetu cijepanja** tog čvora u točki s varijable X_j kao redukciju nečistoće čvora koja se tim cijepanjem postiže:

$$Q(j, s, m) = Q(m) - \frac{n_{m_L}}{n_m} \cdot Q(m_L) - \frac{n_{m_R}}{n_m} \cdot Q(m_R), \tag{2.4}$$

s tim da je n_m broj opservacija u čvoru m, a n_{m_L} i n_{m_R} označavaju broj opservacija u čvorovima m_L i m_R koji predstavljaju dva čvora djeteta nastala tim cijepanjem. Sada možemo definirati neke od mjera nečistoće čvorova:
Klasifikacijska greška (engl. *misclassification error*):

$$\frac{1}{n_m} \sum_{x_i \in R_m} \mathbb{1}_{\{y_i \neq k(m)\}}(x_i) = 1 - \hat{p}_{mk(m)} \tag{2.5}$$

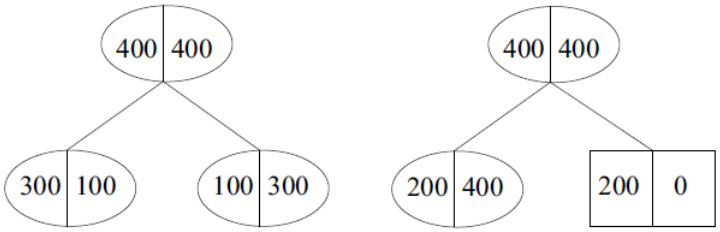
Klasifikacijska greška je jednostavno udio opservacija u čvoru m koje ne pripadaju većinskom razredu. Uobičajeno ju je koristiti prilikom podrezivanja stabla. Unatoč tome što je intuitivno prikladna mjera za opisivanje nečistoće čvora, postoje nedostaci korištenja klasifikacijske greške prilikom kreiranja stabla. Razmotrimo dvoklasni problem s 400 opservacija u svakom razredu i označimo to s $(400, 400)$. Pretpostavimo da cijepanje kreira čvorove $(300, 100)$ i $(100, 300)$, dok neko drugo cijepanje kreira čvorove $(200, 400)$ i $(200, 0)$ (slika 2.1). Kvaliteta oba cijepanja iznosi 0.25, no drugo cijepanje daje čisti čvor. Želimo takvu mjeru nečistoće čvora koja će drugom cijepanju dati prednost pred onim prvim.

U općenitom dvoklasnom problemu označimo razrede s 0 i 1, a njihove relativne frekvencije označimo redom s $1 - p$ i p. Klasifikacijska greška za taj slučaj prikazana je na slici 2.2. Kako se krećemo od maksimuma u $p = 1/2$ do minimuma u $p = 1$, funkcija bi trebala padati brže od linearne. Slično, kako se krećemo od minimuma u $p = 0$ do maksimuma u $p = 1/2$, funkcija bi trebala rasti sporije od linearne. To je ekvivalentno zahtjevu da funkcija bude strogo konkavna, stoga sada imamo modificirane zahtjeve za funkciju ϕ:

1. $\phi(0) = \phi(1) = 0$,

2. $\phi(p) = \phi(1 - \phi(p))$,

3. $\phi''(p) < 0, 0 < p < 1$.



Slika 2.1: Dvoklasni problem s 400 opservacija u svakom razredu i dvije mogućnosti cijepanja.

Dvije mjere koje zadovoljavaju navedena svojstva su Gini indeks i unakrsna entropija. Za dvoklasni slučaj su također prikazane na slici 2.2. Za razliku od klasifikacijske greške, prikladne su za korištenje kod kreiranja stabla, a može ih se koristiti i kod podrezivanja.
Gini indeks:

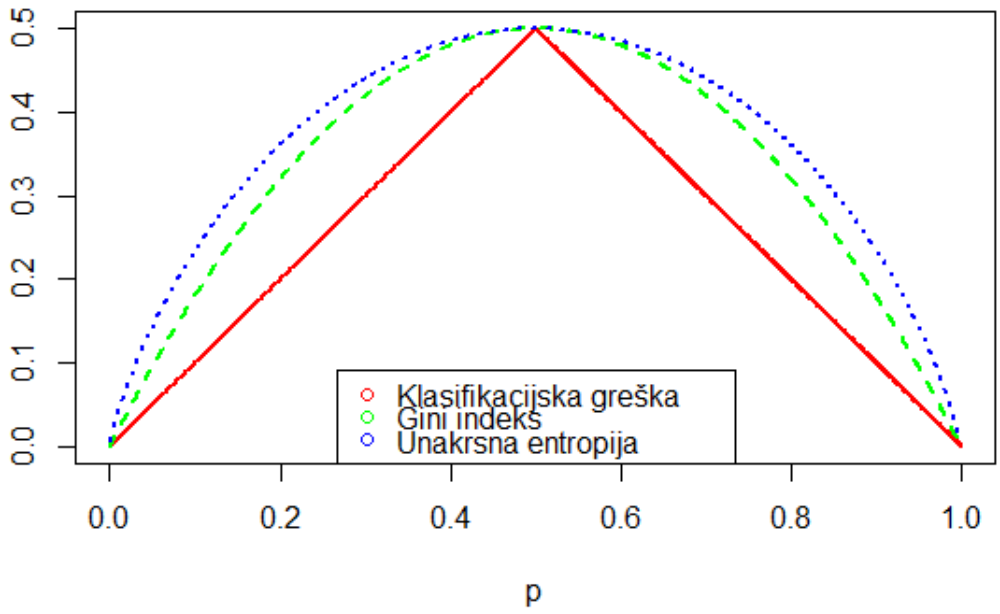
$$\sum_{k \neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2.6}$$

U dvoklasnom slučaju ovo je varijanca Bernoullijeve slučajne varijable definirana izvlačenjem (s ponavljanjem) opservacije iz čvora na slučajan način i opažanjem njenog razreda. Stoga redukciju nečistoće čvora promatramo kao redukciju varijance, a Gini indeks višeklasnog slučaja je zapravo mjera ukupne varijance preko K razreda. Nije teško vidjeti da poprima male vrijednosti ako su svi $\hat{p}_{mk}$ blizu 0 ili 1, a to znači da čvor sadrži pretežito opservacije iz istog razreda.
Unakrsna entropija (engl. cross-entropy) ili devijacija:

$$-\sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk} \tag{2.7}$$

Unakrsna entropija je prosječna količina informacija generirana izvlačenjem (s ponavljanjem) opservacije iz čvora na slučajan način i opažanjem njenog razreda pa ako je čvor čist, opažanje razreda doprinosi s 0 informacija. S obzirom na to da je $0 \leq \hat{p}_{mk} \leq 1$, slijedi $0 \leq -\hat{p}_{mk} log \hat{p}_{mk}$. Mjera unakrsne entropije će također poprimiti vrijednost blizu 0 ako su svi $\hat{p}_{mk}$ blizu 0 ili 1. Kao i Gini indeks poprima male vrijednosti ako čvor sadrži pretežito opservacije iz iste klase.

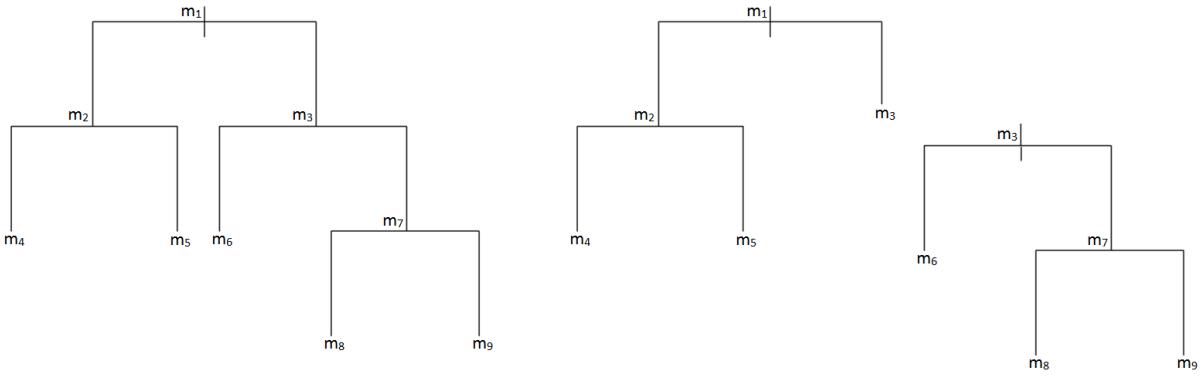
S obzirom na to da u svakom koraku odabiremo najčišći čvor na temelju podataka za treniranje, generirano stablo je pretjerano prilagođeno podacima (engl. *overfit*) i sigurno neće raditi tako dobro na testnim podacima.



Slika 2.2: Mjere nečistoće čvorova za dvoklasnu klasifikaciju kao funkcije relativne frekvencije p razreda 1. Unakrsna entropija je skalirana da prođe kroz $(0.5, 0.5)$.

2.2 Podrezivanje klasifikacijskog stabla

Koliko bi stablo trebalo biti veliko? Osim pretjerane prilagođenosti podacima, premalo stablo možda ne bi obuhvatilo važnu strukturu. Veličina stabla je parametar podešavanja i upravlja kompleksnošću modela te bi optimalna veličina stabla trebala biti prilagođeno odabrana iz podataka. Preferirana strategija je izrada velikog stabla T_0 na ranije opisani način zaustavljajući proces cijepanja tek kad je dosegnuta neka minimalna veličina čvora (recimo 5). Tada se to veliko stablo podrezuje kako bi dobili podstablo čijim odabirom imamo najmanju grešku na podacima za testiranje stabla. Ukoliko nemamo dovoljno podataka koje bi podijelili na podatke za trening i na podatke za testiranje, možemo tu grešku procjeniti korištenjem krosvalidacije koja je kasnije opisana, no s obzirom na veliki broj mogućih podstabala potrebno je na neki način smanjiti izbor podstabala koje uzimamo u obzir. To postižemo koristeći trošak posljedice složenosti podrezivanje (engl. *cost-complexity pruning*) koje još zovemo podrezivanje najslabije karike (engl. *weakest link*



(a) Stablo T_0 (b) Stablo $T_0 - T_{m_3}$ i grana T_{m_3}

Slika 2.3: Podrezivanje inicijalnog stabla T_0 u čvoru m_3

pruning).
Podrezivanje inicijalnog stabla T_0 u čvoru m znači da taj čvor postaje krajnji čvor ili list, a svi njegovi potomci su uklonjeni. T_m je grana koja se sastoji od čvora m i svih njegovih potomaka, a nastalo podstablo označavamo s $T_0 - T_m$ (slika 2.3).

Podrezano podstablo $T \subseteq T_0$ je bilo koje stablo nastalo podrezivanjem stabla T_0 u 0 ili više čvorova, tj. urušavanjem bilo kojeg broja unutarnjih (ne krajnjih) čvorova.

Neka je $\widetilde{T}$ skup svih krajnjih čvorova (listova) u T, $|\widetilde{T}|$ broj krajnjih čvorova u T i n_m broj opservacija u čvoru m, odnosno u području R_m koje taj čvor predstavlja. Definiramo klasifikacijsku grešku stabla Q_T kao udio opservacija iz skupa podataka za trening koje su pogrešno klasificirane stablom T:

$$Q_T = \sum_{m \in \widetilde{T}} \frac{n_m}{n} Q(m), \tag{2.8}$$

gdje naravno koristimo klasifikacijsku grešku kao mjeru nečistoće čvora. Sada definiramo cijenu složenosti (engl. *cost complexity*):

$$C_\alpha(T) = Q_T + \alpha|\widetilde{T}|. \tag{2.9}$$

Kada se broj krajnjih čvorova poveća za jedan (jedno dodatno cijepanje binarnog stabla), tada se $C_\alpha(T)$ povećava za α (ako udio pogrešno klasificiranih opservacija ostane isti) pa $\alpha|\widetilde{T}|$ predstavlja kaznu za kompleksnost stabla. Ovisno o vrijednosti od $\alpha(\geq 0)$ kompleksno stablo koje ne proizvodi pogrešku bi sada moglo proizvesti veću ukupnu grešku od manjeg stabla koje proizvodi mnoge pogreške. Možemo reći da podesivi parametar α upravlja kompromisom između veličine stabla i ocjene prilagodbe podacima. Ideja je za svaki α pronaći podstablo $T(\alpha) \subseteq T_0$ za koje vrijedi:

1. $C_\alpha(T(\alpha)) = min_{T \subseteq T_0} C_\alpha(T)$
2. Ako je $C_\alpha(T) = C_\alpha(T(\alpha))$, tada $T(\alpha) \subseteq T$.

Prvi uvijet kaže da ne postoji podstablo od T_0 koje ima manju vrijednost od C_α nego $T(\alpha)$, a drugi kaže da ako postoji izjednačenje, tj. ako postoji više stabala koja postižu taj minimum, odabiremo najmanje podstablo za $T(\alpha)$. Iako α može biti bilo koji nenegativan broj, postoji samo konačno mnogo podstabala od T_0. Možemo konstruirati padajući niz $T_1 \supset T_2 \supset T_3 \supset ... \supset \{m_1\}$ podstabala od T_0 takav da je T_k najmanje minimizirajuće podstablo za $\alpha \in [\alpha_k, \alpha_{k+1})$, a $\{m_1\}$ korijen stabla. Prvo stablo u nizu T_1 je najmanje podstablo od T_0 koje ima jednak udio pogrešno klasificiranih opservacija kao i T_0 (T_1=$T_{\alpha=0}$). Za pronalazak stabla T_1 odaberemo bilo koji par listova (ispod zajedničkog čvora) koji se mogu stopiti sa čvorom bez povećanja udijela pogrešno klasificiranih opservacija. Nastavljamo tako dugo dok takav par listova više ne postoji. Na taj način dobijemo stablo T_1 koje ima jednaku vrijednost od $C_\alpha(T)$ kao i T_0 za $\alpha = 0$, no s obzirom na to da je manje, ima prednost pred stablom T_0. Preostaje još pronaći ostala podstabla u nizu kao i odgovarajuće vrijednosti od α.
Neka je T_m grana stabla T s korijenom u čvoru m. Za koju vrijednost od α podstablo $T - T_m$ postaje bolje od stabla T? Kada bismo podrezivali u m, doprinos ukupnom $C_\alpha(T - T_m)$ bi bio $C_\alpha(\{m\}) = Q_{\{m\}} + \alpha$. Doprinos grane T_m ukupnom $C_\alpha(T)$ je $C_\alpha(T_m) = Q_{T_m} + \alpha|\widetilde{T}_m|$. $T - T_m$ postaje bolje stablo kada je $C_\alpha(\{m\}) = C_\alpha(T_m)$ jer za tu vrijednost od α imaju istu cijenu, no $T - T_m$ je manje stablo. Kada je $C_\alpha(\{m\}) = C_\alpha(T_m)$ tada imamo:

$$Q_{T_m} + \alpha|\widetilde{T}_m| = Q_{\{m\}} + \alpha, \tag{2.10}$$

iz čega slijedi formula za α:

$$\alpha = \frac{Q_{\{m\}} - Q_{T_m}}{|\widetilde{T}_m| - 1}. \tag{2.11}$$

Da bi iz trenutnog podstabla T_k dobili sljedeće podstablo T_{k+1}, za svaki unutarnji čvor m podstabla T_k izračunamo:

$$g_k(m) = \frac{Q_{\{m\}} - Q_{T_{k,m}}}{|\widetilde{T}_{k,m}| - 1}, \tag{2.12}$$

vrijednost od α za koju je $T_k - T_m$ bolje stablo od T_k. Tada odaberemo najslabije karike, odnosno čvorove za koje g_k postiže minimum i u njima podrežemo T_k kako bi dobili sljedeće podstablo T_{k+1}. To ponavljamo sve dok ne dođemo do korijena stabla. Iz niza podstabala generiranog podrezivanjem odaberemo ono finalno stablo koje ima najmanji udio pogrešno klasificiranih opservacija kada je primijenjeno na skup podataka za testiranje, označimo tu grešku s $R^{ts}(T)$ za stablo T. Procjenjujemo stvarnu grešku kao:

$$SE(R^{ts}) = \sqrt{\frac{R^{ts}(1 - R^{ts})}{n_{test}}}, \tag{2.13}$$

gdje je n_{test} veličina skupa podataka za testiranje.

2.3 Krosvalidacija

Kada je skup podataka premali za dijeljenje koristimo ga kao skup podataka za trening, no tada nemamo skup podataka za testiranje pa posežemo za peterostrukom ili deseterostrukom krosvalidacijom (engl. *cross-validation*). Konstruiramo stablo na prije opisani način i izračunamo $\alpha_1, \alpha_2, ..., \alpha_K$ i $T_1 \supset T_2 \supset ... \supset T_K$, gdje je T_k je najmanje minimizirajuće podstablo za $\alpha \in [\alpha_k, \alpha_{k+1})$. Sada želimo odabrati stablo iz tog niza. Procjenjujemo grešku stabla T_k iz tog niza na indirektan način.

1. korak Neka je $\beta_1 = 0$, $\beta_2 = \sqrt{\alpha_2\alpha_3}$, $\beta_3 = \sqrt{\alpha_3\alpha_4}$, ... ,$\beta_{K-1} = \sqrt{\alpha_{K-1}\alpha_K}$,$\beta_K = \infty$. Smatramo da je β_K tipična vrijednost za $[\alpha_k, \alpha_{k+1})$, stoga kao vrijednost odgovara T_k.

2. korak Podijelimo skup podataka u v grupa $G_1, G_2, ..., G_v$ (jednakih veličina) i za svaku grupu G_j:

1. Generiramo niz stabala s podrezivanjem na svim podacima osim na G_j i odredimo $T^{(j)}(\beta_1), T^{(j)}(\beta_2), ..., T^{(j)}(\beta_K)$ za taj niz.
2. Izračunamo grešku od $T^{(j)}(\beta_k)$ na G_j.

Uočimo da je $T^{(j)}(\beta_k)$ najmanje minimizirajuće podstablo niza izgrađenog na svim podacima osim na G_j, za $\alpha = \beta_k$.

3. korak Za svaki β_k sumiramo greške od $T^{(j)}(\beta_k)$ nad G_j, $(j = 1, 2, ..., v)$. Neka β_h ima najnižu ukupnu grešku. Budući da β_h odgovara T_h, odaberemo T_h iz niza stabala generiranih na svim podacima kao finalno stablo. Koristimo grešku izračunatu krosvalidacijom kao procjenu greške odabranog stabla. Važno je primijetiti da u opisanoj proceduri koristimo krosvalidaciju da bi odabrali najbolju vrijednost parametra kompleksnosti iz skupa $\beta_1, ..., \beta_K$. Jednom kad je najbolja vrijednost utvrđena, vratimo odgovarajuće stablo originalnog niza.

Poglavlje 3

Specifični zahtjevi s obzirom na podatake

3.1 Kategorijski prediktori

Za kategorijske prediktore koji imaju q mogućih neuređenih vrijednosti, imamo $2^{q-1} - 1$ mogućih odabira mjesta za cijepanje podataka pa izračuni postaju zahtjevni za velike q. No, s ishodom $0 - 1$, izračuni se pojednostavljuju. Poredamo razrede prediktora prema proporciji koji padaju u ishod klase 1. Tada cijepamo taj prediktor kao da je uređeni prediktor. Može se pokazati da to daje optimalo cijepanje, u terminima unakrsne entropije ili Gini indeksa, od svih mogućih $2^{q-1} - 1$ cijepanja. Ovaj rezultat također vrijedi za kvantitativni ishod i kvadratnu pogrešku gubitka - kategorije su poredane po rastućim sredinama ishoda. Algoritam za particioniranje ima sklonost favorizirati kategorijske prediktore s mnogo razina q; broj particija raste eksponencijalno u q i što više izbora imamo, vjerojatnije je da ćemo naći dobar izbor za podatke koje imamo. Ovo može dovesti do pretjerane prilagođenosti podacima ako je q velik i takve bi varijable trebalo izbjegavati.

3.2 Matrica gubitaka

U klasifikacijskim problemima, posljedice pogrešne klasifikacije opservacija su u nekim razredima ozbiljnije nego u drugim. Npr., vjerojatno je gore predvidjeti da osoba neće imati srčani udar kad će se to u stvarnosti dogoditi, nego obratno. Da bismo to uzeli u obzir, definiramo matricu gubitaka L, gdje je $L_{kk'}$ gubitak nastao klasificiranjem opservacija razreda k u razred k', gdje su $k, k' = 1, 2, ..., K$. Smatramo da za ispravnu klasifikaciju nije nastao gubitak, tj. $L_{kk} = 0, \forall k$. Da bismo uključili gubitke u proces modeliranja, mogli bismo modificirati Gini indeks u $\sum_{k \neq k'} L_{kk'} \hat{p}_{mk} \hat{p}_{mk'}$. To bi bio očekivani gubitak nastao randomiziranim pravilom. Ovo funkcionira za slučaj u kojem imamo više od 2 razreda, no

u dvoklasnom slučaju nema efekta jer je koeficijent $\hat{p}_{mk}\hat{p}_{mk'}$ jednak $L_{kk'} + L_{k'k}$. Za dvije klase bolji je pristup ponderaranja (dodavanje težine) opservacijama u razredu k za $L_{kk'}$. Ovo se može koristiti u slučaju više klasa samo ako, kao funkcija od k, $L_{kk'}$ ne ovisi o k'. Ponderiranje opservcija se može također koristiti i s devijacijom. Učinak ponderiranja opservacija je izmjena prethodnih vjerojatnosti razreda. U krajnjem čvoru klasificiramo na klase $k(m) = arg\min_k \sum_l L_{lk}\hat{p}_{ml}$.

3.3 Nedostatak prediktivnih vrijednosti

Pretpostavimo da za naše podatke nedostaju neke vrijednosti jednog ili svih prediktora. Mogli bismo odbaciti svaku opservaciju za koju neke vrijednosti nedostaju, no to bi moglo dovesti do ozbiljnog osiromašivanja skupa podataka za treniranje. Alternativno, mogli bismo u takvom prediktoru pokušati popuniti te vrijednosti s recimo srednjom vrijednosti preostalih opservacija. Za modele koji se baziraju na stablu postoje dva bolja pristupa. Prvi je primijenjiv na kategorijske prediktore. Jednostavno napravimo novu kategoriju za "nedostajuće vrijednosti". Iz toga bi mogli otkriti da se opservacije za koje nedostaju vrijednosti za neka mjerenja ponašaju drugačije od onih za koje to nije slučaj. Drugi, općenitiji pristup je konstrukcija surogat varijabli. Kad uzimamo u obzir prediktor za cijepanje, koristimo samo opservacije za koje taj prediktor ima vrijednosti. Odabirom najboljeg (primarnog) prediktora i točke cijepanja, formiramo listu surogat prediktora i točaka cijepanja. Prvi surogat je prediktor i odgovarajuća točka cijepanja koja najbolje oponaša cijepanje podataka za trening postignuto primarnim cijepanjem. Drugi surogat je prediktor i odgovarajuća točka cijepanja koja je druga najbolja itd. Kod slanja opservacija niz stablo ili u fazi treniranja podataka ili kroz predikciju, koristimo surogat cijepanja po redu, ako je primarni prediktor cijepanja s nedostajućim vrijednostima. Surogat cijepanja iskorištavaju korelacije između prediktora za pronalazak i ublažavanje efekta izostanka podataka. Što je veća korelacija između prediktora s nedostajućim vrijednostima i ostalih prediktora, manji je gubitak informacija zbog izostanka vrijednosti.

Poglavlje 4

Primjer u R-u

4.1 Neželjena pošta

Podatke za ovaj primjer prikupili su Mark Hopkins, Erik Reeber, George Forman i Jaap Suermondt (Hewlett-Packard Labs, Palo Alto, Kalifornija), a dostupni su za javnost na `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/spambase/`. Podaci se sastoje od relativnih frekvencija 57 najčešćih riječi i interpunkcijskih znakova iz 4601 e-mail poruke i za svaku poruku je dostupna informacija o tome da li je poruka željena ili neželjena. Podaci su prikupljeni u razdoblju od lipnja do srpnja 1999. godine. Cilj je dizajnirati automatski detektor neželjene poruke. Budući da je vlasnik doniranih podataka George Forman (tel. 650-857-7835) riječ "George" i kod područja "650" upućuju na željenu poruku. Ovo je korisno kod personaliziranja filtera neželjene poruke. U slučaju generiranja filtera za općenite svrhe trebalo bi ukloniti takve indikatore ili prikupiti puno veću kolekciju neželjenih poruka. Neželjena pošta ili spam najčešće su poruke marketinškog karaktera. Osim što nepoznati pošiljatelji nude svoje usluge i time zatrpavaju korisnički pretinac za poštu, velik broj takvih poruka je i potencijalna opasnost. One mogu biti zaražene virusom te ih je najbolje pokušati detektirati i filtrirati. Kolekcija neželjenih poruka u podacima čini 39.4% ukupnih poruka i pristigla je od administratora pošte kao i od pojedinaca. Preostalih 60.6% poruka su željene poruke pristigle iz poslovnih i privatnih izvora. U podacima nema nedostajućih prediktivnih vrijednosti.

Tablica 4.1: Deskriptivna statistika ulaznog skupa podataka

	Min	Max	Ar. sred.	St. dev.	Koef. var. (%)
word_freq_make	0	4.54	0.10455	0.30536	292
word_freq_address	0	14.28	0.21301	1.2906	606
word_freq_all	0	5.1	0.28066	0.50414	180
word_freq_3d	0	42.81	0.065425	1.3952	2130

Tablica 4.1: Deskriptivna statistika ulaznog skupa podataka

	Min	Max	Ar. sred.	St. dev.	Koef. var. (%)
word_freq_our	0	10	0.31222	0.67251	215
word_freq_over	0	5.88	0.095901	0.27382	286
word_freq_remove	0	7.27	0.11421	0.39144	343
word_freq_internet	0	11.11	0.10529	0.40107	381
word_freq_order	0	5.26	0.090067	0.27862	309
word_freq_mail	0	18.18	0.23941	0.64476	269
word_freq_receive	0	2.61	0.059824	0.20154	337
word_freq_will	0	9.67	0.5417	0.8617	159
word_freq_people	0	5.55	0.09393	0.30104	320
word_freq_report	0	10	0.058626	0.33518	572
word_freq_addresses	0	4.41	0.049205	0.25884	526
word_freq_free	0	20	0.24885	0.82579	332
word_freq_business	0	7.14	0.14259	0.44406	311
word_freq_email	0	9.09	0.18474	0.53112	287
word_freq_you	0	18.75	1.6621	1.7755	107
word_freq_credit	0	18.18	0.085577	0.50977	596
word_freq_your	0	11.11	0.80976	1.2008	148
word_freq_font	0	17.1	0.1212	1.0258	846
word_freq_000	0	5.45	0.10165	0.35029	345
word_freq_money	0	12.5	0.094269	0.44264	470
word_freq_hp	0	20.83	0.5495	1.6713	304
word_freq_hpl	0	16.66	0.26538	0.88696	334
word_freq_george	0	33.33	0.7673	3.3673	439
word_freq_650	0	9.09	0.12484	0.53858	431
word_freq_lab	0	14.28	0.098915	0.59333	600
word_freq_labs	0	5.88	0.10285	0.45668	444
word_freq_telnet	0	12.5	0.064753	0.40339	623
word_freq_857	0	4.76	0.047048	0.32856	698
word_freq_data	0	18.18	0.097229	0.55591	572
word_freq_415	0	4.76	0.047835	0.32945	689
word_freq_85	0	20	0.10541	0.53226	505
word_freq_technology	0	7.69	0.097477	0.40262	413
word_freq_1999	0	6.89	0.13695	0.42345	309
word_freq_parts	0	8.33	0.013201	0.22065	1670
word_freq_pm	0	11.11	0.078629	0.43467	553

Tablica 4.1: Deskriptivna statistika ulaznog skupa podataka

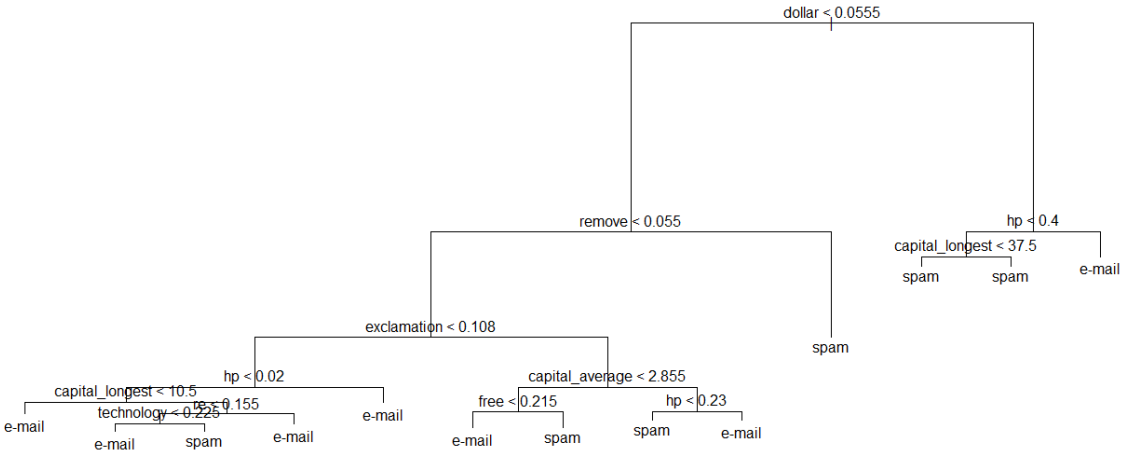
	Min	Max	Ar. sred.	St. dev.	Koef. var. (%)
word_freq_direct	0	4.76	0.064834	0.34992	540
word_freq_cs	0	7.14	0.043667	0.3612	827
word_freq_meeting	0	14.28	0.13234	0.76682	579
word_freq_original	0	3.57	0.046099	0.22381	486
word_freq_project	0	20	0.079196	0.62198	785
word_freq_re	0	21.42	0.30122	1.0117	336
word_freq_edu	0	22.05	0.17982	0.91112	507
word_freq_table	0	2.17	0.0054445	0.076274	1400
word_freq_conference	0	10	0.031869	0.28573	897
char_freq_;	0	4.385	0.038575	0.24347	631
char_freq_(	0	9.752	0.13903	0.27036	194
char_freq_[	0	4.081	0.016976	0.10939	644
char_freq_!	0	32.478	0.26907	0.81567	303
char_freq_$	0	6.003	0.075811	0.24588	324
char_freq_#	0	19.829	0.044238	0.42934	971
capital_run_length_average	1	1102.5	5.1915	31.729	611
capital_run_length_longest	1	9989	52.173	194.89	374
capital_run_length_total	1	15841	283.29	606.35	214
spam	0	1	0.39404	0.4887	124

Za svaku poruku dostupan je njezin tip zapisan u nominalnoj varijabli ”spam” gdje 1 označava neželjenu poruku, a 0 željenu poruku. Imamo 48 kontinuiranih varijabli čiji je naziv oblika ”word_freq_WORD”. Svaka bilježi postotak riječi u poruci koja odgovara riječi upisanoj umjesto ”WORD”, tj. 100·(broj pojavljivanja riječi ”WORD” u poruci)/(ukupan broj riječi u poruci). Slično, imamo 6 kontinuiranih varijabli oblika ”char_freq_CHAR” u kojima su postoci interpunkcijskih znakova u poruci navedenih umjesto ”CHAR”, tj. 100·(broj pojavljivanja znaka ”CHAR” u poruci)/(ukupan broj znakova u poruci). I na kraju imamo 3 kontinuirane varijable u kojima su mjere vezane uz neprekinute nizove uzastopnih velikih slova u poruci: ”capital_run_length_average” je prosječna duljinu takvog niza, ”capital_run_length_longest” je duljina najduljeg takvog niza i ”capital_run_length_total” je zbroj svih duljina takvih nizova koji se pojavljuju u poruci. Sve varijable s deskriptivom podataka navedene su u tablici 4.1. U tablici 4.2 je popis riječi i znakova koji pokazuju najveću prosječnu razliku između željene pošte koju označavamo kao ”e-mail” i neželjene pošte koju nazivamo ”spam”.

Primjenimo metodologiju klasifikacijskog stabla opisanu ranije s obzirom na to da imamo nadzirani problem s kategorijskim ishodom, tj. klasifikacijski problem. Odabe-

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
e-mail	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

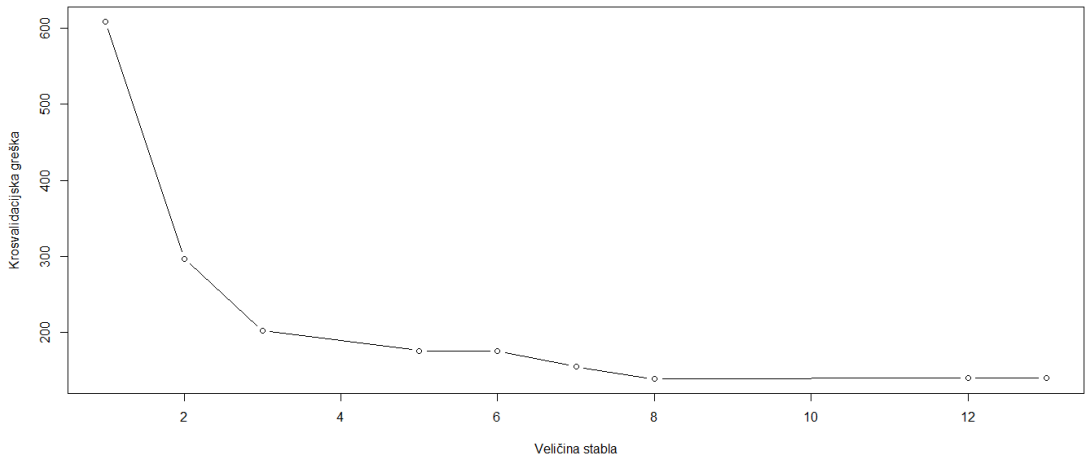
Tablica 4.2: Prosječni postotak riječi i znakova koji pokazuju najveću razliku između željene pošte (e-mail) i neželjene pošte (spam).



Slika 4.1: Klasifikacijsko stablo dobiveno korištenjem unakrsne entropije kao mjere nečistoće čvorova.

remo slučajan uzorak od 1536 podataka kao skup podataka za treniranje, a preostali podaci čine skup podataka za testiranje. Za mjeru nečistoće čvora odaberemo unakrsnu entropiju kako bismo kreirali stablo. Dobiveno stablo ima 13 krajnjih čvorova, a korištene varijable su word_freq_dollar, word_freq_remove, word_freq_exclamation, word_freq_hp, capital_run_length_longest, word_freq_re, word_freq_technology, capital_run_length_average, word_freq_free (slika 4.1).

Sada primjenjujemo deseterostruku krosvalidaciju za generiranje niza najboljih podstabala. Krosvalidacijska greška je najmanja za stablo koje ima otprilike 8 krajnjih čvorova (slika 4.2), no u nizu generiranih podstabala nema stabla s 8 krajnjih čvorova stoga finalno podrezano stablo ima 9 krajnjih čvorova i prikazano je na slici 4.3. Korištene su varijable word_freq_dollar, word_freq_remove, word_freq_exclamation, word_freq_hp, capital_run_length_longest, word_freq_re, word_freq_technology. Promotrimo korijen stabla



Slika 4.2: Krosvalidacijska greška spljoštava se na stablu s 8 krajnjih čvorova

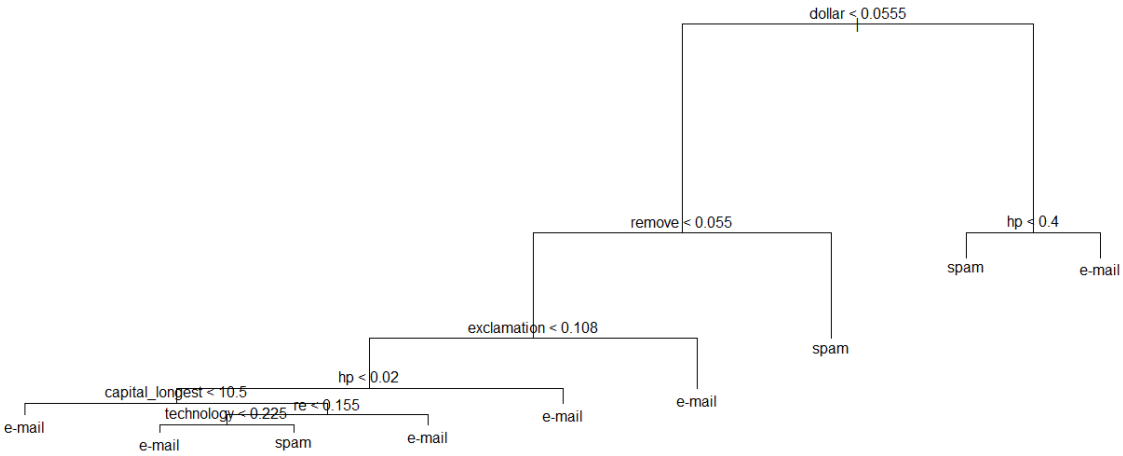
pri kojem je uvijet da je postotak znaka $ u poruci manji od 5.5%. Ako je uvijet ispunjen opservacije pripadaju lijevoj grani, ako nije ispunjen pripadaju desnoj grani. Promotrimo desnu granu i vidimo da je sljedeći uvijet da je postotak riječi hp u poruci manji od 40%. Ako je ispunjen radi se o neželjenoj poruci (spam), ako nije radi se o željenoj poruci (e-mail).

Nisu sve greške jednake. Želimo izbjeći filtriranje željene pošte dok propuštanje neželjene pošte kroz filter ima manje ozbiljne posljedice. **Osjetljivost** ili TPR definiramo kao vjerojatnost predikcije neželjene poruke kad je stvarno stanje neželjena poruka, to još zovemo stvarni pozitivan postotak (engl. *true positive rate*). **Specifičnost** ili TNR je vjerojatnost da ćemo predvidjeti da je poruka željena kad je i stvarno stanje željena poruka što još zovemo stvarni negativan postotak (engl. *true negative rate*). Neka TP označava stvarno pozitivne rezultate, TN stavarno negativne rezultate, FP lažno pozitivne rezultate i FN lažno negativne rezultate. Sada su osjetljivost i specifičnost dane sljedećim formulama:

$$TPR = \frac{TP}{TP + FN} \tag{4.1}$$

$$TNR = \frac{TN}{TN + FP} \tag{4.2}$$

Za ovaj primjer lako je izračunati osjetljivost i specifičnost pomoću podataka iz matrice konfuzije 4.3 gdje su dane vrijednosti od TP, TN, FP i FN. Dakle, u testnim podacima su 1782 željene poruke (e-mail) i klasificirane kao željene poruke, a 360 takvih poruka je krivo klasificirano kao neželjene poruke (spam). 844 neželjenih poruka je klasificirano točno kao

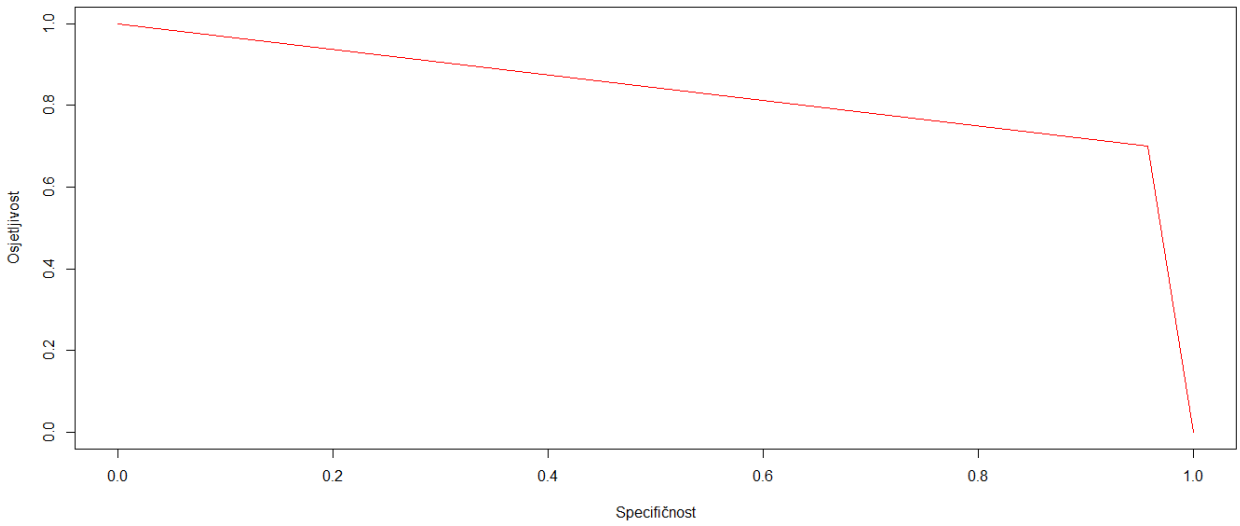


Slika 4.3: Podrezano klasifikacijsko stablo.

neželjene, a 79 pogrešno kao željene poruke. Stoga uvrštavanjem u formulu 4.1 dobijemo da je osjetljivost 0.93, a uvrštavanjem u formulu 4.2 dobijemo da je specifičnost 0.83. Za procjenu kompromisa između osjetljivosti i specifičnosti koristimo ROC krivulju (engl. *receiver operating characteristic curve*). To je graf osjetljivosti naspram specifičnosti kako variramo parametre klasifikacijskog pravila. Neka je $L_{kk'}$ gubitak vezan za predviđanje objekta razreda k kao da su u razredu k'. Variranjem relativnih veličina gubitaka L_{01} i L_{10} povećavamo osjetljivost i smanjujemo specifičnost ili obratno. U ovom primjeru želimo izbjeći označavanje željene poruke neželjenom, stoga želimo da specifičnost bude vrlo visoka. To možemo postići uzimanjem $L_{01} > 1$ s recimo $L_{10} = 1$. Variranjem gubitka L_{01} između 0.1 i 10 i primjenom na podrezano stablo (slika 4.3) proizveli smo ROC krivulju prikazanu na slici 4.4. Kako bismo postigli specifičnost blizu 100% osjetljivost mora pasti na otprilike 50%. Nejednaki gubitci uključeni su u proces kreiranja stabla. Odabrano je $L_{01} = 5$ i $L_{10} = 1$ čime se postiže veća specifičnost.

	Stvarno stanje	
Predviđanje	e-mail	spam
e-mail	TN=1782	FN=360
spam	FP=79	TP=844

Tablica 4.3: Matrica konfuzije



Slika 4.4: ROC krivulja

4.2 Prilog: kod u R-u

```
library(tree)
library(ISLR)
library(ROCR)

# Ucitavanje podataka:
spam_data = read.table("D:/Diplomski rad/spam_data.txt",
header = TRUE, row.names=NULL)
names(spam_data) = c("make", "address", "all", "3d", "our",
"over", "remove", "internet", "order", "mail", "receive",
"will", "people", "report", "addresses", "free", "business",
"email", "you", "credit", "your", "font", "000", "money",
"hp", "hpl", "george", "650", "lab", "labs", "telnet",
"857", "data", "415", "85", "technology", "1999", "parts",
"pm", "direct", "cs", "meeting", "original", "project", "re",
"edu", "table", "conference", "semicolon", "brackets_round",
"brackets", "exclamation", "dollar", "hash", "capital_average",
"capital_longest", "capital_total", "spam")
```

```
spam = spam_data[,58]
razred=ifelse(spam>0,"spam","e-mail")
spam_data=data.frame(spam_data,razred)

# Kreiranje stabla:
set.seed(2)
train=sample(1:nrow(spam_data), 1536)
spam_data.test=spam_data[-train,]
razred.test=razred[-train]
tree.spam=tree(razred~.-spam,spam_data,subset=train,
split="deviance")
summary(tree.spam)
plot(tree.spam)      # ,type="uniform"
text(tree.spam,pretty=0)
tree.pred=predict(tree.spam,spam_data.test,type="class")
table(tree.pred,razred.test)
(1736+1020)/3065      #=0.8991843

# Krosvalidacija:
set.seed(3)
cv.spam=cv.tree(tree.spam,FUN=prune.misclass,K=10)
names(cv.spam)
cv.spam
plot(cv.spam$size,cv.spam$dev,type="b",xlab='Velicina stabla',
ylab='Krosvalidacijska greska')
plot(cv.spam$k,cv.spam$dev,type="b")

# Podrezivanje stabla:
prune.spam=prune.misclass(tree.spam,best=8,
loss=matrix(c(0,1,5,0),nrow=2,ncol=2))
summary(prune.spam)
plot(prune.spam)
text(prune.spam,pretty=0)
tree.pred=predict(prune.spam,spam_data.test,type="class")
table(tree.pred,razred.test)
(1782+844)/3065      #=0.85677

class(tree.pred5)
str(tree.pred5)
```

```
class(razred.test)
str(razred.test)
razred.test=factor(razred.test)
tree.pred=c(tree.pred)
razred.test=c(razred.test)

# ROC krivulja:
predictions=tree.pred
labels=razred.test
pred=prediction(predictions,labels)
perf=performance(pred, measure = "tpr", x.measure = "tnr")
plot(perf,col=rainbow(10),xlab='Specificnost',
ylab='Osjetljivost')
```

Bibliografija

[1] A.J. Feelders, *Lecture Notes on Classification Trees*, 2016., http://www.cs.uu.nl/docs/vakken/mdm/trees.pdf.

[2] T. Hastie, R. Tibshirani i J. Friedman, *The Elements of Statistical Learning*, Springer, 2013., http://statweb.stanford.edu/~tibs/ElemStatLearn/.

[3] G. James, D. Witten, T. Hastie i R. Tibshirani, *Introduction to Statistical Learning with Applications in R*, Springer, 2012., http://www-bcf.usc.edu/~gareth/ISL/.

Sažetak

Metoda klasifikacijskih stabla temelji se na binarnom stablu. Ako je uvijet pri određenom čvoru zadovoljen, podaci pripadaju lijevoj grani, ako nije pripadaju desnoj grani. Kriteriji prema kojima se postavljaju uvijeti pri čvorovima su mjere nečistoće čvora: klasifikacijska greška, Gini indeks i unakrsna entropija. Od svih mogućih cijepanja, odabire se ono koje postiže najveću redukciju nečistoće čvora, odnosno ono koje ima najbolju kvalitetu cijepanja. Podrezivanje stabla provodi se nakon kreiranja inicijalnog stabla kako bi se izbjegla pretjerana prilagođenost podacima. Od svih mogućih podstabala, odaberemo niz podstabala koristeći cijenu složenosti, a krosvalidacijom procjenimo grešku pojedinog podstabla te odaberemo ono s najnižom ukupnom greškom.

Summary

The method of classification trees is based on binary tree. If the condition at a particular node is met, the data belong to the left branch, if condition isn't met, data belong to the right branch. The criteria for the conditions at the nodes are node impurity measures: classification error, Gini index and cross entropy. Of all the possible splittings, we choose the one that achieves the greatest reduction in node impurity, or one that has the best quality of splitting. Pruning the tree is carried out after the initial growing of the tree in order to avoid overfitting of the data. Of all the possible subtrees, we choose a run of subtrees using the cost-complexity criterion and cross-validation error estimation of each subtree and choose the one with the lowest total error.

Životopis

Anja Damiš

Osobne informacije:

[redacted]

Obrazovanje:

2012.-2016. Diplomski sveučilišni studij Matematička statistika
PMF - Matematički odsjek, Zagreb

2008.-2012. Preddiplomski sveučilišni studij matematika;
smjer: nastavnički
PMF - Matematički odsjek, Zagreb

2004.-2008. Gimnazija Josipa Slavenskog Čakovec

Tečajevi:

Uvod u SQL
ECDL AM5: Baze podataka - napredna razina (Access 2010)
SAS osnove i programski jezik
SAS grafika
Sveučilišni računarski centar Srce, Zagreb

Radno iskustvo:

Rad na algoritmu ocjenjivanja kvalitete vožnje (u R-u)
Poduzeće Ekobit, Zagreb

Poduke iz matematike
Centar za poduke Vitruvije, Čakovec

Uređivanje i vođenje štanda s voćem (sezonski)
Poduzeće Agra, Čakovec