Exploring Differential Privacy in Practice

Davi Grossi Hasuda and Juliana de Melo Bezerra
Computer Science Department, ITA, São José dos Campos, Brazil

Keywords: Privacy, Differential Privacy, Classification Algorithms, Accuracy, Data Analysis.

Abstract: Every day an unimaginable amount of data is collected from Internet users. All this data is essential for designing, improving and suggesting products and services. In this frenzy of capturing data, privacy is often put at risk. Therefore, there is a need for considering together capturing relevant data and preserving the privacy of each person. Differential Privacy is a method that adds noise in data in a way to keep privacy. Here we investigate Differential Privacy in practice, aiming to understand how to apply it and how it can affect data analysis. We conduct experiments with four classification techniques (including Decision Tree, Näive Bayes, MLP and SVM) by varying privacy degree in order to analyze their accuracy. Our initial results show that low noise guarantees high accuracy; larger data size is not always better in the presence of noise; and noise in the target does not necessary disrupt accuracy.

1 INTRODUCTION

It is noticeable that data is an important asset of the globalized world. Every moment, a lot of new data is being generated by users of the Internet worldwide, which is actually useful for many companies that invest in storing and processing all the data they can collect (World, 2018). Amazon.com, for instance, developed its Recommender System by searching for users with similar interests, and made suggestions based on this similarity (Smith and Linden, 2017). Generally, for companies to understand how the user experience is evolving with the product or service, they have to collect user data (Havir, 2017). Another important factor of the globalized world is the fact that some of this data is used to train neural networks, and very often this training requires a great amount of data. Face ID, for example, the Apple's system to recognize someone's face and authenticate based on that, took over 1 billion images to train its neural network (Apple, 2017b).

In the midst of the frenzy of collecting data, many times privacy is jeopardized (Buffered.com, 2017). One of the most emblematic case was the scandal involving Facebook and Cambridge Analytica (Granville, 2018), where Facebook provided the data and Cambridge Analytica used it improperly to influence the presidential run in the United States. Another example was with Tanium, a cybersecurity startup, that exposed the network of a client without permission (Winkler, 2017). Concerned about these privacy scandals, there are some efforts emerging in order to preserve privacy. GDPR, for instance, is the General Data Protection Regulation (GDPR, 2018) from the European Union that rewrites how data sharing must work on the Internet. GDPR describes constraints and rules when accessing and sharing user data. Another effort, which is the focus of our paper, is Differential Privacy (DP)(Dwork, 2006). DP establishes constraints to algorithms that concentrate data in a statistical database. Such constraints limit the privacy impact on individuals whose data is in the database.

Simple anonymization processes can be very ineffective for assuring privacy. For example, there is the case of 2006 Netflix Prize, a competition promoted by Netflix where competitors must develop an algorithm to predict ratings from users. For that, Netflix shared a dataset with over 100 million ratings by over 480 thousand users. All the names were removed, and some fake ratings were added. But as shown later, it was not enough, since a de-anonymization process was possible by comparing the Netflix dataset with an IMDb dataset (Dwork and Roth, 2014). So, the process that privatize data must be linkage attack-proof. Besides, it must not compromise the final result of machine learning algorithms and statistical studies where they are used. It means that, after the privatization process, it is expected that the data is still useful (Abadi et al., 2016). Fortunately, Differential Privacy already takes that in count. Moreover, DP provides a way of measuring privacy (Dwork and Roth, 2014).

Hasuda, D. and Bezerra, J.
Exploring Differential Privacy in Practice.
DOI: 10.5220/0010440408770884
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1*, pages 877-884
ISBN: 978-989-758-509-8

Most of the papers regarding DP focus on theory, considering definition, foundations and algorithms related to DP (Dwork, 2006; Dwork and Roth, 2014; Dwork et al., 2006; Dwork and Rothblum, 2016). (Jain and Thakurta, 2014) propose a privacy preserving algorithm for risk minimization. (McSherry and Talwar, 2007) indicate that participants have limited effect on the outcome of the DP mechanism. (Minami et al., 2016) focus on the privacy of Gibbs posteriors with convex and Lipschitz loss functions. (Mironov, 2017) discuss a new definition for differential privacy. (Foulds et al., 2016) try to bring a practical perspective of DP, however it focuses on the Variational Bayes family of methods. (Apple, 2017a) present how they determined the most used emoji while preserving users privacy. We then observed that it is missing more pragmatic approaches about how to implement and use DP algorithms.

In this paper, we apply Differential Privacy in practice. There are two main types of privatization: Online (or adaptative or interactive) and Offline (or batch or non-interactive) (Dwork and Roth, 2014). The online type depends on the queries made and the number of them (which can be limited). The offline type of privatization does not make assumptions about the number or type of queries made to the dataset, so all the data can be stored already privatized. We focus on offline methods, specifically on the Laplace mechanism (Dwork and Roth, 2014). We study the impact of this DP mechanism in data analysis. Four classification algorithms were considered, including Decision Tree, Naïve Bayes, Multi-Layer Perceptron Classifier (MLP) and Support Vector Machines (SVM). We are then able to compare the accuracy of each algorithm when using not privatized data or data with different degrees of privatization.

This paper is organized as follows. Section 2 briefly presents DP and related methods. Section 3 presents our programming support, methodology, results and discussions. Section 4 summarizes contributions and outlines future work.

2 BACKGROUND

In this section, we describe the coin method, which is a simple example of DP. Later the definition of DP is presented. We also shows an important DP mechanism called Laplace mechanism, which in turn is a particular case of Exponential mechanism (Dwork, 2006)(Dwork and Roth, 2014).

2.1 Coin Method

(Warner, 1965) describes of a simple DP method. In this experiment, the goal was to collect data that may be sensitive to people and, because of that, they might be willing to give a false answer, in order to preserve their privacy. Let's suppose we want to make a survey to know how many people make use of illegal drugs. It is expected that many people that do use illegal drugs might lie in their answer. But in order to get a clear look at the percentage of people that use illegal drugs, we can use the coin mechanism in order to preserve people's privacy.

It goes according to Figure 1: when registering someone's answer, first a coin is tossed. If the result of the first toss is *Heads*, we register the answer the person gave us (represented by A). On the other hand, if the result of the first toss is *Tails*, we toss the coin again. Being the second result *Heads*, we register *Yes* (the person does use illegal drugs); being *Tails*, we register *No* (the person does not use illegal drugs).

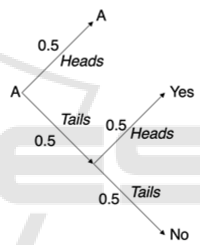

Figure 1: Coin mechanism diagram.

By the end of the experiment, there will be a database with answers from all the subjects, but it is expected that 50% (assuming that the coin has a 50% chance of getting each result) were artificially generated. So, if we look at the answer of a single person, there will be no certainty if that was the true answer.

At the same time, if we subtract 25% of the total answers with the answer *Yes* and 25% of the total answers with the answer *No*, we can have a clear view of the percentage of the population that make use of illegal drugs. It was possible, concomitantly, to have a statistically accurate result (assuming that there were enough people involved in the study) and preserve everyone's privacy.

2.2 Differential Privacy

The basic structure of a DP method consists of a mechanism that has the not privatized data as input, and outputs the privatized data. DP establishes constraints that this mechanism must conform to, in order to limit the privacy impact of individuals whose data is in the dataset.

A mechanism M with domain $\mathbb{N}^{|X|}$ is ε-differentially private if for all $S \subseteq Range(M)$ and for all $x, y \in \mathbb{N}^{|X|}$ such that $\|x-y\| \leq 1$:

$$\frac{P[M(x) \in S]}{P[M(y) \in S]} \leq \exp(\varepsilon)$$

In this definition, we have $P[E]$ as the probability of a certain event E happening and $\|x\| = \sum_{i=1}^{|X|} |x_i|$.

What this definition is making is comparing two datasets (x, y) that are neighbors ($\|x-y\| \leq 1$) and seeing the probability of the resulted dataset after privatization being alike. The mechanism is simply adding noise to the data.

2.3 Exponential Mechanism

One of the most common used DP Mechanism is called the Exponential Mechanism. Let's consider the formal definitions below.

D: domain of input dataset

R: range of 'noisy' outputs

$\mathbb{R}$: real numbers

Let's define a scoring function $f : D \times R \rightarrow \mathbb{R}^k$ where it returns a real-valued score for each dimension it wants to evaluate, given an input dataset $x \in D$ and a output $r \in \mathbb{R}$. In simple terms, such score tells us how 'good' the output r is for this x input. Given all of that, the Exponential Mechanism is:

$M(x, f, \varepsilon)$ = output r with probability proportional to $exp(\frac{\varepsilon}{2\Delta f(x,r)})$ or simply:

$$P[M(x, f, \varepsilon) = r] \propto exp(\frac{\varepsilon}{2\Delta f(x, r)})$$

The Δ is the sensitivity of a scoring function. Formally, we can define the sensitivity of a function as being: For every $x, y \in D$ such that $\|x-y\| = 1$, Δ is the maximum possible value for $\|f(x) - f(y)\|$.

Sensitivity value helps us understanding our data and balances the scale of the noise that must be added, so it makes sense to the data we are analyzing. Imagine a case where the data we want to add noise is a colored image, with RGB values for each pixel ranging from 0 to 255 for each of the tree colors. We need to add a noise to each subpixel that can (with no difficulty) reach values from -255 to 255, so when we look the value of a single subpixel, we don't know what value it was initially. But if we apply this exact same noise to a black and white image, where each pixel can be whether 0 (black) or 1 (white), the noise will be much bigger than the data, and almost all the utility will be lost. For this case, the noise must have a smaller scale. The sensitivity balances the scale of the noise with the possible values the data can reach. We are not going to demonstrate that the Exponential Mechanism is ε- differentially private it here, since it can be found in literature and would deviate from the purpose of this paper.

2.4 Laplace Mechanism

The equation that describes the Laplace Distribution is:

$$f(x \mid \mu, b) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b})$$

If the value of b is increased, for example, the curve becomes less concentrated and more spread. The μ value is the mean of the distribution. This distribution can be useful in DP for adding noise to the original dataset.

The Laplace Mechanism is simply a type of Exponential Mechanism, which makes it easier to be understood. To get to the Laplace Mechanism, we first use the Exponential Mechanism, but with a defined scoring function. Let's consider the following scoring function:

$$f(x, r) = -2\,|x-r|$$

In this scoring function, we are saying that the output $r = x$ is the best for the output. This implies that the format (structure) of the output and the input are the same. If the input is an array of ten zeros, for instance, the best output is the same array of ten zeros.

With that, we can define the mechanism as respecting the equation:

$$P[M(x, r, \varepsilon) = r] \propto exp(-\frac{\varepsilon|x-r|}{\Delta})$$

Using Math manipulations, it's possible to get other form to define the Laplace Mechanism, as follows:

$$M(x, f, \varepsilon) = x + (Y_1, ..., Y_k)$$

where Y_i are independent and identically distributed random variables drawn from $Lap(\frac{\Delta}{\varepsilon})$.

As this mechanism is simple a kind of Exponential Mechanism, we can say by extent that the Laplace Mechanism is ε-differentially private.

3 APPLYING DIFFERENTIAL PRIVACY

In this section, we present the environment to support programming with DP. We discuss the developed experiments, by present and analyzing the results.

3.1 Differential Privacy Lab

As we want to get a better sense on how to use and apply DP mechanisms, the first step was to build a lab in which we could run all of our experiments. For that, we developed an open source code that implements the Laplace Mechanism and is structured to easily integrate different tests, datasets and data analysis techniques. The chosen programming language was Python 3, which is very common in data analysis. Furthermore, we implemented some of it inside Jupyter Notebooks, which is an interactive environment suitable for running the experiments we want to build. The source code is available at https://github.com/dhasuda/Differential-Privacy-Lab.

The code was built to be easily extended. You can plug in a new dataset or your own DP Mechanism. In Figure 2, there is an UML diagram with the representation of the architecture. Each Jupyter Notebook has basically 3 dependencies: *DataProvider*, *Adapter* and *Privatizer*. The *DataProvider* is the dataset, with all the information needed and available to the analysis. *Adapter* is simply a class that adapts the format of the data from *DataProvider* to the format of the *Privatizer*.

Privatizer is the class responsible for implementing the DP Mechanism. All privatizers must inherit from the *AbstractPrivatizer* abstract class. The abstract class implements the privatize method, that estimates the sensitivity of the data (when it is not provided) and, for each value, adds the noise. The shape of the noise is implemented inside *privatizeSingleAnswer*. We then implemented the Laplace Mechanism, which is available in *privatizers/laplacePrivatizer.py*.

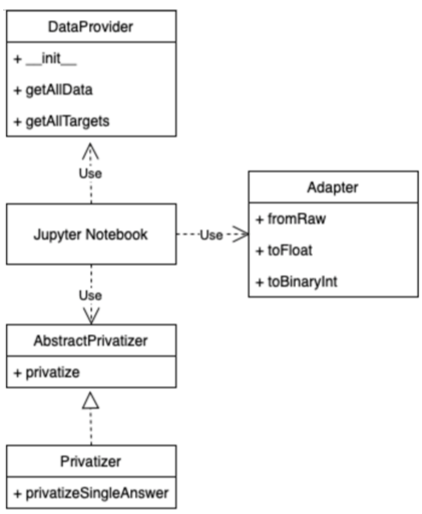

Figure 2: DP lab architecture.

Tests are really important in any code development, that's why there are tests for most of the *.py* files. The script that runs all of the tests is the *runAllTests.sh*. Every time you change something from the existing code, make sure all the tests pass.

3.2 Methodology and Results

The dataset for the experiments was named *fetch_covtype*. It is a tree-cover type dataset (the predominant type of tree cover) available inside *scikit-learn*, an open source, simple and efficient set of tools for data analysis. This dataset contains 581,012 samples, with a dimensionality of 54 and 7 possible classes. For the initial investigation, we add noise to attributes and target (classes).

We implement the analysis of the data for the following classification technique: Decision Tree, Naïve Bayes, MLP and SVM. We use the libraries available in *scikit-learn*. Inside the notebooks (one for each technique), there is a section where we adjust the size of the training data. We can define an array with all training data sizes to test, and for each size of choice, there is a random selection of samples from the database. All the samples that are not used in the training are then considered in the testing, in order to measure the accuracy of the algorithm.

Firstly there is the model training using the raw data. After that, there are multiple data trainings with different values for ε in the Laplace Mechanism that is ε-differentially private. All the values of ε are defined in an array as well. After training the model with the noisy data for different values of ε, all the results of accuracy are printed in graphs. Here we show the results for a fixed size of training data. We use 1,000 samples for Decision Tree, SVM and MLP techniques. We use 100 samples for training in Naïve Bayes algorithm, due to limited processing time. The varying value for this experiment is then the ε value.

For Decision Tree algorithm, results are in Figure 3. We can observe that the bigger the ε value (which leans less privacy), the closer the accuracy of the model is compared to the model trained without privatized data. It is possible to see the convergence of values. Besides, for very small values of ε there is a less consistent accuracy. For Naïve Bayes algorithm presented in Figure 4, the behavior is very similar to the Decision Tree experiment, with the difference of a faster convergence of values when decreasing the privacy level (i.e. increasing of ε value).

According to Figure 5, the SVM experiment shows the same convergence pattern, but with a slower convergence compared to the previous experiments and also a more stable response for very small values of ε. The MLP experiment (presented in Figure

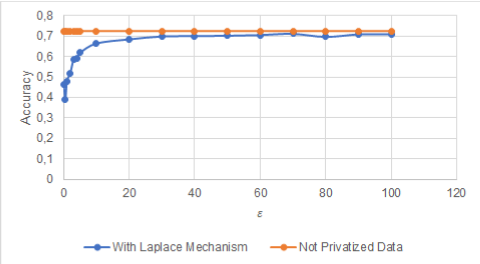

Figure 3: Accuracy of Decision Tree algorithm.

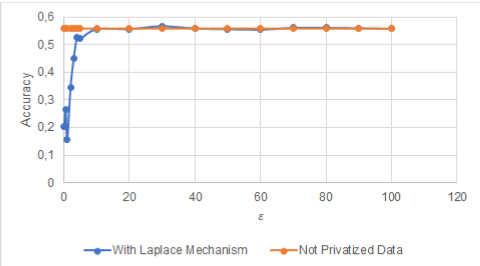

Figure 4: Accuracy of Naïve Bayes algorithm.

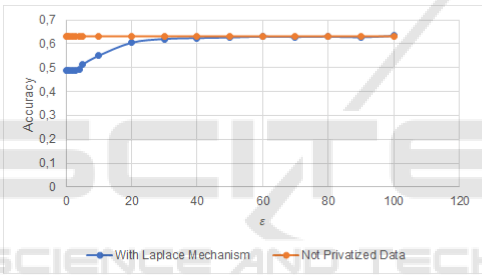

Figure 5: Accuracy of SVM algorithm.

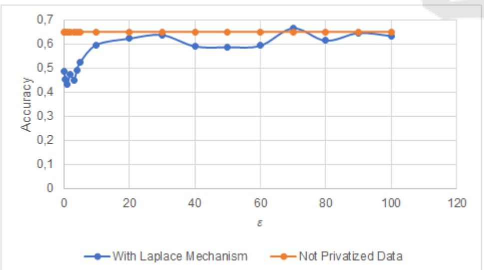

Figure 6: Accuracy of MLP algorithm.

6) is the one with less stable results, and also the one with the biggest difference in accuracy between the model with privatized and not privatized input data. But it is still possible to recognize a convergence pattern, even though it is not as uniform as the previous experiments. In all experiments there was a common pattern, as expected: less privacy means closer results between using privatized and non-privatized data.

3.3 Analysis

We already know that privacy in differentially private algorithms can be measured by the ε value. But what values of ε in a ε-differentially private mechanism are good and really preserve the privacy? How to understand the impact of the ε value? Thinking about this question, we propose a more intuitive way of understanding such value, and we called it *D* Coefficient. *D* Coefficient is based on the Coin Mechanism (Warner, 1965), described in Section 2.1.

The Coin Mechanism considers that there are two possible responses (for instance, *Yes* or *No*) of a person for a question. When registering someone's answer, a first coin is tossed. If the result is *Heads*, it is registered the answer the person gave. On the other hand, if the result is *Tails*, the coin is tossed again. Being the second result *Heads*, it is registered *Yes*; being *Tails*, the response is then considered as *No*.

For defining *D* Coefficient, the modification we made was in the first coin toss of the Coin Mechanism. Instead of getting a 50% chance of getting heads, we decided we would get a *D* chance of getting heads. In other words, the probability of saving the true answer 'A' (and not generating it artificially) will be D, as illustrated in Figure 7. *D* probability is itself the *D* Coefficient that can be calculated, based on the ε value we want to achieve, as:

$$D(\varepsilon) = \frac{exp(\varepsilon) - 1}{exp(\varepsilon) + 1}$$

D Coefficient represents the chance of saving the original answer, if a Coin Mechanism with the same privacy level was used. The entire demonstration of this formula is out of the scope of this paper. Considering the data from the four classification algorithms previously presented, we calculated the value of ε*. We define ε* as the minimal ε that gives us less than 10% difference between accuracy without privacy and accuracy with privacy. We chose 10% in order to have low interference of DP in the analysis, which means that it would be possible to achieve similar findings using privatized data.

We then calculate $D(\varepsilon^*)$. For Näive Bayes, we found $\varepsilon^* = 4$ and $D(\varepsilon^*) = 0.964028$. For the other classification algorithms, we found $\varepsilon^* = 10$ and $D(\varepsilon^*) = 0.999909$. We observed that D value was very close to 1, which is not a good finding. It means that if we use ε* as the privacy level in the Coin Mechanism, the chances of the first coin toss outputting *Heads* would be incredibly high (over 95% for all the experiments). So, the data would be not privatized as expected, since the majority of records would keep the original data.

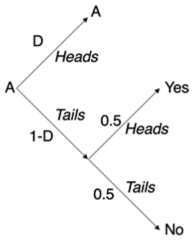

Figure 7: Coin mechanism diagram with variable coin probability D in the first toss.

3.4 More Investigations

One of the parameters that was kept constant in the last experiment was the training data size, i.e. how many samples we used to train each model. For each classification, we did not vary the amount of data used in the training. Now we investigate if more samples can give as better accuracy when dealing with DP. Given the results from *D* coefficient, here we use values of ε between 1 and 3 (which are values of privacy of our interest).

We calculate the relative accuracy for different training data sizes and different ε values. The relative accuracy is defined by dividing the accuracy of the algorithm that used privatized data by the accuracy of the algorithm trained with the raw data. This measurement (relative accuracy) was chosen because we expect the accuracy of the model with no privatized data to increase as the training data increases in size. We ran each experiment five times, and the output is the average of found values.

Figure 8 shows the relative accuracy for the Decision Tree algorithm, where N represents the number of samples in the training dataset. We found no improvement in the relative accuracy when increasing the size of the input data. Actually, for the biggest input size we got the worst results. The Naïve Bayes algorithm, presented in Figure 9, does not follow the behavior of the Decision Tree when comparing the relative accuracy. Here we have a more optimistic result: the best results are the ones with the biggest input sizes. For the MLP algorithm, shown in Figure 10, we observed an unexpected behavior, where it is possible to highlight that the worst result came from the biggest training dataset. Results of SVM algorithm, presented in Figure 11, also indicate that the bigger the input size, the worse is the relative accuracy.

In the chosen database, there are seven possible classification of the predominant type of tree cover (integer value from 1 to 7). The noise, on the other hand, is a real number drawn from a random variable (driven from the Laplace mechanism). In the first experiment, we added noise (related to ε value) also to the classification number, and then rounded the resulting number to match one of the possible classifications. Here we investigate the application of noise only to the attributes and not to the target (classes).

We then compare three values here: accuracy with no privatization, accuracy with full privatization and accuracy with 'semi- privatization' (no noise in the target). We keep the training dataset size fixed. We made five rounds to get each value, and the results present the average value of all these rounds. For the Decision Tree algorithm presented in Figure 12 (for N = 10,000), both lines where data is somehow privatized are close to each other, suggesting that keeping the original values for the target data does not add utility to the privatized data. A similar, but even more optimistic patter, emerges in the Naïve Bayes experiment, as shown in Figure 13 shows (for N = 100). The fully privatized data gets better results compared to the semi-privatized in most of points in the chart.

The MLP algorithm, shown in Figure 14 (for N = 10,000) has points with better results with the fully privatized data compared to the semi-privatized. For the SVM algorithm presented in Figure 15 (for N = 1,000), the results are very close to the Decision Trees, with not much gain in utility by removing the noise from the target. Similar results were found for the analyzed classification algorithms, which means that removing the noise from the target data does not translate to a gain in overall data utility. It is important to point out that not privatizing the target data does decrease the level of privacy of the data. It makes easier, for instance, to implement a linkage attack.

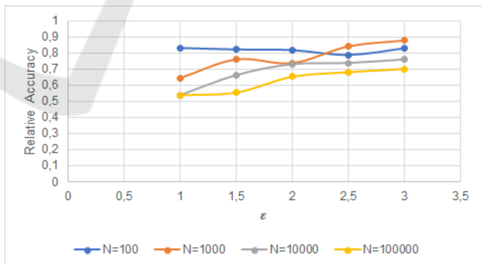

Figure 8: Relative accuracy for Decision Tree algorithm by varying dataset size.

4 CONCLUSIONS

We studied how the theory of Differential Privacy relates to its application, by analysing the impact of DP mechanisms in the utility of the data for different data analysis algorithms. In fact we used four classification techniques: Decision Tree, Näive Bayes, SVM and MLP. By utility we mean the possibility of ex-

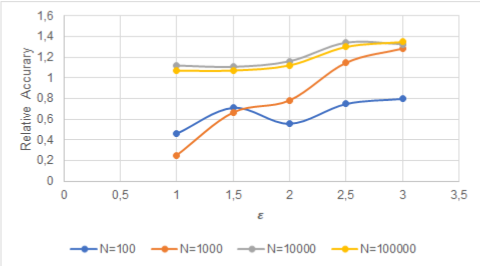

Figure 9: Relative accuracy for Näive Bayes algorithm by varying dataset size.

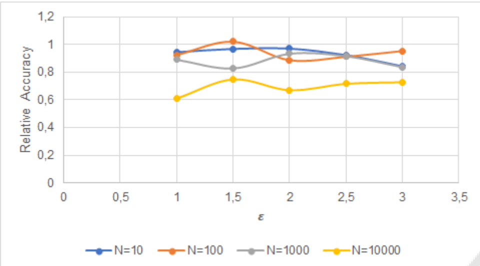

Figure 10: Relative accuracy for MLP algorithm by varying dataset size.

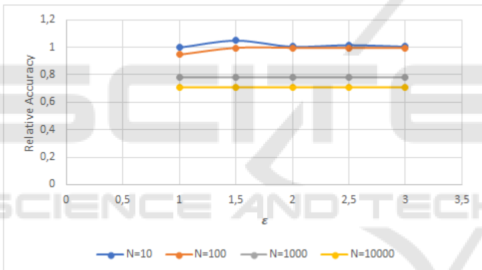

Figure 11: Relative accuracy for SVM algorithm by varying dataset size.

tracting good results when using privatized data instead of non-privatized data. In other words, we were interest to know if the classification techniques could keep their accuracy in the presence of data privatized by a DP mechanism.

The path to understand all of these impacts included the development of a Differential Privacy Lab. We projected it with extensibility in mind. Every part is modular and can be easily replaced. With that decision, we aimed to make a product that was elastic enough to fit into the workflow of anyone starting to develop a Differential Privacy solution. We proved the capabilities of this lab with the experiments we ran on it, and with all the conclusions we got using this tool.

During the analysis of the experiment results, while thinking about the level of privacy, the D Coefficient emerged as a more intuitive way of understanding the level of privacy of a DP mechanism. We

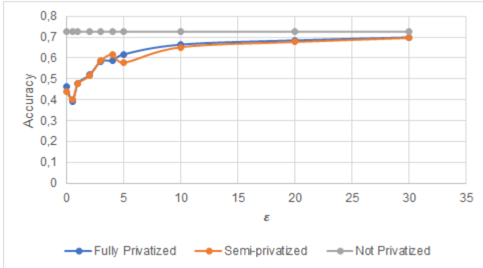

Figure 12: Accuracy of Decision Tree algorithm considering distinct application of noise.

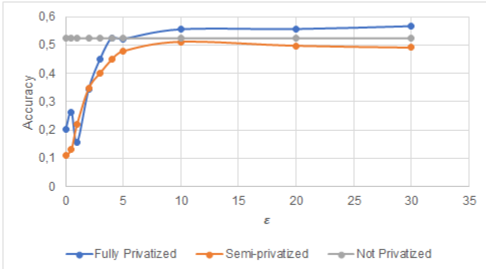

Figure 13: Accuracy of Näive Bayes algorithm considering distinct application of noise.

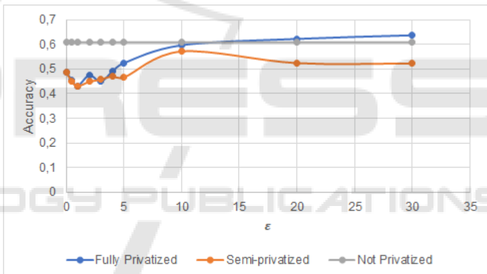

Figure 14: Accuracy of MLP algorithm considering distinct application of noise.

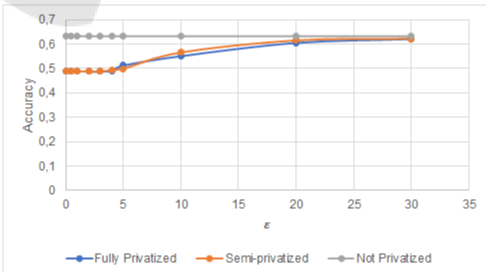

Figure 15: Accuracy of SVM algorithm considering distinct application of noise.

found that small amounts of noise can lead to a big drop in utility. It leads to a concern regarding data privatization, which is that, even with the application of DP Mechanisms, we cannot say the data is being properly privatized and maintaining its utility.

Besides, we found that increasing the number of samples in the training dataset does not always improve accuracy. While removing the noise from the target data, we observed that there were no significant gain in utility. In fact, privacy is a little damaged when data is not fully privatized. Therefore, we encourage adding noise even to the target data.Of course, more experiments need to be conducted to confirm our initial findings.

We believe that the presented experiments as well the developed lab is a sound basis for understanding DP and applying it in practice. As future work, we intend to design new experiments considering different datasets. There can be studied the impact of the privatization in techniques other than classification ones. It is also possible to make other types of exploration, such as combining different DP mechanisms for distinct parts of the data.

REFERENCES

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*.

Apple (2017a). Learning with privacy at scale. [Online; accessed 05-Oct-2019].

Apple (2017b). An on-device deep neural network for face detection. [Online; accessed 11-June-2019].

Buffered.com (2017). The biggest privacy & security scandals of 2017. [Online; accessed 11-June-2019].

Dwork, C. (2006). Differential privacy. In *International Colloquium on Automata, Languages, and Programming (ICALP)*.

Dwork, C., McSherryKobbi, F., and Smith, N. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Dwork, C. and Rothblum, G. N. (2016). Concentrated differential privacy.

Foulds, J., Geumlek, J., Welling, M., and Chaudhuri, K. (2016). On the theory and practice of privacy-preserving bayesian data analysis. In *Thirty-Second Conference on Uncertainty in Artificial Intelligence (UAI)*.

GDPR (2018). General data protection regulation (gdpr). Technical report, Official Journal of the European Union. [Online; accessed 11-June-2019].

Granville, K. (2018). Facebook and cambridge analytica: What you need to know as fallout widens. [Online; accessed 11-June-2019].

Havir, D. (2017). A comparison of the approaches to customer experience analysis. *Economics and Business Journal*, 31(1):82–93.

Jain, P. and Thakurta, A. (2014). (near) dimension independent risk bounds for differentially private learning. In *31st International Conference on International Conference on Machine Learning (ICML)*.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*.

Minami, K., Arai, H., Sato, I., and Nakagawa, H. (2016). Differential privacy without sensitivity. In *30th International Conference on Neural Information Processing Systems (NIPS)*.

Mironov, I. (2017). Renyi differential privacy. In *IEEE 30th Computer Security Foundations Symposium (CSF)*.

Smith, B. and Linden, G. (2017). Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21(3):12–18.

Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–39.

Winkler, R. (2017). Cybersecurity startup tanium exposed california hospital's network in demos without permission. [Online; accessed 11-June-2019].

World, D. I. (2018). How much data is generated per minute? the answer will blow your mind away. [Online; accessed 11-June-2019].