

Podniková analytika

Úvod do PA

Obsah

- Základné informácie k predmetu (výučbe)
- Podniková Analytika (PA) / Podniková inteligencia (PI)
- Dátová analytika (DA) vs. PA
- Pragmatický / evolučný pohľad na PA
- Rozsah PA(DA) – typy analytik, typy úloh + príklady
- Dáta v PA(DA) – metriky, typy atribútov

Predmet PA – Ciel' a témy

- Ciel' predmetu: Práca s **R/Python** na vybraných problémoch analýzy dát + základy metód analýzy dát, reportovania a využitia dát, atď.
- Hlavné témy:
 - Základné pojmy PA/DA
 - Rozhodovacie procesy a systémy pre podporu rozhodovania
 - Dátové sklady
 - Proces objavovania znalostí v databázach, resp. iných typoch zdrojov
 - Optimalizačné prístupy pre využitie dát získaných analýzou

Predmet PA – Organizácia výučby

- Rozsah
 - 2h Prednášky + 2h Cvičenia
 - Pre organizáciu výučbu a úlohy k zápočtu – **MS Teams**, Moodle (testy)
- Podmienky absolvovania:
 - Zápočet
 - Absolvovanie cvičení (max. 3 neúčasti)
 - Aspoň 21 bodov zo 40 k zápočtu
 - 10 bodov - písomka 1 (test cez moodle) - teória - prednášky 1-4
 - 10 bodov - písomka 2 (test cez moodle) - teória - prednášky 5-8
 - 15 bodov - zadania v Teams, práca na úlohách (assignments) - rôzne úlohy Python/R
 - 5 bodov - účasť na prednáškach (za každé 2 účasti získate 1 bod, a to do maxima 5 bodov, čiže plný počet získate ak sa zúčastníte aspoň 10 prednášok z 12 plánovaných)
 - Skúška (60 bodov)

Predmet PA ... vstup = Python a R

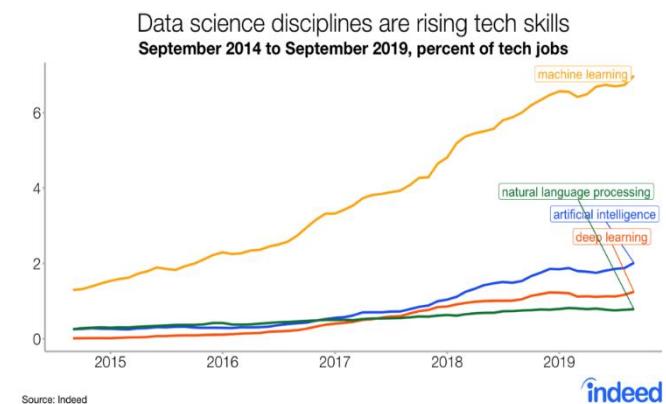
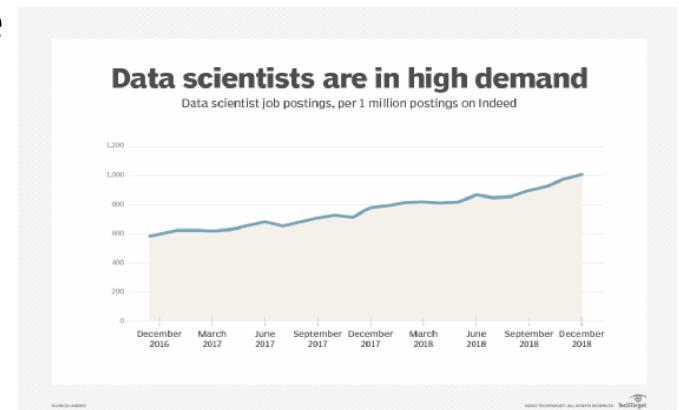
- Programovanie v Python a R = vid'. JDA, aspoň
 - Základy jazyka Python a R, tvorba funkcií
 - Práca s tabuľkami dát (data frames, ...)
 - Grafy v Python a R, explorácia dát pomocou grafov
 - Reportovanie (použitie Jupyter notebookov, (R)markdown), tvorba web aplikácií (Rshiny)
- Študenti ŠP IS
 - Odporúčam rýchlo začať, napr. s tutoriálmi k R, Python ste zrejme mali

Prečo práve analýza dát ?

- Analýza dát = Dátová analytika (DA) ... Inak aj napr. Data Science, často Business analytics (+ Machine/Deep Learning + Artificial Intelligence + Big Data ...)
- Z obsahovej stránky
 - Je to jedna z kľúčových oblastí informatiky (HI, aj IS)
 - Patrí k samozrejmým a silnejúcim prvkom v rámci procesov firiem, organizácií, ...
 - Konkurencieschopnosť bez analýzy dát prakticky neexistuje, to platí ja pre iné sféry ako firmy (napr. výskumné procesy)
- Z pragmatickej stránky
 - Firme alebo organizáciu prináša profit (na prvý pohľad môže byť rôzneho typu – priamo financie, úspora času, optimalizácia pracovného priestoru, ...)
 - Optimalizovanie procesov ako cieľ si vždy vyžaduje analýzu stavu (dostupných dát) ako vstup
 - Pre analytika: prináša dobre platenú prácu !

Prečo pracovať v analýze dát ?

- CareerCast.com publikoval analýzu aktuálnych zamestnaní, kde uviedol ako top zamestnanie pozíciu **Data Scientist**, obzvlášť schopný pracovať s **Big Data**
- Harvard Business Review – „Data Scientist: The Sexiest Job of the 21st Century“
- Podpora v rámci HI = máme
 - BC HI
 - JDA (základy R/Python, práca s dátami)
 - Štatistika (Num. mat.)
 - PA (aplikovanie R/Python v analýze dát)
 - ING HI ... množstvo predmetov rozširujúcich širší úvod / prehľad z PA
 - Objavovanie znalostí
 - Technológie sprac. veľkých dát
 - Strojové učenie
 - Pokročilé metódy analýzy dát (deep learning)
 - Apl. Štatistika
 - Ekonometria
 - Manažment znalostí (text mining)
 - Manažérské informačné systémy
 - Sémantický a sociálny web
 - ...



Definícia podnikovej analytiky

Podniková analytika (PA, angl. **Business Analytics / BA**) predstavuje postupy použitia *dát, znalostí, (informačných) technológií, aplikácií a metód* (štatistické analýzy, kvantitatívne metódy, matematické / počítačové modely), pre uľahčenie historického *porozumenia* podnikových operácií a tvorbe lepších *na faktoch založených rozhodnutí* ďalšieho plánovania.

Dátová analytika (DA) \approx PA

- Mohli by sme definovať **dátovú analytiku** ako rozšírenie PA nad rámec podnikových operácií a procesov môžeme analyzovať podnikové procesy, ale aj medicínske procesy (napr. diagnostika chorôb s podporou DA), procesy vo verejnej správe, vedecko-výskumné procesy, hľadať a skúmať pritom rôzne typy spoločensko-ekonomických ako aj prírodrovedných javov (napr. hľadať v seismických dátach zemetrasenia, klasifikovať ich a predvídať, či predvídať vývoj klimatických dopadov, ...), a iné ... => a tak by sme mohli trochu zovšeobecniť pôvodnú definíciu PA nasledovne a dostať tak:

Dátová analytika (DA) predstavuje postupy použitia dát, znalostí, (informačných) technológií, aplikácií a metód (štatistické analýzy, kvantitatívne metódy, matematické / počítačové modely), pre uľahčenie historického porozumenia **ľubovoľných skúmaných procesov (rôznych organizačných, spoločensko-ekonomických, vedeckých, výskumných a aplikačných procesov, vrátane ich rôznorodých zdrojov dát) a na faktoch založeného rozhodovania pre potreby plánovania ďalších krokov v kontexte ďalšieho skúmania alebo optimalizácie týchto procesov.**

- Inak povedané DA \approx PA (\approx je mat. „približne sa rovná“) ... dokonca môžeme povedať, že DA = PA, ak doménu PA rozšírimo nad rámec „business“ ... metódy a nástroje sú prakticky totožné (je potrebné ich len rozumne aplikovať podľa danej domény, procesu, problému, ...)

Význam podnikovej analytiky (a DA)

- Existuje silný vzťah medzi aplikáciou (použitím) PA v podniku a:
 - ziskovosťou podniku
 - príjmami z podnikania
 - výnosmi pre akcionárov
- Výhody PA
 - Zlepšuje porozumenie dát
 - PA je veľmi dôležitá pre podniky v rámci snahy udržať si konkurencieschopnosť v podnikaní
 - Umožňuje vytvárať informatívne reporty
- Význam DA vieme podobne dodefinovať - napr. výskum
 - DA prináša zlepšenie daného procesu (napr. analýza odlesňovania), jeho pochopenia (príčiny a možné dôsledky odlesňovania), prináša výsledky efektívnejšie (vedci efektívnejšie skúmajú odlesňovanie, sú schopní vďaka analytike priniesť nové lepšie znalosti a návrhy na vhodné zásahy), oproti iným skupinám nepoužívajúcim DA dosahujú lepšie výsledky a skôr, ...

Podniková inteligencia

- Podniková inteligencia (Business Intelligence, BI)
 - Existujú rôzne pohľady na to, čo je BI a BA (Business Analytics) a aký je medzi nimi vzťah
 - Často sa BA a BI používajú rovnako pre spojený obsah, ale existujú aj odlišné pohľady, napr.:
 - BI predstavuje techniky používané pre sledovanie, objavovanie a analýzu podnikových dát => metódy a technológie pre prístup k dátam, interakciu s dátami a analýzu dát s cieľom riadiť podnikanie, zlepšiť výkonnosť, objaviť nové poznatky a fungovať efektívne ... cieľom je najmä (systematická) práca s dátami, podpora operatívneho plánovania a sledovanie efektívnosti
 - BA sú znalosti, technológie, aplikácie a metódy pre kontinuálne skúmanie historickej výkonnosti podniku s cieľom hlbšieho porozumenia a riadenia podnikového plánovania => cieľom je lepšie pochopenie histórie a faktami riadené strategické plánovanie

Podniková analytika vs. inteligencia

- BA
 - Cieľom je analýza histórie podniku a jeho operácií (viac ad-hoc popis)
 - Použitie najmä pre lepšie pochopenie a porozumenie
 - Zameriava sa na exploratívne a interaktívne analýzy
 - Pri popisovaní sa často pozera na základné dátá (aj v detailoch)
 - Snaží sa ukázať výnimočné udalosti na základných dátach
 - Dátová analýza, Dátová veda
 - **Odpovedá na otázku Prečo sa „to“ deje + Čo sa stane v budúcnosti ? => Explorácia, Predikcia a Predvídadost’**
- BI
 - Orientuje sa na štandardné metriky popisu efektívnosti podniku
 - Uprednostňujú sa pevne definované pohľady na dátá
 - Používateľ zvyčajne dostáva agregované dátá
 - Výnimočné udalosti sa sledujú na agregovanej úrovni
 - Má technologický a viac štandardizovaný charakter ako BA
 - Dátové sklady, Vizualizácie, Dashboard-y
 - **Odpovedá na otázku Čo sa deje v podniku ? => Viditeľnosť**

BA/BI – Pragmatický pohľad

- Pohľad na rozdiely je rôznorodý – často sa jedným z pojmov popisuje súbor oboch
- BI je niekedy uvažovaná ako jeden z evolučných krokov podnikovej analytiky
 - podobne ako tzv. Systémy pre podporu rozhodovania (DSS – Decision Support Systems)
- Pragmatický pohľad (spájajúci)
 - BA je všetko čo poskytuje pohľad na dátu a chovanie podniku s cieľom podporiť proces ďalšieho rozhodovania a plánovania
 - Zahŕňa exploratívnu historickú analýzu, predikciu a preskripciu, riadenie procesu toku dát, agregované sledovanie efektívnosti podniku, a to vrátane príslušných metód a technológií pre takéto postupy

Evolučný pohľad na BA

- Definícia BA ako počítačovej podpory rozhodovacích procesov prináša so sebou pohľad ako na súbor techník, ktorý
 - prešiel postupnou evolúciou a pridaním nových dostupných metód a technológií
 - pričom vývoj a súbor techník zahŕňa
 - DSS (Decision Support Systems) – zač. 70-te roky – práca s rozhodnutiami a detailnou analýzou v štruktúrovanej forme – zamerané na *analytikov*
 - BI (Business Intelligence) – dátové sklady a spracovanie tokov dát, reportovanie, dolovanie dát) – zač. 80/90-te roky – podpora rozhodnutí cez viditeľné agregované dáta – zamerané na *manažérov*
 - BA (Business Analytics) – už nielen podpora *manažérskych* rozhodnutí pomocou BI techník agregovaného sledovania činnosti a reportovania, ale s použitím rozšírení BI techník aj detailná ad-hoc analýza s priamou podporou pre *analytikov* v rámci historickej analýzy a strategického plánovania – nástroje DSS + BI => BA (analytika a dátová veda = Data Science)

Typy analytických úloh v PA/DA

- Základné delenie podľa prístupu = podľa odlišnosti vstupov, prostriedkov a cieľov danej analytiky
 - Deskriptívna analytika
 - Prediktívna analytika
 - Preskriptívna analytika
- Iné delenia
 - Typy PA/DA podľa aplikačnej domény
 - Typy PA/DA podľa zvolených nástrojov = vid'. Evolučný pohľad na BA
 - ...

Deskriptívna (reportovacia) analytika

- Deskriptíva = popis
- Základná otázka
 - Čo sa deje v organizácii (skúmaných procesoch), aké sú trendy, aké sú príčiny ?
- Postup
 - Používa dáta pre pochopenie minulosti a prítomnosti
- Metódy a technológie
 - Exploratívna analýza, vizualizácia, periodické reportovanie, analýza trendov, dashboard-y, dátové sklady, sledovanie metrík (napr. výkonnosť podniku, resp. metriky v kontexte skúmaného procesu DA)

Prediktívna analytika

- Predikovať, predvídať, predpovedať, ...
- Základná otázka
 - Aké sú predpoklady o budúcnosti na základe historických dát ?
- Postup
 - Detailne analyzuje výkonnosť (metriky) v minulosti pre predikciu budúcnosti ... Nehovorí čo máme urobiť do budúcnosti, ale ako to bude vyzeráť ak by bol vývoj podobný historickým dátam
- Metódy a technológie
 - Štatistická analýza a dolovanie dát (data mining)

Preskriptívna analytika

- Preskriptívna (normatívna / rozhodovacia) analytika
- Preskripcia = predpis
- Základná otázka
 - Čo sa aktuálne deje a ktoré rozhodnutia pre budúcnosť prijať aby sme dosiahli najlepšiu výkonnosť ?
- Postup
 - Vytvára modely a riešenia pre rozhodnutia o budúcnosti s cieľom zistíť (a zabezpečiť) ich maximálnu možnú efektívnosť => cieľom je dodať rozhodnutie alebo odporúčanie pre špecifickú akciu, pričom tak môže byť urobené v podobe reportu alebo automaticky systémom rozhodovania
- Metódy a technológie
 - Operačný výskum, optimalizačné metódy, multikriteriálne rozhodovanie, simulácie, rozhodovacie procesy, expertné systémy, systémy manažmentu znalostí, ...

Typy BA podľa aplikačnej domény

- Manažment vzťahov so zákazníkmi, behaviorálna analytika
- Finančné a marketingové aktivity
- Analytika maloobchodných tržieb
- Rozhodnutia o cenách produktov
- Analýza rizík, analýza podvodov
- Riadenie podnikových zdrojov
- Plánovanie ľudských zdrojov
- Tímové stratégie
- Analýza dopravných problémov
- Manažment dodávateľských reťazcov
- Telekomunikácie
- ...

Príklad – Analýza maloobchodu

Rozhodnutie o znížení cien v maloobchode

- Väčšina predajcov sa snaží sezónne vyprázdníť sklady redukciou cien
- Otázka: Kedy a ako (o kol'ko) redukovať ceny ?
- Analytické prostriedky pre tento typ úlohy
 - Deskriptívna analytika: prehľadanie historických dát pre podobné produkty (ceny, počty predaných produktov, reklama, ...)
 - Prediktívna analytika: model predikcie predajov na základe navrhnutnej ceny
 - Preskriptívna analytika: nájdenie najlepšej kombinácie nastavenia ceny a použitej reklamy pre maximalizáciu tržieb

Príklad – Analýza zákazníkov

- Spoločnosť vlastniaca hotely a kasína
- Používa deskriptívnu analytiku na:
 - Popis vyťaženosť hotelov
 - Prehľad aktivít v kasínach
 - Náhľad na odpovedajúce výsledky z pohľadu ziskovosti
- Používa prediktívnu analytiku na:
 - Predikciu požiadaviek na obsadenie izieb
 - Segmentáciu zákazníkov podľa ich hráčskych aktivít (v kasíne)
- Používa preskriptívne modely (s cieľom optimalizácie zisku) na:
 - Nastavenie cien izieb
 - Alokáciu obsadenia izieb
 - Ponúkanie výhod a odmien zákazníkom podľa ich zaradenia do špecifického segmentu

Príklad (DA) – Analýza klimatických zmien

- Podpora výskumných procesov v oblasti štúdia klimatických zmien – využitie IT nástrojov pre podporu vedcov pri ich skúmaní, získavaní znalostí, pochopení stavu a dopadov klímy atď.
- Deskriptívna analytika
 - Popis dát o stave a historickom vývoji klímy
- Prediktívna analytika
 - Modely predikcie vývoja klímy
- Preskriptívna analytika
 - Ktoré parametre by sme vedeli nastaviť aby sme dosiahli vývoj klímy vhodný pre našu budúcnosť

Čo môžu byť dáta v PA/DA ?

- Základné stupne dát v PA / DA
 - Dáta – nazbierané fakty, čísla, obrázky, ... => tvoria Databázy – kolekcie počítačových súborov obsahujúcich dáta
 - Informácie – získané z analýzy dát, správne interpretované dáta (t.j. vieme interpretovať čo dané číslo znamená, obrázok obsahuje, atď.)
 - Znalosti – informácie interpretované v kontexte ich použitia – napr. vieme že ak je číslo interpretované ako počet predajov X dostatočne veľké, vykážeme zisk Y a vieme na základe toho uskutočniť akciu Z (napr. nákup nových strojov)
- Príklady vstupných dát podnikov pre BA
 - Výročné správy
 - Účtovné audity
 - Finančné analýzy zisku
 - Ekonomické trendy
 - Marketingové prieskumy
 - Výkonnosť operačného manažmentu
 - Merania výkonnosti ľudských zdrojov
 - ...

Metriky

- Špecifický typ vstupných alebo výstupných dát o podniku, pri rozšírení na DA všeobecne metriky ohodnocujúce skúmaný proces
- **Metriky** sa používajú pre kvantifikáciu výkonnosti alebo stavu podniku (procesu)
- **Merania** (namerané hodnoty) sú numerické hodnoty metrík
- V PA rozlišujeme **diskrétné** a **spojité** metriky
- **Diskrétné metriky** zahŕňajú najmä početnosti a proporcne hodnotenia
 - s časovou zložkou alebo bez časovej zložky
 - napr. počet alebo podiel včasných dodávok
- **Spojité metriky** sa merajú na spojitej veličine
 - Čas dodávky
 - Váha balíka
 - Kúpna cena

Atribúty dát a ich podtypy

- Vlastnosti nejakého objektu/procesu/pozorovania/záznamu ... často popisujeme pomocou atribútov = napr. objednávka má viacero atribútov / sledovaných vlastností / metrík / ... (vid'. príklad nižšie)
- Škály hodnôt atribútov popisujúcich objekty (merania o nich) v BA môžu mať rôzny charakter:
 - Kategoriálne (kvalitatívne) atribúty
 - Nominálne alebo ordinálne atribúty
 - Kvantitatívne (numerické) atribúty
 - Intervalové alebo racionálne atribúty
- V rámci jedného objektu (dátovej tabuľky objektov) sú tieto typy často kombinované

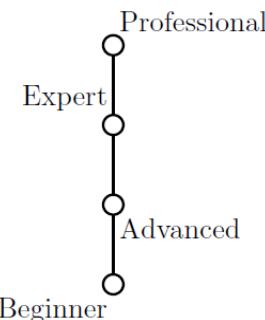
Objednávky									
Dodávateľ	Číslo obj.	Číslo produktu	Popis produktu	Cena/1ks [\$]	Počet ks	Cena objednávky [\$]	Dátum obj.	Dátum dodania	
Firma A	P1204	8523	Produkt 1	\$ 8,00	5	\$ 40,00	12.10.2015	14.10.2015	
Firma B	P1208	21477	Produkt 2	\$ 15,50	4	\$ 62,00	13.10.2015	15.10.2015	
Firma B	P1211	7555	Produkt 3	\$ 10,00	2	\$ 20,00	22.10.2015	22.10.2015	
Firma C	P1213	2564	Produkt 4	\$ 5,00	8	\$ 40,00	27.10.2015	28.10.2015	
.....									
KATEGORIÁLNE				RACIONÁLNE				INTERVALOVÉ	

Nominálne kategoriálne atribúty

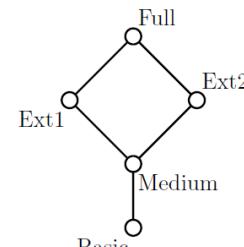
- Kategoriálne (kvalitatívne) atribúty nominálneho charakteru – obsahujú diskrétné (faktorové) hodnoty bez usporiadania
- Hodnoty sú definované v kategóriách vzhľadom k špecifikovanej charakteristike objektov
- Jedna hodnota (kategória) nemá žiadny kvantitatívny vzťah k ďalším hodnotám (kategóriám) – v princípe ide o neporovnateľné prvky z pohľadu usporiadania (vieme rozhodnúť len o tom, či sa dve hodnoty od seba líšia alebo nie, ale nie že je napríklad jedna menšia ako druhá)
- Príklady:
 - Názvy a pomenovania skupín a jednotlivých objektov
 - Lokácia zákazníka (štát, mesto,...), farby (žltá, čierna, zelená, ...)
 - Klasifikácia zamestnanca (manažér, tester, vývojár, ...)
 - Názov firmy, produktu, číslo objednávky (ID atribúty), ...

Ordinálne kategoriálne atribúty

- Kategoriálne (kvalitatívne) atribúty ordinálneho charakteru – diskrétné (faktorové) hodnoty s usporiadaním
- Hodnoty sú definované v kategóriách, pričom tieto sú ohodnotené alebo usporiadane vzhľadom k vzťahu k ďalším kategóriám
- Neexistuje fixná jednotka merania
- Usporiadanie
 - Úplné = všetky hodnoty atribútu sa dajú navzájom porovnať
 - Čiastočné usporiadanie = niektoré hodnoty sa navzájom nedajú porovnať, ale existuje usporiadaná štruktúra
- Príklady:
 - Dotazníkové odpovede v škálach (napr. 5 hodnotová Likertova škála), ranking-y, známky študentov ... úplné usporiadanie
 - Obr. A – úplné usporiadanie – úroveň schopností vodiča automobilu
 - Obr. B – čiastočné usporiadanie – typ poistky, (Ext1, Ext2) sú lepšie ako Medium poistka, ale nedajú sa porovnať navzájom, nevieme povedať že Ext1 je lepšie ako Ext2, obe sú však slabšie ako Full poistka



Obr. A



Obr. B

Vyjádří svou míru souhlasu s výroky týkajících se bydliště.

	Naprosto souhlasím	Spiše souhlasím	Nevím	Spiše nesouhlasím	Naprosto nesouhlasím
a) Lidé z města se nestarají o životní prostředí	1	2	3	4	5

Príklad Likertovej škály

Intervalové kvantitatívne atribúty

- Intervalové dáta – jednotlivé hodnoty sú dobre usporiadane (spojité) s dobre definovaným (konštantným) rozdielom, bez skutočného nulového bodu (inak povedané nula nie je celkom nula)
- Hodnoty takého atribútu nemajú skutočný nulový bod = napr. pri meraní teploty v °C nie je 0 skutočný nulový bod, vieme napríklad nameráť aj -20 °C
- Bez absolútnej nuly však potom pomery hodnôt (a teda násobenie a delenie) nedávajú zmysel – príklady:
 - Ak máme 10°C a 20°C, tak 20°C neznamená 2x teplejšie ako 10°C ... môžeme povedať len že je o 10°C teplejšie
 - Časové údaje (dátum) – ak máme dva dni, vieme vyjadriť len rozdiel (o 10 dní skôr), nie že 2x skôr
- Vieme rozumne pracovať iba s rozdielmi (intervalmi), čiže sčítať a odčítať hodnoty, pomery hodnôt (ratio = podiel) sice mateamticky existujú, ale ich význam je problematický

Racionálne kvantitatívne atribúty

- Racionálne (numerické) atribúty – dobre usporiadane spojité dáta s absolútou nulou (je možné významovo robiť všetky matematické operácie, vrátane podielov – ratio)
- Nulový bod je skutočne nulová nameraná hodnota, nie je možné dosiahnuť menej v danom atribúte ... dôsledkom je rozumná interpretácia podielov hodnôt
- Pomerné hodnoty dávajú význam (napr. 3x väčšia cena objednávky, 2x ľahší balík, 3x kratší čas dodania, atď.)
- Z pohľadu štatistických metód poskytujú priestor na takmer všetky typy analýz
- Príklady:
 - Počet predajov produktu
 - Hmotnosť, dĺžka objektu
 - Celkový čas dodania v dňoch
 - Teplota v Kelvinoch (protože 0° Kelvinovej stupnice je absolútna nula, v tomto prípade 100 K je skutočne 2x teplejšie ako 50 K)

Príklad porovnania typov atribútov podľa možných operácií s hodnotami

Je možné počítať	Nominálny	Ordinálny	Intervalový	Racionálny
Početnosti / frekvencie distribúcie hodnôt	X	X	X	X
Medián alebo percentily		X	X	X
Je možné pripočítať alebo odpočítať hodnoty			X	X
Stred (priemer), smerodajná odchýlka, štandardná odchýlka od stredu, ...			X	X
Pomer hodnôt alebo napr. variačný koeficient (relatívna štandardná odchýlka)				X

Podniková analytika

Rozhodovacie procesy a úvod do
dátových skladov

Obsah

- Rozhodovací proces
 - Rozhodovacie modely
 - Systémy pre podporu rozhodovania (Decision Support Systems - DSS)
 - Vlastnosti DSS
 - Architektúra / základné komponenty DSS
-

- Deskriptívna analytika
- Informačné systémy podniku, úvod od dátových skladov (DS), schéma a základné bloky DS

Rozhodovanie ako proces

- Rozhodovanie je proces výberu medzi 2 a viacerými alternatívnymi akciami za účelom dosiahnutia cieľa (resp. cieľov) operatívneho manažmentu, kontroly, či plánovania.
- Základné fázy procesu rozhodovania:
 - Získavanie informácií (fáza „Intelligence“)
 - Odhalovanie reality (reálneho systému), zber dát, identifikácia a definovanie problému, priradenie vlastníctva problému => Výstup: **Presne definovaný problém**
 - Návrh alternatív (fáza „Design“)
 - Formulácia modelu, určenie kritérií pre výber alternatív, hľadanie alternatív, predikcia očakávaných výsledkov => Výstup: **Alternatívy**
 - Výber riešenia (fáza „Choice“)
 - Výber riešenia problému, analýza citlivosti riešenia, výber alternatívy, plán implementácie výberu => Výstup: **Zvolená alternatíva**
 - Implementácia riešenia + Monitorovanie
 - Aplikácia riešenia na reálny systém, zistovanie úspešnosti (monitorovanie nasadenia riešenia), v prípade potreby návrat k predchádzajúcim krokom (resp. opakovanie cyklu)

Rozhodovanie – získavanie informácií

- Zahŕňa
 - poznanie prostredia (občasné alebo kontinuálne) + monitoring z implementácie
 - identifikáciu a dekompozíciu problému (+ priradenie vlastníka)
- Možné problémy so získavaním dát
 - Neprístupnosť dát, drahý prístup, dáta nie sú (dostatočne) presné, nie sú zabezpečené, dát je veľa, ...
- Možná podpora BI/BA nástrojmi
 - Zdroje dát – interné/externé, dátové sklady + zdroje dát v špecifických systémoch manažmentu, CRM systémy (Customer Relationship Man.), Geografické IS, ... => vyhľadávanie informácií
 - Manažérské informačné systémy, Systémy pre podporu plánovania zdrojov (ERP – Enterprise Resource Planning), Systémy pre podporu man. podnikových procesov (Business Process/Activity Management – BPM/BAM)
 - Expertné systémy – poskytujú odporúčanie pre klasifikáciu jednotlivých prípadov v rámci problému rozhodovania
 - Kolaboratívne nástroje – podpora skupinovej práce

Rozhodovanie – návrh alternatív

- Zodpovedá procesu prípravy modelu
 - Model = zjednodušená reprezentácia alebo abstrakcia reálneho systému
 - Model zachytáva najdôležitejšie vlastnosti systému
 - Môže byť popísaný slovne, vizuálne, matematickým vzorcom, alebo tabuľkou
 - Základnou vlastnosťou DSS a BI/BA systémov je použitie modelu
 - Výhody modelu:
 - Manipulácia s modelom (zmena rozhodovacích premenných, experimentovanie,...) je jednoduchšia
 - Modely umožňujú urýchlenú simuláciu v čase
 - Náklady práce s modelom sú výrazne nižšie ako práca s reálnym systémom + bezpečnejšie z pohľadu chýb
 - Neurčitosť v reálnom systéme môže model odhadnúť lepšie a poskytnúť lepší pohľad na riziko
 - Existujúci model uľahčuje ďalšie rozšírenia, ako aj pochopenie problému, učenie a trénovanie pracovníkov pre prácu na reálnom systéme
 - Konceptualizuje identifikovaný problém do kvantitatívnej a/alebo kvalitatívnej formy

Rozhodovacie modely

- **Rozhodovací model** je model použitý pre pochopenie, analýzu a uľahčenie procesu rozhodovania
 - Proces modelovania = hľadanie vzťahu medzi premennými modelu
- Vstupy rozhodovacieho modelu podľa typu:
 - Dáta
 - Nekontrolované premenné (dané prostredím)
 - Rozhodovacie premenné (kontrolované) => produkujú alternatívy rozhodnutí
- Výstupné premenné rozhodovacieho modelu
 - Výkonnostné merania (meranie = hodnoty rozhodovacích veličín modelu)
 - Behaviorálne merania
- Príklad: Model predaje-reklama
 - V potravinovom priemysle manažéri typicky potrebujú vedieť ako nastavenie ceny, kupónov a reklamných stratégii ovplyvňuje predaje
 - Pomocou prediktívnej BA môžeme vytvoriť model predikujúci predaje podľa cien, kupónov a reklamy ... ten potom môžeme optimalizovať

$$\begin{aligned} \text{Predaje} = & 1000 - 0.04(\text{cena}) + 50(\text{kupóny}) + 0.1(\text{reklama}) + \\ & + 0.2(\text{cena})(\text{reklama}) \end{aligned}$$

Typy rozhodovacích modelov

- Normatívne (preskriptívne) modely
 - Vybraná alternatíva dosahuje najlepší možný výsledok, predstavuje typický príklad výsledku preskriptívnej analýzy
 - Podľa získania modelov sa rozlišuje:
 - Optimalizácia – vyberáme alternatívu prinášajúcu najvyšší level dosiahnutia cieľa pri čo najmenších nákladoch
 - Suboptimalizácia – optimalizácia môže mať problém zahrnúť všetky aspekty, preto sa často niektoré vzťahy zjednodušujú => zjednodušenie modelu a dosiahnutie suboptimálneho výsledku
- Deskriptívne modely
 - Popisujú stav a vzťahy v rámci modelu, nie vždy povedia manažérovi čo urobiť
 - Snaha je popísať rôzne konfigurácie vstupov a procesov => väčšinou vieme takto popísať len časť alternatív => vybraná alternatíva nemusí byť najlepšia (iba postačujúca)
 - Metódy získania modelu:
 - Matematické – simulácia, rozhodovacie siete => generovanie alternatív a prehľadávanie priestoru riešení
 - Nematematické – popisné diagramy a mapy, scenáre – často užitočné ako zdroj pre simulácie a prehľadávanie

Preskriptívne rozhodovacie modely

- Z pohľadu rozhodovacieho procesu ide o normatívny model => cieľom je pomôcť identifikovať najlepšie riešenie
 - Optimalizácia – nájdenie hodnoty rozhodovacích premenných ktoré minimalizujú (alebo maximalizujú) niečo ako náklady (alebo zisk)
 - Cieľová (kriteriálna) funkcia – rovnica ktorá minimalizuje (alebo maximalizuje) požadovanú hodnotu
 - Ohraničenia – obmedzenia alebo reštrikcie na alternatívy
 - Optimálne riešenie – hodnoty rozhodovacích premenných v minimálnom (alebo maximálnom) bode
 - Ak uskutočňujeme optimalizáciu => zabezpečujeme nájdenie najlepšieho riešenia => normatívny optimalizačný model
 - Ak riešime suboptimalizáciu => zabezpečujeme nájdenie „čo najlepšieho dosiahnutelného“ riešenia => normatívny suboptimalizačný model

Preskriptívne rozhodovacie modely (2)

- Rozlišujeme:
 - Deterministické preskriptívne (normatívne) modely – vstupy modelu sú s určitosťou známe
 - Stochastické preskriptívne (normatívne) modely – majú jeden alebo viac vstupov s neurčitosťou
- Pre metódy nájdenie riešenia (najlepšej alternatívy) týchto modelov sa používajú pojmy:
 - Algoritmus – systematická procedúra používaná pre nájdenie optimálneho riešenia rozhodovacieho modelu => prípad optimalizácie
 - Prehľadávací algoritmus (heuristiký) – algoritmus použitý pre hľadanie riešení komplexných problémov bez garancie nájdenia optimálneho riešenia => suboptimalizácia

Rozhodovanie – výber alternatív

- Z modelu návrhu alternatív dostávame množinu pre výber riešenia na implementáciu + (väčšinou) odporúčanie najlepšieho riešenia
- Pre každú alternatívu môžeme
 - Analyzovať životaschopnosť a ziskovosť alternatívy
 - Využiť analytické postupy, algoritmy a prehľadávanie pre zistenie vhodného kandidáta (často prelínajúce sa s fázou návrhu alternatív)
 - Uskutočniť analýzu citlivosti – aké citlivé (robustné) sú alternatívy na zmenu parametrov (dôležité pre implementáciu v reálnom systéme)
 - Realizovať What-If analýzu – ide o exploráciu veľkých zmien parametrov a ich dopadov na špecifické ciele daného procesu rozhodovania

Návrh/Výber – podpora nástrojmi BA/BI

- Pre vývoj modelu je dôležitý prístup k dátam a prostriedky na modelovanie pomocou dát
 - Dátové sklady + OLAP + Data mining
 - Predikčné moduly systémov DSS
 - Podporné: CRM, ERP, KMS (Knowledge Man.Sys., minulé riešenia), SCM + ak sa vyžaduje kolaboratívna torba modelu => kolaboratívne platformy
- Pre fázu výberu
 - Výsledky modelov z Data mining a pod.
 - DSS – poskytujú väčšinou nástroje na analýzu citlivosti a what-if analýzu
 - KMS (znalosti o minulých riešeniacach) + znanosti o podniku (BPM, BAM) + test dopadov rozhodnutí pri riešenom probléme (podľa úlohy: CRM, ERP, SupplyChainMan.-SCM, ...) + kolaboratívne platformy pre konsenzus výberu v skupinovom rozhodnutí

Rozhodovanie – implementácia

- Táto fáza zabezpečuje implementáciu riešenia vybraného vo fáze „Choice“ do reálneho systému
 - Dlhý proces, pričom nemusí íť o počítačovú implementáciu
 - môže sa stretnúť s rôznymi problémami, preto sú:
 - dôležité prvky: presvedčiť o potrebe riešenia, podpora top manažmentu, tréning ľudí na nové veci
 - Veľmi dôležité: zber a analýza dát pre hodnotenie úspešnosti zmeny (monitorovanie a spätná väzba)
 - Podpora nástrojmi BA/BI
 - Reportovacie nástroje + špecifické systémy pre sledovanie činnosti podniku v rôznych doménach podľa potreby (BAM, BPM, SCM, CRM, ERP ...)
 - Expertné systémy + KMS – poradný systém pri problémoch s implementáciou, vrátane popisu podobných príkladov
 - Dáta z monitorovania môžu otvoriť nový cyklus pre nový identifikovaný problém

Systém pre podporu rozhodovania (DSS)

- DSS (Decisions Support System) predstavuje prístup (metodológiu) pre podporu rozhodovania – riešenie problému alebo evaluácia možností
 - interaktívny, flexibilný, počítačový informačný systém pre podporu riešenia neštruktúrovaných manažérskych problémov
 - Pričom používa dátá, jednoduché UI, vlastné pohľady používateľa, modely (vytvorené iteratívne a interaktívne), má často podporu skupinovej práce
- BI systém – monitoruje situácie a identifikuje problémy a možnosti pomocou analytických metód – reportovanie v agregovanej forme hrá hlavnú úlohu
- DSS na rozdiel od BI má vlastné databázy a slúži na riešenie problémov => DSS aplikácie

Schopnosti DSS aplikácií

- Nie je jasný konsenzus nad charakteristikami
- Patria tu:
 - Podpora (rôznych typov a štýlov) procesov rozhodovania (fáz)
 - Podpora semi-štruktúrovaných a neštruktúrovaných problémov, s nezávislými, sekvenčnými aj opakovanými rozhodnutiami
 - Prístup k dátam, modelovanie a analýza
 - Interaktivita, jednoduchosť používania, adaptabilita a flexibilita, efektívnosť
 - Podpora manažérov rôznych úrovní, individuálne alebo v skupinách, kontrola procesu človekom
 - Stand-alone aplikácie, integrované alebo webovské
 - Jednoduchosť vývoja pre koncového používateľa

Klasifikácia DSS

- Klasifikácia systémov je len približná, konkrétny systém nemusí patríť presne do jednej špeciálne = hybridný systém
- AIS SIGDSS (špeciálna skupina venujúca sa problematike DSS) používa klasifikačnú schému:
 - Komunikačne riadené a skupinové DSS
 - Groupware-(y), kolaboratívne platformy, KMS pre komunity – skupinová komunikácia a spolupráca na riešení problému
 - Dátovo riadené DSS
 - DSS aplikácie s dátovými skladmi, OLAP + BPM s reportovaním
 - Dokumentovo riadené DSS
 - KMS – zber a zdieľanie dokumentov o riešených problémoch
 - Znalostne riadené DSS, data mining a expertné systémy
 - UI/znalostné prístupy, inteligentné data mining metódy, pravidlové systémy a ES => automatické DSS (automatizácia procesu rozhodovania)
 - Modelovo orientované DSS
 - Tvorba komplexných modelov, ich simulácií a optimalizácií + what-if analýz
 - Hybridné DSS (kombinácia prvkov 2 a alebo viac predchádzajúcich kategórií)

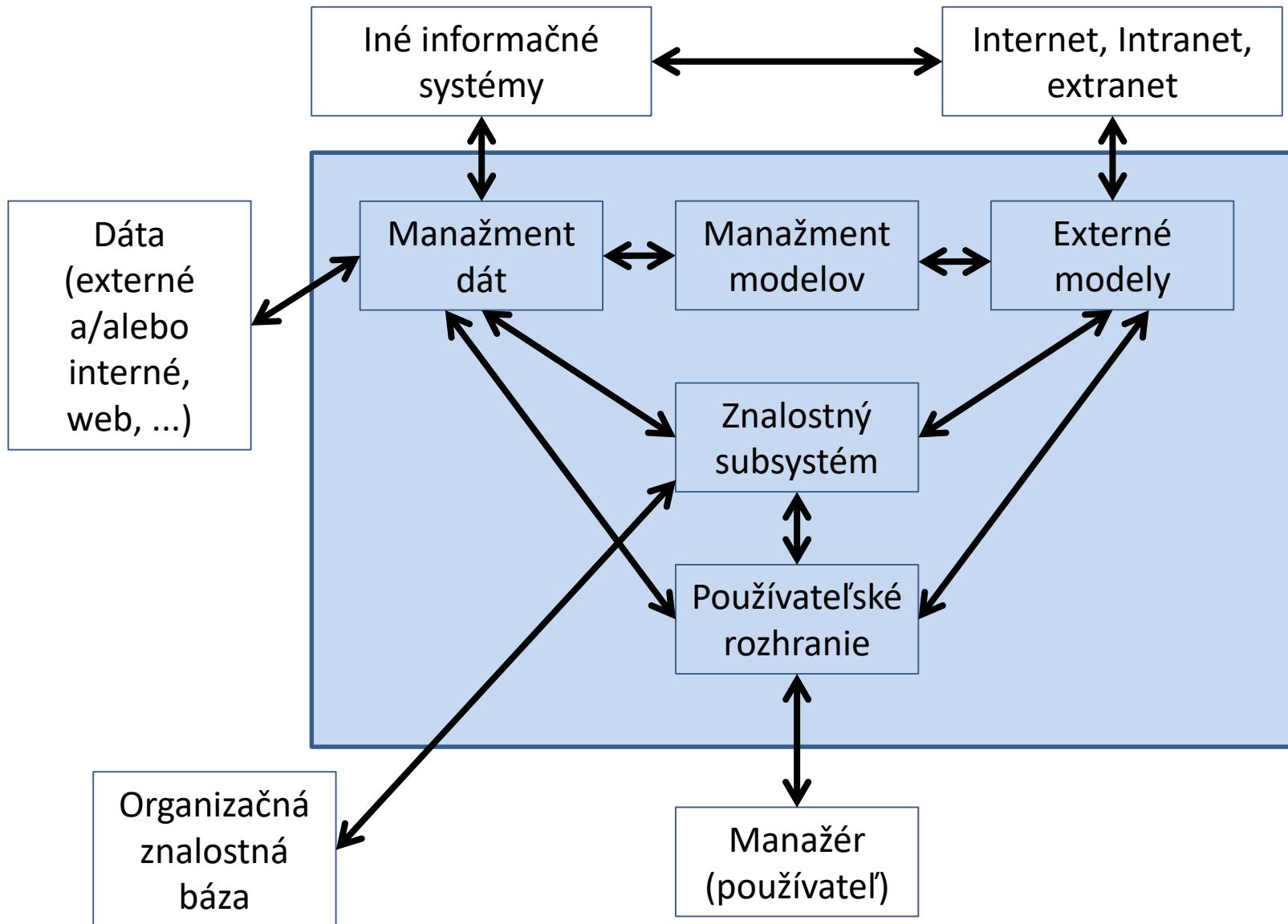
Primárne zameranie:

Deskriptívne prístupy

Prediktívne prístupy

Preskriptívne prístupy

Schéma klasického DSS



Základné komponenty DSS

- Subsystém manažmentu dát
 - DSS databáza, DBMS, dátový adresár a prostriedky pre dopytovanie
 - DBMS manažuje subsystém obsahujúci relevantné dáta k situácii, môže byť napojený na podnikový dátový sklad
- Subsystém manažmentu modelov
 - Adresár modelu, Modelová báza, Manažment modelovej bázy (riadenie + modelovací jazyk + pripojenia), simulačné prostredie
 - Poskytuje subkomponenty pre finančné, štatistické a iné kvantitatívne modelovanie => analytické prostriedky + modelovacie jazyky a riadenie + pripojenie na ďalšie subs.
- Subsystém používateľského rozhrania
 - Podporuje rôznorodé rozhranie pre používateľov, najmä GUI – web rozhrania, dashboard(y) + mobilné rozhrania, ...
- Znalostný subsystém
 - Podporuje ostatné komponenty alebo pracuje ako nezávislý komponent, môže byť napojený na organizačnú znalostnú bázu podniku (časť KMS podniku)

Deskriptívna analytika

- Ciel: Popísat čo sa deje v podniku, sprostredkovať historické dáta, hľadať aktuálne trendy, špecifické situácie, zabezpečiť viditeľnosť dát, ...
- Postupy
 - Dátové sklady a
 - On-line analytika (OLAP)
 - Vizuálna analytika
 - Dashboard-y
 - Reportovanie (metriky)

Informačné systémy v podniku

- Delenie podľa informačnej potreby
 - Operatívne systémy
 - Predstavujú každodenné potreby v prevádzke (faktúry, sklady, objednávky, dáta o výrobe, mzdy, ...)
 - Databázy E/R => operatívne (operačné) databázy – práca s transakciami => **OLTP** (On-Line Transaction Processing)
 - Systémy pre spracovanie transakcií, spracovanie hned' po vložení dát
 - Manažérské systémy
 - Systémy pre podporu rozhodovania a plánovania na úrovni manažmentu => reporty, prehľady, agregované pohľady, ...
 - Databáza obsahuje časovú zložku, summarizácie, agregácie
 - Spracovanie na tejto úrovni sa deje nad špeciálnymi databázami nazývanými dátové sklady cez tzv. **OLAP** (On-Line Analytical Processing) funkcie (len čítanie)

Význam strategických informácií

- V databázach podnikov / organizácií je veľké množstvo dát transakčného typu
 - banky, výrobné podniky, obchody, ...
- Pre manažment je dôležité mať strategické informácie o podniku, explicitne a v dostatočne zrozumiteľnej forme => vyžaduje sa agregácia / sumárne porozumenie dát o transakciách podľa potreby analýzy a návrhu rozhodnutí
 - Cieľom sú dobre viditeľné dáta pre ďalšie úlohy, napr. ako nastaviť ceny, riešiť dodávky, alokovať zdroje, ...
- Vlastnosti strategických informácií
 - Integrácia a integrita dát, Prístupnosť dát, Aktuálnosť dát, Vierohodnosť dát
 - Ako to dosiahnuť vo vhodnej forme pre manažérov?

Nevhodnosť klasických databáz + OLTP

- Pri použití klasických databáz a OLTP (= operačné databázy) je poskytovanie strategických informácií náročné
 - Cieľom E/R databáz + OLTP je najmä ukladanie dát + dopytovanie, sú výhodné na prácu s jednoduchými transakciami, nevhodné pre zložitejšiu analýzu
 - OLTP databázy sú silne decentralizované a nehomogénne, prípadné analýzy sú zložité (rozsiahle výstupy, opakovanie výpočtov = degradácia výkonu, často bez historických dát, neintuitívne nástroje)
 - My však chceme
 - kombinovať dátá z mnohých systémov (databáz)
 - sumarizovať / agregovať dátá
 - poskytovať ich v intuitívnej forme
 - udržiavať takýto proces pre aktuálnosť a integritu, aj vzhľadom na historické dátá
 - analyzovať takéto dátá s cieľom podporiť rozhodovanie a plánovanie
- => Informačné databázy ... Dátové sklady + OLAP

Dátový sklad (DS)

- DS je v podstate štruktúrované úložisko dát - oddelená databáza slúžiaca k podpore rozhodovania + s podporou pre historickú analýzu dát, často vrátane prostriedkov pre prevod transakčných dát na strategické informácie, ich správu a následné použitie = Dátový sklad + OLAP
- **Bill Inmon ... „Dátový sklad je subjektovo orientovaná, integrovaná, časovo variantná (historická) a nemenná (jednoznačná) kolekcia dát (štruktúrované úložisko dát) použitá na získavanie informácií a podporu rozhodovania manažmentu.“**
- **Kimball (funkcionálny pohľad)... „Dátový sklad je kópia transakčných dát špecificky štruktúrovaná pre dopytovanie a analýzu“**
- Proces konštrukcie a použitia dátových skladov = Data Warehousing (dátový sklad = Data Warehouse = DW)

Vlastnosti dátových skladov

- Subjektová orientácia
 - Dáta sú zapisované podľa predmetu záujmu a organizované podľa hlavných subjektov (zákazník, produkt, položka, pobočka, ...)
 - Cieľom je jednoduchý a výstižný pohľad na dátu pre konkrétnu analýzu => nepotrebné dátu pre vylučujeme
 - Orientácia na aplikáciu: ukladanie dát do DS je na základe aplikácie (napr. objednávky, predaje, personalistika, ...)
- Integrovanosť
 - Dáta týkajúce sa konkrétneho predmetu sa ukladajú iba raz => cieľom je jednotná terminológia a jednotky veličín
 - Pre načítanie dát sa vytvára spojenie s heterogénnymi zdrojmi dát – E/R databázy, textové súbory, on-line transakcie, ...
 - Problémom je riešenie nekonzistencia dát => je potrebné zabezpečiť predspracovaním => úprava, čistenie, transformácia, zjednotenie (integrácia) dát, overenie konzistencie názvov premenných, štruktúr a jednotiek

Vlastnosti dátových skladov (2)

- Časová variantnosť (historické dáta)
 - Čas = kľúčový atribút pre analýzu vývoja podnikových operácií, časový horizont DS je zvyčajne dlhší ako pri operačnej databáze
 - Operačná databáza: zvyčajne iba aktuálne dáta alebo kratšie obdobia
 - Dáta v DS: informácie z historickej perspektívy (napr. posledných 5 rokov)
 - Každá dôležitá štruktúra v DS má časovú zložku (explicitnú alebo implicitnú) – pri operačnej databáze kľúč nemusí vždy obsahovať časový element
 - Dáta sú uložené ako séria zaznamenaných stavov, kde jeden stav reprezentuje určitý časový úsek (napr. deň, mesiac, ...)
- Nemennosť (jednoznačnosť)
 - Dôležitou vlastnosťou je fyzické oddelenie transformovaných dát od dát z operačných databáz
 - V DS sa dáta po načítaní (vložení transformačným procesom) už väčšinou nemenia, neodstraňujú, iba pribúdajú ďalšie = LOAD
 - DS má dva základné typy operácií: vloženie dát (LOAD) a prístup k dátam (READ)
 - Vzhľadom k tomuto zjednodušeniu nie je potrebné riešiť spracovanie transakcií, zotavenie, mechanizmy riadenia paralelného prístupu, optimalizácia a normalizácia sa riešia jednoduchšie, výsledok dopytu je jednoznačný (nemenný)

DS verzuš Operačná databáza

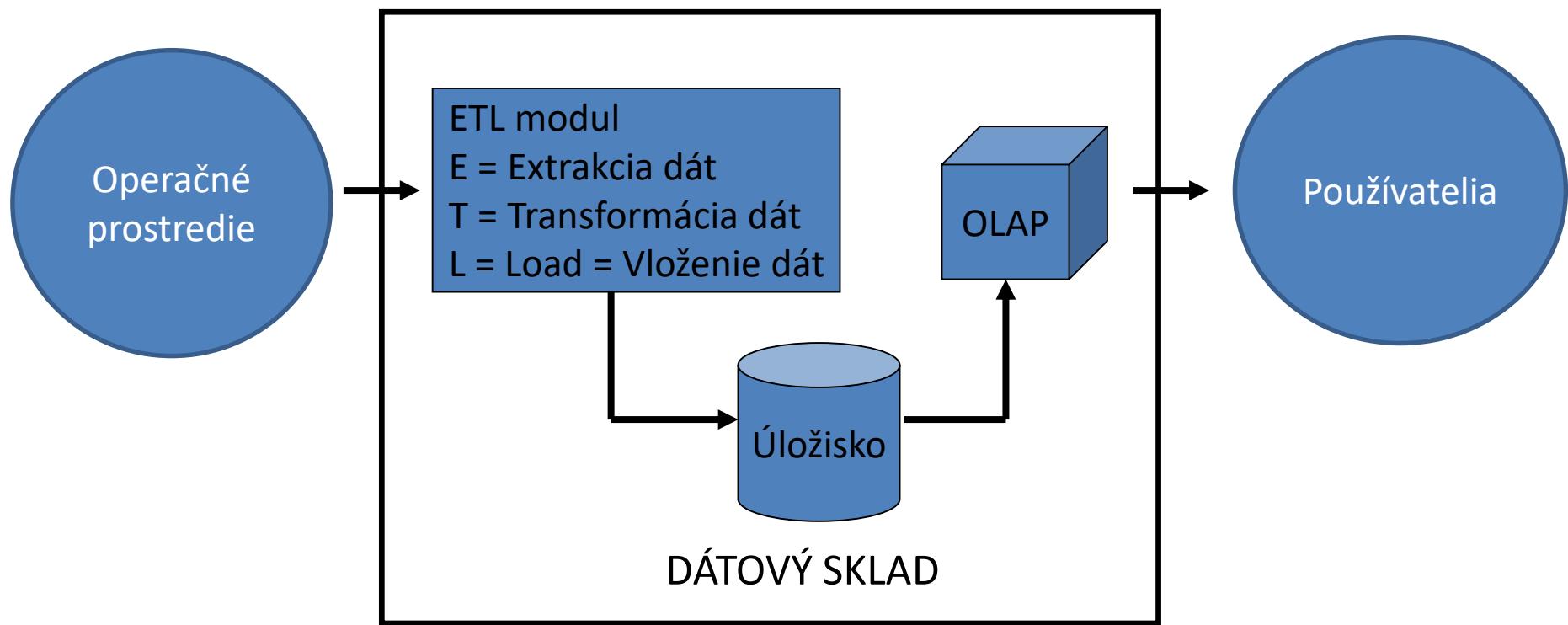
	DS + OLAP	Operačná DB + OLTP
Hlavná úloha	Analýza dát a podpora rozhodovania	Každodenné operácie = transakcie
Činnosti	Analýza a rozhodovanie	Operatívne procesy
Pôvod dát (čas)	Snímky za časové úseky	30-60 dní
Zdroje dát	Operačné, interné, externé	Operačné, interné
Obsah dát	Historický a agregovaný	Aktuálny a detailný
Pohľad na dátu	Evolučný a integrovaný	Aktuálny a lokálny
Organizácia dát	Podľa subjektov a času	Podľa aplikácie
Model návrhu	Napr. hviezdicová schéma + subjekt	E/R model + aplikácia
Operácie	Len zápis dát pri vkladaní a čítanie	Podľa databázového jazyka = SQL
Princíp prístupu	Komplexné dopyty	Jednoduché transakcie
Prístupné záznamy	milióny	desiatky – stovky
Používateľia	Analytik, manažér (max 100-vky)	DB špecialisti, úradníci (1000-ky)
Použitie	Ad-hoc	Opakovane
Veľkosť dát	Veľké až veľmi veľké (cca 100 GB-TB)	Malé až veľké (cca 100 MB-GB)
Rýchlosť odozvy	sekundy až hodiny	milisekundy až sekundy
Metriky výkonnosti	Časová odozva	Priepustnosť transakcií

Požiadavky na DS

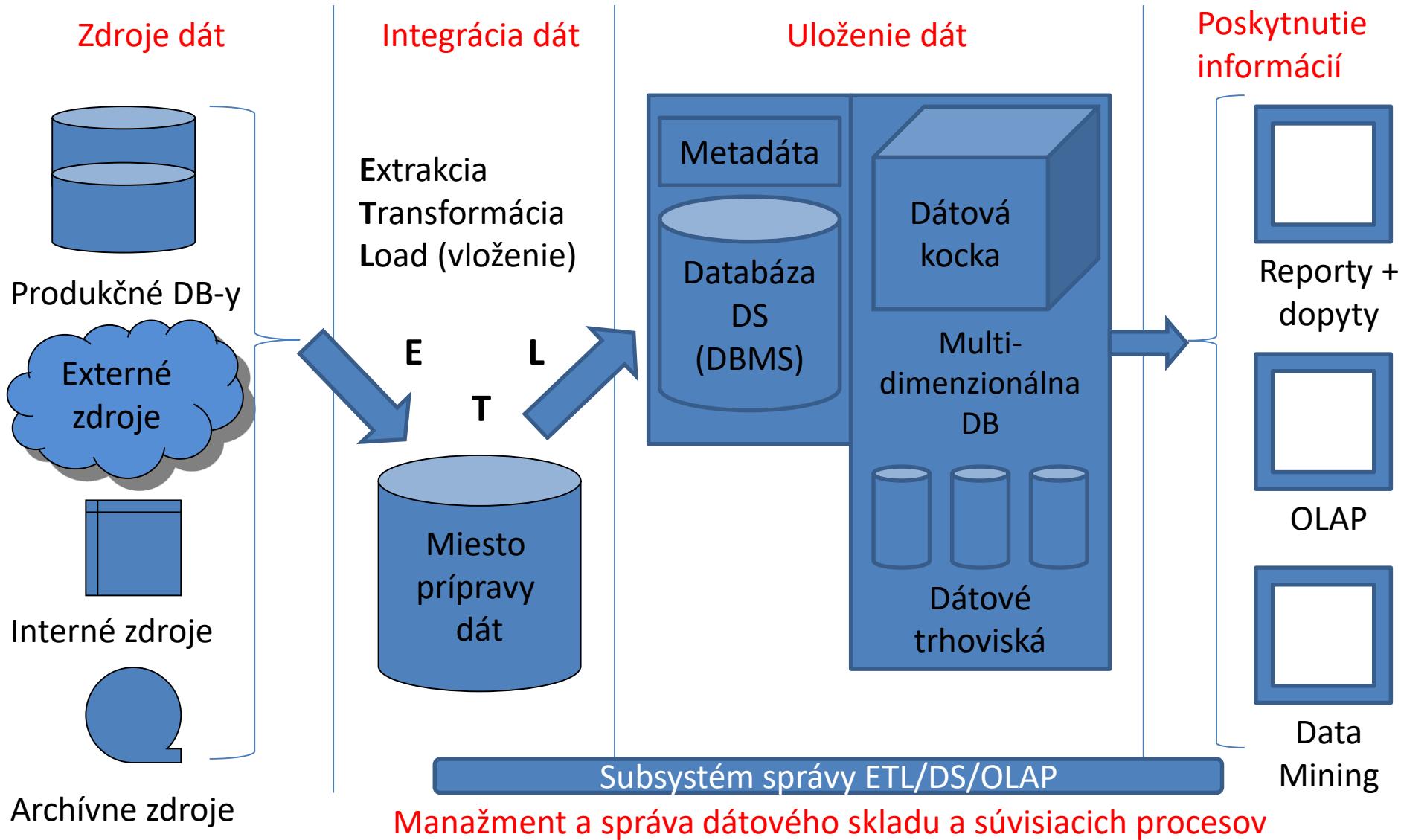
- Databáza by mala byť navrhnutá pre komplexné analytické dopyty
- Musí mať možnosť integrovať dátá z viacerých aplikácií v podniku (z databáz, interných aj externých zdrojov)
- Podporuje efektívne časté operácie čítania z databáz
- Rozhranie by malo byť interaktívne, jednoduché a dlhodobo použiteľné aj bez pomoci IT konzultanta
- Musí poskytovať načítanie dát po určenej časovej perióde
- Mal by poskytnúť možnosť použitia aktuálnych aj historických dát
- Mal by umožňovať spustenie dopytu a získanie výsledkov on-line
- Mal by dávať možnosť získať ľubovoľnú výstupnú (tlačovú) zostavu

Základná schéma DS

- Predstavuje vlastne modelovanie dátového toku v podniku
 - ETL (Extract-Transform-Load) = získanie, transformácia, vloženie (načítanie) dát do databázy (úložiska) DS
 - OLAP – sprístupnenie DS používateľom



Komponenty DS + dátový tok



Zdroje dát a miesto prípravy dát

- Zdroje dát
 - Produkčné dáta – dáta získané z rôznych operačných DB podniku pomocou jednoznačných dopytov
 - Interné dáta – dáta uložené v privátnych súboroch (napr. XLS) zamestnancov podniku (organizácie)
 - Archívne dáta – jeden zo základných predpokladov úspešnej analýzy, zväčša veľké množstvo uložených „raw“ dát
 - Externé dáta – dáta z rôznych zdrojov, ktoré môžu byť užitočné pre podnik a cieľové analýzy
- Miesto prípravy dát
 - Fyzické miesto, kde prebieha príprava dát – tzv. fáza ETL – ktorá predstavuje medzistupeň medzi vstupnými dátami a DS
 - Môže byť súčasťou DS
 - Ide o miesto špeciálne určené na túto úlohu
 - Výstup: dáta pripravené pre analýzu, ktoré je možné uložiť do dátového skladu

Uloženie a poskytnutie informácií

- Uloženie dát = jadro DS
 - Oddelené úložisko pre uloženie veľkého množstva najmä historických dát
 - Navrhnuté pre analýzu, nie pre rýchly prístup k dátam
 - S výnimkou administrátora je prístup iba na čítanie
 - Úložisko a odpovedajúce časti musia byť prístupné pre viac druhov nástrojov => musia existovať odpovedajúce rozhrania
- Poskytnutie informácií
 - Často realizované ako jeden modul rozhrania pre poskytovanie informácií z DS v rôznych formách
 - Rozhrania na rôzne výstupy:
 - Pre začínajúcich používateľov: tlačové zostavy, jednoduché dopyty
 - Bežní používatelia: štatistická analýza, rôzne vizualizácie dát a reportovanie, preddefinované dopyty
 - Pokročilí používatelia: multi-dimenziorná analýza, zadáva vlastné OLAP dopyty, používa/napĺňa EIS systémy (data mining, ...)

Metadáta v DS a ich základné typy

- Metadáta predstavujú „dáta popisujúce dáta“
 - Informácie o dátových štruktúrach, súboroch, adresách (dátový slovník)
 - Informácie o dátach v databáze (tzv. katalóg)
- Metadáta sú jednou z dôležitých častí DS
- Metadátové zdroje v procese / typy metadát
 - Operačné (operatívne) metadáta
 - Obsahujú informácie o všetkých zdrojoch dát pre dátový sklad (aká je ich štruktúra, umiestnenie, ...)
 - ETL metadáta
 - Informácie o tom, ako bola realizovaná ETL fáza – aké metódy boli použité pri extrakcii, transformácii a vložení dát do dátového skladu, aké boli problémy / obmedzenia pri procese, ...
 - Metadáta pre koncových používateľov
 - Informácie o dátovom sklade a uložených dátach, rozhraniach na nich, ako aj ďalšie obchodné a iné informácie využiteľné v analýze

Obsah metadát

- Subsystém Metadáta predstavuje priestor pre uloženie dát popisujúcich objekty dátového skladu, t.j., sú v ňom uložené:
 - Popis štruktúry DS
 - Schéma, dimenzie, hierarchie, umiestnenie a obsah dátových trhovísk (Data Mart = dátové trhovisko)
 - Operačné metadáta
 - História (pôvod) dát, monitorovacie informácie (štatistiky, chyby, ...), stav dát (či sú archívne, aktuálne)
 - Algoritmy používané pre summarizáciu / agregáciu
 - Mapovanie z operačného prostredia do DS
 - Dáta týkajúce sa činnosti systému DS
 - Schéma skladu, odvodené dáta
 - Obchodné údaje
 - Definície obchodných pojmov, vlastníci dát, ...

Subsystém pre správu a manažment DS

- Táto časť je z technického pohľadu nadradená všetkým ostatným časťiam dátového skladu
- Jeho úlohou je koordinácia (riadenie) jednotlivých zložiek dátového skladu a im odpovedajúcich procesov
- Pre beh dátového kladu používa informácie uložené v metadátovej časti úložiska
- Je spravovaný administrátorom systému dátového skladu
- Najdôležitejšie funkcie subsystému
 - Monitorovanie všetkých operácií s dátovým skladom
 - Ošetrenie chýb procesov a zotavenie sa z nich
 - Extrakcia dát zo zdroja pre účely aktualizácie dátového skladu (resp. riadenie tohto často periodického procesu)
 - Kontrola správnosti transformácie dát
 - Zabezpečenie správnych funkcií pri poskytovaní informácií
 - Bezpečnosť dát a autorizácia používateľov

Podniková analytika

Dátové sklady - dokončenie

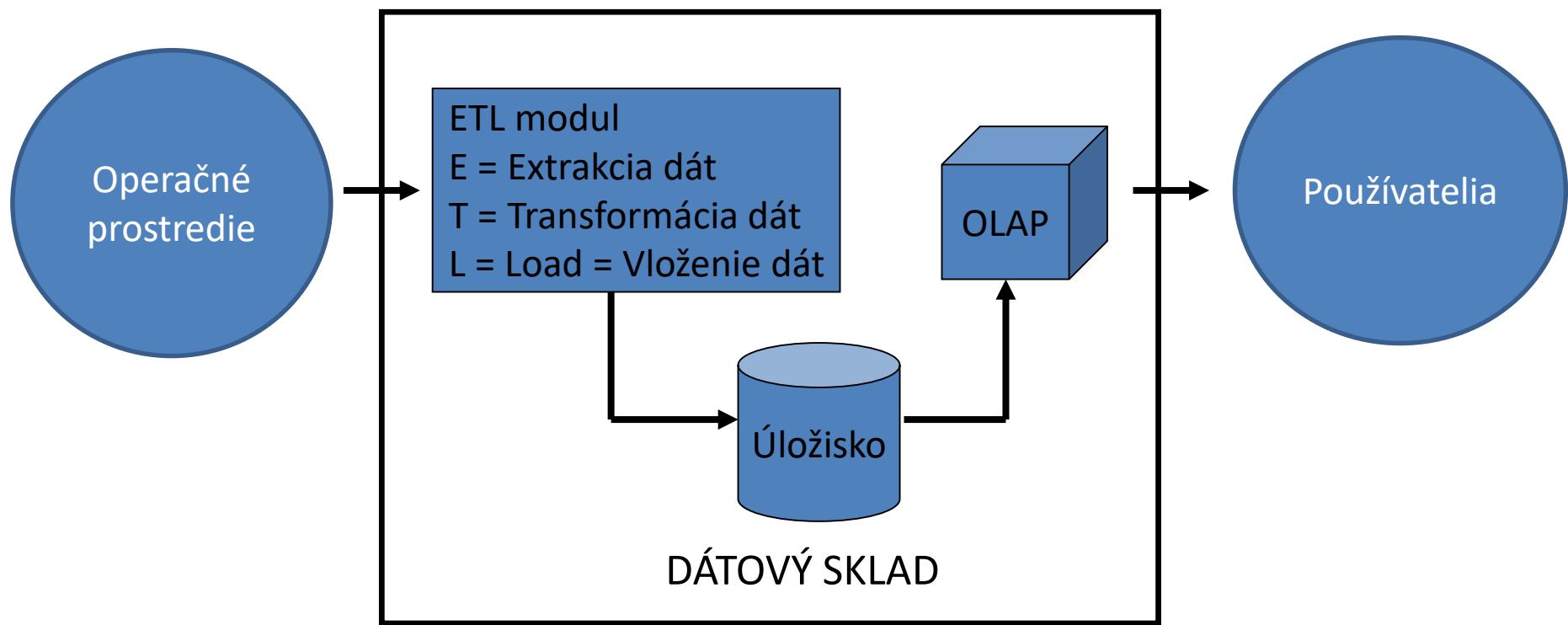
Obsah

- vid'.Prednáška2 – Úvod do dátových skladov (DS)
-

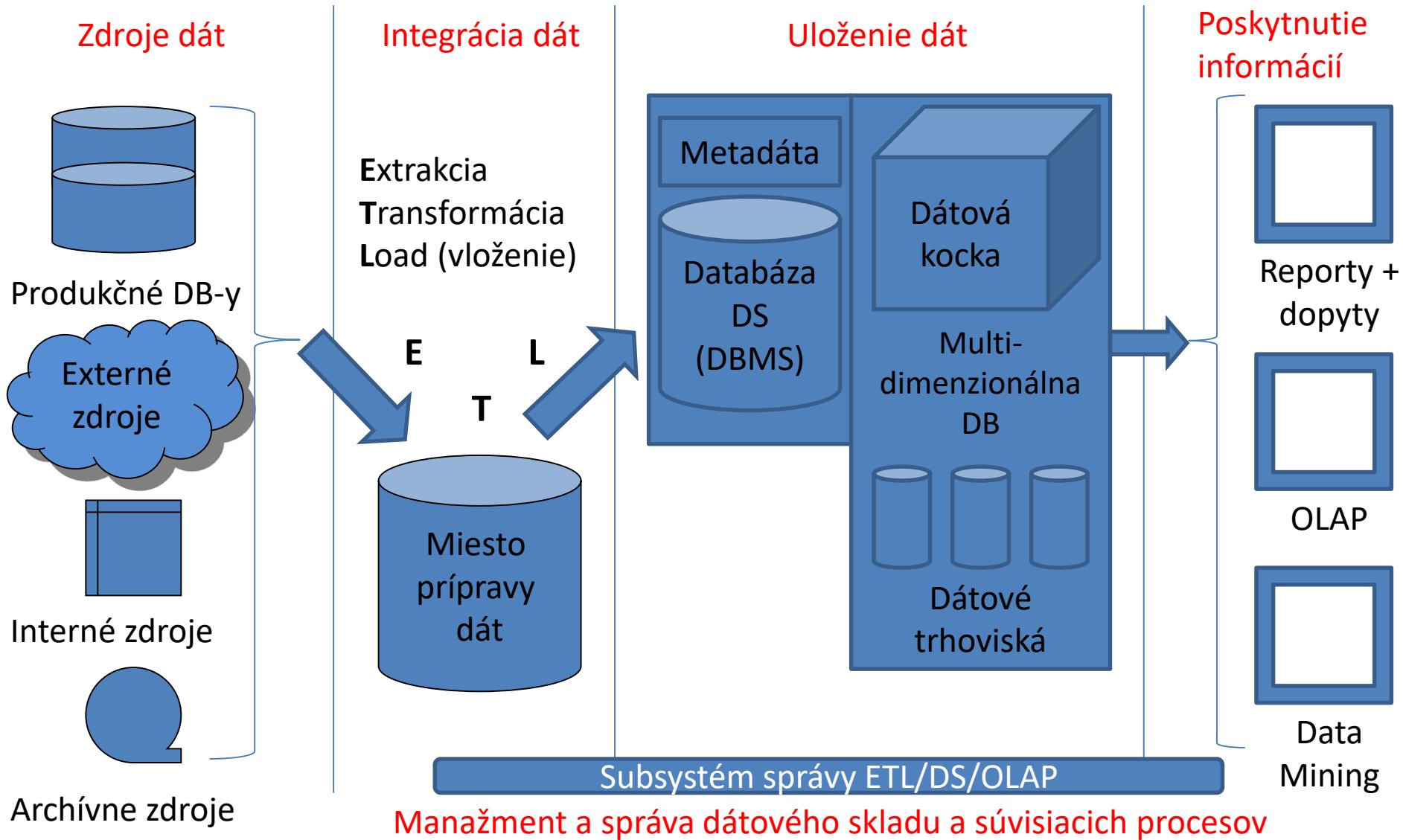
- Modelovanie dát v DS
- Multidimenzionálne modelovanie – dátová kocka
- OLAP funkcie
- Rôzne verzie DS/OLAP systémov

Základná schéma DS

- Predstavuje vlastne modelovanie dátového toku v podniku
 - ETL (Extract-Transform-Load) = získanie, transformácia, vloženie (načítanie) dát do databázy (úložiska) DS
 - OLAP – sprístupnenie DS používateľom



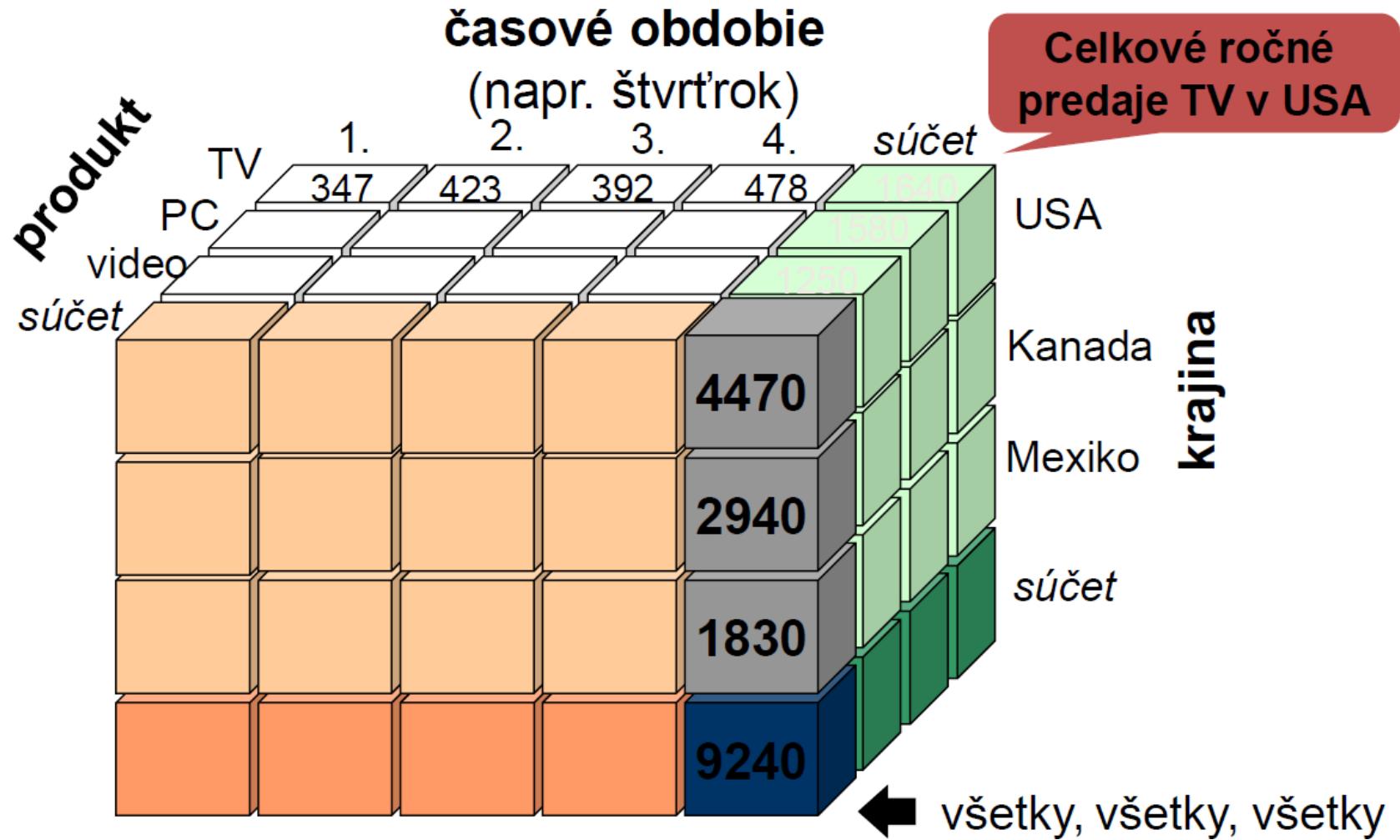
Komponenty DS + dátový tok



Multi-dimenzionálny model dát

- Dátový sklad je veľmi často modelovaný ako **multidimenzionálna databáza**
 - Slúži ako základ pre získanie summarizovaných a agregovaných dát, pričom mnohé výpočty sú pre zrýchlenie predpočítané
 - Často obsahuje nenormalizované tabuľky = cieľom je rýchlosť
- Multidimenzionálny model reprezentuje dáta ako **dátovú kocku**, kde
 - **Dimenzia** je atribút (alebo ich množina) v schéme, napr. región, produkt, čas
 - **Bunka** obsahuje hodnotu agregovanej veličiny (alebo viacerých veličín) – počet, priemer, súčet, ...
- Výhody MD modelu
 - Rýchly prístup k dátam, možnosť komplexných analýz, pohľad na dáta z rôznych úrovní abstrakcie, dobrý vstup pre predikciu
- Nevýhody MD modelu
 - Vyššie nároky na kapacitu úložiska DS z dôvodu redundancie dát, pri zmene dimenzie nutnosť prepočítať agregované dáta

Príklad jednoduchej dátovej kocky

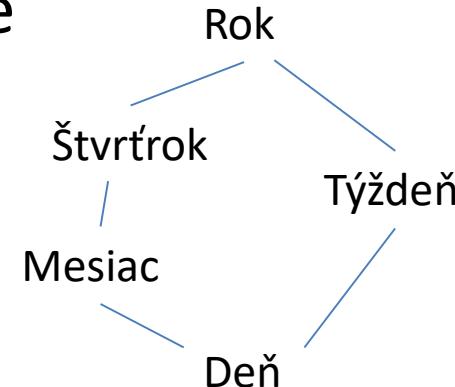
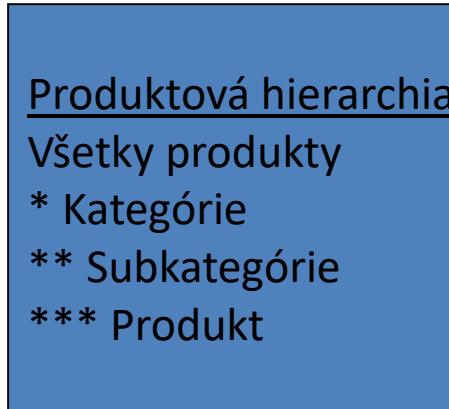


Dátová kocka

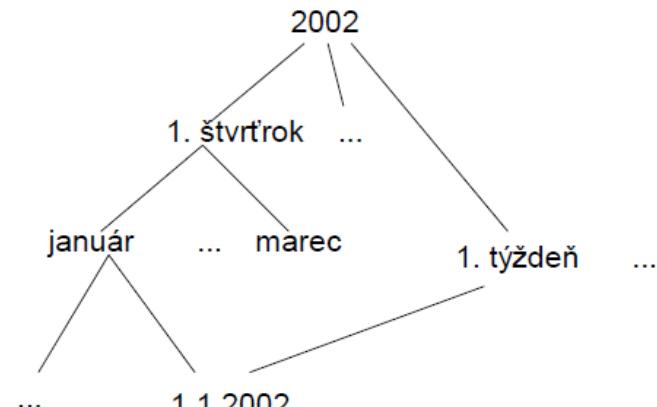
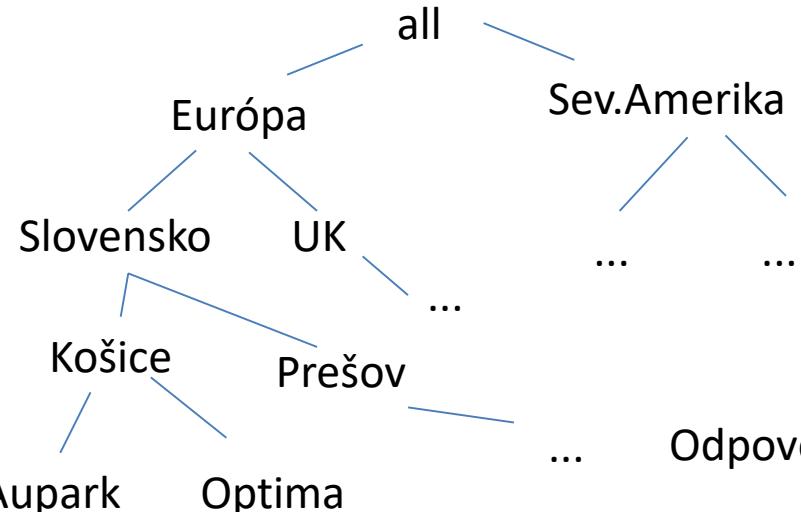
- Dátová kocka (Data Cube) = multidimenzionálny model DS – má dva typy atribútov:
 - Dimenzie – nezávislé atribúty (napr. zákazník, pobočka, časová zložka)
 - Fakty (veličiny) – závislé atribúty – numerické atribúty (napr. počet, suma predajov, zisk) = hodnoty pre danú bunku v kocke
- Tabuľka faktov
 - Je zvyčajne len jedna, najväčšia v DS, obsahuje namerané hodnoty (veličiny), v kombinácii s tabuľkami dimenzií vytvára schému systému DS
- Dimenzie určujú viacozmerný dátový priestor s bunkami pre ktoré sú v tabuľke faktov uložené fakty (veličiny) pre príslušnú vymedzenú oblasť. Každá dimenzia:
 - Je definovaná tabuľkou dimenzie, vo všeobecnosti je definovaná kombináciou viacerých atribútov (položka má meno, kód, typ, ...)
 - Ak je viac atribútov => sú obvykle organizované do hierarchie
 - Všetky možné kombinácie hodnôt jednotlivých atribútov vytvárajú stromovú štruktúru, tzv. = hierarchiu konceptov

Príklady hierarchických atribútov

- Dimenzie - veľmi častá kombinácia: časové, geografické a produktové dimenzie



Dobre (úplne) usporiadane hierarchie



Odpovedajúce príklady hierarchií konceptov

Základné schémy tabuliek MDB

Existujú tri základné schémy multidimenzionálnej DB - MDB (dátovej kocky):

- Hviezdicová schéma – obsahuje jednu tabuľku faktov v strede, na ktorú sú vždy priamo napojené tabuľky dimenzií
- Vločková schéma – obdoba hviezdicovej schémy, avšak niektoré tabuľky dimenzií v normalizovanej podobe => sú rozdelené do ďalších tabuliek
- Súhvezdie faktov – existuje viacero tabuliek faktov, ktoré zdielajú tabuľky dimenzií

Príklad hviezdicovej schémy

Časová dimenzia

time_key
day
week_of_the_year
month
quarter
year

Tabuľka faktov

time_key
item_key
location_key
units_sold
dollars_sold
avg_sales

Produktová dimenzia

item_key
item_name
subcategory
category
supplier_name
supplier_type

Veličiny (obsah buniek dátovej kocky)



Rozšírenia:

- Vločková schéma – normalizácia tabuľiek dimenzií, napr. dodávateľia (supplier), alebo mestá v samostatných tabuľkách dimenzií (cez kľúč pripojené na pôvodné)
- Súhvezdie faktov – ďalšia tabuľka faktov s príslušnými napojeniami na tabuľky dimenzií (napr. tabuľka faktov dodávky + napojenia na časovú, produktovú a geografickú dimenziu)

Geografická dimenzia

location_key
branch_name
address
city
country
continent

Fáza ETL

- Hlavný cieľ = centralizácia dát
 - Dôležité pre zabezpečenie kvalitných dát v DS
 - Nikdy nekončiaci proces (nutnosť aktualizácie)
- Základné prvky
 - Extrakcia – výber dát rôznymi metódami
 - Transformácia – overenie, čistenie, integrácia, časové označenie dát
 - Vloženie (Loading) – presun dát do DS
- Hlavné úlohy ETL procesu
 - Určiť interné/externé zdroje dát + ktoré budú uložené v DS
 - Príprava mapovania medzi zdrojovými a cieľovými dátami
 - Určenie pravidiel pre extrakciu, transformáciu a čistenie dát
 - Plán pre agregáciu tabuľiek
 - Návrh oblasti (miesta) pre prípravu dát
 - Príprava procedúry pre nahrávanie (vloženie) dát do DS
 - ETL pre tabuľky dimenzií a faktov (prvky schémy)

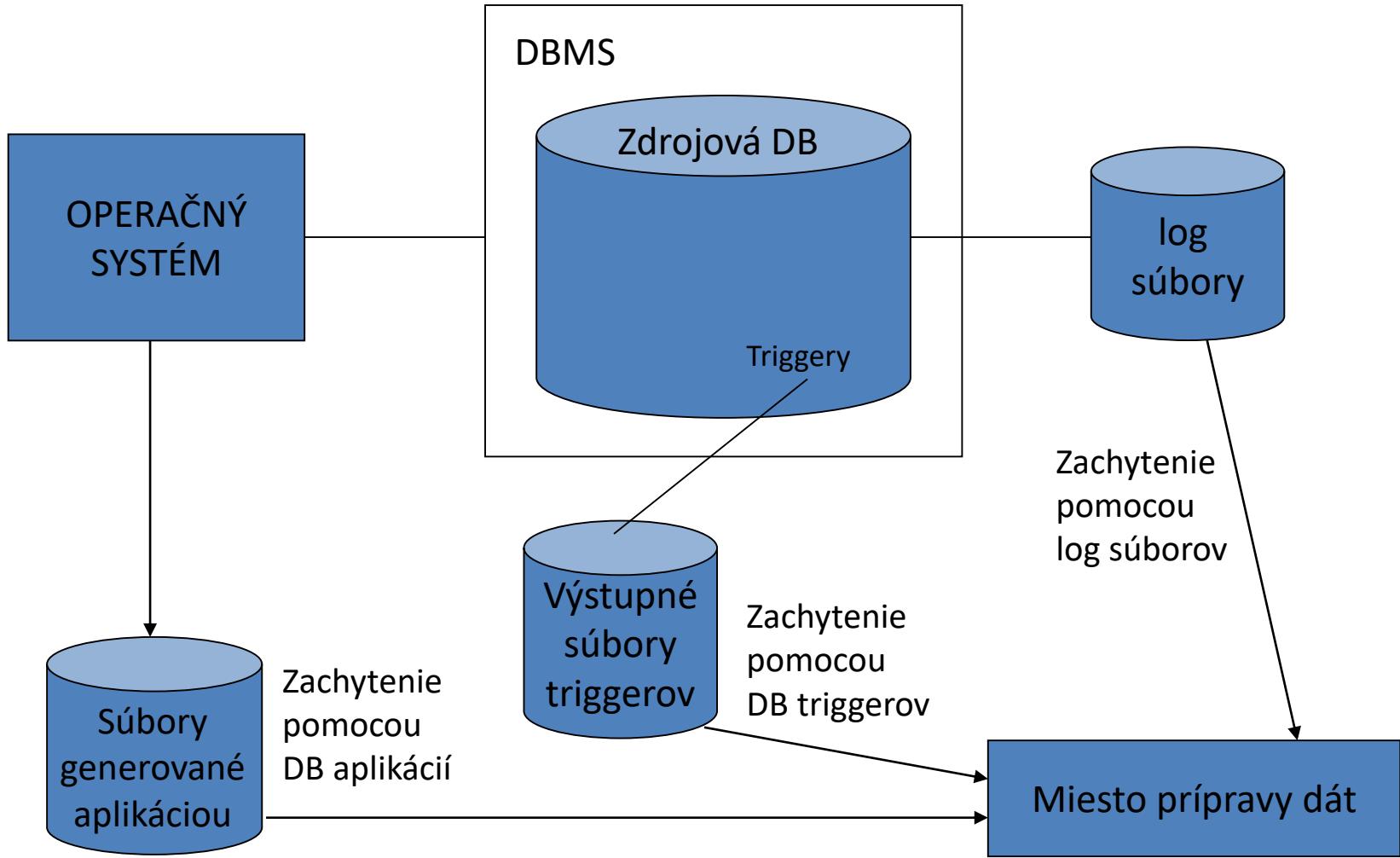
Extrakcia dát

- Zdrojom dát z nehomogénneho prostredia
- Rôzne možnosti extrakcie
 - Periodická extrakcia – z interných zdrojov
 - Občasná extrakcia – z externých zdrojov (Internet)
 - Prvá extrakcia – najmä z archívnych dát
- Extrakcia ako proces
 - Identifikácia zdrojov
 - (Postup) Výpis položiek pre tabuľku faktov, výpis dimenzií, nájdenie zdrojov a položiek pre cieľovú položku v DS (ak je ich viac => výber preferovaného zdroja), identifikácia viacnásobných zdrojov pre 1 cieľ (konsolidačné pravidlá), identifikácia viacnásobných cieľov pre 1 zdroj (deliace pravidlá), určenie implicitných hodnôt, zistenie chýbajúcich hodnôt v zdrojoch
 - Určenie metódy extrakcie – 1. Vlastné SQL, 2. cez nástroje
 - Určenie frekvencie extrakcie pre každý zdroj
 - Určenie časového okna pre extrakciu
 - Paralelná alebo sériová extrakcia (pre každý zdroj)
 - Spracovanie výnimiek

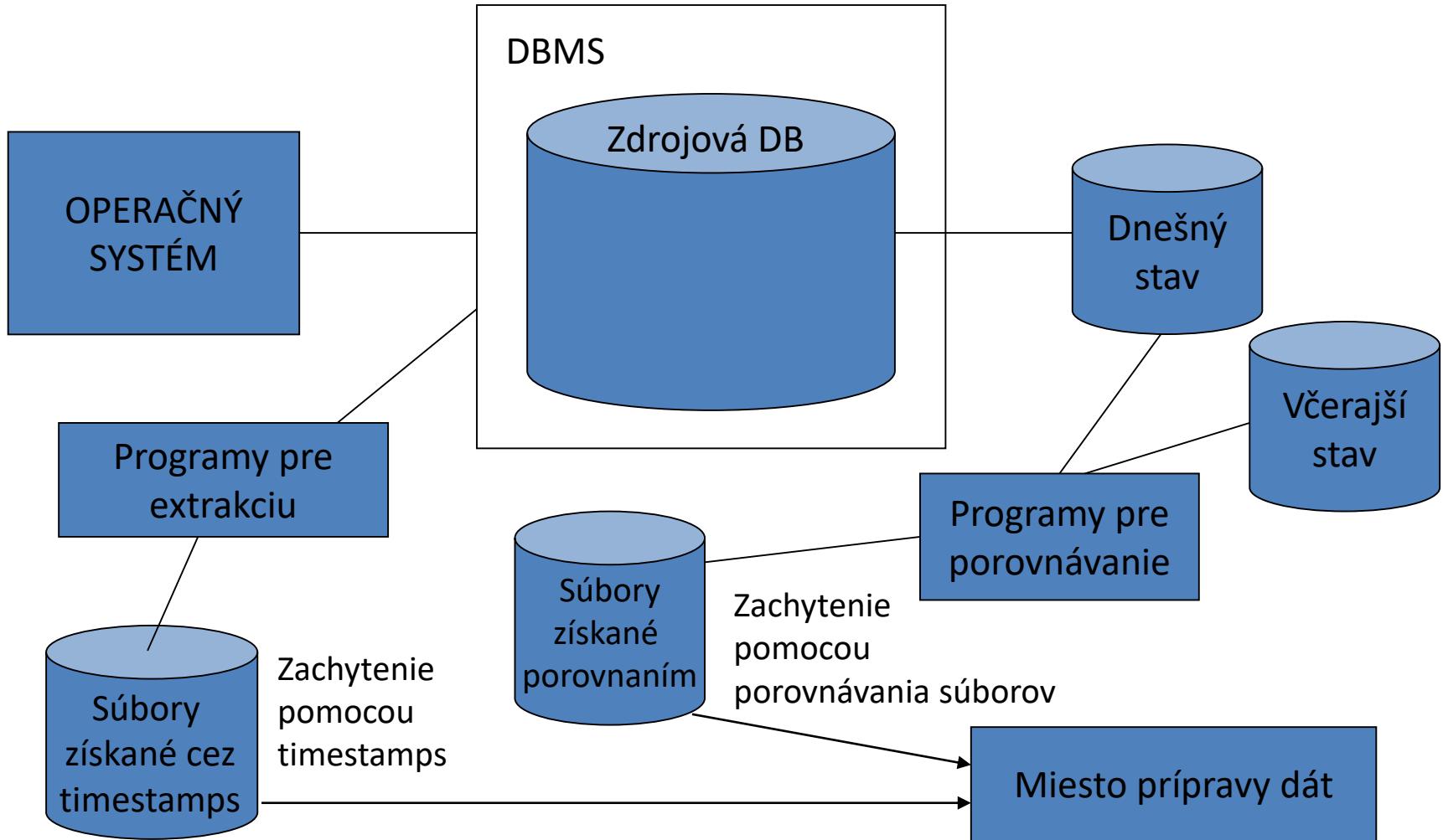
Metódy extrakcie

- Metóda extrakcie statických dát
 - Vytvorenie obrazu zdrojovej DB na výstupe, najmä pre iniciálne vloženie dát
- Metódy extrakcie pri aktualizácii dát
 - 2 základné prístupy – odlišujú sa spôsobom zachytenia zmien v DB od posledného vloženia dát
 - Metódy priamej extrakcie
 - Využívajú log súbory a databázové triggery
 - Zachytenie sa deje aj pomocou samotných DB aplikácií, editácia aplikáciou uloží záznamy o zmenách
 - Metódy odloženej extrakcie
 - Nezachycujú sa zmeny pri vzniku, ale porovnávaním
 - Používajú sa tzv. timestamp (časové známky) alebo priame porovnávanie obsahu dát z rôznych období (neefektívne)

Metódy priamej extrakcie



Metódy odloženej extrakcie



Transformácia dát

- Cieľ: zvýšiť kvalitu vstupných dát – kvalita je často premenlivá => čistenie dát, odstránenie anomálií + úprava dát podľa potreby
- Časté problémy
 - Nejednoznačnosť údajov – napr. rôzne uložené dáta o pohlaví (M, muž, Muž, ...)
 - Chýbajúce hodnoty – treba ich doplniť, ignorovať alebo nejako označiť
 - Duplicítne hodnoty – odstránenie (niekedy časovo náročné)
 - Konvencie názvov pojmov a objektov – zjednotenie terminológie medzi heterogénnymi zdrojmi dát
 - Rôzne peňažné meny – napr. prechod SKK na EUR
 - Formáty čísel a textových reťazcov
 - Referenčná integrita – neustále zmeny v reálnom svete skresľujú dátá (napr. po zrušení pobočky zostanú údaje o zamestnancoch v DB)
 - Chýbajúce dátumy – časová dimenzia je dôležitá, ale v zdrojoch často chýbajú, je potrebné ich doplniť

Typické úlohy a typy transformácie dát

- Typické úlohy
 - Selekcie (výber vhodných atribútov pre cieľový DS)
 - Rozdeľovanie / Spájanie (rozdelenie záznamov, spájanie z rôznych zdrojov)
 - Konverzie záznamov (štandardizácia => lepšie pochopenie dát)
 - Sumarizácie (je vhodnejšie sumarizovať / agregovať detailné dáta)
 - Obohatenie (vytvorenie lepšieho pohľadu na dáta na základe rôznych zdrojov)
- Hlavné typy transformácií
 - Výpočet odvodených polí – summarizácie
 - Zjednotenie prístupu k obsahu dát – Rozdelenie polí na časti (meno a priezvisko), zlúčenie informácií (informácie o produkte z viacerých tabuľiek), odstránenie duplikácií
 - Formátovacie typy konverzií – Revízie formátu dát (zjednotenie formátu ukladania), dekódovanie polí (viď. rôzne kódovanie pohlavia), konverzie znakových štandardov, konverzie merných jednotiek, dátumov a časov
 - Zjednotenie štruktúry klúčov

Vloženie dát do DS

- Presun dát a ich uloženie v DS, ideálne automaticky, s rôznou periódou presunu
- Typy vloženia dát do DS
 - Iniciálne vloženie (všetky dáta od prázdnego DS)
 - Inkrementálne vloženie (premietne zmeny v DB do DS – vykonávane periodicky)
 - Prepínanie dát – kompletné zmazanie obsahu DS a vloženie aktuálnych dát
- Módy nahrávania dát do DS
 - Load (Nahratie / Vloženie) – ak cieľová tabuľka obsahuje dáta, tieto sú vymazané a nahradené aktuálnymi
 - Append (Pridanie) – pridanie nových dát ku existujúcim, pri duplike si používateľ môže zvoliť ďalší postup
 - Deštruktívne zlúčenie – pridanie, pričom pri rovnakých kľúčoch sa prepisuje hodnota daného riadku
 - Konštruktívne zlúčenie – pri rovnakých kľúčoch sa pridáva nový prvok a označí sa ako nový, starý v DS zostane

OLAP systémy

- Realizujú dopytovanie sa nad dátovou kockou (MDB) => manuálne dolovanie v dátach => myšlienkový pochod analytika sa transformuje do operácií s dátovou kockou => interaktívna práca s dátami cez OLAP sa nazýva aj OLAP Session
- E.F. Codd => „**OLAP je voľne definovaný súbor princípov, ktoré poskytujú dimenzionálny rámec pre podporu rozhodovania**“
- OLAP systémy umožňujú pracovníkom zodpovedným za rozhodnutia prístup k strategickým informáciám potrebným pre tvorbu rozhodnutí
- Požiadavky na OLAP systémy
 - Codd definoval 12 pravidiel OLAP
 - Tieto boli doplnené ďalšími

Príklad jednoduchej OLAP Session

LINE	TOTAL SALES
Clothing	\$12,836,450
Electronics	\$16,068,300
Video	\$21,262,190
Kitchen	\$17,704,400
Appliances	\$19,600,800
Total	\$87,472,140

1

High level summary by product line

2

Drill down by year

3

Rotate columns to rows

LINE	1998	1999	2000	TOTAL
Clothing	\$3,457,000	\$3,590,050	\$5,789,400	\$12,836,450
Electronics	\$5,894,800	\$4,078,900	\$6,094,600	\$16,068,300
Video	\$7,198,700	\$6,057,890	\$8,005,600	\$21,262,190
Kitchen	\$4,875,400	\$5,894,500	\$6,934,500	\$17,704,400
Appliances	\$5,947,300	\$6,104,500	\$7,549,000	\$19,600,800
Total	\$27,373,200	\$25,725,840	\$34,373,100	\$87,472,140

YEAR	Clothing	Electronics	Video	Kitchen	Appliances	TOTAL
1998	\$3,457,000	\$5,894,800	\$7,198,700	\$4,875,400	\$5,947,300	\$27,373,200
1999	\$3,590,050	\$4,078,900	\$6,057,890	\$5,894,500	\$6,104,500	\$25,725,840
2000	\$5,789,400	\$6,094,600	\$8,005,600	\$6,934,500	\$7,549,000	\$34,373,100
Total	\$12,836,450	\$16,068,300	\$21,262,190	\$17,704,400	\$19,600,800	\$87,472,140

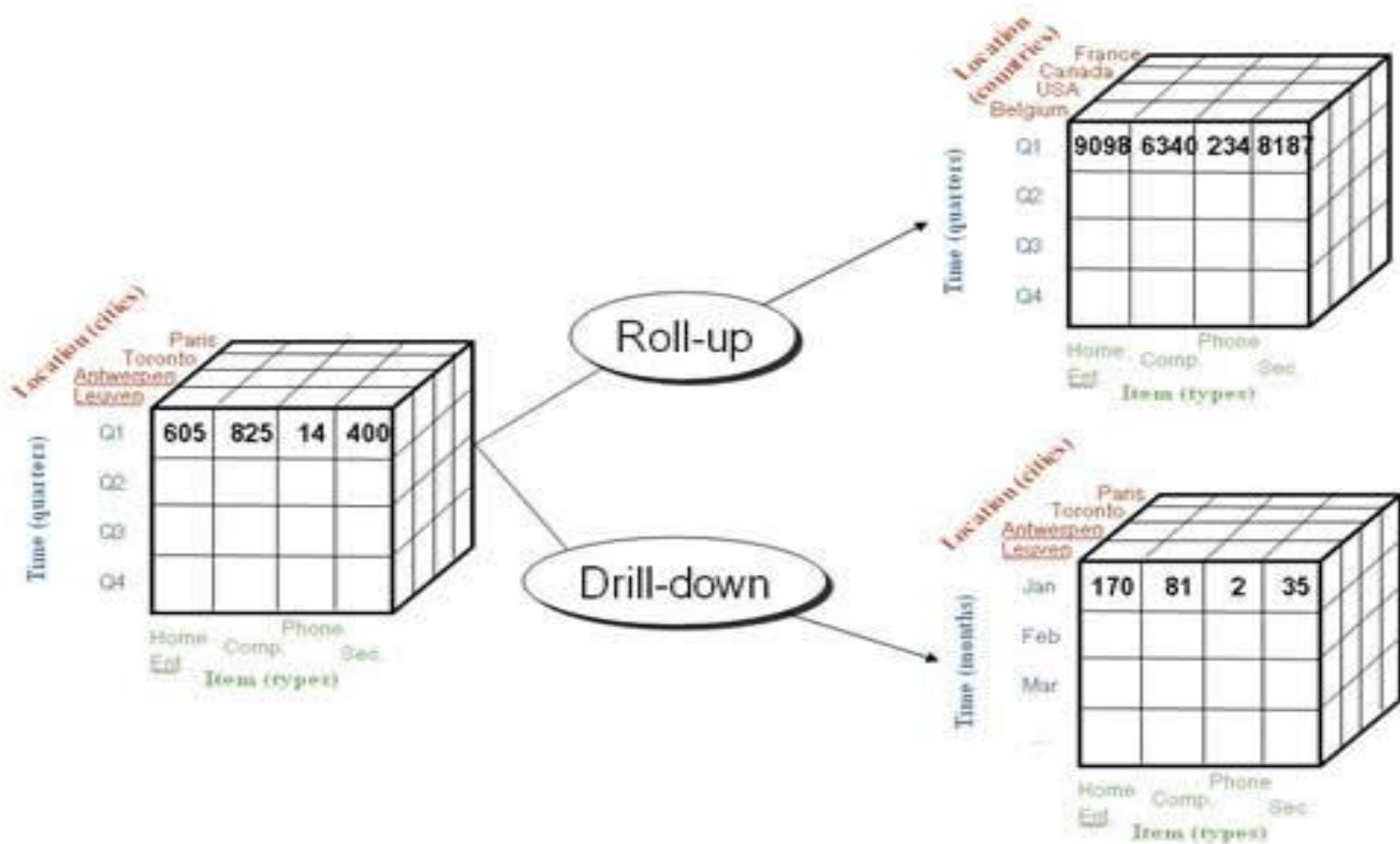
Figure 15-3 Simple OLAP session.

Pravidlá OLAP (Codd)

1. Multi-dimenzionálny konceptuálny model = použitie MDB (dát. kociek)
2. Transparentnosť (používateľ musí mať možnosť využiť všetky prostriedky systému)
3. Dostupnosť (prístup len k potrebným dátam, ale bez závislosti na pôvode zdroja)
4. Konzistentné vykazovanie (s rastúcou veľkosťou by sa nemal znížovať výkon)
5. Architektúra klient-server
6. Generická dimenzionalita (každá dimenzia je štrukturálne a funkčne ekvivalentná)
7. Dynamické ošetrenie riedkych matíc (optimalizácia pre prípad riedkych vstupov)
8. Podpora pre viacero používateľov
9. Neobmedzené krížové dimenzionálne operácie (kalkulácie v rámci dimenzií, aj medzi nimi)
10. Intuitívna manipulácia s dátami (zmena na detailnú úroveň a späť)
11. Flexibilné vykazovanie (možnosť usporiadalať riadky / stĺpce potreby)
12. Neobmedzené dimenzie a úrovne agregácií

OLAP operácie

- **Roll-up** – agregovanie hodnôt v rámci dátovej kocky vystúpením hore v hierarchii konceptov alebo redukciou dimenzií
- **Drill-down** – navigácia k detailnejším dátam zostúpením v hierarchii konceptov alebo pridaním dimenzií (inverzná operácia k roll-up)



Príklad Roll-up – vizualizácia tabuľiek dát

Store: New York

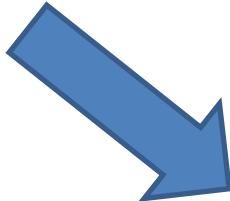
PAGES: STORE dimension

Months	Products					
	<u>COLUMNS:</u> PRODUCT dimension					
	Hats	Coats	Jackets	Dresses	Shirts	Slacks
	Jan	200	550	350	500	520
	Feb	210	480	390	510	530
	Mar	190	480	380	480	500
	Apr	190	430	350	490	510
	May	160	530	320	530	550
	Jun	150	450	310	540	560
	Jul	130	480	270	550	570
	Aug	140	570	250	650	670
	Sep	160	470	240	630	650
	Oct	170	480	260	610	630
	Nov	180	520	280	680	700
	Dec	200	560	320	750	770

Figure 15-6 A Three-dimensional display.

Roll-up

V rámci dimenzie produktov
(PRODUCT dimension) posun z
Products úrovne smerom
vyššie na *Sub-categories*



Store: New York

Sub-categories

PAGES: STORE dimension

COLUMNS: PRODUCT dimension

Months	Outer	Dress	Casual
	Jan	1,100	1,020
	Feb	1,080	1,040
	Mar	1,050	980
	Apr	970	1,000
	May	1,010	1,080
	Jun	910	1,100
	Jul	880	1,120
	Aug	960	1,320
	Sep	870	1,280
	Oct	910	1,240
	Nov	980	1,380
	Dec	1,080	1,520

Figure 15-13 Three-dimensional display with roll-up.

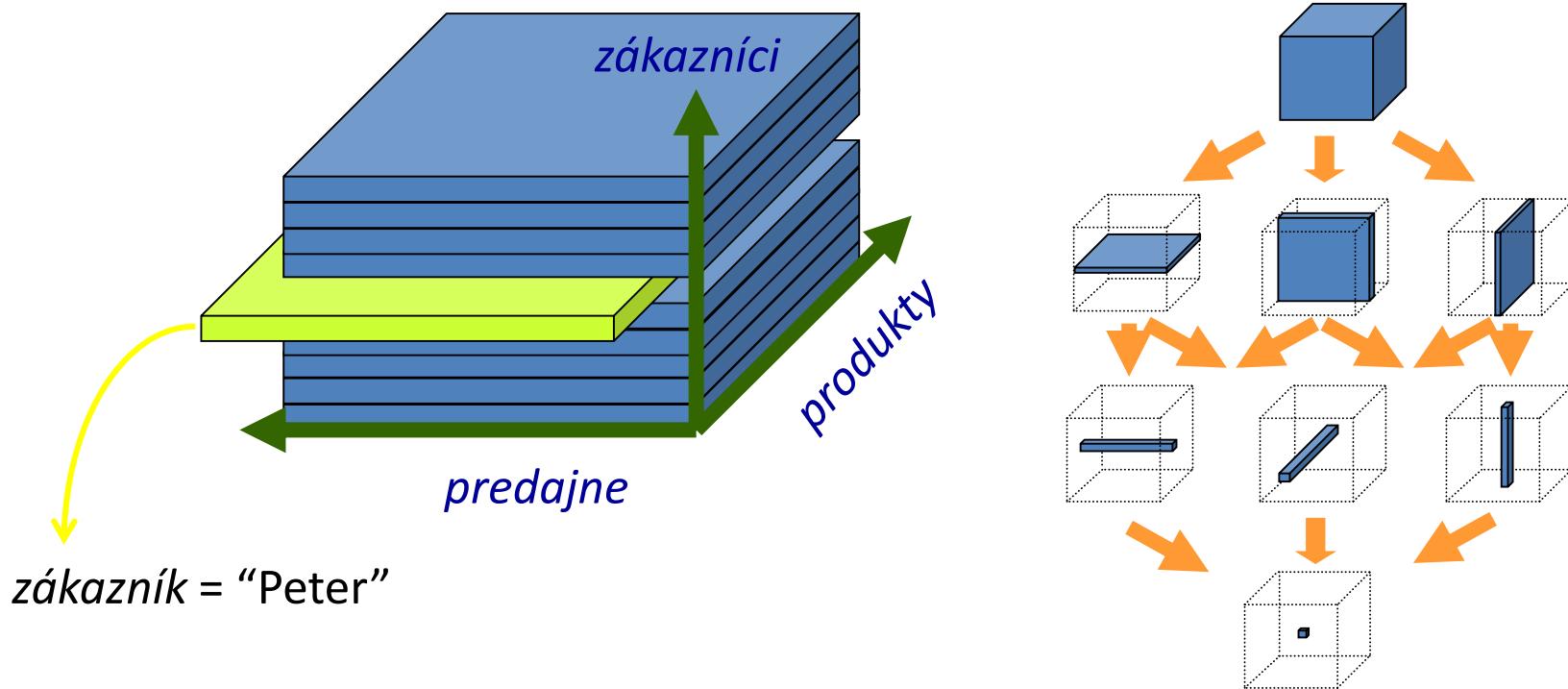
OLAP operácie (2)

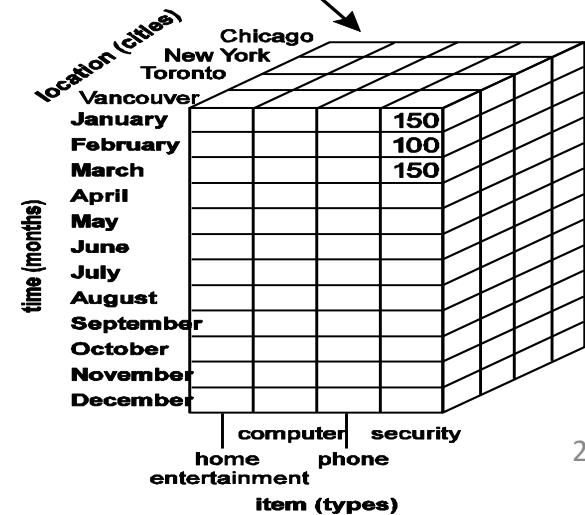
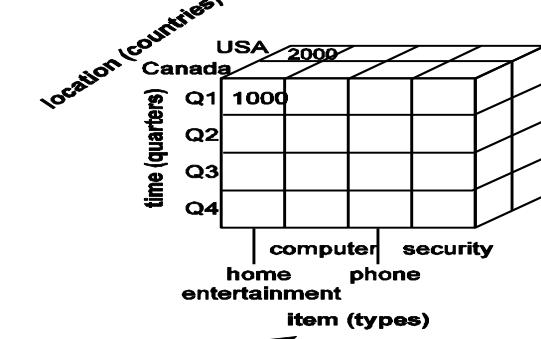
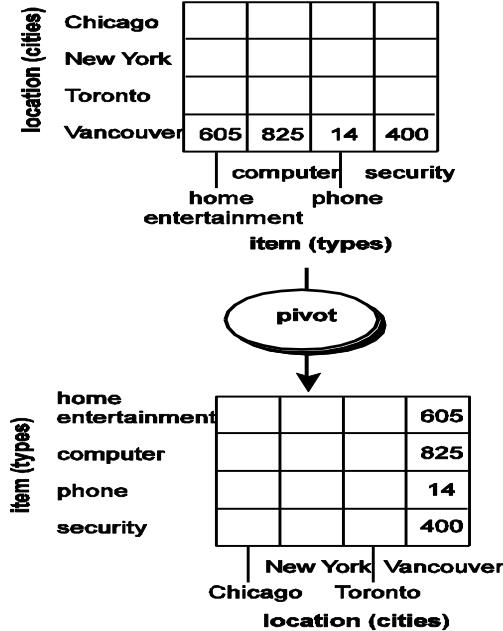
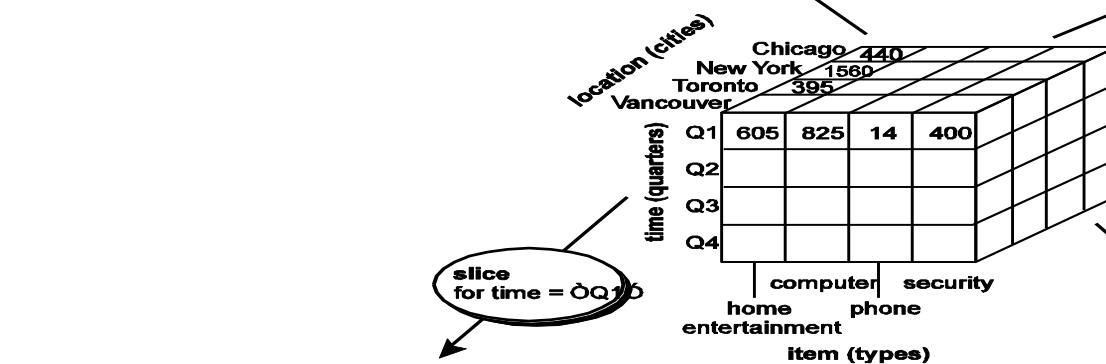
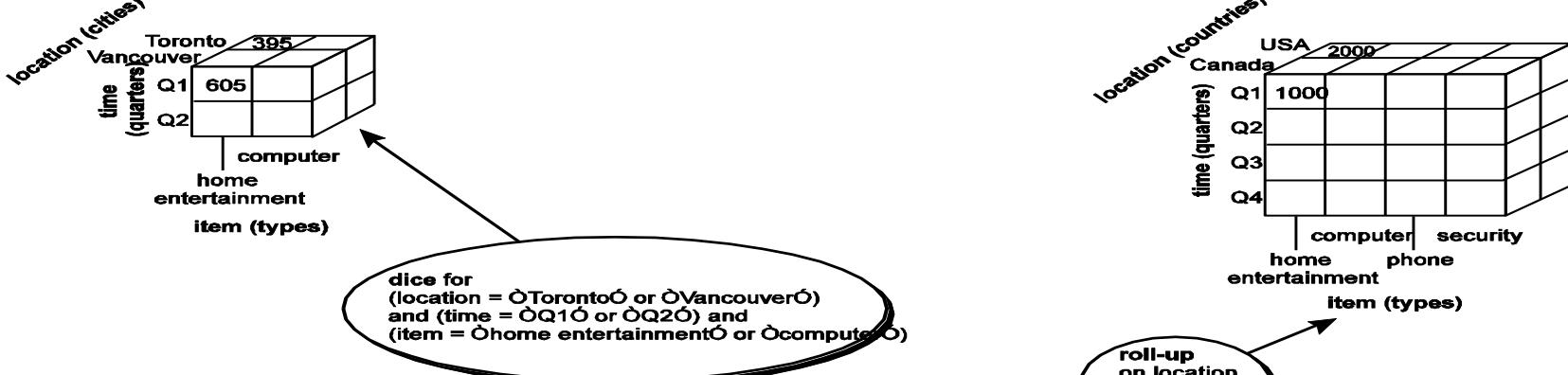
Okrem **Roll-up** a **Drill-down** ešte rozlišujeme:

- **Slice** – vytvorenie pohľadu na kocku pre konkrétné hodnoty jednej dimenzie (Slice – napr. čas=január)
- **Dice** – vytvorenie pohľadu pre výber hodnôt viacerých (Dice – napr. čas = január a február + reg. = Košice) dimenzií
- **Drill-Across** – prechod na inú hierarchiu definovanú nad rovnakou dimenziou (napr. z časovej hierarchie deň-mesiac-kvartál-rok na časovú hierarchiu deň-týždeň-rok)
- **Drill-Through** – prechod na úroveň záznamov v tabuľke – čítanie dát v tabuľke faktov
- **Rotation (Pivot)** – vizualizačná operácia, ktorá rotuje (zamení) osi pre zmenu aktuálneho pohľadu na tabuľku

Príklad výberu „rezov“ pre Slice/Dice

- Výber jednej hodnoty – Slice – 3 možnosti podľa dimenzie
- Výber dvoch hodnôt – Dice – 3 možnosti kombinácií
- Výber 3 hodnôt = pri 3 dimenziách určím konkrétnu bunku dátovej kocky





Úložiská OLAP serverov (architektúry)

- Relačný OLAP (ROLAP)
 - Využíva relačnú databázu (tabuľky) pre ukladanie/manažment dát + OLAP nástroje na podporu chýbajúcich častí
 - Komplexné SQL dopyty, dátové kocky vyrába dynamicky (poskytnuté cez prezentáčnú vrstvu), zložitosť analýzy je nižšia, kvalita výsledkov nižšia, žiadna redundancia
- Multi-dimenziorný OLAP (MOLAP)
 - Vlastná viacrozmerná vektorová pamäťová štruktúra (polia) pre agregácie + má aj prístup k základným dátam DS
 - Dátové kocky sú predpočítané, rýchly prístup k agregovaným dátam, zamerané na výkon, umožňuje zložitejšiu analýzu, nevýhody: vysoké pamäťové nároky, redundancie
- Hybridný OLAP (HOLAP)
 - Kombinácia MOLAP + ROLAP
 - Na detailnej úrovni dát v relačnej DB, agregácie v poliach MDB
- DOLAP (Desktop OLAP)
 - Špeciálny prípad ROLAP, kde dátá sú v rámci servera v relačnej DB + MDB je vytváraná na klientskom počítači (vyžaduje špeciálny software)

Podniková analytika

Deskriptívna analytika a úvod do
prediktívnej analytiky

Obsah

- Deskriptívna analytika
 - Reportovanie (metriky)
 - Vizuálna analytika
 - Dashboard-y
 - Prediktívna analytika
 - KDD - objavovanie znalostí v databázach
 - Dolovanie v dátach (Data Mining – DM)
 - Porovnanie OLAP a DM
 - Procesný pohľad na KDD
-

Reportovanie

- Report = komunikačný artefakt (1 alebo viac dokumentov) pripravený vzhľadom k špecifickému záujmu v prezentovateľnej forme
- Ak sa týka podniku = business report
 - Dôležitá časť BI sú reportovanie a BPM (Business Performance Management)
 - Je základným prvkom v rozhodovaní na manažérskej úrovni
 - Môžu to byť rôzne typy dokumentov a ich kombinácie – naratívne dokumenty s grafickými prvkami, laboratórne reporty, reporty predajov, výročné správy, finančné správy, procedúry, poznámky zo stretnutí, ...
- Základ reportovania sú dokumenty, postupne došlo k vývoju s použitím dashboard-ov a vizuálnej analytiky

Delenie podnikových reportov

- Podľa času
 - Periodický – pravidelné reportovanie (často automatické)
 - Ad-hoc – report vytvorený pre jednu špecifickú akciu v momente keď je to potrebné
- Podľa typu reportovania
 - Reporty založené na metrikách
 - Service-Level Agreements (SLA) – externe, voči zákazníkovi
 - Key Performance Indicator (KPI) – pre interný manažment
 - Reporty v podobe dashboardov
 - Reporty typu „balanced scorecard“

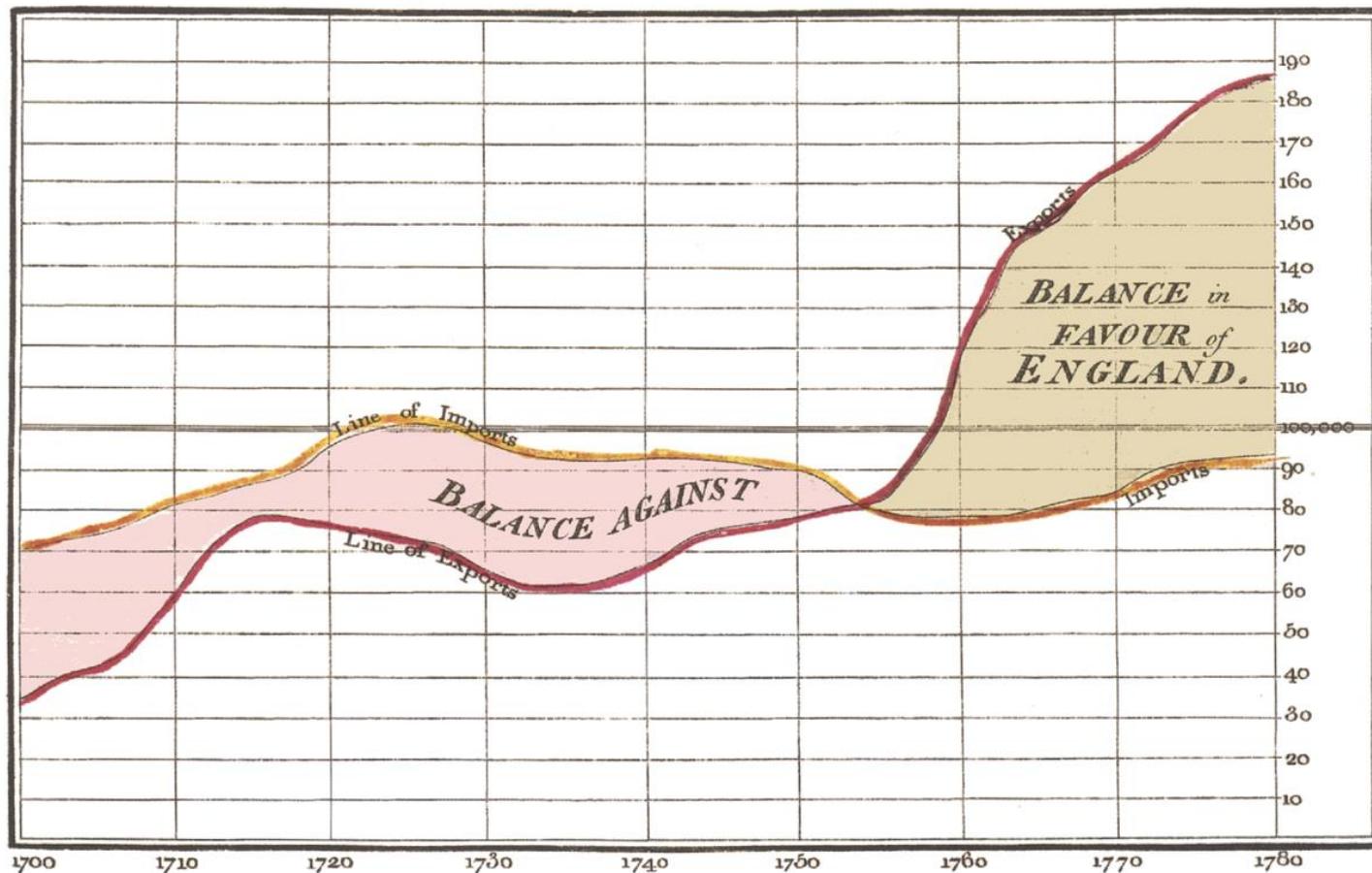
Typické komponenty systému podnikového reportovania

- OLTP – subsystém pre výber meraní rôznych aspektov reportovania a ich zápis do databázy (môže ísť o získavanie dát cez ERP, POS systémy, RFID čítačiek, z Web serverov, ...)
- Dátový prístup – systém pre záznam a prenos dát
- ETL – podobne ako v DS, medzikrok pre extrakciu, overenie a vloženie dát v správnom formáte
- Úložisko dát – súbor, tabuľka (Excel), zvyčajne ale relačná databáza alebo dátový sklad
- Logika systému (Business Logic) – subsystém pre jednotlivé procedúry systému
- Publikovanie – subsystém pre generovanie reportov a ich publikovanie používateľom
- Assurance – časť zodpovedná za kontrolu kvality reportov a ich správneho doručenia

Vizualizácia dát

- = „použitie vizuálnych reprezentácií pre exploráciu, hľadanie významu a komunikáciu dát“
- Vizualizácia dát => vizualizácia informácií
 - prostriedkom je napr. exploratívna analýza dát resp. vizuálna analytika
- Prvé vizualizácie dnešného typu sa začali používať už v 18 storočí
 - William Playfair – vymyslel viacero prvých typov grafov
 - vid' prvý publikovaný príklad z roku 1786 (ďalší slajd) v jeho *Commercial and Political Atlas* – časový graf bilancie obchodných vzťahov Anglicka voči Dánsku + Nórsku
- Dnes existuje celá oblasť venujúca sa návrhu a využitiu postupov pre vizualizáciu dát

Exports and Imports to and from DENMARK & NORWAY from 1700 to 1780.



The Bottom line is divided into Years, the Right hand line into £10,000 each.
Published as the Act directs, 1st May 1786. by W^m Playfair
Neale sculpt^r 352, Strand, London.

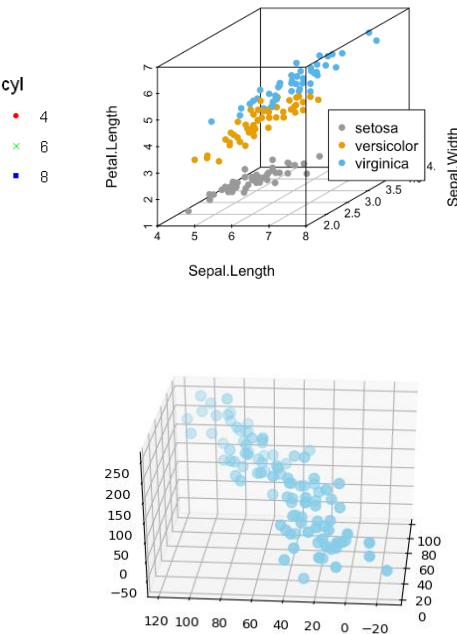
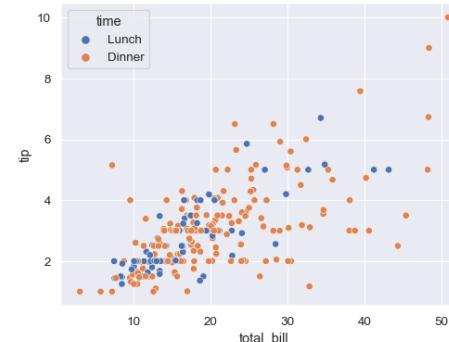
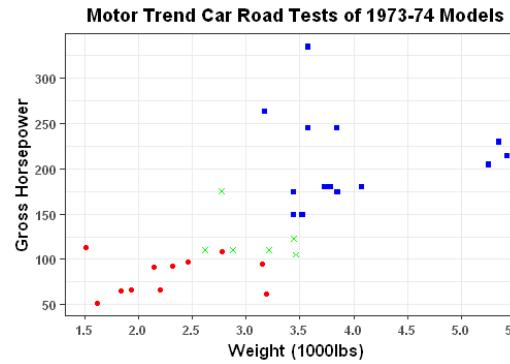
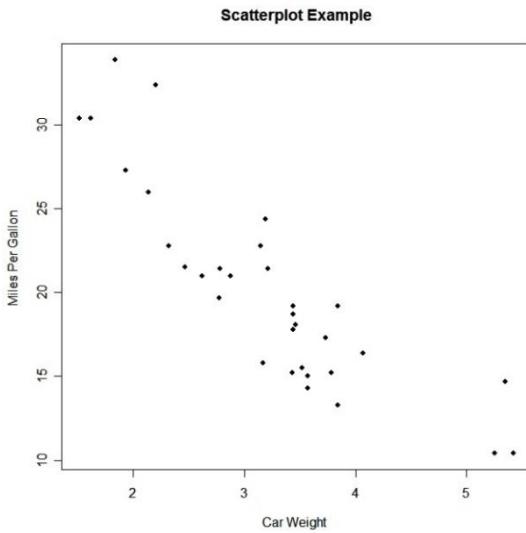
Playfair's trade-balance time-series chart, published in his *Commercial and Political Atlas*, 1786 (Zdroj: wikipedia)

Princípy vizualizácie

- Cieľ: pochopenie dát, hľadanie vzorov, pomoc pri návrhu ďalších krokov, vizualizácia výsledkov analýz
- Prostriedky: rôzne typy statických alebo dynamických grafov a diagramov, s/bez interaktivity
- Základné princípy:
 - Graf musí obsahovať porovnanie vybraných aspektov dát
 - Ak je to možné, mal by sa snažiť vystihnúť kauzalitu, príčinu sledovaného javu
 - Je dobré ukázať viacozmerné dáta pomocou rôznych postupov (využitie farieb a veľkostí, využitie zobrazenia viacerých podmienených grafov podľa ďalších atribútov, ...)
 - Mal by obsahovať dostatočný a zrozumiteľný popis pre jeho pochopenie (popis osí, škály, ...)
 - Obsah musí byť relevantný a kvalitný

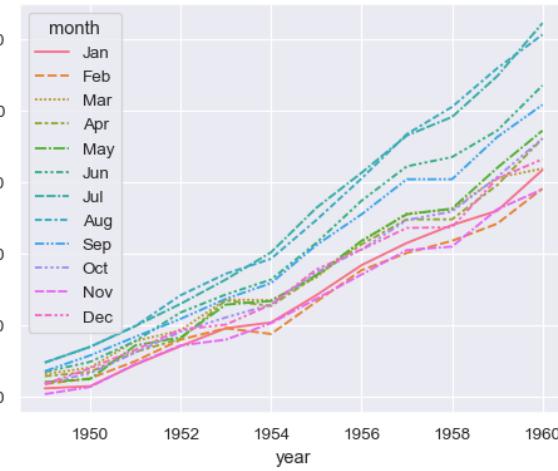
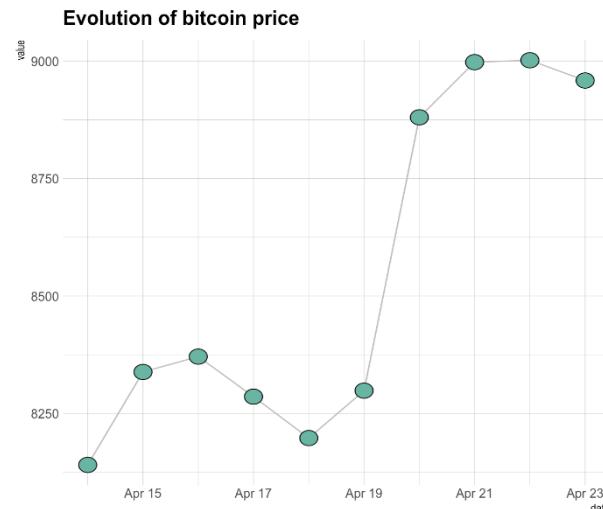
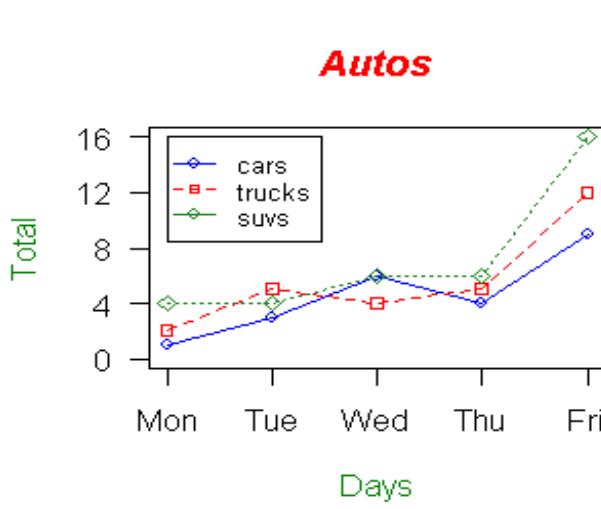
Klasické typy grafov / diagramov

- Bodový graf (Scatter plot)
 - Vzťah dvoch (2D) alebo troch (3D) premenných vyjadrený ako súbor bodov v dátach
 - R - tradičné plot funkcie – Base: plot, Lattice: xyplot, ggplot2: ggplot alebo qplot, ...
 - Python – matplotlib: scatter, seaborn: scatterplot, ...
 - 3D – napr. scatterplot3D balík v R, mplot3D toolkit v python matplotlib, ...



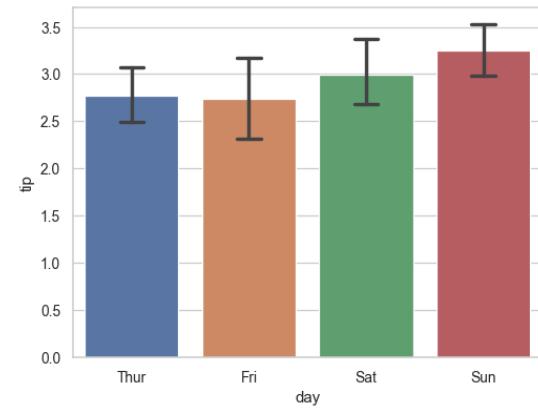
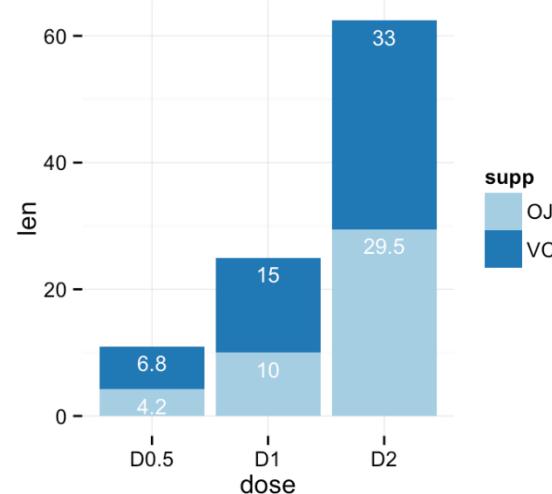
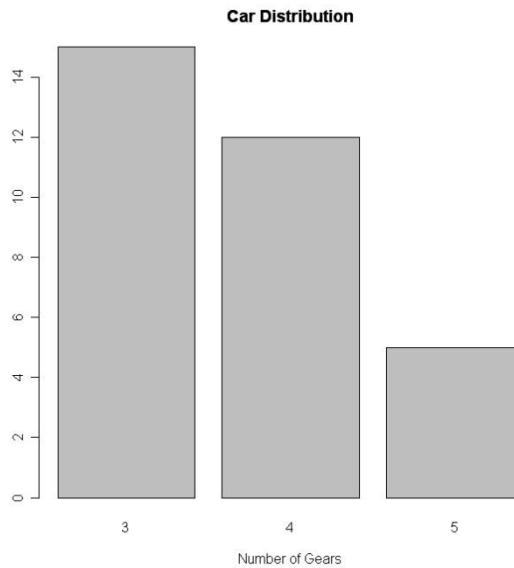
Klasické typy grafov / diagramov (2)

- Čiarový graf (Line chart)
 - Prepojenie dátových hodnôt premennej čiarou
 - Často priebeh premennej v čase (sledovanie zmeny premennej)
 - R – Base: napr. funkcia lines() pre pridanie do plot, ggplot: pridaním geom_line vrstvy, ...
 - Python – štandardný matplotlib plot, lineplot v seaborn, ...



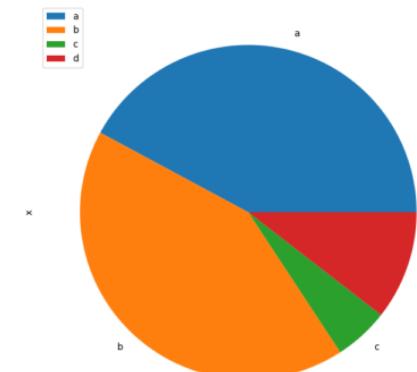
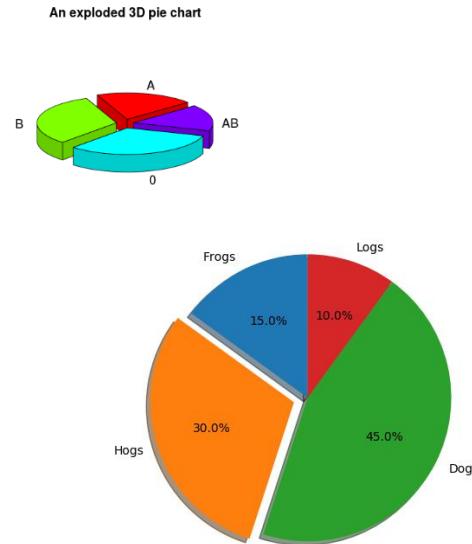
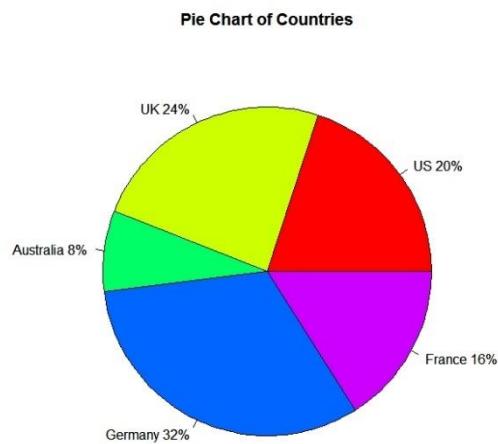
Klasické typy grafov / diagramov (3)

- Stĺpcový graf (bar chart)
 - Ideálny pre nominálne dátu alebo numerické dobre rozdelené do kategórií
 - Porovnanie dát cez jednotlivé kategórie / nominálne hodnoty
 - R – Base funkcia: barplot, lattice: barchart, ggplot: geom_bar
 - Python – seaborn: barplot, matplotlib: bar



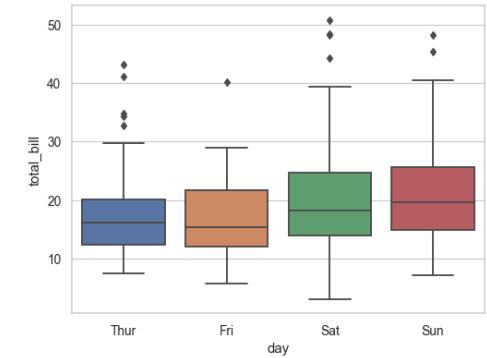
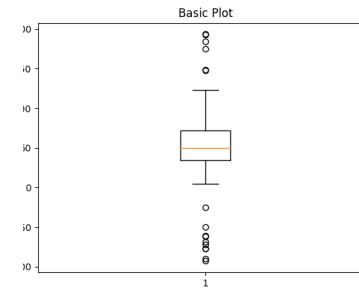
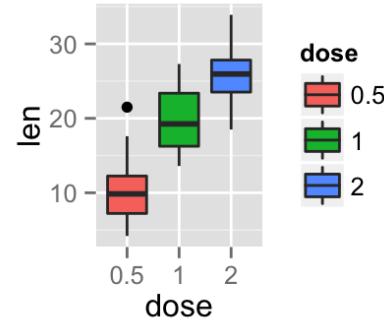
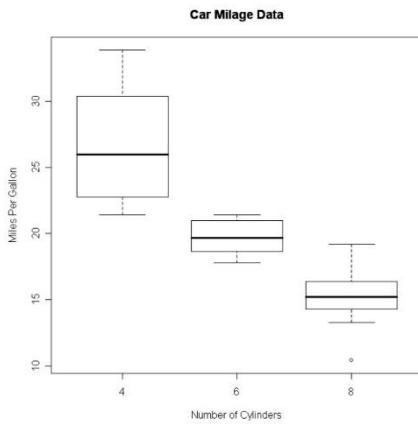
Klasické typy grafov / diagramov (4)

- Koláčový graf (pie chart)
 - Používa sa na viz. zobrazenie relatívnych početností nad danou meranou veličinou (%)
 - R - pie(), pie3D()
 - Python – matplotlib: pie, pandas: plot(kind='pie', ...)



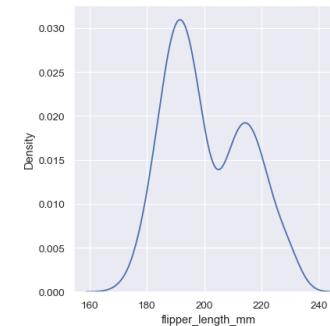
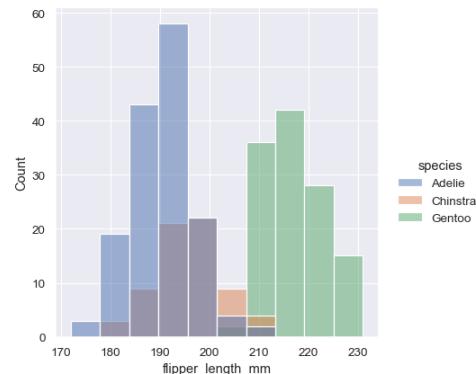
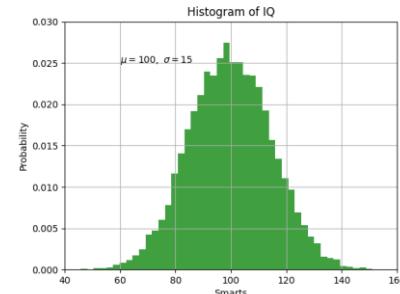
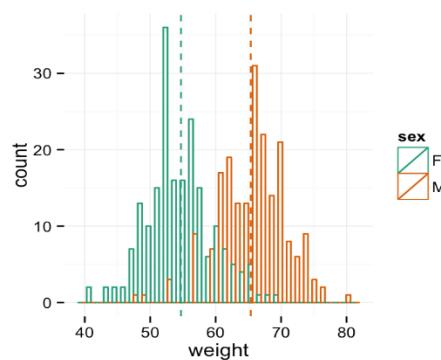
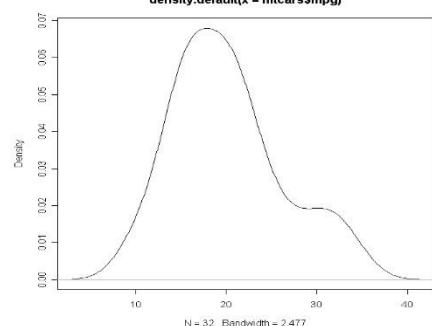
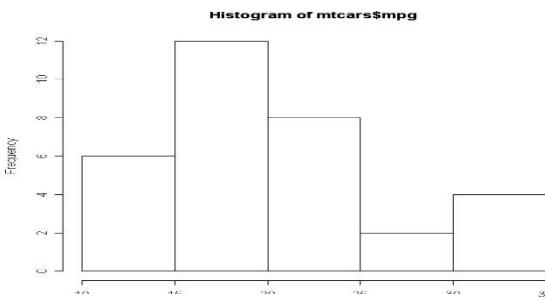
Špecializované typy grafov

- Boxplot
 - Náhľad numerického atribútu (hodnoty `summary()` = sumarizácia 5 čísel, v grafickej podobe)
 - min [1Q Medián 3Q] max
 - Dobré pre porovnanie cez viaceré podmienené grafy nad kategóriami
 - R – base: `boxplot()`, lattice: `bwplot`, ggplot: `geom_boxplot`
 - Python – `matplotlib`: `boxplot`, `seaborn`: `boxplot`



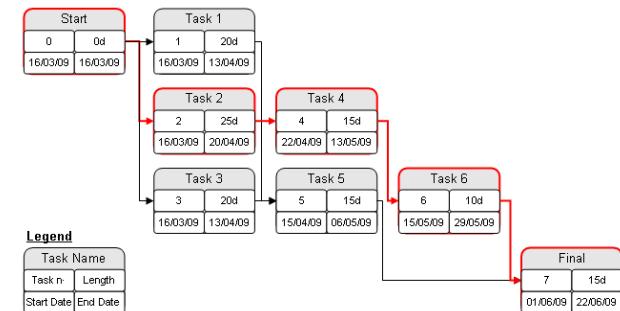
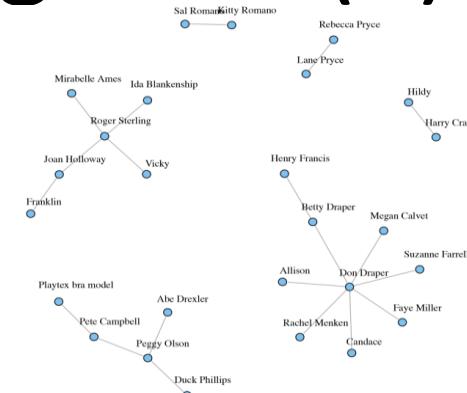
Špecializované typy grafov

- Histogram / graf hustoty spojitej veličiny
 - Histogram – verzia stĺpcového grafu pre zobrazenie distribúcie hodnôt numerického atribútu voči zvoleným intervalom
 - Spojitá verzia = density graf (hustotný graf)
 - R – base: hist() pre histogram, plot(density(x)) pre density graf, lattice: histogram, ggplot: geom_histogram
 - Python – matplotlib: histogram, seaborn: displot (kind=kde pre density graf)



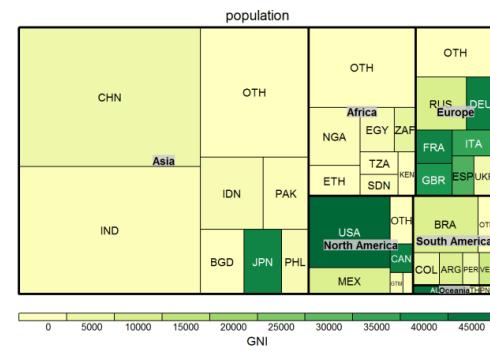
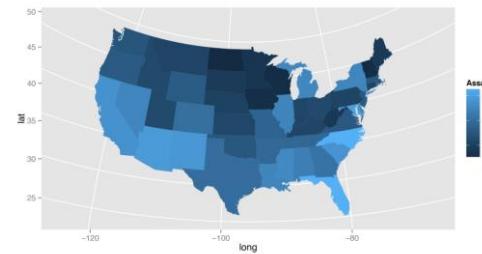
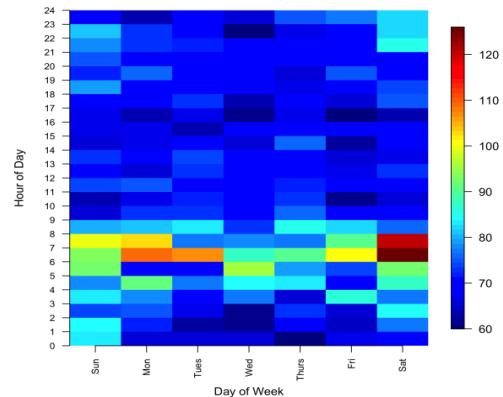
Špecializované typy grafov (2)

- Sietové grafy
 - Vizualizácia grafov pozostávajúcich z uzlov a hrán (vyjadrujúce vzťahy medzi uzlami)
- PERT diagram
 - Špeciálny sietový graf pre plánovanie projektov
- Ganttov diagram
 - Špeciálny typ horizontálneho stĺpcového grafu
 - Zobrazovanie projektových úloh v časových líniach
- Balíky ako igraph, timevis, **plotly**, DiagrammeR, ...



Špecializované typy grafov (3)

- Heat mapy
 - Porovnanie diskrétnych alebo spojitých veličín použitím škály definovej intenzitou a/alebo paletou farieb
- Geografické mapy
 - Zobrazenie informácií na mape vďaka geografickej informácií o dátových bodoch
 - Často kombinované s ďalšou informáciou (napr. s heat mapovaním)
 - Napr. cez maps, ggmap, leaflet (leaflet package v R alebo Folium pre python, plotly, bokeh, ...)
- Tree map
 - Zobrazenie hierarchickej informácie pomocou vnorených obdĺžnikov
 - Napr. treemap balík v R, plotly pre R/python, ...
- A množstvo ďalších !!!
 - Kombinácia uvedených typov (a ďalších postupov) + použitie podmienených grafov, farieb, veľkostí, atď. + využitie dynamického zobrazovania a interaktivity
 - RShiny + RMarkdown + rôzne rozšírenia napr. ako v prípade <http://www.htmlwidgets.org/>
 - Python – napr. Dash, Pyxley, použitie markdown v Jupyter notebookoch, ... + seaborn, ...
 - **Použitie všeobecných knižníc (pre R, Python, ...) ako plotly, bokeh, D3, Highcharts, ...**



Metriky hodnotenia výkonnosti

- Základný prvok pre hodnotenie výkonnosti podniku = Business Performance Management
 - Využitie pre reportovanie
 - Dashboard-y
 - ...
- Systémy merania výkonnosti
 - Key Performance Indicator (KPI)
 - Metodológie
 - Balanced Scorecards (strategický manažment, zameraný na rast)
 - Six Sigma (performance measurement system, zameraný na aktuálnu a krátkodobú ziskovosť)

KPI (Key Performance Indicator)

- Reprezentuje strategickú hodnotu a meria výkonnosť voči očakávanému cieľu
- Je to numerická hodnota, zvyčajne však nie hrubá, ale sú to rôzne upravené hodnoty na základe vstupov, ako:
 - Priemerná hodnota
 - Percentuálna hodnota
 - Rate (konverzia) – dosiahnutá hodnota pre špecifický cieľ, napr. miera nezamestnanosti, počet prejdení na koniec procesu (ako napríklad objednávka v e-shope)
 - Pomery – porovnania (ratio)
 - „per X“ hodnota – napr. Náklady na 100 kusov (náklady per 100 ks)
 - Zložitejšie odvodenia – odvodené atribúty
- Základné rozdelenie KPIs
 - Výstupné (lagging / outcome) KPIs – meranie aktivity v minulosti, napr. príjmy
 - Operačné (operational / driver / leading indicators) KPIs – meranie aktivít s významným vplyvom na výstupné KPI, napr. aktuálne najlepšie predaje

KPIs v praxi

- Operačné oblasti pre meranie KPI
 - Meranie výkonnosti voči zákazníkom
 - Metriky pre meranie spokojnosti zákazníkov, rýchlosť a úspešnosti riešenia problémov, udržania zákazníkov, ...-
 - Výkonnosť služieb
 - Metriky pre call centrá, úspešnosť riešenia požiadaviek, SLAs (Service Level Agreements), efektívnosť doručenia, úspešnosť obnovenia služieb, ...
 - Predaje (operácie)
 - Počet nových účtov, počet dohodnutých stretnutí, priemerná doba hovoru
 - Plán predajov / odhady
 - Metriky ako pomery cena/nákup, celkovo uzavreté kontrakty, pomer medzi odhadom a plánom, ...

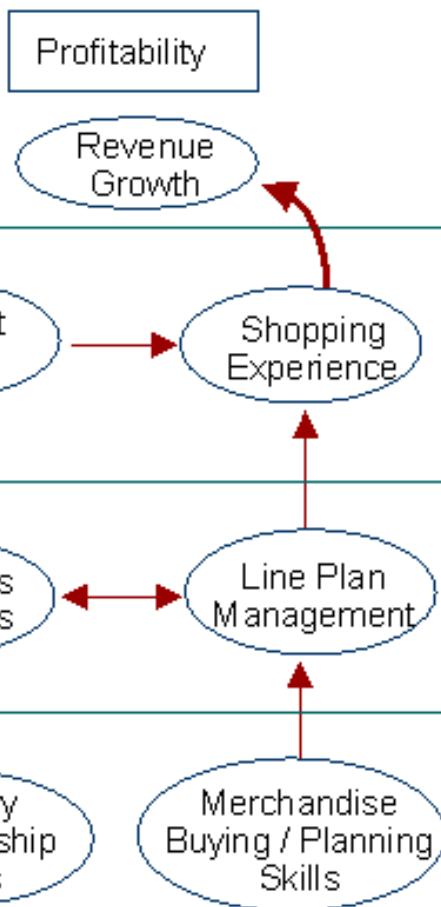
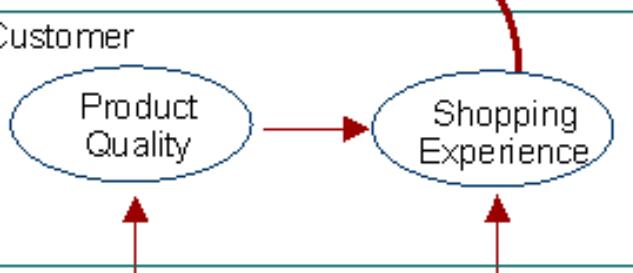
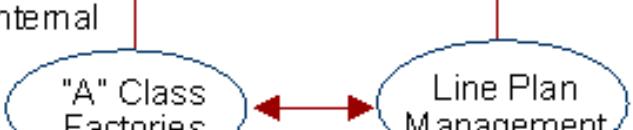
Príklady KPI

- Príklad na konverziu (rate)
 - Step Completion Rate – koľko používateľov prejde všetky kroky na koniec (stránka Ďakujeme za nákup)
- Priemerná hodnota
 - AVO (Average Order Value) – priemerná hodnota nákupov
- Per X
 - Náklady na zákazníka (per customer)
 - Príjmy na zákazníka
 - (+avg) Priemerný počet návštěv na jedného používateľa
- Odvodené hodnoty
 - Lojalita zákazníka / novosť – čas od poslednej návštavy (registrovaného / anonymného)
 - Frekvencia návštěv zákazníka (retention)
- Percentuálne hodnoty
 - % príjmov generovaných z marketingovej kampane

Balanced scorecard

- Balanced Scorecard (BSC) - súbor vybraných KPIs, ktoré vyvážene charakterizujú firmu alebo organizáciu, nástroj strategického manažmentu, súčasť analýzy stratégií a návrhu akcií pre strategický rast a rozvoj
- 4 základné perspektívy
 - Finančná perspektíva (financial) - finančné ukazovatele ako obrat, zisk, zadlženosť, ... (cieľom je byť atraktívny pre partnerov a investorov)
 - Zákaznícka perspektíva (customer) - podiel na trhu, spokojnosť zákazníkov, podpora lojality, obraz firmy, ... (hodnota, ktorú je zákazník ochotný zaplatiť)
 - Procesná perspektíva (internal processes) – zníženie nákladov, skrátenie doby výroby, ... (excelentné interné procesy)
 - Perspektíva potenciálu (learning and growth) – schopnosť firmy rásť, t.j. zvyšovanie kvalifikácie zamestnancov, spokojnosť zamestnancov, ... (kvalitná infraštruktúra a excelentný ľudský kapitál)

Príklad – strategická mapa a BSC

Strategická mapa	BSC			
Sourcing & Distribution Pathway	Measurement	Target	Initiative	Budget
Financial 	<ul style="list-style-type: none"> Operating Income Sales vs. Last Yr 	<ul style="list-style-type: none"> 20% Increase 12% Increase 	<ul style="list-style-type: none"> Likes Program 	\$xxx
Customer 	<ul style="list-style-type: none"> Return Rate <ul style="list-style-type: none"> Quality Other Customer Loyalty <ul style="list-style-type: none"> Ever Active % # units 	<ul style="list-style-type: none"> Reduce by 50% each yr 60% 2.4 units 	<ul style="list-style-type: none"> Quality management Customer loyalty 	\$xxx \$xxx
Internal 	<ul style="list-style-type: none"> % of Merchandise from "A" factories Items in-Stock vs. Plan 	<ul style="list-style-type: none"> 70% by year 3 85% 	<ul style="list-style-type: none"> Corporate Factory Development Program 	\$xxx
Learning 	<ul style="list-style-type: none"> % of Strategic Skills Available 	<ul style="list-style-type: none"> yr 1 50% yr 3 75% yr 5 90% 	<ul style="list-style-type: none"> Strategic Skills plan Merchants Desktop 	\$xxx \$xxx

Six Sigma

- Od gréckeho písmena sigma, ktorý sa používa pre vyjadrenie odchýlky v štatistike
- Six Sigma je metodológia výkonnostného manažmentu zameraná na redukciu počtu chýb v podnikových procesoch na minimálne (blízko nuly) DPMO (Defects Per Million Oportunities)
- Používa tzv. DMAIC výkonnostný model zlepšovania
 - Je to uzavretý cyklus zlepšovania podnikových procesov s krokmi
 - Define – definujú sa ciele a obmedzenia zlepšenia aktivity vybraného procesu
 - Measure – meria sa aktuálny systém, monitoruje sa stav a vyberajú sa vhodné metriky
 - Analyze – analyzuje sa systém a identifikuje sa „gap“ medzi súčasnou a očakávanou výkonnosťou
 - Improve – Iniciujú sa akcie pre redukciu gap-u hľadaním lepších, rýchlejších, lacnejších postupov (s použitím metód projektového manažmentu)
 - Control – vylepšený systém sa nasadzuje (inštitucionalizuje) zavedením modifikácií systémov, zmenou praktík, postupov, procedúr, ...
- Oproti BSC – je zameraný na ziskovosť (nie rast), vytvára pohľad na aktuálny stav a identifikuje metriky ktoré vedú k lepšej ziskovosti, zahŕňa aj operačný manažment (nielen strategický) => BSC sa zameriava na zlepšenie celkovej stratégie a Six Sigma na zlepšenie procesov

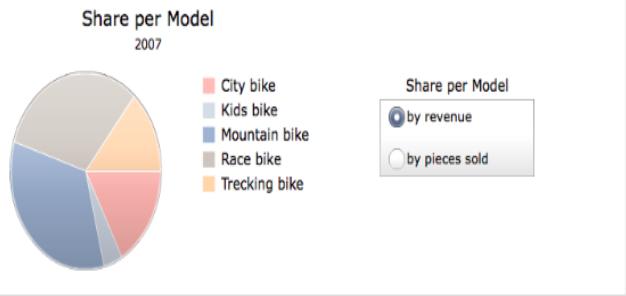
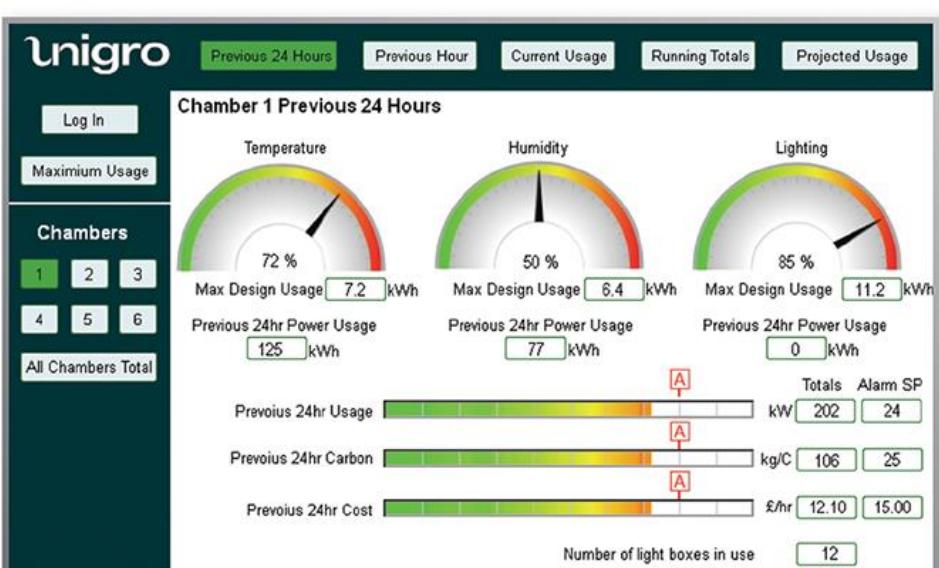
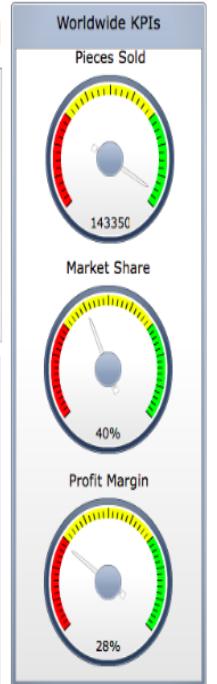
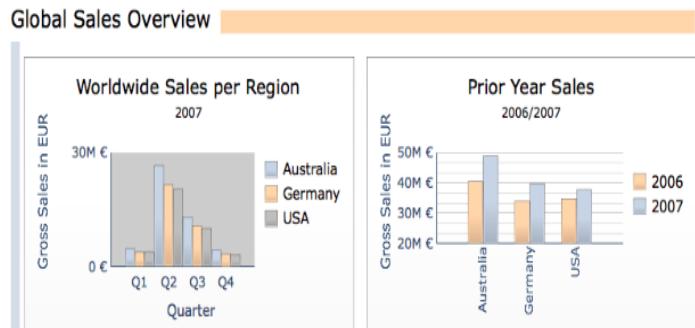
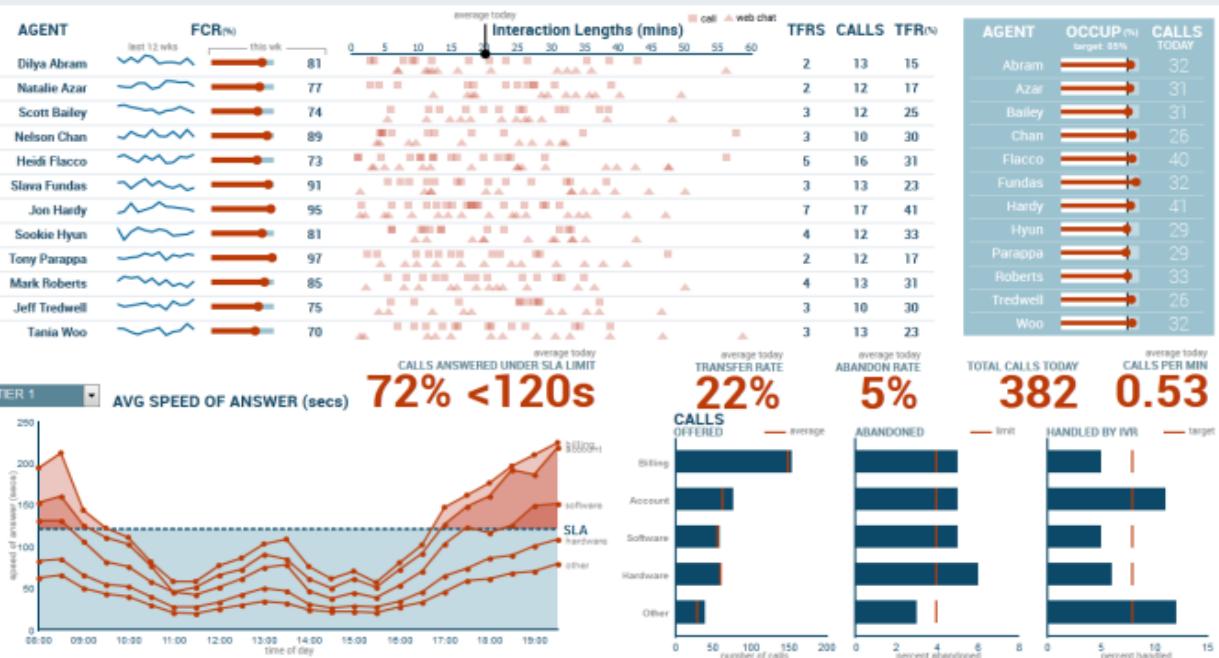
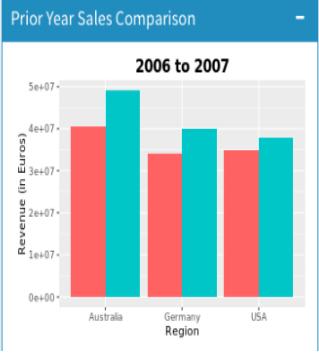
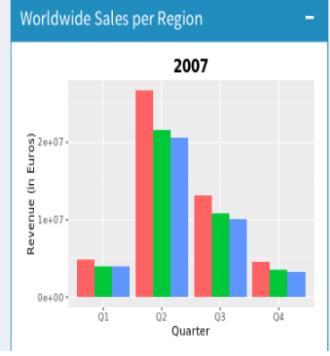
Dashboard vs. Report

- Dashboard
 - Poskytuje vizuálne zobrazenie dôležitých informácií ktoré vznikli zlúčením a sú usporiadane na jednej obrazovke => informácie môžu byť globálne podchýtené, dosiahnuteľné a ďalej skúmané jedným spôsobom
- Rozdiel medzi reportom a dashboard-om (DSB)
 - Vždy je menej DSBs ako reportov – zväčša je DSB vytvorený z detailov dosiahnuteľných v reportoch, často práve veľký počet reportov vedie k potrebe vytvorenia jednotiaceho DSB
 - DSB zlučuje kľúčové dáta z viacerých reportov do jednej reprezentácie
 - DSB zjednodušuje pohľad na dátu v reportoch
 - DSB obsahuje viac vizuálnych prvkov ako tabuliek
 - DSB poskytuje často možnosť linkovať pôvodné (raw) dátu
 - Zamestnanci vytvárajú reporty, DSBs identifikujú manažéri a vedúci zamestnanci

CALL CENTER DASHBOARD

Dundas Data Visualization Inc.

inquiry
Billing
date picker
today

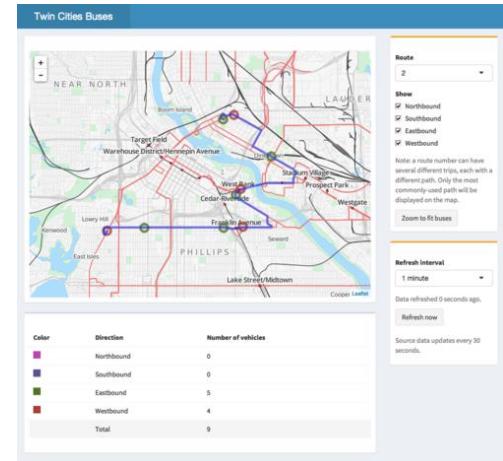


Návrh dashboardov

- Eckerson (2006) – najdôležitejšia vlastnosť dashboardu je zobrazenie 3 úrovní na jednej stránke
 - Monitorovanie – grafické abstrahované dátá pre sledovanie KPIs
 - Analysis – sumarizované dimenzionálne dátá pre analýzu podstaty problému
 - Manažment – detailné operačné dátá pre identifikáciu aká akcia by sa mala urobiť pre nápravu problému
- Dashboard
 - Používa vizuálne komponenty pre zvýraznenie dát a výnimiek vyžadujúcich akcie – je dôležité vybrať informácie ktoré zvýrazniť
 - Ak existujú, je dobré použiť štandardné KPIs v rámci daného odvetvia
 - Mal by vyžadovať minimálny tréning / byť jednoduchý na používanie
 - Kombinuje dátá z rôznych zdrojov / systémov do jedného sumarizovaného pohľadu na podnik / proces
 - Umožňuje zostúpiť na úroveň reportov resp. konkrétnych dát v nich (drill-down, drill-through) pre lepšie pochopenie a evaluáciu kontextu
 - Prezentuje dynamický pohľad s obnovou dostatočnou pre potreby sledovaných procesov
 - Vyžaduje minimum kódovania pre implementáciu, nasadenie a údržbu (skôr ide o vyskladanie prvkov – reportovacích elementov – na dashboard, ktoré sú zodpovedné za správne fungovanie)

Dashboard-y v R/Python

- R
 - Vlastné Rshiny rozhranie
 - Ďalšie balíky pre Rshiny
 - Napr. balík shinydashboard
 - <https://rstudio.github.io/shinydashboard/>
 - Priamo v Rmarkdown
 - Tvorba priamo v rozhraní Rmarkdown, cez flexdashboard (špeciálny typ dokumentu) <http://rmarkdown.rstudio.com/flexdashboard/>
- Python
 - Plotly & Dash
 - <https://plotly.com/dash/>
 - Bokeh
 - <https://bokeh.org/>
 - Pyxley – podobné Rshiny



Prediktívna analytika a KDD

- Prediktívna analytika => hľadanie vzorov v dátach pre získanie odhadu o budúcom vývoji na základe aktuálnych a historických dát => táto potreba tradične vedie na metódy „dolovania v dátach“
- Objavovanie znalostí v databázach = KDD = Knowledge Discovery in Databases
 - Iteratívny a interaktívny proces semiautomatickej extrakcie znalostí z databáz
 - Nájdené znalosti musia byť platné (z pohľadu štatistiky), doposiaľ neznáme a potenciálne užitočné, pričom by mali byť pre človeka zároveň zrozumiteľné a pochopiteľné
- Často sa celý proces zamieňa za Dolovanie v dátach = Data Mining (DM) => je to len jeden z krokov procesu KDD, konkrétnie aplikácia algoritmov na hľadanie vzorov v dátach
- Pre KDD sa používajú metódy z rôznych oblastí ako:
 - Štatistika
 - Strojové učenie (Machine Learning), Teória informácií
 - Umelá inteligencia
 - Databázové systémy, Dátové sklady, ...
 - Vizualizačné techniky
 - ...

OLAP vs. DM

	OLAP	Data Mining (KDD)
Motivácia použitia	Viditeľnosť historických dát v podniku (otázka: Čo sa deje ?)	Predikcia budúcich hodnôt, hľadanie skrytých vzorov (otázka: Aký bude vývoj ?)
Granularita dát	Sumarizované dáta	Dáta na úrovni záznamov
Počet dimenzií	Obmedzený počet dimenzií	Veľký počet dimenzií (neobmedzený)
Počet vstupných atribútov	Nízky počet atribútov	Veľa atribútov
Veľkosť dát v rámci 1 dimenzie	Nie príliš veľké	Zvyčajne rozsiahle pre každú dimenziu
Prístup k analýze	Riadený používateľom, interaktívna analýza = „manuálne dolovanie“	Riadené dátami = „automatické dolovanie“
Techniky analýzy	Operácie s dátovou kockou	Predspracovanie dát a aplikácia algoritmov pre získanie znalostí
Stav a vývoj technológií	Známe a rozsiahlo využívané postupy	Stále sa vyvíjajúce postupy (veľká časť v reálnej praxi ešte ani nie je)

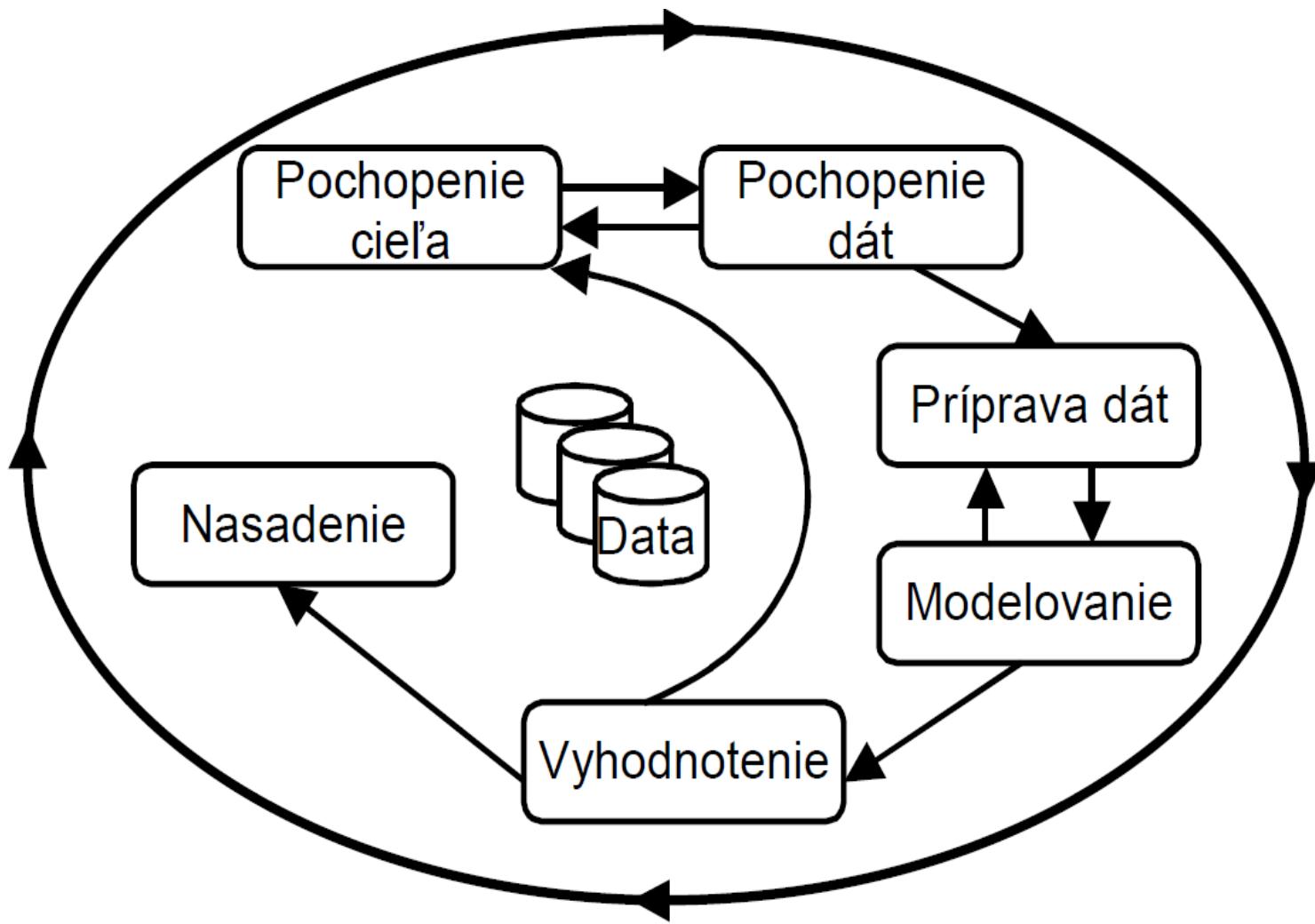
Základný pohľad na proces KDD

- Takýto proces je iteratívny a interaktívny, kroky sa často revidujú, opakujú, kým sa nedopracujeme k výsledku podľa našich potrieb
- Proces je možné zhrnúť do krokov:
 1. Pochopenie riešeného problému (aplikácie) – existujúce zdroje, znalosti, analýza cieľov, ...
 2. Integrácia a čistenie dát – integrácia heterogénnych zdrojov, odstránenie šumu, nekonzistentných hodnôt, ošetrenie chýbajúcich hodnôt
 3. Selekcia dát – na základe analýzy atribútov sa uskutoční výber podmnožiny dát pre cieľ KDD
 4. Transformácia dát – úprava dát do vhodnej reprezentácie (z pohľadu cieľa)
 5. Dolovanie v dátach = Data Mining – aplikácia metód pre získanie vzorov v dátach (popisné metódy, klasifikácia / predikcia, zhlukovanie, detekcia anomálií, ...)
 6. Vyhodnotenie výsledkov DM – meranie úspešnosti nájdených modelov / popisov (vzorov) => výber potenciálne zaujímavých vzorov
 7. Prezentácia / vizualizácia vzorov (znalostí)
 8. Použitie objavených znalostí v aplikácii

Štandardizácia procesu KDD – CRISP-DM

- Pre zjednotenie rámca KDD procesu začali postupne vznikať rôzne metodológie s cieľom
 - Štandardizácie postupov
 - zdieľania skúseností z úspešných prípadov (Best Practices)
- Vznikli napr. CRISP-DM, SEMMA, 5A
- CRISP-DM (Cross Industry Standard Process for Data Mining)
 - Všeobecná metodológia (bez naviazania na doménu, nástroje, ...)
 - Vychádza z praktických skúseností
 - Poskytuje procesný model zložený zo 6 fáz (nadväzuje na body KDD a rozvíja ich => vid' predch. slajd)

Procesný model CRISP-DM



Popis fáz CRISP-DM

- **Pochopenie cieľa** (problému)
 - 1. Pochopenie požiadaviek a cieľov úlohy (obchodné)
 - 2. Posúdenie aktuálnej situácie a definovanie kritérií úspešnosti (obchodné)
 - 3. Stanovenie technickej verzie cieľov a kritérií z pohľadu DM
 - 4. Návrh plánu projektu
- **Pochopenie dát**
 - 1. Prvotný zber dát
 - 2. Popis dát (význam atribútov, prehľadové charakteristiky dát)
 - 3. Prieskum dát (vizualizácia, vztahy medzi dvojicami atribútov, jednoduché štatistické analýzy a vizualizácie)
 - 4. Verifikácia kvality dát (analýza chýbajúcich hodnôt, anomálií)
- **Príprava dát** – rôzne postupy pre predspracovanie dát pre DM
 - Integrácia dát – kombinácia rôznych zdrojov dát
 - Selekcia dát – podľa potrieb cieľa, algoritmov DM, podľa dostupnosti, ...
 - Čistenie dát – normalizácia, vyhľadzovanie (ošetrenie anomálií), ošetrenie chýbajúcich hodnôt, redukcia dát
 - Konštrukcia dát – vytvorenie odvodených atribútov
 - Formátovanie dát – úprava vstupov pre DM nástroj / algoritmus

Popis fáz CRISP-DM (2)

- **Modelovanie** (iteratívny proces aplikácie metód DM)
 - 1. Výber techniky DM / modelovania (napr. rozhodovacie stromy, regresný model, asociačné pravidlá, ...)
 - 2. Návrh testovania (napr. presnosť klasifikačného modelu, rozdelenie dát na trénovaciu a testovaciu množinu, ...)
 - 3. Tvorba modelu (aj s optimalizáciou parametrov modelu)
 - 4. Vyhodnotenie modelov (z pohľadu DM + vzhľadom k aplikácii)
- **Vyhodnotenie**
 - 1. Verifikácia naplnenia (obchodných) cieľov
 - 2. Kontrola správnosti aplikácie celého procesu
 - 3. Stanovenie ďalších krokov (nasadenie, ukončenie, reštart, ...)
- **Nasadenie** – plán nasadenia výsledkov DM do praxe, opakovateľnosť implementácie procesu DM
 - 1. Plán pre nasadenie
 - 2. Plán pre monitorovanie a udržiavanie aplikácie
 - 3. Záverečná správa – zhrnutie / prezentácia projektu a dosiahnutých výsledkov
 - 4. Posúdenie projektu – vyhodnotenie celého projektu

Podniková analytika

Procesy KDD, pochopenie cieľa a dát,
predspracovanie dát

Obsah

- Základné typy úloh DM
- Príklady úloh KDD
- KDD - objavovanie znalostí v databázach => CRISP-DM
 - Pochopenie cieľa
 - Pochopenie dát
 - Dáta a ich predspracovanie
 - Dôležitosť predspracovania
 - Rôzne kroky predspracovania

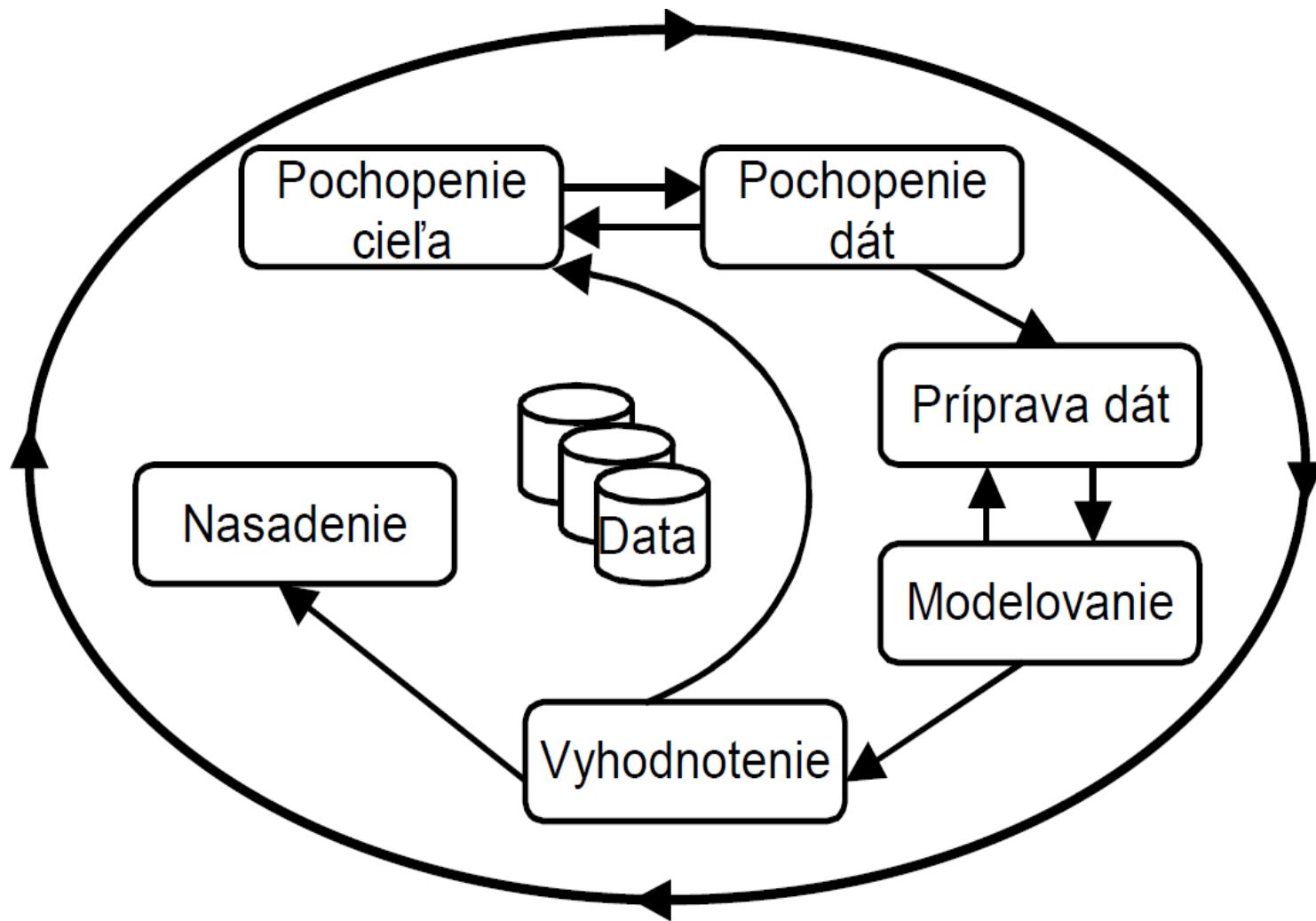
Typy úloh DM

- DM = kľúčový krok v procese KDD
- Základné rozdelenie úloh
 - Prediktívne dobovanie v dátach (kontrolované učenie = „učenie s učiteľom“, supervised learning – prediktívny DM)
 - Klasifikácia (kategorizácia) – predikuje kategoriálne atribúty (tryedy)
 - Predikcia = regresia – predikuje spojité (numerické) výstupné atribúty
 - Deskriptívne dobovanie v dátach (nekontrolované učenie, „učenie bez učiteľa“, unsupervised learning – deskriptívny DM) – popis (charakterizácia) konceptov, relácií, vzťahov, ...
 - Zhlukovanie – zgrupuje dátové položky podľa ich (ne)podobnosti medzi sebou (na základe hodnôt atribútov)
 - Asociačné pravidlá – popisuje vzťahy medzi atribútmi v podobe pravidiel (IF množina_atts_1 THEN množina_atts_2)
 - Ďalšie popisné metódy – vizualizačné metódy, extrakcia príznakov (metódy redukcie dimenzionality), vysvetľujúce modely

Príklady úloh

- Klasifikácia (určenie kategoriálnej hodnoty = triedy / kategórie)
 - Vytvorenie anti-spam filtra
 - Priamy marketing (rozhodnutie kúpiť alebo nekúpiť nejaký produkt)
 - Určenie podvodných transakcií
- Predikcia (Regresia ... regresné modely pre určenie numerickej hodnoty)
 - Investície – predikcia finančných / burzových indexov (predikcia časových radov)
 - Predikcie predajov produktu na základe reklamy
 - Predikcia funkcií rôznych meraných veličín (rýchlosť vetra na základe teploty, tlaku, ...)
- Zhlukovanie
 - Segmentácia zákazníkov (trhu) do skupín podľa podobnosti ich transakcií
 - Zhlukovanie dokumentov, zhlukovanie burzových dát
- Asociačná analýza (asociačné pravidlá)
 - Analýza nákupného košíka
 - Manažment inventáru podnikových pobočiek

Procesný model CRISP-DM



Pochopenie úlohy

- Stanovenie cieľov
 - Aký typ znalostí chcem nájsť ?
 - Aké dátá pre náš proces máme ?
 - Je daný problém riešiteľný ?
 - Budú získané výsledky užitočné v praxi ?
 - Aký tvar a formu by malo mať zobrazenie výsledkov procesu ?
 - Sú dátá vhodné pre naše metódy DM ?
- Pochopenie cieľov úlohy
 - Dôležité aspekty (obchodné)
 - Náklady
 - Prínos
 - Stanovenie predbežného plánu
 - Technické aspekty (formát a bezpečnosť)
 - Forma odovzdania dát
 - Anonymizácia dát
 - Formát dát
 - Dôležité rešpektovať špecifiká domén (priemysel, štátna správa, zdravotníctvo, veda, obchod)

Rôzne pohľady na typy dát pre DM

- Typy dát pre DM podľa zamerania
 - Demografické dáta (charakteristika osôb)
 - Behaviorálne dáta (nákupy, predaje, chovanie osôb, ...)
 - Psychografické dáta (prieskumy dotazníkmi – pomáhajú pri analýze chovania zákazníkov)
- Typy databáz z hľadiska obsahu
 - Zákaznícke databázy – údaje o zákazníkoch + niektorých ich aktivitách
 - Transakčné databázy – údaje o aktivitách zákazníkov (často anonymizované)
 - Databáza marketingovej história – oslovenia, kampane
 - Podnikový dátový sklad
 - ...
- Typy dát podľa formátu
 - E/R a transakčné databázy
 - Objektové databázy
 - Multimedialni databáze
 - Webové zdroje
 - Textové dokumenty
 - Priestorové, časové dáta
 - ...

Predspracovanie dát a atribúty pre DM

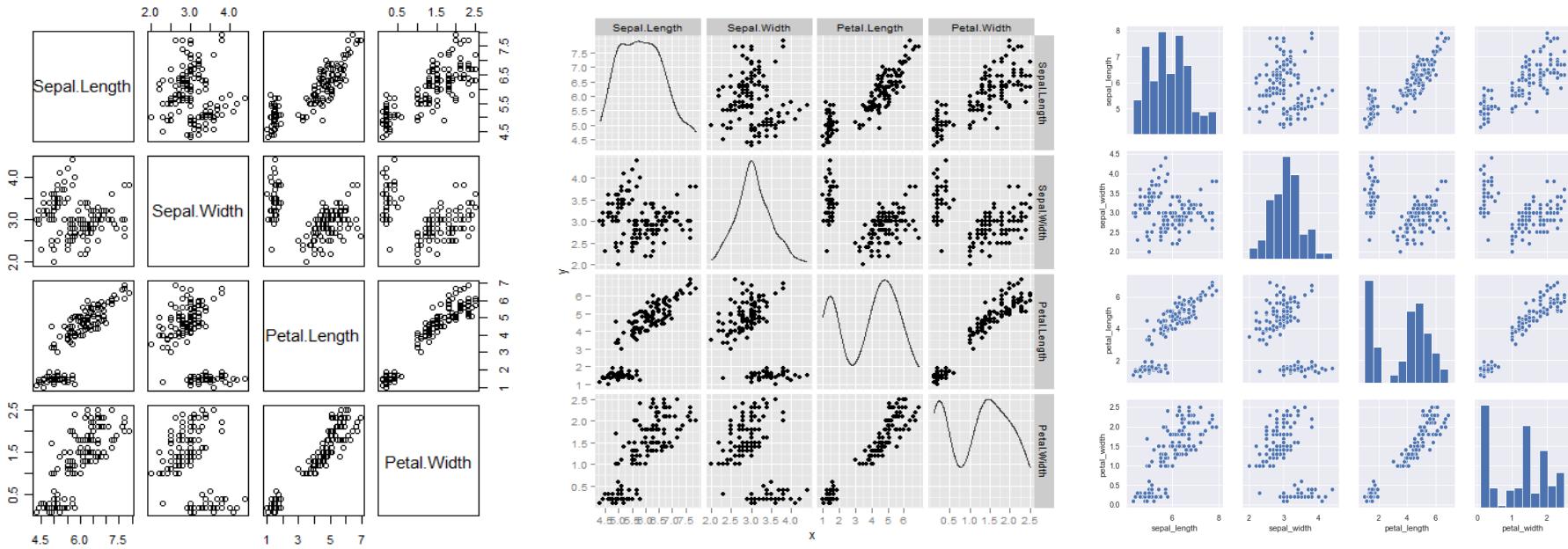
- Prečo predspracovanie dát ?
 - Množstvo dát – potreba výberu relevantnej časti
 - Chybné / nekonzistentné dáta, chýbajúce hodnoty
 - Chýbajúce (výška = nenameraný existujúci údaj) vs. Prázdné hodnoty (vlastníctvo auta => nevyplnenie má charakter hodnoty)
 - Zvýšenie efektivity, presnosti a uľahčenie procesu KDD
- Odpovedajúce typy dát (atribútov) – vid'. Analogicky ku atribútom z predchádzajúcich prednášok
 - Kvantitatívne (numerické) atribúty – číselné údaje, vektorové čísel, viacozmerné polia (obrázok po pixeloch)
 - Kategoriálne atribúty – popis kvalitatívnych vlastností
 - Ordinálne vs. Nominálne
 - Zložené typy dát – atribúty tvoriace hierarchiu konceptov

Pochopenie dát – atribúty

- Popis jednotlivých atribútov
 - Tabuľky početností, rozdelenie hodnôt (histogramy)
 - Boxplot = kvartily (Min, 1-Q, Med, 3-Q, Max), zovš. Percentily (kvantily) ... (napr. v R *boxplot*, *quantile*), priemer (mean), rozptyl / variancia (štandardná odchýlka), ...
 - Jednoduchá summarizácia o 1 atribúte, prípadne náhľad atribútu
 - Klasická summarizácia
 - R: **summary** (pre numerické atribúty vráti Min, 1-Q, Med, Mean, 3-Q, Max, pre diskrétné vráti početnosti)
 - Python: **describe** – štandardne pre numerické atribúty, pre zobrazenie aj iných typov treba nastaviť *include* parameter - **describe(include = 'all')**, prípadne použiť **value_counts** pre kategoriálne atribúty, ktorý tiež ukáže početnosti
 - Náhľad atribútov
 - **str** v R, **info** v Python pandas
 - Ďalšie charakteristiky (dosiahnutelné kombináciou niektorých z predchádzajúcich)
 - Šikmost – určuje či sú hodnoty okolo zvoleného stredu rozložené súmerne alebo sú hodnoty „zošikmené“ na jednu resp. druhú stranu (vid. Histogram), udáva či sú hodnoty okolo stredu rovnomerne na obe strany
 - Špicatost – určuje aký priebeh má graf rozdelenia hodnôt okolo zvoleného stredu rozdelenia, špicatejšie rozdelenie = viac sústredené okolo stredu

Pochopenie dát – vzťahy atribútov

- Globálny pohľad na závislosti medzi atribútmi
 - Scatterplot matrix – všetky atribúty voči sebe navzájom po dvojiciach
 - Hrubý pohľad na korelácie atribútov v množine dát
 - Funkcie - R: pairs, plot, splom (lattice), plotmatrix(ggplot2)
 - Python – napr. v seaborn pairplot
 - Príklad: iris data



Závislosti medzi atribútmi – korelácie

- Závislosť medzi numerickými atribútmi = korelačný koeficient, matica korelácií, funkcia *cor*, resp. *rcor* balíka Hmisc

Príklad: > x <- mtcars[1:3]; y <- mtcars[4:6]; cor(x, y)

	hp	drat	wt
mpg	-0.776	0.681	-0.868
cyl	0.832	-0.699	0.782
disp	0.791	-0.710	0.888

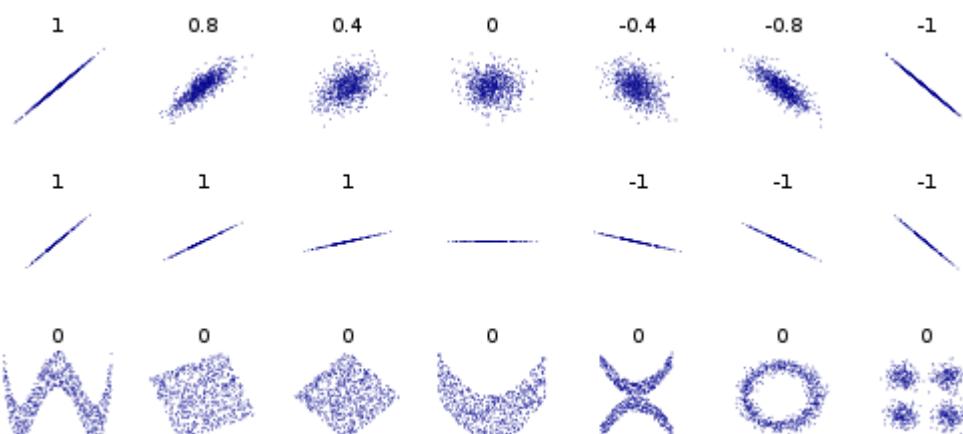
- Python – napr. *corrcoef* v numpy, pandas *corr*

- Sila korelácie

- Silná korelácia – absolútna hodnota korelácie je ≥ 0.8
- Stredná korelácia – absolútna hodnota korelácie je < 0.8 ale ≥ 0.5
- Slabá korelácia – absol.
hodnota korelácie < 0.5

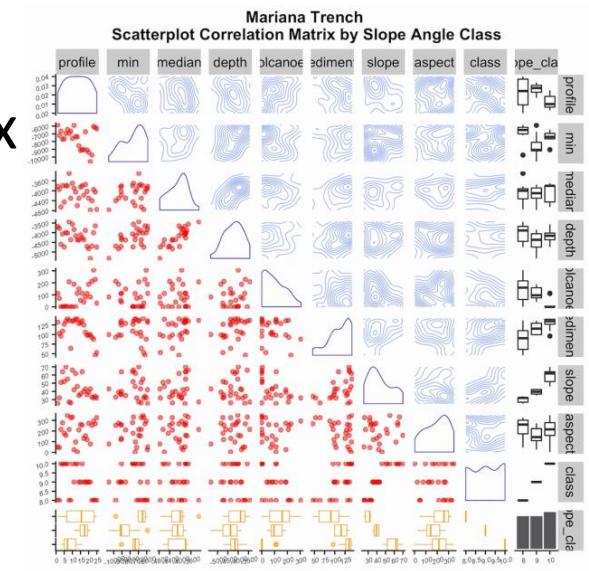
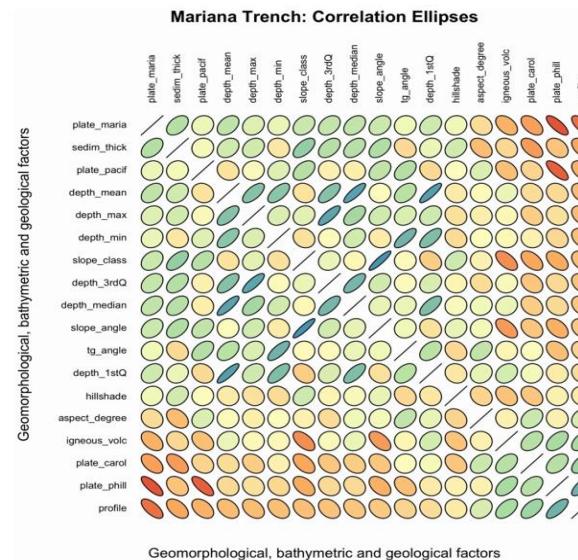
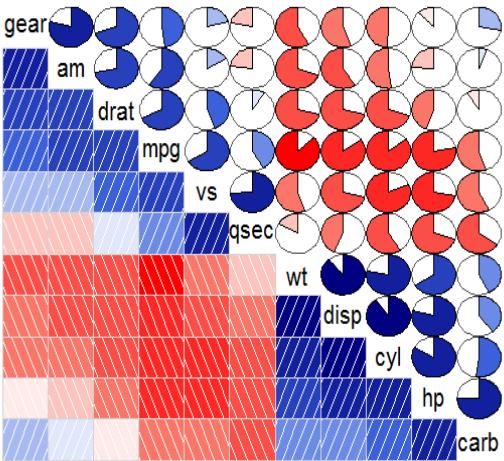
- Vzťah korelácie a
závislosti

- Priama a nepriama
úmera



Korelačné diagramy

- Špeciálny typ grafov pre korelácie = correlogram obdoba scatterplotmatrix (porovnanie atribútov)
- R – existuje **corrgram** balík, plotcorr v ellipse, ...
- Python – napr. pandas funkcia corr pre získanie hodnôt, následne heatmapy, scatterplot, či iné, môžeme skombinovať do rôznej informácie o koreláciách



Mariana Trench: Spearman correlation for the environmental factors

	profile	Min	1stQ	Median	Mean	3rdQ	Max	tg_angle	plate_phill	plate_carol	plate_maria	sedim_thick	depth_mean	slope_class	depth_min	depth_3rdQ	depth_median	slope_angle	lg_angle	depth_1stQ	hillshade	aspect_degree	igneous_volc	plate_phill	profile
profile	1	-0.25	-0.045	0.32	-0.36	-0.32	-0.42	-0.26	0.73	-0.38	-0.64	0.63	0.51	-0.47	-0.22	-0.24	-0.11	0.2							
Min	-0.25	1	0.63	0.22	0.55	-0.03	0.18	0.72	0.001	0.091	-0.08	0.2	-0.0089	0.0089	0.26	0.74	-0.2	0.072							
1stQ	-0.045	0.63	1	0.61	0.82	0.25	0.26	0.32	0.14	-0.28	0.068	0.2	0.3	0.18	-0.31	0.34	-0.16	0.09							
Median	-0.32	0.22	0.61	1	0.8	0.67	0.47	0.095	-0.18	-0.02	0.4	-0.29	0.018	0.49	0.2	0.1	0.1	0.15							
Mean	-0.36	0.55	0.82	0.8	1	0.68	0.65	0.37	-0.11	-0.11	0.3	-0.077	0.1	0.36	-0.042	0.35	-0.07	-0.017							
3rdQ	-0.32	-0.03	0.25	0.67	0.68	1	0.77	-0.01	-0.12	-0.09	0.37	-0.33	0.0019	0.44	0.21	0.04	0.058	-0.19							
Max	-0.42	0.18	0.26	0.47	0.65	0.77	1	0.27	-0.18	-0.04	0.25	-0.26	-0.14	0.36	0.26	0.210	0.230	0.074							
tg_angle	-0.26	0.72	0.3	0.059	0.37	-0.01	0.27	1	-0.34	0.48	0.14	0.19	-0.210	0.042	0.038	0.98	-0.26	0.35							
plate_phill	0.73	0.0017	0.14	-0.18	-0.11	-0.12	-0.18	-0.34	1	-0.71	-0.8	0.55	0.56	-0.38	-0.43	-0.27	-0.093	0.022							
plate_pacif	-0.38	0.091	-0.28	-0.02	-0.11	0.0950	0.0240	0.46	0.71	1	0.28	-0.2	-0.39	0.11	0.32	0.45	-0.058	0.28							
plate_maria	-0.84	-0.08	0.06	0.4	0.3	0.37	0.25	0.14	-0.8	0.28	1	-0.58	-0.4	0.48	0.32	0.1	0.054	0.091							
plate_carol	0.63	0.2	0.2	-0.29	-0.07	-0.33	-0.26	0.19	0.55	-0.2	-0.58	1	0.43	-0.6	0.41	0.21	-0.39	0.37							
igneous_volc	0.510	0.081	0.3	-0.018	0.1	0.0019	0.14	-0.21	0.56	-0.39	-0.4	0.43	1	0.18	-0.5	-0.25	-0.17	-0.2							
sedim_thick	-0.470	0.089	0.18	0.49	0.38	0.44	0.38	0.042	-0.38	0.11	0.48	-0.6	-0.18	1	0.49	0.031	0.14	-0.29							
slope_angle	-0.22	-0.26	-0.31	0.2	-0.042	0.21	0.260	0.033	0.43	0.32	-0.41	-0.5	0.49	1	-0.051	0.1	0.022								
slope_class	-0.24	0.74	0.34	0.1	0.35	-0.04	0.21	0.98	-0.27	0.45	0.1	0.21	-0.25	0.031	0.051	1	-0.24	0.38							
hillshade	-0.11	-0.2	-0.16	0.1	-0.07	0.058	0.0223	0.260	0.0930	0.058	-0.39	-0.17	0.14	0.1	-0.24	1	-0.3	-0.03	1						
aspect_degree	0.2	0.072	0.09	0.15	-0.017	0.19	-0.074	0.35	-0.022	0.28	-0.09	0.037	-0.2	-0.29	0.022	0.38	-0.3	0.1							

Závislosť kategoriálnych atribútov

- Na základe hodnôt z kontingenčnej tabuľky
 - h_{ij} – početnosť prípadov kombinujúcich x_i a y_j + marginálne súčty
 - Pre nezávislé atribúty platí $\frac{h_{ij}}{n} = \frac{h_{i\bullet}}{n} \cdot \frac{h_{\bullet j}}{n} \Rightarrow h_{ij} = \frac{h_{i\bullet} \cdot h_{\bullet j}}{n}$
 - Existuje viacero testov ... Chi-kvadrát test (`chisq.test()`) nezávislosti atribútov vychádza z globálnej charakteristiky G a jej štatistickej významnosti voči chi-kvadrát rozdeleniu v rámci dát (hladiny významnosti)

$x_i \backslash y_i$	y_1	\dots	y_m	
x_1	h_{11}	\dots	h_{1m}	$h_{1\cdot}$
x_2	h_{21}	\dots	h_{2m}	$h_{2\cdot}$
\dots	\dots	\dots	\dots	\dots
x_k	h_{k1}	\dots	h_{km}	$h_{k\cdot}$
	$h_{\cdot 1}$	\dots	$h_{\cdot m}$	n

$$G = \sum_{i=1}^k \sum_{j=1}^m \frac{(h_{ij} - \frac{h_{i\bullet} \cdot h_{\bullet j}}{n})^2}{h_{i\bullet} \cdot h_{\bullet j}}$$

Typické úlohy predspracovania dát v DM

- Často je predspracovanie časovo najnáročnejšou etapou procesu KDD a uskutočňuje sa opakovane (podľa aktuálnych potrieb), pričom zahŕňa:
- Čistenie dát
 - Spracovanie chýbajúcich hodnôt
 - Odstránenie / úprava výrazne odchýlených hodnôt
 - Odstránenie nekonzistentností v dátach (ručne alebo opravná rutina)
- Integráciu dát
 - Identifikácia rovnakých entít v rámci rôznych zdrojov dát
 - Odstránenie redundancií
 - Detekcia a odstránenie konfliktov v dátach
- Transformáciu dát
 - Odvodenie nových atribútov
 - Agregácia hodnôt (nad DS)
 - Generalizácia (využitie hierarchie konceptov pre zjednodušenie vstupu)
 - Normalizácia hodnôt
 - Diskretizácia (prevod numerických atribútov na kategoriálne)
- Redukcia (výber = selekcia)
 - Redukcia dimenzionality – selekcia atribútov
 - Redukcia početnosti dát – selekcia objektov (vzoriek)

Čistenie dát

- Spracovanie chýbajúcich hodnôt – riešenia:
 - Nič nerobiť (niektoré algoritmy to zvládnu)
 - Vynechanie záznamu z analýzy
 - Doplnenie hodnoty
 - Manuálne (napr. expertom), konštantou „neznáma“ hodnota (NA), jednoduchou štatistikou (priemer, medián,...), predikovanou hodnotou (predikčný model)
- Spracovanie chybných hodnôt (výrazných odchylok) = vyhladenie šumu
 - Binning (lokálne vyhladzovanie) ... Optimálne rozdelenie dát na biny ... určenie napr. v R cez **quantile** + **cut**, prípadne špeciálne balíky (napr. **smbinning** pre scoring modely)
 - Usporiadanie dát a úprava na rovnakú hĺbku/šírku intervalu
 - Pre všetky intervale sa môže nahradíť hodnota jeho stredom, mediánom, min, max
 - Regresia (prediktívny model z ktorého vyberieme hodnotu)
 - Zhlukovanie (skupiny objektov podľa podobných hodnôt => z nich vyberieme náhradu pre chybnú hodnotu v atribúte)

Integrácia dát

- Integrácia viacerých zdrojov (tabuľiek) do jednej databázy (tabuľky)
- Problémy
 - Redundancia – odstránenie odvoditeľných atribútov
 - Určenie ekvivalentných entít z viacerých zdrojov
 - Detekcia a riešenie konfliktov v hodnotách atribútov = rozdiely v dátach na úrovni hodnôt v jednotlivých tabuľkách
 - napr. rôzne kódovanie, jednotky, vyjadrenia hodnôt
- Často je dôležité prihliadať už aj na metódu DM alebo cieľ

Transformácia dát

- Sumarizácia / agregácia / vyhľadzovanie dát
 - Vytvorenie / nahradenie atribútu resp. hodnôt agregovanými hodnotami (napr. s použitím DS)
 - Vyhľadzovanie ... Využívané pre čistenie, ale aj ako prostriedok transformácie dát pre potreby algoritmu
 - Binning, regresia (predikcia), zhlukovanie, ...
- Generalizácia
 - V prípade atribútov s hierarchickou štruktúrou => posun vyššie v hierarchii pre získanie všeobecnejšej úrovne (posun v hierarchii konceptov)
 - Dáta na nižšej úrovni sú nahradené vyššou úrovňou (napr. ulica → mesto)
- Normalizácia
 - Hodnoty atribútu sú upravené na zvolenú „normalizovanú“ formu, napr. preškálované na iný rozsah (napr. interval $<-1,1>$ alebo $<0,1>$), štandardizované vzhľadom na strednú hodnotu a štandardnú odchýlku, ...
 - Min-max normalizácia – z intervalu $<\min_A, \max_A>$ do intervalu $<\text{new_min}_A, \text{new_max}_A>$

$$v' = \frac{v - \min_A}{\max_A - \min_A} \cdot (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

• Normalizácia na nulový stred
(tzv. štandardizácia alebo z-score normalizácia)
– odpočítame od strednej hodnoty a predelíme štandardnou odchýlkou

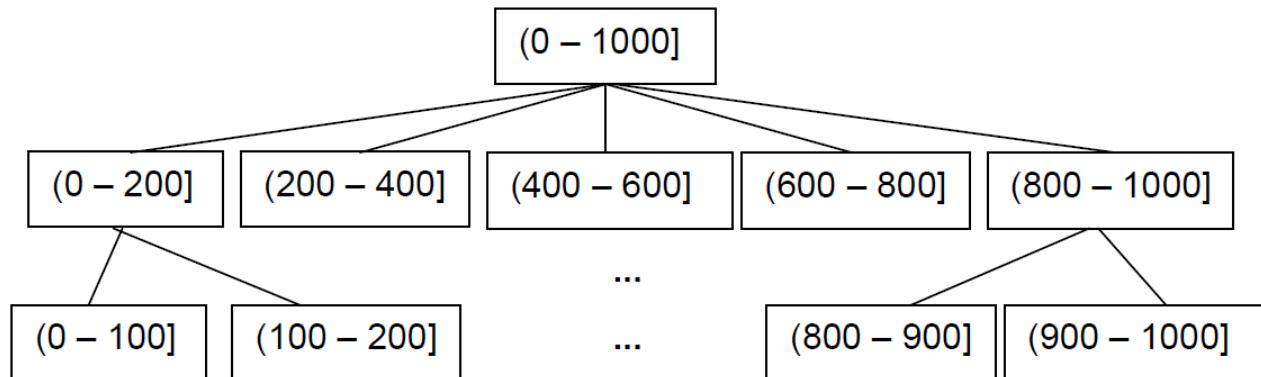
$$v' = \frac{v - \bar{x}_A}{s_A}$$

• Normalizácia decimálnym škálovaním $v' = \frac{v}{10^j}$ pričom $j=\text{najmenšie celé číslo také že } \text{Max}(|v'|) < 1$

• Normalizácia záznamu ako vektora na jednotkovú veľkosť

Transformácia dát - diskretizácia

- Diskretizácia hodnôt numerických atribútov
 - Predstavuje aj redukčnú metódu, hierarchia konceptov predstavuje takisto prípad diskretizácie
 - Ručne definovanie (najmä hierarchií) je náročné => používajú sa automatické postupy (binning, zhlukovanie, 3-4-5 pravidlo, ...)



- Metódy
 - Ekvidistančné
 - Ekvifrekvenčné
 - Pokročilé metódy

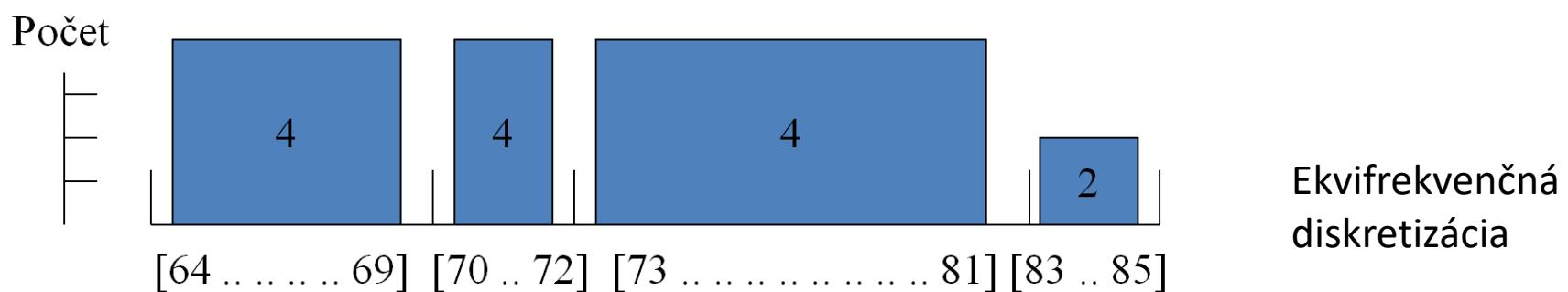
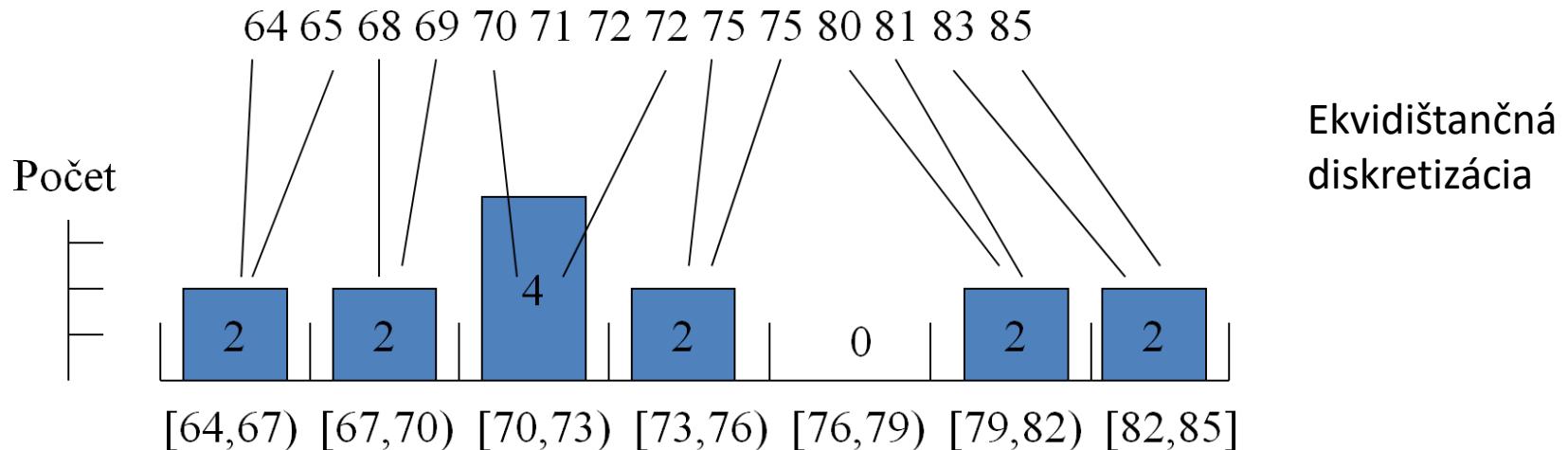
Diskretizácia v R

- `cut`, `findInterval` funkcie

- Pokročilejšie balíky `discretization`, `infotheo`

Diskretizácia Python – napr. pandas `cut` funkcia, funkcie na diskretizáciu v scikit `learn.preprocessing`

Príklady – Diskretizácia



Redukcia dát

- Agregácie = redukcia dát cez summarizáciu
 - napr. aj pomocou dátovej kocky cez OLAP vieme urobiť
- Redukcia rozmerov (dimenzií) problému => redukcia atribútov
 - Odstránenie redundantných, nadbytočných (nerelevantných) a nepoužívaných atribútov
 - Vytvorenie redukovanej množiny atribútov pomocou transformácie (LSI = Latent Semantic Indexing)
- Kompresia dát = zmenšenie objemu dát
- Redukcia početnosti dát = nahradenie dát alternatívou menšou reprezentáciou
 - Parametrické
 - Príklad: Namiesto dát sa použijú len parametre lineárnej regresie + príklady s najväčšou odchýlkou od modelu
 - Neparametrické = vzorkovanie (sampling)
 - Histogramy (binning), zhlukovanie
 - Prístupy k vzorkovaniu: Náhodná vzorka veľkosti N bez nahradenia (alebo s nahradením), Rozvrstvená vzorka (stratified sample), Náhodné vzorkovanie s konvergenciou variability (pre numerické atribúty)

Podniková analytika

Modelovanie I

Obsah

- Prediktívne dolovanie v dátach
 - Základný prístup
 - Tvorba a testovanie modelov
- Prehľad vybraných metód dolovania v dátach
 - Predikcia
 - Regresné modely
 - Klasifikácia
 - Rozhodovacie stromy
 - Naive Bayes
 - kNN

Prediktívne dolovanie v dátach

- Prediktívny DM = kontrolované učenie na dátach
 - Klasifikácia – modelovanie a predikovanie nominálneho atribútu = triedy
 - Predikcia – modelovanie a predikovanie numerických hodnôt (atribútov)
- Základný prístup pre prediktívny DM
 - Fáza 1 – Trénovanie – vybudovanie modelu na základe dát z trénovacej množiny
 - Fáza 2 – Testovanie – vytvorený model sa používa na predpovedanie hodnoty cieľového atribútu pre testovacie objekty, t.j. pre objekty ktoré sme vyčlenili z trénovania, ale majú informáciu o cieľovej hodnote - pre potreby nezávislého testovania = testovacia množina (aby sme vedeli vybrať vhodný model na finálne nasadenie)
 - Fáza 3 – Nasadenie – najlepší model (natrénovaný vo Fáze 1 tak aby bol čo najlepší v testovaní vo Fáze 2) nasadíme a do neho už vstupujú príklady u ktorých nevieme vôbec cieľovú triedu a verzia nasadeného modelu sa používa na skutočne neznáme prípady
- Príklady
 - Klasifikácia žiadateľov o hypotéku => cieľový atribút má napr. hodnotu poskytnúť / neposkytnúť
 - Predikcia predajov => cieľový atribút je počet predaných výrobkov v závislosti od času, reklamy, ...

Predikcia / Regresia

- Predikcia (Regresia) je teda DM metóda, ktorá:
 - konštruuje predikčný (regresný) model
 - používa tento model na predikciu numerických hodnôt cieľového atribútu nových prípadov (neznámych), alebo doplnenie chýbajúcich hodnôt známych príkladov
- (Niektoré) používané metódy:
 - Lineárna regresia a viacnásobná regresia (model vo forme lineárnej funkcie)
 - Nelineárna regresia
 - Modelové (regresné) stromy – po častiach lomená lineárna funkcia (kombinácia rozdelenia na podčasti dát a lineárnej regresie na jednotlivých úsekokoch)
 - Učenie založené na inštanciách (kNN – k najbližších susedov)
 - Neurónové siete

Lineárna regresia

- Regresia je proces vytvorenia funkcie nezávislých premenných (tzv. prediktorov) pre predikciu závislých premenných („response“)
- Lineárna regresia predikuje výstupné hodnoty premennej y na základe lineárneho modelu => dáta sú aproximované pomocou priamky
- Jednoduchá dvojrozmerná verzia => výstup y (predikovaný atribút) je modelovaný pomocou jednej premennej x (predikujúci atribút)
 - Ide o rovnici priamky, kde

$$y = \alpha + \beta x$$

- β – smernica priamky
- α – posun priesečníka s osou y oproti nule

Výpočet modelu lineárnej regresie

- Regresné koeficienty α, β môžeme určiť napr. použitím metódy najmenších štvorcov (minimalizácia chyby medzi dátami a hľadanou priamkou)
- Trénovacie dáta pre LR => body $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ => pre výpočet (štatistický odhad) regresných koeficientov a, b potom môžeme použiť nasledujúce vzťahy:

$$\alpha = \bar{y} - \beta \bar{x} \quad \beta = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Príklad – lineárna regresia

Počet rokov praxe	3	8	9	13	3	6	11	21	1	16
Príjem	30	57	64	72	36	43	59	90	20	83

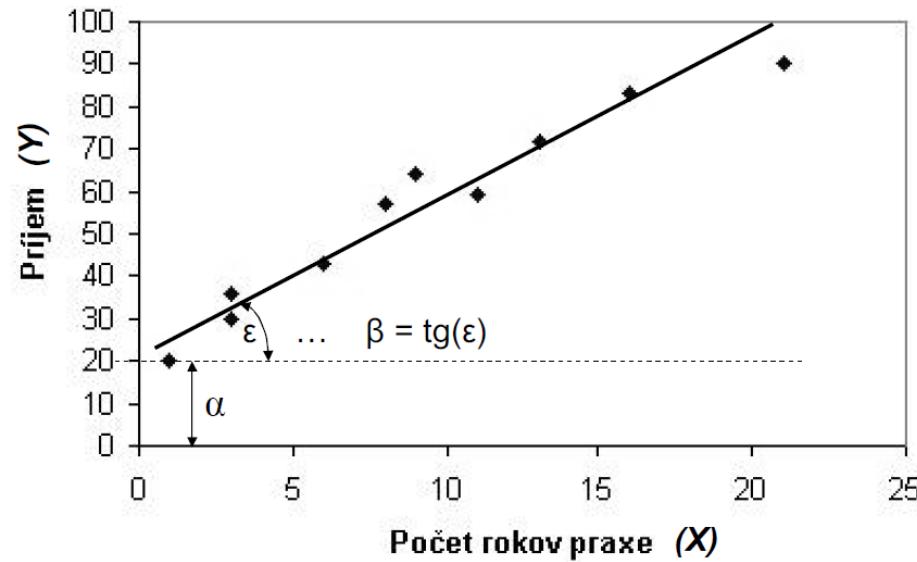
$$\bar{x} = 9.1, \bar{y} = 55.4$$

$$\beta = \frac{(3-9.1) \cdot (30-55.4) + (8-9.1) \cdot (57-55.4) + \dots + (16-9.1) \cdot (83-55.4)}{(3-9.1)^2 + (8-9.1)^2 + \dots + (16-9.1)^2} = 3.5$$

$$\alpha = 55.4 - (3.5) \cdot (9.1) = 23.6$$

$$Y = 23.6 + 3.5X$$

Model môžeme následne použiť na predikovanie hodnôt príjmu ... napr. absolvent danej vysokej školy s 10-ročnou praxou by mal zarábať okolo 58.600,- dolárov ročne



Viacnásobná a polynomiálna regresia

- Viacnásobná regresia = rozšírenie základnej lineárnej verzie o ďalšie predikujúce atribúty (viacrozmerný lineárny model)
- Predpokladáme, že y je lineárne závislý na k predikujúcich atribútoch x_1 až $x_k \Rightarrow$ hľadáme lineárny model v $(k+1)$ -rozmernom priestore v tvare

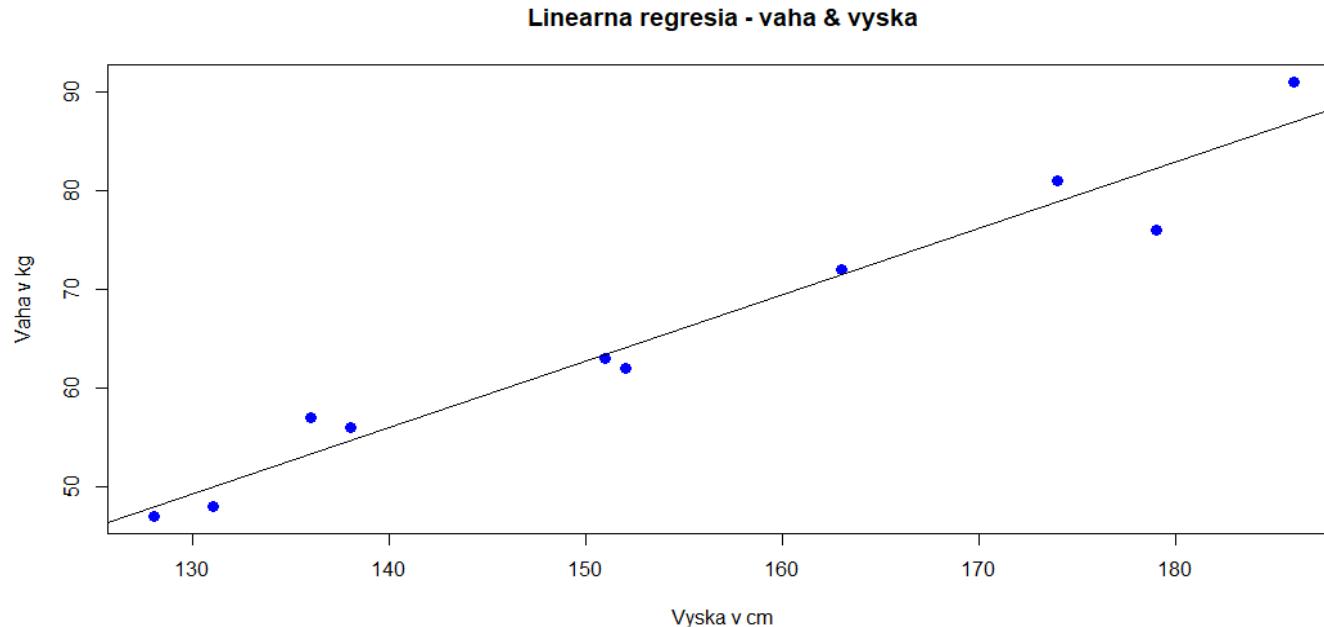
$$y = c_0 + c_1x_1 + c_2x_2 + \cdots + c_kx_k$$

- Riešenie v R – Použitie funkcie `lm()`
- Python - scikit-learn, `LinearRegression` v `sklearn.linear_model`
- Polynomiálna regresia = pridanie polynomiálnych termov k základnému lineárному modelu
 - Aplikovaním transformácií na jednotlivé nelineárne členy polynómu ju je možné konvertovať na lineárny model

$$Y = \alpha + \beta_1X + \beta_2X^2 + \beta_3X^3 \longrightarrow X_1 = X, X_2 = X^2, \quad X_3 = X^3 \longrightarrow Y = \alpha + \beta_1X_1 + \beta_2X_2 + \beta_3X_3$$

Príklad – dáta

```
tab = data.frame("vyska" = c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131),  
                 "vaha" = c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48))  
model = lm(vaha ~ vyska, data = tab)  
with(tab, plot(vyska,vaha,col = "blue",main = "Linearna regresia - vaha & vyska",  
               cex = 1.3,pch = 16,xlab = "Vyska v cm",ylab = "Vaha v kg"))  
abline(model)
```



Popis modelu a predikcia hodnôt

- Model Vybudovanie lineárneho modelu

```
model = lm(vaha ~ vyska, data=tab)
```

```
model
Call:
lm(formula = vaha ~ vyska, data = tab)
```

```
Coefficients:
(Intercept)      vyska
-38.4551        0.6746
```

Viac info o modeli – `attributes()`, `summary()`, `residuals()`

- Tento model môžeme použiť na predikciu

- Priamo použitím koeficientov

- Cez funkciu `predict()` ... Príklad: chceme zistiť hodnotu váhy pre výšku 170

```
nove = data.frame(vyska = 170)
```

```
predict(model,nove) # výsledok predikcie = 76.22869
```

- Samozrejme, môžeme predikovať pre viac hodnôt, napr. tak že vložíme viac hodnôt výšky

```
nove2 = data.frame(vyska=c(170,145,189))
```

```
predict(model,nove2)
```

1	2	3
---	---	---

76.22869	59.36343	89.04629
----------	----------	----------

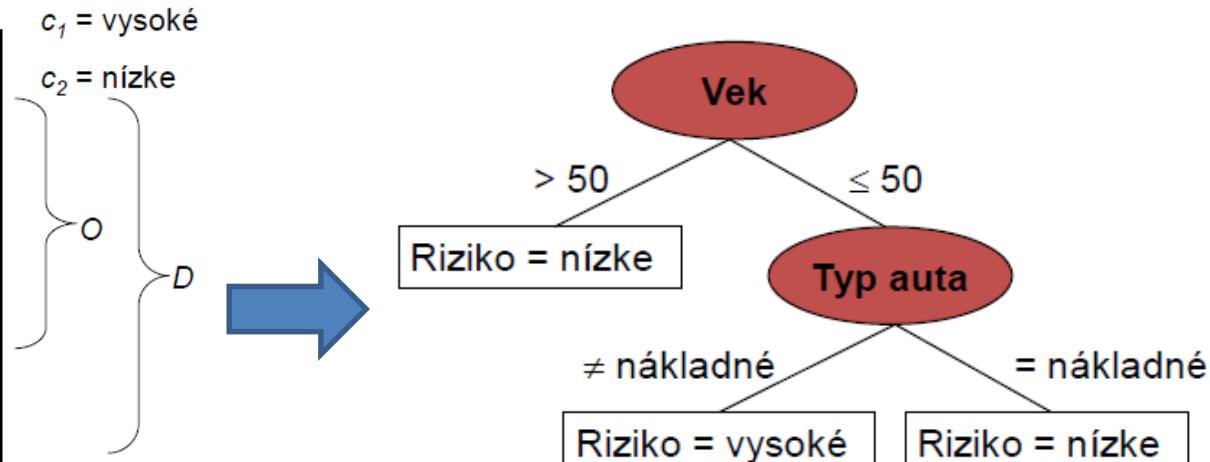
Klasifikácia

- Klasifikačná úloha predstavuje proces vytvorenia modelu pre predikciu atribútu triedy (vopred známych hodnôt)
 - Základné prvky: Trénovacie dátá, Klasifikátor (model), Testovacie dátá
 - Základné kroky: Konštrukcia modelu (učenie), Vyhodnotenie modelu (testovanie), Klasifikácia nových príkladov (nasadenie / aplikácia)
- Konštrukcia modelu
 - Množina príkladov použitá pre konštrukciu klasifikačného modelu = **trénovacia množina (TM)**
 - Vytvorený model môže byť reprezentovaný v rôznej forme:
 - Logické konjunkcie (VSS, EGS, HGS ...), Rozhodovacie stromy (ID3, C4.5, ID5R, ...), Rozhodovacie zoznamy (NEX, CN2, RISE, ...), Pravdepodobnostný popis (Naivný Bayes, Bayesovské siete, ...), Neurónové siete + inštančné prístupy (kNN), ...
 - Optimalizácia parametrov modelu (voliteľné)
 - Pre tento krok sa používa **validačná množina** dát (musí byť nezávislá od TM)
- Vyhodnotenie modelu (testovanie) – klasifikácia „neznámych“ prípadov
 - Odhad presnosti modelu = vyhodnotenie modelu = porovnanie skutočného zaradenia testovacích príkladov s klasifikáciou na základe vygenerovaného modelu (**testovacia množina**)
 - Presnosť sa vyjadruje napr. percentuálnym podielom správne klasifikovaných testovacích príkladov (existujú viaceré metriky, podrobnejšie v ďalšej prednáške)
 - Testovacia množina musí byť nezávislá od trénovacej a validačnej množiny

Príklad

- Klasifikátor = model = funkcia $K: D \rightarrow C$
 - C (tryedy ci), D množina vš. objektov (z toho časť O má priradenú triedu z C , odkiaľ sú trén., test. či valid. množ.)

ID (A_1)	Vek (A_2)	Typ auta (A_3)	Riziko (C)
1	23	rodinné	vysoké
2	17	športové	vysoké
3	43	športové	vysoké
4	68	rodinné	nízke
5	32	nákladné	nízke
6	35	rodinné	
7	58	rodinné	



- Rozhodovacie stromy
 - Jednoduché pre pochopenie a prezentáciu
 - Medziľahlý uzol = vybraný popisný atribút
 - Listový uzol = hodnota atribútu triedy
 - Hrana = test vybraného atribútu – výstupné hrany daného uzla pokrývajú všetky možnosti hodnôt, pričom jednotlivé vetvy sú disjunktné
 - Postupnosť testov od koreňového uzla po listový predstavuje zároveň rozhodovacie (klasifikačné) pravidlo

Základný postup pri tvorbe RS

- Výber trénovacej (TM) a testovacej (TSM) (prípadne validačnej VM) množiny
 - Rozdelenie príkladov z O – napr. 70% TM a 30% TSM (existuje viacero prístupov ako rozdeľovať a následne testovať)
- Učenie rozhodovacieho stromu (RS):
 - 1. Začíname s celou množinou (TM)
 - 2. Výber testovacieho atribútu (podľa zvoleného kritériá)
 - 3. Rozdelenie do častí podľa hrán + test ukončenia pre každú
 - Ak podčasť splňa podmienku => listový uzol s triedou
 - Inak pre danú podčasť dát opakujeme od kroku 2
- V prípade potreby použijeme VM a optimalizujeme parametre príslušného algoritmu RS
- Vyhodnotenie modelu
 - Aplikujeme naučený model RS na príklady z TSM
 - Vyhodnotíme úspešnosť priradenia príkladov klasifikátorom do tried voči ich skutočnému zaradeniu = kontingenčná tabuľka + presnosť (resp. chyba) klasifikácie

Výber testovacieho atribútu

- Existujú rôzne kritériá, najpoužívanejší princíp je založený na **entropii = neurčitosť**
 - V rámci teórie informácií existuje prístup na meranie „množstva“ informácie cez entropiu
- RS => každý uzol je možné z pohľadu jemu odpovedajúcich dát vyhodnotiť podľa entropie
 - Čím rozdielnejšie sú príklady (tryedy) v uzle, tým je jeho entropia H vyššia (pre pravdepodobnosti sa používa odhad počet ci / počet vš.)
 - Testovací atribút vyberáme tak, aby sme čo najviac znížili entropiu v poduzloch (výsledok: najvyššia entropia = root, najnižšia = listy)
 - Entropia uzlu S pred rozdelením $H(S) = -\sum_{j=1}^n p(c_j) \log_2(p(c_j))$
 - Entropia uzlu S (m vetiev s_j) pre výber atribútu $A_i \longrightarrow H(S, A_i) = \sum_{j=1}^m p(s_j)H(S_j)$
 - Potom klasické kritérium je výber atribútu s najvyšším informačným ziskom (Information Gain = IG), čiže najlepšou schopnosťou zmenšiť entropiu $I(S, A_i) = H(S) - H(S, A_i)$
 - V praxi: používajú sa rozšírenia pre zlepšenie kvality a možností klasifikácie ako pomerový informačný zisk, testovanie spojitych atribútov, riešenie chýbajúcich hodnôt, ... + existujú aj iné kritériá (GINI)
 - Používané algoritmy: ID3, C4.5, CART (GINI)

Príklad – Rozhodovacie stromy v R

- IRIS dáta
- Pripravíme si trénovaciu a testovaciu časť (70% na 30%, indikátor rozdelenia cez sample)

```
set.seed(1234)
ind <- sample(2, nrow(iris), replace = TRUE, prob = c(0.7, 0.3))
train.data <- iris[ind == 1, ]
test.data <- iris[ind == 2, ]
```
- Existujú rôzne implementácie použiteľné pre tvorbu klasifikačných stromov (balíky v R ako rpart, tree, party, partykit, maptree, randomForest, C5.0, python – sklearn.tree, ...)
- Použijeme napr. **party** balík a metódu *ctree* pre rekurzívnu tvorbu klasifikačných stromov

Príklad – RS cez ctree (party) na IRIS dátach

```
library(party)
```

```
myFormula <- Species ~ Sepal.Length + Sepal.Width +  
Petal.Length + Petal.Width
```

```
iris_ctree <- ctree(myFormula, data = train.data)
```

- Overenie na trénovacej množine + print

```
table(predict(iris_ctree), train.data$Species)  
print(iris_ctree)
```

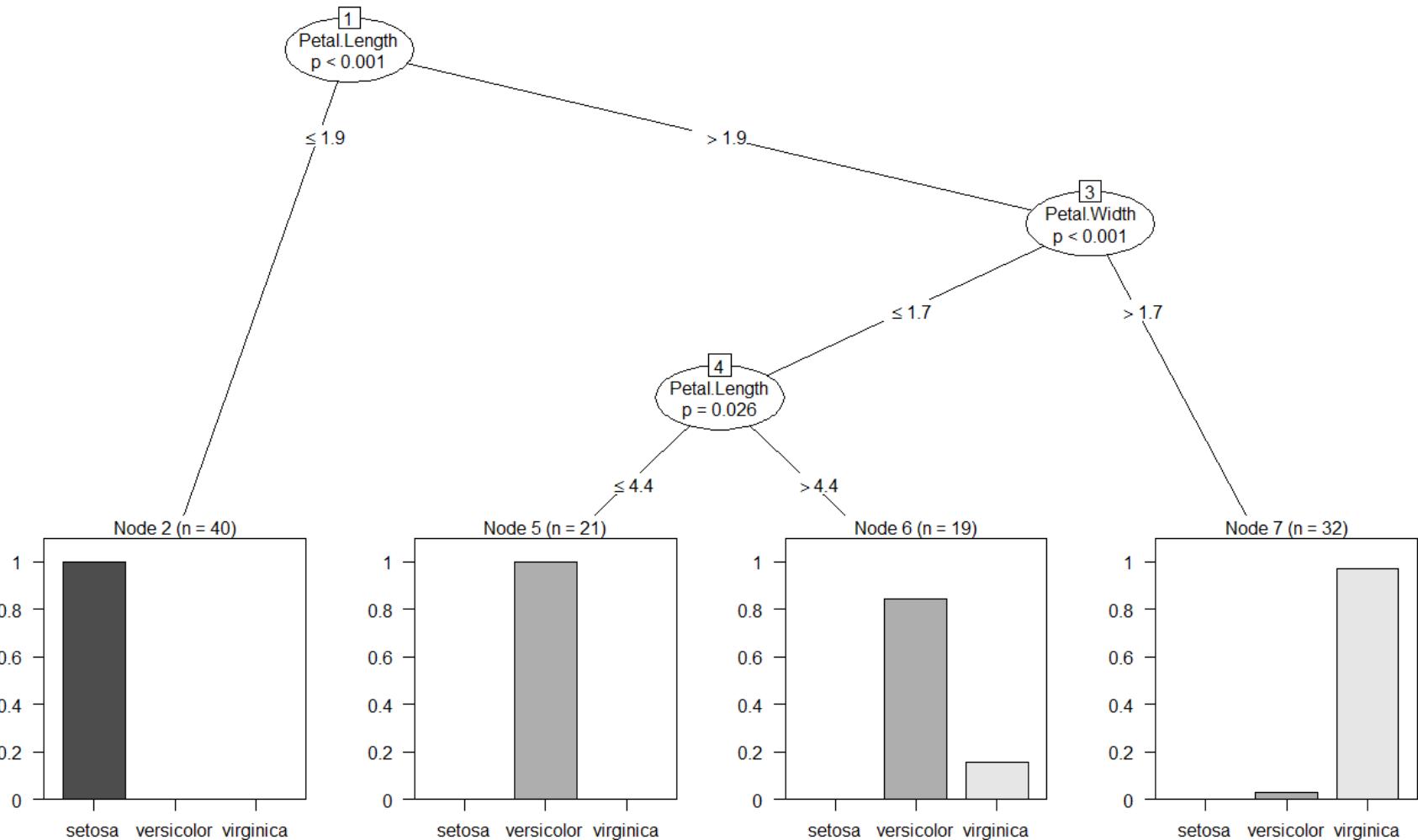
```
Conditional inference tree with 4 terminal nodes
```

```
Response: Species  
Inputs: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width  
Number of observations: 112  
1) Petal.Length <= 1.9; criterion = 1, statistic = 104.643  
2)* weights = 40  
1) Petal.Length > 1.9  
3) Petal.Width <= 1.7; criterion = 1, statistic = 48.939  
4) Petal.Length <= 4.4; criterion = 0.974, statistic = 7.397  
5)* weights = 21  
4) Petal.Length > 4.4  
6)* weights = 19  
3) Petal.Width > 1.7  
7)* weights = 32
```

	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	37	3
virginica	0	1	31

Príklad – výstupný strom (cez plot)

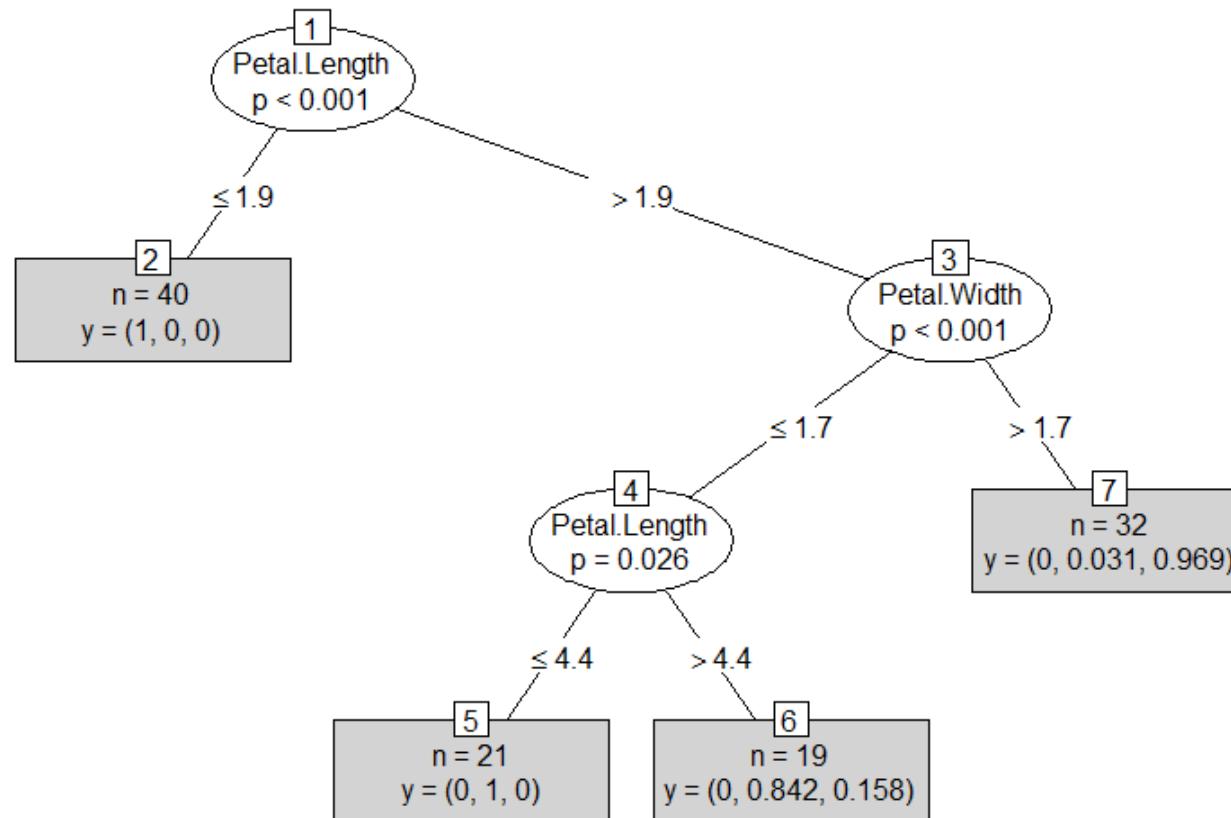
`plot(iris_ctree)`



Príklad – výstupný strom (cez plot)

Zjednodušená verzia

```
plot(iris_ctree, type="simple")
```



Príklad – overenie na testovacej množine

```
testPred <- predict(iris_ctree, newdata =  
test.data)  
table(testPred, test.data$Species)
```

testPred	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	12	2
virginica	0	0	14

Pravdepodobnosť prístup – NB

- Bayesovská klasifikácia – určenie pravdepodobnosti že príklad patrí do triedy
- Základ: Výpočet podmienených pravdepodobností
- $p(c_i | X)$... hľadáme podmienenú pravdepodobnosť triedy c_i za predpokladu vektora atribútov X
- Existuje Bayesovo pravidlo $\longrightarrow p(c_i / X) = \frac{p(c_i)p(X / c_i)}{p(X)}$
 - $p(c_i)$... určíme ako podiel početnosti triedy c_i
 - nakoľko $p(X)$ je v menovateli, stačí určiť už len $p(X | c_i)$
 - **Naivný Bayesov (NB) klasifikátor** – predpoklad nezávislosti atribútov objektu X ($x_1, x_2, \dots, x_k, \dots$) nám dovoľuje výpočet $p(X / c_i) = \prod p(x_k / c_i)$, pričom jednotlivé $p(x_k | c_i)$ už vieme určiť k je to (pre kategoriálne atribúty) pomerová početnosť príkladov triedy c_i majúcich hodnotu x_k (t.j. nakoľko c_i ovplyvňuje hodnoty atribútov)
- Pre prípad numerických atribútov – diskretizácia a použitie kategorickej hodnoty, alebo sa predpokladá normálne (Gaussovo) rozdelenie hodnôt atribútu

Príklad – NB klasifikácia

Chceme klasifikovať príklad
Outlook=sun, Temp=cool,
Humid=high, Wind=strong

$$p(\text{yes}) = 9/14 \quad p(\text{no}) = 5/14$$

$$p(\text{wind=strong} | \text{yes}) = 3/9$$

$$p(\text{wind=strong} | \text{no}) = 3/5$$

$$p(\text{outlook=sun} | \text{yes}) = 2/9$$

....

Po určení všetkých hodnôt dostávame (zjednodušene yes=y, no=n + iba hodnota pri podm. pravdepodobnosti):

$$\begin{aligned} p(y|X) &= p(y) * p(\text{sun}|y) * p(\text{cool}|y) * p(\text{high}|y) * p(\text{strong}|y) = \\ &= (9/14) * (2/9) * (3/9) * (3/9) * (3/9) = 0.00529 \end{aligned}$$

$$\begin{aligned} p(n|X) &= p(n) * p(\text{sun}|n) * p(\text{cool}|n) * p(\text{high}|n) * p(\text{strong}|n) = \\ &= (5/14) * (3/5) * (1/5) * (4/5) * (3/5) = 0.02057 \dots \text{vybraná trieda} \\ &\text{je „no“} \end{aligned}$$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Naive Bayes klasifikácia

- Použitie Naive bayes
 - R napríklad balíky **e1071**, **klaR**
 - Python - `sklearn.naive_bayes` napr. `GaussianNB`
- Príklad: IRIS dát – model na báze NB v R a jeho kontingenčná tabuľka (len pre trénovacie dát)

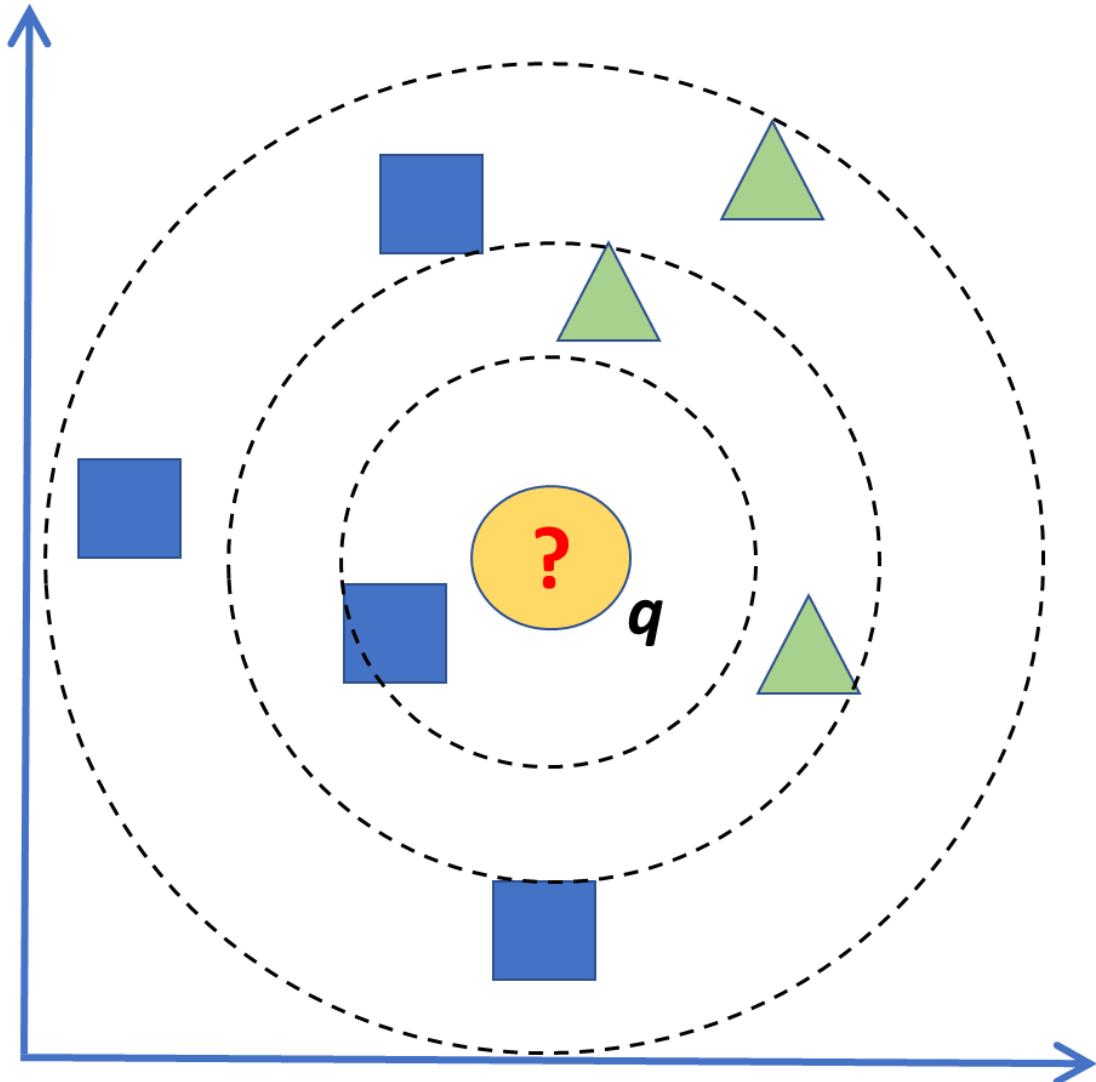
```
library(e1071)
classifier<-naiveBayes(iris[,1:4], iris[,5])
table(predict(classifier, iris[,-5]), iris[,5])
```

	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	47	3
virginica	0	3	47

Inštančné učenie (klas., pred.)

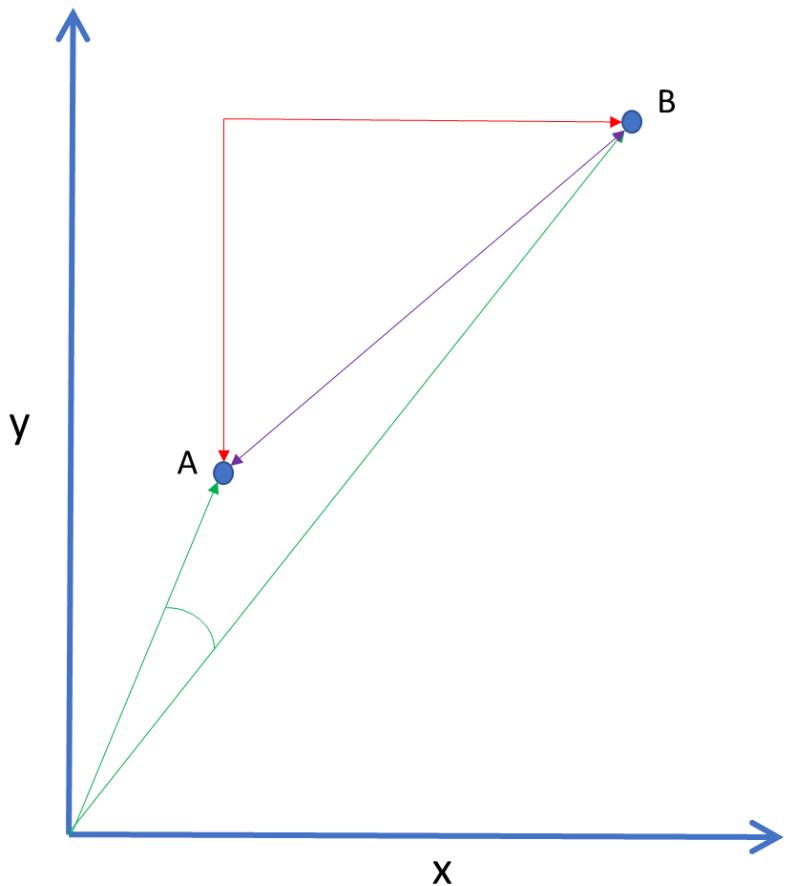
- Učenie založené na inštanciách (tzv. lenivé učenie)
 - Budujeme „model“ až v momente keď je to treba => pozeráme sa na dátu až pri samotnej klasifikácii/predikcii (neexistuje model vytvorený pred samotným výpočtom)
 - kNN = k Nearest Neighbors = k najbližších susedov
 - R - funkcie `knn`, `knn.predict` + balík `knnflex`
 - Python – `sklearn.neighbors ... KNeighborsClassifier`
 - Výsledná klasifikácia/predikcia sa určuje pomocou hodnôt cieľového (klasifikačnej triedy / predikovaného atribútu) k-najbližších susedov (príkladov) z trénovacej množiny ku práve klasifikovanému / predikovanému príkladu
 - Výsledná predikcia
 - kategorického atribútu = majoritná trieda medzi susedmi
 - numerického atribútu = ako (vážený) priemer hodnôt k-najbližších susedov
 - Normalizácia atribútov dát môže pomôcť pri výpočte vzdialenosí – pri nenormalizovaných dátach môže jeden atribút dominovať nad ostatnými
 - Výška osoby – v rozsahu od 1.5 do 1.9 (údaje v metroch)
 - Váha osoby – v rozsahu od 50 do 120 (údaje v kg)
 - Ročný príjem osoby – v rozsahu od 5000 do 150 000 (údaje v mene)

kNN - výber hodnoty k



- $k=1 \rightarrow$
- $k=3 \rightarrow$
- $k=7 \rightarrow$
- Dôležitá vol'ba parametra k
 - veľmi malé hodnoty – citlivosť na šum
 - veľmi veľké hodnoty – možnosť zahrnúť aj príklady inej triedy
 - pre binárnu klasifikáciu – dobré voliť nepárne hodnoty k

kNN - metriky



- **Euklidovská metrika** – vzdialenosť v priestore príznakov

$$Eucl(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

- **Kosínusová metrika** – zohľadňuje uhol medzi vektormi, Kosínusová metrika sa používa, keď magnitúda vektorov nie je podstatná (napr. v textových dátach – počet výskytu slov)

$$CosSim(x, y) = \frac{\sum_{i=1}^d x_i y_i}{\sqrt{\sum_{i=1}^d x_i^2 \sum_{i=1}^d y_i^2}}$$

- **Manhattan metrika**

$$dist(x, y) = Manh(x, y) = \sum_{i=1}^d |x_i - y_i|$$

Príklad – kNN v Python

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

# nacitanie iris dat
irisData = load_iris()

# data a target obsahuju v tomto pripade presne prve styri predikujuce atributy a cielovy atribut
X = irisData.data
y = irisData.target

# Rozdelenie trenovacia a testovacia mnozina 80 ku 20
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.2, random_state=42)

knn = KNeighborsClassifier(n_neighbors=7) # nastavenie na algoritmus kNN a nastavenie poctu susedov

knn.fit(X_train, y_train) # spustenie algoritmu

# predikcia na testovacej mnozine – na datche ktore kNN nepouzila na trenovanie (teda na vytvorenie
# kNN klasifikatoru)
print(knn.predict(X_test))
[1 0 2 1 1 0 1 2 2 1 2 0 0 0 1 2 1 1 2 0 2 0 2 2 2 2 2 0 0]
```

Podniková analytika

Modelovanie II

Obsah

- Vyhodnotenie klasifikácie
 - Základné prístupy hodnotenia
 - Vybrané mierky hodnotenia, kontingenčná tabuľka vyhodnotenia klasifikácie (confusion matrix)
- Zlepšenie klasifikácie
- Deskriptívne dolovanie v dátach
 - Zhlukovanie
 - Asociačné pravidlá

Vyhodnotenie modelov klasifikácie

- Pre vyhodnotenie sa používa chyba klasifikácie, t.j. podiel chybne klasifikovaných objektov (alebo úspešnosť klasifikácie = podiel správne klasif. objektov)
- Pre zamedzenie preučenia sa delí množina na trénovaciu a testovaciu časť => napr. 70% ku 30% náhodne rozdelenej množiny, 80/20, 90/10, ...
- Zložitejší model pre validáciu
 - m-násobná krížová validácia (cross-validation) = rozdelí množinu O na m rovnako veľkých podmnožín, následne použije m-1 podmnožín na trénovanie a zvyšnú na testovanie => výsledný odhad chyby = priemer z m validácií
 - Ak rozdelenie nie je náhodné, ale zachováva distribúciu tried => rozstratená krížová validácia (stratified cross-validation)
 - Zaužívané odporúčanie: 10-násobná krížová validácia

Kontingenčná tabuľka klasifikácie (confusion matrix)

Konting. tabuľka klasifikácie (confusion matrix)	V skutočnosti je z danej triedy	V skutočnosti nie je z danej triedy
Klasifikátor tvrdí že je to daná trieda (pozitívna identifikácia)	TP (počet „True Positive“ príkladov, čiže správne identifikovaných ako pozitívne)	FP (počet „False Positive“ príkladov, čiže nesprávne identifikovaných ako pozitívne)
Klasifikátor tvrdí že to nie je daná trieda (negatívna identif.)	FN (počet „False Negative“ príkladov, čiže nesprávne označených ako negatívne)	TN (počet „True Negative“ príkladov, čiže správne identifikovaných ako negatívne)

Skutočná klasifikácia -> ----- Predikcia klasifikátora	Skutočná trieda príkladu emailu = JE SPAM	Skutočná trieda príkladu emailu = NIE JE SPAM
Podľa klasifikátora email: JE SPAM	40	5
Podľa klasifikátora email: NIE JE SPAM	10	45

- Mierky = pomer vybraných (súčtov) prvkov, napr.
 - **accuracy** (úspešnosť klasif.) $(TP + TN) / \text{súčet všetkých}$
 - chyba klasifikácie $(FN + FP) / \text{súčet všetkých}$
 - **precision** (presnosť) $TP / (TP + FP)$
 - **recall** (návratnosť) $TP / (TP + FN)$
 - **F1 skóre** $2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$
 - ...

https://en.wikipedia.org/wiki/Confusion_matrix

Mikro/makro spriemerovanie

- Ak máme viac triednu klasifikáciu (viac tried c_j), často sa používa spriemerovanie viacerých kontingenčných tabuľiek pre vyhodnotenie klasifikácie
- 1. Vytvoríme kontingenčnú tabuľku klasifikácie (confusion matrix) pre každú triedu
 - Ak priemerujeme cez viac tried (napr. n tried), tak máme preto n takých tabuľiek a teda hodnoty TP_i , TN_i , ... pre všetky $i=1,\dots,n$
- 2. Následne spriemerníme hodnoty, pričom existuje
 - Mikrospriemerovanie – spočítame hodnoty v tabuľkách dokopy do jednej spoločnej tauľky a potom použijeme vzorec pre zvolenú mierku (napr. accuracy)
 - Makrospriemerovanie – najprv spočítame zvolenú mierku pre každú tabuľku triedy zvlášť a potom urobíme ich priemer
- Mikrospriemerovanie je viac ovplyvnené triedami s veľkým počtom príkladov (ak sú rôzne početnosti tried, preváži hodnotenie klasifikácie tried s veľkým počtom príkladov)
- Makrospriemerovanie lepšie odráža priemernú presnosť jednotlivých podmnožín bez ohľadu na početnosti ich príkladov (prejavia sa tak aj hodnotenia menej príkladovo početných tried)

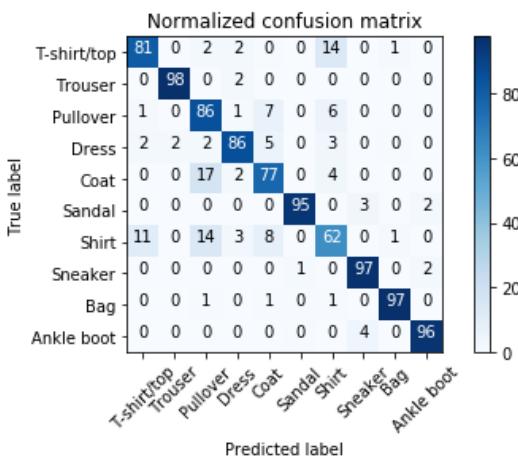
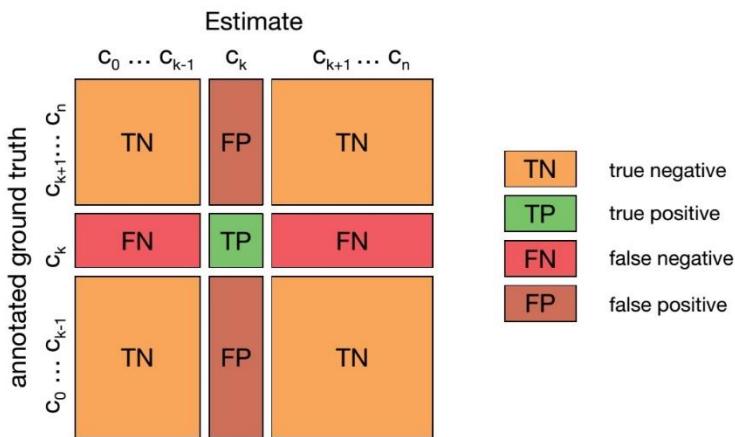
Hodnotenie v podobe jednej tabuľky (confusion matrix) pri viac tried. klasifikácii

		Skutočná		
Predikovaná	Trieda	A	B	C
	A	2	2	0
	B	1	2	0
	C	0	0	3

Presnosť A =
 $2/(2+2+0)$

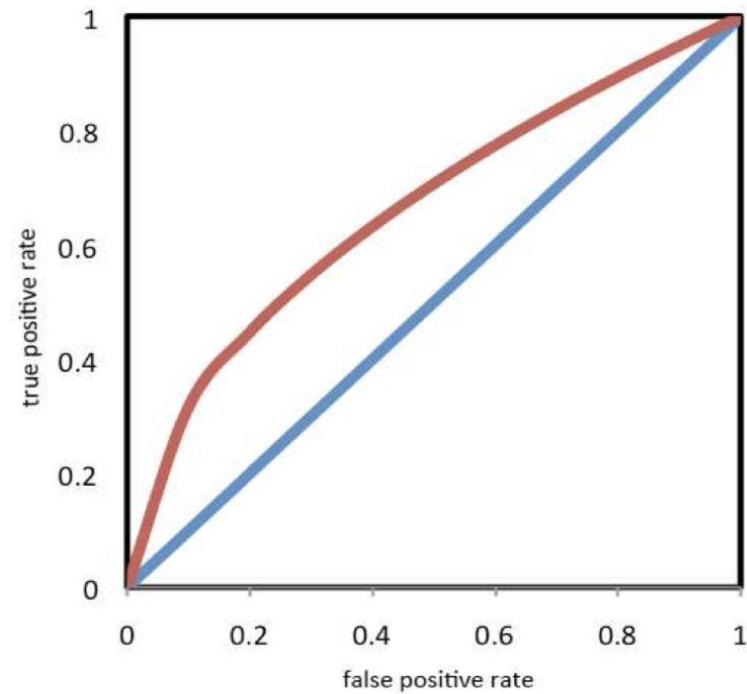
		Skutočná		
Predikovaná	Trieda	A	B	C
	A	2	2	0
	B	1	2	0
	C	0	0	3

Návratnosť A =
 $2/(2+1+0)$



ROC krivka

- ROC (Receiver Operator Characteristics)
- Pre binárnu klasifikáciu (pre klasifikáciu do viacerých tried - ROC krivky pre jednotlivé triedy)
- Závislosť skutočne pozitívnych prípadov (true positive) a falošne pozitívnych (false positive)
- AUC (Area Under Curve)
 - plocha pod krivkou (0.5 = náhoda, 1 = max)



Zlepšenie klasifikácie skladaním modelov

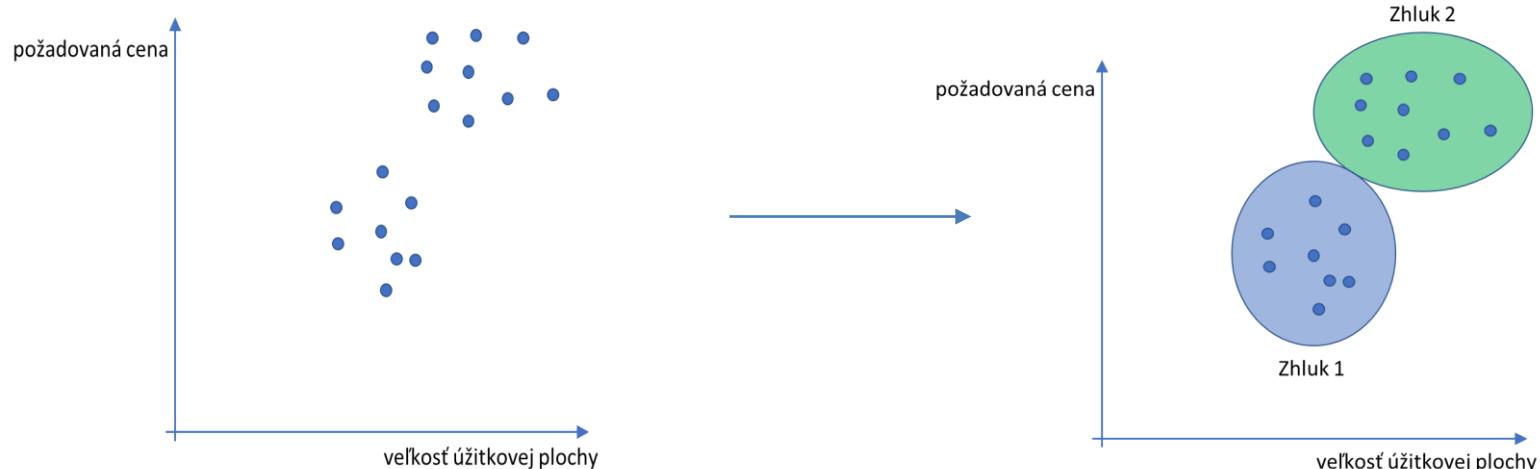
- Vylepšenie klasifikácie je možné dosiahnúť skladaním viacerých rôznych klasifikátorov
 - V R napr. **adabag** balík, v Python napr. scikit-learn metódy v **sklearn.ensemble**
- Výsledkom je jeden zložený klasifikátor K^* pozostávajúci z viacerých klasifikátorov K_1, K_2, \dots, K_n
 - **Bagging** – podľa zvolenej stratégie v rôznych krokoch i vyberieme podmnožinu O_i a ku každej vytvoríme nejaký klasifikátor $K_i \Rightarrow$ výsledná klasifikácia objektu = klasifikácia $K^* = \text{„hlasovanie“}$ klasifikátorov K_1 až K_n
 - **Boosting** – každému príkladu dáme váhu, následne trénujeme postupnosť klasifikátorov K_1, K_2, \dots , pričom pre chybne klasifikované príklady zvýšime v každom kroku váhu (aby sme sa v ďalšom kroku na nich viac sústredili) \Rightarrow výsledok K^* je opäť hlasovanie, pričom váha hlasu klasifikátora odpovedá jeho presnosti (chybovosti)

Zlepšenie klasifikácie rozhodovacích stromov

- Vylepšenie klasifikácie rozhodovacích stromov
= orezávanie RS
 - Pre-prunning – počas procesu tvorby RS – ak nejaká štatistika (napr. informačný zisk) pre zvolený atribút nepresiahne zvolený prah => zastavíme ďalšie vetvenie
 - Post-prunning – po vygenerovaní RS – dodatočne odstránime vetvy porovnaním medzi očakávanou chybou pre daný uzol a jeho jednoduchou náhradou listovým uzlom => ak sa chyba zmenší => orezanie

Zhlukovanie

- Nekontrolované učenie – vstupné dátá nemajú informáciu o rozdelení do tried, algoritmus sa v podstate snaží navrhnuť rozdelenie do „tried“ = zhlukov



Počet izieb	Úžitková plocha	Požadovaná cena	Typ
2	64	70 000	Byt
3	82	142 000	Rad.dom
2	36	64 000	Byt
4	96	190 000	Dom
...			

Algoritmus zhlukovania

Zhluk
zhluk1
zhluk2
zhluk1
zhluk2
...

Zhlukovanie (2)

- Cieľ: Roztriedenie objektov do vopred nestanovených zhlukov („cluster“) podľa kritéria (ne)podobnosti, pričom
 - Rozdiely medzi objektmi jedného zhluku by mali byť čo najmenšie (vnútorná kompaktnosť zhlukov)
 - Rozdiely medzi objektmi z rôznych zhlukov by mali byť čo najväčšie (jasná odlišnosť zhlukov medzi sebou)
- Zhluky nie sú známe vopred (ich počet môže a nemusí byť daný vopred) => nemajú dopredu daný význam (ani ho nemusíme nájsť)
 - Typické hodnoty zhluku = často tzv. stred zhluku (reprezentant)
- Problém: Ako definovať rozdiel / podobnosť objektov = metriku – objekty sú definované často ako vektory
 - Euklidovská vzdialenosť = príklad **nepodobnosti** (čím je číslo väčšie, tým sú si objekty menej podobné)
 - Kosínusová metrika = príklad funkcie **podobnosti** (čím je číslo väčšie, tým sú si objekty podobnejšie) = normovaný skalárny súčin vektorov = kosínus uhla vektorov ($\cos 0^\circ = 1$, $\cos 90^\circ = 0$)

Vyhodnotenie zhlukovanie

- Keďže zhlukovanie je úlohou nekontrolovaného typu – väčšinou nie sú k dispozícii externé validačné kritériá => na reálnych dátach je preto zložité vykonať vhodnú validáciu zhlukov
- **Interné validačné kritériá** sa používajú, ak nie sú k dispozícii žiadne externé kritériá na vyhodnotenie kvality zhlukov
 - Väčšinou ide o kriteriálnej funkciu, ktorá je zhlukovaním optimalizovaná (vedie to k tomu, že dané kritérium favorizuje taký algoritmus, ktorý práve toto kritérium využíva na optimalizáciu)
 - Suma štvorcov vzdialenosí od centroidov – určia sa reprezentanti zhluku (centroidy) a potom sa spočíta súčet kvadrátov vzdialenosí ostatných bodov od centroidu
 - Vzdialenosť dvoch zhlukov (vzdialenosť dvoch najbližších susedov z dvoch rôznych zhlukov, vzdialenosť dvoch najvzdialenejších susedov z dvoch rôznych zhlukov, priemer vzdialenosí medzi všetkými objektami z dvoch zhlukov)
 - Pomer vnútrozhlukovej a medzizhlukovej vzdialenosťi – vyberieme množinu párov bodov, kde časť patrí do jedného zhluku a časť do ostatných, spočítame priemerné vzdialenosí v rámci jedného zhluku a potom vzdialenosť s bodmi z ostatných zhlukov
 - Problém je že nehovoria nič o interpretácii zhlukov

Vyhodnotenie popisom zhlukov

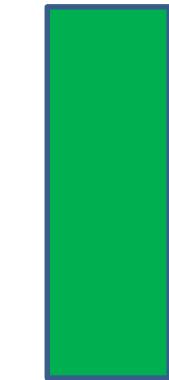
- Pri menšom počte atribútov môžeme zhluky popísať napr. základnými štatistikami a zobraziť ich graficky, určiť priemerné hodnoty atribútov, atď.



priemerná
veľkosť
úžitkovej plochy

priemerná
požadovaná
cena

Zhluk 1



priemerná
veľkosť
úžitkovej plochy

priemerná
požadovaná
cena

Zhluk 2

- Pri zložitejších vzťahoch s viacerými atribútmi môžeme použiť metódy, ktoré nám priamo vygenerujú pre zhluky pravidlá so zložitejšími podmienkami, napr.: **ak** požadovaná cena $< 70\ 000$ a počet izieb > 3 a typ = byt **potom** zhluk 1

Vyhodnotenie zhlukovania – externé kritéria

- Možné použiť, ak je k dispozícii informácia o skutočnom zatriedení do zhlukov
- V prípade reálnych dát, veľmi často nemožné – cieľ sa dá splniť približne, ak sú k dispozícii označenia tried
- Ak je možné – preferuje sa validáciu externými kritériami oproti interným
- Potom sa dá spočítať napr. confusion matrix (počty prvkov priradených do skutočných zhlukov oproti algoritmom vytvorených)
- Ostatné externé kritériá:
 - čistota zhlukov (purity) – kvalitne určený zhluk by mal obsahovať iba príklady skutočne patriace do daného jedného zhluku

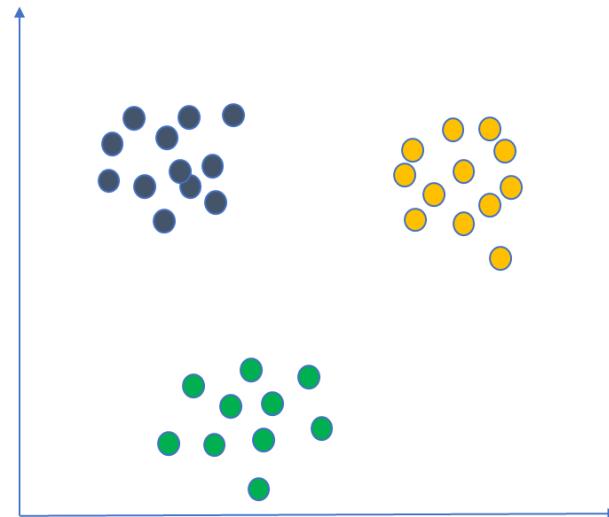
Metódy zhlukovania

- Existujú rôzne typy metód, rozdelenie môže byť napríklad nasledujúce:
 - Rozdeľujúce (centroidné) metódy
 - Hierarchické metódy – vytváranie viacúrovňovej hierarchie zhlukov
 - Metódy založené na hustote a mriežke
 - Metódy založené na modeloch – napr. SOM (Self-Organizing Maps)
- Rozdeľujúce metódy
 - Vytvárajú a udržiavajú model v podobe centroidov (bod v priestore atribútov) reprezentujúcich zhluky => väčšinou každý objekt patrí do jedného z nich (sú teda často disjunktné)
 - Najstaršie prístupy: klasický iteratívny algoritmus
 - k-means (k-stredov), k-medians, k-medoids
- Hierarchické zhlukovanie – dva prístupy
 - Aglomeratívne – začíname jednotlivými objektmi (samostatné zhluky), ktoré sa spájajú na základe kritéria (v hierarchii) až po jeden konečný zhluk
 - Divízne – opačný proces, zhluk všetkých objektov postupne delíme až po najnižšiu úroveň (ked' každý zhluk je objekt)

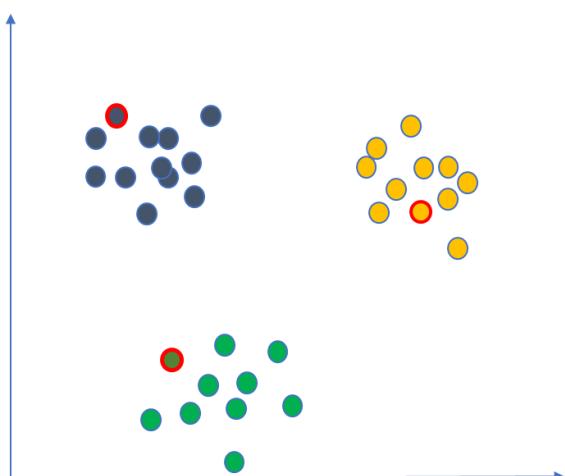
k-Means algoritmus

- Zhluk = centroid ... špeciálny bod okolo ktorého sa koncentrujú príklady zhluku
- Objekty $X = \{x_1, \dots, x_n\}$, k zhlukov $Y = \{y_1, \dots, y_k\}$
- Výpočet polohy centroidu: $\longrightarrow y_j = \frac{\sum_{x_i \in Y_j} x_i}{|Y_j|}$
- Chybová funkcia = sumár odchýlok objektov od stredov ich zhlukov
- Základný algoritmus:
 - 1. Inicializácia zhlukov – k náhodne vybraných centier
 - 2. Priradenie objektov najbližšiemu zhluku (v zmysle minimalizácie vzdialenosť alebo maximalizácie podobnosti)
 - 3. Výpočet nových centier zhlukov
 - 4. Ukončovacia podmienka (ak nesplnená => návrat na 2):
 - Bol dosiahnutý daný počet iterácií
 - Chybová funkcia je menšia ako zvolený prah

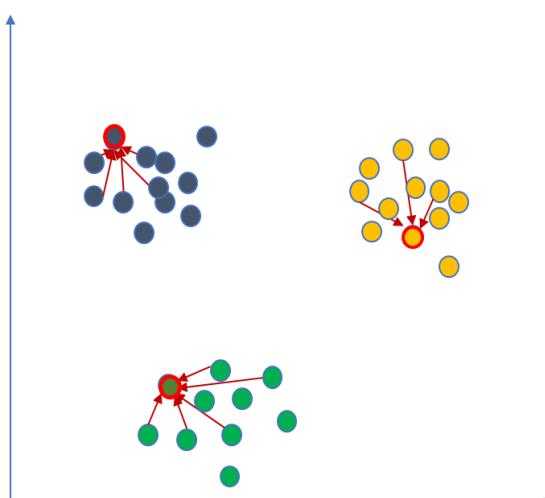
Kmeans – priebeh učenia



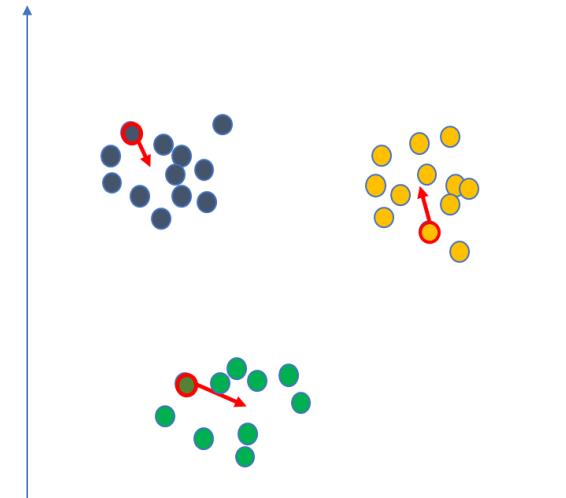
- 1. Inicializácia K centier zhlukov (centroidov)
- 2. Priradenie každej inštancie (vektora) k najbližšiemu centroidu
- 3. Prepočítanie centroidov ako priemerných hodnôt inštancií v rámci zhluku



1. Inicializácia



2. Priradenie vektorov



3. Prepočet stredov

K-Means – vlastnosti a rozšírenia

- Výhody k-Means:
 - Jednoduchá a často používaná zhlukovacia technika, ľahko pracuje s rôznymi typmi metrík
 - Nezávisí na zoradení príkladov
 - Umožňuje priamočiaru paralelizáciu
- Nevýhody k-Means
 - Riziko padnutia do lokálneho minima (náhodná inicializácia centier)
 - Nutnosť nastaviť hodnoty k
 - Citlivosť na zmeny súradníc (závisí na type použitej metriky), odchýlené body (outliers) a nevyváženosť zhlukov
- Príklady rozšírení
 - k-Medians – ak použijeme medián dátových bodov v rámci príkladov z jedného zhluku
 - algoritmus vyberá príklady trocha robustnejšie ako k-Means, nakoľko medián nie je tak citlivý na outliersy (výrazne odchýlené hodnoty) ako priemer
 - k-Medoids – v prípade k-Medoids sú reprezentanti vždy konkrétny objekt z dátovej množiny
 - Vhodné vtedy, ak chceme aby bol reprezentant zhluku konkrétny objekt – napr. segmentácia zákazníkov (typický zákazník danej skupiny zákazníkov)
 - Bisecting k-means = hierarchické (divízne) rozšírenie k-means
- V R napr. pre k-means zhlukovanie – `kmeans()`, k-medoids – `pam()`, `clara()` (**cluster** balík) a `pamk()` (**fpc** balík) ... stabilnejšie a robustnejšie
- V Python, napr. `scikitslearn.cluster.KMeans`

Príklad – k-means v R

```
set.seed(8953)  
iris2 <- iris  
iris2$Species <- NULL  
(kmeans.result <- kmeans(iris2, 3))  
table(iris$Species, kmeans.result$cluster) # porovnanie voci povodnym Species
```



	1	2	3
setosa	0	0	50
versicolor	48	2	0
virginica	14	36	0

K-means clustering with 3 clusters of sizes 62, 38, 50

cluster means:

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1      5.901613     2.748387     4.393548    1.433871
2      6.850000     3.073684     5.742105    2.071053
3      5.006000     3.428000     1.462000    0.246000
```

clustering vector;

within cluster sum of squares by cluster:

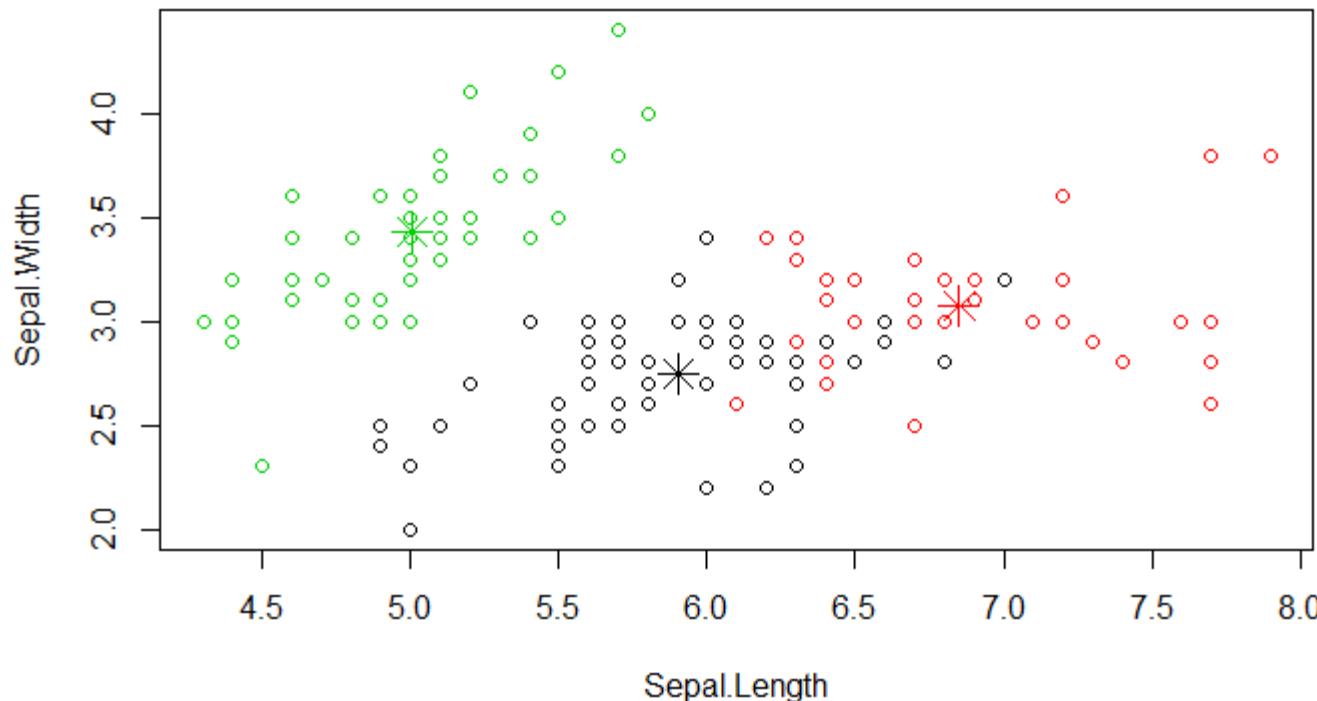
```
[1] 39.82097 23.87947 15.15100  
(between_SS / total_SS =  88.4 %)
```

Available components:

```
[1] "cluster"      "centers"       "totss"        "withinss"      "tot.withinss" "betweenss"     "size"         "iter"  
[9] "ifault"
```

Príklad – graf znázornených bodov podľa klastrov (pre vybrané atribúty)

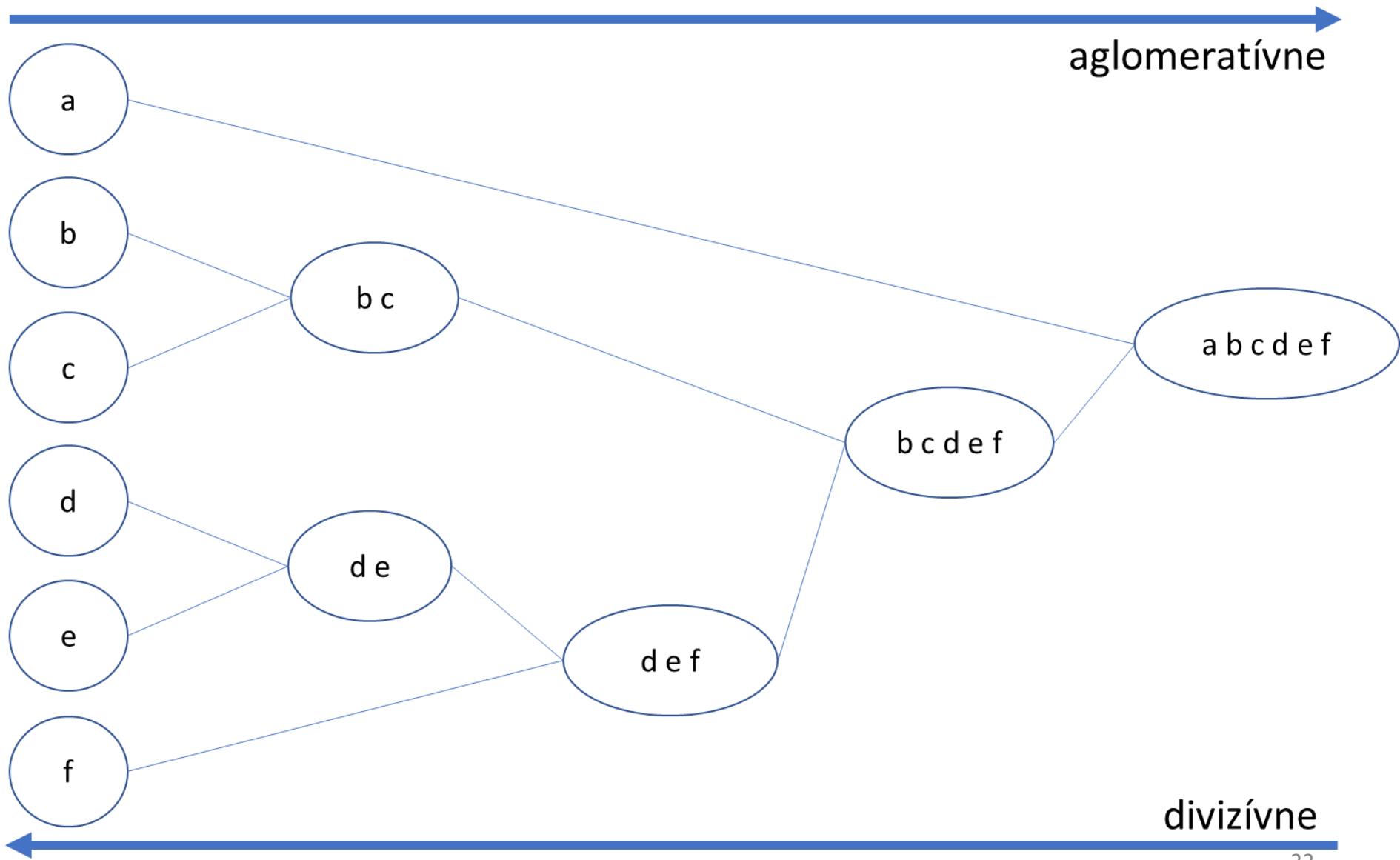
```
plot(iris2[c("Sepal.Length", "Sepal.Width")], col = kmeans.result$cluster)
points(kmeans.result$centers[, c("Sepal.Length", "Sepal.Width")], col = 1:3,
       pch =
```



Hierarchické zhlukovanie

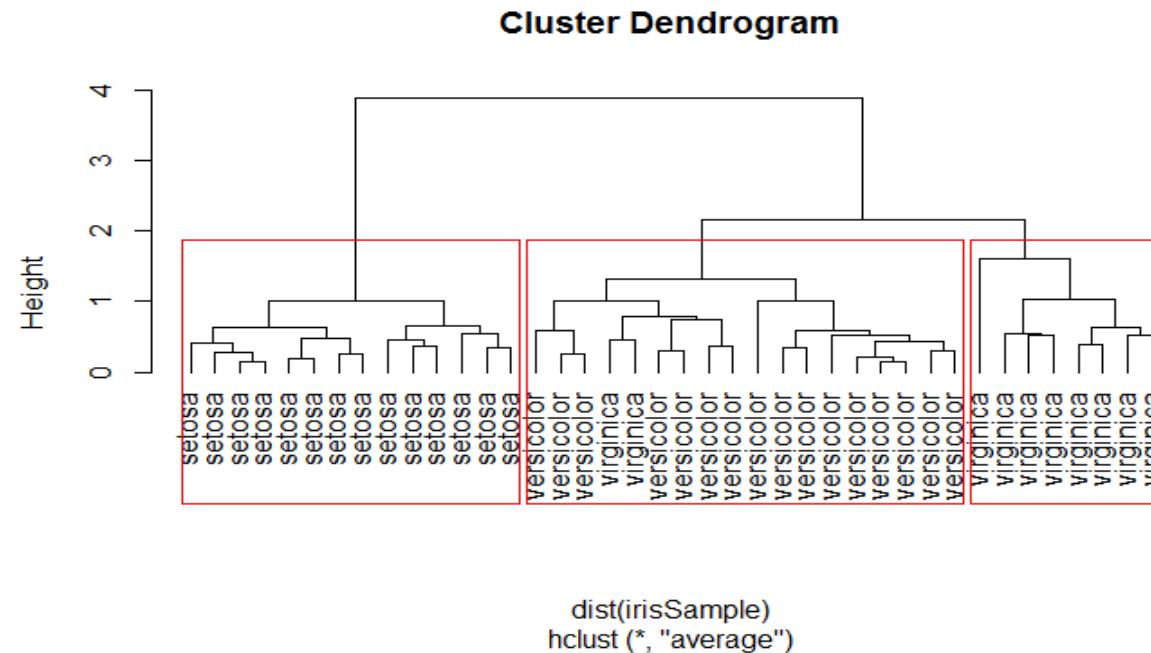
- Hierarchické metódy – aglomeratívne vs. Divízne
 - Vzniká hierarchická štruktúra zhlukov (strom zhlukov)
 - Proces postupného spájania / delenia zobrazuje tzv. dendrogram
 - Možnosť skoršieho ukončenia procesu spájania / delenia => splnenie podmienky (napr. počet zhlukov)
- Algoritmus aglomeratívneho zhlukovania
 - 1. Vyskúšajú sa všetky páry aktuálnych zhlukov (na začiatku sú zhluky samotné príklady), s najpodobnejších dvojíc sa vytvoria nové zhluky (disjunktné)
 - 2. Definujú sa atribúty zhluku (napr. priemer)
 - 3. Ak máme už len jeden zhluk alebo splnená iná podmienka = koniec, inak späť na 1 z aktuálnej zostavou zhlukov
- Divízne je obdobou algoritmu, začiatok je jeden zhluk všetkých príkladov a hľadáme rozklad aktuálnych zhlukov na menšie v každom kroku (koniec na jednotlivých)
- Hierarchické zhlukovanie v R – `hclust` funkcia (priamo v R), prípadne funkcia `pvclust` (balík `pvclust`)
- Python – napr. metódy v rámci `sklearn.cluster`

Hierarchia zhlukov - dendrogram



Príklad – hclust na iris + dendrogram

```
set.seed(2835); idx <- sample(1:dim(iris)[1], 40) # vzorka 40 prikladov z IRIS  
irisSample <- iris[idx, ]; irisSample$Species <- NULL # sample tabulka a  
vynulovanie Species  
hc <- hclust(dist(irisSample), method = "ave") # hierarchicke zhluovanie  
plot(hc, hang = -1, labels = iris$Species[idx]) # vykreslenie klastrov  
rect.hclust(hc, k = 3) # orezanie dendrogramu na úroveň 3 zhluov  
groups <- cutree(hc, k = 3) # získanie ID zhluov
```



Asociačné pravidlá

- Asociačné pravidlá popisujú asociácie alebo korelácie medzi množinami položiek
 - Pravidlo typu $X \Rightarrow Y$, kde X, Y sú disjunktné podmnožiny atribútov
 - Aplikácia: tzv. Analýza nákupného košíka pre podporu rozhodovania v obchode (krížový marketing, návrh katalógov, rozvrhovanie položiek, ...)
- Základné vlastnosti pravidla sú:
 - Support (Podpora) $P(X \cup Y)$
 - Confidence (Spoľahlivosť) $P(Y|X) = P(X \cup Y)/P(X)$
 - Lift ... $\text{confidence}(X \Rightarrow Y)/P(Y) = P(X \cup Y)/(P(X)*P(Y))$
- Algoritmy pre hľadanie asociačných pravidiel = hľadanie frekventovaných množín položiek (frequent itemsets) = množiny položiek s *minsup* podporou
 - APRIORI – iteratívny, po úrovniach a do šírky prehľadávací algoritmus založený na početnostiach transakcií s krokmi (1.) hľadania frekventovaných položkových množín dĺžky $1, \dots, k$, a následného (2.) generovania pravidiel s potrebnou podporou a spoľahlivosťou – v R napr. **apriori()** funkcia (balík **arules**), v Python - napr. **mlxtend** balík
 - ECLAT – hľadá frekventované položkové množiny ekvivalentných tried prehľadávaním do hĺbky a výpočtom prienikov množín (namiesto počítania transakcií) – funkcia **eclat()**

Asociačné pravidlá – príklad

ID transakcie	kúpené položky
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. podpora 50%
Min. spoľahlivosť 50%

frekventované množiny položiek	podpora
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

Pre pravidlo $A \Rightarrow C$:

$$\text{podpora} = \text{podpora}(\{A, C\}) = 2 / 4 = 50\%$$

$$\text{spoľahlivosť} = \text{podpora}(\{A, C\}) / \text{podpora}(\{A\}) = 2 / 3 = 66.6\%$$

Pre pravidlo $C \Rightarrow A$:

$$\text{podpora} = \text{podpora}(\{C, A\}) = 2 / 4 = 50\%$$

$$\text{spoľahlivosť} = \text{podpora}(\{C, A\}) / \text{podpora}(\{C\}) = 2 / 2 = 100\%$$

Existujú aj rozšírenia ako **kvantitatívne asociačné pravidlá** (pre zložitejšie atribúty) a **hierarchické asociačné pravidlá** (pre atribúty definované hierarchiou konceptov)

Príklad – Titanic dáta

- V datasets v R – Titanic – 4rozmerná tabuľka o pasažieroch Titaniku (2201)
 - Atribúty: Class (1st, 2nd, 3rd, Crew), Sex(Male,Female), Age(Child, Adult), Survived(No,Yes)
 - Použijeme „raw“ verziu dát (nie sumárnu) ... (<https://peter.butka.website.tuke.sk/res/titanic.csv>)
 - Príklad z dát (sample 5 vzoriek)

```
idx <- sample(1:nrow(titanic), 5); titanic[idx, ]
```

	Class	Sex	Age	Survived	> summary(titanic)	Class	Sex	Age	Survived
1168	Crew	Male	Adult	No		1st :325	Female: 470	Adult:2092	No :1490
923	Crew	Male	Adult	No		2nd :285	Male :1731	Child: 109	Yes: 711
950	Crew	Male	Adult	No		3rd :706			
627	3rd	Male	Adult	No		Crew:885			
21	3rd	Male	Child	No					

Príklad – AR z Titanic dát

- Apriori algoritmus v R – default nastavenia: supp=0.1; conf=0.8; maxlen=10 (max dĺžka AR)

```
library(arules)
```

```
rules.all <- apriori(titanic) # vyprodukuje info o výpočte  
inspect(rules.all) # vypíše pravidlá
```

lhs	rhs	support	confidence	lift
1 {}	=> {Age=Adult}	0.9504771	0.9504771	1.0000000
2 {class=2nd}	=> {Age=Adult}	0.1185825	0.9157895	0.9635051
3 {class=1st}	=> {Age=Adult}	0.1449341	0.9815385	1.0326798
4 {Sex=Female}	=> {Age=Adult}	0.1930940	0.9042553	0.9513700
5 {class=3rd}	=> {Age=Adult}	0.2848705	0.8881020	0.9343750
6 {Survived=Yes}	=> {Age=Adult}	0.2971377	0.9198312	0.9677574
7 {class=Crew}	=> {Sex=Male}	0.3916402	0.9740113	1.2384742
8 {class=Crew}	=> {Age=Adult}	0.4020900	1.0000000	1.0521033
9 {Survived=No}	=> {Sex=Male}	0.6197183	0.9154362	1.1639949
10 {Survived=No}	=> {Age=Adult}	0.6533394	0.9651007	1.0153856
11 {Sex=Male}	=> {Age=Adult}	0.7573830	0.9630272	1.0132040
12 {Sex=Female, Survived=Yes}	=> {Age=Adult}	0.1435711	0.9186047	0.9664669
13 {class=3rd, Sex=Male}	=> {Survived=No}	0.1917310	0.8274510	1.2222950
14 {class=3rd, Survived=No}	=> {Age=Adult}	0.2162653	0.9015152	0.9484870
15 {class=3rd, Sex=Male}	=> {Age=Adult}	0.2099046	0.9058824	0.9530818
16 {Sex=Male, Survived=Yes}	=> {Age=Adult}	0.1535666	0.9209809	0.9689670
17 {class=Crew, Survived=No}	=> {Sex=Male}	0.3044071	0.9955423	1.2658514
18 {class=Crew, Survived=No}	=> {Age=Adult}	0.3057701	1.0000000	1.0521033
19 {class=Crew, Sex=Male}	=> {Age=Adult}	0.3916402	1.0000000	1.0521033
20 {class=Crew, Age=Adult}	=> {Sex=Male}	0.3916402	0.9740113	1.2384742
21 {Sex=Male, Survived=No}	=> {Age=Adult}	0.6038164	0.9743402	1.0251065
22 {Age=Adult, Survived=No}	=> {Sex=Male}	0.6038164	0.9242003	1.1751385
23 {class=3rd, Sex=Male, Survived=No}	=> {Age=Adult}	0.1758292	0.9170616	0.9648435
24 {class=3rd, Age=Adult, Survived=No}	=> {Sex=Male}	0.1758292	0.8130252	1.0337773
25 {class=3rd, Sex=Male, Age=Adult}	=> {Survived=No}	0.1758292	0.8376623	1.2373791
26 {class=Crew, Sex=Male, Survived=No}	=> {Age=Adult}	0.3044071	1.0000000	1.0521033
27 {class=Crew, Age=Adult, Survived=No}	=> {Sex=Male}	0.3044071	0.9955423	1.2658514

Príklad – AR z Titanic dát (2)

```
# LEN PRAVIDLA MAJUCE V PRAVEJ CASTI SURVIVED + ine parametre  
rules <- apriori(titanic, control = list(verbose=F), parameter = list(minlen=2,  
                     supp=0.005, conf=0.8), appearance =  
                     list(rhs=c("Survived=No","Survived=Yes"), default="lhs"))  
quality(rules) <- round(quality(rules), digits=3) # hodnotenia na 3 desat. Miesta  
rules.sorted <- sort(rules, by="lift") # usporiadanie pravidiel podla lift  
inspect(rules.sorted)
```

	lhs	rhs	support	confidence	lift
1	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.011	1.000	3.096
7	{Class=2nd, Sex=Female, Age=Child}	=> {Survived=Yes}	0.006	1.000	3.096
4	{Class=1st, Sex=Female}	=> {Survived=Yes}	0.064	0.972	3.010
10	{Class=1st, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.064	0.972	3.010
2	{Class=2nd, Sex=Female}	=> {Survived=Yes}	0.042	0.877	2.716
5	{Class=Crew, Sex=Female}	=> {Survived=Yes}	0.009	0.870	2.692
11	{Class=Crew, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.009	0.870	2.692
8	{Class=2nd, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.036	0.860	2.663
9	{Class=2nd, Sex=Male, Age=Adult}	=> {Survived=No}	0.070	0.917	1.354
3	{Class=2nd, Sex=Male}	=> {Survived=No}	0.070	0.860	1.271
12	{Class=3rd, Sex=Male, Age=Adult}	=> {Survived=No}	0.176	0.838	1.237
6	{Class=3rd, Sex=Male}	=> {Survived=No}	0.192	0.827	1.222

Prvé dve pravidlá sú redundantné – pretože prvé hovorí že každé dieťa z druhej triedy prežilo, zároveň muselo prežiť aj každé dievča z druhej triedy (druhé pravidlo). Ak pravidlo (ako to druhé v zozname) je rozšírením (doplnením) pravidla iného (prvého v zozname), pričom lift rozšíreného je rovnaký alebo menší ako jednoduchšieho, je rozšírené pravidlo redundantné.

V zozname sú aj ďalšie redundantné pravidlá – v poradí 4,7,8 riadok (! Nie číslo v tabuľke pred pravidlom !).

Odstránenie redundantných pravidiel

- Môžeme nájsť redundantné pravidlá a odstrániť ich (prunning), napr.
`is.redundant(rules.sorted)` # T/F ak je/nie je pravidlo redundantné
výsledok: [1] FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE FALSE FALSE
výpis všetkých redundantných pravidiel
`inspect(rules.sorted[is.redundant(rules.sorted)])`

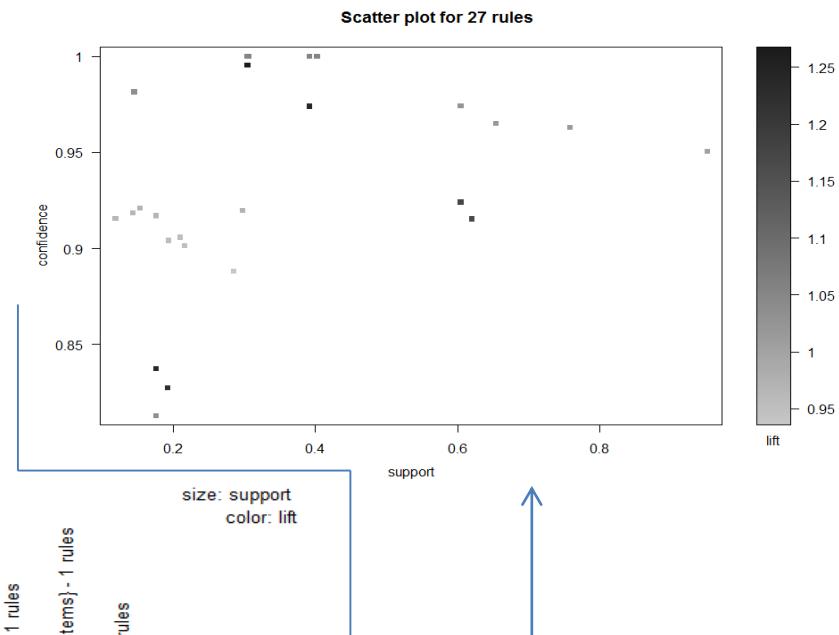
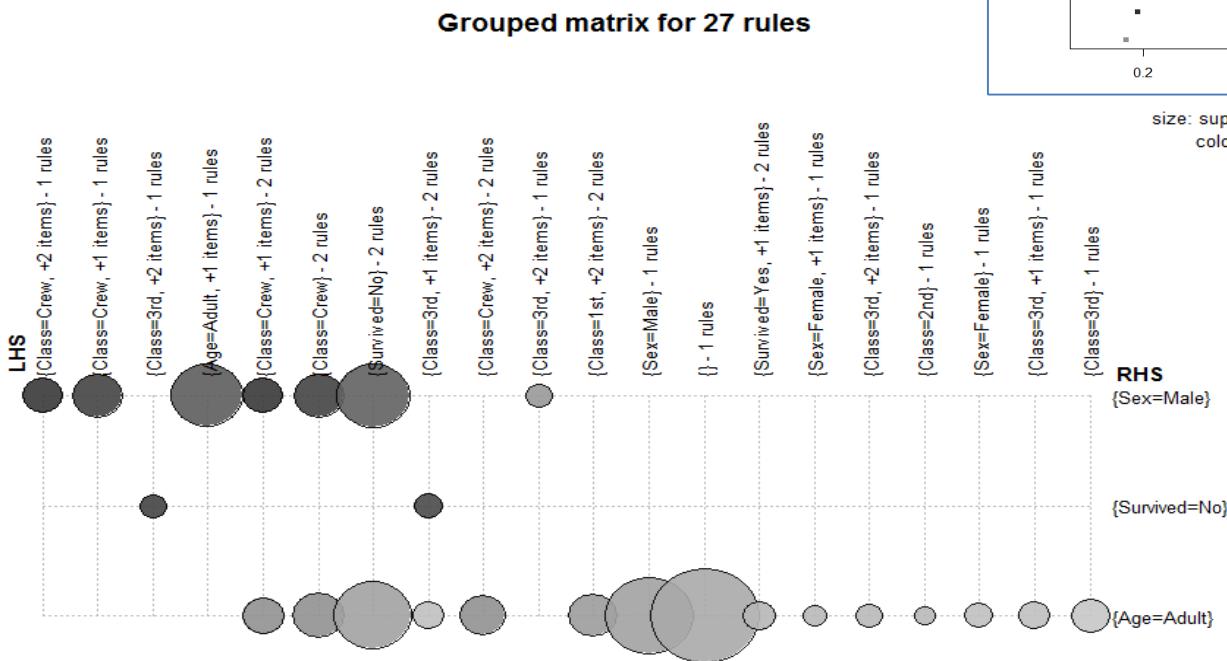
```
lhs                                rhs          support confidence lift
[1] {Class=2nd,Sex=Female,Age=Child} => {Survived=Yes} 0.006    1.000    3.096
[2] {Class=1st,Sex=Female,Age=Adult}  => {Survived=Yes} 0.064    0.972    3.010
[3] {Class=Crew,Sex=Female,Age=Adult} => {Survived=Yes} 0.009    0.870    2.692
[4] {Class=2nd,Sex=Female,Age=Adult}  => {Survived=Yes} 0.036    0.860    2.663
```

```
# výpis neredundantných pravidiel
inspect(rules.sorted[!is.redundant(rules.sorted)])
```

```
lhs                                rhs          support confidence lift
[1] {Class=2nd,Age=Child}           => {Survived=Yes} 0.011    1.000    3.096
[2] {Class=1st,Sex=Female}          => {Survived=Yes} 0.064    0.972    3.010
[3] {Class=2nd,Sex=Female}          => {Survived=Yes} 0.042    0.877    2.716
[4] {Class=Crew,Sex=Female}         => {Survived=Yes} 0.009    0.870    2.692
[5] {Class=2nd,Sex=Male,Age=Adult}  => {Survived=No}   0.070    0.917    1.354
[6] {Class=2nd,Sex=Male}            => {Survived=No}   0.070    0.860    1.271
[7] {Class=3rd,Sex=Male,Age=Adult}  => {Survived=No}   0.176    0.838    1.237
[8] {Class=3rd,Sex=Male}            => {Survived=No}   0.192    0.827    1.222
```

Vizualizácia asociačných pravidiel

- Balík **arulesViz**
- Bodový graf (cez plot)

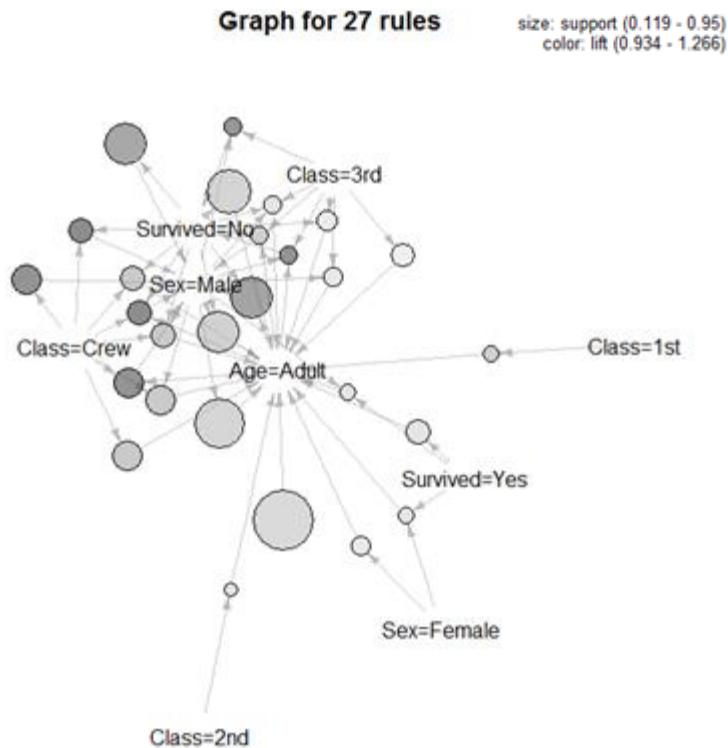


`plot(rules.all)`

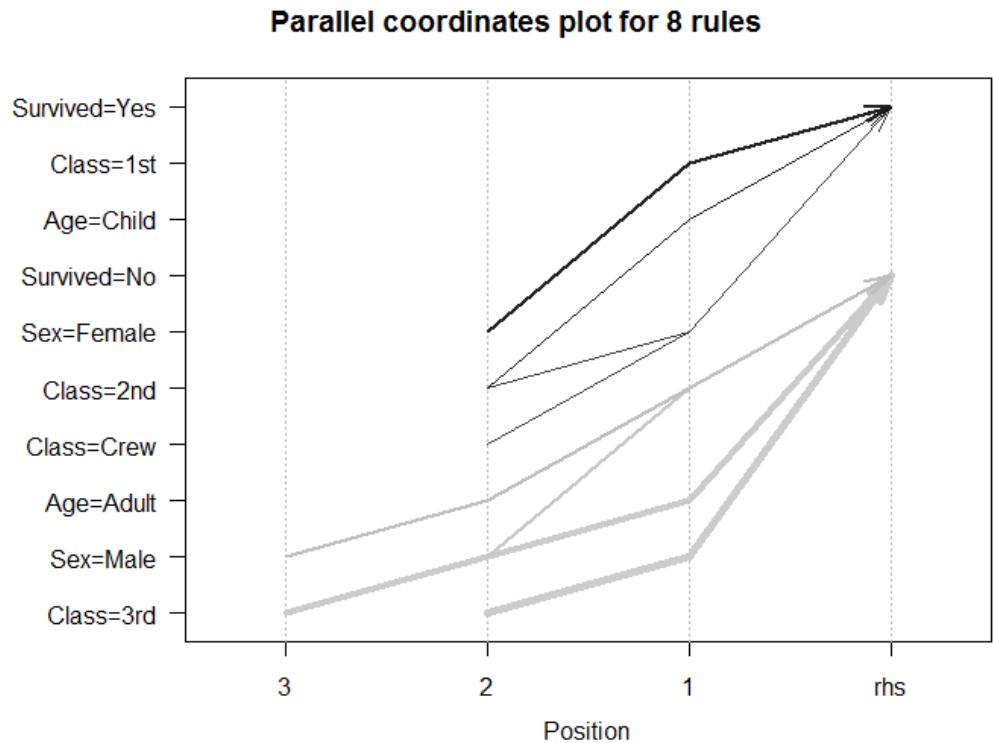
`plot(rules.all, method = "grouped")`

Vizualizácia asociačných pravidiel (2)

- Existujú aj ďalšie metódy pre zobrazovanie
 - graph, paracord, ...



```
plot(rules.all, method = "graph", control = list(type = "items"))
```



```
plot(rules.pruned, method = "paracoord", control = list(reorder = TRUE))
```

Dátová analytika v medicínskej diagnostike

František Babič

frantisek.babic@tuke.sk

Osnova

Riešenie reálnej
úlohy pomocou
dátovej analytiky

Prečo dátová
analytika

Čo majú spoločné?

Objavovanie znalostí v databázach

Dolovanie znalostí

Dolovanie znalostí z dát

Dátová analytika

Analýza dát



ANALÝZA MEDICÍNSKÝCH DÁT => PODPORA DIAGNOSTIKY VYBRANÝCH OCHORENÍ

Motivácia – prečo ?

Objem
zhromažďovaných
dát sa neustále
zvyšuje.

Medicínska
diagnostika je
komplexný proces.

Zdravotný stav sa
vyvíja a mení v
čase.

Zdravotné faktory
sa navzájom
ovplyvňujú.

Doktori nie sú
stroje.

Doktori - očakávania

Jednoducho pochopiteľné výsledky.

Dôvody, prečo model odporúča práve tento výsledok.

Možnosť späťne zobraziť proces tvorby modelu = proces „rozhodovania“ algoritmu.

Analytici - očakávania

Dáta v dostatočnej kvalite a objeme.

Popis a vysvetlenie dát.

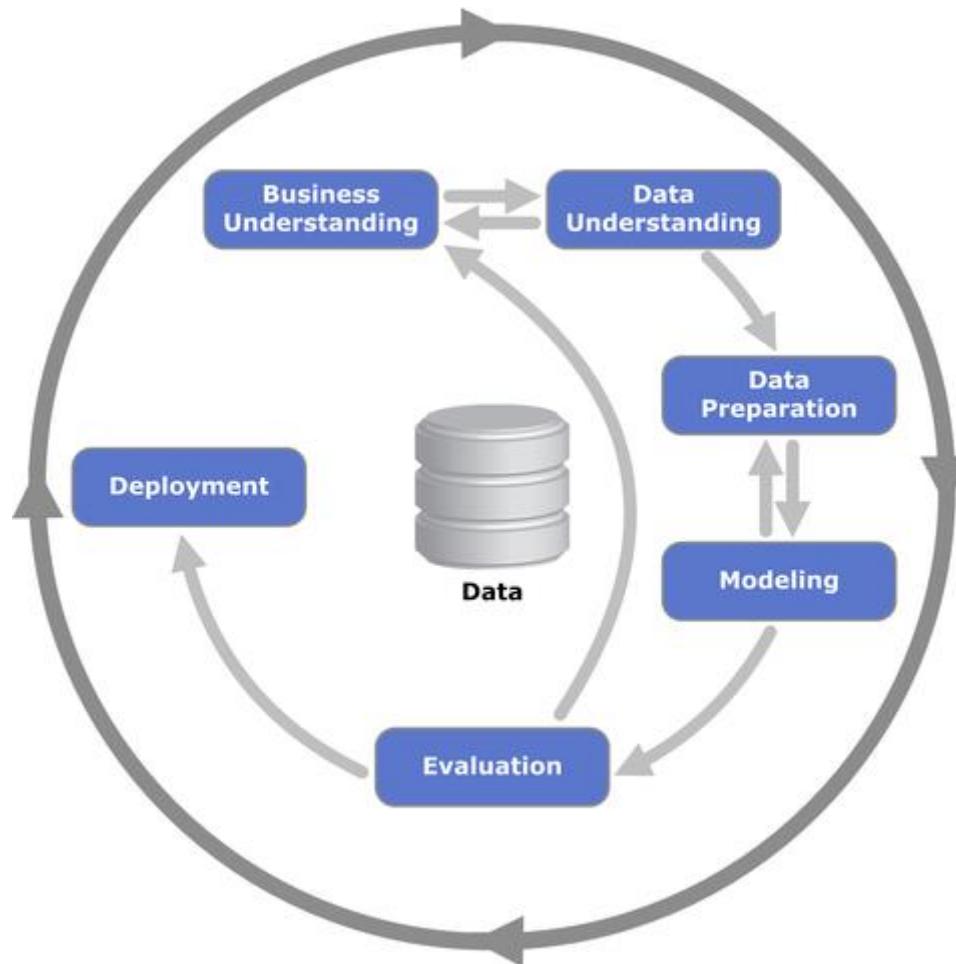
Experné znalosti.

Podpora pri vyhodnotení z pohľadu cieľového
používateľa.

Rada 1, ako analytický proces realizovať
efektívne a doviest' k očakávanému výsledku.

**KOMUNIKUJTE, KOMUNIKUJTE,
KOMUNIKUJTE!**

CRISP-DM metodológia

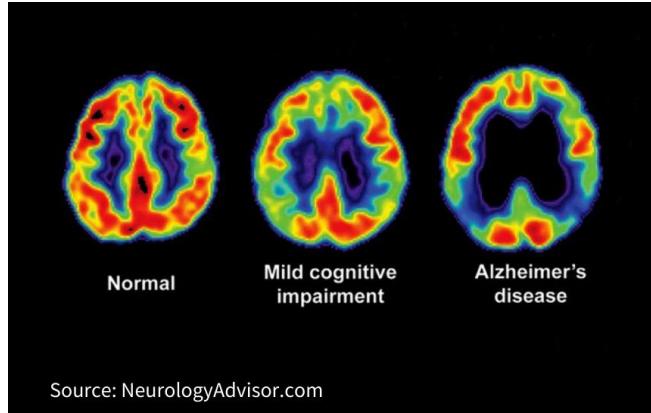


Pochopenie cieľa (1)

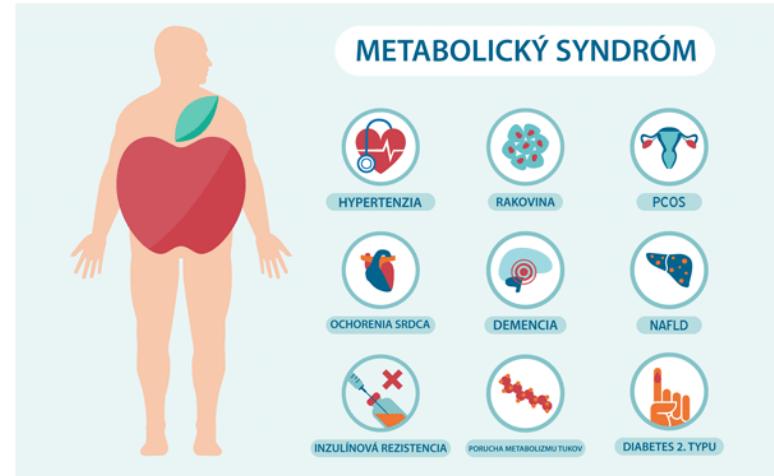
- Z pohľadu biznisu:
 - Poskytovať kvalitnejšiu zdravotnú starostlivosť.
 - Zvýšiť spokojnosť pacientov s poskytovanou starostlivosťou.
 - Znížiť výdavky na nesprávne určenú diagnózu alebo liečebný postup.
- Komunikácia a dohoda s medicínskym expertom na úlohách.

Skúmané ochorenia

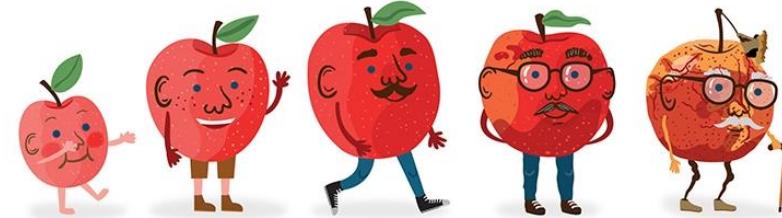
Mierne kognitívne zhoršenie



Syndróm krehkosti (únava, nevýkonnosť, chudnutie, svalová slabosť a pribúdanie funkčných deficitov)



Metabolický syndróm



Pochopenie cieľa (2)

- Z pohľadu dátovej analytiky:
 - Binárna klasifikácia (0 – zdravý človek, 1 – pozitívne diagnostikovaný pacient)
 - Exploratívna dátová analytika (čo najlepšie pochopenie pacientov na základe vstupných dát)
 - Zhlukovanie (identifikácia skupín pacientov s rovnakými charakteristikami)
 - Multinomiálna logistická regresia (porovnanie viacerých skupín pacientov, identifikácia rozdielov)

Pochopenie dát (1)

- Elektronické zdravotné záznamy = dôležitý zdroj informácií o zdravotnom stave pacientov, prekonaných ochoreniach, rodinnej anamnéze, atď.
- Rôzne vzorky z klinickej praxe v Chorvátsku.
(počet vstupných vzoriek je dôležitý!)
- Každý pacient je charakterizovaný atribútmi z krvných testov alebo zdravotnej karty.

Pochopenie dát (2)

- Dátové vzorky mali rôzny objem, od 93 pacientov cez 450 až k cca 1 000.
- Pacienti boli dôchodcovia, viac žien ako mužov.
- Každú dátovú vzorku sme sa snažili pochopiť pomocou tradičných popisných charakteristík a vhodných grafických metód.

Pochopenie dát (3)

Numerické atribúty:

- max, min, priemer, medián, boxplot, kvartily.
- korelačná analýza.
- test normality rozloženia dát
- parametrické a neparametrické štatistické metódy

Nominálne atribúty:

- početnosť hodnôt
- histogram
- Pearsonov Chi-kvadrát test nezávislosti
- Fisherov test

Štatistické metódy

Shapiro-Wilk test normality rozloženia dát

2-výberový Welchov t-test (parametrický test,
0/1 cieľový atribút)

Mann-Whitney-Wilcoxon test (neparametrický
test, 0/1 cieľový atribút)

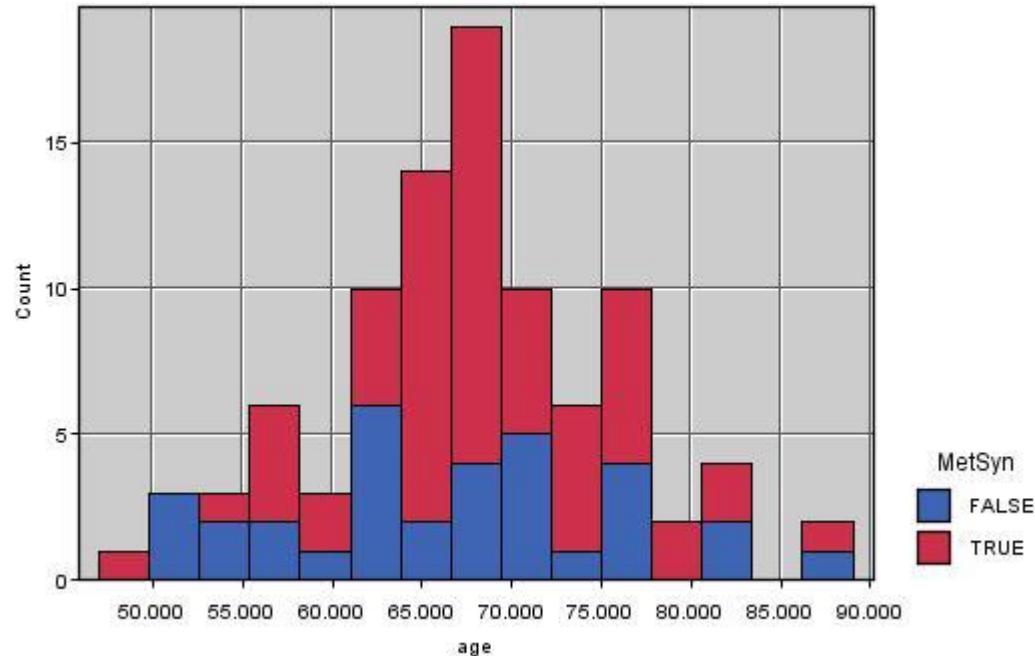
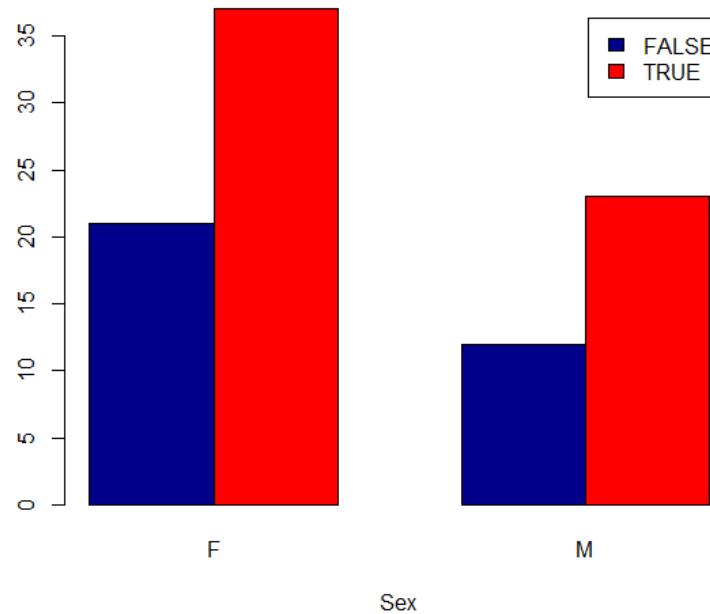
Rada 2, ako správne použiť jednotlivé štatistické metódy

**SPRÁVNE STANOVTE A CHÁPTE
VSTUPNÉ HYPOTÉZY (H_0, H_1)**

Pochopenie dát (4)



MetSyn

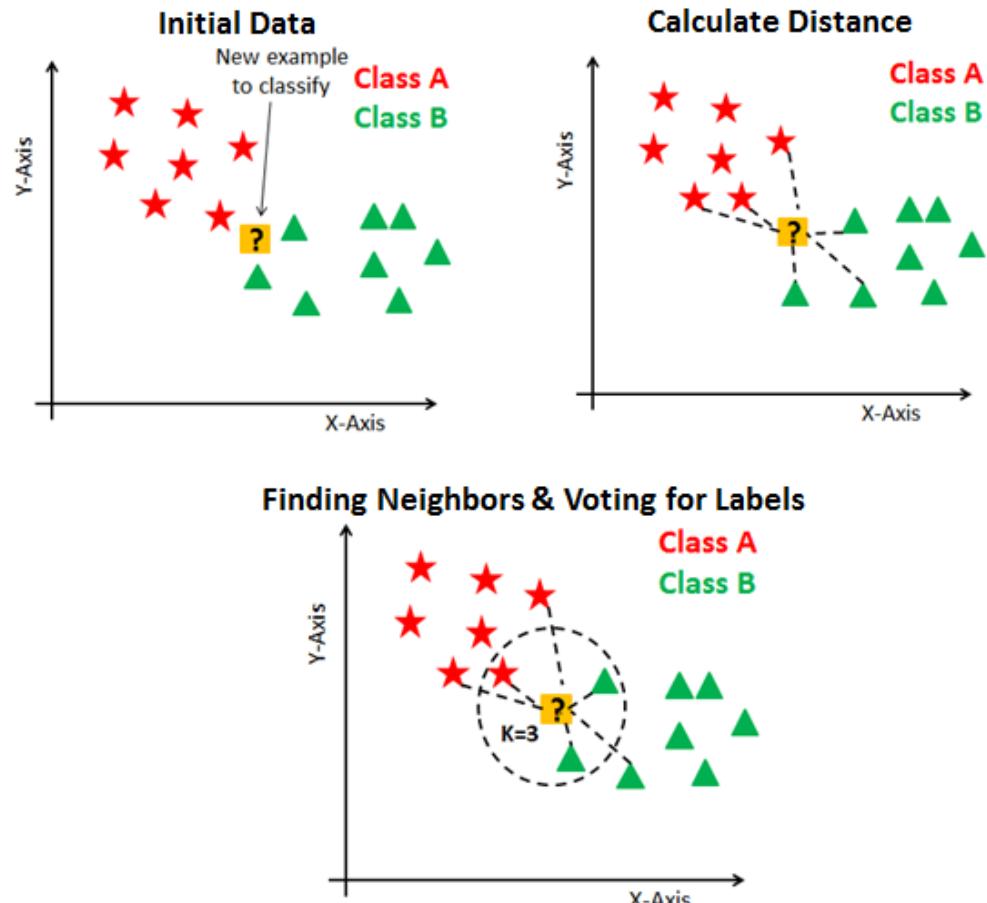


Dôležité zistenia pre ďalší postup

- Vyváženosť hodnôt cieľového atribútu.
 - Nadvzorkovanie, podvzorkovanie.
- Výskyt a početnosť chýbajúcich hodnôt v dátovej vzorke.
 - Doplnenie pomocou vhodnej metódy.
- Potenciálne dôležité a skryté závislosti medzi vstupnými atribútmi.
 - Výber najdôležitejších atribútov, odstránenie redundancie, zníženie dimenzie.

Príprava dát (1)

- Chýbajúce hodnoty boli doplnené pomocou metódy k-najbližších susedov.

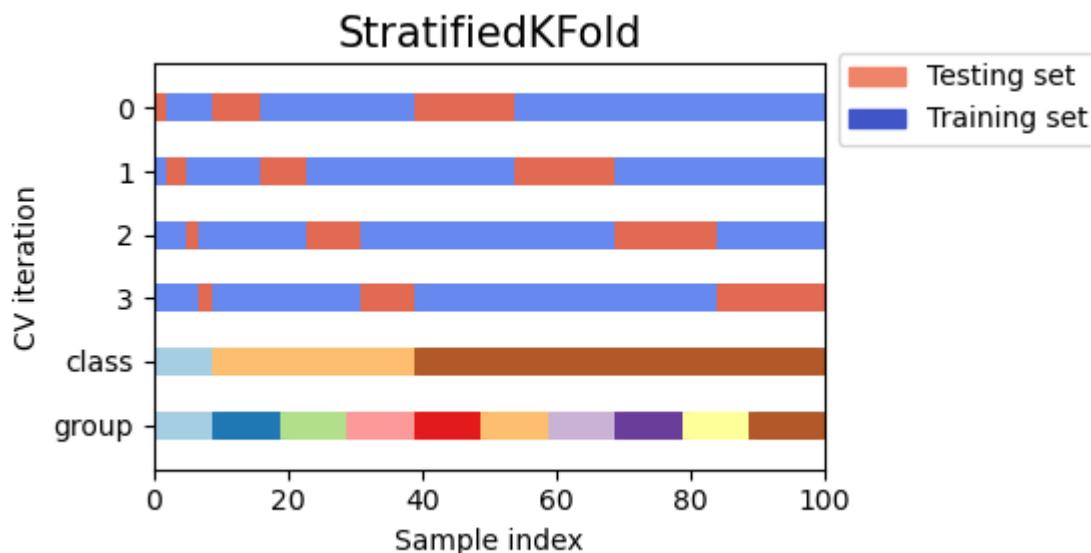


Príprava dát (2)

- Korelačná analýza: NEU vs. LY (- 0,922), HB vs. HTC (0,963) and E vs. HTC (0,812) – minimálna hranica 0,8.
- Odstránili sme atribúty NEU a HTC.
- Medicínska literatúra a experti hovoria:
 - Neutrofily (typ bielych krviniiek), Lymfocyty (typ bielych krviniiek).
 - Hemoglobin (množstvo krvného farbiva), hemotokrit (pomer objemu červených krviniiek k celkovému objemu krvi), erytrocyty (množstvo červených krviniiek v krvi).

Príprava dát (3)

- Rozdelenie na trénovaciu a testovaciu vzorku = stratifikovaný hold-out.
- Použitie stratifikovanej 10-násobnej krížovej validácie.



Rada 3, ako správne predspracovať dátá
**TESTOVACIU VZORKU
NEUPRAVUJTE!**

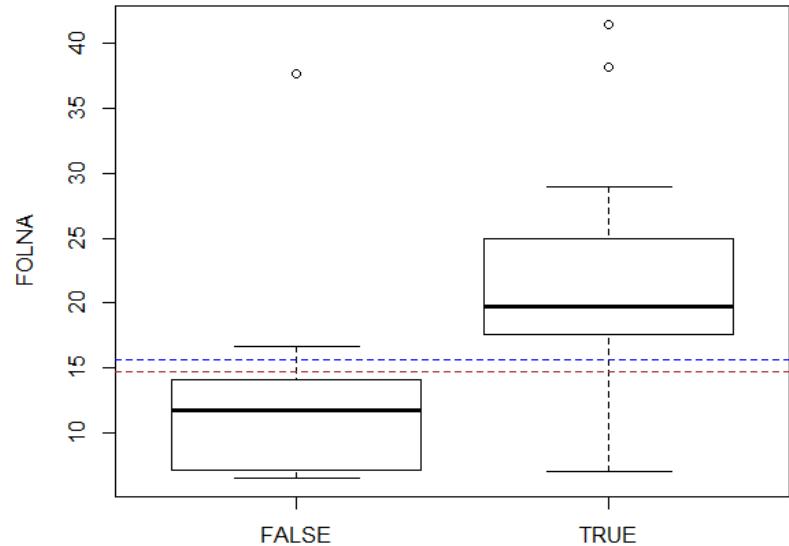
MODELOVANIE A VYHODNOTENIE

Exploračná analýza dát

- Exploračná analýza dát
- Identifikácia nových hraničných hodnôt
- Youdenov index

$J = \text{senzitivita} + \text{špecifickosť} - 1$

$J = \frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}} - 1$



Atribút FOLNA, muži, (červená čiara predstavuje hraničný bod nájdený rozhodovacím stromom, modrá hraničný bod nájdený štatistickou analýzou)

Binárna klasifikácia

Rozhodovacie stromy: C4.5, C5.0, Náhodné lesy

Logistická regresia: binárna, multinomiálna

Zhlukovanie: K-Means, K-NN

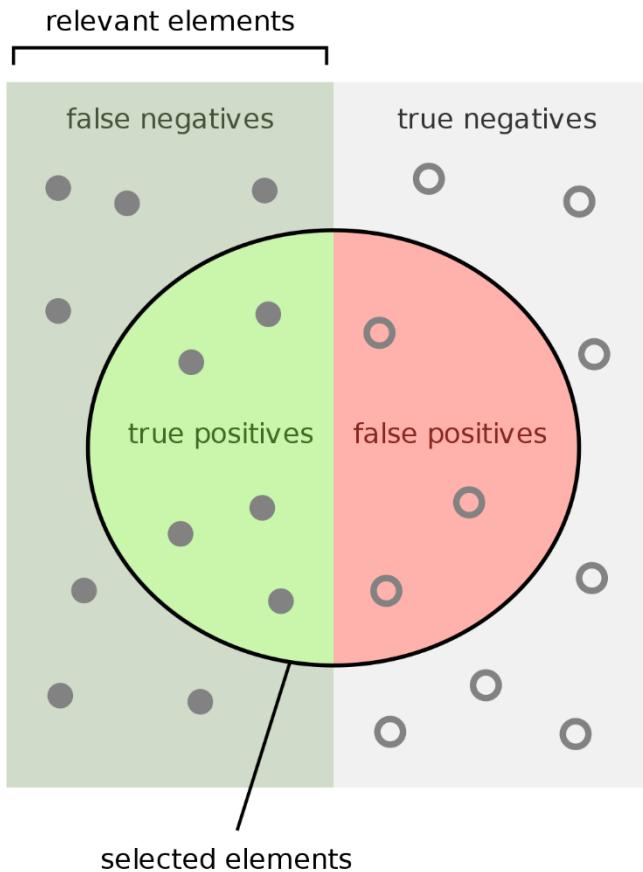
Premenná	Hraničná hodnota	Senzitivita (%)	Špecifita (%)	PPV (%)	NPV (%)
FOLNA (M)	15,6	95,65	83,33	91,67	90,91
HbA1c (M)	4,5	39,13	100	100	46,15
MO (ž)	5,5	86,5	14,3	64	37,5
TSH (ž)	2,69	22,22	100	100	41,67

$$TPR = \frac{TP}{(TP + FN)}$$

$$PPV = \frac{TP}{(TP + FP)}$$

$$TNR = \frac{TN}{(TN + FP)}$$

$$NPV = \frac{TN}{(TN + FN)}$$



How many relevant items are selected?
e.g. How many sick people are correctly identified as having the condition.

$$\text{Sensitivity} = \frac{\text{true positives}}{\text{relevant elements}}$$

How many negative selected elements are truly negative?
e.g. How many healthy people are identified as not having the condition.

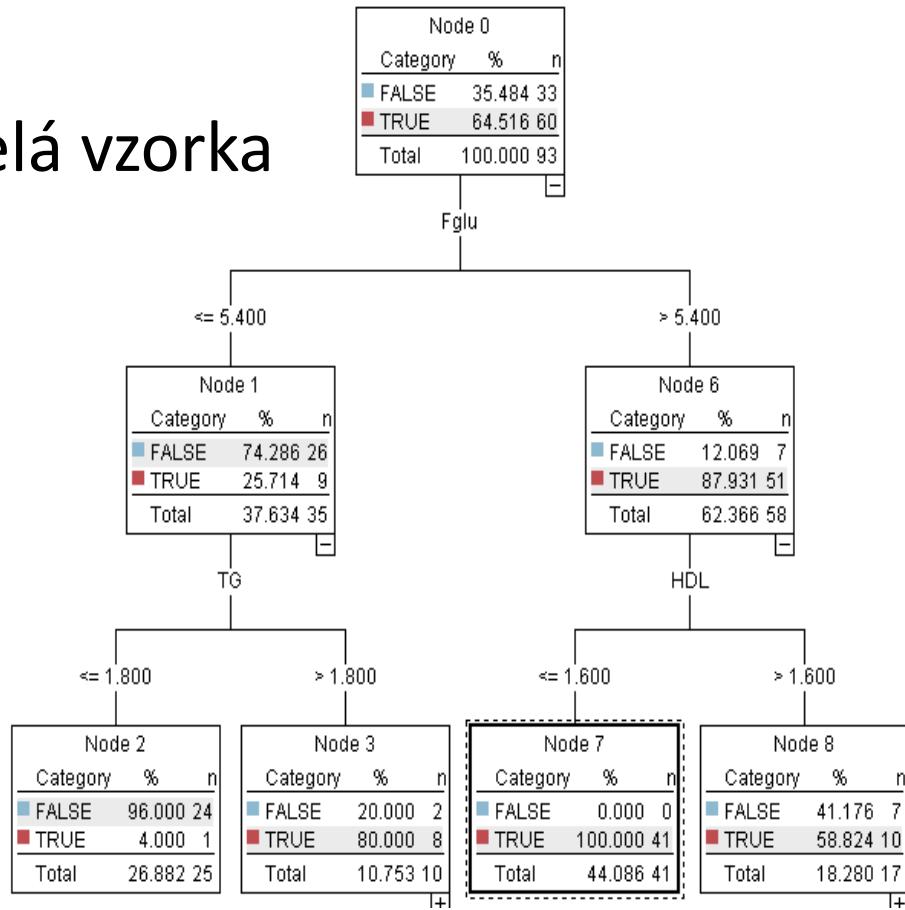
$$\text{Specificity} = \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}$$



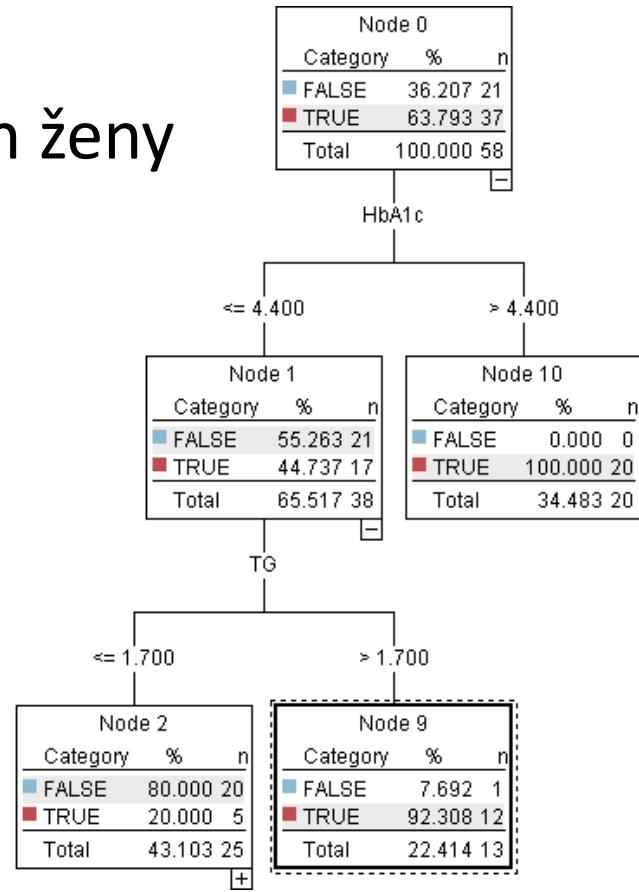
Rada 4, ako správne vypočítať tieto metriky
**SKONTROLUJTE, AKO DANÝ BALÍK
ALEBO PROG. JAZYK ZOBRAZUJE
PREDIKOVANÉ A REÁLNE HODNOTY.**

Rozhodovacie stromy - ukážka

celá vzorka



len ženy



Asociačné alebo rozhodovacie pravidlá

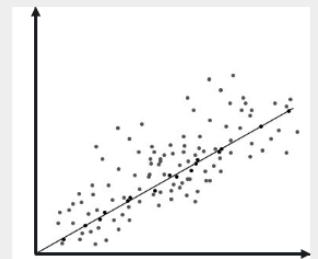
- **AK** index telesnej hmotnosti (BMI) je > 25.391 ,
POTOM u žien s trvaním hypertenzie (HYPDU) > 5 rokov sa bude MetSy prejavovať na 90.5%.
- **AK** index telesnej hmotnosti (BMI) je > 24.911 , **A** zároveň hladina glukózy nalačno (FGlu) je ≤ 5.3 , **A** zároveň hodnota HDL cholesterolu (HDL) je ≤ 1.25 ,
POTOM u žien s trvaním hypertenzie (HYPDU) > 10 rokov sa bude MetSy prejavovať na 85.7%.

Logistická regresia

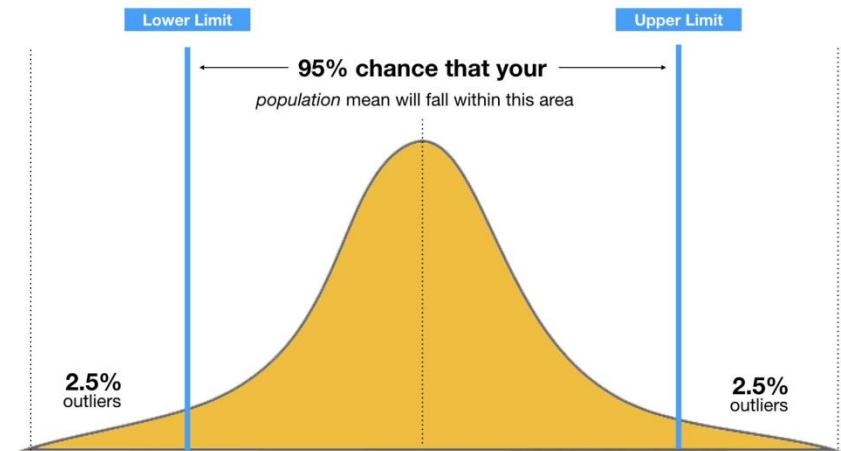
$$\log = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots \beta_n x_n$$

Parameter	z value	Pr(> z)	OR	CI	
				5%	95%
Sta Yes	5.90	0.36e-08***	15.829	7.329	34.186
BB Yes	2.04	0.040*	2.685	1.213	5.945
met Yes	3.54	0.4e-03***	44.471	7.622	259.450
anal Yes	1.01	0.311	1.599	0.745	3.433

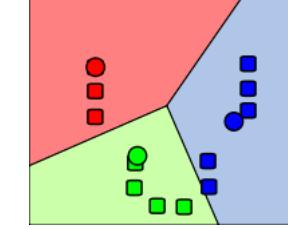
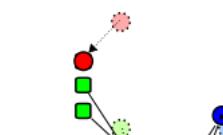
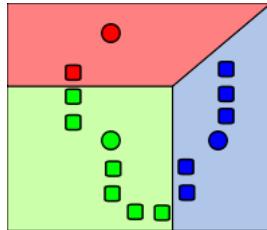
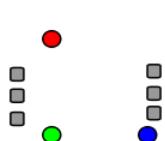
Multicollinearity is a statistical phenomenon in which two or more variables in a regression model are dependent upon the other variables in such a way that one can be linearly predicted from the other with a high degree of accuracy.



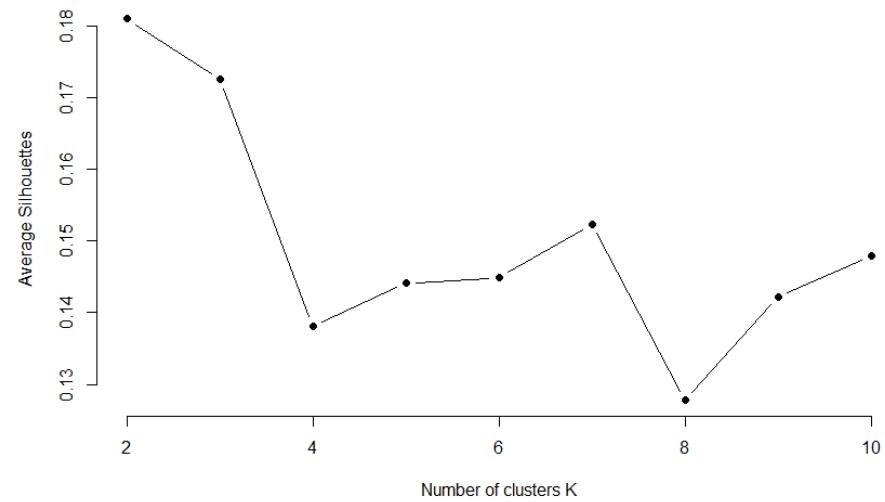
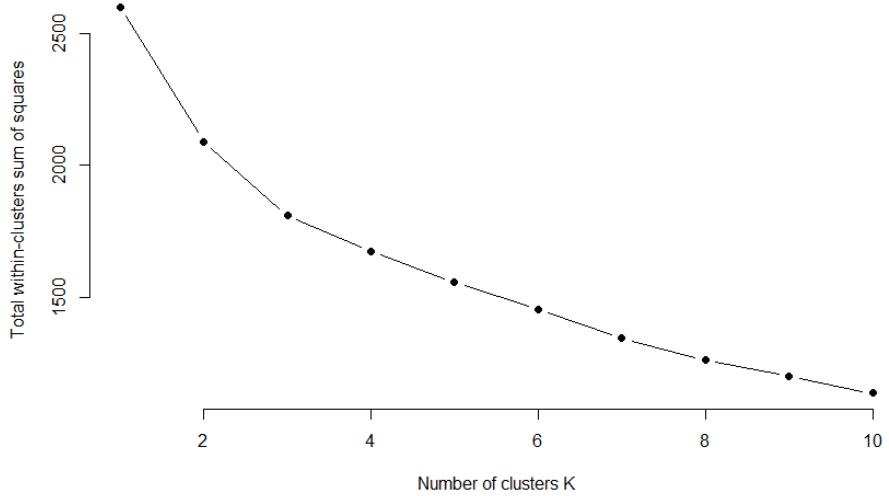
WallStreetMojo



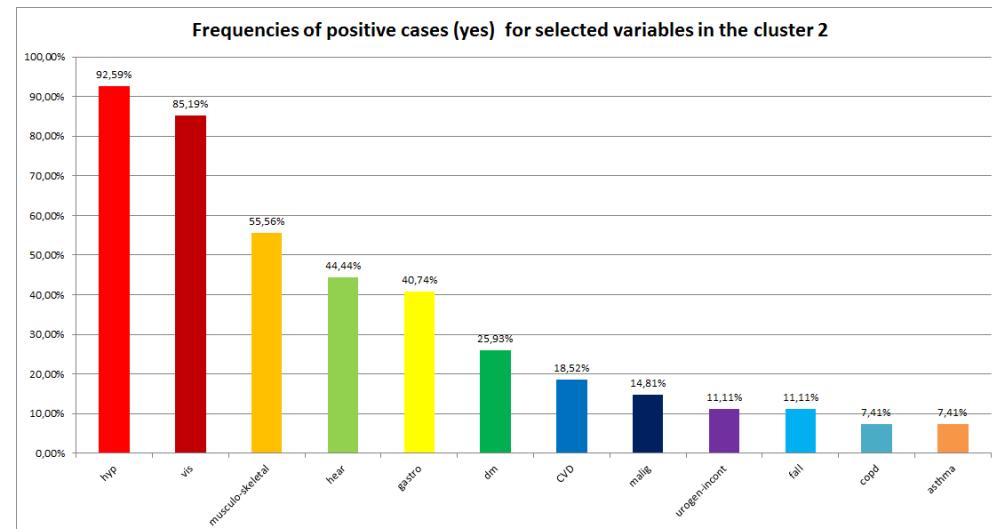
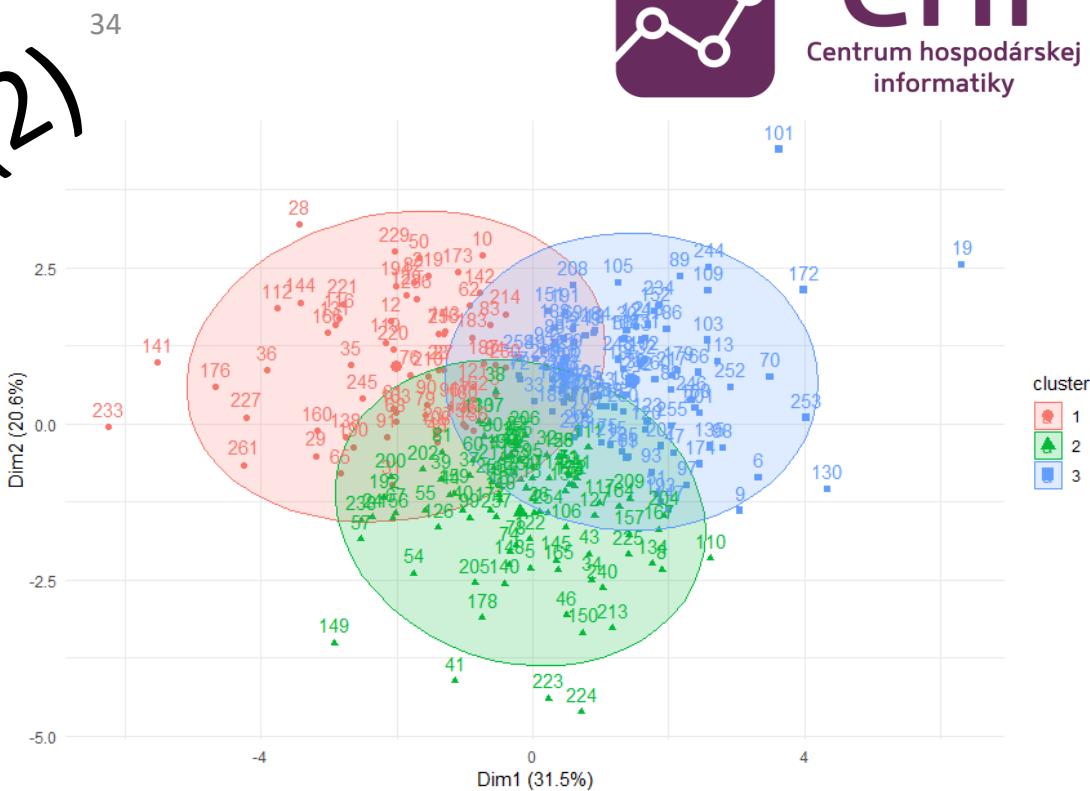
Zhluková analýza (1)



Zdroj: www.wikipedia.org



Zhluková analýza (2)



KRUSKAL-WALLIS TEST

Population Mean Ranks Equal?

1 metric / ordinal outcome variable on 3(+) groups

group	outcome
1	7
1	2
1	3
2	4
2	8
2	6
3	1
3	9
3	5

Nasadenie (1)



- Prediktívna medicína
- Preventívna medicína
- Personalizovaná medicína
- Participatívna medicína

Nasadenie (2)

Inteligentný rozhodovací systém na podporu medicínskej diagnostiky.

- Vstup: príznaky identifikované pacientom alebo hodnoty meraní pomocou medicínskych prístrojov.
- Báza znalostí: pravidlá charakteristické pre danú cielovú skupinu pacientov s potvrdenou diagnózou.
- Výstup: pravdepodobnosť pozitívnej diagnostiky pre nového pacienta.

Výsledok v jednoduchej a zrozumiteľnej forme.

Rada 5, na záver

- Okrem najbežnejšie používaných metrík v prípade klasifikačných modelov používajte aj ROC a AUC. Pohľad na metriky je možné doplniť o CI 95% (confidence interval, interval spoľahlivosti).
- Predtým ako spustíte logistickú regresiu, zamyslite sa, či máte k dispozícii dostatočný počet vstupných záznamov (pravidlo 1:10).
- Nebojte sa používať aj iné klasifikačné metódy ako stromy, napríklad NB, NN, SVM, kNN.
- V realite nedostanete dáta v Exceli, nezabúdajte na SQL!

**ĎAKUJEM ZA POZORNOSŤ ☺
OTÁZKY ???**

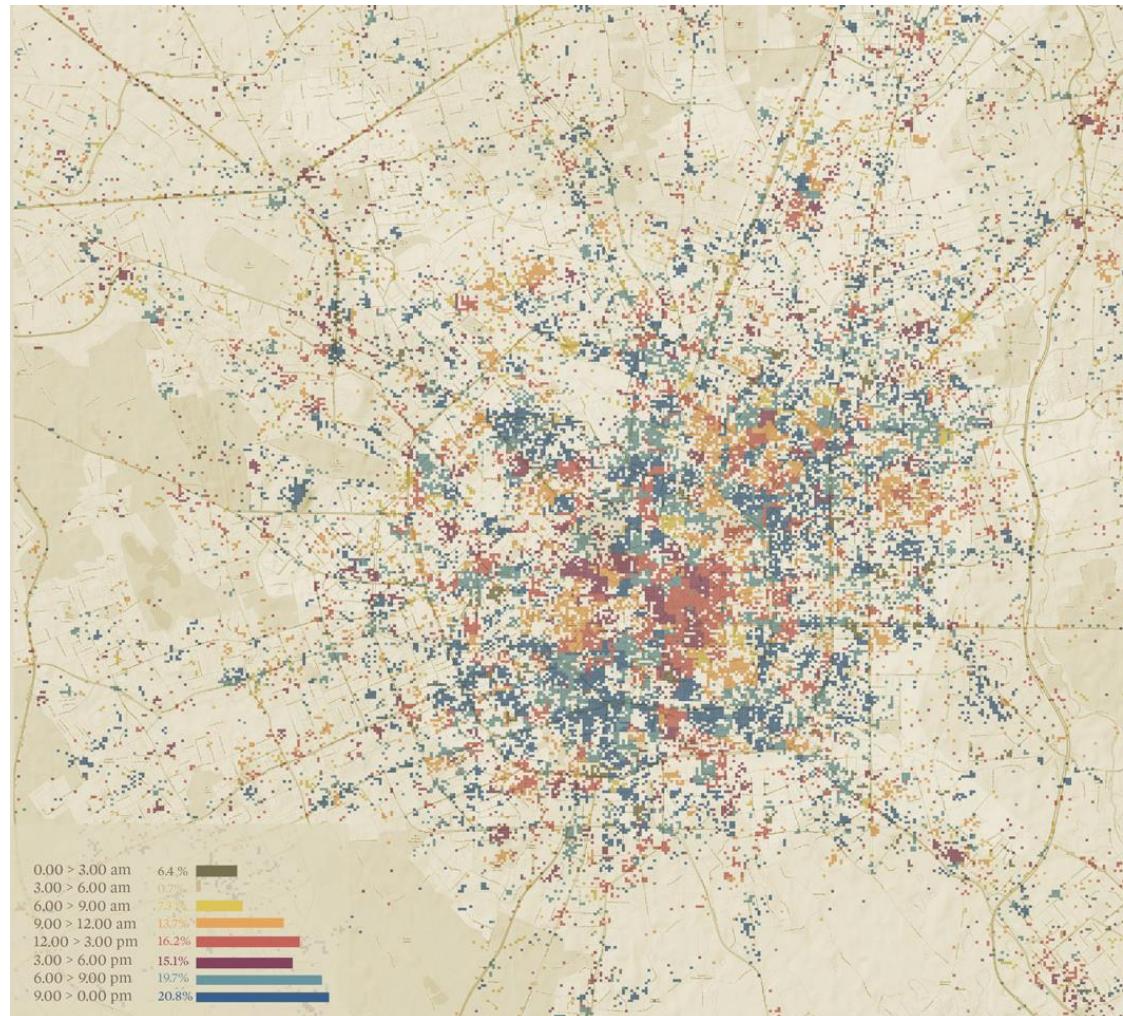
TEXT MINING

Objavovanie znalostí v textoch

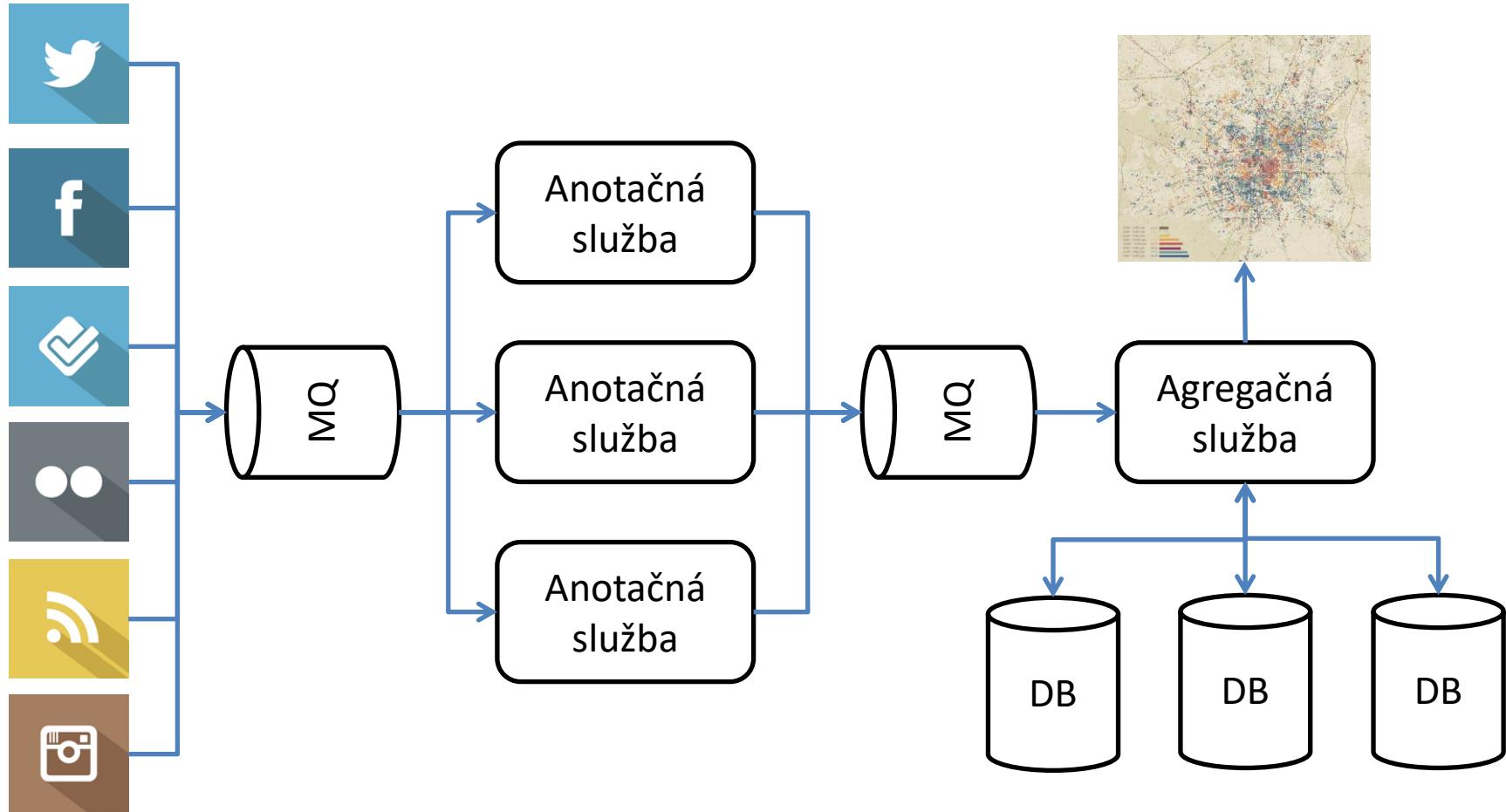
Projekt *Urban Sensing (1)

- Integrovanie obsah zo sociálnych sietí (Twitter, Facebook, Instagram, Foursquare, ...)
- Extrahovanie pomenovaných entít a analýza sentimentu textového obsahu
- Agregovanie dát podľa rôznych kritérií v reálnom čase
- Interaktívna vizualizácia dát (základné zobrazenie podľa geografickej mapy)
- Požadovaná škálovateľnosť do 10 000 príspevkov za s
- Objem dát cca 1T/rok/oblasť záujmu

Projekt *Urban Sensing (2)



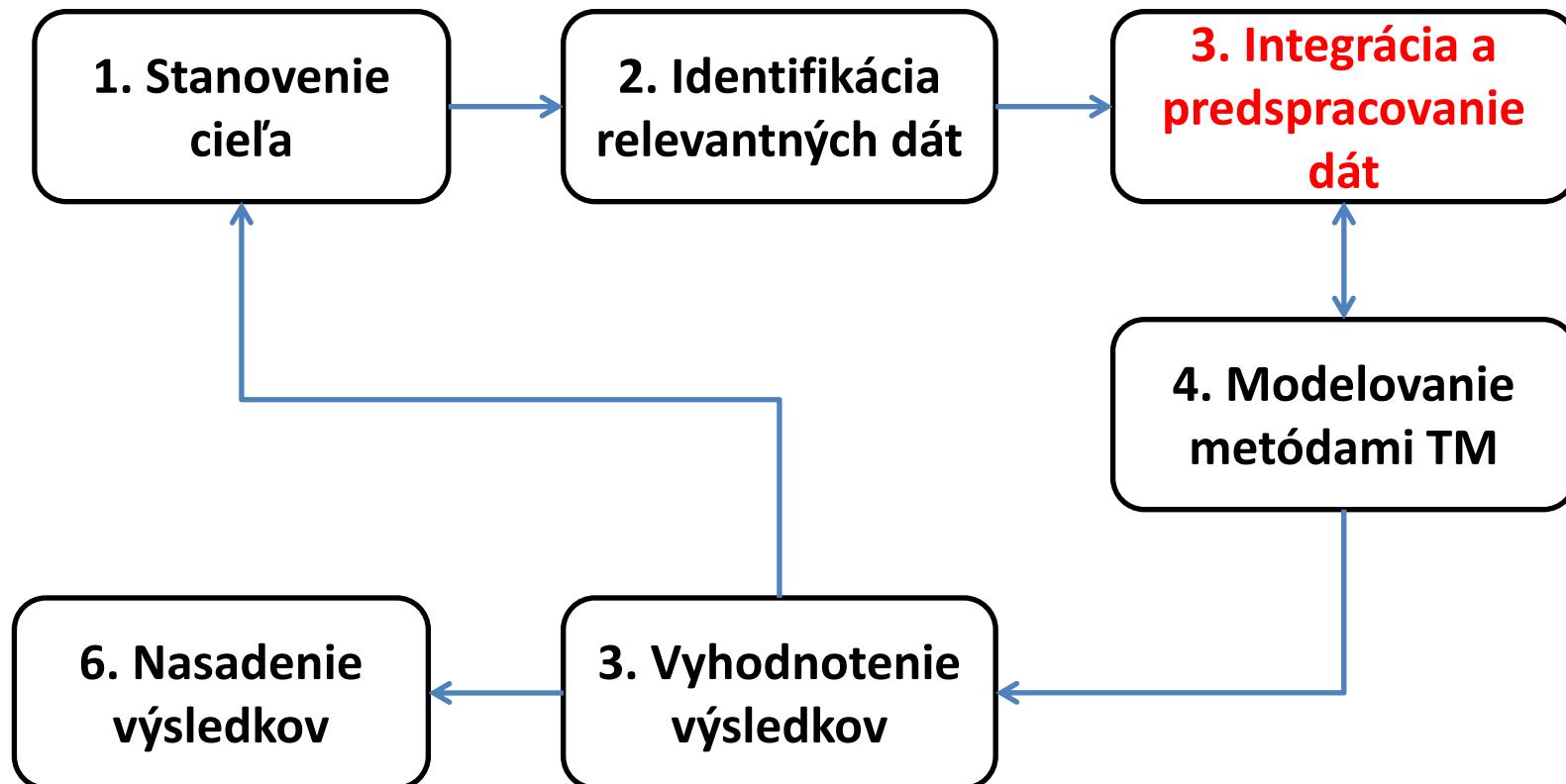
Architektúra systému *Urban Sensing



Úlohy dolovania z textov

- **Klasifikácia**
 - Zaradenie dokumentu do preddefinovaných kategórií
- **Zhlukovanie**
 - Nájdenie a popis zhlukov podobných dokumentov
- **Extrahovanie tém**
 - Vyextrahovanie hlavných tém v dokumentoch
- **Analýza sentimentu**
 - Určenie polarity textu
- **Extrahovanie informácií**
 - Extrahovanie entít, udalostí, vzťahov a faktov

Proces objavovania znalostí v textoch



Integrácia a harmonizácia dát

- Zo vstupného formátu vyextrahujeme **čistý text**
- Zo vstupného formátu vyextrahujeme **metainformácie**
 - **Bibliografické údaje** napr. informácie o autorovi, dátum publikovania, pôvodný zdroj alebo umiestnenie dokumentu, zdrojový formát dokumentu a pod.
 - **Metaúdaje** vložené autorom, ktoré dodatočne charakterizujú obsah, ako napr. **kľúčové slová**, zaradenie dokumentu do kategórií a pod.
 - **Linky**, ktoré odkazujú na súvisiace dokumenty
- Snažíme sa čo najviac zjednotiť štruktúru metainformácií, tak aby bola spoločná pre rôzne vstupné formáty

Integrácia dát – príklad Twitter

JSON objekt získaný z Twitter služby:

```
{  
    "coordinates": null,  
    "created_at": "Thu Oct 21 16:02:46 2015",  
    "id": 0123456789,  
    "entities": {  
        "urls": [],  
        "user_mentions": [ ],  
        "hashtags": [ {  
            "name": "FridayReads",  
            "id": 123654789, ...} ]  
    },  
    "text": "I'm loving 'The Sound of Things Falling' by Juan Gabriel Vasquez #FridayReads",  
    "user": {  
        "name": "lenadunham",  
        "id": 987456321,  
        ...  
    },  
    ...  
}
```

Extrahovaný čistý text:

I'm loving 'The Sound of Things Falling' by
Juan Gabriel Vasquez #FridayReads

Extrahované metainformácie:

autor: lenadunhan

dátum vytvorenia: 21.10.2015

zdroj:

<https://api.twitter.com/1.1/statuses/show.json?id=0123456789>

zdrojový formát: application/json

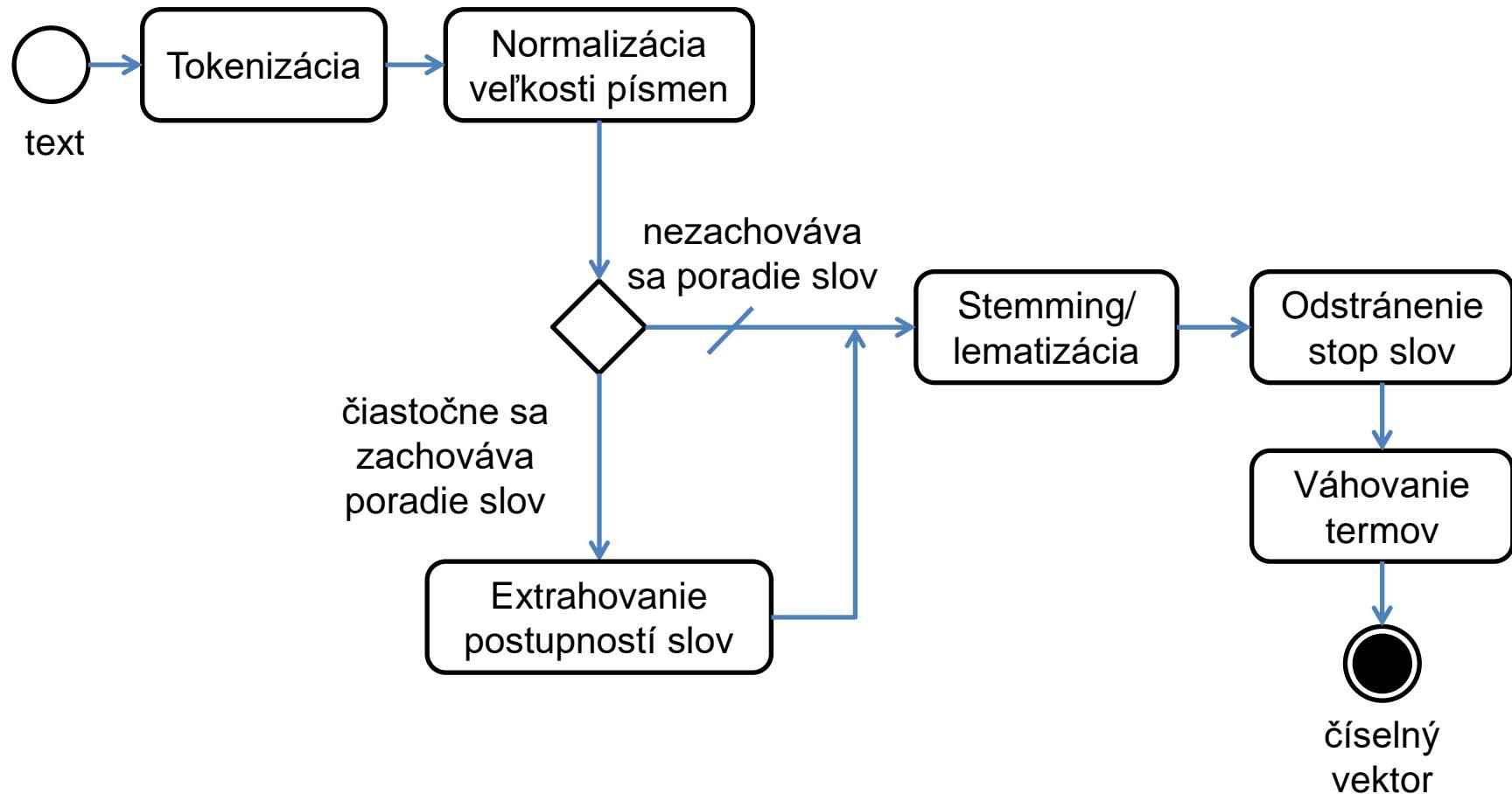
kľúčové slová: FridayReads

Predspracovanie textov

- Pred aplikovaním metód text miningu je potrebné previesť text do **štruktúrovanej reprezentácie**
- Najčastejšie sa používa tzv. **vektorová reprezentácia** textov – z množiny dokumentov sa vytvorí tzv. **dokument-term matica**, ktorej riadky zodpovedajú dokumentom a stĺpce zodpovedajú jednotlivým termom (slovám), ktoré sa vyskytli v niektorom z dokumentov
- Hodnota $x_{i,j}$ na i -tom riadku a j -tom stĺpci udáva váhu ako daný term j reprezentuje obsah dokumentu i
 - Najjednoduchšie je tzv. **binárne váhovanie** - $x_{i,j} = 0$ alebo 1
 - Najčastejšie **Tf-Idf** váhovanie:

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

Vektorová reprezentácia textov



Tokenizácia

- Rozdelenie textu na základné lexikálne jednotky - **tokeny**, ako napr. slová, interpunkčné znaky, číselné údaje, dátumy, internetové adresy a pod.
- Vstup: postupnosť znakov
- Výstup: postupnosť tokenov
- Pre európske jazyky sú slová oddelené medzerou alebo interpunkčnými znakmi - – () , . ; : / „ ? ! ...
 - Najčastejšie sa text rozdelí pomocou regulárnych výrazov
 - Výnimky sa následne ošetria podľa slovníka
 - Skratky, názvy, zložené výrazy
 - **viac - menej → viac-menej**

Extrahovanie termov

- Cieľom je previesť tokeny na termy, ktoré budú zahrnuté do vektorovej reprezentácie
 - Vstup: postupnosť tokenov
 - Výstup: postupnosť termov
-
1. Znormujeme veľkosť písmen (výnimkou môžu byť názvy a skratky, kde má striedanie veľkých a malých písmen význam)
 2. Prevedieme slovo na jeho základný tvar (tzv. lemu - lematizácia) alebo koreň slova (tzv. stem - stemming), t.j. všetky tvary toho istého slova budú namapované na jeden term
 3. Odstránime neplnovýznamové slová (spojky, predložky, častice, zámená, čísla), keďže pri zanedbaní poradia nemajú veľký vplyv na reprezentáciu významu

Stemming

- Pre daný jazyk sa navrhnu pravidlá, ktoré prevedú rôzne tvary slova na ich spoločný koreň odstránením prípon a predpôn
- Relatívne presná metóda pre jazyky s jednoduchšou morfológiou, napr. pre angličtinu {clos-ing, clos-es, clos-er, clos-e} → clos+0
- Stemming nemusí byť jednoznačný, t.j. slová s rôznym významom môžu byť namapované na ten istý koreň
- Koreň už nemusí byť slovom (nemusí byť úplne zrozumiteľný)
- Pravidlá majú tvar [podmienka] S1 → S2: ak je splnená podmienka (napr. ak slovo končí na -s alebo obsahuje spoluhlásku), potom nahradí predponu/príponu S1 reťazcom S2
- Pre angličtinu sa najčastejšie používa **Porterov stemmer**

Odstránenie stop slov

- Ak zanedbáme poradie slov, najdôležitejšie pre reprezentáciu významu textu sú slovesá, podstatné mená a prídavné mená
- Odstránením neplnovýznamových slov sa zníži rozmer príznakového priestoru (neplnovýznamové slová spôsobujú pri modelovaní dátový šum)
- Odstráňa sa slová zo slovníka tzv. **stop slov**
 - Spojky, predložky, častice, zámená, pomocné slovesá
 - Napr. pre slovenčinu: a, aby, aj, ako, ale, alebo, ani, áno, asi, bez, by, byť, cez, čo, či, dnes, do, další, ešte, ho, i, iba, ja, je, jeho, jej, k, kam, každý ...
- Môžu sa odstrániť aj čísla, dátumy, a pod., ako napr. 1 000, 26.10.
2015

Extrahovanie postupností slov

- Niektoré mená a názvy a veľa odborných termínov je tvorených viacerými slovami (**mennými frázami**) - mali by byť indexované ako jeden term, aby sa zachoval ich význam
- Postupnosti slov je možné vyextrahovať štatisticky **spočítaním frekventovaných n -gramov**, t.j. n za sebou idúcich slov
 - Jazykovo nezávislá metóda
 - Vyextrahované postupnosti nemusia byť gramaticky správne frázy
 - Extrahovanie gramatických fráz si vyžaduje zložitejšiu morfológickú a syntaktickú analýzu

Predspracovanie textov v R (1)

```
library("twitteR")
library("wordcloud")
library("tm")

# prihlásenie k Twitter službe
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

# získame posledných 1500 tweetov
r_stats <- searchTwitter("#Rstats", n=1500)

# vyberieme z nich textový obsah
r_stats_text <- sapply(r_stats, function(x) x$getText())
# a vytvoríme korpus - textovú dátovú množinu
r_stats_text_corpus <- Corpus(VectorSource(r_stats_text))

# prevedieme text na malé písmena, odstráňme interpunkciu a stop slová
r_stats_text_corpus <- tm_map(r_stats_text_corpus, content_transformer(tolower))
r_stats_text_corpus <- tm_map(r_stats_text_corpus, removePunctuation)
r_stats_text_corpus <- tm_map(r_stats_text_corpus,
function(x)removeWords(x,stopwords()))
```



Predspracovanie textov v R (2)

```
wordcloud(r_stats_text_corpus, min.freq=25, max.words=100)
```

```
# vytvoríme dokument-term maticu  
dtm <- DocumentTermMatrix(r_stats_text_corpus)  
findFreqTerms(dtm, lowfreq=50)  
removeSparseTerms(dtm, 0.4)
```



Redukcia príznakového priestoru

- Dokument-term matica je veľká (počet dokumentov $n \times$ počet termov m) a **riedka** (obsahuje veľa 0 hodnôt)
- Zvyčajne iba malá časť termov je dôležitá pre danú úlohu objavovania znalostí, veľká časť termov je irrelevantná a spôsobuje šum pri štatistickom spracovaní, preto je výhodné použiť **metódy redukcie príznakového priestoru:**
 - **Selekcia termov** - z pôvodnej množiny termov sa vyberie iba podmnožina r termov
 - **Redukcia termov** - pôvodné termy sa nahradia novými r príznakmi ($r < m$), ktoré sú kombináciou pôvodných termov (transformované príznaky už nemusia byť ľahko interpretovateľné ako slová)

Úlohy dolovania z textov

- Klasifikácia
 - Zaradenie dokumentu do preddefinovaných kategórií
- Zhlukovanie
 - Nájdenie a popis zhlukov podobných dokumentov
- Extrahovanie tém
 - Vyextrahovanie hlavných tém v dokumentoch
- Analýza sentimentu
 - Určenie polarity textu
- Extrahovanie informácií
 - Extrahovanie entít, udalostí, vzťahov a faktov

Extrahovanie tém (1)

- Úlohou je 1) nájsť témy zastúpené v množine dokumentov a 2) popísat' témy tak aby ich bolo možné interpretovať
- Najčastejšie sa používajú nekontrolované metódy
- Vhodná je vektorová reprezentácia + slovné spojenia
- Predpoklady:
 - Obsah jedného dokumentu môže byť zložený z viacerých témy
 - Témy je možné reprezentovať možinou charakteristických slov alebo fráz
 - Jedno slovo môže vyjadrovať rôzne témy (v každej sa však vyskytuje v kontexte iných slov)

Extrahovanie tém (2)

- Vyhodnotenie
 - Na nezávislej množine sa otestuje ako dobre model dokáže popísat nové dátá (za predpokladu rovnakého zastúpenia tém)
 - Dôležitá je interpretácia expertom a vizualizácia výsledkov

Pravdepodobnostný model tém

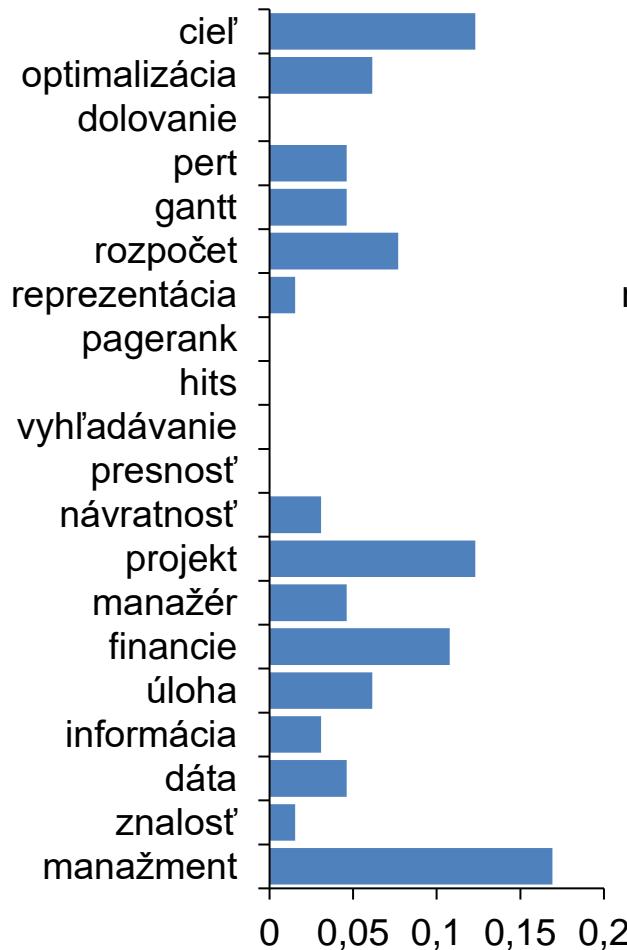
- Pravdepodobnostný model tém pre danú množinu n dokumentov a počet tém k priradí:
 - Pre každú tému t rozdelenie pravdepodobnosti $\beta_t = (\beta_{t,1}, \beta_{t,2}, \dots, \beta_{t,m})$, $\beta_{t,i} \in (0, 1)$, $\beta_{t,1} + \beta_{t,2} + \dots + \beta_{t,m} = 1$, kde pravdepodobnosť $\beta_{t,i}$ určuje, do akej miery term i vyjadruje tému t
 - Pre každý dokument j rozdelenie pravdepodobnosti $\theta_j = (\theta_{j,1}, \theta_{j,2}, \dots, \theta_{n,j})$, $\theta_{j,t} \in (0, 1)$, $\theta_{j,1} + \theta_{j,2} + \dots + \theta_{j,n} = 1$, kde pravdepodobnosť $\theta_{j,t}$ určuje, do akej miery dokument j obsahuje tému t

Latentná Dirichletova Alokácia – LDA

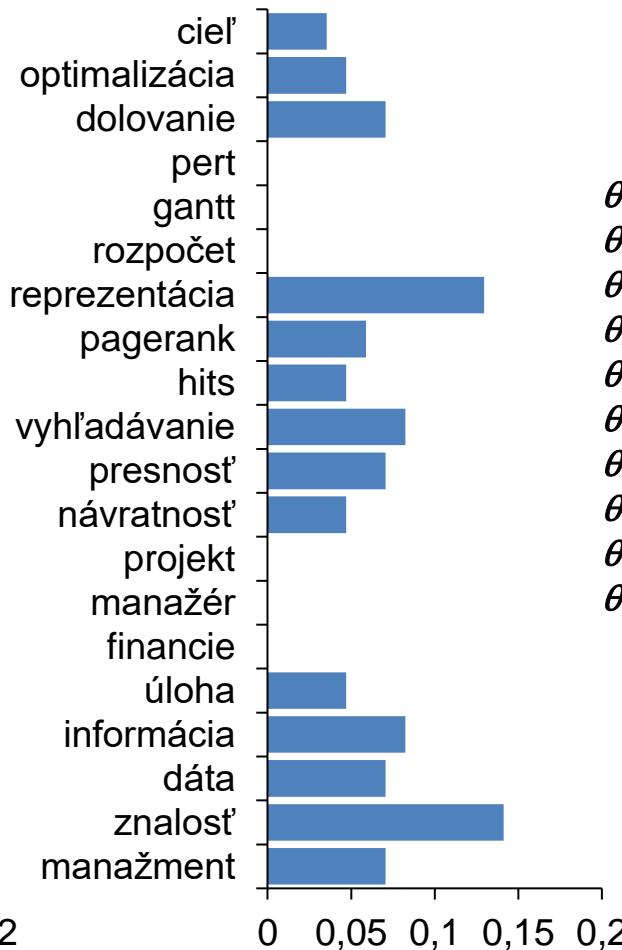
- **Metóda LDA** je pravdepodobnostná metóda založená na predpoklade, že pravdepodobnosti termov pre každú tému a pravdepodobnosti tém pre každý dokument majú Dirichletové rozdelenie
- Vstupné parametre:
 - k - počet extrahovaných tém
 - α_t - parameter Dirichletovho rozdelenia pre priradenie termov, určuje počet charakteristických termov pre jednu tému
 - α_d - parameter Dirichletovho rozdelenia pre priradenie tém, určuje predpokladaný počet rôznych tém v jednom dokumente
- Vstupné dátá: term-dokument matica s frekvenciami termov (*nnn* váhovanie)

LDA – príklad

β_1 - Téma 1



β_2 - Téma 2



Téma 1 Téma 2

$$\begin{aligned}\theta_1 &= (1,12E-03, \textcolor{red}{9,99E-01}) \\ \theta_2 &= (9,95E-04, \textcolor{red}{9,99E-01}) \\ \theta_3 &= (8,14E-04, \textcolor{red}{9,99E-01}) \\ \theta_4 &= (1,19E-03, \textcolor{red}{9,99E-01}) \\ \theta_5 &= (1,28E-03, \textcolor{red}{9,99E-01}) \\ \theta_6 &= (\textcolor{red}{9,99E-01}, 1,12E-03) \\ \theta_7 &= (\textcolor{red}{9,99E-01}, 1,05E-03) \\ \theta_8 &= (\textcolor{red}{9,99E-01}, 1,49E-03) \\ \theta_9 &= (\textcolor{red}{9,97E-01}, 2,55E-03) \\ \theta_{10} &= (\textcolor{red}{9,99E-01}, 1,38E-03)\end{aligned}$$

Analýza tém v dátových prúdoch

- Textové dáta sú publikované postupne v čase – napr. novinové články, správy na sociálnom webe, atď.
- Cieľom je analyzovať, ako sa témy menili v čase
 - Detegovať vznik novej témy, alebo ďalší výskyt predošej témy
 - Analyzovať trendy (stúpajúca/klesajúca populárnosť témy)
- Najjednoduchší spôsob je analyzovať celú množinu za dané obdobie a zobraziť histogram dokumentov zaradených do jednotlivých tém
- Rozšírené metódy okrem priradenia tém termom a dokumentom modelujú aj výskyt témy v čase – rozdelenie pravdepodobnosti pre časovú os

Interpretovanie tém

- Podľa pravdepodobnostného modelu:
 - Vieme zistiť, ktoré slová sú charakteristické pre danú tému (majú väčšiu pravdepodobnosť $\beta_{t,i}$)
 - Podľa $\theta_{j,t}$ vieme rozhodnúť, ktorý dokument obsahuje danú tému
- Okrem slov môžeme vyextrahovať vety, ktoré obsahujú čo najviac slov charakteristických pre danú tému
- Dôležitá je vizualizácia a interaktívne prehliadanie

Latentná Dirichletova Alokácia v R (1)

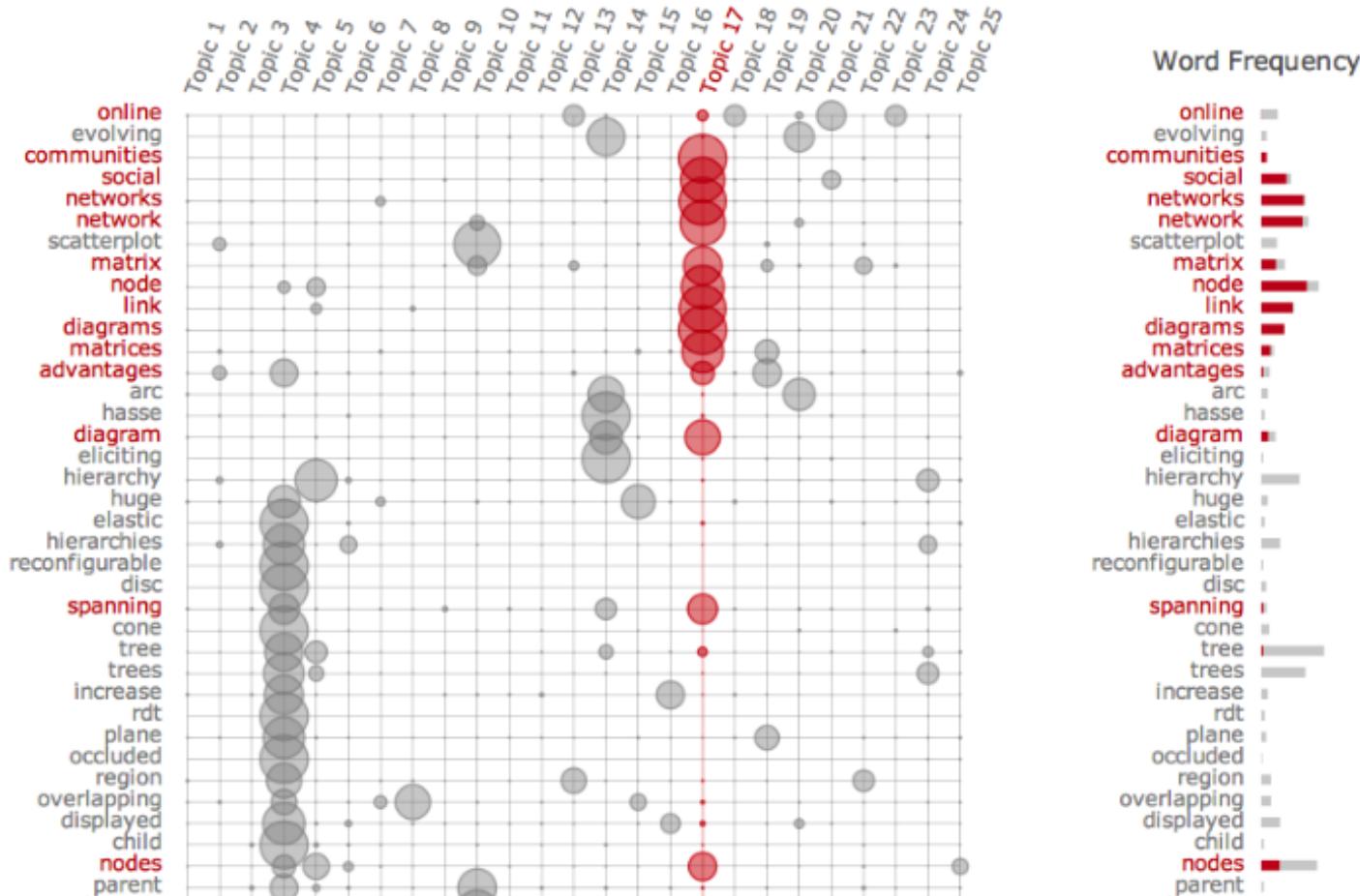
```
dtm <- as.DocumentTermMatrix(tdm)
library(topicmodels)
lda <- LDA(dtm, k = 8) # nájdi 8 tém
(term <- terms(lda, 6)) # prvých 6 termov pre každú tému

##          Topic 1        Topic 2        Topic 3        Topic 4        Topic 5 ...
## [1,] "r"           "data"       "mining"      "r"           "data" ...
## [2,] "example"     "introduction" "r"           "package"     "mining" ...
## [3,] "code"         "big"         "data"        "use"         "applicat...
## [4,] "mining"       "analysis"    "text"        "group"       "r" ...
## [5,] "data"         "slides"      "package"    "example"     "use" ...
## [6,] "rule"         "mining"     "time"       "cluster"    "due" ...

##          Topic 6        Topic 7        Topic 8
## [1,] "data"        "research"    "analysis"
## [2,] "r"           "position"    "r"
## [3,] "job"         "data"        "network"
## [4,] "mining"      "university" "computational"
## [5,] "lecture"     "analytics"   "tutorial"
## [6,] "university" "scientist"  "slides"
```



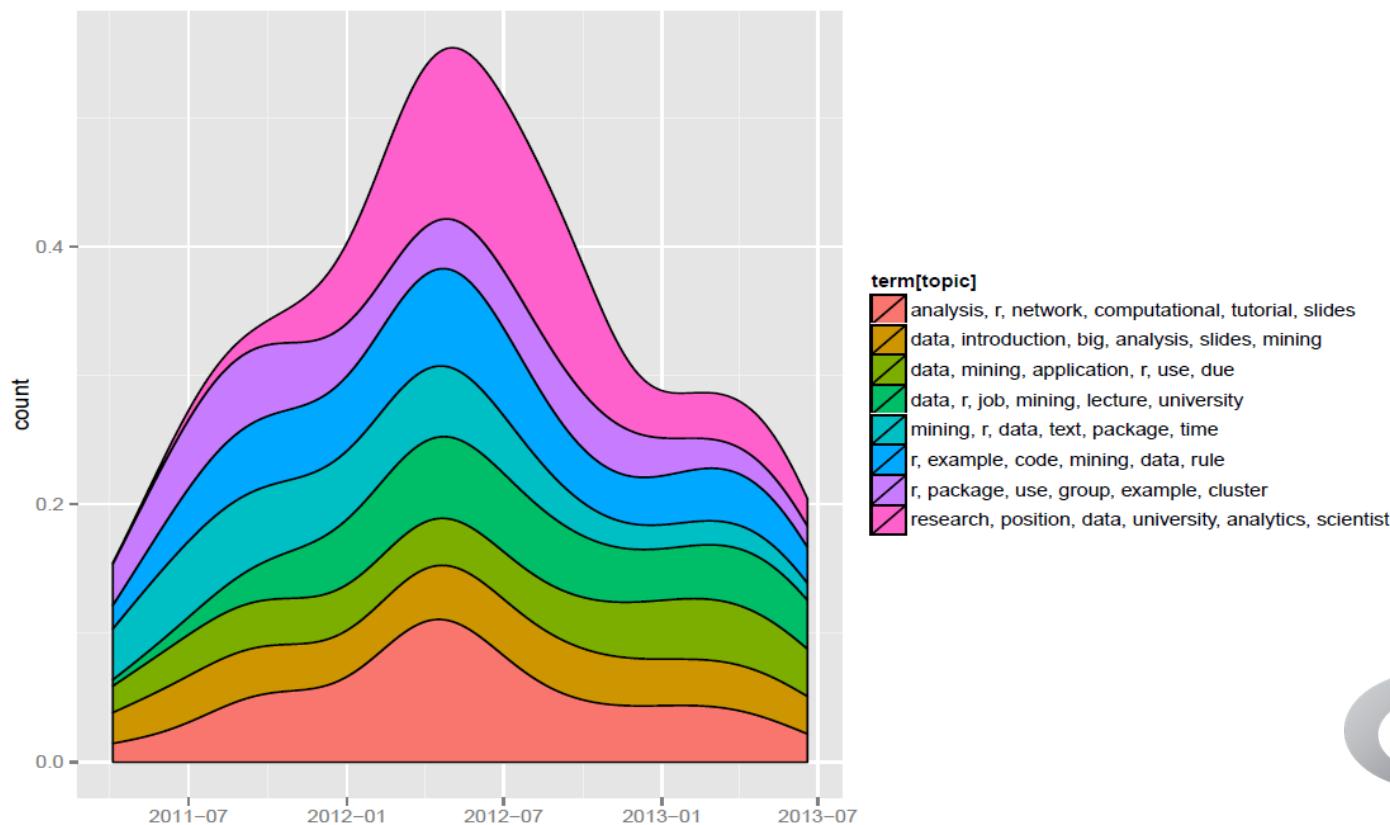
Latentná Dirichletova Alokácia v R (2)



<http://vis.stanford.edu/papers/termite>

Latentná Dirichletova Alokácia v R (3)

```
# vyber najpravdepodobnejšiu tému pre každý tweet
topic <- topics(1da, 1)
topics <- data.frame(date=as.IDate(tweets.df$created), topic)
qplot(date, ..count.., data=topics, geom="density", fill=term[topic], position="stack")
```



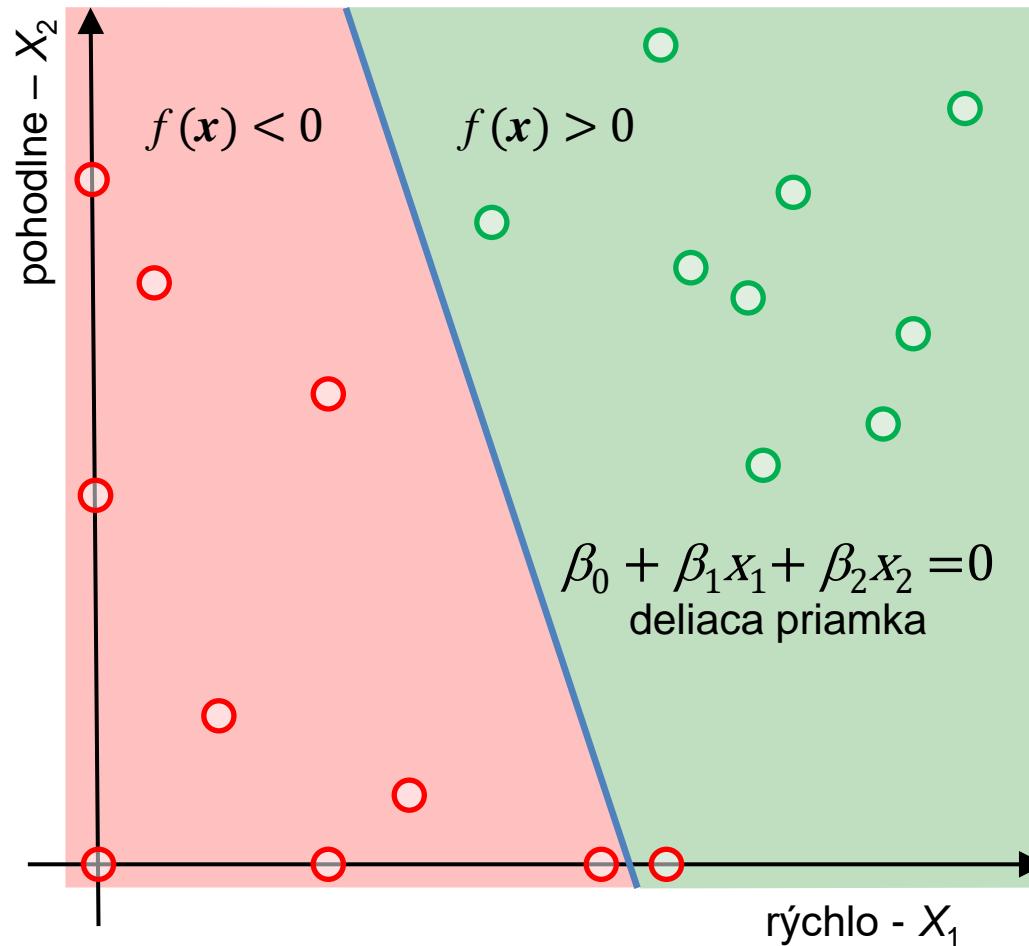
Analýza sentimentu

- Základným cieľom je priradiť textom subjektívnu polaritu – t.j. rozhodnúť, či je text pozitívny, alebo negatívny
- Rozšírená úloha rozlišuje viacero stupňov polarity (napr. počet hviezdičiek pri hodnotení filmov a pod.)
- Využitie hlavne v marketingu, starostlivosti o zákazníka, pri prieskumoch verejnej mienky
- Vhodná je vektorová reprezentácia + slovné spojenia, alebo kratšie postupnosti slov
- Vyhodnotenie na testovacej množine
 - Chyba klasifikácie a kontingenčná tabuľka
 - Subjektívne vnímanie môže spôsobiť nízku zhodu aj medzi ľuďmi (okolo 79%)

Metódy analýzy sentimentu

- Slovníkové metódy
 - Slovník pozitívnych/negatívnych slov + pravidlá pre stupňovanie a negáciu
- Kontrolované metódy učenia
 - Segmentovanie na vety, alebo krátke slovné spojenia, ktoré obsahujú subjektívny obsah môže zlepšiť presnosť
 - Lineárne klasifikátory (SVM, Naivný Bayesov klasifikátor, Logistická regresia)
- Kombinované metódy
 - Počiatočná klasifikácia slovníkovou metódou (tzv. *bootstrap*) + rozšírenie naučeným modelom

Lineárny model v 2-rozmernom priestore



Klasifikácia

- Chceme využiť lineárny model na klasifikáciu do dvoch tried +/-:

$$f(\mathbf{x}_i) = f(x_{i,1}, x_{i,2}, \dots, x_{i,m}) = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_m x_{i,m}$$

- Môžeme si zakódovať triedy ako čísla -1/1 a použiť lineárnu regresiu
– ak je $f(\mathbf{x}_i) < 0$, potom zarad' príklad do triedy -, ak > 0 potom do triedy +

Príklad klasifikačného modelu (2)

- Klasifikáciu je možné definovať aj pravdepodobnostne:
 - Podmienená pravdepodobnosť $P(y = +|x_{i,1}, x_{i,2}, \dots, x_{i,m})$ - ak poznáme vstupné hodnoty, aká je pravdepodobnosť, že má byť príklad zaradený do triedy +
 - Podobne $P(y = -|x_{i,1}, x_{i,2}, \dots, x_{i,m})$ aká je pravdepodobnosť, že má byť zaradený do triedy -
 - Pre dve triedy platí $P(y = +|x_i) = 1 - P(y = -|x_i)$
 - Ak by sme poznali $P(y = +|x_i)$ tak najmenšia chyba klasifikácie by bola ak by sme zaradili príklad do triedy + ak by $P(y = +|x_i) > 0.5$, inak do triedy -

Príklad klasifikačného modelu – logistická regresia (1)

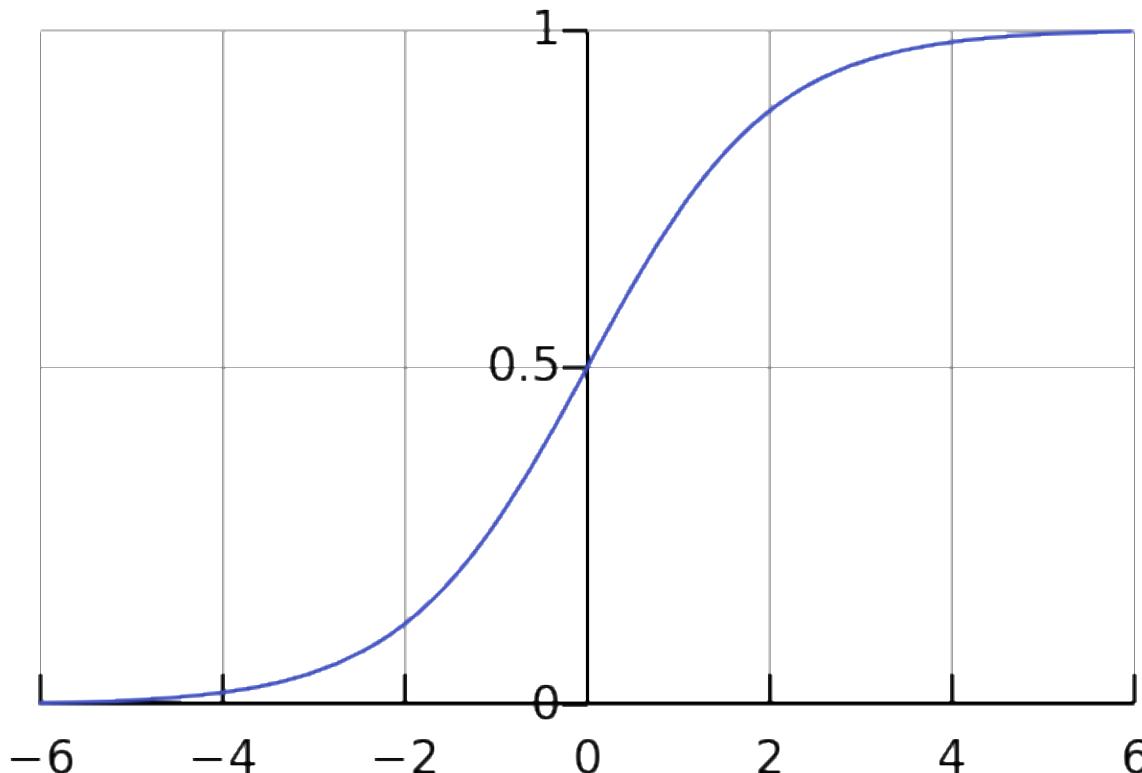
- Pri logistickej regresii chceme využiť lineárny model na odhad pravdepodobnosti $P(y = +|x_i)$
- Problém je, že lineárna funkcia:

$$f(x_i) = f(x_{i,1}, x_{i,2}, \dots, x_{i,m}) = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_m x_{i,m}$$

môže nadobúdať ľubovoľnú hodnotu -/+

- Pre odhad pravdepodobnosti ju musíme ohraňčiť do intervalu 0-1
 - Logistická funkcia $g(z) = \frac{1}{1+e^{-z}}$

Príklad klasifikačného modelu – logistická regresia (2)



- Odhad pravdepodobnosti pre triedu + po dosadení lineárnej funkcie do logistickej transformácie $P(y = +|x_i) = \frac{1}{1+e^{-f(x_i)}}$

Príklad logistickej regresie v R

```

inTrain <- createDataPartition(y=spambase$is_spam,p=.7,list=F)
training <- spambase[inTrain,]

# učenie všeobecného lineárneho modelu - logit rozdelenie pre logistickú regresiu
logit <- glm(is_spam~., data=training, family=binomial("logit"))
summary(logit)

```

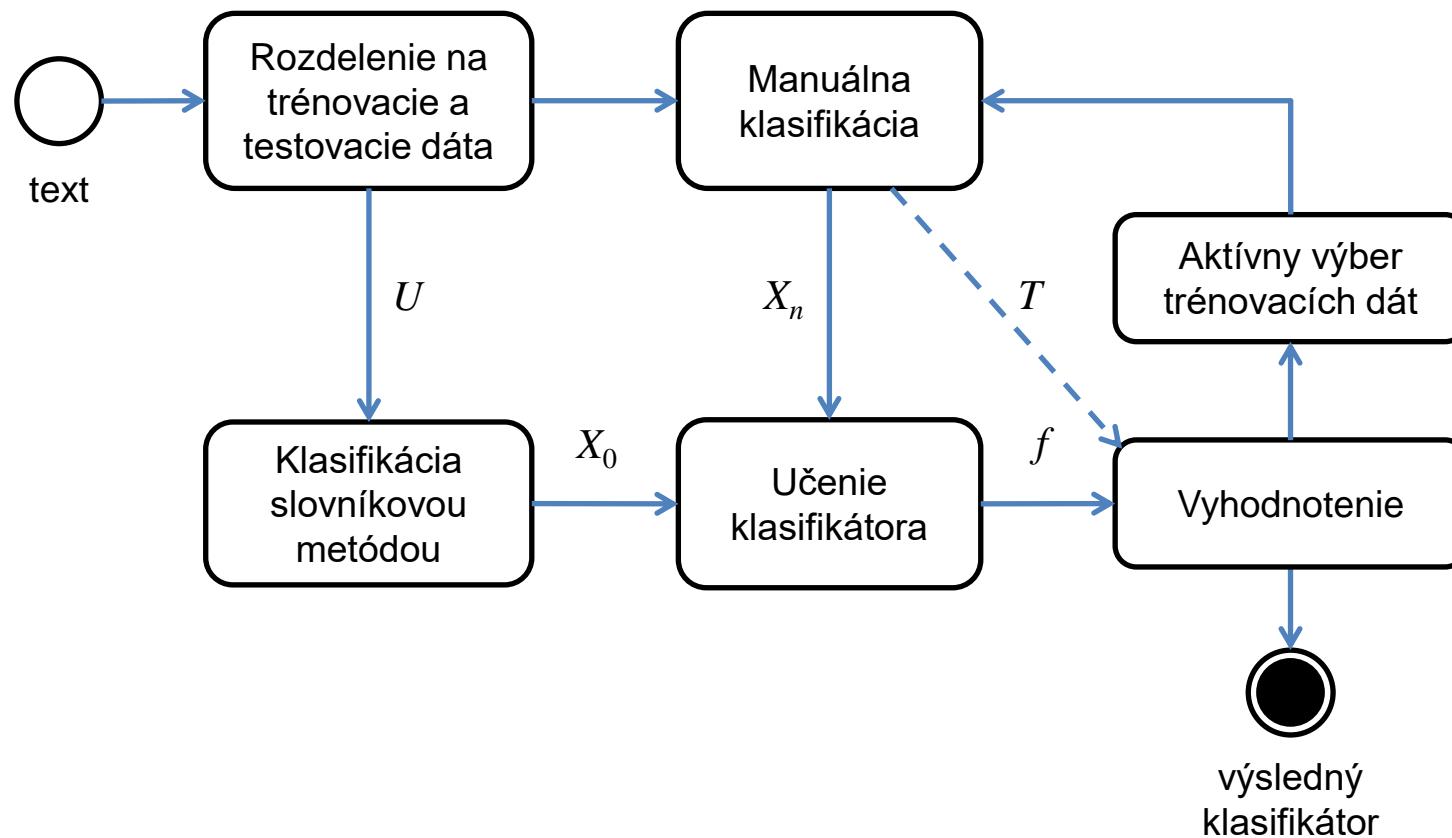
	Estimate	Std. Error	z	value	Pr(> z)
(Intercept)	-1.631e+00	1.714e-01	-9.518	< 2e-16	***
word_freq_make	-5.520e-01	3.087e-01	-1.788	0.073739	.
word_freq_address	-1.223e-01	7.843e-02	-1.559	0.118922	
word_freq_3d	2.573e+00	1.814e+00	1.418	0.156078	
word_freq_our	5.927e-01	1.209e-01	4.904	9.37e-07	***
word_freq_over	8.692e-01	2.764e-01	3.145	0.001660	**
word_freq_remove	2.227e+00	3.924e-01	5.676	1.37e-08	***
word_freq_george	-9.869e+00	2.194e+00	-4.499	6.83e-06	***
...					
char_freq_exclamation	2.334e-01	6.123e-02	3.812	0.000138	***
char_freq_dollar	5.189e+00	8.233e-01	6.304	2.91e-10	***
char_freq_pound	2.082e+00	1.456e+00	1.430	0.152705	
capital_run_length_average	-1.708e-03	2.157e-02	-0.079	0.936906	
capital_run_length_longest	9.058e-03	3.068e-03	2.952	0.003153	**
capital_run_length_total	9.262e-04	2.987e-04	3.101	0.001931	**



Využitie semikontrolovaného a aktívneho učenia (1)

- Kontrolované metódy vyžadujú manuálne klasifikované dáta – veľmi prácne a nákladné
- Na druhej strane, je pomerne ľahké získať neklasifikované dáta
- Pri **semikontrolovanom učení**, máme veľmi malú trénovaciu množinu klasifikovaných príkladov X a veľa neklasifikovaných dát U
 - Zhlukovanie spojenej množiny a zaradenie neklasifikovaných textov podľa najčastejšie s vyskytujúcej triede v zhluku
- Pri **aktívnom učení**, naučíme počiatočný model na X a z U vyberieme príklady, ktoré model klasifikuje s najmenšou istotou klasifikácie, tie potom manuálne klasifikujeme a pridáme do X a pokračuje v učení
 - Cieľom je dosiahnuť čo najlepšiu presnosť s čo najmenším počtom manuálne klasifikovaných príkladov

Využitie semikontrolovaného a aktívneho učenia (2)



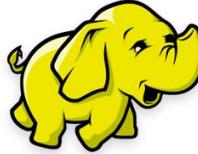
Klasifikácia emócií

- Cieľom je rozpoznať emócie vyjadrené v teste
- Klasifikačná úloha, ktorá zaradí text do preddefinovaných kategórií podľa rozdelenia emócií
- Základné rozdelenie podľa Ekmana: hnev, znechutenie, strach, šťastie/veselosť, smútok a prekvapenie
- Existuje aj viacero vektorových modelov - odhadujú sa spojité premenné v rôznych dimenziách, ktoré vyjadrujú napr. polaritu, intenzitu, pasívnosť/aktívnosť

Spracovanie veľkých dát

- Disková kapacita + operačná pamäť + výpočtový výkon – vertikálne škálovanie jedného servera
- Pre veľké dáta musí byť spracovanie rozdelené na viacero serverov – **horizontálne škálovanie**
- Základné technológie:
 - Distribuované súborové systémy
 - Distribuované databázy (SQL/NoSQL)
 - Rámce pre distribuované dávkové spracovanie
 - Rámce pre distribuované spracovanie dátových prúdov

Apache Hadoop + Spark

- Apache Hadoop 
 - YARN – rozvrhovanie výpočtových úloh na jednotlivé servery
 - HDFS – distribuovaný súborový systém
 - Hive – SQL databáza
- Apache Spark 
 - Rámec pre distribuované dávkové a prúdové výpočty + knižnica distribuovaných algoritmov strojového učenia + knižnica grafových algoritmov

Apache Spark v R

```
# inicializácia Spark prostredia
sc <- sparkR.init()
sqlContext <- sparkRSQl.init(sc)
# načítanie lokálnych dát, tweets už je distribuovaná množina dát
tweets <- read.df(sqlContext, "./examples/tweets.json", "json")

# odfiltrujeme najobľúbenejšie tweety
favorite_tweets <- filter(tweets, tweet$favorite_count > 10)
# spočítame, koľko krát sa prepošali tweety pre každý tag (klúčové slovo)
tags_followers_counts <- summarize(groupBy(tweets, tweets$tag), count=
    n(tweets$followers_count))
# zoradíme tagy podľa počtov a zobrazíme najviac preposielané
head(arrange(tags_followers_counts, desc(tags_followers_counts$count)))

# model sa buduje podobne ako na lokálnych dátach
logit <- glm(favorited~, data=tweets, family=binomial("logit"))
predictions <- predict(logit, newData=tweets)
```



Podniková analytika

Preskriptívna analytika I

Obsah

- Preskriptívna analytika, základné metódy a prístupy
- Modelovo orientované rozhodovanie
 - Optimalizácia
 - Multikriteriálne rozhodovanie
- Modelovanie a analýza
 - Heuristické prehľadávanie
 - Simulácia
- Automatické systémy DSS, Expertné systémy
- Systémy manažmentu znalostí

Čo je preskriptívna analýza – príklad 1

- Vid'. Pr.1 – Rozhodnutie o znížení cien v maloobchode
- Väčšina predajcov sa snaží sezónne vyprázdníť sklady redukciou cien
- Otázka: **Kedy a ako (o koľko) redukovať ceny ?**
- Analytické prostriedky pre tento typ úlohy
 - Deskriptívna analytika: prehľadanie historických dát pre podobné produkty (ceny, počty predaných produktov, reklama, ...)
 - Prediktívna analytika: model predikcie predajov na základe navrhnutej ceny
 - **Preskriptívna analytika: nájdenie najlepšej kombinácie nastavenia ceny a použitej reklamy pre maximalizáciu tržieb**

Čo je preskriptívna analýza – príklad 2

- Spoločnosť vlastníaca hotely a kasína
- Používa deskriptívnu analytiku na:
 - Popis vyťaženosť hotelov
 - Prehľad aktivít v kasínach
 - Náhľad na odpovedajúce výsledky z pohľadu ziskovosti
- Používa prediktívnu analytiku na:
 - Predikciu požiadaviek na obsadenie izieb
 - Segmentáciu zákazníkov podľa ich hráčskych aktivít (v kasíne)
- Používa preskriptívne modely (s cieľom optimalizácie zisku) na:
 - Nastavenie cien izieb
 - Alokáciu obsadenia izieb
 - Ponúkanie výhod a odmien zákazníkom podľa ich zaradenia do špecifického segmentu

Preskriptívna analytika

- Preskriptívna (normatívna / rozhodovacia) analytika
- Čo sa aktuálne deje a **ktoré rozhodnutia pre budúcnosť prijať aby sme dosiahli najlepšiu výkonnosť ?**
- Preskriptívna analytika
 - je (zvyčajne) modelovo orientovaná (nemusí ísť vždy o matematický model) => vytvára modely a riešenia pre rozhodnutia o budúcnosti s cieľom zistiť (a zabezpečiť) ich maximálnu možnú efektívnosť
 - má za cieľ dodať rozhodnutie alebo odporúčanie pre špecifickú akciu, pričom tak môže byť urobené v podobe reportu alebo automaticky systémom rozhodovania
 - zahŕňa veľmi rôznorodé postupy = celé oblasti metód, ktoré sa tu využívajú: operačný výskum, optimalizačné metódy, multikriteriálne rozhodovanie, simulácie, expertné systémy, systémy manažmentu znalostí, ...

Základné metódy a prístupy

- Základom je podporiť rozhodovací proces (vid'. Predn. č.1) v dosiahnutí optimálneho výsledku (v rámci DSS)
- Prostriedkom je modelovanie problému rôznymi prístupmi:
 - Modelovo orientované rozhodovanie (tzv. analytické modelovanie)
 - Matematické modelovanie
 - Matematická optimalizácia – lineárne/celočíselné programovanie, dynamické programovanie
 - Hľadanie viacerých cieľov, Analýza citlivosti, What-If analýza
 - Multikriteriálne rozhodovanie
 - Modelovanie a analýza komplexných problémov
 - Prehľadávanie a heuristiky
 - Simulácie
 - Automatické systémy rozhodovania a expertné systémy
 - Systémy pre podporu manažmentu znalostí
 - Kolaboratívne systémy
 - ...

Kategórie a typy modelov v DSS

- Konštruované modely (a aj prístupy tvorby) môžeme deliť na
 - Statické
 - Dynamické
- Pričom tieto môžu byť konštruované v prostredí s
 - Určitostou
 - Neurčitostou
 - Rizikom
- Kategórie modelov a rôzne techniky reprezentácie a riešenia => prehľad v tabuľke

Prehľad kategórií modelov v DSS

Kategória	Proces a Cieľ	Techniky
Optimalizácia problémov s niekoľkými alternatívami	Hľadanie najlepšieho riešenia, malé množstvo alternatív	Rozhodovacie tabuľky, stromy rozhodnutí, AHP (Analytic Hierarchy Process)
Algoritmická optimalizácia	Hľadanie najlepšieho riešenia, veľký počet možností, algoritmický proces	Operačný výskum: lineárne, celočíselné, dynamické programovanie, sieťové modely
Analytická (jednokroková) optimalizácia	Hľadanie najlepšieho riešenia v jednom kroku pomocou vzorca	Špecifické modely operačného výskumu, modely zásob
Simulácia	Experimentálne hľadanie dostatočne dobrého riešenia alebo najlepšieho medzi overenými možnosťami	Rôzne typy simulácií
Heuristiky	Hľadanie dostatočne dobrého riešenia pomocou prehľadávania a pravidiel	Heuristické programovanie, expertné systémy
Predikčné modely	Predikcia budúcnosti pre dané scenáre	Predpovedné modely, Markovovská analýza
Iné modely	Riešenie what-if prípadov pomocou vzorca	Finančné modelovanie, operačný výskum: teória hromadnej obsluhy

Určitosť, neurčitosť a riziko

- V rámci procesu evaluácie a porovnania alternatív je potrebné predikovať budúce výstupy pre každú alternatívu (vid'. RP)
- Rozhodovacie situácie sa klasifikujú podľa toho čo rozhodovateľ vie o predpovedaných výsledkoch (alebo nakoľko im verí), klasicky sa používa rozdelenie do 3 kategórií (od úplnej znalosti až úplné neznalosť) => Určitosť, Riziko a Neurčitosť
- Rozhodovanie v prostredí s určitosťou
 - Predpokladom je úplná znalosť výstupov pre každú akciu
 - Často to môže byť len optimistický predpoklad ktorý zjednodušuje zvolený model rozhodovania
- Rozhodovanie v prostredí s neurčitosťou
 - Existuje viacero možných výstupov pre akcie, avšak nepoznáme pravdepodobnosť ich výskytov
 - Riešenie týchto problémov je najzložitejšie (často sa mu snažíme vyhnúť – zjednodušením, aspoň posunom na úroveň rizika)
- Rozhodovanie v prostredí s rizikom (pravdepodobnostné/stochasticné)
 - Jednoduchší prípad neurčitosti, keď poznáme alebo vieme odhadnúť pravdepodobnosť výskytov možných výstupov akcií
 - Analýza rizika = kalkulácia rizika asociovaného z rôznymi alternatívami = výpočet očakávaných hodnôt a výber najlepšej možnosti

Štruktúra matematických modelov pre RP

- Komponenty kvantitatívneho modelu
 - Matematické vzťahy (prepájajú premenné)
 - Premenné
 - Rozhodovacie premenné
 - alternatívy investovania, množstvo peňazí investície / marketingu, miesto pre reklamu, využitie zdrojov, rozvrh rozvozu / práce, ...
 - Nekontrolované premenné a parametre
 - Inflácia, kapacita strojov, úroveň daní, regulácie, technológie, cena materiálu, ...
 - Výstupné – reflektujú efektívnosť systému
 - Zisk, návratnosť investície, riziko, podiel trhu, náklady, spokojnosť zákazníka, úroveň kvality, chybovosť, ...
 - Pomocné premenné
- Príklad: Jednoduchý matematický finančný model
 $Z = V - N$, kde Z=zisk, V=výnos, N=náklady

Príklad: statický / dynamický model

A	D	E	F	G	H	I	J	K	L
1									
2	Jednoduchý statický úverový model								
3									
4	Velkosť úveru		150 000,00 €						
5	Úrok (ročný)		8,00%						
6	Počet rokov		30				$=12*G6$		
7									
8	Počet mesiacov		360				$=G5/12$		
9	Úrok / mesiac		0,67%						
10									
11	Mesačná splátka úveru		1 100,65 €				$=-PMT(G9;G8;G4)$		
12									
13									
14				$= \$G\12			$=F20+G20$		
15									
16	Dynamický model (s priplatením 100 Eur každý mesiac)								
17									
18	Mesiac	Normálna platba	Priplatie	Platba celkovo	Dílžna suma				
19	0				150 000,00 €				
20	1	1 100,65 €	100,00 €	1 200,65 €	149 799,35 €		$=I19*(1+\$G\$9)-H20$		
21	2	1 100,65 €	100,00 €	1 200,65 €	149 597,37 €				
22	3	1 100,65 €	100,00 €	1 200,65 €	149 394,04 €				
23	4	1 100,65 €	100,00 €	1 200,65 €	149 189,35 €				
24	5	1 100,65 €	100,00 €	1 200,65 €	148 983,30 €				
25								
26								
27	269	1 100,65 €	100,00 €	1 200,65 €	303,62 €				
28	270	1 100,65 €	100,00 €	1 200,65 €	-895,00 €				

Matematická programová optimalizácia

- Matematické programovanie je množina nástrojov pre algoritmickú optimalizáciu v prípadoch, kedy je potrebné alokovať obmedzené zdroje medzi „súťažiace“ aktivity
- Typické prístupy:
 - Lineárne / celočíselné programovanie
 - Dynamické programovanie
 - Nelineárne programovanie
 - Modely pre plánovanie a rozvrhovanie
 - Priradzovacie a distribučné modely
 - ...

Optimalizácia - lineárne programovanie

- Optimalizačné úlohy sú často definované ako:
 - súbor obmedzujúcich podmienok na výber kandidátov riešení (tzv. constraints)
 - optimalizačná (kriteriálna) funkcia - KF
 - Používa sa pre výber optimálneho kandidáta (hľadáme takého čo maximalizuje alebo minimalizuje KF)
- Lineárne programovanie (linear programming)
 - Vstupom sú
 - premenné x_1, x_2, \dots, x_n – reálne premenné
 - Súbor ohraničení – m lineárnych (ne)rovníc o premenných => generujú množinu prípustných riešení (MPR) = body v n-rozmernom priestore, ktoré vyhovujú ohraničeniam
 - KF = lineárna kombinácia premenných + fakt o úlohe optimalizácie (min alebo max)
$$KF: \max(c_1x_1 + c_2x_2 + \dots + c_nx_n)$$
 - Výstup: bod/body MPR, pre ktoré je KF optimálna

Lineárne programovanie (2)

- Základné vlastnosti / predpoklady LP
 - Aditívnosť – celkové sumy hodnôt (ziskov, surovín, času, nákladov, ...) sú súčty dielčích hodnôt
 - Proporcionalita – priama úmera medzi spotrebou a hodnotou / veľkosťou produkcie
 - Deliteľnosť – čísla môžu byť zlomkové
 - (Nezápornosť) – hodnoty premenných kladné (nie vždy nutná podmienka)
- Riešenie
 - Rôzne algoritmy, najčastejšie SIMPLEX-ová metóda
 - Prakticky: použijeme nejaký tzv. LP Solver = knižnica / balík / program, do ktorého vložíme zadanie problému (premenné + kriteriálna funkcia + ohraničenia) a dostaneme riešenie úlohy (ak existuje)

Príklad – lineárne programovanie

- Máme problém definovaný nasledovne:

$$KF : \max(143x + 60y)$$

$$120x + 210y \leq 15000$$

$$110x + 30y \leq 4000$$

$$x + y \leq 75$$

$$x \geq 0$$

$$y \geq 0$$



Ohraničenia =
Constraints
(posledné 2 sú
klasické tzv.
default
ohraničenia o
nezápornosti
vstupov)

- Existuje viacero balíkov pre riešenie takýchto úloh
 - IpSolve, IpSolveAPI, Rglpk, linprog, limSolve, ...
- Použijeme napr. balík IpSolveAPI
 - > `install.packages("IpSolveAPI")`
 - > `library(IpSolveAPI)`

Príklad – lineárne programovanie (2)

- Reprezentácia problému v IpSolveAPI
 - > `lpmodel <- make.lp(0, 2) # prázdny LP solver s 2 premennymi`
 - > `lp.control(lpmodel, sense="max") # maximalizacia`
 - > `set.objfn(lpmodel, c(143, 60)) # definícia KF (v anglictine casto objective function)`
 - > `add.constraint(lpmodel, c(120, 210), "<=", 15000)`
 - > `add.constraint(lpmodel, c(110, 30), "<=", 4000)`
 - > `add.constraint(lpmodel, c(1, 1), "<=", 75)`

- Ako vyzerá model ?

> `lpmodel`

Default ohraničenia ($x, y \geq 0$) sú pridané automaticky (Lower - Upper)
... je možné ich zmeniť

Model name:			
Maximize		c1	c2
R1		143	60
R2		120	210
R3		110	30
Kind		1	1
Type		Std	Std
Upper		Real	Real
Lower		Inf	Inf
		0	0

Príklad – lineárne programovanie (3)

- Riešenie – použitie solve funkcie

```
> solve(lpmodel)
```

```
[1] 0
```

```
> get.objective(lpmodel) # dosiahnuta hodnota KF
```

```
[1] 6315.625
```

```
> get.variables(lpmodel) # hodnoty premennych pre optimum
```

```
[1] 21.875 53.125
```

- Interpretácia riešenia

- Matematicky: Daný problém má optimálne riešenie, konkrétnie v bode $[21.875, 53.125]$ s hodnotou KF (ktorá je maximálna) 6315.625
- Vstupný model (premenné + ohraničenia + KF) vzniká z reálnej úlohy
 - na konci je potrebné interpretovať výsledok voči reálnej úlohe
 - V tomto prípade => farmár sa mal optimálne rozhodnúť o zasadení množstva pšenice (x) a jačmeňa (y), pričom bol obmedzený podmienkami a chcel maximalizovať zisk (KF) => odpoveď na reálnu úlohu teda je ... Farmár by mal vysadiť 21.875 árov pšenice a 53.125 árov jačmeňa (1 ár = 100m^2), aby tak dosiahol maximálny zisk 6315.625 dolárov

Ak neplatí proporcionalita ?

- Ak neplatí pre LP proporcionalita – t.j. napríklad KF je $X_1^2 + X_2^3 \Rightarrow$ nelineárne programovanie
- Zložitosť riešenia je vo všeobecnosti omnoho vyššia (nemusí existovať možnosť univerzálne nájsť úplne = optimálne riešenie \Rightarrow to vedie na nájdenie suboptimálnych riešení)
- Existujú rôzne stratégie riešenia, často len numerické aproximácie (podľa zložitosti úlohy)
 - Rôzne gradientové metódy, Lagrangeove multiplikátory, Kuhn-Tuckerove metódy, ...
 - Prípadne sa zvolí iný prístup: heuristika, simulácie

Ďalšie typy matematického programovania

- Ak porušíme deliteľnosť = úlohy majú mať celočíselné premenné => celočíselné programovanie = integer programming (IP)
 - V špeciálnom prípade ak sú premenné binárne, tak hovoríme o bivalentnom alebo binárnom programovaní
- Ak kombinujeme celočíselné a reálne hodnoty = zmiešané úlohy vedú na tzv. zmiešané celočíselné programovanie = mixed integer programming
- Problém úloh celočíselného programovania je v tom, že MPR nie je súvislá množina, len časť bodov – klasické algoritmy LP môžu mať problém nájsť optimálne riešenie
- Algoritmy
 - Úplné – napr. metóda vetvenia a medzí, zaručuje nájdenie riešenia = úplne prehľadávanie (avšak v nepolynomiálnom čase)
 - Približné – rôzne (meta)heuristické prístupy – čiastočné prehľadávanie, genetické algoritmy, simulované žíhanie, ...

Celočíselné programovanie

- Príklad – priradzovací problém (assignment)
 - Matica nákladov priradenia i-tého objektu do j-tého miesta – COST matica – známa
 - Cieľ: rozhodnúť o matici priradení $X \Rightarrow X$ má 1 tam, kde platí že i-tý objekt ide do j-tého miesta, inak 0 (ak i-tý objekt do j-tého miesta nedávame) \Rightarrow to všetko za predpokladu minimálnych nákladov
 - Použijeme napr. **IpSolve** balík (má priamo ľahko aplikovateľnú lp.assign funkciu pre tento problém)

```
> library(lpSolve)
```

```
> assign.costs <- matrix(c(7, 7, 3, 2, 2, 7, 7, 2, 1, 9, 8, 2, 7, 2, 8, 10),  
4, 4)
```

```
> lp.assign(assign.costs)
```

```
> lp.assign(assign.costs)$solution
```

```
> lp.assign(assign.costs)  
Success: the objective function is 8
```

```
> assign.costs
```

	[,1]	[,2]	[,3]	[,4]
[1,]	7	2	1	7
[2,]	7	7	9	2
[3,]	3	7	8	8
[4,]	2	2	2	10

```
> lp.assign(assign.costs)$solution
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0	0	1	0
[2,]	0	0	0	1
[3,]	1	0	0	0
[4,]	0	1	0	0

Čo to znamená použiť heuristiku ?

- Existuje veľké množstvo techník, ale základný princíp je rovnaký
- Heuristika
 - Prehľadáva priestor riešení
 - Vo všeobecnosti tu neexistujú modely ako LP, IP
- Postup
 - Naštartujeme prehľadávanie definovaním (alebo výberom) jedného alebo viacerých platných riešení
 - Následne zlepšujeme cieľ prehľadávania (zvolené kritérium) výberom nových / úpravou existujúcich riešení (systematicky tak, aby sa kritérium zlepšovalo)
 - Zastavíme prehľadávanie keď uplynie nami stanovený čas, počet krokov, alebo dosiahneme požadovanú hodnotu kritéria (kvalitu nájdeného riešenia)
- Výsledok: najlepšie nájdené riešenie z nášho procesu (berieme ho za dostatočne dobré riešenie v danom momente) = ide o suboptimálne riešenie ... Môže to byť aj optimálne riešenie, ale nemáme to zaručené

Hľadanie (sledovanie) viacerých cieľov

- V praxi sledujú manažéri viacero cieľov pri optimalizácii rozhodnutí podniku, pričom tieto môžu byť čiastočne konfliktné
 - Okrem zisku, môže firma sledovať aj ciele ako rast firmy, rozvoj produktov a zamestnancov, ...
 - Nakoľko väčšina metód sa zameriava na 1 cieľ => zvyčajne je potrebné transformovať viac cieľov na problém z jednou mierou => napr. vytvoriť LP/IP problém z ohraničeniami a jednou kombinovanou KF
- Sledovanie viacerých cieľov samozrejme takto naráža na rôzne problémy:
 - Ťažkosť explicitného vyjadrenia cieľov podniku, tieto sa navyše môžu meniť v čase, dôraz na jednotlivé ciele je rôzny na rôznych úrovniach podniku, ciele sa menia pri zmenách v podniku a prostredí, vzťahy medzi cieľmi sa ťažko kvantifikujú, rozhoduje sa zvyčajne v skupinách ľudí s rôznou agendou a prioritami, ...
- pričom najpoužívanejšie metódy riešenia sú
 - teória úžitku, cieľovo orientované programovanie, vyjadrenie cieľov ako ohraničení (a použitie LP/IP), bodovací systém

Analýza citlivosti

- Vytvorený model závisí na vstupných dátach
- Analýza citlivosti sa snaží odhadnúť dopady ktoré majú zmeny vo vstupných dátach a parametroch na výstupné premenné
- Takáto analýza zvyšuje flexibilitu a adaptáciu riešenia na meniace sa podmienky, lepšie pochopenie problému a zvyšuje spoľahlivosť modelu
- Analýza citlivosti testuje vzťahy ako:
 - Dopad zmien nekontrolovaných (externých) premenných a parametrov na výstupy
 - Dopad zmien rozhodovacích premenných na výstupy
 - Efekt neurčitosti na odhad externých premenných
 - Efekty rôznych závislostí a interakcií medzi premennými
 - Robustnosť rozhodnutí v rámci meniacich sa podmienok
- Analýza citlivosti sa používa pre:
 - Revíziu modelov pre elimináciu zvýšenej citlivosti
 - Pridanie detailov o citlivých premenných a scenároch
 - Získavanie lepších odhadov citlivých externých premenných
 - Úpravu reálneho systému pre redukciu aktuálnej citlivosti
- Základné typy analýzy citlivosti
 - Automatická – v rámci LP sa dá ľahko zistiť aké zmeny vstupov ešte (výrazne) nezmenia výsledok
 - Pokus-omyl – What-If analýza a spätné hľadanie cieľa (goal seeking)

Príklad: What-If analýza

- What-If analýza je štruktúrovaná ako otázka „Čo sa stane s riešením ak sa vstupná premenná, predpoklad alebo parameter zmenia ?“
 - Aký bude podiel trhu ak zvýšime rozpočet na reklamu o 5% ?
 - Existujú špeciálne softvéry na takúto analýzu (pre jej ľahšiu implementáciu), prípadne môžeme použiť štatistický alebo tabuľkový softvér (napr. Excel)
- Príklad:

A	B	C	D	E	F
1 What If analýza					
2					
3 Príjem za položku	1,20 €				
4 Náklady na položku	0,60 €				
5 Fixné náklady	30 €				
6 Počiatočný predaj	120				
7 Rast predajov na štvrtok	0,04				
8					
9 Ročný čistý zisk	178 €				
10					
11					
12	1.štvrťrok	2.štvrťrok	3.štvrťrok	4.štvrťrok	Ročne
13 Predaje	120	125	130	135	510
14 Príjmy	144 €	150 €	156 €	162 €	611 €
15 Premenlivé náklady	72 €	75 €	78 €	81 €	306 €
16 Fixné náklady	30 €	31 €	32 €	34 €	127 €
17 Čistý zisk	42 €	44 €	45 €	47 €	178 €
18					

Vstupné premenné
a parametre

Výstupná hodnota

	Počiatočné predaje			
	100	110	120	
Rast	0,02	124 €	148 €	173 €
	0,04	127 €	153 €	178 €
	0,06	131 €	157 €	184 €

Rozhodovacie tabuľky a stromy rozhodnutí

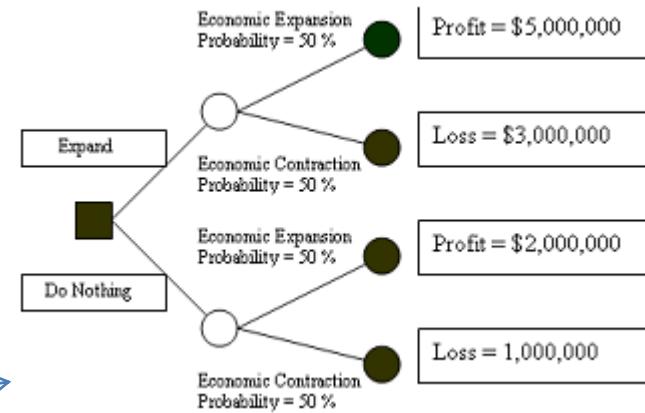
- V prípade malého počtu alternatív sa používajú tabuľky (rozhodovacie tabuľky) alebo grafy (stromy rozhodnutí)
- Rozhodovacia tabuľka
 - organizuje informáciu v tabuľkovom prehľade
 - Ak máme viac cieľov => viedie na multikriteriálne rozhodovanie
 - Otázky ktoré zodpovedáme v jednotlivých častiach tabuľky
 - 1. Aké sú rozhodovacie podmienky riešenia problému ?
 - 2. Ktoré činnosti sú požadované pre riešenie problému ?
 - 3. Aké sú možné kombinácie podmienok ?
 - 4. Ktoré činnosti je potrebné vykonať pri pôsobení jednotlivých kombinácií podmienok ?

Záhlavie tabuľky (názov problému)	Záhlavie kombinácií (pravidiel riešenia)
1. Kvadrant (zoznam) podmienok	3. Kvadrant stavov (kombinácií) podmienok
2. Kvadrant (zoznam) činností	4. Kvadrant voľby (kombinácií) činností

Rozhodovacie tabuľky a stromy rozhodnutí (2)

- Príklad rozhodovacej tabuľky: prechod križovatky

PRECHOD KRIŽOVATKY	Kombinácie				
	1	2	3	4	5
Riadená križovatka	A	A	N	N	N
Svieti zelená	A	N	-	-	-
Voľno v smere premávky	-	-	A	N	N
Prichádzajúce vozidlo v dostatočnej vzdialosti	-	-	-	A	N
PREJST	X	-	X	X	-
POČKAŤ	-	X	-	-	X



- Strom rozhodnutí
 - Kombinácia rozhodnutí (alternatív) a nekontrolovaných premenných prostredia je modelovaná ako strom postupného rozhodovania, kde listové uzly sú potom výstupné hodnoty

Podniková analytika

Preskriptívna analytika II

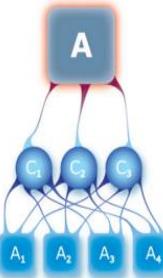
Obsah

- Preskriptívna analytika, základné metódy a prístupy
- Modelovo orientované rozhodovanie
 - Optimalizácia
 - Multikriteriálne rozhodovanie
- Modelovanie a analýza
 - Heuristické prehľadávanie
 - Simulácia
- Automatické systémy DSS, Expertné systémy
- Systémy manažmentu znalostí

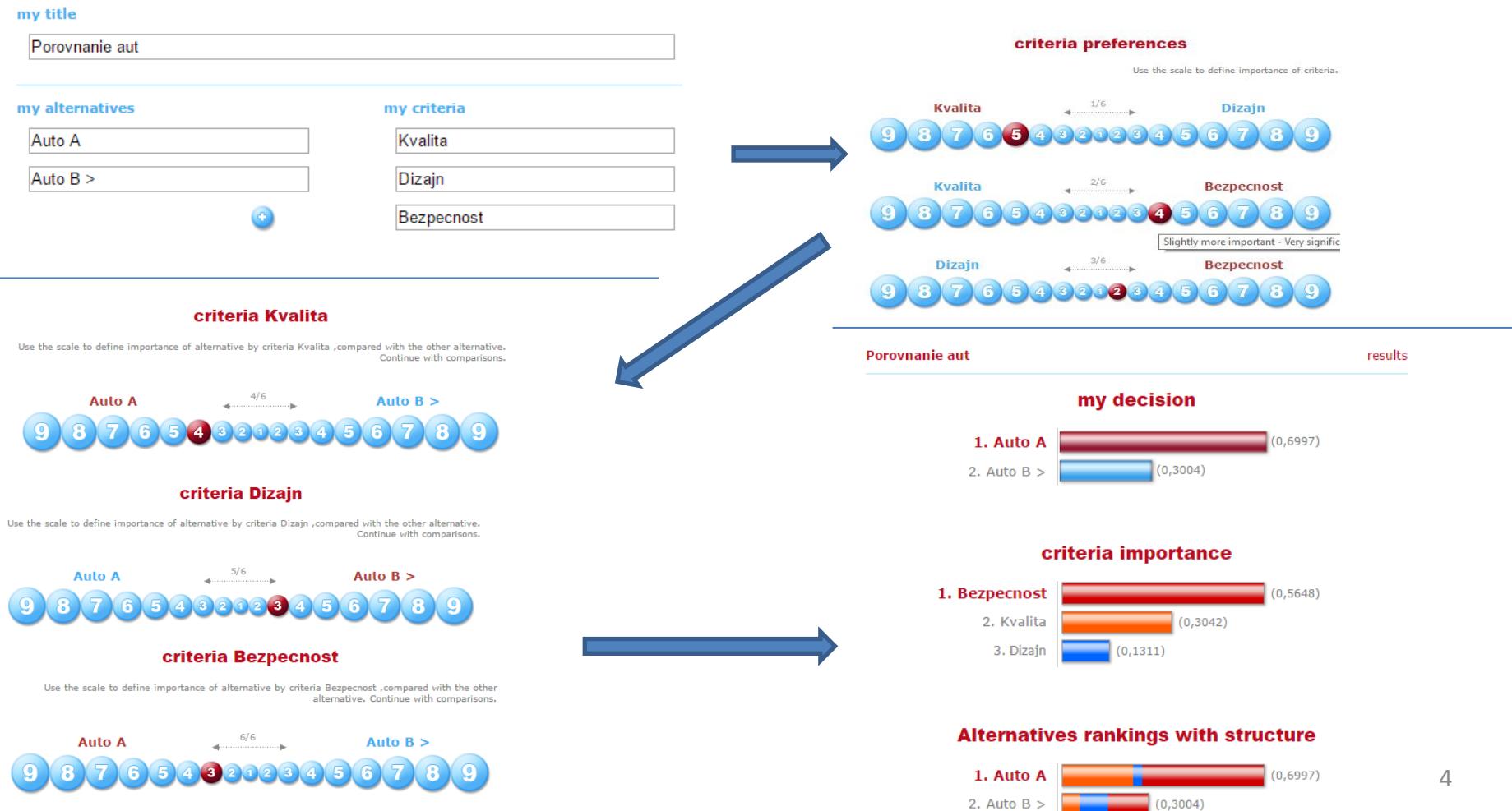
Multikriteriálne rozhodovanie + AHP

- Ak máme viac kritérií, často sa týmto priradia váhy a rozhodnutia sú ohodnotené vzhľadom k nim
 - napr. spočítame váhy za jednotlivé kritéria = pre každú alternatívu vyhodnotíme „súčet“ váh cez všetky kritériá a vyberieme najlepšiu alternatívu (tzv. rozhodovacia matica)
- Problém: ako určiť váhy ? => napríklad metódou ohodnocovania AHP (Analytic Hierarchy Process)
 - Máme alternatívy pre výber a kritériá
 - Párovo pre každú kombináciu kritérií zvolíme o koľko viac uprednostňujeme jedno z daných kritérií (a v nich po subkritériách, ak tieto existujú)
 - Následne porovnáme jednotlivé alternatívy medzi sebou po jednotlivých kritériách (resp. subkritériách)
 - Pre porovnanie sa používa ranking 1-9 (na stranu kritéria alebo alternatívy, ak sú rovnaké = 1)
 - AHP určí váhy jednotlivých kritérií => tieto následne použijeme na optimalizáciu => AHP určí aj odporúčanú alternatívu

Príklad – AHP



- Príklad: <http://www.123ahp.com/>
- Zjednodušená verzia Cars - máme 2 autá (A,B) a tri kritéria (Kvalita, Dizajn, Bezpečnosť)



Prehľadávacie metódy a heuristiky

- Hľadanie riešenia prehľadávaním priestoru alternatív
- Prehľadávacie prístupy – rozdelenie
 - Optimalizácia (analytická)
 - Generovanie vylepšených riešení (alebo priame získanie najlepšieho riešenia) => STOP ak nie je možné zlepšenie => Optimálne riešenie (najlepšie - optimum)
 - Slepé prehľadávanie
 - Úplné prehľadávanie: Postup – všetky možnosti sú preskúmané => STOP ak sme preskúmali všetko sú všetky => Optimálne riešenie (najlepšie - optimum)
 - Parciálne (neúplné) prehľadávanie (suboptimalizácia): Postup – testujeme iba časť alternatív => systematicky vyradujeme menej vhodné alternatívy riešenia => STOP ak máme dostatočne dobré riešenie => Najlepšie riešenie medzi overenými alternatívmi (suboptimum)
 - Heuristiky (heuristicke prehľadávanie)
 - Uvažujeme iba „sľubné“ riešenia (riadime sa odhadom = heuristickou informáciou = ako dobré zrejme riešenie je) => STOP ak máme dostatočne dobré riešenie => Dostatočne dobré riešenie (suboptimum)

Simulácie

- Simulácia predstavuje techniku experimentálneho modelovania reality
 - Často sú problémy natoľko komplexné a zahŕňajú „náhodnosť“, že klasická optimalizácia sa nedá použiť
 - Simulácia nie je striktne modelom (ten reprezentuje realitu), simulácia realitu skôr imituje => predstavuje skôr deskriptívny ako normatívny prístup
 - Simulácia popisuje alebo predikuje charakteristiky systému za rôznych podmienok => uskutočnením mnohých simulácií vieme estimovať (hodnotu a varianciu) efektu analyzovaných akcií
- Výhody
 - Deskriptívnosť, komprimácia časovej zložky (simulujeme v krátkom čase výpočtu cez dlhé obdobia), manažér vie otestovať veľa možností bez rizika, v rozumnom čase a za menej peňazí
 - Simulačný model je vytvorený z pohľadu manažéra, vytvára sa priamo na daný problém (manažér nepotrebuje nejakú všeobecnú znalosť)
 - Simulácia je často pre komplexný problém jedinou podporou dostupnou v DSS, snaha je podchytíť reálnu komplexitu (zjednodušenia nie sú potrebné)
 - Simulácia produkuje automaticky mnohé dôležité výkonnostné metriky podniku
- Nevýhody
 - Optimálne riešenie nie je garantované (avšak zvyčajne sa podarí nájsť dostatočne dobré riešenie)
 - Konštrukcia samotného simulačného modelu je pomalá a nákladná
 - Riešenia z jednej simulácie nie sú prenositelné na iný problém
 - Simulačný softvér niekedy vyžaduje špeciálne znalosti (je potrebné komunikovať s IT špecialistom)

Metodológia simulácie

- 1. Definovanie problému – klasifikácia problému, špecifikácia potreby simulácie, definovanie ohraničení, prostredia, parametrov pre pochopenie problému
- 2. Konštrukcia simulačného modelu – určenie premenných a ich vzťahov, ako aj získavania dát, často sa používa vývojový diagram
- 3. Testovanie a validácia modelu – overenie nakol'ko simulačný model popisuje študovaný systém
- 4. Návrh experimentu – určíme ako dlho chceme simulovať + riešime nastavenie 2 konfliktných cieľov (presnosť, náklady) + je dôležité identifikovať typické (priemerné prípady pre náhodné premenné), najlepšie (malé náklady, veľký zisk) a najhoršie (veľké náklady, malý zisk) scenáre => vieme tak určiť rozmedzie premenných a asistovať pri debugovaní simulačného modelu
- 5. Uskutočnenie experimentu – realizácia experimentu v softvéri
- 6. Evaluácia výsledkov – interpretácia výsledkov experimentu, štatistické výhodnotenie, analýza citlivosti, ...
- 7. Implementácia výsledkov – tak ako v iných prípadoch implementácie riešenia je potrebné zaviesť zmeny v reálnom systéme => intenzívnejšie zapojenia manažérov pri simuláciách vedie väčšinou k vyššej úspešnosti implementácie

Rôzne typy simulácie

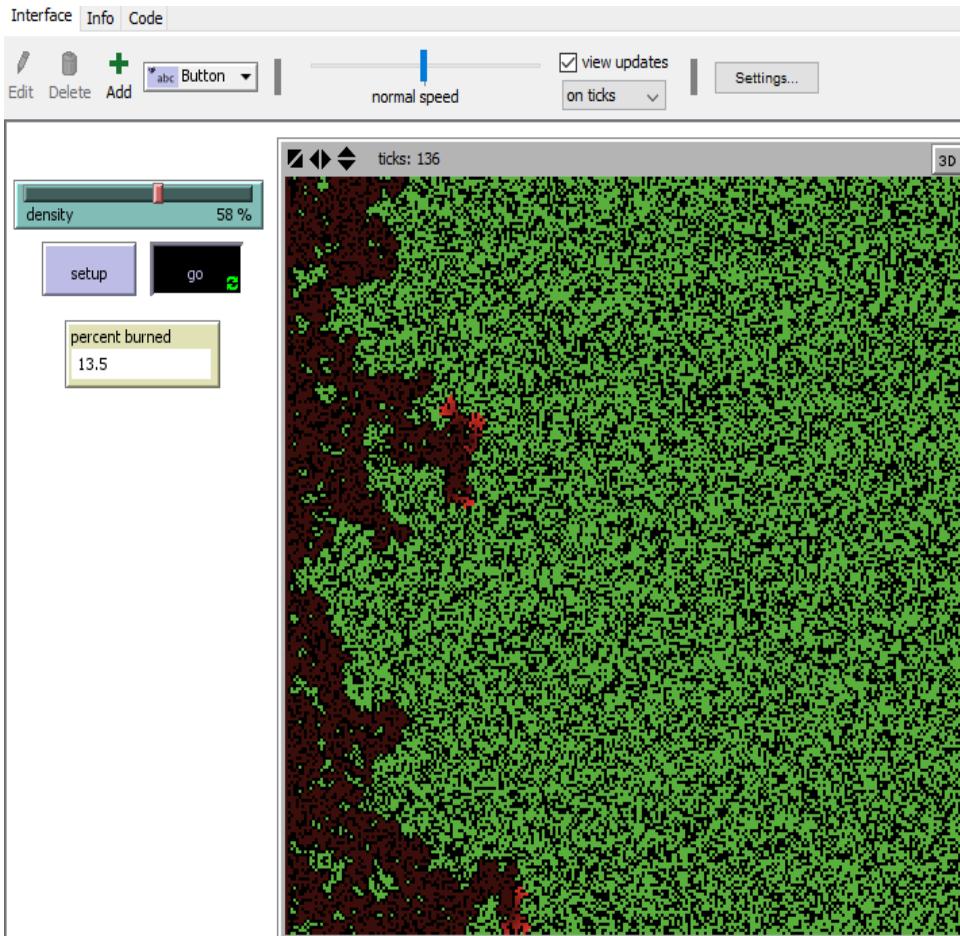
- Pravdepodobnosťná simulácia
 - Jedna alebo viac premenných sú pravdepodobnosťné
 - Môžu byť modelované diskrétnou alebo spojitou distribúciou
 - Príklad: Monte Carlo simulácie
 - Premenné modelu sú neurčité resp. sledujú určitú distribúciu => experiment kde generujeme hodnoty pre neurčité parametre a sledujeme ich dopad
 - Príklad: Diskrétne simulácie udalostí
 - Štúdium interakcií medzi entitami systému, napr. zákazníci, server (stránka obchodu), modelujeme príchod na stránku, spracovanie požiadaviek, s rôznymi časovými vlastnosťami, priemernú výkonnosť servera, ...
- Časovo závislá / nezávislá simulácia
 - Existujú rôzne prípady simulácií podľa časového hľadiska: niekedy stačí vedieť o počte za nejaké obdobie (vieme že sú 3 objednávky za deň, ale nezáleží nám na presnom momente), niekedy (fronty spracovania) potrebujeme vedieť čas udalostí presne, niekedy čas nehrá žiadnu rolu (návrh rozmiestnenia objektov)
- Vizuálna interaktívna simulácia
 - Oblúbená množina techník (a nástrojov), kde je simulácia resp. jej priebeh vizualizovaný pre lepšie pochopenie vývoja a získanie väčšej dôvery v odporúčania
- Modelovanie systémovej dynamiky – makroúrovňové simulačné modely, vhodné pre agregované modelovanie
- Agentové modelovanie

Agentové modelovanie

- ABM(Agent-Based Modeling) – simulačná technika pre komplexné rozhodovanie kde systém alebo siet' je modelovaná ako množina autonómnych jednotiek (agentov)
 - Agenti majú množinu pravidiel (akcií) o ktorých sa rozhodujú (samostatne) ... Agent = človek, časť systému, produkt, ...
 - Ide o prístup zdola-nahor => množina agentov svojim lokálnym správaním sa postupne vytvára spoločné správanie sa systému
- Nástroje:
 - Všeobecný programovací prostriedok
 - Špeciálny multiagentový simulačný softvér (platforma)
 - NetLogo (<https://ccl.northwestern.edu/netlogo/>)
 - RePast (<http://repast.sourceforge.net/>)
 - SWARM (http://www.swarm.org/wiki/Main_Page)

Príklad – Netlogo + R

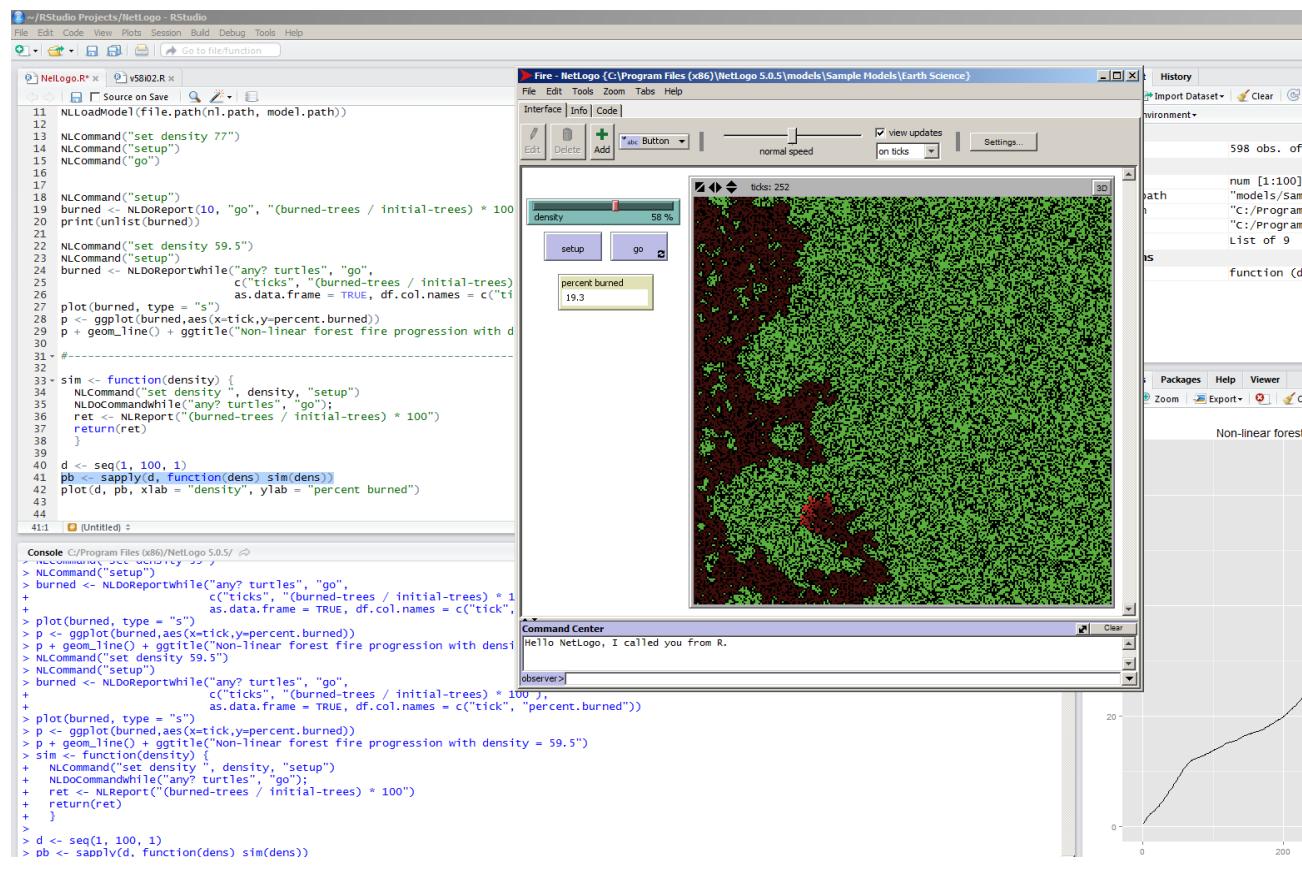
- Príklad: Lesný požiar cez NetLogo (demo: Fire)



```
globals [  
    initial-trees ;; how many trees (green patches) we started with  
    burned-trees ;; how many have burned so far  
]  
  
breed [fires fire] ;; bright red turtles -- the leading edge of the fire  
breed [embers ember] ;; turtles gradually fading from red to near black  
  
to setup  
    clear-all  
    set-default-shape turtles "square"  
    ;; make some green trees  
    ask patches with [(random-float 100) < density]  
        [ set pcolor green ]  
    ;; make a column of burning trees  
    ask patches with [pxcor = min-pxcor]  
        [ ignite ]  
    ;; set tree counts  
    set initial-trees count patches with [pcolor = green]  
    set burned-trees 0  
    reset-ticks  
end  
  
to go  
    if not any? turtles ;; either fires or embers  
        [ stop ]  
    ask fires  
        [ ask neighbors4 with [pcolor = green]  
            [ ignite ]  
            set breed embers ]  
    fade-embers  
    tick  
end  
  
;; creates the fire turtles  
to ignite ;; patch procedure  
    sprout-fires 1  
    [ set color red ]  
    set pcolor black  
    set burned-trees burned-trees + 1  
end  
  
;; achieve fading color effect for the fire as it burns  
to fade-embers  
    ask embers  
        [ set color color - 0.3 ;; make red darker  
            if color < red - 3.5 ;; are we almost at black?  
                [ set pcolor color  
                    die ] ]  
end  
  
|  
; Copyright 1997 Uri Wilensky.  
; See Info tab for full copyright and license.
```

Príklad – NetLogo + R (2)

- Existuje prepojenie – balík RNetLogo – umožňuje volať NetLogo z R, získať dáta o simulácii, spúšťať ich, ...
- <http://www.r-bloggers.com/agent-based-models-and-rnetlogo/>



- Nastavíme pripojenie na NetLogo
- Následne načítame simulačný NetLogo model (napr. Fire z knižnice modelov v Netlogu)
- môžeme spustiť simulácie (aj viacero pre rôzne parametre)
- môžeme získať dátu o ich výsledkoch a dať ich napr. do data frame
- Vizualizovať výsledky, v R s väčšou podporou ako v NetLogo

Automatické systémy rozhodovania

- ADS (Automated Decision Systems)
 - Pravidľovo orientovaný systém poskytujúci riešenie (zvyčajne v jednej doméne – finančníctvo, výroba, ...) pre špecifický často sa opakujúci manažérsky problém
 - Predstavujú vlastne automatizovaný systém pre odporúčania
 - Základným prvkom systému je množina rozhodovacích pravidiel
 - Príklad: Ponúknutú zľavu ak priemerné predaje klesnú o 10 %
 - Pravidlá sa aplikujú automaticky
 - Príklad systémov:
 - Automatický systém pre nákup leteniek
 - Vyhľadávanie ponúk izieb (booking.com) a návrh cien pre optimálny marketing
 - Úprava cien položiek podľa dostupnosti služieb a pravidiel pre ich spotrebu

Expertné systémy (ES)

- ES sú informačné systémy používajúce uloženú expertnú znalosť (expertízu) pre získanie rozhodnutí (odporúčaní) v špecifickej doméne
 - Expert = osoba majúca špeciálnu znalosť, skúsenosť, schopnosti pre riešenie problémov
 - Expertíza = rozsiahla, úlohovo-špecifická znalosť ktorú expert vlastní a poskytuje do ES
 - Vyšší level expertízy => vyššia úspešnosť rozhodnutí
 - Získava sa často tréningom, čítaním, praktickou skúsenosťou, ...
 - Často sú súčasťou experízy aj skúseností z minulých úspechov a neúspechov
 - Vlastnosti ES
 - Expertíza – experti sa líšia úrovňou expertízy, ES musí extrahovať expertnú znalosť dostatočne robustne
 - Symbolické odvodzovanie – znalosť je reprezentovaná v symbolickom jazyku a pre získavanie rozhodnutí sa používa proces odvodzovanie (spätné / dopredné reťazenie)
 - Hĺbka znalostí – znalostná báza musí nutne obsahovať komplexné znalosti, ktoré nie je ľahké nájsť medzi neexpertmi
 - Samovysvetľovanie – ES musí byť schopný vysvetliť vlastné odvodenie rozhodnutí
 - Typy (generácie) ES
 - Prvá generácia – klasické bázy IF-THEN pravidiel
 - Druhá generácia – pokročilejšie modelovanie znalostí a metódy odvodzovania (napr. s fuzzy logikou)

Experné systémy (2)

- Základné prvky ES
 - Subsystém získavania znalostí
 - Časť zodpovedná za extrakciu znalostí od experta a jej prevedenie do podoby znalostí v znalostnej báze
 - Expert + Znalostný inžinier (IT špecialista)
 - Znalostná báza
 - Skutočný základ ES, obsahuje relevantné znalosti o probléme, pričom obsahuje
 - Fakty popisujúce charakteristiku problémovej situácie
 - Špeciálne heuristiky alebo pravidlá experta pre riešenie problémov v danej doméne
 - (Meta-pravidlá) – všeobecné pravidlá odvodzovania
 - Inferenčný (odvodzovací) mechanizmus
 - Interpretačný systém – program realizujúci odvodzovanie nad bázou znalostí a aktuálnymi údajmi o prípade
 - Rozhranie používateľa – väčšinou v prirodzenom jazyku
 - Báza údajov
 - popis aktuálnych údajov o práve riešenom prípade
 - Vysvetľovací mechanizmus
 - systém pre poskytnutie vysvetlení odvodení, ku ktorým dospel inferenčný mechanizmus

Príklad – Produkčný systém pravidiel ES

Báza znalostí - všeobecné znalosti v báze (produkčné pravidlá – AND/OR graf)

Ak zviera má tmavé škvarky a je dravec a má pieskovú farbu potom je gepard.

Ak zviera má pieskovú farbu je dravec a má čierne pruhy potom je tiger.

Ak zviera je cicavec a žerie mäso potom je dravec.

Ak má srst alebo dáva mlieko potom je cicavec.

Báza údajov – čo vieme o prípade

ma_srst

ma_cierne_pruhy

ma_pieskovu_farbu

zerie_maso

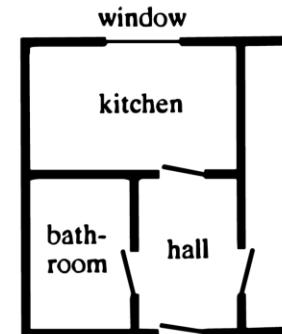
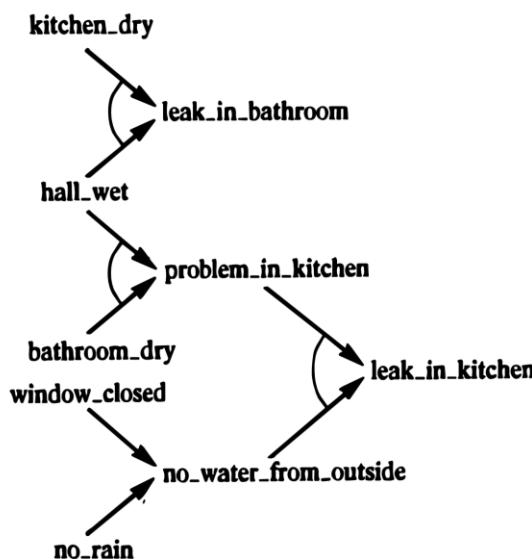


Z toho Inferenčný Mechanizmus (ideme v tomto prípade zdola nahor) odvodí že je to 1. cicavec (srst) 2. dravec (cicavec a žerie maso) 3. tiger (dravec + má pruhy a je pieskový) => koncový („koreňový“) uzol => je to tiger

Iný príklad:

Voda v byte ...

Ako by k tomu mohlo dôjsť = báza znalostí

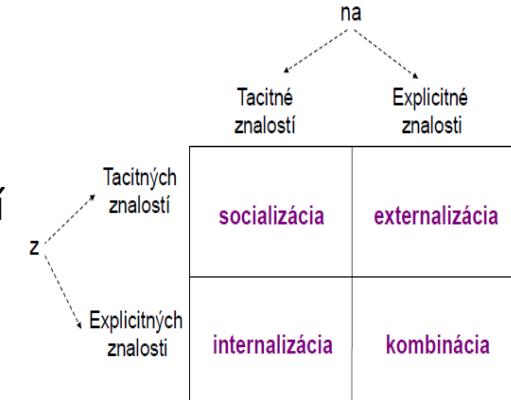


Manažment znalostí

- Jedným z prístupov ako riešiť problémy preskriptívnej analýzy je aj využitie predchádzajúcich znalostí uložených v KMS = Knowledge Management System = Systém manažmentu znalostí => znalosť je takisto podnikový zdroj
- Podnik potrebuje explicitne pracovať so znalosťami
 - Pre riešenie komplexných úloh, konkurencieschopnosť s inými podnikmi, uchovanie znalostí ľudí, ich zdielanie v podniku, ...
 - Ukazuje sa že systematická práca so znalosťami zvyšuje podniku šancu prežiť a konkurovať na trhu (znalostná ekonomika)
- Dáta – Informácie – Znalosti (jedna z definícií)
 - 1. Dáta: fakty, obrázky, zvuky
 - 2. Dáta + interpretácia + význam = Informácie (formátované, filtrované a summarizované dátá)
 - 3. Informácie + akcia + aplikácia = Znalosti (idey, pravidlá a procedúry, ktoré vedú akcie a rozhodnutia)
- Vhodná definícia (z pohľadu PA) [Turban1992]: Znalosť je informácia, ktorá je organizovaná a analyzovaná, aby sa stala zrozumiteľnou a použiteľnou na riešenie problémov alebo na rozhodovanie.

Manažment znalostí (2)

- Základné rozdelenie znalostí
 - Explicitné – vyjadrené, dokumentované, formalizované a ľahko zdieľané cez IKT
 - Tacitné (nevyjadrené) – uchovávané v ľudskej mysli, ťažko formalizovateľné do explicitnej formy (alebo vôbec)
- Vytváranie znalostí v organizácii je neustály proces = životný cyklus znalostí
 - Jeden z modelov, tzv. SECI model, zahŕňa 2 formy znalostí (tacitné a explicitné), dynamickú interakciu (prenos) a 4 procesy tvorby znalostí (socializácia, externalizácia, kombinácia a internalizácia)
 - Organizácie vytvárajú znalosti cez interakcie medzi explicitnými a nevyjadrenými znalostami = znalostná konverzia => znalosti sa tak rozširujú (kvantity aj kvality)
 - Prvky SECI modelu
 - **Socializácia** – prevod nových nevyjadrených znalostí cez zdieľanú skúsenosť – napr.: spoločným trávením času, formálne a neformálne stretnutia, aj mimo pracoviska, komunikácia (organizácia často získava a využíva nevyjadrené znalosti uložené u zákazníkov alebo dodávateľov tým, že s nimi komunikuje)
 - **Externalizácia** – prevod nevyjadrených znalostí na explicitné, znalosť sa stáva dostupnejšou, zdieľa sa s ostatnými – napr. vytvorenie dokumentu, grafu, ...
 - **Internalizácia** – proces vytvorenia tacitnej znalosti z explicitnej, jednotlivci konvertujú explicitný zdroj na svoju internú (nevyjadrenú) podobu – obohatenú o vlastné premýšľanie a skúsenosti, tzv. „učenie prácou“ – napr. školenia, štúdium návodov, ...
 - **Kombinácia** – premena explicitných znalostí do komplexnejších a systematickejších súborov explicitných znalostí, explicitné znalosti sú zbierané (z rôznych zdrojov) a následne kombinované pre formovanie nových znalostí => po nich nasleduje zdieľanie kombinovanej znalosti v organizácii



Manažment znalostí (3)

- Cieľom systému manažmentu znalostí = KMS = je práve využitie informačných prostriedkov pre vytvorenie vhodného prostredia pre intenzívny priebeh procesov tvorby znalostí
- Základné funkcie KMS
 - Vytvorenie a extrakcia znalostí + uloženie
 - Ľudia vytvoria novú znanosť (nové riešenie problému), túto znanosť musí KMS extrahovať a uložiť do svojho úložiska
 - Zdieľanie a manažment uložených znalostí
 - Existujúce znalosti v KMS sú zdieľané ľuďmi v podniku riešiacimi podobný problém (dôležité: vedieť identifikovať uložené znalosti užitočné pre nový prípad)
 - Uložené znalosti musí KMS manažovať a udržiavať v rozumnom formáte, ako aj zabezpečiť ich relevantnosť a presnosť
 - Podpora životného cyklu vytvorených znalostí
 - Znalosti prechádzajú celým životným cyklom a obohacujú sa o nové poznatky a skúsenosti => lepšia budúca aplikovateľnosť a užitočnosť + (udržanie relevantnosti a presnosti)
- Technologické predpoklady KMS
 - Technológie pre podporu základných funkcií KMS – práca z dátami, z rôznymi zdrojmi, dokumentmi, prepájanie ľudí, ...
 - Dôležité prvky:
 - Intranet / Internet, Systémy pre správu dokumentov, Nástroje pre vyhľadávanie informácií (Information Retrieval), Kolaboratívne systémy = Groupware (komunikácia, brainstorming, spoločná práca na dokumentoch a vývoji,...) , Dátové sklady a DM nástroje, ...

Po PA – čo ďalej ?

- PA ako predmet nadviazal na základy R/Python (predmet JDA), cieľom bol základný prehľad o problematike PA (resp. DA – dátovej analytiky) + vyskúšať si R/Python pre riešenie vybraných typov úloh
- !!! V praxi existuje oveľa viac metód a postupov analýz, aj rôznorodejšie úlohy ktoré sa dajú riešiť !!! (nie je možné pokryť to samozrejme v jednom predmete)
- Podrobnejšie sa vybraným oblastiam budú venovať ďalšie predmety v Ing. Štúdiu (v študijnom programe HI), napr.
 - Zimný semester 1.ročník Ing.
 - Detailnejší pohľad na **heuristické metódy** a postupy, **optimalizáciu** pomocou takýchto metód => **Heuristické optimalizačné procesy** (HOP)
 - Detailnejší pohľad na **proces KDD** (CRISP-DM) a súvisiace postupy + vybrané **algoritmy + práca s** ďalšími rozširujúcimi **nástrojmi pre KDD** ako RapidMiner => *Objavovanie znalostí* (OZ)
 - Doplnenie znalostí zo **štatistiky** => *Aplikovaná štatistika* (AŠ)
 - (+ voliteľný) **Ekonometria, analýza časových radov** dát, **regresné** a dynamické modely => *Inžinierska ekonometria* (IE)
 - Letný semester 1.ročník Ing.
 - **Technológie** a metódy pre **BigData** analýzy, možnosť práce s privátnym **cloudom** a súvisiacimi technológiami, **NoSQL** (všetko dnes veľmi žiadane) => *Technológie spracovania veľkých dát* (TSVD)
 - (+ voliteľný) Rozšírenie vedomostí na tému inteligentné **algoritmy pre učenie modelov** (základ pre vytváranie modelov v procese KDD ako rozhodovacie stromy, atď.) => *Strojové učenie* (SU)
 - Zimný semester 2.ročník Ing.
 - Podrobnejší pohľad na **text-mining, vyhľadávanie informácií** (IR – information retrieval), **manažment znalostí** a dokumentov v organizácii => *Manažment znalostí* (MZ)
 - Detailnejšie pochopenie IT prostriedkov **manažérskej úrovne podniku**, podrobny pohľad na problematiku **dátových skladov** + praktické zručnosti s ich tvorbou => *Manažérské informačné systémy* (MIS)
 - (+ voliteľný predmet) *Pokročilé metódy analýzy dát - neurónové siete / hlboké učenie – deep learning*
 - (+ voliteľný predmet) podrobnejší pohľad na problematiku sémantického webu a **analýzy sociálnych sietí** => *Semantický a sociálny web* (SaSW)