Otkrivanje znanja u skupovima podataka

Pripremio:

Prof.dr.sc. Nikola Bogunović
Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Temeljem izvornih dokumenata (autori zadržavaju sva prava):

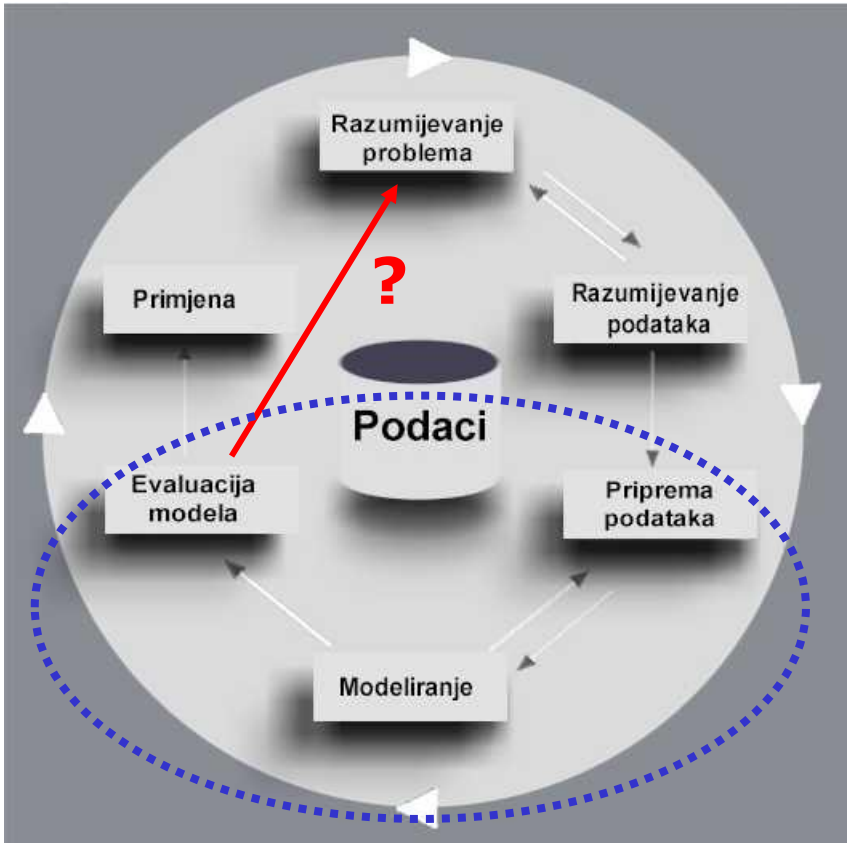
- *I.H.Witten, E.Frank, M.Hall, C.Pal,*
  DATA MINING, Practical Machine Learning Tools and Techniques
  Morgan Kaufmann, 2017.

- *T.Michell*
  MACHINE LEARNING
  McGraw Hill, 1997

Otkrivanje znanja u skupovima podataka

- Priprema podataka za dubinsku analizu
- Provođenje analize
- Mjerenje uspješnosti
- Rukovanje atributima

Otkrivanje znanja u skupovima podataka



Priprema podataka za dubinsku analizu

Otkrivanje znanja u skupovima podataka

Priprema podataka za dubinsku analizu

Za problem **klasifikacije**, u principu postoje tri skupa podataka:

- Podaci za **učenje** (poznata klasifikacija).
- Podaci za **testiranje** modela (poznata klasifikacija).
- **Novi podaci** s **nepoznatom** klasifikacijom.

Podaci za učenje i podaci za testiranje moraju biti izabrani slučajno i nezavisno, t.j. moraju biti ***strattificirani.*** To znači da u skupu za učenje i u skupu za testiranje mora biti približno podjednak broj pripadnika svakog razreda.

Temeljem uočene pogreške klasifikacije na **podacima za testiranje** nastojimo **procijeniti pogrešku klasifikacije novih** podataka.

Problem stratifikacije

Stratifikacija – kako osigurati približno podjednak broj pripadnika svakog razreda

Primjer: 1000 pripadnika razreda A
100 pripadnika razreda B

- Jednostavno rješenje:
  Slučajno odabrati 100 pripadnika iz razreda A.

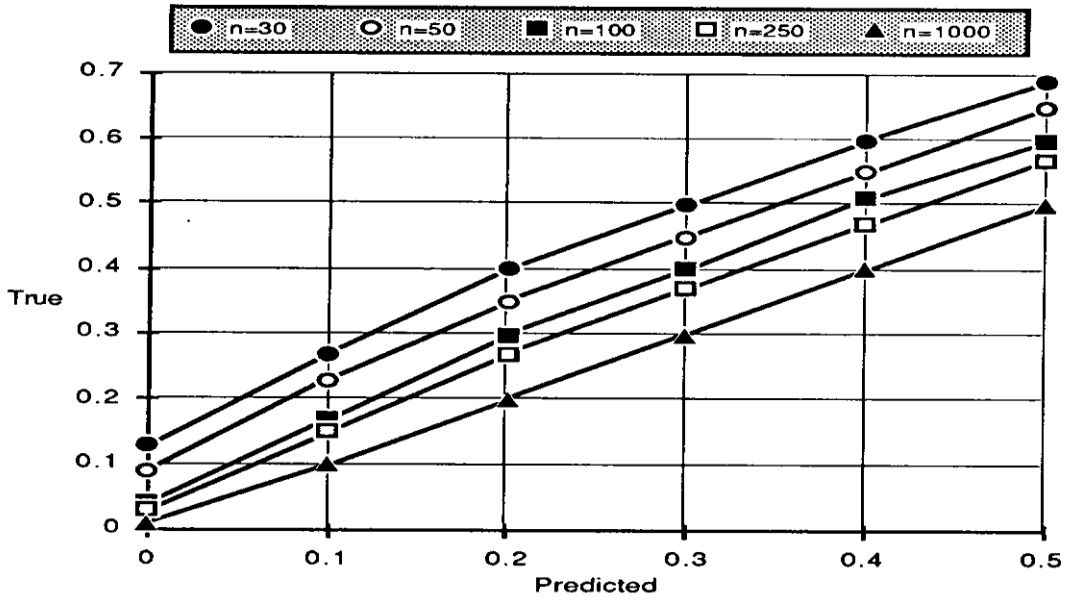
- Bolje rješenje:
  Razred A slučajno podijeliti na 10 podskupova.
  Izgraditi 10 modela (ansambli klasifikatora) s pojedinim skupovima iz A te skupom B.
  U klasifikaciji odabrati većinski izbor.

Koliko podataka

Koliko je potrebno podataka za testiranje da bi uočena pogreška bila blizu stvarnoj pogrešci novih podataka ?

Potrebno je s matematičkom izvjesnošću odrediti interval u kojem se može očekivati razlika između uočene (test) i stvarne pogreške.

Preuzeto iz:
dms.irb.hr



S 95% izvjesnosti može se tvrditi:

Za 100 primjera razlika pogreške testnih i novih je unutar 5%.

Za 1000 primjera razlika pogreške testnih i novih je unutar 1%.

Podaci za učenje: barem 3 do 10 puta više.

Otkrivanje znanja u skupovima podataka

Kako razdijeliti ukupnu dostupnu populaciju primjera na skup za učenje i skup za testiranje ?

Pretpostavka: postoji **dovoljno velika populacija** (npr. više od 1000 primjera).

Skup za učenje i skup za testiranje treba biti slučajno generiran iz populacije uzimajući u obzir stratifikaciju.

Neka je: ***n*** = ukupan broj primjera u populaciji, ***t*** = broj primjera za testiranje.

Tradicijski postupak: ***Slučajno*** se odabere skup $\{t\}$, tako da je $|t|$ oko 1/3 populacije, pa se računa pogreška.

Poboljšani tradicijski postupak: određivanje skupa $\{t\}$ i ocjena pogreške izvede se kroz ***i*** iteracija. Broj ***i*** (slučajnih izbora) je mnogo manji od ***n***. Pogreška je srednja vrijednost pogrešaka dobivenih u pojedinim iteracijama.

Otkrivanje znanja u skupovima podataka

Kako razdijeliti ukupnu dostupnu populaciju primjera na skup za učenje i skup za testiranje ?

Pretpostavka: postoji dovoljno velika populacija (npr. više od 1000 primjera).

Postupci međuvalidacije (unakrsne validacije , engl. *cross validation*)

Postupak tradicijske međuvalidacije:
Stratificirana populacija se slučajno podijeli na tri podskupa A, B, C. Postupak izračuna pogreške ponavlja se tri puta tako da se svaki puta koristi jedan različit podskup za testiranje a ostatak (dva podskupa) za učenje. Pogreška je srednja vrijednost tri izračuna.

10-struka međuvalidacija:
Stratificirana populacija se slučajno podijeli na 10 podskupova. Postupak izračuna pogreške ponavlja se 10i puta tako da se svaki puta za učenje koristi jedan različit podskup za testiranje a ostatak od devet podskupova za učenje. Pogreška je srednja vrijednost 10 izračuna.

10 puta 10-struka međuvalidacija:
Postupak 10-struke međuvalidacije se ponavlja 10 puta. Pogreška je srednja vrijednost 100 izračuna.

Otkrivanje znanja u skupovima podataka

Kako razdijeliti ukupnu dostupnu populaciju primjera na skup za učenje i skup za testiranje ?

Pretpostavka: postoji **mali broj primjera** (npr. oko 100).

Postupak izostavljanja jednog primjera: (engl. *leave one out*)

Sustav se uči s (***n-1***) primjerom, a jedan primjer se ostavlja za testiranje. To se ponavlja (***n-1)*** puta tako da se iskoriste svi primjeri za testiranje. Pogreška je broj krivo klasificiranih pojedinačnih primjera podijeljen s (***n***). Postupak je računalno skup pa je zato i primjeren za populacije s malim brojem primjera.

Otkrivanje znanja u skupovima podataka

Kako razdijeliti ukupnu dostupnu populaciju primjera na skup za učenje i skup za testiranje ?

Pretpostavka: postoji **vrlo mali broj primjera** (npr. 30 do 50).

"Bootstrap" postupak (postupak samopodizanja):

- Neka je broj primjera u populaciji ***n***. Generiraj skup za učenje tako da se slučajno ***n puta*** izvlače primjeri, ali se nakon izvlačenja vraćaju natrag u početni skup. Na taj način neki primjeri u ***n*** pokušaja biti će izvučeni nekoliko puta (jer postoji vraćanje) a neki uopće neće bit izvučeni.
- Neizvučeni primjeri služe za testiranje. Očekivani postotak tih primjera prema cijeloj populaciji je 36.8%.
- Vjerojatnost da se izvuče jedan konkretan primjer je 1/n, a da se taj ne izvuče je (1 – 1/n). Za n izvlačenja $(1 - 1/n)^n = 0.368 \times n$.
- Za dobu procjenu potrebno je postupak ponoviti oko 100 puta i izračunati srednju pogrešku.
- Postupak je računalno vrlo skup.

Izražavanje pogrešaka i uspješnosti u postupcima dubinske analize podataka

Pogreške u postupcima dubinske analize podataka

- **Pogreška** (engl. *error*) je krivo klasificiranje primjera generaliziranim (induciranim) modelom (konceptom).
- **Učestalost pogreške** (engl. error rate) = broj_pogrešaka/broj primjera

- To je vrlo grubi način izražavanja pogrešaka jer npr.:

  Pogreška u dijagnosticiranju neke osobe kao zdrave iako je teško bolesna smatra se mnogo ozbiljnijom pogreškom nego dijagnosticiranju nekoga kao bolesnog iako je zdrav.

- Uvodi se razlikovanje pogrešaka po **konfuzijskoj matrici** (engl. *confusion matrix*)

Otkrivanje znanja u skupovima podataka

Konfuzijska matrica u izražavanju pogrešaka u postupcima dubinske analize podataka

Primjer konfuzijske matrice za klasifikaciju u tri razreda:

Stvarni razredi

Razredi po klasifikatoru	1	2	3
1	30	1	0
2	1	43	5
3	0	2	75

Ovaj primjer stvarno pripada razredu 1, a klasifikator ga je svrstao u razred 2.

- Broj ispravno klasificiranih primjera – duž dijagonale
- Pogrešne klasifikacije - ostalo

Otkrivanje znanja u skupovima podataka

Konfuzijska matrica za klasifikaciju dva razreda

- To je najčešći slučaj; klasifikacija u više razreda može se svesti na seriju klasifikacije u dva razreda.
- Odgovara predikciji pojavljivanja ili ne događaja (hipoteze), tzv. razred **pozitivnih** i razred **negativnih** primjera.
- Postoje dva moguća tipa pogreške:
  - Krivo klasificiranje primjera u pozitivne iako to nisu - krivi pozitivni primjeri (**FP** – engl. *false positives*).
  - Krivo klasificiranje primjera u negativne iako to nisu - - krivi negativni primjeri (**FN** – engl. *false neagtives*).

Stvarni razredi

	Razred pozitivnih (**C+**)	Razred negativnih (**C-**)
Predikacija pozitivnih (**R+**)	Pravi pozitivni (**TP**)	Krivi pozitivni (**FP**)
Predikcija negativnih (**R-**)	Krivi negativni (**FN**)	Pravi negativni (**TN**)

Otkrivanje znanja u skupovima podataka

Definicije indikatora pogreške za klasifikaciju u dva razreda

Domena medicine:

Osjetljivost – engl. sensitivity = broj_pozitivnih / broj_stvarnih_P

Specifičnost – engl. specificity = broj_negativnih / broj_stvarnih_N

svi stvarni P

$$Sensitivity = \frac{TP}{TP + FN} = \text{True Positive Rate}$$

svi stvarni N

$$Specificity = \frac{TN}{TN + FP} = \text{True Negative Rate}$$

T = ispravno

F = krivo

klasificirani

- Visoka **osjetljivost** (sensitivity) u dijagnostici bolesti:
  - ispravno klasificiranje pacijenata koji **imaju bolest**. **(pozitivni)**
- Visoka **specifičnost** (specificity) u dijagnostici bolesti:
  - ispravno klasificiranje pacijenata koji **nemaju bolest (negativni)**.
- Teško je postići oboje !

Definicije indikatora pogreške za klasifikaciju u dva razreda

Domena informacijski sustavi za dohvat podataka:

Fokus na **pozitivnom primjerima**.

Odziv (Opoziv) - engl. recall – kao i osjetljivost u medicini

Preciznost – engl. precision = broj_pozitivnih / broj_svrstanih_u_P

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} = Sensitivity = \mathrm{True\ Positive\ Rate}$$

kao u domeni medicine

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP}$$

predikcija pozitivnih

Neke dodatne izvedene mjere:

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

$$G = \sqrt{\mathrm{precision} \cdot \mathrm{recall}}$$

Otkrivanje znanja u skupovima podataka

Definicije indikatora pogreške za klasifikaciju u dva razreda

Učinkovitost sustava obično se mjeri **frekvencijskim omjerima**:

Osjetljivost (sensitivity)	TP / C+
Specifičnost (specificity)	TN / C-
Prediktivna vrijednost pozitivnih (+)	TP / R+
Prediktivna vrijednost negativnih (-)	TN / R-
Točnost (accuracy), **Uspješnost** (success)	(TP + TN) / ((C+) + (C-))
Pogreška (error rate) = 1 - točnost	1– [(TP + TN) / ((C+) + (C-))]

C – stvarna pripadnost razredu (C+ = TP + FN, C- = FP + TN)

R – predikcija po klasifikatoru (R+ = TP + FP. R- = FN + TN)

- Nedostatak ovih mjera: različiti indikatori s različitim pridruženim značenjima. Cilj: **vizualizirati više indikatora odjednom**.

Kappa statistika

Primjer **izvorne** konfuzijske matrice za tri razreda:

		Predicted class			
		a	b	c	Total
Actual class	a	88	10	2	100
	b	14	40	6	60
	c	18	10	12	40
	Total	120	60	20	

- Testni skup ima 200 primjera. Od toga 88+40+12=140 je ispravno klasificirano (uspješnost je 140/200=0.7 , ili 70%).
- Promatrani klasifikator je svrstao (predvidio) 120 u razred "a", 60 u razred "b" i 20 u razred "c".
- Kako bi izgledala konfuzijska matrica za **slučajan** klasifikator koji bi svrstao isti broj primjera (100-60-40l) u pojedine razrde (a, b, c) u omjeru 120-60-20 kao i promatrani klasifikator ?

Otkrivanje znanja u skupovima podataka

Kappa statistika

Primjer konfuzijske matrice za tri razreda, slučajan klasifikator u istim omjerima:

		Predicted class a	b	c	Total
Actual class	a	60	30	10	100
	b	36	18	6	60
	c	24	12	4	40
	Total	120	60	20	

Slučajni klasifikator: 60-30-10 zadržava ukupno 100 ali je u omjeru 120-60-20 (podijelimo s 2)

- U takvom slučajnom klasifikatoru (koji je uzeo u obzir i omjere u izvornom klasificiranju) ispravno je klasificirano 60+18+4=**82** primjera.
- Kappa **oduzima** tih 82 od uspješnosti izvornog klasifikatora (140-82=58) i stavlja u omjer prema oduzimanju tih 82 od idealnog klasifikatora (200).
- Kappa = (140 – 82) / (200 – 82) = 58/118, ili 49.2 %.
- Kappa statistika (maks=100%) izražava mjeru uspješnosti promatranog klasifikatora (140) prema idealnom (200) uz korekciju slučajnog izbora (82).
- Veći Kappa = bolji promatrani klasifikator prema slučajnom odabiru.

Otkrivanje znanja u skupovima podataka

Izražavanje pogrešaka ROC krivuljom

(engl. *Receiver operating characteristic*)

- Cilj svakog klasifikatora je odrediti što veći postotak pravih pozitivnih primjera i što manji postotak krivih pozitivnih primjera.
- Grafički prikazujemo:
  - na ordinati **učestalost pravih pozitivnih** (jednako kao osjetljivost ili odziv)

    = broj_pravih_pozitivnih / ukupan_broj_pozitivnih

$$tp = \frac{TP}{TP + FN} \times 100\%$$

  - na apscisi **učestalost krivih pozitivnih** (to nije specifičnost)

    = broj_krivih_pozitivnih / ukupan_broj_negativnih

$$fp = \frac{FP}{FP + TN} \times 100\%$$

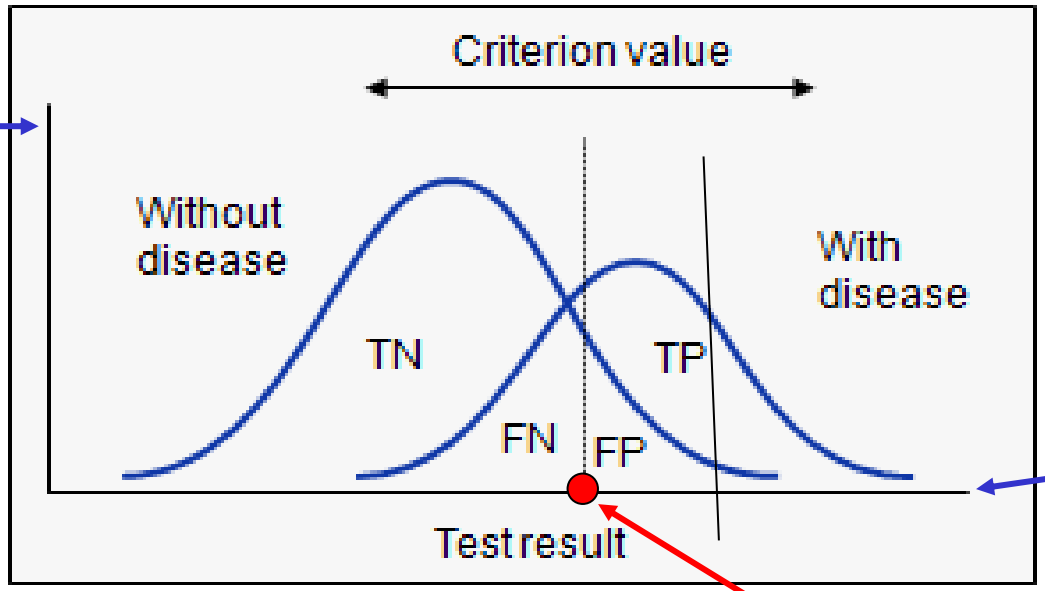
Otkrivanje znanja u skupovima podataka

Izražavanje pogrešaka ROC krivuljom

Pretpostavimo mjerenje neke značajke (npr. očni tlak) u zdravih i bolesnih osoba (kontinuirane vrijednosti, normalna razdioba):

Populacija

(broj osoba na kojima je mjereno, normalna razdioba)



Kontinuirane vrijednosti značajke (npr. očni tlak)

Za neki prag:

Površina TN – sigurno nemaju bolest

Površina TP – sigurno imaju bolest

Površina FN – krivo svrstani da nemaju bolest

Površina FP – krivo svrstani da imaju bolest

Kriterijska vrijednost (diskriminacijski prag)

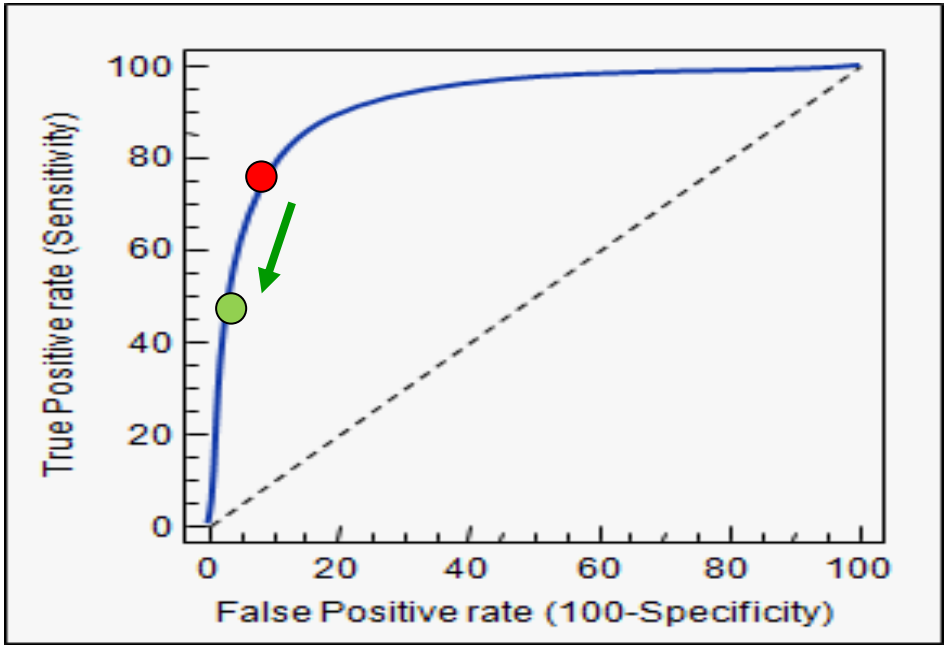
"Svi iznad praga su bolesni a ispod nisu.„

Računamo učestalosti !

Izražavanje pogrešaka ROC krivuljom

- Svaki diskriminacijski prag daje neku vrijednost za ***tp*** i ***fp***.
- Neka je to za naš primjer diskriminacijskog praga točka ●.
- Mjerenje na većem broju pragova daje **tp-fp** točke = **ROC krivulja**.
- Ako povećavamo diskriminacijski prag – slijedi manja učestalost krivih pozitivnih (smjer ←) i naravno manja učestalost pravih.

tp



fp

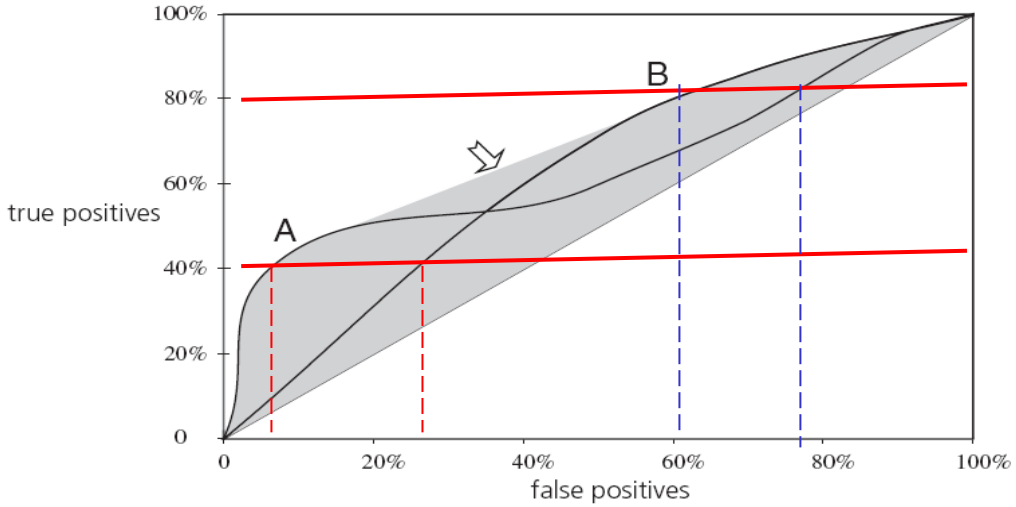
Izražavanje pogrešaka ROC krivuljom

- Dobar klasifikator nastoji s povećanjem diskriminacijskog praga znatno više smanjiti učestalost krivih pozitivnih nego što smanjuje učestalost pravih pozitivnih (razdiobe populacija pozitivnih i negativnih su više razdvojene).
- Što je klasifikator bolji (koncept je bolje naučen-generaliziran) to se njegova ROC krivulja približava gornjem lijevom uglu. Idealno postoje samo površine **TP** (pravi pozitivni) i **TN** (pravi negativni).
- Klasifikator s ROC krivuljom po dijagonali je slučajan izbor (nema kvalitete u određivanju pravih prema krivih pozitivnih).
- ROC krivulja je dobila ime iz područja detekcije signala gdje nastojimo razdvojiti prijam dva signala.
- Ocjenjivanje dva klasifikatora prema obliku ROC krivulje može slijediti i druge kriterije – vidi slijedeću sliku.

Otkrivanje znanja u skupovima podataka

Izražavanje pogrešaka ROC krivuljom

Neka ROC krivulje za klasifikatore A i B imaju oblik kao na slici:



- **Klasifikator A** je bolji ako analiziramo mali uzorak pozitivnih primjera (npr. 40% pravih pozitivnih) i daje samo oko 5% krivih pozitivnih (klasifikator B bi dao više od 25% krivih pozitivnih)
- **Klasifikator B** je bolji ako analiziramo veći uzorak pozitivnih primjera (npr. 80% pravih pozitivnih) i daje 60% krivih pozitivnih (klasifikator A bi dao tek nešto manje od 80% krivih pozitivnih).

Izražavanje pogrešaka ROC krivuljom

- U prethodnom primjeru za analizu pozitivnih uzoraka između 40% i 80% treba slučajno kombinirati **klasifikatore A i B** s odgovarajućim vjerojatnostima kako bi se ostvarile vanjske točke na osjenčanom području (konveksnoj plohi).

Neka:

A daje: ***tA*** (true rate), ***fA*** (false rate) pozitivnih

B daje: ***tB*** (true rate), ***fB*** (false rate) pozitivnih

Ako se modeli A i B koriste **slučajnim izborom** s vjerojatnošću $p + q = 1$ slijedi učestalost pravih i krivih pozitivnih primjera:

$p \cdot tA + q \cdot tB$ (**true positive rate**)

$p \cdot fA + q \cdot fB$ (**false positive rate**)

To predstavlja točku na pravcu koji spaja: (tA, fA) i (tB, fB)

Varijacija *p* i *q* pomiče točku duž pravca.

Otkrivanje znanja u skupovima podataka

Izražavanje pogrešaka krivuljom izdizanja

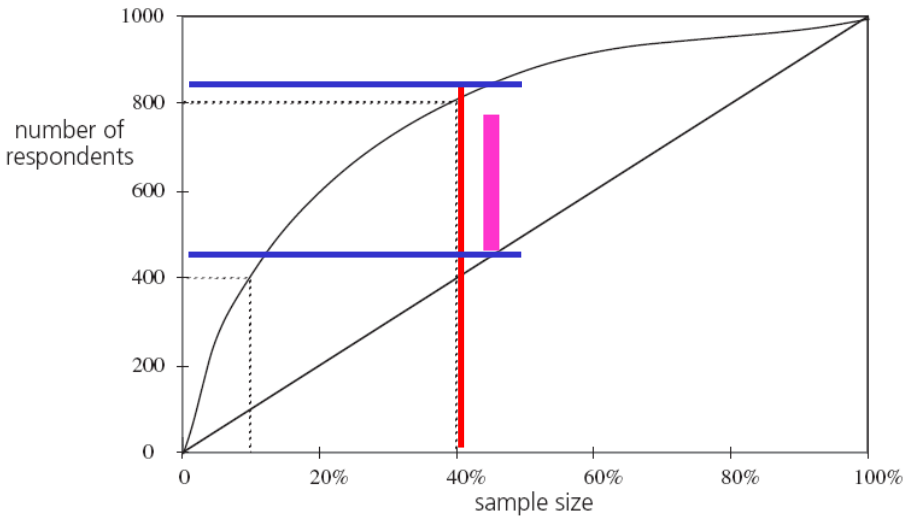
(engl. *lift chart*)

- Vrlo česta primjena u području marketinga (CRM).
- Neka za klasifikaciju u dva razreda odabrani postupak učenja generira skup svih **testnih** primjera uređenih po predikciji vjerojatnosti pripadanja pozitivnom razredu (dakle ne po stvarnoj pripadnosti DA/NE).
- Za taj testni skup poznato je koji su **pravi** pozitivni primjeri.
- Odnos predikcijom po vjerojatnosti uređenih primjera i pravih pozitivnih primjera je **krivulja izdizanja** (koliko bolje od slučajnog izbora).
- Tražimo **klasifikator** koji bi **za neki podskup po vjerojatnosti uređenih primjera dao što više pravih pozitivnih primjera.**
- Npr. postoji predikacija po vjerojatnosti uređenog testnog skupa osoba koje bi mogle odgovoriti na našu promotivnu akciju. Poznati su i pravi pozitivni (odgovorili su na akciju) u tom testnom skupu.
- Iz krivulje izdizanja za izgrađeni klasifikator i skup novih osoba uređenih po vjerojatnosti pripadanja pozitivnom razredu **očekujemo** koliko bi novih osoba u skupu pripadalo pozitivnom razredu – npr odgovorilo na promotivnu akciju.

Otkrivanje znanja u skupovima podataka

Izražavanje pogrešaka krivuljom izdizanja

Primjer krivulje izdizanja
za **skup testnih primjera**
(**poznata klasifikacija**)
i odabrani klasifikator



- Na x osi su svi **testni** primjeri uređeni po padajućoj vjerojatnosti pripadanja pozitivnom razredu **kako je dao odabrani klasifikator**.
- Na y osi je broj poznatih (**testnih**) **pozitivnih** (od 0 do npr. 1000).
- Bez klasifikatora možemo tvrditi da ako slučajno uzmemo 40% svih primjera da će među njima vjerojatno biti 40% pravih pozitivnih (t.j 400). To je izbor po dijagonali na slici (slučajan izbor).
- S klasifikatorom ako uzmemo 40% prvih **uređenih po vjerojatnosti**, među njima će biti 80% pozitivnih (t.j. 800), što je **izdizanje** 2 puta.
- **Očekujemo da klasifikator vjerojatnosnim uređivanjem nepoznatih primjera generira jednako uzdizanje.**

Otkrivanje znanja u skupovima podataka

Izražavanje pogrešaka krivuljom Odziv-Preciznost

Primjer iz područje rukovanja dokumentima:

Treba odabrati jedan od dva sustava A i B.

A: na upit dohvaća 100 dokumenata u kojima je 40 relevantno.

B: na upit dohvaća 400 dokumenata od kojih je 80 relevantno.

Odgovor ovisi o relativnoj **cijeni krivih pozitivnih** (dokumenti koji su dohvaćeni ali nisu relevantni) i o relativnoj cijeni **krivih negativnih** (dokumenti koje sustav nije dohvatio a relevantni su).

Primjena ranije definiranih mjera:

$$\textbf{Odziv} = \frac{\text{broj\_dohvaćenih\_i\_relevantnih\_dokumenata}}{\text{svi\_relevantni\_u\_skupu (ne samo dohvaćeni)}}$$

$$\textbf{Preciznost} = \frac{\text{broj\_dohvaćenih\_i\_relevantnih\_dokumenata}}{\text{svi\_dohvaćeni}} = \frac{40}{100} \text{ (za A)}$$

- Odnos Odziv-Preciznost za razne brojeve dokumenata čini krivulju koja govori o kakvoći sustava za dohvaćanje.

Otkrivanje znanja u skupovima podataka

Modificirani indikatori pogreške

- Umjesto minimizacije pogreške (ili omjera temeljenih na pogrešci) minimizira se **cijena koštanja** pogreške.
- Svakom tipu pogreške pridružena je **težina** pogrešne klasifikacije (kazna).
- Pojedinačna cijena je umnožak pogreške i njenog težinskog faktora.
- Iz pojedinačnih slijedi **srednja i sumarna cijena**.
- Za konfuzijsku matricu (za n razreda) postoji n^2 vrijednosti.
- Ako je E_{ij} **broj pogrešaka** za pojedini tip i C_{ij} **pridružena cijena**, totalna cijena za krivu klasifikaciju je:

$$Cost = \sum_{i=1} \sum_{j=1} E_{ij} C_{ij}$$

- U analizi **rizika** i analizi **odluka** treba koristiti **cijenu** (kaznu) i **dobit**.
- **Racionalan cilj klasifikatora je u maksimiziranju dobiti, t.j.: maksimiziranje razlike u dobiti zbog ispravne klasifikacije i gubitaka zbog pogrešne klasifikacije.**

Otkrivanje znanja u skupovima podataka

Pogreške u postupcima dubinske analize koji daju vjerojatnosti pripadnosti razredu

- Postupci dubinske analize podataka tipično klasificiraju primjere u jedan od unaprijed definiranih razreda.
- Neki postupci dubinske analize podataka (npr Bayes-ov) daju **vjerojatnosti** propadanja pojedinom razredu.
  - Neka za ***k*** razreda postupak dubinske analize podataka generira **za svaki primjer (jedan)** vektor vjerojatnosti:

    $(p_1 \ldots p_k)$, gdje je p_j vjerojatnost pripadanja toga primjera ***j***–tom razredu. Pri tome je za svaki primjer:

    $\Sigma_{j=1..k}\, p_j = 1$
  - Analogno se može definirati vektor stvarnog pripadanja razredu **za svaki primjer**:

    $(a_1 \ldots a_k)$, gdje je $a_i=1$ za pripadnost ***i***-tom razredu, a svi ostali a=0.

Otkrivanje znanja u skupovima podataka

Pogreške u postupcima dubinske analize koji daju vjerojatnosti pripadnosti razredu

Pogreška temeljena na **funkciji kvadrata gubitka**

(engl. *quadratic loss function*)

- Za **pojedini primjer** definiramo gubitak: $\Sigma_{j=1..k} (p_j - a_j)^2$

  Kako je samo jedan a=1 (ostali su 0), to u gornjoj sumi je doprinos krive klasifikacije $(p_j - 0)^2$, odnosno $(p_i - 1)^2$ je doprinos ispravne klasifikacije.
- Ako je ***i*** ispravan razred, **gubitak za pojedini primjer** iznosi:

  $1 - 2p_i + \Sigma_{j=1..k} (p_j)^2$ (gdje je ***k*** broj razreda)

  Formula slijedi uz $(p_i - 1)^2 = 1 - 2p_i + (p_i)^2$ gdje je ***i*** ispravan razred, a član $(p_i)^2$ se uključuje u opću sumu.
- Naveden izraz se sumira po svim primjerima u testnom skupu.

Otkrivanje znanja u skupovima podataka

Pogreške u postupci dubinske analize koji daju vjerojatnosti pripadnosti razredu

Pogreška temeljena na **funkciji gubitka informacije**

(engl. *information loss*)

- Funkciju gubitka informacije definiramo: $-\log_2 p_i$

  gdje je ***i* stvaran razred** (ne uzima u obzir vjerojatnosti ostalih razreda). Jako penalizira male vjerojatnosti.
- Funkcija se izražava u bitovima i daje najmanji broj bitova s kojima se može izraziti (kodirati) informacija o ispravnoj klasifikaciji primjera u odnosu na razdiobu ($p_1 \ldots p_k$).
- Npr. bacanje novčića, dva razreda s jednakom vjerojatnosti, pojava "pisma" treba 1 bit jer:

  $-\log_2 (1/2) = 1$ (1/2 je vjerojatnost stvarnog razreda)
- Problem ako neka vjerojatnost = 0 (vidi Bayesov klasifikator).

Otkrivanje znanja u skupovima podataka

Pogreške u predviđanju numeričkih vrijednosti

- Mnogi postupci strojnog učenja **predviđaju numeričke vrijednosti** skupa primjera (a **ne pripadnost razredu**).
- Ranije definicije pogrešaka (pojedina pogreška postoji kao kriva klasifikacija) nisu u ovim slučajevima adekvatne. Pogreška ne samo da postoji ili ne nego **ima i svoju numeričku mjeru**.
- Neka za testni skup od ***n*** primjera (poznata je točna vrijednost svakog primjera) označimo:

  $(p_1, \ldots, p_n)$ - predviđene (strojno naučene) vrijednosti u skupu ***n*** primjera (to nisu vjerojatnosti)

  $(a_1, \ldots, a_n)$ – stvarne vrijednosti u skupu ***n*** primjera
- Za pojedinu metodu (postupak) strojnog učenja definiraju se razne mjere za pogrešku za ***n*** primjera.

Otkrivanje znanja u skupovima podataka

Pogreške u predviđanju numeričkih vrijednosti

mean-squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
root mean-squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
mean absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
relative squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$, where $\bar{a}=\frac{1}{n}\sum_i a_i$
root relative squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
relative absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
correlation coefficient	$\frac{S_{PA}}{\sqrt{S_P S_A}}$, where $S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_p=\frac{\sum_i (p_i-\bar{p})^2}{n-1}$, and $S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}$

Otkrivanje znanja u skupovima podataka

Pogreške u predviđanju numeričkih vrijednosti

Objašnjenje oznaka s prethodne slike (***n*** primjera):

- **Mean squared error** - Srednja kvadratna pogreška.
- **Root mean squared error** – Korijen iz srednje kvadratne pogreške.
- **Mean absolute error** – Srednja apsolutna pogreška (ne uzima u obzir predznak, ne izdiže izuzetke (*outliers*) kao srednja kvadratna.
- **Relative squared error** – Relativna kvadratna pogreška. Relativno (normalizirano) prema jednostavnom prediktoru (t.j. srednjoj vrijednosti).
- Srednja vrijednost: $\bar{a} = \frac{1}{n}\sum_i a_i$
- **Root relative squared error** - Korijen iz relativne kvadratne pogreške.
- **Relative absolute error** – Relativna apsolutna pogreška. Također normalizirano prema jednostavnom prediktoru – srednjoj vrijednosti.

Pogreške u predviđanju numeričkih vrijednosti

Objašnjenje oznaka s prethodne slike:

Correlation coefficient – Korelacijski koeficijent (ρ_{ap})

- Mjeri statističku korelaciju između ***a***-ova i ***p***-ova.
- $\rho_{ap} = 1$ za potpuno korelirane, $\rho_{ap} = 0$ za nekorelirane, $\rho_{ap} = -1$ za negativno (suprotno) korelirane rezultate.
- ρ_{ap} se razlikuje od drugih mjera koje mijenjaju pogrešku ako se generirane (naučene) vrijednosti p_i pomnože s istim faktorom a vrijednosti a_i se ostave neizmijenjene . ρ_{ap} **je neosjetljiv na skaliranje.**
- Veći ρ_{ap} označuje bolji postupak strojnog učenja (kod ostalih mjera manje vrijednosti označuju bolji postupak strojnog učenja).

Otkrivanje znanja u skupovima podataka

Ispravna usporedba više postupaka dubinske analize podataka

- Pri predlaganju novog postupka strojnog učenja potrebno je usporediti njegovu učinkovitost s uobičajenim postupcima na istom problemu (skupu podataka).
- Za veliki skup podataka (vid ranije graf mogućeg odstupanja pogreške) provodi se **međuvalidacija** s približno jednako velikim skupovima slučajno odabranim iz domene kao i u postupku s kojim se komparira, te računa srednja pogreška.
- Međutim, traži se usporedba postupaka **za cijelu domenu**, t.j. za **sve moguće podskupove za učenje i testiranje**.
- Međuvalidacijom se *slučajno odabiru primjeri* pa je potrebno utvrditi da li se dobivena srednja pogreška statistički signifikantno razlikuje za npr. dva postupka koja uspoređujemo.
- Za utvrđivanje signifikantne razlike koristi se ***t-test*** (***Student t-test***), odnosno ***Udvojeni*** (engl *paired*) ***t-test***) jer se radi o usporedbi dva postupka s istim mađuvalidacijskim eksperimentom.

Rukovanje skupom atributa (značajki)

- Odabir relevantnih atributa
- Kreiranje novih atributa
- Smanjenje dimenzionalnosti

Atributi (značajke) i njihove vrijednosti

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	32	85	Ne	Ne
Sunčano	27	90	Da	Ne
Oblačno	30	86	Ne	Da
Kišovito	17	96	Ne	Da
Kišovito	15	80	Ne	Da
Kišovito	12	70	Da	Ne
Oblačno	11	65	Da	Da
Sunčano	19	95	Ne	Ne
Sunčano	16	70	Ne	Da
Kišovito	22	80	Ne	Da
Sunčano	22	70	Da	Da
Oblačno	19	90	Da	Da
Oblačno	28	75	Ne	Da
Kišovito	18	91	Da	Ne

Otkrivanje znanja u skupovima podataka

1. **Proces odabira relevantnih atributa (iz skupa izvornih)**

   (engl. feature selection, attribute selection, variable selection)

2. **Proces kreiranja novih atributa iz ulaznih podataka**

   (engl. feature extraction)

Otkrivanje znanja u skupovima podataka

Uvodno:

- Svi atributi nisu jednako relevantni za uspješnost klasifikacije ili predikcije
- Atribut s vrlo malo vrijednosti u svojoj koloni je irelevantan
- Atribut čija se numerička vrijednost vrlo malo mijenja je irelevantan
- Atribut koji predstavlja dupliranje drugog atributa je irelevantan
- Atributi koji su korelirani međusobno a nisu korelirani s ciljnim atributom su irelevantni
- **Irelevantni atributi usporavaju postupke analize i negativno utječu na rezultat**

Tražimo relevantne atribute !

Otkrivanje znanja u skupovima podataka

Proces odabira relevantnih atributa (engl. feature selection, attribute selection, variable selection)

- Reducira broj potrebnih atributa u skupu podataka.
- Otkriva i izbacuje nepotrebne, irelevantne i redundantne atribute koji ne doprinose točnosti predikcije.
- Proces odabira **rezultira u podskupu izvornih atributa**
- **Ciljevi** procesa odabira samo relevantnih atributa:
  - povećati točnost,
  - ubrzati rad,
  - povećati bolje razumijevanje procesa koji je generirao podatke
- Odabir relevantnih atributa može se promatrati kao **kombinacija pretraživanja prostora atributa i ocjene uspješnosti postupka dubinske analize svakog podskupa atributa** (npr. učestalost pogreške)

Otkrivanje znanja u skupovima podataka

Prema pristupu **evaluacijskoj metrici** u procesu odabira podskupa relevantnih atributa razlikujemo:

- Postupci omotača (engl. *wrapper*)
- Filtarski postupci (engl. *filters*)
- Ugrađeni postupci (engl. *embedded*)

Odabir atributa postupcima omotača

- Koriste odabrani klasifikacijski/prediktivni model za evaluaciju podskupova atributa.
- **Svaki podskup atributa sudjeluje u učenju modela te se** mjeri uspješnost **na skupu primjera za testiranje.**
- Budući da svaki podskup sudjeluje u učenju, postupak je računalno dugotrajan ali daje najbolji podskup za odabrani prediktivni model
- Postoji opasnost od prevelikog slaganja s podacima za učenje (engl. *overfitting*)

Otkrivanje znanja u skupovima podataka

Filtarski postupci odabira atributa

- Slični su postupcima omotača ali **za evaluaciju ne koriste odabrani prediktivni model već neku drugu mjeru** koja ubrzava postupak
- Uobičajene mjere uključuju:
  - Informacijska dobit (entropija) ili GINI „nečistoća”
  - Korelacijski koeficijent
  - Udaljenost razreda u klasifikaciji
  - Test signifikantnosti za svaku kombinaciju razred/atributi
  - …
- Filtarski postupci nisu prilagođeni pojedinom prediktivnom modelu pa se rabe u **fazi pred-procesiranja**
- Podskup atributa izdvojen filtarskom metodom je općenitiji od skupa izdvojenog postupkom omotača
- Neki filtri daju prioritetni slijed atributa umjesto najboljeg podskupa a odabir broja atributa s vrha liste izvodi se mađuvalidacijom

Otkrivanje znanja u skupovima podataka

Ugrađeni postupci odabira atributa

- Ova grupa postupaka **uključena je u tehnike** koje se koriste tijekom procesa izgradnje klasifikacijskog/prediktivnog modela
- Posebno se koriste u regresijskoj analizi (LASSO, RIDGE, …) gdje se regresijski koeficijenti penaliziraju po nekoj metrici, zatim u klasifikaciji slučajnim šumama (engl. *random forest*), i sl.
- Ugrađen postupak vrijedi samo za odabrani algoritam (npr. za Suppot Vector Machine – SVM). Nije uporabiv za stabla odlučivanja.
- Postupak vraća skup atributa + klasifikator/prediktor
- Nije podložan prevelikom slaganju s podacima za učenje (engl. *overfitting*)
- Po brzini je između filtarskih i postupaka omotača

Otkrivanje znanja u skupovima podataka

Svaki postupak odabira relevantnog podskupa atributa sastoji se iz dva stupnja:

1. Pretraživanje prostora atributa i formiranje podskupa
2. Ocjena kvalitete podskupa atributa

Pretraživanje prostora atributa i formiranje podskupa

- Iscrpno pretraživanje *(engl. Exhaustive)*
- Prvi najbolji *(engl. Best first)*
- Simulirano napuštanje *(enl. Simulated annealing)*
- Genetski algoritam
- Pohlepno pretraživanje prema naprijed *(Greedy forward selection)*
- Pohlepno pretraživanje prema natrag *G(reedy backward elimination)*
- Optimizacija rojem čestica *(engl. Particle swarm optimization)*
- Raspršenoo pretraživanje *(engl. Scatter Search)*
- *. . .*

Otkrivanje znanja u skupovima podataka

Neki postupci pretraživanja

Iscrpno pretraživanje

- Uzima sve kombinacije atributa (sve podskupove)
- Računalno nije izvedivo osim za male podskupove; $O(2^n)$

Pohlepna pretraživanja (prema naprijed i prema natrag, kombinacija)

Jednostavan heuristički algoritam (heuristički = ne garantira najbolje rješenje)

Prema naprijed:

- Na svaki pojedinačni atribut primijeni se odabrana mjera uspješnosti klasifikacije. Uzima se najbolji.
- Najbolji atribut se uparuje s svim ostalima pojedinačno i mjeri se uspješnost.
- Najbolji par proširuje se s preostalim pojedinačnim atributima
- Nastavlja se sve dok se ne postigne kriterij završetka (npr. stagniranje uspješnosti klasifikacije ili utvrđeni broj atributa)

Otkrivanje znanja u skupovima podataka

Pohlepno pretraživanje prema natrag (eliminacija):

- Započinje se s punim skupom atributa
- Oduzima se pojedini atribut i mjeri (ne)uspješnost
- Odbacuje se najneuspješniji atribut ili skup najneuspješnijih
- Postupak se rekurzivno ponavlja do kriterija završetka (najčešće konačan broj atributa)

- Pohlepno pretraživanje prema naprijed i natrag može se kombinirati u bidirekcijsko pretraživanje
- Ako se u postupku u svakom koraku pamti nekoliko najuspješnijih podskupova radi se o zrakastom pretraživanju (engl. Beam search)

Pretraživanje prema prvom najboljem (engl. Best first)

- Pretraživanje se ne zaustavlja kada uspješnost predikcije/klasifikacije pada (kao kod pohlepnog pretraživanja)
- Čuva se lista podskupova anaiziranih do tada i uređenih po uspješnosti **na koje se pretraživanje može vratiti**

Otkrivanje znanja u skupovima podataka

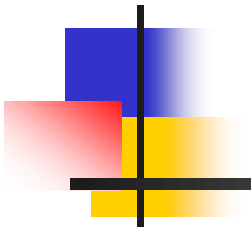
Ocjena kvalitete podskupa atributa (mjere)

- Informacijska dobit (engl. Information gain)
- Omjer informacijske dobiti (engl. Gain ratio)
- Simetrična neizvjesnost (engl. Symetrical uncerainty)
- Međusobna informacija (engl. Mutual information)
- GINI indeks
- Chi-Square
- Euklidska udaljenost
- T-test
- Minimum redundancije i maksimum relevantnosti (mRmR)
- Fisher-ova mjera
- Korelacijski zasnovana mjera
- Las Vegas filter
- . . .

Otkrivanje znanja u skupovima podataka

Neki postupci ocjene kvalitete podskupa atributa

- **Informacijska dobit** - vidi stabla odlučivanja
- **GINI indeks** (nešto jednostavniji izračun od informacijske dobiti) – vidi slučajne šume
- **Euklidska udaljenost** – vidi Učenje i klasifikacija temeljena na pohranjenim primjerima (engl. Instance based learning)
- **Korelacijski zasnovana mjera** – vidi CoiL_natjecanje
- Vrlo jednostavan algoritam **One_R** **izdvaja najznačajniji atribut** i često se upotrebljava u ocjeni atributa – vidi algoritam

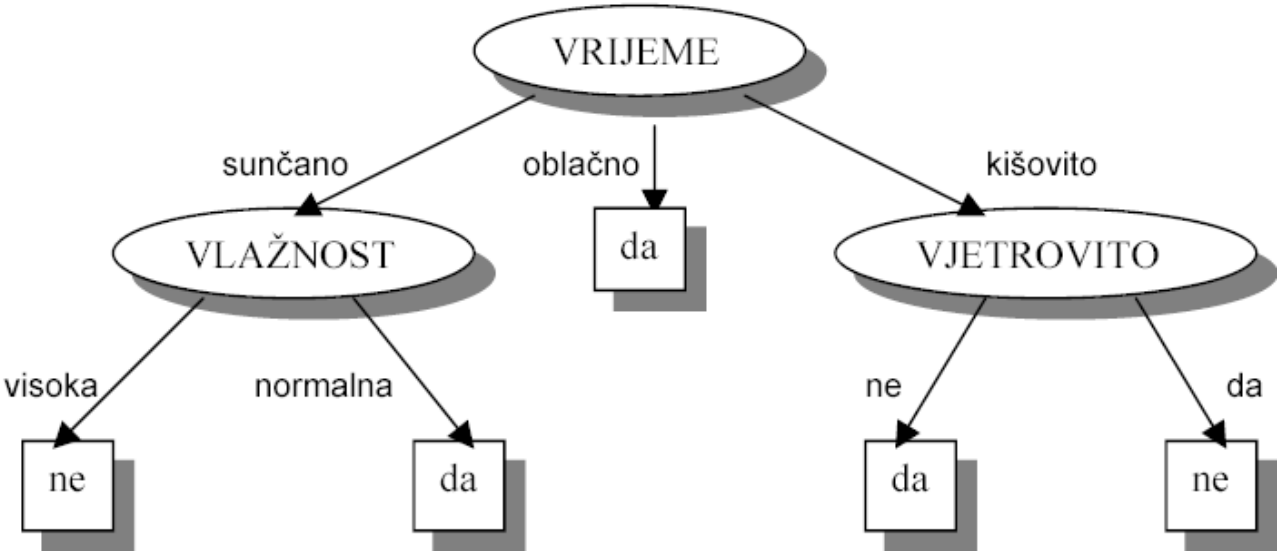


Odabir atributa

Informacijska dobit
(u stablima odlučivanja)

Odabir atributa

Stabla odlučivanja su grafovi u kojem su atributi čvorovi a lukovi njihove moguće nominalne vrijednosti ili intervali numeričkih vrijednosti. Npr. za primjer igranja tenisa:



Listovi stabla (odluka): igrati ili ne

Izgradnja stabla:

Rekurzivno selektiraj atribut i stavi ga u korijen stabla ili podstabla.

Problem:

Kako odrediti toplogiju stabla (redoslijed značajki/atributa) ?

Odabir atributa

Postupak započinje s pripremljenim podacima prikazanim tablično:

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	Topla	Visoka	Ne	Ne
Sunčano	Topla	Visoka	Da	Ne
Oblačno	Topla	Visoka	Ne	Da
Kišovito	Blaga	Visoka	Ne	Da
Kišovito	Hladno	Normalna	Ne	Da
Kišovito	Hladno	Normalna	Da	Ne
Oblačno	Hladno	Normalna	Da	Da
Sunčano	Blaga	Visoka	Ne	Ne
Sunčano	Hladno	Normalna	Ne	Da
Kišovito	Blaga	Normalna	Ne	Da
Sunčano	Blaga	Normalna	Da	Da
Oblačno	Blaga	Visoka	Da	Da
Oblačno	Topla	Normalna	Ne	Da
Kišovito	Blaga	Visoka	Da	Ne

Odabir atributa

Kriterij za odabir redoslijeda atributa

- Svaka grana pojedinog atributa koja završava sa samo jednom vrstom klasifikacije (DA ili NE) je poželjna (jer se ne treba dalje razvijati stablo).
- Tražimo mjeru "čistoće" za svaki čvor (atribut). Tom mjerom bi odabrali atribut s najčišćom djecom (njegovim podčvorovima).
- Mjera koju tražimo je ***informacija*** i daje vrijednost u bitovima.
- Za svaki čvor ta mjera daje očekivanu količinu informacije potrebnu za specificiranje klasifikacije u DA ili NE igrati, primjera koji je dosegao taj čvor.

Odabir atributa

Poželjna obilježja mjere (količina informacije koja nam je potrebna da bi donijeli odluku o klasifikaciji):

- Ako je broj DA=0 ili broj NE=0, informacija je 0 (jednoznačno klasificiranje, ne trebamo dodatnu informaciju).
- Ako je broj DA i NE jednak, informacija koju trebamo za odluku je maksimalna (najveća dilema).
- Mjera mora biti primjenljiva za više razreda (ne samo 2).

- Funkcija koja zadovoljava sva tri uvjeta je ***entropija.***

Za n vrijednosti čije su vjerojatnosti p_i entropija iznosi:

$$Entropija(p_1, p_2, \ldots, p_n) = - p_1 \log p_1 - p_2 \log p_2 - \ldots - p_n \log p_n$$

gdje je $\Sigma_i \, p_i = 1$

- Najčešće se koristi log za bazu 2. Argumenti entropije p_i su razlomci <1, pa je logaritam <0 te zato minus predznak.

Odabir atributa

Izračun informacije za atribute u primjeru igranja tenisa

- Informacijska vrijednost prije kreiranja stabla dana je s idealnom klasifikacijskom kolonom (9 DA i 5 NE, kolona **Igrati** u tablici). Vjerojatnosti su dane frekvencijskom interpretacijom.

  info[9, 5] = entropija(9/14, 5/14) = 0.94 bita

- Mora se analizirati koji atribut **najmanje smanjuje** ovu idealnu vrijednost: 0.94 – info_atributa = ***dobitak***

Atribut: ***Vrijeme*** i sve njegove vrijednosti

Vrijeme.sunčano (2 DA, 3 NE)	info[2, 3] = 0.971 bita
Vrijeme.oblačno (4 DA, 0 NE)	info[4, 0] = 0 bita
Vrijeme.kiša (3 DA, 2 NE)	info[3, 2] = 0.971 bita

Srednja vrijednost atributa ***Vrijeme*** za svih 14 primjera:

info{[2, 3], [4, 0], [3, 2]) = (5/14) 0.971 + (4/14) 0 + (5/14) 0.971 = **0.693**

To predstavlja količinu informacije koja bi bila potrebna za klasifikaciju novoga primjera prema stablu s tri grane atributa ***Vrijeme***.

Odabir atributa

Izračun informacije za atribute u primjeru igranja tenisa

Utjecaj atributa ***Vrijeme*** je:

dobitak(***Vrijeme***) = 0.94 – 0.693 = 0.247 bita

Analogno računamo za ostale atribute:

dobitak(***Temperatura***) = 0.94 – 0.911 = 0.029

dobitak(***Vlažnost***) = 0.94 – 0.788 = 0.152

dobitak/(***Vjetrovito***) = 0.94 – 0.892 = 0.048

Atribut ***Vrijeme*** je korijenski čvor jer ima **najveći dobitak**.

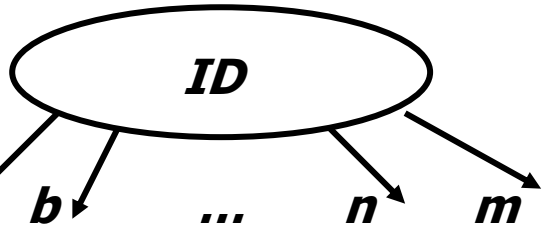
- Dalje računamo dobitke ne više prema klasifikacijskoj koloni **Igrati** nego prema info svake grane atributa ***Vrijeme.***

Odabir atributa

Atributi s visokim grananjem

Neka u tablici za igranje tenisa postoji dodatni atribut Identifikacijski kod ***ID*** s vrijednostima ***a, b, … m, n*** (jedinstvenim za svaki primjer).

Klasifikacija po tome atributu svrstava svaki primjer iz skupa za učenje u jedinstven razred (DA ili NE).



Vrijednosti: ***a*** ***b*** ***…*** ***n*** ***m***

Klasifikacija: NE NE … DA NE

Npr. u ovom retku DA

Računamo info za atribut ***ID*** (svaka grana samo jedan NE ili jedan DA):

info([0, 1], [0, 1], [1, 0], …,[0, 1)]) = 0 jer log 1 = 0

Dobitak atributa ***ID*** : 0.94 - 0 = 0.94 bita = najveća moguća

Izborom ID kao korijenskog čvora prema dobitku međutim nema nikakvu vrijednost jer se nepoznati primjer ne bi mogao uopće klasificirati (nepoznati primjer nema niti jednu vrijednost ***a,*** …, ***m***).

Odabir atributa

Atributi s visokim grananjem

Dobit prema info[] mjeri preferira atribute s visokim grananjem !!

Uvodimo kompenzaciju (kaznu) prema broju grana:

Ako zanemarimo informaciju o klasifikaciji, info vrijednost grananja atributa ***ID*** :

info([1, 1, …, 1)] = -1/14 x log(1/14) x 14 = 3.807

t.j. 14 grana po 1 vrijednost.

Podijelimo izvorni info s 3.807 = ***omjer dobitka***: 0.94 / 3.807 = 0.246

Zbog nepristranosti to moramo učiniti za sve atribute:

Vrijeme: info[5, 4, 5] = 1.577, omjer dobitka = 0.247 / 1.577 = 0.156

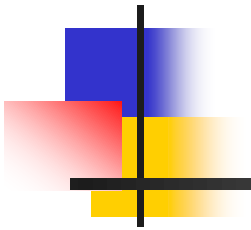
Temperatura: omjer dobitka = 0.029 / 1.362 = 0.021

Vlažnost: omjer dobitka = 0.152 / 1 = 0.152

Vjetrovito: omjer dobitka = 0.048 / 0.985 = 0.049

Nažalost ***ID*** je još uvijek najbolji kao korijenski atribut !!

U praksi provodimo **ad-hoc test** kako bi utvrdili potpuno nekorisne atribute (ovdje atribut ***ID***).



Odabir atributa

GINI nečistoća
(u slučajnim šumama)

Odabir atributa

Slučajne šume - *RF*

(engl. *random forests*) – Breiman, 2001

"GINI impurity" - GINI nečistoća kriterij za izbor značajke u čvoru

- Maksimalno čist čvor – klasificira samo u jedan razred (npr. završni čvorovi su maksimalno čisti).
- Maksimalno nečist čvor – jednak broj klasifikacija po razredima.
- Nastojimo da djeca čvorova budu maksimalno čista, t.j. kriterij odabira značajke je što veće smanjenje nečistoće.
- ***GINI nečistoća za čvor t:*** $\mathbf{i(t) = 1 - \Sigma_i\, p_i(t_i)^2}$

  gdje ***i = 1 ... r*** je broj razreda, a ***p(t)*** su vjerojatnosti klasifikacije primjera u čvoru (to su relativne frekvencije temeljem podataka za učenje) .

Prijeri nečistoće:

- čvor s dva jednako vjerojatna razreda: $i = 1 - (0.5)^2 - (0.5)^2 = 0.5$
- čvor s klasifikacijom samo u jedan razred: $i = 0$ (maksimalno čist čvor).

Odabir atributa

Slučajne šume - *RF*

(engl. *random forests*) – Breiman, 2001

"GINI impurity" - GINI nečistoća kriterij za izbor značajke u čvoru

Prema GINI kriteriju odabire se čvor (dijete) s najmanjom nečistoćom. Ako je ***t*** roditeljski čvor, maksimizira se razlika:

$$\textit{Delta} = i(t) - \Sigma_j [p_j \times i(t_j)]$$

sumira se po svim vrijednostima značajke, t.j. granama (djeci) čvora ***t*** , gdje je ***i(t)*** nečistoća ***t*** čvora (roditelja) , p_j vjerojatnost klasifikacije primjera u j-tu granu (dijete), a $i(t_j)$ je nečistoća j-te grane (djeteta čvora ***t***).

Npr. u primjeru igranja tenisa, nečistoća ciljnog atributa (9 x DA, 5 x NE) je:

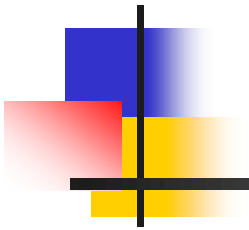
i(ciljni) = $1 - (9/14)^2 - (5/14)^2 = 0.44$ -> nečistoća roditeljskog čvora

U stablu se traži značajka (čvor) koja u svojim granama ima što manju nečistoću te ostvaruje maksimalni ***Delta*** (što bliže 0.44).

Otkrivanje znanja u skupovima podataka

Neki postupci ocjene kvalitete podskupa atributa

- **Informacijska dobit** - vidi stabla odlučivanja
- **GINI indeks** (nešto jednostavniji izračun od informacijske dobiti) – vidi slučajne šume
- **Euklidska udaljenost** – vidi Učenje i klasifikacija temeljena na pohranjenim primjerima (engl. Instance based learning)
- **Korelacijski zasnovana mjera** – vidi CoiL_natjecanje
- Vrlo jednostavan algoritam **One_R** **izdvaja najznačajniji atribut** i često se upotrebljava u ocjeni atributa – vidi algoritam



Odabir atributa

Indukcija jednog atributa ***1R***

(engl. *1 rule*)

Odabir atributa

Osnovna zamisao:

1R generira pravilo koje testira jedan atribut.

Algoritam:

```
Za svaki atribut,
    Za svaku vrijednost atributa napravi pravilo:
      - izbroji koliko često se pojavljuje razred
      - pronađi razred koji se najčešće pojavljuje
      - napravi pravilo koje dodjeljuje taj razred
        toj vrijednosti atributa.
    Izračunaj procjenu greške pravila.
Izaberi pravila s najmanjom procjenom greške.
```

Npr. igranje tenisa – vidi slijedeću sliku

Atribut:	***Vrijeme***
Vrijednost:	***Sunčano***
Klasifikacija:	(2xDA, 3xNE), najčešće ***NE***
Pogreška:	Od 5 vrijednosti ***Sunčano***, 2 su pogrešne klasifikacije (2xDA)

Odabir atributa

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	Topla	Visoka	Ne	Ne
Sunčano	Topla	Visoka	Da	Ne
Oblačno	Topla	Visoka	Ne	Da
Kišovito	Blaga	Visoka	Ne	Da
Kišovito	Hladno	Normalna	Ne	Da
Kišovito	Hladno	Normalna	Da	Ne
Oblačno	Hladno	Normalna	Da	Da
Sunčano	Blaga	Visoka	Ne	Ne
Sunčano	Hladno	Normalna	Ne	Da
Kišovito	Blaga	Normalna	Ne	Da
Sunčano	Blaga	Normalna	Da	Da
Oblačno	Blaga	Visoka	Da	Da
Oblačno	Topla	Normalna	Ne	Da
Kišovito	Blaga	Visoka	Da	Ne

Odabir atributa

Postupak provodimo za **sve atribute i njihove vrijednosti**:

Atribut: ***Vrijeme***
Vrijednost: ***Oblačno***
Klasifikacija: (4xDA, 0xNE), najčešće ***DA*** pa je pogreška NE
Pogreška: Od 4 vrijednosti ***Oblačno*** nema pogrešne klasifikacije.

Atribut	**KLasifikacija**	**Pogreške**	**Ukupno pogreške**
Vrijeme	Sunčano – NE	2/5	4
	Oblačno – DA	0/4	
	Kišovito – DA	2/5	
Temperatura	Toplo – NE	2/4	5
	Blago – DA	2/6	
	Hladno – DA	1/4	
Vlažnost	Normalna – DA	1/7	4
	Visoka – NE	3/7	
Vjetrovito	Nije – DA	2/8	5
	Je – NE	3/6	

Odabir atributa

Za ***Temperatura.Toplo*** slučajan izbor između dva jednaka ishoda (2xDA, 2xNE).

Za ***Vjetrovito.Je*** slučajan izbor između dva jednaka ishoda (3xDA, 3xNE).

Dva atributa imaju jednaku uspješnost (4 pogrešaka u 14 primjera).
Slučajno odabiremo atribut ***Vrijeme***, pa je pravilo:

Ako ***Vrijeme*** =	***Sunčano***	NE	igrati
	Oblačno	DA	igrati
	Kišovito	DA	igrati

(Čini se da se tenis igra u dvorani).

- Do sada su u primjeru tenisa svi atributi bili nominalni (kategorički).
- Ukoliko su vrijednosti nekih atributa numeričke potrebno je uvesti diskretizaciju, t.j. preslikavanje numeričkih vrijednosti u konačan broj nominalnih.

Problem: **Kako provesti diskretizaciju** ?

Odabir atributa

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	32	85	Ne	Ne
Sunčano	27	90	Da	Ne
Oblačno	30	86	Ne	Da
Kišovito	17	96	Ne	Da
Kišovito	15	80	Ne	Da
Kišovito	12	70	Da	Ne
Oblačno	11	65	Da	Da
Sunčano	19	95	Ne	Ne
Sunčano	16	70	Ne	Da
Kišovito	22	80	Ne	Da
Sunčano	22	70	Da	Da
Oblačno	19	90	Da	Da
Oblačno	28	75	Ne	Da
Kišovito	18	91	Da	Ne

Odabir atributa

Atribut ***Temperatura***

Vrijednosti se urede (po veličini) i uz svaku vrijednost navede se razred.

11	12	15	16	17	18	19	19	22	22	27	28	30	32
da	ne	da	da	da	ne	ne	da	da	da	ne	da	da	ne

Niz se diskretizira na mjestima promjene razreda.

Granice razreda su: 11.5, ,13.5, 17.5, 19, 24.5, 27.5, 31

11	12	15	16	17	18	19	19	22	22	27	28	30	32
da	ne	da	da	da	ne	ne	da	da	da	ne	da	da	ne

Problemi:

1. vrijednost 19 se klasificira u dva razreda.

   Rješenje: granica se pomakne desno između 19 i 22 (na 20.5). Novi problem: ne postoji čisti razred već samo većinski (2x**NE**, 1xDA).

2. Postoji mnogo kategorija.

Odabir atributa

- ***1R*** prirodno gravitira prema odabiru atributa koji se dijeli na mnogo particija, jer finija raspodjela čini izglednijim da primjer ima isti razred kao većina u particiji.

  Ekstreman slučaj: Svaki primjer zaseban razred (zasebna particija).

  Npr. novi atribut je identifikacijski kod

  Na skupu za učenje = pogreška 0 ("overfitting")

  Na testnom skupu = nema slaganja (pogreška 100%)

- Rješenje toga problema je **heurističko pravilo**:
  - Minimalan broj primjera (instancija) većinskog razreda u pojedinoj particiji je 3 ili 4.
  - Ako dvije susjedne particije imaju isti većinski razred mogu se spojiti.

Odabir atributa

Primjena heuristike (uz 3 primjera većinskog razreda i spajanjem):

11	12	15	16	17	18	19	19	22	22	27	28	30	32
da	ne	da	da	da	ne	ne	da	da	da	ne	da	da	ne

To dovodi do pravila ***1R***:

Ako ***Temperatura*** ≤ 24.5 DA igrati

> 24.5 NE igrati

Za drugu particiju je jednak broj DA i NE; slučajno odabrano NE. Ukoliko bi bilo odabrano DA, sve postaje jedna particija s klasifikacijiom DA.

Gornje pravilo ima 5 pogrešaka na promatranom skupu primjera.

Analogno za atribut ***Vlažnost*** (najbolje ***1R*** pravilo jer samo 3 pogreške):

Ako ***Vlažnost*** ≤ 82.5 DA igrati

$> 82.5 \leq 95.5$ NE igrati

> 95.5 DA igrati

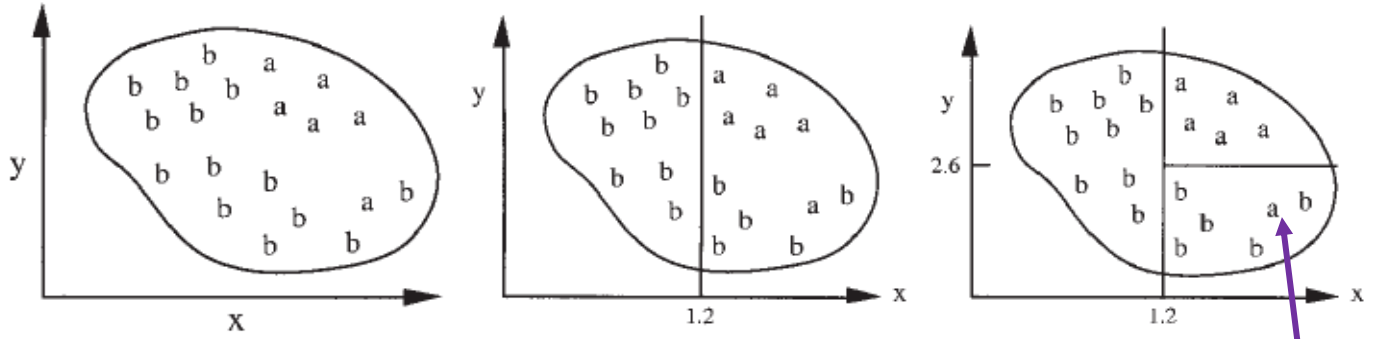
Provedena analiza 1993 god., uz realne probleme i 6 primjera u particiji vrlo dobre rezultate ("Jednostavna rješenja su u prednosti – Occam").

Primjer odabira (engl. *feature selection*) relevantnih atributa i njihovih vrijednosti **specijalizacijom** klasifikacijskih pravila (kriterij preciznosti) - **PRISM**

Indukcija pravila pokrivanjem - ***PRISM***

- Princip pokrivanja: Pronađi **opis pravilima** koji uključuje što više primjera jednog razreda uz što manje primjera ostalih razreda.

Neka su predstavljeni primjeri **za učenje** u dvije dimenzije (dva atributa x i y):



Želimo izdvojiti primjere "a" razreda.

Ako (x > 1.2) tada "a". Pravilo uključuje neke primjere koji **ne pripadaju** u "a" razred (uključena su 3 primjera iz "b" razreda).

Ako (x > 1.2) **i** (y > 2.6) tada "a".

Prošireno pravilo više ne uključuje "b" primjere, ali **izostavlja** jedan "a".

Za potpun opis "a" uključuje se **DODATNO** pravilo (opis "a" s 2 pravila):

Ako (x > 1.4) i (y < 2.4) tada "a".

Indukcija pravila pokrivanjem - ***PRISM***

- U nastavku analiziramo **binarnu klasifikaciju** (za više razreda može se svesti na pozitivne primjere jednog razreda i negativne primjere za sve ostale razrede).
- Neka pravilo **pokriva** (obuhvaća, diskriminira) ***t*** primjera, od kojih ***p*** pripada pozitivnom razredu (a ***t-p*** negativnom).
- U konstrukciji pravila sukcesivno izabiremo članove (atribute i njihove vrijednosti) na AKO strani koji maksimiziraju ***p/t* = preciznost** (engl. *accuracy*).
- Pravilo **Ako (x> 1.2) tada "a"** ,ima preciznost 8/11. To je 8 a-ova i 11 primjera koje pokriva (8 a-ova i 3 b-ova).
- Ova mjera preferira pravila koja pokrivaju malo primjera u kojima nema negativnih. To je **specijalizacija** pravila.
- Problem: maksimalna specijalizacija pravila pokriva samo jedan primjer (preciznost je 100%).
- Potrebno je na skupu primjera za testiranje odrediti kolika specijalizacija je optimalna. To je postupak koji traži kompromis između **specijalizacije** i **generalizacije**.

Indukcija pravila pokrivanjem - ***PRISM***

PRISM – algoritam temeljen na kriteriju preciznost ***p/t*** :

```
Za svaki razred C:
Inicijaliziraj E skup primjera.
      Kreiraj pravilo R s praznom AKO stranom koje
            predviđa razred C.
      Dok R savršeno radi (ili više nema atributa):
      |     Za svaki atribut A koji nije spomenut u R
      |           i svaku vrijednost v:
      |                 Razmotri dodatni uvjet A=v.
      |                 Odaberi A i v da maks p/t
      |                  (jednakost riješi s većim p).
      Dodaj A=v u R.
      Iz E makni primjere pokrivene u R.
```

Indukcija pravila pokrivanjem - ***PRISM***

Godine	Dioptrija	Astigmatizam	Stvaranje suza	Preporuka
mlad	minus	ne	smanjeno	ne
mlad	minus	ne	normalno	meke
mlad	minus	da	smanjeno	ne
mlad	minus	da	normalno	tvrde
mlad	plus	ne	smanjeno	ne
mlad	plus	ne	normalno	meke
mlad	plus	da	smanjeno	ne
mlad	plus	da	normalno	tvrde
prije dalekovidnosti	minus	ne	smanjeno	ne
prije dalekovidnosti	minus	ne	normalno	meke
prije dalekovidnosti	minus	da	smanjeno	ne
prije dalekovidnosti	minus	da	normalno	tvrde
prije dalekovidnosti	plus	ne	smanjeno	ne
prije dalekovidnosti	plus	ne	normalno	meke
prije dalekovidnosti	plus	da	smanjeno	ne
prije dalekovidnosti	plus	da	normalno	ne
dalekovidnost	minus	ne	smanjeno	ne
dalekovidnost	minus	ne	normalno	ne
dalekovidnost	minus	da	smanjeno	ne
dalekovidnost	minus	da	normalno	tvrde
dalekovidnost	plus	ne	smanjeno	ne
dalekovidnost	plus	ne	normalno	meke
dalekovidnost	plus	da	smanjeno	ne
dalekovidnost	plus	da	normalno	ne

Indukcija pravila pokrivanjem - ***PRISM***

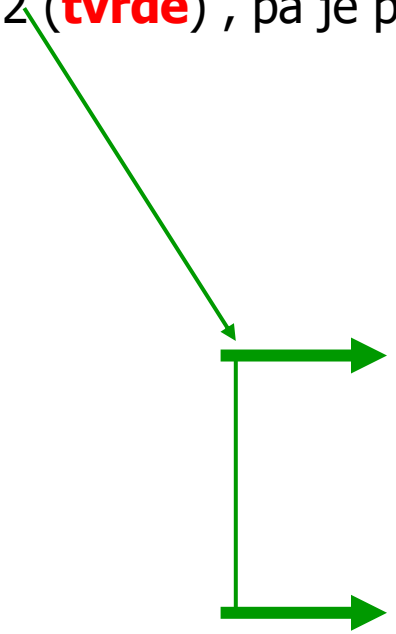
Konstruiramo pravilo za preporuku ***tvrde*** leće (samo taj razred).

Početno pravilo ima praznu AKO stranu: ***Ako (?) tada tvrde leće***.

Ovo **najopćenitije pravilo** ima točnost 4/24, odnosno 17% što ne zadovoljava.

Krećemo sa specijalizacijom: za nepoznati član postoji 9 opcija (3 vrijednosti za ***Godine*** i po 2 za ***Dioptrija***, ***Astigmatizam***, ***Stvaranje_suza***) :

Npr.. ***atribut.vrijednost*** za ***Godine.mlad*** pokriva 8 primjera, a od toga ispravno klasificira 2 (**tvrde**) , pa je preciznost te opcije ***p/t*** = 2/8



Godine	Preporuka
mlad	ne
mlad	meke
mlad	ne
mlad	tvrde
mlad	ne
mlad	meke
mlad	ne
mlad	tvrde

Indukcija pravila pokrivanjem - ***PRISM***

Analogno se računa preciznost svakog atributa i svake njegove vrijednosti.

Godine.mlad	2/8
Godine.prije_dalekovidnosti	1/8
Godine.dalekovidnost	1/8
Dioptrija.minus	3/12
Dioptrija.plus	1/12
Astigmatizam.ne	0/12
Astigmatizam.da	**4/12**
Stvaranje suza.smanjeno	0/12
Stvaranje suza.normalno	**4/12**

Između dvije jednake vrijednosti odaberemo slučajno ***Astigmatizam.da***, te pravilo sada glasi:

Ako Astigmatizam.da i (?) tada tvrde leće.

Ovo pravilo pokriva 12 primjera pa nastavljamo specijalizaciju pravila na tom **smanjenom** (**pokrivenom**) skupu od 12 primjera.

Indukcija pravila pokrivanjem - ***PRISM***

Godine	Dioptrija	Astigmatizam	Stvaranje suza	Preporuka
mlad	minus	da	smanjeno	ne
mlad	minus	da	normalno	tvrde
mlad	plus	da	smanjeno	ne
mlad	plus	da	normalno	tvrde
prije dalekovidnosti	minus	da	smanjeno	ne
prije dalekovidnosti	minus	da	normalno	tvrde
prije dalekovidnosti	plus	da	smanjeno	ne
prije dalekovidnosti	plus	da	normalno	ne
dalekovidnost	minus	da	smanjeno	ne
dalekovidnost	minus	da	normalno	tvrde
dalekovidnost	plus	da	smanjeno	ne
dalekovidnost	plus	da	normalno	ne

Za ovaj smanjeni skup primjera računamo preciznog svakog **preostalog** atributa i njegove vrijednosti.

Postoji 7 opcija (3 vrijednosti za ***Godine*** i po 2 za ***Dioptrija*** i ***Stvaranje_suza***).

Indukcija pravila pokrivanjem - ***PRISM***

Uz ***Astigmatizam.da*** imamo preciznosti p/t:

Godine.mlad	2/4
Godine.prije dalekovidnosti	1/4
Godine.dalekovidnost	1/4
Dioptrija.minus	3/6
Dioptrija.plus	1/6
Stvaranje_suza.smanjeno	0/6
Stvaranje_ suza.normalno	**4/6**

Očito je ***Stvaranje_suza.normalno*** najbolja opcija, pa pravilo glasi:

Ako Astigmatizam.da i Stvaranje_suza.normalno i (?) tada tvrde leće.

- Moguće je ovdje zaustaviti specijalizaciju pravila (to treba potvrditi mjerom uspješnosti klasifikacije na testnom skupu).
- Ako želimo nastaviti sa specijalizacijom, analiziramo primjere (ima ih t=6) koje sada pokriva navedeno prošireno pravilo.

Indukcija pravila pokrivanjem - ***PRISM***

Uz ***Astigmatizam.da*** i ***Stvaranje_suza.normalno*** postoji 6 primjera:

Godine	Dioptrija	Astigmatizam	Stvaranje suza	Preporuka
mlad	minus	da	normalno	tvrde
mlad	plus	da	normalno	tvrde
prije dalekovidnosti	minus	da	normalno	tvrde
prije dalekovidnosti	plus	da	normalno	ne
dalekovidnost	minus	da	normalno	tvrde
dalekovidnost	plus	da	normalno	ne

Za novi uvjet postoji 5 opcija (3 za ***Godine*** i 2 za ***Dioptrija***). Računamo p/t:

Godine.mlad	2/2
Godine.prije_dalekovidnosti	1/2
Godine.dalekovidnost	1/2
Dioptrija.minus	**3/3**
Dioptrija.plus	1/3

Odabir između prve i četvrte opcije je u korist četvrte zbog većeg pokrivanja ***p***.

Indukcija pravila pokrivanjem - ***PRISM***

Prošireno (specijalizirano) pravilo glasi:

Ako Astigmatizam.da i Stvaranje_suza.normalno i Dioptrija.minus tada tvrde leće.

Ovo pravilo pokriva **3** primjera (od izvornih 6) koji klasificiraju ***tvrde*** leće.

Godine	Dioptrija	Astigmatizam	Stvaranje suza	Preporuka
mlad	minus	da	normalno	tvrde
mlad	plus	da	normalno	tvrde
prije dalekovidnosti	minus	da	normalno	tvrde
prije dalekovidnosti	plus	da	normalno	ne
dalekovidnost	minus	da	normalno	tvrde
dalekovidnost	plus	da	normalno	ne

?

Daljnje pokrivanje (specijalizacija) ovoga pravila da se obuhvati i četvrti primjer za ***tvrde*** leće nije moguće, jer za njega treba ***Dioptrija.plus*** (a pravilo je već odabralo ***Dioptrija.minus***).

Indukcija pravila pokrivanjem - ***PRISM***

- Iz cijelog skupa od 24 primjera **izbace se tri primjera pokrivena do sada** induciranim pravilom i traži se **dodatno** pravilo na preostalom skupu od **21.** primjera.
- Ponovo se započinje s praznom ako stranom: ***Ako (?) tada tvrde leće***.
- Analognim postupkom prema maksimalnom ***p/t*** slijede najbolji izbori za članove AKO strane:

  Godine.mlad je najbolji izbor za prvi uvjet, uz p/t = 1/7

  Astigmatizam.da je najbolji izbor za drugi uvjet je, uz p/t = 1/3

  Stvaranje_suza.normalno je najbolji treći uvjet uz p/t = 1/1
- **Dodatno** pravilo je:

  Ako Godine.mlad i Astigmatizam.da i Stvaranje_suza.normalno tada tvrde leće.
- Ovo dodatno pravilo pokriva i jedan (prvi) od 3 primjera pokrivena ranijim pravilom. To ne smeta jer se radi o klasifikaciji istog razreda (***tvrde***).

- Opisani postupak generiranja pravila ponavlja se za ostala dva razreda (***meke*** leće, ***ne*** leće).

Indukcija pravila pokrivanjem - ***PRISM***

Sažetak *PRISM* algoritma:

- Algoritam "**podijeli pa vladaj**" izborom ***atribut.vrijednost*** prema kriteriju ***p/t = pozitivni/pokriveni = preciznost*** , rekurzivno odvaja pokrivene primjere i traži ponovo.
- Generira pravila postupkom koji sugerira redoslijed primjene (uređena lista pravila) jer se primjeri pokriveni ranijim pravilom izbacuju iz skupa. **Primjena po čvrstom redoslijedu nije nužna** jer se radi o klasifikaciji jednog (istog) razreda (pravilo se odnosi na neki primjer ili ne).
- Može se pojaviti **konflikt u klasifikaciji** nepoznatih primjera (prihvaćaju ga pravila koja ga svrstavaju u više razreda). Heuristika: izaberi do tada najuspješnije pravilo i prihvati njegovu klasifikaciju.
- Može se dogoditi da se **niti jedno pravilo ne odnosi na novi primjer**. Heuristika: pridruži takav primjer najbrojnijem razredu.

Kreiranje novih atributa iz ulaznih podataka

(engl. feature extraction)

Otkrivanje znanja u skupovima podataka

Kreiranje skupa atributa (engl. Feature extraction)

- To je postupak **transformacije ulaznih podataka u skup značajki** koje omogućuju razlikovanje kategorija ulaznih obrazaca/uzoraka. Transformacija mora sačuvati bitne informacije ulaznog skupa.
- Transformacijom se reducira ulazni skup podataka pa se ponekad naziva smanjenje dimenzionalnosti (iako je **odabir manjeg skupa relevantnih atributa** također smanjenje dimenzionalnosti).
- To je postupak u kojem tražimo karakteristike koje su svojstvene i diskriminantne za neki objekt ili proces. Potrebno poznavanje domene.
- To je postupak u kojem se kontinuirani signal transformira u diskretni skup podataka s ciljem dubinske analize.
- To je postupak kreiranja **novih atributa** iz izvornog skupa atributa s ciljem povećanja efikasnosti i točnosti modela. Klasifikacija se izvodi s novim skupom.
- **Često se najprije provodi kreiranje skupa a zatim odabir relevantnih atributa.**

Procesi kreiranja atributa (engl. Feature extraction)

Neke popularne tehnike tehnike:

- Independent component analysis
- Stroj potpornih vektora (SVM)
- Partial least squares
- Principal component analysis (PCA)
- Multifactor dimensionality reduction
- Nonlinear dimensionality reduction
- Multilinear Principal Component Analysis
- Multilinear subspace learning
- . . .

Otkrivanje znanja u skupovima podataka

Primjer PCA

- Računa svojstvene vektore (daju smjer/osi u novom prostoru) i svojstvene vrijednosti (daju iznose na osima) iz dekompozicije kovarijantne (korelacijske) matrice (matrice kovarijanci vrijednosti).
- To su „**glavne komponente**" = **novi atributi**
- Nakon rangiranja uzima se samo nekoliko najviših svojstvenih vrijednosti prema njihovom objašnjavanju varijance u podacima, t.j. prema sadržaju informacije (omjer varijance pojedine glavne komponente prema ukupnoj varijanci).
- Zatim se konstruira tzv. „projekcijska matrica" veličine
  - (broj_izvornih_atributa X broj_novih_atributa)
- „Projekcijskom matricom" **transformiraju se izvorni podaci u novi prostor manjih dimenzija.**
- Novi atributi tipično gube fizikalnu interpretaciju.

Otkrivanje znanja u skupovima podataka

- Kovarijantna matrica pokazuje korelaciju između parova atributa.

Za dvije varijable:

$$\begin{bmatrix} var(x), cov(x,y) \\ cov(x,y), var(y) \end{bmatrix}$$

$cov(x, x) = var(x)$

$cov(x, y) = cov(y, x)$

Varijanca: $var(x) = (1/(n-1)) \Sigma (x_i - x_{mean})^2$

Kovarijannca: $cov(x, y) = (1/(n-1)) \Sigma (x_i - x_{mean})(y_i - y_{mean})$

- Iz kovarijantne matrice računamo svojstvene vektore i svojstvene vrijednosti.
- Svojstvene vrijednost su rješenja jednadžbe: $|\mathbf{A}-\lambda \cdot \mathbf{I}|=0$

  A = kovarijantna matrica, I = jedinična matrica.
- Svojstvene vrijednosti pokazuju varijancu podataka. To su Glavne komponente (PC – Principal Components)
- Broj svojstvenih vrijednosti (PC) = dimenziji podataka (broj atributa/varijabli)

Otkrivanje znanja u skupovima podataka

Primjer PCA https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html

"Iris" dataset (UCI repozitorij): 150 primjeraka, **4 atributa**, 3 razreda

Transformacija s 4 na 2 dimenzije (**dva nova** atributa PC_1 i PC_2)

Izvorni atributi:

1. ***sepal length* in cm**
2. ***sepal width* in cm**
3. ***petal length* in cm**
4. ***petal width* in cm**

Petal – latica
Sepal – dio ispod latice
Iris = Perunika