Učno gradivo za predmet

Odkrivanje znanja v podatkih

Doc. dr. Petra Kralj Novak

Univerzitetni program I. stopnje: Računalništvo in spletne tehnologije
Študijsko leto 2018/2019, poletni semester

Kazalo

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Prvi sklop

Uvod in priprava podatkov

Odkrivanje znanja v podatkih: Motivacija

Tehnološki napredek na področju IKT omogoča shranjevanje čedalje večje količine podatkov na vseh področjih človeškega delovanja, npr.

- v poslovnem svetu (beleženje prodaje, beleženje delovanja strojev, zgodovina strank, obiski spletnih strani, marketinške kampanije, ankete, bančne transakcije…)
- v znanosti (sekvenciranje DNK, vremenski podatki s satelitov, veliki hadronski trkalnik v CERNu,…)
- na svetovnem spletu (spletne strani in povezave med njimi, socialni mediji, beleženja iskanj, nakupov,…)
- multimedijski podatki (slike, video, radiološke slike [CT, MRI]…)

Z analizo podatkov poskušamo v podatkih najti vzorce oziroma še neznane zakonitosti: novo znanje.

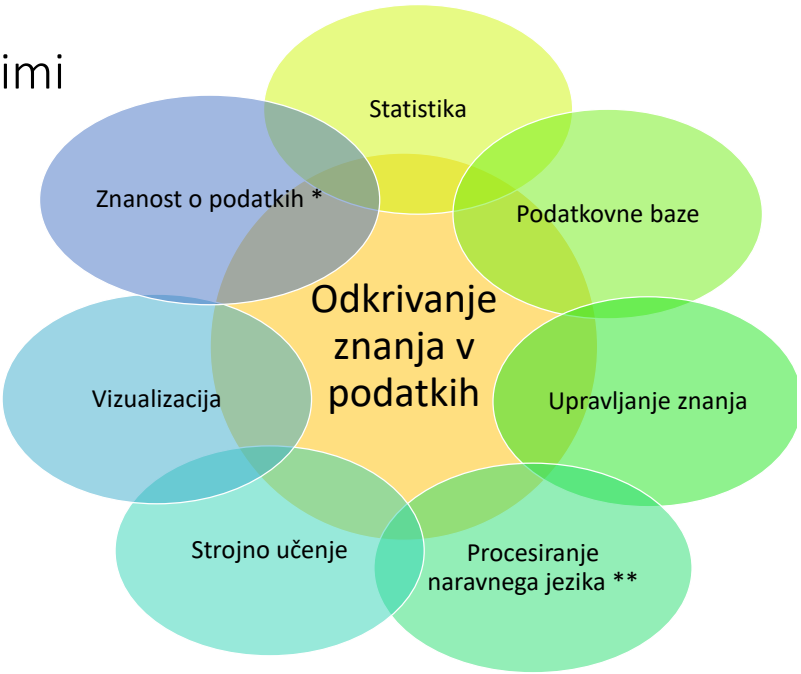
Definicija

Odkrivanje znanja iz podatkov (Knowledge Discovery from Data) je proces odkrivanja doslej neznanega in potencialno uporabnega znanja iz podatkov.

Povezanost z drugimi vedami

* Data science

** Natural language processing



Primeri

Poglejmo si par primerov, kje uporabljamo odkrivanje znanja iz podatkov.

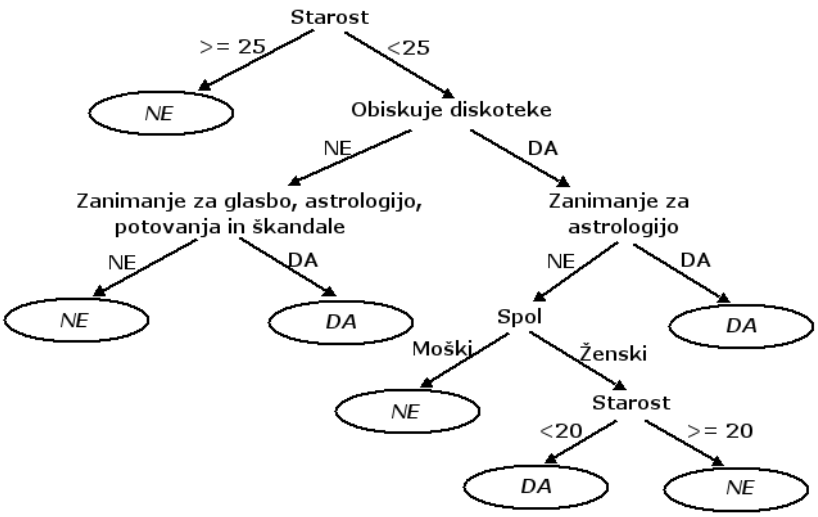
- Primer 1: Bralci revije Antena
- Primer 2: Napoved požarne ogroženosti gozdov
- Primer 3: Stranski učinki (kombinacije) zdravil
- Primer 4: Samovozeča vozila
- Primer 5: Priporočilni sistemi

Primer 1:

Bralci revije Antena

Primer 1:

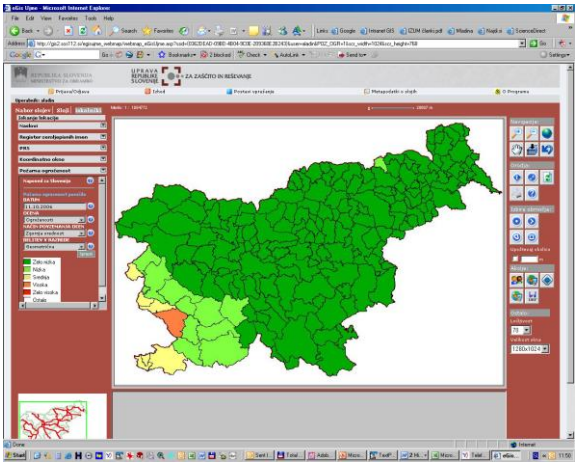
Bralci revije Antena



Odločitveno drevo

Primer 2:

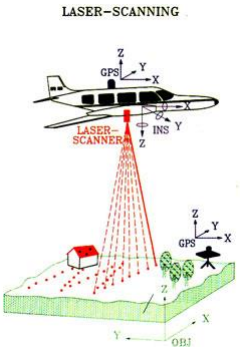
Napoved požarne ogroženosti gozdov

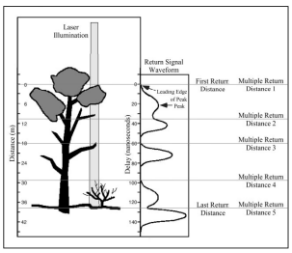


Primer 2:

Merjenje gozdov

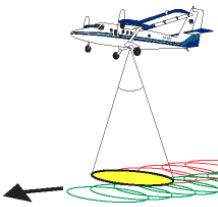
- Merjenje gozda letalom opremljenim z LIDAR laserjem je natančno
- Podatki so pomembni za ocenjevanje požarne ogroženosti
- Merjenje z LIDARjem je drago





Primer 2:

Satelitski posnetki so cenejši



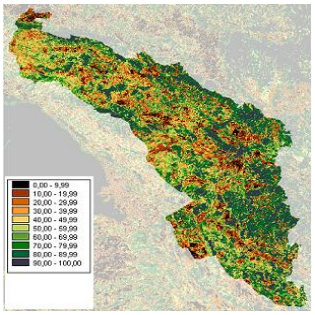
1) Z laserjem izmerimo del gozda.
2) Celo področje posnamemo s satelitom.
3) Naučimo se modela, ki zna iz satelitskih slik oceniti parametre gozda za področja, ki niso bila izmerjena z laserjem.

Primer 2:

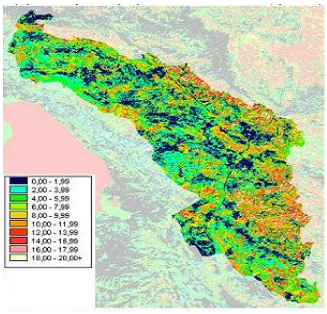
Rezultati modela strojnega učenja

- Minimalna izguba točnosti
- Pocenitev merjenja iz **660** na **0,01** US$/km^2

Višina dreves



Prosojnost gozda



Stojanova, D., Panov, P., Gjorgjioski, V., Kobler, A., & Džeroski, S. (2010). **Estimating vegetation height and canopy cover from remotely sensed data with machine learning**. *Ecological Informatics*, *5*(4), 256-266.

Primer 3:

Stranski učinki (kombinacije) zdravil

Žitnik M, Agrawal M, Leskovec J. Modeling Polypharmacy Side Effects with Graph Convolutional Networks, 2018.

Vir slike: seniorhomes.com

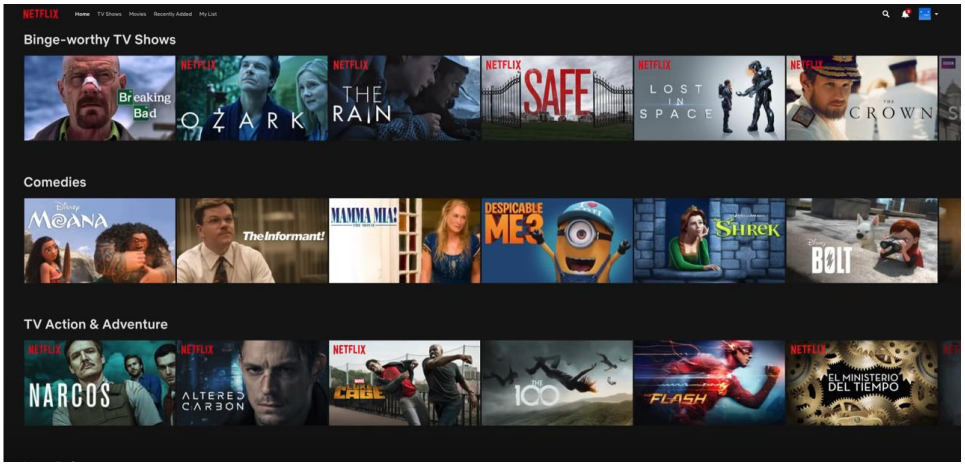
Primer 4:

Samovozeča vozila

Slika: pbs.org

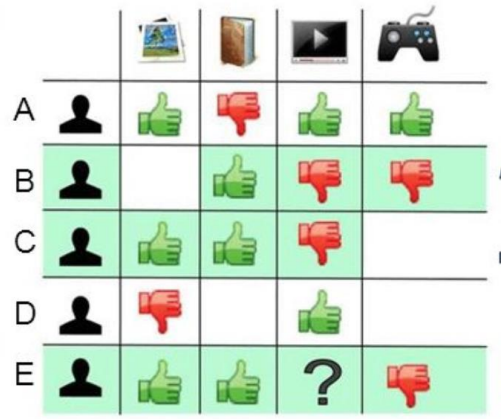
Primer 5:

Priporočilni sistemi (recommender systems)



Primer 5:

Priporočilni sistemi (recommender systems)



- Uporabniki so si podobni, če so jim všeč isti izdelki
- Izdelki so si podobni, če všeč istim uporabnikom

Najvrednejša podjetja 2018

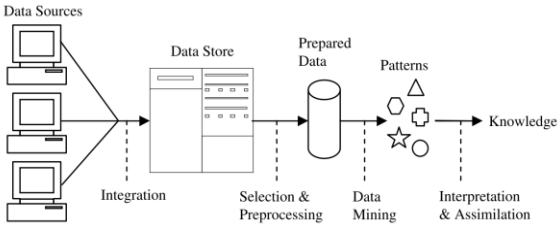
Ranking of the companies rank 1 to 100	Market value in billion U.S. dollars
Apple	926.9
Amazon.com	777.8
Alphabet	766.4
Microsoft	750.6
Facebook	541.5
Alibaba	499.4
Berkshire Hathaway	491.9
Tencent Holdings	491.3
JPMorgan Chase	387.7
ExxonMobil	344.1
Johnson & Johnson	341.3
Samsung Electronics	325.9
Bank of America	313.5

- Informacijske tehnologije
- Finance & Holdingi
- Nafta
- Farmacija

https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-value/

Faze odkrivanja znanja iz podatkov

- integracija podatkov
- priprava podatkov: izbira in predprocesiranje
- **podatkovno rudarjenje**, **vrednotenje**
- interpretacija, predstavitev in uporaba pridobljenega znanja.



The Knowledge Discovery Process

Naloge podatkovnega rudarjenja

- Napovedovanje (označeni podatki)
  - Klasifikacija (classification)
  - Numerično napovedovanje ali tudi regresija (numerical prediction, regression)
- Opisna (neoznačeni podatki)
  - Razvrščanje v skupine (clustering)
  - Odkrivanje asociacijskih pravil (association rule discovery)
- Odkrivanje ubežnikov (outlier detection)
- Priporočitveni sistemi (recommender systems)
- Iskanje in izbiranje informacij (information retrieval)
- Strojno prevajanje
- …

Rudarjenje kompleksnih podatkovnih tipov

- Analiza časovnih vrst (time series analysis)
  - Finančni podatki, monitoriranje srčnega utripa, …
- Analiza besedil (text mining)
  - Novice, komentarji, Wikipedia, leposlovje, …
- Analiza grafov (graph mining)
  - Zemljevidi, molekule
- Analiza socialnih medijev (grafi + besedila)
  - Facebook, Twitter
- Računalniški vid

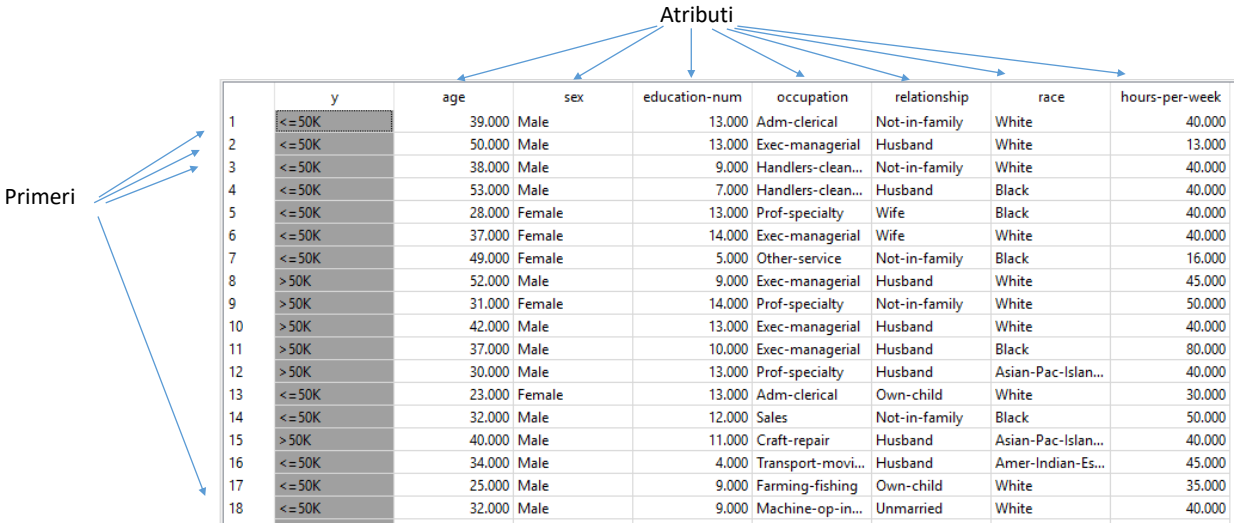
Predprocesiranje

- Podatkovni tipi
- Čiščenje podatkov
- Manjkajoče vrednosti

Podatki za podatkovno rudarjenje

- Imamo množico objektov (npr. pacienti, stranke, nakupi, molekule, dokumenti, recepti,…)
- Obravnavana množica ne obsega vseh objektov, ampak zgolj primere (reprezentativni vzorec)
- Primere opisujemo z njihovimi lastnostmi (atributi, značilke) (npr startst, cena, barva, meritve, …)
- V osnovni obliki imamo primere v tabelarični obliki
  - Vrstice so primeri
  - Stolpci so atributi

Primer: podatkovna zbirka “adult”



	y	age	sex	education-num	occupation	relationship	race	hours-per-week
1	<=50K	39.000	Male	13.000	Adm-clerical	Not-in-family	White	40.000
2	<=50K	50.000	Male	13.000	Exec-managerial	Husband	White	13.000
3	<=50K	38.000	Male	9.000	Handlers-clean...	Not-in-family	White	40.000
4	<=50K	53.000	Male	7.000	Handlers-clean...	Husband	Black	40.000
5	<=50K	28.000	Female	13.000	Prof-specialty	Wife	Black	40.000
6	<=50K	37.000	Female	14.000	Exec-managerial	Wife	White	40.000
7	<=50K	49.000	Female	5.000	Other-service	Not-in-family	Black	16.000
8	>50K	52.000	Male	9.000	Exec-managerial	Husband	White	45.000
9	>50K	31.000	Female	14.000	Prof-specialty	Not-in-family	White	50.000
10	>50K	42.000	Male	13.000	Exec-managerial	Husband	White	40.000
11	>50K	37.000	Male	10.000	Exec-managerial	Husband	Black	80.000
12	>50K	30.000	Male	13.000	Prof-specialty	Husband	Asian-Pac-Islan...	40.000
13	<=50K	23.000	Female	13.000	Adm-clerical	Own-child	White	30.000
14	<=50K	32.000	Male	12.000	Sales	Not-in-family	Black	50.000
15	>50K	40.000	Male	11.000	Craft-repair	Husband	Asian-Pac-Islan...	40.000
16	<=50K	34.000	Male	4.000	Transport-movi...	Husband	Amer-Indian-Es...	45.000
17	<=50K	25.000	Male	9.000	Farming-fishing	Own-child	White	35.000
18	<=50K	32.000	Male	9.000	Machine-op-in...	Unmarried	White	40.000

Vrste atributov

- V podatkovnem rudarjenju v glavnem ločimo dve vrsti atributov:
  - Kategorični (nominalni, binarni, ordinalni)
    - Kategorije: barva, spol, vrsta,…
  - Numerični (celoštevilski, realni, ordinalni, binarni)
    - Odmerjene vrednosti: starost, srčni utrip

- Določite tipe so atributov v zbirki „adult“.

Čiščenje podatkov

- Napake v podatkih se pojavljajo iz različnih razlogov (napake v meritvah, napake pri vnosu, subjektivne ocene – npr bolečina)
  - Primeri očitnih napak:
    - Starost 1000 let
    - Različni zapisi iste vrednosti: NO, N0, No, no,…
    - Tipkarski škrati: bbrown → brown
    - V sicer številskem atributu, vrednost 6M
    - Decimalna pika ali decimalna vejica
    - Vrednosti določenega atributa so pri vseh primerih enake (npr rojstni datum 1.1.1930)
    - Različne enote, byte vs. GB
- Šum (noise) v podatkih so vrednosti, ki so mogoče, a netočne (šum zaradi napak merilnih inštrumentov, ocenjene vrednosti, zaokroževanje,…).

Manjkajoče vrednosti (missing values)

- Nekateri algoritmi strojnega učenja znajo upoštevati manjkajoče vrednosti
- Nekateri algoritmi tega ne znajo
  - Strategije z manjkajočimi vrednostmi:
    - odstrani primer
    - odstrani atribut
    - nadomesti manjkajočo vrednost z najpogostejšo vrednostjo / povprečjem
- V programu Orange, manjkajoče vrednosti označimo z “?”

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Drugi sklop

Klasifikacija

- Odločitvena drevesa
- TDIDT algoritem
- Entropija
- Informacijski pridobitek
- Učna in testna množica
- Klasifikacijska točnost

Klasifikacijski problem

- Cilj: Dodeliti primerom kategorijo

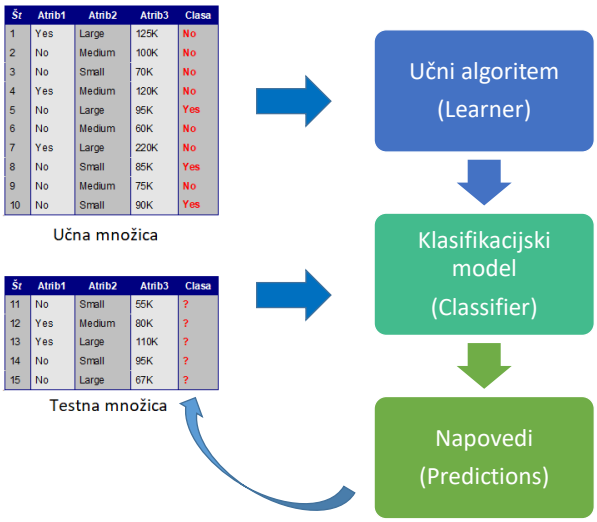
- Dana je množica primerov. Primeri so opisani z atributi.
- Ciljna spremenljivka je atribut, ki nas posebno zanima. Pri klasifikaciji je ciljna spremenljivka kategorična.
- Vrednosti ciljne spremenljivke so razredi (class).
- Na učnih podatkih zgradimo model, ki (čimbolj točno) klasificira nove primere.

Primeri klasifikacijskih problemov

- Cilj: Dodeliti primerom kategorijo

  - Bralec revije ali ne
  - Pacient potrebuje antibiotično zdravljenje ali ne
  - Stranke, ki so verjetni kupci
  - Privrženci določene politične stranke na volitvah
  - Komu odobriti kredit
  - Klasifikacija rastlin / živali / galaksij v razrede
  - Katere naprave se kvarijo
  - ….

Osnovna shema klasifikacije

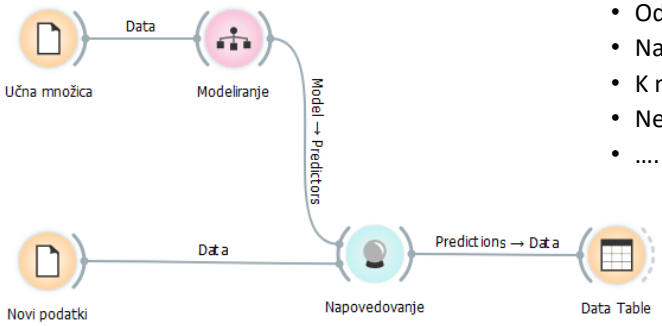


- Klasifikacijski model je preslikava iz atributov v razrede
  - f(X) = Y
- V fazi učenja, sta znana X in Y, učimo se preslikave f
- V fazi napovedovanja sta znana f in X, ki nam določita Y

Osnovna shema klasifikacije v orange

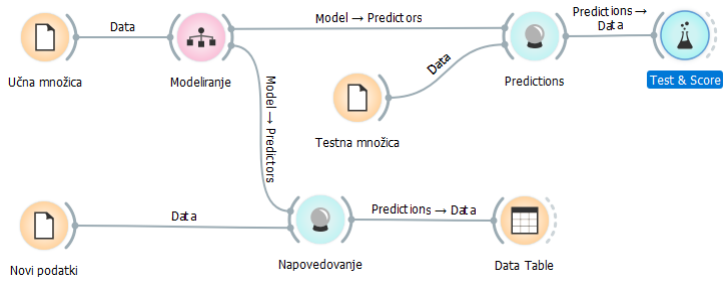
DATA MINING FRUITFUL&FUN

- Na učnih podatkih zgradimo model
- Z modelom klasificiramo nove primere
- Algoritmi za gradnjo modelov:
  - Odločitvena drevesa
  - Naivni Bayesov klasifikator
  - K najbližjih sosedov (KNN)
  - Nevronske mreže
  - ....

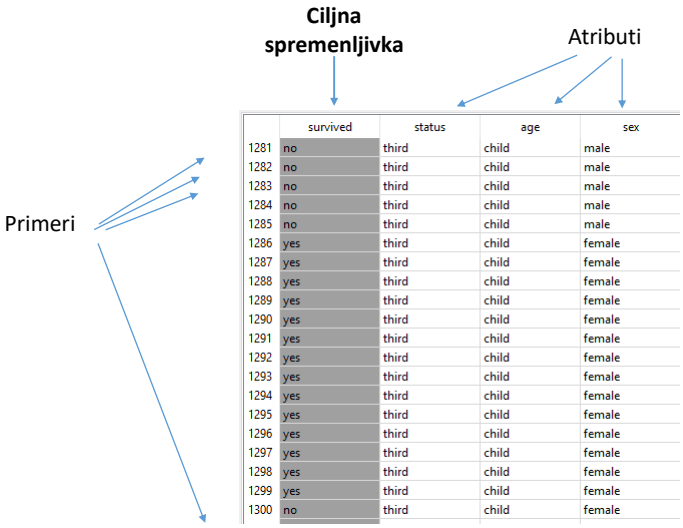


Dopolnjena osnovna shema klasifikacije

- Na učnih podatkih zgradimo model
- Na testnih podatkih ocenimo kvaliteto modela
- Z modelom klasificiramo nove primere

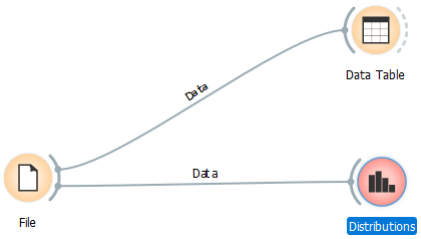


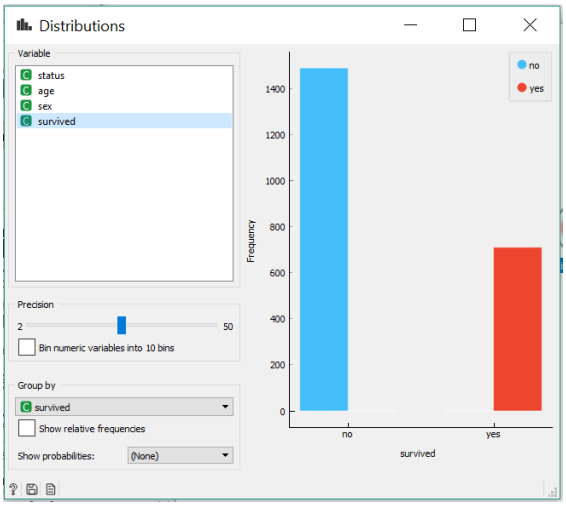
Primer: podatkovna zbirka "titanic"



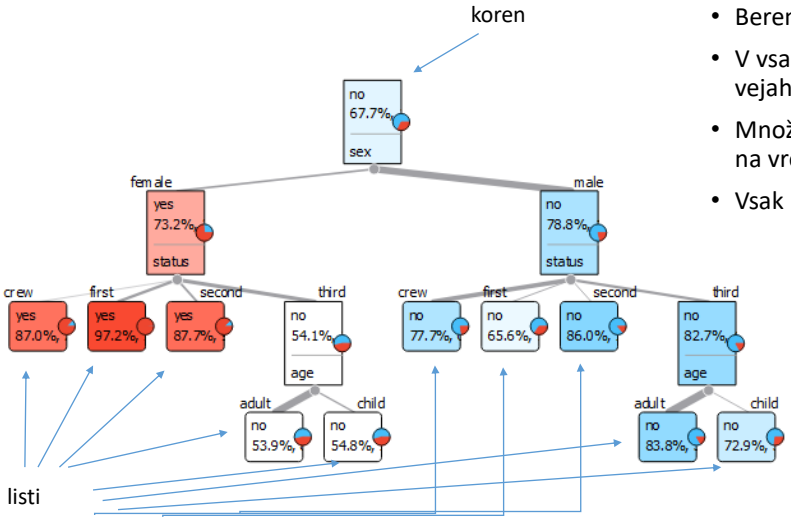
	survived	status	age	sex
1281	no	third	child	male
1282	no	third	child	male
1283	no	third	child	male
1284	no	third	child	male
1285	no	third	child	male
1286	yes	third	child	female
1287	yes	third	child	female
1288	yes	third	child	female
1289	yes	third	child	female
1290	yes	third	child	female
1291	yes	third	child	female
1292	yes	third	child	female
1293	yes	third	child	female
1294	yes	third	child	female
1295	yes	third	child	female
1296	yes	third	child	female
1297	yes	third	child	female
1298	yes	third	child	female
1299	yes	third	child	female
1300	no	third	child	female

Klasifikacija: distribucija ciljne spremenljivka



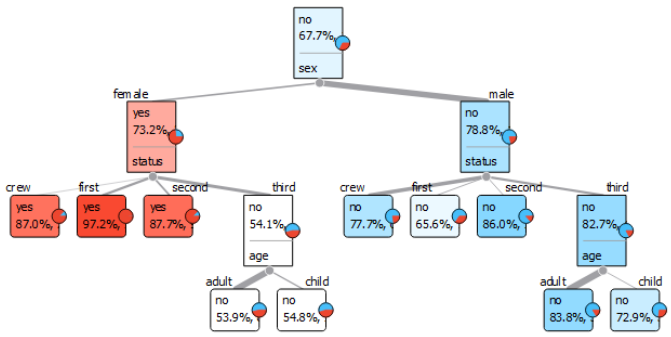


Odločitveno drevo: Preživeli na Titaniku



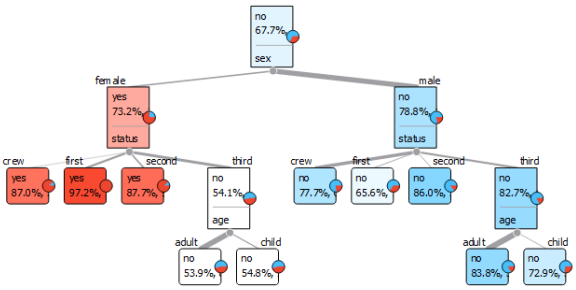
- Beremo on zgoraj navzdol
- V vsakem vozlišču je atribut, na vejah so vrednosti tega atributa
- Množica primerov se razdeli glede na vrednosti atributa
- Vsak primer gre natanko v en list

Vaja: Klasificiraj primere



	status	age	sex	survived?
1	third	child	male	
2	third	child	female	
3	crew	adult	male	
4	first	adult	male	
5	second	adult	male	
6	third	adult	male	
7	first	adult	female	
8	second	adult	female	
9	third	adult	female	
10	third	child	male	

Drevo lahko prepišemo v seznam pravil

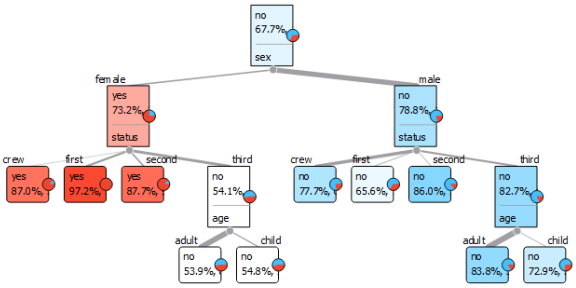


- Vsako pot od korena do lista prepišemo v eno pravilo

→toliko pravil kolikor listov

- Vsak primer spada v natanko en list, torej zanj velja natanko eno pravilo

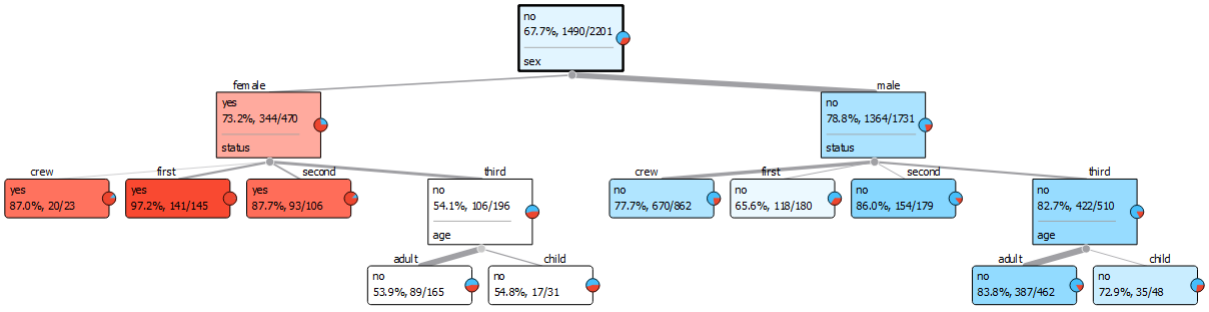
Drevo lahko prepišemo v seznam pravil



- sex = female & status = crew → survived = yes
- sex = female & status = first → survived = yes
- sex = female & status = second → survived = yes
- sex = female & status = third & age = adult → survived = no
- sex = female & status = third & age = child → survived = no
- sex = male & status = crew → survived = no
- sex = male & status = first → survived = no
- sex = male & status = second → survived = no
- sex = male & status = third & age = adult → survived = no
- sex = male & status = third & age = child → survived = no

Odločitveno drevo lahko tudi interpretiramo

- Najpomembnejši atribut?
- Orange vizualizira:
  - Število primerov v vsakem vozlišču
  - Delež primerov večinskega razreda
  - Intenzivnost barve ponazarja gotovost napovedi, debelina povezave …



TDIDT – Top Down Induction of Decision Trees

- Odločitnena drevesa gradimo od zgoraj navzdol
- Vseh možnih odločitvenih dreves za dano množico primerov je zelo veliko
- Pomembno je, kateri atribut izberemo
- Hevristika: izberemo tisti atribut, ki **najbolje loči** razrede

↑

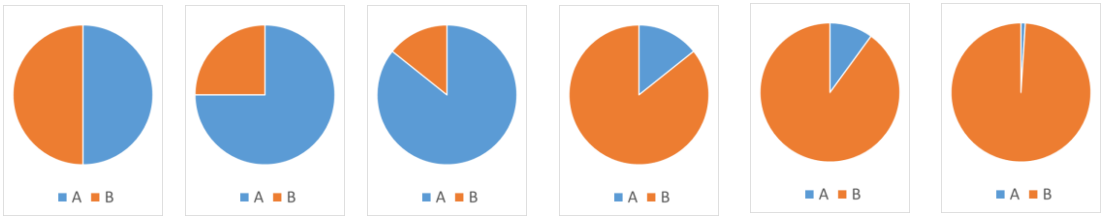
Informacijski pridobitek (information gain)

↑

Entropija

Entropija

- Entropija je količina, ki meri negotovost izzida poskusa.

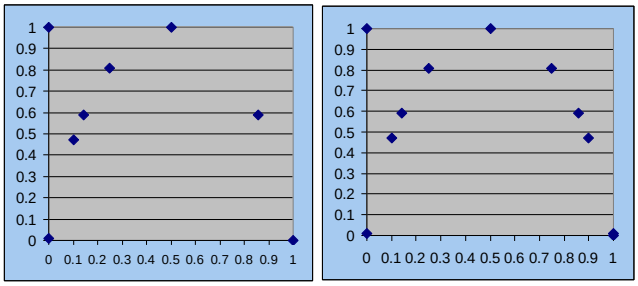


½ ¼ 1/7 6/7 9/10 99/100

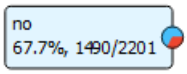
Entropija

$$E(S) = -\sum_{c=1}^{N} p_c \cdot \log_2 p_c$$

- Izračunajmo:
  - E (0,1) = 0
  - E (1/2 , 1/2) = 1
  - E (1/4 , 3/4) = 0.81
  - E (1/7 , 6/7) = 0.59
  - E (6/7 , 1/7) = 0.59
  - E (0.1 , 0.9) = 0.47
  - E (0.001 , 0.999) = 0.01



Entropija množice – primer



Preživeli na Titaniku

- Vseh potnikov je bilo 2201
- Preživelo je 720 potnikov

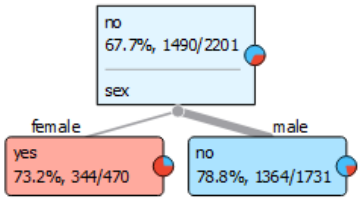
- Celotna množica ima 2201 primerov
- 1490 v razredu NO
- Ostalih 720 v razredu YES

Izračunajmo entropijo

	NO	YES	total
	1490	711	2201
verjetnosti razredov pi	0.677	0.323	
pi * log2 (pi)	-0.38	-0.53	
entropija	0.908		

Informacijski pridobitek

- Je mera, ki nam pove, koliko informacije (o razredih) pridobimo, če množico razdelimo glede na vrednosti nekega atributa.
- Koliko zmanjšamo entropijo



Informacijski pridobitek (Information gain)

$$Gain(S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot E(S_v)$$

množica S

atribut A

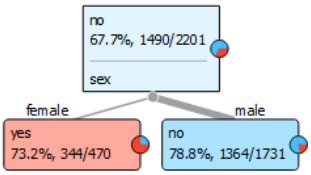
Entropija celotne množice S

Število primerov v podmnožici S_v (verjetnost veje)

Entropija podmnožice S_v

Število primerov v množici S

Informacijski pridobitek - primer

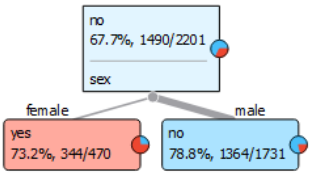


1. Atribut sex razdeli celotno množico na dve podmnožici:
   - **female** ima 470 primerov (344 preživelih)
   - **male** ima 1731 primerov (1364 umrlih)
2. Izračunamo entropijo vsake podmnožice
3. Izračunamo informacijski pridobitek

$$Gain(S,A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot E(S_v)$$

$$Gain\ (S, Sex) = 0{,}911\ - \left(\frac{470}{2201} * 0{,}868 + \frac{1731}{2201} * 0{,}745\right) = 0{,}142$$

Informacijski pridobitek - primer



1. Atribut sex razdeli celotno množico na dve podmnožici:
   - **female** ima 470 primerov (344 preživelih)
   - **male** ima 1731 primerov (1364 umrlih)
2. Izračunamo entropijo vsake podmnožice
3. Izračunamo informacijski pridobitek

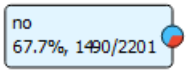
$$Gain(S,A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot E(S_v)$$

$$Gain\ (S, Sex) = 0{,}911\ - \left(\frac{470}{2201} * 0{,}868 + \frac{1731}{2201} * 0{,}745\right) = 0{,}166$$

female	NO	YES	total
	136	334	470
verjetnosti razredov pi	0,289	0,711	
pi * log (pi, 2)	-0,52	-0,35	
entropija	0,868		

male	NO	YES	total
	1364	367	1731
verjetnosti razredov pi	0,788	0,212	
pi * log (pi, 2)	-0,27	-0,47	
entropija	0,745		

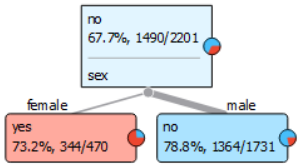
Informacijski pridobitek - primer



- Celotna množica ima 2201 primerov
- 1490 v razredu NO
- Ostali v razredu YES

Entropija

	NO	YES	total
	1490	711	2201
verjetnosti razredov pi	0.677	0.323	
pi * log2 (pi)	-0.38	-0.53	
entropija	0.908		



- Atribut sex razdeli celotno množico na dve podmnožici:
  - Female ima 470 primerov (344 preživelih)
  - Male ima 1731 primerov (1364 umrlih)
- Izračunamo entropijo vsake podmnožice

TDIDT – Top Down Induction of Decision Trees

- Odločitnena drevesa gradimo od zgoraj navzdol
- Vseh možnih odločitvenih dreves za dano množico je zelo veliko
- Pomembno je, kateri atribut izberemo
- Hevristika: izberemo tisti atribut, ki **najbolje loči** razrede

↑

Informacijski pridobitek (information gain)

↑

Entropija

Indukcija odločitvenega drevesa Algoritem ID3

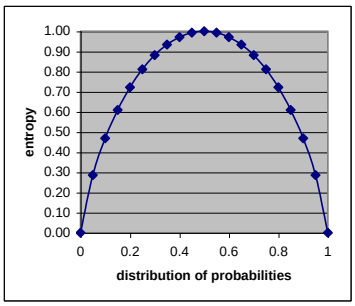
Induce a decision tree on set S:

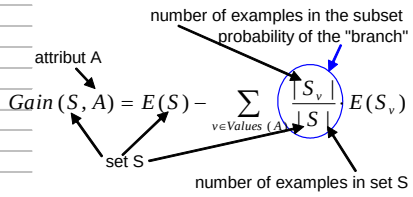
1. Compute the **entropy** E(S) of the set S
2. **IF** E(S) = 0
3. The current set is “clean” and therefore a leaf in our tree
4. **IF** E(S) > 0
5. Compute the **information gain** of each attribute Gain(S, A)
6. The attribute A with the highest information gain becomes the root
7. Divide the set S into subsets S_i according to the values of A
8. Repeat steps 1-7 on each S_i

Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106

Entropija in informacijski pridobitek

probability of class 1	probability of class 2	**entropy $E(p_1, p_2)$ =**
p_1	**$p_2 = 1-p_1$**	$-p_1*\log_2(p_1) - p_2*\log_2(p_2)$
0	1	0.00
0.05	0.95	0.29
0.10	0.90	0.47
0.15	0.85	0.61
0.20	0.80	0.72
0.25	0.75	0.81
0.30	0.70	0.88
0.35	0.65	0.93
0.40	0.60	0.97
0.45	0.55	0.99
0.50	0.50	1.00
0.55	0.45	0.99
0.60	0.40	0.97
0.65	0.35	0.93
0.70	0.30	0.88
0.75	0.25	0.81
0.80	0.20	0.72
0.85	0.15	0.61
0.90	0.10	0.47
0.95	0.05	0.29
1	0	0.00



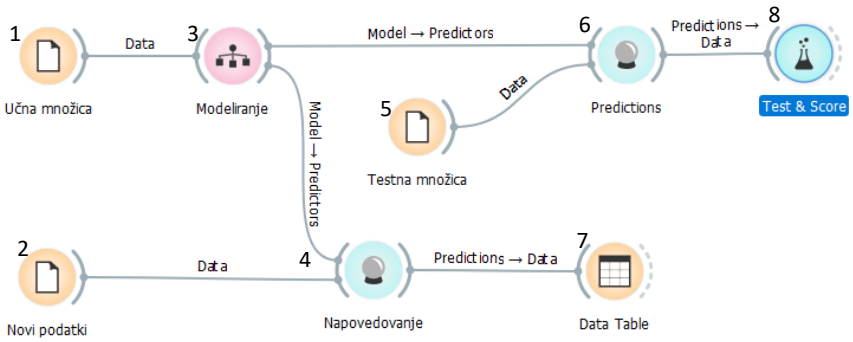


Naloge

1. Katere nove besede in besedne zveze smo danes spoznali?
   1. Odkrivanje zakonitosti v podatkih, klasifikacija, entropija…..
2. Kako bi računal entropijo za trorazredno ciljno spremenljivko Leče = {trde=4, mehke=5, ne=13}
3. V opisu algoritma ID3 smo kot ustavitveni kriterij uporabili entropija E(S)=0. Kateri kriteriji bi bili še smiselni?
4. Je informacijski pridobitek lahko negativen?
5. *Kako bi izračunal informacijski pridobitek zveznega atributa?

Shema klasifikacije

1. Na učnih podatkih zgradimo model: 1,3
2. Na testnih podatkih ocenimo kvaliteto modela: 5,6,8
3. Z modelom klasificiramo nove primere: 2,4,7



Vaja
Gradnja in evalvacija odločitvenega drevesa

Person	Age	Prescription	Astigmatic	Tear_Rate	**Lenses**
P1	young	myope	no	normal	**YES**
P2	young	myope	no	reduced	**NO**
P3	young	hypermetrope	no	normal	**YES**
P4	young	hypermetrope	no	reduced	**NO**
P5	young	myope	yes	normal	**YES**
P6	young	myope	yes	reduced	**NO**
P7	young	hypermetrope	yes	normal	**YES**
P8	young	hypermetrope	yes	reduced	**NO**
P9	pre-presbyopic	myope	no	normal	**YES**
P10	pre-presbyopic	myope	no	reduced	**NO**
P11	pre-presbyopic	hypermetrope	no	normal	**YES**
P12	pre-presbyopic	hypermetrope	no	reduced	**NO**
P13	pre-presbyopic	myope	yes	normal	**YES**
P14	pre-presbyopic	myope	yes	reduced	**NO**
P15	pre-presbyopic	hypermetrope	yes	normal	**NO**
P16	pre-presbyopic	hypermetrope	yes	reduced	**NO**
P17	presbyopic	myope	no	normal	**NO**
P18	presbyopic	myope	no	reduced	**NO**
P19	presbyopic	hypermetrope	no	normal	**YES**
P20	presbyopic	hypermetrope	no	reduced	**NO**
P21	presbyopic	myope	yes	normal	**YES**
P22	presbyopic	myope	yes	reduced	**NO**
P23	presbyopic	hypermetrope	yes	normal	**NO**
P24	presbyopic	hypermetrope	yes	reduced	**NO**

Na podani množici zgradi in evalviraj odločitveno drevo, ki bo klasificiralo, če naj pacientom zdravnik predpiše leče ali ne.

Vaja: Gradnja in evalvacija odločitvenega drevesa

Razdelimo primere na učno in testno množico

Person	Age	Prescription	Astigmatic	Tear_Rate	**Lenses**
P1	young	myope	no	normal	**YES**
P2	young	myope	no	reduced	**NO**
P3	young	hypermetrope	no	normal	**YES**
P4	young	hypermetrope	no	reduced	**NO**
P5	young	myope	yes	normal	**YES**
P6	young	myope	yes	reduced	**NO**
P7	young	hypermetrope	yes	normal	**YES**
P8	young	hypermetrope	yes	reduced	**NO**
P9	pre-presbyopic	myope	no	normal	**YES**
P10	pre-presbyopic	myope	no	reduced	**NO**
P11	pre-presbyopic	hypermetrope	no	normal	**YES**
P12	pre-presbyopic	hypermetrope	no	reduced	**NO**
P13	pre-presbyopic	myope	yes	normal	**YES**
P14	pre-presbyopic	myope	yes	reduced	**NO**
P15	pre-presbyopic	hypermetrope	yes	normal	**NO**
P16	pre-presbyopic	hypermetrope	yes	reduced	**NO**
P17	presbyopic	myope	no	normal	**NO**
P18	presbyopic	myope	no	reduced	**NO**
P19	presbyopic	hypermetrope	no	normal	**YES**
P20	presbyopic	hypermetrope	no	reduced	**NO**
P21	presbyopic	myope	yes	normal	**YES**
P22	presbyopic	myope	yes	reduced	**NO**
P23	presbyopic	hypermetrope	yes	normal	**NO**
P24	presbyopic	hypermetrope	yes	reduced	**NO**

Vaja: Gradnja in evalvacija odločitvenega drevesa

Učna množica

Person	Age	Prescription	Astigmatic	Tear_Rate	**Lenses**
P1	young	myope	no	normal	**YES**
P2	young	myope	no	reduced	**NO**
P4	young	hypermetrope	no	reduced	**NO**
P5	young	myope	yes	normal	**YES**
P6	young	myope	yes	reduced	**NO**
P7	young	hypermetrope	yes	normal	**YES**
P8	young	hypermetrope	yes	reduced	**NO**
P10	pre-presbyopic	myope	no	reduced	**NO**
P11	pre-presbyopic	hypermetrope	no	normal	**YES**
P14	pre-presbyopic	myope	yes	reduced	**NO**
P17	presbyopic	myope	no	normal	**NO**
P18	presbyopic	myope	no	reduced	**NO**
P19	presbyopic	hypermetrope	no	normal	**YES**
P20	presbyopic	hypermetrope	no	reduced	**NO**
P21	presbyopic	myope	yes	normal	**YES**
P22	presbyopic	myope	yes	reduced	**NO**
P24	presbyopic	hypermetrope	yes	reduced	**NO**

Vaja: Gradnja in evalvacija odločitvenega drevesa

Indukcija odločitvenega drevesa Algoritem ID3

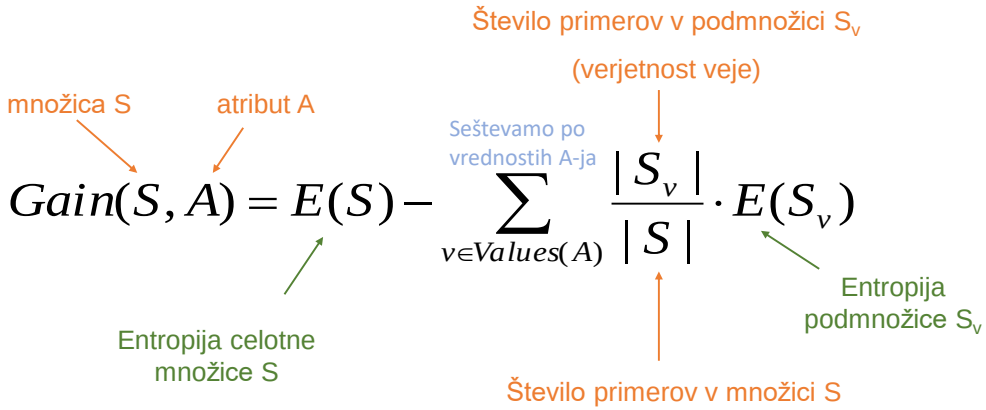
Induce a decision tree on set S:

1. Compute the **entropy** E(S) of the set S
2. **IF** E(S) = 0
3. The current set is “clean” and therefore a leaf in our tree
4. **IF** E(S) > 0
5. Compute the **information gain** of each attribute Gain(S, A)
6. The attribute A with the highest information gain becomes the root
7. Divide the set S into subsets S_i according to the values of A
8. Repeat steps 1-7 on each S_i

Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106

Vaja: Gradnja in evalvacija odločitvenega drevesa

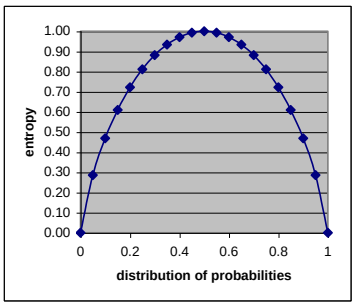
Informacijski pridobitek (Information gain)

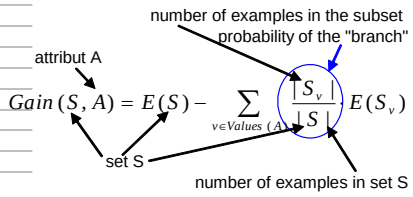


Vaja: Gradnja in evalvacija odločitvenega drevesa

Entropija in informacijski pridobitek

probability of class 1	probability of class 2	**entropy $E(p_1, p_2)$ =**
p_1	**$p_2 = 1-p_1$**	$-p_1*\log_2(p_1) - p_2*\log_2(p_2)$
0	1	0.00
0.05	0.95	0.29
0.10	0.90	0.47
0.15	0.85	0.61
0.20	0.80	0.72
0.25	0.75	0.81
0.30	0.70	0.88
0.35	0.65	0.93
0.40	0.60	0.97
0.45	0.55	0.99
0.50	0.50	1.00
0.55	0.45	0.99
0.60	0.40	0.97
0.65	0.35	0.93
0.70	0.30	0.88
0.75	0.25	0.81
0.80	0.20	0.72
0.85	0.15	0.61
0.90	0.10	0.47
0.95	0.05	0.29
1	0	0.00



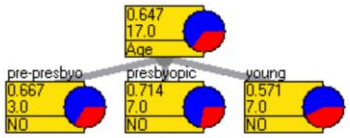


Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Age on set S:

Training set

Person	Age	Prescription	Astigmatic	Tear_Rate	Lenses
P1	young	myope	no	normal	YES
P2	young	myope	no	reduced	NO
P4	young	hypermetrope	no	reduced	NO
P5	young	myope	yes	normal	YES
P6	young	myope	yes	reduced	NO
P7	young	hypermetrope	yes	normal	YES
P8	young	hypermetrope	yes	reduced	NO
P10	pre-presbyopic	myope	no	reduced	NO
P11	pre-presbyopic	hypermetrope	no	normal	YES
P14	pre-presbyopic	myope	yes	reduced	NO
P17	presbyopic	myope	no	normal	NO
P18	presbyopic	myope	no	reduced	NO
P19	presbyopic	hypermetrope	no	normal	YES
P20	presbyopic	hypermetrope	no	reduced	NO
P21	presbyopic	myope	yes	normal	YES
P22	presbyopic	myope	yes	reduced	NO
P24	presbyopic	hypermetrope	yes	reduced	NO



The attribute Age splits the set S into three subsets: Age=young, Age=pre-presbyopic and Age=presbyopic with 7, 3 and 7 instances respectively.
In the subset Age = young, there are 3 items with Lenses=YES and 4 with Lenses=NO.
E(Age=young) = E(3/7, 4/7) = 0.99.
Similar for the other two sets:
E(Age=pre-presbyopic) = E(1/3, 2/3) = 0.92
E(Age=presbyopic) = E(2/7, 5/7) = 0.86

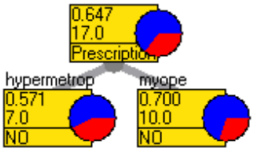
Gain (S,Age) =
E(S) – 7/17 E(Age= young) – 3/17 E(Age=pre-presbyopic) – 7/17 E(Age=presbyopic) =
= 0.94 – 7/17 * 0.99 – 3/17 *0.92 – 7/17 *0.86 = 0.02

Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Prescription on set S:

Training set

Person	Age	Prescription	Astigmatic	Tear_Rate	Lenses
P1	young	myope	no	normal	YES
P2	young	myope	no	reduced	NO
P4	young	hypermetrope	no	reduced	NO
P5	young	myope	yes	normal	YES
P6	young	myope	yes	reduced	NO
P7	young	hypermetrope	yes	normal	YES
P8	young	hypermetrope	yes	reduced	NO
P10	pre-presbyopic	myope	no	reduced	NO
P11	pre-presbyopic	hypermetrope	no	normal	YES
P14	pre-presbyopic	myope	yes	reduced	NO
P17	presbyopic	myope	no	normal	NO
P18	presbyopic	myope	no	reduced	NO
P19	presbyopic	hypermetrope	no	normal	YES
P20	presbyopic	hypermetrope	no	reduced	NO
P21	presbyopic	myope	yes	normal	YES
P22	presbyopic	myope	yes	reduced	NO
P24	presbyopic	hypermetrope	yes	reduced	NO



E(Prescription=hypermetrope) =
= E(3/7, 4/7) = 0.99

E(Prescription=myope) =
= E(3/10, 7/10) = 0.88

Gain (S, Prescription) =
= E(S) –
– 7/17 E(Prescription=hypermetrope)
– 10/17 E(Prescription=myope) =
= 0.94 – 7/17* 0.99 – 10/17 * 0.88 = 0.02

Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Astigmatic on set S:

Training set

Person	Age	Prescription	Astigmatic	Tear_Rate	Lenses
P1	young	myope	no	normal	YES
P2	young	myope	no	reduced	NO
P4	young	hypermetrope	no	reduced	NO
P5	young	myope	yes	normal	YES
P6	young	myope	yes	reduced	NO
P7	young	hypermetrope	yes	normal	YES
P8	young	hypermetrope	yes	reduced	NO
P10	pre-presbyopic	myope	no	reduced	NO
P11	pre-presbyopic	hypermetrope	no	normal	YES
P14	pre-presbyopic	myope	yes	reduced	NO
P17	presbyopic	myope	no	normal	NO
P18	presbyopic	myope	no	reduced	NO
P19	presbyopic	hypermetrope	no	normal	YES
P20	presbyopic	hypermetrope	no	reduced	NO
P21	presbyopic	myope	yes	normal	YES
P22	presbyopic	myope	yes	reduced	NO
P24	presbyopic	hypermetrope	yes	reduced	NO

E(Astigmatic=no) = E(3/9, 6/9) = 0.92

E(Astigmatic =yes) = E(3/8, 5/8) = 0.95

Gain (S, Astigmatic) =
= E(S) –
- 9/17 E(Astigmatic=no)
- 8/17 E(Astigmatic=yes) =

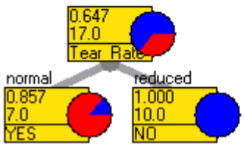
= 0.94 – 9/17* 0.92 – 8/17 * 0.95 = 0.006

Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Tear_Rate on set S:

Training set

Person	Age	Prescription	Astigmatic	Tear_Rate	Lenses
P1	young	myope	no	normal	YES
P2	young	myope	no	reduced	NO
P4	young	hypermetrope	no	reduced	NO
P5	young	myope	yes	normal	YES
P6	young	myope	yes	reduced	NO
P7	young	hypermetrope	yes	normal	YES
P8	young	hypermetrope	yes	reduced	NO
P10	pre-presbyopic	myope	no	reduced	NO
P11	pre-presbyopic	hypermetrope	no	normal	YES
P14	pre-presbyopic	myope	yes	reduced	NO
P17	presbyopic	myope	no	normal	NO
P18	presbyopic	myope	no	reduced	NO
P19	presbyopic	hypermetrope	no	normal	YES
P20	presbyopic	hypermetrope	no	reduced	NO
P21	presbyopic	myope	yes	normal	YES
P22	presbyopic	myope	yes	reduced	NO
P24	presbyopic	hypermetrope	yes	reduced	NO



E(Tear_Rate=normal) = E(6/7, 1/7) = 0.59

E(Tear_Rate=reduced) = E(0/10, 10/10) = 0

Gain (S, Tear_Rate) =
= E(S) –
- 7/17 E(Tear_Rate=normal)
- 10/17 E(Tear_Rate=reduced) =

= 0.94 – 7/17* 0.59 – 10/17 * 0 = 0.70

Vaja: Gradnja in evalvacija odločitvenega drevesa

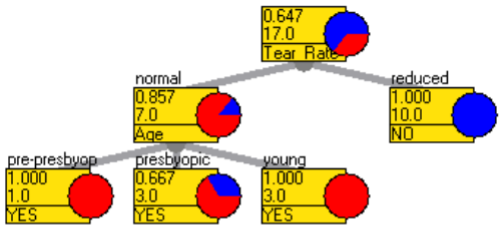
- Atribut z največjim informacijskim pridobitkom (0,7) je tear_rate, zato ga izberemo kot koren drevesa.
- Ta atribut razdeli učno množico na dve podmnožici:
  - *tear_rate = normal*,
  - *tear_rate = reduced*.
- Na vsaki od teh podmnožic rekurzivno gradimo drevo.
  - Podmnožica *tear_rate = reduced je "čista" (vsi primeri pripadajo istemu razredu, zato se algoritem ustavi.*
  - *Nadaljujemo z množico tear_rate = normal.*

Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Age on set Tear_Rate=normal:

Training set

Person	Age	Prescription	Astigmatic	Lenses
P1	young	myope	no	YES
P5	young	myope	yes	YES
P7	young	hypermetrope	yes	YES
P11	pre-presbyopic	hypermetrope	no	YES
P17	presbyopic	myope	no	NO
P19	presbyopic	hypermetrope	no	YES
P21	presbyopic	myope	yes	YES



E(Age=young | Tear_Rate=normal) = E(3/3, 0/3) = 0
E(Age=pre-presbyopic | Tear_Rate=normal) = E(1/1, 0/1) = 0
E(Age=presbyopic | Tear_Rate=normal) = E(2/3, 1/3) = 0.92

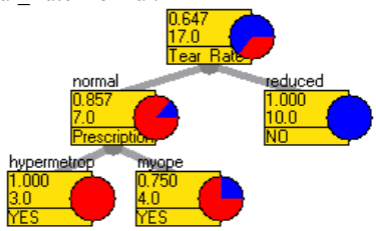
Gain (S Tear_Rate=normal, Age) =
E(S Tear_Rate=normal) – 3/7 E(Age=young | Tear_Rate=normal)
– 1/7 E(Age=pre-presbyopic | Tear_Rate=normal)
– 3/7 E(Age=presbyopic | Tear_Rate=normal) =
= 0.59 – 3/7 * 0 - 1/7 * 0 - 3/7 *0.92 = 0.20

Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Prescription on set Tear_Rate=normal:

Training set

Person	Age	Prescription	Astigmatic	**Lenses**
P1	young	myope	no	**YES**
P5	young	myope	yes	**YES**
P7	young	hypermetrope	yes	**YES**
P11	pre-presbyopic	hypermetrope	no	**YES**
P17	presbyopic	myope	no	**NO**
P19	presbyopic	hypermetrope	no	**YES**
P21	presbyopic	myope	yes	**YES**



E(Prescription=myope | Tear_Rate=normal) = E(3/4, 1/4) = 0.81
E(Prescription=hypermetropy | Tear_Rate=normal) = E(3/3, 0/3) = 0

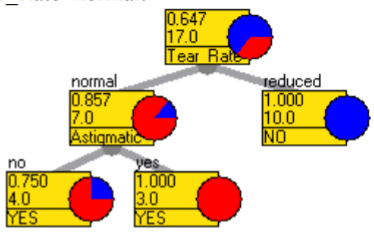
Gain (S Tear_Rate=normal, Prescription) =
E(S Tear_Rate=normal) – 4/7 E(Prescription=myope | Tear_Rate=normal)
– 3/7 E(Prescription=hypermetropy | Tear_Rate=normal) =
= 0.59 – 4/7 * 0.81 - 3/7 * 0 = 0.13

Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Astigmatic on set Tear_Rate=normal:

Training set

Person	Age	Prescription	Astigmatic	**Lenses**
P1	young	myope	no	**YES**
P5	young	myope	yes	**YES**
P7	young	hypermetrope	yes	**YES**
P11	pre-presbyopic	hypermetrope	no	**YES**
P17	presbyopic	myope	no	**NO**
P19	presbyopic	hypermetrope	no	**YES**
P21	presbyopic	myope	yes	**YES**

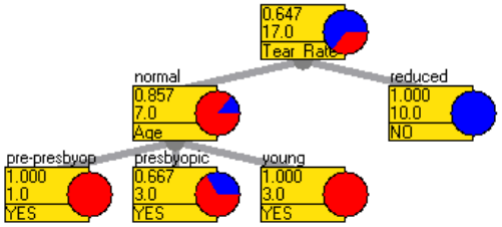


E(Astigmatic=no | Tear_Rate=normal) = E(3/4, 1/4) = 0.81
E(Astigmatic=yes | Tear_Rate=normal) = E(3/3, 0/3) = 0

Gain (S Tear_Rate=normal, Astigmatic) =
E(S Tear_Rate=normal) – 4/7 E(Astigmatic=no | Tear_Rate=normal)
– 3/7 E(Astigmatic=yes | Tear_Rate=normal) =
= 0.59 – 4/7 * 0.81 - 3/7 * 0 = 0.13

Vaja: Gradnja in evalvacija odločitvenega drevesa

- Atribut z največjim informacijskim pridobitkom pri pogoju *tear_rate=normal* nivoju (0,2) je *Age*, zato ga uporabimo za gradnjo drevesa.

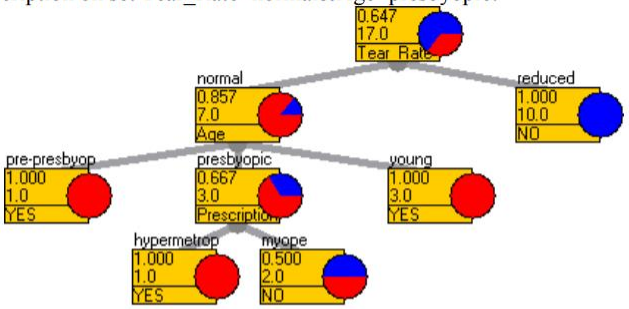


Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Prescription on set Tear_Rate=normal&Age=presbyopic:

Training set

Person	Prescription	Astigmatic	**Lenses**
P17	myope	no	**NO**
P19	hypermetrope	no	**YES**
P21	myope	yes	**YES**



E(Prescription=myope | Tear_Rate=normal&Age=presbyopic) = E(1/2, 1/2) = 1
E(Prescription=hypermetropy | Tear_Rate=normal&Age=presbyopic) = E(1/1, 0/1) = 0

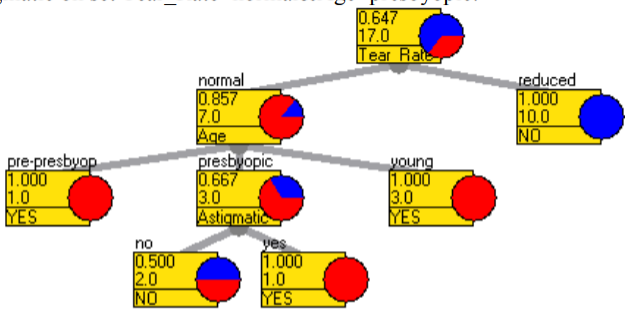
Gain (S Tear_Rate=normal&Age=presbyopic, Prescription) =
E(S Tear_Rate=normal&Age=presbyopic)
– 2/3 E(Prescription=myope | Tear_Rate=normal&Age=presbyopic)
– 1/3 E(Prescription=hypermetropy | Tear_Rate=normal&Age=presbyopic) =
= 0.92 – 2/3 *1 – 1/3 * 0 = 0.25

Vaja: Gradnja in evalvacija odločitvenega drevesa

Information gain of the attribute Astigmatic on set Tear_Rate=normal&Age=presbyopic:

Training set

Person	Prescription	Astigmatic	Lenses
P17	myope	no	NO
P19	hypermetrope	no	YES
P21	myope	yes	YES

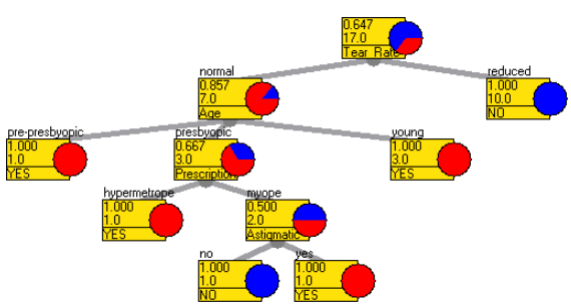


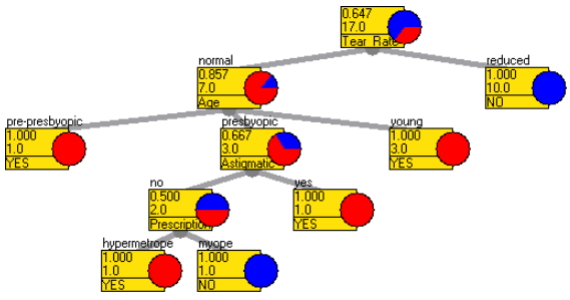
E(Astigmatic=no | Tear_Rate=normal&Age=presbyopic) = E(1/2, 1/2) = 1
E(Astigmatic=yes | Tear_Rate=normal&Age=presbyopic) = E(1/1, 0/1) = 0

Gain (S Tear_Rate=normal&Age=presbyopic, Prescription) =
E(S Tear_Rate=normal&Age=presbyopic)
– 2/3 E(Astigmatic=no | Tear_Rate=normal&Age=presbyopic)
– 1/3 E(Astigmatic=yes | Tear_Rate=normal&Age=presbyopic)=
= 0.92 – 2/3 *1 – 1/3 * 0 = 0.25

Vaja: Gradnja in evalvacija odločitvenega drevesa

- Oba atributa Prescription in Astigmatic imata enak informacijski pridobitek 0.25. Algoritem ID3 bi izbral prvega.

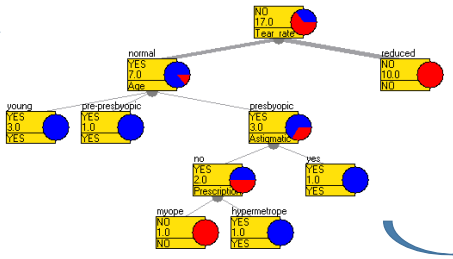




Vaja: Gradnja in evalvacija odločitvenega drevesa

Vaja: Klasifikacija z drevesom

Person	Age	Prescription	Astigmatic	Tear_Rate	**Lenses**
P3	young	hypermetrope	no	normal	**YES**
P9	pre-presbyopic	myope	no	normal	**YES**
P12	pre-presbyopic	hypermetrope	no	reduced	**NO**
P13	pre-presbyopic	myope	yes	normal	**YES**
P15	pre-presbyopic	hypermetrope	yes	normal	**NO**
P16	pre-presbyopic	hypermetrope	yes	reduced	**NO**
P23	presbyopic	hypermetrope	yes	normal	**NO**



Klasifikacijska točnost = (3+2)/ (3+2+2+0) = 71%

	Napoved „DA“	Napoved „NE“
Dejanski „DA“	TP=3	FN=0
Dejanski „NE“	FP=2	TN=2

Naloga

- Novi izrazi…
- Kolikšen je informacijski pridobitek *atributa* “PersonId”?
- Koliko bi bila klasifikacijska točnost drevesa “lenses”, če bi ga porezali pri atributu “Astigmatic”?

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Tretji sklop

Evalvacija

- Cilji
- Metode
- Metrike

Cilj evalvacije

- Kako dober je model?
- Metoda
  - Kako merimo
- Metrika
  - Kaj merimo

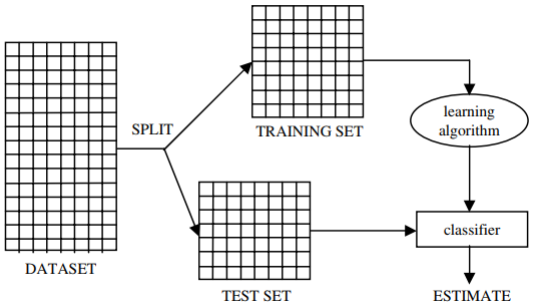
Metoda: Testiranje na testni množici

Person	Age	Prescription	Astigmatic	Tear_Rate	**Lenses**
P1	young	myope	no	normal	**YES**
P2	young	myope	no	reduced	**NO**
P3	young	hypermetrope	no	normal	**YES**
P4	young	hypermetrope	no	reduced	**NO**
P5	young	myope	yes	normal	**YES**
P6	young	myope	yes	reduced	**NO**
P7	young	hypermetrope	yes	normal	**YES**
P8	young	hypermetrope	yes	reduced	**NO**
P9	pre-presbyopic	myope	no	normal	**YES**
P10	pre-presbyopic	myope	no	reduced	**NO**
P11	pre-presbyopic	hypermetrope	no	normal	**YES**
P12	pre-presbyopic	hypermetrope	no	reduced	**NO**
P13	pre-presbyopic	myope	yes	normal	**YES**
P14	pre-presbyopic	myope	yes	reduced	**NO**
P15	pre-presbyopic	hypermetrope	yes	normal	**NO**
P16	pre-presbyopic	hypermetrope	yes	reduced	**NO**
P17	presbyopic	myope	no	normal	**NO**
P18	presbyopic	myope	no	reduced	**NO**
P19	presbyopic	hypermetrope	no	normal	**YES**
P20	presbyopic	hypermetrope	no	reduced	**NO**
P21	presbyopic	myope	yes	normal	**YES**
P22	presbyopic	myope	yes	reduced	**NO**
P23	presbyopic	hypermetrope	yes	normal	**NO**
P24	presbyopic	hypermetrope	yes	reduced	**NO**

30% primerov damo v testno množico

- Na testni množici ocenimo delovanje modela (npr. klasifikacijsko točnost).

Metoda: Testiranje na testni množici

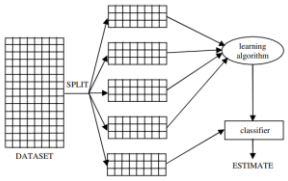


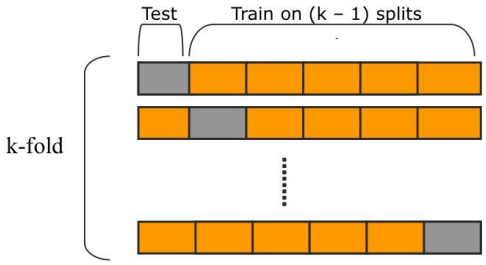
Metoda: Naključno vzorčenje (Random sampling)

- Večkrat ponovimo „Testiranje na testni množici“ z različnim izborom podmnožice za testiranje.
- Rezultate povprečimo, računamo varianco,…
- Bolj zanesljivi rezultati, ki ne zavisijo od naključnega vzorca.

Metoda: K-kratno prečno preverjanje

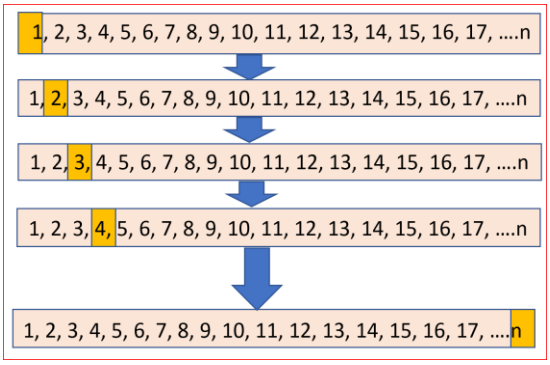
- Standard v strojnem učenju
- Množico razdelimo na *k* podmnožic
- Ponovimo k-krat:
  - Vsakič drugo podmnožico vzamemo za testne podatke
  - Vse ostale podatke uporabimo kot učne podatke
- Vsak primer je natanko enkrat v testni množici





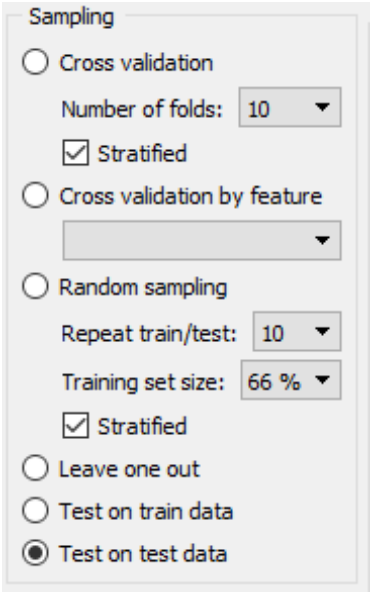
Metoda: Pusti enega zunaj (Leave one out)

- Za majhne množice primerov
- Podobno prečnemu preverjanju, le da je v testni množici le en primer
- Učenje ponovimo n-krat, če je n primerov v množici



Metode evalvacije v Orange

- Prečno preverjanje
- Naključno vzorčenje
- Pusti enega zunaj
- Testiranje na učni množici
- Testiranje na testni množici



Mere za evalvacijo klasifikacije

Kontingenčna tabela (Confusion matrix)

- Matrika (pravilnih in napačnih) razvrstitev

Primer: car

Predicted

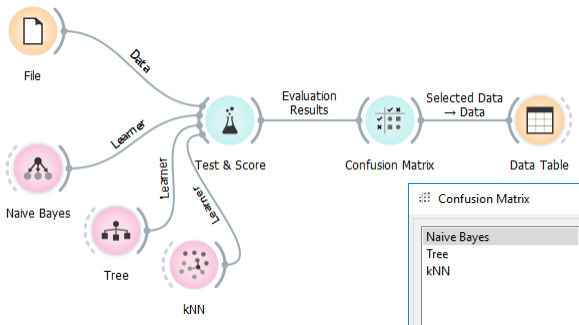
Actual	unacc	acc	good	v-good	Σ
unacc	1154	54	2	0	1210
acc	94	276	14	0	384
good	0	44	22	3	69
v-good	0	25	0	40	65
Σ	1248	399	38	43	1728

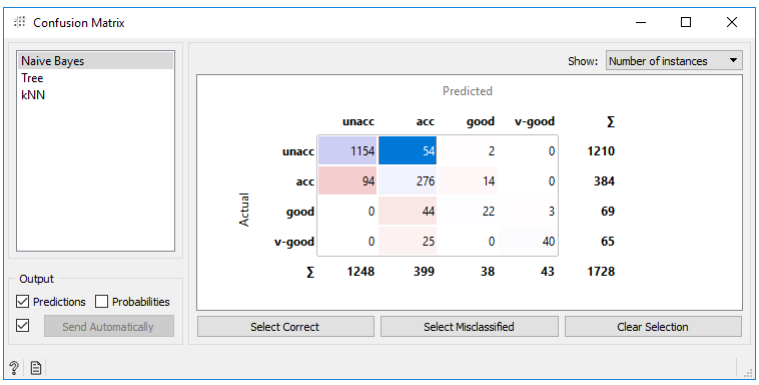
Primer: titanic

Predicted

Actual	no	yes	Σ
no	1364	126	1490
yes	362	349	711
Σ	1726	475	2201

Interaktivna kontingenčna tabela v Orange





Kontingenčna tabela (Confusion matrix)

- Matrika (pravilnih in napačnih) razvrstitev
  - Vrstice so dejanski razredi
  - Stolpci so napovedani razredi
  - Pravilne klasifikacije so na diagonali

Actual \ Predicted	unacc	acc	good	v-good	Σ
unacc	1154	54	2	0	1210
acc	94	276	14	0	384
good	0	44	22	3	69
v-good	0	25	0	40	65
Σ	1248	399	38	43	1728

Iz kontingenčne tabele lahko izračunamo:
- Število pravilno klasificiranih primerov
- Število napačno klasificiranih primerov
- Klasifikacijsko točnost (classification accuracy)
- Za vsak razred
  - Priklic (recall)
  - Natančnost (precision)
  - F1

Kontingenčna tabela za dva razreda

Klasifikacija klasifikatorja

Dejanska klasifikacija	Correct classification	Classified as +	Classified as −
	+	true positives	false negatives
	−	false positives	true negatives

TP: true positives
The number of positive instances that are classified as positive

FP: false positives
The number of negative instances that are classified as positive

FN: false negatives
The number of positive instances that are classified as negative

TN: true negatives
The number of negative instances that are classified as negative

- Diagonala: pravilno klasificirani
- Izven diagonale: napačno klasificirani
- Klasifikacijska točnost =
- = pravilno klasificirani / vsi primeri
- = pravilno klasificirani / (pravilno klasificirani + napačno klasificirani)

Klasifikacijska točnost (classification accuracy)

- Delež pravilno klasificiranih primerov
- V kontingenčni tabeli so pravilno klasificirani primeri na diagonali

Klasifikacijska točnost =
= pravilno klasificirani / vsi primeri
= pravilno klasificirani / (pravilno klasificirani + napačno klasificirani)

Vaja: Kontingenčna tabela

titanic

Actual \ Predicted	no	yes	Σ
no	1364	126	1490
yes	362	349	711
Σ	1726	475	2201

car

Actual \ Predicted	unacc	acc	good	v-good	Σ
unacc	1154	54	2	0	1210
acc	94	276	14	0	384
good	0	44	22	3	69
v-good	0	25	0	40	65
Σ	1248	399	38	43	1728

1. Za dani kontingenčni tabeli izračunaj:

	titanic	car
Število vseh primerov		
Število razredov		
Število dejanskih primerov v posameznem razredu		
Število klasificiranih primerov v posamezen razred		
Število nepravilno klasificiranih primerov		
Klasifikacijska točnost (delež pravilno klasificiranih primerov)		

Klasifikacija v večinski razred

Predicted

Actual	unacc	acc	good	v-good	Σ
unacc	1154	54	2	0	1210
acc	94	276	14	0	384
good	0	44	22	3	69
v-good	0	25	0	40	65
Σ	1248	399	38	43	1728

Predicted

Actual	no	yes	Σ
no	1364	126	1490
yes	362	349	711
Σ	1726	475	2201

- Kolikšna je klasifikacijska točnost klasifikatorja, ki vse primere klasificira v večinski razred?
- Car: 70% Titanic: 68%

Neuravnotežene množice in množice z različnimi cenami napačne klasifikacije

- Primerov enega razreda je bistveno več kot drugega (ostalih)
- Pogosto nas manjšinjski razred bolj zanima
- Primeri:
  - Presejalni testi (npr. nuhalna svetlina v nosečnosti, Zora, Dora, Svit,…)

Zora
Državni program zgodnjega odkrivanja predrakavih sprememb maternicnega vratu

Dora
DRŽAVNI PRESEJALNI PROGRAM ZA RAKA DOJK

Svit
DRŽAVNI PROGRAM PRESEJANJA IN ZGODNJEGA ODKRIVANJA PREDRAKAVIH SPREMEMB IN RAKA NA DEBELEM ČREVESU IN DANKI

  - Intrusion detection
  - Zloraba kreditnih kartic

Primer: Zloraba kreditnih kartic

*„Fed report notes the fraud rate for debit and prepaid signature transactions in 2012 was approximately 4.04 basis points (bps), or about **four per every 10,000 transactions**.“*

- Kolikšna je klasifikacijska točnost klasifikatorja, ki vse klasificira v razred „not fraudulent“?
  - Odgovor: 99.96%
- Je lahko klasifikator s klasifikacijsko točnostjo 98% boljši od onega, ki klasificira z 99.96%?

https://www.pymnts.com/in-depth/2014/a-tale-of-two-fraud-stats/

Primer: Zloraba kreditnih kartic

Dve kontingenčni tabeli za dva klasifikatorja

		Napoved		
		Fraud	**Not Fraud**	
Dejansko	**Fraud**	**0**	**4**	**4**
	Not Fraud	**0**	**10000**	**10000**
		0	**10004**	**10004**

		Napoved		
		Fraud	**Not Fraud**	
Dejansko	**Fraud**	**4**	**0**	**4**
	Not Fraud	**300**	**9700**	**10000**
		304	**9700**	**10004**

Klasifikacijska točnost

- CA = (0 + 10000)/10004
  = 99,96%

- CA = (4 + 9700)/10004
  = 97,00%

Model s slabšo klasifikacijsko točnostjo je boljši.

Priklic, natančnost in F1

- Mere za posamezen razred
  - Priklic (recall)
    - Število pravilno klasificiranih pozitivnih primerov izmed vseh dejensko pozitivnih primerov
  - Natančnost (precision)
    - Število pravilno klasificiranih pozitivnih primerov izmed vseh napovedanih pozitivnih primerov
  - F1
    - Harmonično povprečje priklica in natančnosti $F_1 = 2 * \frac{precision * recall}{precision + recall}$
- Mere lahko povprečimo po razredih (macro average) ali utežimo po primerih (micro average)

Priklic, natančnost in F1

		Predicted class		Total instances
		+	−	
Actual class	+	TP	FN	P
	−	FP	TN	N

True Positive Rate or Hit Rate or Recall or Sensitivity or TP Rate	TP/P	The proportion of positive instances that are correctly classified as positive
Precision or Positive Predictive Value	TP/(TP+FP)	Proportion of instances classified as positive that are really positive
F1 Score	(2 × Precision × Recall) /(Precision + Recall)	A measure that combines Precision and Recall
Accuracy or Predictive Accuracy	(TP + TN)/(P + N)	The proportion of instances that are correctly classified

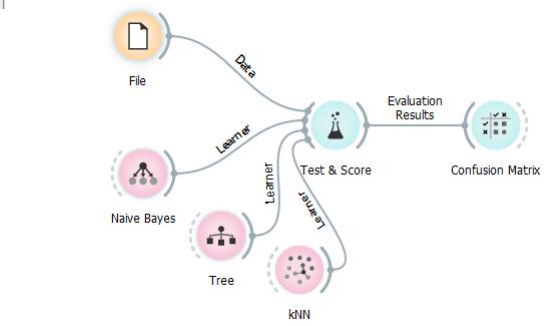
- Priklic
- Natančnost
- Mera F1
- Klasifikacijska točnost

Evalvacija klasifikacije v Orange

- AUC
  - Area under curve
  - AUROC
  - Površina pod ROC krivuljo
- CA – classification accuracy
  - Klasifikacijska točnost
- F1 – harmonično povprečje priklica in natančnosti
- Precision – natančnost
- Recall - priklic

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
kNN	0.951	0.845	0.823	0.835	0.845
Naive Bayes	0.971	0.863	0.858	0.859	0.863
Tree	0.991	0.951	0.951	0.951	0.951



Naloge

- Novi izrazi…
- Metrika ali metoda?
  - Klasifikacijska točnost, uporaba testne množice, specifičnost, AUC, prečno preverjanje, randomizacija, mera F1, površina pod ROC krivuljo
- Kaj je namen evalvacije?
- Kaj bi dobili, če bi testirali na učni množici?
- Kdaj je smiselno uporabiti elevacijske metrike za posamezen razred?
- Je klasifikacijska točnost 87% dobra?

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Četrti sklop

Klasifikacija 2

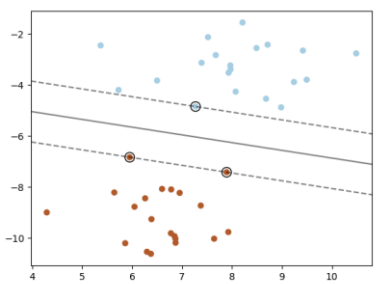
- Jezikovna pristranskost
- Pretirano prilagajanje učni množici
- Naivni Bayesov klasifikator
- SVM

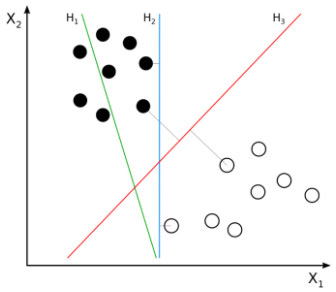
Jezikovna pristranskost

- Jezikovna pristranskost določa jezik dovoljenih hipotez pri posameznem modelu (algoritmu).
- Primeri:
  - Odločitvena drevesa lahko v pogojih primerjajo samo s konstantami (in ne z drugimi spremenljivkami - atributi)
  - SVM z linearnim jedrom poišče hiperravnino, ki najbolje loči med primeri (ne more pa vijugati)
- Z izborom določenega tipa modela se odločimo tudi za njegove omejitve.

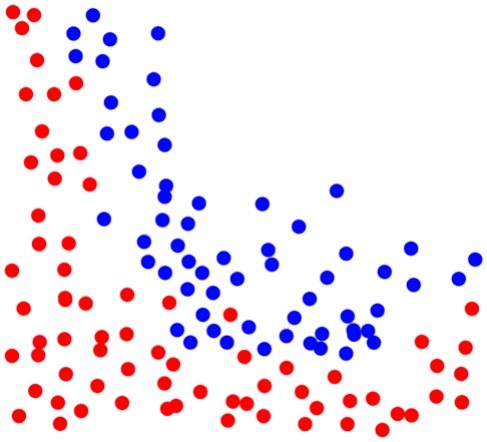
Metoda podpornih vektorjev* (SVM)

- SVM = support vector machines
- Poišči premico (hiperravnino), ki loči med razredoma in ima pri tem največji rob (margin).
- SVM uporablja jedrne funkcije (kernel trick), da preslika podatke iz vhodnega prostora (ang. input space) v nek višjedimenzijski prostor značilk (ang. feature space), v katerem so primeri linearno ločljivi

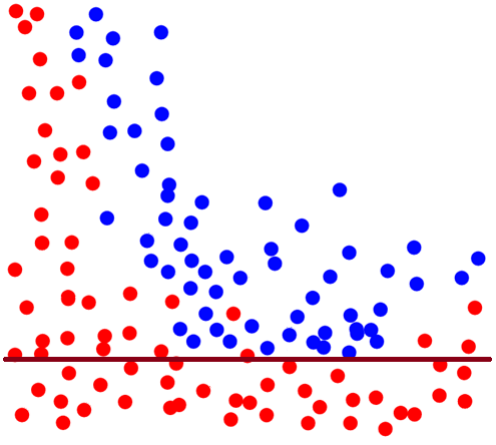




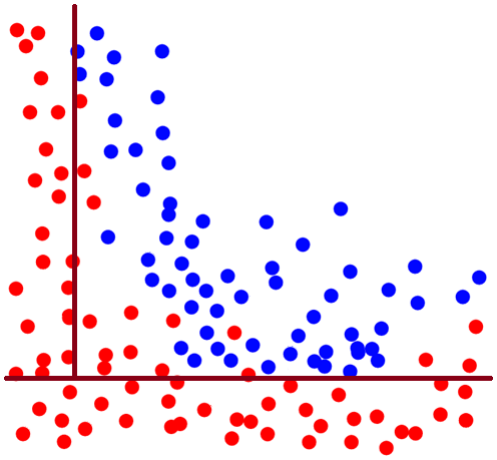
Ločiti modre od rdečih



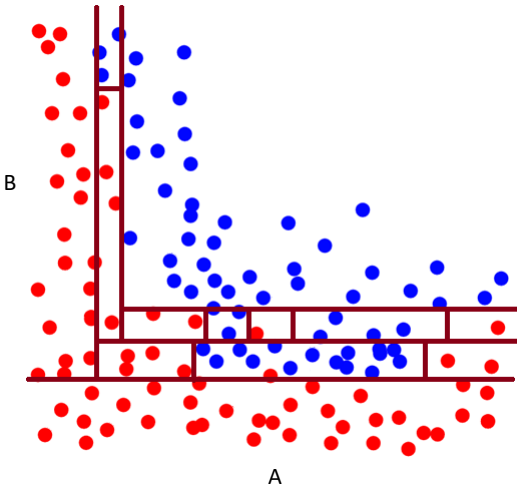
Gradnja odločitvenega drevesa …



Gradnja odločitvenega drevesa …

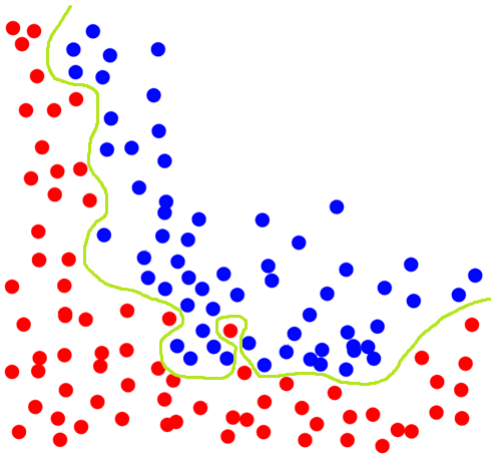


Gradnja odločitvenega drevesa …

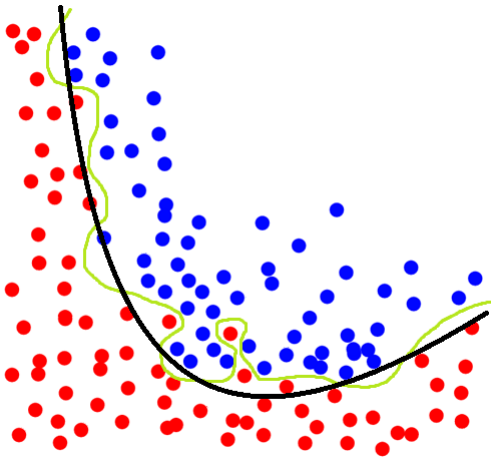


- Jezikovna pristranskost
  - Odločitvena drevesa imajo samo pogoje, kjer attribute primerjajo s konstantami (Samo vodoravne in navpične delitve, npr A > 1/4)
  - Odločitvena drevesa nimajo pogojev tipa A>B
- Ta model se pretirano prilagaja učni množici

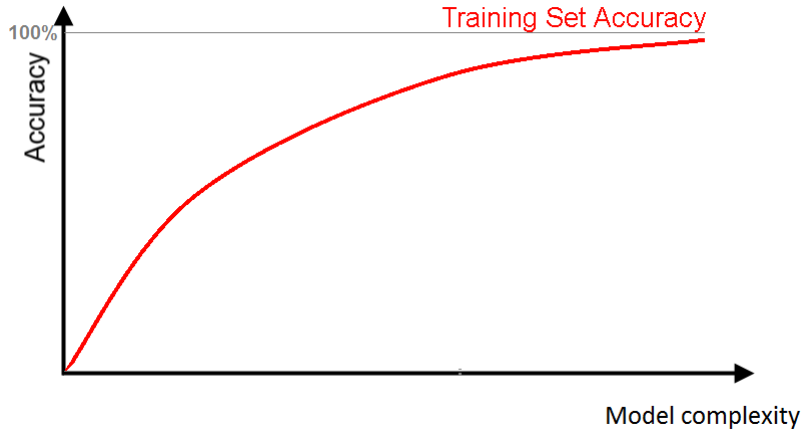
Tudi drugi modeli se lahko pretirano prilagajajo učni množici (primer na sliki: SVM)



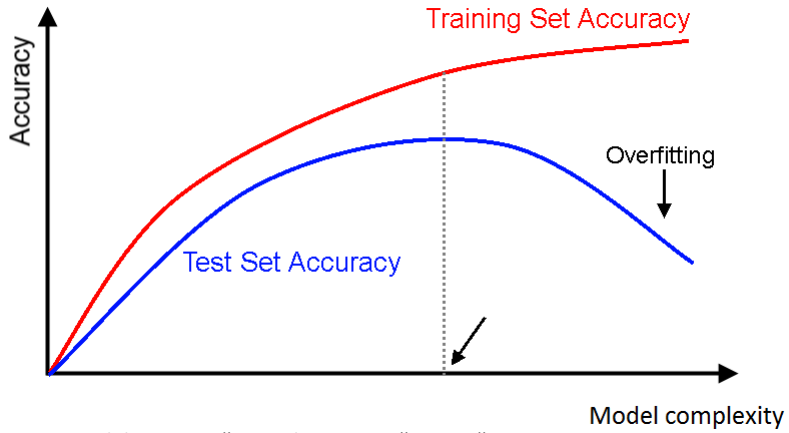
Tudi drugi modeli se lahko pretirano prilagajajo učni množici (primer na sliki: SVM)



Kompleksnost modela in performansa na učni množici



Performansa na testni množici



Z večanjem kompleksnosti modela se povečuje prilagajanje učni množici
- Pretirano prilagajanje učni množici – model se nauči tudi šuma, zato slabo klasificira nove primere
- Z regularizacijo (npr primerno rezanje drevesa) dobimo bolj interpretabilne modele, ki bolje delujejo na novih primerih

Naivni Bayesov klasifikator

Ideja Naivnega Bayesovega klasifikatorja

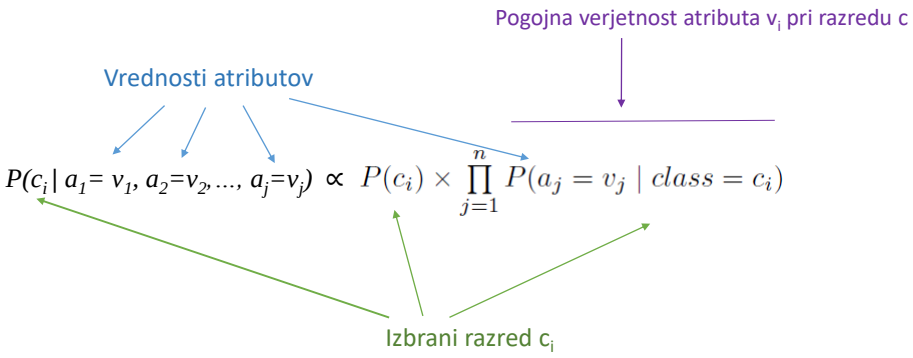
- Zanima nas, kakšna je verjetnost razreda C pri podanih vrednostih atributov X_1, X_2, X_3, …. , X_n

$$P(C|X_1X_2 \dots X_n)$$

- „**Naivno**“ predpostavimo, da so atributi med seboj verjetnostno neodvisni

$$P(X_1X_2 \dots X_n|C) \approx P(X_1|C) \cdot P(X_2|C) \cdot \dots \cdot P(X_n|C)$$
$$P(X_1X_2 \dots X_n) \approx P(X_1) \cdot P(X_2) \cdot \dots \cdot P(X_{n-1}) \cdot P(X_n)$$

Naivni Bayesov klasifikator



To ni verjetnost (ker se verjetnosti vseh razredov ne seštejejo v 1)! Formula je poenostavljena za hitrejšo implementacijo, je pa rezultat sorazmeren z verjetnostjo razreda pri danih vrednostih atributov.

Vaja: Naivni Bayesov klasifikator

Color	Size	Time	Caught
black	large	day	YES
white	small	night	YES
black	small	day	YES
red	large	night	NO
black	large	night	NO
white	large	night	NO

$$P(c_i \mid a_1 = v_1, a_2 = v_2, \ldots, a_j = v_j) \propto P(c_i) \times \prod_{j=1}^{n} P(a_j = v_j \mid class = c_i)$$

- Ali pajek ujame belo mravljo ponoči?
- Ali pajek ujame črno veliko mravljo podnevi?

Vaja: Naivni Bayesov klasifikator

Ali pajek ujame belo mravljo ponoči?

Color	Size	Time	Caught
black	large	day	**YES**
white	small	night	**YES**
black	small	day	**YES**
red	large	night	**NO**
black	large	night	**NO**
white	large	night	**NO**

$$P(c_i|\, a_1=v_1, a_2=v_2, \ldots, a_j=v_j) \propto P(c_i) \times \prod_{j=1}^{n} P(a_j = v_j \mid class = c_i)$$

$$v_1 = \text{“}Color = white\text{”}$$
$$v_2 = \text{“}Time = night\text{”}$$
$$c_1 = YES$$
$$c_2 = NO$$

$$\begin{aligned} P(C_1|v_1, v_2) &= \\ &= P(\text{YES}|C = w, T = n) \\ &= P(\text{YES}) \cdot P(C = w|\text{YES}) \cdot P(T = n|\text{YES}) \\ &= \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{1}{3} \\ &= \frac{1}{18} \end{aligned}$$

$$\begin{aligned} P(C_2|v_1, v_2) &= \\ &= P(\text{NO}|C = w, T = n) \\ &= P(\text{NO}) \cdot P(C = w|\text{NO}) \cdot P(T = n|\text{NO}) \\ &= \frac{1}{2} \cdot \frac{1}{3} \cdot 1 \\ &= \frac{1}{6} \end{aligned}$$

Vaja: Naivni Bayesov klasifikator

Ali pajek ujame črno veliko mravljo podnevi?

Color	Size	Time	Caught
black	large	day	**YES**
white	small	night	**YES**
black	small	day	**YES**
red	large	night	**NO**
black	large	night	**NO**
white	large	night	**NO**

$$P(c_i|\, a_1=v_1, a_2=v_2, \ldots, a_j=v_j) \propto P(c_i) \times \prod_{j=1}^{n} P(a_j = v_j \mid class = c_i)$$

Ant 2: Color = black, Size = large, Time = day

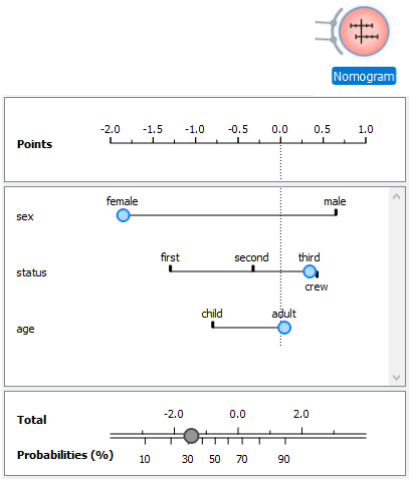
$$v_1 = \text{“}Color = black\text{”} = \text{“}C = b\text{”}$$
$$v_2 = \text{“}Size = large\text{”} = \text{“}S = l\text{”}$$
$$v_3 = \text{“}Time = day\text{”} = \text{“}T = d\text{”}$$
$$c_1 = YES$$
$$c_2 = NO$$

$$\begin{aligned} P(C_1|v_1, v_2, v_3) &= \\ &= P(\text{YES}|C = b, S = l, T = d) \\ &= P(\text{YES}) \cdot P(C = b|\text{YES}) \cdot P(S = l|\text{YES}) \cdot P(T = d|\text{YES}) \\ &= \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} \\ &= \frac{4}{54} = \frac{2}{27} \end{aligned}$$

$$\begin{aligned} P(C_2|v_1, v_2, v_3) &= \\ &= P(\text{NO}|C = b, S = l, T = d) \\ &= P(\text{NO}) \cdot P(C = b|\text{NO}) \cdot P(S = l|\text{NO}) \cdot P(T = d|\text{NO}) \\ &= \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{3}{3} \cdot 0 \\ &= 0 \end{aligned}$$

Uporaba naivnega Bayesa v praksi

- Pogosto uporabljen v praksi
  - Diagnostika v medicini
    - Ker so atributi izbrani tako, da so čimbolj neodvisni
    - Ni občutljiv na manjkajoče vrednosti
  - Klasifikacija teksta (atributi so besede)
    - Klasifikacija novic v kategorije
    - Detekcija neželene pošte (spam)
  - ….
- Zakaj?
  - Enostaven
  - Neobčutljiv na manjkajoče vrednosti
  - Uporabi vse razpopožljive atribute
  - Malo (brez) parametrov
  - Vizualizacija z nomogramom

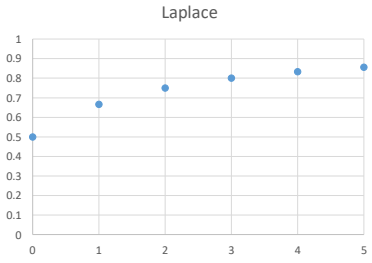


Ocenjevanje verjetnosti

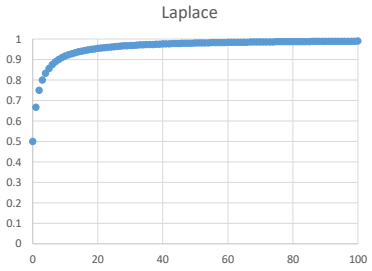
Ocenjevanje verjetnosti

- Če vržemo kovanec enkrat in pade cifra, je v 100% primerov padla cifra. Ne bi pa trdili, da je verjetnost, da pade cifra 100%. Tudi če dvakrat zapored pade cifra, ne bi trdili, da je 100% verjetno, da pade cifra.
- V strojnem učenju pogosto ocenjujemo verjetnosti iz malih (pod)množic podatkov:
  - V peti globini odločitvnega drevesa imamo samo še cca 1/32 vseh učnih primerov.
  - Pri Naivnem Bayesu pri ocenjevanju pogojnih verjetnosti atributov (npr. koliko je verjetno, da se beseda "evalvacija" pojavi v kategoriji "črna kronika")
- Poleg relativne frekvence, ki je dobra za oceno verjetnosti na velikih množicah, uporabljamo še Laplaceovo oceno in m-oceno, ki pri majhnih množicah ocenjujejo bliže apriorni distribuciji, pri velikih se pa asimptotično približujejo relativni frekvenci.

Laplaceova ocena - konvergenca



0 do 5 pozitivnih poskusov



0 do 100 pozitivnih poskusov (pri sto zaporednih metih kovanca pade cifra)

**m-ocena je nadgradnja Laplaceove ocene, kjer za apriorno verjetnost vzamemo dejansko distribucijo v učni množici.

Naloge

- Dopolni besedišče …
- Kaj je jezikovna pristranskost modela?
- Primerjaj odločitvena drevesa in Naivni Bayesov klasifikator.
- Kako evalviramo Naivi Bayesov klasifikator? Metode, metrike.
- Oceni verjetnosti z relativno frekvenco in z Laplaceovo oceno:

Število dogodkov		**Relativna frekvenca**		**Laplaceova ocena**	
tipa C1	tipa C2	P(C1)	P(C2)	P(C1)	P(C2)
0	2				
12	88				
12	988				
120	880				

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Peti sklop

Numerična predikcija

- Algoritmi
- Metrike za evalvacijo

Numerična predikcija

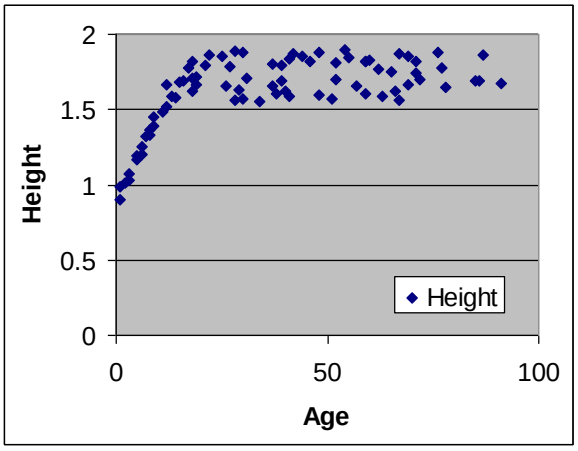
… napovedujemo številske vrednosti

- Ciljna spremenljivka je numerična
- Pogosto uporabljamo izraz regresija

- Primer: napovedujemo telesno višino

Primer

Podatki o starosti (Age: x-os) in telesni višini (Height: y-os) ljudi



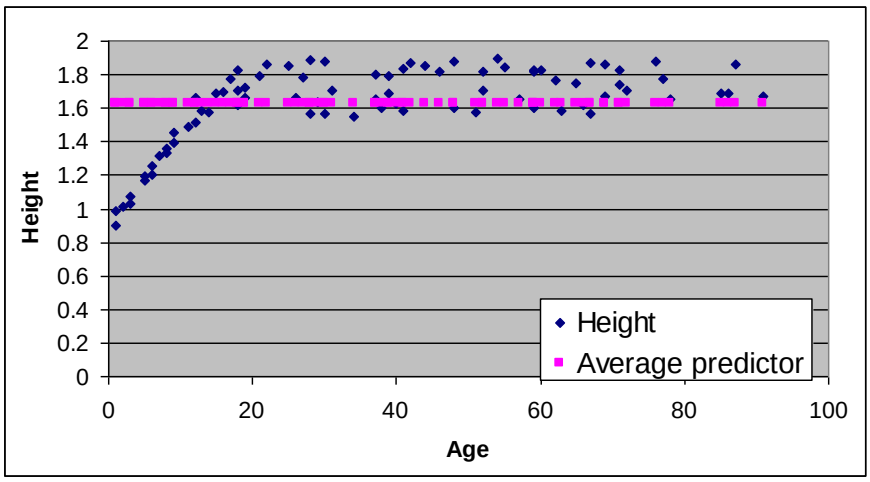
Age	Height
3	1.03
5	1.19
6	1.26
9	1.39
15	1.69
19	1.67
22	1.86
25	1.85
41	1.59
48	1.60
54	1.90
71	1.82
…	…

Testna množica

Age	Height
2	0.85
10	1.4
35	1.7
70	1.6

Najenostavnejši model: povprečje

- Vedno napove povprečje ciljne spremenljivke



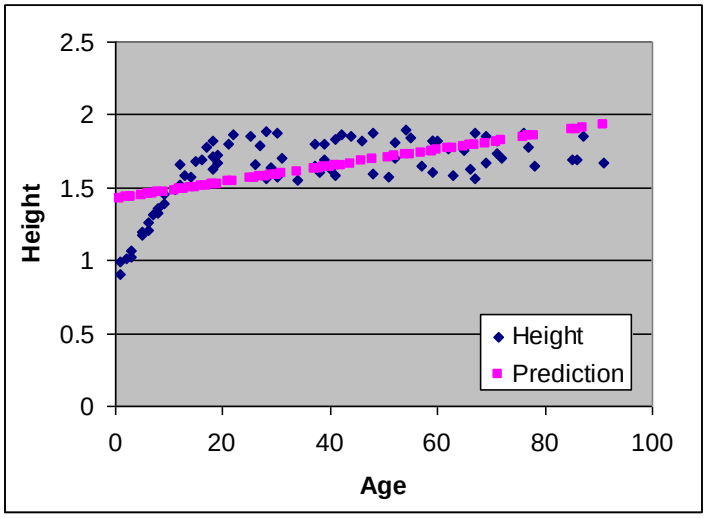
Napovedovanje s povprečjem

Povprečje ciljne spremenljivke je 1.63.

Age	Height	Baseline
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Linearna regresija

Height = 0.0056 * Age + 1.4181

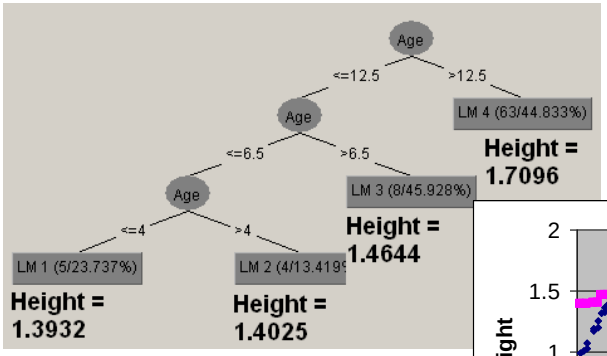


Napovedovanje za linearno regresijo

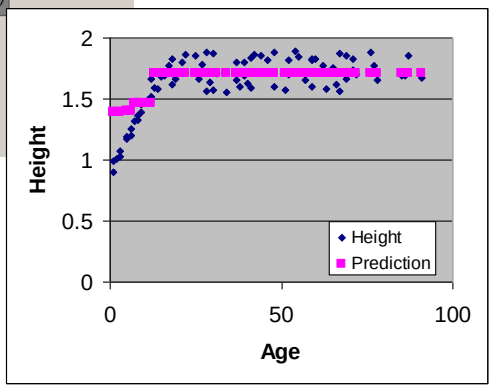
Height = 0.0056 * Age + 1.4181

Age	Height	Linear regression
2	0.85	
10	1.4	
35	1.7	
70	1.6	

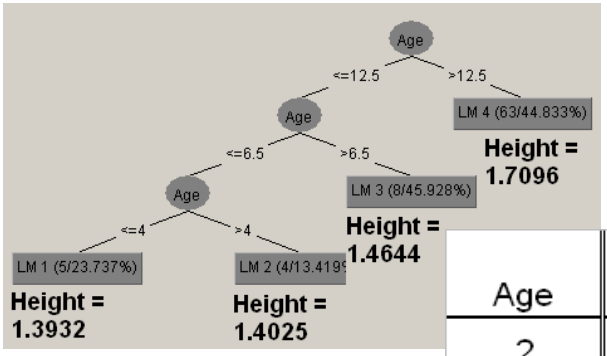
Regresijsko drevo



Napove povprečno vrednost ciljne spremenljivke primerov, ki so v učni množici v listu.



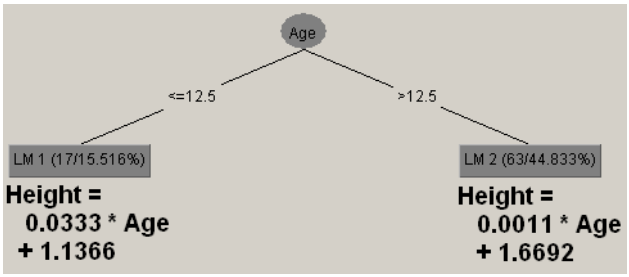
Napovedovanje z regresijskim drevesom

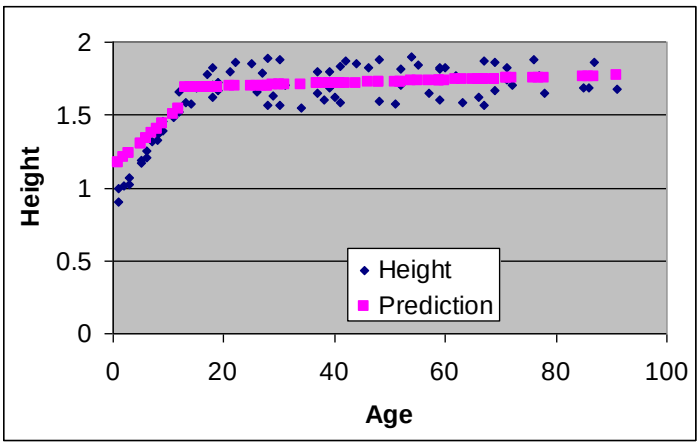


Age	Height	Regression tree
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Modelno drevo (Model tree)

V listih ima model.



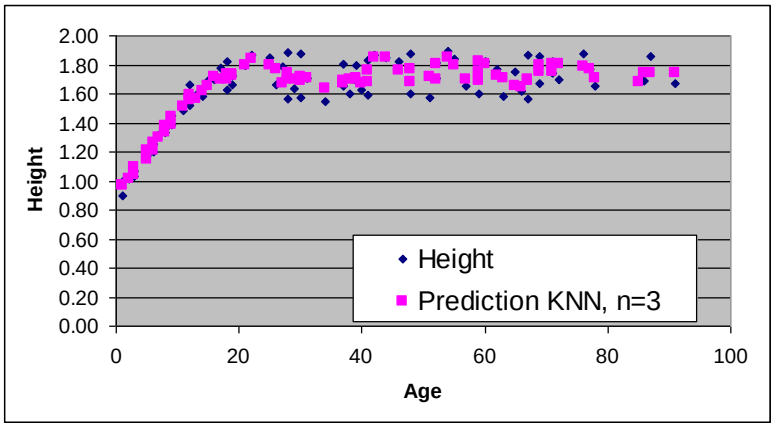


Napovedovanje z modelnim drevesom

Age	Height	Model tree
2	0.85	
10	1.4	
35	1.7	
70	1.6	

K najbližjih sosedov (KNN – K nearest neighbors)

- Napove povprečje na podlagi K najbližjih sosednjih primerov v učni množici (najbolj podobnih po vrednosti atributov)
- Majhen K – tesno prilagajanje učnim primerom, večji K – boljše posploševanje
- V tem primeru: K=3



Napovedovanje s KNN

Age	Height
1	0.90
1	0.99
2	1.01
3	1.03
3	1.07
5	1.19
5	1.17

Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Napovedovanje s KNN

Age	Height
8	1.36
8	1.33
9	1.45
9	1.39
11	1.49
12	1.66
12	1.52
13	1.59
14	1.58

Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Napovedovanje s KNN

Age	Height
30	1.57
30	1.88
31	1.71
34	1.55
37	1.65
37	1.80
38	1.60
39	1.69
39	1.80

Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Napovedovanje s KNN

Age	Height
67	1.56
67	1.87
69	1.67
69	1.86
71	1.74
71	1.82
72	1.70
76	1.88

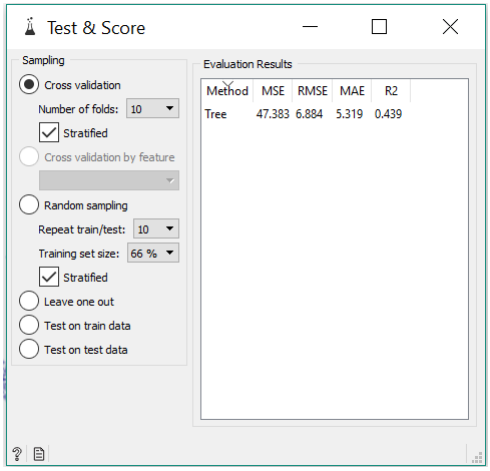
Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Kateri model je najboljši?

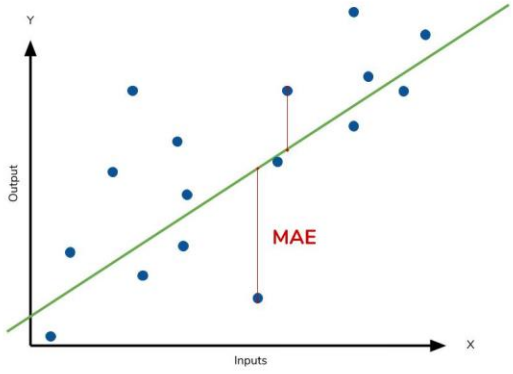
Age	Height	Baseline	Linear regression	Regression tree	Model tree	kNN
2	0.85	1.63	1.43	1.39	1.20	1.00
10	1.4	1.63	1.47	1.46	1.47	1.44
35	1.7	1.63	1.61	1.71	1.71	1.67
70	1.6	1.63	1.81	1.71	1.75	1.77

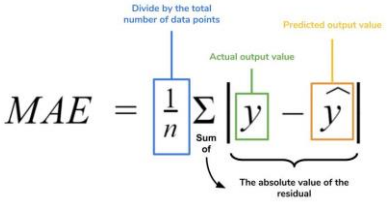
Metrike za oceno napake

- MSE – mean squared error
  - srednja kvadratna napaka
- RMSE – root mean squared error
  - koren srednje kvadratne napake
- MAE – mean absolute error
  - srednja absolutna napaka
- Korelacijski koeficient R^2



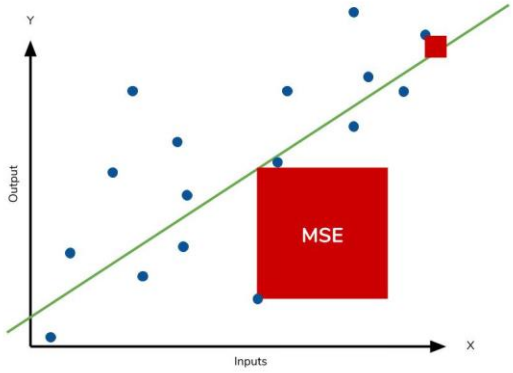
MAE: Mean absolute error

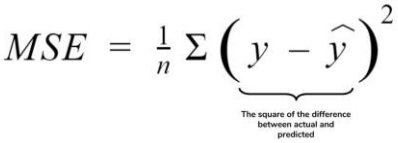




Povprečje razlik med napovedmi in podatki.
Enota napake je enaka enoti ciljne spremenljivke.

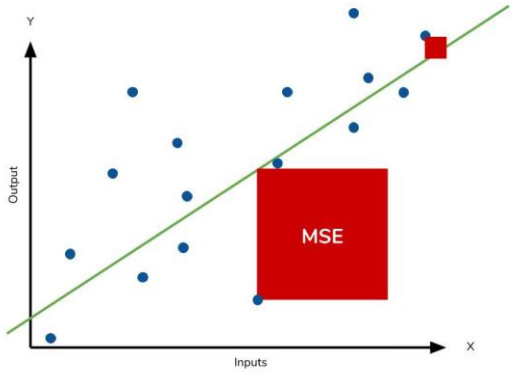
MSE: Mean squared error





Povprečje kvadratov razlik med napovedmi in podatki.
Večje napake bistveno več prispevajo k napaki.
Kvadratne enote napake.

RMSE: Root mean square error

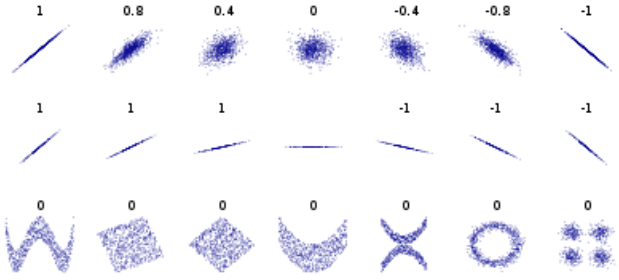


$$RMSE = \sqrt{MSE}$$

Povprečje kvadratov razlik med napovedmi in podatki.
Enote napake so enake enotam ciljne spremenljivke.

Korelacijski koeficient

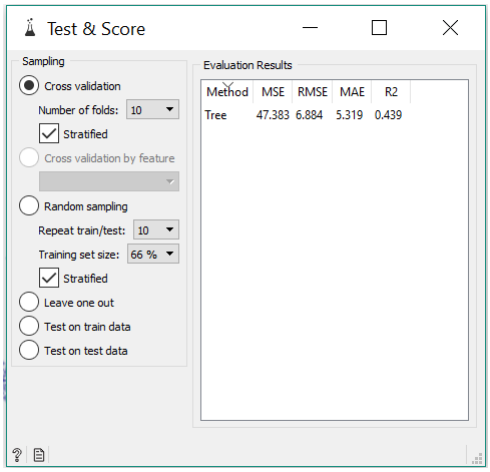
- Pearsonov koeficient korelacije je matematična in statistična številska mera, ki predstavlja **velikost linearne povezanosti** spremenljivk X in Y.



Osi na slikah so dejanska in napovedana vrednost, podobno kot kontingenčna tabela pri klasifikaciji.
Nima enote.

Mere za ocenjevanje napake pri numerični predikciji v Orange

- MSE – mean squared error
  – srednja kvadratna napaka
- RMSE – root mean squared error
  - koren srednje kvadratne napake
- MAE – mean absolute error
  – srednja absolutna napaka
- Korelacijski koeficient R^2



Naloge

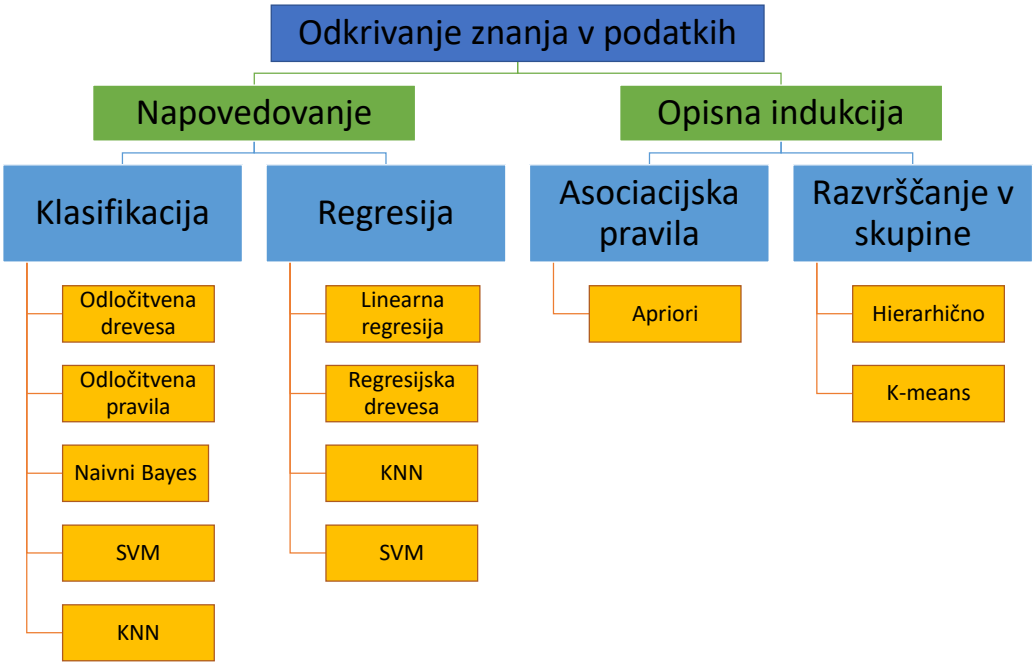
- Ali lahko KNN uporabimo za klasifikacijo?
- S katerimi metodami evalviramo regresijske modele?
- Več je bolje ali manj je bolje?
  - MAE, MSE, RMSE, R2
  - Klasifikacijska točnost, natančnost, priklic, mera F1
- Enote
  - MAE, MSE, RMSE, R2
  - Klasifikacijska točnost, natančnost, priklic, mera F1

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Šesti sklop

Asociacijska pravila



Asociacijska pravila - primer

- 25 trgovin Osco Drug store
- Analiza nakupov 1.2 milijona tranksakcij
- Nepričakovani vzorec
  - med 17. in 19. uro popoldne se pojavlja vzorec **plenice → pivo**
- Razlaga
  - moški, ki so jih žene popoldne poslale kupit plenice, so si kupili tudi pivo...

Vir: http://www.dssresources.com/newsletters/66.php

Asociacijska pravila

- Podatkovno rudarjenje (data mining)
  - Najti nekaj vrednega, ki je skrito (zakopano) v podatkih
- Nimamo ciljne spremenljivke (ne napovedujemo)
- Iščemo zanimive vzorce, povezave

- Tipični primer uporabe
  - podatki o prodaji (t.i. market basket analysis)
  - atributi: vsi artikli v trgovini
  - primeri: nakupi strank
  - vrednosti atributov so količine artiklov (običajno samo 1 in 0 --- je kupil, ni kupil)

Primer

- Podatki iz trgovine
  - kupec 1: kruh, maslo, banane, pivo, salama, sir, mleko
  - kupec 2: moka, mleko, olje, jabolka, kruh, jajca
  - kupec 3: sir, paradižnik, olive, jogurt, sladoled, vino
  - kupec 4: krompir, čebula, paprika, sir, olje, sol, moka, čokolada, kruh
  - kupec 5: kruh, sir, salama, pomaranče, mleko, napolitanke, cigarete
- Cilji analize
  - Kaj kupci kupujejo skupaj?
  - Kateri artikli pogojujejo nakup drugih artiklov?

Asociacijska pravila

- So pravila oblike **X → Y** , kjer sta X in Y množici postavk (*items*)
- Naloga učnega algoritma je poiskati **vsa** povezovalna pravila, ki so dovolj pogosta, kar izračunamo s podporo (*support*) in zanesljiva, kar izračunamo z zaupanjem (*confidence*).

$$Support(X \rightarrow Y) = \frac{število(XinY)}{število\_transakcij}$$

$$Confidence(X \rightarrow Y) = \frac{število(XinY)}{število(X)}$$

Podpora in zaupanje

- imamo ***n*** transakcij in pravilo $A \rightarrow B$
- podpora:

$$supp(A) = \frac{|A|}{n}$$

$$supp(A \rightarrow B) = \frac{|A \wedge B|}{n}$$

- zaupanje:

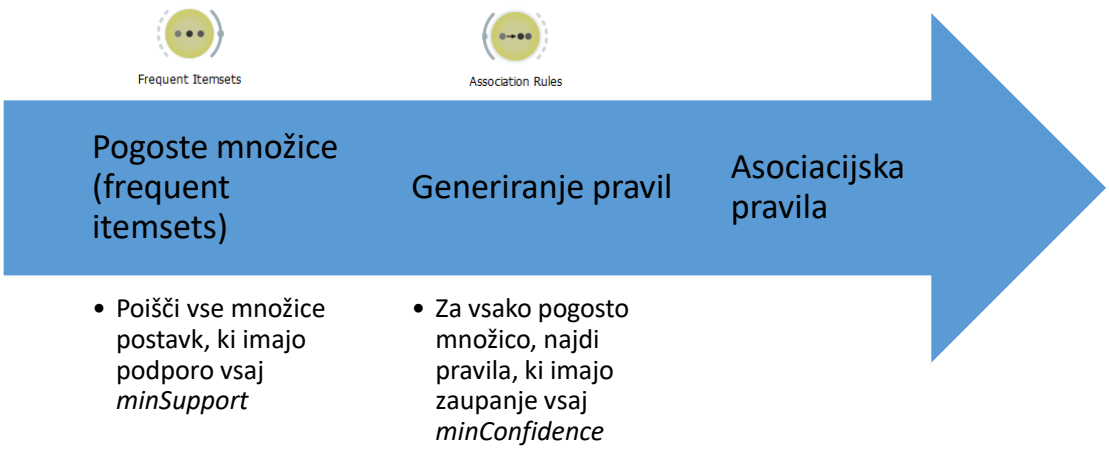
$$conf(A \rightarrow B) = \frac{|A \wedge B|}{|A|} = P(B|A)$$

*A in B sta množici, ki mata lahko več elementov

Težek problem

- V praksi
  - ogromno število transakcij (npr. več milijonov)
  - veliko število atributov - postavk (npr. več deset tisoč)
- Preveč možnih kombinacij
  - Za 10000 atributov je možnih n-terk $2^{10000} - 1$
  - Če je na prodaj 10000 artiklov, je možnih $2^{10000} - 1$ različnih nakupov
- Rešitev
  - algoritem *Apriori*

Apriori



Intuicija Apriorija: pogoste množice

- V *n* transakcijah se pojavi par {rokavice, šal} (lahko je še kaj zraven)
- V koliko transakcijah se lahko pojavi {rokavice, šal, kapa}?

- Kvečjemu *n.*

{rokavice, šal, kapa} se lahko pojavi n-krat samo, če so vsi, ki so kupili rokavice in šal, kupili tudi kapo. Večinoma pa se, če dodamo artikel, število transakcij, v katerih se množica pojavi, zmanjša.

- Podpora (supoort) je mera, ki pove, v koliko % transakcijah se določena množica pojavi.
- Podpora je anti-monotona $\forall A, B : A \subseteq B \Rightarrow supp(A) \geq supp(B)$

  A = {rokavice, šal}

  B = {rokavice, šal, kapa}

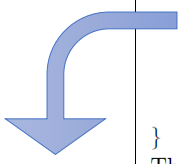
Apriori: iskanje pogostih množic

- Množice predstavimo tako, da so elementi v množicah urejeni

1. Najprej zgradimo množice z enim elementom, ki imajo dovoljšno podporo
   a. Generiramo vse množice velikosti L iz množic velikosti L-1 tako, da združimo množice, ki imajo vse elemente, razen zadnjega, enake.
   b. Odstranimo množice, katerih podmnožice velikosti L-1 ni na prejšnjem nivoju
   c. Odstranimo množice, ki imajo premajhno podporo
2. Če so nastale nove množice, se vrni na korak a (L:= L+1)

Apriori

Iskanje pogostih množic



Create L_1 = set of supported itemsets of cardinality one
Set k to 2
while $(L_{k-1} \neq \emptyset)$ {
    Create C_k from L_{k-1}
    Prune all the itemsets in C_k that are not
        supported, to create L_k
    Increase k by 1
}
The set of all supported itemsets is $L_1 \cup L_2 \cup \cdots \cup L_k$

(Generates C_k from L_{k-1})
Join Step
Compare each member of L_{k-1}, say A, with every other member, say B, in turn. If the first $k-2$ items in A and B (i.e. all but the rightmost elements of the two itemsets) are identical, place set $A \cup B$ into C_k.
Prune Step
For each member c of C_k in turn {
Examine all subsets of c with $k-1$ elements
Delete c from C_k if any of the subsets is not a member of L_{k-1}
}

Apriori: Iskanje pogostih množic

- L1 (prvi nivo): množice z enim elementom, ki imajo dovoljšno podporo
- k=2 (k je številka nivoja)
- Dokler so množice na prejšnjem nivoju
  - Naredi množice C_k iz L_{k-1}
  - Odstrani vse množice, ki nimajo dovolj podpore
  - k=k+1

Create L_1 = set of supported itemsets of cardinality one
Set k to 2
while $(L_{k-1} \neq \emptyset)$ {
    Create C_k from L_{k-1}
    Prune all the itemsets in C_k that are not
        supported, to create L_k
    Increase k by 1
}
The set of all supported itemsets is $L_1 \cup L_2 \cup \cdots \cup L_k$

Apriori: genriranje naslednjega nivoja iz prejšnjega

- Množice so urejene!
- Join Step:
  - Združimo tiste množice, ki imajo enake vse elemente, razen zadnjega
- Prune Step:
  - Odstrani množico, če katere od njenih podmnožic ni na prejšnjem nivoju

(Generates C_k from L_{k-1})
Join Step
Compare each member of L_{k-1}, say A, with every other member, say B, in turn. If the first $k-2$ items in A and B (i.e. all but the rightmost elements of the two itemsets) are identical, place set $A \cup B$ into C_k.
Prune Step
For each member c of C_k in turn {
Examine all subsets of c with $k-1$ elements
Delete c from C_k if any of the subsets is not a member of L_{k-1}
}

Pravila iz pogostih množic

- Iščemo pravila z določenim zaupanjem
- Vse številke, ki jih potrebujemo pri računaju zaupanja, so v grafu pogostih množic (ne gremo gledat v bazo)
- Ni potrebno, da preverimo vsa možna pravila, ker velja
  Conf(A U B → C) >= Conf (A → B U C)

Max Bramer: Principles of data mining (2007) str. 214 – 216

Vaja: Pogoste množice

Z uporabo algoritma Apriori poišči pogoste množice s podporo vsaj 2/6.

Transakcija 1 : arašidi, banane, Coca-cola, datelji
Transakcija 2 : banane, Coca-cola
Transakcija 3 : banane, Coca-cola
Transakcija 4 : arašidi, Coca-cola, datelji
Transakcija 5 : arašidi, banane, datelji
Transakcija 6 : arašidi, banane, Coca-cola

Vaja: Povezovalna pravila

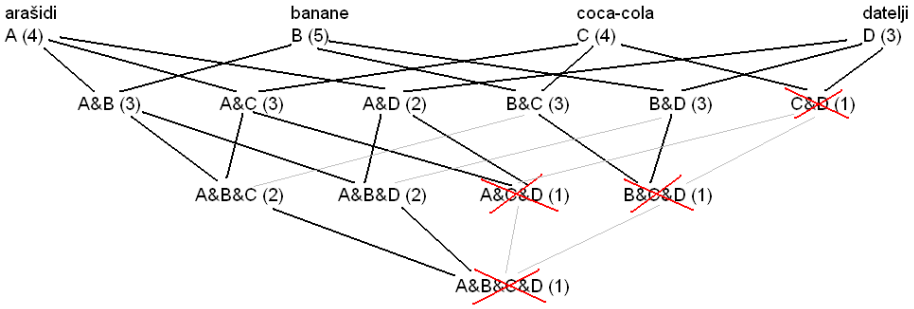
Z uporabo algoritma Apriori poišči povezovalna pravila s podporo vsaj 2/6 in zaupanjem vsaj 75% v spodnjih transakcijah.

Prepišemo iz transakcijske baze v predstavitev z atributi in primeri.

arašidi	banane	coca-cola	datelji
1	1	1	1
	1	1	
	1		1
1		1	
1	1		1
1	1	1	

Pogoste množice

arašidi	banane	coca-cola	datelji
1	1	1	1
	1	1	
	1		1
1		1	
1	1		1
1	1	1	



Povezovalna pravila …

A&B	**Support = 3/6**
A → B	Confidence = 2/4 = 50%
B → A	Conficence = 2/5 = 40%
A&C	**Support = 3/6**
A → C	Conficence = 3/4 = 75%
C → A	Conficence = 3/4 = 75%
A&D	**Support = 2/6**
A → D	Conficence = 2/4 = 50%
D → A	Conficence = 2/3 = 67%
B&C	**Support = 3/6**
B → C	Conficence = 3/5 = 60%
C → B	Conficence = 3/4 = 75%
B&D	**Support = 3/6**
B → D	Conficence = 3/5 = 60%
D → B	Conficence = 3/3 = 100%

Povezovalna pravila …

A&B&C	**Support = 2/6**
A → B&C	Confidence = 2/4 = 50%
A&B → C	Confidence = 2/3 = 67%
A&C → B	Confidence = 2/3 = 67%
B → A&C	Confidence = 2/5 = 40%
B&C → A	Confidence = 2/3 = 67%
C → A&B	Confidence = 2/4 = 50%
A&B&D	**Support = 2/6**
A → B&D	Confidence = 2/4 = 50%
A&B → D	Confidence = 2/3 = 67%
A&D → B	Confidence = 2/2 = 100%
B → A&D	Confidence = 2/5 = 40%
B&D → A	Confidence = 2/3 = 67%
D → A&B	Confidence = 2/3 = 67%

Lift in Leverage

- Lift (izboljšava): koliko bolj pogosto se skupaj pojavljata L in R, kot če bi bila neodvisna

$$\text{lift}(L \rightarrow R) = \frac{\text{support}(L \cup R)}{\text{support}(L) \times \text{support}(R)}$$

- Leverage (vzvod): razlika med podporo pravila in zmnožkom podpor leve in desne strani pravila

$$\text{leverage}(L \rightarrow R) = \text{support}(L \cup R) - \text{support}(L) \times \text{support}(R)$$

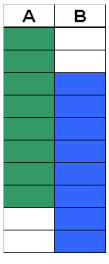
Mere kvalitete asociacijskih pravil v Orange

Supp	Conf	Covr	Strg	Lift	Levr	Antecedent		Consequent
0.050	0.178	0.283	0.618	1.017	0.001	Fresh Vegetables	→	Fresh Fruit
0.050	0.287	0.175	1.619	1.017	0.001	Fresh Fruit	→	Fresh Vegetables

Supp(L→R)	N(L U R)/n	Support	Podpora
Conf(L→R)	Supp(L U R) / Supp(L)	Confidence	Zaupanje
Covr(L→R)	Supp(L)	Coverage	Pokritost
Strg(L→R)	Supp(R) / Supp(L)	Strength	Moč
Lift(L→R)	Supp(L U R) / (Supp(L) * Supp(R))	Lift	Izboljšava
Levr(L→R)	Supp(L U R) - (Supp(L) * Supp(R))	Leverage	Vzvod

Naloge

- Besedišče …
- Kako algoritem Apriori uporabi atribut, ki ima veliko različnih vrednosti (npr mesec rojstva), če ima nastavitev minSupport = 10%?
- Kako isti atribut uporabi algoritem za gradnjo odločitvenih dreves?
- Kaj pa atribut „PersonID“?
- Kako prevedemo tabelarične podatke v transakcijsko obliko?
- Imamo atributa A in B, vsak od njiju ima vrednost “1” v 80% primerov. Oba hkrati imata vrednost “1” v najmanj primerih, kar se da (glej sliko). Kakšna bodo asociacijska pravila:
  - minSupport = 50%, min conf = 70%
  - minSupport = 20%, min conf = 70%
- Kaj pa če upoštevamo tudi negativne vrednosti?



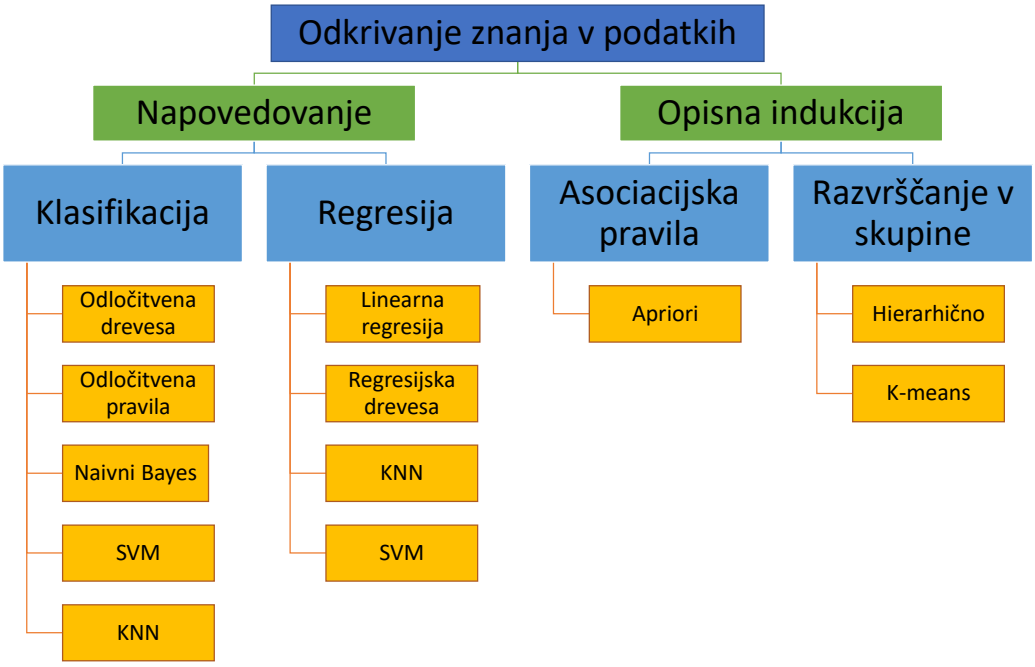
Fakulteta za informacijske študije
Faculty of information studies

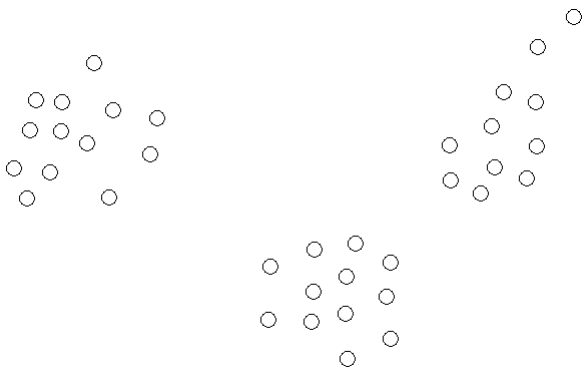
Odkrivanje znanja v podatkih

Sedmi sklop

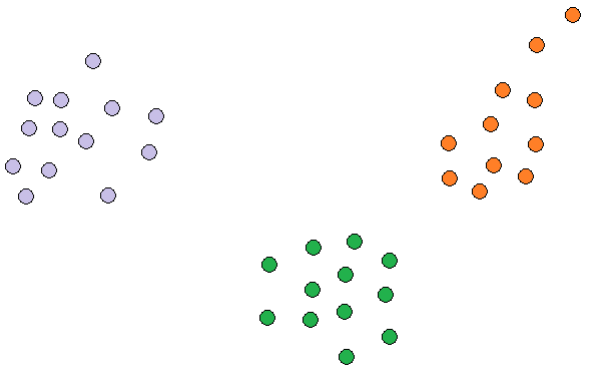
Razvrščanje v skupine

- K-means
- Hierarhično razvrščanje v skupine

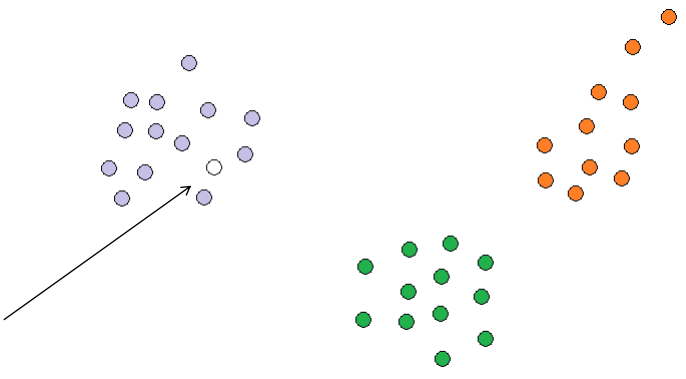

Odkrivanje znanja v podatkih
Napovedovanje
Opisna indukcija
Klasifikacija
Regresija
Asociacijska pravila
Razvrščanje v skupine
Odločitvena drevesa
Odločitvena pravila
Naivni Bayes
SVM
KNN
Linearna regresija
Regresijska drevesa
KNN
SVM
Apriori
Hierarhično
K-means



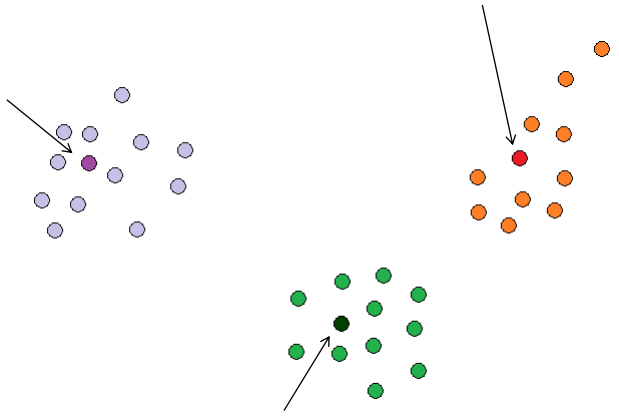
Skupine



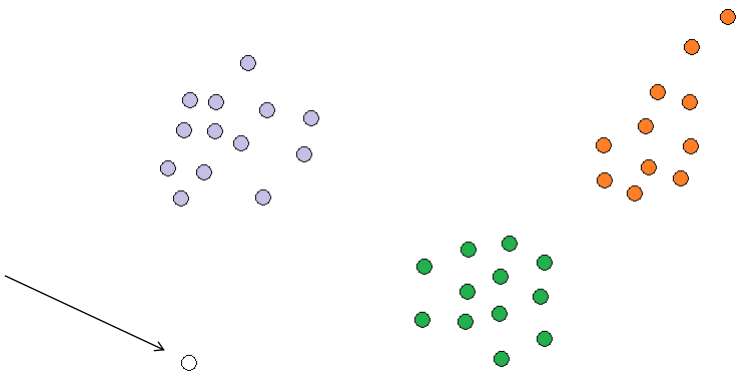
Nenadzorovana klasifikacija



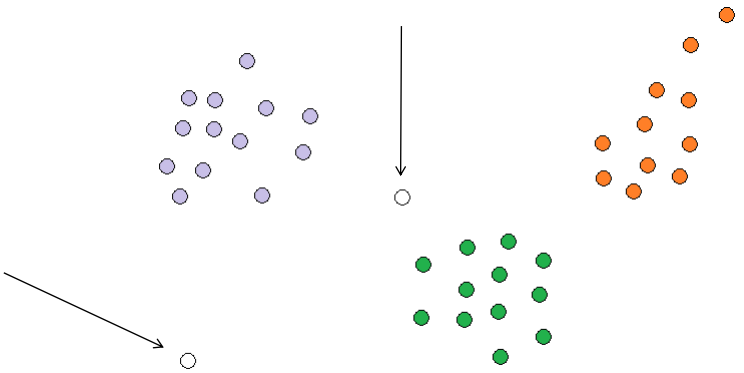
Povzemanje podatkov s centroidom/medoidom



Odkrivanje ubežnikov



Odkrivanje ubežnikov



Odkrivanje skupin

- Gručenje, clustering
- Odkrivanje skupin, ali razvrščanje podatkov v skupine, je ena od osnovnih tehnik podatkovne analitike.
  - Segmentacija uporabnikov: kakšni so tipični uporabniki, koliko različnih skupin
  - V medicini: tipične skupine bolnikov, da bi lahko njim prilagodili bolnišnični sistem
  - Skupine med seboj podobnih si dokumentov, slik
  - Odkrivanje osamelcev – taki, ki ne spadajo v nobeno skupino (morebitni vdori v računalniški sistem)

Grupiranje podobnih dokumentov

- „Jaguar“ je, odvisno od konteksta:
  - avto
  - klub
  - mačka
  - žival
  - restavracija
  - …

Tipi razvrščanja v skupine

- Particioniranje
  - k-Means, k-Medoids, k-Modes
- Hierarhično razvrščanje v skupine
  - Agglomerative hierarchical clustering
- Razvrščanje, ki temelji na mreži
  - Različne resolucije
  - Učinkovito in skalabilno
- Razvrščanje, ki temelji na gostoti primerov
  - Skupina je gosta množica točk, ki je od drugih skupin ločena z območjem redkih točk
  - Algoritmi: DBSCAN, OPTICS, DenClue

K-Means

Algoritem k-Means

1. Choose **k** random instances as cluster centers
2. Assign each instance to its closest cluster center
3. Recompute cluster centers by computing the average (aka *centroid*) of the instances pertaining to each cluster
4. If cluster centers have moved, go back to Step 2

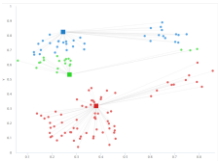
(Equivalent termination criterion: stop when assignment of instances to cluster centers has not changed)

Alternative: K-medoids, K-modes

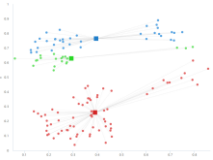
Interactive k-Means (Educational)

K-Means primer

Naključna inicializacija



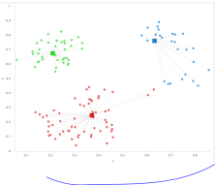
Računanje centroidov



Pripisovanje točk najbližjemu centroidu

Računanje centroidov

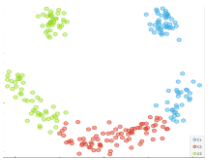
Pripisovanje točk najbližjemu centroidu

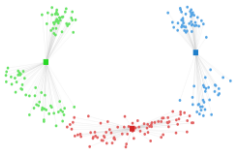


Računanje centroidov…

Lastnosti k-Means

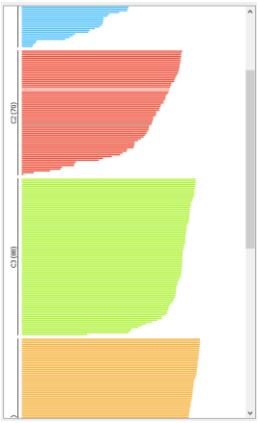
- V naprej določimo število skupin k
- Lahko konvergira v lokalni minimum (lahko ne najde dobre rešitve zaradi nesrečno izbranih začetnih točk)
- Išče „okrogle“ skupine



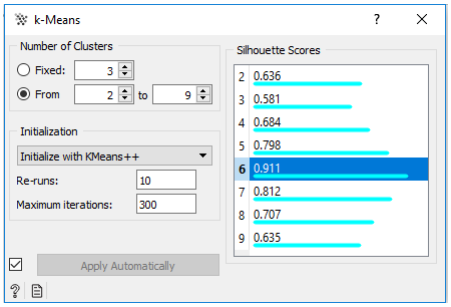


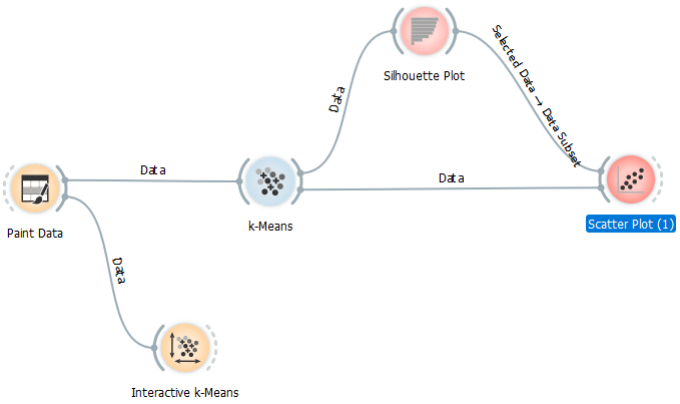
Silhuetni koeficient

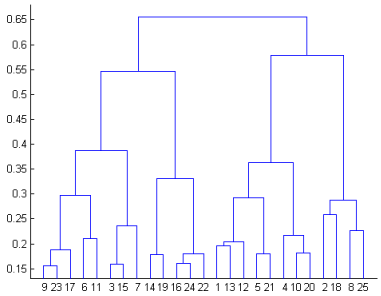
- Silhuetni koeficient je mera kvalitete razbitja, ki uspešno združuje tako kohezijo kot ločljivost.
- Za primer x_i je njegova silhueta enaka $s_i = (b_i - a_i)/\max(a_i, b_i)$
  - a_i povprečna razdalja primera x_i do vseh ostalih primerov v svoji skupini.
  - b_i je povprečna razdalja primera x_i do primerov v najbližji sosednji skupini
- Silhueta razbitja je enaka povprečni silhueti primerov v učni množici.
- Silhuetni koeficient lahko uporabljamo tudi za iskanje ubežnikov (outliers) v klasifikacijskih problemih.



k-Means + Silhuette + „reruns“







Hierarhično odkrivanje skupin

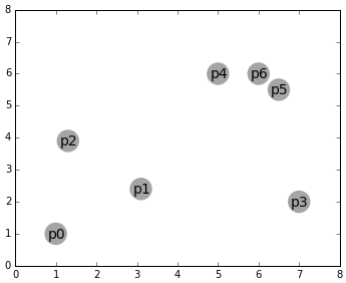
Agglomerative clustering

1. Start with a collection **C** of **n** singleton clusters
   - Each cluster contains one data point $\mathbf{c_i} = \{\mathbf{x_i}\}$
2. Repeat until only one cluster is left:
   1. Find a pair of clusters that is closest: min $\mathbf{D(c_i, c_j)}$
   2. Merge the clusters $\mathbf{c_i}$ and $\mathbf{c_j}$ into $\mathbf{c_{i+j}}$
   3. Remove $\mathbf{c_i}$ and $\mathbf{c_j}$ from the collection **C**, add $\mathbf{c_{i+j}}$

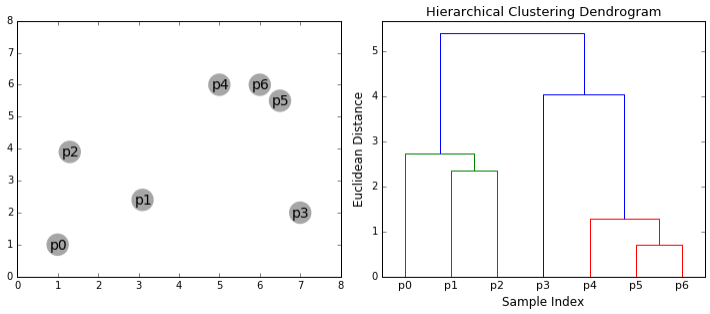
Nek novo ime skupine, ni vsota

- Time and space complexity
- Sensitive to noisy data

Hierarhično odkrivanje skupin: primer

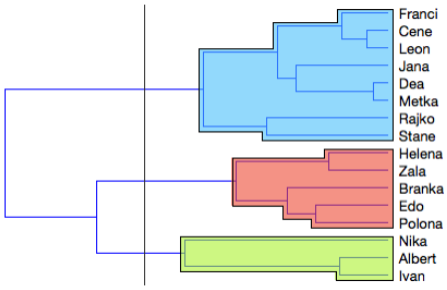


Agglomerative clustering - dendrogram

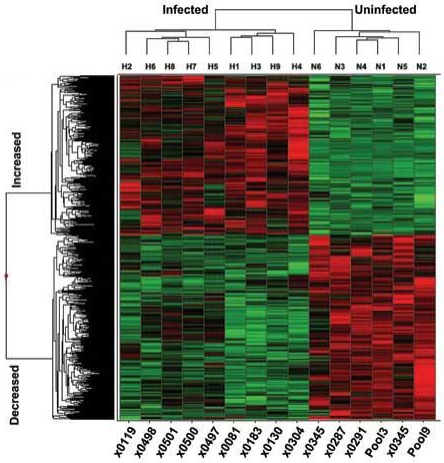


Dendrogram

- Postopek združevanja v skupine in rezultat tega postopka — hierarhijo skupin, lahko ponazorimo v drevesnem izrisu ali dendrogramu (gr. dendron pomeni drevo, gramma pa risba).
- Dendrogram je izrisan od leve proti desni. Stičišča skupin so od desnega roba odmaknjena skladno z razdaljo med skupinami.



Primer: Hierarhično razvrščanje v skupine genov



Omejitve hierarhičnega iskanja skupin

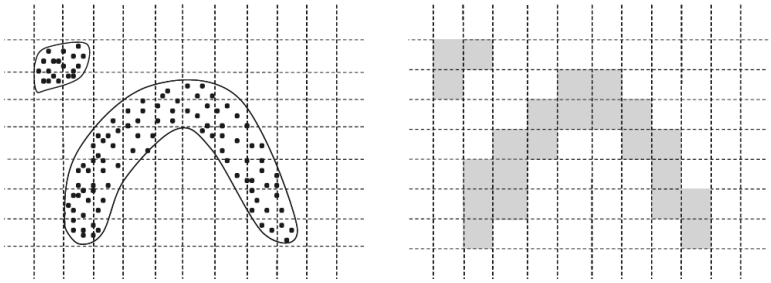
- Potrebuje razdalje med vsemi pari točk
- Velika časovna kompleksnost $O(n^3)$

Grid-based (parameters **p** and **τ**)

1. Discretize each dimension of **D** into **p** ranges
2. Determine dense grid cells at level **τ**
3. Create graph where dense grid cells are connected if they are adjacent
4. Determine connected components of graph
5. Return: points in each connected component as a cluster

Grid-based (parameters **p** and **τ**)

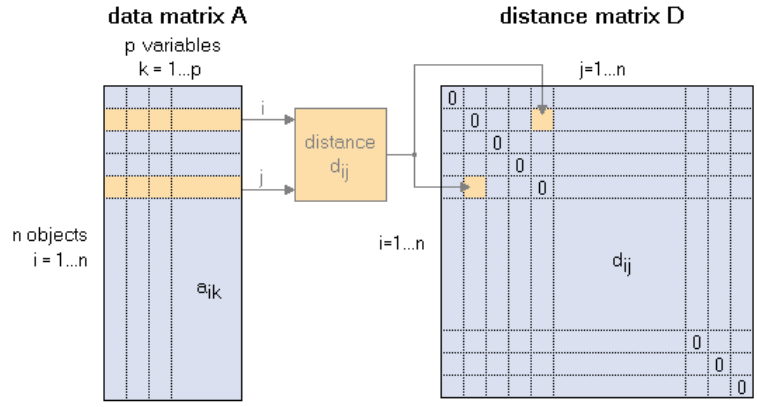
1. Discretize each dimension of **D** into **p** ranges
2. Determine dense grid cells at level **τ**
3. Create graph where dense grid cells are connected if they are adjacent
4. Determine connected components of graph
5. Return: points in each connected component as a cluster



Mere podobnosti in razdalje

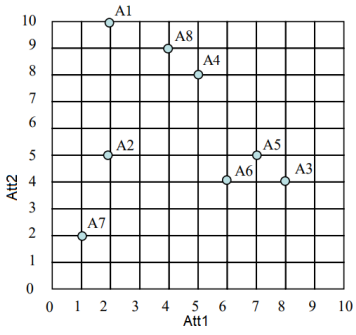
- Od podatkov zavisi, katero mero podobnosti/razdalje uporabimo:
  - Tipi atributov: binarni, kategorični, numerični
  - Gostota (npr marekt basket vs. tabelarični)
  - Število atributov
  - …

Matrika razdalj



Matrika razdalj - primer

	Att1	Att2
A1	2	10
A2	2	5
A3	8	4
A4	5	8
A5	7	5
A6	6	4
A7	1	2
A8	4	9



	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

Euclidian → $Dist(A,B) = \sqrt[2]{(Att1(A) - Att1(B))^2 + (Att2(A) - Att2(B))^2}$

Mere razdalje

Euclidean	$d(x,y) = \sqrt{\sum (x_i - y_i)^2}$
Squared Euclidean	$d(x,y) = \sum (x_i - y_i)^2$
Manhattan	$d(x,y) = \sum \lvert (x_i - y_i) \rvert$
Canberra	$d(x,y) = \sum \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$
Chebychev	$d(x,y) = \max(\lvert x_i - y_i \rvert)$
Bray Curtis	$d(x,y) = \frac{\sum \lvert x_i - y_i \rvert}{\sum x_i + y_i}$
Cosine Correlation	$d(x,y) = \frac{\sum (x_i y_i)}{\sqrt{\sum (x_i)^2 \sum (y_i)^2}}$
Pearson Correlation	$d(x,y) = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (y_i - \overline{y})^2} \sqrt{\sum (y_i - \overline{y})^2}}$
Uncentered Peason Correlation	$d(x,y) = \frac{\sum x_i y_i}{\sqrt{\sum (y_i - \overline{y})^2} \sqrt{\sum (y_i - \overline{y})^2}}$
Euclidean Nullweighted	Same as Euclidean, but only the indexes where both x and y have a value (not NULL) are used, and the result is weighted by the number of values calculated. Nulls must be replaced by the missing value calculator (in dataloader).

Minkowski distance

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer. (Chapter 3)

Naloge

- Besedišče …
- Razlika med razdaljo (distance) in podobnostjo (similarity)
- Kateri algoritmi temeljijo na razdaljah/podobnosti
- Kaj je dendrogram
- Kateri algoritem iskanja skupin je primeren za velike množice podatkov?
- Kako določimo primeren K pri K-means algoritmu?

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Osmi sklop

Rudarjenje besedil

- Od množice besedil do podatkov

Besedilo …

Rdeča kapica

Nekoč je živela ljubka deklica in vsi, ki so jo poznali, so jo imeli radi. Njena babica, ki je imela zanjo še prav poseben prostor v srcu, ji je podarila čudovito rdečo kapico. Tako navdušena je bila deklica nad kapico, da jo je nosila, kamorkoli je šla, zato so jo vsi klicali Rdeča kapica.

Nekega dne jo je mama poklicala k sebi in rekla: "Rdeča kapica, vzemi tole košarico in nesi svoji bolni babici nekaj hrane in pijače. Dobro ji bo delo. Pojdi skozi gozd, ne ustavljaj se in ne išči bližnjic!"

"V redu," je rekla Rdeča kapica, pograbila košarico in odhitela k babici, ki je živela v vasici na drugi strani gozda.

… podatki?

Rudarjenje besedil

- Iskanje in izbiranje informacij (information retrieval)
- Povzemanje
- Klasifikacija (classification)
- Odkrivanje skupin (clustering)
- Odkrivanje entitet in povezav med njimi

…

- Strojno prevajanje

Predprocesiranje in vektorizacija

Predprocesiranje

- Čiščenje (pdf, html → txt, odstranjevanje URL-jev, …)
- Tokenizacija (razdelimo na besede, odstranimo ločila)
- Lematizacija ali korenjenje (eno od tega)
- Filtriranje blokiranih besed
- N-grami
- Odstranjevanje preredkih besed
- Oblikoslovno označevanje (POS tagger)

Čiščenje

- Odstranjevanje formatiranja (pdf, html → txt)
- Odstranjevanje slik
- Odstranjevanje “boilerplate”: glava, številke strani,…
- …

Nekoč je živela ljubka deklica in vsi, ki so jo poznali, so jo imeli radi. Njena babica, ki je imela zanjo še prav poseben prostor v srcu, ji je podarila čudovito rdečo kapico. Tako navdušena je bila deklica nad kapico, da jo je nosila, kamorkoli je šla, zato so jo vsi klicali Rdeča kapica.

Nekega dne jo je mama poklicala k sebi in rekla: "Rdeča kapica, vzemi tole košarico in nesi svoji bolni babici nekaj hrane in pijače. Dobro ji bo delo. Pojdi skozi gozd, ne ustavljaj se in ne išči bližnjic!"

"V redu," je rekla Rdeča kapica, pograbila košarico in odhitela k babici, ki je živela v vasici na drugi strani gozda.

Tokenizacija

Tokenizacija je razdelitev besedil v manjše enote (navadno besede in ločila), ki jim pravimo žetoni. Gre za osnovno operacijo, ki se uporablja kot prvi korak pri praktično vsaki obdelavi besedila.

Nekoč je živela ljubka deklica in vsi, ki so jo poznali, so jo imeli radi. Njena babica, ki je imela zanjo še prav poseben prostor v srcu, ji je podarila čudovito rdečo kapico. Tako navdušena je bila deklica nad kapico, da jo je nosila, kamorkoli je šla, zato so jo vsi klicali Rdeča kapica.

Lematizacija

Lematizacija je proces transformacije besede v njeno osnovno obliko (tako, ki jo najdemo v slovarju). To je možno pri tistih besednih vrstah, ki so pregibne. Pri lematizaciji (za razliko od korenjenja) velja, da je končni produkt vedno slovnično pravilna beseda. Ta operacija je ključna pri skoraj vsaki obdelavi naravnega jezika, kjer je cilj razumevanje tega jezika, saj nam omogoča enačenje vseh pomensko enakih besed, čeprav se zaradi svoje pregibnosti v tekstu pojavljajo v različnih časih, sklonih in spregatvah.

* Ni v Bramerjevi knjigi, ker je to specifično za morfološko bogate jezike

Lematizacija

biti živeti ljubek ves poznati imeti rad Njen biti imeti zame
Nekoč je živela ljubka deklica in vsi, ki so jo poznali, so jo imeli radi. Njena babica, ki je imela zanjo še prav poseben
srce on biti podariti čudovit rdeč kapica Tak navdušen biti biti kapica dati biti nosilo
prostor v srcu, ji je podarila čudovito rdečo kapico. Tako navdušena je bila deklica nad kapico, da jo je nosila, kamorkoli
biti iti ves klicati Rdeč
je šla, zato so jo vsi klicali Rdeča kapica.

Nek dan biti poklicati se reči Rdeč vzeti tale košarica svoj bolan babica hrana
Nekega dne jo je mama poklicala k sebi in rekla: "Rdeča kapica, vzemi tole košarico in nesi svoji bolni babici nekaj hrane in
pijača Dober on biti iti ustavljati iskati bližnjica
pijače. Dobro ji bo delo. Pojdi skozi gozd, ne ustavljaj se in ne išči bližnjic !"

red biti reči Rdeč pograbiti košarica odhiteti babica biti živeti vasica drug stran gozd
"V redu," je rekla Rdeča kapica, pograbila košarico in odhitela k babici, ki je živela v vasici na drugi strani gozda.

LemmaGen: http://lemmatise.ijs.si

Korenjenje (stemming)

Korenjenje ali stematizacija je tehnika za iskanje besednega korena. Tehnika se pogosto uporablja za indeksiranje besed v spletnih iskalnikih, ki namesto vseh oblik ene in iste besede shranjujejo le korene. Ta tehnika je zelo podobna tehniki lematizacije, le da tu velja, da je korenjenje možno izvesti le s pomočjo določenih pravil, ki se v različnih naravnih jezikih razlikujejo. Korenjenje je zaradi manjše kompleksnosti hitrejša operacija od operacije lematizacije, prav tako tu ne potrebujemo slovarja besed. Stematizacija se razlikuje od lematizacije po tem, da je rezultat lematizacije vedno slovnično pravilno beseda (osnovna oblika besede), medtem ko pri stematizaciji ostane besedni koren, ki ni nujno slovnično pravilna beseda, ampak le njen osnovni del brez končnice. Pri lematizaciji nam pomaga, da poznamo, kakšno oblikoskladenjsko označbo ima beseda, medtem ko pri stematizaciji tega ne potrebujemo. V angleščini se za stematizacijo najpogosteje uporablja Porterjev algoritem.

Korenjenje

Once upon a time there lived in a certain village a little country girl, the prettiest creature who was ever seen. Her mother was excessively fond of her; and her grandmother doted on her still more. This good woman had a little red riding hood made for her. It suited the girl so extremely well that everybody called her Little Red Riding Hood.

One day her mother, having made some cakes, said to her, "Go, my dear, and see how your grandmother is doing, for I hear she has been very ill. Take her a cake, and this little pot of butter."

Little Red Riding Hood set out immediately to go to her grandmother, who lived in another village.

Onc upon a time there live in a certain villag a littl countri girl , the prettiest creatur who wa ever seen . Her mother wa excess fond of her ; and her grandmoth dote on her still more . Thi good woman had a littl red ride hood made for her . It suit the girl so extrem well that everybodi call her Littl Red Ride Hood .

One day her mother , have made some cake , said to her , " Go , my dear , and see how your grandmoth is do , for I hear she ha been veri ill . Take her a cake , and thi littl pot of butter . "

Littl Red Ride Hood set out immedi to go to her grandmoth , who live in anoth villag .

Korenjenje ali lematizacija?

Lematizacija

- Igra, igre, igri → igra
- Igralec, igralci, igralca → igralec
- Igralnica, igralnice → igralnica
- Igram, igramo → igrati

Korenjenje

- Igra, igre, igri, igralec, igralci, igralca, igralnica, igralnice, igram, igramo → "igr"

Blokirane besede (Stopwords)

- *Blokirane besede* (*prazne besede*, *stopwords*) so besede, ki jih pri rudarjenju besedil izločimo, ker niso polnopomenske.
  - Slovenščina: je, in, se, v, da, na, pa, kakor, ne, so, bi, z, še, za, to, po, tako, ni, že, s, ko, tudi, kaj, si, ki, kar, ali, iz, zdaj, bo, od, bilo, bila, bil, vse, kako, če, pri, o, ob, samo, k, več, naj, sem, tam, pred, ta, sta, saj, le, tem, pod, med, ker, do, res, prav, tega, tu, ter, nič, nad, morda, vendar, no, čez, kdaj, bom, ves, a, sam, bili, brez, boš, proti, kjer, okrog, bolj, vsa, kje, torej, toda, sama, šele, vso, teh, tja, celo, mogoče, mu, ga, ji, jo, ti, mi, jih, te, potem, me
  - Angleščina: i, me, my, myself, we, our, ours, ourselves, you, your, yours, yourself, yourselves, he, him, his, himself, she, her, hers, herself, it, its, itself, they, them, their, theirs, themselves, what, which, who, whom, this, that, these, those, am, is, are, was, were, be, been, being, have, has, had, having, do, does, did, doing, a, an, the, and, but, if, or, because, as, until, while, of, at, by, for, with, about, against, between, into, through, during, before, after, above, below, to, from, up, down, in, out, on, off, over, under, again, further, then, once, here, there, when, where, why, how, all, any, both, each, few, more, most, other, some, such, no, nor, not, only, own, same, so, than, too, very, s, t, can, will, just, don, should, now

Odstranjevanje blokiranih besed

Once upon a time there lived in a certain village a little country girl, the prettiest creature who was ever seen. Her mother was excessively fond of her; and her grandmother doted on her still more. This good woman had a little red riding hood made for her. It suited the girl so extremely well that everybody called her Little Red Riding Hood.

One day her mother, having made some cakes, said to her, "Go, my dear, and see how your grandmother is doing, for I hear she has been very ill. Take her a cake, and this little pot of butter."

Little Red Riding Hood set out immediately to go to her grandmother, who lived in another village.

time lived village country girl prettiest creature mother excessively fond grandmother doted good woman red riding hood suited girl extremely called Red Riding Hood

day mother cakes dear grandmother hear ill cake pot butter Red Riding Hood set immediately grandmother lived village

N-grami

- N-grami so zaporedne, med seboj prekrivajoče se sekvence objektov dolžine *N* znotraj daljšega zaporedja objektov.
- V primeru procesiranja naravnega jezika so ti objekti običajno besede (žetoni znotraj tokeniziranega teksta), včasih pa tudi fonemi ali črke.
- N-grame pri obdelavi naravnega jezika uporabljamo kot približek za fraze. Tiste, ki se pojavijo preredko, pa zavržemo.

N-grami

"Rdeča kapica" je več kot "rdeča" "kapica"

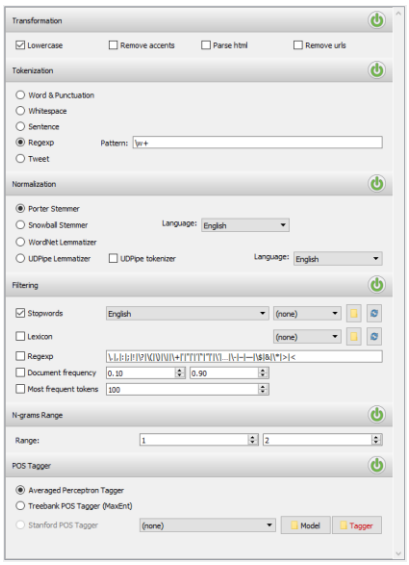
Nekoč je živela ljubka deklica in vsi, ki so jo poznali, so jo imeli radi. Njena babica, ki je imela zanjo še prav poseben prostor v srcu, ji je podarila čudovito rdečo kapico. Tako navdušena je bila deklica nad kapico, da jo je nosila, kamorkoli je šla, zato so jo vsi klicali Rdeča kapica.

Nekega dne jo je mama poklicala k sebi in rekla: "Rdeča kapica, vzemi tole košarico in nesi svoji bolni babici nekaj hrane in pijače. Dobro ji bo delo. Pojdi skozi gozd, ne ustavljaj se in ne išči bližnjic!"

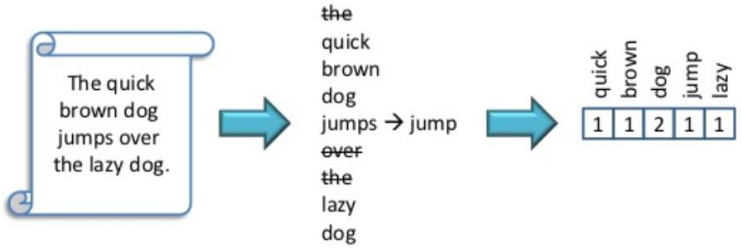
"V redu," je rekla Rdeča kapica, pograbila košarico in odhitela k babici, ki je živela v vasici na drugi strani gozda.

Procesiranje besedil v Orange

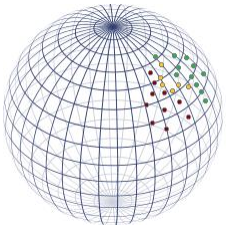
- Transformation
- Tokenization
- Normalization (Stemming & Lemmatisation)
- Filtering (stopwords, frequency)
- N-gram
- POS Tager
  - POS = Part Of Speech



Vektorizacija: vreča besed (bag of words)



- Vsaka beseda (ali bigram) je ena dimenzija
- Vsak dokument je točka v hiperprostoru
- Uporabimo "redko" predstavitev, ker je veliko ničel



Vektorizacija: Vreča besed (Bag of words)

- Dokument lahko predstavimo kot vrečo besed (bag of words)
- Izgubi se vrstni red besed
- Pomenske besede so preštete

A Tale About th...	youth=24.000, yourselv=1.000, younger=3.000, young=2.000, yet=4.000, yell=1.000, ye=3.000, wretch=1.000, would=12.000, world=1.000,
Brier Rose	year=7.000, wretchedli=1.000, wound=3.000, wise=1.000, wing=2.000, window=1.000, wheel=1.000, wear=1.000, wand=1.000, wall=3.000,
Cat and Mouse ...	yearn=2.000, wors=1.000, winter=2.000, wine=1.000, whenev=1.000, wager=1.000, verili=1.000, usual=1.000, untru=1.000, uncommon=1.
Cinderella	worn=1.000, within=1.000, whatev=1.000, wept=1.000, wash=2.000, use=1.000, unknown=2.000, unawar=1.000, twig=2.000, turtl=4.000, t
Hansel and Gretel	yield=1.000, wood=8.000, wither=1.000, witch=8.000, wild=2.000, whosoev=1.000, weep=1.000, week=1.000, weari=2.000, voic=1.000, tos
Herr Korbes	whistl=1.000, towel=2.000, splash=1.000, rooster=6.000, rage=1.000, prick=2.000, pole=1.000, onto=3.000, needl=3.000, millston=3.000, m
Jorinda and Jori...	yard=1.000, willow=1.000, whu=3.000, whose=3.000, wander=1.000, villag=1.000, underwood=1.000, tu=3.000, trembl=1.000, travel=1.00
Little Red Ridin...	yesterday=2.000, wolf=18.000, whither=1.000, weak=2.000, velvet=1.000, twice=1.000, trough=4.000, thrice=1.000, tender=1.000, sweetli=
Mother Holle	widow=1.000, welcom=1.000, violent=1.000, unkindli=1.000, ugli=3.000, thoroughli=1.000, sunshin=1.000, summon=1.000, stepdaught=
Old Sultan	tooth=1.000, thiev=1.000, swore=1.000, sword=1.000, sultan=14.000, stout=1.000, shoot=1.000, shade=1.000, scratch=1.000, rogu=1.000,
Pack of Scound...	whether=1.000, wake=1.000, waddl=1.000, vow=1.000, total=1.000, tavern=1.000, tailor=1.000, suspect=1.000, streak=1.000, start=1.000, s

Katere besede in besedne zveze so pomembne?

- Pogosto nastopa v dokumentu
- Poredko nastopa v ostalih dokumentih korpusa

Rdeča kapica

Nekoč je živela ljubka deklica in vsi, ki so jo poznali, so jo imeli radi. Njena babica, ki je imela zanjo še prav poseben prostor v srcu, ji je podarila čudovito rdečo kapico. Tako navdušena je bila deklica nad kapico, da jo je nosila, kamorkoli je šla, zato so jo vsi klicali Rdeča kapica.

Nekega dne jo je mama poklicala k sebi in rekla: "Rdeča kapica, vzemi tole košarico in nesi svoji bolni babici nekaj hrane in pijače. Dobro ji bo delo. Pojdi skozi gozd, ne ustavljaj se in ne išči bližnjic!"

"V redu," je rekla Rdeča kapica, pograbila košarico in odhitela k babici, ki je živela v vasici na drugi strani gozda.

Uteževanje besed

- Vsako besedo, ki nastopa v dokumentu, obtežimo s pomembnostjo v dokumentu
- Beseda je pomembna (ključna) za dokument, če se v njem pogosto pojavlja, v ostalih dokumentih pa poredko

The Straw, the Coal, and the Bean	ventured=3.784, tripped=3.784, travelling=2.686, thanked=3.091, sheer=3.784, sewed=3.091, seam=3.784, rushing=3.784, repair=3.784, c
The Willow-Wren and the Bear	wrens=15.137, wren=26.489, whirring=3.784, swarming=3.784, string=3.784, sting=7.568, spies=3.784, settle=3.784, rib=3.091, plume=3.7
The Wolf and the Man	tickled=3.784, protect=3.784, nonetheless=3.784, imagined=3.784, hail=3.784, gun=3.784, fired=3.784, employ=3.784, double=3.784, dis
The Juniper Tree	workshop=3.784, wet=3.784, waved=3.784, veins=3.784, urged=3.091, uneasiness=3.784, undone=3.091, underneath=30.274, uncle=3.78
A Tale About the Boy Who Went Forth	youth=64.454, younger=9.273, young=1.386, yet=4.046, yes=2.681, yelled=3.784, wretch=2.398, would=0.847, world=0.788, works=3.784,
Hansel and Gretel	yielded=3.091, woods=2.175, wood=6.257, withered=3.091, witches=3.784, witch=16.785, wild=4.350, whosoever=2.686, weep=1.992, w
Little Red Riding Hood	yesterday=5.371, wolf=33.089, whither=2.686, weak=5.371, velvet=3.784, twice=2.175, trough=15.137, thrice=3.784, thoughts=3.784, ten
Rapunzel	wretchedness=3.784, wrapped=3.091, wound=3.784, wetted=3.784, weave=3.784, wall=9.273, venomous=3.784, unfastened=3.784, twi
Cinderella	ye=12.364, worn=3.091, wondered=2.686, within=1.705, windows=3.091, whatever=1.992, wept=1.587, ways=3.091, watered=3.784, was
Rumpelstiltskin	whistled=6.182, weeping=2.175, wealth=2.686, used=1.705, treasure=2.686, tonight=2.686, tom=3.091, timothy=3.784, throne=3.091, tas
The Blue Light	wreaths=3.784, wounds=3.784, wheresoever=3.784, war=2.686, wages=3.784, valuable=3.784, unseen=3.784, underground=3.784, troub
The Elves and the Shoemaker	workmanship=3.784, worked=3.091, wights=3.784, waistcoat=3.784, twinkling=3.784, troubles=3.784, thriving=3.784, tapping=3.091, stit
The Fisherman and His Wife	yellow=3.784, whirlwind=3.784, week=3.784, waves=18.546, wave=3.091, trumpets=3.784, troop=3.091, trim=3.784, tops=3.784, thunde

Uteževanje besed: tf-idf

- tf-idf: term frequency-inverse document frequency
- Za vsako besedo
  - tf – število pojavitev določene besede znotraj dokumenta
  - idf – factor, ki penalizira besede, ki se pojavljajo v mnogih dokumentih v korpusu
- tf-idf utež besede w v dokumentu = tf(w, dokument) * idf
- Besede z velikim tf-idf so ključne za dokument.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Tf-idf primer

- Little red riding hood

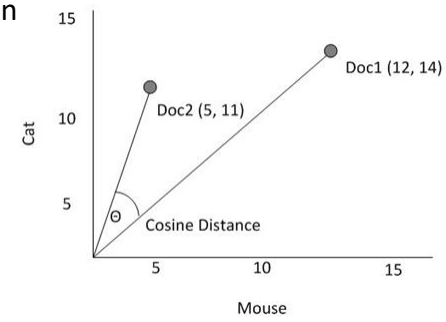
Unigrami in bigrami uteženi s tf-idf v korpusu 44 Grimmovih pravljic.

Unigrams	331
grandmother	98.389
cap	63.068
wolf	33.089
red	30.919
trough	15.137
cake	12.364
flowers	8.699
wine	7.970
snips	7.568
sausages	7.568
league	7.568
latch	7.568
curtains	7.568
bringing	7.568
path	7.353

Bigrams and unigrams	922
grandmother	98.389
red cap	80.367
cap	63.068
little red	56.763
wolf	33.089
red	30.919
trough	15.137
cake	12.364
cap however	11.353
flowers	8.699
good morning	8.057
wine	7.970
two snips	7.568
take grandmother	7.568
straight grandmother	7.568

Podobnost med besedili

- Želimo meriti podobnost besedil po vsebini
- Vsak dokument predstavimo kot točko v prostoru (vektor od izhodišča do točke)
- Prostor ima toliko dimenzij, kolikor je besed in bigramov v korpusu
- Evklidska razdalja slabo deluje (ker je veliko dimenzij)
- Dokumenta sta si podobna, če je kot med njima majhen
- Kot ni občutljiv na dolžino dokumenta



Kosinusna podobnost

- Za učinkovito računanje, računamo kosinus kota (*cos(0)=1*): kosinusna podobnost

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- Dolžina vektorja (dokumenta) ne vpliva na podobnost
- Če vektorje v naprej normaliziramo na dolžino 1, se računanje poenostavi v računanje skalarnega produkta (*dot product*)

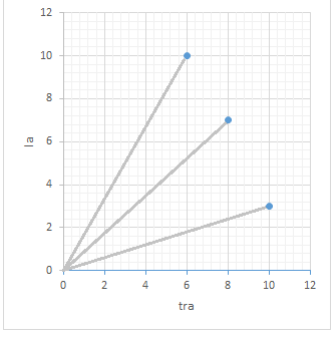
$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

Vaja: kosinusna podobnost

		tra	la
Besedilo 1	tra la la tra la la tra la la tra la la tra la la tra		
Besedilo 2	tra tra tra tra tra tra tra tra tra tra tra la la la		
Besedilo 3	tra la tra la tra la tra la tra la tra la tra la tra la tra		

Vaja: kosinusna podobnost

		tra	la
Besedilo 1	tra la la tra la la tra la la tra la la tra la la tra	6	10
Besedilo 2	tra tra tra tra tra tra tra tra tra tra tra la la la	10	3
Besedilo 3	tra la tra la tra la tra la tra la tra la tra la tra la tra	8	7



- Kosinusna podobnost dokumentov Besedilo 1 in Besedilo 2

$$\frac{(6*10)+(10*3)}{\sqrt{(6^2)+(10^2)}\sqrt{(10^2)+(3^2)}} = \frac{60+30}{\sqrt{36+100}\sqrt{100+9}} = \frac{90}{\sqrt{136}\sqrt{109}} = 0.7391963$$

- Kosinusna podobnost dokumentov Besedilo 1 in Besedilo 3

$$\frac{(6*8)+(10*7)}{\sqrt{(6^2)+(10^2)}\sqrt{(8^2)+(7^2)}} = \frac{48+70}{\sqrt{36+100}\sqrt{64+49}} = \frac{118}{\sqrt{136}\sqrt{113}} = 0.9518606$$

Podobnost med dokumenti uporabljamo za

- Iskanje skupin (clustering)
- Klasificiranju z algoritmi, ki temeljijo na razdalji (npr KNN)

Klasifikacija besedil

- Klasifikacija novic v kategorije (šport, kronika, gospodarstvo, znanost…)
- Zadovoljstvo uporabnikov (analiza sentimenta)
- Detekcija sovražnega govora
- …

Klasifikacija besedil

- Ker so množice dokumentov (korpusi) navadno velike, uporabimo K-means clustering

Iskanje skupin dokumentov

- Uporabimo metode, ki upoštevajo prispevek vseh značilk:
  - Naivni Bayesov klasifikator
  - SVM (linear kernel)
  - KNN
  - Logistična regresija
  - Globoke nevronske mreže
- Metode, ki temeljijo na par značilkah (npr. odločitvena drevesa) so manj primerne

Naloge

- Kateri so koraki predprocesiranja besedil v rudarjenju besedil (po vrsti)?
- Kateri od teh korakov so specifični za jezik?
- Kaj je lematizacija, kaj je korenjenje?
- Kaj so blokiranje besede? Par primerov.
- Navedi primere pogostih bigramov. Kaj nam doprinesejo pri analizi teksta?
- Kako izračunamo pomembne besede za dokument?
- Pomen kratice tf-idf.
- Kako izračunamo podobnost med besedili?
- Na kaj mislimo s tem, da je (nareven) jezik “redundanten”?
- Kaj je vektorizacija?
- Kako predstavimo vektorizirana besedila v računalniku?

Fakulteta za informacijske študije
Faculty of information studies

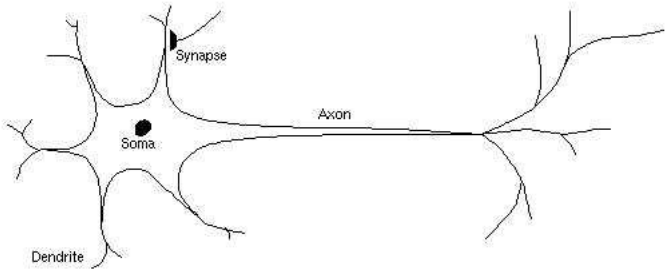
Odkrivanje znanja v podatkih

Deveti sklop

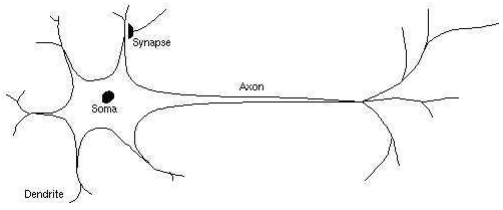
Nevronske mreže

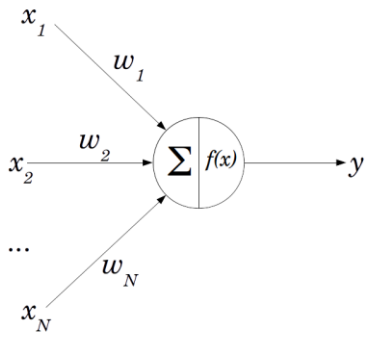
- Umetne nevronske mreže in globoko učenje
- Vložitve ve vektorski prostor (Embedding)

Nevron (naravni)



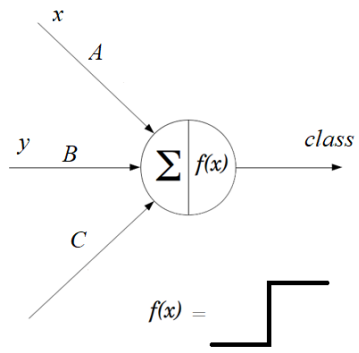
Nevron, perceptron





Perceptron je matematični model biološkega nevrona s pragovno aktivacijsko funkcijo.

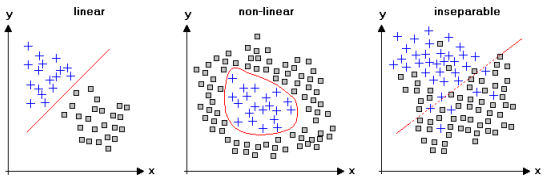
Perceptron je matematični model biološkega nevrona s pragovno aktivacijsko funkcijo



- En sam nevron lahko loči med primeri, ki so linearno separabilni.
- Učimo s spreminjanjem uteži A, B in C.

Output of P = {1 if A x + B y > C

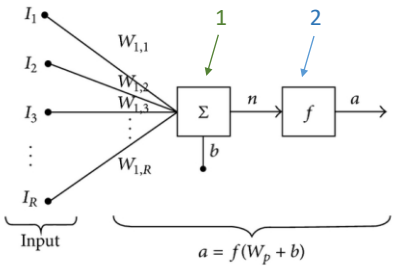
{0 if A x + B y < = C



Slika: http://www.vias.org/tmdatanaleng/cc_data_structure.html

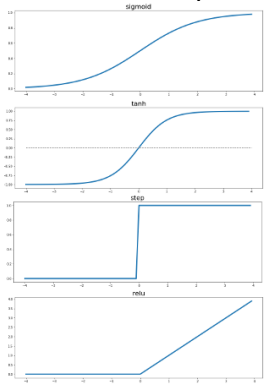
Aktivacijske funkcije

Umetni nevron

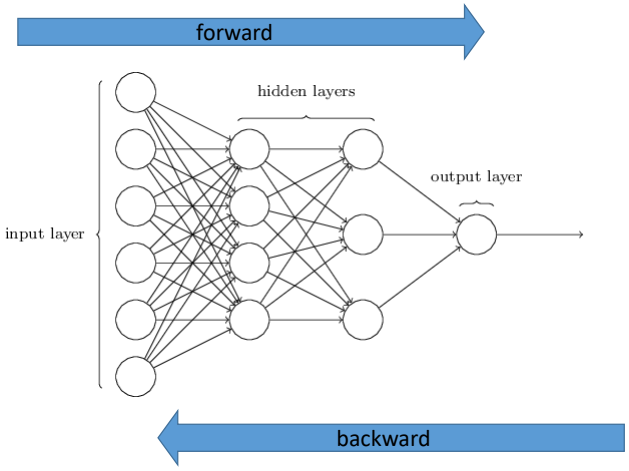


1. Seštevek (zmnožek vhoda s pripadajočo utežjo) + b (bias)
2. Rezultat gre skozi aktivacijsko funkcijo

Aktivacijske funkcije $f(W_p + b)$



Nevronska mreža: več slojev nevronov



- **Vhodni sloj (input layer)**

Sloj nevronov, ki prejema podatke od zunanjih virov in jih predaja naprej v mrežo v obdelavo.

- **Skriti sloj (sloji) (hidden layer)**

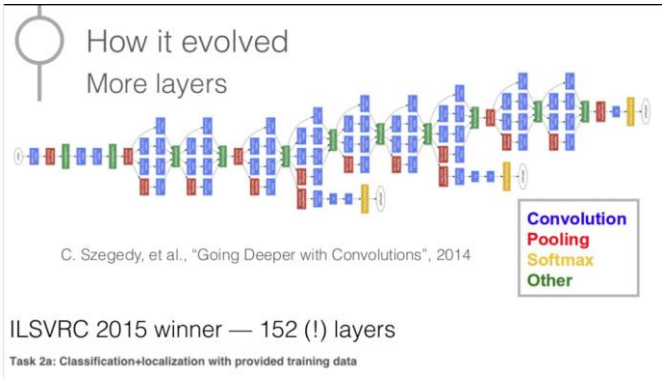
Sloji nevronov, ki prejema podatke od prejšnjega sloja in jih obdela v ozadju. Ta sloj nima direktne povezave z zunanjim svetom. Vse povezave med skritim slojem in ostalimi sloji so skrite v notranjosti sistema.

- **Izhodni sloj (output layer)**

Sloj nevronov, ki prejema obdelane podatke iz zadnjega skritega sloja in oddaja izhodne signale sistema.

Globoko učenje (Deep learning)

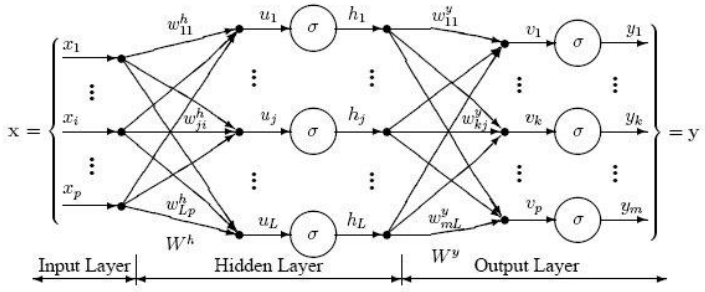
- Globoko učenje (deep learning) obravnava večslojne nevronske mreže.
- Primer: Na tekmovanju "Large Scale Visual Recognition Challenge 2015" je imela nevronska mreža za "image classification, detection, and localization" zmagovalne ekipe **152 slojev**.



https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8

Preslikava vhod-izhod

- Preslikava vhod-izhod mreže je izvedena glede na uteži in aktivacijske funkcije nevronov vseh slojev.
- Vhodni podatek (vektor x na levi) vstopi v mrežo, ustrezne komponente vektorja se pomnoži s pripadajočimi utežmi in vsota gre skozi aktivacijske funkcije pripadajočih nevronov. Ti izhodi se spet množijo z ustreznimi utežmi v skritem sloju …. Dokler ne pride do izhodnega sloja.



Učenje nevronske mreže

- Pri danih vhodih in izhodih (učni primeri), je cilj **prilagoditi uteži** nevronske mreže tako, da se napovedi mreže čimbolj ujemajo s podatki.

Učenje nevronske mreže: Backpropagation

Na začetku so uteži mreže naključne.

1. Večkrat ponovi sprehod čez vse učne podatke (paremater epochs):
    1. Za vsako podmnožico (batch velikosti batch_size):
        1. Forward step: Na podmnožici vhodov (batch) izračunaj izhod mreže in primerjaj z želeno vrednostjo – od tod dobimo napako mreže.
        2. Backward step: določa, za koliko se bo spremenila skupna napaka mreže, če določeno utež spremimo za majhen delta (računamo odvod funkcije napake po posamezni uteži)
        3. Spremeni vrednosti uteži glede na parameter learning_rate

            new weight = old weight — Derivative Rate * learning rate

Train

- **Forward propagation** (check performance)
    - **loss function** is an error metric between actual and predicted
    - absolute error, sum of squares
- **Backpropagation** (direction of parameter/weight change)
    - how much the total error will change if we change the internal weight of the neural network with a certain small value **Δw (gradient)**
    - backpropagate the errors using the derivatives of these functions: auto-differentiation
- **Optimization** (change weights based on learning rate, gradient descent)
    - New weight = old weight — Derivative Rate * learning rate
    - **Batch size** is a hyperparameter that controls the number of training samples to work through before the model's internal weights are updated.
    - The number of **epochs** is a hyperparameter that controls the number of complete passes through the training dataset.

Neural networks and back-propagation explained in a simple way

One-hot Encoding

```
# one-hot encoding class labels

from keras.utils import np_utils

y_train[:10]
```

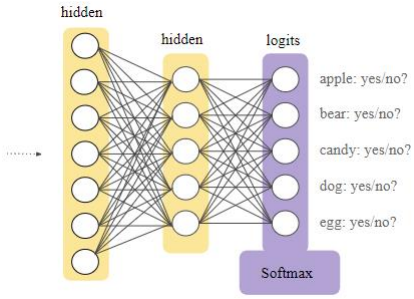
```
array([5, 0, 4, 1, 9, 2, 1, 3, 1, 4], dtype=uint8)
```

```
y_train_OneHotEncoding = np_utils.to_categorical(y_train)
y_train_OneHotEncoding[:10]
```

```
array([[ 0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.],
       [ 1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  1.],
       [ 0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.]])
```

0 1 2 3 4 5 6 7 8 9

Softmax



- Softmax sloj se implementira tik pred izhodnim slojem.
- Softmax mora imeti enako število nevronov kot izhodni sloj.
- Softmax je funkcija, ki skalira izhodne vrednosti tako, da se seštejejo v 1 (diskretna porazdelitev verjetnosti).

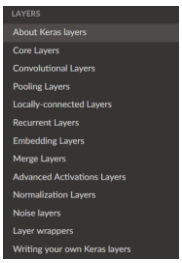
Izhodi splošne nevronske mreže se ne nujno seštejejo v 1.

Arhitektura

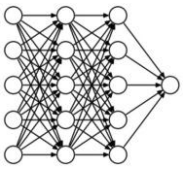
- Sloji: tip , inicializacija, regularizacija
  - Dense (gost)
  - Convolutional (konvolucijski)
  - Pooling
  - Dropout – for regularization
  - Recurrent
  - Embedding
- Aktivacijske funkcije
  - relu
  - softmax (output layer)
- Funkcije napake
  - Klasifikacija
    - **categorical_crossentropy**, categorical_hinge, sparse_categorical_crossentropy, binary_crossentropy, …
  - Numerična predikcija (regresija)
    - **mean_squared_error**, mean_absolute_error, mean_absolute_percentage_error, mean_squared_logarithmic_error, cosine_proximity, …

- **Model.compile**

Tipi slojev (types of layers)

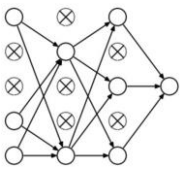


Dense



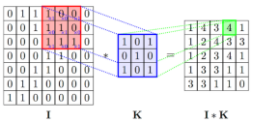
Fully connected.

Dropout



During training, some neurons on a particular layer will be deactivated. This improves generalization because it forces the layer to learn with different neurons the same "concept".

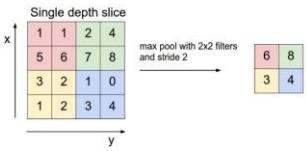
Convolutional



The convolution layer comprises of a set of independent filters. Each filter is independently convolved with the image.

Example: link

Pooling



A max-pooling layer takes the maximum of features over small blocks of a previous layer.

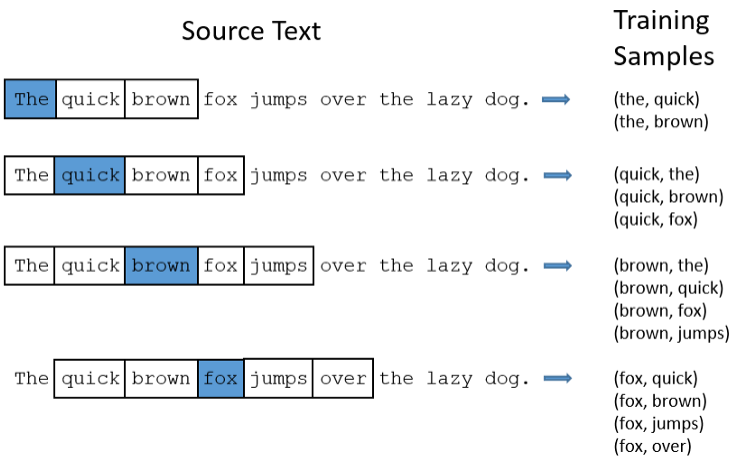
Embeddings
(Vložitve v vektorski prostor)

Na primeru Word2vec

Jezikovni kontekst besed

- Če vemo okolico, lahko napovemo manjkajočo besedo.
  - Danes je lep _____.
  - Včeraj sem _____ na pizzo.
  - Doma imam _____ in mačko.
  - _____ imam psa in mačko.
  - Doma imam psa in _____.

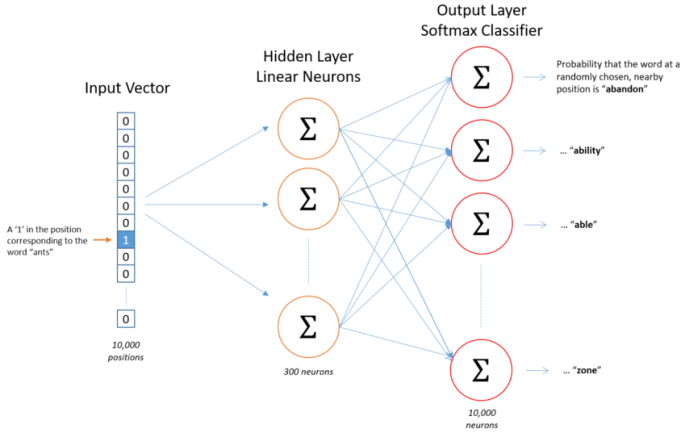
Učna množica: pari (beseda, beseda iz konteksta)



https://becominghuman.ai/how-does-word2vecs-skip-gram-work-f92e0525def4

Nevronska mreža za napovedovanje konteksta

- Besede oštevilčimo.
- Če ima naše besedišče 10.000 besed, je vhodni vektor dolžine 10.000 in ima ničle povsod, razen na mestu, ki ustreza izbrani besedi (one-hot-encoded).
- Izhodni vektor (izhodni sloj nevronske mreže) je tudi velikosti 10.000, kjer so komponente verjetnost, da se beseda nahaja v kontekstu izbrane besede.



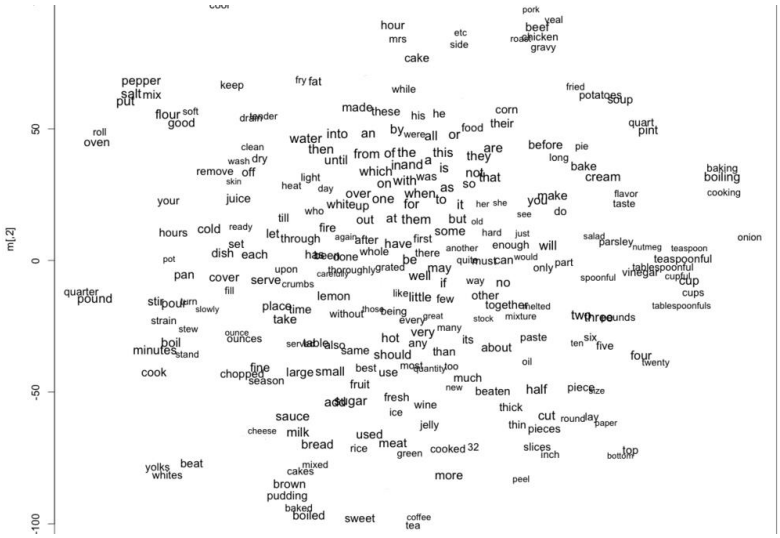
Podrobnejša razlaga: https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Vložitev (embedding)

- Želimo tako preslikavo iz objektov (npr besede, molekule,…) v abstrakten vektorski prostor, da si bojo podobni objekti blizu.
- Matrika uteži za skriti sloj nevronske mreže (na prejšnji prosojnici) nudi ravno tako preslikavo v 300 dimenzijski abstraktni prostor.
- Ta matrika uteži je “look-up” tabela, ki za vsako besedo dodeli preslikavo v 300 dimenzijski prostor (ker je nevronov na skritem nivoju 300).

Word2vec

- Metoda za vložitev besed v vektorski prostor.
- Skupaj so besede, ki so “zamenljive” v smiselnem stavku.
- Del plitve (dvoslojne) nevronske mreže za napovedovanje konteksta besed



Vizualizacija: projekcija vloženih besed v 300 dimenzij na dve dimenziji

Razlaga: https://becominghuman.ai/how-does-word2vecs-skip-gram-work-f92e0525def4

Naloge

- Kaj je perceptron?
- Kaj je nevronska mreža?
- Kako se vhod nevronske mreže preslika v izhod?
- Kaj je globoko učenje?
- Kaj je “backpropagation”.
- Kaj spreminjamo pri učenju nevronske mreže?
- Tipi slojev nevronske mreže.

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Deseti sklop

Laboratorijsko delo

- Uvod v Orange
- Podatki in vizualizacija v Orange
- Klasifikacija in evalvacija
- Jezikovna pristranskost
- Asociacijska pravila
- Razvrščanje v skupine
- Rudarjenje besedil v Orange
- Rudarjenje besedil v Pythonu s knjižnicama NLTK in SciKitLearn
- Nevronske mreže (Python in Keras)

Fakulteta za informacijske študije
Faculty of information studies

Uvod v orange

Petra Kralj Novak

Orodja za podatkovno rudarjenje

- Weka
- Rapid Miner
- Orange
- Online
  - ClowdFlows
- V oblaku (cloud)
  - Amazon Web Services
  - Google Cloud platform
- Models
  - Algorithmia
- ….

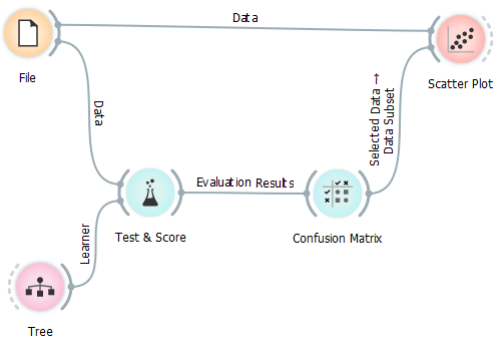
Google Cloud Platform

ALGORITHMIA

Orange

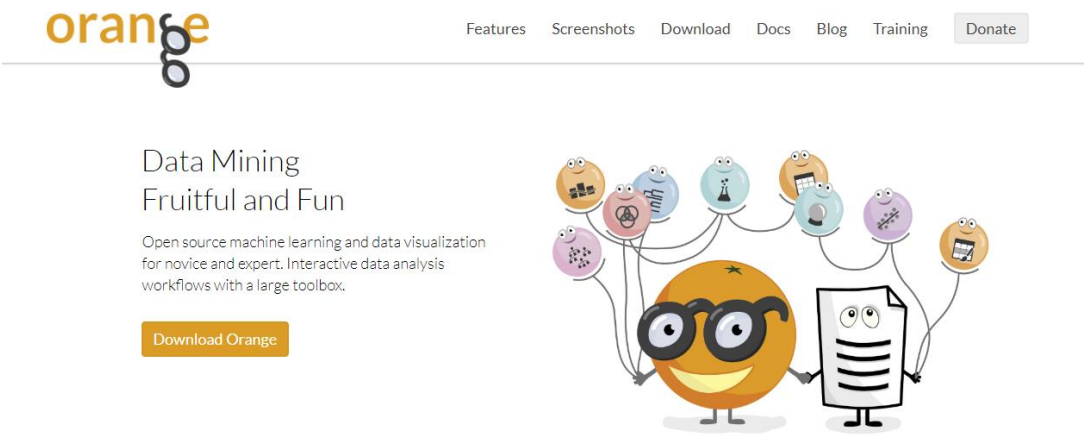
- Okolje za vizualno programiranje za podatkovno rudarjenje
- Porostodostopno in odprtokodno (open source)
- Ni potrebe po programiranu (lahko pa)
  - Vizualno programiranje
  - Interaktivno
- Enostavno
  - Intuitivno
  - Enostano eksperimentitanje
- Razvijajo ga na Fakulteti za računalništvo in informatiko Univerze v Ljubljani (od leta 1996)

Primer vizualnega programa v Orange

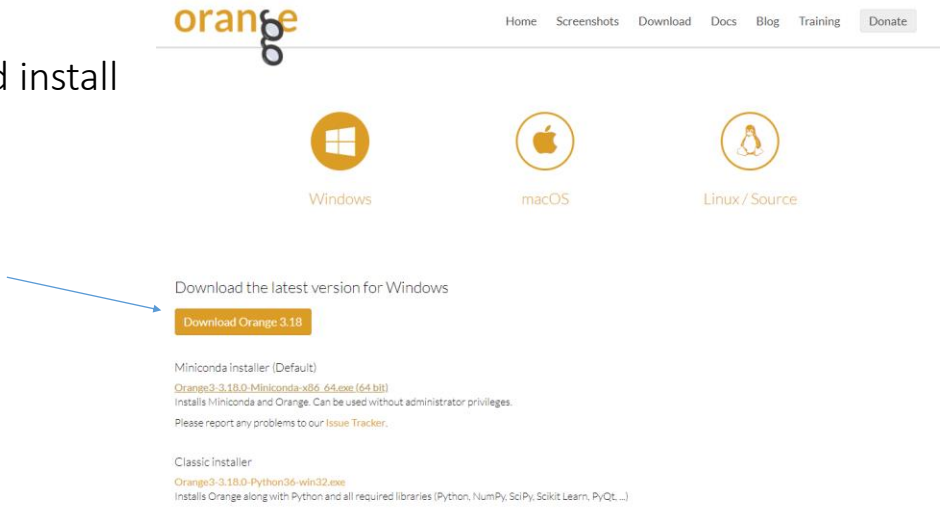


- Widgeti (krogci) so akcije (funkcije, procedure, vizualizacije)
- Po povezavah se prenašajo objekti
  - podatki, izbori podatkov,
  - inicializirani algoritmi za učenje (learner),
  - naučeni modeli (classifier, …)
  - rezultati evalvacije,…
- Widgeti imajo vhode in izhode, preko katerih se povezujejo z ostalimi. Tipi izhodov in vhodov se morajo ujemati.
- Vhodi so na levi, izhodi na desni.

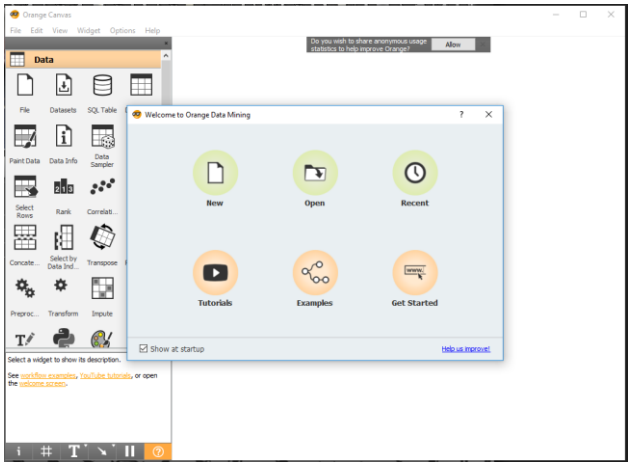
Orange: https://orange.biolab.si/



Orange:
download and install

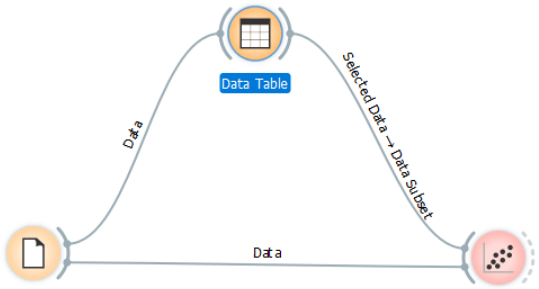


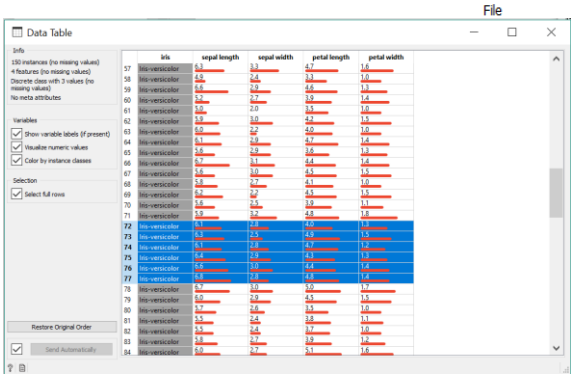
Orange welcome screen



Interaktivnost v Orange

Izbrani podatki v enem widgetu so lahko vhod v drugega





Laboratorijska vaja 1
Podatki in vizualizacija v Orange

1. S pomočjo programom Orange izpolnite tabelo. Podatkovne zbirke iz prvega stolpca najdete v mapi ..\Orange\Lib\site-packages\Orange\datasets. Izpolnite tabelo.

	Število primerov	Število atributov	Število numeričnih atributov	Število kategoričnih atributov	Ciljna spremenljivka	Število ordinalnih atributov
Zoo						
Iris						
Auto-mpg						
Wine						
Titanic						

2. Z uporabo urejevalnika besedil (npr Notepad, Beležnica) preglejte in opišite datoteko tipa ".tab".
3. Izberite podatkovno zbirko, pripravite dve zanimivi vizualizaciji in ju obrazložite.

Laboratorijska vaja 2
Klasifikacija in evalvacija

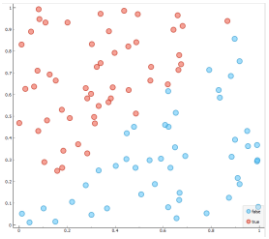
- Primerjaj tri metode evalvacije
  - Množico razdeli na učno (70%) in testno (30%)
  - Prečno preverjanje
  - Naključno vzorčenje
- Testiraj dva modela:
  - Odločitvena drevesa
  - Naivni Bayesov klasifikator
- Uporabi metrike:
  - Klasifikacijska točnost (CA)
  - Povprečno F1 mero
  - Površino pod ROC krivuljo (AUC)
- Uporabi podatkovno množico „car“

Laboratorijska vaja 3: Jezikovna pristranskost (*ang. language bias*)

- V Excelu zgeneriraj 100 primerov:
  - Atribute A, B in C z naključnimi številskimi vrednostmi
  - Ciljno spremenljivko „A>B“, ki ima vrednost „true“, če je A>B, sicer pa “false”
  - Shrani v datoteko
- V programu Orange skušaj napovedati „A>B“ iz atributov A, B in C
  - Nastavi ciljno spremenljivko
  - Uporabi ločeno testno množico
- Kako dober je model?
- Kako vpliva velikost učne množice na kvaliteto modela?
- Nabor MS Excel ukazov:

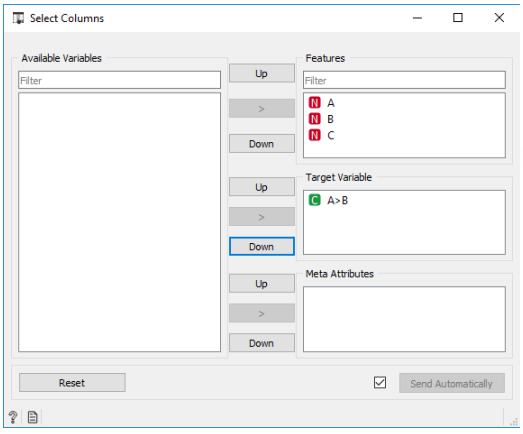
  `= RAND()`

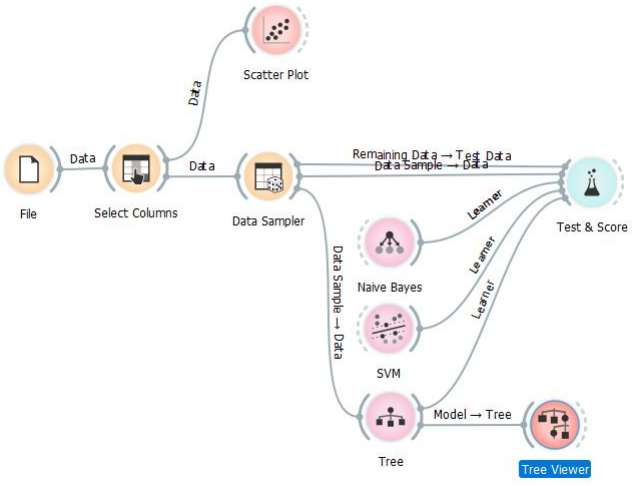
  `= IF(A2>B2,“true",“false")`



A	B	C	A>B
0.90	0.79	0.11	true
0.50	0.20	0.97	false
0.22	0.98	0.74	true
0.74	0.03	0.26	false
0.52	0.73	0.24	true
0.55	0.31	0.24	false
0.18	0.59	0.29	true
0.65	0.53	0.56	false
0.27	0.07	0.75	false
0.01	0.12	0.54	true
0.31	0.90	0.08	true
0.31	0.92	0.68	true

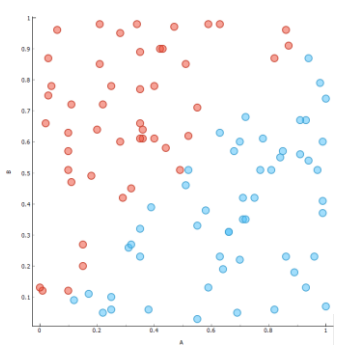
Laboratorijska vaja: jezikovna pristranskost (*ang. language bias*)



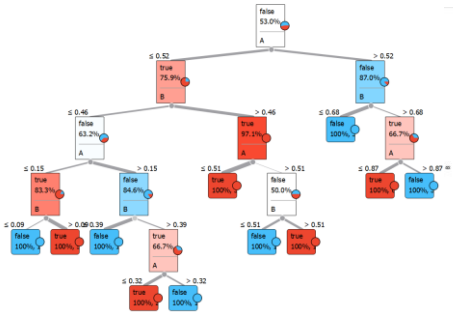


Laboratorijska vaja: jezikovna pristranskost (*ang. language bias*)

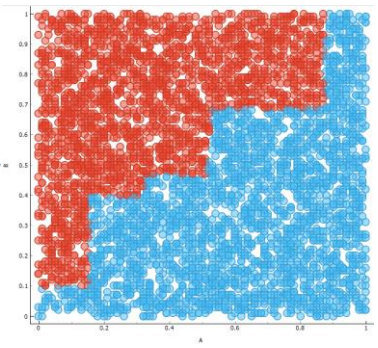
Kako izgleda in kako klasificira odločitveno drevo?



Učna množica

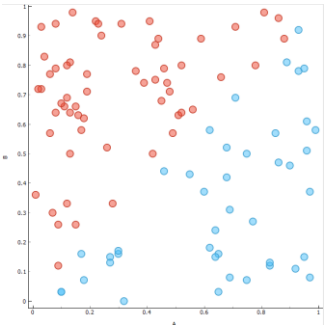


Odločitveno drevo

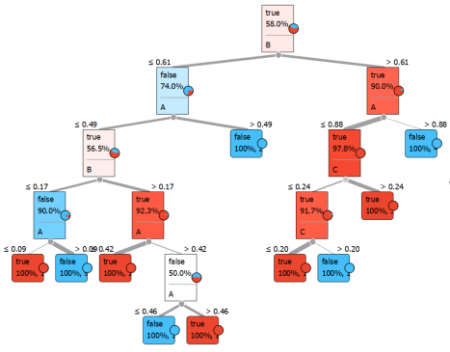


Testna množica

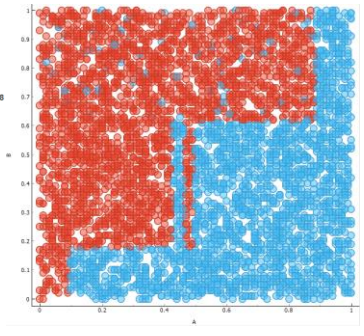
Z drugim (naključnim) izborom učnih in testnih podatkov lahko dobimo bolj ponesrečene modele



Učna množica



Odločitveno drevo



Testna množica

Več možnih rešitev

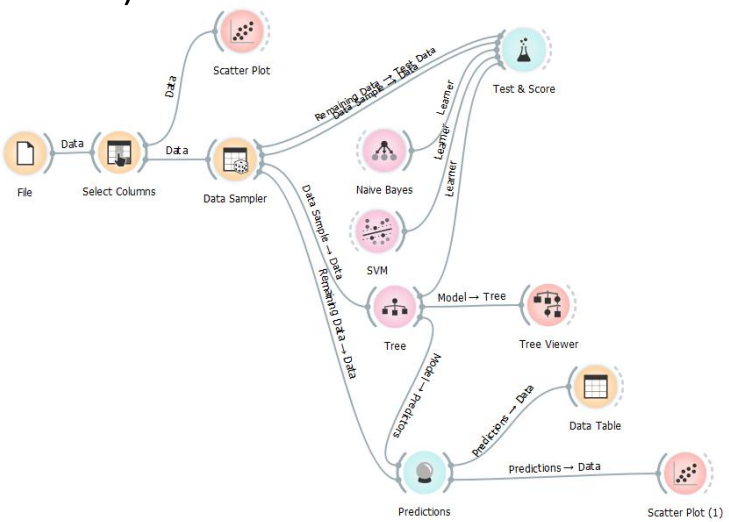
- Predprocesiranje podatkov
  - Iz obstoječih značilk (atributov) naredimo nove, npr A>B ali A+B
  - Praktični primeri:
    - Imamo podatek o telesni višini in telesni masi ljudi
      → Uvedemo nov atribut BMI (body mass index)
    - Imamo podatka o zaslužku in porabi
      → Uvedemo nov atribut dobiček

$$BMI = \frac{Weight\,(kg)}{[Height(m)]^2}$$

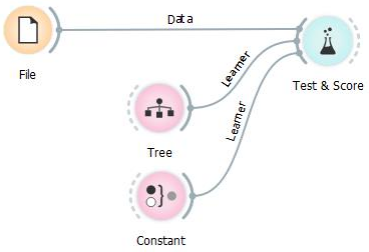
- Ansambel
  - Več različnih modelov, ki glasujejo glede klasifikacije.
  - Ansambel odločitvenih dreves (zgrajenih na podmnožicah primerov in atributov), je naključni gozd (Random Forest).
  - Na tem primeru ima odločitveno drevo klasifikacijsko točnost 88,2%, random forrest pa 90,8%.
  - V splošnem imajo ansambli boljšo klasifikacijsko točnost kot osnovni modeli.

Laboratorijska vaja: jezikovna pristranskost (*ang. language bias*)

- Celoten delotok



Laboratorijska vaja 4
Numerična predikcija



- S pomočjo programa Orange in ročnega kalkulatorja izračunaj RRSE za izbrani model numerične predikcije.
- Podatki: regresija Age-Height (na Moodlu)

- RRSE = root relative squared error
  - Imenovalec: vsota kvadratov napak
  - Števec: vsota kvadratov razlik med dejansko vrednostjo in povprečjem

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

  - RRSE: Razmerje med napako modela in napako povprečja (const modela)
  - Namig: Če števec in imenovalec množimo z 1/n, nastaneta RSE modela in const modela

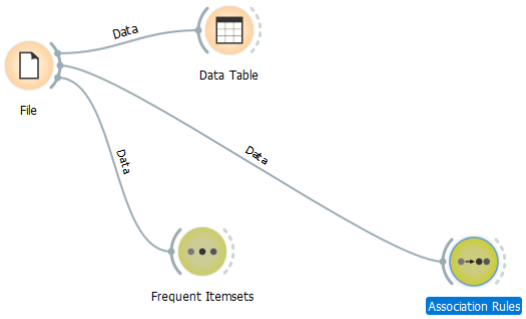
Laboratorijska vaja 5
Asociacijska pravila

1. Primerjaj datoteki „FoodMart.basket“ in „FoodMart.csv“ (link na Moodlu)
2. S pomočjo programa Orange generiraj in poglej pogoste množice in asociacijska pravila za podatke „FootMart“. Kakšna je razlika, če uporabimo datoteko basket ali csv?
3. S pomočjo programa Orange generiraj in poglej pogoste množice in asociacijska pravila za podatke „Voting.tab“
4. Izračunaj „conviction“ za izbrano pravilo

$$\mathrm{conv}(X \Rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)}$$

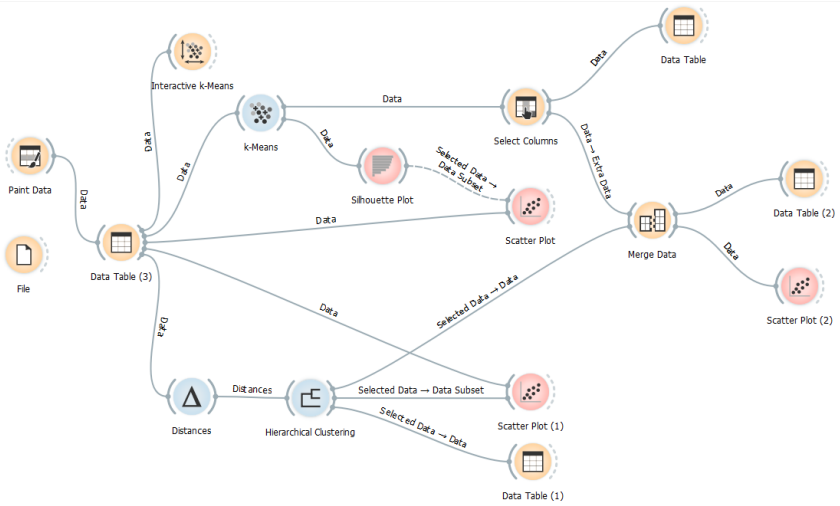
Asociacijska pravila: Delotok v Orange

- Nastavimo majhen minSupport in ga postopoma večamo (za prevelik minSupport bo zmanjkalo delovnega spomina)



Laboratorijska vaja 6
Razvrščanje v skupine

- Primerjava hierarhičnega in k-Means odkrivanja skupin na narisanih podatkih
- Na podatkih „wine.tab“, kjer primerjamo tudi s pravimi razredi



Laboratorijska vaja 7
Rudarjenje besedil v Orange

Orange ni primerno orodje za običajne naloge rudarjenja besedil, so pa v njem lepo ilustrirani osnovni koncepti

- Getting Started with Orange 16: Text Preprocessing
  - https://www.youtube.com/watch?v=V70UwJZWkZ8&t=8s
- Getting Started with Orange 17: Text Clustering
  - https://www.youtube.com/watch?v=rH_vQxQL6oM
- Getting Started with Orange 18: Text Classification
  - https://www.youtube.com/watch?v=zO_zwKZCULo

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Laboratorijska vaja 8:

Rudarjenje besedil v Pythonu s knjižnicama NLTK in SciKitLearn

Python

- Enostaven za branje (kot pseudo-koda) in pisanje
- Interpretiran (skriptni) jezik
- Objektno orientiran
- Dinamično tipiziran
- Veliko knjižnjic (tudi za podatkovno rudarjenje) in podpore (e.g. StackOverflow)
- Sintaksa:
  - Gnezdenje z indentacijo
  - Nizi z " in '

```
print ("Hello world!")
seznam = ["tra", 'la', 'l', 4, 853.6, ["I'm happy!", 'tra-la-la']]
for element in seznam:
    print (element)
```

Naloga: klasifikacija sentimenta

Large Movie Review Dataset: http://ai.stanford.edu/~amaas/data/sentiment/

- 50000 zelo polariziranih ocen filmov
- 25000 v učni, 25000 v testni množici
- Uravnoteženo med razredoma (50% pozitivnih, 50% negativnih)
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

Naložimo učno in testno množico v Pandas DataFrame

```
Train corpus sample:
                                                text     label
0  Bromwell High is a cartoon comedy. It ran at t...  positive
1  Homelessness (or Houselessness as George Carli...  positive
2  Brilliant over-acting by Lesley Ann Warren. Be...  positive
3  This is easily the most underrated film inn th...  positive
4  This is not the typical Mel Brooks film. It wa...  positive

Train corpus shape: (25000, 2)
Test corpus shape: (25000, 2)
```

Inicializacija predprocesiranja

```
import nltk

tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
stopwords = nltk.corpus.stopwords.words("english")
lem = nltk.stem.wordnet.WordNetLemmatizer()


def preprocess(text):
    tokens = tokenizer.tokenize(text)
    lowercased = [word.lower() for word in tokens]
    without_stopwords = [word for word in lowercased if word not in stopwords]
    lemmatized_tokens = [lem.lemmatize(x) for x in without_stopwords]
    return lemmatized_tokens
```

Tf-idf transformacija

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(tokenizer=preprocess, stop_words=None, max_features=5000)

tfidf.fit(train_corpus['text'])
train_data = tfidf.transform(train_corpus['text'])
test_data = tfidf.transform(test_corpus['text'])
```

Funkcija preprocess(text) je definirana v razdelku o predprocesiranju.

Klasifikacija in evalvacija

```
from sklearn import metrics
from sklearn import linear_model
from sklearn import naive_bayes
from sklearn import svm

classifiers = [
    ("Naive Bayes", naive_bayes.MultinomialNB()),
    ("logistic regression", linear_model.LogisticRegression(solver='lbfgs')),
    ("svc", svm.LinearSVC())]

for name, classifier in classifiers:
    classifier.fit(train_data, train_corpus['label'])
    predictions = classifier.predict(test_data)
    print(name, metrics.accuracy_score(predictions, test_corpus['label']))
```

Rezultati (klasifikacijska točnost)

Naši (enostavni) modeli

```
Naive Bayes 0.84216
Logistic regression 0.88164
SVC 0.86564
```

Rezultati iz članka

Features	PL04	Our Dataset	Subjectivity
Bag of Words (bnc)	85.45	87.80	87.77
Bag of Words (bΔt'c)	85.80	88.23	85.65
LDA	66.70	67.42	66.65
LSA	84.55	83.96	82.82
Our Semantic Only	87.10	87.30	86.65
Our Full	84.65	87.44	86.19
Our Full, Additional Unlabeled	87.05	87.99	87.22
Our Semantic + Bag of Words (bnc)	88.30	88.28	88.58
Our Full + Bag of Words (bnc)	87.85	88.33	88.45
Our Full, Add'l Unlabeled + Bag of Words (bnc)	88.90	88.89	88.13
Bag of Words SVM (Pang and Lee, 2004)	87.15	N/A	90.00
Contextual Valence Shifters (Kennedy and Inkpen, 2006)	86.20	N/A	N/A
tf.Δidf Weighting (Martineau and Finin, 2009)	88.10	N/A	N/A
Appraisal Taxonomy (Whitelaw et al., 2005)	90.20	N/A	N/A

Table 2: Classification accuracy on three tasks. From left to right the datasets are: A collection of 2,000 movie reviews often used as a benchmark of sentiment classification (Pang and Lee, 2004), 50,000 reviews we gathered from IMDB, and the sentence subjectivity dataset also released by (Pang and Lee, 2004). All tasks are balanced two-class problems.

Klasifikacija novih primerov

```
new_documents = ["I loved this movie.",
                 "The storyline was boring, while the action was great.",
                 "The end was very unexpected, I didn't let me sleep for a week."]

new_data = tfidf.transform(new_documents)

for name, classifier in classifiers:
    predictions = classifier.predict(new_data)
    print(predictions)
```

```
['positive' 'negative' 'positive']
['positive' 'negative' 'positive']
['positive' 'negative' 'positive']
```

Laboratorijska vaja
Nevronske mreže v Pythonu s knjižnjico Keras

1. Natreniraj **osnovno nevronsko mrežo** na MNIST podatkih.
2. Natreniraj **konvolucijsko nevronsko mrežo** na MNIST podatkih.
3. Natreniraj **globjo konvolucijsko nevronsko mrežo** na MNIST podatkih.

Izvorna koda za vajo je dostopna na: http://source.ijs.si/pkraljnovak/DM_course

MNIST – ročno napisane številke

- Slikice velikosti 28 x 28 pixlov (skupaj 784 pixlov).
- Normalizirane po velikosti in centrirane.
- 60,000 slikic za učenje in 10,000 slikic za testiranje.

Keras: The Python Deep Learning library

- Keras je visokonivojska knjižnica za nevronske mreže za Python. Za računanje lahko uporablja različne nizkonivojske knjižnice: TensorFlow, CNTK ali Theano.
- Googlov Tensorflow je nizkonivojska knjižnica za nevronske mreže, ki jo lahko uporabljamo v programskih jezikih Python in C++. Lahko deluje na grafičnih procesorjih (GPU) ali CPU-jih.

Load the data: 9_neural_nets-0-load_data.py

```
from keras.datasets import mnist
import matplotlib.pyplot as plt

# Plot ad hoc mnist instances

(X_train, y_train), (X_test, y_test) = mnist.load_data()   # Dataset of 60,000 28x28
grayscale images of the 10 digits, along with a test set of 10,000 images.
# plot 4 images as gray scale
plt.subplot(221)
plt.imshow(X_train[0], cmap=plt.get_cmap('gray'))
plt.subplot(222)
plt.imshow(X_train[1], cmap=plt.get_cmap('gray'))
plt.subplot(223)
plt.imshow(X_train[2], cmap=plt.get_cmap('gray'))
plt.subplot(224)
plt.imshow(X_train[3], cmap=plt.get_cmap('gray'))
# show the plot
plt.show()
```

Prepare data: 9_neural_nets-1-perceptron.py

```
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)

# load data
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# flatten 28*28 images to a 784 vector for each image
num_pixels = X_train.shape[1] * X_train.shape[2]
X_train = X_train.reshape(X_train.shape[0], num_pixels).astype('float32')
X_test = X_test.reshape(X_test.shape[0], num_pixels).astype('float32')

# train-validation split
X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train, test_size=0.1, random_state=42)

# normalize inputs from 0-255 to 0-1
X_train = X_train / 255
X_test = X_test / 255

# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_validation = np_utils.to_categorical(y_validation)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]
```

Define + compile, fit, predict: 9_neural_nets-1-perceptron.py

```
# define baseline model
def baseline_model():
    # create model
    model = Sequential()
    model.add(Dense(num_pixels, input_dim=num_pixels, kernel_initializer='normal', activation='relu
    model.add(Dense(num_classes, kernel_initializer='normal', activation='softmax'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model


# build the model
model = baseline_model()
# Fit the model
model.fit(X_train, y_train, validation_data=(X_validation, y_validation), epochs=10, batch_size=200

# Final evaluation of the model
print("Final evaluation of the model")
scores = model.evaluate(X_test, y_test, verbose=1)
print("Baseline Error: %.2f%%" % (100 - scores[1] * 100))
```

Convolutional model

```
def baseline_model():
    # create model
    model = Sequential()
    model.add(Conv2D(32, (5, 5), input_shape=(1, 28, 28), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

Fakulteta za informacijske študije
Faculty of information studies

Odkrivanje znanja v podatkih

Slovar pojmov

Slovar pojmov A-D

A
association rules
povezovalna pravila

attribute-value data
tabelarični podatki

B
bag of words
vreča besed

C
centroid
sredina skupine (umeten primer)

classification accuracy
klasifikacijska točnost

clustering
iskanje skupin, gručenje

confidence
zaupanje

confusion matrix
kontingenčna tabela

cosine similarity
kosinusna podobnost

cross validation
prečno preverjanje

D
data mining
podatkovno rudarjenje

decision tree
odločitveno drevo

deep learning
globoko učenje, učenje globokih nevronskih mrež

deep neural networks
globoke nevronske mreže (nevronske mreže z več skritimi sloji)

dendrogram
drevesni izris hierarhičnega iskanja skupin

distance
razdalja

Slovar pojmov E-O

E
entropy
entropija

F
FN - false negatives
število pozitivnih primerov, ki so napačno klasificirani kot negativni

FP - false positives
število negativnih primerov, ki so napačno klasificirani kot pozitivni

frequent itemset
pogosta množica postavk

H
hierarchical clustering
hierarhično odkrivanje skupin

I
information gain
informacijski pridobitek

item
postavka

itemset
množica postavk

K
KNN - k nearest neighbors
k najbližjih sosedov

M
machine learning
strojno učenje

MAE - mean absolute error
srednja absolutna napaka

medoid
primer v skupini, ki je najbližje centroidu

MSE - mean squared error
srednja kvadratna napaka

N
Naive Bayes classifier
naivni Bayesov klasifikator

neural networks
nevronske mreže

O
outlier
osamelec

Slovar pojmov P-Z

P
precision
natančnoat, preciznost

R
recall
priklic

regression tree
regresijsko drevo

S
separate test set
ločena testna množica

silhouette coefficient
silhuetni koeficient

similarity
podobnost

stemming
korenjenje

stopwords
blokirane besede

supervised learning
nadzorovano učenje

support
podpora

SVM - support vector machines
metoda podpornih vektorjev

T
TDIDT
Top Down Induction of Decision Trees (algoritem za učenje odločitvenih dreves)

text mining
rudarjenje besedil

TN - true negatives
število pravilno klasificiranih negativnih primerov

TP - true positives
število pravilno klasificiranih pozitivnih primerov

transaction data
transakcijski podatki

U
unsupervised leanring
nenadzorovano učenje

Viri

- Aggarwal, C.C. *Data mining: The Textbook*. Springer, 2015.
- Bramer, M. Principles of data mining, 2007.
- Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A. and Štajdohar, M., 2013. Orange: data mining toolbox in Python. The Journal of Machine Learning Research, 14(1), pp.2349-2353.
- Nielsen A. M. Neural networks and deep learning. Vol. 25. San Francisco, CA, USA: Determination press, 2015. Url: http://neuralnetworksanddeeplearning.com/
- Juršic, M., Mozetic, I., Erjavec, T. and Lavrac, N., 2010. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. Journal of Universal Computer Science, 16(9), pp.1190-1214.
- Martinc, M., 2015. Učinkovito procesiranje naravnega jezika s Pythonom (Diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko).

Spletni viri

- Orange – Data Mining Data Mining Fruitful and Fun: https://orange.biolab.si/
- Natural Language Toolkit (NLTK): https://www.nltk.org/
- LemmaGen: http://lemmatise.ijs.si/
- Keras: https://keras.io/
- Brownlee, J. Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras

https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/

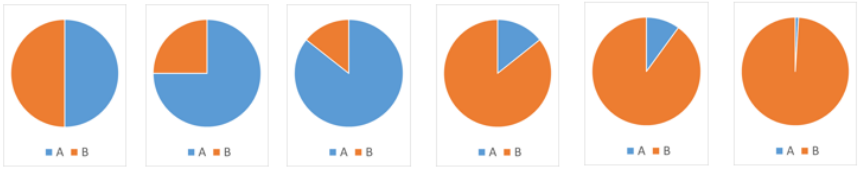
- Chablani M. Word2Vec (skip-gram model): PART 1 – Intuition

https ://towardsdatascience.com/word2vec skip gram model part 1 intuition 78614e4d6e0b

Primeri izpitnih vprašanj za Odkrivanje znanja v podatkih

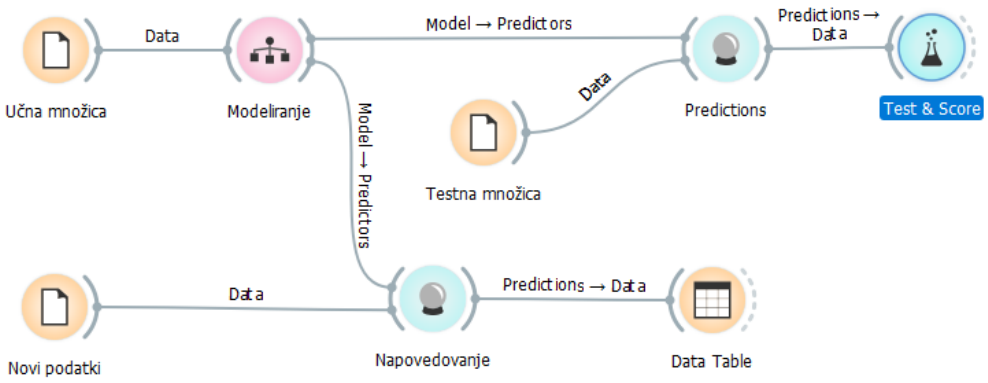
Fakulteta za informacijske študije: Računalništvo in spletne tehnologije
Študijsko leto 2018/2019
Petra Kralj Novak

1. Vrste (tipi) atributov v podatkovnem rudarjenju, primeri.
2. Za dano podatkovno množico, določite za vsak atribut njegov tip. Če je več možnosti, odgovor tudi utemeljite.
3. Naštejte vire napak v podatkih, primeri.
4. Kaj je šum v podatkih.
5. Kako obravnavamo manjkajoče vrednosti v podatkih?
6. V katerem primeru je entropija največja, zakaj?



½ ¼ 1/7 6/7 9/10 99/100

7. Kje v strojnem učenju uporabljamo entropijo?
8. Kako bi dano odločitveno drevo klasificiralo primere? (Drevo, primeri.)
9. Kaj je klasifikacijski problem. Primeri.
10. Opiši algoritem TDIDT (Top Down Induction of Decision Trees).
11. Kaj je informacijski pridobitek?
12. Kateri so ustavitveni kriteriji pri gradnji odločitvenega drevesa v algoritmu ID3 in njegovih nadgradnjah? (Vsaj 3)
13. *Je informacijski pridobitek lahko negativen? Zakaj?
14. Kateri widgeti predstavljajo gradnjo, kateri evalvacijo in kateri aplikacijo modela?



15. Kaj je namen evalvacije?
16. Naštejte tri (3) metode evalvacije in tri (3) metrike.
17. Opišite Testiranje na testni množici.
18. Opišite metodo evalvacije »Naključno vzorčenje (Random sampling)«.
19. Opišite K-kratno prečno (cross validation) preverjanje.
20. Opišite metodo evalvacije »Pusti enega zunaj (Leave one out)«. Kdaj jo uporabljamo?
21. Kaj bi dobili, če bi testirali na učni množici?
22. Kaj je kontingenčna tabela (Confusion matrix)?
23. Iz kontingenčne tabele (Confusion matrix) razberi:
    - Število pravilno klasificiranih primerov
    - Število napačno klasificiranih primerov
    - Klasifikacijsko točnost (classification accuracy)
    - Za vsak razred
      - Priklic (recall)
      - Natančnost (precision)
      - F1

		Predicted		
		no	**yes**	**Σ**
Actual	**no**	1364	126	**1490**
	yes	362	349	**711**
	Σ	**1726**	**475**	**2201**

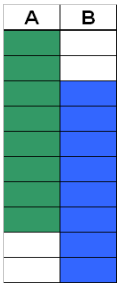
24. Kaj je klasifikacijska točnost, zakaj jo uporabljamo?
25. *Kolikšna je klasifikacijska točnost klasifikatorja, ki vse primere klasificira v večinski razred?
26. Naštejte in kratko opišite primere podatkovnih množic z različnimi cenami napačne klasifikacije.
27. Kdaj je smiselno uporabiti elevacijske metrike za posamezen razred?
28. Kaj je prednost »Mere F1 (F1 measure)« v primerjavi s klasifikacijsko točnostjo (classification accuracy)?
29. Je klasifikacijska točnost 87% dobra? Odgovor utemeljite?
30. *Kolikšen je informacijski pridobitek atributa “ID”?

31. Kaj je ansambel modelov? Kakšne so njegove prednosti in slabosti?
32. Naštejte tri (3) algoritme za klasifikacijo.
33. Kaj je jezikovna pristranskost modela? Kaj je jezikovna pristranskost odločitvenih dreves?
34. Kaj je pretirano prilagajanje učni množici (overfitting)?
35. Kateri problem strojnega učenja naslovimo z Laplaceovo oceno verjetnosti (Laplace estimate)?
36. Opišite razliko med ocenjevanjem verjetnosti z relativno frekvenco (relative frequency) in z Laplaceovo oceno (Laplace estimate).
37. Oceni verjetnosti z relativno frekvenco in z Laplaceovo oceno:

Število dogodkov		**Relativna frekvenca**		**Laplaceova ocena**	
tipa C1	tipa C2	P(C1)	P(C2)	P(C1)	P(C2)
0	2				
12	88				
12	988				
120	880				

38. Primerjaj odločitvena drevesa in Naivni Bayesov klasifikator (interpretabilnost modela, manjkajoče vrednosti)
39. Kaj je numerična predikcija (regresija), primeri.
40. Naštejte algoritme za numerično predikcijo (regresijo).
41. Opišite metodo K najbližjih sosedov (KNN – K nearest neighbors).
42. Naštejte metrike za oceno napake pri numerični predikciji (regresiji).
43. Naštejte metode evalvacije za numerično predikcijo.
44. V katerih enotah merimo MAE (Mean absolute error), MSE (Mean squared error), RMSE (Root mean square error), korelacijski koeficient, klasifikacijsko točnost (classification accuracy), natančnost (precision), priklic (recall), mera F1 (F1 measure).
45. Kako uporabimo algoritem K najbližjih sosedov (KNN – K nearest neighbors) za klasifikacijo in kako za numerično predikcijo (regresijo)?
46. Več je bolje ali manj je bolje? Zaloge vrednosti?
    - MAE, MSE, RMSE, R2 Klasifikacijska točnost, natančnost, priklic, mera F1
47. Opišite asociacijska pravila. Kdaj jih uporabljamo, na kakšnih podatkih, kaj je tipičen primer uporabe?
48. Katere so faze algoritma Apriori?

49. Kaj nam pove podpora (support) asociacijskega pravila? Primer.
50. Kako algoritem Apriori uporabi atribut, ki ima veliko različnih vrednosti (npr mesec rojstva), če ima nastavitev minSupport = 10%? Kakšne rezultate pričakujemo? Kaj se zgodi, če pri pripravi podatkov za Apriori algoritem »pozabimo« izključiti atribut “ID”?
51. Kako algoritem za gradnjo odločitvenih dreves (ID3) uporabi atribut, ki ima veliko različnih vrednosti (npr mesec rojstva)? Kakšne rezultate pričakujemo? Kaj se zgodi, če pri pripravi podatkov za Apriori algoritem »pozabimo« izključiti atribut “ID”?
52. Kako prevedemo tabelarične podatke v transakcijsko obliko? Primer.
53. Imamo atributa A in B, vsak od njiju ima vrednost “1” v 80% primerov. Oba hkrati imata vrednost “1” v najmanj primerih, kar se da (glej sliko). Kakšna bodo asociacijska pravila:
    - minSupport = 50%, min conf = 70%
    - minSupport = 20%, min conf = 70%



54. Kaj nam koristi odkrivanje skupin (3) (gručenje, clustering)?
55. Naštejte tipe razvrščanja v skupine (gručenje, clustering) (2)?
56. Opišite algoritem k-Means. Katera dva koraka se izmenjujeta?
57. Kaj je silhuetni koeficient?
58. Opišite hierarhično odkrivanje skupin (Agglomerative clustering). Kaj je rezultat?
59. Kaj je dendrogram?
60. V katerih primerih je bolj primeren hierarhičen, v katerih pa K-means clustering (razvrščanje v skupine)?
61. Kateri algoritmi temeljijo na razdaljah/podobnosti?
62. Kako določimo primeren K pri K-means algoritmu?
63. Kateri so koraki predprocesiranja besedil v rudarjenju besedil (po vrsti)? Kateri od teh korakov so specifični za jezik?
64. Kaj je lematizacija, kaj je korenjenje? Prednosti in slabosti, tudi glede na jezik.
65. Kaj so blokiranje besede? Par primerov.
66. Navedi primere pogostih bigramov. Kaj nam doprinesejo pri analizi teksta?
67. Kako izračunamo pomembne besede za dokument?
68. Kako izračunamo podobnost med besedili?
69. Na kaj mislimo s tem, da je (nareven) jezik “redundanten”?
70. Kaj je vektorizacija?

71. Kako predstavimo vektorizirana besedila v računalniku?
72. Kaj je perceptron?
73. Kaj je nevronska mreža?
74. Kako se vhod nevronske mreže preslika v izhod?
75. Kaj je globoko učenje?
76. Kaj je “backpropagation”.
77. Kaj spreminjamo pri učenju nevronske mreže?
78. Tipi slojev nevronske mreže.

Računske naloge:

79. Gradnja odločitvenega drevesa.
80. Klasifikacija z Naivnim Bayesovim klasifikatorjem.
81. Poiščite asociacijska pravila z algoritmom Apriori
82. Izpolnite kontingenčne tabele in izračunaj priklic, natančnost, F1 in klasifikacijsko točnost.

Idealen kladifikator

	Da	Ne	
Da			200
Ne			800

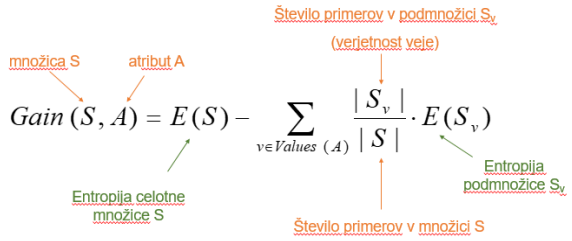
Vse klasificira kot "Da"

	Da	Ne	
Da			200
Ne			800

Vse klasificira kot "Ne"

	Da	Ne	
Da			200
Ne			800

Informacijski pridobitek (information gain)



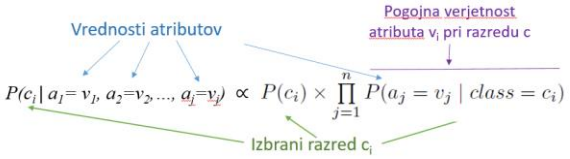
Entropija

$$E(S) = -\sum_{c=1}^{N} p_c \cdot \log_2 p_c$$

Tabela entropij za dva razreda

probability of class 1	probability of class 2	entropy $E(p_1, p_2)$ =
p_1	$p_2 = 1-p_1$	$-p_1*\log_2(p_1) - p_2*\log_2(p_2)$
0	1	0.00
0.05	0.95	0.29
0.10	0.90	0.47
0.15	0.85	0.61
0.20	0.80	0.72
0.25	0.75	0.81
0.30	0.70	0.88
0.35	0.65	0.93
0.40	0.60	0.97
0.45	0.55	0.99
0.50	0.50	1.00
0.55	0.45	0.99
0.60	0.40	0.97
0.65	0.35	0.93
0.70	0.30	0.88
0.75	0.25	0.81
0.80	0.20	0.72
0.85	0.15	0.61
0.90	0.10	0.47
0.95	0.05	0.29
1	0	0.00

Naivni Bayesov klasifikator



		Predicted class		Total instances
		+	−	
Actual class	+	TP	FN	P
	−	FP	TN	N

- Priklic = Recall
- Natančnost = Precision
- Mera F1 = F1 score
- Klasifikacijska točnost = Classification Accuracy

True Positive Rate or Hit Rate or Recall or Sensitivity or TP Rate	TP/P	The proportion of positive instances that are correctly classified as positive
Precision or Positive Predictive Value	TP/(TP+FP)	Proportion of instances classified as positive that are really positive
F1 Score	(2 × Precision × Recall) /(Precision + Recall)	A measure that combines Precision and Recall
Accuracy or Predictive Accuracy	(TP + TN)/(P + N)	The proportion of instances that are correctly classified

Relativna frekvenca: **P(c) = n(c) /N**

Laplaceova ocena: **P(c) = (n(c) + 1) / (N + k)**

n(c) … število dogodkov tipa c
N … število poskusov
k … število različnih tipov dogodkov (npr razredov)

Podpora = Support $Support(X \to Y) = \frac{število(XinY)}{število\_transakcij}$

Zaupanje = Confidence $Confidence(X \to Y) = \frac{število(XinY)}{število(X)}$

Tf-idf:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Kosinusna podobnost:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$