21. Vrednovanje modela

Strojno učenje 1, UNIZG FER, ak. god. 2022./2023.

Jan Šnajder, predavanja, v1.3

Do sada smo se na ovom kolegiju bavili algoritmima strojnog učenja: kako ti algoritmi rade, od čega su sastavljeni i koje induktivne pretpostavke su u njih ugrađene. Upoznali smo niz algoritama, dok je naravno još više onih koje nismo ni spomenuli. Činjenica da su nam na raspolaganju brojni algoritmi je izvrsna, ali ujedno predstavlja praktičan izazov: kako znati koji je algoritam najbolji za naš problem? Da bismo mogli odgovoriti na ovo pitanje, moramo nekako usporediti predikcije algoritama. Kriterij koji nam je pritom ključan je, naravno, točnost algoritma na neviđenim podatcima, odnosno pogreška generalizacije. A da bismo što bolje procijenili tu točnost odnosno pogrešku, moramo znati kako **ispravno vrednovati (evaluirati) algoritam**. Premda se na prvi pogled možda ne čini tako, vrednovanje algoritama strojnog učenja vrlo je sklizak teren i česti su slučajevi gdje ono nije ispravno odrađeno, kako u industrijskoj praksi tako i u akademiji. Cilj ovog predavanja jest da vi nipošto ne budete među onima koji ne znaju vrednovati algoritme strojnog učenja!

Tri su stvari bitne za vrednovanje algoritama strojnog učenja: koju **mjeru vrednovanja** koristiti, kako pravedno (realistično) **procijeniti pogrešku modela** i kako napraviti **statističku analizu rezultata** kako bismo bili donekle sigurni da naš rezultat nije puka slučajnost. Danas ćemo se baviti prvim dvjema temama (mjerama vrednovanja i procjenom pogreške), dok statističku usporebu modela ostavljamo za iduće predavanje. Također, u ovom predavanju ograničit ćemo se na vrednovanje klasifikatora; vrednovanje regresije i algoritama grupiranja je važno, ali izvan dosega.

[1] [2]

1 Osnovne mjere vrednovanja

Mjere vrednovanja (evaluacijska mjera, evaluacijska metrika) je metoda koja na neki način kvantificira točnost (ili pogrešku) klasifikatora. Postoji mnogo mjera vrednovanja, i zapravo je najveći izazov izabrati onu mjeru koja je prikladna za problem koji rješavamo.

[3]

Polazna točka kod vrednovanja klasifikatora jest da klasifikator primijenimo na skup primjera (tipično je to ispitni skup, ali možemo ga primijeniti i na cijeli skup podataka) tako da za svaki primjer dobijemo predikciju klasifikatora. Te oznake nazivamo **predviđene oznake** (engl. *predicted labels*). Naravno, da bismo mogli vrednovati klasifikator, za isti taj skup primjera moramo imati i točne oznake svih primjera. Te oznake nazivamo **stvarne oznake** (engl. *true (actual) labels*). Ove oznake možemo organizirati u vektore oznaka. Tako ćemo tako dobiti **vektor predviđenih oznaka**:

$$\mathbf{y}_{pred} = (y_{pred}^{(1)}, \ldots, y_{pred}^{(i)}, \ldots, y_{pred}^{(N)})^{\mathrm{T}}$$

gdje je $y_{pred}^{(i)} = h(\mathbf{x}^{(i)})$ predviđena oznaka za i-ti primjer. Analogno, imamo i **vektor stvarnih oznaka**:

$$\mathbf{y}_{true} = (y_{true}^{(1)}, \ldots, y_{true}^{(i)}, \ldots, y_{true}^{(N)})^{\mathrm{T}}$$

gdje je $y_{true}^{(i)}$ stvarna oznaka za i-ti primjer. Ovi su vektori jednake duljine; pretpostavimo da je ta duljina jednaka N, broju primjera u skupu označenih primjera.

Sve mjere vrednovanja kojima ćemo se danas baviti temelje se na usporedbi vektora $\mathbf{y}_{pred}$ i vektora $\mathbf{y}_{true}$. Očito, ako želimo izračunati običnu **točnost** (engl. *accuracy*) klasifikatora, možemo jednostavno pobrojati za koliko se primjera te oznake podudaraju i taj broj podijeliti s ukupnim brojem primjera: 4

$$Acc = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{y_{pred}^{(i)} = y_{true}^{(i)}\}$$

Pogrešku onda možemo izračunati jednostavno kao $1 - Acc$.

Ograničimo se u nastavku na problem binarne klasifikacije (višeklasnom klasifikacijom bavit ćemo se kasnije). Za binarnu klasifikaciju oznake su $y \in \{0, 1\}$, pa su $\mathbf{y}_{pred}$ i $\mathbf{y}_{true}$ binarni vektori. U tom slučaju točnost možemo definirati i kao $Acc = \frac{1}{N}\mathbf{y}_{pred}^{\mathrm{T}}\mathbf{y}_{true}$.

1.1 Matrica zabune

Mjera točnosti koju smo upravo definirali samo je jedna od mnogo mjera koje možemo izračunati na temelju usporedbe vektora $\mathbf{y}_{pred}$ i $\mathbf{y}_{true}$. Mjera točnosti je intuitivna, ali, kao što ćemo uskoro vidjeti, nije uvijek prikladna. Koje još sve mjere vrednovanja možemo izračunati postaje jasnije ako iz vektora $\mathbf{y}_{pred}$ i $\mathbf{y}_{true}$ izgradimo **matricu zabune (kontingencije)** (engl. *confusion (contingency) matrix*). Matrica zabune je kvadratna matrica koja na sažet način prikazuje broj podudaranja i nepodudaranja predikcija i stvarnih oznaka. Za binarni klasifikator, matrica zabune je dimenzija 2×2 i općenito izgleda ovako:

$$\begin{array}{c} \\ y_{pred} = 1 \\ y_{pred} = 0 \end{array} \begin{array}{c} \begin{array}{cc} y_{true} = 1 & y_{true} = 0 \end{array} \\ \begin{pmatrix} \mathrm{TP} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TN} \end{pmatrix} \end{array}$$

Ovdje retci odgovaraju predviđenim oznakama (*pred*), a stupci stvarnim oznakama (*true*). Razlikujemo četiri vrste slučajeva, koji odgovaraju elementima matrice:

- **stvarno pozitivni** (engl. *true positive, TP*): predikcija je 1 i stvarna oznaka je 1;
- **lažno pozitivni** (engl. *false positive, FP*): predikcija je 1 a stvarna oznaka je 0;
- **lažno negativni** (engl. *false negative, FN*): predikcija je 0 a stvarna oznaka je 1;
- **stvarno negativni** (engl. *true negative, TN*): predikcija je 0 i stvarna oznaka je 0. 5

Primijetite da, kako je uobičajeno u strojnom učenju, ovdje za "negativan" primjer koristimo oznaku $y = 0$. Za zadane konkretne vektore oznaka $\mathbf{y}_{pred}$ i $\mathbf{y}_{true}$, matrica zabune bilježi ukupan broj primjera koji su stvarno pozitivni, lažno pozitivni, lažno negativni i stvarno negativni. Dakle, elementi matrice zabune su:

$$\mathrm{TP} = \sum_{i=1}^{N} \mathbf{1}\{y_{pred}^{(i)} = y_{true}^{(i)} = 1\} \qquad \mathrm{FP} = \sum_{i=1}^{N} \mathbf{1}\{y_{pred}^{(i)} = 1 \wedge y_{true}^{(i)} = 0\}$$

$$\mathrm{FN} = \sum_{i=1}^{N} \mathbf{1}\{y_{pred}^{(i)} = 0 \wedge y_{true}^{(i)} = 1\} \qquad \mathrm{TN} = \sum_{i=1}^{N} \mathbf{1}\{y_{pred}^{(i)} = y_{true}^{(i)} = 0\}$$

Pogledajmo napokon jedan primjer.

▸ **PRIMJER**

Raspolažemo sa $N = 7$ primjera. Predikcije binarnog klasifikatora i stvarne oznake primjera su sljedeće:

$$\mathbf{y}_{pred} = (1, 0, 1, 0, 0, 0, 1)^{\mathrm{T}}$$
$$\mathbf{y}_{true} = (1, 0, 0, 0, 1, 0, 0)^{\mathrm{T}}$$

Iz ovoga dobivamo sljedeću matricu zabune:

$$\begin{array}{c} \\ y_{pred}=1 \\ y_{pred}=0 \end{array} \begin{array}{c} \begin{array}{cc} y_{true}=1 & y_{true}=0 \end{array} \\ \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix} \end{array}$$

Iz matrice zabune možemo vidjeti da je najviše stvarno negativnih primjera: od pet negativnih primjera, tri su stvarno negativna, a dva lažno negativna. Od dva primjera koji su pozitivni, jedan je stvarno pozitivan, a jedan je lažno negativan.

Primijetite da smo matricu zabune definirali tako da retci odgovaraju predviđenoj oznaci, a stupci stvarnoj oznaci, te smo prvo naveli vrijednost 1 a zatim vrijednost 0. Imajte na umu da smo takvu organizaciju proizvoljno odabrali. U literaturi ćete naići na drugačije organizirane matrice zabune, pa je dobro uvijek prvo provjeriti što znače pojedini elementi.

Pogledajmo još jedan primjer.

▸ **PRIMJER**

Testiramo klasifikator za dijagnostiku bolesti na $N = 150$ primjera. Usporedbom predviđenih i stvarnih oznaka dobili smo sljedeću matricu zabune:

$$\begin{array}{c} \\ y_{pred}=1 \\ y_{pred}=0 \end{array} \begin{array}{c} \begin{array}{cc} y_{true}=1 & y_{true}=0 \end{array} \\ \begin{pmatrix} 6 & 12 \\ 2 & 130 \end{pmatrix} \end{array}$$

Iz ove matrice možemo iščitati da je ukupan broj primjera TP + FP + FN + TN = 150. Također, možemo iščitati da je ukupan broj pozitivnih primjera jednak TP + FN = 8, dok je ukupan broj negativnih primjera jednak TN + FP = 142. Brojevi na dijagonali matrice zabune odgovaraju točno klasificiranim primjerima, pa je ukupan broj točno klasificiranih primjera jednak tragu matrice (zbroju dijagonalnih elemenata). Ovdje je to TP + TN = 136.

1.2 Mjere vrednovanja nad matricom zabune

Sada kada znamo što je matrica zabune, nad njom možemo definirati niz mjera vrednovanja. Krenimo s očitim. **Točnost** (engl. *accuracy*) je udio točno klasificiranih primjera u skupu svih primjera:

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

U prethodnom primjeru, točnost je jednaka $Acc = \frac{136}{150} = 90.7\%$.

Već smo rekli da je točnost intuitivna mjera vrednovanja, ali da nije uvijek prikladna. Što je točno problem s točnošću? Problem je ovaj: ako je skup podataka izrazito **neuravnotežen** (engl. *imbalanced*) – a to znači da ili imamo mnogo više pozitivnih primjera od negativnih primjera, ili obrnuto – onda je trivijalno ostvariti klasifikator koji ima visoku točnost (odnosno nisku pogrešku). Jednostavno, klasifikator definiramo tako da uvijek predviđa većinsku klasu. Npr., u gornjem primjeru, broj pozitivnih primjera je 8, a broj negativnih je 142. Ako napravimo klasifikator koji jednostavno uvijek predviđa negativnu oznaku ($y = 0$), točnost takvog modela bit će visokih 142/150 = 94.67%. 6

Zbog toga su osmišljene alternativne mjere vrednovanja. Zapravo, kad pogledamo konfuzijsku matricu, vidimo da ona nudi mnoge mogućnosti kako definirati mjere vrednovanja (što zbrajati i što s čime dijeliti). Tako se pored točnosti uobičajeno koriste sljedeće mjere:

- **Preciznost** (engl. *precision*):

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Preciznost definiramo kao udio stvarno pozitivih primjera (TP) u skupu svih primjera koje je klasifikator označio pozitivno (TP + FP). Idealno, P = 1, tj. svi primjeri koje je klasifikator označio pozitivno doista i jesu pozitivni.

- **Odziv** (engl. *recall*):

$$\mathrm{R} = \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Odziv je udio stvarno pozitivnih primjera (TP) u skupu svih skupu svih pozitivnih primjera (TP+FN). Ova se mjera naziva "odziv" jer nam govori koliko se pozitivnih primjera "odazvalo" klasifikatoru. Idealno, R = 1, tj. sve pozitivne primjere klasifikator će označiti kao takve. Alternativni nazivi za odziv su **stopa stvarnih pozitiva** (engl. *true positive rate, TPR*) i **osjetljivost** (engl. *sensitivity*). 7

- **Ispadanje** (engl. *fall-out, false positive rate*):

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

Ispadanje (ili **stopa lažnog alarma**) je udio lažno pozitivnih primjera (FP) u skupu svih negativnih primjera (FP + TN). Idealno, FPR = 0, tj. klasifikator niti jedan negativni primjer neće lažno proglasiti pozitivnim.

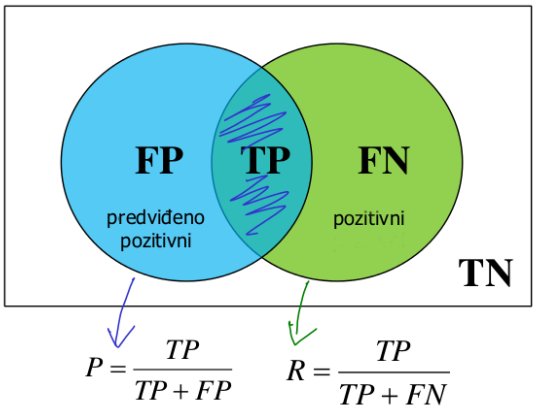
- **Specifičnost** (engl. *specificity, true negative rate*):

$$\mathrm{S} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

Specifičnost je udio stvarno negativnih primjera (TN) u skupu svih negativnih primjera (TN+FP). Idealno, S = 1, tj. klasifikator će sve negativne primjere klasificirati kao takve.

Premda se sve navedene mjere koriste u praksi, pojedina područja obično koriste samo neke od njih. Na primjer, u pretraživanju informacija, obradi prirodnog jezika i računalnom vidu uglavnom se koriste preciznost i odziv, dok se kod primjena strojnog učenja u medicini i biostatistici uobičajeno koriste odziv (tada pod nazivom "osjetljivost") i specifičnost.

Mi ćemo se u nastavku koncentrirati na preciznost (P) i odziv (R). Te su dvije mjere međusobno komplementarne. To možemo lakše shvatiti ako ih vizualiziramo Vennovim dijagramom, ovako:



Područje unutar pravokutnika odgovara svim primjerima iz označenog skupa primjera. Zeleni krug odgovara primjerima koji su pozitivni (primjeri za koje $y_{true} = 1$), dok plavi krug odgovara primjerima koje je klasifikator označio kao pozitivne (primjeri za koje $y_{pred} = 1$). Presjek tih dvaju krugova odgovara stvarno pozitivnim primjerima (TP). Dio zelenog kruga koji je izvan presjeka odgovara primjerima koji jesu pozitivni, ali nisu klasificirani kao pozitivni, dakle to

su lažno negativni primjeri (FN). Analogno, dio plavog kruga koji je izvan presjeka odgovara primjerima koji su klasificirani kao pozitivni, ali oni to zapravo nisu, dakle to su lažno pozitivni primjeri (FP). Područje izvan oba kruga odgovara primjerima koji niti su pozitivni niti su klasificirani kao pozitivni, dakle negativni su i klasificirani su kao negativni, tj. to su stvarno negativni primjeri (TN). Mjere preciznosti P i odziva R možemo sada definirati kao omjere površina na sljedeći način. Preciznost nam govori koliko je primjera koje je klasifikator proglasio pozitivnima stvarno pozitivno. To je udio stvarno pozitivnih primjera u skupu pozitivno klasificiranih primjera, što odgovara omjeru površine presjeka krugova i površine plavog kruga. S druge strane, odziv nam govori koliko pozitivnih primjera je klasifkator stvarno proglasio pozitivnima. To je udio stvarno pozitivnih primjera u skupu pozitivnih primjera, što odgovara omjeru površine presjeka krugova i površine zelenog kruga. Idealna situacija jest ona kad je plavi krug savršeno preklopljen sa zelenim krugom: tada su svi pozitivno klasificirani primjeri stvarno pozitivni ($P = 1$) i, obrnuto, svi pozitivni primjeri su također klasificirani kao pozitivni ($R = 1$).

Vratimo se na naš prethodni primjer. Tamo je $P = \frac{6}{6+12} = 33.33\%$, a $R = \frac{6}{6+2} = 75\%$. To bi značilo da naš klasifikator nije baš precizan, ali možemo reći da ima relativno pristojan odziv. Obje mjere daju znatno manju vrijednost od mjere točnosti, koja je, zahvaljujući velikom broju negativnih primjera, iznosila 90.7%. Dakle, P i R nam daju realističniju procjenu točnosti klasifikatora (točnosti u širem smislu) od točnosti (u užem smislu).

Pogledajmo još dva primjera.

▸ **PRIMJER**

Od $N = 1000$ primjera, njih 100 je pozitivno. Klasifikator ispravno klasificira 90 pozitivnih i 650 negativnih primjera. Izračunajmo Acc, P i R.

Iz navedenog zaključujemo da je matrica zabune sljedeća:

$$\begin{array}{c} \\ y_{pred} = 1 \\ y_{pred} = 0 \end{array} \begin{array}{c} \begin{array}{cc} y_{true} = 1 & y_{true} = 0 \end{array} \\ \begin{pmatrix} 90 & 250 \\ 10 & 650 \end{pmatrix} \end{array}$$

Iz ovoga slijedi:

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} = \frac{90 + 650}{1000} = 0.74$$
$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = \frac{90}{90 + 250} = 0.265$$
$$R = \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{FN}} = \frac{90}{90 + 10} = 0.9$$

Preciznost je ovdje mnogo lošija od odziva jer postoji mnogo više lažno pozitivnih primjera nego lažno negativnih primjera. Drugim riječima, ovaj je klasifikator dobar u nalaženju pozitivnih primjera, ali loš u izbjegavanju lažno pozitivnih primjera.

▸ **PRIMJER**

Od $N = 1000$ primjera, njih 100 je pozitivno. Klasifikator ispravno klasificira 50 pozitivnih i 850 negativnih primjera. Izračunajmo Acc, P i R.

Iz navedenog možemo zaključiti da matrica zabune izgleda ovako:

$$\begin{array}{c} \\ y_{pred} = 1 \\ y_{pred} = 0 \end{array} \begin{array}{c} \begin{array}{cc} y_{true} = 1 & y_{true} = 0 \end{array} \\ \begin{pmatrix} 50 & 50 \\ 50 & 850 \end{pmatrix} \end{array}$$

Iz ovoga slijedi:

$$Acc = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{50 + 850}{1000} = 0.9$$
$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{50}{50 + 50} = 0.5$$
$$R = \frac{\text{TP}}{\text{FP} + \text{FN}} = \frac{50}{50 + 50} = 0.5$$

Primijetimo da su preciznost i odziv ovdje jednaki, budući da je broj lažno pozitivnih primjera jednak broju lažno negativnih primjera.

Primijetite da su i preciznost i odaziv definirane s brojem stvarno pozitivnih primjera (TP) u brojniku i da su neovisne o broju stvarno negativnih primjera (TN). Možemo reći da su obje mjere fokusirane na pozitivne primjere. Posljedično, ako klasifikator ne uspije baš niti jedan pozitivan primjer označiti kao pozitivan, onda u brojniku imamo TP = 0 te će i preciznost i odziv biti jednaki nuli. Poseban slučaj takvog slučaja jest klasifikator koji sve primjere označi kao negativne: u tom slučaju TP = 0 i TP + FP = 0, odziv je nula no preciznost je nedefinirana. Suprotan (i manje realan) slučaj bio bi klasifikator koji sve negativne i samo negativne primjere proglasi pozitivnima: tada TP = 0 i TP + FN, preciznost je nula no odziv je nedefiniran. U ovakvim slučajevima, kada su ili preciznost ili odziv nedefinirani, uobičajeno je (premda to nije pravilo) postaviti ih na nulu.

1.3 Mjera F_1

Preciznost i odziv daju nam različitu informaciju. U praksi su ove dvije mjere često izravno suprotstavljene: ako klasifikator oblikujemo tako da ima visok odziv, onda je tipično da ćemo to platiti s nešto nižom preciznošću, i obrnuto, klasifikator s visokom preciznošću obično će imati niži odziv. Zbog toga, kada izvještavamo o rezultatu klasifikatora, svakako trebamo navesti vrijednosti obje mjere. Međutim, često nam je potrebno točnost klasifikatora (u širem smislu) karakterizirati samo jednim brojem. Jedna mjera vrednovanja koja radi upravo to jest **mjere** F_1 (engl. F_1 *score*). Mjera F_1 definirana je kao **harmonijska sredina preciznosti i odziva**. [8] Konkretno:

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R}$$

Vrijednost mjere F_1 je iz intervala $[0, 1]$, gdje više znači bolje. U našem prvom primjeru, gdje $P = 0.33$ i $R = 0.75$, imamo $F = \frac{2}{1/0.33+1/0.75} = 0.458$. U drugom primjeru, gdje $P = R = 0.5$, imamo $F = \frac{2}{1/0.5+1/0.5} = 0.5$. Iz ovoga vidimo da, ako su preciznost i odziv različiti, mjera F_1 bit će bliža od te dvije manjoj vrijednosti. Ako su pak preciznost i odziv jednaki, onda će i mjera F_1 biti njima jednaka.

Ljubopitljivi među vama zapitat će se zašto baš harmonijska sredina. Što fali običnoj artimetičkoj sredini? Odgovor je da, zato što su preciznost i odziv različite stvari, aritmetička sredina ovdje nema smisla. Naime, pogledamo li definicije preciznosti i odziva, vidimo da je u obje mjere brojnik identičan (TP), međutim nazivnik je različit. To zapravo znači da su skale ovih dviju mjera različite. Da bi skale bile iste, nazivnik mora biti isti, a to će biti ako P i R zamijenimo sa $1/P$ i $1/R$. Upravo to radi harmonijska sredina: harmonijska sredina je recipročna vrijednost aritmetičke sredine recipročnih vrijednosti. Dakle, koristimo harmonijsku sredinu kako bismo preciznost i odziv kombinirali na istoj skali. Obična aritmetička sredina ovdje jednostavno nema smisla jer ne možemo usrednjavati kruške i jabuke. [9]

Zanimljivo svojstvo harmonijske sredine jest da je "stroža" od aritmetičke sredine. Pritom mislimo da će, za različite P i R, harmonijska sredina biti manja od aritmetičke. Npr., za $P = 0.1$ i $R = 0.8$, mjera F_1 iznosi 0.178, dok bi aritmetička sredina dala 0.45. U ovakvoj

situaciji točnost od 0.45 doima se previsoka. Kod vrednovanja klasifikatora bolje je biti što stroži, i to upravo ostvarujemo mjerom F_1.

Rubni slučajevi mjere F_1 su oni kada je $P = 0$ ili $R = 0$ ili kada su P ili R nedefinirani. Obično se tada vrijednost mjere F_1 postavlja na nulu. Npr., za klasifikator koji sve primjere označi kao negativne P je nedefiniran a R je nula, pa se F_1 postavlja na nulu. 10

Pogledajmo još jedan primjer.

▸ **PRIMJER**

Od $N = 1000$ primjera, njih 100 je pozitivno. Klasifikator primjere klasificira potpuno nasumično, s jednakom vjerojatnošću za obje oznake. (Takav klasifikator nije od pretjerane koristi, ali može poslužiti za usporedbu s drugim klasifikatorima; o tome više kasnije.) Izračunajmo **očekivane** vrijednosti za Acc i mjeru F_1.

Budući da klasifikator primjere označava nasumično s jednakom vjerojatnošću za obje oznake, očekivano je da će njih 500 označiti kao pozitivne a njih 500 kao negativne. To znači da $\mathrm{TP} + \mathrm{FP} = \mathrm{FN} + \mathrm{TN} = 500$. Također znamo da je $\mathrm{TP} + \mathrm{FN} = 100$ i $\mathrm{FP} + \mathrm{TN} = 900$. Iz toga možemo zaključiti da **očekivana matrica zabune** izgleda ovako:

$$\begin{array}{c} \\ y_{pred} = 1 \\ y_{pred} = 0 \end{array} \begin{array}{c} \begin{array}{cc} y_{true} = 1 & y_{true} = 0 \end{array} \\ \begin{pmatrix} 50 & 450 \\ 50 & 450 \end{pmatrix} \end{array}$$

Iz ovoga sada možemo izračunati očekivane vrijednosti mjera vrednovanja:

$$\begin{aligned} Acc &= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} = \frac{50 + 450}{1000} = 0.5 \\ P &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = \frac{50}{500} = 0.1 \\ R &= \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{FN}} = \frac{50}{100} = 0.5 \\ F_1 &= \frac{2PR}{P + R} = \frac{2 \cdot 0.1 \cdot 0.5}{0.1 + 0.5} = 0.167 \end{aligned}$$

Očekivana točnost nasumičnog klasifikatora s jednakom vjerojatnošću za obje oznake je $Acc = 0.5$ i jednaka je očekivanom odzivu $R = 0.5$. (Uvjerite se da bi to bilo tako neovisno o udjelu pozitivnih primjera u skupu primjera). Međutim, očekivana preciznost je samo $P = 0.1$, pa je očekivana vrijednost mjere F_1 jednaka mizernih 0.167, što je znatno niže i od točnosti i odziva.

Na kraju primijetimo još da mjera F_1, računajući harmonijsku sredinu preciznosti i odziva, zapravo daje jednaku važnost i preciznosti i odzivu. Međutim, u nekim situacijama to nije ono što želimo. O tome smo već pričali kada smo govorili o asimetričnim funkcijama gubitka. Na primjer, ako nam je važnije imati manje lažno negativnih primjera nego lažno pozitivnih primjera (npr., kod dijagnostike malignih bolesti), onda to znači da nam je važnije imati velik odziv nego veliku preciznost. U takvim se situacijama koristi poopćenje mjere F_1, mjera F_β, koja može dati veći naglasak na preciznost ili odziv, ovisno o parametru β. Zapravo, mjera F_1 je poseban slučaj mjere F_β gdje je $\beta = 1$, što znači da preciznost i odziv imaju jednaku važnost. 11

Sada znamo kako računati uobičajene mjere za vrednovanje binarnog klasifikatora. Pogledajmo kako to proširiti na slučaj višeklasne klasifikacije.

2 Višeklasna klasifikacija

Kod binarne klasifikacije ($K = 2$), matrica zabune dimenzija je 2×2. Kod višeklasne klasifikacije ($K > 2$), matrica zabune dimenzija je $K \times K$. Na primjer, za klasifikaciju $N = 13$ primjera u

$K = 3$ klase, gdje za oznake vrijedi $\mathcal{Y} = \{1, 2, 3\}$, matrica zabune mogla bi izgledati ovako:

$$\begin{array}{c} \\ y_{pred} = 1 \\ y_{\text{pred}} = 2 \\ y_{pred} = 3 \end{array} \begin{array}{c} \begin{array}{ccc} y_{true} = 1 & y_{true} = 2 & y_{true} = 3 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 2 & 2 & 3 \\ 0 & 0 & 4 \end{pmatrix} \end{array}$$

Mjera vrednovanja koju odmah možemo izračunati iz ove matrice jest točnost (Acc). Točnost smo definirali kao udio točno klasificiranih primjera u ukupnom broju primjera, pa ćemo ju izračunati tako da zbroj elemenata na dijagonali matrice (trag matrice) podijelimo s ukupnim brojem primjera (zbrojem svih elemenata u matrici). U ovom našem primjeru, dobivamo $Acc = \frac{1+2+4}{13} = 0.538$.

2.1 Konfuzijska matrica 2×2 iz matrice $K \times K$

Ovo je jednostavno, no već znamo da je problem s točnošću taj da će točnost biti visoka za trivijalne klasifikatore koji sve primjere klasificiraju u većinsku klasu. Upravo smo zato uveli alternativne mjere, poput preciznosti, odziva i mjere F_1. No, problem s tim mjerama jest da njih možemo primijeniti samo na binarne klasifikatore, odnosno samo na matrice zabune dimenzija 2×2, gdje postoji jedna klasa koju smatramo pozitivnom i jedna klasa koju smatramo negativnom. Naime, samo kod takvih matrica možemo govoriti o stvarno pozitivnim (TP), lažno pozitivnim (FP), lažno negativnim (FN) i stvarno pozitivnim (TP) primjerima. Kod matrice $K \times K$ to nema smisla, jer što je pozitivno a što negativno ovisi o tome o kojoj klasi pričamo.

Međutim, ono što ima smisla jest fiksirati jednu klasu i nju tretirati kao pozitivnu, pa onda napraviti podjelu primjera na stvarno pozitivne, lažno pozitivne, lažno negativne i stvarno negativne u odnosu na tu odabranu klasu. Ilustrirajmo to na gornjem primjeru za klasu s oznakom $y = 2$. Za tu klasu, stvarno pozitivni primjeri su oni primjeri za koje $y_{\text{pred}} = 2$ i $y_{\text{true}} = 2$, a to u gornjoj matrici odgovara elementu [2,2]. Lažno pozitivni primjeri su oni za koje $y_{\text{pred}} = 2$ i $y_{\text{true}} \neq 2$, a to su elementi [2,1] i [2,3] u gornjoj matrici. Slično, lažno negativni primjeri su oni za koje $y_{\text{true}} = 2$ i $y_{\text{pred}} \neq 2$, a to su elementi [1,2] i [3,2] u gornjoj matrici. Konačno, stvarno negativni primjeri su svi oni za koje $y_{\text{pred}} \neq 2$ i $y_{\text{true}} \neq 2$, a to su elementi [1,1], [1,3], [3,1] i [3,3] u gornjoj matrici. Zbrajanjem odgovarajućih elemenata matrice dobivamo da je, za klasu $y = 2$, broj stvarno pozitivnih primjera jednak 2, broj lažno pozitivnih primjera jednak $2 + 3 = 5$, broj lažno negativnih primjera jednak $1 + 0 = 1$ te broj stvarno negativnih primjera jednak $1 + 0 + 0 + 4 = 5$. Iz ovoga vidimo da, ako fiksiramo jednu klasu i primjere iz te klase promatramo kao pozitivne primjere, a primjere iz svih ostalih kasa kao negativne, dobivamo zapravo matricu zabune 2×2 za dotičnu klasu. Konkretno, za klasu $y = 2$, dobivamo matricu:

$$\begin{array}{c} \\ y_{pred} = 2 \\ y_{pred} \neq 2 \end{array} \begin{array}{c} \begin{array}{cc} y_{true} = 2 & y_{true} \neq 2 \end{array} \\ \begin{pmatrix} 2 & 5 \\ 1 & 5 \end{pmatrix} \end{array}$$

Na isti način možemo izvesti matrice zabune za klase $y = 1$ i $y = 3$.

Ideja je, dakle, da matricu zabune višeklasnog klasifikatora dimenzija $K \times K$ dekomponiramo u K matrica zabune binarnog klasifikatora dimenzija 2×2. Formalno, za matricu zabune C dimenzija $K \times K$ i klasu s oznakom $y = j$ definiramo matricu zabune binarne klasifikacije:

$$\begin{array}{c} \\ y_{pred} = j \\ y_{pred} \neq j \end{array} \begin{array}{c} \begin{array}{cc} y_{true} = j & y_{true} \neq j \end{array} \\ \begin{pmatrix} \text{TP}_j & \text{FP}_j \\ \text{FN}_j & \text{TN}_j \end{pmatrix} \end{array}$$

gdje

- $\text{TP}_j = C[j, j]$ (j-ti element dijagonale matrice zabune)

- $\mathrm{FP}_j = \sum_{i:i\neq j} C[j,i]$ (zbroj elemenata j-tog retka izvan dijagonale)
- $\mathrm{FN}_j = \sum_{i:i\neq j} C[i,j]$ (zbroj elemenata j-tog stupca izvan dijagonale)
- $\mathrm{TN}_j = N - \mathrm{TP}_j - \mathrm{FP}_j - \mathrm{FN}_j$ (zbroj elemenata izvan retka j i stupca j)

Ako ste se u svemu ovome pogubili, sljedeći primjer mogao bi pomoći.

▸ **PRIMJER**

Skup od $N = 15$ primjera klasificiramo u $K = 3$ klase. Oznake klasa neka su $\mathcal{Y} = \{1, 2, 3\}$. Predikcije klasifikatora i stvarne oznake primjera su sljedeće:

$$\mathbf{y}_{pred} = (2,2,2,2,2,2,1,3,1,2,2,1,3,2,2)^{\mathrm{T}}$$
$$\mathbf{y}_{true} = (1,2,2,2,3,1,1,1,2,2,3,3,3,2,2)^{\mathrm{T}}$$

Konstruirajmo najprije višeklasnu matricu zabune dimenzija 3×3:

$$\begin{array}{c} \\ y_{pred}=1 \\ y_{pred}=2 \\ y_{pred}=3 \end{array} \begin{array}{c} \begin{array}{ccc} y_{true}=1 & y_{true}=2 & y_{true}=3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 2 & 6 & 2 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$

Iz ovoga sada na gore opisani način izvodimo tri matrice zabune za binarnu klasifikaciju, po jednu za svaku klasu:

$$\begin{pmatrix} \mathrm{TP}_j & \mathrm{FP}_j \\ \mathrm{FN}_j & \mathrm{TN}_j \end{pmatrix} \quad \Rightarrow \quad \begin{array}{ccc} y=1 & y=2 & y=3 \\ \begin{pmatrix} 1 & 2 \\ 3 & 9 \end{pmatrix} & \begin{pmatrix} 6 & 4 \\ 1 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 1 \\ 3 & 10 \end{pmatrix} \end{array}$$

2.2 Mikro- i makro-uprosječivanje

Sada znamo kako matricu zabune dimenzija $K \times K$ dekomponirati u K matrica zabune dimenzije 2×2. No, što dalje? Imamo K matrica zabune, a ne samo jednu, pa nije odmah jasno kako primijeniti ranije opisane mjere vrednovanja koje smo definirali nad samo jednom matricom zabune dimenzija 2×2. Tu imamo dvije mogućnosti. 12

- Prva mogućnost je da izračunamo željenu mjeru vrednovanja zasebno na svakoj od K matrica zabune, i zatim jednostavno izračunamo prosjek. To je takozvano **makro-uprosječivanje**. Definirajmo to formalno. Neka je m neka odabrana mjera vrednovanja (npr., P, R ili F_1). Onda **makro-mjeru** m (ili makro-m), što označavamo sa m^M, definiramo kao:

$$m^M = \frac{1}{K} \sum_j m_j$$

  gdje je K ukupan broj klasa, a m_j je mjera izračunata na matrici zabune 2×2 za klasu $y = j$.

- Druga mogućnost je da najprije pozbrajamo svih K binarnih matrica zabuna te da zatim na takvoj združenoj matrici zabune izračunamo željenu mjeru vrednovanja. To je takozvano **mikro-uprosječivanje**. Dakle, združena matrica zabune je:

$$\begin{pmatrix} \sum_j \mathrm{TP}_j & \sum_j \mathrm{FP}_j \\ \sum_j \mathrm{FN}_j & \sum_j \mathrm{TN}_j \end{pmatrix}$$

  i nad takvom matricom računamo neku odabranu mjeru vrednovanja m, kao da se radi o običnoj binarnoj klasifikaciji. To je onda **mikro-mjera** m (ili mikro-m), što označavamo sa m^μ.

▸ **PRIMJER**

Nastavimo s prethodnim primjerom. Matrice zabune 2×2 za svaku od triju klasa bile su:

$$\begin{array}{ccc} y=1 & y=2 & y=3 \\ \begin{pmatrix} 1 & 2 \\ 3 & 9 \end{pmatrix} & \begin{pmatrix} 6 & 4 \\ 1 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 1 \\ 3 & 10 \end{pmatrix} \end{array}$$

Pogledajmo prvo makro-mjere, i to makro-preciznost, makro-odziv i makro-F_1. Najprije računamo ove mjere za svaku klasu zasebno:

$$\begin{array}{lll} P_1 = 0.333 & R_1 = 0.25 & F_{1,1} = 0.286 \\ P_2 = 0.6 & R_2 = 0.857 & F_{1,2} = 0.706 \\ P_3 = 0.5 & R_3 = 0.25 & F_{1,3} = 0.333 \end{array}$$

a zatim uprosječujemo kroz klase:

$$P^M = 0.478 \qquad R^M = 0.452 \qquad F_1^M = 0.442$$

Za izračun mikro-mjera, najprije izračunavamo združenu matricu zabune zbrajanjem triju matrica zabune za svaku klasu:

$$\begin{pmatrix} 1 & 2 \\ 3 & 9 \end{pmatrix} + \begin{pmatrix} 6 & 4 \\ 1 & 4 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 3 & 10 \end{pmatrix} = \begin{pmatrix} 8 & 7 \\ 7 & 23 \end{pmatrix}$$

Iz te matrice sada izravno računamo mikro-mjere:

$$P^\mu = 0.533 \qquad R^\mu = 0.533 \qquad F_1^\mu = 0.533$$

Primijetimo još i da je točnost izračunata izravno na matrici $K \times K$ jednaka $Acc = \frac{8}{15} = 0.533$.

U prethodnom smo primjeru dobili $P^\mu = R^\mu = F_1^\mu = Acc$. Lako je pokazati da ova jednakost uvijek vrijedi. Zbog toga nema smisla govoriti posebno o mikro-preciznosti ili mikro-odzivu; umjesto toga, uobičajeno se govori ili o točnosti ili o mikro-F_1 (ovo drugo pogotovo ako se uspoređuje s makro-F_1). 13 14

Sažmimo što smo do sada opisali. Kod višeklasne klasifikacije u K klasa, matrica zabune dimenzija je $K \times K$. Nad tom matricom možemo izravno izračunati mjeru točnosti. Alternativa su mjere dobivene mikro- ili makro-uprosječivanjem, odnosno **makro-preciznost, makro-odziv, makro-F_1 i mikro-F_1**, gdje je ova potonja zapravo jednaka **točnosti**. Izbora, dakle, ne nedostaje, pa se postavlja pitanje koju mjeru vrednovanja koristiti. Sve se ove mjere koriste u praksi, i koju ćemo mjeru koristiti ovisi o tome na što želimo staviti naglasak. Ako nam je važno da naš klasifikator ne producira mnogo lažno pozitivnh primjera (npr., kod klasifikacije neželjene pošte), koristit ćemo mjeru preciznosti. Ako nam je pak važno da klasifikator ne producira mnogo lažno negativnih primjera, odnosno da identificira sve primjere neke klase (npr., kod dijagnostike malignih bolesti), onda ćemo koristiti mjeru odziva. Ako nam je oboje jednako važno, koristit ćemo mjeru F_1. Nadalje, budući da makro-mjere pri uprosječivanju daju jednaku težinu svim klasama, makro-mjere ćemo koristiti upravo u situaciji kada sve klase želimo vrednovati jednako neovisno o broju primjera koje sadrže. Suprotno tomu, kod izračuna mikro-mjere klase s većim brojem primjera imat će veći utjecaj na vrijednost mjere, pa ćemo te mjere koristiti ako klasifikaciju svakog primjera želimo vrednovati jednako, neovisno o tome iz koje klase primjer dolazi. Općenito, zato što klasifikatori na manjim klasama ubičajeno rade lošije, možemo očekivati da će makro-mjere imati manju vrijednost od mikro-mjera, dakle u praksi je uobičajeno da vrijedi $F_1^M < F_1^\mu$. Konačno, primijetimo da, umjesto izračuna mikro- ili makro-prosjeka, čime se neminovno gubi dio informacije, ponekad ima više smisla izvijestiti o točnosti, preciznosti, odzivu ili mjeri F_1 svake klase pojedinačno. 15

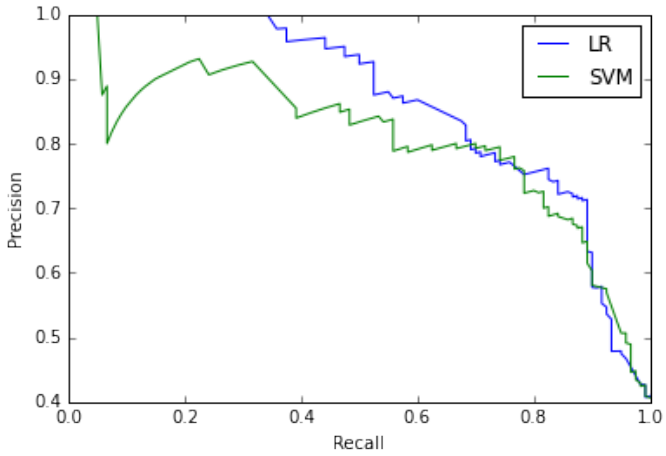
3 Vrednovanje klasifikatora s pragom

Do sada smo se bavili vrednovanjem klasifikatora koji na izlazu daje oznaku klase. Međutim, mnogi klasifikatori (npr., logistička regresija, naivan Bayesov klasifikator) na izlazu daju **vjerojatnost klasifikacije** primjera u neku klasu. Čak i onda kada to nije slučaj (npr., SVM), vjerojatnost klasifikacije može se dobiti nekom metodom kalibracije klasifikatora (npr., Plattova metoda, o kojoj smo pričali u 9. predavanju).

Standardno, klasifikacijski prag je 0.5, tj. primjer $\mathbf{x}$ klasificira se u pozitivnu klasu $y = 1$ ako je $P(y = 1|\mathbf{x}) \geqslant 0.5$. Međutim, u nekim je situacijama možda bolje prag postaviti na vrijednost višu ili nižu od 0.5. Što ćemo dobiti ako prag postavimo vrijednost višu od 0.5, npr. na 0.8? U tom slučaju, samo primjere za koje je vrlo vjerojatno da su pozitivni bit će klasificirani u $y = 1$, u protivnom će biti klasificirani u klasu $y = 0$. Kakav je utjecaj toga na klasifikacijske pogreške? Ako klasifikator samo primjere za koje je vrlo siguran klasificira u klasu $y = 1$, onda će producirati manje lažno pozitivnih primjera, a to znači da će preciznost porasti. Suprotno, ako prag smanjimo na vrijednost manju od 0.5, klasifikator će u klasu $y = 1$ klasificirati i primjere s manjom vjerojatnošću pripadanja toj klasi, što znači da potencijalno smanjujemo broj lažno negativnih primjera, odnosno povećavamo odziv. Dakle, ugađajem klasifikacijskog praga izravno utječemo na preciznost i odziv klasifikatora. 16

3.1 Krivulja preciznost-odziv

Ako želimo vrednovati klasifikator s obzirom na sve moguće vrijednosti klasifikacijskog praga, možemo skicirati **krivulju preciznost-odziv** (engl. *precision-recall curve, PR curve*). Krivulju dobivamo tako da, za već trenirani model, vrijednost praga postepeno smanjujemo od 1 do 0 (u nekim fiksnim koracima) te za svaku vrijednost računamo odziv i preciznost klasifikatora. Zatim skiciramo dobivenu krivulju, s odzivom na x-osi i preciznošću na y-osi. Na primjer:



Na slici su prikazane dvije krivulje preciznost-odziv, jedna (plava) za model logističke regresije i druga (zelena) za model SVM. Obje su dobivene na skupu podataka Titanic. Kako smanjujemo klasifikacijski prag, tako odziv raste, ali preciznost pada, pa krivulja preciznost-odziv pada kako se pomičemo udesno (premda može biti kratkotrajnih porasta). Kod potpunog odziva ($R = 1$), preciznost mora pasti (nagradno pitanje: na koju vrijednost?), no želimo da se taj pad dogodi što kasnije, tj. na što višim razinama odziva. Drugim riječima, preferiramo krivulje koje su što bliže točki ($P = 1, R = 1$). Tako je na gornjoj slici krivulja logističke regresije bolja je od krivulje SVM-a, pa možemo zaključiti da logistička regresija na ovom skupu podataka radi bolje od SVM-a (iznenađujuće, ali događa se).

Zamislite da imamo dva klasifikatora sa svojim krivuljama preciznost-odziv. Q: Kako bi izgledala krivulja prvog klasifikatora, ako je on definitivno bolji od drugog klasifikatora? A:

Bila bi uvijek iznad krivulje drugog klasifikatora. Naime, ako je krivulja klasifikatora h_1 uvijek iznad krivulje klasifikatora h_2 (znači P klasifikatora h_1 je veća od P klasifikatora h_2 za svaku razinu odziva R), onda stvarno nema razloga da ikada upotrijebimo klasifikator h_2 jer on uvijek ima lošiji odziv ili lošiju preciznost (ili oboje) od klasifikatora h_1. Međutim, ako se krivulje preciznost-odziv križaju, onda to znači da niti jedan klasifikator nije bolji od drugoga. Na gornjoj slici imamo upravo takvu situaciju: preciznost logističke regresije nije na svim raznima odziva viša od preciznosti SVM-a (npr., za $R = 0.7$ i $R = 0.95$ preciznost SVM-a je nešto viša). U ovakvim slučajevima odabrat ćemo klasifikator koji je najbolji za željenu razinu odziva, ili možemo koristiti različite klasifikatore za različite razine odziva. Iz ovoga možemo zaključiti da krivulja preciznost-odziv zapravo definira parcijalni uređaj između klasifikatora po kriteriju preciznosti i odziva (neki su klasifikatori bolji od nekih drugih, dok neki nisu međusobno usporedivi). 17

Premda je krivulja preciznost-odziv vrlo informativna, nekad je točnost klasifikatora potrebno kvantificirati jednim brojem. To možemo načiniti tako da izračunamo **prosječnu preciznost** (engl. *average precision*). Prosječna preciznost zapravo je integral krivulje preciznost-odziv. Što je krivulja preciznost-odziv bliža točki $(P = 1, R = 1)$, to će prosječna preciznost biti veća.

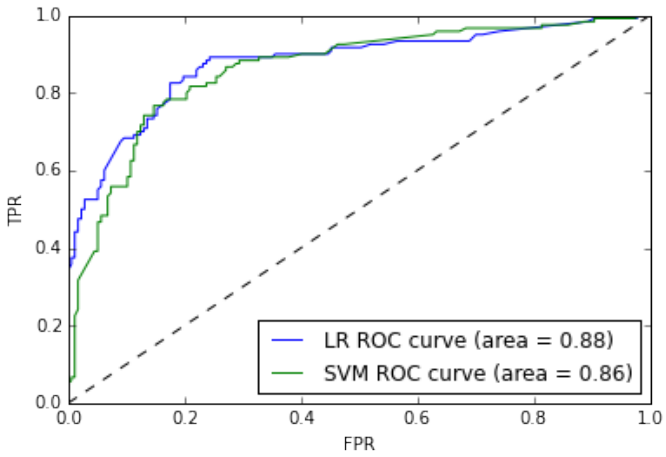
3.2 Krivulja ROC

Druga mjera vrednovanja koja uzima u obzir varijabilnost praga i koja se često koristi jest **površina pod krivuljom ROC** (engl. *area under ROC curve*), skraćeno **AUC**. No, pogledajmo najprije što je to krivulja ROC. Krivulja ROC (naziv dolazi od *receiver operating characteristics*, što nema previše veze s primjenom u strojnom učenju) je vrijednost **stope stvarnih pozitiva** (TPR), što je isto kao i **odziv**, kao funkcije **stope lažnog alarma** (FPR), odnosno **ispadanja** (engl. *fall-out*). Prisjetimo se, FPR je: 18

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

FPR mjeri koliki je udio primjera koje je klasifikator pogrešno proglasio pozitivnima. Idealno, FPR = 0. Mjera TPR, odnosno odziv, mjeri koliko je pozitivnih primjera klasifikator detektirao, i idealno je TPR = 1.

Kako smanjujemo prag, tako će FPR rasti, jer će klasifikator sve više primjera proglašavati pozitivnima, pa ćemo imati i sve više lažno pozitivnih primjera (FP). No, istovremeno će sa smanjivanjem praga rasti i odziv, jer će klasifikator detektirati sve više stvarno pozitivnih primjera (TP). Idealno bi bilo da odziv raste vrlo strmo, tako da FPR možemo zadržati na nekoj niskoj vrijednosti. Dakle, kako izgleda krivulja ROC? Ovako:



Ovo je krivulja ROC za logističku regresiju i SVM na istim podatcima kao i ranije (Titanic). Klasifikator je to bolji što njegova krivulja ROC prolazi bliže točki $(\text{FPR} = 0, \text{TPR} = 1)$. I

ovdje vidimo da je logistička regresija na ovom skupu podataka bolja od SVM-a, osim kod nekih razina odziva (istih onih kao i za krivulju preciznost-odziv).

Krivuja ROC zapravo izgleda vrlo slično krivulji preciznost-odziv (uzevši u obzir zrcaljenje), pa se možda pitate zašto nam trebaju i jedna i druga. Trebaju nam obje jer svaka ima svoje prednosti. Prednost krivulje preciznost-odziv je što je lako razumljiva: mnoga područja primjene strojnog učenja koriste preciznost i odziv kao standardne mjere vrednovanja. No, velika prednost krivulje ROC jest što ona za **nasumični klasifikator** (klasifikator koji primjere klasificira u nasumično odabrane klase) odgovara pravcu od $(0, 0)$ do $(1, 1)$. Ono što je posebno dobro jest da to vrijedi neovisno o tome je li broj pozitivnih i negativnih primjera uravnotežen. Zašto? (Meni to nije bilo odmah jasno, vama možda je.) Ako klasifikator slučajno odabere $k\%$ primjera i proglasi ih pozitivnima, onda će u tom skupu imati $k\%$ od ukupno pozitivnih primjera, dakle odziv (odnosno TPR) je $k\%$. Slično, u tom odabranom skupu bit će $k\%$ od ukupnog broja negativnih primjera, a te je primjere klasifikator proglasio pozitivnima, dakle FPR je također $k\%$. Drugim riječima, TRP = FPR, i imamo pravac, neovisno o distribuciji primjera u pozitivnu i negativnu klasu! Zašto nam je to korisno? Zato što je nasumični klasifikator dobra referentna točka za vrednovanje klasifikatora: ako naš klasifikator daje predikcije koje su tek neznatno bolje od nasumičnog klasifikatora (krivulja ROC je blizu pravca TRP = FPR) ili nedajbože radi lošije od nasumičnog klasifikatora (krivulja ROC je ispod tog pravca), onda je to jedan beskoristan klasifikator. 19

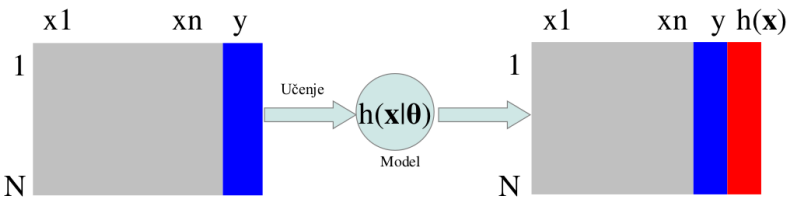
Mjera AUC jednostavno je definirana kao površina ispod krivulje ROC. AUC je u intervalu $[0, 1]$, što više, to bolje. Za nasumični klasifikator, AUC = 0.5.

4 Procjena pogreške modela

Do sada smo govorili o tome kako izmjeriti točnost (odnosno pogrešku) klasifikatora koristeći razne mjere vrednovanja izračunate nad matricom zabune. Sve mjere vrednovanja izračunavaju se na skupu primjera koji imamo. U statističkome smislu, to je jedan **slučajan uzorak**. Dakle, naše mjere su funkcije slučajnog uzorka, što ih i samima čini slučajnim varijablama. Kada primjenimo mjeru na uzorak primjera, ono što dobivamo zapravo je empirijska **procjena** te mjere. U tom smislu govorimo o **procjeni pogreške** (engl. *error estimation*) modela (ovdje "pogreška" može biti bilo koja mjera vrednovanja, ali se uobičajeno govori baš o pogrešci).

Postavlja se pitanje kako napraviti **dobru** procjenu pogreške modela. Dobra procjena znači da je procjenitelj **nepristran** (njegovo očekivanje jednako je pravoj vrijednosti populacije) i da je **konzistentan** (varijanca procjene se smanjuje kako veličina uzorka raste). To su statističke kvalitete procjenitelja koje bismo voljeli imati. Pored ovih kvaliteta, u strojnom učenju nam je bitno da je procjena pogreške **poštena**. Pod time mislimo da želimo mjeriti sposobnost generalizacije modela, dakle pogrešku modela na još neviđenim podatcima (uz standardnu pretpostavku da se ti podatci pokoravaju istoj distribuciji kao i podatci nad kojima smo učili model, tj. da je skup primjera za učenje reprezentativan uzorak problema koji rješavamo). U praksi to također znači da ćemo preferirati stvari postaviti tako da je naša procjena pogreške **pesimistična** prije nego optimistična. Naime, ako je procjena pogreške pesimistična, znamo da će u stvarnosti klasifikator raditi tako ili čak još bolje.

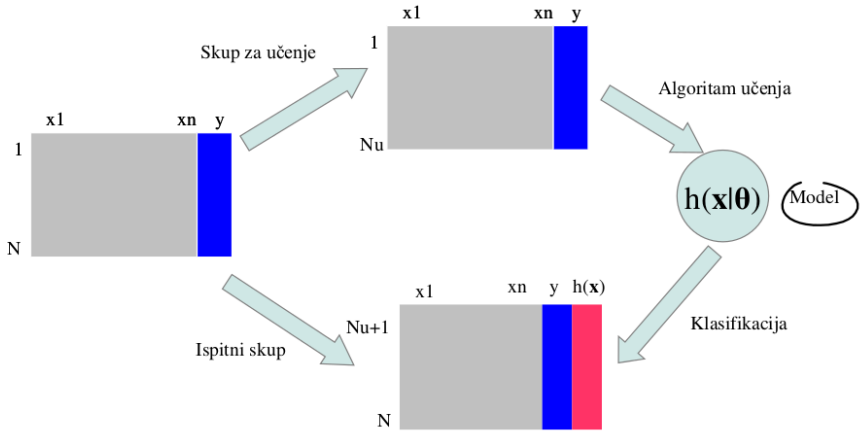
U statistici (i strojnom učenju) razvijen je niz postupaka za procjenu pogreške. Krenimo od loših (nepoštenih) procjena. Definitivno nepoštena procjena pogreške (odnosno točnosti, preciznosti, odziva, mjere F_1, AUC itd.) jest procjena na istom skupu primjera na kojem je klasifikator učen. Grafički to možemo prikazati ovako:



Ovdje smo dakle svih N primjera iskoristili za učenje modela, dobili smo naučeni model h, i zatim taj model primijenili na istih tih N primjera, usporedili oznake $\mathbf{y}_{pred}$ koje dobivamo modelom h sa stvarnim oznakama $\mathbf{y}_{true}$, izveli matricu zabune te na temelju nje izračunali neku mjeru vrednovanja. Ovo je naravno loše jer ne mjerimo pogrešku generalizacije već pogrešku učenja, koja je tipično manja od pogreške generalizacije te opada sa složenošću modela (to već znamo).

4.1 Metoda izdavajanja

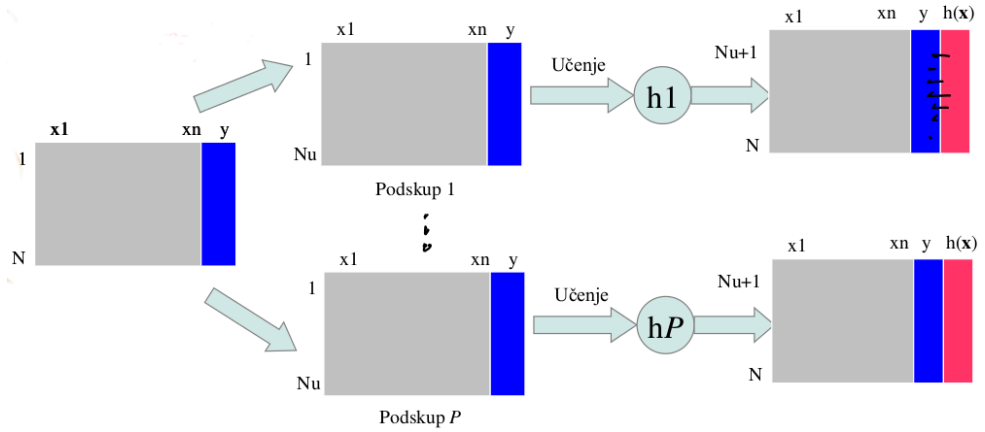
Alternativa je **metoda izdvajanja** (engl. *holdout method*), gdje skup primjera razdvajamo na **skup za učenje** (engl. *training set*) i **ispitni skup** (engl. *test set*):



Ovo je zapravo najjednostavnija varijanta nečega što smo već zvali **unakrsna provjera**. Prednost ove metode jest da doista procjenjujemo pogrešku generalizacije, a ne pogrešku učenja.

Međutim, nedostatak ovog pristupa je da doslovno bacamo primjere za učenje: budući da smo dio primjera morali ostaviti postrani za ispitivanje, imamo manje primjera za učenje, a to je uvijek loše jer gubimo vrijednu informaciju potrebnu za poboljšanje modela. To je pogotovo problematično ako je primjera i inače malo (npr., ako ukupno imamo 100-tinjak primjera). Drugi problem je loša točnost procjene: naša procjena pogreške temelji se na samo jednom uzorku, što znači da će procjena vrlo varirati ovisno o uzorku, odnosno o načinu kako smo primjere podijelili u dva skupa. Točnost procjene će doduše rasti što je ispitni skup veći, ali problem je što je skup primjera uvijek ograničen.

Oba nedostatka moguće je riješiti postupcima temeljenima na **ponovnom uzorkovanju** (engl. *resampling*). Ti postupci imaju dugu tradiciju u statistici. U strojnom učenju ti nam postupci omogućavaju da model učimo na većini raspoloživih primjera (pa dobivamo bolji model), a da istovremeno dobijemo i bolju procjenu pogreške. Najjednostavnija takva metoda je **ponovljeno izdvajanje** (engl. *repeated holdout*):



Ovdje jednostavno podjelu na skup za učenje i skup za ispitivanje radimo više (ukupno P) puta, svaki puta nanovo trenirano model na skupu za učenje i ispitujemo ga na skupu za ispitivanje. Procjena pogreške (ili koje već mjere) je srednja vrijednost pogreške na svim ispitnim skupovima.

4.2 k-struka unakrsna provjera

Ovo je lijepo, i čini se da rješava naš problem (sve primjere iskorištavamo, procjena je bolja jer je računamo kao srednju vrijednost više procjena), međutim problem je da nemamo nikakvu kontrolu koji su primjeri koliko puta upotrijebljeni. Metoda koja zadržava prednosti ponovljenog izdvajanja, a dodatno nam daje tu kontrolu, jest metoda k-**struke unakrsne provjere** (engl. *k-folded cross validation*):

Skup primjera dijelimo u k disjunktnih podskupova (tj. particija ili "preklopa"). Zatim učimo klasifikator na $(k-1)$ preklopa, a ispitujemo ga na k-tom preklopu, pa sve to ponavljamo ukupno k-puta, svaki puta koristeći drugi preklop kao ispitni skup. Procjenu pogreške (ili koje god mjere) izračunavamo kao srednju vrijednost pogreške na k preklopa. Tipično uzimamo $k = 10$ ili $k = 5$.

Ponekad se može dogoditi da je podjela skupa primjera na skup za učenje i skup za ispitivanje takva da ne zrcali pravu razdiobu primjera u skupu. To može rezultirati pretjerano pesimističnom procjenom. (Q: Zašto? A: Zato što to narušava našu pretpostavku da skup za učenje i skup za ispitivanje dolaze iz iste distribucije.) Rješenje u takvim situacijama jest da se skupovi **stratificiraju**, odnosno da se pobrinemo da razdioba klasa bude sačuvana u oba skupa. To možemo jednostavno ostvariti na sljedeći način:

1. Skup primjera podijelimo u K podskupova, po jedan za svaku klasu;
2. Svaki takav podskup podijelimo u k preklopa;
3. Združimo K preklopa (po jedan od svake klase) u jedan preklop, i ponovimo to k puta.

Što su prednosti, a što nedostatci k-struke unakrsne provjere? Prednost jest da je svaki primjer bio iskorišten za ispitivanje, i to točno jednom. Također, svaki je primjer bio iskorišten i za učenje (i to $k-1$ puta). Prednosti su također da ju je jednostavno provesti (i implementirana je u mnogim standardnim alatima) te da nije računalno suviše zahtjevna (osim ako k nije prevelik). Naravno, prednost je i što dobivamo točniju procjenu pogreške jer su ispitni skupovi nepreklapajući. Nedostaci, međutim, su to što k klasifikatora nije međusobno nezavisno. Naime, svaka dva klasifikatora dijele $k-2$ preklopa, tj. $(k-2)/k$ skupa za učenje. To, pogotovo ako je k visok, dovodi do visoke varijance procjene pogreške (jer postoje korelacije između k vrijednosti pogrešaka).

Ekstremni slučaj k-struke unakrsne provjere jest kada je broj preklopa jednak broju primjera, $k = N$. To znači da u svakoj od N iteracija unakrsne provjere klasifikator učimo na svim primjerima osim na jednom te klasifikator zatim ispitujemo na tom jednom primjeru. To se zove **unakrsna provjera "izdvoji jednog"** (engl. *Leave-One-Out Cross Validation, LOOCV*). Očita prednost metode LOOCV jest da iskorištavamo gotovo potpun skup primjera i dobivamo bolju procjenu pogreške. Nedostatak je da LOOCV može biti računalno vrlo zahtjevan, pogotovo za veliki N. Drugi problem je opet visoka varijanca procjene pogreške, budući da su klasifikatori učeni na skoro istim skupovima, pa dakle nisu više nezavisni. Treći problem jest da, kada ispitujemo na samo jednom primjeru, ne možemo izračunati sve one mjere koje smo uveli (jer su one definirane nad matricom zabune koju dobivamo iz skupa primjera), već samo pogrešku ili točnost na jednom primjeru. U načelu, LOOCV se prepruča koristiti kada je skup podataka srednje veličine. 21

Da sažmemo: za procjenu pogreške **najbolje je koristiti k-struku unakrsnu provjeru**. Ta metoda nije bez nedostataka, ali je među najboljima koje imamo. Ako je skup podataka vrlo velik, dovoljno je koristiti običnu unakrsnu provjeru (dakle, s $k = 1$). Ako je skup podataka srednje veličine, onda možemo koristiti LOOCV. Što je veliko, a što malo ovisi o problemu (za teži problem trebam nam više primjera za učenje, pa preferiramo manji k) i željenoj pouzdanosti procjene (željenom intervalu pouzdanosti procjene pogreške; o tome više drugi put).

4.3 Unakrsna provjera uz odabir modela

Do sada smo govorili o procjeni pogreške modela čije hiperparametre ne trebamo optimirati, dakle situaciji kada imamo jedan već odabrani modela i želimo vrednovati koliko on dobro generalizira. Međutim, stvarnost je rijetko tako idealna. U stvarnosti ćemo, pored treniranja i procjene pogreške modela trebati prije toga napraviti **odabir modela**, tj. optimizaciju hiperparametara iliti ugađanje složenosti modela. Do sada smo već usvojili ideju da, kod unakrsne provjere, skup primjera trebamo podijeliti na skup za učenje $\mathcal{D}_{\text{train}}$ i skup za ispitivanje $\mathcal{D}_{\text{test}}$, koji su međusobno disjunktni. No, ako trebamo napraviti odabir modela, onda unakrsnu provjeru moramo raditi nad tri međusobno disjunktna skupa:

1. $\mathcal{D}_{\text{train}}$ – **skup za učenje** (engl. *train(ing) set*), na kojemu treniramo model;
2. $\mathcal{D}_{\text{val}}$ – **skup za provjeru (validaciju)** (engl. *validation set*), na kojemu procjenjujemo pogrešku generalizacije modela kod optimizacije hiperparametara;
3. $\mathcal{D}_{\text{test}}$ – **ispitni skup** (engl. *test set*), na kojemu procjenjujemo pogrešku generalizacije modela s optimiziranim hiperparametrima (dakle, modela optimalne složenosti).

pri čemu su ovi skupovi međusobno disjunktni, $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{val}} = \mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \mathcal{D}_{\text{val}} \cap \mathcal{D}_{\text{test}} = \varnothing$.

Stvar funkcionira ovako: iz neke familije modela odaberemo jedan model $\mathcal{H}$ te ga treniramo na skupu $\mathrm{D}_{\text{train}}$. Nakon što smo naučili model, procjenjujemo njegovu pogrešku generalizacije na skupu D_{val}. Zatim odaberemo drugi model iz familije modela, te ponavljamo to isto. Zatim odaberemo neki treći model, te ponavljamo isto. I tako dalje, ponavljamo treniranje na $\mathcal{D}_{\text{train}}$ i ispitivanje na D_{val} sve dok ne pronađemo optimalan model $\mathcal{H}^*$ (optimalne hiperparametre)

na skupu D_{val}. Nakon što smo pronašli optimalan model $\mathcal{H}^*$, taj model treniramo na skupu $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}}$. Koristimo oba skupa kako bismo iskoristili što više podataka. Na kraju procijenimo pogrešku generalizacije tako naučenog modela $\mathcal{H}^*$ na ispitnome skupu $\mathcal{D}_{\text{test}}$, koji do sada uopće nismo koristili. Tako procijenjena ispitna pogreška je ono što objavljujemo. To je pogreška generalizacije modela optimalne složenosti na našem skupu primjera. Znamo da je model optimalne složenosti jer smo proveli optimizaciju hiperparametara. Također, znamo da je to pogreška generalizacije jer je dobivena na skupu primjera koji nije korišten niti za treniranje niti za odabir modela.

Zastanite ovdje na trenutak i zapitajte se razumijete li zašto smo uveli treći skup primjera ($\mathcal{D}_{\text{val}}$). Možda vam se čini da nam skup $\mathcal{D}_{\text{val}}$ uopće ne treba jer smo optimizaciju hiperparametara mogli napraviti na ispitnom skupu $\mathcal{D}_{\text{test}}$, i na tom istom skupu onda u konačnici procijeniti pogrešku optimalnog modela. No, razmislite što bi se dogodilo da za optimizaciju hiperparametara modela koristimo isti skup na kojem u konačnici procjenjujemo pogrešku generalizacije modela. Dobili bismo nepoštenu (pretjerano optimističnu) procjenu pogreške, jer bismo složenost modela namijestili upravo prema ispitnom skupu. Zbog toga skup primjera na kojem procjenjujemo pogrešku modela mora biti potpuno netaknut. To znači da ne smije biti upotrijebljen ni za kakve optimizacije: niti za optimizaciju parametara modela (treniranje modela) niti za optimizaciju hiperparametara modela (odabir modela optimalne složenosti).Također pogrešno bi bilo optimizaciju hiperparametara provesti na skupu za učenje $\mathcal{D}_{\text{train}}$. To je očito loša ideja jer na taj način odabir modela ne bismo radili prema pogrešci generalizacije nego pogrešci učenja, i sigurno bismo dobili presložen model. Dakle, kako god okrenemo, treba nam treći, validacijski skup, $\mathcal{D}_{\text{val}}$.

No, što ako želimo napraviti k-struku unakrsnu provjeru *zajedno* s odabirom modela? To je potpuno realističan scenarij: često trebamo napraviti i jedno i drugo. Tu se situacija onda dodatno komplicira. Pogledajmo to iduće.

4.4 Ugniježđena k-struka unakrsna provjera

Često želimo raditi i unakrsnu provjeru i odabir modela. To znači da trebamo prolaziti kroz k preklopa, ali isto tako moramo raditi na tri skupa podataka. U tom slučaju radit ćemo **ugniježđenu unakrsnu provjeru** (engl. *nested cross validation*). Ta metoda ima dvije ugniježđene petlje: **vanjsku petlju** za treniranje i ispitivanje modela (kao i ranije) te **unutarnju petlju** za odabir modela (za treniranje i provjeru). Postupak možemo opisati sljedećim pseudokodom:

▸ **Ugniježđena unakrsna provjera** $k \times l$

```
1:  podijeli D na vanjske preklope D_i, i = 1,...,k
2:  za i = 1,...,k radi:                                   vanjska petlja
3:    D_train ← D\D_i, D_test ← D_i
4:    za svaku odabranu vrijednost hiperparametra α radi:
5:      podijeli D_train na unutarnje preklope D_j, j = 1,...,l
6:      za j = 1,...,l radi:                               unutarnja petlja
7:        D_train' ← D_train\D_j, D_val ← D_j
8:        treniraj model na D_train' i procijeni pogrešku na D_val
9:      izračunaj prosjek mjere na l unutarnjih preklopa
10:   odaberi hiperparametar α koji minimizira procjenu pogreške
11:   nauči odabrani model na D_train i procijeni pogrešku na D_test
12: izračunaj prosjek mjere na k vanjskih preklopa
```

Pseudokod implementira dvije petlje: vanjska započinje na liniji 2 i odvija se k puta, a unutarnja petlja započinje na liniji 6 i odvija se l puta. To zovemo ugniježđenom unakrsnom

provjerom $k \times l$. Odabir vrijednosti za k i l je proizvoljan, a tipično se uzima 5×5, 10×5, 10×5, 5×10 ili slično. Na početku (linija 1) skup podataka dijelimo na k vanjskih preklopa. U svakoj iteraciji vanjske petlje, $k-1$ vanjskih preklopa uzimamo kao skup za učenje $\mathcal{D}_{\text{train}}$, a 1 preklop kao skup za testiranje $\mathcal{D}_{\text{test}}$. Na primjer, ako je $k = 5$, onda ćemo 4/5 skupa podataka koristiti za treniranje, a 1/5 za ispitivanje. Kroz iteracije ćemo se pomicati po preklopima: prvo ćemo model ispitivati na prvoj petini skupa podataka, zatim na drugoj petini, itd. U liniji 4 iteriramo kroz hiperparametre modela. Ako parametra nema mnogo, to se može ostvariti iscrpnom pretragom (npr., kroz rešetku $C \times \gamma$ kod SVM-a). Međutim, ako je parametra mnogo, ovdje nam treba neki pametniji, moguće heuristički način pretraživanja parametara. Nakon što smo odabrali jednu vrijednost hiperparametara $\boldsymbol{\alpha}$, u liniji 5 skup za učenje $\mathcal{D}_{\text{train}}$ dalje dijelimo na l unutarnjih preklopa. Od toga ćemo $l-1$ preklopa koristiti za učenje modela u unutarnjoj petlji, a 1 preklop za provjeru modela. Na primjer, ako je $l = 10$, onda ćemo uzeti 9 unutarnjih preklopa za učenje (skup $\mathcal{D}_{\text{train}'}$) te 1 unutarnji preklop za provjeru (skup $\mathcal{D}_{\text{val}}$). Primijetite da, budući da ovo sve radimo na 4/5 skupa, zapravo ćemo model trenirati na $4/5 \cdot 9/10$ cjelokupnog skupa primjera, a provjeravati na $4/5 \cdot 1/10$ cjelokupnog skupa primjera. U liniji 6 iteriramo kroz sve unutarnje preklope, rotirajući skup za provjeru l puta. U liniji 8 model učimo na skupu za učenje $\mathcal{D}_{\text{train}'}$, koji je veličine $\frac{k-1}{k} \cdot \frac{l-1}{l}$, a ispitujemo na skupu za provjeru, koji je veličine $\frac{k-1}{k} \cdot \frac{1}{l}$. Nakon što smo to izvrtili l puta, imat ćemo l procjena mjere vrednovanja na $\mathcal{D}_{\text{val}}$. U liniji 9 zatim računamo prosjek te mjere. Time smo dobili procjenu pogreške generalizacije (ili koje god mjere) za model s hiperparametrima $\boldsymbol{\alpha}$. Sada sve ovo ponavljamo za neki drugi odabir vrijednosti hiperparametra $\boldsymbol{\alpha}$, dakle iteriramo opet od linije 4, sve dok ne ispitamo sve hiperparametre koje smo željeli ispitati. Nakon toga, u liniji 10, raspolažemo procjenama pogrešaka za sve modele koje smo isprobali, te odabiremo onaj model (one hiperparametre $\boldsymbol{\alpha}$) koji minimizira procjenu pogreške (ili maksimiziraj neku drugu mjeru vrednovanja). Time smo završili s odabirom modela. Sada taj model učimo na skupu $\mathcal{D}_{\text{train}}$ iz vanjske petlje (skupu koji smo imali prije podjele na unutarnje preklope) i njegovu pogrešku procijenjujemo na $\mathcal{D}_{\text{test}}$. Sve ovo ćemo ponoviti k puta, počevši od linije 1. Kada završimo, na liniji 12, imat ćemo k procjena pogreške za optimalne modele, gdje su optimalni modeli odabrani na temelju l procjena pogreške u unutarnjoj petlji. U konačnici nam ostaje samo izračunati srednju vrijednost k procjena pogrešaka u vanjskoj petlji. 22

Sažetak

- **Matrica zabune** sažeto prikazuje broj ispravnih i neispravnih klasifikacija
- **Točnost** je najjednostavnija mjera vrednovanja klasifikatora, ali je zavaravajuća ako su klase neuravnotežene
- Pored točnosti, nad binarnom matricom zabune možemo definirati niz mjera vrednovanja, uključivo **preciznost** i **odziv**, koje bolje vrednuju klasifikator kod neuravnoteženih klasa
- **Mjera F1** kombinira preciznost i odziv pomoću njihove harmonijske sredine
- Višeklasne klasifikatore tipično vrednujemo pomoću **makro-preciznosti**, **makro-odziva**, **makro-F1** te **točnosti**
- Klasifikatore s pragom vrednujemo pomoću **krivulje preciznost-odziv** i **ROC krivulje**
- Procjenu generalizacijske pogreške/točnosti klasifikatora treba se napraviti na **odvojenom skupu podataka**
- **Unakrsna provjera uz odabir modela** skup podataka dijeli na tri podskupa: skup za učenje, skup za provjeru i skup za ispitivanje
- k-**struka unakrsna provjera** daje bolju procjenu pogreške od metode izdvajanja, a k-**struka ugniježđena unakrsna provjera** kombinira procjenu pogreške i odabir modela

Bilješke

[1] Lijep pregled problematike vrednovanja algoritama strojnog učenja možete naći u (Raschka, 2018) (zahvaljujem Filipu Karlu Došiloviću na ovoj informaciji). Nešto starija, no mnogo detaljnija referenca je (Japkowicz and Shah, 2011). Očekivano, oba ova izvora idu u mnogo više detalja nego što ćemo mi obuhvatiti ovim predavanjem, pa se zainteresirani pojedinci za produbljivanje znanja upućuju na dotične izvore.

[2] Vrednovanje grupiranja malo smo dotakli u pretprošlom predavanju, kada smo pričali o mjerama točnosti grupiranja u kontekstu odabira optimalnog broja grupa (Randov indeks, metoda siluete). Osnovna referenca za ocvu problematiku je (Hubert and Arabie, 1985). Dobar pregled evaluacijskih metrika za regresiju možete naći u dokumentaciji za `scikit-learn`.

[3] U nastavku koristim naziv **mjera vrednovanja**. U engleskom se tipično koristi **evaluacijska metrika** (engl. *evaluation metric*). Mjera i metrika nisu jedno te isto, ali nema potrebe da se time ovdje zamaramo.

[4] Primijetite da točnost nije ništa drugo nego procjena očekivanja funkcije gubitka 0-1 izračunata na označenom skupu podataka (uzorku).

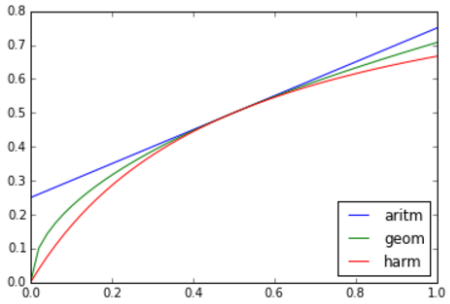
[5] Ovdje postoji mogućnost terminološke konfuzije. **Stvarno pozitivni** (engl. *true positives*) primjeri su primjeri koji su pozitivni i označeni kao takvi. Međutim, **lažno negativni primjeri** (engl. *false negatives*) su zapravo također pozitivni primjeri. Ovdje treba biti pragmatičan i usvojiti sljedeće tumačenje: stvarno pozitivni primjeri su podskup pozitivnih primjera, a prilog "stvarno" znači "klasificirani kao pozitivni i stvarno su pozitivni".

[6] **Neuravnoteženost skupa podataka** (engl. *dataset imbalance, class imbalance*) nije samo problem kod vrednovanja modela, već i kod njegovog treniranja. Standardni algoritmi strojnog učenja minimiziraju empirijsku pogrešku koja je aproksimacija gubitka 0-1, gdje se gubitak na primjeru tretira jednako nevisno iz koje klase dolazi. Posljedično, modeli će tipično predviđati loše na klasama s manje primjera jer je tamo ukupan gubitak algoritma manji nego na većim klasama. Treniranje na neuravnoteženim skupovima podataka je realan problem koji se često pojavljuje u praksi. Na ovom kolegiju nažalost nemamo vremena baviti se njime. Zainteresirane upućujem na (Chawla et al., 2004) i (Krawczyk, 2016) te alat `imbalanced-learn` za `scikit-learn`.

[7] U kontekstu mjere vredovanja modela strojnog učenja, **odziv** (ili **odaziv**) ispravan je hrvatski prijevod engleskog termina *recall*. Međutim, u bespućima internetske zbiljnosti možda ćete nabasati na *opoziv*, što nije točan prijevod u kontekstu mjera vrednovanja, premda jest jedan od mogućih prijevoda ove riječ (imenica *recall* u engleskom jeziku ima barem pet značenja; v. ovdje). Pogrešan prijevod možda je motiviran naslovom filma "Totall Recall", koji je kod nas preveden kao "Totalni opoziv" (što je također moguće pogrešan prijevod, ako uzmete u obzir radnju filma).

[8] **Mjera** F_1 zapravo je poseban slučaj **F-mjere**, koju je 1992. godine predložio je van Rijsbergen (van Rijsbergen, 1979) u kontekstu pretraživanja informacija (engl. *information retrieval*). Mjera se od tamo preuzeta u druga područja, uključivo strojno učenje i obradu prirodnog jezika.

[9] Od triju **pitagorejskih sredina** – aritmetičke, geometrijske i harmonijske – harmonijska sredina za različita opažanja daje najmanju, dok aritmetička sredina daje najveću srednju vrijednost. Drugim riječima, aritmetička sredina je gornja ograda geometrijske sredine, dok je geometrijska sredina gornja ograda harmonijske sredine (za dokaz, v. ovdje). To možemo pokazati i grafički. Razmotrimo sredinu dviju vrijednosti, pri čemu fiksirajmo jednu na 0.5. Ako sada mijenjamo drugu vrijednost u intervalu od 0 do 1, onda za aritmetičku, geometrijsku i harmonijsku sredinu dobivamo sljedeće:



Vidimo da, ako su obje vrijednosti jednake 0.5, onda su sve tri sredine također jednake 0.5. Međutim, čim su vrijednosti različite, harmonijska sredina manja je i od aritmetičke i od geometrijske sredine.

10 Premda ne postoji neko dogovoreno pravilo za tretiranje rubnih slučajeva za mjere P, R i F_1, uobičajeno je da se nedefinirana vrijednost postavlja na nulu. Tako to radi i `scikit-learn`.

11 Mjera F_1 poseban je slučaj mjere F_β, definirane kao:

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

Parametar β, gdje $\beta > 0$, određuje koliko je puta odziv važniji od preciznosti. Uobičajeno se koristi $\beta = 0.5$ (mjera $F_{0.5}$) ako se želi naglasiti preciznost, ili $\beta = 2$ (mjera F_2) ako se želi naglasiti odziv.

12 Zapravo, imamo tri mogućnosti. Pored mikro-prosjeka i makro-prosjeka postoji i treća mogućnost: izračun **težinskog prosjeka** mjera kroz sve klase, gdje su težine udjeli svake klase u skupu primjera. Formalno, za neku odabranu mjeru vrednovanja m, mjera **težinski-**m definirana je kao:

$$m^{avg} = \sum_j \frac{N_j}{N} m_j$$

gdje je N_j broj primjera u klasi $y = j$, tj. $N_j = \sum_{i=1}^{N} \mathbf{1}\{y^{(i)} = j\}$, a m_j je mjera izračunata na matrici zabune 2×2 za klasu s oznakom $y = j$ (kao što to radimo kod makro-mjera). Težinske mjere, npr., **težinski-**F_1 (engl. *weighted* F_1), ima smisla koristiti kad god nam je važnost svake klase proporcionalna njezinoj veličini.

13 Pokažimo da vrijedi $P^\mu = R^\mu = F_1^\mu = Acc$. Pokažimo najprije $P^\mu = R^\mu$. Intuitivno, mjere mikro-preciznost i mikro-odziv daju jednake vrijednosti zato što je u združenoj matrici zabune broj lažno pozitivnih primjera uvijek jednak broju lažno negativnih primjera (ono što je lažno pozitivno za jednu klasu lažno je negativno za drugu). Naime, za matricu zabune C dimenzija $K \times K$, broj lažno pozitivnih primjera i broj lažno negativnih primjera za klasu j jednak je:

$$\mathrm{FP}_j = \sum_{i:i\neq j} C[j,i] \qquad\qquad \mathrm{FN}_j = \sum_{i:i\neq j} C[i,j]$$

U združenoj matrici zabune onda imamo zbroj lažno pozitivnih primjera kroz sve klase, $\sum_j \mathrm{FP}_j$, i zbroj lažno negativnih primjera kroz sve klase, $\sum_j \mathrm{FN}_j$, a za njih vrijedi:

$$\sum_j \mathrm{FP}_j = \sum_j \sum_{i:i\neq j} C[j,i] = \sum_j \sum_{i:i\neq j} C[i,j] = \sum_j \mathrm{FN}_j$$

Posljedično:

$$P^\mu = \frac{\sum_j \mathrm{TP}_j}{\sum_j \mathrm{TP}_j + \sum_j \mathrm{FP}_j} = \frac{\sum_j \mathrm{TP}_j}{\sum_j \mathrm{TP}_j + \sum_j \mathrm{FN}_j} = R^\mu$$

Mjera F_1^μ je harmonijska sredina mjera P^μ i R^μ, pa, budući da $P^\mu = R^\mu$, to vrijedi $F_1^\mu = P^\mu = R^\mu$. Konačno, pokažimo da je mjera točnosti, definirana nad matricom zabune C dimenzija $K \times K$ kao

$$Acc = \sum_j \frac{C[j,j]}{N}$$

jednaka trima navedenim mjerama. Dovoljno je da pokažemo da vrijedi $Acc = P^\mu$. U združenoj matrici broj stvarno pozitivnih primjera odgovara ukupnome broju točno klasificiranih primjera, budući da:

$$\sum_j \mathrm{TP}_j = \sum_j C[j,j]$$

dok zbroj stvarno pozitivnih primjera i lažno pozitivnih primjera odgovara ukupnome broju primjera, budući da:

$$\sum_j \mathrm{FP}_j = \sum_j \sum_{i:i\neq j} C[i,j] = N - \sum_j \mathrm{TP}_j$$

iz čega slijedi $N = \sum_j \mathrm{TP}_j + \sum_j \mathrm{FP}_j$. Uvrstimo li ovo u izraz za mjeru točnosti, dobivamo:

$$Acc = \frac{\sum_j C[j,j]}{N} = \frac{\sum_j \mathrm{TP}_j}{\sum_j \mathrm{TP}_j + \sum_j \mathrm{FP}_j} = P^\mu$$

14 Na isti način kao što smo izračunali makro-preciznost i makro-odziv, mogli smo izračunati i **makro-točnost**, Acc^M. Također, kao što smo izračunali mikro-preciznost i mikro-odziv, mogli smo izračunati i **mikro-točnost**, Acc^μ. No, te dvije mjere uvijek daju jednake vrijednosti. Naime:

$$Acc^M = \frac{1}{K}\sum_j \frac{\mathrm{TP}_j + \mathrm{TN}_j}{\mathrm{TP}_j + \mathrm{FP}_j + \mathrm{FN}_j + \mathrm{TN}_j} = \frac{\sum_j(\mathrm{TP}_j + \mathrm{TN}_j)}{KN}$$
$$= \frac{\sum_j(\mathrm{TP}_j + \mathrm{TN}_j)}{\sum_j(\mathrm{TP}_j + \mathrm{FP}_j + \mathrm{FN}_j + \mathrm{TN}_j)} = \frac{\sum_j \mathrm{TP}_j + \sum_j \mathrm{TN}_j}{\sum_j \mathrm{TP}_j + \sum_j \mathrm{FP}_j + \sum_j \mathrm{FN}_j + \sum_j \mathrm{TN}_j} = Acc^\mu$$

Premda općenito daju različitu vrijednost od mjere točnosti Acc izračunate na matrici $K \times K$, ove se mjere u praksi rijetko koriste, budući da pate od istih problema kao i mjera točnosti (visoka točnost trivijalnog klasifikatora na skupu primjera s neuravnoteženim klasama).

15 Klasifikatori će na stvarnim podatcima općenito biti lošiji na klasama s manjim primjerima, pa je **makro-F_1 tipično manji od mikro-F_1** ($F_1^M < F_1^\mu$). U tom smislu, ako vam netko prezentira samo vrijednost mjere F_1^μ, a da za to ne ponudi adekvatno objašnjenje, zapitajte se da to nije možda zato što klasifikator radi dobro samo na klasama s velikim brojem primjera.

16 Ovdje pretpostavljamo, što je uobičajena pretpostavka, da za neki odabrani prag p, $p \in [0,1]$, primjer klasificiramo u klasu $y = 1$ akko $P(y = 1|\mathbf{x}) \geqslant p$, a u klasu $y = 0$ akko $P(y = 1|\mathbf{x}) < p$. Drugim riječima, model binarnog klasifikatora definiramo kao $h(\mathbf{x}) = \mathbf{1}\{p(y|\mathbf{x}) \geqslant p\}$. (Primijetite da je prag p hiperparametar modela, budući da se ne optimira učenjem modela, ali se može optimirati post hoc). Međutim, u nekim primjenama, pogotovo kod donošenja odluka visokog rizika, bolje je klasifikator definirati tako da se suzdrži od klasifikacije ako ona nije dovoljno pouzdana, tj. ako je $P(y|\mathbf{x})$ previše blizu vrijednosti 0.5 (što je "previše blizu" ovisi, naravno, o konkretnom slučaju). Pretpostavka u takvim slučajevima jest da je bolje da se klasifikator suzdrži od nepouzdane klasifikacije (i prepusti takve slučajeve drugim mehanizmima, na primjer ručnoj obradi stručnjaka) nego da napravi pogrešnu klasifikaciju. Takav se pristup naziva **klasifikacija s opcijom odbijanja** (engl. *classification with reject option*). Očekivano, za klasifikatore s opcijom odbijanja trebaju nam neke druge mjere vrednovanja od ovih koje smo opisali, budući da sada treba vrednovati i koliko je klasifikator dobar u procjeni svoje vlastite pouzdanosti, odnosno koliko je dobro kalibriran, te treba uzeti u obzir kompromis između cijene pogrešne klasifikacije i cijene neprovođenja klasifikacije. Više o klasifikaciji s odbijanjem možete pročitati u (Herbei and Wegkamp, 2006), a o vrednovanju takvih klasifikatora u (Condessa et al., 2017).

17 Ovdje se zapravo radi o **višekriterijskoj optimizaciji**: nastojimo optimizirati vrijednost praga na temelju dvaju kriterija, preciznosti i odziva. Kombinacije preciznosti i odziva koje su bolje od nekih drugih kombinacija preciznosti i odziva, a međusobno nisu usporedive, čine **Paretovu frontu**.

18 Neobičan naziv "**ROC krivulja**" (engl. *receiver operating characteristics*) dolazi od primjene ove metode u Drugom svjetskom ratu za procjenu uspješnosti analize radarskih signala.

19 Ako je binarni klasifikator lošiji od nasumičnog, njegove predikcije uvijek možemo obrnuti i tako dobiti bolji klasifikator. Ipak, takve situacije su neobične u praksi. Ako je točnost klasifikatora lošija

od nasumičnog odabira, to upućuje ili na tehničku pogrešku (npr., u optimizaciji) ili da ne postoji jasan signal koji bi klasifikator mogao naučiti, pa je klasifikator naučio šum. Ni u kojem od ta dva slučaja obrtanje oznaka ne rješava stvarni problem.

20 Široko korištene metode **ponovnog uzorkovanja** (engl. *resampling*) u statistici su **jackknife**, **bootstrap** i **permutacijski test**. Posljednje dvije neizostavne su u *toolboxu* onih koji se žele baviti znanošću o podatcima. Više možete pronaći u (Davison and Hinkley, 1997) i (Good, 2006).

21 Možemo, međutim, pomoću LOOCV dobiti predikcije za svaki primjer pojedinačno pa sve to objediniti i nad time izračunati koju god mjeru vrednovanja želimo, no takva je procjena lošija jer nije dobivena kao srednja vrijednost procjena nad više uzoraka.

22 Pažljivi čitatelj primijetit će da nam ugniježđena unakrsna provjera $k \times l$ nam procjenu prosječne pogreške modela, ali nam ne daje jednoznačan odgovor na pitanje koji je model zapravo najbolji, jer optimalni modeli mogu biti različiti u svakoj iteraciji vanjske petlje. Također ne dobivamo odogovor na pitanje koji je naučeni model (hipoteza) najbolji, jer se i naučeni modeli se mogu razlikovati u svakoj iteraciji vanjske petlje. Ovo je u redu, jer unakrsna provjera služi za nepristranu i poštenu procjenu pogreške generalizacije algoritma strojnog učenja, a ne jednog specifičnog modela. U konačnici, međutim, nama treba jedan naučen model, kako bismo ga isporučili odnosno ugradili u neki drugi sustav. Do njega tipično dolazimo tako da odabiremo onaj model (one hiperparametre) koji su najčešće davali optimalan model (ili interpoliramo između vrijednosti hiperparametara koji su davali najbolje modele) u unutarnjoj petlji. Takav model onda učimo na kompletnom skupu označenih primjera $\mathcal{D}$, i njega isporučujemo. Za taj model više nemamo procjenu pogreške, jer više nemamo izdvojenih podataka na kojima bismo ga ispitali, ali očekujemo da će pogreška biti jednaka ili manja od one koje smo dobili ugniježđenom unakrsnom provjerom. Zašto? Zato što smo model učili na kompletnom skupu $\mathcal{D}$, a ne samo na skupu za učenje. U tom smislu, procjena pogreške dobivena ugniježđenom unakrsnom provjerom na skupu $\mathcal{D}$ je pesimistična procjena stvarne pogreške optimalnog modela naučenog na cjelom skupu $\mathcal{D}$.

Literatura

N. V. Chawla, N. Japkowicz, and A. Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6, 2004.

F. Condessa, J. Bioucas-Dias, and J. Kovačević. Performance measures for classification systems with rejection. *Pattern Recognition*, 63:437–450, 2017.

A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*. Number 1. Cambridge university press, 1997.

P. I. Good. *Resampling methods*. Springer, 2006.

R. Herbei and M. H. Wegkamp. Classification with reject option. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, pages 709–721, 2006.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

N. Japkowicz and M. Shah. *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

S. Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

C. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.