Personal Data De-identification for Data Science: Challenges, Methodologies, and Best Practices

Authored by: Halyna Oliinyk

1touch.io

Introduction

Terms like ‘sensitive data’ and ‘personal data’ have been floating in the air ever since GDPR, CCPA, and similar privacy acts were introduced to companies across the globe. One challenge they present is that the complexity of the federal laws and complicated terminology used to identify the corresponding subjects make it difficult for those in the technical field to truly grasp. It becomes harder than ever for the data scientists to figure out the main challenges of processing datasets containing sensitive information and how the data should be anonymized properly.

The main idea behind these regulations is the need to protect the data subjects’ rights. One method is to not save any data that is not necessary for your business uses. Another objective is to protect data from possible breaches, which unfortunately has been happening quite often to the world’s biggest companies (such as the recent British Airways data breach). In terms of the development of machine-learning algorithms to analyze possibly sensitive datasets, **no one actually needs real personal data to create a functioning data science pipeline**[1].

After researching this topic and learning the reasons behind it, it seems that the highest priority that comes to an engineer’s mind is the need to anonymize potentially sensitive data to avoid the possibility of sensitive data leakage. Another potential problem is that even partially ‘anonymized’ datasets that don’t have any kind of personal data can reveal personal information when under an effective attack. Here’s why:

Possible Resources of the Data Breach

The presence of personally identifiable information (PII)[2]. As its name would give away, by using this data we can uniquely identify the person (e.g. passport ID, national ID, tax ID). When performing any type of anonymization (anonymization types will be given in more detail later) this data is often removed or replaced with a form random strings.

Sensitive Information[3]. This information doesn’t reveal any personal data, but contains the data about the person, which should be protected (e.g. HIV status);

Quasi-Identifiers (QI)[4]. These records also don’t reveal PII on their own but combined with other information can be used to uniquely identify a person. For instance, ZIP code can not identify a person on its own, but the combination of state, gender and ZIP code can do it.

Standard Ways to De-identify Personal Data

Usually, maintainers of the database try to eliminate all channels that could potentially help an attacker leverage queries to gain personal and sensitive information about a specific person.

Here are a few examples[5]:

Pseudonymization: This method of processing of personal data is based on replacing the values, which contain personal information, with pseudorandom strings. De-identified data is stored separately from the ‘additional information’, which doesn’t contain any kind of personal/sensitive information,

making the data identifiable only when both elements are together. In practice, one 'real' sensitive value corresponds to one pseudorandom value, ensuring that analytical correlations are still possible. Because of this transitive dependency of 'real' value and 'random' value, cryptographic methodologies are often used (hash functions like SHA-512). This ensures that the attacker, who doesn't have access to the secret key, can't decrypt the pseudonymized values.

Anonymization: The main difference between pseudonymization and anonymization is that by using rules or cryptographic algorithms (pseudonymization), there is still a pathway to retrieve sensitive/personal data from the de-identified information. With anonymization, however, there's no going back (as long as it is an irreversible removal of any information that could lead to the individual being identified). Just as with pseudonymization, anonymized data should be stripped away from any kind of identifiable information.

Suppression: This technique is quite similar to the previous one, but instead of replacing sensitive/personal data with the random strings, it is replaced with hard-coded sequences, such as '***'. Suppression is also called data masking, and as with anonymization, there's no way to retrieve the original values.

Encryption: Let's compare encryption to pseudonymization. While they both use the same algorithm, the difference is pseudonymization uses a secret key to produce pseudorandom values. Additionally, encryption is regulated by GDPR because the encryption strength is expected to be good enough: controllers are required to implement risk-based measures to protect data security.

No kind of machine learning can be performed on the data, which is anonymized using previously described techniques. The reason for it is obvious: all of the features, which have some kind of information gain, are removed from the data. However, this doesn't mean that data scientists should use the datasets containing personal information, even though there are a variety of approaches to make meaningful computations on the de-identified datasets which don't reveal sensitive information at the same time.

Machine Learning Way to De-identify Personal Data

Homomorphic Encryption[6]: The main idea behind homomorphic encryption is that the inferences we make based on computations of encrypted data should be as accurate as if we had used decrypted data. Homomorphic encryption is an evolving field, and at this point in time has certain limitations. For example, only polynomial functions can be computed and only additions and multiplications of integers modulo-n are allowed. Most mathematical operations, which are used even in the simplest neural networks are not allowed when performing model training with homomorphically encrypted data. As you can understand, the final concepts of this methodology are still being developed.

The main idea behind homomorphic encryption is that we don't need to remove any kind of values from the dataset, or mask/anonymize personal data in any way. However, as of the time of writing, there is

not enough practical evidence to state that they can be used for the production-level methodologies; furthermore, there are not so many functional homomorphic encryption pipelines.

Let's imagine a situation where we've removed all personal data from the dataset (or anonymized and stored it separately from other values). Most likely, even after removal of the personal data, QIs are still left in the database.

The biggest problem of storing quasi-identifiers is that when enduring an attack on the database, it isn't all that difficult to combine QI values with other open data sources and reveal the identity of the person together with their personal/sensitive information. A good example of that is when the Netflix Prize competition open data was combined with IMDB's movie ratings dataset: entire movie-watching history of individuals was compromised.

As a result of datasets, insecure data science pipelines, which make predictions using datasets and QIs, potentially sensitive/personal information can be revealed even after the personal/sensitive data itself has been removed. We need to make sure that no queries that have the potential to reveal individual personal information that can be leveraged. Furthermore, we must make sure that no inference on the data subject can be made by running multiple predictions using machine learning algorithms.

Standard Ways to Process Datasets with QI Values

K-anonymity[7]: This approach is quite different from the one that I described earlier. With K-anonymity, we are not aiming to 'hide' any data, but rather are softly 'masking' the QI values. The most popular techniques used in k-anonymity are purging and generalization. Purging simply replaces QI values with random strings like '-' (similar to suppression). Generalization doesn't remove QI values completely but replaces them with ranges instead of set numbers(e.g. 20-30 years old). The main goal of k-anonymity is to provide a guarantee that any arbitrary query on a large dataset will not reveal information that can help narrow a group down below a threshold of 'k' individuals. Strictly speaking, 'k-anonymity' ensures that all possible equivalence groups of a dataset have at least 'k' records (equivalence groups are the subsets of datasets, which have the same value for one or more QIs). For instance, a 3-anonymity dataset ensures that for each query that a potential attacker can perform, we will have at least 3 individuals, which cannot be distinguished based on the QI values.

l-diversity: Unfortunately, k-anonymity techniques may still be subject to attacks, which is usually because each of the equivalence groups may not have attribute diversity. A rare case for this is when all QI records of the equivalence group are the same, enabling the attacker to easily make an inference. l-diversity makes sure that there is enough diversity among QI records in each of the possible equivalence groups.

T-closeness: When speaking about the distributions, which are created by purging

and generalization techniques, it is worth noting that the distributions of data in the equivalence groups should be similar to the distributions in the whole dataset. Specifically, the difference should not be bigger than the pre-specified value 't'. Earth Mover's distance is used to measure the distance between the distributions.

One may learn that preserving these rules, which are defined by l-diversity, k-anonymity and t-closeness can cause complex combinatorial problems. At this point, machine learning techniques become quite useful as long as they can operate data in separate hyperplanes and perform computations there, which can be very complex tasks if you are using the approaches described earlier.

Machine learning methods to process datasets with QI values

Differential Privacy (DP)[8]: This mathematical framework gives the ability to control to what extent the model 'remembers' and 'forgets' potentially sensitive data, which is its big advantage. The most popular concept of DP is 'noisy counting', which is based on drawing samples from Laplace distribution and using them to make the dataset represent augmented values, not the real one. However, the main disadvantage of Differential Privacy is the potential for the attacker to estimate the actual value from the repeated queries. Predictions made by using different private datasets are accurate enough, but with each new query made by the attacker, more and more sensitive information is getting released.

Federated Learning[9]: The core idea of federated learning is very similar to distributed learning, because we're not trying to train our model with all of the data at once, but instead are training it on subsets of it. This is quite a powerful method as long as we can effectively train and improve the model on separate devices while holding different subsets of data and gradually improve it.

'Private Aggregation of Teacher Ensembles' (PATE): This framework uses pieces of the different privacy methods, which is storing personal/sensitive data in a way that doesn't reveal any kind of individual personal information. The core idea of PATE is that if two models trained on separate data agree on some outcome, it is less likely that sharing the outcome to the consumer will leak any sensitive data about a specific user. Training methodology is quite similar to federated learning (and bagging techniques, of course) because at the first step we need to split our dataset into smaller subsets and then train different models on them. Predictions are made by aggregating all of the predictions from different models and injecting noise into them.

Another important feature of PATE is that we're continuously training our downstream 'student' model using this 'noisy' data and finally showing the user not the 'teacher' models, but rather the 'student' ones, which ensures that sensitive/personal data is not revealed during inference phase.

Conclusion

As I indicated in this whitepaper, the main finding I would like to emphasize is that there are many ways to eliminate personal data from the dataset while performing a powerful data science analysis. Evolving fields like homomorphic encryption and differential privacy are getting continuously developed by a big community of researchers. Already, it is possible to process a fully anonymized dataset and at the same time extract the relevant features from it without compromising personal data.

At 1touch.io we are using all of the aforementioned techniques to make sure that we're using only synthetic data, which contains only fictional personal entities that can't specifically identify real individuals.

References

[1] duncangreavesblog. (2019, April 28). 9 Reasons Smart Data Scientists Don't Touch Personal Data. Retrieved from https://informationwithinsight.com/2019/04/28/9-reasons-smart-data-scientists-dont-touch-personal-data/?utm_source=datafloq&utm_medium=ref&utm_campaign=datafloq

[2] CFR § 200.79 - Personally Identifiable Information (PII). Retrieved from https://www.law.cornell.edu/cfr/text/2/200.79

[3] What personal data is considered sensitive? Retrieved from https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/legal-grounds-processing-data/sensitive-data/what-personal-data-considered-sensitive_en

[4] Arvidson, K., & Bladh, P. (2017, January 30). Quasi identifiers and the challenges of anonymising data. Retrieved from https://www.basalt.se/quasi-identifiers-challenges-anonymising-data/

[5] Teska, A. Pseudonymization, Anonymization, Encryption ... what is the difference? Retrieved from https://teskalabs.com/blog/data-privacy-pseudonymization-anonymization-encryption

[6] Michele Minelli. Fully Homomorphic Encryption for Machine Learning. Computer Science [cs]. PSL University, 2018. English. Fftel-01918263

[7] Seidl, T. Chapter 8: Privacy Preserving Data Mining. Retrieved from https://www.dbs.ifi.lmu.de/Lehre/KDD/SS16/skript/8_PrivacyPreservingDataMining.pdf

[8] Elamurugaiyan, A. (2018, August 31). A Brief Introduction to Differential Privacy. Retrieved from https://medium.com/georgian-impact-blog/a-brief-introduction-to-differential-privacy-eacf8722283b

[9] Bhattacharya, S. A Beginners Guide to Federated Learning. Retrieved from https://hackernoon.com/a-beginners-guide-to-federated-learning-b29e29ba65cf