Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Data mining a freeware

Diplomová práce

Autor: Bc. Lukáš Zasadil
Studijní obor: IM-2

Vedoucí práce: doc. RNDr. Hana Skalská, CSc.

Hradec Králové 30. dubna 2021

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

……..

V Hradci Králové dne 30.4.2021 Lukáš Zasadil

Poděkování

Děkuji vedoucí diplomové práce doc. RNDr. Haně Skalské, CSc., za metodické vedení práce a cenné rady při jejím vypracování.

Anotace

Diplomová práce se zabývá problematikou data mining a softwarů, které se pro jeho realizaci využívají. V rámci teoretické části jsou nejprve vysvětleny pojmy, jež jsou zásadní pro pochopení dané problematiky. Dále jsou v této části představeny některé metody data miningu, osoby a jejich role ve sledované oblasti a kritéria pro hodnocení softwaru. V praktické části jsou představeny tři softwary, jimiž jsou Weka, RapidMiner a R, kde je napříč těmito programy popsáno uživatelské rozhrání a ukázána práce s nimi, nad datasety, které jsou v práci představeny. V závěru praktické části práce jsou jednotlivé softwary porovnány, a to dle specifikovaných kritérií.

Klíčová slova:

Data mining, freeware, software, metodologie

Annotation

This diploma thesis addresses issues regarding data mining and software that is used for its realization. Theoretical part explains concepts, that are necessary in order to understand concerned domain. This section also introduces data mining methods, individuals and their roles in this domain aswell as software evaluation criteria. Practical part introduces three software programs, on which not only the user interface is described, but also the work process is shown on each of them. At the end of the practical part of the work, the individual software are compared according to the specified criteria.

Keywords:

Data mining, freeware, software, methodology

Obsah

Úvod

S neutuchajícím technologickým vývojem lze pozorovat taktéž nárůst rychlosti, s níž počítačové systémy pracují. Jde o rychlost, s níž jsou schopny získávat, zpracovávat a celkově hromadit data. S tímto faktem se také pojí skutečnost, že neustále roste počet zdrojů, z nichž jsou data získávána, a s ním rovněž počet prostředků pro jejich získávání. Mezi zdroje, které mají za následek nárůst objemu dat, můžeme řadit například také zařízení pracující na principu internetu věcí, senzorové technologie a sociální sítě. Veškerý tento nárůst lze zaznamenat napříč odvětvími, jako je například bankovnictví či zdravotnictví. S veškerými zmíněnými nárůsty vzniká problematika zpracování velkého množství dat.

V teoretické části bude nejen představen pojem data mining, ale taktéž pojmy s ním související. Dále teoretická část obsahuje popis analytických metod data miningu, mezi které například spadá regresní analýza, metodik pro zpracování dat, jako je CRISP DM a další. V praktické části budou představeni tři zástupci softwaru pro data mining, vedle popisu jejich uživatelského prostředí bude přiblížena i práce s nimi na zvolených datasetech, jejichž příprava bude uskutečněna dle metodiky CRISP-DM. Rostoucí množství dat zvyšuje také náročnost rozhodování. Z tohoto důvodu bude v práci představen zástupce softwaru pro podporu rozhodování, kde v praktické části budou jeho prostřednictvím jednotlivé softwary pro data mining hodnoceny a komparovány podle navržených kritérií (v programu pro podporu rozhodování Criterium Decision Plus). Výběr kritérií a přiřazování jejich váhy se bude odvíjet od vhodnosti softwaru pro nového uživatele.

1 Literární rešerše

Data a jejich sběr hrály důležitou roli již ve vzdálené minulosti, čemuž napovídá i původ tohoto slova. Jedním z prvních, kdo ho použil, byl Eukleidés, v jehož knize s názvem „*Data*“ se objevilo řecké slovo „dedomena“, tedy dát [1]. V rámci jednotlivých profesních oblastí je pojem chápán jinak, avšak obecně charakterizuje rozličné symboly a znaky, jejichž význam je jasný teprve v daném kontextu. V rámci data miningu se taková získaná a nezpracovaná data označují jako data surová, jejich velká uskupení, která svým rozsahem nejsou zpracovatelná běžnými softwarovými prostředky, se nazývají big data [2].

Ta jsou dále zpracovávána, po svém zavedení do kontextu, do něhož spadají, se stávají více komplexními a z nich následně čerpáme informace. Spojením více informací o konkrétním subjektu nebo faktu získáváme znalost. Znalosti jako výstupy napomáhají činit rozhodnutí a řešit problémy, a to nejen ve sféře běžného lidského života, ale například i ve sféře profesní, kde jsou získávané znalosti užívány například v rozličných systémech na podporu rozhodování, kam je vkládají doménoví experti, tedy specialisté pro konkrétní profesní oblast [2].

1.1 Data mining

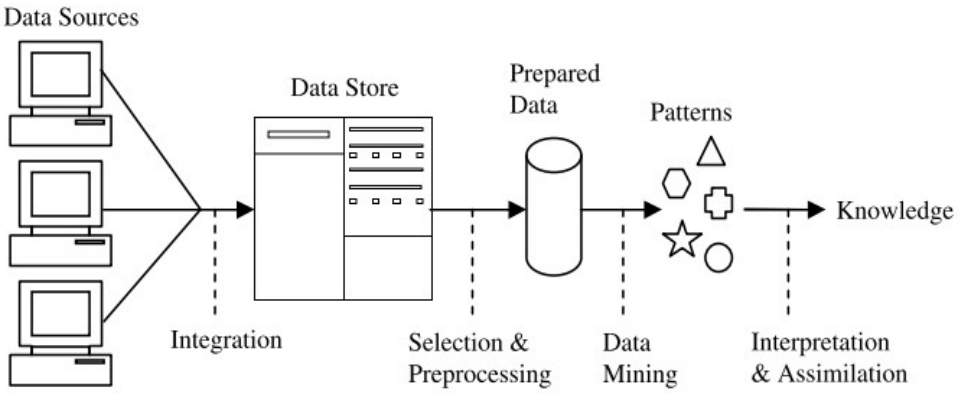
Pojem v doslovném překladu znamená dolování dat. Jeho přesná přijatelná definice neexistuje. Jednou z nich je následující: data mining je proces, jehož cílem je dát nad doménou smysl velkému množství dat, která jsou většinou nestrukturovaná. Uživatelé tohoto procesu jsou většinou doménoví experti pro danou oblast, kteří data nejen vlastní, získávají, ale také o nich nabývají prostřednictvím data miningu další znalosti [3].

Prvním klíčovým pojmem ve výše uvedené definici je pojem smysl. V dané definici říká, že nově získaná znalost z dat musí mít několik atributů, jimiž jsou: srozumitelnost, správnost, užitečnost a novost. Dalším pojmem je velké množství. Data mining se nezabývá množstvím dat, jež je možné zpracovat pomocí standardních technik, či dokonce manuálně. Data mining pracuje vždy s rozsáhlým množstvím dat [3].

Pro představu například americký operátor AT&T denně zprostředkovává přes 300 milionů hovorů pro 100 milionů lidí, přičemž veškeré informace ukládá do databáze, jejíž velikost představuje jednotky multiterabytů. Dalším příkladem může být americký obchodní řetězec Wal-Mart. Denně vyřizuje 21 milionů transakcí a informace o nich ukládá do databáze o velikosti jednotek terabytů. Dalším klíčovým pojmem je nestrukturovanost. Data, jež se v data miningu využívají, jsou většinou nestrukturovaná, zejména z důvodu jednoduchosti jejich získávání a pořizovací ceny [3]. V oblasti databází lze rozlišovat tři druhy dat z hlediska strukturovanosti. Strukturovaná data jsou nejlépe vyhledatelná a organizovatelná. Tato data lze uložit ve formě řádků a sloupců či jiné fixní, předdefinované struktury. Druhou skupinou jsou data nestrukturovaná, jež nemohou být uchována ve formě řádků a sloupců, či ve formě jiné předdefinované struktury. Jako příklad je možné uvést data senzorů a audio. Posledním typem jsou data, která jsou částečně strukturovaná. Jsou kombinací dvou předešlých, respektive mají některé konzistentní charakteristiky. Příkladem částečně strukturovaných dat je e-mail, jehož strukturovanou část představuje čas odeslání, jméno příjemce a předmět zprávy, jeho nestrukturovaná část představuje obsah zprávy [4]. Posledním klíčovým slovem uvedeným v definici je doména. Úspěch veškerých projektů data miningu se pojí se znalostí domény, v níž se pracuje [3]. Příkladem jiné definice data miningu může být vymezení tohoto pojmu jako procesu, v němž se aplikují výpočetní metody na velká množství dat, pro odhalení netriviálních a relevantních informací [5].

1.1.1 Získávání znalostí

Získávání znalostí je definováno jako proces, který je netriviální, v němž probíhá extrakce implicitních, předem neznámých a potenciálně užitečných informací z dat. V rámci tohoto procesu představuje data mining pouze jednu z částí, která je však ústřední [6].



Obrázek 1 Proces získávání znalostí

Na obrázku je zobrazen proces získávání znalostí. Tento proces se skládá z následujících kroků [6]:

- Získávání znalostí ze zdrojů – vybraná data jsou integrována a uložena do data storu, jež reprezentuje úložiště.
- Selekce dat a předzpracování – získaná data jsou segmentována na základě zvolených kritérií. Takto vybraná data jsou dále čištěna, dochází k očištění dat o záznamy, které nejsou validní, relevantní, a je taktéž upravován jejich formát a atributy.
- Data mining – nad připravenými daty jsou pomocí metod dolování dat vyhledávány vzory.
- Interpretace a začlenění vzorů – vzory jsou vhodně interpretované a jejich začleněním je tvořen výstup, jímž je znalost.

1.1.2 Historie data miningu

Historie data miningu zahrnuje mnoho dějinných milníků. Prvním z těch zásadních je zveřejnění článku Thomase Bayese v roce 1763, v němž jako autor popisoval Bayesův teorém, zabývající se podmíněnou pravděpodobností jevu ve vztahu k opačné podmíněné pravděpodobnosti. Následným milníkem je aplikace regresní analýzy, kterou uplatnili v roce 1805 Adrien-Marie Legendre a Carl Friedrich Gauss. Tento milník je významný zejména z toho důvodu, že regrese je považována za jeden z klíčových nástrojů v data miningu. V roce 1936 byl v článku poprvé představen nápad univerzálního stroje, schopného vykonávat početní úkony. Dnešní výpočetní zařízení

následují vizi Alana Turinga, uvedenou ve zmíněném článku. V rámci další historie vznikalo mnoho vědeckých prací a firem, jež se soustředily na takové oblasti, které se s data miningem pojí [7].

Vyvíjely se rovněž databáze, výpočetní technika a další technologie, které postupně umožňovaly pracovat s většími objemy dat. Tento vývoj poskytl možnost začít na přelomu 70. a 80. let 20. století využívat metody data miningu. Jeho první využití spočívalo zejména ve vyhledávání velkých datových souborů, konkrétně souvztažností v nich. Využití v rámci dalších let vzrůstalo zejména od vzniku metod, pomocí nichž je možné se vyvarovat korelacím, které jsou nesprávné. V roce 2001 byl pojem datová věda představen jako samostatná disciplína, jež zahrnuje i poznatky z matematiky, statistiky a data miningu, který je její součástí. V roce 2003 se daty řízený přístup ukázal jako vhodný i v oblasti sportu. Tým Oakland Athletics tehdy sestavil své mužstvo z nedoceněných levných hráčů, což ho přivedlo až do play-off mezi lety 2002 a 2003. V posledních letech získává vedle data miningu popularitu také hluboké učení, které je rovněž součástí datové vědy. O růstu zájmu a významu celé vědy svědčí také fakt, že i Bílý dům má prvního hlavního vědeckého pracovníka, zabývajícího se datovou vědou [7].

1.1.3 Typy zdrojů dat pro data mining

V procesu získávání znalostí jsou data získávána z rozličných zdrojů, kde jsou dále integrována do data storu. Jako zdroje dat rozeznáváme [8]:

Flat-file data

Flat-file data jsou taková data, která mají formu textovou či binární a jsou v takové struktuře, jež může být jednoduše extrahována algoritmy pro dolování dat. Tato data nemají mezi sebou žádné relace jako data v relační databázi. Tento typ dat lze prezentovat jako jednu tabulku, kde každý řádek představuje záznam, jehož sloupce jsou odděleny znakem, například čárkou. Tedy například soubory typu CSV představují flat file data [8].

Relační databáze

Relační databáze je definována jako kolekce dat organizovaných v tabulkách. Tyto databáze mají své fyzické a logické schéma. Fyzické schéma definuje strukturu tabulek, vztahy mezi tabulkami popisuje schéma logické [8].

Datový sklad

Datový sklad je typem relační databáze, kde se shrnují data z rozličných zdrojů, nad nimiž je možné vykonávat dotazy. Může spojovat více databází dohromady, nejedná se tedy například pouze o databáze relační [8].

Transakční databáze

Transakční databáze představují kolekce dat, jež jsou identifikovány prostřednictvím data, času či jiným způsobem, který reprezentuje databázovou transakci. Takové databáze umožňují vrátit změny jednotlivých transakcí. Transakční databáze se využívají například v bankovnictví, transakce jako jednotka práce modifikuje či přistupuje k obsahu databáze. Jednotlivé transakce mají vlastnosti ACID. Jsou tedy [8]:

- **atomické** – jednotlivé operace jsou dále nedělitelné;
- **konzistentní** – operacemi nejsou porušena integritní omezení;
- **izolované** – více transakcí může probíhat současně a neovlivňují se;
- **trvalé** – po dokončení transakce jsou utvořené změny trvalé.

Multimediální databáze

Multimediální databáze obsahují audio- a videosoubory, obrázky a textová média. Slouží pro uložení komplexních informací ve specifickém formátu [8].

Prostorové databáze

Prostorové databáze ukládají geografické informace ve formě souřadnic, topologií, čar, polygonů a dalších [8].

Databáze časových řad

Obsahem databáze časových řad jsou data o přihlášení uživatelů, burzovní údaje a další. Pole čísel zpracovává pod indexem, jako je například čas a datum [8].

WWW

WWW je nejvíce heterogenní repozitář, jelikož jeho data jsou různorodá a z rozličných zdrojů. Je taktéž dynamický ve smyslu měnící se velikosti dat, jež jsou v tomto repozitáři uložena, a to vzhledem k dynamice, s níž se mění [8].

1.1.4 Typy dat pro data mining

Typy dat, s nimiž se v úlohách data miningu nejčastěji setkáváme, odrážejí nejčastější typy úloh, mezi které se řadí úlohy pro efektivní prodej, cílený marketing a podporu zákazníků. Z tohoto hlediska rozlišujeme tři typy dat [9]:

Demografická data – Jsou hůře získatelná, většinou popisují proměnné, které jsou stálého charakteru, jako jsou například etnikum, státní příslušnost, počet dětí a dosažená úroveň vzdělání. Demografická data se taktéž vyznačují nízkou pořizovací cenou.

Data psychologická – Získávají se například dotazníky, výzkumy a pozorováními týkajícími se nákupního charakteru. Data jsou nepřesná z důvodu jejich proměnlivosti v čase a ovlivnitelnosti zejména momentálním stavem subjektů. Spadají sem například data týkající se životního stylu a hodnot.

Data behaviorální – V rámci behaviorálních dat jsou středem zájmu především ta týkající se nákupů, činnosti zákaznických služeb a rizikovosti pojištění.

1.1.5 Metodiky zpracování dat

Metodiky poskytují jednotný rámec pro řešení různých úloh z oblasti získávání znalostí. Výhoda ve využívání metodik spočívá v možnosti sdílet a přenášet zkušenost z úspěšných projektů, také umožňuje řešit úlohy nabývání znalostí rychleji, efektivněji a spolehlivěji, s nižšími náklady. Některé metodiky vznikly jako produkt komerčních společností, jiné ve spolupráci výzkumných a komerčních institucí [10].

Metodika SEMMA

Za vznikem metodiky SEMMA stojí firma SAS Institute INC., která ji vyvinula pro své vlastní účely. Metodika se skládá z 5 kroků, přičemž jejich jednotlivé názvy stojí za názvem celé metodiky [10].

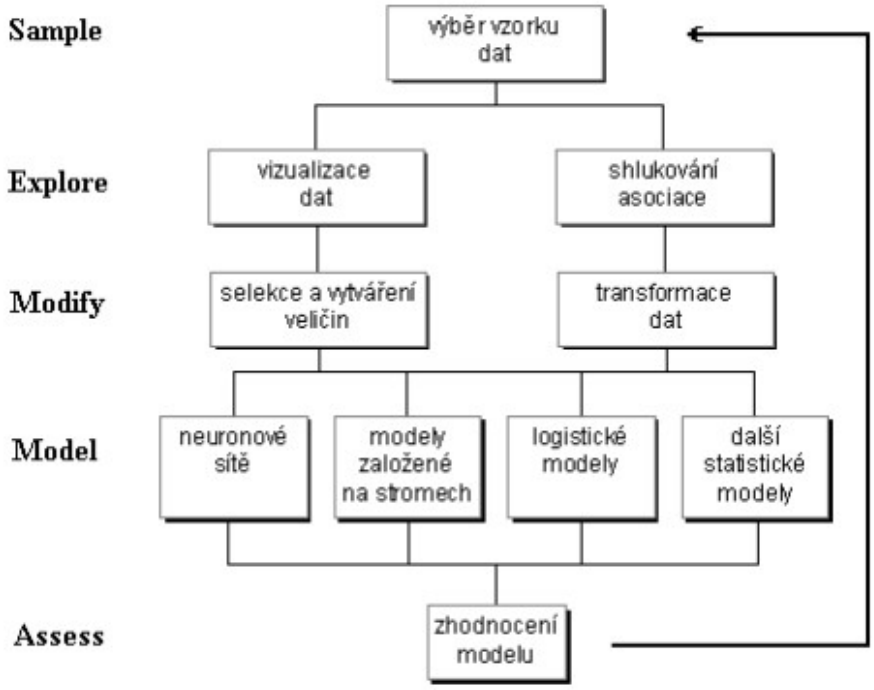
Sample – V tomto kroku dochází k výběru vhodných objektů z velkých datových souborů. Z vybraných objektů je utvářen model, který je poté testován [10].

Explore – Zde dochází k prozkoumání datového souboru, jenž je popisován identifikací významných proměnných, jsou vypočítávány popisné statistiky a další [10].

Modify – Příprava dat pro modelování – skládá se z kroků, jako jsou identifikace odlehlých pozorování, transformace stávajících proměnných, nahrazení chybějících hodnot a další [10].

Model – Krok spočívá ve vytvoření modelu pomocí dataminingových metod (například prostřednictvím regresních analýz a rozhodovacích stromů) [10].

Assess – Vyhodnocuje úspěšnost modelu a interpretuje konečné výsledky [10].

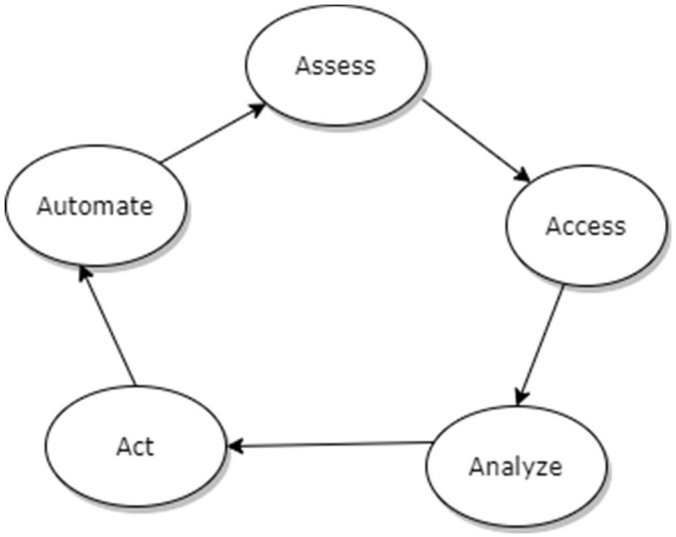


Obrázek 2 Schéma SEMMA

Metodika 5A

Metodiku 5A navrhla firma SPSS, dnes známá jako IBM. Název metodiky je opět odvozen od názvů jednotlivých kroků [10]:

- **Assess** – stanovení cílů projektu, posouzení potřeb [10];
- **Access** – příprava a shromáždění dat [10];
- **Analyze** – aplikace vybraných data miningových metod, přeměna dat na informace a znalosti [10];
- **Act** – přeměna znalostí do jednoznačné a srozumitelné podoby, tak aby bylo možné je akčně použít [10];
- **Automate –** převedení výsledků analýzy do praxe [10].



Obrázek 3 Schéma metodiky 5A

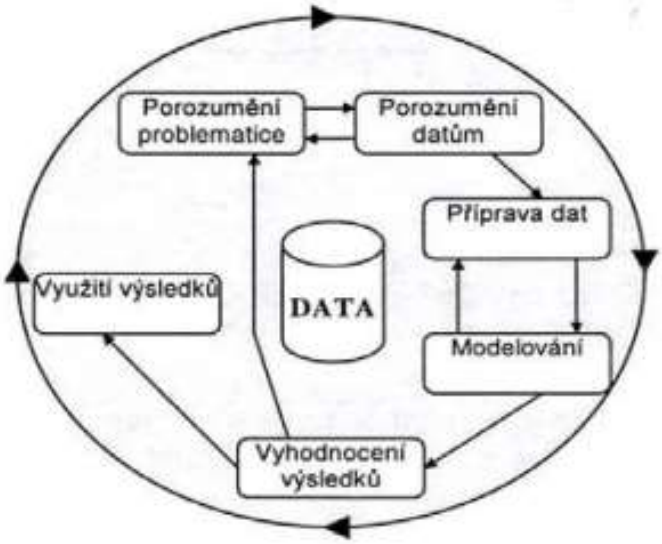
Metodika CRISP-DM

Metodika CRISP-DM (CRoss-Industry Standard Process for Data Mining) vznikla v rámci evropského výzkumného projektu, a to s cílem navrhnout univerzální postup, použitelný v nejrůznějších komerčních aplikacích. Na vzniku metodiky se podílely čtyři společnosti, jimiž jsou konkrétně NCR, DaimlerChrysler, ISL a OHRA. Jednotlivé společnosti vycházely při řešení projektů data miningu z vlastních zkušeností [10].

Dle této metodiky životní cyklus procesu získávání znalosti prochází šesti fázemi, jejichž pořadí není pevně dáno a výsledek předchozího kroku ovlivňuje vlnu kroků následujících. Metodika je charakteristická cyklickou povahou. Jednotlivé fáze jsou rozličně časově náročné, obecně se však uvádí, že časově nejnáročnější jsou fáze týkající se porozumění problému a fáze přípravy dat [10].

Metodika CRISP-DM zahrnuje tyto fáze [10]:

- porozumění problematice;
- porozumění datům;
- příprava dat;
- modelování;
- vyhodnocení výsledků;
- využití výsledků.

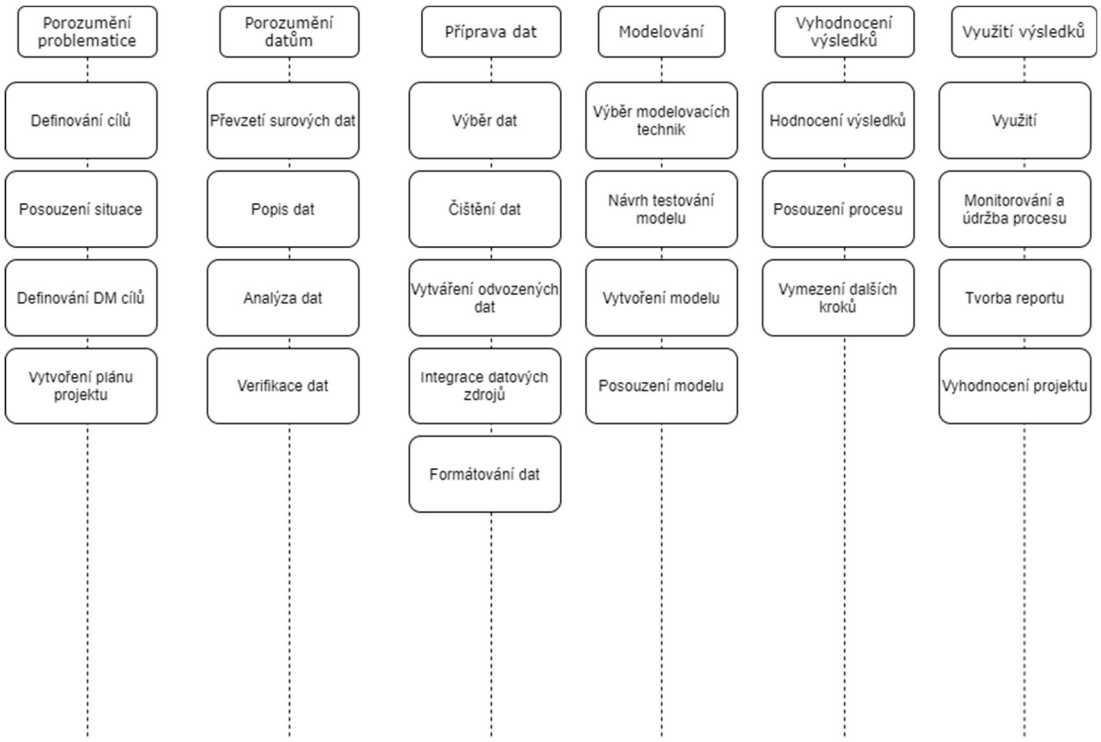


Obrázek 4 Schéma CRISP-DM

Porozumění problematice je průvodní fáze a má za úkol z manažerského hlediska pochopit hlavní cíle a požadavky projektu, jež jsou poté – opět z manažerského hlediska – transformovány do zadání úlohy pro dobývání znalostí. Fáze porozumění datům prvotně začíná sběrem dat a následně získáním průvodní představy o nich. Ta je utvářena například díky posouzení kvality dat, vytipováním podmnožin záznamů a zjištěním deskriptivních charakteristik. Následuje příprava dat, v níž se odehrávají činnosti, které vedou k vytvoření konečného datového souboru, z něhož jsou dolována data a který je zpracováván jednotlivými analytickými metodami. Data obsahují údaje důležité pro danou úlohu a mají takovou podobu, jaká je vyžadována vlastními analytickými algoritmy. Pod přípravu dat spadají činnosti jejich čištění, transformace, integrování a formátování. Činnosti jsou obvykle prováděny v libovolném pořadí a opakovaně [10].

Modelování je fáze, v níž jsou nasazovány algoritmy pro získávání znalostí, respektive analytické metody. V této fázi je vybírána nejvhodnější metoda pro řešení dané úlohy, obvykle je používáno více metod, jejich výsledky jsou kombinovány a metody iterativně opakovány. Některé metody mohou vyžadovat návrat k předchozímu kroku z důvodu nutnosti přípravy dat v závislosti na metodě [9]. Předposlední fází je vyhodnocení výsledků. Vyznačuje se tím, že byly získány znalosti, které se zdají být v pořádku. V této fázi se vyhodnocuje, zda bylo dosaženo výsledků specifikovaných v první fázi [10].

V poslední fázi, jíž je využití výsledků, jsou získané znalosti upravovány do podoby, která je pro zákazníka (respektive manažera) použitelná. Výsledkem v závislosti na typu úlohy může být například pouhé sepsání závěrečné zprávy, nebo naopak i hardwarová, softwarová či organizační implementace [10].



Obrázek 5 Přehled procesů CRISP-DM

Six Sigma

Six Sigma je metodika specializovaná na snížení chybovosti a jiných variabilit v činnostech společností. Six Sigma si našla uplatnění nejen v data miningu, ale především v průmyslu a jejím cílem je eliminace chyb a zefektivňování procesu. V jejím rámci je uplatňován systematický přístup, který využívá především práci s daty a fakty. Poprvé byla zavedena ve společnosti Motorola, kde prostřednictvím této metodiky byla posuzována kvalita, a to na základě směrodatných odchylek proměnlivosti výrobních procesů, nikoliv výrobků. Její název je odvozen od konceptu, který je v této metodice uplatňován [11].

Tento koncept říká, že defektní položka může být minimalizována údržbou šesti standardních odchylek mezi středem procesu a jeho horními a dolními

specifikovanými limity. Cílem metody Six Sigma je zvládat procesy takovým způsobem, že se v nich nebudou vyskytovat více než 3,4 chyby na jeden milion příležitostí. Metodologie je založená na struktuře DMAIC. Název této metody odpovídá jejím jednotlivým etapám. Jsou zde obsaženy etapy define, measure, analyze, improve a control. V prvním kroku jsou definovány cíle a specifika projektu s ohledem na požadavky zákazníka. V dalších krocích jsou získávána data, je uskutečněna jejich analýza, při níž jsou zkoumány vztahy příčin a následků. Zjištěná vylepšení jsou aplikována a v poslední fázi jsou následně kontrolovány eliminace jednotlivých odchylek, které se mohly vytvořit [11].

Obrázek 6 Schéma Six Sigma

1.1.6 Typy úloh

Úlohy, s nimiž se v oblasti data miningu běžně setkáváme, můžeme rozdělit do dvou základních skupin, a to na prediktivní a deskriptivní [12].

Prediktivní úlohy

Výstupem prediktivních úloh je model utvořený na základě dostupných dat. Tento model pomáhá v predikování hodnot cílové proměnné a k získávání znalostí o dané oblasti, z nichž data pocházejí. Rozlišujeme tyto konkrétní prediktivní úlohy [12]:

- klasifikace (Classification);
- predikce (Prediction);
- analýza časových řad (Time-series analysis).

Úlohy klasifikace spočívají v přiřazení třídy objektu na základě jeho atributů. Z celé řady objektů a jejich parametrů jsou vybrány charakteristické atributy pro jednotlivé třídy, každý nový objekt je následně přiřazen do jedné z tříd. Úkolem klasifikačních

úloh je co nejpřesnější přiřazení objektů do jednotlivých tříd. Klasifikace se využívá například v přímém marketingu ke snížení marketingových nákladů zaměřením na skupinu zákazníků, u nichž je pravděpodobné, že si zakoupí nový produkt. Například na základě dostupných demografických údajů a údajů o věku jsou zasílány propagační materiály s konkrétními produkty. Predikce slouží k předpovědi budoucích dat či doplnění dat chybějících. Úlohy predikce zahrnují vývoj modelu, který je založen na dostupných datech, a podle něj se získávají data další, jež jsou objektem zájmu [12].

Model může například předvídat příjem zaměstnance na základě vzdělání, zkušeností, demografických faktorů a pohlaví. Predikční analýza se používá také v případě lékařské diagnózy, detekce podvodů a tak podobně. U úlohách založených na analýze časových řad je časovou řadou myšlen sled událostí, kde další událost je vždy určena jednou nebo více událostmi předchozími. Časová řada odráží měřený proces a existují komponenty, které ovlivňují chování tohoto procesu. Prostřednictvím analýzy časových řad se odhalují užitečné vzory, trendy, pravidla a statistiky, významnou aplikací analýzy časových řad je například predikce na akciových trzích [12].

Deskriptivní úlohy

Úlohy deskriptivní se orientují na nalézání opakujících se vzorů v datech, kde z nalezených vzorů vyvozují nové významné informace o datech. Příkladem opakujících se vzorů může být identifikace produktů, jež jsou kupovány společně v rámci jednoho nákupního koše [12]. Obecně rozlišujeme následující deskriptivní úlohy [12] :

- Asociace (Association);
- Sumarizace (Summarization);
- Shlukování (Clustering).

Asociace objevuje přímé spojení mezi sadami položek, respektive identifkuje vztahy mezi objekty. Asociační analýza se užívá například v reklamě, v designu katalogů a v přímém marketingu. V praxi tak může například maloobchodník identifikovat produkty, které zákazníci běžně kupují společně. V případě, že zjistí, že

pivo a plenky se kupují většinou společně, může zaměřit marketing na prodej plenek s tím, že se podpoří i prodej piva [12].

Sumarizace, respektive shrnutí jako zobecnění dat představuje shrnuté relevantní údaje. Jejich výsledkem je menší sada dat, která poskytuje souhrnné informace o datech. Například pokud zákazník provede nákupy, lze tyto informace shrnout do celkových produktů a výdajů. Tyto souhrnné informace mohou být užitečné například pro prodejní týmy nebo týmy zabývající se vztahy se zákazníky pro analýzu chování zákazníků a nákupů. Data lze shrnovat z různých úhlů do různých úrovní abstrakce. Shlukování se používá pro identifikaci vzájemně podobných objektů. O podobnosti lze rozhodovat na základě řady faktorů. Například pojišťovací společnost může seskupovat zákazníky na základě věku, bydliště a příjmu. Tyto informace o skupině pomohou zákazníkům lépe porozumět a mohou následně usnadnit poskytování lépe přizpůsobených služeb. Podstatným rozdílem od klasifikace je to, že shlukování nepotřebuje mít definované třídy předem a objekt navíc může náležet současně do dvou, případně více různých shluků [12].

1.2 Analytické metody data miningu

Pro řešení úloh data miningu existuje řada metod, z nichž většina umožňuje implementaci prostřednictvím více druhů algoritmů. Konkrétní metoda a algoritmus jsou vybírány na základě povahy řešené úlohy. Podle disciplín rozlišujeme metody umělé inteligence (subsymbolické a symbolické) a metody statistické [10].

1.2.1 Metody statistické

Metody jsou charakteristické tím, že jsou teoreticky prozkoumané, zdůvodněné a léty praxe ověřené [10].

Mezi statistické metody patří [10]:

- regresní analýza;
- diskriminační analýza;
- shluková analýza;
- kontingenční tabulka.

Regresní analýza

V regresní analýze se popisuje vztah mezi proměnnými, kde je pouze jedna proměnná závislá a dále je jedna nebo více proměnných nezávislých, přičemž jejich hodnoty se odvíjejí od závislé [10].

Tuto závislost mezi proměnnými vyjadřuje regresní funkce, která obsahuje několik neznámých parametrů. Na základě počtu nezávislých proměnných rozlišujeme jednoduchou vícenásobnou regresi. Na základě lineárnosti a nelineárnosti parametrů rozeznáváme lineární a nelineární regresní model [10].

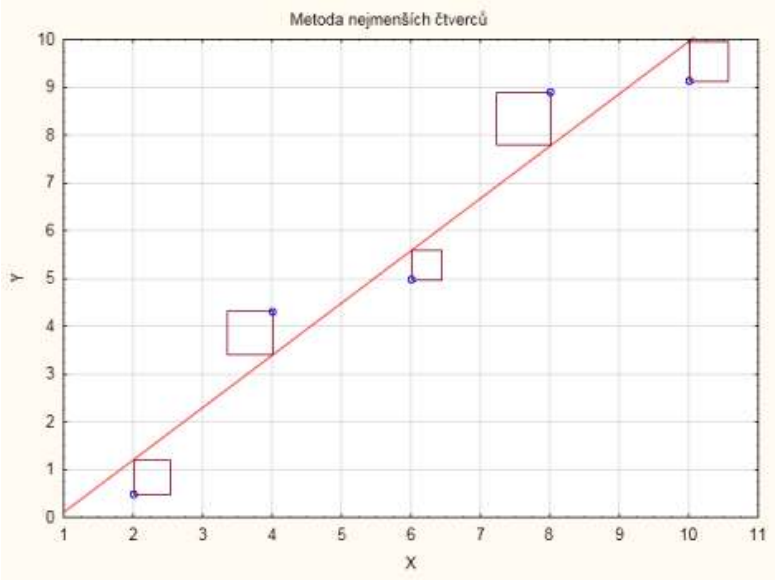
Nejobvyklejším typem je přímková regrese, která má následující tvar [10]:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Závislá proměnná je označena jako Y, nezávislá proměnná jako X, toto označení je obvyklé pro všechny regresní modely. Další proměnné jsou β_0 a β_1, což jsou parametry regresní rovnice, a ε značí náhodnou odchylku.

U nelineárních regresních funkcí se provádí odhad parametrů prostřednictvím numerických metod, jimiž se zlepšuje počáteční odhad hodnot parametrů, lineární regresní funkce se řeší prostřednictvím metody nejmenších čtverců.

Metoda nejmenších čtverců spočívá v minimalizaci součtu čtverců odchylek naměřených hodnot a hodnot vypočítaných [10].



Obrázek 7 Metoda nejmenších čtverců

Diskriminační analýza

Jako úlohu diskriminační analýzy lze chápat úlohu klasifikace příkladů do předem zadaných tříd. Příklady, respektive objekty jsou charakterizovány pozorovatelnými znaky. Cílem diskriminační analýzy je nalézt diskriminační funkci, která by na základě zadaných hodnot vektoru optimálně zařadila objekt do nějaké předem známé třídy.

S objekty se prostřednictvím metody diskriminační analýzy pracuje v rámci několika částí, a to v části analytické, diskriminační a klasifikační. V analytické části se pracuje s trénovací množinou, vzniklou na základě objektů z podmnožiny všech objektů. Předpokládá se, že na základě této množiny objektů je možné určit, z jaké třídy objekty pocházejí [10].

V diskriminační části je na základě pozorovaných znaků objektů z trénovací množiny sestaveno rozhodovací pravidlo, jehož prostřednictvím jsou roztříděny zbylé objekty, tedy objekty, které se nacházejí mimo trénovací množinu. Správně utvořené pravidlo minimalizuje pravděpodobnost, že objekt bude chybně klasifikován. V poslední části, tedy klasifikační, jsou dle sestaveného pravidla roztříděny zbylé objekty. Pravidlo je taktéž zhodnocení z hlediska predikčních schopností [10].

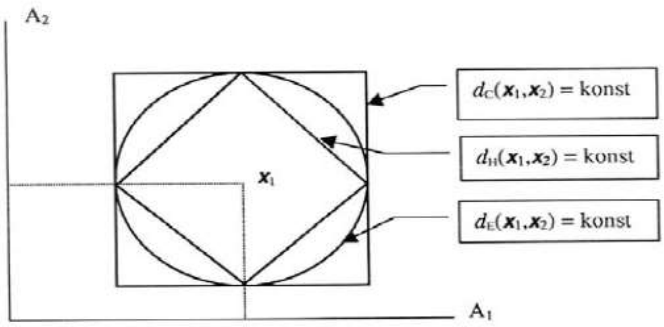
Shluková analýza

Shluková analýza je analýza, jejímž cílem je odpovědět na otázku, zda pozorované objekty je možné rozdělit do skupin (shluků) vzájemně blízkých případů tak, aby se jednotlivé shluky významně lišily [10].

Ve shlukové analýze dochází k měření vzdálenosti, která se používá k hodnocení podobnosti mezi objekty, používají se nejčastěji tři druhy měření vzdálenosti [10]:

- Hammingova vzdálenost;
- Eukleidova vzdálenost;
- Čebyševova vzdálenost.

Každá z těchto vzdáleností je určována jiným způsobem, jak napovídá graf.



Obrázek 8 Vzdálenosti shlukové analýzy

V grafu jsou znázorněny body, které mají stejnou vzdálenost od bodu x_1.

- d_c – znázorňuje bod Hammingovy vzdálenosti;
- d_h – znázorňuje bod Eukleidovy vzdálenosti;
- dε – znázorňuje bod Čebyševovy vzdálenosti.

Metody shlukové analýzy se dělí na hierarchické a nehierarchické, v závislosti na druhu se používají rozličné metody [10].

Nehierarchické shluky jsou takové, jež jsou definované v jednom kroku, respektive jejich první rozklad na podmnožiny se dále nedělí. Oproti nehierarchickým jsou hierarchické více členité. Na začátku tvoří každý objekt jeden shluk, shluky jsou postupně spojovány do té doby, než se dosáhne stavu, kdy jeden objekt obsahuje všechny shluky [10].

Kontingenční tabulka

Kontingenční tabulka umožňuje přehlednou vizualizaci vztahu mezi dvěma kategoriálními znaky, tedy znaky, které nejsou měřitelné. Průsečík řádku a sloupce označuje hodnotu, jež odpovídá oběma znakům tabulky (v řádcích se nachází kategorie jednoho znaku, ve sloupcích kategorie znaku druhého). Kontingenční tabulka poskytuje i ze součtů jednotlivých řádků a sloupců označující četnost jednotlivých kategorií [10].

	Y_1	Y_2	⋯	Y_s	Σ
X_1	n_{11}	n_{12}	⋯	n_{1s}	$n_{1.}$
X_2	n_{21}	n_{22}		n_{2s}	$n_{2.}$
⋮	⋮	⋮		⋮	⋮
X_r	n_{r1}	n_{r2}	⋯	n_{rs}	$n_{r.}$
Σ	$n_{.1}$	$n_{.2}$	⋯	$n_{.s}$	n

Obrázek 9 Kontingenční tabulka

Pro zjištění, zda mezi znaky existuje prokazatelný a výrazný vztah, se používá test, který je neparametrický a nazývá se chí-kvadrát. Dále se vypočítá testové kritérium a to se porovnává s kritickou hodnotou. Je-li hodnota testového kritéria následně menší než hodnota kritická, nulová hypotéza se zamítá [10].

1.2.2 Symbolické metody umělé inteligence

Symbolické metody umělé inteligence se orientují především spíš na vztahy logického typu než na matematické formule. Při užívání těchto metod se nalézají vztahy mezi daty a odhaluje se jejich struktura [10].

Mezi nejčastěji používané metody patří [10]:

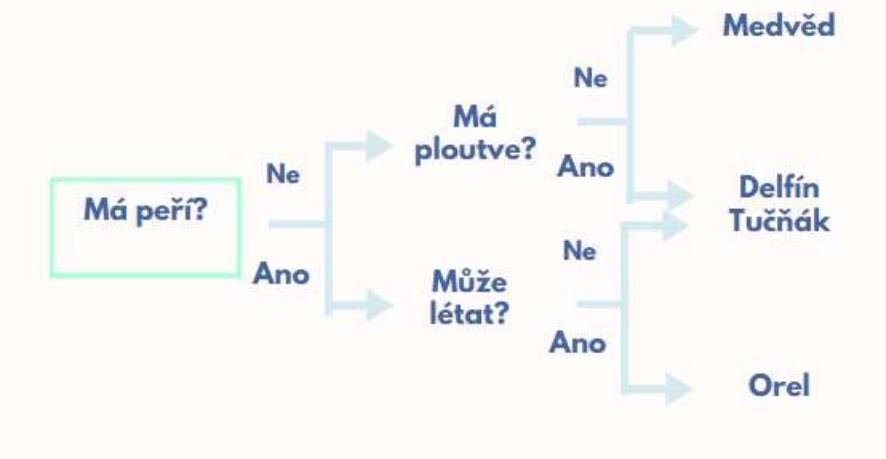
- asociační pravidla;
- rozhodovací stromy;
- rozhodovací pravidla.

Asociační pravidla

Asociační pravidla se používají pro hledání asociací mezi atributy objektů. Typickým příkladem jejich využití je analýza spotřebního koše, kde je cílem zjistit, jaké druhy zboží zákazníci kupují dohromady, respektive zjistit komplementární zboží [10].

Rozhodovací stromy

Stromy se skládají z uzlů, jež reprezentují určitou podmínku a hranu. Uzel bez příchozích hran je nazýván kořen, uzel s odchozími hranami se nazývá uzel vnitřní. Veškeré uzly kromě kořenu mají jednu hranu, která je příchozí. Posledním typem uzlu, s nímž se lze v rozhodovacích stromech setkat, je list. Jedná se o uzel, ze kterého nevycházejí žádné hrany. Při tvorbě rozhodovacích stromů je datový soubor postupně rozdělován do souborů [10].



Obrázek 10 Rozhodovací strom

Na obrázku je vyobrazen rozhodovací strom identifikace zvířete. V tomto případě je kořenem peří, dalším příkladem ploutve a létání jsou uzly rozhodovacího stromu, které jsou taktéž vnitřními uzly, následně jednotlivá zvířata představují listy. Výhodou rozhodovacích stromů je jasná názornost a přehlednost, nevýhodou je ale složitější zpracování [10].

Rozhodovací pravidla

Rozhodovací pravidlo se skládá ze dvou částí – první je předpoklad a druhou částí je třída. S rozhodovacími pravidly je možné se setkat ve všech programovacích jazycích [10]. Obecně je syntaxe vždy stejná a její vzor je následující.

IF / POKUD předpoklad THEN / POTÉ třída

Pokud bychom rozhodovací pravidla aplikovali na rozhodovací strom, poté podmínka IF vyjadřuje logický součin všech podmínek testů z kořene do listu stromu. Následně je pravidlu THEN přidělena hodnota. Nejznámější metodou pro tvoření rozhodovacích pravidel je metoda pokrývání množin. Obecný postup pro řešení úlohy touto metodou je takový, že je nalezeno pravidlo, které pokrývá příklady s danou hodnotou atributu. Dané příklady se mají z množiny odstranit, a pokud v množině zbývají nepokryté případy, opakuje se postup s hledáním pravidel. U složitějších rozhodovacích pravidel lze předpoklady kombinovat [10].

1.2.3 Subsymbolické metody umělé inteligence

Subsymbolické metody jsou takové, které dokážou přistupovat k řešení úkolů bez konkrétních reprezentací znalostí. Ve většině metod je učení chápané jako aproximace nějaké funkce. Charakteristické pro subsymbolické metody je, že získané znalosti bývají pro uživatele méně srozumitelné [10].

Do subsymbolických metod spadají například následující metody [10]:

- genetické algoritmy;
- bayesovské sítě;
- umělé neuronové sítě.

Genetické algoritmy

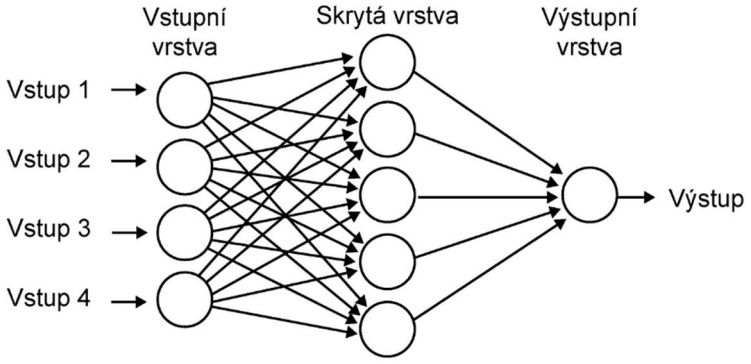
Genetické algoritmy vycházejí z Darwinovy evoluční teorie. V oblasti genetických algoritmů je několik důležitých pojmů, mezi něž patří [10]:

- **chromozóm** – řetězec informací, nesoucí vlastnosti a chování jedince;
- **gen** – nejmenší část chromozómu;
- **populace** – skupina jedinců popsaných chromozómy;
- **fitness hodnota** – číselné vyjádření kvality jedince.

Genetický algoritmus pracuje s náhodně vybranou počáteční populací, která se postupně zdokonaluje prostřednictvím operací, mezi něž patří selekce, křížení a mutace. Selekce je operace, jejímž prostřednictvím se z populace vyberou jedinci, kteří se mohou stát rodiči. Křížení je operace, při níž se vytvoří ze dvou rodičů potomek či potomci. Rozlišuje se jednobodové a dvoubodové křížení. U dvoubodového získá potomek prostřední část řetězce bitů jednoho rodiče a okraje řetězce druhého rodiče. U jednobodového křížení potomek vzniká tak, že začátek řetězce bitů je utvořen z jednoho rodiče a konec řetězce z rodiče druhého. Poslední operace, zvaná jako mutace, se projevuje jako změna bitů z 0 na 1 či opačně. Tato modifikace značí vlastnost v populaci, která se u žádného z jedinců doposud nevyskytla. Veškeré výše zmíněné operace jsou realizovány vždy nad celou populací a činnost algoritmu pokračuje do té doby, než se vyskytne minimálně jeden jedinec s požadovanými vlastnostmi [10].

Umělé neuronové sítě

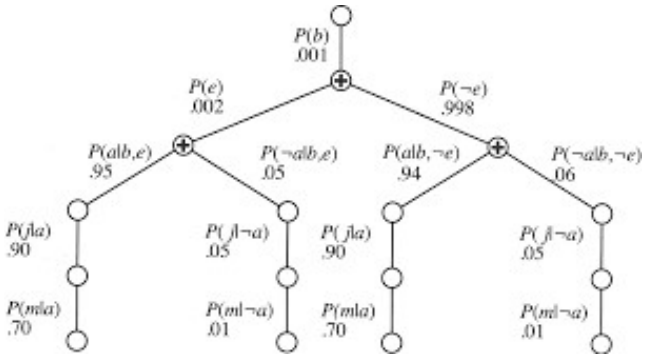
Umělé neuronové sítě byly inspirovány strukturou lidské nervové soustavy. Základním prvkem je neuron, tvořící základní výpočetní jednotku. Neurony jsou navzájem propojeny, předávají si signály. Jakmile neuron překročí určitou hranici signálů, začne působit svým výstupem na další neuron. Neuron může mít více vstupů, ale vždy jen jeden výstup. Jeho vstup může být tvořen vnější informací, nebo výstupem z jiného neuronu. Neurony se těmito propojeními uspořádávají do nervových sítí, čímž se zvyšuje jejich výpočetní síla. Neuronové sítě mají několik vrstev. Výstupy z poslední vrstvy jsou výstupy celé neuronové sítě, před ní je jedna nebo více vrstev skrytých. Neuronové sítě mají významnou charakteristickou vlastnost, a sice schopnost učit se řešit dané úlohy na základě znalostí, které jsou reprezentovány v podobě vah vazeb mezi neurony [13].



Obrázek 11 Uspořádání neuronové sítě

Bayesovské sítě

Bayesovská síť je pravděpodobnostní model, který je charakteristický využitím grafové reprezentace pravděpodobnostních vztahů mezi individuálními jevy. Uzel v rámci sítě odpovídá náhodné veličině, kde se veškeré veličiny vztahují k jednomu neznámému jevu, a hrana mezi uzly zobrazuje pravděpodobnostní závislost mezi veličinami. Bayesovská síť umožňuje tvořit model systémů, v nichž se pracuje se znalostmi zatíženými mírou neurčitosti. Využívají se například v medicíně při hledání příčin onemocnění a při volbě vhodné léčby. Znalost z databází se bayesovskou sítí dobývá dvěma způsoby, struktura sítě je navržena expertem a odvozují se pravděpodobnosti či se z dat odvozuje vedle pravděpodobností také struktura sítě [10].



Obrázek 12 Bayesovská síť

1.3 Osoby a jejich role v data miningu

Data mining jako proces vyžaduje v případě jednoduchých a jednostranných projektů pouze činnost data minera, který obstarává jednotlivé úlohy, avšak v případě větších projektů je vyžadována zainteresovanost více osob, a to jak na straně zadavatele úkolu, respektive firmy, která si projekt objednala, tak na straně dodavatele, tedy firmy, která je zodpovědná za dodání projektu. Mezi hlavní role z oblasti data miningu v rámci firmy, jež je zodpovědná za dodání projektu, můžeme zařadit [14]:

- zákazník;
- manažer;
- oborový specialista;
- informační technik;
- analytický pracovník;
- obchodní specialista.

1.3.1 Zákazník

Zákazníkem bývá nejčastěji firma. Zadání projektu je představováno jako jednání mezi manažery z jednotlivých firem, kteří spolu komunikují o zadání projektu, jeho průběhu a závěru a z důvodu získání nutných informací [14].

1.3.2 Manažer

Manažer zastává roli, která vyžaduje rozsáhlé všeobecné znalosti. Obvykle disponuje mělkými znalostmi, ale záběr jeho znalostí je široký, díky tomu má i vysoký potenciál pro samostatnou práci na projektu. Mezi jeho činnosti patří například dohlížení na chod projektu a jeho řízení, dohled na chod firmy, dohled nad podřízenými a jejich řízení, vyhledávání potenciálních zákazníků a komunikace s klienty [14].

1.3.3 Oborový specialista

Oborový specialista je člověk, který se vyzná v oboru a na nějž se firma zákazníka zaměřuje. Díky svým znalostem dokáže nejlépe aplikovat výsledky na firemní činnosti, čímž zajistí úspěšnost projektu. Výstupy analytických pracovníků umí vhodně aplikovat na procesy firmy tak, aby byla zajištěna maximální efektivita. Oborový specialista má přehled o aktuální situaci na trhu a cíle firmy umí interpretovat do data miningových cílů [14].

1.3.4 Informační technik

Informační technik je osoba zodpovědná za chod hardwaru a softwaru nutného pro práci na projektu. Zajišťuje, aby technologie ve firmě byly optimalizované pro chod softwaru využívaného při data miningu. Informační technik se taktéž do jisté míry stará o bezpečnostní politiku, zabezpečuje, aby nikdo kromě povolených pracovníků neměl k datům přístup [14].

1.3.5 Analytický pracovník

Analytický pracovník disponuje detailními znalostmi o technikách a metodách využívaných pro analýzu. Na výstupy analýzy následně navazuje prací s modely a daty. Analytický pracovník úzce komunikuje s obchodním, respektive oborovým specialistou, který výsledky analytického pracovníka umí efektivně aplikovat do chodu firmy [14].

1.4 Dokumentace softwarových požadavků

Dokumentace softwarových požadavků je etapou, která je nutná nejen při výběru softwaru, ale také jeho navrhování a utváření. Jedná se o dokument, či sadu dokumentů, které vyhraňují vlastnosti a požadované chování softwaru, jež odrážejí potřeby koncových uživatelů. Dokumentace se nesoustředí na technická řešení, ale specifikuje funkční a nefunkční požadavky systému: pod funkčními požadavky chápeme požadavky na funkční vlastnosti systému, jakými funkcemi má software disponovat. Nefunkční požadavky jsou takové, které se netýkají druhu poskytovaných funkcí, ale popisují, jakým způsobem program funguje. Dokumentace rozděluje problém, který software řeší, na menší, spravovatelné celky, slouží jako reference pro testování a validaci programu a taktéž usnadňuje komunikaci mezi zadavatelem pro tvorbu softwaru a zpracovatelem zakázky, jelikož jednoznačně definuje uživatelské potřeby. Všeobecně lze softwarové požadavky rozdělit do třech kategorií [15].

Význam

V rámci této kategorie definujeme prostředí, v němž bude systém používán, respektive je zde nastíněna problematika, kterou má daný software řešit. Definuje, jaký přínos bude software pro firmu a koncové uživatele mít [15].

Obecný popis

Obecný popis definuje, jaké funkce software přinese koncovému uživateli, včetně charakteristik daného uživatele. Vedle funkcí, jež software nabízí, definuje tato kategorie také omezení, a to nejen funkcí, ale i vstupů a dalších. (Těmito omezeními můžeme rozumět rozsah hodnot, dostupnost systému a další.) Kategorie udává také požadavky na bezpečnost a spolehlivost, včetně výkonnosti systému [15].

Specifické požadavky

Nejrozšířenější a nejvíce specifickou kategorií jsou specifické požadavky. Mohou udávat nároky na vzhled uživatelského rozhraní, dokumentaci softwaru, přehlednost, intuitivnost, náročnost provádění jednotlivých operací, náročnost zaškolení, dostupnost nápovědy a další [15].

1.5 Programy pro podporu rozhodování

Programy pro podporu rozhodování využívají tzv. vícekriteriální analýzy variant, což je taková analýza, která je schopná vyhodnotit více společných, konfliktních kritérií. (Konfliktním kritériem je například cena/kvalita, s tím, že je velice nepravděpodobné, aby cenově nejlevnější řešení bylo tím nejkvalitnějším.) Vícekriteriální analýza předpokládá explicitní výčet všech variant. Jednotlivým volbám jsou přiřazovány váhy, které číselně vyjadřují důležitost konkrétních kritérií, přičemž čím je jeho váha číselně větší, tím významnější kritérium je. Určování hodnot těchto vah je většinou subjektivní, obecně však existuje několik metod pro jejich stanovení. Příkladem je metoda pořadí, bodovací a Fullerova [16].

V metodě pořadí se respektuje pořadí, v němž jsou kritéria preferována, celočíselné body vah jsou poté přiřazovány sestupně podle důležitosti, normalizace je dosaženo prostřednictvím vydělením vah součtem celkově přidělených bodů. Metoda bodovací je obdobou metody pořadí, avšak vyžaduje informaci o preferenci jednotlivých kritérií, kde se bodovým ohodnocením určuje, jak moc je dané kritérium preferováno. Normalizace je dosaženo stejným způsobem jako v předešlém případě. Fullerova metoda se užívá v případech velkého počtu kritérií, kde jejich počet komplikuje možnost jejich řádného hodnocení. V rámci této metody jsou hodnoceny vždy dvojice kritérií, kde se z této dvojice – prostřednictvím bodového ohodnocení – určuje kritérium významnější. Normalizace je poté dosaženo stejným způsobem jako v předchozích případech [16].

Jednou z technik, kterou tyto softwary pro utvoření rozhodnutí užívají, je analytický hierarchický proces. Tato metoda byla navržena již v 70. letech profesorem Thomasem L. Saatym. Spočívá v rozložení nestrukturované situace na jednodušší dílčí části, respektive úrovně. Nejvyšší úroveň představuje cíl analýzy, jehož hodnota je jedna a dělí se mezi prvky na úrovni druhé, která se člení na úrovně další. Typické úlohy obsahují tři úrovně, tou první je cíl analýzy, druhou představují kritéria a třetí hodnocené varianty, kde na všech těchto úrovních je použita Saatyho metoda kvantitativního párového porovnání. Porovnávají se kritéria a varianty mezi sebou, dle preference a váhy dané preference [16].

Druhou používanou technikou je technika SMART (z anglického Simple Multi-Attribute Rating Technique), která byla navržena v roce 1977. V této technice nejsou kritéria v hierarchii. Hodnoty jsou přiřazovány v drtivé většině případů napřímo. Například rychlost je udaná v kilometrech za hodinu a není vyjádřena přes jinou škálu. Obě tyto metody se liší technikou výpočtu, a tedy i použitými vzorci, technika AHP se používá především v případech, kdy se předpokládá, že alternativy jsou konečné. Oproti tomu SMART je více škálovatelný, navíc je možné alternativy přidat i později, a to zejména z důvodu přímého přiřazování hodnot [16].

2 Cíl práce

V rámci praktické části budou představeny tři softwary užívané v oblasti data miningu. Programy byly vybrány na základě ankety KDnuggets, kde respondenti volili nejužívanější software pro data mining. V rámci prvních čtyř příček se vedle Excelu umístily právě volené softwary, jimiž jsou RapidMiner, Weka a R [17]. Jedná se zejména o freewarové programy, kde daný druh licence umožňuje tyto programy používat bez zakoupení jakékoliv licence, a to v rozsahu stanoveném licenčními podmínkami daného programu. Ostatní druhy softwaru se označují jako shareware (kde je plná funkcionalita omezena v rámci časového okna, poté jsou funkce omezené, a to až do zakoupení licence) a proprietární software, pro jejichž užívání je často nutné zakoupení licence a jejichž zdrojový kód není veřejně přístupný.

3 Metodologie

Jednotlivé softwary budou představeny ze všech úhlů a následně bude v závěru použita metodika komparace, kde budou jednotlivé programy komparovány mezi sebou dle stanovených kritérií. Jejich váha bude dána zejména z úhlu pohledu vhodnosti softwaru pro začátečníka. Pro komparaci těchto softwarů bude použitý program Criterium Decision plus, respektive program utvořený na podporu rozhodování, který je dostupný na virtuální učebně UHK.

3.1 Praktická část

V praktické části budou představeny a popsány jednotlivé programy, s jednotlivými softwary bude taktéž utvořena ukázka, a to představení práce s datasetem. Použitý dataset se po instalaci nachází v adresáři programu Weka, konkrétně iris.arff . Datový soubor obsahuje 150 záznamů, na základě atributů je objekt přiřazen vždy do jedné z tříd. Konkrétně se rozeznávají 3 třídy, soubor obsahuje 50 záznamů pro každou z nich. Každý jednotlivý záznam má 5 atributů:

- Sepal length in cm – délka kalichu;
- Sepal width in cm – šířka kalichu;
- Petal length in cm – délka okvětního lístku;
- Petal width in cm – šířka okvětního lístku;
- Class – třída, která je předpovídána na základě atributů; v souboru jsou rozeznávány 3 třídy:
  - Iris Setosa;
  - Iris Versicolour;
  - Iris Virginca.

3.2 Weka

V následující části bude představen software pro data mining s názvem Weka, který byl vyvinut Univerzitou Waikato na Novém Zélandu. Název softwaru je akronym pro Waikati Envrionment for Knowledge Analysis, v češtině Waikado představuje prostředí pro analýzu znalostí [18].

Software je open source, jeho zdrojový kód, který je psaný v Javě, je tedy dostupný veřejnosti. Jeho vývoj probíhá od roku 1993 a jeho první implementace v Javě proběhla v roce 1997. Tento vývoj je v aktuální době završen verzí 3.8.3, jež byla vydána roku 2018. Software je volitelně stažený a dostupný na stránkách https://www.cs.waikato.ac.nz/~ml/weka/index.html. Možností pro stažení je v nabídce hned několik. V první řadě je nabízena varianta mezi stabilní a vývojářskou, respektive developerskou verzí. Jednotlivé verze v rámci stabilních verzí softwaru zajišťují takové zásahy do kódu softwaru, jež neporušují kompatibilitu s předchozími verzemi. V obou verzích softwaru je nabízeno stažení pro systém Windows, Linux a Mac OS. Součástí

všech verzí je i Java Virtual Machine, která je nutná pro spuštění softwaru, jelikož zpracovává mezikód tak, aby byl srozumitelný pro daný typ procesoru [18].

Při každém spuštění programu se zobrazí úvodní nabídka s výběrem jednotlivých modulů. Na výběr jsou následující moduly:

- Explorer;
- Experimenter;
- KnowledgeFlow;
- Workbench;
- Simple CLI.

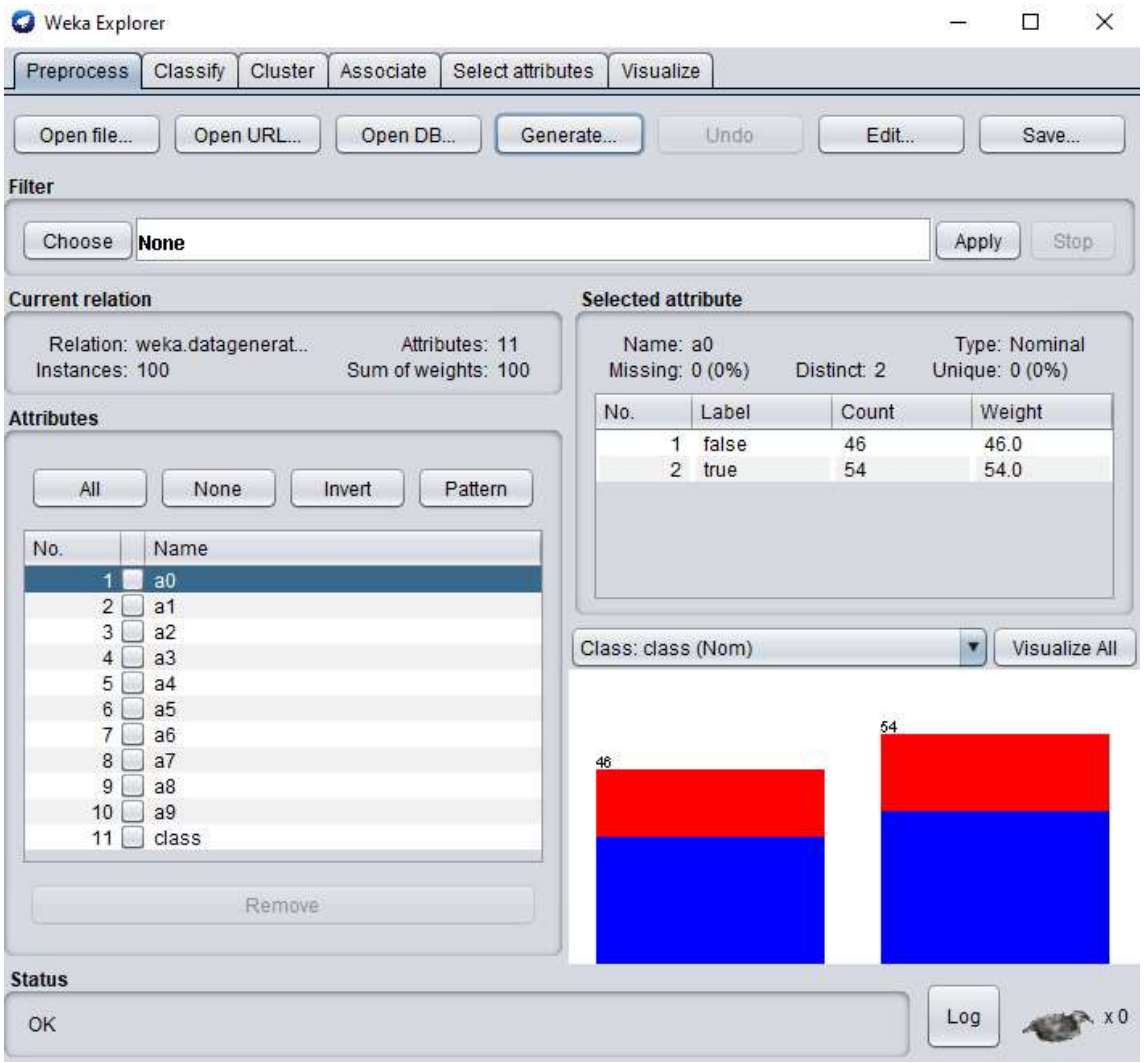
Vedle uvedených modulů jsou v záhlaví vertikálně i čtyři následující položky:

- Program – nabízí možnosti, jako jsou zobrazení využití paměti a zobrazení logů programu;
- Visualization – nabízí možnosti vizualizace dat, jako jsou například grafy a rozhodovací stromy;
- Tools – jsou zde nástroje, které umožňují dotazování do strukturovaných databází, nástroje pro práci s bayesovskými sítěmi a další;
- Help – záložka, jež odkazuje na návody k používání, příklady užití a další, které jsou umístěné na domovské stránce.

Obrázek 13 Úvodní nabídka Weka

3.2.1 Modul Explorer

V modulu Explorer je výchozí panel Preprocess. Na něj navazují další panely, jimiž jsou Classify, Cluster, Associate, Select attributes a Visualize. Na základě aktivního panelu se mění obsah okna programu. Identickou částí napříč aktivními panely je dolní sekce s polem Status, v němž je uživatel informován ohledně probíhající akce, a tlačítko Log, jež po zvolení zobrazí záznamy o provedených akcích programu.



Obrázek 14 Záložka Preprocess

Preprocess

Záložka Preprocess slouží především pro načtení dat. Ve vrchní části grafické prostředí softwaru umožňuje nahrání dat, kde lze jako zdroj dat vybrat soubory různého typu s řadami přípon, například:

- ARFF (například *.arff);
- C4.5 (například *.names);
- LibSVM;
- XRFF;
- BSI.

Vedle možností importu souborů s těmito příponami je možné data načíst i z URL adresy. Záložka Preprocess je úvodní záložkou modulu Explore, která po nahrání dat zpřístupní další záložky modulu.

V dalších částech softwaru se nachází pole „Current relation“, které obsahuje informace o relaci, jako jsou její název, počet záznamů a počet atributů jednotlivých záznamů. Pod výše zmíněným oknem se nachází pole „Attributes“, kde je vedle seznamu atributů také několik tlačítek pro práci s nimi.

Mezi tato tlačítka patří:

- All – vyselektuje všechny atributy;
- None – odznačí všechny atributy;
- Invert – provede inverzní výběr označených atributů;
- Pattern – umožňuje výběr atributů podle vzorů daných regulárními výrazy.

Vedle pole „Attributes“ se nachází pole „Selected attribute“. Zde se o vybraném attributu zobrazují informace, například počet unikátních a chybějících hodnot a typ atributu. Záložka také obsahuje pole „Filter“, s jehož pomocí je možné filtrovat nejen atributy, ale i záznamy.

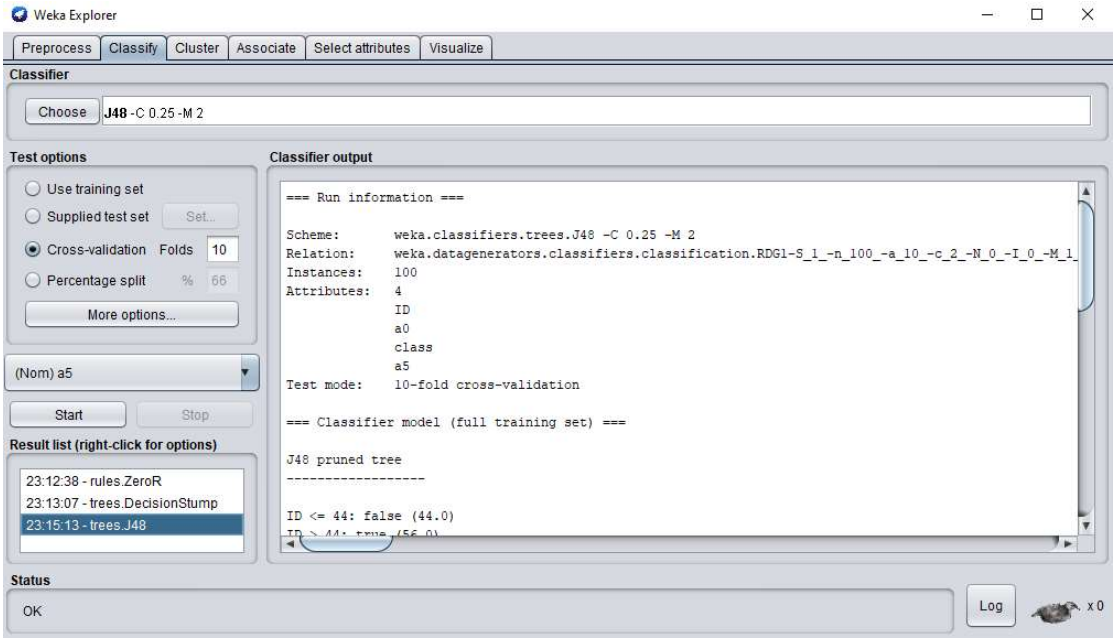
Classify

Záložka Classify nabízí metody umožňující klasifikaci na vybraném souboru dat. Program nabízí přes 100 klasifikačních pravidel. Nabízené algoritmy jsou rozloženy do několika skupin, jimiž jsou:

- Bayes;
- Functions;
- Lazy;
- Meta;
- Miscelanious;
- Rules;
- Trees.

Tato klasifikační pravidla jsou dostupná skrze pole „Clasifier“. Záložka dále obsahuje pole „Test options“, jež umožňuje nastavit vlastnosti testování.

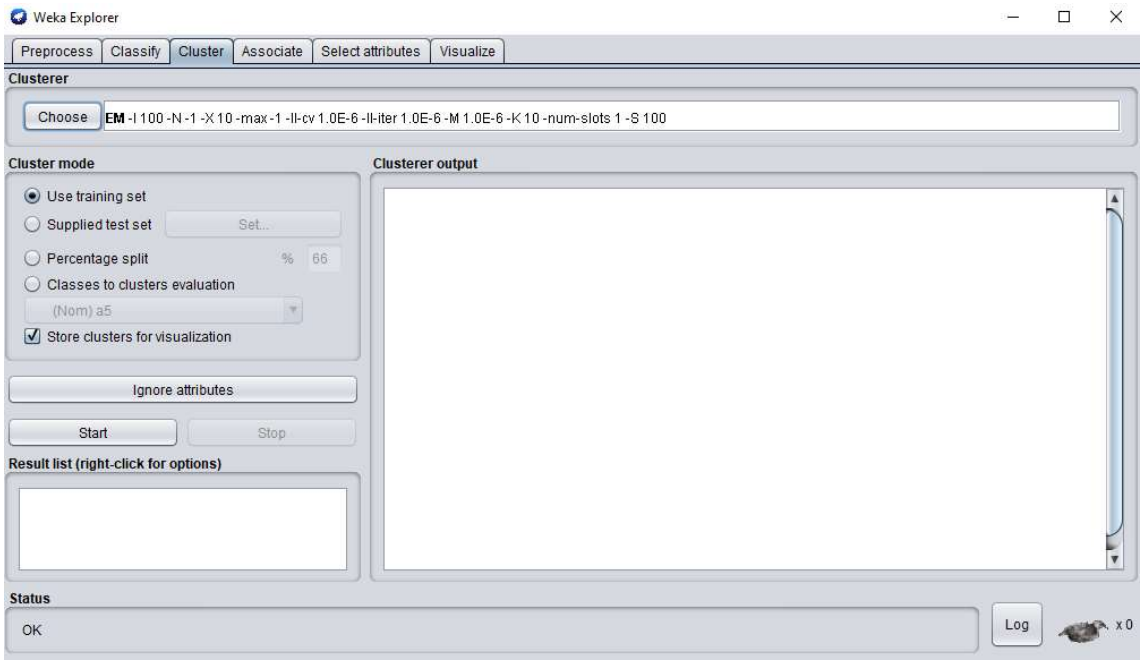
Mezi možné vlastnosti patří například „Percentage split“, který rozdělí datový soubor v nastaveném poměru, kdy první poměrná část je použita k vytvoření modelu a druhá k jeho testování, další z možností je „Cross validation“, jež rozdělí datový soubor na několik stejně velkých částí, kde testování probíhá mezi jednotlivými částmi (dle zvoleného klasifikačního pravidla se bude lišit přesnost modelu). V rámci klasifikace je defaultně používán poslední atribut z datového souboru, případně lze nastavit atribut jiný. Po spuštění procesu klasifikace tlačítkem Start je výsledek po dokončení procesu zobrazen v poli „Classifier output“. Posledním polem v této záložce je „Result list“, v jehož rámci jsou zobrazovány výsledky klasifikací, z nichž je možné činit řadu akcí, jako vybírat výsledky analýz, eventuálně zobrazit výsledky vizuálně a další.



Obrázek 15 Záložka Classify

Cluster

Záložka Cluster umožňuje tvořit shlukové analýzy. Je takřka identická se záložkou Classify, hlavní rozdíl je v tom, že namísto pravidel pro klasifikaci se vybírají pravidla pro klasifikaci, a to v poli „Clusterer“.



Obrázek 16 Záložka Cluster

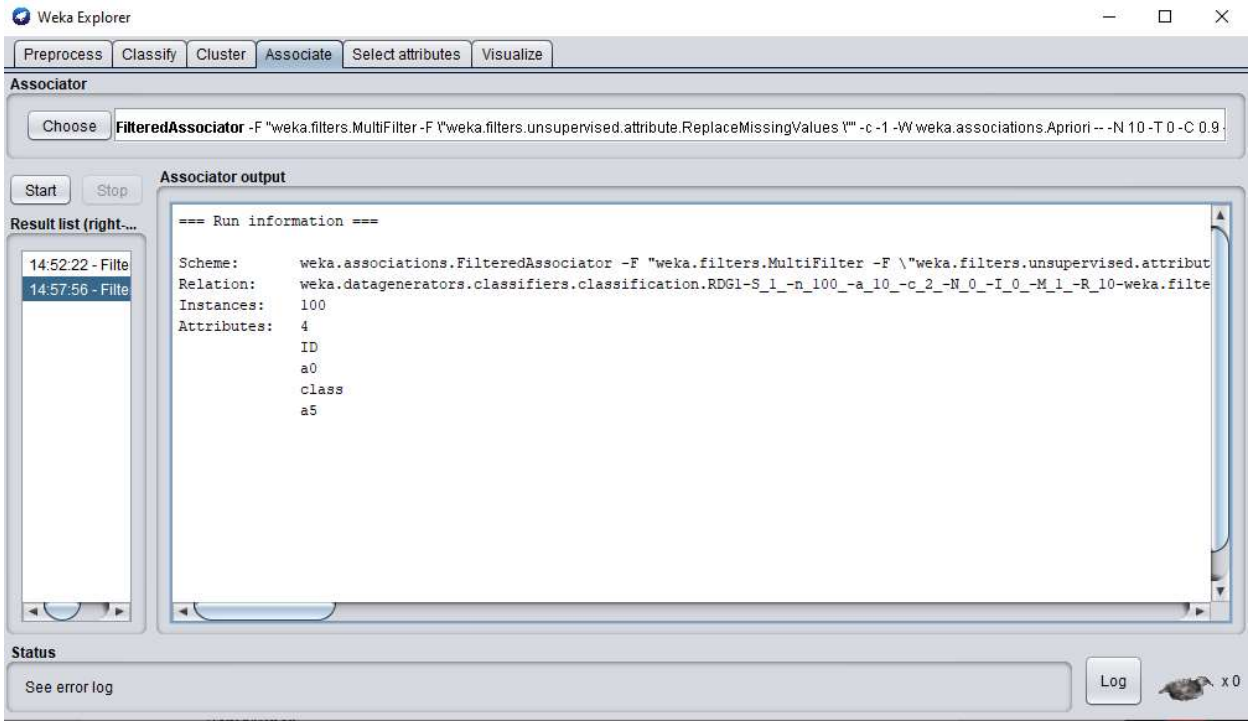
V poli Cluster mode jsou nabízeny možnosti testování, obdobné jako v předešlé záložce, avšak s tím podstatným rozdílem, že jsou aplikovány na shluky. Dále je v této části také možné nastavit několik možností pro shlukování:

- Je možné nastavit atribut, který třídu identifikuje.
- Je zde nabídka, zda po dokončení shlukování bude možné data vizualizovat (důvodem, proč je tato možnost explicitně nabízena, je, že při větším množství záznamů může být shluková analýza při následné vizualizaci náročná na výpočetní výkon, především paměť zařízení).
- Lze explicitně zvolit, které atributy budou ignorovány.

Pole „Clusterer output“ zobrazuje výsledek shlukové analýzy, stejně jako tomu bylo v záložce Classify. Obdobné je pole „Result list“, zobrazující výsledky jednotlivých analýz. Při pravém stisknutí tlačítka myši je vyvolána kontextová nabídka pro daný výsledek shlukové analýzy. Zde je řada nabídek, včetně nabídky pro vizualizaci výsledků, pokud byla povolena.

Associate

Associate slouží k sestavení asociačních pravidel. Záložka disponuje pouze třemi oblastmi. První z nich je „Associator“, umožňující volbu algoritmu, proces je následně spuštěn tlačítkem Start, kde je výsledek, respektive seznam nalezených asociačních pravidel zobrazen v poli „Associator output“. Posledním polem v této záložce je „Result list“, kde je k dispozici seznam výsledků. Pro každou instanci spuštění je daný výsledek zobrazen v seznamu.



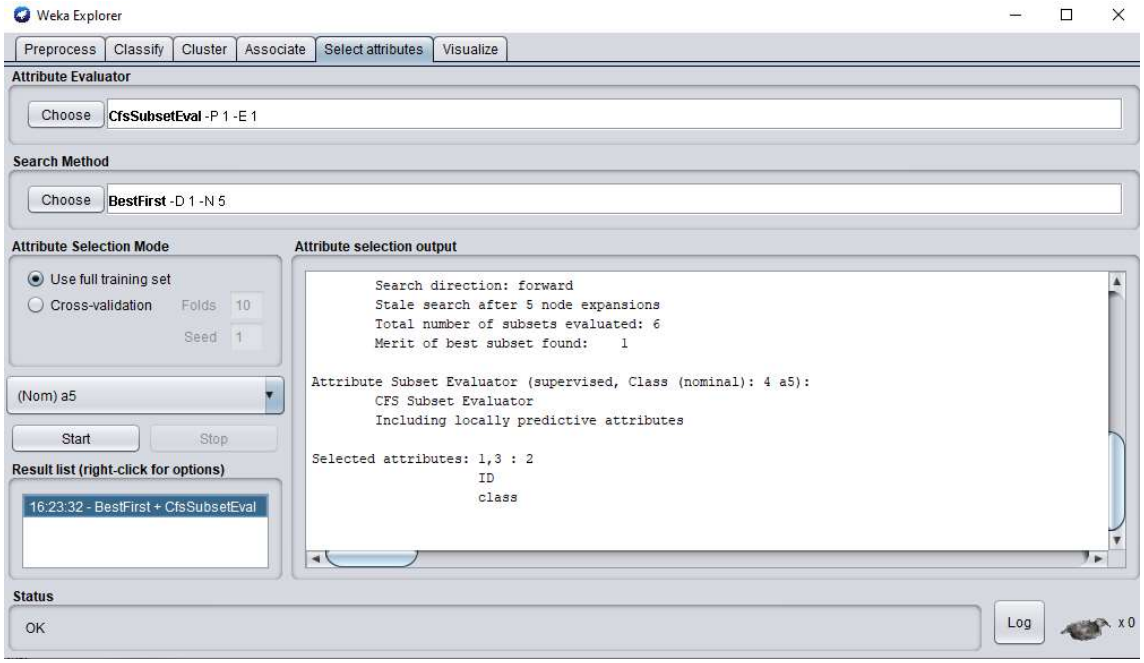
Obrázek 17 Záložka Associate

Select attributes

Záložka Select Attributes slouží ke zkoumání jednotlivých atributů v tom smyslu, jak moc jsou vhodné pro predikci, případně je možno atributy, které jsou nevýznamné, z predikce vynechat. Vynechání atributů má za výsledek rychlejší běh algoritmů, jejichž délka vykonání významně roste s množstvím záznamů a jejich atributů.

Záložka obsahuje pole „Attribute evaluator“, určující metodu pro posouzení vhodnosti jednotlivých atributů. Pole „Search method“ vymezuje, jakým způsobem budou vybírány vyhovující podmnožiny.

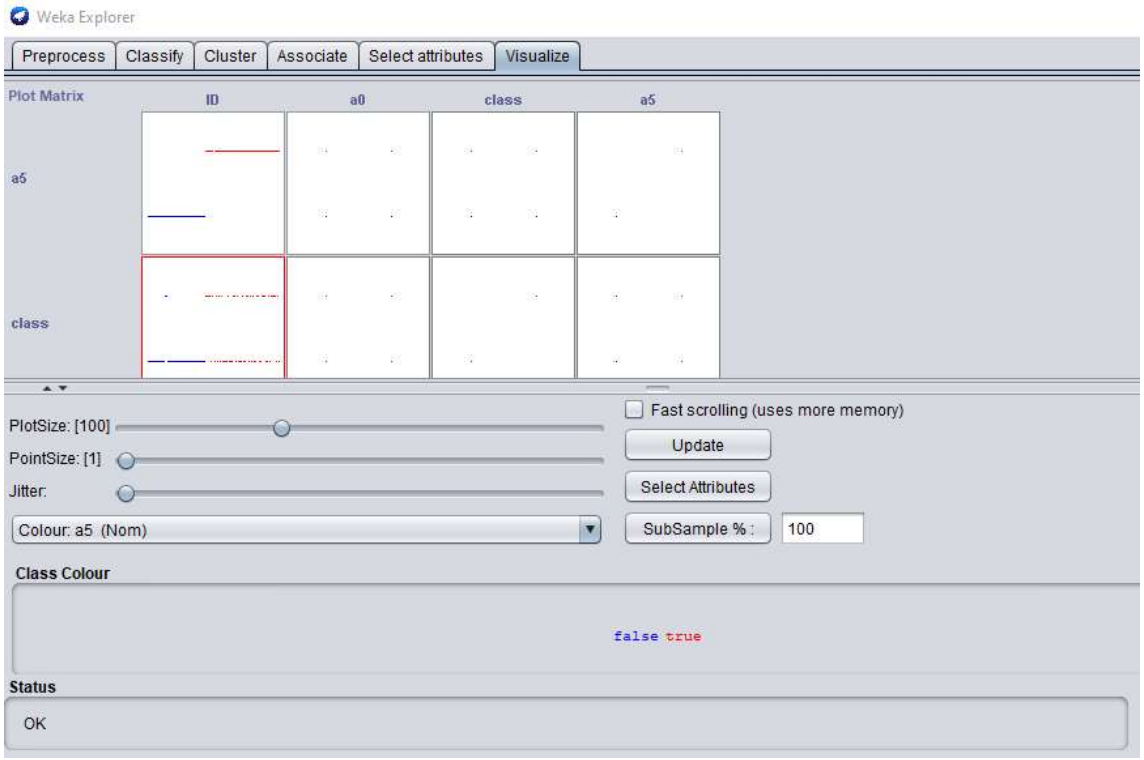
Oblast „Attribute Selection Mode“ určuje způsob ohodnocení atributů podmnožiny (dostupné možnosti jsou Use full training set a Cross-validation). Výsledek výběru atributů je zobrazen v poli „Attribute selection output“. Záložka obsahuje stejně jako předešlé záložky pole „Result list“, kde jsou po každé instanci vyvolané tlačítkem „Start“ uloženy výsledky.



Obrázek 18 Záložka Select attributes

Visualize

Další záložka umožňuje vizualizaci dat. Ta jsou vizualizována jako 2D grafy, osy grafu odpovídají hodnotám atributů z dat, která byla načtena v úvodní záložce „Preprocess“, body v grafu odpovídají testovým instancím. V rámci jednotlivých polí umožňuje záložka práce s grafem, včetně jeho zvětšení, vybrání atributů, jež v grafu budou, a další.



Obrázek 19 Záložka Visualize

Praktický příklad

V rámci praktického příkladu bude použit dataset zmíněný v úvodu praktické části. Po nahrání dat dialogové okno automaticky propíše atributy z datasetu, kde jsou dále zobrazeny obecné charakteristiky pro jednotlivé atributy, jako jsou minimum, maximum a další. V rámci příkladu bude použit klasifikátor DecisionTable, běh byl spuštěn s výchozím nastavením klasifikátoru, nicméně je možné upravit parametry klasifikátoru dle dokumentace Weky. Po zvolení se klasifikátor zobrazí v následujícím formátu:

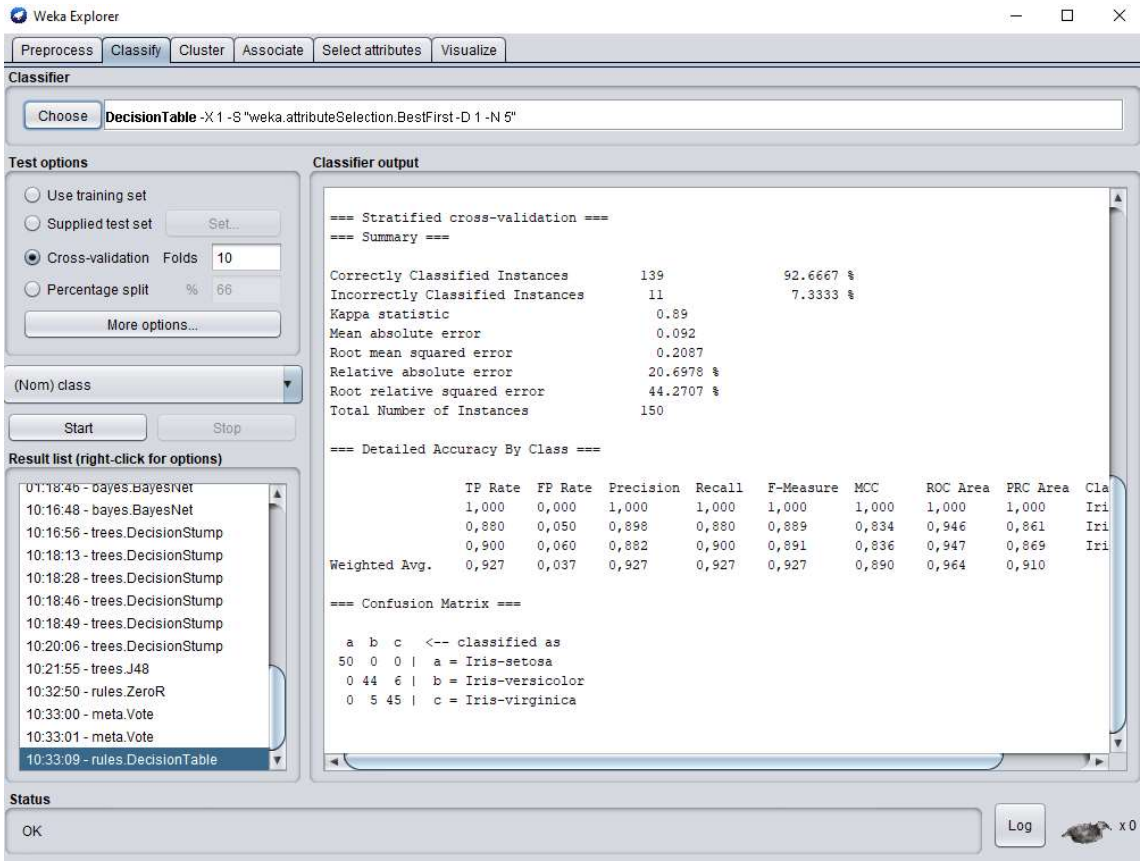
DecisionTable-X 1 -S „weka.attributeSelection.BestFirst -D 1 -N 5“

Většina parametrů souvisí s použitou vyhledávací, výchozí metodou BestFirst. Jednotlivé parametry dle dokumentace Weky značí [19]:

- X – Number of folds – na kolik podmnožin bude množina rozdělená;
- S – velikost vyhledávací cache pro evaluované podmnožiny;
- D – směr vyhledávání;
- N – počet nezlepšujících se jednotek, které budou uvažovány před ukončením vyhledávání.

Parametry S, D, N souvisejí s metodou BestFirst, což je vyhledávací metoda, jež prozkoumává grafy na základě nejslibnějších uzlů v grafu [20].

Klasifikátor DecisionTable určuje třídu na základě naplnění podmínek, které určují náležitost dané třídy.



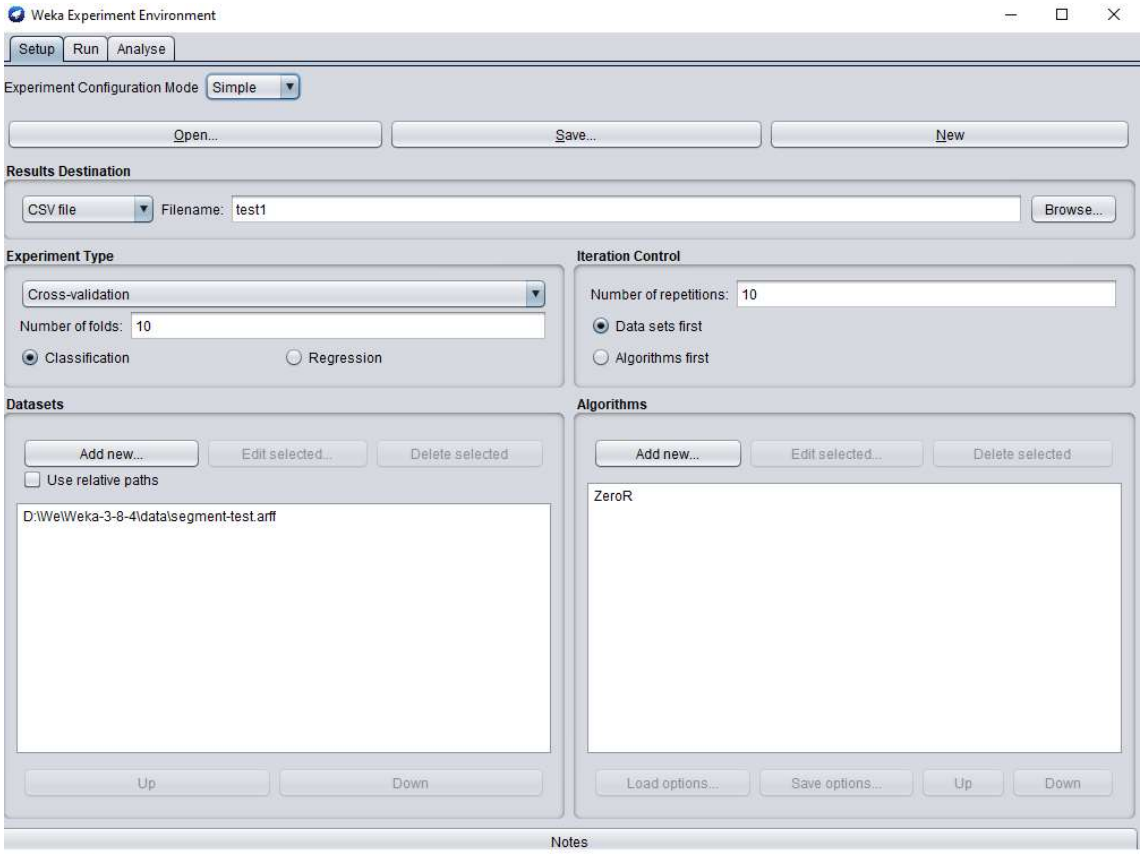
Obrázek 20 Výstup záložky Classify s datasetem

Dle výstupu je patrné, že bylo zařazeno 92,6 % případů, tedy 139 záznamů. Dalšími parametry jsou například RMSE neboli root-mean-square error (standardní odchylka residuí, jež představují vzdálenost od přímky lineární regrese), kde hodnota tohoto parametru je 0,2087.

3.2.2 Modul Experimenter

Modul Experimenter umožňuje uskutečňovat opakované experimenty, lze tedy například testovat, jak je algoritmus výkonný (výkonnost algoritmů se posuzuje nad stejnými daty). Modul obsahuje tři záložky, jimiž jsou Setup, Run a Analyse.

Záložka Setup je záložkou úvodní, jež zpřístupňuje ostatní záložky. Umožňuje spravovat nastavení experimentu, jako je počet iterací, použité algoritmy a další. V případě jednoduchého režimu modulu je možné srovnávat jen klasifikační a regresní algoritmy, ostatní experimenty se provádějí při zvolené možnosti „Advanced“, tedy pokročilé. Výsledky je možné ukládat do formátu ARFF, CSV anebo jako JDBC, kde výstupem je URL adresa, výstup můžeme eventuálně používat v rámci programovatelného rozhraní dalšího programu. Po úvodním nastavení experimentu navazuje záložka Run. V rámci této záložky dochází k zahájení experimentu. V případě jakékoliv chyby je v průběhu experimentu v této záložce uživatel informován, stejně jako je informován o počátku a skončení průběhu experimentu.



Obrázek 21 Modul Experimenter

Poslední ze záložek, záložka Analyse, slouží pro analýzu výsledků nejen aktuálně uskutečněných experimentů (při zvolení možnosti „Experiment“), ale také experimentů, jež jsou zvoleny prostřednictvím možnosti „File“ nebo „Database“. Je tedy nejen možné pracovat se soubory jako takovými, ale taktéž s výstupy databází.

V oblasti „Configure test“ se nachází řada možností, jež specifikují možnosti analýzy. Je tak možné například zvolit druh testu (jako například párový T-test), či zvolit, dle jakého údaje bude výkonnost algoritmů porovnávána. Existuje řada metrik pro porovnání výkonnosti těchto algoritmů, mezi ně patří například F-test, podíl správně a podíl špatně zařazených prvků.

Po požadovaném nastavení lze testovat výsledky experimentu prostřednictvím volby „Perform test“, výsledky se následně zobrazí v poli „Test ouput“. Každá instance testu je uložitelná prostřednictvím volby „Save output“, každý výsledek se rovněž během testu propíše do seznamu výsledků „Result list“.

3.2.3 KnowledgeFlow

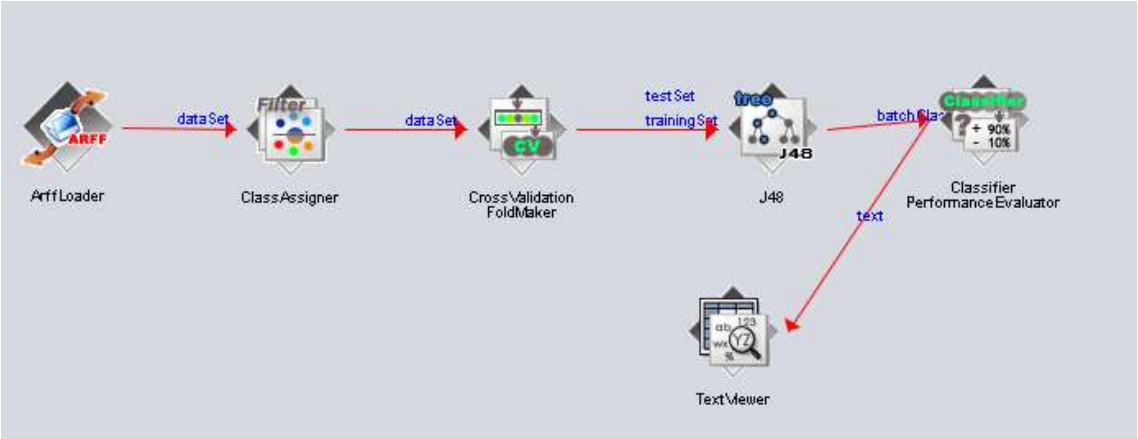
Modul KnowledgeFlow je nástrojem pro tvorbu návrhu pro „machine learning pipeline“. Tento termín označuje návrh takové úpravy dat, jež umožňuje, respektive usnadňuje automatizaci procesů strojového učení. Modul je velmi podobný modulu Explorer, co se uživatelského rozhraní týká. V rámci uživatelského rozhraní je nabízená četná řada komponent, vzájemně propojených tak, že tvoří tok, který – jak bylo popsáno výše – může usnadňovat automatizace procesů strojového učení.

Komponenty jsou nabízeny v řadě několika kategorií, můžeme tak například nalézt tyto kategorie:

- DataSources (zdroje dat) – komponenty reprezentují zdroje dat dle jejich formátu.
- Data Sinks (datové vany) – fungují jako data úložiště, jsou rozděleny na formáty stejně jako zdroje dat.
- Classifiers (klasifikátory).
- Flow (tok) – jsou zde komponenty, které upravují tok v rámci „diagramu“. Tyto komponenty například mohou získávat data z experimentu, nastavovat proměnné a další.
- Visualization (vizualizace).
- Filters (filtry) – filtry je v rámci diagramu možné i řetězit.

Příklad KnowledgeFlow

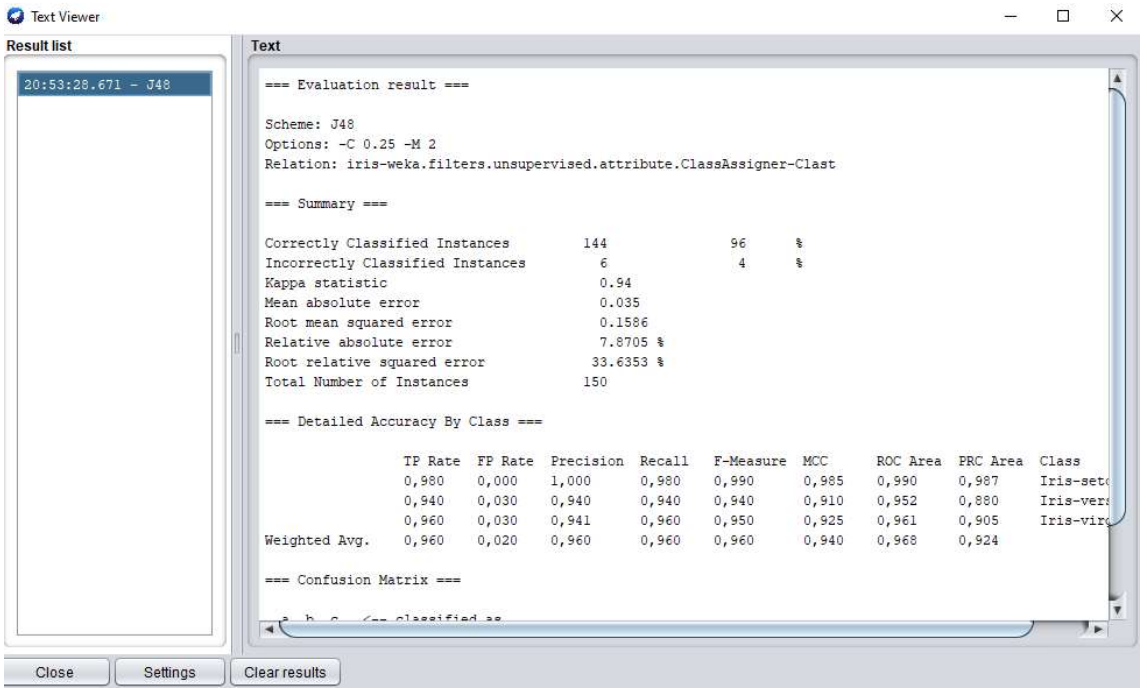
Na základě dostupných prvků byl sestaven následující model:



Obrázek 22 Model pro klasifikace datasetu Iris

Komponenta ArffLoader odkazuje na lokální úložiště v počítači, kde je uložen dataset, s kterým se pracuje v rámci modelu. Při propojování dalších komponent je vždy po vložení následující komponenty označena komponenta předchozí a z kontextové nabídky následně zvolena možnost dataSet. Po zvolení této možnosti je zobrazena šipka, jež slouží k propojení komponent. Další komponenta, ClassAssigner, slouží pro výběr třídy pro klasifikaci (při nezměněných parametrech je vybrán poslední sloupec). Další komponenta, CrossValidationFoldMaker, určuje způsob testování a J48 způsob, respektive algoritmus klasifikace. Komponenta PerformanceEvaluator slouží pro vyhodnocení výkonnosti klasifikátoru a poslední komponenta – TextViewer – pro zobrazení textového výstupu.

Po spuštění v levém horním rohu je možné textový výstup zobrazit. Zobrazuje se v následující formě dle obrázku.



Obrázek 23 Textový výstup z modelu

Výstup informuje o mnoha náležitostech, jako je počet správně klasifikovaných případů, kterých bylo 144, což činí 96 % z celkového počtu záznamů.

3.2.4 Modul Workbench

Workbench kombinuje ve svém grafickém prostředí všechny předešlé moduly do jednoho rozhraní, což je zároveň jeho hlavní výhodou. Vyšší nepřehlednost je vykoupena faktem, že není nutné přeskakovat mezi jednotlivými moduly a záložkami ve více oknech. V rámci jednoho okna je tak možné využívat modul Explorer a prostředí Experiment Environment pro tvoření experimentů. Mezi prostředími se přepíná v záhlaví grafického rozhraní.

3.2.5 Modul Simple CLI

Tento modul představuje jednoduchou příkazovou řádku, jak je patrné z jeho názvu Simple CLI (neboli Simple command line interface). Stejně jako ostatní příkazové řádky i tento umožňuje automatické doplňování příkazů uživatele prostřednictvím tabulátoru.

Platí zde i obecné příkazy, jako je například help, pro výpis dostupných příkazů:

- capacilities <classname> <args>
  - Vypíše metody dané třídy
- Cls
  - Smaže veškerý text z příkazové řádky
- echo mag
  - Vypíše zprávu
- Exit
  - Vypne příkazovou řádku
- help [command1] [command2]
  - Vypíše help pro specifický příkaz nebo příkazy
- History
  - Vypíše historii zadaných příkazů pro danou relaci
- java <classname> <args>
  - Vyvolá třídu se vstupními argumenty
- Kill
  - Zastaví běžící proces bez jakýchkoliv ohledů
- script <script_file>
  - Spustí skript
- set [name=value]
  - Nastaví proměnnou a její hodnotu
- unset name
  - Odstraní proměnnou

3.3 RapidMiner

Jedná se o software, který poskytuje prostředí pro přípravu dat, strojové učení, prediktivní analýzy a další. Velký důraz při vývoji RapidMineru byl také kladen na to, aby podporoval všechny kroky potřebné pro strojové učení (příprava dat, vizualizace výsledků, validace modelů a jejich optimalizace).

RapidMiner byl dříve – od roku 2001 do roku 2006 – vyvíjen pod názvem YALE. Od roku 2006 vývoj probíhal pod záštitou společnosti Rapid-I, jejíž někteří spoluzakladatelé se podíleli již na dřívějším vývoji. V roce 2013 došlo k přejmenování společnosti Rapid-I na RapidMiner a tento název si ponechává dodnes [21].

Software je dostupný ze stejnojmenných oficiálních stránek pod odkazem https://rapidminer.com/. Co se týká podpory, webové stránky odkazují na dokumentace k softwaru, na komunitu, kde je eventuálně možné se poradit s ostatními uživateli, a na formulář, který umožňuje kontaktovat napřímo zastoupení RapidMineru. Jak již bylo zmíněno, software je dostupný z webových stránek ve dvou verzích, RapidMiner Go a RapidMiner Studio. Verze RapidMiner Go je verze, která je přístupná a užívaná ve webovém prohlížeči, a nevyžaduje tedy stažení, ovšem některé funkce nemusejí být v této verzi k dispozici. Druhou dostupnou verzí je RapidMiner Studio (tato verze bude i dále v práci zmiňována), vyžaduje stažení a nabízí veškeré funkce. RapidMiner Studio je možné stáhnout pro platformu Windows, Mac OS a Linux. V rámci platformy Windows je nabízena 32bitová a 64bitová verze, přitom daná verze se volí na základě nainstalovaného operačního systému. Ke stažení je vždy nabízena poslední verze programu, v současné době je poslední verzí 9.8. Co se týká licencování, RapidMiner Studio je volně dostupný k výzkumným a studentským účelům, jinak je nutné zakoupení licence. Z tohoto ohledu je RapidMiner částečně freeware.

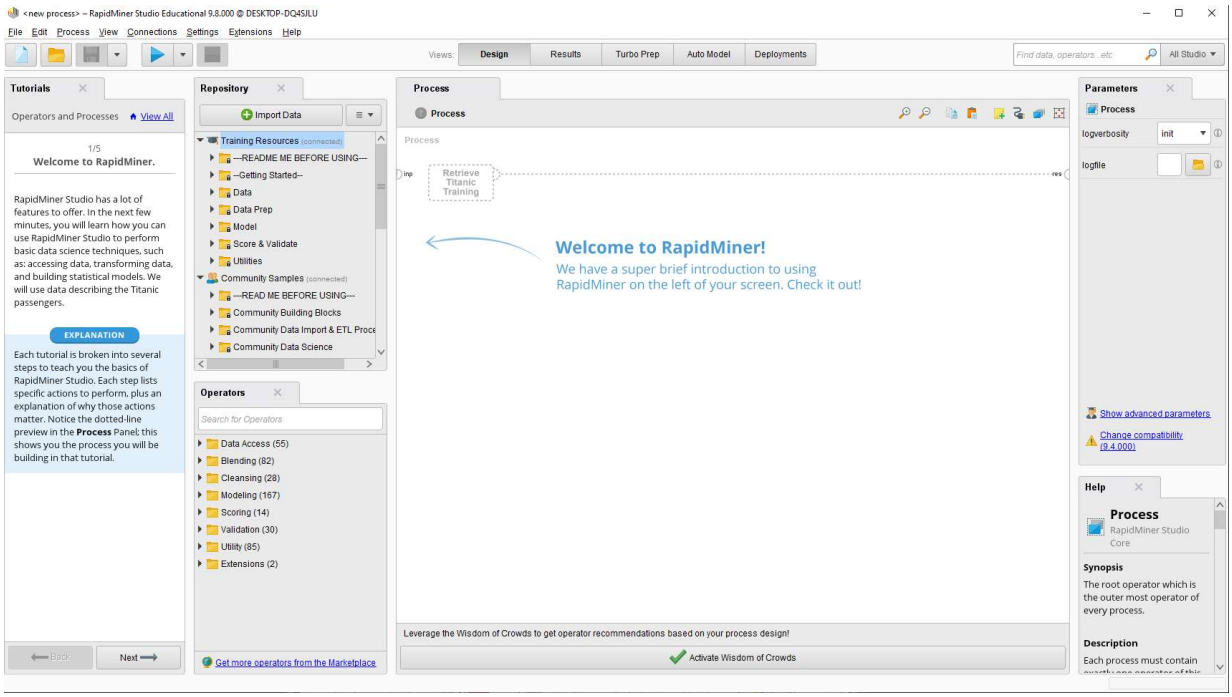
Pro práci s programem je zapotřebí znát dva pojmy, s nimiž se v programu pracuje. Těmi jsou proces a operátor.

Operátor je chápán jako základní stavební jednotka, jež transformuje vstupní data na výstup, který odpovídá danému operátoru. Každá takováto stavební jednotka má minimálně jeden vstup a jeden výstup. Spojováním vstupů a výstupů v souvislém pořadí je docíleno výstupu a je uskutečněn takzvaný proces. Proces je druhým důležitým pojmem, jehož znalost je při práci s RapidMinerem důležitá. Cílem každého projektu v RapidMineru je vyhotovit proces, který konvertuje vstupní data na výstupná, například i na prediktivní model.

Obrázek 24 Operátor Read Excel

Na obrázku je uveden operátor „Read Excel“, jehož úkolem je načíst data z excelovského souboru. Operátory mají různé parametry a možnost nastavení. Pokud je jakýkoliv problém v jeho rámci, je o tom uživatel vyrozuměn výstražným trojúhelníkem v operátoru. Po kliknutí na tento výstražný trojúhelník se zobrazí konkrétní popis problému.

Po instalaci programu a jeho spuštění se zobrazí dialogové okno vyžadující registraci uživatele. Tu není možné přeskočit, mimo ni existuje druhá možnost, a to ukončit program (registrace je pro využívání programu nutná). Po registraci a spuštění programu se zobrazí úvodní obrazovka.



Obrázek 25 Úvodní obrazovka RapidMineru

Úvodní obrazovka se skládá z několika oken. V nejvrchnější části uživatelského prostředí se zobrazí položky File, Edit, Process, Veiw, Connections, Settings, Extensions a Help.

File

File umožňuje utvořit nový proces, uložit jej, otevřít uložené procesy, ty exportovat, importovat data a další.

Edit

V Editu jsou nástroje pro editaci, jako jsou například kopírování, smazání, selekce a vložení, kde jsou veškeré tyto nástroje vykonávány ve společnosti s nějakým operátorem nebo sadou operátorů. Je zde také volba „Show operator info“, která zobrazí podrobné informace o vybraném operátoru. Také se tu nachází nabídka „Save as building block“, jež umožňuje uložit operátor s nastavenými hodnotami a dalšími definicemi uložit pro další znovupoužití.

Process

Process nabízí možnosti pro tvorbu a validaci procesů. Je zde možné proces spustit, validovat, eventuálně nastavit možnost pro automatické spojování operátorů.

View

View umožňuje přepínání mezi pohledy a utvářet pohledy nové. Taktéž je zde možné nastavit, jaké karty mají být v uživatelském prostředí zobrazovány. Mezi možnými zobrazovanými kartami jsou dostupné například:

- **Help** – Karta, která zobrazuje buď obecné informace, pokud není vybráno žádné konkrétní pole, nebo podrobnější informace při aktivním vybraném poli. Například pokud je označen proces, v kartě se zobrazí jeho bližší popis.
- **Parameters** – Karta, která umožňuje měnit parametry operátorů. Dostupné parametry se liší na základě vybraného operátoru. Operátor „Read CSV“, který umožňuje čtení dat ze souboru CSV, respektive comma separated values, umožňuje nastavit oddělovač, jímž jsou data v záznamu oddělována. U řady operátoru se nachází možnost nastavit „Compatibility level“, která umožňuje nastavení chování operátoru dle verze programu, a to z toho důvodu, že chování některých operátorů se liší v závislosti na verzi programu.

- **Result History** – V této kartě jsou uchovávány záznamy pro jednotlivé běhy procesu. Když je proces spuštěn, je utvořen nový záznam, který představuje výsledky procesu. Ty je možné si detailně zobrazit po rozkliknutí záznamu.
- **Log** – Zobrazuje informace o běhu programu.
- **Tree** – Umožňuje zobrazit strom aktivního modelu.
- **Repository** – Jedná se o kartu, která funguje jako centrální úložiště entit. V repozitáři se nacházejí například data, procesy a výsledky. Tyto entity se tam ukládají pro znovupoužitelnost mezi projekty.

Connections – Je zde řada nabídek pro vytvoření spojení mezi databázemi, cloudovými úložišti a další. Existuje například podpora i pro propojení Twitter účtu, tedy účtu sociální sítě Twitter. Následně je při tomto propojení možné získávat data z propojeného účtu. Lze tak například vyhledávat příspěvky s konkrétními klíčovými slovy a další.

Settings – V rámci nastavení lze spravovat RapidMiner. Vedle obecného nastavení, zahrnujícího takové položky, jako jsou jazyková preference, číselné formáty, nastavení možnosti paralelního vykonávání procesů (tedy vykonávání více procesů současně), je možné nastavit také například to, jaká rozšíření RapidMineru se mají automaticky spustit s jeho spuštěním.

Extensions – Umožňuje správu rozšíření RapidMineru, včetně jejich vyhledávání a instalace. V rámci nastavení je možné se připojit k RapidMiner Marketplace, který nabízí řadu volně stažitelných, ale i placených rozšíření.

V sekci nejstahovanější jsou například nabízena tato rozšíření:

- **Weka Extension** – Umožňuje rozšířit modelovací metody RapidMineru o metody, které jsou v systému Weka.
- **Text Processing** – Rozšíření umožňuje rozšířit operátory pro textovou analýzu. Prostřednictvím tohoto doplňku je například možné využívat zdroje dat, jako je obyčejný text, HTML či soubor PDF.
- **Web Mining** – Rozšiřuje možné zdroje dat o internetové zdroje. Díky tomuto doplňku je možné využívat takové zdroje, lze využívat jako zdroj webové stránky, služby a RSS feed.

Help – Zobrazuje řadu možností pro získání informací o RapidMineru a jeho použití.

Mezi položky v této nabídce patří například:

- **Tutorial** – Jedná se o nápovědu integrovanou v RapidMineru, která vysvětluje základní použití programu.
- **Community** – Odkazuje na webovou stránku komunitního fóra, kde je možné diskutovat a vyměňovat si zkušenosti s ostatními uživateli RapidMineru.
- **Doccumentation** – Odkazuje na dokumentaci RapidMineru umístěnou na webových stránkách.

3.3.1 Pohledy RapidMineru

Pod úvodní řadou nabídek se nachází několik tlačítek pro utváření nového projektu, jejich ukládání a otevření projektů stávajících. V této řadě je rovněž k dispozici řada tlačítek pro přepínání mezi pohledy. Dostupné pohledy jsou Design, Results, Turbo Prep, Auto Model, Deployments. Jsou základem pro práci s RapidMinerem.

Design

Pohled design je úvodním pohledem každého nového projektu. V tomto pohledu se plánuje a definuje tok dat a identifikují se klíčové kroky v průběhu procesu. V rámci toku dat se definují takové náležitosti, jako je například import a příprava dat, stavba a validace modelu i včetně jeho aplikace. Veškeré pohledy mají defaultně nastavené karty, které se automaticky otevřou při otevření daného pohledu. Při pohledu Design to je například karta Repository, Operators a Parameters.

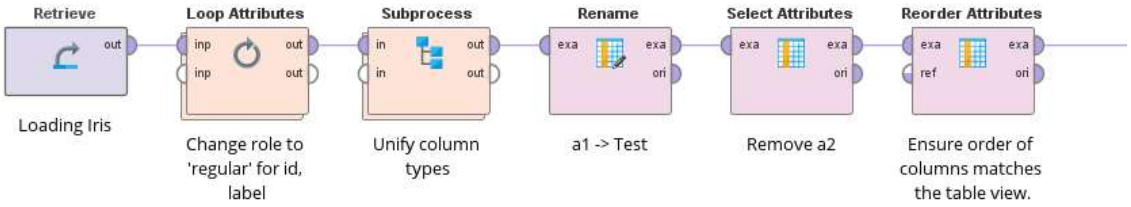
Results

Results je pohled, v němž je možné zobrazit výsledky, konkrétně po sestavení procesu a jeho spuštění. Proces je možné pro zobrazení výsledků spustit vícero způsoby. Mezi hlavní dva způsoby spuštění patří lokální spuštění (používá se pro nenáročné procesy a testování procesů) a spuštění na pozadí (nevyužívají plný výpočetní výkon zařízení, umožňují tak během svého chodu nadále pracovat

s programem). Procesy je taktéž možné spouštět z Repository při pravém kliknutí tlačítka myši a zvolení možnosti Run Process in Background.

Turbo Prep

Záložka Turbo Prep umožňuje rychlou přípravu dat, respektive poskytuje urychlenou práci v jednom okně namísto využití pohledu „Design“ a „Results“. Zatímco se uskutečňuje příprava dat Turbo Prep, na pozadí se sestavuje proces. Tedy veškerou akci nad vykonanými daty RapidMiner následně přetvoří do procesu tak, aby bylo možné dále k procesu přidávat operátory.

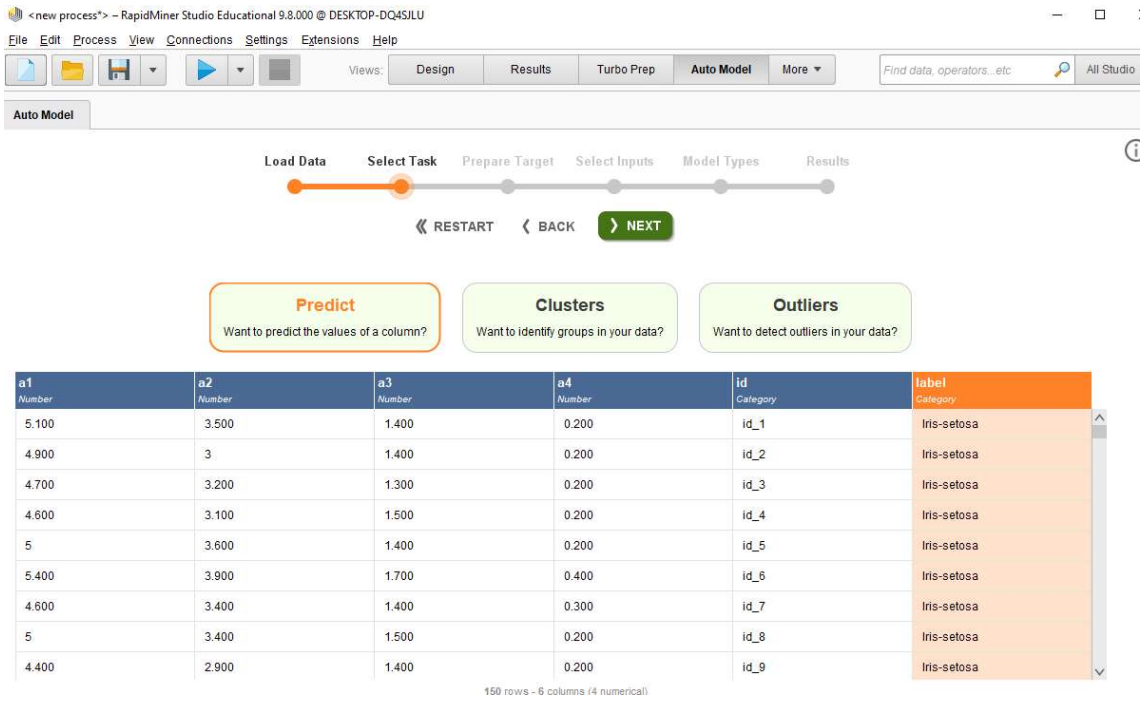


Obrázek 26 Výstup Turbo Prep

Na obrázku je vidět příklad Turbo Prep datasetu Iris, s nímž se pracovalo již v softwaru Weka. Jak již bylo zmíněno, výstupem Turbo Prep je proces, v němž jsou promítnuté veškeré akce, které byly nad daty uskutečněné. Jak již princip procesu napovídá, nejprve dochází k načtení datasetu, následně dochází k označení dat jako „Regular,“ což označuje běžnou vstupní proměnnou. V rámci dalších operátorů dochází k unifikování typů sloupců, přejmenování parametru a1 a smazání sloupce s parametrem a2. Poslední operátor zajišťuje, aby pořadí sloupců odpovídalo tabulkovému náhledu dat.

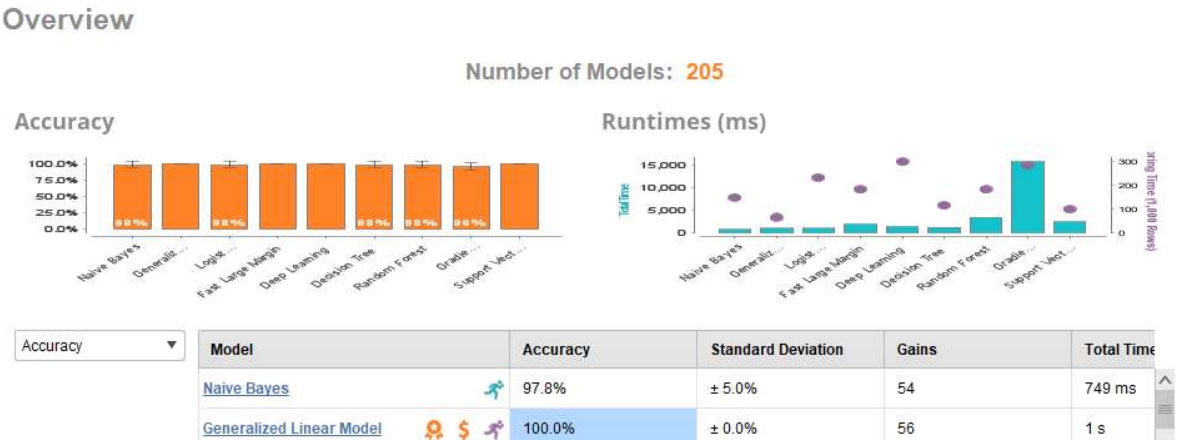
Auto Model

Náhled Auto Model urychluje a usnadňuje stav a validaci modelů. Tvoří rovněž procesy, které je následně možné modifikovat či nasazovat do produkce. Náhled adresuje tři významné oblasti problémů, jimiž jsou predikce, shluky a odlehlé hodnoty.



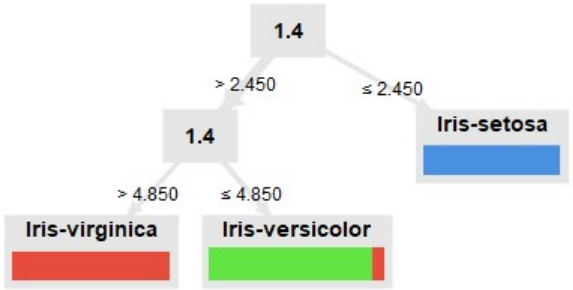
Obrázek 27 Ukázka Auto Model

V rámci predikcí je možné řešit úlohy regrese a klasifikace. Když jsou do softwaru nahrána data, jsou vybraná vstupní data a označena proměnná, která má být predikována, je následně utvořen výstup pro řadu metod současně. Jednotlivé metody jsou ohodnoceny na základě řady parametrů, jako je přesnost, celková doba běhu a směrodatná odchylka. Pro dataset Iris je výstup následující:



Obrázek 28 Výstup Auto Model

Nejlepších výsledků bylo dosaženo prostřednictvím GLM a Naive Bayes, tedy zobecněného lineárního modelu a naivního Bayesovského klasifikátoru. Naivní bayesovský klasifikátor měl pro daný dataset celkově nejkratší dobu běhu s konečnou přesností 97,8 %, kde u GLM byla tato přesnost 100%. Při klasifikaci prostřednictvím rozhodovacích stromů je možné rozhodovací strom případně i zobrazit.



Obrázek 29 Výstup decision tree

Deployments

V rámci tohoto náhledu dochází k nasazení modelu, jímž je myšleno model uveřejnit do podoby, kdy je možné ho použít buď v rámci nově utvářeného programu, který by na základě modelu utvářel predikce, nebo model znovu použít v RapidMineru pro jiné datasety. Model je možné v rámci programování použít prostřednictvím jakékoliv softwarové aplikace, která umožňuje přenos přes http (jeden z nejběžnějších internetových protokolů, sloužících ke komunikaci s webovými servery, jejichž prostřednictvím se přenášejí například i dokumenty ve formátu html). Průvodce má na výběr, zda model bude nasazován lokálně, nebo vzdáleně (způsob nasazení je dán způsobem, jakým chceme model používat).

3.4 R

R je nejenom program, ale také vlastní programovací jazyk, určený pro statistické analýzy. I když má program podobu příkazového řádku, lze z něj tvořit grafické výstupy a eventuálně je taktéž možné stáhnout rozšíření, které disponuje i GUI (graphical user interface), respektive grafickým uživatelským rozhraním. Mezi tato rozšíření patří například RKWard a RStudio. Tato rozšíření existují nejen pro rozhraní, ale taktéž je možné stáhnout taková rozšíření programu, jež mu přidají například nové

funkce. Tato rozšíření jsou dostupná i z oficiálních webových stránek, prostřednictvím balíčků. Rozšíření jsou z oficiálních stránek dostupná zdarma a jsou vydávána od roku 2006 (první vydání programu bylo roku 1993) [22]. Mezi zmíněná rozšíření patří například [23]:

jsonlite – Toto rozšíření umožňuje pracovat se soubory typu JSON. V tomto rozšíření jsou například i funkce pro validaci a úpravu JSON dat. JSON je zkratka pro JavaScript Object Notation. Jde o velmi rozšířený formát, využívaný pro uchování a přenos dat především mezi webovými stránkami a servery. Toto rozšíření neumožňuje pouze tento formát jako samotný používat, ale také zavádět prostředky, které následně mohou komunikovat s webovým rozhraním.

xmlconvert – Umožňuje konverzi mezi XML dokumenty na takový formát, který je pro jazyk R srozumitelný. Formát XML je svým principem podobný formátu JSON, včetně principu jeho užití. Jedná se o jeho konkurenta.

dabr – Rozšiřuje funkci R o funkce určené pro správu databází. Mezi takové funkce patří například select (umožňuje selekci záznamu), insert (umožňuje vložení záznamu do tabulky), delete (smaže daný záznam). Taktéž je umožněno činit zálohy tabulek, a to jejich exportem do souborů typu CSV (comma sepearated values).

ralger – Rozšíření, které umožňuje extrakci dat z webových stránek.

Webové stránky taktéž disponují sekcí, týkající se dokumentace programu. V jejím rámci jsou zmíněny základní principy užití programu.

Software je stejně jako v předešlých dvou případech volně dostupný z oficiálních stránek https://cran.r-project.org/index.html. Je možné jej stáhnout pro platformu Windows, Linux a Mac OS. Vedle poslední stabilní verze je nejen možné stáhnout verze předešlé (pro jednotlivé platformy), ale rovněž lze stáhnout verze testovací, které budou následně vydávány.

Programovací jazyk R je case sensitive, je nutné tedy při psaní příkazu dbát na rozlišování velkých a malých písmen. Je také objektově orientovaný, což představuje skutečnost, že veškeré náležitosti, jako jsou data, funkce a příkazy, jsou představovány objekty. Každý objekt je instancí nějaké třídy, třídu libovolného objektu lze zjistit prostřednictvím příkazu class(). Výpis všech aktuálních objektů náležejících do určité třídy lze zjistit prostřednictvím příkazu ls(). Veškeré psané příkazy jsou v programu označeny červeně. Pokud je zadán příkaz nekompletní a jsou vyžadovány ještě další

parametry, změní se znak „>“, označující zadání příkazu, na symbol „+“, který označuje, že následně psaný příkaz bude připojen k příkazu předešlému + [22].

Jako každý programovací jazyk i R pracuje s daty a používá datové typy. Mezi tyto datové typy se řadí například [22]:

- **character** – slouží pro znak či řetězec znaků;
- **numeric** – slouží pro definici nejen celých, ale například i decimálních čísel;
- **logical** – uchovává hodnoty typu TRUE a FALSE;
- **complex** – pro definici čísel s reálnou a imaginární částí (například 2+1i);
- **integer** – celočíselný datový typ.

Software umožňuje nejen definovat datové typy, ale také jinou práci s daty, jako je například jejich načítání a ukládání. Co se týká čtení dat, můžeme rozlišovat dva typy příkazů. K prvnímu typu příkazů patří příkazy read. Příkazy tohoto typu zpracovávají vstupní data tak, že jsou nejdříve v podobě jednoho dlouhého řádku, kde jsou následně rozdělena do řádků a sloupců tabulky [22]. Mezi příkazy spadající do této skupiny spadá například [22]:

- read.table(file, header, sep, dec, skip, row.names, col.names, na.strings) – čte data v tabulkovém formátu:
  - file – atribut příkazu, který určuje název načteného souboru;
  - header – hodnota typu boolean určuje, zda je v datech hlavička, či nikoliv;
  - sep – znak, který odděluje hodnoty atributů;
  - dec – určuje, jakým znakem jsou oddělovány desetinné části;
  - skip – počet řádků, které mají být přeskočené;
  - row.names – řetězec, který odpovídá názvům sloupců;
  - na.strings – v tomto parametru se definuje, jakým řetězcem je nahrazena chybějící hodnota;
- read.csv() – čte data ve formátu CSV (comma separated value);
- read.delim() – čte data, kde hodnoty atributů jsou odděleny tabulátorem.

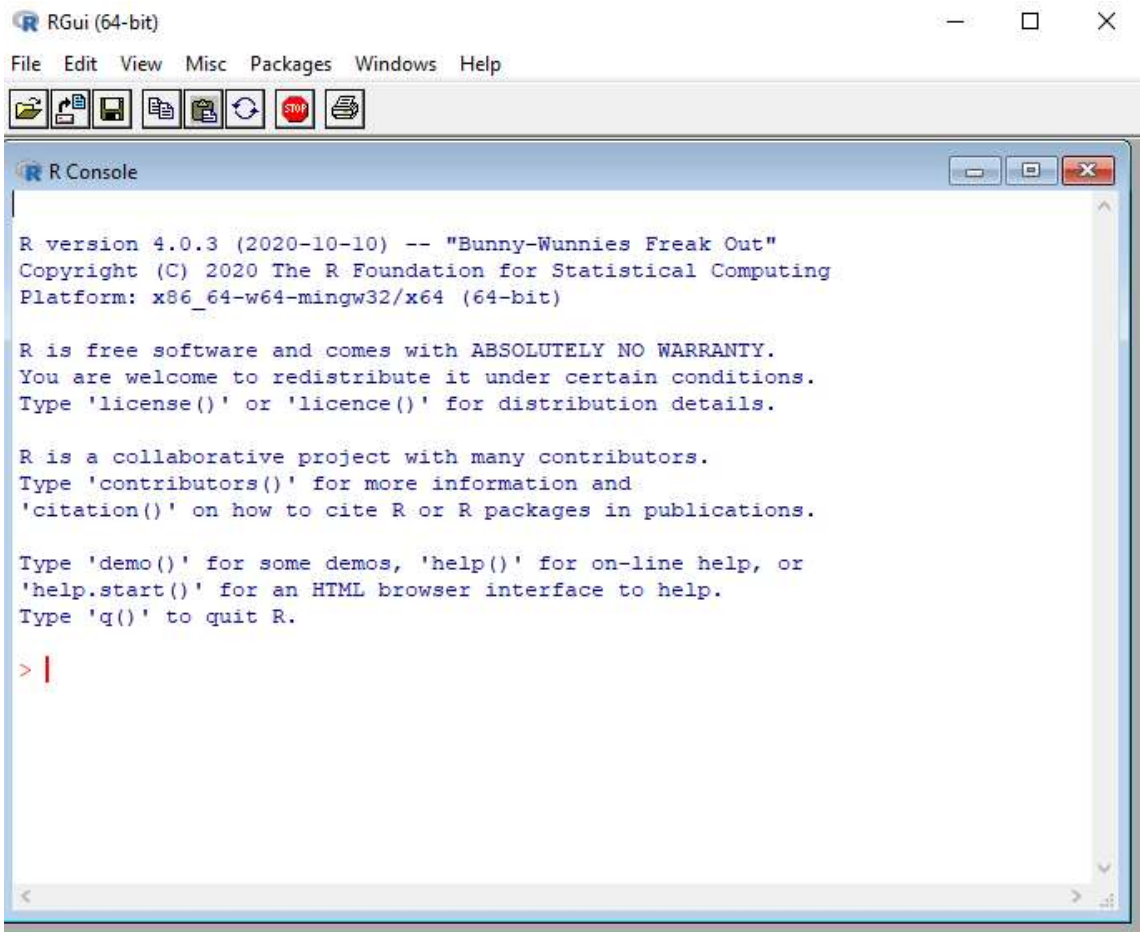
Druhý typ příkazů představují takové, které čtou data v předem určeném pořadí, respektive postupně (sekvenčně).

Mezi tyto příkazy patří například:

- scan(file, what, sep, dec, nmax,…).

3.4.1 Práce s programem

Po úvodní instalaci se při spuštění programu zobrazí konzole, v níž jsou zobrazeny úvodní zprávy. Mezi nimi jsou vypsány příkazy, které poskytují informace k užití programu. Příkladem je příkaz demo (), díky kterému je možné zobrazit ukázky příkazů.



Obrázek 30 Úvodní obrazovka R

Další z nabízených možností je příkaz help (). Ten otevře oficiální webové stránky programu, konkrétně v sekci s dokumentací, v níž je vysvětlována syntax příkazů a kde je uživatel taktéž seznamován s nejzákladnějšími příkazy. Nabídku příkazů lze získat rovněž stisknutím tlačítka tab, kde toto tlačítko doplňuje rozepsané příkazy, eventuálně nabízí příkazy s odpovídajícími znaky. Program lze ukončit příkazem q().

Okno programu mimo konzole poskytuje také několik nabídek, rozmístěných do dvou řad. Ve svrchní řadě je možné volit z těchto možností:

File – Při této volbě je zde řada možností pro práci s projektem. Mezi tyto nabídky například patří: „Load workspace“ (nahraje uložené objekty), „Save workspace“ (uloží utvořené objekty pro možnost jejich následného pozdějšího otevření), „Print“ (vytištění obsahu z konzole, kde je vedle klasického tisku možný i tisk například do PDF), „Load history“ (načte historii zadaných příkazů ze zvoleného adresáře) a „Save history“ (uloží historii příkazů ze zadaného adresáře).

Edit – Jsou zde nabídky pro kopírování, vložení a selekci označených příkazů. Je zde také možná úprava rozhraní dle preferencí uživatele. Mezi možnosti úprav patří například velikost písma, font, zarovnání a počet řádků.

View – Je zde možnost pro schování a zobrazení nabídky vrchní kontextové nabídky.

Misc – Nachází se zde například možnost pro zastavení aktivního výpočtu nebo i všech plánovaných nadcházejících výpočtů. Je zde také nabídka pro výpis a smazání všech objektů.

Packages – Je zde nabídka pro správu balíčků, jejich instalaci, odinstalování a aktualizaci. Jak již bylo naznačeno, balíček je chápán jako skupina funkcí a datových souborů. Balíčky je možné stáhnout z oficiálních webových stránek, nicméně součástí softwaru je 29 předinstalovaných balíčků, z nichž nejdůležitější je balíček zvaný base, který obsahuje základní příkazy programu. Balíčky je možné instalovat přímo z programu, eventuálně je možné balíčky stáhnout prostřednictvím prohlížeče z oficiálních stránek a poté do programu nahrát. Mezi základní příkazy pro práci s balíčky patří:

- installed.packages() – zobrazí seznam nainstalovaných balíčků;
- available.packages() – zobrazí seznam nainstalovaných balíčků;
- install.packages() – nainstaluje balíčky;
- download.packages() – stáhne balíček na disk (lokální úložiště).

Windows – Jsou zde nabídky pro úpravu okna konzole.

Help – Je zde řada nabídek, jak získat informace pro práci s programem, respektive jak vyvolat nápovědu. Nabídka help odkazuje na nejrůznější zdroje informací, mezi něž se například řadí nejen odkaz na internetovou dokumentaci, ale taktéž odkaz na dokumentaci, jež byla stažena společně s programem a je spolu s programem uložena v instalačním adresáři. Nápovědu je možné vyvolat i prostřednictvím příkazů, například help.start() (zobrazí HTML nápovědu).

V řadě pod úvodní kontextovou nabídkou se nachází skupina grafických tlačítek. Jedná se o tlačítka pro otevření a uložení projektu, kopírování a vložení selekce, zastavení procesu a tisk kódu. Ty jsou identické s nabídkami ukrytými mezi kontextovými nabídkami ve svrchní části, jsou zde ale vloženy z toho důvodu, aby k nim měl uživatel softwaru rychlý přístup.

3.4.2 Praktický příklad

V praktickém příkladu bude použitý stejný dataset jako v předešlých dvou případech. V rámci příkladu bude utvořen rozhodovací strom. K tomu je zapotřebí instalace dodatečného balíčku tree, staženého z oficiálních stránek, po stažení následně byla zvolena jeho instalace prostřednictvím nabídky Packages v záhlaví programu. Balíček je možné taktéž instalovat prostřednictvím příkazu install.packages(„tree“). Po instalaci daného balíčku je nutné jej načíst do pracovního prostředí, toho lze docílit prostřednictvím příkazu library(tree). V prostředí je nutné naimportovat i dataset, nad kterým budou operace vykonávány. Toho lze docílit prostřednictvím data(iris), následně jej lze pojmenovat prostřednictvím names(iris).

Pomocí tabie(iris$Species) lze zjistit počet reprezentantů ve třídách a prostřednictvím názvu datasetu lze zobrazit jednotlivé položky v něm.

```
> table(iris$Species)

    setosa versicolor  virginica
        50         50         50

> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
```

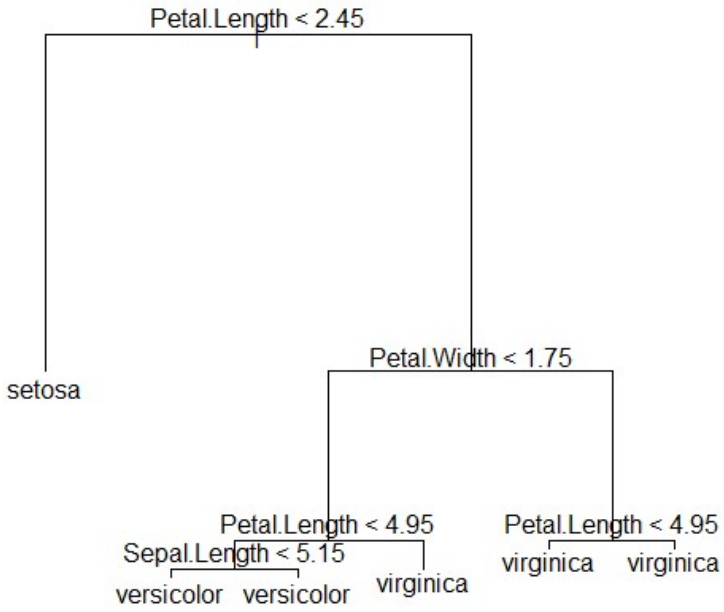
Obrázek 31 Zobrazení datasetu v R

Rozhodovací strom lze sestavit prostřednictvím příkazu tree1 <- tree(Species ~ Sepal.Width + Sepal.Length + Petal.Length + Petal.Width, data = iris). Výsledky rozhodovacího stromu lze zobrazit prostřednictvím summary(tree1). Z výsledku je patrné, že bylo špatně zařazeno 6 případů ze 150.

```
Classification tree:
tree(formula = Species ~ Sepal.Width + Sepal.Length + Petal.Length +
    Petal.Width, data = iris)
Variables actually used in tree construction:
[1] "Petal.Length" "Petal.Width"  "Sepal.Length"
Number of terminal nodes:  6
Residual mean deviance:  0.1253 = 18.05 / 144
Misclassification error rate: 0.02667 = 4 / 150
```

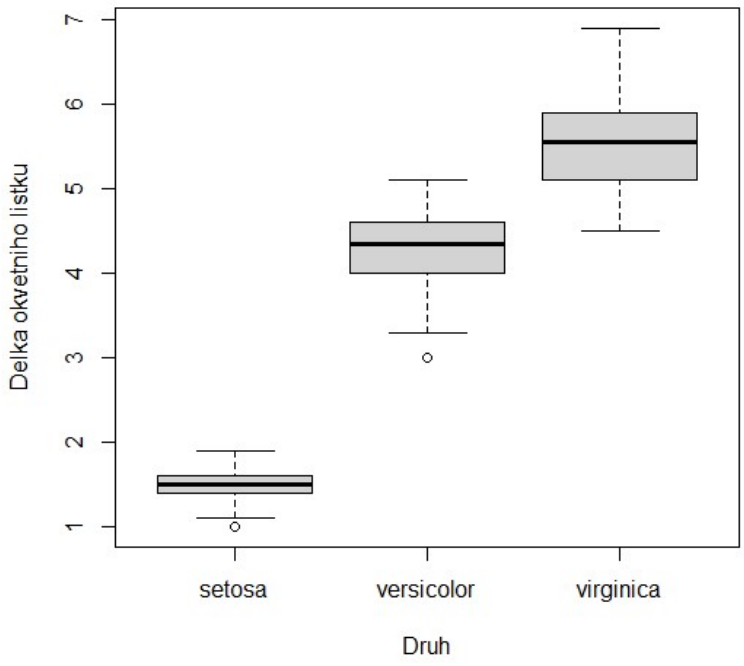
Obrázek 32 Výsledek klasifikace

Rozhodovací strom lze vizualizovat prostřednictvím příkazů plot(tree1) a popisy stromu je možno zobrazit prostřednictvím text(tree1).



Obrázek 33 Rozhodovací strom

Dále lze například prostřednictvím příkazu boxplot(formula=Petal.Length ~ Species, data=iris, xlab="Druh", ylab="Delka okvetniho listku") zobrazit boxplot graf načteného datasetu, kde je z příkladu patrné, že třída Iris Setosa má výrazně jiné délky okvětních lístků.



Obrázek 34 Boxplot graf

4 Výsledky porovnání a rozbor výsledků

Tato část diplomové práce se bude zabývat porovnáním dříve představených jednotlivých softwarů na základě stanovených kritérií. Cílem diplomové práce je nalézt software, který bude nejvhodnější z pohledu nového uživatele. Na základě tohoto faktu budou volena kritéria a jim přiřazena příslušná váha. Samotné porovnání bude uskutečněno prostřednictvím nástroje pro podporu rozhodování.

4.1.1 Dokumentace softwarových požadavků

Významem, respektive cílem dokumentace softwarových požadavků a s ní i diplomové práce je určit, který software je z vybraných tím nejvhodnějším pro nového, individuálního uživatele, a to za účelem řešení dataminingových úloh. S vybraným programem by měl uživatel získat co nejširší spektrum znalostí, které budou dále uplatnitelné, a to nejen pro řešení složitějších úloh, ale případně i pro práci s jinými programy pro zpracování dataminingových úloh. Z obecného pohledu je tedy software vybírán na základě vhodnosti pro nového uživatele bez dosavadní zkušenosti se zpracováním dataminingových úloh. Výběr programu byl omezen na základě jednoho kritéria, jímž je jeho licencování, respektive program musí být volně dostupný pro užití. Ze specifického hlediska bude vybírán program na základě těchto navržených kritérií:

Funkcionalita – Programy budou posuzovány na základě jejich funkcionality (jaký druh funkcí pro zpracování dat nabízejí).

Dokumentace – Bude posuzováno, jak jsou jednotlivé funkce a možnosti práce se softwarem dokumentovány.

Intuitivnost – Bude nahlíženo na to, jak jednoduchá je práce s programy pro dosažení určitého cíle, bez čtení dokumentace. Bude také zohledněno, jak časově náročné je uskutečnění požadované operace.

Nápověda – Bude hodnocena nápověda implantovaná v softwaru samotném.

Rozšířitelnost – Kritérium nahlíží na skutečnost, zda je možné rozšířit funkcionalitu softwaru, například prostřednictvím komunitních balíčků a podobně.

Znovupoužitelnost – Zohledňuje se, zda je při práci se softwarem možné získat obecné znalosti, jestli je naučená činnost opětovně použitelná i při práci s jinými programy.

Pro podporu rozhodování bude využit software Criterium DecisionPlus, dostupný na virtuálních učebnách.

4.1.2 Hodnocení kritérií

Pro podporu rozhodování bude využit software Criterium DecisionPlus ve verzi 3.0, dostupný na virtuálních učebnách. Software je vyvíjen společností InfoHarvest Inc. a umožňuje použít obě nejčastější techniky pro podporu rozhodování, jimiž jsou AHP a SMART. V softwaru byla nejdříve utvořena hierarchie cíle, kritérií a vybraných alternativ.

Obrázek 32 Hierarchie kritérií

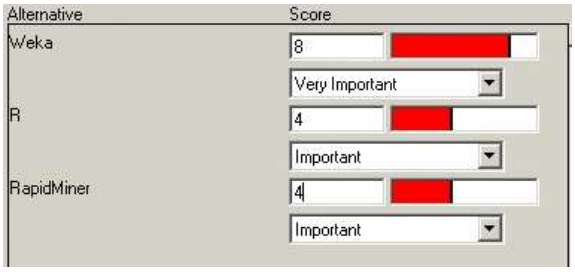
Při běhu programu byla využita metoda AHP, jelikož je tato technika vhodnější pro podporu rozhodování v případech, kdy pracujeme s konečným počtem alternativ. Pro hodnocení kritérií bylo využito metody párového porovnávání. Při této metodě se porovnávají jednotlivé softwary mezi sebou z pohledu daného kritéria. Škála pro

hodnocení kritérií samotných a jejich hodnocení vůči alternativám je nastavena 0–10, kde nula představuje nejnižší možné bodové ohodnocení.

V rámci modelu byla nastavena i důležitost jednotlivých kritérií, kde největší váhu má kritérium dokumentace a nápověda. Tato kritéria byla ohodnocena 10 body, z důvodu jejich důležitosti pro nového uživatele. Dalším, druhým nejvíce hodnoceným kritériem je intuitivnost, která získala 6 bodů, a není tedy tak významná jako dokumentace a nápověda. Tento fakt je podmíněn tím, že z dokumentace a nápovědy intuitivnost v řadě případů nejenže explicitně vyplývá, ale také z pohledu nového laického uživatele není intuitivnost tak podstatná, pokud neví, co hledá. Veškeré ostatní parametry jsou ohodnoceny 5 body, jelikož jsou shledány jako stejně důležité – sám program implicitně popisuje ordinální hodnotu 5 bodů z 10 u kritéria jako důležité (při překladu z anglického important).

Funkcionalita

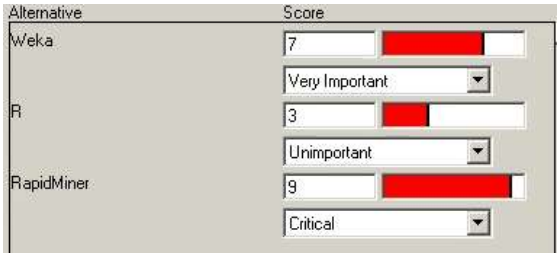
Pod pojmem funkcionalita je míněn přehled funkčních oblastí, jakými software disponuje. Z hlediska funkcionality autor této práce nejlépe hodnotí program Weka, jelikož nabízí největší rozsah funkcí, a to napříč svými moduly. Technicky vzato, v rámci svého modulu CLI vykazuje stejnou funkcionalitu jako R, tedy nabízí veškeré oblasti funkcí prostřednictvím příkazové řádky. V rámci modulu KnowledgeFlow vykazuje obdobnou funkcionalitu jako RapidMiner, kde se prostřednictvím stavebních bloků utváří „tok znalostí“. Navíc Weka obsahuje modul experimenter, který poskytuje rozhraní pro testování výkonnosti jednotlivých algoritmů. Z pohledu množství dostupných algoritmů jsou na tom všechny tři softwary obdobně, avšak oficiální počty nejsou nikde udány. RapidMiner a R hodnotíme jako ekvivalentní, jelikož každý z nich vykazuje stejnou funkcionalitu jako jeden z modulů softwaru Weka. Z pohledu funkcionality je hodnocení následovné.



Obrázek 33 Hodnocení funkcionality

Dokumentace

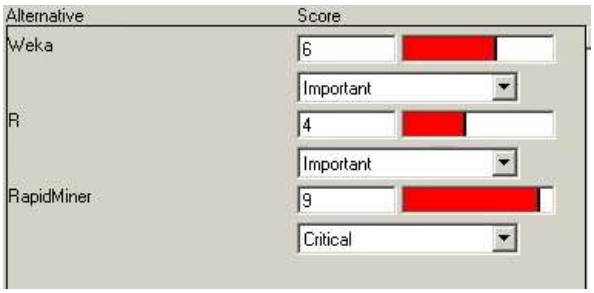
Dokumentace u Weky je poskytována prostřednictvím webu, kde je umístěn rozcestník. Největší část dokumentace představuje z webových stránek volně dostupná PDF publikace, vysvětlující základní principy práce se softwarem, okrajově popisující i některé z využívaných algoritmů. Dokumentace je taktéž k dispozici v rámci videí, složená do jednoho celistvého bezplatného kurzu. RapidMiner obsahuje již v sobě základní tutoriál, který uživatele seznamuje se základy svého prostředí, a dále poskytuje taktéž řadu kurzů, rozdělených podle složitosti a pokročilosti uživatele. Dokumentace R je z pohledu nového uživatele nejméně uživatelsky přívětivá, seznamuje pouze s jednotlivými funkcemi, nikoliv však s posloupností příkazů, jak dosáhnout daných cílů. Software Weka a RapidMiner mají ekvivalentní hodnocení oproti R. Hodnocení bylo následovné.



Obrázek 34 Hodnocení dokumentace

Intuitivnost

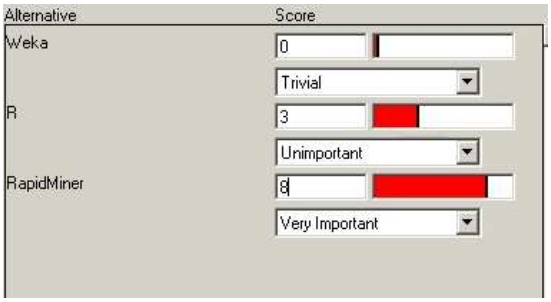
RapidMiner při svém úvodním spuštění nabízí řadu řádně dokumentovaných šablon, kde jsou již popsány a předdefinovány prvky modelu. Tyto šablony jsou vytvořeny pro nejčastější typy úloh, jako jsou marketingové predikce, detekce anomálií a další. Při tvorbě vlastního modelu software popisuje jednotlivé operátory a jejich funkce. Weka ani R takové šablony nenabízejí, avšak uspořádání prostředí Weky jednoznačně ukazuje před zpřístupněním další sekce, jaké operace uživatel musí učinit. Z hlediska tohoto kritéria bylo bodování následovné.



Obrázek 35 Hodnocení intuitivnosti

Nápověda

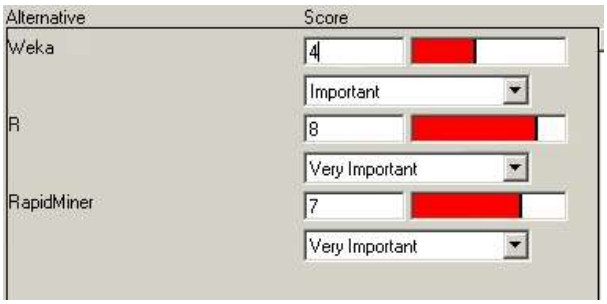
Weka neobsahuje žádnou integrovanou nápovědu. V R příkaz nápovědy „help.start()“ pouze odkáže na úvodní stránku R. V RapidMineru je softwarová nápověda integrovaná nejlépe, jsou tu například popsány i funkce jednotlivých ústředních prvků, respektive operátorů.



Obrázek 36 Hodnocení nápovědy

Rozšířitelnost

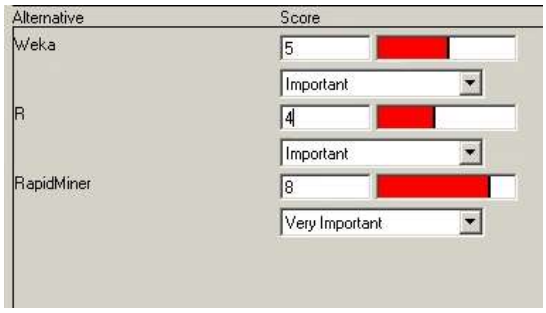
U veškerých softwarů je možné jejich funkcionalitu rozšířit prostřednictvím balíčků, dostupné množství balíčků u jednotlivých softwarů není nikde uvedeno, avšak již z pozorování lze vidět, že R nabízí největší množství těchto rozšíření. Odkazuje na ně prostřednictvím svých oficiálních stránek. RapidMiner má vyhrazené dialogové okno pro stažení zmíněných rozšíření, avšak většina z nich je zpoplatněná. Nejmenší počet rozšíření nabízí Weka, všechna jsou však volně dostupná.



Obrázek 37 Hodnocení rozšířitelnosti

Znovupoužitelnost

Nejvíce získaných, znovupoužitelných znalostí poskytuje RapidMiner, v rámci své dokumentace poskytuje nejvíce všeobecných informací o metodách a vhodnosti jejich použití. Je také schopen v rámci jednoho běhu programu poskytnout i výstup v takové formě, že lze tabulkově porovnat úspěšnost jednotlivých metod, například při úlohách klasifikace. Při práci s programem Weka lze rovněž získat značné množství opětovně použitelných znalostí, kde lze například experimentovat s výkonností jednotlivých algoritmů. Z tohoto ohledu je nejhůře hodnocený software R. Pro práci s ním je zapotřebí mít předešlé znalosti – vedle informací, které si uživatel explicitně vyhledá – a software neposkytuje ani žádné dodatečné, znovu použitelné znalosti.

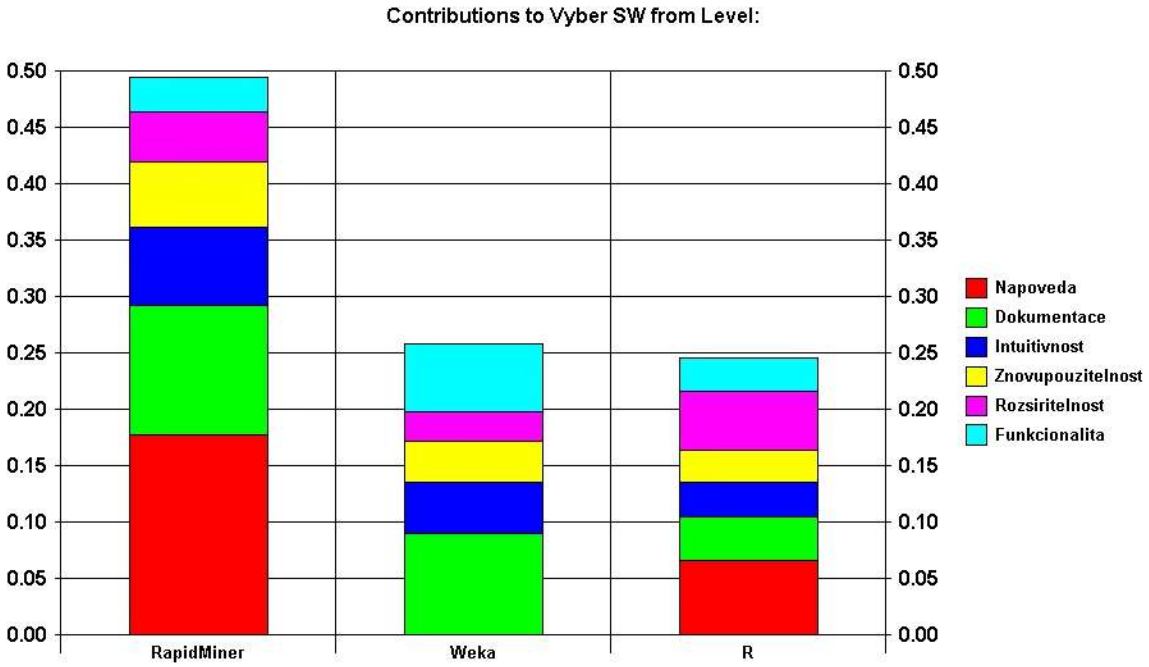


Obrázek 38 Hodnocení znovupoužitelnosti

4.1.3 Výstup hodnocení

Na základě přiřazených vah a hodnocení jednotlivých kritérií a alternativ poskytuje Criterium Decision Plus jednoznačné rozhodnutí. Nejlépe hodnocenou variantou je RapidMiner, který je oproti dvěma dalším výrazně upřednostňován. Jak napovídá graf hodnocení, doporučení RapidMineru je z nejpodstatnější části odůvodněno hodnocením kritéria nápovědy. R je v těsném závěsu za softwarem Weka taktéž z důvodu vysokého hodnocení tohoto kritéria. Nejmenší rozdíly mezi přírůstky

pro rozhodnutí konkrétních variant lze spatřit nad kritérii rozšířitelnosti a znovupoužitelnosti. Druhou nejlépe hodnocenou variantou je Weka a v těsném závěsu za ní je R. Absence bodového přírůstku u Weky, co se kritéria nápovědy týká, je takřka vyrovnána její dokumentací. Významně odlišná je mezi těmito dvěma softwary také rozšířitelnost a funkcionalita.



Obrázek 39 Výstupní graf Criteria DecisionPlus

Shrnutí a závěr

V rámci diplomové práce byl představen pojem data mining, včetně zdrojů, typů dat a osob, které pracují s dolováním dat. Cílem diplomové práce bylo nalézt software pro data mining, který bude vhodný pro zcela nového uživatele. Byly představeny tři softwary tohoto typu licencování. Na základě stanoveného cíle práce byla v rámci sekce týkající se dokumentace softwarových požadavků určena zásadní kritéria a podle nich byly jednotlivé programy hodnoceny. Konečné rozhodnutí bylo učiněno prostřednictvím programu pro podporu rozhodování, zvaného Criteria Decision Plus. V modelu utvořeném v tomto programu byly stanoveny váhy jednotlivých kritérií, kde se hodnotily taktéž jednotlivé alternativy vůči nim.

Na základě stanoveného modelu a učiněných hodnocení byl doporučen zejména software RapidMiner, jenž Criterium Decision Plus doporučil především díky hodnocení jeho integrované nápovědy. Druhou doporučenou alternativou je software Weka a v těsném závěsu za ním je R, kde významný rozdíl mezi těmito dvěma alternativami je poměr hodnocení nápovědy a dokumentace.

Seznam použité literatury

[1] TAISBAK, C. M. *Dedomena: Euclid's Data, Or, The Importance of Being Given*. London: Museum Tusculanum Press, 2003. ISBN 9788772898155.

[2] STAHLBOCK, R., S. F. CRONE and S. LESSMANN. *Data Mining: Special Issue in Annals of Information Systems* [online]. London: Springer, 2010 [cit. 2020-02-12]. ISBN 978-1-4419-1280-0. Dostupné z: https://link.springer.com/book/10.1007/978-1-4419-1280-0

[3] PAT RESEARCH. Predictive Analytics Today. *Predictiveanalyticstoday.com* [online]. 2013 [cit. 2020-02-12]. Dostupné z: https://www.predictiveanalyticstoday.com

[4] What's The Difference Between Structured, Semi-Structured And Unstructured Data? *Forbes* [online]. 2019 [cit. 2021-04-18]. Dostupné z: https://www.forbes.com/sites/bernardmarr/2019/10/18/whats-the-difference-between-structured-semi-structured-and-unstructured-data/?sh=3df50c172b4d

[5] *Encyclopedia of Systems Biology* [online]. New York: Springer, 2013 [cit. 2021-04-18]. Dostupné z: https://link.springer.com/referencework/10.1007/978-1-4419-9863-7

[6] BRAMER, M. *Principles of Data Mining* [online]. 2nd ed. London: Springer, 2013 [cit. 2020-02-12]. ISBN 978-1-4471-4884-5. Dostupné z: https://link.springer.com/book/10.1007/978-1-4471-4884-5

[7] LI, R. History of data mining – Hacker Bits. In: *Hackerbits.com* [online]. 25. 9. 2017 [cit. 2020-02-12]. Dostupné z: https://hackerbits.com/data/history-of-data-mining/

[8] RAJPUT, A. Types of Sources of Data in Data Mining – GeeksforGeeks. In: *Geeksforgeeks.org* [online]. 11. 6. 2018 [cit. 2020-02-12]. Dostupné z: https://www.geeksforgeeks.org/types-of-sources-of-data-in-data-mining/

[9] RUD, O. P. *Data Mining: Praktický průvodce dolováním dat pro efektivní prodej, cílený marketing a podporu zákazníků*. Praha: Computer Press, 2001. ISBN 80-7226-577-6.

[10] BERKA, P. *Dobývání znalostí z databází.* Praha: Academia, 2003. ISBN 80-200-1062-9.

[11] LEAN SIX SIGMA. DMAIC. Lean6sigma.cz [online]. © 2020 [cit. 2020-10-24]. Dostupné z: https://lean6sigma.cz/dmaic/

[12] WIDE SKILLS. Data Mining Tasks. WideSkills.com [online]. © 2018 [cit. 2020-02-12]. Dostupné z: https://www.wideskills.com/data-mining-tutorial/05-data-mining-tasks

[13] DURČÁK, P. Statistika a pravděpodobnost: Přírodovědecká fakulta Masarykovy univerzity. In: *Napočítači.cz* [online]. 8. 9. 2017 [cit. 2020-10-24]. Dostupné z: https://www.napocitaci.cz/33/neuronove-site-a-princip-jejich-fungovani-uniqueidgOkE4NvrWuNY54vrLeM670eFNQh552VdDDulZX7UDBY/

[14] REXER, Karl. 2nd Annual Data Miner Survey – Summary Report. [Dokument typu PDF]. Rexer Analytics. 2008 [cit. 2021-04-13]

[15] Nailing Your Software Requirements Documentation. *Lucidchart* [online]. [cit. 2021-04-13]. Dostupné z: https://www.lucidchart.com/blog/software-requirements-documentation#

[16] *IOPScience* [online]. [cit. 2021-04-13]. Dostupné z: https://iopscience.iop.org/

[17] What Analytics, Big Data, Data mining, Data Science software you used in the past 12 months for a real project? *KDnuggets* [online]. KDnugget, 2013 [cit. 2021-04-20]. Dostupné z: https://www.kdnuggets.com/polls/2013/analytics-big-data-mining-data-science-software.html

[18] THE UNIVERSITY OF WAIKATO. Weka: The workbench for machine learning. *Cs.waikato.ac.nz* [online]. © 2020 [cit. 2020-11-05]. Dostupné z: https://www.cs.waikato.ac.nz/~ml/weka/index.html

[19] THE UNIVERSITY OF WAIKATO. Decision Table: Weka: The Workbench for machine learning. In: *Weka.sourceforge.io* [online]. © 2020 [cit. 2020-11-05]. Dostupné z: https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/DecisionTable.html

[20] THE UNIVERSITY OF WAIKATO. Best First: Weka: The Workbench for machine learning. In: *Weka.sourceforge.io* [online]. © 2020 [cit. 2020-11-05]. Dostupné z: https://weka.sourceforge.io/doc.dev/weka/attributeSelection/BestFirst.html

[21] *RapidMiner* [online]. © 2020 [cit. 2020-11-06]. Dostupné z: https://rapidminer.com/

[22] *The R Project for Statistical Computing* [online]. © 2020 [cit. 2020-11-11]. Dostupné z: https://www.r-project.org/

[23] The R Project for Statistical Computing . Contributed packages. *Cran.r-project.org* [online]. © 2020 [cit. 2020-11-11]. Dostupné z: https://cran.r-project.org/

Seznam obrázků

Zdroje obrázků

Obrázek 1 Proces získávání znalostí

Zdroj: BRAMER, M. *Principles of Data Mining* [online]. 2nd ed. London: Springer, 2013 [cit. 2020-02-12]. ISBN 978-1-4471-4884-5. Dostupné z: https://link.springer.com/book/10.1007/978-1-4471-4884-5

Obrázek 2 Schéma SEMMA

Zdroj: BERKA, P. *Dobývání znalostí z databází.* Praha: Academia, 2003. ISBN 80-200-1062-9.

Obrázek 3 Schéma metodiky 5A

Zdroj: Vlastní zpracování.

Obrázek 4 Schéma metodiky CRISP-DM

Zdroj: BERKA, P. *Dobývání znalostí z databází.* Praha: Academia, 2003. ISBN 80-200-1062-9.

Obrázek 5 Přehled procesů CRISP-DM

Zdroj: Vlastní zpracování.

Obrázek 6 Schéma Six Sigma

Zdroj: ISEL GLOBAL. Benefits of Six Sigma Process Improvement Methodology. *Iselglobal.com* [online]. © 2018 [cit. 2020-10-24]. Dostupné z: https://iselglobal.com/2018/08/benefits-of-six-sigma-process-improvement-methodology/

Obrázek 7 Metoda nejmenších čtverců

Zdroj: BUDÍKOVÁ, M. a kol. Regresní analýza. In: *Is.muni.cz* [online]. © 2016 [cit. 2020-10-24]. Dostupné z: https://is.muni.cz/do/rect/el/estud/prif/ps15/statistika/web/pages/regres-anal.html

Obrázek 8 Vzdálenosti shlukové analýzy

Zdroj: BERKA, P. *Dobývání znalostí z databází.* 1. vyd. Praha: Academia, 2003. ISBN 80-200-1062-9.

Obrázek 9 Kontingenční tabulka

Zdroj: Vlastní zpracování.

Obrázek 10 Rozhodovací strom

Zdroj: Vlastní zpracování.

Zdroj: Vlastní zpracování, snímek z programu Weka

Obrázek 22 Model pro klasifikace datasetu Iris

Zdroj: Vlastní zpracování, snímek z programu Weka

Obrázek 23 Textový výstup z modelu

Zdroj: Vlastní zpracování, snímek z programu Weka

Obrázek 24 Operátor Read Excel

Zdroj: Vlastní zpracování, snímek z programu RapidMiner

Obrázek 25 Úvodní obrazovka RapidMineru

Zdroj: Vlastní zpracování, snímek z programu RapidMiner

Obrázek 26 Výstup Turbo Prep

Zdroj: Vlastní zpracování, snímek z programu RapidMiner

Obrázek 27 Ukázka Auto Model

Zdroj: Vlastní zpracování, snímek z programu RapidMiner

Obrázek 28 Výstup Auto Model

Zdroj: Vlastní zpracování, snímek z programu RapidMiner

Obrázek 29 Výstup decision tree

Zdroj: Vlastní zpracování, snímek z programu RapidMiner

Obrázek 30 Úvodní obrazovka R

Zdroj: Vlastní zpracování, snímek z programu R

Obrázek 31 Zobrazení datasetu v R

Zdroj: Vlastní zpracování, snímek z programu R

Obrázek 32 Výsledek klasifikace

Zdroj: Vlastní zpracování, snímek z programu R

Obrázek 33 Rozhodovací strom

Zdroj: Vlastní zpracování, snímek z programu R

Obrázek 34 Boxplot graf

Zdroj: Vlastní zpracování, snímek z programu R

Obrázek 35 Hierarchie kritérií

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Obrázek 36 Hodnocení funkcionality

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Obrázek 37 Hodnocení dokumentace

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Obrázek 38 Hodnocení intuitivnosti

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Obrázek 39 Hodnocení nápovědy

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Obrázek 40 Hodnocení rozšířitelnosti

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Obrázek 41 Hodnocení znovupoužitelnosti

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Obrázek 42 Výstupní graf Criteria DecisionPlus

Zdroj: Vlastní zpracování, snímek z programu Criterium DecisionPlus

Přílohy

Zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor:	**Bc. Lukáš Zasadil**
Studium:	I1800754
Studijní program:	N6209 Systémové inženýrství a informatika
Studijní obor:	Informační management
Název diplomové práce:	**Data mining a freeware**
Název diplomové práce AJ:	Data mining and freeware

Cíl, metody, literatura, předpoklady:

Cíl práce:

Vyhledat a sestavit přehled reprezentantů komerčních a freeware produktů pro Data Mining. Popsat požadované funkcionality a porovnat vybrané produkty z této oblasti podle zvolených kritérií.

Úvod

Literární rešerše

Cíl práce

Metodologie

Výsledky porovnání a rozbor výsledků

Shrnutí a závěr.

Seznam literatury

Přílohy

Zadání práce (kopie)

Doplní student

Garantující pracoviště:	Katedra informatiky a kvantitativních metod, Fakulta informatiky a managementu
Vedoucí práce:	prof. RNDr. Hana Skalská, CSc.
Datum zadání závěrečné práce:	14.1.2018