Anonimizacija podataka u bazama podataka

Črepinko, Marko

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Črepinko

ANONIMIZACIJA PODATAKA U BAZAMA PODATAKA

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Črepinko

Matični broj: 45072/16 - R

Studij: Informacijski sustavi

ANONIMIZACIJA PODATAKA U BAZAMA PODATAKA

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Kornelije Rabuzin

Varaždin, rujan 2020.

Marko Črepinko

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Tema ovog rada nosi naziv „Anonimizacija podataka u bazama podataka”, a nastoji objasniti problematiku stvaranja anonimiziranih skupova podataka, tj. takvih skupova podataka koji predstavljaju vrlo mali rizik od identifikacije pojedinca na kojeg se zapisi skupa podataka odnose, čak i kada se zapisi jednog izvora spoje sa zapisima iz drugih izvora podataka. Također su predstavljene tehnike provođenja procesa anonimizacije, posebno uzimajući u obzir zakonske regulative te tehničke i organizacijske mjere koje pospješuju efektivnu provedbu anonimizacije i zaštite podataka. Kao primarne tehnike anonimizacije podataka odabrane su nasumično generiranje sintetičkih podataka te generiranje sintetičkih podataka uz očuvanje statističkih karakteristika izvornog skupa podataka. Na kraju su svi podaci pohranjeni u odgovarajuće baze podataka, u čijem su dizajnu implementirane važne tehničke i organizacijske mjere zaštite podataka i ograničenja prava pristupa, s obzirom na izvršitelja obrade podataka i razloge same obrade.

Ključne riječi: anonimizacija, zaštita podataka, privatnost, baza podataka, osobni podatak, izravan identifikator, neizravan identifikator, voditelj obrade, izvršitelj obrade

Sadržaj

1. Uvod....1
2. Metode i tehnike rada....2
3. Anonimizacija podataka....3
3.1. Pregled potrebnih definicija i pojmova....5
3.2. Nastavak rasprave o svrsi i potrebi za anonimizacijom....12
3.3. Pregled mogućih napada na skupove podataka....14
3.3.1. Napad na javno objavljene skupove podataka....15
3.3.2. Nenamjerna otkrivanja identiteta iz anonimiziranih skupova podataka....16
3.3.3. Zlonamjerna otkrivanja identiteta iz anonimiziranih skupova podataka....16
3.3.4. Povreda podataka....17
3.4. Zaključna razmatranja....17
4. Pregled osnovnih tehnika anonimizacije....19
4.1. Opis odabranog skupa podataka....19
4.1.1. Opis izvornog slučaja korištenja odabranog skupa podataka....19
4.1.2. Kratki opis atributa odabranog skupa podataka....21
4.2. Opis zamišljenog slučaja korištenja u implementaciji....22
4.3. Inicijalna priprema odabranog skupa podataka za anonimizaciju....23
5. Provedba anonimizacije za potrebe analitičkog i statističkog istraživanja....28
5.1. Početni koraci u anonimizaciji....30
5.1.1. Istraživanje atributa odabranog skupa podataka....33
5.2. Generiranje sintetičkih podataka uz očuvanje statističkih svojstava skupa podataka. 41
5.1.1. Posljedni korak u anonimizaciji....55
6. Provedba anonimizacije za potrebe razvoja, testiranja i održavanja aplikacijskog rješenja 59
6.1. Generiranje nasumičnih sintetičkih vrijednosti....61
7. Implementacija potrebnih baza podataka....66

8. Zaključak....80
9. Literatura....81
Popis slika....84
Popis tablica....86
Prilozi....87
Prilog 1 – Obrada atributa rizika....87
Prilog 2 – Obrada ostalih atributa....88
Prilog 3 – Izdvajanje i preslikavanje numeričkih atributa rizika realnog ili cjelobrojnog raspona vrijednosti s pripadnim tekstualnim oznakama....89
Prilog 4 – Problem malog broja zapisa....90
Prilog 5 – SQL definicije baza podataka, relacija i uloga....91

1. Uvod

U današnje informacijsko doba, motivacija za zaštitom podataka, a pritom i privatnosti pojedinca, nikada nije bila veća. Strahovite količine podataka koje se dnevno generiraju predstavljaju veliki rizik od otkrivanja osjetljivih podataka o pojedincu na kojeg se odnose, ponekad noseći sa sobom zaista zabrinjavajuće posljedice. Motivacija za skupljanjem tolikih podataka upravo leži u njihovoj obradi i informacijama, tj. znanju koje se obradom može otkriti. Te podatke je baš za te svrhe obrade često potrebno dijeliti, bilo javno ili međusobno između organizacija ili odijela istih, ali na način da se identitet pojedinca na kojeg se podaci odnose više ne može ustanoviti. Dakle, ti podaci trebaju biti anonimizirani, tj. pojedinca u skupu podataka činiti anonimnim, a posebno u slučajevima povezivanja zapisa iz više skupova podataka.

Zaštita podataka u mnogim je zemljama zakonski regulirana, poput: europske Opće uredbe za zaštitu podataka (*GDPR*) ili američkog Zakona o zdravstvenom osiguranju (*HIPAA*). Kako je ovaj rad pisan na području EU, tema je osobito sagledana sa pravnog stajališta Opće uredbe za zaštitu podataka, te organizacijskih mjera i preporuka zaštite podataka koje Uredba donosi. Dakle, sadržaj ovog rada opisuje holistički pristup anonimizaciji podataka, opisanog kroz zamišljeni slučaj korištenja kakav se može sresti u praksi. Autoru je cilj upoznati čitatelja s konceptom anonimizacije, osvještujući ga o mogućim rizicima nepotrebnog i nepromišljenog korištenja osobnih podataka, te tehničkim i organizacijskim mjerama zaštite identiteta pojedinca u skupu podataka, tj. provođenjem procesa anonimizacije nad odabranim skupom podataka. Sve veća potreba za poznavanjem tih tehnika rada, ali i brojnih izvora rizika za narušavanje privatnosti podataka, upravo su poslužili kao motivacija autoru za ovaj rad, stoga je njegova namjera potaknuti i čitatelja ovog djela da se i sam kasnije uputi u proučavanje ove vrlo važne tematike.

Tijek ovog djela započinje opisom korištenih metoda i tehnika rada, a nastavlja se uvodom u koncept anonimizacije, objašnjenjem definicija važnih za razumijevanje problematike koja se njome nastoji razriješiti, te stvaranjem intuicije oko potrebe provođenja ovog procesa. Potom slijedi kratki pregled najčešćih metoda anonimizacije. Na kraju pak dolazi praktična implementacija zamišljenog slučaja korištenja, tj. demonstracija implementacije tehničkih i organizacijskih mjera anonimizacije, te interpretacija dobivenih rezultata.

2. Metode i tehnike rada

U ovome će poglavlju biti opisani glavni programski alati korišteni pri izradi ovog rada.

Sam tekstualni sadržaj izrađen je u *LibreOffice Writer* uređivaču teksta, koji je dio *LibreOffice* paketa otvorenog koda, dok je kao sustav referenciranja korišten *Zotero*, također programski alat otvorenog koda. Za izradu isječaka kodova prikazanih u ovome radu, korištena je mrežna stranica *hilite.me*.

Praktični dio rada, najvećim je dijelom odrađen u razvojnom okruženju *JupyterLab*, inače jednom od najznačajnijih alata u podatkovnoj znanosti. Ovo je okruženje odabrano zbog vrlo fleksibilnih mogućnosti upravljanja izvođenjem programskog koda te zbog vrlo lakog dokumentiranja istog. Glavni programski jezik korišten za potrebe izrade praktičnog rješenja implementiranog u ovome radu jest skriptni programski jezik *Python*, verzije 3.8.5. Uz *Python*, kao glavni alat za obradu i analizu skupova podataka, korištena je *Pythonova* knjižnica *pandas*, verzije 1.1.0. Kao glavni alati korišteni za sam proces anonimizacije, poslužile su knjižnice *Faker* i *DataSynthesizer*. *Faker*-ov je zadatak bio generirati „lažne”, tj. sintetičke podatke poput adresa, adresa e-pošte, poštanskih brojeva, spolova, itd. Dok je *DataSynthesizer*-ov zadatak bio najprije analizirati statističke karakteristike skupa podataka, te potom prema izrađenom modelu generirati sintetičke podatke, tj. novi skup podataka koji je zadržao statistička svojstva izvornog skupa podataka.

Kao sustav za upravljanje bazom podataka, kraće SUBP, korištenom u implementaciji praktičnog rješenja ovog rada, korišten je *PostgreSQL*, verzije 12.3. Za potrebe anonimiziranja podataka izravno u implementiranim bazama podataka, korišten je *PostgreSQL* programski dodatak *PostgreSQL Anonymizer*, verzije 0.6.0. Taj programski dodatak pruža vrlo korisne funkcionalnosti za potrebe anonimizacije podataka i implementiranja prava pristupa istima.

Sva su programska rješenja korištena u ovome radu instalirana i pogonjena na operacijskom sustavu *Manjaro Linux* verzije 20.1, Linux jezgre verzije *5.7.19-2-MANJARO*.

3. Anonimizacija podataka

Godine 1997., američke su istraživačice u području zaštite podataka, Latanya Sweeney i Pierangela Samarati, preklapanjem podataka iz dva različita skupa podataka uspješno identificirale koji se od zapisa tih skupova podataka odnose na tadašnjeg guvernera savezne američke države Massachusetts – Williama Welda [1, p. 2]. Značajnost otkrića podataka o spomenutom guverneru leži u činjenici kako su korišteni skupovi podataka navodno bili anonimizirani, tj. nisu sadržavali nikakve podatke koji bi i na koji način trebali moći izravno i nedvosmisleno identificirati pojedinca na kojeg se neki zapis odnosi, poput imena i prezimena ili pak broja socijalnog osiguranja (eng. *Social Security number*), što bi bio ekvivalent hrvatskom osobnom identifikacijskom broju (OIB) ili jedinstvenom matičnom broju građana (JMBG). Konkretno, spomenuti su skupovi podataka predstavljali popise posjeta bolnicama, tj. podatke o ambulantnim pregledima, te skup podataka o registracijama glasača grada Cambridgea. U svome su izvornom radu [2], spomenute istraživačice ovim postupkom demonstrirale tzv. identifikaciju povezivanjem (eng. *Re-identificaiton by linking*). Kao što je spomenuto, iz korištenih su skupova podataka otklonjeni svi podaci koji bi mogli izravno identificirati pojedinca. Tako je skup podataka o ambulantnim pregledima, između ostalih, sadržavao sljedeće podatke: poštanski broj (ZIP), datum rođenja, spol, etnicitet, datum posjete bolnici, dijagnozu, itd. S druge je pak strane, skup podataka o registracijama glasača sadržavao, između ostalih, sljedeće podatke: datum rođenja, spol, poštanski broj, ime, adresu, datum registracije, političku opredijeljenost, datum posljednjeg glasanja, itd. Skup atributa koji predstavljaju presjek ovih dvaju skupova podataka je očit: podaci o poštanskom broju, datumu rođenja i spolu iz oba skupa mogu omogućiti povezivanje pripadnih zapisa iz oba skupa podataka i tako pomoći pri otkrivanju osobnih i osjetljivih informacija o pojedincu. Iz skupova podataka o glasačima, u ovome se primjeru mogu otkriti osobni podaci o spomenutom guverneru poput: imena, točne adrese, datuma rođenja i spola. No, ti osobni podaci, iako otkriveni, samo otkrivaju članstvo neke osobe u pripadnom skupu podataka, što samo po sebi i ne mora biti od pretjerane važnosti ili brige. U vrijeme ovog pokusa, guverner Weld bio je punoljetna osoba s pravom glasanja, stoga činjenica da se podaci o njemu nalaze u skupu podataka o glasačima zaista nije začuđujuća. No, ti osobni podaci zapravo služe za povezivanje osoba s onim osjetljivim podacima, čije bi povezivanje s pripadnim pojedincem istome moglo donijeti, npr., osudu društva ili uže okoline zbog političkog opredjeljenja, sramotu zbog rijetke ili sramotne bolesti ili neki drugi oblik stigme. Sada je jasno kako su spomenuti skupovi podataka zapravo bili daleko od anonimiziranih, tj. omogućili su identifikaciju pojedinca čiji su se podaci nalazili u

njihovim zapisima. No, prije detaljnijeg objašnjenja koncepta anonimizacije, potrebno se zapitati zašto su uopće ovi skupovi podataka bili objavljeni. Sasvim je razumno kako su podaci o liječenjima, tj. zdravstveni podaci, itekako korisni za brojna medicinska istraživanja koja mogu uvelike ubrzati napredak i kvalitetu života neke zemlje ili čak pomoći i pri globalnom unaprjeđenju ljudskog zdravlja. Kao što je navedeno u [3], medicinska istraživanja pružaju uvid u kretanja razvoja bolesti i pripadnih faktora rizika, uzorke u liječenjima i pripadne troškove, kvalitetu i efikasnost zdravstvenog sustava, dok dostupnost stvarnih zdravstvenih podatka omogućuje različite pristupe istraživanjima, koji su pak bitni za evaluaciju rezultata tih istraživanja. Danas, kako je ljudska vrsta svoje živote i svakodnevicu učinila ovisnom o digitalnim uređajima, globalizaciji i trenutnoj dostupnosti informacija, u svijetu se generira više podataka no ikada prije. Povećanje procesorske moći suvremenih računala, pojava sve jeftinijih sustava za pohranu podataka i sveopća umreženost, omogućili su pojavu velikih podataka (eng. *Big Data*) i popratnih metoda obrade istih, poput strojnog učenja (eng. *Machine learning*), rudarenja podataka (eng. *Data mining*) i umjetne inteligencije (eng. *Artificial Intelligence*). Kroz primjene navedenih metoda podataka, u svijetu se stvara još više podataka, što opet dovodi do brojnih prilika za nova istraživanja. U [4] je navedeno predviđanje kako se do 2025. godine, očekuje da će se u svijetu godišnje generirati do 175 zetabajta (ZB) podataka, odnosno 175 bilijuna gigabajta. Kao glavni izvori stvaranja ove kolosalne količine podataka, navode se, između ostalih i spomenute tehnike obrade podataka, ali i posljedice stvaranja tehnologija autonomnih vozila. Također se navode i mutlimedijski sadržaji visoke rezolucije, 5G telekomunikacije, ali i za temu anonimizacije najznačanije – uređaji povezani pomoću interneta stvari (eng. *Internet of Things,* kraće IoT), s posebnim naglaskom na razne senzore, smještene u naša tijela, domove, tvornice, gradove, itd. Problem koji se ovdje nazire jest očuvanost privatnosti podataka koji se svakodnevno sakupljaju kao uzrok i posljedica ljudskog suživota s modernom tehnologijom. Nije pogrešno pretpostaviti kako od sakupljanja tih podataka ne možemo pobjeći, pogotovo ako želimo koristiti modernu tehnologiju i usluge koje ona omogućuje, no pitanje koje se nameće jest koliko se zapravo osobnih informacija nalazi u tim podacima. Konkretno, podaci sakupljeni pomoću senzora u pametnim domovima, mogli bi dovesti do otkrivanja brojnih informacija o dnevnim navikama ukućana, tj. do izrade profila osoba. Ti bi podaci, baš kao i oni iz zdravstvenih izvještaja, bili od ogromne koristi pri nekom znanstvenom istraživanju, no mogli bi također biti i od velike štete na račun pojedinca čija se intima može komprimirati u slučaju neovlaštenog odvajanja podataka, kolokvijalno zvanog „curenja podataka” (eng. *Data leakage*). Sličan se rizik ostvario i u primjeru otkrivanja zapisa koji se odnose na guvernera Welda, gdje nije došlo do curenja podataka, već do ciljanog identificiranja pojedinca, gdje se to zbog navodne anonimiziranosti skupova podataka nije smjelo dogoditi. Kada bi, na

primjer, guverner Weld živio u pametnoj kući, a podaci sa senzora iste bili javno objavljeni, također bez podataka o imenu i prezimenu ili nekom drugom podatku koji bi guvernera mogao izravno identificirati, guvernerovo ljudsko pravo na privatnost također bi bilo narušeno budući da bi se kombinacijom podataka iz drugih skupova guverner mogao ponovno identificirati. Dakle, problem kojeg anonimizacija pokušava ***umanjiti*** jest mogućnost identificiranja, odnosno prepoznavanja, pojedinca iz nekog skupa podataka ili iz kombinacije više skupova podataka.

Spomenuto pravo privatnosti štićeno je zakonom brojnih država, a između ostaloga i Europske unije. U slučaju EU, to je pravo deklarirano stavkom 1. članka 8. *Povelje Europske unije o temeljnim pravima*. Stoga u kontekstu prava privatnosti valja razmišljati i o zaštiti osobnih podataka koje svakodnevno upotrebljavamo, bilo navođenjem istih u službenim dokumentima i okruženju ili pak, npr.,, u situacijama prijave na neku mrežnu stranicu. Zakonska osnova zaštite osobnih podataka tek počinje biti predmetom usvajanja brojnih država. EU je u ovome području napravila značajan iskorak donošenjem *Opće uredbe o zaštiti podataka* (eng. *General Data Protection Regulation,* kraće *GDPR*), kraće OUZP. OUZP je detaljno objašnjen u poglavlju 4 ovog rada, no zasad je potrebno istaknuti sadržaj uvodne izjave 1, u kojoj je istaknuto da svatko ima pravo na zaštitu svojih osobnih podataka. Razlog navođenja ove zakonske potpore je isticanje važnosti zaštite podataka za kojom nikada nije postojala veća potreba, a u svijetlu te potrebe javlja se i potreba za anonimizacijom podataka. Kako je demonstrirano na spomenutom primjeru iz [1], ta se potreba očituje u mogućnosti, odnosno riziku izdvajanja i prepoznavanja zapisa o osobama iz skupova podataka koji sadržavaju bilo kakve osjetljive podatke, a pri čemu bi identitet pojedinaca trebao biti zaštićen.

Nakon ovog kratkog uvoda u koncept anonimizacije, slijedi objašnjenje potrebnih definicija, a potom i detaljno objašnjenje problematike koju anonimizacija nastoji razriješiti.

3.1. Pregled potrebnih definicija i pojmova

U ovom će poglavlju biti predstavljene i objašnjene definicije potrebne za bolje razumijevanje koncepta i procesa anonimiziacije podataka, pravne osnove uz koje se veže, organizacijske uloge koje su zadužene za provedbu procesa anonimizacije i zaštite podataka, te tehnički pojmovi potrebni za bolje razumijevanje praktičnog dijela ovog rada. Većina je predstavljenih definicija u ovom poglavlju preuzeta iz OUZP-a zbog opće prihvaćenosti te regulative.

Najprije je potrebno objasniti pojam skupa podataka. ***Skup podataka*** (eng. *Dataset*) najlakše je prikazati pomoću tablice.

Tablica 1: Prikaz strukturiranog skupa podataka

Ime pacijenta	Prezime pacijenta	Dijagnoza	Datum posjete
Mirko	Mirkić	Prehlada	01/01/20
Ana	Anastazić	Karcinom jetre	02/01/20
Iva	Ivić	Suhi kašalj	14/03/20

Tablica 1 prikazuje primjer strukturiranog skupa podataka koji sadrži zapise o pacijentovim posjetima liječniku. Skup podataka moguće je usporediti sa pojmom ***entiteta*** iz teorije baza podataka. Rabuzin u [5] pojam entiteta, u kontekstu ERA modeliranja, opisuje kao objekte o kojima prikupljamo podatke, poput kupaca, studenata, zaposlenika, itd. Entitet bi u ovom slučaju bio posjet pacijenta bolnici, a jedan redak ove tablice predstavljao bi jednu instancu tog entiteta, tj. podatke o jednoj posjeti. Jedna instanca entiteta, odnosno jedan redak skupa podataka, sadrži više različitih podataka koji se odnose na tu instancu. U ovom su to slučaju sljedeći podaci: „Ime pacijenta", „Prezime pacijenta", „Dijagnoza" i „Datum posjete". Takve podatke nazivamo ***atributima***, odnosno svojstvima „entiteta koja nas zanimaju i čije vrijednosti želimo pohraniti...". Ono što je još bitno za istaknuti jest kako „atributi obično poprimaju vrijednosti iz neke domene". [5] Na primjer, za atribut „Ime pacijenta" može se reći kako poprima vrijednosti iz domene znakovnih nizova, dok atribut „Datum posjete" poprima vrijednosti iz domene datumskih vrijednosti. Dakle, neki skup podataka ***D*** sastoji se od konačnog skupa atributa $\{A_1, ..., A_n\}$, dok n-torke $(a_1, ..., a_n)$ predstavljaju vrijednosti tih atributa, tj. pojedine retke skupa podataka. U nastavku će rada retci skupa podataka biti nazivani ***zapisima***, a bitno je još napomenuti kako se obično jedan zapis iz skupa odnosi na jednu osobu. Također, ovakav opis skupa podataka zapravo predstavlja strukturirani skup podataka, obzirom na to što se svaki zapis sastoji od vrijednosti svih atributa iz skupa $\{A_1, ..., A_n\}$, tj. kroz zapise skupa podataka ne pojavljuju se novi ili brišu postojeći atributi. Prije nastavka izlaganja rada, autor napominje da će u nadolazećim poglavljima umjesto izraza „skup podataka" biti korišten engleski naziv „dataset".

Dakle, kada se proces anonimizacije sprovodi nad nekim skupom podataka, anonimizacija se obavlja upravo nad vrijednostima atributa tog skupa, no ne nužno i nad vrijednostima svih atributa. No, više o tome slijedu u nastavku.

Kako bi se proces anonimizacije što bolje opisao, potrebno je opisati nad kakvom se vrstom podataka taj proces primjenjuje, s obzirom na odnos tih podataka s pojedincem na kojeg se oni odnose. Vrsta podataka nad kojima se anonimizacija primjenjuje već je ranije spomenuta, a radilo se o osobnim podacima. U [6, str. 33] (OUZP), ***osobni podaci*** definirani su stavkom 1. članka 4. kao

„svi podaci koji se odnose na pojedinca čiji je identitet utvrđen ili se može utvrditi („**ispitanik**”)*;* pojedinac čiji se identitet može utvrditi jest osoba koja se može identificirati izravno ili neizravno, osobito uz pomoć **identifikatora** kao što su ime, identifikacijski broj, podaci o lokaciji, mrežni identifikator ili uz pomoć jednog ili više čimbenika svojstvenih za fizički, fiziološki, genetski, mentalni, ekonomski, kulturni ili socijalni identitet tog pojedinca.”

Dakle, prema navedenoj definiciji, osobni podaci mogu identificirati pojedinca, što je vrlo bitna karakteristika tih podataka za provođenje procesa anonimizacije, s obzirom na to što se anonimizacijom to svojstvo nastoji umanjiti u što većoj mjeri. Dodatno, gornjom se definicijom definira i ***ispitanik***, odnosno pojedinac za kojeg postoje osobni podaci koji bi ga mogli identificirati. Također je potrebno opisati i jednu posebnu skupinu osobnih podataka koja će u ovome radu biti nazivana ***osjetljivim podacima***. Toj skupini podataka pripadaju svi oni podaci koji su vrlo osobne, intimne prirode i pojedincu mogu nanijeti štetu u smislu, npr., javne sramote ili osude. Ova skupina osobnih podataka obuhvaća podatke poput podataka o zdravlju pojedinca, seksualnoj orijentaciji, vjerskome ili političkom opredjeljenju, podataka o djeci (maloljetnicima) i sl. Ova je skupina od iznimne važnosti za proces anonimizacije jer se upravo iz tih podataka često izvlači najviše novih informacija, kao što će biti demonstrirano u praktičnom dijelu rada.

Navedena definicija vrlo jasno izriče skupine podataka koje se smatraju osobnim, a svima im je zajedničko što pojedinačno ili u kombinaciji mogu dovesti do identificiranja pojedinca na kojeg se odnose. No, ove se skupine podataka mogu razvrstati u više različitih kategorija, s obzirom na to kakvu vrstu identifikatora predstavljaju. Konkretno, za potrebe anonimizacije, bitno je razlikovati izravne i neizravne, tzv. kvazi-, identifikatore. Autori Arbuckle i El Emam u [7] kao izravne identifikatore navode podatke poput: imena, telefonskog broja, adrese e-pošte, identifikacijskog broja, broja zdravstvenog kartona, broja socijalnog osiguranja, itd. Pod neizravne identifikatore svrstavaju podatke poput: spola, datuma rođenja, adrese, poštanskog broja, geo lokacijskih podataka, podataka o blizini kulturnih znamenitosti, etnicitet, domorodački identitet, podaci o obrazovanju, bračno stanje, podaci o prihodima, govorni jezik, zanimanje, kodova medicinskih dijagnoza, zemlje rođenja, iznosa kilaže po rođenju, datuma bitnih događaja, itd. Isti autori u [7] ***izravne identifikatore*** opisuju kao podatke koji se izravno mogu koristiti za jedinstvenu identifikaciju pojedinca, dok

u [8] ***neizravne identifikatore*** opisuju kao podatke koji se u kombinaciji s drugim neizravnim identifikatorima koriste za identifikaciju pojedinca.

Potonja definicija iz [6] i navedeni opisi identifikatora navode tek malen dio podataka koji mogu poslužiti pri identificiranju pojedinca, a većina tih podataka dnevno se generira, odnosno može se generirati i sakupljati na pametnim telefonima. Autori u [9] istražuju kakvi se sve podaci mogu sakupljati pomoću pametnih telefona u svrhe psiholoških istraživanja. Između ostalog, navode mogućnosti praćenja svakodnevnog ponašanja korisnika, potom podatke o društvenim interakcijama, veličini društvenih grupa u kojima se osoba kreće, pozivima i tekstualnim porukama, dnevnim aktivnostima poput vremena spavanja i tjelovježbe, lokacijskih podataka, podataka s raznih senzora poput svjetlosnog senzora ili senzora blizine (eng. *Proximity sensor*) i brojnih drugih. Osnovno pitanje koje se ovdje nameće jest kako ispravno zaštiti te podatke, ne samo prilikom obrade istih, već i prilikom njihovog sakupljanja na uređaju korisnika. Anonimizacija također i u ovome slučaju može biti od pomoći, no za sad je bitno razumjeti nad kakvim se sve podacima primjenjuje postupak anonimizacije.

Sakupljanje svih tih podataka ne bi imalo smisla bez da se ti podaci kasnije ne obrađuju, bilo sa svrhom vođenja evidencije korisnika neke usluge, statističke obrade podataka, primjene tehnika strojnog učenja za otkrivanje znanja u podacima ili pak za potrebe istraživanja navika ponašanja pojedinaca. Tako je u [6, p. 33] dana sljedeća definicija obrade podataka:

„”**obrada**” znači svaki postupak ili skup postupaka koji se obavljaju na osobnim podacima ili na skupovima osobnih podataka, bilo automatiziranim bilo neautomatiziranim sredstvima kao što su prikupljanje, bilježenje, organizacija, strukturiranje, pohrana, prilagodba ili izmjena, pronalaženje, obavljanje uvida, uporaba, otkrivanje prijenosom, širenjem ili stavljanjem na raspolaganje na drugi način, usklađivanje ili kombiniranje, ograničavanje, brisanje ili uništavanje”.

Također, OUZP zahtjeva da se za svaku vrstu obrade osobnih podataka, od ispitanika traži privola, koja je stavkom 11. članka 4. definirana kao:

„”**privola**” ispitanika znači svako dobrovoljno, posebno, informirano i nedvosmisleno izražavanje želja ispitanika kojim on izjavom ili jasnom potvrdnom radnjom daje pristanak za obradu osobnih podataka koji se na njega odnose”. [6, p. 34]

Iznimno je bitno što je definicija obrade podataka definirana od strane zakona, upravo zbog što jasnijeg razumijevanja slučajeva u kojima se ti isti podaci moraju štiti, a pritom i identitet pojedinaca na koje se odnose. Zakonom su također definirane i četiri vrlo važne

uloge u procesu obrade podataka. Što je također od velike važnosti, obzirom na to što se podatak kroz svoj životni vijek, od nastojanja pa do brisanja, provlači kroz nekoliko faza i može dostići više destinacija. Stoga je vrlo bitno definirati uloge koje po definiciji imaju interakciju s podacima. Slijede definicije tih uloga iz [6, p. 33 - 34]:

- **voditelj obrade** - „fizička ili pravna osoba, tijelo javne vlasti, agencija ili drugo tijelo koje samo ili zajedno s drugima određuje svrhe i sredstva obrade osobnih podataka; kada su svrhe i sredstva takve obrade utvrđeni pravom Unije ili pravom države članice”
- **izvršitelj obrade -** „fizička ili pravna osoba, tijelo javne vlasti, agencija ili drugo tijelo koje obrađuje osobne podatke **u ime voditelja obrade**”
- **primatelj -** „fizička ili pravna osoba, tijelo javne vlasti, agencija ili drugo tijelo kojem se otkrivaju osobni podaci, neovisno o tome je li on treća strana. Međutim, tijela javne vlasti koja mogu primiti osobne podatke u okviru određene istrage u skladu s pravom Unije ili države članice ne smatraju se primateljima”
- **treća strana** - „znači fizička ili pravna osoba, tijelo javne vlasti, agencija ili drugo tijelo koje nije ispitanik, voditelj obrade, izvršitelj obrade ni osobe koje su ovlaštene za obradu osobnih podataka pod izravnom nadležnošću voditelja obrade ili izvršitelja obrade”

OUZP definira i jednu posebnu vrstu obrade podataka koja je naizgled vrlo slična procesu anonimizacije, ali kao rezultat daje skup podataka u kojem se osobni podaci, iako izmijenjeni, mogu ponovno povezati s pojedincem na kojeg se odnose. ***Pseudonimizacija*** je u [6, p. 33] definirana kao:

"obrada osobnih podataka na način da se osobni podaci više ne mogu pripisati određenom ispitaniku bez uporabe dodatnih informacija, pod uvjetom da se takve dodatne informacije drže odvojeno te da podliježu tehničkim i organizacijskim mjerama kako bi se osiguralo da se osobni podaci ne mogu pripisati pojedincu čiji je identitet utvrđen ili se može utvrditi”.

Primjer pseudonimizacije najlakše je objasniti zamišljanjem tablice, poput tablice 1, ali u kojoj su svi izravni identifikatori zamijenjeni nekim pseudonimom, kao što je npr. nasumični niz znakova. Posljedica ovakvog postupka pseudonimizacije jest dodatan skup podataka, koji zapravo predstavlja šifrarnik s podacima o tome koji se niz nasumičnih brojeva odnosi na koju osobu. Tablica 2 predstavlja jedan redak iz tablice 1, nakon provođenja procesa pseudonimizacije, a tablice 3 predstavlja spomenuti šifrarnik.

Tablica 2: Primjer pseudonimiziranog skupa podataka

Pseudonim	Dijagnoza	Datum posjete
220698	Prehlada	01/01/20

Tablica 3: Primjer šifrarnika korištenog pri pseudonimizaciji podataka

Pseudonim	Ime pacijenta	Prezime pacijenta
220698	Mirko	Mirkić

U navedenoj definiciji pseudonimizacije stoji i zahtjev da se dodatne informacije, uz pomoć kojih je jedino moguće ponovno identificirati pojedinca, drže odvojeno, a tehničke i organizacijske mjere kojima se to postiže mogu se ogledati u pohrani takvih informacija u zaštićenom okruženju s vrlo ograničenim pravima pristupa. Takvo okruženje mogla bi predstavljati baza podataka s tablicom koja predstavlja opisani šifrarnik u kojem su podaci dodatno kriptirani, s vrlo ograničenim pravima pristupa. U [10] je navedeno kako se pseudonimizacijom rizik od identifikacije pojedinaca minimizira na dva načina: 1) zbog same činjenice da se tablica koja predstavlja šifrarnik drži na odvojenom mjestu, ili je čak i uništena i 2) šteta koja nastaje pri mogućem komprimiranju skupa podataka u kojem su osobni podaci zamijenjeni pseudonimima, znatno je manja o štete koja nastaje u situaciji komprimiranja tablice koja predstavlja šifrarnik. Osobni se podaci mogu pseduonimizirati mehanizmima kriptiranja, na način da ključ korišten prilikom kriptiranja zadržava osoba koja je obavila kriptiranje; ili pak korištenjem funkcije sažimanja (eng. *Hashing function*). U oba su slučaja potrebne dodatne tablice koje omogućuju spajanje pseudonima s izvornim vrijednostima identifikatora.

Dakle, kako je pseudonimiziranim podacima ipak moguće utvrditi identitet pojedinca povezivanjem pseudonima i izravnih identifikatora, ili komprimiranjem podatka, oni se i dalje trebaju smatrati osobnim podacima, kao što je navedeno uvodnom izjavom 26. u [6, p. 5]. No, iako je tome tako, pseudonimizacija i anonimizacija predstavljaju procese kojima se nastoji umanjiti posljedica povrede osobnih podataka. Stoga, prije navođenja definicije anonimizacije, potrebno je još navesti definiciju ***povrede osobnih podataka***, koja je u [6, p. 34] definirana kao:

„kršenje sigurnosti koje dovodi do slučajnog ili nezakonitog uništenja, gubitka, izmjene, neovlaštenog otkrivanja ili pristupa osobnim podacima koji su preneseni, pohranjeni ili na drugi način obrađivani".

Do sada su pružene sve najbitnije definicije i objašnjenja pojmova potrebnih za razumijevanje procesa i rezultata anonimizacije podataka, ali i osoba zaduženi za njegovo sprovođenje, a početak ovog poglavlja trebao je stvoriti početnu intuiciju problematike koja se rješava ovim procesom. Stoga je preostalo definirati proces anonimizacije. Jedna od najjednostavnijih definicija dana je u [11] (ISO standard 25237:2017), a glasi:

„**anonimizacija je proces, izvršen samostalno od strane voditelja obrade ili u suradnji s drugima, kojim se osobni podaci nepovratno mijenjaju na način da se pojedinac na kojeg se ti podaci odnose više ne može identificirati, bilo na izravan ili neizravan način**".

Nadalje, OUZP-om je u uvodnoj izjavi 26. [6, p. 5], za anonimizaciju je navedeno sljedeće:

„...Načela zaštite podataka stoga se ne bi trebala primjenjivati na anonimne informacije, odnosno informacije koje se ne odnose na pojedinca čiji je identitet utvrđen ili se može utvrditi ili na osobne podatke koji su učinjeni anonimnima na način da se identitet ispitanika ne može ili više ne može utvrditi. Ova se Uredba stoga ne odnosi na obradu takvih anonimnih informacija, među ostalim za statističke ili istraživačke svrhe."

Ovime je zapravo objašnjen rezultat procesa anonimizacije, a to je anoniman skup podataka koji sadrži zapise o pojedincima, koji ne mogu dovesti do njihove identifikacije.

No, kako će kasnije u radu biti demonstrirano, proces anonimizacije zapravo se ne treba smatrati procesom koji daje jedan savršeni rezultat. Razlog tome jest činjenica, kako se povećanjem anonimnosti podataka povećava i privatnost, no to pak onda rezultira smanjenjem iskoristivosti podataka. To se smanjenje iskoristivosti manifestira kroz promjene podataka koje mogu u velikoj mjeri narušiti statistička obilježja skupa podataka koja se očitavaju u iznosima vrijednosti njegovih atributa i njihovom međudjelovanju. Stoga se rezultat procesa anonimizacije zapravo nalazi u rasponu između potpuno anonimiziranog i za obradu neiskoristivog skupa podataka na jednom kraju, te neanonimiziranog i za obradu u potpunosti iskoristivog skupa podataka. Ranije je već navedeno kako anonimizacija nastoji umanjiti rizik od razotkrivanja identiteta pojedinca, a taj cilj anonimizacije na vrlo dobar način opisuje i definicija dana američkim Zakonom o zdravstvenom osiguranju (eng. *Health Insurance Portability and Accountability Act*, skraćeno HIPAA), koja kaže da tzv. metoda stručne procjene (eng. *Expert Determination*), za uspješnu anonimizaciju podataka zahtjeva osobu upoznatu s općim statističkim i znanstvenim metodama, koja će anonimizaciju izvršiti

s takvim rezultatom da je rizik od utvrđivanja identiteta pojedinca vrlo malen. [12] Dodatno je još navedeno kako **rizik od utvrđivanja identiteta pojedinca u anonimiziranom skupu podataka uvijek postoji i da on nikad ne može biti jednak nuli, tj. ne postojati.**

Više o tome slijedi u nastavku, gdje se detaljnije istražuje problematika koja se rješava anonimizacijom podataka.

3.2. Nastavak rasprave o svrsi i potrebi za anonimizacijom

Definicija anonimizacije iz [12] zapravo opisuje srž tog procesa, a kad god se raspravlja o rizicima, od velike je važnosti razumjeti i njihove izvore. U primjeru izdvajanja pojedinca povezivanjem iz [1], bitan rizik predstavlja činjenica kako su loše anonimizirani osobni podaci bili javno objavljeni te im je svatko mogao pristupiti. Važnost objavljivanja zdravstvenih i drugih vrsta podataka već je ranije objašnjena – takvi se skupovi podataka ne mogu samo pohraniti u neki idealno osigurani sustav pohrane i zauvijek zaboraviti. Mogućnosti obrade i objavljivanje rezultata istraživanja predstavljaju svrhu javnog dijeljenja bilo kakvih skupova podataka, a jasno je kako su nova i kontinuirana istraživanja temelj svakog napretka u bilo kojem području ljudskog djelovanja. Svakodnevno generirane ogromne količine podataka plod su suvremenog čovjeka koji u svome evolucijskom razvoju sve više ovisi o obradi podataka zbog otkrivanja novih znanja. Autori u [8], stoga napominju da je temelj tih aktivnosti temeljenih na podacima (eng. data-driven) upravo njihovo odgovorno korištenje i dijeljnje. Odgovornim korištenjem i dijeljenjem, poštuje se čovjekovo pravo privatnosti, no autori u nastavku napominju kako razvojem tehnologije nastaju i nove prijetnje anonimiziranim skupovima podataka, što znači da nastaju i novi, sve veći rizici od povreda osobnih podataka. Neovlašteni pristupi podacima, javna dijeljenja podataka, kako od strane privatnih organizacija, tako i od strane državnih statističkih zavoda, potom dijeljenja podataka među odijelima organizacije te između poslovnih partnera, samo su neki od slučajeva gdje je anonimnost podataka od velike važnosti, a pogotovo za ispitanika, kako je on definiran u [6]. Stoga, ako se sigurnost i privatnost podataka naruše na bilo koji način, krajnje žrtve koje će pretrpjeti najviše štete zapravo su pojedinci čiji su podaci komprimirani. Dakle, dobro sprovedena anonimizacija biti će od velike koristi pri osiguranju dobre razine privatnosti podataka u slučajevima javnog dijeljenja podatka ili bilo kakvog neovlaštenog pristupa podacima. Još jedan faktor koji utječe na rizik od otkrivanja identiteta jest upravo broj atributa koji predstavljaju osobne podatke u skupu podataka. Intuitivno se može zaključiti da, što je taj broj veći, to je identifikacija pojedinca lakša, a pogotovo ako su ti podaci loše ili nikako pseudonimizirani ili anonimizirani. U praktičnom primjeru ovog rada biti će

demonstrirano kako anonimizacija zapravo pomaže otkloniti višak atributa koji su nepotrebni za analitičke i istraživačke svrhe obrade podataka, poput izravnih identifikatora.

U uvodnoj izjavi 26. u [6] je navedeno kako se načela zaštite podataka OUZP-a ne odnose na anonimne podatke, kojima se identitet pojedinca ne može utvrditi. Ovo je jedan od osnovnih razloga zašto se podaci annonimiziraju. Motivacija za time leži u činjenici da se, prema većini zakonskih načela zaštite podataka, podaci ne smiju obrađivati u svrhe za koje nisu prikupljeni, odnosno u svrhe za koje ispitanik nije dao svoju privolu, te kako je izrečeno stavkom 1.(b) članka 1. u [6, p. 35] osobni podaci moraju biti

"prikupljeni u posebne, izričite i zakonite svrhe te se dalje ne smiju obrađivati na način koji nije u skladu s tim svrhama; daljnja obrada u svrhe arhiviranja u javnom interesu, u svrhe znanstvenog ili povijesnog istraživanja ili u statističke svrhe, u skladu s člankom 89. stavkom 1. ne smatra se neusklađenom s prvotnim svrhama ("ograničavanje svrhe")".

Dodatno se u uvodnoj izjavi 32. [6, p. 6] navodi da bi se privola ispitanika trebala tražiti za svaku svrhu obrade podataka za slučaj da obrada ima višestruke svrhe obrade. Također, člankom 9. ograničava se obrada posebnih kategorija podataka. U stavku 1. tog članka stoji sljedeće:

"Zabranjuje se obrada osobnih podataka koji otkrivaju rasno ili etničko podrijetlo, politička mišljenja, vjerska ili filozofska uvjerenja ili članstvo u sindikatu te obrada genetskih podataka, biometrijskih podataka u svrhu jedinstvene identifikacije pojedinca, podataka koji se odnose na zdravlje ili podataka o spolnom životu ili seksualnoj orijentaciji pojedinca." [6, p. 38]

Dok se stavkom 2.(a) navodi kako se načelo iz stavka 1. ne primjenjuje u slučaju da je ispitanik dao izričitu privolu za obradu tih podataka. Dakle, postupkom anonimizacije, organizacije mogu pojavom novih svrha obrade iskoristiti anonimizirane podatke, koji izvorno nisu bili prikupljeni u te svrhe zbog toga što se više ne smatraju osobnim podacima. Također, kako će kasnije biti demonstrirano, anonimizirani podaci mogu biti i od velike koristi za potrebe testiranja i razvoja programskih rješenja, u tom je slučaju vrlo bitno koristiti podatke kakvi se mogu očekivati u stvarnom okruženju. **Dakle, anonimizacija se ne primjenjuje samo kada je osjetljive podatke potrebno javno dijeliti ili štiti od neovlaštenog pristupa, već i kada se njihova vrijednost nastoji maksimizirati primjenom u različite svrhe obrade.** Iznimna važnost potonje tvrdnje očitava se i u situacijama kada prikupljeni podaci više nisu potrebni za obradu u primarne svrhe prikupljanja te se stoga moraju uništiti. Otklanjanjem, tj. dovoljnim minimiziranjem rizika od identifikacije, ti se podaci mogu pohraniti i koristiti u budućnosti kada će biti potrebni za neke

druge svrhe. OUZP ne specificira dozvoljene periode zadržavanja podataka, ali u članku 5. stavku 1.(e) jasno je napomenuto kako osobni podaci moraju biti:

„čuvani u obliku koji omogućuje identifikaciju ispitanika samo onoliko dugo koliko je potrebno u svrhe radi kojih se osobni podaci obrađuju; osobni podaci mogu se pohraniti na dulja razdoblja ako će se osobni podaci obrađivati isključivo u svrhe arhiviranja u javnom interesu, u svrhe znanstvenog ili povijesnog istraživanja ili u statističke svrhe..."

Ranije je navedeno kako dobro sprovedena anonimizacija smanjuje rizik od povrede privatnosti podataka, no, isto tako smanjuje i kvalitetu skupa podataka zbog *prejakih* izmjena vrijednosti podataka. *Prejake* se izmjene mogu opisati kao one izmjene koje podatke mijenjaju u tolikoj mjeri da skup podataka više nije od koristi za statističku obradu i analizu. Autori u [7] napominju, u kontekstu zdravstvenih podataka, kako je uvijek bolje rizik od identifikacije smatrati većim od predviđenoga, tj. potrebno je bolje anonimizirati podatke te tako povećati njihovu privatnost. Također napominju i kako se većom kvalitetom, tj. iskoristivošću podataka, pojedinci mogu lakše identificirati. Samim time, kvalitetniji skupovi podataka, lakše su mete napada na anonimizirane skupove. Razvoj novih tehnologija uvelike pomaže pri tome, pa se OUZP-om u stavku 1. članka 35. navodi sljedeće:

„Ako je vjerojatno da će neka vrsta obrade, osobito putem novih tehnologija i uzimajući u obzir prirodu, opseg, kontekst i svrhe obrade, prouzročiti visok rizik za prava i slobode pojedinaca, voditelj obrade prije obrade provodi procjenu učinka predviđenih postupaka obrade na zaštitu osobnih podataka. Jedna procjena može se odnositi na niz sličnih postupaka obrade koji predstavljaju slične visoke rizike." [6, p. 53]

Kako autori u [8] ističu, OUZP promiče pristup zaštiti podataka temeljen na procjenama rizika. Takav pristup je holistički i stvara prilike za postavljanje brojnih bitnih pitanja koja se tiču zaštite podataka kroz cijeli njihov životni vijek. Stoga, procjenu učinka obrade treba sagledati i u kontekstu otpornosti anonimiziranog skupa podataka na pojedine vrste napada sa svrhom identificiranja pojedinca, koji predstavljaju jedan od glavnih rizika djelotvorne anonimizacije kojom se nastoji očuvati i podatkovna iskoristivost skupa podataka koji se nastoji anonimizirati, a ne samo njihova privatnost. U nastavku slijedi pregled nekih od najznačajnijih vrsta napada na anonimizirane skupove podatka.

3.3. Pregled mogućih napada na skupove podataka

U ovom će poglavlju biti razmotreni mogući napadi na skupove podataka. **Svi navedeni napadi imati će za cilj otkrivanje identiteta osobe na koju se odnosi jedan ili više zapisa tog skupa podataka, odnosno identificiranje pojedinca.** Ti će napadi na

podatke biti razmotreni s aspekta rezultata procesa anonimizacije, tj. kakvi sve faktori koji se pojavljuju u slučaju pojedinih napada utječu na rizik od identifikacije pojedinca i povrede osobnih podataka, kako je ona definirana u [6]. Za diskusiju o kontekstu ovih napada, primarno su korišteni izvori [7] i [8], s obzirom na to što u njima autori Arbuckle i El Emam na vrlo temeljit način predstavljaju dugogodišnja stručna iskustva u području anonimizacije podataka. Ovi su radovi također vrlo vrijedni i zbog toga što pružaju holistički pogled na proces anonimizacije podataka, postavljaju pitanja i donose odgovore za sve točke koje podaci prolaze u svom životnom ciklusu – od trenutka nastajanja, tj. preuzimanja podataka od ispitanika, preko obrade i anonimizacije od strane izvršitelja obrade, pa do njihovog dijeljenja primatelju ili trećoj strani.

3.3.1. Napad na javno objavljene skupove podataka

U primjeru otkrivanja identiteta guvernera Welda iz [1], prikazan je primjer napada na podatke kada su javno objavljeni, a primatelj podataka nije poznat, tj. on može biti bilo tko, stoga je i njegova motivacija za mogućim napadom nepoznata. Autori u [7], [8] takve napade nazivaju *demonstracijskim napadima*. Naime, ističu kako su to namjerni napadi na podatke s ciljem otkrivanja identiteta, obavljeni pretežito od strane akademskih zajednica, istraživača i medija općenito. Cilj im je podići svijest o procesu anonimizacije, a za metu napada i kao glavni primjeri uspješne identifikacije, obično se uzimaju zapisi koji se odnose na osobe koje je najlakše identificirati u skupu podataka, poput prisutnosti neke javne, poznate osobe u zapisima skupa podataka. Autori dalje ističu da je baš to razlog zašto je ova vrsta napada najgora s obzirom na to što predstavlja najveći rizik od identifikacije. Ovi napadi mogu, npr. iskoristiti činjenicu kako skupovi podataka sadržavaju vrijednosti nekih atributa koje u velikoj mjeri odskaču od, npr. srednje vrijednosti za taj atribut, poput rukovoditeljskih plaća koje su obično znatno veće od plaća radnika iz operativnog sloja organizacije; ili kada vrlo mali broj zapisa dijeli neku vrijednost konkretnog atributa, ili pak specifične kombinacije vrijednosti više atributa. Tako je u [1] navedeno kako je **guvernera Welda konačno identificirala činjenica da je on bio jedina osoba s peteroznamenkastim poštanskim brojem** u skupini od šestero, koja je dijelila isti datum rođenja, a u kojoj su se nalazile samo tri muške osobe. Ta je činjenica dalje pomogla Sweeney da pomoću kombinacije poštanskog broja, datuma rođenja i spola pronađe pripadne zapise u skupu zdravstvenih podataka. Stoga, kako se napadačevi tehnološki resursi i motivacija ne mogu pretpostaviti, većina rizika ovisi o odabranoj razini anonimnosti podataka, ali još više o obilježjima samog dataseta, kao što je to demonstrirano u [1]. Nadalje, svrha ovih napada jest skrenuti pozornost na moguće slabe točke procesa i rezultata anonimizacije. Dakle, demonstracijski napadi koriste se pretežito u

istraživačke svrhe, od strane napadača, tj. primatelja podataka, kojima nije cilj zlouporabiti podatke o pojedincima čiji je identitet otkriven.

3.3.2. Nenamjerna otkrivanja identiteta iz anonimiziranih skupova podataka

Identitet pojedinca u skupu podataka može biti otkriven i sasvim slučajno. Autori u [7] i [8] takav tip napada nazivaju nenamjernim jer oni ne moraju nužno predstavljati „napad" u pravom smislu te riječi. Do ovakvih otkrivanja identiteta može doći sasvim slučajno, kada izvršitelj obrade u skupu podataka prepozna podatke koji bi se mogli odnositi na neku osobu koju izvršitelj poznaje, dakle na njegovog poznanika. Rasprostranjenost neke karakteristike, poput neke bolesti u određenoj socio-ekonomsko-geografskoj skupini i prosječan broj poznanika koje osoba ima, samo su neki od faktora koji su bitni za procjenu rizika od ovog napada. Voditelj obrade, stoga bi trebao dobro poznavati skup podataka s obzirom na navedene faktore, te prema procijenjenom riziku, između ostaloga, odlučiti o prikladnom izvršitelju obrade.

3.3.3. Zlonamjerna otkrivanja identiteta iz anonimiziranih skupova podataka

U sljedeću skupinu napada ubrajaju se napadi gdje je identifikacija pojedinca namjerna s jasnom motivacijom i predvidim resursima te predvidim napadačem. Kao primjer takvih napada, može poslužiti zamišljeni slučaj u kojem se podaci prosljeđuju organizacijskom odjelu (ili nekom sveučilištu) koji će zaprimljene podatke, npr., obrađivati tehnikama strojnog učenja. Kako je ovo situacija u kojoj sam izvršitelj obrade predstavlja i potencijalnog napadača na podatke, faktori poput motivacije, intelektualnih i tehnoloških kapaciteta izvršitelja kao napadača su od značajnog utjecaja na rizik od identifikacije. U praktičnom dijelu ovog rada, predstavljen je zamišljeni scenarij u kojem se već anonimizirani podaci prosljeđuju eksternaliziranom izvršitelju obrade, tj. tvrtki koja za voditelja obrade izrađuje mobilnu aplikaciju. Pretpostavljena će tvrtka anonimizirane podatke koristiti za potrebe razvoja, testiranja i održavanja aplikacije. Kako će kasnije biti demonstrirano, ovakav slučaj odnosa između voditelja i izvršitelja obrade, zahtjeva podosta stroge ugovorne klauzule u smislu dozvoljenih aktivnosti obrade koje izvršitelj smije obavljati nad podacima, te tehnička ograničenja pristupa podacima. OUZP stavkom 1. članka 28. za izvršitelja obrade izriče sljedeće:

„Ako se obrada provodi u ime voditelja obrade, voditelj obrade koristi se jedino izvršiteljima obrade koji u dovoljnoj mjeri jamče provedbu odgovarajućih tehničkih i organizacijskih mjera na način da je obrada u skladu sa zahtjevima iz ove Uredbe i da se njome osigurava zaštita prava ispitanika”. [6, p. 49]

Dakle, rizici kod ovog napada minimiziraju se ponajprije organizacijskim i tehničkim mjerama, kojima se nastoje ograničiti dozvoljene aktivnosti izvršitelja obrade.

3.3.4. Povreda podataka

Posljednja skupina napada na podatke predstavlja sama povreda podataka, onako kako je definirana u [6]. Rizici kod ovog napada najviše se minimiziraju dobrom implementacijom tehničkih i organizacijskih mjera zaštite podataka. Načela OUZP-a primarno se fokusiraju na tehničku i integriranu zaštitu podataka (eng. privacy by design). Stoga se stavkom 1. članka 25. navodi sljedeće:

„Uzimajući u obzir najnovija dostignuća, trošak provedbe te prirodu, opseg, kontekst i svrhe obrade, kao i rizike različitih razina vjerojatnosti i ozbiljnosti za prava i slobode pojedinaca koji proizlaze iz obrade podataka, voditelj obrade, i u vrijeme određivanja sredstava obrade i u vrijeme same obrade, provodi odgovarajuće tehničke i organizacijske mjere, poput pseudonimizacije, za omogućavanje učinkovite primjene načela zaštite podataka, kao što je smanjenje količine podataka, te uključenje zaštitnih mjera u obradu kako bi se ispunili zahtjevi iz ove Uredbe i zaštitila prava ispitanika.„ [6, p. 48]

U praktičnome dijelu ovoga rada biti će predstavljene neke od mogućih tehničkih i organizacijskih mjera zaštite podataka.

3.4. Zaključna razmatranja

U ovome su poglavlju predstavljeni osnovni koncepti, definicije i pojmovi potrebni za shvaćanje procesa anonimizacije. Objašnjenje navedenoga uvelike se baziralo na OUZP-u, s obzirom na to što je ta regulativa na snazi i u Republici Hrvatskoj, te stoga predstavlja polaznu **zakonsku** referencu za potrebe diskusije o anonimizaciji podataka u ovome radu. Ovim se poglavljem nastojala stvoriti početna intuicija svrhe i cilja anonimizacije, bez pretjeranog zalaženja u tehničke detalje provedbe tog procesa. Važno je, prije svega, razumjeti kako se o anonimizaciji ne može raspravljati bez obuhvaćanja rasprave o mjerama zaštite podataka i osiguranja privatnosti istih. Ukratko je objašnjen i holistički pristup

anonimizaciji, kojim se nastoje sagledati svi bitni faktori koji utječu na rizik od identifikacije pojedinca u skupu podatka, poput motivacije, tehničkih i intelektualnih resursa napadača te same prirode podataka koji su predmet anonimizacije. Takav holistički pristup najbolje predočuje činjenicu da proces anonimizacije ima relativan rezultat, s obzirom na odnos između iskoristivosti i privatnosti skupa podataka – što je skup podataka bolje anonimiziran, to mu je iskoristivost manja. No, rizik od identifikacije, ma koliko god bio nizak, uvijek će postojati. Ukratko su predstalvjene i ideje iza nekih tehničkih i organizacijskih mjera anonimizacije i zaštite podataka, a detaljnija obrada tih ideja slijedi u praktičnom dijelu rada.

U sljedećem poglavlju, kao uvod u praktični dio rada, slijedi pregled osnovnih tehnika anonimizacije podataka i jednostavnih primjera njihove implementacije.

4. Pregled osnovnih tehnika anonimizacije

U ovome je poglavlju dan pregled osnovnih i polaznih tehnika anonimizacije. Na početku je opisan odabrani skup podataka koji će biti anonimiziran u nadolazećim poglavljima, potom izvorni slučaj korištenja tog skupa podataka, a predstavljen je i kratki opis atributa tog skupa podataka. Nakon toga slijedi opis zamišljenog scenarija i slučaja izvršenja procesa anonimizacije nad odabranim skupom podataka. Potom, prije samog početka procesa anonimizacije, sljede uvodne transformacije skupa podataka koje su nužne kako bi se isti što uspješnije anonimizirao. Tome slijedi demonstracija procesa anonimizacije za potrebe statističke i analitičke obrade ili istraživanja. Nakon interpretacije rezultata anonimizacije, sljedi demonstracija procesa anonimizacije za potrebe razvoja, testiranja i održavanja aplikacijskog rješenja. Na kraju pak slijedi implementacija potrebnih baza podataka za pohranu izvornog i anonimiziranog skupa. U tome će dijelu također biti podrobnije predstavljene tehničke i organizacijske mjere koje se tiču zaštite podataka u smislu prava pristupa i korištenja istih.

4.1. Opis odabranog skupa podataka

Za potrebe praktičnog dijela ovog rada, bilo je potrebno odabrati skup podataka koji sadrži atribute koji predstavljaju osobne podatke, poput imena, datuma rođenja, spola, etniciteta, adrese, rase, itd., koji će se biti anonimizirani. Odabrani skup podataka korišten je od strane autora u [13] gdje istražuju pristranost algoritma za procjenu rizika od recidivizma kriminalnih prijestupnika, te rezultata istih. Problematika koju autori nastoje razotkriti također je od iznimne važnosti za diskusiju o anonimizaciji, s obzirom na stigmatizirajuće modele strojnog učenja zbog čijih se pristranih rezultata mogu donositi odluke na uštrb neke društvene skupine. Kao što će biti prikazano, odabrani skup podataka obiluje osobnim podacima, a sadrži i dosta atributa koji se mogu opisati osjetljivima.

4.1.1. Opis izvornog slučaja korištenja odabranog skupa podataka

Autori u [8] dosta pozornosti posvećuju i diskusiji o anonimizaciji u kontekstu izrade modela strojnog učenja. Kao jedan od glavnih problema ističe se pristranost samog skupa podataka nad kojim se uči model, pa samim time i pristranih rezultata. Takvi pristrani rezultati

mogu imati vrlo loše posljedice na donošenje odluke, a pogotovo kada se odluka tiče čovjekove privatnosti i prava slobode. Pristranosti mogu biti razne i ticati se, npr., demografskih, socio-ekonomskih i drugih faktora. Kako ističe autorica u [14], za te je pristranosti kriv upravo dizajner algoritma, kada namjerno ili ne, kreira algoritam koji zbog tehničkih malfunkcija daje krive rezultate, ili pak kada koristi nepotpune, neuravnotežene ili jednostavno krive podatke za učenje algoritma. Kao primjer, autorica navodi algoritam kojeg je Amazon koristio za selekciju kandidata za zapošljavanje, koji je preferirao osobe muškog roda, odbacujući životopise kandidatkinja. Još jedan takav primjer, u kojem pristrani algoritam odlučuje o čovjekovoj budućnosti u smislu narušavanja prava slobode, jest korištenje algoritama za savjetovanje pri donošenju odluke o kazni za kriminalnog prijestupnika, već prema ocjeni rizika od recidivizma ili nasilja. Zaista zabrinjavajuća priča o posljedicama koje takav primjer ostavlja na ljudske živote [13] govori o tzv. COMPAS (eng. *Correctional Offender Management Profiling for Alternative Sanctions*), sustavu za procjenu rizika od recidivizma čiji su rezultati rasno pristrani na uštrb crnačke populacije, koje su uključivale nerazumno visoke kazne, **znatno više od onih predloženih od strane zakona, porote ili odvjetnika**. Naime, kako autori u [15] zaključuju iz svog istraživanja, spomenuti algoritam u velikoj mjeri precjenjuje rizik od recidivizma kada je u pitanju osoba crnačkog podrijetla, naspram osobe bjelačkog podrijetla, te precjenjuje rizik od nasilja kada je u pitanju osoba bjelačkog podrijetla, naspram osobe crnačkog podrijetla. Dobiveni rezultati bili su toliko pogrešni, da su čak i za osobe bez kriminalne prošlosti, a članovima crnačke populacije, predviđene vrijednost rizika bile nerazumno velike. Koliko je to pitanje ozbiljno, govori i sama potreba o njihovoj zakonskoj obradi, koja se tek počinje realizirati. Tako je 2018. donesena Deklaracija o etičnosti i zaštiti podataka u umjetnoj inteligenciji [16], koja opisuje temeljne etičke principe razvoja umjetne inteligencije.

U ovom je radu korišten upravo skup podataka kojeg su analizirali i autori u [15], baš iz razloga jer obiluje atributima koji predstavljaju osjetljive podatke, poput podataka o spolu, etnicitetu te o procijenjenim rizicima. Autori su navedeni skup podataka priložili putem *GitHub* repozitorija [17], odakle je on i preuzet, a znatiželjan čitatelj može posjetiti navedeni repozitorij kako bi se i sam pobliže upoznao sa zanimljivim metodama analize pristranosti algoritma COMPAS sustava. Konkretan se skup podataka iz spomenutog repozitorija može preuzeti na poveznici https://raw.githubusercontent.com/propublica/compas-analysis/master/compas-scores-raw.csv (dostupno 10.09.2020.). Navedeni skup podataka pročišćen je od strane autora istraživanja u [15], te je zato i odabran za potrebe ovog rada.

4.1.2. Kratki opis atributa odabranog skupa podataka

Obilatost ovog dataseta atributima koji predstavljaju osjetljive podatke od iznimne je koristi za kvalitetnu demonstraciju procesa anonimizacije. U ovom će pregledu atributa biti objašnjeni oni najbitniji, a opis onih manje bitnih slijedi tijekom objašnjenja implementacije praktičnog rješenja. Između ostaloga, sadrži direktne identifikatore, poput imena i prezimena, kvazi identifikatore poput datuma rođenja, te ono najbitnije - sadrži osjetljive atribute koji predstavljaju procijenjene vrijednosti rizika od budućeg počinjenja kaznenih dijela, te podatke o spolu, etnicitetu, govornom jeziku, itd. Imena svih atributa dostupna su samo na engleskom jeziku, no većina njih je razumljiva naziva. Popis i pripadno objašnjenje, te vrsta identifikatora (u zagradi) atributa dani su u nastavku:

- Person_ID - identifikacijski broj (IB) pojedinca (izravni identifikator)
- AssessmentID - IB procjene rizika (izravni identifikator)
- Case_ID - IB kriminalnog slučaja korištenog u procjeni (izravni identifikator)
- Agency_Text - naziv ustanove, pr. policijske postaje, koja je priložila kriminalni zapis/slučaj (kvazi-identifikator)
- LastName, FirstName, MiddleName - redom: prezime, ime, srednje ime pojedinca (izravni identifikator)
- Sex_Code_Text - tekstualna oznaka spola (kvazi-identifikator, osjetljivi podatak)
- Ethnic_Code_Text - tekstualna oznaka etničke, odnosno rasne pripadnosti (kvazi-identifikator, osjetljivi podatak)
- DateOfBirth - datum rođenja (kvazi-identifikator)
- AssessmentReason - razlog procjene (kvazi-identifikator, osjetljivi podatak)
- Language - govorni jezik osobe, pretpostavljeno materinji (kvazi-identifikator, osjetljivi podatak)
- LegalStatus - trenutni pravni status pojednica, ovisno o dijelu procesa suđenja u kojem se trenutno nalazi (kvazi-identifikator, osjetljivi podatak)
- CustodyStatus - opsiuje trenutnu pravnu razinu slobode pojedinca (kvazi-identifikator, osjetljivi podatak)
- MaritalStatus - bračno stanje (kvazi-identifikator, osjetljivi podatak)
- Screening_Date - datum izvršene procjene (kvazi-identifikator)

- RecSupervisionLevel - predložena razina nadgledanja aktivnosti pojedinca (kvazi-identifikator, osjetljivi podatak)
- RecSupervisionLevelText - tekstualna oznaka predložene razine nadgledanja aktivnosti pojedinca (kvazi-identifikator, osjetljivi podatak)
- DisplayText - tekstualna oznaka vrste rizika koji se procjenjuje
- RawScore - numerička vrijednosti procijenjenog rizika (kvazi-identifikator, osjetljivi podatak)
- DecileScore - decilna oznaka procijenjene vrijednosti rizika (kvazi-identifikator, osjetljivi podatak)
- ScoreText - opisna oznaka procijenjene vrijednosti rizika (kvazi-identifikator, osjetljivi podatak)

Nadalje, zbog povećanja praktičnih primjera u ovome radu, izvornom će datasetu biti pridodani atributi (kvazi-identifikatori) poput adrese i e-pošte, koji će poslužiti u demonstrativne svrhe postupka anonimizacije.

4.2. Opis zamišljenog slučaja korištenja u implementaciji

Iako su ovi podaci javno dostupni, možemo zamisliti scenarij u kojem, osim tijela vlasti, te podatke mogu na transparentan način vidjeti jedino pojedinci za koje je vršena procjena rizika. Dodatno, ti podaci, baš kao i u realnom primjeru ovog članka, mogu biti od neizmjerne koristi za istraživače diljem svijeta kojima je namjera istraživati korelacije između socio-ekonomskih faktora i kriminalnih nagona pojedinaca. Stoga, motivacija za anonimizacijom ovih podataka u ovom zamišljenom scenariju proizlazi iz nastojanja zaštite identiteta pojedinaca prilikom javnog objavljivanja ovih podataka. Također, ovom zamišljenom scenariju može biti pridodana i situacija u kojoj se za dostavu tih podataka procjene pojedincima koji su predmet istih koristi mobilna aplikacija, koju pak razvija eksternalizirani ugovoreni izvršitelj obrade. Za testiranje aplikacija potrebni su stvarni podaci, no u ovom će slučaju biti potrebno zaštiti i sve izravne te kvazi identifikatore, kako ih izvršitelj ne bi mogao zlouporabiti. U tom će segmentu anonimizacije biti predstavljene i različite metode ograničavanja pristupa podacima kroz ugovorene politike i ograničena prava pristupa stvarnim podacima. Kratko će biti predstavljena i sama zaštita tih podataka metodama kriptiranja, dok će testni podaci biti rezultat skupova nasumično generiranih podataka koji će

samo "izgledati" kao stvarni podaci, tj. slagati će se s njima po tipu podataka i vrstama varijabli koje oni predstavljaju. Za potrebe anonoimizacije ovog dataseta u svrhe javnog objavljivanja i daljnje analize podataka u kojoj izravni identifikatori nisu bitni, odabran je pristup anonimizacije generiranjem sintetičkih podataka na način da konačni anonimizirani dataset zadržava statistička svojstva izvornog dataseta. Taj pristup predstavlja odmak od tradicionalnih metoda poput "k-anonymity", "l-diversity" ili pak "t-closeness". Također, biti će predstavljen i koncept diferencijalne privatnosti, što je pristup dodatnog osiguranja podataka od razlučivanja pojedinaca ili skupina pojedinaca zbog naknadnih promjena datasetova, čije se razlike mogu iskoristiti za otkrivanje pojedinaca u datasetu.

Slijedi demonstracija svih tehničkih procesa uključenih u anonimizaciju odabranog dataseta. Potrebno je prije napomenuti da je sav programski kod dostupan u prilozima ovog rada, a za samo objašnjenje praktičnih primjera koristit će se manji isječci kodova radi očuvanja sažetosti rada. Sav kod u prilozima dobro je komentiran, kako bi se čitatelj mogao što bolje snaći u njegovu razumijevanju.

4.3. Inicijalna priprema odabranog skupa podataka za anonimizaciju

Na slici 1 prikazan je ispis prvih 6 zapisa dataseta.

	0	1	2	3	4	5
Person_ID	50844	50844	50844	50848	50848	50848
AssessmentID	57167	57167	57167	57174	57174	57174
Case_ID	51950	51950	51950	51956	51956	51956
Agency_Text	PRETRIAL	PRETRIAL	PRETRIAL	PRETRIAL	PRETRIAL	PRETRIAL
LastName	Fisher	Fisher	Fisher	KENDALL	KENDALL	KENDALL
FirstName	Kevin	Kevin	Kevin	KEVIN	KEVIN	KEVIN
MiddleName	NaN	NaN	NaN	NaN	NaN	NaN
Sex_Code_Text	Male	Male	Male	Male	Male	Male
Ethnic_Code_Text	Caucasian	Caucasian	Caucasian	Caucasian	Caucasian	Caucasian
DateOfBirth	12/05/92	12/05/92	12/05/92	09/16/84	09/16/84	09/16/84
ScaleSet_ID	22	22	22	22	22	22
ScaleSet	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen
AssessmentReason	Intake	Intake	Intake	Intake	Intake	Intake
Language	English	English	English	English	English	English
LegalStatus	Pretrial	Pretrial	Pretrial	Pretrial	Pretrial	Pretrial
CustodyStatus	Jail Inmate	Jail Inmate	Jail Inmate	Jail Inmate	Jail Inmate	Jail Inmate
MaritalStatus	Single	Single	Single	Married	Married	Married
Screening_Date	1/1/13 0:00	1/1/13 0:00	1/1/13 0:00	1/1/13 0:00	1/1/13 0:00	1/1/13 0:00
RecSupervisionLevel	1	1	1	1	1	1
RecSupervisionLevelText	Low	Low	Low	Low	Low	Low
Scale_ID	7	8	18	7	8	18
DisplayText	Risk of Violence	Risk of Recidivism	Risk of Failure to Appear	Risk of Violence	Risk of Recidivism	Risk of Failure to Appear
RawScore	-2.08	-1.06	15	-2.84	-1.5	19
DecileScore	4	2	1	2	1	3
ScoreText	Low	Low	Low	Low	Low	Low
AssessmentType	New	New	New	New	New	New
IsCompleted	1	1	1	1	1	1
IsDeleted	0	0	0	0	0	0

Slika 1: Izgled izvornog dataseta

Prikazano je prvih 6 zapisa izvornog dataseta. Potrebno je obratiti pozornost kako za svaku procjenu postoje tri zapisa. Na gornjem je ispisu to očito jer se sa svaka tri zapisa ponavlja ista vrijednost atributa “AssessmentID". Detaljnijom analizom, može se ustanoviti da je taj atribut svojevrstan primarni ključ ovog dataseta, s obzirom na to što za jednu osobu ili slučaj, tj. jednu vrijednosti atributa "Person_ID" ili "Case_ID", respektivno, može postojati više procjena. U svakom je zapisu jedne procjene zapravo vršena procjena različite vrste rizika. Ti se zapisi mogu spojiti u jedan, uz dodavanje dodatnih atributa, po jedan za svaku vrstu rizika. Tako je u prvom koraku pripreme dataseta za anonimizaciju potrebno spojiti te zapise svake procjene. Nadalje, u cijelom se datasetu vrše samo tri različite procjene rizika, a to su: rizik od recidivizma - "Risk of Recidivism", rizik od nasilja - "Risk of Violence", te rizik od nepojavljivanja, pretpostavljeno na suđenju - Risk of Failure to Appear. Za pohranu podataka o riziku svakog zapisa pojedine procjene, korištena je Pythonova struktura podataka rječnik, koja predstavlja niz parova ključ-vrijednost. Tako svaki IB procjene predstavlja ključ, a skup vrijednosti rizika procjene predstavlja pripadne vrijednosti za taj ključ. Dakle, ovim se korakom povećava preglednost dataseta, te se smanjuje nepotrebna redundantnost redaka. Potom je potrebno iterirati kroz preostali dataset i uzimati samo jedan redak za svaku procjenu, pretvarati taj redak u rječnik i spajati ga s pripadajućim zapisom u 'assessments_risks' rječniku, prema ključu 'Assessment_ID'. Taj dodani zapis predstavljati će nove atribute rizika. Opisani postupak predstavlja svojevrsnu normalizaciju dataseta u smislu uklanjanja redundantnih zapisa. Primjer izvlačenja podataka svake procjene dan je u sljedećem isječku koda, s početkom u liniji 15, dok je dio s normalizacijom prikazan s početkom u liniji 43.

```
# učitavanje dataseta ----
df_new_compas = pd.read_csv("./data/my_datasets/new_compas.csv")

# izvlačenje IB-eva procjena ----
# lista sadrži 20281 elemenata
list_of_assessmentid = df_compass.AssessmentID.unique().tolist()

# izvlačenje podataka pojedine vrste rizika za svaku procjenu ----
'''
Lista rizika
['Risk of Violence', 'Risk of Recidivism', 'Risk of Failure to Appear']
'''
assessments_risks = {} # prazan rječnik
# izdvajanje zapisa procjena
for assess_id in list_of_assessmentid:
    assess_records = df_compass[df_compass["AssessmentID"] == assess_id].to_dict() #svi zapisi procjene

    raw_risks_scores = list(assess_records["RawScore"].values())
    decile_risks_scores = list(assess_records["DecileScore"].values())
    score_text = list(assess_records["ScoreText"].values())

    #risk of violence
    rov_raw_score, rov_decile_score, rov_score_txt = raw_risks_scores[0], decile_risks_scores[0], score_text[0]

    #risk of recidivism
    ror_raw_score, ror_decile_score, ror_score_txt = raw_risks_scores[1], decile_risks_scores[1], score_text[1]

    #risk of failure to appear
    rofa_raw_sore, rofa_decile_score, rofa_score_txt = raw_risks_scores[2], decile_risks_scores[2], score_text[2]

    assessments_risks[assess_id] = {
        "rov_raw" : rov_raw_score,
        "rov_decile" : rov_decile_score,
        "rov_score_txt" : rov_score_txt,
        "ror_raw" : ror_raw_score,
        "ror_decile" : ror_decile_score,
        "ror_score_txt" : ror_score_txt,
        "rofa_raw" : rofa_raw_sore,
        "rofa_decile" : rofa_decile_score,
        "rofa_score_txt" : rofa_score_txt
    }

# normalizacija
for assess_id in assessments_risks:
    reduced_vals = df_reduced_compas[df_reduced_compas["AssessmentID"] == assess_id].iloc[0].to_dict()
    assessments_risks[assess_id].update(reduced_vals)
# lista normaliziranih zapisa
new_dataset_records = list(assessments_risks.values())

# stvaranje novog dataseta iz liste zapisa
df_new_compas = pd.DataFrame(new_dataset_records)
```

Kao što je ranije navedeno, izvornom će datasetu biti dodani novi atributi:

- address - nasumično generirana adresa (kvazi identifikator)
- postcode - nasumično generirani poštanski broj (kvazi identifikator)
- email - adresa e-pošte koja se sastoji od imena pojedinca i nasumično generirane domene

U ovom je koraku za nasumično generiranje podataka korišten programski modul Faker [18]. Taj programski modul svojim bogatim rječnicima omogućuje generiranje vrlo velikog skupa različitih vrsta podataka, poput imena, prezimena, imena gradova, adresa ulica, poštanskih brojeva, boja, IBAN-ova, geo-podataka, IP adresa, itd. S obzirom da omogućuje generiranje podataka koji s aspekta anonimizacije predstavljaju osobne podatke, vrlo je koristan izvor za generiranje skupova testnih podataka. U ovom je koraku dodavanja novih atributa najprije potrebno izdvojiti osobne podatke svake osobe, generirati vrijednosti novih atributa, te ih spojiti s postojećima. Potom je prilikom dodavanja email adresa, potrebno voditi računa o

tome da se ne dogodi da osobe s istim inicijalima imaju istu email adresu. Provjera je vršena pomoću rječnika, čiji ključevi predstavljaju slova engleske abecede, a njihove vrijednosti predstavljaju liste zauzetih email adresa koje počinju sa pripadnim slovom ključa. Vrijednosti adresa i poštanskih brojeva u potpunosti se nasumično generiraju modulom Faker. Na kraju je potrebno još odbaciti nepotrebne atribute "AssessmentType", "IsCompleted" i "IsDeleted", koji i nemaju pretjeranu svrhu za ovaj rad, obzirom na to što su vrijednosti ovih atributa svih zapisa jednake.

```
    # idvajanje izravnih identifikatora ----
    df_new_personal_info = df_new_compas[["AssessmentID", "Person_ID", "FirstName", "MiddleName", "LastName"]].copy()
    df_new_personal_info = df_new_personal_info.drop_duplicates(subset=["Person_ID"])
    # ---- dodavanje adresa e-pošte ----
    # definiranje rječnika za provjeru zauzetih adresa e-pošte
    alphabet_list = list(string.ascii_lowercase)
    email_dict = { letter : set() for letter in alphabet_list }
    # definiranje skupa mogućih domena e-pošte
    email_domains = ["gmail.com", "yahoo.com", "hotmail.com", "outlook.com", "mail.com", "webmail.com"]

    # definiranje pseudoslučajnih vjerojatnosti odabira pojedine domene
    domains_number = len(email_domains)
    domains_weights = random.choices(range(15, 60) , k = domains_number)

    # funkcija za odabir domene prema nasumičnoj vjerojatnosti
    def get_edomains_weights(number_of_domains):
        domains_weights = random.choices(range(15, 60) , k = number_of_domains)
        return domains_weights

    # funkcija za generiranje email adrese prema inicijalima imena i srednjeg imena, te cijelog prezimena
    def generate_email(email_prefix, email_domains):
        domains_weights = get_edomains_weights(len(email_domains))
        random_domain = random.choice(range(0, len(email_domains)))
        email_domain = random.choices(email_domains, weights = domains_weights, k = 6)[0]
        return email_prefix + "@" + email_domain

    # lista jedinstvenih IB-eva osoba
    list_of_personid = df_new_personal_info.Person_ID.unique().tolist()

    # generiranje adresa e-pošte za svaku osobu u datasetu ----
    for person_id in list_of_personid:
             f_name,  m_name,  l_name  =  tuple(df_new_personal_info[df_new_personal_info["Person_ID"]  ==  person_id]
    [["FirstName", "MiddleName", "LastName"]].iloc[0].to_dict().values())
        #print(f_name, " ", m_name, " ", l_name)

        if pd.isnull(m_name):
            email_prefix = f_name[0].lower() + "." + l_name.lower()
        else:
            email_prefix = f_name[0].lower() + m_name[0].lower() + "." + l_name.lower()

        email_address = generate_email(email_prefix, email_domains)

        while email_address in email_dict[email_address[0]]:
            random_num = str(random.choice(range(100, 999)))
            email_prefix = email_prefix + random_num
            email_address = generate_email(email_prefix, email_domains)

        email_dict[email_address[0]].add(email_address)

        df_new_personal_info.loc[df_new_personal_info.Person_ID == person_id, "Email"] = email_address
    # ---- dodavanje adresa i poštanskih brojeva (ZIP) pomoću knjižnice Faker ----
    faker_generator = faker.Faker(["en-US"])



    # generiranje adresa i ZIP-ova za svaku osobu u datasetu
    for person_id in list_of_personid:
        fake_address = faker_generator.street_address()
        fake_postcode = faker_generator.postcode()
        df_new_personal_info.loc[df_new_personal_info.Person_ID == person_id, "Address"] = fake_address
        df_new_personal_info.loc[df_new_personal_info.Person_ID == person_id, "Postcode"] = fake_postcode

    # spajanje i pohrana novog dataseta ----
    df_new_compas_pi = pd.merge(df_new_compas,

```

```
66
67                                  df_new_personal_info[["Person_ID", "Email", "Address", "Postcode"]],
68                                  how="inner",
69                                  on="Person_ID")
70      columns_to_drop = ["AssessmentType", "IsCompleted", "IsDeleted"]
71      df_new_compas_pi.drop(columns = columns_to_drop, inplace = True)
72
73      df_new_compas_pi.to_csv("./data/my_datasets/new_compas_pi.csv", index = False)
74
```

Slika 2 prikazuje izgled dataseta nakon početnih transformacija.

	0	1	2
AssessmentID	649	1310	1547
Person_ID	866	1502	1733
Case_ID	559	1198	1430
Agency_Text	PRETRIAL	PRETRIAL	PRETRIAL
LastName	Boles	DIXON	Orie
FirstName	Lashawn	DUVAL	Horace
MiddleName	Nicole	J	NaN
Email	ln.boles@hotmail.com	dj.dixon@outlook.com	h.orie@outlook.com
Address	8308 Christine Court Suite 642	63592 Rodgers Street Apt. 002	1017 James Prairie
Postcode	20129	37594	96288
Sex_Code_Text	Female	Male	Male
Ethnic_Code_Text	African-American	African-American	African-American
DateOfBirth	01/05/80	01/07/79	04/17/86
ScaleSet_ID	22	22	22
ScaleSet	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen
AssessmentReason	Intake	Intake	Intake
Language	English	English	English
LegalStatus	Pretrial	Pretrial	Pretrial
CustodyStatus	Jail Inmate	Jail Inmate	Jail Inmate
MaritalStatus	Single	Single	Single
Screening_Date	11/24/13 0:00	3/24/13 0:00	10/2/14 0:00
RecSupervisionLevel	1	1	2
RecSupervisionLevelText	Low	Low	Medium
rov_raw	-2.86	-1.95	-1.57
rov_decile	2	5	6
rov_score_txt	Low	Medium	Medium
ror_raw	-1.46	-0.8	-0.1
ror_decile	2	3	7
ror_score_txt	Low	Low	Medium
rofa_raw	17	24	18
rofa_decile	2	6	3
rofa_score_txt	Low	Medium	Low

Slika 2: Izgled izvornog dataseta nakon početnih transformacija (autorski rad, 2020)

Ovime su završene početne transformacije dataseta. U sljedećem poglavlju slijedi demonstracija procesa anonimizacije za potrebe statističkog i analitičkog istraživanja.

5. Provedba anonimizacije za potrebe analitičkog i statističkog istraživanja

U ovome će dijelu rada biti izvršen postupak anonimizacije izvornog dataseta za daljnje potrebe analize i obrade podataka, kao što je npr. strojno učenje. Razlog zašto je ovaj dio anonimizacije odvojen jest činjenica kako za analitičke potrebe nisu potrebni svi atributi koji predstavljaju nekakve osobne podatke pojedinca, poput adresa, poštanskih brojeva, email adresa, itd. Također nisu potrebni ni izravni identifikatori, poput punog imena i prezimena, dok npr., podatak o datumu rođenja, odnosno godini starosti može biti od značajne koristi. Stoga, je najprije potrebno odstraniti nepotrebne atribute koji predstavljaju izravne identifikatore. Potom slijedi vrlo bitan dio anonimizacije svakog dataseta, a to je istraživanje prirode njegovih atributa, što obuhvaća analizu tipa podataka i domenu atributa, koju vrstu statističke varijable pojedini atribut predstavlja, te dozvoljene intervale vrijednosti ili pak konačne skupove vrijednosti koje neki atribut može poprimiti. Nakon tog koraka, slijedi obrada temporalnih atributa dataseta.

U koraku nakon, predstavljena je sama srž anonimizacije generiranjem umjetnih, tzv. sintetičkih uz očuvanje statističkih svojstava dataseta. U tu je svrhu korišten alat ***DataSyntheziser***, verzije 0.1.2, dostupan putem *Github* repositorija [19] Ovaj alat prati i popratni znanstveni rad, autora Ping, Stoyanovich i Have, u kojem ga detaljnije opisuju. Rad je dostupan na poveznici [20]. Alat DataSyntheziser je zapravo prgogramski modul programskog jezika Python. Razlog zašto je odabran za ovaj rad jest jednostavnost njegove primjene i obilatost funkcionalnostima. Sam se alat sastoji od tri dijela, odnosno modula:

- ***DataDescriber*** - programska klasa koja implementira metode za stvaranje detaljnog opisnika atributa i statističkih svojstava dataseta. Primjenom Bayesovih mreža i koncepta uzajamne obavijenosti parova atributa (eng. pairwise mutual information), ova klasa omogućuje opisivanje korelacija atributa. Preciznije, ovim se postupkom utvrđuju vjerojatnosti da će promjena vrijednosti nekog atributa utjecati na promjenu vrijednosti nekog drugog atributa (Baysova mreža) i u kolikoj mjeri (uzajamna obavijenost). Kolokvijalno rečeno, zanima nas na koji način vrijednosti atributa uvjetuju jedni druge. Dodatno, ovim se postupkom izrađuje i opisnik vjerojatnosnih distribucija vrijednosti pojedinih atributa, što se pak kasnije koristi za njihov vizualni prikaz pomoću histograma. Dodatno se u ovom procesu opisuju i vrste statističkih

varijabli koje atributi predstavljaju, poput neprekidnih numeričkih ili pak kvalitativnih, tzv. kategoričkih, varijabli.

- ***DataGenerator*** - Koristi statistički opisnik izrađen pomoću metoda DataDescriber modula za generiranje sintetičkih podataka uz zadržavanje opisanih statističkih svojstava.
- ***ModelInspector*** - Služi za vizualizaciju histograma vrijednosti atributa i prikaz toplinske mape uzajamne obavijenosti parova atributa. Spomenuta toplinska mapa vrlo je važna za vizualni prikaz korelacija atributa, gdje se na ljestvici od 0 do 1 opisuje vjerojatnost da poznavajući vrijednost jednog atributa možemo znati vrijednost drugog atributa. Primjer za to dan je u jednom slučaju korištenja (eng. use - case) u kojem je korišten ovaj alat za anonimizaciju zdravstvenih podataka, dostupan na [21]. Između ostalog, u tom se primjeru istražuje korelacija atributa vremena čekanja na red za prijem u ambulanti hitne pomoći i godina startosti pacijenta. Utvrđeno je kako godine pacijenta u dobroj mjeri predviđaju koliko će pacijent čekati na red, tj. što je pacijent stariji - to će kraće čekati. Stoga je ovaj modul od iznimne važnosti za razumijevanje odrađenog procesa anonimizacije, a dodatno omogućuje i usporedbu navedenih metrika između izvornog i anonimiziranog dataseta.

Potrebno je još napomenuti kako ovaj programski modul podržava tri različita načina rada:

- **generiranje nasumičnih podataka**: u potpunosti se zanemaruju korelacije atributa dataseta i vjrojatnosne distribucije njihovih vrijednosti. Cilj ovog načina rada jest generirati dataset s podacima koji samo sliče izvornim podacima, tj. oponašati domene i tipove podataka atributa, što je prikladno za, npr. generiranje datasetova u svrhe testiranja rada aplikacija.
- **generiranje podataka uz očuvanje vjerojatnosnih distribucija vrijednosti atributa**: ponovno, u obzir se ne uzimaju korelacije atributa, već samo vjerojatnosne distribucije njihovih vrijednosti. Ovaj modul, kako je opisano u [21] također rezultira poprilično "uvjerljivim" statističkim svojstvima anonimiziranog dataseta u usporedbi s izvornim.
- **generiranje podataka uz očuvanje korelacija atributa**: generiraju se vrijednosti prema vjrojatnosnim distribucijama i uzajamnim korelacijama atributa. Ovo je najkompleksniji način rada od sva tri, ali, uz pravilnu upotrebu, donosi i najpreciznije podatke. Dodatno, ovaj modul omogućuje i primjenu diferencijalne privatnosti u

datasetu, generiranjem "smetnji" u vrijednostima dataseta prema Laplaceovoj distribuciji. Razina tzv. smetnji određuje se tzv. ϵ (epsilon) parametrom.

5.1. Početni koraci u anonimizaciji

Već je ranije navedeno da za potrebe obrade za koje se vrši anonimizacija u ovom dijelu rada nisu potebni direktni identifikatori, stoga će njihovo uklanjanje biti korak u procesu anonimizacije. Naime identifikacijski broj, kraće IB, procjene, osobe te kriminalnog slučaja nemaju pretjeranu važnost za sam postupak strojnog učenja, no oni ipak zadržavaju informaciju o tome postoji li više različitih procjena za istu osobu. Taj problem ćemo rješiti jednostavnim transformiranjem dataseta, tako da mu dodamo atribut "FirstAssessment", koji će predstavljati informaciju o tome predstavlja li konkretni zapis prvu procjenu za neku osobu ili ne. Taj će atribut moći poprimiti vrijednosti iz skupa {0, 1}, pri čemu nula (0) predstavlja da se zapis odnosi na „ponovljenu" (ne prvu) procjenu, dok jedinica (1) predstavlja da se zapis odnosi na prvu procjenu. Na taj način ipak možemo očuvati informaciju o ponavljanju procjena, a da ipak smanjimo rizik od reidentifikacije. Iako, postoji mogućnost da se razazna grupa osoba s ponovljenim procjenama, ako napadač ima uvid u izvorni dataset ili ako ima specifičnu informaciju o tome ponavljaju li se procjene za neku osobu. Proces provjere vršiti će se na način da se inicijalizira prazan skup, inače Pythonova podatkovna struktura koja ne dozvoljava pohranu duplikata, u koju će se iterativno spremati IB-evi osoba iz posjećenih zapisa. Ako se IB osobe već nalazi u skupu, vrijednost novododanog atributa „FirstAssessment" posjećenog zapisa postavit će se na jedan (1), a u suprotnom na nulu (0). Nakon što se zabilježe podaci o ponavljanjima procjena, potrebno je ukloniti nepotrebne izravne i kvazi-identifikatore. Sljedeći isječak koda prikazuje opisani postupak.

```
# učitavanje dataseta
df_new_compas_ml = pd.read_csv("./data/my_datasets/new_compas_pi.csv")

# 1) Označavanje procjena koje se odnose na istu osobu
#-----------------------------------------------------
# dohvaćanje liste jedinstvenih IB-eva procjena
assessment_id_list = list(df_new_compas_ml.AssessmentID.unique())

# inicijaliziranje praznog skupa za provjeru članstva IB-a osobe
# ako IB postoji u skupu, posjećena procjena predstavlja "ponovljenu" procjenu za tu osobu
person_id_visited_set = set()

# iteriranje kroz zapise i zapisivanje podatka o "ponovljenoj" procjeni
for assess_id in assessment_id_list:
    visited_pid = df_new_compas_ml.loc[df_new_compas_ml["AssessmentID"] == assess_id]["Person_ID"].iloc[0]

    if visited_pid in person_id_visited_set: #provjera članstva u skupu person_id_visited_set
        df_new_compas_ml.loc[df_new_compas_ml["AssessmentID"] == assess_id, ["FirstAssessment"]] = 0

    else:
        df_new_compas_ml.loc[df_new_compas_ml["AssessmentID"] == assess_id, ["FirstAssessment"]] = 1
        person_id_visited_set.add(visited_pid)

# promjena tipa podatka atributa "FirstAssessment" sa znakovnog na cjelobrojni
df_new_compas_ml[["FirstAssessment"]] = df_new_compas_ml[["FirstAssessment"]].astype(int).copy()

# 2) Izbacivanje svih nepotrebnih izravnih identifikatora i kvazi-identifikatora
```

```
#-------------------------------------------------------------------------
# specificiranje liste nepotrebnih atributa
unneeded_attributes = [
    "AssessmentID", "Person_ID", "Case_ID", "FirstName",
    "MiddleName", "LastName", "Email", "Address"
]

# izbacivanje atributa
df_new_compas_ml = df_new_compas_ml.drop(columns = unneeded_attributes).copy()

# spremanje nove verzije dataseta
df_new_compas_ml.to_csv("./data/anon_datasets/ANON_new_compas_ml.csv", index = False)
```

Za izradu kvalitetnog dataseta koji je anonimiziran, a koji opet zadržava statistička svojstva izvornog dataseta, potrebno se najprije upoznati s atributima. Cilj ovog dijela rada jest istražiti preostale vrste atributa, kakvim vrstama statističkih varijabli oni pripadaju, koje su njihove domene i razumijeti njihovo značenje. U prošlom su dijelu uklonjeni svi direktni identifikatori i većina kvazi-identifikatora, poput imena, prezimena, email adrese te adrese stanovanja, te identifikacijski brojevi generirani iz baze podataka koji bi, uvidom u izvorni dataset, mogli neposredno pomoći pri prepoznavanju pojedinaca. Svi ti atributi nisu potrebni za bilo kakvo pronalaženje znanja u ovom datasetu, a kojem je cilj iskoristiti informacije o procijenjenim vrijednostima različitih vrsta rizika i profilima pripadnih osoba. Stoga, kako bi proces anonimizacije bio kvalitetno izveden, potrebno je razumijeti važnost svakoga od ovih atributa, a posebice u ovakvom slučaju kada bi "curenje" podataka moglo rezultirati javnom sramotom za nekog pojedinca. Osnovni korak u anonimizaciji već je obavljen izbacivanjem nepotrebnih atributa, a sljedeći korak zahtjeva upoznavanje s domenama, tj, tipovima podataka pojedinih atributa. U ovome postupku pomaže atribut Pandasovih DataFrame objekata ***dtypes*** kojim se dohvaća specifikacija tipova podataka atributa dataseta, kako ih Pandas prepoznaje. Također, od koristi će biti i DataFrame metoda ***nuinque()*** koja se koristi za dohvaćanje broja jedinstvenih vrijednosti pojedinog atributa dataseta. Ta je metoda od ključne važnosti za raspoznavanje vrsta statističkih varijabli koje pojedini atributi predstavljaju, te iz kojih točno skupova ili intervala pojedini atributi poprimjau vrijednosti. Slika 4 prikazuje brojeve njihovih jedinstvenih vrijednosti, dok slika 3 prikazuje specifikaciju tipova podataka pojedinih atributa, dok sljedeći isječak koda prikazuje uporabu navedenih metoda.

```
# 1) --- OBRADA ATRIBUTA DATASETA ---
# učitavanje dataseta
df_anon_ml = pd.read_csv("./data/anon_datasets/ANON_new_compas_ml.csv")


# 1.1) Istraživanje tipova podataka i domena atributa dataseta
#-----------------------------------------------------------
# pregled tipova podataka atributa dataseta prema Pandas-u
df_anon_ml.dtypes

# pregled broja jedinstvenih vrijednosti svakog atributa, bez nedostajećih, tzv. "Nan" vrijednosti
df_anon_ml.nunique(dropna = True)
```

```
Agency_Text                object
Postcode                    int64
Sex_Code_Text              object
Ethnic_Code_Text           object
DateOfBirth                object
ScaleSet_ID                 int64
ScaleSet                   object
AssessmentReason           object
Language                   object
LegalStatus                object
CustodyStatus              object
MaritalStatus              object
Screening_Date             object
RecSupervisionLevel         int64
RecSupervisionLevelText    object
rov_raw                   float64
rov_decile                  int64
rov_score_txt              object
ror_raw                   float64
ror_decile                  int64
ror_score_txt              object
rofa_raw                  float64
rofa_decile                 int64
rofa_score_txt             object
FirstAssessment             int64
dtype: object
```

Slika 4: Specifikacija tipova podataka dataseta (autorski rad, 2020)

```
Agency_Text                    4
Postcode                   17004
Sex_Code_Text                  2
Ethnic_Code_Text               9
DateOfBirth                10382
ScaleSet_ID                    2
ScaleSet                       2
AssessmentReason               1
Language                       2
LegalStatus                    7
CustodyStatus                  6
MaritalStatus                  7
Screening_Date               705
RecSupervisionLevel            4
RecSupervisionLevelText        4
rov_raw                      485
rov_decile                    11
rov_score_txt                  3
ror_raw                      457
ror_decile                    11
ror_score_txt                  3
rofa_raw                      41
rofa_decile                   10
rofa_score_txt                 3
FirstAssessment                2
dtype: int64
```

Slika 3: Broj jedinstvenih vrijednosti pojedinog atributa dataseta (autorski rad, 2020)

Vidimo da je Pandas ispravno prepoznao sve numeričke atribute, bilo u obliku tipa podataka "int64" ili "float64", no nije prepoznao atribute znakovne domene i atribute s vremenskim, odnosno datumskim podacima. No, to je od manje važnosti, s obzirom na to što se pregledom zapisa tipovi podataka tih atributa mogu vrlo lako raspoznati. Ono što je bitno su ispravno detektirani tipovi numeričkih atributa jer nam ukazuju na vrste statističkih varijabli kojima vrijednosti tih atributa pripadaju. Konkretno, zanima nas koji numerički atributi imaju vrijednosti koje pripadaju kvalitativnim varijablama, a koje pak pripadaju numeričkim varijablama. Glavno obilježje numeričke varijable jest da poprima vrijednosti iz skupa realnih brojeva, pa takvi atributi mogu poprimiti beskonačno mnogo različitih vrijednosti (kontinuirane varijable), iako je ponekad u pitanju nekakav zatvoreni interval. Dok pak kvalitativne varijable vrijednosti mogu obično poprimiti iz konačnog i prebrojivog skupa, što, naravno, može biti i skup prirodnih ili pak cijelih brojeva, no to ih ne čini kontinuiranim varijablama [22]. Primjer kvalitativne varijable bio bi skup dana u tjednu, dok bi primjer numeričke varijable bio iznos (atmosferskog) tlaka zraka. Uvidom u zapise dataseta, vidimo kako većina atributa s detektiranim cjelobrojnim tipom podataka "int64" zapravo opisuju, odnosno kategoriziraju neki od tekstualnih atributa ili atributa s detektiranim tipom podataka "float64", respektivno. Odmah je očito kako atributi "Postcode", "DateOfBirth", "Screening_Date", te atributi s procjenjenim vrijednostima rizika "rov_raw", "ror_raw", te "rofa_raw", velikim brojem

jedinstvenih vrijednosti odskaču od ostalih atributa, a pogotovo od onih tekstualnog tipa podataka. Zbog toga možemo s velikom vjerojatnošću pretpostaviti da se radi o atributima čije vrijednosti možemo smatrati numeričkim kontinuiranim varijablama. Iznimke su, naravno, datumski/vremenski atributi, jer oni, iako ne sadrže vrijednosti iz skupa realnih brojeva, predstavljaju kontinuirane numeričke varijable. Najzanimljiviji su nam atributi s procjenjenim vrijednostima rizika. Svaki od rizika: rizik od nasilja, recidivizma i nepojavljivanja na suđenju - osim procjenjene numeričke vrijednosti, ima i pripadnu vrijednost s obzirom na decil kojem procjenjena vrijednost pripada, te tekstualnu vrijednost koja na dodatan kvalitativni i ljudima razumljiviji način kategorizacije, opisuje tu procjenjenu vrijednost. Kažemo da atributi koji sadrže pripadnu informaciju o decilu kojemu procjenjna vrijednost rizika pripada, kategoriziraju procjenjene vrijednosti rizika. Kako se radi o informaciji o decilima kojima procjenjene vrijednosti rizika pripadaju, s velikom vjerojatnošću možemo pretpostaviti kako za svaki decil postoji određeni interval, tj. raspon kojemu te vrijednosti pripadaju. Te decilne raspone dodatno opisuju i tekstualne vrijednosti koje predstavljaju ordinalne varijable, za koje je karakteristična hijerarhijska podjela kategorija. Kako bismo utvrdili točnu prirodu atributa koji opisuju procjenje vrijednosti rizika, potrebno je dataljnije istražiti ih. Kao prvo, potrebno je odrediti minimalne i maksimalne raspone za procjenjene vrijednosti rizika, potom raspone za pojedince decile, te na kraju, koje kvalitativne oznake opisuju pojedine decile. Za moguće decile i njihove opsine, tekstualne vrijednosti, moramo znati i skup vrijednosti koje oni mogu poprimiti, tj. jedinstvene vrijednosti tih atributa.

5.1.1. Istraživanje atributa odabranog skupa podataka

Najprije će biti obrađeni atributi rizika: zanimaju nas preslikavanja između raspona pojedinih decila i pripadnih tekstualnih oznaka. Kako je proces utvrđivanja ovih informacija podosta dugačak, programsko rješenje ovdje neće biti navedeno, već samo rezultati. Čitatelja se, stoga upuće da pogleda priloge radu 1, 2 i 3.

Slike 7, 6 i 5 prikazuju utvrđena preslikavanja za rizike od recidivizma, nepojavljivanja na sudu i nasilja, respektivno.

	decile	raw_min	raw_max	txt_label
0	-1	-1.00	-1.00	NaN
1	1	-3.21	-1.25	Low
2	2	-1.65	-0.86	Low
3	3	-1.30	-0.60	Low
4	4	-1.02	-0.38	Low
5	5	-0.81	-0.18	Medium
6	6	-0.58	0.01	Medium
7	7	-0.36	0.19	Medium
8	8	-0.13	0.40	High
9	9	0.13	0.67	High
10	10	0.44	2.36	High

Slika 5: Preslikavanja numeričkih raspona i tekstualnih oznaka za rizik od recidivizma (autorski rad, 2020)

	decile	raw_min	raw_max	txt_label
0	1	11.0	16.0	Low
1	2	16.0	19.0	Low
2	3	18.0	21.0	Low
3	4	20.0	23.0	Low
4	5	22.0	25.0	Medium
5	6	23.0	27.0	Medium
6	7	25.0	29.0	Medium
7	8	27.0	31.0	High
8	9	29.0	35.0	High
9	10	34.0	51.0	High

Slika 6: Preslikavanja numeričkih raspona i tekstualnih oznaka za rizik od pojavljivanja na sudu (autorski rad, 2020)

	decile	raw_min	raw_max	txt_label
0	-1	-1.00	-1.00	NaN
1	1	-4.79	-2.95	Low
2	2	-2.94	-2.56	Low
3	3	-2.55	-2.24	Low
4	4	-2.23	-1.98	Low
5	5	-1.97	-1.74	Medium
6	6	-1.73	-1.50	Medium
7	7	-1.49	-1.26	Medium
8	8	-1.25	-1.00	High
9	9	-0.99	-0.63	High
10	10	-0.62	1.52	High

Slika 7: Preslikavanja numeričkih raspona i tekstualnih oznaka za rizik od nasilja (autorski rad, 2020)

Očito je kako se svaka od 3 vrste rizika opisuje kao visokom (eng. High), srednjom (eng. Medium), niskom (eng. Low), ili vrijednošću "nan", koja zapravo označuje kako osoba nema pridjeljen pripadni rizik. Utvrđena će mapiranje biti potrebna na kraju kada će anonimizirani dataset ponovno trebati opisati tim vrijednostima. Stoga, atributi koji opisuju decil u koji spada pojedina procjenjna vrijednost te atributi koji riječju opisuju razinu rizika, nisu potrebni u procesu anonimizacije te mogu biti odstranjeni iz dataseta. Ti atrubuti izravno ovise o procjenjenoj vrijednosti rizika te služe samo za njezino dodatno opisivanje.

Ovo je također i jedan bitan korak u procesu *deanonimizacije* dataseta, a obavlja ga napadač kada za pojedini atribut želi saznati raspone vrijednosti koje on može poprimiti. Tako bi, npr., napdač mogao detektirati neke rubne vrijednosti za koje može pretpostaviti da ih sadrži mali broj zapisa, što mu može pomoći kod identificiranja pojedinca ili pronalaženja zapisa koji s aspekta anonimizacije predstavlja najslabiju kariku. Potrebno je istražiti i ostale atribute.

Pregledom skupa jedinstvenih vrijednosti atributa "Agency_Text", možemo biti sigurni kako njegove vrijednosti zapravo predstavljaju kvalitativne varijable. Domenu tog atributa predstavlja sljedeći konačan skup tekstualnih vrijednosti: *{'Broward County', 'DRRD', 'PRETRIAL', 'Probation'}*.

Za atribut „Postcode”, ranije je utvrđeno kako za njega ukupno postoji 17004 različite vrijednosti, te da Pandas njegove vrijednosti prepoznaje kao tip podataka "int64", tj.

cjelobrojni tip (eng. integer). No, iako vrijednosti koje ovaj atribut može poprimiti pripadaju cjelobrojnom tipu, to ne znači da one predstavljaju kontinuiranu varijablu. Naprotiv, vrijednosti ovog atributa čine ga kvalitativnom varijablom, s obzirom na to što je skup vrijednosti koje može poprimiti ograničen. To intuitivno ima smisla jer raspon poštanskih brojeva zasigurno nije neograničen, s obzirom na to kako svaka vrijednost poštanskog broja obilježava neko naselje, grad ili pak regiju, ovisno o državi. Neke od vrijednosti atributa "Postcode":

Atribut „Sex_Code_Text" obilježje spola, možemo zaključiti kako njegove vrijednosti predstavljaju kvalitativnu varijablu, s obzirom na to što opisuju različite spolove, a sam atribut vrijednosti poprima iz konačnog skupa tekstualnih oznaka *{'Male', 'Female'}*.

Karakteristike atributa „Ethnic_Code_Text" jednake su atributu "Sex_Code_Text", osim što on predstavlja etničko obilježje osobe, te mu je skup mogućih vrijednosti drugačiji: 'Caucasian', 'Other', 'Oriental', 'Native American', 'African-American', 'African-Am', 'Arabic', 'Asian', 'Hispanic'}. Očito je kako se u skupu jedinstvenih vrijednosti atributa "Ethnic_Code_Text" pojavljuju dvije vrlo slične vrijednosti: "African-American" i "African-Am". S obzirom da bi mogle upućivati na istu oznaku, potrebno je zamjeniti sve vrijednosti "African-Am" s "African-American". Moguće je da je ovo greška kod unosa podataka u bazu podataka, stoga će biti korisno sanirati ovu pojavu, zamjenom svih vrijednosti "African-Am" s "African-American". Tome koristi sljedeći isječak koda:

```
# učitavanje dataseta
df_anon_ml = pd.read_csv("./data/anon_datasets/ANON_new_compas_ml.csv")

# zamjena vrijednosti
df_anon_ml.loc[df_anon_ml.Ethnic_Code_Text == "African-Am", "Ethnic_Code_Text"] = "African-American"
```

Atributi *„ScaleSet_ID", „ScaleSet"* predstavljaju prvi par atributa kod kojeg je tekstualna vrijednost također opisana i numeričkom, tj. tekstualne vrijednosti atributa "ScaleSet" imaju pripadajuće numeričke parnjake u atributu "ScaleSet_ID". Možemo pretpostaviti kako je "ScaleSet_ID" zapravo primarni ključ relacije baze podataka u kojoj su pohranjeni podaci atributa "ScaleSet". Pandas nam može pomoći pri provjeri ovog preslikavanja tekstualnih vrijednosti atributa "ScaleSet" na numeričke vrijednosti atributa "ScaleSet_ID". Izvršenjem sljedećeg isječka koda, dobiva se preslikavanje kao na slici 8, koje je također potrebno pohraniti, kao i preslikavanja kod atributa rizika:

```
# izdvajanje atributa "ScaleSet_ID" i "ScaleSet", te odbacivanje duplikata
df_anon_ml[["ScaleSet_ID", "ScaleSet"]].drop_duplicates().reset_index(drop = True)
```

	ScaleSet_ID	ScaleSet
0	22	Risk and Prescreen
1	17	All Scales

Slika 8: Spajanje identifikatora i tekstualnih naziva atributa ScaleSet (autorski rad, 2020)

Vidimo da je pretpostavka o preslikavanju točna, s obzirom na to da su iz dataseta izdvojena samo ova dva atributa te izbačeni svi duplikatni redovi. Prema tome, jedan od ova dva atributa nije potreban, a kako je tekstualni atribut prirodno razumljiviji, moguće je izbaciti atribut "ScaleSet_ID".

Sljedeći par atributa koji bi također mogli sadržavati tablicu preslikavanja su atributi „RecSupervisionLevel" i „RecSupervisionLevelText". Ovaj je primjer sličan preslikavanjima prisutnima kod atributa rizika, tj. ovo je još jedan par atributa u datasetu u kojem opisne, tekstualne vrijednosti imaju i svoje numeričke parnjake. Očito ovaj atribut predstavlja neku možebitnu razinu nadgledanja pojedinca, te je ona opisana riječima i brojkom. Kako god da bilo, radi se o kvalitativnoj varijabli.

Potom atribut „AssessmentReason" svojim vrijednostima opisuje razlog procjene. Priroda ovog dataseta jest takva što svi zapisi sadrže istu vrijednost ovog atributa, te on kao takav nema nikakvog, osim semantičkog, utjecaja na svojstva dataseta. No, ipak će biti zadržan u procesu anonimizacije zbog toga što bi se u budućnosti mogli pojaviti zapisi s različitim vrijednostima tog atributa. Skup jedinstvenih vrijednosti ovog atributa, barem za sad, je: {'Intake'}.

Jezik je jedan od ključnih atributa u procesu anonimizacije, s obzirom na to što predstavlja kvazi-identifikator. Vrijednosti koje taj atribut može poprimiti prema ovom datasetu su sljedeće: {'English', 'Spanish'}.

„LegalStatus" je još jedan važan atribut u procesu anonimizacije, s obzirom na to što odaje podosta bitnu informaciju o pojedinčevom pravnom statusu, odnosno položaju. Ovo je još jedan od atributa čije vrijednosti predstavljaju kvalitativnu varijablu, a sam atribut može poprimiti vrijednosti iz sljedećeg skupa: {'Conditional Release', 'Deferred Sentencing', 'Other', 'Parole Violator', 'Post Sentence', 'Pretrial', 'Probation Violator'}.

Kod atributa „CustodyStatus”, baš kao i atribut "LegalStatus", i ovdje nailazimo na primjer atributa čije vrijednosti predstavljaju kvalitativnu varijablu, a sam atribut može poprimiti vrijednosti iz sljedećg skupa: {'Jail Inmate', 'Parole', 'Pretrial Defendant', 'Prison Inmate', 'Probation', 'Residential Program'}.

Atribut „MaritalStatus” je primjer još jednog bitnog kvazi-identifikatora, obzirom na to što sadržava podatak o bračnom statusu pojedinca na kojeg se pripadni zapis odnosi, a mogući skup vrijednosti ovog atributa u ovom datasetu je sljedeći: {'Divorced', 'Married', 'Separated', 'Significant Other', 'Single', 'Unknown', 'Widowed'}.

Za kraj su još ostali temporalni atributi "DateOfBirth" i "Screening_Date", koji sadrže informaciju o datumu rođenja te datumu i vremenu procjene rizika, respektivno. Po prirodi svojih vrijednosti, ovi atributi predstavljaju numeričke varijable, stoga za njih samo postoji određeni interval vrijednosti koje mogu poprimiti. Vidjet ćemo kako je taj interval podosta kratak za atribut "Screening_Date" - iznosi tek 20 mjeseci. Kada metoda anonimizacije generiranjem sintetičkih podataka ne bi bila korištena, tj. kada bi se koristila, npr., metoda k-anonimnosti za ovaj atribut, napadču bi ovakav vrlo kratak vremenski period mogao biti od pomoći da razazna zapise nekog pojedinca, kada bi napdač znao u kojem je periodu unutar tih 20 mjeseci procjena tog pojedinca bila obavljena. Razlog tome je što bi korištenjem metode k-anonimnosti broj različitih grupa, tj. vremenskih intervala, bio vrlo ograničen, može se pretpostaviti i nedovoljan za kvalitetnu anonimizaciju i zaštitu privatnosti. Korištenjem metode generiranja sintetičkih podataka smanjuje se rizik od prepoznavanja zapisa pojedinca, s obzirom na to što napadač ne može nikako biti siguran da se neke od, npr., procjenjenih vrijednosti rizika odnose na ciljanog pojedinca. Dodatno osiguranje tome je i to što je vjerojatnost da ovaj atribut korelira s nekim od atributa rizika zaista vrlo mala, drugim riječima, možemo pretpostaviti kako oni gotovo nikako ne zavise jedni o drugima. Raspon datuma za atribut "DateOfBirth" je {01.01.1947, 31.12.1995}. Atribut "Screening_Date" uz informaciju o datumu, sadrži i podatak o satu kada je procjena obavljena. Zbog očuvanja što veće privatnosti, taj će podatak o satu biti uklonjen prije samog procesa anonimizacije, obzirom na to što nije pretjerano potreban. Raspon datuma i vremena ovog atributa je {01.01.2013. 0:00, 09.09.2014. 0:00}.

Kako su adrese uklonjene iz dataseta, korisno je zadržati neku informaciju o tome odakle dolazi osoba na koju se odnosi zapis. Tako će u sljedećem koraku u anonimizaciji poštanski brojevi biti podijeljeni u decile, te će se umjesto njih, koristiti oznaka decila kojem poštanski broj pojedinog zapisa pripada. Sam atribut „Postcode” biti će uklonjen. Bitno je napomenuti kako ovaj korak dosta poopćuje podatak o lokaciji, no barem se ostvaruje bolja

anonimnost podataka. Sljedeći isječak koda prikazuje izradu decilnih raspona poštanskih brojeva, njihovo umetanje u dataset i uklanjanje atributa „Postcode”, a slika 9 prikazuje tablicu s popisom decila i pripadnih početaka raspona.

```
# 1.) izrada decila
_, bins = pd.qcut(
    df_anon_ml["Postcode"],
    10,
    retbins = True,
    labels = False
)

# pretvorba tipa podataka oznaka decila u cjelobrojni tip
bins  = [int(bin) for bin in bins]

# mapiranje rednih brojeva decila i početaka raspona
bins_list = list(zip(range(0, 11), bins))

# 2.) Zamjena poštanskih brojeva njihovim decilnim skupinama

df_anon_ml["PostcodeDecile"] = pd.cut(
    df_anon_ml["Postcode"],
    bins = bins,
    labels = False,
    include_lowest = True
) + 1

# 3.) Odbacivanje atributa "Postcode" (poštanski broj)
df_anon_ml.drop(columns = ["Postcode"], inplace = True)
```

	Decile	LowestValue
0	0	506
1	1	10330
2	2	19842
3	3	29407
4	4	39462
5	5	49311
6	6	59366
7	7	69914
8	8	79957
9	9	90028
10	10	99942

Slika 9: Prikaz decilnih skupina poštanskih brojeva (autorski rad, 2020)

Zakonske regulative, poput američke HIPAA-e, ne dozvoljavaju obradu cijelih datuma, tj. dopuštaju obradu samo komponente godine. To uvelike povećava anonimnost podataka, smanjujući mogućnost da napadač identificira pojedinca ako zna datum događaja, ili mogući datumski/vremenski raspon, u kojem je pojedinac sudjelovao, kao što je npr. rođenje osobe. Ovo je prilika demonstrirati obradu takvih temporalnih podataka. Konkretno, u kontekstu ovog dataseta, pruženi su egzaktni datumi rođenja svake osobe. Cilj je te datume rođenja preračunati u godine starosti, koje će se kasnije poopćiti, tj. podijeliti u intervale od po, recimo, 5 ili 10 godina. Dodatan problem kojeg je potrebno rješiti jest ispravno pretvoriti skraćeni format godine u kojem su navedene samo posljednje dvije znamenke u puni format sa sve 4 znamenke, kako bi se lakše izračunala starosna dob osobe. Kad se već govori o starosnoj dobi osobe, dodatan način kojim bi se postigla još veća anonimnost jest maknuti godine starosti iz dataseta, a ostaviti tekstualne oznake koje bi opisivale životnu dob osobe, npr.: dijete, adolescent, odrasla osoba, osoba treće dobi. Anonimnost, a samim time i privatnost, uvelike su se povećali, no podaci su sada dosta manje iskoristivi, ako je predmet istraživanja koristiti što točnije podatke o starosti osobe. Sljedeći isječak koda opisuje navedeni postupak izračunavanja godina starosti pojedinca, njihovo umetnaje u dataset te odbacivanje atributa „Date_Of_Birth”.

```
# Izračunava točnu godinu iz temporalnog zapisa
def correct_dob_year(date_of_birth):
    """
        Format of date_of_birth: "%m/%d/%y"
        !!Short year format!!
    """
    day, month, year = get_date_components(date_of_birth, "/")
    year = "".join(["19", year])
    corrected_dob = "/".join([day, month, year])
    return corrected_dob

# razlama temporalni zapis o datumu na njegove komponente: mjesec, dan, godinu
def get_date_components(date_string, date_components_delimiter):
    return tuple(date_string.split(date_components_delimiter))

# funkcija za pretvaranje datuma rođenja u godine starosti
def get_age_in_years(date_of_birth):
    """
        Format of date_of_birth: "mm/dd/yy"

        When 2-digit years are parsed, they
        are converted according to the POSIX
        and ISO C standards: values 69–99 are
        mapped to 1969–1999, and values 0–68
        are mapped to 2000–2068.
        [https://docs.python.org/3.8/library/time.html]

        Dataset doesn't contain any person born after 1995.,
        so all the year values lesser than 69 must be corrected.
    """
    dob_corrected = correct_dob_year(date_of_birth)
    # correcting the year value changes the date's format
    corrected_dob_format = "%m/%d/%Y"
    dob_dt = datetime.datetime.strptime(dob_corrected, corrected_dob_format)
    today_dt = datetime.datetime.today()
    age_in_years = today_dt.year - dob_dt.year

    if (today_dt.month, today_dt.day) < (dob_dt.month, dob_dt.day):
```

```
        age_in_years = age_in_years - 1

    return age_in_years

# umetanje stupca s iznosom godina starosti
df_anon_ml["Age"] = df_anon_ml.apply(lambda x: get_age_in_years(x["DateOfBirth"]), axis=1)

# odbacivanje atributa s punim datumom rođenja
df_anon_ml.drop(columns = ["DateOfBirth"], inplace = True)
```

Slika 10, na sljedećoj stranici, prikazuje konačni izgled dataseta nakon ovih koraka anonimizacije.

	0	1	2
Agency_Text	PRETRIAL	PRETRIAL	PRETRIAL
Sex_Code_Text	Female	Male	Male
Ethnic_Code_Text	African-American	African-American	African-American
PostcodeDecile	3	4	10
ScaleSet	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen
AssessmentReason	Intake	Intake	Intake
Language	English	English	English
LegalStatus	Pretrial	Pretrial	Pretrial
CustodyStatus	Jail Inmate	Jail Inmate	Jail Inmate
MaritalStatus	Single	Single	Single
Screening_Date	11/24/2013	03/24/2013	10/02/2014
RecSupervisionLevelText	Low	Low	Medium
rov_raw	-2.86	-1.95	-1.57
ror_raw	-1.46	-0.8	-0.1
rofa_raw	17	24	18
FirstAssessment	1	1	1
Age	40	41	34

Slika 10: Konačan izgled dataseta nakon početnih koraka anonimizacije (autorski rad, 2020)

5.2. Generiranje sintetičkih podataka uz očuvanje statističkih svojstava skupa podataka

U ovom će poglavlju biti predstalvjeno korištenje Pythonove biblioteke, odnosno programskog modula, **DataSynthesizer**. Iako je već izvršen dio anonimiziacije nad datasetom za strojno učenje, ovim će alatom biti odrađen ostatak postupka anonimizacije, tj. generiranje sintetičkih podataka.

Na početku je potrebno stvoriti opisnik svih atributa dataseta, kojeg će DataSynthesizer koristiti za ispravnu statističku analizu dataseta, čiji će rezultati biti od važnosti za generiranje sintetičkih podataka. Konkretno, bitno je specificirati koji atributi sadrže tekstualne, numeričke ili pak temporalne podatke. Potom, koji atributi predstavljaju kvalitativne, tj., kolokvijalno rečeno, kategoričke varijable, te koji atributi predstavljaju neprekidne (numeričke/temporalne) varijable. Trenutno, prema verziji dataseta „ANON_new_compas_ml.csv", te s obzirom na vrstu varijable koju atributi dataseta predstavljaju, vrijede sljedeće karakteristike atributa:

- **Kategorički/kvalitativni atributi:** "Agency_Text", "Sex_Code", "Ethnic_Code_Text", "ScaleSet", "AssessmentReason", "Language", "LegalStatus", "CustodyStatus", "MaritalStatus", "RecSupervisionLevelText", "PostcodeDecile", "FirstAssessment"

- **Neprekidni (numerički/temporalni) atributi:** "Age", "Screening_Date", "rov_raw", "ror_raw", "rofa_raw"

Dok s obzirom na tipove podataka vrijednosti atributa dataseta, vrijede ovi opisi**:**

- Tekstualni atributi: "Agency_Text", "Sex_Code", "Ethnic_Code_Text", "ScaleSet", "AssessmentReason", "Language", "LegalStatus", "CustodyStatus", "MaritalStatus", "RecSupervisionLevelText"

- Temporalni atributi: "Screening_Date"

- Cjelobrojni atributi: "Age", "PostcodeDecile", "FirstAssessment"

- Decimalni/realni atributi: "rov_raw", "ror_raw", "rofa_raw"

DataSynthesizer-ova klasa ***DataDescriber*** pruža metodu *describe_dataset_in_correlated_attribute_mode()* za brzo i lako učitavanje podataka o datasetu. Kao njezini argumenti, također se prosljeđuju vrijednosti parametara *dataset_file*, *epsilon*, *k*, *attribute_to_datatype* i *attribute_to_is_categorical*, koji predstavljaju apsolutnu putanju do CSV datoteke s podacima, vrijednosti razine primjene diferencijalne privatnosti, najveći broj čvorova roditelja nekog čvora Bayesove mreže, rječnik s popisom tipova podataka pojedinog atributa, te rječnik s popisom kategoričkih atributa, respektivno. Najprije je potrebno definirati navedene rječnike, a potom i ostale parametre. Kao rezultat te metode, dobiva se opis statističkih svojstava dataseta. Najbitnije svojstvo koje se nastoji očuvati u ovom procesu anonimizacije su korelacije atributa i distribucije vjerojatnosti njihovih vrijednosti (*opisane funkcijom gustoće vjerojatnosti za kontinuirane varijable i funkcijom mase vjerojatnosti za kvalitativne varijable*). Rezultat navedene metode je izrađena Bayesova mreža (aciklički graf), čiji se čvorovi sastoje od atributa dataseta, koja predstavlja korelacijski model atributa. [20] Bridovi mreže opisuju distribucije uvjetnih vjerojatnosti, tj. uvjetne distribucije, oblika *P(dijete | roditelj)*, koje kasnije omogućuju generiranje vrijednosti za atribut dijete (zavisna varijabla). Pri opisivanju tih uvjetnih distribucija, algoritam generiranja Bayesove mreže uključuje „šum”, tj. u zadanoj mjeri mijenja izračunate distribucije kako bi se postigla tražena razina diferencijalne privatnosti. Kako autori navode, što je parametar epsilon veći, to je manja razina privatnosti, tj. manji je šum. Sljedeći isječak koda prikazuje izradu opisnika atributa dataseta, korištenjem metoda klase DataDescriber.

```
# 1) Specificiranje tipova podataka atributa
#-------------------------------------------
attributes_datatypes = {
    "Agency_Text" : "String",
    "Sex_Code" : "String",
    "Ethnic_Code_Text" : "String",
    "ScaleSet" : "String",
    "AssessmentReason" : "String",
    "Language" : "String",
    "LegalStatus" : "String",
    "CustodyStatus" : "String",
    "MaritalStatus" : "String",
    "RecSupervisionLevelText" : "String",
    "Age" : "Integer",
    "PostcodeDecile" : "Integer",
    "FirstAssessment" : "Integer",
    "rov_raw" : "Float",
    "ror_raw" : "Float",
    "rofa_raw" : "Float",
    "Screening_Date" : "DateTime"
}

# 2) Specificiranje kategoričkih/kvalitavinih varijabli koje pojedini atributi predstavljaju
#-------------------------------------------------------------------------------------------
categorical_attributes = {
    "Agency_Text" :  True,
    "Sex_Code" : True,
    "Ethnic_Code_Text" : True,
    "ScaleSet" : True,
    "AssessmentReason" : True,
    "Language" : True,
    "LegalStatus" : True,
```

```
    "CustodyStatus" : True,
    "MaritalStatus" : True,
    "RecSupervisionLevelText" : True,
    "Age" : False,
    "PostcodeDecile" : False,
    "FirstAssessment" : False,
    "rov_raw" : False,
    "ror_raw" : False,
    "rofa_raw" : False,
    "Screening_Date" : False
}

# 3) Specificiranje putanja datoteka
#-----------------------------------
# putanja izvornog dataseta
dataset_anon_ml_path = "./data/anon_datasets/ANON_new_compas_ml.csv"
# putanja gdje će se spremiti JSON datoteka koja predstavlja opisnik atributa
dataset_data_description = "./data/anon_datasets/dsynth_correlated/correlated_description.json"
# putanja gdje će se spremiti CSV datoteka s generiranim sintetičkim podacima - anonimizirani dataset
synthetic_dataset_correlated_path = "./data/anon_datasets/dsynth_correlated/synthetic_compas_dataset_correlated.csv"


# 4) Specificiranje dodatnih parametara
#---------------------------------------
# epsilon parametar razine diferencijalne privatnosti
dif_epsilon = 0.3
# broj dopuštenih čvorova roditelja čvorova Bayesove mreže
max_bayes_parrents = 3
# potrebno je generirati 20281 zapisa
number_of_records = 20281

# 5) Izrada opisnika atributa modulom DataDescriber
#--------------------------------------------------

# 5.1) Izrada Bayesove mreže
#---------------------------
data_describer = DataDescriber()
data_describer.describe_dataset_in_correlated_attribute_mode(
    dataset_file = dataset_anon_ml_path,
    epsilon = dif_epsilon,
    k = max_bayes_parrents,
    attribute_to_datatype = attributes_datatypes,
    attribute_to_is_categorical = categorical_attributes
)

# prikaz dobivene mreže
display_bayesian_network(data_describer.bayesian_network)

# pohrana opisnika atributa
data_describer.save_dataset_description_to_file(dataset_data_description)
```

DataSynthesizer pruža klasu ***DataGenerator*** koja pak implementira metodu *generate_dataset_in_correlated_attribute_mode()*, kojoj je svrha generirati sintetičke podatke, prema ranije izrađenom opisniku pomoću metode *describe_dataset_in_correlated_attribute_mode()* klase DataDescriber. Obje metode služe svrsi opisivanja dataseta i generiranja sintetičkih podataka s očuvanjem korelacija između vrijednosti atributa. Kao argumente, metoda *generate_dataset_in_correlated_attribute_mode()* prima broj zapisa koji će se generirati te putanju do ranije izrađenog opisnika. Sljedeći siječak koda prikazuje postupak generiranja sintetičkih podataka s očuvanjem korelacija između atributa dataseta, pomoću DataGenerator klase.

```
# 1) Instanciranje objekta DataGenerator
#--------------------------------------
data_generator = DataGenerator()

# 2) Generiranje podataka
#------------------------
data_generator.generate_dataset_in_correlated_attribute_mode(number_of_records,
                                                             dataset_data_description)

# 3) Spremanje podataka
#----------------------
data_generator.save_synthetic_data(synthetic_dataset_correlated_path)
```

Prilikom anonimiziranja vrijendosti atributa "Screening_Date", datumske su vrijednosti morale biti pretvorene u broj sekundi koji je protekao od tzv. "*Unix Epoch*" trenutka [23], tj. datuma 01. siječnja 1970. godine u 00:00h. Te je vrijednosti potrebno ponovno pretvoriti u čiteljive datume. Također, potrebno je obraditi i numeričke atribute rizika, tj. zaokružiti njihove vrijednosti na dvije decimale. Tijekom generiranja podataka, Pandas je cjelobrojne kvalitativne atribute pretvorio u realne vrijednosti, što se također treba sanirati. Sljedeći isječak koda donosi sprovedene obje transformacije.

```
    # 1) Učitavanje anonimiziranog dataseta
    #--------------------------------------
    df_synth_anon_correlated = pd.read_csv(synthetic_dataset_correlated_path)

    # 2) Promjena vrijednosti atributa "Screening_Date" iz formata 'since epoch' u "%Y-%m-%d"
    #-----------------------------------------------------------------------------------------
    screening_datetime_from_epoch = pd.to_datetime(df_synth_anon_correlated["Screening_Date"])

    #screening_datetime_from_epoch.head() #primjer vrijednosti u "since epoch" formatu

    # pretvorba u razumljivi datumski oblik
    df_synth_anon_correlated["Screening_Date"] =
    df_synth_anon_correlated.apply(lambda x: time.strftime('%m/%d/%Y', time.localtime(x["Screening_Date"])), axis=1)

    # 3) Promjena tipa podataka za cjelobrojne kvalitativne atribute
    #---------------------------------------------------------------
    # specificiranje atributa koji trebaju biti cjelobrojnog tipa
    integer_cols = {
        "Age": "int32",
        "PostcodeDecile": "int32",
        "FirstAssessment": "int32"
    }

    # pretvorba tipova podataka pomoću Pandas-a
    df_synth_anon_correlated = df_synth_anon_correlated.astype(integer_cols).copy()

    # 4) Izgled dataseta
    #-------------------
    df_synth_anon_correlated.head(3).transpose()

    # 5) POHRANA ANONIMIZIRANOG DATASETA
    #-----------------------------------
    df_synth_anon_correlated.to_csv(synthetic_dataset_correlated_path, index = False)
```

Slika 11 predstavlja izgled anonimiziranog dataseta.

	0	1	2
Agency_Text	Probation	PRETRIAL	PRETRIAL
Sex_Code_Text	Female	Male	Male
Ethnic_Code_Text	Caucasian	African-American	African-American
Age	29	29	32
PostcodeDecile	9	10	9
ScaleSet	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen
AssessmentReason	Intake	Intake	Intake
Language	English	English	English
LegalStatus	Pretrial	Pretrial	Post Sentence
CustodyStatus	Pretrial Defendant	Pretrial Defendant	Probation
MaritalStatus	Single	Married	Married
Screening_Date	06/25/2013	07/06/2014	04/30/2014
RecSupervisionLevelText	Medium	Medium	Low
rov_raw	-1.45	-1.87	-2.6
ror_raw	-0.45	-0.28	-0.63
rofa_raw	20.81	17.03	13.73
FirstAssessment	1	1	1

Slika 11: Izgled anonimiziranog dataseta, dobivenog generiranjem sintetičkih podataka (autorski rad, 2020)

5.2.1. Diskusija o rezultatima

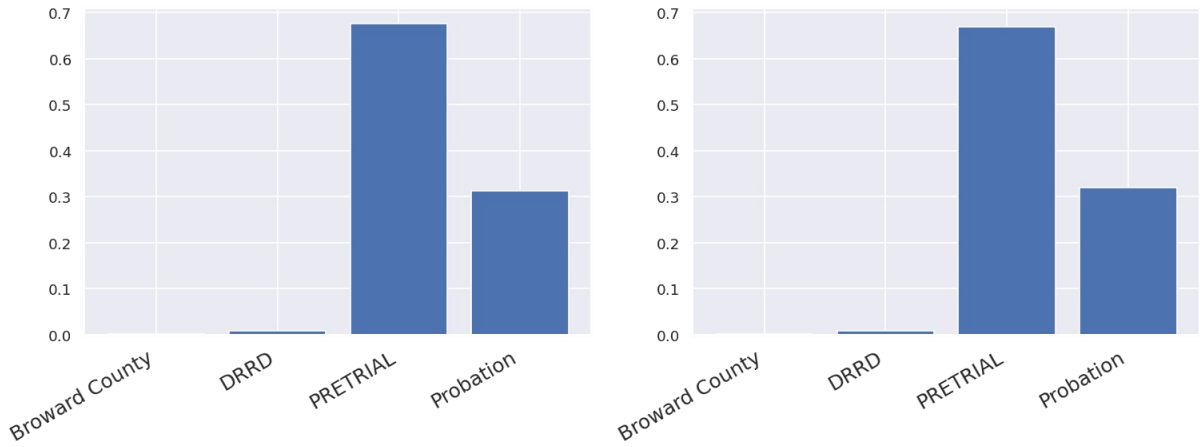
DataSynthesizer pruža klasu ***ModelInspector*** koja omogućuje metode za prikaz histograma pojedinih atributa i toplinskih mapa njihovih korelacija. Stoga je cilj u ovome dijelu istražiti distribucije vjerojatnosti vrijednosti pojedinih atributa izvornog i anonimiziranog dataseta, toplinske mape njihovih korelacija, te ih usporediti.

Najprije je potrebno učitati zapise izvornog i anonimiziranog dataseta u Pandasove DataFrame objekte, potom je potrebno učitati opisnik atributa izvornog dataseta stvorenog pomoću DataDescriber klase. Nakon što su datasetovi i opisnik učitani, potrebno ih je proslijediti kao parametre konstruktoru ModelInspector klase. Metodom *compare_histograms()* klase ModelInspector mogu se usporediti histogrami pojedinačnih atributa, dok se metodom *mutual_information_heatmap()* prikazuju toplinske mape uzajamne obavijenosti parova atributa izvornog i anonimiziranog dataseta. Sljedeći isječak koda prikazuje pozive tih metoda i instanciranje ModelInspector objekta.

```
# 1) Učitavanje izvornog i anonimiziranog dataseta te opisnika atributa
#-----------------------------------------------------------------------
# izvorni dataset
df_anon_ml = pd.read_csv(dataset_anon_ml_path)

# anonimizirani dataset
df_synth_anon_correlated = pd.read_csv(synthetic_dataset_correlated_path)

# opisnik dataseta atributa
data_description = read_json_file(dataset_data_description)['attribute_description']

# 2) Instanciranje objekta ModelInspector
#----------------------------------------
data_model_inspector = ModelInspector(df_anon_ml, df_synth_anon_correlated, data_description)

# 3) Prikaz histograma atributa izvornog dataseta (lijevo) i anonimiziranog dataseta (desno)
#-----------------------------------------------------------------------------------------------
print("--Agency_Text --")
data_model_inspector.compare_histograms("Agency_Text")

print("-- Sex_Code_Text --")
data_model_inspector.compare_histograms("Sex_Code_Text")

print("-- Ethnic_Code_Text --")
data_model_inspector.compare_histograms("Ethnic_Code_Text")

print("-- Age --")
data_model_inspector.compare_histograms("Age")

print("-- PostcodeDecile --")
data_model_inspector.compare_histograms("PostcodeDecile")

print("-- ScaleSet --")
data_model_inspector.compare_histograms("ScaleSet")

print("-- AssessmentReason --")
data_model_inspector.compare_histograms("AssessmentReason")

print("-- Language --")
data_model_inspector.compare_histograms("Language")

print("-- LegalStatus --")
data_model_inspector.compare_histograms("LegalStatus")

print("-- CustodyStatus --")
data_model_inspector.compare_histograms("CustodyStatus")

print("-- MaritalStatus --")
data_model_inspector.compare_histograms("MaritalStatus")

print("-- RecSupervisionLevelText --")
data_model_inspector.compare_histograms("RecSupervisionLevelText")

print("-- rov_raw --")
data_model_inspector.compare_histograms("rov_raw")

print("-- ror_raw --")
data_model_inspector.compare_histograms("ror_raw")

print("-- rofa_raw --")
data_model_inspector.compare_histograms("rofa_raw")

print("-- FirstAssessment --")
data_model_inspector.compare_histograms("FirstAssessment")


# 4) Prikaz toplinskih mapa korelacija atributa izvornog i anonimiziranog dataseta
#---------------------------------------------------------------------------------
data_model_inspector.mutual_information_heatmap()
```

Slijede objašnjenja dobivenih rezultata ispisima histograma i toplinskih mapa iz prethodnog isječka koda.

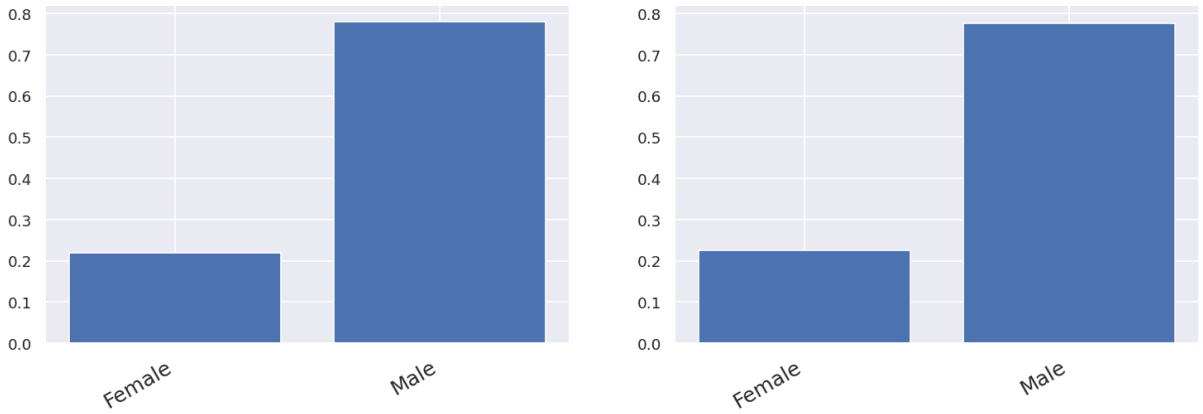
Agency_Text



Slika 12: Usporedba histograma atributa "Agency_Text" (autorski rad, 2020)

Očito je da su vrijednosti izvornog i anonimiziranog skupa vrlo sličnih distribucija, no, ipak za vrijednosti "PRETRIAL" i "Probation" u anonimiziranom skupu, kada se pobliže pogleda, postoje vrlo male razlike u očitanju s histograma.

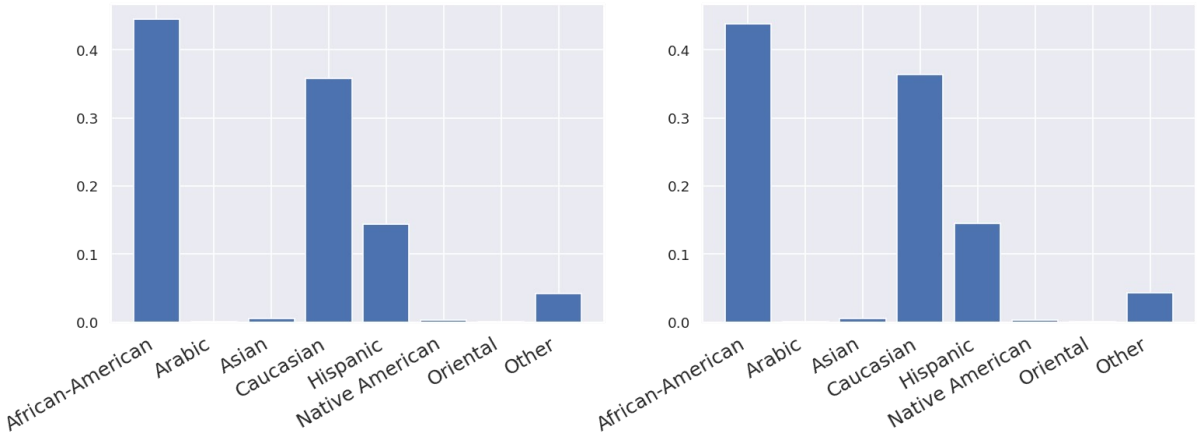
Sex_Code_Text



Slika 13: Usporedba histograma atributa "Sex_Code" (autorski rad, 2020)

Omjer muških i ženskih osoba u anonimiziranom datasetu je poprilično održan.

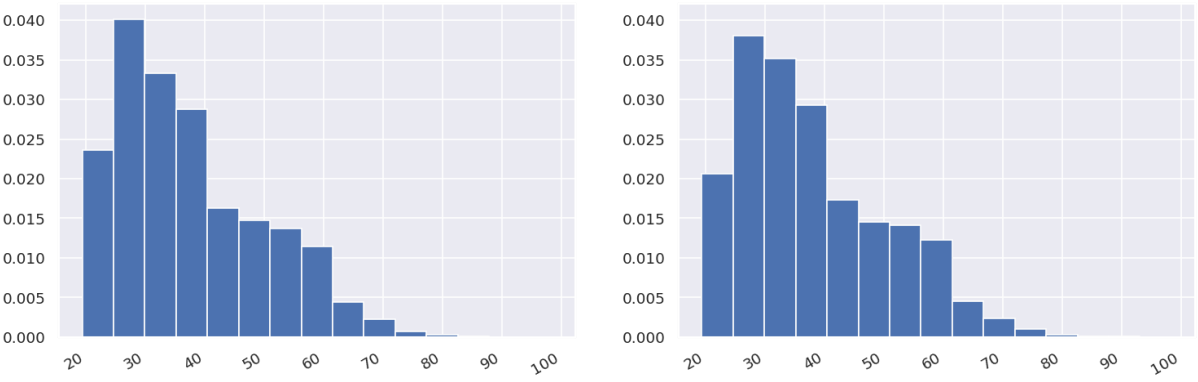
Ethnic_Code_Text



Slika 14: Usporedba histograma atributa "Ethnica_Code_Text" (autorski rad, 2020)

Kod distribucija vrijednosti za ovaj atribut, gotovo da i nema razlika između anonimiziranog i izvornog skupa. Dodatno, s histograma se može očitati da postoji vrlo mali broj vrijednosti "Arabic", "Native American" i "Oriental". Ova će pojava trebati biti zbrinuta kasnije. Također, s ovih histograma možemo primjetiti da se u datasetu zaista nalazi najviše osoba crnačke populacije, s obzirom na to da je vjerojatnost da je osoba crnačkog podrijetla najveća u usporedbi s vjerojatnostima ostalih rasa, zbog čega u konačnici i dolazi do pristranosti rezultata modela strojnog učenja.

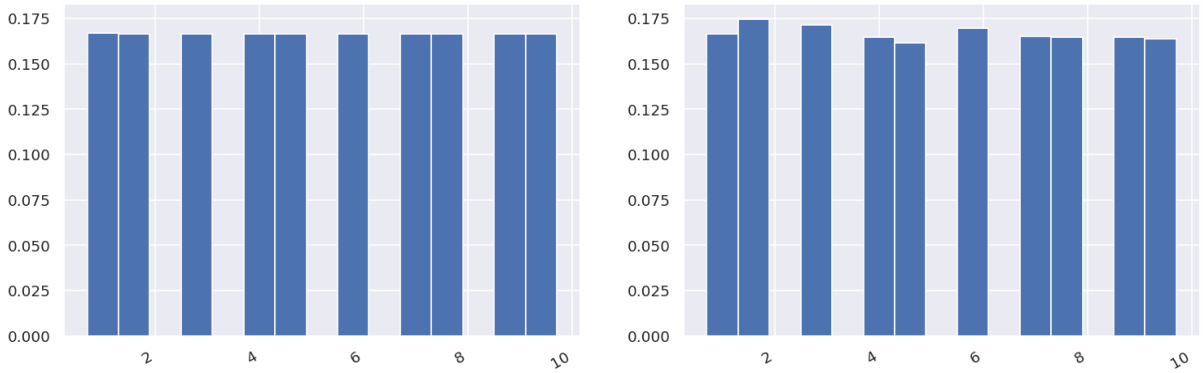
Age



Slika 15: Usporedba histograma atributa "Age" (autorski rad, 2020)

Ovdje postoje već vidljivije razlike u distribucijama između anonimiziranog i izvornog skupa. Dodatno upozorenje može se očitati i kroz vrlo male vjerojatnosti da je osoba starija od 65 godina, što upućuje da se u datasetu nalazi najmanje osoba te životne dobi i podaci o njima predstavljaju veći rizik od identifikacije. Grupiranje u veće starosne skupine može pomoći pri rješavanju ovog problema.

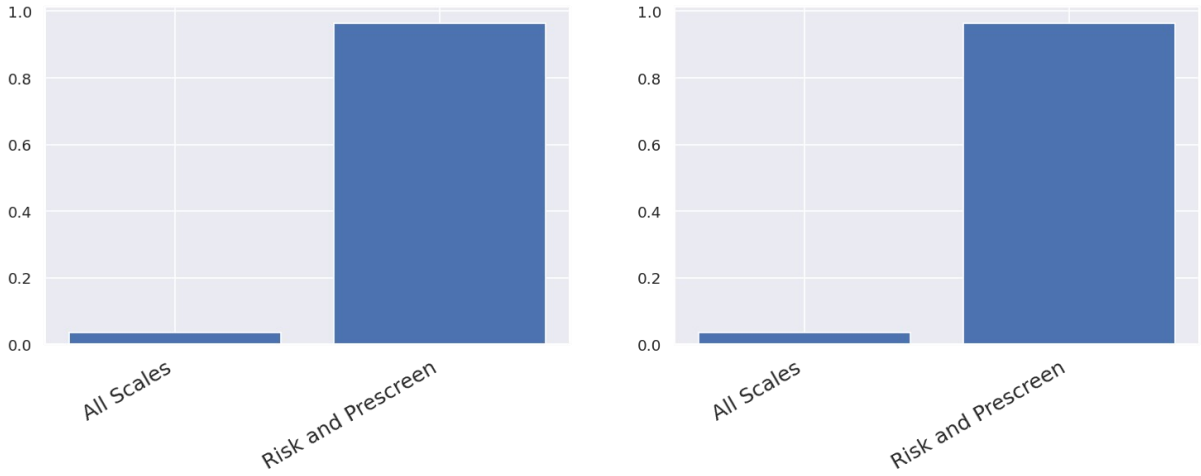
PostcodeDecile



Slika 16: Usporedba histograma atributa "PostcodeDecile" (autorski rad, 2020)

Ovdje se možda i najbolje može primijetiti efekt diferencijalne privatnosti i šuma koji se njome stvara. Ispada da su za izvorni dataset poštanski brojevi generirani, od strane Faker modula, prema uniformnoj distribuciji. Na histogramu s desne strane vidi se da distribucija poštanskih kodova, **nakon anonimiziacije**, više nije u potpunosti uniformna.

ScaleSet

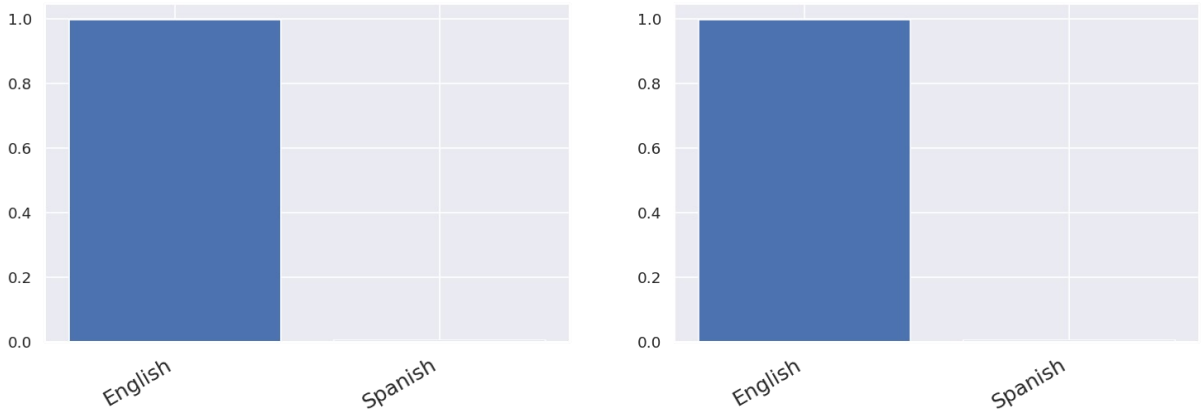


Slika 17: Usporedba histograma atributa "ScaleSet" (autorski rad, 2020)

AssessmentReason

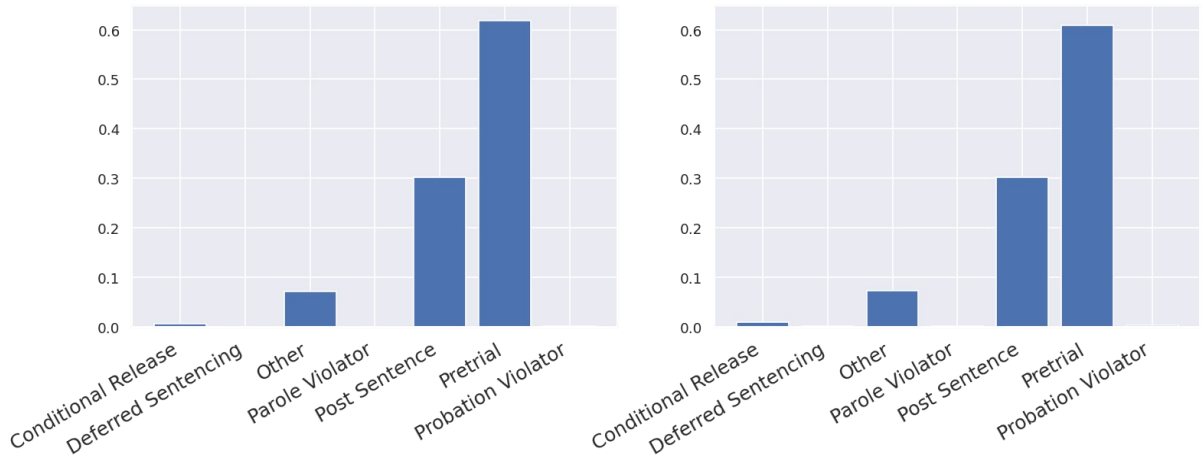
Vrijednosti svih zapisa su iste, s obzirom da se mogla generirati samo jedna vrijednost.

Language



Slika 18: Usporedba histograma atributa "Language" (autorski rad, 2020)

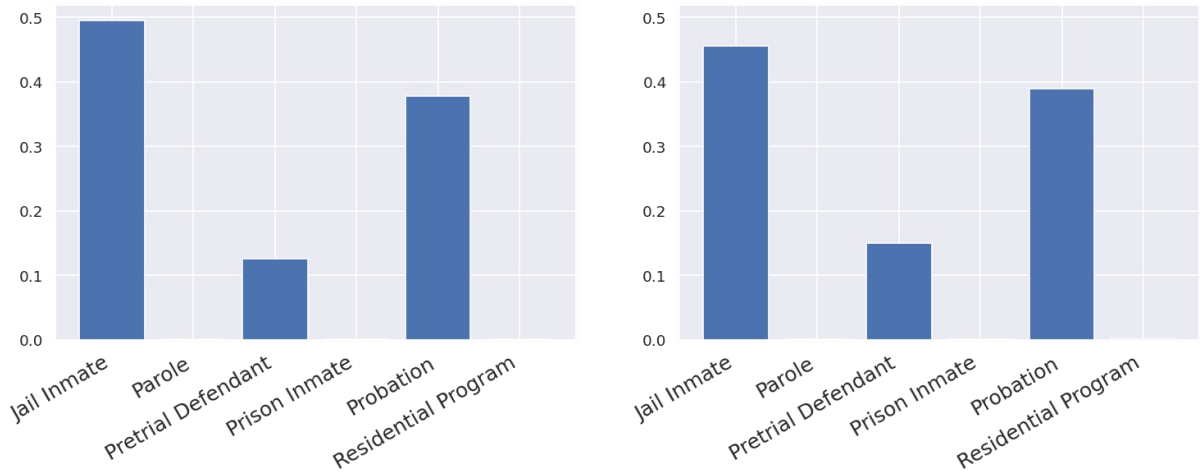
Primjer još jednog atributa kod kojeg će se vjerojatno mali broj zapisa koji uključuju španjolski jezik trebati sanirati.



Slika 19: Usporedba histograma atributa "LegalStatus"

LegalStatusTakođer se može očitati vjerojatno mali broj zapisa s određenim vrijednostima, koji će trebati biti sanirani.

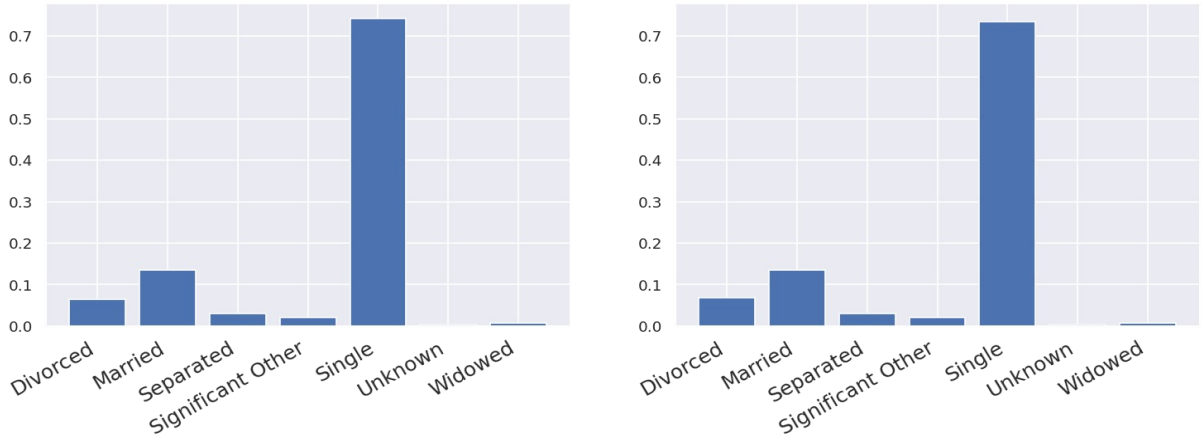
CustodyStatus



Slika 20: Usporedba histograma atributa "CustodyStatus"

Također se javlja problem vjerojatno malog broja zapisa s određenim vrijednostima. Biti će potrebno provjeriti koliko je taj broj zaista malen, te predstavlja li rizik za anonimiziaciju.

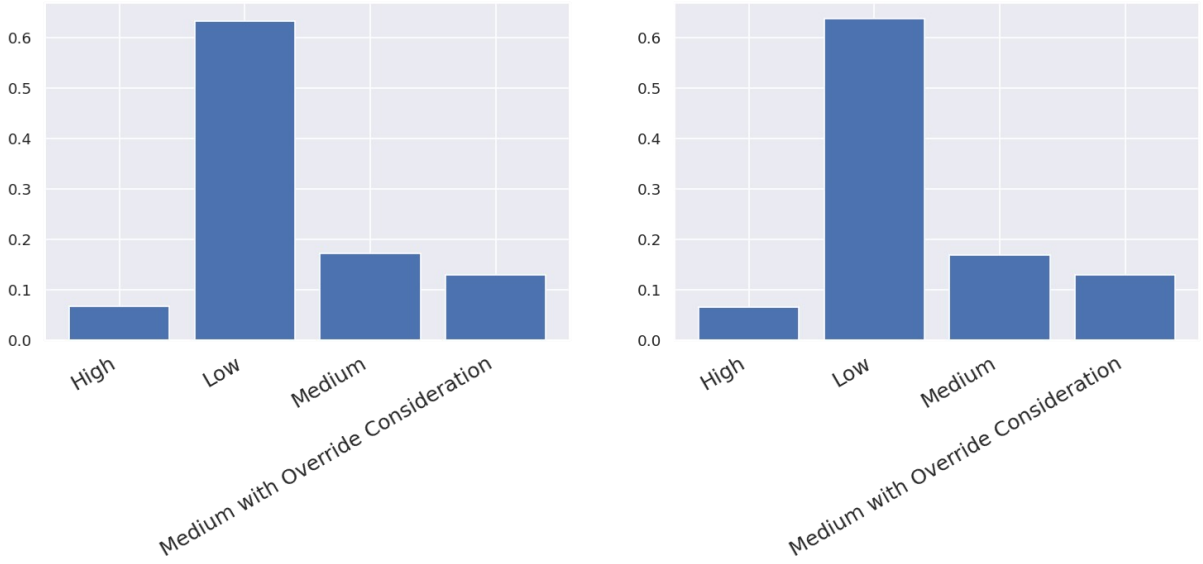
MaritalStatus



Slika 21: Usporedba histograma atributa "MaritalStatus"

Također se javlja problem malog broja zapisa.

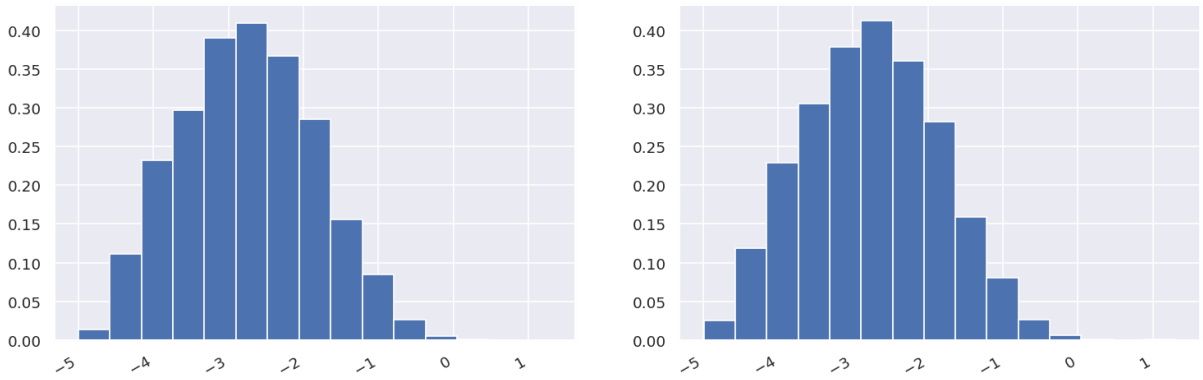
RecSupervisionLevelText



Slika 22: Usporedba histograma atributa "RecSupervisionLevelText" (autorski rad, 2020)

Distribucije su dobro očuvane.

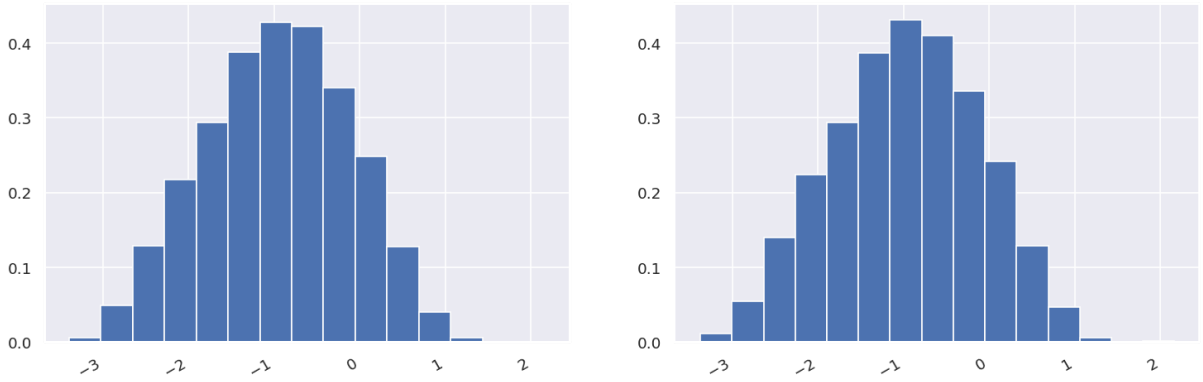
rov_raw



Slika 23: Usporedba histograma atributa "rov_raw"

Distribucije su dobro očuvane.

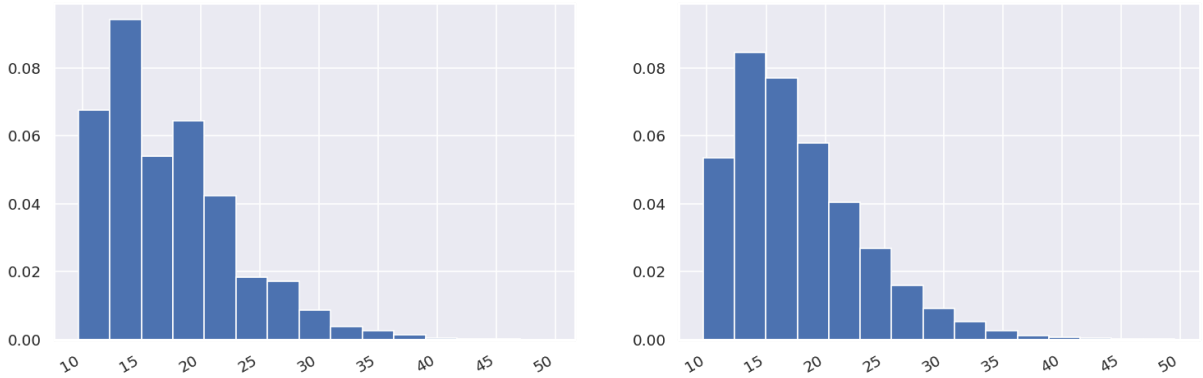
ror_raw



Slika 24: Usporedba histograma atributa "ror_raw"

Distribucije su dobro očuvane.

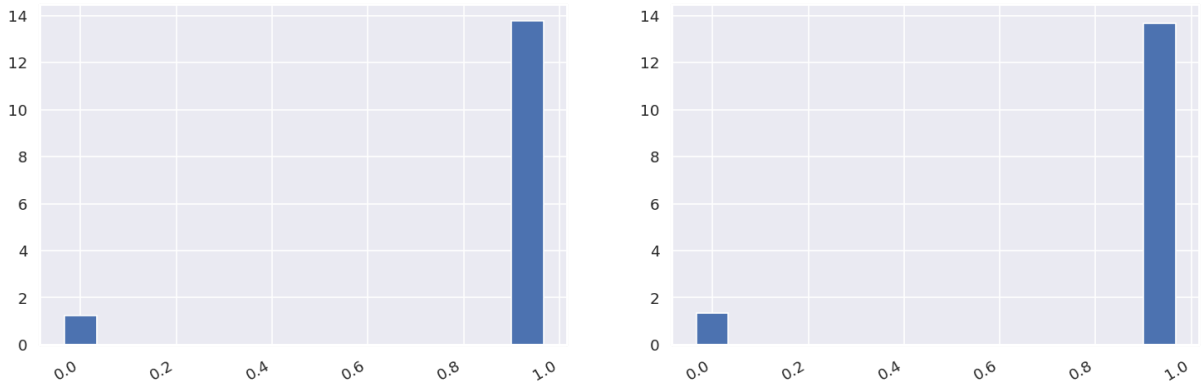
rofa_raw



Slika 25: Usporedba histograma atributa "rofa_raw"

Mogu se primjetiti razlike u distribucijama u rasponu vrijednosti od 25 do 40. Ovdje se također nazire problem gdje se visoke vrijednosti javlja zaista vrlo rijetko, što bi moglo predstavljati rizik za anonimizaciju. Dodatan šum mogao bi popraviti situaciju.

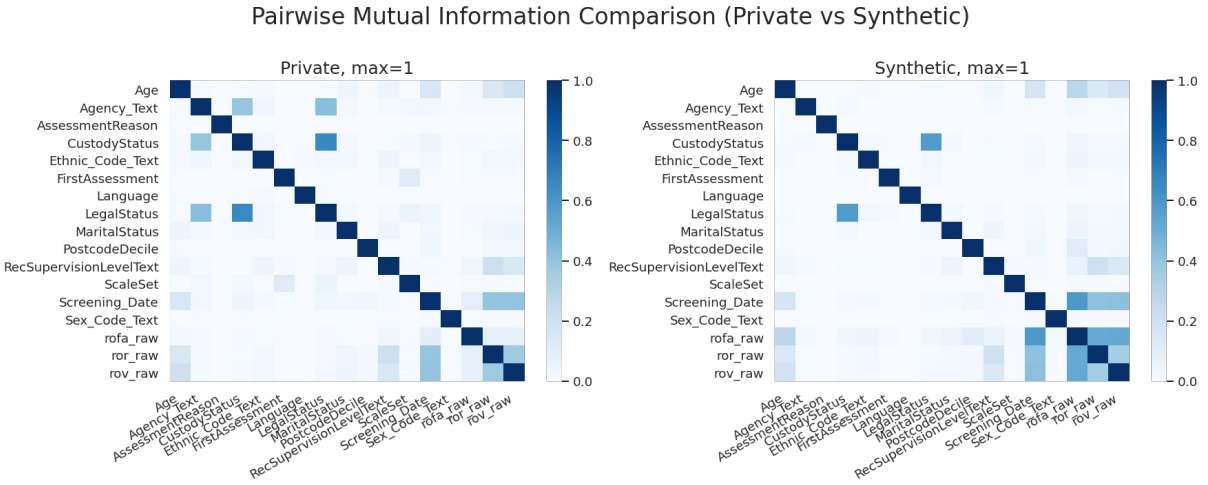
FirstAssessment



Slika 26: Usporedba histograma atributa "FirstAssessment" (autorski rad, 2020)

Omjer ponovljenih vrijednosti poprilično se precizno održao.

Toplinska mapa



Slika 27: Usporedba toplinskih mapa korelacije atributa izvornog (lijevo) i anonimiziranog (desno) skupa podataka (autorski rad, 2020)

(što je boja na poziciji a_{ij} tamnija, to pripadni atributi bolje koreliraju)

Odmah je vidljivo kako se na toplinskoj mapi, slika 27, anonimiziranog dataseta u drugom kvadrantu više ne javljaju toliko izražene korelacije nekih atributa kao u slučaju izvornog dataseta, no zato se povećane korelacije mogu očitati u četvrtom kvadrantu. Razlog tome može biti nepreciznost modela Bayesove mreže ili pak efekt diferencijalne privatnosti. U svakom slučaju, većina korelacija atributa je očuvana.

5.1.1. Posljedni korak u anonimizaciji

Usporedbom histograma kvalitativnih varijabli, uvidjeli smo da poneki atributi sadrže vrlo malen broj nekih konkretnih vrijednosti. Kod ovog se problema javlja pojava u datasetu koja omogućuje zbog vrlo malog broja zapisa, koji imaju određenu vrijednost nekog atributa, npr. vrijednost "Spanish" za atribut "Language", napadaču da raspozna identitet osobe. Iako je proces anonimizacije sproveden generiranjem sintetičkih podataka, posotji mogućnost da zbog korelacije atributa neki zapisi i dalje ostanu korisni za napadača. Ovaj se problem može rješiti na više načina, npr. otklanjanjem zapisa koji predstavljaju vrlo mali skup zapisa s određenom vrijednošću nekog atributa ili pak transformacijom vrijednosti tih zapisa u sljedeću najsmisleniju vrijednosti za koju postoji veći broj zapisa. Za otkrivanje zapisa koji

trebaju biti zbrinuti, potrebno je najprije istražiti točne brojeve jedinstvenih vrijednosti pojedinog atributa, te dodati, npr., proizvoljno pravilo da svi zapisi s određenom vrijednossću nekog atributa, moraju biti zbrinuti, ako njihov broj ne prelazi 30. Potreba za ovim postupkom mogla se prepoznati već prilikom proučavanja histograma pojedinih atributa anonimiziranog dataseta sa sintetičkim vrijednostima. U tom se dijelu jasno može očitati s grfikona koje skupove vrijednosti pojedinih atributa treba pobliže razmotriti, s obzirom na njihov udio u ukupnom broju vrijednosti pripadnog atributa. Ovaj će postupak biti izveden nad svim atributima koji predstavljaju kvalitativne varijable. Korisno je napomenuti kako se ovaj postupak može primjeniti i nad atributima "Age" te nad atributima rizika s decilnim ili tekstualnim oznakama kako bi se uklonili malobrojni zapisi s vrijednostima iz određenih raspona. To ovdje nije učinjeno, no ti se slučajevi rješavaju u dijelu s implementiranjem anonimizacije izravno u bazi podataka, gdje se vrijednosti godina poopćuju, a u vrijednosti atributa rizika dodaje se dodatan šum.

U prilogu 4 je predstavljeno potpuno programsko rješenje istraživanja malog broja vrijednosti kod svih atributa, a u ovome će dijelu biti prikazani samo atributi nad čijim je vrijednostima vršena intervencija.

Konkretno, radi se o atributima „Ethnic_Code_Text”, „LegalStatus” i „CustodyStatus”. Slika 28 prikazuje frekvencije pojava svake od jedinstvenih vrijednosti atributa „Ethnic_Code_Text”. Vidimo da broj zapisa s vrijednostima "Arabic" ili "Oriental" ne premašuje brojku 30, stoga će oni biti svedeni na vrijednosti "Other", obzirom na to što je ta vrijednost dovoljno općenita.

```
African-American    8879
Caucasian           7371
Hispanic            2941
Other                870
Asian                101
Native American       74
Arabic                26
Oriental              19
Name: Ethnic_Code_Text, dtype: int64
```

Slika 28: Frekvencije pojava anonimiziranih vrijednosti atributa "Ethnic_Code_Text" (autorski rad, 2020)

Slika 29 prikazuje frekvencije pojava svake od jedinstvenih vrijednosti atributa „LegalStatus”. Kako je broj zapisa s vrijednošću "Deferred Sentencing" atributa "LegalStatus" manji od 30,

potrebno je ukloniti te zapise, s obzirom na to što oni ne mogu biti svrstani pod neku drugu, općenitiju kategoriju.

```
Pretrial               12343
Post Sentence           6133
Other                   1477
Conditional Release      193
Probation Violator        64
Parole Violator           46
Deferred Sentencing       25
Name: LegalStatus, dtype: int64
```

Slika 29: Frekvencije pojava anonimiziranih vrijednosti atributa "LegalStatus" (autorski rad, 2020)

Slika 30 prikazuje frekvencije pojava svake od jedinstvenih vrijednosti atributa „CustodyStatus”. U slučaju atributa "CustodyStatus", također postoji mali broj zapisa sa vrijednostima "Prison Inmate" i "Parole" koje treba ukloniti.

```
Jail Inmate            9243
Probation              7907
Pretrial Defendant     3025
Residential Program      38
Name: CustodyStatus, dtype: int64
```

Slika 30: Frekvencije pojava anonimiziranih vrijednosti atributa "CustodyStatus" (autorski rad, 2020)

Sljedeći isječak koda prikazuje intervencije vršene nad navedenim atributima.

```
# 1) Ethnic_Code_Text
df_synth_anon_correlated.Ethnic_Code_Text.value_counts()

# 1.2) Saniranje malog broja zapisa s vrijednostima "Arabic" i "Oriental" - KORAK ANONIMIZACIJE
df_synth_anon_correlated.loc[df_synth_anon_correlated.Ethnic_Code_Text == "Arabic", "Ethnic_Code_Text"]
= "Other"
df_synth_anon_correlated.loc[df_synth_anon_correlated.Ethnic_Code_Text == "Oriental",
"Ethnic_Code_Text"] = "Other"


# 2) LegalStatus
df_synth_anon_correlated.LegalStatus.value_counts()

# 2.1) Uklanjanje malog broja zapisa s vrijednostima "Deferred Sentencing"
df_synth_anon_correlated.drop(
df_synth_anon_correlated.loc[df_synth_anon_correlated["LegalStatus"] == "Deferred Sentencing"].index,
inplace=True)
# 3) CustodyStatus
```

```
df_synth_anon_correlated.CustodyStatus.value_counts()

# 3.1) Uklanjanje malog broja zapisa s vrijednostima "Prison Inmate" i "Parole" - KORAK ANONIMIZACIJE
df_synth_anon_correlated.drop(
df_synth_anon_correlated.loc[df_synth_anon_correlated["CustodyStatus"]   ==   "Prison   Inmate"].index,
inplace=True)

df_synth_anon_correlated.drop(
df_synth_anon_correlated.loc[df_synth_anon_correlated["CustodyStatus"]       ==       "Parole"].index,
inplace=True)
```

Ovime je završen proces anonimizacije generiranjem sintetičkih podataka uz očuvanje korelacija atributa. Sljedi opis procesa anonimizacije za potrebe razvoja, testiranja i održavanja aplikacijskog rješenja.

6. Provedba anonimizacije za potrebe razvoja, testiranja i održavanja aplikacijskog rješenja

Za potrebe demonstracije procesa anonimizacije dataseta u svrhu razvoja, testiranja i održavanja programskom rješenja, zamišljen je scenario u kojem ugovoreni izvršitelj obrade razvija mobilnu aplikaciju preko koje korisnici mogu pratiti izračunate procjene vlastitih rizika od počinjenja kaznenog djela. Motivacija za anonimizacjiom i zaštitom tih podataka proizlazi iz činjenice kako se toj trećoj strani ne može u potpunosti vjerovati. No, aplikacijsko rješenje treba biti razvijeno, a za potrebe razvoja i testiranja potrebni su realni podaci, ili barem podaci koji svojim vrijednostima i tipom podataka oponašaju realne podtake. Generiranje takvog skupa podataka predstavlja jednu od središnjih svrha anonimizacije u poslovnom okruženju gdje osobni i osjetljivi podaci moraju kolati između ureda organizacije, ili pak između organizacije i ugovorenih partnera. Za taj je proces anonimizacije najprije potrebno istražiti prirodu i karakteristike pojedinih atributa dataseta, što je učinjeno u dijelu anonimizacije dataseta za potrebe analize i obrade podataka, odnosno za potrebe strojnog učenja. Nakon što su poznate domene i tipovi podataka atributa te vrste statističkih varijabli koje oni opisuju, potrebno je izgenerirati dataset s nasumičnim podacima, po uzoru na izvorne podatke. Dakle, za potrebe razvoja aplikacijskog rješenja trebamo sve atribute, ali ne trebamo održati nikakve korelacije između njih. Dakle, potrebni su nasumični podaci s realnim vrijednostima, koje se mogu očekivati u produkcijskom okruženju - dakle, potrebni su podaci koji **samo izgledaju** stvarno.

Tehnički izazovi generiranja realnog skupa podataka koji se ovdje nameću su sljedeći: **1)** ponavljanja zapisa za istu osobu; dakle, da osoba može vidjeti sve procjene koje su se vršile nad njome **2)** konzistentnost punog imena, email adrese i spola **3)** očekivane vrijednosti za procjene rizika, tj. da se procjenjene vrijednosti rizika nalaze u stvarnim, očekivanim rasponima.

Slijede prijedlozi rješenja za pojedini problem:

1) - problem će biti rješen izračunavanjem koliki se broj IB-jeva osoba ponavlja u datasetu, te iskoristiti taj broj za izračun vjerojatnosti da se neki zapis ponovno odnosi na istu osobu: ta će se vjerojatnost koristiti prilikom generiranja nasumičnog IB-a osobe, konkretno, predstavljati će vjerojatnost da će se *(n+1)-vi* zapis odnositi na istu osobu na koju se odnosio i n-ti zapis **2)** - kod rješavanja ovog problema, koristiti će se Faker programski modul, pomoću kojeg će se konzistentno generirati ženska ili muška imena, već prema tome koji će spol biti nasumično izabran. Nećemo održavati omjer muških i ženskih osoba iz izvornog

dataseta **3)** - ovaj će problem biti rješen na način da se izračunaju minimalne i maksimalne vrijednosti za svaku vrstu rizika. Na taj će se način dobiti realan raspon brojeva iz kojeg se mogu nasumično generirati procjenjene vrijednosti rizika.

Rješenje problema 1) prikazano je u sljedećem isječku koda. Konkretno, vjerojatnost da se *(n+1)-vi* zapis odnosi na osobu iz *n-tog* zapisa dobivena je dijeljenjm broja osoba za koje postoji više od jedne procjene u izvornom datasetu, s ukupnim brojem osoba za koje postoji barem jedna procjena u izvornom datasetu.

```
# izvlačenje jedinstvenih IB-eva osoba
list_all_person_id = df_new_compas_dev.Person_ID.values.tolist()

# izrada skupa IB-eva osoba koji se odnose na osobe za koje postoji više od 1 procjene u datasetu
duplicates_set = set([p_id for p_id in list_all_person_id if list_all_person_id.count(p_id) > 1])

# broj osoba za koje postoji više od jedne procjene u datasetu
num_duplicates = len(duplicates_set)
num_duplicates # 1533

# ukupni broj osoba za koje postoji barem jedan zapis
num_total_persons = len(df_new_compas_dev.Person_ID.unique().tolist())
num_total_persons #18610

# izračun vjerojatnost da se neki zapis odnosi na osobu za koju već postoji zapis o procjeni
prob_repeat = num_duplicates / num_total_persons
prob_repeat = round(prob_repeat, 5) # zaokruženo na 5 decimala
print(f"{prob_repeat=}") # prob_repeat=0.08238

# izračun vjerojatnost da se neki zapis NE odnosi na osobu za koju već postoji zapis o procjeni,
# tj. da se zapis o procjeni odnosi na potpuno "novu" osobu
prob_not_repeat = 1 - prob_repeat
print(f"{prob_not_repeat=}") # prob_not_repeat=0.91762
```

Potrebno je uvesti sljedeće jednostavno pravilo koje će upućivati da se nasumičnim odabirom jedinice (1) iz skupa {0, 1} određuje da se sljedeći zapis odnosi na istu osobu kao i prethodni. Tako se za vrijednost 0 (novi zapis) koristi vjerojatnost odabira 0.91762, dok se za vrijednost 1 (zapis se odnosi na već postojeću osobu) koristi vjerojatnost odabira 0.08238.

Nakon ovog dijela, potrebno je iz izvornog dataseta dohvatiti skupove jedinstvenih vrijednosti pojedinih atributa, istraženih u prošlom poglavlju. Taj je proces prikazan na sljedećem isječku koda.

```
# Jedinstvene vrijednosti atributa "AssessmentReason"
assessment_reason_unq = list(df_new_compas_dev.AssessmentReason.unique())

# Jedinstvene vrijednosti atributa "Language"
lang_unq = list(df_new_compas_dev.Language.unique())

# Jedinstvene vrijednosti atributa "LegalStatus"
legal_stat_unq = list(df_new_compas_dev.LegalStatus.unique())

# Jedinstvene vrijednosti atributa "CustodyStatus"
custody_stat_unq = list(df_new_compas_dev.CustodyStatus.unique())
```

```
12
13 # Jedinstvene vrijednosti atributa "MaritalStatus"
14 marital_stat_unq = list(df_new_compas_dev.MaritalStatus.unique())
15
16 # Jedinstvene vrijednosti atributa "Sex_Code_Text"
17 sex_code_unq = list(df_new_compas_dev.Sex_Code_Text.unique())
18
19 # Jedinstvene vrijednosti atributa "Ethnic_Code_Text"
20 ethnic_code_unq = list(df_new_compas_dev.Ethnic_Code_Text.unique())
21
22 # Jedinstvene vrijednosti atributa "ScaleSet_ID"
23 scale_set_unq = list(df_new_compas_dev.ScaleSet.unique())
24
25 # Jedinstvene vrijednosti atributa "Agency_Text"
26 agency_text_unq = list(df_new_compas_dev.Agency_Text.unique())
27
28 # Jedinstvene vrijednosti atributa "RecSupervisionLevel"
29 rec_super_lvl_unq = list(df_new_compas_dev.RecSupervisionLevelText.unique())
```

6.1. Generiranje nasumičnih sintetičkih vrijednosti

Za potrebe nasumičnog generiranja podataka, definirane su mnoge funkcije. Zbog sažetosti, funkcije su opisane komentarima u sljedećem isječku koda. No, potrebno je napomenuti korištenje metoda biblioteka ***Numpy*** i ***Faker***. Numpy se pretežito koristi za generiranje vrijednosti iz konačnog skupa, prema zadanim vjerojatnostima ili pak za generiranje vrijednosti prema zadanoj distribuciji, npr. uniformnoj. Za navedene sluačjeve koriste se metode *choice()* ili *choices()*, te metoda *uniform*(), respektivno. Sve navedene metode dio su Numpy-evog modula *random*. Metode biblioteke Faker korištene su za generiranje izravnih i kvazi-identifikator, poput imena, prezimena, e-mail adresa, poštanskih brojeva, itd. Bitno je još naglasiti kako će se za potrebe generiranja zapisa koji se odnosi na osobu iz prošlog, podaci prošlog zapisa spremati u varijablu „*prev_record*”, koja predstavlja rječnik sa podacima prošlog zapisa. Ako se izračuna da se novi zapis odnosi na već postojeću osobu, tada će navedena varijabla koristiti za kopiranje podataka o sljedećim izravnim i kvazi-identifikatorima: "Person_ID", "FirstName", "MiddleName", "LastName", "Email", "Address", "Postcode", "Sex_Code_Text", "Ethnic_Code_Text", "DateOfBirth", "Language" i "MaritalStatus". Nakon generiranja nasumičnih vrijednosti, još će biti potrebno u dataset uvrstiti preslikavanja numeričkih i tekstualnih oznaka kvalitativnih varijabli „ScaleSet” i „RecSupervisionLevel”. Također, prije početka generiranj podataka, potrebno je odrediti raspone vrijednosti neprekidnih numeričkih varijabli rizika i datuma rođenja, odnosno izvršenja procjene.

Sljedeći isječak koda predstavlja proces generiranja nasumičnih sintetičkih podataka.

```
# 1) Određivanje datumskih realnih datumskih raspona vrijednosti temporalnih atributa
#-------------------------------------------------------------------------------------
# datum rođenja
dob_start = datetime.date.fromisoformat("1940-01-01")
dob_end = datetime.date.fromisoformat("2000-12-31")

# datum procjene
assess_date_start = datetime.date.fromisoformat("2012-01-01")
assess_date_end = datetime.date.fromisoformat("2013-01-01")

# 2) Učitavanje raspona vrijednosti atributa rizika
#--------------------------------------------------
# 2.1) Rizik od nasilja
#----------------------
# min, max vrijednosti za rizik od nasilja
rov_min, rov_max = df_new_compas_dev.rov_raw.min(), df_new_compas_dev.rov_raw.max()

# mapirani rasponi rizika od nasilja i tekstualne oznake
rov_txt_map_list = pd.read_csv("./data/my_datasets/rov_description.csv").to_dict("records")

# 2.2) Rizik od recidivizma
#--------------------------
# min, max vrijednosti za rizik od recidivizma
ror_min, ror_max = df_new_compas_dev.ror_raw.min(), df_new_compas_dev.ror_raw.max()

# mapirani rasponi rizika od recidivizma i tekstualne oznake
ror_txt_map_list = pd.read_csv("./data/my_datasets/ror_description.csv").to_dict("records")

# 2.2) Rizik od nepojavljivanja na sudu
#--------------------------------------
# min, max vrijednosti za rizik od nepojavljivanja na sudu
rofa_min, rofa_max = df_new_compas_dev.rofa_raw.min(), df_new_compas_dev.rofa_raw.max()

# mapirani rasponi rizika od nepojavljivanja na sudu i tekstualne oznake
rofa_txt_map_list = pd.read_csv("./data/my_datasets/rofa_description.csv").to_dict("records")

# 3) Definicije potrebnih funkcija za generiranje podataka
#----------------------------------------------------------
# funkcija za dohvaćanje sljedećeg po redu broja za neki IB
def get_next_id(previous_id):
    next_id = previous_id + 1
    return next_id

# funkcija za generiranje datuma rođenja iz vremenskog raspona
# kao argumente prima prvi i zadnji datum raspona, respektivno, te Faker objekt
def get_date_of_birth(dob_start_date, dob_end_date, faker_generator):
    new_dob = faker_generator.date_between_dates(dob_start_date, dob_end_date)
    new_dob = new_dob.strftime("%m/%d/%Y")
    return new_dob

# funkcija za generiranje datuma provedene procjene iz vremenskog raspona
def get_assessment_date(assess_start_date, assess_end_date, faker_generator):
    new_assessment_date = faker_generator.date_time_between(assess_start_date, assess_end_date)
    new_assessment_date = new_assessment_date.strftime("%m/%d/%Y %H:%M")
    return new_assessment_date

# funkcija za dohvaćanje tekstualne oznake za rizik, koja će se prikazati korisniku,
# zajedno s iznosom procjenjene vrijednosti rizika - pripadni decil neće biti prikazani.
# kao argument prima vrijednost rizika i listu rječnika sa preslikanim text vrijednostima
def get_risk_txt_label(risk_raw_val, risk_txt_map):
    risk_txt_label = ""
    for i in range(len(risk_txt_map)):
        risk_min = risk_txt_map[i]["raw_min"]
        risk_max = risk_txt_map[i]["raw_max"]
        if (risk_raw_val >= risk_min) and (risk_raw_val <= risk_max):
            risk_txt_label = risk_txt_map[i]["txt_label"]
            break
        else:
            continue

    return risk_txt_label

# funkcija koja vraća "ime, srednje ime i prezime, ovisno o spolu.
# kao argument prima spol osobe te Faker objekt
def get_full_name(person_gender, fake_generator):
    # nasumična odluka ima li osoba srednje ime
    middle_name = np.random.choice(np.arange(0, 2))

    if person_gender == "Female":
       first_name = fake_generator.first_name_female()
        last_name = fake_generator.last_name_female()
        if middle_name:
            middle_name = fake_generator.first_name_female()
        else:
            middle_name = np.NaN

    else:
```

```
            first_name = fake_generator.first_name_male()
            last_name = fake_generator.last_name_male()
            if middle_name:
                middle_name = fake_generator.first_name_male()

            else:
                middle_name = np.NaN

        return (first_name, middle_name, last_name)


    # funkcija koja stvara email adresu za proslijeđeno ime, srednje ime i prezime osobe
    # kao argument prima ime, srednje ime i prezime osobe, respektivno, te Faker objekt
    def get_email(first_name, middle_name, last_name, fake_generaor):
        random_email_domain = fake_generaor.free_email_domain()

        if pd.isnull(middle_name):
            email = first_name[0].lower() + "." + last_name.lower() + "@" + random_email_domain

        else:
            email = first_name[0].lower() + middle_name[0].lower() + "." + last_name.lower() + "@" + random_email_domain

        return email


    # funkcija koja vraća vjerojatnost odnosi li se novi zapis na osobu iz prošlog zapisa
    def next_person_repeats(prob_repeats, prob_not_repeats):
        probabilities = [prob_repeats, prob_not_repeats]
        repeats = np.random.choice([1, 0], p = probabilities)
        if repeats == 0:
            return False
        else:
            return True

    # kopira zajedničke atribute iz staroga u novi zapis iste osobe.
    # kao argumente prima stari i novi zapis, respektivno
    def copy_prev_to_new(prev_rec, new_rec):
        new_rec["Person_ID"] = prev_rec["Person_ID"]
        new_rec["FirstName"] = prev_rec["FirstName"]
        new_rec["MiddleName"] = prev_rec["MiddleName"]
        new_rec["LastName"] = prev_rec["LastName"]
        new_rec["Email"] = prev_rec["Email"]
        new_rec["Address"] = prev_rec["Address"]
        new_rec["Postcode"] = prev_rec["Postcode"]
        new_rec["Sex_Code_Text"] = prev_rec["Sex_Code_Text"]
        new_rec["Ethnic_Code_Text"] = prev_rec["Ethnic_Code_Text"]
        new_rec["DateOfBirth"] = prev_rec["DateOfBirth"]
        new_rec["Language"] = prev_rec["Language"]
        new_rec["MaritalStatus"] = prev_rec["MaritalStatus"]

        return new_rec

    # 4) Inicijaliziranje početnih varijabli
    #---------------------------------------
    # inicijalizacije
    random_anon_dset_length = 21000 # stvaramo 21000 zapisa
    fake_generator = faker.Faker()
    prob_repeat = 0.08238
    prob_not_repeat = 0.91762
    records_list = [] # sadrži nove zapise
    prev_record = {} # služi za trenutnu pohranu podataka "prošlog" zapisa osobe

    # na početku inicijaliziramo vrijednosti IB-jeva, te ih odmah inkrementiramo u prvoj iteraciji
    last_assessment_id = 99
    last_person_id = 399
    last_case_id = 699

    # 5) Generiranje nasumičnih podataka
    #-----------------------------------
    for i in range(0, random_anon_dset_length):
        new_rec = dict()

        last_assessment_id = get_next_id(last_assessment_id)
        last_case_id = get_next_id(last_case_id)

        new_rec["AssessmentID"] = last_assessment_id
        new_rec["Case_ID"] = last_case_id
        new_rec["Agency_Text"] = np.random.choice(agency_text_unq)
        new_rec["ScaleSet"] = np.random.choice(scale_set_unq)
        new_rec["AssessmentReason"] = np.random.choice(assessment_reason_unq)
        new_rec["LegalStatus"] = np.random.choice(legal_stat_unq)
        new_rec["CustodyStatus"] = np.random.choice(custody_stat_unq)
        new_rec["RecSupervisionLevelText"] = np.random.choice(rec_super_lvl_unq)
        new_rec["Screening_Date"] = get_assessment_date(assess_date_start, assess_date_end, fake_generator)
        new_rec["rov_raw"] = round(np.random.uniform(rov_min, rov_max), 2)
        new_rec["rov_score_txt"] = get_risk_txt_label(new_rec["rov_raw"], rov_txt_map_list)
        new_rec["ror_raw"] = round(np.random.uniform(ror_min, ror_max), 2)
```

```
        new_rec["ror_score_txt"] = get_risk_txt_label(new_rec["ror_raw"], ror_txt_map_list)
        new_rec["rofa_raw"] = round(np.random.uniform(rofa_min, rofa_max), 0)
        new_rec["rofa_score_txt"] = get_risk_txt_label(new_rec["rofa_raw"], rofa_txt_map_list)


        # izračunavanje hoće li se trenutni zapis o procjeni odnositi na
        # osobu iz prošlog zapisa
        if next_person_repeats(prob_repeat, prob_not_repeat) and (i > 0):
                new_record = copy_prev_to_new(prev_record, new_rec)

        else:
            last_person_id = get_next_id(last_person_id)
            new_rec["Person_ID"] = last_person_id

            sex_label = np.random.choice(sex_code_unq)

            first_name, middle_name, last_name = get_full_name(sex_label, fake_generator)

            email = get_email(first_name, middle_name, last_name, fake_generator)

            new_rec["FirstName"] = first_name
            new_rec["MiddleName"] = middle_name
            new_rec["LastName"] = last_name
            new_rec["Email"] = email
            new_rec["Address"] = fake_generator.street_address()
            new_rec["Postcode"] = fake_generator.postcode()
            new_rec["Sex_Code_Text"] = sex_label
            new_rec["Ethnic_Code_Text"] = np.random.choice(ethnic_code_unq)
            new_rec["DateOfBirth"] = get_date_of_birth(dob_start, dob_end, fake_generator)
            new_rec["Language"] = np.random.choice(lang_unq)
            new_rec["MaritalStatus"] = np.random.choice(marital_stat_unq)

        records_list.append(new_rec)
        prev_record = new_rec

    # 6) Stvaranje novog dataseta iz generiranih nasumičnih zapisa
    #-------------------------------------------------------------
    # stvaranje novog Pandas DataGrame objekta
    df_randomized_anon = pd.DataFrame.from_records(records_list)


    # 7) Dodavanje preslikavanja za atribute "ScaleSet" i "RecSupervisionLevelText"
    #------------------------------------------------------------------------------
    # "ScaleSet"
    df_scaleset_map = pd.read_csv("./data/my_datasets/scale_set_map.csv")
    scale_set_map = dict(zip(df_scaleset_map.ScaleSet, df_scaleset_map.ScaleSet_ID))
    df_randomized_anon["ScaleSet_ID"] = df_randomized_anon.apply(lambda x: scale_set_map[x["ScaleSet"]], axis = 1)

    # "RecSupervisionLevelText"
    df_rec_super_map = pd.read_csv("./data/my_datasets/rec_supervision_map.csv")
    rec_sup_map = dict(zip(df_rec_super_map.RecSupervisionLevelText, df_rec_super_map.RecSupervisionLevel))
    df_randomized_anon["RecSupervisionLevel"] =
    df_randomized_anon.apply(lambda x: rec_sup_map[x["RecSupervisionLevelText"]], axis = 1)

    # 8) Konačan izgled dataseta s nasumično generiranim podaicma
    #------------------------------------------------------------
    df_randomized_anon.head(3).transpose()

    # spremanje novog dataseta
    df_randomized_anon.to_csv("./data/anon_datasets/randmoized_anon_compas.csv", index = False)
```

Slika 31 predstavlja konačni izgled anonimiziranog dataseta generiranjem nasumičnih vrijednosti.

	0	1	2
AssessmentID	100	101	102
Case_ID	700	701	702
Person_ID	400	401	402
FirstName	Mary	Debra	Jennifer
MiddleName	Rebecca	Kara	NaN
LastName	Bean	Schwartz	Zhang
Email	mr.bean@yahoo.com	dk.schwartz@yahoo.com	j.zhang@yahoo.com
Address	79397 Stanley Ports	94161 Michelle Lights	32678 Hurst Falls
Postcode	93764	78808	85379
Sex_Code_Text	Female	Female	Female
Ethnic_Code_Text	Native American	Caucasian	Arabic
DateOfBirth	10/20/1999	01/24/1983	08/02/1944
Language	English	Spanish	English
MaritalStatus	Separated	Married	Single
Agency_Text	PRETRIAL	Probation	DRRD
ScaleSet	Risk and Prescreen	All Scales	All Scales
ScaleSet_ID	22	17	17
AssessmentReason	Intake	Intake	Intake
LegalStatus	Probation Violator	Post Sentence	Post Sentence
CustodyStatus	Residential Program	Prison Inmate	Residential Program
RecSupervisionLevel	4	1	3
RecSupervisionLevelText	High	Low	Medium with Override Consideration
Screening_Date	01/11/2012 21:03	06/24/2012 23:47	05/30/2012 16:29
rov_raw	-1.06	-2.48	0.35
rov_score_txt	High	Low	High
ror_raw	-0.7	-1.69	-2.19
ror_score_txt	Low	Low	Low
rofa_raw	28	27	38
rofa_score_txt	High	High	High

Slika 31: Izgled anonimiziranog dataseta generiranjem nasumičnih vrijednosti

7. Implementacija potrebnih baza podataka

Ranije je objašnjenjo kako anonimizacija podataka velikim dijelom obuhvaća i organizacijsku stranu preko organizacijskih politika i pravila između različitih odijela organizacije i restrikcija svrhe korištenja i dijeljenja podataka putem posebnih ugovora s trećim stranama koje ih koriste. Implementacija takvih poslovnih pravila i pravila pristupa podacima u svrhu kreiranja nekog informacijskog sustava često kreće upravo od definiranja prava pristupa podacima u bazama podataka. Svaki moderni sustav za upravljanje bazama podataka omogućuje različite razine implementacije prava pristupa pojedinim ulogama, tj. korisnicima koji koriste baze podataka. [24] U svrhu anonimiziacije, često je potrebno ograničiti pristup pojedinim atributima relacija čiji podaci mogu identificirati pojedinca, bilo neposredno ili kroz kombinaciju s vrijednostima drugih atributa, tj. kroz kombinaciju kvazi identifikatora. Može se zaključiti kako takav pristup zaštite podataka osigurava vrlo transparentan način praćenja protoka podataka kroz organizaciju i među poslovnim partnerima. Tako će se u ovome radu u svrhu pohrane podataka za zamišljene scenarije za koje su generirani anonimizirani podaci, koristiti tri različite baze podataka s različitim ulogama. Zamišljena je jedna baza podataka u čije će se relacije pohraniti testni podaci, naziva "dev_test_mock_db", potom jedna baza podataka za potrebe pohranjivanja podataka izvornog dataseta, nazova "central_db", te jedna baza podataka koja će zapravo predstavljati spremište podataka za anonimizirane datasetove i rezultate obrada tih datasetova, naziva "ml_warehouse". Ovako snažna podjela podataka omogućuje vrlo efektivno kontroliranje prava pristupa pojedinih uloga, tako uloge koje ne trebaju, a i ne smiju, koristiti izvorne podatke neće ni imati pristup toj bazi podataka, a samim time ni njezinim relacijama. Nadalje, pojedinim će ulogama biti odobren pristup isključivo korisnički definiranim SQL funkcijama koje će omogućiti lako i brzo dohvaćanje podataka iz relacija kojima te uloge prirodno nemaju pristup. Tako će BP "dev_test_mock_db" omogućavati pristup ulozi "dev_tester_role" koja će moći pristupati svim relacijama i funkcijama te baze podataka. Njezina je svrha omogućiti neograničeni pristup izvršitelju obrade koji razvija aplikacijsko rješenje. Sama je BP "dev_test_mock_db", što se shema relacija tiče, preslika BP "central_db".

Nadalje, BP "central_db", omogućiti će pristup ulogama "dev_production_role" i "ml_dpo_role". Ulozi "dev_production_role" biti će omogućeno izvršavanje upita nad spomenutom bazom isključivo preko korisnički definiranih funkcija *get_person_assessment_info()* i *get_person_info()*, koje će služiti za dohvaćanje podataka iz relacija koje sadrže podatke o sprovedenim "COMPAS" procjenama i osobama nad kojima su te procjene sprovedene, respektivno. Također, te će dvije funkcije prilikom dohvaćanja

zapisa iste i kriptirati. Na taj način korisnik uloge "dev_production_role", tj. ugovoreni izvršitelj, neće ni u kojem trenutku moći otkriti podatke korisnika. U tu se svrhu može zamisliti slučaj korištenja u kojem se ti podaci dekriptiraju tek neposredno prije njihovog prikazivanja na korisnikovom uređaju. Sustav kriptiranja tih podataka može zavisiti od implementacije do implementacije, npr. pomoću korisnikove lozinke i soli, ili pak pomoću sustava asimetričnog kriptiranja, tj. pomoću javnog i privatnog ključa korisnika, ili pak pomoću simetričnog ključa kriptiranja. U ovoj je implementaciji odabran pristup kriptiranja simetričnim ključem. Tom se metodom podaci kriptiraju i dekriptiraju istim korisnikovim ključem. Ovakav bi pristup u realnom svijetu predstavljao jedan sigurnosni rizik, a to je kako na siguran način korisniku dostaviti ključ i pohraniti ga na njegovu uređaju. Naravno, to je izvedivo korištenjem dodatnih sigurnosnih protokola komunikacije programske podrške korisnikovog uređaja i servera na kojem se pogoni baza podataka, ili pak kriptiranje simetričnog ključa korisnikovom lozinkom te dostavljanja tako kriptiranog ključa korisniku. U ovom je radu ovaj pristup odabran zbog jednostavnosti, obzirom na to što je svrha njegovog korištenja demonstrirati korištenje funkcija kriptiranja dostupnih u PostgreSQL SUBP.

Ulozi "ml_dpo_role" biti će pak odobreno izvršavanje upita isključivo pomoću korisnički definirane funckije *get_data_for_external_anon()*, kojom će se dohvaćati zapisi relacije s podacima o provedenim "COMPAS" procjenama, bez skupa atributa koji sadrže podatke o izravnim identifikatorima. Korisnik te uloge može biti glavna odgovorna osoba za privatnost i zaštitu podataka u organizaciji, tzv. službenik za zaštitu podataka (eng. *Data Privacy Officer*).

Baza podataka "ml_warehouse" također će imati omogućen pristup ulozi "ml_dpo_role", koja će moći izvršavati CRUD operacije nad relacijama navedene baze podataka. Također, navedena će BP omogućiti i ulogu "public_data_sharing_role", čiji će korisnici moću postavljati upite nad BP isključivo pomoću funkcije *generate_temp_public_data()*. Navedena funkcija dohvaća podatke iz relacije "assessments_report", koja sadrži anonimizirane zapise izvornog dataseta. Ti su zapisi anonimizirani u dijelu anonimizacije za potrebe analize i obrade podataka, odnosno za potrebe strojnog učenja.

Za kraj je još potrebno navesti, kako se kao sustav za upravljanje bazama podataka koristio sustav PostgreSQL, verzije 12.3. Kao dodatno rješenje za primjenu anonimizacije nad podacima izravno u BP koristilo se proširenje za PostgreSQL sustav ***PostgreSQL Anonymizer***, verzije 0.6.0, koji je dostupan na GitLab repozitoriju putem poveznice [24]. Dodatno, za dohvaćanje podataka iz PostgreSQL baza, koristio se Python programski modul "psycopg2" verzije 2.8.5. Korištenje tog programskog modula omogućuje lakše rukovanje

postalvjanjem upita nad BP i preglednijim prikazom rezultata. Kako su SQL definicije baza podataka i uloga poprilično trivijalne [25], zbog sažetosti teksta ovog rada, priložene su u prilogu 5, stoga će se ovdje prikazati samo definicije korisničkih SQL funkcija kojima se implementiraju prava pristupa.

Kao što je već napomenuto, baza podataka "dev_test_mock_db" sadrži anonimizirane skupove testnih podataka, te je po sovjoj strukturi preslika BP "central_db" koja sadrži relacije s podacima izvornog dataseta. Atributi dataseta slažu se s atributima relacija ove i "central_db" baze, stoga neće biti ponovno opisani. Bitno je napomenuti kako se podaci o provedenim procjenama bilježe u relaciju *assessments*, podaci o osobama koje su predmet kakvog kaznenog slučaja bilježe se u relaciju *people*, dakle to je relacija s osobnim podacima, te se podaci o svakom kaznenom slučaju bilježe u relaicju *cases*. Bitno je još napomenuti kako se relacija ***people_symetric_keys*** koristi za pohranu simetričnih ključeva pojedinaca iz dataseta. Ti su ključevi generirani pomoću Pythonove hash funkcije *sha256()* iz programskog modula ***hashlib***. Kao ulazni argument toj funkciji proslijeđen je nasumični niz bajtova, generiran funkcijom *urandom()* iz programskog modula ***os***. Spomenuta se relacija sastoji od atributa „person_id” i „symetric_key”, koji predstavljaju IB osobe, tj. vanjski ključ na relaciju *people*, te pripadni simetrični ključ osobe, respektivno. Slijede definicije funkcija ove baze podataka.

```
CREATE OR REPLACE FUNCTION get_person_assessment_info(arg_person_id TEXT)
    RETURNS TABLE(rov_raw TEXT, rov_txt TEXT, ror_raw TEXT, ror_txt TEXT,
                  rofa_raw TEXT, rofa_txt TEXT, screening_date TEXT,
                  legal_status TEXT, custody_status TEXT, rec_super_txt TEXT,
                  assess_reason TEXT)
AS $$
BEGIN
    RETURN QUERY
        SELECT
            (PGP_SYM_ENCRYPT(a.rov_raw::TEXT, pk.symetric_key)::TEXT) AS rov_raw,

            (PGP_SYM_ENCRYPT(a.rov_score_txt::TEXT, pk.symetric_key)::TEXT) AS rov_txt,

            (PGP_SYM_ENCRYPT(a.ror_raw::TEXT, pk.symetric_key)::TEXT) AS ror_raw,

            (PGP_SYM_ENCRYPT(a.ror_score_txt::TEXT, pk.symetric_key)::TEXT) AS ror_txt,

            (PGP_SYM_ENCRYPT(a.rofa_raw::TEXT, pk.symetric_key)::TEXT) AS rofa_raw,

            (PGP_SYM_ENCRYPT(a.rofa_score_txt::TEXT, pk.symetric_key)::TEXT) AS rofa_txt,

            (PGP_SYM_ENCRYPT(a.screening_date::TEXT, pk.symetric_key)::TEXT) AS screening_date,

            (PGP_SYM_ENCRYPT(a.legal_status::TEXT, pk.symetric_key)::TEXT) AS legal_status,

            (PGP_SYM_ENCRYPT(a.custody_status::TEXT, pk.symetric_key)::TEXT) AS custody_status,

            (PGP_SYM_ENCRYPT(a.rec_supervision_level_text::TEXT, pk.symetric_key)::TEXT) AS rec_super_txt,

            (PGP_SYM_ENCRYPT(a.assessment_reason::TEXT, pk.symetric_key)::TEXT) AS assess_reason
        FROM
            assessments a
        INNER JOIN
            people_symetric_keys pk ON a.person_id=pk.person_id
        WHERE
            a.person_id::TEXT=arg_person_id;
END;
$$ LANGUAGE plpgsql;

```

```sql
CREATE OR REPLACE FUNCTION get_person_info(arg_person_id TEXT)
    RETURNS TABLE(f_name TEXT, m_name TEXT, l_name TEXT, dob TEXT, email TEXT,
                  address TEXT, postcode TEXT, sex TEXT, ethnicity TEXT,
                  language TEXT, marital_status TEXT)
AS $$
BEGIN
    RETURN QUERY
        SELECT
            (PGP_SYM_ENCRYPT(p.first_name::TEXT, pk.symetric_key)::TEXT) AS f_name,

            (PGP_SYM_ENCRYPT(p.middle_name::TEXT, pk.symetric_key)::TEXT) AS m_name,

            (PGP_SYM_ENCRYPT(p.last_name::TEXT, pk.symetric_key)::TEXT) AS l_name,

            (PGP_SYM_ENCRYPT(p.date_of_birth::TEXT, pk.symetric_key)::TEXT) AS dob,

            (PGP_SYM_ENCRYPT(p.email::TEXT, pk.symetric_key)::TEXT) AS email,

            (PGP_SYM_ENCRYPT(p.address::TEXT, pk.symetric_key)::TEXT) AS address,

            (PGP_SYM_ENCRYPT(p.postcode::TEXT, pk.symetric_key)::TEXT) AS postcode,

            (PGP_SYM_ENCRYPT(p.sex::TEXT, pk.symetric_key)::TEXT) AS sex,

            (PGP_SYM_ENCRYPT(p.ethnicity::TEXT, pk.symetric_key)::TEXT) AS ethnicity,

            (PGP_SYM_ENCRYPT(p.language::TEXT, pk.symetric_key)::TEXT) AS language,

            (PGP_SYM_ENCRYPT(p.maritalStatus::TEXT, pk.symetric_key)::TEXT) AS marital_status
        FROM
            people p
        INNER JOIN
            people_symetric_keys pk ON p.person_id=pk.person_id
        WHERE
            p.person_id::TEXT=arg_person_id;
END;
$$ LANGUAGE plpgsql;
```

Ovdje je još potrebno komentirati funkcije iz proširenja *pgcrypto* PostgreSQL SUBP-a [26]. Ovo proširenje omogućuje korištenje kriptografskih funkcija u PostgreSQL sustavu [5]. Konkretno, ovdje je korištena funkcija PGP_SYM_ENCRYPT(), koja redom prima parametre: podatak kojeg je potrebno kriptirati, simetrični ključ kojim se podatak kriptira. Dakle, ova se funkcija koristi za kriptiranje simetričnim ključem, što znači da dolazi u paru s funkcijom *PGP_SYM_DECRYPT()*, koja se pak koristi za dekriptiranje podatka simetričnim ključem. Primjer SQL upita:

```sql
1 SELECT * FROM PGP_SYM_ENCRYPT('tajni_podatak', 'aspp12asdpk2');
```

daje rezultat sa slike 32.

```
                                                   pgp_sym_encrypt
-----------------------------------------------------------------------------------------------------------------------------------
 \xc30d04070302019cbf869628153165d23e01dea171131a9de0a6ab3556e36295b25165e8f78615ad08d2dfaa1254ceb83202f18c9c446d3808d762612caeaefdc3c440558480f8c9d9abda62ff790d
(1 row)
```

Slika 32: Primjer rezultata SQL upita (autorski rad, 2020)

Dok se dekriptiranjem rezultata sa slike 32 pomoću upita:

```
SELECT * FROM  PGP_SYM_DECRYPT('\xc30d04070302019cbf869628153165
d23e01dea171131a9de0a6ab3556e36295b25165e8f78615ad08d2dfaa1254ce
b83202f18c9c446d3808d762612caeaefdc3c440558480f8c9d9abda62ff790d',
'aspp12asdpk2');
```

dobiva rezultat sa slike 33.

```
 pgp_sym_decrypt
-----------------
 tajni_podatak
(1 row)
```

Slika 33: Primjer rezultata SQL upita

(autorski rad, 2020)

Dakle, pomoću funkcije *PGP_SYM_ENCRYPT()*, kriptiraju se podaci o procjenama iz atributa relacije *assessments*, te osobni podaci korisnika aplikacije iz atributa relacije *pople*. Ti se podaci kriptiraju u tijekom izvođenja korisničkih funkcija *get_person_assessment_info()* i *get_person_info()*, respektivno. Bitno je još napomenuti kako obje ove funkcije kao argument primaju IB osobe za koje se dohvaćaju potrebni podaci. Ove se funkcije definiraju i u ovoj bazi podataka zbog svrhe testiranja, no njihova je stvarna svrha zaštiti podatke iz relacija BP *central_db*. Potom, nakon što se relacije napune podacima iz dataseta za svrhe testiranja, nad relacijama ove BP potrebno je provesti dodatan postupak anonimiziacije funkcijama PostgreSQL proširenja *PostgreSQL Anonymizer*. Pojedine su funkcije ukratko objašnjene u komentarima u sljedećem isječku koda.

```
-- anonimizacija za bazu s testnim podacima --

/* dodaje vremenske odmake datumskim atributima, na način da pomiče navedeni datum unutar vremenskog intervala
specificiranog u posljednjem parametru */
SELECT anon.add_noise_on_datetime_column('assessments', 'screening_date', '15 days 1 months 1 years');

SELECT anon.add_noise_on_datetime_column('people', 'date_of_birth', '23 days 8 months 2 years');


/* dodaje numeričke odmake numeričkim atributima unutar omjera specificiranog u posljednjem parametru */
SELECT anon.add_noise_on_numeric_column('assessments', 'rov_raw', 0.15);

SELECT anon.add_noise_on_numeric_column('assessments', 'ror_raw', 0.25);

SELECT anon.add_noise_on_numeric_column('assessments', 'rofa_raw', 0.15);


/* premješta vrijednosti zadanog atributa, proslijeđenog u drugom parametru; zahtjeva navođenje naziva primarnog ključa
relacije anonimiziranog atributa, u posljednjem parametru */
SELECT anon.shuffle_column('assessments', 'rov_score_txt', 'id');

SELECT anon.shuffle_column('assessments', 'ror_score_txt', 'id');

SELECT anon.shuffle_column('assessments', 'rofa_score_txt', 'id');
```

Shema relacija i struktura same BP "central_db" jednake su onima iz BP „dev_test_mock_db”, stoga ovdje neće biti detaljnije objašnjene, obzirom na to što se za njih

koriste identične SQL definicije. Razlika koja je predstavljena u sljedećem upitu, nalazi se u korisnički definiranim funkcijama *get_person_assessment_info()* i *get_person_info()* za dohvat kriptiranih podataka iz relacija *assessments* i *people*, respektivno. Slijede njihove SQL definicije.

```
/*
Korisnička funkcija get_person_assessment_info(arg_person_id).
*/
BEGIN;

CREATE OR REPLACE FUNCTION get_person_assessment_info(arg_person_id TEXT)
    RETURNS TABLE(rov_raw TEXT, rov_txt TEXT, ror_raw TEXT, ror_txt TEXT,
                  rofa_raw TEXT, rofa_txt TEXT, screening_date TEXT,
                  legal_status TEXT, custody_status TEXT, rec_super_txt TEXT,
                  assess_reason TEXT)
AS $$
BEGIN
    RETURN QUERY
        SELECT
            (PGP_SYM_ENCRYPT(a.rov_raw::TEXT, pk.symetric_key)::TEXT) AS rov_raw,

            (PGP_SYM_ENCRYPT(a.rov_score_txt::TEXT, pk.symetric_key)::TEXT) AS rov_txt,

            (PGP_SYM_ENCRYPT(a.ror_raw::TEXT, pk.symetric_key)::TEXT) AS ror_raw,

            (PGP_SYM_ENCRYPT(a.ror_score_txt::TEXT, pk.symetric_key)::TEXT) AS ror_txt,

            (PGP_SYM_ENCRYPT(a.rofa_raw::TEXT, pk.symetric_key)::TEXT) AS rofa_raw,

            (PGP_SYM_ENCRYPT(a.rofa_score_txt::TEXT, pk.symetric_key)::TEXT) AS rofa_txt,

            (PGP_SYM_ENCRYPT(a.screening_date::TEXT, pk.symetric_key)::TEXT) AS screening_date,

            (PGP_SYM_ENCRYPT(a.legal_status::TEXT, pk.symetric_key)::TEXT) AS legal_status,

            (PGP_SYM_ENCRYPT(a.custody_status::TEXT, pk.symetric_key)::TEXT) AS custody_status,

            (PGP_SYM_ENCRYPT(a.rec_supervision_level_text::TEXT, pk.symetric_key)::TEXT) AS rec_super_txt,

            (PGP_SYM_ENCRYPT(a.assessment_reason::TEXT, pk.symetric_key)::TEXT) AS assess_reason
        FROM
            assessments a
        INNER JOIN
            people_symetric_keys pk ON a.person_id=pk.person_id
        WHERE
            a.person_id::TEXT=arg_person_id;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;

REVOKE ALL ON FUNCTION get_person_assessment_info(TEXT) FROM PUBLIC;

GRANT EXECUTE ON FUNCTION get_person_assessment_info(TEXT) TO dev_production_role;

COMMIT;



/*
Korisnička funkcija get_person_info(arg_person_id).
*/
BEGIN;

CREATE OR REPLACE FUNCTION get_person_info(arg_person_id TEXT)
    RETURNS TABLE(f_name TEXT, m_name TEXT, l_name TEXT, dob TEXT, email TEXT,
                  address TEXT, postcode TEXT, sex TEXT, ethnicity TEXT,
                  language TEXT, marital_status TEXT)
AS $$
BEGIN
    RETURN QUERY
        SELECT
            (PGP_SYM_ENCRYPT(p.first_name::TEXT, pk.symetric_key)::TEXT) AS f_name,

            (PGP_SYM_ENCRYPT(p.middle_name::TEXT, pk.symetric_key)::TEXT) AS m_name,

            (PGP_SYM_ENCRYPT(p.last_name::TEXT, pk.symetric_key)::TEXT) AS l_name,

            (PGP_SYM_ENCRYPT(p.date_of_birth::TEXT, pk.symetric_key)::TEXT) AS dob,

            (PGP_SYM_ENCRYPT(p.email::TEXT, pk.symetric_key)::TEXT) AS email,

            (PGP_SYM_ENCRYPT(p.address::TEXT, pk.symetric_key)::TEXT) AS address,

```

```
78
79              (PGP_SYM_ENCRYPT(p.postcode::TEXT, pk.symetric_key)::TEXT) AS postcode,
80
81              (PGP_SYM_ENCRYPT(p.sex::TEXT, pk.symetric_key)::TEXT) AS sex,
82
83              (PGP_SYM_ENCRYPT(p.ethnicity::TEXT, pk.symetric_key)::TEXT) AS ethnicity,
84
85              (PGP_SYM_ENCRYPT(p.language::TEXT, pk.symetric_key)::TEXT) AS language,
86
87              (PGP_SYM_ENCRYPT(p.maritalStatus::TEXT, pk.symetric_key)::TEXT) AS marital_status
88          FROM
89              people p
90          INNER JOIN
91              people_symetric_keys pk ON p.person_id=pk.person_id
92          WHERE
93              p.person_id::TEXT=arg_person_id;
94  END;
95  $$ LANGUAGE plpgsql SECURITY DEFINER;
96
97  REVOKE ALL ON FUNCTION get_person_info(TEXT) FROM PUBLIC;
98
99  GRANT EXECUTE ON FUNCTION get_person_info(TEXT) TO dev_production_role;
100
101 COMMIT;
102
```

Korisnička funkcija *get_person_assessment_info()* je namijenjena ulozi dev_production_role. Osmišljena je na način da dohvaća sve potrebne atribute iz relacije *assessments*, ali kriptirane pomoću simetričnog ključa osobe, čija je vrijednost za atribut person_id iz relacije *assessments* jednaka vrijednosti argumenta arg_person_id. Na taj se način štite podaci korisnika od zlonamjernog čitanja treće strane koja je zadužena za njihovo prikazivanje na korisnikovom uređaju. Dohvaćeni podaci iz relacije assessments dekriptiraju se tek na korisnikovom uređaju, gdje je ponovno dostupan simetrični ključ korisnika. Ovakav način zaštite podataka zapravo predstavlja oblik pseduonimizacije, prodiskutiran u poglavlju 3.

Korisnička funkcija *get_person_info()* je namijenjena ulozi dev_production_role. Osmišljena je na način da dohvaća sve potrebne atribute iz relacije *people*, ali kriptirane pomoću simetričnog ključa osobe, čija je vrijednost za atribut person_id iz relacije *people* jednaka vrijednosti argumenta arg_person_id. Na taj se način štite podaci korisnika od zlonamjernog čitanja treće strane koja je zadužena za njihovo prikazivanje na korisnikovom uređaju. Dohvaćeni podaci iz relacije people dekriptiraju se tek na korisnikovom uređaju, gdje je ponovno dostupan simetrični ključ korisnika.

Potrebno je još istaknuti kako su ove funkcije definirane u transakcijskom bloku (BEGIN; ... COMMIT;). Ovakav način definiranja štiti zlouporabu funkcija kada se oznakom *SECURITY DEFINER* dopusti izvršenje funkcije s pravima vlasnika funkcije, što je u ovome slučaju administrator baze podataka. To bi u slučaju definiranja funkcije izvan transakcijskog bloka stvorilo kratak vremenski period u kojem bi ova funkcija, prema predefiniranom pravilu PostgreSQL sustava, bila dostupna svim korisnicima sheme *Public* na izvršenje [27]. Tek se izvršenjem naredbi:

```
REVOKE ALL ON FUNCTION get_person_info(TEXT) FROM PUBLIC;
GRANT EXECUTE ON FUNCTION get_person_info(TEXT) TO dev_production_role;
```

najprije opoziva pravo izvršenja ovih funkcija svim ulogama sheme *Public*, a potom se dozvoljava isključivo odabranoj ulozi, što je u potonjem primjeru uloga *dev_production_role* [5]. Ovakav je pristup ograničenju prava izvršenja samo određenim ulogama primijenjen i na ostalim funkcijama kojama je cilj zaštiti ili anonimizirati podatke.

Ostala je još funkcija *get_data_for_external_anon()*, koja je dozvoljena za izvršavanje isključivo ulozi *ml_dpo_role*. Ova funkcija dohvaća sve podatke iz relacije *assessments*, zajedno s tekstualnim oznakama iz ostalih relacija, koji su bitni za analitičke procese. Dakle, izostavlja bilo kakve osobne podatke koji su izravni identifikatori pojedinaca. Jedina iznimka ovdje jest dohvaćanje IB-a osobe, no on je maskiran znakovima "--". Ova je odluka ipak nužna kako bi se na neki način uspjelo kasnije utvrditi koliki se broj zapisa odnosi na osobe za koje već postoje zapisi, tj. za koliko osoba postoji ponovljena procjena rizika. Bitno je za reći kako rezultat ove funkcije predstavlja skup podataka koje je najprije potrebno anonimizirati, dakle u ovome bi to slučaju bili podaci izvornog dataseta.

```
/*
Korisnička funkcija get_data_for_external_anon()
*/
BEGIN;

CREATE OR REPLACE FUNCTION get_data_for_external_anon()
    RETURNS TABLE(person_id TEXT, dob DATE, sex TEXT, ethnicity TEXT,
                  language TEXT, postcode TEXT, maritalStatus TEXT,
                  agency_text TEXT, scale_set TEXT, assessment_reason TEXT,
                  legal_status TEXT, custody_status TEXT, screening_date DATE,
                  rec_supervision_level_text TEXT, rov_raw REAL, ror_raw REAL,
                  rofa_raw REAL)
AS $$
BEGIN
    RETURN QUERY

        SELECT
            anon.partial(p.person_id::TEXT, 1, '--', 2) as person_id,
            p.date_of_birth,
            p.sex::TEXT,
            p.ethnicity::TEXT,
            p.language::TEXT,
            p.postcode::TEXT,
            p.maritalStatus::TEXT,
            c.agency_text::TEXT,
            s.scale_set::TEXT,
            a.assessment_reason::TEXT,
            a.legal_status::TEXT,
            a.custody_status::TEXT,
            a.screening_date,
            a.rec_supervision_level_text::TEXT,
            a.rov_raw,
            a.ror_raw,
            a.rofa_raw
        FROM
            people p
        INNER JOIN assessments a
            ON a.person_id = p.person_id
        INNER JOIN cases c
            ON a.case_id = c.case_id
        INNER JOIN scaleSets s
            ON a.scaleSet_id = s.scaleSet_id;

END;
```

```
45  $$ LANGUAGE plpgsql SECURITY DEFINER;
46
47  REVOKE ALL ON FUNCTION get_data_for_external_anon() FROM PUBLIC;
48
49  GRANT EXECUTE ON FUNCTION get_data_for_external_anon() TO ml_dpo_role;
50
51  COMMIT;
```

Nadalje, BP "ml_warehouse" služi samo za pohranu anonimiziranih skupova podataka, koji se mogu izvoziti u svrhu javnog dijeljenja isith. U ovoj bazi kreira se sljedeća relacija:

```
 1 CREATE TABLE IF NOT EXISTS assessments_report(
 2     record_id SERIAL PRIMARY KEY,
 3     agency_text VARCHAR(30),
 4     sex VARCHAR(30) NOT NULL,
 5     ethnicity VARCHAR(30) NOT NULL,
 6     age SMALLINT NOT NULL,
 7     postcode_decile SMALLINT NOT NULL,
 8     scale_set VARCHAR(50) NOT NULL,
 9     assessment_reason VARCHAR(50),
10     language VARCHAR(25) NOT NULL,
11     legal_status VARCHAR(50) NOT NULL,
12     custody_status VARCHAR(50) NOT NULL,
13     maritalStatus VARCHAR(50) NOT NULL,
14     screening_date DATE NOT NULL,
15     rec_supervision_level_text VARCHAR(50) NOT NULL,
16     rov_raw REAL NOT NULL,
17     ror_raw REAL NOT NULL,
18     rofa_raw REAL NOT NULL,
19     first_assessment CHAR(1)
20 );
```

Dakle, očito je kako ona u potpunosti oponaša strukturu anonimiziranog dataseta. Nadalje, za ovu su BP predviđene dvije uloge: *ml_dpo_role* i *public_data_sharing_role*. Uloga *ml_dpo_role* predstavlja korisnika koji je glavni odgovoran za rukovanje osobnim podacima u organizaciji, npr. voditelja Data Privacy odijela, te kao takva može stvarati nove relacije i uređivati postojeće. Dok uloga *public_data_sharing_role* predstavlja korisnike koji, npr. preko API-ja, dohvaćaju anonimizirane skupove podataka iz, npr. tablice *assessments_report*. Za tu je ulogu i definirana funkcija koja dohvaća te podatke, s obizrom na to kako spomenuta uloga nema nikakvih dodatnih prava u ovoj BP, osim prava na izvršenje funkcije *generate_temp_public_data()*. Osmišljena je s ciljem javnog dijeljenja anonimiziranog dataseta za potrebe daljnje podatkovne obrade i analize. Ova funkcija primjenjuje dodatne metode anonimizacije, kako bi se privatnost podataka dodatno pospješila, makar uz neznačajan gubitak korisnosti dataseta. Slijedi definicija funkcije:

```
/*
Korisnička funkcija generate_temp_public_data()
*/
BEGIN;

CREATE OR REPLACE FUNCTION generate_temp_public_data()
    RETURNS TABLE(id INTEGER, age int4range, sex TEXT, ethnicity TEXT,
        language TEXT, postcode_decile SMALLINT, maritalStatus TEXT,
        scale_set TEXT, assessment_reason TEXT, legal_status TEXT,
        custody_status TEXT, rec_supervision_level_text TEXT,
        agency_text TEXT, screening_date DATE, rov_raw REAL,
        ror_raw REAL, rofa_raw SMALLINT, first_assessment CHAR)
AS $$
BEGIN
    CREATE TEMP TABLE IF NOT EXISTS temp_table AS
        SELECT
            ar.record_id AS id,
            anon.generalize_int4range(ar.age, 10) AS age,
            ar.sex::TEXT,
            ar.ethnicity::TEXT,
            ar.language::TEXT,
            ar.postcode_decile,
            ar.maritalStatus::TEXT,
            ar.scale_set::TEXT,
            ar.assessment_reason::TEXT,
            ar.legal_status::TEXT,
            ar.custody_status::TEXT,
            ar.rec_supervision_level_text::TEXT,
            anon.partial(ar.agency_text, 1, 'xxxx', 2)::TEXT as agency_text,
            ar.screening_date,
            ar.rov_raw,
            ar.ror_raw,
            ar.rofa_raw,
            ar.first_assessment
        FROM assessments_report as ar;


PERFORM anon.add_noise_on_datetime_column('temp_table', 'screening_date', '13 days 3 months 1 years');
PERFORM anon.shuffle_column('temp_table', 'first_assessment', 'id');
PERFORM anon.add_noise_on_numeric_column('temp_table', 'rov_raw', 0.05);
PERFORM anon.add_noise_on_numeric_column('temp_table', 'ror_raw', 0.05);
PERFORM anon.add_noise_on_numeric_column('temp_table', 'rofa_raw', 0.05);


RETURN QUERY
    SELECT
        ttable.id,
        ttable.age,
        ttable.sex,
        ttable.ethnicity,
        ttable.language,
        ttable.postcode_decile,
        ttable.maritalStatus,
        ttable.scale_set,
        ttable.assessment_reason,
        ttable.legal_status,
        ttable.custody_status,
        ttable.rec_supervision_level_text,
        ttable.agency_text,
        ttable.screening_date,
        ttable.rov_raw,
        ttable.ror_raw,
        ttable.rofa_raw::SMALLINT,
        ttable.first_assessment
    FROM
        temp_table as ttable;

DROP TABLE temp_table CASCADE;

END
$$ LANGUAGE plpgsql SECURITY DEFINER;

REVOKE ALL ON FUNCTION generate_temp_public_data() FROM PUBLIC;

GRANT EXECUTE ON FUNCTION generate_temp_public_data() TO public_data_sharing_role;

COMMIT;
```

Ova se funkcija sastoji od tri dijela:

1) Dohvaćaju se podaci iz relacije *assessments_report* u privremenu tablicu *temp_table*. Pritom se vrijednosti atributa *age* anonimiziraju poopćenjem u intervale od po 10 godina, dok se vrijednosti atributa *agency_text* dijelomično maskiraju nizom znakova 'xxxx'. Za to su zaslužne funkcije *anon.generalize_int4range()* i *anon.partial()*, respektivno,

2) Dohvaćeni zapisi dodatno se anonimiziraju, bilo dodavanjem 'šuma' numeričkim atributima u vrlo maloj mjeri kako bi se očuvala statistička svojstva dataseta, bilo premještanjem vrijednosti određenih artibuta. Za to je zaslužan niz naredbi u linijama koda 38 - 42 prošlog isječka.

3) Dohvaćaju se svi zapisi iz privremene *temp_table* tablice, te se na kraju privremena tablica uništava.

Potrebno je još prikazati primjere ispisa iz relacija baza podataka za definirane korisničke funkcije. Slika 34 prdstavlja primjer ispisa upita koji dohvaća informacije o osobi iz relacije „people”. Upit je izvršen ulogom administratora, prema primjeru upita iz korisničkih funkcija *get_person_info()* i *get_person_assessment_info(),* definiranima nad bazom „central_db”. Podaci su dohvaćeni za osobu s vrijednošću atributa „person_id” jednakom *3890*. Slika 35 pak prikazuje ispis koji se dobiva pozivom upita koji izvršava funkciju *get_person_info()*, koja je namijenjana ulozi „dev_production_role”. Vidljivo je da su svi podaci koje dohvaća uloga „dev_production_role” za neku osobu kriptirani, a prema ranije objašnjenoj definiciji, može se reći da su pseudonimizirani. Stoga, s aspekta anonimizacije ti podaci još uvijek predstavljaju osobne neanonimizirane podatke, no barem su na ovaj način zaštićeni od očiju nepovjerljivog vršitelja obrade.

	0
f_name	GREGORY
m_name	NaN
l_name	WILLIAMS
dob	2063-03-30
email	g.williams805@gmail.com
address	672 Jenkins Fields
postcode	55839
sex	Male
ethnicity	African-American
language	English
marital_status	Married

Slika 34: Dohvaćeni podaci osobe iz relacije people (autorski rad, 2020)

	0
f_name	\xc30d040703020ab3ede7db29573e60d23801076116cb...
m_name	\xc30d040703020ed94ff132a7371e6cd234017299cd21...
l_name	\xc30d04070302f70e42a106b636687ed2390168a013ce...
dob	\xc30d040703029abfcf1c18e4dfad64d23b014e99555c...
email	\xc30d0407030225a34c2028bd1ab666d24801536a7b27...
address	\xc30d04070302f5a98036acafee4c71d24301b24ee834...
postcode	\xc30d04070302f845fe56ce8a020664d23601d742fd8d...
sex	\xc30d04070302fce5ce4f54ee200f6ad23501fe07fc9a...
ethnicity	\xc30d0407030240505a0b208fc1be7bd2410193aec0ea...
language	\xc30d0407030222b06fb021073e1769d23801d9b98c35...
marital_status	\xc30d040703024a48fa80edcdcdf575d23801105b691c...

Slika 35: Dohvaćeni kriptirani osobni podaci iz relacije people (autorski rad, 2020)

Na slici 36 vidljiv je ispis dobiven izvršavanjem upita koji pokreće funkciju *get_data_for_external_anon()*, koja je namijenjena ulozi „ml_dpo_role". Ovakva struktura skupa podataka vrlo je slična izvornom skupu, s jedinom razlikom što su u ovom primjeru ipak anonimizirani podaci za atributa "person_id", funkcijom *anon.partial()* PostgreSQL dodatka *PostgreSQL Anonymizer*, koja određni niz znakova nekog stringa mijenja kombinacijom nekih drugih znakova. U ovom je slučaju to kombinacija "--" (dvije crtice). Ovo je tzv. metoda anonimizacije podataka maskiranjem, kojom se podaci prikazuju u izobličenom obliku.

	0	1	2
person_id	8--66	1--02	1--33
dob	1980-01-05	1979-01-07	1986-04-17
sex	Female	Male	Male
ethnicity	African-American	African-American	African-American
language	English	English	English
postcode	20129	37594	96288
maritalstatus	Single	Single	Single
agency_text	PRETRIAL	PRETRIAL	PRETRIAL
scale_set	Risk and Prescreen	Risk and Prescreen	Risk and Prescreen
assessment_reason	Intake	Intake	Intake
legal_status	Pretrial	Pretrial	Pretrial
custody_status	Jail Inmate	Jail Inmate	Jail Inmate
screening_date	2013-11-24	2013-03-24	2014-10-02
rec_supervision_level_text	Low	Low	Medium
rov_raw	-2.86	-1.95	-1.57
ror_raw	-1.46	-0.8	-0.1
rofa_raw	17	24	18

Slika 36: Izgled dataseta namijenjenog anonimizaciji (autorski rad, 2020)

Primjer anonimiziranog dataseta metodom očuvanja statističkih karakteristika izvornog dataseta već je prikazan, stoga je još ostalo za prikazati primjer ispisa podataka za potrebe javnog dijeljenja. Ti se podaci dohvaćaju korisničkom funkcijom *generate_temp_public_data()*, koja je namijenjena ulozi „public_data_sharing_role". Slika 37 prikazuje strukturu tog skupa podataka.

	0	1	2
id	678	1153	3001
age	[40, 50)	[40, 50)	[60, 70)
sex	Male	Male	Female
ethnicity	Caucasian	Hispanic	Caucasian
language	English	English	English
postcode_decile	2	5	4
maritalstatus	Single	Married	Unknown
scale_set	Risk and Prescreen	All Scales	All Scales
assessment_reason	Intake	Intake	Intake
legal_status	Other	Other	Pretrial
custody_status	Probation	Probation	Jail Inmate
rec_supervision_level_text	Low	Low	Low
agency_text	Pxxxxon	PxxxxAL	PxxxxAL
screening_date	2015-07-19	2013-09-05	2015-05-05
rov_raw	-3.98678	-3.00646	-3.71969
ror_raw	-2.06007	-1.50021	-2.24937
rofa_raw	14	18	14
first_assessment	1	1	1

Slika 37: Izgled dataseta anonimiziranog za potrebe javnog dijeljenja (autorski rad, 2020)

U opisu funkcije za dohvaćanje podataka namijenjene ulozi "public_data_sharing_role", spomenute su i funkcije PostgreSQL dodatka PostgreSQL Anonymizer kojima se dodaje "šum" u vrijednosti atributa privremene tablice – *anon.add_noise_on_datetime_column() i add_noise_on_numeric_column()*; te funkcija kojom se premještaju vrijednosti atributa - *anon.shuffle_column()*. „Šum" je u ovome primjeru dodan u temporalni atribut "screening_date" te numeričke atribute rizika, dok se premještanje vrijednosti izvršilo nad atributom "first_assessment". Također je vidljiv i postupak anonimizacije vrijendosti atributa "agency_text", već ranije opisanom metodom maskiranja podataka. Posebno je bitno istaknuti primjer poopćivanja vrijednosti atributa "age", gdje su godine grupirane u intervale od po 10 godina. Specifičan primjer poopćivanja predstavljen je u [1] i [2], koji donosi dodatno ograničenje da se u svakoj poopćenoj skupini u skupu podataka mora nalaziti

najmanje *k* članova s istim vrijednostima. U ovom bi to slučaju značilo da za svaki interval od 10 godina mora postojati *k* osoba koje dijele godine iz tog intervala. U ovom je primjeru za potrebe poopćivanja vrijednosti atributa "age" iskorištena funkcija *generalize_int4range()* dodatka *PostgreSQL Anonymizer*. Sličan primjer poopćivanja prikazan je ranije nad vrijednostima atributa „Postcode" izvornog dataseta.

Na kraju je još potrebno komentirati mogućnosti proširenja *PostgreSQL Anonymizer*. Kao što je demonstrirano, ovo proširenje nudi za sada skroman skup funkcionalnosti anonimizacije koje se mogu upotrijebiti izravno nad relacijama baza podataka. Dodatno, a što ovdje nije demonstrirano na taj način, ovo proširenje omogućuje diferenciranje pravila anonimizacije izravno nad tablicama za sve uloge koje se proglase maskirajućima, tj. ulogama za koje vrijede ta anonimizacijska pravila. To može predstavljati dodatan sloj specificiranja organizacijskih politika i prava pristupa podacima koje se ostvaruju u bazama podataka. No, zbog veće je fleksibilnosti ovdje odabran sličan pristup, uz primjenu istih funkcionalnosti kroz korisnički definirane funkcije.

Cilj je ovog dijela anonimizacije, koja je primijenjena izravno u bazama podataka, bio posebno istaknuti formiranje ograničenja na prava pristupa podacima kroz različite uloge. Ovaj je segment anonimizacije i zaštite podataka također poduprt i od strane GDPR-a, a značajan je zbog osiguranja što veće transparentnosti prilikom pristupanja podacima. Dodatno, a što ovdje nije pokazano, svaki se pristup podacima može dokumentirati u dnevnike sustava kako bi se uvijek znalo tko je i kada koristio podatke. No, kako bi proces definiranja prava prisutpa zaista bio koristan, **od neizmjerne je važnosti na početku odrediti koje su stvarne i dovoljne potrebe pojedine uloge koja pristupa podacima.** Već taj korak može uvelike pospješiti anonimizaciju i očuvanje privatnosti podataka jer sprječava nepotrebno kolanje osjetljivih i osobnih podataka sustavom ili pak između organizacije i neke treće strane. **Jednom, kada su potrebe svih uključenih aktora dobro opisane, podaci se mogu ispravno zaštiti i anonimizirati te javno dijeliti.**

Za kraj, potrebno je još jednom napomenuti kako će rizik od curenja podataka ili prepoznavanja pojedinaca uvijek postojati, no sve dokle se taj rizik nastoji minimizirati u što većoj mjeri, a uz očuvanje korisnosti podataka, proces anonimizacije ne prestaje i zahtjevati će prilagodbe kroz vrijeme.

8. Zaključak

U ovome je radu predstavljen koncept anonimizacije podataka – proces nepovratne promjene podataka nekog skupa podataka s ciljem minimizacije rizika od utvrđivanja identiteta pojedinca na kojeg se odnosi neki zapis iz anonimiziranog skupa. Kao primarna tehnika anonimizacije, korištena je metoda generiranja sintetičkih podataka s očuvanjem statističkih karakteristika izvornog skupa podataka. U te je svrhe korištena knjižnica programskog jezika *Python*, naziva *DataSynthesizer*. Spomenuta knjižnica pruža metode generiranja takvih podataka i njihovu analizu kroz 3 koraka: **1)** analiza statističkih karakteristika dataseta, tj. izrada modela korelacije atributa dataseta te vjerojatnosnih distribucija pojedinih atributa **2)** generiranje sintetičkih podataka prema utvrđenim korelacijama i vjerojatnosnim distribucijama **3)** grafička analiza dobivenih rezultata. Spomenuti je proces izvršen nad izvornim datasetom zbog daljnjih potreba statističke obrade i analize, ali i ono bitnije – zbog daljnjeg dijeljenja anonimiziranog skupa podataka. Kroz grafičku je analizu utvrđeno kako anonimizirani skup u dobroj mjeri održava statistička svojstva izvornog dataseta, čija su odstupanja primarno posljedica korištenja diferencijalne privatnosti, tj. generiranja podataka s namjernim određenim odstupanjem od vjerojatnosnih distribucija pojedinih atributa. Također je predstvaljen i proces anonimizacije za svrhe izrade, testiranja i održavanja aplikacijskog rješenja za prikaz podataka izvornog skupa. Taj je proces uključivao nasumično generiranje podataka koji su trebali svojim rasponima vrijednosti i tipom podataka predstavljati podatke izvornog skupa, tako doprinoseći stvaranju realnijeg okruženja za izradu aplikacije.

Svi su podaci na kraju pohranjeni u baze podataka, koje su pak implementriane na način da pojedinim ulogama, tj. izvršiteljima obrade koji pristupaju podacima ograniče pristup podacima, na način da im se pruže samo oni zaštićeni ili anonimizirani podaci koji ispunjuju svrhe obrade. Tijekom pristupanja tim podacima od strane različitih uloga, oni se dodatno anonimiziraju korištenjem funkcionalnosti pruženih kroz programski dodatak *PostgreSQL* SUBP-u zvanom *PostgreSQL Anonymizer*.

Opisani proces anonimizacije donosi iscrpnu analizu izvornog skupa podataka, njegovih atributa i zapisa, koja je odrađena s velikom lakoćom uz pomoć *Pythonove* knjižnice *pandas*. Utvrđeno je kako odabrani izvorni skup podataka obiluje raznim osobnim podacima, odnosno raznim izravnim i neizravnim identifikatorima, što je pružilo brojne mogućnosti za demonstracije različitih tehnika anonimizacije.

9. Literatura

[1] L. Sweeney, ‘k-Anonymity: a Model for Protecting Privacy’, *Int. J. Unc. Fuzz. Knowl. Based Syst.*, vol. 10, no. 05, pp. 557–570, Oct. 2002, doi: 10.1142/S0218488502001648.

[2] P. Samarati and L. Sweeney, ‘Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression’, *Proceedings of the IEEE Symposium on Research in Security and Privacy (S&P)*, p. 19, May 1998.

[3] S. J. Nass, L. A. Levit, L. O. Gostin, and I. of M. (US) C. on H. R. and the P. of H. I. T. H. P. Rule, *The Value, Importance, and Oversight of Health Research*. National Academies Press (US), 2009.

[4] T. Coughlin, ‘175 Zettabytes By 2025’, *Forbes*, 2018. https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/ (accessed Sep. 06, 2020).

[5] K. Rabuzin, *SQL - napredne teme*. Varaždin: Fakultet organizacije i informatike, Sveučilište u Zagrebu, 2014.

[6] EU parlament and Vijeće EU, ‘Uredba (EU) 2016/679 Europskog parlamenta i Vijeća od 27. travnja 2016. o zaštiti pojedinaca u vezi s obradom osobnih podataka i o slobodnom kretanju takvih podataka te o stavljanju izvan snage Direktive 95/46/EZ (Opća uredba o zaštiti podataka)’. 2018, Accessed: Sep. 02, 2020. [Online]. Available: https://eur-lex.europa.eu/legal-content/HR/TXT/PDF/?uri=CELEX:32016R0679&from=HR.

[7] K. El Emam and L. Arbuckle, *Anonymizing Health Data*, 1st ed. O’Reilly Media, Inc., 2013.

[8] L. Arbuckle and K. El Emam, *Building an Anonymization Pipeline: Creating Safe Data*, 1st ed. O’Reilly Media, Inc., 2020.

[9] G. M. Harari, N. D. Lane, R. Wang, B. S. Crosier, A. T. Campbell, and S. D. Gosling, ‘Using Smartphones to Collect Behavioral Data in Psychological Science: Opportunities, Practical Considerations, and Challenges’, *Perspect Psychol Sci*, vol. 11, no. 6, pp. 838–854, Nov. 2016, doi: 10.1177/1745691616650285.

[10] Laboratorij za sustave i signale Zavoda za elektroničke sustave i, ‘Anonimizacija i pseudonimizacija podataka’. CERT, 2018, Accessed: Jun. 25, 2020. [Online]. Available: https://www.cert.hr/wp-content/uploads/2018/08/anonimizacija_i_pseudonimizacija_podataka.pdf.

[11] International Organization for Standardization (ISO), ‘ISO 25237:2017, Health informatics — Pseudonymization’. pub-ISO, 2017, Accessed: Sep. 03, 2020. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:25237:ed-1:v1:en:fn:1.

[12] O. for C. Rights (OCR), ‘Methods for De-identification of PHI’, *HHS.gov*, Sep. 07, 2012. https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html (accessed Sep. 07, 2020).

[13] J. Angwin, J. Larson, S. Mattu, L. Kirchner, and ProPublica, ‘Machine Bias’, *ProPublica*, May 23, 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing?token=TuMy8gExpvZxdxiWRs7mTz21zSyVml5E (accessed Sep. 10, 2020).

[14] M. Rouse, ‘What is Machine Learning Bias (AI Bias)?’, *SearchEnterpriseAI*, 2018. https://searchenterpriseai.techtarget.com/definition/machine-learning-bias-algorithm-bias-or-AI-bias (accessed Sep. 10, 2020).

[15] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, ‘How We Analyzed the COMPAS Recidivism Algorithm’, *ProPublica*, May 23, 2016. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm?token=7enZ92BgsMqAFD5AxOxy-lfaE6sqoIHh (accessed Sep. 10, 2020).

[16] ‘Declaration on Ethics and Data Protection in Artificial Intelligence’, Brussels, Belgium, Aug. 2020, Accessed: Sep. 10, 2020. [Online]. Available: https://edps.europa.eu/sites/edp/files/publication/icdppc-40th_ai-declaration_adopted_en_0.pdf.

[17] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, *Data and analysis for ‘Machine Bias’*. ProPublica, 2017.

[18] D. Faraglia, *Faker*. 2020.

[19] H. Ping, S. Russell, J. Stoyanovich, and A. Thevapalan, *DataResponsibly/DataSynthesizer*. Data, Responsibly, 2020.

[20] H. Ping, J. Stoyanovich, and B. Howe, ‘DataSynthesizer: Privacy-Preserving Synthetic Datasets’, in *Proceedings of SSDBM ’17*, Chicago, IL, USA, Jul. 2017, p. 5, doi: http://dx.doi.org/10.1145/3085504.3091117.

[21] The Open Data Institute, *A hands-on tutorial showing how to use Python to do anonymisation with synthetic data*. The Open Data Institute, 2020.

[22] N. Adžaga, A. M. Špoljarić, and N. Sandrić, ‘Vjerojatnost i statistika’, *Građevinski fakultet, Sveučilište u Zagrebu*, p. 145, 2017.

[23] ‘Unix time’, *Wikipedia*. Aug. 31, 2020, Accessed: Sep. 10, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Unix_time&oldid=976000718.

[24] D. Clochard, *PostgreSQL Anonymizer*. Dalibo Labs, 2020.

[25] K. Rabuzin, *Uvod u SQL*. Varaždin: Fakultet organizacije i informatike, Sveučilište u Zagrebu, 2011.

[26] ‘PostgreSQL: Documentation: pgcrypto’. https://www.postgresql.org/docs/current/pgcrypto.html (accessed Sep. 10, 2020).

[27] ‘PostgreSQL: Documentation: CREATE FUNCTION’. https://www.postgresql.org/docs/current/sql-createfunction.html (accessed Sep. 10, 2020).

Popis slika

Slika 1: Izgled izvornog dataseta.....23

Slika 2: Izgled izvornog dataseta nakon početnih transformacija (autorski rad, 2020).....27

Slika 3: Broj jedinstvenih vrijednosti pojedinog atributa dataseta (autorski rad, 2020).....32

Slika 4: Specifikacija tipova podataka dataseta (autorski rad, 2020).....32

Slika 5: Preslikavanja numeričkih raspona i tekstualnih oznaka za rizik od recidivizma (autorski rad, 2020).....34

Slika 6: Preslikavanja numeričkih raspona i tekstualnih oznaka za rizik od pojavljivanja na sudu (autorski rad, 2020).....34

Slika 7: Preslikavanja numeričkih raspona i tekstualnih oznaka za rizik od nasilja (autorski rad, 2020).....34

Slika 8: Spajanje identifikatora i tekstualnih naziva atributa ScaleSet (autorski rad, 2020)....36

Slika 9: Prikaz decilnih skupina poštanskih brojeva (autorski rad, 2020).....38

Slika 10: Konačan izgled dataseta nakon početnih koraka anonimizacije (autorski rad, 2020) 40

Slika 11: Izgled anonimiziranog dataseta, dobivenog generiranjem sintetičkih podataka (autorski rad, 2020).....45

Slika 12: Usporedba histograma atributa "Agency_Text" (autorski rad, 2020).....47

Slika 13: Usporedba histograma atributa "Sex_Code" (autorski rad, 2020).....47

Slika 14: Usporedba histograma atributa "Ethnica_Code_Text" (autorski rad, 2020).....48

Slika 15: Usporedba histograma atributa "Age" (autorski rad, 2020).....48

Slika 16: Usporedba histograma atributa "PostcodeDecile" (autorski rad, 2020).....49

Slika 17: Usporedba histograma atributa "ScaleSet" (autorski rad, 2020).....50

Slika 18: Usporedba histograma atributa "Language" (autorski rad, 2020).....50

Slika 19: Usporedba histograma atributa "LegalStatus".....51

Slika 20: Usporedba histograma atributa "CustodyStatus".....51

Slika 21: Usporedba histograma atributa "MaritalStatus".....52

Slika 22: Usporedba histograma atributa "RecSupervisionLevelText" (autorski rad, 2020)…52

Slika 23: Usporedba histograma atributa "rov_raw".....53

Slika 24: Usporedba histograma atributa "ror_raw".....53

Slika 25: Usporedba histograma atributa "rofa_raw".....54

Slika 26: Usporedba histograma atributa "FirstAssessment" (autorski rad, 2020).....54

Slika 27: Usporedba toplinskih mapa korelacije atributa izvornog (lijevo) i anonimiziranog (desno) skupa podataka (autorski rad, 2020).....55

Slika 28: Frekvencije pojava anonimiziranih vrijednosti atributa "Ethnic_Code_Text" (autorski rad, 2020).....56

Slika 29: Frekvencije pojava anonimiziranih vrijednosti atributa "*LegalStatus*" (autorski rad, 2020).....57

Slika 30: Frekvencije pojava anonimiziranih vrijednosti atributa "CustodyStatus" (autorski rad, 2020).....57

Slika 31: Izgled anonimiziranog dataseta generiranjem nasumičnih vrijednosti.....65

Slika 32: Primjer rezultata SQL upita (autorski rad, 2020).....69

Slika 33: Primjer rezultata SQL upita (autorski rad, 2020).....70

Slika 34: Dohvaćeni podaci osobe iz relacije people (autorski rad, 2020).....76

Slika 35: Dohvaćeni kriptirani osobni podaci iz relacije people (autorski rad, 2020).....77

Slika 36: Izgled dataseta namijenjenog anonimizaciji (autorski rad, 2020).....77

Slika 37: Izgled dataseta anonimiziranog za potrebe javnog dijeljenja (autorski rad, 2020). .78

Popis tablica

Tablica 1: Prikaz strukturiranog skupa podataka.....6

Tablica 2: Primjer pseudonimiziranog skupa podataka.....10

Tablica 3: Primjer šifrarnika korištenog pri pseudonimizaciji podataka.....10

Prilozi

Prilog 1 – Obrada atributa rizika

```
# PRILOG 1 - Obrada atributa rizika
# 1) --- OBRADA ATRIBUTA RIZIKA ---
# 1.1) Određivanje raspona procjenjenih vrijednosti rizika
#-------------------------------------------------------
# rizik od nasilja
rov_minimum_value = df_anon_ml.rov_raw.min()
rov_maximum_value = df_anon_ml.rov_raw.max()
rov_minimum_value, rov_maximum_value #(-4.79, 1.52)

# rizik od recidivizma
ror_minimum_value = df_anon_ml.ror_raw.min()
ror_maximum_value = df_anon_ml.ror_raw.max()
ror_minimum_value, ror_maximum_value #(-3.21, 2.36)

# rizik od nepojavljivanja na suđenju
rofa_minimum_value = df_anon_ml.rofa_raw.min()
rofa_maximum_value = df_anon_ml.rofa_raw.max()
rofa_minimum_value, rofa_maximum_value #(11.0, 51.0)

# 1.3) Određivanje decila procjenjenih vrijednosti rizika
#-------------------------------------------------------
# rizik od nasilja
rov_deciles = set(df_anon_ml.rov_decile.unique())

# rizik od recidivizma
ror_deciles = set(df_anon_ml.ror_decile.unique())

# rizik od nepojavljivanja na suđenju
rofa_deciles = set(df_anon_ml.rofa_decile.unique())

# 1.3) Određivanje pripadnih tekstualnih opisnih oznaka rizika
#-------------------------------------------------------------
# rizik od nasilja
rov_txt_label = set(df_anon_ml.rov_score_txt.unique())
rov_txt_label #{'High', 'Low', 'Medium', nan}

# rizik od recidivizma
ror_txt_label = set(df_anon_ml.ror_score_txt.unique())
ror_txt_label #{'High', 'Low', 'Medium', nan}

# rizik od nepojavljivanja na suđenju
rofa_txt_label = set(df_anon_ml.rofa_score_txt.unique())
rofa_txt_label #{'High', 'Low', 'Medium'}


# 1.4) Odstranjivanje decilnih i opisnih oznaka atributa rizika - KORAK ANONIMIZACIJE
# ------------------------------------------------------------------------------------
# specificiranje atributa rizika koji trebaju biti uklonjeni
risk_decile_attribs = [
    "rov_decile", "ror_decile", "rofa_decile"
]

risk_txt_label_attribs = [
    "rov_score_txt", "ror_score_txt", "rofa_score_txt"
]
```

```
56
57 # odbacivanje nepotrebnih atributa rizika
58 df_anon_ml.drop(columns = [*risk_decile_attribs, *risk_txt_label_attribs], inplace = True)
```

Prilog 2 – Obrada ostalih atributa

```
 1 # PRILOG 2 - Obrada ostalih atributa
 2 # 2) --- OBRADA OSTALIH ATRIBUTA ---
 3 # 2.1) Agency_Text
 4 agency_text_unq_vals_set = set(df_anon_ml.Agency_Text.unique())
 5 agency_text_unq_vals_set # {'Broward County', 'DRRD', 'PRETRIAL', 'Probation'}
 6
 7 # 2.2) Postcode
 8 #--------------
 9 postcode_vals = set(df_anon_ml.Postcode.unique())
10 list(postcode_vals)[:5] # [32773, 65543, 32778, 65552, 98324]
11
12 # 2.3) Sex_Code_Text
13 #-------------------
14 sex_code_vals = set(df_anon_ml.Sex_Code_Text.unique())
15 list(sex_code_vals) # ['Male', 'Female']
16
17 # 2.4) Ethnic_Code_Text
18 #----------------------
19 ethnic_code_vals = set(df_anon_ml.Ethnic_Code_Text.unique())
20 list(ethnic_code_vals)
21 # ['Caucasian', 'Other', 'Oriental', 'Native American',
22 # 'African-American', African-Am', Arabic', Asian', Hispanic']
23
24 # zamjena sličnih vrijednosti
25 df_anon_ml.loc[df_anon_ml.Ethnic_Code_Text   ==   "African-Am",   "Ethnic_Code_Text"]   =
26 "African-American"
27
28 # 2.5) ScaleSet_ID, ScaleSet
29 #---------------------------
30 # izdvajanje atributa "ScaleSet_ID" i "ScaleSet", te odbacivanje duplikata
31 df_anon_ml[["ScaleSet_ID", "ScaleSet"]].drop_duplicates().reset_index(drop = True)
32
33 # izbacivanje atributa "ScaleSet_ID"
34 df_anon_ml.drop(columns = ["ScaleSet_ID"], inplace = True)
35
36 # 2.6) AssessmentReason
37 #----------------------
38 assessment_reason_vals = set(df_anon_ml.AssessmentReason.unique())
39 assessment_reason_vals # {'Intake'}
40
41 # 2.7) Language
42 #--------------
43 language_vals = set(df_anon_ml.Language.unique())
44 language_vals # {'English', 'Spanish'}
45
46 # 2.8) LegalStatus
47 #-----------------
48 legal_status_vals = set(df_anon_ml.LegalStatus.unique())
49 legal_status_vals
50 # {'Conditional Release', 'Deferred Sentencing', 'Other',
51 # 'Parole Violator', Post Sentence', Pretrial', 'Probation Violator'}
52
53 # 2.9) CustdoyStatus
54 #-------------------
```

```
custody_status_vals = set(df_anon_ml.CustodyStatus.unique())
custody_status_vals
# {'Jail Inmate', 'Parole', 'Pretrial Defendant',
# 'Prison Inmate', 'Probation', 'Residential Program'}

# 2.10) RecSupervisionLevel, RecSupervisionLevelText
#-----------------------------------------------------
# izdvajanje atributa "RecSupervisionLevel" i "RecSupervisionLevelText", te odbacivanje
duplikata
df_anon_ml[["RecSupervisionLevel",
"RecSupervisionLevelText"]].drop_duplicates().reset_index(drop = True)

# uklanjanje atributa RecSupervisionLevel
df_anon_ml.drop(columns = ["RecSupervisionLevel"], inplace = True)

# 2.11) MaritalStatus
#--------------------
marital_status_vals = set(df_anon_ml.MaritalStatus.unique())
marital_status_vals
# {'Divorced', 'Married', 'Separated', 'Significant Other',
# 'Single', 'Unknown', 'Widowed'}

# 2.12) FirstAssesssment
#-----------------------
first_assess_vals = set(df_anon_ml.FirstAssessment.unique())
first_assess_vals # {0, 1}

# 2.13) DateOfBirth
#------------------
dob_min, dob_max = df_anon_ml.DateOfBirth.min(), df_anon_ml.DateOfBirth.max()
dob_min, dob_max # ('01/01/47', '12/31/95')

# 2.14) Screening_Date
#---------------------
screen_date_min, screen_date_max =
df_anon_ml.Screening_Date.min(), df_anon_ml.Screening_Date.max()

screen_date_min, screen_date_max # ('1/1/13 0:00', '9/9/14 0:00')
```

Prilog 3 – Izdvajanje i preslikavanje numeričkih atributa rizika realnog ili cjelobrojnog raspona vrijednosti s pripadnim tekstualnim oznakama

```
    # PRILOG 3 - Izdvajanje i preslikavanje numeričkih atributa rizika
    # realnog ili cjelobrojnog raspona vrijednosti s pripadnim tekstualnim oznakama

    # 1) specificiranje atributa koji se odnose na pojedinu vrstu rizika
    #-------------------------------------------------------------------
    # risk of violence
    rov_assessment_cols = [
        'rov_raw', 'rov_decile', 'rov_score_txt'
    ]

    # risk of recividism
    ror_assessment_cols = [
        'ror_raw', 'ror_decile', 'ror_score_txt'
    ]

```

```

    # risk of failure to appear
    rofa_assessment_cols = [
        'rofa_raw', 'rofa_decile', 'rofa_score_txt'
    ]

    # 2) preslikavanje numeričkih na tekstualne oznake
    #-------------------------------------------------
    # risk of violence
    df_rov = df_new_compas[[*rov_assessment_cols]]

    rov_description = []
    for rov_decile in rov_deciles:
        decil_min = df_rov[df_rov["rov_decile"] == rov_decile].min()[0]
        decil_max = df_rov[df_rov["rov_decile"] == rov_decile].max()[0]
        decil_txt_label = df_rov[df_rov["rov_decile"] == rov_decile].min()[2]
        rov_description.append({"decile": rov_decile,
        "raw_min": decil_min, "raw_max": decil_max, "txt_label": decil_txt_label})

    # risk of recividism
    df_ror = df_new_compas[[*ror_assessment_cols]]

    ror_description = []
    for ror_decile in ror_deciles:
        decil_min = df_ror[df_ror["ror_decile"] == ror_decile].min()[0]
        decil_max = df_ror[df_ror["ror_decile"] == ror_decile].max()[0]
        decil_txt_label = df_ror[df_ror["ror_decile"] == ror_decile].min()[2]
        ror_description.append({"decile": ror_decile,
        "raw_min": decil_min, "raw_max": decil_max, "txt_label": decil_txt_label})

    # risk of failure to appear
    df_rofa = df_new_compas[[*rofa_assessment_cols]]

    rofa_description = []
    for rofa_decile in rofa_deciles:
        decil_min = df_rofa[df_rofa["rofa_decile"] == rofa_decile].min()[0]
        decil_max = df_rofa[df_rofa["rofa_decile"] == rofa_decile].max()[0]
        decil_txt_label = df_rofa[df_rofa["rofa_decile"] == rofa_decile].min()[2]
        rofa_description.append({"decile": rofa_decile,
        "raw_min": decil_min, "raw_max": decil_max, "txt_label": decil_txt_label})

    # spremanje podataka o preslikavanjima
    df_rov_description = pd.DataFrame(rov_description)
    df_rov_description.to_csv("./data/my_datasets/rov_description.csv", index = False)

    df_ror_description = pd.DataFrame(ror_description)
    df_ror_description.to_csv("./data/my_datasets/ror_description.csv", index = False)

    df_rofa_description = pd.DataFrame(rofa_description)
    df_rofa_description.to_csv("./data/my_datasets/rofa_description.csv", index = False)
```

Prilog 4 – Problem malog broja zapisa

```
# PRILOG 4 - Problem malog broja zapisa
# 1) Učitavanje anonimiziranog dataseta
#--------------------------------------
df_synth_anon_correlated =
pd.read_csv("./data/anon_datasets/dsynth_correlated/synthetic_compas_dataset_correlated.csv")

# 2) Otklanjanje problema malog broja zapisa - KORAK ANONIMIZACIJE
#-----------------------------------------------------------------
# 2.1) Agency_Text
df_synth_anon_correlated.Agency_Text.value_counts()

# 2.2) Sex_Code_Text
df_synth_anon_correlated.Sex_Code_Text.value_counts()

# 2.3) Ethnic_Code_Text
df_synth_anon_correlated.Ethnic_Code_Text.value_counts()

# 2.3.1) Saniranje malog broja zapisa s vrijednostima "Arabic" i "Oriental" - KORAK ANONIMIZACIJE
df_synth_anon_correlated.loc[df_synth_anon_correlated.Ethnic_Code_Text == "Arabic", "Ethnic_Code_Text"] = "Other"
df_synth_anon_correlated.loc[df_synth_anon_correlated.Ethnic_Code_Text == "Oriental", "Ethnic_Code_Text"] = "Other"
df_synth_anon_correlated.Ethnic_Code_Text.value_counts()

```

```
# 2.4) PostcodeDecile
df_synth_anon_correlated.PostcodeDecile.value_counts()

# 2.5) ScaleSet
df_synth_anon_correlated.ScaleSet.value_counts()


# 2.6) AssessmentReason
df_synth_anon_correlated.AssessmentReason.value_counts()

# 2.7) Language
df_synth_anon_correlated.Language.value_counts()

# 2.8) LegalStatus
df_synth_anon_correlated.LegalStatus.value_counts()

# 2.8.1) Uklanjanje malog broja zapisa s vrijednostima "Deferred Sentencing"
df_synth_anon_correlated.drop(df_synth_anon_correlated.loc[df_synth_anon_correlated["LegalStatus"] ==
"Deferred Sentencing"].index, inplace=True)

df_synth_anon_correlated.LegalStatus.value_counts()

# 2.9) CustodyStatus
df_synth_anon_correlated.CustodyStatus.value_counts()

# 2.9.1) Uklanjanje malog broja zapisa s vrijednostima "Prison Inmate" i "Parole" - KORAK ANONIMIZACIJE
df_synth_anon_correlated.drop(df_synth_anon_correlated.loc[df_synth_anon_correlated["CustodyStatus"] ==
"Prison Inmate"].index, inplace=True)

df_synth_anon_correlated.drop(
df_synth_anon_correlated.loc[df_synth_anon_correlated["CustodyStatus"] == "Parole"].index,
inplace=True)

df_synth_anon_correlated.CustodyStatus.value_counts()

# 2.10) MaritalStatus
df_synth_anon_correlated.MaritalStatus.value_counts()

# 2.11) RecSupervisionLevelText
df_synth_anon_correlated.RecSupervisionLevelText.value_counts()

# 2.12) FirstAssessment
df_synth_anon_correlated.FirstAssessment.value_counts()
```

Prilog 5 – SQL definicije baza podataka, relacija i uloga

```
-- PRILOG 5 - SQL definicije baza podataka, relacija i uloga --
-- *** KREIRANJE BAZA PODATAKA *** --
CREATE DATABASE dev_test_mock_db;
CREATE DATABASE central_db;
CREATE DATABASE ml_warehouse;


-- *** KREIRANJE ULOGA S MOGUĆNOŠĆU PRIJAVE U SUSTAV *** --
CREATE ROLE dev_tester_role WITH LOGIN;
CREATE ROLE dev_production_role WITH LOGIN;
CREATE ROLE ml_dpo_role WITH LOGIN;
CREATE ROLE public_data_sharing_role WITH LOGIN;


-- *** PRIDJELJIVANJE PRAVA PRISTUPA BP POJEDINIM ULOGAMA *** --
GRANT CONNECT ON DATABASE dev_test_mock_db TO dev_tester_role;
GRANT CONNECT ON DATABASE central_db TO dev_production_role;
GRANT CONNECT ON DATABASE central_db TO ml_dpo_role;
GRANT CONNECT ON DATABASE ml_warehouse TO ml_dpo_role;
GRANT CONNECT ON DATABASE ml_warehouse TO public_data_sharing_role;


```

```
-- *** KREIRANJE RELACIJA BP dev_test_mock_db i central_db *** --
CREATE TABLE IF NOT EXISTS cases (
    case_id SERIAL PRIMARY KEY,
    agency_text VARCHAR(30)
);

CREATE TABLE IF NOT EXISTS people (
    Person_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    middle_name VARCHAR(50),
    last_name VARCHAR(50) NOT NULL,
    date_of_birth DATE NOT NULL,
    email VARCHAR(50),
    address VARCHAR(100) NOT NULL,
    postcode VARCHAR(15) NOT NULL,
    sex VARCHAR(30) NOT NULL,
    ethnicity VARCHAR(30) NOT NULL,
    language VARCHAR(25) NOT NULL,
    maritalStatus VARCHAR(50) NOT NULL
);

/*
Takozvani simetrični ključ izrađen je primjenom hash funkcije
sha256, čiji je rezultat duljine 64 znaka.
*/
CREATE TABLE IF NOT EXISTS people_symetric_keys (
    person_id INT NOT NULL,
    symetric_key CHAR(64),

    CONSTRAINT pk_person_id FOREIGN KEY(person_id)
    REFERENCES people(person_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS scaleSets (
    scaleSet_id SERIAL PRIMARY KEY,
    scale_set VARCHAR(50)
);

CREATE TABLE IF NOT EXISTS risk_of_violence_description (
    decile SMALLINT PRIMARY KEY,
    raw_min REAL NOT NULL,
    raw_max REAL NOT NULL,
    txt_label VARCHAR(15)
);


CREATE TABLE IF NOT EXISTS risk_of_recidivism_description (
    decile SMALLINT PRIMARY KEY,
    raw_min REAL NOT NULL,
    raw_max REAL NOT NULL,
    txt_label VARCHAR(15)
);

CREATE TABLE IF NOT EXISTS risk_of_failure_to_appear_description (
    decile SMALLINT PRIMARY KEY,
    raw_min REAL NOT NULL,
    raw_max REAL NOT NULL,
    txt_label VARCHAR(15)
);

CREATE TABLE IF NOT EXISTS assessments(
    assessment_id SERIAL PRIMARY KEY,
    case_id INT NOT NULL,
    person_id INT NOT NULL,
```

```
    scaleSet_id INT NOT NULL,
    screening_date DATE NOT NULL,
    assessment_reason VARCHAR(50),
    legal_status VARCHAR(50),
    custody_status VARCHAR(50),
    rec_supervision_level SMALLINT NOT NULL,
    rec_supervision_level_text VARCHAR(50) NOT NULL,
    rov_raw REAL NOT NULL,
    rov_score_txt  VARCHAR(20),
    ror_raw REAL NOT NULL,
    ror_score_txt VARCHAR(20),
    rofa_raw REAL NOT NULL,
    rofa_score_txt  VARCHAR(20),

    CONSTRAINT case_id FOREIGN KEY(case_ID)
    REFERENCES cases(case_id)
    ON DELETE CASCADE ON UPDATE CASCADE,

    CONSTRAINT person_id FOREIGN KEY(Person_ID)
    REFERENCES people(person_id)
    ON DELETE CASCADE ON UPDATE CASCADE,

    CONSTRAINT scale_set_id FOREIGN KEY(ScaleSet_ID)
    REFERENCES scaleSets(scaleSet_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```