VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

HODNOCENÍ KVALITY VÝSLEDKŮ SHLUKOVÉ ANALÝZY

EVALUATION OF THE CLUSTER ANALYSIS QUALITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **MICHAEL SCHMID**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. IVANA BURGETOVÁ, Ph.D.**
SUPERVISOR

BRNO 2018

Zadání bakalářské práce/20753/2017/xschmi08

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Schmid Michael**
Obor: Informační technologie
Téma: **Hodnocení kvality výsledků shlukové analýzy**
Evaluation of the Cluster Analysis Quality
Kategorie: Data mining

Pokyny:

1. Seznamte se s metodami shlukové analýzy a prostudujte možnosti hodnocení kvality výsledků shlukové analýzy.
2. Navrhněte aplikaci, která pro zadané výsledky shlukové analýzy zhodnotí jejich kvalitu pomocí různých kritérií.
3. Implementujte navrženou aplikaci.
4. Po dohodě s vedoucí vyberte vhodné datové sady pro testování vytvořené aplikace.
5. Aplikaci otestujte a zhodnoťte dosažené výsledky.

Literatura:

- Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Third Edition. Morgan Kaufmann Publishers, 2012, 703 p., ISBN 978-0-12-381479-1
- Aggarwal, Ch. C., Reddy, Ch. K.: Data Clustering Algorithms and Applications, CRC Press, 2014, ISBN 978-1-4665-5821-2

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Burgetová Ivana, Ing., Ph.D.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá shlukovou analýzou, zejména způsoby hodnocení kvality jejích výsledků. Poskytuje teoretický úvod ke shlukové analýze a k metrikám schopným hodnotit její kvalitu. Dokumentuje vývoj aplikace, která je schopna za pomocí zmíněných metrik provádět hodnocení kvality výsledků shlukové analýzy. Podstatná část práce se věnuje experimentování s vytvořenou aplikací, včetně návrhu experimentů a analýzy chování shlukovacích algoritmů a metrik v kombinaci s různými datovými sadami.

Abstract

This bachelor's thesis concerns cluster analysis and possible ways to evaluate the quality of its results. The thesis contains theoretical introduction to cluster analysis and metrics used for evaluation of quality of its results. The thesis also documents development of an application capable of evaluating quality of results of cluster analysis using mentioned metrics. Important part of the thesis describes experiments conducted with implemented application, including design of the experiments and analysis of behavior of clustering algorithms and metrics when they are used in combination with various datasets.

Klíčová slova

Shluková analýza, Shlukování, Kvalita shlukové analýzy, Dolování dat, Davies–Bouldinův index, Dunnův separační index, Silhouette index, RMSSTD, K–means, K–medoids, DBSCAN

Keywords

Cluster analysis, Clustering, Cluster analysis quality, Data mining, Davies–Bouldin index, Dunn Separation index, Silhouette index, RMSSTD, K-Means, K-Medoids, DBSCAN

Citace

SCHMID, Michael. *Hodnocení kvality výsledků shlukové analýzy.* Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ivana Burgetová, Ph.D.

Hodnocení kvality výsledků shlukové analýzy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením paní Ing. Ivany Burgetové, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michael Schmid
15. května 2018

Poděkování

Tímto bych rád poděkoval Ing. Ivaně Burgetové, Ph.D. za vedení této bakalářské práce, odborné konzultace k vyvinuté aplikaci a teoretickým základům shlukové analýzy a způsobům jejího hodnocení.

Obsah

Kapitola 1

Úvod

Existuje mnoho způsobů jak provádět shlukovou analýzu. Různé algoritmy, určené pro tuto statistickou metodu, mohou produkovat různé výsledky, jejichž kvalita není vždy ekvivalentní a do shlukování vstupuje také velké množství dalších faktorů, které mohou mít na výsledek shlukové analýzy nemalý vliv.

Shlukování lze použít pro analýzu dat v různých odvětvích lidské činnosti a je tedy žádoucí, aby bylo možno ohodnotit kvalitu výsledků, které shlukování produkuje.

Tato práce se zabývá způsoby hodnocení kvality výsledků algoritmů určených pro shlukování dat. Poskytuje teoretický úvod ke shlukové analýze a hodnocení kvality jejích výsledků. Dokumentuje vývoj a funkčnost aplikace, která je schopna za použití několika různých metrik ohodnotit kvalitu výsledků shlukové analýzy datové sady — zpracovává tedy data, obsahující informace o objektech a jejich atributech a o příslušnosti těchto objektů do shluků — a vyhodnotit nejlepší rozdělení podle každé z těchto metrik.

Část této práce se věnuje také experimentální části projektu – experimentování s výslednou aplikací v kombinaci s různými datovými sadami za účelem porovnávání výsledků jednotlivých metrik pro hodnocení kvality (indexů) a vyhodnocení vhodnosti těchto indexů pro použité datové sady. Obsahuje dokumentaci průběhu experimentů od jejich návrhu přes jejich provedení až po výsledky jednotlivých experimentů, které jsou zhodnoceny a ze získaných informací je vyvozen závěr.

Cílem této bakalářské práce je poskytnout aplikaci schopnou analyzovat výsledky získané shlukovou analýzou. S její pomocí poté otestovat jednotlivé metriky hodnotící kvalitu výsledků shlukové analýzy, jejich fungování v kombinaci s různými datovými sadami, typy atributů a shlukovacími algoritmy.

Kapitola 2

Shluková analýza

Shluková analýza je statistická metoda používaná pro dolování dat. Tato metoda nachází využití v mnoha oborech lidské činnosti, jako například v bioinformatice, marketingu nebo v medicíně.[11]

Shluková analýza je prováděna nad množinou objektů za účelem jejich rozdělení do shluků obsahujících vzájemně si podobné objekty. Provádí tedy klasifikaci dat za účelem nalezení podobností v těchto datech.

Algoritmy používané pro shlukovou analýzu fungují na principu rozdělování datových objektů do shluků na základě podobnosti těchto objektů tak, aby si objekty patřící do jednoho shluku byly co nejpodobnější a zároveň se od sebe objekty náležící různým shlukům co nejvíce lišily.[11]

Pro provádění shlukové analýzy existuje množství algoritmů, z nichž některé byly použity pro shlukovou analýzu datových sad vybraných pro experimentování s výslednou aplikací pro hodnocení kvality shlukové analýzy. Tyto použité shlukovací algoritmy budou popsány dále v této kapitole. Část této kapitoly také pojednává o způsobech standardizace vstupních dat a možných způsobech zpracování dat obsahujících atributy různých typů.

2.1 Algoritmus K-means

Algoritmus K-means je metoda nehiearchického shlukování, která rozděluje objekty do shluků (skupin) podle vzdálenosti objektů od imaginárních těžišť shluků.[11]

Tato metoda funguje na základě přiřazování objektů k centroidům, což jsou body umístěné do prostoru obsahujícího objekty. K centroidům se vždy rozdělí všechny objekty obsažené v datové sadě na základě vzdálenosti. Tímto způsobem se utvoří shluky. Po utvoření shluků se přepočítají souřadnice centroidů, aby se centroidy přesunuly do těžišť takto vzniklých shluků. Po přesunutí centroidů se opět opakuje proces přiřazování objektů k jim nejbližším centroidům a tento postup se neustále opakuje, dokud nedojde k ustálení shluků.[11]

Výsledky tohoto algoritmu mohou být ovlivněny tím, že počáteční pozice centroidů jsou zvoleny náhodně. Je vhodné provádět shlukování pomocí K-means v několika bězích, za účelem minimalizace dopadu počátečního rozdělení na výsledky shlukování.

Metoda K-means zpravidla dobře konverguje. Tento algoritmus byl vybrán pro experimentování s aplikací z toho důvodu, že je potřeba algoritmu specifikovat počet shluků, do kterých má objekty rozdělit. Díky této vlastnosti je možné jednu datovou sadu rozdělit do různého počtu shluků, použít tato rozdělení pro hodnocení kvality a vyhodnotit na základě každé z těchto metrik nejlepší počet shluků.

K-means patří k nejběžnějším a nejčastěji používaným algoritmům[8] pro shlukování dat, což je jeden z důvodů, proč byl tento algoritmus vybrán pro shlukování dat za účelem experimentování s vytvořenou aplikací.

Kvůli svým vlastnostem není tento algoritmus vhodný pro shlukování datových sad s objekty rozmístěnými do specifických tvarů, pokud chceme tyto tvary najít a rozdělit do shluků. Důvodem je to, že K-means tvoří shluky v závislosti na vzdálenosti objektů od centroidů a dá se tedy předpokládat, že bude mít tendenci tvořit shluky kulovitého tvaru.

2.2 Algoritmus K-Medoids

K-Medoids je algoritmus podobný algoritmu K-Means. Také jako vstupní parametr přijímá počet shluků, do kterých se mají objekty rozdělit. Narozdíl od algoritmu K-Means ale nepočítá s fiktivními centroidy, nýbrž místo nich používá objekt z datové sady náležící danému shluku, který je nejblíže jeho středu.[10]

Tento algoritmus je schopný zmenšit dopady výskytu odlehlých hodnot na výsledky shlukování[11], ale je výpočetně náročnější algoritmus než K-Means, proto je vhodné jej používat pro menší datové sady.[10]

2.3 Hierarchické shlukování

Hierarchické shlukování postupně rozděluje nebo slučuje jednotlivé shluky až do splnění zadaného kritéria (kterým může být například počet shluků) nebo do přiřazení každého objektu do jeho vlastního shluku (v případě hierarchického shlukování rozdělováním) či sloučení všech objektů do jednoho shluku (v případě slučování objektů)[10]. Díky tomu vytváří stromovou strukturu zachycující postupné rozdělování/spojování shluků. Lze jej provádět dvěma způsoby[8]:

- Rozdělováním (top-down)
- Slučováním (bottom-up)

Stav rozdělovacího hierarchického shlukování po inicializaci je takový, že veškeré objekty shlukované datové sady patří do jednoho shluku. Poté probíhá iterativně rozdělování do menších shluků dokud algoritmus nedosáhne požadovaného výsledku (například dosáhne požadované úrovně shlukování nebo každý objekt je zařazen do jiného shluku). Je možné tento algoritmus použít v kombinaci s jinými shlukovacími metodami, které provádějí rozdělování v jednotlivých iteracích.[10]

Slučovací hierarchické shlukování má stav po inicializaci opačný oproti rozdělovacímu shlukování, tedy každý objekt náleží svému vlastnímu shluku. Poté dochází ke slučování shluků do chvíle, než algoritmus dosáhne požadované úrovně shlukování nebo do chvíle, kdy všechny objekty patří do jednoho shluku.[10]

2.4 DBSCAN

DBSCAN neboli *Density-Based Spatial Clustering of Applications with Noise* je metoda, která shlukuje objekty na základě hustoty jejich výskytu.[11] Stanovuje tedy hranice shluků v místech, kde hustota objektů není dostatečně velká a shluky vytváří z oblastí v datovém prostoru, které obsahují velkou hustotu objektů.

Tento algoritmus pro svou funkčnost potřebuje dva vstupní argumenty. První z nich je ϵ a udává velikost okolí objektu, ve kterém se vyhledávají další objekty, které je možno přiřadit do stejného shluku jako daný objekt. Druhý z těchto argumentů je přirozené číslo udávající minimální počet objektů v okolí objektu, aby tento objekt mohl být vybrán jako střed shluku. [8] Vzhledem k tomu, že argument ϵ udává velikost prohledávaného okolí, tak na jeho velikosti závisí, jak velkou vzdálenost mezi objekty je schopen algoritmus překonat při rozdělování do shluků. Při větší hodnotě tohoto argumentu má tedy tendenci tvořit menší počet větších shluků.

DBSCAN bere ohled na vzdálenost mezi oblastmi s vysokou hustotou, je tedy schopen najít v datové sadě například obrazce, které nelze najít pomocí shlukovacích metod založených čistě na vzdálenosti mezi objekty.

2.5 Standardizace vstupních dat

Vzhledem ke skutečnosti, že můžeme chtít shlukovat objekty popsané atributy různých typů (nominální, binární, číselné, ordinální), které je zapotřebí zpracovávat dohromady, je nutné vstupní data standardizovat.

Standardizaci atributů je potřeba provádět i v případě, že je objekt popsán atributy, které jsou jednotného typu. Tato standardizace se provádí většinou z důvodu, že je potřeba stanovit vzdálenost klasifikovaných objektů.

2.5.1 Binární atributy

Binární atributy v zásadě není zapotřebí standardizovat. Při výpočtu vzdálenosti mezi objekty se binární atributy pouze porovnávají na shodu. Pokud se shodují, tak neovlivní výslednou vzdálenost objektů. V opačném případě se vzdálenost objektů zvýší o 1.

2.5.2 Nominální atributy

U nominálních atributů, které jsou de facto zobecněním atributů binárních (tedy mohou nabývat konečné množiny hodnot), byla uvažována standardizace více způsoby:

Transformace na binární atributy

První možností je provést transformaci nominálních atributů na atributy binární. To se provádí tím způsobem, že se n-hodnotový nominální atribut převede na n binárních atributů, kde každý z těchto binárních atributů odpovídá jedné z hodnot původního nominálního atributu. Výsledné převedení je tedy de facto bijektivním zobrazením množiny hodnot nominálního atributu do množiny binárních atributů. Následně se jeden z těchto binárních atributů, který odpovídá původní hodnotě binárního atributu, nastaví na hodnotu 1 a všechny ostatní na hodnotu 0. Ovšem tento způsob standardizace nemusí být příliš vhodný pro účely výpočtů shlukové analýzy nebo metrik hodnotících její kvalitu, jelikož v případě nominálního atributu s velkým množstvím přípustných hodnot, je tento atribut transformován na velký počet atributů binárních. Toto může mít za následek mnoho porovnávání při výpočtu vzdáleností objektů a v důsledku této skutečnosti může být negativně ovlivněna doba takového výpočtu

Jednoduché porovnání na shodu

Nominální atributy se pouze porovnávají na shodu při výpočtu vzdálenosti objektů a tuto vzdálenost ovlivňují stejným způsobem jako atributy binární. Tedy při shodě atributů nedojde ke změně výsledné vzdálenosti objektů, při neshodě se tato vzdálenost zvětší o 1. Tento způsob výpočtu je oproti transformaci na binární atributy efektivnější, a proto také byl vybrán pro účely implementace aplikace, které je součástí této bakalářské práce.

2.5.3 Ordinální atributy

U ordinálních atributů je situace vzhledem k předchozím dvěma jmenovaným případům trochu složitější. Pokud uvážíme, že ordinální atributy mají jasně definované pořadí hodnot, kterých mohou nabývat, je potřeba toto pořadí reflektovat i při jejich zpracování. Vzhledem k tomu, že ostatní atributy normalizujeme na interval 0–1, tak jsou tyto ordinální atributy normalizovány na stejný interval. Tedy pokud má ordinální proměnná hodnoty (seřazeno od minimální po maximální) *A, B, C*, tak se těmto hodnotám přiřadí číselné reprezentace A=0, B=0,5, C=1.

2.5.4 Číselné atributy

Číselné atributy je při hodnocení datových sad s různými typy atributů také nutno standardizovat. Představme si situaci, kdy bychom měli datovou sadu obsahující jeden číselný atribut, který dosahuje vysokých hodnot společně s dvaceti binárními atributy. Pokud by byl příliš velký rozsah hodnot číselného atributu mezi objekty této datové sady, mohlo by se stát, že vliv binárních atributů na výsledné indexy by byl zanedbatelný. Z tohoto důvodu se číselné atributy standardizují na interval 0–1. Standardizace může probíhat podle následujícího vzorce[11] (tento způsob je použit také v implementované aplikaci):

$$\frac{|x_1 - x_2|}{max(x) - min(x)} \tag{2.1}$$

kde x_1 je hodnota atributu x prvního objektu, x_2 je hodnota atributu x druhého objektu a $min(x), max(x)$ jsou minimální a maximální hodnota atributu x napříč všemi objekty náležícími této datové sadě.

Standardizace číselných atributů se ovšem neprovádí vždy. Je potřeba uvážit, zdali je tento proces potřebný pro účel výpočtu. Pokud součástí datové sady jsou pouze číselné atributy, někdy není potřeba je standardizovat, ba naopak by tato standardizace mohla výpočet zbytečně znepřesnit. Číselné atributy se zpravidla standardizují pouze u datových sad se smíšeným typem atributů nebo například v případě velkého rozptylu hodnot atributů.

Kapitola 3

Kvalita výsledků shlukové analýzy

Jak jsme si již dříve nastínili, shluková analýza vzhledem k velkému množství existujících shlukovacích algoritmů může produkovat velice různé výsledky ne vždy vyhovující kvality. Naštěstí však existují metriky, pomocí kterých se dá tato kvalita měřit.

Jedním z cílů této práce je také zhodnotit vlastnosti těchto metrik a popřípadě odhalit jejich slabé stránky a nedokonalosti v určitých situacích.

V této kapitole se zaměříme na úvod do problematiky hodnocení kvality výsledků shlukové analýzy a rozbor jednotlivých algoritmů pro výpočet indexů, které jsou schopny tuto kvalitu zhodnotit. Tento rozbor se bude týkat zejména indexů zakomponovaných do výsledné aplikace.

3.1 Úvod do problematiky

Většina algoritmů používaných pro výpočet indexů reflektujících kvalitu shlukování, se zakládá na dvou základních faktorech, které by kvalitní nashlukování mělo splňovat[10]. Tyto faktory jsou

- kompaktnost shluků
- separabilita shluků

Kompaktnost je vlastnost shluků, která určuje míru podobnosti objektů uvnitř jednotlivých shluků, kdežto separabilita (nebo také oddělitelnost) shluků značí míru rozdílnosti, tedy nakolik se jednotlivé shluky mezi sebou liší.

Pro vývoj aplikace byly vybrány 4 standardní algoritmy hodnotící kvalitu. Těmito algoritmy jsou:

- Davies-Bouldin index
- Dunn separation index
- Silhouette index
- Root mean square standard deviation

Kromě těchto standardních algoritmů aplikace navíc umožňuje uživateli zadat referenční rozdělení do shluků na jehož základě spočítá pro jednotlivá nashlukování, jaká část objektů (údaj je značen desetinným číslem na intervalu $\langle 0, 1 \rangle$) byla přiřazena do shluků stejně,

jako v referenčním rozdělení. Všechny tyto způsoby hodnocení shlukování budou teoreticky popsány a o jejich implementaci do výsledné aplikace bude pojednávat kapitola 4.

Pro účely vývoje aplikace byl také uvažován index *Clustering Validation index based on Nearest Neighbors* (zkráceně *CVNN*), ale od jeho implementace bylo nakonec upuštěno z důvodů přílišné časové náročnosti výpočtu.

Vysvětlení symbolů použitých v níže uvedených vzorcích: P je počet atributů objektů patřících do analyzované datové sady; NC je počet shluků; C_i je i-tý shluk; n_i je počet objektů v C_i; c_ii je střed shluku C_i; $d(x, y)$ je vzdálenost mezi x a y.

3.2 Davies-Bouldin index

Davies-Bouldinův index je metrika, která hodnotí kvalitu na základě poměru součtu vzdáleností uvnitř jednotlivých shluků (v ideálním případě je tento součet co nejmenší $\implies$ kompaktnost shluků je co možná největší) a vzdálenosti mezi těmito shluky (ta je v ideálním případě co největší $\implies$ toto signalizuje velkou míru separace shluků). Tento index, jak vyplývá z předchozího tvrzení, je v optimálním případě minimální.[8]

$$\frac{1}{NC}\Sigma_i max_{j,j\neq i} \left\{ \left[\frac{1}{n_i}\Sigma_{x\in C_i} d(x, c_i) + \frac{1}{n_j}\Sigma_{x\in C_j} d(x, c_j) \right] / d(c_i, c_j) \right\} \tag{3.1}$$

Vzorec pro výpočet Davies-Bouldin indexu [8]

3.3 Dunn separation index

Dunnův separační index, jak už název napovídá, je index, který hodnocení kvality zakládá na vzdálenosti mezi jednotlivými shluky a na průměru těchto shluků. Výsledný index se tedy spočítá jako podíl nejmenší vzdálenosti mezi shluky a maximálního průměru shluku. Vyšší hodnota indexu značí vyšší oddělitelnost objektů a tedy lepší rozdělení do shluků.[8]

$$min_i \left\{ min_j \left(\frac{min_{x\in C_i, y\in C_j} d(x, y)}{max_k \left\{ max_{x,y\in C_k} d(x, y) \right\}} \right) \right\} \tag{3.2}$$

Vzorec pro výpočet Dunn Separation indexu [8]

3.4 Silhouette index

Silhouette index hodnotí kvalitu shlukování podle vzdálenosti mezi shluky (hledáním nejbližšího objektu patřícího do jiného shluku) a podle kompaktnosti shluků, tedy vzdálenosti objektů patřících do jednoho shluku. Výsledek tohoto algoritmu je z intervalu od -1 do 1. Ideální shlukování je takové, pro které je tento index maximální.[8]

$$\begin{gathered} \frac{1}{NC}\Sigma_i \left\{ \frac{1}{n_i}\Sigma_{x\in C_i} \frac{b(x) - a(x)}{max[b(x), a(x)]} \right\} \\ a(x) = \frac{1}{n_i - 1}\Sigma_{y\in C_i, y\neq x} d(x, y), b(x) = min_{j,j\neq i}[\frac{1}{n_j}\Sigma_{y\in C_j} d(x, y)] \end{gathered} \tag{3.3}$$

Vzorec pro výpočet Silhouette indexu [8]

3.5 Root-mean-square standard deviation

Hodnocení kvality shlukování na základě střední kvadratické směrodatné odchylky, bere v potaz vzdálenosti jednotlivých objektů od středu shluku, kterému tyto objekty náleží. Z toho důvodu obecně tento algoritmus poskytuje lepší hodnoty pro větší počet menších shluků než pro malý počet velkých shluků (experimentálně zjištěno).[8]

$$\left\{\Sigma_i \Sigma_{x \in C_i} \|x - c_i\|^2 / \left[P \Sigma_i \left(n_i - 1\right)\right]\right\}^{\frac{1}{2}} \tag{3.4}$$

Vzorec pro výpočet RMSSTD [8]

3.6 CVNN

Validace na základě nejbližších sousedů je metoda, která počítá kompaktnost a separabilitu shluků na základě nejbližších sousedů. Ovšem počet nejbližších sousedů, který je v tomto algoritmu zahrnut, je potřeba stanovit experimentálně. Kvůli velkému množství cyklů a časové náročnosti takového výpočtu nakonec nebyl tento index do výsledné aplikace zahrnut.

3.7 Porovnávání vůči referenčnímu rozdělení

Při implementaci aplikace byla zahrnuta možnost porovnávat jednotlivá rozdělení do shluků vůči referenčnímu rozdělení. Pokud má uživatel k dispozici nějaké referenční řešení, je možno spočítat, jak velké části tohoto řešení odpovídají jednotlivé výsledky shlukové analýzy. Tato shoda je udávána desetinným číslem z intervalu $\langle 0, 1 \rangle$, kde hodnota 1 znamená stoprocentní shodu.

Kapitola 4

Aplikace pro hodnocení kvality shlukové analýzy

Nedílnou součástí této práce je vývoj aplikace umožňující hodnocení kvality výsledků shlukové analýzy. Tato kapitola se věnuje dokumentaci jejího vývoje a funkčnosti. Dokumentuje nástroje použité k vývoji aplikace, způsob použití, problémy objevené při implementaci a úpravy návrhu vyplývající z těchto problémů.

4.1 Nástroje a technologie použité pro vývoj

Aplikace je naprogramována s použitím programovacího jazyka Java 9 SE. Pro vývoj bylo zvoleno vývojové prostředí IntelliJ IDEA 2017.2.5. Pro sestavení aplikace a vygenerování .jar souboru je použit nástroj maven. Pro generování některých datových sad použitých pro testování aplikace a experimenty byl využit jazyk Python 3.6.5. Pro shlukovou analýzu testovacích datových sad byl použit nástroj RapidMiner Studio Educational 8.0.001. Za účelem verzování a zálohování vývoje byl použit nástroj Git a repozitář na serveru GitHub. K programům IntelliJ IDEA, RapidMiner Studio a službě GitHub byly získány licence pro studijní účely.

4.2 Návrh aplikace

Tato podkapitola pojednává o návrhu aplikace pro hodnocení kvality výsledků shlukové analýzy. Dokumentuje její návrh a strukturu, formát vstupních dat a popisuje její implementaci.

4.2.1 Struktura zdrojového kódu

Zdrojové kódy aplikace jsou vzhledem k rozsahu projektu rozčleněny do dvou modulů:

- caqe
- utils

Prvním je modul `caqe` (tato zkratka znamená Cluster Analysis Quality Evaluation), který obsahuje hlavní části aplikace. Těmi jsou hlavní třída `Main`, poté třídy `Object, Attribute` a `Cluster` uchovávající informace o objektech, jejich atributech a shlucích, dále třída `Deserializer`, která obstarává zpracování vstupního souboru a získávání dat z něj. Třída,

která obstarává výpočty se nazývá `Evaluation` a obsahuje algoritmy pro výpočty indexů (Dunn Separation, Davies-Bouldin, Silhouette, RMSSTD, referenční rozdělení), standardizaci vstupních atributů a pomocné výpočty - například výpočty vzdáleností mezi objekty, či shluky.

Třídy Cluster, Object a Attribute jsou mezi sebou provázány, jelikož objekty třídy Cluster obsahují objekty typu Object, které jsou zase popsány pomocí objektů třídy Attribute.

Třída Main, kde začíná běh programu obstarává instanciaci třídy Deserializer, která získává informace ze vstupních souborů a ukládá je do tříd Cluster, Object a Attribute. Třída Main dále vytváří objekty třídy Evaluation, který využívá informace uložené v třídách Cluster, Object a Attribute pro vypočítání jednotlivých indexů pro hodnocení kvality výsledků shlukové analýzy.

V příloze C je na obrázku C.1 zobrazen vztah mezi třídami.

Druhý modul, který se nazývá `utils` obsahuje pomocné části aplikace, jako jsou například výjimky nebo výčtové typy.

4.2.2 Vstupní data

Vstupními daty aplikace rozumíme nashlukovanou datovou sadu, tedy objekty s jejich atributy přiřazené do shluků. Vzhledem k jednoduché struktuře vstupních dat byl jako vhodný formát vstupního souboru vybrán formát csv.

Každý řádek vstupního souboru kromě prvního, obsahuje informaci o jednom konkrétním objektu. První řádek vstupního souboru je vždy řádkem hlavičkovým. To znamená, že obsahuje informace o datech obsažených v jednotlivých sloupcích. Tato data mohou být atributy objektu, označení shluku do kterého objekt patří nebo označení referenčního shluku, ke kterému objekt náleží.

Jednotlivé hlavičky sloupců, specifikující typ sloupce, mohou obsahovat následující označení:

- refLabel - pro sloupec obsahující referenční shlukování
- labelN - pro sloupce obsahující jednotlivá shlukování
- Jakékoli jiné označení - pro sloupce obsahující atributy objektů

Pokud má sloupec obsahovat informace o nějakém shlukování (kromě referenčního), použijeme hlavičku sloupce labelN (kde N může být libovolné přirozené číslo). To naznačuje, že do jednoho vstupního souboru je možno uvést libovolný počet různých nashlukování datové sady.

Vzhledem ke skutečnosti, že aplikace umožňuje provést hodnocení kvality i pro datové sady obsahující objekty s jinými typy atributů než jsou atributy číselné, je potřeba specifikovat typy těchto atributů. Z tohoto důvodu je aplikace schopna zpracovat ještě druhý vstupní soubor, který tyto informace obsahuje.

Druhý vstupní soubor, který je aplikace schopna zpracovat obsahuje hlavičkový řádek, který v jednotlivých sloupcích obsahuje informace o typech atributů. Pokud tento soubor není aplikaci předán, aplikace vyhodnocuje veškeré atributy objektu jako atributy číselné. Typy atributů, které je aplikace schopna zpracovat, mohou být následující (v závorce je uvedena textová reprezentace typu ve vstupním souboru):

- Číselné (`numeric`)
- Binární (`binary`)

- Nominální (`nominal`)
- Ordinální (`ordinal`)

Číselnými atributy se rozumí reálná čísla - jsou tedy vnitřně reprezentovány jako typ `Double` od čehož se odvíjí jejich rozsah.

Binární atributy jsou atributy dvojhodnotové. Vnitřně jsou reprezentovány jako typ `String`, těmito atributy tedy může být libovolný řetězec znaků, za předpokladu, že se v celé množině objektů náležících konkrétní datové sadě nevyskytují více než dvě různé hodnoty konkrétního atributu.

Atributy nominální jsou podobně jako binární atributy vnitřně reprezentovány datovým typem `String`. V čem se ale od binárních atributů liší, je skutečnost, že množina jejich hodnot může mít kardinalitu větší než 2.

Poslední možný případ jsou atributy ordinální. V některých směrech (jako je kardinalita množiny hodnot nebo přípustný formát atributů) se podobají atributům nominálním. Zásadní rozdíl je ovšem v tom, že ordinální atributy mají přesně definovanou posloupnost hodnot. To znamená, že pro každou hodnotu kromě nejmenší existuje právě jedna předchozí hodnota a pro každou hodnotu vyjma nejvyšší existuje právě jedna následující hodnota. Vzhledem ke skutečnosti, že hodnota ordinálního atributu může být libovolný řetězec znaků, je potřeba získat ze vstupního souboru informaci o pořadí těchto hodnot. Uživatel tyto hodnoty specifikuje ve stejném vstupním souboru, který identifikuje typy atributů a to tím způsobem, že ve sloupci odpovídajícím konkrétnímu nominálnímu atributu tyto hodnoty řádek po řádku specifikuje. Sloupce odpovídající jiným typům atributů než nominálním, zůstanou prázdné.

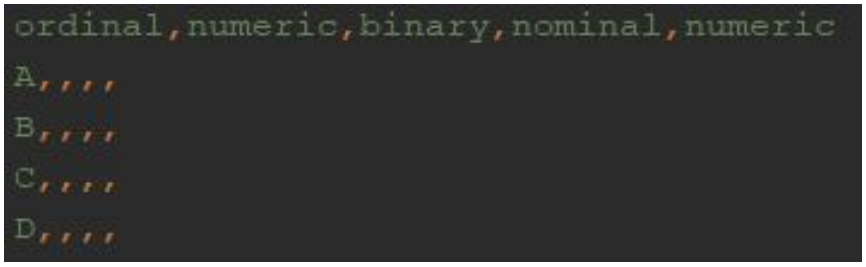
Přesná specifikace typů a v případě ordinálních atributů i posloupnosti hodnot, je velice důležitá kvůli správné standardizaci atributů a následnému výpočtu metrik hodnotících kvalitu nashlukování.

```
attr_0, attr_1,attr_2,attr_3,attr_4,label2,label1,refLabel
A,4,true,blue,4,c1,c1,c1
A,4,true,blue,4,c2,c1,c1
B,2,true,red,3,c1,c2,c2
B,2,true,red,3,c2,c2,c2
C,8,false,green,8,c1,c3,c3
C,8,false,green,8,c2,c3,c3
D,25,false,yellow,25,c1,c4,c4
D,25,false,yellow,25,c2,c4,c4
```

Obrázek 4.1: Příklad vstupního souboru obsahujícího objekty

Na obrázku 4.1 je znázorněn požadovaný formát vstupního souboru. Můžeme v něm vidět, jakým způsobem definovat jednotlivá nashlukování (sloupce label1 a label2), referenční nashlukování (sloupec refLabel) nebo atributy objektů (ostatní sloupce, které mohou mít libovolnou hlavičku – zde attr_0 až attr_4). Jednotlivé řádky pak obsahují objekty náležící této datové sadě.

Vzhledem k tomu, že tato datová sada obsahuje různé typy atributů, je potřeba specifikovat tyto typy v druhém vstupním souboru. Tento soubor je zobrazen na obrázku 4.2. Na



Obrázek 4.2: Příklad vstupního souboru obsahujícího specifikaci typů atributů

prvním řádku tohoto souboru se specifikují typy atributů (numeric, binary, nominal, ordinal). Ostatní řádky tohoto souboru jsou využity pouze v případě, že datová sada obsahuje ordinální atributy. U těchto atributů je potřeba specifikovat pořadí jejich možných hodnot, tedy vypsat do příslušného sloupce hodnoty v požadovaném pořadí (zde hodnoty A, B, C, D v prvním sloupci). Ve sloupcích odpovídajících jiným typům atributů než ordinálním, budou tyto řádky prázdné.

4.3 Implementace porovnávání s referenčním rozdělením

Tato podkapitola pojednává o způsobu implementace porovnávání výsledků shlukové analýzy s referenčním shlukováním a o problémech, které se v průběhu implementace vyskytly.

Pro získání hodnoty této metriky je potřeba mít k dispozici referenční rozdělení do shluků. Toto rozdělení musí být zadáno ve vstupním souboru s hlavičkou sloupce `refLabel`. Hlavička je striktně zadána, aby bylo možno rozlišit referenční rozdělení od ostatních sloupců souboru.

Při implementaci porovnávání s referenčním rozdělením, se vyskytl zásadní problém. Pokud chceme umožnit zadávání jakéhokoli označení jednotlivých shluků, je třeba počítat s tím, že referenční shluky budou označeny jinými názvy než shluky, které vůči nim chceme porovnávat. Nastává tedy otázka, na základě čeho namapujeme názvy shluků v hodnocených rozděleních na rozdělení referenční. Pro vyřešení tohoto problému byly zváženy dvě heuristiky.

4.3.1 Experimentální přiřazení

První heuristika, která byla zvážena, bylo přiřadit označení shluků experimentálně. Tedy vyzkoušet všechna možná přiřazení a zjistit, které z nich zaručí nejlepší výsledek. Při návrhu algoritmu, který by toto experimentální přiřazení provedl, však byla zjištěna zásadní překážka, kterou je náročnost takového výpočtu. Pokud bychom chtěli vyzkoušet všechny způsoby přiřazení, znamenalo by to vyzkoušet všechny možné způsoby uspořádání označení shluků. Například pokud bychom měli 3 shluky v testovaném nashlukování označené jako $1, 2, 3$ oproti 3 shlukům v referenčním nashlukování označeným jako A, B, C, je potřeba zjistit všechna možná uspořádání čísel $1, 2, 3$, což je znázorněno v následujících tabulkách.

A	1	A	1	A	2	A	2	A	3	A	3
B	2	B	3	B	1	B	3	B	1	B	2
C	3	C	2	C	3	C	1	C	2	C	1

Tabulka 4.1: Znázornění možných kombinací při porovnávání s referenčním rozdělením

Zde ale vyvstává onen zásadní problém. A sice pokud hledáme všechny možné kombinace množiny prvků, takovýto výpočet je kombinatorickou permutací, což v praxi znamená, že počet možných přiřazení se rovná $n!$. Pokud je počet shluků nízký, jako jsou třeba 3 shluky v uvedeném příkladě, je počet iterací proveditelný. Problém ovšem nastává se zvyšujícím se počtem shluků. Pokud se počet shluků dostane do dvouciferných čísel (přičemž např. 10 shluků není jako výsledek shlukové analýzy žádný neobvyklý výsledek), počet možných přiřazení se dostává do řádu milionů. Proto byla tato heuristika vyhodnocena jako nevyhovující a pro implementaci této funkcionality byl zvolen jiný přístup.

4.3.2 Přiřazení na základě vzdálenosti shluků

Druhý přístup, který byl nakonec vyhodnocen jako vhodnější a byl implementován do výsledné aplikace, je přiřazení na základě vzdálenosti. Označení shluků se provádí podle toho, který referenční shluk je svým středem nejblíže středu testovaného shluku. Oproti velké výpočetní složitosti experimentálního přístupu je zde potřeba pouze spočítat vzdálenosti středů testovaných a referenčních shluků. Tento přístup je výhodný i z implementačního hlediska, jelikož algoritmy pro výpočet vzdálenosti shluků jsou nedílnou součástí výpočtů metrik pro hodnocení kvality, je tedy možné pro tento přístup využít již implementovaný kód.

Pro implementaci této funkcionality byla využita struktura `SortedMap`, díky které je možné řadit dvojice shluků podle vzdálenosti jejich středů. Díky tomu lze poté jednoduše vybírat dvojice na základě nejmenší vzdálenosti a takto namapovat jednotlivé shluky na shluky referenční.

Nutno také podotknouti, že počet shluků a počet referenčních shluků nemusí být vždy stejné číslo. V takovém případě dojde k injektivnímu zobrazení menší z těchto dvou množin do množiny větší, což v praxi znamená, že některé ze shluků větší množiny zůstanou nepřiřazeny. Toto způsobí, že objekty z těchto shluků automaticky zhoršují výsledek referenčního porovnávání.

4.4 Výstupní data

Výstupní data, tedy hodnoty indexů a případně výsledky porovnání s referenčním rozdělením se tisknou na standardní výstup v podobě tabulky. Řádky tabulky obsahují jednotlivá rozdělení do shluků a sloupce obsahují jednotlivé metriky pro hodnocení kvality. Zelenou barvou je pro každou metriku označena hodnota u toho nashlukování, které bylo danou metrikou vyhodnoceno jako nejlepší.

Jednotlivé metriky hodnotící kvalitu shlukování jsou v aplikaci počítány pro referenční shlukování stejně, jako pro všechna ostatní shlukování.

Vzhledem k tomu, že předpokládaný způsob použití referenčního shlukování je porovnávání jednotlivých nashlukování vůči ideálnímu (které je zadáno jako referenční), bylo potřeba zvážit, jakým způsobem se budou zvýrazňovat výsledky při zadání referenčního rozdělení. Pokud je jako referenční řešení zadáno nejlepší možné, dá se předpokládat, že

většina metrik jej také vyhodnotí jako nejlepší možné (nehledě na to, že samotné desetinné číslo vyjadřující procentuální shodu s referenčním rozdělením, bude v tomto případě vždy 1.0). Jelikož ale aplikace má za úkol vyhodnotit nejlepší nashlukování z výsledků shlukové analýzy a ne z předem známého referenčního rozdělení u kterého předpokládáme jeho optimálnost, bylo rozhodnuto výsledky metrik hodnotících kvalitu tohoto rozdělení nezahrnovat do porovnávání hodnot metrik při zjišťování nejlepšího shlukování.

	DB index	DS index	SIndex	RMSSTD	reference comparison
label2	Infinity	0.0	-0.25	0.2492688042	0.25
label1	0.0	Infinity	1.0	0.0	1.0
refLabel	0.0	Infinity	1.0	0.0	1.0

Obrázek 4.3: Příklad výstupu

Na obrázku 4.3 je zobrazen výstup aplikace (tento výstup byl získán spuštěním aplikace se vstupními daty uvedenými v kapitole 4.2.2).

Kapitola 5

Experimentální část projektu

Součástí této bakalářské práce je také experimentování s vytvořenou aplikací za účelem testování různých kombinací metrik, datových sad a typů atributů a zhodnocení chování těchto metrik v různých případech. Následující kapitola poskytuje dokumentaci provedených experimentů a závěrů, které byly těmito experimenty zjištěny.

5.1 Plánování experimentů

Zásadním krokem při provádění experimentů s aplikací je navržení jejich struktury. Potřebujeme si dobře promyslet mnoho faktorů, které do experimentů vstupují a provést takové experimenty, abychom z jejich výsledků byli schopni získat relevantní informace a vyvodit z nich co nejvíce vypovídající závěr. Za účelem experimentování bylo získáno či vygenerováno několik datových sad různých typů, které budou popsány dále.

Pro získání datových sad byla použita webová stránka Východofinské univerzity [9] a online repozitář obsahující velké množství těchto sad [4].

5.1.1 Jednoduché datové sady

Experimenty s jednoduchými datovými sadami (malý počet objektů popsaných malým počtem atributů, vhodné typy a hodnoty atributů) jsou zařazeny z toho důvodu, že u nich lze učinit předpoklad výsledků, a tudíž na nich můžeme dobře testovat funkčnost aplikace a demonstrovat obvyklé chování algoritmů a metrik s nimiž experimentujeme.

5.1.2 Složitě shlukovatelné sady

Do množiny experimentů je také potřeba zařadit datové sady, které nejsou jednoduše shlukovatelné a u kterých je velká pravděpodobnost, že výsledky jak shlukové analýzy, tak hodnocení kvality, budou velmi rozdílné v závislosti na použitých algoritmech. Toho můžeme dosáhnout použitím sad, ve kterých nelze nalézt žádné očividné souvislosti mezi objekty a neexistuje zřejmé rozdělení do shluků. Tyto experimenty provádíme z důvodu zjišťování chování shlukovacích algoritmů a metrik v neobvyklých situacích, popřípadě za účelem vyhodnocení vhodnosti algoritmů nebo metrik pro různé typy situací.

5.1.3 Sady se specifickými vlastnostmi

Specifickými vlastnostmi datových sad rozumíme například konkrétní typ atributů datové sady nebo různé kombinace typů atributů. V našem případě například budeme provádět

experimenty se sadami obsahujícími pouze binární atributy, kterými budeme zjišťovat, jaké výsledky budou jednotlivé metriky v kombinaci s těmito specifickými typy sad produkovat. Dále budeme experimentovat s datovou sadou obsahující objekty rozmístěné do určitých tvarů. Tedy s takovou datovou sadou u které lze předpokládat dobré nashlukování pomocí algoritmů založených na hustotě a špatné výsledky algoritmů jiných typů.

5.2 Experimentování s počtem shluků

První experiment, který si zde uvedeme, spočívá v hodnocení kvality rozdělení jedné datové sady do různého počtu shluků. Je tedy potřeba vybrat pro tento experiment vhodný algoritmus, který toto umožňuje.

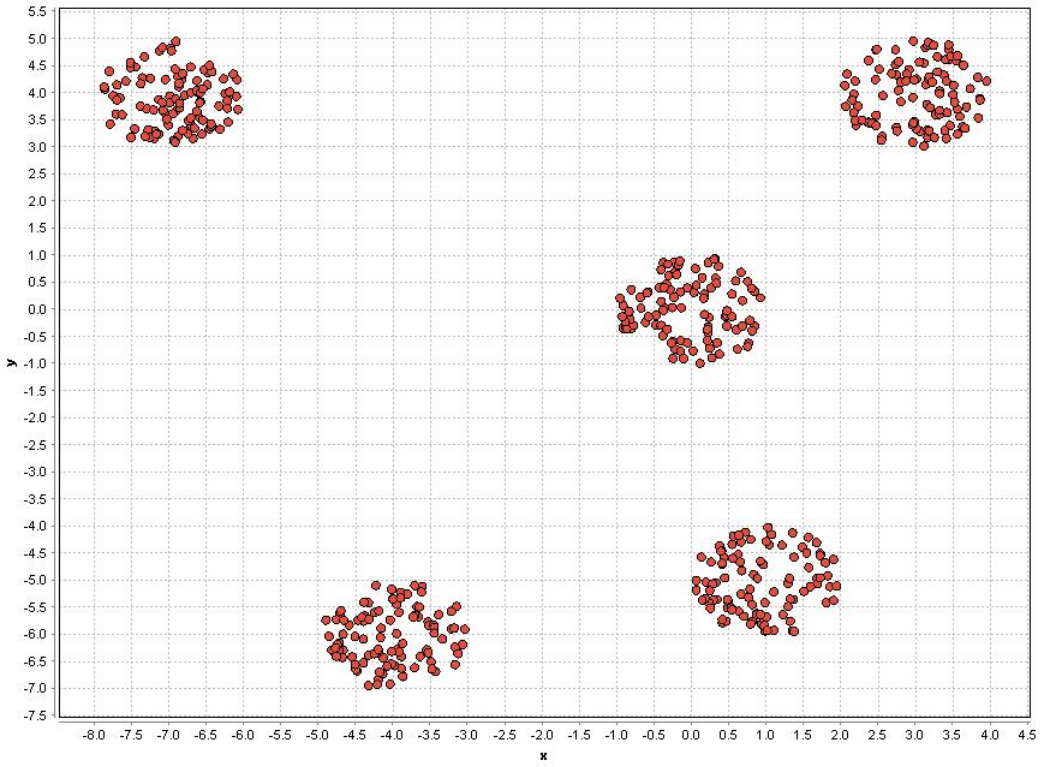
Algoritmus K-Means byl použit pro nashlukování datové sady do různých počtů shluků a poté bylo provedeno hodnocení kvality shlukové analýzy pro každé z těchto rozdělení.

Vzhledem k tomu, že algoritmus K-Medoids je taktéž schopný rozdělit objekty do předem stanoveného počtu shluků a je principiálně podobný algoritmu K-means, byl také použit pro shlukování v tomto experimentu. Výsledky analýzy algoritmem K-medoids byly využity k porovnání jednotlivých nashlukování tímto algoritmem, ale také za účelem porovnání jeho výsledků s algoritmem K-Means.

5.2.1 Jednoduše shlukovatelná sada

Pro první část tohoto experimentu byla uměle vytvořena datová sada, obsahující 500 objektů vyskytujících se v rovině v pěti kruhových oblastech a tyto oblasti jsou od sebe navzájem dobře odděleny (obrázek 5.1). Tedy taková datová sada, u které předpokládáme, že ji algoritmy K-means a K-medoids jsou schopny nashlukovat nejlépe při požadavku na vytvoření pěti shluků. Na tuto datovou sadu byly postupně použity oba výše zmíněné algoritmy a jejich výsledky posloužily jako vstup pro aplikaci, kterou byla zhodnocena kvalita těchto shlukování.

Shlukování této datové sady proběhlo postupně 10 různými způsoby, s požadavkem na rozmístění 2–11 centroidů. Pro každé rozdělení bylo provedeno 10 běhů shlukovacích algoritmů za účelem minimalizace dopadu vlivu výběru počátečních souřadnic. Pro tyto výsledky byla následně vyhodnocena jejich kvalita a na jejím základě byla vybrána shlukování s nejlepšími výsledky, která jsou znázorněny v následujících tabulkách (nejlepší hodnoty každého indexu jsou zvýrazněny zeleně):



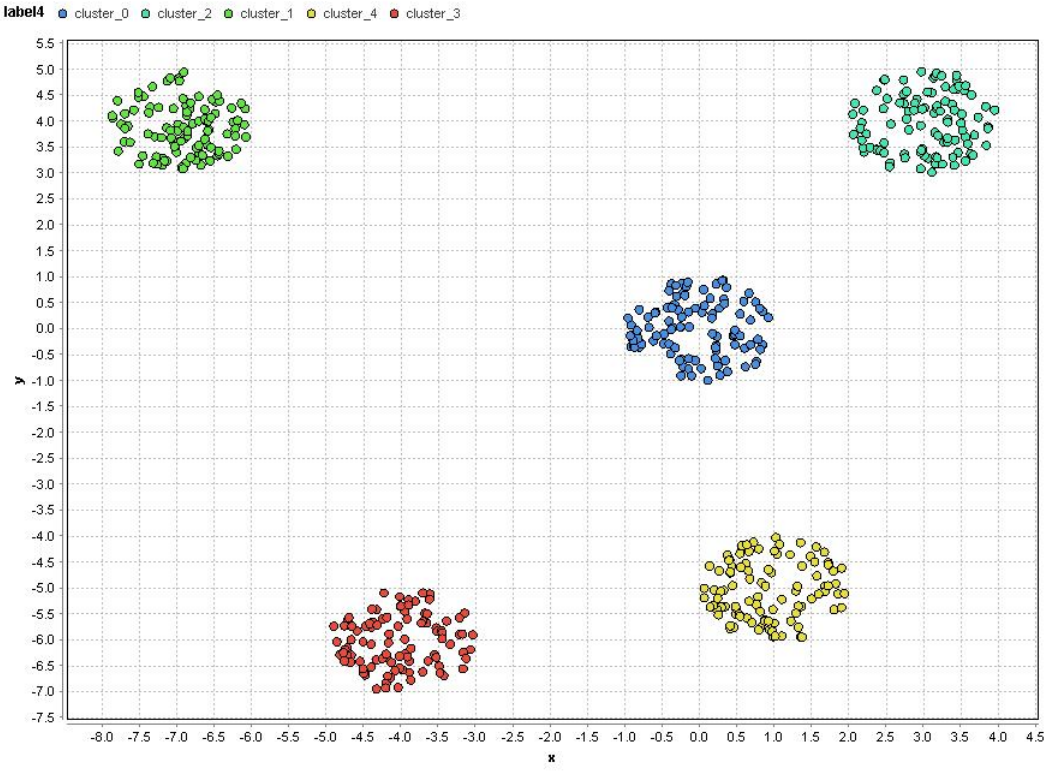
Obrázek 5.1: Jednoduchá datová sada

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
2	0.85796	0.26764	0.53619	2.78610	0.4
3	0.54377	0.45689	0.71536	1.67486	0.6
4	0.40576	0.46078	0.77132	1.23454	0.8
5	0.25045	1.55823	0.82645	0.50263	1.0
6	0.51695	0.07191	0.68007	0.48036	0.904
7	0.71863	0.07191	0.57729	0.45820	0.82
8	0.68334	0.03035	0.56127	0.44579	0.782
9	1.02495	0.03618	0.41298	0.42005	0.66
10	0.92530	0.03577	0.43422	0.39942	0.612
11	1.05098	0.03640	0.37627	0.37808	0.51

Tabulka 5.1: Hodnocení kvality pro rozdělení pomocí algoritmu K-means

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
2	1.02686	0.00682	0.45060	2.89343	0.276
3	0.54377	0.45689	0.71536	1.67486	0.6
4	0.51008	0.00794	0.70917	1.28868	0.8
5	0.25045	1.55823	0.82645	0.50263	1.0
6	0.54976	0.03971	0.66893	0.48293	0.908
7	0.79929	0.02146	0.55187	0.46728	0.85
8	0.92038	0.02542	0.47912	0.44185	0.758
9	0.83827	0.02542	0.48119	0.42737	0.73
10	1.00177	0.02542	0.40374	0.41073	0.65
11	0.93075	0.01858	0.41600	0.39968	0.644

Tabulka 5.2: Hodnocení kvality pro rozdělení pomocí algoritmu K-medoids



Obrázek 5.2: Výsledek shlukování datové sady do 5 shluků pomocí K-means a K-medoids (totožný výsledek pro oba algoritmy)

Jak můžeme pozorovat z výsledků hodnocení kvality, pro sadu takto ideálně rozmístěných objektů téměř všechny metriky jako ideální počet vybraly pět shluků. Jediná metrika, která se od tohoto trendu odchyluje, je RMSSTD, která zakládá hodnocení kvality na kompaktnosti shluků, tedy na vzdálenosti objektů od středů shluků. Proto tato metrika obecně preferuje rozdělení do více shluků.

Dále si můžeme povšimnout, že indexy pro rozdělení do pěti shluků jsou pro všechny metriky totožné u obou shlukovacích algoritmů. Tato skutečnost značí, že výsledky obou algoritmů jsou totožné.

Pro tuto datovou sadu bylo provedeno také porovnání vůči referenčnímu rozdělení. Jako referenční rozdělení bylo použito přirozené rozdělení do pěti shluků podle pozice objektů. Z toho vyplývá, že pokud použijeme vhodný shlukovací algoritmus, který nashlukuje objekty očekávaným způsobem do pěti shluků, měly by výsledky tohoto algoritmu přesně odpovídat referenčnímu rozdělení. Pokud se podíváme do tabulky na řádek obsahující výsledky pro pět shluků, zjistíme, že skutečně existuje stoprocentní shoda s referenčním rozdělením u obou algoritmů.

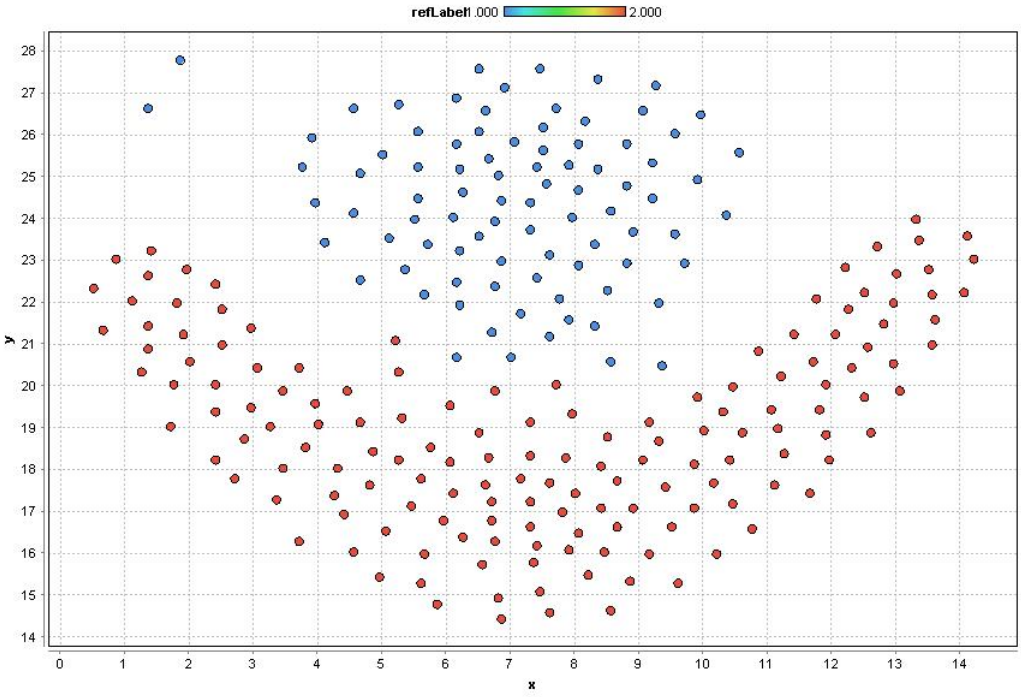
Je třeba také zmínit, že u takto ideálně rozmístěných objektů se výsledky indexů pro ideální rozdělení liší od výsledků indexů pro ostatní rozdělení dokonce řádově (alespoň u DB a DS indexu), což je dáno právě velkou kompaktností a separabilitou shluků u ideálního rozdělení oproti ostatním rozdělením, kde je kompaktnost a separabilita o poznání nižší.

S touto datovou sadou bylo použito také shlukování pomocí DBSCAN a také hierarchické shlukování v kombinaci s DBSCAN. Výsledkem bylo rozdělení do pěti shluků naprosto stejným způsobem jako při použití algoritmů K-means a K-medoids na tuto datovou sadu. Tudíž i výsledky jednotlivých metrik hodnotících kvalitu rozdělení byly totožné jako pro rozdělení do pěti shluků pomocí výše zmíněných algoritmů. Tento výsledek u takto jednoznačně shlukovatelné sady není překvapující.

5.2.2 Obtížně shlukovatelná sada

Pro druhou část experimentu byla použita datová sada obsahující 240 objektů reprezentujících body v rovině (obrázek 5.3)[1]. Tato datová sada je na rozdíl od datové sady z předchozí části experimentu těžce shlukovatelná a dá se předpokládat, že většina algoritmů bude mít se shlukováním v tomto případě problém. Datová sada totiž obsahuje objekty od sebe navzájem vzdálené a nelze mezi nimi najít žádné zjevné souvislosti.

Tato datová sada obsahuje očekávané rozdělení do shluků, vůči kterému bude provedeno porovnávání jednotlivých výsledků získaných shlukovou analýzou. Toto rozdělení vypadá následovně:



Obrázek 5.3: Referenční rozdělení obtížně shlukovatelné sady

Objekty této datové sady mají dva atributy udávající souřadnice objektu. Tato datová sada byla postupně nashlukována pomocí algoritmů K-Means a K- Medoids do 2–11 shluků a poté bylo pomocí aplikace provedeno hodnocení kvality každého z těchto rozdělení. Shlukování pomocí každého z těchto algoritmů bylo provedeno desetkrát pro každý z požadovaných počtů shluků. Hodnocení kvality vyprodukovalo následující výsledky:

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
2	1.11541	0.03339	0.37883	2.56174	0.838
3	0.79952	0.03954	0.41775	2.00923	0.708
4	0.69559	0.04780	0.44814	1.60206	0.621
5	0.84184	0.05399	0.40047	1.47559	0.488
6	0.84637	0.04645	0.39743	1.33832	0.433
7	0.81114	0.07522	0.38883	1.22260	0.329
8	0.79888	0.08133	0.38751	1.11602	0.313
9	0.84820	0.07430	0.37132	1.066782	0.246
10	0.82016	0.08264	0.36524	1.01759	0.221
11	0.80824	0.07853	0.35937	0.97307	0.167

Tabulka 5.3: Hodnocení kvality pro rozdělení pomocí algoritmu K-means

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
2	1.12476	0.03846	0.34544	3.05190	0.671
3	0.95696	0.03715	0.34337	2.99407	0.633
4	1.32438	0.03795	0.24255	2.66564	0.508
5	1.20486	0.03553	0.16209	2.65425	0.496
6	1.50695	0.03795	0.15185	2.65527	0.492
7	1.01835	0.03588	0.27393	2.48560	0.413
8	1.18557	0.03147	0.26139	2.45621	0.088
9	1.06883	0.04167	0.24811	2.39555	0.492
10	1.29560	0.03831	0.17902	2.066	0.233
11	0.92242	0.04272	0.26898	1.96137	0.379

Tabulka 5.4: Hodnocení kvality pro rozdělení pomocí algoritmu K-medoids

Jak můžeme pozorovat ve výsledcích hodnocení kvality, pro každý z algoritmů dostáváme velice odlišné výsledky. Můžeme pozorovat, že pro algoritmus K-Means obecně všechny indexy vyhodnotily lepší výsledky než pro algoritmus K-Medoids. To by mohlo znamenat, že pro tuto sadu je vhodnější algoritmus K-Means, ale u takto problematické sady je potřeba uvažovat i fakt, že i jednotlivé metriky hodnotící kvalitu shlukování mohou mít problém vypočítat výsledky, ze kterých by bylo možné vyvodit smysluplný závěr.

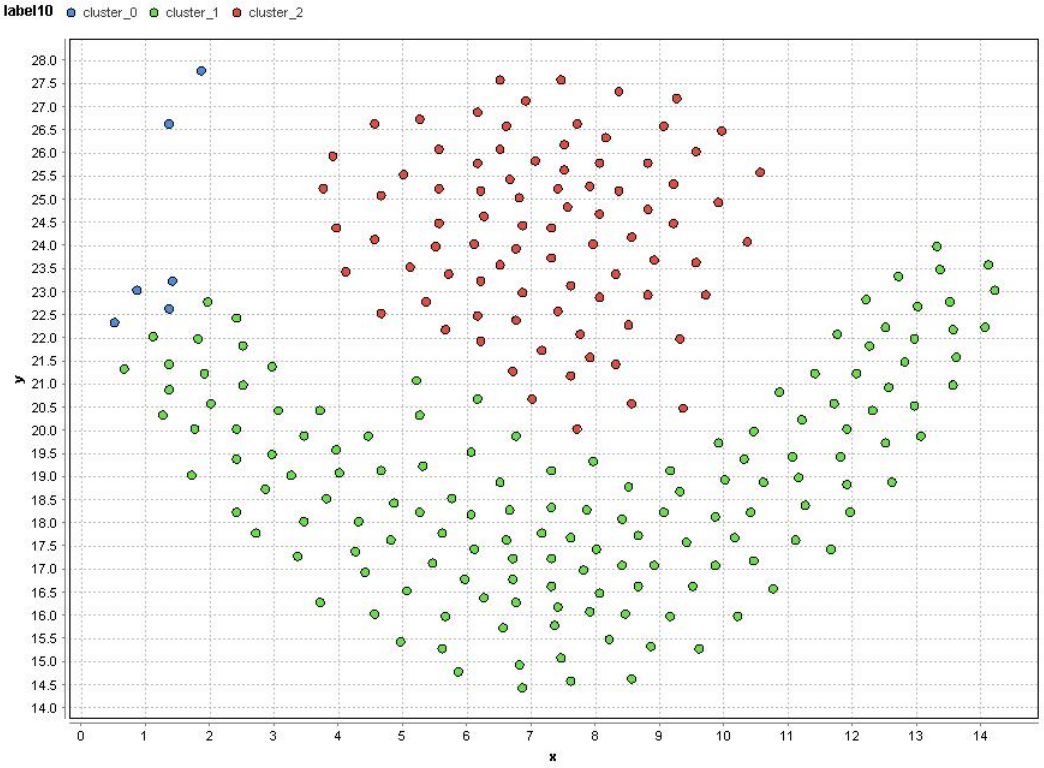
Dále lze pozorovat, že index RMSSTD v obou případech vyhodnotil rozdělení do více shluků jako více se blížící optimálnímu stavu. To je způsobeno s největší pravděpodobností tím, že RMSSTD zakládá hodnocení kvality na vzdálenosti objektů od středů shluků. Při menším počtu shluků existuje větší pravděpodobnost, že se ve shlucích vyskytují objekty příliš vzdálené od jejich středů a tím pádem RMSSTD v takovém případě vyhodnotí rozdělení jako nepříliš kvalitní.

Lze si všimnout, že v některých případech jsou indexy pro různé počty shluků totožné. Tento jev může nastat v případě, kdy jsou počty shluků podobné (například zde u algoritmu K-medoids v případě Dunn Separation indexu u rozdělení na 4 a 6 shluků) a nashlukování jsou si velmi podobná, natolik aby jejich odlišnosti v případě některých metrik neovlivnily průběh výpočtu.

Pro demonstraci obtížnosti shlukování této datové sady, byla tato sada nashlukována ještě pomocí jiného algoritmu, než jsou dva výše zmíněné. Jelikož jsou K-means a K-medoids principiálně podobné byl vybrán ještě odlišný způsob, kterým je shlukování založené na hustotě objektů pomocí algoritmu DBSCAN. Tento algoritmus byl postupně zkoušen s různými hodnotami parametru ϵ v intervalu od 0.5 do 2.0 a s parametrem určujícím minimální počet objektů v intervalu od 5 do 30. V drtivé většině těchto shlukování bylo výsledkem přiřazení téměř celé sady do jednoho shluku. Nakonec byly ale experimentálně zjištěny hodnoty těchto parametrů, pro které je algoritmus DBSCAN schopen rozdělit datovou sadu do dvou shluků přibližně očekávaným způsobem. Pokud $\epsilon = 1.9$ a minimální počet objektů je 20, získáme nashlukování, pro které hodnocení kvality produkuje výsledky uvedené v tabulce 5.5:

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
3	0.51717	0.00743	0.57784	21.64868	0.96666

Tabulka 5.5: Hodnocení kvality pro rozdělení pomocí algoritmu DBSCAN



Obrázek 5.4: Obtížně shlukovatelná sada rozdělená algoritmem DBSCAN

Je patrné, že mezi výsledky hodnocení kvality neexistuje prakticky žádná pozorovatelná souvislost. Každý z použitých algoritmů datovou sadu nashlukoval jinými způsoby a indexy pro hodnocení kvality produkují velice odlišné výsledky. Lze tedy prohlásit, že použitá datová sada je velmi obtížně shlukovatelná použitými algoritmy.

U použité datové sady je k dispozici i referenční rozdělení do dvou shluků. Můžeme pozorovat, že shlukování pomocí metody DBSCAN rozdělilo datovou sadu do tří shluků s přesností 96.6% oproti referenčnímu rozdělení. U algoritmu K-Medoids, přestože většina metrik vykazuje horší výsledky, při rozdělení do dvou shluků odpovídalo referenčnímu řešení 67.1% objektů a u K-means byla tato hodnota téměř 84%. I z těchto hodnot lze pozorovat, že s touto datovou sadou mají problémy nejen shlukovací algoritmy, ale i metriky hodnotící kvalitu jejich výsledků.

5.2.3 Závěr experimentu

V průběhu tohoto experimentu bylo zjištěno, že u jednoznačně shlukovatelných sad mohou různé shlukovací metody produkovat identické výsledky. Tudíž výsledky s identickou kvalitou. Dále v průběhu experimentu s obtížně shlukovatelnou sadou bylo demonstrováno, že existují datové sady, u nichž většina shlukovacích algoritmů není schopna produkovat jednoznačné výsledky a s nimiž mají problém i metriky hodnotící kvalitu výsledků shlukové analýzy.

5.3 Experimentování s binárními sadami

Druhý experiment který byl v rámci tohoto projektu vyzkoušen, bylo experimentování s binárními sadami. Tedy se sadami, jejichž objekty jsou popsány pouze binárními atributy za účelem zjištění chování shlukovacích algoritmů a metrik hodnotících kvalitu v kombinaci s těmito sadami.

5.3.1 Experiment s jednoduchou datovou sadou

Pro účely prvního experimentu byla vygenerována jednoduchá datová sada obsahující náhodný vzorek 500 lidí. Tato datová sada obsahuje 3 binární atributy. První z nich udává, jestli daný člověk je muž nebo žena (hodnoty všech atributů byly generovány na základě rovnoměrného rozdělení pravděpodobnosti - u tohoto atributu s pravděpodobností 50.8% žena a 49.2% muž). Druhý atribut označuje, zdali má daná osoba vysokoškolské vzdělání (pravděpodobnost tohoto jevu je 19.8% u žen a 19% u mužů). A konečně třetí atribut označuje, zdali je daná osoba kuřákem (pravděpodobnost tohoto jevu je 16% u žen a 28% u mužů). Všechny tyto hodnoty byly získány z dokumentů Českého statistického úřadu. [6][7]

Jak si lze u této datové sady povšimnout, tak vzhledem ke skutečnosti, že obsahuje tři binární atributy, existuje celkem osm možných kombinací těchto atributů. Bylo by možné tedy nashlukovat datovou sadu pohodlně do osmi různých shluků. Vyzkoušíme tedy různá rozdělení do shluků a spočítáme hodnocení kvality pro nashlukování pomocí algoritmů K-means, K-medoids a DBSCAN.

Algoritmy K-means a K-medoids byly spouštěny v 10 bězích za účelem minimalizace dopadu náhodné volby počátečních pozic centroidů na výsledek shlukování.

Jako referenční rozdělení pro tuto datovou sadu použijeme rozdělení do osmi shluků, které bylo zmíněno v předchozím odstavci, tedy každá možná kombinace atributů bude přiřazena vlastnímu shluku.

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
2	0.77703	0.5	0.64060	0.13078	0.654
3	0.63772	0.5	0.66701	0.10177	0.752
4	0.37468	1.0	0.79637	0.08013	0.828
5	0.45451	1.0	0.74734	0.08769	0.794
6	0.31030	1.0	0.80451	0.08296	0.816
7	0.37746	1.0	0.86627	0.07244	0.86
8	0.21695	1.0	0.88339	0.06061	0.902
9	0.59160	1.0	0.71509	0.09063	0.78
10	0.39012	1.0	0.79025	0.06530	0.886
11	0.33961	1.0	0.85959	0.10745	0.692

Tabulka 5.6: Hodnocení kvality shlukování binární sady algoritmem K-means

Z tabulky výsledků můžeme vyvodit několik závěrů. Předně je na první pohled zřejmé, že Dunn-Separation index je pro hodnocení výsledků shlukování binárních sad tohoto typu naprosto nevhodná metrika. Z dvouhodnotových výsledků vyskytujících se v tabulce nelze ani zdaleka vyčíst nejlepší nashlukování pro tuto datovou sadu. U ostatních metrik však již dostáváme více informací o kvalitě rozdělení. Všechny ostatní použité metriky se shodly na rozdělení do osmi shluků jako nejlepším možném.

Zajímavý výsledek dostáváme u RMSSTD, jelikož jak jsme si nastínili u předchozího experimentu, tak tato metrika obecně upřednostňuje větší počty shluků. Jenže v tomto případě existuje osm možných kombinací a pokud by každá z těchto kombinací patřila do svého vlastního shluku, každý shluk by byl soustředěn v jednom bodě a tím pádem by byla směrodatná odchylka nulová. Jelikož v tomto případě je RMSSTD index nenulový, lze z tohoto výsledku usoudit, že ani rozdělení na osm shluků neproběhlo optimálním způsobem.

Nedokonalost rozdělení můžeme také pozorovat u porovnání s referenčním rozdělením. Jelikož byla datová sada generována uměle a každé z osmi možných kombinací byl přiřazen vlastní referenční shluk, tak by v optimálním případě výsledek shlukové analýzy stoprocentně odpovídal referenčnímu rozdělení. Zde ovšem vidíme jako nejvyšší hodnotu v tabulce 0.902, rozdělení do osmi shluků tedy vykazuje pouze devadesátiprocentní shodu s referenčním rozdělením.

Abychom zjistili příčinu tohoto jevu, resp. nedokonalosti tohoto rozdělení, stačí zanalyzovat výsledky, které jsme dostali shlukovou analýzou pomocí K-means. Z nich lze zjistit, že ačkoli byl algoritmus K-means spuštěn s požadavkem na vytvoření osmi shluků, tak dva shluky neobsahují ani jeden objekt a de facto se tedy datová sada rozdělila do šesti shluků. Kvalita výsledků získaných pomocí algoritmu K-means ve velké míře závisí na počátečních pozicích centroidů a může se stát, že některým centroidům nebudou přiřazeny žádné objekty obzvláště u takovéto datové sady, kde jsou objekty soustředěny do osmi bodů a v některých bodech se nachází o mnoho objektů více než v ostatních. Četnosti možných kombinací atributů jsou znázorněny v následující tabulce:

Pohlaví	Kuřák	Univerzitní vzdělání	četnost
Muž	Ano	Ano	14
Muž	Ano	Ne	49
Muž	Ne	Ano	32
Muž	Ne	Ne	140
Žena	Ano	Ano	5
Žena	Ano	Ne	38
Žena	Ne	Ano	35
Žena	Ne	Ne	187

Tabulka 5.7: Četnosti kombinací atributů v použité datové sadě

V ideálním případě by každý z centroidů byl jako jediný umístěn do jedné z osmi skupin objektů.

Z těchto důvodů u takovéto datové sady mohou tedy algoritmy K-means a K-medoids selhávat. Vzhledem ke skutečnosti, že jde o sadu, kde jsou objekty soustředěny do míst s vysokou hustotou, lze předpokládat, že například algoritmus DBSCAN, který je na hustotě výskytu objektů založen, by měl vyprodukovat ideální nashlukování. Algoritmus DBSCAN bude otestován s touto datovou sadou později v tomto experimentu.

Pro srovnání byl proveden ještě experiment se shlukováním této datové sady pomocí algoritmu K-medoids.

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
2	0.77703	0.5	0.64060	0.13078	0.654
3	0.53646	0.5	0.65873	0.10885	0.682
4	0.38803	0.5	0.73793	0.08938	0.816
5	0.32965	0.5	0.82487	0.07185	0.892
6	0.40133	0.5	0.81028	0.08021	0.828
7	0.39335	0.5	0.81609	0.07339	0.886
8	0.14922	1.0	0.91415	0.05266	0.926
9	0.42014	0.5	0.78529	0.07339	0.886
10	0.14922	1.0	0.91415	0.05266	0.926
11	0.28101	1.0	0.80080	0.07491	0.85

Tabulka 5.8: Hodnocení kvality shlukování binární sady algoritmem K-medoids

Z tabulky můžeme vidět, že DS index produkuje i pro algoritmus K-medoids v kombinaci s binární datovou sadou neuspokojivé výsledky. Také si můžeme povšimnout, že v tomto případě algoritmus K-medoids vyprodukoval dle použitých metrik kvalitnější výsledky než K-means.

Další věc, kterou lze z tabulky zjistit, je, že všechny metriky jsou pro rozdělení do 8 a 10 shluků totožné, což naznačuje stejný výsledek u obou počtů shluků. K tomuto jevu pravděpodobně došlo z důvodu soustředění 500 objektů do osmi různých pozic, při snaze o rozdělení do shluků může dojít v rámci jedné pozice k zanedbání některých shluků - tedy některé ze shluků nebudou mít přiřazen ani jeden objekt.

Opět ovšem nedošlo k optimálnímu rozdělení objektů. Byl tedy proveden ještě jeden pokus s jiným algoritmem. Tímto algoritmem je DBSCAN a v následující tabulce jsou zobrazeny jeho výsledky:

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
8	0.0	Infinity	1.0	0.0	1.0

Tabulka 5.9: Hodnocení kvality shlukování binární sady pro rozdělení pomocí algoritmu DBSCAN

Z tabulky je patrné, že algoritmus DBSCAN vyprodukoval ideální rozdělení do osmi shluků, které stoprocentně odpovídá referenčnímu řešení. Také hodnoty ostatních indexů jsou nejlepší jakých tyto indexy mohou nabývat. Jelikož je DBSCAN metoda založená na shlukování podle hustoty objektů a tato datová sada vytváří osm různých bodů s velkou hustotou objektů, funguje DBSCAN velice dobře.

5.3.2 Experiment s více atributy

Pro druhou část experimentu s binárními atributy byla vybrána datová sada obsahující simulovaná data reprezentující vzorek pacientů, kteří utrpěli zranění hlavy. Tato sada obsahuje 11 atributů a pro tento experiment z ní bylo využito prvních 1000 objektů. [2] [3]

Počet shluků	DB index	DS index	Silhouette index	RMSSTD
2	1.87983	0.09090	0.40208	0.03995
3	1.64920	0.1	0.29875	0.04264
4	1.54270	0.1	0.42203	0.03692
5	1.21762	0.11111	0.29368	0.03393
6	1.24070	0.11111	0.27061	0.03462
7	1.26611	0.11111	0.28072	0.03110
8	1.75411	0.11111	0.29026	0.03169
9	1.06033	0.125	0.33969	0.02904
10	1.16168	0.14285	0.36582	0.03286
11	1.09738	0.14285	0.34576	0.02772

Tabulka 5.10: Hodnocení kvality shlukování binární sady algoritmem K-medoids

Z výsledků v tabulce lze u DS indexu zjistit, že stále obsahuje velké množství duplicitních hodnot a to i přesto, že tato datová sada má více atributů než sada z první části tohoto experimentu a obsahuje dvojnásobné množství objektů.

U této datové sady, kde nejsou objekty tolik soustředěny v jednom bodě, můžeme u RMSSTD opět sledovat obvyklý trend v upřednostňování většího počtu shluků, i když ne v takové míře jako tomu bylo u experimentů s počty shluků.

Indexy DB a Silhouette se shodují na rozděleních do 9–11 shluků a produkují zde lepší výsledky než u ostatních počtů shluků (kromě Silhouette indexu u 2 a 4 shluků), což může naznačovat, že pro tuto datovou sadu je vhodnější rozdělení do většího počtu shluků.

5.3.3 Závěr experimentu

V průběhu tohoto experimentu bylo zjištěno, že DS index není příliš vhodný pro hodnocení kvality použitých binárních sad. Dále bylo ukázáno, že RMSSTD nemusí vždy upřednostňovat větší počet shluků a v některých ideálních případech může preferovat optimální rozdělení.

5.4 Experimentování se specifickou datovou sadou

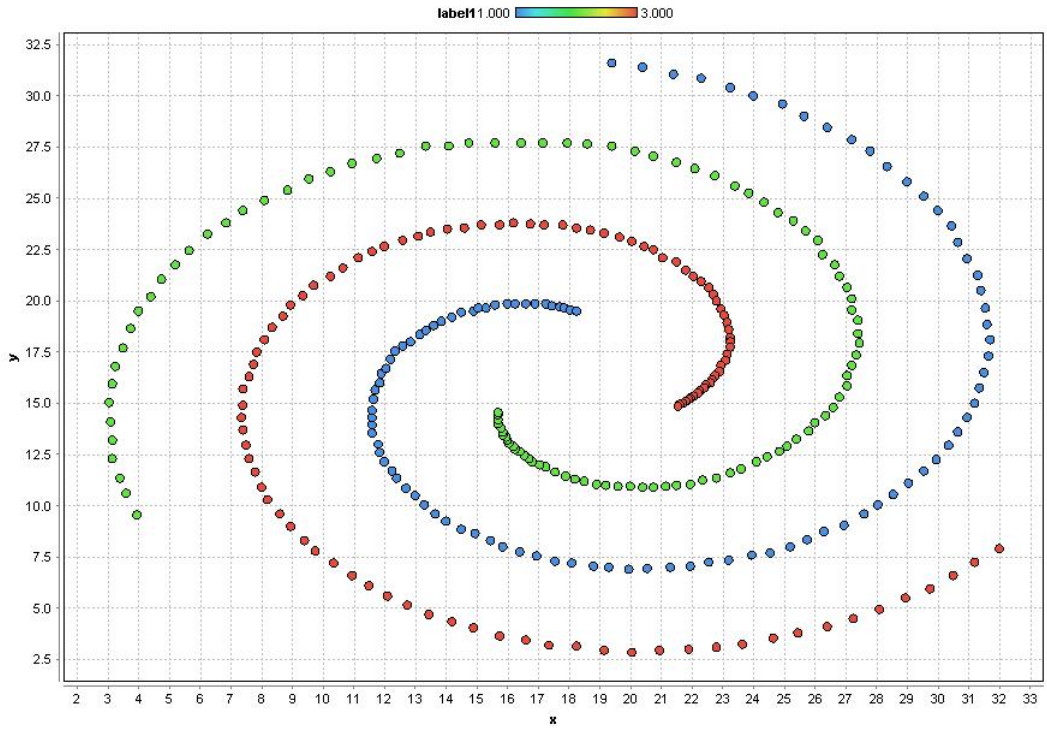
V následující sekci jsou popsány experimenty provedené s různými shlukovacími algoritmy nad datovou sadou se specifickými vlastnostmi.

5.4.1 Datová sada a návrh experimentu

Datová sada použitá v tomto experimentu obsahuje dvoudimenzionální data reprezentující souřadnice objektů v rovině. Sada obsahuje 312 objektů, které jsou uspořádány do tvaru spirály o třech ramenech (obrázek 5.5) [5]. Očekáváme tedy, že v ideálním případě budou objekty rozděleny do tří shluků, kde každý shluk bude obsahovat objekty z jednoho ramena spirály.

Tato konkrétní sada byla vybrána z důvodu, že objekty jsou strukturovány do ramen spirály a tyto ramena mají tedy poměrně vysokou hustotu objektů. Dá se tedy předpokládat, že v tomto případě budou pro shlukování nejlepší volbou algoritmy založené na hustotě, tedy například DBSCAN. Naopak u algoritmů jako K-means nebo K-medoids, které mají tendenci rozdělovat objekty do shluků kruhového tvaru, lze očekávat, že nebudou schopny nashlukovat sadu předpokládaným způsobem.

K této sadě je k dispozici také referenční rozdělení, jsme tedy schopni zjistit u jednotlivých nashlukování, z jaké části odpovídají požadovanému rozdělení do tří shluků.



Obrázek 5.5: Referenční rozdělení spirálovité datové sady

V průběhu experimentu bude postupně provedeno shlukování pomocí DBSCAN a K-Means. Poté bude provedeno hodnocení kvality těchto nashlukování, za účelem zjištění chování jednotlivých metrik u sady s takovým specifickým uspořádáním, jako je tato.

5.4.2 DBSCAN algoritmus

V první části tohoto experimentu byla datová sada nashlukována pomocí algoritmu DBSCAN. Jako vstupní parametry tohoto algoritmu bylo při prvním běhu zadáno $epsilon = 1.0$ a minimální počet objektů ve shluku 5. Poté byl spuštěn ještě druhý běh algoritmu, kterému byly zadány vstupní argumenty $epsilon = 2.0$ a minimální počet objektů ve shluku 5. Po vyhodnocení kvality výsledků těchto shlukování dostáváme hodnoty metrik uvedené v tabulce 5.11.

Počet shluků	ϵ	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
5	1.0	3.09737	0.01520	0.26191	6.82566	0.19551
4	2.0	4.61179	0.03546	0.18522	6.96559	0.99679

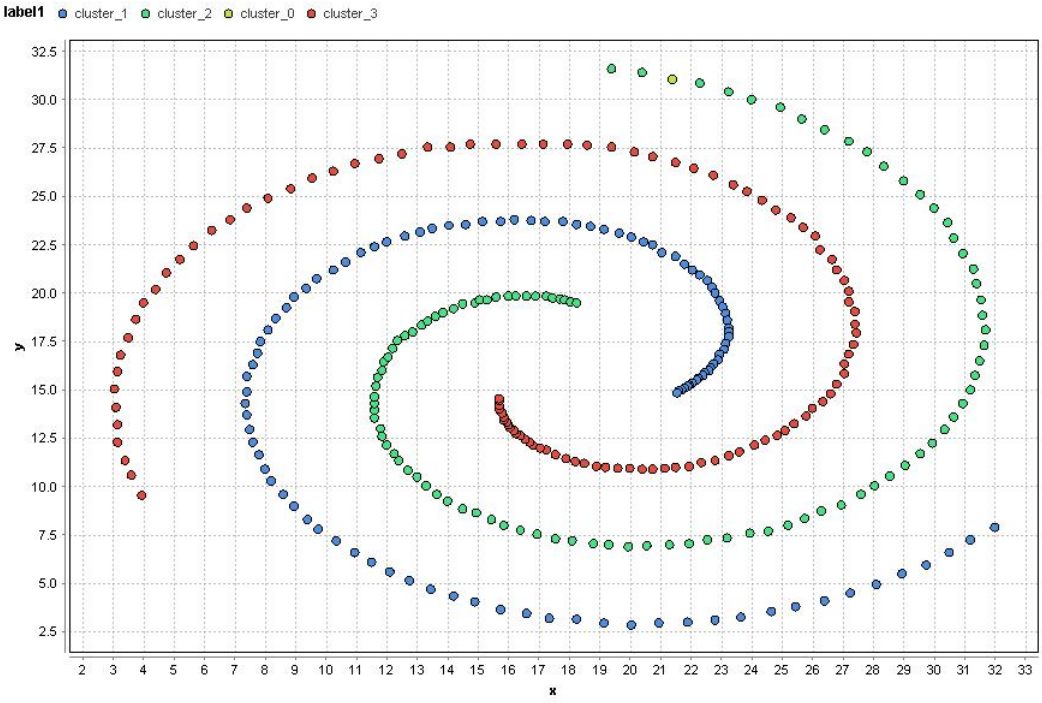
Tabulka 5.11: Hodnocení kvality shlukování pro rozdělení pomocí algoritmu DBSCAN

Algoritmus sadu rozdělil při prvním běhu do pěti shluků, což je vzhledem k použité datové sadě neočekávaný výsledek. Při pohledu na hodnoty metrik v tabulce zjistíme, že hodnoty u Davies-Bouldin indexu a Dunn Separation indexu jsou v tomto případě někdy i o jeden až dva horší, než tomu bylo například u experimentů s těžce shlukovatelnou sadou, což naznačuje, že s touto datovou sadou mají metriky problém. Co se týče výsledku

porovnání vůči referenčnímu rozdělení, můžeme z něj vyčíst že pouze 19.5% objektů bylo zařazeno do shluků stejně. Tento velice špatný výsledek může být způsoben nízkou hodnotou parametru epsilon.

K druhému běhu algoritmu s jinými vstupními hodnotami bylo přistoupeno zejména kvůli neuspokojivým výsledkům prvního běhu i přesto, že od něj byl očekáván optimální výsledek referenčního porovnání. Tento algoritmus byl tedy spuštěn znovu, ovšem s jiným vstupním parametrem epsilon. Při druhém běhu byla hodnota tohoto parametru stanovena na 2.0, pro minimální velikost shluku zůstala hodnota stejná, tedy 5 objektů.

Při analýze výsledků druhého běhu algoritmu DBSCAN zjistíme, že hodnoty DB indexu, Silhouette indexu a RMSSTD jsou ještě horší než při běhu prvním. A to navzdory tomu, že shlukování proběhlo téměř optimálním způsobem. Porovnávání s referenčním rozdělením totiž vyprodukovalo hodnotu 99.7%, což vzhledem k počtu objektů obsažených v analyzované datové sadě znamená, že všechny objekty s výjimkou jediného byly přiřazeny do shluků stejně jako v rozdělení referenčním. Jediný špatně přiřazený objekt byl přiřazen do svého vlastního shluku (lze pozorovat v obrázku 5.6, objekt znázorněný světle zelenou barvou v horním rameni spirály), proto můžeme v tabulce vidět, že počet shluků, do kterých byla datová sada rozdělena, je 4. Jediný index, který toto rozdělení preferuje, je DS index.



Obrázek 5.6: Téměř ideální rozdělení pomocí DBSCAN

5.4.3 K-means algoritmus

Dalším algoritmem, který bude v rámci tohoto experimentu testován, je K-means. Jak už jsme si nastínili v úvodu experimentu, tak K-means se typicky snaží tvořit kruhové shluky, díky tomu, že přiřazuje objekty k centroidům na základě vzdálenosti. Měl by tedy

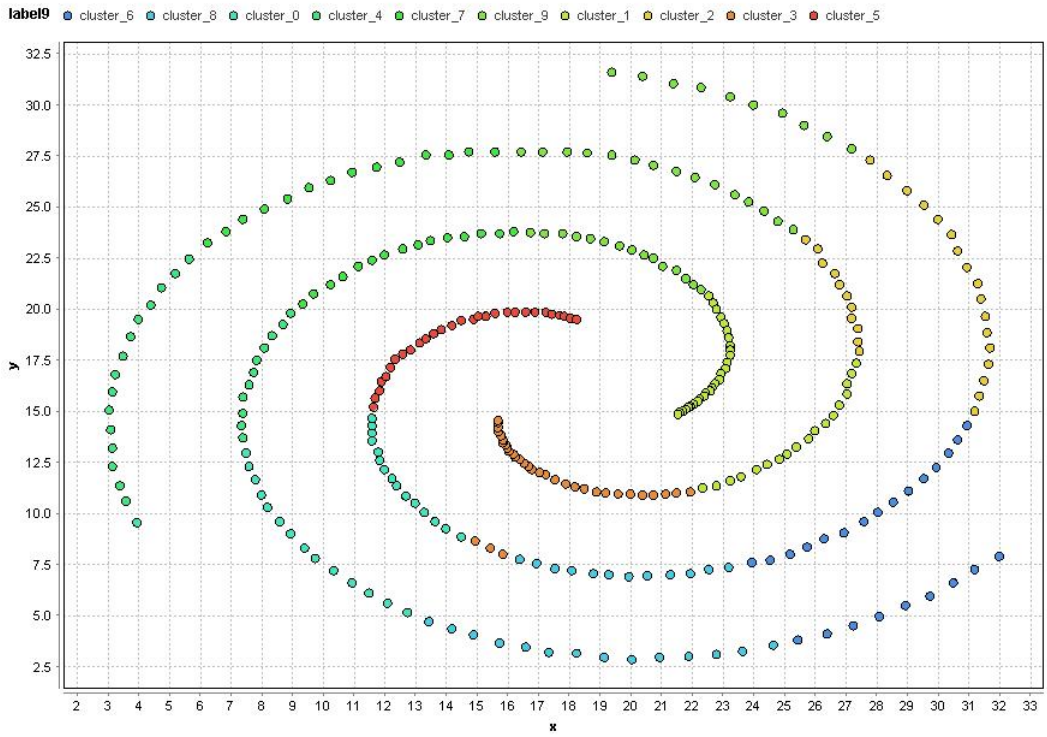
již z jeho podstaty být pro shlukování takovéto datové sady zcela nevhodný, pokud chceme jako výsledek dostat samostatný shluk pro každé rameno spirály.

Pro účely tohoto experimentu byla datová sada nashlukována postupně se dvěmi až jedenácti počátečními centroidy a výsledky jsou následující:

Počet shluků	DB index	DS index	Silhouette index	RMSSTD	Referenční porovnání
2	1.16740	0.00716	0.34710	5.61954	0.32051
3	0.88507	0.00666	0.36091	4.45895	0.33974
4	0.87880	0.01784	0.35409	3.86561	0.23717
5	0.89349	0.01416	0.34631	3.46113	0.18589
6	0.87588	0.02455	0.34877	3.13011	0.14423
7	0.87416	0.02563	0.34424	2.96264	0.11217
8	0.84973	0.02189	0.36424	2.76403	0.06410
9	0.89669	0.02525	0.35125	2.61331	0.02564
10	0.82347	0.03060	0.37202	2.45992	0.0
11	0.81274	0.02611	0.36802	2.34143	0.0

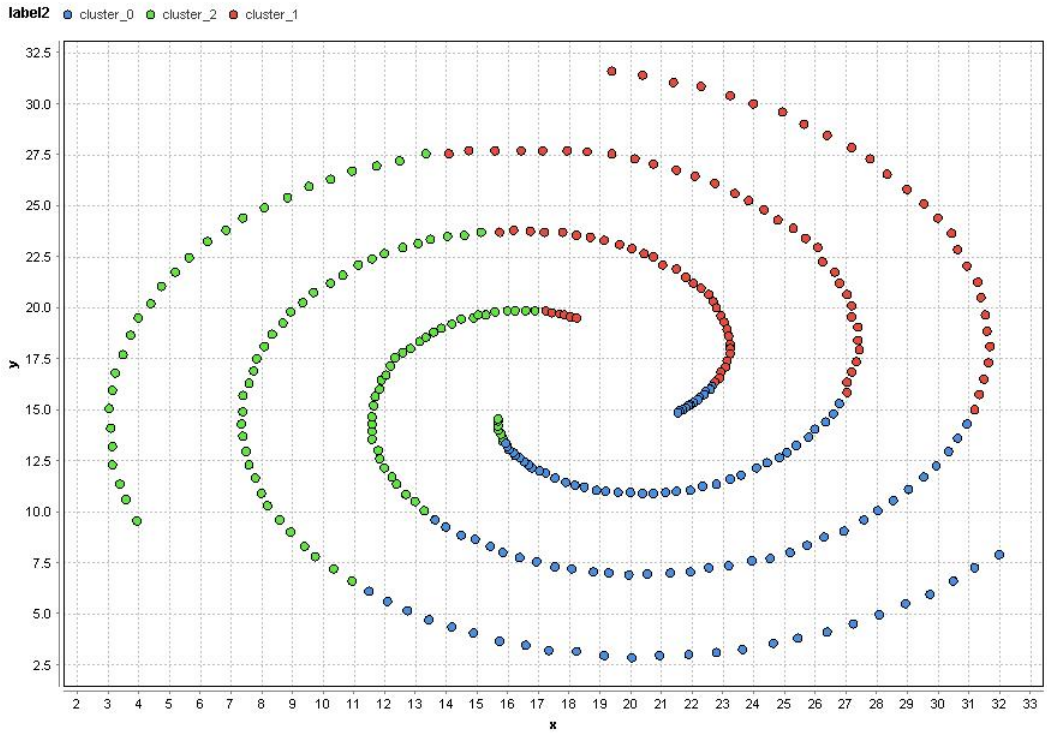
Tabulka 5.12: Hodnocení kvality shlukování spirálovité sady algoritmem K-means

Z výsledků hodnocení kvality jsme se dozvěděli, že všechny metriky upřednostňují rozdělení do více shluků. Konkrétně do 10 a 11 shluků přestože zde není absolutně žádná shoda s referenčním rozdělením. Nulová shoda s referenčním rozdělením je dána skutečností, že díky specifickému spirálovitému uspořádání objektů je střed jednotlivých ramen nejblíže středu shluku, který se utvořil z části ramene jiného a jelikož mapování shluků získaných shlukovacím algoritmem na shluky referenční se provádí na základě vzdáleností středů, tak se na sebe namapují shluky, které nemají žádné společné objekty. Na následujícím obrázku můžeme vidět rozdělení do 10 shluků:



Obrázek 5.7: Rozdělení spirálovité datové sady do deseti shluků algoritmem K-Means

Kde ovšem dochází k největší shodě s očekávaným rozdělením, je u rozdělení do tří shluků. Přesto tato shoda nastává pouze zhruba u třetiny objektů. Tato skutečnost je dána tím, že K-means rozdělí datovou sadu do tří přibližně kulovitých shluků a každý z těchto shluků překryje přibližně třetinu každého ramene spirály. Rozdělení do tří shluků je znázorněno na obrázku 5.8:



Obrázek 5.8: Rozdělení spirálovité datové sady do tří shluků algoritmem K-Means

Pokud srovnáme výsledky hodnocení kvality pro DBSCAN a pro K-means, zjistíme, že většina metrik upřednostňuje shlukování pomocí K-means, v případě DB indexu má tato metrika dokonce řádově lepší hodnoty než u DBSCAN. Jediná metrika, která lépe hodnotí výsledky DBSCAN, je Dunn Separation index. Tedy ta stejná metrika, která jako jediná správně vyhodnotila téměř ideální nashlukování pomocí DBSCAN s $\epsilon = 2.0$ jako lepší než DBSCAN s $\epsilon = 1.0$.

5.4.4 Závěr experimentu

V tomto experimentu bylo ukázáno, že metriky pro hodnocení kvality nejsou univerzální a nefungují u všech datových sad stejným způsobem. Z výsledků je patrné, že u této datové sady u rozdělení pomocí DBSCAN se vstupním parametrem epsilon = 2.0, které dopadlo téměř stejně jako bylo očekáváno, většina indexů dopadla hůře než u rozdělení s epsilon = 1.0, které bylo dle výsledků porovnání s očekávaným rozdělením o 80% horší. Dále tyto metriky vyhodnotily shlukování pomocí K-means jako lepší než pomocí DBSCAN, přestože K-Means není pro tuto datovou sadu vhodným algoritmem.

Na základě těchto informací můžeme vyvodit závěr, že u této datové sady minimálně DB index, Silhouette index a RMSSTD nefungují očekávaným způsobem. Jediný DS index byl

schopný ze všech různých způsobů nashlukování označit za nejlepší ten, kteý téměř kopíruje očekávané rozdělení.

Kapitola 6

Závěr

Cílem této bakalářské práce bylo získat teoretické znalosti o shlukové analýze a hlavně o možnostech a způsobech hodnocení kvality jejích výsledků. Neméně důležitá část práce obnášela využití nabytých znalostí v praxi a vytvoření aplikace schopné hodnotit kvalitu výsledků shlukové analýzy.

V průběhu práce na tomto projektu se vyskytlo několik úskalí, která jsou většinou zdokumentována v textu práce. To hlavní spočívalo v tom, že ačkoli se shlukové analýze a dolování dat věnuje mnoho publikací, tak hodnocení kvality výsledků se tyto publikace většinou dotýkají pouze okrajově nebo vůbec. Naštěstí však některé z těchto publikací obsahují alespoň matematické vzorce pro výpočet metrik hodnotících kvalitu shlukové analýzy, které je možno algoritmizovat a tyto algoritmy použít pro implementaci požadované aplikace.

Informace, které jsem měl k dispozici jsem tedy prostudoval a na jejich základě aplikaci implementoval. Po konzultaci s vedoucí mé práce jsem vybral datové sady, tyto jsem podrobil shlukové analýze a následně použil pro experimenty s vytvořenou aplikací. Těmito experimenty byly zjištěny informace jak o chování samotných shlukovacích algoritmů, tak o chování metrik hodnotících kvalitu shlukování.

Z výsledků experimentů byla zjištěna například tendence RMSSTD zpravidla lépe hodnotit rozdělení do více shluků, jelikož je tento algoritmus založen na kompaktnosti těchto shluků. U experimentů s binárními datovými sadami pak bylo zjištěno, že DS index není vhodný pro hodnocení jejich kvality, jelikož u takovýchto datových sad produkuje pouze omezené množství hodnot a často udává stejnou hodnotu pro větší množství různých nashlukování. U experimentů s datovou sadou obsahující objekty uspořádané do tvaru spirály byl jako nejlepší metrika pro hodnocení kvality určen Dunn Separation index, který preferoval očekávané rozdělení. Navíc také tento index upřednostňoval algoritmus DBSCAN před algoritmem K-means a potvrdil tak předpoklad, že pro datovou sadu specifického tvaru jako je tato, je vhodnější DBSCAN jakožto algoritmus založený na hustotě objektů, než K-means jakožto algoritmus založený na vzdálenosti od reprezentujících objektů.

Pro potenciální budoucí verze aplikace existuje prostor pro její rozšíření. Ať už o další metriky pro hodnocení kvality, tak například o další modul, který by byl schopen provádět shlukovou analýzu, aby uživatel pro shlukování datových sad nemusel používat další software. Dále by bylo vhodné zkoumat chování metrik v kombinaci s dalšími datovými sadami, jelikož vzorek obsažený v této práci se věnuje pouze několika základním typům datových sad. Různých datových sad lze ovšem nalézt či vygenerovat nespočet.

Literatura

[1] Flame.
URL https://cs.joensuu.fi/sipu/datasets/flame.txt

[2] Minor Head Injury (Simulated) Data.
URL
https://vincentarelbundock.github.io/Rdatasets/csv/DAAG/head.injury.csv

[3] Minor Head Injury (Simulated) Data Description.
URL
https://vincentarelbundock.github.io/Rdatasets/doc/DAAG/head.injury.html

[4] Rdatasets.
URL https://vincentarelbundock.github.io/Rdatasets/

[5] Spiral.
URL https://cs.joensuu.fi/sipu/datasets/spiral.txt

[6] Ženy a muži v datech. Český statistický úřad.
URL https://www.czso.cz/documents/10180/45709986/30000417.pdf/1fa799cb-c008-4271-a09c-9035e22923cc?version=1.2

[7] Zaostřeno na ženy a muže. Český statistický úřad.
URL https://www.czso.cz/documents/10180/45709978/30000217.pdf/92da801e-56d6-4d45-8cef-67497ed59949?version=1.1

[8] Aggarwal, C. C.; Reddy, C. K.: *Data Clustering: Algorithms and Applications*. CRC Press, 2012, ISBN 978-1-4665-5821-2.

[9] et al, P. F.: Clustering basic benchmark. 2015.
URL http://cs.uef.fi/sipu/datasets/

[10] Cios, K. J.; Pedrycz, W.; Swiniarski, R. W.; aj.: *Data Mining: A Knowledge Discovery Approach*. Springer Science+Business Media, 2007, ISBN 978-0-387-33333-5.

[11] Han, J.; Kamber, M.; Pei, J.: *Data Mining: Concepts and Techniques Third edition*. Morgan Kaufmann Publishers, 2012, ISBN 978-0-12-381479-1.

Příloha A

Obsah CD

Přiložené CD obsahuje archiv `Bachelors_thesis.tar.gz`, ve kterém se nacházejí následující součásti projektu:

- Zdrojové kódy aplikace včetně Makefile pro vytvoření spustitelného .jar souboru
- Datové sady použité v experimentální části projektu
- Tento text bakalářské práce v elektronické podobě, včetně zdrojových souborů systému LaTeX.

Příloha B

Návod k aplikaci pro hodnocení kvality výsledků shlukové analýzy

Tato aplikace je distribuována jako konzolová a je k ní dodán Makefile pro snadné sestavení java archivu a spuštění s několika vzorovými vstupními soubory. Tyto vstupní soubory jsou tytéž, které byly použity v experimentální části tohoto projektu.

Seznam cílů programu make:

- all – Sestaví java archive CAQE-1.0.jar
- simpleKMeans, simpleKMedoids – Spustí aplikaci s jednoduchou datovou sadou nashlukovanou pomocí K-Means nebo K-Medoids.
- difficultKMeans, difficultKMedoids, difficultDBSCAN – Spustí aplikaci s těžce shlukovatelnou sadou nashlukovanou pomocí K-Means, K-Medoids nebo DBSCAN
- binaryKMeans, binaryKMedoids – Spustí aplikaci s binární datovou sadou nashlukovanou pomocí KMeans nebo KMedoids
- complexBinary – Spustí aplikaci s komplexnější binární datovou sadou nashlukovanou pomocí KMedoids
- spiralKMeans, spiralDBSCAN1, spiralDBSCAN2 – Spustí aplikaci se spirálovitou datovou sadou nashlukovanou pomocí KMeans nebo pomocí DBSCAN s $\epsilon = 1$ (spiralDBSCAN1) nebo s $\epsilon = 2$ (spiralDBSCAN2).

Pro spuštění aplikace s libovolným vstupním souborem se používá po jejím přeložení příkaz:

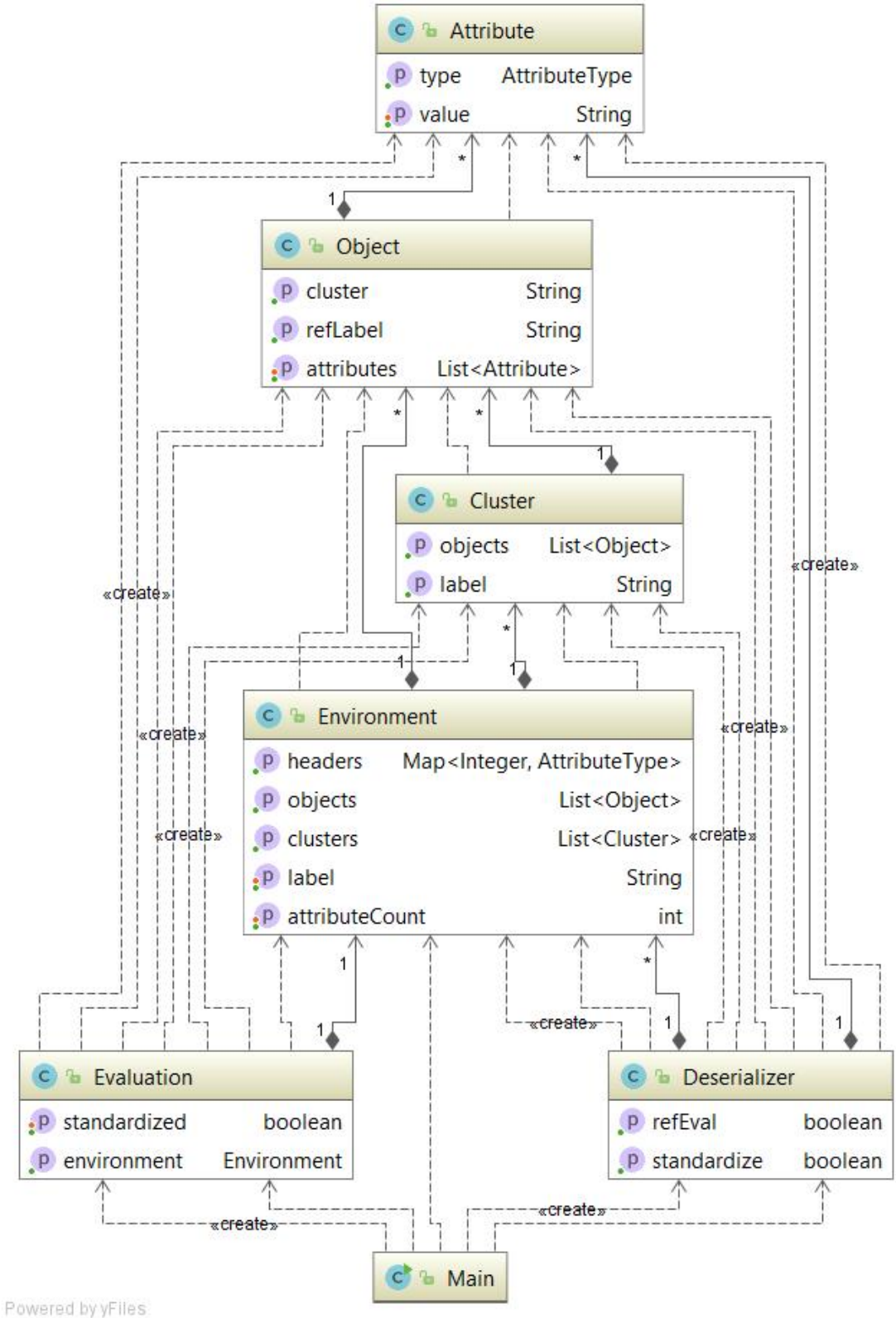
```
java -jar CAQE-1.0.jar inputFile headerFile
```

kde inputFile je cesta ke vstupnímu souboru obsahujícímu nashlukovanou datovou sadu a headerFile cesta k souboru obsahujícímu specifikaci typů atributů.

Příloha C

Diagram tříd

V této příloze se nachází diagram zobrazující vztahy mezi třídami aplikace. Tento diagram byl vygenerován z vývojového prostředí IntelliJ IDEA 2017.2.5.



Obrázek C.1: Diagram tříd