Creació d'una llibreria de Python per a la anonimització de dades

Nil Folquer Covarrubias

Resum– La recopilació de dades personals és un dels temes més populars avui dia a la nostra societat i dels més polèmics. Estem vivint una era on tothom està connectat i això fa que les empreses facin negoci amb les dades que deixem al nostre pas a internet. En aquest projecte, es buscarà estudiar les maneres d'aconseguir anonimitzar un conjunt de dades de manera que aquest no pugui ser re-identificable i no perdi informació. La capacitat de poder anonimitzar les dades facilita la seva recol·lecció i el poder compartir o vendre sense incomplir les lleis vigents en tema de protecció de dades. Per tant, es crearà una llibreria per al llenguatge *Python* que sigui capaç d'aplicar diversos mètodes de anonimització de dades i de proveir també eines que determinin la capacitat de pèrdua d'informació o risc de re-identificació de dades ja anonimitzades.

Paraules clau– python, pandas, sdcMicro, anonimització de dades, rank swapping, privacitat

Abstract– The collection of personal data is one of the most popular topics in our society today and one of the most controversial. We live in an era in which we are connected and that makes it possible for companies to negotiate with the people who let us pass through the Internet. In this project, we will seek to study ways to anonymize a set of data so that it can not be re-identifiable and not lose information. The ability to anonymize data makes it easier to collect, share or sell it without violating data protection laws. Therefore, a library will be created for the *Python* language that is still able to apply various methods of data anonymization and also to provide those that determine the capacity of loss of information or re-identification risk of data already anonymized.

Keywords– python, pandas, sdcMicro, data anonimization, rank swapping, privacy

1 Introducció - Context del treball

En els ultims anys un dels temes més populars en quant a debat tecnològic és el de l'enorme quantitat de dades que les empreses propietaries de xarxes socials o serveis de missatgeria instantània que recopilen sobre tu i amb les que fan negoci. A aquest fenòmen se'l anomena *Big Data*. Aquestes dades tenen un gran valor a nivell personal ja que amb el suficient estudi d'aquestes es poden determinar gustos, personalitats, interessos de la gent, però també comporta un problema en quant a la seguretat i privacitat de les persones. Amb la fi de poder garantir la seguretat i la privacitat de les persones en vers les seves dades recollides, la Unió Europea va crear l'any 2018 el *Reglament General de Protecció de Dades (GPDR)*[1]

La GPDR requereix que les organitzacions obtinguin explícitament el consentiment dels individus abans de recopilar i processar les seves dades personals. També dona a les persones el dret a accedir, corregir i esborrar les seves dades. Aquesta llei per a les empreses suposa un gran repte per a les empreses que es basen en el *Big Data* ja que aquestes han d'aconseguir el consentiment de millions de persones i que a més aquestes són capaces d'accedir, modificar o esborrar les seves dades de manera ràpida i eficient.
Una solució a aquest problema és l'anonimització de dades, que és el procés d'esborrar o ofuscar dades personal d'un conjunt de dades de manera que no es pugui re-identificar ningún individu [reference]. Això permet que les organitzacions puguin seguir recopilan dades per al seu anàlisi sense arriscar-se a incomplir amb la normativa europea.
Hi ha moltes tècniques d'anonimització tals com la supressió, la generalització i la perturbació. En aquest informe s'explicaran algunes d'aquestes tècniques i s'implementaràn utilitzant el llenguatge de programació *Python*.

- E-mail de contacte: nil.folquer@uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Guillermo Navarro Arribas
- Curs 2022/23

2 OBJECTIUS

A continuació es detallen els objectius principals proposats per a la correcta realització del treball ordenats per prioritat:

- Aprendre els diferents tipus d'algorismes que permeten anonimitzar un conjunt de dades. Aquest algorismes són:
  - *Noise*
  - *RecordSwap*
  - *RankSwap*
  - *LocalSupression*
  - *RandomSampling*
- Implementar els següents algorismes anteriorment mencionats:
  - *Noise*
  - *RankSwap*
- Els algorismes abans mencionats s'implementaran utilitzant diverses subfuncions on l'usuari podrà escollir a conveniència quin mecanisme vol utilitzar.
- Implementar un mecanisme de càrrega de conjunt de dades.
- Implementar algun mecanisme de quantificació en quant a la anonimització aconseguida al utilitzar els algorismes implementats.
- Aquestes implementacions han d'estar recollides en una llibreria de Python que pugui facilitar el seu ús.
- Implementar variants dels algorismes escollits i comparar-ne resultats.

Com a objectius opcionals es proposa:

- Implementar els algorismes restants abans mencionats:
  - *RecordSwap*
  - *LocalSupression*
  - *RandomSampling*
- Implementar un mecanisme de visualització (gràfics, diagrames, etc) per a la visualització i comparativa de les dades tractades.

3 ESTAT DE L'ART

Actualment existeixen múltiples opcions en quant a tècniques per anonimitzar dades i cada cop van apareixent noves tècniques i es van desenvolupant noves tecnologies per fer que les empreses puguin seguir analitzant les dades dels seus usuaris sense que s'incompleixi ninguna llei.

Una de les tècniques més usades és la privacitat diferencial la qual permet garantir mitjançant la incorporació de soroll al conjunt de dades que no hi haurà pèrdua d'informació al fer un anàlisi [2].
Una altra tècnica és basa en l'anomenat xifrat homomòrfic. D'aquesta manera es poden xifrar les dades i per tant que siguin il·legibles per a un possible atacant però a la vegada es pugui seguir analitzant aquestes dades sense la necessitat de desencriptar les dades del conjunt.

Per últim hi ha una tendència a utilitzar dades sintètiques però similars a dades reals amb la diferència que aquestes no contindran mai dades personals. Per a poder generar aquestes dades artificials s'utilitzen amb freqüència models entrenats amb xarxes neurals [3].

Per a totes aquestes tècniques existeixen multitud de llibreries de software que faciliten el procés de la transformació de les dades tals com: *sdcMicro* [4], *anonypy* [5], *ecto_anon* [6], *ARX* [7], *anonymizedf* [8], etc. Durant aquest treball s'ha utilitzat la llibreria *sdcMicro* per al llenguatge *R* com a inspiració juntament amb la llibreria *open-data-anonymizer* per al llenguatge *Python*.

4 METODOLOGIA

Per a la realització d'aquest treball, s'utilitzarà una metodologia *Agile* per a fer la implementació de la llibreria. Aquesta metodologia és perfecte per a projectes que necessiten flexibilitat i una data d'entrega curta. Agile divideix el projecte en diverses tasques les quals s'han de completar i entregar en qüestió de poques setmanes. Al mateix temps aquestes tasques es poden dividir en subtasques.

Per a la organització d'aquesta metodologia comptarem amb l'ajuda d'una eina anomenada Github Projects la qual ens permet representar visualment les tasques i subtasques, el seu estat (per començar, en curs i finalitzada) i el temps restant de cada una. Aquesta eina es va decidir utilitzar degut a que al treballar amb codi, aquest es guarda en un repositori del mateix Github i per tan dona versatilitat a l'hora d'organitzar les tasques a fer si tot està en un mateix lloc.

Quan es va començar el projecte, es va obrir una *issue* per tasca. Una *issue* es un espai dins d'un repositori que permet rastrejar i fer un seguiment del projecte durant el seu desenvolupament. Les *issues* poden representar idees, tasques, errors... [9] Aquestes *issues* es representen després en un espai de projecte que també proporciona Github (Github Projects) i que permet llistar les *issues* amb una vista de Kanban. Kanban és un mètode per a la gestió del flux d'un treball. Ajuda a definir, gestionar i millorar l'eficiència del treball [10]. Aquest mètode consisteix en un tauler amb tres columnes: To-Do, In Progress, Done. A la primera columna estan les tasques que encara estan per realitzar, a la columna In Progress es troben les tasques en les que s'està treballant actualment i l'última entrada reflexa les tasques acabades. Cada *issue* conté el nom de la tasca, una breu descripció d'aquesta i la data d'inici i fi. Les *issues*, tenen també un apartat de comentaris. Es va decidir que per a cada avenç important dins de la tasca, s'escrivís un petit comentari explicant la feina feta.

5 DESENVOLUPAMENT

Durant aquest treball s'ha desenvolupat una llibreria de *Python* que implementa mètodes d'anonimització de dades així com d'avaluació de la privacitat. Aquesta llibreria té com a dependències: *pandas*, *numpy* i *pyreadr*, sent *pandas* la més important ja que utilitzarem l'objecte *dataframe* que és el que s'utilitza per operar amb els conjunts de dades carregats en memòria. L'objectiu doncs, és utilitzar l'herència que

pandas ofereix per tal de crear un objecte amb les mateixes funcionalitats que l'objecte *dataframe* original de *pandas* i estendre'l amb les funcions d'anonimització i de detecció de pèrdua d'informació [11].
A continuació es detallen els mètodes que s'implementaran a la llibreria.

5.1 Mètodes d'anonimització

Aquest mètodes s'encarreguen d'anonimitzar les dades d'un *dataset* per tal de reduir la probabilitat de re-identificació de la columna de dades a la que s'aplica la funció.

La manera d'anonimització varia en funció del mètode escollit i a més depèn del tipus de dades que s'emmagatzema en la columna desitjada. En aquest treball s'han desenvolupat dos mètodes de tipus pertorbatiu és a dir que alteren les dades i només les dades de tipus numèriques.

5.1.1 Soroll additiu no correlacionat

Les funcions de soroll additiu són les encarregades d'afegir error a les dades numèriques per tal de modificar-les. La següent funció matemàtica descriu el procès d'adició de soroll:

$$X' = X + \varepsilon$$

On X és la dada a modificar, ε és el soroll a afegir i X' és el resultat d'aplicar el soroll a la dada. Això s'aplica per a totes les dades dins d'una columna de dades seguint una distribució normal. La funcio encarregada de generar el número aleatori en base a una distribució normal es realitza mitjançant la llibreria de *Python numpy* i que es pot representar de la següent manera:

$$\varepsilon = N(\mu, \sigma * p)$$

En aquesta funció, μ fa referència a la mitjana que ha de ser 0 ja que volem que sigui una distribució normal estàndard, σ és la desviació estàndard de les dades, obtinguda cridant a la funcio *numpy.std()* de la llibreria *numpy* i que aquesta és multiplicada per p, una variable amb un rang entre 0 i 100 que estableix l'usuari i que indica la quantitat de soroll (en percentatge) que es vol aplicar [12].

5.1.2 Soroll additiu correlacionat

El soroll additiu correlacionat s'utilitza quan es vol mantindre la covariància i la correlació entre el conjunt de dades original i el conjunt modificat ja que la matriu de covariància dels errors generats(el soroll aplicat) és proporcional a la covariància de les dades originals. La formula per aplicar soroll additiu correlacionat és la següent:

$$X' = X + \varepsilon$$

A simple vista s'observa que la funció és la mateixa que en el cas no correlacionat. La diferència però, la trobem en la generació de la variable ϵ:

$$\varepsilon = N(\mu, cov(X) * p)$$

On μ representa la mitjana aritmètica de la columna X, $cov(X)$ és el resultat d'aplicar la funcio *numpy.cov()* de *numpy* amb la columna a aplicar el soroll com a paràmetre y que aquesta és multiplicada per p, una variable amb un rang entre 0 i 100 que estableix l'usuari i que indica la quantitat de soroll (en percentatge) que es vol aplicar. A major percentatge, major és el soroll generat.

Original	Anonimitzada
25	27
38	39
28	29
44	39
18	17
34	37
29	26
63	64
24	23
55	54

TAULA 1: SOROLL ADDITIU CORRELACIONAT AMB UN NOISE DEL 20%

5.1.3 Rank Swapping

Rank swapping és un mètode d'anonimització de dades que consisteix en intercanviar les files d'un conjunt de dades sempre que es trobin dins d'un mateix rang. Aquest mètode serveix tant per a dades numèriques com per a dades ordinals, és a dir una variable categòrica que es troba dins d'una escala definida (e.g ['primaria', 'secundaria' i 'batxillerat'] per referir-se al nivell educatiu) com per a dades numèriques ja que aquests dos tipus de dades poden ser ordenats. L'algorisme de *rank swapping* pot ser aplicat en un o més camps sempre i quan tots compleixin les condicions explicades anteriorment. En cas d'aplicar el mètode en més d'un camp, s'aplicarà de manera independent a cada columna de dades sense guardar cap relació entre elles. L'algorisme és el següent:

Algorithm 1 Rank Swapping: *rankSwap(X, p)*

1: $(a_1, ..., a_n)$ ←valors de la columna X ordenats en ordre ascendent
2: Marquem a_i com a valor *unswapped*
3: **for** i=1 **to** n **do**
4: **if** a_i == *unswapped* **then**
5: Seleccionem una posició l de manera aleatòria dins d'un rang determinat $[i+1, min(n, i+p*|X/100)]$
6: Canviem a_i per a_l
7: Desfem l'ordre ascendent d'X;

Podem veure que per utilitzar aquesta funció cal especificar un percentatge *p* com a paràmetre de funció. *p* correspon a un percentatge del total d'entrades a la columna *X* [13].

5.2 Mètodes d'avaluació

Els mètodes d'avaluació són aquelles funcions que s'encarreguen d'avaluar el grau d'anonimat de les dades un cop aplicades les funcions abans esmentades o el de determinar el nivell de pèrdua d'informació de les dades anonimitzades envers les dades originals. Per poder utilitzar aquestes funcions, s'haurà d'haver modificat un *dataset* original amb algun dels mètodes prèviament explicats i en tots els casos les dades hauran de ser de tipus numèric.

5.2.1 Risc de re-identificació

El risc de re-identificació és defineix com estimar el risc d'identificar un registre dins d'un conjunt de dades protegit utilitzant funcions d'anonimització on es tenen algunes dades del *dataset* original. Aquest risc es calculat de manera empírica és a dir es genera de manera computacional en base a un input i per tant es basa en un seguit d'algorismes que s'encarreguen de comparar i establir relacions entre les dades protegides.

5.2.2 Pèrdua d'informació

La pèrdua d'informació s'utilitza per quantificar el nivell de pèrdua d'informació d'un conjunt de dades protegit. Per tant, aquesta pèrdua d'informació correspon a la diferència que hi ha entre les dades protegides i les originals. Com més gran sigui aquest nivell de pèrdua, més informació s'ha perdut durant la seva anonimització.
Sigui *X* un conjunt de dades i *X'* el mateix conjunt de dades però protegit, establim que:

$$PI_f(X, X') = \textit{divergència(f(X), f(X'))}$$

on definim que *divergència* és una manera de comparar dos elements.
En el cas de les dades numèriques ens basem en fets estadístics per a determinar la pèrdua d'informació de les dades anonimitzades. En aquest treball s'han implementat 3 definicions de divergència [14]:

$$divergencia_{MSE}(M, M') = \frac{\sum_{ij}(M_{ij} - M'_{ij})^2}{c(M)}$$

$$divergencia_{MAE}(M, M') = \frac{\sum_{ij}|M_{ij} - M'_{ij}|}{c(M)}$$

$$divergencia_{MRE}(M, M') = \frac{\sum_{ij}\frac{|M_{ij} - M'_{ij}|}{|M_{ij}|}}{c(M)}$$

on $c(M)$ és el número d'elements de la columna M.
A partir d'aquestes tres funcions de divergència podem establir:

$$PI_{Id}(X, X') = divergencia_d(X, X')$$

$$PI_{Cov}(V(X), V(X')) = divergencia_d(V, V')$$

En aquesta última equació, *V(X)* i *V(X')* indiquen que la pèrdua d'informació es calcula en base a la covariància (*Cov*) dels dos *datasets*

5.3 Mètodes adicionals

Per a poder completar la implementació de la llibreria, cal afegir uns quants mètodes que donguin utilitats extra i en alguns casos essencials per a que pugui ser utilitzada de forma correcta.

5.3.1 Mètodes de càrrega de datasets

Per a poder treballar amb una llibreria de tractament de dades, primerament el que es necessitarà serà carregar els conjunts de dades a memòria. Només acceptarem els següents formats d'arxiu:

- **csv**: es tracta d'un tipus de fitxer on el contingut està separat per comes. Normalment la primera línia del document representa les capçaleres de la taula. És el format més utilitzat dins del *data science* per la seva facilitat de lectura [15].
- **rda/rdata**: el format *rdata* és el format utilitzat dins del llenguatge *R*. Aquest fitxer està disposat com a una seqüència d'objectes d'un tipus determinat (e.g. nombres, cadenes ASCII, etc) [16]. No es tant comú però com que el treball es basa en una implementació escrita en *R*, es vol donar l'opció de poder llegir aquests arxius.

5.3.2 Mètodes de visualització de resultats

Les columnes de dades anonimitzades poden ser vistes juntament amb la columna de dades original per tal de veure les seves diferències.

	source	anonimized
0	25	33
1	38	37
2	28	19
3	44	35
4	18	9
5	34	28
6	29	28
7	63	60
8	24	24
9	55	55

Fig. 1: Taula de comparació

5.3.3 Mètodes de resum

S'han implementat mètodes que recopilen informació sobre la relació entre les dades originals i les modificades a mode de resum i les mostra en una taula per a que l'usuari les analitzi de manera fàcil i ràpida.

5.3.4 Mètodes de càlcul de *performance*

Aquests mètodes serveixen per mesurar la velocitat amb la que la llibreria fa les operacions corresponents. Són una eina molt útil per als desenvolupadors ja que a partir d'aquests números, es poden determinar millores i optimitzacions que

puguin fer la llibreria més ràpida i lleugera. Aquestes funcions es basen en el mòdul *time* de *python* que és capaç de mesurar el temps en l'ordre dels nanosegons.

6 Resultats

En aquesta secció es realitzarà una prova de totes les funcions implementades dins de la llibreria. Amb els resultats d'aquestes proves es podran extreure conclusions que seran interpretades i explicades.

6.1 Conjunts de dades utilitzats

Per a poder realitzar les proves es necessita primer una font de dades. A continuació es detallen quatre conjunts de dades que s'utilitzaran en les proves. Aquests *datasets* han sigut descarregats a través del lloc web *kaggle*:

- **Adult income** Conjunt de dades sobre el salari anual, el nivell escolar, el tipus de treball, estat civil, etc, de persones adultes. [17]
- **NBA rookies** Aquest *dataset* guarda les estadístiques de diversos jugadors de bàsquet de l'NBA durant el seu any de debut. [18]
- **Hotel reservations** En aquest conjunt de dades trobem diverses reserves fetes a diferents hotels amb informació tal com la durada de l'estada, l'any de la reserva, el preu, etc. [19]
- **Day to day USA covid cases** Aquest *dataset* conté informació sobre els casos de covid diaris als diferents comtats d'Estats Units. [20]

En la següent taula, es mostra la quantitat de registres que formen part del conjunt:

Conjunt de dades	Registres
NBA rookies	1538
Hotel reservations	36275
Adult income	48842
Day to day USA covid cases	627920

TAULA 2: Quantitat de registres per a cada dataset

Aquests conjunts de dades contenen dades numèriques i categòriques entre d'altres. Només es podran fer les proves amb les columnes amb dades de tipus numèric ja que és l'únic format de dades acceptat per les funcions implementades dins de la llibreria.

6.2 Anàlisi de resultats

Per a cada funció implementada, s'ha realitzat un seguit de proves amb els *datasets* mencionats en el punt anterior. Per a poder quantificar un resultat en les funcions implementades, s'ha decidit utilitzar la funció:

$$divergencia_{MSE}(M, M')$$

que s'encarrega de calcular l'error quadràtic mitjà entre el conjunt de dades original i el conjunt anonimitzat.
En el cas de les funcions de soroll hem decidit aplicar la funció en un rang entre 0 a 100 en passos de 5 (e.g. [0, 5, 10, 15, 20, ..., 95, 100]). Per tant executarem la funció 20 vegades. El mateix farem per a la funció de *rank swapping* ja que aquesta pren com a paràmetre un percentatge equivalent al rang d'intercanvi de valors.

6.2.1 Resultats soroll additiu no correlacionat

Per a aquest test, s'ha aplicat la funció *addNoise* amb el paràmetre *additive* que indica que es vol utilitzar la funció de forma no correlacional per a la columna *avg price per room* del conjunt de dades *hotel reservations* i s'ha aplicat a varies columnes soroll additiu no correlacionat dins del conjunt de dades *adult income*.
Es pot observar en les figures 2 i 3 que l'augment de pèrdua d'informació augmenta de forma exponencial a mesura que augmentem el percentatge de soroll. S'aprecia que aproximadament fins al 30% de soroll la quantitat d'informació que es perd és ínfim i per tant podem aconseguir bons resultats en quant a nivell d'anonimat i conservant la majoria de la informació per poder fer estudis sense que aquests es vegin afectats pel soroll aplicat. També es pot observar com el tipus de dada numèrica no afecta de cap manera amb l'error que produeix la seva versió anonimitzada ja que per a aquesta prova s'han utilitzat dues dades de rangs diferents com és el preu d'una habitació d'hotel, l'edat d'una persona o les hores treballades per setmana i en tots els resultats l'error és similar.

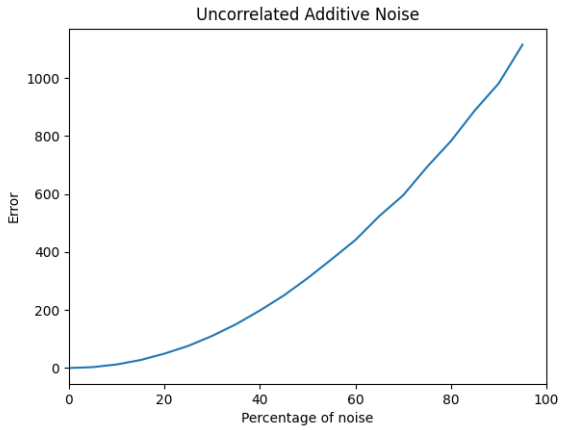

Fig. 2: Soroll additiu no correlacionat aplicat al dataset *Hotel Reservations* a la columnna *avg price per room*

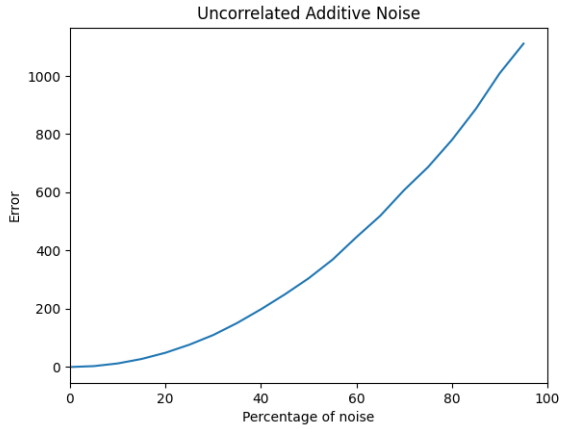

Fig. 3: Soroll additiu no correlacionat aplicat al dataset *Adult income* a les columnnes: *age, fnlwgt, hours per week*

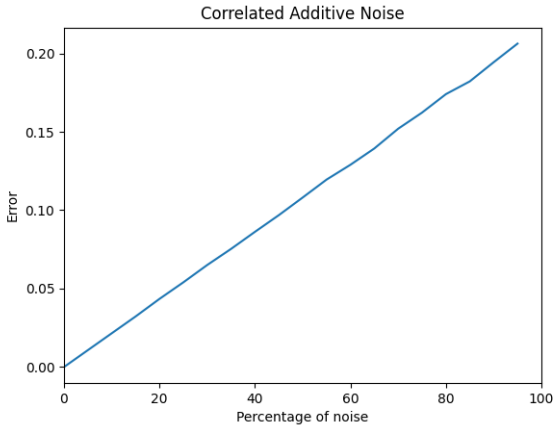

Fig. 4: Soroll additiu correlacionat aplicat al dataset *hotel reservations* a les columnes *number of adults i number of children*

6.2.2 Resultats soroll additiu correlacionat

En aquest apartat, es compararan els nivells de pèrdua d'informació entre les següents columnes dels conjunts de dades a les que s'aplicarà soroll additiu correlacionat:

- Al conjunt *hotel reservations* aplicarem el soroll en les columnes: *number of adults* i *number of children*. Volem mantenir la relació de dades entre aquestes dos columnes ja que la informació que contenen ha de mantindre un cert criteri si es vol en un futur fer algun tipus d'anàlisi.
- Al conjunt *NBA rookies* aplicarem soroll a les columnes: *GP, MIN, PTS* que contenen la informació sobre els partits jugats, els minuts que el jugador va jugar i els punts anotats. Es vol mantenir la relació numèrica entre aquestes dades ja que han de mantenir certa coherència si es vol fer un futur estudi de les dades.

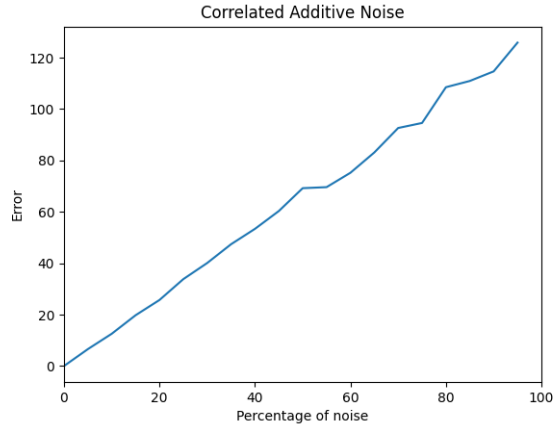

Fig. 5: Soroll additiu correlacionat aplicat al dataset *NBA rookies* a les columnes: *GP (partis jugats), MIN (minuts jugats), PTS (punts anotats)*

Pel que fa al soroll additiu correlacionat, podem treure com a conclusió que la quantitat d'informació perduda és proporcional al nivell de soroll aplicat, és a dir les gràfiques dels resultats tenen forma lineal, una diferència considerable en comparació amb el soroll additiu sense correlació entre dades. Per tant, en les figures 4 i 5, el nivell de pèrdua d'informació és més baixa que en el soroll no correlacionat ja que les dades segueixen guardant una relació entre elles i que per tant, provoca que l'error no es dispari com en el cas anterior. Aquesta funció hauria de ser utilitzada quan en un conjunt de dades a anonimitzar, dues o més columnes han de conservar la relació numèrica, com és el cas dels partits jugats amb els minuts jugats en el *dataset NBA rookies* ja que un jugador per exemple, no pot jugar més minuts que la suma de minuts totals dels partits que ha jugat durant la seva carrera. Gràcies a aquesta relació aconseguim que l'error que generem al aplicar la distorsió tot i que en grans quantitats faci que el número augmenti, segueix mantenint sentit en quant a lògica de les dades.

6.2.3 Resultats *rank swap*

En el cas del *rank swapping* s'han executat la funció utilitzant els conjunts de dades *NBA rookies* i s'ha procedit a desordenar la columna *partits jugats (GP)* i el conjunt *hotel reservations* utilitzant la columna de *lead time*.
Quan apliquem *rank swapping*, trobem que les dades de la columna a la que estem aplicant la funció tenen molt a veure amb el resultat final d'aquesta. Tot i així, es pot constatar que un baix percentatge conserva gran part de la informació de les dades mentre que a percentatges més alts, la pèrdua és de tipus exponencial fins a arribar al 90% o inclús 80% (com és el cas del resultat de *NBA rookies*), on la pèrdua d'informació és la mateixa i per tant perd tota la efectivitat. Tal i com passa en el cas del soroll correlacionat, podem deduir que el tipus de dada utilitzada influeix en el nivell d'informació que s'obtindrà un cop aplicat el mètode.

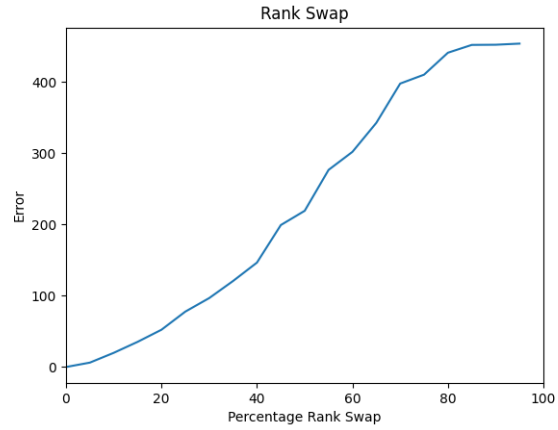

Fig. 6: Rank swap aplicat al dataset *NBA rookies* a la columna *GP(partits jugats)*

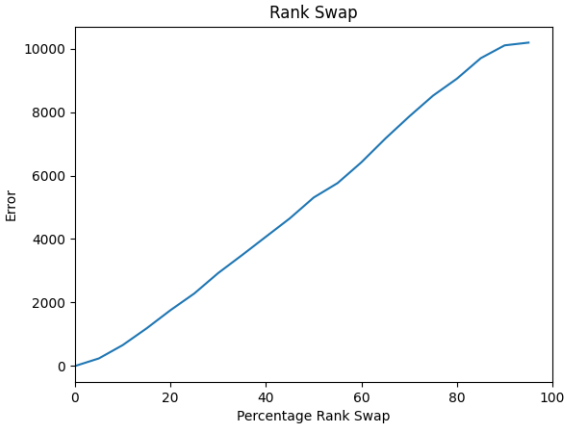

Fig. 7: Rank swap aplicat al dataset *hotel resevations* a la columna: *lead time*

6.3 Anàlisi de rendiment

Per acabar, s'ha decidit fer una prova de rendiment a les diferents funcions implementades amb l'objectiu de comparar-les amb l'implementació d'*R: sdcMicro*. Aquesta prova consisteix en executar les funcions de *noise* i *rankSwap* utilitzant 3 *datasets* amb una quantitat de registres diferents. Per a fer aquest test s'utilitzaran els següents conjunts de dades:

- **NBA rookies**: 1538 registres.
- **Hotel reservations**: 36275 registres.
- **Day to day USA covid cases**: 627920 registres.

Al principi de la funció es guarda el temps exacte en el que la funció ha començat a executar-se i es guarda altre cop el temps quan la funció acaba. Amb aquestes dos dades doncs, es possible calcular el temps que ha trigat la funció en córrer amb la següent equació:

$$t_{total} = t_{final} - t_{inici}$$

Cal remarcar que totes les funcions siguin quins siguin els paràmetres donats a la funció, operarà amb tots els registres. Això significa que es passin els paràmetres que es passin, els temps d'execució serà el mateix aproximadament ja que factors externs a la implementació poden fer augmentar aquest temps. Tot i així s'ha definit que per a les funcions de *noise*, la quantitat de soroll a aplicar sigui del 20% i en el cas del *rank swap* aquest tingui un rang del 20%. La següent taula mostra els resultats de la prova de rendiment:

	Soroll additiu no correlacionat	Soroll additiu correlacionat	Rank Swap
NBA rookies	0.001 segons	0.002 segons	0.072 segons
Adult Income	0.004 segons	0.002 segons	3.622 segons
Day to day USA covid	0.087 segons	0.088 segons	9 minuts i 20 segons

TAULA 3: TEMPS D'EXECUCIÓ DE LES DIFERENTS FUNCIONS EN BASE ALS DIFERENTS CONJUNTS DE DADES

Com es pot observar, les dues funcions relacionades amb l'addició de soroll no triguen cap cas ni una dècima de segon en completar-se. Això és degut a que les operacions que s'hi duen a terme són operacions simples com la suma i que a més aquestes estan optimitzades gràcies a la llibreria *pandas* per a poder sumar vectors en el menor temps possible. La petita diferència de temps entre el soroll correlatiu i el no correlatiu és causat pel càlcul de la covariància.
On si que trobem uns resultats molt significatius és en la funció de *rank swap*. Com es pot veure la diferència de temps entre conjunts de dades a mesura que el nombre de registres augmenta, és molt considerable. La conseqüència d'aquest rendiment és degut a com està feta la funció ja que és necessari per poder canviar dos elements de lloc accedir a la posició de memòria del primer element i guardar-ne el valor, accedir a la posició del segon element i canviar el valor i per últim, accedir novament a la posició anterior i guardar-ne el valor. És a dir, per fer un simple canvi de valors s'han de fer 3 accessos a memòria i això suposa un cost de temps molt elevat sense tindre en compte que s'està utilitzant un llenguatge que no treballa directament amb memòria i que per tant és més lent que un llenguatge de baix nivell. Una possible optimització d'aquesta funció, seria fer els canvis de valors en grup o fer ús del *framework cython* que dóna la possibilitat d'escriure funcions en *C*, que és un llenguatge de més baix nivell que *Python* i que per tant fa un millor ús de la memòria, el que provoca un millor rendiment. Quan la funció acaba de ser executada, *cython* trasllada el resultat a *Python* [21].

7 CONCLUSIONS

Durant el curs d'aquest treball, s'han pogut completar tots els objectius proposats dins del temps establert en la planificació del projecte i per tant s'ha desenvolupat de forma correcta.
En quant als resultats, la llibreria tot i tenir les funcions proposades com a objectius, dista molt d'estar completa per a poder ser utilitzada en un àmbit més professional.
El fet que la llibreria estigui desenvolupada utilitzant altres llibreries com *pandas* o *numpy* com a base, fa que la tasca de desenvolupament no s'hagi allargat al no tenir que implementar utilitats tals com estructures de dades per a guardar els *datasets*, funcions per a la càrrega de fitxers, etc. Es pot concloure llavors que les funcions estan desenvolupades de

forma correcta tot seguint els manuals obtinguts en la recerca i l'anàlisi de la llibreria *sdcMicro* que tal i com s'ha mencionat prèviament, és la llibreria en la que s'ha basat aquest treball. En el cas de les funcions de soroll additiu, correlacional i no correlacional, com a part positiva podem destacar que s'executen de manera ràpida i donen l'opció de treballar amb múltiples columnes dins d'un conjunt. Com s'ha pogut observar en l'apartat de resultats, la funció de soroll no correlacional té una tendència a perdre informació quan més soroll hi apliquem sempre, independentment del tipus de dada que estiguem intentant anonimitzar, mentre que en el cas del soroll correlacional, la quantitat de pèrdua d'informació no és tan elevada però implica tindre que relacionar dues variables que puguin guardar relació entre elles com era el cas dels partits i els minuts jugats pels jugadors de la *NBA* el qual pel simple fet de tindre que mantindre la relació entre partits i minuts fa que la pèrdua d'informació disminueixi de forma considerable.
Amb el cas de la funció de *rank swap* la situació és diferent, ja que l'execució d'aquesta funció comporta un poder de computació elevat. *sdcMicro* utilitza codi de baix nivell escrit en el llenguatge *c* per a poder fer tipus d'operacions i així aconseguir millors temps d'execució. En el cas de la implementació en *python*, no es fa ús d'un altre llenguatge que pugui dur a terme l'execució de la funció i per tant reduir el temps d'execució. La llibreria té més utilitats com els mètodes de quantificació de pèrdua d'informació d'un conjunt de dades anonimitzat respecte al original i una funció que compara dos columnes iguals amb la finalitat de mostrar la columna original i la modificada entre d'altres.
Per acabar, aquest projecte es podria continuar desenvolupant per a millorar el que ja existeix i per afegir noves funcionalitats. Aquestes són algunes de les línies de continuació d'aquest projecte:

- Optimització de la funcio *rank swap* utilitzant el framework *cython* que permet escriure funcions de *python* en codi *C* per a fer-les més òptimes.
- Afegir més funcions d'anonimització pertorbatius: *record swap*, *microaggregation*, etc.
- Afegir funcions d'anonimització de tipus no pertorbatiu: *local supression* o globalització de variables.
- Possibilitat de generar dades sintètiques ja sigui a partir de dades existents o crear-ne un de nou utilitzant models d'intel·ligència artificial.
- Fer la llibreria compatible amb *Jupyter Notebook*.

AGRAÏMENTS

Vull agrair aquest treball a la meva família, en especial als meus pares per tot el suport que m'han donat durant aquests anys. També agraeixo als meus amics i al Sindic totes les estones que hem passat. Per acabar vull donar les gràcies al meu tutor del treball pel suport que m'ha donat.

REFERÈNCIES

[1] BOE. (2018, Desembre) Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. [Online]. Available: https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673

[2] A. E. de Protección de Datos. (2023, Octubre) Anonimización y seudonimización (ii): la privacidad diferencial. [Online]. Available: https://www.aepd.es/es/prensa-y-comunicacion/blog/anonimizacion-y-seudonimizacion-ii-la-privacidad-diferencial

[3] N. C. Abay, *Privacy Preserving Synthetic Data Release Using Deep Learning*, Octubre 2018.

[4] M. Templ. sdcmicro: Statistical disclosure control methods for anonymization of data and risk estimation. [Online]. Available: https://cran.r-project.org/web/packages/sdcMicro/index.html

[5] T. Fujita. (2022) Anonypy. [Online]. Available: https://github.com/glassonion1/anonypy

[6] C. Quaresma. (2022) ecto_anon. [Online]. Available: https://github.com/WTTJ/ecto_anon

[7] F. Prasser. (2022) Arx. [Online]. Available: https://arx.deidentifier.org/

[8] A. Frid. (2020) anonymizedf. [Online]. Available: https://github.com/AlexFrid/anonymizedf

[9] G. Docs. About issues. Github. [Online]. Available: https://docs.github.com/es/issues/tracking-your-work-with-issues/about-issues

[10] kanbanizer.com. (2022) Que es kanban. [Online]. Available: https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban

[11] Pandas. (2022) Extending pandas. [Online]. Available: https://pandas.pydata.org/docs/development/extending.html

[12] V. Torra, *Data Privacy: Foundations, New Developments and the Big Data Challenge*, S. I. Publishing, Ed., 2017.

[13] I. H. S. Network. Anonymization methods — sdc practice guide documentation. [Online]. Available: https://sdcpractice.readthedocs.io/en/latest

[14] M. Templ, *Statistical Disclosure Control for Microdata*, S. I. Publishing, Ed., 2017.

[15] Y. Shafranovich. (2005, Octubre) Common format and mime type for comma-separated values (csv) files. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4180.html

[16] loc.gov. (2017, Juny) R data format family (.rdata, .rda). [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000470.sht

[17] B. B. Ronny Kohavi. Adult data set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/adult

[18] G. Salzer. Nba rookies performance statistics and minutes. [Online]. Available: https://www.kaggle.com/datasets/thedevastator/nba-rookies-performance-statistics-and-minutes-p

[19] A. Raza. Hotel reservations dataset. [Online]. Available: https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset

[20] D. K. P. Covid-19 dataset. [Online]. Available: https://www.kaggle.com/datasets/imdevskp/corona-virus-report

[21] S. Behnel. (2022) cython. [Online]. Available: https://cython.org/#about