Poglavje 4

Odkrivanje zakonitosti v podatkih (Podatkovno rudarjenje, Data Mining)

- ali drugače rečeno –

Kako z umetno inteligenco odkrijemo **koristne** stvari (vzorce, anomalije, zakonitosti, znanje, ...) v podatkovni bazi

Kaj je OZP?

- Odkrivanje zakonitosti v podatkih je proces identifikacije veljavnih, novih, potencialno uporabnih in razumljivih vzorcev **v podatkih**, kar lahko imenujemo tudi znanje.
  - Vzorci
  - Povezave (asociacije)
  - Spremembe
  - Anomalije
  - ...

  ⇒ **Nova (spo)znanja**

Drugi izrazi za OZP

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

Zakaj odkrivati zakonitosti v podatkovnih bazah?

- Podatki in 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 podatkovnih bazah so podatki pogosto "doma" in še posebej lepo organizirani
- V podatkovni bazi želimo odkriti neko novo, koristno znanje (ali splošneje: zakonitost)
- Kako?

- Odkrivanje zakonitosti v podatkovnih bazah kot *standardiziran* proces (podatkovno rudarjenje oz. *data mining*)

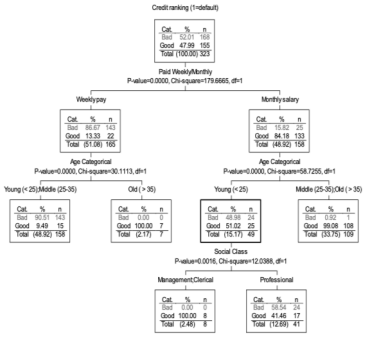
Podatkovno rudarjenje (*data mining*)

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

Podatki (npr. v podatkovni bazi) ali skladišču)

Sistem za podatkovno rudarjenje

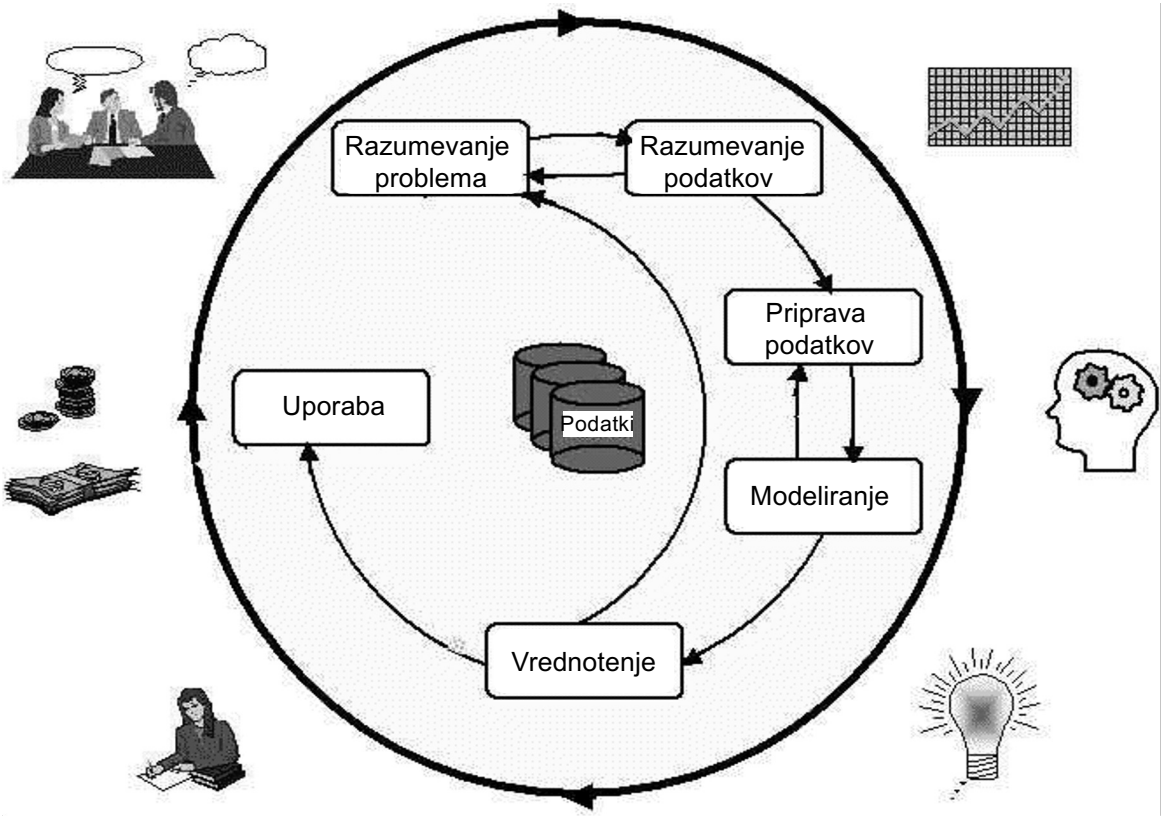
Model podatkov (odločitveno drevo, globoka nevronska mreža, …)



CRISP-DM: standardiziran proces za podatkovno rudarjenje

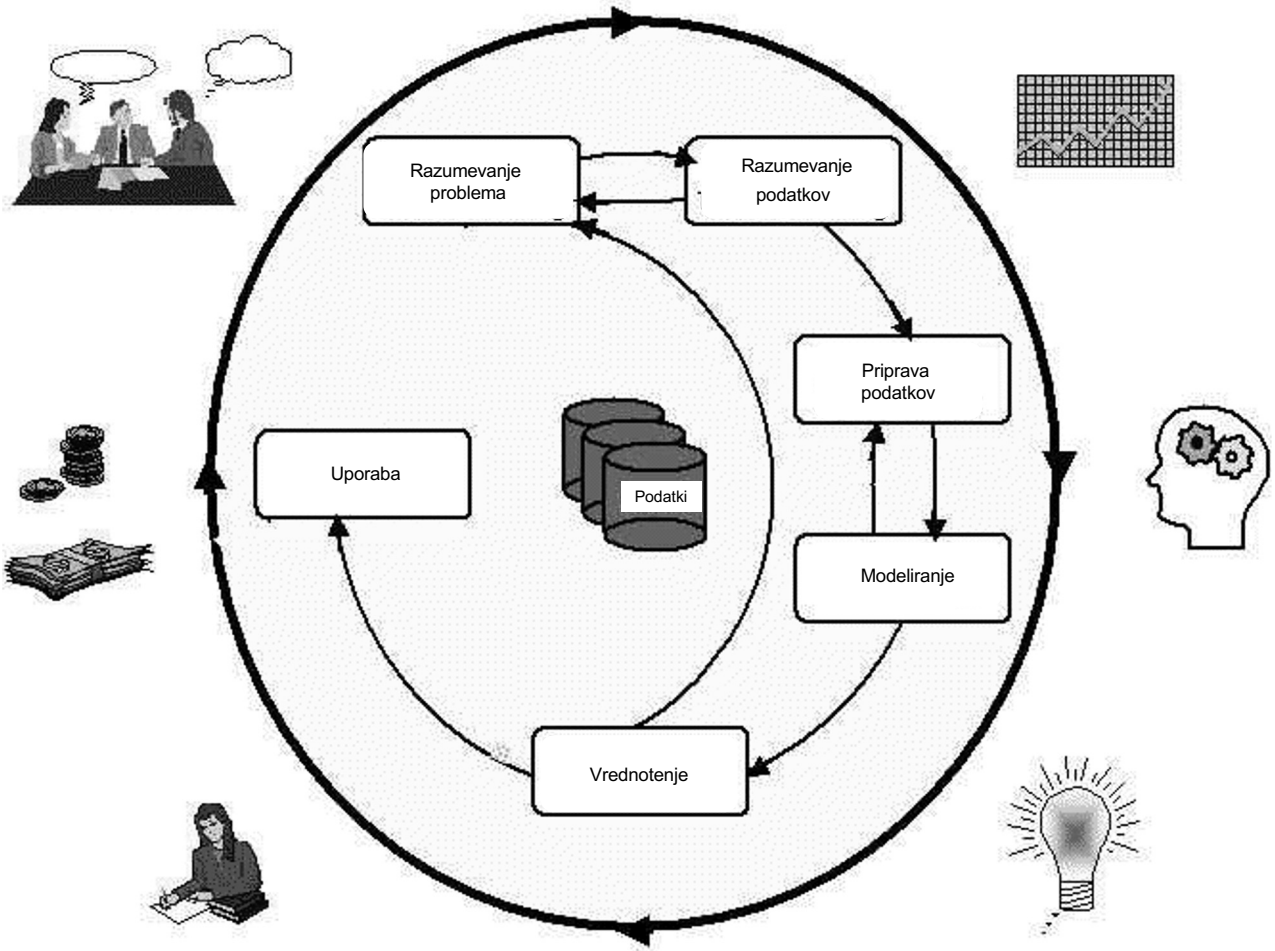
- CRoss Industry Standard Process for Data Mining
- Financiran s strani Evropske komisije
- Trenutno je v CRISP-DM interesnem združenju več kot 200 vodilnih podjetij z vsega sveta
- Zakaj standard?
  - Proces mora biti zanesljiv in ponovljiv; tudi za ljudi z omejenim poznavanjem podatkovnega rudarjenja
  - 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



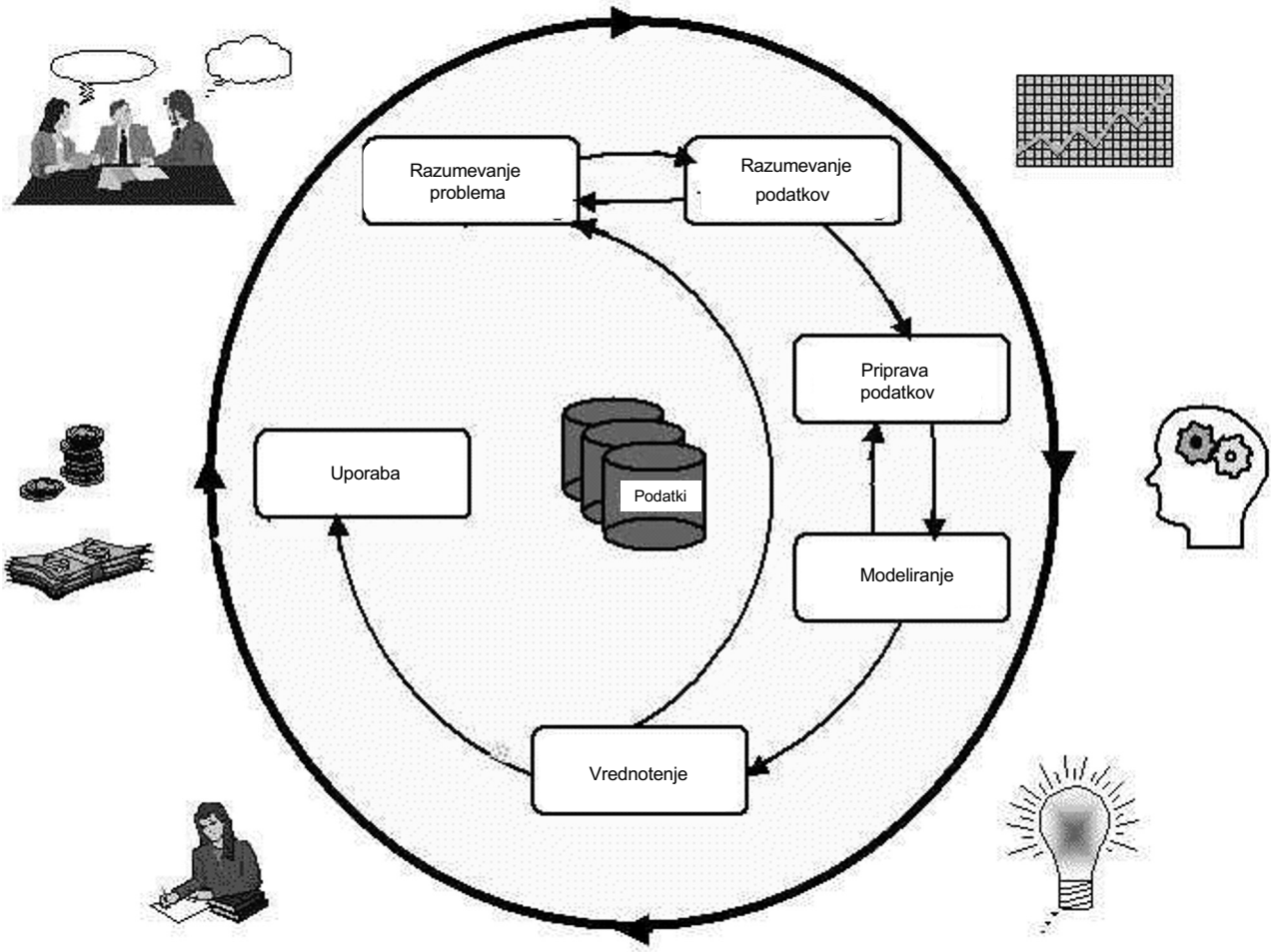
1. Razumevanje problema

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



2. Razumevanje podatkov

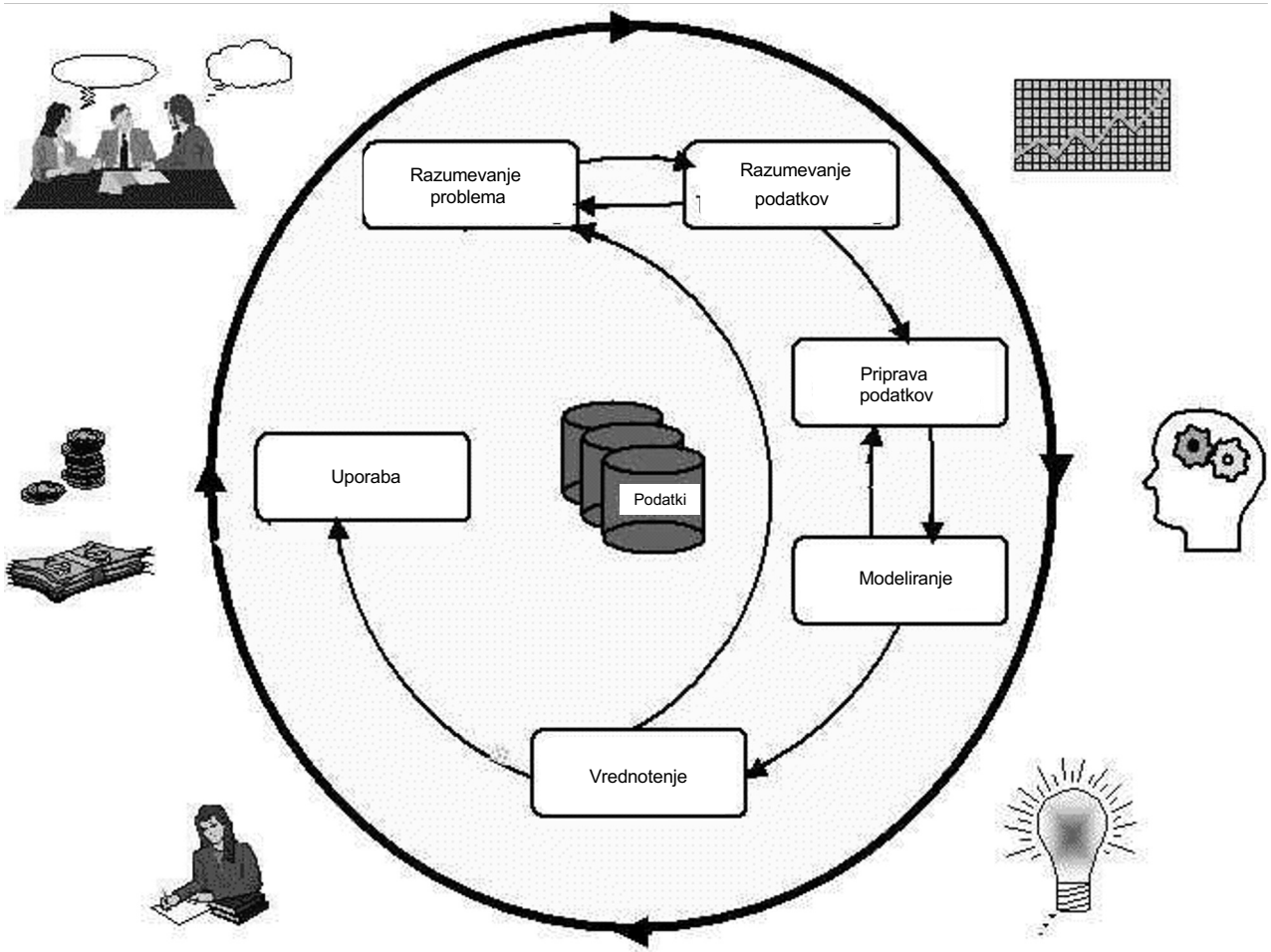
- spoznavanje podatkov
- preverjanje kvalitete
- iskanje izjem



3. Priprava podatkov

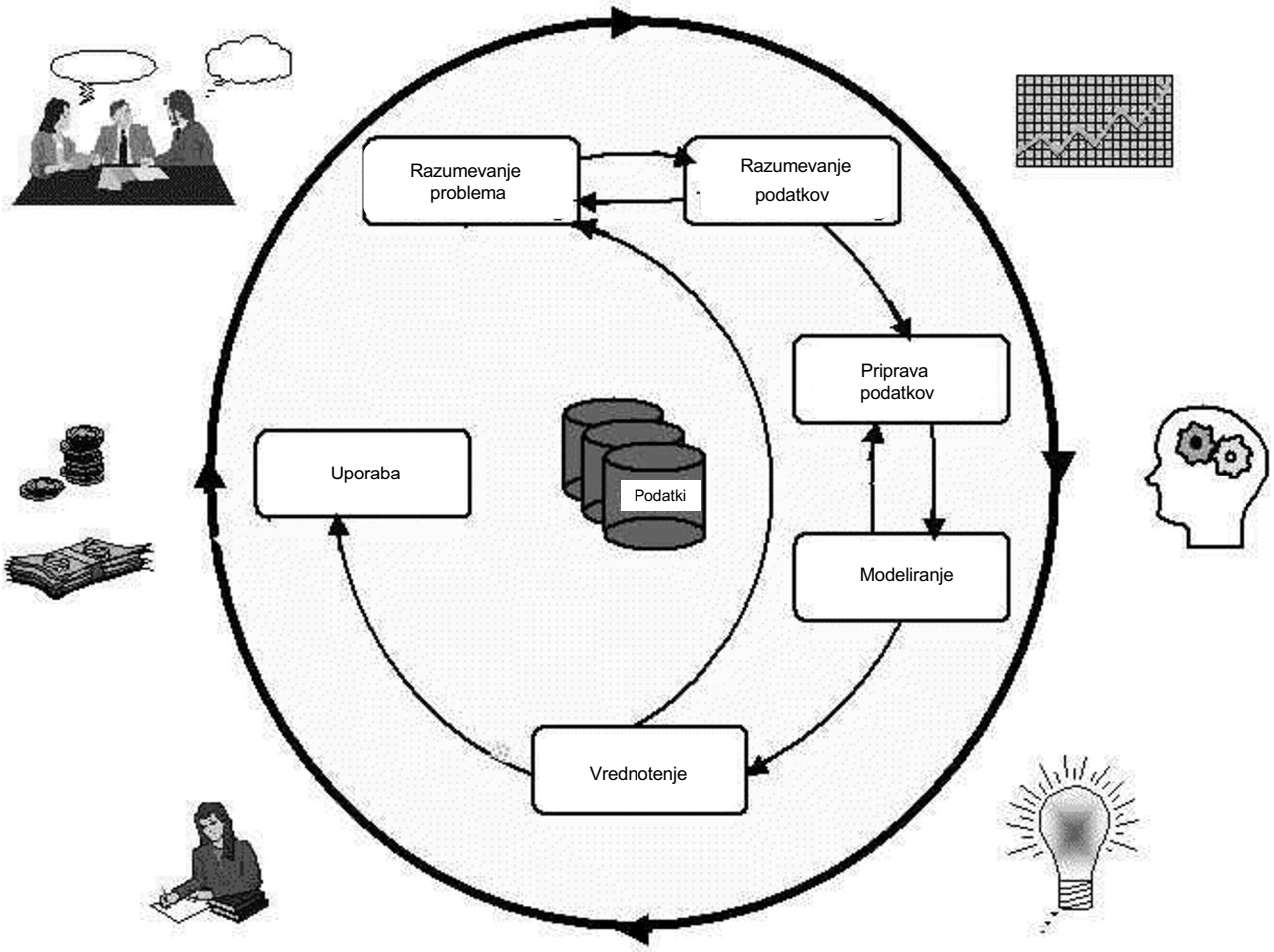
- Zbiranje
- Vrednotenje
- Poenotenje
- Čiščenje in filtriranje
- Transformiranje

- Priprava podatkov običajno vzame 90% celotnega časa



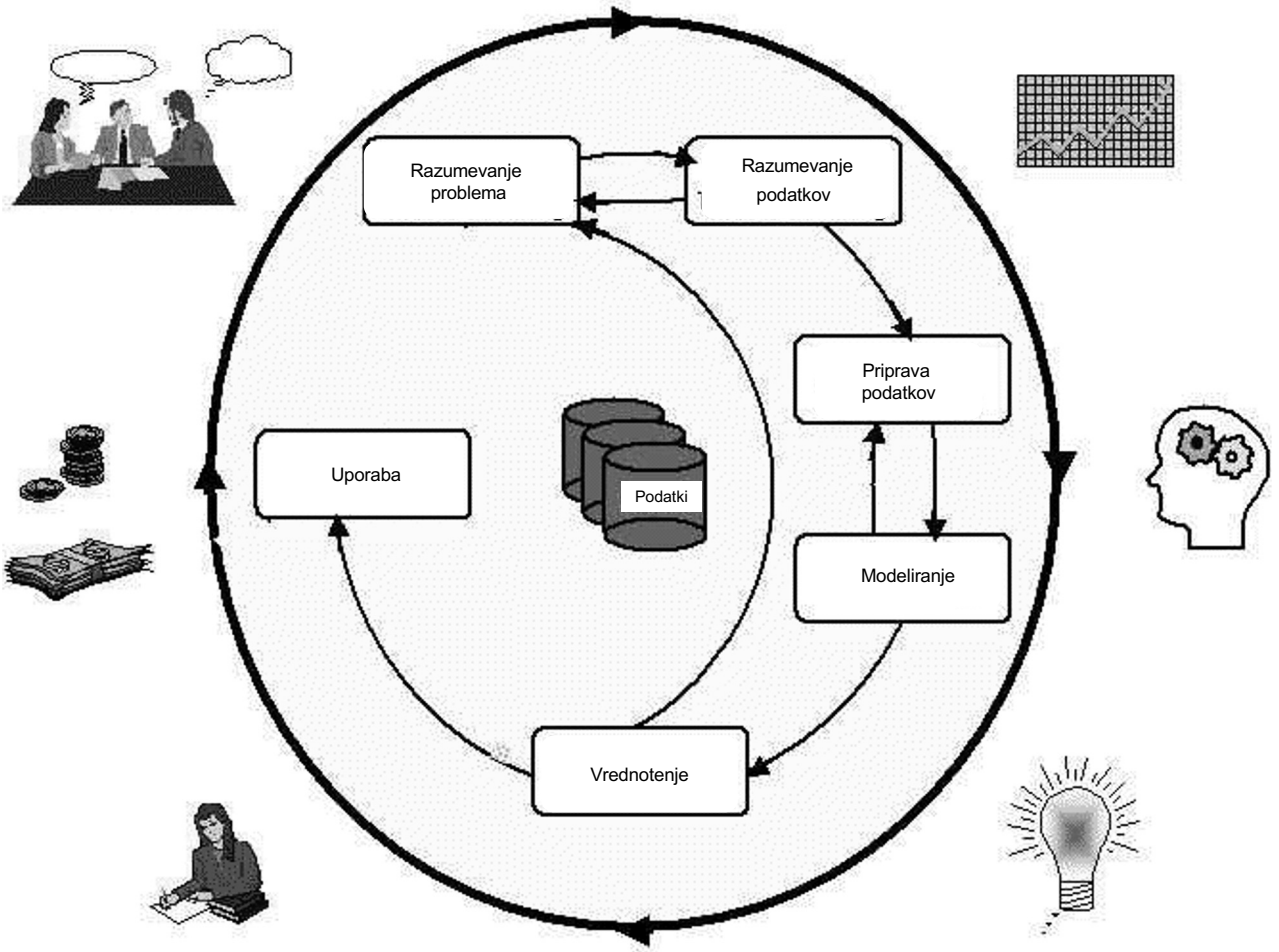
4. Modeliranje

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



5. Vrednotenje

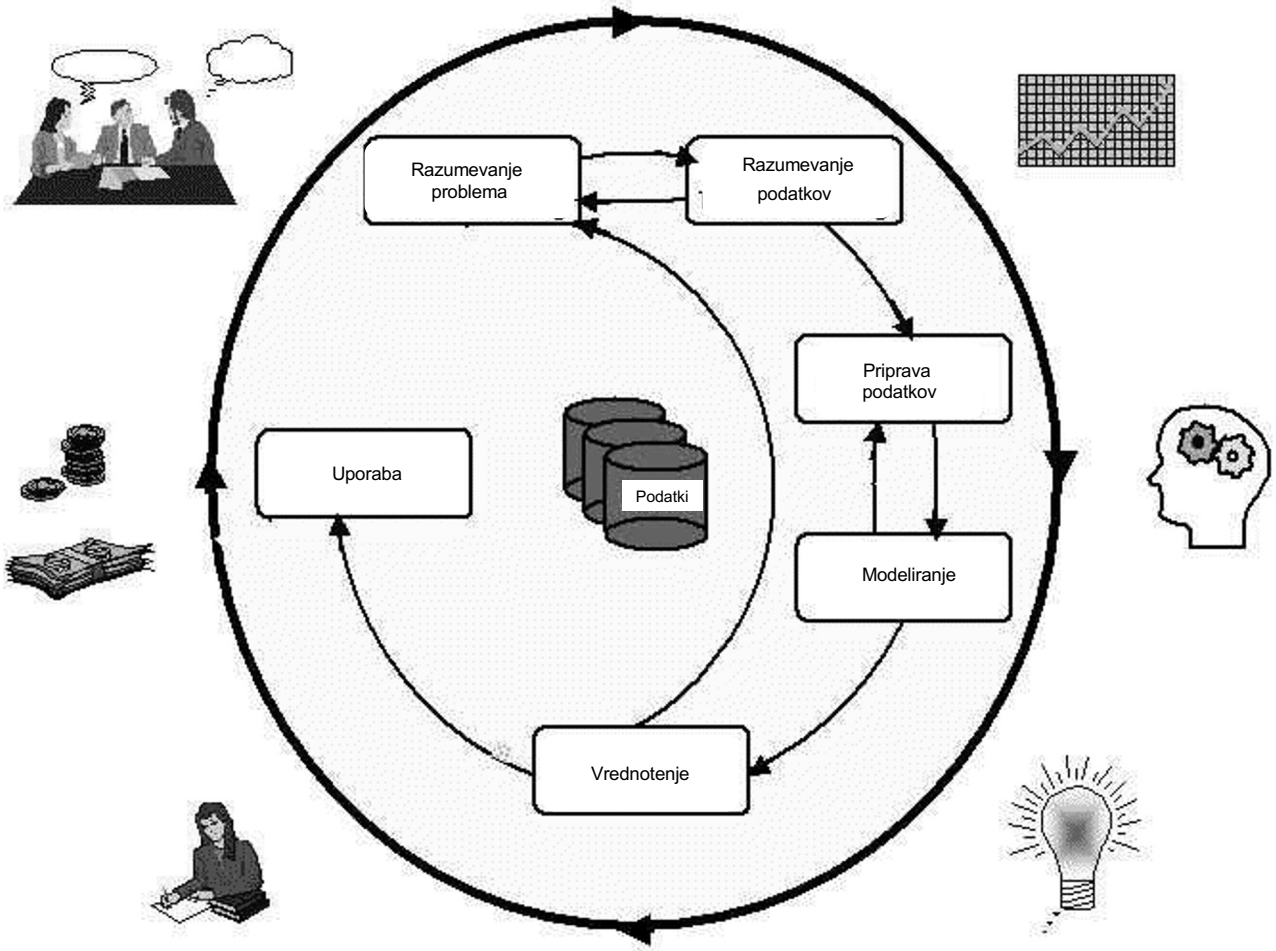
- Vrednotenje rezultatov DM glede na kriterije uspešnosti iz prve faze
- Sprejem konkretnih modelov
- Ocena procesa DM
- Odločitev o naslednjem koraku



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 (december 2025) standard 4.4, razvoj le počasi teče naprej
- Nekatere 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)
  - …

Standardiziran opis modelov - PMML

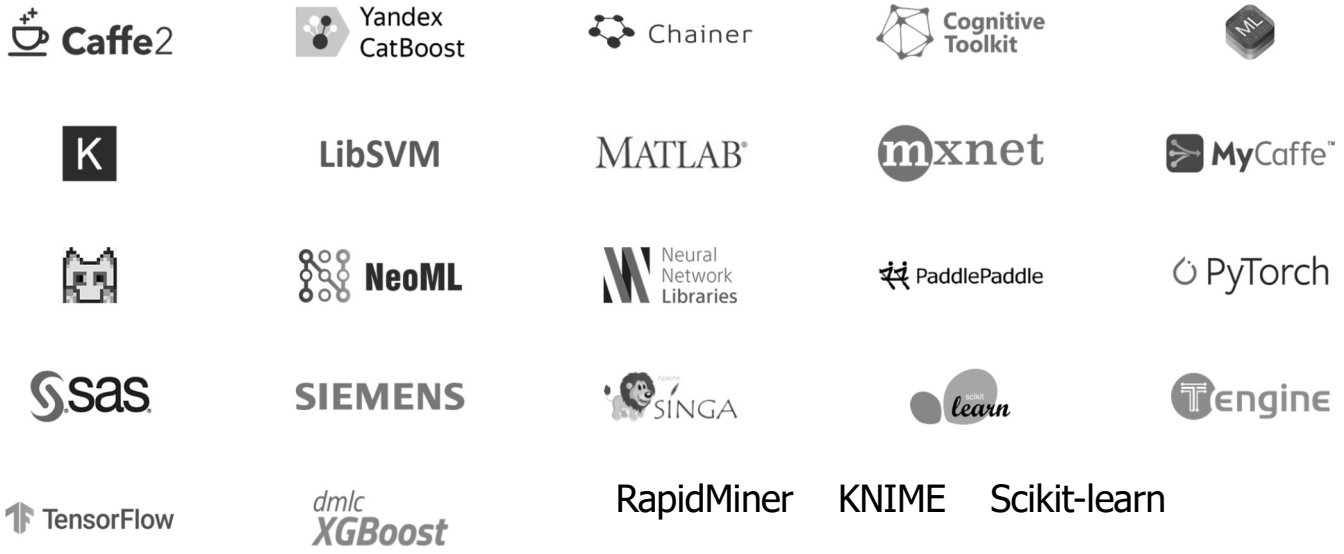
- Mnogi pomožni elementi (slovar podatkov, statistike, transformacije)
- Možne so razširitve obstoječih vrst modelov

- Kritika PMML
  - Razvoj se je skoraj ustavil (4.4 že več kot 3 leta)
  - Neprilagojenost na globoke modele
  - Zasnovan za klasično poslovno okolje,
  - XML (velikost!),
  - Slaba praktična sprejetost v modernih odprtokodnih orodjih

- Sodobni pristopi
  - "Vgrajena" binarna serializacija modelov (pickle, joblib, protobuf, ...)
  - ONNX kot standariziran binarni pristop (temelji na **proto**col **buf**fer tehnologiji)

Moderen standardiziran opis modelov - ONNX

- ONNX: open neural network exchange (december 2025: v1.20.0)
  - Predstavitev modelov v vmesni (grafni) predstavitvi
  - Široka podpora odprtokodnih orodij (ne samo nevronske mreže)
  - Odprtokoden izvor, podpora raznolikim programskim jezikom (Python, R, C/C++/C#, Java, JavaScript/TypeScript, Rust, Go, Swift, Julia, ...) – izvajanje, učenje modelov
  - Podpora in sprejetost v visokotehnoloških sodobnih podjetjih



RapidMiner KNIME Scikit-learn

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 omogočaju delo tudi z nestrukturiranimi, npr. tekstovno rudarjenje)
- Čimmanj manjkajočih (NULL) vrednosti
- Skrb za konsistentnost podatkov: podvajanje vrstic je lahko močno škodljivo

- Kaj pa nestrukturirani podatki? Slike, video, prosti tekst?

Primer podatkov

atributi (neodvisne spremenljivke)

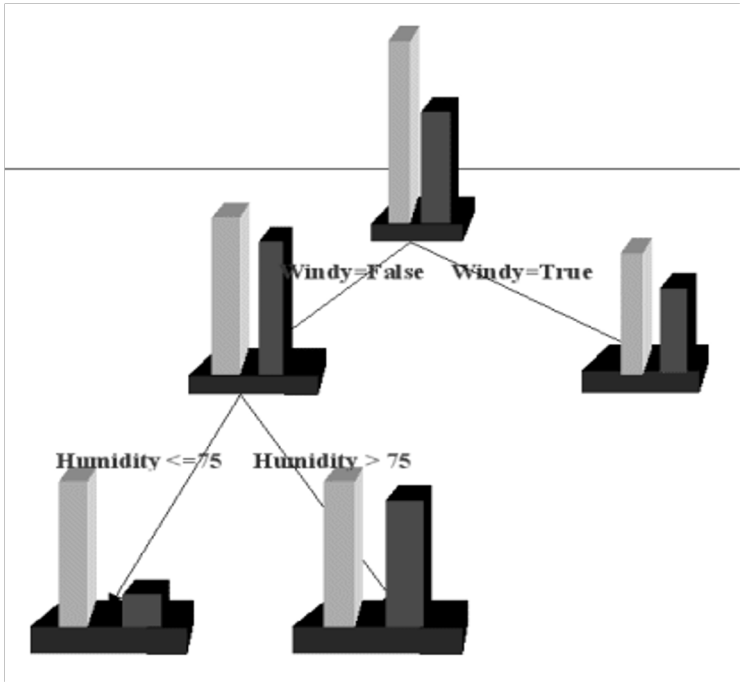
Day	Outlook	Temperature	Humidity	Wind	Play Golf?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
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
(odvisna spremenljivka)

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 Wind)
  - povezave, asociacije
- Kako lahko napovedujemo vrednost razrednega atributa (play golf): zgradimo napovedni model
  - nadzorovano učenje
- Model lahko uporabimo tudi za vpogled v strukturo problema
  - Razlaga delovanja modela; niso vsi tipi modelov primerni za to



☐ Yes ☐ No

Porazdelitev odgovorov na vprašanje, ali gremo igrat golf.

Skrb za kvaliteto podatkov (npr. podvajanje vrstic - stiki)

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)
  - Z orodji zgolj preveri hipotezo
  - Sprotno analitično procesiranje (*on-line analytical processing - OLAP*)
  - Statistične metode (raziskovalna in potrjevalna analiza podatkov)

- Induktivne (metoda postavi in preveri 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 lahko nastane pri veliki dimenzionalnosti podatkov
- Metode strojnega učenja
  - Avtomatsko postavljanje in preverjanje hipoteze
  - Učenje kot modeliranje podatkov
  - Prilagoditev za delo z velikimi množicami podatkov (klasične in "big data" metode)

Metode glede na način uporabe modela

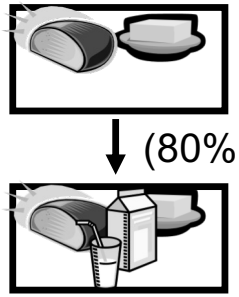
- **Napovedovalne**: model uporabimo za napovedovanje, kaj se bo zgodilo v prihodnosti ali z novimi podatki
- **Opisovalne**: model uporabimo za (človeku razumljiv) opis podatkov

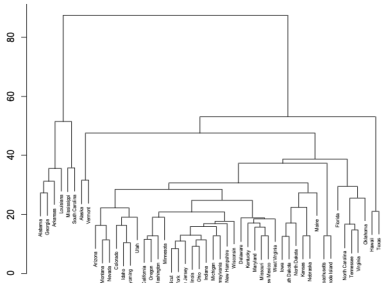
- Nekatere napovedovalne metode lahko uporabimo tudi kot opisovalne
- Posebej nas zanimajo induktivne metode, torej tiste, ki samostojno izdelajo na osnovi zbranih podatkov
- Področje umetne inteligence: strojno učenje (nadzorovano za napovedovalne metode, nenadzorovano za opisovalne metode)

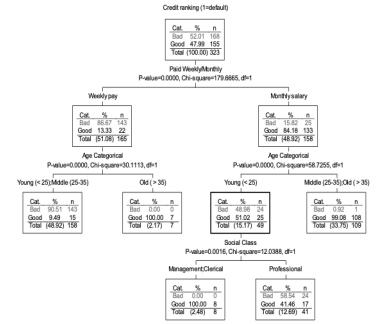
Podatkovno rudarjenje

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





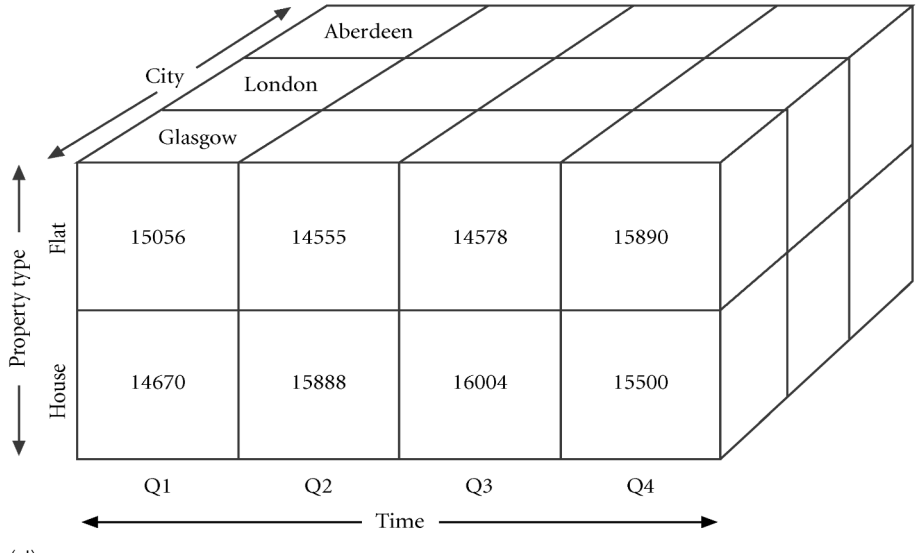


Opisovalno modeliranje (nenadzorovano učenje)

- Nimamo usmeritve (odvisne spremenljivke), 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 povezav oz. asociacij)
    - Temelji na odkrivanju povezav med podmnožicami atributov

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

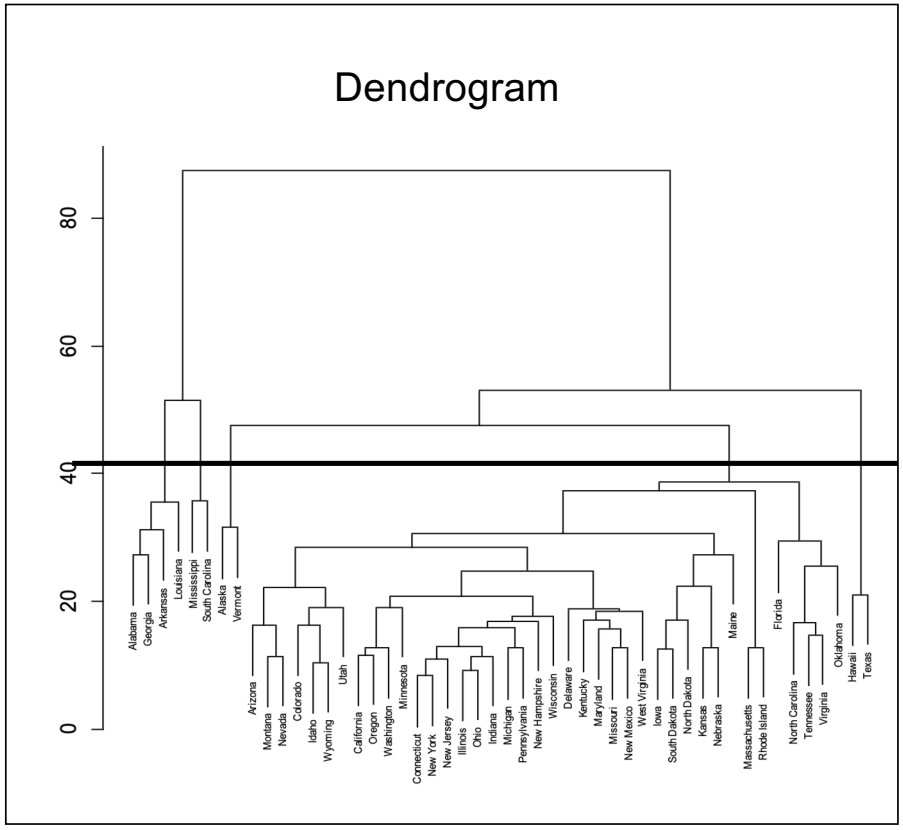
Property Type	City	Time	Total Revenue
Flat	Glasgow	Q1	15056
House	Glasgow	Q1	14670
Flat	Glasgow	Q2	14555
House	Glasgow	Q2	15888
Flat	Glasgow	Q3	14578
House	Glasgow	Q3	16004
Flat	Glasgow	Q4	15890
House	Glasgow	Q4	15500
Flat	London	Q1	19678
House	London	Q1	23877
Flat	London	Q2	19567
House	London	Q2	28677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



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 tudi specializirane implementacije (npr. *DBScan, Oracle O-Cluster; različice K-Means*)
- Primer: hierarhično gručenje

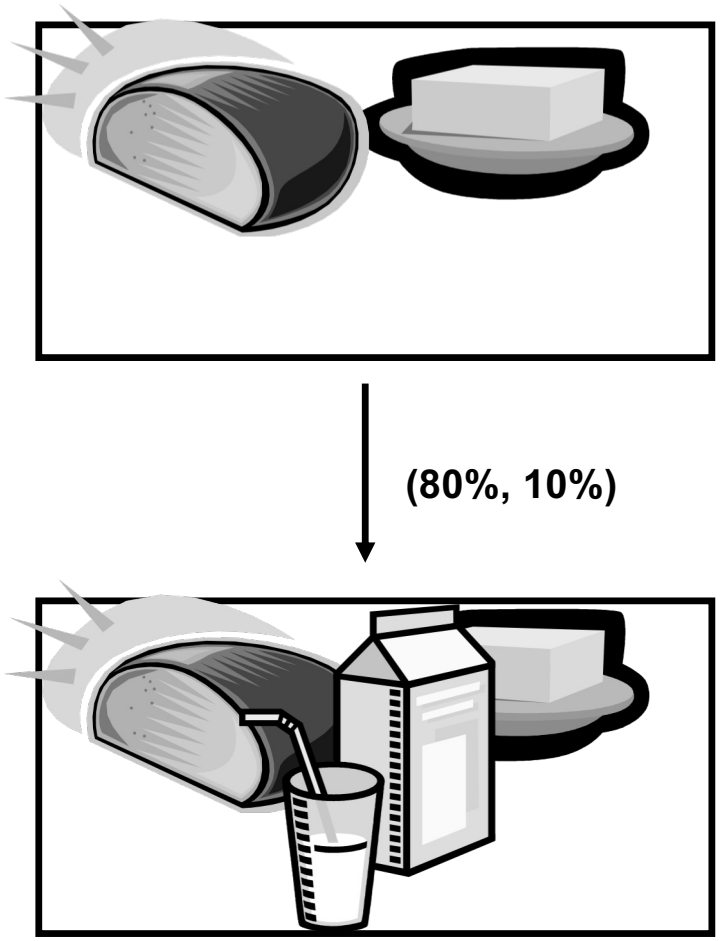


Povezovalna analiza

- 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 (N atributov - 2^N potencialnih podmnožic)
- Algoritem Apriori (Agrawal, 1995) naredi asociacijsko analizo praktično uporabno; moderni algoritmi so **FP-Growth**, Eclat, FIN, PrePost+

- Uporaba asociacijskih pravil



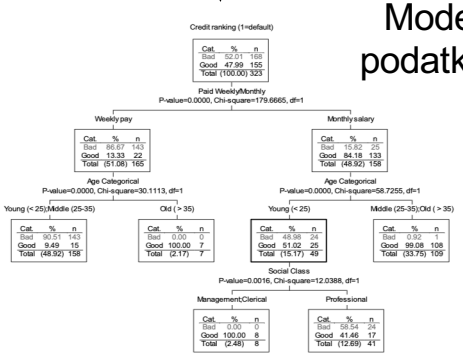
Napovedovalno modeliranje (nadzorovano učenje)

- 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



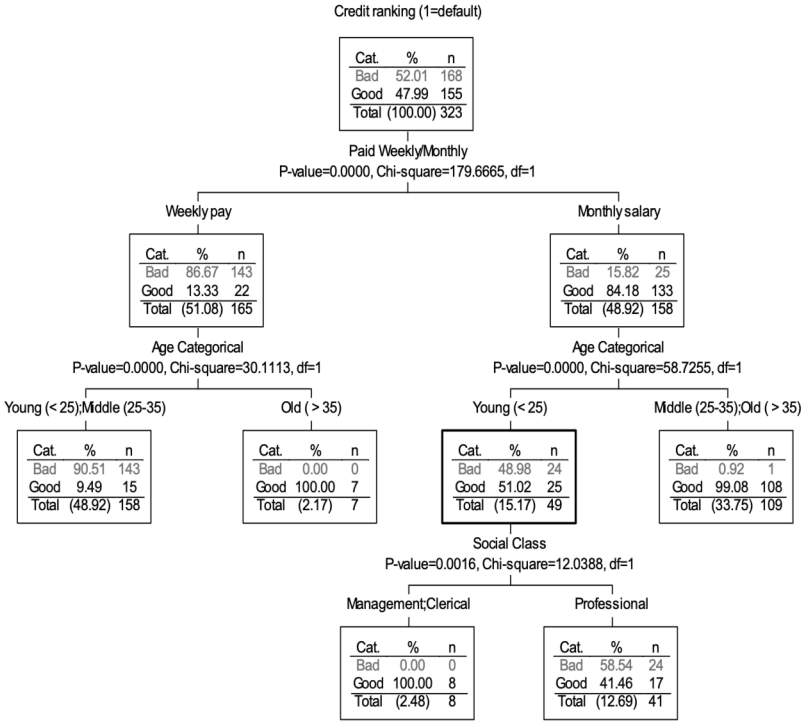
Napoved ne-znanega razreda

- 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

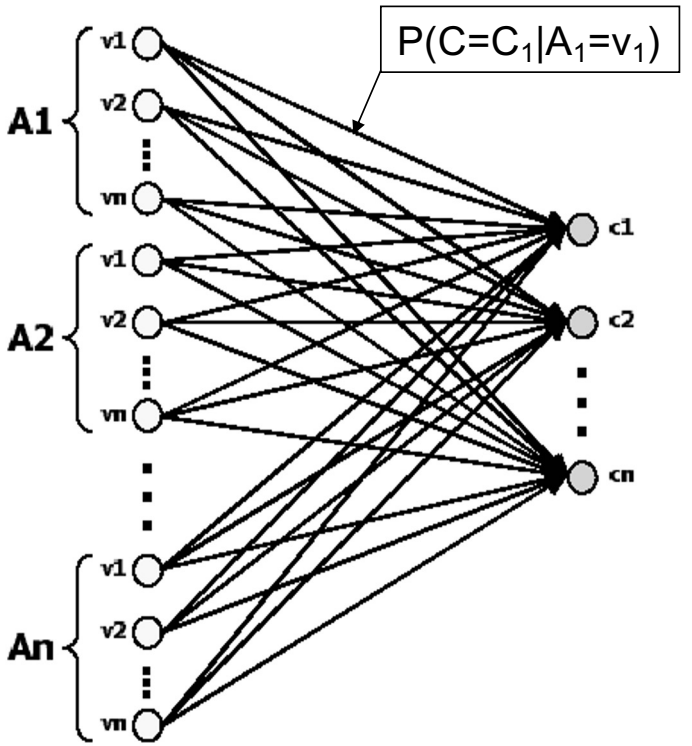
- Razmeroma hitra metoda, modeli so človeku razumljivi, obstajajo modifikacije algoritmov za gradnjo dreves na zelo velikih podatkovnih bazah (VLDT)
- Izboljšava: ansambli dreves
  - Random forest, **XGBoost**
  - Izguba razumljivosti



Naivni Bayesov klasifikator - izračun pogojnih verjetnosti $P(C=c|A_1=v_1, A_2=v_2, \ldots)$

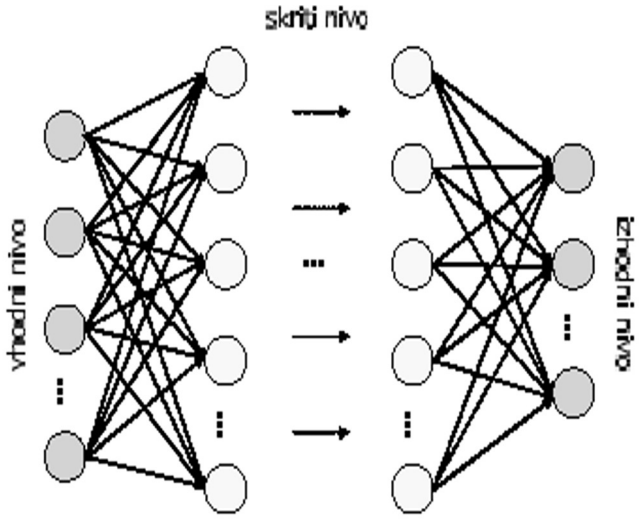
- Temelji na izračunu pogojne verjetnosti razreda pri podanih vrednostih atributov
- Naivnost: predpostavka pogojne neodvisnosti atributov
- Implementacija v obliki tabele pogojnih frekvenc (verjetnosti)
- Zelo enostavna, hitra in presenetljivo zmogljiva metoda
- Možnost človeku razumljive interpretacije 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
- Različne aktivacijske funkcije (npr. ReLU)
- 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, $\gg$ 2 skrita nivoja (tipično 10-100)



Množica drugih popularnih metod

- Metoda podpornih vektorjev (SVM)
- Naključni gozdovi (random forests)
- Gradientni boosting (XGBoost, CatBoost, LightGBM, ...)
- Splošne ansambelske metode (bagging, boosting, stacking)
- …

- Kaggle tekmovanja v podatkovnem rudarjenju:
  običajno je bolj kot izbor metode pomembno predprocesiranje podatkov in izračun/izpeljava novih, boljših atributov

Kaj potrebujemo za podatkovno rudarjenje?

- Podatki (najbolje iz podatkovnega skladišča/baze)
- Proces (CRISP-DM)
- Orodja (aplikacije)
- Kako dostopamo do podatkov
  - Direktno iz orodja (npr. z uporabo ODBC/JDBC/…)
  - Preko drugega orodja in pomnilnika (npr. Python/Pandas, R DataFrame)

- Koristno:
  - Prisotnost problemskega eksperta
  - Prisotnost eksperta za podatkovno rudarjenje

Razvrstitev orodij (aplikacij)

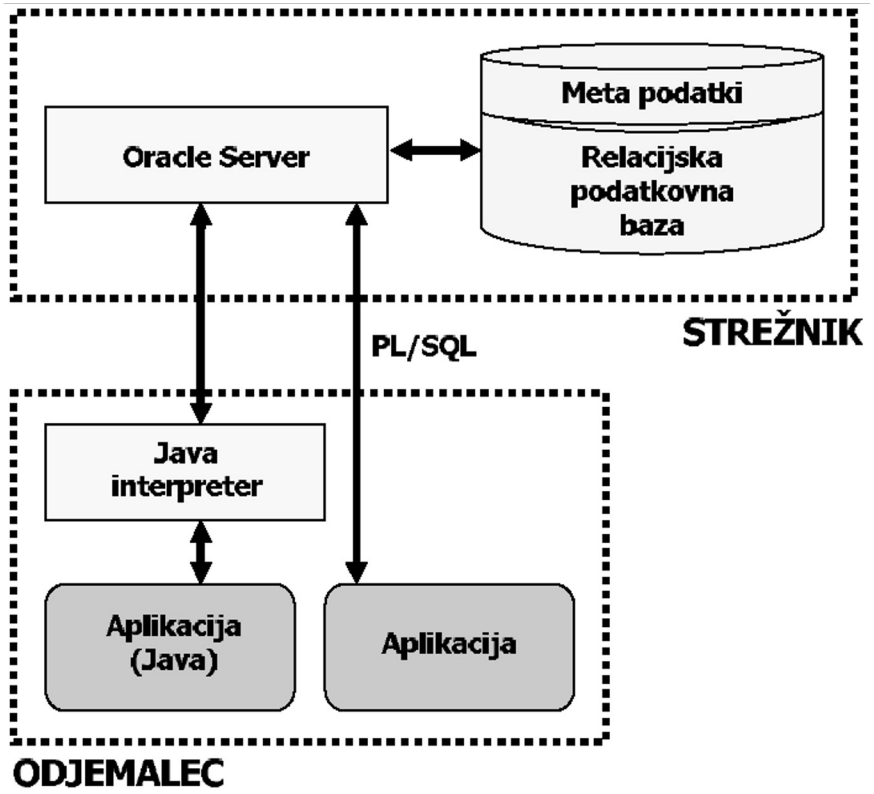
- Glede na namembnost
  - Raziskovalna
  - Komercialna
- Glede na dostop do podatkov
  - Tekstovne datoteke, drugi viri (API)
  - **Branje iz podatkovne baze**
- Glede na mesto rudarjenja
  - Lokalno (pri uporabniku)
  - Oddaljeno (znotraj podatkovnega strežnika)

Komercialna orodja

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

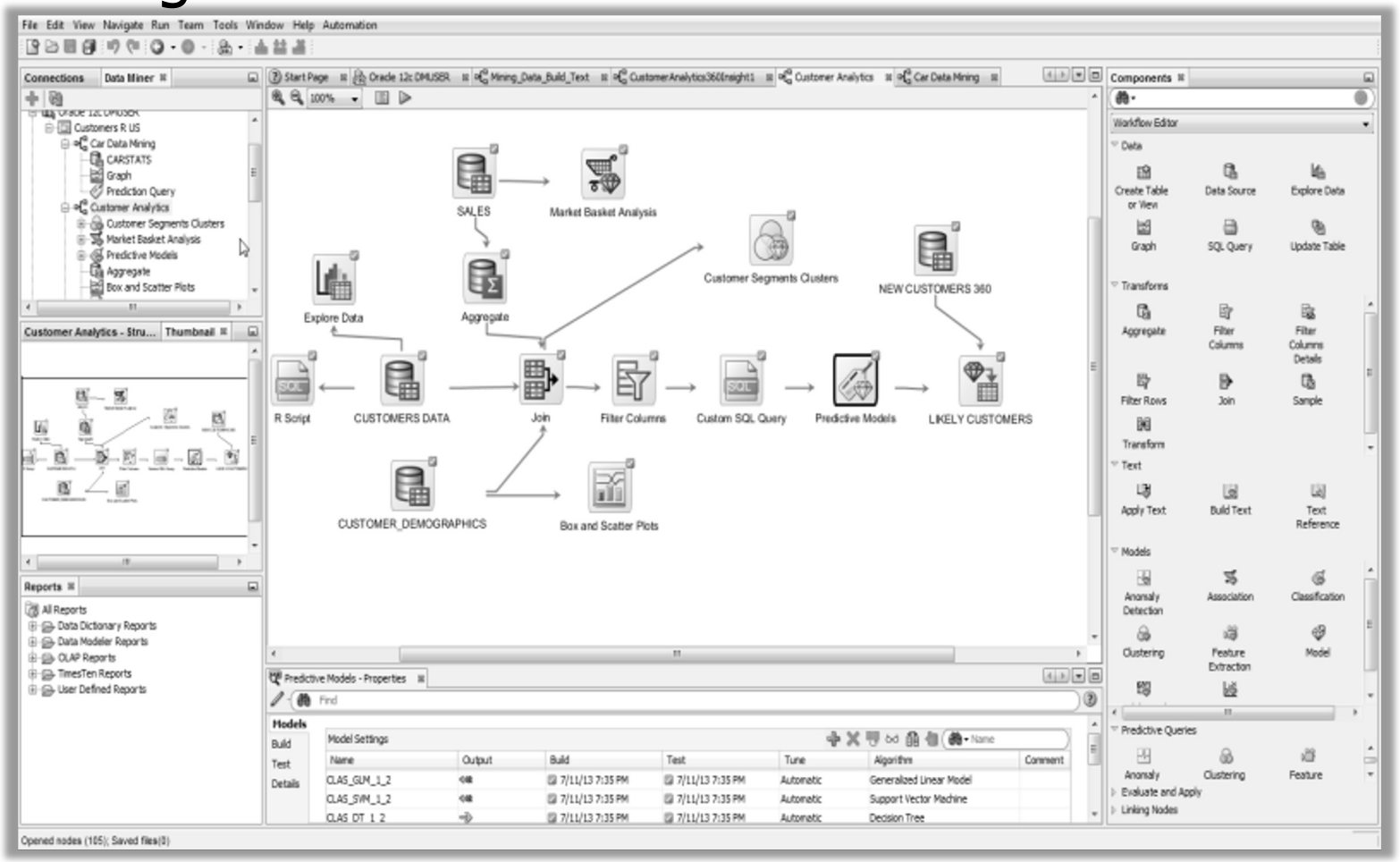
Oracle Data Mining – ODM

- Del Oracle Advanced Analytics (OAA)
- Vse analitske metode (OLAP, DM) se izvajajo na strežniku (PL/SQL, Java)
- 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
  - AI vektorsko iskanje (RAG, LLM)



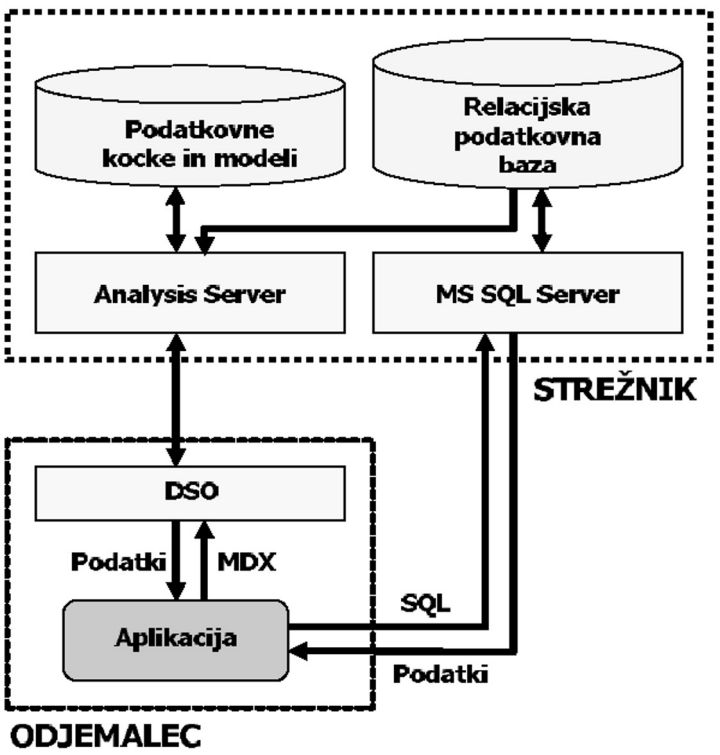
Oracle Data Mining – ODM

- Vizualna specifikacija toka opravil (workflow)
- Možna povezava z zunanjimi analitskimi orodji (npr. R)



Microsoft SQL server

- dodaten strežniški modul (Analysis services)
- celotno procesiranje na strežniku, kontrola preko SQL
- SQL Server 2000: skromna podpora strojnemu učenju (odločitvena drevesa, gručenje); MDX, OLAP
- SQL Server 2005/8 dodaja nove metode:
  - regresijska drevesa
  - naivni Bayesov klasifikator
  - povezovalna pravila
  - nevronske mreže
  - verzije do 2022 imajo le manjše dopolnitve
- SQL Server 2025:
  - semantično iskanje z vektorskimi vložitvami, integracija z OpenAI LLM
  - Github Copilot

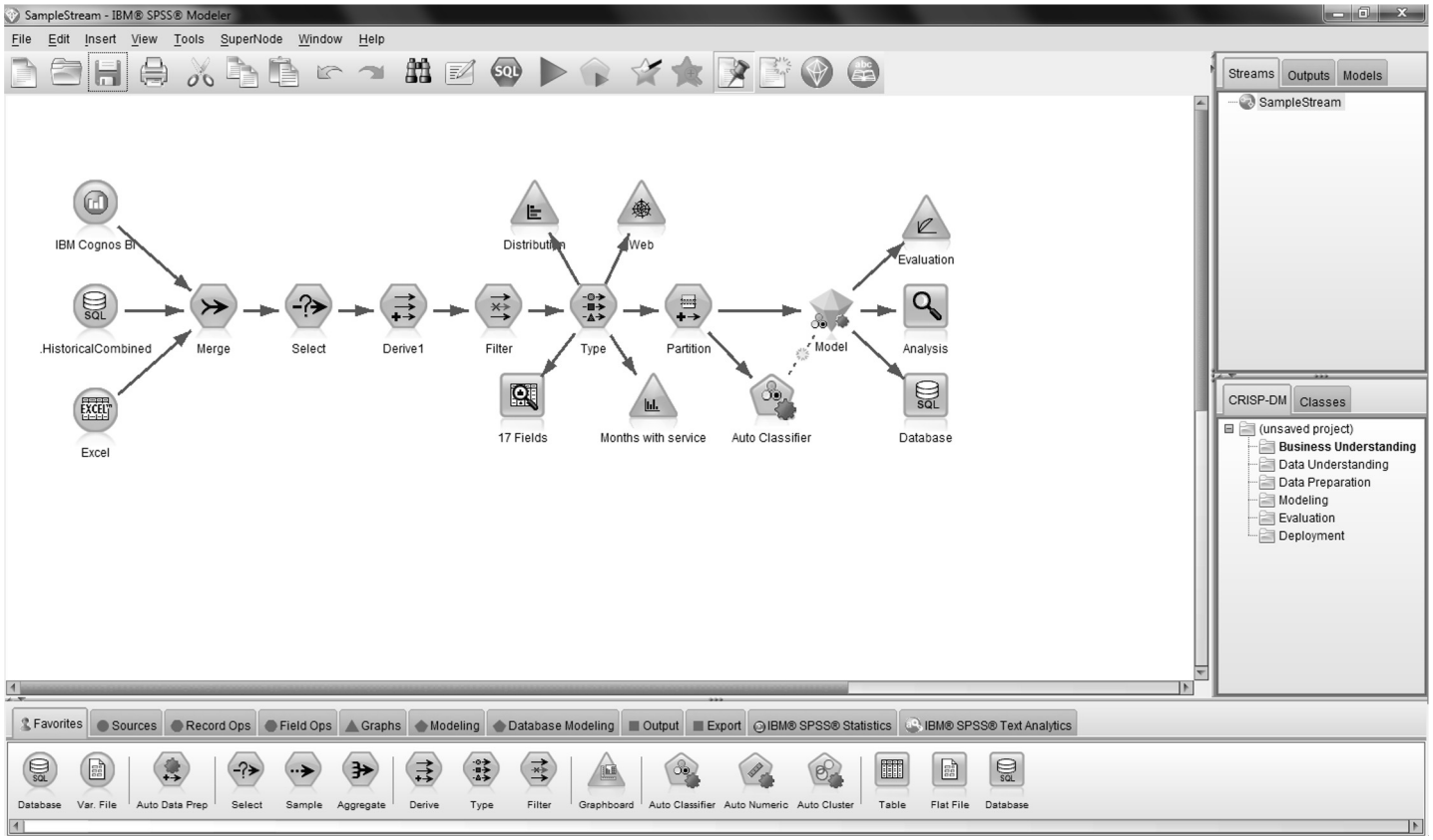


IBM SPSS Modeler

- 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
  - integracija z IBM Watson in Watsonx (LLM)

IBM SPSS Modeler

- Grafično načrtovanje podatkovnega rudarjenja



MySQL AI (plačljiv) - AutoML , GenAI

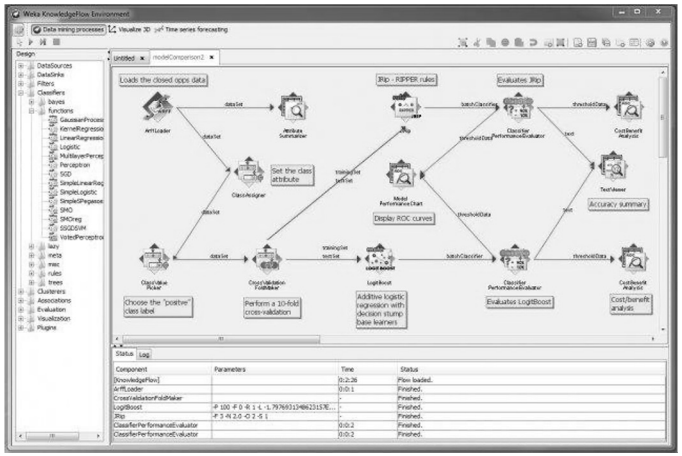
- Vgrajeno strojno učenje (in-database ML):
  - Učenje modelov in izvajanje napovedi neposredno v bazi nad podatki vseh tipov
  - Odpravlja potrebo po premikanju podatkov iz baze
- Generativna UI in vektorska shramba:
  - Vgrajena podpora za vektorsko shrambo in RAG (Retrieval-Augmented Generation)
  - Omogoča povezovanje lastnih podatkov z LLM-ji
- MySQL AutoML:
  - Avtomatska optimizacija metod strojnega učenja (optimizacija parametrov)
  - Bistveno izboljša zmogljivosti in zmanjša "začetno investicijo"
- Varnost in zasebnost:
  - Podatki ne zapustijo baze podatkov, kar zagotavlja višjo varnost pri obdelavi z AI

Raziskovalna orodja

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

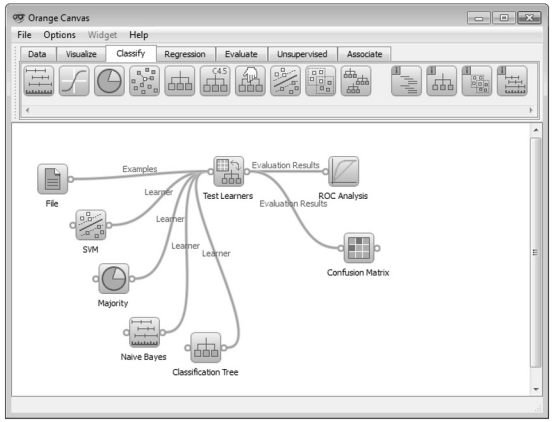
Weka

- Razvita na univerzi Waikato, Nova Zelandija (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, Scikit-Learn)
- 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



Koncept orodij AutoML

- Orodja samodejno prevzamejo naloge priprave podatkov, izbora algoritmov in optimizacije hiperparametrov
- Omogočajo razvijalcem in poslovnim analitikom, da gradijo visoko natančne modele brez poglobljenega znanja strojnega učenja
- Drastično skrajšajo čas razvoja modelov in omogočajo osredotočanje na kompleksne probleme namesto na rutinska opravila.
- Oblačne rešitve: Google Cloud Vertex AI, Microsoft Azure AutoML, AWS SageMaker Autopilot; **visoka skalabilnost**.
- Odprtokodne rešitve/knjižnice: H2O AutoML, Auto-sklearn in TPOT, ki omogočajo napredno modeliranje brez stroškov licenc.
- Vse bolj prisotna tudi v podatkovnih bazah in skladiščih

Primer: orodje MindsDB in AI tabele

- MindsDB uvaja koncept *AI Table* – virtualne tabele, po kateri je mogoče poizvedovati in ki predstavlja polno naučen model strojnega učenja.
- Omogoča dostop do napredne analitike prek znanega vmesnika SQL.
- Uporabnik poizveduje po virtualni tabeli s standardnim stavkom SELECT
- Sklop WHERE se uporablja za podajanje vhodnih atributov za model.
- Model se izvede v realnem času in vrne svojo napoved, kot da bi bila stolpec v tabeli.
- Ključna prednost: kdor pozna SQL, lahko gradi in uporablja modele ML brez potrebe po orodjih, kot je Python, ali specializiranih ogrodjih za ML.
- Celoten proces je interno podprt z vgrajenim AutoML pogonom Lightwood
- MindsDB je **neodvisen** od SUPB (zunanje orodje, ki podpira SQL)

Strojno učenje v MindsDB s tremi SQL ukazih

1. korak: povezava z virom podatkov: CREATE DATABASE za vzpostavitev poimenovane povezave z obstoječim virom podatkov, npr. povezave na bazo MySQL.
   ```
   CREATE DATABASE rentals_db WITH ENGINE = 'mysql', ...;
   ```
2. korak: učenje modela: ukaz CREATE MODEL za učenje modela (AI table) na podatkih (SELECT …) iz povezanega vira, pri čemer določite stolpec za napovedovanje. Proces podpira AutoML!
   ```
   CREATE MODEL mindsdb.home_rentals_pred FROM
   rentals_db (SELECT ...) PREDICT rental_price;
   ```
3. korak: izvedba napovedi s preprosto poizvedbo SELECT na AI tabeli, pri čemer v sklopu WHERE navedemo vhodne attribute
   ```
   SELECT rental_price FROM mindsdb.home_rentals_pred
   WHERE num_of_rooms = 3 AND …; -- vrednosti atributov
   ```

Oracle Data Mining: praktična izkušnja izpred nekaj let

- 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; minimalna poraba pomnilnika	Nekaj sekund; velika poraba pomnilnika	Zmanjkalo pomnilnika (JVM default heap), sicer pod sekundo