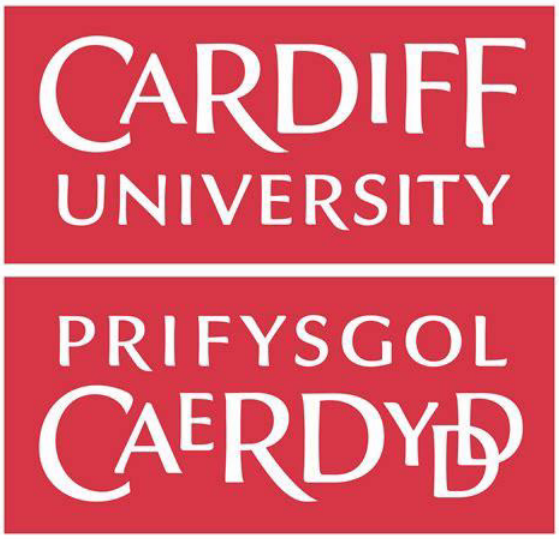


Cardiff University

School of Computer Science and Informatics

Implementation of a Data Privacy Protection Tool for Relational Data

CM3203 Final Year Project – 40 Credits

Final Report

Author: Jack Davies

Supervisor: Dr Jianhua Shao

Moderator: Dr Usashi Chatterjee

Academic Year: 2020/21

Abstract

With the increase in personal data collection, storage, and use by a growing number of corporations and organisations, there has been a corresponding rise in the population's concern for a right to privacy. Many methods have been developed in recent years to allow for said data to be shared and used whilst maintaining the privacy of the individual – achieved primarily by a process known as *Privacy Preserving Data Publishing* (PPDP). However, there has been relatively little study into the utility of this *privacy-preserved* data. This report attempts to examine the utility of data anonymised using one of the most recognised methods – *k-Anonymity.* In it, I detail the implementation of the Mondrian *k*-Anonymity algorithm used to anonymise the ADULT dataset. I describe the use of the ID3 algorithm to measure the classification accuracy of decision trees trained on anonymised, and non-anonymised data in multiple conceivable scenarios, and compare these results to measurements of relevant metrics, with an evaluation of the results. The findings show that, in general, a loss of information can be expected from data anonymisation. However, the extent of this loss can be considered minimal in most realistic situations.

Acknowledgements

I would like to thank Dr Shao for the supervision and guidance provided to me over the course of the project, and for encouraging independent thought on the ideas and concepts within. I would also like to thank my family for pushing me to study from a young age. Finally, I would like to thank my fiancée Tamsyn for her continued support, helping me to repeatedly lift myself out of self-doubt and encouraging me to reach ever greater heights.

Table of Contents

Table of Figures

1 - Introduction

Since the turn of the Millennium and the advent of the Information Age, there has been an explosion in data capture and storage capabilities. Everything from your traditional medical records to your recent purchases and interests are stored on a disk in a data centre at a location likely unknown to yourself. Owing to the rise of smart phones, social media, wearable technology, and the needs of businesses for analytical models, you would be hard-pressed to find a living person that has not had personal data collected on them.

Along with this rise in data collection, there has been a notable shift in public interest towards a more private society, likely due to a greater awareness of the sheer extent of data collection by large organisations and corporations. High profile cases of misuse such as the Facebook-Cambridge Analytica scandal [Lapowski, 2019] have boosted this public awareness of the magnitude of everyday data collection.

Due to this, we have seen a push for more regulation regarding data protection and personal privacy. The European Union's GDPR [European Parliament, 2016], implemented in 2018, has helped ensure the rights of citizens within the EU to control over their personal data, with the post-Brexit UK following in the EU's footsteps in this regard. This should be seen as a positive step; however, such regulations have further restricted the ability of legitimate researchers to access this potentially crucial data [Storgaard, 2019][Maldoff, 2016][Mooney, 2019].

The inherent value of this data is contained within its usefulness to research. Therefore, there is a necessity for a method of ensuring the continued distribution of data that maintains the privacy of the individual – a practice known as Privacy Preserving Data Publishing (PPDP)[Fung, 2007]. One of the primary approaches to PPDP is through the use of a process called *data anonymisation*. Data anonymisation is the practice of modifying a raw dataset containing records that could be linked to an individual, such that these records cannot be linked to an individual with any reliable accuracy.

There have been many approaches to data anonymisation proposed in recent years. Such approaches, described as *privacy models*, include *k*-Anonymity [Samarati & Sweeney, 1998], ℓ-Diversity [Machanavajjhala et al., 2007] and Distributional Privacy [Blum et al., 2013], among others. All of which are useful in different situations. This report will focus on the use of the *k*-Anonymity method using the Mondrian Multidimensional Algorithm [LeFevre et al., 2006] for implementation.

Given that the purpose of data anonymisation is to allow the data to be shared amongst third parties, it is crucial that the usefulness of the data itself is maintained post-anonymisation. Therefore, in this report, I will also examine the utility of anonymised data in decision tree classification using the well-established ID3 algorithm [Qinlan, 1986]. In the ensuing pages, I consider four possible scenarios:

i) Classification of non-anonymised data using decision tree trained on non-anonymised data
ii) Classification of anonymised data using decision tree trained on anonymised data
iii) Classification of anonymised data using decision tree trained on non-anonymised data
iv) Classification of non-anonymised data using decision tree trained on anonymised data

Each scenario will be emulated in a series of experiments where the classification accuracy obtained using the ID3 algorithm will be measured. The results for each scenario will be presented for

comparison and examination, accompanied by measurements of relevant metrics that could conceivably be used to estimate the utility of the data. It is envisioned that by comparing the classification accuracy of records in a given data set in each of these scenarios, insights can be made into the utility of anonymised data in data mining and analysis.

1.1 - Basic Definitions

More detail on the following will be provided in the subsequent pages, this can simply be used for reference.

- Quasi-Identifier (QID) Attribute Set – A set of attributes in a data set that, when joined with other data sets, could lead to information that identifies a distinct individual within the data set.
- *k*-Anonymity – A property possessed by an anonymised data set where each record in the set is indistinguishable from at least *k*-1 other records over the QIDs.
- Generalisation – The process or outcome of converting a set of different values within the domain of a given attribute in a data set to a single value that encapsulates each and every different value.
- Equivalence class – A set of records in a single data set that are equivalent across the QIDs.

1.2 - Aims

The original aims of the project can be found in the initial report. During the project, it was necessary to update these aims. Here is the updated list of aims:

1. Implement a k-Anonymity tool for relational data
    a. Develop a deep understanding of the Mondrian *k*-Anonymity algorithm
    b. Implement the algorithm with the Python programming language, ensuring the parameters can be easily modified by the user
    c. Test the tool on multiple data sets to ensure accuracy and reliability
2. Measure the usefulness of the anonymised data provided by the tool using a machine learning algorithm
    a. Implement the ID3 decision tree algorithm with the Python programming language such that it can be used with data anonymised by the tool created in (1)
    b. Ensure accuracy and reliability of ID3 algorithm by comparing results from an experiment with a publicly available implementation's results
    c. Perform the following tests using the k-fold cross-validation method:
        i. Find classification accuracy for non-anonymised data using a decision tree trained on non-anonymised data
        ii. Find classification accuracy for anonymised data using a decision tree trained on anonymised data
        iii. Find classification accuracy for non-anonymised data using a decision tree trained on anonymised data
        iv. Find classification accuracy for anonymised data using a decision tree trained on non-anonymised data
    d. Repeat the above experiments for different values of *k* (*k*-Anonymity) and different QIDs
3. Take measurements of various relevant metrics for use in comparison to the data obtained in (2)
    a. Research metrics that could be appropriate for estimating classification accuracy

   b. Implement scripts to obtain measurements of the chosen metrics from anonymised data sets
   c. Collate metric results and use to provide further insight on results in (4)
4. Collate and evaluate the results collected from (2) and provide any possible insight into the usefulness of anonymised data regarding analysis
   a. Collate collected data, providing relevant figures and graphs in the report
   b. Examine the data and compare the effectiveness of anonymisation relative to the usefulness of the data in analysis
   c. Provide a report of the evaluation and any insights found

2 - Background

2.1 - Motivation & Context

Privacy Preserving Data Publishing (PPDP) has become a well-studied field over the last decade or so. Data can come in a variety of forms, and there are different approaches to PPDP for different forms of data. There has been a recent increase in uptake for novel methods of data storage such as graph databases, which are very useful for applications like social networks. However, the majority of data is still stored in relational databases. For this reason, this report will focus entirely on data in relational form. Many different models and algorithms to address PPDP for relational data have been proposed. A good overview of the most popular methods today can be found in [Majeed & Lee, 2020, pp. 8520-8524]. The general idea behind these methods is to produce an anonymised version of the raw relational data table, one that will prevent attackers from discovering the identity of the persons the data is describing – this process is known as *data anonymisation*.

One of the most cited, extensively studied and implemented methods of data anonymisation is *k*-Anonymity [Samarati & Sweeney, 1998]. Details on how this method works can be found in the next section [2.2]. Due to its relative simplicity in theory and implementation, *k*-Anonymity will be the method investigated in this report.

Of course, there would be no reason to anonymise data if it were to be simply locked away in a data store never to be seen by anyone. Data is collected because it is useful in analysis and research. Necessarily, this utility must be retained post-anonymisation. While there has been some study into the utility of data post-anonymisation [See: *Related Works* section – 2.4], it tends to be performed simply as a means of evaluating new methods compared to established methods. There is a real gap in the literature for a comprehensive study into the utility of data anonymised by established methods.

For the data publisher, it is generally preferable for them to require no understanding of how the data they are publishing will be used. Indeed, it is often the case that data publishers will simply make their data available publicly, allowing anyone to use it as needed. Doing so is certainly a benefit to publishers, however, it can be a benefit to consumers of the data if publishers simply have a sense of how data anonymisation is affecting the data's usefulness in common situations.

Considering the above, and also time constraints, this report will not seek to evaluate the *general* utility of anonymised data. Instead, it will investigate a common use for personal data – data mining, where the goal is to extract patterns and contextualise information in order to infer knowledge from the data. One aspect of data mining involves machine learning, that is classification. Classification can be used to make predictions based on patterns established from previous data. This report will investigate the utility of anonymised data in building and using a *Decision Tree* classifier.

2.2 - Preliminary Information

As mentioned, this report will focus on the process of data anonymisation through the use of *k*-Anonymity. *k*-Anonymity is a concept first introduced by Sweeney and Samarati in a paper titled *"Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression"* [Samarati & Sweeney, 1998]. In the aforementioned paper, a table is said to provide *k*-Anonymity *"if attempts to link explicitly identifying information to its contents ambiguously map the information to at least k entities"* [Samarati & Sweeney, 1998, pp. 1]. This can be achieved by ensuring that within a data set, the *quasi-identifier* attribute set values are modified such that each unique record is equal to *k*-1 other records. A *quasi-identifier* (QID) is an attribute in a data set that is publicly discoverable (e.g., date of birth via social media profile), which can be used to link a particular record in the data set to the individual the record is attributed to.

Figure 1 shows three tables at different stages of anonymisation. Figure 1a shows the raw data. The QIDs are shown in yellow, sensitive data is shown in blue and an explicit identifier is shown in orange. A naïve approach to anonymisation is shown in 1b. However, simply removing the explicit identifier from the table does not sufficiently anonymise the dataset. An individual who knows the age and postcode of Alice, both publicly discoverable attributes, could accurately determine that Alice has cancer. Figure 1c shows a *k*-Anonymisation of the table. The QIDs of each record have been modified where necessary by a process called *generalisation* to ensure that a given record has QID values equal to at least *k*-1 other records within the table. In this case *k*=2; making this a 2-Anonymisation.

A common approach to generalisation is through the use of a taxonomy tree, where values are abstracted up the tree. An example of this can be seen in Figure 2; 2a shows an example taxonomy tree for the *Occupation* attribute, 2b shows how this could be applied to records in a table. Taxonomy-based generalisation may be more effective for privacy. However, this is only the case if the taxonomy is not released with the anonymised data. It is entirely possible that the taxonomy hierarchy will be publicly available as developing a bespoke taxonomy for each data set generates a great deal of overhead for the data publisher. Furthermore, a taxonomy cannot be effectively used for numerical attributes like *Age*. An alternative method, set-based generalisation, is simpler but still effective. To generalise a set of values, you only have to place each attribute value within the partition into a subset (a "value-set") and use that as the value. This is the type of generalisation seen in Figure 1. Overall, taxonomy-based generalisation would seem to provide no real benefit over the alternative and only serves as an obstacle to implementation. For this reason, the implementation in this report uses set-based generalisation.

The algorithm chosen for implementing *k*-Anonymity in this report is *Mondrian Multidimensional k-Anonymity* [LeFevre et al., 2006]. Whilst there are other popular potential options [LeFevre et al., 2005] [Bayardo, R. J. & Agrawal, R., 2005] [Sweeney, L., 1998], Mondrian is a well-balanced algorithm in that it is relatively simple to implement, but still effective. It is also one of the most cited *k*-Anonymity algorithms and has been the subject of numerous evaluations [See: *Related Works* section – 2.4]. The details on this algorithm can be found in the *Approach* chapter [3.2].

Figure 1

	Name	Age	Postcode	Illness
rec1	Alice	28	CF24 3BW	Cancer
rec2	Bob	30	CF24 3BW	Appendicitis
rec3	Connor	29	CF23 5AB	Diabetes
rec4	Denise	29	CF23 7RZ	Flu
rec5	Ethan	30	CF23 7RZ	Broken nose

Fig. 1a

	Age	Postcode	Illness
rec1	28	CF24 3BW	Cancer
rec2	30	CF24 3BW	Appendicitis
rec3	29	CF23 5AB	Diabetes
rec4	29	CF23 7RZ	Flu
rec5	30	CF23 7RZ	Broken nose

Fig. 1b

	Age	Postcode	Illness
rec1	[28, 30]	CF24 3BW	Cancer
rec2	[28, 30]	CF24 3BW	Appendicitis
rec3	[29, 30]	[CF23 5AB, CF23 7RZ]	Diabetes
rec4	[29, 30]	[CF23 5AB, CF23 7RZ]	Flu
rec5	[29, 30]	[CF23 5AB, CF23 7RZ]	Broken nose

Fig. 1c

Figure 1 - Example of k-Anonymity

Figure 2

Fig. 2a

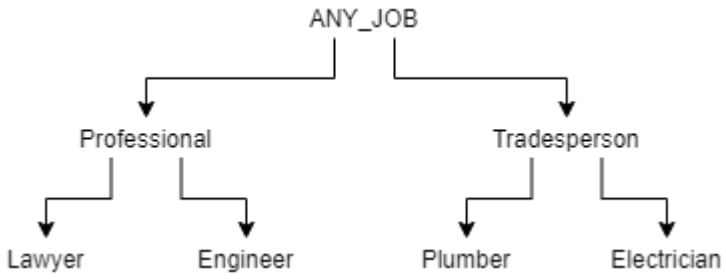


Fig. 2b

	Age	Occupation
rec 1	29	Engineer
rec 2	29	Lawyer

	Age	Occupation
rec 1	29	Professional
rec 2	29	Professional

Figure 2 - Example of taxonomy generalisation

In order to test the utility of a set of data anonymised in the fashion described above, we will use a well-established machine learning algorithm using decision trees for classification. This algorithm will be the ID3 algorithm first outlined by Quinlan in the paper titled *"Induction of Decision Trees"* [Quinlan, 1986]. This particular decision tree algorithm is today considered to be obsolete and has been succeeded by the improved *C4.5* [Quinlan, 1993] and other algorithms. However, considering the primary goal of this report is essentially to evaluate the *difference* in usefulness between non-anonymised and anonymised data sets, the simpler ID3 algorithm will be sufficient. The details on this algorithm can also be found in the *Approach* chapter [3.3].

The data set used for the experiments in this report will be the *ADULT Data Set* [Dua & Graff, 2019]. First extracted in 1994, it has become the standard for evaluation of anonymisation algorithms and a popular choice for evaluation of machine learning algorithms. Furthermore, it is used in many of the related works described in a later section [2.4], making it the prime candidate data set for the experiments in this report. It is primarily available from the UCI machine learning repository [Dua & Graff, 2019]. However, for this report, I obtained the data set from Kaggle [Kaggle, 2016]. as it was provided in a format that better suited my implementation of the *k*-Anonymity and ID3 algorithms.

2.3 - Metrics

The overwhelming majority of works similar to this report base their evaluation of anonymisation algorithms on measurements of certain metrics. There are many general and special purpose metrics to measure the utility of anonymised data, as well as other elements. Along with the primary method of measuring accuracy in decision tree classification in order to evaluate utility, it would seem appropriate to also evaluate the measurements of a selection of these metrics. This will not only allow more comparison between this report and past or future work, but it will also allow for an evaluation of the efficacy of the selected metrics for estimating decision tree classification utility as a side outcome of this report.

One of these metrics is the so-called *discernability metric* (DM) [Skowron & Rauszer, 1992]. This metric tries to measure information loss by applying a penalty for each record in an equivalence class using the formula:

$$DM = \sum_{E \in T} |E|^2$$

Where *E* is an equivalence class and *T* is the table being evaluated. This is a widely used metric for evaluation of anonymisation algorithms and, on the surface, this would appear to be a good selection to base measurement of anonymised data utility on. However, in a later chapter [5.2], I will show how this particular metric may not be very reliable for estimating the utility regarding classification.

Another, named *ILoss* [Xiao and Tao, 2006, pp. 7], is a metric that tries to consider the information loss of generalisation. In Fung et al., it states that "*ILoss measures the fraction of domain values generalized by [a generalised value] v_g"* [Fung et al., 2010, pp. 22]. The *ILoss* measurement for a specific generalised value is given by:

$$ILoss(v_g) = \frac{|v_g| - 1}{|D_A|}$$

Where *A* is the attribute of the value v_g, $|v_g|$ is the number of values within the domain of *A* that are descendants of v_g, and $|D_A|$ is the total number of values in the domain of *A*. The total *ILoss* for a

given table can then be found by simply summing the measurement for each generalised value in the table. This metric appears to be a more suitable measure of anonymised data utility as it considers individual generalisations when calculating information loss. Indeed, I show later in this report how this metric could be considered a reliable estimate for utility regarding classification in certain scenarios.

The *ILoss* metric falls short in that it does not consider the size of the entire data set. This means that it is not useful for comparing different data sets. It is also general-purpose, meaning that for any given use for anonymised data, the metric will have a limit in its reliability. A further alternative is the *Classification Metric* (CM) [Iyengar, 2002, pp. 282]. This is a specialised metric designed specifically for measuring utility regarding classification, it also takes into account the size of the data set through normalisation. The metric is defined:

$$CM = \frac{\sum_{r \in T} penalty(r)}{N}$$

Where *r* is a row in the table *T*, *N* is the total number of rows and the *penalty* function is defined:

$$penalty(r) = \begin{cases} 1 - \ if \ r \ is \ suppressed \\ 1 - \ if \ class(r) \neq majority(E(r)) \\ 0 - \ all \ other \ cases \end{cases}$$

Here *E(r)* is the equivalence class record *r* belongs to. In this report, *suppression*, another method of anonymisation different to *generalisation*, is not used. Therefore, we can ignore the first penalty case. The case where the class of record *r* is not the majority of its equivalence class adds a penalty if we have an equivalence class that is not homogeneous. Iyengar states *"Rows in a [equivalence class] G with different class labels cannot be discriminated using the [QIDs]. Therefore, for accurate classification, it is preferable if all the rows in G have the same class label"* [Iyengar, 2002, pp. 282]. This is intuitive, as if you consider an equivalence class with numerous classes represented, you cannot know which class a particular QID value belongs to in that instance. So, it is natural that this could lead to misclassification. For example, consider Figure 1c again, and records 1 and 2. The generalisation for age in this equivalence class is [28,30], but the illness for each record is different. If both records were equivalent in illness, and this was the class you were trying to determine, you would know with certainty that at least two persons of ages 28 and 30 had that illness, meaning either age chosen to finally represent the record when training the classifier will be assigned correctly. A side note to this; if an equivalence class is homogeneous, a so-called *Homogeneity Attack* [Aggarwal & Yu, 2008, pp. 26] could be performed on the data set. This is bad for privacy; however, this report's main aim is to evaluate the utility of anonymised data, not privacy alone. Overall, this would appear to be a very useful metric to use in this report. Results given later in the report indicate that this is indeed the case.

2.4 - Related Work

Tang et al. [2010] make a concentrated attempt at ensuring anonymised data is still useful. It suggests a novel utility-based *k*-Anonymisation algorithm and evaluates the utility of data anonymised by it in comparison to the Mondrian algorithm and one other. It shows that its suggested utility-based algorithm is superior to the other algorithms tested. However, it does so by using a combination of metrics including the aforementioned *CM*, and its own measure of *query answerability*. It does not actually perform classification experiments as this report proposes to do.

Ayala-Rivera et al. [2014] perform experiments on several *k*-Anonymity algorithms including the Mondrian algorithm used in this report, using the ADULT data set also used in this report, in order to

measure the efficiency and utility of the different algorithms for practitioners. Its main aim appears to be to find the best algorithm in general, not for a specific purpose. It shows inconclusive results on which algorithm is the best and interestingly suggests that different algorithms may be useful for different purposes. Again, the experiments here are purely metric-based and there is no measurement of actual classification accuracy or any other type of data mining utility.

Li et al. [2011] evaluate a modified version of the Mondrian algorithm (*InfoGain Mondrian)* using a decision tree classification accuracy model similar to my own, again on the ADULT data set. It does so in order to compare the new algorithm it introduces (IACk), to the more well-established algorithms, showing that IACk outperforms Mondrian in multiple classification tasks, including with decision trees. As such, it differs from this report in that it does not go into the same level of depth as is proposed here regarding decision tree classification. It also differs in how the experiment is set up, using a different decision tree algorithm, and as mentioned, a modified version of Mondrian.

Shao & Beckford [2017] attempt to evaluate the decision tree classification accuracy of data anonymised using the Mondrian algorithm. It does so by performing a series of experiments using the ADULT data set to train and test an ID3 decision tree. It shows that whilst there is a degradation of classification accuracy for anonymised data sets compared to non-anonymised, the degradation is minimal. This is a very similar study to the one described in this report. However, it differs in that this report considers multiple possible scenarios for training and testing of the ID3 decision tree. It also differs slightly in the implementation of the ID3 algorithm; this report utilises a random, and statistical approach to building the decision tree in order to perform a comparison, whereas the mentioned paper uses a purely random approach. Furthermore, the mentioned paper only considers the *Strict* version of the Mondrian algorithm, whereas this report considers both the *Strict* and *Relaxed* versions of the algorithm. Nonetheless, the setup and overall objectives of the mentioned paper are in line with this report and, as such, this report may be considered an extension of it.

In summary of the related work, it appears that there has not been a great deal of deep investigation into the utility of anonymised data, particularly regarding data mining and decision trees. The majority of related work investigates this using an array of general-purpose or specific-purpose metrics. Most other work that tries to examine the utility of anonymised data in data mining tends to do so only to make a surface comparison to a newly proposed algorithm. This gives a motivation for the work in this report which appears to be a more in-depth study of the utility of anonymised data regarding classification analysis.

3 - Approach

In this chapter, I will detail the approach to implementing the *k*-Anonymity tool and ID3 decision tree classifier, the pre-processing required to prepare the ADULT data set for the experiments, and the methodology of the experiments themselves. Firstly, however, I will detail the general approach to solving what I believe to be the main aim of the project – evaluating the utility of anonymised data.

3.1 - Solving the Problem

The primary method of evaluating anonymised data utility in this report is through measuring decision tree classification accuracy. By measuring the classification accuracy of the decision tree built and tested on data in its raw, non-anonymised form, it can then be compared to the classification accuracy of a decision tree involving anonymised data. The evaluation will consider four possible scenarios:

i) Classification of non-anonymised data using decision tree trained on non-anonymised data – this will serve as the baseline case for decision tree classification accuracy in this report. Whilst data anonymisation is necessary for the distribution of data amongst third parties, an analysis may still be performed on non-generalised data collected in-house, thus this scenario is still commonplace.

ii) Classification of anonymised data using decision tree trained on anonymised data – this would be considered the standard approach to data mining using decision trees for most third-party researchers. Due to the discussed reasons, the researcher will not have access to the original data and would need to build trees and perform classification over anonymised datasets.

iii) Classification of anonymised data using decision tree trained on non-anonymised data – this would be considered a less common scenario; however, it is still possible. Consider the case where a government organisation is legally allowed access to non-anonymised data collected through legislative means. However, the same organisation may not be allowed access to original data collected via private corporations, such as through social media. In this case, the organisation could train their classification model with the non-anonymised data, for use on the anonymised data from the social media company.

iv) Classification of non-anonymised data using decision tree trained on anonymised data – again, this would be considered a less common scenario; but still possible. For example, it is possible that a group of research organisations may cooperate to produce a single anonymised dataset containing data collected from each organisation. This data could then be shared amongst all organisations involved in the project. Each organisation would then be able to create decision tree models based on a much larger dataset than before. They could then use these models to classify future data collected themselves.

By considering these four possible scenarios, a more comprehensive evaluation can be performed.

Not all *k*-Anonymisations are created equal. Inherently, the *k*-value and QID set can be changed to whatever the data publisher deems appropriate. Therefore, it would also be necessary for a comprehensive evaluation to consider a range of *k*-values and QIDs. In theory, this will provide data publishers with some insight into optimal *k*-values and an optimal amount of QIDs to consider using when anonymising their data. Note that no real-life data publisher will be publishing the ADULT data set, so the actual QIDs and count thereof will not be particularly relevant. However, if we consider the *type* of data each QID added represents (numerical, categorical, etc.) and the count as a

percentage of the total number of attributes, this may be more useful for publishing anonymised data in general.

In addition to the above, there are two other possibilities to consider. Each involves an aspect of one of the two main algorithms used in this report. Firstly, the Mondrian *k*-Anonymity algorithm is implementable in two different modes: *Strict* or *Relaxed*. Both are useful in different situations, and therefore it is important in a comprehensive evaluation to examine both. Secondly, when building decision trees using the ID3 algorithm, it will not be possible to use generalised values as branches in the tree. These generalised values will need to be *mapped back* to specific values. One approach to this is to randomly map the generalisation back to any of the values in its value-set. A second approach could be to consider the effect of the publisher releasing basic statistics from the original data set and how we could use these statistics to map back values more accurately. More details on these methods can be found in the ID3 section of this chapter [3.3].

Another aspect of evaluation previously suggested is to analyse measurements of established metrics. In this report, for each anonymisation used, measurements of the *Discernability Metric* (DM), *ILoss*, and the *Classification Metric* (CM) will be taken. These measurements can then be compared to each other, and maybe more interestingly, compared to the actual classification accuracy measurements. This will hopefully provide some insight into how useful these metrics are in estimating the utility of anonymised data.

This would appear in principle to amount to a relatively comprehensive investigation of the utility of anonymised data in classification using decision trees. Further details on all of the above will be provided in the following pages of this chapter.

3.2 - Mondrian *k*-Anonymity

As mentioned, in this report it was decided that the Mondrian multidimensional algorithm [LeFevre et al., 2006] would be most suitable for the implementation of *k*-Anonymity.

An overview of the Mondrian *k*-Anonymity algorithm can be seen in Figure 3. Initially, the input data set is repeatedly cut into partitions along all viable dimensions until a minimal partition size is reached, with each one having at least *k* records. Following this, each partition is generalised, and the resulting generalised partitions are then merged into one final anonymised data set which can then be published. The algorithm is implementable in two modes: *Strict* or *Relaxed*. The pseudocode for the *Strict* algorithm can be seen in Algorithm 1, following which is a breakdown of this algorithm and the *Relaxed* algorithm.

Figure 3

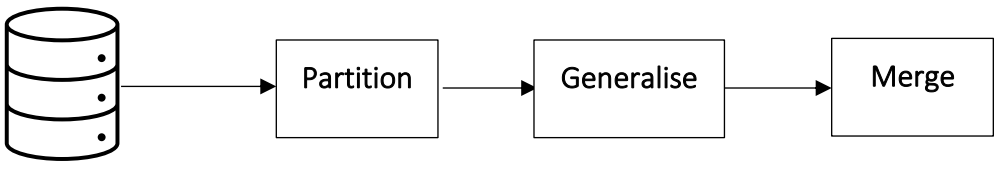


Figure 3 - Mondrian k-Anonymity Overview

Algorithm 1 – Strict Mondrian Multidimensional Algorithm (LeFevre et al. 2006)

```
1.  AnonymiseStrict(data set D)
2.  if (D cannot be partitioned):
3.          return D
4.  else:
5.          X_i = chooseAttribute(D)
6.          F = frequencySet(D, X_i)
7.          pv = median(F)
8.          lhs = {t ∈ D | t.X_i ≤ pv}
9.          rhs = {t ∈ D | t.X_i > pv}
10. return AnonymiseStrict(lhs) ∪ AnonymiseStrict(rhs)
```

The Mondrian algorithm is a recursive algorithm, the data set *D* used as input is initially the entire data set that requires anonymisation, however, in recursive calls, it is a subset of this. It is also a greedy algorithm, meaning that it will try to find a solution by simply selecting the current optimal option in each iteration.

The algorithm in both *Strict* and *Relaxed* modes will first determine if the data set *D* can be partitioned (line 2). LeFevre et al. on allowable multidimensional cuts state: *"Consider multiset P of points in d-dimensional space. A cut perpendicular to axis X_i at x_i is allowable if and only if Count(P.X_i > x_i) ≥ k and Count(P.X_i ≤ x_i) ≥ k"* [LeFevre et al., 2006, pp. 4]. Figure 4 shows a visual representation of a *Strict* partitioning for a *2*-Anonymisation. Notice that cuts along each axis are only possible in the positions shown, a cut at any other position results in a region with fewer than *k* points. The *Strict* and *Relaxed* algorithms differ in how the partitioning is performed. A *Strict* partitioning does not allow intersecting values between the two subsets of *D* resulting from a cut. A *Relaxed* partitioning allows intersecting values, for example, a cut along the axis of attribute *Age* could result in two partitions with generalised values [28,29] and [29,30], although this is not possible in the example shown in Figure 4. In theory, the *Relaxed* algorithm will result in a greater number of possible cuts, leading to smaller equivalence classes. Note also that every partitioning possible in the

Figure 4

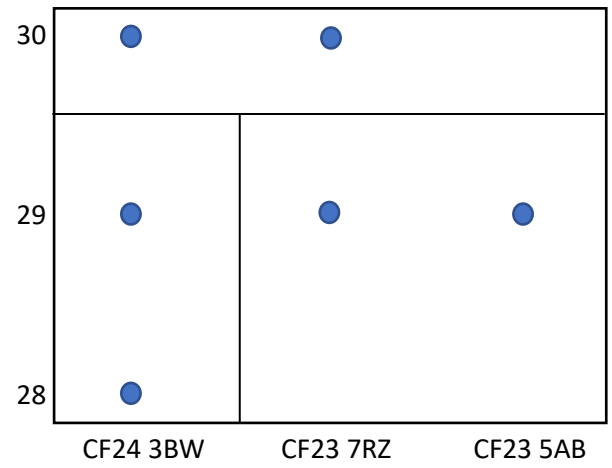


Figure 4 - Visualisation of Mondrian Partitioning

Strict algorithm is also possible in the *Relaxed* algorithm, this is not the case in the reverse. If *D* cannot be partitioned, then it is simply returned.

Once it has been established that data set *D* can be partitioned, the dimension (attribute) to attempt partitioning on needs to be selected (line 5). According to the literature, "*We have some flexibility in choosing the dimension on which to partition. As long as we make an allowable cut when one exists, this choice does not affect the partition size upper-bound*" [LeFevre et al., 2006, pp. 6]. The method suggested in the literature is to simply choose the dimension with the widest range of values. This is the method used in this report's implementation.

Line 6 establishes a frequency set *F* of all values belonging to attribute X_i in data set *D*. The order of the values in *F* is alphabetical for categorical values or ascending for numerical values. It is necessary to establish a consistent order in this because line 7 finds the median value *pv* of frequency set *F*. The median value, or "pivot", is then used as a basis for partitioning the data set *D*. The median is found in the standard way.

For the *Strict* algorithm, the process is now straightforward: simply take all records with a value for attribute X_i less-than or equal to *pv* in its position in the frequency set *F* and place them in a partition *lhs* (left-hand side). All remaining records, that is, records with values for X_i greater-than *pv* in its position in *F*, are placed in the partition *rhs* (right-hand side). An example of how this works can be seen in Figure 5. 5a shows a frequency table of records with values of a given attribute X_i. 5b shows how records are partitioned in the *Strict* algorithm.

In the *Relaxed* algorithm, partitioning is slightly more complex. Firstly, records added to *lhs* are those with values for X_i *strictly* less-than *pv* in *F*. In the same way, records added to *rhs* are those with *strictly* greater values for X_i. Values where $X_i = pv$ are added to a temporary partition *med*. Values in *med* are then shared between *lhs* and *rhs* in the fashion shown in Figure 5c. This results in the two partitions containing intersecting values for X_i.

With the data set *D* successfully partitioned into *lhs* and *rhs*, all that remains is to recursively partition *lhs* and *rhs*. The final result of this will be a set of partitions. Values for QIDs in each partition are generalised forming a set of distinct equivalence classes which can then be merged to construct the *k*-Anonymisation of the input data set.

Figure 5

Fig 5a.

#	Value	Frequency
0	A	10
1	B	20
2	C	6
3	D	80
4	E	35

Fig 5b.

$$Total\ Records = 151$$
$$Median = \frac{151 + 1}{2} = 76$$
$$pv = D$$
$$lhs = \{rec \in dataset \mid rec_{Xi} \in \{A, B, C, D\}\}$$
$$rhs = \{rec \in dataset \mid rec_{Xi} \in \{E\}\}$$
$$|lhs| = 116$$
$$|rhs| = 35$$

Fig 5c.

$$lhs = \{rec \in dataset \mid rec_{Xi} \in \{A, B, C\}\}$$
$$med = \{rec \in dataset \mid rec_{Xi} = pv\}$$
$$rhs = \{rec \in dataset \mid rec_{Xi} \in \{E\}\}$$

The records in *med* are distributed into *lhs* and *rhs* by the following pseudocode:

$$\textbf{while } |lhs| < Median:$$
$$\text{NextRecord}(med) \rightarrow lhs$$
$$rhs = med \cup rhs$$

Where *NextRecord* moves the next record in the set to another set. This results in:

$$|lhs| = 76$$
$$|rhs| = 75$$

Figure 5 - Mondrian Partitioning Example

3.3 - ID3 Decision Trees

For this report, the ID3 algorithm was selected as the means of decision tree classification. It is described in detail in [Quinlan, 1986]. However, the implementation in this report is based on the version described in [Mitchell, 1997, pp. 55-60]. *Algorithm 2* shows the pseudocode of the basic ID3 implementation.

Algorithm 2 – ID3 Decision Tree (Mitchell, 1997) *based on* (Quinlan, 1986)

---

1. ID3 *(data set D, class attribute φ, defining attribute set X)*
2. Create a $Root$ node for the tree
3. **if** all records in D are of the same class φ_i:
4. return the single-node tree with class label φ_i
5. **if** X is empty**:**
6. return the single-node tree with class label of most common φ_i in D
7. **else:**
8. $A =$ attribute in X with highest information gain
9. the decision attribute for $Root = A$
10. **for** value v in A:
11. append new tree branch below $Root$, corresponding to the test $A = v$
12. $D_v =$ subset of D where $A = v$
13. **if** D_v is empty:
14. append new leaf node to branch with class label of most common φ_i in D
15. **else:**
16. append subtree ID3$(D_v, \varphi, X - \{A\})$ to branch
17. **return** $Root$

Anyone with a peripheral understanding of decision trees should be able to understand the general process of this algorithm. The only part that may require some explanation is found in line 8 – *information gain*. In addition to this, we must acknowledge the problems with using generalised values in building decision trees and also how to deal with numerical values.

Entropy & Information Gain

Entropy is a measure commonly used in information theory; it is a measure of the impurity of a collection of items. The items, in this case, are records in the data set. The entropy of a set of records *D* with an array of different classification values in class attribute φ is given by the formula:

$$Entropy(D) = \sum_{i \in \varphi} -P_i \log_2 P_i$$

Where P_i is the proportion of *D* with class *i* in class attribute φ. Note: $\log_2 0$ is defined as 0 for entropy calculation.

The entropy can then be used in the following calculation of information gain:

$$InformationGain(D, A) = Entropy(D) - \sum_{v \in A} \frac{|D_v|}{|D|} Entropy(D_v)$$

Where *A* is the selected attribute and D_v is the subset of records with value *v* for *A*.

Dealing with Generalised Values

Generalised values represent an obstacle to the implementation of the ID3 decision tree algorithm. A generalised value essentially represents a range of equally possible values. We need to consider how to deal with them in calculating entropy and also how to create a branch in the tree from them.

Regarding the calculation of entropy, P_i is the proportion of D belonging to class i. This can be written $\frac{|D_{\varphi=i}|}{|D|}$ where D could be the entire data set (or partition), which poses no issue where generalised values are concerned. However, in the information gain calculation, we need to also find the entropy of D_v. In an anonymised data set, for any given record, value v may be generalised, we need to consider how to deal with this. Shao & Beckford [2017, pp. 4] suggest simply considering the generalised value as its own unique value in attribute A. This would appear to be a viable solution, however, doing so would necessitate using a generalised value as a branch in the tree. The problem with this is values in classification must then be generalised in the same way, leading to potential complications further down the line. Another proposition by Shao & Beckford [2017, pp. 4], and the one used in this report, is to consider all values in a generalisation to be equally likely the true value. In a generalisation containing r values, we can consider each value in the generalisation to be worth $\frac{1}{r}$. Figure 6 shows an example of this. The table shows records for a generic attribute *Attrib* and corresponding class. Non-generalised values are simply worth a count of 1 in calculating P_i, but generalised values are worth $\frac{1}{r}$ in this case r = 2, so each generalised value is worth 0.5 in the calculation.

Figure 6

Attrib	Class
A	X
B	X
B	Y
[A,B]	Y
[A,B]	Y
[A,B]	X

$$Entropy(Attrib_A) = -\frac{1.5}{2.5}\log_2\frac{1.5}{2.5} - \frac{1}{2.5}\log_2\frac{1}{2.5}$$

$$Entropy(Attrib_B) = -\frac{1.5}{3.5}\log_2\frac{1.5}{3.5} - \frac{2}{3.5}\log_2\frac{2}{3.5}$$

Figure 6 - Generalisation Entropy Calculation Example

Using this method means we are not railroaded into using generalised values for branches in the decision tree, avoiding complications there. Also, because the generalised values are equally distributed, the entropy of a particular attribute will be determined by the distribution of non-generalised values only. Thus, attributes with fewer generalisations will be favoured as the next node in the decision tree, leading to a decision tree that is based to a greater degree on actual data.

As stated, we ideally want to avoid using generalised values as branches in the decision tree. The approach taken to deal with this in this report is essentially to *map-back* generalisations to one of the values contained in its value-set. We do this in two ways, both suggested in Shao & Beckford [2017, pp. 5]:

- Random Mapping: We select a value from the generalisation's value-set at random to be the attribute value for a given record. This value can then be used as a branch in the tree. The benefit of this is twofold: simplicity of implementation and prevention of biases.
- Statistical Mapping: If we consider the case whereby a frequency distribution for all original QID values is released with an anonymisation, we can then use the corresponding distribution to assist in mapping back a generalisation to a single value for the given attribute. We can do this by weighting the probability of selection of a particular value by its frequency in the original data set. In theory, this could result in records that more closely represent the original data. Although, this is not guaranteed and could lead to biases towards a particular value in the attribute domain that is most prevalent. It could also cause problems with privacy if the mapped-back values are too similar to the original data, undermining the whole point of anonymisation. However, if privacy loss is minimal, the trade-off for better classification accuracy may be worthwhile.

Both methods are tested in this report in order to provide a more comprehensive evaluation. In addition to this, a side-evaluation will be provided on how accurately values are mapped back to their original values. This should provide some insight into how problematic the mapping process could be in retaining the privacy gained through anonymisation.

Numerical Attributes

The standard ID3 algorithm is only designed to handle categorical data. However, it can be modified to handle numerical data. In real-life situations, a researcher will likely use numerical data in classification, for this reason, it was decided that the ID3 implementation in this report would be modified to deal with numerical data.

The basic idea behind the modification used in this report is expressed in [Mitchell, 1997, pp.72-73]. Essentially, instead of appending a branch to a tree corresponding to the question "*does the value for attribute A in the record equal v?",* we ask the question *"is the value for attribute A less than or equal to v?"* In order to do this, we need to determine what the value *v* is.

There are numerous ways of determining the value for *v*. A common approach, and the one used in this implementation, is to find the mid-points between adjacent numerical values, then test each to find the threshold that produces the greatest information gain. An example of how this works can be seen in Figure 7. Here, for brevity, we only show the calculation of information gain for one midpoint. First, we find the information gain from data where the value for *Age* is less than or equal to the midpoint. Then we find it for data where the value is greater than the midpoint. Of course, we are more interested in the information gain where there are more records involved. As can be seen, $InfoGain(Age_{\leq 18.5})$ has a high value, but only 1 record is involved in the calculation, which tells us less about the entire dataset. Therefore, we take the weighted average as representing the information gain for that midpoint. If we repeat this process for all midpoints, we can then use the midpoint with the highest information gain as the threshold for testing in the tree.

Figure 7

Age	Class
18	A
19	B
20	B
21	B
22	A

$$Midpoints = [18.5, 19.5, 20.5, 21.5]$$
$$Test - \ 18.5:$$
$$InfoGain(Age_{\leq 18.5}) = Entropy(Dataset\ D) - \frac{|D_{Age \leq 18.5}|}{|D|} Entropy(D_{Age \leq 18.5})$$
$$InfoGain(Age_{\leq 18.5}) = 0.971 - \frac{1}{5}(0) = 0.971$$
$$InfoGain(Age_{>18.5}) = Entropy(Dataset\ D) - \frac{|D_{Age > 18.5}|}{|D|} Entropy(D_{Age > 18.5})$$
$$InfoGain(Age_{>18.5}) = 0.971 - \frac{4}{5}(0.811) = 0.322$$
$$WeightedAvg = \frac{1}{5}(0.971) + \frac{4}{5}(0.322) = 0.452 = InfoGain(Age_{18.5})$$
$$Repeat\ for\ all\ midpoints.$$

Figure 7 - Numerical Information Gain Calculation Example

3.4 - ADULT Data Set

The ADULT data set [Dua & Graff, 2019] has become the standard data set for the evaluation of anonymisation algorithms. Indeed, it is present in almost all of the works cited in the *Related Works* section [2.4] of this report. Essentially, it is an extract from the census with 15 attributes: *age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, gender, capital-gain, capital-loss, hours-per-week, native-country, income*. In this report, *fnlwgt* and *education-num* were omitted from any experiments. *fnlwgt* is essentially the number of people the census believes the entry represents – this is not very useful for the experiments in this report. *education-num* is simply a numerical representation of the *education* attribute and is therefore unnecessary. The *income* attribute is the target classification attribute; therefore, we have 12 other attributes that can be used to classify a record in the decision tree. The data set consists of 48,843 records in total, however, some of these have missing values. For simplicity, these records were removed prior to use – resulting in 45,222 remaining records.

3.5 - Implementation Testing

It is important for guaranteeing the reliability of results that the implementations of the various algorithms for this report are tested. Details on the testing of the two main algorithms follow. All other smaller algorithms or scripts will be tested in a similar manner, for brevity, the details of each will not be given here.

Testing the *k*-Anonymity Tool

In this paper, the approach to ensuring validity will be primarily through testing on small-scale synthetic data. This essentially means taking a mock data set of limited records, using the tool to *k*-Anonymise it, then establishing if the output is correct by comparison to the expected output. A further method of ensuring validity can be done through simple checks, such as ensuring that each equivalence class is larger than *k*, and comparison of measurements for the running time of various anonymisations to ensure consistency with expectation. Performing these experiments should be sufficient in ensuring the validity of the tool.

Testing the ID3 Classifier

As previously stated, the utility of anonymised data in this report will be evaluated by measuring classification accuracy from an ID3 decision tree, trained and tested on anonymised data in different scenarios. Before this can begin, as with the *k*-Anonymity tool, the ID3 algorithm must be tested to ensure validity. Again, a simple method of checking the algorithm would be to test the algorithm on a small set of synthetic data, comparing the results to expected output, this can be done for both building of the tree and classification. A second method will involve a comparison of classification accuracy with an established implementation of the ID3 algorithm. For this, the popular *Weka* [University of Waikato, 2021] machine learning tool can be used. These two methods of ensuring validity should be sufficient.

3.6 - Experimental Methodology

Once validity has been established for both the *k*-Anonymity tool and the ID3 algorithm, the process of evaluating the utility of anonymised data can begin.

Each experiment will use the following process:

1. *k*-Anonymise ADULT data set with quasi-identifiers {QIDs} for given *k*-value
2. Use 6-fold cross-validation testing to measure classification accuracy:
    a. Take [training] & [testing] data sets and split each into 6 subsets of equal size such that subsets in one data set contain the same records as their counterpart subsets in the other data set
    b. Take 5 subsets of [training] and build a decision tree using ID3 from these subsets
    c. Take the remaining subset not used for training from [testing] and attempt to classify all records within
    d. Compare test classification with actual classification for each record
    e. Store results
    f. Repeat above until each subset has been the test set
3. Calculate the average of the 6 cross-validation results and store the average result

The value of 6 was chosen as the number of cross-validation folds simply because 6 is the largest reasonable number that divides the total record count of 45,222 exactly, resulting in equal-sized subsets of 7537 records.

In this report, the set of attributes used to classify a record will remain fixed as the 12 attributes [*age, workclass, education, marital-status, occupation, relationship, race, gender, capital-gain, capital-loss, hours-per-week, native-country]* for all experiments.

The above must be performed on data sets anonymised using the *Strict* and *Relaxed* versions of the Mondrian *k*-Anonymity algorithm.

The above must also be performed with ID3 decision trees trained using the *random* and *statistical* mapping back of generalised values.

{QIDs} will vary in size from 8 to 12. The QIDs used can be seen in Figure 8, with the newly added QID shown in **bold**:

Figure 8

QID Count	{QIDs}
8	[*age, workclass, education, marital-status, occupation, race, gender, native-country]*
9	[*age, workclass, education, marital-status, occupation,* ***relationship****, race, gender, native-country*]
10	[*age, workclass, education, marital-status, occupation, relationship, race, gender,* ***capital-gain****, native-country*]
11	[*age, workclass, education, marital-status, occupation, relationship, race, gender, capital-gain,* ***capital-loss****, native-country*]
12	[*age, workclass, education, marital-status, occupation, relationship, race, gender, capital-gain, capital-loss,* ***hours-per-week****, native-country*]

Figure 8 - Quasi-Identifiers used for Experiments

For each {QIDs}, experiments will be performed on all values of *k* in {2, 5, 10, 25, 50, 100}. These values of *k* were chosen as these were the values used for testing in the [LeFevre et al., 2006, pp. 9] paper which originally described Mondrian *k*-Anonymity, they are a reasonable mix of realistic (commonly used) values for *k* and high values used for testing purposes.

Figure 9 shows all combinations of training/testing data sets used in experiments for given {QIDs}, algorithm type and generalisation mapping type:

Figure 9

		TESTING						
	Data Set	Non-Anonymised	*k*=2	*k*=5	*k*=10	*k*=25	*k*=50	*k*=100
TRAINING	Non-Anonymised	Non-Anon. Only	Non-Anon/Anon	Non-Anon/Anon	Non-Anon/Anon	Non-Anon/Anon	Non-Anon/Anon	Non-Anon/Anon
	k=2	Anon/Non-Anon	Anon. Only *k*=2					
	k=5	Anon/Non-Anon		Anon. Only *k*=5				
	k=10	Anon/Non-Anon			Anon. Only *k*=10			
	k=25	Anon/Non-Anon				Anon. Only *k*=25		
	k=50	Anon/Non-Anon					Anon. Only *k*=50	
	k=100	Anon/Non-Anon						Anon. Only *k*=100

Figure 9 - Experiments to be performed

Grey cells show experiments that will not be undertaken in this report.

Note that *Non-Anon. Only* will be performed just once as it is deterministic and involves no anonymisation to require testing of different algorithms, {QIDs} or generalisation mapping types. Moreover, the *Non-Anon/Anon* experiments will not need to be repeated for different generalisation mapping types, as the non-anonymised data set obviously does not contain generalisations needing to be mapped back.

From the above information, we can calculate that the total number of experiments that will be performed in this report will be 301 excluding repetitions. Due to the nature of cross-validation testing, repetitions are essentially intrinsic. However, owing to the numerous random elements in the ID3 algorithm, repeating the experiments will ensure more reliable results. Considering time constraints, a single repetition of all experiments with non-deterministic results will be performed. It follows that there are a grand total of 601 experiments to be performed.

4 - Deliverables on Approach

4.1 - Mondrian *k*-Anonymity Tool

Implementation

My implementation of the Mondrian *k*-Anonymity tool was written in Python and can be found in the file [anonymise.py]. Python was chosen as the programming language for this implementation as it has some native data structures and established libraries that would be useful here, it is also a language that I have modest experience with. Here I will explain some key elements of the implementation, including how to use the tool. This implementation deals with data in *.csv* form; chosen simply because it is the data format most familiar to me.

The *Strict* implementation generally follows the algorithm described in *Algorithm 1*, with some changes for the *Relaxed* algorithm. However, it was necessary to abstract the code described in *Algorithm 1* into multiple functions.

The main function in the code is the *anonymise* function. This function takes as input the list *args* used to initialise and perform an anonymisation. The following *args* are passed by the user in the command line at runtime:

- Input Filename: The non-anonymised .csv file path
- QID List: The list of attribute indices to be used as QIDs, separated by commas
- *k*-value: The chosen *k*-value for this anonymisation
- Output Filename: The destination path for output .csv file
- Headers: A Boolean (0/1) flag for removal of attribute headers in the input file
- Strict: A Boolean (0/1) flag for selection of the *Strict* algorithm. 0 = False = *Relaxed*

For example, the following command line input would create a *Strict* 10-anonymisation file named *outputfile.csv* of the header-less file *inputfile.csv* using QIDs 0, 1 and 2:

anonymise.py inputfile.csv 0,1,2 10 outputfile.csv 0 1

The *anonymise* function executes one of two different functions depending on whether the *Strict* or *Relaxed* algorithm was selected: *mondrianStrict* and *mondrianRelaxed* respectively. The difference between these two functions is entirely to do with the partitioning of the data set. As mentioned, the *Strict* algorithm prevents intersecting values between the two partitions made from a cut on a particular attribute. The median value at which a cut is performed is found using a frequency set for the domain of a given attribute, it is, therefore, possible that the median value could be the last value in the ordered list of attribute values. In this case, all records will be placed in the *lhs* partition, as all record values for the given attribute will be ≤ median value, resulting in no partitioning at all. Therefore, the *mondrianStrict* function features a check whereby the median value *medVal* is compared to the final value in the ordered list of values in the attribute domain. If there is a match, then the *mondrianStrict* function is recursively called with the previously used attribute removed from the QID list, essentially retrying with the next best attribute. This is not an issue in the *Relaxed* algorithm as even if the median value is the last in the ordered list, some records with this value will still be placed in the *rhs* partition – so partitioning is still performed.

Both of these functions will append their optimised partitions to the global list *partitions* – these are essentially the *equivalence classes*. Once the entire data set has been partitioned accordingly, the function will return to the *anonymise* function, which will then cycle through each partition in the list *partitions*, generalising each one with the *generalise* function and appending the generalised

partition into a new list named *anon_partitions*. Each record in all partitions of *anon_partitions* is placed into a single list named *k_anonymisation* which is then written to the output *.csv* file. The anonymisation is then ready for use.

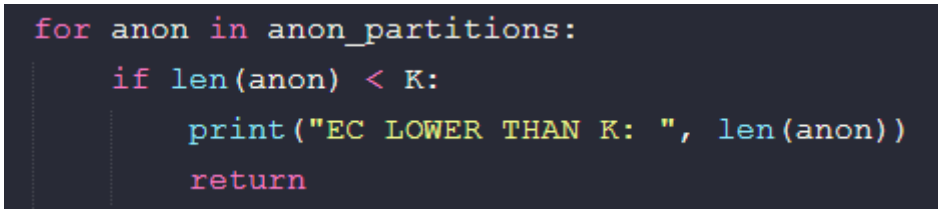
It will not be necessary here to expand on anything else in this implementation, as the majority of it simply follows *Algorithm 1* with the details already explained. However, a breakdown of each function can be found in the comments on [anonymise.py] if further detail is needed.

Testing

Firstly, this tool was tested by creating some synthetic data and comparing the actual output of the tool to the expected output. The results of the tests can be found in the supporting materials [MondrianTests] file. The expected output in each test was determined by hand, as this is the only reliable method of doing so. Hence, the maximum size of the data set in terms of records, and the amount of QIDs used, must be limited due primarily to time constraints and the difficulty of performing multidimensional partitioning by hand. However, the tests assessed a range of different conditions for both the *Strict* and *Relaxed* algorithm, in theory, the output should remain correct when the data set is scaled up. All test cases were passed.

The second method of testing mentioned in the *Approach* chapter [3.5] was to measure the size of equivalence classes to ensure none were smaller than *k*. Naturally, this should not happen if the algorithm is implemented correctly. However, to provide insurance, a check is added into the *anonymise* function of [*anonymise.py*]. The code for this check can be seen in Figure 10. When the generalised partitions are merged to create the *k*-Anonymisation, the size of each partition is checked to ensure it is greater than or equal to *k*. If a partition is found to be *smaller* than *k*, an error message will be printed to the terminal, and the process will stop. This error never occurred during the creation of any anonymisations using the tool. Therefore, it can be safely concluded that every equivalence class in each anonymisation is greater than or equal to *k* and that this test has been passed.

Figure 10



```
for anon in anon_partitions:
    if len(anon) < K:
        print("EC LOWER THAN K: ", len(anon))
        return
```

Figure 10 - Code Snippet of EC Test

The final set of tests for the Mondrian *k*-Anonymity tool involved taking measurements of the running time of anonymisations. The running time is simply checked against expectations; you would expect it to decrease as the *k*-value increases for both the *Strict* and *Relaxed* algorithms due to fewer partitions being necessary, with the *Strict* algorithm taking less time in each case due to a simpler partitioning method. It would be difficult to compare these timing results to any literature due to differences in hardware.

Figure 11 shows the timing results from the anonymisations with *k*-values in {2, 5, 10, 25, 50, 100} using 8 QIDs, for both the *Strict* and *Relaxed* algorithms. We can see, as expected, that running time decreases with increasing *k*-values, and the *Strict* algorithm executed faster than the *Relaxed* algorithm. Therefore, we can consider this test to have been passed.

Figure 11

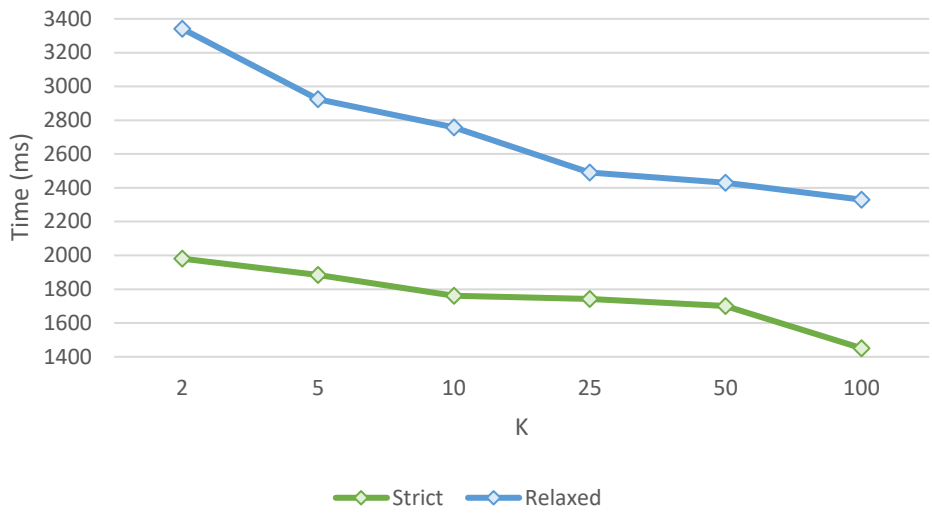


Figure 11 - Timing Results of Mondrian Algorithms

4.2 - ID3 Decision Tree

Implementation

It was considered initially that the ID3 decision tree classifier could be a publicly available implementation. However, due to various specifics outlined in the *Approach* chapter [3.3] i.e., properly dealing with numerical values and generalised values, a bespoke implementation was decided on as the best option.

For this implementation, again the Python programming language was used, simply for compatibility purposes with the *k*-Anonymity tool. Again, here I will explain some key specifics of the implementation, with further detail in the code comments.

The entire implementation includes two aspects: *training* and *testing*. *Training* involves the building of the decision tree from the input data set. *Testing* (or *classification*) involves classifying records from another data set using a given decision tree. These two aspects were split into two separate files in this implementation.

The *training* part of the implementation essentially follows *Algorithm 2*, with a few changes. It can be found in the [id3.py] file. The *init* function is the initial function for creating a decision tree, here the following parameters must be provided:

- Input Data Set Path: Path to the data set used to train the ID3 decision tree
- Target: The target attribute used for classification
- Attributes: A list of *describing* attributes used for training of the tree
- Attribute Names: A list of names corresponding to the *Attributes* parameter – used for naming of nodes in the tree
- Training Indices: A list of record index pairs – each pair references the start and end records of a training set, non-inclusive (e.g. [0, 100] will build a tree using records 0-99)
- Type: Flag determining type of generalisation mapping to use (0 = *Random*, 1 = *Statistical*)
- Output File: The *.txt* file to output the decision tree to

- Output Mapping File: The *.csv* file to output the mapped data set to (generalised values –> specific values). Used to determine generalisation mapping accuracy later

After some preamble, the *id3* function will be called. This is the main ID3 algorithm in this implementation and is designed as a recursive function. Consequently, when this function is called, it could be to begin the creation of the main ID3 tree, or simply a subtree that will be appended to the main tree. Here, the third-party library *AnyTree* [c0fec0de 2020] is used to create the decision tree. *AnyTree*, as the name suggests, is a library that allows for the creation of any kind of tree data structure. It is flexible and allows for nodes to have any amount of user-defined attributes, it also features a JSON exporter/importer tool which allows for decision trees to be saved for later use. This would prove a very useful tool as the training of decision trees in this implementation is non-deterministic and can take a long time; this tool allows for consistent reuse of the same tree for testing purposes, saving time, and helping to ensure validity.

As with *Algorithm 2*, the root node must first be created. The *bestAttrib* function is called which finds the best attribute to use as the next node. The best attribute is the attribute with the greatest information gain, as explained in the *Approach* chapter [3.3]. Note that *bestAttrib* iterates over all attributes and checks first if the current attribute is numerical, if so: information gain is calculated using the *calcInfoGainNumeric* function, otherwise, the *calcInfoGainCategoric* function is executed. Both of these functions work as described in the *Approach* chapter [3.3].

Once the root node is created, the most common classification from the data set is found to be used as the default, whereby if the entire data set is of the same classification the root node is returned, with the label of the most common classification, as the finished tree (or subtree). Every value in the domain of the best attribute is then iterated over, and a branch off of the root is created signifying the yes/no question *"Is the value for the given attribute in this record equal to this value?"* With the records split into their respective subsets based on the answer to this question. Note: *AnyTree* does not use a separate data type for branches, thus branches are simply nodes in this implementation.

Each subset (if not empty, whereby the default classification is used), is then used as the input data set in a recursive call of *id3* which creates a subtree appended at the end of the branch. The *id3* function finally returns the entire tree as a pointer to the root and is then exported as a JSON in *.txt* format into the *Output File*.

The *testing* part of the implementation can then import the JSON formatted decision tree to classify all records in a given data set. The file *[id3classify.py]* contains the code for this part. The *classify* function within is the main function. This function is fairly straightforward and, as mentioned, details can be found in the comments of the code. It essentially iterates over every record in the data set and traverses the input decision tree, classifying the record using the standard approach. The one difference is, if the value being checked in a given record is generalised, a match is found if the attribute value in the tree being compared is *in* the generalisation value-set of the record, as opposed to non-generalised values where a match is found if the two values are *equal*. Note: the classification of every record is not stored, as this is unnecessary for this implementation. Instead, it is simply compared to the true classification value as and when it is classified, and the *correct* or *incorrect* counters are incremented. The accuracy of the classification can then be determined by simply finding the percentage of correct values out of the total. Results are written to the output *.txt* file.

Testing

Similar to the *k*-Anonymity tool, the main method of testing the ID3 implementation is through the use of small-scale synthetic data, whereby an expected output is calculated by hand and compared to the output from the algorithm. It is difficult to perform many of these tests as calculating information gain and the associated entropies can be an arduous task by hand, so a limited number of tests were performed which cover a range of potential scenarios. The results can be found in the supporting material [ID3Tests]. The first half of test cases tested the *training* part of the implementation, the second half tested the *classification* part of the implementation using selected trees trained from the first half tests. All test cases were passed.

As discussed in the *Approach* chapter [3.5], the next test involved a comparison of classification accuracies for a consistent data set between the ID3 implementation discussed in this report, and an established implementation [Frank et al., 2012] using the *Weka* tool. For this, the Non-Anonymised ADULT data set was used, as the Weka implementation of ID3 cannot deal with generalised values. 6-fold cross-validation testing was used as standard, with the following attributes selected to classify the *Income* attribute: [*workclass, education, marital-status, occupation, relationship, race, gender, native-country*]. Note: no numerical attributes were used for this test, as the Weka implementation cannot deal with numerical values.

The full results can be found in the supporting materials. With the results from the Weka implementation in [wekaresults.txt] and the results from the implementation discussed in this report in the [id3testresults.txt] file. Essentially, the Weka results showed a ~78.8% accuracy in classification, with my implementation resulting in a ~81.3% accuracy. This difference can be explained by the fact that the Weka implementation does not use the default class value of an attribute when attempting to classify an unseen value. This is illustrated in the [wekaresults.txt] file by the 1852 unclassified instances. In my implementation, these instances would have been classified as some default class. If we were to assume that all of these unclassified instances would have been classified correctly by using a default as with my implementation, we can calculate a classification accuracy of ~82.9%. Obviously, it is unlikely *all* of these instances would be classified correctly by using a default, hence the 1.6% difference between my implementation's classification accuracy, and the 82.9% calculated Weka accuracy. It is impossible to know which records would be classified correctly and incorrectly, so we cannot determine if the two implementations match exactly. However, the difference is very small, thus, it would be reasonable to say that the implementation in this report sufficiently matches the established implementation.

4.3 - Metric Tests

Separate scripts were used for calculating the three metrics used in this report. The files [discernabilityMetric.py], [iLossMetric.py] and [classificationMetric.py] calculate their respectively named metrics for a given dataset. Discussing the implementations in detail is not necessary here, as they are basic implementations of simple formulae. However, each of these scripts was tested to ensure validity. The results can be found in [MetricTests], whereby each script was tested on two different small-scale synthetic data sets, comparing the output to an expected output calculated by hand. All tests were passed successfully.

4.4 - Other

All other scripts are essentially auxiliary to the above. Most are concerned with simply running one of the above scripts for a range of parameters to gather experimental results. Details on each one can be found in the code comments.

5 - Results

5.1 - Classification Accuracy

As established, the primary method of measuring the utility of anonymisations in this report is to measure the classification accuracy in ID3 decision tree classification. This chapter will begin by detailing the results from the classification experiments outlined in the *Approach* chapter [3.6].

Non-Anonymised Only Results

To get a baseline measurement to use as a comparison with other scenarios, it is necessary to find the classification accuracy of the ID3 classifier for non-anonymised data using a tree trained on data that has also not been anonymised. It is expected that these classification accuracy results will be the highest measured, simply because original data is used, meaning that maximal information is available to the classifier.

Figure 12 shows the results from the 6-fold cross-validation testing done on non-anonymised data. The average result is the value to be used as the baseline classification accuracy – 81.617%. For the sake of brevity, this will be the only set of results where each individual cross-validation test will be shown – it is shown here simply to highlight the methodology. All subsequent cross-validation testing results will only show the average of the 6 tests

Figure 12

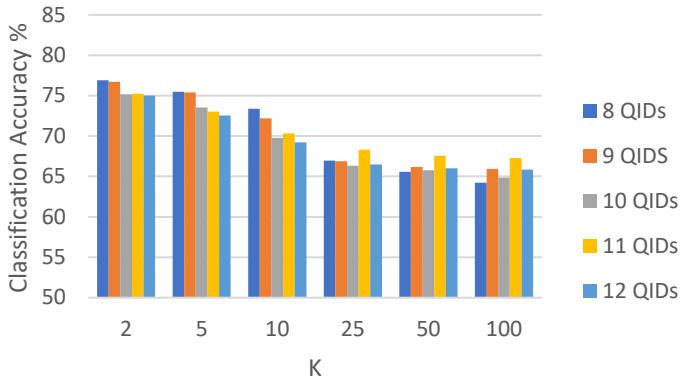
Cross-Validation Test #	Classification Accuracy %
1	81.597
2	81.345
3	81.730
4	81.292
5	82.685
6	81.053
AVG.	81.617

Figure 12 - Classification Results for Non-Anonymised Data

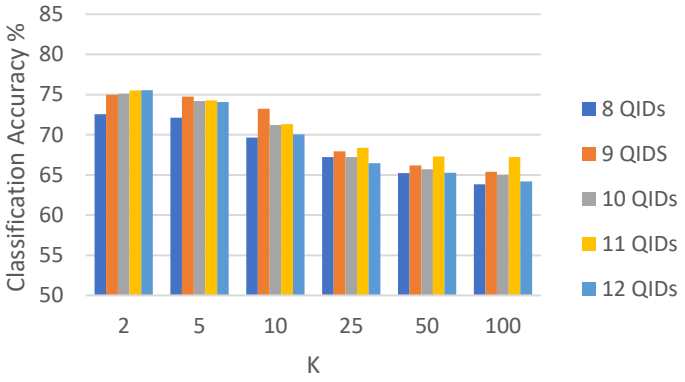
Anonymised Only Results

The results in this part of the section are those collected from measuring the classification accuracy of the ID3 classifier for anonymised data using a tree trained on anonymised data. Figure 13 shows the results from each experiment split into four figures. Each figure shows results from a different combination of Mondrian algorithm type and generalisation mapping type. Each column in a given figure represents a classification accuracy measurement for an anonymisation with the corresponding *k*-value, with different colours representing different QID counts used, as per the legend.

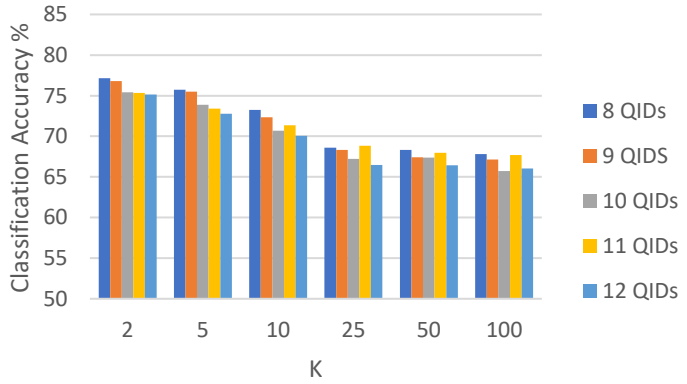
Figure 13



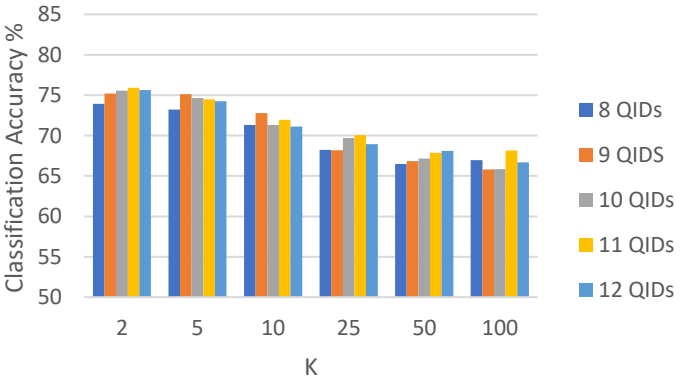
13a. – Relaxed Algorithm with Random Generalisation Mapping



13b. – Strict Algorithm with Random Generalisation Mapping



13c. – Relaxed Algorithm with Statistical Generalisation Mapping



13d. – Strict Algorithm with Statistical Generalisation Mapping

Figure 13 - Classification Results for Anonymised Data

The main thing that can be seen from these results is that there is a general downward trend in classification accuracy for increasing *k*-values in all cases. This is to be expected as increasing the *k*-value in a *k*-Anonymisation will result in equivalence classes of a larger average size, which would make discernability between records in an equivalence class more difficult. It would also increase the average size of generalisations in equivalence classes, resulting in a tree built and tested with reduced information. In this sense, these results would agree with the literature. Shao & Beckford [2017, pp. 6] show classification accuracy results from a similar study previously mentioned, there is a clear downward trend in these results for increasing *k*-values, although the exact results differ due to differences in methodology.

In addition to the above, in these results, there appears to be a moderate amount of variability between consecutive QID counts for different values of *k*. There is no clear pattern or trend. This would appear counter-intuitive at first thought. It would be expected that increasing the number of QIDs in an anonymisation would result in reduced classification accuracy, as you are increasing the number of attributes in the data that could be generalised – resulting in greater information loss. This is in fact what is shown in Shao & Beckford [2017, pp. 6], where the *k*-value was fixed and an increasing number of QIDs showed a downward trend in classification accuracy. However, we must consider the *nature* of the QID being added in each iteration. It can be seen in the results from Shao & Beckford that the change in classification accuracy between consecutive numbers of QIDs is not constant, which would suggest that different attributes carry different amounts of information for classification purposes – which is intuitive as you would assume an attribute such as *Occupation* to have a greater weight in classifying the income of a person than, for example, *Marital Status*.

In LeFevre et al. (2006 pp. 9), we can see a similar study, the value of *k* was again fixed and the number of QIDs was increased from 2 through 8, with the discernability penalty (DM) measured. Of course, this is not the same as measuring the classification accuracy. Indeed, I will later show how DM may not be a reliable *estimation* for classification accuracy, but there is a slight correlation between the two in that a lower discernability penalty suggests a higher classification accuracy and vice versa. The results show a clear downward trend in discernability penalty, which would suggest an increasing classification accuracy. LeFevre et al. state concerning these results *"this decrease is due to the sparsity of the original data, which contains fewer duplicate tuples as the number of attributes increases"* [LeFevre et al., 2006, pp. 9], which would seem intuitive. It is important to note that these results were collected from tests on synthetic data, not the ADULT data set. However, they could suggest that an increase in classification accuracy is *possible* with an increase in the number of QIDs.

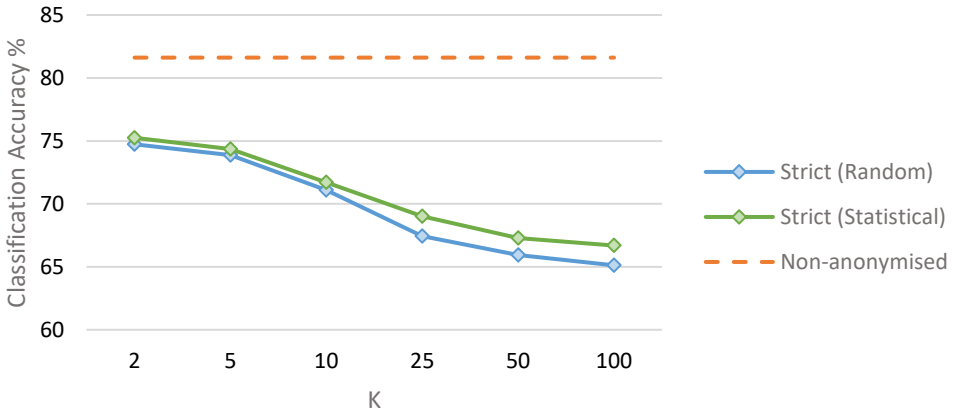
Moreover, we must consider the random elements present in the ID3 algorithm when building the tree and classifying using the tree. These elements have been established previously in the report. To reiterate, they are the mapping-back of generalised values in the building of the tree, and the selection of a single value in a generalisation to represent a record when attempting to classify it. As established, increasing the number of QIDs can reduce the size of equivalence classes, shown by the decrease of DM measured in LeFevre et al. [2006, pp. 9]. However, it could also increase the number and size of generalisations within those equivalence classes, as new QIDs may contain values needing generalisation. An increase in the number of generalisations and the size thereof decreases the chance of the true value in a generalisation being selected, this is the case in the mapping-back process when building the tree and when classifying a record. This could conceivably create variability in classification accuracy in different run-throughs of the experiment.

A combination of all of the above could provide logical reasoning for the variability we are seeing between consecutive QIDs in the results in Figure 13. It should be noted that with sufficient repeat experiments, the random elements could be eliminated from the data, and a clearer pattern could emerge. Due to time constraints, the experiments in their entirety were only repeated once for this report. It would be naïve to make further suggestions on how the number of QIDs in an anonymisation affects the classification accuracy at this stage. Further experiments would need to be performed to make any solid observations.

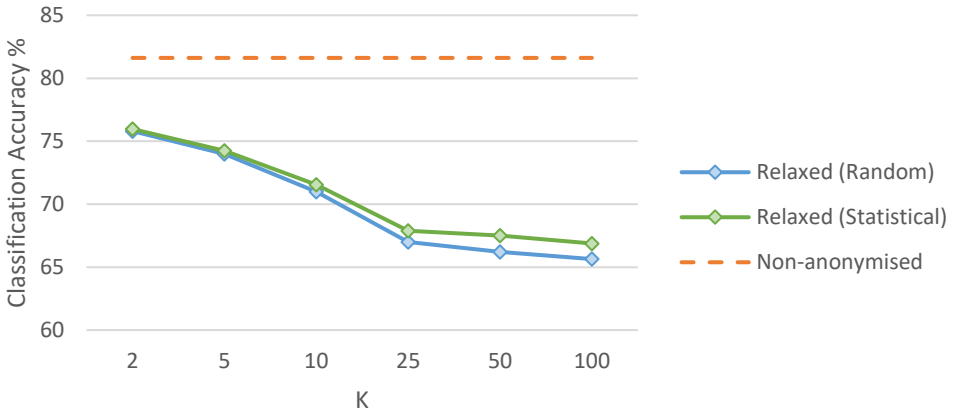
Figure 14 shows comparisons between the two types of generalisation mapping used. 14a shows the comparison for the *Strict* Mondrian algorithm, 14b shows the comparison for the *Relaxed* Mondrian algorithm, and 14c shows the average of each mapping type over both algorithms. Each data point in the figures is an average of the classification accuracy over all QIDs for the corresponding *k*-value, including the repeat experiment results. The *Non-Anonymised Only* result is also shown on each figure for comparison.

These results show more clearly the overall decline in classification accuracy for increasing values of *k*. Along with the results from various literature previously mentioned [Shao & Beckford, 2017, pp. 6] [LeFevre et al., 2006, pp. 9], we can state with some certainty that there is a clear negative correlation between the *k*-value used for anonymisation and the classification accuracy measured – something that is perfectly intuitive.

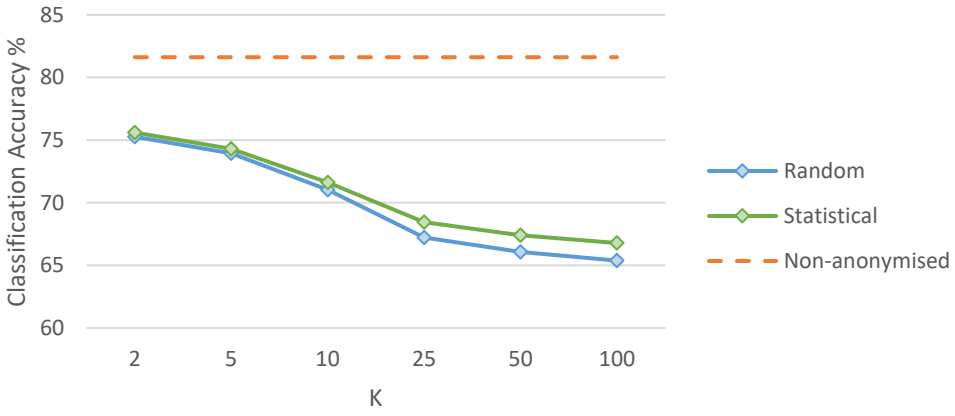
Figure 14



14a. – Strict Algorithm – Generalisation Mapping Type Comparison



14b. – Relaxed Algorithm – Generalisation Mapping Type



14c. –Average Generalisation Mapping Type Comparison (Avg. of 14a & 14b)

Figure 14 - Mapping Type Comparison for Anonymised Data

It appears from the results that there is very little change in classification accuracy between the two types of generalisation mapping. There is a slight improvement when using the Statistical approach, although the magnitude thereof is minimal, with an improvement of ~1.4% in the best case. This increase is intuitive, as you would expect that attempting to maintain the frequency distribution of the domain of an attribute would result in more accurate mapping-back selections than simply selecting at random. Figure 15 shows the accuracy of the values mapped back using both approaches, with each data point corresponding to the data points shown in Figure 14c. This was measured by simply comparing the mapped-back value selected from a generalisation to the true value in the original data. As we can see, the Statistical approach produced much more accurate mappings of generalisations compared to the Random approach. It is a point of contention as to whether the accuracy shown is sufficiently low enough to ensure the privacy of an individual in a data set is sustained, this is something that would require further investigation and therefore will be avoided in this report. However, the gain in classification accuracy from the Statistical approach is so minimal that even if privacy could be said to be maintained using this approach, the extra work required from the data publisher to release such statistics would be unwarranted.

Figure 15

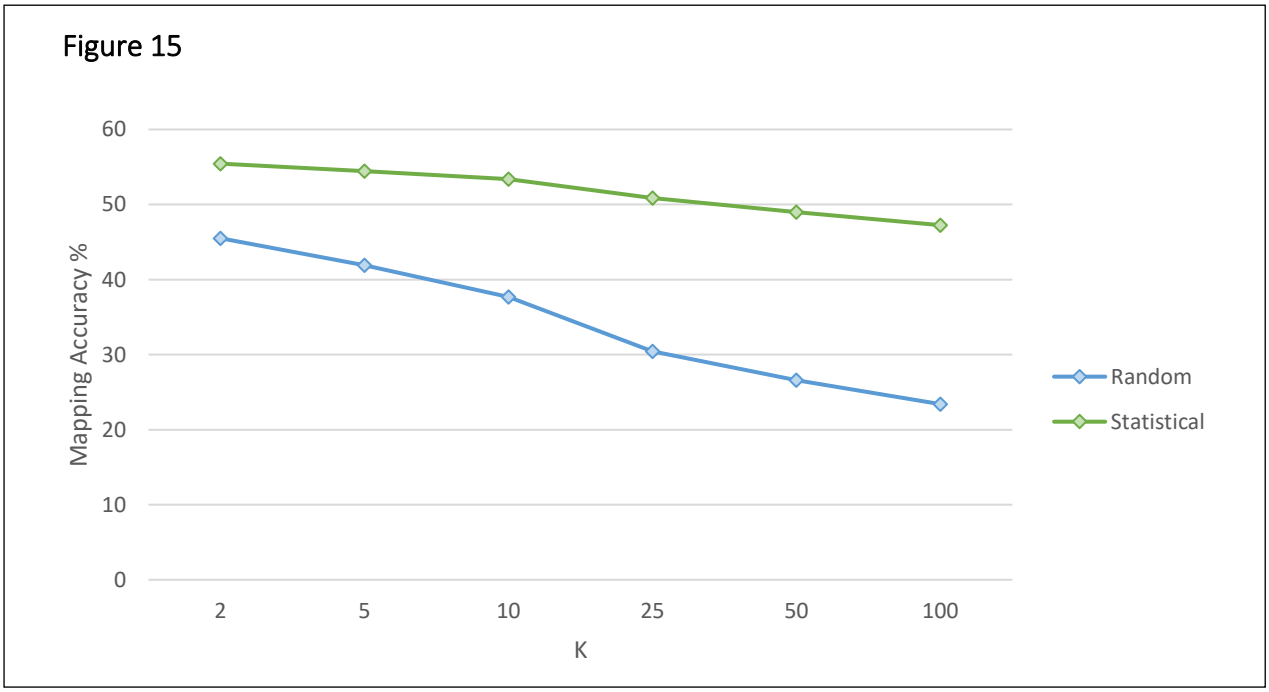


Figure 15 - Comparison of Average Mapping Accuracy for Generalisation Mapping Type

Regarding the Mondrian *k*-Anonymity algorithm type, Figure 16 shows a comparison of classification accuracy between the two types of the algorithm: *Strict* and *Relaxed*. These results are compiled from average classification accuracy over all QID counts for a given *k*-value, disregarding the generalisation mapping method. From the results, we can see that there is very little difference between the *Strict* and *Relaxed* algorithms. It is necessary to mention now that the *Relaxed* algorithm tends to result in an anonymisation with smaller equivalence classes on average than the *Strict* algorithm, this will be shown in a later section [5.2] discussing the DM results. However, the *Relaxed* algorithm also has a slightly higher ILoss measure, suggesting a greater number and/or size of generalisations within the equivalence classes. This could be the reason for the minimal difference

between the classification accuracies measured, as the generalisation count and size are countering the equivalence class size.

Figure 16

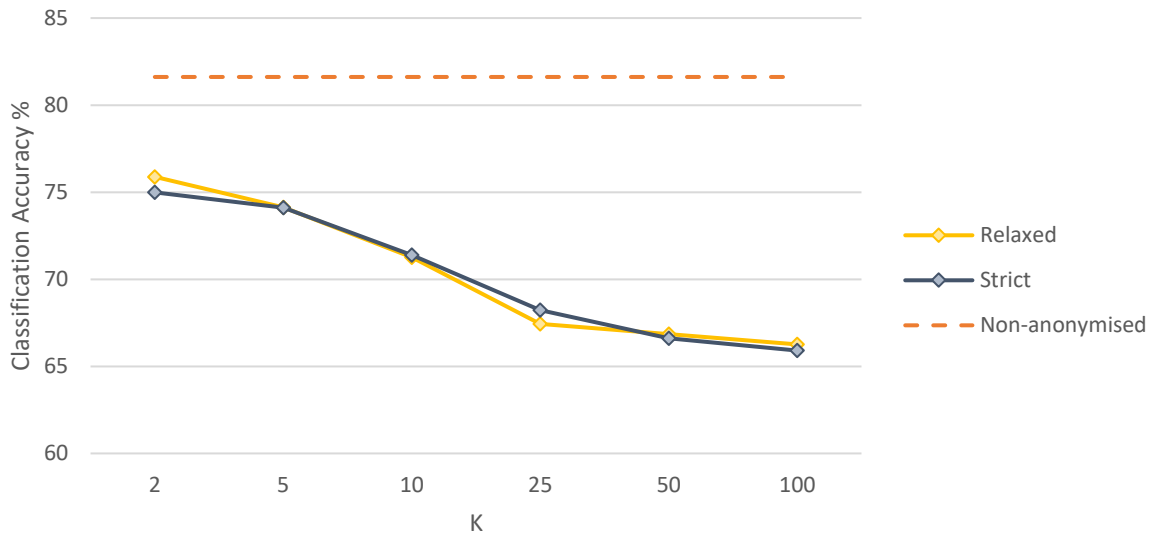
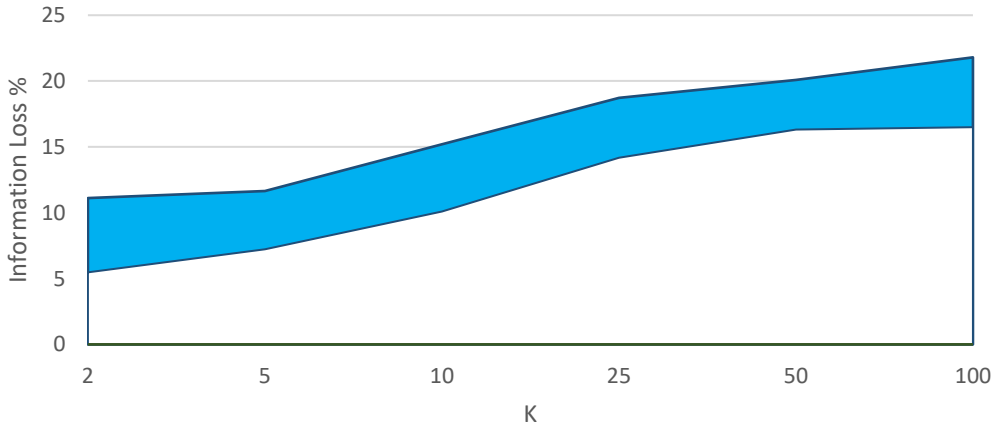
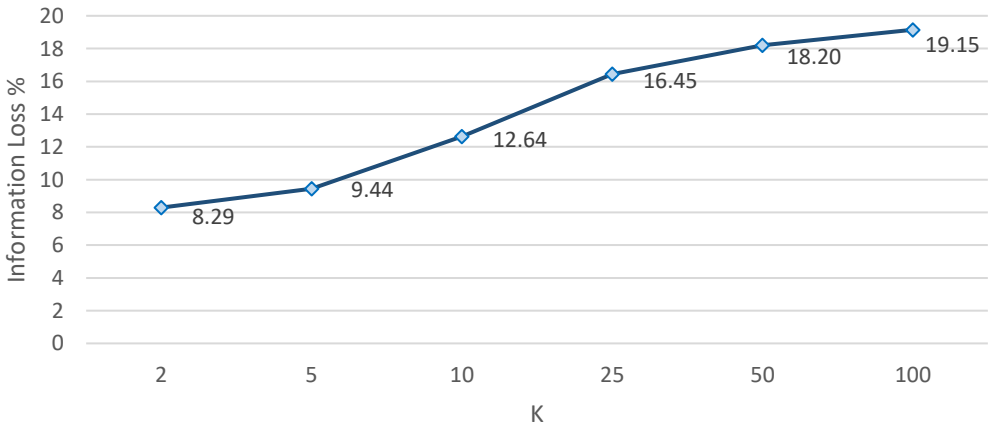


Figure 16 – Comparison of Average Classification Accuracy for k-Anonymity Algorithm Type

Figure 17



17a. – Measured Information Loss Range for Any Algorithm/Mapping type



17b. – Average Information Loss for Any Algorithm/Mapping type

Figure 17 - Information Loss for Anonymised Data

Figure 17 shows the measured information *loss* for any algorithm type and generalisation mapping type, over all QID counts for given *k*-values. Here, *information loss* is a measure of the loss of classification accuracy using the measurement of 81.617% found from the *Non-Anonymised Only* experiment as a baseline, expressed as a percentage drop. 17a is a visualisation of the range of measured *information loss*, that is, based on the results from experiments in this report, we can expect the actual *information loss* for a given *k*-value to fall within the coloured area. 17b is the average of the measured *information loss* for the corresponding *k*-value. These figures can essentially be considered an estimate of the utility of anonymised data in comparison to non-anonymised data for decision tree classification, regardless of algorithm, mapping type or QID count.

Based on Figure 17, we can expect a loss of between ~5.5% and ~21.8% in classification accuracy when classifying anonymised records using a tree trained on anonymised data. The range in information loss here could be considered *minimal* (~5%) to *significant* (~20%). However, we must consider the loss for *k*-values that are more commonly used. El Emam & Dankar [2008, pp. 631] state that *"It is uncommon for data custodians to use values of k above 5, and quite rare that values of k greater than 15 are used in practice."* For values in this range, the information loss is more reasonable. For the most recommended *k*-value, that is *k*=5, we can expect a loss of ~9.4% in classification accuracy, with actual measured classification accuracy at around 74%. Considering the benefits provided by anonymisation regarding individual privacy, this value would probably be acceptable to researchers.

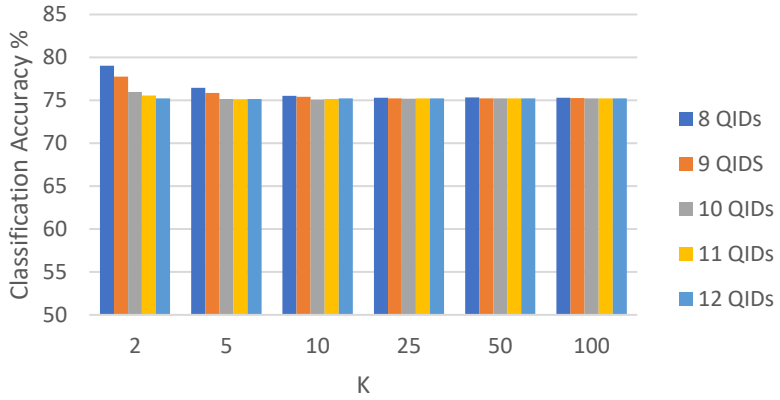
In comparison to the *Non-Anonymised Only* result, it would appear that information loss is inevitable when attempting to classify anonymised data using a classifier trained on anonymised data. However, the extent of information loss can be mitigated to a reasonable level by selecting smaller *k*-values when generating *k*-Anonymisations.

Non-Anonymised / Anonymised Results

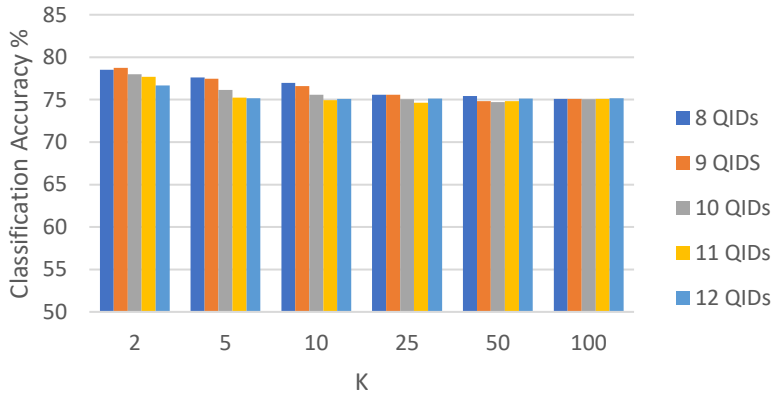
This part of the section will discuss results found from the classification of Anonymised data using a decision tree trained on Non-Anonymised data. Figure 18 shows results for classification accuracy of both algorithm types. Note that generalisation mapping does not occur in this scenario, as the tree is built from Non-Anonymised data. Figure 18a shows results from the classification on data anonymised using the *Relaxed* algorithm, 18b shows results from the *Strict* algorithm, 18c is an average of results for each algorithm over the full range of QID counts.

The starkest observation from these results is that classification accuracy varies very little across the board, regardless of *k*-value, number of QIDs, and algorithm type. This would suggest that anonymisation has a much-reduced effect on the utility of data when it comes to actual classification using a decision tree – the majority of the information lost from anonymisation shown in the previous part of this section extends from the *building* of the decision tree. Figure 18c shows in fact that the *Strict* algorithm performs slightly better in this scenario than the *Relaxed* algorithm for the more commonly used *k*-values. The reason for this is likely similar to what was previously mentioned: whilst the *Relaxed* algorithm results in smaller equivalence classes, the number and size of generalisations within these equivalence classes are generally greater on average. When an anonymised record is classified, the random element is selecting the value from a generalisation value-set that will represent this record for classification. The algorithm has a better chance of selecting the true value from a generalisation with fewer values in its value-set, in theory leading to better classification accuracy.

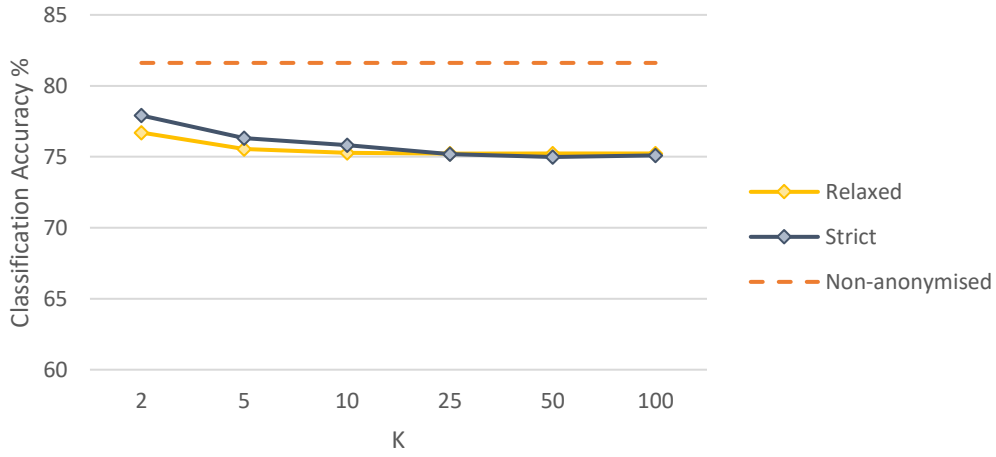
Figure 18



18a. – Relaxed Algorithm Classification Accuracy



18b. – Strict Algorithm Classification Accuracy

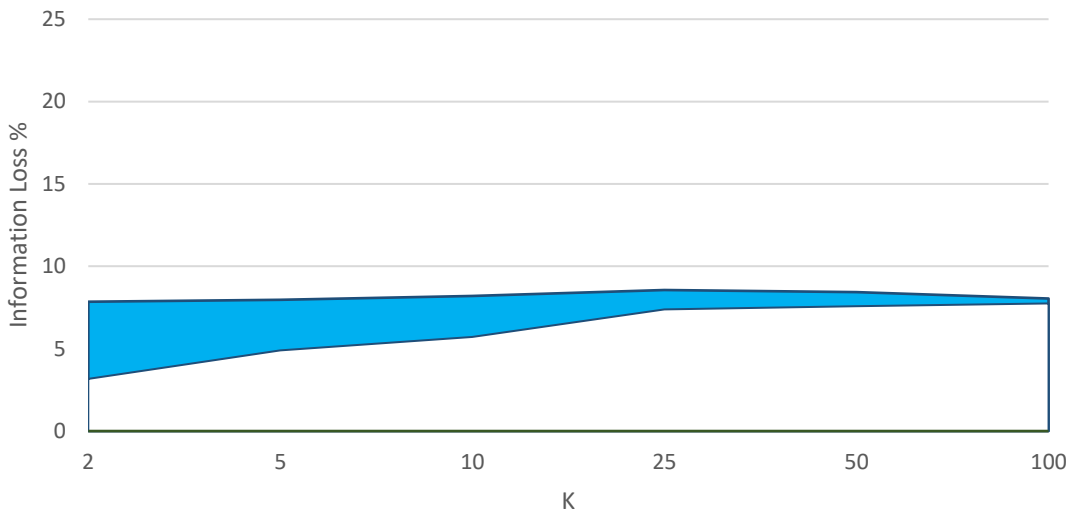


18c. – Algorithm type average classification accuracy comparison

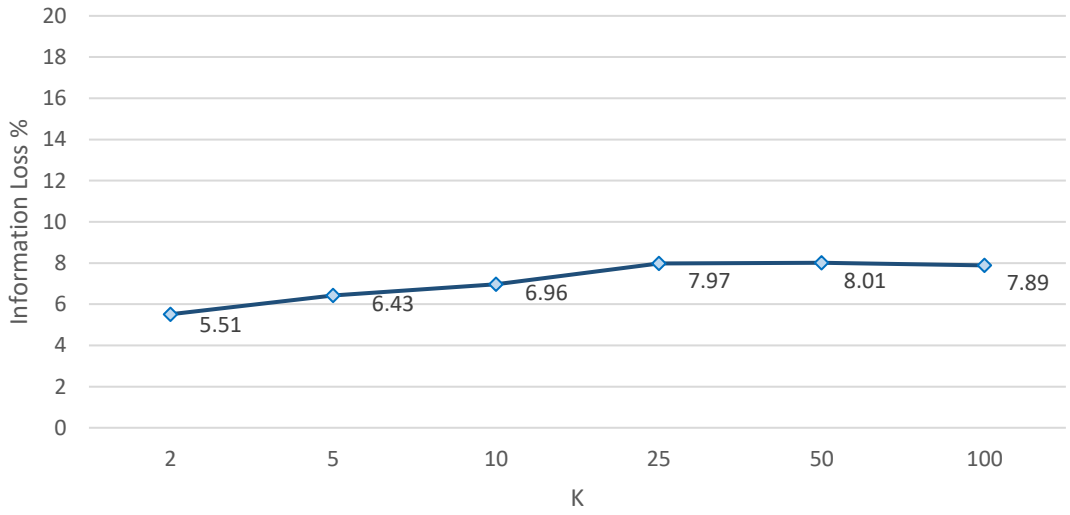
Figure 18 - Classification Accuracy Results for Non-Anon/Anon. Data

Similar to the *Anonymised Only* part of this section, Figure 19 shows the measured information loss for a given *k*-value. We can see that the drop in classification accuracy in comparison to the *Non-Anonymised Only* baseline is minimal for all *k*-values, much reduced from the *Anonymised Only* results, which is completely as you would expect because anonymised data is used half as much as in the *Anonymised Only* section. It should be mentioned that the levelling off and slight improvement in information retention seen for the higher *k*-values is most likely due to the fact that there is a maximum limit on generalisation size for a given attribute. Once this threshold has been reached, in this case, it would seem to be around *k*=25, the chance of randomly selecting the true value from a generalisation value-set for a given record when classifying remains stable for all subsequent *k*-values. With sufficient repetitions, we would likely see the gradient of the line move closer to 0 for *k* ≥ 25.

Figure 19



19a. – Measured Information Loss Range for Any Algorithm type



19b. – Average Information Loss for Any Algorithm type

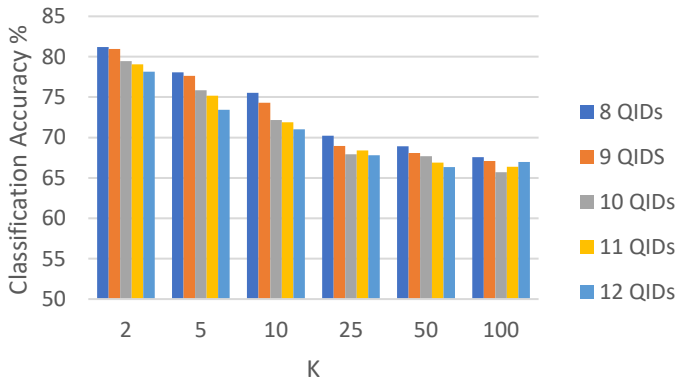
Figure 19 - Information Loss for Non-Anon/Anon. Data

Again, information loss in this scenario appears to be inevitable. However, the amount of information lost from anonymisation, regardless of parameters used, is minimal and is much reduced from the previous scenario.

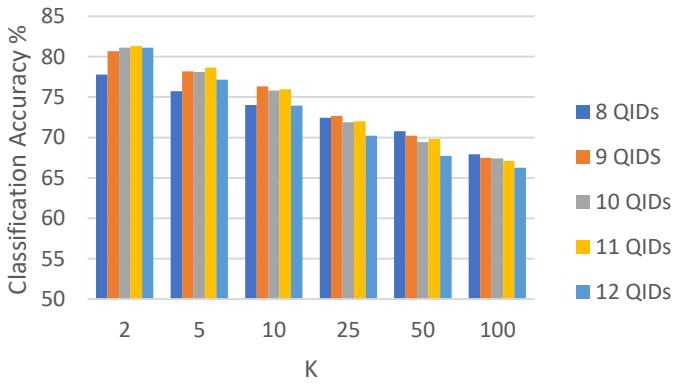
Anonymised / Non-Anonymised Results

This part of the section will examine the results from the classification of Non-Anonymised data using a decision tree trained on Anonymised data. Once more, Figure 20 shows the results for both algorithm and generalisation mapping types over all QID sets and *k*-values.

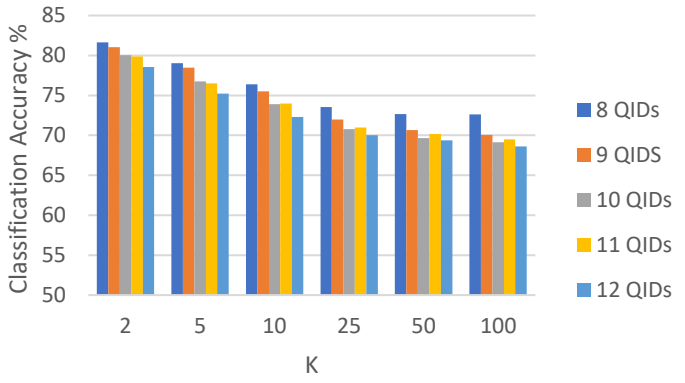
Figure 20



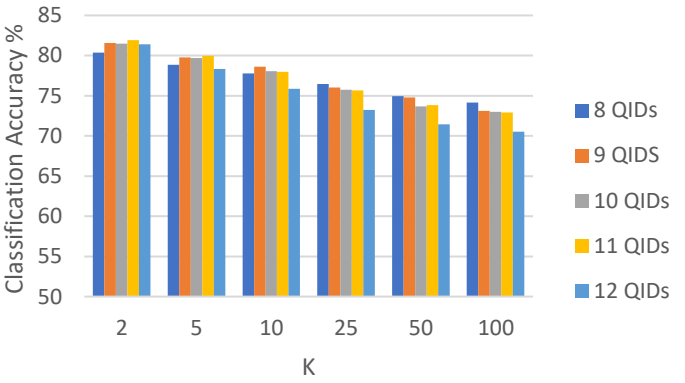
20a. – Relaxed Algorithm with Random Generalisation Mapping



20b. – Strict Algorithm with Random Generalisation Mapping



20c. – Relaxed Algorithm with Statistical Generalisation Mapping



20d. – Strict Algorithm with Statistical Generalisation Mapping

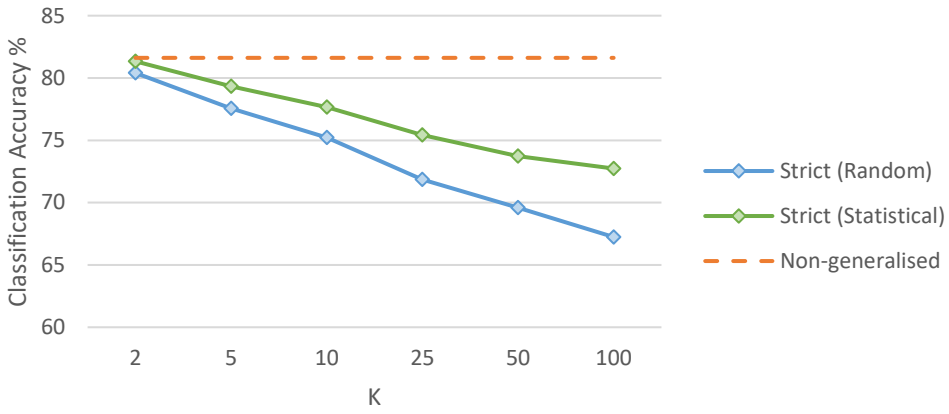
Figure 20 - Classification Results for Anon/Non-Anon. Data

Again, we can see a clear decline in classification accuracy for increasing values of *k*, further substantiating claims from the previous scenario that anonymisation has more of an effect on classification when building a decision tree than when classifying using it. We can also see more of a pattern emerging in successive QID counts, whereby the classification accuracy decreases as the number of QIDs increases. Anomalies are still present, but this could potentially show that the random elements in the ID3 algorithm may be to blame for the variance seen in the results for the *Anonymised Only* scenario. Again, this is just a suggestion, and more tests would need to be done to confirm this.

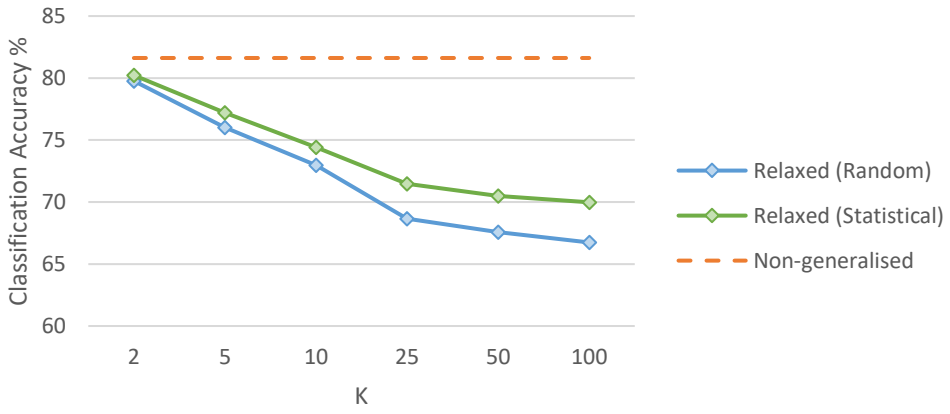
Figure 21 shows a comparison between the two types of generalisation mapping in the same manner as was shown in the *Anonymised Only* scenario. Here, the statistical approach appears to show a much more noticeable improvement over the random approach than before. The reasons for

this are not clear, again it could be simply because the random element involved in the classification part of the ID3 algorithm is eliminated in this scenario, whereas it was present in the experiments performed in the *Anonymised Only* scenario, by chance pulling the two sets of results closer together. The results do still align with the expectations in that you would expect the statistical approach to perform better than the random approach. Moreover, the difference is still relatively small, with a maximum difference of ~4.3%, and this being for the *k*=100 case, which would seldom be seen in real situations.

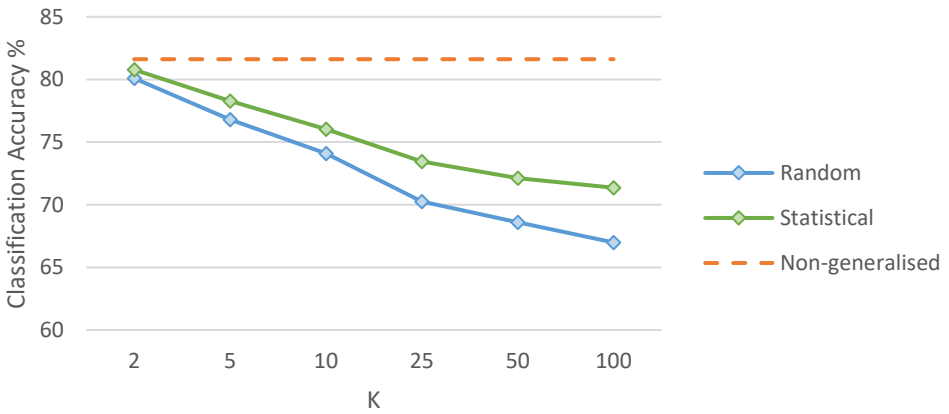
Figure 21



21a. – Strict Algorithm – Generalisation Mapping Type Comparison



21b. – Relaxed Algorithm – Generalisation Mapping Type



21c. –Average Generalisation Mapping Type Comparison (Avg. of 21a & 21b)

Figure 21 - Generalisation Mapping Comparison for Anon/Non-Anon. Data

We can also see from Figure 22 that the *Strict* algorithm appears to perform better than the *Relaxed* algorithm in this scenario. This may suggest, as implied before, that the number and size of generalisations within an equivalence class has a greater impact on the classification accuracy than the size of the equivalence class itself within an anonymisation.

Figure 22

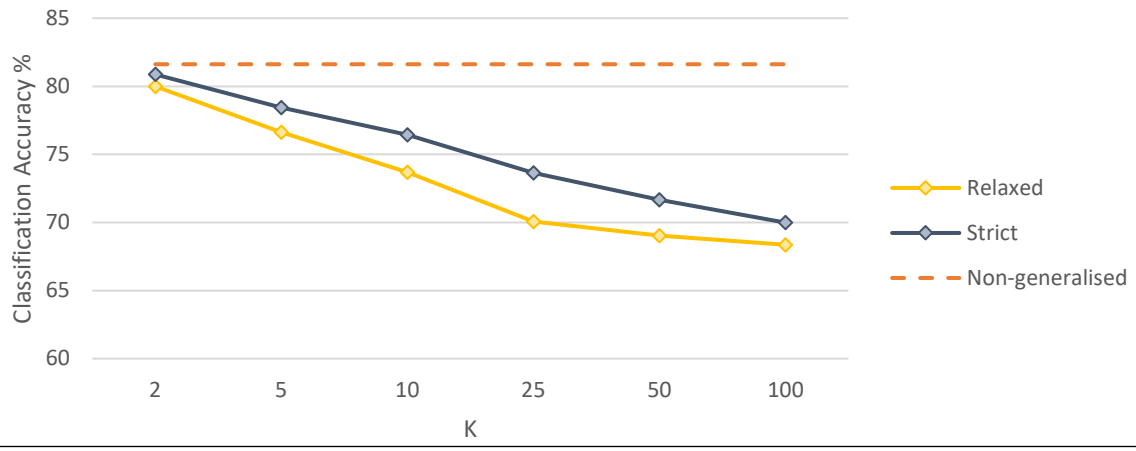
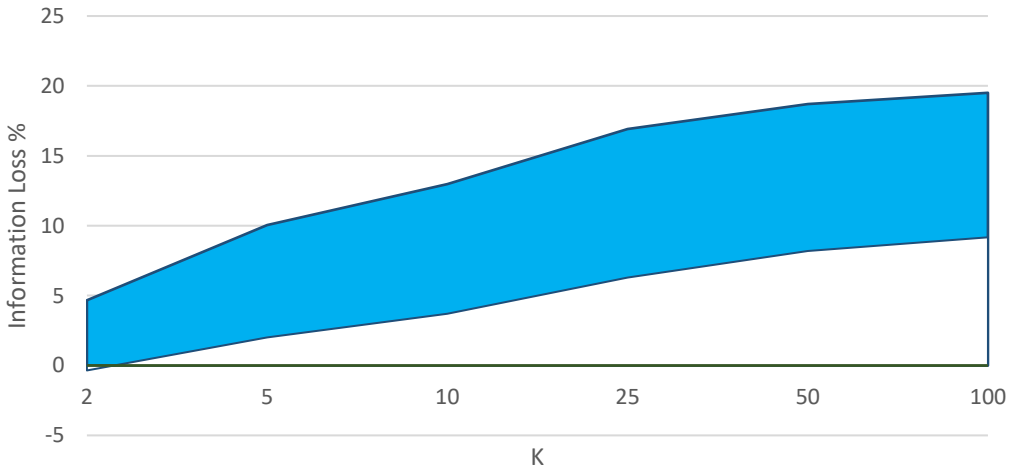
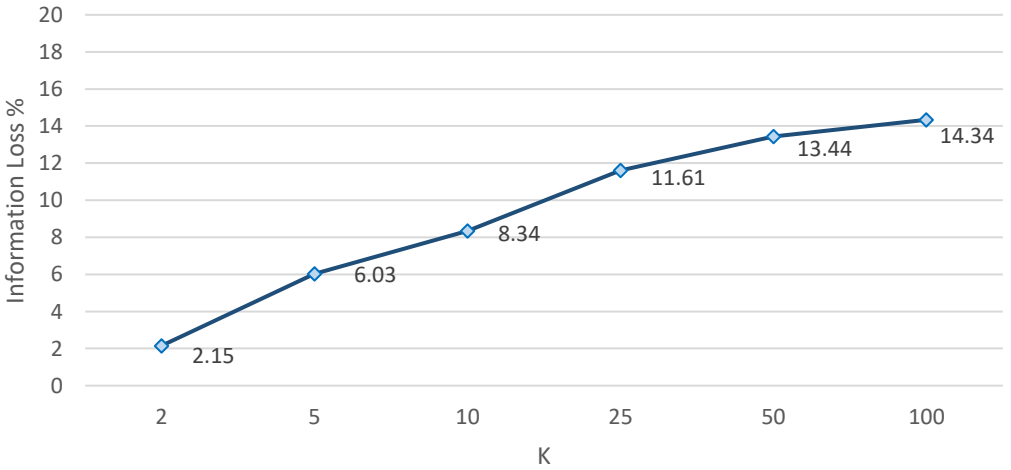


Figure 22 – Comparison of Average Classification Accuracy for k-Anonymity Algorithm Type for Anon/Non-Anon. Data

Figure 23



23a. – Measured Information Loss Range for Any Algorithm/Mapping type for Anon. / Non-Anon. Scenario



23b. – Average Information Loss for Any Algorithm/Mapping type for Anon. / Non-Anon. Scenario

Figure 23 - Information Loss for Anon/Non-Anon. Data

From Figure 23, we can see again the information loss compared to the baseline from the *Non-Anonymised Only* scenario. It would appear that this scenario is a slight improvement in terms of classification accuracy on the *Anonymised Only* scenario for all values of *k*, which is intuitive as anonymisation and the associated random element has been eliminated from the classification part of the algorithm.

It is most interesting to note that in Figure 23a the coloured area of the graph slightly crosses into the negative information loss area. This is of course due to the fact that in two cases in the experiments for this scenario, the classification accuracy actually *improved* over the baseline of 81.617%. This may seem completely counter-intuitive with the baseline being derived from the case whereby there was maximum information for training the classifier. However, this can be explained if you consider that training the decision tree classifier on *Non-Anonymised* data may actually be producing a slightly *overfitted* tree, where outliers in the data set are pulling the predicted class in the "wrong" direction. If we consider the situation shown in Figure 24. Both tables show the same record from a *Non-Anonymised* data set (24a) and an *Anonymised* data set (24b) The record would appear to be an outlier in the data, as you would expect someone of that age, education, and occupation to be earning >50k. Indeed, most of the records in the ADULT data set similar to this do have income >50k. Therefore, when calculating the classification accuracy for the baseline in the *Non-Anonymised Only* scenario, this record would pull the classification of similar records in the "wrong" direction.

However, when the same record has attribute values that have been generalised, through random chance during mapping back, a more *expected* value for those attributes could be selected. Essentially eliminating the outlying record. Figure 24b shows how this could be possible, where the ID3 algorithm could select *High-School* and *Craft-Repair* to represent the *Education* and *Occupation* attributes in this instance, the record would now seem to align better with similar records in the data.

This raises the question about how much we can trust classification models. We must remember that they are only *models*, extrapolating estimates from the data provided to them. This also shows the importance of having a large amount of reliable data to build classifiers from. It could be the case in reality that the record shown in Figure 24 is not an outlier whatsoever and is in fact part of the majority when the whole population is considered, even if it is an outlier in the data collected. '*Estimates'* are just that, and should not be relied upon as if they were completely reliable data.

Figure 24

24a.

Age	Education	Occupation	Income (Class)
53	Doctorate	Exec-Managerial	≤ 50k

24b.

Age	Education	Occupation	Income (Class)
53	[Doctorate, High School]	[Exec-Managerial, Craft-Repair]	≤ 50k

Figure 24 - Example of Improvement to Expected Classification

It is important to reiterate that the information *gain* seen in this scenario only occurred in two cases: *Relaxed* algorithm with Statistical generalisation mapping for *k*=2 with 8 QIDs, and *Strict* algorithm

with Statistical generalisation mapping for *k*=2 with 11 QIDs. It is also important to note that the increase was minimal, with the highest of the two being just a 0.35% increase on the baseline. Therefore, this should not be looked at as straying far from the fundamental expectations of the experiments, generally, information for classification is lost by anonymisation in almost all cases.

5.2 - Metric Comparisons

This section will outline the results from measurements of the three well-established metrics traditionally used in the literature to measure the effectiveness and utility of anonymisations. The three metrics measured were *Discernability Metric*, *ILoss*, and *Classification Metric*. Each metric was explained in detail in the *Background* chapter [2.3] of this report.

Discernability Metric

Figure 25 shows the results from measurement of the *Discernability Metric* (DM) on each anonymisation created using the *k*-Anonymity tool. 25a shows the average discernability over all QID counts for *Strict* and *Relaxed* anonymisations of each *k*-value. Figures 25b and 25c show the *Relaxed* and *Strict* results, respectively.

We can see from the results that the discernability penalty is far lower in all cases with the *Relaxed* algorithm than the *Strict* algorithm. As mentioned in the previous section [5.1], this shows that the *Relaxed* algorithm results in smaller average equivalence classes. DM is purely a measure of the size of equivalence classes in an anonymisation, it does not consider individual records.

Figure 25

Figure 25a – Average Discernability over all QID counts for each anonymisation

Anonymisation	Avg. Discernability
Relaxed *k*=2	129440.4
Relaxed *k*=5	252117.6
Relaxed *k*=10	499579.2
Relaxed *k*=25	1997238
Relaxed *k*=50	3994310
Relaxed *k*=100	7988454
Strict *k*=2	1226025
Strict *k*=5	1589356
Strict *k*=10	2330011
Strict *k*=25	6190686
Strict *k*=50	11765210
Strict *k*=100	23331286

Figure 25b – Relaxed Discernability

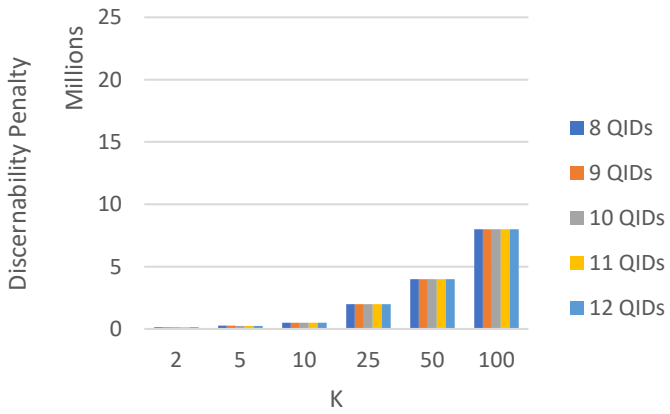


Figure 25c – Strict Discernability

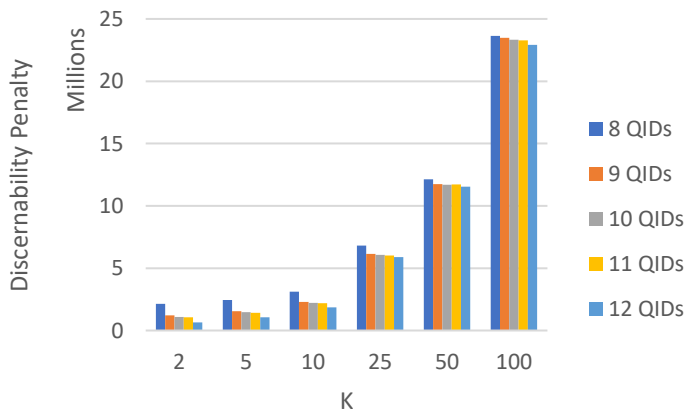
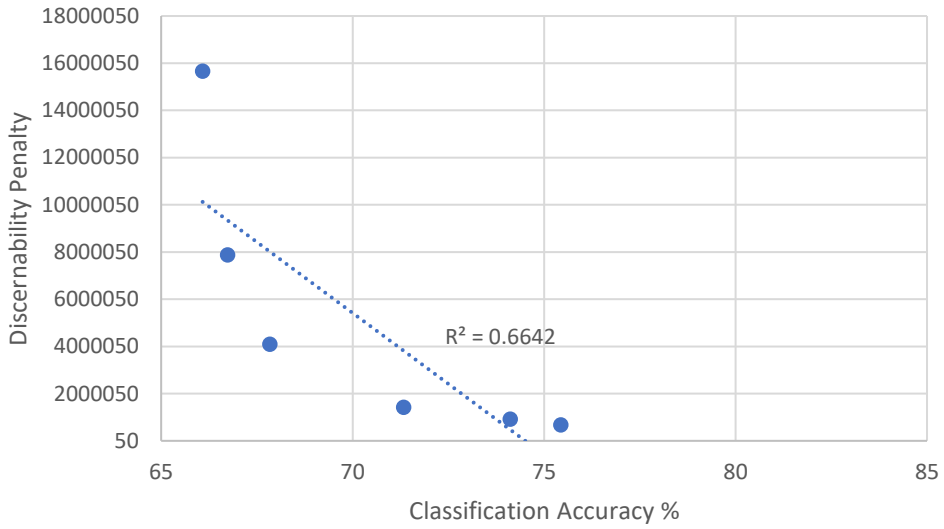
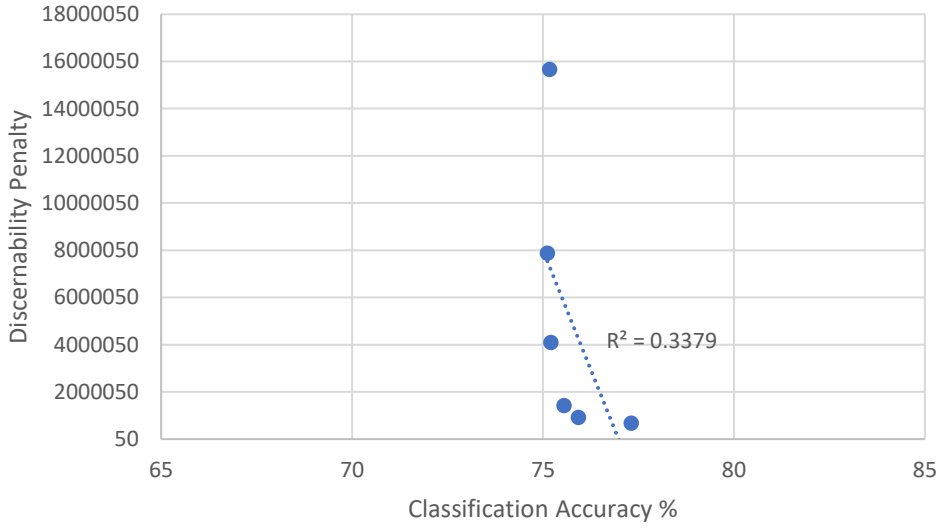


Figure 25 - Discernability Metric Results

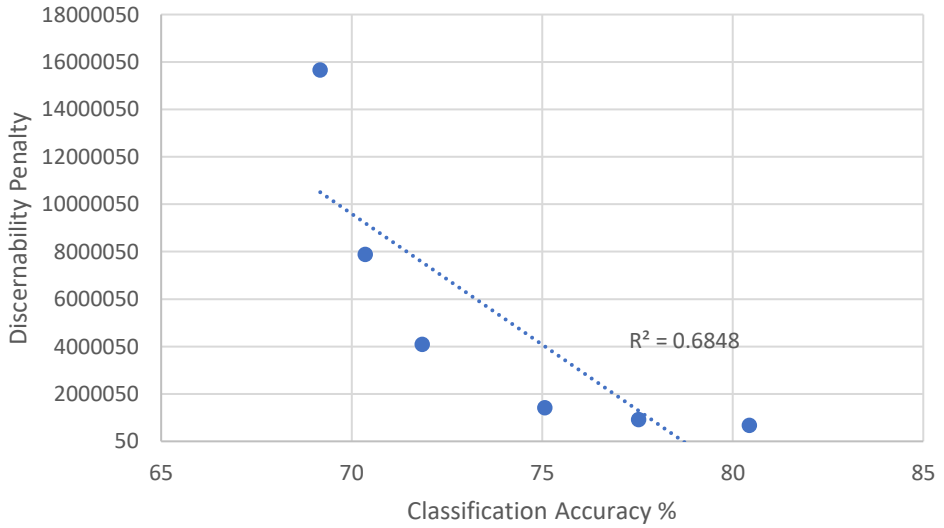
Figure 26



26a. – Correlation between Discernability Penalty and Classification Accuracy for Anon. Only Scenario



26b. – Correlation between Discernability Penalty and Classification Accuracy for Non-Anon/Anon Scenario



26c. – Correlation between Discernability Penalty and Classification Accuracy for Anon/Non-Anon Scenario

Figure 26 - Discernability Metric / Classification Accuracy Correlation

Figure 26 shows scatter graphs for each of the three experimental scenarios where anonymisation was involved. The graphs show a comparison between the classification accuracy and the DM penalty where the R^2 value is the absolute correlation measure between the two, and each point on the graph is representative of an anonymisation *k*-value. We can see from these results that the correlation between DM and classification accuracy is generally weak in all cases. As mentioned in the previous section [5.1], the correlation is clearly negative, indicating that an increase in DM suggests a decrease in classification accuracy, and vice versa. However, the correlation is weak because the actual value of classification accuracy cannot be predicted with any accuracy by examining the DM penalty. This gives substance to my previously mentioned suggestion that DM is not a reliable *estimate* for classification accuracy. However, while DM may not be a reliable *estimate*, it can still be considered a useful measure for comparing the classification utility of anonymisations from multiple different algorithms, simply by measuring if there is an increase or decrease in DM between them. It is also often used to better understand the privacy created from anonymisation. Nevertheless, I suggest to data publishers who know their data will be used for classification to consider other metrics when measuring the utility of their anonymisations for the purposes of selecting anonymisation parameters.

ILoss

Figure 27 shows the ILoss measurements from all anonymisations. Again, 27a shows the average over all QID counts, 27b shows the measurements for the *Relaxed* algorithm and 27c shows the measurements for the *Strict* algorithm.

Figure 27

Anonymisation	Avg. ILoss
Relaxed *k*=2	32168.8
Relaxed *k*=5	56009.04
Relaxed *k*=10	82511.95
Relaxed *k*=25	142352.3
Relaxed *k*=50	172780
Relaxed *k*=100	200520.9
Strict *k*=2	36593.44
Strict *k*=5	53136.39
Strict *k*=10	70367.98
Strict *k*=25	99306.13
Strict *k*=50	126395.4
Strict *k*=100	157944.9

Figure 27a – Average ILoss over all QID counts for each anonymisation

Figure 27b – Relaxed ILoss

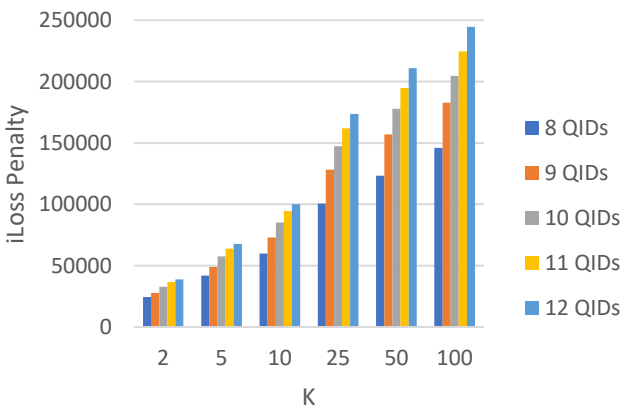


Figure 27c – Strict ILoss

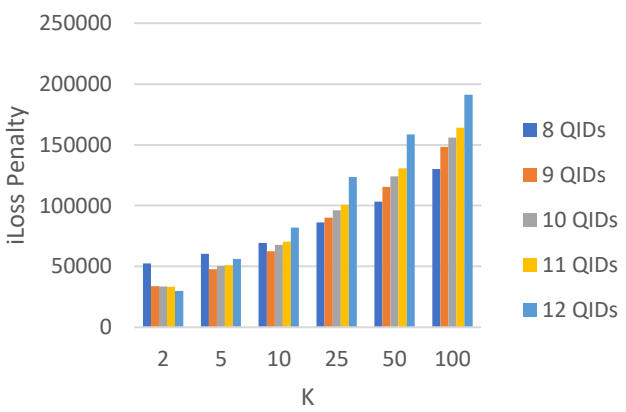
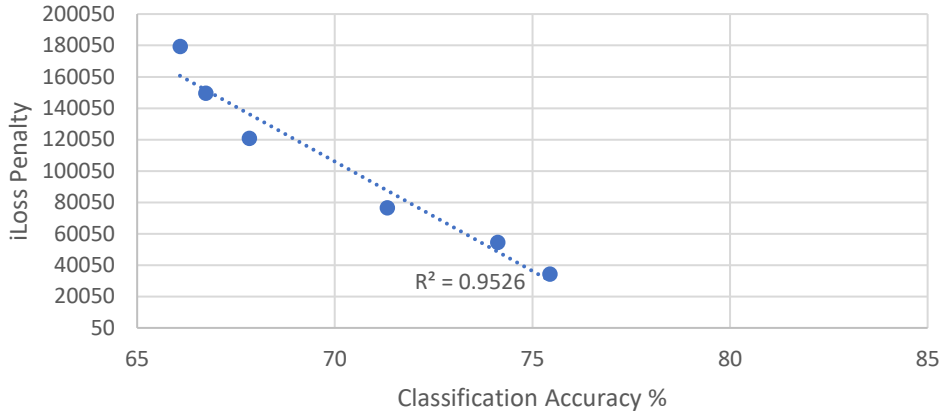


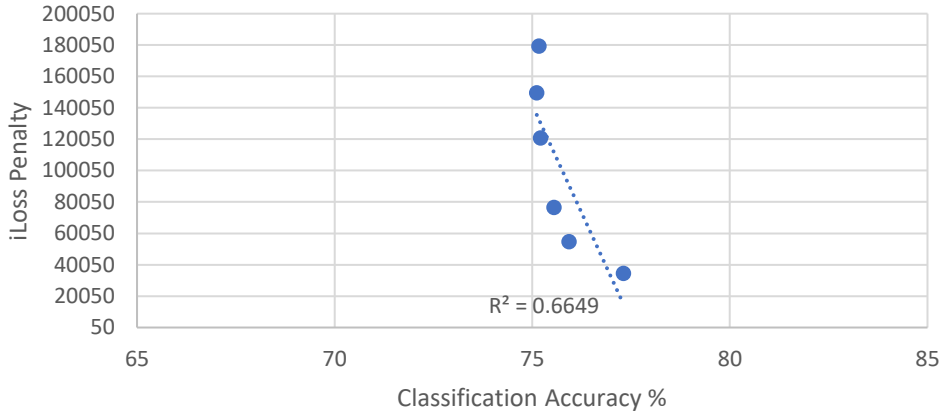
Figure 27 - ILoss Results

It can be seen from these results that the measured ILoss in almost all cases is higher in the *Relaxed* algorithm. ILoss considers the size of generalisations within its calculation, as such these results confirm that generalisations are generally larger with anonymisation from the *Relaxed* algorithm.

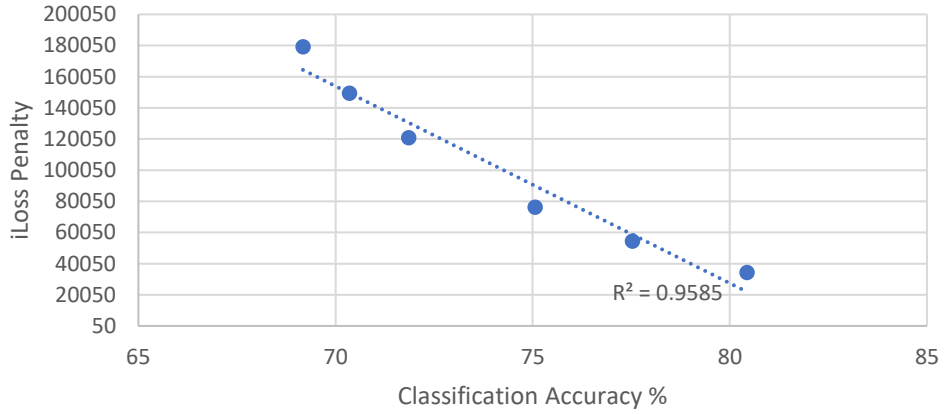
Figure 28



28a. – Correlation between ILoss and Classification Accuracy for Anon. Only Scenario



28b. – Correlation between ILoss and Classification Accuracy for Non-Anon/Anon Scenario



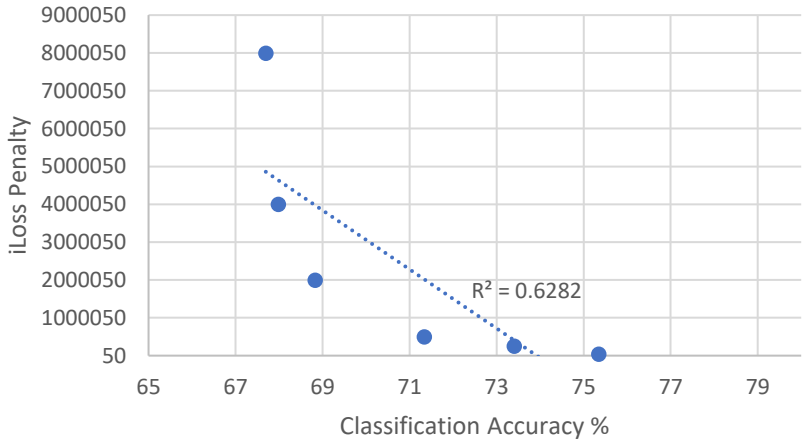
28c. – Correlation between ILoss and Classification Accuracy for Anon/Non-Anon Scenario

Figure 28 - ILoss / Classification Accuracy Correlation

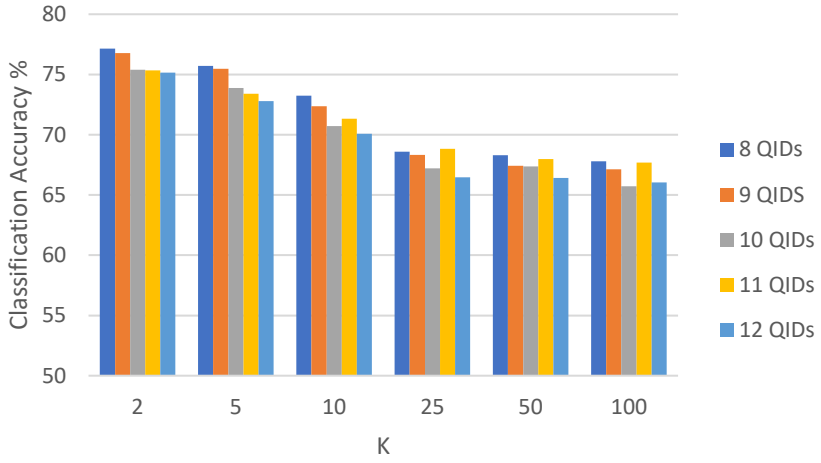
Similar to the DM section, Figure 28 shows scatter graphs for the three experimental scenarios involving anonymised data, this time showing the correlation between ILoss and classification accuracy. In comparison to DM, we can see a much stronger correlation between this metric and the classification accuracy measured. Considering this correlation and Figure 27 again, it is notable that generally, ILoss tends to increase with QID count, suggesting that we should expect classification accuracy to decrease with it. This once again raises the question of why we did not see a clear trend in the results for classification accuracy over consecutive QID counts.

If we note that the results in Figure 28 are an average over all QID counts, and algorithm and generalisation mapping types, then we take a closer look at one anonymisation in particular: Figure 29 shows the correlation between ILoss and classification accuracy for the anonymisation created using the *Relaxed* algorithm, with statistical generalisation mapping and 11 QIDs used in the *Anonymised Only* scenario (29a). It also shows again the measured classification results for the *Relaxed* algorithm with statistical mapping of generalisation in the *Anonymised Only* scenario (29b).

Figure 29



29a. – Correlation between ILoss and Classification Accuracy for Anon. Only Scenario with Relaxed Algorithm, Statistical Generalisation Mapping and 11 QIDs



29b. – Classification Accuracy for Relaxed Algorithm with Statistical Generalisation Mapping in Anon. Only

Figure 29 – Correlation between ILoss and Classification Accuracy Highlighting Anomalous Result

We can see here that the correlation, in this case, is much weaker. All other correlations from the same scenario, algorithm, and generalisation mapping type had R^2 values ≥ 0.9. Looking at the respective classification accuracy in 29b (11 QIDs), it would appear that this is an anomaly, almost all

other consecutive QID counts show a general decline in classification accuracy for all *k*-values. This further supports the idea proposed in the previous section [5.1] that the variability shown in classification accuracy for consecutive QID counts is likely primarily due to random elements. As suggested, these could be eliminated with sufficient repetitions, whereby, based on these ILoss measurements and the literature [Shao & Beckford, pp.6], we would expect then to see a decline in classification accuracy with an increase in QID count, at least in this scenario.

Classification Metric

This part of the section examines measurements of the Classification Metric (CM) on anonymisations. Once again, Figure 30 shows the measurements with 30a showing the average over all QID counts, and 30b and 30c showing measurements from the *Relaxed* and *Strict* algorithms, respectively.

Figure 30

Figure 30a – Average CM over all QID counts for each anonymisation

Anonymisation	Avg. CM
Relaxed *k*=2	0.138733
Relaxed *k*=5	0.175242
Relaxed *k*=10	0.19529
Relaxed *k*=25	0.217456
Relaxed *k*=50	0.222662
Relaxed *k*=100	0.225355
Strict *k*=2	0.146415
Strict *k*=5	0.162585
Strict *k*=10	0.173884
Strict *k*=25	0.185994
Strict *k*=50	0.196957
Strict *k*=100	0.20804

Figure 30b – Relaxed CM

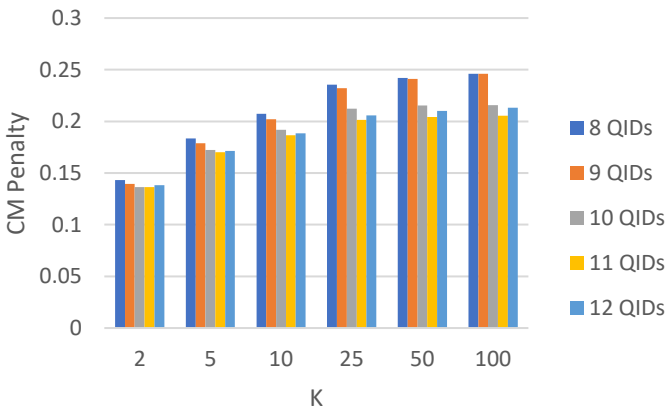


Figure 30c – Strict CM

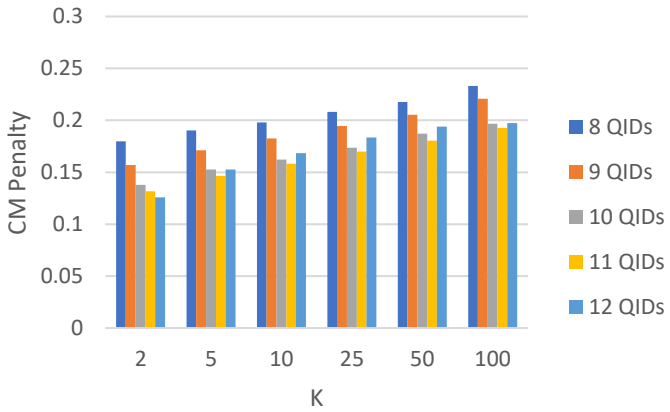
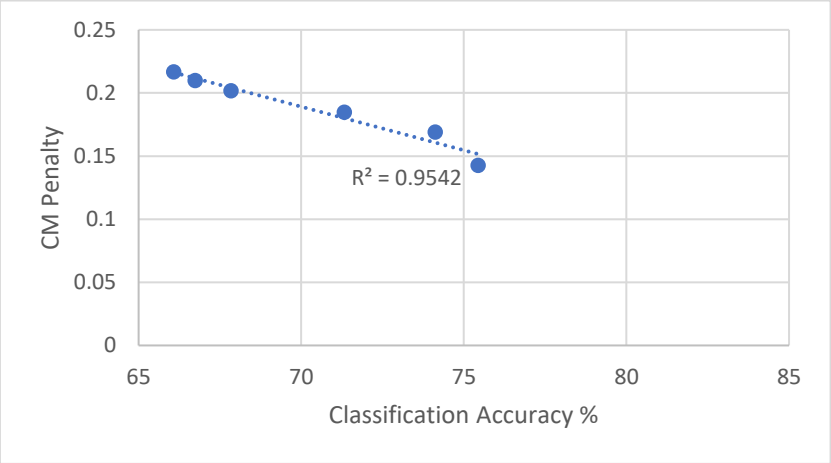


Figure 30 - Classification Metric Results

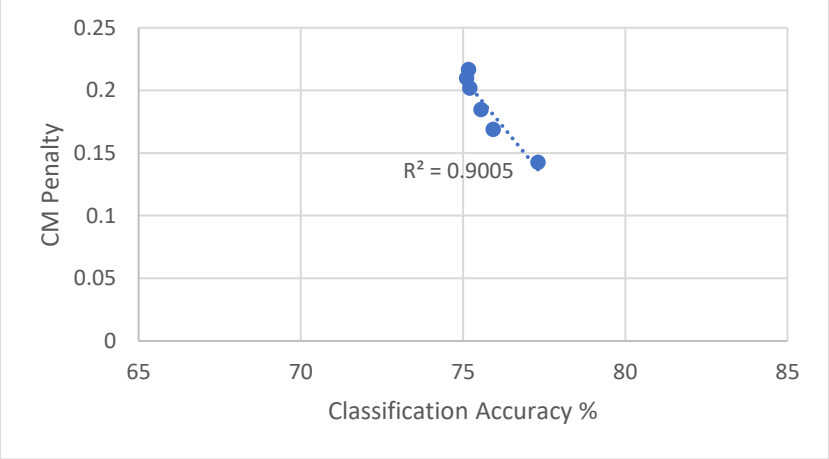
Similar to ILoss, this metric considers individual records, however in this case the class attribute is considered rather than generalisation size and count. This would seem to be the most specialised metric of the three tested, specifically focusing on classification with anonymisations. We can see there are generally similar results between the two algorithms, although there is a very slight increase in the *Relaxed* algorithm for most cases. This slight increase is likely simply because the two

algorithms result in *different* equivalence classes in their anonymisations, this metric compares individual records to their equivalence classes, assigning a penalty if the individual record is the minority in its equivalence class. Hence, you would expect *different* equivalence classes to result in changes to penalties for individual records.

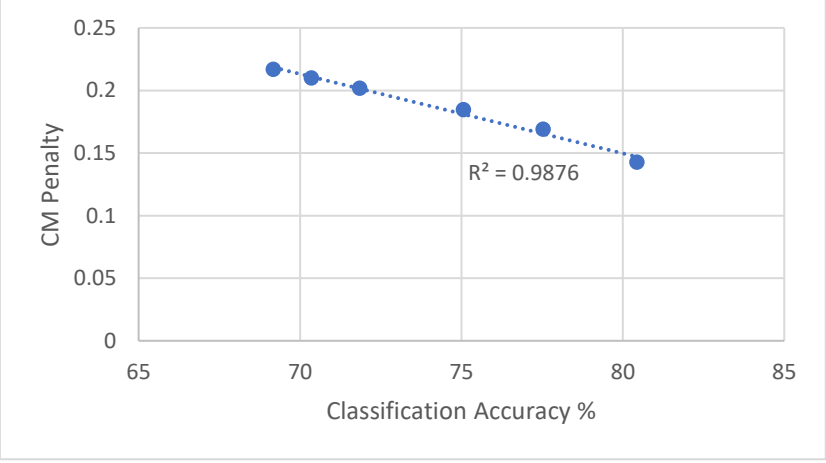
Figure 31



31a. – Correlation between CM and Classification Accuracy for Anon. Only Scenario



31b. – Correlation between CM and Classification Accuracy for Non-Anon/Anon Scenario



31c. – Correlation between CM and Classification Accuracy for Anon/Non-Anon Scenario

Figure 31 - Classification Metric / Classification Accuracy Correlation

Finally, Figure 31 shows the correlations between CM and classification accuracy in each of the three scenarios previously mentioned. As with ILoss, we can again see a strong correlation between this metric and the measured classification accuracy. Due to the nature of this metric and its original purpose of being a metric to better estimate the utility of anonymisations regarding classification, these results are to be expected.

It is interesting to note that for each of the three metrics tested, the scenario where Anonymised records were classified using a decision tree trained on Non-Anonymised data shows the weakest correlation of all scenarios [Figs. 26b, 28b & 31b]. The reason for this is likely one that has been previously stated: the final classification accuracy using the ID3 algorithm is affected more by anonymisation during the training of the decision tree than during the classification of records. In the *Non-Anonymised / Anonymised* case, anonymisation in the training part of the algorithm is eliminated, and the classification results became more stable whilst the metric measures showed a disproportionate change. Therefore, these results would appear to support this statement.

The benefit of evaluating these metrics regarding the main aim of this report is now hopefully clear to the reader. If a particular metric can be proven to be reliable in estimating the utility of anonymisations for specific tasks, then this metric could be used by data publishers to quickly evaluate their anonymisations regarding the said task. With enough of these, the publisher would be able to select the best anonymisation for all potential uses. They would not need to perform arduous experiments similar to those found in this report. It is clear that naturally, the latter two metrics tested are more suitable for estimating the utility of anonymised data regarding classification of the ID3 algorithm specifically – further tests would be required to confirm how useful these metrics are for other classification tasks. It should be said that the Discernability Metric still has its uses and should not be disregarded. I believe that these results show that it is important when performing tasks such as evaluating anonymisation algorithms or deciding on parameters used to anonymise data for publishing, a range of metrics should be used to create a clearer overall picture.

5.3 - Conclusions

Overall, what we can see from these results is that you can expect the anonymisation of data to harm the utility regarding decision tree classification. Obviously, these results only relate to the ID3 decision tree algorithm specifically – more research is necessary to draw general conclusions. That said, it would not be unreasonable to suggest that anonymisation would have a negative effect on ML classification in general. Logically, you would expect this to be the case.

Clearly, there are factors that affect the loss of utility from anonymisation. Primarily, the scenario in which the data is used. We saw from the results in this report that generally the scenario whereby anonymised data was classified using a decision tree trained on non-anonymised data obtained the most accurate classification. In addition, for the lower values of *k* – more commonly used in practice, the *Anonymised / Non-Anonymised* scenario performed just as well, if not better. Both scenarios resulted in more accurate classification than the *Anonymised Only* scenario. The seemingly obvious recommendation to data publishers would be to try to find ways of utilising their data such that the need for anonymisation is minimised. Of course, this is not always possible, and due to legal regulations, it is harder than ever.

That said, from these results, in the *commonly used* range of *k*-values, classification resulting from anonymised data alone still had relatively good measurements for classification accuracy. Clearly then, the parameters used in anonymisation are also important. For *k*-Anonymisation with the Mondrian algorithm, lower values of *k* are certainly recommended; they will generally provide plenty

of privacy whilst maintaining maximal utility. There is no conclusive evidence to support either of the two types of Mondrian algorithm: with both the *Strict* and *Relaxed* versions generally performing similarly. Finally, from the results, it would appear that integrating statistical information does not induce noticeably better decision tree classification results. Certainly not enough to merit the extra effort required to release such statistics. It is also inconclusive as to whether these statistics could result in compromised privacy, something that must be avoided at all costs.

Unfortunately, perhaps due to limited repetition of the experiments, the original plan of making observations about the classification accuracy regarding the number of QIDs used in anonymisation could not be performed. The results showed too much variability in this regard to draw any real conclusions.

Considering the metrics also evaluated, we saw that some metrics performed better than others when the estimation of classification accuracy was considered. As mentioned, it is my recommendation that when evaluating anonymisation utility, where experiments such as those in this report are not performed, it is important to use a range of metrics to get a good overview of utility in different situations. Doing so will enable maximisation of utility *and* privacy when anonymising the data. For classification using decision trees, the *ILoss* and *CM* measures would be the recommendations of this report.

In conclusion, whilst it would seem that loss of utility in decision tree classification appears to be inevitable in almost all situations where anonymisation is involved, the *amount* of lost utility is key. For all commonly used *k*-values in *all* situations, the loss of classification accuracy was relatively small. Data publishers often do not have a choice in whether or not to anonymise their data, but based on the results from this report, they should not be concerned about rendering their data useless by performing such anonymisation. For researchers using the data, some results in this report have proven that we should not place our complete trust in mathematical models. Some nuance is necessary when deriving conclusions purely based on *any* model.

6 - Future Work

Essentially, the main aim of this report was to provide insight into the utility of anonymised data regarding classification. There is great scope for expanding this. One of the motivations for this report was due to there being relatively few examinations of the utility of anonymised data, which seems like an oversight on the side of research because the entire justification for collecting personal data is to use it for research purposes, commercial or otherwise. Therefore, future work could include a more comprehensive evaluation of anonymisation utility for other classification methods such as *SVM* or *Naïve Bayes*, or other statistical models such as linear regression.

Further to this, it would make sense to more closely examine other PPDP methods or algorithms in the above regard. This report only examined one method (Mondrian *k*-Anonymity), while it is a popular method, there are many others that have uses in different situations. It is important that an array of methods are tested in order to provide proper insight.

In addition, as shown, metrics can be a reliable method of estimating the utility of anonymised data. It would not be inconceivable to create a tool that could be used by data publishers to quickly evaluate their anonymised data. Perhaps the tool could aggregate measurements of an array of metrics and provide a score to the data publisher that would allow them to quickly determine the utility of their data for a range of tasks. Further, taking *k*-Anonymity as an example, this tool could even be configured to automatically find the best parameters for a *k*-Anonymisation based on the aggregate scoring from the metrics. If the purpose of the anonymised data is known, metrics could be selected manually by the user to get a quick, specialised evaluation. Such a tool would be relatively simple to implement, however, further research may be required to find the optimal specialised metrics for specific tasks.

7 - Self-Reflection

My initial reasoning for expressing interest in this project was due to my personal interest in the individual's right to privacy, something which I see being invaded to a greater degree more often than ever before. I had initially intended to take on a project that would be primarily practical, simply because this is something that I felt I would be more comfortable with. However, having spoken with Dr Shao about the project, the research aspect of it appealed to me and I decided that it would be a good idea to step out of my usual boundaries.

I believe that I have learned a great deal from this project overall. Primarily in regard to the knowledge gained in the field of data management, an area that was very unfamiliar to me before the beginning of this project. I had to essentially start from the ground up on this project, with no notions on the concept of PPDP and *k*-Anonymity, and only a very limited knowledge of machine learning and decision tree classification. Furthermore, I do not come from a statistics background, having only basic knowledge and no formal education in it specifically. This aspect was particularly challenging for me, especially when collating and representing the data in such a way as to infer information and meaning, whilst also making that clear to the reader. I believe the work done in this regard should stand me in good stead for any similar future projects.

Along with the improved knowledge on the subject matter, I believe I have certainly improved in the soft skills associated with project management. Having to attend regular weekly meetings made it imperative for me to have new content to discuss every week, to ensure this I would make a to-do list and schedule of work for the week leading up to the next meeting. I also think that the project has allowed me to think more critically about any given subject when reading the associated material. Having to read a great deal of scientific literature, analysing specifics, and trying to find

connections is something that I had very little experience with until now. In addition, the writing of this report required me to think scientifically, considering every angle, and being comprehensive in my processes. All of these things are non-specific skills that have been practised by undertaking this project and will be useful in many other situations.

For the practical side of the project, I do not think I gained any *new* skills in this regard. I already had a good amount of experience with Python from previous projects in my undergraduate degree course. If I were to be critical of myself, I would say that this was the part of the project that I could have done better, purely due to the planning aspect. I believe I am a competent coder, but I tend to jump into coding before planning everything down to the minutiae, my experience with Python may have added to my hubris in this case. This works for me on most occasions with smaller projects, but for larger projects, it is likely to result in errors. As was the case in this project when I had to essentially restructure the entire Mondrian *k*-Anonymity implementation due to its inefficiency and lack of clarity in code. In future projects I think I should make more of an effort to plan my code, perhaps utilising UML, or a more comprehensive design document such that a structure is settled before coding begins.

Overall, I believe I have done a good job on this project. I have been thorough in my investigation of the subject material, ensuring I had a good understanding of all concepts when necessary. Despite the lack of proper planning in the early stages, I think that the finished *k*-Anonymity tool and ID3 implementation are of good quality. I also think that I have done a good job in establishing a detailed methodology and providing clear results. Finally, I think that I have been somewhat successful in providing an aspect of originality in my project.

References

Adult Census Income. Kaggle. 2016. Version 3. Available at: https://www.kaggle.com/uciml/adult-census-income [Accessed: 12 April 2021].

Aggarwal, C. & Yu, P. 2008. *A General Survey of Privacy-Preserving Data Mining Models and Algorithms. Privacy-Preserving Data Mining: Models and Algorithms. Springer, 2008,* pp. 12-51.

Ayala-Rivera, V. et al. 2014. *A Systematic Comparison and Evaluation of k-Anonymization Algorithms for Practitioners. Transactions on Data Privacy (7), 2014,* pp. 337-370.

Bayardo, R. J. & Agrawal, R. 2005. *Data Privacy Through Optimal k-Anonymization. Proceedings of the 21st International Conference on Data Engineering, 2005.*

Blum et al. 2013. *A learning theory approach to noninteractive database privacy. Journal of the ACM, 60(2), pp. 1-25.*

c0fec0de. 2010. *AnyTree.* Version 2.8.0 [Python Library]. Available at: https://pypi.org/project/anytree/ [Accessed: 15 April 2021].

Dua, D. & Graff, C. 2019. *UCI Machine Learning Repository: Adult Data Set*. *Irvine, CA: University of California, School of Information and Computer Science.* Available at: https://archive.ics.uci.edu/ml/datasets/Adult [Accessed: 12 April 2021].

El Emam & Dankar, F. K. 2008. *Protecting Privacy Using k-Anonymity*. *Journal of the American Medical Informatics Association, 15(5), 2008*.

European Parliament. 2016. *General Data Protection Regulation.* Available at: https://gdpr-info.eu/ [Accessed: 10 April 2021].

Frank, E. et al. 2012. *simpleEducationalLearningSchemes: Simple learning schemes for educational purposes (Prism, Id3, IB1 and NaiveBayesSimple).* Version 1.01 [Weka plugin]. Available at: https://weka.sourceforge.io/packageMetaData/simpleEducationalLearningSchemes/index.html [Accessed 16 April 2021].

Fung, B C M. et al. 2010. *Privacy-preserving data publishing: A survey of recent developments*. ACM Computing Surveys, 42(4).

Fung, Benjamin C. M. 2007. *Privacy-Preserving Data Publishing.* PhD Thesis, Simon Fraser University.

Iyengar, V. S. 2002. *Transforming data to satisfy privacy constraints. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 279-288.

Lapowsky, I. 2019. How Cambridge Analytica Sparked the Great Privacy Awakening. Available at: https://www.wired.com/story/cambridge-analytica-facebook-privacy-awakening/ [Accessed: 10 April 2021].

LeFevre et al. 2005. *Incognito: Efficient Full-Domain K-Anonymity. Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 49- 60.*

LeFevre, K. et al. 2006. *Mondrian Multidimensional k-Anonymity. Proceedings of the 22nd International Conference on Data Engineering (ICDE'06).*

Li, J. et al. 2011. *Information based data anonymization for classification utility. Data & Knowledge Engineering (70), 2011,* pp. 1030-1045.

Machanavajjhala, A. et al. 2007. *ℓ-Diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD),* 1(1), pp. 3-es.

Majeed, A. & Lee, S. 2020. *Anonymization Techniques for Privacy Preserving Data Publishing: A Comprehensive Survey*. IEEE Access, volume 9, pp. 8512-8545.

Maldoff, G. 2016. *How GDPR changes the rules for research.* Available at: https://iapp.org/news/a/how-gdpr-changes-the-rules-for-research/ [Accessed: 10 April 2021].

Mitchell, T. 1997. *Machine Learning*. *McGraw Hill.*

Mooney, S. 2019. *The impact of GDPR on research.* Available at: https://universityobserver.ie/the-impact-of-gdpr-on-research/ [Accessed: 10 April 2021].

Quinlan, J. R. 1986*. Induction of decision trees. Machine learning*, 1(1), pp.81-106.

Quinlan, J. R. 1993. *C4.5: Programs for machine learning. Morgan Kaufmann Publishers, inc. 1993.*

Samarati, P. & Sweeney, L. 1998. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.*

Shao, J. & Beckford, J. 2017. *Learning Decision Trees from Anonymized Data. 8th Annual International Conference on ICT: Big Data, Cloud and Security, Singapore, 2017.*

Skowron, A. & Rauszer, C. 1992. *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Set Theory.*

Storgaard, E. 2019. *The Effect of GDPR on Health Research.* Available at: https://www.farrinstitute.org/news/the-implications-of-gdpr-on-health-research [Accessed: 10 April 2021].

Sweeney, L. 1998. *Datafly: A system for providing anonymity in medical data. Database Security XI. Springer, Boston, MA. pp. 356 – 381.*

Tang, Q. et al. 2010. *Utility-based k-anonymization. The 6th International Conference on Networked Computing and Advanced Information Management*, pp. 318-323.

University of Waikato, 2021. *Weka*. Version 3.8. Available at: http://old-www.cms.waikato.ac.nz/~ml/weka/ [Accessed: 16 April 2021].

Xiao, X. & Tao, Y. 2006. *Personalized privacy preservation. Proceedings of the ACM SIGMOD Conference. ACM.*