Natural Language Processing in Industry - Anonymization of Personally Identifiable Information

Pablo Ortega Sanchez
University Heidelberg
ort.san.pablo@gmail.com

Abstract

The General Data Protection Regulation (GDPR) requires companies and researchers to find methods to use personal data without violating the regulation. Data anonymization approaches aim to keep the semantics of the original data for analytical reasons, while anonymizing private information and unique identifiers. The main benefit of such approaches is that anonymized data is no personal information according to the GDPR (EU, 2016) and can be used without the consent of the people the data originally belonged to as long as the anonymization process meets the GDPR's requirements.

1 Introduction

In machine learning, one of the most important ingredients is data. Whether it is weather data from weather stations used to forecast the weather, or house prices, or information about one's last doctors appointment, there probably is or will be a machine learning architecture that tries to approximate that data in order to learn something new. With the rise of big data companies and widely used location-tracking through apps and smartphones, data is gathered increasingly and used to generate monetary value. Unfortunately it is often unclear how this data is used and if any of it is personal and so privacy concerns come to light. The GDPR prohibits the processing of personal data, except for when it is allowed by law or the owner of such personal data has consented. (EU, 2016) According to the GDPR

> "'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;" (EU, 2016)

Since the GDPR specifically speaks of "any information" relating to a "data subject" one could argue that this is a very broad statement, which can yield very restrictive interpretations as to what can be considered personal. One must also consider that, depending on the task at hand, specific data can be crucial. This is where pseudonymization comes into play, which only masks personally identifiable data with pseudonyms. However, this pseudonymized data often can be reproduced using additional information, e.g. through another data set. The balance between protecting the data subject's right to privacy and producing useful data is vague and different in many cases. Anyone gathering, processing or releasing data must always be able to justify the amount of data and its types and also consider any adversarial attacks that could occur. At no time during the gathering, processing and after a release personal data should be accessible by any means. So during an anonymization process one has to spend resources to evaluate the irreversibility of said process.

The simplest solution to this regulation would be to just delete all the data but of course that would defeat the whole purpose of collecting it. Unfortunately, other solutions are much more complex. It's amazing how much data can tell about people, even when there are no obvious connections. But before we take a look at such connections we should get a better overview of a few terms.

2 Anonymization and Pseudonymization

Anonymized data must not include any personal data or data that can be used to reference personal data. One must consider that removing rows or columns in order to anonymize lessens the amount of information that is contained in the data and can make the data useless for machine learning tasks. Fortunately there are more methods to anonymization than simply removing data.

A different approach is pseudonymization, which masks personally identifiable information with pseudonyms. This keeps the data integrity which is preferable for machine learning and is an "appropriate technical and organizational measure", according to the GDPR (EU, 2016). Taking a look at some negative examples will show why regulations like the GDPR are necessary. Still, that does not inherently mean that the GDPR is good or bad. However, one should look at whether the GDPR is sufficiently strict.

Identifiers: It can be hard to understand why the definition of personal data is so broad. Of course it is reasonable to remove or mask information like names, addresses or credit card numbers, but also ids, social security numbers or custom email domains. Intuitively, these data points can be used to identify persons and connect them to any additional information in a given data set. But what about postal codes, gender, date of birth; information, that, on its own would not identify a person? It turns out that the real danger lies in potential adversaries combining data, even from different sources. Such data often are called quasi identifiers, which simply means that they are not identifying on their own but can be when combined. It is important to note that given the right circumstances almost anything could be a quasi identifier. And it is the duty of anyone working with data to consider any measures adversaries could reasonably take and to take security measures. Data that should be included in anonymization efforts includes login details, time zones, plug-in details, device type, browser type, cookies, IP-addresses but also untranslated texts and language preferences. Sensitive data is anything that should not be revealed, but is not identifying on its own, such as diseases or salary.

Looking at Golle (2006) and Sweeney (2000) between 63% and 87.1% of United States citizens were likely to be identified by their 5-digit ZIP, gender and date of birth. The number drops significantly, to 3.7%, when replacing the date of birth with just the month and year of birth. The percentage can be lowered to 0.00000%(but still not 0) by replacing the 5-digit ZIP by county and the date of birth by a 2 year age range. In the following paragraphs methods will be explained that will provide a measurable amount of anonymity.

k-anonymity: The goal of k-anonymity is to mask and remove data, and change the identifying attributes by which data is grouped until there are k-1 identical records for every record. These identical records are called an equivalence group and share the same values in identifying attributes. This assures that the maximum likelihood of a record being identified is $1/k$. The above example using United States citizens found that the median anonymity for people younger than 50 is k=200 given the gender, ZIP-code and year of birth. Replacing the ZIP-code with the county produces a k=3000 anonymity - this is called generalization (Golle, 2006).

Generalization can be used on a variety of data. Age can be grouped together, the same goes for diseases, professions, ZIP-codes and even gender.

If there is specific attributes in a data set, e.g. personally identifiable information one can also opt to suppress information. This means to completely remove whole records or specific attributes and can drastically reduce the information contained in a data set and thus the usability in a given task, such as statistics or machine learning.

Table 1 shows an unanonymized, fictitious data set. In order to remove personally identifiable information one can begin by suppressing names and also group age ranges together. Taking a look at table 2 shows that k-anonymity is still not achieved. There is only one record in the age range 11-20. One could now generalize further, or even suppress whole records. Table 3 has the outlier removed. The table is now k=2 anonymized for attributes age range, gender and state of domicile, the disease attribute is sensitive, but also no identifier. Finally, table 4 shows k=2 anonymity with the sensitive disease attribute included. Most of the information that was contained in the data set is lost, making it probably useless for further tasks. Given the right circumstances the knowledge of all the records in table 4 being identical could be combined with new information and reveal personal information, which might make table 4 more vulnerable to outside attacks than table 3. Of course this is just a fictional

Name	Age	Gender	State of domicile	Disease
Jarno	26	Male	Hawii	Viral infection
Barbara	23	Female	Hawii	Cancer
Kim	29	Male	Michigan	Tuberculosis
John	19	Male	Michigan	Viral infection
Juan	25	Male	Hawii	Heart-related
Sunny	27	Male	Michigan	No illness
Sabrina	28	Female	Alabama	Tuberculosis
Yasmin	24	Female	Hawii	Viral infection
Rahdni	30	Female	Alabama	Cancer

Table 1: Unanonymized data set

Age range	Gender	State of domicile	Disease
21-30	Male	Hawii	Viral infection
21-30	Female	Hawii	Cancer
21-30	Male	Michigan	Tuberculosis
11-20	Male	Michigan	Viral infection
21-30	Male	Hawii	Heart-related
21-30	Male	Michigan	No illness
21-30	Female	Alabama	Tuberculosis
21-30	Female	Hawii	Viral infection
21-30	Female	Alabama	Cancer

Table 2: Suppressed names + age ranges in groups

example, but considering that we track and store more data than ever before it surely is just a matter of time before the next breach or deanonymization happens. In the case of this small data set, size was a considerable problem and one must keep in mind that given a large enough data set high k-values are possible. Golle (2006), for example, examined the United States citizens data with k-values in the hundreds, sometimes over 10000.

l-diversity: Another method of anonymization is l-diversity, which helps further in keeping sensitive information unpairable with personally identifiable information. It's essentially what can be observed between table 3 and table 4. For each pair of data, that share the same nonsensitive attributes in order

Age range	Gender	State of domicile	Disease
21-30	Male	Hawii	Viral infection
21-30	Female	Hawii	Cancer
21-30	Male	Michigan	Tuberculosis
21-30	Male	Hawii	Heart-related
21-30	Male	Michigan	No illness
21-30	Female	Alabama	Tuberculosis
21-30	Female	Hawii	Viral infection
21-30	Female	Alabama	Cancer

Table 3: Suppressed names + age ranges in groups, outlier removed. 2-anonymity for Age range, Gender and State of domicile attributes

Age range	Gender	Disease
21-30	Female	Cancer
21-30	Female	Cancer

Table 4: k=2, at loss of most information

to achieve k-anonymity, there should be a diverse representation of sensitive attributes with the same frequency. According to Machanavajjhala et al. (2006) l should be at least 2, which holds true in table 3: 21-30, Male, Hawaii has either a viral infection or heart-related disease. 21-30, Female, Hawaii has either cancer or a viral infection. 21-30, Male, Michigan has either tuberculosis or no illness and 21-30, Female, Alabama has either tuberculosis or cancer. While an adversary that knows a person is listed in table 4 can deduce that that person has cancer. This is of course a very minimal example.

t-closeness In 2007 Li et al. (2007) argued that l-diversity

> "is neither necessary nor sufficient to prevent attribute disclosure"

and introduced their privacy approach t-closeness. Each equivalence group should have a distribution of all sensitive attributes t-close to the distribution in the full data set. T is a threshold that describes the maximum distance between the two distributions.

Differential Privacy Differential privacy is not so much a method like the preceding paragraphs but a goal that can be reached. The goal is that no person is ever negatively affected by providing data even when taking all other potential data or information into account. A differentially private database tries to provide meaningful information about a group without revealing individual information. This further means that conclusions drawn from a data set should not stand or fall because of the inclusion of an individual. Differential privacy even argues that an individual's information was not leaked even if conclusions further affect it. Medical databases may suggest to insurance companies that lifestyle choices such as smoking or increased consumption of alcohol warrant a rise in insurance premiums. While that may affect an individual one can not call this a privacy breach (Dwork and Roth, 2014).

Algorithms are differentially private if its output over a data set does not reveal whether an individual was included or not. The output changes barely when any record is added or suppressed no matter how special any one's data may be. A simple example for such an algorithm is the computation of a mean or median. An algorithm that counts specific attributes, such as "how often did red win?" and then adds some random noise so that the goals specified above are still met is a more complex example(Harvard).

In addition, the nature of differential privacy is aligned with the concept of machine learning. Learn the characteristics and rules of data from its distribution and not from single data points and then generalize so that new data from the distribution can be correctly described.

3 Data Breaches

When anonymization efforts fail, potential adversaries may be able to reproduce personally identifiable data. This happened on multiple occasions. Since 2009 the Taxi and Limousine Commission New York City (TLC) releases annual data sets containing information on trips taken by cabs in New York City (TLC, 2009-2019). Such data sets are released to analyze and even improve public policies. This particular data set includes pickup locations and destinations, distance, tip and fare, as well as pseudonymized medallion and license numbers. The TLC followed guidelines to withdraw personally identifiable information from the data before releasing it, answering a request of the Freedom of Information Law of New York City. This law prohibits the TLC from publishing incorrect data and also requires the publication of any modification on the data.

Combining this data with metadata of paparazzi pictures of celebrities made it possible to track taxi trips taken by said celebrities, revealing start and destination, fare and tip of those data subjects; a breach in privacy (Trotter, 2014).

Douriez et al. (2016) - following Vijay Pandurangan, who swiftly was able to reverse the anonymization efforts of the TLC by reversing the MD5 hashing of medallion and license numbers (Pandurangan, 2014)- raised the question whether anonymization is possible considering that adversaries may use additional data to attack data sets. In cooperation with the TLC they evaluated the case and concluded that using privacy preserving techniques is not enough, as their own attack approach was able to "identify a significant fraction of the taxis". Since, the TLC removed the medallion and license numbers from new data sets but one must keep in mind that this data set is only one of many.

In 2013 de Montjoye et al. (2013) analyzed cellphone data of 1.5 million people and found that four spatio-temporal points, so time and place were enough to uniquely identify 95% of the individuals.

In 2006 Netflix released supposedly anonymized data containing a subscriber ID, movie titles, year of release and date on which the subscriber rated the movie. Only days later Narayanan and Shmatikov (2006) revealed that they were able to identify users by cross-referencing the provided data with data from imdb.com, "uncovering their apparent political preferences and other potentially sensitive information".

This raises some questions about the validity of anonymization methods. If it is not possible to actually guarantee the security of private data, even if every effort is made to do so - and the usability of the data should not be reduced to zero - how can individuals still be protected? This is the question that the EU is certainly constantly asking itself and is trying to solve by regularly updating the GDPR.

4 Python Examples

Now we will look at easy examples of code that implement some degree of privacy. Unfortunately, the "Anonymization Tools" provided are no means of achieving k-anonymity, l-diversity, t-closeness or differential privacy, which all require heavy computation and smart algorithms but its functions include name pseudonymization - which could be expanded to social security numbers or credit card numbers, but most of the time these identifiers should be suppressed completely, symmetric encryption, suppression, numerical encoding and generalizing.

The first cell of the python notebook contains imports. We will use pandas and numpy for simple operations, clean_pandas for encryption, Faker for pseudonymization and DataFrameMapper in combination with LabelEncoder to numerically encode. In the second cell of the python notebook the data is loaded and parameters are introduced. The way it is set up one can simply insert the column names into the corresponding task parameters and the operations will be executed by running the remaining cells of the notebook.

	R_fighter	B_fighter
0	Robert Whittaker	Darren Till
1	Mauricio Rua	Rogerio Nogueira
2	Fabricio Werdum	Alexander Gustafsson
3	Carla Esparza	Marina Rodriguez
4	Paul Craig	Gadzhimurad Antigulov
...	...	...
4302	Duane Ludwig	Darren Elkins
4303	John Howard	Daniel Roberts
4304	Brendan Schaub	Chase Gormley
4305	Mike Pierce	Julio Paulino
4306	Eric Schafer	Jason Brilz

4307 rows × 2 columns

Figure 1: Data before Pseudonymization

Pseudonymization After taking a look at the test data set ”ufc-master.csv” one can see that the columns ”R_fighter” and ”B_fighter” (figure 1) contain names of athletes. We can then declare that ”name_cols = [’R_fighter’, ’B_fighter’]”. The first important function that is provided is ”anon_df”. This will take a data set, a list of columns and a function name and then apply said function to each passed column of the data set. The functions ending on ”pseud” look whether a value is in its dictionary. If it is, that means that a pseudonym has already been generated and it will be inserted into the data field again. If it is not, a new pseudonym will be generated and both value and pseudonym will be paired. A pseudonymized table can be found in figure 2. The ”pseud” functions can be expanded, by copying and pasting it and replacing the Faker providers like ”fake.name()” with an other like fake.address(). This will then generate a fake address if called.

Symmetric Encryption Now lets say that we want to encrypt. For this example we will encrypt the location of our data set ”ufc-master”. It is debatable whether that is a useful step towards anonymization, but it will be sufficient as demonstration. Our initial data set can be found in figure 3. We will then declare ”encr_cols = [’location’]”. In the sixth cell where all the transformations happen we make use of Clean Pandas’ ”encrypt()” function

	R_fighter	B_fighter
0	Jennifer Blankenship	Luis Clark
1	Laura Shepherd	Jill Patterson
2	Dennis Miller	Kayla Santos
3	Jason Morrison	Cathy Schmidt
4	Erica Orozco	Phillip Nash
...	...	...
4302	Danielle Turner	Michael Knight
4303	Hannah Thomas	Casey Martinez
4304	Jacqueline Berry	Michael Maxwell
4305	Phillip Vargas	Brian Rosales
4306	Jeffrey Ramirez	Kristen Myers

4307 rows × 2 columns

Figure 2: Data after Pseudonymization

	location
0	Abu Dhabi, Abu Dhabi, United Arab Emirates
1	Abu Dhabi, Abu Dhabi, United Arab Emirates
2	Abu Dhabi, Abu Dhabi, United Arab Emirates
3	Abu Dhabi, Abu Dhabi, United Arab Emirates
4	Abu Dhabi, Abu Dhabi, United Arab Emirates
...	...
4302	Broomfield, Colorado, USA
4303	Broomfield, Colorado, USA
4304	Broomfield, Colorado, USA
4305	Broomfield, Colorado, USA
4306	Broomfield, Colorado, USA

4307 rows × 1 columns

Figure 3: Data before Encryption and after Decryption

	location
0	b'gAAAAABfJC_4r4kNuDG7PYtkhWzF0sZPNmKRx096Dln5...
1	b'gAAAAABfJC_4r4kNuDG7PYtkhWzF0sZPNmKRx096Dln5...
2	b'gAAAAABfJC_4r4kNuDG7PYtkhWzF0sZPNmKRx096Dln5...
3	b'gAAAAABfJC_4r4kNuDG7PYtkhWzF0sZPNmKRx096Dln5...
4	b'gAAAAABfJC_4r4kNuDG7PYtkhWzF0sZPNmKRx096Dln5...
...	...
4302	b'gAAAAABfJC_41mu2hUf9YOirYic-4lCEd-5cMQUgY9Rp...
4303	b'gAAAAABfJC_41mu2hUf9YOirYic-4lCEd-5cMQUgY9Rp...
4304	b'gAAAAABfJC_41mu2hUf9YOirYic-4lCEd-5cMQUgY9Rp...
4305	b'gAAAAABfJC_41mu2hUf9YOirYic-4lCEd-5cMQUgY9Rp...
4306	b'gAAAAABfJC_41mu2hUf9YOirYic-4lCEd-5cMQUgY9Rp...

4307 rows × 1 columns

Figure 4: Data after Encryption

	weight_class
0	Middleweight
1	Light Heavyweight
2	Heavyweight
3	Women's Strawweight
4	Light Heavyweight
...	...
4302	Lightweight
4303	Welterweight
4304	Heavyweight
4305	Welterweight
4306	Light Heavyweight

4307 rows × 1 columns

Figure 5: Data before numerical encoding

to compute an ”encryption_key” for later decryption, a ”dtype_dict” that stores the data type of the columns that are passed through and the encrypted data. The encrypted data uses Fernet, which is symmetrical encryption(Burgess, 2018). The resulting data can be found in figure 4. Note that identical values in the source data, such as ”Broomfield, Colorado, USA” receive identical values in the target data. We can now use Clean Pandas’ ”decrypt()” function by passing it the ”encr_cols, the ”encryption_key” and ”dtype_dict” to decrypt.

Suppression Suppression of attributes means to remove those attributes from the data. In this case we will suppress the ”date” attribute from our data. For this we simply declare ”drop_cols = [’date’]” and run the notebook. The command ”result_df.drop(columns=drop_cols, inplace=True)” will then remove all declared columns. In practice, suppression can also mean the deletion of single individuals, if they could otherwise be easily identified by their unique attribute values.

Numerical Encoding The concept of numerical encoding is to hide the true identity of an attribute. As example we can encode the ”weight_class” attribute numerically, so that the true values are hidden. This of course doesn’t mean that it is impossible to find the true values by cross-referencing. We declare ”enc_cols = [’weight_class’]” and ”encode_cols = [([’weight_class’], LabelEncoder())]”. The data can be found in figure 5. In lines 2-4 of the sixth cell the encoding is executed. First in line 2 the ”DataFrameMapper” is initialized and provided with the columns to encode and the ”LabelEncoder”. In line 3 the ”DataFrameMapper” then writes a new column, that has its values numerically assigned by the ”LabelEncoder”. Line 4 concatenates the new column to the data set and drops the original columns. The numerically encoded column will always be the last column in the data set. Figure 6 shows the numerically encoded ”weight_class”.

Generalizing What generalizing means is that values can be grouped together to be not as specific. The most common example is age. Instead of displaying the age one can group it into intervals to achieve a reasonable amount of privacy. This can often be seen in released data sets and statistics. Figure 7 shows our example data. We can generalize the ”R_odds” attribute of our data set by declaring ”gen_cols = [’R_odds’]”. My generalization function will then take the highest and lowest values per attribute as a span and then generate ten boundaries in between this span. Each value of the attribute will then be assigned to its corresponding interval resulting in figure 8. Note that the intervals are dependant on the original values of the attribute. This means that depending on the representation in the data, the selected intervals are not always useful. A solution is attribute wise declaration of a data generalization hierarchy, which can be described as different stages of generalization. For the example age the first stage is the original value, e.g. ”27”. The second stage could be ”25-30”. The third stage could be ”21-30” and so on. This is not

	weight_class
0	7
1	5
2	4
3	12
4	5
...	...
4302	6
4303	8
4304	4
4305	8
4306	5

4307 rows × 1 column

Figure 6: Data after numerical encoding

	R_odds
0	-130
1	-190
2	260
3	145
4	-137
...	...
4302	-155
4303	-210
4304	-260
4305	-420
4306	140

4307 rows × 1 columns

Figure 7: Data before Generalization

	R_odds
0	(-325.0, -50.0]
1	(-325.0, -50.0]
2	(225.0, 500.0]
3	(-50.0, 225.0]
4	(-325.0, -50.0]
...	...
4302	(-325.0, -50.0]
4303	(-325.0, -50.0]
4304	(-325.0, -50.0]
4305	(-600.0, -325.0]
4306	(-50.0, 225.0]

4307 rows × 1 columns

Figure 8: Data after Generalization

implemented in my code.

5 ARX

While researching data anonymization I came across ”ARX”, an open source data anonymization tool. The functions of this tool are far greater than the functions of my ”Anonymization Tools”. Every attribute can be tagged as ”insensitive”, ”sensitive”, ”identifying” and ”quasi-identifying”. There are options to declare data generalization hierarchies for every attribute as well as weights for every attribute to optimize for. But what are we optimizing exactly? ARX supports a wide range of privacy models, such as k-anonymity, l-diversity, t-closeness and differential privacy. It is also possible to set a suppression limit and a balance between suppression and generalization.

After ARX is configured one can begin the anonymization process. After it is complete there are a variety of options the user can now take. It is possible to review the transformation process of the tool and even select different transformations that would yield the same result. It is also possible to analyze the new data set with regards to potential adversarial attacks, information loss and a lot more.

6 Conclusion

Finally, anonymization is a process that is very necessary and it is good that the EU published its own guidelines on what it deems anonymous. I have

learned that common sense is not enough to see the connections that ultimately make personal data visible. The combination of several data sets from different sources showed me more than once that it is very difficult for me to imagine what complex connections and conclusions are possible. This problem, namely that whenever you are dealing with data you also have to assess the steps of possible adversaries, makes it in my opinion an almost impossible task to achieve anonymization without losing the statistical value of the data. This is especially true when you realize that once data has been released the "security" can no longer be increased, while the work of the actual adversaries is just beginning. However, prohibiting the storage and processing of data cannot be a solution, but rather the continuous improvement of security standards, such as the GDPR, together with the exchange of experts and the recognition of the methods of adversaries.

With regard to machine learning I think that the basic concepts of privacy and machine learning do not collide. It is true that more accurate data can also lead to more accurate predictions, but with the concept of differential privacy, I think a middle way can be found. It should not be forgotten that the fact that more and more data is being collected, combined, published and processed than ever before suggests that breaches of privacy as described in this report are likely to become more frequent and affect more and more people. Hopefully, the consequential damages for individuals will be minimal.

References

Aaron Burgess. 2018. Clean pandas.

Marie Douriez, Harish Doraiswamy, Juliana Freire, and Cláudio T. Silva. 2016. Anonymizing nyc taxi data: Does it matter?

Cynthia Dwork and Aaron Roth. 2014. *The Algorithmic Foundations of Differential Privacy*, volume 9.

EU. 2016. General data protection regulation. 2016/679. Current version of the OJ L 119, 04.05.2016; cor. OJ L 127, 23.5.2018.

Philippe Golle. 2006. Revisiting the uniqueness of simple demographics in the us population.

Differential Privacy Group Harvard. Differential privacy.

Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity.

Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. 2006. l-diversity: Privacy beyond k-anonymity.

Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. (3).

Arvind Narayanan and Vitaly Shmatikov. 2006. Robust de-anonymization of large sparse datasets.

Vijay Pandurangan. 2014. On taxis and rainbows lessons from nyc's improperly anonymized taxi logs.

Latanya Sweeney. 2000. Simple demographics often identify people uniquely. (Data Privacy Working Paper 3).

New York City TLC. 2009-2019. Tlc trip record data.

J.K. Trotter. 2014. Public nyc taxicab database lets you see how celebrities tip.