UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA

DIPLOMOVÁ PRÁCE

Anonymizace dat

Katedra matematické analýzy a aplikací matematiky
Vedoucí bakalářské práce: **Mgr. Iveta Bebčáková, Ph.D.**
Vypracoval(a): **Bc. Zuzana Mikolašová**
Studijní program: B1103 Aplikovaná matematika
Studijní obor Aplikace matematiky v ekomomii
Forma studia: prezenční
Rok odevzdání: 2022

BIBLIOGRAFICKÁ IDENTIFIKACE

Autor: Bc. Zuzana Mikolašová

Název práce: Anonymizace dat

Typ práce: Diplomová práce

Pracoviště: Katedra matematické analýzy a aplikací matematiky

Vedoucí práce: Mgr. Iveta Bebčáková, Ph.D.

Rok obhajoby práce: 2022

Abstrakt: Anonymizace dat je odebrání informací, které mohou vést k identifikaci konkrétní osoby. S anonymizovanými daty pak lze volněji nakládat, protože nejsou regulovány pravidly zákona o ochraně osobních údajů. Cílem diplomové práce je tedy naprogramovat nástroj pro anonymizaci dat pomocí programovacího jazyka Python.

Klíčová slova: anonymizace dat, ochrana osobních údajů, GDPR, anonymizační techniky, Python

Počet stran: 66

Počet příloh: 1

Jazyk: český

BIBLIOGRAPHICAL IDENTIFICATION

Author: Bc. Zuzana Mikolašová

Title: Data anonymization

Type of thesis: Master's

Department: Department of Mathematical Analysis and Application of Mathematics

Supervisor: Mgr. Iveta Bebčáková, Ph.D.

The year of presentation: 2022

Abstract: Data anonymization is the process of romoving information that can lead to identification of specific person. Anonymised data can be used more freely as these are no longer under the regulation specified by the Data Protection Act. Therefore, the aim of this thesis is to program a tool for data anonymization using the Python programming language.

Key words: data anonymization, data protection, GDPR, anonymization techniques, Python

Number of pages: 66

Number of appendices: 1

Language: Czech

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracovala samostatně pod vedením paní Mgr. Ivety Bebčákové, Ph.D. a všechny použité zdroje jsem uvedla v seznamu literatury.

V Olomouci dne ....

podpis

Obsah

Poděkování

Ráda bych vyjádřila své poděkování paní Mgr. Ivetě Bebčákové, Ph.D. za odborné vedení a vstřícný přístup. Zároveň bych ráda poděkovala Ing. Lukáši Novákovi, Ph.D. za poskytnutí odborných rad při konzultacích k praktické části této diplomové práce. Poděkování patří také rodině a přátelům za trpělivost a podporu po celou dobu studia.

Úvod

Tématem diplomové práce je Anonymizace dat. V současné době jsou neustále zvyšovány požadavky na ochranu osobních údajů a zároveň roste počet příležitostí, kdy mohou být osobní údaje z databází, dokumentů, webových stránek neoprávněnými osobami sbírány a posléze zneužity. Proto je třeba předcházet těmto situacím a vytvářet procesy a nástroje, které osobní údaje ochrání před neoprávněným přístupem. Anonymizace dat, která je stále předmětem intenzivních výzkumů, je jedním z nástrojů ochrany osobních údajů.

Tato diplomová práce vznikla ve spolupráci se společností Pricewaterhouse-Coopers Audit, s.r.o., která poskytuje auditorské, daňové a poradenské služby klientům.

Předpokladem realizace této diplomové práce byla v prvé řadě rešerše v oblasti zákonných požadavků na ochranu osobních údajů. Hlavním cílem práce bylo naprogramování vlastního nástroje pro anonymizaci dat s využitím jak znalostí nabytých výše uvedenou rešerší, tak samostudiem programovacího jazyka Python.

Diplomová práce je rozdělena na část teoretickou a část praktickou. Teoretická část obsahuje dvě kapitoly a praktická kapitolu jednu.

V rámci teoretické části je v první kapitole objasněno několik základních pojmů a především souvislostí s ochranou osobních údajů včetně zásad pro zpracování osobních údajů. Druhá kapitola se věnuje anonymizaci obecně. Hlavním důvodem anonymizace je bezpečnost, a proto je jí také věnována část práce. Důležitou součástí druhé kapitoly jsou pak jednotlivé základní anonymizační techniky, jejich obecný popis, včetně jejich nedostatků. Závěrem druhé kapitoly je

pak přehled jednotlivých metod a doporučení, jak k celému procesu anonymizace nejlépe přistupovat.

Praktická část se věnuje Anonymizéru, což je onen zmiňovaný vlastní nástroj pro anonymizaci dat. Hned v úvodu třetí kapitoly jsou objasněny požadavky na Anonymizér, které vzešly ze společné konzultace s již zmíněnou společností PricewaterhouseCoopers Audit, s.r.o.. Celá třetí kapitola je relativně obsáhlá a snaží se popsat Anonymizér jak ze strany uživatelského použití, tak umožňuje nahlédnout do pozadí aplikace v podobě ukázek a popisů jednotlivých algoritmů.

Část I

Teoretická část

Kapitola 1

Ochrana osobních údajů

Osobní údaje vyžadují vyšší stupeň ochrany, protože jejich únik může znamenat pro správce dat mnohem větší ztráty nebo právní postihy, než je tomu u ostatních dat.

Cílem první kapitoly je nahlédnutí do problematiky ochrany osobních údajů a především vysvětlení klíčových pojmů pro plynulejší pochopení této práce, především metodiky anonymizačních technik.

Listina základních práv Evropské unie říká, že:

„1. Každý má právo na ochranu osobních údajů, které se ho týkají.
2. Tyto údaje musí být zpracovány korektně, k přesně stanoveným účelům a na základě souhlasu dotčené osoby nebo na základě jiného oprávněného důvodu stanoveného zákonem. Každý má právo na přístup k údajům, které o něm byly shromážděny, a má právo na jejich opravu.
3. Na dodržování těchto pravidel dohlíží nezávislý orgán.“[1, Článek 8]

1.1. GDPR

Zkratka GDPR vychází z anglického General Data Protection Regulation. Český ekvivalent v plném znění je Nařízení Evropského parlamentu a Rady (EU) 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (zkráceně tedy Obecné nařízení o ochraně osobních údajů)[7].

Obecné nařízení přímo stanovuje pravidla pro zpracování osobních údajů, včetně práv fyzických osob a také s volným pohybem těchto údajů. Toto nařízení je charakteristické pro jeho univerzální využití ve všech státech EU (dále včetně Islandu, Norska a Lichtenštejnska)[5].

1.2. Zákon o ochraně osobních údajů

Zákon č.101/2000 Sb., o ochraně osobních údajů byl od 25. května 2018 pro české právní prostředí nahrazen Obecným nařízením o ochraně osobních údajů.

Zákon č. 110/2019 Sb., o zpracování osobních údajů vešel v účinnost 24. dubna. 2019 a nahradil tak starší Zákon č.101/2000 Sb., o ochraně osobních údajů. Tuto novelizaci lze chápat jako implementaci Obecného nařízení o ochraně osobních údajů, která zároveň upravuje dílčí záležitosti, ustanovení, organizaci a další aspekty týkající se Úřadu pro ochranu osobních údajů, které nejsou Obecným nařízením ukotveny [4].

1.3. Základní pojmy

Jak bylo výše uvedeno, je třeba vyjasnit pár základních pojmů, kterým je potřeba dobře porozumět pro pochopení této diplomové práce a obecně práci s osobními údaji. Následující definice vychází z Článku 4, Nařízení Evropského parlamentu a Rady (EU) 2016/679, kde jsou ukotveny v plném rozsahu.

Za **Osobní údaj** se dle GDPR považuje:
„veškeré informace o identifikované nebo identifikovatelné fyzické osobě (dále jen 'subjekt údajů'); identifikovatelnou fyzickou osobou je fyzická osoba, kterou lze přímo či nepřímo identifikovat, zejména odkazem na určitý identifikátor, například jméno, identifikační číslo, údaje o lokaci, síťový identifikátor nebo na jeden či více zvláštních prvků fyzické, fyziologické, genetické, psychické, ekonomické, kulturní nebo společenské identity této fyzické osoby“ [2, Článek 4, bod 1].

V případě anonymizace se pod pojmem identifikovatelná fyzická osoba rozumí také potenciální možnost identifikace pomocí vyčlenění, propojitelnosti nebo odvození[1].

Citlivý údaj je osobní údaj, který vypovídá o národnosti, rase, etnickém původu, politických postojích, členství v politických stranách či hnutích nebo odborových či zaměstnaneckých organizacích, náboženství a filozofickém přesvědčení, trestné činnosti, zdravotním stavu a sexuálním životě fyzické osoby.

Zpracování osobních údajů se rozumí především jejich shromažďování, ukládání, úprava nebo pozměňování, šíření, výměna a mnoho dalších. Obecně se jedná tedy o jakékoliv operace nebo soustavu operací, které správce údajů či zpracovatel provádí.

Správcem údajů je jakýkoliv subjekt, který určuje účel a prostředky zpracování osobních údajů, za které následně i odpovídá.

Zpracovatel není nutným prvkem pro zpracování. Jde pouze o subjekt, který si najímá správce a dává mu pověření pro zpracování osobních údajů.

[1] Těmto pojmům je věnována kapitola Anonymizace dat, podkapitola Anonymizační techniky.

Za **anonymní** považujeme takový **údaj**, který nelze spojit s identifikovanou nebo identifikovatelnou fyzickou osobou. To znamená, že nesmí existovat vazba mezi fyzickou osobou a jejím anonymizovým záznamem.

1.4. Zásady zpracování dat

Zpracování osobních údajů je postaveno na devíti hlavních zásadách. Za jejich dodržování je odpovědný správce údajů a jedná se o tzv. princip odpovědnosti správce [6].
Tyto zásady lze stručně objasnit takto[2]:

- zákonnost, korektnost, transparentnost – správce údajů je povinen zpracovávat osobní údaje pouze na základě minimálně jednoho právního důvodu a toto zpracovaní musí být transparentní vůči subjektu údajů neboli fyzické osobě
- omezení účelu – osobní údaje nesmí být shromažďovány k jiným než legitimním účelům
- minimalizace údajů – relevantnost a přiměřenost shromažďovaných osobních údajů vzhledem k účelu zpracování
- přesnost – osobní údaje nesmí být nijak zkreslené, musí být přesné
- omezení uložení – osobní údaje, které umožňují identifikovat konkrétní osobu, mohou být uloženy pouze po určitou a nezbytnou dobu v závislosti na účelu zpracování
- integrita a důvěrnost – osobní údaje musí být technicky a organizačně zabezpečené

[2]Plné znění těchto zásad je ukotveno v článku 5 odst. 1 Nařízení Evropského parlamentu a Rady (EU) 2016/679.

Kapitola 2

Anonymizace dat

2.1. Anonymizace obecně

Dle Evropského sboru pro ochranu osobních údajů [8] se dá anonymizace považovat
za **nevratné** odebrání informací za účelem zabránit identifikaci konkrétní fyzické osoby. Taková údaje pak lze považovat za dostatečně anonymizované.

Dle Obecného nařízení o ochraně osobních údajů [2] musí být anonymizace provedena takovým způsobem, aby identifikace jedince nebyla možná nebo výpočetně složitá, nákladná, a to i za předpokladu budoucího vývoje a zlepšení technických prostředků. Při splnění tohoto předpokladu je možné data považovat za anonymní.

Údaje, které byly anonymizované, přestávají být osobními. Zpracování takových údajů nevyžaduje dodržování zákona o osobních údajích. To znamená, že organizace může taková data používat i k jiným účelům, než byly pořízeny, a může je uchovávat po neomezenou dobu [2].

2.2. Anonymizace a bezpečnost

Hlavním důvodem anonymizace dat je bezpečnost, neboli zajistit, aby nemohlo dojít k identifikaci jedince. Za předpokladu případného úniku neanonymizovaných dat vzniká problém jak na straně správce údajů, konkrétně například společnosti, tak na straně jedinců neboli zákazníků, kterých se údaje týkají. Kromě finančních náhrad může být jedním z další následků úniku nedůvěra ve společnost, především pokud budou data klientů zneužitelná třetí stranou.

Častým problémem při úniku dat může být problém nebo jakási mezera v zabezpečení firemní databáze.

Právní důsledky

V případě úniku dat zákazníka je poskytovatel ze zákona odpovědný za náhradu škody, nebo v případě smluvní odpovědnosti vyplývá povinnost zaplatit smluvní pokutu. Za únik osobních údajů se dle GDPR může pokuta dostat
až na 20 miliónů eur nebo 4 % z celkových tržeb společnosti. Útok či zneužití dat může být trestným činem jako například neoprávněný přístup k počítačovému systému a nosiči informací. Únik jakýkoliv informací je nutné hlásit Úřadu pro ochranu osobních dat do 72 hodin [14].

Postihy za únik dat

Aerolinkám British Airways byla vyměřena zatím nejvyšší GDPR pokuta ve výši 204,6 milionů eur. Útočníci využili slabého bezpečnostního zabezpečení webových stránek a přesměrovali zákazníky na podvodnou doménu, pomocí níž získali data zákazníků včetně údajů o platebních kartách. Celkově se jednalo o informace přibližně 500 000 klientů.

Na pomyslném žebříčku nejvyšších GDPR pokut v rámci České republiky je nejmenovaná banka, která uchovávala osobní data po uplynutí lhůty pro jejich likvidaci. Pokuta byla stanovena ve výši 250 000 Kč.

V České republice byly i vyšší sankce za únik dat, ale ty vznikly ještě před účinností GDPR. Tyto částky se vyšplhaly až do řádu miliónů a byly souzeny dle zákona č.101/2000 Sb [15].

2.3. Anonymizační techniky

Celá podkapitola vychází především ze Stanoviska č.5/2014 k technikám anonymizace [9], pokud nebude uvedeno jinak.

Anonymizační techniky lze rozdělit do dvou základních skupin. První z nich je **randomizace** a druhá **generalizace**. Přestože obě skupiny obsahují celou řadu technik, tato podkapitola se zaměřuje především na ty z nich, které budou následně využity v praktické části.

Speciálním přístupem je obecně známé **maskování**, kdy se údaj úplně nebo částečně „začerní“ například pomocí hvězdiček.

Tato práce se okrajově zabývá i **pseudonymizací**, která zde slouží především jako možný doplněk k anonymizačním technikám. Bude jí věnována speciální podkapitola.

Obecně lze říci, že při anonymizaci je nutno věnovat pozornost především těmto třem základním rizikům[1]:

- **vyčlenění** – možnost identifikovat fyzickou osobu uvnitř větší skupiny na základě izolace údajů
- **propojení** – propojení nejméně dvou záznamů jedné fyzické osoby
- **dedukce** – vyvození neznámých informací na základě určité pravděpodobnosti

[1]Tato tři rizika můžeme zároveň v jejich znegované formě považovat za kritéria účinné anonymizace (tj. nemožnost vyčlenit jednotlivou osobu, nemožnost propojit dva záznamy týkající se jedné osoby a nemožnost dedukce osoby).

Z těchto rizik mimo jiné plyne, že například pouhé "začernění"osobních údajů, konkrétně například jména a příjmení, není dostatečně účinná anonymizační technika. Problém vyplývá z faktu, že ostatní informace v datech by nadále mohly jednoznačně odkazovat na konkrétního člověka nebo na velmi malou skupinu lidí. A právě z tohoto důvodu by se taková data nedala považovat za anonymizovaná.

O následujících anonymizačních technikách lze konstatovat, že poskytují dostatečnou záruku utajení, a tudíž je lze využít v procesu anonymizování. Je však nutné pečlivě zvážit a promyslet jejich použití. Aby byla anonymizace cílená a výsledkem byly užitečné údaje, je doporučeno nejprve sestavit předpoklady a cíle, kterých chceme anonymizací dosáhnout. Současné použití více různých technik se obecně považuje za optimální řešení. Nedílnou součástí vhodné kombinace technik je dobrá znalost originálních dat, jelikož je nutné k anonymizačnímu procesu přistupovat vždy individuálně, případ od případu.

Anonymizace a její zpětná identifikace je stále v oblasti výzkumu, a proto se musí počítat se zbytkovým rizikem. Tím může být jednak vývoj nových technologií pomáhající identifikaci, nebo naopak pouhé statistické zpracování anonymizovaných dat může kupříkladu i nadále obohatit již existující profily fyzických osob, a tím způsobit nové problémy. Správci údajů by tudíž měli stále analyzovat související rizika a pravidelně je přehodnocovat, jelikož anonymizace není jednorázový úkon. Více o problematice a analýze rizik je popsáno v Metodice obecného posouzení vlivu na ochranu osobních údajů [10].

Terminologie použita v následujících podkapitolách [12] :

- subjekt údajů – fyzická osoba
- soubor údajů – soubor záznamů (tabulka) týkající se subjektů údajů
- atribut – sloupec (např. celé jméno, rok narození, místo narození)
- záznam – týká se jednoho subjektu údajů a skládá se ze souboru hodnot pro každý atribut (např. Alena Nováková, 1996, Bílovec)

- zřejmý atribut – atribut, který sám umožňuje jedinečnou identifikaci subjektu údajů
- kvazi-identifikátory – kombinace atributů, které dokáží vyčlenit subjekt údajů
- pozorovatel – třetí strana, tj. ani správce ani zpracovatel
- útočník – třetí strana, která se náhodně či úmyslně dostane k původním záznamům

2.3.1. Randomizace

Randomizační techniky odstraňují vazbu mezi údaji a fyzickou osobou takovým způsobem, že hodnoty atributů nahradí údaji náhodnými a přidají nepřesnosti či údaje zamění. Lze tedy říci, že se hodnoty modifikují tak, aby se náhodně lišily od jejich skutečných hodnot.

A) Přidání šumu

Přidávání šumu, anglicky „noise addition“ spočívá v pozměnění hodnot kvantitativního atributu tak, aby si zachovaly stejné celkové rozdělení, avšak konkrétní hodnoty byly méně přesné. Pozorovatel by tudíž měl mít dojem, že jsou hodnoty přesné, což bude pravda jen do určité míry.

Pro účely ochrany soukromí musí být přidaná míra nepřesností náležitě promyšlena. Tato míra musí být dostatečně velká, a zároveň dostatečně malá, aby zůstala zachována užitečnost údajů.

Technika přidání šumu je postačující pouze jako doplňkové opatření, které útočníkovi ztíží cestu k získání osobních údajů. Doporučuje se zároveň vymazání zřejmých atributů nebo kvazi-identifikátorů.

Rizika a obvyklé chyby

Přestože jsou záznamy méně přesné, riziko vyčlenění i propojení je zde pořád velké. Jakási dedukce je samozřejmě možná, ale její míra úspěšnosti bude značně nižší.

Pokud se přidání nepřesností vymyká logice a měřítku hodnot, útočník může tuto nepřesnost vyfiltrovat. Další chybou může být přidávání šumu do velmi malého souboru údajů, přičemž bude snadné propojit byť nepřesné údaje s externí databází a následně identifikovat subjekty údajů.

Nedostatky metody

Důvod, proč se přidání šumu doporučuje pouze jako doplňkové opatření, se dá pochopit na příkladu, kdy výzkumní pracovníci dělali pokus zahrnující zpětnou identifikaci s databází zákazníků poskytovatele videoobsahu Netflix.

Analyzovaly se vlastnosti databáze, které obsahovaly 100 miliónů hodnocení více než 18 000 filmů vyjádřených téměř 500 000 uživateli na stupnici 1-5 včetně data hodnocení. Tuto databázi společnost Netflix zveřejnila v anonymizované podobě v souladu s interní politikou a všechny informace, které by mohly identifikovat fyzickou osobu, byly odstraněny. Do této databáze byla přidána určitá míra nepřesností, tudíž hodnocení bylo zvýšeno či sníženo, a bylo pozměněno také datum.

Závěr byl takový, že i přes přidání nepřesností bylo zjištěno, že 99 % uživatelů bylo možno unikátně identifikovat za použití 8 hodnocení, z nichž dvě mohou být zcela špatně, a dat, kdy jednotlivá hodnocení proběhla s nepřesností až 14 dní. [11].

B) Permutace

Jak již název napovídá, metoda permutace spočívá v záměně hodnot atributu v rámci tabulky tak, aby se hodnoty uměle propojily s jinými subjekty údajů, čímž ale nedojde ke změně rozsahu hodnot atributu. Tato metoda se využívá v případě, že je důležité zanechat originální hodnoty, a tím také zachovat rozdělení atributu v rámci souboru údajů.

Existuje-li však mezi dvěma atributy vysoká korelace nebo logický vztah, který se tímto prohozením zruší, mohl by to útočník zjistit a vyměnit zpět [2].

Stejně jako přidání šumu, také permutace by měla být kombinována s jinými technikami nebo s odstraněním zřejmých atributů a kvazi-identifikátorů.

Rizika a obvyklé chyby

Riziko vyčlenění subjektu stále existuje, ale záznamy budou méně spolehlivé. Permutace může zabránit možnosti propojení, na druhou stranu mohou vzniknout chybná propojení subjektu údajů s nesprávnou hodnotou atributu. Pokud některé atributy spolu vzájemně souvisejí nebo mezi nimi existuje logický vztah, potom lze stále provést dedukci. Tato dedukce však bude dána jen s určitou pravděpodobností, protože útočník nebude vědět, které atributy byly vyměněny.

Za vhodné atributy pro permutaci se považují citlivé (např. rasa, náboženství a další), či jinak rizikové atributy, protože jiné nijak významně nepřispějí k ochraně osobních údajů.

Nedostatky metody

Tabulka níže demonstruje problém, že použití permutace na dva atributy spojené logickým vztahem není vhodná a účinná anonymizační technika. Na základě dedukce lze vyvodit, že generální ředitel bude pravděpodobně muž s rokem narození 1985 a má nejvyšší plat. Naopak nezaměstnaný se narodil v roce 1990 a má plat nejnižší.

[2] Tento problém je popsán níže v odstavci Nedostatky metody.

Rok	Pohlaví	Zaměstnání	Příjem(provedena permutace)
1990	M	nezaměstnaný	40 000
1992	Ž	inženýr	43 000
1985	M	generální ředitel	13 000
1990	Ž	inženýr	100 000

Tabulka 2.1: Příklad špatného použití permutace, zpracování vlastní dle [9].

2.3.2. Generalizace

Druhou velkou skupinou jsou generalizační techniky. Jsou založeny na zobecnění informací do takové míry, aby byl výrazně zvýšen počet subjektů, kterým lze danou hodnotu atributu přiřadit.

Typickým příkladem může být zobecnění měst na celé regiony či kraje, měsíc místo týdne nebo při kvantitativních atributech nahrazení konkrétní hodnoty intervalem.

Generalizace může být použita jako samostatná anonymizační technika a zabránit tak vyčlenění jedince ze skupiny, ale nemusí být ve všech případech dostatečně efektivní. Tyto techniky lze obecně doporučit pro velké sady dat, aby byla stejná hodnota přiřazena alespoň pro několik jedinců. Příkladem může být generalizovaný údaj o narození "srpen 1996". Pokud by soubor údajů obsahoval informace o všech osobách České republiky, dá se předpokládat, že je zajištěna dostatečná anonymita subjektu. Pokud by se však soubor údajů týkal menší skupiny lidí, například kolektivu jedné školní třídy, o dostatečnou anonymizaci zpravidla nepůjde.

A) Agregace

Agregace se využívá především u kvantitativních proměnných a jde o zobecnění hodnot pomocí shluků či intervalů. Dá se říci, že se jedná o jednodušší verzi metody k-anonymity, protože zde jako parametr vystupuje pouze požadovaný počet intervalů.

Riziko vyčlenění je zde velké především proto, že může nastat situace, kdy je v intervalu pouze jeden subjekt. Je proto nutné pečlivě zvážit použití této metody

a především dobře znát rozdělení a povahu dat, aby se tomuto problému předešlo.

B) K-anonymita

Cílem k-anonymity je potlačit vyčlenění jedince tak, že je daný subjekt přiřazen ke skupině nejméně k dalších osob. Pokud si například zvolíme $k = 10$, subjekt údajů bude sdílet stejnou hodnotu atributu s nejméně desíti dalšími subjekty.

Ke k-anonymitě existují dvě rozšíření, a to l-diverzita a t-blízkost. Parametr l pak určuje, kolik různých hodnot by navíc mělo být ve skupině s minimálně k subjekty. T-blízkost poté ještě rozšiřuje l-diverzitu tak, že hodnoty navíc musí odrážet původní rozdělení jednotlivého atributu.

Rizika a obvyklé chyby

Riziko vyčlenění by nemělo nastat právě z důvodu, že k subjektů sdílí stejnou hodnotu atributu. Stále je však možné propojit záznamy podle skupin. Největší nedostatek má metoda z podhledu dedukce [3].

Určení hodnoty k může být velmi obtížné, avšak obecným faktem je, že čím je hodnota k vyšší, tím jsou vyšší záruky utajení. Při výběru atributů ke generalizaci je nutné přihlížet ke všem kvazi-identifikátorům. Pokud totiž lze nějaký atribut použít k vyčlenění subjektu z klastru, pak subjekty nejsou dostatečně chráněny. Naopak pokud je hodnota k příliš malá, útok v podobě dedukce bude mít vyšší míru úspěšnosti.

Nedostatky metod Generalizace obecně

Z tabulky níže vyplývá, že riziko dedukce je u těchto metod opravdu velké. Pokud víme, že se soubor údajů týká skupiny studentů matematiky na UPOL a zároveň útočník ví, že se v souboru údajů nachází konkrétní osoba narozená v roce 1990, rovněž zjistí, že se léčí s astmatem.

[3] Toto riziko je detailněji popsáno níže v odstavci Nedostatky metody.

Rok	Pohlaví	Místo narození(generalizováno na kraj)	Diagnóza
1990	M	Olomoucký	astma
1992	M	Olomoucký	vysoký tlak
1985	Ž	Olomoucký	diabetes
1990	M	Olomoucký	astma
1988	Ž	Olomoucký	diabetes

Tabulka 2.2: Příklad špatně promyšlené Generalizace, zdroj vlastní.

2.3.3. Maskování

Technika maskování je založena na nahrazení citlivých údajů jiným libovolným záznamem, který však nebude obsahovat informaci o původní hodnotě.

Metoda stejně jako u generalizace nebo přidání šumu neprohazuje hodnoty mezi jednotlivými subjekty, ale pouze snižuje informaci obsaženou v souboru dat.

Nejčastějším použitím bývá nahrazení určitého počtu znaků hodnoty pomocí teček či hvězdiček nebo dokonce plné nahrazení hodnoty atributu.

Známým příkladem může být maskované telefonní číslo při vícefázovém ověření přihlášení do emailového účtu, ze kterého lze vidět pouze poslední tři číslice (např. *** *** 441).

V některé literatuře metoda maskování spadá pod obecnou skupinu randomizačních technik, a to právě z důvodu, že určitou část hodnoty nahradíme náhodným znakem.

2.4. Pseudonymizace

Pojmem pseudonymizace rozumíme zpracování údajů, při kterém za použití dodatečných informací lze přiřadit osobní údaje k fyzické osobě. Z toho jasně vyplývá, že pseudonymizované údaje dále podléhají Zákonu o zpracování osobních údajů[2].

Základní rozdíl mezi anonymizací a pseudonymizací je především v tom, že data, na která se použije pseudonymizační technika, se dají vrátit do původního stavu. To znamená, že k datům existuje **klíč**, který se dá použít na zpětnou identifikaci.

Nejpoužívanějšími technikami pseudonymizace jsou **šifrování**, **hashování** a **tokenizace**. V případě šifrování může vlastník klíče jednoduše převést pseudonymizovaná data na původní a tak identifikovat každý subjekt údajů. Hashování je funkce, která v určitých případech[4] může být považována za jednosměrnou funkci, avšak není ze zákona považována za anonymizační techniku [13].
Tokenizace se používá především ve finanční oblasti k nahrazení identifikačních čísel jinými hodnotami, které jsou odvozeny od předchozích funkcí.

Techniky pseudonymizace se obecně využívají především k omezení možnosti propojení. Zde je potřeba dbát na to, aby se nepoužíval stejný klíč pro více různých databází. Bohužel ani použití různých rotujících klíčů nemusí být moudré, protože útočník může pomocí vyskytujících se vzorců klíč prolomit. Tajný klíč by samozřejmě neměl být uložen společně s pseudonymizovanými údaji.

Rizika a obvyklé chyby

Nejčastější chybou je mylná domněnka, že pseudonymizované údaje jsou anonymizované. Pokud by byly například nahrazeny identifikační čísla všech subjektů souboru údajů nějakým tokenem, nijak by to neochránilo subjekty před identifikací.

[4] V případě, že rozsah originálních dat je malý a předvídatelný, lze jednoduše vytvořit databázi hashovacích klíču a tu poté porovnat s anonymizovaným údajem a tím zpětně identifikovat subjekt.

2.5. Shrnutí a doporučení

Anonymizace dat, a především zpětná identifikace, jsou předmětem intenzivních výzkumů, a proto není možné u žádné z technik popsaných v této diplomové práci tvrdit, že splňují všechna kritéria účinné anonymizace. Jednotlivé techniky pomáhají určitou míru rizika částečně nebo úplně potlačit. Vybrané z nich jsou ve vztahu k rizikům přehledně zobrazeny v tabulce níže.

Technika	**Riziko vyčlenění**	**Riziko propojení**	**Riziko dedukce**
Randomizace	ano	ano	nemusí
Přidání šumu	ano	nemusí	nemusí
Generalizace	nemusí	ano	ano
Agregace	nemusí	ano	ano
Pseudonymizace	ano	ano	ano
Hashování	ano	ano	nemusí

Tabulka 2.3: Přehled rizik v závislosti na technikách, zpracování vlastní dle [9].

Aby byla dosažena dostatečná míra záruky soukromí, je potřeba náležitě promyslet kombinaci anonymizačních technik. S tím souvisí i jasné stanovení možných souvislostí a cílů procesu anonymizace. Optimální řešení je tedy na individuálním posouzení, avšak toto řešení by mělo splňovat všechna tři kritéria, to znamená proces úplné anonymizace.

Zároveň lze říci, že každá technika má své výhody a nevýhody. Ve většině případů však není možné dát obecná doporučení například na volbu parametrů, protože ke každému souboru údajů je nutno přistupovat individuálně.

Jak již bylo zmíněno, anonymizovaný soubor stále může představovat zbytkové riziko pro subjekty údajů, a to především za použití dalších externích zdrojů informací.

Existují osvědčená opatření, která snižují riziko identifikace:

- Odstranit zjevné atributy a kvazi-identifikátory
- Jasně stanovit cíl, kterého má být anonymizací dosaženo

- Použít kombinaci více různých technik a náležitě ji promyslet
- Kontrolovat, zda jsou rizika vyhodnocována náležitým způsobem, rizika sledovat a pravidelně vyhodnocovat nová potenciální rizika, včetně monitorování rizika zbytkového

Část II

Praktická část

Kapitola 3

Anonymizér

Cílem[1] praktické části diplomové práce bylo naprogramovat jednoduchý nástroj pomocí programovacího jazyka Python, který bude schopný anonymizovat datovou tabulku, načtenou jako CSV soubor. Je zřejmé, že do následného procesu anonymizování bude muset vstupovat lidský faktor, aby určil cíl, kterého má být anonymizací dosaženo, a především kombinaci správných technik.

Tento nástroj, říkejme mu Anonymizér, má být univerzální z pohledu velikosti i obsahu vstupního souboru.

Další důležitou implementací bylo automatické rozeznání alespoň několika základních datových typů neboli obsahu sloupců tabulky, jelikož to je klíčové pro volbu metody anonymizace.

Nedílnou součástí je prostředí aplikace. Tady bylo potřeba skloubit uživatelsky příjemné rozhraní, přehlednost a intuitivní ovládání. Zároveň je třeba zmínit, že cílem není dokonalé grafické uživatelské rozhraní aplikace, ale jakési nahlédnutí do objektového programování, ve kterém se obvykle tento typ aplikací vytváří.

Anonymizovaná data se mimo jiné používají při výzkumech, kdy se pracuje s velkým množstvím dat a podstatné jsou trendy či obecné závěry, nikoliv jednotlivci. Z pohledu statistického zpracování anonymních dat by se dalo očekávat, že po jakési analýze bude možné rozklíčovat výsledky, aby byly nezkreslené. Nabízela se tudíž otázka implementace zpětného „odanonymizování“, neboli

[1] Cíl i jednotlivé požadavky na Anonymizér vzešly ze společné konzultace se společností PricewaterhouseCoopers Audit, s.r.o.

zpětné identifikace dat. Jak bylo ovšem objasněno v kapitole 2, anonymizace musí být nevratná, a z toho vyplývá, že ukládání jakéhokoliv klíče k možnosti zpětné identifikace by bylo v rozporu s cílem této diplomové práce. Závěrem je nutné zmínit, že anonymizace dat se neprovádí pouze za účelem statistické analýzy, ale především k ochraně osobních údajů.

Soubor údajů nebo obecně data, která jsou používaná v této kapitole pro demonstraci fungování Anonymizéru, jsem sama vytvořila pomocí generátorů náhodných hodnot knihovny Faker jazyka Python a upravila jej dle vlastních potřeb. Názvy sloupců jsem záměrně volila jako anglický překlad datových typů, které Anonymizér umí rozpoznat, především pro jednoduchou kontrolu správnosti fungování automatizovaného poznávání datových typů. Tento testovací soubor je zároveň součástí CD přiloženého k diplomové práci.

Poznámka. Při praktické části jsem používala intuitivní terminologii,
a proto se může lehce lišit od oficiálních pojmů. Ve všech případech se jedná o synonyma, jako je nahrazení slov anonymizační technika za anonymizační metodu, atribut za sloupec či datový typ a záznam subjektu údajů za řádek.

3.1. Python, knihovny a instalace Anonymizéru

Jak bylo již dříve zmíněno, Anonymizér je napsaný pomocí programovacího jazyka Python, konkrétně ve verzi 3.6.13 a má téměř 2 800 řádků kódu. Je potřeba objasnit, proč byla použita právě tato verze Pythonu a nikoli verze nejnovější. Stalo se tak především z důvodu kompatibility použitých modulů neboli knihoven.

Nyní si krátce představíme klíčové knihovny, jejich verze a důvod použití.

- NumPy, verze 1.22.4 – práce s vektory, maticemi a vícerozměrnými poli
- Pandas, verze 1.3.4 – práce s tabulkami
- Statics, verze 1.03.5 – základní operace s reálnými čísly

- Faker, verze 8.12.1 – generátor náhodných dat
- PyCryptodome a PyCryptodomex, obojí verze 3.11.0 – knihovny kryptografických algoritmů
- ckwrap, verze 0.1.9 – implementace k-means shlukování na 1D datech
- tkinter, tkintertable a pathlib, verze nehraje důležitou roli – tvorba grafického uživatelského rozhraní

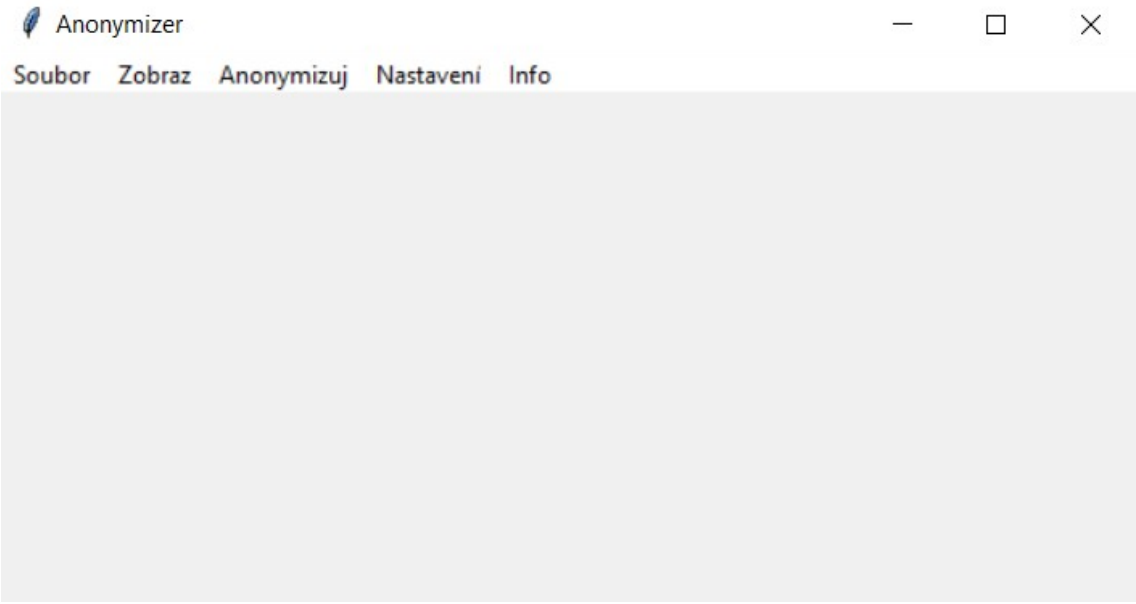
Celá aplikace je k této diplomové práci přiložena na CD. Adresář **Aplikace** je potřeba zkopírovat do uživatelského zařízení. Tento adresář obsahuje veškeré potřebné knihovny Pythonu, samotný Python i vytvořenou aplikaci **Anonymizer.exe**, jenž je umístěná v podadresáři **dist**. Uživatel proto nemusí instalovat Python ani výše uvedené knihovny. Chceme-li spustit aplikaci, přejdeme do podadresáře **dis**t a spustíme **Anonymizer.exe**.

Do adresáře **Testovací data** jsem zároveň vložila již zmíněná data, vytvořená pomocí generátoru náhodných hodnot knihovny Faker, která byla použita jak v rámci testování aplikace, tak k demonstraci ukázek metod v rámci této kapitoly.

Poznámka. Anonymizér byl testován na operačním systému Windows 10 a podporuje zpracování pouze souborů ve formátu CSV. Teto formát vychází z anglického „comma-separated values“ a jde tedy o soubor dat s hodnotami oddělenými čárkou. Některé algoritmy předpokládají data pocházející z českého prostředí, na což bude upozorněno v dalším textu. V případě zpracování dat obsahujících českou diakritiku, aplikace předpokládá použití kódování ANSI Central European (1250).

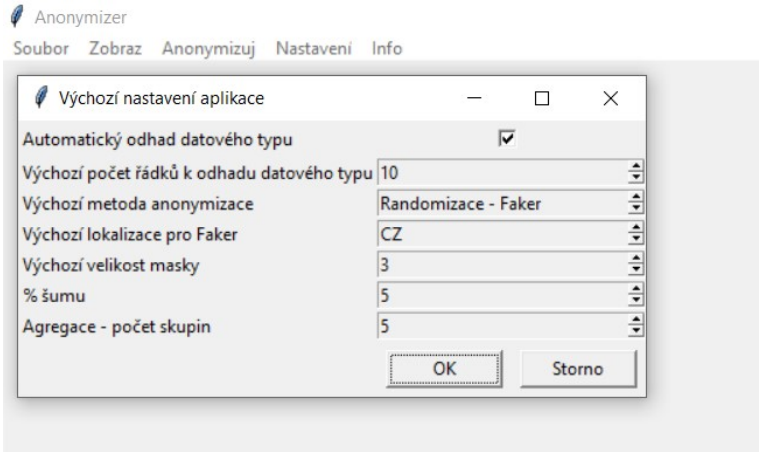
3.2. Prostředí aplikace

Na obrázku 3.1 je ukázka hlavního okna Anonymizéru po spuštění aplikace. V tuto chvíli aplikace obsahuje pouze hlavní menu.



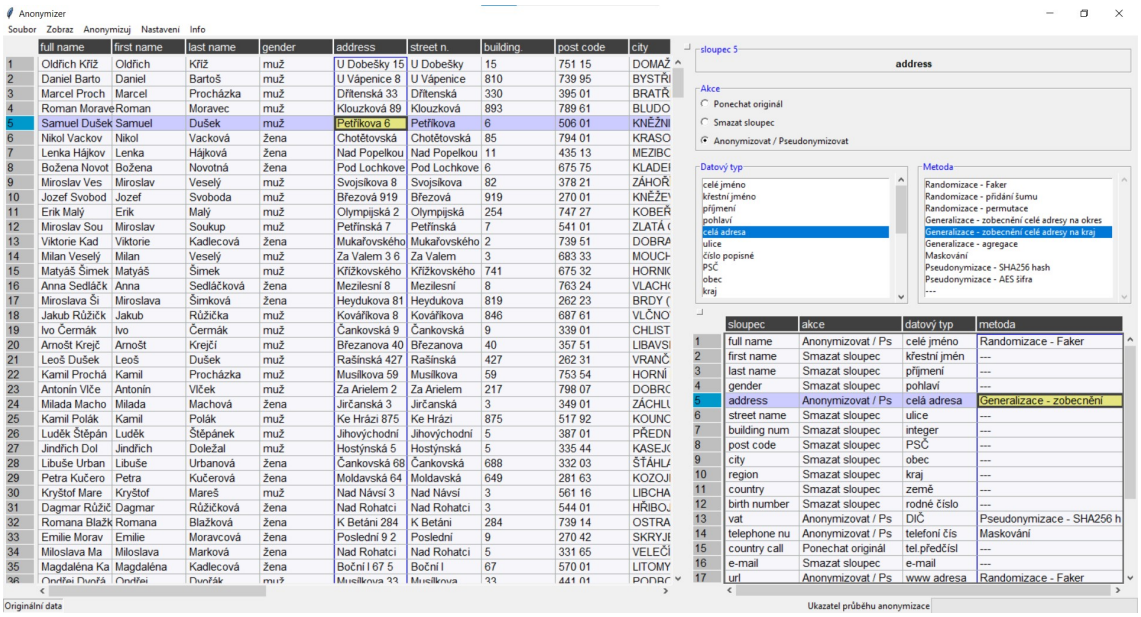
Obrázek 3.1: Hlavní okno Anonymizéru po spuštění.

Globální nastavení aplikace je možno provést volbou menu **Nastavení**.



Obrázek 3.2: Globální nastavení aplikace.

Po načtení souboru příkazem **Soubor** > **Načti originální data...** se zobrazí základní okno aplikace.



Obrázek 3.3: Hlavní okno Anonymizéru s ukázkou nastavení požadavků.

Toto okno můžeme pomyslně rozdělit na tři základní objekty. Levá tabulka zobrazuje vstupní neboli originální data, která byla načtena pomocí **Soubor** > **Načti originální data...** v hlavním menu aplikace. Tato tabulka je zobrazena pouze v režimu čtení, což znamená, že se nedají data manuálně upravovat. Toto zobrazení především umožňuje uživateli pohodlně nahlédnout do dat při určování požadavků anonymizace.

Při načítání originálních dat do aplikace se v průběhu procesu načítání kontroluje, zda není na řádku ve vstupním CSV souboru více polí, než v záhlaví celé tabulky. Pokud tato situace nastane, je pravděpodobné, že při přípravě dat došlo k chybě a oddělovač dat, tedy čárka, je obsažena v některém datovém poli. Bylo by obtížné pracovat s chybnými vstupními daty, proto algoritmus takovýto řádek ze zpracování vyloučí.

Vpravo nahoře jsou umístěny ovládací prvky, které umožňují nastavit požadavky na anonymizaci jednotlivých sloupců. Tyto prvky jsou rozděleny do tří

skupin.

Před volbou požadavků je třeba zvolit sloupec, pro který jsou požadavky určeny. Toto provedeme kliknutím levým tlačítkem myši na jakoukoliv buňku levé tabulky obsahující data. Výběr se tedy neprovádí kliknutím na záhlaví sloupce, ani na index řádku tabulky. Po výběru sloupce je název zvoleného sloupce zobrazen v prvním poli vpravo nahoře.

„Akce“ volí požadovaný postup pro výše uvedeným způsobem zvolený sloupec. Jednotlivé možnosti jsou ponechání originálních hodnot ve zvoleném sloupci, smazání obsahu sloupce, anebo samotné provedení anonymizace/pseudonymizace.

Dalším krokem je určení datového typu zvoleného sloupce. Ve výchozím nastavení aplikace je přednastaven požadavek na automatické rozeznání datového typu. Pokud tento požadavek nebyl vypnut, již při načítání originálních dat se automaticky vybere optimální možnost datového typu ze seznamu[2] a zapíše se do pravé spodní tabulky. Je nutné zkontrolovat správnost odhadu především u sloupců, které budou anonymizovány, případně vybrat ze seznamu vhodnější možnost.

Poslední volbou je výběr anonymizační metody. Výchozí nastavení je volba metody Generalizace - Faker, jelikož je aplikovatelná na všechny datové typy v nabídce. Metodu lze jednoduše změnit výběrem ze seznamu možným metod[3].

Tabulku vpravo dole lze považovat za jakýsi přehled nastavení požadavků na anonymizaci jednotlivých sloupců. První sloupec obsahuje názvy všech sloupců originálních dat a další sloupce odrážejí uživatelské požadavky, které byly nastaveny pomocí výše popsaných ovládacích prvků.

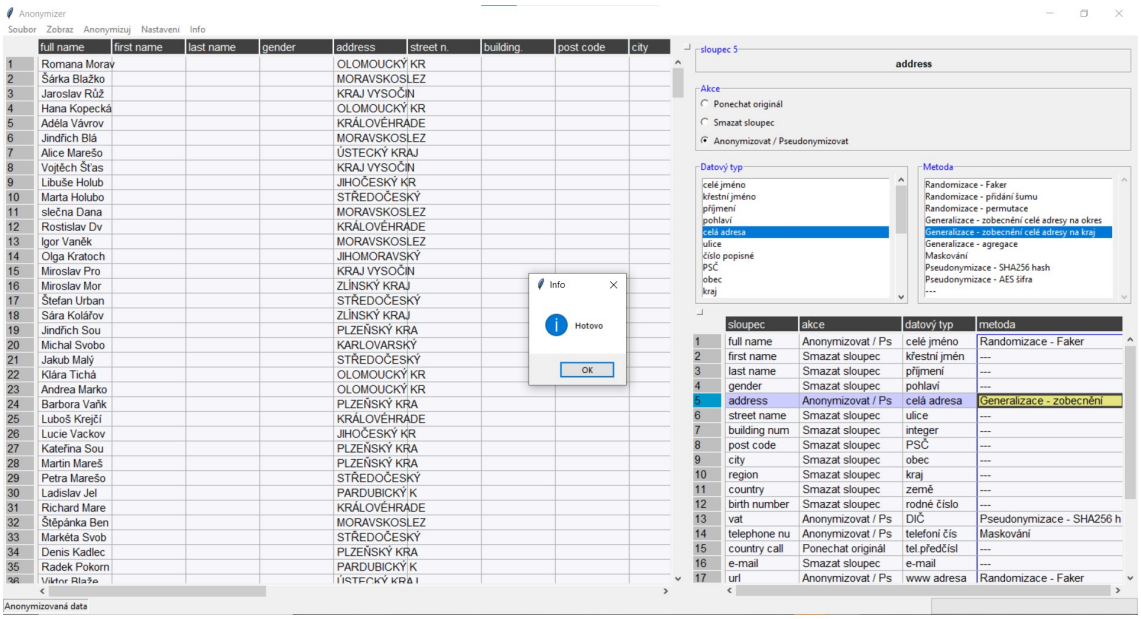
Před spuštěním anonymizačního procesu je nutné zkontrolovat nastavení v tabulce obsahující přehled požadavků a následně v hlavním menu vybrat položku **Anonymizuj > Proveď anonymizaci**. Pokud nastavení metod bylo v souladu s datovým typem, neměl by nastat žádný problém[4]. Pokud anonymizační proces

[2]Seznam všech datových typů, které umí Anonymizér rozpoznat je uveden v podkapitole Automatické rozeznání datového typu.

[3]Popis jednotlivých metod bude vysvětlen níže v podkapitole Metody anonymizace

[4]Přehled kompatability metod a jednotlivých datových typů je zobrazen v Obrázku 3.6

trvá déle, především v závislosti na velikosti vstupního souboru, pak lze ve stavovém řádku vidět grafický ukazatel průběhu anonymizace. Tento ukazatel je doplněn názvem právě zpracovávaného sloupce. Po ukončení procesu anonymizace je zobrazen informativní dialog s textem „Hotovo“ (Obrázek 3.4).



Obrázek 3.4: Hlavní okno aplikace po úspěšném procesu anonymizace.

Po potvrzení dialogu jsou v levé tabulce zobrazena anonymizovaná data. Informace o tom, která data jsou v levé tabulce zobrazena, zda originální nebo anonymizovaná, je zobrazena v levé části stavového řádku aplikace. Mezi zobrazením těchto tabulek lze přepínat příkazem z menu **Zobraz** > **Originální data** nebo **Zobraz** > **Anonymizovaná data**.

Problém může nastat tehdy, když volba uživatelského nastavení metody a datového typu nejsou kompatibilní. V takovém případě se ukáže dialog s upozorněním, že tuto kombinaci nastavení nelze použít, avšak ostatní požadavky budou zpracovány. Sloupec s nevhodným nastavením v anonymizované tabulce bude ponechán v originální podobě.

Po dokončení procesu anonymizace má uživatel dvě možnosti. Pokud je s anonymizací spokojený, pomocí hlavního menu **Soubor** > **Ulož anonymizovaná**

data jako... tabulku jednoduše uloží ve formátu CSV do svého počítače. Pokud však uživatel přehodnotí své požadavky, například chce-li zvolit jinou metodu pro některý sloupec, pak provede korekci příslušného nastavení dle výše uvedeného postupu pro vybraný sloupec. Volbu sloupce, ve kterém chce provést změny, lze realizovat jak v tabulce anonymizovaných dat, tak v tabulce originálních dat.

Poznámka. Ve volbě hlavního menu **Info** > **Nápověda** se zobrazí základní informace o programu a jeho fungování. Součástí nápovědy je i stručné představení jednotlivých implementovaných metod, včetně jejich kompatibility s jednotlivými datovými typy.

3.3. Automatické rozeznání datového typu

Z hlediska běžné programátorské terminologie datový typ proměnné určuje především charakter ukládaných dat a operace, které s těmito datovými typy lze provádět. Mezi běžné datové typy z tohoto hlediska patří logická hodnota, celé číslo, desetinné číslo, textový řetězec a další [23].

Pro účely této aplikace však datový typ představuje určitý obor charakteristických hodnot, pro které jsou definovány metody anonymizace.

Anonymizér umí rozeznat těchto 25 základních datových typů, které můžeme pomyslně rozdělit do pěti kategorií:

- OSOBA : celé jméno, křestní jméno, příjmení, pohlaví
- ADRESA : celá adresa, ulice, PSČ, obec, kraj, země
- ČÍSLA : rodné číslo, DIČ, telefonní číslo, tel.předčíslí
- INTERNET : e-mail, www adresa, IPv4 adresa, IPv6 adresa, MAC adresa

- OBECNÉ : datum, čas, string, integer, real

V případě, že obsah datového sloupce nespecifikuje datový typ z kategorie osoba, adresa, čísla a internet, pak je takovému sloupci přiřazen obecný datový typ string, integer případně real.

Rozeznávací algoritmus je navržen na bázi testování jednotlivých polí tabulky. Tyto testy probíhají již při načítání dat, a aby tento proces nebyl pro obsáhlejší datasety časově náročný, testuje se jen určitý počet řádků. Výchozí nastavení je 10 řádků, ale není problém počet zvýšit či snížit v globálním nastavení Anonymizéru. Povolený rozsah je 1–100. Pokud jsou data například neobvyklá či nepochází z České republiky, je zároveň možnost toto automatické rozpoznávání datového typu v globálním nastavení zcela vypnout.

Každé pole z předem zadaného počtu testovacích řádků prochází celkem 25 testy, které jsou navíc podpořeny 33 dalšími podpůrnými testy. Výsledky testů se zapisují do speciální struktury [5], která byla vytvořena v podobě slovníku. Klíčem slovníku je potom název sloupce a originální neboli testovací hodnota. K tomuto klíči je přiřazen seznam, přičemž první položkou seznamu je počet stejných originálních hodnot v daném sloupci mezi testovacími řádky. Znamená to tedy to, že pokud budeme testovat 10 prvních řádků celého datového souboru a ve sloupci „name“ bude třikrát jméno Alena, tato hodnota se bude testovat pouze jednou, aby se zbytečně nezahlcoval algoritmus. Počet výskytů se ukládá z důvodu nezkreslení výsledků a v případě pozitivních testů se jim pak násobí. Druhou položkou seznamu je vnořený slovník se všemi 25 testy, do kterého se postupně zapisují jednotlivé výsledky z hlediska testů datového typu.

Celkový přehled vyhodnocení datových typů lze zkontrolovat přímo v Anonymizéru v hlavním menu **Zobraz > Výsledky určení datového typu > Celkový přehled**. Zobrazí se tabulka, ve které je v prvním sloupci vypsáno všech 25 možných datových typů. Jednotlivé sloupce pak odrážejí sloupce z načtené ta-

[5] Obecné naznačení struktury slovníku si lze představit takto:
{(sloupec,original_hodnota):[pocet_vyskytu_originalu, {test1:True/False, test2:True/False,..}]}

bulky originálních dat. Tato tabulka říká, kolik polí z předem stanoveného počtu řádků splnilo daný test. Jelikož hodnota zároveň může splnit více různých testů, není nutné, aby suma všech čísel ve sloupci odpovídala testovanému počtu řádků. To se stává pouze u velmi jednoznačných sloupců jako je například e-mail.

Optimální rozhodnutí o odhadu datového typu sloupce pak odpovídá maximu pozitivních testů ve sloupci. Obrázek č.3.3 demostruuje situaci, kdy bylo testováno 10 řádků tabulky a kupříkladu „first name“ bylo 10x pozitivně testováno na datový typ křestní jméno, 7x na přímení a jednou dokonce na obec. Maximum pozitivních testů připadá na datový typ křestní jméno, a proto se následně tento datový typ zapsal do pravé spodní tabulky u příslušného sloupce.

Celkový přehled určení datového typu

	Datový typ	full name	first name	last name	gender	address	street name	building number	post code
1	celé jméno	10	0	0	0	0	0	0	0
2	křestní jmén	0	10	2	0	0	0	0	0
3	příjmení	0	7	10	0	0	2	0	0
4	pohlaví	0	0	0	10	0	0	0	0
5	celá adresa	0	0	0	0	10	0	0	0
6	ulice	0	0	1	0	0	10	0	0
7	číslo popisn	0	0	0	0	0	0	0	0
8	PSČ	0	0	0	0	0	0	0	10
9	obec	0	1	1	0	0	1	0	0
10	kraj	0	0	0	0	0	0	0	0
11	země	0	0	0	0	0	0	0	0
12	rodné číslo	0	0	0	0	0	0	0	0
13	DIČ	0	0	0	0	0	0	0	0
14	telefoní čís	0	0	0	0	0	0	0	0
15	tel.předčísl	0	0	0	0	0	0	0	0
16	e-mail	0	0	0	0	0	0	0	0
17	www adresa	0	0	0	0	0	0	0	0
18	IPv4 adresa	0	0	0	0	0	0	0	0
19	IPv6 adresa	0	0	0	0	0	0	0	0
20	MAC adresa	0	0	0	0	0	0	0	0
21	datum	0	0	0	0	0	0	0	0
22	čas	0	0	0	0	0	0	0	0
23	string	0	0	0	0	0	0	0	0
24	integer	0	0	0	0	0	0	10	0
25	real	0	0	0	0	0	0	0	0
26	---	0	0	0	0	0	0	0	0

Obrázek 3.5: Celkový přehled určení datového typu.

3.3.1. Testy datových typů

V této části algoritmizování bylo potřeba především dobře pochopit formát každého datového typu, aby bylo možné sestavit funkční test. Šlo především o podchycení podmínek, z jakých znaků se například může skládat MAC adresa nebo v jakém tvaru musí být email a jeho doména.

Podporou rozpoznávání datových typů je 10 CSV souborů[6], které Anonymizér využívá v určitých případech pro určení datového typu. Jedná se o soubory obsahující databázi ženských jmen, mužských jmen, ženských příjmení, mužských příjmení, ulic, obcí a krajů platných na území České republiky. Zvláštní význam má soubor **PSC_obec_okres_kraj.csv**, který vytváří vazbu mezi PSČ, obcí, okresem a krajem. Poslední dva soubory se týkají zemí a telefonních předčíslí. Tyto databáze jsem si vytvořila sama za použití veřejně dostupných zdrojů Ministerstva vnitra České republiky, Českého statistického úřadu a Státní správy zeměměřictví a katastru.

Aby mohlo být testování úspěšné, je důležitý správný formát originální hodnoty, například nutný znak plus před telefonním předčíslím.

Cílem jednotlivých testů nebylo podchytit úplně všechny alternativy formátů pro jednotlivé datové typy, přesto je většina testů určujících datový typ založena na kombinaci nejdůležitějších podmínek pro úspěšný odhad datového typu.

Speciální přístup při vyhodnocování datového typu je v případě datového typu **string**. Všechny originální hodnoty by bylo možno považovat za tento datový typ. Proto se datový typ string přiřazuje až v případě, kdy obsah načteného pole nebyl ztotožněn s žádným jiným datovým typem.

Nyní si představíme několik konkrétních testů s jejich podmínkami. Testy datových typů, které souvisejí s podpůrnými CSV soubory, jsou založeny především na jakémsi vyhledávacím algoritmu. Algoritmus se pokouší vyhledat originální hodnotu testovaného pole ve slovnících, které jsou vytvořené na základě těchto

[6]Jednotlivé CSV soubory jsou umístěny na přiloženém CD ve složce **dist\pomocne_csv**. Jednotlivé názvy souborů jsou voleny intuitivně: kraje.csv, krestni_muzi.csv, krestni_zeny.csv, obce.csv, predcisli.csv, prijmeni_muzi.csv, prijmeni_zeny.csv, PSC_obec_okres_kraj.csv, ulice.csv, zeme.csv .

podpůrných CSV souborů. Tento vyhledávací algoritmus se využívá pro datové typy křestní jméno, přímení, ulice, kraj, obec, země a telefonní předčíslí.

E-mail

Na první pohled by se dalo konstatovat, že odhad e-mailu je velmi jednoduché otestovat díky speciálnímu znaku, jímž je zavináč. Rozhodla jsem se však podmínky trochu rozšířit, protože znak zavináče se teoreticky může vyskytovat i v jiných sloupcích.

Algoritmus tohoto testu má v sobě zahrnutých dalších několik podpůrných testů, které v Ukázce kódu 3.2 již nejsou ukázány. Jejich názvy jsou však pojmenovány intuitivně, a tak lze jednoduše odhadnout, jaký mají jednotlivé funkce cíl.

Testem se tedy rozumí splnění několika podmínek, aby byla hodnota sloupce označena za e-mail. První podmínkou je již zmíněný znak zavináče vyskytující se někde uprostřed celého řetězce. Pomocí jednoduchých úprav si najdeme pozici tohoto znaku v řetězci, a tak potenciální e-mail rozdělíme na část se jménem a část s doménou. Na e-mailové jméno jsou kladeny určité podmínky, a to především co se týče jeho délky a povolených znaků. Jednotlivé znaky jsou převáděny na číselnou hodnotu pomocí funkce **ord()** a následně porovnávány na povolený rozsah dle ASCII tabulky[7]. Část označována jako jméno může obsahovat maximálně 64 znaků [30].

Poslední podmínka se týká domény. Doména musí být v rozsahu 4 až 255 znaků. Z toho vyplývá, že nejmenší doména první úrovně může vypadat například takto „a.cz“. Musí končit generickou nebo geografickou doménou nejvyššího řádu[8]. Algoritmus aplikace kontroluje kromě platných generických a geografických domén také povolené znaky domény, což jsou znaky anglické abecedy, tečka a pomlčka.

[7] ASCII je americký standard pro převod znaků na čísla. Jedná se o tabulku o 128 řádcích, kde každý řádek obsahuje znak anglické abecedy (případně znak používaný v informatice) jeho číselný ekvivalent [18].

[8] Generické domény sdružují obecné domény (např. .org“), geografické domény sdružují domény jednoho státu (např. „.cz“).[19]

Nesmí obsahovat tečku na začátku ani na konci, musí být pouze mezi písmeny [20].

Jestliže jedna z těchto podmínek není splněna, pak hodnota sloupce nemůže být označena za datový typ e-mail.

Jak jsem již dříve uvedla, tento výčet podmínek určitě nemá obsažené veškeré podmínky na e-mailové jméno či doménu, ale pouze ty základní pro dostatečně úspěšnou identifikaci nebo především odlišení od ostatních datových typů. Cílem automatického rozeznávaní datových typů totiž není kontrolovat správnost formátu či platnost e-mailu jako takovou.

Ukázka kódu 3.1: Test e-mail

```
def test_email(hodnota_pole):
    p1=uprostred_retezce_obsahuje_prave_jeden_znak(hodnota_pole,'@')
    p=True
    if p1:
        pozice_zavinace=prvni_pozice_znaku_v_retezci(hodnota_pole,'@')
        jmeno=jmeno_email(hodnota_pole,pozice_zavinace)
        domena=domena_email(hodnota_pole,pozice_zavinace)
        p2=jmeno_email_povolene_znaky(jmeno)
        p3=len(jmeno)<=64
        p4=Domenove_jmeno(domena)
        p=p1 and p2 and p3 and p4
    else:
        p=False
    return p
```

Celá adresa

V praxi může být adresa zapsána v různých formátech. Je velmi obtížné odhadnout, jak budou konkrétně uspořádány jednotlivé klíčové údaje v adrese. Správný formát adresy obsahuje následující údaje: ulice, číslo popisné, PSČ a obec. V malých obcích je však obvyklé, že není udáván název ulice. Číslo popisné bývá v adrese udáváno vždy, ale neexistuje žádné omezení, které by vyjmenovávalo povolená čísla popisná. Navíc je docela jednoduché zaměnit část PSČ a číslo popisné.

Poměrně dlouhý řetězec adresy je pomocí funkce **split()** rozdělen na dílčí řetězce. Algoritmus vyhledává dva po sobě jdoucí řetězce, které dohromady představují platné PSČ jak z hlediska formátu, tedy tři číslice mezera dvě číslice, tak z hlediska existence v rámci České republiky, což je kontrolováno díky pomocnému slovníku.

Protože jedno PSČ může být přiděleno více obcím, je následně provedena kontrola, zda některý ze zbývajících řetězců v adrese odpovídá některé z obcí, které algoritmus vyhledal ve slovníku obsahujícím dvojice PSČ a název obce.

Aby algoritmus zbytečně netestoval krátké řetězce, tak kontroluje, že je minimální počet podřetězců větší nebo roven čtyřem. Předpokládáme totiž, že adresa musí obsahovat PSČ, název obce a číslo popisné. Správný formát PSČ se skládá ze dvou řetězců, proto hovoříme o čtyřech podřetězcích.

Ukázka kódu 3.2: Test adresy

```
def test_adresa(hodnota_pole):
    split_adresy=hodnota_pole.split(' ')
    pocet=0
    for polozka in split_adresy:
        pocet=pocet+1
    p1=pocet>=4
    if not(p1):
        return False
    else:
        neexistuje_obec=True
        index=0
        for polozka in split_adresy[:-1]:
            try:
                PSC=split_adresy[index]+' '+split_adresy[index+1]
                seznam_obci=s_PSC_obce[PSC]
                for polozka_seznamu_obci in seznam_obci:
                    obec_nalezena=False
                    if hodnota_pole.find(polozka_seznamu_obci)!=-1:
                        return True
            except:
                pass
            index=index+1
        return False
```

Celé jméno

Kromě běžných českých jmen a příjmení jsou aktuálně v databázi platných jmen a příjmení také jména cizinců žijících na území České republiky, která jsou často tvořena z několika částí. Při vyhodnocování datového typu celé jméno navíc nevíme, zda křestní jméno je na začátku řetězce, nebo na jeho konci. Databáze jmen jsou poměrně rozsáhlé, jedná se o desítky tisíc položek. Pokud bychom vyhledávali všechna možná jména v hodnoceném vzorku, bylo by to velmi časově náročné. Proto jsem zvolila opačný postup.

Pomocí funkce **split()** jsem rozdělila originální hodnotu na dílčí řetězce. Algoritmus následně v cyklu vyhledává, zda prvních 1 až n částí tvoří křestní jméno. Pokud algoritmus neztotožní žádnou skupinu řetězců s křestním jménem z databáze, začne vyhodnocovat tutéž podmínku od druhého dílčího řetězce až do konce originální hodnoty. S tímto krokem algoritmus pokračuje dál.

Pokud algoritmus nalezne křestní jméno, následně ověřuje, jestli je ve zbývající části testované hodnoty platné příjmení. V takovém případě testovací funkce považuje obsah hodnoty za datový typ celé jméno.

Ukázka kódu 3.3: Test celého jména

```
def cele_jmeno(hodnota_pole):
    seznam_casti_jmena=hodnota_pole.split(' ')
    testovany_zlomek_jmena=''
    i=0
    for cast in seznam_casti_jmena[:-1]:
        if i==0:
            testovany_zlomek_jmena=testovany_zlomek_jmena+cast
        else:
            testovany_zlomek_jmena=testovany_zlomek_jmena+' '+cast
        i=i+1
        delka_testovaneho_zlomku=len(testovany_zlomek_jmena)
        zbytek_jmena=hodnota_pole[delka_testovaneho_zlomku+1:]
        if krestni_jmeno(testovany_zlomek_jmena):
            if test_prijmeni(zbytek_jmena):
                return True
        if test_prijmeni(testovany_zlomek_jmena):
            if krestni_jmeno(zbytek_jmena):
                return True
```

IPv6 adresa

Internetový protokol nazývaný IP je sada norem, které určují způsob, jakým zařízení připojená k internetu spolu komunikují. IP adresa je základním identifikátorem komunikujícího zařízení.

IPv6 je nejnovější internetový protokol verze 6 pro komunikaci a přenos dat v síti. Největším rozdílem IPv6 a IPv4 je především podstatné rozšíření adresního prostoru, neboť se blíží okamžik, kdy budou volné IPv4 adresy vyčerpány.

IPv6 adresa je tvořena osmi bloky hexadecimálních čísel oddělenými dvojtečkami. Každý blok může být tvořen maximálně čtyřmi číslicemi [22].

IPv6 adresa může mít různé formáty[9]. Pro testování jsem se rozhodla předpokládat plné formáty, nikoliv zkrácené alternativy.

Z tohoto důvodu by měla být splněna podmínka, že počet znaků testovaného řetězce by měl být mezi 15 a 39, přičemž sedm z nich jsou dvojtečky. Mezi povolené znaky IPv6 adresy patří, krom již zmíněné dvojtečky, velká písmena A-F, malá písmena a-f a číslice 0-9 (šestnáctková soustava). Vzhledem k tomu, že uvažujeme plné formáty, znak dvojtečky by proto neměl být na začátku, ani na konci celého řetězce, ale pouze mezi jinými povolenými znaky. Když jsou splněny všechny tyto požadavky, kontroluje se obsah jednotlivých bloků.

Poznámka. Ukázky kódů 3.5 a 3.6 jsou pouze podpůrné testy. Rozhodla jsem se je přiložit jen pro demonstraci, jelikož jsou v obdobných variantách používány téměř ve všech předešlých Ukázkách kódů.

[9] Příklad plného formátu IPv6 adresy:
2001:0db8:0000:0000:0000:0000:1428:57ab
Příklad zkráceného formátu IPv6 adresy:
2001:db8::1428:57ab

Ukázka kódu 3.4: Test IPv6 adresa

```
def IPv6_adresa(hodnota_pole):
    p1=(len(hodnota_pole)>=15) and (len(hodnota_pole)<=39)
    p2=IPv6_povolene_znaky(hodnota_pole)
    p3=pocet_vyskytu_znaku_v_retezci(hodnota_pole,':')==7
    p4=not(vlastnost_DT_obsahujeZnakNaZacatku(hodnota_pole,':'))
    p5=not(vlastnost_DT_obsahujeZnakNaKonci(hodnota_pole,':'))
    p6=vlastnost_DT_ZnakMeziPismeny(hodnota_pole,':')
    p7=True
    if p1 and p2 and p3 and p4 and p5 and p6:
        split_adresy=hodnota_pole.split(':')
        for adresa in split_adresy:
            if int('0x'+adresa,16)<0 or int('0x'+adresa,16)>0xFFFF:
                p7=False
    else:
        p7=False
    return p7
```

Ukázka kódu 3.5: Podpůrný test povolených znaků IPv6 adresy.

```
def IPv6_povolene_znaky(hodnota_pole): # A–F, a–f, 0–9, :
    jenHexaCisliceaDvojtecka=True
    for znak in hodnota_pole:
        p1=ASCII_JeCislice(znak)
        p2=mala_pismena_Hexa(znak)
        p3=velka_pismena_Hexa(znak)
        p4=znak==':'
        if not(p1 or p2 or p3 or p4):
            jenHexaCisliceaDvojtecka=False
    return jenHexaCisliceaDvojtecka
```

Ukázka kódu 3.6: Podpůrný test demonstrující převod znaků na čísla a kontrolu povoleného rozsahu dle ASCII tabulky.

```
def ASCII_JeCislice(znak):
    jenCislice=False
    if (ord(znak)>=48 and ord(znak)<=57) :
        jenCislice=True
    return jenCislice
```

3.4. Metody anonymizace

Ještě před tím, než budou vysvětleny jednotlivé anonymizační metody implementované v Anonymizéru, je potřeba představit dvě základní struktury, které jsou v kódu využívány a souvisí s anonymizačním procesem.

Již při načítaní dat se vytvoří slovník unikátních klíčů. Klíč obsahuje název sloupce a unikátní hodnotu v daném sloupci. Vytvoření tohoto slovníku je motivováno požadavkem nahrazovat stejnou originální hodnotu stejnou anonymizovanou hodnotou. Při inicializaci tohoto slovníku se tomuto klíči přiřazuje výchozí hodnota v podobě řetězce „fake“. Teprve po spuštění anonymizačního algoritmu je tato hodnota nahrazena konkrétní hodnotou dle použité metody. Použití slovníku zároveň zjednodušuje algoritmy. Této struktuře budu pro zjednodušení říkat **slovník unikátních hodnot**[10]. Když přijde na řadu proces anonymizace, u některých anonymizačních metod se anonymizují pouze tyto unikátní hodnoty.

Druhou důležitou strukturou je **dočasný slovník** [11], který je jakýmsi pracovním prostorem všech algoritmů. Klíčem tohoto slovníku je číslo řádku a název sloupce dle originálně načtených dat. Ke každému klíči je přiřazen seznam dvou hodnot, přičemž první z nich je hodnota originálu a druhá je anonymizovaná hodnota. Tento přístup umožňuje opětovné spouštění anonymizačního procesu s tím, že jsou původní originální data ochráněna před modifikací.

Nyní si vysvětlíme základní principy fungování jednotlivých anonymizačních metod implementovaných v Anonymizéru. Základní přehled metod a jejich kompatibilitu s jednotlivými datovými typy lze vyčíst z obrázku 3.6.

[10] Obecné naznačení struktury slovníku unikátních hodnot :
$\{(sloupec, unikatni\_hodnota) : fake\}$

[11] Obecné naznačení struktury dočasného slovníku :
$\{(cislo\_radku, sloupec) : [originalni\_hodnota, anonymizovana\_hodnota]\}$

	Faker	přidání šumu	permutace	zobecnění celé adresy na okres	zobecnění celé adresy na kraj	agregace	maskování	hash	šifra
celé jméno	✔		✔				✔	✔	✔
křestní jméno	✔		✔				✔	✔	✔
příjmení	✔		✔				✔	✔	✔
pohlaví	✔		✔				✔	✔	✔
celá adresa	✔		✔	✔	✔		✔	✔	✔
ulice	✔		✔				✔	✔	✔
číslo popisné	✔		✔				✔	✔	✔
PSČ	✔		✔				✔	✔	✔
obec	✔		✔				✔	✔	✔
kraj	✔		✔				✔	✔	✔
země	✔		✔				✔	✔	✔
rodné číslo	✔		✔				✔	✔	✔
DIČ	✔		✔				✔	✔	✔
telefoní číslo	✔		✔				✔	✔	✔
tel.předčíslé	✔		✔				✔	✔	✔
e-mail	✔		✔				✔	✔	✔
www adresa	✔		✔				✔	✔	✔
IPV4 adresa	✔		✔				✔	✔	✔
IPV6 adresa	✔		✔				✔	✔	✔
MAC adresa	✔		✔				✔	✔	✔
datum	✔		✔				✔	✔	✔
čas	✔		✔				✔	✔	✔
string	✔		✔				✔	✔	✔
intiger	✔	✔	✔			✔	✔	✔	✔
real	✔	✔	✔			✔	✔	✔	✔

Obrázek 3.6: Kompatibilita metod a datových typů. Zdroj vlastní.

3.4.1. Randomizace - Faker

Faker je knihovna Pythonu, která generuje falešná data.

Pomocí funkce **Faker()** se vytvoří a inicializuje generátor, který následně umí genegovat falešná data různých druhů. Pokud nemá funkce v závorce žádný argument národního prostředí neboli lokalizovaného poskytovatele, výchozí nastavení bude americká angličtina "en_US".

Předpokladem Anonymizéru jsou vstupní data, která pocházejí z České republiky, a proto byla při inicializaci generátoru použita lokalizace "cs_CZ".

Proces anonymizace se při této metodě provádí nad slovníkem unikátních hodnot. To znamená, že algoritmus funguje následujícím způsobem. Pomocí názvu sloupce, který je uvedený v klíči slovníku unikátních hodnot, si vyhledá jeho příslušný datový typ, který byl buď odhadnut algoritmem nebo určen manuálně. Tímto způsobem generátor zjistí, jaký typ dat má generovat. Použití vlastnosti **.unique** při generování pak zaručuje, že vygenerované hodnoty jsou jedinečné pro konkrétní anonymizační proces. To znamená, že každé položce slovníku unikátních hodnot originálních dat bude přiřazena jedinečná falešná hodnota příslušného datového typu. Nabízí se otázka, co se stane v případě, že dojdou unikátní hod-

noty generátoru, protože je zřejmé, že jednotlivé databáze hodnot datových typů jsou konečné. V takovém případě nastane v běhu programu výjimka. Algoritmus je navržen tak, aby generoval hodnoty do té doby, dokud bude mít k dispozici unikátní hodnoty falešných dat. Když nastane situace, že je unikátních hodnot originálních dat více než v databázi generátoru, začnou se generovat falešné hodnoty složené z názvu sloupce a čísla (např. first name 1, first name 2, atd.). Tento přístup zajistí, že bude zachována unikátnost generovaných anonymizovaných dat a zároveň nebude anonymizační algoritmus předčasně přerušen kvůli nedostatku unikátních falešných hodnot.

Upozorňuji, že jedinečnost vygenerovaných hodnot se týká jednoho celého anonymizačního procesu, což znamená, že při použití Fakeru pro stejný datový typ u více sloupců roste pravděpodobnost vyčerpání unikátních falešných hodnot. Toto riziko roste s rozsáhlostí zpracovávaného datového souboru.

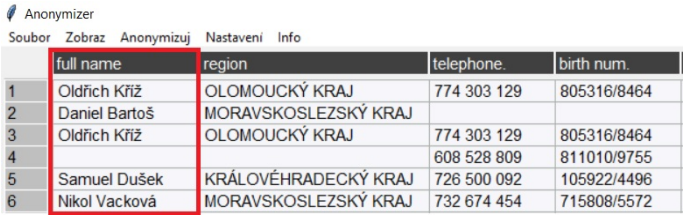
V rámci metody Faker se nekontrolují logické souvislosti mezi jednotlivými sloupci, například mezi jménem a příjmením, která jsou uvedena ve dvou rozdílných sloupcích. Faker by mohl například vygenerovat mužské jméno a ženské přímení. Pokud bychom chtěli tuto logiku zachovat, doporučila bych jeden z uvedených sloupců smazat a druhému manuálně změnit datový typ na celé jméno, a tak docílit smysluplné kombinace jmen a přímení, přestože původní obsah sloupce byl například jen přímení. Podobná situace by mohla nastat v případě generování jednotlivých částí adresy (ulice, obec, PSČ,...) a zároveň celé adresy.

Nebudu zde ukazovat celý kód k metodě, jelikož je velmi obsáhlý. Konkrétní funkce, kterou se generují jednotlivé falešné hodnoty, například celého jména, pak má tvar **faker.unique.name()** a odkazuje se tím na již zmíněny generátor inicializovaný výše.

Generátor Faker je defaultně nastaven tak, aby generoval falešné hodnoty i pro prázdnou hodnotu pole. Tuto vlastnost jsem se rozhodla potlačit, aby byla zachována původní struktura dat.

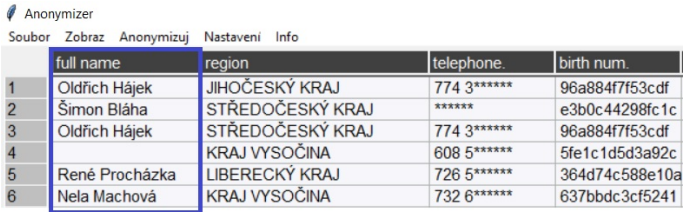
V tabulce 3.1 jsou ukázky použití metody Faker na konkrétních datových typech.

Poznámka. V některých případech Faker generuje vizuálně korektní data, avšak při bližším ověření zjistíme, že například rodné číslo nebo DIČ nesplňují kontrolní součty, celá adresa je tvořena geograficky nesouvisejícími údaji a co se týče generování stringu, jsou to mnohdy až vtipné řetězce.



	full name	region	telephone.	birth num.
1	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
2	Daniel Bartoš	MORAVSKOSLEZSKÝ KRAJ		
3	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
4			608 528 809	811010/9755
5	Samuel Dušek	KRÁLOVÉHRADECKÝ KRAJ	726 500 092	105922/4496
6	Nikol Vacková	MORAVSKOSLEZSKÝ KRAJ	732 674 454	715808/5572

Obrázek 3.7: Ukázka originálních hodnot sloupce full name, datového typu celé jméno.



	full name	region	telephone.	birth num.
1	Oldřich Hájek	JIHOČESKÝ KRAJ	774 3******	96a884f7f53cdf
2	Šimon Bláha	STŘEDOČESKÝ KRAJ	******	e3b0c44298fc1c
3	Oldřich Hájek	STŘEDOČESKÝ KRAJ	774 3******	96a884f7f53cdf
4		KRAJ VYSOČINA	608 5******	5fe1c1d5d3a92c
5	René Procházka	LIBERECKÝ KRAJ	726 5******	364d74c588e10a
6	Nela Machová	KRAJ VYSOČINA	732 6******	637bbdc3cf5241

Obrázek 3.8: Ukázka anonymizovaných hodnot sloupce full name, po použití metody Faker.

3.4.2. Randomizace - přidání šumu

Přidání šumu spočívá v pozměnění hodnot tak, aby si zachovaly celkové rozdělení, avšak konkrétní hodnoty budou méně přesné.

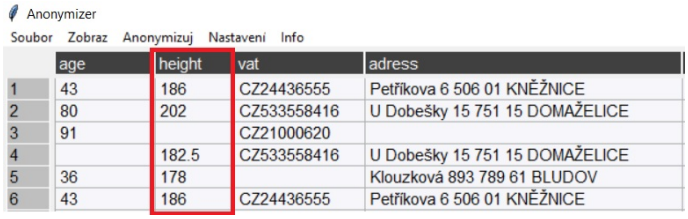
Metoda je použitelná pouze na sloupce, které obsahují číselné hodnoty. Algoritmus nejprve připraví seznam všech originálních hodnot zpracovávaného sloupce, poté vypočte standardní odchylku z těchto hodnot pomocí funkce **statistics.stdev()**. Dle přednastaveného procenta šumu, který je definován v globálním nastavení programu je vypočteno procento této standardní odchylky. Šum je následně generován pomocí funkce
np.random.normal(0,procento-smerodatne-odchylky,len(seznam-hodnot)). Z předpisu funkce vyplývá, že je šum generován normálním rozdělením se střední hodnotou rovnou nule a za směrodatnou odchylku je považováno předem určené procento z variability originálních hodnot, aby se přidání nepřesností nevymykalo měřítku. Počet vygenerovaných hodnot šumu je určen počtem originálních hodnot, což je zároveň poslední parametr této funkce.

Posledním krokem algoritmu je přičtení jednotlivých hodnot šumu k originálním hodnotám.

V rámci této metody algoritmus pracuje přímo s originálními daty v **dočasném slovníku**, nikoliv se **slovníkem unikátních hodnot**, proto i dvě stejné hodnoty originálních dat mohou nabývat rozdílné anonymizované hodnoty.

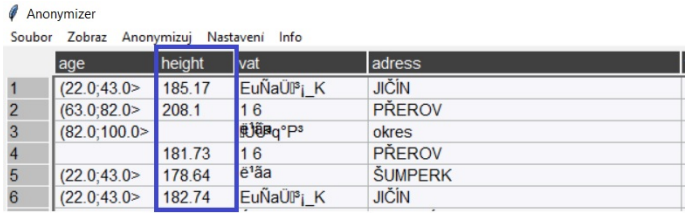
V případě, že by se v originálních datech vyskytla kromě číselných hodnot také nečíselná hodnota, pak by bylo anonymizované pole nečíselné hodnoty prázdné. Pokud by ve sloupci nebyly žádné číselné hodnoty, pak jsou anonymizovaná pole vyplněna textem „fake“.

Poznámka. Procento šumu lze v dialogu **Výchozího nastavení aplikace** nastavit v rozsahu 5 až 20 %.



	age	height	vat	adress
1	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE
2	80	202	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
3	91		CZ21000620	
4		182.5	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
5	36	178		Klouzková 893 789 61 BLUDOV
6	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE

Obrázek 3.9: Ukázka originálních hodnot sloupce height, datového typu real.



	age	height	vat	adress
1	(22.0;43.0>	185.17	EuÑaÜÏ³¡_K	JIČÍN
2	(63.0;82.0>	208.1	1 6	PŘEROV
3	(82.0;100.0>		ßÜëªq°P³	okres
4		181.73	1 6	PŘEROV
5	(22.0;43.0>	178.64	ë¹ãa	ŠUMPERK
6	(22.0;43.0>	182.74	EuÑaÜÏ³¡_K	JIČÍN

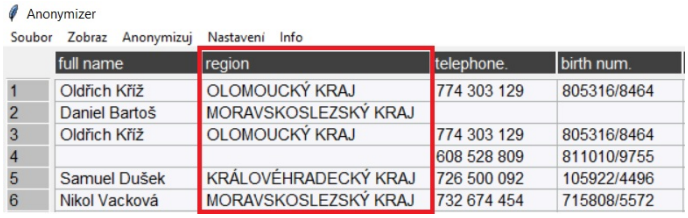
Obrázek 3.10: Ukázka anonymizovaných hodnot sloupce height, po použití metody přidání 5% šumu.

3.4.3. Randomizace - permutace

Metoda permutace pracuje s údaji v jediném sloupci. Cílem této metody je tedy náhodně přehodit pořadí údajů v rámci jednoho sloupce. Stejně jako v předchozím případě, i tato metoda pracuje nad celým rozsahem tabulky, tj. **dočasným slovníkem**.

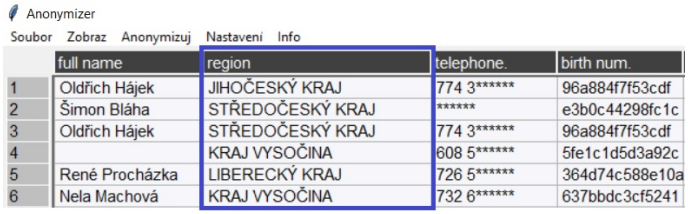
Algoritmus pracuje na principu seznamu, do kterého uloží všechny originální hodnoty. V následujícím kroku zjistí počet položek seznamu. Dále pomocí funkce **random.randint(0,int(pocet hodnot)-1)** generuje ukazatel na jednu položku v tomto seznamu originálních hodnot. Tuto položku následně zkopíruje na cílovou pozici v tabulce anonymizovaných hodnot a zároveň ze zdrojového seznamu originálních hodnot tuto položku vyjme. Cílová pozice se určuje v cyklu počínaje prvním řádkem v tabulce s krokem jedna. Celý cyklus se opakuje až do okamžiku, kdy jsou všechny položky ze seznamu originálních hodnot přesunuty do cílových řádků anonymizovaného sloupce.

Poznámka. Indexy položek v seznamu jsou v programovacím jazyku Python indexovány od nuly, je to zohledněno ve vstupních parametrech při generování ukazatele.


Anonymizer
Soubor Zobraz Anonymizuj Nastavení Info

	full name	region	telephone.	birth num.
1	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
2	Daniel Bartoš	MORAVSKOSLEZSKÝ KRAJ		
3	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
4			608 528 809	811010/9755
5	Samuel Dušek	KRÁLOVÉHRADECKÝ KRAJ	726 500 092	105922/4496
6	Nikol Vacková	MORAVSKOSLEZSKÝ KRAJ	732 674 454	715808/5572

Obrázek 3.11: Ukázka originálních hodnot sloupce region, datového typu kraj.


Anonymizer
Soubor Zobraz Anonymizuj Nastavení Info

	full name	region	telephone.	birth num.
1	Oldřich Hájek	JIHOČESKÝ KRAJ	774 3******	96a884f7f53cdf
2	Šimon Bláha	STŘEDOČESKÝ KRAJ	******	e3b0c44298fc1c
3	Oldřich Hájek	STŘEDOČESKÝ KRAJ	774 3******	96a884f7f53cdf
4		KRAJ VYSOČINA	608 5******	5fe1c1d5d3a92c
5	René Procházka	LIBERECKÝ KRAJ	726 5******	364d74c588e10a
6	Nela Machová	KRAJ VYSOČINA	732 6******	637bbdc3cf5241

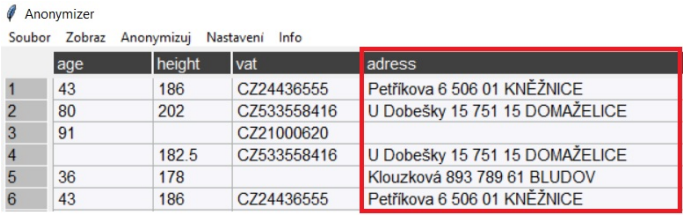
Obrázek 3.12: Ukázka anonymizovaných hodnot sloupce region, po použití metody permutace.

Na Obrázku 3.12 lze vidět anonymizovaný sloupec region za použití metody permutace. Tento výřez může být trochu zavádějící, a proto jej okomentuji. Vstupní soubor má tisíc řádků, a proto jsem pro demonstraci jednotlivých metod volila pouze výřezy prvních šesti řádků. Z toho vyplývá, že hodnoty, které jsou ukázány v Obrázku 3.11, byly přesunuty na jiné řádky, než prvních šest z výřezu. Například prázdná hodnota z řádku čtyři byla přesunuta na řádek 20, což v tomto výřezu nejde vidět. Stejné objasnění platí i u ostatních hodnot, které na první pohled nemusí být logicky vysvětleny.

3.4.4. Generalizace - zobecnění celé adresy na okres

Jak již vyplývá z názvu metody, cílem je náhrada celé adresy (ulice, číslo popisné, PSČ, obec) názvem okresu. Algoritmus této metody nejprve vyhledá v celé adrese PSČ, následně zjistí, kterým obcím toto PSČ přísluší. Pak ověří, zda se v celé adrese některá z vyhledaných obcí nachází. Pokud ano, použije kombinaci PSČ a obce k vyhledání odpovídajícího okresu v pomocném slovníku, který byl vytvořen na základě souboru **PSC_obce_okres_kraj.csv**.

Pokud nastane situace, že je ve sloupci prázdná hodnota nebo neplatná adresa, algoritmus toto pole nahradí obecnou hodnotou „okres“.


Anonymizer

Soubor Zobraz Anonymizuj Nastavení Info

	age	height	vat	adress
1	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE
2	80	202	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
3	91		CZ21000620	
4		182.5	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
5	36	178		Klouzková 893 789 61 BLUDOV
6	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE

Obrázek 3.13: Ukázka originálních hodnot sloupce adress, datového typu adresa.


Anonymizer

Soubor Zobraz Anonymizuj Nastavení Info

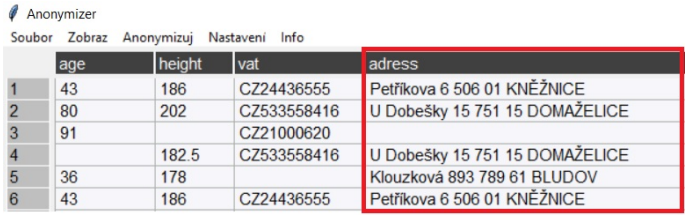
	age	height	vat	adress
1	(22.0;43.0>	185.17	EuÑaÚ|³¡_K	JIČÍN
2	(63.0;82.0>	208.1	1 6	PŘEROV
3	(82.0;100.0>		ŧ¼ëąq°P³	okres
4		181.73	1 6	PŘEROV
5	(22.0;43.0>	178.64	ë¹åa	ŠUMPERK
6	(22.0;43.0>	182.74	EuÑaÚ|³¡_K	JIČÍN

Obrázek 3.14: Ukázka anonymizovaných hodnot sloupce adress, po použití metody zobecnění celé adresy na okres.

3.4.5. Generalizace - zobecnění celé adresy na kraj

Algoritmus této metody je založen na stejném principu jako u zobecnění celé adresy na okres, pouze v pomocném slovníku je jako anonymizovaná hodnota použit název kraje.

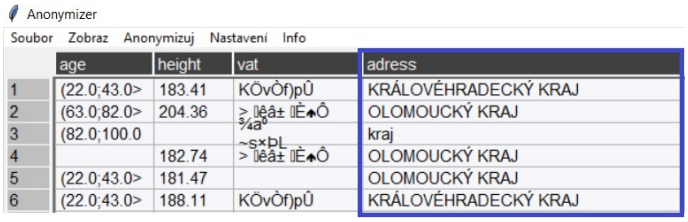
Poznámka. Protože názvy obcí, ulic, okresů či krajů mohou mít různou velikost písma. Někdy jsou psány všechny znaky velkým písmenem, jindy je pouze první písmeno velké. Proto jsou všechny pomocné slovníky převedeny na velké písmo. Před vyhledáváním v těchto pomocných slovnících je hledaný výraz převeden pomocí funkce **upper()** na text s velkým písmem. Toto platí i o slovnících pro vyhledávání jmen, přímení a dalších.


Anonymizer

Soubor Zobraz Anonymizuj Nastavení Info

	age	height	vat	adress
1	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE
2	80	202	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
3	91		CZ21000620	
4		182.5	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
5	36	178		Klouzková 893 789 61 BLUDOV
6	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE

Obrázek 3.15: Ukázka originálních hodnot sloupce adress, datového typu adresa.


Anonymizer

Soubor Zobraz Anonymizuj Nastavení Info

	age	height	vat	adress
1	(22.0;43.0>	183.41	KÖvÒf)pÛ	KRÁLOVÉHRADECKÝ KRAJ
2	(63.0;82.0>	204.36	> ¦êâ± ¦È♠Ô	OLOMOUCKÝ KRAJ
3	(82.0;100.0		¾a° ~s×bL	kraj
4		182.74	> ¦êâ± ¦È♠Ô	OLOMOUCKÝ KRAJ
5	(22.0;43.0>	181.47		OLOMOUCKÝ KRAJ
6	(22.0;43.0>	188.11	KÖvÒf)pÛ	KRÁLOVÉHRADECKÝ KRAJ

Obrázek 3.16: Ukázka anonymizovaných hodnot sloupce adress, po použití metody zobecnění celé adresy na okres.

3.4.6. Generalizace - agregace

Implementovanou metodu agregace lze využít pouze na číselné hodnoty. Tato metoda spočívá ve vytvoření určitého počtu skupin neboli intervalů, jimiž nahradí konkrétní originální hodnoty v anonymizovaném sloupci.

Algoritmus využívá knihovny **ckwrap** [29].

Počet skupin je definován na globální úrovni aplikace v dialogu **Výchozí nastavení aplikace** a to konkrétně **Agregace - počet skupin**. Počet skupin lze nastavit v rozsahu 3 až 10.

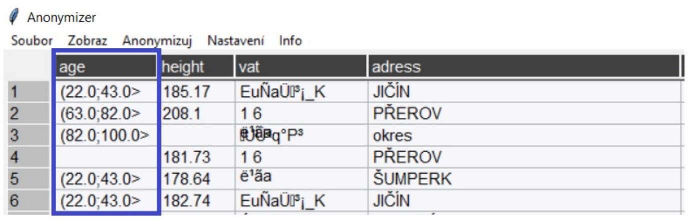
Agregaci provedeme voláním funkce **ckwrap.ckmeans(vsechny originalni hodnoty, pocet skupin)**. Výstupem funkce je seznam, který tvoří cílovou skupinu pro každou jednotlivou originální hodnotu.

Originální hodnotu algoritmus nahrazuje číselným intervalem, do kterého všechny hodnoty dané skupiny spadají.

V případě, že by se v originálních datech vyskytla kromě číselných hodnot také nečíselná hodnota, pak by bylo anonymizované pole nečíselné hodnoty prázdné. Pokud by ve sloupci nebyly žádné číselné hodnoty, pak jsou anonymizovaná pole vyplněna textem „fake“.

	age	height	vat	adress
1	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE
2	80	202	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
3	91		CZ21000620	
4		182.5	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
5	36	178		Klouzková 893 789 61 BLUDOV
6	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE

Obrázek 3.17: Ukázka originálních hodnot sloupce age, datového typu integer.



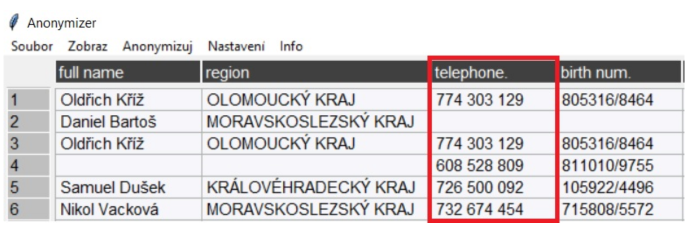
	age	height	vat	adress
1	(22.0;43.0>	185.17	EuÑaÜ▯³¡_K	JIČÍN
2	(63.0;82.0>	208.1	1 6	PŘEROV
3	(82.0;100.0>		[illegible]	okres
4		181.73	1 6	PŘEROV
5	(22.0;43.0>	178.64	ë¹ãa	ŠUMPERK
6	(22.0;43.0>	182.74	EuÑaÜ▯³¡_K	JIČÍN

Obrázek 3.18: Ukázka anonymizovaných hodnot sloupce age, po použití metody agregace do pěti skupin.

3.4.7. Maskování

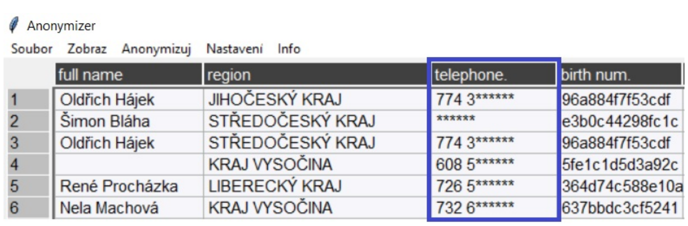
Cílem metody je nahradit konec anonymizovaného textového řetězce zvoleným počtem znaků hvězdička. Počet znaků se nastavuje opět v dialogu **Výchozí nastavení aplikace** parametrem **Velikost masky**. Tento parametr může být nastaven v rozsahu 3 až 6.

V situaci, kdy anonymizovaný řetězec je kratší než zvolená velikost masky, řetězec bude nahrazen počtem hvězdiček odpovídajícím parametru Velikost masky ve Výchozím nastavení aplikace.



	full name	region	telephone.	birth num.
1	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
2	Daniel Bartoš	MORAVSKOSLEZSKÝ KRAJ		
3	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
4			608 528 809	811010/9755
5	Samuel Dušek	KRÁLOVÉHRADECKÝ KRAJ	726 500 092	105922/4496
6	Nikol Vacková	MORAVSKOSLEZSKÝ KRAJ	732 674 454	715808/5572

Obrázek 3.19: Ukázka originálních hodnot sloupce telephone, datového typu telefon.



	full name	region	telephone.	birth num.
1	Oldřich Hájek	JIHOČESKÝ KRAJ	774 3******	96a884f7f53cdf
2	Šimon Bláha	STŘEDOČESKÝ KRAJ	******	e3b0c44298fc1c
3	Oldřich Hájek	STŘEDOČESKÝ KRAJ	774 3******	96a884f7f53cdf
4		KRAJ VYSOČINA	608 5******	5fe1c1d5d3a92c
5	René Procházka	LIBERECKÝ KRAJ	726 5******	364d74c588e10a
6	Nela Machová	KRAJ VYSOČINA	732 6******	637bbdc3cf5241

Obrázek 3.20: Ukázka anonymizovaných hodnot sloupce telephone, po použití metody maskovaní s velikostí masky šešt.

3.4.8. Pseudonymizace - SHA256 hash

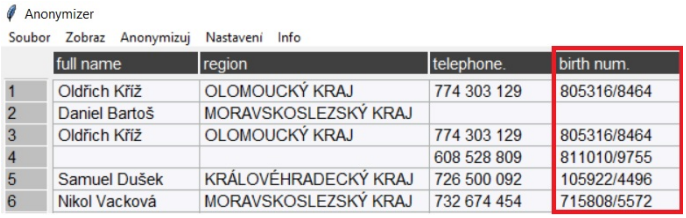
Pomocí hashovací funkce vytváříme takzvaný hash, což je někdy také nazýváno jako kontrolní součet vstupních dat. Hlavní vlastností kontrolního součtu je, že poměrně malá změna vstupních dat vede ke změně výsledného kontrolního součtu [25].

Většina bezpečnostních algoritmů se postupem času upřesňuje a vyvíjí. Pro demonstrační účely mého Anonymizéru jsem zvolila algoritmus SHA256.

Pro implementaci jsem využila existující knihovny **PyCryptodome** [26], konkrétně modul **SHA256** [28].

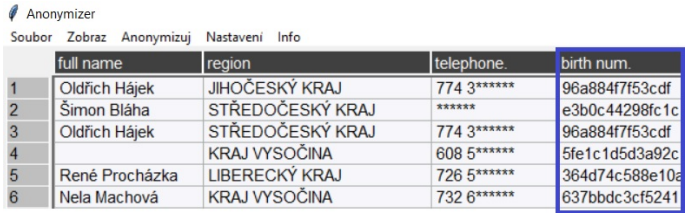
Prvním krokem algoritmu je převod originální hodnoty pomocí funkce **encode()** do kódovacího formátu, který vyžaduje hash funkce. Pomocí funkce **SHA256.new(originalni hodnota)** se získá ukazatel na vytvořený hash objekt. Hexadecimální podoba hash hodnoty se získá pomocí funkce **hexdigest()** .

Poznámka. V případě prázdné hodnoty pole se i přesto vytvoří anonymizovaná hodnota v podobě hash hodnoty. Jak vyplývá z popisu hash funkce, pro stejné vstupní hodnoty budou anonymizované hodnoty v podobě hash hodnoty stejné. Jak již bylo upozorněno v teoretické části, pokud je vstupní rozsah originálních dat malý, lze jednoduše vytvořit databázi hashovacích klíčů, poté je porovnat s anonymizovaným údajem, a tak údaje zpětně identifikovat.



	full name	region	telephone.	birth num.
1	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
2	Daniel Bartoš	MORAVSKOSLEZSKÝ KRAJ		
3	Oldřich Kříž	OLOMOUCKÝ KRAJ	774 303 129	805316/8464
4			608 528 809	811010/9755
5	Samuel Dušek	KRÁLOVÉHRADECKÝ KRAJ	726 500 092	105922/4496
6	Nikol Vacková	MORAVSKOSLEZSKÝ KRAJ	732 674 454	715808/5572

Obrázek 3.21: Ukázka originálních hodnot sloupce birth numb., datového typu rodné číslo.



	full name	region	telephone.	birth num.
1	Oldřich Hájek	JIHOČESKÝ KRAJ	774 3******	96a884f7f53cdf
2	Šimon Bláha	STŘEDOČESKÝ KRAJ	******	e3b0c44298fc1c
3	Oldřich Hájek	STŘEDOČESKÝ KRAJ	774 3******	96a884f7f53cdf
4		KRAJ VYSOČINA	608 5******	5fe1c1d5d3a92c
5	René Procházka	LIBERECKÝ KRAJ	726 5******	364d74c588e10a
6	Nela Machová	KRAJ VYSOČINA	732 6******	637bbdc3cf5241

Obrázek 3.22: Ukázka anonymizovaných hodnot sloupce birth numb., po použití metody SHA256 hash.

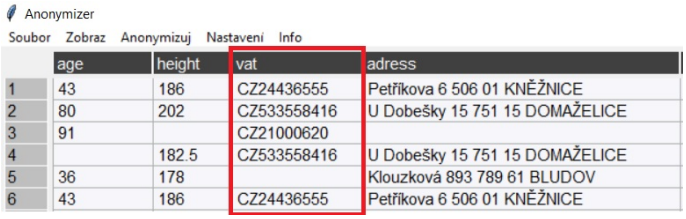
3.4.9. Pseudonymizace - AES šifra

Advancer Encryption Standard (AES) je jednou ze standardních metod šifrování dat v informatice.

Pro použití této metody jsem využila opět knihovnu **PyCryptodome**, modul **AES** [27].

Narozdíl od hashovací funkce, AES šifrování vyžaduje zadání šifrovacího klíče ve formátu textového řetězce definované délky. Před zašifrováním každé originální hodnoty je nutné vygenerovat unikátní náhodné číslo (nonce). Následně originální hodnotu převést na správný formát pomocí funkce **encode()**, podobně jako v případě hashovací funkce. Výsledkem šifrování pomocí funkce **cipher.encrypt_and_digest(originalni hodnota)** je dvojice hodnot. První hodnotou je ciphertext, což je samotná zašifrovaná hodnota, a druhou hodnotou je takzvaný tag, což je šifrovaný klíč, který by bylo možno použít spolu s hodnotou nonce pro případné zpětné rozšifrování. Tyto hodnoty (nonce, tag) však aplikace neukládá, protože z hlediska anonymizace je zpětné obnovení originálních hodnot nežádoucí.

Poznámka. V případě prázdné hodnoty pole zůstane i anonymizovavá hodnota pole prázdná. Narozdíl od hashovací funkce, šifra je pokaždé jiná i v případě stejných vstupních hodnot.



	age	height	vat	adress
1	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE
2	80	202	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
3	91		CZ21000620	
4		182.5	CZ533558416	U Dobešky 15 751 15 DOMAŽELICE
5	36	178		Klouzková 893 789 61 BLUDOV
6	43	186	CZ24436555	Petříkova 6 506 01 KNĚŽNICE

Obrázek 3.23: Ukázka originálních hodnot sloupce vat, datového typu DIČ.

	age	height	vat	adress
1	(22.0;43.0>	185.17	EuÑaÚ|³¡_K	JIČÍN
2	(63.0;82.0>	208.1	1 6	PŘEROV
3	(82.0;100.0>		ë¹ãa°P³	okres
4		181.73	1 6	PŘEROV
5	(22.0;43.0>	178.64	ë¹ãa	ŠUMPERK
6	(22.0;43.0>	182.74	EuÑaÚ|³¡_K	JIČÍN

Obrázek 3.24: Ukázka anonymizovaných hodnot sloupce vat, po použití metody AES šifra.

Závěr

Hlavním cílem této diplomové práce bylo vytvoření vlastního nástroje pro anonymizaci dat pomocí programovacího jazyka Python. Dílčími cíli proto byla jednak literární rešerše zákonných požadavků na ochranu dat a především pak samostudium programovacího jazyka Python. Ráda bych zde zmínila, že jsem Python znala jen velmi okrajově a nikdy jsem v něm před zadáním této diplomové práce nepracovala.

Teoretická část této práce obsahuje dvě kapitoly, kde jsem se snažila čtenáře uvést do problematiky ochrany osobních údajů a s tím související anonymizace dat, včetně zásad zpracování. Jak jsem již v textu vícekrát zmiňovala, určení cíle a metody anonymizace není vůbec lehký úkol. Existuje však několik doporučení, která mohou při volbě anonymizačního procesu pomoct. Této problematice je věnována podkapitola 2.5 Shrnutí a doporučení.

V rámci praktické části práce jsem se naopak snažila využít nově nabytých znalostí programovacího jazyka Python k vytvoření aplikace, jejíž fungování a principy jsem následně popsala v třetí kapitole.

Tato práce pro mě má velký přínos, a to především ve zvýšení úrovně programovacích dovedností, jelikož jsem doposud ani zdaleka nic podobně obsáhlého nenaprogramovala. Pominu-li pochopení principů algoritmizovaní a naučení se pracovat s jednotlivými strukturami, které Python nabízí, za jeden z největších oříšků tvorby aplikace považuji také nalezení kompatibility jednotlivých knihoven, včetně konvertování skriptu do EXE souboru.

Aplikace Anonymizér má zajisté prostor pro zlepšení, přesto jsem přesvědčena o tom, že je aplikace funkční a v určitých ohledech využitelná v praxi.

Vzhledem k možnosti využití Anonymizéru jsem se celou práci snažila zpracovat tak, aby byla přínosná i pro čtenáře bez větších programovacích znalostí.

Literatura

[1] EU: *Listina základních práv Evropské unie.*[online][cit. 2021-11-20]. dostupné z: https://eur-lex.europa.eu/legal-content/cs/ALL/?uri=CELEX:12012P/TXT

[2] EU: *Nařízení Evropského parlamentu a Rady (EU) 2016/679* .[online][cit. 2021-11-20]. dostupné z: http://data.europa.eu/eli/reg/2016/679/oj

[3] ČR: *Zákon č.101/2000 Sb.*[online],2021,[cit. 2021-11-20]. dostupné z: https://www.zakonyprolidi.cz/cs/2000-101

[4] CHLEBUS, T.; DOSTÁL, J.: *Nový zákon o zpracování osobních údajů.*[online]. 2019, [cit. 2021-11-20]. dostupné z: https://www.epravo.cz/top/clanky/novy-zakon-o-zpracovani-osobnich-udaju-109312.html

[5] Ministerstvo Vnitra České Republiky: *Ochrana osobních údajů.*[online][cit. 2021-11-20]. dostupné z: https://www.mvcr.cz/gdpr/clanek/gdpr-web-uvod-ochrana-osobnich-udaju.aspx

[6] Ministerstvo Vnitra České Republiky: *Zásady zpracování osobních údajů*[online][cit. 2021-11-20]. dostupné z: https://www.mvcr.cz/gdpr/clanek/zasady-zpracovani-osobnich-udaju.aspx

[7] Ministerstvo Vnitra České Republiky:*Co je GDPR.*[online][cit. 2021-11-20]. dostupné z: https://www.mvcr.cz/gdpr/clanek/co-je-gdpr.aspx

[8] Data Protection Commission: *Guidance Note:Guidance on Anonymisation ans Pseudonymisation.*[online]. 2019, [cit. 2021-11-20]. dostupné z: https://www.dataprotection.ie/sites/default/files/uploads/2019-06/190614%20Anonymisation%20and%20Pseudonymisation.pdf

[9] ARTICLE 29 DATA PROTECTION WORKING PARTY: *Opinion 05/2014 on Anonymisation Techniques.*[online]. 2014, [cit. 2021-11-20]. dostupné z: https://www.pdpjournals.com/docs/88197.pdf

[10] Úřad pro ochranu osobních údajů: *Metodika obecného posouzení vlivu na ochranu osobních údajů.*[online]. 2020, [cit. 2021-11-20]. dostupné z: https://www.uoou.cz/assets/File.ashx?id_org=200144&id_dokumenty=46487

[11] Narayanan, A.; Shmatikov, V.: *Robust De-anonymization of Large Sparse Datasets*[online]. 2008, [cit. 2021-11-20]. dostupné z: https://www.cs.utexas.edu/ shmat/shmat_oak08netflix.pdf

[12] EU: *ČSN ISO/IEC 20889 - Terminologie a klasifikace technik odstranění identifikace dat zvyšujících soukromí.*[online]. 2021, [cit. 2021-11-20]. dostupné z: https://www.nlfnorm.cz/terminologicky-slovnik/172203

[13] EU: *What is personal data?*[online]. [cit. 2021-11-20]. dostupné z: https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-personal-data

[14] Kristýna Delmar: *Únik dat: co při něm hrozí a jak se mu bránit?*2020, [cit. 2021-11-20]. dostupné z: https://www.fbadvokati.cz/cs/clanky/3474-unik-dat-co-pri-nem-hrozi-a-jak-se-mu-branit

[15] Algotech: *GDPR v praxi: Největší průšvihy a pokuty v historii.*[online].2020, [cit. 2021-11-20]. dostupné z: https://www.algotech.cz/novinky/2020-05-04-gdpr-v-praxi-nejvetsi-prusvihy-a-pokuty-v-historii

[16] Ministerstvo Vnitra České republiky: *Četnost jmen a příjmení.*[online].2007,[cit. 2021-11-20]. dostupné z: https://web.archive.org/web/20081208010154/http://www.mvcr.cz/clanek/cetnost-jmen-a-prijmeni.aspx

[17] Státní správa: *Kraje, okresy, obce.*[online].2021, [cit. 2021-11-20]. dostupné z: https://www.statnisprava.cz/rstsp/redakce.nsf/i/kraje_okresy_obce

[18] Wikipedie: *ASCII.*[online], [cit. 2021-11-20]. dostupné z: https://cs.wikipedia.org/wiki/ASCII

[19] Wikipedie: *List of Internet top-level domains.*[online], [cit. 2021-11-20]. dostupné z: https://en.wikipedia.org/wiki/List_of_internet_top-level_domains

[20] Wikipedie: *Domain name.*[online], [cit. 2021-11-20]. dostupné z: https://en.wikipedia.org/wiki/Domain_name

[21] Wikipedie: *"Neplatné uživatelské jméno"při pokusu o vytvoření uživatelského jména, které obsahuje speciální znak v Office 365*[online], [cit. 2021-11-20]. dostupné z: https://docs.microsoft.com/cs-cz/office/troubleshoot/office-suite-issues/username-contains-special-character

[22] Wikipedie: *IPv6*[online], [cit. 2021-11-20]. dostupné z: https://cs.wikipedia.org/wiki/IPv6r

[23] Wikipedie: *Data type.*[online], [cit. 2021-11-20]. dostupné z: https://en.wikipedia.org/wiki/Data_type

[24] Faker documentation: *Welcome to Faker's documentation!*[online].2014, [cit. 2021-11-20]. dostupné z: https://faker.readthedocs.io/en/master/index.html

[25] Wikipedie: *Secure Hash Algorithm*[online], [cit. 2021-11-20]. dostupné z: https://cs.wikipedia.org/wiki/Secure_Hash_Algorithm

[26] PyCryptodome : *Welcome to PyCryptodome's documentation*[online], [cit. 2021-11-20]. dostupné z: https://pycryptodome.readthedocs.io/en/latest/index.html

[27] PyCryptodome : *AES*[online], [cit. 2021-11-20]. dostupné z: https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html

[28] PyCryptodome : *SHA-256*[online], [cit. 2021-11-20]. dostupné z: https://pycryptodome.readthedocs.io/en/latest/src/hash/sha256.html

[29] PyPi : *ckwrap*[online], [cit. 2021-11-20]. dostupné z: https://pypi.org/project/ckwrap/

[30] Wikipedie : *Email adress*[online], [cit. 2021-11-20]. dostupné z: https://en.wikipedia.org/wiki/Email_address

Ukázky kódu

Příloha A

Obsah přiloženého CD

- Aplikace
  - build
  - dist
    - pomocne_csv
      - kraje.csv
      - krestni_muzi.csv
      - krestni_zeny.csv
      - obce.csv
      - predcisli.csv
      - prijmeni_muzi.csv
      - prijmeni_zeny.csv
      - PSC_obec_okres_kraj.csv
      - ulice.csv
      - zeme.csv
    - Anonymizer.exe ..... Aplikace Anonymizér
- Testovací data
  - TestovaciData.csv ..... Testovací data