



## Poglavje 5

# Odkrivanje zakonitosti v podatkovnih bazah (Data Mining)

## Kaj je OZP?

- Odkrivanje zakonitosti v podatkih je proces identifikacije veljavnih, novih, potencialno uporabnih in razumljivih vzorcev v podatkih, kar lahko imenujemo tudi znanje.
- Poleg iskanja vzorcev gre lahko tudi za iskanje asociacij, sprememb in anomalij

## Drugi izrazi za OZP

- Podatkovno rudarjenje
- Odkrivanje znanja iz podatkovnih baz
- Iskanje zakonitosti v podatkih
- Induktivno strojno učenje (metodologija)
- Avtomatsko modeliranje (metodologija)

## Zakaj odkrivati zakonitosti v podatkovnih bazah?

---



- Podatkovne baze nas obdajajo vsepovsod
- Velikosti podatkovnih baz (še posebej pa podatkovnih skladišč) gredo v tera/peta/exabyte, npr. poslovni podatki, podatki iz opazovanja vesolja, človeški genom, webscale aplikacije, strojni vid, ...
- Zakaj sploh zbiramo tolikšne količine podatkov?
  - Prepričani smo, da se v podatkih skrivajo strateško pomembne informacije.
  - Človek zelo dobro odkriva enostavne (očitne) zakonitosti
  - Izkaže se, da so bolj zapletene zakonitosti dobro skrite!!!
- Fayyad (KDD, 1996): *"Zdaj, ko smo zbrali tolikšno količino podakov, kaj naj počnemo z njimi?"*

## Odkrivanje zakonitosti v podatkovnih bazah (*knowledge discovery in databases – KDD*)

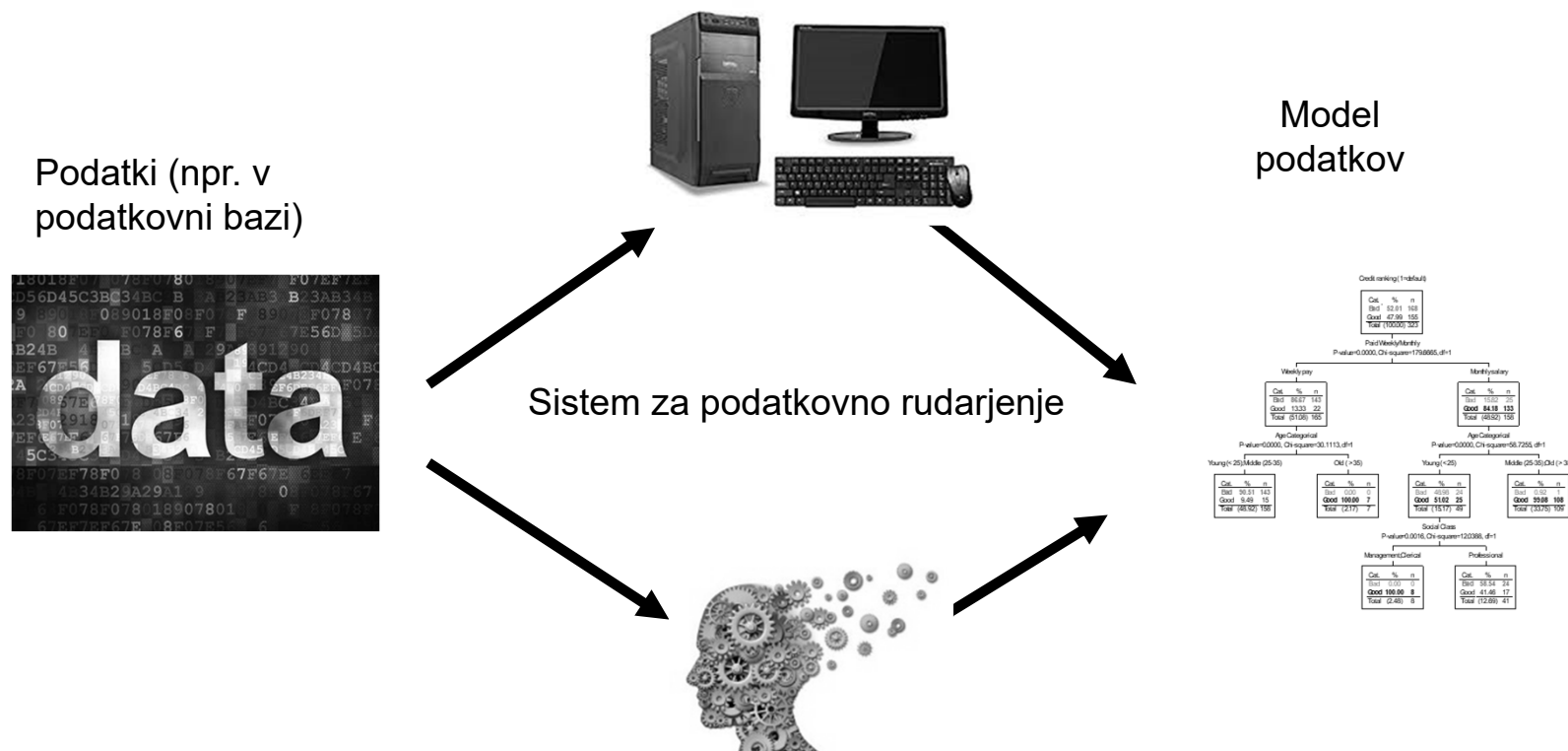
---



- V podatkovni bazi želimo odkriti neko novo, koristno znanje (ali splošneje: zakonitost)
- Odkrivanje zakonitosti v podatkovnih bazah kot *standardiziran* proces (podatkovno rudarjenje oz. *data mining*)

# Podatkovno rudarjenje (*data mining*)

- Popularno ime (*buzzword*) za metode in tehnike odkrivanja zakonitosti v podatkih
- Podatkovno rudarjenje kot modeliranje (določenega vidika) podatkov
- Model nam predstavlja novo znanje (vpogled) o problemu



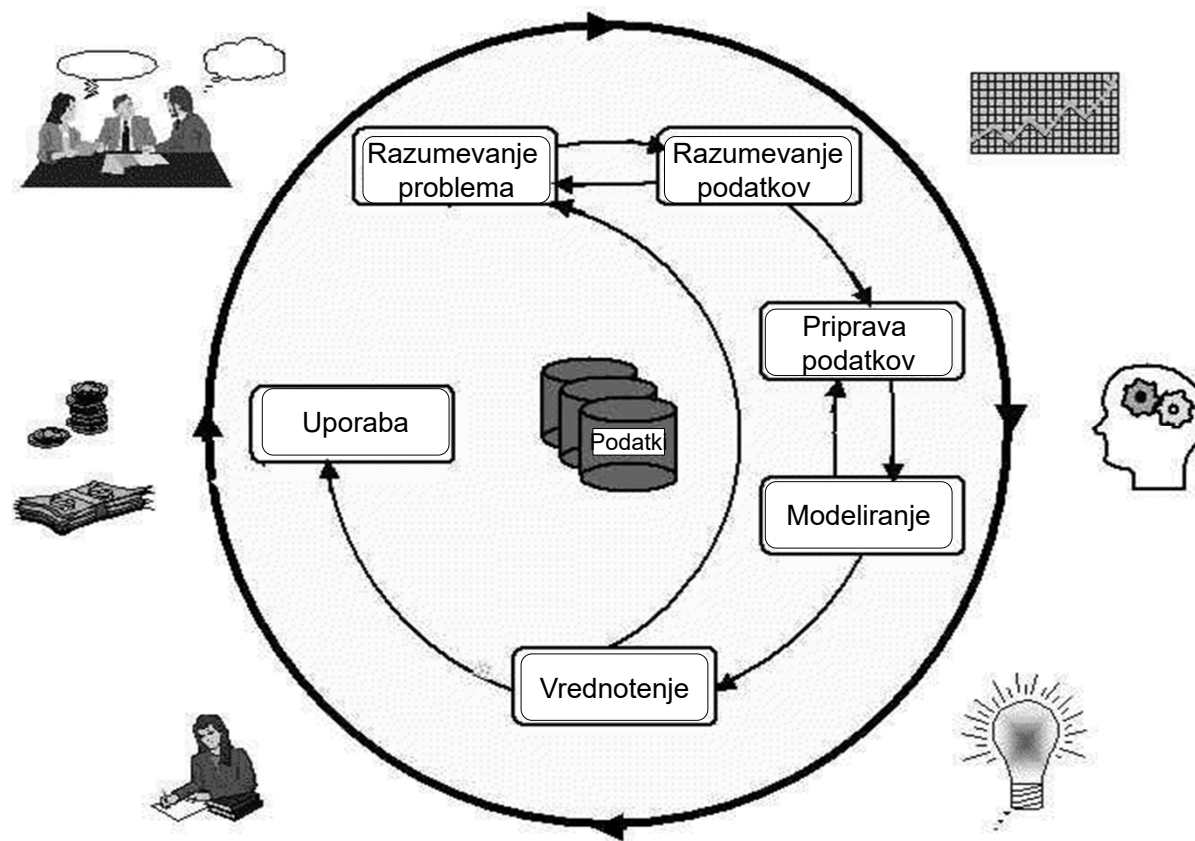
## CRISP-DM: standardiziran proces za podatkovno rudarjenje

---



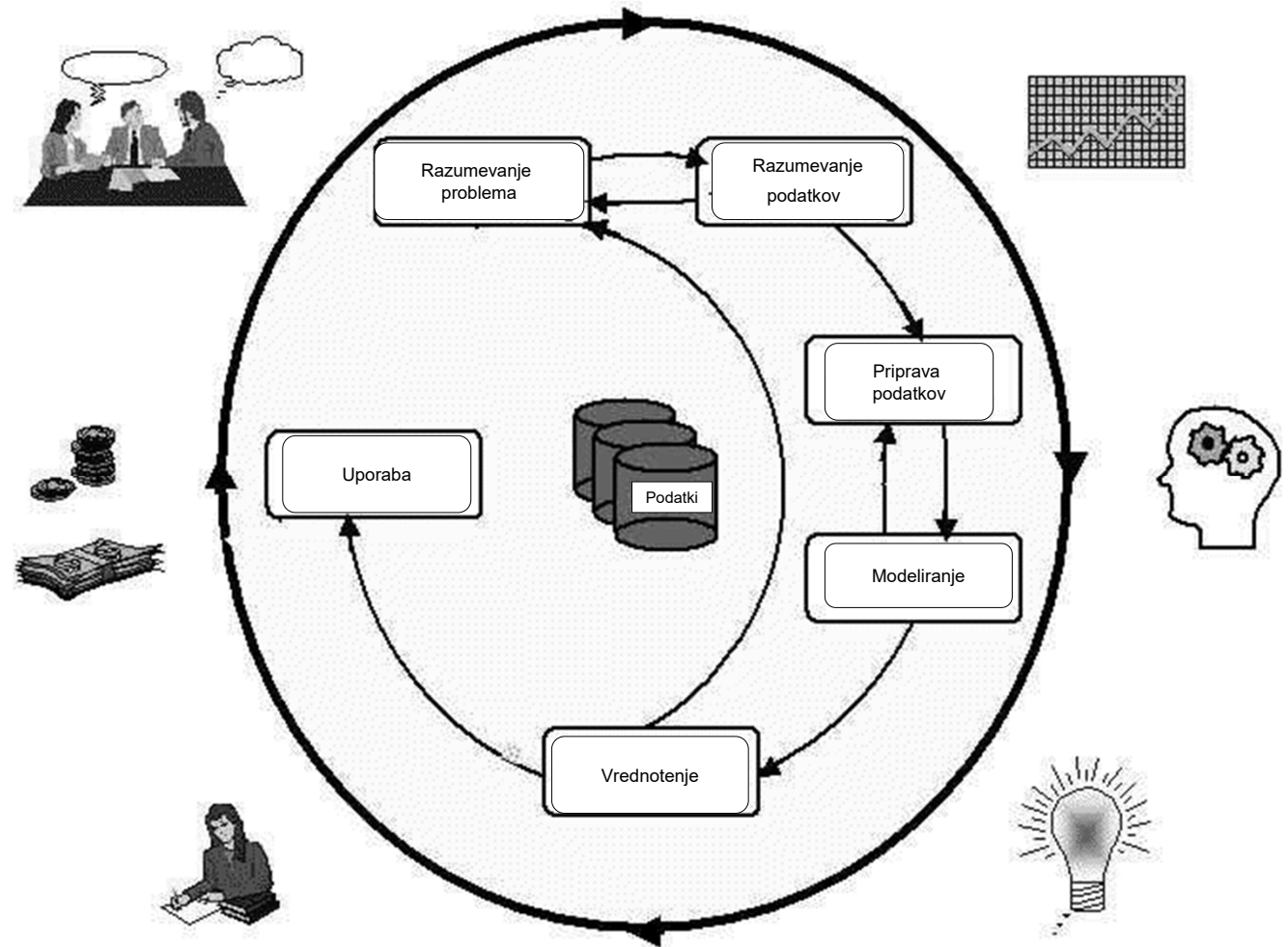
- Cross Industry Standard Process for Data Mining
- Financiran s strani Evropske komisije
- Trenutno je v CRISP-DM interesnem združenju več kot 200 vodilnih podjetij z vsega sveta
- Zakaj standard?
  - Proces mora biti zanesljiv in ponovljiv; tudi za ljudi z omejenim poznavanjem podatkovnega rudarjenja
  - Standard nudi okvir za opis postopka in omogoča ponovljivost DM projektov
  - Nudi pomoč pri načrtovanju in upravljanju DM projektov
  - Nudi trdno osnovo pri spoznavanju z DM
  - Daje vtis stabilnosti in zrelosti področja

# CRISP-DM proces



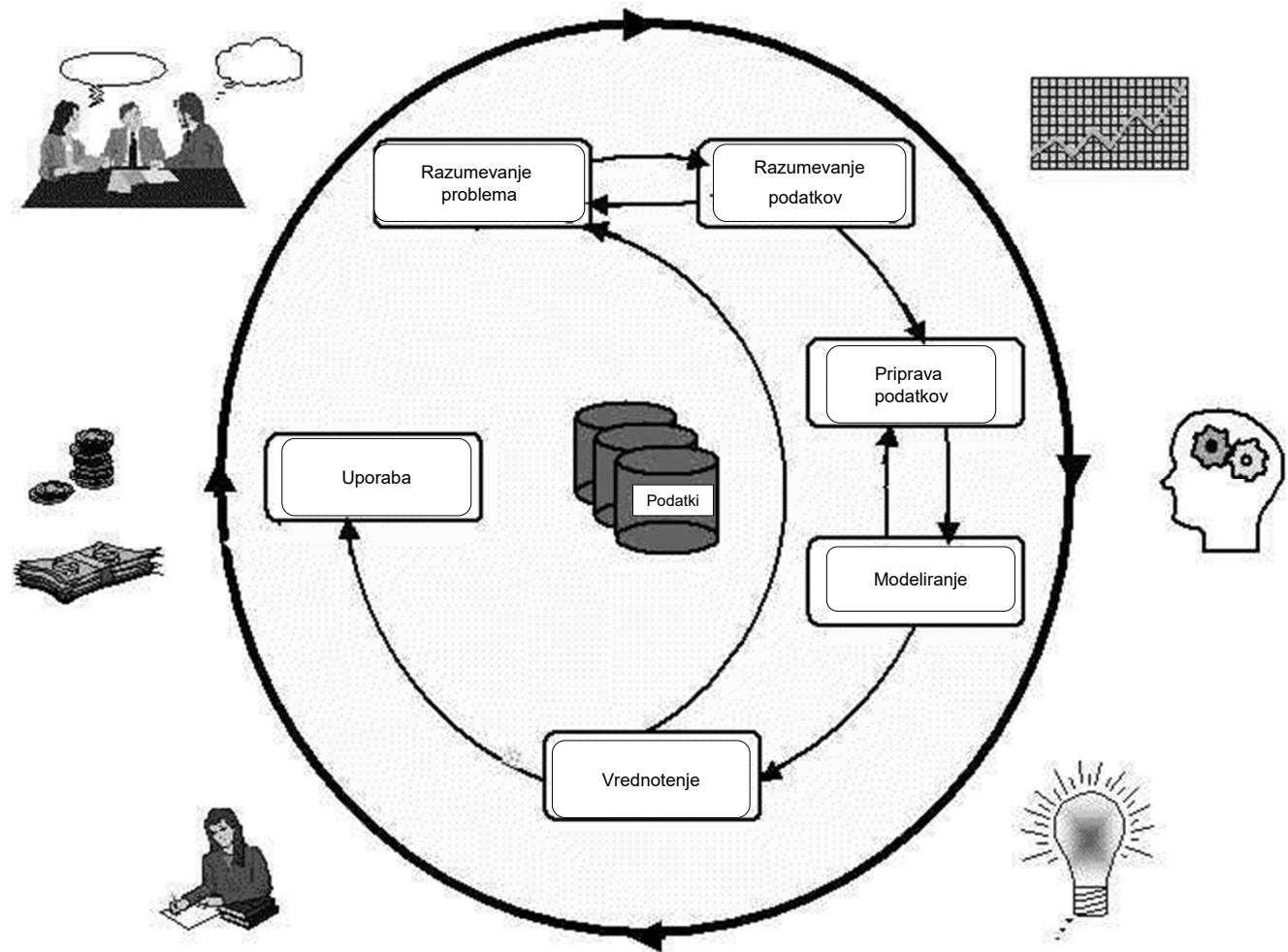
# 1. Razumevanje problema

- končni (aplikativni) cilj
- cilj podatkovnega rudarjenja
- kriterij uspešnosti



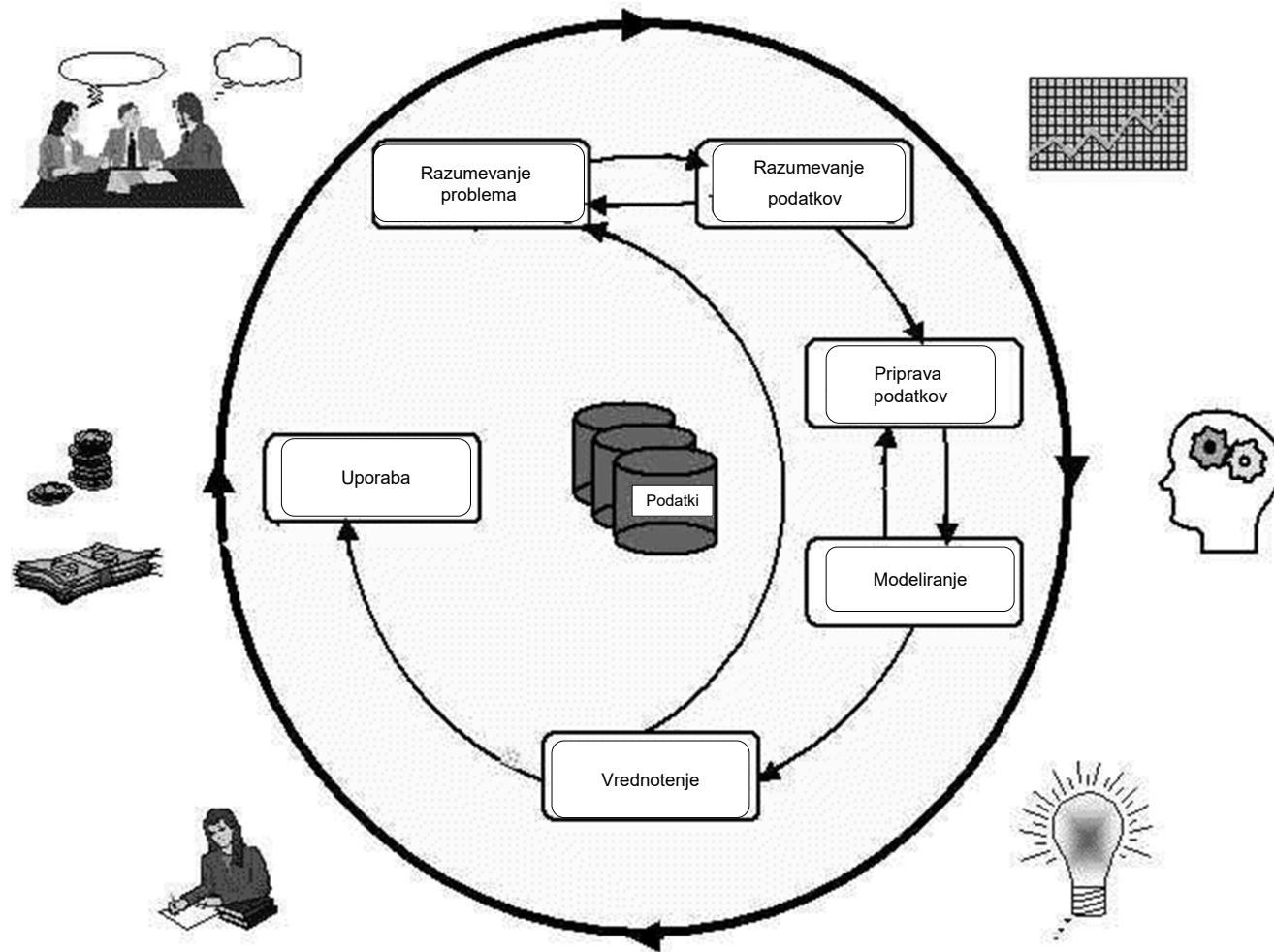
## 2. Razumevanje podatkov

- spoznavanje podatkov
- preverjanje kvalitete
- iskanje izjem



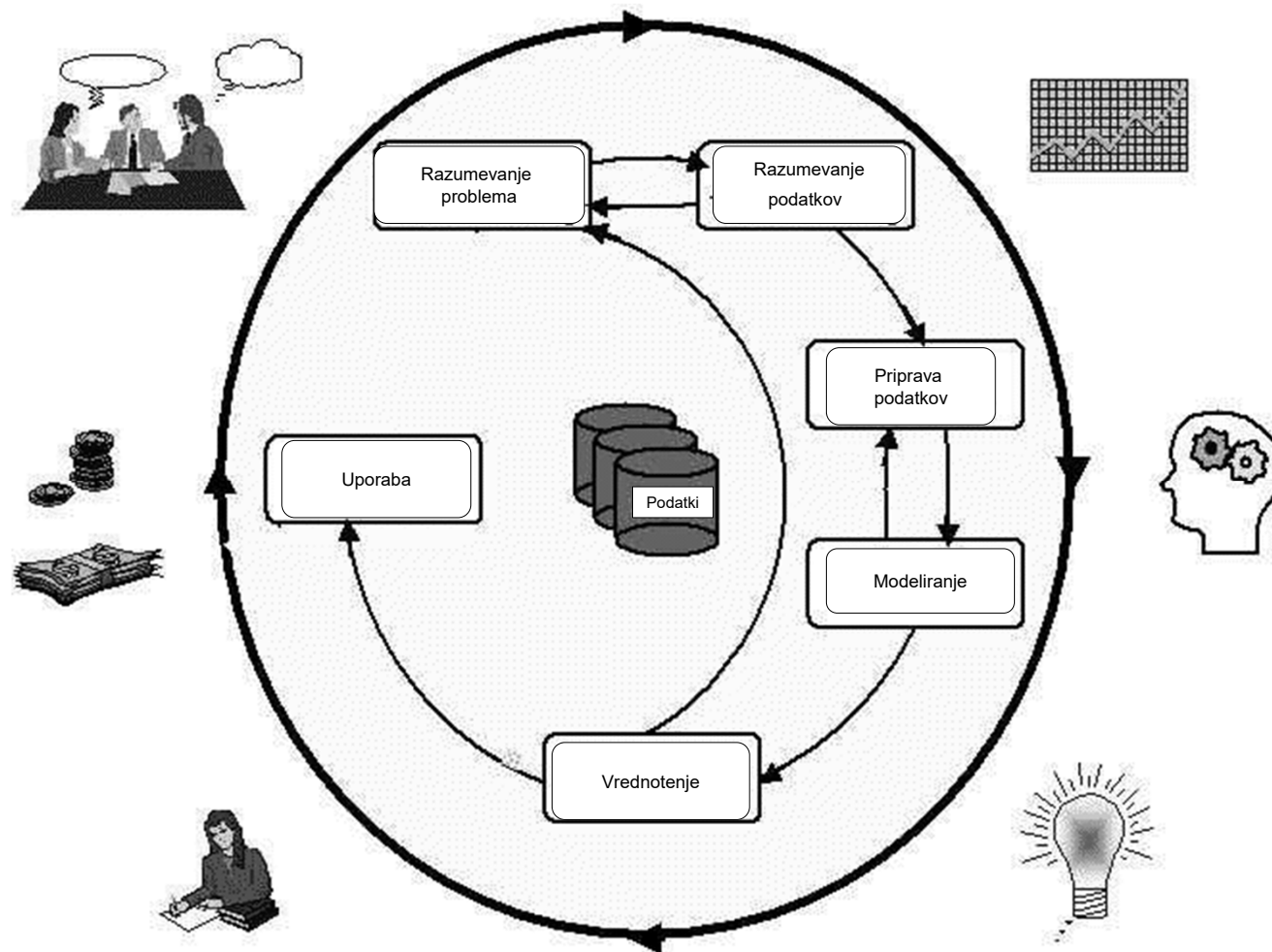
### 3. Priprava podatkov

- Zbiranje
  - Vrednotenje
  - Poenotenje
  - Čiščenje in filtriranje
  - Transformiranje
- 
- Priprava podatkov običajno vzame 90% celotnega časa



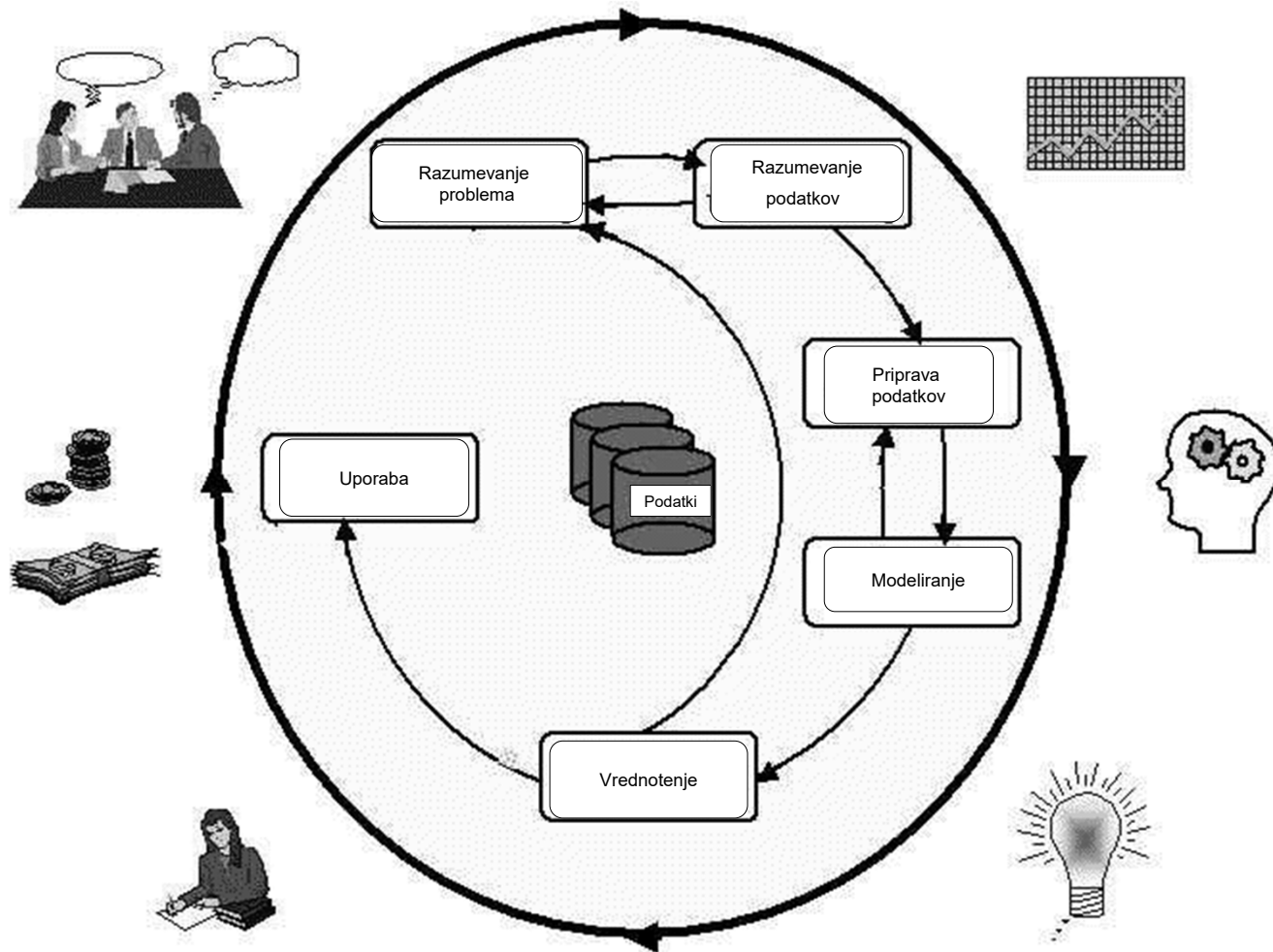
## 4. Modeliranje

- izbor primerne metode glede na cilj DM
- gradnja modela
- interno vrednotenje modela
- iterativen postopek (priprava-modeliranje)



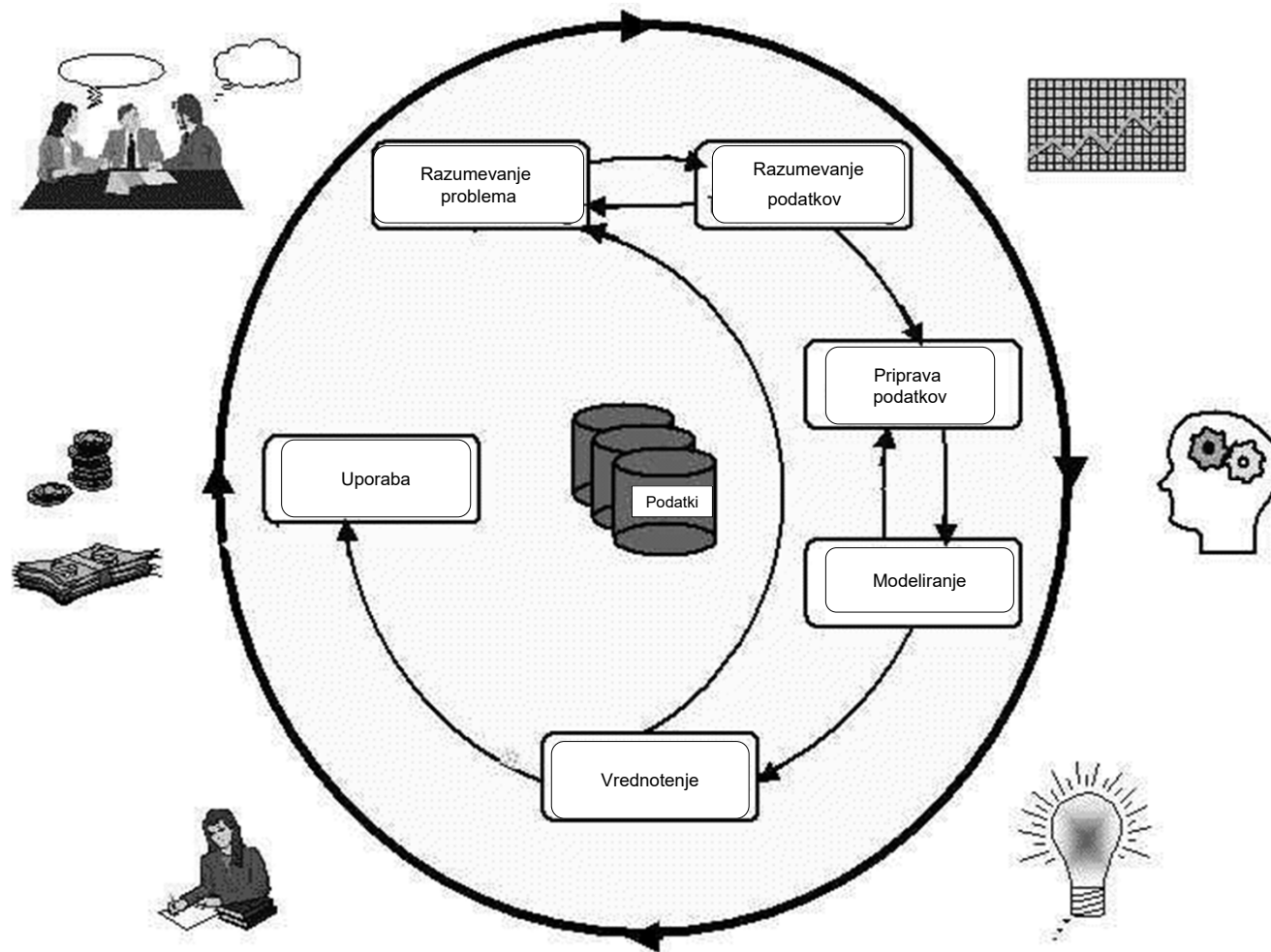
## 5. Vrednotenje

- Vrednotenje rezultatov DM glede na kriterij uspešnosti iz prve faze
- Sprejem konkretnih modelov
- Ocena procesa DM
- Odločitev o naslednjem koraku



## 6. Uporaba

- Ugotoviti kako, kdo in kdaj bo uporabljal rezultate (modele, odkrito znanje)
- Praktična uporaba zakonitosti, npr. nad novimi podatki, kot novo znanje
- Za praktično uporabo potrebujemo način za prenos modela v produkcijo



# Standardiziran opis modelov - PMML

---

- Pod okriljem *Data Mining Group - DMG* (več industrijskih partnerjev)
- Temelji na označevalnem jeziku XML
- Veljaven (november 2020) standard 4.4, razvoj teče naprej
- Nekateri podprte vrste modelov:
  - Odločitvena drevesa
  - Asociacijska pravila
  - Razvrščanje v skupine (clustering)
  - Naivni Bayesov klasifikator
  - Linearna in splošna regresija
  - Nevronske mreže
  - Sekvenčna pravila
  - Metoda podpornih vektorjev (SVM)
  - ...
- Mnogi pomožni elementi (slovar podatkov, statistike, transformacije)
- Možne so razširitve obstoječih vrst modelov

# Standardiziran opis modelov - ONNX



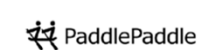
## ■ ONNX: open neural network exchange (november 2020: v1.81)

- Kritika PMML: poslovno okolje, XML (velikost!), slaba praktična sprejetost v odprtokodnih orodjih
- Predstavitev modelov v vmesni (grafni) predstavitvi
- Široka podpora odprtokodnih orodij (ne samo nevronske mreže)
- Odprtokoden izvor
- Podpora in sprejetost tudi v sodobnih podjetjih



LibSVM

MATLAB®



SIEMENS



# Posebnosti odkrivanja zakonitosti iz podatkovnih baz

---



- **Količina podatkov:**
  - Podatkovne baze so večje kakor delovni pomnilnik računalnika
  - Posledica: uporaba specializiranih, pogosto razmeroma enostavnih metod (zaradi hitrosti in porabe prostora)
- **Varnost:**
  - V mnogih primerih je zaželeno, da podatki ne zapuščajo za njih predvidene lokacije (podatkovne baze)
  - Posledica: implementacija metod čim bližje podatkovni bazi
- **Manjša fleksibilnost:**
  - V velikih podatkovnih sistemih posameznik nima kontrole nad celoto, ampak se mora držati določenih pravil
  - Posledica: manjši izbor metod, izvajanje postopkov "po kalupu"

# Struktura podatkov (primer: nadzorovano učenje)

---



- Pogled na podatkovno bazo v obliki tabele (pogleda)
- Vsaka vrstica predstavlja nov objekt (primer), opisan z množico značilk (atributov)
- Tabele lahko vsebujejo tudi atribut, ki predstavlja ciljno funkcijo modela (razred)
- Struktura atributov določena vnaprej (čeprav nekatere metodo omogočajo delo tudi z nestrukturiranimi, npr. tekstovno rudarjenje)
- Čimmanj manjkajočih (NULL) vrednosti
- Skrb za konsistentnost podatkov: podvajanje istih vrstic je lahko močno škodljivo

# Primer podatkov

atributi

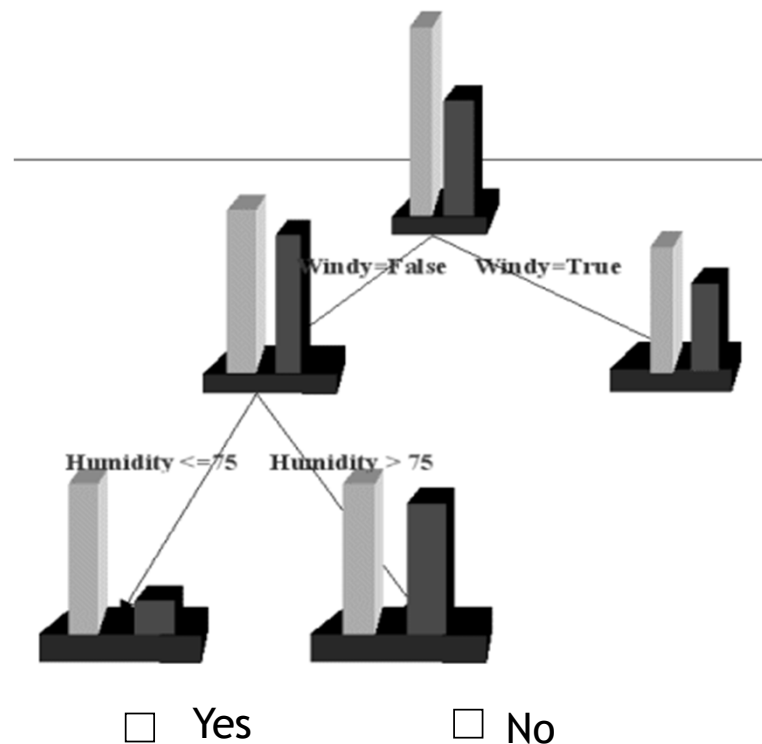
| Day | Outlook  | Temperature | Humidity | Wind   | Play Golf? |
|-----|----------|-------------|----------|--------|------------|
| 1   | Sunny    | Hot         | High     | Weak   | No         |
| 2   | Sunny    | Hot         | High     | Strong | No         |
| 3   | Overcast | Hot         | High     | Weak   | Yes        |
| 4   | Rain     | Mild        | High     | Weak   | Yes        |
| 5   | Rain     | Cool        | Normal   | Weak   | Yes        |
| 6   | Rain     | Cool        | Normal   | Strong | No         |
| 7   | Overcast | Cool        | Normal   | Strong | Yes        |
| 8   | Sunny    | Mild        | High     | Weak   | No         |
| 9   | Sunny    | Cool        | Normal   | Weak   | Yes        |
| 10  | Rain     | Mild        | Normal   | Weak   | Yes        |
| 11  | Sunny    | Mild        | Normal   | Strong | Yes        |
| 12  | Overcast | Mild        | High     | Strong | Yes        |
| 13  | Overcast | Hot         | Normal   | Weak   | Yes        |
| 14  | Rain     | Mild        | High     | Strong | No         |
| 15  | Sunny    | Hot         | Normal   | Weak   | ???        |

vrednosti atributov

posebej odlikovan atribut: razred

# Kaj lahko ugotovimo iz podatkov?

- Katere vrstice so si med seboj podobne, razdelimo jih na skupine
  - gručenje
- Katere kombinacije vrednosti atributov pogosto nastopajo skupaj (npr. Rain in Weak)
  - povezave, asociacije
- Kako lahko napovedujemo vrednost razrednega atributa (play golf): zgradimo napovedni model
  - nadzorovano učenje
- Model lahko uporabimo tudi za vpogled v strukturo problema
  - vsi tipi modelov niso primerni



Porazdelitev odgovorov na vprašanje, ali gremo igrat golf.

## Skrb za kvaliteto podatkov (slabo npr. podvajanje vrstic)

| Day | Outlook  | Temperature | Humidity | Wind   | Play Golf? |
|-----|----------|-------------|----------|--------|------------|
| 1   | Sunny    | Hot         | High     | Weak   | No         |
| 2   | Sunny    | Hot         | High     | Weak   | No         |
| 3   | Sunny    | Hot         | High     | Weak   | No         |
| 4   | Sunny    | Hot         | High     | Weak   | No         |
| 5   | Sunny    | Hot         | High     | Weak   | No         |
| 6   | Sunny    | Hot         | High     | Weak   | No         |
| 7   | Sunny    | Hot         | High     | Weak   | No         |
| 8   | Overcast | Cool        | Normal   | Strong | Yes        |
| 9   | Sunny    | Hot         | High     | Weak   | No         |
| 10  | Sunny    | Hot         | High     | Weak   | No         |
| 11  | Sunny    | Hot         | High     | Weak   | No         |
| 12  | Sunny    | Hot         | High     | Weak   | No         |
| 13  | Sunny    | Hot         | High     | Weak   | No         |
| 14  | Sunny    | Hot         | High     | Weak   | No         |
| 15  | Sunny    | Hot         | High     | Weak   | No         |

# Delitev metod podatkovnega rudarjenja

---



- Deduktivne (uporabnik postavi hipotezo o modelu)
  - Sprotno analitično procesiranje (*on-line analytical processing* - OLAP)
  - Statistične metode (raziskovalna in potrjevalna analiza podatkov)
- Induktivne (metoda postavi hipotezo o modelu)
  - Metode strojnega učenja in umetne inteligence

# Tipi metod za podatkovno rudarjenje

---



- Statistične metode (raziskovalna in potrjevalna analiza podatkov)
  - Analitik postavi in z izbrano statistično metodo striktno preveri hipotezo
  - Potrebna prilagoditev za delo z velikimi množicami podatkov
- Sprotno analitično procesiranje (*on-line analytical processing - OLAP*)
  - OLAP: večdimenzionalno grupiranje in povzemanje podatkov
  - Analitik postavi in “preveri” hipotezo
  - Metode prilagojene za delo z velikimi množicami podatkov
  - Problem pri veliki dimenzionalnosti podatkov
- Metode strojnega učenja
  - Avtomatsko postavljanje in preverjanje hipoteze
  - Učenje kot modeliranje podatkov
  - Pogosto potrebna prilagoditev za delo z velikimi množicami podatkov

# Metode glede na način uporabe modela

---

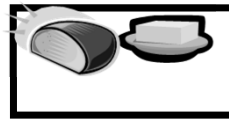


- Napovedovalne: model uporabimo za napovedovanje, kaj se bo zgodilo v prihodnosti ali z novimi podatki
- Opisovalne: model uporabimo za (človeku razumljiv) opis podatkov
- Nekaterne napovedovalne metode lahko uporabimo tudi kot opisovalne
- Posebej nas zanimajo induktivne metode, torej tiste, ki samostojno izdelajo na osnovi zbranih podatkov
- Področje umetne inteligence: strojno učenje (nadzorovano za napovedovalne metode, nenadzorovano za opisovalne metode)

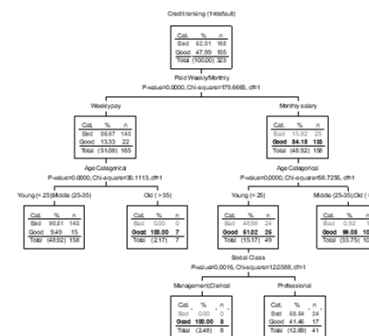
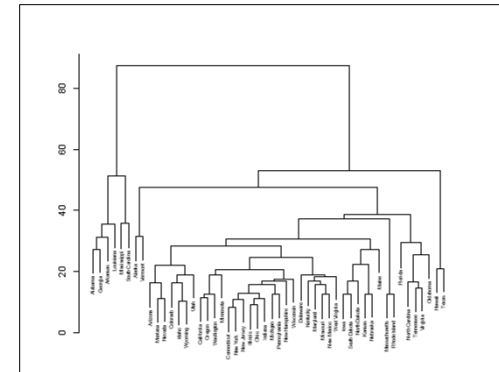
# Podatkovno rudarjenje

| Property Type | City    | Time  | Total Revenue |
|---------------|---------|-------|---------------|
| Flat          | Glasgow | Q1    | 15056         |
| House         | Glasgow | Q1    | 14670         |
| Flat          | Glasgow | Q2    | 14555         |
| House         | Glasgow | Q2    | 15888         |
| Flat          | Glasgow | Q3    | 14578         |
| House         | Glasgow | Q3    | 16004         |
| Flat          | Glasgow | Q4    | 15890         |
| House         | Glasgow | Q4    | 15500         |
| Flat          | London  | Q1    | 19678         |
| House         | London  | Q1    | 23877         |
| Flat          | London  | Q2    | 19567         |
| House         | London  | Q2    | 28677         |
| .....         | .....   | ..... | .....         |
| .....         | .....   | ..... | .....         |

- OLAP
- skupine
- asociacije
- modeliranje podatkov



↓ (80%, 10%)



## Opisovalno modeliranje (nenadzorovano učenje)

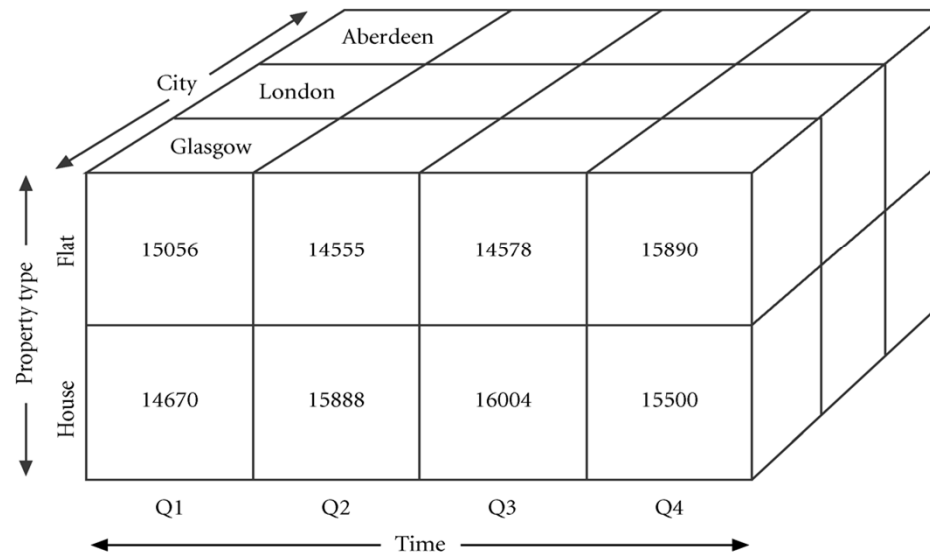
---



- Nimamo usmeritve, ki bi nadzorovala učenje
- Pomembno pri začetni analizi podatkov
- Pristopi
  - OLAP (on-line analytical processing)
    - Temelji na agregatih po različnih dimenzijah
  - Razvrščanje v skupine (gručenje, ang. *clustering*)
    - Temelji na grupiranju podobnih objektov
    - Mera podobnosti je zelo pomembna
  - Povezovalna analiza (odkrivanje asociacij)
    - Temelji na odkrivanju povezav med podmnožicami atributov

# OLAP - grupiranje in povzemanje podatkov po različnih dimenzijah (atributih)

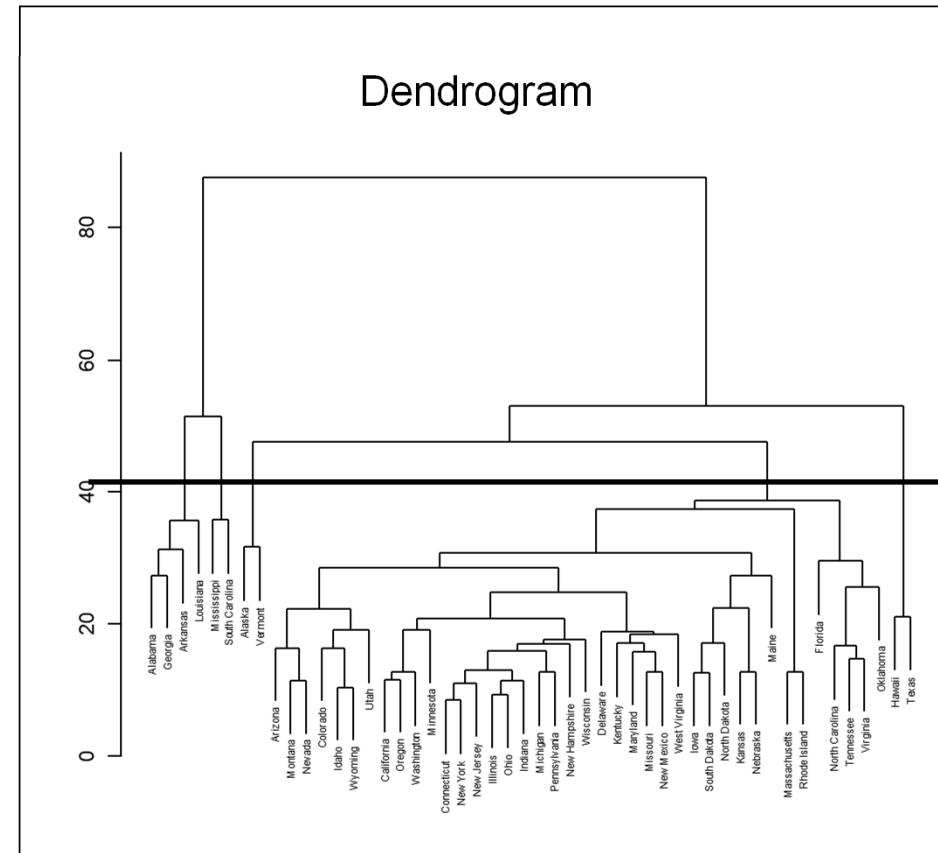
| Property Type | City    | Time  | Total Revenue |
|---------------|---------|-------|---------------|
| Flat          | Glasgow | Q1    | 15056         |
| House         | Glasgow | Q1    | 14670         |
| Flat          | Glasgow | Q2    | 14555         |
| House         | Glasgow | Q2    | 15888         |
| Flat          | Glasgow | Q3    | 14578         |
| House         | Glasgow | Q3    | 16004         |
| Flat          | Glasgow | Q4    | 15890         |
| House         | Glasgow | Q4    | 15500         |
| Flat          | London  | Q1    | 19678         |
| House         | London  | Q1    | 23877         |
| Flat          | London  | Q2    | 19567         |
| House         | London  | Q2    | 28677         |
| .....         | .....   | ..... | .....         |
| .....         | .....   | ..... | .....         |



Sorodno: vrtilne tabele v Excelu - preprosto povzemanje, manjša količina podatkov

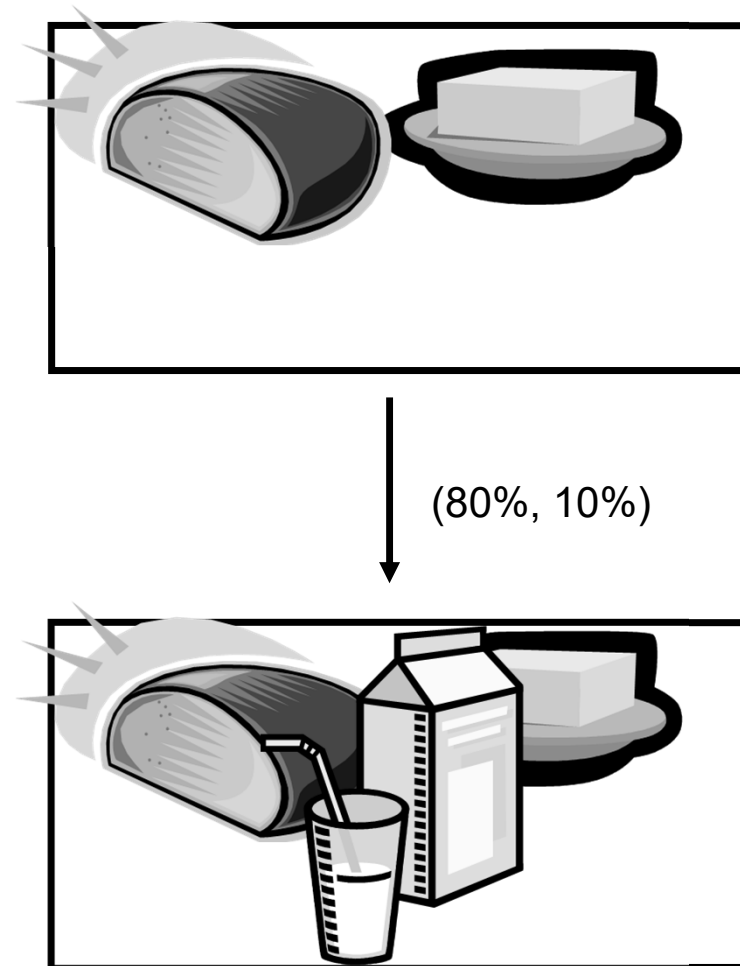
# Gručenje, razvrščanje v skupine (clustering)

- Razvrščanje podatkov v skupine (gruč) glede na njihovo *podobnost*
- Definicija podobnosti odvisna od problema in tipov atributov
- Število gruč je lahko (ni pa nujno) vnaprej določeno
- Običajni statistični algoritmi so slabo skalabilni, zato v velikih bazah potrebujemo specializirane implementacije (npr. *DBScan*, *Oracle OCluster*)
- Primer: hierarhično gručenje



# Povezovalna analiza

- Temelji na opazovanju kombinacij atributov, ki pogosto nastopajo skupaj
- Pogosto se uporablja v problemu analize "nakupovalne košarice":  
*kruh & maslo* → *mleko*  
(zaupanje 80%, podpora 10%)
- Problem kombinatorične eksplozije ( $2^N$  potencialnih podmnožic)
- Algoritem Apriori (Agrawal, 1995) naredi asociacijsko analizo praktično uporabno
- Uporaba asociacijskih pravil



# Napovedovalno modeliranje (nadzorovano učenje)

- Učenje kot gradnja modela, ki ga lahko uporabimo za napovedovanje vrednosti vnaprej določenega atributa (razreda)

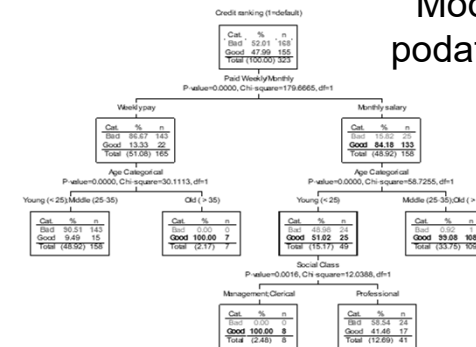
Podatki (npr. v podatkovni bazi)



? Podatki z ne-  
znanim razredom



Model  
podatkov



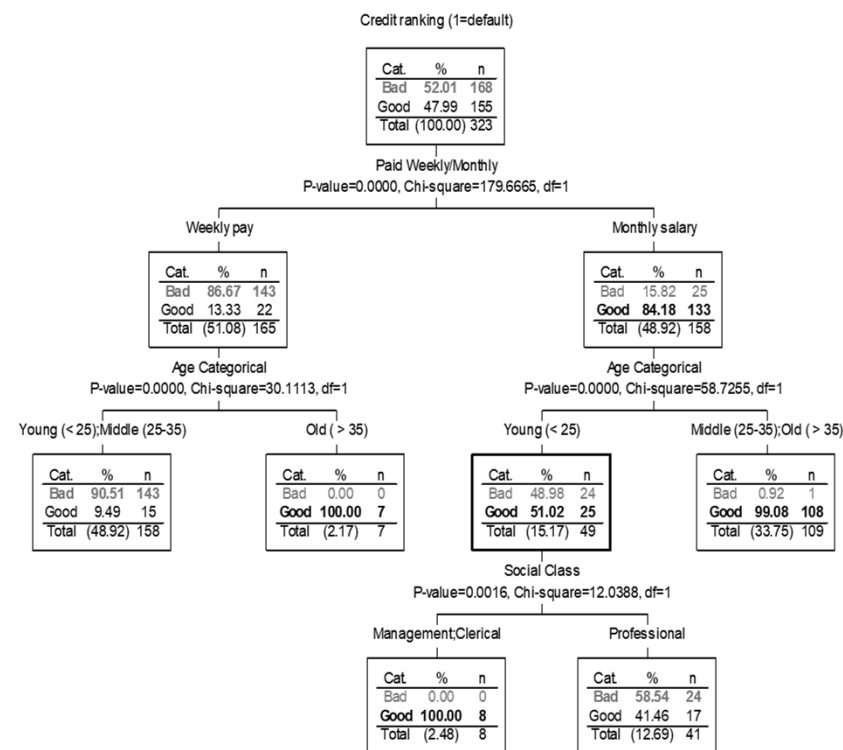
Napoved ne-  
znanega razreda



- Pri izbranem učnem algoritmu je gradnja modela samodejna na osnovi učnih podatkov
- Model predstavlja formaliziran opis problema

# Odločitvena drevesa

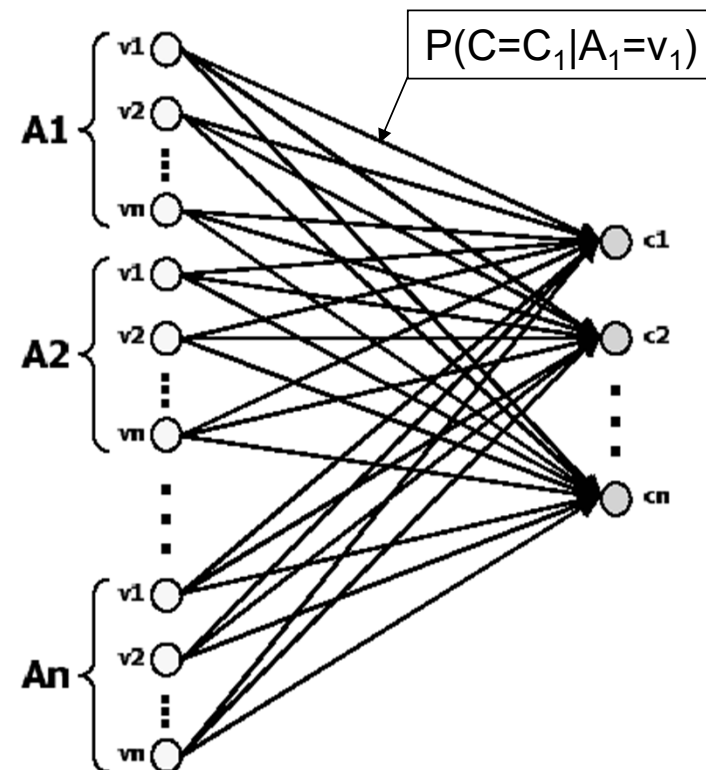
- Podatke rekurzivno delijo na podmnožice
- V vsakem vozlišču se nahaja kriterij delitve (npr.  $cena > 10$ ), kvalitetnejši so na vrsti višje v drevesu
- Cilj delitve je dobiti čimbolj čiste množice glede na ciljni atribut (razred)
- Pomembnost funkcije za oceno kvalitete atributov
- Razmeroma hitra metoda, modeli so človeku razumljivi, obstajajo modifikacije algoritmov za gradnjo dreves na zelo velikih podatkovnih bazah (Domingos, 2004)



# Naivni Bayesov klasifikator

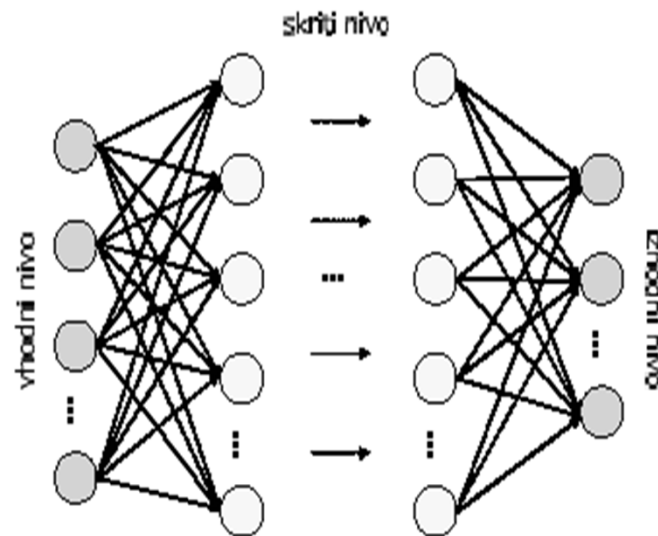
$$P(C=c|A_1=v_1, A_2=v_2, \dots)$$

- Temelji na izračunu pogojne verjetnosti razreda pri podanih vrednostih atributov
- Naivnost: predpostavka pogojne neodvisnosti atributov
- Implementacija v obliki tabele pogojnih frekvenc (verjetnosti)
- Zelo enostavna, hitra in presenetljivo zmogljiva metoda
- Možnost človeku razumljive interpretacije modelov
- Obstajajo implementacije, ki omilijo predpostavko neodvisnosti (npr. delno naivni Bayesov klasifikator, Oraclova prilagodljiva Bayesova mreža ABN), vendar za ceno mnogo daljšega časa učenja



# Nevronske mreže (plitve in globoke)

- Preprosta struktura, primerna za vzporedno izvajanje
- Pragovne aktivacijske funkcije
- Velika sposobnost modeliranja
- Zelo priljubljena metoda na mnogih področjih
- Dolgi učni časi izvajanja na zaporednih arhitekturah
- Težavna interpretacija modelov
- Pogosto preveliko prilagajanje učnim podatkom
- Plitve: 1-2 skrita nivoja, polno povezana
- Globoke: različni tipi nivojev, >> 2 skrita nivoja (tipično 10-100)



## Množica drugih popularnih metod

- Metoda podpornih vektorjev (SVM)
  - Naključni gozdovi (random forests)
  - Ansambelske metode (bagging, boosting, stacking)
  - Gradientni boosting
  - ...
- 
- Kaggle tekmovanja v podatkovnem rudarjenju:  
običajno je bolj kot izbor metode pomembno  
predprocesiranje in izračun novih, boljših atributov

## Kaj potrebujemo za podatkovno rudarjenje?

---



- Podatki (najbolje iz podatkovne baze/skladišča)
- Proces (CRISP-DM)
- Orodja (aplikacije)
  
- Koristno:
  - Prisotnost problemskega eksperta
  - Prisotnost eksperta za podatkovno rudarjenje

# Razvrstitev orodij (aplikacij)

---



- Glede na namembnost
  - Raziskovalna
  - Komericalna
- Glede na dostop do podatkov
  - Tekstovne datoteke, drugi viri (API)
  - Branje iz podatkovne baze
- Glede na mesto rudarjenja
  - Lokalno (pri uporabniku)
  - Oddaljeno (znotraj podatkovnega strežnika)

# Komercialna orodja

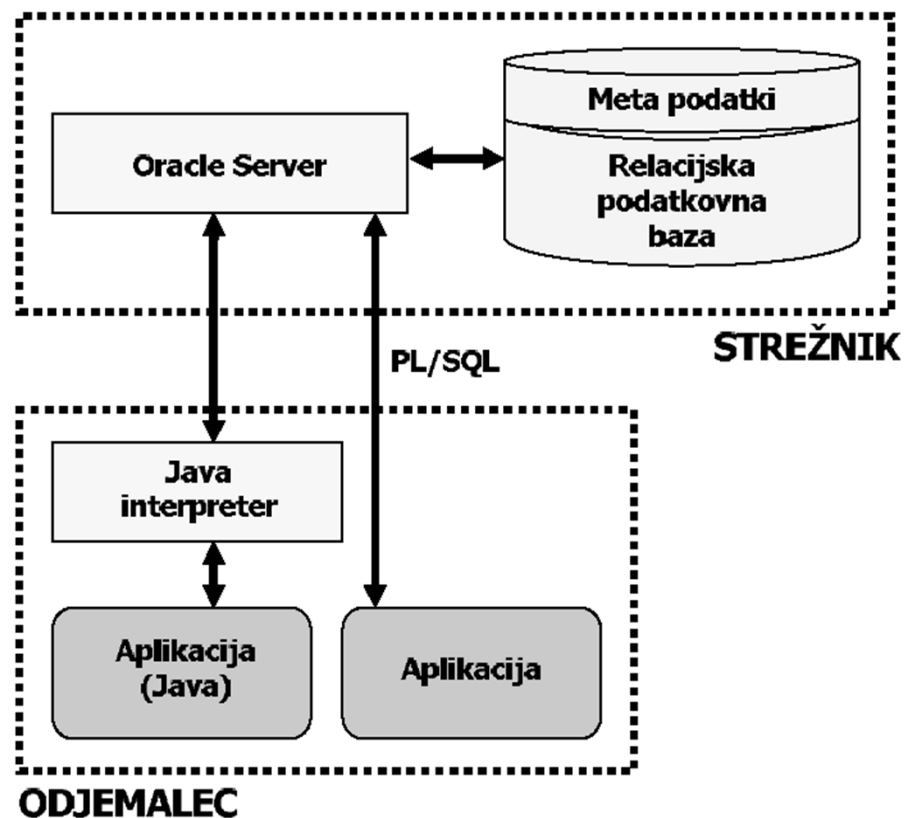
---



- Strežniško usmerjena (procesiranje na strežniku), izvor: podatkovne baze, poslovne aplikacije:
  - IBM DB2 Intelligent Miner for Data
  - Oracle Data Mining
  - Microsoft SQL Server Analysis Services
- Odjemalčevsko usmerjena (procesiranje pri odjemalcu), izvor: statistični programi:
  - IBM SPSS Modeler (Clementine)
  - SAS Enterprise Miner
  - Insightful Miner
  - Statistica Data Miner

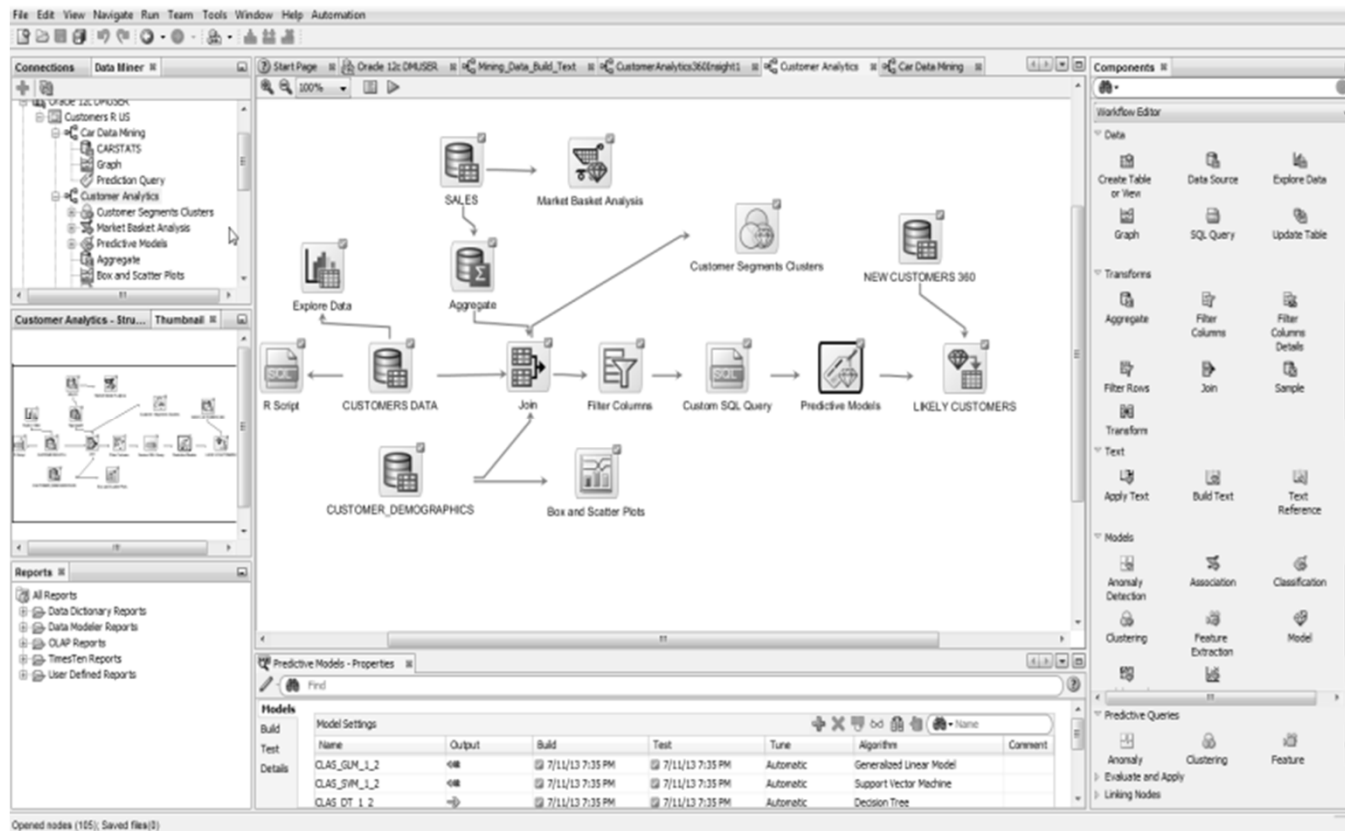
# Oracle Data Mining – ODM

- Del Oracle Advanced Analytics (OAA)
- Vse analitske metode (OLAP, DM) se izvajajo na strežniku
- Odjemalec omejen na nadzor delovanja
- Modeli se hranijo na strežniku
- Hitrost (počasnost) izvajanja
- Metode:
  - filtriranje, diskretizacija, ocenjevanje atributov
  - Naivni Bayesov klasifikator
  - Linearni modeli (GLM)
  - SVM, tekstovno rudarjenje, primerjava podzaporedij, odločitvena drevesa
  - Vizualizacije podatkov in rezultatov



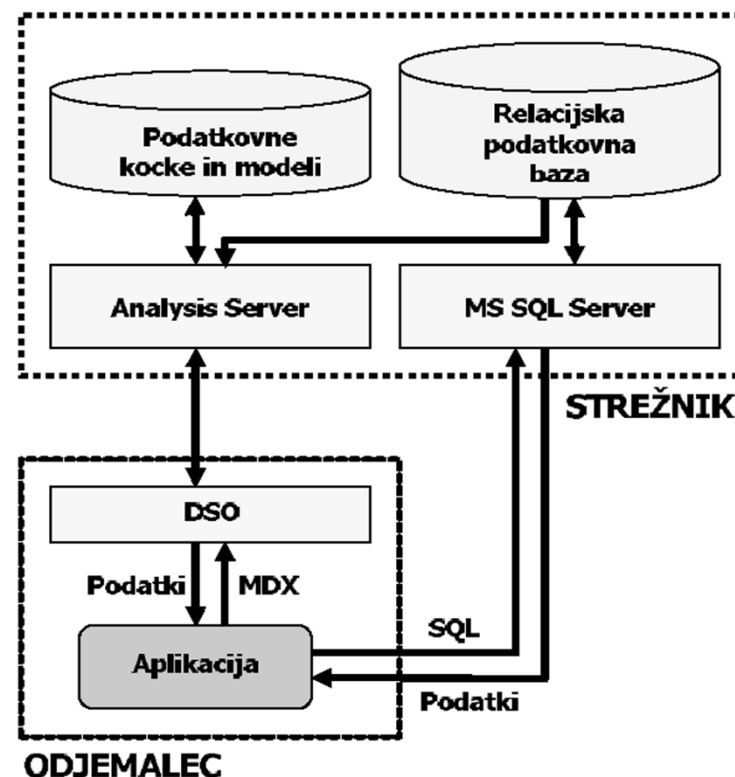
# Oracle Data Mining - ODM

- Vizualna specifikacija toka opravil (workflow)
- Povezava z običajnimi analitskimi orodji (npr. R)



# Microsoft SQL server (od 2000 dalje)

- dodaten strežniški modul (Analysis services)
- celotno procesiranje na strežniku, kontrola preko SQL
- SQL Server 2000: skromna podpora strojnemu učenju (odločitvena drevesa, gručenje)
- SQL server 2005/8 dodaja nove metode:
  - regresijska drevesa
  - naivni Bayesov klasifikator
  - asociacijska pravila
  - nevronske mreže
- Kasnejše verzije imajo le malenkostne dopolnitve



# IBM SPSS Modeler

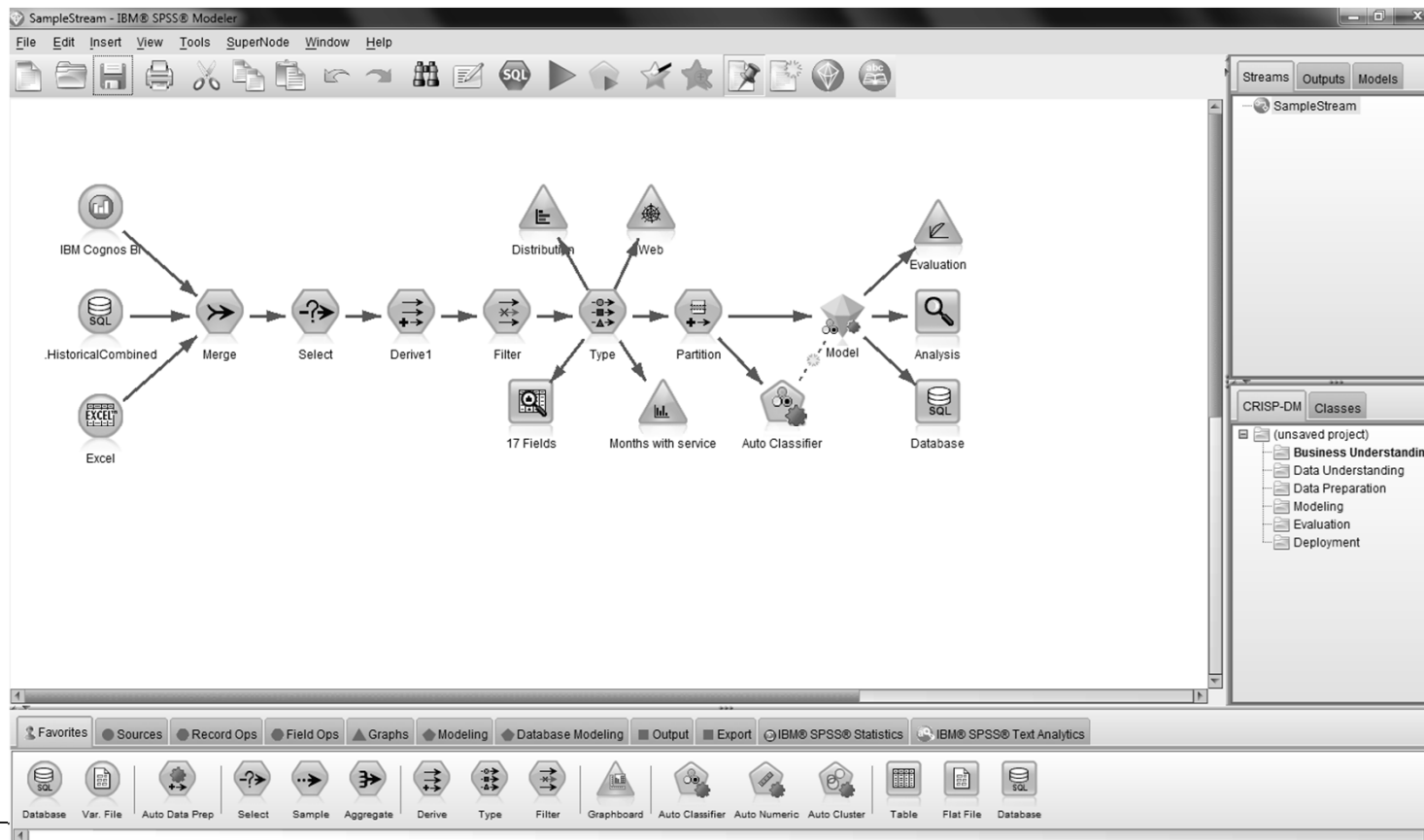
---



- Originalno (SPSS) Clementine
- Podatkovno, spletno in tekstovno rudarjenje
- Vir podatkov: tekstovne datoteke, podatkovne baze (ODBC), spletne strani
- Omogoča grafično načrtovanje rudarjenja
- Metode:
  - nevronske mreže
  - odločitvena, klasifikacijska in regresijska drevesa,
  - gručenje, povezovalna analiza (asociacijska pravila)
  - filtriranje podatkov in rezultatov
  - ...

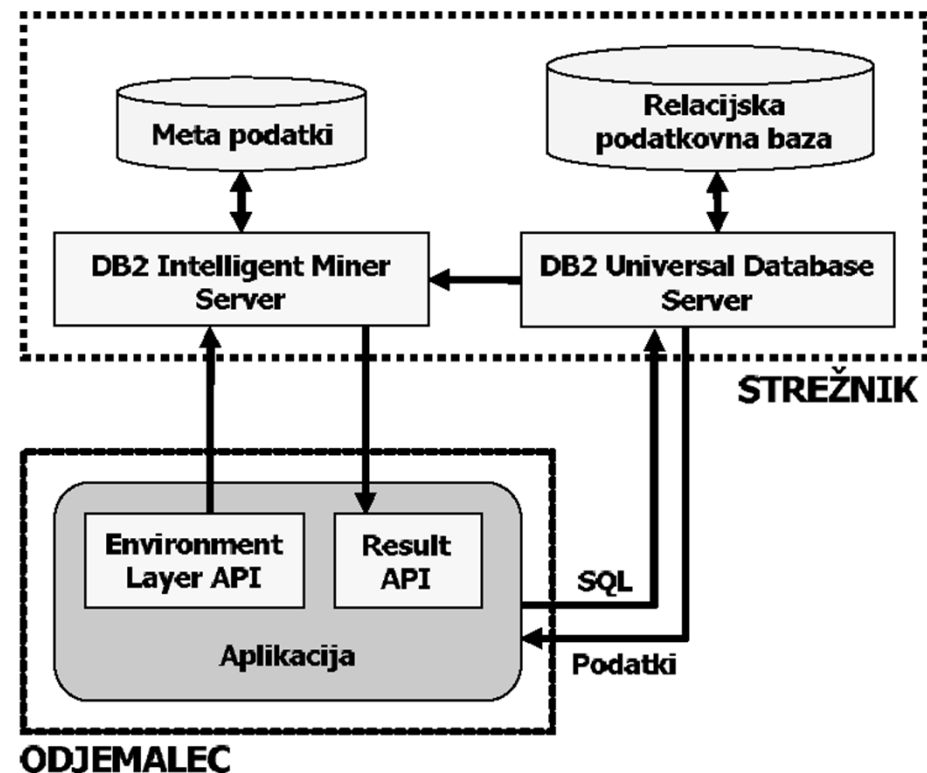
# IBM SPSS Modeler

- Grafično načrtovanje podatkovnega rudarjenja

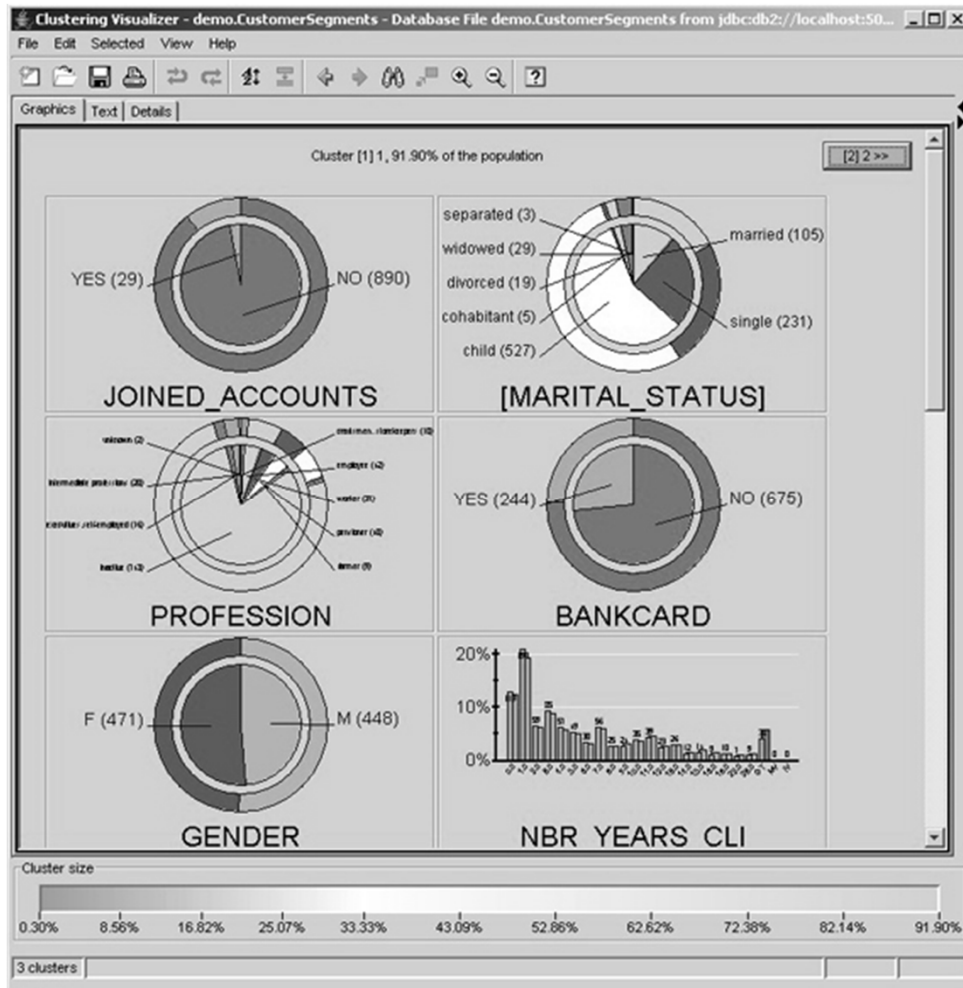


# IBM DB2 Intelligent Miner

- samostojna aplikacija, osnovana na C++ API
- arhitektura odjemalec/strežnik, ni vezana na DB2 bazo
- podatkovno rudarjenje na strežniku
- interpretacija rezultatov na odjemalcu
- implementirane metode:
  - gručenje, asociacijska analiza
  - odl. drevesa, nevronske mreže, najbližji sosed, ...
  - filtriranje, diskretizacija, ocenjevanje atributov
  - statistične metode



# IBM DB2 Intelligent Miner



# Raziskovalna orodja

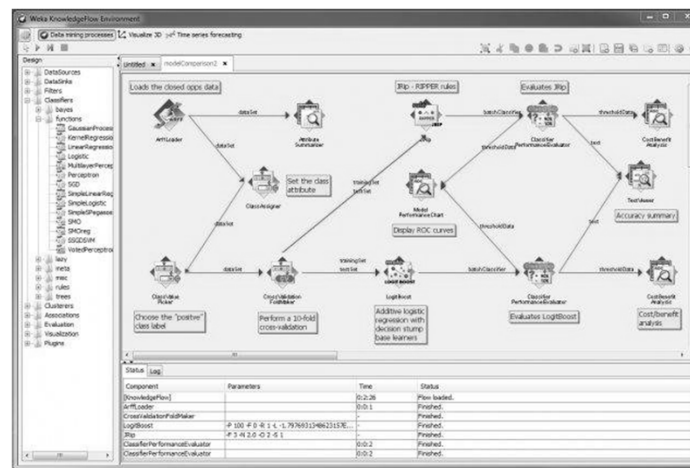
---



- Obstaja cela množica orodij in knjižnic
- Večinoma specializirana
- Običajno razvita z določenim namenom
- Splošnonamenska, dovolj široko uporabna
  - Scikit-Learn
  - Orange
  - WEKA
  - ...
- Izvajajo se zunaj SUPB, potrebujejo dostop do podatkov

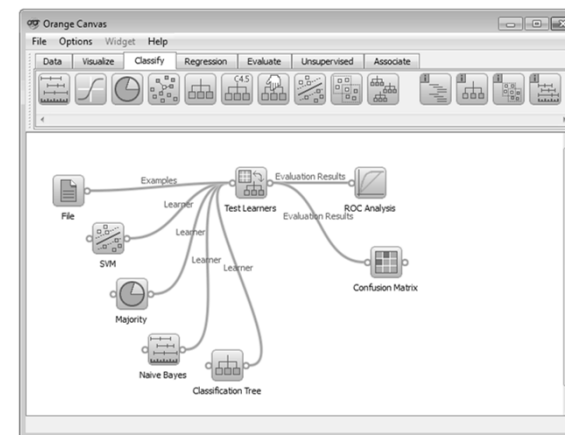
# Weka

- Razvita na univerzi Waikato, Nova Zelandija ([www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka))
- Napisana v Javi, zato binarno prenosljiva
- Veliko implementiranih metod:
  - Odločitvena drevesa in pravila
  - Naivni Bayesov klasifikator
  - Odločitvene tabele
  - Metoda podpornih vektorjev (SVM)
  - Metoda najbližjih sosedov
  - Logistična in linearna regresija
  - Nevronske mreže
  - Meta-učne sheme (boosting, bagging, stacking, ...)
- Dostop do podatkov: tekstovne datoteke, JDBC
- Relativna počasnost, pomnilniška zahtevnost



# Orange

- Razvit na FRI (orange.biolab.si)
- Kombinacija C++ (procesiranje) / Python (nadzor, sklearn)
- Omogoča grafično načrtovanje rudarjenja (Orange Widgets)
- Enostavno razširljiv in prenosljiv (Windows, Linux, Mac)
- Med najbolj fleksibilnimi in zmogljivimi orodji
- Vsebuje množico metod, med drugim:
  - Asociacijska analiza
  - Klasifikacijska in regresijska drevesa
  - Rojenje
  - Funkcijska dekompozicija
  - Logistična regresija
  - Naivni Bayesov klasifikator
  - Metoda podpornih vektorjev (SVM)
- Dostop do podatkov običajno preko tekstovnih datoteko
- Možnost neposrednega dostopa do podatkovne baze

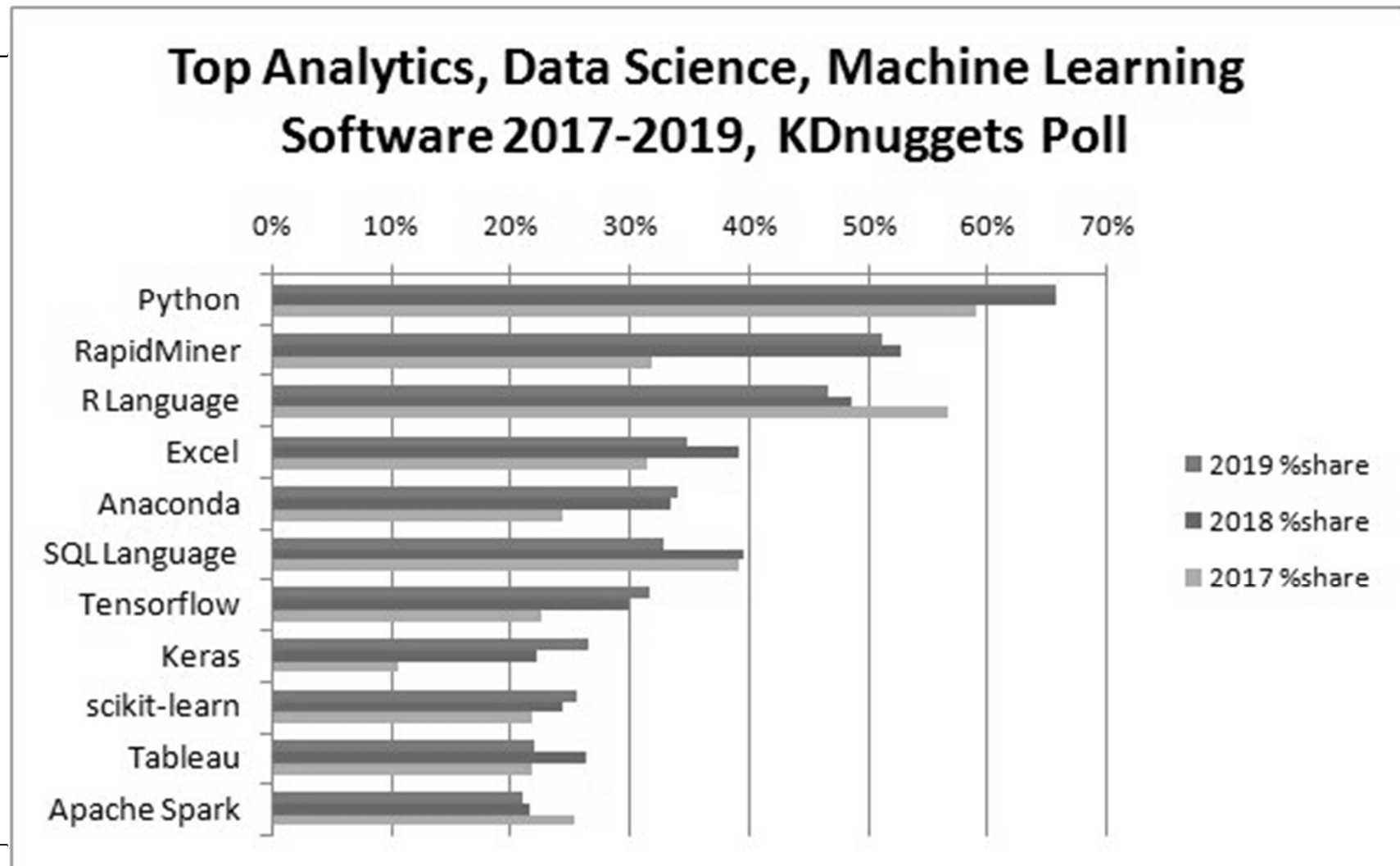


# Oracle Data Mining: praktična izkušnja

- Java ali PL/SQL aplikacija, ki nadzoruje procese na strežniku
- Podatki ne zapuščajo strežnika, zato posebej primerno za kritične aplikacije (varnost)
- Ni (praktičnih) omejitev glede velikosti vhodnih podatkov
- Razmeroma skromno okolje glede implementiranih algoritmov in nadzora nad njimi
- Počasnost! Primer: tabela s 600.000 vrsticami in 30 stolpci

| Oracle DM | Orange       | Weka                                                       |
|-----------|--------------|------------------------------------------------------------|
| Pol ure   | Nekaj sekund | Zmanjkalo pomnilnika (JVM default heap), sicer pod sekundo |

# KDD Nuggets ankete 2017-2019



# Zaključki

---



- Vedno več uporabnih odprtokodnih in komercialnih produktov, z izvorom predvsem iz analize podatkov (statistični produkti) in hranjenja podatkov (podatkovne baze)
- Vedno več produktov integriranih s podatkovnimi bazami in zato primernih za uporabo v (tradicionalno konzervativnih) poslovnih aplikacijah