Moderní techniky pro aplikace dataminingu

Bc. Adam Viktorin

Diplomová práce
2015

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Adam Viktorin**
Osobní číslo:	**A13453**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Počítačové a komunikační systémy**
Forma studia:	**prezenční**
Téma práce:	**Moderní techniky pro aplikace dataminingu**
Téma anglicky:	**Modern Techniques for Data-mining Applications**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na dané téma.**
2. **Prostudujte a popište jednotlivé nejrozšířenější data mining techniky.**
3. **Proveďte studii hybridizací jednotlivých technik, fuzzy a jiných bio-inspirovaných přístupů.**
4. **Otestujte vybrané techniky a jejich případné hybridizace na reálných příkladech databázových benchmark setů.**
5. **Proveďte analýzu dosažených výsledků.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **LAROSE, Daniel T. Discovering knowledge in data: an introduction to data mining. Hoboken, N.J.: Wiley-Interscience, c2005, xv, 222 s. ISBN 978-0-471-66657-8.**
2. **Information visualization in data mining and knowledge discovery. San Francisco: Morgan Kaufmann Publishers, c2002, xiii, 407 s. ISBN 15-586-0689-0.**
3. **WITTEN, I, Eibe FRANK a Mark A HALL. Data mining: practical machine learning tools and techniques. 3rd ed. Amsterdam: Morgan Kaufmann, 2011, xxxiii, 629 s. Morgan Kaufman series in data management systems. ISBN 978-0-12-374856-0.**
4. **HAN, Jiawei, Micheline KAMBER a Jian PEI. Data mining: concepts and techniques. 3rd ed. Waltham: Morgan Kaufmann, c2012, xxxv, 703 s. Morgan Kaufmann series in data management systems. ISBN 978-0-12-381479-1.**
5. **AL]., Soumen Chakrabarti ... [et]. Data mining know it all. Burlington, Mass: Morgan Kaufmann Pub, 2009. ISBN 00-808-7788-5.**
6. **LINOFF, Gordon a Michael J BERRY. Data mining techniques: for marketing, sales, and customer relationship management. 3rd ed. Indianapolis: Wiley, c2011, xl, 847 s. ISBN 978-0-470-65093-6.**
7. **SKALSKÁ, Hana. Data mining a klasifikační modely. Vyd. 1. Hradec Králové: Gaudeamus, 2010, 154 s. ISBN 978-80-7435-088-7.**
8. **LABERGE, Robert. Datové sklady: agilní metody a business intelligence. 1. vyd. Brno: Computer Press, 2012, 350 s. ISBN 978-80-251-3729-1.**

Vedoucí diplomové práce: **doc. Ing. Roman Šenkeřík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **12. ledna 2015**

Termín odevzdání diplomové práce: **15. května 2015**

Ve Zlíně dne 6. února 2015

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

Ing. Miroslav Matýsek, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 7. 5. 2015

.....
podpis diplomanta

ABSTRAKT

Tato práce představuje datamining a testuje účinnost jednotlivých technik na benchmark datasetech Iris a Wine. První část se věnuje detailnímu popisu dataminingu jako procesu získávání informací z dat, popisu datových typů, jejich vizualizaci a předzpracování pro účely dataminingových algoritmů. V druhé části jsou detailně popsány jednotlivé datasety, dále jsou představeny moderní datamining algoritmy a jejich funkce je testována a porovnána. Mezi techniky používané těmito moderními algoritmy patří: fuzzy logika, umělé neuronové sítě, evoluční technologie a heuristika.

Klíčová slova: Datamining, clustering, fuzzy logika, neuronové sítě, evoluční technologie, heuristika, preprocessing, RS, DE, DBSCAN, FKM, KMPP, Iris, Wine, dataset.

ABSTRACT

This thesis is presenting datamining and testing functionality of some datamining techniques on benchmark datasets Iris and Wine. The first part is focused on detailed description of datamining as a process of knowledge discovery in data, data attribute types description, visualization of data and data preprocessing. Datasets and modern datamining techniques are described and tested in the second part of this thesis. Modern datamining techniques include: fuzzy logic, artificial neural networks, evolution technologies and heuristic.

Keywords: Datamining, clustering, fuzzy logic, neural networks, evolution technologies, heuristic, preprocessing, RS, DE, DBSCAN, FKM, KMPP, Iris, Wine, dataset.

Chtěl bych poděkovat především třem lidem, kteří se podíleli na vedení mé diplomové práce, panu Romanu Šenkeříkovi za trpělivost u konzultací i s jeho velmi nabitým programem, panu Michalovi Pluháčkovi za odbornou pomoc při řešení problémů praktické části a za jeho cenné rady týkající se jak diplomové práce, tak postgraduálního studia a paní Zuzaně Komínkové Oplatkové za cenné připomínky a rady, které mi poskytla naprosto nezištně.

Dále bych chtěl poděkovat své rodině a přítelkyni, za jejich neutuchající víru v mé schopnosti a podporu, kterou mi poskytují. Stejně tak děkuji přátelům, jejichž podpora je pro mne velmi důležitá. V neposlední řadě bych také chtěl poděkovat všem lidem na fakultě aplikované informatiky, kteří se snaží studium co nejvíce přiblížit studentům a dělají tak z učení zábavu.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

V dnešní moderní společnosti, díky technologickému pokroku a rozšířenosti výpočetní techniky, denně vzniká nepřeberné množství dat. Tato data se různě (lokální počítače, databáze, cloudová úložiště, datová centra) ukládají k dalšímu zpracování, na které už se ve velkém množství případů ani nedostane. Jednoduše je tato práce, pokud je vykonávána manuálně, velmi časově a finančně náročná, navíc při rychlosti, se kterou data vznikají, není ani možné data ručně zpracovat. Nicméně známý citát, často spojovaný (neprokazatelně) se Sirem Francisem Baconem, zní: „Scientia potentia est“ - vědění je síla. A protože je cena informací vyčíslitelná, a v mnohých odvětvích velmi vysoká (finančnictví, akciové trhy, elektronika, apod.), roste poptávka po automatizovaném získávání informací z dat.

Datamining je tedy rychle se rozvíjející vědní disciplínou, která využívá znalosti z různých vědních oborů, jako jsou: Statistika a analýza dat, informační technologie, evoluční technologie, fuzzy logika, teoretická matematika, diskrétní matematika, kombinatorika a umělá inteligence.

Velmi zajímavou částí této vědní disciplíny jsou algoritmy evolučních technologií, což jsou algoritmy inspirované životem a jeho vznikem. Využívá se zde teorie evoluce, která popisuje vznik a vývoj života na základě přirozeného výběru, křížení a mutace jedinců populace.

Neméně zajímavá je fuzzy logika využívající k ohodnocení výroků a náležitosti do množin pravděpodobnost, namísto binárních hodnot, které používa klasická Booleova logika. Fuzzy logika je člověku bližší, protože je bližší přirozenému jazyku. Příkladem může být popis teploty vody ve sklenici. Zatímco podle Booleovy logiky se dá vyjádřit pouze dvoustavovou hodnotou (teplá/studená). Fuzzy logika, stejně jako člověk, dokáže rozlišit větší množství stavů, např. Ledová, studená, chladná, vlažná, teplá, horká, vařící. Tato logika se velmi hojně využívá v dataminingu například při klasifikaci dat, kde jsou jednotlivá pravidla vytvořena právě z prvků fuzzy logiky. Využití umělé inteligence v dataminingu je také důležité a to především využití strojového učení, které je velmi vhodné ke klasifikaci dat.

Pod pojmem moderní techniky v názvu této diplomové práce jsou myšleny především techniky využívající hybridizace a paralelizace výše zmíněných metod, jako je například využití evolučních algoritmů pro vygenerování pravidel klasifikátoru dat využívajícího

prvky fuzzy logiky nebo použití evolučních výpočetních technik pro vhodné nastavení neuronové sítě.

První část této práce je věnována detailnímu pohledu na datamining a možnostem jeho nasazení. Jsou zde zmíněny jak základní pojmy a úkoly dataminingových procesů, tak i úkony spojené s přípravou dat pro dataminingové algoritmy. Data, jako taková, jsou blíže specifikována a jednotlivé části preprocessingu jsou rozebrány do detailu.

Druhá část se věnuje testování již existujících, klasických i hybridních, algoritmů na dvou známých benchmark datasetech Iris a Wine. Cílem datamining procesu na jednotlivých datasetech je klasifikace druhů kosatců a vína do tříd. K tomuto účelu byly vybrány algoritmy, které využívají fuzzy logiku, umělé neuronové sítě, evoluční technologie a heuristiku. Cílem práce je ukázat, že moderní clustering algoritmy jsou na takové úrovni, že je lze použít i pro klasifikační úlohy.

I. TEORETICKÁ ČÁST

1 DATAMINING

Datamining lze definovat různě, v této práci ovšem bude použita následující definice:

Def. 1.: *Datamining je proces, který nalézá užitečné vzory a trendy ve velké množině dat. [1]*

Termín *datamining* je ve skutečnosti vytvořen špatně, protože pokud vycházíme z předpokladu, že vznikl spojením slov, stejně jako například *těžba zlata* (angl. *gold mining*), kde se z kamení a písku těží zlato, pak by se mělo použít spíše označení *těžba informací/znalostí* (angl. *knowledge mining*). Takové označení ovšem nenaznačuje, že se informace získávají z velké množiny dat a označení *těžba informací z dat* (angl. *knowledge mining from data*) je zase příliš dlouhé, proto se zažilo označení *datamining*. [2]

Datamining bývá často považován za synonymum k jinému používánému termínu, KDD (Knowledge Discovery in Data). V KDD je termínem datamining označen jeden krok tohoto procesu. [2]

V této práci jsou termíny datamining a KDD zaměnitelné, tedy datamining označuje celý proces získávání informací z dat včetně jejich předzpracování (preprocessingu) a následné prezentace.

Z pohledu datového skladiště může být datamining vnímán jako pokročilá fáze OLAP (Online Analytical Processing). Datamining však jde daleko dál, než jen k sumarizačnímu analytickému zpracování dat v datových skladištích. Datamining přidává pokročilé techniky analýzy dat. [3]

Na trhu je množství software označených jako datamining systémy, některé z nich ovšem nesplňují základní požadavky na takový systém. Systém analyzující data, který ovšem nezvládá jejich velké množství by mohl být lépe kategorizován jako systém strojového učení, nástroj pro statistickou analýzu dat nebo experimentální systémový prototyp. Systém, který pouze získává data nebo informace včetně nalezení agregovaných hodnot, nebo systém používající deduktivní zodpovídání dotazů na velkých databázích by mohl být lépe kategorizován jako databázový systém, systém získávání informací nebo deduktivní databázový systém. [3]

Datamining zahrnuje integraci technik z velkého množství disciplín – databázove technologie, technologie datových skladišť, statistika, strojové učení, výkonné výpočetní technologie, rozpoznávání vzorů a trendů, neuronové sítě, vizualizace dat, získávání

informací, zpracování signálů, zpracování obrazu, analýza prostorových dat, analýza dočasných dat, evoluční technologie. Z databázového hlediska je kladen důraz především na efektivitu a škálovatelnost technik dataminingu. Škálovatelný algoritmus je takový, jehož čas běhu roste lineárně s velikostí dat bez použití nadbytečných systémových zdrojů (výpočetní výkon, paměť). [3]

Aplikací dataminingu jsou z databázových systémů získávány zajímavé informace, zákonitosti a informace vyšší úrovně. Tyto informace jsou dále zkoumány z různých úhlů a mohou být využity k rozhodování, řízení procesů, informačnímu managementu a zpracování dotazů. Proto je datamining považován za velmi důležitý nástroj v databázových a informačních systémech. [3]

1.1 Příklady využití dataminingu

Podle MGI (McKinsey Global Institute) reportu [4] většina amerických společností, s více než tisícem zaměstnanců, průměrně skladuje 200 TB dat. MGI předpokládá, že meziročně vzroste objem generovaných dat o 40%. Tato skutečnost vytváří prostor pro společnosti zabývající se snižováním množství dat, které je třeba ukládat. Podle MGI mohou tyto společnosti očekávat nárůst zisků až 60%. I společnosti, které těchto služeb již využívají, by mohly ušetřit mnohem více, pokud by zvýšily efektivitu a kvalitu procesu.

V roce 2012 proběhla v Americe volba prezidenta, při které podle technologického posudku MIT (Massachusetts Institute of Technology) [5] především efektivní využití dataminingu pomohlo prezidentu Barracku Obamovi k vítězství nad Mittem Romneym. Obamův tým nejprve použil datamining pro určení voličů Barracka Obamy a poté se zaměřil na to, aby je kampaň přiměla jít k volbám. Jiný datamining model byl využit pro předpovězení výsledků voleb v jednotlivých volebních okrscích. Ve velmi důležitém nerozhodném volebním okrsku Hamilton v Ohiu model předpověděl, že Obama získá 56.4% hlasů, ve skutečnosti obdržel 56.6% hlasů. Tak přesný model umožnil ideální využití prostředků na kampaň a Barrack Obama volby vyhrál.

Měsíčně kontaktuje zákaznické centrum západního pobřeží Bank of America až 13 milionů zákazníku. V minulosti si každý zákazník vyslechl stejné marketingové nabídky, ať už byly v jeho zájmu, či nikoliv. Místo nabízení stejného produktu všem Bank of America začala využívat informace o zákaznících pro lepší cílení marketingu, ke kterému využívá technik dataminingu. [6]

1.2 Nepravdy o dataminingu

V této části jsou vyčteny jednotlivé nepravdivé představy o dataminingu, se kterými se dnes můžeme setkat.

1.2.1 Automatický dataminingový nástroj

Klamná představa, že existují datamining nástroje, které mohou být puštěny na datových zdrojích a naleznou požadované řešení automaticky.

Realita je taková, že neexistují automatické dataminingové nástroje, které by mechanicky vyřešily problém za uživatele bez jeho přičinění. Datamining je spíše proces, který je třeba implementovat. [1]

1.2.2 Proces dataminingu je autonomní

Představa, že datamining proces je autonomní, a vyžaduje pouze malé množství dohledu uživatele, je taktéž nepravdivá.

Dataminingové nástroje nejsou kouzelné, bez interakce zkušeného uživatele tyto nástroje poskytnou pouze špatné odpovědi na špatné otázky aplikované na špatných datech. Je třeba poznamenat, že špatná analýza dat, je horší, než žádná analýza. Špatná analýza vede ke špatným obchodním rozhodnutím. I po nasazení vhodného modelu vstup nových dat často vyžaduje upravení modelu a kontrolu kvality, což je práce pro zkušené analytiky. [1]

1.2.3 Datamining je rychle návratná investice

Návratnost procesu dataminingu se liší v závislosti na pořizovacích nákladech, platech analytiků, ceně přípravy datových skladišť a dalších faktorech. [1]

1.2.4 Datamining software lze jednoduše používat

Tato představa není úplně zavádějící, ale pro jednoduché použití datamining software je potřeba, aby měl uživatel základní znalosti technik dataminingu, přípravy dat, analytiky a byl seznámen s řešeným datamining problémem. [1]

1.2.5 Datamining identifikuje příčiny obchodních/výzkumných problémů

Datamining proces pouze odhalí vzory a trendy, nalezení přičin už je na uživatelích. [1]

1.2.6 Datamining automaticky pročistí databázi

Tento úkol je na obsluze databáze, která bude muset připravit data pro proces dataminingu, často i data, na která se roky nesahalo. [1]

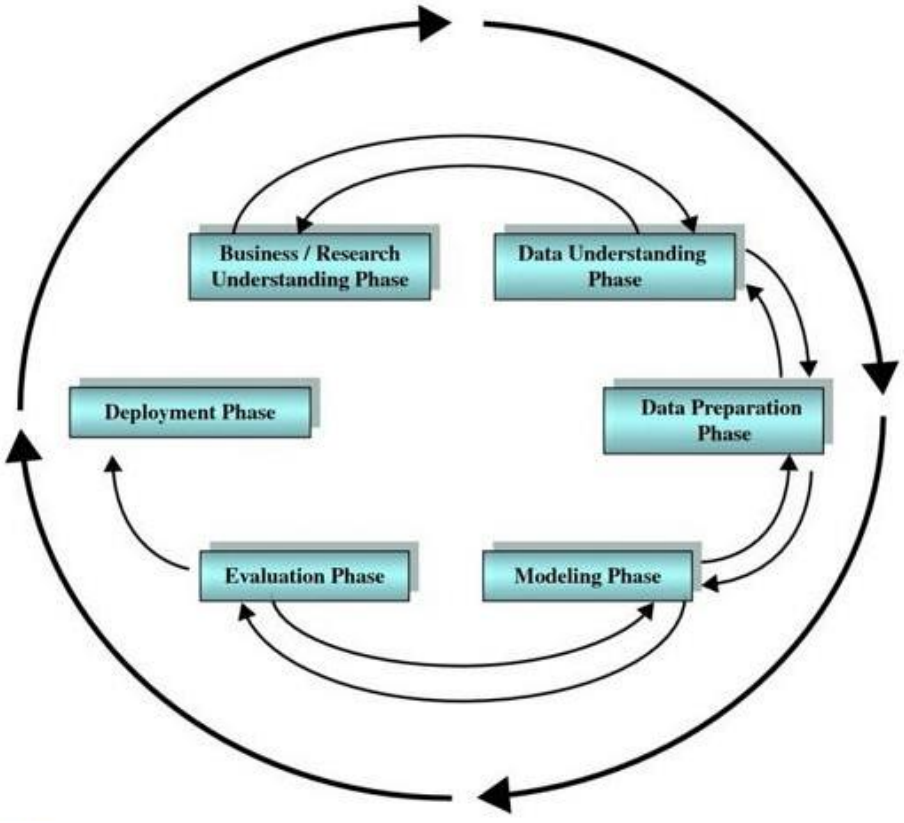
1.2.7 Datamining vždy poskytuje pozitivní výsledky

Není zaručeno, že použití technik dataminingu povede k nalezení použitelných dat. Ovšem pokud jsou datamining techniky aplikovány zkušenými analytiky, kteří mají přehled o cílích a požadavcích, pak může datamining poskytnout použitelné výsledky. [1]

1.3 CRISP-DM

Tato část se věnuje meziprůmyslovému standardu, který popisuje datamining jako proces, který je třeba implementovat.

CRISP-DM (The Cross-Industry Standard Practice for Data Mining) [7] je meziprůmyslový standard, který vznikl za účelem sjednocení implementace dataminingu do strategie řešení problémů v obchodních a výzkumných jednotkách. CRISP-DM je dílem analytiků Daimler-Chrysler, SPSS (Statistical Package for the Social Sciences) a NCR. [1]



Obr. 1. CRISP-DM. [1]

Význam popisků v obrázku (Obr. 1) je následující:

Business/Research Understanding Phase – fáze pochopení obchodu/výzkumu, **Data Understanding Phase** – fáze pochopení dat, **Data Preparation Phase** – fáze přípravy dat, **Modeling Phase** – fáze modelování, **Evaluation Phase** – fáze vyhodnocování, **Deployment Phase** – fáze nasazení.

Datamining projekt se podle CRISP-DM skládá z šesti fází, které jsou zobrazeny na obrázku (Obr. 1). Sekvence fází je adaptivní, tedy následující fáze v sekvenci často závisí na výstupech z předcházející fáze. Nejdůležitější závislosti mezi fázemi jsou naznačeny šipkami. Pokud je například aktuální fáze modelování, pak v závislosti na chování a charakteristice modelu může být následující fází návrat do fáze přípravy dat pro vylepšení modelu před posunutím vpřed do fáze vyhodnocování. [1]

Iterativní stránka CRISP-DM je naznačena vnějším kruhem šipek. Často je řešení konkrétního obchodního/výzkumného problému pouze zdrojem dalších otázek, na které může být opět použit stejný proces. Zkušenosti z předchozích projektů by měly být použity jako vstup do nových projektů. [1]

Problémy, které nastanou během fáze vyhodnocování mohou způsobit návrat do kterékoliv z předcházejících fází. [1]

Jednotlivé fáze se skládají z několika kroků a jsou popsány níže.

1.3.1 Fáze pochopení obchodu/výzkumu

Kroky:

1. Přesné stanovení cílů a omezení projektu v termínech obchodu/výzkumné jednotky jako celku.
2. Přeložení těchto cílů a omezení tak, aby vznikla formulace, která definuje datamining problém.
3. Příprava předběžné strategie pro dosažení těchto cílů. [7]

1.3.2 Fáze pochopení dat

Kroky:

1. Sběr dat.
2. Použití výzkumné analýzy dat k seznámení se s daty a vznik počátečního náhledu.
3. Vyhodnocení kvality dat.

4. Výběr zajímavých podmnožin dat, které mohou obsahovat hledané vzory. [7]

1.3.3 Fáze přípravy dat

Tato časově náročná fáze obsahuje veškeré kroky, které je třeba provést pro získání finálního datasetu, na který budou aplikovány následující fáze.

Kroky:

1. Výběr případů a proměnných, které se budou analyzovat a jsou vhodné pro analýzu.
2. Provedení transformace proměnných, u kterých je třeba.
3. Čištění původních dat tak, aby byly vhodné pro modelovací nástroje. [7]

1.3.4 Fáze modelování

Kroky:

1. Výběr vhodných modelovacích technik.
2. Kalibrace nastavení modelu kvůli optimalizaci výsledků.
3. Často může být použito více modelovacích technik na jeden datamining problém.
4. Možný návrat zpět do fáze přípravy dat, aby bylo vyhověno požadavkům modelovacích technik. [7]

1.3.5 Fáze vyhodnocování

Kroky:

1. Vyhodnocení kvality a efektivity modelů z fáze modelování.
2. Vyhodnocení, zda bylo dosaženo cílů z fáze pochopení obchodu/výzkumu.
3. Rozhodnutí, zda nebyl zanedbán některý podstatný aspekt obchodu/výzkumu.
4. Rozhodnutí podle výsledků dataminingu. [7]

1.3.6 Fáze nasazení

Kroky:

1. Využití vytvořených modelů.
2. Příklad jednoduchého nasazení: Vygenerování výstupní zprávy.
3. Příklad komplexnějšího nasazení: Implementace paralelního procesu dataminingu na jiném oddělení.

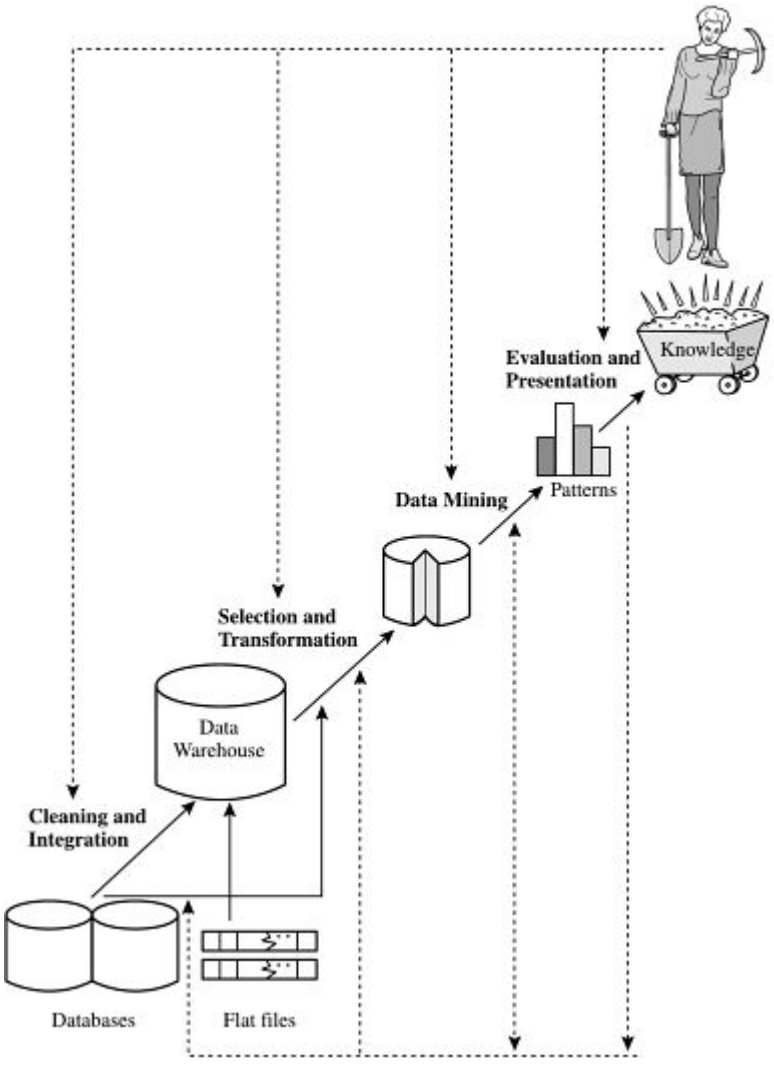
4. U obchodních problémů je nasazení většinou zodpovědností zákazníka. [7]

1.4 Proces dataminingu - kroky

V této části je uveden obecnější přístup k dataminingu, který je shodný s KDD.

Definice dataminingu (Def. 1) říká, že datamining je proces a jako takový se skládá z několika kroků.

Jednotlivé části procesu jsou zachyceny na obrázku (Obr. 2). Ačkoliv je na obrázku datamining zobrazen pouze jako jeden krok, stejně tak může být označen celý proces.



Obr. 2. Proces získávání informací. [2]

Význam popisků v obrázku (Obr. 2) je následující:

Cleaning and Integration – čištění a integrace, **Databases** – databáze, **Flat files** – záznamy, **Data Warehouse** – datový sklad, **Selection and Transformation** – selekce a

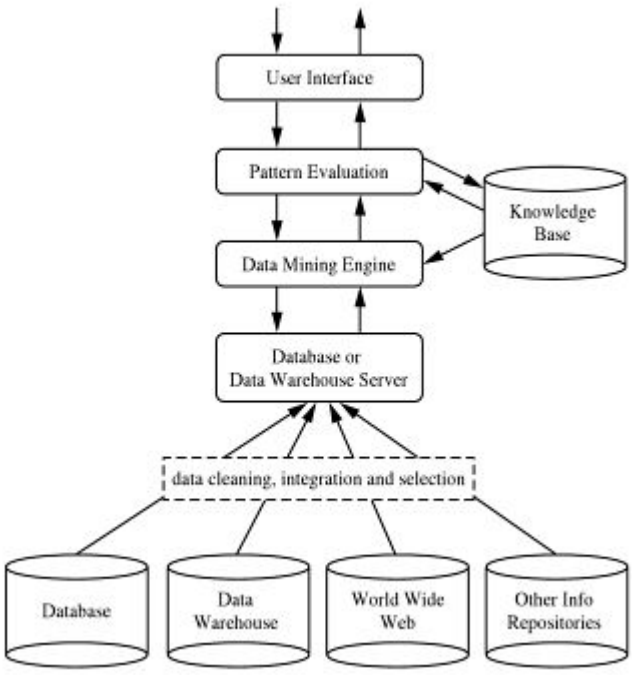
transformace, **Data Mining** – hledání informací v datech, **Evaluation and Presentation** – vyhodnocení a prezentace, **Patterns** – vzory, **Knowledge** – informace.

Proces získávání informací je iterativní a probíhá v následujících krocích:

1. **Čištění dat** – odstranění šumu a nekonzistentních dat.
2. **Integrace dat** – kombinace více datových zdrojů.
3. **Selekce dat** – výběr dat relevantních pro analýzu.
4. **Transformace dat** – data jsou přetransformována do podoby, která je vhodná pro algoritmus dataminingu.
5. **Datamining** – využití inteligentních metod pro nalezení užitečných vzorů a trendů.
6. **Vyhodnocení vzorů a trendů** – identifikace skutečně důležitých vzorů a trendů založená na vyhodnocení míry důležitosti.
7. **Prezentace informací** – prezentace informací získaných z dat.

Kroky 1 až 4 jsou různé formy předzpracování dat, kdy jsou data připravována pro datamining algoritmus. Algoritmus může spolupracovat s uživatelem nebo se znalostní bází. Důležité vzory a trendy jsou prezentovány uživateli a mohou být uloženy do znalostní báze. Nejdůležitějším krokem procesu je krok 5, který využívá datamining algoritmu. [2]

1.5 Architektura datamining systému



Obr. 3. Architektura běžného datamining systému. [3]

Význam popisků v obrázku (Obr. 3) je následující:

Database – databáze, **Data Warehouse** – datový sklad, **Other Info Repositories** – další informační repozitáře, **data cleaning, integration and selection** – čištění dat, integrace a selekce, **Database or Data Warehouse Server** – databázový server nebo server datového skladiště, **Data Mining Engine** – dataminingový nástroj, **Knowledge Base** – znalostní báze, **Pattern Evaluation** – modul vyhodnocování vzorů, **User Interface** – uživatelské rozhraní.

Architektura běžného datamining systému je vyobrazena na obrázku (Obr. 3) a její jednotlivé komponenty jsou popsány níže.

1.5.1 Databáze, datové sklady, World Wide Web a další informační repozitáře

Tato komponenta může být složena z jedné nebo více databází, datových skladů, tabulek nebo dalších zdrojů informací. Čištění dat a jejich integrace může být prováděna na této úrovni. [3]

1.5.2 Databázový server nebo server datového skladiště

Databázový server nebo server datového skladiště je zodpovědný za výběr relevantních dat na základě datamining požadavku uživatele. [3] Uživatel může například chtít zjistit, zda existuje spojitost mezi prodejem lyžařské výbavy a ročním obdobím. Databázový server nebo server datového úložiště tedy poskytne informace pouze o prodejích lyžařské výbavy, nikoliv o prodejích nafukovacích lehátek, které by byly pro potřeby řešení zadaného problému irelevantní.

1.5.3 Dataminingový nástroj

Komponenta nezbytná pro datamining systém. V ideálním případě se skládá z funkčních modulů zaměřených na úkoly jako: charakterizační, asociační a korelační analýza, klasifikace, predikce, analýza clusterů, analýza extrémů a evoluční analýza. [3]

1.5.4 Znalostní báze

Doména znalostí, která se používá k upřesnění hledání, nebo k určení míry zajímavosti nalezených vzorů a trendů. Báze může obsahovat hierarchii konceptů, které jsou použity pro organizaci atributů nebo hodnot atributů do různých úrovní abstrakce. Dalším typem dat ve znalostní bázi mohou být přesvědčení uživatele, která se využívají k určování míry

zajímavosti vzoru nebo trendu na základě jeho pravděpodobnosti. Dále zde mohou být konstanty určující míru zajímavosti, prahy a metadata. [3]

1.5.5 Modul vyhodnocování vzorů

Komponenta běžně využívá míru zajímavosti dat a spolupracuje s datamining modulem, který se tak zaměřuje pouze na vyhledávání zajímavých vzorů a trendů. Může využívat prahy míry zajímavosti pro filtrování objevených vzorů a trendů.

Modul může být také integrován do datamining modulu, což závisí na implementaci použité datamining metody. Pro efektivní datamining je doporučeno implementovat vyhodnocování míry zajímavosti vzorů a trendů co nejhlouběji do procesu, aby bylo zaručeno rychlé vyřazení těch nezajímavých. [3]

1.5.6 Uživatelské rozhraní

Komponenta komunikuje mezi uživateli a datamining systémem, umožňuje jim spolupracovat se systémem ve smyslu specifikace datamining dotazu nebo úkolu. Poskytuje informace pro upřesnění vyhledávání a vykonává průzkumný datamining na základě průběžných výsledků. Dále je uživatelům díky této komponentě umožněno procházení databází, schémat skladiště dat nebo datové struktury, vyhodnocování nalezených vzorů a trendů a jejich vizualizace v různých podobách. [3]

1.6 Úkoly dataminingu

Úkoly dataminingu specifikují, jaké vzory a trendy jsou hledány v datech. Obecně lze úkoly rozdělit na dvě kategorie: **deskriptivní** a **prediktivní**. Deskriptivní úkoly charakterizují vlastnosti dat v cílovém data setu. Prediktivní úkoly se zaměřují na předpověď budoucího vývoje na základě aktuálních dat. [2]

V této části jsou stručně popsány jednotlivé úkoly, jako jsou: Charakterizace a rozlišování, hledání frekventovaných vzorů, asociací a korelací, klasifikace a regrese, analýza clusterů a analýza extrémů. Dále je zde uveden pojem míry zajímavosti vzoru reprezentující hodnotu informace, kterou daný vzor poskytuje.

Pro větší názornost jsou v této části uvedeny příklady úkolů na datech získaných z fiktivního obchodu s elektronikou – **FakeElectro**.

1.6.1 Charakterizace a rozlišování

Vstupní data mohou být asociována s třídami nebo koncepty. V mnoha případech je potřeba užitečné třídy a koncepty stručně, ale přesně, shrnout. Takové shrnutí je nazváno deskripcí třídy/konceptu. Deskripce může být získána třemi způsoby:

1. **Charakterizací dat** – shrnutí dat zkoumané (cílové) třídy v obecných termínech.
2. **Rozlišováním dat** – porovnání cílové třídy s jednou nebo více srovnávacími (kontrastními) třídami.
3. **Kombinací charakterizace a rozlišování dat.** [2]

Charakterizace dat je shrnutí základních charakteristik a rysů dat cílové třídy. Data odpovídající třídě specifikované uživatelem jsou typicky získána pomocí dotazování. Příkladem může být zkoumání charakteristik software produktů, které v předchozím roce dosáhly navýšení prodeje o 10%. K získání takových dat může být použito SQL (Structured Query Language) dotazu na databázi prodejů.

Pro efektivní shrnutí a charakterizaci dat existuje několik metod: Jednoduché shrnutí dat založené na statistických úkonech a grafech, roll-up operace na OLAP kostce může být použita pro provedení uživatelem řízeného shrnutí dat podle specifikované dimenze (atributu), atributově orientovaná indukce lze použít pro generalizaci a charakterizaci dat bez interakce uživatele při každém kroku.

Výstup charakterizace dat může být prezentován v různých podobách. Příkladem jsou křivky, sloupcové grafy, koláčové grafy, multidimenzionální datové kostky a multidimenzionální tabulky. Výstupní popis může být také prezentován pomocí **generalizačních vztahů** nebo ve formě pravidel – **charakterizační pravidla**. [2]

Příklad chrakterizace dat

Zadání datamining úkolu: *Shrnutí charakteristik zákazníků, kteří ve FakeElectro ročně utratí více, než 100 000 Kč.*

Výsledkem je obecný profil těchto zákazníků, který udává, že jsou ve věku 30 až 40 let, zaměstnaní a mají vysoké příjmy. Dataminng systém by měl umožnit tato data dále upřesnit. A to například dále rozdělit zákazníky podle typu zaměstnání.

Rozlišování dat je porovnávání základních rysů cílové třídy se základními rysy jedné nebo více kontrastních tříd. Cílová a kontrastní třídy mohou být specifikovány uživatelem a odpovídající data jsou získána pomocí databázového dotazování. Příkladem může být

zkoumání základních rysů software produktů, u kterých za poslední rok stouply prodeje o 10%, proti produktům, u kterých za stejné období prodeje klesly o 30%. Metody používané pro rozlišování dat jsou shodné s metodami používanými pro charakterizaci dat.

Podoba výstupu je podobná jako u charakterizace dat, ale je doplněna o porovnávací prvky, které napomáhají k rozlišení cílové a kontrastních tříd. Rozlišovací popis dat vyjádřený ve formě pravidel se nazývá **rozlišovacími pravidly**. [3]

Příklad rozlišování dat

Zadání datamining úkolu: *Porovnání dvou skupin zákazníků FakeElectro. První skupina jsou zákaznící nakupující pravidelně (více než dvakrát za měsíc) a druhá skupina obsahuje zákazníky nakupující zřídka (méně, než třikrát ročně).*

Výsledný popis udává porovnání základních charakteristik profilů zákazníků v těchto dvou skupinách. V první skupině je 80% zákazníků ve věku 20 až 40 let a má univerzitní vzdělání, kdežto ve druhé skupině je 60% seniorů nebo mladistvých bez univerzitního vzdělání. Pomocí následného rozlišení zaměstnání, nebo přidání výše příjmu může být získáno přesnější rozlišení mezi skupinami.

1.6.2 Hledání frekventovaných vzorů, asociací a korelací

Frekventované vzory jsou už podle názvu vzory, které se v datech objevují často. Je mnoho druhů frekventovaných vzorů včetně frekventovaných itemsetů, frekventovaných subsekvencí (sekvenční vzory) a frekventovaných substruktur.

Frekventovaný itemset obvykle značí množinu objektů, které se často vyskytují současně v transakčních data setech. Příkladem může být obsah nákupního koše – mléko a chleba, zubní pasta, kartáček.

Příkladem frekvenčního vzoru je nákup notebooku, následovaný nákupem digitálního fotoaparátu a paměťové karty.

Substruktura může označovat skupinu různých strukturních forem (grafy, stromy nebo matice) kombinovanou s itemsety nebo subsekvencemi. Pokud se substruktura vyskytuje frekventovaně, je označována jako strukturní vzor.

Vyhledávání frekventovaných vzorů vede k objevení zajímavých asociací a korelací mezi daty. [2]

Příklad asociační analýzy

Zadání datamining úkolu: *Které produkty jsou ve FakeElectro nakupovány zároveň (v rámci jedné transakce).*

Příklad vzoru nalezeného v databází FakeEelectro může být následující:

Pravidlo 1.: *koupí(x, "tiskárna") → koupí(x, "papír") [podpora = 2%, jistota = 46%]*

Proměnná *x* v pravidle značí zákazníka, **jistota** 46% znamená, že pokud zákazník koupí tiskárnu, je 46% šance, že koupí zároveň i papír. **Podpora** 2% značí, že 2% všech analyzovaných transakcí obsahuje nákup tiskárny a papíru zároveň. Takto zadané asociační pravidlo obsahuje jeden predikát a je proto označováno jako **jednodimenzionální asociační pravidlo**. Vynecháním predikátové notace může být pravidla zapsáno jednoduše: *"tiskárna → papír [2%, 46%]."*

Zadání datamining úkolu: *Nalezení pravidla pro nákup notebooku v databázi nákupů ve FakeElectro.*

Příklad nalezeného asociačního pravidla:

Pravidlo 2.: *příjem(x, "40 000 až 52 000") ∧ věk(x, "25 až 32") → koupí(x, "notebook") [podpora = 3%, jistota = 55%]*

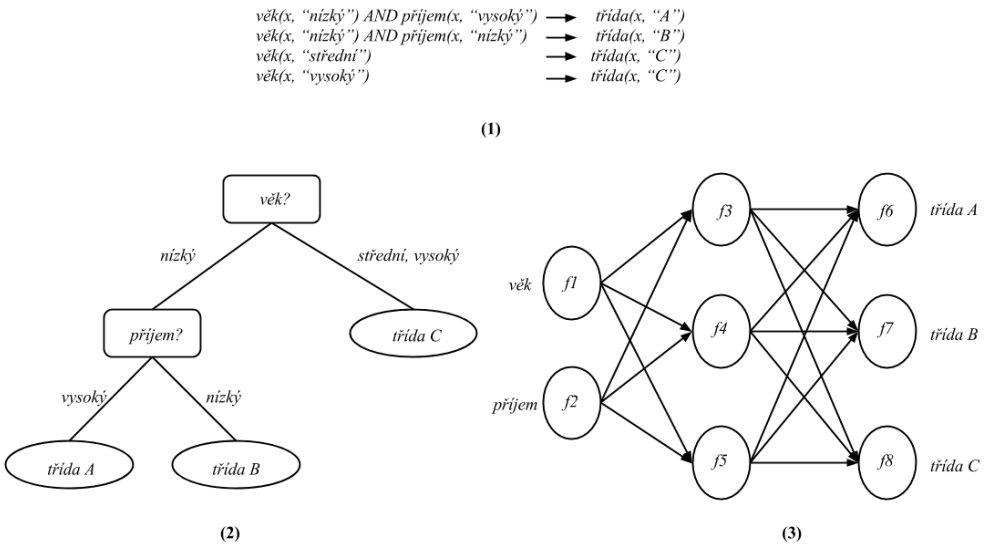
Výklad pravidla říká, že ze všech zákazníků v datasetu jsou 3% ve věku 25 až 32 let s příjmem 40 000 – 50 000 Kč měsíčně. Pravděpodobnost, že zákazník v tomto věkovém a platovém rozsahu koupí notebook je 55%. Tato asociace již obsahuje více, než jeden atribut nebo predikát (příjem, věk a nákup). Adaptací terminologie multidimenzionálních databázových systémů, kde je každý atribut označován jako dimenze může být pravidlo (Pravidlo 2) označeno termínem **multidimenzionální asociační pravidlo**. [2]

Aby mohlo být asociační pravidlo označeno za zajímavé, je potřeba, aby splnilo podmínky **minimální úrovně podpory** a **minimální úrovně jistoty**. Použitím dalších analýz lze nalézt zajímavé statistické korelace mezi asociovanými páry atribut-hodnota. [3]

1.6.3 Klasifikace a regrese pro prediktivní analýzu

Klasifikace je proces nalezení **modelu** (nebo funkce), který popisuje a rozlišuje datové třídy a koncepty. Model je odvozen na základě analýzy množiny **trénovacích dat** (objekty, u kterých je známo, do které třídy patří). Model se používá k predikci třídy u objektů, u kterých není známa. [2]

Model lze vizualizovat v různých podobách: Klasifikační pravidla (např. IF-THEN pravidla), rozhodovací stromy, matematické formulace, nebo neuronové sítě. Jak taková reprezentace vypadá, je naznačeno na obrázku (Obr. 4).



Obr. 4. Různé reprezentace klasifikačního modelu. (1) IF-THEN pravidla, (2) rozhodovací strom, (3) neuronová síť.

Rozhodovací strom je stromová struktura podobná vývojovému diagramu, kde každý uzel reprezentuje test hodnoty atributu, každá větev výstup z testu a listy stromu reprezentují třídy. Rozhodovací stromy mohou být převedeny na **klasifikační pravidla**. **Neuronová síť**, použitá pro klasifikaci, se skládá z neuronů s váhovanými synapsemi. Pro klasifikaci dat lze použít mnoho dalších metod, jako například naivní Bayesův klasifikátor, SVM (Support Vector Machine) a klasifikace pomocí algoritmu k-nejbližších sousedů. [3]

Klasifikace určuje diskrétní hodnotu (příslušnost ke třídě), kdežto regrese modeluje funkce spojité. Regrese se používá k odhadu chybějících nebo nedostupných hodnot atributů dat. Termín predikce se používá pro oba typy výstupů, jak pro diskrétní, tak pro spojité hodnoty. Regresní analýza je statistický nástroj, který se nejčastěji používá pro numerickou predikci. Regrese taktéž zahrnuje i identifikaci rozložení dat. [3]

Klasifikaci i regresi často předchází analýza relevance, která se snaží identifikovat relevantní atributy pro klasifikaci/regresi, tyto atributy jsou použity pro klasifikační/regresní proces. Ostatní atributy mohou být zanedbány.

Příklad klasifikace a regrese

Zadání datamining úkolu: *Klasifikace produktů FakeElectro do tříd na základě odezvy na prodejní kampaň (dobrá odezva, mírná odezva, žádná odezva). Model má být vytvořen na základě následujících rysů produktů: cena, značka, země původu, typ a kategorie. Výsledná klasifikace by měla co nejvíce rozlišit jednotlivé třídy.*

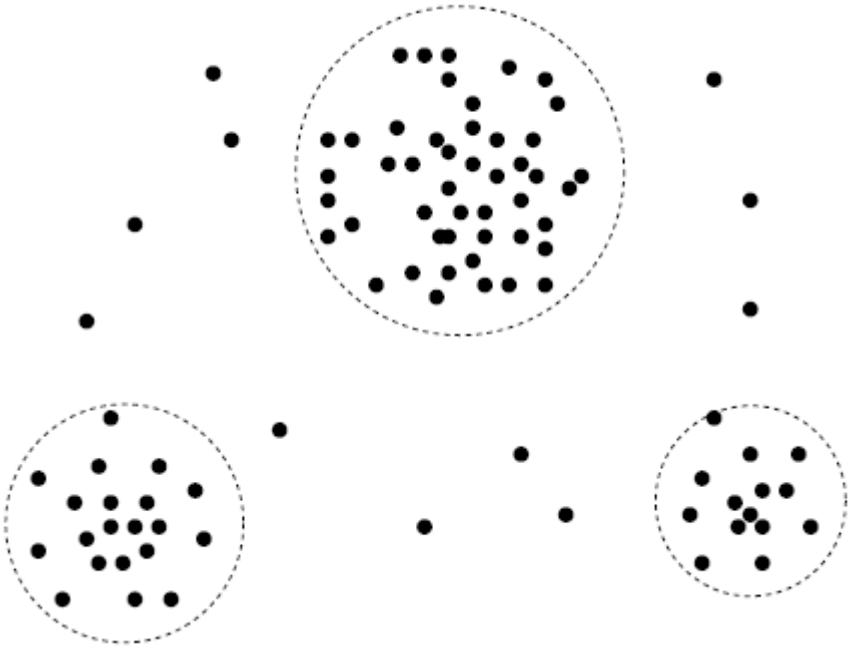
Příklad výsledku klasifikace:

Výsledek je prezentován ve formě rozhodovacího stromu, kde nejddůležitějším faktorem pro klasifikaci je *cena*. Další rysy, které pomáhají rozlšit mezi třídami mohou být *značka* a *země původu*. Tento výsledek může být nápomocen pro vytvoření více efektivní prodejní kampaně.

Za předpokladu, že se zadání lehce změní a cílem bude určení zisku jednotlivých produktů z výprodeje ve FakeElectro, je vhodné použít regresní analýzu. [2]

1.6.4 Analýza clusterů

Klasifikace a regresní analýza vytvářejí model na základě množiny trénovacích dat. Analýza clusterů naopak analyzuje data bez předchozích znalostí, protože v mnoha případech množina trénovacích dat není známa. Právě proto může být analýza clusterů použita k nalezení tříd charakterizujících data. Data jsou sdružována do tříd na základě principu **maximalizace podobnosti uvnitř třídy a minimalizace podobnosti vně třídy**. Což znamená, že shluky (clustery) objektů jsou vytvářeny tak, že objekty uvnitř clusteru jsou si velmi podobné, ale s objekty v jiných clusterech mají málo společného. Každý takový cluster lze považovat za třídu objektů, z nichž lze vyvodit pravidla. [3]



Obr. 5. Příklad cluster analýzy na dvoudimenzionálních datech. Tečkovaná čára ohraničuje jednotlivé clustery. [2]

Příklad analýzy clusterů

Zadání datamining úkolu: *Identifikace subpopulací zákazníků FakeElectro.*

Příklad výsledku:

Příkladem může být obrázek (Obr. 5), na kterém se zákazníci vyskytují především ve třech oblastech (podle bydliště). Takový výsledek je vhodný například pro cílený marketing v těchto oblastech.

1.6.5 Analýza extrémů

Extrémy jsou objekty v množině dat, které mají hodnoty atributů výrazně odlišné od zbytku dat, nebo neodpovídají modelu. Velké množství datamining metod se extrémy nezabývá a vyřazuje je z datasetu jako šum nebo vyjímky, nicméně existují aplikace (např. detekce podvodu), kde jsou takové unikátní objekty mnohem více zajímavé, než objekty odpovídající představám. Analýza extrémních dat je nazývána analýzou extrémů, nebo také hledáním anomálií. [2]

Extrémy mohou být detekovány pomocí statistických testů, které předpokládají rozložení množiny nebo pravděpodobnostní model dat, nebo také použitím míry vzdálenosti, kde jsou objekty nevyskytující se v clusterech považovány za extrémy. [2]

Příklad analýzy extrémů

Analýza extrémů může odhalit podvodné používání kreditní karty. Typickým příkladem je detekce většího množství odchozích plateb o nezvykle vysoké výši oproti klasickému využívání účtu. Dalším příkladem je změna lokace nákupů (odlišné státy) nebo frekvence (častější) plateb. [2]

1.6.6 Míra zajímavosti vzoru

Datamining systém má potenciál generovat veké množství vzorů a pravidel. Většina z nich je ovšem z hlediska dané problematiky nezajímavá, proto existuje míra zajímavosti vzoru.

Aby byl vzor zajímavý, musí splňovat následující požadavky:

1. Je pochopitelný pro uživatele.
2. Platí na nových nebo testovacích datech s určitým stupňem jistoty.
3. Je potencionálně použitelný.
4. Je nový.

Vzor může být taktéž zajímavý, pokud potvrzuje uživatelem předem stanovenou hypotézu. Zajímavý vzor reprezentuje znalosti. [2]

Existuje několik objektivních způsobů, jak měřit míru zajímavosti vzoru. Tyto způsoby jsou založeny na struktuře nalezených vzorů a na statistice. Objektivní způsob určení míry zajímavosti u asociačního pravidla ($X \rightarrow Y$) je **podpora**, která udává procento transakcí transakční databáze, které splňují toto pravidlo. Což je vlastně pravděpodobnost $P(X \cup Y)$, kde $X \cup Y$ říká, že transakce obsahuje jak X, tak Y, což je sjednocení množin X a Y. Dalším způsobem určení míry zajímavosti u asociačních pravidel je **jistota**, která určuje jistotu nalezené asociace. Formálně je to pravděpodobnost $P(Y \mid X)$, což je pravděpodobnost, že transakce obsahující X obsahuje také Y. [2]

Obecně je míra zajímavosti spojena s prahem, který může určit uživatel. Pravidla, která nepřekročí práh jistoty např. 50% mohou být považována za nezajímavá. Pravidla nesplňující prahové hodnoty odrážejí šum a vyjímky a pravděpodobně jsou méně hodnotná. [2]

Další objektivní způsoby určení míry zajímavosti jsou **přesnost** a **pokrytí**, užívané u klasifikačních pravidel. Přesnost určuje, kolik procent dat je správně klasifikováno

pravidlem a pokrytí je podobné podpoře. Pokrytí udává pocento dat, na které se pravidlo aplikuje. [3]

Ačkoliv objektivní způsoby pomáhají identifikovat zajímavé vzory, bývají neefektivní, pokud se nekombinují se subjektivními způsoby, které odrážejí konkrétní požadavky a zájmy uživatelů. Různí uživatelé mají různé nároky a proto je pro ně zajímavost vzoru subjektivní. Mnoho objektivně zajímavých vzorů je také pouze reprezentací zdravého rozumu a tudíž jsou v důsledku nezajímavé. [3]

Subjektivní způsoby určení míry zajímavosti jsou pevně spjaty s konkrétní představou uživatele o datech. Subjektivně zajímavé vzory jsou ty, které jsou **neočekáváné** (popření představy uživatele) nebo jsou spojeny se strategickou informací, na základě které uživatel může jednat – **akční** vzory. Příkladem akčních vzorů je zemětřesení: "Po sérii slabých zemětřesení často nastává zemětřesení silnější." Nalezení takového vzoru a jeho vyhodnocení může předejít ztrátám na životech. Taktéž vzory, které jsou **očekávané**, mohou být zajímavé. V případě, že potvrzují uživatelem stanovenou hypotézu. [3]

Úplnost datamining algoritmu udává množství identifikovaných zajímavých vzorů. Často je neefektivní a nereálné, aby datamining systém generoval všechny možné vzory. Místo toho se využívá uživatelem specifikovaných konstant a míry zajímavosti pro upřesnění hledání. Hledání asociačních pravidel je úkol, u kterého uživatelem specifikované hodnoty a míry zajímavosti postačují k dosažení úplnosti. [2]

Nalezení pouze zajímavých vzorů je optimalizační problém. U většiny datamining systémů je požadováno, aby byly nalezeny pouze takové vzory, což zvyšuje efektivitu těchto systému. Oblast optimalizace je stále velmi atraktivní oblastí datamining systémů. [2]

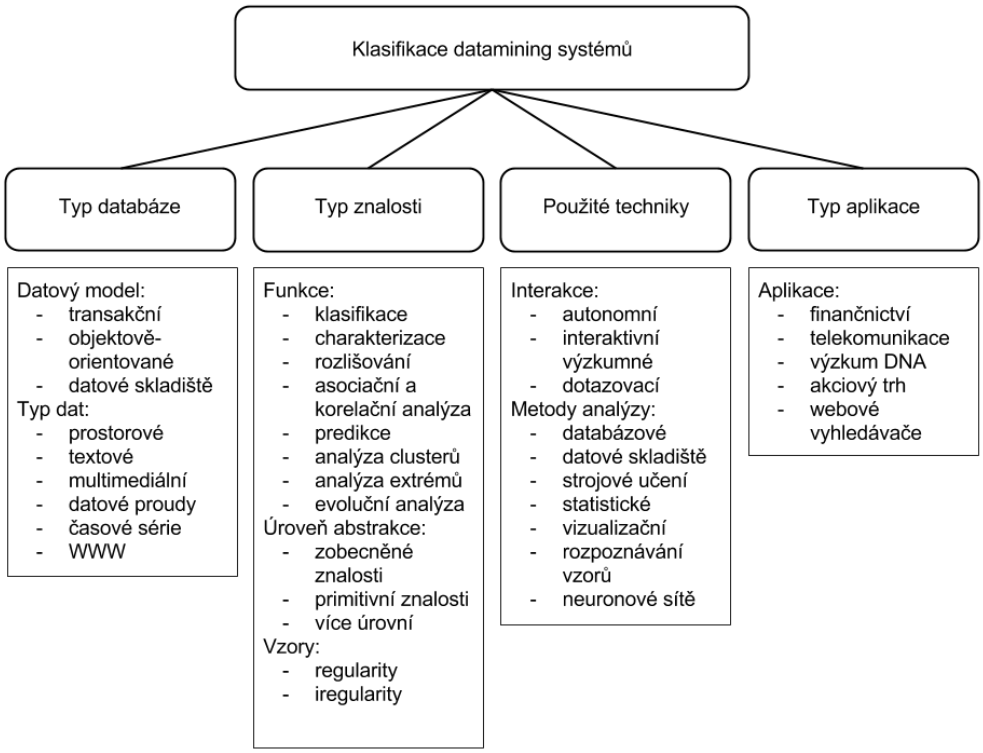
Míra zajímavosti vzoru je nezbytná pro efektivní nalézání vzorů požadovaných uživateli. Je využívána i nadále, po skončení datamining algoritmu, pro řazení a filtrování vzorů. Důležitější ovšem je, že je využívána v průběhu pro vedení a upravování procesu nalézání vzorů, pro zvýšení efektivity hledání vyřazováním částí množiny vzorů, které nesplňují požadavky na míru zajímavosti. [2]

1.7 Klasifikace datamining systémů

Datamining je mezidisciplinární obor, který spojuje více technik z oblasti databázových systémů, statistiky, strojového učení, vizualizace a informační vědy. V závislosti na přístupu mohou být použity i techniky z jiných oblastí, jako jsou neuronové sítě, teorie

fuzzy množin, teorie hrubých množin, reprezentace znalostí, induktivní logické programování, nebo supervýpočty. V závislosti na struktuře dat, která mají být analyzována nebo na konkrétní aplikaci, může datamining systém využívat i techniky z oblasti analýzy prostorových dat, získávání informací, rozpoznávání vzorů, analýzy obrazu, zpracování signálů, počítačové grafiky, webových technologií, ekonomie, bioinformatiky, nebo psychologie. Vzhledem k rozdílnosti jednotlivých oborů, které datamining využívá, vzniká velké množství datamining systémů, které je potřeba klasifikovat. [3]

Klasifikace podle rozličných kritérií je zobrazena na obrázku (Obr. 6) a podrobněji popsána níže.



Obr. 6. Klasifikace datamining systémů.

1.7.1 Klasifikace podle typu databáze

Datamining systém může být klasifikován podle typu databáze, na které má pracovat. Databázové systémy se dělí podle různých kritérií (typ dat, datový model, aplikace), kde každé kritérium může vyžadovat rozdílný datamining systém. Datamining systémy lze tedy dělit stejně, jako databázové systémy. Pokud se jedná o klasifikaci na základě datového modelu, mohou být datamining systémy děleny na transakční, objektově-orientované, nebo

systémy datových skladišť. Při klasifikaci podle typu dat se může jednat o systémy prosotorové, textové, multimediální, datových proudů, časových sérií nebo WWW (World Wide Web). [3]

1.7.2 Klasifikace podle typu znalostí

Podobně jako podle typu databáze, mohou být datamining systémy klasifikovány podle typu znalostí, které se snaží vytěžit. Jedná se o dělení podle datamining funkcí, jako jsou klasifikace, charakterizace, rozlišování, asociační a korelační analýza, predikce, analýza clusterů, analýza extrémů a evoluční analýza. Rozsáhlejší datamining systémy většinou integrují více funkcionalit zároveň. Dále mohou být systémy rozděleny podle úrovně abstrakce získávaných znalostí: zobecněné znalosti (vysoká úroveň abstrakce), primitivní znalosti (úroveň čistých dat – bez abstrakce), znalosti na více úrovních (různá míra abstrakce). Pokročilý datamining systém by měl dokázat získat znalosti na více úrovních abstrakce. [3]

Podobně mohou být datamining systémy rozděleny podle toho, jestli nalézají regularity (běžně se vyskytující vzory) nebo iregularity (vyjímky, extrémy). Deskripce konceptu, asociační a korelační analýza, klasifikace, predikce a analýza clusterů obecně nalézají regularity a extrémy považují za šum. [3]

1.7.3 Klasifikace podle použitých technik

Dalším způsobem klasifikace datamining systémů je klasifikace podle použitých datamining technik. Techniky mohou být rozděleny podle úrovně interakce uživatele (autonomní systémy, interaktivní výzkumné systémy, dotazovací systémy) nebo podle metod analýzy dat (databázové, metody datových skladišť, metody strojového učení, statistické, vizualizační, metody rozpoznávání vzorů, metody neuronových sítí, apod.). Moderní datamining systémy pracují s více technikami a kombinují je pro zajištění lepších výsledků. [3]

1.7.4 Klasifikace podle typu aplikace

Datamiming systémy mohou být zaměřeny na konkrétní aplikace v konkrétních odvětvích. Příkladem mohou být aplikace ve finančnictví, telekomunikacích, výzkumu DNA (Deoxyribonukleová kyselina), aplikace pro akciový trh, webové vyhledávače, atd. Různé aplikace obvykle vyžadují integraci specifických metod a proto je použití univerzálního datamining systému nevhodné. [3]

1.8 Datamining problémy

Datamining je velmi dynamická a rychle se rozvíjející oblast s obrovským potenciálem. A jako taková se skládá z několika základních problémů, které je možné rozdělit do pěti skupin – metodologie, interakce, efektivita a rozšiřitelnost, rozdílnost datových typů a společnost a datamining. Řešení těchto problémů bylo v poslední době cílem vývoje a výzkumu a proto se z některých problémů staly spíše požadavky na datamining systém. Problémy a nalézání jejich řešení přinášejí neustálé zlepšování technik dataminingu. [2]

1.8.1 Metodologie

Vývoj nových datamining metodologií zahrnuje zkoumání nových druhů znalostí, hledání v multidimenzionálních prostorech, intergaci metod z dalších oborů a zvažování sémantických vazeb mezi datovými objekty. Dále by měly datamining metodologie pokrývat problémy, jako jsou nejistota dat, šum a nekompletní data. Některé metody zkoumají, jak mohou být uživatelem specifikované hodnoty použity pro určení zajímavosti vzoru a upřesnění procesu hledání takových vzorů. Níže jsou uvedeny problémy spadající do kategorie metodologie. [2]

1.8.1.1 Hledání různých a nových znalostí

Datamining pokrývá široké spektrum úkolů zabývajících se analýzou dat a nalézáním znalostí od charakterizace a rozlišování po asociační a korelační analýzu, klasifikaci, regresi, analýzu clusterů, analýzu extrému, sekvenční analýzu, rozpoznávání trendů a evoluční analýzu. Všehchny tyto úkoly mohou pracovat se stejnou databází, ale každý jinak a je třeba vyvinout datamining techniku pro každý úkol zvlášť. Díky rozličnosti aplikací vzniká stále více takových úkolů a efektivně se využívá jejich hybridizace a paralelizace. Příkladem může být efektivní nalezení znalostí v informační síti. Využití integrované analýzy clusterů a hodnotící (ranking) funkce může vést k nalezení kvalitních clusterů a vysoce hodnocených objektů. [2, 3]

1.8.1.2 Hledání znalostí v multidimenzionálním prostoru

Jedná se o hledání znalostí ve velkých datasetech. Vyhledávají se zajímavé vzory v kombinacích několika atributů (dimenzí) dat na různé úrovni abstrakce, což je nazýváno – multidimenzonální datamining. Data mohou být agregována nebo ve formě

multidimenzionální datové kostky. Hledání znalostí na datových kostkách může výrazně zvýšit výkonnost a fleixibilitu dataminingu. [2]

1.8.1.3 Mezioborová snaha

Výkonnost datamining algoritmu může být výrazně zvýšena kombinací metod z více oblastí. Například při prohledávání dat v přirozeném jazyce je vhodné použít metody vyhledávání informací a zpracování přirozeného jazyka. [2]

1.8.1.4 Zvýšení výkonnosti hledání v síťovém prostředí

Velké množství datových objektů se vyskytuje v propojeném prostředí, ať už se jedná o web, databáze nebo soubory. Sémantické spojení mezi objekty může sloužit ke zvýšení výkonnosti datamining procesu a znalosti nalezené na konkrétní množině dat mohou být použity pro zlepšení hledání znalostí na množině objektů spojené s původní množinou. [2]

1.8.1.5 Práce s nejistotou, šumem a nekompletními daty

Data se v přirozené podobě vyskytují včetně šumu, chyb, vyjímek, nejistot nebo mohou být nekompletní. Všechny tyto nedokonalosti mohou vést ke zhoršení výsledků datamining procesu. Proto se v datamining procesu používají techniky čištení dat, preprocessing (předzpracování), detekce a odstraňování extrémů a řešení nejistoty. [2]

1.8.1.6 Vyhodnocování vzorů a vzorově-řízené hledání

Jak už bylo zmíněno výše, všechny nalezené vzory v datamining procesu nemusí být zajímavé. Míra zajímavosti je pro uživatele subjektivní a proto je potřeba do datamining procesu integrovat subjektivní techniky. Tyto techniky odhadují míru zajímavosti vzoru podle přesvědčení a očekávání uživatele. Použitím uživatelem specifikovaných konstant pro míru zajímavosti je možné změnšit prostor možných nalezených vzorů a tím docílit zvýšení výkonu datamining procesu. [2, 3]

1.8.2 Interakce

Uživatel hraje v datamining pocesu velmi důležitou roli a následující oblasti jsou pro výzkum velmi důležité – jak pracovat s datamining systémem, jak do datamining systému začlenit uživatelské znalosti, jak vizualizovat a interpretovat výsledky. [2]

1.8.2.1 Interaktivní hledání

Datamining proces by měl být vysoce interaktivní a proto musí zahrnovat flexibilní uživatelské rozhraní. Uživateli musí být umožněno vybrat vzorek dat, prozkoumat základní charakteristiky a následně předpovědět potencionální výsledek. Interaktivní datamining by měl uživateli poskytnout možnost dynamicky měnit oblast prohledávání pro upřesnění specifikace datamining požadavků na základě průběžných výsledků a následné operace nad datovou kostkou a znalostním prostorem. [2]

1.8.2.2 Zahrnutí znalostí

Znalostní báze, konstanty, pravidla a další informace vztahující se ke zkoumané oblasti by měly být zahrnuty do procesu hledání znalostí. Tyto informace lze použít k vyhodnocování vzorů nebo k navigaci vyhledávacího procesu směrem k zajímavým vzorům. [2]

1.8.2.3 Přímý datamining a dotazovací jazyk

Stejně jako jsou dotazovací jazyky (např. SQL) důležité pro flexibilní vyhledávání a umožňují uživateli přímé dotazování, vyšší dotazovací jazyky pro datamining nebo jiná flexibilní uživatelská rozhraní poskytují uživateli volnost v definování přímých datamining úkolů. Dále by měly zahrnovat možnost specifikace relevantní množiny dat pro analýzu, domény znalostí, specifikaci typů znalostí, které se vyhledávájí a specifikaci podmínek a omezení pro nalezené vzory. Optimalizace zpracování těchto úkolů je vyvíjející se oblastí dataminingu. [2]

1.8.2.4 Prezentace a vizualizace výsledků

Prezentace a vizualizace výsledků datamining procesu, tak aby bylo jednoduché pochopit nalezené znalosti, je stěžejní částí interaktivního datamining systému. V systému musí být zavedeny expresivní reprezentace znalostí a uživatelsky-přívětivá rozhraní a vizualizační techniky. [2]

1.8.3 Efektivita a rozšiřitelnost

Efktivita a rozšiřitelnost datamining systémů jsou hlavní vlastnosti, které slouží k jejich porovnávání. Vzhledem ke stále narůstajícímu objemu dat, které je potřeba zpracovat, jsou tyto vlastnosti velmi důležité.

1.8.3.1 Efektivita a rozšiřitelnost datamining algoritmu

Aby bylo možné vytěžit informace z velkého množství dat z různých zdrojů, nebo dynamických datových proudů, je důležité, aby byl datamining algoritmus efektivní a rozšiřitelný. Pro potřeby aplikací je podstatné, aby byl výpočetní čas algoritmu krátký a předvídatelný. Efektivita, rozšiřitelnost, výkonnost, optimalizace a schopnost pracovat v reálném čase (datové proudy) jsou klíčová kritéria, díky kterým probíhá neustálý vývoj nových datamining algoritmů. [2, 3]

1.8.3.2 Paralelní, distribuované a inkrementální algoritmy

Ohromná velikost datasetů, data rozdělená na více úložištích a vysoká výpočetní náročnost některých datamining algoritmů jsou faktory, které motivují vývoj paraleleních a distribuovaných algoritmů. Tyto algoritmy nejdříve rozdělí data na menší části a každá taková část je zpracována paralelně. Paralelní procesy mohou spolupracovat a paralelně nalezené vzory jsou nakonec sloučeny. Další novou oblastí je provádění výpočtů v cloudu, kdy velké množství výpočetních jednotek spolupracuje na zpracování náročných datamining úkolů. [2, 3]

Náročnost datamining procesu a ikrementální povaha dat daly za vznik inkrementálním algoritmům, které zpracovávají data po menších částech, které jsou do systému zaváděny (přírůstky). Původně nalezené vzory a znalosti se upravují podle těchto přírůstků.

1.8.4 Rozdílnost datových typů

Široká škála různých typů databází a datových typů představuje taktéž nové oblasti ve výzkumu datamining systémů.

1.8.4.1 Zpracování různých typů dat

Rozdílné aplikace produkují velké množství rozdílných datových typů, ať už se jedná o data strukturovaná, jako jsou relační data a data datových skladišť, data částečně strukturovaná a nestrukturovaná, stálá nebo dynamická data (datové proudy), jednoduché datové objekty, dočasná data, biologické sekvence, prostorová data, hypertextová data, multimediální data, data v programovacím jazyce, webová data nebo data ze sociálních sítí. Kvůli takové rozdílnosti dat nelze použít univerzální datamining nástroj, ale spíše vznikají konkrétně zaměřené datamining nástroje, které se zabývají pouze určitou oblastí, datovými typy a nalézání znalostí v nich. [2, 3]

1.8.4.2 Zpracování dynamických, síťových a globálních datových repozitářů

Globální informační systémy a sítě (ohromné distribuované celky) jsou tvořeny spojováním více zdrojů dat pomocí Internetu. Nalézání vzorů a znalostí v takto rozdílných datových zdrojích obsahujících strukturovaná, částečně strukturovaná a nestrukturovaná propojená data je velmi složité, nicméně výsledky na obrovských propojených sítích mohou vést k nalezení vzorů, které by nebylo možné nalézt na menších, homogenních datasetech. Právě proto je dnes vývoj datamining systémů pracujících s webem a více informačními zdroji zajímavou a rozvíjející se oblastí výzkumu. [2]

1.8.5 Společnost a datamining

Vliv dataminingu na společnost, ať už v rámci zachování soukromí jednotlivce nebo v rámci zneužívání získaných znalostí je taktéž velmi důležitým tématem.

1.8.5.1 Vliv dataminingu na společnost

Se stále rostoucím využíváním datamining technik roste také počet možností zneužití získaných znalostí. Jak využít datamining pro prospěch společnosti, zabránit zneužívání a narušování soukromí jednotlivců, jsou problémy, které musí moderní datamining systémy řešit. [2, 3]

1.8.5.2 Datamining zachovávající soukromí

Datamining pomáhá ve vědeckém výzkumu, finančním, ekonomickém i bezpečnostním sektoru (odhalování problémů v realném čase). Na druhou stranu vzniká velké riziko zpřístupnění osobních dat. Základní filozofií datamining technik je zajistit dostatečnou bezpečnost citilivých dat a zachování soukromí jednotlivce při úspěšném provádění dataminingu. [2]

1.8.5.3 Skrytý datamining

Nelze předpokládat, že by všichni uživatelé ovládali datamining techniky a využívali je v praxi, proto je snaha impelementovat datamining do aplikací skrytě, tak aby mohl uživatel využívat jeho výhod bez nutné znalosti metod a algoritmů. Příkladem jsou, již dnes, webové vyhledávače nebo internetové obchody, které využívají informací o předchozím chování uživatele pro upřesnění výsledků hledání nebo nabízení produktů, o které by mohl mít uživatel zájem. [2]

2 DATA

Data se v reálném světě vyskytují v různých podobách a většinou obsahují šum, extrémy a vyjímky. Před zpracováním dat pomocí datamining technik je třeba data dostat do podoby, ve které budou vhodná pro datamining algoritmus – preprocessing. Aby bylo možné preprocessing provést, je třeba znát různé formy **atributů dat** a způsoby jejich **statistického popisu**. Pro zjednodušení pochopení dat a spojitostí mezi nimi se používá různých **vizualizačních technik**. **Podobnost** dat je důležitá pro analýzu clusterů, vyhledávání extrémů a klasifikaci dat. Všechny tyto termíny jsou podrobněji popsány v této části.

2.1 Atributy

Datasety jsou tvořeny datovými objekty, které reprezentují konkrétní entity. Datové objekty jsou běžně popsány pomocí atributů. V databázích je jeden datový objekt reprezentován řádkem tabulky a jednotlivé sloupce reprezentují atributy. [2]

Def. 2.: *Atribut je datové pole popisující charakteristiku nebo vlastnost datového objektu. [2]*

V literatuře se často zaměňují slova atribut, dimenze, vlastnost a proměnná. Označení atribut je používáno v dataminingu, dimenze v oblasti datových skladišť, vlastnost v oblasti strojového učení a proměnná ve statistice. Množina atributů popisující datový objekt se nazývá vektor atributů (vektor vlastností). Rozdělení dat podle jednoho atributu se nazývá univariantní, podle dvou - bivariantní, atd. Typ atributu je určen hodnotami, kterých může atribut nabývat. [2]

2.1.1 Nominální

Nominální atributy nabývají hodnot popsaných symboly, které nelze seřadit, každá hodnota popisuje kategorii nebo stav objektu. Bývají nazývány též kategorické. Typickým příkladem jsou výčtové typy v programovacích jazycích. Na nominálních atributech nelze provádět matematické operace s použitelným výsledkem, ze základních statistických metod lze smysluplně použít pouze modu (hodnota, která se ve statistickém souboru vyskytuje nejčastěji). [2]

Příkladem nominálního atributu z reálného světa je barva očí.

2.1.2 Binární

Binární atribut je nominální atribut, který může nabývat pouze dvou stavů: 0 a 1, pokud se jedná o typ Boolean, pak nabývá hodnot *pravda (true)* a *nepravda (false)*. Stav 0 běžně vyjadřuje nepřítomnost atributu. Binární atribut je symetrický, pokud jsou oba stavy stejně hodnotné a mají stejnou váhu (pohlaví: muž, žena), pokud stavy nejsou stejně důležité, jedná se o asymetrický binární atribut (výsledek HIV (Human Immunodeficiency Virus) testu: negativní, pozitivní). U asymetrických binárních atributů se méně očekávaná hodnota běžně kóduje jako 1. [2]

2.1.3 Ordinální

Ordinální atribut je takový, jehož hodnoty lze seřadit, ale nelze z nich určit vzájemnou vzdálenost. Ordinální atributy jsou vhodné pro uchování vlastností dat, které nelze objektivně popsat (míra spokojenosti studentů s vyučujícím). Ordinálních hodnot mohou nabývat atributy diskretizované ze spojitých veličin do několika skupin. Použitelné statistické údaje u ordinálních atributů jsou medián (střední hodnota seřazené posloupnosti) a modus, kdežto průměr není definován. [2]

Nominální, binární a ordinální atributy jsou *kvalitativní* – popisují vlastnost objektu bez udání kvantity. Hodnoty kvalitativních atributů jsou často vyjádřeny slovně, pokud jsou použita čísla, jedná se o kódování skupin pro počítačové zpracování. [2]

2.1.4 Numerické

Numerické atributy jsou *kvantitativní*, tedy popisují kvantitu v celých nebo reálných číslech. Oproti kvalitativním atributům je u numerických atributů možné určit rozdíl mezi dvěma hodnotami.

Atributy s intervalovým měřítkem jsou měřeny na stupnici o stejně velkých jednotkách. Hodnoty jsou seřazeny a mohou být pozitivní, nulové, nebo negativní. Příkladem je teplota ve stupních Celsia, která nemá pravý nulový bod, neboť 0°C není stav bez teploty. Takže lze vyjádřit rozdíl teplot, ale nikoliv poměr. Poměr vyjadřují **atributy s poměrovým měřítkem**, které mají vlastní nulu a jednotlivé hodnoty lze vyjádřit poměrově, tedy že jedna hodnota je násobkem jiné. Příkladem atributu s poměrovým měřítkem je počet slov v dokumentu. [2]

U atributů s intervalovým i poměrovým měřítkem je možné vyjádřit jejich průměr, modus a medián.

2.1.5 Diskrétní a spojité

Další možností, jak rozdělit atributy, je rozdělení na diskrétní a spojité. **Diskrétní** atribut má spočitatelně velkou množinu různých stavů, ve kterých se může nacházet. Spočitatelně velká množina je taková, ve které je možné mít nekonečně mnoho hodnot, ale jednotlivé hodnoty je možné vyjádřit přirozenými čísly. **Spojitý** atribut může nabývat nekonečně mnoha stavů a nejčastěji se vyjadřuje pomocí realných čísel. [2]

2.2 Statistický popis dat

Základní statistický popis dat a jeho znalost je velmi důležitá pro správné provedení preprocessingu. Pomocí statistického popisu lze nalézt extrémy a šum, které je třeba odstranit. [2]

2.2.1 Centrální tendence

K měření centrální tendence dat se používají různé způsoby, zde jsou uvedeny následující čtyři:

1. **Průměr** – součet hodnot dělený jejich počtem (1).
2. **Medián** – střední hodnota seřazené posloupnosti.
3. **Modus** – nejčastěji vyskytující se hodnota.
4. **Mid-range** – součet minima a maxima hodnot dělený dvěma (2).

Nejběžnějším způsobem, jak určit střed dat je pomocí průměru. Někdy se používá vážený průměr, kdy je každá hodnota ve vstupním setu spojena s váhou, která odráží důležitost, nebo pravděpodobnost výskytu hodnoty. Vážený průměr (3) se počítá jako suma hodnot násobených jejich vahami dělená součtem vah.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{1}$$

$$M = \frac{\max x + \min x}{2} \tag{2}$$

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} \tag{3}$$

Průměr ovšem podléhá zkreslení, pokud je počítán na setu obsahujícím extrémy, proto se někdy používá ořezaný průměr, kdy se dataset omezí tak, že z každé strany seřazené

posloupnosti jsou odebrány hodnoty (běžně se používá 2 – 5%). Pro asymetrická data je lepší mírou pro určení centrální hodnoty medián. [2, 3]

Modus je dalším způsobem, jak změřit centrální tendenci datasetu. Jedná se o hodnotu, která se v souboru dat vyskytuje nejčastěji. Pokud je taková hodnota jenom jedna, je datový soubor unimodální, dále bimodální, trimodální a multimodální. Opačným extrémem je datový soubor obsahující všechny hodnoty pouze jednou, takový soubor modus nemá. [2, 3]

Posledním zde uvedeným způsobem, jak změřit centrální tendencí je mid-range (střed rozsahu), který je jednoduché spočítat i ve velkých datových souborech , což neplatí například pro medián, kdy je třeba data prvně seřadit.

Pro symetrická unimodální data platí, že průměr, medián a modus mají stejnou hodnotu. V reálném světě jsou data většinou spíše zkreslená a to buď pozitivně (modus má nižší hodnotu, než medián), nebo negativně (modus má vyšší hodnotu, než medián). [2]

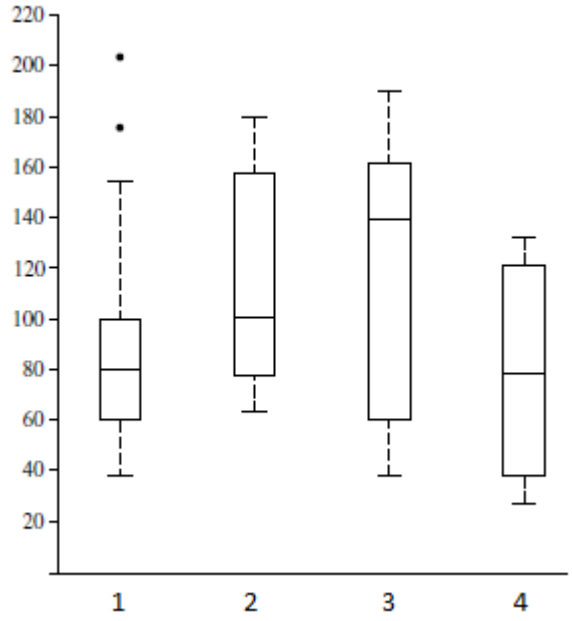
2.2.2 Rozložení dat

Další užitečnou statistikou dat je měření rozložení a rozpětí dat. Pro tyto účely slouží rozsah, kvantily, kvartily, percentily, mezikvartilové rozpětí, variance (rozptyl) a směrodatná odchylka. Krabicový graf pěti-číselného shrnutí je vhodný pro identifikaci extrémů. [2, 3]

Rozsah datového souboru je definován jako rozdíl mezi maximální a minimální hodnotou, **kvantily** jsou body rozložení dat v pravidelných intervalech, které rozdělují datový soubor na stejně velké části. 2-kvantil rozděluje datový soubor na dvě části a je shodný s mediánem, q-kvantil rozděluje datový soubor na q částí, z čehož vyplývá, že q-kvantil má vždy q-1 bodů. 4-kvantil rozděluje datový soubor na čtvrtiny a body jsou označovány jako **kvartily** a 100-kvantil body se označují jako **percentily**. **Medián, kvartily a percentily** jsou nejčastěji používánými **kvantily**. [2, 3]

Kvartily pomáhají k určení středu rozložení, rozpětí a tvaru. První kvartil je 25. percentil a odděluje spodních 25% dat, třetí kvartil je 75. percentil a odděluje horních 25% dat. Druhý kvartil rozděluje datový soubor na poloviny, je shodný s mediánem a označuje střed rozložení. Vzdálenost mezi prvním a třetím kvartilem je míra rozpětí, které určuje šířku intervalu střední poloviny dat. Šířka tohoto intervalu se nazývá **mezikvartilové rozpětí**. [3]

Žádná numerická míra rozpětí není použitelná pro jeho určení u dat se zkresleným rozložením, právě proto se s udáním mediánu udávají nejčastěji i hodnoty prvního a třetího kvartilu. Pro určení extrémů se běžně používá pravidlo, které říká, že hodnoty spadající o jedna a půl násobek mezikvartilového rozpětí pod první kvartil nebo nad třetí kvartil, jsou extrémy. Medián a první a třetí kvartil ale neudávají informace o minimu a maximu datového souboru a proto se používá tzv. **pěti-bodové shrnutí**, které se zapisuje ve tvaru: Minimum, první kvartil, medián, druhý kvartil, maximum. Pro zobrazení pěti-bodového shrnutí se používají krabicové grafy (Obr. 7). [2, 3]



Obr. 7. Krabicový graf. [2]

V krabicovém grafu (Obr. 7.) jsou vyobrazeny statistiky čtyř datových souborů označených čísly 1, 2, 3 a 4. Popis výstupů statistiky z prvního datového souboru je následující: Medián tohoto datového souboru je 80 a je označen vodorovnou čarou uvnitř obdélníku. Obdélník symbolizuje mezikvartilové rozpětí, první kvartil má hodnotu 60 a třetí kvartil hodnotu 100. Minimum datového souboru je označeno vodorovnou čarou a spojeno se spodní hranou obdélníku přerušovanou čarou. Jeho hodnota je 38. Podobně je to u maxima, které má hodnotu 155. Dva černé body nad maximem s hodnotami 175 a 202 označují dva extrémy, které byly z datového souboru vyňaty, protože překračovaly jedna a půl násobek mezikvartilového ropětí. U následujících tří datových souborů je význam podobný.

Rozptyl a **směrodatná odchylka** jsou míry definující rozložení dat, které popisují, jaké je rozpětí rozložení datového souboru. Nízká směrodatná odchylka značí, že jsou data velmi blízko průměrné hodnotě, kdežto vysoká značí daleko širší rozložení. Rozptyl je sumou kvadrátu vzdáleností hodnot od průměru dělenou počtem hodnot (4) a značí se σ^2, směrodatná odchylka je odmocninou rozptylu a značí se σ. [2]

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{4}$$

Směrodatná odchylka by se měla pro určení rozpětí statistického souboru udávat pouze, pokud je pro střední hodnotu dat použit průměr. Nulová směrodatná odchylka udává, že všechny hodnoty datového souboru jsou stejné. [2]

2.2.3 Grafy a možnosti zobrazení statistického popisu dat

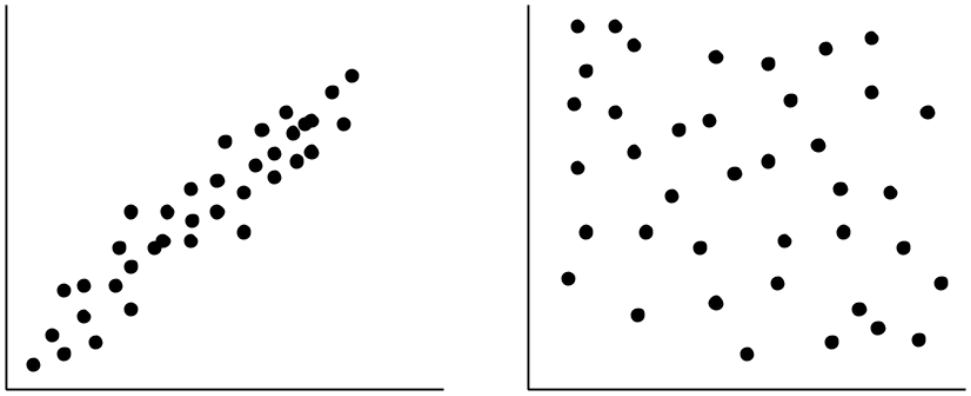
Pro grafické zobrazení statistického popisu dat se používají následující druhy grafů:

1. **Kvantilový graf** – zobrazuje univariantní rozložení dat. Hodnoty jsou seřazeny podle velikosti a následně vyvedeny do grafu, kde na ose x je percentil a na ose y hodnota. V grafu se dále vyznačí 25., 50. a 75. kvantil. Tyto kvantily korespondují s prvním kvartilem, mediánem a třetím kvartilem. V grafu lze vysledovat neobyklé hodnoty a chování. Zobrazením více statistických souborů v jednom grafu lze jednoduše porovnávat jejich medián a kvartily. [2]
2. **Kvantil-kvantil graf** – zobrazuje kvantily jednoho datového souboru vůči jinému. Takový graf umožňuje uživateli porovnat rozložení a hlavně identifikovat posuny jednoho datového souboru vůči jinému. [2] Například může jít o porovnávání prodejních dat ze dvou různých poboček firmy.
3. **Histogram (frekvenční histogramy)** – jedná se o grafickou metodu, jak sumarizovat rozložení atributu. V případě, že je atribut nominální, pak je pro každou hodnotu zobrazen sloupec o výšce zachycující frekvenci výskytu (počet) hodnoty v datovém souboru. Výsledný graf se nazývá **sloupcový**. [2]

   V případě, že se jedná o atribut numerický, pak je graf nazván **histogramem**. Interval možných hodnot atributu je rozdělen na jednotlivé subintervaly, které jsou nejčastěji stejně velké. Pro každý takový subinterval je zobrazen sloupec o výšce značící počet hodnot v datovém souboru spadajících do tohoto intervalu. [2]

4. **Korelační diagram** – slouží k určení, zda mezi dvěma numerickými atributy existuje nějaký vztah, vzor nebo trend. Tvoří se tak, že jsou jednotlivé páry hodnot dvou atributů vykreslovány do dvourozměrného grafu jako body. Korelační diagramy jsou vhodné i k určování clusterů a extrémů. [2]

   Mezi atributy existuje korelace, pokud je možné na základě korelačního diagramu říct, že hodnota jednoho atributu ovlivňuje hodnotu atributu druhého. Korelace může být pozitivní, negativní, nebo žádná. Pokud je korelace pozitivní, s rostoucí hodnotou prvního atributu roste i hodnota atributu druhého. Obrázek (Obr. 8) zobrazuje v levé části pozitivní korelaci atributů a v pravé části atributy bez korelace.



Obr. 8. Korelační diagramy.

2.3 Vizualizace dat

Vizualizace dat je zaměřena na efektivní zachycení dat pomocí jejich grafické reprezentace. Vizualizační techniky lze v datamining procesu použít k nalezení vztahů mezi daty, které nejsou patrné z datového souboru. [2]

2.3.1 Pixelové vizualizační techniky

Barva pixelu u těchto technik určuje hodnotu dimenze. Pro dataset o *n* dimenzích vznikne *n* oken, každé pro jednu dimenzi. Jednotlivá okna obsahují tolik pixelů, kolik je záznamů a každý záznam má pevnou pozici v jednotlivých oknech. Pro určení pozic záznamů je vybrán jeden atribut (dimenze), podle kterého se záznamy seřadí. Další možností je řadit záznamy podle toho, jak splňují požadavky definované dotazem.

Naplnění oken daty lineárně u širokých oken vede k tomu, že blízká data mohou být od sebe ve vizualizaci velmi daleko, proto se někdy používá plnění oken pomocí vyplňujících křivek – Hilbertova křivka, Grayův kód, Z-křivka.

Okna nemusí být obdélníková, ale mohou být například kruhová, čehož se využívá pro usnadnění porovnávání dimenzí. [2]

2.3.2 Geometrické vizualizační techniky

Nevýhodou pixelových vizualizačních technik je to, že z nich nelze vyčíst rozložení dat v multidimenzionálním prostoru. Geometrické vizualizační techniky pomáhají uživatelům nalézt zajímavé projekce multidimenzionálních datasetů. Základní myšlenkou je zobrazení multidimenzionálního prostoru ve dvou dimenzích. Bodový graf zobrazuje body pomocí kartézských souřadnic ve dvou dimenzích a třetí dimenze může být přidána pomocí obarvení bodů, nebo jejich různým značením. Rozšířením je použití kartézských souřadnic ve 3D a čtvrtá dimenze je opět přidána obarvením grafu. Pro více dimenzí už jsou bodové grafy nepoužitelné. Používají se matice bodových grafů. Pro *n*-dimenzionální data je zkonstruována matice o rozměrech *n* x *n*, která obsahuje 2D bodové grafy, které zobrazují vztahy mezi jednotlivými dimenzemi. Čím více dimenzí, tím je matice rozměrnější a tato technika méně použitelná. Další možnou technikou je technika paralelních souřadnic, kdy je každá dimenze vykreslena jako osa paralelní s ostatními osami a datový záznam je zobrazen jako polygonální čára protínající každou osu v odpovídajícím bodě. Tato technika je ovšem použitelná pouze pro omezené množství datových záznamů, při větším množství se stává graf nepřehledným. [2, 8]

2.3.3 Ikonové vizualizační techniky

Ikonové vizualizační techniky využívají malých ikon k reprezentaci multidimenzionálních dat. V roce 1973 statistik Herman Chernoff vymyslel způsob, jak pomocí obličejů zobrazit data o osmnácti atributech (dimenzích). Jednotlivé části obličejů odpovídají konkrétním dimenzím a tak například tvar nosu nebo šířka úst zobrazují jednotlivé hodnoty. Chernoffovy obličeje využívají schopnost lidské mysli rozeznat malé rozdíly ve výrazu tváře a schopnost vnímat více charakteristik zároveň. Asymetrický Chernoffův model využívá obě poloviny obličeje zvlášť a umožňuje tak zobrazit až 36 dimenzí. [2]

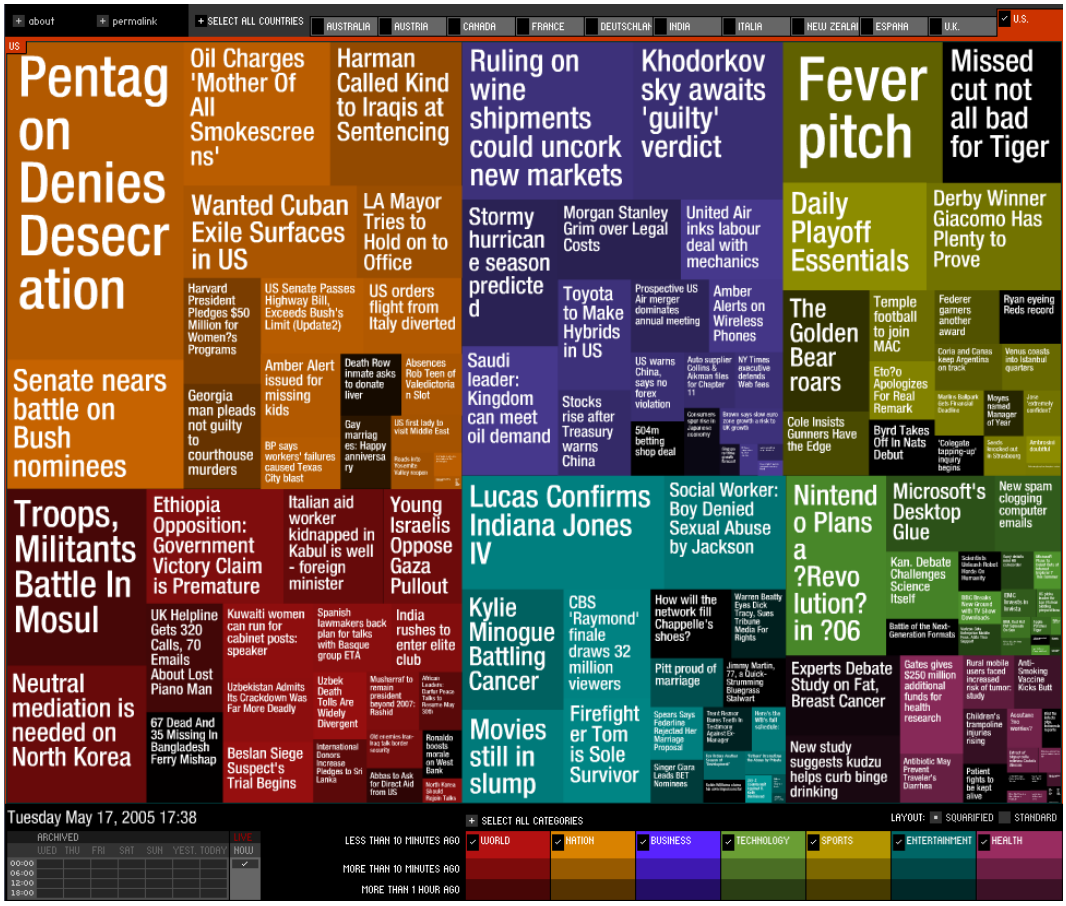
Další možností je použití hůlkových panáčků pro zobrazení více dimenzí. Jednotliví panáčci jsou vykresleni ve 2D prostoru, který pokryje dva atributy a zbylé atributy jsou

zakódovány do těla panáčků skládajícího se z pěti částí, konkrétně do úhlů a délek končetin. Tato vizualizační technika je vhodná především pro data silně závislá na dvou atributech, které jsou voleny jako osy. [2, 8]

2.3.4 Hierarchické vizualizační techniky

Všechny předchozí vizualizační techniky se snažily zobrazit více dimenzí zároveň, kdežto hierarchické vizualizační techniky rozdělují dimenze do subprostorů, které jsou zobrazeny v hierarchii. Příkladem takové techniky je *n*-Vision (světy ve světech), kde se zafixuje určitá podmnožina množiny dimenzí a zbylé dimenze so zobrazí jako vnitřní 3D graf s počátkem v zafixovaných hodnotách vnějšího grafu. Pokud je dimenzí více, než šest, dochází ke stále hlubšímu zanoření a tím vznikají "světy ve světech." Dalším příkladem je technika stromových map, která zobrazuje data jako zanořené obdélníky. [2]

Na obrázku (Obr. 9) je zobrazena stromová mapa pro Google články. Články jsou rozděleny do sedmi kategorií (rozdělení podle barev) a v každé kategorii jsou články rozděleny do dalších subkategorií.



Obr. 9. Stromová mapa – články na Google. [9]

2.4 Podobnost dat

Mnoho datamining aplikací (analýza clusterů, analýza extrémů a klasifikace nejbližších sousedů) využívá míru podobnosti objektů. U analýzy clusterů se pomocí podobnosti určuje, zda prvek do clusteru spadá, či nikoliv; u analýzy extrémů jsou za potencionální extrémy považovány objekty, jež jsou velice nepodobné ostatním a u klasifikace nejbližších sousedů se využívá podobnosti k určení třídy objektu. Míry podobnosti a nepodobnosti jsou v oblasti dataminingu velmi významné. [2]

Míra podobnosti se nejčastěji udává jako hodnota z intevalu <0, 1>, kde 1 znamená, že jsou objekty shodné a 0, že shodné nejsou. Míra nepodobnosti je doplňkem míry podobnosti.

2.4.1 Matice dat a matice nepodobnosti

Matice dat a matice nepodobnosti jsou dva objekty, se kterými nejčastěji pracují algoritmy analýzy clusterů a algoritmy klasifikace nejbližších sousedů.

Matice dat je struktura obsahující *n* datových objektů, má rozměry *n* x *p*, kde *p* je počet atributů (5). [2]

Matice nepodobnosti je struktura, která má rozměry *n* x *n*, neboť se jedná o matici obsahující hodnoty nepodobnosti mezi jednotlivými *n* záznamy. Z toho vyplývá, že je symetrická a na hlavní diagonále jsou nuly (6). Funkce nepodobnosti se označuje *d(i, j)* a vyjadřuje nepodobnost mezi *i*-tým a *j*-tým datovým objektem. [2]

$$\begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} 0 & \cdots & d(1,n) \\ \vdots & \ddots & \vdots \\ d(n,1) & \cdots & 0 \end{bmatrix} \tag{6}$$

2.4.2 Měření vzdálenosti dat

Pro různé druhy atributů se používají různé druhy měření vzdálenosti (odlišnosti). Pro nominální atributy se nejčastěji používá poměr atributů odlišných hodnot vůči všem atributům; pro symetrické binární atributy se používá stejného principu, jako u nominálních atributů, ale pro nesymetrické se používá Jaccardův koeficient; u numerických atributů se používá Eukleidovská, Manhattanská, Minkowského a

Chebyshevova metrika; pro aplikace pracující s řídkými numerickými vektory se používá kosinová míra podobnosti a Tanimotův koeficient. [2]

3 PREPROCESSING

Data v databázích jsou často nekompletní, nekozistentní a zašuměná, což vede k nepřesnostem datamining procesů a výsledků. Právě proto, je potřeba data před zahájením dataminingu upravit do vhodné podoby (preprocessing). Základními částmi preprocessingu jsou: Čištění dat, jejich integrace, redukce a transformace. [2, 3, 8]

Kvalita dat pro zpracování je určena následujícími faktory: **Přesnost**, **úplnost**, **konzistence**, **včasnost**, **uvěřitelnost** a **interpretovatelnost**. Nepřesná, neúplná a nekonzistentní data jsou nejběžnějším obsahem skutečných databází a datových skladišť. Nepřesná data jsou data, jejichž atributy nemají správné hodnoty; neúplná data postrádají některé podstatné atributy; nekonzistentní data mají například hodnoty jednoho atributu v různém formátu. Včasnost dat je pro kvalitu dat důležitá především u procesů pracujících v daných časových intervalech, pokud jsou data v okamžiku zpracování nedostupná, dochází k nepřesnostem. Uvěřitelnost dat reprezentuje, jakou důvěru mají v data uživatelé, interpretovatelnost zase schopnost uživatelů porozumět datům. [3]

3.1 Čištění dat

Čištění dat probíhá pomocí doplňování chybějících hodnot, vyhlazování zašuměných dat, identifikace nebo vyřazování extrémů a řešení nekonzistencí. Nevyčištěná data představují problém pro proces dataminingu, který může navrátit nespolehlivé výsledky. [2, 3]

3.1.1 Chybějící hodnoty

Existuje několik způsobů, jak naložit s chybějícími hodnotami a jejich použití je voleno podle typu chybějících hodnot a úlohy dataminingu.

1. **Vyřazení záznamu** – obvyklé, pokud chybí hodnota atributu učujícího třídu. Metoda je efektivní, pokud záznam obsahuje více chybějících hodnot atributů, v opačném případě jsou vyřazena data, která by mohla být jinak dobře použitelná. [3, 8]
2. **Manuální doplnění hodnot** – časově náročný úkon, který není vhodný pro data obsahující mnoho chybějících hodnot. Doplňováním vzniká šum. [3, 8]
3. **Nahrazení chybějící hodnoty konstantou** – chybějící hodnoty jsou doplněny jednou globální hodnotou. Pokud je ovšem procento chybějících hodnot velké, jsou nalezeny falešné zajímavé koncepty a clustery. [3, 8]

4. **Nahrazení chybějící hodnoty průměrem/mediánem** – podobně jako v předchozím případě mohou být naelezeny falešné zajímavé koncepty a clustery. Průměr se používá pro data s normálním rozložením, medián pro data zkreslená. [3, 8]
5. **Nahrazení chybějící hodnoty průměrem/mediánem třídy** – pouze u klasifikačních problémů. [3, 8]
6. **Nahrazení nejvíce pravděpodobnou hodnotou** – hodnota se vygeneruje pomocí prediktivního modelu, který je třeba vytvořit z úplných dat. Dále lze použít regresi, Bayesův formalismus, cluster analýzu nebo indukci rozhodovacím stromem. Tato metoda je nejoblíbenější, protože využívá k doplnění nejvíce dat z datového souboru. [3, 8]

Obecně je nejlepší využít více metod pro doplnění chybějících hodnot a analyzovat výsledná řešení. [8]

3.1.2 Šum a extrémy

Šum je náhodná chyba nebo odchylka v měřené veličině, kterou je možné odstranit následujícími způsoby.

1. **Binning (kontejnerování)** – metoda vyhlazuje data na základě hodnot v okolí. Setříděné hodnoty jsou rozděleny do binů (kontejnerů). Protože vyhlazování probíhá pomocí okolních hodnot, jedná se o lokální vyhlazování. V případě vyhlazování průměrem kontejneru se hodnoty originálních dat nahradí průměrnou hodnotou kontejneru, podobně při vyhlazování mediánem kontejneru, kde se využívá mediánu hodnot. Kontejnerování je taktéž způsob diskretizace hodnot. [3]
2. **Regrese** – regresní technika přizpůsobuje hodnoty dat funkci. [3]
3. **Analýza extrémů** – extrémy mohou být detekovány pomocí analýzy clusterů (Obr. 5). [3]

3.2 Integrace dat

Při integraci dat z více zdrojů může dojít k problémům, které je potřeba řešit. V různých zdrojích může být pro stejný atribut použito jiné jméno, pro hodnoty jiné kódování, některé atributy jsou odvozené od jiných, což vede ke zpomalování procesu nalézání znalostí v datech. Kvůli zmíněným problémům je potřeba při integraci dat podniknout kroky na jejich odstranění, což je také součástí preprocessingu dat. Po úspěšné integraci dat může

být zařazen další krok čištění dat, který odstraní redundantní data vzniklé právě integrací z více zdrojů. [2, 3]

3.2.1 Problém identifikace entit, detekce konfliktních hodnot

Identifikace entit je proces, který určuje shodné entity ve více zdrojích dat. Základem je porovnávání metadat (název, význam, datový typ, omezení rozsahu hodnot, pravidla pro null hodnoty), kterým se dá vyhnout chybám v integraci schémat. Metadata mohou být použita při transformaci dat, takže jsou využívána i při čištění dat. Při porovnávání atributů dat z různých databází během procesu integrace je potřeba brát zřetel na strukturu dat. Především na funkční vztahy mezi atributy a jejich zachování i v cílové databázi. [3]

Konfliktní hodnoty jsou takové, které popisují stejnou entitu, ale mají různou hodnotu. To může vzniknout, pokud zdrojové databáze používají různé typy vyjádření hodnot (např. metrický vs. imperiální systém, různé měny, nebo různé známkování). Další možností je odlišná úroveň abstrakce atributu. Tyto problémy je třeba vyřešit pomocí transformačních pravidel. [3]

3.2.2 Redundance a korelační analýza, duplicitní záznamy

Redundance při integraci vzniká, pokud je možno některé atributy odvodit z jiných (např. měsíční plat z ročního platu), nebo pokud jsou nekonzistentní názvy atributů ve zdrojových datasetech. Redundance dat může být detekována pomocí korelační analýzy. Korelační analýza určuje míru svázanosti dvou atributů. Pro nominální data se používá test pomocí χ^2 (chí-kvadrát), pro numerické hodnoty korelační koeficient a kovariance. [3]

Detekce duplicitních záznamů je dalším důležitým bodem při integraci dat. Pokud se v databázích používají denormalizované tabulky, vzniká redundance záznamů. Nedůslednou aktualizací dat (neaktualizují se všechny duplicitní záznamy) v tabulkách vznikají nekonzistentní záznamy. [3]

3.3 Redukce dat

Rychlost datamining algoritmu je závislá na velikosti množiny dat, nad kterou probíhá, a právě proto je jedním z kroků preprocessingu dat jejich redukce. Redukcí dat je získána redukovaná reprezentace dat, která poskytuje podobné analytické výsledky. Redukce se skládá ze dvou částí. Z redukce dimenzí, které je dosaženo pomocí kompresních technik, výběru podmnožiny atributů (vynechání nepodstatných atributů) a konstrukcí atributů (z

původní množiny atributů je odvozena menší množina použitelných atributů). Druhou částí je redukce původního datasetu, který je nahrazen alternativní reprezentací vzniklou použitím parametrických modelů (např. regrese), nebo neparametrických modelů (histogramy, clustery, vzorkování, agregace dat). Redukce dat je smysluplná pouze, pokud výpočetní doba nepřekročí výpočetní dobu datamining algoritmu. [2]

3.3.1 Vlnková transformace

Diskrétní vlnková transformace je metoda zpracování signálů, která daný vektor X transformuje na vektor X', který je stejně velký, ale numericky odlišný. Vektor X' je složen z vlnkových koeficientů. Při aplikaci této techniky na data, je každý datový záznam považován za vektor. Redukce touto metodou je zajištěna tak, že transformovaná data jsou zkrácena - ukládá se pouze komprimovaná aproximace dat (nejsilnější z vlnkových koeficientů). Výsledný popis dat může být velmi řídký, což zrychluje činnost algoritmů pracujících s řídkými daty. Získání původních dat se provádí pomocí inverzní diskrétní vlnkové transformace. Používané vlnkové transformace jsou: Haar-2, Daubechies-4 a Daubechies-6. Vlnková transformace poskytuje dobré výsledky u řídkých a zkreslených dat, stejně tak u dat se setříděnými atributy a její použití je například u komprese otisků prstů. [3]

3.3.2 Analýza hlavních komponent

Nejpopulárnější metodou pro redukci dimenzí dat je analýza hlavních komponent (Karhunen-Loeve transformace). Původní data ve vektorovém tvaru jsou transformována na nové vektory s redukovaným počtem dimenzí. Cílem této techniky je koncentrovat informaci o rozdílech mezi jednotlivými záznamy do malého počtu dimenzí. Transformace vektorů probíhá tak, že se původní data nejdříve normalizují, poté jsou vypočítány bázové vektory (hlavní komponenty). Hlavní komponenty jsou seřazeny sestupně podle síly (nejrozdílnější atributy mají největší sílu). Protože jsou komponenty seřazeny sestupně, ty nejméně důležité je možno eliminovat. [3, 8]

V porovnání s vlnkovou transformací je analýza hlavních komponent výkonější na řídkých datech a slabší na datech s velkým počtem dimenzí. [2]

3.3.3 Výběr podmnožiny atributů

Výkonnost datamining algoritmu klesá, pokud dataset obsahuje irelevantní nebo redundantní atributy. Stejně tak je problém, pokud chybí atributy relevantní. Manuální

selekce je časově náročná a vyžaduje vysokou úroveň znalostí. Proces výběru podmnožiny atributů redukuje velikost datasetu odstraňováním irelevantních a redundantních atributů. Cílem je nalézt minimální množinu atributů, pro kterou platí stejné pravděpodobnostní rozdělení, jako pro originální množinu atributů. Výhodou při práci s redukovanou množinou atributů je i to, že výsledné vzory obsahují méně atributů a jsou tedy srozumitelnější. [3]

Pro n atributů existuje 2^n možných podmnožin, proto se používají heuristické metody pro nalezení vhodné podmnožiny. Nejčastěji se využívají hladové algoritmy, jejichž základem je použití lokálních extrémů k nalezení globálně optimálního řešení. Nejlepší a nejhorší atributy jsou typicky určeny pomocí statistické významnosti. Příkladem používaných heuristických metod jsou. Dopředná selekce, zpětná eliminace, kombinace dopředné selekce a zpětné eliminace, indukce rozhodovacího stromu. [3]

V některých případech lze využít konstrukce atributů pro zvýšení přesnosti a pochopitelnosti struktury multidimenzionálních dat. Kombinací atributů lze nalézt vztahy, které jsou důležité při získávání informací z dat. [2]

3.3.4 Parametrická redukce dat

Parametrická redukce dat využívá metod regrese a regresních modelů. V případě lineární regrese jsou data proložena přímkou a v multidimenzionálním prostoru je použito vícenásobné lineární regrese. Log-lineární regresní modely aproximují diskrétní multidimenzionální pravděpodobnostní rozdělení. Každý záznam v n-dimenzionálním prostoru je považován za bod. Log-lineární model lze použít pro odhad pravděpodobnosti výskytu každého bodu v prostoru na základě menší podmnožiny dimenzionálních kombinací. [3]

Regresní a log-lineární modely lze použít na řídká data, ale jejich aplikace je limitována. U zkreslených dat je velmi vhodné použít regresní model, kdežto jeho použití u multidimenzionálních dat o velkém počtu dimenzí je výpočetně náročné. [2]

3.3.5 Histogramy

Histogramy používají binning pro aproximaci rozdělení dat a jsou populární formou redukce. Pro tvorbu binů (kontejnerů) se používá dvou metod:

1. **Metoda stejné šířky** – všechny kontejnery mají stejný rozsah.

2. **Metoda stejné frekvence** – kontejnery jsou konstruovány tak, aby každý obsahoval přibližně stejné množství prvků. [3]

Histogramy jsou efektivní pro aproximaci řídkých i hustých dat, stejně tak pro uniformní i zkreslená data. Multidimenzionální histogramy mohou zachytit vztahy mezi atributy a jsou efektivní až pro pět atributů. [2]

3.3.6 Cluster analýza

Analýza clusterů považuje datové záznamy za objekty a rozděluje je do skupin (clusterů). Kvalita clusteru může být reprezentována jeho průměrem (maximální vzdálenost mezi dvěma objekty clusteru). Jiný způsob je centroidní vzdálenost, která udává průměr vzdálenosti všech objektů od těžiště (centroidu) clusteru. [3]

Při redukci dat je reprezentace clusteru použita pro nahrazení originálních dat a efeketivnost této metody je silně závislá na povaze dat. Efektivita je vysoká pro data, která jsou organizována ve vzdálených clusterech, nízká pro promíchaná data. [2]

3.3.7 Vzorkování

U vzorkování je rozsáhlý dataset reprezentován daleko menší množinou (vzorkem). Za předpokaldu, že je dataset označen *D*, počet záznamů v něm *N* a velikost vzorku *s*, lze nejběžnější vzorkovací techniky popsat následovně:

1. **Náhodný vzorek bez nahrazování** – náhodný výběr *s* záznamů z *D*. Pravděpodobnost, že bude vytažen vzorek je *1/N*.
2. **Náhodný vzorek s nahrazováním** – náhodný výběr *s* záznamů z *D*, ale záznamy mohou být vybrány vícekrát.
3. **Vzorek clusterů** – *D* je rozdělen do clusterů a je vybráno *s* clusterů.
4. **Vrstevnatý vzorek** – *D* je rozdělen na vzájemně nesouvislé vrstvy. Vzorek je vytvořen výběrem záznamů z každé vrstvy. [3]

Při redukci dat je vzorkování nejčastěji použito k odhadu výsledku agregovaného dotazu. Pomocí centrální limitní věty je možné určit, jak velký vzorek je dostatečný pro odhad dané funkce s danou maximální úrovní chyby. Velikost takového vzorku může, v porovnání s velikostí datasetu, velmi malá. [2]

3.4 Transformace dat

Některé datamining algoritmy, pro správný běh, vyžadují data v určitém formátu. Převedení dat do potřebné podoby je označováno jako transformace. Příkladem transformace dat může být normalizace, diskretizace, nebo generování hierarchie konceptů. [2, 3]

3.4.1 Normalizace

Normalizace dat je proces, který upravuje atributy tak, aby měly stejnou váhu. Především je vhodná pro klasifikační úlohy používající neuronové sítě, nebo měření vzdálenosti u klasifikace nejbližších sousedů a analýzy clusterů. Existuje několik druhů normalizace:

1. **Min-max normalizace** – provádí lineární transformaci původních dat. Převádí hodnoty atributu *A* z rozsahu <smin, smax> do nového rozsahu <nmin, nmax>. A nová hodnota *nh* je vypočítána ze staré hodnoty *sh* vztahem (7). [3]
2. **Z-skóre normalizace –** hodnoty atributu *A* jsou normalizovány na základě průměru $\bar{A}$ a směrodatné odchylky σ_A (8). Tato metoda je vhodná, pokud není známé minimum a maximum atributu. [3]
3. **Normalizace desítkovým měřítkem –** normalizace probíhá posunem desetinné čárky atributu *A*. Velikost posunu závisí na maximální absolutní hodnotě *A* (9). Kde *j* je nejmenší celé číslo splňující $\max(|nh_i|) < 1$. [3]

$$nh_i = \frac{sh_i - smin}{smax - smin}(nmax - nmin) + nmin \tag{7}$$

$$nh_i = \frac{sh_i - \bar{A}}{\sigma_A} \tag{8}$$

$$nh_i = \frac{sh_i}{10^j} \tag{9}$$

3.4.2 Diskretizace

Diskretizace hodnot může být uskutečněna binningem, kdy jsou numerická data rozdělena do kontejnerů. Binning může být prováděn rekurzivně, čímž se generuje hierarchie konceptů. Binning nepoužívá informace o třídách, takže se jedná o diskretizaci bez učitele, je citlivý na uživatelem specifikovaný počet kontejnerů a na výskyt extrémů. [3]

Diskretizace pomocí histogramů je taktéž diskretizací bez učitele a nevyužívá informace o třídách. Pro definici histogramů je možné použít různá rozdělující pravidla (stejná šířka, stejná frekvence). Algoritmus může být spouštěn rekurzivně na každé části, čímž se generuje multiúrovňová hierarchie konceptů, dokud není dosaženo požadovaného počtu konceptů, což specifikuje minimální šířku části, nebo minimální počet hodnot v jedné části. [3]

Analýza clusterů lze taktéž využít pro diskretizaci hodnot a to tak, že se numerický atribut *A* rozdělí do clusterů. Cluster analýza bere v potaz rodělení hodnot atributu a jejich vzdálenosti, takže poskytuje kvalitní diskretizaci. [3]

Techniky generování rozhodovacích stromů lze taktéž využít při diskretizaci. Takový způsob diskretizace již využívá informace o třídách a je proto označován jako diskretizace s učitelem. Pro rozdělení hodnot numerického atributu jsou voleny hodnoty, které mají nejmenší entropii. A následnou rekurzí se dělí vzniklé intervaly hodnot a utváří se hierarchie konceptů. [3]

Pro diskretizace lze použít i míru korelace, konkrétně ChiMerge algoritmus vycházející z chí-kvadrátu. Oproti předchozím technikám se tato technika uplatňuje odspoda-nahoru, kde se rekurzivně hledají sousední intervaly a spojují se do větších intervalů. Na začátku algoritmu je jako interval považována každá samostatná hodnota atributu. [3, 8]

3.4.3 Generování hierarchie konceptů pro nominální data

Nominální atributy mohou nabývat konečného množství hodnot, které nelze seřadit. Manuální definice hierarchie konceptů je časově náročný úkol, naštěstí existují způsoby, jak automaticky definovat hierarchie na úrovni definice schématu databáze. Metody pro generování hierarchie konceptů pro nominální data jsou následující:

1. **Specifikace částečného řazení atributů na úrovni schématu uživatelem nebo expertem** – hierarchie konceptů pro nominální atributy často obsahují skupiny atributů, u kterých je možno jednoduše definovat řazení (ulice < město < okres < kraj < stát). [2]
2. **Specifikace části hierarchie explicitním sdružováním dat** – manuální definice části hierarchie konceptů ({okres Zlín, okres Vsetín, okres Uherské Hradiště, okres Kroměříž} ∈ Zlínský kraj). [2]

3. **Specifikace množiny atributů bez specifikace částečného řazení** – uživatel může specifikovat množinu atributů, které tvoří koncept hierarchie, ale nespecifikuje explicitně řazení. Na základě množství odlišných hodnot vyskytujících se v jednotlivých atributech se vytvoří řazení (největší počet odlišných hodnot = nejnižší úroveň v hierarchii konceptu). [2]
4. **Specifikace části množiny atributů** – uživatel specifikuje pouze část atributů, které mají být v hierarchii. V takovém případě je třeba zakomponovat do databázového schématu sémantiku dat, aby mohly být atributy s těsným sémantickým spojením seskupeny. [2]

II. PRAKTICKÁ ČÁST

4 BENCHMARK DATASETY

Benchmark datasety se používají k měření výkonnosti datamining algoritmů. Hlavní myšlenkou je testování algoritmů na stejných datech, takže výkonnost ve formě přesnosti a doby zpracování je dobře porovnatelná. Repozitářů datasetů je několik a mezi ty nejznámější patří UCI (University of California, Irvine) Machine Learning Repository, ze kterého pocházejí i datasety popsané v následující části – Iris a Wine.

4.1 IRIS

Dataset Iris je nejpoužívanějším datasetem z repozitáře UCI, kde byl uložen již v roce 1988, obsahuje záznamy o 150 květech kosatců tří druhů – Iris Setosa (Obr. 10), Iris Versicolour (Obr. 11), Iris Virginica (Obr. 12). Květiny byly nasbírány a změřeny Edgarem Andersonem a dataset byl popularizován Ronaldem Fisherem. Každý druh je v datasetu obsažen 50 květy, u kterých byla měřena šířka a délka kališních a okvětních lístků. Dataset je kompletní – nechybí žádná data a úkolem datamining algoritmu je klasifikovat jednotlivé květy. Druh Iris Setosa je od dalších dvou lineárně separabilní, kdežto Iris Virginica a Iris Versicolour lineárně separabilní nejsou (Obr. 13). Statistika datasetu je zobrazena v tabulce (Tab. 1). [10]

Obr. 10. Iris Setosa. [11]

Obr. 11. Iris Versicolor. [11]

Obr. 12. Iris Virginica. [11]

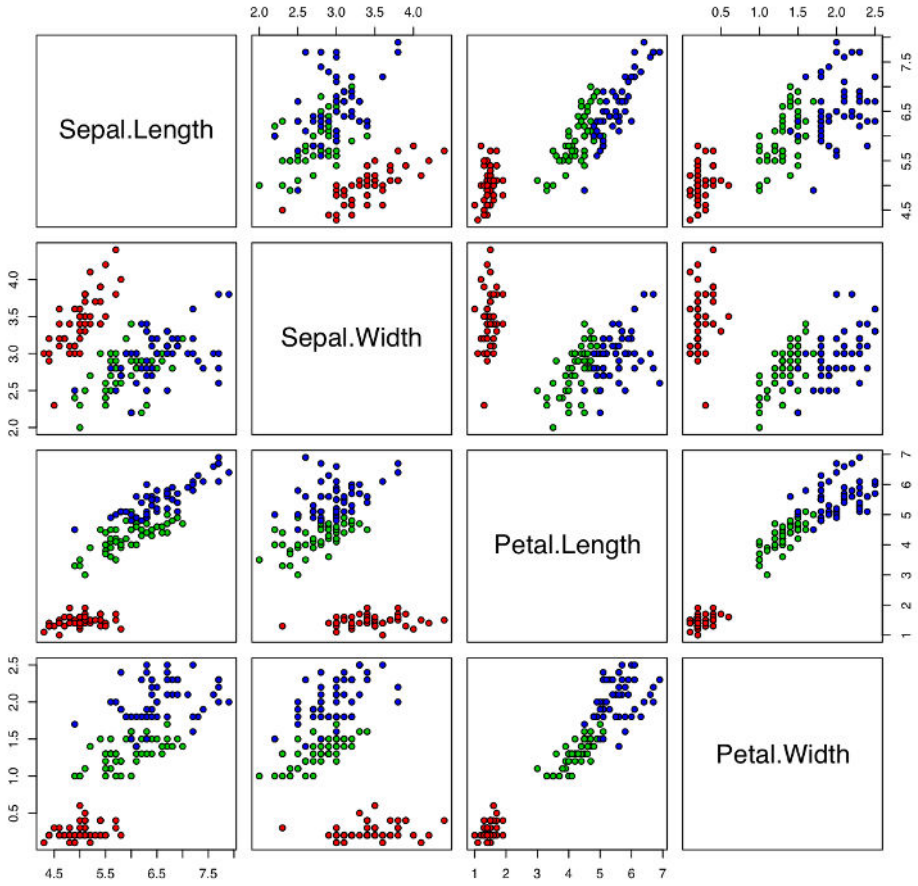
Tab. 1. Statistika Iris datasetu. [10]

Atribut	**Minimum [cm]**	**Maximum [cm]**	**Průměr [cm]**	**Směrodatná odchylka [cm]**	**Koeficient korelace**
Délka kališního lístku	4,30	7,90	5,84	0,83	0,7826
Šířka kališního lístku	2,00	4,40	3,05	0,43	-0,4194
Délka okvětního lístku	1,00	6,90	3,76	1,76	0,9490
Šířka okvětního lístku	0,10	2,50	1,20	0,76	0,9565

Protože je dataset čtyřdimenzionální, tak je pro jeho zobrazení použita matice bodových grafů o rozměrech 4 x 4. V jednotlivých 2D grafech je zobrazena závislost atributu v řádku na atributu ve sloupci. Jednotlivé druhy kosatců jsou rozlišeny barvami: Setosa – červená, Versicolour – zelená, Virginica – modrá.

Význam popisů v obrázku je následující:

Sepal.Length – délka kališních lístků, **Sepal.Width** – šířka kališních lístků, **Petal.Length** – délka okvětních lístků, **Petal.Width** – šířka okvětních lístků.



Obr. 13. Vizualizace Iris datasetu. [12]

4.2 WINE

Dataset Wine obsahuje výsledky chemické analýzy tří kultivarů vín ze stejné oblasti Itálie. Dataset je opět z databáze UCI a byl zde vložen v roce 1991. Jedná se o klasifikační úlohu, dataset má 13 dimenzí, kde všechny jsou reálné a jsou následující: Alkohol, kyselina jablečná, popel, zásaditost popela, hořčík, fenoly, flavanoidy, neflavanoidní fenoly, proanthokyanidiny, barevná intenzita, odstín, OD280/OD315, proline. V datasetu je celkem 178 záznamů a jejich četnost v jednotlivých třídách je 59, 71 a 48. Základní statistika je uvedena v následující tabulce (Tab. 2). [10] Tento dataset je třetím nejpoužívanějším z UCI repozitáře.

Jelikož Iris dataset obsahuje pouze čtyři dimenze, často se vyskytuje v kombinaci s Wine datasetem, který má dimenzí třináct a tak lze odhalit vlastnosti algoritmu v závislosti na dimenzi datasetu. [10]

Tab. 2. Statistika Wine datasetu.

Atribut	**Minimum**	**Maximum**	**Průměr**	**Směrodatná odchylka**	**Koeficient korelace**
Alkohol	11,03	14,83	13,00	0,81	-0,3282
Kyselina jablečná	0,74	5,80	2,34	1,12	0,4378
Popel	1,36	3,23	2.37	0,27	-0,0496
Zásaditost popela	10,60	30,00	19,49	3,34	0,5179
Hořčík	70,00	162,00	99,74	14,28	-0,2092
Fenoly	0,98	3,88	2,30	0,63	-0,7192
Flavanoidy	0,34	5,08	2,03	1,00	-0,8475
Neflanavoidní fenoly	0,13	0,66	0,36	0,12	0,4891
Proanthokyanidiny	0,41	3,58	1,59	0,57	-0,4991
Barevná intenzita	1,28	13,00	5,06	2,32	0,2657
Odstín	0,48	1,71	0,96	0,23	-0,6174
OD280/OD315	1,27	4,00	2,61	0,71	-0,7882
Proline	278,00	1680,00	746,89	314,91	-0,6337

Dataset je z klasifikačního hlediska dobře definován a jednotlivé třídy mají přesnou strukturu, proto je považován za jednodušší a vhodný pro prvotní testování klasifikátoru. [10]

5 ANALÝZA DATAMINING ALGORITMŮ

V této části je představeno několik moderních datamining algoritmů. Tyto algoritmy jsou postupně stručně popsány a jejich funkce je testována na benchmark datasetech Iris a Wine. Na závěr je uvedena analýza výsledků.

5.1 Klasifikace

Algoritmy v této části přistupují k daným problémům, jako ke klasifikačním.

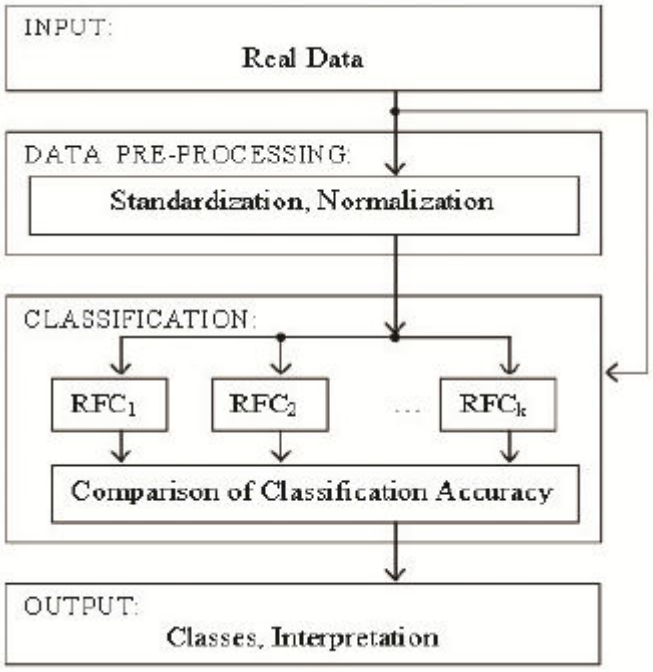
5.1.1 Rough-Fuzzy klasifikátor – RFC-FS

Klasifikátor vyžívající teorii hrubých a fuzzy množin, který na Iris a Wine datasetu testovali Jiří Krupka a Pavel Jirava ve [13] vznikl vygenerováním IF-THEN pravidel v RST (Rough Sets Theory) nástroji nazvaném RSTbox a následném využití fuzzy kontroleru s Mamdaniho odvozením (Mamdaniho fuzzy systém) pro optimalizaci pravidel. Mamdaniho fuzzy systém využívá pro výpočet výstupu ze vstupních hodnot šestici kroků:

1. Určení množiny fuzzy pravidel.
2. Fuzzyfikace (převedení do oblasti fuzzy logiky) vstupních hodnot pomocí funkce příslušnosti.
3. Kombinace fuzzyfikovaných vstupních hodnot podle fuzzy pravidel pro určení jejich síly.
4. Nalezení důsledku pravidla kombinací jeho síly a výstupní funkce příslušnosti.
5. Získání fuzzy výstupu kombinací důsledků.
6. Defuzzyfikace výstupu. [14]

Klasifikační model je na následujícím obrázku (Obr. 14), ze kterého je patrné, že se klasifikační proces skládá ze tří fází – předzpracování dat, klasifikace dat a intepretace výstupu. Význam jednotlivých popisků je následující:

INPUT – vstup, **Real Data** – originální data, **DATA PRE-PROCESSING** – předzpracování dat, **Standardization, Normalization** – standardizace, normalizace, **CLASIFICATION** – klasifikace, **RFC_i** – Rough Fuzzy Classifier (klasifikátor využívající teorii hrubých množin), **Comparison of Classification Accuracy** – porovnání přesnosti klasifikace, **OUTPUT** – výstup, **Classes, Interpretation** – třídy, intepretace.



Obr. 14. Klasifikační model. [13]

Výsledný klasifikátor byl testován dvojím způsobem – test na celém datasetu a 'holdout' metodou. V prvním případě byl pro naučení klasifikátoru použit celý dataset a stejně tak se testovala jeho výkonnost na stejné množině. Výsledky jsou uvedeny v souhrnné tabulce (Tab. 3) pod označením RFC-FS1. V druhém případě bylo pro Iris dataset vybráno 120 a pro Wine dataset 138 vzorků, které vytvořily trénovací množinu a testovací množina čítala 30 (Iris) a 40 (Wine) vzorků. Výsledky jsou v tabulce (Tab. 3) uvedeny pod označením RFC-FS2.

5.1.2 Umělá neuronová síť se symbolickou regresí – PNN-SR

Klasifikace Iris datasetu byla autory Zuzanou Komínkovou Oplatkovou a Romanem Šenkeříkem uvedena ve [15]. Daná metoda řeší syntézu klasifikátoru pomocí AP (Analytic Programming) na bázi podobnosti s umělými neuronovými sítěmi. Evolučně vyšlechtěný klasifikační vztah poté rozdělí příslušná data do definovaných tříd.

AP je metoda symbolické regrese s využitím evolučních výpočetních technik obdobně jako GP (Genetic Programming), které využívá GE (Grammatical Evolution) pro syntézu programů. Oproti GP lze u AP použít jakýkoliv evoluční algoritmus. Základem AP je GFS (General Functional Set), což je množina funkcí, operátorů a terminálů (konstanty nebo nezávislé proměnné). Z prvků této množiny se syntetizuje program mapováním domény jedinců na doménu programů, což je prováděno dvěma operacemi – DSH (Discrete Set

Handling) a bezpečnostními pocedurami. DSH se používá pro mapování jedince populace do GFS. Atributy jedince populace jsou celá čísla a jsou to indexy do GFS. [16]

Jako evoluční algoritmus pro AP byla zvolena DE (Differential Evolution) a to jak pro hlavní proces, tak pro metaevoluční část. Konkrétní nastavení evolučního algoritmu bylo následující:

Hlavní proces

- Velikost populace $NP = 20$.
- Mutační konstanta $F = 0{,}8$.
- Práh křížení $CR = 0{,}8$.
- Počet generací $G = 50$.
- Maximální počet ohodnocení účelové funkce $CFE = 1000$.

Metaevoluce

- Velikost populace $NP = 40$.
- Mutační konstanta $F = 0{,}8$.
- Práh křížení $CR = 0{,}8$.
- Počet generací $G = 150$.
- Maximální počet ohodnocení účelové funkce $CFE = 6000$.

Prvky v GFS

- *GFS2arg* = +, -, /, *, ^, exp
- *GFS0arrg* = x_1, x_2, x_3, x_4, K

Terminály $x_{1\text{-}4}$ jsou jednotlivé atributy v datasetu.

Trénovací množinu tvořilo 75 prvků Iris datasetu, testovací množinu zbylých 75. Konkrétní výsledky jsou zobrazeny v tabulce (Tab. 3) pod označením PNN-SR.

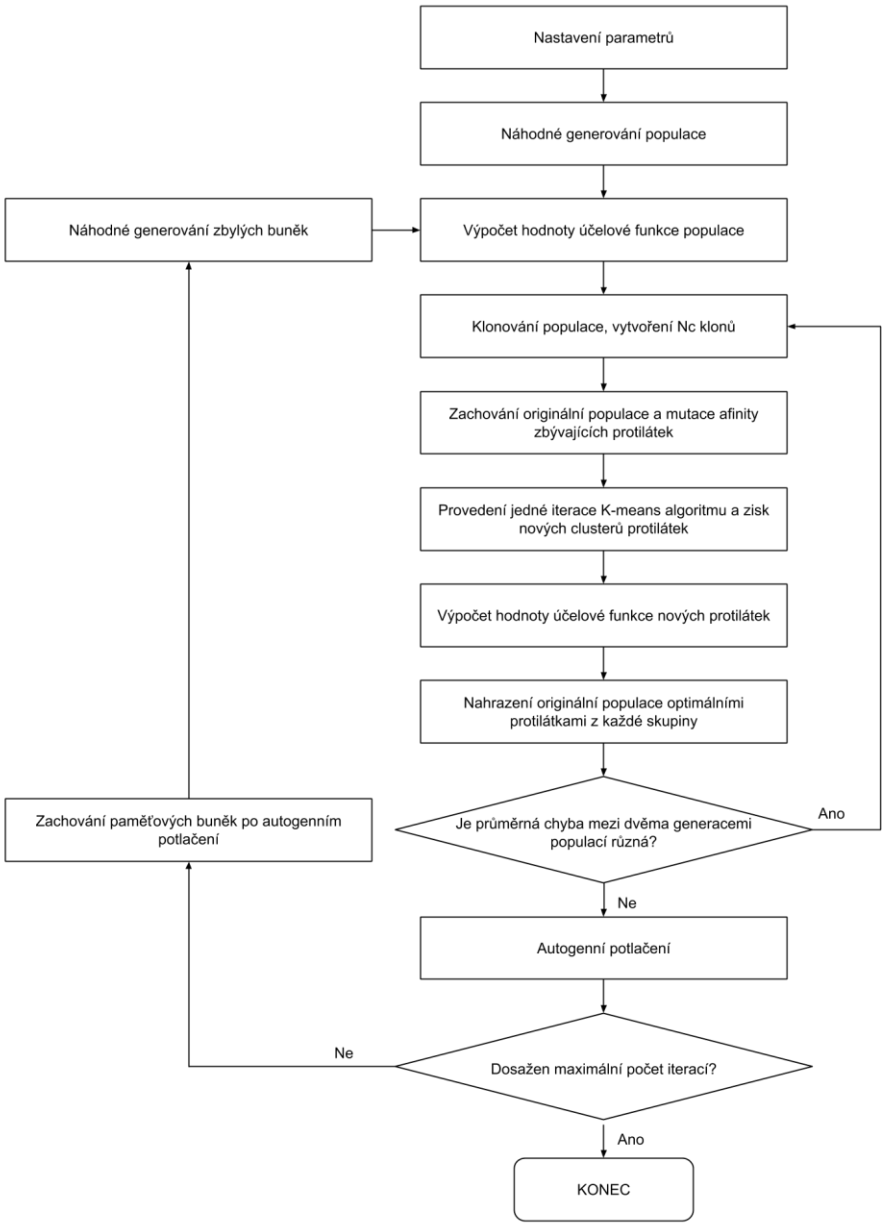
5.2 Clustering

Ačkoliv jsou obě úlohy na benchmark datasetech klasifikační, je možno k nim přistupovat i jako ke clusteringovým a následující algoritmy je takto řeší.

5.2.1 Umělá imunitní síť a K-means – aiNet, aiNetK

Autoři studie [17] R J. Kuo, S. S. Chen, W. C. Cheng a C. Y. Tsai testovali clustering Iris a Wine datasetu pomocí algoritmu umělé imunitní sítě a jeho hybridizace s K-means algoritmem.

Algoritmus umělé imuntiní sítě je inspirován reálnou funkcí imunitního systému, využívá tedy protilátek, paměťových buněk, klonování, autogenního potlačení a mutace afinity. Vývojový diagram algoritmu aiNetK je zobrazen na obrázku (Obr. 15) a konkrétní kroky jsou popsány níže.



Obr. 15. Vývojový diagram aiNetK algoritmu. [17]

Nastavení parametrů

Nejdříve je potřeba nastavit potřebné parametry aiNetK algoritmu – počet generací, počet paměťových buněk M, počet zbývajících buněk R, klonovací multiplikátor N_c, práh chyby mezi dvěma generacemi a práh autogenní suprese σ_S, počet clusterů K. [17]

Náhodné generování populace

První generace je generována náhodně, obsahuje P protilátek s M paměťovými buňkami a R zbývajícími buňkami. Každá protilátka je tvořena K centroidovými vektory (10), kde P_{id} je i-tá protilátka, i je její index a d je index clusteru. [17]

$$P_{id} = \left(X_{i1}, \dots, X_{ij}, \dots, X_{ik}\right) \tag{10}$$

Výpočet hodnoty účelové funkce populace

Hodnota účelové funkce je vypočtena pomocí Eukleidovské metriky. Nejprve je vypočtena vzdálenost každého datového vektoru od centroidu clusteru (11), kde X_i je i-tý datový vektor a C_{ij} je centroid i-té protilátky a j-tého clusteru. Poté je vytořeno K clusterů přiřazením každého datového vektoru nejbližšímu centroidu pro každou protilátku a je vypočtena hodnota účelové funkce Ab_i (12), kde D_i (13) je suma Eukleidovských vzdáleností (*SED*) i-té protilátky mezi X_i a C_{ij}. Počet datových vektorů j-tého clusteru je označen n_{ij}. [17]

$$d\left(X_i, C_{ij}\right) = \left\|X_i - C_{ij}\right\| \tag{11}$$

$$(Ab)_i = \frac{1}{1 + D_i}, 0 \leq (Ab)_i \leq 1, i \in P \tag{12}$$

$$D_i = SED_i = \sum_{j=1}^{K} \sum_{\forall X_i \in n_{ijj}} \left\|X_i - C_{ij}\right\| \tag{13}$$

Klonování populace, vytvoření N_c klonů

Každá protilátka v populaci je klonována N_c krát, vzniká $P \times N_c$ protilátek. [17]

Zachování originální populace a mutace afinity zbývajících protilátek

Nejprve se uloží originální populace a poté jsou mutovány afinity zbývajících protilátek, aby bylo zabráněno zhoršení parametrů variací. Variace probíhá v krocích. V prvním kroku se vypočítá mutační poměr α_i (15), který je závislý na hodnotě účelové funkce. Pro výpočet se používá normalizovaná hodnota afinity $(Ab^*)_i$ (14). Čím vyšší afinita protilátky, tím

nižší mutační poměr. Druhým krokem je mutace afinity podle vztahu (16), kde $c = (c_1, ..., c_x)$ je klonovaná afinita, c' je afinita po mutaci a $N(0, 1)$ je standardizované normální rozložení. [17]

$$(Ab^*)_i = \frac{(Ab)_i}{(Ab)_{max}}, i \in (P * N_c) - P \tag{14}$$

$$\alpha_i = \left(\frac{1}{\rho}\right) e^{-(Ab^*)_i}, \rho \; je \; konstanta \tag{15}$$

$$c' = c + \alpha_i \times N(0,1) \tag{16}$$

Provedení jedné iterace K-means algoritmu a zisk nových clusterů protilátek

Nejprve je vypočítána Eukleidovská vzdálenost každého datového vektoru X každé protilátky pro všechny clustery. V každé protilátce jsou všechny datové vektory X přiřazeny nejbližšímu centroidu clusteru a je vypočten nový centroid na základě vztahu (17), kde $C_{ij\,new}$ je nový centroid j-tého clusteru v i-té protilátce. [17]

$$C_{ij\,new} = \frac{1}{n_{ij}} \sum_{\forall X_i \in n_{ij}} X_i \tag{17}$$

Výpočet hodnoty účelové funkce nových protilátek

Centroidy clusterů se mohou po mutaci afinity a jedné iteraci K-means algoritmu změnit a proto se musí opět vypočítat hodnoty účelových funkcí nových protilátek. [17]

Nahrazení originální populace optimálními protilátkami z každé skupiny

Nyní existuje P skupin nových protilátek clusterů a každý cluster má N_c klonů. V každé skupině jsou seřazeny prrotilátky sestupně podle hodnoty účelové funkce a do nové generace je z každé skupiny vybrána jedna protilátka. Takto vybrané protilátky nahrazují originální populaci P. [17]

Určení, zda je průměrná chyba mezi dvěma populacemi různá

Pro výpočet průměrné chyby populace (populační chyby) je použit vztah (18), kde n značí velikost populace. Pokud je tato hodnota vyšší, než práh chyby mezi dvěma generacemi (práh musí být předem určen), populace nebyla dostatečně prohledána a je třeba v algoritmu pokračovat od kroku **klonování populace**. [17]

Ve studii [17] byl pro určení prahu chyby mezi dvěma generacemi použit Taguchiho návrh parametrů.

$$Populační\ chyba = \sum_{i}^{p} \frac{SED_i}{n} \tag{18}$$

Autogenní potlačení

Autogenní potlačení má za úkol rozšířit prohledávanou oblast pomocí mazání příliš podobných protilátek a tak zamezuje konvergenci k lokálnímu extrému. Populace jsou seřazeny sestupně a poté je vypočítána Eukleidovská vzdálenost mezi párovými protilátkami (19). Pokud je vzdálenost menší, než práh autogenního potlačení σ_S, pak je protilátka s nižší afinitou smazána a druhá protilátka je použita jako paměťová buňka M a je násobena d %, což je počet zbývajících buněk R. Tím se poměr paměťových a zbývajících buněk v každé generaci mění. [17]

$$d(C_i, C_j) = \|C_i - C_j\| \tag{19}$$

Určení, zda byl dosažen maximální počet iterací

Pokud bylo dosaženo nastaveného počtu iterací, pak je algoritmus ukončen, jinak se vrací ke kroku **výpočtu hodnot účelové funkce populace**. [17]

Konkrétní nastavení testovaných algoritmů aiNet a aiNetK bylo ve studii [17] následující:

- Velikost rodičovské buňky $N = 30$.
- Velikost paměťové buňky $M = 20$.
- Konstatní parametr $\rho = 5$.
- Klonovací multiplikátor $N_c = 11$.
- Procento nových buněk $d = 0.9$.
- Práh potlačení *potlačení* = 0.01.
- Beta $\beta = 0.005$.

Konkrétní výsledky na Iris a Wine datasetu jsou uvedeny v tabulce (Tab. 3) pod označením aiNet pro aiNet algoritmus a aiNetK pro aiNetK algoritmus.

5.2.2 Optimalizace hejnem částic a heuristické hledání - PSO, PSOHS

Ve své výzkumné práci [18] se autoři Abdolreza Hatamlou a Masoumeh Hatamlou zabývají hybridizací PSO (Particle Swarm Optimization) – Optimalizace hejnem částic a HS (Heuristic Search) – heuristickým prohledáváním. PSO je zde použita pro nalezení

iniciálního řešení clustering problému a poté je aplikováno HS pro prohledání okolí nalezeného řešení a jeho vylepšení.

Pro výpočet hodnoty účelové funkce autoři použili funkci MSE (Mean-Square quantization Error) (20), kde X je konečná množina n datových vektorů $X = (x_1, \ldots, x_n)$ a S je rozdělení X do k clusterů $S = (S_1, \ldots, S_k)$. Centroidy clusterů jsou označeny C, $C = (c_1, \ldots, c_k)$. Funkce $d(x_i, c_l)$ je funkcí vzdálenosti mezi datovým vektorem x_i a centroidem clusteru $S_l(c_l)$. Metrika je použita Eukleidovská. [18]

$$f(X,S) = \sum_{l=1}^{k} \sum_{x_i \in S_l} d(x_i, c_l)^2 \tag{20}$$

PSO algoritmus je optimalizační algoritmus založený na populaci. Imituje sociální chování zvířat v hejnech. V PSO je hejno tvořeno částicemi, kde každá částice je kandidátním řešením daného optimalizačního problému. Hodnota účelové funkce částice je vypočtena ze souřadnic částice. Na počátku je náhodně vygenerována populace částic, každá částice je ohodnocena a pohybuje se prohledávaným prostorem, přičemž si uchovává informaci o nejlepší pozici (určeno nejlepším hodnocením účelové funkce dané částice - *Pbest*). Částice s nejlepší pozicí v celé populaci má hodnotu *Gbest*. Během iterací částice upravují směr a rychlost svého pohybu podle vztahů (21) a (22). Kde W je setrvačnost, $X_i(t) = (x_{i1}, x_{i2}, \ldots, x_{id})$ a $V_i(t) = (v_{i1}, v_{i2}, \ldots, v_{id})$ jsou pozice a rychlost i-té částice v iteraci t, d je počet dimenzí prohledávaného prostoru. Parametry c_1 a c_2 jsou akcelerační koeficienty, které směrují částice ke *Gbest* a *Pbest*, $rand_1$ a $rand_2$ jsou náhodná reálná čísla z intervalu 0 až 1. [18]

$$V_i(t+1) = W \times V_i(t) + c_1 \times rand_1 \times (Pbest - X_i(t)) + c_2 \times rand_2 \times (Gbest - X_i(t)) \tag{21}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{22}$$

V případě algoritmu PSOHS jsou jedinci/částice tvořeny jednorozměrným polem o délce $d \times k$, každá d-tice hodnot v poli označuje jeden centroid clusteru. Iniciální řešení je získáno pomocí PSO algoritmu, toto řešení je poté upravováno HS algoritmem. Velikost kroku pohybu je na začátku algoritmu určena prahem. Tato hodnota je postupně přidávána ke všem atributům řešení. Nejprve je prahová hodnota přičtena k prvnímu atributu prvního centroidu a je vypočtena nová hodnota účelové funkce. Pokud je tato hodnota lepší, pak je původní centroid nahrazen, pokud je hodnota horší, původní centroid je znovu načten a

změní se směr prohledávání – v příští iteraci se bude prahová hodnota odečítat. Jestliže nedojde ke zlepšení řešení, je prahová hodnota dělena dvěma a pokračuje se opět přičítáním. Toto se provádí pro všechny atributy všech centroidu, dokud není dosaženo ukončující podmínky – počet iterací (ve výzkumné práci [18] byl počet iterací 50). [18]

Zde je uveden pseudokód PSOHS algoritmu:

První krok – PSO algoritmus

```
1 1.1 Generování prvotní populace
2 1.2 Výpočet hodnot účelové funkce populace
3 1.3 Pokud je potřeba, upravení hodnot Gbest a Pbest
4 1.4 Upravení rychlosti a pozice populace podle vztahů (21) a (22)
5 1.5 Opakování kroků 1.2 až 1.4 dokud není dosažena ukončující
  podmínka
6 1.6 Předání výstupu PSO algoritmu herusitického prohledávání
```

Druhý krok – HS algoritmus

```
7 Opakuj
8 Pro všechny centroidy i = 1, ..., k
      a. Pro všechny atributy j = 1, ..., d
           i.  Pokud SDi(j) == 1
                  1. Ci(j) = Ci(j) + SSi(j);
                  2. Výpočet hodnoty účelové funkce nového centroidu
                  3. Pokud je nová hodnota lepší
                        a. ulož centroid
                  4. Jinak
                        a. načti starý centroid
                        b. SDi(j) = -1;
          ii.  Pokud SDi(j) == -1
                  1. Ci(j) = Ci(j) - SSi(j);
                  2. Výpočet hodnoty účelové funkce nového centroidu
                  3. Pokud je nová hodnota lepší
                        a. ulož centroid
                  4. Jinak
                        a. načti starý centroid
                        b. SDi(j) = 0;
         iii.  Pokud SDi(j) == 0
                  1. SSi(j) = SSi(j)/2;
                  2. SDi(j) = 1;
      b. Konec cyklu
9 Konec cyklu
```

```
10 Dokud nejsou dosaženy ukončující podmínky
```

V psuedokódu je použito označení $SD = (SD_1, SD_2, ..., SD_k)$ pro směr prohledávání a $SS = (SS_1, SS_2, ..., SS_k)$ pro velikost kroku pohybu, kde SD_i je směr prohledávání i-tého centroidu a SS_i je velikost kroku pohybu i-tého centroidu. Na začátku algoritmu jsou Všechny SD_i nastaveny na pole prvků 1 a SS_i na pole $Max(dataset)$, kde všechny prvky tohoto pole jsou maximální hodnoty dané dimenze v datasetu. $C = (C_1, C_2, ..., C_k)$ je pole centroidů k clusterů a $C_i(j)$ značí j-tý atribut i-tého centroidu. [18]

Ve výzkumné práci [18] byly výsledky PSO a PSOHS porovnány pro 50 nezávislých běhů a nastavení parametrů je stejné, jako ve studii [19]. Výsledky PSO a PSOHS algoritmů jsou shrnuty v tabulce (Tab. 3) a označeny jako PSO pro PSO algoritmus a PSOHS pro PSOHS algoritmus.

5.3 Analýza výsledků

Výsledky výše zmíněných datamining algoritmů jsou shrnuty v tabulce (Tab. 3). V tabulce je uvedena přesnost v procentech (poměr mezi správně klasifikovanými objekty/objekty zařazenými do správného clusteru a všemi objekty v datasetu).

Tab. 3. Shrnutí výsledků klasifikace a clusteringu na datasetech Iris a Wine.

		Přesnost [%]	
Úloha	**Algoritmus**	Iris	Wine
Klasifikace	RFC-FS1	95,31	96,60
	RFC-FS2	93,33	95,00
	PNN-SR	97,34	-
Clustering	aiNet	86,00	89,33
	aiNetK	60,11	95,51
	PSO	89,94	71,21
	PSOHS	90,00	71,69

Z výsledků v tabulce (Tab. 3) vyplývá, že pro zvolené datasety jsou vhodnější algoritmy řešící úlohu jako klasifikační, které dosahují vyšší přesnosti, což je důsledkem toho, že klasifikace probíhá na základě učení s učitelem, kdežto u clusteringu není použita množina

testovacích dat. Dále je z výsledků patrné, že přesnost clusteringu určitých algoritmů je závislá na dimenzionalitě datasetu, příkladem je algoritmus aiNetK, který dosáhl na Wine datasetu velmi dobrého výsledku 95,51%, kdežto na Iris datasetu pouhých 60,11%. Algoritmy PSO a PSOHS mají výrazně vyšší přesnost (89,94% a 90,00%) na Iris datasetu, než na Wine datasetu (71,21% a 71,69%). Na základě výsledků lze říci, že algoritmy využívající umělou imunitní síť pracují lépe ve vícedimenzionálním prostoru, kdežto algoritmy využívající PSO v prostoru s menším počtem dimenzí.

6 VLASTNÍ IMPLEMENTACE CLUSTERINGU

V rámci této práce byla testována množina moderních clustering algoritmů na Iris a Wine datasetech. Jednotlivé výsledky byly zaznamenány a jsou zobrazeny v tabulce (Tab. 4).

6.1 Algoritmy

Do množiny testovaných algoritmů byly zařazeny následující: RS (Random Search), DE, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), FKM (Fuzzy K-Means), KMPP (K-Means Plus Plus).

Implementace algoritmů RS a DE byla vytvořena v jazyce Java, ve kterém je, díky jeho objektovému charakteru, vývoj evolučních algoritmů velice pohodlný. Implementace algoritmů DBSCAN, FKM a KMPP byla využita z knihovny Apache Commons Math 3.5, která je rovněž v jazyce Java. Pro účely porovnávání výsledků byla vytvořena testovací aplikace ClusteringTest, která pro základní statistické výsledky používá objekt DescriptiveStatistics z knihovny Apache Commons Math 3.5. Pro jednotlivé datasety byly vytvořeny vhodné objekty s názvy IrisDataset.java a WineDataset.java.

6.1.1 Random Search - RS

Algoritmus náhodného prohledávání je do množiny algoritmů zařazen pouze pro porovnání s více sofistikovanými algoritmy. Jedná se o náhodné generování řešení, která jsou dále ohodnocena a je uchováváno jenom nejlepší řešení do vyčerpání maximálního počtu iterací. Řešením je zde vektor obsahující souřadnice jednotlivých centroidů clusterů, celková délka řešení je tedy $K \times n$, kde K je počet clusterů a n je dimenze problému. Pro ohodnocení řešení jsou u obou datasetů použity následující vztahy (23, 24).

$$Q = \frac{1}{1 + SD} \tag{23}$$

$$SD = \sum_{j=1}^{K} \sum_{\forall X_i \in K_j} (d(X_i, C_j))^2 \tag{24}$$

Kde Q je hodnota udávající kvalitu řešení v rozsahu <0, 1> a hodnota 1 značí nejlepší možné řešení. SD je suma kvadratických vzdáleností jednotlivých jedinců v clusterech. K označuje počet clusterů, $\forall X_i \in K_j$ jsou všechny prvky clusteru s indexem j a C_j je centroid clusteru s indexem j.

Vzdálenost $d(X_i, C_j)$ byla v testu použita dvojí – Eukleidovská a Chebyshevova (25, 26).

$$\text{Eukleidovská } d(X_i, C_j) = \sqrt{\sum_{k=1}^{n} (x_k - c_k)^2} \tag{25}$$

$$\text{Chebyshevova } d(X_i, C_j) = \max_k(|x_k - c_k|) \tag{26}$$

Kde x_k značí k-tý atribut řešení X_i a c_k značí k-tý atribut centroidu C_j.

Výsledky algoritmu Random Search jsou v tabulce (Tab. 4) pod označením RS-E pro řešení s Eukleidovskou metrikou a RS-CH pro řešení s Chebyshevovou metrikou.

6.1.2 Differential Evolution - DE

I algoritmus diferenciální evoluce lze za jistých podmínek použít pro clustering. Řešení je zde reprezentováno stejně jako u algoritmu náhodného prohledávání, tedy udává souřadnice centroidů clusterů, rovněž pro ohodnocení řešení jsou využity stejné vztahy (23, 24) a metrika je také dvojí (25, 26).

Algoritmus diferenciální evoluce je v pseudokódu uveden zde, konkrétně se jedná o variantu DERand1Bin:

1 Generování prvotní populace
2 Výpočet hodnot účelové funkce populace
3 Pro každý vektor generace
    a. Náhodný výběr tří různých rodičů r_1, r_2, r_3 pro aktivní vektor x
    b. Výpočet šumového vektoru podle vztahu (27)
    c. Vytvoření zkušebního vektoru
        i. Pro každou dimenzi je generována náhodná hodnota
        ii. Pokud je tato hodnota menší než CR nebo se jedná o náhodně vygenerovanou dimenzi pro křížení, pak je do této dimenze zkušebního vektoru zapsána hodnota z šumového vektoru
        iii. Jinak je zde zapsána hodnota z aktivního vektoru
    d. Pokud je hodnota účelové funkce zkušebního vektoru lepší než hodnota účelové funkce aktivního vektoru, do nové generace postupuje zkušební vektor
    e. Jinak do nové generace postupuje původní vektor
4 Dokud není dosažen maximální počet ohodnocení účelové funkce, opakuje se krok 3

$$v_j = r_{1,j} + F * (r_{2,j} - r_{3,j}) \tag{27}$$

Kde v_j je *j*-tý atribut šumového vektoru *v*, $r_{x,j}$ je *j*-tý atribut *x*-tého rodiče r_x a *F* je mutační konstanta.

Pro oba datasety a oba druhy metrik bylo nastavení řídicích parametrů algoritmu DE následující:

- Velikost populace $NP = 100$.
- Mutační konstanta $F = 1.5$.
- Práh křížení $CR = 0{,}7$.
- Maximání počet ohodnocení účelové funkce $CFE = 100\,000$.

Výsledky algoritmu DE jsou uvedeny v tabulce (Tab. 4) a jsou označeny podle metriky jako DE-E – Eukleidovská metrika a DE-CH – Chebyshevova metrika.

6.1.3 Density-Based Spatial Clustering of Applications with Noise – DBSCAN

DBSCAN algoritmus představený v roce 1996 se liší od přístupů předchozích algoritmů v tom, že clustery nejsou založeny pouze na vzdálenosti bodů od centroidů, ale na hustotě výskytu v prohledávané oblasti. Hlavní myšlenkou je, že uvnitř clusterů je vyšší hustota výskytu bodů, než mimo ně, kde se jedná o šum. [20]

Algoritmus rozděluje body v prostoru na tři typy – **jádrové**, **hraniční** a **šumové**. Pro rozlišení těchto bodů a definici clusterů na základě hustoty jsou potřebné následující definice.

Def. 3.: *Eps-sousedství - Eps-sousedství bodu p, označené $N_{Eps}(p)$, je definováno vztahem (28). [20]*

$$N_{Eps}(p) = \{q \in D | d(p,q) \leq Eps\} \tag{28}$$

Kde *D* je celková množina bodů, *d (p,q)* vzdálenost mezi body *p* a *q* a *Eps* je konstantní parametr vzdálenosti.

Def. 4.: *Přímá dosažitelnost – Bod p je přímo dosažitelný z bodu q, pokud platí vztahy (29, 30). [20]*

$$p \in N_{Eps}(q) \tag{29}$$

$$\text{Podmínka jádrového bodu} - \left|N_{Eps}(q)\right| \geq MinPts \tag{30}$$

Kde *MinPts* je konstantní nejmenší počet bodů, které jsou třeba k označení bodu za jádrový. Přímá dosažitelnost je symetrická pro pár jádrových bodů. Pokud bod *p* nesplňuje

podmínku jádrového bodu, ale je přímo dosažitelný z jádrového bodu *q*, jedná se o bod hraniční. [20]

Def. 5.: *Dosažitelnost – Bod p je dosažitelný z bodu q, pokud existuje řetězec bodů* $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ *takových, že* p_{i+1} *je přímo dosažitelný z* p_i. *[20]*

Dosažitelnost je tranzitivní, symetrická pouze pro jádrové body. [20]

Def. 6.: *Propojení – Bod p je propojený s bodem q, pokud existuje bod o, ze kterého jsou oba body p i q dosažitelné. [20]*

Propojení je symetrické, pro dosažitelné body i reflexivní. [20]

Cluster je tedy maximální množina bodů, které jsou propojené. Šum jsou pak body, které nespadají do žádného clusteru.

Základní algoritmus DBSCAN pro svou funkci potřebuje nastavení dvou parametrů – *Eps* a *MinPts*. Obě tyto konstanty poté rozhodují o množství vytvořených clusterů. Algoritmus je v pseudokódu uveden zde:

```
5 Všechny body jsou na začátku algoritmu označeny jako NEKLASIFIKOVANÉ
6 Pro všechny body z datasetu
      a. Pokud je bod NEKLASIFIKOVÁN
            i. Rozšíření clusteru
                  1. Pokud bod splňuje jádrovou podmínku, všechny
                     dosažitelné body jsou označeny stejným cluster
                     identifikátorem
                        a. Pro každý dosažitelný bod test podmínky
                           jádra a všechny dosažitelné body jsou
                           klasifikovány stejným cluster
                           identifikátorem
                        b. Opakování předchozího bodu, dokud už nejsou
                           žádné dosažitelné NEKLASIFIKOVANÉ body
                  2. Pokud nesplňuje je klasifikován jako ŠUM
7 Konec
```

Takto vytvořené clustery jsou pro stejně nastavené *Eps* a *MinPts* konstanty vždy stejné a proto není třeba algoritmus spouštět vícekrát. Nastavení řídicích parametrů pro Iris dataset bylo následující:

Eukleidovská metrika

- Parametr vzdálenosti *Eps* = 0,47.
- Nejmenší počet bodů *MinPts* = 9.

Chebyshevova metrika

- Parametr vzdálenosti *Eps* = 0,4.
- Nejmenší počet bodů *MinPts* = 14.

Nastavení řídicích parametrů pro Wine dataset:

Eukleidovská metrika

- Parametr vzdálenosti *Eps* = 3.
- Nejmenší počet bodů *MinPts* = 1.

Chebyshevova metrika

- Parametr vzdálenosti *Eps* = 2.
- Nejmenší počet bodů *MinPts* = 1.

Nastavení řídicích parametrů bylo testováno a zvolené hodnoty poskytovaly nejlepší výsledky. Výsledky jsou uvedeny v tabulce (Tab. 4) a jsou označeny DBSCAN-E pro Eukleidovskou metriku a DBSCAN-CH pro Chebyshevovu metriku.

6.1.4 Fuzzy K-Means – FKM

Algoritmus FKM je rozšířením klasického algoritmu K-Means o využití fuzzy náležitosti. V K-Means algoritmu se centroid clusteru získá zprůměrováním všech hodnot atributů bodů spadajících do tohoto clusteru. Pro náležitost do clusteru se opět využívá vzdálenosti mezi body. V FKM algoritmu je navíc zavedena částečná příslušnost bodu do clusteru, tedy každý bod s nějakou vahou spadá do všech clusterů a centroidy clusterů se vypočítávají s ohledem na tyto váhy pomocí následujících vztahů (31, 32). [21]

$$\text{Výpočet váhy} - \ w_{i,j} = \frac{1}{\sum_{k=1}^{K} \left(\frac{d(X_i, C_j)}{d(X_i, C_k)} \right)^{\frac{2}{m-1}}} \tag{31}$$

$$\text{Výpočet centroidu } k - \ C_k = \frac{\sum_{i=1}^{n} {w_{i,k}}^m * X_i}{\sum_{i=1}^{n} {w_{i,k}}^m} \tag{32}$$

Vztah (31) ukazuje, jak je možné spočítat váhu $w_{i,j}$ *i*-tého prvku vzhledem k *j*-tému clusteru, kde *K* je celkový počet clusterů, $d(X_i, C_j)$ je funkce vzdálenosti podle vztahu (25) nebo (26) mezi bodem X_i a centroidem clusteru C_j, *m* je konstantní fuzzyfikační faktor. Vztah (32) udává funkci pro výpočet centroidu C_k *k*-tého clusteru, kde *n* je celkový počet prvků datasetu.

Nalezení vhodných centroidů clusterů probíhá pomocí minimalizace funkce dané vztahem (33). [21]

$$J = \sum_{i=1}^{n} \sum_{j=1}^{K} {w_{i,j}}^{m} * d(X_i, C_j)^2 \tag{33}$$

Parametry potřebné pro spuštění FKM algoritmu z knihovny Apache Commons Math 3.5 jsou následující:

Iris dataset:

- Počet clusterů $K = 3$.

Eukleidovská metrika

- Fuzzyfikační faktor $m = 1{,}3$.

Chebyshevova metrika

- Fuzzyfikační faktor $m = 1{,}05$.

Wine dataset:

- Počet clusterů $K = 3$.

Eukleidovská metrika

- Fuzzyfikační faktor $m = 12$.

Chebyshevova metrika

- Fuzzyfikační faktor $m = 12$.

Nastavení fuzzyfikačního faktoru bylo testováno a byla vybrána hodnota poskytující nejlepší výsledky. Výsledky FKM algoritmu jsou uvedeny v tabulce (Tab. 4) a jsou označeny podle použité metriky: FKM-E – Eukleidiovská metrika a FKM-CH – Chebyshevova metrika.

6.1.5 K-Means Plus Plus – KMPP

KMPP algoritmus je K-means algoritmus rozšířený o výběr počátečních centroidů clusterů nikoliv náhodně, ale postupně v krocích. První centroid je vybrán náhodně, poté jsou vypočteny vzdálenosti (25, 26) všech bodů datasetu vůči tomuto centroidu a následující centroid je zvolen opět náhodně, ale s využitím vážené pravděpodobnosti, kdy pravděpodobnost, že bude bod zvolen je závislá na dříve vypočtené vzdálenosti. Toto je

opakováno, dokud nejsou vybrány všechny centroidy. Úprava K-means algoritmu je jednoduchá, ale velice účinná, zvyšující rychlost konvergence a kvalitu řešení. [21]

KMPP algoritmus byl v této práci omezen maximálním množstvím ohodnocení účelové funkce *CFE* = 100 000. Výsledky jsou uvedeny v tabulce (Tab. 4) a jsou označeny podle použité metriky: KMPP-E – Eukleidovská metrika a KMPP-CH – Chebyshevova metrika.

6.2 Základní výsledky

Zde jsou uvedeny základní statistické výsledky jednotlivých algoritmů a jsou vyobrazeny v tabulce (Tab. 4). Testování probíhalo na jednom stroji, kdy posloupnost spouštění jednotlivých algoritmů byla stále stejná:

1. RS – IrisDataset – Eukleidovská metrika.
2. RS – IrisDataset – Chebyshevova metrika.
3. RS – WineDataset – Eukleidovská metrika.
4. RS – WineDataset – Chebyshevova metrika.
5. DE – IrisDataset – Eukleidovská metrika.
6. DE – IrisDataset – Chebyshevova metrika.
7. DE – WineDataset – Eukleidovská metrika.
8. DE – WineDataset – Chebyshevova metrika.
9. DBSCAN – IrisDataset – Eukleidovská metrika.
10. DBSCAN – IrisDataset – Chebyshevova metrika.
11. DBSCAN – WineDataset – Eukleidovská metrika.
12. DBSCAN – WineDataset – Chebyshevova metrika.
13. FKM – IrisDataset – Eukleidovská metrika.
14. FKM – IrisDataset – Chebyshevova metrika.
15. FKM – WineDataset – Eukleidovská metrika.
16. FKM – WineDataset – Chebyshevova metrika.
17. KMPP – IrisDataset – Eukleidovská metrika.
18. KMPP – IrisDataset – Chebyshevova metrika.
19. KMPP – WineDataset – Eukleidovská metrika.
20. KMPP – WineDataset – Chebyshevova metrika.

Konkrétní nastavení jednotlivých algoritmů je uvedeno v předcházející části této práce. Nastavení řídicích parametrů testu bylo následující:

- Maximální počet ohodnocení účelové funkce *CFE* = 100 000.
- Počet běhů *Runs* = 1 000.

6.2.1 Výsledky bez normalizace datasetů

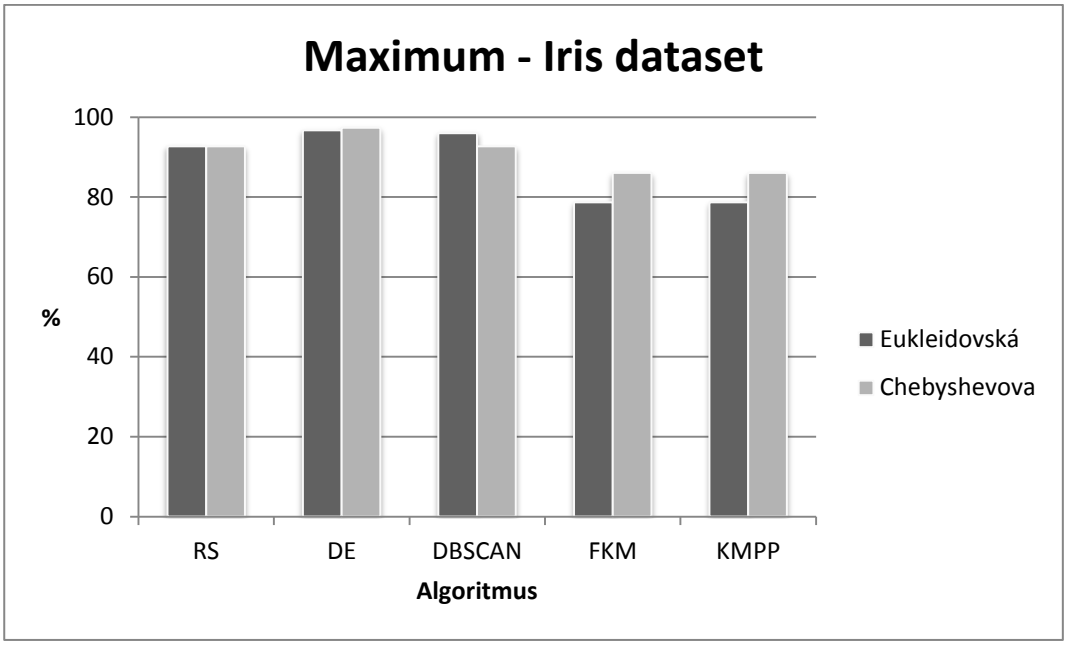
Tab. 4. Výsledky testu clustering algoritmů – bez normaliazce.

	Iris dataset				**Wine dataset**			
Algoritmus	**Max [%]**	**Avg [%]**	**Median [%]**	**STD [%]**	**Max [%]**	**Avg [%]**	**Median [%]**	**STD [%]**
RS-E	92,67	87,16	88,00	4,79	71,35	33,85	35,39	20,32
RS-CH	92,67	84,03	86,00	7,96	70,79	32,73	20,22	20,14
DE-E	96,67	89,17	89,33	2,37	70,79	32,36	18,54	20,39
DE-CH	97,33	88,32	88,67	4,41	70,79	34,04	35,39	20,71
DBSCAN-E	96,00	96,00	96,00	0,00	74,16	74,16	74,16	0,00
DBSCAN-CH	92,67	92,67	92,67	0,00	75,28	75,28	75,28	0,00
FKM-E	78,67	78,57	78,67	1,07	70,79	32,82	20,79	21,04
FKM-CH	86,00	84,00	82,67	2,76	70,79	33,02	36,52	21,63
KMPP-E	78,67	72,32	66,67	5,99	70,79	27,91	20,79	21,88
KMPP-CCH	86,00	82,50	82,67	5,84	70,79	29,76	20,79	22,40

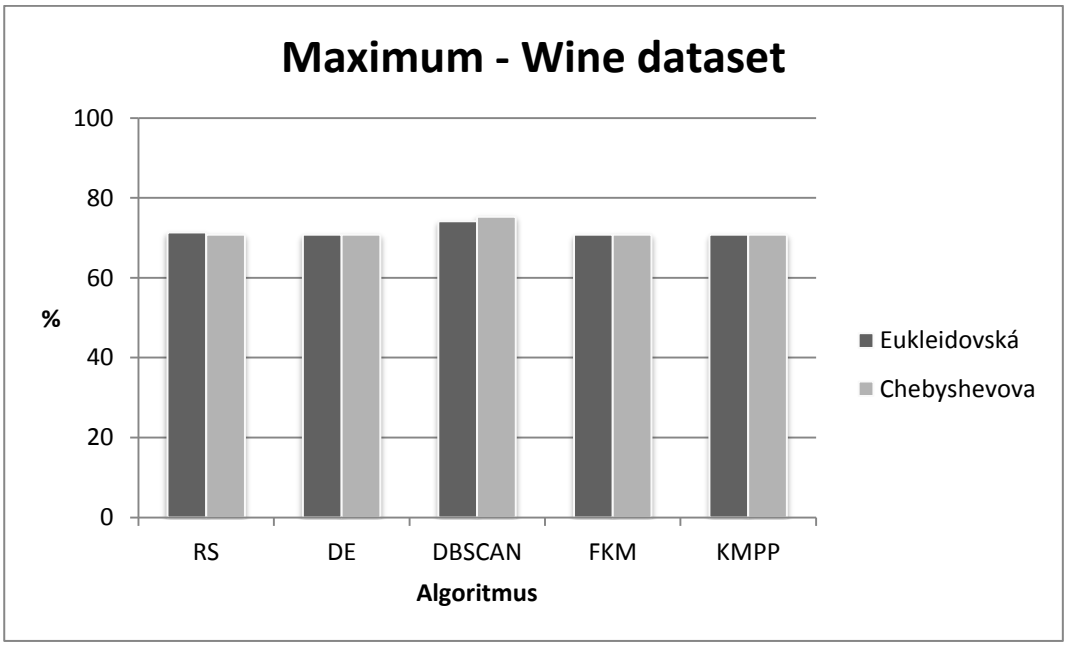
Vysvětlivky k tabulce (Tab. 4):

Max – maximum, **Avg** – průměrná hodnota, **Median** – medián, **STD** (Standard Deviation) – směrodatná odchylka.

Pro přehlednost jsou výsledky vyobrazeny na obrázcích (Obr. 16 až Obr. 23). Vždy jsou vyobrazeny výsledné hodnoty v procentech a jsou rozděleny podle metriky na Eukledisovskou a Chebyshevovu.

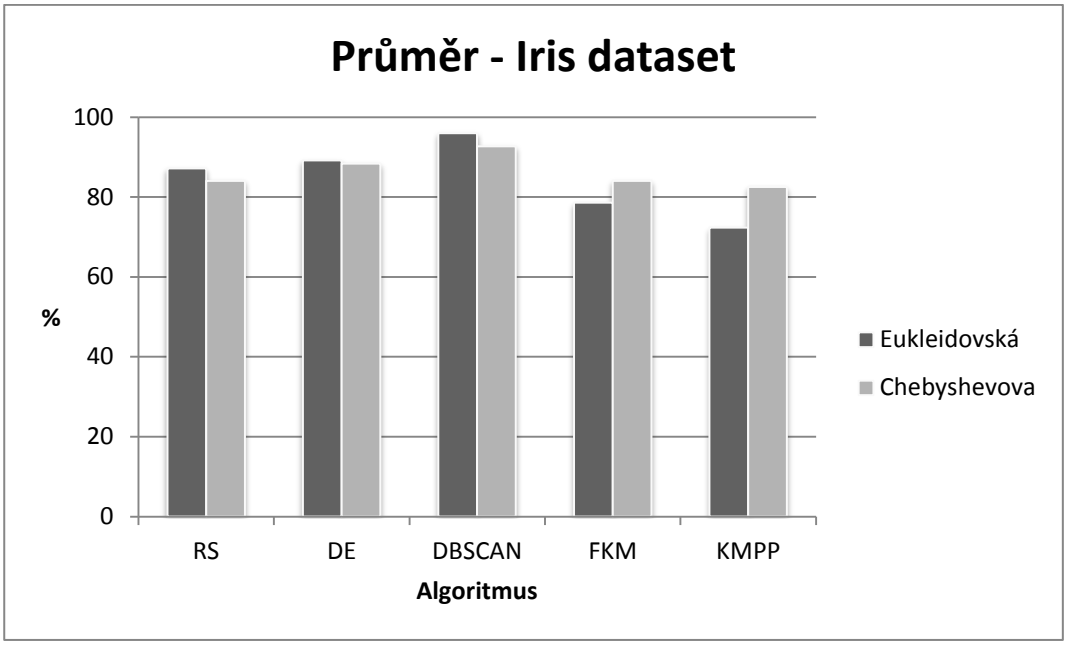


Obr. 16. Graf porovnaných maximálních hodnot jednotlivých algoritmů na Iris datasetu.

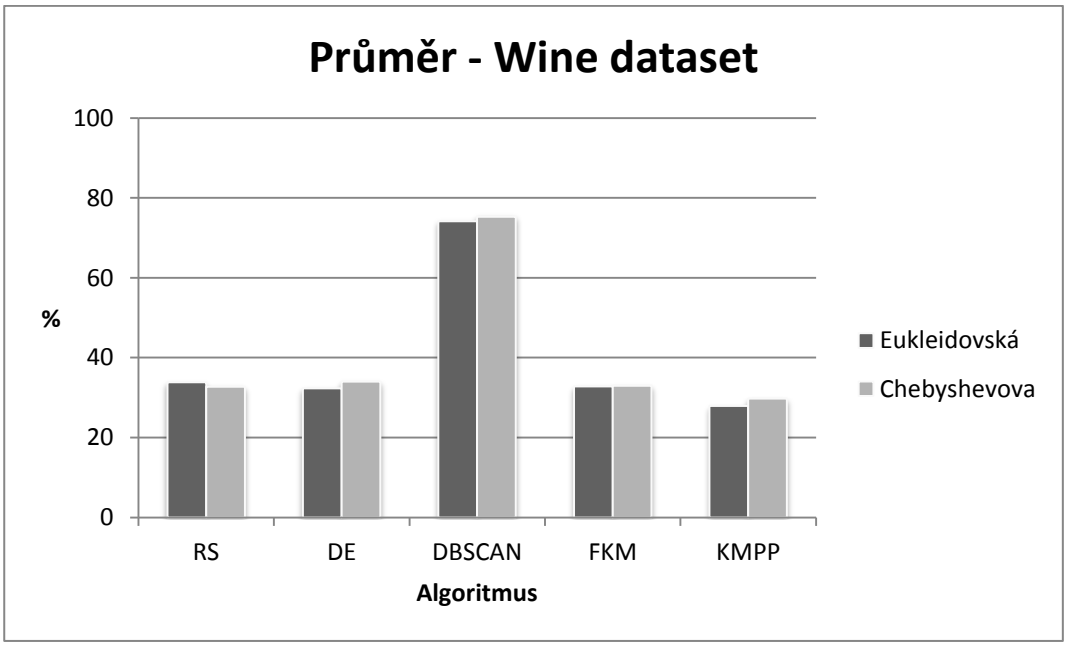


Obr. 17. Graf porovnaných maximálních hodnot jednotlivých algoritmů na Wine datasetu.

Z obrázků (Obr. 16, Obr. 17) je patrné, že dosažená hodnota jednotlivých algortmů je nezávislá na použité metrice. Dále je možné vypozorovat, že maximální úspěšnost clusteringu je u vícedimenzionálního Wine datasetu nižší, což je pravděpodobně způsobeno rozdílnými rozsahy jednotlivých atributů – toto je možné odstranit normalizací datasetu, jak je naznačeno níže v této práci.

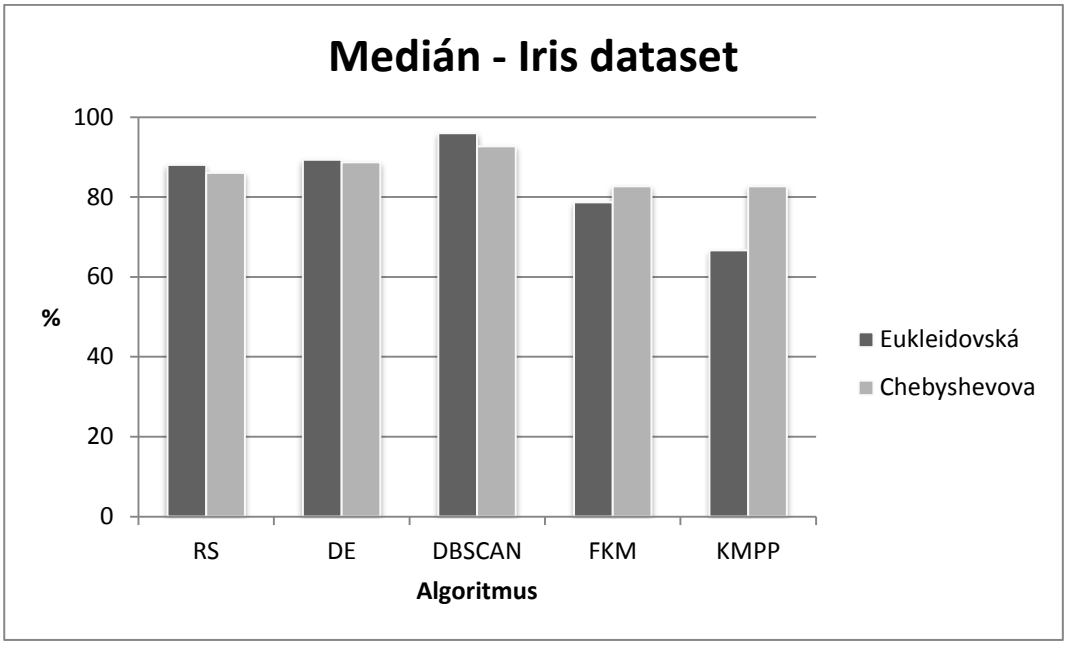


Obr. 18. Graf porovnaných průměrných hodnot jednotlivých algoritmů na Iris datasetu.

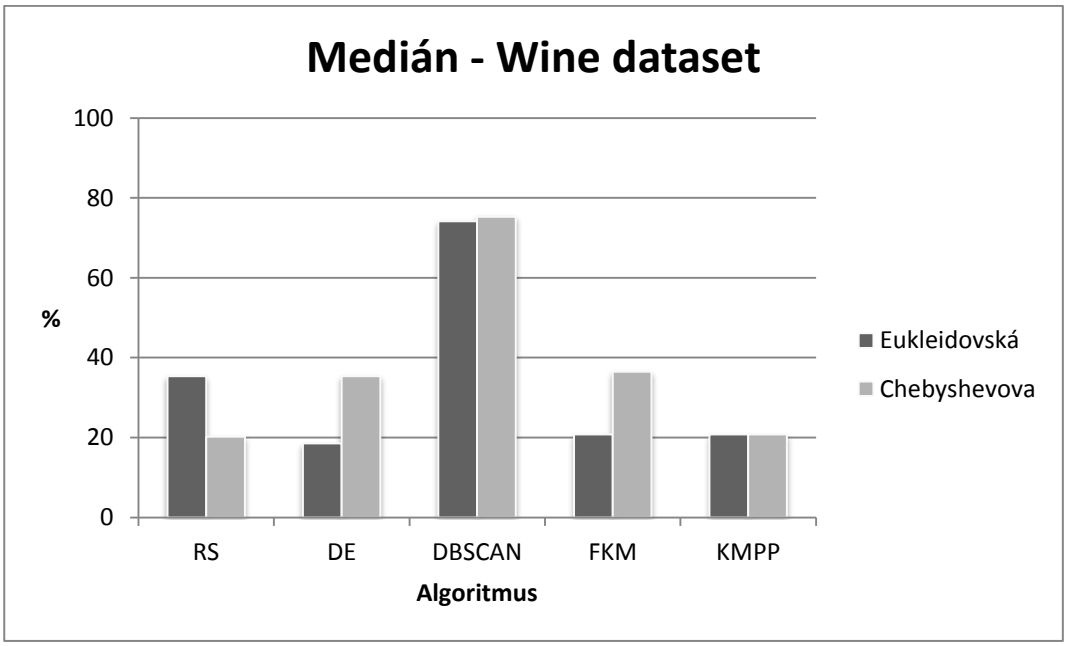


Obr. 19. Graf porovnaných průměrných hodnot jednotlivých algoritmů na Wine datasetu.

Z obrázků (Obr. 18, Obr. 19) je patrné, že průměrná úspěšnost clusteringu je výrazně nižší u Wine datasetu. Vysoká úspěšnost DBSCAN algoritmu je způsobena tím, že jeho výstup je pokaždé stejný, tedy jak průměrná hodnota, tak medián jsou stejné jako maximální hodnota a směrodatná odchylka je rovna nule, což je vidět i v následujcících obrázcích (Obr. 20 až Obr. 23).

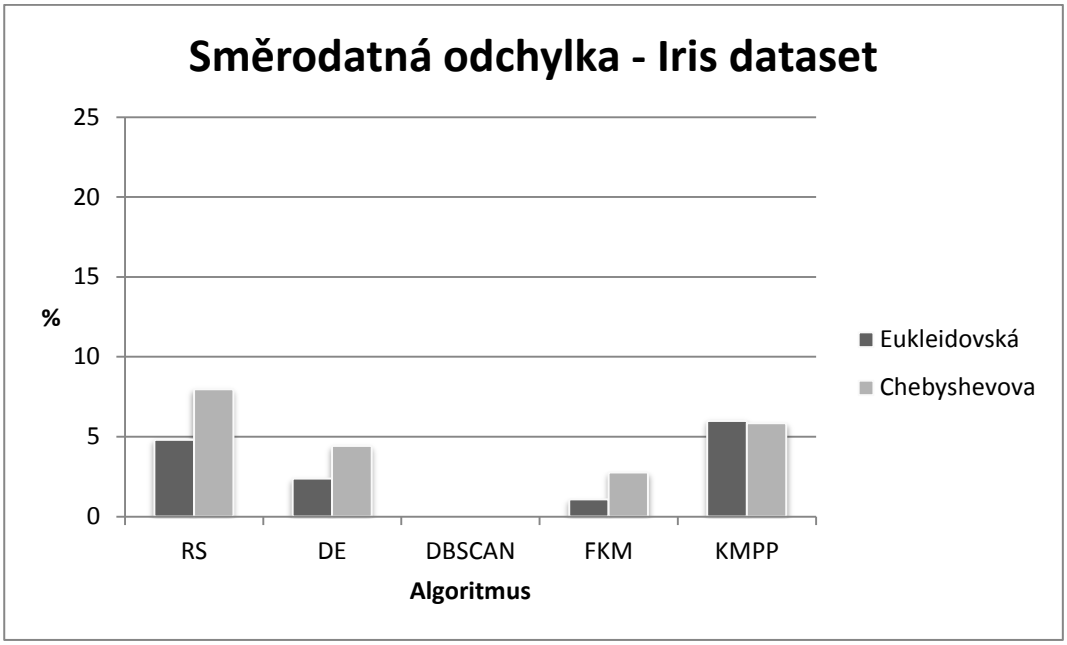


Obr. 20. Graf porovnaných hodnot mediánu jednotlivých algoritmů na Iris datasetu.

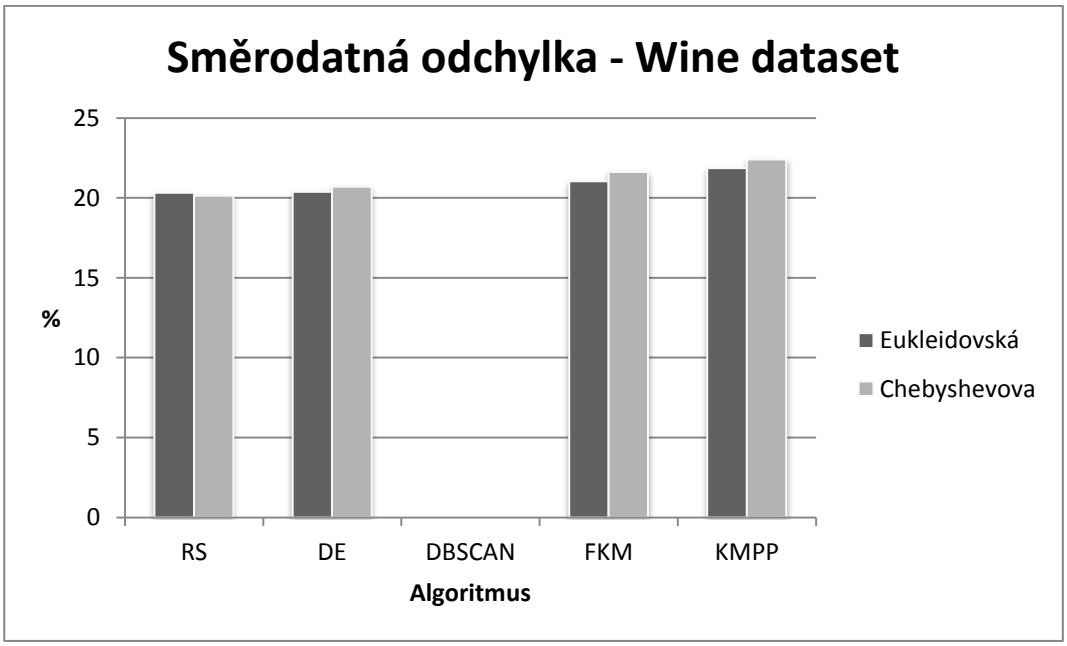


Obr. 21. Graf porovnaných hodnot mediánu jednotlivých algoritmů na Wine datasetu.

Z obrázků (Obr. 20, Obr. 21) je opět patrné, že dosažená mediánová úspěšnost clusteringu na Wine datasetu je výrazně nižší, než na Iris datasetu.



Obr. 22. Graf porovnaných hodnot směrodatné odchylky jednotlivých algoritmů na Iris datasetu.



Obr. 23. Graf porovnaných hodnot směrodatné odchylky jednotlivých algoritmů na Wine datasetu.

Na obrázcích (Obr. 22, Obr. 23) je zřejmé, že vyšší dimenzionalita způsobuje větší rozptyl výsledků jednotlivých algoritmů a opět na použité metrice příliš nezáleží.

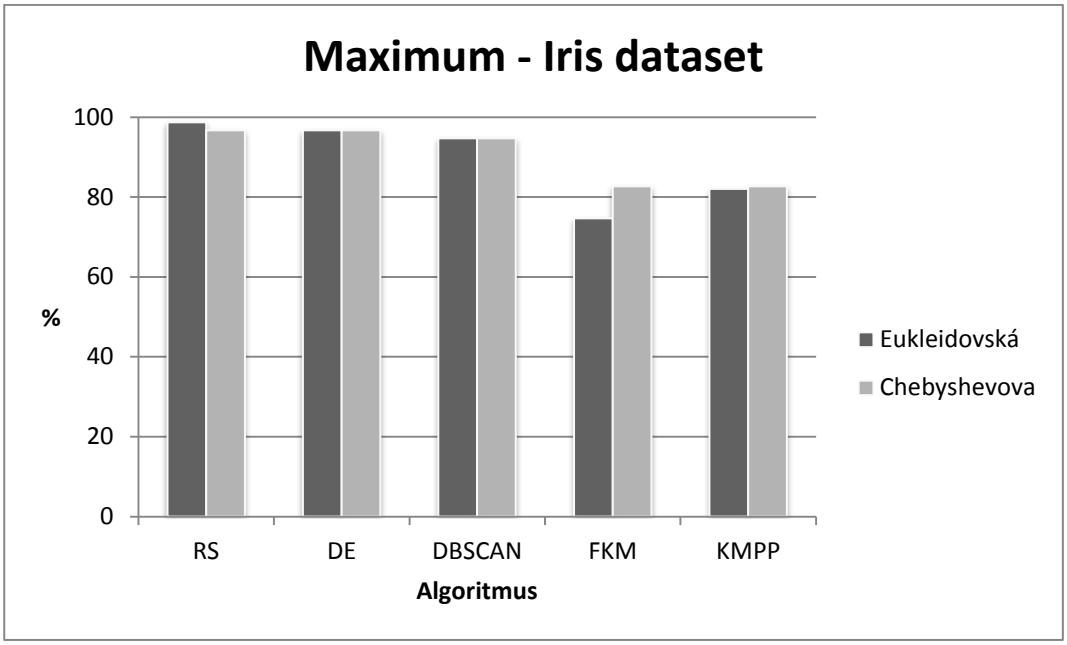
6.2.2 Výsledky po normalizaci datasetů

V této části jsou uvedeny výsledky jednotlivých algoritmů na obou datasetech. Na hodnoty atributů záznamů v datasetech byla použita min-max normalizace do rozsahu <0, 1>. Po normalizaci datasetu bylo třeba přenastavit řídící parametry některých algoritmů – DBSCAN (Iris – *Eps* = 0,17, *MinPts* = 17 a *Eps* = 0,12, *MinPts* = 12, Wine – *Eps* = 0,51, *MinPts* = 21 a *Eps* = 0,23, *MinPts* = 10) a FKM (Iris – *m* = 5,9 a *m* = 19,99, Wine – *m* = 1,09 a *m* = 1,03).

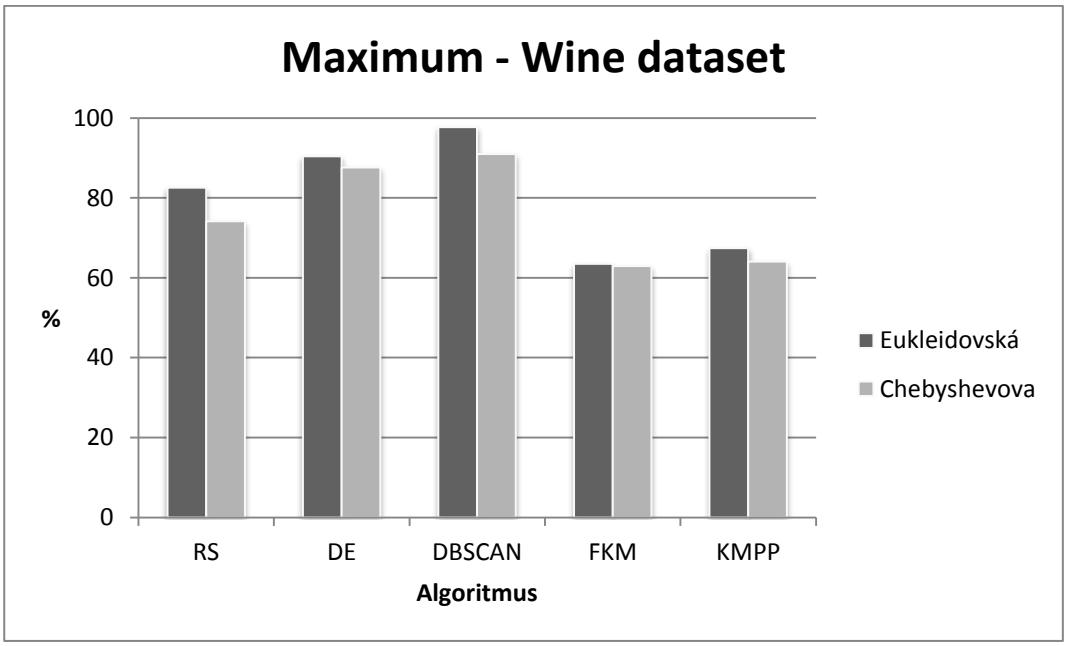
Tab. 5. Výsledky testu clustering algoritmů – s normalizací.

	Iris dataset				**Wine dataset**			
Algoritmus	**Max [%]**	**Avg [%]**	**Median [%]**	**STD [%]**	**Max [%]**	**Avg [%]**	**Median [%]**	**STD [%]**
RS-E	98,67	85,36	86,00	5,74	82,58	34,08	34,83	18,47
RS-CH	96,67	78,51	82,00	9,56	74,16	33,26	33,15	13,69
DE-E	96,67	88,25	88,00	2,85	90,45	32,88	33,71	21,56
DE-CH	96,67	85,61	86,67	5,98	87,64	32,98	33,71	18,92
DBSCAN-E	94,67	94,67	94,67	0,00	97,75	97,75	97,75	0,00
DBSCAN-CH	94,67	94,67	94,67	0,00	91,01	91,01	91,01	0,00
FKM-E	74,67	74,67	74,67	0,00	63,48	32,59	25,28	21,95
FKM-CH	82,67	82,31	82,67	0,69	62,92	32,79	25,84	18,06
KMPP-E	82,00	76,12	74,67	4,73	67,42	33,99	25,84	21,24
KMPP-CCH	82,67	72,19	70,00	5,60	64,04	34,07	35,96	18,44

Porovnání základních statistických údajů podle použité metriky je uvedeno v grafech na následujících obrázcích (Obr. 24 až Obr. 31).

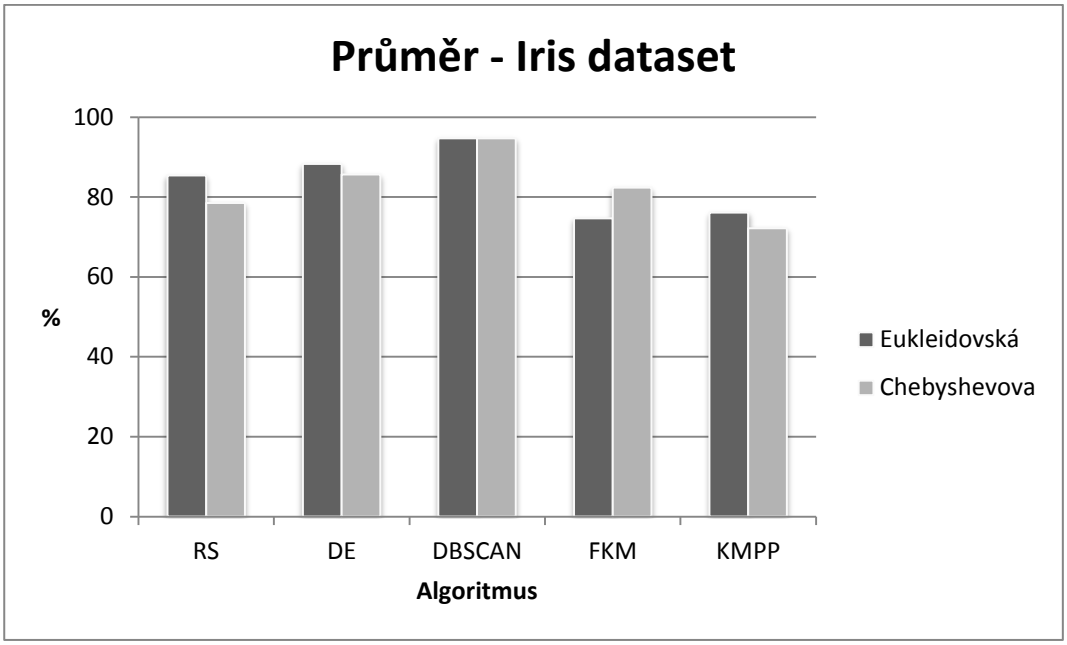


Obr. 24. Graf porovnaných maximálních hodnot jednotlivých algoritmů na Iris datasetu.

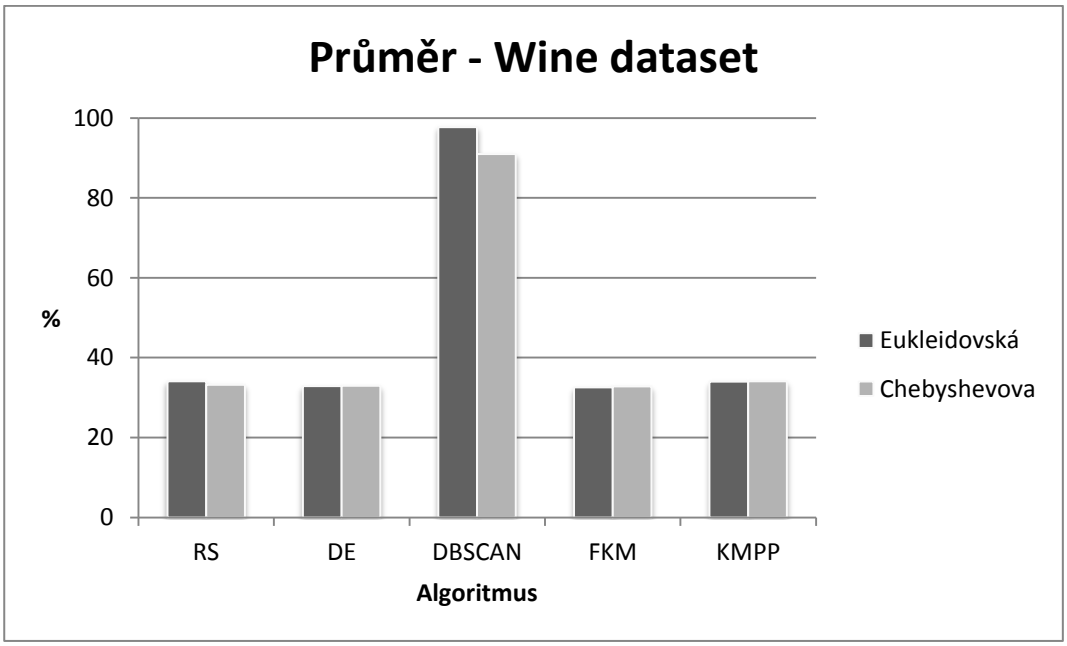


Obr. 25. Graf porovnaných maximálních hodnot jednotlivých algoritmů na Wine datasetu.

Z grafů v obrázcích (Obr. 24, Obr. 25) je patrné, že použitá metrika nemá příliš velký vliv na získanou maximální úspěšnost klasifikace. Tím, že jsou oba datasety normalizované a rozsahy hodnot jednotlivých atributů jsou stejné, se potvrdil předpoklad, že algoritmy dosahují lepších maximálních výsledků na méně dimenzionálních datech.

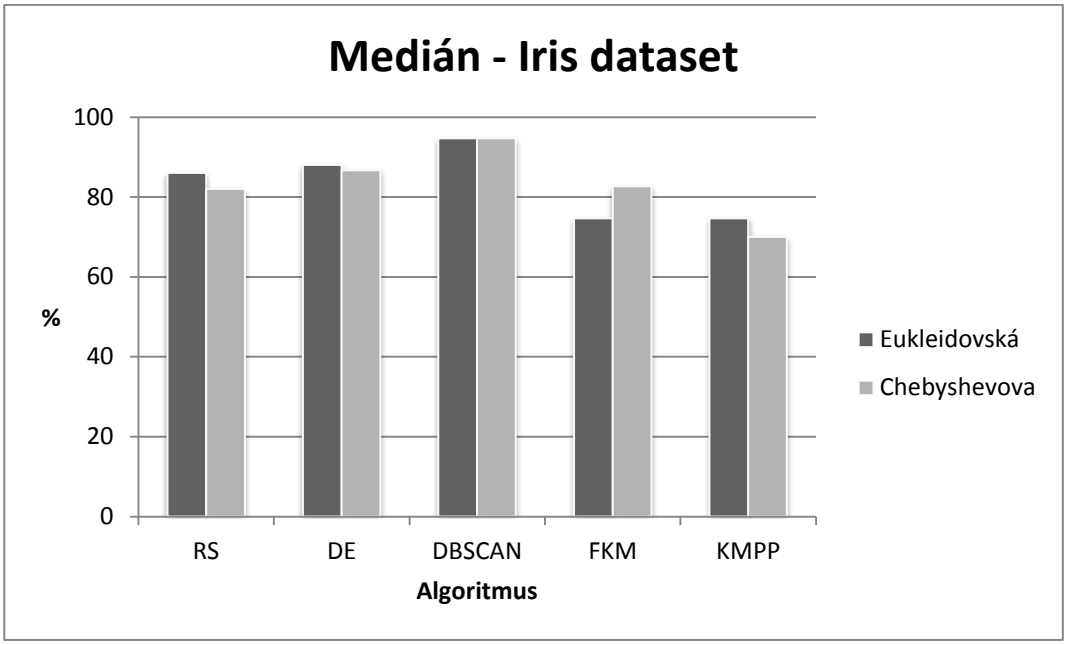


Obr. 26. Graf porovnaných průměrných hodnot jednotlivých algoritmů na Iris datasetu.

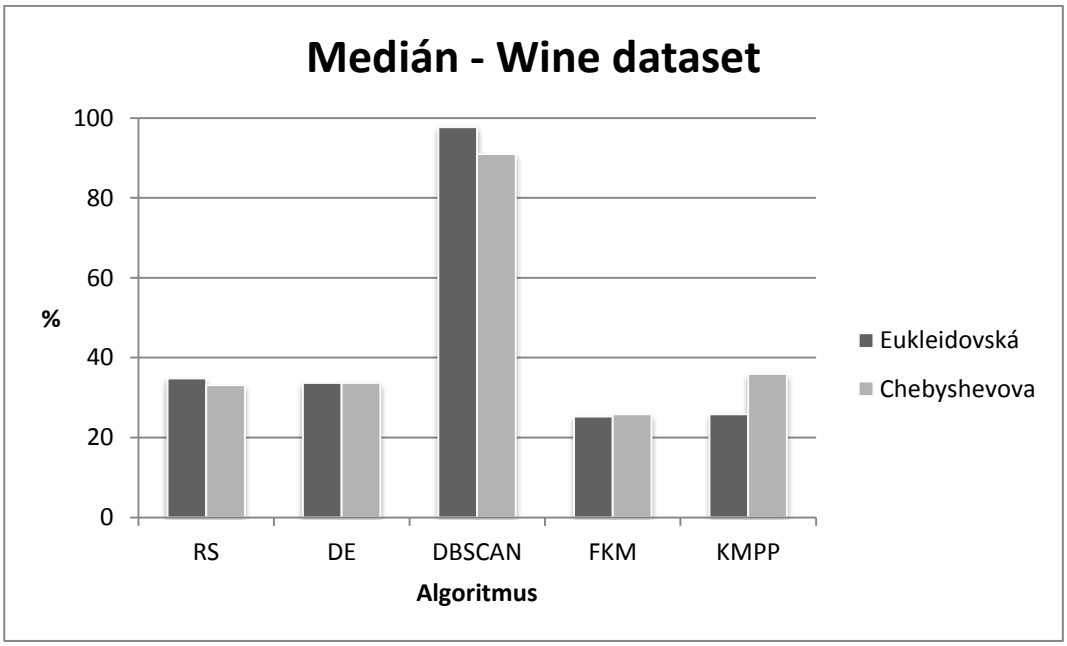


Obr. 27. Graf porovnaných průměrných hodnot jednotlivých algoritmů na Wine datasetu.

Z grafů v obrázcích (Obr. 26, Obr. 27) je patrné, že kromě DBSCAN algoritmu je průměrný výsledek lepší na Iris datasetu, takže se dá předpokládat závislost na dimenzionalitě problému. Opět je hodnota DBSCAN algoritmu výrazně vyšší, než u ostatních algoritmů, protože je výsledek algoritmu vždy stejný a tedy roven maximální dosažené úspěšnosti.

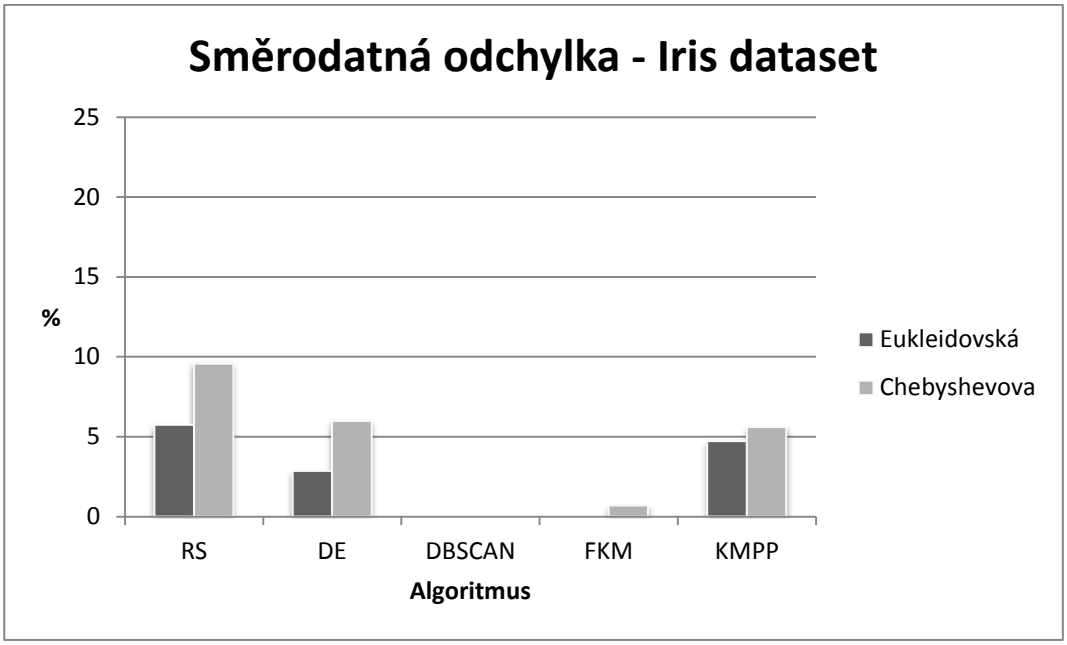


Obr. 28. Graf porovnaných hodnot mediánu jednotlivých algoritmů na Iris datasetu.

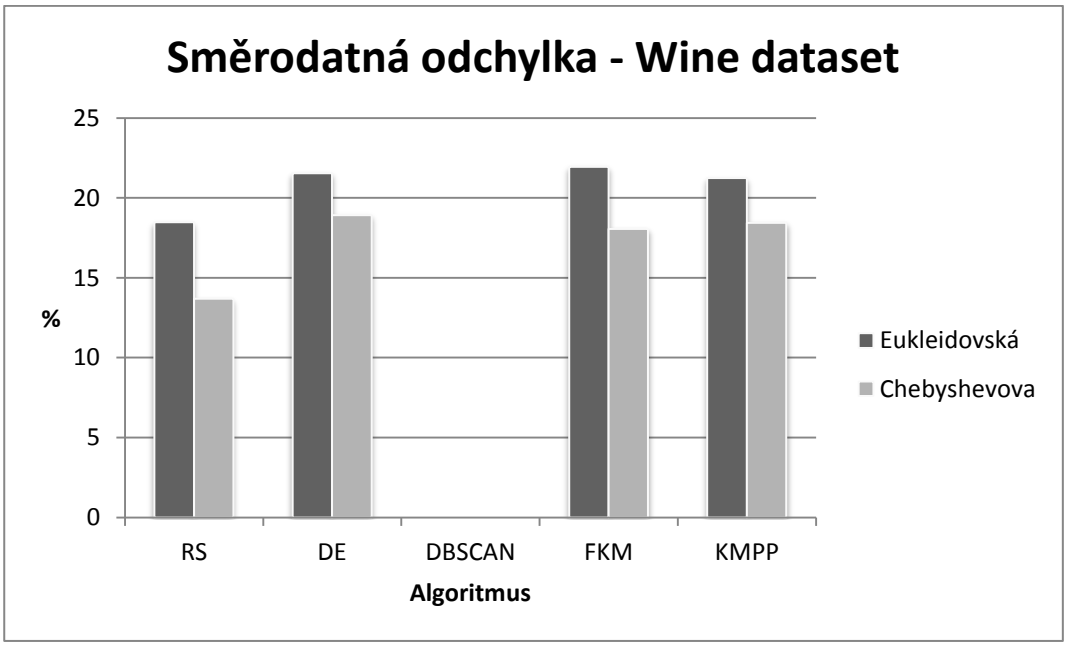


Obr. 29. Graf porovnaných hodnot mediánu jednotlivých algoritmů na Wine datasetu.

Z grafů v obrázcích (Obr. 28, Obr. 29) je patrné, že Wine dataset je pro algoritmy složitější, s vyjímkou DBSCAN algoritmu, kterému vyšší dimenzionalita problému nečinní problémy.



Obr. 30. Graf porovnaných hodnot směrodatné odchylky jednotlivých algoritmů na Iris datasetu.



Obr. 31. Graf porovnaných hodnot směrodatné odchylky jednotlivých algoritmů na Wine datasetu.

Z grafů v obrázcích (Obr. 30, Obr. 31) je patrné, že směrodatná odchylka je výrazně vyšší u vícedimenzionálního Wine datasetu.

6.3 Zavedení penalizace do výpočtu účelové funkce

Algoritmy RS a DE využívají pro určení kvality řešení účelovou funkci danou vztahy (23, 24). Problém nastává u lineárně neseparovatelných tříd, jako je tomu například u Iris datasetu, kde jsou neseparovatelné třídy Virginica a Versicolour. Může nastat případ, kdy je jeden nebo více centroidů v podstatě nevyužito, protože jim není přiřazen žádný prvek datasetu, tudíž jejich suma kvadratických vzdáleností prvků $SD = 0$. Takové řešení může být ohodnoceno velmi dobře, i když reálná kvalita řešení je nízká, protože je snížen počet aktivních centroidů. Tento problém lze řešit zavedením penalizace.

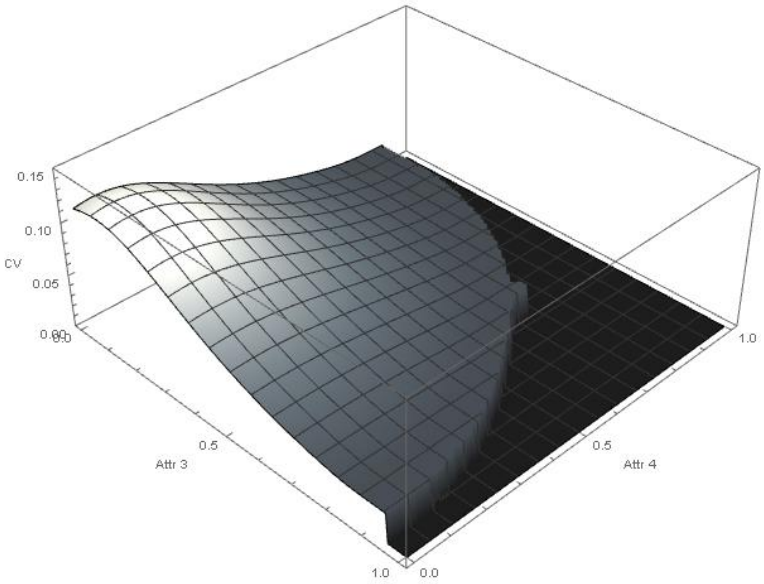
Penalizace zde byla zavedena tak, že pro centroidy, které nemají přiřazen žádný prvek není $SD = 0$, ale $SD = (2 - 2^{-52}) * 2^{1023}$. Použitím tak vysoké konstanty je dané řešení degradováno a v rámci evoluce nepřežije do následujících generací. Výsledky na normalizovaných a nenormalizovaných datasetech jsou uvedeny v tabulce (Tab. 6) a porovnání úspěšnosti klasifkace algoritmů RS a DE s využitím penalizace je zobrazeno na obrázcích (Obr. 38, Obr. 39) pro nenormalizovaná a obrázcích (Obr. 40, Obr. 41) pro normalizovaná data.

Nastavení parametrů testů bylo stejné, jako při testování bez penalizace a opět byly testovány oba datasety s využitím jak Eukleidovské, tak Chebyshevovy metriky.

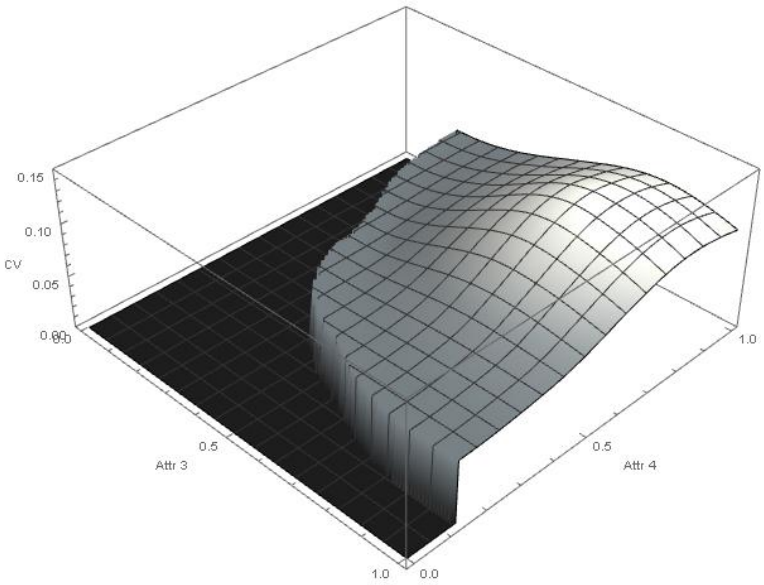
6.3.1 Účelová funkce Iris datasetu

Účelová funkce, která je použita algoritmy náhodného prohledávání a diferenciální evoluce je založena na vzdálenostech jednotlivých záznamů datasetu od centroidů clusterů. Protože jsou v Iris datasetu tři třídy, řešení dosazované do účelové funkce obsahuje souřadnice tří clusterů. Každý cluster má čtyři atributy, takže hodnota účelové funkce je závislá na dvanácti proměnných (tři krát čtyři atributy). Zobrazení závislosti hodnoty účelové funkce na souřadnicích clusterů tedy není možné běžným způsobem.

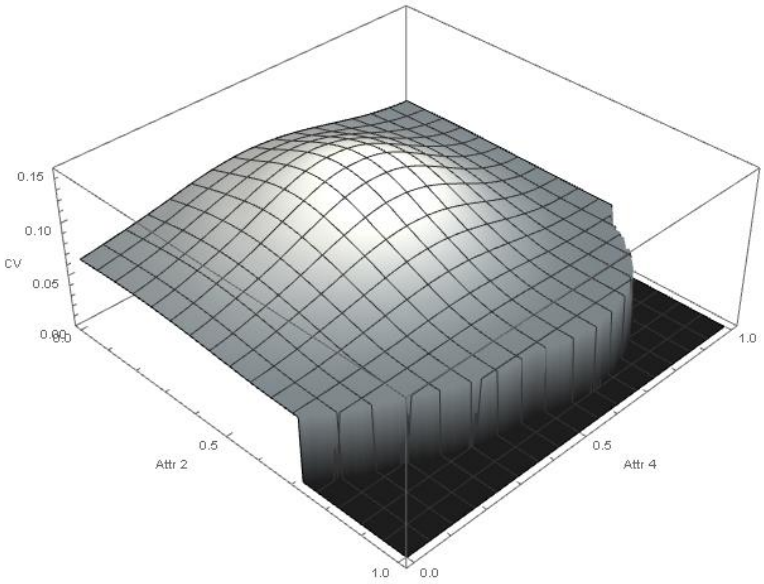
Protože jsou známé třídy jednotlivých záznamů Iris datasetu, bylo možné spočítat centroidy jednotlivých skupin jako průměr všech bodů. Následující obrázky (Obr. 32 až Obr. 34) zobrazují závislost účelové funkce na dvou atributech jednoho z clusterů, zbylé atributy tohoto clusteru mají stejnou hodnotu, jako spočítaný cluster a další dva clustery jsou také statické a vypočítané z datasetu.



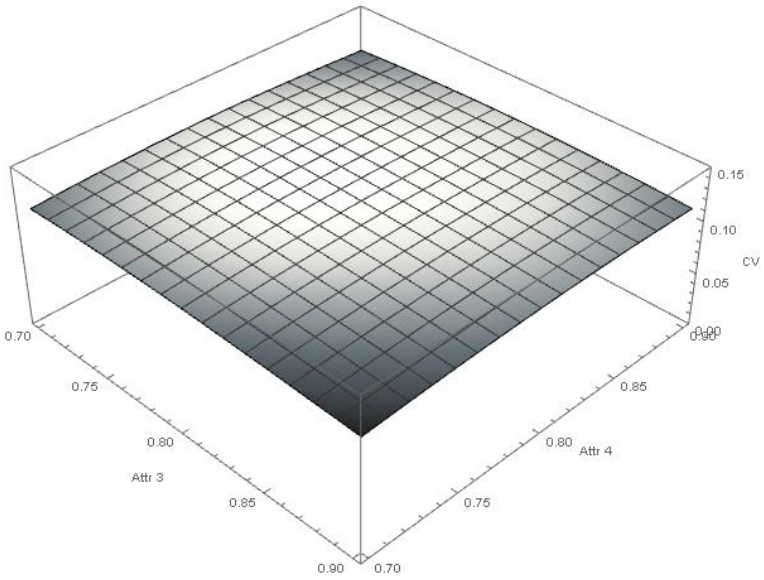
Obr. 32. Graf závislosti hodnoty účelové funkce na délce a šířce okvětního lístku Iris setosa.



Obr. 33. Graf závislosti hodnoty účelové funkce na délce a šířce okvětního lístku Iris virginica.



Obr. 34. Graf závislosti hodnoty účelové funkce na šířkách kališního a okvětního lístku Iris versicolour.

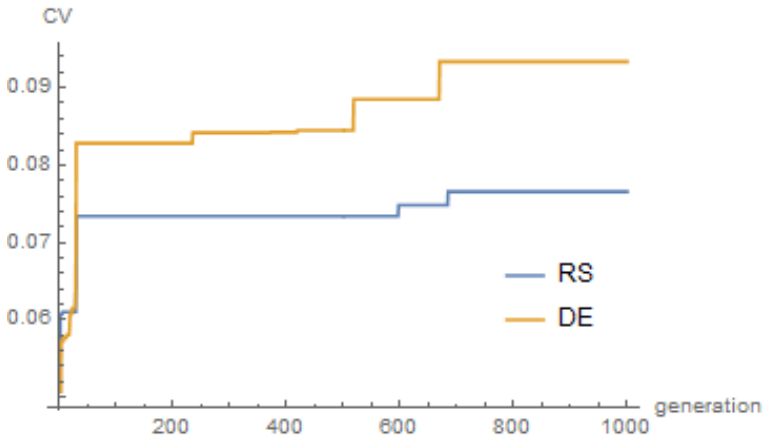


Obr. 35. Graf závislosti hodnoty účelové funkce na délce a šířce okvětního lístku Iris virginica v detailu.

V grafech jsou patrné ploché oblasti s velmi nízkou hodnotou účelové funkce. Tyto oblasti jsou důsledkem zavedení penalizace. Na obrázku (Obr. 35) je v detailu zabrána plochá oblast globálního extrému.

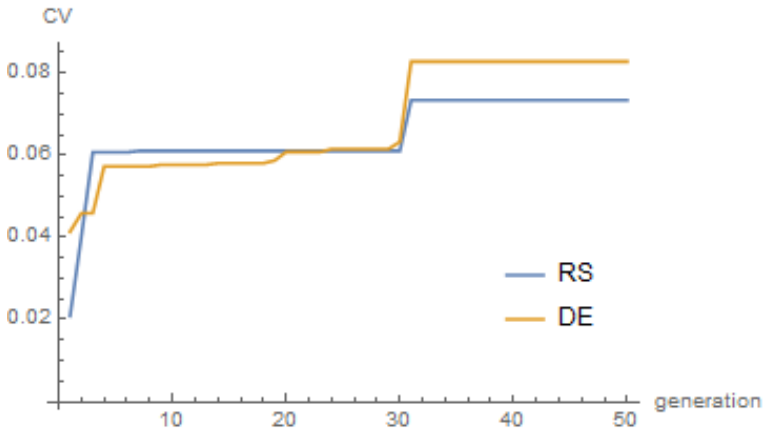
6.3.2 Vývoj hodnot účelové funkce na Iris datasetu

V grafu na obrázku (Obr. 36) je zobrazen příklad průběhu vývoje hodnoty účelové funkce nejlepšího prvku generace v algoritmech RS a DE s využitím Eukleidovské metriky a penalizace účelové funkce. Protože se v algoritmu RS nerozlišují generace, je jednou generací myšleno sto ohodnocení účelové funkce. Vývoj je zobrazen pro náhodně vybraný běh algoritmů. Ukončující podmínkou bylo 100 000 ohodnocení účelové funkce.



Obr. 36. Graf vývoje hodnoty účelové funkce nejlepšího prvku algoritmů RS a DE na Iris datasetu.

Na obrázku (Obr. 37) je zobrazen detailněji průběh prvních 50 generací.



Obr. 37. Detail vývoje hodnoty účelové funkce nejlepšího prvku algoritmů RS a DE na Iris datasetu – prvních 50 generací.

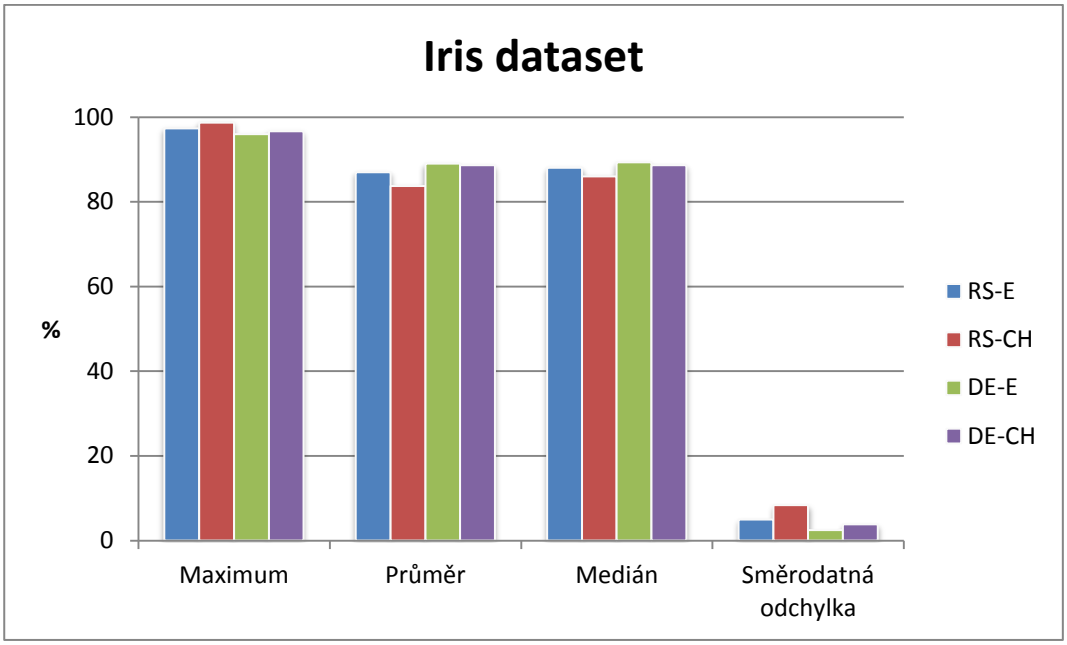
6.3.3 Výsledky

Z tabulky (Tab. 6) a z grafů v obrázcích (Obr. 38 až Obr. 41) je patrné, že na použité metrice opět příliš nezáleží. Dále je zřejmé, že výsledky clusteringu na Iris datasetu, který

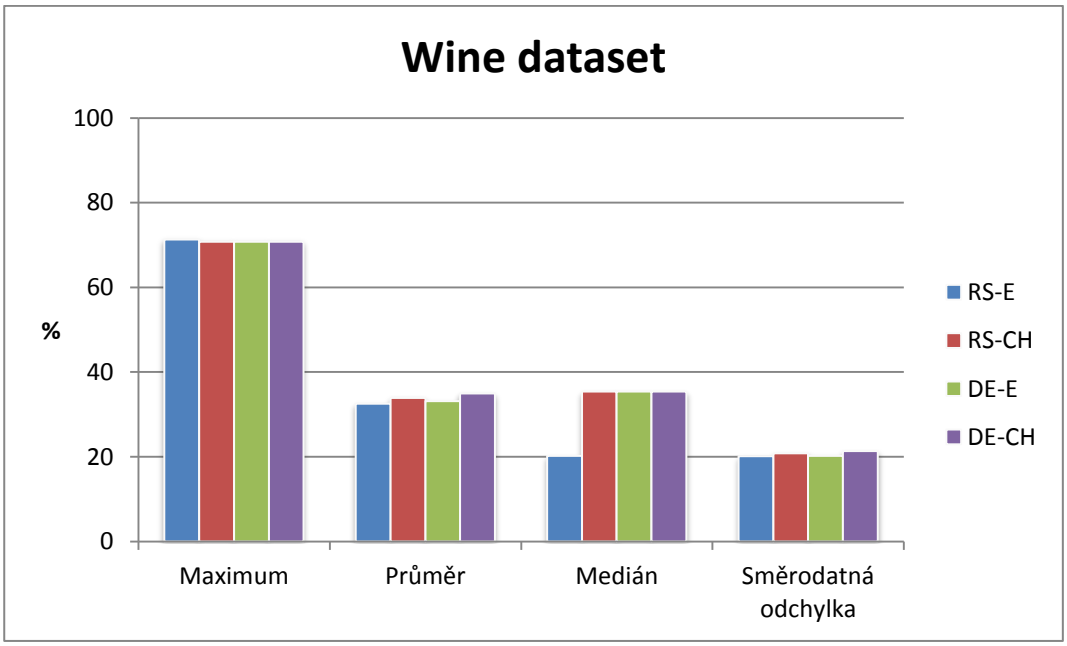
je méně dimenzionální a rozsahy hodnot jednotlivých atributů jsou přibližně stejné, není příliš ovlivněn normalizací. Naopak clustering na Wine datasetu, který je vícedimenzionální a především jsou rozsahy hodnot atributů velmi odlišné, dává výrazně lepší maximální výsledky (průměrný rozdíl 15,31% ve prospěch normalizovaných dat). Nicméně průměrné hodnoty, hodnoty mediánu a směrodatná odchylka se prakticky nemění. Opět se ukázalo, že vícedimenzionální Wine dataset je, i po zavedení penalizace, obtížnější pro cluster analýzu.

Tab. 6. Výsledky testů clustering algoritmů – s penalizací účelové funkce.

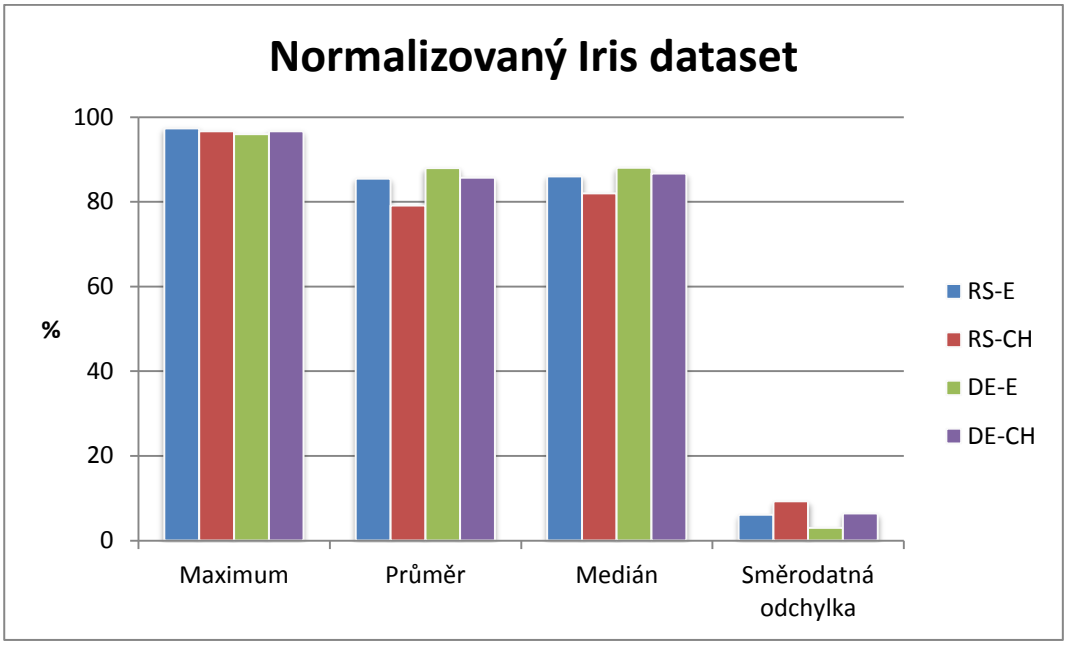
		Iris dataset				**Wine dataset**			
Norm.	**Alg.**	**Max [%]**	**Avg [%]**	**Median [%]**	**STD [%]**	**Max [%]**	**Avg [%]**	**Median [%]**	**STD [%]**
Ano	RS-E	97,33	85,43	86,00	6,07	87,08	32,24	32,58	18,48
	RS-CH	96,67	79,08	82,00	9,27	83,15	33,57	33,71	14,21
	DE-E	96,00	87,94	88,00	3,03	92,70	33,54	33,71	21,46
	DE-CH	96,67	85,72	86,67	6,37	82,02	32,86	33,15	18,63
Ne	RS-E	97,33	87,00	88,00	4,95	71,35	32,57	20,22	20,17
	RS-CH	98,67	83,72	86,00	8,35	70,79	33,89	35,39	20,85
	DE-E	96,00	89,05	89,33	2,44	70,79	33,12	35,39	20,22
	DE-CH	96,67	88,63	88,67	3,81	70,79	34,99	35,39	21,34



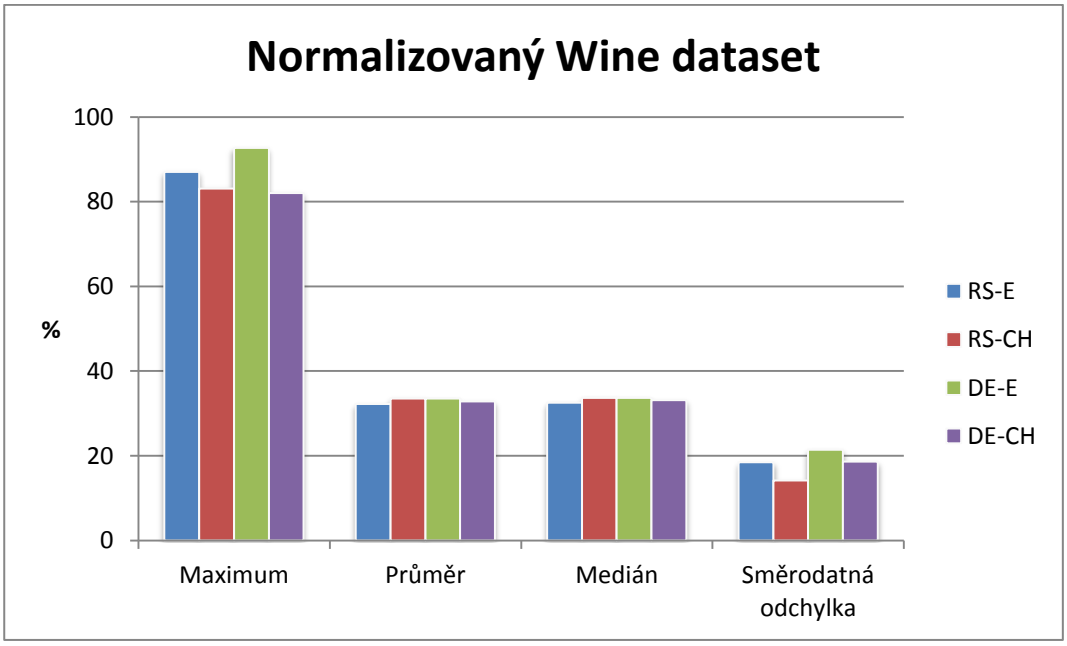
Obr. 38. Porovnání výsledků RS a DE algoritmu na nenormalizovaném Iris datasetu s penalizací účelové funkce.



Obr. 39. Porovnání výsledků RS a DE algoritmu na nenormalizovaném Wine datasetu s penalizací účelové funkce.



Obr. 40. Porovnání výsledků RS a DE algoritmu na normalizovaném Iris datasetu s penalizací účelové funkce.



Obr. 41. Porovnání výsledků RS a DE algoritmu na normalizovaném Wine datasetu s penalizací účelové funkce.

6.4 Analýza všech výsledků

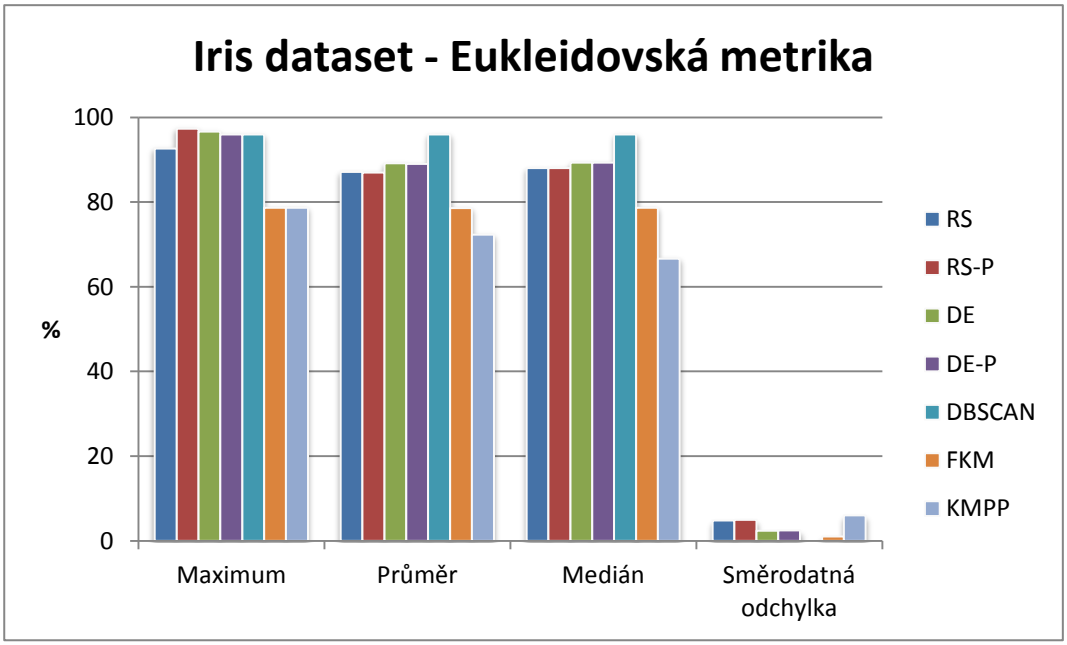
V této části jsou porovnány výsledky jednotlivých algoritmů dohromady. Výsledky jsou rozděleny podle toho, zda byl dataset normalizován a dále podle metriky. Označení

algoritmů v grafech je stejné, jako v předchozích případech, algoritmy RS a DE s penalizací jsou označeny RS-P a DE-P.

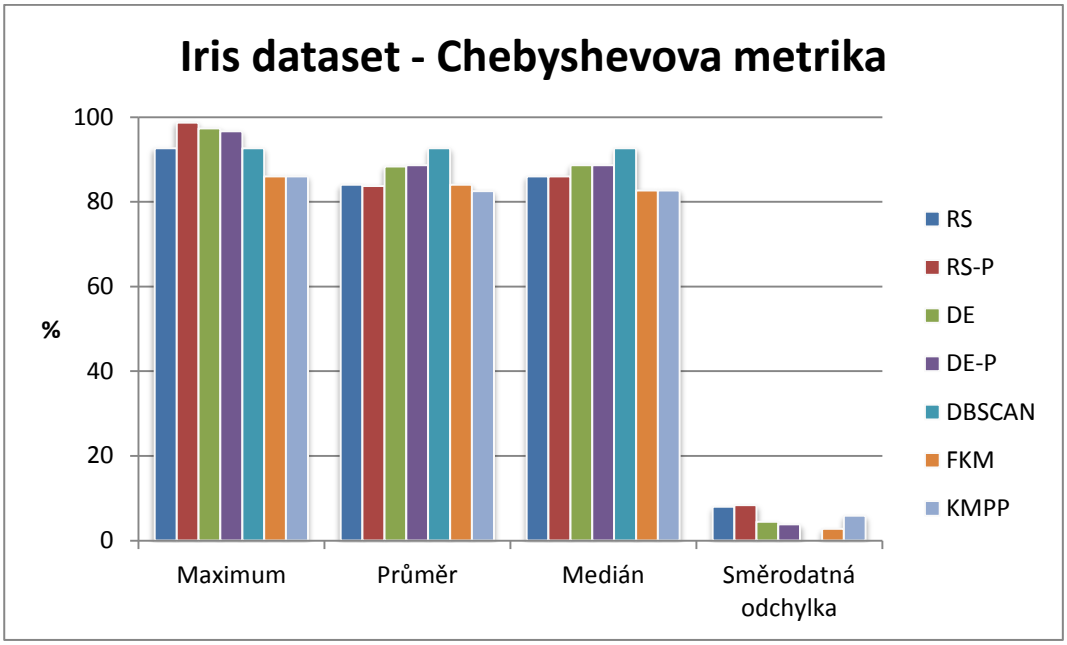
Pricip funkce DBSCAN algoritmu zajišťuje, že po každém spuštění vrátí stejný výsledek, proto jsou hodnoty průměrné úspěšnosti clusteringu a mediánu vysoko nad ostatními algoritmy a směrodatná odchylka je nulová.

Dále jsou v této části porovnány (Obr. 50 až Obr. 53) výsledky ze všech testů a jsou vybrány nejvhodnější algoritmy pro oba datasety. Použitá metrika je označena příponou u algoritmu (-E a -CH), stejně tak normalizace datasetu (-N) a penalizace účelové funkce (-P), takže algoritmus diferenciální evoluce s Eukleidovskou metrikou, puštěný na normalizovaném datasetu s využitím penalizace je označen DE-ENP.

6.4.1 Nenormalizované datasety

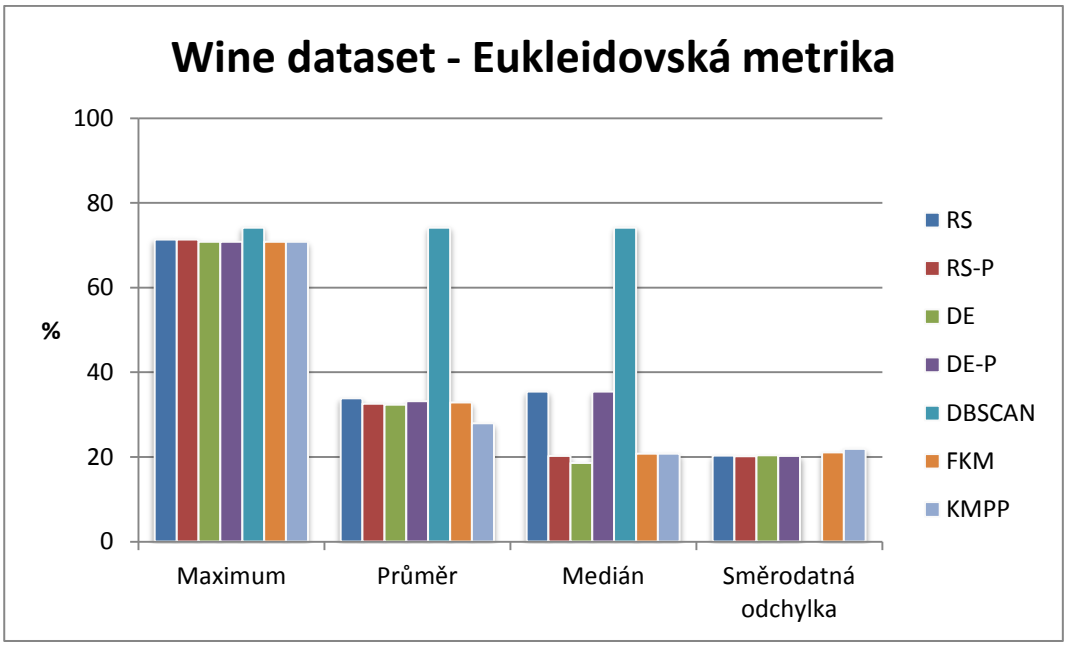


Obr. 42. Graf porovnání statistických vlastností algoritmů na Iris datasetu s Eukleidovskou metrikou.

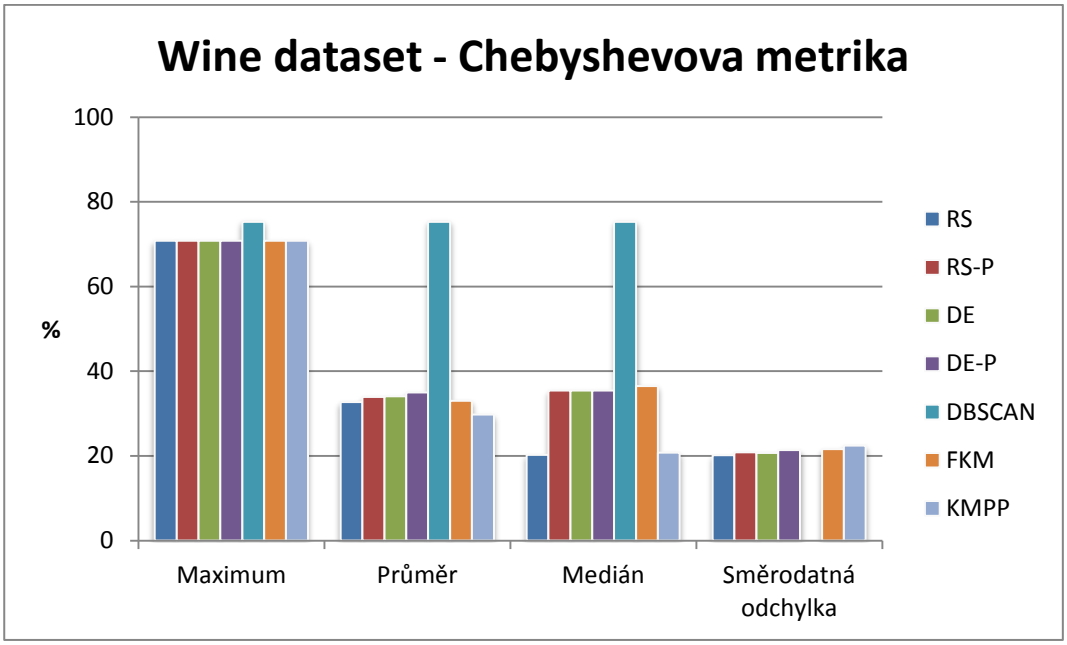


Obr. 43. Graf porovnání statistických vlastností algoritmů na Iris datasetu s Chebyshevovou metrikou.

Z grafů v obrázcích (Obr. 42, Obr. 43) je patrné, že na řešení problému klasifikace nenormalizovaného Iris datasetu je vhodné použít jeden z algoritmů – RS, DE, DBSCAN. Algoritmus DBSCAN poskytuje pokaždé stejný výsledek a proto je vhodný, pokud je potřeba výsledek získat dostatečně rychle. Nicméně algoritmy RS a DE dokáží poskytnout lepší maximální výsledek. Na použité metrice příliš nezáleží.



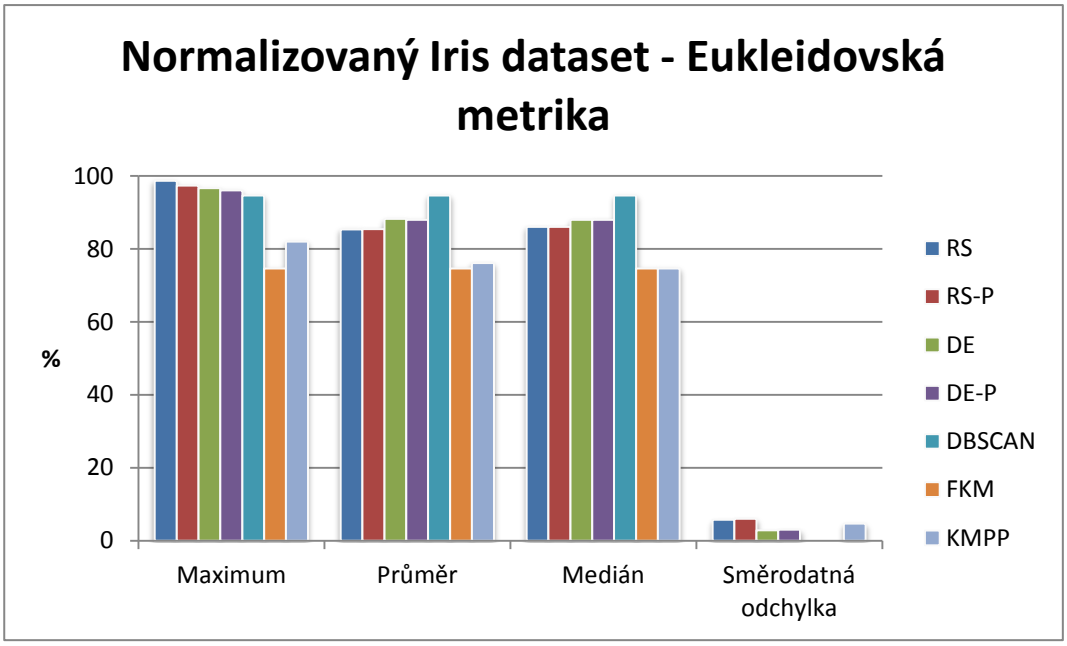
Obr. 44. Graf porovnání statistických vlastností algoritmů na Wine datasetu s Eukleidovskou metrikou.



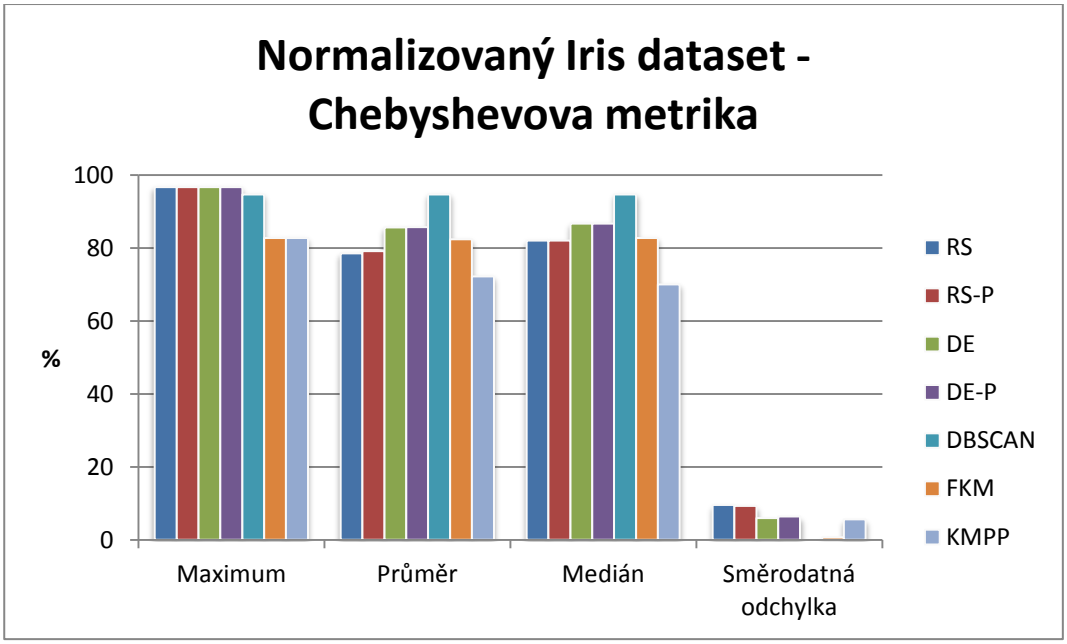
Obr. 45. Graf porovnání statistických vlastností algoritmů na Wine datasetu s Chebyshevovou metrikou.

Z grafů v obrázcích (Obr. 44, Obr. 45) je patrné, že zvolená metrika nemá příliš velký vliv na výsledek clusteringu jednotlivých algoritmů a že jednotlivé algoritmy poskytují přibližně stejnou kvalitu řešení s vyjímkou DBSCAN algoritmu, který dosahuje úspěšnosti 75,28%.

6.4.2 Normalizované datasety

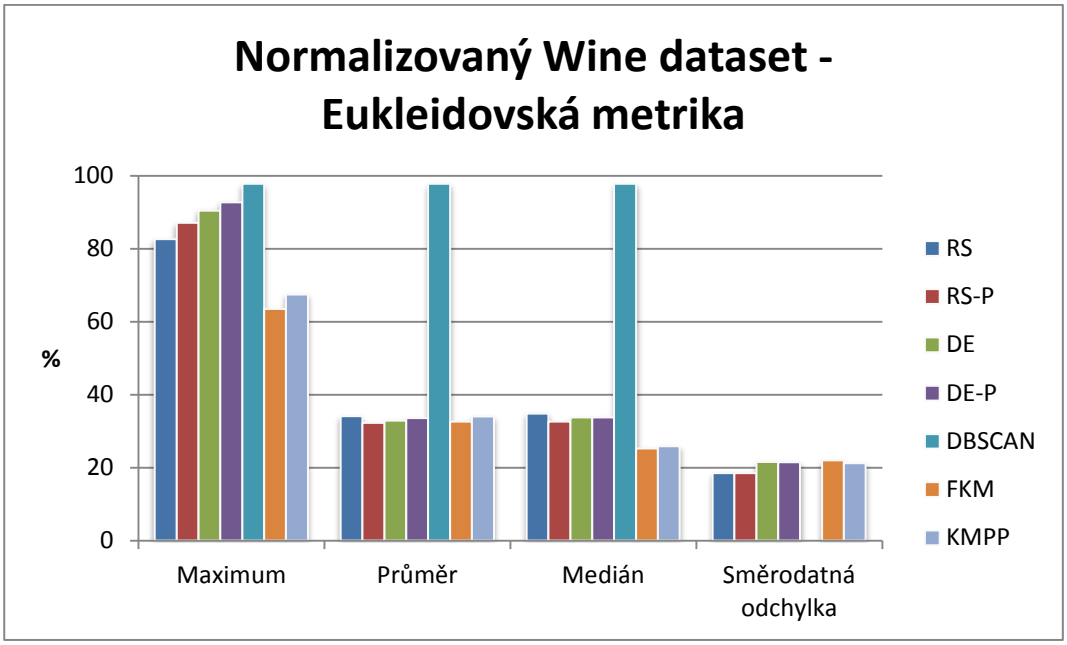


Obr. 46. Graf porovnání statistických vlastností algoritmů na normalizovaném Iris datasetu s Eukleidovskou metrikou.

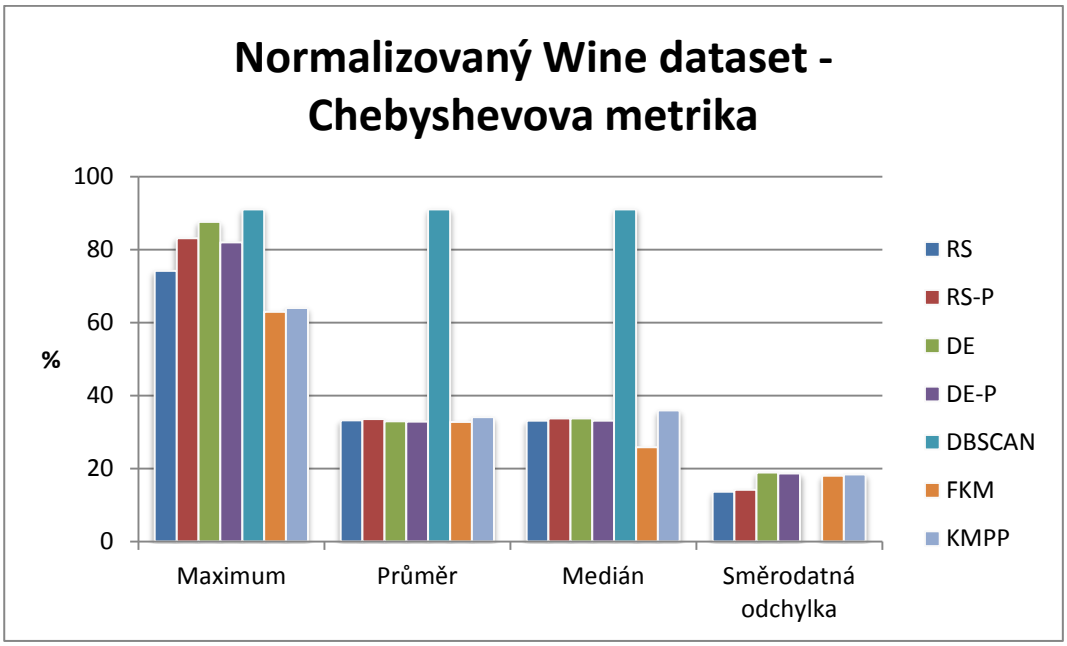


Obr. 47. Graf porovnání statistických vlastností algoritmů na normalizovaném Iris datasetu s Chebyshevovou metrikou.

Na grafech v obrázcích (Obr. 46, Obr. 47) je patrné, že na Iris datasetu dobrého maximálního výsledku dosahují algoritmy RS, DE a DBSCAN a na zvolené metrice opět nezávisí. Penalizace účelové funkce nepřinesla žádné výraznější zlepšení.



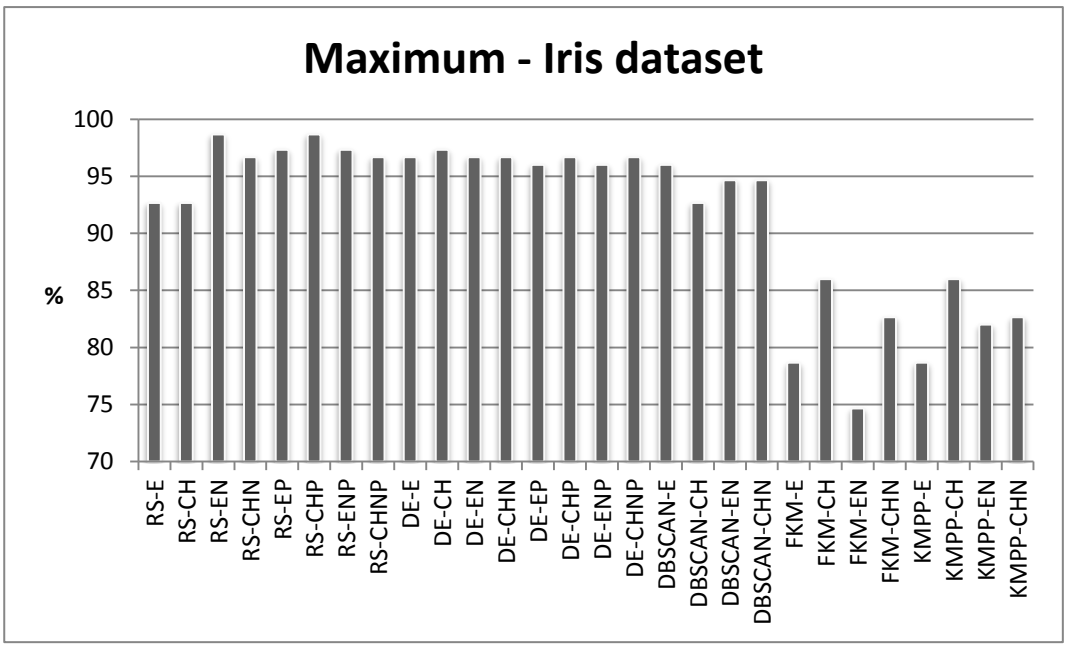
Obr. 48. Graf porovnání statistických vlastností algoritmů na normalizovaném Wine datasetu s Eukleidovskou metrikou.



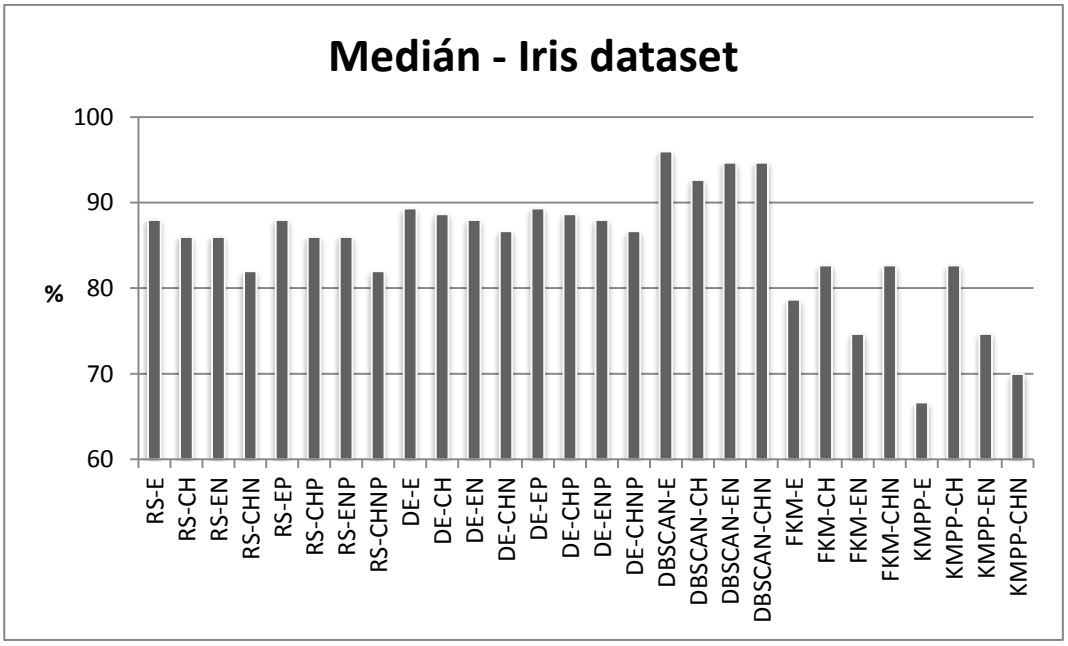
Obr. 49. Graf porovnání statistických vlastností algoritmů na normalizovaném Wine datasetu s Chebyshevovou metrikou.

V grafech na obrázcích (Obr. 48, Obr. 49) je zaznamenán výsledek testu na normalizovaném Wine datasetu. V porovnání s nenormalizovaným datasetem jsou výsledky odlišné a použitá metrika již hraje roli u maximální úspěšnosti algoritmů.

6.4.3 Komplexní porovnání výsledků na jednotlivých datasetech

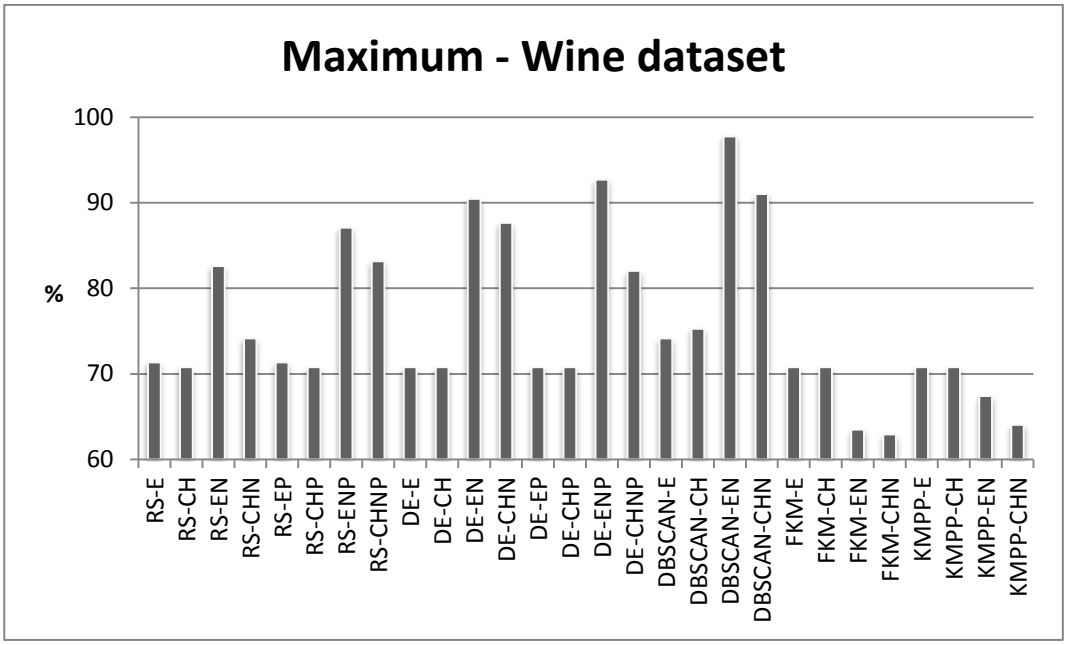


Obr. 50. Komplexní porovnání maximální úspěšnosti algoritmů na Iris datasetu.

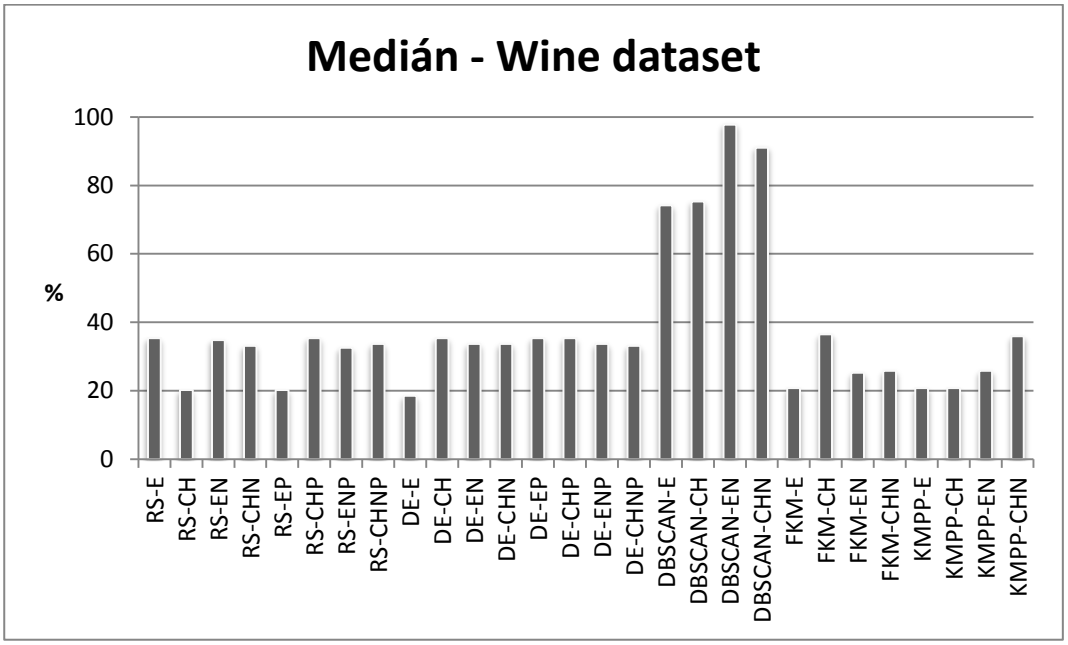


Obr. 51. Komplexní porovnání hodnot mediánu algoritmů na Iris datasetu.

Z grafů v obrázcích (Obr. 50, Obr. 51) lze vypozorovat, že algoritmy RS, DE a DBSCAN dosahují nejvyšších maximálních výsledků a dosažené výsledky nejsou příliš závislé ani na použité metrice, ani na normalizaci Iris datasetu. V mediánových hodnotách vyníká DBSCAN algoritmus, který pokaždé poskytuje stejný – maximální výsledek.



Obr. 52. Komplexní porovnání maximální úspěšnosti algoritmů na Wine datasetu.



Obr. 53. Komplexní porovnání hodnot mediánu algoritmů na Wine datasetu.

Z grafu na obrázku (Obr. 52) je patrné, že použití normalizace datasetu je prospěšné pro většinu použitých algoritmů. To je způsobeno tím, že Wine dataset má velmi odlišné rozsahy jednotlivých atributů, takže jejich normalizací dochází k usnadnění prohledávání prostoru možných řešení. Algoritmus DBSCAN na normalizovaném Wine datasetu s využitím Eukleidovské metriky poskytuje nejlepší řešení klasifikačního problému.

Porovnáním mediánových hodnot na jednotlivých datasetech v obrázcích (Obr. 51, Obr. 53) lze vysledovat, že řešení klasifikačního problému na vícedimenzionálním Wine

datasetu je pro algoritmy náročnější. Vyjímkou je DBSCAN algoritmus s použitím normalizovaného Wine datasetu a Eukleidovské metriky.

6.5 Shrnutí výsledků

Ačkoliv již bylo výše zmíněno, že pro klasifikaci datasetů Iris a Wine je vhodnější využívat přímo klasifikační techniky, testy prokázaly, že je možné s poměrně vysokou přesností využít i metody clusteringové. Níže v této části jsou uvedeny konkrétní hodnoty a hodnocení na jednotlivých datasetech. Obecně lze říci, že z klasických algoritmů (DBSCAN, FKM, KMPP) je nejvhodnější algoritmus DBSCAN a zbylé dva jsou překonány.

Centroidový přístup algoritmů RS a DE je problematický především, pokud se v datasetu vyskytují třídy lineárně neseparovatelné, protože zvolená účelová funkce na takovém datasetu nemusí být přímo úměrná kvalitě klasifikace. Dalším problémem účelové funkce je její plochost, která dělá problémy evolučním algoritmům, které v takovém případě degradují na algoritmus náhodného prohledávání. Oba zmíněné problémy jsou ve výsledcích patrné – algoritmus náhodného prohledávání, který byl původně zařazen pouze pro porovnání, v některých případech překonává více sofistikované algoritmy nebo se jim velmi blíží.

Normalizace datasetu přinesla podstatnější zlepšení výsledků pouze u Wine datasetu, kde jsou větší rozdíly v rozsazích hodnot jednotlivých atributů. Zavedení penalizace do účelové funkce pro algoritmy RS a DE na první pohled nepřineslo zvýšení úspěšnosti těchto algoritmů, ale u algoritmu diferenciální evoluce způsobuje zrychlení konvergence ke globálnímu extrému, protože řešení, která jsou penalizována dále nefigurují v generacích a tím pádem poskytují prostor pro řešení kvalitnější.

6.5.1 Iris dataset

Iris dataset, který je méně dimenzionální a jednotlivé rozsahy atributů se příliš neliší je pro clusteringové algoritmy jednodušší, jak je patrné z dosažených výsledků. Maximální dosažené výsledky jsou uvedeny pouze pro představu o nejúspěšnější klasifikaci, ale statisticky se může jednat pouze o odlehlé extrémy, proto jsou uvedeny i hodnoty průměrné a hodnoty mediánu.

Maximální úspěšnost klasifikace

Nejlepší klasifikace datasetu (98,67% - 148 ze 150 záznamů) dosáhl algoritmus RS a to hned ve dvou konfiguracích - normalizovaný dataset/Eukleidovská metrika/bez penalizace a nenormalizovaný dataset/Chebyshevova metrika/s penalizací.

Dále za zmínku stojí algoritmus DE (97,33% - 146 ze 150 záznamů) v konfiguraci – nenormalizovaný dataset/Chebyshevova metrika/bez penalizace a algoritmus DBSCAN (96% - 144 ze 150 záznamů) v konfiguraci – nenormalizovaný dataset/Eukleidovská metrika.

Průměrná úspěšnost klasifikace

Nejvyšší průměrné hodnoty klasifikace dosahuje algoritmus DBSCAN (97,33%) v konfiguraci – nenormalizovaný dataset/Eukleidovská metrika. Tato hodnota je stejná, jako hodnota maximální, protože algoritmus DBSCAN díky své funkci poskytuje pokaždé stejný výsledek.

Dále stojí za zmínku algoritmus DE (89,17%) v konfiguraci – nenormalizovaný dataset/Eukleidovská metrika/bez penalizace.

Mediánová úspěšnost klasifikace

Stejně jako u průměrné hodnoty nejvyšší mediánovou hodnotu poskytuje algoritmus DBSCAN (97,33%), důvod je stejný jako v případě průměru.

Za zmínku stojí i algoritmus DE (89,33%) ve dvou konfiguracích – nenormalizovaný dataset/Eukleidovská metrika/bez penalizace a nenormalizovaný dataset/Eukleidovská metrika/s penalizací.

6.5.2 Wine dataset

Vícedimenzionální Wine dataset pro clusteringové algoritmy znamená větší potíže a proto především hodnoty průměrné klasifikace a mediánu jsou o poznání nižší než u Iris datasetu. Rozsahy hodnot jednotlivých atributů o Wine datasetu jsou velmi odlišné a proto jejich normalizace do rozsahu <0, 1> přinesla lepší výsledky.

Maximální úspěšnost klasifikace

Nejlepší klasifikace datasetu bylo dosaženo algoritmem DBSCAN (97,75% - 174 ze 178 záznamů) v konfiguraci – normalizovaný dataset/Eukleidovská metrika. Blíže k tomuto výsledku se dostal pouze algoritmus DE (92,70% - 165 ze 178 záznamů) v konfiguraci – normalizovaný dataset/Eukleidovská metrika/s penalizací.

Průměrná úspěšnost klasifikace

Nejvyšší průměrná hodnota je opět u algoritmu DBSCAN (97,75%), který tuto hodnotu dosahuje při každém běhu. Ostatní algoritmy se pohybují okolo hranice 30%.

Mediánová úspěšnost klasifikace

Podobná situace jako u průměrné hodnoty nastává i u hodnoty mediánu, algoritmus DBSCAN (97,75%), zbylé algoritmy maximálně 36%.

ZÁVĚR

V první části této práce byl představen datamining jako proces pro získávání informací z dat, byl proveden detailní rozbor možností a schopností dataminingových algoritmů, stejně jako možnosti využití v praxi. Byly uvedeny jednotlivé druhy dat, na kterých jsou dataminingové algoritmy schopny pracovat a také byly uvedeny konkrétní způsoby přípravy dat pro tyto účely. Preprocessing dat představuje nedílnou součást procesu získávání informací z dat a i proto je mu v této práci věnován dostatečný prostor.

Druhá část práce představuje dva z nejpoužívanějších datasetů pro testování úspěšnosti klasifikace dat algoritmů, dataset Iris obsahující údaje o květech kosatců a dataset Wine obsahující údaje o chemickém složení kultivarů vín. Nejprve jsou porovnány výsledky z publikovaných prací zabývajících se klasifikací těchto datasetů, dále jsou implementovány algoritmy náhodného prohledávání (heuristika), diferenciální evoluce (evoluční algoritmus), DBSCAN (algoritmus založený na hustotě prvků v prostoru), FKM (fuzzy logika) a KMPP (klasický přístup ke clusteringu). Výsledky klasifikace těmito algoritmy jsou detailně popsány a rozebrány.

V rámci práce byl otestován způsob předzpracování dat, který jednotlivé atributy dat normalizuje do rozsahu <0, 1>, čímž odstraňuje rozdíly v rozsazích. Tato technika přinesla výrazné zvýšení účinnosti klasifikace na Wine datasetu, čímž se potvrdila důležitost preprocessingové části datamining procesu.

Dále bylo představeno penalizování hodnoty účelové funkce, které sice nepřineslo výrazné zlepšení v dosažené úrovni klasifikace, ale zvýšilo rychlost konvergence algoritmu diferenciální evoluce bez snížení spolehlivosti. Také byl otestován vliv dvou použitých metrik na výslednou úspěšnost klasifikace.

V práci je ukázáno, že pro klasifikační úlohy lze s úspěchem použít algoritmy clusteringové využívající evoluční techniky a hustotu prvků v prostoru. Především algoritmus DBSCAN, který nevyužívá vzdálenosti bodů od centroidů clusterů, prokázal svou kvalitu při klasifikaci dat ze tříd, které jsou lineárně neseparovatelné.

Pokračování této práce by mohlo být v testování více druhů metrik u již zmíněných algoritmů, nalezení lepší účelové funkce, než je centroidová vzdálenost, hybridizaci umělých neuronových sítí s fuzzy logikou a evolučními technologiemi nebo hledání

nového přístupu ke clusteringu. Záměr autora je věnovat se těmto činnostem v postgraduálním studiu.

CONCLUSION

Datamining was introduced as a process for knowledge discovery in the first part of this thesis. There is also a detailed analysis of possibilities of datamining algorithms as well as their practical usage. Preprocessing is a crucial part of knowledge discovery and that is why it is described with data attribute types and their visualization.

Two of the most popular datasets for classification are introduced in the second part of this thesis, Iris dataset with information about Iris petals and sepals and Wine dataset with information about chemical analysis of three wine cultivars. There is a comparison between results from different papers describing different datamining algorithms. Next, there is a comparison between results of implemented algorithms including: Random search (heuristic algorithm), differential evolution (evolution algorithm), DBSCAN (density based algorithm), FKM (fuzzy logic) and KMPP (classical clustering approach).

Normalization of datasets as a part of preprocessing was implemented and tested which showed significant improvement of classification results on Wine dataset and proved the importance of preprocessing phase of datamining process.

The penalization of cost value function for random search and differential evolution algorithms was introduced. This did not bring a significant improvement in classification, but it speed up the convergence of differential evolution algorithm without affecting its reliability. The effect of used metric on classification was tested as well.

This thesis shows that it is possible to use clustering algorithms for classification problems with good results, especially evolution techniques and density based algorithms. DBSCAN algorithm, which does not use a distance between cluster centroid and data points, is a very powerful algorithm for classification of data which are not linearly separable.

Continuation of this thesis could be in testing of more metric types on centroid based algorithms, finding a better cost value function, hybridization of artificial neural networks with fuzzy logic and evolution techniques or searching for a new approach to data clustering. Author’s intention is to focus on these goals during his postgraduate.

SEZNAM POUŽITÉ LITERATURY

[1] LAROSE, Daniel T. *Discovering knowledge in data: an introduction to data mining*. Hoboken, N.J.: Wiley-Interscience, c2005, xv, 222 s. ISBN 978-0-471-66657-8.

[2] HAN, Jiawei, Micheline KAMBER a Jian PEI. *Data mining: concepts and techniques*. 3rd ed. Waltham: Morgan Kaufmann, c2012, xxxv, 703 s. Morgan Kaufmann series in data management systems. ISBN 978-0-12-381479-1.

[3] HAN, Jiawei, Micheline KAMBER a Jian PEI. *Data Mining: Southeast Asia Edition: Concepts and Techniques*. 2. vyd. Morgan Kaufmann, 2006, 800 s. ISBN 978-0-08-047558-5.

[4] MANYIKA, James, Michael CHUI, Brad BROWN, Jacques BUGHIN, Richard DOBBS, Charles ROXBURGH a Angela Hung BYERS. Big data: the next frontier for innovation, competition, and prductivity. [online]. 2014 [cit. 2015-03-03]. Dostupné z: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation

[5] ISSENBERG, Sasha. How President Obama's campaign used big data to rally individual voters. [online]. 2012 [cit. 2015-03-03]. Dostupné z: http://www.technologyreview.com/featuredstory/509026/how-obamas-team-used-big-data-to-rally-voters/

[6] FABRIS, Peter. Advanced Navigation. *CIO Magazine*. 1998. Dostupné z: http://www.cio.com/archive/051598_mining.html

[7] CHAPMAN, Peter, Julian CLINTON, Randy KERBER, Thomas KHABAZA, Thomas REINART, Colin SHEARER a Rudiger WIRTH. *CRISP-DM Step-by-Step Data Mining Guide*. 2000. Dostupné z: www.the-modeling-agency.com/crisp-dm.pdf

[8] KANTARDZIC, Mehmed. *Data mining: concepts, models, methods, and algorithms*. 2nd ed. Hoboken, N.J.: IEEE Press, c2011, xvii, 534 p. ISBN 978-1-11-802913-8

[9] Newsmap. In: *UMD Department of Computer Science* [online]. [cit. 2015-03-19]. Dostupné z: http://www.cs.umd.edu/class/spring2005/cmsc838s/viz4all/ss/newsmap.png

[10] LICHMAN, M. UNIVERSITY OF CALIFORNIA, Irvine, School of Information and Computer Sciences. *UCI Machine Learning Repository* [online]. 2013 [cit. 2015-04-02]. Dostupné z: http://archive.ics.uci.edu/ml

[11] SIGNA. *The Species Iris Group of North America* [online]. 2015 [cit. 2015-04-02]. Dostupné z: http://www.signa.org/index.pl?Database

[12] INDON. Anderson's Iris data set. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2007 [cit. 2015-04-02]. Dostupné z: http://commons.wikimedia.org/wiki/File:Anderson%27s_Iris_data_set.png

[13] KRUPKA, Jiri a Pavel JIRAVA. Rough-fuzzy Classifier Modeling Using Data Repository Sets. In: *Procedia Computer Science*. Elsevier B.V., 2014, s. 701-709. ISSN 18770509. DOI: 10.1016/j.procs.2014.08.152. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S187705091401117X

[14] Fuzzy Sets and Pattern Recognition. PRINCETON UNIVERSITY. *Computer Science Department* [online]. 2007 [cit. 2015-04-07]. Dostupné z: http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm

[15] KOMINKOVA OPLATKOVA, Zuzana, Roman SENKERIK, Roman ŠENKEŘÍK. Iris Data Classification By Means Of Pseudo Neural Networks Based On Evolutionary Symbolic Regression. In: EDITED BY: WEBJØRN REKDALSBAKKEN, Robin T. *ECMS 2013 Proceedings edited by: Webjorn Rekdalsbakken, Robin T. Bye, Houxiang Zhang: May 27th - May 30th, 2013, Ålesund, Norway*. ECMS, 2013-5-27, s. 355-360. ISBN 9780956494467. DOI: 10.7148/2013-0355. Dostupné z: http://www.scs-europe.net/dlib/2013/2013-0355.htm

[16] ZELINKA, Ivan, Donald DAVENDRA, Roman SENKERIK, Roman JASEK a Zuzana OPLATKOVÁ. Analytical Programming - a Novel Approach for Evolutionary Synthesis of Symbolic Structures. In: *Evolutionary Algorithms*. InTech, 2011-04-26. ISBN 978-953-307-171-8. DOI: 10.5772/16166. Dostupné z: http://www.intechopen.com/books/evolutionary-algorithms/analytical-programming-a-novel-approach-for-evolutionary-synthesis-of-symbolic-structures

[17] KUO, R. J., S. S. CHEN, W. C. CHENG a C. Y. TSAI. Integration of artificial immune network and K-means for cluster analysis. In: *Knowledge and Information*

Systems. 2014, s. 541-557. ISSN 0219-1377. DOI: 10.1007/s10115-013-0649-3. Dostupné z: http://link.springer.com/10.1007/s10115-013-0649-3

[18] HATAMLOU, Abdolreza a Masoumeh HATAMLOU. PSOHS: an efficient two-stage approach for data clustering. In: *Memetic Computing*. 2013, s. 155-161. ISSN 1865-9284. DOI: 10.1007/s12293-013-0110-x. Dostupné z: http://link.springer.com/10.1007/s12293-013-0110-x

[19] CHING-YI CHEN a FUN YE. Particle swarm optimization algorithm and its application to clustering analysis. In: *IEEE International Conference on Networking, Sensing and Control, 2004*. IEEE, 2004, s. 789-794. ISBN 0-7803-8193-9. DOI: 10.1109/ICNSC.2004.1297047. Dostupné z: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1297047

[20] ESTER, Martin, Hans-Peter KRIEGEL, Jörg SANDER a Xiaowei XU. *A Density-Based Algorithm for Discovering Clusters: in Large Spatial Databases with Noise*. München, Germany, 1996. Conference Paper. University of Munich.

[21] NOCK, Richard a Frank NIELSEN. On Weighting Clustering. In: *IEEE transactions on pattern analysis and machine intelligence*. New York: IEEE Computer Society, 2006, s. 13. ISSN 0162-8828.

[22] ARTHUR, David a Sergei VASSILVITSKII. *K-means++: The Advantages of Careful Seeding*. PA, USA, 2007. Society for Industrial and Applied Mathematics Philadelphia.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AP	Analytic Programming.
CRISP-DM	The Cross-Industry Standard Practice for Data Mining.
DBSCAN	Density-Based Spatial Clustering of Applications with Noise.
DE	Differential Evolution.
DNA	Deoxyribonukleová kyselina.
DSH	Discrete Set Handling.
FKM	Fuzzy K-Means.
FS	Fuzzy Set.
GE	Grammatical Evolution.
GFS	General Functional Set.
GP	Genetic Programming.
HIV	Human Immunodeficiency Virus.
HS	Heuristic Search.
KDD	Knowledge Discovery in Data.
KMPP	K-Means Plus Plus.
MGI	McKinsey Global Institute.
MIT	Massachusetts Institute of Technology.
MSE	Mean-Square quantization Error.
OLAP	Online Analytical Proceessing.
PNN	Pseudo Neural Network.
PSO	Particle Swarm Optimization.
RS	Random Search.
RST	Rough Sets Theory.
SPSS	Statistical Package for the Social Sciences.

SQL	Structured Query Language.
SR	Symbolic Regression.
STD	Standard Deviation.
SVM	Support Vector Machine.
TB	Tera Byte.
UCI	University of California, Irvine.
WWW	World Wide Web.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK