



TITLE: Implementation of a data privacy protection method for transaction data

MODULE: CM3203

STUDENT: DOMINYQUE MOHAMMED

STUDENT NUMBER: C1839296

SUPERVISOR: JIANHUA SHAO

MODERATOR: JAMES OSBORNE

Abstract

Data privacy is a growing concern and branch of study in computer science as every day, more and more data is collected around the world. Data protection tools like anonymisation algorithms have been a key solution to the data privacy issue and are still researched today. There are two major aspects to ensuring an anonymisation technique is efficient and those are: anonymising the data, and simultaneously making sure that the information loss is not too great during the anonymisation process due to the generalisation or suppression techniques used.

The main goal of this study was not only to implement an existing anonymisation algorithm but to also create an educational tool that lecturers can use to give a visual explanation of how the algorithm works. The data protection tool implemented was the 'k-anonymisation approach using clustering techniques', that has the promise of guaranteeing a reduction in information loss and data quality when compared to other anonymisation algorithms.

Python was used to implement the 'k-anonymisation with clustering' algorithm. With this approach being a 'k-member clustering problem', the typical understanding of clustering changes in this scenario to mean that each cluster needs to contain at least k records; this idea is not a requirement in standard clustering problems. After the algorithm was implemented and shown to be running correctly, an interface was then attached to it. This would allow users to select which anonymisation algorithm they would like to test out, upload their datasets or use the pre-loaded dataset, and get a breakdown of the functionality happening, i.e., receive explanations and receive a new text document containing the anonymised data. This project was made in collaboration with another student who did another anonymisation algorithm. However, due to the time constraints of this project, that approach has not been added to the interface. Hence, the user only has the anonymization approach discussed in this report to use.

Based on the work done, it shows that with the smaller k is, coupled with a large dataset, the lesser the information loss was, which was as expected. The project produced presents this information in the form of an educational tool so that users can see how and why we arrive at that conclusion of results. The educational tool does have the ability to be in continued development as there are many more existing anonymisation approaches that can be added to it. The addition of more algorithms to the tool will allow users to have a clear visual of the results and gain the ability to easily make comparisons on the different methods.

Acknowledgements

First and foremost, I have received such a great deal of support, motivation, and assistance from my supervisor, Mr. Jianhua Shao. He guided me throughout the project by taking the time to explain the basic concepts to me, answering all my questions, as well as, imparting multiple scientific articles to assist me in better understanding the topic area. He was the source of invaluable feedback and advice during the course of the development process in this project.

And lastly, I would like to thank my family and friends for lending me a sympathetic ear when it was needed and for giving me strong words of encouragement that helped power me through completing this project.

Table of Contents

Abstract	2
Acknowledgements	3
1. Introduction.....	5
2. Background.....	7
Identifiers	7
Generalisation	7
Clustering	8
Educational Tool	9
Metrics	10
3. Specification & Design.....	11
Project scope.....	11
User Interface	12
Changes to User Interface Design- final state.....	13
Data Flow	14
Activity Diagram.....	15
Static Architecture	16
4. Implementation.....	17
Structure of Algorithm	17
Dataset structure	18
Breakdown of Important Functions.....	18
Problems Encountered	22
Instructions for Running the Tool	23
5. Results and Evaluation	24
Does the system meet requirements?.....	24
Evaluating Performance of Anonymisation Algorithm	25
Critical Appraisal of Overall Project	27
6. Future Work.....	28
7. Conclusions.....	29
8. Reflection on Learning	31
Table of Abbreviations	33
Glossary	33
Appendix.....	35
References	40

1. Introduction

Companies, governments, and individuals are collecting information every day to store and analyse that information so that they can make better decisions in the long run [1]. Certain organisations like hospitals for example, are required to publish their data to improve patient care and obtain important statistics for certain diseases, quality of care, etc. However, publishing the raw data will be a violation of privacy as the private information of those individuals would be revealed. Thus, several algorithms and approaches have been created to ensure that data can be useful while still maintaining that individual privacy, this is called ‘privacy-preserving data publishing’ (PPDP). PPDP seeks to hide the sensitive data of record owners thereby guaranteeing these individuals privacy protection. Maintaining the privacy of record owners is not only for safety precautions but to also prevent cases of fraud, identity theft, and loss of trust in the companies publishing the data. We can imagine that hiding more data values in the raw data will fix the privacy problem, which it will, but this greatly increases the information loss and vice versa if we were to show more data values rather than hiding them. Hence, the approaches proposed for solving this problem need to consider both the privacy and the usefulness of data.

As mentioned before, a majority of the solutions/ approaches that aim to address this privacy issue rely on the use of a concept called k-anonymity. K-anonymity states that any record in the data is indistinguishable from at least $(k-1)$ other records for a set of attributes called the quasi-identifier [2]. The different types of approaches/ algorithms either fall into one of these two types: a hierarchy based generalisation and a hierarchy-free generalisation.

With a hierarchy based generalisation, which is the focus of this project, one will need to have a hierarchy of generalisations/ relationships for each quasi-identifier (qid). Whereas, hierarchy-free generalisation, also known as set-based generalisation, it is a flat hierarchy, i.e., there is no need for a hierarchy of generalisations of the quasi-identifiers. Of course, both of these classes of algorithms will have their pros and cons. With hierarchy based generalisation, because the user needs to create hierarchies for the qids, the accuracy depends quite heavily on the user’s knowledge and ability to create well-thought-out hierarchies. If the hierarchies are of poor quality then the quality of data will suffer. On the other hand with set-based generalisation, each set is given a score, and therefore, it will not work for certain datasets.

Adding to this algorithm is the clustering technique. Now, clustering usually works on the concept of grouping similar items together. In this algorithm, it is vaguely the same where we group similar records but the difference is that each cluster must contain at least k records. Working clustering techniques into this approach means that information loss is greatly reduced as there is less misrepresentation done when the records in a cluster are changed to have the same qid (because the records in the cluster are already similar) [3]. So putting together the k-anonymity with hierarchy based generalisation and clustering, we have our algorithm.

However, it is not just the algorithm that is tested in this project; the main goal is to turn this into an educational tool. In this tool, the user has the option to view, investigate and obtain explanations and information about the running of the algorithm that they choose.

In conclusion, the main contributions achieved through this project are as follows:

1. Implemented the K-anonymization Using Clustering Technique,
2. Evaluated its efficiency in terms of data utility/ information loss and running time,
3. Created a user interface for users to easily run the algorithm with a pre-loaded dataset and receive metrics.
4. Providing the user with a breakdown of how the algorithm works, as well as an anonymised version of their dataset.

2. Background

K-anonymity works on the basis that the data being anonymised is data that is stored in a table with rows and columns. Furthermore, the k-anonymity model assumes that each record represents a distinct individual [2]. Hence, we can safely assume that in a record (row) each field (attribute) is a piece of information relating to a distinct individual. This k-anonymity problem can be broken down into three parts: the identifiers, the generalisation, and the clustering. Each will be thoroughly discussed and explained.

Identifiers

Certain attributes fall under the term ‘Explicit Identifiers’. These are attributes that directly identify the individual such as name and social security number. As the main aim of these approaches is to hide the identity of individuals, these explicit identifiers need to be removed from the dataset. However, even when all explicit identifiers are removed, privacy still remains an issue. Attackers can look at and group the remaining attributes to identify an individual and these attributes are known as quasi-identifiers (qid). Although these qids are not direct identifiers, they can be used in conjunction with external information to identify or reduce the probability of uncertainty about the identity. A few examples of qids are: race, income, postal code, age, job status, education, and date of birth.

How do we determine what attributes qualify as a quasi-identifier? One method is using logical reasoning. The examples listed above are all indirectly a key characteristic of who you are yet, by themselves do not reveal your identity. Another method is to measure the statistical distribution to find any unique values [4]. For example, if you take the data point of postcode ‘CF10 3AT’ in your dataset; if there are many records with that postcode, then the probability of an attacker identifying that individual is very low. Consequently, if there are a few records with that postcode, then the probability of identification increases.

The third and last identifier that will be spoken about is sensitive identifiers. Quasi-identifiers are sensitive, however, they are publicly known. The sensitive identifiers are typically not public knowledge such as medical records. It is noted that an attribute can fall into multiple categories, for example, a person’s social security number is an explicit and sensitive identifier.

Recalling the definition of k-anonymity: meaning that any record in a k-anonymous table is indistinguishable from at least (k-1) other records with concerning the quasi-identifiers [2]. The outcome of this is that an attacker should not be able to learn anything new about any individual given the anonymised dataset even if they have external information from other sources. The probability of an attacker determining which record in a k-anonymised table corresponds to a targeted individual is $1/k$ [3].

Generalisation

Moving onto another major concept of this algorithm, the generalisation method. It is noted that either generalisation or suppression can be used in strengthening the k-anonymity

algorithm. However, as generalisation was implemented for this project, it will be the only technique discussed in this section. Generalisation is as it sounds; it makes the data values in the data set less specific whilst remaining constant. For example, using the postcode again, if there's a postcode of 'CF10 3AT', this can be generalised to 'CF10***'. This means for every postcode beginning with 'CF10', it will be generalised to that form. Furthermore, generalisation can be done which completely hides the data value and that is simply done by substituting the value with '*'.

Why is this useful? Only the quasi-identifiers are generalised. These are the indirect identifiers that when pieced together can reveal the identity, however, if the data becomes less specific, it significantly lowers the chance of identification occurring.

As mentioned briefly, generalisation falls into two branches: Hierarchy based and set-based. Set-based generalisation/ hierarchy-free generalisation refers to when the k-anonymity problem is converted into a partitioning problem. As this project does not utilise this, it will not be further discussed. Hierarchy based generalisation is when there is a set way for generalising each quasi-identifier, i.e., there will be a corresponding data file for each qid that stipulates how the attribute should be generalised. An example is shown in Fig1, which shows how the attribute of 'postcode' should be generalised. The asterisk (*) at the top indicates the highest form of generalisation that can be done, completely hiding the data value. As you descend through the graph, the data value becomes more specific[5].

As always, there are disadvantages and advantages to each type of generalisation. In this case, the efficiency and accuracy of hierarchy based generalisation depends on the person constructing the hierarchies for the quasi-identifiers. If the hierarchies are poorly done, then the utility of the data is affected. However, the main advantage of this method is the consistency of the data as all data values are given the same generalisation, for instance, all postcodes of CF10 3AT will be generalised to CF10***.

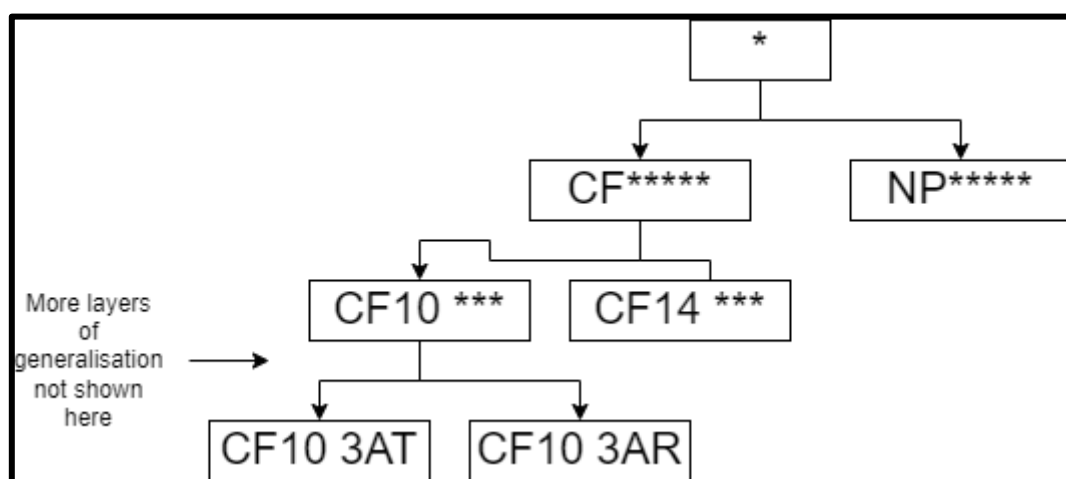


Fig 1: Hierarchy based generalisation of the attribute postcode

Clustering

A key part of this k-anonymisation approach is the clustering technique that has been added. Clustering, as we know it, is the process of grouping similar objects together, and this

concept carries on for this approach but is altered slightly. In terms of k-anonymisation using clustering, it is the process of grouping similar records together but each cluster/group needs to contain at least k records. Therefore, if k is set to 5, each cluster will need to have at least 5 similar records- no less. Fig 2 shows one way how 15 data points/records would be clustered. There is no limit on how many clusters you can or cannot have, the only requirement is the minimum number of records in the cluster.

How do we determine which records are similar? The closest data points are connected to form a cluster and then the next closest data point to those initial data points is then added to form a bigger cluster. This approach iteratively does this until each cluster has at least the minimum number of records allowed, which is k.

Why use clustering? With grouping comparable records together, it helps in reducing the data distortion when the records need to be generalised- modifying their quasi-identifiers. With measuring distance between records in a cluster, it allows for performance metrics to be done, that is, how quickly the program runs and measuring how much information loss there was. These metrics are discussed further in the Specification & Design section.

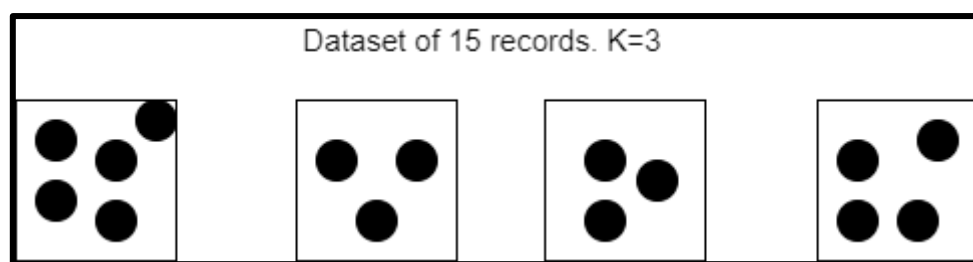


Fig 2: Cluster partitioning example

[Educational Tool](#)

The main goal of this project is not to determine the performance of the algorithm or compare it to another, but it is to create an interface that allows lecturers to showcase exactly how the algorithm works. By allowing students to see the raw data versus what the anonymised version looks like, they can see first-hand how the anonymisation of data can affect how useful it is.

This tool will currently only have one anonymization algorithm attached to it, which is the approach discussed in this paper. However, it is a good starting point to pool multiple of these approaches in one area so that comparisons can be done in lectures to determine which algorithm out-performs the others in terms of efficiency and running time, and/or reducing information loss. This will also directly benefit students as they would have access to the tool and can use it for themselves. This would allow them to explore the tool and the anonymisation algorithms in their own time to ensure they fully grasp the concepts. Learning about anonymisation algorithms, or protecting the privacy of individuals in general, is key for any and every computer science student. As the Internet and the world grows more connected making it much easier for companies to collect data about you, there need to be assurances that your data would not be revealed in such a way that it puts you in danger or makes you

uncomfortable. Therefore, having this tool available for the upcoming generations of developers, would allow them to be aware of the reasons why these approaches exist and teach them how to implement these anonymisation approaches.

Metrics

There are several anonymisation methods/ approaches and some of these algorithms are best suited for a particular dataset. For example, the k-anonymisation with clustering approach works well for categorical data over numeric or continuous data. Therefore, the evaluation of these anonymisation approaches is of utmost importance so that we can select the best approach given the data we have. There are many ways of evaluating these algorithms, such as analysing performance time, the degree of information loss, calculating how a record is indistinguishable from another, measuring the quality of utility by using a workload-based calculation, and much more [6]. This project uses both run time to measure efficiency, and a metric called ‘Normalised Certainty Penalty’ (NCP) which measures the degree of information loss.

What is NCP? NCP penalises items/records based on the way that they are generalised. This is mostly done for categorical attributes, therefore, it fits in quite nicely with this k-anonymisation approach [7]. NCP grants a higher penalty value to records that have high support meaning that these records affect more transactions equalling greater distortion levels. The equation for NCP is as follows:

$$NCP(Dataset) = \frac{\sum_{i \in I} (\text{sup}(i, Dataset) \times NCP(i))}{\sum_{i \in I} (\text{sup}(i, Dataset))}$$

In conclusion, when these three key parts (identifiers, generalisation and clustering) are all integrated into one k-anonymisation algorithm, the outcome produced is an anonymised dataset in which multiple steps are taken to ensure that the utility of information is still of a high degree. Transforming this approach into an educational tool, equipped with valuable metrics, will give the users a better understanding of how the approach works and why these approaches must exist. This approach proves that we can have both privacy and usefulness of data.

3. Specification & Design

Project scope

This project requires the implementation of an existing algorithm, the K-Anonymisation using the Clustering Technique, and attaching a user interface to it, so that it can be used as an educational tool. The targeted audience for this tool would be lecturers for showcasing how the algorithm works when teaching this topic, and the students so that they can try it out for themselves, get a closer look and investigate the features in their own time for better understanding. See Fig 3 for the Use Case Diagram indicating the user's possible interactions with the system. The following points are the user requirements for the tool:

1. The implementation of the 'efficient k-anonymisation with clustering' algorithm produces an anonymised dataset
2. The system is able to provide metrics: these will include performance (running time) and measuring information loss.
3. The system provides the user with an interface that is easy to use- allows for the easy selection of the desired algorithm, all instructions will be easy to follow.
4. The system gives the user the option to add their own datasets and hierarchies and will perform the anonymisation on those datasets.
5. The system will give explanations via the interface as to how the algorithm works.

It is noted that there are restrictions involved in the project such as the timeframe. The project began at the end of January 2022 and needs to be presented in May 2022, allowing for only a few months for the development process.

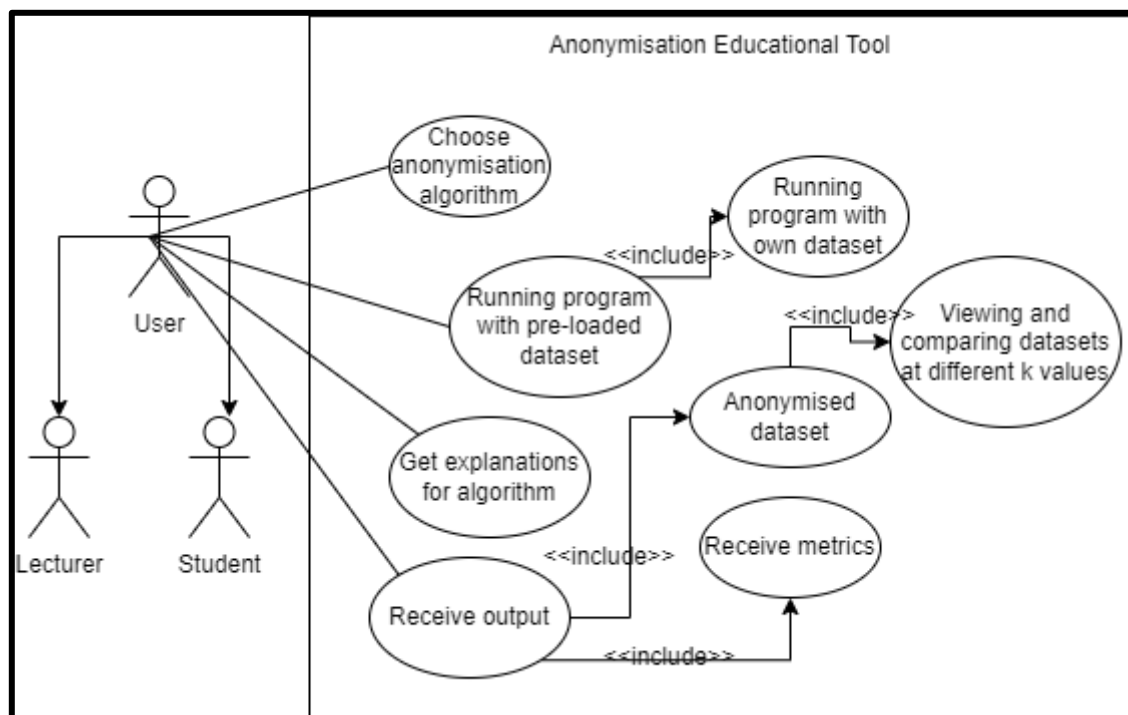
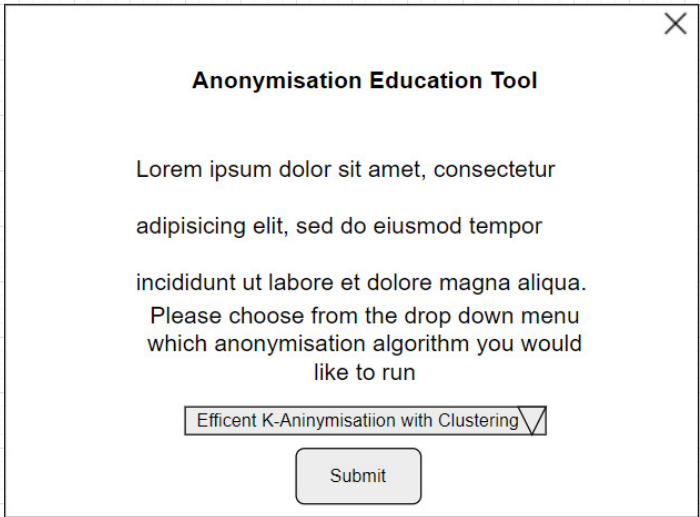
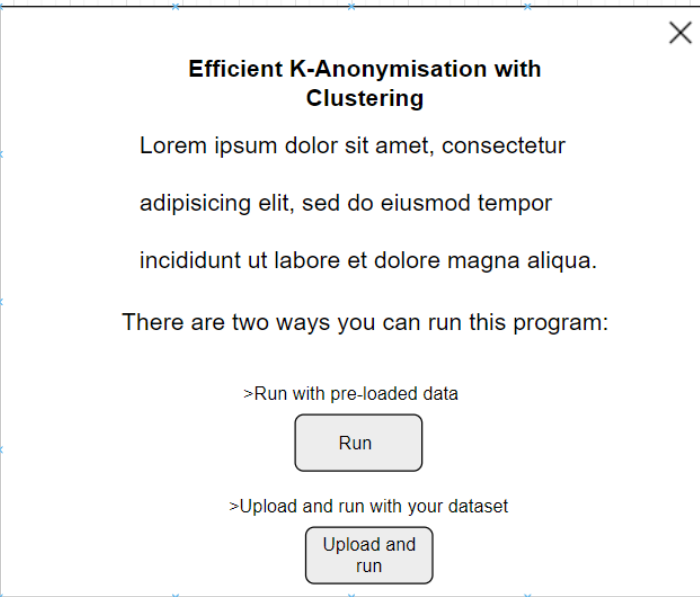


Fig 3: Use Case Diagram

User Interface

The user interface will be simple and easy to use. When allowing the user to choose their algorithm, the choices will be presented via a drop-down menu. This is done so that the user can see all the options available for them to choose from and it makes it much easier to add more anonymisation algorithms in the future than if buttons were used. The majority of the input needed from the user will be done through clicking buttons, no text input would be required from them. The advantage of doing this is to prevent any user errors from incorrect spellings. As this educational tool is very straightforward and has a clear purpose for the information that is needed to be displayed, it calls for a very basic yet effective interface. The following table, Table 1, shows the prototypes of how the interface screens should look. It is important to note that these are the intended designs and the actual user interface might differ slightly.

Screen Number	Function	Prototype
1	This screen will present the user the opportunity to select which algorithm they would like to use as well as gives a slight introduction into what the tool does.	
2	This screen is where the user decides if they want to use the preloaded data or upload their own datasets.	

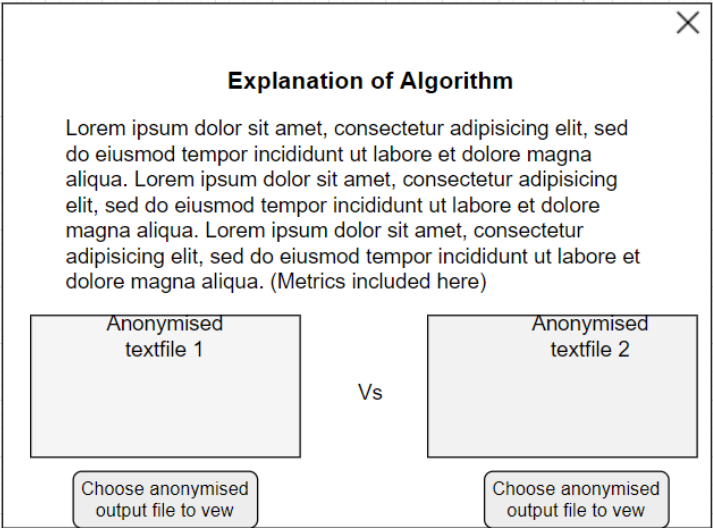
3	<p>Here, the explanation of how the algorithm works is presented. The metrics will be discussed as well- showing the running time and degree of information loss percentages. Even more than that, this screen will allow the user to select two different anonymised datasets (for example, an anonymised dataset when k was 3 vs when k was 5) so they can analyse them side by side.</p>	
---	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------

Table 1: Prototype of User Interface

Changes to User Interface Design- final state

The look of the user interface screens remained fairly constant throughout the development process. However, it is important to note any differences that occurred in the final stage of development of the user interface and state the reasoning for the changes.

- There is a separate screen that only involves the metrics information- this was due to the Python user interface library used- Tkinter, with the way the functions were defined, it did not allow for the metrics to be included in the explanations screen.
- The explanations screen does not have a button bringing the user to another screen that displays an anonymised text file of their choice. Due to the timeframe of the project, the logistics of having two text files on one screen could not be done. To mitigate this, the user can simply open another ‘display text file’ screen and have the two screens side by side to compare.

Data Flow

For this anonymisation approach, there are a few data requirements to follow to receive the best results from the tool. The main data file that contains all the relational information should be in a .data format. Also, hierarchy files for each of the quasi-identifiers will need to be uploaded and these should be in a .txt file. For instance, if the user has 5 quasi-identifiers (occupation, race, sex, income, and marital status) for their raw dataset, then there need to be 5 hierarchal generalisation text files- one for each. The anonymised datasets that are produced from the tool will be in a .data format.

To keep the program organised and ensure that all the aims were being met, a data flow diagram (Fig 4) was done. This helps in giving a visual representation of the educational tool and the way it should work. It showcases the way information flows through the different processes that are required for the tool.

The major processes outlined are: Creating clusters, performing generalisations, production of explanations, and creating metrics.

The main data stores outlined are: Raw data file (initial .data file), clusters (which are stored in memory), hierarchies of quasi-identifiers (.txt files), anonymised data (.data files), and explanation data (stored in memory).

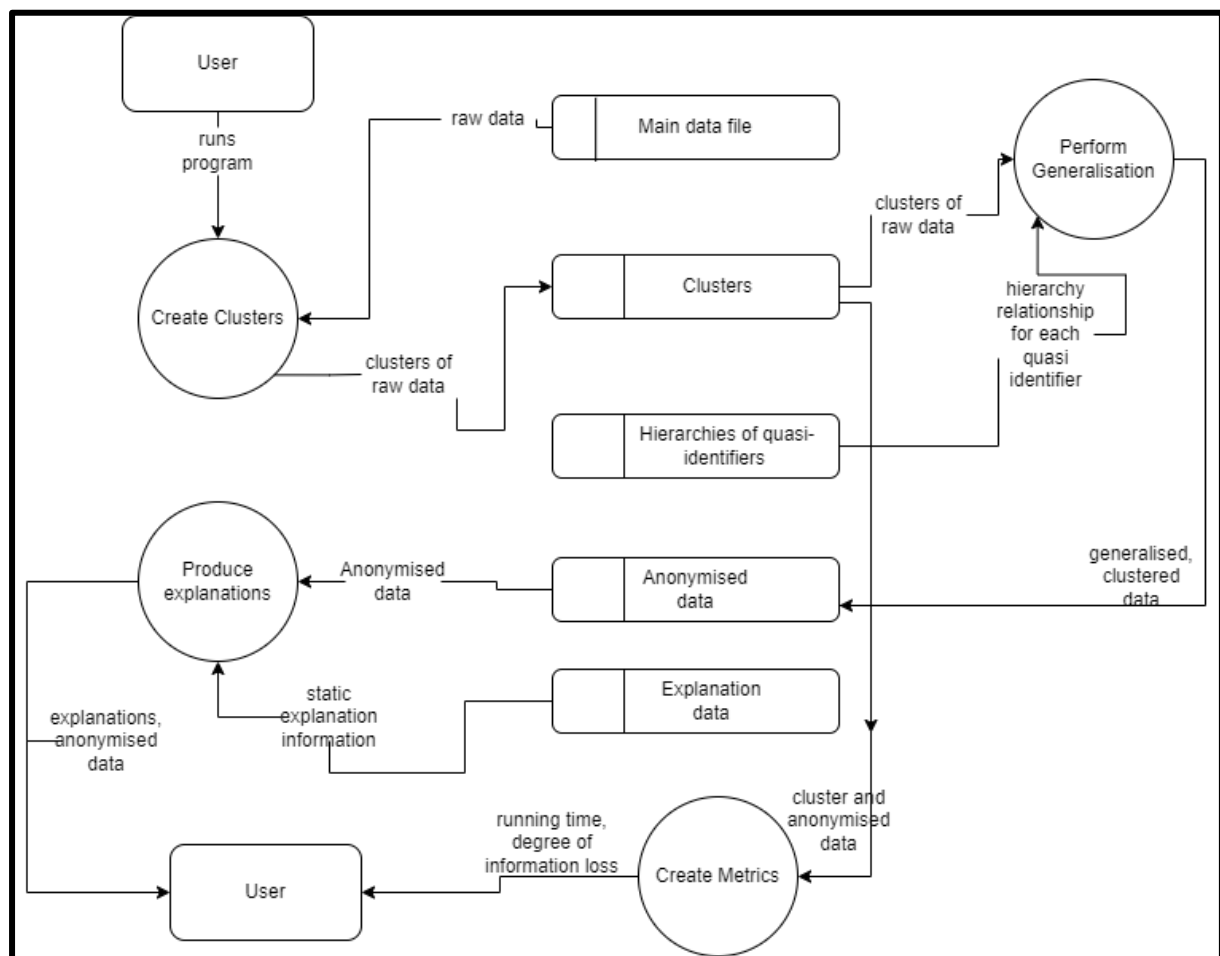


Fig 4: Data Flow Diagram showing the flow of data

Activity Diagram

The activity diagram is a way of graphically representing the flow of logic for the system being designed. In this case, it looks at the possible choices a user can make and based on those choices, the functions the program will run. Alongside the use case diagram (Fig 3), activity diagrams give an overall view of the dynamic architecture of the system. It shows the changes that are being made by the system in accordance with the user input it receives.

In this case, the user has a choice to make as soon as the tool is started- the choice of which anonymisation algorithm is to be used. For this project, we will only focus on the use case where the user picks the K-Anonymisation with Clustering. The other area where user input is required is the input for running the algorithm- does the user want to use the pre-loaded data or their own datasets? If the latter is chosen, then there are a few extra steps of uploading those files and setting the quasi-identifiers (so that the algorithm will know which attributes need to be modified in the dataset). After that choice has been made, the algorithm becomes static, in terms of, nothing else will be affected by user choice (Fig 5).

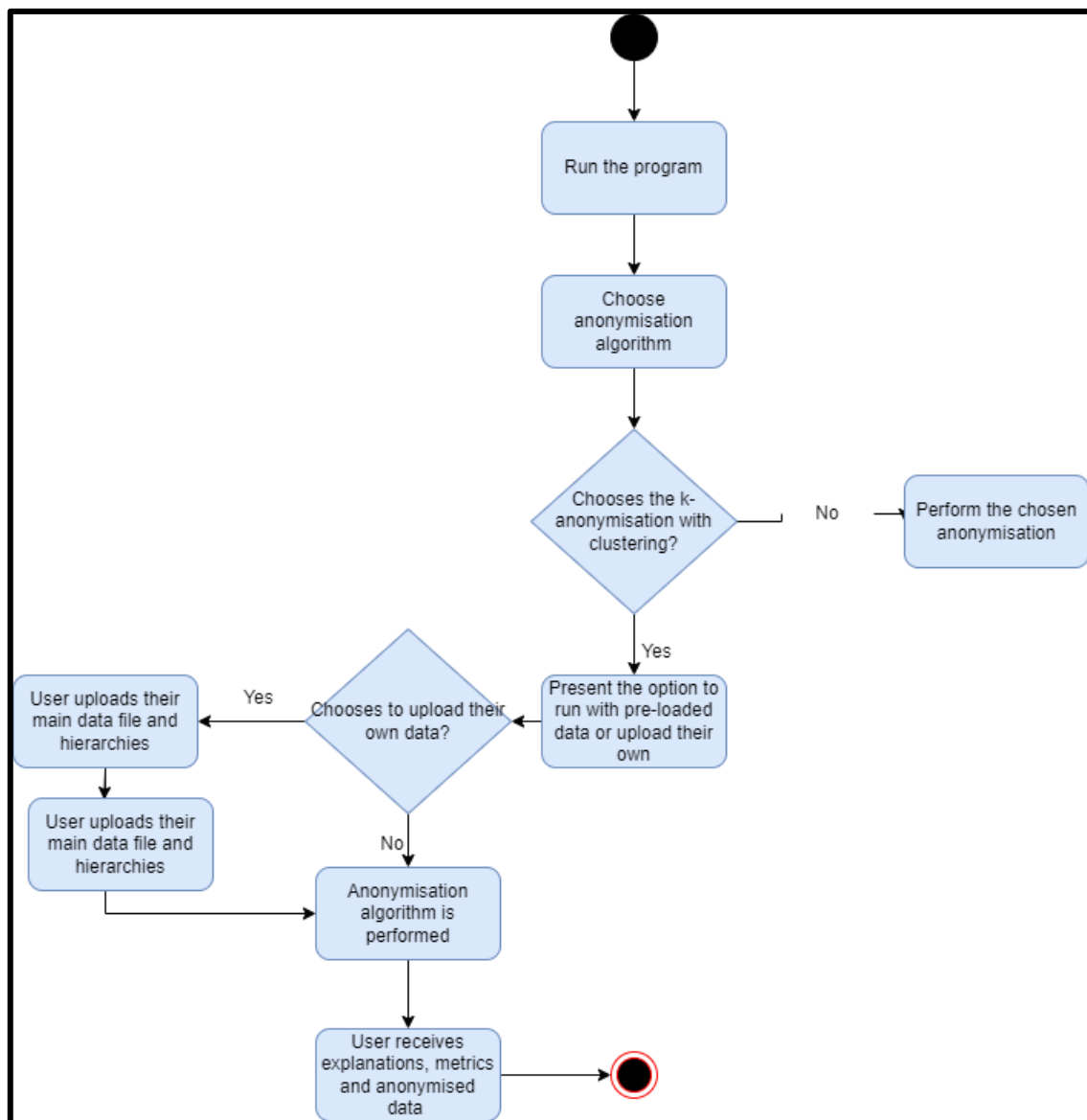


Fig 5: Activity Diagram showing the flow of logic

Static Architecture

Although for this project, an existing algorithm is being implemented, as more components are being added to it to create functionality for the educational tool, a class diagram was done (Fig 6). This allows for planning the objects and classes that will be required for the project.

For this project, the main class needed is the interface- as everything happens through it, i.e., the user makes all decisions using this class. All other important classes need to communicate with it. Another important class is the Cluster class where all the functionality for the k-anonymisation with clustering problem is stored which will be explained in more detail in the 'Implementation' section. The Run class allows the user to run the anonymisation algorithm and get results without going through the interface. It is also the class that holds the function for writing the anonymised data to output files. The last two classes: GeneralisationTree and Read_Data are respectively for generalising the data and reading the raw main data file for the preloaded data.

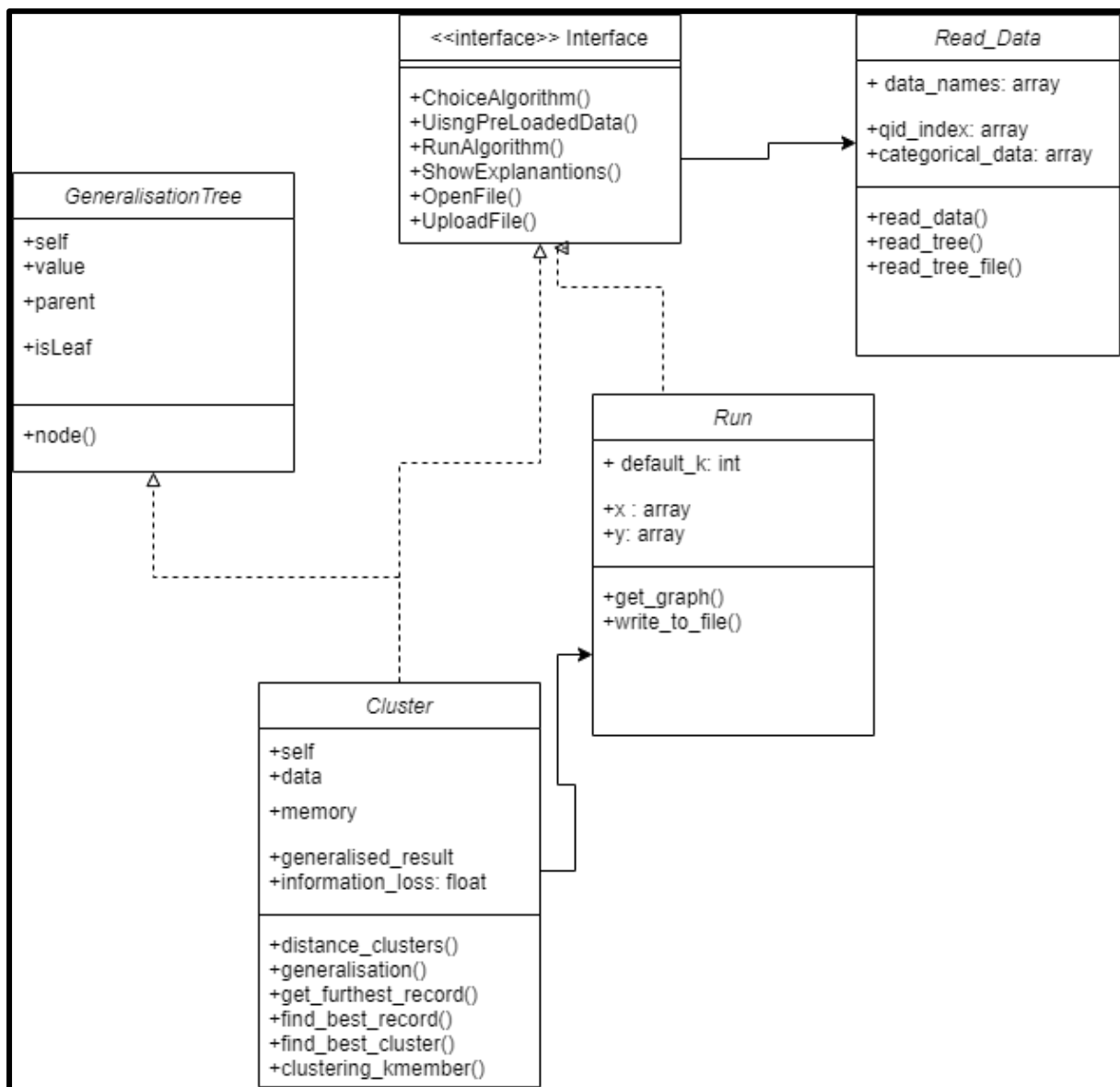


Fig 6: Class Diagram showing the static structure of the system

4. Implementation

Structure of Algorithm

The K-Anonymisation with Clustering approach was implemented using Python 3.8.3 with Sublime Text 3 as the IDE. The algorithm was run on a 2.60GHz Intel® Core™ laptop with 16GB of RAM. The operating system on the laptop was Microsoft Windows 10 Pro. There were a few Python libraries that were installed and used throughout the development process and these are:

1. Random: This was used to pick and retrieve the first random record for the clustering to commence
2. Time: Calculating the running time- how long the clustering process takes
3. Functools: this was used to return functions and other such callable objects. Such methods being: cmp_to_key(func)
4. Matplotlib.pyplot: this module was used to create a graph of the value of k versus the degree of information loss.
5. Tkinter: this is the GUI toolkit for Python and thus was used to create the user interface for the educational tool.
6. unittest: this framework allows for the construction and running of tests against the algorithm to ensure it outputs the correct results.

The structure of algorithm is depicted in Fig 7.

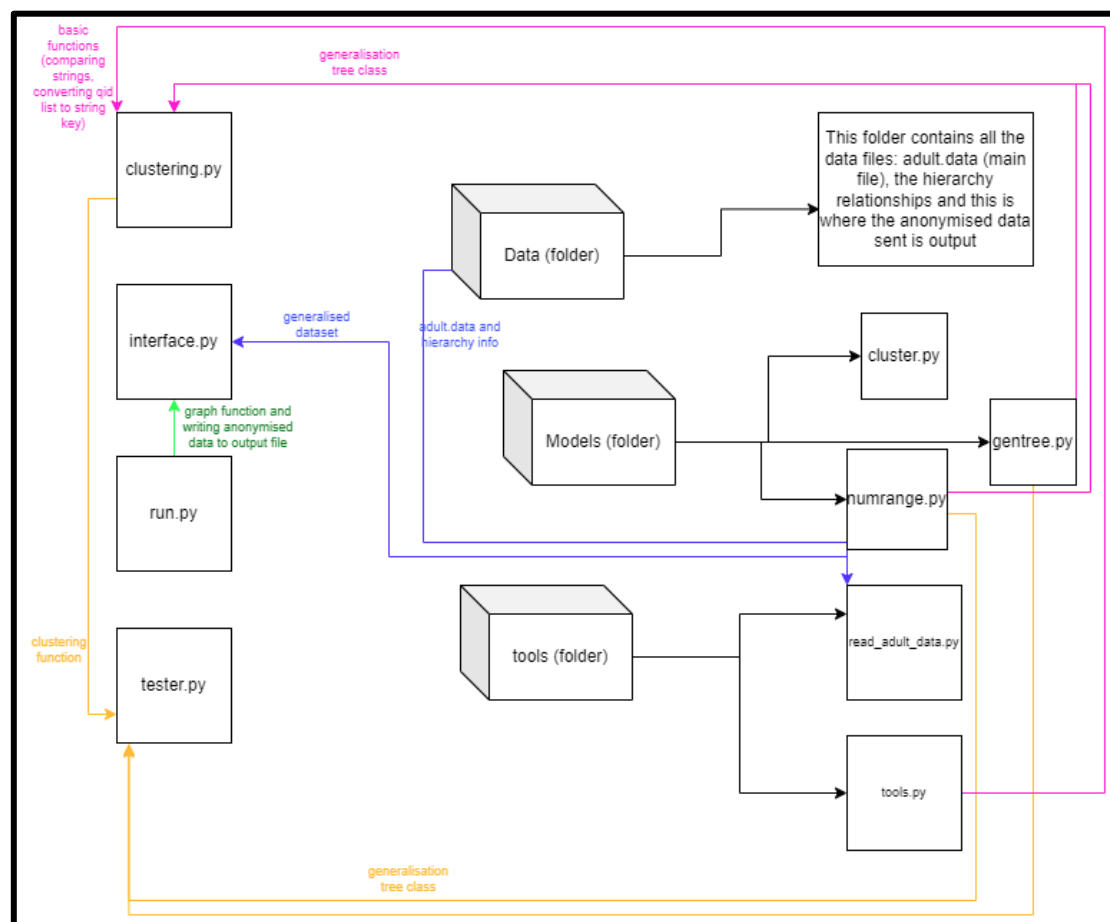


Fig 7: Structure of the python files showing file and function dependence

Dataset structure

For this project, there is the main data file called 'adult.data'. This file is from the UC Irvine Machine Learning Repository [8] and one thousand rows were selected to be used for this report. This was done to shorten the processing time during the development process when testing the algorithm. As well, all records with missing values were removed and eleven of the original attributes were retained. These attributes are: age, work class, education, marital status, occupation, relationship, race, sex, hours worked per week, native country, and income. The attribute 'age' was treated as a numeric attribute while the rest was treated as categorical. Of these eleven attributes, seven of them were chosen to be quasi-identifiers: work class, marital status, occupation, relationship, race, sex, and native country. As mentioned previously, each quasi-identifier needs to have a generalisation hierarchy and these are the names of those files: adult_workclass.txt, adult_sex.txt, adult_relationship.txt, adult_race.txt, adult_origin_country.txt, adult_occupation.txt and adult_marital_status.txt.

After the anonymisation algorithm is run, it is set to output an anonymised dataset from each value of k starting from 2 and ending at 10. These output data files are saved to the data folder where the above datasets are held. The structure of the name of the anonymised dataset is saved as follows: anonymised_k_is_x, for example, anonymised_k_is_2 or anonymised_k_is_7.

Breakdown of Important Functions

The existing algorithm of the k-anonymisation using clustering came from a research paper written by the individuals who created the approach: Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li [3]. The paper detailed the necessary functions for the anonymisation algorithm through pseudocode. This section will highlight the most important functions, giving thorough explanations.

The following points relate to the core of the anonymisation algorithm:

- Starting with the clustering_kmember(data, k) function (Fig 9). This function had two inputs, the data/ the set of records and a threshold value of k. This was the main function whose main responsibility was to call all the other important functions so that its output can be a set of clusters each containing at least k records. It first starts by retrieving a random record from the data and using that record, calls the first function: get_furthest_record().
- The get_furthest_record(record, data) function (Fig 10) was responsible for determining the furthest record from the random record chosen and this is then made into a cluster.
- The clustering_kmember(data, k) function then calls the find_best_record(cluster, data) (Fig 11). This takes the cluster that we made in the previous function and the dataset. Using those two values, the function then finds a record (for simplicity it will be referred to as r_j) from the dataset that makes the information loss minimal: $IL(cluster \cup \{r_j\})$. This process is looped/ repeated until the absolute value of the cluster is equal to the value of k: $|cluster| = k$. When $|cluster|$ does reach the value of k, we choose a record that is furthest from the random record chosen in the beginning and the clustering process is repeated until there are less than k records left.

- According to the rules of this approach, each cluster needs to have at least k records. When that requirement is fulfilled, we need to allocate the last remaining records to one of the clusters. This is where the `find_best_cluster(record, clusters)` function (Fig 12) comes into play. This function takes one of the remaining records and the clusters as the input. It then allocates the record to a cluster where the information loss would be minimal and this is done by calculating the distances between the records. This process is repeated until all the remaining records have found their cluster homes.

```
def clustering_kmember(data, k=25):  
    """Explanation: this is the structure of the k-anonymisation algorithm,  
    this calls all the necessary functions"""  
    clusters = []  
    rand_seed = random.randrange(len(data))  
    rand_i = data[rand_seed]  
    while len(data) >= k:  
        rand_seed = get_furthest_record(rand_i, data)  
        rand_i = data.pop(rand_seed)  
        cluster = Cluster([rand_i], rand_i)  
        while len(cluster) < k:  
            rand_seed = find_best_record(cluster, data)  
            rand_j = data.pop(rand_seed)  
            cluster.record_addition(rand_j)  
        clusters.append(cluster)  
    while len(data) > 0:  
        t = data.pop()  
        cluster_i = find_best_cluster(t, clusters)  
        clusters[cluster_i].record_addition(t)  
    return clusters
```

Fig 9: clustering_kmember function

```
def get_furthest_record(record, data):  
    """  
    Explanantion: If r is a randomly picked record from S (set of records)  
    This function finds the furthest record from r  
    """  
  
    max_dist = 0  
    max_i = -1  
    for i in range(len(data)):  
        current = distance_in_clusters(record, data[i])  
        if current >= max_dist:  
            max_dist = current  
            max_i = i  
    return max_i
```

Fig 10: get_furthest_record function

```
def find_best_record(cluster, data):
    """
    Explanation: We need to ensure that IL (information loss)
    is minimal therefore we return the record's index with the minimum
    difference on IL
    """
    min_diff = 100000000000
    min_i = 0
    for i, record in enumerate(data):
        Info_loss_diff = diff_dist(record, cluster)
        if Info_loss_diff < min_diff:
            min_diff = Info_loss_diff
            min_i = i
    return min_i
```

Fig 11: find_best_record function

```
def find_best_cluster(record, clusters):
    """
    Explanation: Given a set of clusters and a record, this function
    needs to output a cluster such that information loss is minimal
    """
    min_diff = 100000000000
    min_i = 0
    best = clusters[0]
    for i, c in enumerate(clusters):
        Info_loss_diff = diff_dist(record, c)
        if Info_loss_diff < min_diff:
            min_diff = Info_loss_diff
            min_i = i
            best = c
    return min_i
```

Fig 12: find_best_cluster function

The following point relates to the metrics calculation:

- The metrics used in this system were running time and degree of information loss (NCP). To calculate the running time, it was a simple line of code that measured how long the process took using the Python library, time. The NCP works on based on penalizing records if the generalisation is quite strong. This is because the more generalised the data is, the less useful it becomes due to how generic and non-specific the information is. Fig 13 shows the NCP(record) function.

The following point relate to the most important part of the interface code:

- As the interface is the bridge between the user and the anonymisation algorithm, it has the responsibility of calling and running that code. The function `runCluster()` handles this by opening a new interface window, calling the `clustering_based_k_anon()` function (which calls all the functions spoken about in this section), and presents the output of the anonymisation algorithm. This output contains the metrics for each value of `k` and a graph showing the value of `k` versus the degree of information loss. The function is shown in Fig 14.

```
def NCP(record):  
  
    """  
    Explanation: as mentioned above, this measures the degree of information loss  
    """  
  
    ncp = 0.0  
    list_key = quasi_to_key(record)  
    try:  
        return NCP_cache[list_key]  
    except KeyError:  
        pass  
    for i in range(QI_length):  
        width = 0.0  
        if IS_CAT[i] is False:  
            try:  
                float(record[i])  
            except ValueError:  
                temp = record[i].split(',')  
                width = float(temp[1]) - float(temp[0])  
        else:  
            width = len(A_trees[i][record[i]]) * 1.0  
            width /= QI_range[i]  
            ncp += width  
    NCP_cache[list_key] = ncp  
    return ncp
```

Fig 13: NCP function

```
def runCluster():  
    newWindow = Toplevel(window)  
    newWindow.title("Efficient k-Anon with Clustering: PreLoaded Running")  
    newWindow.geometry("700x700")  
    Label(newWindow, text="Running the clustering based K-Anonymisation algorithm...", font=('Helvetica', 18, 'bold')).pack()  
    Label(newWindow, text="K starts at 2 and increments until 10. A countdown will appear of the k's as the program progresses").pack()  
    data = read_adult()  
    a_trees = read_adult_tree()  
    data_back = copy.deepcopy(data)  
    k=2  
    x = []  
    y = []  
    while (k<11):  
        Label(newWindow, text="k is: %d" %k).pack()  
        result, eval_result = clustering_based_k_anon(a_trees, data, k)  
        write_to_file(result, k)  
        data = copy.deepcopy(data_back)  
        Label(newWindow, text="NCP (degree of information loss): %0.2f" % eval_result[0] + "%").pack()  
        x.append(k)  
        y.append(eval_result[0])  
        Label(newWindow, text="Running time: %0.2f" % eval_result[1] + " seconds").pack()  
        print()  
        k = k+1  
    get_graph(x, y)
```

Fig 14: runCluster function

Problems Encountered

Due to the timeframe of the project, the final version of the code had to be delivered in 12 weeks. The first few weeks were spent completely in research mode, trying to understand the concept of the k-anonymisation approach, seeing what similar projects existed, and finding out what modules and libraries would be needed to transform it into an educational tool. Therefore, there was not as much development time to complete all the aims of the project.

1. This project was meant to be done in a collaborative format where another student would have attached their anonymisation algorithm to the interface so that the educational tool would have more than one option. However, due to the timeframe of the project and difficulties encountered in the other algorithm, putting both projects together was not feasible.
2. Initially, the explanations of the anonymisation algorithm were meant to be dynamic, that is, as the program was running, in real-time, the explanations would pop up on a screen showing the exact processes happening. For example, the explanations would have shown which was the random record chosen and the furthest record from it, etc. Nevertheless, due to the time frame, working out the logistics of making dynamic explanations was not doable. Instead, a static page of brief explanations was included which still gives the user a better understanding of the algorithm.
3. In conjunction with the explanations page and as mentioned in the User Interface section, there was meant to be a section on the page that allowed the user to compare two anonymised datasets of their choice. However, due to a lack of time, the execution of that design was not feasible. Instead, a separate screen called 'Activity' allows the user to view an anonymised text file of their choice. They can open this screen twice to analyse and compare two anonymised text files together.
4. The functionality to allow the user to upload their own datasets is not available. The program was tested using the preloaded dataset and therefore, reading that dataset is hardcoded. There was not enough time to add the flexibility to read a new dataset as it would also require the user to set the quasi-identifiers, and link which hierarchy files correlate to the quasi-identifiers they set. This work is estimated to need one to two weeks of development time which was not possible.

In conclusion, the main aims of the project were able to have some, if not, most of the functionality up and running. However, there were a few features that could not be completed due to the due date of the end-deliverables. Although there was the initial plan where the weeks' work was planned out, it could not and did not account for the time taken to fully understand the problem this project was aiming to solve. In doing background research to get a concrete grasp of the topic area, the majority of the research papers used very domain-specific vocabulary and were very math-forward. This led to having to constantly research and learn about new terms and trying to break down the mathematical formulas in a way that was easy to understand. It should also be noted that part of not fulfilling all the aims could be due to having over-ambitious project aims for the deadline given.

Instructions for Running the Tool

This section provides a brief outline of how to navigate the interface of the educational tool.

1. Run 'interface.py'. With this, the educational tool is presented.
2. From the drop down menu, select 'Efficient K-Anonymisation using Clustering Techniques' and click the 'Submit choice' button.
3. The next screen gives a short summary of the chosen algorithm and now the user is presented with two button options: 'Run with pre-loaded data' or 'Upload file'. Choose the former.
4. The current screen will now give a run-through of the information in the pre-loaded dataset. Click the 'Run Program' button.
5. The anonymisation algorithm now runs and will take a few minutes to complete. When that is done, the metrics screen and a graph of the results will appear. Go back to the screen from step 4 and choose 'Explain Please' button.
6. A more in-depth explanation of the algorithm is displayed. Click 'Open Activity Section'. With this activity screen, you can choose and open any anonymised text file to analyse.

5. Results and Evaluation

Does the system meet requirements?

The end-result system does work as intended in that it can be used as an educational tool for learning about anonymisation algorithms. However, not every goal was achieved that would label the end-result product as ‘complete’. There are a few missing features based on what the initial design outlined and this is discussed in detail.

First and foremost, the educational tool was meant to be equipped with two anonymisation algorithms but as mentioned in the ‘Implementation’ section, this was not viable. The tool only houses the k-anonymisation with clustering technique and can hold more and will be discussed under the ‘Future Work’ section. The purpose of having multiple algorithms in one tool is that it would allow users to analyse how different k-anonymisation approaches work by possibly using the same dataset. Additionally, it would be efficient for the users if all the anonymisation approaches they were investigating and learning about were all in one tool. This would have allowed students to download the tool and add their datasets, or change the code to get hands-on experience in working with these algorithms.

Secondly, did the end-result system meets all the user requirements?

- Requirement 1: *The implementation of the ‘efficient k-anonymisation with clustering’ algorithm produces an anonymised dataset.* The program does produce an anonymised dataset. The program is set to produce an anonymised dataset for every value of k starting at 2 and ending at 10. This means that after running the anonymisation algorithm, the user gets nine anonymised datasets. This was done so that users can see the difference in the way the data looks based on the value of k. Screenshots of the anonymised datasets can be found in the Appendix, Fig 15 and 16 which show the respective datasets for when k is equal to two and k is equal to seven. From the text files, for example when k =2 (Fig 15), it can be seen that the data is clustered into sets of two records showing that the clustering works as stipulated given the rule: clusters must have at least k records. Additionally, the clustered sets look identical in terms of the data shown indicating that the generalisation works as it makes the data more generic; no one individual can be singled out from the anonymised data. Therefore, this requirement has been fulfilled.
- Requirement 2: *The system is able to provide metrics: these will include performance (running time) and measuring information loss.* The system can provide these metrics. The user can run ‘run.py’ if they would like to bypass the interface to receive both the anonymised datasets and metrics (Appendix, Fig 17). Moreover, the user can use the interface to receive the metrics. A metrics window appears after the ‘Run Program’ button is selected (Appendix, Fig 18). It should be noted that due to the processing time of producing nine anonymised datasets, it takes a couple of minutes for the anonymisation algorithm to finish running and to receive the metrics. The metrics presented are running time and a percentage of information loss for each value of k. These metrics enable the user to have numeric and mathematical proof of the effect k-anonymisation has on data utility and see how efficient or inefficient running the algorithm is on different datasets. A graph is also presented for both methods of running the algorithm (Appendix, Fig 19). This graph shows the value of k against the

percentage of information loss giving the user a visual representation of the impact of the value of k on preserving the utility of data. Therefore, this requirement is fulfilled.

- Requirement 3: The system provides the user with an interface that is easy to use- allows for the easy selection of the desired algorithm, all instructions will be easy to follow. The system is equipped with an interface that is easy to use. The interface screens are comprised of buttons and drop-down menus as the only form of user input. This greatly reduces the chances of user error, facilitating the ease of use with this interface. With each screen, any buttons that need clicking are explained or the button title is clear and easy to understand, for example, 'Run Program'. All interface screens are shown in the Appendix, Table 2 along with a description of each screen's function. Unfortunately, there was not enough time to conduct any usability and heuristics testing for the interface but this is discussed in the 'Future Work' section.
- Requirement 4: The system gives the user the option to add their own datasets and hierarchies and will perform the anonymisation on those datasets. Unfortunately, there was not enough time to add the functionality to allow the user to upload their own datasets. This was due to the hardcoded programming of reading in the adult.data dataset for testing purposes. This does reduce the flexibility of the program and will hamper usability as it lessens the choices available to the user. Therefore this requirement was not fulfilled.
- Requirement 5: The system will give explanations via the interface as to how the algorithm works. As mentioned previously, the idea was to have dynamic explanations so with regards to that aspect, the requirement was not fulfilled. However, static explanations have been given painting a clear picture of how the algorithm works. Additionally, an activity screen was added so that users can analyse and compare anonymised datasets. These screens (5 and 6) are in the Appendix, Table 2.

Evaluating Performance of Anonymisation Algorithm

The main goal of this project was not to analyse the efficiencies and inefficiencies of the k -anonymisation algorithm, however, as a huge part of the educational tool deals with showcasing the metrics, it is important to highlight important statistics. The tables below shows the running time and percentage of information loss (IL) for different values of k with differing size datasets.

Dataset size: 200		
Value of k	Running time (s)	Percentage of IL (%)
2	0.13	17.45
3	0.14	29.95
4	0.14	36.02
5	0.14	41.34
6	0.15	44.13
7	0.15	48.92
8	0.16	51.83
9	0.14	53.73
10	0.14	53.99

Table 3: Metrics for dataset of size 200 records

Dataset size: 500		
Value of k	Running time (s)	Percentage of IL (%)
2	0.72	13.08
3	0.81	20.87
4	0.82	26.65
5	0.83	31.43
6	0.79	34.80
7	0.83	39.06
8	0.79	39.92
9	0.82	44.25
10	0.86	45.62

Table 4: Metrics for dataset of size 500 records

Dataset size: 1000		
Value of k	Running time (s)	Percentage of IL (%)
2	2.73	8.65
3	3.02	14.43
4	3.09	18.72
5	3.19	23.20
6	3.25	26.97
7	3.29	29.57
8	3.28	32.94
9	3.31	34.44
10	3.33	37.23

Table 5: Metrics for dataset of size 1000 records

Looking at the tables above, we can infer that with a bigger dataset, the degree of information loss greatly drops. With the dataset of 200, having k at 10 gave an IL of 53.99% whereas with dataset 100, that percentage drops by 16.73%. Another key observation is that with the larger the value k is, the more information loss occurs. This is due to the fact that if there are more records in a cluster, it will lead to greater distortion as all records in a cluster will be generalised to share the same quasi-identifier. The last important information obtained from these results is that the running time increases as the dataset gets larger. With the dataset of 200, the running time for k=10 was 0.14 seconds however, it dataset of 1000, k=10 increased by 3.19 seconds.

Therefore, when considering using this algorithm for any real-world scenario, one would need to deliberate which requirement is more important to have: a quick performing system or anonymising the data. It should be noted that for the majority of the use cases for implementing these anonymisation algorithms, the speed of the program is not a necessity and the datasets are usually much larger than 1000 records. We can conclude that the k-anonymisation with clustering technique approach does indeed satisfy the claims for efficiently protecting data whilst maintaining data utility.

To gain more confidence in this algorithm, implementing other anonymisation approaches and running these same metrics: running time, and degree of IL on them will allow for a clear comparison as to how this k-anonymisation approach holds up to other existing algorithms.

Critical Appraisal of Overall Project

As noted in the previous sections, Python was used to develop the system. It is believed that Python is the most suitable programming language chosen for the type of system [9]. Python is a relatively simple language and due to how complex the problem was and how much time it took to grasp the concept, coding the system in Python was a much quicker process due to its user-friendly language and syntax. Additionally, Python supports object-oriented programming. For this system, the idea is to add more anonymisation algorithms allowing them to share certain classes, such as the generalisation methods and metrics, to be used by other algorithms as well. Hence why there are a cluster class and taxonomy tree classes for the generalisation used in this algorithm. Lastly, Python has a large range of libraries and frameworks. Certainly, for this system, quite a few libraries were used as using existing prewritten code aids in optimising tasks. For example, using the UnitTest library helped in constructing and running tests against the anonymisation code to ensure it was working correctly.

Even so, the one element that would be changed if time allowed it was the GUI library used, Tkinter. Tkinter is one of the most simple GUI libraries in Python due to the minimalism of its syntax. As experience was had with developing with Tkinter previously and due to the time constraints of the project, Tkinter was chosen to develop the interface. However, to create a more sophisticated and dynamic interface, another GUI framework such as Kivy, would be more appropriate.

In terms of methodology, a more agile approach was taken. Every week, a meeting was had with the supervisor who would advise on which feature should be developed in time for the next meeting. During the development week, programming and testing were done for the desired feature and presented to the supervisor at the next meeting for feedback. In this case, the supervisor acted as the user as this was their project proposal. It is believed that this methodology worked well for this project as there was feedback at every feature development as well as it allowed for flexibility and changing of plans.

In conclusion, the choices made for developing this system were quite appropriate. Given the time constraints and giving a grace period for understanding the concept of the problem of the project proposed, the interface used was the best choice. If work was to be continued on this project, possibly swapping the GUI framework used would make the educational tool of a higher standard.

6. Future Work

After approximately working on this project for 12 weeks, the deadline has approached and not all the aims planned for this project were achieved. Due to time constraints and underestimating the complexity of the problem proposed, there was not enough time to develop all the features that were in the initial plan. However, this type of project-the educational tool- the very nature of this system needs to be constantly updated and developed. There will always be the need to add more information to the existing approaches in the tool or enhance the tool with more anonymisation approaches as research in this field is ongoing.

The following list comprises the features that were either in the initial plan or labelled as '*could have*' features (elements that arose during project meetings that would have been nice to have in the system but not necessary):

- Initially, this tool was meant to be equipped with two anonymisation approaches, to begin with. Due to complications from the other project handling the second anonymisation approach, this could not be done. Nevertheless, as an educational tool, it should contain multiple anonymisation approaches for lecturers and students to explore and learn.
- The ability for the user to input their datasets- This will allow users to have multiple datasets at their disposal to see how differently the algorithm treats each one. In so doing, users would get a better idea of which types of data (numeric, categorical, continuous, etc) work best with different anonymisation approaches. Even more so, users can compare metrics of the different approaches to see which results in the least amount of information loss.
- Dynamic explanations- The tool can be updated to produce explanations as the anonymisation algorithms are running. Users would then get real-time information, and data snippets of the work happening 'behind the scenes'. It might result in users gaining more understanding of the algorithms.
- Usability testing- this was not allocated time in the initial plan as it was not feasible to conduct heuristics experiments/ interviews in the given timeframe. However, it is important to receive user feedback on interfaces to ensure it is user-friendly and easy to navigate from those not involved in the development of the system. As well, with the information obtained from the usability testing, a more sophisticated interface can be achieved.

In conclusion, the educational tool made by this project does have the ability to benefit many students in learning about anonymisation algorithms, and assisting lecturers in teaching the topic by providing visuals. However, to truly receive the benefits of this system, more development work needs to be done to ensure the end-users are getting the most out of it.

7. Conclusions

The main aim of this project was to implement an existing anonymisation approach, the ‘K-Anonymisation using Clustering Technique’, using Python, and create an educational tool from it. This tool is meant to be used by lecturers and students to provide and encourage a more visual and hands-on experience in learning about anonymisation algorithms and approaches. From the proposed and initial plan, the tool was meant to: produce an anonymised text file from either a pre-loaded dataset or a user-uploaded dataset, provide metrics to evaluate the performance of the anonymisation algorithm, have an attached user interface to allow users to easily interact with the algorithm, and offer explanations to describe the processes occurring in the algorithm.

A solid attempt was made at implementing all the desired features of the educational tool, however, due to time constraints, it was not possible. The features that were able to be developed were as follows: a minimalist interface was created to allow easy navigation of the system, anonymised datasets are produced but only from the pre-loaded dataset, metrics and a graph of running time and degree of information loss is presented after the anonymisation algorithm has been run, and static explanations are provided, coupled with the ability of the user to analyse an anonymised text file of their choosing. Therefore, it can be said that the core of the proposed solution has been successfully built.

More than creating the educational tool, this project also evaluated the anonymisation algorithm and drew a few conclusions from the results. Based on the results recorded, the k-anonymisation using clustering technique seems to work best with larger datasets. By ‘work best’, it means that the percentage of information loss significantly drops regardless of the value of k, although, a smaller k value assists in keeping the information loss percentage down. There is one consideration that users will need to keep in mind with this approach and that is the matter of running time. As the dataset grows, so does running time. Though, as the majority of these anonymisation approaches are to modify data in an acceptable manner to be published, running time may not be that big of an issue.

Nevertheless, with the very nature of this system, it is meant to be constantly updated to reflect the theories and progress made within this field of study. There are a few areas that will benefit the tool with future work such as: adding more anonymisation approaches which will widen the choices the user can make in selecting an algorithm to learn about, adding the ability to allow the user to upload their datasets to be anonymised, conduct usability testing and interviews to ensure the tool is as user-friendly as it can be, and include dynamic explanations so that users can get real-time descriptions of the events and tasks being done to the data.

In conclusion, all the ‘must-have’ features for the educational tool have been developed in this project. This was aided by developing it in an agile methodology which facilitated receiving feedback from the ‘user’ and working on implementing those features for the next meeting. The ‘user’ in this case was the project supervisor who proposed the project, therefore, their feedback was valuable in ensuring the end-result system was as they intended it to be. But even with creating the system in an agile environment and producing features every week for confirmation from the user, the project scope was too big for the time given to

deliver the end deliverable (the educational tool). Thus, there are quite a few elements that need to be added to the tool but this project achieved the fundamental and essential features, and analysed and evaluated the k-anonymisation algorithm with success.

8. Reflection on Learning

There was much to be learnt throughout this dissertation process. Not only in terms of growing my programmer skillset but also in learning important life lessons that will be useful in progressing my career after university as well as in life in general.

Firstly, with respect to hard skills, that is, the technical knowledge gained, there was the mastery of Python. Prior to this project, I had only done a semester of Python programming. However, as the entirety of creating this educational tool was done in Python, it provided me with the opportunity to fine-tune my Python skills. Additionally, I had to explore Tkinter, the GUI toolkit used for the interface creation, on a much deeper level than I had before, thus expanding my programming skillset.

With respect to reflecting on the approach to the problem and any and all decisions made, there are a few areas that stand out. There was the issue of the shortage of time to complete this project and that meant that I had to make decisions as to what features or tests would be omitted in order to still produce a project I was not only content with but ensuring that the project achieved the majority of the initial aims set out. I do believe that the right decisions were made in choosing which elements were to be included up to this point of the development life cycle of this tool as they were considered important ‘must-haves’ for the user. This project also forced me to take a realistic viewpoint as I had to understand that there was the possibility that not every aim that I planned out would make it into the final system. Therefore, I had to allow being kind and give myself grace when the coding had to cease and not all the elements were included. This was one of the most invaluable lessons I had to learn in this process as being a perfectionist can sometimes become a handicap in being productive.

Another important area to note is the lack of knowledge on this topic area. As a student, this was my first step into the world of anonymisation algorithms. Therefore, the first two to three weeks of this project I spent thoroughly researching and reading journal articles as well as seeking out the advice of my supervisor. My supervisor and I worked as a team which was beneficial as they were able to clear up any misconceptions or confusions I had surrounding any part of my algorithm. As well, several of the scientific journals in this topic area are very math-forward and use unfamiliar jargon, and in talking with my supervisor, I was then able to start grasping the concepts as they broke them down into more digestible and easy to understand phrases. As a result, I have written my report in such a way that does not focus too much on the mathematical aspects but more so in a way that anyone new to the topic area can appreciate the anonymisation approach without too much difficulty. I am quite proud of the way I overcame this knowledge barrier by not being afraid to seek help which is often a problematic task for me. However, this allowed me in seeing that asking for help can aid in progressing work and thus, results in less time wasted.

The third area in which this project impacted me was time management. This project had a strict deadline of 12 weeks where, in this case, an educational tool needed to be working along with a report detailing the entire process of creating the tool. Time had to be allocated not only for this project but for the other module that was occurring weekly as well as catering for leisure time to ensure a positive mental health environment was present. Meetings with the supervisor were held on a weekly basis to make certain that progress was

being done on the development of the tool but for each week I made sure that I wrote notes so that report writing would be a much simpler process in the last few weeks before hand-in. In so doing, I was able to guarantee that I would have a running tool with the majority of the elements implemented as well as creating a lower-stress environment for the report writing process. This approach to time management will help me in the future when it comes to maintaining a healthy work-life balance and safeguarding that I deliver all my work tasks before deadlines have passed.

The fourth and last area that I will reflect on is gaining the ability to be critical of my work. The report for this project required a critical appraisal of the system, meaning that one would need to highlight the good parts but also, bring light on the areas that could have been approached differently to turn out better. As a programmer, you would want to make sure your system is the best it can be, but in order to achieve that, you would also need to have the ability to notice the areas for improvement.

As this project draws to a close, I can say that I did enjoy the process. There were frustrations of course but in all, both the positives and negatives can only help in developing myself as a software developer. I have encountered problems that will no doubt arise again in the future but, because of this experience, I would have the knowledge and capability to better combat those problems.

Table of Abbreviations

ABBREVIATION	MEANING
IL	Information Loss
NCP	Normalised Certainty Penalty
QID	Quasi-Identifier

Glossary

TERM	DEFINITION
ANONYMISATION	Process of removing any specific information from the data that would allow attackers to identify individuals. The purpose of this is to allow the individuals whom the data describe to remain nameless.
CLUSTERING	Process of grouping data points into a number of groupings/ units where the data points in each group are similar to each other.
DATA PRIVACY	Safeguarding of personal data/ any identifiable information from those who do not have access to such material.
DATA PUBLISHING	The release of collected and research data to the public to be used by others- individuals, companies and governments.
DATA UTILITY	This refers to how valuable the data is to businesses, entities and governments within particular usage situations.
END-USER	The person whom the system and/or product is intended for.
FRAMEWORK	Platform or conceptualisation as to how to develop a software application. Provides a guide for the developer to follow.
GENERALISATION	Process of making specific information more abstract. In this case, making the data less identifiable for protection of identity.
HEURISTICS	Technique outlined to provide developers with a guide for designing efficient solutions.
HIERARCHY	A system in which items are ranked and/or the process of grouping records into levels.

INFORMATION LOSS	Reduction in the value of data and information therefore, the best decisions cannot be made based on the data provided.
K-ANONYMISATION	Process of hiding the identities of individuals in datasets by simplifying the data so that the information in the dataset could be compatible to any individual.
METRICS	A system of measures for assessment purposes which are particularly useful for comparisons and monitoring performance.
NORMALISED CERTAINTY PENALTY	A metric for monitoring the degree of information loss in anonymisation algorithms. It punishes records based on how they have been generalised.
PROTOTYPE	Model of how a proposed system or product should work. It can show the necessary features that are required, how the data flows, and how the product/system should look.
PSEUDOCODE	Description not using a programming language syntax, to describe the steps and tasks the algorithm/ system should perform. This is useful to developers in designing their code.
QUASI-IDENTIFIER	Pieces of information that separately, do not necessarily reveal the identity of individuals but when pieced together, can reveal an individual's identity.
USABILITY TESTING	Process of assessing a system and/or product by presenting it to representative users. This allows for developers to receive feedback as to the areas of the system that users find more difficult to navigate and use.
USER INTERFACE	The space where human-computer interaction occurs. This is how the system receives any user input it needs in order to complete tasks.

Appendix



```
anonymized_k_is_2 - Notepad
File Edit Format View Help
30,60;State-gov;Divorced;Sales;*;Female;*;<=50K
30,60;State-gov;Divorced;Sales;*;Female;*;<=50K
17,33;Private;Never-married;Farming-fishing;White;Male;United-States;<=50K
17,33;Private;Never-married;Farming-fishing;White;Male;United-States;<=50K
39,55;gov;Married-civ-spouse;Sales;Asian;Female;*;<=50K
39,55;gov;Married-civ-spouse;Sales;Asian;Female;*;<=50K
24,90;Private;Never-married;Other-service;Black;Male;United-States;<=50K
24,90;Private;Never-married;Other-service;Black;Male;United-States;<=50K
20,30;*;*;Sales;Asian;Female;Thailand;<=50K
20,30;*;*;Sales;Asian;Female;Thailand;<=50K
73,81;Self-emp-not-inc;Married-civ-spouse;Exec-managerial;White;Male;United-States;<=50K
73,81;Self-emp-not-inc;Married-civ-spouse;Exec-managerial;White;Male;United-States;<=50K
29,33;*;Never-married;*;Asian;Female;Philippines;<=50K
29,33;*;Never-married;*;Asian;Female;Philippines;<=50K
```

Fig 15: Screenshot of the anonymised text file when $k = 2$. This only shows the first 14 rows.



```
anonymized_k_is_7 - Notepad
File Edit Format View Help
30,60;*;Divorced;Sales;*;Female;*;<=50K
30,60;*;Divorced;Sales;*;Female;*;<=50K
30,60;*;Divorced;Sales;*;Female;*;<=50K
30,60;*;Divorced;Sales;*;Female;*;<=50K
30,60;*;Divorced;Sales;*;Female;*;<=50K
30,60;*;Divorced;Sales;*;Female;*;<=50K
30,60;*;Divorced;Sales;*;Female;*;<=50K
17,60;non-gov;Never-married;Farming-fishing;White;Male;United-States;<=50K
17,60;non-gov;Never-married;Farming-fishing;White;Male;United-States;<=50K
17,60;non-gov;Never-married;Farming-fishing;White;Male;United-States;<=50K
17,60;non-gov;Never-married;Farming-fishing;White;Male;United-States;<=50K
17,60;non-gov;Never-married;Farming-fishing;White;Male;United-States;<=50K
17,60;non-gov;Never-married;Farming-fishing;White;Male;United-States;<=50K
17,60;non-gov;Never-married;Farming-fishing;White;Male;United-States;<=50K
```

Fig 16: Screenshot of the anonymised text file when $k = 7$. This only shows the first 14 rows.

```

K=2
NCP (degree of information loss): 8.83%
Running time: 2.73 seconds

K=3
NCP (degree of information loss): 14.51%
Running time: 2.84 seconds

K=4
NCP (degree of information loss): 19.32%
Running time: 3.00 seconds

K=5
NCP (degree of information loss): 23.06%
Running time: 3.07 seconds

K=6
NCP (degree of information loss): 25.92%
Running time: 3.14 seconds

K=7
NCP (degree of information loss): 29.90%
Running time: 3.23 seconds

K=8
NCP (degree of information loss): 31.72%
Running time: 3.42 seconds

K=9
NCP (degree of information loss): 34.44%
Running time: 3.25 seconds

K=10
NCP (degree of information loss): 37.23%
Running time: 3.33 seconds

```

Fig 17: Screenshot of the metrics when run.py is run

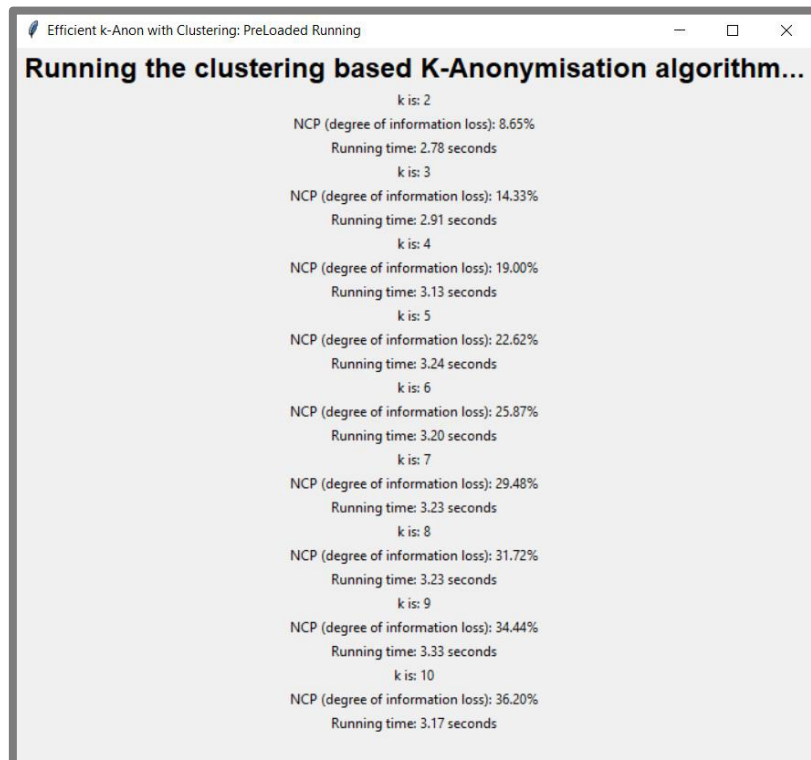


Fig 18: Screenshot of the metrics via the interface

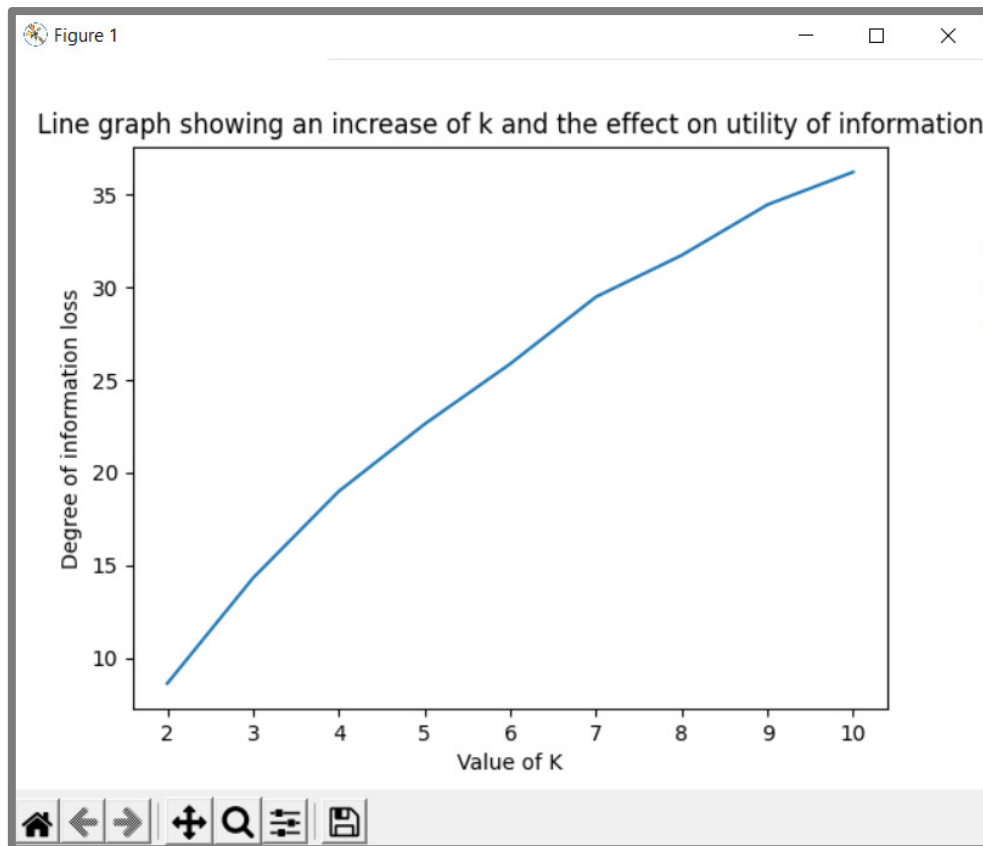
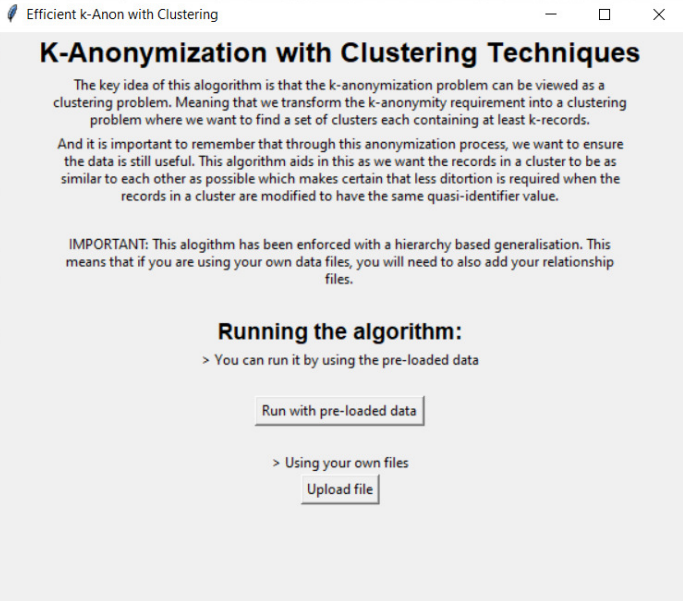
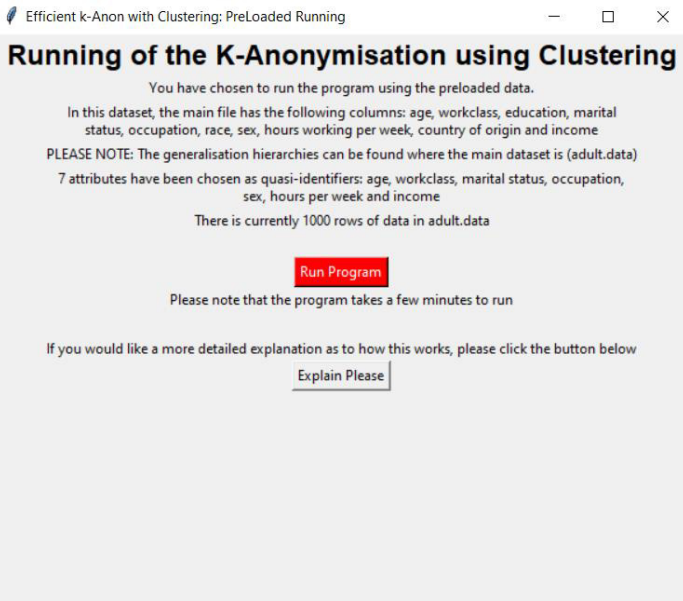
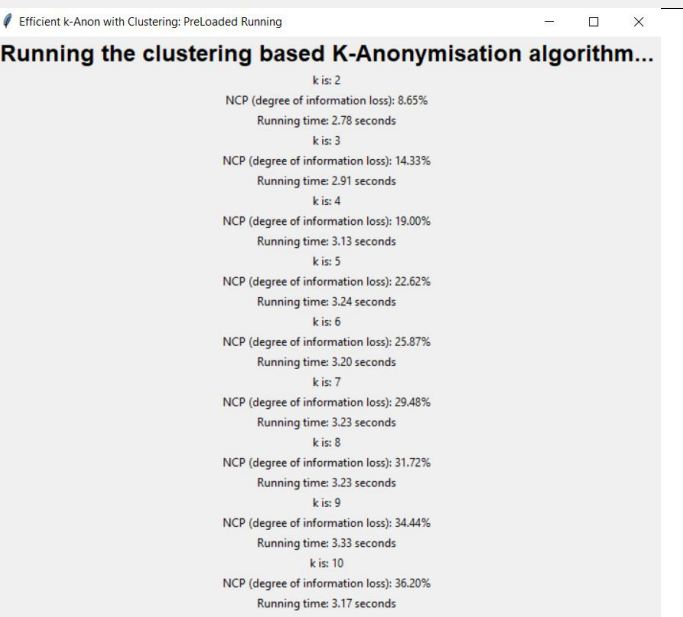


Fig 19: Screenshot of the graph produced when the program is run

Screen Number	Function	Interface Screen
1	This is the introductory screen that explains the use of the tool and presents the options of the anonymisation approaches that can be used through a drop down menu.	<p>Anonymization Algorithms</p> <h3>Welcome to Anonymisation solutions</h3> <p>We are all aware that everyday data is being collected from multiple entities: governments, corporations, accepting cookies! This is done to harvest more information about customers to get to know them better; to make better decisions for consumers. However, some data that is collected may contain highly sensitive information.</p> <p>Therefore, in order to protect the privacy of individuals, different methods and tools have been created to ensure that data remains useful whilst preserving individual privacy</p> <p>Here we are showing two different solutions. Please click one to find out more about it:</p> <p>Efficient K-Anonymization Using Clustering Techniques</p> <p>Submit choice</p>

2	After choosing the k-anonymisation using clustering technique, the screen presented gives a brief explanation about this particular algorithm and presents the option for the user to run with preloaded data or their own datasets.	 <p>K-Anonymization with Clustering Techniques</p> <p>The key idea of this algorithm is that the k-anonymization problem can be viewed as a clustering problem. Meaning that we transform the k-anonymity requirement into a clustering problem where we want to find a set of clusters each containing at least k-records.</p> <p>And it is important to remember that through this anonymization process, we want to ensure the data is still useful. This algorithm aids in this as we want the records in a cluster to be as similar to each other as possible which makes certain that less distortion is required when the records in a cluster are modified to have the same quasi-identifier value.</p> <p>IMPORTANT: This algorithm has been enforced with a hierarchy based generalisation. This means that if you are using your own data files, you will need to also add your relationship files.</p> <p>Running the algorithm:</p> <p>> You can run it by using the pre-loaded data</p> <p>Run with pre-loaded data</p> <p>> Using your own files</p> <p>Upload file</p>
3	After confirming the choice of the preloaded dataset, the interface explains to the user the contents of the dataset. From there the user can choose to run the program or get explanations.	 <p>Running of the K-Anonymisation using Clustering</p> <p>You have chosen to run the program using the preloaded data.</p> <p>In this dataset, the main file has the following columns: age, workclass, education, marital status, occupation, race, sex, hours working per week, country of origin and income</p> <p>PLEASE NOTE: The generalisation hierarchies can be found where the main dataset is (adult.data)</p> <p>7 attributes have been chosen as quasi-identifiers: age, workclass, marital status, occupation, sex, hours per week and income</p> <p>There is currently 1000 rows of data in adult.data</p> <p>Run Program</p> <p>Please note that the program takes a few minutes to run</p> <p>If you would like a more detailed explanation as to how this works, please click the button below</p> <p>Explain Please</p>
4	Metrics screen	 <p>Running the clustering based K-Anonymisation algorithm...</p> <p>k is: 2 NCP (degree of information loss): 8.65% Running time: 2.78 seconds</p> <p>k is: 3 NCP (degree of information loss): 14.33% Running time: 2.91 seconds</p> <p>k is: 4 NCP (degree of information loss): 19.00% Running time: 3.13 seconds</p> <p>k is: 5 NCP (degree of information loss): 22.62% Running time: 3.24 seconds</p> <p>k is: 6 NCP (degree of information loss): 25.87% Running time: 3.20 seconds</p> <p>k is: 7 NCP (degree of information loss): 29.48% Running time: 3.23 seconds</p> <p>k is: 8 NCP (degree of information loss): 31.72% Running time: 3.23 seconds</p> <p>k is: 9 NCP (degree of information loss): 34.44% Running time: 3.33 seconds</p> <p>k is: 10 NCP (degree of information loss): 36.20% Running time: 3.17 seconds</p>

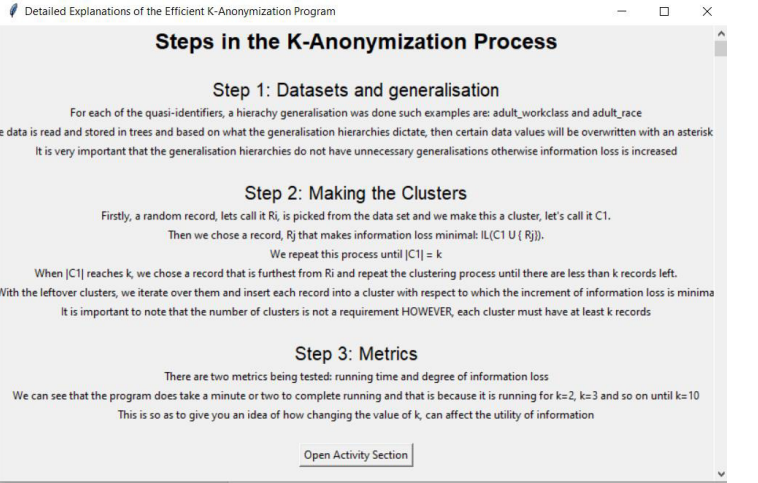
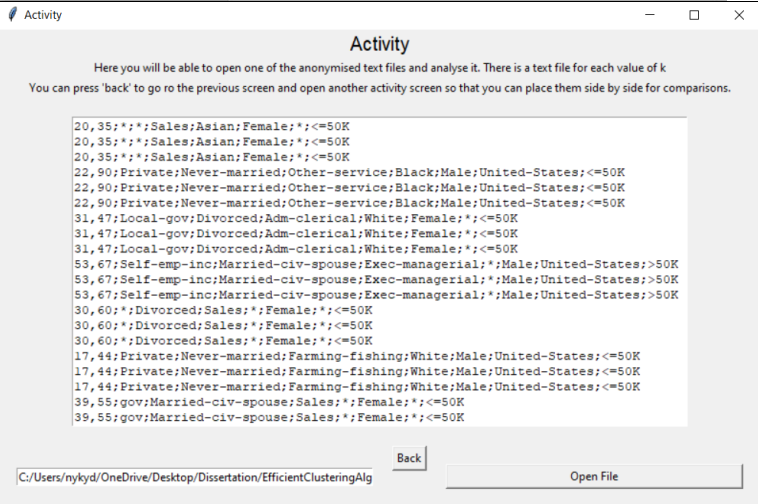
5	<p>This is the explanation screen when the ‘explain please’ button is selected. It explains the basic areas of getting the anonymisation algorithm to work.</p>	 <p>Detailed Explanations of the Efficient K-Anonymization Program</p> <h3>Steps in the K-Anonymization Process</h3> <p>Step 1: Datasets and generalisation For each of the quasi-identifiers, a hierarchy generalisation was done such examples are: adult_workclass and adult_race e data is read and stored in trees and based on what the generalisation hierarchies dictate, then certain data values will be overwritten with an asterisk It is very important that the generalisation hierarchies do not have unnecessary generalisations otherwise information loss is increased</p> <p>Step 2: Making the Clusters Firstly, a random record, lets call it R_i, is picked from the data set and we make this a cluster, let's call it C1. Then we chose a record, R_j that makes information loss minimal: $IL(C1 \cup \{R_j\})$. We repeat this process until $C1 = k$ When $C1$ reaches k, we chose a record that is furthest from R_i and repeat the clustering process until there are less than k records left. With the leftover clusters, we iterate over them and insert each record into a cluster with respect to which the increment of information loss is minimal It is important to note that the number of clusters is not a requirement HOWEVER, each cluster must have at least k records</p> <p>Step 3: Metrics There are two metrics being tested: running time and degree of information loss We can see that the program does take a minute or two to complete running and that is because it is running for $k=2$, $k=3$ and so on until $k=10$ This is so as to give you an idea of how changing the value of k, can affect the utility of information</p> <p>Open Activity Section</p>
6	<p>If the ‘Open Activity Section’ is clicked on screen 5, the screen allows the user to open any of the anonymised text files to be analysed</p>	 <p>Activity</p> <p>Here you will be able to open one of the anonymised text files and analyse it. There is a text file for each value of k You can press 'back' to go to the previous screen and open another activity screen so that you can place them side by side for comparisons.</p> <pre> 20, 35, *, *, Sales;Asian;Female; *, <=50K 20, 35, *, *, Sales;Asian;Female; *, <=50K 20, 35, *, *, Sales;Asian;Female; *, <=50K 22, 90;Private;Never-married;Other-service;Black;Male;United-States;<=50K 22, 90;Private;Never-married;Other-service;Black;Male;United-States;<=50K 22, 90;Private;Never-married;Other-service;Black;Male;United-States;<=50K 31, 47;Local-gov;Divorced;Adm-clerical;White;Female; *, <=50K 31, 47;Local-gov;Divorced;Adm-clerical;White;Female; *, <=50K 31, 47;Local-gov;Divorced;Adm-clerical;White;Female; *, <=50K 53, 67;Self-emp-inc;Married-civ-spouse;Exec-managerial; *, Male;United-States;>50K 53, 67;Self-emp-inc;Married-civ-spouse;Exec-managerial; *, Male;United-States;>50K 53, 67;Self-emp-inc;Married-civ-spouse;Exec-managerial; *, Male;United-States;>50K 30, 60; *, Divorced;Sales; *, Female; *, <=50K 30, 60; *, Divorced;Sales; *, Female; *, <=50K 30, 60; *, Divorced;Sales; *, Female; *, <=50K 17, 44;Private;Never-married;Farming-fishing;White;Male;United-States;<=50K 17, 44;Private;Never-married;Farming-fishing;White;Male;United-States;<=50K 17, 44;Private;Never-married;Farming-fishing;White;Male;United-States;<=50K 39, 55;gov;Married-civ-spouse;Sales; *, Female; *, <=50K 39, 55;gov;Married-civ-spouse;Sales; *, Female; *, <=50K </pre> <p>Back Open File</p> <p>C:/Users/nykyd/OneDrive/Desktop/Dissertation/EfficientClusteringAlg</p>

Table 2: Table explaining the functionality behind every interface screen

References

1. Innovative Advertising. (2018). Importance of Data Collection & How to Use It. [online] Available at: <https://innovativeadagency.com/blog/importance-data-collection/#:~:text=Why%20is%20Data%20Collection%20so.>
2. Fung, B.C.M., Wang, K., Chen, R. and Yu, P.S. (2010). Privacy-preserving data publishing. *ACM Computing Surveys*, 42(4), pp.1–53.
3. Loukides, G. and Shao, J.-H. (2008). An Efficient Clustering Algorithm for k-Anonymisation. *Journal of Computer Science and Technology*, [online] 23(2), pp.188–202. Available at: <https://dl.acm.org/citation.cfm?id=1993435> [Accessed 5 Aug. 2019].
4. Google Cloud Blog. (n.d.). Take charge of your data: Understanding re-identification risk and quasi-identifiers with Cloud DLP. [online] Available at: <https://cloud.google.com/blog/products/identity-security/taking-charge-of-your-data-understanding-re-identification-risk-and-quasi-identifiers-with-cloud-dlp>.
5. reader.elsevier.com. (n.d.). Elsevier Enhanced Reader. [online] Available at: <https://reader.elsevier.com/reader/sd/pii/S0167404821003126?token=8A822E7CD9D29B8BA9CA0474A52476AA95C82EEC785204C55FF58382F595625A6DBB2374474556CA4BBE1FA73BC42DD6&originRegion=eu-west-1&originCreation=20220503162704> [Accessed 3 May 2022].
6. Narula, D., Kumar, P. and Upadhyaya, S. (2016). Data Utility Metrics for k-anonymization Algorithms. *International Journal of Scientific & Engineering Research*, [online] 7. Available at: <https://www.ijser.org/researchpaper/Data-Utility-Metrics-for-k-anonymization-Algorithms.pdf> [Accessed 5 May 2022]
7. Loukides, G. and Gkoulalas-Divanis, A. (2012). Utility-preserving transaction data anonymization with low information loss. *Expert Systems with Applications*, 39(10), pp.9764–9777. doi:10.1016/j.eswa.2012.02.179.
8. Uci.edu. (2019). UCI Machine Learning Repository: Adult Data Set. [online] Available at: <https://archive.ics.uci.edu/ml/datasets/Adult>.
9. Edoxi Training. (n.d.). Top Advantages of Python Over Other Programming Languages. [online] Available at: <https://www.edoxitraining.com/studyhub-detail/advantages-of-python-over-other-programming-languages>.