Vhodni podatki:

koncepti / tipi problemov,
atributi, primeri

Pregled vsebine

- Terminologija
- Kaj je koncept / tip problema?
  - klasifikacija, asociacije, razvrščanje v skupine, napovedovanje numeričnih vrednosti
- Kaj je primer?
  - relacije, datoteke, rekurzivni zapis
- Kaj je atribut?
  - nominalni, urejeni, intervalni, razmernostni
- Priprava vhodnih podatkov
  - ARFF, atributi, manjkajoče vrednosti, spoznajmo podatke

Terminologija

- Vhodni podatki glede na:
  - Tip problema: česa vse se lahko naučimo?
    - Cilj: razumljiv, uporaben opis koncepta
  - Primere: posamezni, neodvisni primeri koncepta
    - Pozor: možne so "zapletenejše" oblike vhodnih podatkov
  - Atribute: merijo lastnosti posameznih primerov
    - Osredotočili se bomo predvsem na nominalne in numerične atribute

Kaj je to koncept?

- **Naloge podatkovnega rudarjenja** (stili učenja):
  - Klasifikacija:
    napovedovanje nominalnega razreda
  - Asociacije:
    odkrivanje povezav med značilkami/atributi
  - Razvrščanje v skupine:
    združevanje podobnih primerov v skupine
  - Numerično napovedovanje / regresija:
    napovedovanje numeričnih vrednosti (razreda)
- **Koncept**: zadeva, ki se je želimo naučiti
- **Opis koncepta**: izhod algoritma

Klasifikacija

- Primeri problemov:
  ugotavljanje prebegov, uporaba DNK podatkov pri diagnozi, vremenski podatki za napovedovanje igranja tenisa
- Klasifikacija je **nadzorovano učenje**
  - Za podatke, iz katerih se učimo, poznamo dejanski izid (vrednost razreda)
- Izid imenujemo **razred** primera
- Uspeh učenja lahko merimo na "svežih" podatkih, za katere prav tako poznamo izid (**testni podatki**)
- V praksi se uspeh učenja pogosto meri subjektivno

Asociacije

- Primeri problemov:
  analiza nakupovalnih navad – kateri izdelki se pogosto kupujejo skupaj? (npr.: mleko+kosmiči, čips+omaka)
- Lahko uporabimo tudi brez poznavanja razreda; kakršnakoli odvisnost je potencialno "zanimiva"
- Razlika v primerjavi s klasifikacijo:
  - Napovedujemo lahko vrednost kateregakoli atributa (ne le razreda) ter več atributov hkrati
  - Torej: #asociacijska pravila >> #klasifikacijska pravila
  - Zato: potrebne so dodatne omejitve
    - Minimalna pokritost, minimalna točnost, rangiranje …

Razvrščanje v skupine

- Primeri problemov: profiliranje kupcev
- Naloga:
  najti skupine primerov, ki so si med seboj podobni
- Razvrščanje v skupine je **nenadzorovano**
  - Razred primerov ni poznan
- Uspeh učenja se pogosto meri subjektivno

	Sepal length	Sepal width	Petal length	Petal width	Type
1	5.1	3.5	1.4	0.2	Iris setosa
2	4.9	3.0	1.4	0.2	Iris setosa
…					
51	7.0	3.2	4.7	1.4	Iris versicolor
52	6.4	3.2	4.5	1.5	Iris versicolor
…					
101	6.3	3.3	6.0	2.5	Iris virginica
102	5.8	2.7	5.1	1.9	Iris virginica
…					

Numerično napovedovanje

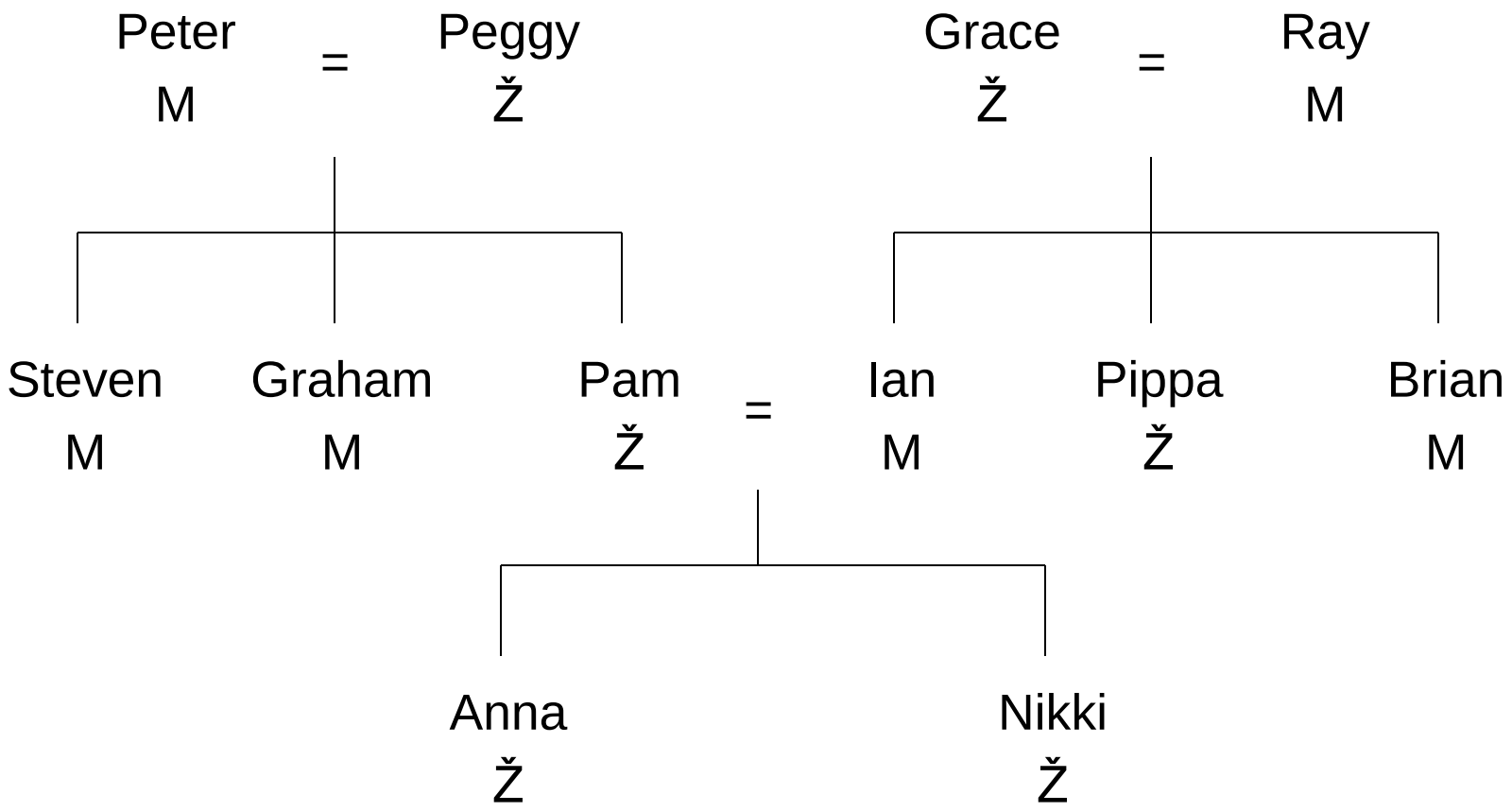
- Pogosto imenovano tudi **regresija**
- Gre za vrsto klasifikacije, le da je **razred numeričen**
- Učenje je **nadzorovano**
  - Za podatke, iz katerih se učimo, poznamo vrednost razreda
- Uspeh učenja merimo na testnih podatkih

Outlook	Temperature	Humidity	Windy	Play-time
Sunny	Hot	High	False	5
Sunny	Hot	High	True	0
Overcast	Hot	High	False	55
Rainy	Mild	Normal	False	40
…	…	…	…	…

Kaj je primer / instanca?

- **Instanca**: točno določen tip primera
  - Stvar, ki jo želimo klasificirati, asociirati, razvrstiti v skupino
  - Posamezen, neodvisen primer ciljnega koncepta
  - Opisana z določeno množico atributov
- Vhod v shemo učenja:
  množica instanc / podatkovna množica (dataset)
  - Predstavljena na enostaven način (ena tabela, tekst …)
- Omejene oblike vhodnih podatkov
  - Brez odvisnosti med posameznimi objekti
  - V praksi najpogostejša oblika pri podatkovnem rudarjenju

Družinsko drevo



Družinsko drevo kot tabela

Ime	Spol	Starš_1	Starš_2
Peter	Moški	?	?
Peggy	Ženska	?	?
Steven	Moški	Peter	Peggy
Graham	Moški	Peter	Peggy
Pam	Ženska	Peter	Peggy
Ian	Moški	Grace	Ray
Pippa	Ženska	Grace	Ray
Brian	Moški	Grace	Ray
Anna	Ženska	Pam	Ian
Nikki	Ženska	Pam	Ian

Relacija "sestra"

prva oseba	druga oseba	sestra?
Peter	Peggy	Ne
Peter	Steven	Ne
…	…	…
Steven	Peter	Ne
Steven	Graham	Ne
Steven	Pam	Da
…	…	…
Ian	Pippa	Da
…	…	…
Anna	Nikki	Da
…	…	…
Nikki	Anna	Da

prva oseba	druga oseba	sestra?
Steven	Pam	Da
Graham	Pam	Da
Ian	Pippa	Da
Brian	Pippa	Da
Anna	Nikki	Da
Nikki	Anna	Da
Vsi ostali primeri		**Ne**

*predpostavka "**zaprtega sveta**"*

… vse v eni tabeli

prva oseba				druga oseba				sestra?
Ime	**Spol**	**Starš_1**	**Starš_2**	**Ime**	**Spol**	**Starš_1**	**Starš_2**	
Steven	Moški	Peter	Peggy	Pam	Ženska	Peter	Peggy	Da
Graham	Moški	Peter	Peggy	Pam	Ženska	Peter	Peggy	Da
Ian	Moški	Grace	Ray	Pippa	Ženska	Grace	Ray	Da
Brian	Moški	Grace	Ray	Pippa	Ženska	Grace	Ray	Da
Anna	Ženska	Pam	Ian	Nikki	Ženska	Pam	Ian	Da
Nikki	Ženska	Pam	Ian	Anna	Ženska	Pam	Ian	Da
Vsi ostali primeri								**Ne**

```
ČE spol druge osebe = Ženska
   IN starš prve osebe = starš druge osebe
   POTEM sestra? = Da
```

Generiranje preproste datoteke

- Proces generiranja (ene) preproste datoteke imenujemo "**denormalizacija**"
  - Različne relacije združimo skupaj v eno samo (join)
  - Je možno za vsako končno množico končnih relacij
- Težava:
  relacije, ki nimajo vnaprej določenega števila objektov
  - Primer: koncept *družine*
- Denormalizacija lahko ustvari "lažne" odvisnosti, ki zgolj odražajo strukturo podatkovne baze
  - Primer: "naziv dobavitelja" točno določa "naslov dobavitelja"

Kaj je atribut?

- Vsak primer je opisan z vnaprej določenim številom značilk – to so "**atributi**" primera
- Toda: v praksi lahko število atributov variira
  - Možna rešitev: posebna oznaka "nepomembno"
- Soroden problem: obstoj nekega atributa je odvisen od vrednosti nekega drugega atributa
- Možni **tipi** atributov (različne "stopnje meritev"):
  - ***nominalni,***
  - ***ordinalni / urejeni,***
  - ***intervalni*** in
  - ***razmernostni***

Nominalne vrednosti

- Vrednosti so različne oznake
  - Same vrednosti služijo kot labele / imena
  - *Nominalen* prihaja iz latinskega izraza za *ime*
- Primer:
  atribut "outlook" iz podatkov o vremenu (weather.arff)
  - vrednosti: "sunny", "overcast" in "rainy"
- Med vrednostmi nominalnega atributa ni nikakršne povezave (urejenosti vrednosti, razdalje …)
- Možne le primerjave "je enako", "ni enako" (=, ≠)

Urejene vrednosti

- Podobno kot nominalen atribut, le da je vrednosti možno "urediti"
- Toda: ni možno definirati razdalje med vrednostmi
- Primer:
  atribut "temperature" iz podatkov o vremenu
  - vrednosti: "hot" > "mild" > "cool"
- Pozor: seštevanje in odštevanje vrednosti nima pomena
- Primer pravila:

  `temperature < hot → play = yes`
- Razlikovanje med *nominalnimi* in *ordinalnimi* atributi ni vedno enostavno (npr.: atribut "outlook")

Intervalne vrednosti (numerične)

- Intervalne vrednosti niso le urejene, ampak jih izražamo v fiksnih in enakih enotah
- 1. primer: atribut "temperature" v Fahrenheit-ih
- 2. primer: atribut "year" (leto)
- ***Razlika*** dveh vrednosti *ima smisel*
- ***Seštevek*** ali ***produkt*** vrednosti *nima pomena*
  - Ničla ni definirana!

Razmernostne vrednosti (numerične)

- Pri razmernostnih vrednostih je definirana tudi **ničla**
- Primer: atribut "distance" (razdalja)
  - Razdalja med nekim objektom in tem istim objektom je nič
- Z razmernostnimi vrednostmi ravnamo kot z realnimi števili
  - Vse aritmetične operacije so "dovoljene"
- Toda: ali je ničla definirana "sama po sebi"?
  - Odgovor je odvisen od znanstvenih dejstev
    (npr.: Fahrenheit ni poznal spodnje meje za temperaturo)

Tipi atributov v praksi

- Velika večina algoritmov strojnega učenja podpira le dva tipa atributov: ***nominalne*** in ***numerične***
- Za nominalne atribute zasledimo tudi izraze "kategorni", "preštevni" ali "diskretni"
  - Toda: "preštevni" in "diskretni" namigujeta na urejenost
- **Poseben primer**: le dve vrednosti ("bool-ovi" ali binarni atributi)
- Numeričnim atributom včasih pravimo tudi "zvezni"
  - Toda: "zvezni" namiguje na matematično kontinuiranost

Tipi atributov: povzetek

- **Nominalni** (npr.: barva = rdeča, morda ...)
  - Samo testi enakosti (=, ≠)
  - Pomemben poseben primer: **binarni** (True / False)
- **Urejeni** (npr.: ocena=A, B, C, D, E, F)
- **Numerični** (**zvezni**) (npr.: leto)
  - Intervalne vrednosti – sorodne celim številom
  - Razmernostne vrednosti – sorodne realnim številom

Zakaj tipi atributov?

- ***Vprašanje: zakaj morajo algoritmi strojnega učenja poznati tip atributov?***
- Odgovor: da bi lahko naredili prave primerjave med vrednostmi ter se naučili pravilnih konceptov.

  Primeri:
  - **Outlook > "sunny"** nima pomena, dočim
  - **Temperature > "cool"** ali
  - **Humidity > 70** imata pomen
- Uporabnost poznavanja tipa atributov tudi za: preverjanje veljavnih in manjkajočih vrednosti ...

Pretvorba: urejeno → binarno

- Preprosta pretvorba:
  ordinalni atribut z *n* vrednostmi lahko "zakodiramo" z *n–1* binarnimi atributi
- Primer: atribut "temperature"

Izvorni podatki

Temperature
Cold
Medium
Hot

⇨

Pretvorjeni podatki

Temperature > cold	Temperature > medium
False	False
True	False
True	True

- Bolje kot da bi ga zgolj "zakodirali" direktno kot nominalni atribut

Metapodatki

- Informacije o podatkih, ki vključujejo predznanje o problemu / konceptu
- Lahko jih uporabimo za omejevanje preiskovalnega prostora
- Primeri:
  - Upoštevanje dimenzije podatkov
    (npr.: izrazi / rezultati morajo biti v prave dimenzije)
  - Krožne ureditve
    (npr.: stopinje v krogu – 0° - 360°)
  - Delne ureditve
    (npr.: relacije splošno/specifično)

Priprava vhodnih podatkov

- Težava: različni podatkovni viri
  (npr.: oddelek za prodajo, finančni oddelek …)
  - Razlike:
    stili hranjenja zapisov, dogovori, intervali dostopa, agregacija, primarni ključi, napake …
  - Podatki morajo biti zbrani, povezani, prečiščeni …
  - Podatkovno skladišče: konsistentnost dostopanja do podatkov
- Denormalizacija ni edina težava
- Včasih potrebujemo "zunanje podatke"
  (t.i. "overlay data")
- Pomembno: tip in nivo agregacije podatkov

ARFF format

```
%
% ARFF datoteka o vremenu z nekaterimi numeričnimi atributi
%
```

komentarji

```
@relation weather
```

naziv podatkov

```
@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {true, false}
@attribute play? {yes, no}
```

opis atributov
(imena, tipi)

```
@data
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
...
```

podatki
(v CSV formatu)

Tipi atributov v Weki

- ARFF podpira (le) numerične in nominalne atribute
- Interpretacija je odvisna od (učnega) algoritma;
  - Numerične atribute lahko interpretiramo kot:
    - Ordinalne, če so uporabljene primerjave < ali >;
    - Razmernostne, če se računajo razdalje (lahko je potrebna normalizacija/standardizacija)
  - Algoritmi, ki delujejo na podlagi primerov (instance-based ML algorithms) interpretirajo razdaljo med nominalnimi vrednostmi (0 – enake vrednosti, 1 – različne vrednosti)
- Cela števila: nominalno, urejeno ali razmernostno?

Nominalno vs. urejeno

- "age" kot ***nominalni*** atribut:

```
ČE age = young IN astigmatic = no
    IN tear production rate = normal
    POTEM recommendation = soft

ČE age = pre-presbyopic IN astigmatic = no
    IN tear production rate = normal
    POTEM recommendation = soft
```

- "age" kot ***urejeni*** atribut:

(npr.: "young" < "pre-presbyopic" < "presbyopic")

```
ČE age ≤ pre-presbyopic IN astigmatic = no
    IN tear production rate = normal
    POTEM recommendation = soft
```

Manjkajoče vrednosti

- Pogosto označene tudi kot vnosi "izven obsega"
  - Tipi manjkajočih vrednosti:
    neznana, nezapisana, nerelevantna
  - Vzroki za manjkajoče vrednosti:
    - nepravilno delujoča oprema
    - sprememba v zasnovi / izvedbi eksperimenta
    - združevanje različnih podatkovnih zbirk
    - nezmožnost izvedbe meritve …
- Sam pojav manjkajoče vrednosti je lahko pomembna informacija (npr.: manjkajoča meritev pri zdravniškem pregledu)
  - Večina algoritmov tega ne predvideva
    ➔ "manjkajoče" je včasih potrebno zapisati kot ločeno vrednost

Manjkajoče vrednosti – primer

- Vrednost je lahko manjkajoča, ker ni bila zapisana ali je ni bilo možno zapisati
- V medicinskih podatkih iz tabele je vrednost atributa **noseča?** pri **Jane** dejansko manjkajoča, medtem ko je pri **Joe**-ju ali **Anna**-i ni možno zapisati
- Nekateri programi znajo manjkajoče vrednosti avtomatsko določiti

Baza sprejemov v bolnišnico

ime	starost	spol	noseča?	...
Mary	25	F	N	...
Jane	27	F	**?**	...
Joe	30	M	**?**	...
Anna	2	F	**?**	...
...	...	...	...	...

Nenatančne vrednosti

- Vzrok: podatki niso bili zbrani za potrebe podatkovnega rudarjenja
- Rezultat: napake in izpuščeni zapisi, ki pa ne vplivajo na namen zbranih podatkov (npr.: starost kupca)
- Tiskarske napake v vrednostih nominalnih atributov $\Rightarrow$ dosledno je potrebno pregledati vrednosti
- Tiskarske napake in napake v meritvah pri numeričnih atributih $\Rightarrow$ identificirati je potrebno osamelce (outliers)
- Napake so lahko namenske (npr.: napačne poštne številke)
- Druge težave: podvojeni podatki, "zastareli" podatki …

"Iluzija" natančnosti

- Primer: izraženost nekega gena je lahko zapisana kot $\mathbf{x}_{83} = \mathtt{193.3742}$ pri napaki meritve `+/- 20`.
- Dejanska vrednost je nekje znotraj intervala `[173, 213]`, torej bi lahko izraženost gena zaokrožili na vrednost `190`.
- *Ne predpostavljajmo, da je vsaka decimalka pomembna !!!*

Spoznajmo podatke

- Preproste metode vizualizacije podatkov so včasih lahko zelo koristne
  - Nominalni atributi: ***histogrami***
    (ali porazdelitev vrednosti odraža naše predznanje?)
  - Numerični atributi: ***grafi***
    (opazimo kake očitne osamelce?)
- 2D in 3D prikazi za ugotavljanje odvisnosti
- Vprašajmo za nasvet problemskega strokovnjaka
- "Preveč" podatkov za učinkovito analizo?
  Vzemimo vzorec !

Povzetek

- **Koncept**: zadeva, ki se je želimo naučiti
- **Primer**: posamezen primer koncepta
- **Atribut**: meri značilnosti primerov