K-Anonymity: The devil's advocate

Lilit Yenokyan
LMU
Munich, Germany
lilit.yenokyan@campus.lmu.de

Patricio Massaro
LMU
Munich, Germany
p.massaro@campus.lmu.de

***Abstract*—As personalized and information-rich environments become more prevalent, the risk of private data becoming public increases. With the ubiquitous amount of available online data, the need for privacy protection became a critical issue. One of the main methods to address this concern is anonymization. However, from a data scientist's perspective, this privacy-preserving method can amount to vast data loss and a decrease in the performance quality of the machine learning models trained on these data. The purpose of this work is to perform an analysis of different $k-anonymity$ methods and check their influence on different regression and classification tasks. Varying values of k with datasets of various sizes and objectives are examined. We compare the performance of different models using task-specific metrics and visualize them in an intuitive manner. The results indicate that factors such as the anonymization algorithm and the hierarchy structure used in it play a key role in differences in performance. Based on these findings, different strategies to balance privacy requirements with minimal impact on performance are discussed.**

***Index Terms*—Keywords: Anonymization, K-anonymity, Mondrian, Datafly**

I. Introduction

Nowadays, massive amounts of data can be available practically to anyone through a single Google search. These data can be used for many purposes, one of them being maliciously identifying people, their home addresses, and other sensitive information. Therefore, people responsible for sharing these data must be very cautious. As Uncle Ben said, "with great power comes great responsibility." One way to handle this responsibility is to protect data privacy through anonymization. There are many anonymization methods like $K-anonymity$, $L-diversity$, and $T-closeness$. These methods transform the sensitive information in the datasets to make them less vulnerable to de-identification attacks.

Nevertheless, the information contained in these datasets is highly valuable in different research fields, e.g medicine. Regardless of the method of anonymization it always leads to a certain information loss which can be crucial in research. For example, K-anonymity drops or masks some information to make the record "hide" in a group of other k-1 similar records which, consequently, leads to data loss [1] .

In this project, we aim to analyze the effect k-anonymization has on machine learning tasks, particularly on regression and classification. The work is structured as follows: On section II, the general idea of k-anonymity and some of the algorithms used in the field are introduced. In section III three datasets covering different domains that are used for the analysis are introduced. In section IV a complete walk-through of the implementation pipeline is performed, explaining the data preprocessing and the machine learning tasks. In section V the results are visualized and discussed. Finally, the conclusions and findings of the analysis are displayed in section VI.

II. K-Anonymity

A. General idea

K-anonymity allows the transformation of the dataset in a way that for example individuals are not identifiable. The core principle is simply to be lost in the crowd of similar records, in fact, in at least K-1 such records. The attributes in the dataset can be $direct\ identifier$ attributes and $quasi-identifier$ attributes. The direct identifier attributes must be removed as they alone are enough to identify records, examples of such attributes are name, social security number, phone number, etc. The quasi-identifiers are not necessarily sensitive attributes, however, they can be used for re-identification. There is no fixed list of such attributes but usually, they are somehow available through other datasets. Examples of such attributes can be Zip code, age, gender, and profession, the combination of which might locate the individuals.

K-anonymization can be achieved through two methods: $generalization$ and $suppression$. Generalization is the process of transforming (generalizing) records that have identical values to records that have the same values for the given QID attribute. Consequently, it makes the QID values less precise, i.e. numerical values get transformed into a range containing these values, and categorical values may get replaced by more general values with the help of $generalization\ hierarchies$.

Suppression is the process of completely removing the attribute's values from the data table. It is noteworthy that the attributes being suppressed must be irrelevant to the data collection task. For example, if we have a goal to predict (in a supervised setting) the values of the "income" attribute for a dataset, "income" shall not be suppressed. If anonymization of this attribute is absolutely necessary it should be done through generalization.

B. Algorithms

There are several methods to transform a dataset into a k-anonymized one. In this section, we will give an overview of the k-anonymization algorithms that we use in the project.

a) Classic and Basic Mondrian: The Mondrian algorithm was first introduced in [2]. The authors propose to recursively divide the raw dataset with the KD-tree into groups that contain at least K records [3] . Then generalize the QID attributes in each group to have the same intergroup values. The partitions of the tree are done using the attributes that give the widest range of values in each split. The initial or classic Mondrian is intended for numerical values and has to convert categorical attributes to numerical ones and use some intuitive ordering. To handle categorical values [4] proposes generalization hierarchies. This method is referred to as Basic Mondrian. We use both methods in our experiment as they have core differences when it comes to numerical values.

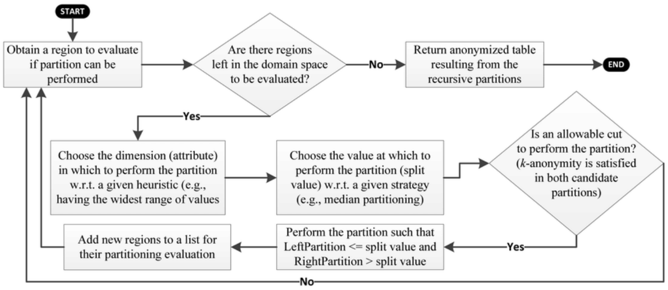
START
Obtain a region to evaluate if partition can be performed
Are there regions left in the domain space to be evaluated?
No
Return anonymized table resulting from the recursive partitions
END
Yes
Choose the dimension (attribute) in which to perform the partition w.r.t. a given heuristic (e.g., having the widest range of values
Choose the value at which to perform the partition (split value) w.r.t. a given strategy (e.g., median partitioning)
Is an allowable cut to perform the partition? (k-anonymity is satisfied in both candidate partitions)
Yes
Perform the partition such that LeftPartition <= split value and RightPartition > split value
Add new regions to a list for their partitioning evaluation
No

Fig. 1: Classic Mondrian algorithm [5]

b) Datafly: The datafly is a greedy single-dimensional k-anonymity algorithm. It works by generalization, substitution, and suppression. Datafly recursively divides the dataset into groups by the QID that has the highest frequency. If after the splits the k-anonymity is satisfied in the groups, suppression is applied, otherwise, the next frequent QID is used to perform a split. As a result, the performed generalization is not the most optimal and leads to over-generalization.

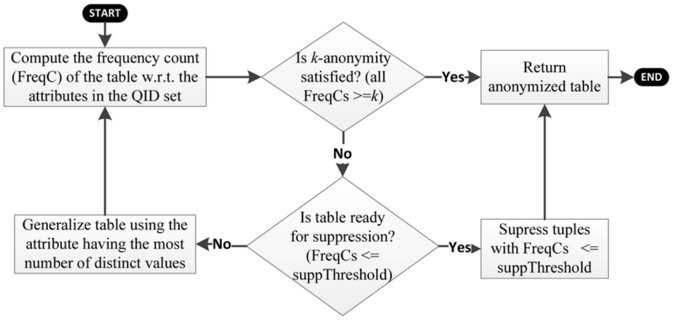
START
Compute the frequency count (FreqC) of the table w.r.t. the attributes in the QID set
Is k-anonymity satisfied? (all FreqCs >=k)
Yes
Return anonymized table
END
No
Is table ready for suppression? (FreqCs <= suppThreshold)
Yes
Supress tuples with FreqCs <= suppThreshold
No
Generalize table using the attribute having the most number of distinct values

Fig. 2: Datafly algorithm [5]

III. Datasets

As discussed earlier the goal of this project is to compare the performance of different machine learning tasks on the original and k-anonymized datasets. To make the final discussion more inclusive we work with datasets from heterogeneous domains. In this section, we will discuss ADULT, CA-Housing, and CMC datasets. Note that all three datasets are publicly available.

A. ADULT dataset

The $ADULT$ dataset is census data collected by the US Census Bureau in 1996. It contains 16 columns describing different demographics of people such as age, occupation, and marital status. The target column is the income column which contains two values $<= 50K$ and $> 50K$. Therefore, it is commonly used for binary classification tasks. We as well use this dataset to predict the income of people. As mentioned earlier, attributes such as age, education, and occupation alone do not contain sensitive information, but taking them as a group and having some external information would make this dataset open to re-identification attacks. Therefore, we take attributes "age", "education", "marital_status", "native_country", "occupation", "race", "sex", and "work class" as quasi-identifiers for the anonymization algorithm. The dataset has around 45K rows [6].

B. California Housing dataset

The next well-known dataset we use is the $California\ Housing\ Prices$ dataset. It contains some information about one of California's district's houses from 1990 census data. The attributes include house coordinates (longitude and latitude), the house's median age, the number of rooms, etc. We use the dataset for a regression task where the target variable is the median house value. Obviously, the locations of the houses contain sensitive information and must be anonymized. However, we cannot suppress or remove these two attributes altogether as the location of the house greatly impacts the house value, the variable of interest. Therefore, we use generalization hierarchies to generalize the longitude and the latitude. In a similar fashion, a hierarchy is applied to the "median_income" variable. The dataset has around 20K samples [7].

C. Contraceptive Method Choice dataset

K-anonymization is largely used in the medical field to protect the information of patients. Therefore, as part of our discussion, we had to include medical data. We use the $Contraceptive\ Method\ Choice$ dataset from the 1987 National Indonesia Contraceptive Prevalence Survey. The dataset contains demographic and socio-economic information about married women and their choice of contraceptive method. The latter has three distinct values: not using, using long-term, and short-term. The prediction goal is the choice of the contraceptive method, hence we have a multiclass classification problem. We use generalization hierarchies to anonymize "education", "age" and the "number of children ever born" attributes. The dataset consists of 1473 instances [6].

IV. Implementation

To conduct performance analysis, training of multiple machine learning models is needed. The proposed workflow, depicted in Fig. 3, is designed with scalability in consideration. It involves: (1) Anonymizing the dataset for various values of k, (2) Preprocessing each anonymized dataset, and (3) Training a baseline model without anonymization and comparing its performance to that of the k-anonymized datasets.

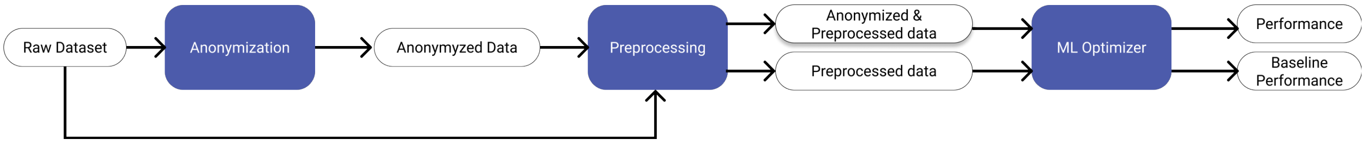

Fig. 3: Implementation pipeline: anonymization for a set of k, preprocessing and model training

A. Anonymization

The anonymization of the datasets is the first and foremost step in the pipeline. *k-anonymity* implementation available on GitHub is used. The authors provide five k-anonymization methods including the $datafly$, $classic\ Mondrian$, and $basic\ Mondrian$ methods. To anonymize any dataset using this code, one must provide the dataset along with its generalization hierarchies for all quasi-identifier attributes. One of the reasons for choosing this specific code is that it allows specifying the attributes for the anonymization. As a result of which one can easily change task targets and, therefore, exclude these targets from the list of QIDs. For example, for the $CAHousing$ dataset, they use "ocean proximity" as the prediction target and generalize the "median house value" which we easily reversed to fit our project.

The main source code for the methods have been used from this implementaton, and some changes regarding the function calls was made to better fit this project's needs.

B. Preprocessing

The preprocessing step is crucial in any machine learning pipeline. Common tasks include removing irrelevant columns and encoding categorical variables using one-hot encoding. Additionally, the datasets used are usually well-prepared and do not require extensive handling of missing values or outliers.

However, as the anonymization prioritizes security and does not regard ML performance, it introduces several challenges in the preprocessing step:

- An attribute may be completely suppressed to ensure k-anonymity. In that case, the feature should be removed.
- For some records, a numerical attribute is displayed as an interval but for others is displayed as the number itself. The proposed solution consists of imputing the mean of the intervals, as depicted in table II

ID	Age	Imputed
1	20∼36	28
2	31	31
3	20∼30	25

TABLE I: Imputation procedure for anonymized numerical attributes

- A similar phenomenon happens in categorical attributes, where different hierarchies are applied depending on the record, generating overlapping categories. A possible approach is to apply the most general hierarchy of the feature but doing so would increase the k value of the data, making the results of the analysis invalid. It was decided not to make further transformations and apply one-hot encoding.

ID	Occupation	Imputed
1	Programmer	?
2	Tech-sector	?
3	Private-sector	?

TABLE II: Imputation procedure for anonymized numerical attributes

The output of this step is a set of preprocessed datasets, ready to be used in the training of ML models.

C. Machine Learning

One possible source of error in the analysis is the ML algorithm used for the benchmark. Spurious relationships could modify performance for certain values of k, making the results invalid. To avoid that, Two of the best-performing algorithms for tabular datasets are used, XGBoost [8] and random forest [9]. To validate the obtained results, consistent behavior of the performance metric should be seen between the algorithms. Moreover, in previous research such as [10], one model with default hyperparameters was trained for each k. This could lead to invalid results due to the fact that the default configuration could benefit some values of k while degrading the performance for others. The proposed solution is to use Bayesian optimization (BO) with the *Hyperopt* tool [11] to determine the optimal configuration for each anonymized dataset. As shown in Fig. 4, the BO is conducted on a training dataset using 4-fold cross validation, and the performance is evaluated on a separate test set.

The Optimizer class performs three key tasks: 1) identifies the dataset and problem type, 2) defines the search space and performs BO with 4-fold cross-validation, and 3) makes predictions with the best model and reports relevant metrics, saving them in json format.

D. Metrics

The final step of the Optimizer class is to report the relevant performance metrics for the test set. Three measures are reported, depending on the corresponding task. Each metric is detailed below:

1) ROCAUC: The receiver operating curve is a graph that shows the performance of a binary classification model at all classification thresholds. It compares the true positive and the false positive rates, which are based on the confusion matrix.

$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{FP+TN} \tag{1}$$

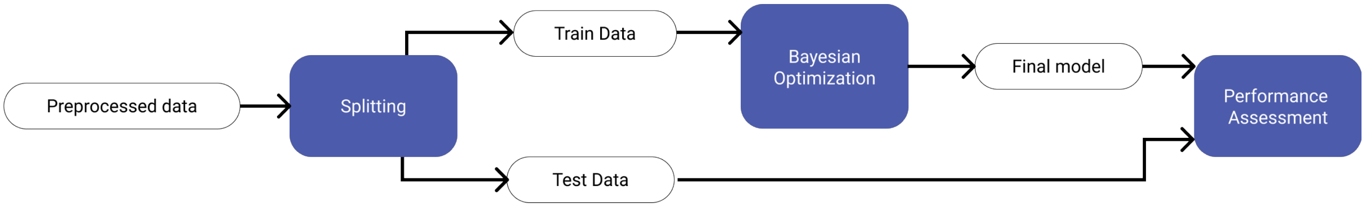

Fig. 4: ML Optimizer diagram

Components	Count
ML Algorithms	2
Datasets	3
Ks	21
Bayesian opt. iterations	20
Cross-validation	4
Pipeline runs	4
Total models	40320

TABLE III

As the area under this curve is an aggregate measure of performance across all possible thresholds, it is a robust metric for balanced and moderately imbalanced datasets. The values of the ROCAUC are between 0 and 1, where a value of 1 represents a perfect classifier.

2) RMSE: The Root mean squared error is one of the most used metrics in regression due to its interpretability and mathematical properties. It is defined as :

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{N}(y-\hat{y})^2} \tag{2}$$

3) Macro F1-Score: The F1-score is a popular metric for evaluating the performance of a classification model, defined as the harmonic mean of the precision and recall, which are based on the confusion matrix.

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{3}$$

$$F1_{macro} = \frac{1}{M}\sum_{i=1}^{M} F1_i \tag{4}$$

As this metric only works for binary classification, it must be adapted for the multi-class case. The F1-score for each class is calculated in a one-vs-rest fashion, and then they are averaged in different ways. The macro F1-score corresponds to the arithmetic mean of the F1-scores for each component. This metric is suitable where there is not a severe imbalance between the samples for each class.

E. Methodology

The pipeline was executed four times for each dataset. Each execution requires 42 bayesian optimizations (21 for each algorithm). One optimization belongs to the non-anonymized datasets, the other twenty are for different values of k between 2 and 100. Each BO requires 20 steps of 4-fold cross-validation to get the best-performing model. A total of 13440 models were trained in order to display the results from sec. V.

V. Discussion

This section examines the outcomes of the pipeline runs. The behavior of the performance metric as the level of anonymity in the datasets increases is depicted in Figure 5. The figure presents each dataset as a row and each machine-learning algorithm as a column. The baseline performance from the non-anonymized dataset is illustrated using a red dashed line. The x-axis represents the value of k, while the y-axis shows the value of the corresponding metric for the machine learning task.

The first row shows the results for the ADULT dataset, where the area under the ROC curve is used as the performance metric. The variation of the metric between the runs is low, indicating consistent results for both algorithms. However, a significant drop can be noticed for the $datafly$ algorithm, starting with a low k. A deeper analysis showed that the data was severely anonymized from the start, with most of its QIDs completely suppressed. This shows one of the most well-known disadvantages of datafly, its tendency to over-generalization. Additionally, the fact that most of the QIDs are categorical explains the difference in the performance of the $classic\ modrian$ with respect to the $basic\ mondrian$. The former uses an ordering-based approach that is not well suited for categorical attributes, while the latter uses a hierarchy-based method to fulfill the k-anonymity condition.

w

In the second row, the results for the California Housing datasets are shown. In this regression task, RMSE is used as the metric. As opposed to the case of the ADULT dataset, the QIDs of this dataset are mostly numeric, which drives different behavior on the performance measure. The results show a large gap for all algorithms, indicating that when the attributes have high cardinality, as in the case of numerical attributes, the information is lost quickly during the anonymization process. Contrary to the ADULT dataset, $classic\ mondrian$ demonstrates better and more robust performance than $basic\ mondrian$. This is due to the data being more favorable for an ordering-based algorithm like $classic\ mondrian$, while $basic\ mondrian$ is highly dependant on the granularity of the hierarchies defined by the user. There is a discrepancy in the results between XGBoost and random forest for $basic\ mondrian$ when k is between 40 and 60. A possible reason for that could be that random forest has a more limited

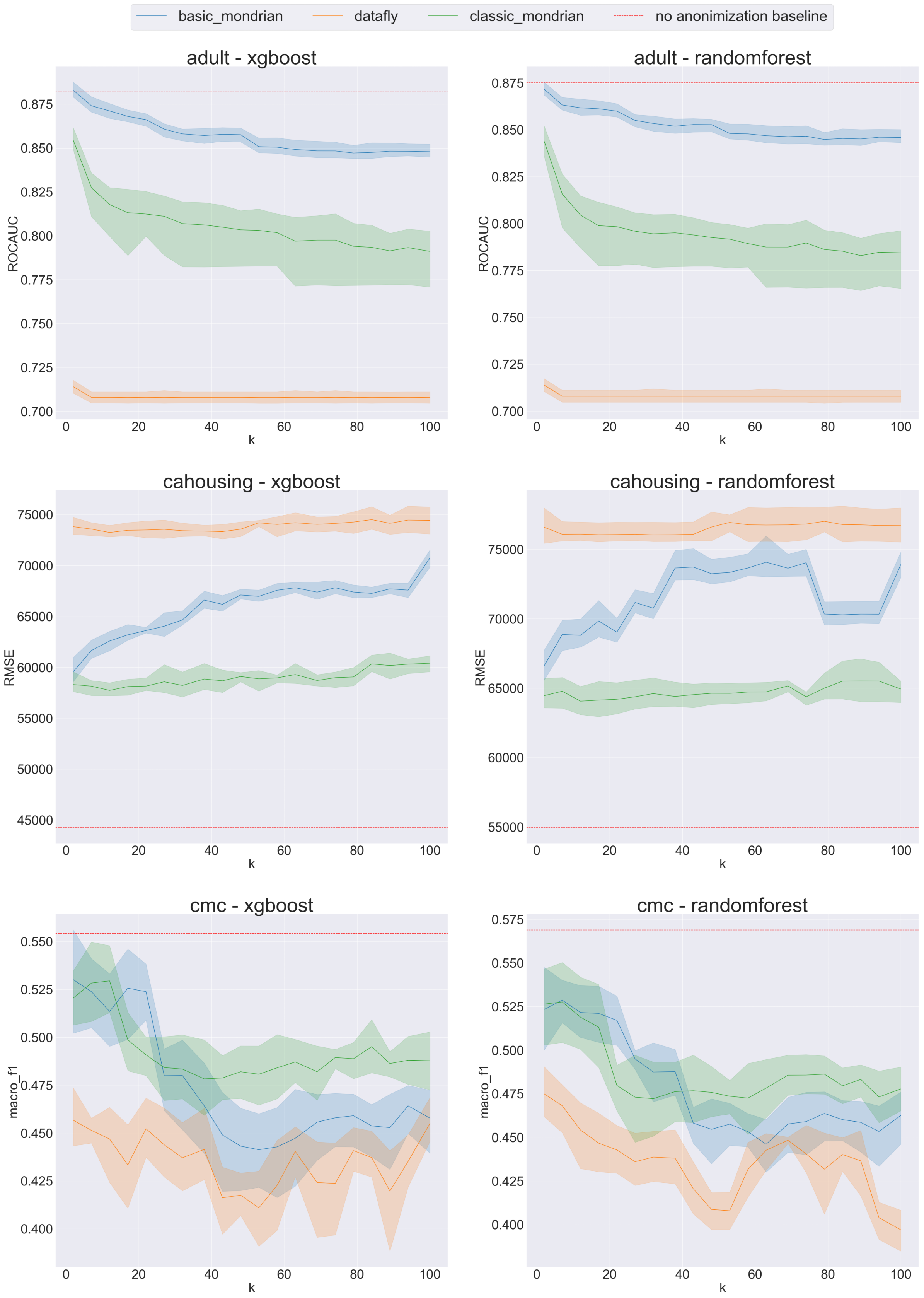

Fig. 5: Performance of ML tasks using two different algorithms as k increases. The first row corresponds to the ADULT dataset. The reported metric is the area under the ROC curve. The second row displays the California Housing dataset, using RMSE as performance measure. The third rows shows the impact of anonymization in the birth control dataset, a multi-class classification where the macro F1-Score is used.

hyperparameter search space, where the optimal configuration is more difficult to find for the BO in only 20 iterations.

The third row displays the results of the multi-class classification task using the birth control dataset (CMC), where the Macro F1-Score is used as the performance metric. One notable difference with the ADULT and California Housing is the size of the CMC dataset, which has orders of magnitude fewer records. This smaller dataset size may not provide enough data for the proposed models to converge correctly, resulting in a mediocre performance of the machine learning algorithms for the non-anonymized datasets and high variability in the results of each pipeline run. Additionally, in such a small dataset the impact of k is higher, as can be seen in the table IV. As in California Housing, the QIDs of the CMC dataset are numerical, but the attributes have less cardinality and the generalization hierarchy proposed is more granular. This explains the smaller gap in performance for both $mondrian$ algortihms comparing to the RMSE, and a smaller difference between the ordering-based ($basic\ mondrian$) and hierarchy-based ($classic\ mondrian$) algorithms Moreover, the $datafly$ shows again the worst performance because of it's over-generalization,

Dataset	Maximum K	% of the dataset
ADULT	100	0.307
CAH	100	0.484
CMC	100	6.789

TABLE IV

VI. Conclusion

The objective of this work is to analyze the effect of k-anonymization on the performance of the machine learning tasks for different datasets. To achieve this goal, a resilient and scalable pipeline is a must. The proposed solution was able to train a vast amount of models with as fair as possible conditions for the analysis, without requiring human intervention, while also displaying them in an intuitive manner.

As expected, anonymization, particularly k-anonymity, leads to information loss which in its turn leads to a decrease in the model performance. Additionally, as k increases the drop in performance also often increases.

Another significant point to note is the importance of the role of the quasi-identifiers. In the anonymization phase, the types of values of the QIDs (numerical vs. categorical) must be taken into account, and the used method should be based on that. In the case of numerical attributes with high cardinality, the more flexible ordering-based approaches show more robust performance because they do not rely on user-defined hierarchies. Methods like classic Mondrian work with numerical ordering and hence, lead to generalization while minimizing information loss. However, when working with categorical values, methods such as $basic\ mondrian$ are preferable as they are based on generalization hierarchies. Having a good set of generalization hierarchies is crucial to the task, especially if the QIDs are numerical, where applying any not granular enough hierarchy implies significant information loss. Additionally, anonymization methods that over-generalize such as $datafly$ could be beneficial in a privacy perspective, buy may make the ML task unfeasible.

An issue consistently present in all the discussed models is the creation of overlapping categories. This is when the anonymization method, with the use of the generalization hierarchies, creates numerical intervals or categorical values that overlap within the records. While in a privacy perspective, this phenomena is not an issue and it is even a wanted feature, During modeling these anonymized values may present problems when they go through a ML pipeline. In numerical attributes, mean imputation can partially solve the issue. In categorical attributes, the problem gets trickier, as one-hot encoding may introduce distortions. One solution is to apply the higher hierarchy found in the data, but this approach modifies the value of k-anonymity and should be taken into account. Another solution could be to do mode imputation, which might lead to higher information loss and even full-column suppression.

The result of this work suggest that in order to balance privacy concerns and utility in machine learning tasks, the parties involved should work together. Achieving a desirable outcome is not solely dependent on the value of k, but also on selecting a suitable algorithm that takes into account the type of quasi-identifiers (QIDs) and implementing a proper generalization hierarchy.

References

[1] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[2] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*, pages 25–25. IEEE, 2006.

[3] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[4] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Workload-aware anonymization. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 277–286, 2006.

[5] Vanessa Ayala-Rivera, Patrick Mcdonagh, Thomas Cerqueus, and Liam Murphy. A systematic comparison and evaluation of k-anonymization algorithms for practitioners. *Transactions on Data Privacy*, 7:337–370, 12 2014.

[6] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[7] Cam Nugent. California housing prices, Nov 2017.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[9] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[10] Djordje Slijepčević, Maximilian Henzl, Lukas Daniel Klausner, Tobias Dam, Peter Kieseberg, and Matthias Zeppelzauer. k-anonymity in practice: How generalisation and suppression affect machine learning classifiers, 2021.

[11] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science Discovery*, 8(1):014008, 2015.