TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

BAKALÁŘSKÁ PRÁCE

Liberec 2013 Jaromír Šída

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2646 / Informační technologie
Studijní obor: 1802R007 / Informační technologie

Příspěvek nasazení dataminingu k bezpečnosti ve společnosti

Datamining contribution to the world security

Bakalářská práce

Autor: Jaromír Šída

Vedoucí práce: RNDr. Klára Císařová, Ph.D.

V Liberci 10. 5. 2013

Zadání

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou 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 bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou 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.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Poděkování

Tímto bych chtěl poděkovat paní RNDr. Kláře Císařové za poskytnutí mnoha cenných rad, připomínek a informací k vypracování této práce. a za ochotný a milý přístup po celou dobu spolupráce.

Rovněž bych chtěl poděkovat mjr. Karlu Mauderovi z Odboru analytiky, Krajského ředitelství policie v Libereckém kraji, za nápomoc a poskytnutí informací z reálného prostředí kriminality.

Abstrakt

Tato bakalářská práce se zabývá využitím metod dataminingu k posílení bezpečnosti ve společnosti. V teoretické části jsou nejprve představeny základní pojmy z oblasti dataminingu, metodika CRISPM -DM a její jednotlivé fáze. Dále je pozornost upřena na metody shlukové analýzy, kterých se v tomto odvětví také používá. Shluková analýza je zde prezentována především z hlediska využití v dataminingu, a tomu je přizpůsobeno množství a skladba obsahu. V práci jsou zmíněny například metriky, hierarchickém shlukování a jeho aglomerativní i divizivní variantě, nehierarchickém shlukování a na závěr jsou detailněji popsány algoritmy K-Means a TwoStep.

Praktická část se zabývá takzvanou případovou studií, tedy na zadaných, „školních“, datech budou aplikovány metody dataminingu za účelem zjištění ideálního pokrytí oblasti daným počtem hlídek, vytvoření modelu schopného identifikovat potencionálně související případy trestné činnosti a jako poslední krok jsou popsány případy kapesních krádeží z hlediska času a místa spáchání.

Dalším bodem praktické části bakalářské práce je vlastní implementace algoritmu K-Means, který je v tomto případě speciálně upraven za účelem použití na dataminingových úlohách a schopnosti porovnat jeho výsledky s výsledky z programu IBM SPSS Modeler 14.2. Součástí implementace je i uživatelské rozhraní, které umožňuje základní editace vstupních parametrů algoritmu.

V rámci zhodnocení řešení je popsána situace v praxi, která byla zjištěna díky konzultaci s Krajským ředitelstvím policie Libereckého kraje. Dále jsou v práci zmíněny dosavadní přístupy, budoucí záměry, omezení vyplývající ze situace v České Republice a omezení vyplývající z povahy některých deliktů. Práce stručně seznamuje se systémem VICLAS, který policie používá k evidenci násilných trestných činů a následnému vyhledávání souvisejících kriminálních případů.

Klíčová slova

datamining, shluková analýza, vytěžování dat, k-means, twostep

Abstract

This thesis is focused on the utilization of methods used in datamining in the field of crime prevention. In the theoretical part of this thesis there is a basic introduction to the datamining and its methods and approaches such as CRISP-DM methodology with a description of it's phases. After the datamining is introduced, the theoretical part is dedicated to the methods of cluster analysis which are also used in the datamining. The aim of the topic is to cover all the basic information about cluster analysis from the perspective of datamining. We will discuss the term *metric*, *hierarchical clustering* within it's both versions – agglomerative and divisional, *non hierarchical clustering* and finally there will be chapters regarding algorithms K-Means and TwoStep in more detailed way.

The practical part will be focused on “so called” case study, which means that on the given data, will be applied some of the methods of datamining in order to be able to efficiently cover the selected area with a desired amount of police patrols, to be able to discover cases that are somehow similar, and to describe the cases of pickpocket thefts in the meaning of time and place when it happens.

The next part of the practical part is to implement a very own version of the K-Means algorithm, which in this case will be modified to meet the requirements of the usage in the datamining and to have the option to easily visually compare the results from our implementation and the results of the implementation in the IBM SPSS Modeler 14.2. This implementation also involves user interface that will give the user the option to modify basic input arguments of the algoritm.

As a part of the conclusion of this thesis there will be presented a current state and methods used in this field of interest in the “real world”. We gathered this information thanks to the meeting that we had with the employee of the Directory of Czech Police of the Liberec Region. There will be mentioned current approaches, future intentions, the restrictions that came from the situation in the Czech Republic and the restrictions that came from the nature of the torts. The database system called VICLAS, where the records of the violent crimes are created and stored with the ability to detect similar cases, will be also mentioned in the end.

Keywords

datamining, cluster analysis, data harvesting, k-means, twostep

Obsah

Seznam ilustrací

1 Úvod

S masivním rozmachem počítačů a výpočetní techniky obecně, jenž provázel společnost v průběhu druhé poloviny 20. století, se i datová úložiště dočkala stále většího a většího množství dat, která se v nich uchovávala. Je tedy pochopitelné, že před více než 20 lety vznikla oblast zájmu, nesoucí označení *„Dolování dat“,* neboli datamining, známý také jako *Vytěžování informací* (*Information Harvesting*) či *Objevování znalostí v databázích (Knowledge Discovery in Databases*).

Datamining nachází široké uplatnění v mnoha oblastech lidské činnosti, jako je:

- **Finančnictví** Snaha využít dosavadní zkušenosti s klienty a „předvídat“ rizikovost poskytnutí služeb. Takovými službami mohou být půjčky, hypotéky apod.
- **Marketing** Například u plošných nabídek produktů je celková procentuální úspěšnost nízká, proto je hlavním cílem dostatečně vymezit cílovou skupinu a tím optimalizovat výdaje použité na reklamní kampaň.
- **Komunikace** Filtrování zpráv, jenž nesou charakteristiky SPAMu. Expertní systémy se zde „učí“ charakteristické znaky spamu, přičemž znalostní výstup je použit u rozhodování, zda nově příchozí zpráva je SPAM, či ne.
- **Bezpečnost** Pomoci metod dataminingu se prochází data obsahující záznamy o kriminální činnosti, a hledají se případy trestné činnosti vykazující podobný charakter. Tyto případy jsou poté kandidátem pro to, být přezkoumány odborníky.
- **Zdravotnictví** Na základě znalostí o pacientech a znalostí o chorobách, jejich příčinách a symptomech, lze definovat rizikové skupiny pro různá onemocnění a činit kroky prevence.

Z výše uvedeného výčtu je vidět rozmanitost využití dataminingu, což mě společně s aktuálností přimělo, si toto téma zvolit pro svou bakalářskou práci. Ze zmíněných oblastí jsem si zvolil bezpečnost, jelikož je z mého pohledu nejatraktivnější.

2 Podrobnější představení problematiky

V této sekci budou představeny základní pojmy a přístupy v oblasti dataminingu společně se stručným popisem metodiky CRISP-DM. Dále bude obsaženo téma shlukové analýzy a na závěr algoritmy K-Means a TwoStep Vše v souladu se zadáním bakalářské práce.

2.1 Pojem datamining

2.1.1 Definice pojmu a řešitelský tým

Pokud bychom chtěli najít definici dataminingu, pravděpodobně bychom nejčastěji narazili na formulaci „*Netriviální získávání implicitních, dříve neznámých a potencionálně užitečných informací*“.

Lze to pochopit tak, že zadavatel projektu má pouze cíl, ale sám o sobě netuší, jaké závislosti se v datech skrývají, a co by bylo možné najít. Tomu se říká definování manažerského problému. Takto definovaný problém je však třeba přeformulovat tak, aby se dal pojmout z perspektivy DM. Tento krok je náplní expertů z oblasti dolování dat. Je tedy vidět, že na řešení daného projektu se podílí hned několik skupin participantů. Zaprvé jsou zde experti z domény projektu, kteří se snaží přiblížit danou problematiku expertům z oblasti metodik DM, jenž tvoří druhou skupinu. Často se v projektech vyskytují také experti na data poskytnutá danou organizací.

2.1.2 Typy úloh

V dataminingu se řeší tři základní typy úloh [1]:

- Klasifikační / Prediktivní
- Deskriptivní
- Hledání „*nugettů*“

Klasifikační (prediktivní) úlohy mají za cíl vytvořit model tak, aby byl schopný posuzovat nové případy (pozorování). Zpravidla se tento model validuje pomocí testovacích dat, kde se zkoumá úspěšnost modelu. Pozornost je zde zaměřena především na co nejlepší klasifikaci (predikci) a maximální pokrytí oblasti zájmu, kdy oblast zájmu vychází z konceptu vytvořeného řešitelským týmem.

Deskriptivní úlohy mají za úkol zpracovat vstupní data tak, aby výstupem modelu bylo popsání konkrétních jevů v dané problematice. To znamená, že pokud šlo v případě klasifikační úlohy o zařazení pozorování do určité skupiny, deskriptivní úloha má za cíl tuto skupinu popsat, přičemž pozornost je upřena spíše na dominantních skupiny v oblasti zájmu. Speciální důraz je pak kladen na srozumitelné vyjádření znalostí, například ve formě vizuálních prostředků. S danými znalostmi totiž bude zacházet zadavatel, který by leckteré výstupy z modelu nemusel pochopit.

V úlohách, kde se hledají takzvané „*nuggety*“, jde o vyhledání nových a překvapivých znalostí, které nemusí nutně pokrývat celý problém nebo většinu oblasti zájmu.

2.1.3 Techniky DM

V procesech dataminingu se používá několik různých technik. Jedná se o techniky statistické či matematické. Pro představu si jich pár vyjmenujeme:

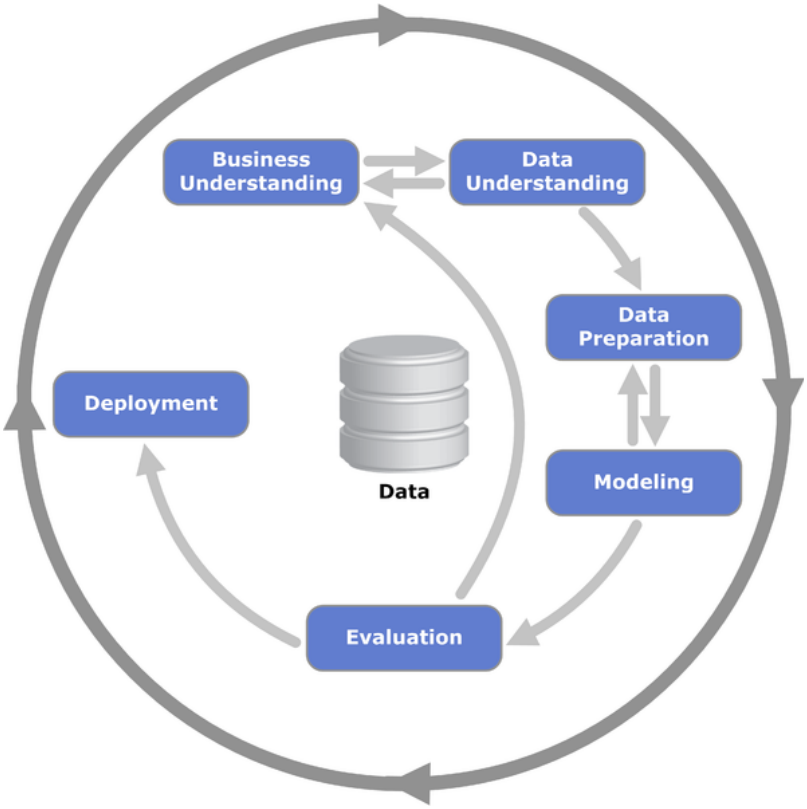
- **Rozhodovací stromy** – strom, který segmentuje oblast zájmu na základě jednoduchých pravidel.
- **Seskupování –** statistická metoda, která se snaží vytvářet shluky pozorování, které mají podobné charakteristiky.
- **Neuronové sítě** – snaha převedení „*lidského chování*“ do prostoru klasifikace a rozhodování. Používáno především v klasifikačních / prediktivních úlohách.
- **Bayesovské klasifikátory** – založené na pravděpodobnostních metodách. Vychází z Bayesovy věty.

2.1.4 Metodika CRISP-DM

Na poli dataminingu vznikalo několik konkurenčních metodik. Tyto metodiky byly zpravidla utvářeny společnostmi, kteřé měly vlastní software ke zpracovávání úloh, a sloužily tedy výhradně pro tento software. Takovými metodikami jsou například 5A (SPSS) či SEMMA (firma SAS). Na druhou stranu jsou zde nezávislé přístupy, utvořené skupinou komerečních subjektů, jenž se vyznačují snahou vytvořit unvierzální přístup k úlohám v dataminingu. Jednou z nezávislých metodik je také CRISP- DM, se kterou přišla skupina složená z firem Daimler Chrysler, ISL (software Clemetine), NCR (dodavatel datových skladů) a holanské pojišťovny OHRA.

2.1.5 Šest kroků CRISP-DM

CRISP-DM (*Cross Industry Standard Process for Datamining*), stejně jako ostatní metodiky, definuje jednotlivé kroky a fáze při řešení úloh. Výsledek kroku je přímo ovlivňován stavem v kroku minulém, avšak nejedná se vyloženě o cyklus, jelikož cílový krok se určuje na základě výsledku současného kroku. Proces je zobrazen na Ilustraci 1.



Ilustrace 1: Fáze a přechody v CRISP-DM
Zdroj: http://en.wikipedia.org

Metodika CRISP-DM se skládá ze šesti kroků životního cyklu projektu:

1. Porozumění problematice (*Business Understanding*)
2. Porozumění datům (*Data Understanding*)
3. Příprava dat (*Data Preparation*)
4. Modelování (*Modeling*)
5. Vyhodnocení (*Evaluation*)
6. Nasazení (*Deployment*)

První krok, *porozumění problematice*, je vstupní branou pro úlohu. Jedná se o definování manažerského problému, který je následně „přeložen“ do řeči dataminingu. Definují se zde závislosti, zkoumají možnosti získání dat a vyhodnocuje se potencionální přínos v dané problematice na základě předběžných analýz. V této fázi probíhá veškerá komunikace mezi experty z oblasti dolování dat a experty na danou problematiku.

Druhý krok, *porozumění datům*, se zabývá zkoumáním dostupných dat. Provádí se první náhledy do dat, hledá se jejich význam, odhaduje se užitečnost a hledají se první odhady vazeb. Mohou se zkoumat základní parametry jako střední hodnoty, směrodatné odchylky a podobně. V této fázi s spolu komunikují experti z oblasti dolování dat a experti na poskytnutá data.

Třetí fází je *příprava dat.* V této fázi jsme již data pochopili, a proto je možné je připravovat pro použití do konkrétních analytických metod. Taková příprava spočívá v jejich selekci, transformaci, agregaci čištění, kategorizování, doplňování prázdných hodnot či například odvozování dat nových z dat stávajících.

Data upravená ve třetí fázi postupují dále do části nesoucí název *modelování*. Zde se aplikují analytické algoritmy. Jedná se o iterativní proces, kdy může být potřeba několik pokusů a několik různých nastavení vstupních parametrů, než je dosaženo požadovaných vlastností. Také výstup z modelování může zapříčinit návrat do kroku přípravy dat za účelem přizpůsobení informací vstupujících do modelování.

Předposlední fáze, *evaluation,* spočívá v tom, že jsou výsledky expertního týmu, zabývajícího se vytěžováním informací, předloženy manažerům – zadavatelům a jejich expertům ke zhodnocení. Nyní se ukáže, nakolik pro ně budou výsledky přínosné a překvapivé. V krajních případech může nevyhovující výsledek vést k přeformulování problému a opakování celého procesu.

Jsou-li zadavatelé spokojeni s nalezenými znalostmi, zbývá už jen poslední krok a to *nasazení*. Nasazení je praktické využití znalostí. V případě deskriptivních úloh může mít podobu grafů, tabulek, závěrečné zprávy či podobných forem. V případě úloh klasifikačních se pak jedná o uživatelské rozhraní, eventuálně implementace jiným způsobem tak, aby se dané znalosti daly využívat osobami, které buď v procesu dosud nefigurovaly, nebo nemají žádné analytické znalosti.

2.2 Shluková analýza

V odstavci 2.1.2 jsou zmíněny typy úloh, které se v dataminingu řeší, přičemž jednou z nich byla úloha *deskriptivní*. Právě tento typ je často řešen pomocí *shlukové analýzy*, součásti statistického přístupu k řešení zadání v dataminingu.

Shluková analýza, jak již název napovídá, slouží k vyhledávání a utváření shluků ve vstupních datech dané úlohy. Shlukem rozumíme množinu objektů, jež vykazuje podobné vlastnosti.

2.2.1 Modelový příklad

Řekněme, že máme úlohu, která si klade za cíl najít neznámé hypotézy v oblasti prodeje aut, které by mohly být využity například k zacílení reklamní kampaně. Pro názornost byl záměrně zvolen příklad, jehož výsledky nebudou nijak překvapivé a budou korespondovat se všeobecně známými fakty.

Našimi *vstupními daty* mohou být informace o věku kupujícího, počtu jeho dětí, pohlaví, svobodný (ano/ne), výše příjmu, typ (sportovní, rodinné, SUV,...) a cena vozidla nového. Každou položku popisující prodej budeme nazývat atribut. Atributům se ve statistické oblasti říká také příznaky, pro srozumitelnost budeme však používat jednotné označení. Vektor těchto atributů bude reprezentovat náš objekt. Objektem je tedy záznam o nákupu vozidla určitou osobou, o němž máme patřičné údaje.

Metody shlukové analýzy nám mají pomoci nalézt v těchto datech souvislosti, o kterých předem nic nevíme, a které bude možné později popsat ve srozumitelné a v co možná nejakurátnější formě. Tento příklad bude použit v následujících odstavcích tak, aby se dosáhlo co n představy na straně čtenáře.

2.2.2 Základní pojmy

Vektor a atributy

Mějme vektor: $X=(x_1, x_2, x_3, ..., x_n)$, kde X nám reprezentuje daný objekt a $x_1 ... x_n$ nám reprezentují jednotlivé atributy o počtu n . Použijeme-li strukturu dat z modelové úlohy, pak takový objekt a jeho vektor atributů může vypadat následovně: $X=(40, 2, 'Muž', Ne, 38000, 'rodinné', 550000)$

Matice vstupních dat

Maticí vstupních dat budeme uvažovat strukturu, která ve svých řádcích obsahuje jednotlivé objekty a sloupce jsou tvořeny jejich atributy. Matici o pěti objektech transformovanou do tabulky ilustruje tabulka 1. Do této tabulky byly použity hodnoty, jež by mohly tvořit vstup zmíněné modelové úlohy, přičemž bylo dodrženo jejich pořadí.

Věk	Počet dětí	Pohlaví	Svobodný	Výše příjmu (Kč)	Typ nového vozidla	Cena nového vozidla (Kč)
18	0	Muž	Ano	17 000	sportovní	80 000
40	2	Muž	Ne	38 000	rodinné	550 000
28	1	Žena	Ne	25 000	rodinné	370 000
35	2	Muž	Ne	42 000	SUV	650 000
22	0	Muž	Ano	30 000	sportovní	500 000

Tabulka 1: Tabulka transformované matice dat

Typy atributů

Jak je vidět v tabulce 1, atributy mohou být různého typu.

1. Prvním typem jsou atributy **reálné** neboli **kontinuální**, které se vyznačují tím, že jsou tvořeny intervalem reálné osy. V naší modelové úloze jsou příkladem *výše příjmu* a *cena nového vozidla*.

2. Druhý typ je tvořen **spočetnou** nebo **konečnou** skupinou čísel. Do této kategorie patří *počet dětí* a *věk.*

3. Třetí typ se skládá z množiny atributů o dvou stavech – pravda/nepravda. Takové atributy nazýváme **dichotomickými**. Naším příkladem je údaj *svobodný*.

4. Čtvrtý, poslední typ, je typem **kategorickým**, kde daný údaj může nabývat jedné hodnoty z konečné množiny přípustných pojmenovaných hodnot. Zástupcem toho typu je *typ nového vozidla* a *pohlaví*.

V algoritmech, používaných ve shlukové analýze, se velice často snažíme reprezentovat všechny atributy pomocí čísel. A proto se například třetí typ atributů reprezentuje jako 0 – nepravda / 1 - pravda. U čtvrtého typu zase může docházet k transformaci kategorie o p variantách do p sloupců, přičemž právě jeden (nejsou-li ve vstupních datech prázdné hodnoty) z nich nabývá hodnoty „1“, zatímco zbylé nabývají hodnot „0“.

Standardizace atributů

Podíváme-li se opět na tabulku 1 a budeme-li si všímat údajů *výše příjmu* a *cena nového vozidla,* pak je možné zpozorovat rozdíl jednoho řádu napříč všemi hodnotami. Vezmeme-li v potaz, že pokud bychom mezi sledované atributy přidali ještě jeden, jehož rozptyl by byl v řádu desetin jednotek, pak bychom se ještě výrazněji setkali se situací, kdy by atribut s většími hodnotami lehce dominoval nad ostatními atributy, čímž by výrazně zkreslil vyhodnocování podobnosti. Proto je potřeba hodnoty standardizovat. Toho docílíme následujícím vztahem:

$$y_j^i = \frac{x_j^i - \bar{x}_j}{s_j}$$

Kde y_j^i je standardizovaný j-tý atribut i-tého objektu, x_j^i je j-tý atribut o původní hodnotě, vztahující se k i-tému objektu. Dále $\bar{x}_j$ je střední hodnotou j-tého atributu pro všechny objekty a s_j je směrodatnou odchylkou pro atribut j, opět přes všechny objekty. Pro úplnost doplním vzorce k výpočtu posledních dvou zmiňovaných veličin, kde n značí počet všech objektů v množině vstupních dat.:

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_j^i \quad , \qquad s_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_j^i - \bar{x}_j)^2}$$

V tabulce 2 jsou uvedeny dva zmíněné atributy, jenž byly standardizovány a zaokrouhleny na dvě desetinná místa.

Věk	Počet dětí	Pohlaví	Svobodný	Výše příjmu	Typ nového vozidla	Cena nového vozidla
18	0	Muž	Ano	-1,50	sportovní	-1,78
40	2	Muž	Ne	0,85	rodinné	0,61
28	1	Žena	Ne	-0,60	rodinné	-0,30
35	2	Muž	Ne	1,29	SUV	1,12
22	0	Muž	Ano	-0,04	sportovní	0,36

Tabulka 2: Tabulka vstupních dat po aplikaci standardizace

2.2.3 Vybrané způsoby hodnocení podobnosti objektů

Klíčová problematika shlukové analýzy se týká toho, jak poznat, že některé objekty jsou si podobné. Způsobů, jak vyjádřit onu podobnost, je celá řada, avšak ani o jednom se nedá říci, že je univerzálně vhodný pro všechny úlohy. Pro tuto práci byly vybrány dva způsoby, a to pomocí koeficientů asociace a pomocí metrik.

Koeficienty asociace objektů

Předznamenáváme, že shluková analýza má vlastní pojetí koeficientů asociace a na rozdíl od statistiky, kde se koeficienty asociace zaměřují na vztahy v rámci jednotlivých atributů [2], zde koeficienty vyjadřují míru podobnosti jednotlivých objektů za použití všech atributů. Tento způsob odhadování podobnosti se tedy týká objektů reprezentovanými pouze dichotomickými atributy a k vizualizaci nám pomáhá asociační tabulka. (viz. Tabulka 3)

		X^i	
		1	0
X^j	1	a	b
	0	c	d

Tabulka 3: Asociační tabulka

Ve výše zobrazené tabulce je možné vidět dva objekty X^i a X^j, které obsahují p atributů.

- **a** – počet atributů, kde oba objekty dosahují hodnoty 1
- **b** – počet atributů, kde objekt X^i dosahuje hodnoty 0, přičemž objekt X^j dosahuje hodnoty 1
- **c** – počet atributů, kde objekt X^i dosahuje hodnoty 1, přičemž objekt X^j dosahuje hodnoty 0

- **d** – počet atributů, kde oba objekty dosahují hodnoty 0

S těmito hodnotami dále pracují konkrétní koeficienty. Pro bližší představu je zde uveden Sokalův a Michenerův koeficient S_{ms}, známý také jako „simple matching coefficient“:

$$S_{sm}=\frac{a+d}{a+b+c+d}$$

Metriky

Geometrický model je do posuzování vnesen používáním metrik. Máme-li objekt X^i, mající p atributů, tak ho lze „zobrazit“ do p-rozměrného metrického prostoru. Metrický prostor je tvořen dvojící (M,ρ), kdy M je libovolná neprázdná množina a ρ, označované jako metrika, je zobrazení:

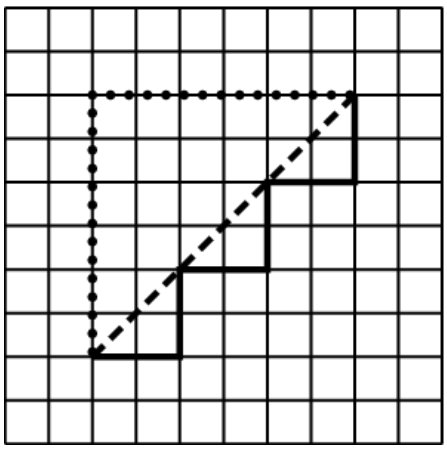
$$\rho : M \times M \rightarrow \mathbb{R}$$

přičemž pro objekty A, B a C z množiny M musí platit:

1. $\rho(A,B)=0 \Leftrightarrow A=B$
2. $\rho(A,B)\geq 0$
3. $\rho(A,B)=\rho(B,A)$
4. $\rho(A,C)\leq \rho(A,B)+\rho(B,C)$

Pro práci byly vybrány tři metriky, které jsou níže popsány. Jedná se o metriky: Euklidovskou, Manhattanskou a Chebyshevovu metriku.

- Euklidovská: $d_e(A,B)=\sqrt{\sum_{i=1}^{p}(a_i-b_i)^2}$
- Manhattanská: $d_m(A,B)=\sum_{i=1}^{p}|a_i-b_i|$
- Chebyshevova: $d_{ch}(A,B)=max|a_i-b_i|$



Ilustrace 2: Euklidovská a Manhattanská metrika
Zdroj: http://everythingmaths.co.za

Na ilustraci 2 je zobrazena euklidovská metrika (čárkovaně), o které se dá říct, že je nejpoužívanější [3], versus manhattanská metrika (tečkovaně a souvislou čarou) v prostoru o dvou dimenzích, propojující dva body.

2.2.4 Hierarchické aglomerativní metody shlukování

Poté, co proběhlo seznámení s tím, jakým způsobem zacházet s atributy jednotlivých objektů, a také jak vyhodnocovat podobnost objektů, je načase začít se věnovat samotným metodám, jež si kladou za cíl podobnosti vyhodnocovat a objekty řadit do podobných skupin – shluků.

První skupina shlukovacích metod je tvořena metodami hierarchickými – aglomerativními. Mějme vstupní data tvořena n objekty. Obecný princip metody s těmito vstupními daty spočívá v následujícím:

1. Vstupní data vytvoří n shluků, tedy každý shluk bude tvořen právě jedním objektem z množiny vstupních dat.

2. V každém kroku se provede sloučení dvou shluků, kdy o tom, které dva shluky budou sloučeny, rozhodne *koeficient nepodobnosti shluků* a to tak, že se vybere nejmenší hodnota z koeficientů vypočítaných ke každé dvojici shluků. Koeficienty nepodobnosti shluků budou popsané v navazující části.

3. Krok 2 probíhá tak dlouho, dokud nebudou všechny objekty sloučeny do jednoho shluku.

Koeficient nepodobnosti shluků

Podobně, jako bylo potřeba definovat způsob, jakým budeme určovat míru (ne)podobnosti objektů, je potřeba mít schopnost vyhodnotit, nakolik si jsou dva shluky (ne)podobné. Způsobů je opět více a níže budou uvedeny tři nejintuitivnější. Stejně jako u koeficientu podobnosti objektů, i zde je potřeba splnit základní podmínky. Proto pro funkci D, reprezentující hodnotu koeficientu a shluky A, B platí:

$$D(A,A)=0,$$
$$D(A,B)\geq 0,$$
$$D(A,B)=D(B,A)$$

Metoda nejbližšího souseda

Zde je nepodobnost shluků vyjádřena (ne)podobností dvou objektů. První objekt je součástí prvního shluku, druhý objekt náleží do shluku druhého a hledá se ten pár, který si je nejvíce podobný, což například z geometrického hlediska, při použití metrik jako koeficientu podobnosti objektu, je ten pár, jenž má nejmenší vzájemnou vzdálenost. Matematické vyjádření pro n objektů, značených O_i a O_j, kdy $(i, j = 1,2,3,\ldots,n)$ a shluky A, B je následující:

$$D(A,B) = \min_{O_i \in A \; O_j \in B} \{d(O_i, O_j)\}$$

Metoda nejvzdálenějšího souseda

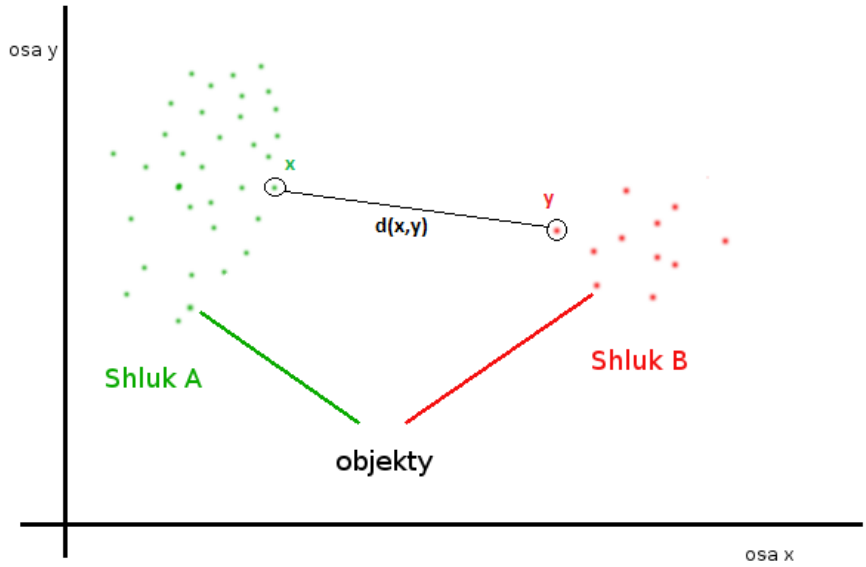
Analogicky k metodě nejbližšího souseda lze zavést metodu nejvzdálenějšího souseda, kdy se hledá ten pár, který je od sebe nejvíce vzdálený. Matematicky vyjádřeno:

$$D(A,B) = \max_{O_i \in A \; O_j \in B} \{d(O_i, O_j)\}$$

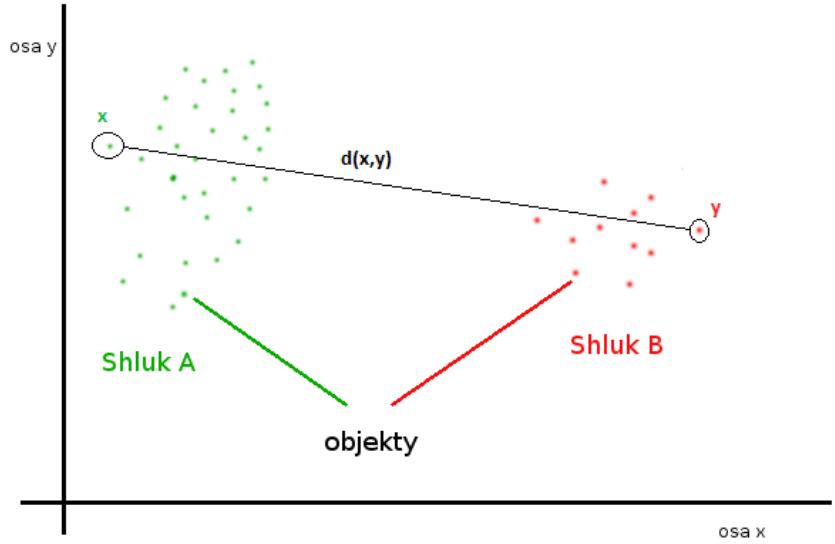
Metoda centroidní

Třetí metodou, jak určit míru (ne)podobnosti shluků, je takzvaná metoda centroidní. Ta spočívá v tom, že pro daný shluk je vypočítán centroid, což je objekt, jehož hodnoty jednotlivých atributů jsou tvořeny středními hodnotami daného atributu napříč všemi objekty onoho shluku. Poté, co se vypočítají centroidy, jsou na ně aplikovány koeficienty podobnosti objektů.

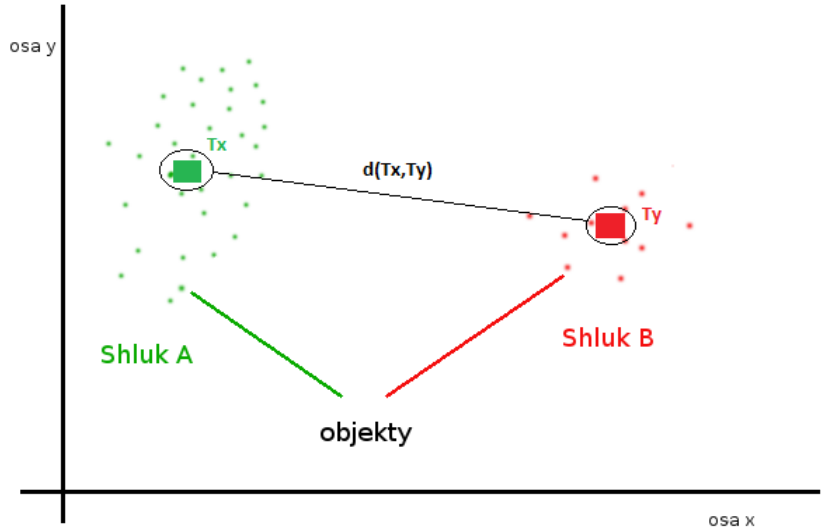
Všechny tři zmíněné metody lze nalézt na ilustracích 3-5.



Ilustrace 3: Metoda nejbližšího souseda



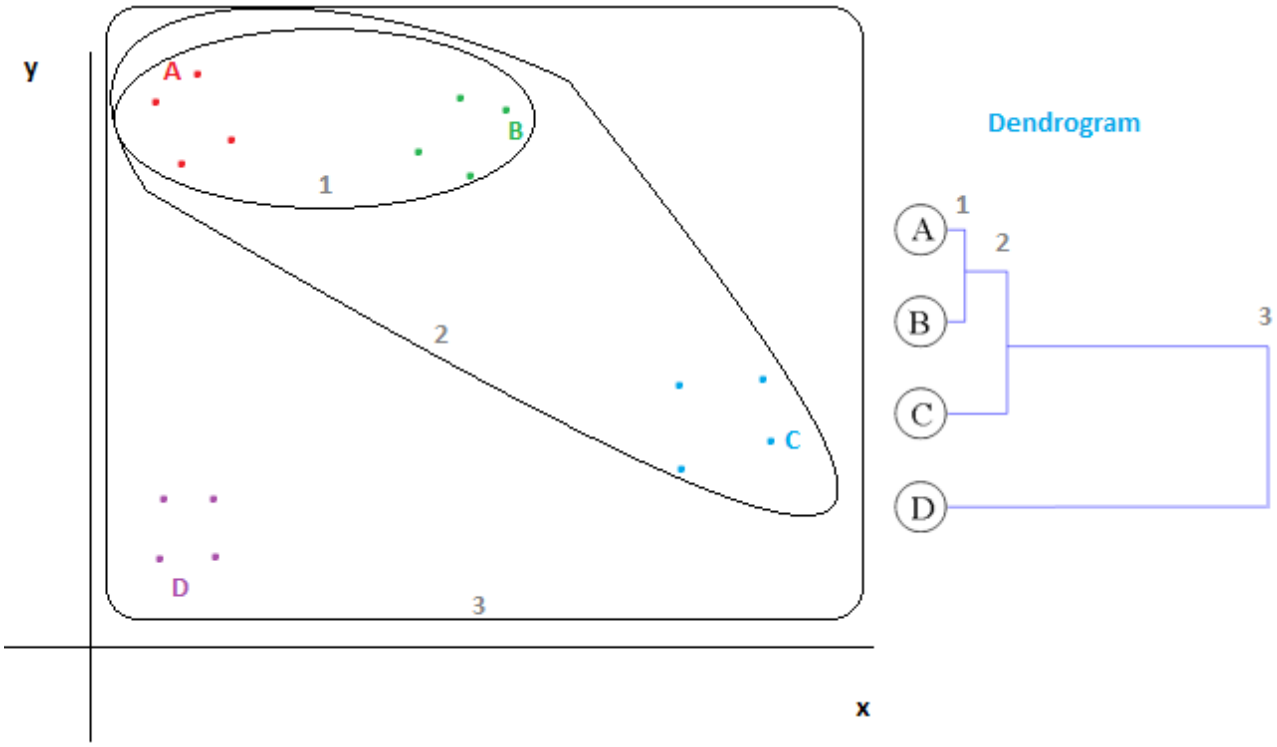
Ilustrace 4: Metoda nejvzdálenějšího souseda



Ilustrace 5: Metoda centroidní

Zobrazení kroků shlukování na dendrogramu

Dendrogram je diagram, zobrazující jednotlivé kroky shlukování. Na ilustraci_6 je možné vidět názorný příklad zakreslení procesu hierarchického aglomerativního procesu do dendrogramu, společně s dvourozměrnou plochou zobrazující jednotlivé objekty a shluky.



Ilustrace 6: Ukázkový dendrogram pro hierarchické aglomerativní shlukování

2.2.5 Hierarchické divizivní metody shlukování

Zatímco aglomerativní metody vycházely ze shluků, obsahující pouze jeden objekt, divizivní metody mají zpočátku jediný shluk, obsahující všechny objekty. V každém kroku rozkládání se jeden shluk rozdvojí. Toto pokračuje až do té doby, dokud nemáme počet shluků roven počtu objektů.

Jako jeden z mnoha způsobů jak postupovat v případě těchto metod bude uveden MacNaughton-Smithův algoritmus [4]:

1. Počáteční shluk nechť pro nás bude shluk A tvořený všemi objekty.
2. Ve shluku A nalezneme objekt s největší průměrnou vzdáleností ke všem ostatním objektům.
3. Z nalezeného objektu vytvoříme nový shluk B .
4. Dále počítáme pro každý objekt v A průměrnou vzdálenost d_a mezi ním a objekty shluku A a průměrnou vzdálenost d_b mezi ním a objekty ve shluku B . Pro každý objekt si pamatujeme rozdíl průměrných vzdáleností $d_a - d_b$.
5. Vybereme ten objekt A , jehož hodnota rozdílu z bodu 4 je maximální.

6. Je-li rozdíl kladný, přiřadíme tento objekt do shluku B. Je-li záporný, ukončíme rozdělování.
7. Je li počet shluků stále menší než počet všech objektů, vybereme další shluk a ten dělíme.

2.2.6 Nehierarchické metody shlukování

V nehierarchickém shlukování, jak již název napovídá, se neuplatňuje hierarchická struktura rozkladů. Metody tohoto shlukování musí hodnotit kvalitu rozložení objektů do shluků, a v případě potřeby toto rozložení modifikovat. K tomu slouží *funkcionál kvality rozkladu*.

Funkcionál kvality rozkladu

Tento ukazatel nám pomáhá dosáhnout optimálního rozložení shluků a prvků v nich tak, že hledáme místa, kde nabývají extrémů. Tyto funkcionály zpravidla vyhodnocují vlastnosti, jako například:

1. Vnitroshluková podobnost objektů
2. Izolovanost shluků
3. Rovnoměrnost rozložení objektů do shluků

Záleží na typu úlohy, kterou z vlastností chceme sledovat a podle které budeme určovat, zda se dá rozložení vylepšit, či nikoliv.

Volba optimálního počtu shluků

Jedná se o klíčový krok, jenž je třeba provést ještě před samotným shlukováním, nebo jej lze provádět současně s během shlukovacího algoritmu. Podle toho je možné dělit nehierarchické shlukování na:

Metody zachovávající počet shluků

Tyto metody se již nadále nezabývají rozhodováním, zda zvolený počet shluků je ideální, a pouze přehazují objekty mezi shluky na základě některého z funkcionálů. K tomu může být použit *součet čtverců chyb*, který lze pochopit jako reprezentaci součtu čtverců vzdálenosti objektů shluku od jeho těžiště. Toho využívá například K-Means algoritmus, o kterém bude řeč v navazující části.

Slabostí těchto metod může být zacházení s izolovanými body. Izolovaný bod je bod, který se výrazně odlišuje od všech ostatních, což v praxi může znamenat například chybu měření. Takový bod by pak sám o sobě zkonzumoval jeden shluk. Tyto metody jsou rovněž více závislé na volbě počátečních bodů, než metody optimalizační.

Metody optimalizující počet shluků

Tyto metody kromě organizace objektů do shluků také řeší jejich optimální počet. K tomu, aby mohly fungovat, je třeba jim poskytnout kromě počátečního počtu shluků také kritéria, za kterých se mají shluky slučovat, či rozdělovat.

2.2.7 Algoritmus K-Means

Algoritmus K-Means se řadí mezi nehierarchické metody zachovávající počet shluků v průběhu algoritmu. Jiný název pro tuto metodu je *MacQueenova metoda k-průměrů*.

Vlastnosti K-Means

K-Means vytváří disjunktní množiny objektů – shluků, které reprezentuje pomocí těžišť – typických objektů. Tyto typické objekty mohou být buď čistě vypočítány, a nebo to mohou být existující objekty ve vstupní množině. Jako koeficient podobnosti objektů (v tomto případě objektu a typického objektu daného shluku) je použita Eukleidovská metrika, což ho předurčuje primárně k řešení úloh, kde jsou vstupní atributy numerické. V případě kategoriálních dat je třeba vstup předzpracovat.

Počáteční typické objekty mohou být zadány několika způsoby. Buď se může jednat o prvních k objektů z množiny vstupních objektů, nebo se může jednat o zcela náhodný výběr, eventuálně se mohou použít metody, které se snaží o co nejrozptýlenější pozice typických objektů.

Kroky algoritmu

1. Výběr k počátečních objektů. Tyto objekty budou reprezentovat typické objekty každého shluku. Metody výběru jsou popsány výše.
2. V druhém kroku se jednotlivé objekty přiřazují k těm shlukům, k jejichž typickým bodům mají nejblíže.
3. Umístění typických bodů je přepočítáno na základě výpočtu obsahujícího i nově přidané objekty.
4. Pokud nedošlo k žádnému prohození, algoritmus se ukončí. V opačném případě se opakují kroky 2 – 4. V některých případech se zavádí i jiná podmínka ukončení algoritmu. Takovou může být buď maximální počet iterací, nebo porovnání rozdílu součtu čtverců chyb z předchozí iterace a aktuální iterace, přičemž pokud je rozdíl menší než určitá hranice, algoritmus končí.

Slabiny algoritmu

Výsledek pro stejnou množinu dat se může měnit v závislosti na několika faktorech. Tato metoda tedy není definitní, stejně tak jako mnoho jiných metod používaných ve shlukové analýze, což je do jisté míry její nevýhoda.

Faktory ovlivňující výsledek jsou například volba počátečních objektů či pořadí objektů ve vstupní množině. Další nevýhodou je zranitelnost před odlehlými objekty (outlinery). Takové body vychýlí těžiště, čímž výrazně ovlivní výsledek shlukování.

2.2.8 Algoritmus TwoStep

Vzhledem k tomu, že tento algoritmus nemá svou standardní všeobecně známou podobu, bude zde popisována varianta použitá v software *IBM SPSS Modeler 14.2*.

Vlastnosti algoritmu

Tento algoritmus je schopen zpracovat a vyhodnotit jak kontinuální, tak kategorické atributy při velkém počtu objektů, jednak pro svou paměťovou nenáročnost, a jednak pro rychlost [5]. Skládá se ze dvou kroků (odtud *TwoStep*).

První z nich je *pre-clustering*, druhý *clustering*. Počet shluků, do kterých mají být vstupní objekty rozřazeny, může být předem dán, nebo zjištěn automaticky pomocí některého z *informačních kriterií* (např. Bayesovské), určeného pro vyhodnocování modelů s různým počtem shluků. TwoStep nepodporuje objekty, které nemají některý atribut vyplněný hodnotou. Takové objekty neparticipují ve vlastním běhu algoritmu. Pro měření podobností objektů se používá *log-likelihood* funkce, založená na pravděpodobnostních metodách.

CF a CF-Tree

CF (*cluster feature*) je vektor hodnot charakterizující objekty v daném shluku. Obsahuje hodnoty jako počet objektů, rozptyl, střední hodnota. Složení tohoto vektoru je závislé na dané konkrétní implementaci.

CF-Tree je výškově vyvážený strom složený z CF. Má dva parametry: *počet objektů ve shluku* (označme si jako *B* – branching factor) a *práh* (označme jako *T* – treshold). Každý vnitřní (nekoncový) uzel má v sobě ukazatel na potomka (shluk) a jeho CF. Maximální počet potomků je roven *B*. Koncový uzel pak postrádá odkaz na potomka a pouze nese informaci o CF.

Fáze algoritmu

1. Pre-clustering: V této fázi se používá modifikovaný CF-Tree, který navíc jednotlivá CF obohatí o počty objektů s danou kategorií pro každý kategorický atribut. Díky definování *prahu T*, jenž je realizován koeficientem podobnosti objektů, je vstupní počet objektů zmenšen. Dojde tedy ke slučování podobných objektů. Stromová struktura navíc pomáhá rychlejšímu nalezení optimálního pod-shluku pro daný objekt. Pokud objekt nevyhovuje T, vytvoří nový pod-shluk a koncový uzel se stává uzlem vnitřním se dvěma potomky.

2. Clustering: Pod-shluky, vytvořené v první fázi jsou vstupem druhé fáze, která je upraví do finálního počtu shluků. Používá se hierarchických aglomeračních metod, které pracují efektivně pro snížený počet objektů, převedením je na pod-shluky. Je-li potřeba stanovit optimální počet shluků, tak se na výsledky shlukování používají informační kritéria.

Slabiny algoritmu

Stejně jako u *K-Means*, i zde záleží na pořadí objektů. Vliv odlehlých objektů může být eliminován již v průběhu *pre-clusteringu* pouhým nadefinováním hranice, při které je objekt považován za odlehlý.

3 Cíle praktické části

3.1 Případová studie

V praktické části bylo za úkol mimo jiné vypracovat jednu z případových studií. Šlo o úlohu, jež měla za úkol pokusit se nalézt v poskytnutých datech, obsahujících záznamy o různé kriminální činnosti, takové informace, které jsou potencionálně cenné a mohly by tak být použity k preventivním opatřením. Jelikož se jednalo o úlohu deskriptivní, tak těžiště bylo v aplikování algoritmů shlukové analýzy.

Body případové studie

- Vytvořit model vyhodnocující optimální rozložení daného počtu jednotek po městě
- Vytvořit model schopný nalézt související případy vloupání do obytných budov
- Zaměřit se na kapesní krádeže a popsat výskyt a množství kapesních krádeží v závislosti na času a lokalitě

3.2 Implementace algoritmu

V rámci řešení výše zmíněné studie jsme se setkali s algoritmy patřící do sféry shlukové analýzy. Jednalo se o algoritmy TwoStep a K-Means.

Kromě prostudování obou zmíněných byla součástí praktické části této práce implementace jednoho z nich – algoritmu K-Means, a to v libovolném programovacím jazyce. Tato implementace měla být posléze konfrontována s výsledky stejného algoritmu ve výše zmíněném programu.

3.3 Součást e-learningového kurzu

Posledním bodem praktické části bakalářské práce bylo zpracování případové studie a oblasti shlukové analýzy jako součásti kurzu na univerzitním výukovém portálu – e-learningu.

Bylo tedy potřeba se seznámit s administrací portálu, dané téma zpracovat „výkladovým“ způsobem, popsat případovou studii a společně s kontrolními otázkami přispět do kurzu „Datamining“ na e-learningu TUL.

4 Návrh řešení

V této části budou uvedeny zvolené přístupy k řešení případové studie a implementace algoritmu K-Means.

4.1 Případová studie

Na základě zadání byla úloha rozdělena do několika fází:

1. Příprava dat
2. Vytvoření modelu vyhodnocujícího optimální rozložení hlídek po dané lokaci
3. Vytvoření modelu hledajícího související případy vloupání do obytných budov
4. Vytvoření modelu zabývajícího se kapesními krádežemi.

4.1.1 Příprava dat

Počáteční fáze většiny dataminingových úloh se zabývá zpracováním dat. Vstupní data totiž zpravidla nebývají v takové formě, aby je bylo možné či vhodné, použít pro samotné modelování. Důvodem jsou různé druhy datových skladišť, odkud jsou tato data exportována a předána k analytickým účelům. Tato úloha toho nebyla výjimkou, a tak bylo potřeba nejdříve zjistit jaká data máme k dispozici.

Bylo nezbytné zajistit, aby IBM SPSS Modeler (dále odkazován jako „program“) správně identifikoval typy vstupních atributů a správně nakládal s prázdnými hodnotami. Jak se později ukázalo, právě toto byl problematický aspekt, jelikož automatické rozpoznávání typů atributů selhalo a prázdné hodnoty kategorických atributů byly nahrazeny jinou, výchozí, kategorií. A tak již první operace v programu vyžadovala manuální konfiguraci. Po správném nastavení uzlu, přijímajícího vstupní data, se naskytla možnost data prozkoumat a provést první transformace a jiné úpravy v datech.

Po nahlédnutí do dat bylo zjištěno, že ve výchozím stavu zůstat nemohou. Potřebné základní informace byly obsaženy v atributech složitějších. Příkladem takového atributu může být například *časové razítko*, ze kterého je možné vyextrahovat atributy jako čas, datum, týden, část dne, roční období a jiné „jednodušší“ atributy, mnohem vhodnější jako vstupní data do navazujících částí případové studie a modelů v nich použitých. Pro další fáze studie tedy došlo k doplnění dat o tyto atributy.

V momentě, kdy byla fáze přípravy dat hotova, zbývalo určit, co bude obsahem vyexportovaných dat. Jinými slovy byly vytvořeny filtry objektů a atributů. Tato filtrovaná data byla obsahem celkem tří separátních souborů, použitých v navazujících fázích studie.

4.1.2 Vytvoření modelu vyhodnocujícího optimální rozložení hlídek v dané lokaci

Naším hlavním cílem v tomto modelu bylo nalezení optimálního, rovnoměrného pokrytí dané oblasti určitým počtem hlídek. Termín rovnoměrný není myšlen jen pouze ve smyslu plochy, ale také četnosti kriminální činnosti na dané ploše. To znamená, že například centrum, kde se dá předpokládat vyšší kriminalita, budou střežit dvě hlídky, přestože jinou, odlehlejší, oblast o stejné rozloze bude střežit pouze jedna hlídka.

V této části bylo potřeba nejprve nadefinovat jednotlivé úseky dne, jako je ráno, den, noc. Odhad toho, pro jaké hodiny platí, že se jedná o danou část dne, byl proveden na základě intuice. Za účelem ověření tohoto prvotního odhadu mezních hodin, bylo nahlédnuto do dat a hlavní důraz byl kladen na četnost jednotlivých případů v průběhu dne. První odhad bylo třeba pozměnit tak, aby extrémy v četnosti daných záznamů odpovídaly zvoleným mezním hodnotám.

Po filtraci záznamů spadajících do daných částí dne, bylo potřeba zvolit vhodný shlukovací algoritmus. Byly aplikovány čtyři různé přístupy.

První pokus spočíval ve snaze využít algoritmu K-Means s tím, že počty shluků (hlídek) budou určeny analytikem. Ten se prokázal být až příliš náchylný pro odlehlé případy, což mělo za následek vyčlenění celé jedné hlídky do oblasti s příliš nízkým počtem případů. Jelikož početní rozložení případů na danou hlídku nemělo dosahovat příliš velkých rozdílů, byl tento efekt nežádoucí.

Dále byl použit algoritmus Kohonenovy mapy, zástupce segmentačních nástrojů využívajícího neuronovou síť. Jednalo se o prvního ze dvou zástupců algoritmů, pomocí nichž byl praktikován pokus zjistit, zda lze tuto úlohu obohatit o určení optimálního počtu hlídek pro daný úsek dne, bez nutnosti patrnějšího přizpůsobování dat. Výsledek však v tomto případě nebyl vyhovující, jelikož bylo výsledkem více shluků s větším rozptylem počtů případů na shluk, než bylo očekáváno.

Třetím přístupem bylo využití algoritmu TwoStep s automatickým určením optimálního počtu shluků. Byl to tedy druhý zástupce algoritmů pro potencionální doporučování počtu hlídek. Ani zde nebyly výsledky uspokojující. Počet shluků se už sice více blížil naší představě, avšak rozdíl v četnosti případů na shluk byl stále ještě vysoký. Ani parametrizování algoritmu, ve smyslu změny informačních kriterií a metriky, nemělo žádoucí efekt.

Čtvrtým, posledním, přístupem byl opět algoritmus TwoStep, tentokrát však oproštěný o snahy automatického určování počtu shluků. V porovnáním s algoritmem K-Means, kde byl počet shluků rovněž zadáván analytikem, měl o poznání lepší výsledky a prokazoval větší odolnost proti odlehlým případům. Toto si vysvětlujeme použitím *log-likelihood*, podobnostní míry, zavádějící pravděpodobnostní přístup do řešení náležitosti objektu do shluku. Tento poslední přístup se stal přístupem finálním.

4.1.3 Vytvoření modelu hledajícího související případy vloupání do obytných budov

Třetí fáze případové studie spočívala ve vytvoření modelu schopného nalézt takové případy vloupání do obytných budov, které vykazují znaky podobnosti.

Prvním důležitým krokem bylo odhadnutí počtu skupin případů, které spolu souvisí. Z pochopitelných důvodů nebylo možné toto odhadnout bez pomoci nástrojů programu. Vzhledem k tomu, že nebyla žádná představa o tom, zda vůbec nějaké podobné případy existují, byly zvoleny následující způsoby:

Postupné zkoušení různých počtů shluků (skupin) pro K-Means

První nápad spočíval v tom, že budeme modifikovat parametry algoritmu K-Means, konkrétně parametr cílového počtu shluků. K tomu, aby bylo možno posléze nalézt ideální počet, bylo nutné se spolehnout na evaluační mechanismy shluků, které jsou implementované v jednotlivých algoritmech programu. Nicméně výsledky kvalitativního ohodnocení segmentace se nelišily tak výrazně, aby důvěra mohla být vložena právě do tohoto způsobu. Fakt, že segmentace se dle programu pohybovala na rozmezí ohodnocení „Poor“ a „Fair“, neboli „Špatná“ a „Dostačující“ pro všechny zadané počty, také nenaznačoval, že by pouze tento přístup byl dostačující. Jediný přínos, který tento způsob měl, bylo získání představy o tom, že ideální počet shluků se bude pohybovat někde kolem čísla deset.

Použití algoritmu TwoStep s možností automatické detekce optimálního počtu shluků

V rané fázi studie byl použit algoritmus TwoStep s automatickou detekcí optimálního počtu shluků, jehož výsledky byly dále použity k dalšímu zpracování. Nicméně po důkladném auditu dat bylo zjištěno, že program nevhodně vyhodnotil chybějící hodnoty u kategoriálních atributů a nahradil je jakousi výchozí hodnotou.

Po opravení tohoto nedostatku a správném klasifikování chybějících hodnot, jako chybějící, nikoli doplněné, se v souladu se znalostmi zmíněnými v teoretické části projevila neschopnost algoritmu TwoStep poradit si s případy, které mají neúplné hodnoty. Algoritmus totiž takové záznamy vyřazuje z tvorby modelu, a tak program zahlásil chybu nedostatku validních dat.

Použití Kohonenových map

Důsledkem neúspěchu s algoritmem TwoStep bylo nutno použít druhý způsob, jak algoritmicky zjistit ideální počet shluků – Kohonenovými mapami.

Tento způsob nám vyhodnotil počet 11 jako ideální. Tohoto algoritmu bylo nakonec využito i pro samotné hledání shluků v případech.

Kombinace více modelů

Jelikož všechny použité algoritmy jsou závislé na pořadí případů, bylo nutné ověřit stabilitu výsledků při různém pořadí případů. Výsledkem tohoto pokusu byly v určitých případech rozdílné hodnoty, což vedlo k tendenci přetížit celou fázi použitím více modelů a následnému vyhodnocení výsledků.

Už samotný počet shluků vykázal nestabilní hodnoty při různém pořadí případů. V jednom případě se totiž objevil jako ideální počet číslo 12. Pro modelování tak bylo vytvořeno šest různých modelů. Tři vedly do algoritmu K-Means s tím, že pokaždé bylo různé pořadí vstupních objektů. To samé platilo i pro zbylé tři větve, které vedly do Kohonenových map. Tak bylo získáno šesti různých přiřazení stejných objektů do shluků, a další část se zaobírala vyhodnocením spolehlivých výsledků.

Vyhodnocování probíhalo sloučením všech výsledků a nalezení těch záznamů, které tvořily shluk, nehledě na použitý algoritmus a pořadí objektů. Jen o nich se totiž dá uvažovat jako o kandidátech pro analytickou činnost bezpečnostních složek.

V rámci ověření výsledků, byly vyhodnocené shluky filtrovány a jejich konkrétní hodnoty dále zkoumány.

4.1.4 Vytvoření modelu zabývajícího se kapesními krádežemi.

Jak již bylo zmíněno, zde nás zajímají lokace a období, kdy ke krádežím dochází.

Proto bylo potřeba se nejdříve na krádeže podívat z geografického hlediska, aby bylo možné zjistit, zda se vyskytují rovnoměrně, či je to lokální problém. Bylo zjištěno, že kapesní krádeže se vyskytují prakticky výhradně na dvou místech. Pro lepší orientaci a srozumitelnost výsledků je těmto lokacím přiřazen název, a to na základě dodatečných informací, obdrženým k datům

Po pojmenování kritických lokací jsme se mohli zabývat časovou doménou událostí. Použitím histogramů, s parametrem zabývajícím se atributem obsahujícím informaci o týdnu události, jsme byli schopni patřičně oddělit jednotlivé časové úseky a nadále zkoumat vliv těchto časových úseků na lokaci.

Výsledky, které nám byly představeny programem, už zbývalo pouze zobrazit do dostatečně vypovídající a srozumitelné formy výstupu.

V této, poslední, fázi naší případové studie tedy nebyl použit jediný analytický algoritmus a všechny výsledky vznikly pouze díky vytvoření nových atributů a jejich vhodném zobrazení do vizualizačních nástrojů.

4.2 Implementace algoritmu

Algoritmus K-Means jsme se rozhodli implementovat na platformě .Net s programovacím jazykem C#. Je tedy pochopitelné, že k implementaci byl využit objektový přístup. Dále lze program dělit do dvou částí. Části obsahující metody a objekty umožňující konverzi a provedení samotného algoritmu a dále uživatelské rozhraní, které má na starosti načítání dat, interakce s uživatelem a zobrazování výsledných shluků. Jednalo se o klasickou Windows Forms aplikaci.

Celý vývoj byl rozdělen do dvou verzí. První verze byla zjednodušená oproti finální zamýšlené funkcionalitě a měla ověřit samotný algoritmus pro nejjednodušší kontinuální, data pro různé metriky. Druhá verze si pak měla poradit s kategoriálními/dichotomickými atributy, zobrazit přehled dat, a mít možnost tato data přetypovat. Pochopitelně u obou verzí byla možnost zvolit si počet výsledných shluků. Díky rozdělení do dvou verzí byla speciální pozornost věnována modularitě programu.

Již na začátku bylo jasné, že aplikace bude muset řešit:

1. načtení hodnot ze souboru a jejich konvertování do žádoucí podoby – do objektů
2. provedení samotného shlukování
3. nakonec zobrazení výsledků v přehledné formě – ideálně grafu.

4.2.1 Načítání hodnot a jejich konverze

V první verzi programu se počítalo pouze s numerickými atributy zadanými v exponenciálním tvaru v souboru s hodnotami oddělenými mezerou. Takto načtená data bylo možné rovnou použít jako hodnoty členů objektů.

Ve druhé verzi se pak zacházelo s klasickými CSV (comma separated values) soubory obsahujícími různé, předem neznámé typy atributů. Takto načtené atributy bylo potřeba nejdříve rozpoznat. Rozpoznávaly se tři typy atributů:

- dichotomické
- kategorické
- kontinuální

Uživateli však byla dána možnost tyto atributy přetypovat pro případ, že by to uživatel chtěl, či by byl v programu nedostatek.

Po rozpoznání typů atributů bylo potřeba vhodně přeměnit kategorické atributy na kontinuální, jelikož algoritmus K-Means ve svých implementacích používá Euklidovu metriku. Jelikož bylo cílem výsledky konfrontovat s výsledky IBM SPSS Modeleru, bylo nutné se inspirovat dokumentací k tomuto programu a použít stejnou konverzi kategorických atributů, jako je použita ve výše zmíněném programu.

4.2.2 Shlukovací algoritmus

Díky konverzi kategorických dat na data kontinuální bylo možné použít jeden a ten samý algoritmus pro obě verze programu. V tomto případě byly použity již hotové kolekce platformy .Net., ve kterých byly objekty uchovány, a které reprezentovaly samotné shluky. Další záměr byl nechat uživateli možnost výběru mezi více metrikami.

Přestože tento program není určen pro široké spektrum uživatelů, chtěli jsme zabudovat ošetření vstupů od uživatele. Program byl tedy například ošetřen před nevalidním počtem shluků, jako je počet 1 a méně, či počet shluků větší, nežli počet dostupných dat.

Výstupem algoritmu bylo zvoleno pouze číselné označení shluku, do kterého daný objekt patří.

4.2.3 Zobrazení výsledků do grafu

Pro pohodlné porovnání výsledků našeho algoritmu a algoritmu z programu IBM SPSS Modeler bylo třeba vytvořit vizuální zobrazení dat do grafu, z něhož bude na první pohled zřejmé, zda výsledky korespondují s profesionálním řešením, či ne.

Opět bylo využito již hotových komponent platformy .Net, s vhodnou konfigurací. Navíc, vzhledem k absenci funkce *jitter*, tedy vychýlení překrývajících se objektů, bylo dalším požadavkem rozšířit zobrazení o možnost skrýt vybrané shluky.

4.3 Vytvoření části kurzu Datamining na e-learningovém portálu FM TUL

Zde byl postup již předem známý. Nejprve bylo potřeba vypracovat teoretickou část této bakalářské práce, jejíž obsah měl být samostatnou kapitolou ve zmiňovaném kurzu. Poté bylo potřeba vypracovat případovou studii a její popis přepsat do vhodné podoby pro výukové účely. A tak, po seznámení se s administrací portálu, mohl být tento materiál vytvořen, nahrán na server a jako poslední krok k němu mohly být vytvořeny kontrolní otázky.

5 Realizace řešení

5.1 Zvolený software

Případová studie

K realizaci případové studie byl použit software IBM SPSS Modeler ve verzi 14.2. Tento program je komplexní nástroj pro řešení dataminingových úloh. Vyznačuje se grafickým zpracováním prostředí, kde se vytváří modely a klade důraz na to, aby uživatel nepotřeboval rozsáhlé znalosti z oblasti programování a algoritmů.

Projekty jsou zde nazývány *Streamy*, česky *Proudy.* Obsahem proudů jsou uzly, které se řadí do různých kategorií, v závislosti na tom, jakou činnost vykonávají. Existují tedy například uzly sloužící k načítání dat, uzly provádějící operace se záznamy, uzly provádějící operace s atributy, uzly modelovací, uzly sloužící k různým grafickým zobrazením dat, a tak dále. Tyto uzly jsou propojovány uživatelem, a tak se výstup z jednoho uzlu stává vstupem druhého, odkud vychází ono označení *proud*.

Implementace algoritmu

Pro tvorbu programu, implementujícího algoritmus K-Means, byl zvolen software Microsoft Visual Studio 2010.

5.2 Případová studie

V této sekci budou popsány kroky, provedené v jednotlivých fázích, popsaných v Návrhu řešení za pomoci zmíněného software. Kompletní ilustrace výsledných proudů lze nalézt v příloze. Na nosiči DVD-ROM, přiloženém k této práci jako zvláštní příloha, jsou umístěny proudy této studie.

5.2.1 Proud přípravy dat

Ilustrace proudu této fáze je obsahem Přílohy 1.

Vstupním bodem je uzel s názvem „Načtení dat“, který načítá data ze statistického souboru. V tomto uzlu byl proveden první náhled do dat, za účelem zjištění, jaké atributy se v nich nachází a v jaké formě.

Mezi daty tak byly nalezeny atributy jako datum, číslo případu, lokalita, kde k případu došlo, různé příznakové i kategoriální atributy, vyjadřující způsob provedení vloupání, jednalo-li se o případ vloupání do objektu a také kód případu, číselně reprezentující, o jaký trestný čin/přestupek se jedná. Z data bylo zjištěno, že se jedná o záznamy za jeden rok na určité lokalitě. Kompletní seznam atributů je obsahem Přílohy 2.

Ze získaných dat bylo patrné, že některá nebyla správně rozpoznána, respektive, že automaticky přiřazené typy jsou chybné. Všechny dichotomické atributy byly vedeny jako nominální. Příloha 2 již obsahuje data opravená, pro představu v jakém stavu se nacházela bez provedených úprav, přiložena ilustrace 7.

Field	Measurement	Values
cislo_pripadu	Continuous	[1.09701001E8,1.09701662E8]
Kod	Nominal	"008/01","008/06","028/03","030/0...
souradnice_X	Continuous	[210.0,500.0]
souradnice_Y	Continuous	[-900.0,-510.0]
Datum	Continuous	[1995-01-01 04:10:00,1995-12-3...
MO_vstup	Nominal	Lstí,Vloupání,Vstup
MO_misto	Nominal	Dveře,Okno,Pož_sch
MO_zabezp	Nominal	Alarm,Odemčeno,Otevřeno,Zamč...
MO_cinnost	Nominal	Charita,N/A,Obchod,Prodejna,Sta...
MO_odchod	Nominal	A,N
MO_zabdvere	Nominal	A,N
MO_neporadek	Nominal	A,N
MO_sejf	Nominal	A,N
MO_zpusobvstupu	Nominal	N/A,"Rozbité dveře","Rozbité okno...
Kaud	Nominal	A,N
Kvid	Nominal	A,N
Kpocitac	Nominal	A,N
Pobleceni	Nominal	A,N
Ppenize	Nominal	A,N
Pkredit	Nominal	A,N
Plek	Nominal	A,N
Ptelefon	Nominal	A,N
Phodiny	Nominal	A,N
Pkalk	Nominal	A,N
Palkoh	Nominal	A,N
Pzaznam	Nominal	A,N
Psperky	Nominal	A,N
Ppenez	Nominal	A,N
Pdvere	Nominal	A,N
Pokno	Nominal	A,N
Pautomat	Nominal	A,N
Ptelauto	Nominal	A,N

Ilustrace 7: Chybně rozpoznané typy atributů

Po odstranění nedostatku s nevhodným rozpoznáním typů bylo potřeba ručně nadefinovat, která ze dvou hodnot „A“ a „N“ reprezentuje hodnotu *true* a která reprezentuje *false.* Za účelem vyhnutí se nesrovnalostem v navazujících modelech, byly dva následující uzly modifikovány pravdivou, resp. nepravdivou hodnotu na poněkud tradičnější „1“, resp. „0“.

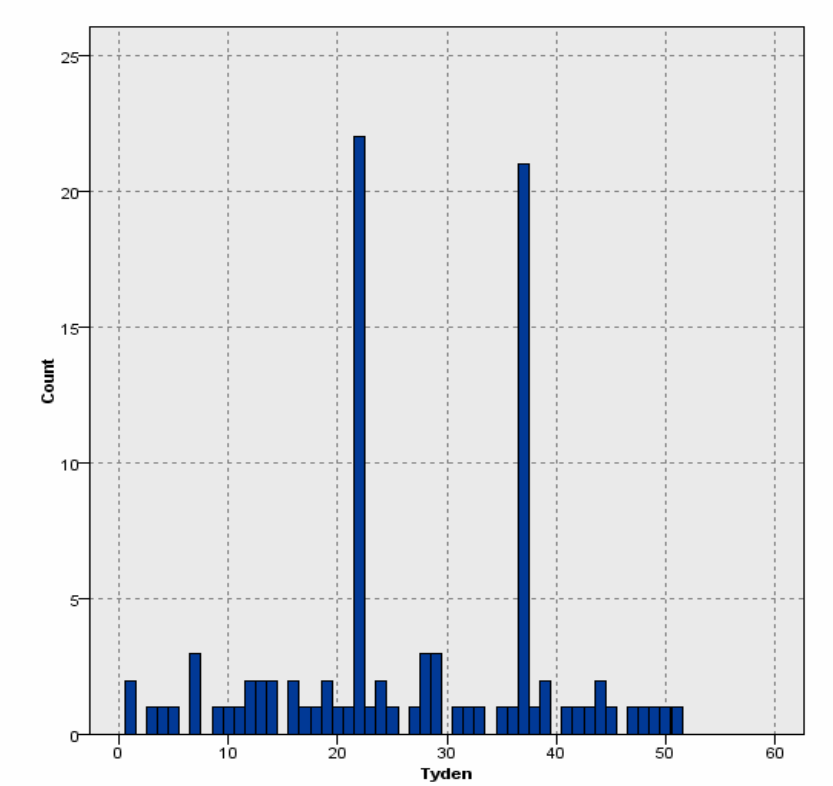
V momentě, kdy byla data upravena do správné formy, bylo třeba doplnit data o odvozené atributy. Nejprve byl atributu „Kod“, doposud nabývajícího kódové číselné hodnoty, přiřazen popis, který usnadnil čitelnost dat. Tyto popisy byly získány z překladového souboru, kde každé kódové reprezentaci byl přiřazen text, ve stylu „Vandalismus“. Vzhledem k tomu, že se v datech objevovaly druhy trestné činnosti, které nebyly předmětem analýz, byly proto takové činnosti sloučeny tak, aby ve výsledku zůstaly kategorie čtyři:

- násilná činnost
- větší majetková trestná činnost (vloupání do budov)
- vandalismus (obsahující též drobnější majetkovou trestnou činnost)
- jiné (zpravidla činy, které se vyskytují náhodně a nemá smysl v nich hledat vyšší souvislosti)

Z data se pak odvodily atributy „hodina“ a „týden“.

Dalším bodem v tomto proudu byl výběr vhodných záznamů do výstupních souborů, které byly použity jako vstup navazujících modelů. Do části, zabývající se nalezením optimálního pokrytí oblasti daným počtem hlídek, se vybraly ty záznamy, které splňovaly příslušnost ke kategoriím násilné a majetkové trestné činnosti, plus kategorii vandalismu. Do části, zabývající se nalezením souvisejících případů vloupání, se vybraly pouze ty záznamy, které příslušely kategorii větší majetkové trestné činnosti. Výstup do poslední části s kapesními krádežemi se skládal pochopitelně pouze ze záznamů, obsahujících kód kapesních krádeží. Ten však musel být ještě zpracován.

Toto zpracování spočívalo v dalším náhledu do dat, tentokrát z perspektivy jednotlivých týdnů a četnosti v těchto týdnech. Tento náhled je zobrazen ilustrací 8.



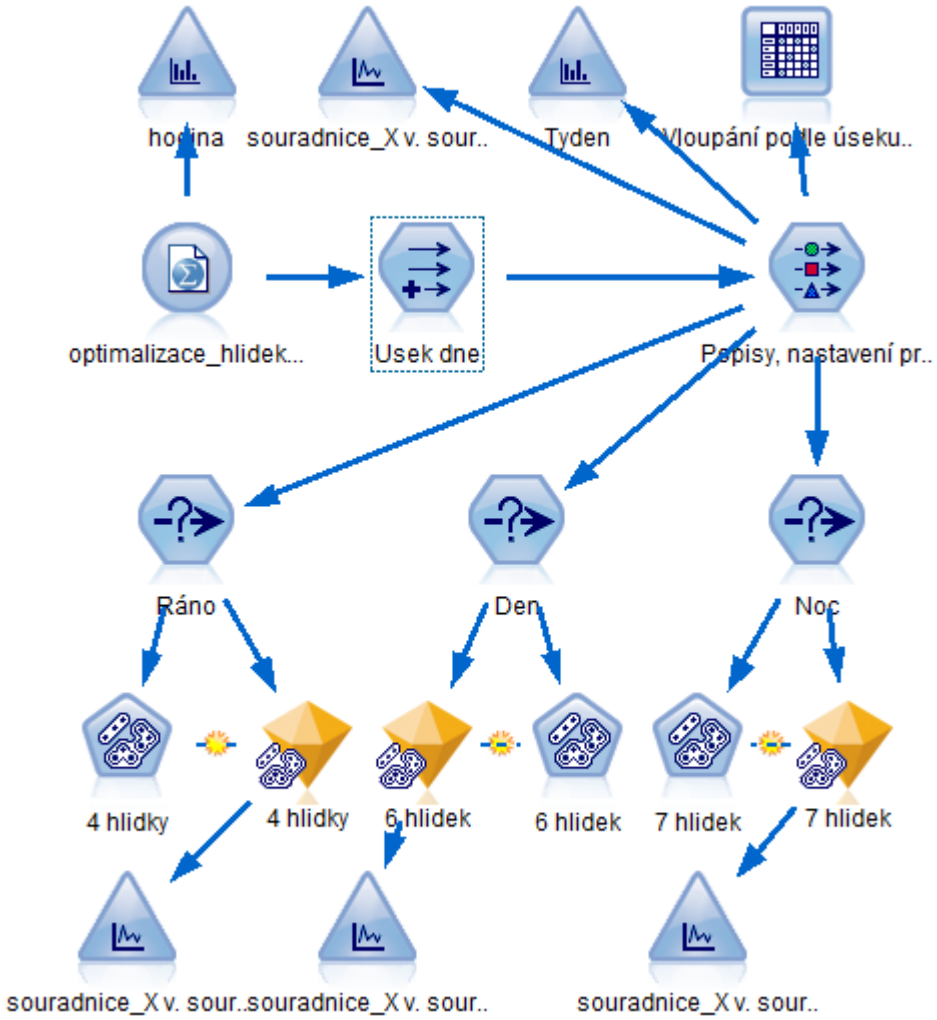
Ilustrace 8: Četnost kapesních krádeží v jednotlivé týdny roku

Bylo zjištěno, že se během roku vyskytují dva týdny vykazující extrémní nárůst četnosti. Kvůli těmto týdnům byl vytvořen nový atribut, který bude indikovat, zda se jedná o případ udávající se v prvním extrémním týdnu, druhém extrémním týdnu, či zda se jedná o případ udávající se v „běžný“ týden.

Posledním krokem před exportováním dat do souboru bylo filtrování nepotřebných atributů.

5.2.2 Proud nalezení optimálního rozložení hlídek

Proud této fáze je obsahem ilustrace 9.



Ilustrace 9: Proud hledání optimálního pokrytí oblasti hlídkami

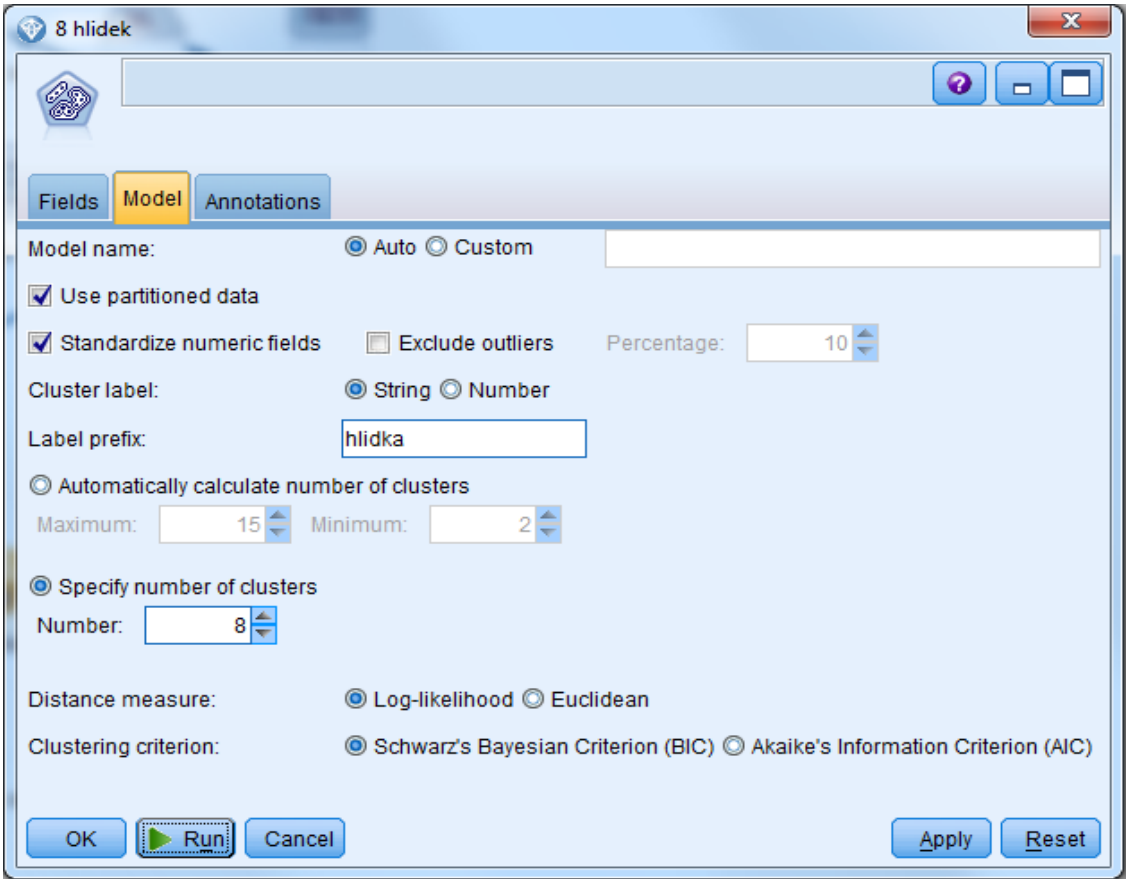
Vstupním bodem je opět načtení případů ze statistického souboru, vyexportovaného z přípravy dat. Tato načtená data jsou zobrazena v histogramu, aby bylo možné určit vhodné hodiny pro definici jednotlivých částí dne. Na základě četnosti případů v jednotlivé hodiny, bylo zvoleno následující rozdělení:

- Ráno: 6.-10. hodina
- Den: 10.-19. hodina
- Noc: 19.-6. hodina

Jelikož jsme nepoužívali automatické definování počtů shluků (policejních skupin) a v další části tohoto proudu bylo nutné zvolit požadovaný počet shluků, byla zobrazena četnost případů v jednotlivé části dne. Tím došlo ke zjištění, že počet případů ráno je poloviční oproti počtu případů ve dne, a ten je poloviční oproti počtu případů v noci. Byly tedy zvoleny následující počty skupin:

- Ráno: 3 skupiny
- Den: 5 skupin
- Noc: 6 skupin

Zbývalo tedy roztřídit záznamy do tří skupin, dle části dne, ve kterých se udály, a tyto skupiny tvořily vstup dohromady tří uzlů s algoritmem TwoStep. Konfigurace modelujícího uzlu s nastavením pro skupinu „Noc“ je zobrazena na ilustraci 10, výsledné rozložení skupin je součástí Přílohy 3.



Ilustrace 10: Nastavení uzlu TwoStep pro optimalizaci rozmístění hlídek v noci

5.2.3 Proud nalezení souvisejících případů

Proud této fáze je obsahem Přílohy 4.

Přestože byly v proudu přípravy dat vyexportovány případy vloupání do obytných, a zároveň do neobytných budov, zde byly vybrány pouze ty obytné. Po nastavení rolí příslušných atributů bylo možné data postoupit k modelování.

Jak lze vidět na ilustraci zobrazující proud, z uzlu pojmenovaného „Nastavení míry a rolí“ vychází šest spojnic do dalších uzlů, kde čtyři z nich jsou uzly třídící. Dva pro každou metodu shlukování. Tyto uzly třídí vstupní množinu dat, a to jednou sestupně a podruhé vzestupně, což nám mělo pomoci v odhalení slabých shluků. Slabý shluk je naše označení pro shluk, který neexistuje ve všech třech modelech dané shlukovací metody.

Výstupy všech šesti modelů putují do uzlu „Merge“, který měl za úkol sjednotit výsledky (zařazení) každého případu napříč spektrem modelů. Cílem bylo tedy mít identifikační číslo záznamu a k němu šest hodnot, určujících náležitost k určitému shluku v daném modelu. Nyní bylo potřeba nalézt skupiny případů, které spolu tvořily shluk ve všech šesti modelech. Takové případy byly označeny za před-kandidáty na analýzu vyšetřovatelem.

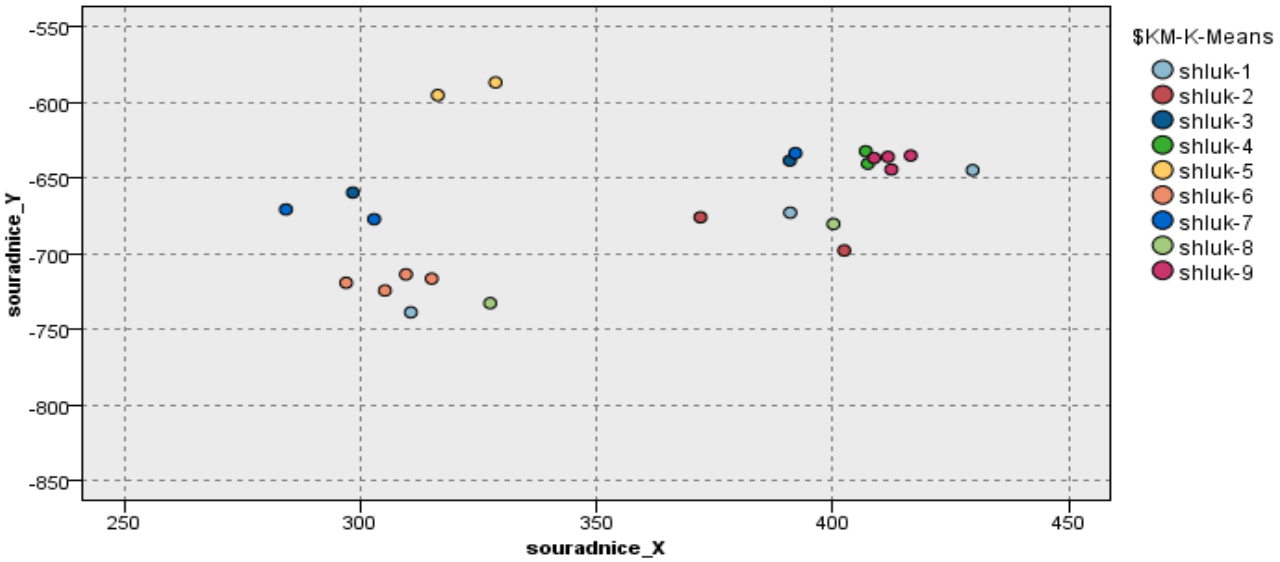
Jelikož data byla po sjednocení nepřehledná, bylo potřeba použít agregační uzel, agregující přes atributy nesoucí označení shluku. Z toho důvodu byla vytvořena tabulka, která ve svých sloupcích měla výsledky z jednotlivých modelů a v řádcích bylo možně vidět, z jakých dílčích výsledků se skládají skupiny před-kandidátů. Dostali jsme 84 skupin před-kandidátů, z nichž však pouhých 9 tvořilo skupiny o dvou a více případech. Prvních 20 skupin, tvořících výstup agregačního uzlu, zobrazuje ilustrace 11.

	$KM-K-Means1	$KM-K-Means2	$KM-K-Means3	$KXY-Kohonen1	$KXY-Kohonen2	$KXY-Kohonen3	Record_Count
1	skupina-8	skupina-2	skupina-7	X=0, Y=2	X=3, Y=0	X=3, Y=0	4
2	skupina-9	skupina-8	skupina-8	X=3, Y=0	X=0, Y=0	X=0, Y=0	4
3	skupina-5	skupina-3	skupina-10	X=0, Y=0	X=0, Y=2	X=0, Y=2	3
4	skupina-4	skupina-3	skupina-9	X=3, Y=0	X=0, Y=0	X=0, Y=0	3
5	skupina-6	skupina-2	skupina-1	X=0, Y=2	X=3, Y=0	X=3, Y=0	2
6	skupina-8	skupina-6	skupina-3	X=3, Y=2	X=3, Y=2	X=3, Y=2	2
7	skupina-1	skupina-9	skupina-11	X=3, Y=2	X=3, Y=2	X=3, Y=2	2
8	skupina-11	skupina-3	skupina-1	X=3, Y=0	X=1, Y=0	X=1, Y=0	2
9	skupina-6	skupina-7	skupina-10	X=0, Y=1	X=0, Y=2	X=0, Y=2	2
10	skupina-7	skupina-4	skupina-2	X=0, Y=0	X=0, Y=2	X=0, Y=2	1
11	skupina-6	skupina-7	skupina-1	X=2, Y=0	X=0, Y=0	X=0, Y=0	1
12	skupina-10	skupina-1	skupina-6	X=3, Y=2	X=2, Y=2	X=2, Y=2	1
13	skupina-4	skupina-7	skupina-1	X=0, Y=2	X=2, Y=0	X=3, Y=0	1
14	skupina-3	skupina-9	skupina-4	X=2, Y=2	X=3, Y=1	X=3, Y=1	1
15	skupina-5	skupina-2	skupina-5	X=0, Y=0	X=0, Y=2	X=0, Y=2	1
16	skupina-4	skupina-4	skupina-4	X=3, Y=2	X=3, Y=2	X=3, Y=2	1
17	skupina-4	skupina-1	skupina-4	X=3, Y=2	X=3, Y=2	X=3, Y=2	1
18	skupina-11	skupina-7	skupina-1	X=0, Y=2	X=3, Y=0	X=2, Y=0	1
19	skupina-11	skupina-7	skupina-1	X=0, Y=2	X=2, Y=0	X=2, Y=0	1
20	skupina-7	skupina-5	skupina-2	X=1, Y=0	X=1, Y=2	X=1, Y=1	1

Ilustrace 11: Výstup agregačního uzlu zobrazující skupiny před-kandidátů

Nakonec bylo vybráno následujících 9 skupin pomocí uzlu s názvem „Početnější skupiny“ a dalším cílem bylo tyto skupiny zobrazit na mapě. K tomu byl, poněkud netradičně, zvolen uzel K-Means, který se jevil jako nejjednodušší a nejrychlejší řešení. Při nastavení cílového počtu shluků na 9 a nastavení vstupních rolí atributům, nesoucích označení výsledných shluků z předchozích modelů, bylo získáno jednotné označení pro celou šestici. Oproti ilustraci 11 by tak nebylo potřeba šest sloupců pro identifikaci unikátní kombinace, ale pouze jeden sloupec s označením nově vzniklého shluku.

Výsledné shluky před-kandidátů pro prozkoumání policejním specialistou jsou zobrazeny v grafu tak, aby bylo možné vidět pozici těchto případů na mapě. Toto zobrazení je obsahem ilustrace 12.



Ilustrace 12: Zobrazení před-kandidátů na mapě

Na výše zmíněné ilustraci je možné vidět, že některé shluky jsou poblíž sebe, aniž by lokalita vstupovala do modelování v jakékoli části tohoto proudu. Zdá se tedy, že tito před-kandidáti jsou adepty na to stát se kandidáty. Zbývalo si jednoho z nich vybrat a ověřit tak funkčnost modelů prostupujících tímto proudem.

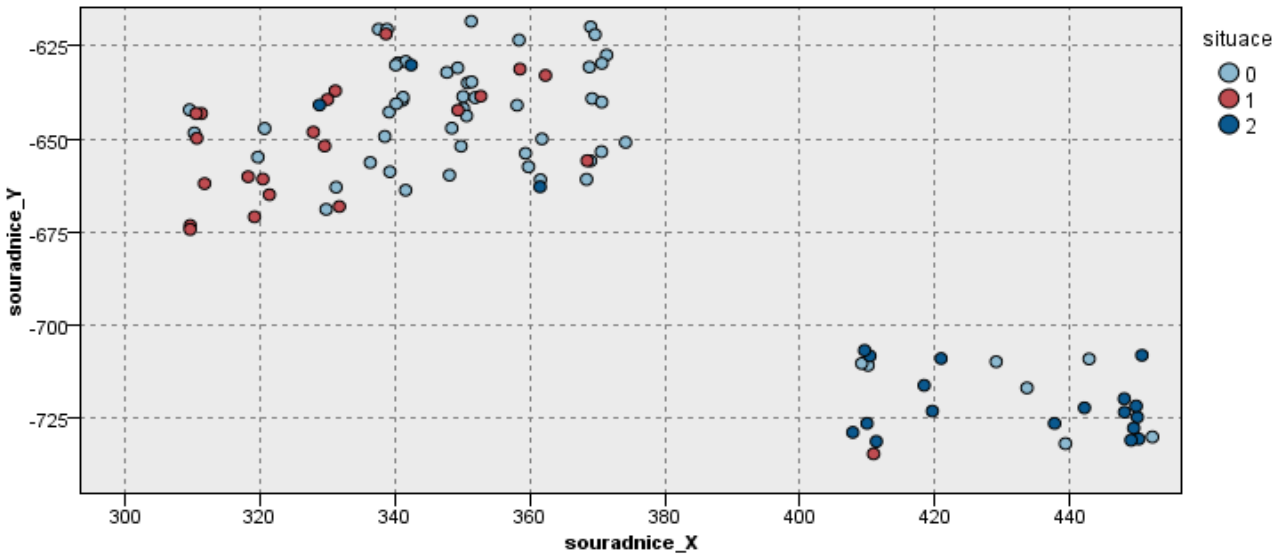
Vybrána byla čtveřice případů umístěná v levé části, nichž byla zkoumána míra odlišnosti v modus operandi a v dalších charakteristických znacích provedení loupeže. Bylo zjištěno, že všechny čtyři případy se staly v jeden den, v rozmezí čtyř hodin. Připomeneme, že datum, čas a ani lokace do modelování nevstupovaly. Dále se všechny případy shodovaly ve všech atributech obsahujících modus operandi. Ve většině zbývajících atributů, které vyjadřovaly například druh odcizené věci, docházelo rovněž ke shodě.

Závěrem lze tedy říct, že model, skládající se z několika dílčích modelů, je funkční a použitelný na daných datech v této úloze.

5.2.4 Proud zabývající se kapesními krádežemi

Proud této fáze případové studie je obsahem Přílohy 5.

Již v proudu přípravy dat byl proveden prvotní přehled o časovém rozpoložení kapesních krádeží. Zjištěno bylo, že v roce, ke kterému máme data, jsou dva týdny vykazující extrémní nárůst. Vzhledem k tomu, že výstupem má být informace nejen o časové doméně, ale také o rozmístění kapesních krádeží, bylo nahlédnuto na případy z hlediska rozmístění na mapě. Ilustrace 13 toto zobrazení demonstruje.



Ilustrace 13: Rozmístění kapesních krádeží na mapě

Na poslední ilustraci bylo zcela zřejmě vidět, že kapesní krádeže se vyskytují ve dvou lokacích. Byl použit soubor obsahující pojmenování lokací společně s jejich souřadnicemi. Bylo tak možné dát oblastem jméno a zároveň se ukázalo, že se ve skutečnosti jedná o lokace tři, z čehož dvě jsou vedle sebe:

- Náměstí
- Stadion
- Park

Tak byl odvozen nový atribut „oblast“, nesoucí název místa krádeže. Po odvození nového atributu nás zajímalo zastoupení krádeží v jednotlivých oblastech v závislosti na tom, o jaký týden se jedná. Závěrem je, že se ve většině týdnů krade zásadně na náměstí. Další lokace se objevovaly pouze v týdnech kolem letní sezóny.

Po dalším zkoumání, které spočívalo například v kategorizaci týdnů (sezónní a mimo-sezónní) či v zobrazování závislostí na histogramech, bylo zjištěno, že na náměstí se tyto případy loupeží vyskytují v průběhu celého roku, zatímco v parku a na stadionu pouze v letní sezóně.

5.3 Implementace algoritmu

Celý projekt byl rozdělen do dvou částí. Do knihovny, implementující algoritmus K-Means a obsahující všechny pomocné datové struktury, a do Windows Forms aplikace, která tvořila uživatelské rozhraní potřebné pro načítání dat, jejich manipulaci a zobrazení výsledků.

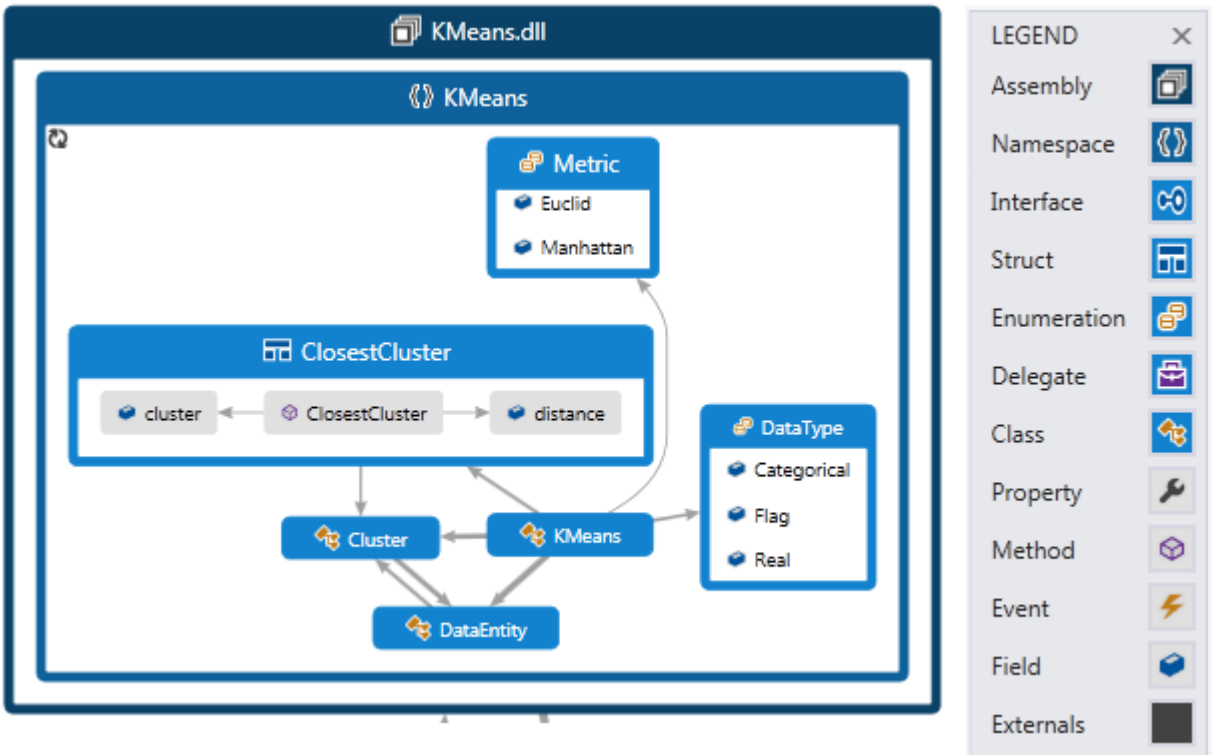
Na nosiči DVD-ROM, přiloženém k této práci jako zvláštní příloha, je umístěn projekt (solution), obsahující veškeré zdrojové kódy.

5.3.1 Knihovna implementující K-Means algoritmus

Vzhledem k tomu, že algoritmus byl popsán již v teoretické části, bude se tato část věnovat vlastní implementaci z hlediska navržení objektů, práce s nimi a pozornost bude věnována rozšířením, která bylo třeba udělat tak, aby se výsledky vlastní implementace daly porovnat s výsledky programu IBM SPSS Modeler.

Diagram závislostí

Diagram na ilustraci 14 zobrazuje diagram závislostí mezi třídami a datovými strukturami uvnitř knihovny obsahující algoritmus K-Means. Zobrazen je detail pouze těch členů, jejichž vnitřní vztahy jsou jednoduché.



Ilustrace 14: Diagram závislostí

Těmi členy jsou:

- výčet *Metric* obsahující podporované metriky
- výčet *DataType* obsahující podporované typy atributů
- struktura *ClosestCluster* používaná jako nadstavba třídy *Cluster* (reprezentující shluk), obsahující kromě reference na objekt třídy *Cluster* navíc vzdálenost objektu od shluku, čehož se používá při vyhledávání nejbližšího shluku.

Na diagramu se dále vyskytují tři třídy, jejichž obsah datových členů a metod není zobrazen, jelikož by nebyl dostatečně srozumitelný. Místo toho jejich funkcionalitu popíšeme slovně. Jedná se o:

- třída *DataEntity*: Tato třída reprezentuje to, čemu se v průběhu této práce říkalo „objekt“, „záznam“ či „případ“ (z případové studie). Obsahuje seznam hodnot atributů, referenci ke shluku k němuž náleží, a pro usnadnění práce také přetížené operátory sčítání a dělení, jenž se používají například u vypočítávání vzdáleností či při standardizaci atributů.

- třída *Cluster*: Jak již bylo řečeno, tato třída reprezentuje shluk. Obsahuje seznam objektů *DataEntity*, které do něj náleží. Uchovává také informaci o centroidu tak, aby nemusel být pokaždé vypočítáván.

- Třída *K-Means*: V podstatě se jedná o hlavní třídu. V této třídě je naimplementován samotný algoritmus, jsou zde všechny ostatní metody sloužící tomuto algoritmu, a zároveň je zde rozhraní mezi uživatelským rozhraním a knihovnou.

Rozšíření oproti obecné definice algoritmu K-Means

K této sekci se váže Příloha 6, obsahující zdrojový kód k hlavní metodě naší implementace. Všechny metody nebudou rozepsány z důvodu rozsahu práce, avšak funkcionalita důležitých metod bude popsána.

Standardizace dat

Před začátkem algoritmu bylo pro dosažení co nejlepších výsledků potřeba data standardizovat. Toho se docílilo tak, že od každého atributu byl odečten průměr napříč všemi hodnotami tohoto atributu a tento rozdíl byl vydělen rozsahem hodnot. Výsledkem tohoto opatření byly hodnoty v rozsahu <0 ; 1> nehledě na jednotky měření.

Zpracování kategorických dat

Dalším, největším rozšířením oproti obecné definici algoritmu K-Means, byla potřeba umět zpracovat kategorické atributy, se kterými v běžném pojetí algoritmus nepočítá. Čili byla potřeba jakási předpříprava dat, kterou obsahovala metoda s názvem „transformData“.

Tato metoda fungovala na tomto principu:

1. každý kategorický atribut o n hodnotách, nahraď n dichotomickými atributy
2. Z nově vytvořených atributů bude maximálně jeden nabývat hodnoty 1, a to ten, který odpovídá původní hodnotě.

Budeme-li mít například atribut *Tvar*, který bude mít hodnoty {“čtverec”, ”obdélník”}, pak se tento atribut transformuje na dva dichotomické atributy s názvy například *Je_Obdélník* a *Je_Čtverec*. Pro objekt, jenž měl v původním atributu *Tvar* hodnotu „obdélník“, bude mít nový atribut *Je_Čtverec* hodnotu „0“ a atribut *Je_Obdélník* hodnotu „1“.

Nicméně udržovat hodnoty 1 a 0 bylo nevhodné, a to z toho důvodu, že K-Means používá eukleidovskou metriku, a ta kategoriálním atributům přisoudila větší váhu. Máme-li totiž jeden atribut, pohybující se v rozmezí hodnot 0 až 1, a dva objekty s naprosto rozdílnou hodnotou tohoto atributu, tedy první objekt bude mít hodnotu 1 a druhý bude mít hodnotu 0, pak výsledek eukleidovské metriky bude 1 a vyšší být nemůže. Pokud ale tento atribut pojmeme jako kategoriální a oba objekty budou mít různou hodnotu, pak po transformaci na dichotomické atributy dostaneme hodnotu 1 + 1 (rozdíl nastane v obou vytvořených atributech).

To bylo odstraněno takzvanou „Set Encoding Value“, neboli hodnotou kategoriálního kódování, která v transformačním kroku přiřazovala místo hodnoty 1 hodnotu jinou, v našem případě odmocninu z jedné poloviny. Díky tomu, při rozdílných kategoriích, hodnoty čtverců rozdílů byly ½ + ½, čili součet 1, stejně jako by tomu bylo u kontinuálních atributů.

Zastavovací podmínky cyklu

Podmínky zastavující iterační cyklus byly v tomto případě následující:

- počet iterací dosáhl maximálního počtu, poskytnutého v parametru metody. Pokud tento parametr nebyl nastaven, použila se maximální hodnota znaménkového celočíselného šestnácti-bitového datového typu.
- V cyklu nebylo provedeno žádné přeřazení jakéhokoli objektu do jiného shluku.

Ošetření vstupů

Jelikož algoritmus pracuje s hodnotami, které mu poskytuje uživatel skrze uživatelské rozhraní, bylo nutné zajistit dostatečnou ochranu proti nevalidním vstupům. Součástí hlavní metody tedy byly následující kontroly:

- Minimální počet zadaných shluků může být dva
- Maximální počet shluků musí být roven nebo menší, než celkový počet objektů v datech
- Uživatelsky zadaný maximální počet iterací musí být vyšší, než jeden.

5.3.2 Uživatelské rozhraní

Uživatelské rozhraní mělo dvě hlavní úlohy:

1. Umožnit uživateli definovat vstupní parametry do algoritmu.
2. Zobrazit výsledky v přehledné, grafické formě

Uživatelská definice vstupních parametrů

Design této části je možné vidět v Příloze 7

Filosofie této části je taková, že uživatel svými kroky postupuje v levém *pracovním sloupci* od shora dolů:

1. Nejprve si tedy přes klasický dialog vybere patřičný soubor s daty. Aplikace je navržena tak, aby pracovala se soubory, které mají data oddělena čárkou, neboli CSV. V datech rovněž musí být první řádek, popisující názvy jednotlivých atributů.

2. Vybere si metriku, za použití takzvaných *radio button* selektorů. K dispozici je metrika eukleidovská a metrika městských bloků, neboli Manhattanská či taxikářova.

3. Oblast pod výběrem metriky obsahuje tři selektory, které slouží k definování, jaké položky nesou identifikátor záznamů a jaké položky mají být použity jako osy výsledného grafu. Tyto tři položky posléze nejsou součástí modelování.

4. Další sekce slouží ke změně datových typů atributů, pokud automatická detekce nebyla dostačující. V odstavcích výše bylo zmíněno, že naše konkrétní implementace zachází s každým typem dat jinak. K tomu, aby uživatel tento typ změnil, stačí označit v tabulce dat, umístěné napravo od *pracovního sloupce*, jakoukoli hodnotu daného atributu, a poté zvolit, o jaký datový typ se jedná. Předvyplněný typ je ten, který byl automaticky rozpoznán.

5. Nakonec uživatel použije tlačítko *Run* a algoritmus provede shlukování.

Zobrazení výsledků

Po provedení algoritmu se uživateli otevře nové okno, kde se nachází graf zobrazující jednotlivá data a jejich příslušnost ke *clusterům* neboli shlukům. Jelikož se v datech této práce některé objekty překrývaly, bylo toto zobrazení dále zlepšeno o možnost skrýt libovolný počet shluků. Díky tomu je možné přehledně pozorovat rozprostření objektů daného shluku..

6 Vyhodnocení řešení

6.1 Případová studie

Případová studie pokrývala několik menších, izolovaných pod-úkolů, a tak i zhodnocení výsledků bude vypadat dle struktury zadání.

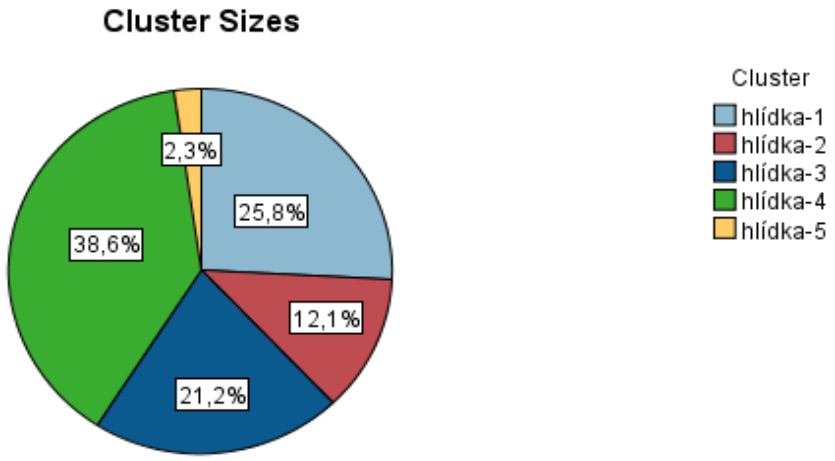
Optimální rozmístění hlídek

Na dané množině dat skutečně došlo k rovnoměrnému pokrytí mapy, tvořené zaznamenanými kriminálními případy, určitým počtem hlídek (viz příloha 3).

Během celé této úlohy bylo třeba definovat řadu parametrů, které by v reálném prostředí vzešly od zadavatele. Nejlepším příkladem takového parametru je počet hlídek, patrolujících pro danou dobu v oblasti.

Tento parametr byl zvolen dle vlastního uvážení, bez znalosti rozlohy oblasti, za předpokladu nižšího počtu hlídek, který bude mít přímou úměrnost s počtem kriminálních případů v tu kterou část dne. Zcela subjektivní také bylo rozhodnutí týkající se sloučení některých kategorií zločinů, a to bez zpětné vazby od lidí v dané problematice se pohybujících. Otázkou tedy je, na kolik byla úloha oproštěna od omezení, jenž se v praxi objevují.

Kromě již zmíněného zvolení počtu hlídek se v této úloze objevila otázka, zda je pro zajištění bezpečnosti ve společnosti lepší rovnoměrné pokrytí plochy, či rovnoměrné pokrytí kriminálních případů. Na ilustraci 15 lze totiž zpozorovat ne zcela vyrovnané procentuální pokrytí případů pěti hlídkami během dne. Znamená to, že v tomto případě bylo raději pokryto větší území za cenu toho, že u této hlídky bude menší pravděpodobnost jejího využití v akci.



Ilustrace 15: Procentuální pokrytí případů pěti hlídkami během dne

Související případy vloupání do obytných budov

V této části je úspěšnost identifikace souvisejících případů silně závislá na kvalitě dat. Lze tvrdit, že veškeré kandidáty na prověření, které náš model našel, opravdu vykazují znaky podobnosti a některé z nich dokonce podpořily i informace o času a lokalitě.

To však nutně neznamená, že všechny nalezené případy lze automaticky prohlásit za související. Vzhledem k tomu, že se jedná o úlohu deskriptivní, lze pouze prohlásit, že nalezené případy nesou do určité míry podobné rysy.

Ona „míra“ vychází z nastavení „přísnosti“ modelu a po patřičné odezvě z praxe by mohla být tématem diskuze. Je zde balance na hranici toho, aby model nenacházel pouze případy, které vykazují tak silnou souvislost, že by si je kriminalisté spojili, aniž by jakýkoli model potřebovali a mezi těmi případy, kdy by podobnost byla vyhodnocována ve větší míře i u těch případů, které spolu nesouvisí.

Kapesní krádeže

Pro poslední úlohu nebylo použito žádných modelovacích algoritmů a celá se skládala pouze z práce analytika, jeho nahlížení do map, histogramů, tabulek a jiných vizualizačních nástrojů. Nejednalo se o jakýsi univerzální přístup pro vyšetřování prevence kapesních krádeží.

Dle zadání byla prozkoumána doména kapesních krádeží z hlediska času a místa a bylo dosaženo kompletních závěrů o tom, kde a kdy k nim dochází. Použité postupy byly jednorázové. V případě takového zadání úlohy, aby se znalosti daly použít v delším časovém horizontu, bylo by nutné upozornit na dva extrémní týdny, které dle našeho názoru musely být způsobeny jistou kulturní událostí, a je tak potřeba zjistit charakter této události, zejména, zda se bude opakovat v následujících letech, a případně posílit prevenci kapesních krádeží v dnech konání zmiňované akce.

6.2 Vlastní implementace algoritmu K-Means

Program je schopný úspěšně pracovat s daty ve stejném formátu, s jakými s nimi pracuje program IBM SPSS Modeler, což bylo jedním z požadavků na aplikaci. Program byl vytvořen modulárně, čili bylo odděleno uživatelské rozhraní od vlastní knihovny obsahující algoritmus. Díky zavedení různých datových typů pro vstupní data, lze použít algoritmus v tradičním smyslu - pouze s kontinuálními hodnotami, stejně dobře jako ve smyslu úloh v dataminingu – i s kategoriálními hodnotami. Navíc po adaptaci uživatelského rozhraní není problém vytvořit robustní platformu, schopnou načítat několik typů souborů, exportovat výsledky do souboru, či mít rozšířené možnosti manipulace s daty. To vše bez zásahu do importované knihovny s algoritmem.

Výstupy z této vlastní implementace a výstupy z implementace ve zmiňovaném profesionálním řešení je možné porovnat přílohu 8 a přílohu 9. Je zřetelné, že výsledky jsou si velice podobné.

6.3 Porovnání postupů v případové studii s postupy v reálném prostředí

V rámci této bakalářské práce byl navštíven Odbor analytiky krajského ředitelství policie pro Liberecký kraj, kde byly získány informace o současném stavu a existujících postupech v záležitostech podobných těm, kterými se zabývá případová studie.

Informace z této schůzky se týkají především Libereckého kraje a metody zde používané nemusí platit celorepublikově.

6.3.1 Současná situace

Pro nenásilné trestné činy, jako jsou loupeže, krádeže aut a podobně se nepoužívá žádného speciálně zaměřeného software. Analytici používají takzvané žurnály, což jsou soubory formátu Microsoft Excel a každý takový žurnál se týká jedné kategorie trestných činů. Existuje tak speciální žurnál pro vykradená auta, ukradená auta, vloupaní a podobně. Tyto žurnály fungují v rámci kraje a jednotlivé okresy k nim přistupují přes Microsoft Sharepoint.

Data v žurnálech jsou z velké míry podobná datům, která byla pro tuto práci k dispozici. Co se rozsahu týče, je jich o trochu více. Dat však není mnoho (začala se sbírat relativně nedávno) a jejich dosavadní forma není bez editace příliš vhodná pro použití v modelech používaných programem IBM SPSS Modeler. Dle vyjádření analytického oddělení však již existují tendence pro vytváření strukturovaných dat, což poskytne prostor pro použití v expertních algoritmech. Nejdříve však bude potřeba adaptovat stávající data do nové, strukturované podoby.

6.3.2 Možnosti nasazení

Při diskuzi nad tématem možnosti zavedení systému podobného tomu, který byl použit v této práci bylo zjištěno, že je nutné vyřešit ještě řadu témat.

Naprosto stěžejní komplikací je současná personální situace spojená s obecným nedostatkem financí v sektoru. Pro zavedení datamining-ových přístupů, by totiž nebyla potřeba jen software-ová platforma, která svou cenou atakuje statisícové částky, ale také dostatečně školený personál. To znamená investovat nemalé částky a alokovat lidskou sílu do pozic, jež se netýkají nasazení v terénu. Bohužel, v současnosti jsou tyto přístupy prevence kriminality okrajovou záležitostí.

I za předpokladu, že výše zmíněné by přestalo býti problémem, je zde další překážka v podobě nutnosti dobře strukturovaných a obsáhlých dat. Vytvoření dobrého úplného záznamu kriminální činnosti trvá nezanedbatelný čas, který zaměstnanec, běžně určený do terénu, tráví u stolu v kanceláři. Nezbytným krokem pro úspěšné nasazení těchto systémů na našem území by tak musela být snaha minimalizovat administrativní zátěž získávání dat.

Pokud by i druhá připomínka byla vyřešena, pak nastává omezení vyplývající ze samotné podstaty té dané kategorie deliktu. Některá data není možné sesbírat z důvodu toho, že jsou neznámá. Například čas odcizení vozidla velmi často není znám přesně a většinou dostáváme interval hodin, ve kterých se daný čin mohl odehrát.

Nutno poznamenat, že policie v nynější době maximálně využívá možností, jenž jim poskytuje „excelovské“ řešení. Kriminalisté si postupem času vytvořili opravdu propracované žurnály, které umožňují filtrování, grafické zobrazování dat, exportování různých přehledů a podobně. Některé znalosti, které byly v tomto případě získány sofistikovaným způsobem pomocí dataminingu, kriminalisté našli pouze za pomoci Excelu. Toto je však možné pouze do určitého množství dat, poté již pouhý Excel nebude stačit.

Závěrem konzultace je, že jediným způsobem, jak zjistit možnosti použití datamining-ových nástrojů je tento přístup ověřit v praxi. Právě tento záměr by mohl případně vyústit v budoucí spolupráci.

6.3.3 VICLAS

Přestože by se z předchozích odstavců mohlo zdát, že u nás nyní není prostor pro používání takto pokročilých metod, existuje a využívá se systém, který se velice blíží systému, používanému v této práci. Tím systémem je VICLAS (**Vi**olent **C**rime **L**inkage **A**nalysis **S**ystem), sloužící pro vyhledávání spojitostí mezi násilnými trestnými činy.

Hlavní myšlenkou tohoto programu je poskytnout jednotnou platformu pro uchovávání detailních záznamů o násilných kriminálních činech, které budou nezávislé na lokaci a bude možno je sdílet napříč krajskými ředitelstvími, na rozdíl od lokálních žurnálů. Jak již bylo řečeno, kromě uchovávání těchto informací se v něm i hledají případy související. Praktické nasazení v České republice se však potýká s několika potížemi.

Zaprvé tento systém přestavuje velkou administrativní zátěž pro vyšetřovatele, jelikož obsahuje zhruba 150 detailních otázek, na které vyšetřovatel musí získat odpovědi, ať už výslechem nebo vyšetřováním. Tyto informace je potřeba do systému zadat.

Za druhé, je zde problém s rozdílností prostředí, ve kterém byl systém vyvinut a prostředí, kde je provozován. Jedná se o systém z Kanady, kde rozloha a počet případů několikanásobně převyšuje situaci v zemích, jako je Česká republika, čímž je degradován její hlavní přínos v centralizaci záznamů a schopnosti pracovat s případy z rozsáhlých oblastí, kde není možné, aby se data vyměňovala například jen na základě žádosti té dané oblasti do oblasti jiné.

Zatřetí, přínos VICLAS není prozatím tak znatelný, jelikož stále ještě není dostatek dat pro tento systém, přičemž se očekává jeho využití v momentě, kdy trestanci začnou opouštět nápravná zařízení a budou opakovat svou předešlou trestnou činnost.

7 Závěr

Téma práce je informatické, ale v této době není datamining součástí studijních programů. Prakticky celý zimní semestr probíhalo intenzivní studování dataminingové techniky, algoritmů a dataminingového nástroje, který má fakulta k dispozici – IBM SPSS Modeler ve verzi 14.2. Navazující práce byla aplikací získaných znalostí a informací do několika úkolů.

Pro připravovaný předmět datamining bylo zadáno rozpracovat konkrétní případovou studii nad daty o kriminálních činech ve virtuální lokalitě.

Součástí tohoto úkolu bylo aplikovat dataminingové postupy na obdržená data, obsahující údaje o kriminálních činech za jeden rok. Pro řešení bylo nutno:

1. Vytvořit model, určující ideální rozmístění hlídek pro danou denní dobu
2. Vytvořit model, schopný nalézat související případy vloupání do obytných budov
3. Popsat výskyty kapesních krádeží z hlediska času a lokality

Model určující optimální rozmístění hlídek dle denní doby

Během návrhu a řešení první části bylo vyzkoušeno více přístupů k problematice ideálního rozmístění hlídek. Za účelem nalezení řešení, které by bylo schopné obstát v praxi, bylo vyzkoušeno několik modelovacích algoritmů, ze kterých byl vybrán ten, který vykazoval nejlepší výsledky. Stěžejní problematikou bylo určení počtu hlídek (shluků) a zvolení takového algoritmu, který nebude příliš náchylný k vlivu odlehlých pozorování. Poslední zmíněné by vedlo k neefektivnímu využití hlídek. V závěru se na základě různých přístupů rozhodlo, že nejlepší řešení bude určovat počty hlídek subjektivně a nespoléhat se na algoritmy se schopností optimálního určení počtu shluků. Vzhledem k uspokojujícím výsledkům, byl jako algoritmus, nejméně náchylný k odlehlým pozorováním, vyhodnocen TwoStep.

Model schopný nalézt související případy vloupání do obytných budov

Opět se nejednalo o zadání, které by od začátku vedlo ke konkrétnímu řešení. Z původního plánu použít jeden algoritmus, bylo po patřičném prostudování literatury usouzeno, že shlukovací algoritmy jsou natolik náchylné vůči pořadí objektů do nich vstupujících, že bude potřeba vytvořit robustní sadu několika modelů, kdy na základě vhodného sloučení a vyhodnocení dílčích výsledků se vytipují případy, vyskytující se ve stejných shlucích napříč všemi modely. Výsledkem tedy bylo šest modelů, přičemž se jednalo o tři K-Means a tři Kohonenovy mapy, kdy do každého typu algoritmu byla přiváděna stejná data, s jiným pořadím. Po následném spojení do jedné matice a nalezení skupin případů, tvořících shluky, nehledě na použitý algoritmus, či pořadí použitém na vstupu tohoto algoritmu, byly vybrány skupiny, mající četnost případů vyšší než jedna. Tyto skupiny byly zobrazeny do mapy. Díky tomu a díky zkoumání skupin z hlediska času, bylo možné označit několik skupin jako kandidáty na prošetření kriminalistou.

Popis výskytů kapesních krádeží z hlediska času a lokality

Posledním úkolem vypracování případové studie bylo zabývat se kapesními krádežemi a získat popis jejich výskytů v závislosti na času a lokalitě. Zde, na rozdíl od předchozích bodů případové studie, nebylo zapotřebí používat jakéhokoli modelovacího prostředku. Celý bod byl vypracován nahlédnutím do dat, kategorizací období v roce a zjištěním v jakých lokalitách se v průběhu celého roku krade. Spojením znalostí o čase se znalostmi o místě bylo zjištěno, že během celého roku dochází ke krádežím na lokalitě označené jako „Náměstí“ a v týdnech letní sezóny se navíc krade v parku a na stadionu.

Veškeré proudy (soubory z používaného dataminingového nástroje) je možné nalézt na DVD-ROM, který je součástí této práce jako zvláštní příloha.

Konzultace případové studie s odborníky z praxe

Pro posouzení relevantnosti použitých dat a postupů v rámci případové studie, došlo ke kontaktování kriminalistů na odboru analytiky KŘP LK, kteří představili současné analytické postupy a nástroje v praxi. Z hlediska této práce byla pozitivně byla vnímána podobnost školních dat s reálnými daty i možnost další spolupráce. Ukázalo se, že dataminingové postupy mohou přispět ke zvýšení bezpečnosti, ale nejednotnost dat a nedostatečný sběr dat, v danou dobu brání účinnému nasazení těchto moderních metod. Na malých úsecích se speciální a řídkou agendou se dodnes používá papírová dokumentace. Není to typické a časem tento jev zmizí. Vývoj informačních technologií a informačních systémů povede zcela jistě i v tomto úseku k zásadním změnám.

Implementace algoritmu

Dalším úkolem byla realizace algoritmu K-Means, resp. vlastní implementace algoritmu K-Means, který bude pracovat podobně jako K-Means používaný v IBM SPSS Modeleru. Toto vyžadovalo vytvořit objektový návrh, seznámit se s odlišnostmi konkrétní implementace ve zmíněném programu a algoritmus naprogramovat. Rovněž bylo potřeba vytvořit uživatelské rozhraní. Během studia specifické implementace řešení od IBM byla zjištěna potřeba obohatit K-Means o možnost zpracovat kategoriální data. To vedlo k potřebě „inteligentnějšího“ uživatelského rozhraní, umožňujícího uživateli upravovat datové typy načtených atributů. Po vytvoření takového rozhraní, byl samotný algoritmus vyzkoušen na datech, použitých při hledání souvisejících případů vloupání do obytných budov. Pro lepší čitelnost výsledků se zavedla možnost určité shluky v grafu skrýt. Výsledky této implementace a implementace v IBM SPSS Modeleru byly velmi podobné.

Veškeré zdrojové kódy a celý projekt je možné nalézt na DVD-ROM, který je součástí této práce jako zvláštní příloha.

Část kurzu Datamining

Poslední částí zadání bylo vytvořit část kurzu Datamining na e-learningovém portále FM TUL. V rámci posledního úkolu, praktické části bakalářské, práce byla vytvořena část kurzu *Datamining,* nesoucí název *Shluková analýza*. Dostupné na: https://elearning.fm.tul.cz/ Všechny materiály jsou také na přiloženém DVD.

8 Seznam použité literatury

[1] BERKA, Petr. *Dobývání znalostí z databází*. Praha: Academia, nakladatelství Akademie věd České republiky, 2003, s. 18. ISBN 80-200-1062-9.

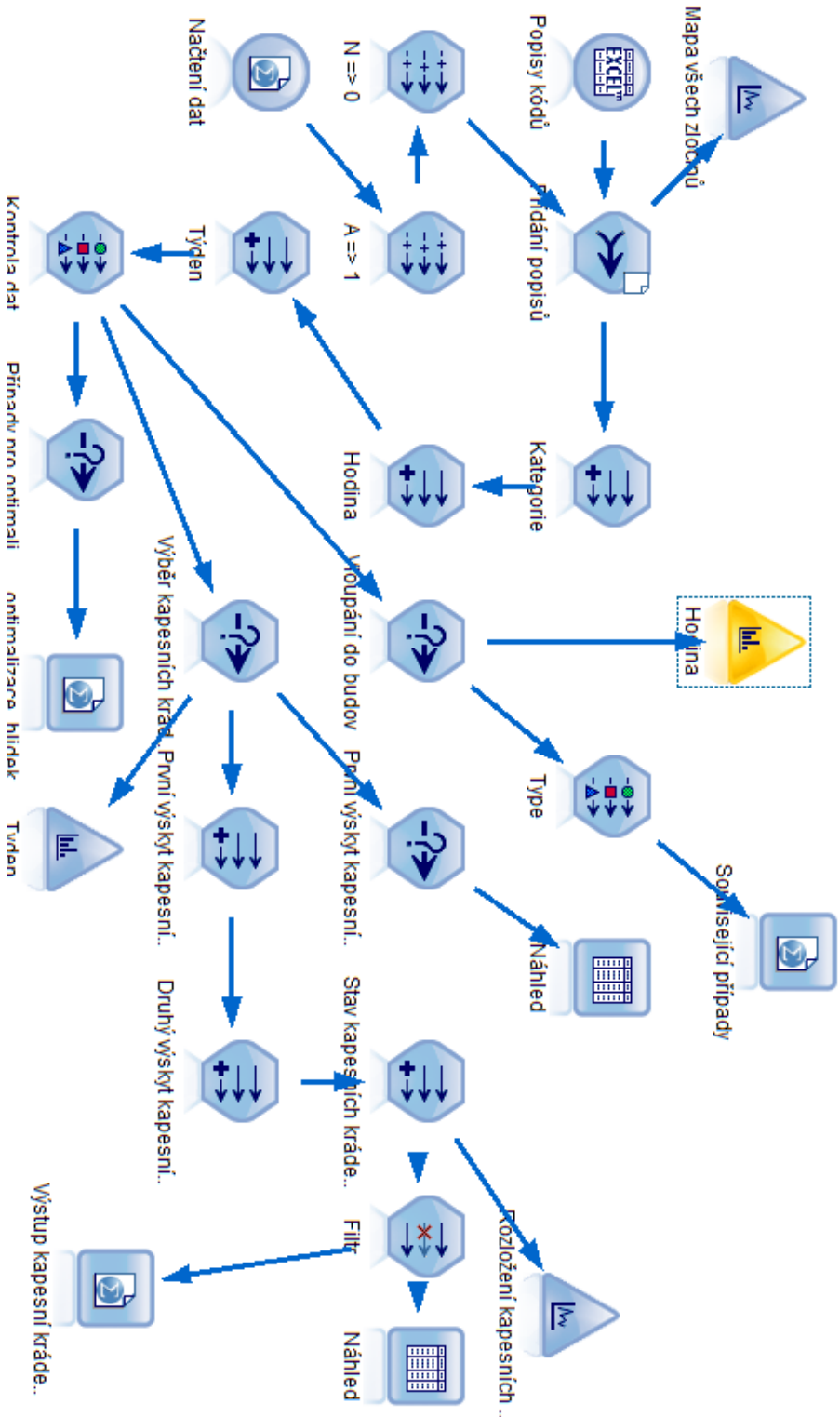
[2] BUDÍKOVÁ, Marie, Maria KRÁLOVÁ a Bohumil MAROŠ. *Průvodce základními statistickými metodami*. Praha: Grada Publishing, a.s., 2010, s. 213. ISBN 978-80-247-3243-5.

[3] HENDL, Jan. *Přehled statistických metod: Analýza a metaanalýza dat*. 3. přeprac. vyd. Praha: Portál, s. r. o., 2009, s. 493. ISBN 978-80-7367-482-3.

[4] LUKASOVÁ, Alena a Jana ŠARMANOVÁ. *Metody shlukové analýzy*. Praha: SNTL - Nakladatelství technické literatury, n. p., 1985, s. 118.

[5] IBM CORPORATION. *IBM SPSS Modeler 14.2 Algorithms Guide*. 2011. Dostupné z: ftp://ftp.software.ibm.com/software/analytics/spss/documentation/modeler/14.2/en/AlgorithmsGuide.pdf

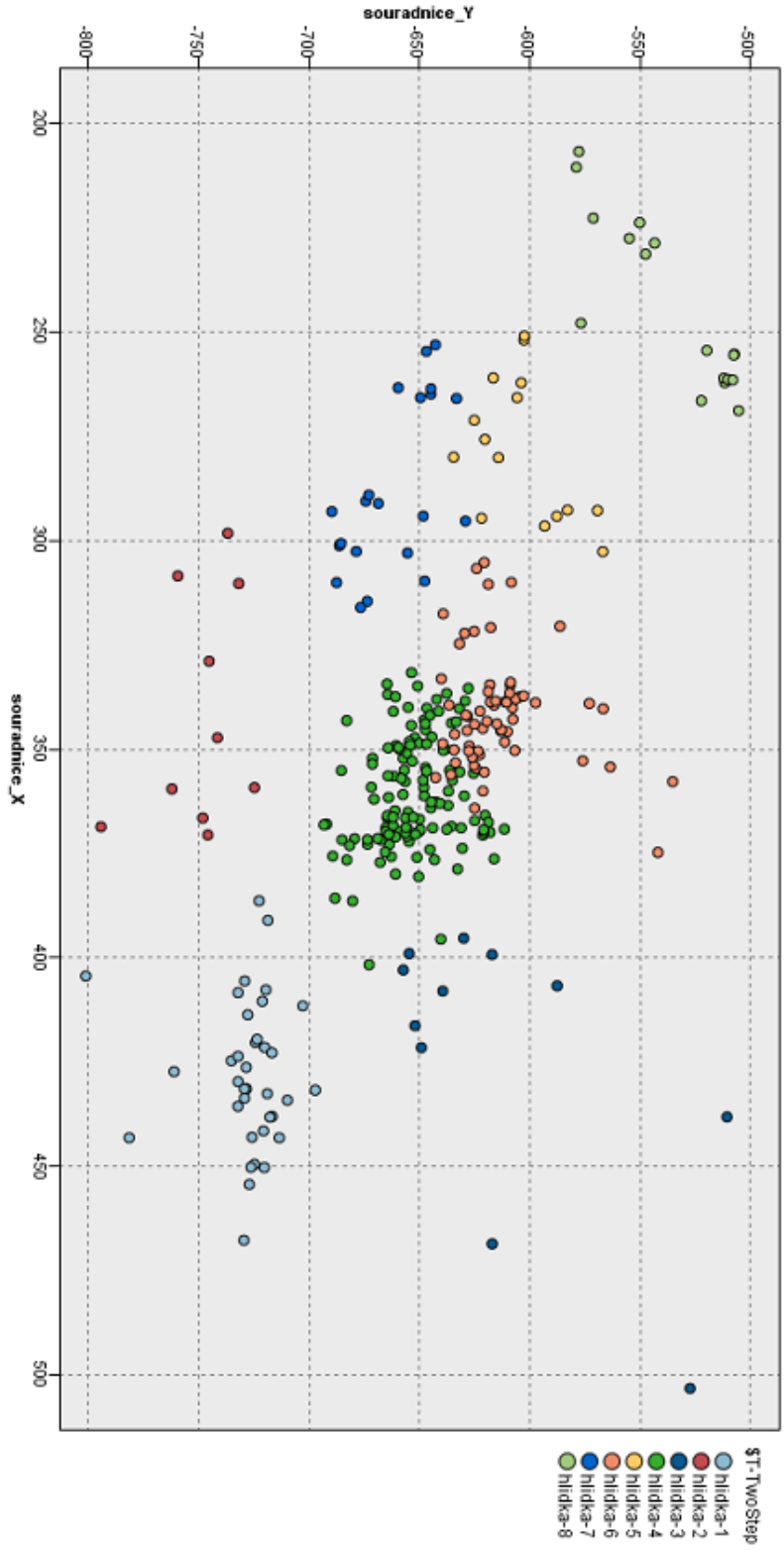
Příloha 1 – Proud „Příprava dat“



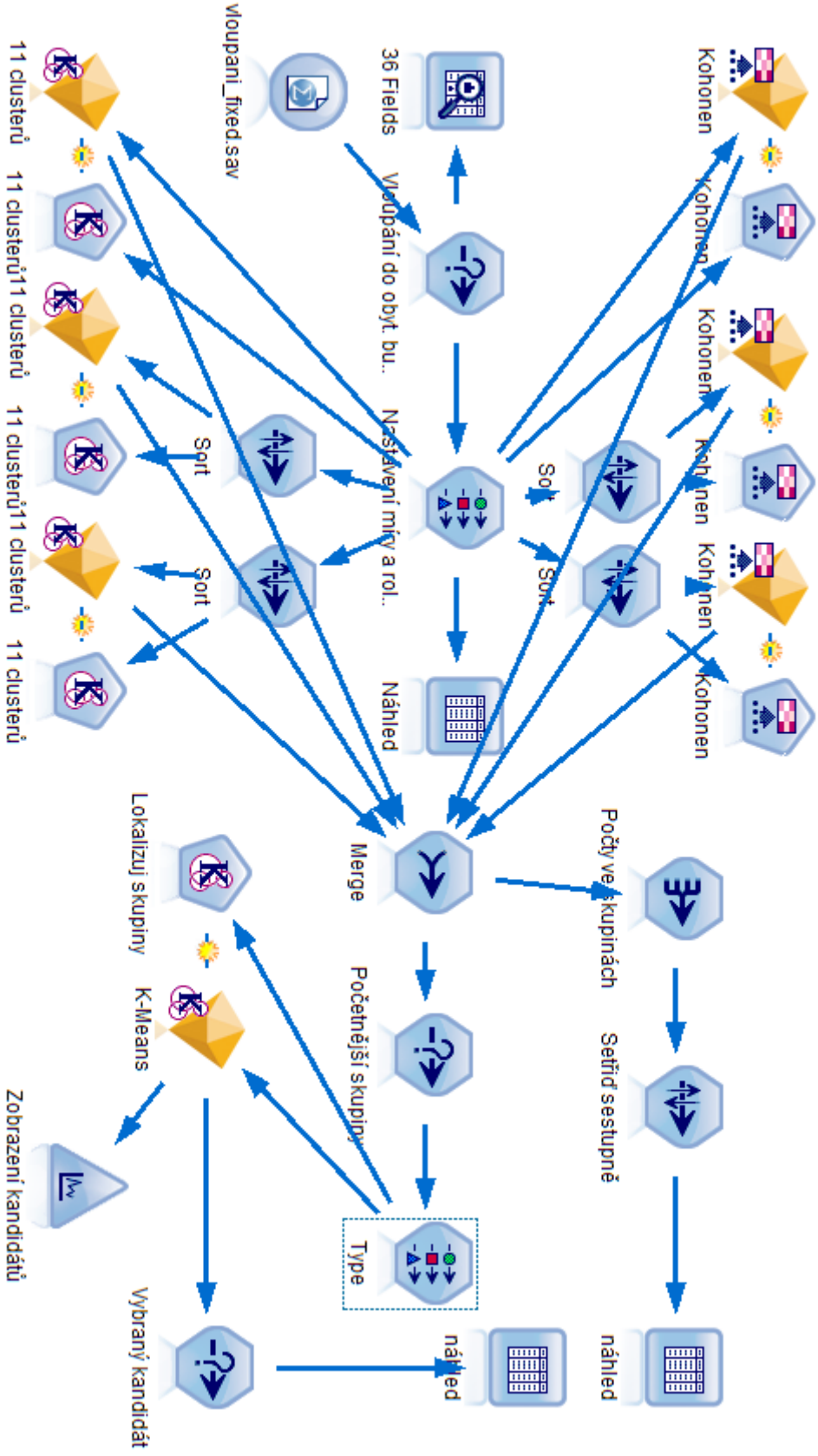
Příloha 2 – Kompletní seznam atributů vstupujících do úlohy

Field	Measurement	Values
cislo_pripadu	Continuous	[1.09701001E8,1.09701662E8]
Kod	Nominal	"008/01","008/06","028/03","030/0...
souradnice_X	Continuous	[210.0,500.0]
souradnice_Y	Continuous	[-900.0,-510.0]
Datum	Continuous	[1995-01-01 04:10:00,1995-12-3...
MO_vstup	Nominal	Lstí,Vloupání,Vstup
MO_misto	Nominal	Dveře,Okno,Pož_sch
MO_zabezp	Nominal	Alarm,Odemčeno,Otevřeno,Zamč...
MO_cinnost	Nominal	Charita,Obchod,Prodejna,Staveb...
MO_odchod	Flag	A/N
MO_zabdvere	Flag	A/N
MO_neporadek	Flag	A/N
MO_sejf	Flag	A/N
MO_zpusobvstupu	Nominal	"Rozbité dveře","Rozbité okno","Vy...
Kaud	Flag	A/N
Kvid	Flag	A/N
Kpocitac	Flag	A/N
Pobleceni	Flag	A/N
Ppenize	Flag	A/N
Pkredit	Flag	A/N
Plek	Flag	A/N
Ptelefon	Flag	A/N
Phodiny	Flag	A/N
Pkalk	Flag	A/N
Palkoh	Flag	A/N
Pzaznam	Flag	A/N
Psperky	Flag	A/N
Ppenez	Flag	A/N
Pdvere	Flag	A/N
Pokno	Flag	A/N
Pautomat	Flag	A/N
Ptelauto	Flag	A/N

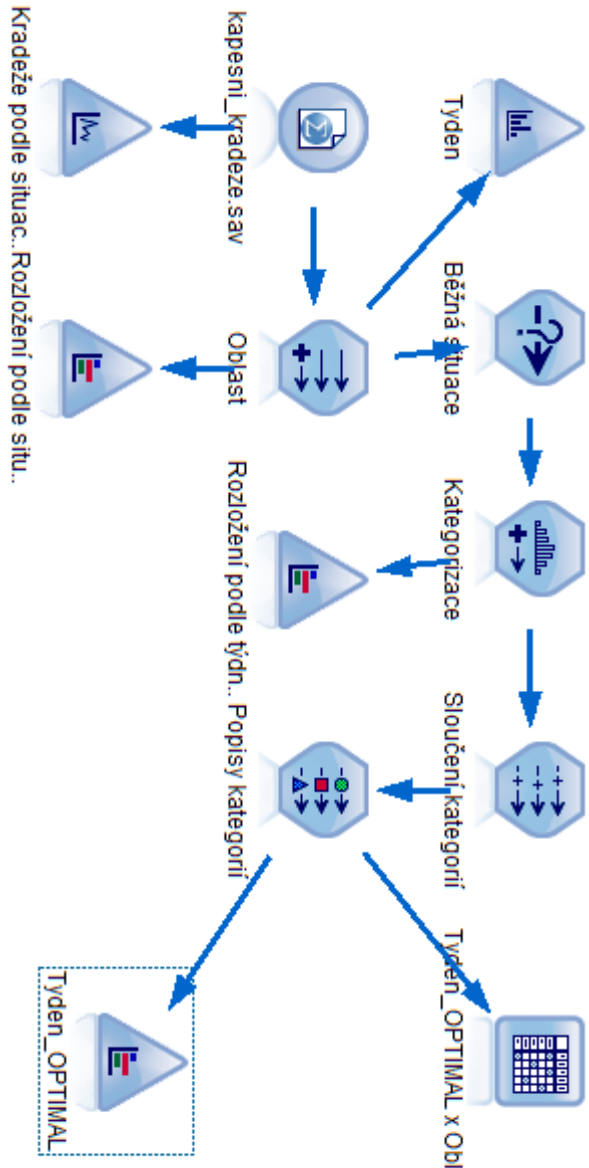
Příloha 3 – Optimalizace nasazení 8 hlídek pro danou oblast v noci



Příloha 4 – Proud nalezení souvisejících případů



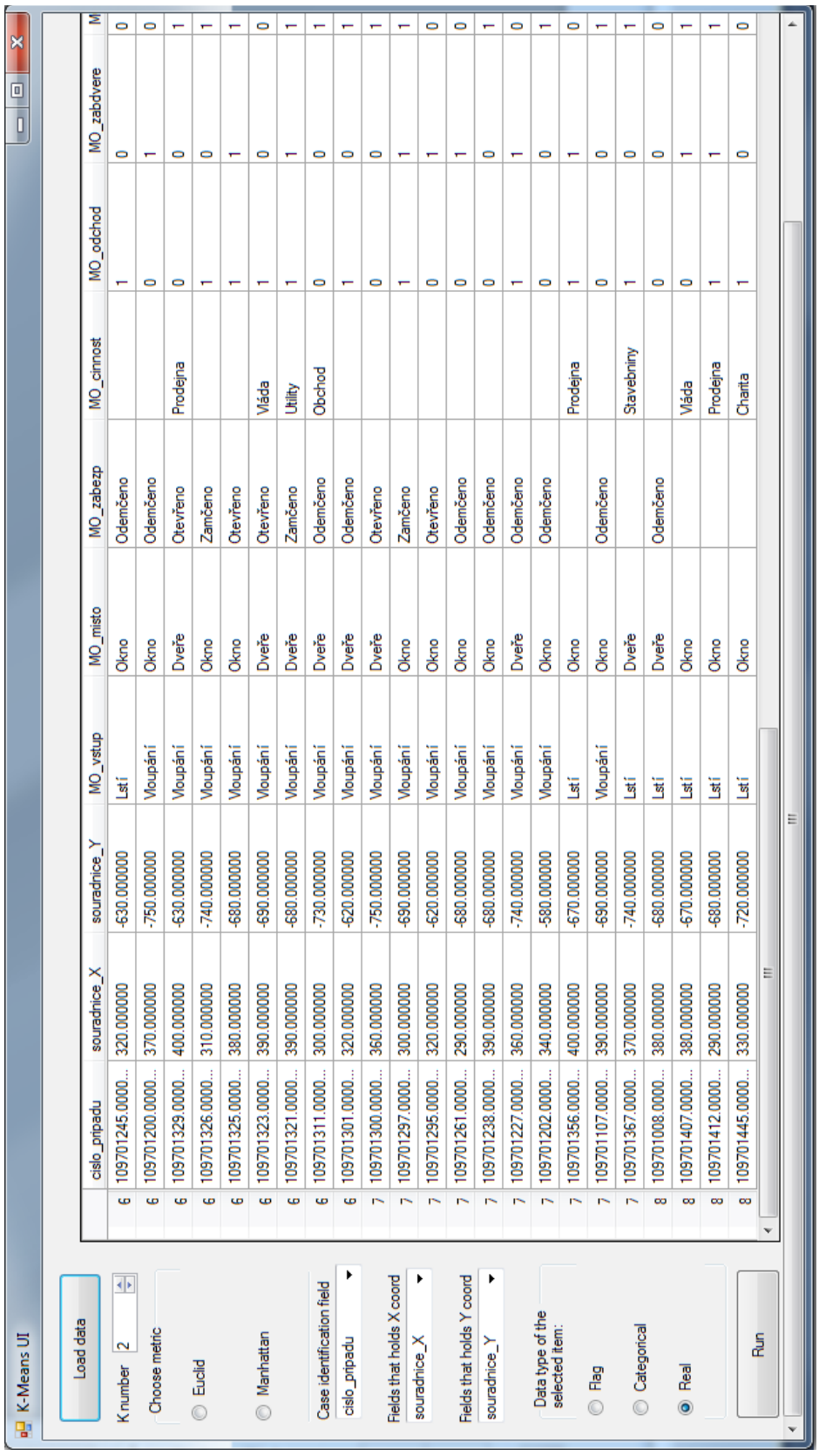
Příloha 5 – Proud zabývající se kapesními krádežemi



Příloha 6 – Hlavní metoda algoritmu

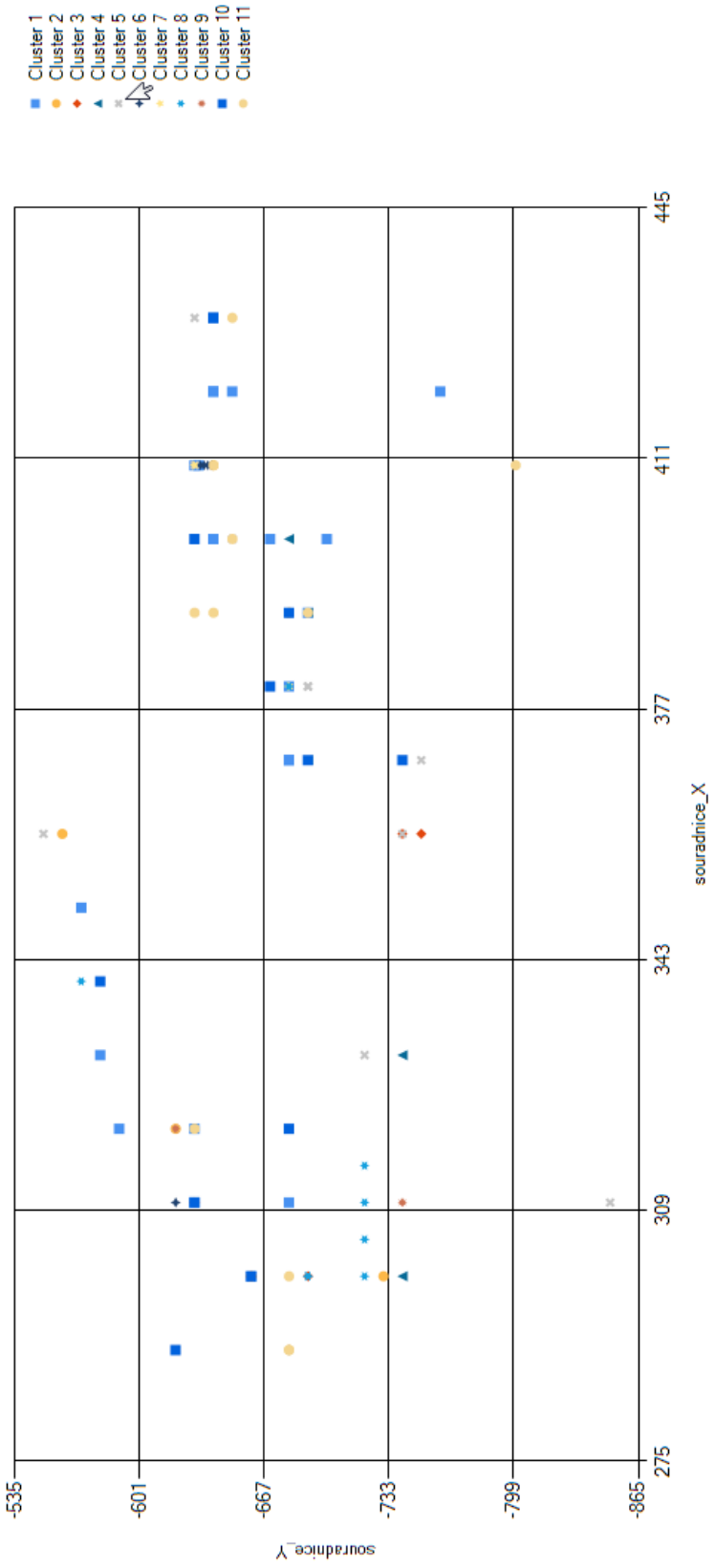
```
public static Tuple<List<int>, List<List<double>>> Run(List<string[]> _data, DataType[] types,
    UInt16 _count, Metric _metric = Metric.Euclid, UInt16 _maxiterations = UInt16.MaxValue)
{
    if (_count < 2) throw new ArgumentException("Minimum \"count\" argument value is 2!");
    if (_count > _data.Count) throw new ArgumentException("The number of cluster is too high!");
    if (_maxiterations == 0) throw new ArgumentException("There must be at least one iteration!");
    count = _count;
    maxiterations = _maxiterations;
    switch (_metric){
        case Metric.Euclid:
            metric = EuclideanDistance; //metric =  delegate; EuclideanDistance = function
            break;
        case Metric.Manhattan:
            metric = ManhattanDistance;
            break;
    };
    normalizeValues(_data,types); //normalizing real type attributes
    data = transformData(_data,types); //transforming categorical attributes on real attributes
    initializeClusters();//initialization of the clusters according to the IBM SPSS Modeler way
    bool change=true;
    while ((0<maxiterations--) && change) //main cycle
    {
        change = false;
        foreach (DataEntity item in data){ // for each object (each row in the data matrix)
            Cluster newcluster = GetClosestCluster(item); //select the closest cluster
            if (newcluster != item.Cluster) //if the the new one is different from previous one
            {
                change = true; //note that in this cycle there was a reassignment
                item.InsertToCluster(newcluster); //reassign the object to another cluster
            }
        }
        clusters.ForEach(x => x.CalculateCentroid()); // recalculate the centroids
    }
    var vectorsList = data.Select(x => x.Vector).ToList();
    var clustersTags = data.Select(x => clusters.IndexOf(x.Cluster)).ToList();
    return new Tuple<List<int>,List<List<double>>>(clustersTags,vectorsList);
}
```

Příloha 7 – Design uživatelského rozhraní



	cislo_pripadu	souradnice_X	souradnice_Y	MO_vstup	MO_misto	MO_zabezp	MO_cinnost	MO_odchod	MO_zabdvere	M
6	1097012450000...	320.000000	-630.000000	Lstí	Okno	Odemčeno		1	0	0
6	1097012000000...	370.000000	-750.000000	Vloupání	Okno	Odemčeno		0	1	0
6	1097013290000...	400.000000	-630.000000	Vloupání	Dveře	Otevřeno	Prodejna	0	0	1
6	1097013260000...	310.000000	-740.000000	Vloupání	Okno	Zamčeno		1	0	1
6	1097013250000...	380.000000	-680.000000	Vloupání	Okno	Otevřeno		1	1	1
6	1097013230000...	390.000000	-690.000000	Vloupání	Dveře	Otevřeno	Vláda	1	0	0
6	1097013210000...	390.000000	-680.000000	Vloupání	Dveře	Zamčeno	Utility	1	1	1
6	1097013110000...	300.000000	-730.000000	Vloupání	Dveře	Odemčeno	Obchod	0	0	1
6	1097013010000...	320.000000	-620.000000	Vloupání	Dveře	Odemčeno		1	0	1
7	1097013000000...	360.000000	-750.000000	Vloupání	Dveře	Otevřeno		0	0	1
7	1097012970000...	300.000000	-690.000000	Vloupání	Okno	Zamčeno		1	1	1
7	1097012950000...	320.000000	-620.000000	Vloupání	Okno	Otevřeno		0	1	0
7	1097012610000...	290.000000	-680.000000	Vloupání	Okno	Odemčeno		0	1	0
7	1097012380000...	390.000000	-680.000000	Vloupání	Okno	Odemčeno		0	0	1
7	1097012270000...	360.000000	-740.000000	Vloupání	Dveře	Odemčeno		1	1	0
7	1097012020000...	340.000000	-580.000000	Vloupání	Okno	Odemčeno		0	0	1
7	1097013560000...	400.000000	-670.000000	Lstí	Okno		Prodejna	1	1	0
7	1097011070000...	390.000000	-690.000000	Vloupání	Okno	Odemčeno		0	0	1
7	1097013670000...	370.000000	-740.000000	Lstí	Dveře		Stavebniny	1	0	1
8	1097010080000...	380.000000	-680.000000	Lstí	Dveře	Odemčeno		0	0	0
8	1097014070000...	380.000000	-670.000000	Lstí	Okno		Vláda	0	1	1
8	1097014120000...	290.000000	-680.000000	Lstí	Okno		Prodejna	1	1	1
8	1097014450000...	330.000000	-720.000000	Lstí	Okno		Charita	1	0	0

Příloha 8 – Vizualizace výstupu vlastní implementace K-Means



Příloha 9 – Výstup K-Means z IBM SPSS Modeler