Katedra informatiky FEI VŠ-TUO

Referát do předmětu Metody analýzy dat

Rapid Miner

Květen 2011

Jan Górecki

1. RAPID MINER

1. 1. Základní informace

RapidMiner (dále jen RM) je open-source systém pro Data Mining. RM je šířen ve dvou verzích:

1) Community edition – verze zdarma ve formě open-source, poskytována bez jakýchkoli záruk
2) Enterprise edition – placená verze poskytována se zárukou

RM je kompletně napsán v jazyce Java, což umožňuje spouštět tento software na téměř jakémkoli operačním systému – pro jeho spuštění je ale tedy **nutné** mít nainstalován Java Runtime Environment (JRE) verze 5 nebo vyšší.

Download a instalace

Adresa pro stažení: http://rapid-i.com/

Verze zdarma se tedy jmenuje Community Edition. V současné době (2010) je nejnovější verze 5.0 – té bude věnován následující popis.

Klikneme na menu - *Downloads* / *Download RapidMiner Community Edition*

Zde si můžeme stáhnout instalační balík RM pro požadovaný operační systém. V následujícím popisu se budeme věnovat 32bitové verzi pro Windows (ostatní verze jsou velmi podobné).

Dále si můžeme v této sekci stáhnout starší verze programu, video tutoriály (velmi užitečné především pro začátek – navíc dále v textu se na ně budeme odkazovat v popisech jednotlivých součástí RM), manuály (v současné době je pro verzi 5.0 manuál jen v němčině), atd.

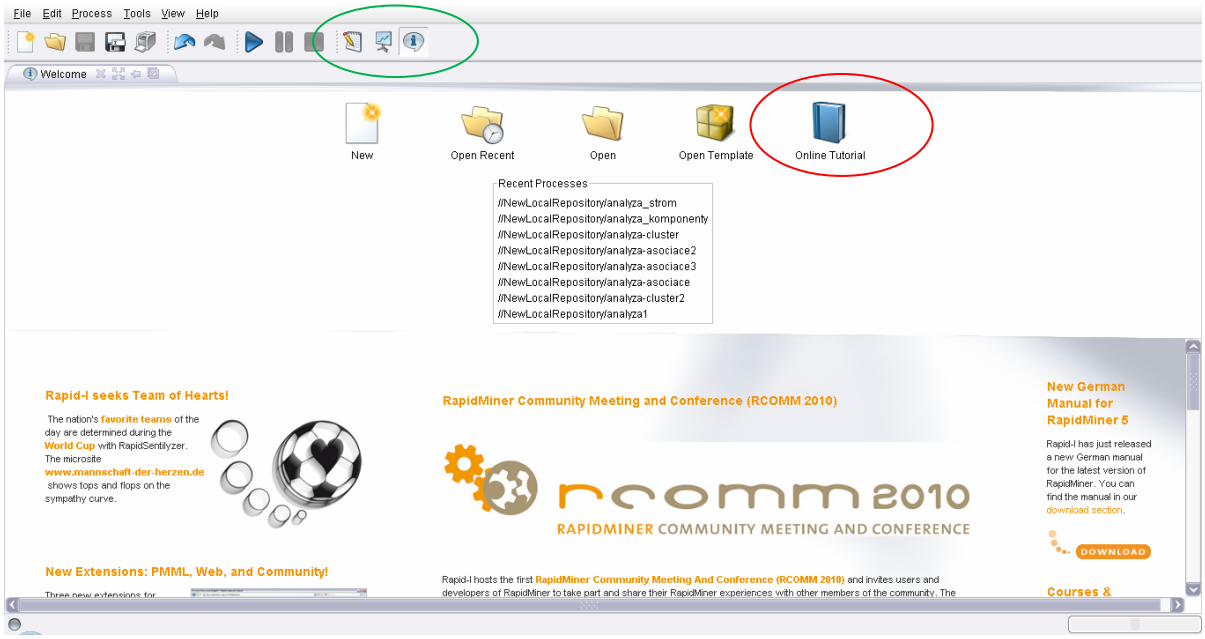
Máme tedy stažený soubor *rapidminer-5.0x32-install.exe*, který spustíme a podle instrukcí, které se nám postupně zobrazují v průvodci, RM nainstalujeme (zhruba 150MB). Pro ostatní systémy je instalace zdokumentována v *Installation Guide* (ve stejné sekci, jako je stažení RM).

RM máme nainstalován a spustíme.

Seznámení se s RM

Při prvním spuštění po Vás RM bude chtít nadefinování cesty pro úložiště (repository) všech dat, které budete v RM používat (tzn. jak načítat, tak ukládat). Tomuto problému se věnuje *Video Tutorial 1* – prohledněte si jej a podle instrukcí úložiště nastavte.

Po nastavení úložiště uvidíme tuto uvítací obrazovku:



Pustíme *Online Tutorial* (ikona v červené elipse). V tutoriálu je dobré přečíst si několik prvních kroků (*Steps*) a vyzkoušet si některé funkce. Spolu s tímto velmi doporučuji shlédnout videa *Quick Tour* a *Video Tutorials*.

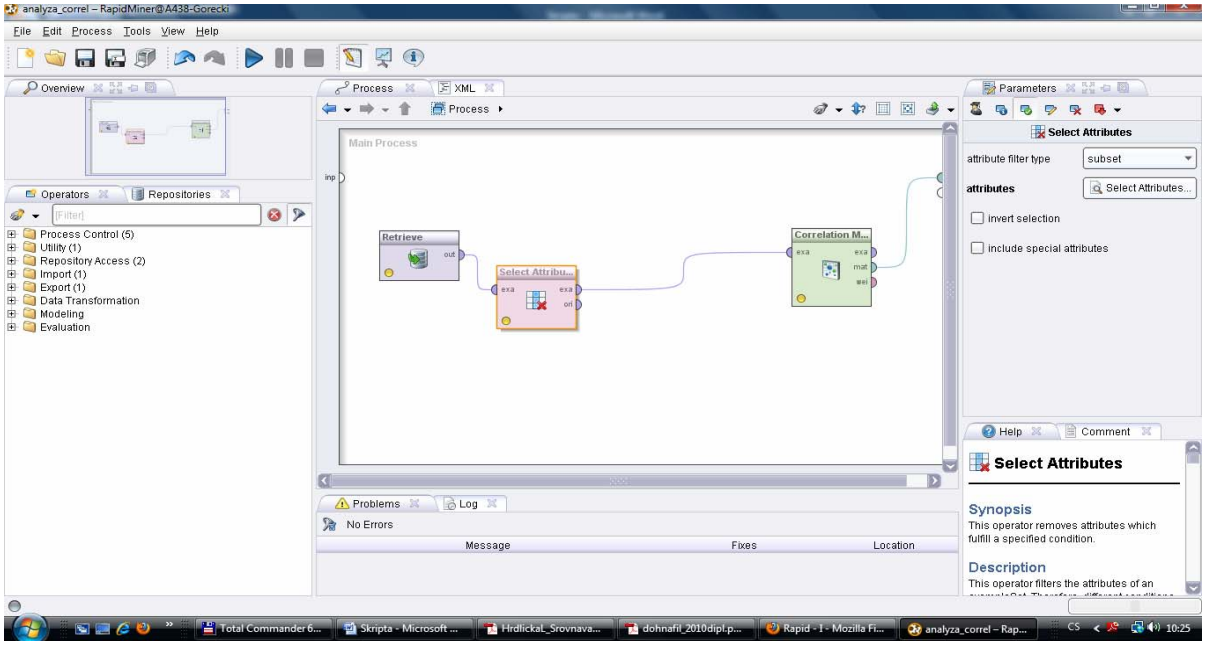
Práce s RM probíhá ve dvou módech:

1) V návrhovém módu, který je určen pro volbu a nastavení dataminingových analýz na zvolená data (tomuto módu odpovídá *Design Workspace* – levá ikona v zelené elipse)
2) V módu pro prohlížení výsledků, ve kterém si lze prohlížet v různých formách výsledky analýz navržených v *Design Workspace* (tomuto módu odpovídá *Result Workspace* – prostřední ikona v zelené elipse)

Pozn.: pravá ikona v zelené elipse přepne RM na uvítací obrazovku.

1.2. Design Workspace 1

V základním nastavení rozložení oken vypadá *Design Workspace* takto:



V levé části vidíme okna *Overview*, *Operators* a *Repositories*.

Okno *Overview* slouží jako lupa okna *Process*.

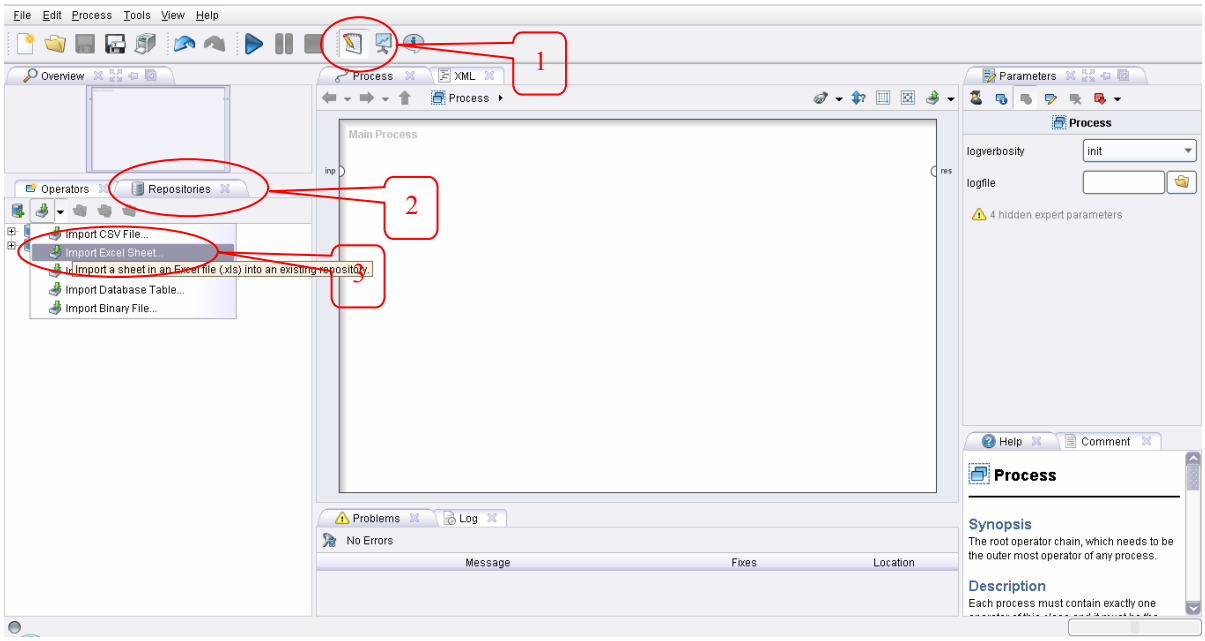
Okno *Operators* slouží pro volbu jednotlivých operátorů, které budou součástí dataminingové analýzy. Pro každou analýzu je vždy nutno vytvořit jeden samostatný proces, který bude té jedné konkrétní analýze odpovídat. Tvorba procesu probíhá tak, že se jednotlivé operátory řetězí mezi sebou v pořadí, ve kterém budou následně spuštěny. Pro vložení operátoru do procesu stačí zvolený operátor přetáhnout myší z okna *Operators* do okna *Process*. Každému operátoru poté odpovídá jedna ikona.
Okno *Repositories* slouží k importu dat do prostředí RM (tomuto se věnuje následující kapitola -

1.3. Import dat), ke vkládání dat do procesu, popř. k otevírání již předem uložených procesů.

Nyní se tedy věnujme importu dat do RM a poté se opět vrátíme k popisu okna Design Workspace.

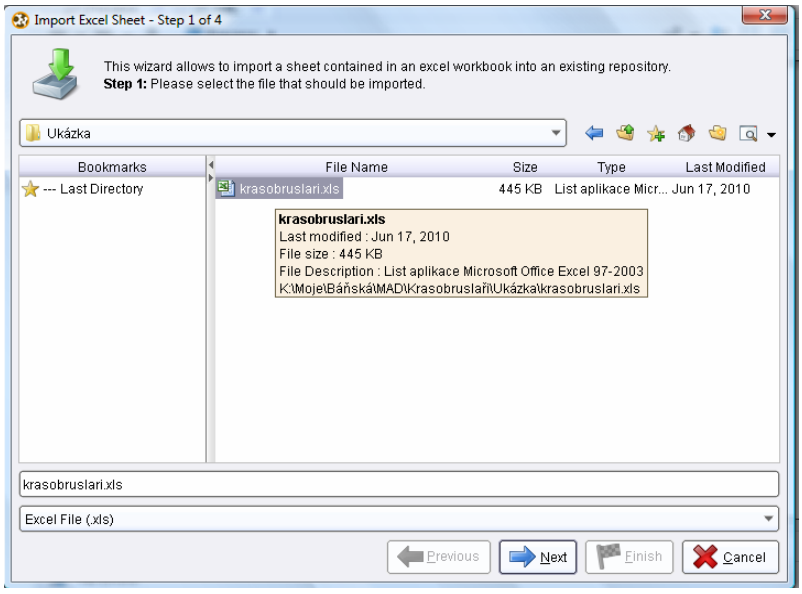
1.3. Import dat

Abychom měli co analyzovat, potřebujeme do RM načíst nějaká data. Datových formátů lze do RM načíst mnoho, my si popíšeme načtení dat ve formátu Microsoft Excel, ostatní formáty lze načíst obdobným způsobem.

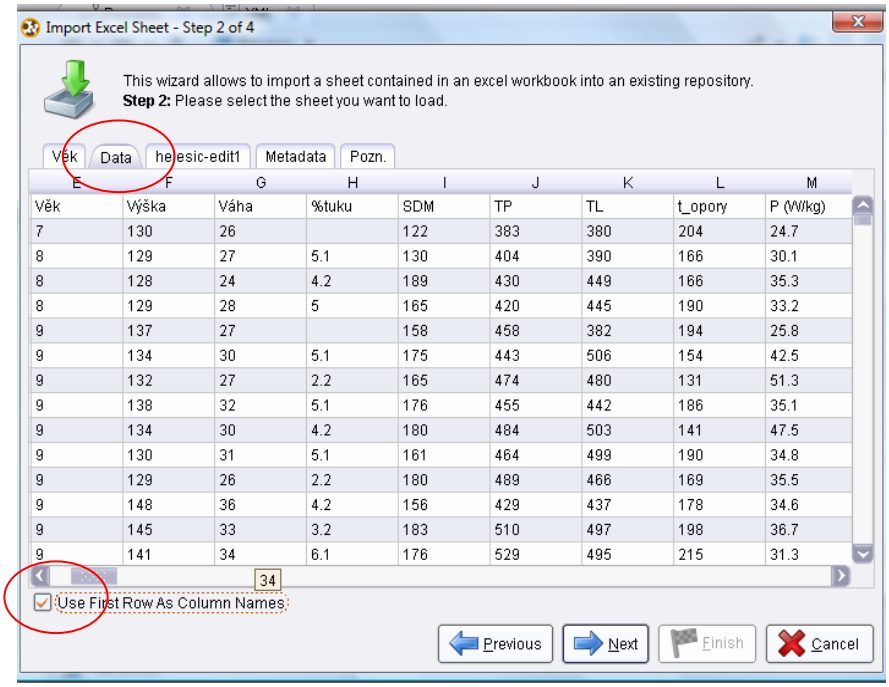


Přepneme se do *Design Workspace* (elipsa 1), zobrazíme si okno *Repository* (elipsa 2) a spustíme menu s *Import Excel Sheet*.

Zvolíme soubor s daty:



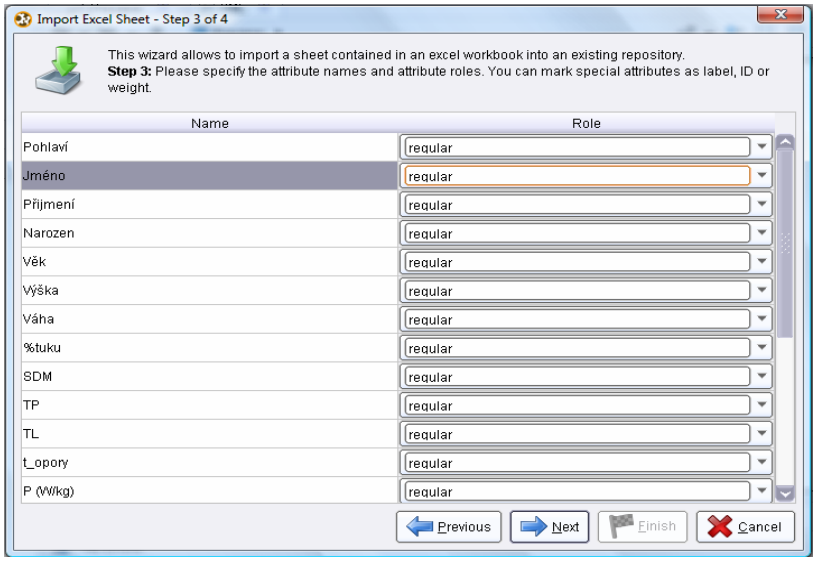
Dále vybereme list, ve kterém jsou požadovaná data (horní elipsa):



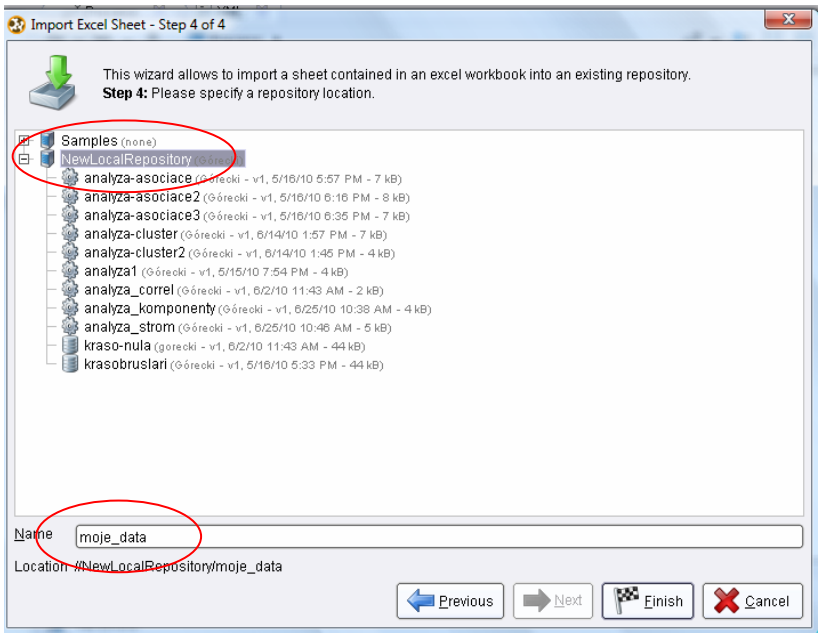
Věk	Výška	Váha	%tuku	SDM	TP	TL	t_opory	P (W/kg)
7	130	26		122	383	380	204	24.7
8	129	27	5.1	130	404	390	166	30.1
8	128	24	4.2	189	430	449	166	35.3
8	129	28	5	165	420	445	190	33.2
9	137	27		158	458	382	194	25.8
9	134	30	5.1	175	443	506	154	42.5
9	132	27	2.2	165	474	480	131	51.3
9	138	32	5.1	176	455	442	186	35.1
9	134	30	4.2	180	484	503	141	47.5
9	130	31	5.1	161	464	499	190	34.8
9	129	26	2.2	180	489	466	169	35.5
9	148	36	4.2	156	429	437	178	34.6
9	145	33	3.2	183	510	497	198	36.7
9	141	34	6.1	176	529	495	215	31.3

Použijeme první řádek jako názvy atributů, pokud je potřeba (spodní elipsa).

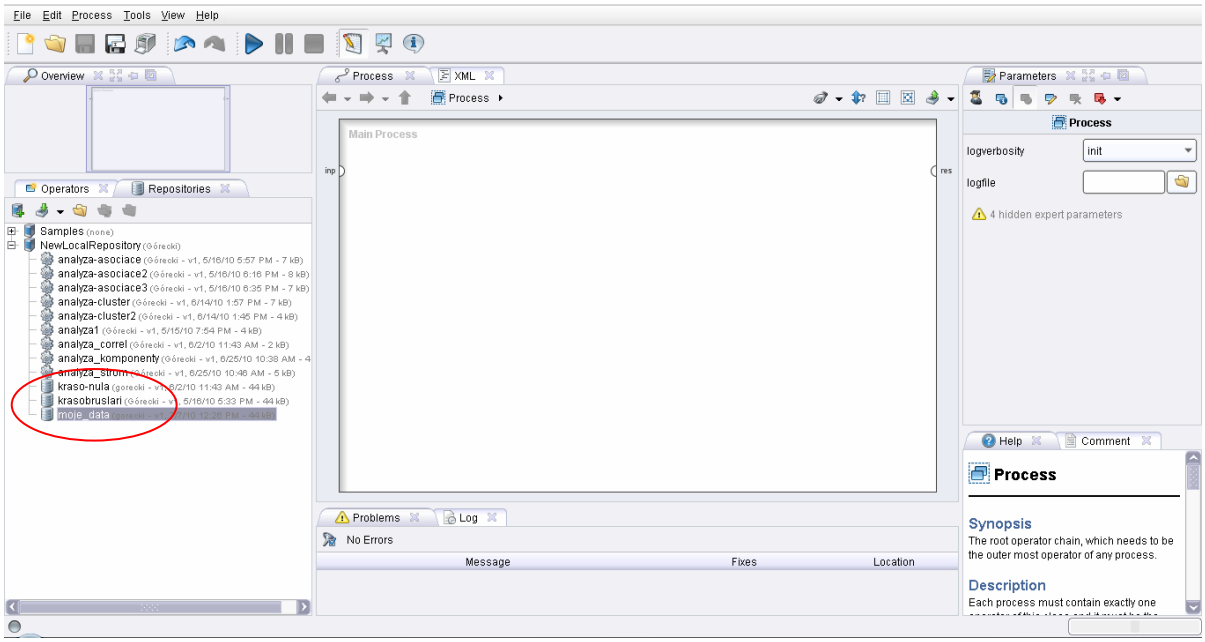
Nastavíme role jednotlivých atributů (můžeme ponechat jak je):



Vybereme název (spodní elipsa) a umístění (horní elipsa) pro importovaná data v úložišti:

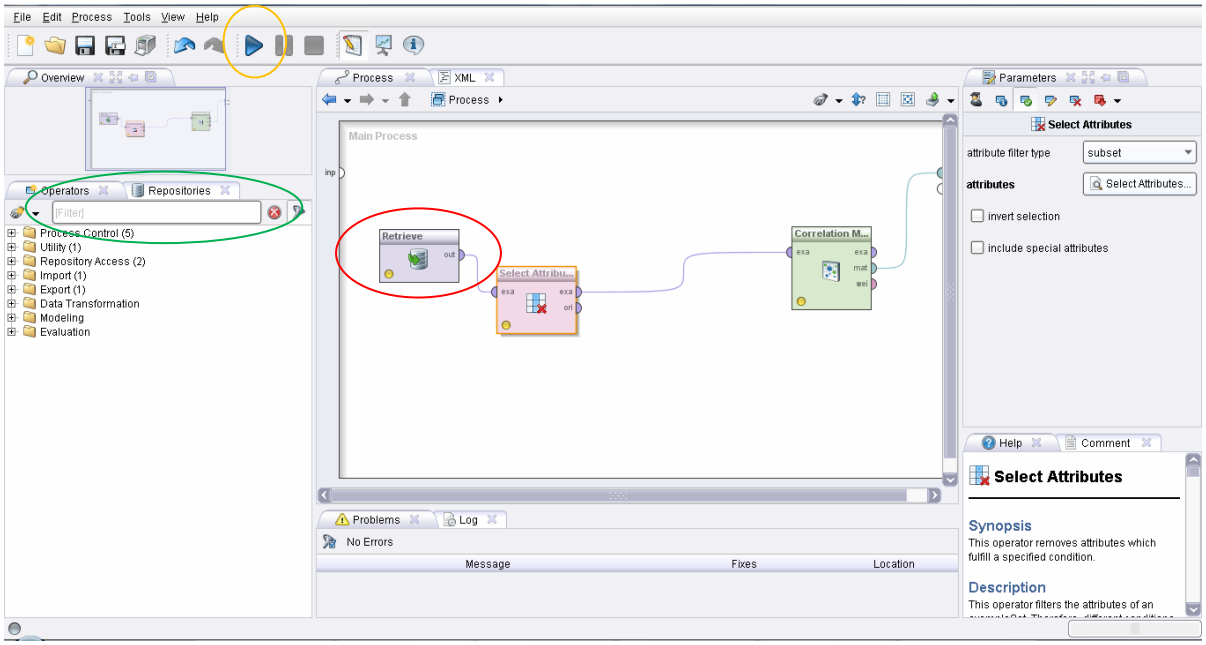


Importovaná data nyní vidíme v úložišti:



1.4. Design Workspace 2

Vraťme se opět k rozložení *Design Workspace*.



Uprostřed vidíme okno *Process*, *XML*, *Problems* a *Log*.

Okno *Process* je stěžejní okno RM. Zde se nastavuje, jak bude proces probíhat (pod pojmem proces tedy rozumíme kompletní jeden typ analýzy dat – tzn. načtení dat, předzpracování, modelování, popř. ověřování přesnosti modelu). Jako první vložíme do procesu data, která chceme analyzovat. Jelikož již máme nějaká data do RM importována (viz. předchozí kapitola), stačí z okna *Repositories* data přetáhnout myší do okna *Process*. V okně *Process* se Data zobrazí jako ikona s popiskem *Retrieve* (viz. červená elipsa). Na tato data nyní můžeme aplikovat operátory, které odpovídají různým dataminingovým metodám (současná verze nabízí zhruba 400 operátorů). Tyto operátory odpovídají různým typům předzpracování dat (standardizace, filtrace), modelování dat (rozhodovací stromy, clustering, asociace) a ověřování (popř. porovnávání) přesnosti modelů dat (např. ověřování přesnosti rozhodovacího stromu pomocí cross-validation).

Na obrázku je pro příklad jako druhá ikona v řetězci použit operátor *Select Attributes* (volba atributů pro zpracování). Tento operátor se přetáhne z okna *Operators*. V tomto okně ho lze najít buďto procházením stromové struktury operátorů nebo jej lze najít vyhledáváním přes filtr (viz. zelená elipsa).

Aby proces fungoval správně, je třeba operátory (ikony) mezi sebou propojit v pořadí podle toho, jak mají za sebou jednotlivé části procesu probíhat. Máme-li tedy načtená data (ikona *Retrieve*), napojíme na ni ikonu procesu *Select Attributes* takto: klikneme levým tlačítkem na malý půlkruh v pravé části ikony Retrieve (out) - začne se nám od něho táhnout čára - a potom klikneme na malý půlkruh v levé části ikony Select Attributes (exa – vstup dat do

operátoru). Tím se operátory propojí, resp. výstup prvního operátoru se napojí na vstup druhého operátoru. Obdobným způsobem můžeme v procesu napojovat další operátory (viz. na obrázku je napojen ještě operátor *Correlation Matrix*). Výstup z posledního operátoru je pak potřeba napojit na malý půlkruh *res* v pravé části okna *Process*.

Pokud klikneme libovolný operátor v okně *Process*, zobrazí se v pravé části *Desing Workspace* v okně *Parameters* parametry tohoto operátoru, které můžeme dle libosti upravovat. Na konkrétním případu operátoru *Select Attributes* můžeme tedy nastavit, které atributy z dat budeme používat v analýze (dále v procesu).

Okno *XML* zobrazuje obsah okna Process v *XML* textové formě.

Okno *Problems* zobrazuje problémy při tvorbě jednotlivých procesů, popř. nabízí možnosti řešení těchto problémů (tzv. *Quick fixes*).

Okno *Log* zobrazuje log RM.

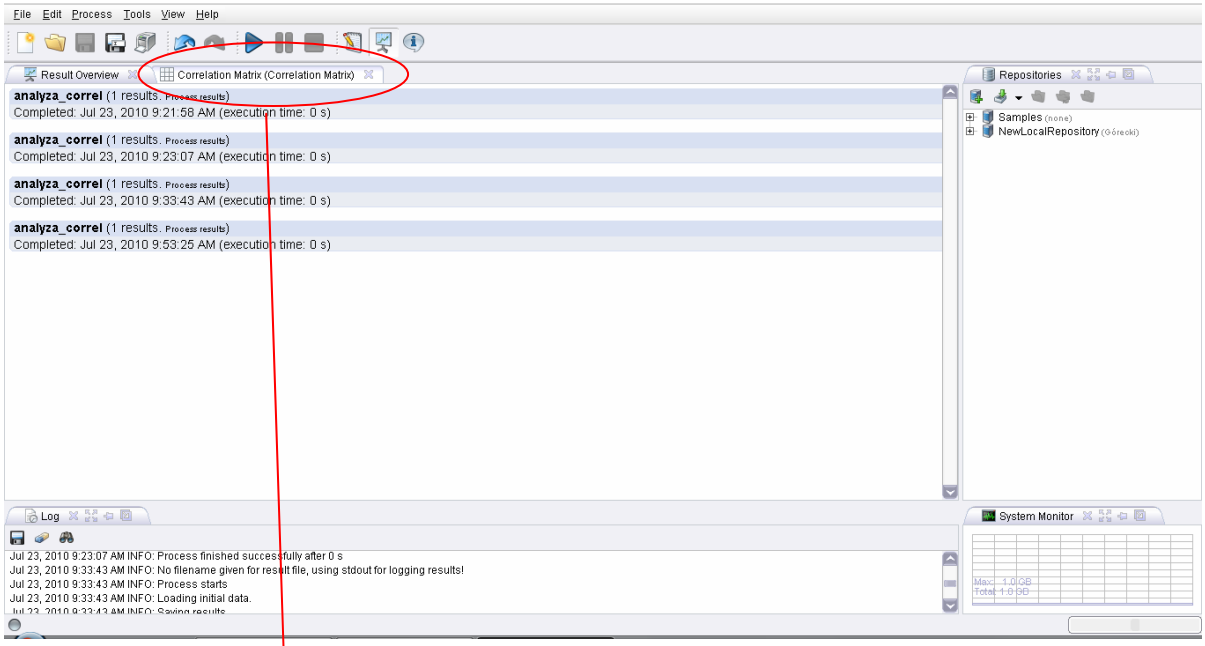
Okno *Help* v pravé části zobrazuje nápovědu k právě označenému objektu.

Okno *Comment* slouží ke vkládání vlastních komentářů.

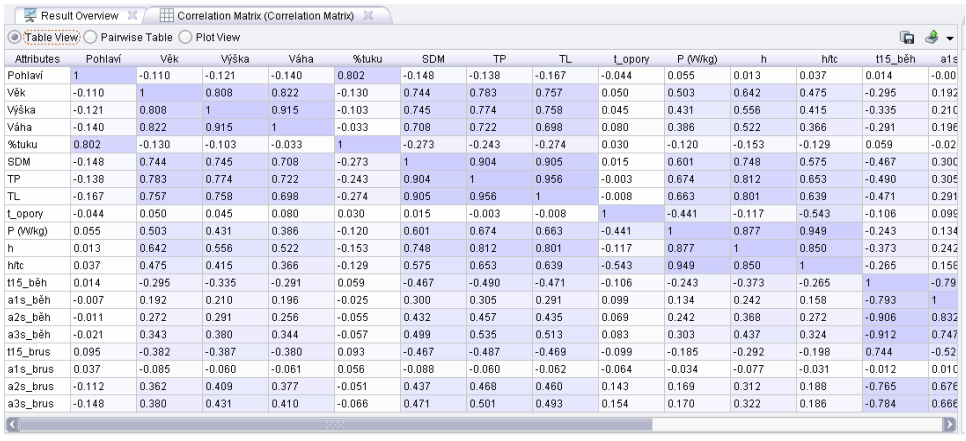
1.5. Result Workspace

Po spuštění (tlačítko v oranžové elipse) a úspěšném průběhu procesu vytvořeném v *Design Workspace* se RM automaticky přepne do módu pro prohlížení výsledků – *Result Workspace*.

Okna obrazovka je při základní rozložení oken rozdělena takto:

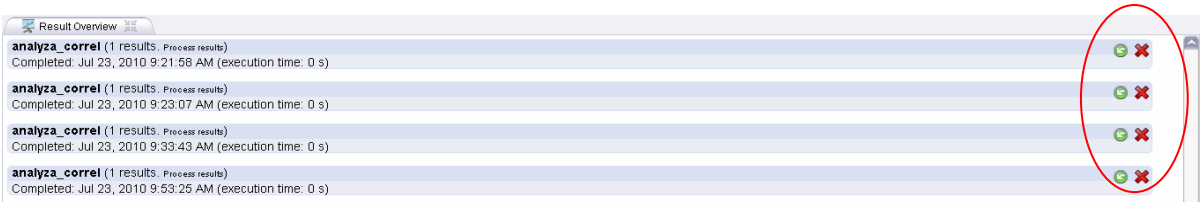


Okno *Result Overview* vlevo nahoře obsahuje přehled výsledků průběhu jednotlivých procesů. Výsledky průběhu procesů jsou uspořádány chronologicky a to tak, že aktuální je nejníže v okně. Samotné výsledky procesů jsou potom zobrazeny v dalších oknech, které zobrazíte kliknutím na jejich záložky (hned vedle záložky okna *Result Overview* – viz. červená elipsa – v tomto případě je to okno s korelační maticí – viz. obrázek níže).



Attributes	Pohlaví	Věk	Výška	Váha	%tuku	SDM	TP	TL	t_opory	P (W/kg)	h	h/tc	t15_běh	a1s
Pohlaví	1	-0.110	-0.121	-0.140	0.802	-0.148	-0.138	-0.167	-0.044	0.055	0.013	0.037	0.014	-0.00
Věk	-0.110	1	0.808	0.822	-0.130	0.744	0.783	0.757	0.050	0.503	0.642	0.475	-0.295	0.192
Výška	-0.121	0.808	1	0.915	-0.103	0.745	0.774	0.758	0.045	0.431	0.556	0.415	-0.335	0.210
Váha	-0.140	0.822	0.915	1	-0.033	0.708	0.722	0.698	0.080	0.386	0.522	0.366	-0.291	0.196
%tuku	0.802	-0.130	-0.103	-0.033	1	-0.273	-0.243	-0.274	0.030	-0.120	-0.153	-0.129	0.059	-0.02
SDM	-0.148	0.744	0.745	0.708	-0.273	1	0.904	0.905	0.015	0.601	0.748	0.575	-0.467	0.300
TP	-0.138	0.783	0.774	0.722	-0.243	0.904	1	0.956	-0.003	0.674	0.812	0.653	-0.490	0.305
TL	-0.167	0.757	0.758	0.698	-0.274	0.905	0.956	1	-0.008	0.663	0.801	0.639	-0.471	0.291
t_opory	-0.044	0.050	0.045	0.080	0.030	0.015	-0.003	-0.008	1	-0.441	-0.117	-0.543	-0.106	0.099
P (W/kg)	0.055	0.503	0.431	0.386	-0.120	0.601	0.674	0.663	-0.441	1	0.877	0.949	-0.243	0.134
h	0.013	0.642	0.556	0.522	-0.153	0.748	0.812	0.801	-0.117	0.877	1	0.850	-0.373	0.242
h/tc	0.037	0.475	0.415	0.366	-0.129	0.575	0.653	0.639	-0.543	0.949	0.850	1	-0.265	0.158
t15_běh	0.014	-0.295	-0.335	-0.291	0.059	-0.467	-0.490	-0.471	-0.106	-0.243	-0.373	-0.265	1	-0.79
a1s_běh	-0.007	0.192	0.210	0.196	-0.025	0.300	0.305	0.291	0.099	0.134	0.242	0.158	-0.793	1
a2s_běh	-0.011	0.272	0.291	0.256	-0.055	0.432	0.457	0.435	0.069	0.242	0.368	0.272	-0.906	0.832
a3s_běh	-0.021	0.343	0.380	0.344	-0.057	0.499	0.535	0.513	0.083	0.303	0.437	0.324	-0.912	0.747
t15_brus	0.095	-0.382	-0.387	-0.380	0.093	-0.467	-0.487	-0.469	-0.099	-0.185	-0.292	-0.198	0.744	-0.52
a1s_brus	0.037	-0.085	-0.060	-0.061	0.056	-0.088	-0.060	-0.062	-0.064	-0.034	-0.077	-0.031	-0.012	0.010
a2s_brus	-0.112	0.362	0.409	0.377	-0.051	0.437	0.468	0.460	0.143	0.169	0.312	0.188	-0.765	0.676
a3s_brus	-0.148	0.380	0.431	0.410	-0.066	0.471	0.501	0.493	0.154	0.170	0.322	0.186	-0.784	0.666

Okno *Result Overview* můžeme dále využít k tomu, abychom se vrátili k výsledkům předchozích procesů. Stačí kliknout na zelenou šipku u procesu, ke kterému se chceme vrátit (viz. obrázek níže) a proces, včetně veškerého nastavení všech parametrů operátorů uvnitř něj, se obnoví do *Design Workspace*. Poté stačí proces znovu spustit a můžeme si starší výsledky opět prohlížet.



Okno *Systém Monitor* vpravo dole v *Result Workspace* zobrazuje využití systémových prostředků.

Zbylá okna *Log* a *Repositories* mají stejný význam jako v *Design Workspace*.

2. ANALÝZA DAT V RAPID MINERU

2. 1. Úvod do analýzy dat v RM

V této části si ukážeme, jak nastavit RM tak, aby provedl určitý typ analýzy na naše data – tzn. vytvoříme si několik procesů (v *Design Workspace*), kde každý bude odpovídat vždy jednomu typu analýzy. Ukážeme si tedy proces, který po spuštění provede toto:

a) Vypočítá korelační matici

b) Vypočte hlavní komponenty

c) Nalezne asociační pravidla

d) Provede aglomerativní shlukování.

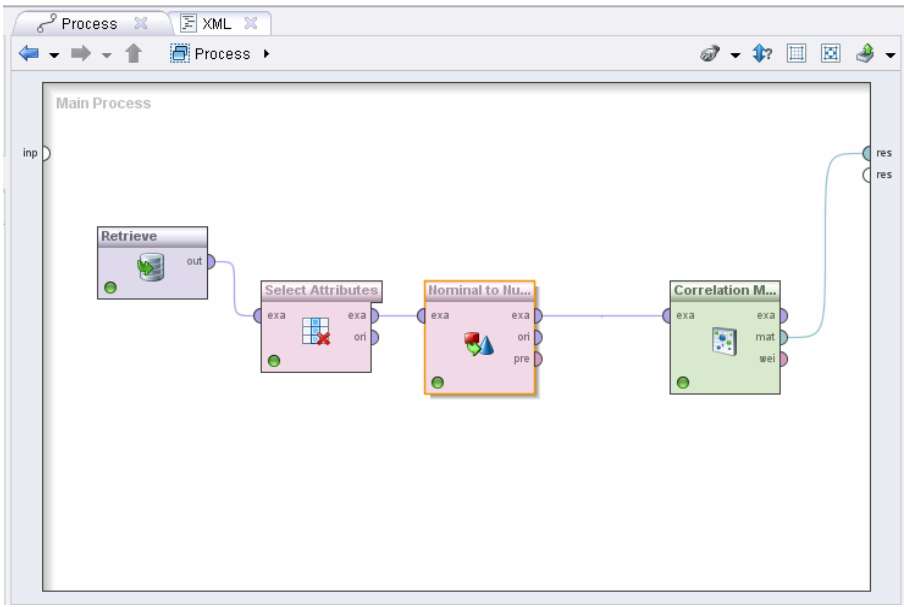
e) Zobrazí rozhodovací strom

Poznámka 1: Některé operátory se v okně *Process* zobrazují pod jiným jménem, než pod jakým je lze vyhledat v okně *Operators*. V takovém případě je potom jméno pro vyhledání operátoru uvedeno v závorkách za názvem operátoru. Např. operátor *AssociationRuleGenerator* (viz. obrázek níže) lze vyhledat po názvem *Create Association Rules*. V takovém případě budou v textu uvedeny oba názvy v pořadí *NázevOperátoruVOkněProcess*(*NázevOperátoruVOkněOperators*), např. tedy *AssociationRuleGenerator* (*Create Association Rules*).

Poznámka 2.: v dalším textu se několikrát objevují dva různé výrazy, které však znamenají vždy jedno a to samé. Jedná se o výrazy *záznam* a *objekt*. Při použití jednoho z těchto výrazů se vždy myslí jeden řádek relační tabulky (v databázovém jazyce jedna entita z relace). Dvojího značení je využito proto, že v určitých případech je vhodnější použít spíše jeden výraz a v ostatních ten druhý. Např. při výpočtu korelační matice je vhodnější používat výraz „*záznamy* s chybějícími údaji“, kdežto při shlukování je vhodnější používat výraz „vzdálenost mezi *objekty*“. Přesto se však vždy jedná o jedno a totéž.

2. 2. Korelační matice

Pro výpočet korelační matice v RM je nastavíme proces takto:



Na svá data (ikona *Retrieve*) navážeme operátor *Select Attributes*, ve kterém vybereme atributy pro výpočet (pokud nechceme použít všechny).

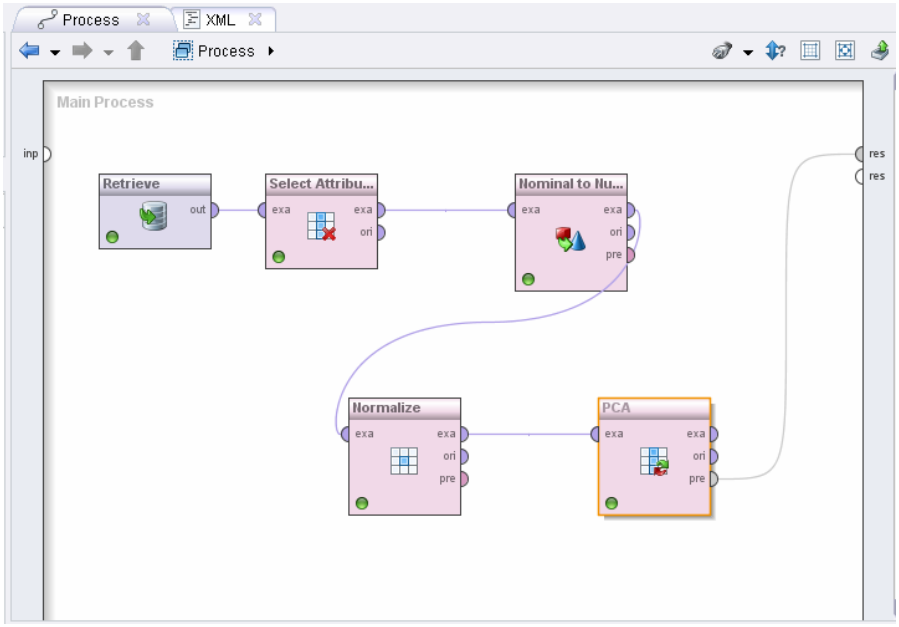
Dále navážeme operátor *Nominal to Numerical* , který převede kategorické atributy na binární pomocí dichotomizace (výpočet korelační matice vyžaduje mít všechny atributy reálné (popř. z nějaké podmnožiny reálných čísel)).

Dále pak navážeme operátor *Correlation Matrix* a ten pak dle obrázku (z operátoru použijeme výstup označený *mat* (matrix)) navážeme na *res* (results - malý půlkruh vpravo). Po spuštění se RM přepne do *Result Workspace* a zobrazí korelační matici.

RM umožňuje počítat koeficienty korelace i mezi atributy s chybějícími údaji. S chybějícími údaji RM nakládá tak, že je ignoruje (tzn. koeficient korelace mezi každými dvěma atributy se nepočítá ze všech záznamů v datech, ale pouze z podmnožiny takových záznamů, kde ani u jednoho atributu (z dvojice, pro kterou se zrovna koeficient počítá) údaj nechybí – jako bychom vždy vzali dva sloupce matice (odpovídající dvěma atributům), vypustili z ní všechny řádky, na kterých chybí alespoň jeden údaj a potom ze zbylých údajů již normálně koeficient korelace spočetli).

2. 3. Metoda hlavních komponent (PCA – Principal Component Analysis)

Pro výpočet hlavních komponent v RM nastavíme proces takto:



Na svá data (ikona *Retrieve*) navážeme operátor *Select Attributes*, ve kterém vybereme atributy pro výpočet (pokud nechceme použít všechny).

Pozn.: Zde je nutno podotknout, že metoda *PCA* neumí pracovat s atributy s chybějícími údaji. V operátoru *Select Attributes* tedy musíme takové atributy vyjmout z výpočtu nebo, což je vhodná alternativa v případě, že záznamů s chybějícími údaji není mnoho, použijme operátor *Filter Examples* a pomocí něj z dalšího výpočtu vyřadíme pouze tyto záznamy (nastavíme *condition class* v operátoru na volbu *no_missing_attributes*).

Dále navážeme operátor *Nominal to Numerical*, který převede kategorické atributy na binární pomocí dichotomizace (binarizace).

Jako další navážeme operátor *Normalize*, který provede standardizaci hodnot atributů. Standardizace atributů je potřeba, abychom odstranili z výpočtu závislost hodnot atributů na měrných jednotkách atributů (tzn. aby výpočet nezávisel na tom, zda např. výška objektu je uvedena v metrech nebo v centimetrech).

Jako poslední navážeme operátor *PCA* (*Principal Component Analysis*), který provede samotnou metodu hlavních komponent. Pro získání vlastních čísel (*Eigenvalues*) a vlastních vektorů (*Eigenvectors*) matice našich dat, je potřeba svázat výstup operátoru *Pre* (Preprocessing model) s půlkruhem vpravo *Res* (dle obrázku). Operátor není nutné pro naše potřeby nijak nastavovat (není třeba měnit žádné jeho parametry).

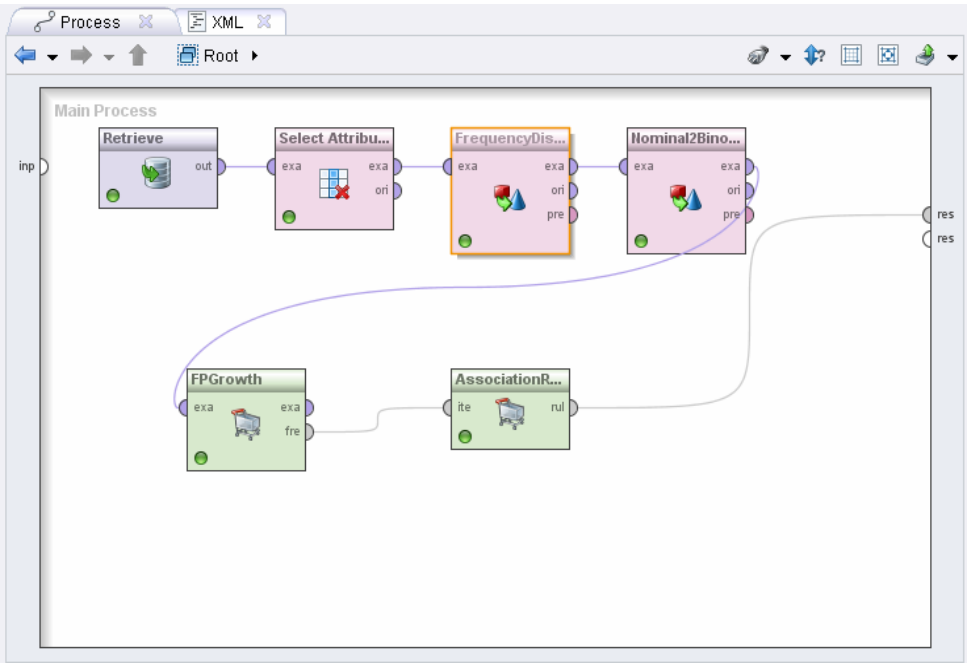
Výsledkem metody je potom sada vlastních čísel (seřazených od největšího k nejmenšímu) a k nim příslušející sada vlastních vektorů, které jsou vypočteny z vycentrované kovarianční matice získané z matice našich dat. Vlastní vektory tvoří novou souřadnou soustavu (vlastní vektory jsou navzájem kolmé, jelikož byly spočteny z kovarianční matice, která je vždy symetrická), jejíž osy odpovídají jednotlivým vlastním vektorům a směr těchto nových os je směr, ve kterém mají naše data rozptyl seřazen od největšího rozptylu k nejmenšímu – rozptyl v daném směru vlastního vektoru je totiž roven velikosti příslušného vlastního čísla – tedy pokud první nová osa odpovídá největšímu vlastnímu číslu, pak je rozptyl podél této osy největší, druhá osa když odpovídá druhému největšímu číslu, tak je rozptyl podél této osy druhý největší, atd.

Pokud je rozptyl v určitém směru (součet rozptylů v určitých směrech) výrazně vyšší než ve směrech ostatních, je potom možno tyto ostatní směry zanedbat a tím snížit dimenzi problému (snížit počet atributů), jelikož pro naše analýzy jsou důležité hlavně osy, podél kterých jsou objekty dobře rozlišitelné – tzn. osy na kterých je vysoký rozptyl.

Jakým způsobem (na základě výsledků získaných metodou hlavních komponent) lze potom snížit počet atributů při dalších analýzách, je popsáno ve skriptech v kapitole Hlavní komponenty.

2. 4. Asociační pravidla

Pro získání asociačních pravidel nastavíme proces takto:



Na svá data (ikona *Retrieve*) navážeme operátor *Select Attributes*, ve kterém vybereme atributy pro výpočet (pokud nechceme použít všechny).

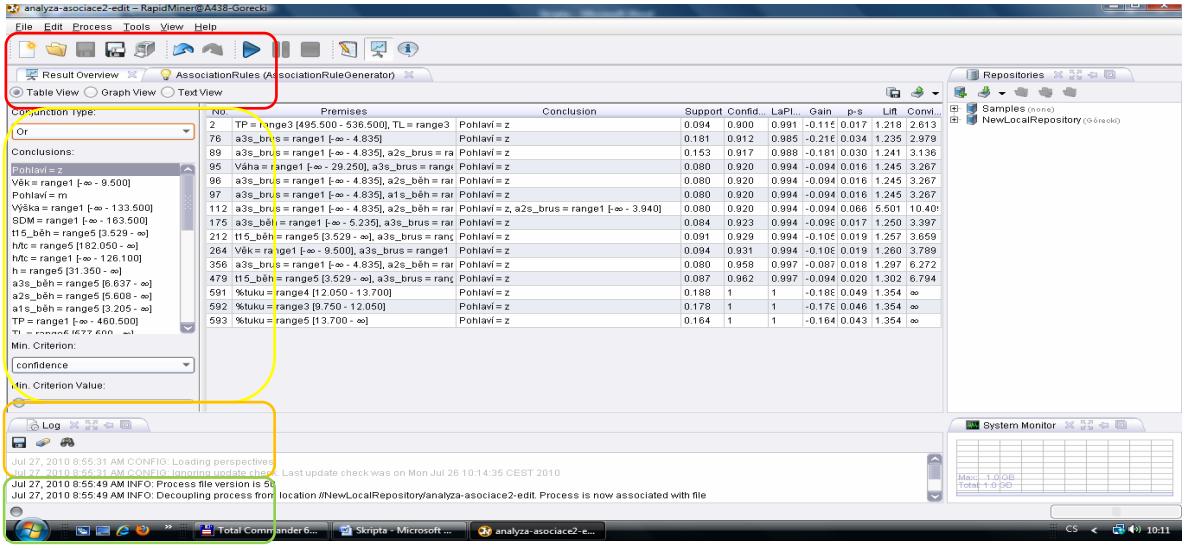
Dále navážeme operátor *FrequencyDiscretization* (*Discretize by Frequency*), který diskretizuje všechny reálné atributy do parametrem (*number of bins*) nastaveného počtu kategorií. Metoda pro vyhledávání asociačních pravidel v RM totiž umí pracovat pouze s binárními atributy, musíme tedy nejdříve reálné atributy diskretizovat do kategorií tímto operátorem a pak navážeme operátor *Nominal2Binominal* (*Nominal to Binominal)*, který každý diskretizovaný (kategoriální) atribut převede na sadu binárních atributů.

Dále navážeme operátor *FPGrowth (FP-Growth* - Frequent Pattern Growth), který nalezne množinu frekventovaných vzorů splňujících daná kritéria – kritériem je zde minimální podpora, vyjádřena v procentech vzhledem k celkovému počtu objektů - parametr *min support* z intervalu <0, 1>.

Algoritmus pro vyhledávání asociací je obdoba algoritmu Apriori, tudíž nejdříve se nalezne množina frekventovaných vzorů s minimální podporou a poté se z této množiny generují asociační pravidla. Toto generování zajišťuje operátor *AssociationRuleGenerator* (*Create Association Rules*), kde je potřeba nastavit kritérium pro vyhledávání pravidel – pokud budeme hledat pravidla s určitou minimální spolehlivostí, zvolíme parametr *criterion* jako *confidence* a parametr *min confidence* nastavíme procentuelně (z intervalu <0, 1>) na požadovanou míru.

Výstup operátoru pak navážeme dle obrázku na půlkruh *Res*.

Po spuštění se nám v *Result Workspace* zobrazí všechna nalezená pravidla splňující nastavená kritéria. Pravidla si potom můžeme prohlížet buďto všechny najednou anebo si je můžeme nechat zobrazovat ve skupinách, kde každá skupina pravidel odpovídá nějakému vybranému sukcedentu.



Podmínku, kterou má obsahovat sukcedent, zvolíme z okna ve žlutém obdélníku (*Conclusions*). Dále můžeme volit logickou spojku pro sukcedent (červený obdélník – *Conjunction type*) – pokud zvolíme spojku *And*, zobrazí se nám všechna pravidla, která mají v sukcedentu zvolenou podmínku buďto samotnou nebo v konjunkci s nějakou jinou podmínkou, pokud zvolíme spojku *Or*, zobrazí se nám všechna pravidla, která mají v sukcedentu zvolenou podmínku opět buďto samotnou nebo v disjunkci s nějakou jinou podmínkou.

Dále můžeme volit kritérium pro zobrazení pravidel (oranžový obdélník – *Min. Criterion*) a jeho nastavení (zelený obdélník – *Min. Criterion Value*). Když potom potáhneme táhlem doprava, mizí pravidla nesplňující kritérium (např. pro kritérium spolehlivost to platí tak, že hodnota táhla vlevo odpovídá minimální spolehlivosti nastavené na operátoru *AssociationRuleGenerator* a hodnota táhla nastaveného úplně vpravo odpovídá spolehlivosti 100%, tzn. zobrazí se pouze pravidla se 100%tní spolehlivostí).

Na jaké hodnoty nastavit minimální podporu a spolehlivost?

Obecně se doporučuje začít s nastavením pro minimální podporu na 5-10% počtu záznamů v datech. Pro hodnotu minimální spolehlivosti se doporučuje toto počáteční nastavení:

a) 90% pro přesná data, kde se neprojevuje nějaké subjektivní hodnocení – např. data z lékařských měření, sportovní výsledky, apod.

b) 70% pro data, kde se projevuje nějaké subjektivní hodnocení - typicky to jsou nějaká data získaná z dotazníků s otázkami typu „Jaký je Váš názor na to či ono?“, apod.

Dále platí, že se zvyšováním hodnoty parametru minimální podpory klesá počet nalezených pravidel (a naopak), a také, že se zvyšováním parametru minimální spolehlivosti taktéž klesá počet nalezených pravidel (a taktéž to platí i naopak). Těchto zákonitostí potom můžeme využít pro to, abychom získali soubor pravidel, který není ani příliš rozsáhlý, ani příliš malý.

Poznámka 1: Nemá smysl hledat pravidla se spolehlivostí 50%. Takové pravidlo říká toto: v případě, že platí podmínka z antecedentu, podmínka v sukcecedentu bude splněna s pravděpodobností stejnou, jakou je pravděpodobnost padnutí panny při hodu korunou. Rozhodování na základě takového pravidla tedy není příliš sofistikované.

Poznámka 2: Není vhodné nastavovat parametr minimální podpory příliš vysoko. Lepší je nejdříve snižovat počet nalezených pravidel zvyšováním minimální spolehlivosti a teprve až když je spolehlivost někde mezi 90% a 95% a soubor pravidel je stale příliš velký, tak potom lze počet pravidel snižovat zvyšováním minimální podpory. Zvyšováním minimální podpory se totiž můžeme připravit o závislosti, které se v datech nemají moc velkou podporu, přesto však mohou být zajímavé.

Které atributy při vyhledávání asociačních pravidel použít a které ne?

Velmi důležitá je volba atributů, které použijme při hledání asociačních pravidel. V RM není nutné se rozmýšlet, jak rozdělit atributy mezi antecedent a sukcedent, jelikož stačí pouze zvolit atributy, které chceme použít a RM projde všechny možné kombinace podmínek pro antecedent i sukcedent.

Obecně je vhodné použít všechny atributy, pro které může mít pravidlo smysl. Tímto způsobem lze získat obrovské množství pravidel, což má své pro a proti.

Pro: Pokud máme v našich datech atributy, mezi kterými je vysoká míra korelace (což se dozvíme z korelační matice), tomuto jevu odpovídající částečná lineární závislost se velmi pravděpodobně objeví v nalezených pravidlech a můžeme se o této závislosti dozvědět další upřesňující informace.

Proti: Pokud je v datech více atributů (mezi kterými je vysoká míra korelace), v pravidlech o těchto závislostech se pak mohou ztratit pravidla o dalších možná zajímavých (zřejmě nelineárních) závislostech (uvědomme si, že pomocí hodnot minimální podpory a spolehlivosti vždy optimalizujeme velikost souboru na nějakou „rozumnou“ míru – kritéria pak při zmenšování souboru pravidel můžeme nastavit příliš vysoko a tím zajímavé závislosti (které mohou mít spolehlivost či podporu o něco málo nižší) ztratit).

Je proto vhodné vždy ze skupiny vysoce korelovaných atributů ponechat jen jeden atribut a zbylé vypustit. V nalezených pravidlech se pak mohou objevit závislosti, o kterých jsme dosud z korelační matice nevěděli. Navíc se po vypuštění atributů sníží velikost souboru nalezených pravidel, a my tím pádem můžeme oslabit kritéria minimální podpory a spolehlivosti a hledat tak další (sice s menší podporou či méně spolehlivé, přesto možná zajímavá) pravidla.

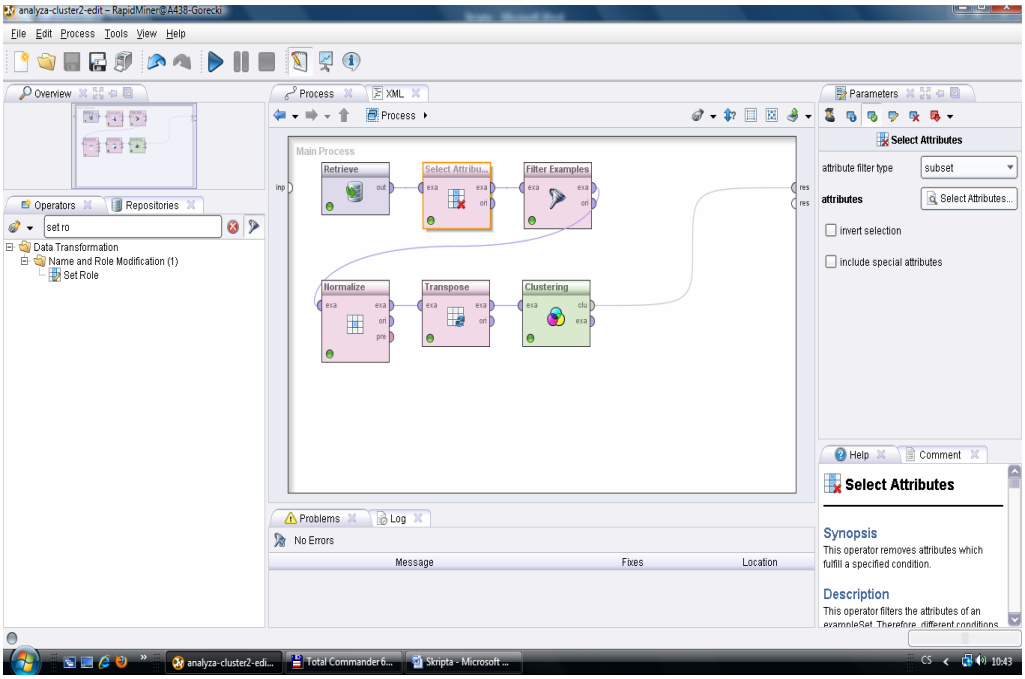
Druhým způsobem, jak snížit počet atributů, je pomocí metody hlavních komponent (viz. kapitola **Chyba! Nenalezen zdroj odkazů.**).

Na kolik kategorií diskretizovat reálné atributy?

Obecně se doporučuje lichý počet kategorií, aby existovala vždy nějaká „prostřední“ kategorie, která bude vyjadřovat průměr. Minimálně je tedy dobré zvolit 3 kategorie, o kterých by se dalo mluvit jako o kategorii podprůměrné, průměrné a nadprůměrné. Pro lepší rozlišení hodnot pak použít 5 kategorií, kde by pak přibyly k předchozím kategoriím kategorie vysoce podprůměrné a vysoce nadprůměrné. V případě potřeby jemnějšího rozlišení pak samozřejmě můžeme dále počet kategorií zvyšovat.

2. 5. Aglomerativní shlukování

Pro aglomerativní shlukování nastavíme proces takto:



Na svá data (ikona *Retrieve*) navážeme operátor *Select Attributes*, ve kterém vybereme atributy pro výpočet (pokud nechceme použít všechny).

Metoda neumí pracovat s atributy s chybějícími údaji. Buďto tedy v operátoru *Select Attributes* takové atributy vyjmeme z výpočtu nebo, což je vhodná alternativa v případě, že záznamů s chybějícími údaji není mnoho, navážeme (jako na obrázku) operátor *Filter Examples* a pomocí něj z dalšího výpočtu vyřadíme pouze tyto záznamy (nastavíme parametr *condition class* v operátoru na volbu *no_missing_attributes*).

Jako další navážeme operátor *Normalize*, který provede standardizaci hodnot atributů.

Jako další **můžeme** navázat operátor *Transpose*, který data transponuje a následné shlukování pak nebude mezi objekty, ale budou se shlukovat atributy. Pokud operátor nepoužijeme, shlukování bude probíhat „normálně“, tzn. budou se shlukovat podobné si objekty.

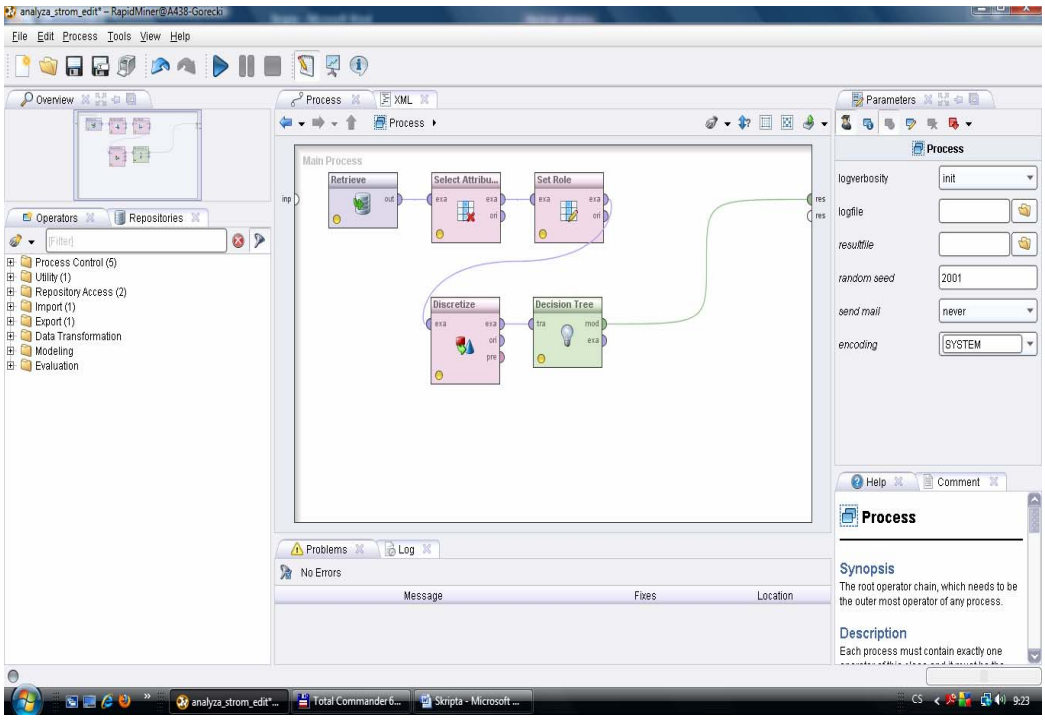
Jako poslední navážeme operátor *Clustering* (*Agglomerative Clustering*), který provede aglomerativní shlukování. Parametr *mode* nastavíme na strategii shlukování, kterou chceme použít – např. *SingleLink* je strategie nejbližšího souseda, která je používána při hledání přirozených shluků, tzv. α-shluků. Parametr *measure type* nastavíme na *MixedMeasures* a parametr *mixed measure* na *MixedEuclidianDistance*, čímž nastavíme jako míru rozdílnosti objektů Euklidovu vzdálenost.

Výsledkem je potom hierarchie shluků, kterou si můžeme nechat zobrazit ve formě grafu, dendrogramu nebo prostého textu.
Pro volbu atributů, které zahrnout do shlukovacího procesu, je vhodné využít informace z kapitoly Asociační pravidla (

Které atributy při vyhledávání asociačních pravidel použít a které ne?).

2. 4. Rozhodovací stromy

Pro získání rozhodovacího stromu (metoda generuje jen **binární** stromy) je nastavíme proces takto:



Na svá data (ikona *Retrieve*) navážeme operátor *Select Attributes*, ve kterém vybereme atributy pro výpočet (pokud nechceme použít všechny).

Tato analýza je určitým způsobem odlišná vůči všem předchozím analýzám. Ve všech předchozích analýzách hrály atributy stejnou roli, jinými slovy při provádění analýz jsme pouze řekli, které atributy chceme do analýzy zahrnout, a během analýzy bylo pak na každý atribut pohlíženo stejně. Zde tomu tak není. Při tvorbě rozhodovacího stromu je nutné specifikovat tzv. klasifikační atribut, který bude použit jiným způsobem než zbylé atributy (pomoci tohoto „specifického“ atributu se objekty rozdělí do tříd). Pro přiřazení této role atributu slouží operátor *Set Role*, který navážeme a nastavíme v něm, jaký atribut má být použit jako klasifikační. Parametr *name* nastavíme na jméno klasifikačního atributu a parametr *target role* nastavíme na *label*.
Dále navážeme operátor *Discretize* (*Discretize by Frequency*), kterým diskretizujeme klasifikační atribut, pokud tento atribut již není diskretizován (kategorický). Metoda pro tvorbu rozhodovacího stromu v RM totiž umí generovat stromy pouze pro kategoriální klasifikační atribut (zbylé atributy mohou být jak reálné, tak kategorické). Parametr *attribute filter* type nastavíme na hodnotu *single*, parametr *attribute* nastavíme na jméno klasifikačního atributu a parametr *number of bins* nastavíme na počet kategorií, do kterých rozdělíme hodnoty atributu (o vhodném počtu kategorií viz.

Na kolik kategorií diskretizovat reálné atributy?).

Jako poslední navážeme operátor *Decision Tree*, který vytvoří rozhodovací strom. Parametr *criterion* nastavíme na *information_gain* (informační zisk). Parametrem *minimal size for split* nastavíme, pro jakou nejmenší velikost uzlu může ještě metoda dělit uzel dále do dalších dvou větví. Parametrem *minimal leaf size* nastavíme minimální velikost listu (koncového uzlu – tzn. kolik minimálně objektů se nachází na koncovém uzlu). Parametrem *maximal depth* můžeme volit maximální hloubku stromu (kolik bude mít strom maximálně úrovní).
Pro volbu atributů, které zahrnout do analýzy a které ne (nyní jsou myšleny atributy, které se objeví v podmínkách na uzlech stromu, netýká se klasifikačního atributu), platí obdobná pravidla jako v kapitole Asociační pravidla (
Které atributy při vyhledávání asociačních pravidel použít a které ne?).

Poznámka: Metoda umí pracovat i s atributy s chybějícími údaji.

Použitá literatura

Plná citace zdroje (zdrojů)