TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

# Shluková analýza v data miningových úlohách

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika
*Studijní obor:* 1802T007 – Informační technologie

*Autor práce:* **Bc. Václav Kozdeba**
*Vedoucí práce:* RNDr. Klára Císařová, Ph.D.

Liberec 2016

**Technická univerzita v Liberci**
**Fakulta mechatroniky, informatiky a mezioborových studií**
Akademický rok: **2016/2017**

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Bc. Václav Kozdeba** |
| Osobní číslo: | **M14000166** |
| Studijní program: | **N2612 Elektrotechnika a informatika** |
| Studijní obor: | **Informační technologie** |
| Název tématu: | **Shluková analýza v data miningových úlohách** |
| Zadávající katedra: | **Ústav mechatroniky a technické informatiky** |

Zásady pro vypracování:

1. Prostudujte základní metody shlukové analýzy a problémy spojené se vstupními daty pro shlukovou analýzu v DM.
2. Analyzujte data miningové úlohy, pro které je shluková analýza vhodným řešením.
3. Vytvořte aplikaci pro výklad vybraných algoritmů shlukové analýzy a jejich testování.
4. Aplikaci testujte na studentech oboru Informační technologie v navazujícím studiu.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

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

Seznam odborné literatury:


[1] **Řezanková H., Húsek D., Snášel V.: Shluková analýza dat, Professional publishing, 2009**

[2] **Lukasová A., Šarmanová J.: Metody shlukové analýzy. SNTL, Praha 1985.**

[3] **Yong Yin, Ikou Kaku, Jiafu Tang: Data Mining, Springer London Ltd, 2011**

[4] **Olivia Parr Rud: Data mining, Computer Press, a.s., 2006**


Vedoucí diplomové práce: **RNDr. Klára Císařová, Ph.D.**
Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: **10. října 2016**

Termín odevzdání diplomové práce: **15. května 2017**

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2016

# Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 3.1.2017

Podpis: 

**Poděkování**

Děkuji své vedoucí diplomové práce RNDr. Kláře Císařové, Ph.D. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování diplomové práce věnovala. Dále děkuji své rodině a přítelkyni, za umožnění studia, správné motivace a podpory, která mi v průběhu studia byla věnována. Nakonec děkuji všem mým spolužákům za pomoc během studia na této univerzitě.

## Abstrakt

Práce je příspěvkem ke kolekci vzdělávacích e-learningových materiálů pro předměty, které se věnují data miningovým postupům. Teoretická část shrnuje vybrané problémy shlukové analýzy, která je zásadní pro část data miningových úloh. V práci jsou shrnuté problémy spojené s předzpracováním vstupních kvalitativních i kvantitativních dat pro shlukové algoritmy. Dále jsou analyzované vybrané postupy shlukování a posouzení kvality shlukovacího procesu.

Výsledkem práce je výukový program MyCluster pro popsané algoritmy, který studentům umožňuje experimentovat s daty, které si sami interaktivně vytvoří. Použít lze i datové množiny dříve vytvořené. K dispozici jsou metody nejbližšího souseda, nejvzdálenějšího souseda, centroidní metoda a metoda K-means a několik volitelných metrik. Student může sledovat důsledky volby jednotlivých parametrů a lépe pochopit principy sledováním animace shlukovacího procesu. Kvalitu shlukovacího procesu pak ilustruje funkcionál kvality, který lze sledovat v souvislostech několika pokusů v historii pokusů. Aplikace byla v průběhu vývoje testována studenty Technické univerzity v Liberci a na základě jejich doporučení byla upravena do finální podoby. Aplikace MyCluster byla naprogramována pomocí programovacího jazyka C# a je k dispozici účastníkům kurzu Datamining a kurzu Programovací techniky a data mining na e-learningovém portálu ALS na TUL.

**Klíčová slova:** data mining, shluková analýza, předzpracování dat, míry podobnosti, optimální počet shluků, funkcionál kvality

## Abstract

This thesis is a contribution to a collection of educational e-learning materials for subjects which deals with data mining procedures. The theoretical part summarizes selected problems of the cluster analysis which is essential for part of data mining tasks. This thesis summarizes problems connected with preprocessing of input qualitative and quantitative data for clustering algorithms. Further analyzed are selected clustering techniques and evaluation of the quality of the clustering process.

The result of this thesis is a learning program MyCluster for described algorithms which allows the students to experiment with the data that they themselves created. Students can also use sets of data that had been created earlier. In program are available methods like nearest neighbor, farthest neighbor, centroid linkage and K-means method and several chosen metrics. Students can monitor the impact of chosen parameters and better understand the principal of clustering process by monitoring the process with animation. The quality of the clustering process then ilustrates the functional of quality which can be monitored in coherence of several trials in the past trials. During the development of this program the students was allowed to test it and based on their recommendations was made into its final appearance. Application MyCluster was programmed using C# and it is available to all participants of Datamining course and Programming technique and data mining course on e-learning portal of ALS on TUL.

**Key words:** data mining, cluster analysis, data preprocessing, similarity measures, optimal number of clusters, cluster evaluation

# Obsah

## Seznam obrázků



## Seznam tabulek

# Úvod

Mezi jednu z mnoha data miningových metod se řadí také shluková analýza. Shluková analýza má úzkou souvislost s naší lidskou povahou, kdy se všechno snažíme zařazovat do určitých skupin, které jsou definovány specifickými vlastnostmi. Cílem metod shlukové analýzy je tedy vytvořit shluky (resp. skupiny) nad datovým souborem, kde platí, že objekty v jednom shluku jsou si více podobné než objekty z různých shluků. Tzn., že objekty jsou do shluků zařazeny na základě jejich vlastností, kterými jsou objekty definovány. Proces shlukování není tak jednoduchý, jak se může na první pohled zdát. Před samotnou aplikací data miningových metod shlukové analýzy na datový soubor je nedílnou součástí tzv. předzpracování dat, protože v reálném světě jsou často v datových souborech chybné či neúplné údaje. Předzpracování dat se skládá z kroků, jako je „očištění" datové souboru, kdy se data miner snaží eliminovat nekonzistenci dat, integrace dat, kdy má data miner společně s odborníkem pro daný problém spojit data z více datových souborů do jednoho souboru, a nakonec datové transformace, kde má každý typ proměnné jiné možnosti transformování.

Jak je psáno v odstavci výše, tak cílem shlukové analýzy je rozdělit množinu objektů na základě jejich podobnosti. K určení velikosti podobnosti nám slouží tzv. míry podobnosti. Výpočet míry podobnosti se rozlišuje na základě datového typu. Shluková analýza se rozděluje na hierarchické a nehierarchické metody. Hierarchické metody vytvářejí při procesu shlukování hierarchickou strukturu shluků, kdežto nehierarchické metody se snaží najít optimální rozložení objektů do předem stanoveného počtu shluků tím, že objekty přemisťují z jednoho shluku do dalšího. U nehierarchických metod, zmiňovaných v této diplomové práci, je požadováno od dataminera zadání počtu požadovaných výsledných shluků. Ke stanovení optimálního počtu shluků existují globální a lokální metody, které se za pomoci vnitro-shlukové a mezi-shlukové variability snaží vypočítat optimální počet shluků v datovém souboru. Po ukončení shlukovacího procesu by měl být dataminer schopen určit kvalitu shluků či celého rozkladu. Určit, zdali je shluk kvalitní či nikoliv je sice kontroverzní téma, ale existují různé druhy výpočtů kvality.

S ohledem na výše uvedené je cílem této diplomové práce vytvořit strukturovaný popis jednotlivých kroků shlukové analýzy. První kapitola je tedy zaměřena na předzpracování dat, druhá kapitola je věnována shlukové analýze, jejím metodám, stanove-

ním optimálního počtu shluků a míře podobnosti. Ve třetí kapitole je popsán funkcionál kvality rozkladu a poslední kapitola teoretické části je věnována praktickému využití metod shlukové analýzy v různých odvětvích.

V praktické části diplomové práce se zabývám popisem mnou vytvořené aplikace MyCluster. Aplikace je vytvořena za účelem učební pomůcky pro studenty Technické univerzity v Liberci. Cílem je, aby aplikace pomohla studentům při výkladu vybraných algoritmů shlukové analýzy, a také aby na těchto algoritmech mohl student provádět experimenty s jeho vlastní množinou bodů, kterou si intuitivně sám vytvoří. Před samotným ukončením vývoje byla aplikace testována samotnými studenty a dle jejich poznatků byla upravena.

# 1 Předzpracování dat

Data jsou pro data miningové úlohy klíčovým prvkem. V tomto směru se často vyskytuje zkratka GIGO (anglická zkratka pro „Garbage In, Garbage Out“), která nám říká, že pokud do sebelepšího modelu (procesu) vložíme špatná data, tak můžeme očekávat pouze špatný výsledek. Špatnými daty může být myšleno, buď chybějící hodnoty tzv. *null* hodnoty, odlehlé hodnoty tzv. *outliers*, nemožné kombinace (například Pohlaví: Muž a Těhotenství: Ano), duplicitní řádky a mnoho dalších. Posouzení kvality dat je také často závislé na řešeném problému. Odlehlá hodnota může být někdy hrubou chybou, například chybou měření nebo při vkládání hodnot ručně, kdy se vkládající splete. V jiných úlohách může odlehlá hodnota naopak signalizovat hledanou anomálii – například u klienta banky, který se chová jinak než většina, a tím u něho „roste“ podezření z podvodné činnosti.

Prvním krokem procesu přípravy dat je takzvané čištění dat (data cleaning), kdy detekujeme a odstraňujeme nebo opravujeme nekonzistenci v datovém souboru. Při řešení nekonzistence v datovém souboru můžeme narazit na již zmíněné chybějící hodnoty, odlehlé hodnoty, nemožné kombinace a tak dále. Všechny uvedené nekonzistence jsou v reálném světě zcela normálním jevem a v datových souborech jsou běžně k nalezení. Chybějící hodnoty můžou být řešeny například těmito způsoby:

- doplnění hodnoty ručně,
- vynechání celého objektu,
- doplnění nejpravděpodobnější hodnoty…

K tomu, abychom zjistili, jestli se v našem datovém souboru vyskytují *outliers* využíváme například následující metody.

První metoda využívá interkvartilového rozpětí. Interkvartilové rozpětí je interval, ve kterém se nachází přesně 50 % hodnot z daného souboru. Spočítá se tedy jako rozdíl horního a dolního kvartilu:

$$IQR = x_{0,75} - x_{0,25}. \tag{1}$$

Potom za *outliers* považujeme ty hodnoty, které jsou od dolního, respektive horního kvartilu vzdálené více než 1,5násobek IQR. Tzn., že pokud je splněna jedna ze dvou následujících podmínek:

$$x_i < x_{0,25} - 1{,}5 * IQR, \tag{2}$$

$$x_i > x_{0,75} + 1{,}5 * IQR, \tag{3}$$

tak potom je hodnota $x_i$ pro náš soubor *outlierem*. Interkvartilové rozpětí je pro posouzení odlehlosti výhodné z toho důvodu, že není citlivé vůči odlehlým hodnotám. Ve druhé metodě se využívá z-skóre, což kvantitativně posuzuje polohu hodnoty $x_i$ vzhledem k aritmetickému průměru a směrodatné odchylce souboru. Z-skóre tedy vypočteme pomocí následujícího vzorce:

$$z_i = \frac{x_i - \mu}{\sigma}, \tag{4}$$

kde $z_i$ je z-skóre pro danou hodnotu, $x_i$ je hodnota ze souboru, $\mu$ je aritmetický průměr souboru a $\sigma$ je směrodatná odchylka souboru. Potom pokud je pro $x_i$ absolutní hodnota z-skóre větší než 3, tak se jedná o odlehlou hodnotu. Z-skóre je díky použití aritmetického průměru souboru a směrodatné odchylky silně ovlivněno odlehlými hodnotami. Další metodou, pomocí které jde odhalit odlehlé hodnoty a není jimi tolik ovlivněna jako z-skóre, je využití mediánové souřadnice. Stejně jako u z-skóre, tak i u mediánové souřadnice, pokud je její absolutní hodnota větší než 3, tak se jedná o odlehlou hodnotu. Pro výpočet se využívá tzv. MAD, což je medián absolutních odchylek od mediánu. Mediánovou souřadnici vypočteme pomocí následujícího vzorce:

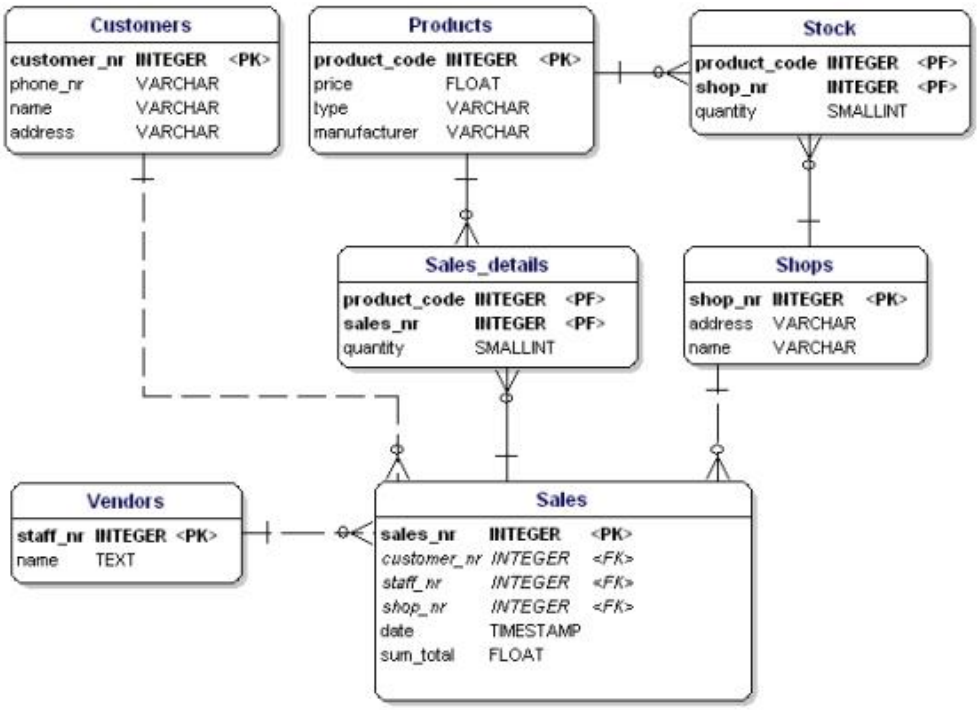
$$m_i = \frac{x_i - x_{0,5}}{1{,}483 * MAD}, \tag{5}$$

kde $m_i$ je mediánová souřadnice pro danou hodnotu $x_i$, $x_i$ je hodnota ze souboru a $x_{0,5}$ je 50 % kvantil, který rozděluje datový soubor tak, že jedna polovina hodnot je menších a druhá polovina hodnot je rovných nebo větších než $x_{0,5}$.

V dalším kroku je nutné zvážit, jestli některé atributy úplně nevynechat z důvodu irelevantnosti v dané analýze nebo proto, že jsou atributy silně korelované.

Často se vyskytujícím krokem je takzvaná integrace dat, kde se „spojují“ data z několika zdrojů v jeden datový soubor. Pro integraci dat je nutná spolupráce mezi dataminerem a odborníkem pro daný problém, který má naprostý přehled a zná veškeré souvislosti mezi jednotlivými zdroji dat a rozumí dané oblasti. Data mining je použitelný pro každou oblast lidské činnosti, kde se ukládají elektronická data. Pro ilustraci šířky data miningových projektů lze zmínit několik příkladů využití data miningu.

V ekonomických úlohách lze technikami data miningu například předvídat finanční zdraví podniku, v obchodě se používá data mining pro optimální práci se zákazníky, to například znamená, že lze odhadovat jejich chování, nasadit akce pro udržení zákazníka a neprodělat, nabídky poskytovat účelně podle zákaznických preferencí. V technických provozech data miningová řešení pomáhají předcházet haváriím či poruchám, v medicíně lze pomocí data miningu predikovat některá onemocnění či šíření epidemií, dále lze z internetových dat hledat bezpečnostní rizika při komunikaci na sociálních sítích. Mnoho dalších příkladů lze najít v literatuře či na webových stránkách, za pozornost stojí https://www.kaggle.com/, http://archive.ics.uci.edu/. Uvedený výčet není, ani nemůže být úplný, má jen ilustrovat široký odborný záběr data miningových úloh či projektů. Dataminer nemůže být znalec všech potřebných odborností, a proto musí spolupracovat s odborníky na danou oblast tzv. „doménovými inženýry“ a to zejména ve fázi porozumění datům a přípravy dat pro modelování. Na obrázku 1 je ukázka databázového schématu, kde jednotlivé tabulky mohou sloužit jako zdroje dat pro datový soubor, který bude vstupem pro data miningové algoritmy. Vstupem pro data miningové algoritmy obvykle bývá tabulka, resp. dvojrozměrné pole dat, ve kterém řádky představují jednotlivé zkoumané případy, někdy označované jako objekty, a ve sloupcích jsou popisné data objektů, označované jako atributy. Problémem databázových zdrojů dat je rozhodnout, které tabulky spojit a jak je správně spojit, abychom měli všechna relevantní a důležitá data o objektech v daném dvourozměrném poli. Při integraci dat dále vzniká značná redundance dat. Redundance vyplývá z „povinné“ vstupní datové struktury pro data miningové algoritmy. Zdroje dat pro data miningové úlohy nemusí být pouze databáze, i když jsou nejobvyklejší formou. Zdrojem dat mohou být i jinak elektronicky uložená data například: csv soubory, statistické soubory, excelovské soubory a další.

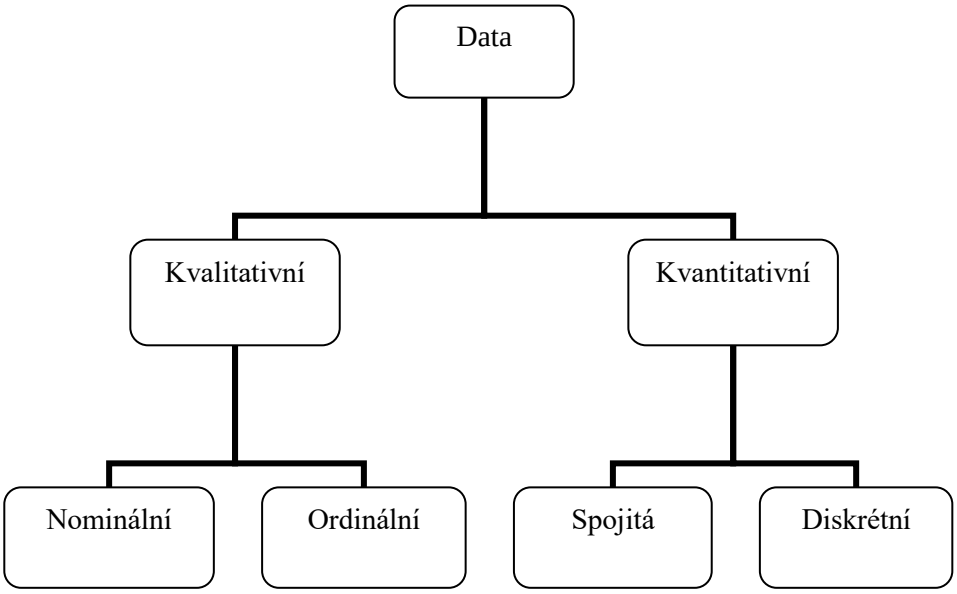



**Obrázek 1 DB schéma**

Zdroj: Introduction to database design. In: *Datanamic* [online]. Nizozemsko: Datanamic Solutions BV, 2008 [cit. 2016-12-08]. Dostupné z: http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html

Dalším krokem ve fázi přípravy dat jsou různé jejich transformace. Pro transformaci dat máme mnoho důvodů, některé plynou z analýzy dat a jiné z data miningových algoritmů, které budou základem modelování. Každý typ proměnné nám určuje, jaký druh transformace s ním jde provádět. Typy proměnných se dělí na dvě základní skupiny. První skupinou jsou data kategorická neboli data kvalitativní, která jsou popsána slovně nebo pomocí popisných znaků. Kvalitativní data se dále dělí na nominální data a ordinální data. U nominálních dat jsme schopni posoudit, zda jsou různé, avšak nejsme schopni stanovit jejich pořadí (například barva očí, značka televize, druh zeleniny). Ordinální data již dokážeme seřadit, ale nejsme schopni určit o kolik je jedna proměnná větší nebo menší než proměnná druhá (například kategorizovaná číselná proměnná). Druhou skupinou jsou data kvantitativní. Jedná se o proměnné vyjádřené pomocí číselných hodnot, které se dále rozdělují na data spojitá a data diskrétní (nespojitá). Příkladem pro spojité hodnoty může být uvedeno měření teploty stroje a pro diskrétní hodnoty může být uveden příjem klienta v tisících či věk v létech. Kvantitativní data lze v případě potřeby algoritmu definovat jako data kategorická, tedy kvalitativní

nebo jindy jako data číselná, tedy kvantitativní. Kategorizaci číselných dat lze realizovat převodem na intervalovou proměnnou. Určení intervalů na daném souboru lze provést několika postupy. Dělit data do intervalů lze podle počtu intervalů do pevné šířky intervalu, nebo podle „kvantilů“, kdy je počet hodnot v intervalu stejný, či dalšími postupy zejména při klasifikačních úlohách, kdy o dělení rozhoduje i cílová (predikovaná) proměnná. Speciálním typem proměnné je proměnná dichotomická, kterou bychom mohli zařadit do skupiny nominálních proměnných s tím rozdílem, že dichotomická proměnná může nabývat pouze dvou hodnot (například muž – žena, horký – studený, zdravý – nemocný). S dichotomickou proměnnou se poté pracuje jako s 1 a 0 (například 1 pro „zdravý“ a 0 pro „nemocný“).




**Obrázek 2 Typy proměnných**

V případě kvalitativních proměnných je jedním ze způsobů transformace převedení na binární proměnné. Tím docílíme, v případě nasazení algoritmů shlukové analýzy, možnosti použití stejného výpočtu míry vzdálenosti jako u kvantitativních dat. U nominální proměnné by transformace převedení na binární proměnné, které nazýváme indikátorové, vypadala takto:

**Tabulka 1 Transformace nominální proměnné „Auto“**

| Auto | $I_1$ | $I_2$ | $I_3$ | $I_4$ |
|---|---|---|---|---|
| Škoda ($I_1$) | 1 | 0 | 0 | 0 |
| Volkswagen ($I_2$) | 0 | 1 | 0 | 0 |
| Ford ($I_3$) | 0 | 0 | 1 | 0 |
| Audi ($I_4$) | 0 | 0 | 0 | 1 |

a u ordinální proměnné si jako příklad můžeme uvést následující tabulku:

**Tabulka 2 Transformace ordinální proměnné „Vzdálenost“**

| Vzdálenost | $X_1$ | $X_2$ |
|---|---|---|
| Malá | 0 | 0 |
| Střední | 1 | 0 |
| Velká | 1 | 1 |

jak je již na první pohled vidět z tabulky 1, tak u nominální proměnné je hammingova vzdálenost, což je počet „bitů“, ve kterých se dvě dané kategorie liší, dvou různých kategorií vždy stejná, kdežto u ordinální proměnné (viz tabulka 2) se hammingova vzdálenost dvou různých kategorií může lišit.

V případě kvantitativních proměnných se obecně využívá takzvané standardizace. Pokud nastává situace, kdy je některá proměnná daného atributu například 1000krát větší v průměru (například pokud bychom měli datový soubor s objekty osob, které by měly atributy „cena vlastněného vozidla“, „věk majitele“ nebo atribut „počet dětí“), tak je nezbytné všechny proměnné všech atributů převést na stejnou škálu hodnot. Obvykle se data transformují na interval <0,1> nebo <-1,1>. K nejpoužívanějším standardizacím patří bezesporu standardizace pomocí směrodatné odchylky a rozpětí. Standardizace pomocí směrodatné odchylky je úzce spojená se z-skorém, kde si jsou obě hodnoty rovny, takže vzorec (4) pro výpočet z-skoré je stejný jako vzorec pro výpočet standardizo-

vané hodnoty pomocí směrodatné odchylky. Standardizace pomocí rozpětí využívá minimálních a maximálních hodnot atributů a vzorec pro výpočet vypadá následovně:

$$s_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}, \tag{6}$$

kde $s_i$ je standardizovaná hodnota pomocí rozpětí, $x_i$ je hodnota ze souboru, $\min(x_i)$ je minimální hodnota vyskytující se v souboru pro daný atribut a $\max(x_i)$ je naopak maximální hodnota, která se vyskytuje v souboru pro daný atribut.

Špatným výsledkem modelování je myšlen výsledek zkreslený (zavádějící). Pro nezkreslené výsledky shlukové analýzy je tedy důležité, aby datovému souboru byl věnován dostatečný čas pro analýzu a úpravy dat popsané výše. Tyto kroky, které bychom mohli nazývat také jako čištění a příprava dat, zaberou zhruba 80 % analytikova času.

# 2 Shluková analýza

Shluková analýza (angl. cluster analysis nebo clustering) je jednou ze známých technik analýzy dat, která má využití v mnoha oborech jako je data mining, strojové učení, obrazová analýza, rozpoznávání vzorů a mnoho dalších. Hlavním cílem shlukové analýzy je vytvoření skupin neboli shluků (angl. clusters) na základě informací obsažených v datech, které popisují objekty a tím i jejich podobnost. Každý shluk tedy obsahuje objekty, u kterých je podobnost větší ve srovnání s podobností s ostatními objekty jiných shluků.

## 2.1 Metody shlukové analýzy

Než se dostaneme k samotnému rozdělení metod shlukové analýzy, tak bychom měli vědět, že existují dvě techniky učení používané v data miningu, učení s učitelem (angl. supervised learning) a učení bez učitele (angl. unsupervised learning).

V případě, že náš vstupní datový soubor obsahuje informace, které nám rozřazují objekty do skupin, pro každý objekt je příslušnost do skupiny, někdy označována jako příslušnost do třídy, známá, jedná se o učení s učitelem. Tento datový soubor nazýváme trénovací množinou, pomocí které se zvolený algoritmus naučí „předpovídat“ příslušnost ke skupině.

Shluková analýza se řadí do skupiny učení bez učitele, jelikož vstupní datový soubor neobsahuje informace o rozřazení objektů do skupin, a to z důvodu, že právě ono rozřazení objektů do skupin je cílem shlukové analýzy.

Existuje mnoho různých metod, které mohou být použity k provedení shlukové analýzy. Základním rozdělením je dělení algoritmů na hierarchické shlukování a nehierarchické shlukování.

## 2.2 Hierarchické metody

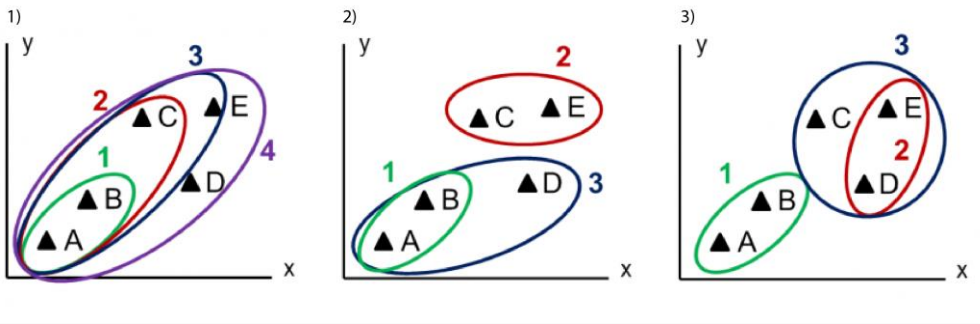
Hierarchické metody vytvářejí hierarchickou strukturu vytvořených shluků. Existují dva typy směru přístupu k této struktuře, a tak máme hierarchické metody aglomerativní a divizní.

Aglomerativní hierarchické metody při inicializaci přistupují ke každému objektu datového souboru jako k jednomu samostatnému shluku. Poté jsou shluky postupně

slučovány na základě jejich podobnosti, dokud není splněna podmínka požadovaného počtu shluků. Jednotlivé metody aglomerativního shlukování se rozlišují na základě určování podobnosti mezi jednotlivými shluky:

- Metoda nejbližšího souseda (angl. Single-link clustering) – je metoda, která určuje podobnost dvou shluků na základě nejmenší vzdálenosti kteréhokoliv jednoho objektu shluku s kterýmkoliv jedním objektem druhého shluku.
- Metoda nejvzdálenějšího souseda (angl. Complete-link clustering) – je metoda, která určuje podobnost dvou shluků na základě největší vzdálenosti kteréhokoliv jednoho objektu shluku s kterýmkoliv jedním objektem druhého shluku.
- Centroidní metoda (angl. Centroid linkage) – je metoda, u které se pro každý shluk vypočte jeho těžiště a následně je podobnost dvou shluku určena na základě nejmenší vzdálenosti kteréhokoliv jednoho těžiště shluku s kterýmkoliv jedním těžištěm druhého shluku.
- Metoda průměrné vazby (angl. Average-link clustering) – je metoda, která určuje podobnost dvou shluků na základě průměrné vzdálenosti mezi každými dvěma objekty, které patří do dvou různých shluků.

Na následujícím obrázku můžeme vidět rozdíly v jednotlivých krocích shlukování mezi metodami nejbližšího souseda, nejvzdálenějšího souseda a centroidní metodou.

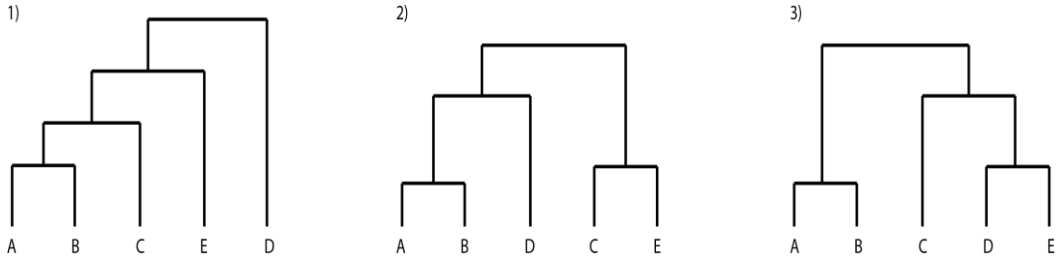



**Obrázek 3: Porovnání aglomerativních hierarchických metod - 1) metoda nejbližšího souseda, 2) metoda nejvzdálenějšího souseda a 3) centroidní metoda**

Zdroj: Hierarchické aglomerativní shlukování. *Matematická biologie* [online]. Brno: Masarykova univerzita, 2015 [cit. 2016-12-30]. Dostupné z: http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--vicerozmerne-metody-pro-analyzu-dat--shlukova-analyza--shlukova-hierarchicka-analyza--hierarchicke-shlukovani--hierarchicke-aglomerativni-shlukovani

Divizní hierarchické metody při inicializaci přistupují ke všem objektům datového souboru jako k jednomu shluku. V dalších krocích je tento shluk rozdělen na dva shluky, kde každý z těchto shluků je opět rozdělen atd. Algoritmus končí v momentě, kdy je splněna požadovaná struktura shluků.

Ke grafickému znázornění výsledků hierarchických metod se využívá binárního stromu tzv. dendrogramu. Dendrogram v jednotlivých úrovních na své ose *x* zobrazuje stupně rozkladů shluků a na ose *y* velikost vzdálenosti. Na obr. 4 jsou znázorněné dendrogramy z porovnání aglomerativních hierarchických metod na obr. 3.




**Obrázek 4 Dendrogram aglomerativních hierarchických metod - 1) metoda nejbližšího souseda, 2) metoda nejvzdálenějšího souseda a 3) centroidní metoda**

## 2.3 Nehierarchické metody

Nehierarchické metody, můžeme se také setkat s názvem metody rozkladu, jsou takové metody, které se snaží naleznout nejvhodnější rozdělení objektů do předem sta-

noveného počtu disjunktních shluků tím, že objekty přemisťují z jednoho shluku do dalšího. Jak je zmíněno výše, tak u těchto metod se často vyžaduje stanovení počtu výsledných shluků uživatelem již před samotným procesem shlukování, proto budou v kapitole 2.4 blíže popsány přístupy, které se zabývají stanovením optimálního počtu shluků. Existují metody, jako metoda k-centroidů (angl. K-means) a její modifikace, u kterých je přiřazovaní objektů ke shlukům jednoznačné, nebo metody, u kterých se počítá tzv. míra příslušnosti objektu k jednotlivým shlukům, označované jako fuzzy shlukování.

### 2.3.1 Metoda k-centroidů

Metoda k-centroidů je nejznámější nehierarchickou metodou ve shlukové analýze. Omezujícím kritériem metody je v možnosti využití na datovém souboru, který obsahuje pouze kvantitativní proměnné (číselné hodnoty). Algoritmus rozděluje objekty do $k$ shluků ($C_1$, $C_2$, …, $C_k$), které jsou reprezentovány jejich centroidem (těžištěm). Centroid je pro každý shluk vypočítán jako průměr všech objektů daného shluku. K výpočtu vzdáleností mezi jednotlivými objekty se využívá Euklidovské metriky. Pokud bychom k výpočtu vzdáleností mezi jednotlivými objekty využili například Manhattanské metriky, jednalo by se o metodu k-mediánů.

Základním, a také velmi ovlivňujícím, krokem metody k-centroidů je počáteční volba $k$ centroidů. Můžeme například zvolit $k$ prvních objektů datového souboru, v datovém souboru vytvořit nové objekty, které reprezentují $k$ centroidů nebo náhodně zvolit $k$ objektů z datového souboru. Počáteční volba $k$ centroidů dokáže ve velké míře ovlivnit konečný výsledek shlukovacího procesu, a proto by se metoda k-centroidů měla vždy provádět v několika pokusech s jinými objekty zvolenými jako centroidy. V dalším kroku metody k-centroidů se každý objekt datového souboru přiřadí ke shluku, k jehož centroidu má nejblíže. Na základě tohoto přiřazení dochází k přepočítání $k$ centroidů. Celý proces se opakuje, dokud není splněna podmínka požadované kvality shluků nebo do té doby, kdy není uskutečněno žádné další přemístění objektu ze shluku do shluku.

Metoda k-centroidů je citlivá na šum v datech (chybějící nebo špatné údaje) a odlehlé hodnoty (*outliers*). Tím, že je centroid počítán jako aritmetický průměr přes všechny objekty shluku, tak jeden jediný outlier nám dokáže velmi „zahýbat“ s výsledným rozložením shluků.

Pokud bychom chtěli minimalizovat citlivost na odlehlé hodnoty, tak použijeme metodu k-medoidů známou také pod zkratkou PAM (angl. Partition Around Medoids). Zmíněný algoritmus je velmi podobný algoritmu k-centroidů, liší se pouze v tom, jak jsou shluky reprezentovány. Namísto centroidů jsou shluky reprezentovány tzv. medoidy. Medoid je konkrétní objekt ze shluku, který je umístěný nejcentrálněji v daném shluku.

### 2.3.2 Fuzzy shluková analýza

U jiných metod nehierarchického shlukování objekt do shluku buď patří, nebo nepatří. Fuzzy shluková analýza se liší od ostatních metod nehierarchického shlukování v tom, že objektu datového souboru přiděluje tzv. míru příslušnosti. Míra příslušnosti je v rozmezí <0,1>, kde v případě nulové míry příslušnosti objekt určitě nepatří do daného shluku, naopak 1 je myšleno jednoznačné přiřazení do shluku. Fuzzy shluková analýza má buď probabilistický, nebo posibilistický přístup. Přístupy jsou si podobné, oba přiřazují míru příslušnosti v rozmezí <0,1>, ale rozlišují se v tom, že probabilistický přístup je omezen podmínkou, že součet měr příslušností určitého objektu ke všem shlukům se musí rovnat jedné, kdežto posibilistický přístup touto podmínkou limitován není. Vyšší míra příslušnosti indikuje vyšší jistotu v přiřazení do shluku.

Za nejznámější fuzzy shlukovací algoritmus se považuje fuzzy c-means, u kterého je hlavním cílem minimalizovat funkcionál:

$$J_m = \sum_{i=1}^{n}\sum_{j=1}^{c}(u_{ij})^m d^2(x_i, v_j), \tag{7}$$

kde $u_{ij}$ je míra příslušnosti $i$-tého objektu k $j$-tému shluku, $m$ je fuzzyfikační konstanta a $d$ Euklidovská vzdálenost mezi $x_i$ $i$-tým objektem a $v_j$ $j$-tým středem shluku, o vzdálenosti bude více v kapitole 2.5.

Kroky algoritmu fuzzy c-means jsou následující:

1. Náhodné zvolení $c$ středů shluků z datového souboru.
2. Výpočet míry příslušnosti $u_{ij}$:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d(x_i, v_j)}{d(x_i, v_k)}\right)^{\left(\frac{2}{m-1}\right)}}. \tag{8}$$

3. Výpočet středů shluků $v_j$:

$$v_j = \frac{(\sum_{i=1}^{n} (u_{ij})^m x_i)}{\sum_{i=1}^{n} (u_{ij})^m)}. \quad (9)$$

4. Dokud není dosažena minimální hodnota funkcionálu *J* nebo pokud není rozdíl mezi dvěma posledními maticemi míry příslušnosti menší než požadovaná přesnost, tak opakujeme kroky 2. a 3.

## 2.4 Stanovení optimálního počtu shluků

Jak již bylo zmíněno v kapitole 2.3, tak u nehierarchických metod typu k-centroidů, k-mediánů, k-medoidů atd. je od uživatele vyžadováno zadání konkrétního počtu požadovaných výsledných shluků *k*. V některých případech je počet shluků předem „čitelný“ z dané úlohy. V ostatních případech, kdy počet shluků *k* není předem „čitelný“, je zapotřebí použití některého z následujících přístupů.

Jednoduchým řešením je využití dendrogramu, který se vyskytuje u grafického znázornění výsledků hierarchického shlukování. V dendrogramu mohou být naznačeny výraznější shluky, ale spíše se jedná o subjektivní přístup.

Přístupy pro stanovení optimálního počtu shluků rozdělil A. D. Gordon v knize [9] do dvou kategorií: globální metody a lokální metody. Globální metody vyhodnocují míru správného rozdělení do *k* shluků *G(k)*, které jsou obvykle založeny na vnitro-shlukové nebo mezi-shlukové variabilitě. Bohužel u globálních metod neexistuje definice pro *G(1)*, a proto nám tyto metody nedokáží pomoci v momentě, kdy by datový soubor měl být rozdělen na méně jak dva shluky, tzn. nerozdělen vůbec. Lokální metody zkoumají, zdali by měla být dvojice shluků sloučena či nikoliv, popřípadě jestli by měl být jeden shluk rozdělen na dva. Nevýhodou u lokálních metod je nutnost specifikace prahové hodnoty, která je závislá na (neznámých) vlastnostech datového souboru.

V knize [9] jsou uvedeny tři globální metody a dvě lokální metody, které si ve studii Milligan and Cooper (1985) vedly nejlépe:

- **Calinski-Harabaszův index**

$$G1(k) = \frac{\left[\frac{B}{(k-1)}\right]}{\left[\frac{W}{(n-k)}\right]}, \tag{10}$$

kde $B$ je mezi-shluková variabilita (součet čtvercových vzdáleností jednotlivých centroidů od centroidu všech objektů), $W$ je vnitro-shluková variabilita (součet čtvercových vzdáleností jednotlivých objektů od svých centroidů), $k$ je počet shluků a $n$ je počet objektů.

- **Baker-Hubertův index**

$$G2(k) = \frac{(S_+ - S_-)}{(S_+ + S_-)}, \tag{11}$$

kde $S_+$ je počet konkordantních srovnání a $S_-$ je počet diskordantních srovnaní. Srovnání se považuje za konkordatní (resp. diskordantní), pokud je vnitro-shluková nepodobnost menší (resp. větší) než mezi-shluková nepodobnost.

- **C-index**

$$G3(k) = \frac{(D(k) - D_{\min})}{(D_{\max} - D_{\min})}, \tag{12}$$

kde $D(k)$ je součet všech vnitro-shlukových nepodobností při rozdělení objektů do $k$ shluků, $D_{min}$ je definován jako součet minimálních nepodobností a $D_{max}$ naopak jako součet maximálních nepodobností.

V případě indexů $G1(k)$ a $G2(k)$ platí, že čím vyšší je vypočtená hodnota, tím lepší je rozdělení do těchto $k$ shluků. Naopak u indexu $G3(k)$ platí, že čím nižší je vypočtená hodnota, tím je rozdělení lepší.

- **Duda-Hartův index**

$$W_2 / W_1 < \frac{1-2}{(\pi * p)} - z * \sqrt{2 * \left(1 - \frac{8}{(\pi^2 * p)}\right)}{m * p}, \tag{13}$$

kde $W_1$ je vnitro-shlukový součet čtvercových vzdáleností, při optimálním rozdělení daného shluku na dva a součtem jejich vnitro-shlukových vzdáleností získáme

proměnnou $W_2$, $z$ je standardní z-skóre, které určuje významnost prováděného testu (v [10] je uvedeno, že nejlepších výsledků je dosaženo, pokud je hodnota nastavena na 3.20), $m$ je počet objektů ve shluku a $p$ je počet kvantitativních proměnných popisujících tyto objekty. Pokud je výše zmíněná podmínka (13) splněna, tak se jedná o homogenní shluk a neměl by být rozdělen.

- **Bealeův index**

$$F = \frac{(\frac{W_1 - W_2}{W_2})}{((\frac{m-1}{m-2}) * 2^{\frac{2}{p}} - 1)}, \tag{14}$$

kde popis proměnných $W_1$, $W_2$, $m$ a $p$ je stejný jako v případě Duda-Hartova indexu. Pokud je hodnota $F$ větší než kritická hodnota z distribuční funkce $F_{p,(m-2)p}$, tak se nejedná o homogenní shluk a měl by být rozdělen.

## 2.5 Míra podobnosti

Určení podobnosti objektů je klíčovým prvkem shlukové analýzy. Jak již bylo zmíněno výše, tak na základě informací obsažených v datech můžeme provést samotné rozdělení objektů do shluků. Tzn., že každý objekt obsahuje atributy, které daný objekt charakterizují. Například člověk je charakterizován pohlavím nebo barvou očí, mobilní telefon rozlišením displeje, automobil obsahem motoru, typem paliva, počtem sedadel atd.

Atributy mohou být různého typu: kvalitativní, kvantitativní nebo dichotomická. Kvalitativní data se rozdělují na ordinální a nominální, kde ordinální data jsou specifické tím, že se dají seřadit a nominální data seřadit nelze. Kvantitativní data jsou vyjádřena číselnou hodnotou a dichotomická data mohou nabývat pouze dvou hodnot.

Shlukování je v podstatě seskupování podobných objektů, a tak je nutné vědět, jak vůbec určit, zda jsou dva objekty podobné či nepodobné. Existuje několik metod, jak míru podobnosti vypočítat. Ve velké míře závisí na tom, u jakého typu dat chceme míru podobnosti počítat.

Pro míru podobnosti platí čtyři pravidla. Prvním pravidlem je, že míra podobnosti mezi dvěma objekty nesmí být záporné hodnoty. Druhým pravidlem je, že míra podobnosti mezi objektem $A$ a $B$ se rovná míře podobnosti mezi objektem $B$ a $A$. Třetím

pravidlem je, že míra podobnosti dvou stejných objektů je rovna 0 a posledním pravidlem je trojúhelníková nerovnost. Pokud budeme mít testovací objekty *A, B, C* a míru podobnosti *d*, tak pravidla jsou:

1. Nezápornost – $d(A, B) \geq 0$.
2. Symetrie – $d(A, B) = d(B, A)$.
3. Totožnost – $d(A, A) = 0$.
4. Trojúhelníková nerovnost – $d(A, B) \leq d(A, C) + d(B, C)$.

### 2.5.1 Kvantitativní data

U kvantitativních dat se namísto míry podobnosti setkáváme s pojmem míra vzdálenosti. Míry vzdálenosti se používají v momentě, kdy máme *n*-rozměrné objekty a jednotlivé proměnné (atributy) jsou číselné. Objekt může být chápan jako vektor n-rozměrného vektorového prostoru $V^n$, kde hodnoty jednotlivých proměnných určují jeho souřadnice. Uvažujme, že máme objekt $A = [A_1, A_2, A_3, \ldots, A_n]$, objekt $B = [B_1, B_2, B_3, \ldots, B_n]$ pro $n \in \mathrm{N}$ a předpis $d(A, B)$ pro určení „vzdálenosti“ resp. míry vzdálenosti objektů *A* a *B*. Pokud jsou pro $d(A, B)$ splněny výše uvedená pravidla, pak mluvíme o metrice.

Neznámějšími typy míry vzdálenosti je Manhattanská vzdálenost $d_M$, čtvercová Euklidovská vzdálenost $d_E$ a Čebyševova vzdálenost $d_Č$. Všechny tyto míry vzdálenosti vycházejí z Minkowského metriky $d_{Mi}$, která je vyjádřena vzorcem:

$$d_{Mi}(A,B) = \sqrt[g]{\sum_{i=1}^{n} (|A_i - B_i|)^g}\ , \tag{15}$$

kde *A* a *B* jsou objekty z datového souboru, které mají *n* proměnných, viz výše. Dále se ve vzorci vyskytuje proměnná *g*. Proměnná *g* nás přivádí ke zmiňovaným vzdálenostem výše, které jsou v Minkowského metrice speciálními případy, pro $g = 1$ dostáváme Manhattanskou vzdálenost, pro $g = 2$ dostáváme Euklidovskou vzdálenost a pro $g \to \infty$ Čebyševovu vzdálenost. Vzdálenosti jsou vyjádřeny následujícími vzorci:

$$d_M(A,B) = \sum_{i=1}^{n} |A_i - B_i|, \tag{16}$$

$$d_E(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}\ , \tag{17}$$

$$d_{\check{C}}(A,B) = \max_{1 \le i \le n}(|A_i - B_i|) . \quad (18)$$

### 2.5.2 Binární data

V případě binárních dat jsou atributy objektů v datovém souboru popsány pouze „jedničkami“ a „nulami“. Pokud takto popsané nejsou, ale nabývají pouze dvou hodnot, tak si je na „jedničky“ a „nuly“ můžeme převést (například muž/žena, den/noc, zaměstnaný/nezaměstnaný atd.). Rozlišujeme symetrické binární atributy a asymetrické binární atributy.

Binární atributy jsou symetrické, jestliže obě hodnoty 0 i 1 mají stejnou váhu důležitosti atributu. Jinými slovy ani u jedné varianty nejsme schopni určit, která má pro nás větší váhu (například muž/žena). Pokud se objekty skládají ze symetrických binárních atributů, tak se k výpočtu míry podobnosti používá například tzv. koeficient prosté shody $d_{SM}$ (angl. simple matching coefficient):

$$d_{SM}(A,B) = \frac{a+d}{a+b+c+d}, \quad (19)$$

kde $a$ je počet případů, kde se stejné atributy z obou objektů rovnají „jedničce“, $d$ je počet případů, kde se stejné atributy z obou objektů rovnají „nule“, $b$ a $c$ je počet případů, kde se hodnoty atributů u obou objektů liší. Přehlednou popisující tabulku jednotlivých proměnných můžeme vidět v tabulce Tabulka 3.

Binární atributy jsou asymetrické, jestliže jedna hodnota má větší váhu oproti druhé hodnotě. Obvykle pozitivní výstup atributu je považován za důležitější, například pravda/lež nebo účast/absence atd. Pro asymetrické binární atributy se tedy nezohledňuje proměnná $d$ (počet shod negativních výsledků). Pokud se objekty skládají z asymetrických binárních atributů, tak se k výpočtu míry podobnosti používá například tzv. Jaccardův koeficient $d_J$, Diceův koeficient $d_D$ a Russelův-Raovův koeficient $d_{RR}$:

$$d_J(A,B) = \frac{a}{a+b+c}, \quad (20)$$

$$d_D(A,B) = \frac{2*a}{2*a+b+c}, \quad (21)$$

$$d_{RR}(A,B) = \frac{a}{a+b+c+d} . \quad (22)$$

Následující tabulka slouží jako přehledný popis zmiňovaných proměnných *a*, *b*, *c* a *d*:

**Tabulka 3 Kontingenční tabulka binárních dat**

| | | Objekt *B* | |
|---|---|---|---|
| | | 1 | 0 |
| Objekt *A* | 1 | *a* | *b* |
| | 0 | *c* | *d* |

### 2.5.3 Kvalitativní data

Rozlišujeme dva druhy kvalitativních dat, nominální data a ordinální data. U nominálních dat neexistuje pojem „vzdálenost“. Nominální data jsou buď shodné, nebo odlišné. Z těchto důvodů se v případě určení míry podobnosti nominálních dat používá pouze koeficient prosté shody $d_{SMN}$. Jedná se o podíl součtu shodných hodnot atributů na celkovém počtu všech hodnot, tj.:

$$d_{SMN}(A,B) = \frac{a+d}{a+b+c+d}, \tag{23}$$

kde jednotlivé proměnné jsou popsány v předchozí tabulce 3. Popřípadě je možné převést nominální data na skupiny pomocných binárních hodnot, jak je uvedeno v kapitole 1. Poté již můžeme použít některou z měr podobnosti pro binární data, které jsou uvedeny v kapitole 2.5.2.

Ordinální data jsou specifická tím, že se dají seřadit, ale nedokážeme říci, o kolik je jedna hodnota větší či menší od té druhé. Ordinální data můžeme přepočítat na kvantitativní hodnoty. Pro daný proces je nutné, aby každé ordinální hodnotě bylo přiřazeno její číselné pořadí $x_{ij} \in 1, 2, 3, \ldots, X_j$ (například: malá – 1, střední – 2, velká – 3) a dále vypočítat tzv. standardizovanou hodnotu pomocí následujícího vzorce:

$$s_{ij} = \frac{x_{ij} - 1}{X_j - 1}, \tag{24}$$

kde $s_{ij}$ je standardizovanou hodnotou objektu $i$ a atributu $j$, $x_{ij}$ je číselné pořadí ordinální hodnoty a $X_j$ je maximální hodnota číselného pořadí pro danou ordinální hodnotu.

# 3 Funkcionál kvality rozkladu

Určení, zdali je shluk kvalitní či nikoliv, je problematické a kontroverzní téma, protože neexistuje univerzální definice kvalitního shluku [12]. V knize [13] autor píše, že určení kvality shluku je většinou v oku „diváka“. Věta je z části pravdivá, ale v mém úsudku dokáže „divák“ určit kvalitu shluku pouze u menších datových souborů, a i to je problematické, protože zobrazit či představit si n-rozměrný prostor není dost dobře možné. U větších datových souborů, k určení kvality shluku nebo celého rozkladu, je vyžadována podpora objektivních metod. Tyto metody se obvykle dělí do dvou kategorií na vnitřní a vnější, kde vnitřní metody počítají pouze s výsledkem daného shlukovacího procesu, kdežto vnější metody počítají s předem známou strukturou shluků a objektů do nich zařazených.

## 3.1 Vnitřní metody

Cílem shlukové analýzy je, aby si objekty v jednom shluku byly podobné a objekty mezi shluky byly odlišné. Proto se v určování kvality pomocí vnitřních metod setkáváme s pojmy *homogenita* a *separace*. Homogenita určuje, jak jsou si objekty v jednom shluku podobné, kde nižší hodnota vypovídá o lepší homogenitě shluku. Separace určuje, jak „dobře“ jsou shluky mezi sebou oddělené a vyšší hodnota vypovídá o lepší separaci shluků. Kvalita shluku se pomocí vnitřních metod převážně určuje na základě toho, jak jsou objekty ve shluku daleko od sebe, tzn. na základě vzdálenosti. Vnitřní metody, na rozdíl od vnějších metod, využívají pouze informací v daném shlukovacím procesu. V následujících kapitolách bude blíže popsána metoda součtu čtvercových chyb, silueta a Daviesův-Bouldinův validační index.

### 3.1.1 Součet čtvercových chyb

Součet čtvercových chyb je součet čtvercových rozdílů (vzdáleností) mezi objektem shluku a jeho centroidem. Jedná se o nejrozšířenější a nejjednodušší metodu pro vypočet kvality shluku či kvality rozkladu. Pokud by byl každý objekt zároveň i shlukem, tak by výsledný součet čtvercových chyb byl roven 0 a to jak v případě výpočtu kvality shluku, tak i v případě výpočtu kvality celého rozkladu.

Pro výpočet kvality shluku použijeme vzorec:

$$SSE = \sum_{i=1}^{n} d(x_i, c)^2 , \tag{25}$$

kde $n$ je počet objektů shluku a $d$ je funkce pro výpočet vzdálenosti mezi objekty $x_i$ a $c$ ($x_i$ je $i$-tý objekt shluku a $c$ je centroid daného shluku).

Pro výpočet kvality rozkladu použijeme vzorec:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d(x, c_i)^2 , \tag{26}$$

kde $K$ je počet shluků a $C_i$ je množina objektů ve shluku $i$.

### 3.1.2 Silueta

Silueta skvěle kombinuje homogenitu a separaci. Dále je vhodnou metodou pro grafickou interpretaci, jak „dobře“ je objekt umístěn v daném shluku. Hodnota siluety je v rozmezí od -1 do 1, kde vyšší hodnota indikuje lepší umístění v daném shluku, a naopak nižší hodnota indikuje, že objekt by v daném shluku být neměl. Pokud je hodnota siluety blízká nule, tak je indikována možnost zařazení objektu do jiného shluku. V případě celkové průměrné hodnoty siluety již vyhodnocujeme kvalitu rozkladu, a čím je hodnota vyšší, tak tím je výsledek shlukovacího procesu lepší. V takovém případě se provádí shlukovací algoritmus v několika variantách (rozdílné počty shluků) a hledá se varianta, kde je celková průměrná hodnota siluety největší.

Pro výpočet siluety pro objekt $i$ použijeme vzorec:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{27}$$

kde $a_i$ je průměrná vzdálenost $i$-tého objektu od ostatních objektů ve stejném shluku, $b_i$ je minimální průměrná vzdálenost $i$-tého objektu od objektů v jiných shlucích.

Pro výpočet celkové průměrné hodnoty siluety použijeme vzorec:

$$S = \frac{1}{n} \sum_{i=1}^{n} S_i , \tag{28}$$

kde $n$ je počet objektů v datovém souboru a $S_i$ je hodnota siluety pro $i$-tý objekt v datovém souboru.

### 3.1.3 Daviesův-Bouldinův validační index

Další metodou, která slouží pro posuzování kvality shlukovacího procesu, je Daviesův-Bouldinův validační index. Stejně jako u siluety, tak i zde je kombinace, jak vnitro-shlukových vzdáleností, tak i separace mezi jednotlivými shluky. V případě tohoto indexu platí, čím menší je jeho hodnota, tím lepší je homogenita a separace shluků.

Pro výpočet Daviesova-Bouldinova indexu použijeme vzorec:

$$DB = \frac{1}{K}\sum_{i=1}^{K} \max_{j=1,\ldots,k, i\neq j} \left( \frac{\frac{1}{n_i}\sum_{x\in C_i} d(x, c_i) + \frac{1}{n_j}\sum_{x\in C_j} d(x, c_j)}{d(c_i, c_j)} \right), \tag{29}$$

kde $K$ je počet shluků, $n_i$ ($n_j$) je počet bodů v $i$-tém ($j$-tém) shluku, $c_i$ ($c_j$) je centroid $i$-tého ($j$-tého) shluku, $C_i$ ($C_j$) je množina objektů ve shluku $i$ ($j$) a $d$ je funkce pro výpočet vzdálenosti mezi danými dvěma objekty.

## 3.2 Vnější metody

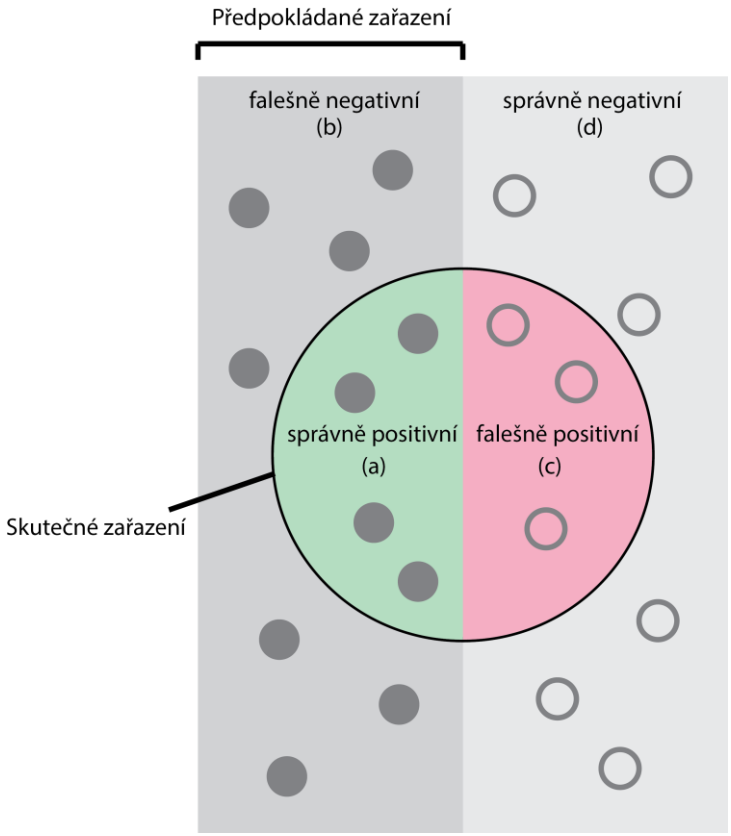
Vnější metody posuzují, z hlediska kvality rozkladu, jak si je výsledek shlukovacího algoritmu podobný s předpokládaným rozdělením. To znamená, že vnější metody mohou být použity v případě, kdy víme nebo očekáváme, do kterých shluků budou objekty zařazeny, a předpokládané rozdělení si dle toho vytvoříme. Možnou variantou je také použití vnějších metod na porovnání dvou, na sobě nezávislých, shlukovacích procesů a jejich rozkladů. U vnějších metod není třeba znát popisy shluků a mohou tak být použité na jakýkoliv shlukovací algoritmus. V následujících podkapitolách budou blíže popsány vnější metody F-míra, Randův index a Folkesův-Mallowsův index pro jednotlivý shluk, které lze následně použít pro hodnocení kvality rozkladu.

### 3.2.1 F-míra

F-míra pro shluk ve svém výpočtu pracuje s proměnnými přesnost $p$ (angl. precision) a úplnost $r$ (angl. recall), kde $p = \frac{a}{a+b}$ a $r = \frac{a}{a+c}$, $a$, $b$, $c$ dle obr. 5.

Pro výpočet f-míry použijeme vzorec:

$$F = \frac{2 * p * r}{p + r} \tag{30}$$




**Obrázek 5 Klasifikace zařazení objektů**

Hodnota f-míry je v rozmezí od 0 do 1, kde přesná shoda, mezi výsledkem shlukovacího algoritmu a námi předpokládaným rozdělením, je určena hodnotu 1 a naopak žádná shoda je určena hodnotou 0.

### 3.2.2 Randův index

Randův index, jako jeden z mála indexů pro výpočet kvality rozkladu, počítá také s proměnnou *d*, která je popsána v obr. 5. Hodnota Randova indexu je v rozmezí od 0 do 1, kde přesná shoda, mezi výsledkem shlukovacího algoritmu a námi předpokládaným rozdělením, je určena hodnotou 1, a naopak hodnota 0 určuje žádnou shodu.

Pro výpočet Randova indexu použijeme vzorec:

$$R = \frac{a+d}{a+b+c+d}. \tag{31}$$

### 3.2.3 Folkesův-Mallowsův index

Folkesův-Mallowsův index je v odborných článcích interpretován jako geometrický průměr. Použité proměnné jsou popsané na obr. 5. Hodnota tohoto indexu je stejná jako v předchozích případech v rozmezí od 0 do 1, kde hodnota 1 určuje přesnou shodu, a naopak žádná shoda je určena hodnotou 0.

Pro výpočet Folkesova-Mallowsova indexu použijeme vzorec:

$$FM = \sqrt{\frac{a}{a+b} * \frac{a}{a+c}}\,. \tag{32}$$

# 4 Využití shlukové analýzy v praxi

V tom, jak jako lidé analyzujeme a popisujeme svět, hrají důležitou roli třídy nebo koncepčně ucelené skupiny objektů, které sdílejí společné charakteristiky. Rozdělovat objekty do skupin anebo přiřazovat jednotlivé objekty do těchto skupin patří mezi naše silné stránky. Například základní rozdělení živočichů na obratlovce a neobratlovce, kdy se obě skupiny vyznačují určitými charakteristikami. Na základě naší povahy tedy existuje pro shlukovou analýzu využití v mnoha oborech. Data miningové úlohy, které využívají metod shlukové analýzy, jsou například predikce odchodu zákazníků, segmentace zákaznického trhu, rozpoznávání komunit na sociálních sítích, detekce podvodů, prevence kriminality a mnoho dalších. Často je shluková analýza částí data miningového projektu a na ní navazují další algoritmy. V přípravě dat dobře identifikuje vzdálené málo četné shluky, bývá často prvním krokem nesupervizovaného modelování a její závěry jsou poté vstupem do následného supervizovaného modelování.

Shluková analýza se využívá v souborech z mnoha různých oborů. Jejich výčet přesahuje rozsah zadání, proto uvedu jenom několik příkladů.

## 4.1 Predikce odchodu zákazníka

Jedním velmi známým využitím metod shlukové analýzy je při analýze zákazníků a jejich postavení ke společnosti z hlediska možného vypovězení smlouvy. Tato technika je známá pod anglickým názvem „churn prediction“ a uplatňuje se v momentě, kdy mají zákazníci smlouvu, buď na dobu určitou, nebo dobu neurčitou a každý měsíc platí předem stanovenou částku za poskytované služby. Nejvíce se predikce odchodu zákazníků týká společností, které podnikají v oboru bankovnictví, telekomunikací či pojištění. Tyto společnosti se k predikování odchodu zákazníka dostávají z důvodů vysoké konkurence a nasyceného trhu. Společnosti musí včas reagovat, aby si zákazníky udrželi. Pokud by společnosti dokázaly predikovat odchod zákazníka a poté mu v takovém kroku dokázali zabránit lákavou promyšlenou nabídkou, tak by toto úsilí společnosti ušetřilo spousty vynaložené práce a hlavně peněz, protože je obecně známo, že udržení stávajícího zákazníka je daleko „levnější“, než získání nového zákazníka.

Churn prediction je v mnoha odborných článcích nejčastěji spojován s telekomunikačními společnostmi, a to z toho důvodu, že telekomunikační společnosti

mají velké množství dat o svých zákaznících. V datech se vyskytují jak záznamy o stávajících zákaznících, tak i o zákaznících, kteří již od společnosti odešli. Především data o zákaznících, kteří již od společnosti odešli, nám poskytují největší službu v predikci odchodu u stávajících zákazníků, protože v sobě skrývají potenciální důvod odchodu. Všichni zákazníci pak mohou být rozděleni do jednotlivých segmentů dle pravděpodobnosti odchodu. U zákazníků s vyšší pravděpodobností odchodu se poté realizují vhodné retenční kampaně, které vedou k jejich udržení. Přínosem analýzy predikce odchodu zákazníků je bezpochyby snížení úrovně odchodu zákazníků, ale také jako společnost zjistíme, které faktory ovlivňují odchod našich zákazníků.

## 4.2 Detekce podvodů

Detekce podvodů je téma, které se vztahuje na mnoho odvětví podnikání. Bankovnictví, pojišťovnictví, telekomunikace, maloobchodní prodej, internetové transakce nebo výběr daní, to je pouze hrstka odvětví, které se potýkají s podvody. V posledních letech se detekce podvodů stala nedílnou součástí bank a pojišťoven. Za pomocí sofistikovaných data miningových nástrojů (jako je shluková analýza) může být v deseti tisících transakcí (v případě bankovních institucí) nalezen určitý vzor podezřelého chování a tím mohou být odhaleny podvodné transakce.

Úloha detekování podvodů může být spojena s hledáním neobvyklého chování či detekcí anomálií. Algoritmy shlukové analýzy umí vyhledat shluky se společnou charakteristikou, ale interpretace shluků je na dataminerovi a „doménovém“ spolupracovníkovi. Ve shlukové analýze pro identifikaci odlehlých hodnot (*outliers*), kde odlehlost není spojena s jedním atributem, ale objektem/y s více (obecně n) atributy, lze použít například algoritmus k-centroidů. Tzn., že hledáme objekty, které vybočují z „normálního chování“ všech ostatních objektů. Tyto objekty mohou tvořit nápadně málo četný shluk nebo v jednom shluku mají nápadně velkou vzdálenost od centroidu.

## 4.3 Segmentace zákazníků

Každá společnost, která má větší množství zákazníků, by se měla snažit segmentovat své zákazníky, tzn. zařadit je do skupiny zákazníků s podobnými nebo souvisejícími vlastnostmi. Zákazníci mohou být takto rozřazeni na základě jejich pohlaví, věku, příjmu, stavu, zaměstnání a mnoho dalších osobních kritérií, které má společnost od zákazníků k dispozici, ale také na základě jejich chování u dané společnosti (například podle jejich útrat či frekvencí nákupů) nebo výzkumu trhu.

Rozdělením zákazníků do segmentů se dostává společnost do lepší pozice na trhu a to především tím, že dokáže vytvořit obchodní kampaně, které přesně odpovídají požadavkům jednotlivých zákaznických segmentů, a tím oslovit cílový segment s vyšší účinností. Z toho vyplývá i finanční úspora na vynaložené obchodní kampaně a zlepšení komunikace se zákazníky.

## 4.4 Aplikace shlukové analýzy při dotazníkových šetřeních

Dotazníková šetření sbírají obvykle velké množství kategoriálních dat, které je potřeba analyticky zpracovat. V těchto datech se ověřují známé resp. předpokládané souvislosti, ale hledají se i informace nové. Algoritmy shlukové analýzy jsou jedním z možných přístupů. O problémech určení optimálního počtu shluků, o přístupech k hodnocení shluků pro dotazníková šetření se dá najít mnoho publikací například [19].

Dalším problémem spojeným s daty z dotazníkových šetření je redukce dimenzionality. Jedná se o to, že vyhledáním podobných segmentů lze analyzovat také vztahy mezi proměnnými, z několika stávajících proměnných po té odvodit proměnnou novou, která původní proměnné nahradí. Při redukci dimenzionality se projektuje vícedimenzionální prostor na prostor s méně dimenzemi při zachování maximální variability dat. Statistických postupů pro redukci dimenzionality v kategoriálních datech je mnoho. Shluková analýza je pouze jednou z možností.

## 4.5 Medicínský výzkum

Počet lékařských záznamů v odvětví medicíny roste každým dnem. Záznamy o nemocích mohou obsahovat příznaky či projevy nemoci, druh podávaných léčiv, roční období, zdravotní stav, informace o diagnózách a mnoho dalších. Takové záznamy, resp. data po nasazení data minigových algoritmů a postupů pomáhají lékařům například ve specifikaci nemocí, s určováním léčby, předepisováním léčiv, ale i v prevenci před nemocemi. V medicínských úlohách je důležité vycházet z důvěryhodných dat, které dlouho nebyly k dispozici. Situace se v posledních létech změnila a dnes je již dostupné velké množství dat, na kterých mohou být data miningové úlohy prováděny. Velké množství historických dat o "jiných" pacientech vede k poznání dopadů léčebných postupů na podobné "shluky" pacientů a před nasazením dané léčby či léků konkrétnímu pacientovi je toto důležitým prvkem rozhodování lékaře o nasazení léčby. Shluková analýza může vést k zisku klíčových informací o nemocích či pacientech nebo objevení souvislostí mezi jednotlivými atributy nemoci a pacienta.

# 5 Popis aplikace MyCluster

Tato část diplomové práce se zabývá podrobným popisem mnou vytvořené aplikace nazvanou MyCluster pro shlukování objektů ve dvourozměrném prostoru. Popsané algoritmy nejbližšího souseda, nejvzdálenějšího souseda, centroidní metoda a metoda k-centroidů v teoretické části diplomové práce jsou naprogramovány v dané aplikaci. V aplikaci je také naprogramován výpočet kvality shlukování pomocí celkové průměrné siluety (obrysového koeficientu). Aplikace byla vytvořena jako učební pomůcka pro studenty a především pro předmět Datamining, se kterým se studenti na fakultě mechatroniky, informatiky a mezioborových studií mohou setkat v navazujícím studiu v oboru Informační Technologie. Studenti by za pomoci aplikace MyCluster měli lépe a snadněji porozumět vybraným algoritmům shlukové analýzy, a to například pomocí vhodné vizualizace (textová, spojnicová nebo animační viz následující kapitola) jednotlivých kroků algoritmů či možností aplikovat odlišné algoritmy shlukové analýzy pro stejný datový soubor a tyto možnosti porovnávat pomocí zobrazení předešlých výsledků shlukování, nebo možností výběru z různých měr vzdáleností. Aplikace je naprogramována pomocí programovacího jazyka C# a bylo použito vývojové prostředí Microsoft Visual Studio, které mám jako student Technické Univerzity v Liberci přístupný zdarma v rámci programu DreamSpark.

Kapitola se zabývá podrobným popisem jednotlivých funkčních či zobrazovacích prvků použitých ve vytvořené aplikaci, vybraných algoritmů shlukové analýzy a použitých měr vzdáleností. Aplikace je vytvořena pro interaktivní experimentování a případné porovnání výsledků v IBM SPSS Modeleru, který je v rámci předmětu Datamining používaný.

## 5.1 Prvky aplikace

Z velké části, jak je vidět na obr. 8 Grafické uživatelské rozhraní aplikace MyCluster, je aplikace tvořena nad komponentou Panel. Část plochy Panelu je věnována interaktivní pracovní grafické ploše, kde jsou vykresleny osy x a y, které nám umožnují jednoznačné popsání polohy bodů vytvořených kliknutím na grafickou plochu. Délkovou jednotkou je jeden pixel daného rozlišení a pro lepší orientaci jsou na obou osách x a y vyznačeny úseky délkové jednotky po 50 pixelech. Je zde také možnost vykreslení svislého a vodorovného rastru, který dopomáhá orientaci v prostoru. Pracovní

plochou je první kvadrant, což pro vizualizaci zcela postačuje. Zadávané body a jejich souřadnice se snadno vizuálně kontrolují.

Jak je vidět na obr. 6, tak vykreslení grafického rozhraní na komponentě Panel je vyvoláno při události Paint a skládá se z několika metod, které svým pojmenováním jednoznačně popisují svou podstatu.

```
1 private void panel1_Paint(object sender, PaintEventArgs e)
2 {
3   NakresliOsy();
4   if (checkBox1.Checked) { VodorovnyRastr(); }
5   if (checkBox2.Checked) { SvislyRastr(); }
6   PopisOsy();
7   VykresliBody();
8 }
```

**Obrázek 6 Ukázka kódu vykreslování soustavy souřadnic a bodů**

Ve výše zmíněné části grafického rozhraní má uživatel možnost, pomocí kliknutí levého tlačítka myši, vytvořit bod s pořadovým číslem o daných souřadnicích *x* a *y*. Takto „naklikané“ body se vypisují do komponenty ListBox v pravé části aplikace ve tvaru „Bod 1: x = 193 y = 178“. Body se ukládají do Listu třídy *TBody*, kde třída *TBody* obsahuje číselné proměnné *cisloBodu* a *cisloShluku*, souřadnicovou proměnnou *poloha* a booleovskou proměnnou *shluk*.

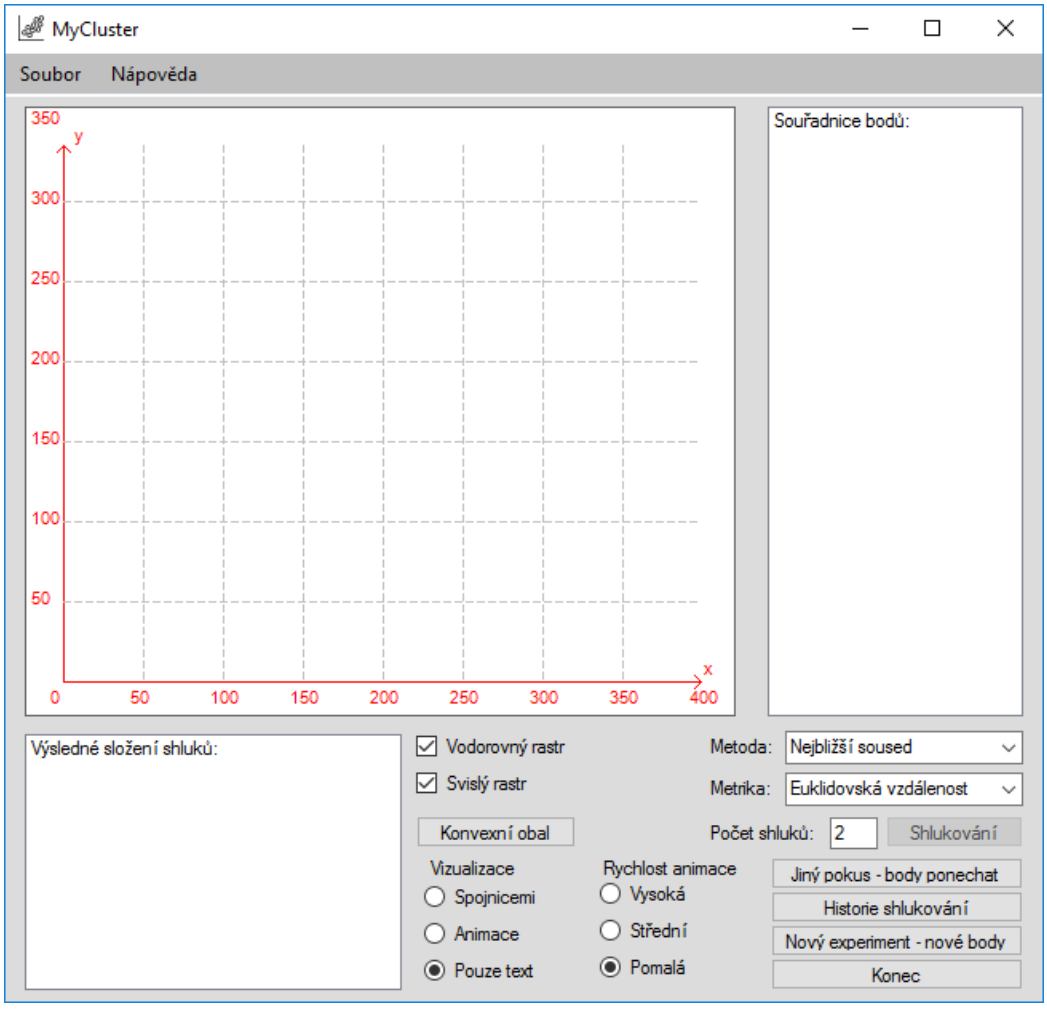
Uživatel nemusí pracovat pouze s přednastavenou velikostí okna aplikace při jejím spuštění, ale může si velikost okna aplikace zvětšovat a pracovní část s „rovinou“ je automaticky překreslována jako klientská, zvětšuje se a ostatní části okna aplikace se zvětšují podle logiky ovládání aplikace.

Spodní levá část je vyhrazena pro komponentu ListBox, ve kterém se po ukončení procesu shlukování vypíše výsledné složení jednotlivých shluků ve tvaru „Shluk 1. se skládá z bodů: 1, 0, 2, 4“ a celková průměrná hodnota siluety zaokrouhlena na čtyři desetinná místa. Jak je popsáno v kapitole 3, tak hodnota celkové průměrné siluety udává kvalitu daného shlukovacího procesu (resp. rozkladu), kde z mé zkušenosti hodnota menší jak 0,25 indikuje velice špatné rozložení, hodnota mezi 0,26 – 0,50 indikuje slabé rozložení, ale může se také jednat pouze o výjimku, hodnota mezi 0,51 – 0,70 indikuje rozumné rozložení a hodnota nad 0,70 indikuje vynikající rozložení. Na obr. 7, je ukázka kódu pro výpočet celkové průměrné hodnoty siluety objektu.

```
1  for (int i = 0; i < Silueta.Count; i++)
2  {
3    Silueta[i].S = (Silueta[i].b - Silueta[i].a) /
                    Math.Max(Silueta[i].a, Silueta[i].b);
4  }
```

**Obrázek 7 Ukázka kódu výpočtu hodnoty siluety**

Posledním úsekem aplikace, který ještě nebyl popsán, je spodní pravá část aplikace. Zde je umístěno několik ovládacích prvků. Jejich funkčnost a význam bude následně vysvětlena:




**Obrázek 8 Grafické uživatelské rozhraní aplikace Shluková analýza**

### Zaškrtávací boxy

**Vodorovný rastr**: možnost vykreslení/nevykreslení vodorovného rastru.

**Svislý rastr**: možnost vykreslení/nevykreslení svislého rastru.

**Tlačítko**

**Konvexní obal**: pro shluky, které nám vznikly za pomoci shlukovacího procesu, si můžeme pomocí této funkce nechat vykreslit konvexní obal pro snadnější orientaci v Panelu aplikace. Bližší popis celého algoritmu konvexního obalu nalezneme v kapitole 5.4.

**Ovládací prvky typu RadioButton**

**Vizualizace**: možnost volby vizualizačního procesu:

- **Pouze text**: zvolením této vizualizační volby docílíme pouze textového výstupu, který se skládá z jednotlivých kroků shlukovacího procesu zobrazovaných v pravém okně spolu se souřadnicemi jednotlivých bodů, výsledného výpisu konečného složení jednotlivých shluků spolu s celkovou průměrnou hodnotou siluety v levém dolním okně.
- **Animace**: v případě vizualizace pomocí animace je uživateli v každém kroku shlukovacího procesu vykreslena spojnice mezi dvěma spojovanými body, které patří do stejného shluku. Uživatel tím získává větší přehled o celém shlukovacím procesu.
- **Spojnicemi**: poslední možností, kterou má uživatel k dispozici, je vizualizace pomocí spojnic, kde na konci celého shlukovacího procesu se mezi jednotlivými body, které patří do jednoho shluku, vykreslí spojnice. Body jsou spojeny na základě vazby, která se odvíjí od typu zvoleného shlukovacího algoritmu.

**Rychlost animace**: v momentě, kdy si uživatel zvolí vizualizaci pomocí animace, tak může v této nabídce vybírat ze tří možných rychlostí – pomalé, střední a rychlé.

**Skryté seznamy ComboBox**

**Metoda**: pomocí komponenty ComboBox je možný výběr ze čtyř různých algoritmů shlukové analýzy – nejbližší soused, nejvzdálenější soused, centroidní metoda a metoda k-centroidů neboli K-Means (bližší popis všech použitých algoritmů naleznete v kapitole 5.2).

**Metrika**: pomocí komponenty ComboBox je možný výběr ze tří různých algoritmů pro výpočtu měr vzdáleností – Euklidovská vzdálenost, Čebyševova vzdá-

lenost a Manhattanská vzdálenost (bližší popis všech použitých algoritmů naleznete v kapitole 5.3).

Volba výběru metody shlukování a metriky ovládacím prvkem ComboBox vyplynula ze snahy maximalizovat pracovní plochu pro interaktivní zadávání bodů.

**TextBox**

**Počet shluků**: zde uživatel určuje počet výsledných shluků požadovaných v daném shlukovacím procesu (předdefinovaný počet je nastaven na hodnotu 2).

**Další tlačítka**

**Shlukování**: po zvolení požadovaného algoritmu shlukové analýzy (metoda), míry vzdálenosti (metrika), požadovaného animačního procesu, vhodného počtu výsledných shluků a samozřejmě po „naklikání“ bodů na Panelu s rovinou, na které budeme chtít aplikovat všechny zmíněné volby, se stane tlačítko dostupným a lze jím spustit proces shlukování.

**Jiný pokus – body ponechat**: v momentě, kdy uživatel chce pouze změnit například algoritmus shlukové analýzy (metoda), míru vzdálenosti (metrika) nebo počet výsledných shluků, tak pomocí daného tlačítka se veškeré potřebné informace o množině bodů zachovají a jsou smazány pouze shlukovací postupy a proměnné z minulého shlukovacího procesu. Body se překreslí v původně „naklikaném“ stavu.

**Historie shlukování**: po každém procesu shlukování jsou o tomto procesu uloženy údaje, jako jsou použitá metoda, metrika, výsledné složení shluků a celková průměrná hodnota siluety.

**Nový experiment – nové body**: v momentě, kdy uživatel potřebuje smazat stávající množinu bodů a mít možnost ihned vytvořit množinu bodů novou, tak pomocí daného tlačítka se veškeré funkční části aplikace vrátí do původního stavu a aplikace je připravena pro nový experiment.

**Konec**: ukončení aplikace.

Aplikace také obsahuje systémovou lištu, která se skládá ze dvou nabídek. První nabídkou je *Soubor,* ve kterém uživatel nalezne funkci jako *Nový experiment*, která umožní uživateli naklikání nové množiny bodů a provádění nového experimentu, funkci *Otevřít*, která umožňuje načtení množiny bodů z csv souboru dříve vytvořeného. Pro úspěšné načtení souboru je nutné, aby formát souboru měl podobu jako uvedený přiklad v tabulce 4 a oddělovačem byl středník „;“.

**Tabulka 4 Formát csv souboru**

| Bod | X | Y |
|---|---|---|
| 0 | 83 | 132 |
| 1 | 207 | 263 |

Dalšími funkcemi obsaženými v nabídce *Soubor* jsou funkce *Uložit* a *Uložit jako…*, kde první funkce uloží množinu bodů pod názvem *Souradnice_bodu.csv* do stejného adresáře, ve kterém se vyskytuje aplikace a druhá funkce umožňuje uživateli zvolit název souboru i jeho umístění. Ukládaný soubor má stejnou podobu jako tabulka 4. Poslední funkcí je *Konec*, která ukončí aplikaci.

Druhou nabídkou, kterou uživatel může vidět na systémové liště, je *Nápověda*. V této nabídce nalezne uživatel tři volby. První je volba *Ovládací prvky,* kde jsou uživateli popsány všechny ovládací prvky aplikace, jak se používají, a čeho se pomocí nich může v aplikaci docílit. Druhou volbou je *Shluková analýza,* kde je uživateli popsán cíl shlukové analýzy, rozdělení na hierarchické a nehierarchické metody a následně popsání všech metod shlukové analýzy použitých v mé aplikaci. K těmto dvěma volbám jsem využil nápovědu ve formátu Microsoft Compiled HTML Help. Poslední volba je *O programu,* kde jsem zaznamenal důvod vzniku aplikace, autora práce a vedoucí práce.

## 5.2 Použité algoritmy shlukové analýzy

V aplikaci MyCluster je na výběr celkově ze čtyř shlukovacích algoritmů, které patří mezi často citované algoritmy používané ve shlukové analýze. Ze zmíněných čtyř shlukovacích algoritmů se tři řadí mezi hierarchické aglomerativní metody, konkrétně se jedná o algoritmus nejbližšího souseda, nejvzdálenějšího souseda a centroidní metodu. Čtvrtý algoritmus se řadí mezi nehierarchické metody, konkrétně se jedná o algoritmus k-centroidů (v aplikaci pod názvem K-Means).

### 5.2.1 Nejbližší soused

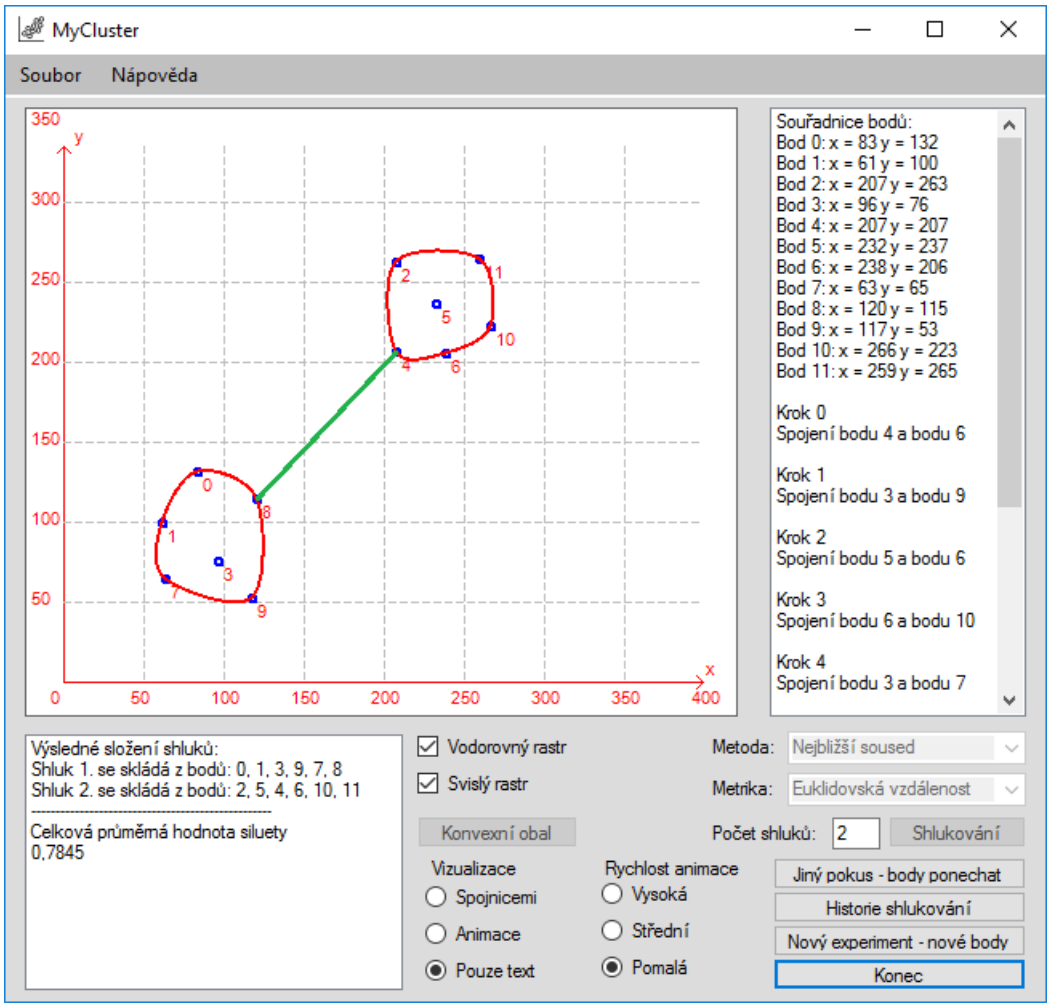
Metoda nejbližšího souseda (angl. Single-linkage nebo the nearest neighbor method) se řadí mezi hierarchické aglomerativní metody. Algoritmus v každém kroku spojí dva shluky, které obsahují nejbližší pár objektů, a pokračuje tak dlouho, dokud není splněna podmínka požadovaného počtu výsledných shluků. Na obr. 9 můžeme vidět ukázku shlukování pomocí algoritmu nejbližšího souseda.

**Výhody:**

- jednoduchost,
- není potřeba přepočítávat matici vzdáleností.

**Nevýhody:**

- vznik zřetězených shluků,
- citlivost na odlehlé hodnoty.




**Obrázek 9 Ukázka shlukování pomocí algoritmu nejbližší soused**

Algoritmus nejbližší soused se skládá z následujících kroků:

1. Prvním krokem je výpočet „matice vzdáleností“, který se skládá ze dvou *for* cyklů. V těchto cyklech probíhá výpočet vzdálenosti na základě zvolené míry vzdáleností, ukázku kódu můžeme vidět na obr 10. Ve skutečnosti jsou hodnoty ukládány do Listu třídy *TVzdalenosti*, která obsahuje číselnou proměnnou *bod*, textový řetězec *shluk* a List *vzdalenost* třídy double.

```
1 private void MaticeVzdalenostiNejblizsiSoused()
2 {
3   double VzdalenostX, VzdalenostY;
4
5   for (int i = 0; i < pocetBodu; i++)
6   {
7     Vzdalenosti.Add(new TVzdalenosti { bod = i, shluk = ""+i+"" });
8
9     for (int j = 0; j < pocetBodu; j++)
10    {
11      if (Metrika.SelectedIndex == 0)
12      {
13        VzdalenostX = Bod[i].poloha.X - Bod[j].poloha.X;
14        VzdalenostY = Bod[i].poloha.Y - Bod[j].poloha.Y;
15
16        Euklid(VzdalenostX, VzdalenostY);
17
18        Vzdalenosti[i].vzdalenost.Add(vzdalenost);
19      }
20    }
21  }
22 }
```

**Obrázek 10 Ukázka kódu výpočtu matice vzdálenosti s Euklidovskou mírou vzdálenosti**

V tabulce 5 je uveden pouze příklad, jak matice vzdáleností (ve skutečnosti jsou hodnoty ukládány do Listu *vzdalenost*) vypadá pro tři body B1, B2 a B3.

**Tabulka 5 Matice vzdálenosti**

| Bod / Bod | B1 | B2 | B3 |
|---|---|---|---|
| B1 | 0 | 40,2 | 136,1 |
| B2 | 40,2 | 0 | 151,8 |
| B3 | 136,1 | 151,8 | 0 |

2. Druhým krokem je nalezení nejmenší vzdálenosti (resp. dvou bodů) z matice vzdáleností. Nejprve naleznu první kladnou nenulovou hodnotu z matice vzdáleností pomocí *for* cyklu, který po nalezení přeruším. Následně, ve dvou vnořených *for* cyklech, pomocí funkce *if* porovnávám nalezenou hodnotu s ostatními hodnotami z matice vzdáleností, a tím na-

leznu nejmenší vzdálenost mezi dvěma body. Zároveň kontroluji, aby nejmenší hodnota nebyla záporná, protože po nalezení nejmenší hodnoty je hodnota přepsána na hodnotu -1.

3. Ve třetím kroku provádím kontrolu, jak je vidět na obr. 11, zdali nalezené body s nejmenší vzdáleností mezi nimi již nepatří do stejného shluku. Kontrolu provádím pomocí atributu *cisloShluku*, který každý bod vlastní. V případě shody nastavuji proměnnou *spojBod* třídy boolean na false, což znamená, že body nebudou spojeny, protože patří do stejného shluku, a vracím se zpět do bodu 2. k nalezení další nejmenší vzdálenosti. V případě neshody shluky sjednotím a pokračuju na další krok.

```
1 if (Bod[bod1].cisloShluku == Bod[bod2].cisloShluku)
2 {
3   spojBod = false;
4   return;
5 }
```

**Obrázek 11 Ukázka kódu kontroly dvou bodů**

4. Pokud není splněna podmínka, že se počet shluků = počtu požadovaných shluků, tak se vracím zpět do bodu 2 k nalezení další nejmenší vzdálenosti. Jinak dojde k vypsání výsledného složení shluků, celkové průměrné hodnoty siluety a algoritmus nejbližší soused je u konce.

### 5.2.2 Nejvzdálenější soused

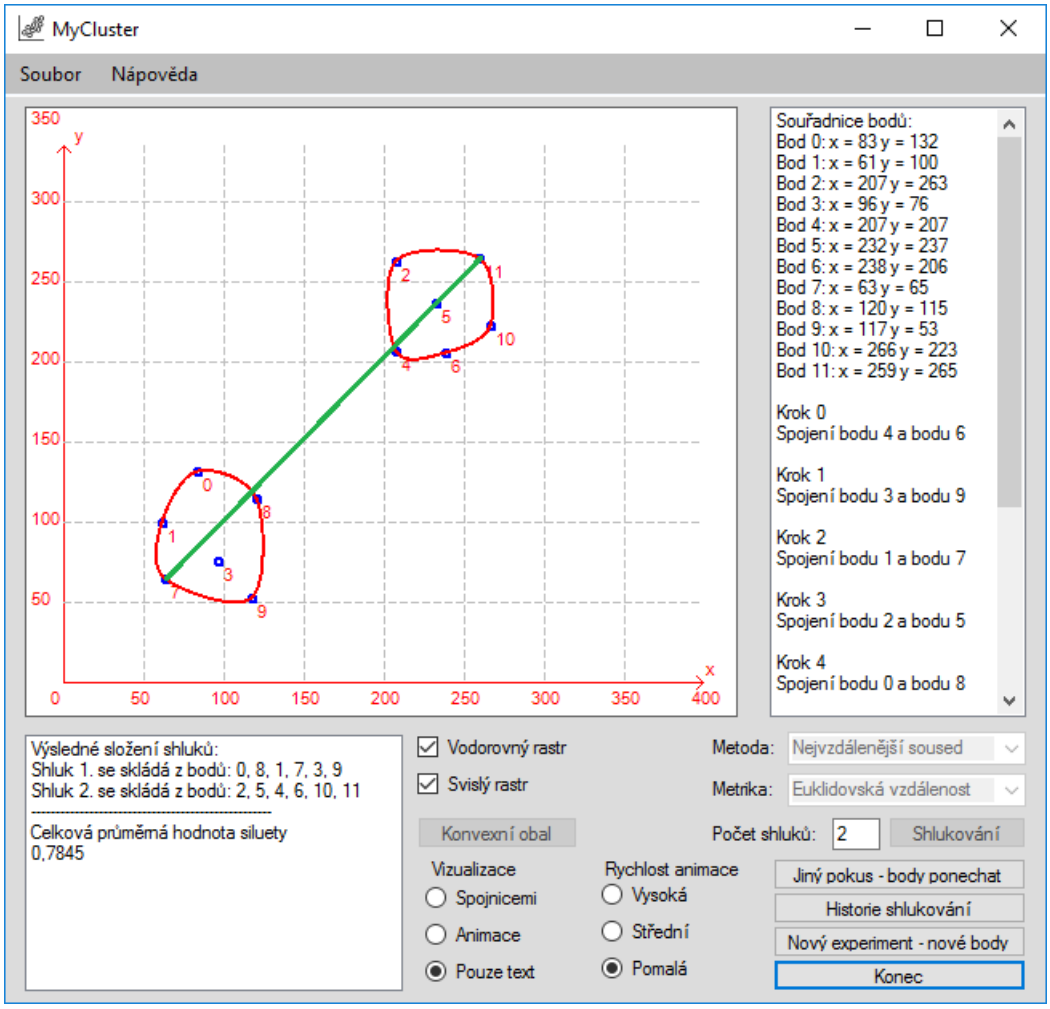
Metoda nejvzdálenější soused (angl. Complete linkage nebo farthest neighbor method) se řadí stejně jako metoda nejbližšího souseda mezi hierarchické aglomerativní metody. Jak už název metody napovídá, jedná se o opak metody nejbližšího souseda, to znamená., že vzdálenost mezi dvěma shluky je u metody nejvzdálenějšího souseda definována jako vzdálenost mezi nejvíce vzdálenými dvěma body, kde každý bod patří do odlišného shluku. Na obr. 12 můžeme vidět ukázku shlukování pomocí algoritmu nejvzdálenějšího souseda.

**Výhody:**

- zabraňuje vzniku zřetězených shluků,
- menší citlivost na odlehlé hodnoty.

**Nevýhody:**

- sklon k rozdělování větších shluků,
- nutný přepočet matice vzdáleností.




**Obrázek 12 Ukázka shlukování pomocí algoritmu nejvzdálenější soused**

Algoritmus nejvzdálenější soused se skládá z následujících kroků:

1. Prvním krokem metody nejvzdálenějšího souseda je výpočet „matice vzdáleností“. Výpočet je proveden stejným postupem jako u metody nejbližšího souseda, který je popsán v kapitole 5.2.1.
2. Druhým krokem je nalezení nejmenší vzdálenosti (resp. dvou bodů) z matice vzdáleností. Jako u metody nejbližšího souseda si nejprve naleznu první kladnou nenulovou hodnotu z matice, a poté porovnávám nalezenou hodnotu s ostatními hodnotami z matice vzdáleností. Tímto

postupem naleznu nejmenší vzdálenost mezi dvěma body z matice vzdálenosti.

3. Po nalezení dvou bodů, které mezi sebou mají nejmenší vzdálenost, přepočítám matici vzdáleností. Matice vzdáleností je přepočítána tak, že u spojování obou bodů (resp. shluků) porovnávám vzdálenosti k ostatním bodům a ty menší vzdálenosti jsou nahrazeny těmi většími. Tímto postupem docílím stavu, že mezi body (resp. shluky) je vždy brán nejvzdálenější soused.
4. Ve čtvrtém kroku provádím kontrolu, zdali již nalezené body, s nejmenší vzdáleností mezi nimi, nepatří do stejného shluku. V případě shody nebudou body spojeny, protože patří do stejného shluku, a vracím se zpět do bodu 2. k nalezení další nejmenší vzdálenosti. V případě neshody body (resp. shluky) sjednotím a pokračuju na další krok.
5. Pokud není splněna podmínka, že se počet shluků = počtu požadovaných shluků, tak se vracím zpět do bodu 2. k nalezení další nejmenší vzdálenosti. Jinak dojde k vypsání výsledného složení shluků, celkové průměrné hodnoty siluety a algoritmus nejvzdálenější soused je u konce.

### 5.2.3 Centroidní metoda

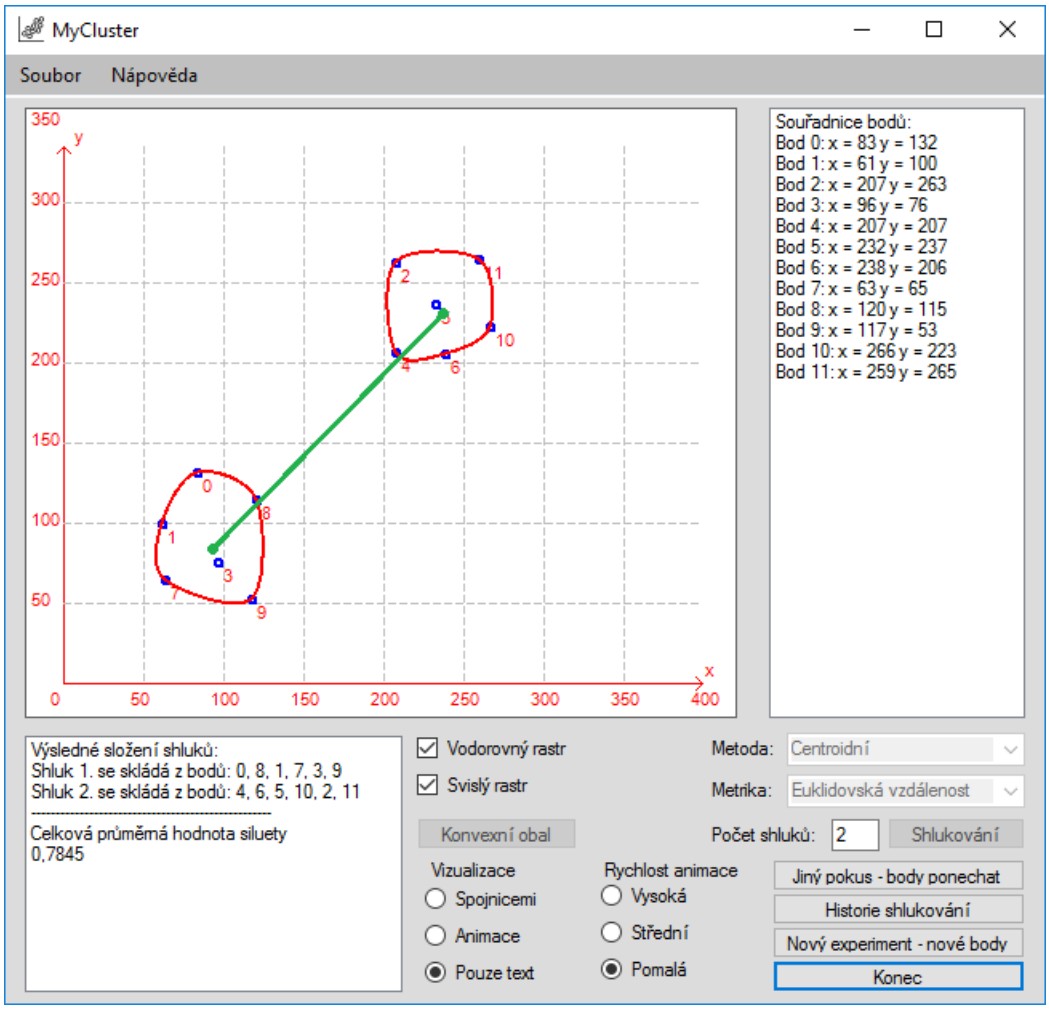
Centroidní metoda (angl. Centroid linkage) je poslední mnou použitá metoda řazená mezi hierarchické aglomerativní metody. U každého shluku se vypočítá tzv. centroid, který je zároveň těžištěm shluku a bere se jako reprezentativní bod shluku. Vzdálenosti jsou počítány mezi jednotlivými centroidy všech shluků a za nejpodobnější shluky považujeme ty s nejméně vzdálenými centroidy. Na obr. 13 můžeme vidět ukázku shlukování pomocí algoritmu centroidní metody.

**Výhody:**

- shluky jsou reprezentovány hodnotou centroidu,
- menší citlivost na odlehlé hodnoty.

**Nevýhody:**

- nebere ohled na změnu vzdálenosti mezi přepočítaným centroidem a jednotlivými body,
- nutný přepočet matice vzdáleností.




**Obrázek 13 Ukázka shlukování pomoci centroidní metody**

Algoritmus centroidní metody se skládá z následujících kroků:

1. Prvním krokem centroidní metody je výpočet „matice vzdáleností mezi centroidy“, který se skládá ze dvou *for* cyklů. V těchto cyklech probíhá výpočet vzdálenosti na základě zvolené míry vzdáleností mezi jednotlivými centroidy, které reprezentují shluky.
2. Druhým krokem je nalezení nejmenší vzdálenosti (resp. dvou bodů nebo shluků) z matice vzdáleností mezi centroidy. Stejně jako u předchozích metod si nejprve naleznu první kladnou nenulovou hodnotu z matice, a poté porovnávám nalezenou hodnotu s ostatními hodnotami z matice. Tímto postupem naleznu nejmenší vzdálenost mezi dvěma body (resp. shluky) z matice vzdálenosti mezi centroidy. Po nalezení nejmenší vzdálenosti je její hodnota přepsána na -1.

3. Ve třetím kroku provádím kontrolu, zdali již nalezené body (resp. shluky), s nejmenší vzdáleností mezi jejich centroidy, nepatří do stejného shluku. V případě shody nebudou body spojeny, protože patří do stejného shluku, a vracím se zpět do bodu 2. k nalezení další nejmenší vzdálenosti. V případě neshody body (resp. shluky) sjednotím a pokračuji na další krok.
4. V momentě sjednocení dvou bodů (resp. shluků) musí být přepočítána matice vzdáleností mezi centroidy. K tomu, abych mohl přepočíst matici vzdáleností mezi centroidy, musím nejdříve znovu vypočíst polohu centroidů, jelikož se po předchozím spojení změnila. Poloha centroidu se počítá jako průměrná poloha všech bodů ve shluku. Po přepočtu polohy centroidů může být přepočítána matice vzdáleností mezi centroidy stejně jako v kroku 1.
5. Pokud není splněna podmínka, že se počet shluků = počtu požadovaných shluků, tak se vracím zpět do bodu 2. k nalezení další nejmenší vzdálenosti. Jinak dojde k vypsání výsledného složení shluků, celkové průměrné hodnoty siluety a algoritmus centroidní metody je u konce.

### 5.2.4 Metoda k-centroidů

Metoda k-centroidů se řadí mezi nehierarchické metody strojového učení bez učitele. Algoritmus k-centroidů je jednoduchá iterační metoda, která rozděluje daný soubor objektů do předem stanoveného počtu *k* shluků.

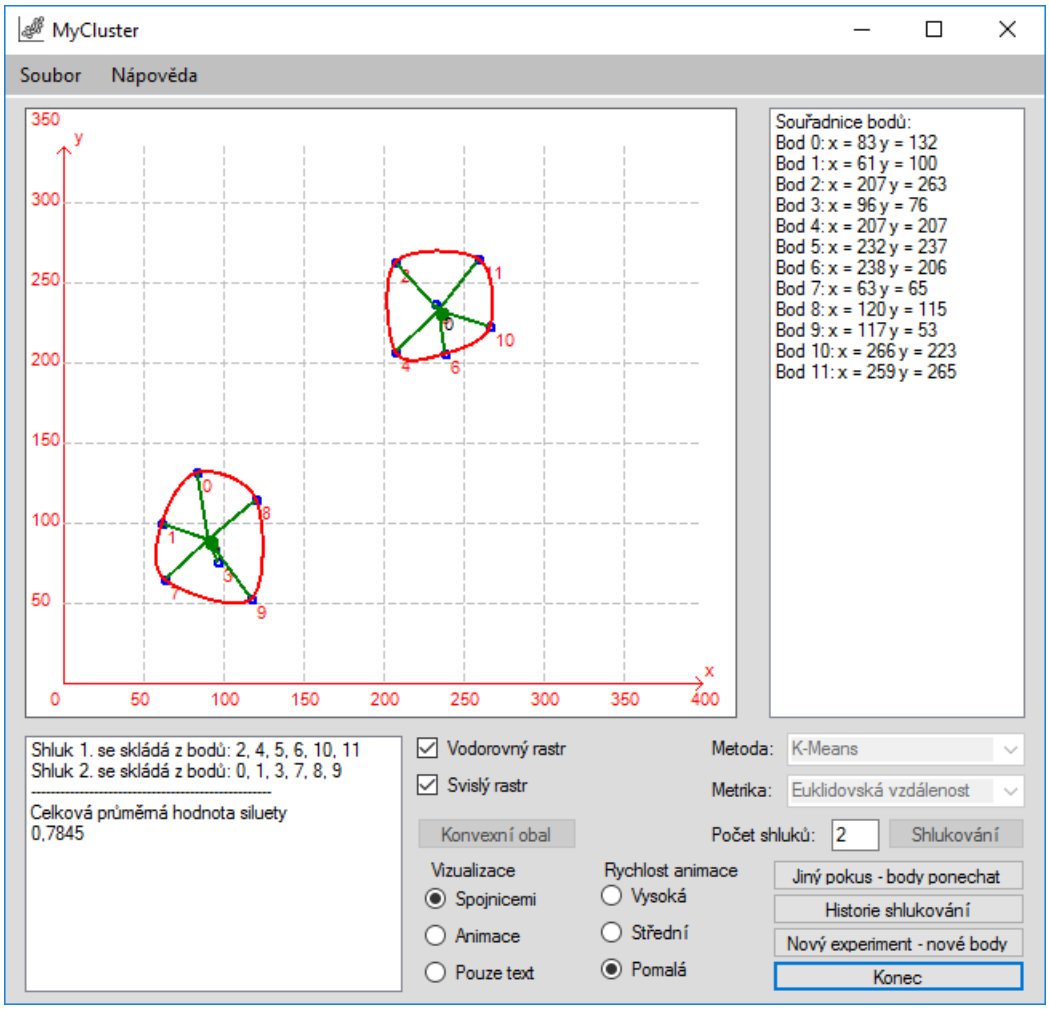
Stejně jako u centroidní metody, popsané v kapitole 5.2.4, jsou shluky reprezentovány centroidy. V mém provedení metody k-centroidů musí uživatel stanovit počet požadovaných shluků a ke každému shluku „nakliknout“ jeho centroid. Na základě jejich polohy jsou k nim přiřazeny body, které k jednotlivým centroidům mají nejmenší vzdálenost. Po zařazení každého bodu do příslušného shluku dochází k přepočítání polohy centroidů a proces se opakuje.

**Výhody:**

- rychlé a efektivní z hlediska výpočetní náročnosti,
- bere ohled na změnu vzdálenosti mezi přepočítaným centroidem a jednotlivými body.

**Nevýhody:**

- odlišná poloha počátečních bodů může vést k odlišným výsledkům shlukování,
- citlivost na odlehlé hodnoty.




**Obrázek 14 Ukázka shlukování pomocí metody k-centroidů**

Algoritmus metody k-centroidů se skládá z následujících kroků:

1. V prvním kroku je nutné, aby uživatel zvolil požadovaný počet shluků a stanovil počáteční polohu jednotlivých centroidů reprezentujících tyto shluky.
2. Druhým krokem je výpočet matice vzdáleností mezi jednotlivými body a centroidy. Vzdálenost je počítána pomocí Euklidovské metriky a je ukládána do Listu *vzdalenost* třídy double, který je součástí Listu *Vzdalenosti* třídy *TVzdalenosti*.

3. Ve třetím kroku procházím pomocí *for* cyklu vypočtenou matici vzdálenosti mezi body a centroidy. U každého bodu naleznu centroid (resp. shluk), ke kterému má nejmenší vzdálenost, a k tomu shluku ho přiřadím (resp. přeřadím). Zároveň nastavuji pomocnou proměnnou *zmena* třídy boolean na hodnotu true, tzn., že byla provedena změna ve složení shluků. Ukázku kódu přiřazení bodu k nejbližšímu shluku můžeme vidět na obr. 15.

```
1 private void PridejBodKMeans()
2 {
3   for (int i = 0; i < Bod.Count; i++)
4   {
5     min = Vzdalenosti[i].vzdalenost.Min();
6     index = Vzdalenosti[i].vzdalenost.IndexOf(min);
7
8     Shluky[index].Body.Add(Bod[i]);
9
10    if (Bod[i].cisloShluku != index)
11    {
12      Bod[i].cisloShluku = index;
13      zmena = true;
14    }
15  }
16 }
```

**Obrázek 15 Ukázka kódu přiřazení bodu k nejbližšímu shluku**

4. Ve čtvrtém kroku dochází ke kontrole pomocné proměnné *zmena*. Pokud platí, že je proměnná rovna hodnotě *true*, tak se algoritmus vrací zpět do 2. kroku k výpočtu matice vzdáleností mezi jednotlivými body a centroidy. Pokud platí, že je proměnná rovna hodnotě *false*, tak dochází k vypsání výsledného složení shluků, celkové průměrné hodnoty siluety a algoritmus metody k-centroidů je u konce.

## 5.3 Použité metriky

Pro výpočet vzdáleností mezi jednotlivými body jsem implementoval tři základní funkce, mezi kterými má uživatel možnost výběru.

- **Euklidovská vzdálenost**

```
1 private void Euklid(double X, double Y)
2 {
3   vzdalenost = Math.Sqrt((Math.Pow(X, 2)) + (Math.Pow(Y, 2)));
4 }
```

**Obrázek 16 Výpočet vzdálenosti pomocí euklidovské metriky**

- **Čebyševova vzdálenost**

```
1 private void Cebysev(double X, double Y)
2 {
3   vzdalenost = Math.Max(X, Y);
4 }
```

**Obrázek 17 Výpočet vzdálenosti pomocí Čebyševovy metriky**

- **Manhattanská vzdálenost**

```
1 private void Manhattan(double X, double Y)
2 {
3   vzdalenost = X + Y;
4 }
```

**Obrázek 18 Výpočet vzdálenosti pomocí manhattanské metriky**

## 5.4 Konvexní obal

Za konvexní obal považujeme tzv. konvexní množinu objektů (v našem případě bodů). Jedná se o nejmenší konvexní mnohoúhelník, který obsahuje všechny objekty, pro splnění konvexnosti musí navíc platit, že pokud vytvoříme úsečku mezi dvěma libovolnými objekty (resp. body), tak se daná úsečka musí nacházet uvnitř obalu.


**Obrázek 19 Rozdíl mezi konvexním a nekonvexním obalem**

Pro zjištění konvexní množiny jsem využil metodu *Graham Scan*, která se vyznačuje především svojí nízkou časovou složitostí. Naopak nevýhodou je možnost aplikace pouze ve dvourozměrném prostoru. V mém případě je tato nevýhoda irelevantní, a proto jsem ji zvolil.

Algoritmus metody Graham Scan se skládá z následujících kroků:

1. Prvním krokem algoritmu je nalezení objektu (resp. bodu), jehož souřadnice y je v daném shluku nejmenší.
2. Dalším krokem je výpočet úhlů, které jsou svírány mezi nalezeným objektem a ostatními objekty ve shluku. Ukázku kódu pro výpočet úhlu mezi body můžete vidět na obr. 20.

```
1  public double Uhel(double px2, double py2)
2  {
3    double uhel = 0.0;
4
5    uhel = System.Math.Atan(System.Math.Abs(py2) / Sys-
                              tem.Math.Abs(px2));
6
7    uhel = uhel * 180 / System.Math.PI;
8
9    if (px2 < 0)
10     uhel = 180 - uhel;
11
12   return uhel;
13 }
```

**Obrázek 20 Ukázka kódu pro výpočet úhlu mezi body**

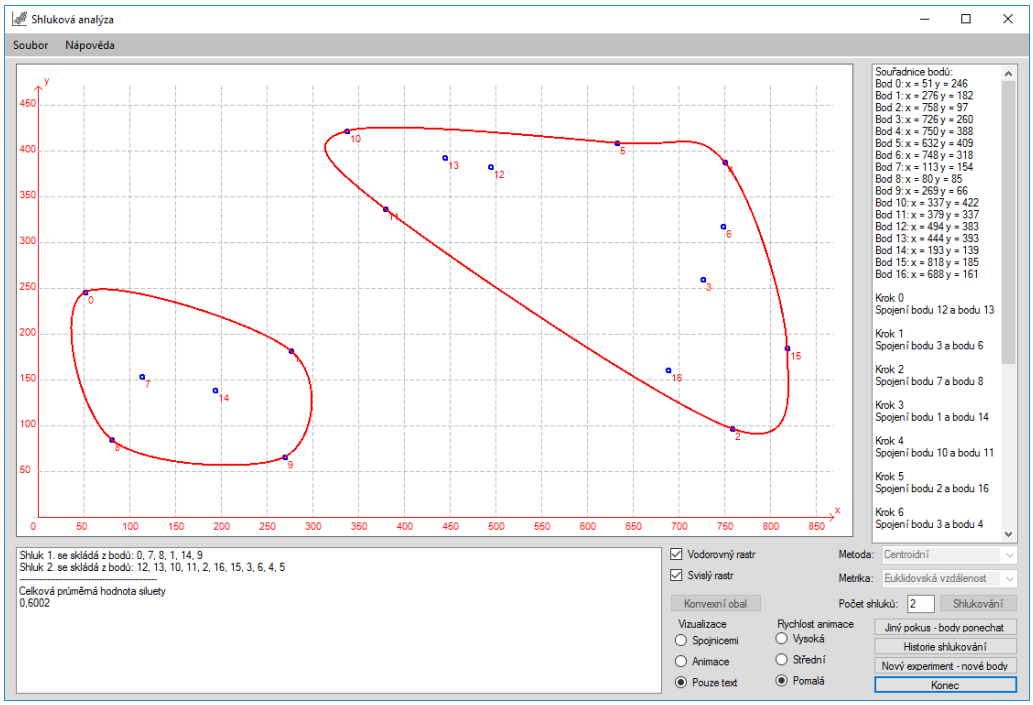
3. Ve třetím kroku dochází k seřazení matice s vypočtenými úhly, které jsou svírány s nalezeným bodem v 1. kroku a všemi ostatními body v daném shluku, od nejmenšího po největší. Společně se řazením této matice také dochází k řazení daných bodů, aby byli ve stejném pořadí jako v matici s vypočtenými úhly.

4. Posledním krokem algoritmu je testování, zda uspořádaná trojice bodů $b_{i-1}$, $b_i$ a $b_{i+1}$, patřících do konvexní množiny, splňuje podmínku „levotočivosti“. Pokud je podmínka splněna, tak body nejspíše patří do konvexní množiny a následuje testování následující trojice bodů $b_i$, $b_{i+1}$ a $b_{i+2}$. Pokud podmínka splněna není, tak se z dané trojice bodů prostřední bod $b_i$ odstraní, protože nemůže být členem konvexní množiny, a následuje testování s předešlými dvěma body $b_{i-2}$, $b_{i-1}$ a $b_{i+1}$.

# 6 Testování aplikace

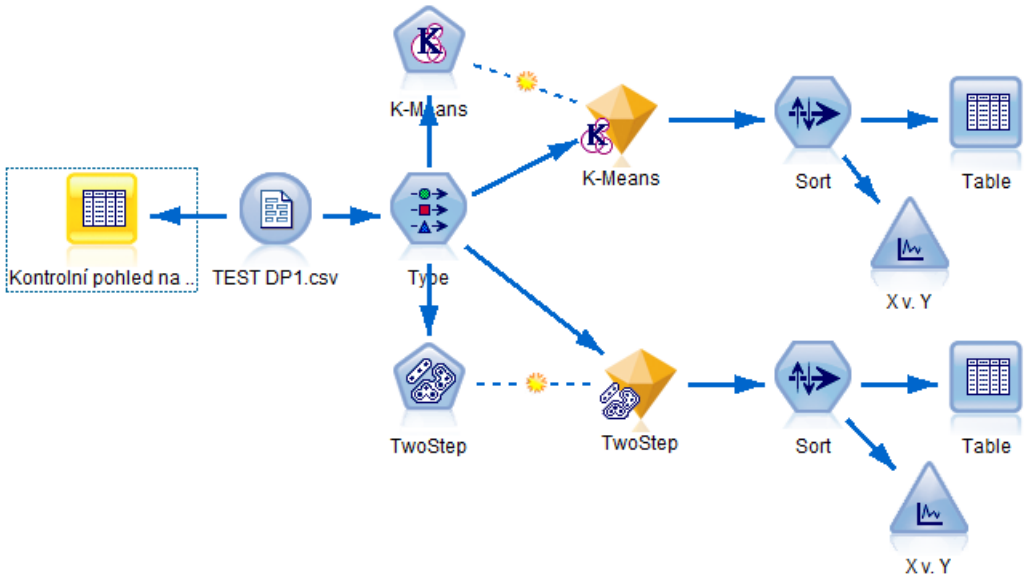
V zadání této diplomové práce je jedním z požadavků testování aplikace mezi studenty oboru Informační technologie v navazujícím studiu. Aplikace byla prezentována v rámci výuky a následně byla poskytnuta studentům na řádné testování a to opakovaně. Na základě zkušeností studentů s používáním aplikace a jejich návrhů na zlepšení, byla aplikace upravena do finální podoby.

Všechny návrhy na zlepšení jsem prodiskutoval s vedoucí práce i se studenty. Z tohoto brainstromingu vyplynuly různé návrhy, například na vizualizaci shlukování animacemi. Dále jsem intenzivně vyvíjel aplikaci tak, aby bylo možno data ukládat do souboru a opakovaně načítat. Zjednodušilo se tak „otravné“ klikání při zadávání bodů, které jsou objekty shlukování. Tím uživatel získal možnost experiment opakovat a také stejná data shlukovat v jiném data miningovém nástroji. Byla to právě otázka studentů, jak by shlukování dopadlo v IBM SPSS Modeleru, která byla inspirací k dalšímu vývoji aplikace. Testování výsledků shlukování aplikace MyCluster v profesionálním data minigovém nástroji poskytlo pro stejné nastavení stejné výsledky a potvrdilo, že algoritmy jsou naprogramované správně.

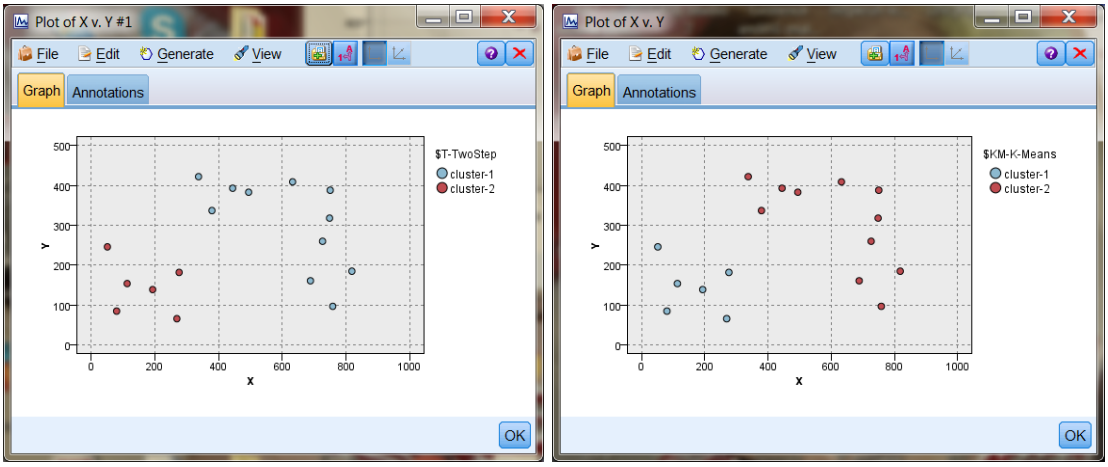



**Obrázek 21 Výsledek shlukovacího procesu v aplikaci MyCluster**

Pro testování v Modeleru jsem postavil proud, kde jsou použité dva modelovací uzly pro K-means a TwoStep, grafy a tabulky, které potvrzují shodu výsledků MyCluster a Modeleru.




**Obrázek 22 Testovací proud v IBM SPSS Modeleru**




**Obrázek 23 Výsledky shlukování v IBM SPSS Modeleru**

i výpočet siluety pro tento test a celou řadu dalších potvrdil shodu.


**Obrázek 24 Kvalita rozkladu v IBM SPSS Modeleru**

Dalším krokem inspirovaným diskusí se studenty, bylo přidání tlačítka *Historie shlukování*, kde si uživatel může zobrazit všechny výsledky z uskutečněných procesů shlukování po spuštění aplikace. Do historie se ukládá, jaká metoda a metrika byla použita, dále výsledné složení shluků a celková průměrná hodnota siluety pro daný proces shlukování. Uživatel si tak například může porovnat výsledné složení shluků u různých metod nebo může porovnat celkovou průměrnou hodnotu siluety u stejné metody, ale odlišného počtu výsledných shluků. Studenti dostali také dotazník, který mohli nepovinně vyplnit a ohodnotit práci s aplikací i napsat další připomínky či jiná sdělení. Dotazník vyplnilo 13 účastníků, odpovědi jsou v příloze, zde jen pro ilustraci uvádím některá sdělení studentů na otázku:

Jak byste zlepšili danou aplikaci? (6 odpovědí, odpověď nebyla povinná)

- Doporučuji dál aplikaci rozvíjet. Je to jistě na další projekt nebo závěrečnou studentskou práci. Děkuji za možnost experimentu a porovnání výsledků. To se mi velice líbilo.

- Doplnil bych plovoucí nápovědu

**Odpověď** - tento návrh je diskutabilní. Aplikace je vybavená nápovědou, která by mohla postačovat.

- Vstup do problémů shlukové analýzy s touto aplikací je jednodušší.

- Uvítal bych podobně zpracované výkladové experimenty spojené s přípravou dat.

**Odpověď** – toto by mohlo být realizované v navazující studentské práci. S přípravou dat je spojené velké množství datových transformací, úprav, analýz atd.

- Možno bych zvýraznila když končí shlukování bez animace, mohlo by se něco výraznějšího stát. Sice se mnění text vlevo dole, ale je nevýrazný, uživateli bych dala nějak najevo "jsem hotov"

**Odpověď** – bylo upraveno.

- Přidala bych vzorce pro výpočet vzdáleností, vzhledem k tomu, že jde o výukový nástroj, myslím, že by tam být měly. Po dokončení shlukování bych vytvořené shluky zvýraznila přímo, aniž by uživatel musel klikat na tlačítko "Konvexní obal", případně bych místo tlačítka použila checkbox, jestli se mají shluky po dokončení shlukování zvýraznit nebo nemají.

**Odpověď** – bylo upraveno.

- Aplikace nefunguje zcela správně, pokud uživatel dodatečně přidá nové body a aniž by měnil jakékoli další nastavení, klikne znovu na tlačítko "Shlukování". Nově přidané body se automaticky zařadí do prvního shluku, který byl dříve vytvořený bez ohledu na jejich souřadnice a na to, do jakého shluku skutečně patří.

**Odpověď** – bylo opraveno.

## Závěr

Zadání vyžadovalo pro předmět Data mining připravit výukový počítačový experiment i s výkladovým textem pro vybrané algoritmy shlukové analýzy. V rámci řešení jsem si prohloubil znalosti o shlukové analýze a zdokonalil se v programování, což bylo zajímavé a přínosné.

Rešeršní část diplomové práce jsem zaměřil na popis jednotlivých kroků, se kterými se dataminer setkává, při aplikování data miningových metod shlukové analýzy. Prvním krokem dataminera je vždy seznámení se s datovými soubory, které má k dispozici, a strukturou dat v nich obsažených. Z těchto důvodů byla 1. kapitola věnována předzpracování dat, kde jsem popsal procesy jako čištění dat, integrace dat a transformace dat.

V diplomové práci jsem se věnoval popisu především vybraných metod shlukové analýzy, hierarchických a nehierarchických, které jsem následně programoval v aplikaci pro experimentování se shlukováním za účelem porozumění a sledování důsledků volby metod a volby dalších parametrů. U hierarchických metod jsou to metody nejbližšího souseda, nejvzdálenějšího souseda, centroidní metoda a metoda průměrné vazby. U nehierarchických metod je to metoda K-means. Podobnosti/nepodobnosti objektů jsou klíčovým problémem metod shlukové analýzy. Pro každý typ proměnné má výpočet míry podobnosti několik možností a algoritmických postupů. Pro kvantitativní, kvalitativní i binární typy proměnných jsem popsal možné postupy pro výpočet míry podobnosti. Vyhodnocení kvality konečné sestavy shluků je problematické, protože neexistuje univerzální definice pojmu kvalitní shluk. I přesto existují metody, které vycházejí z homogenity a separace shluků a umožňují nám tedy výpočet funkcionálu kvality shluku či rozkladu. Tato rešeršní část textu je východiskem pro nápovědu aplikace, která vznikla jako příspěvek k e-learningovým materiálům o data miningu.

Cílem praktické části diplomové práce bylo vytvoření aplikace pro výklad vybraných algoritmů shlukové analýzy a jejich testování. Aplikace by měla sloužit jako podpůrná pomůcka pro studenty oboru Informační technologie na Fakultě mechatroniky a Biomedicínské inženýrství na Fakultě zdravotnických studií, kteří mají předměty spojené s data miningem. Aplikaci jsem naprogramoval pomocí programovacího jazyka C# a vývojového prostředí Microsoft Visual Studio. Student má pro testování metod shlu-

kové analýzy možnost výběru mezi čtyřmi shlukovacími algoritmy (nejbližší soused, nejvzdálenější soused, centroidní metoda a metoda k-centroidů) a třemi metrikami (euklidovská, čebyševova a manhattanská). Během práce s aplikací jsou ukládány výsledky všech provedených experimentů do zvláštního okna pro historii experimentování. Student v historii vidí zvolenou metodu, typ metriky, výsledné složení shluků a funkcionál kvality rozkladu pro daný shlukovací proces. Může tak porovnávat výsledky, sledovat rozdíly kvality shlukování pro různé volby a stejná data, nebo měnit data pro stejné parametry výpočtu. Jako vizualizační prvek při průběhu shlukovacího procesu si může uživatel vybrat mezi textem, spojnicemi či animací. Student si pro vizualizaci výsledného složení shluků může nechat spočítat a vykreslit konvexní obal jednotlivých shluků. Pro snadnější práci s aplikací a pro možnost opakovatelnosti experimentů byla naprogramována možnost ukládání množiny vytvořených bodů (export) a možnost načítání množiny bodů ze souboru (import). Export dat umožňuje studentovi další experimentování se shlukováním v nějakém jiném data miningovém nástroji. Pro IBM SPSS Modeler jsem vytvořil proud, ve kterém stačí data načíst a spustit jej. Pro shlukování je použitý uzel K-means a TwoStep. Výsledky jsou pak zobrazitelné v grafech či tabulkách. Porovnáním přiřazení bodů do shluků a vypočítané siluety (obrysového koeficientu) bylo potvrzeno, že moje aplikace „shlukuje“ body správně. Dále jsem vytvořil nápovědu pro uživatele aplikace, která nejen usnadní orientaci v ovládání aplikace, ale poskytne informace o použitých metodách shlukové analýzy v aplikaci.

Vývoj aplikace byl náročný, grafické prvky jsou poměrně pracné, pokud se nedají použít hotové moduly a to byl můj případ. Aplikaci jsem testoval v rámci výuky se studenty TUL opakovaně. A z diskuse vyplynulo několik doporučení, které přispěly k finální podobě aplikace.

Moje aplikace by mohla pomoct studentům porozumět alespoň části rozsáhlé problematiky shlukové analýzy, protože v rámci výuky není prostor pro detailní výklad celého teoretického základu.

## Seznam použité literatury


[1] GARCÍA, Salvador; LUENGO, Julián; HERRERA, Francisco. *Data preprocessing in data mining*. New York: Springer, 2015.

[2] ŘEZANKOVÁ, Hana, Dušan HÚSEK a Václav SNÁŠEL. *Shluková analýza dat*. Praha: Professional Publishing, 2007. ISBN 978-80-86946-26-9.

[3] LUKASOVÁ, Alena a Jana ŠARMANOVÁ. *Metody shlukové analýzy*. Praha: Státní nakladatelství technické literatury, 1985.

[4] TAN, Pang-Ning, Michael STEINBACH a Vipin KUMAR. *Introduction to data mining*. Boston: Pearson Addison Wesley, c2006. ISBN 0-321-32136-7.

[5] KELBEL, Jan; ŠILHÁN, David. Shluková analýza. *Osobní stránka-Ivan Nagy*, 2002.

[6] ROKACH, Lior; MAIMON, Oded. Clustering methods. In: *Data mining and knowledge discovery handbook*. Springer US, 2005. p. 321-352.

[7] YIN, Yong, et al. *Data mining: Concepts, methods and applications in management and engineering design*. Springer Science & Business Media, 2011.

[8] PROCHÁZKA, Jan. *Metody fuzzy shlukové analýzy při modelování nelineárních systémů*. Praha, 2008.

[9] GORDON, Allan D. Classification, (Chapman & Hall/CRC Monographs on Statistics & Applied Probability). 1999.

[10] CHARRAD, Malika, Nadia GHAZZALI, Véronique BOITEAU a Azam NIKNAFS. NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software* [online]. 2014, **61**(6), - [cit. 2017-01-01]. DOI: 10.18637/jss.v061.i06. ISSN 15487660. Dostupné z: http://www.jstatsoft.org/v61/i06/

[11] MATTHIJS JOOST WARRENS. *Similarity coefficients for binary data properties of coefficients, coefficient matrices, multi-way metrics and multivariate coefficients*. [S.l: s.n.], 2008. ISBN 9789088910524.

[12] BONNER, Raymond E. On some clustering techniques. *IBM journal of research and development*, 1964, 8.1: 22-32.

[13] HANCOCK, John M.; ZVELEBIL, Marketa J. (ed.). *Concise Encyclopaedia of Bioinformatics and Computational Biology*. John Wiley & Sons, 2014.

[14] KARAGIANNIS, Dimitris; REIMER, Ulrich. *Practical aspects of knowledge management*. Springer-Verlag, 2003.

[15] KOVÁCS, Ferenc; LEGÁNY, Csaba; BABOS, Attila. Cluster validity measurement techniques. In: *6th International symposium of hungarian researchers on computational intelligence*. 2005.

[16] LIU, Yanchi, et al. Understanding of internal clustering validation measures. In: *2010 IEEE International Conference on Data Mining*. IEEE, 2010. p. 911-916.

[17] NGUYEN, Quynh H. a V.J. Rayward SMITH. Internal quality measures for clustering in metric spaces. *International Journal of Business Intelligence and Data Mining* [online]. 2008, **3**(1), 4- [cit. 2016-12-29]. DOI: 10.1504/IJBIDM.2008.017973. ISSN 1743-8187. Dostupné z: http://www.inderscience.com/link.php?id=17973

[18] SAITTA, Sandro; RAPHAEL, Benny; SMITH, Ian FC. A bounded index for cluster validity. In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer Berlin Heidelberg, 2007. p. 174-187.

[19] LÖSTER, Tomáš. *Hodnocení výsledků metod shlukové analýzy*. Praha, 2011. Vedoucí práce prof. Ing. Hana Řezanková, CSc.

[20] SOBÍŠEK, Lukáš; ŘEZANKOVÁ, Hana. Srovnání metod pro redukci dimenzionality aplikovaných na ordinální proměnné. *Acta Oeconomica Pragensia*, 2011, 1: 3-19.

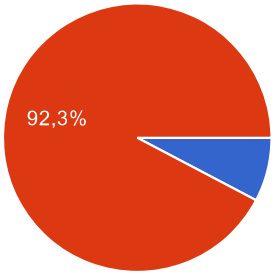
[21] *Matematická biologie* [online]. Brno: Masarykova univerzita, 2016 [cit. 2016-10-15]. Dostupné z: http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--vicerozmerne-metody-pro-analyzu-dat--shlukova-analyza

# Počet odpovědí: 13

Zobrazit všechny odpovědi  Publikovat analýzu

## Souhrn

### Měli jste problém se spuštěním aplikace?

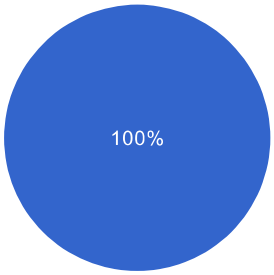



| | | |
|---|---|---|
| Ano | **1** | 7.7 % |
| Ne | **12** | 92.3 % |

### V případě, že ano, jak se chyba projevila?

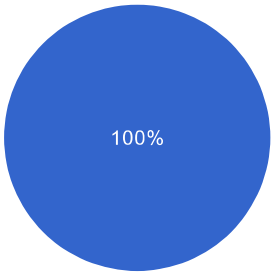
Windows 10 při spuštění hlásí následující sdělení: Systém Windows ochránil váš počítač Filtr Windows SmartScreen zabránil spuštění nerozpoznané aplikace. Spuštění této aplikace by mohlo ohrozit počítač. I přesto je možné aplikaci spustit pomocí tlačítka "přesto spustit"

### Je rozhraní aplikace uživatelsky přehledné a intuitivní?



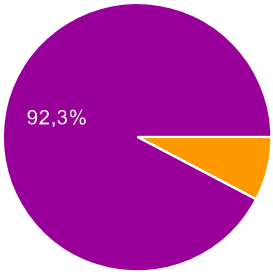

| | | |
|---|---|---|
| Ano | **13** | 100 % |
| Průměrně | **0** | 0 % |
| Spíše ne | **0** | 0 % |
| Ostatní | **0** | 0 % |

### Vystihuje aplikace dostatečně základní principy shlukové analýzy?



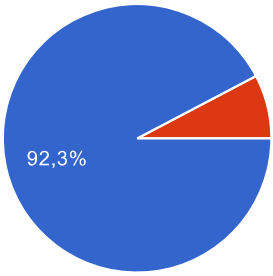

| | | |
|---|---|---|
| Ano | **13** | 100 % |
| Průměrně | **0** | 0 % |
| Ne | **0** | 0 % |

**Jak často aplikace "zamrzla" nebo "spadla"?**



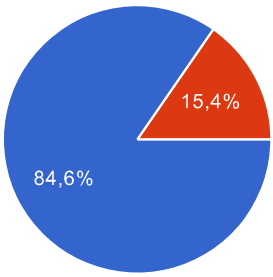

| | | |
|---|---|---|
| Velmi často | **0** | 0 % |
| Poměrně často | **0** | 0 % |
| Občas | **1** | 7.7 % |
| Téměř vůbec | **0** | 0 % |
| Nikdy | **12** | 92.3 % |

**Jak jste celkově spokojen/a s aplikací shlukové analýzy?**




| | | |
|---|---|---|
| Spokojen/a | **12** | 92.3 % |
| Průměrně spokojen/a | **1** | 7.7 % |
| Nespokojen/a | **0** | 0 % |

**Doporučili by jste aplikaci svým známým jako výukový materiál k shlukové analýze?**




| | | |
|---|---|---|
| Ano | **11** | 84.6 % |
| Nevím | **2** | 15.4 % |
| Pravděpodobně ne | **0** | 0 % |

### Jak byste zlepšili danou aplikaci?

Doporučuji dál aplikaci rozvíjet. Je to jistě na další projekt nebo závěrečnou studentskou práci. Děkuji za možnost experimentu a porovnání výsledků. To se mi velice líbilo.

Doplnil bych plovoucí nápovědu

Vstup do problémů shlukové analýzy s touto aplikací je jednodušší.

Uvítal bych podobně zpracované výkladové experimenty spojené s přípravou dat.

Možno bych zvýraznila když končí shlukování bez animace, mohlo by se něco výraznějšího stát. Sice se mnění text vlevo dole, ale je nevýrazný, uživateli bych dala nějak najevo "jsem hotov"

Přidala bych vzorce pro výpočet vzdáleností, vzhledem k tomu, že jde o výukový nástroj, myslím, že by tam být měly. Po dokončení shlukování bych vytvořené shluky zvýraznila přímo, aniž by uživatel musel klikat na tlačítko "Konvexní obal", případně bych místo tlačítka použila checkbox, jestli se mají shluky po dokončení shlukování zvýraznit nebo nemají. Aplikace nefunguje zcela správně, pokud uživatel dodatečně přidá nové body a aniž by měnil jakékoli další nastavení, klikne znovu na tlačítko "Shlukování". Nově přidané body se automaticky zařadí do prvního shluku, který byl dříve vytvořený bez ohledu na jejich souřadnice a na to, do jakého shluku skutečně patří.

# B Obsah přiloženého CD

Přiložené CD obsahuje následující soubory:

- Text diplomové práce
  - Diplomova_prace_2016_Vaclav_Kozdeba.pdf
  - Diplomova_prace_2016_Vaclav_Kozdeba.doc
  - Kopie_zadani_diplomova_prace_2016_Vaclav_Kozdeba.pdf
- Obrázky použité v diplomové práci
- Aplikace MyCluster
  - /MyCluster/ - složka obsahující spustitelný soubor aplikace a nápovědu
- Dotazník hodnocení aplikace
  - /Dotaznik_hodnoceni_aplikace/ - složka obsahující dotazník hodnocení aplikace