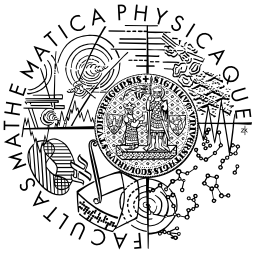


MATEMATICKO-FYZIKÁLNÍ
FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Petra Doubravová

Kvalitativní analýza neznámých dat

Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce:	RNDr. Josef Pelikán
Studijní program:	informatika
Studijní obor:	obecná informatika

Praha 2018

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Ráda bych poděkovala svému vedoucímu, RNDr. Pelikánovi, za vstřícnost, podporu a nadšení v průběhu tvoření této práce.

Název práce: Kvalitativní analýza neznámých dat

Autor: Petra Doubravová

Katedra: Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: RNDr. Josef Pelikán, katedra

Abstrakt: V současné době je produkováno stále více dat a roste poptávka po jejich zpracování. Tato práce si klade za cíl ukázat možnou cestu pro automatickou kvalitativní analýzu neznámých dat. Soustředí se nejen na automatickou aplikaci metod strojového učení, ale i na vizuální stránku reprezentace výsledných modelů. V práci je popsána aplikace odhalení redundance, lineární regrese, shluková analýza a možné způsoby dalšího rozšíření. Navíc přináší přehled možných problémů, které se v této oblasti mohou vyskytnout.

Klíčová slova: kvalitativní analýza dat redundance funkční závislost regrese shluková analýza

Title: Qualitative analysis of unknown data

Author: Petra Doubravová

Department: Department of Software and Computer Science Education

Supervisor: RNDr. Josef Pelikán, department

Abstract: Nowdays, more and more data is produced and demand for their analysis is growing steadily. This thesis is supposed to show a possible way for an automatical qualitative analysis of unknown data. Its focuse is not only on a mechanical application of machine learning methods, but also on a graphical reprezentation of the resulting models. In thesis, application of a redundancy revealing, linear regression, clustering is described as well as potencial ways for future extensions. In addition, overview of possible problems of this kind of task is discussed.

Keywords: qualitative data analysis redundancy functional dependency regression clustering analysis

Obsah

Úvod

Podle magazínu Forbes denně lidstvo vygeneruje $2{,}5 \cdot 10^{18}$ bytů dat a s expanzí internetu věcí bude toto číslo ještě stoupat [Marr (2018)]. Převážná většina evidencí o zaměstnancích, pacientech nebo zákaznících je v elektronické podobě a poptávka po datových analyticích stoupá. Předpokládá se, že tato data mohou obsahovat mnoho cenných informací. Vyznat se však v tabulce s desítkami či stovkami sloupečků a počtem řádků v řádu tisíců není jednoduché. Cílem této práce je vytvořit nástroj, který umožní automatizovat prvotní analýzu neznámých dat jak pro datového analytika, tak pro laika například v manažerské pozici. Pokud analytik dostane data, o kterých nic neví, tento nástroj by mu mohl ušetřit mnoho času úvodního zkoumání některých vztahů. Pro mechanické vyšetření všech možností lineárních závislostí mezi sloupci není potřeba lidské práce. Druhým typem potenciálního uživatele tohoto programu, který jsem nazvala Marple, jsou lidé, kteří se prací s daty nezabývají, ale chtěli by se o nějakém souboru informací dozvědět více. Mohou to být třeba už zmínění manažeři ve firmách, ale i vědečtí pracovníci, kteří jinak vyhledávají služeb odborníků pro odhalení vztahů z naměřených hodnot.

Celé téma automatické analýzy dat je provázeno dvěma hlavními problémy. První z nich je, že **data mohou být příliš velká** (tzv. curse of dimensionality), než aby bylo možné vyzkoušet všechny kombinace či možnosti a slepě prohledávat celý prostor. Druhý problém může na první navazovat, ale nemusí. Mnoho dílčích podproblémů, které je nutné vyřešit, je **velice snadných pro člověka, ale jejich algoritmizace může být zcela netriviální**. Když mluvíme o neznámých datech, nemusí to být přímo tabulka čísel, o které netušíme ani jména sloupečků. Pak by nám pravděpodobně ani odhalení vnitřní struktury k ničemu nebylo. Typicky víme, co který sloupeček znamená v tom významu, že známe jeho jméno a chápeme, co zhruba hodnoty v něm vyjadřují. Počítač, bohužel, tohle neví, takže odhalit na první pohled nepotřebné sloupce, jako je jméno, nebo rozpoznat, jak s kterým sloupcem zacházet, je těžké. Většina použitých metod vyžaduje nastavení metaparametrů - popis toho, jak se má metoda chovat, kriteria pro její rozhodování a zastavení. Existují samozřejmě některé typické hodnoty nebo jejich rozsahy, které jsou pro daný druh metody vhodné, ale během práce na projektu jsem narazila na mnoho případů, kdy přesné nastavení zcela jasné nebylo. S oběma záležitostmi se musíme nějak vyrovnat. Jednou cestou by bylo připustit větší interakci uživatele pro nastavení parametrů a učinění některých rozhodnutí. Druhá možná cesta je použít metody hlubokého učení a pokusit se naučit správné meta-nastavení. Pro problém dimenzionality dat je třeba navrhnout vhodné ořezání možností, nebo algoritmy, kterým toto nevadí.

Cílem této práce nemá být navrhování nových metod strojového učení, ale prozkoumání cesty, jak zautomatizovat jejich použití, popsat problémy, které se mohou vyskytnout a navrhnout řešení. Důležitá je i forma prezentace výsledků, která má být přehledná a jednoduchá i pro neodborníka, takže důležitou součástí práce je úvaha nad znázorněním nalezených závislostí tak, aby byly snadno dostupné.

V následujích kapitolách bude popsáno jak teoretické zázemí, použité metody a učiněná rozhodnutí, tak implementace a ovládání programu. V **první kapitole** budou popsány metody strojového učení, které byly implementovány, diskuze o jejich použití a problémy, které byly vyřešeny. Implementační detaily, některé zajímavé použité knihovny a architektura programu bude popsána v **kapitole druhé**. Uživatelské dokumentaci, tedy popisu zacházení s programem je věnována **třetí kapitola**.

1. Popis problému

V této kapitole budou popsány jednak metody strojového učení použité na vstupní data a zároveň i výstup programu. Bude tedy popsáno, co všechno lze v reportu nalézt a jak byly shluk údaje vypočítány.

Nejprve je třeba si říci, co všechno bychom o neznámých datech vůbec mohli chtít zjišťovat a na jaká omezení při tom narazíme. Je žádoucí vědět nejprve něco o samostatných proměnných a pak teprve o vztazích mezi nimi. Řádky, které nejsou úplné, budou vyřazeny.

1.1 Spojité a kategorické proměnné

U jednotlivých sloupečků nás prvně zajímá jakého jsou typu. Můžeme rozlišovat, jestli jde o spojitou proměnnou, diskrétní nebo kategorickou. Kategorickou proměnnou rozumíme proměnnou reprezentující nějaké "škatulky", třeba pohlaví nebo národnost. Diskrétní proměnná ještě nutně nemusí být kategorická. Sporným příkladem může být věk, který lze brát kategoricky. Ovšem v případě, že se vyskytují hodnoty 0 - 100, lze říci, že je to proměnná číslená. Pokud bychom chtěli roztřídění do určitých kategorií, tak je vhodnější zavést si novou proměnnou s hodnotami "dítě"- "dospělý"- "senior", nebo o něco jemnějšími.

Pokud je proměnná textová, opět není jasné, že je musí jít o výčet. Jako pomocný dataset pro tuto kapitolu bude sloužit Pittsburgh Police dataset[arr (2018)] obsahující záznamy o trestných činech v Pittsburghu (viz Tabulka 1.1). Proměnné týkající se přesného místa činu (INCIDENT_LOCATION) jsou natolik rozmanité, protože záznam je veden na úrovni ulice i s číslem popisným, že se nedá očekávat, že by proměnná mohla být úspěšně brána jako kategorická. Každá kategorie by se totiž vyskytla typicky jednou. Jiná situace by nastala, pokud by se často opakovala jen některá místa, další možná (ale velce nepravděpodobná) situace je, že počet trestných činů třeba souvisí s velikostí čísla popisného. Jinak se ale tato proměnná neuplatní ani pokud její hodnoty zakódujeme do čísel, protože nově vzniklá čísla mají nové logické uspořádání podle velikosti, což patrně původní jména ulic neměla. Když si ale odmyslíme číslo ulice, nebo dokonce abstrahujeme na obasti nespecifikované velikosti, mohlo by být už zajímavé, protože můžeme objevit souvislosti mezi trestnými činy a nějakou oblastí města. V našem případě to není třeba přímo z ulice, protože dataset obsahuje i proměnné popisu-

_id	AGE	GENDER	ARRESTTIME	ARRESTLOCATION	INCIDENTNEIGHBORHOOD
1787	62	M	Mon, 03/10/16 04:47:00 PM PST	S Main ST & Wabash ST Pittsburgh, PA 15220	West End
14902	23	M	Tue, 21/11/17 08:00:00 PM PST	900 Block 2nd AV Pittsburgh, PA 15219	Hazelwood
1788	29	F	Mon, 03/10/16 04:50:00 PM PST	Forbes AV & Market ST Pittsburgh, PA 15222	Central Business District
1790	32	M	Wed, 05/10/16 09:12:00 PM PST	Southern AV & Gray ST Pittsburgh, PA 15211	Mount Washington
14908	45	M	Tue, 21/11/17 10:00:00 AM PST	600 Block 1st AV Pittsburgh, PA 15219	New Homestead
14911	19	F	Tue, 21/11/17 02:14:00 PM PST	600 Block 1st AV Pittsburgh, PA 15219	Mt. Oliver Boro
6753	35	M	Thu, 23/03/17 06:35:00 PM PST	3900 Block Mintwood ST Pittsburgh, PA 15201	Lower Lawrenceville
7808	36	M	Wed, 26/04/17 08:37:00 AM PST	Spring ST & Eleanor ST Pittsburgh, PA 15210	Arlington
8413	32	M	Wed, 10/05/17 11:46:00 PM PST	Letsche ST & Mercy ST Pittsburgh, PA 15214	Fineview
15905	24	F	Tue, 19/12/17 11:50:00 PM PST	Federal ST N & Perrysville AV Pittsburgh, PA 15212	Central North Side
12787	56	M	Tue, 19/09/17 12:30:00 PM PST	1500 Block 5th AV Pittsburgh, PA 15132	Middle Hill

Tabulka 1.1: Ukázka dat k analýze

jící oblast a okrsek místa. Můžeme se u toho ale teoreticky zamyslet, jak takovou souvislost objevit. Ideální by bylo použít kódování, které by vystihovalo blízkost jednotlivých míst, třeba souřadnice. K tomu by ale byla potřeba mapa Pittsburghu. Co když ale vznikne podobná potřeba při kódování jiné proměnné, která se netýká zeměpisné blízkosti a žádná mapa v tom nepomůže, i kdybychom je měli k dispozici? Podobně v případě shlukové analýzy bychom potřebovali jiné vyjádření než jen nahrazení jednotlivých položek výčtu čísly. O shlukové analýze bude řeč později, ale podstatné je, že se snaží hledat v datech nějaké skupinky, které jsou si hodně podobné a naopak každá skupinka se hodně liší od všech ostatních. V tomto kontextu jen převedení na čísla bude zase způsobovat probémy, protože máme-li nějaké proměnné nahrazené čísly 1 a 2, budou se jevit více podobné než 1 s kategorii kódovanou číslem 8. Ve skutečnosti jsou si ale všechny proměnné ve výčtovém typu "stejně blízko".

Zde jsou tyto problémy vyřešeny následovně. Pokud jde o první problém, nejrozumnější by bylo, aby uživatel tuto proměnnou vyřadil z výpočtu podobně třeba jako vlastní jména osob z dříve uvedených důvodů. Samotný program to ale neučiní a proměnnou prostě převede na číselné kódování. Pro účely shlukové analýzy jsou data převedená do one-hot kódování, které je pro toto vhodnější[Goodfellow a kol. (2016), str. 150]. Je otázka, zda použít toto kódování i pro lineární regresi. Knihovna, kterou požívám pro implementaci, toto u kategorických proměnných dělá automaticky a rozhodla jsem se to nijak explicitně neměnit. Je to totiž vhodné pro lineární regresi udělat [James a kol. (2018)], z podobných důvodů jako pro shlukovou analýzu. Pro různá pořadí zakódování proměnné by totiž vycházely různé koeficienty a především různé pravděpodobnosti, že proměnná je pro předpověď užitečná. Zbývá zodpovědět otázku, jak proměnnou poznat. Jako nejpřmočařejší mi přišlo porovnat počet unikátních hodnot ve sloupci ku počtu řádek. Kriterium pro podíl si uživatel může nastavit sám, zrovna jako označit rovnou některý sloupec za kategorický. Navíc jsem připojila podmínku, že pokud sloupec obsahuje jen dvě hodnoty, tak je považován za kategorický bez ohledu na kriterium (v našem případě sloupec GENDER).

1.2 Odstranění duplicit

1.2.1 Jednoduché

Pokud už víme dost o jednotlivých sloupcích, první krok v odhalování závislostí mezi nimi je zjisti, zda některé louepčky nejsou duplicitní. To se může stát několika způsoby. První je triviální případ, že hodnoty jsou skutečně stejné. Pro odhalení případů kdy data obsahují stejné sloupečky, ale každý je kódovnaý jinak (muž/žena x M/F), jsou potom při kontrole duplicit sloupečky převedeny na výčet a ohodnoceny čísly ve stejném pořadí, takže i toto je odhaleno. Zde zároveň můžeme najít závislost, která není vyloženě duplicitou v tom smyslu, že sloupec kódoval stejnou informaci, ale třeba i opačnou. Samozřejmě problém může nastat u spojitých proměných, protože data s n řádky mohou obsahovat n různých hodnot a pokud je takových sloupců více, budou všechny označeny jako redundantní. Náš program proto zkouší tento postup jen na kombinace text-text, text-celé číslo a vice versa.

1.2.2 Korelace

Složitejším případem je, že dvě proměnné na sobě závisí téměř přesně a není potřeba oba sloupce uchovávat. Může se to stát třeba pokud omylem v datech máme dva sloupce se stejnou informací, ale v jiných jednotkách. Prvním plánem bylo použít 1:1 lineární regresi, která původně i byla implementována. Vzniká zde ale problém, jak rozhodnout přesnost modelu. Rychlejší a zajímavější je však použít korelaci či korelační podíl (correlation ratio). Korelace je hodnota v intervalu $[-1,1]$ nebo v případě korelačního podílu $[0,1]$ a funguje jako míra lineární závislosti mezi dvěma proměnnými, tedy funguje z hlediska interpretace stejně jako lineární regrese, ale neposkytuje přesné koeficienty. Pro dva sloupce s číselnými hodnotami stačí spočítat korelaci[James a kol. (2018)]. Složitější situace nastává v případech dvou kategorických proměných a mixu kategorické a spojité proměnné.

Pro kategorické proměnné jsem použila normalizovaný $\chi - kvadrát$ test. $\chi - kvadrát$ test pro vyjadření korelace mezi kategorickými proměnnými je běžně používaná metoda[lumenlearning][cor]. Výsledné číslo ale bohužel neleží v nějakém rozumně omezeném intevalu, takže je ještě třeba celý výsledek podělit počtem řádků. Výsledkem je číslo v uzavřeném intervalu $[0,1]$.

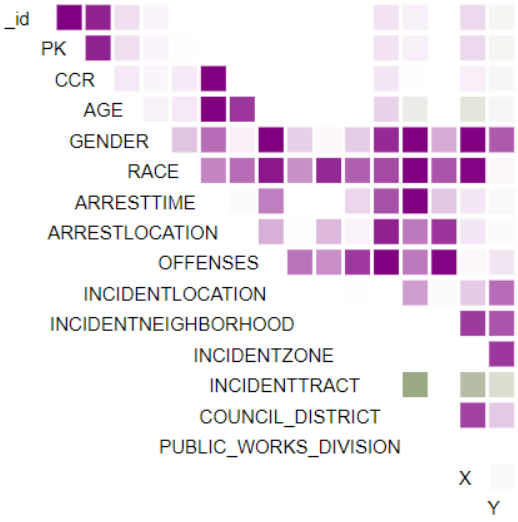
Zde byl použit druhý způsob a to ukazovat v reportu p-value pro jednotlivé korelace, což je vlastně pravděpodobnost, že proměné jsou korelované. Dosáhnu požadovaného oboru hodnot a číslo má přitom lepší interpretaci. Pro případy, kdy jedna proměnná je kategorická a druhá nikoliv, jsem použila korelační koeficient (correlation raito),jak je používán i v faktorové analýze smíšených dat (factor analysis of mixed data, dále jen FAMD).[fam][Fac (2018)] Je ale možné, že $\chi - kvadrát$ test nebude dobře fungovat,pokud data neplňující následující podmínky: data jsou dostatečně velká, méně než 20% proměnných má frekvenci výskytu menší než 5 a žádná nemá méně než 0. Jinak je vhodnější použít Fisherův test[lumenlearning]. Tato malá data v mojí práci nepředpokládám, pravděpodobně by nebyly použity jako výčtový typ.

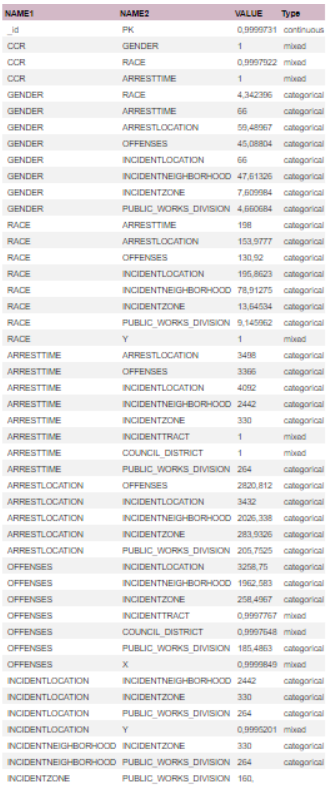
1.2.3 Vizualizace

Původní implementace zobrazení matice jen zobrazovala sloupečky, které byly kvůli vysoké korelaci vyřazené. V některých případech však byla tabulka může být příliš dlouhá a navíc z ní není dobře poznat, které všechny závislosti se zde objevují. Navíc by moho být zajímavé vědět o korelovaných sloupcích i pokud korelace není tak silná, aby jeden z nich byl vyřazen. Nakonec je použito zobrazení pomocí korelační matice s barevnou škálou označující míru korelace (viz Tabulka 1.1)). Tím je dosaženo zobrazení všech informací v rámci jedné obrazovky. Po kliknutí na určitý čtvereček bude zobrazena přesná hodnota korelace.

1.3 Výběr proměnných

V této části bych ráda popsala další problém, který je třeba nějak uchopit a to sice výběr proměnných (variable selection) nebo také redukci dimenze. Jak už bylo vidět z předchozí části o kategorických a spojitých proměnných, ne všechny





NAME1	NAME2	VALUE	Type
_id	PK	0,9999731	continuous
CCR	GENDER	1	mixed
CCR	RACE	0,9997922	mixed
CCR	ARRESTTIME	1	mixed
GENDER	RACE	4,342396	categorical
GENDER	ARRESTTIME	66	categorical
GENDER	ARRESTLOCATION	59,48967	categorical
GENDER	OFFENSES	45,08804	categorical
GENDER	INCIDENTLOCATION	66	categorical
GENDER	INCIDENTNEIGHBORHOOD	47,61326	categorical
GENDER	INCIDENTZONE	7,609984	categorical
GENDER	PUBLIC_WORKS_DIVISION	4,660684	categorical
RACE	ARRESTTIME	198	categorical
RACE	ARRESTLOCATION	153,9777	categorical
RACE	OFFENSES	130,92	categorical
RACE	INCIDENTLOCATION	195,8623	categorical
RACE	INCIDENTNEIGHBORHOOD	78,91275	categorical
RACE	INCIDENTZONE	13,64534	categorical
RACE	PUBLIC_WORKS_DIVISION	9,145962	categorical
RACE	Y	1	mixed
ARRESTTIME	ARRESTLOCATION	3498	categorical
ARRESTTIME	OFFENSES	3366	categorical
ARRESTTIME	INCIDENTLOCATION	4092	categorical
ARRESTTIME	INCIDENTNEIGHBORHOOD	2442	categorical
ARRESTTIME	INCIDENTZONE	330	categorical
ARRESTTIME	INCIDENTTRACT	1	mixed
ARRESTTIME	COUNCIL_DISTRICT	1	mixed
ARRESTTIME	PUBLIC_WORKS_DIVISION	264	categorical
ARRESTLOCATION	OFFENSES	2820,812	categorical
ARRESTLOCATION	INCIDENTLOCATION	3432	categorical
ARRESTLOCATION	INCIDENTNEIGHBORHOOD	2026,338	categorical
ARRESTLOCATION	INCIDENTZONE	283,9326	categorical
ARRESTLOCATION	PUBLIC_WORKS_DIVISION	205,7525	categorical
OFFENSES	INCIDENTLOCATION	3258,75	categorical
OFFENSES	INCIDENTNEIGHBORHOOD	1962,583	categorical
OFFENSES	INCIDENTZONE	258,4967	categorical
OFFENSES	INCIDENTTRACT	0,9997767	mixed
OFFENSES	COUNCIL_DISTRICT	0,9997648	mixed
OFFENSES	PUBLIC_WORKS_DIVISION	185,4863	categorical
OFFENSES	X	0,9999849	mixed
INCIDENTLOCATION	INCIDENTNEIGHBORHOOD	2442	categorical
INCIDENTLOCATION	INCIDENTZONE	330	categorical
INCIDENTLOCATION	PUBLIC_WORKS_DIVISION	264	categorical
INCIDENTLOCATION	Y	0,9995201	mixed
INCIDENTNEIGHBORHOOD	INCIDENTZONE	330	categorical
INCIDENTNEIGHBORHOOD	PUBLIC_WORKS_DIVISION	264	categorical
INCIDENTZONE	PUBLIC_WORKS_DIVISION	160,	

Obrázek 1.1: Srovnání vizualizace korelace.

Vlevo: Nově použité zobrazení pro korelaci. Veškeré závislosti lze spatřit najednou. *Vpravo:* Původní zobrazení. V případě, že vidíme vše najednou, písmena jsou již nečitelná.

proměnné chceme zahrnout do výpočtu. Může to být proto, že nejsou podstatné - například id záznamu, nebo pro to, že prostor všech možností je příliš velký. I kdyby nebyl tento proces tak náročný na výpočet, rozhodně by byl nepřehledný výstup, protože z tisíce možností pro lineární regresi se výsledný model nevybírá uplně lehko. Jak je tomu přesně popíši podrobně v části, která se na ní zaměřuje.

1.3.1 PCA

Bylo by zajímavé poměřit, jak moc je který sloupec důležitý. Určitou představu získáme z lineární regrese a ze shlukové analýzy. Z lineární regrese přímo, nepoužité sloupce se nevyskytují v žádné skupině sloupců, která by mezi sebou měla lineární závislost. To ale ještě nemusí nic znamenat, závislosti mohou být třeba kvadratické nebo i mnohem složitější. Ze shlukové analýzy vidíme, že pokud je nějaká proměnná rozložena v jednotlivých skupinách rovnoměrně, pak zřejmě není tou proměnou, která by skupiny od sebe odlišovala. Dalším možným způsobem, jak se na důležtost proměnné podívat je její rozptyl. Toto bere do úvahy anaýza základních komponent (principal component analysis, dále PCA). PCA se snaží data zobrazit do prostoru s takovou dimnenzí aby žádný sloupeček nebyl zbytečný. Zobrazuje data tak, aby jednotlivé komponenty (nové sloupečky) byly seřazeny od té, co vysvětluje nejvíce rozptylu v datech. Pak je možno si místo původního počtu sloupečků vybrat třeba jen 60%, které ale mohou popi-

sovat 99% rozptylu dat[de Souza (2012)][James a kol., 2018]. Výsledná matice je zjednodušeně řečeno matice vlastních vektorů kovarianční matice původních dat. Příslušná vlastní čísla pak vyjadřují, kolik procent původní variance daná komponenta vysvětluje[Smith (2002)]. Typické použití je takové, že data se zobrazí do nového prostoru a na tato nově získaná data se aplikují další metody, třeba právě lineární regrese. Narazila jsem zmínky o tom[PCA], že lze PCA použít na porovnání důležitosti proměnných. Stačí vzít tolik komponent, aby vysvětlovaly požadované procento rozptylu.Potom lze z každé komponenty zjistit, jak moc který sloupec přispěl k její důležitosti. Stačí tedy udělat průměr pro každý sloupec přes všechny vybrané komponenty vážený důležitostmi komponent. Nenašla jsem však k tomuto žádný věrohodný vědecký zdroj, který by doporučoval PCA používat tímto způsobem. Jelikož jsem se neměla o co opřít, navíc třeba odlišné měřítko jednotlivých sloupců může výrazně ovlivnit výsledek, rozhodla jsem se tuto metodu nepoužít přímo na selekci proměnných pro shlukovou analýzu a lineání regresi, ale v reportu je uvedená předpokládaná dimenzionalita dat. Ta je získaná na základě velikostí vlastních vektorů. Vlastní vektory blízké nule totiž indikují nadbytečný sloupec. Jsem tu nucena zmnit i implementační detail, protože výpočet PCA tak, jak ho provádí knihovna Accord.Net použitá pro většinu statistické práce, takto nefunguje[de Souza (2012)]. Naproti tomu knihovna pro PCA v R ano[Fac, 2018]. Protože data nejsou jen číselná, bylo třeba použít podobně interpretovatelné metody pro kategorická data - MCA(multiple correspondence analysis) a pro smíšená data FAMD, která bylo zmíněna výše. Korelační matice použité v těchtot metodách se shodují s korelačními maticemi použitými pro hledání duplicit.

1.4 Regrese

Lineární regrese je jednoduchá metoda pro nalezení lineářích funkčních závislostí mezi sloupci. Hledá záslosti ve tvaru

$$y = \beta_0 + \sum_{1}^{k} \beta_i x_i$$

. Proměnnou y, která je předpovídaná, budeme označovatjako odpověď. Takováto rovnice nám říká, jak lze předpovědět odověď z proměnných x_i.

1.4.1 Redukce dimenze

Pokud bychom chtěli vyzkoušet všechny možnosti všech podmnožin celých dat, dostaneme se k $\sum_1^n(\binom{n}{i})(n-i)$. Toto je tedy jedno z míst, kde bychom chtěli prohledávaný prostor prořezat a použít nějaké šikovnější řešení. První věcí je selekce sloupců, které do regrese budou zahrnuty. Jak bylo uvedeno v sekci 1.2, vysoce korelované sloupce jsou před regresí vyřazeny. To je důležité, protože přítomnost vysoce korelovaných proměnných v regresi není žádoucí[Goodfellow a kol. (2016)]. Pro další omezení počtu proměnných bychom mohli použít redukci dimenze pomocí PCA, vyřazení proměnných dle informací z PCA nebo selekci v průběhu zkoušení možností. První metoda je užitečná a používaná[James a kol.

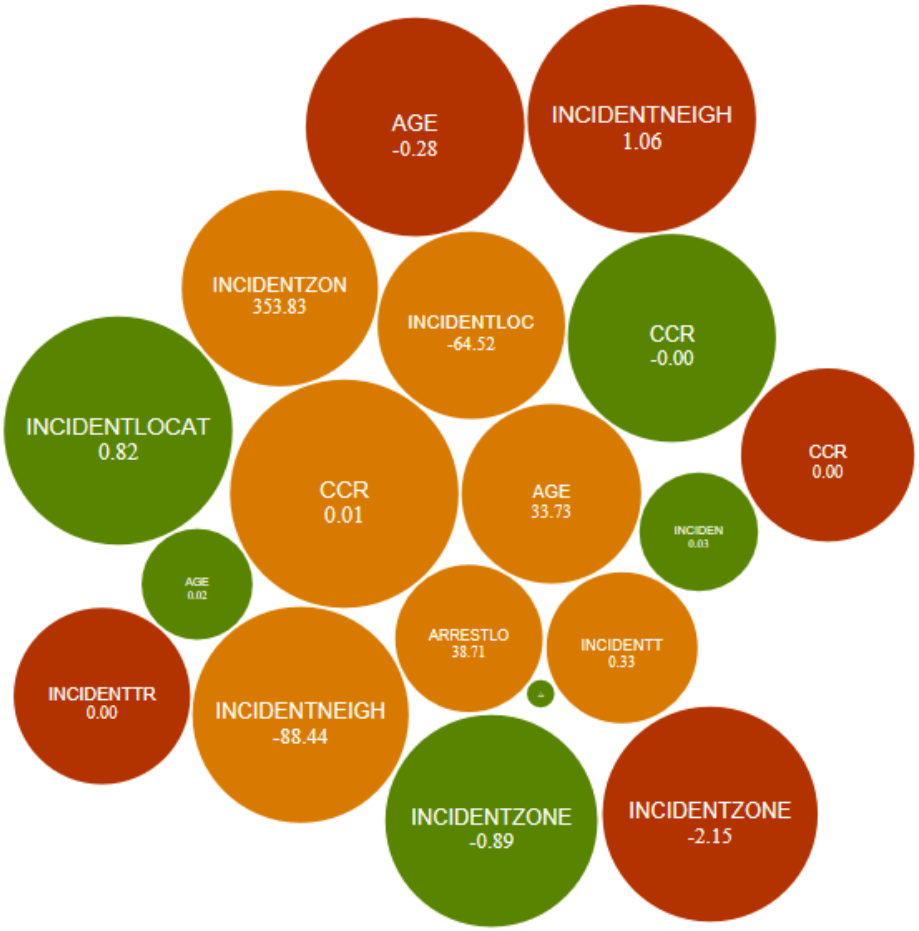
(2018)], ale zpětná interpretace závislostí mezi původními daty není uplně přímočará. Druhá metoda nemá dobré statistické odůvodnění, sloupce s malým rozptylem mohou být v regresi důležité. Proto byla nakonec použita třetí možnost. Snažila jsem se příliš neomezit prostor možností, ale zároveň proces dostatečně zrychlit. Vstupem je soubor sloupců po odstranění duplicit a hodně korelovaných proměnných. Libovolná proměnná je použita jako odpověď a zbylé proměnné jako prediktory. Nyní je třeba vybrat, které proměnné jsou užitečné pro předpověď odpovědi. Toto lze provést buď porovnáním p-hodnoty proměnných a celého modelu nebo postupnou selekcí (stepwise variable selection). P-hodnota nám říká, jaká je pravděpodobnost, že nalezená závislost je jen náhoda a sloupec vlastně k výsledku nepřispívá. Pro celý model pak p-hodnota určuje pravděpodobnost, že nalezená závislost je jen náhoda a data ve skutečnosti tyto vztahy neobsahují. Tato metoda je nyní použita, druhá není implementována, přestože její využití je připraveno i v uživatelském rozhraní. Pro druhou metodu lze začít s jednou proměnnou a přidávat postupně další, dokud nepřekročíme nějakou hranici[James a kol. (2018)], začít se všemi proměnými a odebírat nebo dělat zpětné a dopředné kroky. Pro logistickou regresi se často používá podobný postup pomocí křížové validace[James a kol. (2018)] - jsou postupně přidávány proměnné a model je vždy na 90% dat natrénován a na zbytku otestován. Tento postup je delší a náročnější na výpočet,jeho výsedky ale mohou být přesnější. V práci je použita p-value pro jednotlivé proměnné a pro celé modely a metody nad povolenou hranici věrohodnosti nejsou použity. Naopak, pokud je nalezena věrohodná závislost, pak v dalším kroku jsou zahrnuty jen proměnné, které nejsou součástí modelu z předchozího kroku, protože zavislost, které se účastní už byla odhalena.

Shrnutí: V nejhorším možném případě projdeme všechny možnosti, pokud žádný model nebude dostatečně věrohodný. V případě, že nějaký bude objeven, zmenší se prohledávaný prostor o použité proměnné. Tím se omezil počet prohledávaných modelů a zároveň většina důležitých závislostí bude odhalena.

1.4.2 Intepretace a vizualizace

Pro lepší interpretaci přínosu jednotlivých proměnných může sloužit koeficientů na z-normalizovaných datech[Wooldridge (2016)]. Potom velikost koeficinetu u proměnné určuje jeho relativní důležitost vůči ostatním. Navíc není potřeba pro toto dělat nový výpočet. Z už vypočtených koeficientů jsou ty normalizované jednoduše získat [Wooldridge (2016)].

Pro vizualizaci jsem zvolila bublinový graf bez os, kde velikost bublin určuje právě relativní příspěvek k výsledku a pohyb myši nad bublinu lze získat další informace o koeficientu - hodnotu koeficientu, normalizovaného koeficientu, p-value a další (viz Obrázek 1.2). Barevně jsou pak odlišeny jednotlivé modely (viz). Na první pohled je tedy vidět, jak důležitá proměnná pro regresi je a navíc lze zobrazit všechny potřebné informace na jednu obrazovku.



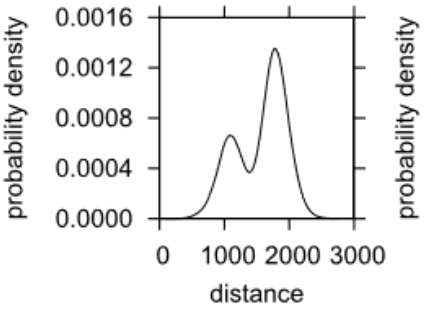
Obrázek 1.2: Vizualizace lineární regrese.

V grafu je vidět znázornění lineární regrese. Pro interpretaci má význam hodnota uvedená u kruhu a jeho vleikost, nikoliv umístění v prostoru.

1.5 Shluková analýza

Shluková analýza je další metodou,která může o datech mnoho prozradit. Narozdíl od regrese a dalších metod, které vyjadřují explicitně závislosti mezi sloupci, shluková analýza pouze ukazuje, zda se v datech nachází skupinky podobného charakteru. Lze použít hierarchickou analýzu nebo některou z k-means metod[James a kol. (2018)S]. Problémem všech těchto potupů je, že pro data s velikou dimenzí, čili data s velkým počtem sloupců, fungují tyto metriky špatně. Pro příliš velké dimenze (curse of diemnsionality) jsou všechny osti příliš velké[Yellamraju a Boutin (2018)]. Toto je dlouhodobý problém, na který bylo navrženo několik řešení, žádné se ale zatím nestalo hromadně používaným. V této práci je použit celkem nedávno publikovaný algoritmus založený na následujících myšlenkách.[Yellamraju a Boutin (2018)] Základní předpoklad pro celou funkčnost je, že pokud data obsahují nějaké skupiny, tak je možné je při vhodném promítnutí do 1D také dobře seprovat na dvě skupiny. Takovéto rozdělení v 1D je přitom velice rychlé a jednoduché. Samozřejmě, že najít to správné promítnutí může být náročné, ale v této me-

todě je využit Monte-Carlo princip, který situaci značně zjednoduší. Navíc je zde popsána i metoda pro měření separovatelnosti. Vygenerujeme náhodný vektor z intervalu [−1,1] z normálního rozdělení na který budeme projektovat. Pokud jsou v datech nějaké skupiny, tak graf rozdělení vzáleností mezi všemi dvojicemi bodů bude obsahovat dva vrcholy, protože data jsou rozdělena na dvě části v 1D (viz Obrázek 1.3). Na počátku je náhodně vygenerováno více vektorů (přesný



Obrázek 1.3: Rozdělení vzdáleností v separovatelných datech.

Graf převzat z Yellamraju a Boutin (2018).

počet je parametrem algoritmu, v mém programu je použito 100, dle doporučení v článku) a výsledek je zprůměrován. Pokud se ukáže, že data sledují normální rozdělení (které je tak náhodné, že žádné skupiny neobsahuje) nebo jsou výsledky ještě horší, v dělení se nepokračuje. Pro tyto účely je vytvořen další stejně velký dataset s čísly vybranými z náhodného rozdělení,vůči němuž se výsledky poměřují. Pokud dělení dopadne dobře, data sou rozdělena na dvě části a algoritmus pokračuje dále hierarchicky. Tato metoda se ukázala jako dostatečně rychlá a funkční.

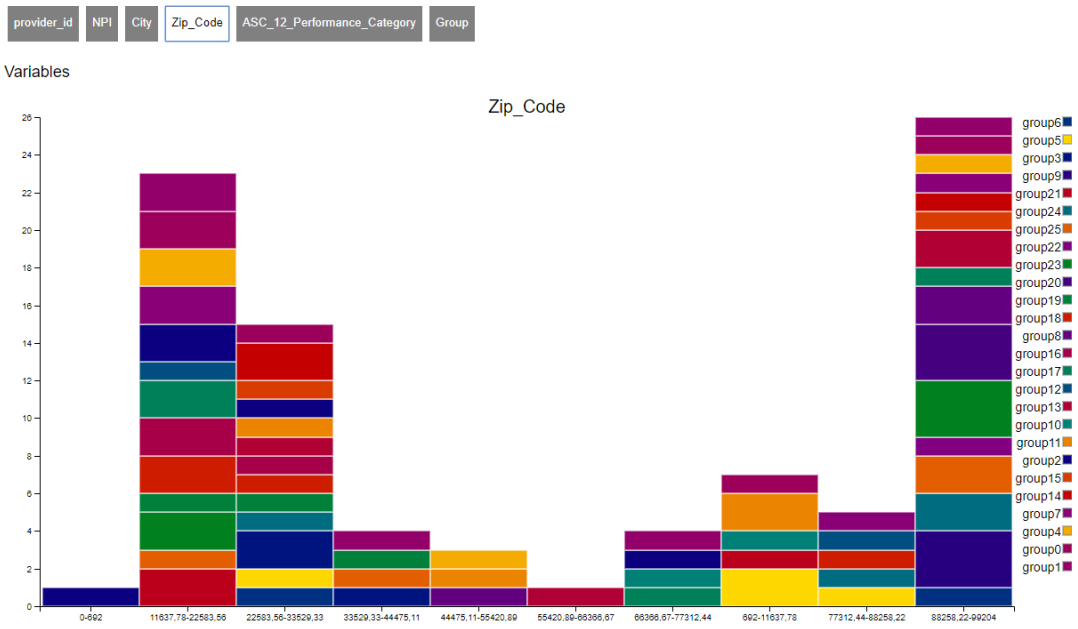
1.5.1 Vizualizace

Shluková analýza číselně poskytuje pouze rozdělení řádků do skupin, které ale nikterak necharakterizuje. To co je ale na skupinách zajímavé, je právě jejich charakteristika. Proto jsem se snažila zvolit zobrazení, z kterého by toto mohlo být vidět. Výsledek je reprezentován děleným sloupcovým grafem pro každou proměnnou (mezi grafy lze přepínat tlačítky nad grafem) (Obrázek 1.4). I spojitá data jsou zde rozdělena do intervalů, graf je tedy histogramem. Různé barvy napříč grafem označují jednotlivé skupiny. Z grafu je tedy patrné, jak je která skupina v rámci proměnné rozdělena. Pokud je rovnoměrně, tak to není proměnná, který by k odlišení této skupiny přispívala.

1.6 Charakterizace skupin

Následující úvahy vychází z potřeby nejak popsat skupiny nalezené ve shlukové analýze, případně najít kompaktní části dat, které možná v analýze nalezeny nebyly.

Se shlukovou analýzou, tak jak byla popsána výše, souvisí ještě jedna, možná



Obrázek 1.4: Zobrazení rozdělení do skupin.

dokonce zajímavější možnost rozdělení dat na skupiny. Ne vždy je nutné hledat skupinky, které jsou striktně oddělené, a lineární závislosti, které platí vždy. Jako jednoduchý příklad může sloužit následující příklad, který nevychází ze skutečných hodnot a slouží jen pro ilustraci. Mějme data vyjadřující počet vlastněných gumových kachniček v závislosti na věku, příjmech a dalších faktorech. Do jistého věku je počet kachniček lineárně závislý na věku dítěte, ale nesouvisí s jeho příjmem (žádný nemá). V jistém věku už tato závislost neplatí. Naopak vidíme souvislost se zájmem o programování. V pozdějším věku můžeme opět vidět souvislost mezi počtem dětí (pokud vlastnictví kachničky přiznáme dítěti i rodičům). Shluková analýza možná najde všechny tři zajímavé skupiny. Pro prodejce kachniček je ale zajímavá přesná závislost vlastněných kusů na dalších faktorech. Je vidět, že jednou z cest popsání skupinek může být hledání závislostí, ať už lienárních nebo složitějších, v nalezených skupinkách.

Trochu šířeji vzhledem k regresi lze říci, že se v datech mohou nálezat úseky, pro které nějaká závislost platí, ale nedá se říci, že by stejná nebo vůbec nějaká platila pro celý soubor. A právě takovéhle informace mohou být důležité, přesto je výše popsané metody neodhalí. Jendou cestou pro přispění k odhalení těchto případů je tedy analýza nalezených skupin. Další možností by mohlo být upravit algoritmus pro shlukovou analýzu tak, aby povoloval i menší vzdálenosti před popsáním skupin. Jinou možností, také založenou na použití volnějších kriterií, je akceptovat v lineární regresi i horší modely a pokusit se je zlepšit odebráním některých řádků. Možností je také vybrání nějakých skupin náhodně. Všechny tyto metody jsou však příliš náročné (kvůli počtu možností), takže by bylo třeba nalést vhodné zjednodušení. Zatím je jediný pokus o interpretaci vyjádřen grafickým znázorněním, které umožňuje posoudit rozložení proměnné napříč skupinami.

1.7 Shrnutí

V této kapitole bylo představeno, jakou analýzu dat by bylo možno provést. Některé možnosti byly vybrány a implementovány, jak bude popsáno podrobněji v následující kapitole. Některé detaily, regrese vyšších řádů a také popis skupim je cestou pro další rozšíření.

2. Implementační dokumentace

Nezajímavější rozhodnutí během práce se udála v oblasti použití statistických metod a je podrobně popsána v předchozí kapitole. Jak je z ní vidět, rozhodně nebyly využity všechny možnosti, jak z dat získat zajímavé informace,což je způsobeno rozsahem projektu i stupněm mé odbornosti. Každopádně zde vzniká potřeba navrhnout architekturu tak, aby byl program snadno rozšiřitelný.

2.1 Architektura

Na obrázku 2.1 je vidět, jaká byla původní představa o programu. Jediná zásadní změna v konečné implementaci se týká části hodnocení modelů. Ve sutečnosti jsou totiž metody aplikované sekvenčně(2.2). Z toho plyne, že nelze napřed vytvořit modely a pak je následně ohodnotit a protřídit, protože jednotlivé kroky na sebe navazují.

2.2 Faktická architektura

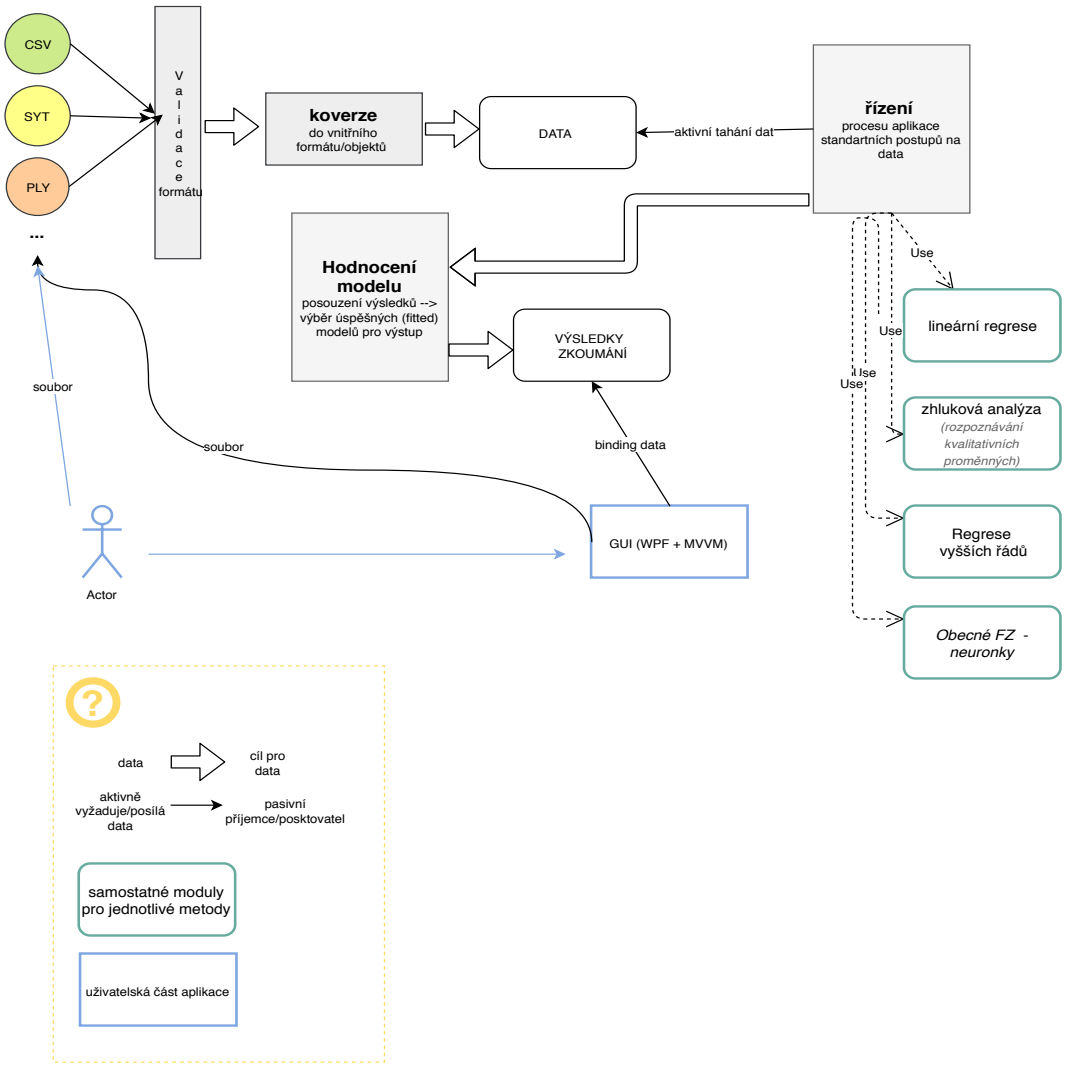
Sestavení se aktuálně skládá ze čtyř projektů: `Data, DependencyResolver, Entities, Marple`.

Projekt "Data"je zodpovědný za vstup/výstup dat do výpočetní části programu (nikoliv za interakci s uživatelem), Protože předpokládaný zdroj dat může být různý (např. textový soubor, databáze), je důležité, aby ostatní části programu nebyly závislé na původu a formátu dat. Našly zde uplatnění dva návrhové vzory: Vkládání závislostí (dependency injection) spolu s obráceným řízením (inversion of control) [Fowler (2004)]. Jak je vidět na diagramu 2.3 , stačí aby datová vrstva implementovala rozhraní IDataLayer a zbytek programu je na jeho skutečné implementaci nezávislý.

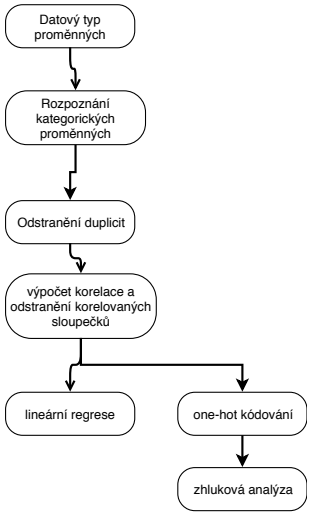
Projekt "Entities"obsahuje všechny datonosné struktury a třídy zodpovědné za statistické metody, jak to odpovídá architektuře z obrázku 2.1. Třídy využívají knihovních funkcí Accor.Net, R a dodávají potřebnou další logiku (např pro průběh lineární regrese).

DependencyResolver je zde pomocná třída pro vkládání závislostí. Použila jsem knihovnu Ninject [Nin], jejíž zapojení do projektu se mi zdálo relativně jednoduché. Má ale sve nevýhody právě v podobě nutnosti dalšího projektu, který vidí "všechno"a spravuje propojení [Swan]. Další zvažovanou variantou byl Castle Windsor [Win], jehož použití se mi ale zdálo složitější a pro účely projektu takového rozsahu se Ninject jevil jako dostačující.

Nejprve bych ráda stručně shrnula průběh výpočtu[Obrázek 2.2] a poté přidala pár poznámek k některým částem.
Část "Řízení"z obrázku 2.1 je reprezentována třídou `ProgramController.cs`, která má za úkol volání datové vrstvy při načítání, řízení jednotlivých kroků



Obrázek 2.1: Abstraktní popis architektury programu.



Obrázek 2.2: Změny datového souboru v průběhu programu

výpočtu, ukládání uživatelských nastavení a výpis reportu.

1. Načtení dat do DataTable struktury.
2. Uložení uživatelských nastavení
3. Zjištění předpokládaného datového typu sloupce
4. Hledání možných výčtových proměnných
5. Odstranění duplicit I
6. Výpočet PCA a korelace
7. Odstranění duplicit II
8. Lineární regrese
9. Clustering
10. Html výtup

2.2.1 Formát dat

Předpokládaný formát dat je csv a podobné formáty pro data uložená v tabulkách. To znamená, že program umožňuje změnu oddělovače záznamů, změnu uvozovače řetězců a data mohou a nemusí obsahovat řádek záhlaví. Přípona souboru i název mohou být libovolné. Původní představa byla, že by program uměl zpracovat i 3D data ve formátu `.ply`, což se zatím nepodařilo implementovat. Z hlediska propojení je ovšem práce snadná. Stačí připojit do `Data.Formats` novou třídu (podobně jako `CSV.cs`), přidat toto do výčtu formátů(`AvailableFormats`) a v příslušném switchi přidat novou možnost. Pokud jde o výpočetní část, nejspíše by musela být trochu jiná, ale podrobné zkoumání tohoto nebylo předmětem této práce. Data mohou obsahovat celá i reálná čísla a textové řetězce.

2.2.2 Uživatelská nastavení

Metaparametry i vlastnosti vstupního souboru jsou ukládány. Využila jsem proto standartní `Properties.Settings` soubor projektu Marple.

2.2.3 Vnitřní reprezentace dat a statistická knihovna

Prvotní plán reprezentace dat souvisí s původním plánem využít knihovnu R.Net pro propojení C# a R. R má bohatou škálu metod a balíčků pro metody strojového učení a statistiku a jeho možnosti by plně pokrývaly potřeby této práce. Dalším požadavkem ovšem bylo relativně přívětivé uživatelské rozhraní, což R nenabízí. Další cestou by byl python, který je k danému druhu práce podobně vhodný a dispnuje i grafickými knihovnami [Pyt], výběr byl ovšem závislý i na mých znalostech a schopnostech, takže jsem python zavrhla. Snažila jsem implementovat vlastní strukturu `DataFrame` (a stejnojmenné rozhraní), která by poskytovala všechny požadované vlastnosti, zejména: uložení dat po jendotlivých řádcích,přístup k jednotlivým sloupečkům jako k celku i v rámci řady, konverze do

číselného vícerozměrného pole a zpět, základní funkce jako počet sloupců a řádek, dále název a jméno sloupce či různý datový typ. Pro tento účel byly vytvořeny i další generické datové struktury na uchování vektorů představujích jednotlivé sloupečky. Knihovna R.Net [RNe] mi ale přišla celkem spoře okomentovaná, navíc jsem neustále narážela na její nepřenositelnost. Fungovala jen s učitou verzí R a ABI(cílovou architekturou procesoru), ani tak se mi ale nepodařilo ji zprovoznit na různých strojích, čímž se pro mne stala nepoužitelnou. Nakonec jsem se proto rozhodla pro knihovnu Accord.Net [Acc], která nevyžaduje žádné speciální podmínky okolního prostředí. Její metody jsou většinou kompatibilní právě s `DataTable` nebo číselnou maticí, pro jejich konverze navíc poskytují rozšiřující metody. Nutno poznamenat, že implementace PCA v Accord.Net se liší d té v R v několika zásadních věcech (například vektory nejsou normalizované) a proto pro tuto část nakonec využívám R skript (FAMD.R) za použití knihovny factoMineR [de Souza (2012)] [RFA] [Fac (2018)], který je volán jako nový proces z hlavního programu a jeho textový výstup je pak naparsován. K tomuto byla převzata celá třída RRunner[RRu].

Rozhodnutí reprezentovat data právě takto má některé nepříjemné dopady. Z obou variant plyne, že data se musí vejít do paměti, nelze tedy použít na příliš velká data(například na platformě Hadoop), což je dalším možným rozšířením programu. Samotná `DataTable` spotřebovává dost paměti a Accord.Net například v objektu `Codification` pamětí také nešetří, takže místo převádění dat na one-hot kódování se stalo kritickou částí programu (`OutOfMemoryException`) a bylo nutné kódovanou tabulku ukládat jen jako matici. Program si neporadí s příliš velkými soubory (výše zmíněná výjimka nastala na souboru Pitsburg Arrest data s cca 21 000 řádky a 20 sloupečky, pro RAM o velikosti 8GB), na druhou stranu jeho rychlost je celkem uspokojivá. Při provolávání R v novém procesu vzniká zpoždění, které je ale konstatní pro všechny velikosti dat a zanedbatelné (asi 3s).

2.3 Výstup

Výstup je vygenerován až nakonec zpracování, a to ve formátu HTML 5, uživatel tedy nemá jinou možnost zasažení do procesu, než jeho kompletní opakování s jinými parametry. Zde je prostor pro rozšíření: výslekdy výpočtu, které nebyly ovlivněny změnou, by mohly být ukládány a využity znova. Jako úzké hrdlo v tomto programu se ale ukazuje spíše paměť než čas a provedení tohoto kroku vyžaduje netriviální úsilí, takže jsem zatím tento krok považovala za nevýhodný.Ve chvíli, kdyby množtví spotřebované paměti nebyl problém a některá část výpočtu se ukázala příliš dlouhá, dávalo by toto smysl.

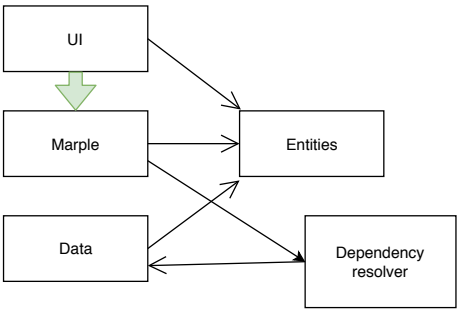
Na začátku nebylo zcela jasné, jaký formát výstupu je preferovaný. Pokud má program sloužit jen jako nápověda pro datového analytika, byl by plně postačující textový soubor s číselnými výsledky. Například u shlukové analýzy ale není dost jasné, jak přehledně její výsledky reprezentovat a tabulka s doplněnými čísly skupin je pro lidské oko zrovna tak nepřehledná jako původní data. Hlavní přínost této práce ale vidím v zjednodušení nepřehledné velké tabulky dat tak, aby bylo možné snadno a rychle pochopit, jaká ta data jsou. Takže bylo třeba použít grafické vyjádření s většími možnostmi. Nabízelo se zobrazení v rámci uživatel-

ského rozhraní, `pdf` soubor nebo `html`. Nevýhodou pdf souboru je, že nemůže být interaktivní,což je pro vizualizaci dat výrazná nevýhoda oproti dvěma zbývajícím řešením. Navíc pokud předpokládám, že report z našeho programu někdo použije pro prezentaci dat například na meetingu, tak si dovedu lépe předtavit, že ukáže polupracovníkům html stránku na tabletu, než nějaký složitější výsledek v programu, který bude náročnější na ovládání. Další výhodou je, že takto jdou výsedky poslat třeba elektronickou poštu.

Pro implementaci interaktivnějšího reportu s lepšímožností zobrazení požadovaných dat jsem použila HTML, kaskádové styly (CSS), JavaScript. Speciálně, k vykreslování grafů jsem použila knihovnu d3js[D3], kde bylo možno čerpat z bohatého souboru příkladů a ukázek grafů. Programátorsky hezčí řešení je rozdělení výsledné stránky na kaskádové styly, skripty a html kód. Původně to takto bylo implementováno, ale protože předpokládám, že uživatel bude chtít report posílat elektornickou poštou nebo přesouvat mezi různými umístěními, je jednodušší, aby měl jeden soubor, kde nehrozí nefunkčnost kvůli chybějícím závilostem.

2.4 Uživatelské rozhraní

Uživatelské rozhraní má sloužit pro příjemnější komunikaci s programem (v porovnání s konzolovou aplikací). Snažila jsem se o maximální přehlednost a jednoduchost. Rozhraní je součástí spouštěného projektu "Marple"a je implementováno za použití WPF. Z hlavního okna se volá `ProgramController`, který řídí další běh aplikace. Pro design rozhraní jsem použila knihovnu Mahapps[mah], která vytváří vzhled odpovídající designu Windows OS(tzv. Metro). Navíc obsahuje implementaci některých užitečných věcí, jako například ProcessRing, UpDown(který WPF narozdíl od WinForms neobshauje) a knihovnu pro ikony.



Obrázek 2.3: Obrácené řízení.

3. Uživatelská dokumentace

V této kapitole bude předvedena práce s programem. Jak bylo uvedeno výše, záměrem práce je snadná prvotní analýza neznámých dat, která může být použita i jako konečný výstup bez interpretace odborníkem (např. datovým analytikem). Z tohoto důvodu jsem se snažila o maximálně uživatelsky přívětivé rozhraní s jednoduchým ovládáním.

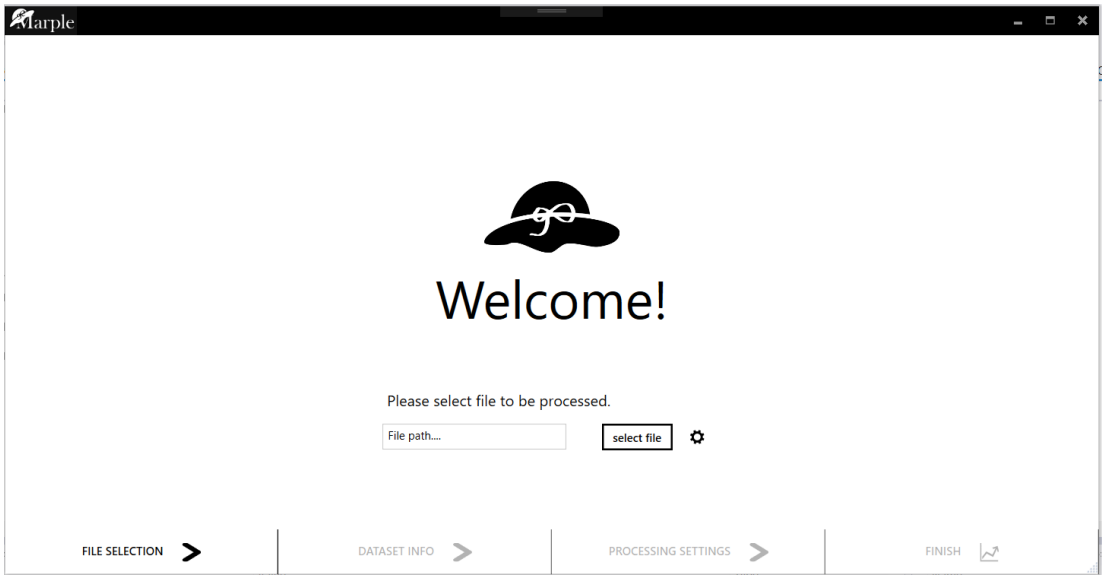
Pro přehled o rychlosti programu může sloužit následující výsledek. Pro dataset s 5122 řádky a 53 sloupci trvala analýza 18.61 s, je ale třeba vzít v úvahu, že neúplné řádky byly vyřazeny a bylo jich zde mnoho [med (2018), Ambulatory surgical measures - Facility].

Uživatelské rozhraní má celkem 4 obrazovky - nahrání souboru, informace o jednotlivých sloupečcích, metaparametry statistických metod a závěrečná obrazovka pro spuštění.

3.1 Zpracování bez dodatečných informací

3.1.1 Nahrání souboru

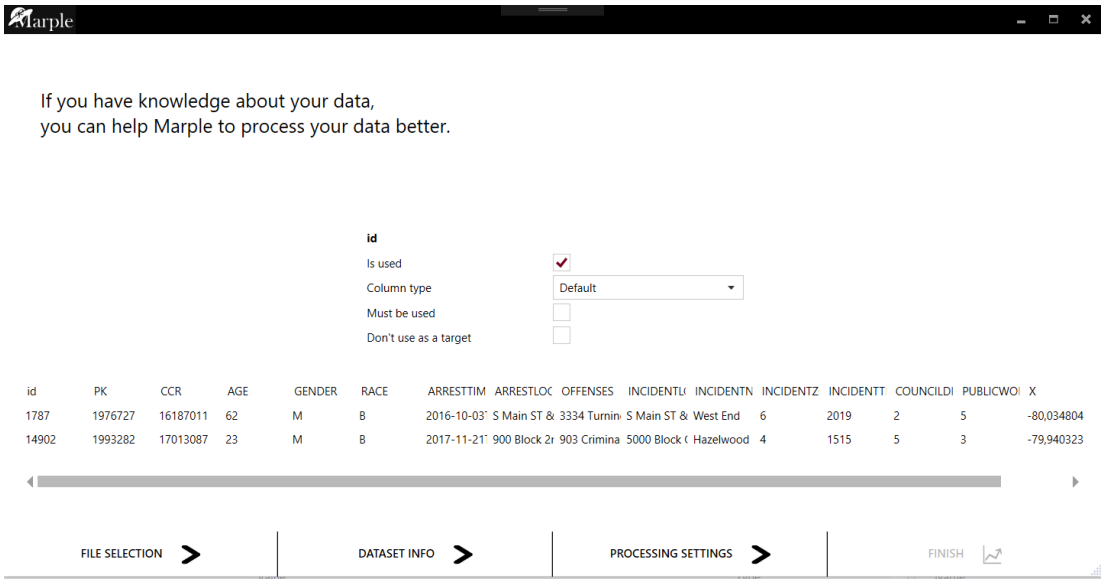
Pro nahrání souboru je nuto použít úvodní obrazovku, dříve nejde pokračovat v práci s programem (viz Obrázek 3.1). Nahrání souboru - lze zadat cestu nebo vybrat z nabídky po stisku tlačítka "Select file".



Obrázek 3.1: Úvodní obrazovka pro nahrání souboru

Poté je možné přejít na další obrazovku. Nastavení pro jednotlivé sloupce zatím vynecháme, ale je dobré si povšimnout tabulky v dolní části obrazovky. Jsou zde vidět hlavičky sloupečků (pokud jsou přitomny) a první dva řádky dat. Pokud data nevypadají podle Vašich představ - například některé sloupečky jsou sloučené do jednoho, není vidět hlavička, i když data ji obsahují apod. - možnou

příčinou je, že byl očekáván jiný formát dat. Na úvodní obrazovce lze po stisku tlačítka nastavení pozměnit parametry pro nahrávání csv souboru.



Obrázek 3.2: Obrazovka informací o souboru

3.1.2 Dokončení

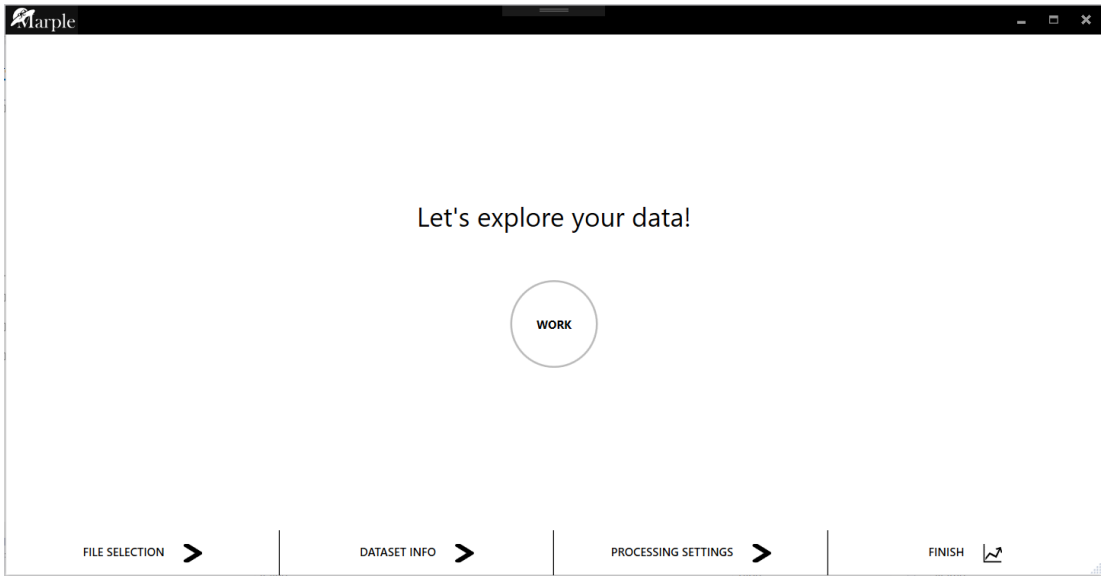
Pokud jsou data v pořádku, je možno pokračovat postupně až na obrazovku s tlačítkem pro spuštění (Obrázek 3.3). Až budou ukončeny všechny výpočty, tak bude z této obrazovky možno zobrazit a uložit výsledný report. Během výpočtu proto, prosím, nezavírejte program. Výpčet je možno kdykoliv ukončit stiskem tlačítka "STOP". To může být vhodné, pokud progrma běží příliš dlouho, ale zejména v situaci, kdy je třeba opravit některé dříve zadané informace (např. nahrát jiný soubor, poupravit jinak metaparametry). Veškeré výpočty po opětovném spuštění budou provedeny znova.

3.2 Dodatečné informace o datech

Je celkem zbytečné, aby se program snažil objevit věci, které Vám jsou očividné nebo nejsou zajímavé. Také je možné, že algoritmus na některou možnou kombinaci dat bude reagovat jinak, než byste si přáli. Defaultní hodnoty metaparametrů pro linearní regresi, shlukovou analýzu a další jsou nastaveny pokud možno rozumně, ale v různých situacích může být vhodné je změnit. Pro první případ lze na druhé obrazovce pro každý sloupec zvlášť nastavit některé věci (Obrázek 3.2). Pro ten druhý slouží třetí obrazovka (Obrázek 3.4), kde lze měnit metaparametry.

3.2.1 Nastavení pro proměnné

- Is used - defualt je ano, je ale možné vyřadit některé sloupečky ze zpracování. Jedná se o sloupce, ve kterých nechcete hledat žádné informace, nebo



Obrázek 3.3: Spuštění programu a zobrazení výsledků

tam nejsou. Příkladem může být třeba id, i když si lze představit, že například vyšší id indikující pozdější záznamy může být zajímavou informací v clusteringu.

- Column type - Lze vybrat, zda se jedná o diskrétní (celočíslené) hodnoty, reálná čísla nebo výčet hodnot. Není třeba explicitně nastavovat u všech sloupců, ačkoliv třeba rozdíl mezi číselnou a slovní hodnotou je zjevný. Program toto patrně odhalí dříve, než byste to vyplnili. Význam to má v případě, že je některá proměnná výčet, ale hodnoty v ní jsou číselné.

- Must be used - Jednou z věcí, která se v datech hledá, jsou duplicitní sloupečky. Je tedy možné, že pokud jsou sloupečky hodně korelované, bude jeden z nich vyřazen. Pokud chcete tomuto zabránit, jelikož ve Vašich datech samozřejmě žádné duplicity nejsou, nebo už se stalo, že byl sloupeček vyřazen na základě korelace s jiným a přitom nemáte dojem, že právě toto jste chtěli, je možno zakázat smazání sloupečku.

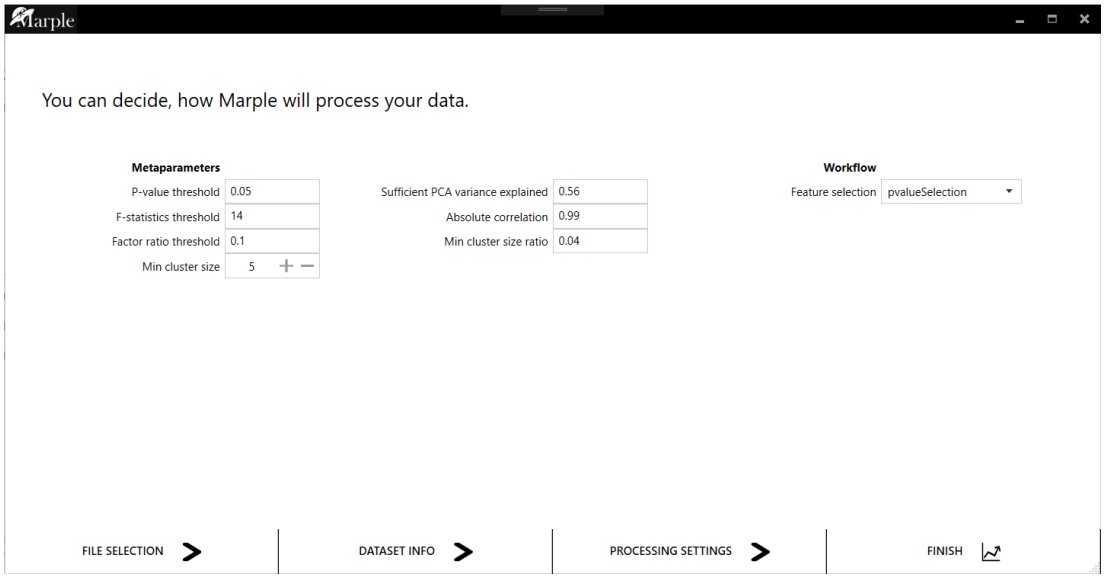
- Don't use as a target - Nebude v lineární regresi použit jako předpovídaná hodnota. Pokud o datech není nic známo, program prozkoumá všechny možné kombinace. Zajímá-li vás však jedna konkrétní proměnná, je možné omezit prohledávaný prostor možností.

3.2.2 Metaparametry

- p-value threshold: tato hodnota je použita v lineární regresi.Na jejím základě se vybírají platné modely i jednotlivé proměnné v regresi.

- F-statistic threshold: Aktuálně není použita, je možno ji použít pro filtrování modelů v lineární regresi.

- Factor ratio threshold: Při rozhodování, zda je proměnná výčet, se spočte



Obrázek 3.4: Nastavení metaparametrů

následující:

$$ratio = \frac{pocet\_unikatnich\_hodnot\_v\_sloupci}{pocet\_radku}$$

Pak se tato hodnota porovná s nastavovanou hranicí, a pokud je hodnota menší, tak se s proměnnou zachází jako s kategorickou.

- Min cluster size a min cluster size ratio: Pokud algoritmus sám neskončí dříe, tak kriteriem k zastavení hierarchického clusteringu je

$$Min(min\_cluster\_size, min\_cluster\_ratio * pocet\_radku).$$

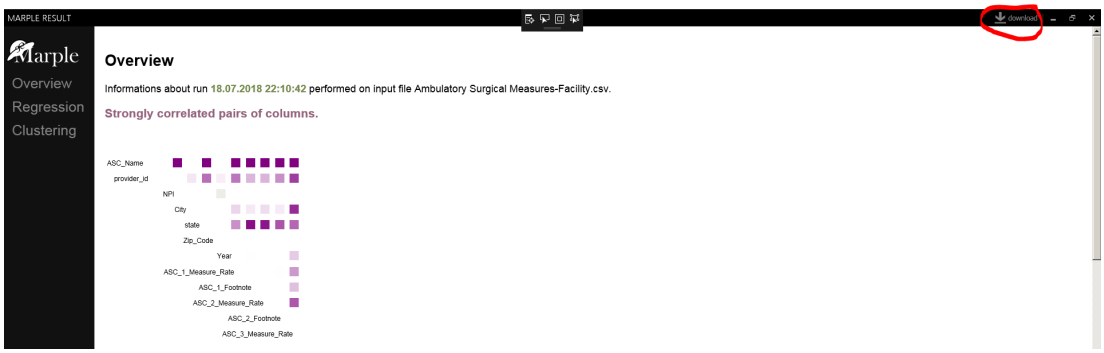
- Jedním z informací v reportu je předpokládaná dimenzionalita dat, která se spočítá s využitím Principal component analysis (dále PCA). Pro tuto část se nevezmou všechny komponenty, ale jen postačující pro splnění tohoto nastavovaného kriteria.

- Absolute correlation nastavuje hranici, nad kterou bude jeden z korelovaných sloupců vyřazen. Nedoporučuji nastavovat na 1, protože díky zaokrouhlovacím chybám často korelace stejných sloupců vyjde nepřesně.

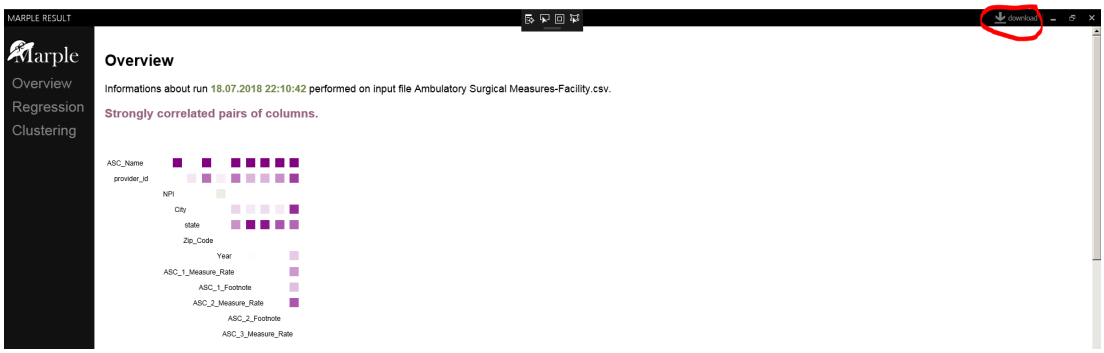
3.3 Uložení reportu

Po dokončení výpočtů a pokud se nevysktla chyba, je možno zobrazit report stiskem tlačítka "VIEW"(Obrázek 3.6).

Ve výsledném reportu je možno v pravo nahoře stikem "download"vybrat umístění a uložit soubor pro pozdější použití í(Obrázek 3.5).



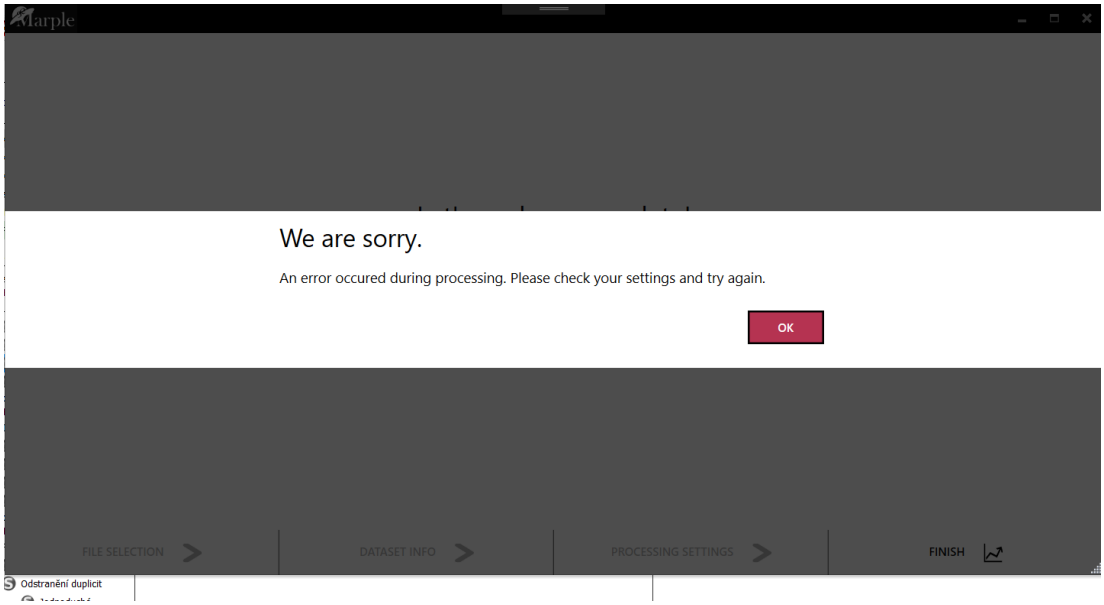
Obrázek 3.5: Zobrazení a stažení reportu.



Obrázek 3.6: Spuštění programu a zobrazení výsledků

3.4 Odchycení chyb

Pokud se během výpočtu objeví chyba, zobrazí se dialog s upozorněním. Po jeho odsouhlasení je možno zkusit výpočet spolu (Obrázek 3.7). Nicméně výpočet je v tomto směru deterministický, takže nemá smysl opakovat výpočet bez změny vstupů.



Obrázek 3.7: Chybové hlášení

Závěr

Cílem práce bylo vytvořit nástroj pro rychlou a snadnou úvodní analýzu dat. Vytvořený program lze přiměřeně jednoduše ovládat a poskytuje o datovém souboru základní informace uživatelsky přívětivou formou, což bylo mým cílem. Program odstraňuje duplicity, poskytuje informace o tom, které sloupce jsou korelované, jaká je předpokládaná dimenzionalita dat, hledá mezi proměnnými lineární modely a skupinky podobných dat (shluková analýza). Výsledky jsou prezentovány jako html soubor s animovanými grafy. Na lineární regresi byl použit algoritmus, který nevynucuje prohledání celého stavového prostoru, zatímco u shlukové analýzy byl problém dimenzionality vyřešen promítáním do prostoru dimenze 1, kde je separace triviální. Program je možno použít bez jakékoliv nápovědy o datech, ale i využít znalostí, které o souboru máme pro dosažení přesnějších výsledků.

Z první kapitoly je ovšem vidět, že možnosti prozkoumání dat, které by mohly být realizovány jsou větší, než jaké byly implementovány. Hlavní možnosti rozšíření vidím v následujích směrech: **metody strojového učení**. Z nich by to mohla být zejména regrese vyšších řádů, krokový algoritmus pro výběr proměnných v lineární regresi, doplnění algoritmu pro korelaci kvalitativních dat, který by zohledňoval velikost dat a snaha o hledání složitějších závislostí či pokus s hlubokými neuronovými sítěmi. Druhou oblastí, která by mohla být snažší na implementaci se stávajícím výpočetním základem je **vizualizace dat**. Bylo by možno zobrazit i příspěvky jednotlivých proměnných, jak byly spočítany v PCA, tabulku na lineární regresi učinti interaktivní místo méně přehledného seznamu, graficky znázornit základní metriky proměných (průměr, rozptyl, kvartily) a pokusit se graficky znázornit i další rozšiřující data. Ještě jednou nezmíněnou oblastí dat, která vůbec nebyla brána v úvahu, jsou data s časovými značkami, ke kterýmje třeba přistupovat jinak, provádět sezónní očištění atp [Wooldridge (2016)]. V této oblasti samozřejmě také leží možnosti rozšíření.

Projekt s podobným záměrem vznikl a stále je rozšiřován týmem z Cambridge university a MIT. Automatic statistician (dále AS)[AS] ovšem vznikal ve větším týmu a podstatně delší dobu. Přesto si myslím, že tato rpáce přináší něco nového. Vizualizace dat staticky, jak tomu je v AS podle mého nemůže zcela využít současné technologické možnosti. Jedná se o pdf s strojově genrovaným textem, který je k nerozeznání od lidské práce. Ale i kdybychom jako jediný pohyblivý prvek použili popisky u sloupečků sloupcového grafu, stále to způsobí, že je možné ke každému sloupci snadno připojit sadu rozličných informací tak, aby nepřekáželi celkovému pohledu na graf a přitom byly snadno dostupné bez náročného hledání. Regrese vyšších řádů nebo další metoda pro výběr proměnných v lineární regresi poskytnou přesnější pohled na data. Pravděpodobně nebude tak detailní jako v AS, ale výhodou Marplu může být, že nabízí mnohem přehlednější znázornění informací. Samozřejmě v současném stavu se nemůže s profesionálními programi měřit, ale ukazuje trochu odlišnou cestu, která v době tabletů a interaktivních tabulí jako běžných pomocnků ve firmách nabízí kvalitnější možnosti prezentace dat, než tištěný přehled.

Původní záměr počítal i s prozkoumáním možností aplikace neuronových sítí (neural networks, dále NN), které jsou v současné době velmi populární. Pokud bychom chtěli aplikovat NN na hledání závislostí, z jistého úhlu pohledu je tato úloha ideální. Je možno vygenerovat libovolně velké datasety, u kerých budou informace známé. Problém NN je, že fungují jako černá skřínka a o tom, jak se naučili predikovat určitou věc, moc nevíme. Respektive můžeme vědět, jenomže síť bývá velká a komplikovaná, takže z její vnitřní struktury poznáme souvislosti stejně těžko. Pro aplikaci NN na nastavení metaparametrů zase stojíme před jinou obtíží: jak mají vypadat trénovací data? Bylo by nutno vzít opravdu velké množství datasetů, nejlépe s reálnými daty, využít expertní znalosti datového odborníka a pokusit se neuronku naučit "jak to měla udělat nejlépe". To je ale příliš složité a hodno samostatného zkoumání a odbornosti přesahující rozsah této práce.

Máme tedy zatím malý, ale snadno rozšiřitený program pro prvotní analýzu dat, který může ušetřit několik hodin práce datovému analytikovi a poskytnout základní přehled o datech komukoliv jinému a to názornou grafickou formou, který bude možno dále rozšiřovat.

Seznam použité literatury

Automatic statistician. URL https://automaticstatistician.com/index/.

Hlavní strana projektu accord.net. URL http://accord-framework.net/.

Hlavní stránka d3.js. URL https://d3js.org/.

Domácí stránka knihovny ninject. URL http://www.ninject.org/.

Sackexchange: Using pca for feature selection. URL https://stats.stackexchange.com/questions/27300/using-principal-component-analysis-pca-for-feature-selection.

Python wiki: Gui programming in python. URL https://wiki.python.org/moin/GuiProgramming.

Famd dokumentace. URL https://www.rdocumentation.org/packages/FactoMineR/versions/1.41/topics/FAMD.

Hlavní strana projektu r.net. URL https://archive.codeplex.com/?p=rdotnet.

Stackoverflow: R from csharp. URL https://stackoverflow.com/questions/4485943/executing-r-script-programmatically.

Hlavní stránka projektu castle windsor. URL http://www.castleproject.org/projects/windsor/.

Chi-square test. URL https://en.wikipedia.org/wiki/Chi-squared_test.

Wikipedia: Factor analysis of mixed data. URL https://en.wikipedia.org/wiki/Factor_analysis_of_mixed_data.

Hlavní stránka projektu mahapps. URL https://mahapps.com/.

(2018). Factominer dokumentace. URL https://www.rdocumentation.org/packages/FactoMineR/versions/1.41/topics/FAMD.

(2018). Pittsburgh police arrest data. URL https://catalog.data.gov/dataset/pittsburgh-police-arrest-data.

(2018). Hospital compare datasets. URL https://data.medicare.gov/data/hospital-compare#.

de Souza, C. R. (2012). A tutorial on principal component analysis with the accord.net framework. *CoRR*, **abs/1210.7463**. URL http://arxiv.org/abs/1210.7463.

Fowler, M. (2004). Inversion of control containers and the dependency injection pattern. URL https://martinfowler.com/articles/injection.html.

Goodfellow, I., Bengio, Y. a Courville, A. (2016). *Deep Learning*. Adaptive computation and machine learning. MIT Press. ISBN 9780262035613. URL https://books.google.cz/books?id=Np9SDQAAQBAJ.

James, G., Witten, D., Hastie, T. a Tibshirani, R. (2018). *An Introduction to Statistical Learning with Applications in R*. 8th corrected printing. Springer Science+Business Media, New York. ISBN 978-1-4614-7137-0.

lumenlearning. Descriptive or inferential statistics? URL https://courses.lumenlearning.com/boundless-statistics/chapter/which-test/.

Marr, B. (2018). How much data do we create every day? URL https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-#53e51d3560ba.

Smith, L. (2002). A tutorial on principal components analysis. *Journal of the Royal Statistical Society, Series B*.

Swan, J. Using ninject to produce a loosely-coupled modular wpf application. URL https://www.codeproject.com/Articles/812379/Using-Ninject-to-produce-a-loosely-coupled-modular.

Wooldridge, J. (2016). *Introductory Econometrics: A Modern Approach*. Cengage Learning Asia Pte Limited. ISBN 9789814732031. URL https://books.google.cz/books?id=p_tIAQAACAAJ.

Yellamraju, T. a Boutin, M. (2018). Clusterability and clustering of images and other real high-dimensional data. *IEEE Transactions on Image Processing*, pages 1927–1938. ISSN 1057-7149. doi: 10.1109/TIP.2017.2789327.

Seznam obrázků

Seznam tabulek

A. Přílohy

A.1 Popis přiložených souborů

- Zdrojové soubory k nalezení ve složce `Sources`. Program lze kvůli technickým potížím spustit ze souboru `Sources\Marple\bin\debug\Marple.exe`
- Dále je přiložen vzorový report na subsetu Arrest dat [arr (2018)]