Anonymization of the COMPAS Dataset

Privacy, Utility and Risk Analysis

João Almeida*, José Donato†
Departamento de Engenharia Informática
Universidade de Coimbra
*jlalmeida@student.dei.uc.pt, †donato@student.dei.uc.pt

I. Introduction

When releasing a dataset containing personal information, care must be taken to ensure the individuals' privacy is not infringed upon.

Even if explicit identifiers are removed, an attacker with background knowledge on their target may use it to identify the seemingly anonymized record (record-linkage threat [5] - cf. Sec.II-G).

Further anonymization operations are required to achieve acceptable privacy levels, inevitably reducing the data's utility.

In this paper we detail the process that went into anonymizing ProPublica's (in)famous COMPAS dataset ([4],[2]), with the aid of the ARX tool [1].

Sec. II focuses on preparing the dataset for the subsequent (Sec. III) application of privacy models. Sec. IV will then explore the results, in the context of the trade-off between privacy and utility.

II. Dataset

A. Characterization

The dataset [2] contains information of criminal offenders, controversially used by the COMPAS risk assessment tool [7] to compute their risk of recidivism and aid decisions of the US's legal system [6]. It contains demographic/criminal history information [6].

This was a straightforward choice for the assignment, given the lack of non-synthetic datasets with this amount of untampered sensitive personal information - those available are pseudoanonymized to respect the individuals' privacy and/or data protection regulations.

B. Sanitization

We perform the filtering described in the original analysis [2], leaving us with 6172 records (originally there were 7214).

```
preprocessed = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    df["is_recid"] != -1) &
    df["c_charge_degree"] != "O") &
    (df["score_text"] != 'N/A')
]
```

While some columns still contain missing values, we will not remove them as is common for applications such as machine learning analysis ([6],[7]), because all ARX's methods correctly handle null values [1]. Those utilizing the released dataset can then choose how to handle this missing data, namely resorting to an imputation method.

Attribute Classification (Sec. II-C)		
PII	QID	SA (sensitive)
id	sex	c_charge_degree
name	dob	c_charge_desc
c_case_number	age	priors_count(*)
	race	decile_score
	c_arrest_date(*)	is_recid
	c_offense_date(*)	

TABLE I

Since the attributes are all well encoded, we proceed to select a subset of what were originally 53 columns. This reduces the models' search space (aggravated by techniques such as local generalization), and mitigates the curse of dimensionality affecting many privacy models (alas, LKC is not supported by ARX).

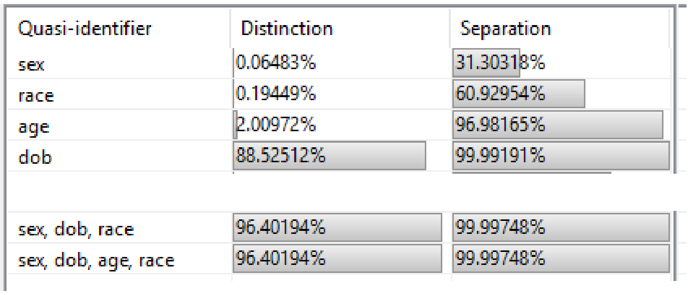
The accompanying `.ipynb` goes over this procedure from a more practical view, including the code to replicate it.

C. Attribute Classification

Table I classifies the attributes we have selected for anonymization.

`sex`, `dob` and `race` are *classic* quasi-identifiers, since they are common background information to have on someone, and together have high values for distinction and separation - the two metrics used as indicators of QIDs.

As can be seen from Fig.1, together these uniquely identify approximately 96.4% of the records in our dataset.



Quasi-identifier	Distinction	Separation
sex	0.06483%	31.30318%
race	0.19449%	60.92954%
age	2.00972%	96.98165%
dob	88.52512%	99.99191%
sex, dob, race	96.40194%	99.99748%
sex, dob, age, race	96.40194%	99.99748%

Fig. 1: Values for distinction and separation (abridged)

The remaining QIDs identified have the caveat that we assume the attacker knows the target is in the dataset (has committed a crime, but not which).

Even so, the identification of QIDs is very tricky procedure, as some attributes can be classified differently depending on how we model the attacker, and also on our own personal interpretation. For this reason, what we what we present in Table I should not be considered definite, and will be subject to slight tweaks during the experimentation phase. Namely, the indicators of a crime having been committed - marked with (*) - but not which, could be interpretted as either QID or SA.

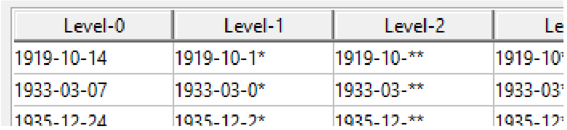
Regarding non-sensitive attributes, we did not identify any. As aforementioned, we have demographic data (typically PII/QID) and criminal history information (typically sensitive since the individual may not wish its disclosure - although in some countries it could be deemed public).

Note that we kept age and dob, even though the former can be directly derived from the latter, as their formats lead to different coding models (Sec. II-D).

D. Attributes' Coding Models

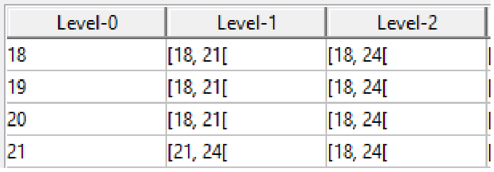
We define hierarchies for the quasi-identifiers which will be manipulated by ARX when applying each privacy model.

For the date attributes, a **masking** hierarchy is used, at a first level generalizing the least significant digit of the day, then the most, and *upwards* to the month and year:



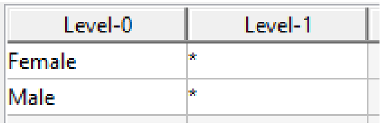
Level-0	Level-1	Level-2	Le
1919-10-14	1919-10-1*	1919-10-**	1919-10
1933-03-07	1933-03-0*	1933-03-**	1933-03
1935-12-24	1935-12-2*	1935-12-**	1935-12

For age we define **intervals** of increasing size (smallest is of 3 years):

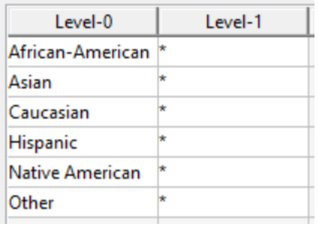


Level-0	Level-1	Level-2
18	[18, 21[	[18, 24[
19	[18, 21[	[18, 24[
20	[18, 21[	[18, 24[
21	[21, 24[	[18, 24[

Finally, for sex and race we use the **ordering**-based hierarchy for a single global generalization - *:



Level-0	Level-1
Female	*
Male	*



Level-0	Level-1
African-American	*
Asian	*
Caucasian	*
Hispanic	*
Native American	*
Other	*

E. Anonymization Goal

Our goal would be to release the dataset to help study these automatic analysis systems, particularly their fairness ([7],[6]), without compromising the identity of the individuals in it - ie. continuing to impact their lives after serving their sentences.

Ideally we wouldn't even be able to tell if someone was in the released dataset (table-linkage[5] / membership disclosure[1]).

F. Privacy Risks

The attributes' distributions lead to distinct QIDs for each record, and so the average re-identification risk is 99.9838% (a single pair of QIDs is identical - lowest prosecutor risk of 50%).

This average risk drops to 92.6766% if we don't assume the attacker has *criminal-related* dates as background knowledge (ie. now considering them sensitive attributes), but it is nevertheless a grim starting point for our process which we'll aim to significantly improve.

Regarding the possible re-identification risks, we can identify some scenarios where each may be a concern:

- prosecutor risk (specific target) is probably the most critical. An example would be in candidate screening, where an attempt is made to identify the candidate in the dataset. Similarly, a nosy acquaintace, or a co-worker wanting to throw another under the bus.
- journalist risk (any one individual as target) will always pose some threat, with researchers attempting to apply their de-anonymization methods. If the results are publicized, they may be replicated by others.
- for marketer risk (as many targets as possible) we can consider an attacker which, while not particularly related with any record, will want to de-identify as many records as possible to extort their owners' over its disclosure.

G. Privacy and Utility Requirements

To support our goal of helping those researching fairness in automatic decision-making, we will minimize the anonymization operations performed on the race and sex attributes, since these are the ones under study in those cases ([6],[7]). In fact, considering the chosen coding models, these should ideally remain unchanged.

Regarding privacy requirements, we will consider increasingly high threat levels [5], choosing appropriate privacy models for each (Sec.III-A) :

- not identifying the target's record (record-linkage)
- additionally, not infering any sensitive value, from the - now several - possible records (attribute-linkage)
- additionally, not concluding the presence of the target in the dataset (membership-linkage)

We thus want to minimize the utility loss of the released dataset compared to the original, among the solutions with acceptable privacy levels. For record-linkage we can consider the equivalence class sizes (namely the minimum and average), but when considering sensitive attributes this task becomes much more challenging - we're not provided any particularly enlightening metric, and will mostly resort to the algorithm's own parameters for the privacy-focused restrictions they impose.

Nevertheless, we should not end up with one single obvious solution given the nature of this problem - multi-objective optimization. While some configurations may provide better solutions according to both objectives (privacy/utility), in the end a decision must be made over which "good" solution to move forward with.

III. Privacy Models

A. Selection

Following [5]'s table, we select a model targeting each privacy threat, among those supported by ARX.

- for record-linkage, k-anonymity
- for attribute-linkage, both l-diversity and t-closeness
- for membership-linkage, δ-presence

As such, we will incrementally explore each concern that should be taken when anonymizing.

1) ***k-anonymity:*** **??**
the classic privacy model.

It creates groups of records of size `k` (equivalence classes) in which all the QIDs have the same values. As such, an attacker in possession of background information (QIDs) will not be able to single out the target's record (record-linkage).

It is nevertheless highly flawed, first and foremost by completely neglecting sensitive attributes.

Another significant flaw of this algorithm (and those based on it), is its deterioration with higher number of QIDs, because creating an equivalence class for records will require more and more generalization. Enabling suppression we can discard the outliers (particularly distinct records), but the curse of dimensionality still looms.

The relaxation introduced by the LKC model, considering only combinations of `L` QIDs for the equivalence classes, would have us certainly choose it, if only it were implemented in ARX.

2) ***l-diversity / t-closeness :*** although it will (almost surely) require more generalization to ensure the required diversity, the removal of the attribute-linkage threat is essential. For example, imagine if even we have a large equivalence class, all records have committed a first degree felony. An attacker, while not being able to identify the specific record, will still learn this fact from the individual's class.

In **l-diversity**, `l` introduce a constraint on the diversity of the sensitive attributes, more specifically on the number of unique values in each equivalence class. For example, in our dataset, if we have four records in the same equivalence class, to respect 4-diversity each record needs to be arrested for a different crime. One major flaw of this algorithm is the utility loss, and studies say that 3-diversity datasets are worse than 100-anonymity in terms of utility [8]. In addition, l-diversity does not require that the distribution of the values (even if diverse) is the same as the rest of the data, and is thus prone to probabilistic information gain attacks.

With this in mind, we also tried **t-closeness**. In this privacy algorithm we also want to make sure that the distribution of sensitive values in each class are close (upper-bounded by a threshold `t`) to the global population's.

3) ***δ-presence :*** δ-presence is concerned about being able to infer the presence of an individual in the dataset (membership-disclosure), controlled through the δ_{max} parameter. Infering their absence is also considered in the algorithm - with δ_{min} - but in our case that is not a concern.

If the attacker has information about the *global* population from which the released dataset was drawn, they can use it to infer with a given probability that the target's in the dataset. This probability is derived from the number of records for the target's equivalence class in the dataset vs. the population, which is exactly what δ-disclosure controls with δ_{max} (upper-bounds it).

For datasets such as this one, just knowing someone's in it is sensitive. Record/attribute-linkage differ in that they assume the victim's record is in the dataset [5].

B. Configuration

In this section we describe the parameter space we'll be exploring in the Analysis step, with some being specific to a privacy model and as such described separately.

The following list details ARX's general parameters, which we'll tune in all configurations:

- **Attribute weights:** in Sec.II-G, we specified that we should minimize the anon. operations on `race` and `sex`. ARX supports this by allowing us to specify higher weights for these QIDs, for which it will attempt to reduce the loss of information [1] (when running with metrics guided by the generalization).
  Alternatively we can define the maximum and minimum generalization level, but that reduces the search space instead of just biasing it. An even worse choice would be to consider them insensitive so they are not manipulated at all, but this would ruin all privacy considerations.
- **Suppression limit:** percentage of records we allow the removal of. Necessary to prevent outliers from deteriorating their equivalence class's utility (making it more general and thus of broader scope / larger).
  We found that we can leave suppression limit at 100%, since ARX presents us with the result space, and full suppression will only be the optimum solution if it is the only solution.
  Nevertheless, we can also bias the suppression vs. generalization choice.
- **Geneneralization type:** when executing the anonymization, we can specify (depending on the model) whether to use a global or local transformation method. In global, all values are generalized to the same hierarchy level and thus we unnecessarily lose out on a lot of utility for

execution speed - for local transformation where different levels are applied for each class, it takes a lot longer.
- **Utility Measure:** we have many such metrics in ARX which will guide the anonymization process to the optimum transformation. Most are either based on the levels of generalization applied (eg. loss, precision), or on the equivalence class sizes (eg. average, discernibility).

ARX will optimize the utility metric under the models' restriction, and present us with the optimum (but also allows us to traverse the result space, to explore certain trade-offs).

Some solutions will be obviously better in both privacy and utility (dominate), but together a Pareto surface is formed from which it is up to us to choose the best one. The utility metric just turns the "2D" optimization problem into "1D".

*1) **k-anonymity**:* Configuring k-anonymity is relatively straightforward, given that `k` is the only parameter and its impact is easily verifiable in the outputted dataset/metrics.

As mentioned, `k` represents the size of the equivalence classes for which the contained records are indistinguishable when accounting only for the QIDs.

However, this `k` is a **minimum**, and some classes may end up larger - not enough variety in the original dataset, or the more likely scenario that too much generalization was needed.

In the dataset, there are is a record that is an outlier, with a single individual aged 96, while the next youngest is 83. This results in a lot of unneeded generalization. To solve this issue we can:

- admit suppression to remove such outliers
- use local generalization, where the 96 year-old is generalized with some other senior citizens, but the younglings not as much (ie. no longer 13+ year bucket sizes).
- (could) use LKC, with $l < |QID|$

A final important point, is that ARX forces us to select a privacy model for each sensitive attribute, but k-anonymity does not consider them at all. To "solve" this we marked them as insensitive.

*2) **l-diversity**:* As said before, with l-diversity we try to guarantee `l` distinct values for each sensitive attribute in the equivalence classes. Entropy based l-diversity exists, taking into account their distribution, but we'll leave this concern for t-closeness which offers a better solution.

Again we have a single parameter `l`, but this time not globally applicable to all sensitive attributes (as `k` was for QIDs). This gives us increased flexibility, but also a greater challenge in exploring all possibilities.

Each sensitive attribute has a different distribution and number of distinct values, and as such we explored giving different `l`'s to each.

Giving a concrete example, `charge_degree` has only two values - `M`(isdemeanour) and `F`(elony), meanwhile `c_charge_desc` has 389. Thus, `charge_degree` can only have an `l` of at most 2, where there could be equivalence classes with just murders in the `c_charge_desc` (1st/2nd degree). But, if `charge_degree`'s `l` is 2, we couldn't just have felonies (would need at least one misdemeanour).

*3) **t-closeness**:* Similarly to l-diversity, we specify `t` for each sensitive attribute. Here, the ideal values from a privacy point of view (minimal `t`) are only really achievable by features with few distinct values, or we risk exploding equivalence classes' sizes.

There are several variations of the Earth mover's distance supported by ARX, adequate to different attribute types [1]:

- **ordered ground:** "calculates distances based on the order of values" [1]. Applied to the numeric `decile_score` and `priors_count` features.
- **equal ground:** which considers all values equally distant. This one is adequate for categorical features, which even if represented numerically, their subtraction is not meaningful. This is the case of `charge_desc`.
- **hierarchical:** we did not use this one.

*4) δ**-presence**:* Again, δ_{max} represents an upper bound on the probability an individual is in the released dataset.

ProPublica's dataset contains COMPAS scores for Broward County, for the years 2013 and 2014, which is already a subset of the 18610 records they originally obtained [3]. However, we don't know these, so when applying δ-presence we'll have to consider a subset of the one we're working with, to properly verify δ's condition.

In ARX we can select a random sample of the dataset on which to apply this model, and as such we'll have an additional parameter - the **sampling probability**.

With δ-presence, we could know an individual was arrested, and as such would be in the *full* dataset, but we release a sample guaranteeing that each equivalence class has $(k*(1-s))/s$ non-released member records.

IV. Results Analysis

Tab.II contains all runs we'll describe this section. In it, there are configurations that provide max privacy at the expense of lower or null utility, or vice-versa. Also, we found solutions that explore a great balance between privacy/utility.

We will use the following metrics to compare different configurations. These are only a subset of those provided by ARX, but should give an adequate view into the privacy/utility trade-off:

- **minimal/average class size:** provide insight into the record-linkage threat - the lower the likelier.
- **prosecutor risk mode:** based on the % intervals provided in ARX 's *Analyze Risk - Distribution of risks (table)* view, useful to understand privacy gain.
- **% missing for `race/sex`:** provide insight into the data utility for fairness studies
- **utility measure value:** provides insight into the general utility of the original dataset, but depends on the one that is used

- **classifier's relative accuracy:** ie. comparing it in the original dataset with `is_recid` as the target feature (will they actually re-offend?). Useful to understand utility loss.

We annex (in `results/`) screenshots of the different ARX perspectives from which these metrics were extracted, for each configuration in Table II.

A. *k-anonymity*

1) Privacy vs. Utility: We started by varying the general parameters described in Sec.III-B, and used a low `k` of 3 to see how they impact privacy/utility without much *noise* from anonymization operations.

For this, `K3` (Tab.II) acts as a baseline, keeping all of ARX's defaults. We immediately notice the terrible average equivalence class size of 2057 (out of 3, with one of size 5385). It is clear something is *wrong* as our only restriction, `k`, is very small considering the size of our dataset.

Admitting **suppression** in `K3-S100`, we notice an immediate improvement - average class size of 53 - albeit at the cost of removing 64 records (outliers which were forcing other records to be generalized much more than they *should*).

However, the real improvement comes with local **generalization**, where we don't force the same generalization level to all of the QID's values. Here, we have an average class size of 3.41, much closer to the desired `k` of 3. This brings the cost of optimizing a much larger search space, which we control with the number of iterations it performs.

Attribute **weights**, which we used for prioritizing `race`/`sex` are not noticed on their own, but will impact the utility metric's decision of the best solution. In `K3-S100-L100-W`, there is now no generalization on `race` (which given its hierarchy, if it existed then it'd be missing).

This last configuration of the general parameters, is what we'll use moving forward when varying `k` and the utility measure, as it correctly prioritizes attributes, and its class sizes are close to the desired value (k=3).

We then increase `k` to 10, 20, 50 and 100. Obvious takeaways are the reduced reidentification risk, at the cost of increased utility loss.

Eventually, `k` is not satisfiable without suppressing outliers, as the generalization which they would force upon many other records, does not outweight their removal anymore. With k=50, the 96-year old discussed in Sec.III-B1 is suppressed, and with k=100 so too are another 52 records.

We end up choosing k=20 since it removes zero outliers. From then, we varied the utility measures in ARX (cf. Tab.II), considering:

- Height / Precision: penalize higher generalization levels
- Average Equivalence Class Size
- Discernibility: penalizes particularly deviant class sizes

However, none of them take into account attribute weights, often fully suppressed (unlike loss). We can confirm that by observing the % of missings in race/sex in those cases.

2) Re-Identification Risk: Not surprisingly, as `k` increases, the re-identification risk reduces.

The highest prosecutor risk (i.e journalist's) is based on the smallest equivalence class size ($1/min(|c|)$), and as such with k=20, we have 5% compared to the 33.3% presented in k=3 (privacy vs. utility tradeoff). Of course for extreme privacy, k=100 can be chosen for an even higher prosecutor risk of 1% at the expense of data utility.

Ideally from an utility point of view, the average prosecutor risk (ie. marketer's) would be equal to ($1/|k|$). However, it's lower in practice: for k=20 we have 4.212%, k=50 we have 1.669%, ... This is nevertheless good for privacy, since it gives us a margin to introduce considerations on sensitive attributes.

B. *l-diversity*

1) Privacy vs. Utility: We now change the sensitive attributes' `l`, starting with a baseline of 2 for each. When compared to previous K=20 without any l-diversity, we see almost no difference. Both configurations have zero records surpressed, same utility loss and prosecutor risk. Only the missing % of race and average class size is a bit higher in l-diversity. Of course this is expected, to satisfy the `l` we need more generalization. Therefore, it leads to an increase of the class sizes, explaining the higher missing values in race attribute.

Now we should consider different values for each, accounting for their distributions. For this we must restrict `l` to be both smaller (or equal) than `k` and than the number of distinct values for the specific sensitive attribute.

As such, for `charge_degree` and `is_recid`, `l` will be set to 2 since they are binary and decently balanced.

For the remaining attributes it's a bit trickier.

- For `decile_score` we have 10 distinct values (1 through 10), with relative frequencies between 0.05 and 0.2, but mostly balanced. As such we'll assign an `l` of 5, which ensures significant diversity even if it can result in some skeweness (unavoidable with distinct l-diversity).
- For `priors_count` we have 36 distinct values (between 0 and 38), more than with `decile_score` but with a very skewed distribution, so we'll also give an `l` of 5 to compensate.
- For `charge_desc` we have 389 distinct values, with some happening very few times. We'll leave `l` as 10, which should ensure good diversity in the crimes committed, given charge_degree's `l` ensures we have both felonies and misdemeanors.

From decile_scores with `l`=5, there may be equivalence classes where the minimum is score is 6. We tried increasing `l` to 8 but with worse results.

Note that these runs are very slow (up to 30min.), due to all the constraints faced by the anonymizer when optimizing.

2) Re-Identification Risk: As expected, the re-identification risk is even lower than 20-anonymity configuration. Since we have more generalization to satisfy the restrictions introduced by l-diversity, these bigger equivalence classes result in a lower re-identification risk. Once again, we see the privacy/utility tradeoff.

C. t-closeness

1) Privacy vs. Utility: The parametrization in t-closeness is far less obvious, as the the complexity of the distance metric (EMD) makes `t`'s impact much less palpable. It also takes way more time to tune, with worse results.

The default `t` of 0.001 is way too low, as it made the records fully generalized to make them have the same distribution (ie. a single equiv. class of size 6172). The only thing we learn from this run is that the relative classification accuracy is irrelevant as a comparison metric, and mostly dependent on the sensitive attributes (it was 103.84% without any QID).

We then increased it significantly to 0.1 across the board, which just led to the suppression of 1175, still leaving a single equivalence class.

Similarly to what we did with l-diversity, we adjusted `t` for the attributes with complex distributions. Several configurations were tested but none beat those with l-diversity.

2) Re-Identification Risk: Once again, as expected, since QIDs have an high level of generalization the re-identifications risks are lower than in the last scenarios.

D. δ-presence

For δ-presence, every configuration we tried either required using a very small sample, and thus lose out on a lot of data (utility), or it applied way too much generalization.

The reduced number of records wouldn't really be a problem, given that many similar datasets (when it comes to their application) are small (eg. *German* is approx. 1k, *Ricci* is approx. 100 [7]), but the problem comes from how *handpicked* these are to guarantee δ-presence.

However, it is an honorable mention for anyone who MUST achieve good privacy results, however poor the underlying data is.

V. Differential Privacy

A. Selection

The assignment proposed applying differential privacy for two functions, but ARX did not appear to support queries like those we had studied.

We identified several libraries/packages supporting these, namely Google's in C++[1] and Benjamin Rubinstein's in R[2], but we opted with the Python package `dp-stats` [9],[10].

B. Query Functions

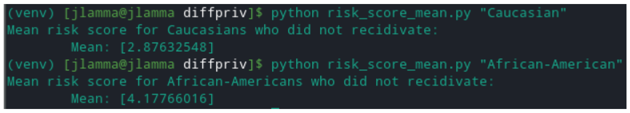
Continuing with our goal of supporting research in fairness, our query functions (annexed in `diffpriv/`) aim to identify racial bias in the COMPAS tool.

They will consider the individuals of a race that did not recidivate (`is_recid==0`), and operate on their **risk scores** that were automatically computed. We provide the **mean** (`risk_score_mean.py`) and the **histogram** (`risk_score_histogram.py`) of these values.

Besides being able to specify the race, the scripts we submit also allow the parameterization of ϵ and δ - differential-privacy specific parameters.

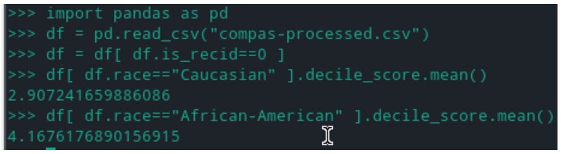
C. Analysis

Running the median query on African Americans and Caucasians, we appear to identify the suspected bias (also seen in the histogram query (Fig.4), although this one is shown for conciseness):



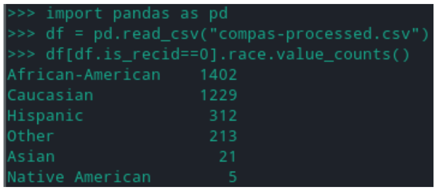
```
(venv) [jlamma@jlamma diffpriv]$ python risk_score_mean.py "Caucasian"
Mean risk score for Caucasians who did not recidivate:
        Mean: [2.87632548]
(venv) [jlamma@jlamma diffpriv]$ python risk_score_mean.py "African-American"
Mean risk score for African-Americans who did not recidivate:
        Mean: [4.17766016]
```

And even though the results are perturbed by differential privacy's Laplacian noise, they're not far off the actual values:



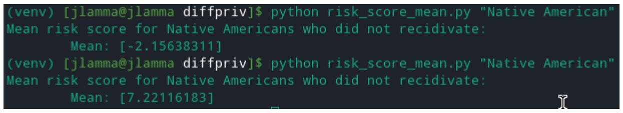
```
>>> import pandas as pd
>>> df = pd.read_csv("compas-processed.csv")
>>> df = df[ df.is_recid==0 ]
>>> df[ df.race=="Caucasian" ].decile_score.mean()
2.907241659886086
>>> df[ df.race=="African-American" ].decile_score.mean()
4.1676176890156915
```

However, if we look at the distribution of `race` in our dataset (Fig.2), the different frequencies will vary the scale of noise to be added in each case. Concretely, if we knew a Native American had been arrested, we would deduce a lot more about his score just from that average - but in DP the presence of a record should be (relatively) imperceptible through the queries. As such, we will need much more noise, leading to extreme deviations (Fig.3), often to invalid values.



```
>>> import pandas as pd
>>> df = pd.read_csv("compas-processed.csv")
>>> df[df.is_recid==0].race.value_counts()
African-American    1402
Caucasian           1229
Hispanic             312
Other                213
Asian                 21
Native American        5
```

Fig. 2: `race` distribution, for `is_recid==0`



```
(venv) [jlamma@jlamma diffpriv]$ python risk_score_mean.py "Native American"
Mean risk score for Native Americans who did not recidivate:
        Mean: [-2.15638311]
(venv) [jlamma@jlamma diffpriv]$ python risk_score_mean.py "Native American"
Mean risk score for Native Americans who did not recidivate:
        Mean: [7.22116183]
```

Fig. 3: Large noise for Native Americans

Important to note that the noise can be controlled with the ϵ, with higher values decreasing it.

Similarly, for the histogram query, we observe the noise added to each count (Fig.4), and also some bins representing more than one score to achieve the privacy requirements (for the less frequent races).

[1] https://github.com/google/differential-privacy

[2] http://www.bipr.net/diffpriv/

VI. Discussion

After analysing all the results, there is a configuration that stands out when thinking of the balance between utility vs privacy: `K20-L2-Score5-Desc10-Priors5`. Adding l-diversity to k-anonymity fixes the neglection of sensitive attributes mentioned in Sec.**??**, and considering each sensitive attribute's `l` independently gives us solid privacy assurances while not deteriorating their utility. Therefore, we achieve a good level of privacy while maintaining the usefulness of the dataset. Despite the lower utility than in the original dataset, in the l-diversity results it would be way harder to both identify a record or infer its attributes.

Comparing with the original dataset, we see a drastic reduction in the separation of the QID combination, from 100% to 2.474%. Furthermore, the highest prosecutor risk drops to 5% and the average to 3.599%, both down from 100%.

VII. Conclusion

Given the nature of the assignment - consulting service - we abstained from mathematical formalism, to instead provide intuitive explanations into the anonymization procedure undertook. With these in mind, and with the annexed resources referenced throughout, it should be simple to make personalized tweaks before publicizing the dataset.

References

[1] ARX - Data Anonymization Tool — A comprehensive software for privacy-preserving microdata publishing
https://arx.deidentifier.org/

[2] https://github.com/propublica/compas-analysis

[3] How We Analyzed the COMPAS Recidivism Algorithm — ProPublica
https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

[4] Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias. ProPublica, May, 23, 2016.

[5] Fung, Benjamin CM, et al. Introduction to privacy-preserving data publishing: Concepts and techniques. CRC Press, 2010.

[6] Valentim, Inês Filipa Rente. Assessing the Fairness of Intelligent Systems. MS thesis. 2019.

[7] Friedler, Sorelle A., et al. ”A comparative study of fairness-enhancing interventions in machine learning.” Proceedings of the Conference on Fairness, Accountability, and Transparency. 2019.

[8] Brickell, Justin, and Vitaly Shmatikov. ”The cost of privacy: destruction of data-mining utility in anonymized data publishing.” Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008.

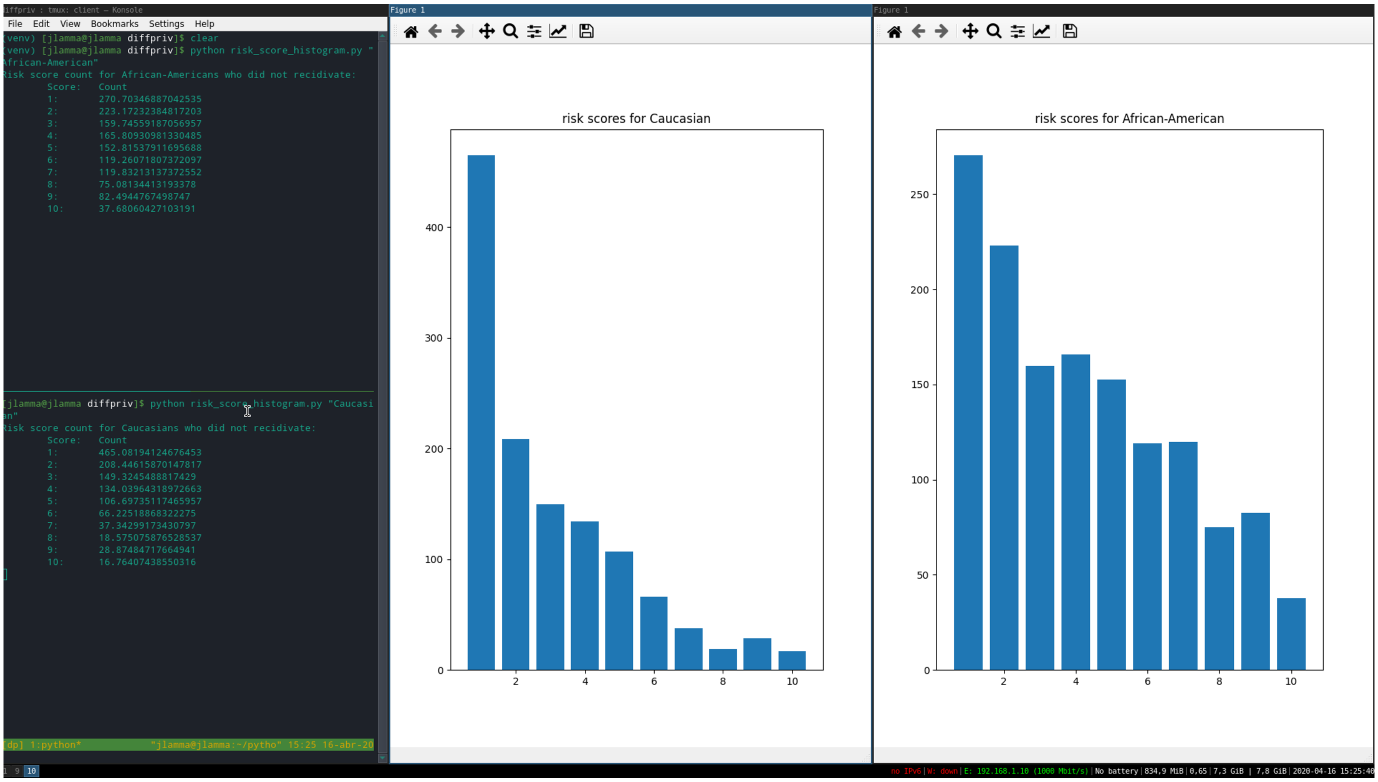
[9] dp-stats: A Python Library for Differentially-private Statistics and Machine Learning Algorithms - DPSTATS.pdf
https://www.ece.rutgers.edu/~hi53/DPSTATS.pdf

[10] https://gitlab.com/dp-stats/dp-stats

ID	Algorithm	Attribute Weights	Suppr. Limit	Generalization	Min/Avg Class	Pros. Risk	`race/sex` % Missing	Utility Measure	Rel. Classif. Acc.%
K3	3-Anonymity		0	Global	3/2057.333	]0.01;0.1]	100/100	0.59176 (Loss)	115
K3-S100	3-Anonymity		100	Global	3/53	]0.1;1]	1.04/1.04	0.025 (Loss)	111
K3-W	3-Anonymity	Race/Sex: 1.0	0	Global	3/2057.333	]0.01;0.1]	100/100	0.45397 (Loss)	108
K3-L100	3-Anonymity		0	Local (100 it.)	3/3.4137	]25;33.4]	1.16/0.14	1.533E-9 (Loss)	106
K3-S100-L100	3-Anonymity		100	Local (100 it.)	3/3.4137	]25;33.4]	1.16/0.14	1.533E-9 (Loss)	107
K3-S100-L100-W	3-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	3/3.4137	]25; 33.4]	0/1	7.7E-10 (Loss)	115
K10-S100-L100-W	10-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	10/12.221	]9;10]	1.554/0	5.1324E-8 (Loss)	111
K20-S100-L100-W	20-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	20/23	]4;5] (60.64)	3.1/0.9	9.4E-8 (Loss)	106
K20-S100-L100-W-AvgClassSize	20-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	20/30.6	]3;4] (74.42)	0.85/100	136 (AvgClassSize)	99
K20-S100-L100-W-Discernibility	20-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	20/126	]0.1;1] (71.11)	100/100	400 (Disc.)	107
K20-S100-L100-W-Height	20-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	20/25.609	]4;5] (51.08)	2.77/0.53	0.518 (Height)	109
K20-S100-L100-W-Precision	20-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	20/23.822	]4;5] (65.64)	11.22/53.79	0.0032 (Precision)	108
K50-S100-L100-W	50-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	50/59.91	]1;2]	8.08/1.11	5.6484E-7 (Loss)	111
K100-S100-L100-W	100-Anonymity	Race/Sex: 1.0	100	Local (100 it.)	100/118	]0.1;1]	8.03/2.69	1.25E-4 (Loss)	110
K20-L2	20-Anonymity 2-Diversity	Race/Sex: 1.0	100	Local (100 it.)	20/26.4	]4;5]	5.4/0.65	9.41E-8	112
K20-L2-Score5-Desc10-Priors5	20-Anonymity 2-Diversity	Race/Sex: 1.0	100	Local (100 it.)	20/28.6	]4;5]	10.5/0.016	9-41E-8	112
K20-L2-Score8-Desc10-Priors5	20-Anonymity 2-Diversity	Race/Sex: 1.0	100	Local (100 it.)	20/27.8	]4;5]	18.1/13.2	6.32E-5	116
K20-T0.001	20-Anonymity 0.001-Closeness	Race/Sex: 1.0	100	Local (100 it.)	6172/6172	]0.01;0.1]	100/100	0.6509	103
K20-T0.1	20-Anonymity 0.1-Closeness	Race/Sex: 1.0	100	Local (100 it.)	4997/4997	]0.01;0.1]	100/100	0.4708	101
K20-T0.1-Score0.2-Desc0.5-Priors0.2	20-Anonymity 0.1-Closeness	Race/Sex: 1.0	100	Local (100 it.)	67/171.444	]0.1;1]	36.9/20.3	6.311E-5	101
K20-T0.1-Score0.2-Desc0.65-Priors0.25	20-Anonymity 0.1-Closeness	Race/Sex: 1.0	100	Local (100 it.)	20/60.5	]0.1;1]	36.5/21.6	5.644E-7	102
K20-T0.1-Score0.2-Desc0.75-Priors0.25	20-Anonymity 0.1-Closeness	Race/Sex: 1.0	100	Local (100 it.)	20/46.6	]0.1;1]	34.73/21.01	5.644E-7	93
K20-T0.1-Score0.2-Desc0.75-Priors0.5	20-Anonymity 0.1-Closeness	Race/Sex: 1.0	100	Local (100 it.)	20/53.66	]0.1;1]	33.16/21.62	5.644E-7	101
K20DP	Delta Presence (0; 0.5) Sampling:0.15	Race/Sex: 1.0	100	Global	23/113	]0.1;1]	6.72/6.72	0.183	85

TABLE II: Better table visualization in https://6p27o.csb.app/

APPENDIX



```
(venv) [jlamma@jlamma diffpriv]$ clear
(venv) [jlamma@jlamma diffpriv]$ python risk_score_histogram.py "
African-American"
Risk score count for African-Americans who did not recidivate:
        Score:   Count
        1:       270.70346887042535
        2:       223.17232384817203
        3:       159.74559187056957
        4:       165.80930981330485
        5:       152.81537911695688
        6:       119.26071807372097
        7:       119.83213137372552
        8:       75.08134413193378
        9:       82.4944767498747
        10:      37.68060427103191
```

```
[jlamma@jlamma diffpriv]$ python risk_score_histogram.py "Caucasi
an"
Risk score count for Caucasians who did not recidivate:
        Score:   Count
        1:       465.08194124676453
        2:       208.44615870147817
        3:       149.3245488817429
        4:       134.03964318972663
        5:       106.69735117465957
        6:       66.22518868322275
        7:       37.34299173430797
        8:       18.575075876528537
        9:       28.87484717664941
        10:      16.76407438550316
```

```
[dp] 1:python*          "jlamma@jlamma:~/pytho" 15:25 16-abr-20
```

```
no IPv6|W: down|E: 192.168.1.10 (1000 Mbit/s)|No battery|834,9 MiB|0,65|7,3 GiB | 7,8 GiB|2020-04-16 15:25:40
```

diffpriv : tmux: client — Konsole
File Edit View Bookmarks Settings Help
1 9 10
Figure 1
risk scores for Caucasian
0
100
200
300
400
2
4
6
8
10
Figure 1
risk scores for African-American
0
50
100
150
200
250
2
4
6
8
10

Fig. 4: risk score histograms with differential privacy