POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Design and Implementation of a privacy-preserving framework for Machine Learning

Supervisors

Prof. Marco MELLIA

Dott. Ing. Martino TREVISAN

Dott. Ing. Nikhil JHA

Candidate

Giovanni CAMARDA

April 2021

Abstract

During the last decade, a myriad of new technologies has changed the way society perceives everyday life, embodying the Big Data Era peculiarities. Almost every technological scenario produces an incredible amount of data, from disparate physical sources and at a very different generation rate, creating an interconnected and interdependent network of people and data. For this reason, data has become for companies and organizations a strategical asset to drive businesses, to tailor user-specific services and to obtain a more relevant position on data markets. More and more companies collect and process customers' personal data requiring it in exchange for services, forcing users to accept a power unbalanced transaction. To tackle this situation, regulations as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) were signed in 2018 and 2020, enforcing data protection respectively in the European Union and California State: their primary goal consists in support free data flow, building trust conditions and rebalancing powers in the relationship between companies and customers.

In this context legal frameworks are necessary but not sufficient, since the absence of an international standard to technically implement data protection in data processing activities is a serious obstacle for companies. The European project PIMCity aims to narrow the gap between regulations and practical privacy-preserving solutions providing a modulable framework thanks to which companies can implement ad-hoc instruments. The PIMCity project provides diverse inter-operating components: the Personal Data Safe (P-DS) to store data from various sources, the Personal Privacy-Preserving Analytics (P-PPA) with which is possible to extract information preserving privacy, the Personal Consent Manager (P-CM) that models the user consent notion and the Personal Privacy Metrics (P-PM) to enhance users' awareness about their shared data.

This thesis presents a generic fully-fledged P-PPA module whose input data can be both in a structured and unstructured format. The project pipeline was developed in Python language, providing REST API to interact with it and exploiting privacy properties as k-anonymity and differential privacy. This module is used as the starting point to define a Machine Learning framework to analyze the amount of information gathered from anonymized

data. We further propose a deeper inquiry to investigate the correlation between increasing privacy constraints and the residual information level of the Machine Learning algorithms output.

Acknowledgements

This thesis work is the final result of this chapter of my life but in different ways, during these years, many people have taken part at this moment. For this reason, I would thank my parents that guided me through difficult moments without any pressure, my patient sister with whom I shared many fights since a child, my friends that represent a fundamental part of who I am and my beloved partner that just makes my life brighter day after day. Last but not least, I would thank my supervisors for their precious advices and the passion they put into their job.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivations

The historical period in which we are currently living is characterized by the most interconnected and interdependent technologies since the majority of organizations and in general technological entities are focusing their efforts to enhance productivity and strategical resources, gathering every kind of insights from data. Almost every human process can be data-driven, or at least data permit to tackles challenges from different points of view.
Data analysts consider the last decade as the Big Data era, characterized by a quasi-exponential increasing of data produced from every kind of different data source. More and more technologies are designed or developed considering the benefits that the Big Data era has proven to offer, boosting decision processes and addressing ad-hoc solutions in a high demanding sector as the technological one.
A large percentage of produced and gathered data is related to personal or private information since almost every technology we use in our everyday life processes and collects it, to grasp data insights for improving service quality or as an asset on data markets: one of the possible and more profitable business model consists in collecting from users desirable information that will be sold or shared with third parties; in fact, many times the user is referred to be the product himself. In several countries, there is almost no restriction that prevents companies to profile users: personal data are harvested so often that it could be perceived as normal, but this is a very risky attitude; the data subject can't know which information is collected or processed, he can't enforce any right to deny its data to be sold or shared.

The privacy and data protection individual's demand is often sacrificed or ignored in favour of a more efficient business conception: for years and even now, the majority of "technological citizens" can't count on an adequate framework that permits them to empower their rights over their personal information. For instance, referring to one of the most typical situations as web browsing, more and more users can describe their daily experience related to an unpleasant sensation of being deprived of their personal data since forced to accept some agreement in exchange for services.
Confining the debate only to web browsing can be considered extremely underestimating since, as clearly addressed in this introduction part, most human technological experiences are defined by this imbalance in powers. Regardless of the tangible technological benefits for the final users, it's by now clear the urgency for a legal and technological framework to permit a sort of power rebalancing. European General Data Protection Regulation (GDPR) [1], California Consumer Privacy Act (CCPA) [2] and other regulations are disclosing the way but, in practical sense, organizations and service providers need technological instruments to facilitate and sustain this shift in perceiving data protection as a fundamental right.
The current scenario proposes various solutions to handle and anonymize data but there is no actual standard and, for this reason, companies are not encouraged to adopt any technological instruments for data processing since the interaction with other realities would be difficult. In this context, the EU-funded PIMCity project [3] aims to establish itself as the standard for managing data flows from the source to the final users carefully handling each step in a privacy-oriented flavour: it includes data storage, anonymized data retrieval, user consensus and controlled market notions in an all-in-one modulable solution, allowing companies to build the framework most suitable for their necessities. PIMCity authors' mission consists of allowing an ethical use of personal information, increasing trust and collaboration between companies and data subjects.
This thesis will implement the Personal-Privacy Preserving Analytics (P-PPA) module, one of the PIMCity key components that will be present in the final PIMS Development Kit (PDK): this module can receive data in both structured and not structured data formats, can interact with outer layers thanks to the implemented REpresentational State Transfer (REST) API, can anonymize input data by applying privacy concepts from k-anonymity and differential privacy. Moreover, the thesis work will also investigate the performance of three publicly available datasets after the k-anonymization

process, assessing the anonymized data usefulness through Machine Learning (ML) algorithms.

Chapter 2

Background

2.1 The Big Data era

Researches estimate that by 2025 will be generated, manipulated and processed approximately 175 zettabytes[1] of data; the Big Data trend can be shaped as a quasi-exponential curve considering the 2010 and 2019 data sphere measurements, corresponding respectively to 2 and 41 zettabytes [4]. Data analysts predict that this trend will surely grow not only for the direction the society has undertaken during the last decades but also for the shift in habits and routines that the Covid-19 world pandemic has caused. It's often cheaper for companies to offer their services in a digital flavour; more and more activities are moving their businesses in this direction increasing data flows and the resulting concerning related to personal and sensitive information disclosure.
We refer to this phenomenon as Big Data, describing a data cluster revealing different characteristics compared with the ones everyone was familiar with before the digital era explosion. The data notion as a structured table is changed since we have to deal with a plethora of new inputs and sources as social media posts, health trackers, videos, Internet of Things (IoT) devices, autonomous drive sensors, Global Positioning System (GPS) and human-machine interaction sensors and so on.
Over the last decade, Big Data was studied and characterized by the already famous six Vs: volume, velocity, variety, variability, value and veracity. Big

[1] One zettabyte corresponds to 10^{21} bytes.

Data is defined by a large amount of different data, dozens of terabytes, produced at various generation rate from a heterogeneous range of sources. Data analysts must take care of the variability since it may be difficult to attach the same meaning to the same data when also plenty of different information has to be considered and, for this reason, also extract value from it is becoming as difficult as strategical. Moreover one of the Big Data features that nowadays is becoming more and more relevant is the data veracity, that is the reliability: working with even partially biased or incorrect data is more dangerous than not having them at all.
The Big Data era influenced and shaped the majority of nowadays technologies, creating a technological ecosystem with which society is by now familiar; a wide range of human activities, from the economical to social and ethical field, exists thanks to technologies whose pervasiveness and omnipresence are fed by Big Data paradigms and characteristics. For this reason, the potential impact that Big Data itself has on individuals needs to be carefully discussed and monitored since the very beginning of the design phase of both technological and non-technological activities.

2.1.1 Data value and data markets

More and more organizations, encouraged by the Big Data diffusion and popularity increase, begin to develop management instruments to lower the data complexity and, at the same time, master the processes that permit them to extract data insight, fundamental to drive company strategies in the medium-long period. This new business and development approach is not a commodity anymore, but has become an actual strategical advantage. Especially in the technological field, the more companies focus their efforts in expanding databases volume, variety, variability and veracity the more commercial value they obtain using Data Mining and ML techniques, creating indeed the concept of data market value as an asset. At present, information is traded between organizations and evaluated through a data pricing model that takes into account the data source, variety and degree of insights that it carries. [5]
For this reason, there exists an issue related to uncontrolled and not regulated data trading, especially because all the actors involved, mainly in the technological area, are well conscious that often the more data are related to personal and intimate individual behaviours the more information carries with it. The majority of data experts agree that it is necessary to deeply

recognize and monitor this situation, increasing the efforts to converge towards a common debate over the importance of data protection policies.

2.2 Technological risks for data protection enforcement

As things stand today, a very large variety of technologies and services are based on information inferred from personal or even on sensitive data, moreover the high level of complexity that characterizes the current technological ecosystem exposes to risks more and more users considering a data protection context.
One situation that must be carefully monitored is the rising of data breaches considering that personal and sensitive data, like password, social security service numbers and credit card data, represents the favourite target for obvious reasons. Experts consider cloud services paired to IoT devices one of the most vulnerable and exposed environment: in January 2020 the National security Agency (NSA) has published a report that underlines how, in addition to the IoT devices notoriety for security bugs, two of the principal vulnerabilities are related to systems configuration errors from administrators and some responsibility overlapping between customers and Communications Service Providers (CSP) [6]. Amazon Web Services and Microsoft Azure are just two from a long list of cloud services that were successfully attacked.
We now mention three scenarios in which technological progress needs to be put under investigation because they can implicate strongly invasive aspects for users and data subjects, even if they're often associated with a considerable improvement of the life quality or in some circumstances they're necessary: the industry 4.0, the facial recognition and the voice-activated devices technological environments.

Industry 4.0

Defined as a new industrial revolution, it enhances industrial dynamics remained almost unchanged for decades exploiting the recent years new hardware and software technologies. A substantial risk situation from a personal data perspective can be the employment of complete or partial robotic devices capable of simplifying a wide range of applications, making

them safe or even possible in cases in which an individual has physical limitations. This type of situation requires the employee's health data or biometric data and, in specific circumstances in which the subject suffers from a medical condition that can vary during time, they can't be anonymized or deleted.

Facial recognition

The facial recognition system is a technology that exploits the recognition of particular facial features to identify a human face starting from some subject's video or photo input. During the last ten years, facial recognition has been characterized by a strong push since has been widely utilised on very common devices like the smartphone.
Facial recognition systems are as simple and intuitive to use as dangerous in some circumstances: the caution and protection level that is required to process biometric data is directly proportional to its unicity itself. Moreover, this technology discloses one of the highest discrimination risks, since identifying individuals is the starting point that permits organizations, companies and governments to pursue discriminatory policies of any kind. Its beneficial aspects are evident but it is fundamental to take into account that there exist consequences that could be irreversible misusing it[2].

Voice-activated devices

The voice-activated devices, referring to both mobile or smart home environments, are more and more widespread and common, substituting the usual channels that have been used until these recent years to interact with machines. These devices utilise the microphone to search for the so-called "keywords" to activate themselves; when one of this command is found they remain in listening mode for a certain period after which the microphone is deactivated.
Exploiting Artificial Intelligence (AI) the developers can implement more and more functionalities that require to grant more privileges from the operating system. Moreover, these devices are often paired to IoT ones, creating a wide network made up of a myriad of different personal data.

[2]Who defines what is a possible misuse? According to what point of view? Ethical? Economical?

Last but not least, some of them have no other options to interact with apart from voice control, exposing the user to some attacks performed using a sort of device chain methodology, thanks to which it is possible to extract sensible information [7].

2.2.1 Data protection is a complex task: aggregated data and profiling

The above-described technological scenarios implicate possible risks for the data subject but there exist other complications related to the fact that it's not always obvious to actually identify dangerous circumstances in terms of data protection. Technologies and data types have changed over the years, for this reason is difficult to establish what kind of information can lead to personal data.
For instance, the European Court of Justice (ECJ) has declared that the Internet Protocol (IP) address can be considered as personal data since even if it refers to an electronic device connected to a network, it can be used to univocally identify an individual if is available information about who is using that machine at a certain time [8]; moreover, is possible to extract insights about someone religion beliefs, or more intimate and sensitive information, linking together his GPS data and the knowledge about what that coordinates correspond to. Considering some information that, individually taken into account would be classified as non-personal data, it is difficult to guarantee that the same data can't eventually lead to personal and private information disclosure if linked with data from other sources. For this reason, it's extremely complex to define exactly which kind of data needs special care, therefore it is also complicated to assess the type and the measure of risks to which individuals are exposed.
In this context, considering the variety, the complexity and the threats to which the data subjects are vulnerable, it is fundamental examining the problem from a semantic point of view rather than from a syntactic one: in a situation that requires to process personal data, the focal point is to limit the possibilities that the data subject is profiled. Prevent individual re-identification it's important as the final step in a data protection context, but it is important also to take into account, with a holistic approach, all the processes and assessments concerning the risks and impact on the data subject. Profiling [9] definition has a very broad flavour since it can include any form of automated personal data processing to evaluate individual

sphere: analyzing his work performance, his health or economical status, his passions or interests etcetera; what is clear is the deeply invasive nature of profiling. A simple classification of individuals based on well-known features like age, sex or height is not necessarily considered profiling. We can identify the important aspects as the purpose and the manner in which data processing is carried out, ensuring that it won't have any negative impact.

2.3 Data protection regulations

All the scenarios and problems presented in this chapter aim to open a debate about the necessity of finding a balance between strategical and economical benefits that companies can obtain exploiting data insights and discrimination risks and personal rights violation that can affect individuals. For many years, for some technologists and legal scholars, it was clear that legislative efforts were necessary to clarify all the uncertainties and gey zones, defining rights and duties for all the entities in the scenario.

2.3.1 General Data Protection Regulation (GDPR)

From Convention 108 in 1985 [10] to the GDPR entry into force in 2018, the European Union has set the golden standard regarding data protection regulations. In the last few years, there was an ongoing process of implementation of the principles and rules contained in GDPR: it's not a change to the previous data protection regulations but it's the final step of a harmonization process.
To deeply understand the reasoning and choices behind this regulation it's really important to underline a fundamental idea: according to the GDPR regulators' intentions, privacy and data protection are two correlated but distinct notions. Data protection involves not only information regarding individuals' private lives but includes also publicly available information like the fist name in a public record. Privacy definition takes into consideration the sphere of individual private life and self-determination, while data protection cares about the use of personal information, both public or private, that can be used to make decisions on a certain individual influencing his life politically, socially, economically. To better clarify this concept GDPR distinguishes personal data, considering also the sensitive one, from non-personal data. Personal data refers to any information associated with an

identified or identifiable natural person that can be singled out exploiting data related directly to him, but also conducting further researches using information that are just indirectly related to him [11]. Moreover, sensitive data require special protection when processed since corresponds to very intimate and discriminatory data as racial or ethnic origin, political opinions, religious or philosophical beliefs, health data, genetic and biometric data and data relating to criminal convictions and offences. Finally, the non-personal data notion it's a non-definition that includes all information that is not possible to link in any manner to a natural person; for example, data shared between machines in a smart industry context or level of pollution in a city. GDPR strongly establishes data protection as one of the fundamental human rights maintaining the balancing interests, previously mentioned, as the focal point of all the regulation: article 1 [12] points out the relationship between data protection and the free movement of data, not considering all the rights taken into account as absolute, yet searches for a balance with the competing interests to facilitate the data movement. Data protection enforcement is an instrument to improve the information flow, avoiding barriers creation, protecting the individual's interests and pursuing the fundamental right not to be discriminated against.

2.3.2 GDPR main pillars

Eight main pillars characterize the GDPR and more in general the European approach to data protection:

Risk assessment and management

GDPR focuses specifically on this concept as a consequence of the shift from the traditional idea of data security familiar the computer science field to the data protection one; to protect from data breaches it's not enough but it's necessary to reduce the negative impacts of the data on individuals. What's new to the previous directives is that GDPR explicitly defines the procedures that need to be adopted in terms of risk management during data processing that can be summarized in two phases:

- the preliminary analysis consists of defining the data flow, what kind of information and for which purposes data are processed, carefully listing all the actors involved.

- the data strategy and management report is a detailed and well-structured document containing a systematic explanation of the envisaged processing operations and purposes, specifying the activities and the risks nature for the data subject. The document must underline the attention to the data processing necessity and proportionality principles related to the chosen purposes; moreover must be carried out the analysis of the rights and freedoms risks for the data subject, and all the countermeasures to reduce them.

Controller and processor

During data lifespan, different kind of persons and entities may interact with personal information; for this reason, GDPR defines the controller and processor figures. The Controller is a natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the personal data processing, being compliant with the ones defined in GDPR.
The processor is a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller. The distinction between the two figures is defined by a de-facto qualification: regardless of contracts or agreements that two entities can have arranged, the controller and the processor positions and different responsibilities are determined by the actual role that they have in the data processing activity.

Right to be informed, right of access and right to erasure

The data subject must be informed if any kind of data processing activities is carried out, specifying exactly which data is collected and used. Moreover, he can access his personal data at any moment and he can object to all of the data processing phases, forcing the controller or processor to erase information if data are considered not relevant nor accurate.

Legal grounds

When deciding to carry out a data processing activity that includes personal data, it's mandatory to identify the specific legal ground for that particular operation as a preliminary step. The first legal ground taken into account is consent, considered one of the best expression of individual self-determination; to be admissible when collecting personal data it needs

to meet some requirements, as specified in article 7 [13]. The other legal grounds are all based on the idea of necessity: personal data needed for performing the specified contract, compliance with legal obligations, data processing necessary to safeguard the vital interests of the data subject or another person, public interest.

Purpose

Every data processing activity can be carried out only specifying an explicit and legitimate purpose that must be strictly observed until the end of the process. The controller can change the purpose while collecting data if the newly selected one is compliant with the original one.

Data minimization

Personal data can be used and collected only when strictly necessary, essential, relevant and not excessive concerning the purpose of data processing since regards fundamental rights sphere.

Storage limitation

Another important aspect in terms of data quality is the notion of storage limitation. The controller or processor has to collect information and use it for a period that is consistent with the purpose specified at the moment of the data gathering.

Accountability

It is a major plus if we consider the previous data protection regulations. The accountability notion extends the liability one; it is a kind of liability evolution since GDPR states that the controller and processor must be able to demonstrate to be compliant with the regulation during each data processing step and not only if harms or damages occur. The goal is to sensitize data processing entities to carefully manage personal information specifying, during the design phase, the legal ground, the purpose and the envisaged risks for data subjects.

2.3.3 California Consumer Privacy Act (CCPA)

The CCPA [2] is a state statute signed into law in 2018. It represents one of the most important efforts, outside the European Union, to ensure data protection and empower consumers to better control their data. The CCPA applies to for-profit companies that process personal California citizens' information, have business interests in the State of California and are considered a large organization according to financial or logistic viewpoints. Its main principles revolve around simple but effective adjustments in the relationship between companies and consumers:

- Right to know: the majority of consumers is not aware that their data has been collected; for this reason, companies must clearly underline what data they're willing to collect, how and for which purposes it will be processed.
- Right to object on data sale: companies can sell personal information but they are obliged to communicate it to the data subject, which can object and deny it. Moreover, the sell notion is considered broadly and include also the data disclosure with third parties.
- Right to access: the data subject can request information about the specific personal data that has been collected; in particular the company is forced to provide information also about the source from where data has been processed and the third parties with whom the company has shared the data if requested.
- Right to delete: the data subject can request the company to cancel data collected on him, with some exception for legal information.
- To not be discriminated: companies cannot discriminate against data subjects that exercised their rights under the Act, they must provide to customers the same goods or services. This approach can present some grey zones, as at the current state, it is difficult to assess if a business is discriminating or just providing different service levels.

2.4 Data protection: thesis goals

In this chapter, we describe some of the risks and technological challenges regarding a data protection context. The Big Data era overwhelmed society

with data characterized by features like volume, velocity and variability that shaped our world perception. The majority of technological instruments essential to our everyday life are part of a complex and interconnected network in which Big Data can be seen as the nervous system. To gather insights and value from data is by now a fundamental, strategical and vital aspect of companies business, as it can't be considered a commodity anymore. For this reason, data are recognised as an important business asset to drive decisions or as trade goods in the data market, depriving the data subject of its auto-determination right: the individual has become the product himself, enforcing almost no control over its personal information, deprived of the possibility to decide which data can be sold or shared with third parties, to modify or erase it if no longer relevant. During the last years, regulations as GDPR and CCPA responded to the increase in consumers' demand for data protection enforcing a power rebalancing between data gathers and data subjects. As things stand, from a technological perspective there are no technical standards to handle the data processing in all its phases nor exist spread instruments to shape the data subject consent nor to regulate the data market and its intrinsic value. As first fundamental step, this thesis work will focus on a technical P-PPA solution to extract useful information from data while preserving the data subject's privacy, a suitable trade-off preserving and retrieving strategical data patterns without compromise and expose individual data privacy.

2.4.1 Personal-privacy preserving analytics

The privacy-preserving mechanism notion involves a large number of privacy correlated concepts that can be summarized by combining protection methods, models and metrics. The schema represented in Figure 2.1 clearly explains this categorization.

Privacy-preserving protection methods can be split into:

- Cryptographic: mostly utilizes homomorphic encryption [15] techniques, performing queries on divided encrypted data and returning the decrypted aggregated results. Although they can guarantee optimal performance on the privacy side, computationally are far from efficient employment.

- Non-perturbative: these methods usually refer to structured data format and imply generalization and suppression techniques applied to sensitive

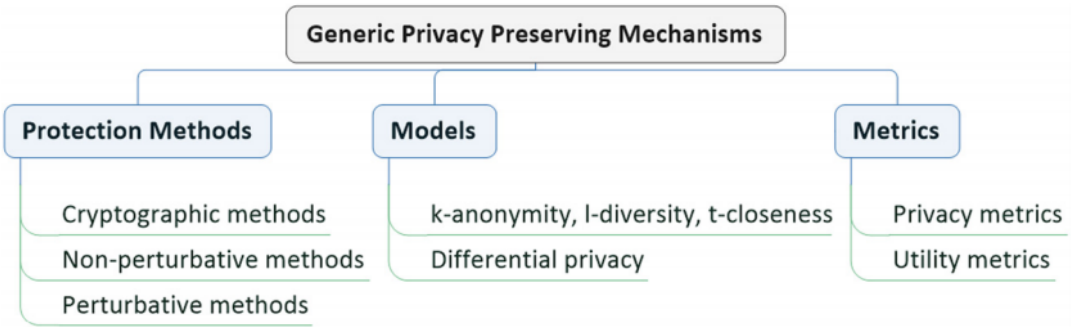


Figure 2.1: Privacy preserving mechanism taxonomy [14].

information of table records.

- Perturbative: the basic idea is to apply to data some form of perturbation noise. The output can be represented by the straight perturbed data or a perturbed query result performed on original data that remains protected.

Taking into account privacy-preserving models, during the last two decades a large number of security properties were developed and broadly embraced, as k-anonymity and differential privacy that will be further investigated in the next chapter.
Summarizing and considering a broader perspective, a privacy-preserving mechanism enforces privacy protection by employing a protection method based on a protection model and evaluating the performance considering utility and privacy metrics according to the previously cited trade-off paradigm. During this thesis, we will present a modular P-PPA framework that according to the input data provides perturbative and non-perturbative methods based on k-anonymity and differential privacy assessing both utility and privacy perspectives.

Chapter 3

Privacy-preserving models

3.1 Privacy-preserving models

The large data type variability produced by every kind of data source implies that data are generated almost at any granularity degree, from individual to aggregate information; in this thesis, we are interested in providing a defined privacy level when individual personal data are handled. Each of these individual records, defined as microdata, can contain:

- personally identifiable attributes (PIAs): this definition can slightly vary according to which regulation or institution we refer to. In this thesis, we will consider the one related to the National Institute of Standards and Technology (NIST) definition of personally identifiable information: "... any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records and any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information." [16].

- quasi-identifier attributes (QIAs): "... is a set of attributes that, in combination, can be linked with external information to reidentify (or reduce uncertainty about) all or some of the respondents to whom the information refers." [17].

- sensitive attributes (SAs): for this definition, we will consider an attribute related to intimate and discriminatory data like racial or ethnic origin, the salary, political opinions, religious or philosophical beliefs, health, genetic and biometric data or linked to criminal convictions and offences; this sensitive attribute notion is collinear with the one from GDPR, cited in the subsection 2.3.1.

To prevent individual identification or re-identification in microdata processing activities, during the last 20 years several privacy-preserving models were developed and exploited implementing technical anonymization solution. In this chapter, we will further present theoretical properties and algorithms regarding the k-anonymity [18] and Differential privacy [19] models.

3.2 k-anonymity anonymization model

Considering data in tabular format for simplicity, a released dataset containing information at the individual level is compliant with the k-anonymity property if each data subject can't be distinguished from at least $k-1$ individuals further present in the released dataset, taking into account all the possible QIA combinations.
One of the techniques used to make a dataset k-anonymous is the generalization one. Let's have a focus on it: generalize for instance an attribute value means to summarize that information with something more general, something at a higher level of the generalization hierarchy. Let's consider a geographical attribute value like *Turin* city: a possible generalization could be obtained substituting this value with the *Italy* one.
Another technique is represented by the suppression one: keeping the same attribute example, it could be possible to achieve k-anonymity deleting the value *Turin* or mapping it to a defined character to express the suppression.

3.2.1 Generalization and suppression

To further explain the generalization theory, it is necessary to formalize an extended domain concept for attribute values: from now on we will refer to *Dom* as the set containing all the attribute values plus its generalization of each level. For instance, we consider the *race* attribute whose domain is

composed of the {asian, white, black} set and the generalization set {people}; both of these two sets are contained in Dom. If generalization sets on more levels were examined, all the attributes sets with their generalizations for each level would be present in Dom.
Moreover, these following conditions will make deterministic the subsequent definitions [20]:

1. For each domain D_i , the set of domains generalization is totally ordered and, therefore, each D_i has at most one direct generalization domain D_j; every level in the hierarchy has at most one generalization, and so on.

2. All values in each domain can always be generalized to a single value.

This concept of order is fundamental to introduce the following definitions thanks to which is possible to describe a wide overview of the generalization theory [20]:

- The mapping between attribute values and their generalization is expressed with the notation $\leq$D. Given two domains D_i and $D_j \in Dom$, $D_i \leq$D D_j states that values in domain D_j are generalizations of values in D_i. Following the previous example:

$$\{asian, white, black\} \leq D \{people\}$$

- In the same way, we can define a value generalization relationship with the notation $\leq$V; taking an element $v_i \in D_i$ and $v_j \in D_j$, $v_i \leq$V v_j states that the attribute value v_i is directly generalized by v_j, immediately above on the hierarchy.

Summing up the above-mentioned hierarchy concepts, is possible to distinguish two structures referring respectively to domain-domain and value-value relationship: for a domain $D \in Dom$, the domain generalization hierarchy is called DGH_D while the value generalization hierarchy is denoted as VGH_D, whose structure is a tree with the primary values in the bottom leaves and the most generalizing one placed in the root, as shown in Figure 3.1 and 3.2.

To work with real-world datasets, we have to extend the previously presented concepts considering not only a domain D, but a domain tuple $DT = \{D_1, \ldots, D_n\}$ such that $D_i \in Dom, i = 1, ..., n$ and its corresponding

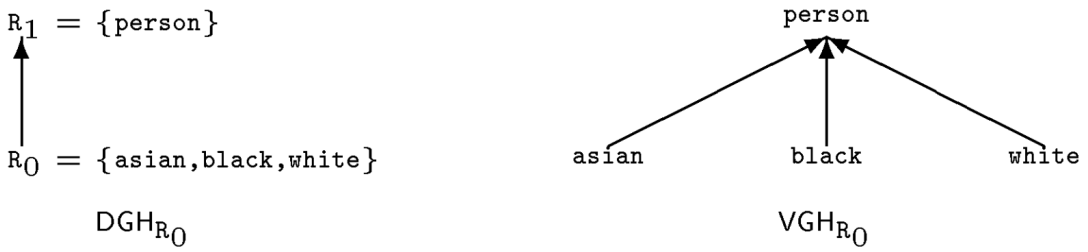


Figure 3.1: Domain generalization hierarchy and value generalization hierarchy for person attribute set [20].

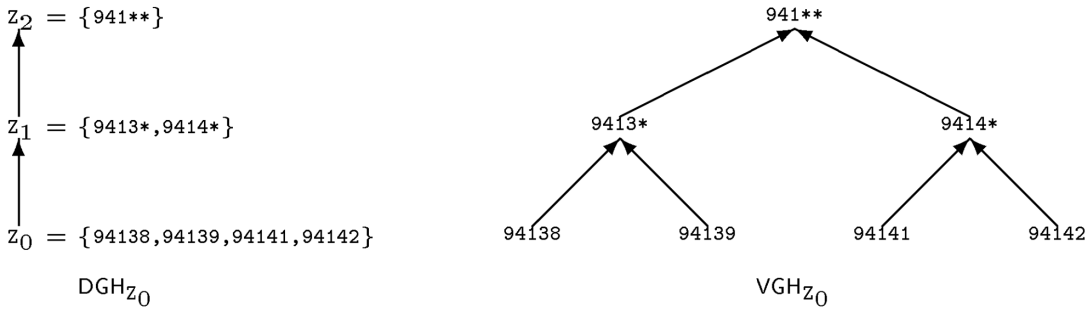


Figure 3.2: Domain generalization hierarchy and value generalization hierarchy for US postal codes attribute set [20].

domain generalization hierarchy is $DGH_{DT} = DGH_{D1} \times ... \times DGH_{Dn}$ where the Cartesian product is ordered through a coordinate-wise method. The derived lattice has DT as the minimal element and the tuple composed of the top of each $DGH_{Di}, i = 1, ..., n$ as the maximal one. Moving from the bottom to the top of the structure, we can define various generalization strategies thanks to which different sets of QIAs can be generalized, considering every QIA domain and generalizations. For instance, considering US postal codes with a generalization level equal to two and the race attribute, there exist three possible paths, as displayed in Fig.3.3.

Each node of the graph represents a possible outcome for generalizing the original table, with a different level of generalization.
Another technique used in literature to obtain a k-anonymous dataset is

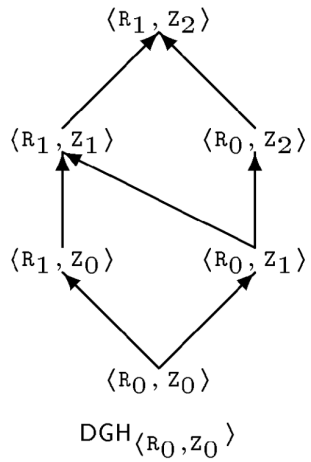


Figure 3.3: Generalization strategies graph for US postal codes and race, two generalization levels [20].

suppression, that can be considered a one level generalization special case substituting each column value with a chosen symbols. For instance, taking into account a table it is possible to suppress an entire attribute column reducing the released amount of information.
Generalization and suppression can be applied at the same time to balance the semantic information loss from the former and the omitted information from the latter. As a consequence, according to the examined information, various metrics could be used as the minimum absolute distance, the minimum relative distance, the maximum distribution and the minimum suppression [21].
Moreover, generalization and suppression techniques can be applied simultaneously to different granularities such as the cell, the tuple or the attribute levels: we refer to the attribute level if a technique is applied to an entire column; climbing down the granularity degree, a tuple level represents a single row and finally, some operation on a single cell takes effect considering the interpolation value from one column and one row. In some cases apply contemporary generalization and suppression has no meaning for structural constraints: generalization can't be applied at the cell level while simultaneously suppress at the tuple or attribute level. The more these techniques are applied at higher granularity the more details the output data will have, consequently increasing the solution complexity; it is not a coincidence that many optimal solution algorithms often exploits both techniques.

3.2.2 Optimal solution algorithms

During the years many pieces of research focused their work on exploiting k-anonymity property constraints trying to balance between the information released and its practical utility.
The k-anonymity problem using generalization ad suppression is computational NP-Hard [22]. k-anonymity-based algorithms found in the literature, include different features and approaches corresponding to diverse computational complexity. Optimal-solution algorithms need exponential computational time in the number QIAs, in fact, when their number is maintained small the k-anonymity problem is still considered tractable.
In the remaining part of this paragraph will be further presented four optimal-solution algorithms.

Samarati's algorithm [20]

Considering a set of QIAs, many different generalization strategy paths can be selected from the bottom to the top of the hierarchy, as explained in section 3.2.1. For each path exists a locally minimal generalization: once every path is covered and each minimal generalization found, one of it must be the path that guarantees the optimal solution to the k-anonymization problem.
Going up in the hierarchy path the number of tuples to be suppressed to guarantee the k-anonymity property decreases; this principle is exploited to tackle the unfeasibility of the problem. The algorithm allows setting a *maxdel* threshold, representing the maximum number of deleted tuples and a height h on the hierarchy tree paths: if no global solution is found for a certain *maxdel* and h, no solution exists for a lower h value.
The algorithm adopts a binary search procedure on the domain generalization hierarchy on each QIA domain, until it reaches the lowest height for which a solution is found. In the presence of multiple k-anonymous nodes at the lowest height, random or preference criteria is used to select one. To reduce the computational effort and avoid to compute explicitly each generalized table, a distance-vector matrix is exploited.

Optimal Lattice Anonymization (OLA) algorithm [23]

It's conceptually similar to Samarati's algorithm in the procedure pipeline, but doesn't consider the height a valid metric for choosing a k-anonymous node on the generalizations lattice: for instance, a k-anonymous node at height three on a path can have a higher information loss with respect to another one on a distinct path but at a higher level, since the height doesn't take into account the attribute generalization depth.
The OLA algorithm searches for the k-anonymous nodes at the lowest height for each QIA domain but keeps in the solution pool also the higher k-anonymous nodes since, for the above-explained reasons, higher height doesn't necessary means higher loss.

Byardo-Agrawal's algorithm [24]

Byardo-Agrawal's algorithm establishes itself as a more efficient algorithm that exploits a value ordering between value attributes for each domain but also through different attribute domains. Each domain is divided into intervals such that every possible value within it is assigned to an interval I and each value in a given interval I precedes the ones belonging to intervals following I. Considering *race* and *US postal code* attributes, the Figure 3.4 shows how it could be applied a first lexical ordering between attributes and a second one internally, obtaining an ordered index-linked to each interval.

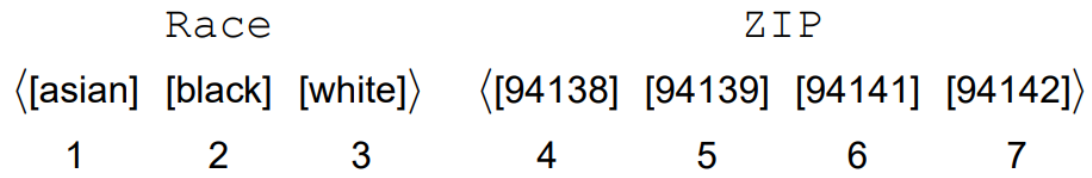


Figure 3.4: Ordering attributes example [24].

A generalization is given by the union of the individual index values for each attribute, moreover, the least value in an attribute domain can be omitted because it will appear in other generalizations in the same domain. Considering both the attributes and indexes from the Figure 3.4, it follows an example:

$$Race : \{1\} = \{asian \vee black \vee white\}$$
$$US\ postal\ code : \{4, 6\} = \{[94138 \vee 94139], [94141 \vee 94142]\}$$
$$Race : \{\} = \{1\} = \{asian \vee black \vee white\}$$
$$US\ postal\ code : \{4\} = \{94138 \vee 94139 \vee 94141 \vee 94142\}$$

Taking into account an interval I a set enumeration tree is built starting from the empty root, adding at each node sets that are composed by appending a single element $i \in I$ interval to the node such that i must follow every single element already present in the node. The Figure 3.5 represents a tree building example starting from $I = \{1,2,3\}$.

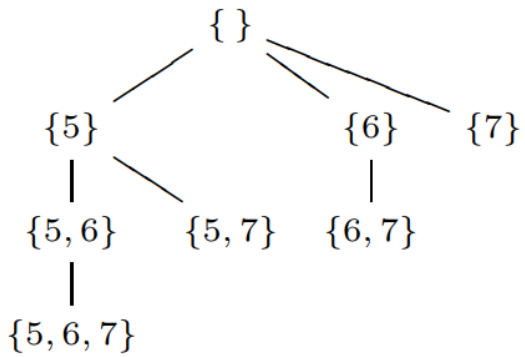


Figure 3.5: Tree building example [24].

Once built the tree, the algorithm visits it in a standard transversal manner computing the information loss for each node and evaluating all the possible solution for the k-anonymity problem. Techniques like pruning or early stop mechanism improve time computation, preventing however to obtain an optimal solution.

Incognito algorithm [25]

Another algorithm that tries to lighten the computational burden for the k-anonymity problem is Incognito. It exploits the simple but pivotal concept concerning the impossibility of an attribute generalization to be compliant with the k-anonymity property if a less general transformation for the same attribute led to a non-k-anonymous output table.

Considering a set of QIA, the algorithm performs $|QI|$ iterations checking k-anonymity condition on all generalizations for all attributes starting taking into consideration every single one, until the last iteration in which is examined a tuple of $|QI|$ attributes. At the end of each step all the generalizations that produce a non-k-anonymous output are eliminated, therefore excluding also all the direct ones; in this way, during every new iteration, just the suitable ones for the previous step are taken into account. The algorithm terminates when the optimal solution is found.

3.2.3 Heuristic algorithms

Considering the k-anonymity problem, as previously presented, some optimal algorithms tries different strategies to optimize the computational pipeline, but the resulting effort could represent anyway an insurmountable obstacle in certain situations and environments.
For this reason, during the years researchers focus their studies on heuristic algorithms to tackle the k-anonymity problem. Heuristic algorithms sacrifice accuracy and completeness to reduce the computational time, in fact, the goal is to find a reasonable solution that will unlikely be the optimal one for problems that require an intractable amount of time.
We descibe two of the most famous and implemented heuristic algorithms for the k-anonymity problem.

Datafly algorithm [26]

After a set of QIAs have been defined, the algorithm first step consists of populating a list of frequencies for the tuple projection of the QIA set on the starting data table. From this list, the algorithm chooses the attribute with the most number of distinct values and generalizes it. If the procedure has led to a k-anonymous table the algorithm stops since a solution is found; on the contrary, the iteration is repeated until in the frequency list is present a number greater than the k value for each of the different tuple occurrences. Therefore the algorithm tries to reach a k-anonymous table generalizing first the attributes that in order correspond to the maximum variance in the frequency list and for this reason, there could be cases in which it tends to overgeneralize; moreover, Datafly doesn't use an information loss metric.

Mondrian algorithm [27]

It is a greedy algorithm used to anonymize table dataset. One of its principal characteristics is the multidimensional approach: it refers to the ability to recoding multiple attributes per row. The Figure 3.6 better explains the difference in taking decision considering one attribute at a time instead of two of them.

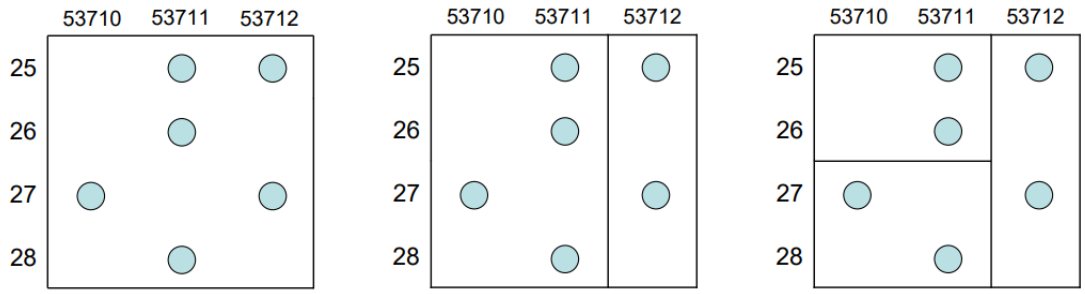


Figure 3.6: Different anonymizations for patiens' age/zip code plane. Left:patients and no splitting; Center: zip code dimension split; Right: zip code and patient' age two-dimension split [27].

The algorithm target consists of continuously split the current data region into partitions compliant with the k parameter. First, it chooses the dimension on which perform the partitioning considering the attribute with the widest range of different values, normalized with respect to the other attribute ranges; secondly, the median value is selected to actually perform the split. Other heuristics can be considered, for instance, a crescent or descent ordering instead of a data distribution approach: for attribute data having a very skewed distribution, a frequency approach is not suggested as this could generate an outliers problem.
After that, the k-anonymity property is checked for the two resulting data partitions and according to the positive or negative outcome, the examined partition is stored in a done-list or is re-inserted in the working data region. This process continues recursively until the region is empty, as displayed in Figure 3.7.

If the algorithm can't rely on user-defined generalization hierarchies it shows better results with numerical attributes, even though can be successfully used also on categorical ones exploiting encoding functions.

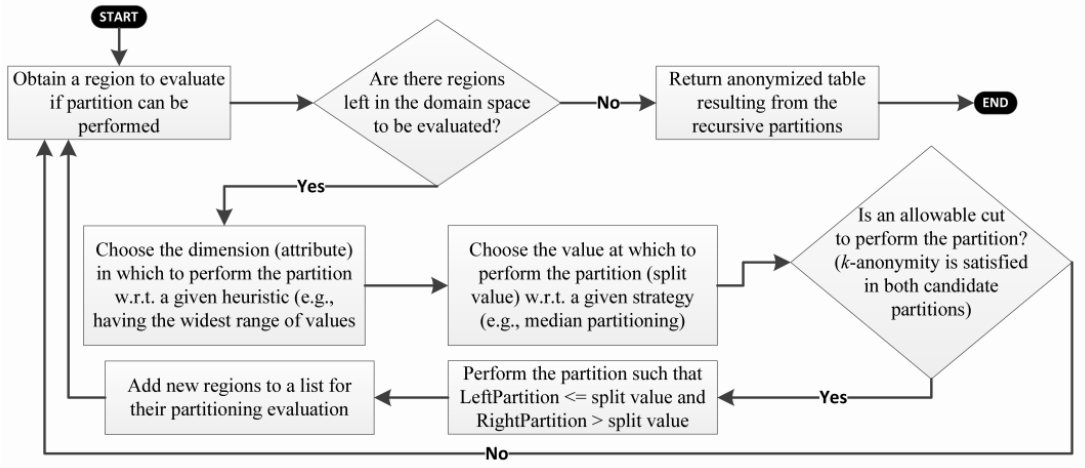


Figure 3.7: Mondrian algorithm main steps [28].

3.2.4 Drawbacks and variants: l-diversity and t-closeness

Real use cases have proven that there exist many different techniques and attacks that permit to gather information from released k-anonymous private data: they can be classified according to attack requirements, danger, amount of extracted knowledge and approach variability. For instance, the *unsorted matching attack* can be performed if the released data hadn't been shuffled since in real-world scenarios the order of the rows from others related dataset can be inferred and the information linked. Moreover, a wrong QIAs selection set can lead to the *complementary release attack* if subsequent releases contain QIAs subsets; considering that many datasets are composed of dynamic data, the tuple addition or removing could be exploited for a *temporal attack* [29].
Two of the major threats are represented by the *homogeneity attack* and the *background knowledge attack*: the former consists in gathering sensitive information, for instance the value of a sensitive attribute, if this value is equal in a group or partition of k elements even if it is part of a k-anonymous table; if is known that a data subject belongs to that particular group, it is consequently possible to associate that sensitive attribute value to that individual. The latter consists of the exploiting of further information that the attacker can use to make reasoning trying to narrow the possible range of values for a sensitive attribute, also utilising information deriving from other QIAs. For example, knowing that Asian people suffer less from some

kind of diseases, an attacker can suppose some illness more probable than others. Obviously the more the range of possible values is numerous, the more difficult would be exploited background knowledge; in this sense, these two attacks are linked together.
To counter these threats l-diversity [30] and t-closeness [31] were studied as k-anonymity variants.

l-diversity

The key principle is related to the concept that attributes in each group have to be *well represented* avoiding, in this way, some information disclosure. In literature are illustrated three different flavours for this principle:

1. Distinct l-diversity: for each equivalent group[1] and sensitive attributes there must be al least l diverse values. This defines a sort of lower bound since a possible counterattack could be when a sensitive value is *too well represented*, and an attacker could conclude that that particular value is predominant in the equivalent class.

2. Entropy l-diversity: considering an equivalence group G and a sensitive attribute c, the entropy is defined:

$$H(G) = -\sum_{c \in C} p(G,c) log p(G,c)$$

   where $p(G,c)$ is the probability of a record in group G to have the c value and l is the l-diversity parameter. If for each group $H(G) \geq log(l)$, the data has Entropy l-diversity property.

3. Recursive (c-l)-diversity: it is a compromise between the two previous definitions ensuring that the most common and rare values are not too common and rare, in each equivalent class.

l-diversity mitigates a lot of k-anonymity vulnerabilities, but there exits some weak situations: for instance if the overall dataset distribution is skewed, the attacker can infer some knowledge regardless of the l-diversity property, named as the *skewness attack*. Moreover, also the *similarity attack* can be still performed on an l-diverse dataset: it refers to a circumstance in

[1] An equivalent class or group is a set of records indistinguishable from each others.

which sensitive attributes for each group are diverse but semantically similar like lung cancer, liver cancer and stomach cancer that are all diverse types of cancer.

t-closeness

Like l-diversity, also t-closeness property can be considered as a k-anonymity improvement. This technique is a trade-off between information disclosure and data precision since tries to reduce the data granularity. If a sensitive attribute distribution in an equivalent class differs from its distribution related to the whole dataset at most by a t value, the data are compliant with the t-closeness property.
In a situation in which also l-diversity property fails to perform valid protection against information disclosure like with skew data or sensitive attribute values semantically similar, data can be manipulated to be compliant with the t-closeness property; an attacker could not rely anymore on rare indicators that provide more information comparing to a common one.

3.3 Differential privacy

Differential privacy [32] can be considered as a set of mathematical definitions that assures a privacy standard in a specified range of applications. The formal differential privacy definition has slightly changed over the years but the main concept remains the same: considering a database on which are willing to be performed statistical queries like median, variance or in general some function to extract overall statistics about data, the applied function is differential privacy compliant if the presence or absence of a singular row in the database doesn't change the amount of gathered information. In other words, if some personal data is stored in a dataset that will be queried with differentially private functions, the data subject to which that information is related can be sure that the entity that is performing statistical requests will not learn any individual information related to him. Differential privacy guarantees can be applied to many different objects like functions, algorithms and processes and hold for every data subject whose data is present in the considered datasets.
In summary, what differential privacy can guarantee is that if the data subject's information is present or not in some database that will be queried in a differential privacy flavour, it doesn't change the possibility for the

data subject to be harmed for any information about its data. Specified that, we need to pay attention to an important detail: differential privacy mechanisms can protect private information, but if something private for the data subject it's actually a generally shared information in that dataset, differential privacy constraints still hold but the private information is revealed.
Differential privacy is formally defined as following [33]:

"Let $\epsilon > 0$. Define a randomized function M[2] to be (ϵ)-differentially private if for all neighboring input datasets $D1$ and $D2$ differing on at most one element, and $\forall S \subseteq \text{Range}(M)$, we have:"

$$\frac{Pr[M(D1) \in S]}{Pr[M(D2) \in S]} \leq e^{\epsilon}$$

"where the probability is taken over the coin tosses of M".
This definition states that the more the ϵ value is small, the more noise will be introduced and, for this reason, more privacy will be obtained. In this sense, the ϵ parameter can be seen as the control knob that represents the trade-off between privacy and data utility.
The function sensitivity notion is a fundamental concept introduced to practically measure the amount of noise needed to grant a certain privacy level [33];
"For $f : D \to R_k$, the sensitivity of f is:

$$\Delta f = \max_{D1,D2} \|f(D_1) - f(D_2)\|_1$$

for all $D1, D2$ differing in at most one element".
Considering a function, its sensitivity is defined as the maximum change in its result applied first on *D1* and then on *D2*. For instance, for a function that measures the dataset row count, the sensitivity is equal to one since if any row is removed, the query result will differ at most by one. In situations in which more complex operations or functions are considered, it can be difficult to exactly assess the sensitivity value for each different dataset to which the function can be applied; nevertheless in practical cases, this concept is often employed in common statistical operation.

[2] A randomized function is an algorithm whose result derives from imposed randomness.

Differential privacy constraints can be enforced considering various situations, as for instance, according to the way it is possible to access and query a dataset, there exists two applicability differential-privacy context [33], non-interactive and interactive one:

- non-interactive: in this scenario, all the received queries are processed all in one, the database is perturbated using differential private mechanisms and after that, the requested statistics or the entire sanitized database is returned. After this process, the database can no longer be queried, no additional statistical information is gathered from it.
- interactive: in this database access methodology, a curator, an entity between the dataset and the queriers, provides to perform a perturbation method to the queries, the statistical response or both since the database can be interactively queried many times. The information stored in the database remains protected since the queriers interact with it only through the curator interface.

In some situation, a non-interactive mechanism can outperform interactive ones in terms of the tradeoff between information utility and data subjects privacy. Despite this, as proven by Cynthia Dwork [19], non-interactive techniques have structural limitations since there could be always found a low-sensitive function that can't be applied on the considered dataset without breaking differential-privacy constraints. Relying on non-interactive solutions to gather statistics from a database maintaining private any individual information means to focus privacy efforts on some class of functions sacrificing others.

3.3.1 Mechanisms

In literature can be found various differentially private mechanisms thanks to which is possible to apply a defined amount of noise to the query or the response or the data itself, as the randomized response and the Laplace mechanism.

Randomized response [34]

One of the most famous and exploited non-interactive mechanism corresponds to randomized response: a randomizing algorithm is used to perturb every single response that together with others is used to perform some

defined statistics. The most classical example is represented by the coin toss algorithm for a situation in which the sensitive information is represented by an answer to a survey question that can have as the outcome *yes* or *no*: if a coin-tossing gives head, as a result the real response is released, on the contrary, a new coin-tossing is performed after which, according to the result, the response becomes simply *yes* or *no*. In this way, the real answer is perturbed and an attacker can't learn anything new about individual responses, that are anyway "randomized towards the truth".

Laplace mechanism [34]

This perturbation mechanism is very intuitive, often used in interactive database accesses. It exploits the Laplace Distribution, centred at 0 with $scale = \frac{\Delta f}{\epsilon}$, whose definition refers to the following distribution probability density function [32].
"Laplace distribution is characterized by location θ(any real number) and scale λ(has to be greater than 0) parameters with the following probability density function":

$$Lap(x|\theta, \lambda) = \frac{1}{2\lambda} \exp -\frac{|x - \theta|}{\lambda}$$

Every query or response, according to the privacy strategy, is perturbed adding noise sampled from the Laplace distribution [32].
"Given any function $f : D_n \to R^k$, the Laplace mechanism is defined as:

$$M_L(x, f(.), \epsilon) = f(x) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d. random variables from the $Lap(\Delta f|\epsilon)$".
Other mechanisms were studied and often exploited to assure differential privacy property, as the exponential mechanism [35] and the median mechanism [36].

3.3.2 Properties

Differential privacy is characterized by some properties that are fundamental to practically implement algorithms and instruments that are compliant with its constraints, putting together several differentially private simple modules into a comprehensive framework that preserves privacy guarantees:

- Quantification of privacy loss [33]: as previously formally stated, differential privacy is a strong mathematical concept that permits to quantify the trade-off choosing between sacrificing accuracy adding more noise preserving even more privacy, or to grant less privacy and return more accurate statistical information. For this reason, differential privacy is not a binary privacy concept but can suit situations e requirements.

- Composition [33]: if the joint distribution of t different differentially privacy outputs with uncorrelated randomized mechanisms are considered, the result will be $\epsilon * t$-differentially private. This concept represents a practical limitation, since even if a low-sensitivity function is utilized to query a dataset returning an output differentially privacy compliant, the more the same query is performed the more an external observer can infer private information from data. For this reason, a *privacy budget* idea was introduced: since every new query contributes to a certain privacy loss, in practical differential privacy implementations a maximum privacy loss, for instance for the same querier, must be defined.

- Group Privacy [32]: differential privacy definition refers to constraints considering datasets that differ from one row, but it can be successfully extended to c rows. In fact, in group privacy, the privacy loss is $\frac{\epsilon}{c}$ with a total loss of ϵ considering the entire group of c rows.

- Closure under post-processing [32]: this property states that no one, without any additional information about a private dataset, can use the output of a differentially private algorithm to make it less private. "That is, a data analyst cannot increase privacy loss, either under the formal definition or even in any intuitive sense, simply by sitting in a corner and thinking about the output of the algorithm, no matter what auxiliary information is available".

Chapter 4

P-PPA framework: focus on design and implementation

As previously mentioned, personal data are considered a strategical resource for companies and entities as a fundamental component for specific business models, based on targeted advertising aiming to maximize profits; moreover, data can be used to tailor a wide range of organizational and strategical company processes. For this reason, it is necessary to increase efforts in the developing and spreading of practical frameworks thanks to which provide adequate and controlled privacy data processes.
There already exist solutions that take care of some aspect revolving around the complex data processing environment but no standard has been adopted, discouraging interested companies to focus on business models that grant privacy necessities. The PIMCity EU-funded project aims to establish itself as the standard thanks to which organizations can build custom-made privacy-preserving solutions. The framework is composed of different modules to cover the majority of data processing situations, some of which are: a module to store personal data called Personal-Data Safe (P-DS), a solution to compute data metrics increasing the users' awareness defined as Personal-Privacy Metrics (P-PM), a practical implementation of the user consent notion called Personal-Consent Manager (P-CM) and finally the P-PPA that provides data anonymization or privacy-preserving statistics.
This chapter will present the structure and the design choices that involve the

practical P-PPA implementation, explaining in detail the instruments employed to synthesise it in the PIMCity context and describing all the modules that provide the analytics gathering in a privacy-controlled environment [37].

4.1 P-PPA project structure overview and data flow analysis

All the P-PPA modules are developed to maximize the modularity and interconnectivity implementing the necessary REST API to gather data and provide results; moreover, the P-PPA framework can handle different data sources and technologies to produce the demanded privacy-preserving answers according to various data buyer necessities. For this reason, the P-PPA Python project presents three different fundamental elements conceptually categorized in the Flask[1] web framework providing the necessary REST APIs, the data management layer that offers PostgreSQL, MongoDB and CSV interfaces and finally the algorithmic module implementing k-anonymity and differential privacy notions; a summary schema is shown in Figure 4.1.

The three above-cited fundamental components will be deeper illustrated and explained further in this chapter.

We now introduce an initial schematic abstraction to illustrate the high-level connections between data sources, data types, and algorithmic-dependent outcomes that characterize the basic context in which we have designed the P-PPA project.

As explained in previous chapters, the current data scenario consists of a wide range of different sources and data types; in Figure 4.2 it is underlined what kind of data and storage format are supported by the P-PPA. Different data storages and respectively technologies are considered, distinguishing between structured and semi-structured data: the structured one is managed in a PostgreSQL[2] database while the semi-structured data is stored in a MongoDB[3] database; moreover the P-PPA module can handle also CSV

[1]Flask website: `https://flask.palletsprojects.com/en/1.1.x/`

[2]PostgreSQL website: `https://www.postgresql.org/`

[3]MongoDB website: `https://www.mongodb.com/`

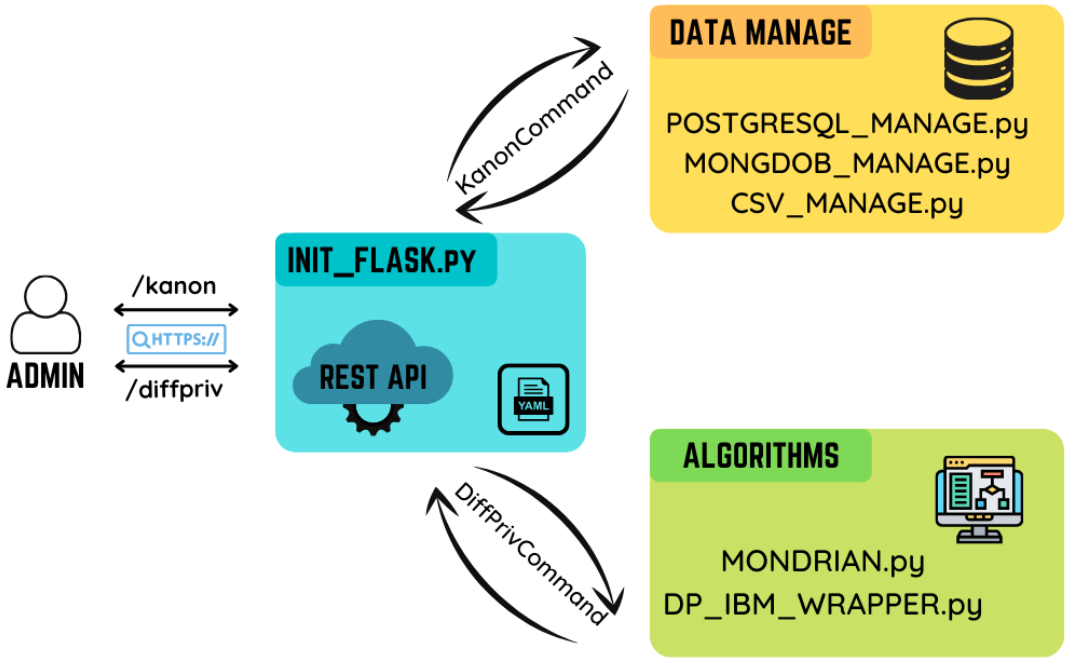


Figure 4.1: PPPA project general overview presenting fundamental modules.

data format, also considered semi-structured.

PostgreSQL database

PostgreSQL is an open-source relational database management system that supports the SQL programming language since it's capable of storing and managing structured data. This kind of data is stored in a table format with keys and attributes, defined by a strict data type hierarchy, order and a defined schema that exactly represents the data type for each sequence of attributes.
These strict constraints make relational datasets useful in situations that require to store fixed data not subject to frequent variations during the time, guaranteeing a range of fundamental properties mentioned with the ACID acronym: atomicity, consistency, isolation, durability. Briefly, every data

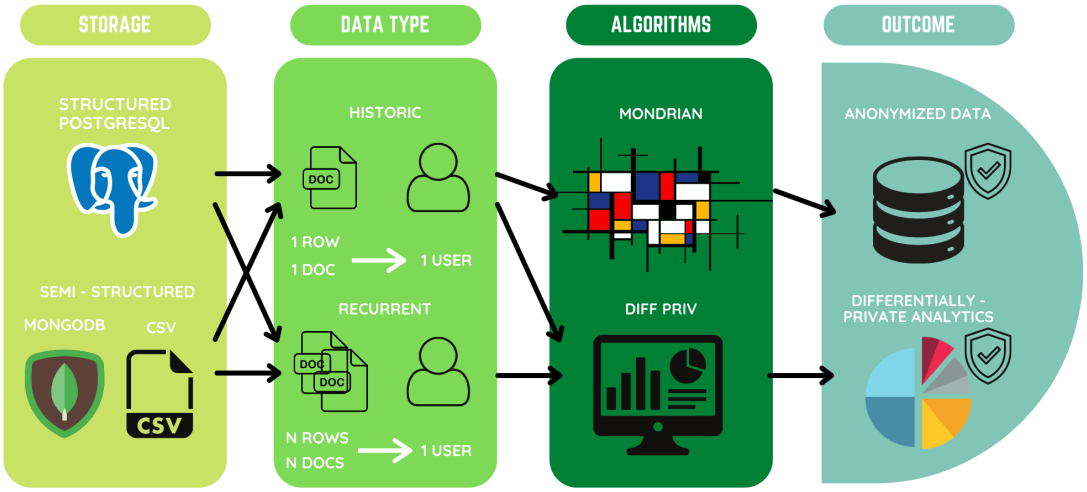


Figure 4.2: PPPA project data flow through chosen privacy algorithms.

change is performed as a single operation and is consistent considering the state before and after a transaction[4] which is independent of the others and persistent once executed even after a system failure.
Be compliant with the ACID properties indicates to prefer a vertical scaling approach over a horizontal[5] one because handling numerous nodes it's computationally in contrast with the above-cited strict properties. Moreover, data must be normalized before being stored in a relational database to avoid information overhead that would mean an additional burden.
The chosen relational database is PostgreSQL for its simplicity, modernity and its statistical analytical approach.

MongoDB database and CSV format

MongoDB is a document-oriented database used for semi-structured data which aren't stored in a relational dataset but presents some structural

[4]A transaction in a SQL database is an individual logical unit of work obtained with one or more SQL instructions;

[5]Vertical scaling approaches prefer to operate with less in number but more powerful machines, vice-versa the horizontal approach.

characteristic as a key-value association, a document concept or a graph structure. Non-relational data management systems don't need a fixed data schema since information can have different formats and data types, useful in a constantly changing data context; moreover, this decrease in constraints has led to different supported operations that are seen as more flexible and faster but with a higher level of query complexity in comparison with the relational databases ones.
These typologies of database were developed with an opposite approach in mind with respect to relational ones, in fact, they were designed to favour horizontal scaling over vertical one, with simpler layouts and more suitability to objected-oriented utilization.
Non-relational databases don't support ACID properties but focus on speed and simplicity, beyond consistency, availability and partition tolerance according to the CAP theorem [38] that states the impossibility in a distributed environment to be simultaneously compliant with more than two of the three previously guarantees.
MongoDB was a natural choice since it's the most widespread non-relational database; it utilises a document format approach that contains the information encoded with XML, YAML, and JSON serializing forms.
Moreover, the P-PPA module supports also CSV data, considered a semi-structured data format.

Data types and algorithmic results

Both structured and semi-structured data can be semantically classified into two types, historic and recurrent. As can be seen in Figure 4.2 we can consider historic data that kind of information characterized by a one-on-one mapping between the number of rows, or documents if we are taking into account data from MongoDB, and the number of people to which that row or document is related. Census data containing age, name, social security number, etcetera, is the perfect historic data example.
On the contrary, if more rows or documents are related to the same individual, we can classify them as recurrent data; they can contains personal information that can be exposed multiple times like visited sites stored in the browsing history or the location positions coordinates associated with the same person. The P-PPA module is agnostic from this point of view since it is designed to handle both data notions.
The data flow continues to the algorithmic part where data is processed

according to the privacy necessities: historic data, independently from the way they are stored, are passed to the k-anonymization module based on the Mondrian algorithm, that provides as output the anonymous data according to the chosen k parameter. In a parallel manner but starting from both historic and recurrent data, the differential privacy module implements the algorithmic concept to output requested statistics, balanced with privacy constraints.

4.2 Flask web framework

We chose Flask web framework among different most famous possibilities, like Django framework[6], because of its simplicity and its expandability capacities, besides the fact that it provides all the basic functions needed for the P-PPA implementation that aims to maintain a simple and modular approach.
Flask is defined as a microframework, since it doesn't come with third parties built-in libraries but allows the user to define and implement components that best fit with the project design choices; it provides a sort of agnostic connecting layer through which it is possible to expand the project functionalities such as a database layer or an authentication module.
The Flask adoption permits to dispatch the required RESTful API set, necessary to make the P-PPA module functionalities to be accessed from others PIMCity components.
With the term REST we intend an architectural framework that defines principles whose goal is to permit the execution of certain operations managing defined representations of resources in a distributed environment:

- Client-server architecture, layer notion: this differentiation in roles is important considering the unbalanced interaction; the server substantially defines functions and data the client can request. The client knows just the API entry point address since all the resources are linked together and discovered when needed, univocally identified by URIs.

- Uniform interface and object representation: all the system components must communicate through uniform conventions, typically HTML, XML

[6]Django website: `https://www.djangoproject.com/`

and JSON are used; resources available for the client must be clearly defined through representations instead of the objects themselves.

- Stateless: no information is maintained on the server-side, every client request must contain all the necessary information to perform it.

Considering a REST scenario, since the objects can be accessed through URLs, the HTTP protocol is largely adopted to perform requests and interactions in general, providing the set of CRUD operations: objects metadata comes in form of HTTP headers while requests and responses correspond to HTTP POST, GET, PUT and DELETE actions.

Table 4.1: Flask implementation classes and parameters.

Class/*Method*	Parameter/Attribute	Description
KanonCommand	Parameters list: token, k, user_choice_index, qi_indexes, whitelist, data_source, csv_path, server_data, host_data, port_data, user_data, pass_data, db_data, table_coll_data.	These class parameters are necessary for authentication, handle the data source/format and provide k-anonymization.
DiffPrivCommand	Parameters list: token, column_indexes, query, epsilon, bounds, axis, dtype, keepdims, bins, weights, density, data_source, csv_path, server_data, host_data, port_data, user_data, pass_data, db_data, table_coll_data.	These class parameters are necessary for authentication, handle the data source/format and provide differentially private statistics.

The project Flask implementation can handle two different requests to respectively perform k-anonymity or differential privacy operations whose resources can be accessed with the defined server address plus */kanon* and */diffpriv* routes. The *KanonCommand* and *DiffPrivCommand* classes can manage and parse different query parameters in their *init* and *get* methods, semantically divided for data retrieving from MongoBD, PostgreSQL and CSV format, or to handle k-anonymity and differential privacy modules; in Table 4.1 we can see parameter lists and a brief description for the two classes. An exception is the *token* parameter, whose presence in the YAML configuration file is checked and required to accept any request from clients. Both *KanonCommand* and *DiffPrivCommand get* methods can return various response types according to the requested operations, nevertheless compliant with JSON format.

KanonCommand class

The *KanonCommand get* method can return the anonymized data in a JSON compliant list format or a *flask.wrappers.Response*. In the former case, the list is composed of another list for each dataframe row, which is formed of a further list of dictionaries whose key is represented by the attribute, and the value by the cell value: for instance a key-value combination as *attr1*:"Jon", *attr2*:"Snow" etcetera. The latter case represents a HTTP code return, according to different situations, as HTTP 401 Bad Request is used if the requested query needs to retrieve data from CSV while the correct path is not specified, HTTP 415 Unsupported Media Type if it is requested an unexpected data source or after some error retrieving data, and HTTP 401 Unauthorized if the query token is not present in the allowed tokens list.

DiffPrivCommand class

The *DiffPrivCommand get* method can return a string that represents a *numpy.simpletype* (for example numpy.float64) or a numpy.advancedtype (for example numpy arrays) for normal statistics as average or sum; moreover, the method supports more complex statistics for the histogram, histogram2d and histogramdd data: the return types correspond to a list of three JSON strings representing the *matrix2d*, *xedge* and *yedge* for the histogram2d case, while it is returned a JSON list of two strings representing the *hist/matrixdd* or the *bins/edges* respectively for the first and third operation. The method can also return the HTTP codes, as its equivalent from the *KanonCommand* class.

4.3 Data management: PostqreSQL, MongoDB and CSV interfaces

The data management layer developed in the P-PPA project provides specific interfaces to connect and gather data from PostgreSQL and MongoDB databases besides a simple CSV retrieving module.
Considering the postgreSQL support, we exploited the psycopg2[7] version

[7]Psycopg2 library website: `https://pypi.org/project/psycopg2/`

2.8.6, the SQLAlchemy[8] version 1.3.20 and the Pandas[9] version 1.1.5 libraries. As we can see in the Table 4.2, the *PostgreSQL* class receives attributes required for the connection to a specified PosrgreSQL database from a defined host and raises a *psycopg*2.*OperationalError* exception if it can't be established. If also the demanded table name is correct, it retrieves the requested data table using a Sqlalchemy engine and transforming it into a pandas dataframe, a standard table data format to which every data object handled from P-PPA project is converted.

Table 4.2: PostgreSQL class with methods and attributes.

Class/*Method*	Parameter/Attribute	Description
PostgreSQL	Parameters list: server_name_connect (str), server_user_name (str), server_user_password (str), db_name_create (str), server_host (str), server_port (str) .	These class parameters are necessary for connecting to the selected PostegreSQL database from the specified host.
retrieve_table: Reads from a postgreSQL database using the database cursor after the connection, and stores data into a DataFrame.	table_name	Table name to retrieve data from the selected database.

Taking into account the MongoDb database support, whose class is presented in the Table 4.3, the data managing package is built using functionalities from Pymongo[10] version 3.11.2 and Pandas version 1.1.5 libraries. Also in this case all the parameters needed for a connection to a MongoDB database are retrieved from the *MongoDB* class, but the *read_data* method, employed to gather the desired collection[11], makes some assumption on data: since the idea is to convert every data format to a Pandas dataframe it is necessary to perform some checks over MongoDB semi-structured data; for this reason, the method checks if the collection contains nested data type, specifically are not allowed documents composed by dictionaries of lists and

[8]SQLAlchemy library website: `https://www.sqlalchemy.org/`

[9]Pandas library website: `https://pandas.pydata.org/`

[10]Pymongo library website: `https://pypi.org/project/pymongo/`

[11]A collection in a MongoDB context consists in a set of documents. It can be compared a row of a data table.

vice-versa to avoid the creation of a potentially enormous number of columns during the conversion to a dataframe. Moreover, if lists are detected, they are reduced to a single value diversifying the averaging according to the single value type: for strings, it is chosen a concatenation summarization while numeric values are just averaged.
If no error occurs the final results consists of a dataframe having as columns every different document attribute in addition to the sub-attributes, taken into account with the *json_normalize* function in the form of *attr.sub_attr* and so on.

Table 4.3: MongoDB class with methods and attributes.

Class/*Method*	Parameter/Attribute	Description
MongoDB	Parameters list: username (str, optional), password (str, optional), db_name (str), host_name (str), port (int) .	These class parameters are necessary for connecting to the selected PostegreSQL database from the specified host.
read_data: Reads from a Mongo database after the connection, performs structural checks to allow only the processing of not double-nested data and finally stores it into a dataframe.	collection	Collection name to retrieve data from the selected database.

The last data managing component is represented by the support to CSV data format: once retrieved the correct path where the CSV file is stored, the *read_from_csv* function performs a simple arguments mapping retracing the ones necessary to use the Pandas *read_csv* method.

4.4 Algorithmic modules implementation

The algorithmic section implementing practical solutions for k-anonymity and differential privacy properties represents the P-PPA framework core and provides data handling and information gathering preserving the data subjects' privacy. Two different modules take into account the compliance with the respective privacy above-cited properties; both will be presented in details further in this chapter.

4.4.1 The Mondrian algorithm: a practical design

To tackle the k-anonymity problem we decided to implement the Mondrian algorithm [27] by Kristen LeFevre. Various Mondrian flavours exist in literature revolving around some different conceptual choices especially about generalization and partitioning mechanisms: for this project, we have assumed no generalization hierarchies for categorical attributes since the P-PPA framework is built to work with generic data in contexts in which there may not be available generalization information; moreover the algorithm strictly partitions data, in contrast with relaxed models, not allowing data overlapping when performing a partition split, overall obtaining less data loss.
The Mondrian P-PPA implementation develops around the only adequate Python project available online [39] expanding it with many major upgrades:

- Except for some outliers, this project should anonymize all datasets while the original work only permits as input two specific scientific datasets whose parameters are hard-coded.
- It is possible to choose column attributes over which the k-anonymity property is checked before performing the Mondrian algorithm.
- By default, all attributes are considered as QIAs but to administrators it is given the possibility to select the desired set of QIAs.
- A "whitelist" function is provided; administrators can select one or more columns and the algorithm will not consider them for checking the k-anonymity property but it will display them anyway in the final anonymized dataset. This functionality can be exploited for classification purposes, selecting the sensitive column as the class label values.
- A *Kanonymity* superclass was designed to permit a modular approach to extend functionalities adding for instance another k-anonymity algorithm like Incognito [25].

Table 4.4: Mondrian class with methods and attributes.

Class/*Method*	Parameter/Attribute	Description
Mondrian	k	k parameter for $k-$anonymity
	user_choice_index	List of integers. Represents the indexes of the dataframe columns over which the $k-$anonymity property is checked before performing the Mondrian algorithm. If the user does not express any preferences about columns to check before manipulating data, the user_choice_index attribute is None.

Table 4.4: Mondrian class with methods and attributes.

Class/*Method*	Parameter/Attribute	Description
	attr_names	List of strings of all dataframe column attributes.
	qi_indexes	List of indexes of columns chosen as QIAs. This parameter is meant to be used by administrators. If no list is specified, the algorithm will consider all columns as QIAs.
	whitelist	List of indexes of columns chosen to be not considered for $k-$anonymity process. These columns will be just displayed as they are.
	qi_indexes_final	same as the qi_indexes variable, but with adjusted (translated) indexes after removing columns that are neither QIAs nor whitelisted.
	whitelist_final	same as whitelist variable, but with adjusted (translated) indexes after removing columns that are neither QIAs or whitelisted.
	GCP [40]	Global Certainty Penalty.
perform: Extends the perform method of $k-$anonymity class. It's the only method that has to be called from outside of this module, all the operations are made here.	df	Input dataframe to be checked.
__check_kanon_from_columns: This method will calculate the simple combinations set of columns and rank it according to first to subset length and then to columns priority order. Finally it will return the top set, if any, according to the ordering.	df	Input dataframe to be checked.
	user_choice_index	Check the Mondrian class attribute result attribute for details.
__read_data: This method takes as input a dataframe and creates the data structures that will be used to execute the Mondrian algorithm.	df	Input dataframe to transform into internal data structures.
__convert_to_df: This method simply converts data from the internal list of lists format to dataframe format, taking out columns that are neither QIAs nor whitelisted from the list comprehending all columns names *attr_names*.	result	This is the result of the Mondrian algorithm in the internal list of lists of strings format, one list of strings for each dataframe row. If a string (but actually an index) corresponds to a categorical attribute, then it will be reconverted to a real string (the previous attribute value) thanks to intuitive_order variable.
__get_result: This method creates the Core object, obtains the list of partitions from the Mondrian algorithm, and returns the final dataframe.	data	List of floats, for each dataframe row, corresponding to non categorical attributes.
	intuitive_order	List of lists of strings corresponding, for each column, to its attribute domain.
__convert_to_raw: During preprocessing, categorical attributes are converted to numerical attribute using intuitive order. This function will convert these values back to they raw values. For example, Female and Male may be converted to 0 and 1 during anonymization.	result	Check the _convert_to_df result attribute for details.
	intuitive_order	Check the _get_result intuitive_order attribute for details.

The *Mondrian* class, that extends the *Kanonymity* superclass, represents the conceptual entry point to obtain a k-anonymous dataset; its details are shown in Table 4.4. After the needed parameters initialization, the $perform$ method sets up the pipeline to provide all the functionalities above-explained: if the *user_choice_index* is set, the columns corresponding to the indicated indexes are checked, though the $\_check\_kanon\_from\_columns$

method, and if they already compose k-anonymous data, the algorithm returns them without further computations.
If no index is specified or no attribute combination leads to k-anonymous data, the *_read_data* method retrieves data from the passed dataframe taking into account all the complexity deriving from settable QIAs and *whitelist* attribute indexes; then, all data are converted into numerical values. The returned information consists of numerical data and a data structure called *intuitive_order* thanks to which, after the anonymization, it is possible to convert categorical attributes to their original values; these structures are exploited from the *_get_result* method to create *Core* class object, described through Table 4.5.

Table 4.5: Core class parameters description.

Class/*Method*	Parameter/Attribute	Description
Core: the majority of the documentation for this class can be found on the [39]. In this table are described the class parameters related to adjustments performed by this project.	data	List of floats, for each dataframe row, corresponding to QIAs and whitelist attributes.
	qi_indexes	Check the Mondrian class attribute result attribute for details.
	qi_len	Number of columns.
	k	k parameter for $k-$anonymity.
	result	List of Partition objects. Check [39] for further details on the Partition class
	qi_dict	List of dict, one dictionary for each column. For each column attribute domain, there's a dictionary with a float (if the attribute was non categorical) or an integer (if the attribute was mapped from categorical to integer) as key and an integer as value: this represents the mapping between the attribute value (the original one or the transformed one because it was categorical) and the index of the span for that attribute domain.
	qi_range	List of floats ranges, one for column attribute domain.
	qi_order	For each column, there's a list of attribute values ordered in ascendant order.

The *Core* class performs the actual Mondrian algorithm splitting multi-dimensional horizontal partitions considering different attributes, according to the order given from their domain span. The partitioning concept is modelled using the *Partition* class after the selection of the attribute with the widest range of different values normalized considering the other attribute ranges: starting from the whole dataset, it is recursively partitioned on the median value calculated on the chosen attribute until both resulting splits contain equal or more rows than the k parameter. At the end of this process, the attribute values of the obtained partitions set belonging to the same column are merged back together obtaining generalization ranges, using the hard-coded *tilde* character. The *_convert_to_raw* method, shown in Table 4.4, uses the list of partitions composed of anonymized original and converted numerical data and the *intuitive_order* structure to convert back categorical attributes to their original values. Finally, the *_get_result* method converts the obtained *list* of *lists* of *strings* into a dataframe utilizing the Pandas library.

4.4.2 Differentially private statistics though the IBM differential privacy library

The P-PPA framework presents a practical implementation of the differential privacy model. Different freely available libraries were taken into consideration but we chose the IBM differential privacy [41], [42] one, since it stood out as the most complete, simple in usage and well-documented framework to gather statistical information with a differentially private flavour. Released under the MIT Open Source license, it requires Python 3.4, it leverages on Python libraries like Pandas and Numpy[12] and is strongly based on the Laplace perturbative mechanism.
The modularity feature that characterizes the whole P-PPA project is observed for the differential privacy package as for the k-anonymity one, since also in this case it is provided a sort of interface, the $Differential\_privacy$ class, that permits to implement differentially private algorithms exploiting mechanisms different from the IBM differential privacy library.

We chose to create a wrapper module, described in Table 4.7, for IBM

[12]Numpy library website: `https://numpy.org/`

Table 4.6: IBM differential privacy supported operations mirrored by the PPPA wrapper module [41].

Tools	**Notes**
histogram, histogram2d, histogramdd	Histogram functions mirroring and leveraging the functionality of their NumPy counterparts, with differential privacy
mean, var, std	Simple statistical functions mirroring and leveraging their Numpy counterparts, with differential privacy, including their not-a-number (NaN) versions

differential privacy library [41] statistical operations like the mean, variance, standard deviation with their respective not-a-number (NAN)s variants, and the histogram in one, two or d-dimensions, which a list is shown in Table 4.6. For these functionalities the IBM differential privacy developers maintained collinear ideas and concepts comparing with the corresponding ones from the Numpy statistical operations implementation; for this reason, they demand the same requirements, parameters and use modalities. According to the requested query, the implemented *Dp_IBM_wrapper* class, after some operational boundary check, calls the right function using the *dispatcher* dictionary that stores the pointers to the wrapper function, in which parameters are manipulated, that finally calls the requested IBM differential privacy operation.

Table 4.7: Dp_IBM_wrapper class with parameters and principal method.

Class/*Method*	Parameter/Attribute	Description
Dp_IBM_wrapper: This class extends the Differential_privacy class. Supported queries: sum, nansum, mean, nanmean, var, nanvar, std, nanstd, histogram, histogram2d, histogramdd.	column_indexes	The list containing one or more integers corresponding to the dataframe columns over which perform the requested query.
	query	The requested query. Check 'dispatcher' keys [37] to be aware of the currently supported queries.
	epsilon	The epsilon parameter for the differential privacy query.

Table 4.7: Dp_IBM_wrapper class with parameters and principal method.

Class/*Method*	Parameter/Attribute	Description
	dispatcher	The dictionary that maps a string, corresponding to the desired query to be performed in a differential privacy manner, with the function pointer that has to be called to execute that query. If some functionalities have to be added, it is just necessary to import the required method from the IBM library and add it as the value of the new query string thanks to which it will be called.
perform: this method extends the one present in the Differential_privacy class. The right function is called from the dispatcher.	df	Input dataframe to query for the requested operation in a differential privacy flavour.
	other_args	other arguments are accepted to reflect IBM library methods. These will be passed to IBM methods and there handled.

Chapter 5

k-anonymization effect on classification task

In this chapter, we will present a detailed analysis to study how the anonymization provided by the Mondrian module determines information loss after a ML pipeline has been applied. We will focus on the classification problem from the supervised ML branch since it's easier to measure outputs and consequently the information loss in comparison to an unsupervised one as, for instance, the clustering task. For this analysis, four different ML models are taken into consideration; each of them was trained on three different publicly available datasets.

5.1 Implemented machine learning algorithms: theoretical notions

The currently most widespread ML algorithms as Adaboost [43] and random forests [44] exploit ensembling mechanisms that permit to enhance the classification performances through voting and weighting policies; despite that, the goal of this analysis is to understand and explore the k-anonymization effect on data accuracy and information retrieval rather than obtain the absolute best classification score. For this reason, we selected simpler and well-known models like decision tree [45], Support Vector Machine [46], K-nearest neighbors [47] and a simple feed-forward neural network.

Decision tree

As can be understood from the self-explicative name, this algorithm is based on a tree structure composed of nodes that represent a possible choice on a specified attribute among all the ones present in the dataset on which the tree is built. After the tree construction, to make a classification the decision tree is climbed down from the root to one leaf that corresponds to one of the possible value of the class label.
This algorithm is mainly used for the simplicity with which it is possible to explore what rules it learned from data besides the ability to take non-linear decisions on complex datasets: this happens thanks to the subsequent evaluation of the different attributes, in this way it can classify correctly also non-linear data regions. On the contrary, it suffers from instability related to little changes in data used for the training, that is linked to the overfitting[1] problem caused by its tendency to split on the same attribute until it reaches a leaf containing just elements of the same class.

Support vector machine

Its first implementation was used to perform binary classification but it can be extended also to multiclass scenarios. Taking into account the binary and linear situation for simplicity, the basic idea consists of finding the equation that better linearly divides the data according to their label; it is treated as an optimization problem that aims to find the hyperplanes[2] that span the largest region in which no miss-classification points are allowed, if the dataset is linearly separable. It is demonstrated that just some particular data points called *support vectors* contribute to the solution. This is mentioned as the *hard margin* problem, but there exists also the *soft margin* one in which miss-classified points are allowed. It is possible to successfully exploit SVM also for non-linear classification tasks, thanks to the *kernel trick* [48].
Support vector machines work very well in practice even with datasets with a low number of rows, they are flexible and powerful thanks to kernel

[1] A model overfits when it excessively learns training data patterns at the point that it can't recognize others in diverse data.

[2] It represents the same concept of a two-dimensions geometrical plane, but in a n-dimensions, with $n > 2$.

application. On the contrary, it's difficult to tune during the learning phase and it's computationally problematic in terms of time and memory.

k-nearest neighbour

One of the simpler ML algorithms for classification tasks, it assigns the class label to the unseen data according to the majority of its k nearest neighbours. Its simple working mechanism represents one of its strengths, but this is counterbalanced since choosing the k parameter can be difficult and exposes the classification to outliers influence; moreover, the computation burden can be a problem since to classify a new data the algorithm must compute the distance between it and each training data.

Neural network

In the last decade, the research has made heavy progress optimizing neural network performances. The basic feed-forward network is composed of three fundamental notions: a neuron, the basic computational unit; an activation function that provides non-linearity, simulating biological neuron impulses; the weights independent in each neuron, multiplied with data and the actual values calculated during the classification task. According to the neuron position, they are classified as input, hidden or output: the hidden neurons constitute the hidden layer that is connected with both input and output layers. Moreover, if each hidden neuron is connected to each input and output one, the layer is considered *fully connected*; more hidden layers can be stacked together obtaining a multilayer fully connected network.
During the learning process every input data passes through the layers, it's multiplied with the weights and each neuron applies the activation function, from the input to the output layer completing the so-called epoch; at the end of each epoch the error, calculated as the difference between the obtained output and the expected one, is back-propagated and the weights consequently adjusted.
A neural network can model very complex non-linear equations gathering patterns and insights from data. For this reason, one of the most important drawbacks is the necessity of large data amounts, besides the fact that it is very difficult to interpret the learned patterns; these aspects interfere with the neural network spread in real-world scenarios: for instance, in many sectors like the medical one, it could be fundamental to explain the

reasons behind some medical treatments rather than just output the medical outcome.

5.2 Datasets presentation and metrics clarification

The statistical analysis performed in this thesis context takes into consideration three publicly available datasets from now mentioned as *Adult* [49], *Credit* [50] and *Diabetes* [51]; all these three datasets are human-related, composed of one row per individual and, except for the Credit one whose attributes are all numerical, with a various ratio between numerical and categorical attributes: for the *Adult* dataset, categorical and numerical attributes are balanced, while for the *Diabetes* one they correspond to a 2:1 ratio, granting overall a heterogeneous mix. The *Adult* dataset predicts whether income exceeds 50000$ per year based on census data, the *Credit* one was created to study customers' default payments in Taiwan and finally thanks to the *Diabetes* dataset it is possible to analyze factors related to hospital readmission for diabetic patients.
Although this analysis aims to focus on information retrieval compared to privacy loss rather than presents a ML classification pipeline at the state of the art, a minimal effort to the data exploration and consequently data preparation was necessary. For *Adult* and *Credit* datasets the data preparation phase was simple and trivial since were just removed rows with missing values and no structural changes were made. On the contrary, *Diabetes* data was mapped to suit a binary classification task rather than the original ternary one since initial classification results indicated that it was a too complex assignment: the original class values distinguished between readmitted patients before 30 days, after 30 days or not readmitted at all for the illness treatment, while the binary classification just divides patients into readmitted or not. Moreover, for each patient, just the row related to the first encounter was retrieved, many low variance attributes were dropped and other biological indicators and diseases codes were grouped to reduce the number of features.
In the Table 5.1 are shown the attributes and the corresponding type for each dataset used for the statistical analysis, after the preparation step.

Table 5.1: Attributes and types list for each of the three proposed dataset.

Adult	**Credit**	**Diabetes**
age (int)	limit_bal (float)	race (str)
workclass (str)	sex (int)	gender (str)
fnlwgt (int)	education (int)	age (float)
education (str)	education (int)	admission_type_id (str)
educationnum (int)	marriage (int)	discharge_disposition_id (int)
maritalstatus (str)	age (int)	admission_source_id (int)
occupation (str)	pay_0 (int)	time_in_hospital (str)
relationship (str)	pay_2 (int)	num_lab_procedures (str)
race (str)	pay_3 (int)	num_procedures (str)
sex (str)	pay_4 (int)	num_medications (int)
capitalgain (int)	pay_5 (int)	number_outpatient (int)
capitalloss (int)	pay_6 (int)	number_emergency (int)
hoursperweek (int)	bill_amt1 (float)	number_inpatient (int)
nativecountry (str)	bill_amt2 (float)	diag_1 (str)
income (str) - Class attribute	bill_amt3 (float)	diag_2 (str)
	bill_amt4 (float)	diag_3 (str)
	bill_amt5 (float)	number_diagnoses (int)
	bill_amt6 (float)	a1cresult (str)
	pay_amt1 (float)	metformin (str)
	pay_amt2 (float)	glipizide (str)
	pay_amt3 (float)	glyburide (str)
	pay_amt4 (float)	pioglitazone (str)
	pay_amt5 (float)	rosiglitazone (str)
	pay_amt6 (float)	insulin (str)
	default (float) - Class attribute	change (str)
		diabetesMed (str)
		readmitted (str) - Class attribute

Considering the results evaluation aspect, the statistical metrics used for the analysis correspond to precision, recall and f1-score, that respectively measure the percentage of actual positive predictions over the total positive predicted samples, the percentage of correctly identified samples over the actual positive samples and the harmonic average between precision and recall metrics.
Since the focus analysis was on the correct classification of the so-called minority class[3], notoriously a much complex task compared to obtaining a good average accuracy, the overall accuracy metric was not taken into account. All the analyzed datasets suffer from an imbalance between classes: considering *Adult* and *Credit* databases, the ratio between majority and minority class is more or less equal to 3 while for the *Diabetes* one it

[3]The minority class represents the samples whose class label corresponds to the one with the minor number of occurrences in the dataset

corresponds to about 1.5, representing respectively a strong and a slight imbalance.
For this reason, to perform principal statistics we have considered *macro* averages over *micro* ones; let's consider for instance the precision and recall definitions according to both averages:

$$\begin{aligned} precision_{macro} &= \frac{\sum_{classes} precision\_of\_class}{number\_of\_classes}, \\ recall_{macro} &= \frac{\sum_{classes} recall\_of\_class}{number\_of\_classes}, \\ precision_{micro} &= \frac{\sum_{classes} TruePos\_of\_class}{\sum_{classes} TruePos\_of\_class + FalsePos\_of\_class}, \\ recall_{micro} &= \frac{\sum_{classes} TruePos\_of\_class}{\sum_{classes} TruePos\_of\_class + FalseNeg\_of\_class}. \end{aligned} \tag{5.1}$$

As it can be noticed from the equations in 5.1, the micro average uniformly considers the predicted samples from the minority and the majority class while on the contrary, the macro average doesn't weight equally samples from different classes: for this reason a low precision or recall value will heavily impact on the overall macro average result.
It's important to notice that, for binary classification tasks, it can be mathematically demonstrated that the recall, precision and f1-score micro average correspond to the accuracy metric; in classification tasks for which every test case is guaranteed to be assigned to exactly one class, it stands that $\sum_{class} TruePos\_of\_class = \sum_{class} FalseNeg\_of\_class$. Since all the analyzed cases for each selected datasets are binary classifications, comparing different averages will correspond to a comparison between macro precision, recall or f1-score and the accuracy as the micro metric.

To measure the information loss for the anonymization task we used the global certainty penalty (GCP) [40] maintaining the same approach of the original Mondrian algorithm Python implementation [39]. This metric is defined starting from the normalized certainty penalty (NCP) definition that is different considering numerical and categorical attributes. Having an equivalent class G, the NCP measure for a single numerical attribute A_{Num} taken from G is defined as [40]:

$$NCP_{A_{Num}}(G) = \frac{max^{G}_{A_{Num}} - min^{G}_{A_{Num}}}{max_{A_{Num}} - min_{A_{Num}}} \tag{5.2}$$

that represents the ratio between its range considering G and the entire table. For a categorical attribute A_{cat}, we consider a situation in which generalization hierarchies nor distance metrics are defined. The NCP for a categorical attribute A_{cat} is [40]:

$$NCP_{A_{Cat}}(G) = \left\{ \begin{array}{ll} 0, & \text{if } card(u) = 1 \\ \frac{card(u)}{|A_{Cat}|}, & \text{otherwise} \end{array} \right\} \tag{5.3}$$

where u is the first ancestor common to all values of the A_{cat} attribute present in G, and card(u) corresponds to the number of its different values. Finally, these two concepts are summed up obtaining the GCP metric for the whole anonymized released table P [40]:

$$GCP(P) = \frac{\sum_{G \in P} |G| \cdot NCP(G)}{d \cdot N} \tag{5.4}$$

where P is the anonymized released table, NCP(G) is the summary of the single NCP for all the QIAs, N is the number of rows in the original non-anonymized data and d corresponds to the number of QIAs.

5.3 Statistical analisys pipeline

This section will explain in details all the technical choices and the steps related to the performed statistical analysis. For this purpose, we have used the above-presented datasets mentioned as *Adult*, *Credit* and *Diabetes* and the *k*-anonymity P-PPA module to tackle the data anonymization task. The algorithmic core is composed of three ML models from the SciKit Python library[4], and a simple neural network build exploiting internal components from the Keras Python library[5]; moreover, the neural network is composed

[4]Link to the SciKit library `https://scikit-learn.org/stable/`

[5]Link to the Keras library `https://keras.io`

of three hidden layers, whose dimensionality changes going through the network end: from 256, 64 to 16 neurons for the hidden layers and 1 for the last output layer. Finally, the ReLu (rectified linear unit) [52] activation function is used for the hidden layers and the sigmoid [53] one for the last output layer.

We can divide the analysis into three phases according to the technical tools and the diverse goals, following consistent reasoning considering all the combinations between datasets and models, with some exception due to computational limits.

Best parameters grid-search

The first step consists of a classical ML model parameters tuning using a 4:1 ratio split for training and test data. The parameters to be tuned and their value considered for each model are shown in the Table 5.2 and Table 5.3. In this phase, the grid-search represents the main operation and consists, as can be assumed from its name, of the training and the evaluation of the ML model initialized with the chosen parameters values, for each of their possible combinations. Moreover, to evaluate each combination, we used a k-fold cross-validation approach [54] that repeats the evaluation dividing the training set into k folds, using $k-1$ folds to train the model with the selected parameters combination and the remaining fold to test it. According to the number of parameters, their domains and the number of folds, this task can be computationally expensive since it can generate thousands of combinations: for this reason, the most resource-demanding models as the SVM and neural network, are analyzed with fewer tuning parameters values combined with fewer data folds.
For each model and dataset, once found the best macro f1-score parameters values, they are used to train a new model taking into account the whole training dataset, not dividing it in folds. At the end of this phase are available the best decision tree, SVM, KNN and neural network model for the *Adult*, *Credit* and *Diabetes* dataset.

Table 5.2: ML models tuning parameters and their meaning.

Model/*Parameter*	**Description**
Decision tree	
n_principal component analysis (PCA) [55]	Defines the number of principal component analysis into which transform the current columns.
max_depth	It represents the maximum value for the tree depth.

Table 5.2: ML models tuning parameters and their meaning.

Model/*Parameter*	**Description**
min_samples_leaf	This is the minimum value of training data required to be a leaf node.
min_samples_split	This is the maximum value of training data required to be a leaf node.
criterion	It sets the measure with which a split on a node is evaluated; the Gini or the entropy measures are used.
Support Vector Machine	
n_PCA	Defines the number of principal component analysis into which transform the current columns.
c	It's a regularization parameter, controls the cost of misclassification on the training data; the bigger the c value is, the narrower is the margin between splitting hyperplanes.
kernel	Specifies what mathematical transformation is done to data to operate in a higher-dimensional space.
gamma	This parameter inversely indicates how influent for the classification are the points that composed the selected support vectors; the higher the gamma value is, the more jagged the boundaries are.
$K-$**nearest neighbors**	
n_PCA	Defines the number of principal component analysis into which transform the current columns.
n_neighbors	To chose the correct label, this value indicates how many neighbour data points, to the input data, are considered.
weights	Indicates how is valued the training points distance compared to the test one.
p	An index related to the chosen distance metric; for instance, $p = 2$ corresponds to the Euclidean distance.
algorithm	KNN is available in various version. The exact algorithm can be specified.
Neural network	
num_epochs	The traning lenth in number of epochs.
lr	The learning rate it's a multiplicative factor that affects the weights updating rate.
lr_reduction_epoch	A sort of decay factor for controlling the learning rate through the epochs.
batch_size	The training data is divided into batches, it determines its dimension.
optimizer	This parameter select one of the optimizer available in Keras library to perform the weights update and handle the learning rate, since this can be considered an optimization problem focus on minimizing the loss for each epoch.

Table 5.3: ML models tuning parameters and their values for each dataset.

	Adult	**Credit**	**Diabetes**
Decision tree			
n_PCA	[8, 10, 12, 14]	[14, 16, 18, 20, 22]	[10, 12, 15, 17, 19]
max_depth	[9, 10, 11, 12, 13, 14]	[8, 10, 12, 13, 14]	[6, 8, 10, 12, 14]
min_samples_leaf	[15, 20, 25, 30, 35, 40, 45, 50]	[40, 45, 50, 55, 60, 65]	[100, 120, 140, 160]
min_samples_split	[25, 30, 35, 40, 45, 50]	[30, 35, 40, 45, 50, 55]	[20, 30, 40, 45, 55]

Table 5.3: ML models tuning parameters and their values for each dataset.

	Adult	**Credit**	**Diabetes**
criterion	["gini", "entropy"]	["gini", "entropy"]	["gini", "entropy"]
Support Vector Machine			
n_PCA	[10, 12, 14]	[14, 16, 18, 20, 22]	[13, 15]
c	[0.6, 1, 3, 6, 10]	[0.05, 0.1, 0.3, 0.6, 1]	[5, 10]
kernel	["linear", "rbf"]	["linear", "rbf"]	["linear", "rbf"]
gamma	[0.01, 0.03, 0.06, 0.1, 0.3, 0.6, 1]	[0.005, 0.01, 0.03, 0.06, 0.1]	[0.1, 1]
$K-$**nearest neighbors**			
n_PCA	[8, 10, 12, 14]	[14, 16, 18, 20, 22]	[10, 12, 15, 17, 19]
n_neighbors	[21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41]	[21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41]	[25, 27, 29, 31, 33, 35, 37]
weights	["distance", "uniform"]	["distance", "uniform"]	["distance", "uniform"]
p	[1, 2]	[1, 2]	[1, 2]
algorithm	["ball_tree", "kd_tree"]	["ball_tree", "kd_tree"]	["ball_tree", "kd_tree"]
Neural network			
num_epochs	[50, 75, 100]	[50, 75, 100]	[50, 75, 100]
lr	[0.01, 0.005, 0.001]	[0.005, 0.001, 0.0005]	[0.005, 0.001, 0.0005]
lr_reduction_epoch	[every num_epochs/5, every num_epochs/2]	[every num_epochs/5, every num_epochs/2]	[every num_epochs/5, every num_epochs/2]
batch_size	[64, 128, 256]	[64, 128, 256]	[64, 128, 256]
optimizer	["Adam", "AdaDelta"]	["Adam", "AdaDelta"]	["Adam", "AdaDelta"]

Evaluation of k-anonymized datasets

In this phase, model parameters collected in the previous step for each dataset are used to re-train all the defined ML models on the k-anonymized data obtained from different k values; these best parameters obtained on non-anonymized datasets, from now on called *original parameters*, are used to re-train and test models with increasing k-anonymization degree datasets: the goal is to understand if and how anonymized datasets confuse classifiers degrading their performances, in this phase using *original parameters*.
For this purpose, after each anonymization, it is necessary another data preparation phase since the P-PPA for the k-anonymization transforms every column in a string, simple o concatenated with the *tilde* character, while the classifiers accept as input only numerical data. Through a custom encoder function, all the originally numerical attributes generalized in a string representing a range with two or more values, are converted back averaging between the maximum and the minimum values between the ones composing that string. The categorical ones are instead converted to numbers using the *LabelEncoder* class fro SciKit library.

Training and evaluation on k-anonymized datasets

In this phase, we want to simulate real-world applications in which we do not want the original data to be available. For this reason, we repeat the parameters grid-search and the k-fold validation for each model and dataset following the previous phase procedures, but this time using anonymized datasets with ascending k values: for each model and dataset anonymized with different k values we obtain the best parameters configuration to which, from now on, we will refer as *custom parameters*. We have to mention that due to the computational effort for training SVM and neural network, to collect their *custom parameters*, we consider just a subset of their original parameters values domains.

5.4 Presentation of obtained results

The above-cited pipeline phases aim to inquiry different aspects concerning the data k-anonymization and the ML models behaviour in different situations; these perspectives can be summed up and semantically divided into three investigations areas:

- GCP evalutation for k-anonymized datasets, varying the k parameter.
- The stability of the ML models analyzing how classification outputs vary choosing different *original parameters*. We repeat analysis to assess scores differences in using *custom parameters* instead of *original* ones.
- Comparison between precision, recall and f1-score metrics for each class; differences considering the macro and micro averages for each metrics. Also in this case, we examine these results from both the *original parameters* and *custom parameters* perspectives.

The last cited and major perspective will be presented in the following subsection; moreover, we recall that the collected statistics refer to all the combinations given by the four ML classification models trained on the three datasets called *Adult*, *Credit*,*Diabetes*, except for the GCP measurements that are independent of the used models.

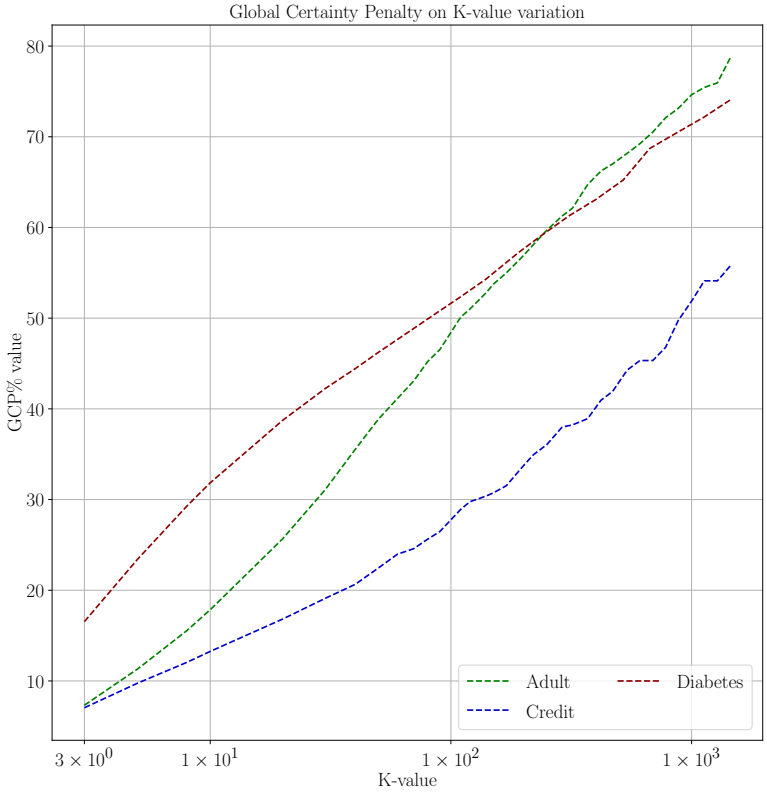
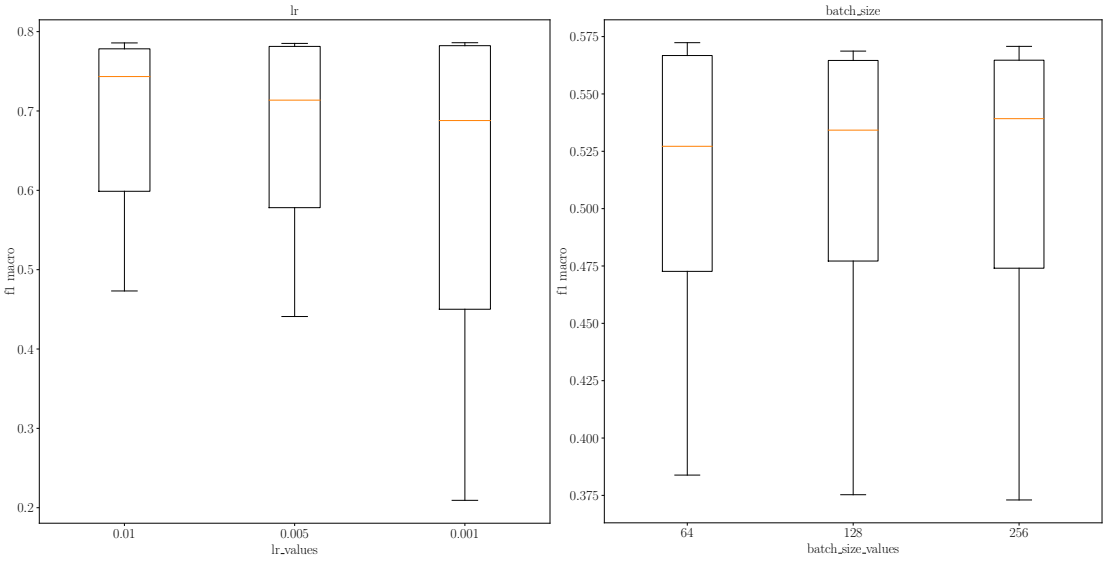


Figure 5.1: GCP for *Adult*, *Credit* and *Diabetes* datasets, varying k.

Information loss measure for proposed datasets

We provide an assessment of the information loss using the above-explained GCP metric, introduced by k-anonymization algorithms as the Mondrian [40]. The proposed Figure 5.1 displays a preserving information overview after anonymizations corresponding to *Adult*, *Credit* and *Diabetes* datasets, measuring the GCP for a set of increasing k values that span on the x-axis with a logarithmic scale. As we can see, the P-PPA k-anonymization module preserves more information for the *Credit* dataset, fixing the k value. For instance, we can locate a sort of $elbow - point$ considering approximately k=40: for the *Credit* dataset the GCP is slightly over 20% while with the same anonymization degree, on the *Diabetes* one the information loss is recorded at about 45%. Moreover, the trends are noticeably different for the rest of the considered k values since the GCP on the *Credit* dataset tends to increase much slower. The same analysis on the *Adult* dataset produces mixed results since before the above-identified $elbow - point$, its

loss behaviour corresponds to higher values but with an increasing rate that can be related with the *Credit* one, while after that, it reaches the *Diabetes* one but slightly smoothly.



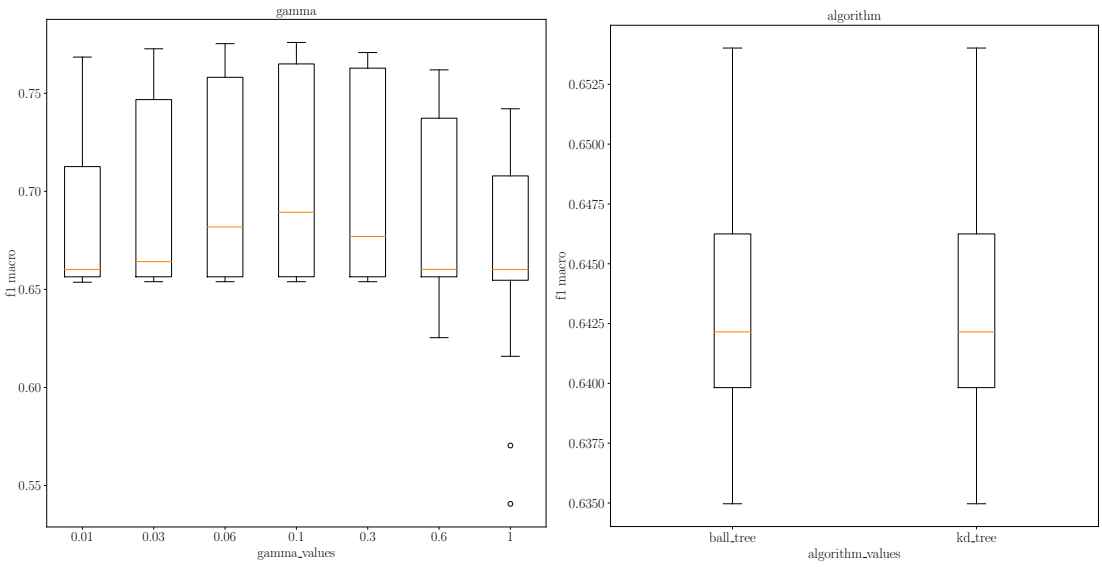
(a) Neural network - lr. **(b)** Neural network - batch size.

Figure 5.2: Boxplots for learning rate and batch size Neural network parameters, on *Adult* and *Diabetes* datasets.

Analysis of the stability for the selected ML models

It is necessary to understand the conditions and the ranges of use in which the classifiers predict a class label within a certain percentage. For this reason, we set up boxplots to analyze the influence of each *original parameter* for its model over the output; moreover, we use scatter plots to study the *custom parameters* impact for different k-anonymization degree comparing their scores with the ones obtained from *original parameters*, represented by k=1.
All the boxplots are built interpolating *original parameter* values distribution and the macro f1-score obtained during the best parameters grid-search, for each dataset, model and parameter values; recall/precision scatter plots visualize all the parameters configurations tried during their tuning after every k-anonymization with ascending k values, considering outliers the points with a precision or recall score too different from the global distribution

and not displaying them.
Examining boxplots, we immediately identified a reasonable consistency across models and datasets since in the majority of possible cases the macro f1-score did not change significantly browsing through the values of the parameters. We can instead notice that models produced different results, uniformly across the datasets, in terms of the ranges of the distributions: the neural network models produce distributions that can span over the 40% of the macro f1-score containing the majority of points in the low score section, while a smaller number contribute to select the best values out of the grid-search, as underlined in Figure 5.2. On the contrary, the Figure 5.3 shows that KNN and decision tree models are characterized by distributions that vary in very small percentage ranges, like 1 or a maximum of 2 points. SVM models establish themselves in the middle, producing a variation between 5 and 10%.



(a) SVM - gamma. **(b)** KNN - algorithm

Figure 5.3: Boxplots for gamma and algorithm parameters for SVM and KNN, on *Adult* and *Credit* datasets.

There exist some exception to these trends, especially for SVM models, since in some case it is difficult to understand which parameter value mainly contributes to higher macro f1-scores but it is clear which ones do not, as we can see in Figure 5.3a; moreover, in other cases like kernel and optimizer

values for SVM and neural network models, the most meaningfull value can be clearly identified, but these remain isolated cases.
Taking into account the effect on models training stability using *original parameters*, obtained with k=1, or *custom parameters* related to k>1, the precision/recall plots underline what could be easily predicted since models trained with the former ones, in most cases, obtain a clearly defined and higher score density population compared to models trained with the latter ones. An example of this can be seen in Figure 5.4.

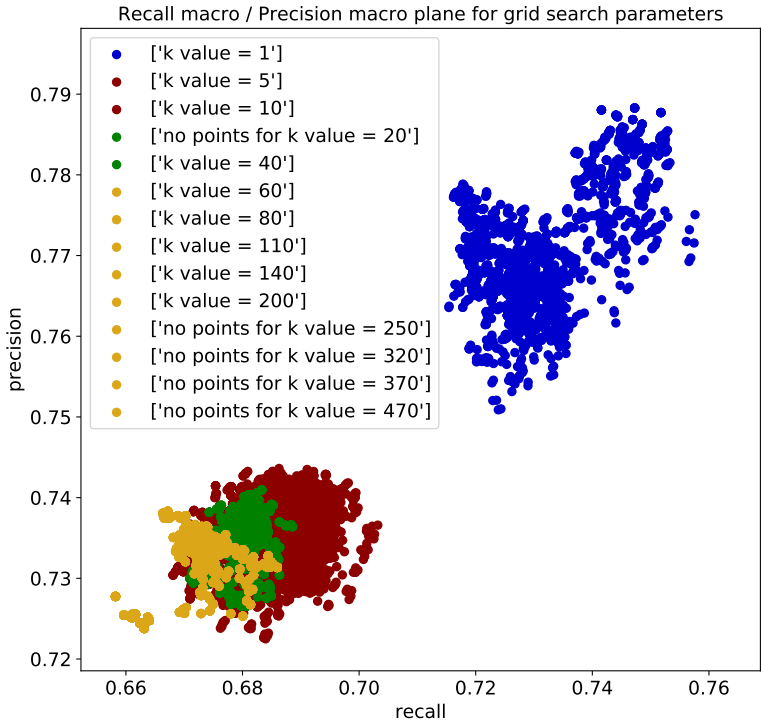


Figure 5.4: Scatter plot representing parameters configurations scores obtained ad different k-anonymization levels, for Decision tree - *Adult*.

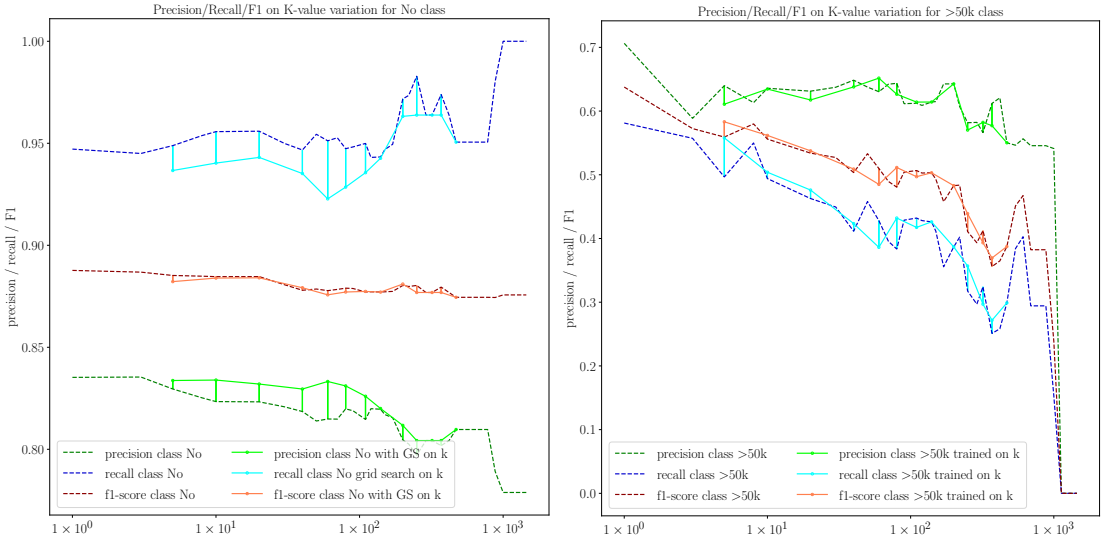
The indicated pattern is valid in general, but especially for decision tree models since the other ones presents more mixed populations.
Moreover, the more the anonymization degree on the *Credit* dataset increases, the more parameters points are not displayed since their precision and recall scores are considered outliers: the dataset structure could represent a possible explanation since it is the only one composed of just numerical values; as described in previous paragraphs, after the anonymization, numerical generalization ranges are re-converted into a single value averaging on the

domain edges and, for this reason, it is possible that they can't preserve much information as the anonymization degree increases. This result can still be explainable compared to the promising GCP scores previously-obtained on this dataset: the GCP formula could value less information loss on numerical attributes rather than on categorical ones, while a classifier precision can be affected from actual data patterns deleted increasing the anonymization level.

5.4.1 Anonymization effect on different training settings and metric averages

As previously stated, the goal of the following analysis is to understand, at first, how the selected classification models perform on arbitrary k-anonymized datasets after an accurate parameter tuning phase on original non-anonymized ones, exploiting the so-called *original parameters*; besides that, since in real use cases the data buyer can't access non-anonymized data and is forced to design, for example, a parameters grid-search on privacy-preserved one, we further investigate the effect on classification using *custom parameters*: as a recall, we refer to *custom parameters* to indicate parameters obtained after the tuning and a grid-search having available just k-anonymized datasets.
For this reason, we designed multiple graphs that interpolate the k considered values on a quasi-logarithmic scale with the precision, recall and f1-score on the y-axis, for each different class label. When we presented the datasets chosen for these experiments, we specified that every classification problem that is taken into consideration would be, or transformed, into a binary task for simplicity. The above-introduced analysis methodology is performed for both the majority and the minority class to carefully inquiry the different trends for each class label. As predictable, most of the times, the higher k is, the more the precision and recall for the majority class respectively decrease and increases, while for the minority one they both decrease; this because the more the data are anonymized, the more classifiers tend to output just the majority label, which is more frequent in the dataset.
We can find some differences in the steepness with which these trends tend to 1 or 0: the decision tree and SVM provide more linear results and begin to produce irregular predictions when reaching reasonable high k levels, between 300 and 500, considering *Adult* and *Credit* datasets. KNN and neural network models are generally less linear the more the k value increases

especially for precision and recall metrics, showing that behaviour starting from small k values in a range between 10 and 100. For SVM and neural network, this can be seen in Figure 5.5.

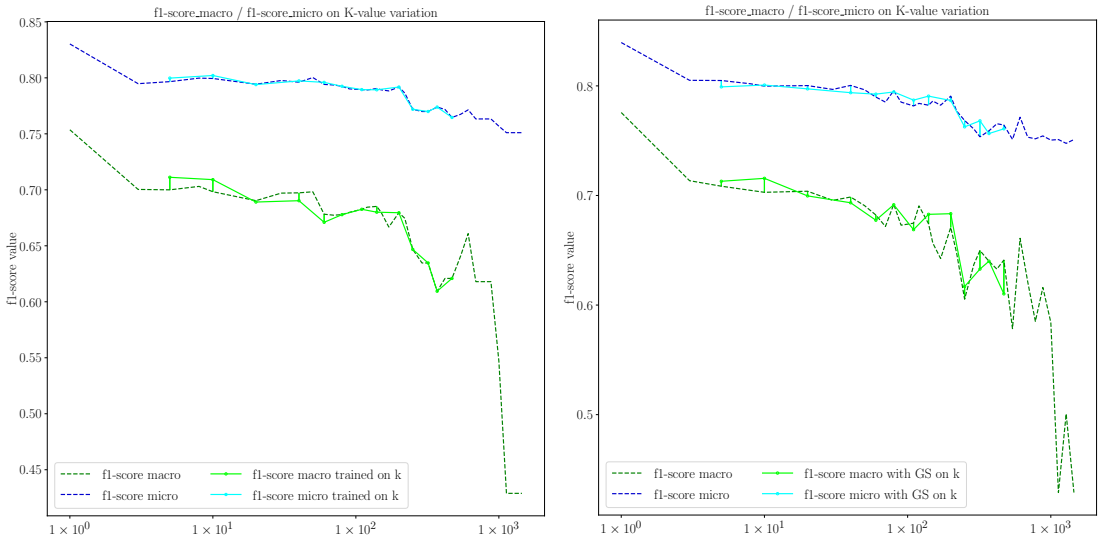


(a) SVM - *Credit* majority class. **(b)** Neural network - *Adult* minority class.

Figure 5.5: Precision, recall and f1-score trends for *Credit* majority and *Adult* minority class from SVM and Neural network, varying k; training on original and k-anonymized dataset.

Finally, all the presented ML models struggle more classifying the *Diabetes* dataset, perhaps due to the heavier data preparation and feature reduction performed on it that could have removed key data patterns that would be useful with higher information degradation levels.
The analysis of the classification results obtained with *custom parameters* on a subset of k values, does not underline particular patterns: the metric trio often records slightly different scores compared with the above-cited situation but considering contemporary all the three metrics, they tend to average the outcome, as we can see from the continuous lines on both plots in Figure 5.5.

About that, we could not find any particular response neither considering different k ranges to analyze nor across diverse models or datasets.
We recall that these measures regard binary classifications of heavily imbalanced datasets considering *Adult* and *Credit*, or slightly for *Diabetes* one. For this reason, we also inquiry the possible difference comparing micro and



(a) decision tree - *Adult* **(b)** KNN - *Adult*

Figure 5.6: Micro and macro f1-score from decision tree and KNN, varying k; training on original and k-anonymized dataset.

macro averages for a range of k values for the diverse k-anonymizations, at first with models represented by *original parameters* and subsequently with *custom* ones. The graphs show that the score differences for micro and macro averages correspond to the dataset imbalance degree, as can be supposed. We can notice how the performances of the classifiers decrease with a less than linear trend the more the k value increases, especially in a reasonable k range between 200 and 300, as shown in Figure 5.6. Considering the whole picture, we can not find significant pattern variations across datasets and models. Also in this case, these outcomes represent an encouraging note since models trained with *custom parameters* do not manifest meaningfully worst results.

Chapter 6

Conclusions

In this thesis work, we have presented a privacy-preserving framework whose functionalities are designed to be embedded in the wider PIMCity project; it aims to provide practical, flexible and modular solutions to cover all the data processing pipeline consisting in the data storage repository for user data, the modelization of the user consensus notion, the data value information retrieval through suitable metrics, the data purchase from interested data buyers and the data anonymization or the release of privacy-preserving statistics. The presented P-PPA framework implements this last-cited fundamental functionality, handling and retrieving data from three different data storage technologies and format as postgresSQL and mongoDB databases with the addition of the CSV format, maintaining a general approach.
The P-PPA module can manage both historical data, characterized by a one-to-one mapping between a record and a user, and recurrent one in which the same user can release more records: the provided k-anonymity module can process historical data, while the differential privacy one can handle both data types. We decided to implement the k-anonymity module exploiting the Mondrian algorithm [27] which can generalize categorical and numerical data without any further information about it; moreover, we modified it adding general capabilities since system administrators can select a specific set of QIAs or specify which attribute not to consider while k-anonymizing the data. The differential privacy module wraps IBM differential privacy library [41] functionalities allowing to query the stored user data and gather statistical information as the mean, standard deviation, variance and data structures to obtain single or multi-dimensional histograms.
As above-cited, we have designed the P-PPA framework to operate in a

flexible environment, easily communicating with other PIMCity components; for this purpose, we have chosen the simple and expandible Flask web framework to provide all the required REST APIs through which it is possible to exchange data with the P-PPA module, exploiting all the known REST mechanism and conciseness.
We have also exploited the above-proposed P-PPA framework to analyze the differences and the consequences that involve a ML classification pipeline with and without applying k-anonmyzation to the data used to train the models. For these experiments, we have taken into consideration the decision tree, the SVM, the KNN and a simple neural network model, while *Adult*, *Credit* and *Diabetes* datasets are selected.
For each model and dataset, we have inquired the relevance of some tuning parameter over the others using non-anonymized data to assess models stability; we have repeated the same experiments performing parameters grid-searches based on k-anonymized data discovering, in both cases, substantial stability and consistency in classification outcomes.
Since we have treated imbalanced binary classification tasks, we have compared results analyzing separately macro and micro averages for recall, precision and f1-score metrics, verifying an almost linear behaviour between datasets imbalance levels and differences in averages scores. We have further studied the k-anonymization effect on classification scores examining individually the ones related to majority and minority classes for higher and higher anonymization levels; as expected, the more privacy is preserved and information is lost, the more classifiers predict the majority class; on average, this behaviour is characterized from a smooth f1-score descending trend emphasizing moderate loss information from the anonymization with k values near a few hundred. Finally, the above-explained methodologies are reproduced utilizing model parameters obtained training them on varying k-anonymized datasets to simulate real-world use cases in which just anonymized data is released: in the majority of the cases the tuning of models remains stable and classification scores tend to deteriorate starting from reasonably high anonymization levels, similar to situations in which non-anonymized data were available.

6.1 Future works and improvements

The P-PPA project comprehends various components as the flask web framework, the data management module and the anonymization algorithmic support; for this reason, possible improvements will be presented following the same thesis logical order.
The currently REST API implementation can be expanded with additional functionalities: at things stand, we have conceived the P-PPA framework in a broader picture, interacting with the other PIMCity modules that take care of the data buying functionality and the privacy budget notion. A possible development is represented by the migration of the latter concept to the P-PPA framework: in this way, it should collect and maintain information about the data buyers' requests carefully checking the remaining privacy budgets, rather than just check if the token linked with the request is present in the YAML configuration file.
Taking into account the data management module, the only non-trivial improvement would be related to the MongoDB database management developing more general and smarter support to different kind of data structures, since the actual implementation solves the dimensionality problem not allowing semi-structured severely nested data. Having said that, the P-PPA framework should receive more information to design this kind of improvements, potentially reducing the generability degree currently obtained.
The same reasoning characterizes the development of a sharper k-anonymity module, specifying generalization hierarchies for categorical attributes allowing the Mondrian algorithm [27] to obtain on point generalization and a more precise loss metric. Moreover, thanks to different adjustment as transforming input data from a multidimensional to a two-dimensional space, the Mondrian algorithm could preserve slightly more information fixing the k value, achieving better results in term of computational time [56]. It is also possible to exploit the P-PPA modularity approach enlarging the support to the k-anonymity property developing Python implementation for other heuristic algorithms as Incognito [25].
The P-PPA framework can handle historical and recurrent data, but both data types are "statically" stored and passed to it; an important improvement would be extending the support to stream data that recalls frequent situations in which is necessary to perform zero-delay anonymization, from here for instance, the z-anonymity [57]. Besides the k-anonymity module, the P-PPA differential-privacy one currently provides basic statistical tools

from the IBM differential privacy library [41] as the average, and standard deviation; taking advantage of the project modularity, it should require minimal efforts to widen the available operation portfolio adding differentially private ML models like the k-means [58] and the PCA [55], implementing other suitable wrapper functions for the IBM differential privacy library.
Finally, the analysis of the k-anonymization effect on the ratio between privacy and information gathering, exploits the P-PPA k-anonymity module based on the Mondrian algorithm, but a further comparison employing other anonymization algorithms can be interesting, extending the reasonings also on other models like differential-privacy and z-anonymity.

Acronyms

ACID	Atomicity, Consistency, Isolation, Durability
AI	Artificial Intelligence. *Glossary:* AI
API	Application Programming Interface
CCPA	California Consumer Privacy Act
CRUD	Create, Read, Update and Delete
CSP	Communications Service Providers
CSV	Comma Separated Value
ECJ	European Court of Justice
GCP	Global Certainty Penalty
GDPR	General Data Protection Regulation
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IID	Independent and Identically Distributed
IoT	Internet of Things. *Glossary:* IoT
IP	Internet Protocol
JSON	JavaScript Object Notation
KNN	K-nearest neighbors
MIT OSL	The MIT Open Source license. *Glossary:* MIT OSL

ML	Machine Learning. *Glossary:* ML
NAN	Not a Number
NCP	Normalized Certainty Penalty
NIST	National Institute of Standards and Technology
NSA	National security Agency
P-CM	Personal Consent Manager
P-DS	Personal Data Safe
P-PM	Personal Privacy Metrics
P-PPA	Personal-Privacy Preserving Analytics
PCA	Principal Component Analysis
PDK	PIMS Development Kit
PIA	Personally Identifiable Attribute
PIM	Personal Information Management
QIA	Quasi-Identifier Attribute
REST	REpresentational State Transfer. *Glossary:* REST
SA	Sensitive Attribute
SQL	Structured Query Language
SVM	Support Vector Machine
URI	Uniform Resource Identifiers
URL	Uniform Resource Locator
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

Glossary

AI	A very wide definition used to describe programmed machines that present human-like features concerning problem-solving and thinking. ML is considered an AI subcategory.
biometric data	Personal data related about the natural person's body as the result of technical processing like body scan; information through which is possible to identify a person uniquely.
Data Mining	A multidisciplinary approach to analyze large datasets to extract useful information and patterns in a comprehensible form.
health data	Personal data related to the physical or mental health of a natural person; includes health care services provision, which reveals information about data subject's health status.
IoT	Describes a network composed of physical objects, linked together by sensors that use and gather data. Multiple traditional technologies can contribute to enlarge the "network of things" since it can link different domains like embedded systems, automation etcetera. It's often a notion related to "smart home" devices.

MIT OSL	The MIT Open Source license is a permissive licence created in the Massachusetts Institute of Technology (MIT) that permits to reuse the software under this license, but also can be merged with other MIT licensed software.
ML	A Computer Science branch dedicated to studying algorithms used to perform prediction or decision based on data features, without being programmed.
natural person	Refers to a human being, a different concept with respect to "legal person" that specifies a company, an entity.
REST	Defines a set of architectural constraints specified for Web Services applications. For example, a RESTful client request is made via defined formats like JSON, HTML, PHP etcetera.

Bibliography

[1] *2018 reform of EU data protection rules*. European Commission. May 25, 2018. URL: https://gdpr.eu/ (visited on 02/23/2021) (cit. on p. 12).

[2] *California Consumer Privacy Act (CCPA)*. State of California Department of Justice. URL: https://oag.ca.gov/privacy/ccpa (visited on 02/25/2021) (cit. on pp. 12, 23).

[3] *PIMCity - Building the next generation personal data platforms is a new EU-funded research project coordinated by Politecnico di Torino*. URL: https://www.pimcity.eu/ (visited on 02/25/2021) (cit. on p. 12).

[4] Yu Chung Wang William and Wang Yichuan. «Analytics in the era of big data: The digital transformations and value creation in industrial marketing, Industrial Marketing Management». In: 86 (Nov. 2020), pp. 12–15. DOI: https://doi.org/10.1016/j.indmarman.2020.01.005 (cit. on p. 14).

[5] F. Liang, W. Yu, D. An, Q. Yang, X. Fu, and W. Zhao. «A Survey on Big Data Market: Pricing, Trading and Protection». In: *IEEE Access* 6 (2018), pp. 15132–15154. DOI: 10.1109/ACCESS.2018.2806881 (cit. on p. 15).

[6] *The official website for NSA - the National Security Agency*. URL: https://media.defense.gov/2020/Jan/22/2002237484/-1/-1/0/CSI-MITIGATING-CLOUD-VULNERABILITIES_20200121.PDF (visited on 02/25/2021) (cit. on p. 16).

[7] E. Alepis and C. Patsakis. «Monkey Says, Monkey Does: Security and Privacy on Voice Assistants». In: *IEEE Access* 5 (2017), pp. 17841–17851. DOI: 10.1109/ACCESS.2017.2747626 (cit. on p. 18).

[8] Court of justice of the European Union commission. *The Court of justice of the European Union*. [Online; accessed 25-02-2021]. 2011. URL: http://curia.europa.eu/juris/document/document.jsf?docid=115202&text=&dir=&doclang=EN&part=1&occ=first&mode=DOC&pageIndex=0&cid=6196016 (cit. on p. 18).

[9] *2018 reform of EU data protection rules - Recital 71 GDPR - Profiling*. European Commission. May 25, 2018. URL: https://gdpr.eu/recital-71-profiling/ (visited on 02/23/2021) (cit. on p. 18).

[10] *Convention 108 and Protocols*. Council of Europe. URL: https://www.coe.int/en/web/data-protection/convention108-and-protocol (visited on 02/23/2021) (cit. on p. 19).

[11] *2018 reform of EU data protection rules - Art. 4 GDPR - Definitions*. European Commission. May 25, 2018. URL: https://gdpr.eu/article-4-definitions/ (visited on 02/23/2021) (cit. on p. 20).

[12] *2018 reform of EU data protection rules - Art. 1 GDPR - Subject-matter and objectives*. European Commission. May 25, 2018. URL: https://gdpr.eu/article-1-subject-matter-and-objectives-overview/ (visited on 02/23/2021) (cit. on p. 20).

[13] *2018 reform of EU data protection rules - Art. 7 GDPR - Conditions for consent*. European Commission. May 25, 2018. URL: https://gdpr.eu/article-7-how-to-get-consent-to-collect-personal-data/ (visited on 02/23/2021) (cit. on p. 22).

[14] Hong-Yen Tran and Jiankun Hu. «Privacy-preserving big data analytics a comprehensive survey». In: *Journal of Parallel and Distributed Computing* 134 (2019), pp. 207–218. ISSN: 0743-7315. DOI: https://doi.org/10.1016/j.jpdc.2019.08.007. URL: https://www.sciencedirect.com/science/article/pii/S0743731519300589 (cit. on p. 25).

[15] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. «Can Homomorphic Encryption Be Practical?» In: *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*. CCSW '11. Chicago, Illinois, USA: Association for Computing Machinery, 2011, pp. 113–124. ISBN: 9781450310048. DOI: 10.1145/2046660.2046682. URL: https://doi.org/10.1145/2046660.2046682 (cit. on p. 24).

[16] Erika, McCallister and Tim, Grance and Karen, Scarfone. *NIST Special Publication 800-122*. [Online; accessed 26-02-2021]. 2010. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf (cit. on p. 26).

[17] Sabrina de Capitani di Vimercati and Sara Foresti. «Quasi-Identifier». In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg and Sushil Jajodia. Boston, MA: Springer US, 2011, pp. 1010–1011. ISBN: 978-1-4419-5906-5. DOI: 10.1007/978-1-4419-5906-5_763. URL: https://doi.org/10.1007/978-1-4419-5906-5_763 (cit. on p. 26).

[18] P. Samarati and L. Sweeney. *Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression*. Tech. rep. 1998. URL: https://dataprivacylab.org/dataprivacy/projects/kanonymity/paper3.pdf (cit. on p. 27).

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. «Calibrating Noise to Sensitivity in Private Data Analysis». In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284. ISBN: 978-3-540-32732-5 (cit. on pp. 27, 40).

[20] P. Samarati. «Protecting respondents identities in microdata release». In: *IEEE Transactions on Knowledge and Data Engineering* 13.6 (2001), pp. 1010–1027. DOI: 10.1109/69.971193 (cit. on pp. 28–31).

[21] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. «κ-Anonymity». In: *Secure Data Management in Decentralized Systems*. Ed. by Ting Yu and Sushil Jajodia. Boston, MA: Springer US, 2007, pp. 323–353. ISBN: 978-0-387-27696-0. DOI: 10.1007/978-0-387-27696-0_10. URL: https://doi.org/10.1007/978-0-387-27696-0_10 (cit. on p. 30).

[22] Adam Meyerson and Ryan Williams. «On the Complexity of Optimal K-Anonymity». In: *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. PODS '04. Paris, France: Association for Computing Machinery, 2004, pp. 223–228. ISBN: 158113858X. DOI: 10.1145/1055558.1055591. URL: https://doi.org/10.1145/1055558.1055591 (cit. on p. 31).

[23] Khaled Emam et al. «A Globally Optimal k-Anonymity Method for the De-Identification of Health Data». In: *Journal of the American Medical Informatics Association : JAMIA* 16 (July 2009), pp. 670–82. DOI: 10.1197/jamia.M3144 (cit. on p. 32).

[24] R. J. Bayardo and Rakesh Agrawal. «Data privacy through optimal k-anonymization». In: *21st International Conference on Data Engineering (ICDE'05)*. 2005, pp. 217–228. DOI: 10.1109/ICDE.2005.42 (cit. on pp. 32, 33).

[25] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. «Incognito: Efficient Full-Domain K-Anonymity». In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. SIGMOD '05. Baltimore, Maryland: Association for Computing Machinery, 2005, pp. 49–60. ISBN: 1595930604. DOI: 10.1145/1066157.1066164. URL: https://doi.org/10.1145/1066157.1066164 (cit. on pp. 33, 53, 79).

[26] Latanya Sweeney. «Datafly: a system for providing anonymity in medical data». In: *Database Security XI: Status and Prospects*. Ed. by T. Y. Lin and Shelly Qian. Boston, MA: Springer US, 1998, pp. 356–381. ISBN: 978-0-387-35285-5. DOI: 10.1007/978-0-387-35285-5_22. URL: https://doi.org/10.1007/978-0-387-35285-5_22 (cit. on p. 34).

[27] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. «Mondrian Multidimensional K-Anonymity». In: *22nd International Conference on Data Engineering (ICDE'06)*. 2006, pp. 25–25. DOI: 10.1109/ICDE.2006.101 (cit. on pp. 35, 53, 77, 79).

[28] Vanessa Ayala-Rivera, Patrick McDonagh, Thomas Cerqueus, and Liam Murphy. «A Systematic Comparison and Evaluation of K-Anonymization Algorithms for Practitioners». In: *Trans. Data Privacy* 7.3 (Dec. 2014), pp. 337–370. ISSN: 1888-5063 (cit. on p. 36).

[29] Abou-El-Ela Hussien, Nermin Hamza, and Hesham Hefny. «Attacks on Anonymization-Based Privacy-Preserving: A Survey for Data Mining and Data Publishing». In: *Journal of Information Security* 04 (Jan. 2013), pp. 101–112. DOI: 10.4236/jis.2013.42012 (cit. on p. 36).

[30] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. «L-Diversity: Privacy beyond k-Anonymity». In: *ACM Trans. Knowl. Discov. Data* 1.1

(Mar. 2007), 3–es. ISSN: 1556-4681. DOI: 10.1145/1217299.1217302. URL: https://doi.org/10.1145/1217299.1217302 (cit. on p. 37).

[31] N. Li, T. Li, and S. Venkatasubramanian. «t-Closeness: Privacy Beyond k-Anonymity and l-Diversity». In: *2007 IEEE 23rd International Conference on Data Engineering*. 2007, pp. 106–115. DOI: 10.1109/ICDE.2007.367856 (cit. on p. 37).

[32] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Vol. 9. 3–4. Hanover, MA, USA: Now Publishers Inc., Aug. 2014, pp. 211–407. DOI: 10.1561/0400000042. URL: https://doi.org/10.1561/0400000042 (cit. on pp. 38, 41, 42).

[33] Cynthia Dwork. «Differential Privacy: A Survey of Results». In: *Theory and Applications of Models of Computation*. Ed. by Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19. ISBN: 978-3-540-79228-4 (cit. on pp. 39, 40, 42).

[34] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. «Calibrating Noise to Sensitivity in Private Data Analysis». In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284. ISBN: 978-3-540-32732-5 (cit. on pp. 40, 41).

[35] F. McSherry and K. Talwar. «Mechanism Design via Differential Privacy». In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. 2007, pp. 94–103. DOI: 10.1109/FOCS.2007.66 (cit. on p. 41).

[36] Aaron Roth and Tim Roughgarden. «Interactive Privacy via the Median Mechanism». In: *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*. STOC '10. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2010, pp. 765–774. ISBN: 9781450300506. DOI: 10.1145/1806689.1806794. URL: https://doi.org/10.1145/1806689.1806794 (cit. on p. 41).

[37] Camarda Giovanni. *personal-privacy-preserving-analytics*. https://gitlab.com/pimcity/wp2/personal-privacy-preserving-analytics. 2021 (cit. on pp. 44, 57).

[38] A. Fox and E. A. Brewer. «Harvest, yield, and scalable tolerant systems». In: *Proceedings of the Seventh Workshop on Hot Topics in Operating Systems*. 1999, pp. 174–178. DOI: 10.1109/HOTOS.1999.798396 (cit. on p. 47).

[39] Gong Qiyuan, Mohanty Amitav, and Kun Liu. *Mondrian*. https://github.com/qiyuangong/Mondrian. 2013 (cit. on pp. 53, 55, 64).

[40] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. «Fast Data Anonymization with Low Information Loss». In: *Proceedings of the 33rd International Conference on Very Large Data Bases*. VLDB '07. Vienna, Austria: VLDB Endowment, 2007, pp. 758–769. ISBN: 9781595936493 (cit. on pp. 54, 64, 65, 70).

[41] N. Holohan, S. Braghin, Pol Mac Aonghusa, and Killian Levacher. «Diffprivlib: The IBM Differential Privacy Library». In: *ArXiv* abs/1907.02444 (2019) (cit. on pp. 56, 57, 77, 80).

[42] N. Holohan, S. Braghin, Pol Mac Aonghusa, and Killian Levacher. *differential-privacy-library*. https://github.com/IBM/differential-privacy-library. 2019 (cit. on p. 56).

[43] Yoav Freund and Robert E Schapire. «A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting». In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504. URL: https://www.sciencedirect.com/science/article/pii/S002200009791504X (cit. on p. 59).

[44] Tin Kam Ho. «Random Decision Forests». In: ICDAR '95. USA: IEEE Computer Society, 1995. ISBN: 0818671289 (cit. on p. 59).

[45] J. Ross Quinlan. «Learning Efficient Classification Procedures and Their Application to Chess End Games». In: *Machine Learning: An Artificial Intelligence Approach*. Ed. by Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 463–482. ISBN: 978-3-662-12405-5. DOI: 10.1007/978-3-662-12405-5_15. URL: https://doi.org/10.1007/978-3-662-12405-5_15 (cit. on p. 59).

[46] Corinna Cortes and Vladimir Vapnik. «Support-Vector Networks». In: *Mach. Learn.* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125. DOI: 10.1023/A:1022627411411. URL: https://doi.org/10.1023/A:1022627411411 (cit. on p. 59).

[47] Evelyn Fix, J. L. Hodges, and USAF School of Aviation Medicine. *Discriminatory analysis: nonparametric discrimination, consistency properties*. Randolph Field, Tex.: USAF School of Aviation Medicine, 1985, 2 v. in 1. URL: http://hdl.handle.net/2027/iau.31858027341571%20(v.1-2) (cit. on p. 59).

[48] Koutroumbas Konstantinos and Theodoridis Sergios. *Pattern Recognition 4th Edition*. Elsevier Science, 2008, b.v. p. 203. ISBN: 9780080949123 (cit. on p. 60).

[49] Ron Kohavi. «Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid». In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 202–207 (cit. on p. 62).

[50] A. Subasi and S. Cankurt. «Prediction of default payment of credit card clients using Data Mining Techniques». In: *2019 International Engineering Conference (IEC)*. 2019, pp. 115–120. DOI: 10.1109/IEC47844.2019.8950597 (cit. on p. 62).

[51] Beata Strack, Jonathan Deshazo, Chris Gennings, Juan Luis Olmo Ortiz, Sebastian Ventura, Krzysztof Cios, and John Clore. «Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records». In: *BioMed research international* 2014 (Apr. 2014), p. 781670. DOI: 10.1155/2014/781670 (cit. on p. 62).

[52] «Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements». In: *IEEE Transactions on Systems Science and Cybernetics* 5.4 (1969), pp. 322–333. DOI: 10.1109/TSSC.1969.300225 (cit. on p. 66).

[53] Jun Han and Claudio Moraga. «The influence of the sigmoid function parameters on the speed of backpropagation learning». In: *From Natural to Artificial Neural Computation*. Ed. by José Mira and Francisco Sandoval. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 195–201. ISBN: 978-3-540-49288-7 (cit. on p. 66).

[54] Allen David M. «The Relationship Between Variable Selection and Data Agumentation and a Method for Prediction». In: *Technometrics* 16.1 (1974), pp. 125–127. DOI: 10.1080/00401706.1974.10489157. eprint: https://www.tandfonline.com/doi/pdf/10.1080/00401706.1974.

10489157. URL: https://www.tandfonline.com/doi/abs/10.1080/00401706.1974.10489157 (cit. on p. 66).

[55] Pearson F.R.S. Karl. «LIII. On lines and planes of closest fit to systems of points in space». In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720. eprint: https://doi.org/10.1080/14786440109462720. URL: https://doi.org/10.1080/14786440109462720 (cit. on pp. 66, 80).

[56] P. S. Wang, P. -Y. Huang, Y. -A. Tsai, and R. Tso. «An Enhanced Mondrian Anonymization Model based on Self-Organizing Map». In: *2020 15th Asia Joint Conference on Information Security (AsiaJCIS)*. 2020, pp. 97–100. DOI: 10.1109/AsiaJCIS50894.2020.00026 (cit. on p. 79).

[57] Nikjil Jha, Thomas Favale, Luca Vassio, Martino Trevisan, and Marco Mellia. «z -anonymity: Zero-Delay Anonymization for Data Streams». In: *To appear in the 2020 IEEE International Conference on Big Data*. 2020 (cit. on p. 79).

[58] J. MacQueen. *Some methods for classification and analysis of multivariate observations*. English. Proc. 5th Berkeley Symp. Math. Stat. Probab., Univ. Calif. 1965/66, 1, 281-297 (1967). 1967 (cit. on p. 80).