**Data mining** je niz aktivnosti čiji je cilj ekstrakcija ili otkrivanje struktura i zakonitosti u podacima. Kombinacijom tehnika iz područja strojnog učenja, statističke analize, tehnika modeliranja i tehnologijom baza podataka, inteligentnom analizom podataka pronalazimo strukturu i odnose između podataka, te izvodimo pravila i modele koji omogućavaju predviđanje i odlučivanje u novim (budućim) situacijama, odnosno primjerima.

Tipične primjene se osnose najviše na ekonomsko/gospodarsko okruženje – segmentiranje tržišta za marketing, profiliranje kupaca, otkrivanje prijevara, analiza rizika i sl.

Pod učenjem u području inteligentne analize podataka podrazumijeva se proces stvaranja modela na bazi dostupnih podataka.

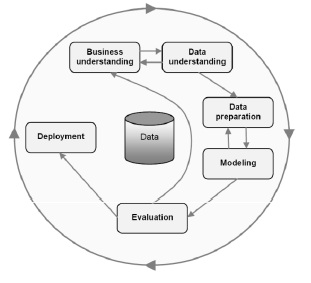
Data mining algoritmi:

* Klasifikacija
* Asocijacija
* Analiza sekvenci
* Analiza devijacija
* Predviđanja

**CRISP-DM metodologija**

Dijeli se na 6 dijelova:

* **Razumijevanje poslovanja**: ova inicijalna faza se fokusira na razumijevanje ciljeva i zahtjeva projekta iz poslovne perspektive, onda pretvaranje ovog znanja u definiranje problema rudarenja podataka, i preliminarni plan napravljane radi zadovoljavanja ciljeva.
* **Razumijevanje podataka**: Ova faza počinje sa početnim skupom podataka i nastavlja s aktivnostima da bi se upoznali s podacima, da bi definirali probleme kvalitete podataka, da bi se otkrili prvi uvidi u podatke ili da bi se detektirali zanimljivi podskupovi za formiranje hipoteze za skrivene informacije
* **Pripremanje podataka**: Ova faza pokriva sve aktivnosti stvaranja finalnog skupa podataka (podataka koje će se koristiti u modelirajućim alatima) iz početnih sirovih podataka. Zadaci pripremanja podataka se često izvode nekoliko puta i nemaju specificiran redoslijed. Zadaci uključuju odabir tablica, zapisa i atributa, kao i transformaciju i pročišćavanje podataka za modelirajuće alate.
* **Modeliranje**: Odabiru se i primjenjuju različite modelirajuće tehnike, a njihovi parametri se kalibriraju do optimalnim vrijednosti. Tipično, postoji nekoliko tehnika za isti tip problema rudarenja podataka. Neke tehnike imaju posebne zahtjeve vezane uz oblik podataka, i zbog toga se često nanovo radi faza pripreme podataka.
* **Evaluacija**: U ovome trenutku imamo izrađeni model koji izgleda visoko kvalitetan iz perspektive podatkovne analize. Prije razvoja modela, važno je podrobnije evaluirati modeli i pregledati izvršene korake radi stvaranja modela koji će sigurno izvršiti poslovne ciljeve. Ključni cilj je odrediti postoje li određeni poslovni problemi koji se nisu dovoljno dobro razmotrili. Na kraju ove faze, znamo kolika je važnost rezultata rudarenja podataka.
* **Razvoj**: Stvaranje modela općenito nije kraj projekta. Iako je svrha modela povećati znanje o podacima, prikupljeno znanje se treba organizirati i prezentirati u obliku koji će korisnik moći koristiti. Ovisno o zahtjevima, faza razvoja može biti jednostavna kao generiranje izvještaja ili kompleksna kao implementiranje ponavljajućeg procesa rudarenja podacima na razini tvrtke. U dosta slučajeva korisnik, a ne podatkovi analitičar, izvodi razvojne korake. Korisnik mora znati koje akcije se trebaju napraviti da bi stvarno mogao iskoristiti stvoreni model.



Sada treba objasniti ciljeve i izlaze svakog od ovih koraka (-.-)

**Razumijevanje poslovanja**

**Cilj: odrediti poslovne ciljeve**

Prvi cilje analitičara podataka je podrobno razumijeti, iz poslovne perspektive, što klijent želi postiči. Često klijent ima dosta ciljeva i ograničenja koja se moraju propisno izbalansirati. Cilj analitičara je otkriti važne faktore, na početku, koji mogu utjecati na ishod projekta. Moguća posljedica zanemarivanja ovog koraka je trošenje velikog truda na stvaranje pravih odgovora na kriva pitanja.

**Izlaz: Pozadina**

Treba zapisati informacije o poslovnoj situaciji na početku projekta.

**Razumijevanje podataka**

**Cilj: skupiti inicijalne podatke**

Treba skupiti podatke izlistane u resursima projekta. Inicijalna kolekcija uključuje punjenje podacima ako su bitni za razumijevanje podataka. Npr. ako se koristi određeni alat za razumijevanje podataka, normalno je da treba napuniti (load) podacima ovaj alat. Ovo može voditi do početnih koraka pripremanja podataka.

**Izlaz: izvješće o početnoj kolekciji podataka**

Lista podatkovnih skupova, zajedno s njihovim lokacijama unutar projekta, metode koje se koriste za njihovo stjecanje i problemi na koje smo naišli. Treba zapisati probleme i nađena rješenja da bi pomogli u budućnosti oko sličnih projekata.

**Priprema podataka**

**CRISP-DM modeliranje**

**Cilj: odabrati tehniku modeliranja**

Kao prvi korak u modeliranju, treba odabrati tehniku modeliranja koja će se koristiti. Ovaj korak se odnosti na specifičnu tehniku modeliranja, npr. stablo odluke sa C4.5 ili generiranje neuralne mreže sa *back propagation*. Ako se primjeni više tehnika, treba napraviti ovaj zadatak za svaku tehniku zasebno.

**Izlaz: tehnika modeliranja**

Treba dokumentirati tehniku modeliranja koja će se koristiti.

**CRISP-DM evaluacija**

**Cilj: evaluirati rezultate**

Ovaj korak procjenjuje koliko model zadovoljava poslovne ciljece i procjenjuje postoji li koji poslovni razlog zbog kojeg bi ovaj model bio manjkav. Druga opcija evaluacija je testirati model sa testnim podacima u stvarnoj aplikaciji ako vremenska i ograničenaj budžeta dopuštaju. Evaluacija također procjenjuje ostale rezultate koje je generiralo rudarenje podacima. Rezultati pokrivaju modele koji su nužno povezani sa izvornim poslovnim ciljevima i sve ostale nalaze koji nisu nužno povezani s izvornim ciljevima poslovanja, ali koji isto mogli otkriti dodatne izazove, informacije ili naputke za budućnost.

**Izlaz: procjena rezultata rudarenja podacima u odnosu na kriterij poslovnog uspjeha, odobreni modeli**

Treba rezimirati rezultate procjene u smislu uspješnosti poslovanja, uključujući finalnu izjavu o tome ispunjava li projekt početne poslovne ciljeve. Nakon procjene modela, modeli koji ispunjavaju odabrane kriterije postaju odobreni modeli.

**CRISP-DM razvoj**

**Cilj: planiranje razvoja**

Da bi se rezultati rudenja mogli razviti u poslovanje, ovaj korak uzima rezultate evaluacije i odabire strategiju za razvoj.

**Izlaz: Plan razvoja**

Rezimirati strategiju razvoja uključujući potebne korake, i kako ih izvesti.

Algoritmi rudarenja podacima se mogu svrstati u 6 osnovnih kategorija:

* Klasifikacija
* Asocijacija
* Regresija
* Predviđanje
* Analiza sekvence
* Analiza devijacije

**Pravila asocijacije**

Najčešće se koriste u maloprodajnim centrima-

Daje rezultate u obliku:

* AKO a, ONDA b uz postotak vjerojatnosti
* AKO b i c, ONDA a (vjerojatnost) ili
* AKO b i NE c, ONDA a (vjerojatnost)...

Koji proizvodi se kupuju zajedno? Kao ih grupirati i povećati prodaju nekog artikla?

Ideja ove tehnike jest uočiti veze, tj. asocijacije među specifičnim vrijednostima određenih definiranih varijabli (elemenata) u velikim skupovima podataka. Drugim riječima, potrebno je otkriti pravilnosti koje se temelje na odnosu specifičnog podatka (elementa) s drugim podatcima (elementima) koji se pojavljuju u istom jedinstvenom slijedu operacija (transakcija).

Ako se npr. sakupljaju podatci o kupljenim knjigama na kasi neke velike knjižare, transakcija svake mušterije se bilježi u bazi podataka, a zabilježena transakcija se sastoji od knjižnih naslova kupljenih od strane pojedine mušterije, novina i sl.

Stoga svaki zapis u bazi predstavlja jednog kupca (transakciju) i može se sastojati od jedinstvenog naslova kojeg je kupio taj kupac, ili se može sastojati od mnoštva različitih stavki koje su kupljene.

Kupljene stavke se unose nasumično, onako kako su registrirani prilikom uzimanja sa pokretne vrpce na blagajni.

Svrha analize jest pronaći asocijacije između kupljenih stavki, tj. izvesti asocijacijska pravila koja indentificiraju stavke (elemente) i učestalost pojava različitih stavki (elemenata), koji se pojavljuju s najvećom (zajedničkom) frekvencijom.

Želimo doznati koje će knjige najvjerojatnije biti kupljene od strane kupca za kojeg znamo da je već kupio ili će kupiti određenu knjigu.

Postojanje ovakve informacije omogućuje prodavačima u knjižari da unaprijed preporuče mušteriji naslove koji bi ga mogli interesirati.

Standardna metodologija podrazumijeva analizu potrošačke košarice za što se najčešće koristi „Apriori“ algoritam. Potrošačka košarica predstavlja skup elemenata koje je kupac kupio u jednoj

transakciji. Transakcija je dobro definirana poslovna aktivnost. Uobičajena analiza provodi se u transakcijskoj bazi podataka, da bi se pronašli skupovi elemenata koji se pojavljuju zajedno u velikom broju transakcija. Skup elemenata naziva se grupa. Grupa koja sadrži „k“ elemenata, naziva se k-ta grupa, a „k“ je duljina grupe. Postotak transakcija koje sadrže određenu grupu naziva se potpora te grupe. Da bi grupa bila zanimljiva za analizu, njena potpora mora biti veća od minimalne koju definira korisnik. Za takve grupe se kaže da su učestale.

Iz baze prodajnih transakcija, želimo otkriti važne poveznice (asocijacije) među elementima i to takve da će postojanje nekih elemenata u transakciji upućivati na postojanje drugih određenih elemenata u istoj transakciji.

Neka je I = {i1, i2, …, im}, skup elemenata. Neka je baza podataka DB (data base) skup transakcija, gdje je svaka transakcija T skup elemenata takvih da vrijedi da je T ⊆ I.

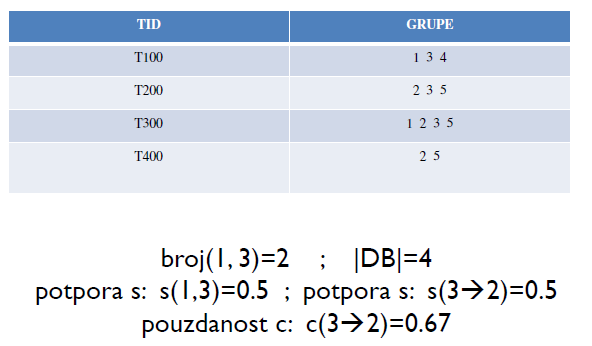
Količina istovrsnih elemenata koji su kupljeni u pojedinoj transakciji se ne razmatra, što znači da je svaki element binarna varijabla koja nam govori je li element kupljen ili ne.

Svaka transakcija ima transakcijsku oznaku (TID – transaction indentifier).

Neka je X skup elemenata. Za transakciju T se kaže da sadrži X ako i samo ako vrijedi da je X ⊆ T. Asocijacijsko pravilo podrazumijeva oblik X ⇒ Y, ako je X slijedi Y, gdje je X ⊆ I, Y ⊆ I, i X ∩ Y = 0. Dvije su temeljne mjere kojima se određuje „vrijednost“ asocijacijskog pravila: **potpora (s) i pouzdanost (c).** Pošto su transakcijske baze podataka uobičajeno velike, korisnike interesiraju samo oni elementi koji se kupuju s velikom učestalošću (frekvencijom), stoga se unaprijed definiraju pragovi potpore i pouzdanost u postotcima, da bi se odbacila ona asocijacijska pravila koja nisu korisna ili interesantna.

**Potpora (s) asocijacijskog pravila** se definira kao **postotni omjer onih zapisa koji sadrže X U Y prema ukupnom broju zapisa u bazi podataka. s=100 x [broj(X U Y) / |DB|].**

**Pouzdanost asocijacijskog pravila (c)** se definira kao **postotni omjer broja transakcija koje sadrže X U Y prema ukupnom broju zapisa u bazi podataka koje sadrže X. c=100 x [broj(X U Y) /broj(X)].**



**Pravila asocijacije**:

Pouzdanost je mjera „snage“ implikacije X ⇒ Y.

Potpora ukazuje na učestalost uzorka koji se pojavljuje u pravilu X ⇒ Y.

Npr. ako je pouzdanost asocijacijskog pravila X ⇒ Y c=80%, to znači da 80% transakcija koje sadrže X također sadrže i Y.

Naravno da je poželjno obratiti pažnju na samo ona pravila koja imaju razumno veliku potporu. Pravila X ⇒ Y koja imaju veliku pouzdanost i potporu nazivaju se jaka pravila, pa je zadatak rudarenja za asocijacijskim pravilima otkriti jaka asocijacijska pravila u velikim bazama podataka.

Iz svega navedenog slijedi da se rudarenje za asocijacijskim pravilima može rastaviti u dvije faze:

1. otkriti velike grupe, tj. skupove elemenata koji imaju transakcijsku potporu (s) iznad od

korisnika određenog minimuma;

2. koristiti velike grupe da bi generirali asocijacijska pravila kod baza podataka koje imaju

pouzdanost (c) iznad od korisnika određenog minimuma.

Sveukupna pouzdanost u rudarenju asocijacijskih pravila primarno je određena prvom fazom. Nakon što se indentificiraju velike grupe, pripadajuća asocijacijska pravila direktno se izvode. Stoga se vidi da je prebrojavanje velikih grupa centralna zadaća većine algoritama za asocijacijska pravila, a oblikuju se efikasna rješenja da bi se zadovoljio kriterij prve faze.

Apriori algoritam omogućio je jedno od prvih temeljnih rješenja problema. Ostali algoritmi izrasli su na Apriori algoritmu i predstavljaju poboljšanja temeljnog rješenja.

**Algoritam apriori** izračunava učestale grupe u bazi podataka kroz nekoliko iteracija. Iteracija „i“ izračunava sve učestale k-grupe (grupe s k elemenata). Svaka iteracija ima dva koraka:

1. generiranje kandidata,

2. brojanje i selekcija kandidata.

Općenito, algoritam za rudarenje asocijacijskih pravila sadrži sljedeće korake:

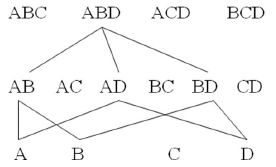
1. u svakom koraku se generira skup od k-grupa kandidata počevši od 1-grupe,

2. izračuna se potpora (s) za sve kandidate u skupu k-grupa,

3. grupe koje nemaju unaprijed određenu minimalnu potporu (s) se odbacuju, a preostale grupe se

nazivaju velike k-grupe.

Ovaj proces se ponavlja dok nema više velikih k-grupa. Efikasnost Apriori algoritma se pokazuje tijekom procesa generiranja kandidata. Apriori koristi tehniku pročišćavanja, naime algoritam smanjuje skup k-grupa kandidata pročiščavajući apriori one grupe kandidata koje ne mogu biti učestale. Pročišćavanje se temelji na opažanju da ako je određena grupa učestala, sve podgrupe od kojih se promatrana grupa sastoji, mogu također biti učestale.



Apriori svojstvo: svaki podskup učestale grupe (podgrupa), također je učestala grupa.

U drugoj fazi, za asocijacijsko pravilo koje pretpostavlja da vrijedi {A, B}→ D, potrebno je da obje grupe {A, B, D} i {A, B} budu s velikom učestalošću. Onda se pouzdanost asocijacijskog pravila računa kao kvocijent potpore pojedinih grupa:

c = s(A, B, D}/s(A, B).

Jaka asocijacijska pravila su ona koja imaju vrijednost pouzdanosti (c) iznad unaprijed zadanog, ili

traženog praga.

Ipak treba voditi računa o tome da neće sva otkrivena jaka asocijacijska pravila, tj. ona koja prelaze prag zahtijevane potpore (s) i pouzdanosti (c), biti dovoljno interesantna da bi se koristila ili uzela u obzir iz razloga jer se ne slažu s nekim otprije poznatim podacima, ili realnošću (postojanje negativne

korelacije). Naime korisnik je taj koji mora voditi računa o onome što želi dobiti rudarenjem podataka, postavljanjem parametara koje će uzeti u obzir i smislom dobivenih rezultata.

Postoje dva tzv. uska grla u Apriori algoritmu. Jedan je kompleksni proces generiranja kandidata za

velike k-grupe, koji koristi najviše vremena, memorije i resursa računala, a drugi je mnogostruka

očitavanja baze podataka.

Zaključak:

Rudarenje za asocijacijskim pravilima ima širok spektar upotrebe od analize potrošačke košarice, medicinske dijagnostike i istraživanja, analize surfanja internetom, itd.

Krajnji korisnici rudarenja za asocijacijskim pravilima u praksi se susreću s nekoliko dobro poznatih problema. Algoritmi ne daju uvijek rezultate u razumnom vremenu, asocijacijska pravila mogu eksponencijalno porasti i postati nezgrapna i nepraktična za upotrebu, pogotovo ako se smanje zahtjevi učestalosti. Također, što je veći skup učestalih grupa, više asocijacijskih pravila se prezentira korisniku, od kojih su mnogi suvišni. Ovo je točno čak i kod oskudnih baza podataka, a kod velikih baza podataka jednostavno je neizvedivo rudariti sve moguće učestale grupe, a kamoli generirati asocijacijska pravila.

**Klastering**

Grupiranje objekata sličnih karakteristika – cilj je pronaći grupe koje se znatno razlikuju.

Najjednostavnija metoda – K-means algoritam – dijeljenje osnovne populacija na *k* segmenata, gdje svaki segment sadrži *n* sličnih elemenata.

Većina metoda klasteriranja koristi Euklidsku mjeru udaljenosti.

Klasteriranje spada u grupu tzv. neusmjerenih metoda (undirected data mining).

Cilj analize klastera je otkrivanje “slicnosti među uzorcima, odnosno podacima”.

Podaci mogu biti:

* podaci o nekim objektima (npr. svojstva, oblici, cijene, ...)
* podaci dobiveni od ispitanika (npr. ocjena kvalitete usluge)
* podaci dobiveni iz mreže (npr. vezano uz sigurnost, kvarovi), itd.

Metode klasteriranja koriste se radi podjele primjera u niz grupa ili podskupova (clusters), koji

zadovoljavaju dva osnovna kriterija:

* svaka grupa predstavlja homogen skup: primjeri koji pripadaju istoj grupi su međusobno slični;
* svaka grupa mora se razlikovati od ostalih grupa, tj. primjeri koji pripadaju određenoj grupi

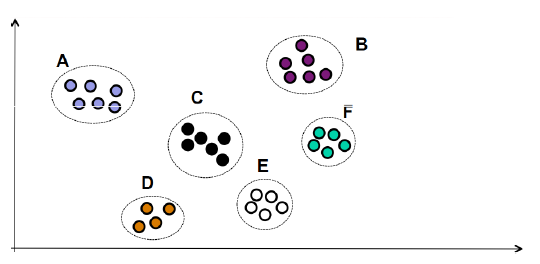
značajno se razlikuju od primjera koji pripadaju ostalim grupama.

Zavisno od konkretne metode, grupe mogu biti definirane na razlicit nacin:

* identificirane grupe mogu biti ekskluzivne, tako da svaki primjer pripada iskljucivo jednoj od grupa;
* grupe se mogu preklapati; primjer može istovremeno pripadati nekolicini grupa;
* grupe mogu biti definirane probabilisticki: u tom slucaju primjer pripada svakoj od grupa s određenom vjerojatnosti
* grupe mogu biti hijerarhijski strukturirane, sa grubom podjelom primjera na najvišem nivou, koji se potom može finije strukturirati na nižim nivoima.

Osnovni pojmovi:

* Centar ili središte klastera je točka čije su vrijednosti parametara srednje vrijednosti parametara svih točaka u klasteru.
* Distanca je općenito udaljenost između dvije točke. Za određivanje pripadnosti grupi, vecina algoritama procjenjuje udaljenost između tocke i klastera centroida. Uobičajena mjera udaljenosti je euklidska metrika koja definira udaljenost između dviju točaka: P = (p1, p2, ....) i Q = (q1, q2, ....) za neki k-dimenzionalni prostor sljedećim izrazom:



**Algoritam K-srednjih vrijednosti**

Ovaj algoritam ima kao ulaznu vrijednost prethodno definiran broj grupa (k).

Srednja vrijednost u algoritmu odnosi se na "prosječnu" lokaciju (u više dimenzionalnom prostoru definiranom atributima).

Vrijednost svakog atributa primjera predstavlja udaljenost tog primjera od ishodišta takvog prostora po koordinati atributa.

Algoritam K-srednjih vrijednosti je jednostavna, iterativna procedura u kojoj centralnu ulogu igra pojam centroida.

**Koraci K-algoritma**

Odabrati K točaka u prostoru zadanih uzoraka - inicijalna grupa centroida

* Preporučljivo je centroide locirati što dalje jedan od drugoga.

Uzeti svaku točku iz zadanog skupa podataka i pridružiti je najbližem centroidu,

Izračunati nove pozicije centroida i ponovno svaku točku pridružiti novim centroidima

Postupak se ponavlja dok centroidi ne prestanu pomicati

**Nedostatci K-algoritma**

* Nije specificiran način kako odabrati inicijalne pozicije centroida
* Popularan način je “slučajno” odabrati k od zadanih uzoraka.
* Rezultat značajno ovisi o inicijalnom odabiru pozicije centrioda. Rješenje je u većem broju pokušaja sa različitim odabirom pozicija inicijalnih centroida.
* Može se dogoditi da uzorak blizak i ne postoji, pa se i ne može podešavati.
* Rezultat ovisi o vrijednosti k.
* Najveći problem je u tome što često unaprijed nije poznato koliko postoji klastera.
* Ne postoji opće teorijsko rješenje za pronalaženje broja klastera za bilo koji skup uzoraka. Jednostavniji pristup je usporediti rezultate za veći broj primjena različitih k klasa i odabrati najbolju prema zadanom kriteriju.

**Hijerarhijsko klasteriranje**

Za hijerarhijsko klasteriranje karakteristično je grupiranje objekata u stablo klastera. Ovo klasteriranje klasificira se na aglomerativno klasteriranje (klasteriranje metodologijom od dna prema vrhu /engl. Bottom-up/) i divizijsko klasteriranje, čiji je smjer klasteriranja od vrha prema dnu (engl. Top-down).

Za odabrani skup N elemenata za grupiranje i NxN matricu udaljenosti (ili sličnosti), osnovni

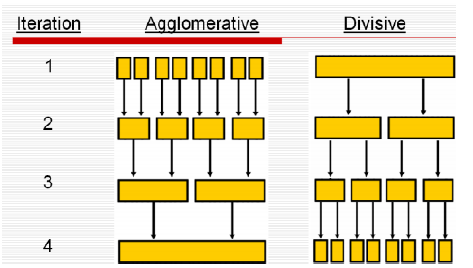
postupak hijerarhijskog grupiranja sastoji se od koraka:

1. Započeti pridruživanjem svakog elementa vlastitoj grupi (klasteru) tako da od N elemenata dobijemo N klastera, od kojih svaki klaster sadrži samo jedan element odnosno član.

2. Pronaći najbliži (najsličniji) par klastera i spojiti ih u jedan klaster, tako da sada imamo jedan klaster manje.

3. Izračunati udaljenosti (sličnosti) između novog klastera i svakog od starih klastera.

4. Ponavljati korake 2 i 3 sve dok svi elementi ne budu članovi jedne grupe veličine N.

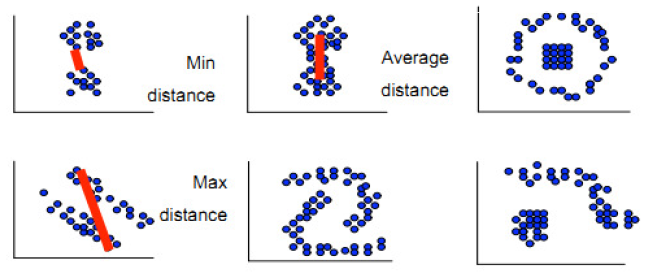


**Hijerarhijske metode na temelju udaljenosti**

**Single-link clustering:** obuhvaća samo najmanju udaljenost – single link: promatramo distancu između dvaju klastera kao najkraću udaljenost između bilo kojeg člana jednog i bilo kojeg člana drugog klastera.

**Complete-link clustering**: obuhvaća sve udaljenosti – complete link: promatramo distancu između dvaju klastera kao najdulju udaljenost između bilo kojeg člana jednog i bilo kojeg člana drugog klastera.

**Average-link clustering**: promatramo distancu između dvaju klastera kao srednju udaljenosti između svih članova jednog i svih članova drugog klastera.



**Posebni aspekti metoda klasteriranja podataka**

Proces pripreme podataka i efikasnost metode:

* pitanje mjerenja udaljenosti primjera (metrika prostora),
* izbor ispravnog broja grupa,
* interpretacija grupa

Većina metoda segmentiranja koristi Euklidsku mjeru udaljenosti u prostoru primjera. Nominalni atributi moraju se prethodno transformirati i normirati prije za primjenu metode segmentiranja. O

ovoj transformaciji umnogome zavisi koliko će ovi atributi biti važni za sam proces segmentiranja. Oni mogu biti dominantni, ali i potpuno nebitni, ako se transformacija izvede na određeni način.

Ako je broj grupa K u metodi K-srednjih vrijednosti pogrešno odabran, konačni rezultati neće biti dobri. Ispravan pristup odabiru broja grupa bio bi da se eksperimentira s različitim brojem grupa. U principu, optimalan broj grupa imat će najpovoljniji omjer intra-grupnih i intergrupnih udaljenosti primjera. Sofisticiranije tehnike segmentiranja mjere ovaj omjer i same automatski optimiziraju broj grupa u dodatnoj petlji (AutoClass). Jednom otkrivene grupe potrebno je interpretirati, kako bi rezultat segmentiranja podataka bio od koristi za proces obrade podataka.

**Primjena tehnika klasteriranja**

Primjena kada se očekuje postojanje "prirodnih" grupa u podacima. Otkriveni segmenti ili grupe podataka trebali bi predstavljati grupe primjera koji imaju mnogo toga zajedničkog. Stvaranje grupa primjera prije primjene neke druge tehnike modeliranja podataka (neuralnih mreža, stabla odlučivanja) može znatno reducirati kompleksnost određenog problema, podjelom skupa primjera za modeliranj. Ovakvi podskupovi primjera za učenje potom se mogu modelirati odvojeno, a takva dvo-stepena procedura na kraju može rezultirati boljim konačnim rezultatima (bilo u prediktivnom ili deskriptivnom smislu), nego bez prethodne primjene tehnika segmentiranja podataka.

Metoda klasteriranja najčešće se koristi za početnu segmentaciju tržišta. Klastere je moguće formirati na temelju čitavog niza varijabli, kao što su spol, godine starosti, prosječna primanja, prosječna potrošnja u nekom intervalu, itd. Isto tako klasteri se mogu primijeniti i za praćenje trendova tržišnih

segmenata, a postoji i niz drugih situacija na koji se klasteriranje može s uspjehom primijeniti.

**Bayesove mreže**

Za donošenje važnih odluka u poslovnom okruženju potrebno je tražiti savjet osoba koji su eksperti u

području od interesa ili u odgovarajućoj domeni znanja. Ovi eksperti ili savjetnici formiraju svoje savjete i izvještaje na osnovu akumuliranog znanja i iskustva u svom području te na osnovu dostupnih izvora informacija. Kako automatizirati ovaj proces ljudskog rezoniranja u ekspertnim područjima? **Ekspertni sustavi.**

Ekspertni sustav se dijeli na:

* Inference Engine
* Knowledge Base

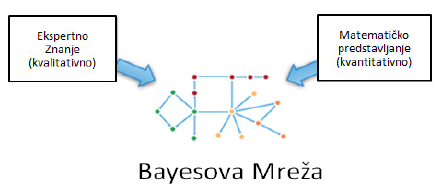
Pri korištenju ekspertnog sustava Inference engine donosi zaključke o podacima dostupnim u bazi znanja na sličan način kao što to radi i čovjek. Trenutno korišteni pristupi za modeliranje znanja koriste stabla odluke, umjetne neuralne mreže i mreže zasnovane na pravilima (eng. 'rule-based'). Za analizu ovakvih modela koriste se procjena gustoće, klasifikacija, regresija te tehnike klasteriranja.

Što ako svi podaci za modeliranje znanja nisu dostupni? Kako ujediniti kvalitativno i kvantitativno znanje? Kako prikazati to znanje na lako razumljiv način svim bitnim sudionicima poslovnog procesa?

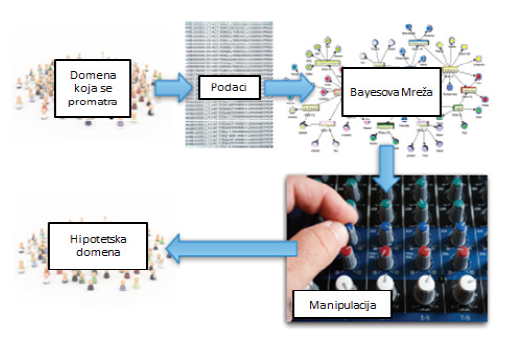
Jedan od odgovora: Bayesove mreže. Primjena: Analitika, simulacije, optimizacija cilja,

detekcija anomalija, dijagnostika, otkrivanje znanja, predstavljanje znanja, opažačko zaključivanje...

Unifikacija znanja:



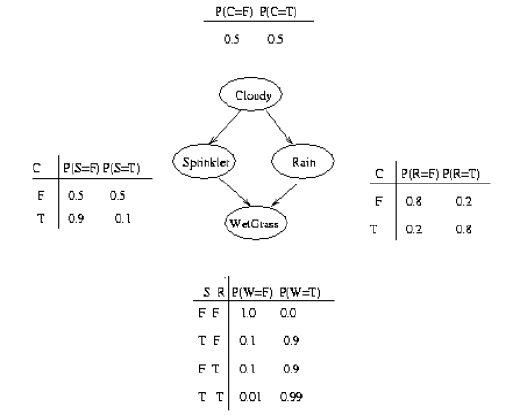
Formalno rezoniranje o posljedicama hipotetskih djelovanja:



Bayesova mreža je grafički model koji prikazuje probabilističke veze u određenom skupu promatranih

varijabli. Bayesova mreža se sastoji od usmjerenog acikličkog grafa u kojem svaki čvor predstavlja slučajnu varijablu, a svaka veza probabilističku zavisnost te skupa tablica uvjetnih vjerojatnosti za svaku varijablu. Usmjereni aciklički graf i tablice vjerojatnosti zajedno opisuju distribuciju vjerojatnosti u domeni.

**Primjer:**



U navedenom primjeru svi čvorovi su binarni to jest mogu poprimiti samo dvije vrijednosti, T ili F (to jest 'True' ili 'False') iako čvorovi mogu poprimati i više od dvije vrijednosti.

Iz grafa možemo očitati da ukoliko je trava mokra (W=True) onda mogu biti dva moguća uzroka za to, to jest ili je uključena prskalica (S=True) ili pada kiša (R=True). Težina ove veze je prikazana u tablici.

Primijetimo i da je u prikazanom grafu zadnji čvor ovisan samo o svojim roditeljima ali ne i o ostalim 'precima' (u ovom slučaju to je prvi čvor). Prema lančanom pravilu vjerojatnosti, združena vjerojatnost svih čvorova u danom grafu je**:**

**P(C, S, R, W) = P(C) \* P(S|C) \* P(R|C,S) \* P(W|C,S,R)**.

Zbog uvjetne neovisnosti čvorova, gornji izraz se može kraće napisati i kao:

**P(C, S, R, W) = P(C) \* P(S|C) \* P(R|C) \* P(W|S,R)**

gdje smo pojednostavili/skratili zadnja dva dijela izraza pošto je R neovisan od S a W je

neovisan od C.

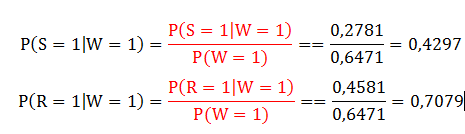
Ovo je vrlo bitno svojstvo Bayesovih mreža jer omogućava kompaktniji prikaz združene distribucije varijabli (ne moraju se zapisivati sve moguće kombinacije, već samo one kombinacije varijabli među kojima postoji ovisnost).

Za n binarnih čvorova potpuna združena distribucija bi zahtjevala vrijednosti dok je za kompaktiranu formu potrebno n vrijednosti gdje je ‘k’ maksimalna vrijednost broja veza koje ulaze u neki čvor.

Uzmimo opet za primjer da je trava u našem grafu mokra. Mogu biti dva moguća uzroka za ovo; ili je

prskalica uključena ili pada kiša. Pitamo se za koje od ova dva uzroka vjerojatnost veća? Korištenjem

Bayesovog pravila računamo vjerojatnost za oba uzroka:



Bayesovo pravilo:



Iz izračunatog vidimo da je vjerojatnije da će trava biti mokra zato što pada kiša. Ovaj primjer pokazuje probabilističko zaključivanje (eng. Probabilistic inference) što je najuobičajenija primjena Bayesovih mreža.

U navedenom primjeru također možemo primijetiti da se dva uzroka za mokru travu na neki način

'natječu' da objasne promatrani podatak. Prema tome, S i R postaju uvjetno ovisni ako promatramo

njihovo dijete W, iako su oni možda marginalno neovisni.

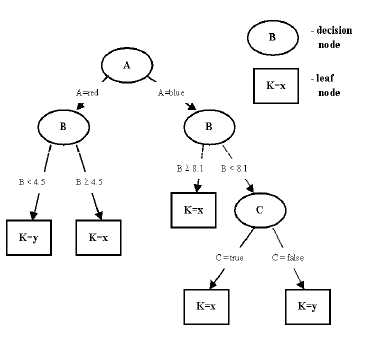
Na primjer, ako je trava mokra a znamo i da pada kiša, onda će vjerojatnost da je prskalica upaljena

pasti na 0.1945. Ovakvo rezoniranje se zove eliminacija (eng. 'explaining away') te je također

bitan dio Bayesovih mreža.

**Stabla odlučivanja**

**Stabla odlučivanja** služe za klasificiranje atributa s obzirom na ciljnu varijablu. Osnovna zadaća je određivanje varijabli i njihovih vrijednosti. Klasifikacijski algoritam u formi stabla – dva tipa čvorova povezanih granama, čvor odluke i krajnji čvor.



Stabla odlučivanja vrlo su moćne i popularne tehnike modeliranja za klasifikacijske i prediktivne probleme. Privlačnost stabla odlučivanja leži u činjenici da, u odnosu na, npr. neuralne mreže nude model podataka u razmljivom obliku, tj. obliku pravila. Ta pravila se lako mogu direktno interpretirati običnim jezikom, ili pak koristiti u nekom od jezika za rad s bazama podataka, npr. SQL.

Za neke je probleme od ključne važnosti samo točnost klasifikacije ili predikcije modela. U takvim slučajevima čitljivost modela nije od presudne važnosti. No, u drugim situacijama upravo sposobnost

interpretiranja modela „ljudskim“ jezikom je od ključne važnosti.

Primjer, marketing: segmentacija populacije kupaca -> efektivna kampanja

Razlikujemo **dva tipa čvorova** povezanih granama:

* krajnji čvor (engl. leaf node) - njime završava određena grana stabla. Definira klasu kojoj pripadaju primjeri koji zadovoljavaju uvjete na toj grani stabla;
* čvor odluke (engl. Decision node) - definira određeni uvjet u obliku vrijednosti određenog atributa (varijable), iz kojeg izlaze grane koje zadovoljavaju određene vrijednosti tog atributa.

**Osnovni preduvjeti za korištenje** tehnike stabla odlučivanja su:

* Opis u obliku parova vrijednosti – konačan broja atributa;
* Prethodno definiran konačan broj klasa (vrijednosti ciljnog atributa) – kategorije definirane unaprijed i konačan broj;
* Klase moraju biti diskretne – svaki primjer mora pripadati samo jednoj od postojećih klasa, kojih mora biti znatno manje negoli broja primjera;
* Značajan broj primjera – poželjno barem nekoliko stotina primjera.

Stablo se dobiva „učenjem“ na podacima, na način da se vrši grananje izvornog skupa podataka u

podskupove na temelju testiranja vrijednosti varijabli. Proces se ponavlja na svakom izvedenom podskupu na rekurzivni način. Rekurzija je završena kada podskup određenog čvora ima sve iste vrijednosti izlazne varijable, ili kada daljnje grananje više ne pridonosi poboljšanju rezultata.

**Određivanje kompleksnosti** se još naziva i **podrezivanje stabla**. Postoje dvije tehnike podrezivanja:

* Odozgo prema dolje - kod te tehnike se prekida daljnji rast stabla pomoću pravila zaustavljanja;
* Odozdo prema gore - kod te tehnike se prvo kreira stablo najveće kompleksnosti, a zatim se odstranjuju grane dok se ne postigne željena složenost.

**ID3 algoritam**

ID3 je jedan od algoritama za klasifikaciju pomoću kojeg se pokušava riješiti problem važnosti atributa u cjelokupnom odlučivanju, odnosno klasificiranju.

ID3 pretražuje preko atributa svih primjera u skupu podataka, te nalazi atribut koji najbolje odvaja primjere određene klase.

Ukoliko atribut savršeno razdvaja klase, ID3 algoritam se zaustavlja. Inače se rekurzivno izvršava na m podskupova tražeći najbolje atribute za njihovo razdvajanje.

Naponema: pogledati primjer na slideu 130, i potražiti objašnjenje.

**Prednosti ID3 algoritma**:

* Sposobnost generiranja razumljivih modela;
* Relativno mali zahtjevi na računalne resurse (vrijeme i memorija);
* Sposobnost korištenja svih tipova atributa (kategorički, numerički);
* Stabla odlučivanja jasno odražavaju važnost pojedinih atributa za konkretni klasifikacijski odnosno predikcijski problem.

**Nedostatci ID3 algoritma**:

* Metode stabla odlučivanja su manje prikladne za probleme kod kojih se traži predikcija kontinuiranih vrijednosti ciljnog atributa;
* U nekim situacijama generiranje stabla odlučivanja može biti računalno zahtjevan problem. Sortiranje kandidata za testiranje na čvorovima stabla može biti zahtjevno, kao i metode skraćivanja stabla, kod kojih je često potrebno generirati velik broj stabala da bi odabrali ono koje je najbolje za klasifikaciju primjera određenog problema;
* Metode stabla odlučivanja sklone su greškama u višeklasnim problemima sa relativno malim brojem primjera za učenje modela;
* Stabla odlučivanja nisu dobro rješenje za klasifikacijske probleme kod kojih su regije odredjenih klasa omedjene nelinearnim krivuljama u višedimenzionalnom atributnom prostoru. Većina metoda stabla odlučivanja testiraju u svojim čvorovima vrijednosti jednog atributa, i time formiraju pravokutne regije u višedimenzionalnom prostoru.

**Neuronske mreže**

Neuron kao glavna komponenta mreže. Neuroni kombiniraju ulaze u jedinstveni rezultat, najčešće

funkcijom zbrajanja

* potom se primjenjuje funkcija transformacije (najčešće rezultat u intervalu 0-1)

Najpoznatija topologija neuralne mreže je više-slojni perceptron.



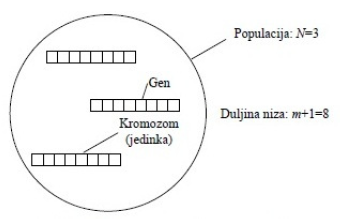
**Genetički algoritmi**

Algoritam oponaša prirodnu evoluciju:

* Ciklusi selekcije, reprodukcije i manipulacije

Stvara se početna populacija:

* Svaka jedinka se ocjenjuje funkcijom cilja, jedinke se rangiraju rezultatima funkcije
* Najniže rangirane jedinke se odbacuju, genomi se rekombiniraju, nastaje iduća generacija
* Jedinke se ocjenjuju sve dok se ne zadovolji uvjet zaustavljanaja evolucijskog procesa



**Klasifikacija u sustavu proračuna**

Prihodi, primici, rashodi i izdaci proračuna i financijskog plana iskazuju se prema proračunskim klasifikacijama. Zakon o proračunu (NN 87/2008), Članak 3, definira u 6 točaka 6 klasifikacija:

* **t. 10.: ekonomska klasifikacija:**  Prikaz prihoda i primitaka po prirodnim vrstama te rashoda i izdataka prema ekonomskoj namjeni kojoj služe i razvrstani su u razrede, skupine, podskupine, odjeljke i osnovne račune,
* **t. 14.: funkcijska klasifikacija:** Prikaz rashoda proračuna te proračunskih i izvanproračunskih korisnika razvrstanih prema njihovoj namjeni,
* **t. 18.: izvori financiranja:** skupine prihoda i primitaka iz kojih se podmiruju rashodi i izdaci određene vrste i namjene,
* **t. 24.: lokacijska klasifikacija:** Prikaz rashoda i izdataka prema teritorijalno definiranim cjelinama u skladu s ustrojem Republike Hrvatske, država Europske unije te ostalih država,
* **t. 31.: organizacijska klasifikacija:** Prikaz povezanih i međusobno usklađenih (hijerarhijski s obzirom na odnose prava i odgovornosti) cjelina proračuna i proračunskih korisnika koje odgovarajućim materijalnim sredstvima ostvaruju postavljene ciljeve, a uspostavljena je kroz razdjele, glave i proračunske korisnike,
* **t. 43.: programska klasifikacija:** prikaz programa i njihovih sastavnih dijelova: aktivnosti i projekata, definiranih u skladu s aktima planiranja te ciljevima i zadaćama proračuna te proračunskih i izvanproračunskih korisnika.

**Dimenzijski model**

Multidimenzionalna baza podataka: mjere, dimenzije, hijerarhije i kocke

Relacijska baza podataka: tablice, retci, stupci, i relacije

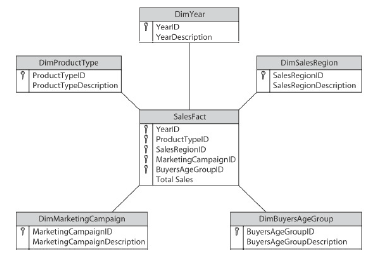
**Atribut** je dodatna informacija koja se odnosti na člana dimentija koji nije karakteristični identifikator ili opis člana.

**Mjera** je numerička kvantiteta koja izražava neki aspekt organizacijskih performansi. Tako prikazana informacija se koristi da bi poduprla ili evaluirala donošenje odluke i performanse organizacija. Mjera se isto može zvati i činjenica.

**Dimenzija** je kategorizacija koja se koristi radi proširivanja agregirane mjere radi otkrivanja njenih sastavnih dijelova.

Shema zvijezde je shema relacijske baze podataka koja sadrži mjere i dimenzije u *data mart*-u.

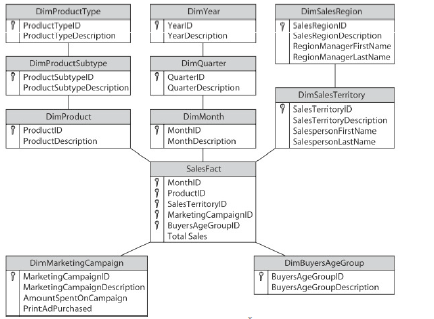
Mjere se spremaju u tablicu činjenica, a dimenzije se spremaju u dimenzijske tablice.



Hijerarhija je struktura koja se sastoji od dvije ili više razina povezanih dimenzija. Dimenzija na višoj razini potpuno sadrži jednu ili više dimenzija na idućoj nižoj razini.

Npr. dimenzija **godina** sadrži kvartale. Nadalje, **kvartali** sadrže **mjesece**.

Shema pahuljice: svaka razina hijerarhije se sprema u posebnu dimenzijsku tablicu. Ova shema predstavlja hijerarhiju na način sličan relacijskim bazama podataka.



**Kocka** je struktura koja sadrži vrijednost za jednu ili više mjera za svaku jedinstvenu kombinaciju članova svih svojih dimenzija. (*A cube is a structure that contains a value for one or more measures for each unique combination of the members of all its dimensions*)

To su detaljne vrijednosti. Kocka također sadrži prikupljene vrijednosti formirane u hijerarhiji dimenzija ili kad se neka dimenzija izostavi iz hijerarhije.

