

Umjetna inteligencija

10. Strojno učenje

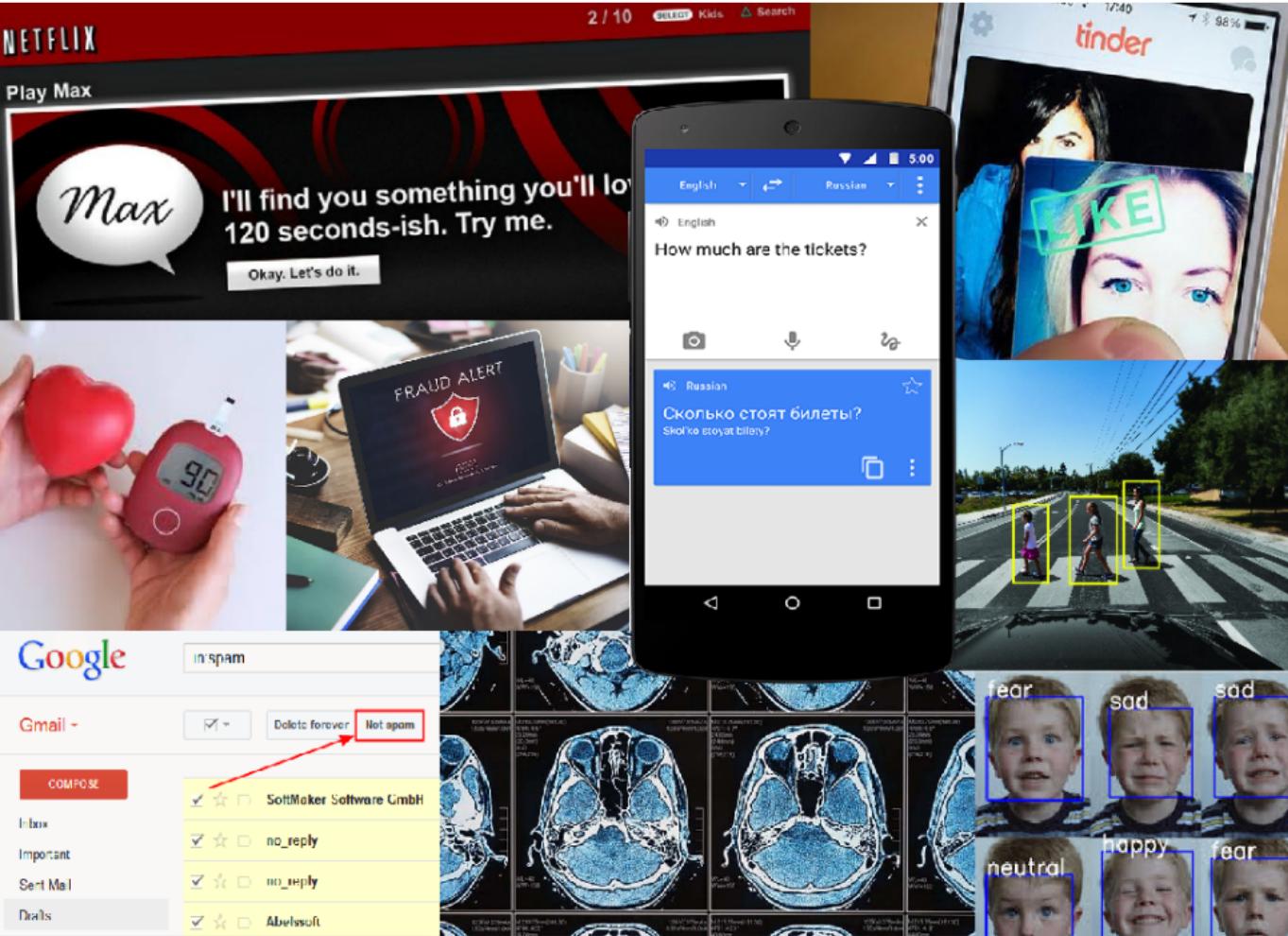
prof. dr. sc. Bojana Dalbelo Bašić
izv. prof. dr. sc. Jan Šnajder

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Ak. god. 2019./2020.



Creative Commons Imenovanje–Nekomercijalno–Bez prerada 3.0



Data Mining Cup 2010

Using the existing characteristics of a customer's initial order, such as order quantity per type of goods, title and delivery weight, a **decision must be made** on whether to send a voucher worth EUR 5.00. The customers who receive a voucher should be those who would not have decided to re-order by themselves.

IEEE ICDM 2010 DM Competition

Modeling the process of traffic jams formation during morning peak in the presence of roadworks, based on initial information about jams broadcast by radio stations. Input data contain identifiers of road segments closed due to roadworks, accompanied by a sequence of segments where the first jams occurred. The algorithm should **predict** a sequence of segments where next jams will occur in the nearest future.

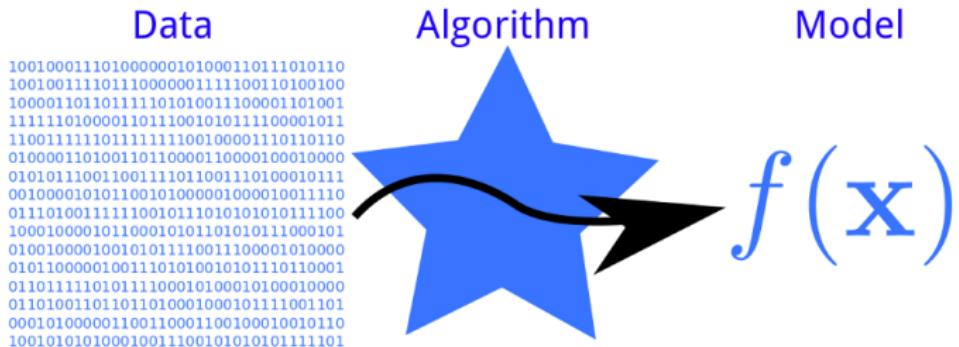
Sadržaj

- 1 Uvod u strojno učenje
- 2 Pristupi strojnom učenju
- 3 Nadzirano učenje
- 4 Algoritam 1: Naivan Bayesov klasifikator
- 5 Algoritam 2: Stabla odluke

Sadržaj

- 1 Uvod u strojno učenje
- 2 Pristupi strojnom učenju
- 3 Nadzirano učenje
- 4 Algoritam 1: Naivan Bayesov klasifikator
- 5 Algoritam 2: Stabla odluke

Što je strojno učenje?



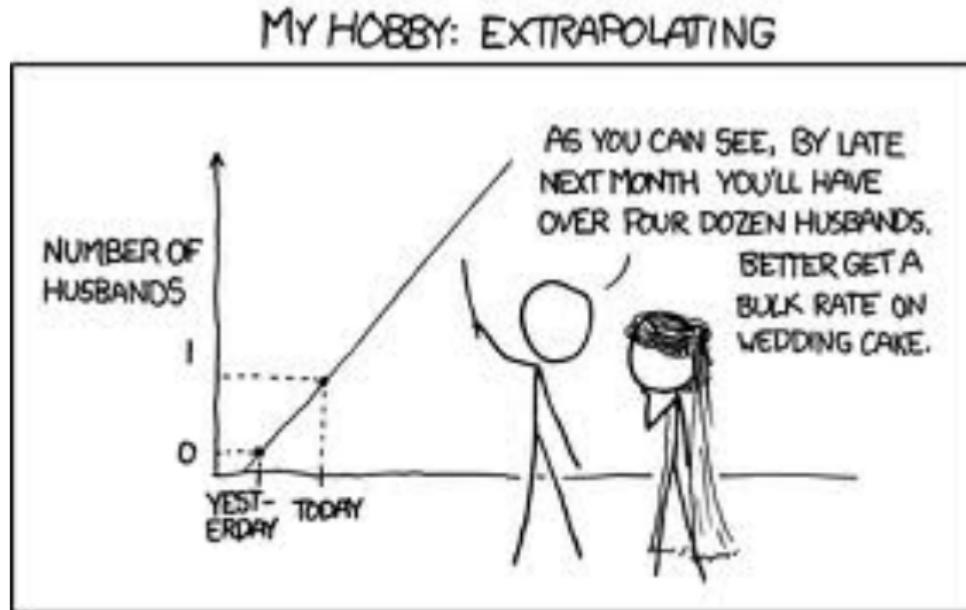
Strojno učenje (Alpaydin 2009)

Strojno učenje jest programiranje računala na način da **optimiziraju** neki **kriterij uspješnosti** temeljem **podatkovnih primjera** ili prethodnog iskustva. Raspolažemo **modelom** koji je definiran do na neke **parametre**, a **učenje** se svodi na izvođenje algoritma koji **optimizira** parametra modela na temelju podataka ili prethodnog iskustva.

Predviđanje

- Na temelju viđenih podataka, model mora moći **predvidjeti** svojstva novih, još neviđenih podataka
- Drugačije rečeno, model mora moći **generalizirati**
- Sa stajališta logike, model zapravo radi **indukciju**: na temelju ograničenog skupa podataka, model izvodi generalno pravilo koja objašnjava kako se podatci ponašaju
- Cilj strojnog učenja: izgraditi modele koji **dobro generaliziraju!**

Generalizacija iziskuje dovoljno podataka!



Zašto strojno učenje?

Barem tri razloga:

- ① **Problemi koji su presloženi da bi ih se riješilo algoritamski** – nemamo ideju kako ih riješiti, jer ne znamo niti kako ih ljudi rješavaju (npr. raspoznavanje govora)
 - ▶ problemi koje nije moguće riješiti na klasičan algoritamski način (*UI-potpuni problemi*)
- ② **Sustavi koji se dinamički mijenjaju** – potrebna prilagodba (npr. prilagodba korisničkih sučelja)
- ③ **Ogromne količine podataka** – raspolažemo ogromnim količima podataka, možemo li iz njih izvući znanje?
 - ▶ znanost o podatcima (engl. *data science*), veliki podatci (engl. *big data*), otkrivanje znanja u skupovima podataka (engl. *data mining*)

NB: Ne zaslužuju svi problemi strojno učenje (npr. softver za obračun plaća). Ne koristite čekić da biste ubili komarca.

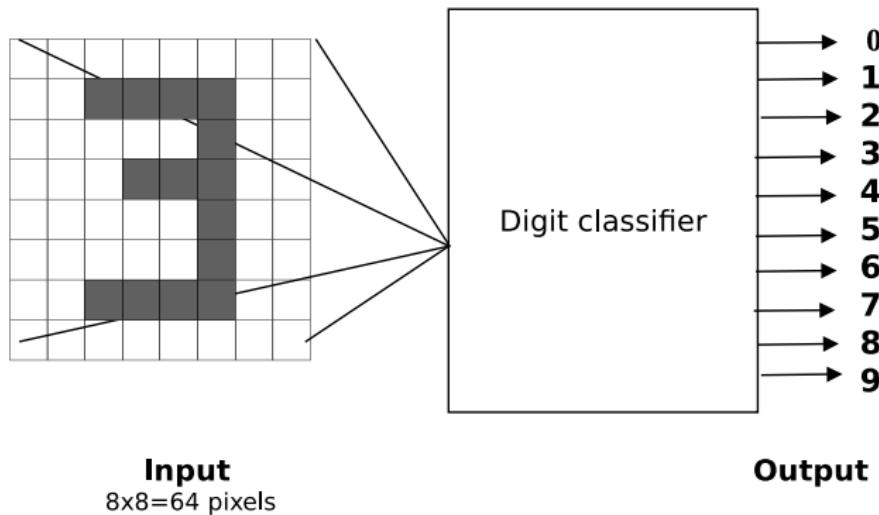
Slučaj 1: Teški problemi

- Raspoznavanje uzorka
- Računalni vid
- Obrada prirodnog jezika
- Raspoznavanje govora
- Robotika
- ...



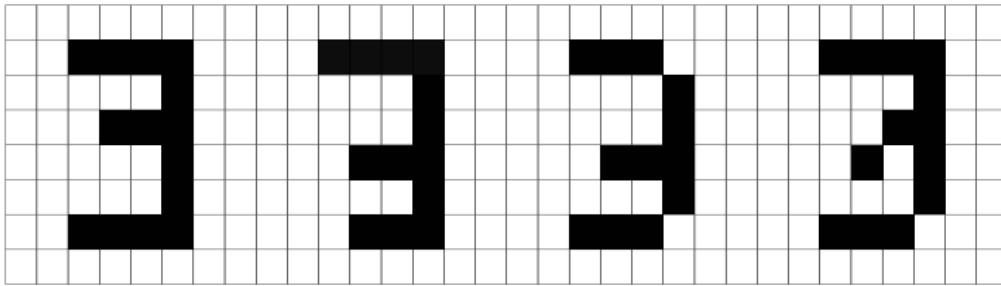
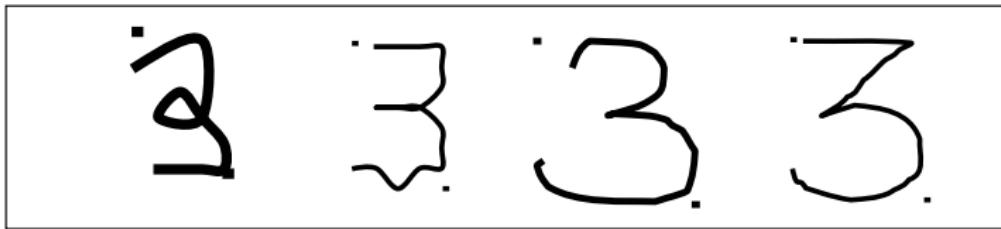
Primjer: Raspoznavanje rukom pisanih znameki

- Sustavu predočavamo **uzorke** (digitalizirani ulaz)
- Sustav treba **raspoznati** kojoj od predefiniranih 10 klasa ulaz pripada
- Ovaj se postupak zove **raspoznavanje uzorka** (engl. *pattern recognition*) ili **klasifikacija uzorka**



Primjer: Raspoznavanje rukom pisanih znameki

- Problem je što nitko ne piše savršene znamenke: postoji velik broj varijanti svake znamenke



Primjer: Raspoznavanje rukom pisanih znameki

- Naivan pristup:
 - ▶ u memoriju pohrani sve moguće varijante svake znamenku
 - ▶ kada se predoči znamenka (matrica binarnih brojeva 8×8), pretraži memoriju i dohvati znamenku
 - **Q:** Zašto ovo nije dobra ideja?
 - **A:** Zato što je broj različitih uzoraka prevelik: $2^{64} = 1.8 \cdot 10^{19}$
-
- ① Memorijski problem
 - ▶ ako svaki uzorak zauzima 8 bajtova $\Rightarrow 8 \cdot 2^{64}$ bajtova = **157 EB**
 - ② Vremenski problem
 - ▶ ako uspoređivanje dvaju uzoraka traje 100 mikrosekundi \Rightarrow prosječno pretraživanja traje $\frac{1}{2} \cdot 2^{64} \cdot 10^{-4}$ sekundi = **29 milijuna godina**
 - ③ Konceptualni problem
 - ▶ čak i kada bismo mogli riješiti ove probleme, kako bismo uopće sakupili podatke? Ručno pridjeljivanje znamekne svakom uzorku očito bi trajalo mnogo više od 29 milijuna godina!

Slučaj 2: Dinamički prilagodivi sustavi

- Inteligentni računalni sustavi moraju se moći **prilagoditi svojoj okolini**
- Sustav koji je sposoban učiti može se bolje prilagoditi novim situacijama
- Robotika
- Višeagentni sustavi
- Inteligentna korisnička sučelja

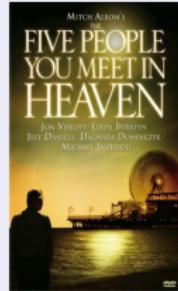
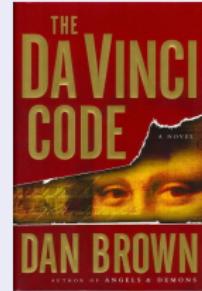


Slučaj 3: Od podataka do znanja

- Podatci → informacije → znanje → odluka
- **Podataka** ima u izobilju (web, tekst, eksperimentalni podatci, skladišta podataka, *deep web, logovi*)
- S druge strane, **znanja** nema puno i ono je skupo
- Cilj: izgradnja **modela** koji objašnjavaju podatke i omogućavaju zaključivanje/predviđanje

Primjer: Analiza ponašanja korisnika na temelju kupovnih transakcija

Osobe koje su kupile "The da Vinci Code" također su kupile i
"The Five People You Meet in Heaven"

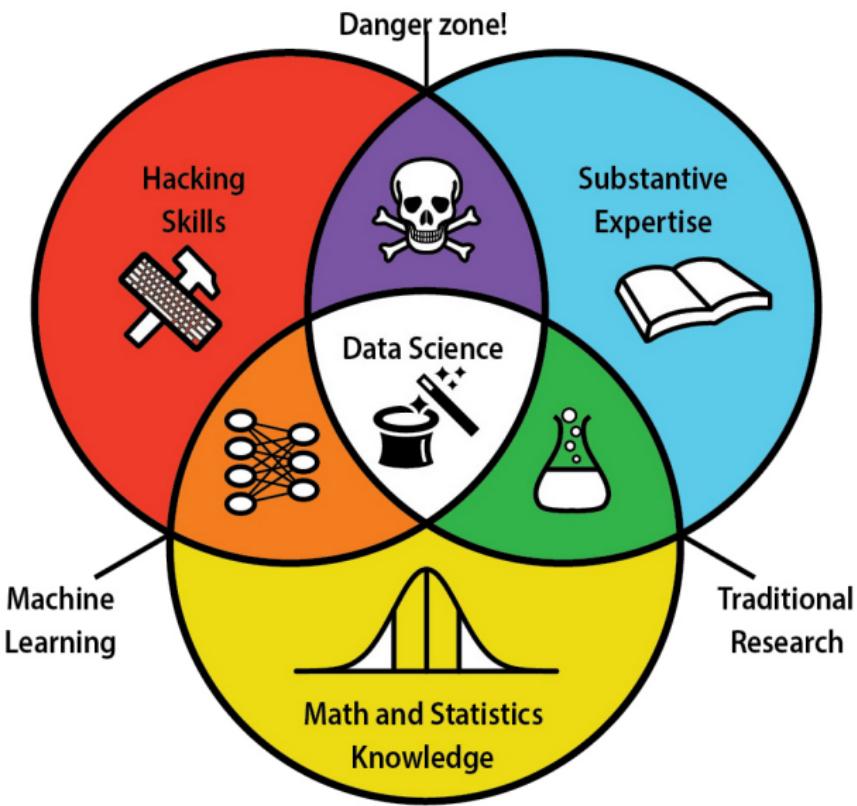


Dubinska analiza podataka

Dubinska analiza podataka (engl. *data mining*) ili otkrivanje znanja u skupovima podataka (engl. *knowledge discovery in datasets*) – primjena strojnog učenja na (polu)strukturirane baze podataka

- Trgovina: analiza potrošačke košarice, CRM
- Financije: Određivanje kreditne sposobnosti, detekcija zlouporaba kartica
- Proizvodnja: optimizacija, *troubleshooting*
- Medicina: postavljanje dijagnoza
- Telekomunikacije: optimizacija usluga
- Bioinformatika: analiza izražajnosti gena, poravnavanje gena
- Text mining: klasifikacija teksta, ekstrakcija informacija
- ...

Znanost o podatcima (engl. *Data Science*)

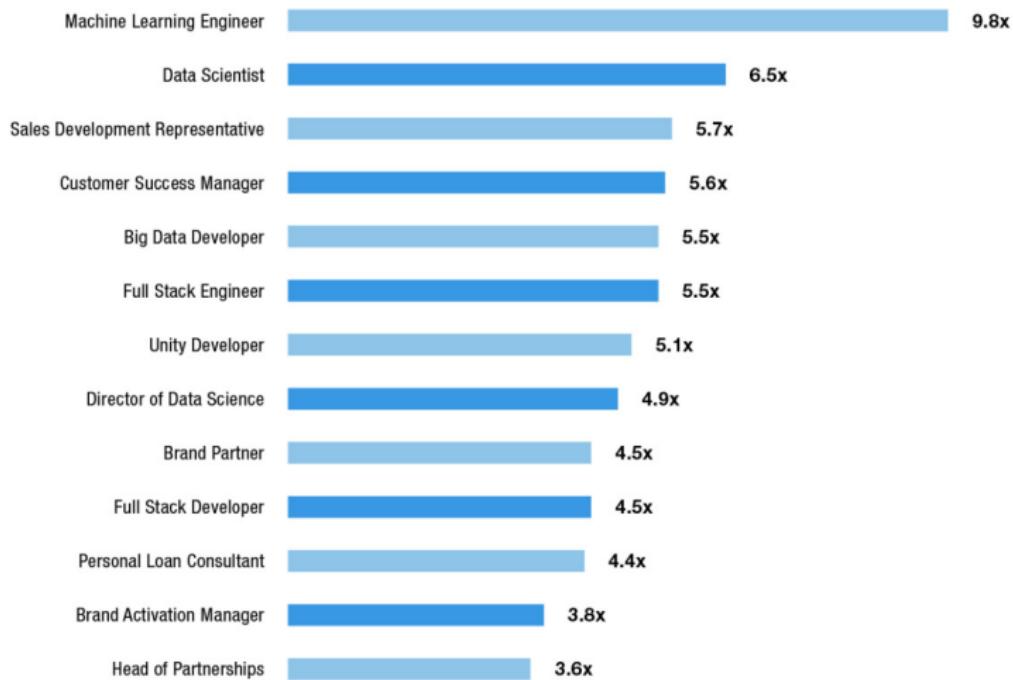


<http://berkeleysciencereview.com/how-to-become-a-data-scientist-before-you-graduate/>

Znanost o podatcima (engl. *Data Science*)

LinkedIn: The fastest growing jobs in the US (2012–2017)

Top 20 Emerging Jobs



Obećanja i opasnosti Znanosti o podatcima

Most of the knowledge in the world in the future is going to be extracted by machines and will reside in machines.

– *Yann LeCun, Director of AI Research, Facebook*

If we used all our technology resources, we could actually give people personalized recommendations for every step of your life.

– *Aneesh Chopra, former CTO of the U.S.*

Strojno učenje i kognitivna znanost

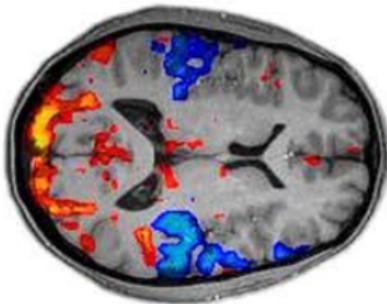
Kognitivno vjerodostojni modeli

Razumijevanje algoritama strojnog učenja \Leftrightarrow Razumijevanje ljudske sposobnosti (ili ograničenja) učenja

Thought Reading Experiment:

<http://www.cs.cmu.edu/afs/cs/project/theo-73/www/index.html>

- **Funkcijska magnetna rezonancija (fMRI)**
 - ▶ Bilježi protok krvi kroz mozak: aktivna područja mozga koriste više kisika.
 - ▶ Oslanjanje na činjenicu da molekule u krvnim stanicama reagiraju u magnetskom polju u ovisnosti o količini kisika



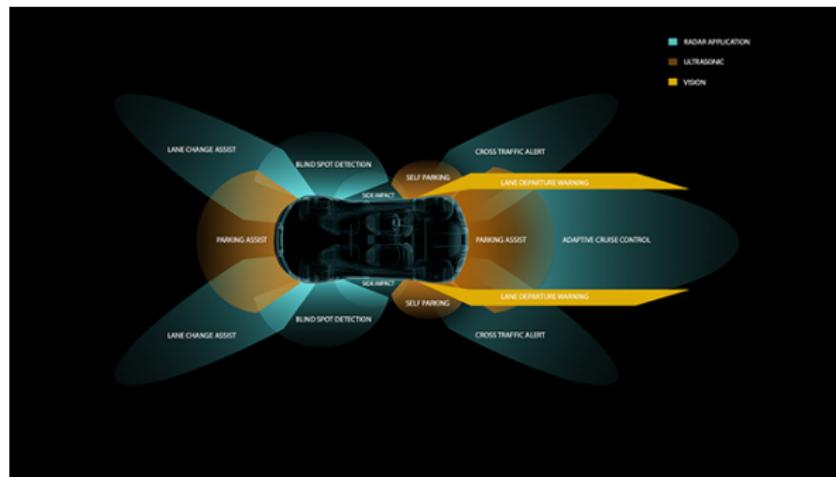
Duboko učenje: trenutačni hit u strojnom učenju

- Pristup strojnom učenju temeljen na **neuronskim mrežama i velikoj količini podataka**
- Takav pristup je sada moguć zbog enormousne količine podataka generirane od strane (internet) korisnika (*user-generated content*)
- Postiže *state-of-the-art* rezultate na širokom spektru zadataka
- Google, Baidu, Paypal, NVIDIA, Facebook, ...

Duboko učenje: autonomno vozilo (NVIDIA)

<http://blogs.nvidia.com/blog/2015/02/24/deep-learning-drive/>

- Razni senzori skupljaju veliku količinu podataka koje DU koristi za razne izazove unutar automatske vožnje
- Prepoznavanje pješaka, automatsko parkiranje, pomoć pri prestrojavanju itd.



Dubinsko učenje: autonomno vozilo (NVIDIA)

<http://blogs.nvidia.com/blog/2015/02/24/deep-learning-drive/>

- Klasifikator raspoznaće razne vrste vozila i prometnih znakova, čak i kada je čovjeku teško obaviti taj zadatak (kada je nevrijeme i sl.)



Dubinsko učenje: prepoznavanje broja kalorija (Google)

<http://www.popsci.com/google-using-ai-count-calories-food-photos>

- Sustav na temelju fotografije jela procjenjuje koliko ono sadrži kalorija



Sadržaj

- 1 Uvod u strojno učenje
- 2 Pristupi strojnom učenju
- 3 Nadzirano učenje
- 4 Algoritam 1: Naivan Bayesov klasifikator
- 5 Algoritam 2: Stabla odluke

Vrste strojnog učenja

(1) Nadzirano učenje (engl. *supervised learning*)

Podatci su u obliku (*ulaz, izlaz*)=(\mathbf{x}, y), treba naći preslikavanje $\hat{y} = f(\mathbf{x})$

- Ako je y je diskretna/nebrojčana vrijednost: **klasifikacija**
- Ako je y kontinuirana/brojčana vrijednost: **regresija**

(2) Nenadzirano učenje (engl. *unsupervised learning*)

Dani su podaci bez ciljne vrijednosti, treba naći pravilnost u podacima

- **grupiranje** (engl. *clustering*)
- **otkrivanje stršećih/novih vrijednosti** (engl. *outlier/novelty detection*)
- **smanjenje dimenzionalnosti** (engl. *dimensionality reduction*)

(3) Podržano/ojačano učenje (engl. *reinforcement learning*)

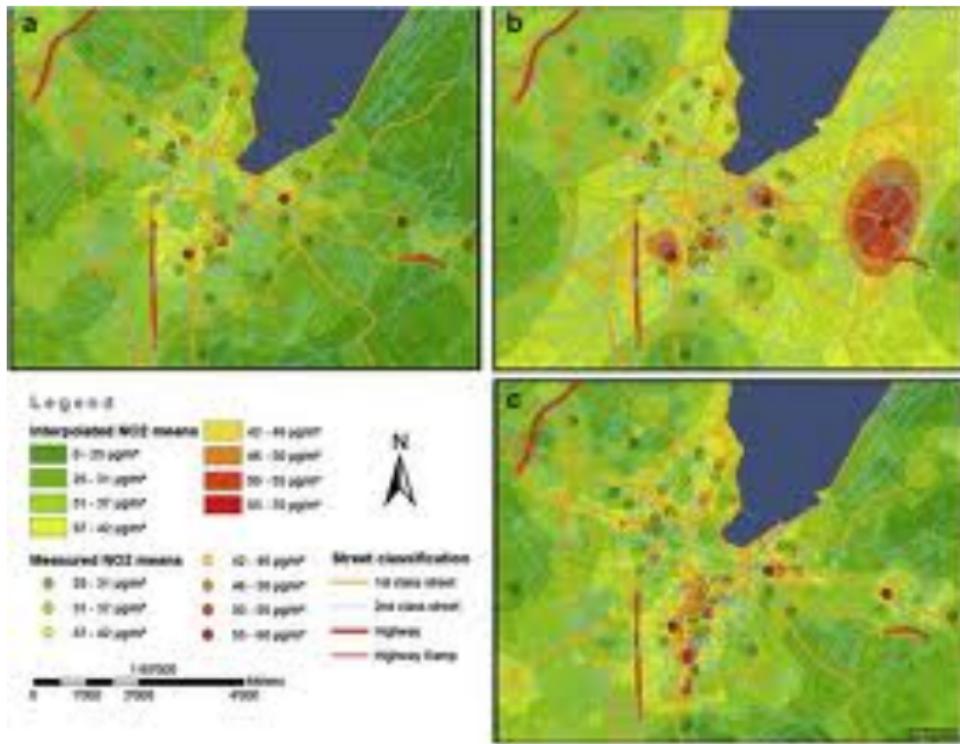
Učenje **optimalne strategije** na temelju pokušaja s odgođenom nagradom

Primjer klasifikacije: predviđanje ishoda izbora



Election Analytics (<http://electionanalytics.cs.illinois.edu/index.html>)

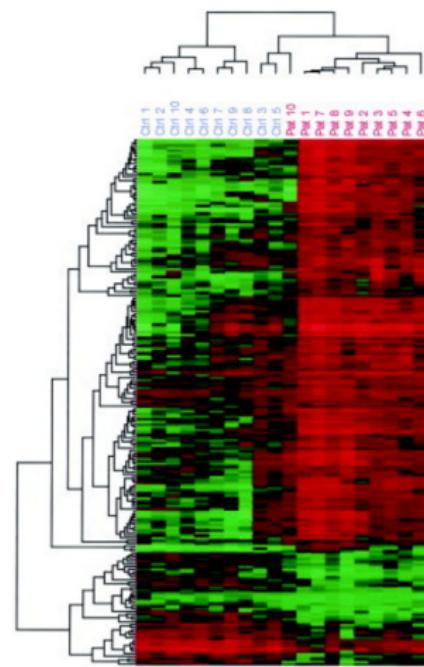
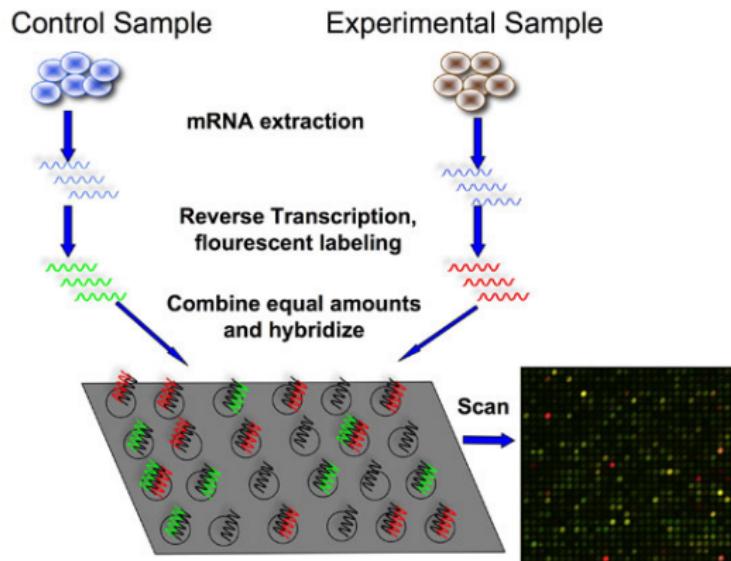
Primjer regresije: predviđanje razine emisije NO₂



Exposure models for traffic related NO₂ (<http://www.sciencedirect.com/science/article/pii/S1352231011009629>)

Primjer grupiranja: analiza DNA-mikropolja

Cilj: grupiranje gena sa sličnom izražajnošću
(slična izražajnost – slična funkcionalnost)

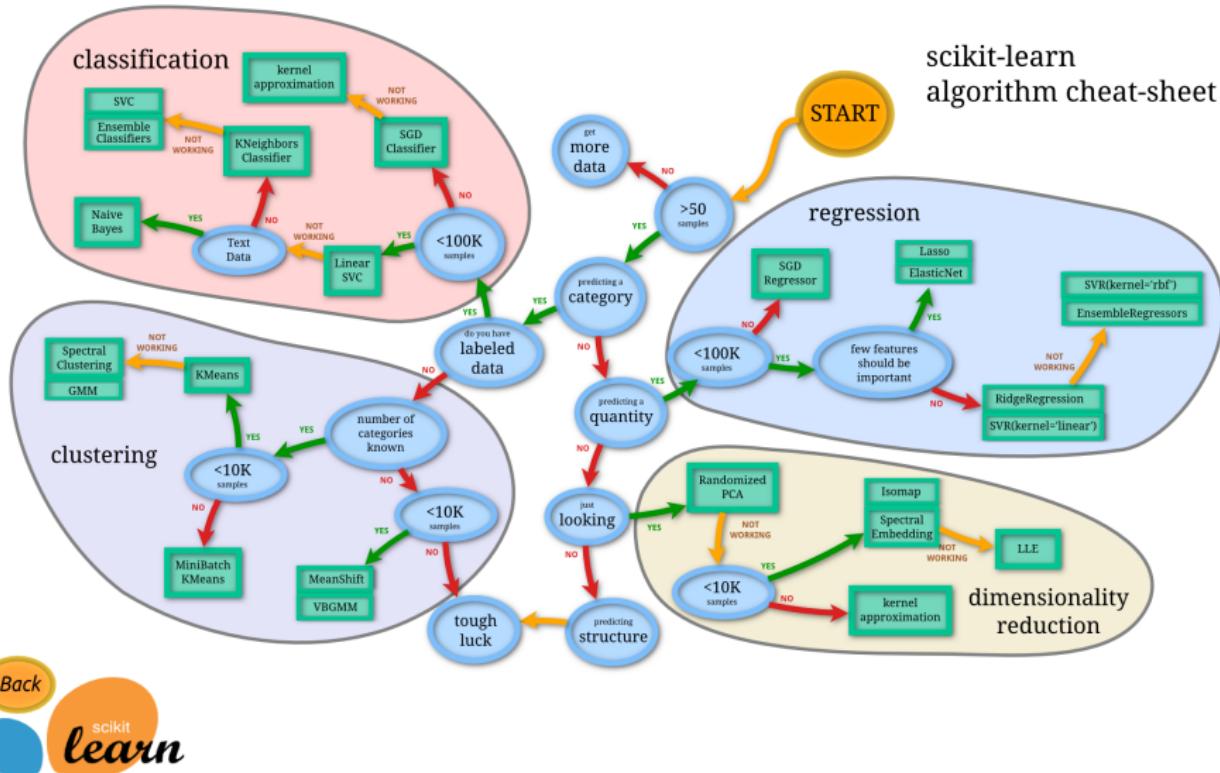


Podržano učenje: AlphaGo

- Dvije duboke neuronske mreže (DNN), trenirane da predviđaju idući potez i smanje prostora pretraživanja
- **Podržano učenje** kao nadogradnja na DNN, za učenje strategije igranja, treniranjem sustava da igra sam protiv sebe
- Za iganje koristi 40 dretvi, 48 CPU-a, i 8 GPU-a. Za treniranje koristi 40 dretvi, 1202 CPU-a i 176 GPU-a



Karta strojnog učenja



http://scikit-learn.org/stable/tutorial/machine_learning_map/

Primjene, zadatci, modeli i algoritmi

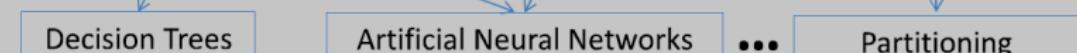
Applications



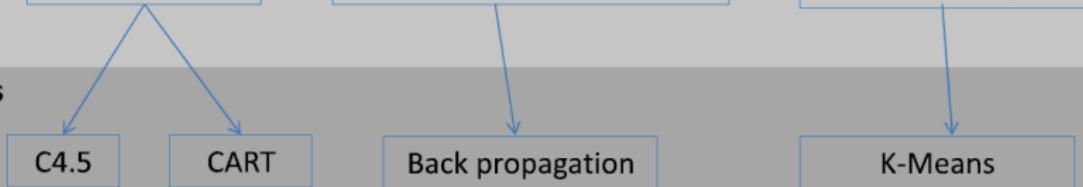
Tasks



Models



Algorithms



Rokach, L. and Maimon, O. Z. (2008). Data Mining with Decision Trees: Theory and Applications.

Pet škola strojnog učenja

Pedro Domingos: *The Master Algorithm*



Škola	Temelji	Glavni algoritam
Symbolists	Logika, filozofija	Indukcija
Connectionists	Neuroznanost	Backpropagation
Evolutionaries	Evolucijska biologija	Genetičko programiranje
Bayesians	Statistika	Probabilističko zaključivanje
Analogizers	Psihologija	Jezgreni strojevi

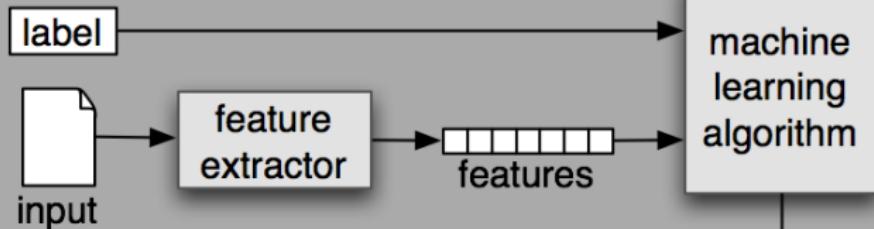
NB: Mi ćemo obraditi jedan bayesovski i jedan simbolistički algoritam

Sadržaj

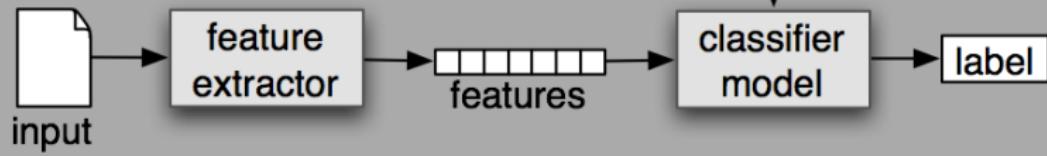
- 1 Uvod u strojno učenje
- 2 Pristupi strojnom učenju
- 3 Nadzirano učenje
- 4 Algoritam 1: Naivan Bayesov klasifikator
- 5 Algoritam 2: Stabla odluke

Treniranje i predikcija

(a) Training



(b) Prediction



Treniranje i predikcija

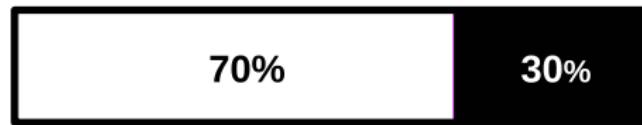
- Nadzirani model treba najprije **trenirati** (naučiti) na označenim primjerima, a nakon toga se može koristiti za **predikciju** (klasifikaciju ili regresiju) na dotad neviđenim primjerima
- Svaki primjer predstavljen je **vektorom značajki** x , gdje svaka značajka kodira jedan aspekt ulaza (npr., visina, težina, prihod, ...)
- Ekstraktor značajki (engl. *feature extractor*) generira vektor značajki x za svaki ulazni primjer (taj korak je identičan i za treniranje i za predikciju)
- Tijekom treniranja (učenja), model na ulazu dobiva **primjere** x i njihove **oznake** (engl. *labels*) y , tj. parove (x, y)
- Nakon što je treniran, isti model koristi se za predikciju
- Tijekom predikcije, model na ulazu dobiva samo primjere x , dok na izlazu daje oznaku y

Prenaučenost

- Naš glavni cilj je da trenirani model dobro generalizira
- Ako je model vrlo složen, onda se može se vrlo dobro prilagoditi podatcima i za te podatke davati (skoro) savršene predikcije
- Ako je model **presložen**, previše će se prilagoditi podatcima na kojima je treniran, a davat će loše predikcije na neviđenim podatcima ⇒ **prenaučenost** (engl. *overfitting*)
- Prenaučen model je beskorisan jer loše generalizira!
- Prenaučenost je glavni problem strojnog učenja
- Kako bismo izbjegli prenaučenost, moramo ispitati koliko dobro će model raditi na neviđenim podatcima ⇒ metoda **unakrsne provjere** (engl. *cross-validation*)

Unakrsna provjera

- Budući da neviđeni primjeri nisu dostupni, dio primjera koje imamo izdvajamo da "glume" neviđene primjere
- Podjela skupa podataka na **skup za učenje** i **skup za testiranje** (tipično 70%–30%)



- Model učimo na skupu za učenje, a zatim pomoću tog modela radimo predikciju na skupu za testiranje i na tom skupu računamo **točnost** (ili **pogrešku**)
- Točnost/pogreška modela na skupu za testiranje upućuje na sposobnost generalizacije modela (model prethodno nije vidio podatke iz skupa za testiranje!)

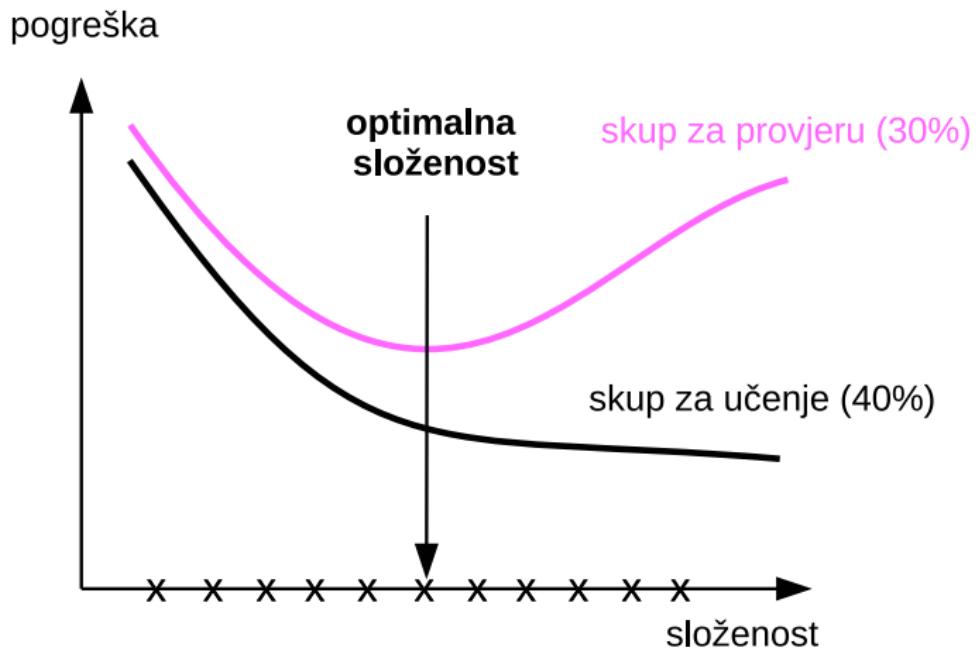
Unakrsna provjera

- Kod nekih je modela moguće ugađati njihovu složenost (npr. podrezivanje stabla odluke)
- Kako bismo odredili optimalnu složenost modela, treba nam treći skup podataka na kojem ćemo računati točnost/pogrešku modela:
skup za provjeru
- Podjela skupa podataka na **skup za učenje, skup za provjeru** (engl. *validation set*) i **skup za testiranje**, npr.:



- Modele različite složenosti treniramo na skupu za učenje i svaki ispitujemo na skupu za provjeru
- Nakon što smo odabrali optimalan model na skupu za provjeru, točnost/pogrešku tog modela izračunavamo na skupu za testiranje

Unakrsna provjera



Occamova britva (načelo parsimonije)



Occam's Razor: No more things should be presumed to exist than are absolutely necessary, i.e., the fewer assumptions an explanation of a phenomenon depends on, the better the explanation.

(William of Occam)

izquotes.com

Sadržaj

- 1 Uvod u strojno učenje
- 2 Pristupi strojnom učenju
- 3 Nadzirano učenje
- 4 Algoritam 1: Naivan Bayesov klasifikator
- 5 Algoritam 2: Stabla odluke

Bayesovo pravilo i Bayesov klasifikator

- Bayesov klasifikator je jednostavan algoritam nadziranoga strojnog učenja koji se temelji na **Bayesovom pravilu** (teoremu)



Thomas Bayes (1702–1761)

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$
A photograph of a chalkboard with the Bayes' Rule formula written in blue chalk. The formula is $P(A|B) = \frac{P(B|A) P(A)}{P(B)}$. The board also has some faint blue markings above and below the formula.

- Bayesov klasifikator zasniva se na istim idejama o kojima smo pričali u kontekstu modeliranja neizvjesnog znanja (dokazi doprinose aposteriornoj vjerojatnosti hipoteze), ali sada **vjerojatnosti želimo dobiti automatski iz podataka**

Podsjetnik: Bayesovo pravilo

- Bayesovo pravilo (vjerojatnost hipoteze H uz dokaz E):

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} = \frac{P(E|H)P(H)}{P(E|H)P(H) + P(E|\neg H)P(\neg H)}$$

- Za više hipoteza (koje čine potpun sustav događaja):

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{\sum_j P(E|H_j)P(H_j)}$$

- Za više hipoteza i više (n) dokaza:

$$P(H_i|E_1, E_2, \dots, E_n) = \frac{P(E_1, E_2, \dots, E_n|H_i)P(H_i)}{\sum_j P(E_1, E_2, \dots, E_n|H_j)P(H_j)}$$

- Uz pretpostavku o **uvjetnoj nezavisnosti** dokaza uz danu hipotezu:

$$P(H_i|E_1, E_2, \dots, E_n) = \frac{P(E_1|H_i)P(E_2|H_i) \cdots P(E_n|H_i)P(H_i)}{\sum_j P(E_1|H_j)P(E_2|H_j) \cdots P(E_n|H_j)P(H_j)}$$

Bayesovo pravilo kao klasifikacijski model

- Bayesovo pravilo izravno se može upotrijebiti kao klasifikacijski model
 - ▶ Hipoteza $H \rightarrow$ **klasa y**
 - ▶ Dokazi $E_1, \dots, E_n \rightarrow$ **primjer $\mathbf{x} = (x_1, \dots, x_n)$**
- Model:

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\sum_{y'} P(\mathbf{x}|y')P(y')}$$

gdje

- ▶ $P(y)$ je **apriorna vjerojatnost klase** (engl. *prior*)
- ▶ $P(\mathbf{x}|y)$ je **izglednost klase** (engl. *class likelihood*)
- ▶ $P(y|\mathbf{x})$ je **aposteriorna vjerojatnost klase** (engl. *posterior*)
- Nas zanima aposteriorna vjerojatnost $P(y|\mathbf{x})$: **koja je vjerojatnost da primjer \mathbf{x} pripada klasi y** (odnosno da ima dotičnu oznaku)
- Primijetite da je to suprotno od izglednosti klase $P(\mathbf{x}|y)$: koja je vjerojatnost da klasa y ima primjer \mathbf{x}

MAP-hipoteza

- $P(y|\mathbf{x})$ nam daje vjerojanost da primjer \mathbf{x} pripada klasi y
- To je korisno, no često nam je dovoljno da samo klasificiramo primjer
- Optimalna klasifikacijska odluka je da primjer klasificiramo u klasu sa **najvećom aposteriornom vjerojatnošću**
- Takvu klasifikacijsku odluku nazivamo **hipoteza maksimum aposteriori** ili **MAP-hipoteza**
- Nazivnik Bayesovog pravila za sve je klase y identičan, pa ga, ako nas zanima maksimum, možemo zanemariti i računati samo brojnik
- Model Bayesovog klasifikatora sa MAP-hipotezom:

$$h_{\text{MAP}} = \operatorname{argmax}_y P(y|\mathbf{x}) = \operatorname{argmax}_y P(\mathbf{x}|y)P(y)$$

gdje funkcija $\operatorname{argmax}_x f(x)$ daje x koji maksimizira $f(x)$

- Hipoteza h_{MAP} je klasa (oznaka) kojoj primjer najvjerojatnije pripada

Naivan Bayesov klasifikator

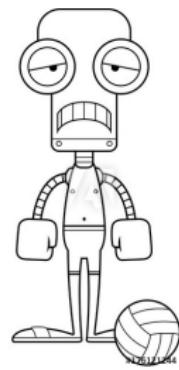
- Isto kao i u slučaju više dokaza, uvodimo **prepostavku uvjetne nezavisnosti**: značajke x_j su uvjetno nezavisne za danu klasu y , tj. vrijedi $P(x_j, x_k|y) = P(x_j|y)P(x_k|y)$
- Uz tu prepostavku, model Bayesovog klasifikatora je:

$$\begin{aligned} h_{\text{MAP}} &= \operatorname{argmax}_y P(y|\mathbf{x}) = \\ &= \operatorname{argmax}_y P(\mathbf{x}|y)P(y) \stackrel{u.n.}{=} \operatorname{argmax}_y P(y) \prod_{j=1}^n P(x_j|y) \end{aligned}$$

- Ovaj model nazivamo **naivan Bayesov klasifikator** (engl. *naïve Bayes classifier*)
- Model je “naivan” jer prepostavka o uvjetnoj nezavisnosti značajki općenito ne vrijedi (između značajki općenito postoje interakcije)
- Međutim, u praksi se pokazuje da model unatoč tome radi solidno

Primjer: Dan za odbojku na pjesku

Dan	Vrijeme	x ₁ Temp	x ₂ Vlažnost	x ₃ Vjetar	x ₄	y Odbojka?
1	sunčano	visoka	visoka	slab		ne
2	sunčano	visoka	visoka	jak		ne
3	oblačno	visoka	visoka	slab		da
4	kišno	srednja	visoka	slab		da
5	kišno	niska	normalna	slab		da
6	kišno	niska	normalna	jak		ne
7	oblačno	niska	normalna	jak		da
8	sunčano	srednja	visoka	slab		ne
9	sunčano	niska	normalna	slab		da
10	kišno	srednja	normalna	slab		da
11	sunčano	srednja	normalna	jak		da
12	oblačno	srednja	visoka	jak		da
13	oblačno	visoka	normalna	slab		da
14	kišno	srednja	visoka	jak		ne



Primjer: Dan za odbojku na pijesku

- Zanima nas klasifikacija za novi (petnaesti) primjer:

$$\mathbf{x} = (\underset{x_1}{\text{Vrijeme=sunčano}}, \underset{x_2}{\text{Temp=niska}}, \underset{x_3}{\text{Vlažnost=visoka}}, \underset{x_4}{\text{Vjetar=jak}})$$

- MAP-hipoteza je:

$$h_{\text{MAP}} = \operatorname{argmax}_{y \in \{\text{da}, \text{ne}\}} P(y|\mathbf{x})$$

$$\stackrel{u.n.}{=} \operatorname{argmax}_{y \in \{\text{da}, \text{ne}\}} P(y) \prod_{j=1}^n P(x_j|y)$$

$$= \operatorname{argmax}_{y \in \{\text{da}, \text{ne}\}} P(y)P(\text{sunčano}|y)P(\text{niska}|y)P(\text{visoka}|y)P(\text{jak}|y)$$

- Dakle, trebamo izračunati:

$$y = \text{da} : P(\text{da})P(\text{sunčano}|\text{da})P(\text{niska}|\text{da})P(\text{visoka}|\text{da})P(\text{jak}|\text{da})$$

$$y = \text{ne} : P(\text{ne})P(\text{sunčano}|\text{ne})P(\text{niska}|\text{ne})P(\text{visoka}|\text{ne})P(\text{jak}|\text{ne})$$

Učenje Bayesovog klasifikatora

- Učenje (treniranje) Bayesovog klasifikatora svodi se na **procjenu vjerojatnosti** na temelju skupa primjera za učenje
- **Q:** Kako bismo procijenili **apriorne vjerojatnosti** $P(\text{da})$ i $P(\text{ne})$?
- Najjednostavnije: **relativna frekvencija** (udio primjera u skupu za učenje koji imaju oznaku $y = \text{da}$ odnosno oznaku $y = \text{ne}$)

$$P(\text{da}) = \frac{9}{14} = 0.64$$

$$P(\text{ne}) = \frac{5}{14} = 0.36$$

- Slično možemo procijeniti **izglednosti**. Npr. $P(\text{sunčano}|\text{da})$ je udio primjera sa $x_1 = \text{sunčano}$ u podskupu primjera sa $y = \text{da}$

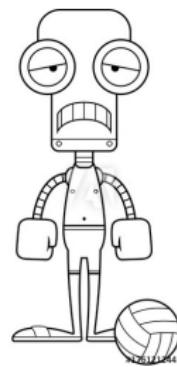
$$P(\text{sunčano}|\text{da}) = \frac{2}{9} = 0.22$$

$$P(\text{sunčano}|\text{ne}) = \frac{3}{5} = 0.6$$

Primjer: Dan za odbojku na pijesku

Dan	x_1 Vrijeme	x_2 Temp	x_3 Vlažnost	x_4 Vjetar	y Odbojka?
1	sunčano	visoka	visoka	slab	ne
2	sunčano	visoka	visoka	jak	ne
3	oblačno	visoka	visoka	slab	da
4	kišno	srednja	visoka	slab	da
5	kišno	niska	normalna	slab	da
6	kišno	niska	normalna	jak	ne
7	oblačno	niska	normalna	jak	da
8	sunčano	srednja	visoka	slab	ne
9	sunčano	niska	normalna	slab	da
10	kišno	srednja	normalna	slab	da
11	sunčano	srednja	normalna	jak	da
12	oblačno	srednja	visoka	jak	da
13	oblačno	visoka	normalna	slab	da
14	kišno	srednja	visoka	jak	ne

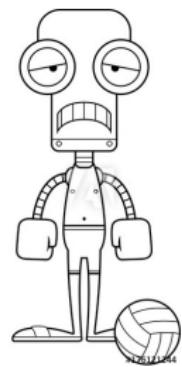
$$P(\text{da}) = \frac{9}{14}$$



Primjer: Dan za odbojku na pjesku

Dan	x_1 Vrijeme	x_2 Temp	x_3 Vlažnost	x_4 Vjetar	y Odbojka?
1	sunčano	visoka	visoka	slab	ne
2	sunčano	visoka	visoka	jak	ne
3	oblačno	visoka	visoka	slab	da
4	kišno	srednja	visoka	slab	da
5	kišno	niska	normalna	slab	da
6	kišno	niska	normalna	jak	ne
7	oblačno	niska	normalna	jak	da
8	sunčano	srednja	visoka	slab	ne
9	sunčano	niska	normalna	slab	da
10	kišno	srednja	normalna	slab	da
11	sunčano	srednja	normalna	jak	da
12	oblačno	srednja	visoka	jak	da
13	oblačno	visoka	normalna	slab	da
14	kišno	srednja	visoka	jak	ne

$$P(\text{sunčano}|\text{da}) = \frac{2}{9}$$



Primjer: Dan za odbojku na pijesku

- Izračun svih potrebnih vjerojatnosti:

$$P(y = \text{da}) = 9/14 = 0.64$$

$$P(y = \text{ne}) = 5/14 = 0.36$$

$$P(x_1 = \text{sunčano}|y = \text{da}) = 2/9 = 0.22$$

$$P(x_1 = \text{sunčano}|y = \text{ne}) = 3/5 = 0.6$$

$$P(x_2 = \text{niska}|y = \text{da}) = 3/9 = 0.33$$

$$P(x_2 = \text{niska}|y = \text{ne}) = 1/5 = 0.2$$

$$P(x_3 = \text{visoka}|y = \text{da}) = 3/9 = 0.33$$

$$P(x_3 = \text{visoka}|y = \text{ne}) = 4/5 = 0.8$$

$$P(x_4 = \text{jak}|y = \text{da}) = 3/9 = 0.33$$

$$P(x_4 = \text{jak}|y = \text{ne}) = 3/5 = 0.6$$

- Izračun MAP-hipoteze:

$$\begin{aligned} y = \text{da} : & P(\text{da})P(\text{sunčano}|\text{da})P(\text{niska}|\text{da})P(\text{visoka}|\text{da})P(\text{jak}|\text{da}) = \\ & 0.64 \cdot 0.22 \cdot 0.33 \cdot 0.33 \cdot 0.33 = 0.0051 \end{aligned}$$

$$\begin{aligned} y = \text{ne} : & P(\text{ne})P(\text{sunčano}|\text{ne})P(\text{niska}|\text{ne})P(\text{visoka}|\text{ne})P(\text{jak}|\text{ne}) = \\ & 0.36 \cdot 0.6 \cdot 0.2 \cdot 0.8 \cdot 0.6 = 0.0207 \end{aligned}$$

$$\Rightarrow h_{\text{MAP}} = \text{ne}$$

- Ako želimo, možemo izračunati i aposteriorne vjerojatnosti klasi:

$$P(\text{da}|\mathbf{x}) = \frac{0.0051}{0.0051 + 0.0207} = 0.198 \quad P(\text{ne}|\mathbf{x}) = \frac{0.0207}{0.0051 + 0.0207} = 0.802$$

Procjena najveće izglednosti

- Procjena vjerojatnosti kao relativnih frekvencija naziva se **procjena najveće izglednosti** (engl. *maximum likelihood estimation, MLE*), jer takva procjena maksimizira izglednost parametara modela (vjerojatnost podataka pod modelom)

Procjena najveće izglednosti (MLE)

Procjene najveće izglednosti za apriorne vjerojatnosti klasa i izglednosti klasa Bayesovog klasifikatora jesu:

$$P(y = v) = \frac{|D_{y=v}|}{|D|}$$

$$P(x_j = w | y = v) = \frac{|D_{y=v \wedge x_j=w}|}{|D_{y=v}|}$$

gdje je D skup primjera, a D_P je podskup primjera koji zadovoljavaju P

Prenaučenost Bayesovog klasifikatora

- MLE je najjednostavnija procjena parametara (vjerojatnosti Bayesovog klasifikatora), međutim podložna je **prenaučenosti**
- **Prenaučenost** \Leftrightarrow **model se previše prilagodio skupu za učenje**
- Kod Bayesovog klasifikatora: izglednosti klase za kombinacije koje se nisu pojavile u skupu primjera za učenje jednake su nuli
- Npr., želimo klasificirati primjer:

$$\mathbf{x} = (\underset{x_1}{\text{Vrijeme=oblačno}}, \underset{x_2}{\text{Temp=niska}}, \underset{x_3}{\text{Vlažnost=visoka}}, \underset{x_4}{\text{Vjetar=jak}})$$

- U skupu za učenje nema primjera sa Vrijeme=oblačno i $y = \text{ne}$, pa:

$$P(\text{oblačno}|\text{ne}) = \frac{0}{5} = 0$$

i posljedično:

$$\begin{aligned}y = \text{ne} : P(\text{ne})P(\text{oblačno}|\text{ne})P(\text{niska}|\text{ne})P(\text{visoka}|\text{ne})P(\text{jak}|\text{ne}) = \\0.36 \cdot 0.0 \cdot 0.2 \cdot 0.8 \cdot 0.6 = 0.0\end{aligned}$$

Zaglađivanje

- Premda se neka kombinacija značajke i oznake nije pojavila u skupu za učenje, ne želimo tu kombinaciju proglašiti potpuno nemogućom
- Rješenje je da malo preraspodijelimo masu vjerojatnosti s kombinacijama koje smo opazili na kombinacije koje nismo opazili
- Takav postupak zove se **zaglađivanje** (engl. *smoothing*)
- Najjednostavniji postupak zaglađivanja je **Laplaceovo zaglađivanje**

Procjene izglednosti klase s Laplaceovim zaglađivanjem

$$P(x_j = w | y = v) = \frac{|D_{y=v} \wedge x_j=w| + \alpha}{|D_{y=v}| + \alpha |V(x_j)|}$$

gdje je $V(x_j)$ skup mogućih vrijednosti značajke (atributa) x_j , a $\alpha \geq 0$ je parametar zaglađivanja

- Odabir $\alpha = 0$ daje nezaglađenu procjenu, a $\alpha = 1$ daje zaglađivanje **“dodaj jedan”** (engl. *add-one smoothing*)

Zaglađivanje “dodaj jedan” – primjer

- Distribucija vjerojatnosti bez zaglađivanja:

$$P(\text{sunčano}|\text{ne}) = \frac{3}{5} = 0.6$$

$$P(\text{oblačno}|\text{ne}) = \frac{0}{5} = 0$$

$$P(\text{kišno}|\text{ne}) = \frac{2}{5} = 0.4$$

- Laplaceovo zaglađivanje uz $\alpha = 1$ (pri čemu $|V(\text{Vrijeme})| = 3$):

$$P(\text{sunčano}|\text{ne}) = \frac{3+1}{5+3} = 0.5$$

$$P(\text{oblačno}|\text{ne}) = \frac{0+1}{5+3} = 0.125$$

$$P(\text{kišno}|\text{ne}) = \frac{2+1}{5+3} = 0.375$$

- ⇒ redistribucija vjerojatnosti s viđenih događaja na neviđene događaje
⇒ efekt zaglađivanja je to manji što imamo više primjera za učenje

Zaglađivanje “dodaj jedan” – primjer

$$\mathbf{x} = (\underset{x_1}{\text{Vrijeme=oblačno}}, \underset{x_2}{\text{Temp=niska}}, \underset{x_3}{\text{Vlažnost=visoka}}, \underset{x_4}{\text{Vjetar=jak}})$$

$$P(y = \text{da}) = 9/14 = 0.64$$

$$P(y = \text{ne}) = 5/14 = 0.36$$

$$P(x_1 = \text{oblačno}|y = \text{da}) = \frac{3+1}{9+3} = 0.33$$

$$P(x_1 = \text{oblačno}|y = \text{ne}) = \frac{0+1}{5+3} = 0.125$$

$$P(x_2 = \text{niska}|y = \text{da}) = \frac{3+1}{9+3} = 0.33$$

$$P(x_2 = \text{niska}|y = \text{ne}) = \frac{1+1}{5+3} = 0.25$$

$$P(x_3 = \text{visoka}|y = \text{da}) = \frac{3+1}{9+2} = 0.36$$

$$P(x_3 = \text{visoka}|y = \text{ne}) = \frac{4+1}{5+2} = 0.71$$

$$P(x_4 = \text{jak}|y = \text{da}) = \frac{3+1}{9+2} = 0.36$$

$$P(x_4 = \text{jak}|y = \text{ne}) = \frac{3+1}{5+2} = 0.57$$

$$y = \text{da} : P(\text{da})P(\text{oblačno}|\text{da})P(\text{niska}|\text{da})P(\text{visoka}|\text{da})P(\text{jak}|\text{da}) = \\ 0.64 \cdot 0.33 \cdot 0.33 \cdot 0.36 \cdot 0.36 = 0.009$$

$$y = \text{ne} : P(\text{ne})P(\text{oblačno}|\text{ne})P(\text{niska}|\text{ne})P(\text{visoka}|\text{ne})P(\text{jak}|\text{ne}) = \\ 0.36 \cdot 0.125 \cdot 0.25 \cdot 0.71 \cdot 0.57 = 0.005$$

$$\Rightarrow h_{\text{MAP}} = \text{da}$$

Sprječavanje podljeva

- Kod izračuna h_{MAP} , množimo izglednosti u brojniku Bayesovog pravila

$$h_{\text{MAP}} = \underset{y}{\operatorname{argmax}} P(y) \prod_{j=1}^n P(x_j|y)$$

- Ako imamo mnogo značajki, ovo može lako dovesti do podljeva
- Kako bismo to spriječili, računamo u **logaritamskoj domeni** (logaritam je monotono rastuća funkcija, pa nema utjecaja na funkciju argmax)

$$\begin{aligned} h_{\text{MAP}} &= \ln \left(\underset{y}{\operatorname{argmax}} P(y) \prod_{j=1}^n P(x_j|y) \right) \\ &= \underset{y}{\operatorname{argmax}} \ln \left(P(y) \prod_{j=1}^n P(x_j|y) \right) \\ &= \underset{y}{\operatorname{argmax}} \left(\ln P(y) + \sum_{j=1}^n \ln P(x_j|y) \right) \end{aligned}$$

Naivan Bayesov klasifikator – napomene

- **Prednosti:**

- ▶ jednostavan i brz algoritam (lako je trenirati model)
- ▶ može raditi s viševrijednosnim značajkama i sa više klase odjednom
- ▶ radi dobro s velikim brojem značajki (linearan je u broju značajki)
- ▶ daje dobre predikcije ako vrijede uvjetne nezavisnosti značajki

- **Nedostatci:**

- ▶ **u stvarnosti, značajke su rijetko uvjetno nezavisne i točnost klasifikacije je to lošija što su značajke više zavisne**

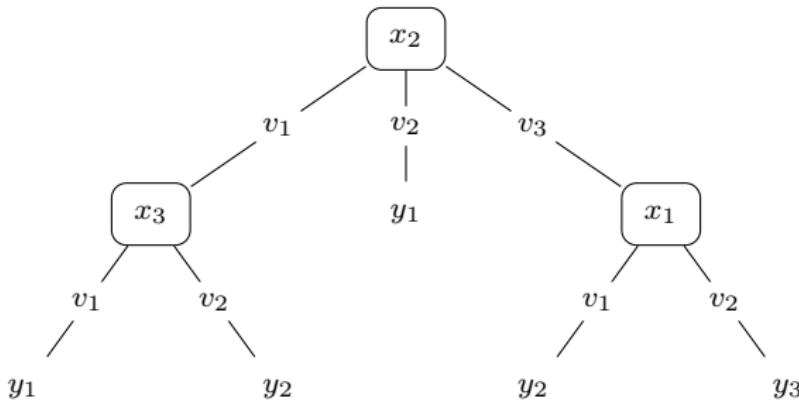
- Tipične primjene: klasifikacija dokumenata, filtriranje neželjene pošte
- Ako prepostavke o uvjetnoj nezavisnosti ne vrijede, može se koristiti **polunaivan Bayesov klasifikator** (složeniji model)
- Ako su značajke kontinuirane, za izglednosti klase koristimo Gaussove distribucije (**Gaussov Bayesov klasifikator**)
- Bayesov klasifikator pripada širokoj porodici modela koje nazivamo **probabilistički grafički modeli**

Sadržaj

- 1 Uvod u strojno učenje
- 2 Pristupi strojnom učenju
- 3 Nadzirano učenje
- 4 Algoritam 1: Naivan Bayesov klasifikator
- 5 Algoritam 2: Stabla odluke

Klasifikacija stablom odluke

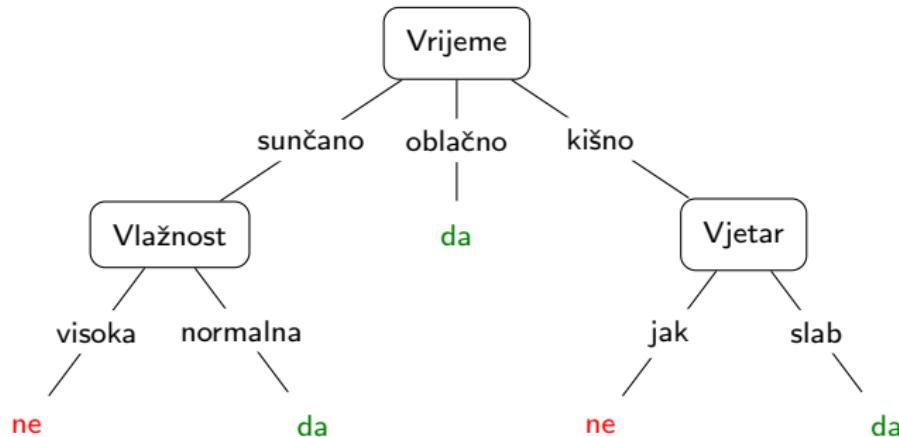
- **Unutarnji čvorovi** odgovaraju značajkama (atributima), **grane** ispod svakog čvora odgovaraju vrijednostima dotične značajke, a **listovi** odgovaraju klasifikacijskim odlukama (oznakama klase)



- Primjer se klasificira slijednim ispitivanjem vrijednosti značajki, krenuvši od korijena (vrha) stabla prema dnu (listovima)
- Kada se dosegne list, primjer se klasificira oznakom lista

Stablo odluke – primjer

- Stablo odluke za skup primjera “Dan za odbojku na pijesku”:



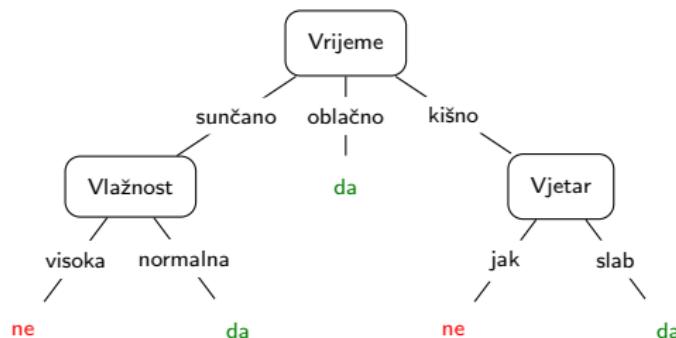
- Klasifikacija novog primjera:

$$\mathbf{x} = (\underset{x_1}{\text{Vrijeme=sunčano}}, \underset{x_2}{\text{Temp=niska}}, \underset{x_3}{\text{Vlažnost=visoka}}, \underset{x_4}{\text{Vjetar=jak}})$$

⇒ **ne**

Stabla odluke i klasifikacija ne temelju pravila

- Stabla odluke povezana su s **klasifikacijom na temelju pravila** (engl. *rule-based classification*)
- Stablo odluke može se smatrati sažetim načinom zapisa skupa pravila
- Svaka staza od korijena do lista daje jedno pravilo u obliku konjunkcije uvjeta nad vrijednostima značajki
- Skup pravila za istu oznaku klase je disjunkcija konjunkcija uvjeta


$$((\text{Vrijeme}=\text{sunčano} \wedge \text{Vlažnost}=\text{normalna}) \vee (\text{Vrijeme}=\text{oblačno}) \vee (\text{Vrijeme}=\text{kišno} \wedge \text{Vjetar}=\text{jak})) \rightarrow (y = \text{da})$$

Algoritam ID3

- **Algoritam ID3 (Iterative Dichotomiser 3)** jednostavan je algoritam za izgradnju stabla odluke koji je predložio Ross Quinlan 1986. godine



Machine Learning 1: 81–106, 1986
© 1986 Kluwer Academic Publishers, Boston – Manufactured in The Netherlands

Induction of Decision Trees

J.R. QUINLAN

*Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Sydney 2007,
Australia*

(Received August 1, 1985)

Key words: classification, induction, decision trees, information theory, knowledge acquisition, expert systems

Abstract. The technology for building knowledge-based systems by inductive inference from examples has been demonstrated successfully in several practical applications. This paper summarizes an approach to synthesizing decision trees that has been used in a variety of systems, and it describes one such system, ID3, in detail. Results from recent studies show ways in which the methodology can be modified to deal with information that is noisy and/or incomplete. A reported shortcoming of the basic algorithm is discussed and two means of overcoming it are compared. The paper concludes with illustrations of current research directions.

Quinlan, J. R. (1986). **Induction of decision trees.** Machine learning, 1(1), 81-106.

Izgradnja stabla odluke

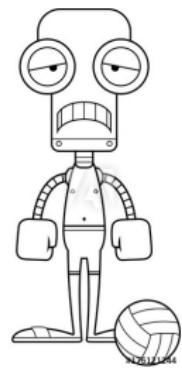
- Klasifikacija stablom odluke je jednostavna, no glavno pitanje je **kako izgraditi stablo odluke** na temelju podataka
- Svako stablo odluke **rekurzivno particionira** skup primjera za učenje
- Za svaki skup za primjera učenje postoji konačan broj mogućih stabala odluke koje savršeno klasificiraju sve primjere iz skupa za učenje (uz pretpostavku da ne postoje identični primjeri s različitim oznakama)
- Ako postoji puno značajki i njihovih vrijednosti, broj mogućih stabala može biti vrlo velik
- U praksi preferiramo stabla sa što manje čvorova: jednostavnija stabla odgovaraju jednostavnijim hipotezama koje **bolje generaliziraju** odnosno koje su otpornije na **prenaučenost**
- Međutim, nalaženje minimalnog stabla koje ispravno klasificira sve primjere iz skupa za učenje je **NP-težak problem** ⇒ treba nam heuristički pristup

Izgradnja stabla odluke

- Struktura stabla određena je time koju značajku ispitujemo u kojem čvoru
- U korijenskom čvoru, biramo između n značajki, u čvorovima prve razine biramo između $n - 1$ značajki, itd.
- Treba nam (numerički) **kriterij za odabir najpovoljnije značajke** po kojoj radimo podjelu u trenutačnom čvoru
- Intuitivno, preferiramo značajke koje **što bolje diskriminira** između primjera za učenje prema ciljnoj oznaci y (idealno tako da svi primjeri za koje značajka ima određenu vrijednost pripadaju istoj klasi)

Primjer: Dan za odbojku na pjesku

Dan	Vrijeme	x ₁ Temp	x ₂ Vlažnost	x ₃ Vjetar	x ₄	y Odbojka?
1	sunčano	visoka	visoka	slab		ne
2	sunčano	visoka	visoka	jak		ne
3	oblačno	visoka	visoka	slab		da
4	kišno	srednja	visoka	slab		da
5	kišno	niska	normalna	slab		da
6	kišno	niska	normalna	jak		ne
7	oblačno	niska	normalna	jak		da
8	sunčano	srednja	visoka	slab		ne
9	sunčano	niska	normalna	slab		da
10	kišno	srednja	normalna	slab		da
11	sunčano	srednja	normalna	jak		da
12	oblačno	srednja	visoka	jak		da
13	oblačno	visoka	normalna	slab		da
14	kišno	srednja	visoka	jak		ne



Odabir značajke za čvor – primjer

Značajka	Vrijednost	# da	# ne
Vrijeme	sunčano	2	3
	oblačno	4	0
	kišno	3	2
Temp	visoka	2	2
	srednja	4	2
	niska	3	1
Vlažnost	visoka	3	4
	normalna	6	1
Vjetar	jak	3	3
	slab	6	2

- Koju značajku odabrati za korijenski čvor?
- Podjela po **Vrijeme** bi mogla biti dobra jer Vrijeme=oblačno daje samo pozitivne primjere, međutim Vrijeme=sunčano i Vrijeme=kišno gotovo da i ne diskriminira između pozitivnih i negativnih primjera

Entropija skupa primjera

- Najbolja podjela je po značajki za koju svi primjeri u svakoj grani pripadaju istoj klasi
- Najlošija podjela je po značajki za koju su u svakoj grani primjeri razdijeljeni uniformo po različitim klasama
- Ovaj kriterij možemo iskazati kao **minimizaciju entropije** oznaka

Entropija skupa primjera

Neka skup D sadrži označene primjere iz K klasa. Entropija skupa D je:

$$E(D) = - \sum_{i=1}^K P(y = i) \log_2 P(y = i)$$

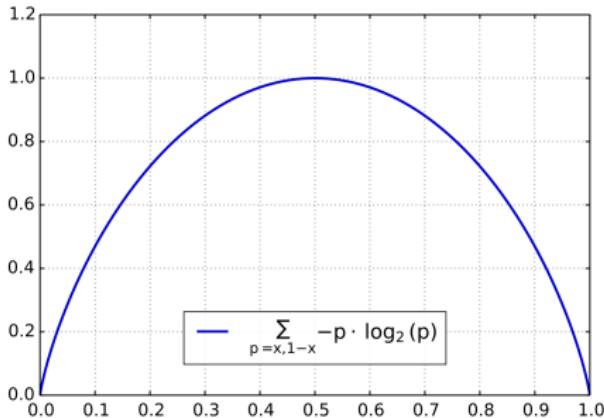
gdje je $P(y = i)$ vjerojatnost klase $y = i$. Dogovorno, $0 \log_2 0 \doteq 0$.

- savršena podjela ima $E = 0$, dok nediskriminativna podjela ima $E = \log_2(K)$

Entropija skupa primjera

- Za binarnu klasifikaciju ($K = 2$):

$$E(D) = - \sum_{y=\{\text{da},\text{ne}\}} P(y) \log_2 P(y)$$



- Najbolja podjela ima $E = 0$, a najlošija $E = 1$

Entropija skupa primjera – primjer

Značajka	Vrijednost	# da	# ne	$E(D)$
Vrijeme	sunčano	2	3	0.971
	oblačno	4	0	0
	kišno	3	2	0.971
Temp	visoka	2	2	1
	srednja	4	2	0.918
	niska	3	1	0.818
Vlažnost	visoka	3	4	0.985
	normalna	6	1	0.592
Vjetar	jak	3	3	1
	slab	6	2	0.811

- Značajka **Vrijeme** ostvaruje najnižu entropiju za vrijednost "oblačno", no treba uzeti u obzir i njezine druge dvije vrijednosti
- Rijetko će entropija značajke odmah biti nula za sve njene vrijednosti
- Odabiremo značajku koja **što više smanjuje entropiju**, a tek dalnjom podjelom u nižim čvorovima svest ćemo entropiju na nulu

Informacijska dobit

- Kriterij **informacijske dobiti** (engl. *information gain*) mjeri očekivano smanjenje entropije skupa primjera uslijed podjele primjera po vrijednostima neke značajke

Informacijska dobit

Informacijska dobit (IG) značajke x na skupu primjera D je:

$$IG(D, x) = \underbrace{E(D)}_{\substack{\text{entropija} \\ \text{početnog skupa}}} - \underbrace{\sum_{v \in V(x)} \frac{|D_{x=v}|}{|D|} E(D_{x=v})}_{\substack{\text{očekivana vrijednost entropije} \\ \text{nakon podjele skupa na temelju značajke}}}$$

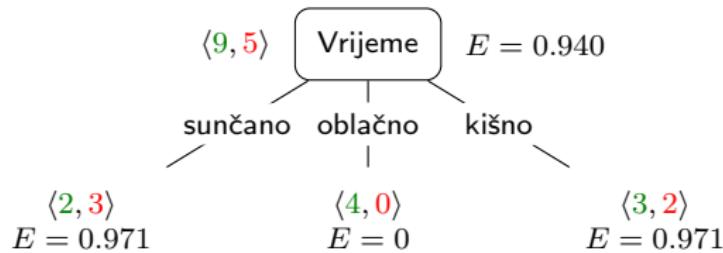
gdje je $E(D)$ entropija skupa primjera D , D_P je podskup primjera koji zadovoljavaju uvjet P , a $V(x)$ je skup mogućih vrijednosti značajke x .

Primjer: Dan za odbojku na pijesku

- Označimo sa $\langle m, n \rangle$ skup D koji ima m pozitivnih primjera ($y = \text{da}$) i n negativnih primjera ($y = \text{ne}$)
- Entropija početnog skupa D je:

$$E(D) = E(\langle 9, 5 \rangle) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940$$

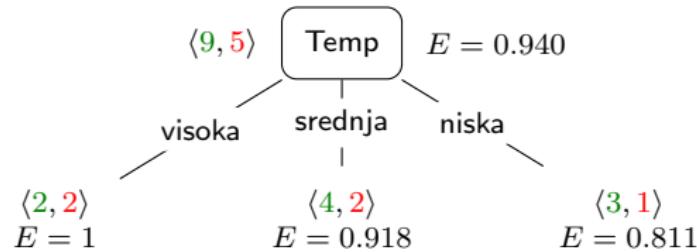
- Tražimo najbolju značajku (prema IG) za korijenski čvor stabla odluke
- Računamo IG za značajku **Vrijeme** na skupu D :



$$IG(D, \text{Vrijeme}) = 0.940 - \frac{5}{14} \cdot 0.971 - \frac{4}{14} \cdot 0 - \frac{5}{14} \cdot 0.971 = 0.246$$

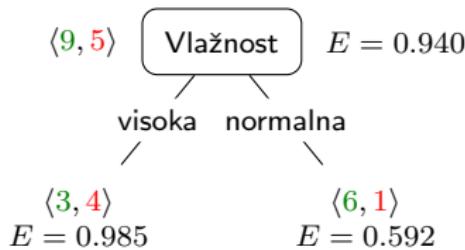
Primjer: Dan za odbojku na pijesku

- IG za značajku **Temp** na skupu D :



$$IG(D, \text{Temp}) = 0.940 - \frac{4}{14} \cdot 1 - \frac{6}{14} \cdot 0.918 - \frac{4}{14} \cdot 0.811 = 0.029$$

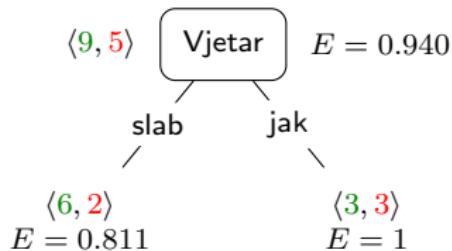
- IG za značajku **Vlažnost** na skupu D :



$$IG(D, \text{Vlažnost}) = 0.940 - \frac{7}{14} \cdot 0.985 - \frac{7}{14} \cdot 0.592 = 0.151$$

Primjer: Dan za odbojku na pijesku

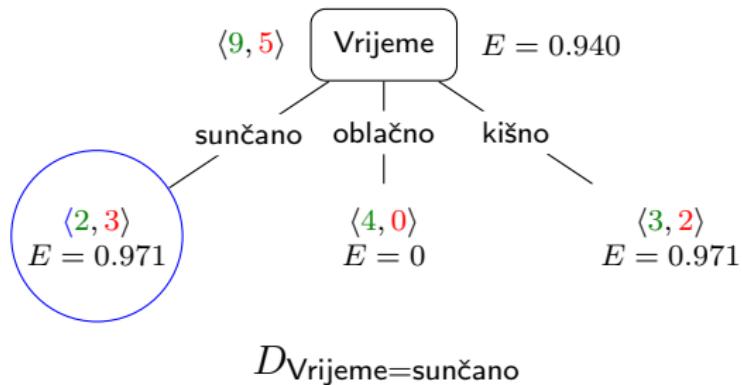
- IG za značajku **Vjetar** na skupu D :



$$IG(D, Vjetar) = 0.940 - \frac{8}{14} \cdot 0.811 - \frac{6}{14} \cdot 1 = 0.048$$

- Informacijska dobit (očekivana redukcija entropije) najveća je za značajku **Vrijeme**, pa tu značajku odabiremo za korijen našeg stabla
- Rekurzivno ponavljamo postupak na podskupovima podataka svake od triju izlaznih grana iz čvora **Vrijeme**, dodajući čvorove-djecu i odabirući između preostale tri značajke

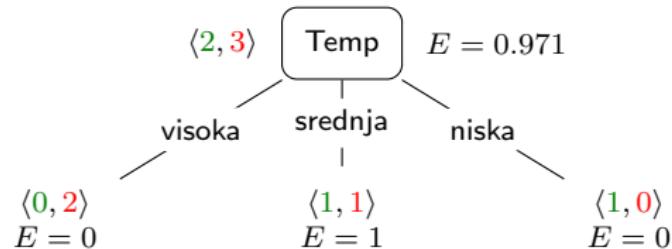
Primjer: Dan za odbojku na pijesku



Dan	x_2 Temp	x_3 Vlažnost	x_4 Vjetar	y Odbojka?
1	visoka	visoka	slab	ne
2	visoka	visoka	jak	ne
8	srednja	visoka	slab	ne
9	niska	normalna	slab	da
11	srednja	normalna	jak	da

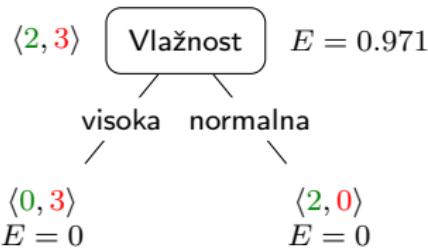
Primjer: Dan za odbojku na pijesku

- IG za značajku **Temp** na podskupu $D_{\text{Vrijeme}=\text{sunčano}}$:



$$IG(D_{\text{Vrijeme}=\text{sunčano}}, \text{Temp}) = 0.971 - \frac{2}{5} \cdot 0 - \frac{2}{5} \cdot 1 - \frac{1}{5} \cdot 0 = 0.571$$

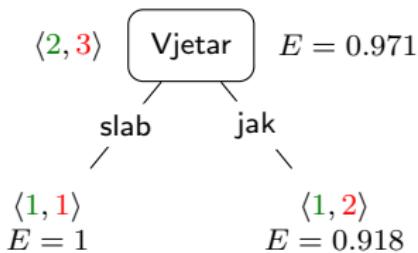
- IG za značajku **Vlažnost** na podskupu D :



$$IG(D_{\text{Vrijeme}=\text{sunčano}}, \text{Vlažnost}) = 0.971 - \frac{2}{5} \cdot 0 - \frac{2}{5} \cdot 0 = 0.971$$

Primjer: Dan za odbojku na pijesku

- IG za značajku **Vjetar** na podskupu $D_{Vrijeme=\text{sunčano}}$:

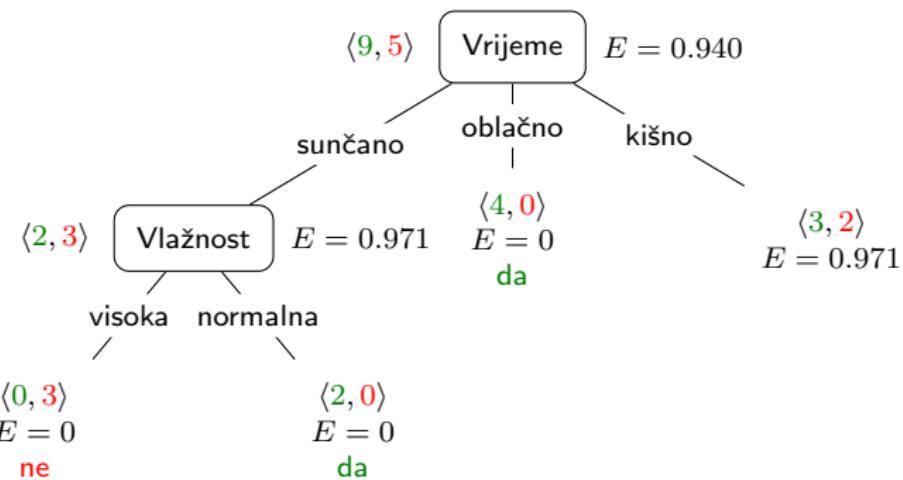


$$IG(D_{Vrijeme=\text{sunčano}}, \text{Vjetar}) = 0.971 - \frac{2}{5} \cdot 1 - \frac{3}{5} \cdot 0.918 = 0.02$$

- Informacijska dobit (očekivana redukcija u entropiji) najveća je za značajku **Vlažnost**, pa tu značajku odabiremo za čvor ispod grane $Vrijeme=\text{sunčano}$

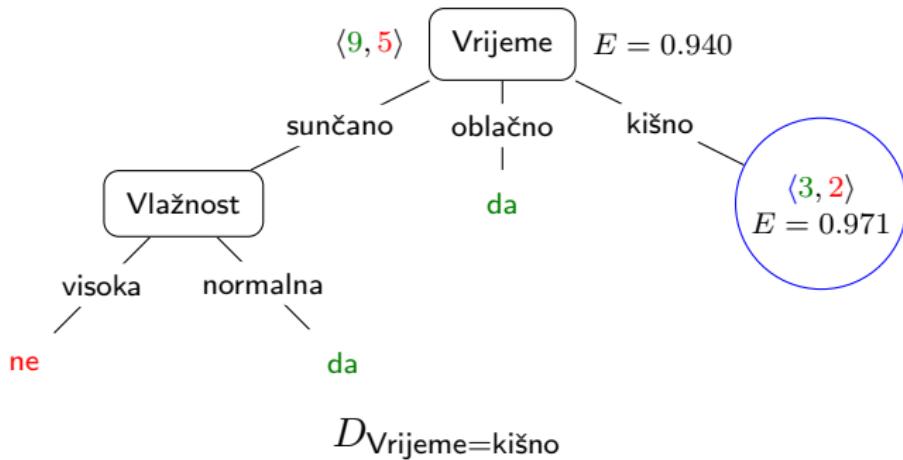
Primjer: Dan za odbojku na pijesku

- Stablo odluke nakon dodavanja drugog čvora:



- Čvorovi s podskupovima primjera $\langle m, 0 \rangle$ i $\langle 0, n \rangle$ imaju $E = 0$ te savršeno diskriminiraju između primjera, pa ih možemo zamijeniti listom s oznakom **da** odnosno **ne**

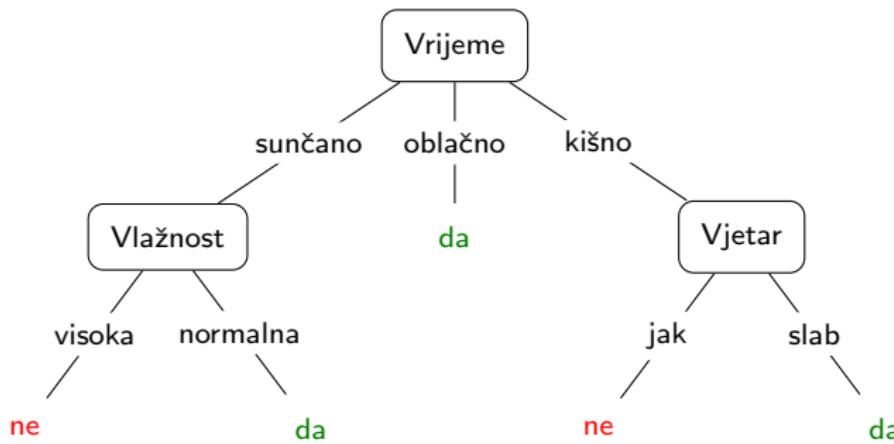
Primjer: Dan za odbojku na pjesku



Dan	x_2 Temp	x_3 Vlažnost	x_4 Vjetar	y Odbojka?
4	srednja	visoka	slab	da
5	niska	normalna	slab	da
6	niska	normalna	jak	ne
10	srednja	normalna	slab	da
14	srednja	visoka	jak	ne

Primjer: Dan za odbojku na pijesku

- Izračunom informacijske dobiti za preostale tri značajke nad podskupom $D_{Vrijeme=kišno}$ utvrđujemo da značajka **Vjetar** ima najveću informacijsku dobit
- Konačno stablo odluke izgleda ovako:



Algoritam ID3

```
function id3( $D, D^{parent}, X, y$ ) -- početno  $D^{parent} = D$ 
if  $D = \emptyset$  then
     $v \leftarrow \operatorname{argmax}_v |D_{y=v}^{parent}|$       -- najčešća oznaka primjera u nadčvoru
    return Leaf( $v$ )
if  $X = \emptyset$  or  $D = D_{y=v}$  then
     $v \leftarrow \operatorname{argmax}_v |D_{y=v}|$       -- najčešća oznaka primjera u čvoru
    return Leaf( $v$ )
 $x \leftarrow \operatorname{argmax}_{x \in X} IG(D, x)$       -- najdiskriminativnija značajka
 $subtrees \leftarrow \emptyset$ 
for  $v \in V(x)$ :
     $t \leftarrow id3(D_{x=v}, D, X \setminus \{x\}, y)$ 
     $subtrees \leftarrow \operatorname{append}(subtrees, (v, t))$ 
return Node( $x, subtrees$ )
```

X – skup svih značajki, y – oznaka klase, D – skup označenih primjera

Algoritam ID3 – primjer

- Početni poziv:

$$\text{id3}(D, D, \{x_1, x_2, x_3, x_4\}, y)$$

- Nakon prve iteracije:

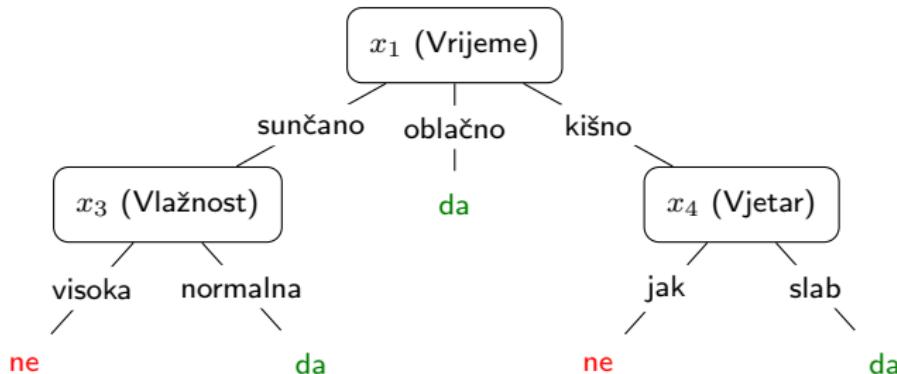
$$\text{Node}\left(x_1, \left\{ \begin{array}{l} (\text{sunčano}, \text{id3}(D_{x_1=\text{sunčano}}, D, \{x_2, x_3, x_4\}, y)), \\ (\text{oblačno}, \text{id3}(D_{x_1=\text{oblačno}}, D, \{x_2, x_3, x_4\}, y)), \\ (\text{kišno}, \text{id3}(D_{x_1=\text{kišno}}, D, \{x_2, x_3, x_4\}, y)) \end{array} \right\} \right)$$

- Nakon druge iteracije:

$$\text{Node}\left(x_1, \left\{ \begin{array}{l} (\text{sunčano}, \text{Node}\left(x_3, \left\{ \begin{array}{l} (\text{visoka}, \text{id3}(D_{x_1=\text{sunčano}}, D_{x_1=\text{sunčano}}, \{x_2, x_4\}, y)), \\ (\text{normalna}, \text{id3}(D_{x_1=\text{sunčano}}, D_{x_1=\text{sunčano}}, \{x_2, x_4\}, y)) \end{array} \right\} \right)), \\ (\text{oblačno}, \text{Leaf(da)}), \\ (\text{kišno}, \text{Node}\left(x_4, \left\{ \begin{array}{l} (\text{jak}, \text{id3}(D_{x_1=\text{sunčano}}, D_{x_1=\text{sunčano}}, \{x_2, x_3\}, y)), \\ (\text{slab}, \text{id3}(D_{x_1=\text{sunčano}}, D_{x_1=\text{sunčano}}, \{x_2, x_3\}, y)) \end{array} \right\} \right))) \end{array} \right\} \right)$$

Algoritam ID3 – primjer

- Nakon treće iteracije:

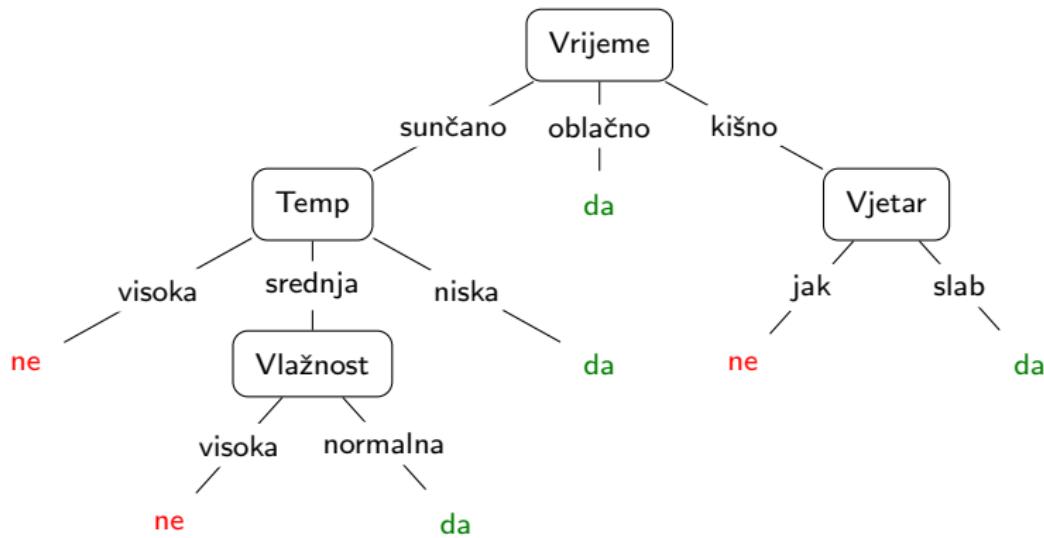
$$\text{Node}\left(x_1, \left\{ \begin{array}{l} (\text{sunčano}, \text{Node}(x_3, \left\{ \begin{array}{l} (\text{visoka}, \text{Leaf(ne)}), \\ (\text{normalna}, \text{Leaf(da)}) \end{array} \right\})), \\ (\text{oblačno}, \text{Leaf(da)}), \\ (\text{kišno}, \text{Node}(x_4, \left\{ \begin{array}{l} (\text{jak}, \text{Leaf(ne)}), \\ (\text{slab}, \text{Leaf(da)}) \end{array} \right\})) \end{array} \right\}) \right)$$


Prenaučenost stabla odluke

- Stablo koje konstruira algoritam ID3 uvijek će savršeno klasificirati sve primjere is skupa D (pod pretpostavkom da ne postoji više identičnih primjera s različitim oznakama)
- To lako dovodi do **prenaučenosti**: model savršeno klasificira primjere iz skupa za učenje, ali loše klasificira neviđene primjere \Rightarrow **loša generalizacija**
- Do prenaučenosti dolazi ako primjeri za učenje sadržavaju **šum** (pogrešne vrijednosti značajki ili pogrešnu oznaku klase)
- Npr., neka je skupu D dodan novi, petnaesti primjer:
$$x = (\underset{x_1}{\text{Vrijeme=sunčano}}, \underset{x_2}{\text{Temp=visoka}}, \underset{x_3}{\text{Vlažnost=normalna}}, \underset{x_4}{\text{Vjetar=jak}})$$
- Neka je ispravna oznaka ovog primjera $y = \text{da}$, međutim neka je zbog šuma oznaka $y = \text{ne}$
- Primijetite da stablo koje smo izgradili ispravno klasificira ovaj primjer

Prenaučenost stabla odluke

- Međutim, ako pogrešno označeni primjer prirodamo skupu za učenje, ID3 daje drugačije stablo odluke:

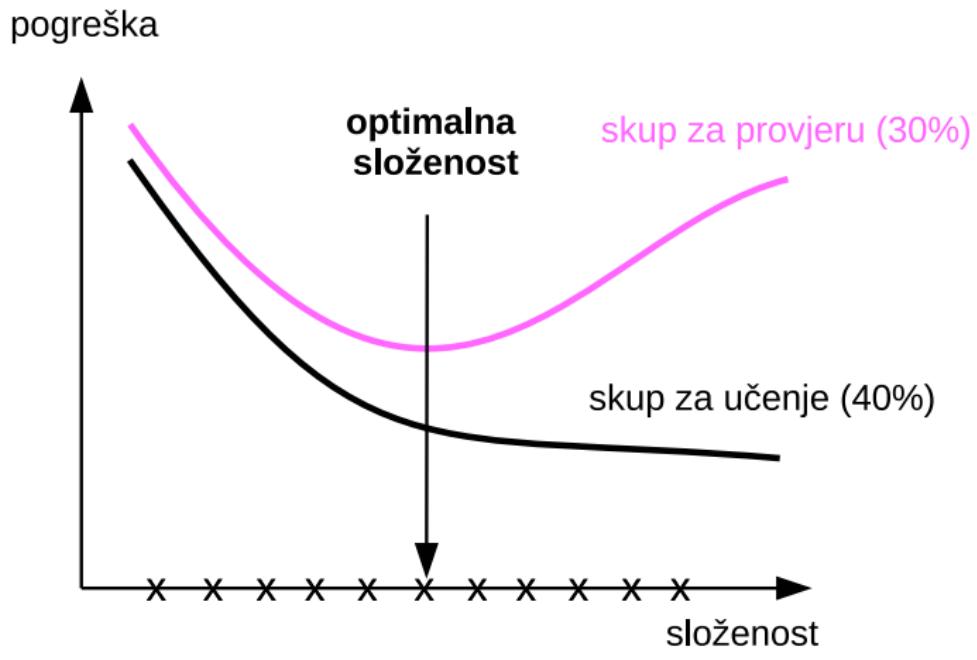


- Ovo stablo ispravno klasificira svih petnaest primjera prema oznakama u skupu za učenje, uključivo i pogrešno označeni petnaesti primjer

Prenaučenost stabla odluke

- Da pristupa za sprječavanje prenaučenosti:
 - ① **Ograničavanje rasta stabla** prije dosezanja savršene klasifikacije na skupu primjera za učenje
 - ② **Naknadno podrezivanje prenaučenog stabla**
 - ★ zamjena postabala listovima
 - ★ pretvorba stabla u ako-onda pravila te uklanjanje uvjeta u antecedentima pravila
- **Q:** Kako odrediti optimalnu veličinu stabla?
- **Intrinzični kriteriji** za zaustavljanje rasta stabla:
 - ▶ dosezanje unaprijed definirane maksimalne dubinu stabla
 - ▶ broj primjera u nekom čvoru je manji od unaprijed zadanog broja
 - ▶ pad entropije je manji od manji od unaprijed definiranog praga
- **Ekstrinzični kriterij** za zaustavljanje rasta ili podrezivanje stabla:
 - ▶ pad točnosti (porast pogreške) na **skupu primjera za provjeru**

Unakrsna provjera



Npr., složenost = dubina stabla

Stabla odluke – napomene

- Algoritam ID3 najjednostavniji je algoritam za izgradnju stabla odluke. Postoje komplikiranije alternative (npr. algoritam C4.5, QUEST)
- Osim informacijske dobiti, mogu se koristiti i drugi kriteriji za odabir čvora (npr. Gini indeks, χ^2 -test)
- Stabla odluke mogu se koristiti za numeričke značajke: podjela u čvoru radi se na temelju praga na vrijednost značajke
- Stabla odluke mogu se koristiti i za regresiju (a ne samo za klasifikaciju): **CART** (classification and regression trees)
- Za razliku od Bayesovog klasifikatora, stabla odluke...
 - ▶ ne daju vjerojatnost klasifikacijske odluke
 - ▶ je lako **interpretirati** (*black box* vs. *glass box* pristup strojnom učenju)
- Nedostatak: **sklonost prenaučenosti** (loša generalizacija), **visoka varijanca** (potpuno različita stabla za male razlike u skupu za učenje)
- Može se ublažiti primjenom ansambla stabala, npr. model **slučajne šume** (engl. *random forest*)

Sažetak

- Strojno učenje bavi se izgradnjom **modela** koji na temelju **podataka** mogu **predviđati** svojstva novih, još neviđenih podataka
- Glavne vrste strojnog učenja: **nadzirano**, **nenadzirano** i **podržano**
- Nadzirane modele **treniramo (učimo)** na primjerima s oznakama klase (**klasifikacija**) ili brojčanim oznakama (**regresija**)
- Modeli strojnog učenja skloni su **prenaučenosti**, što se može spriječiti **unakrsnom provjerom** na skupu za provjeru
- **Naivan Bayesov klasifikator** primjer klasificira u aposteriorno najvjerojatniju klasu (MAP-hipoteza), na temelju izglednosti klasa i apriorne vjerojatnosti klasa procijenjenih na temelju primjera za učenje
- **Stablo odluke** rekurzivno particionira skup primjera prema vrijednostima značajki, koje se kod **algoritma ID3** odabiru na temelju kriterija **informacijske dobiti**



Sljedeća tema: Umjetne neuronske mreže