KOÇ UNIVERSITY

COMP 430/530: Data Privacy and Security – Fall 2024

Homework Assignment #1

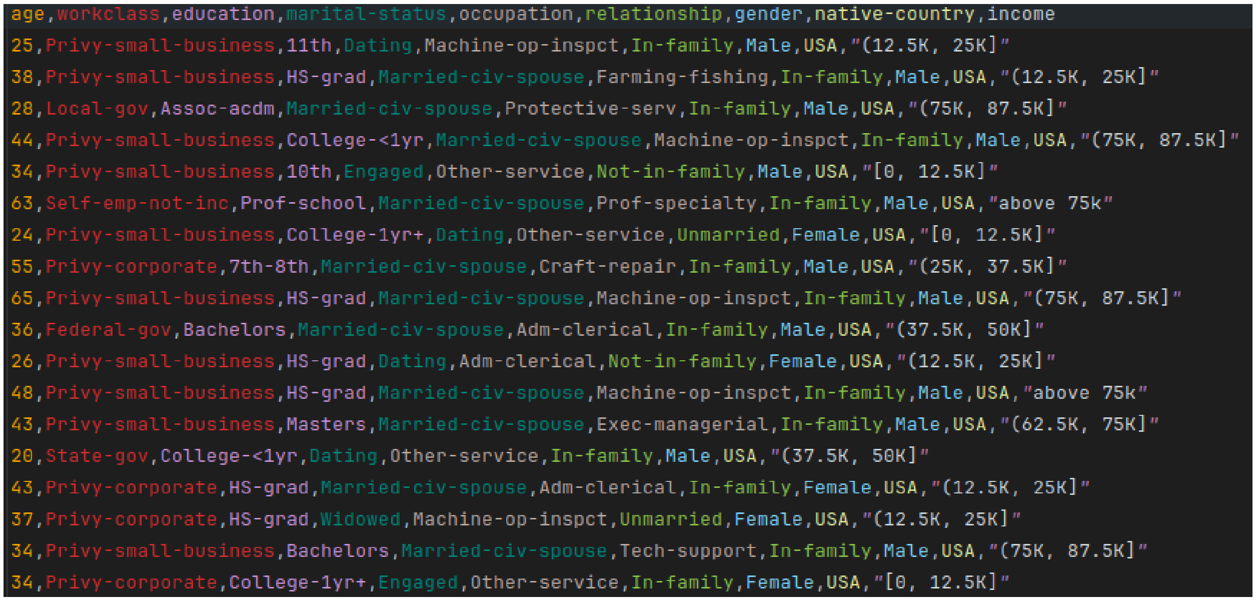
INTRODUCTION AND SETUP

In this assignment, you will implement anonymization algorithms in Python and compare their performance by executing your algorithms on a real dataset. You should use the skeleton Python file that we are providing as your starting point. (Submit **.py** only, we do not accept **.ipynb** extension.)

We also provided you a modified version of the **Adult** dataset, which is a commonly used dataset in the literature (original version is here: http://archive.ics.uci.edu/ml/datasets/Adult, but you do not need the original version, you will be working with the version we provided).

Each row of the dataset corresponds to one individual. Each column (attribute) contains information such as the individual's gender, education level, marital status, etc. The 'income' column represents the salary of individuals and contains 8 distinct salary ranges. Throughout this assignment, you will treat every attribute other than ***income*** as a Quasi Identifier; and treat ***income*** as the Sensitive Attribute.

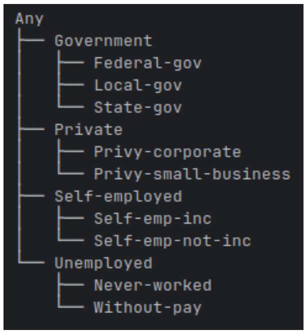
Here is a screenshot from the **Adult** dataset that is provided to you. Notice that the dataset is in **csv** (comma separated values) format and the first row contains attribute names.



```
age,workclass,education,marital-status,occupation,relationship,gender,native-country,income
25,Privy-small-business,11th,Dating,Machine-op-inspct,In-family,Male,USA,"(12.5K, 25K]"
38,Privy-small-business,HS-grad,Married-civ-spouse,Farming-fishing,In-family,Male,USA,"(12.5K, 25K]"
28,Local-gov,Assoc-acdm,Married-civ-spouse,Protective-serv,In-family,Male,USA,"(75K, 87.5K]"
44,Privy-small-business,College-<1yr,Married-civ-spouse,Machine-op-inspct,In-family,Male,USA,"(75K, 87.5K]"
34,Privy-small-business,10th,Engaged,Other-service,Not-in-family,Male,USA,"[0, 12.5K]"
63,Self-emp-not-inc,Prof-school,Married-civ-spouse,Prof-specialty,In-family,Male,USA,"above 75k"
24,Privy-small-business,College-1yr+,Dating,Other-service,Unmarried,Female,USA,"[0, 12.5K]"
55,Privy-corporate,7th-8th,Married-civ-spouse,Craft-repair,In-family,Male,USA,"(25K, 37.5K]"
65,Privy-small-business,HS-grad,Married-civ-spouse,Machine-op-inspct,In-family,Male,USA,"(75K, 87.5K]"
36,Federal-gov,Bachelors,Married-civ-spouse,Adm-clerical,In-family,Male,USA,"(37.5K, 50K]"
26,Privy-small-business,HS-grad,Dating,Adm-clerical,Not-in-family,Female,USA,"(12.5K, 25K]"
48,Privy-small-business,HS-grad,Married-civ-spouse,Machine-op-inspct,In-family,Male,USA,"above 75k"
43,Privy-small-business,Masters,Married-civ-spouse,Exec-managerial,In-family,Male,USA,"(62.5K, 75K]"
20,State-gov,College-<1yr,Dating,Other-service,In-family,Male,USA,"(37.5K, 50K]"
43,Privy-corporate,HS-grad,Married-civ-spouse,Adm-clerical,In-family,Female,USA,"(12.5K, 25K]"
37,Privy-corporate,HS-grad,Widowed,Machine-op-inspct,Unmarried,Female,USA,"(12.5K, 25K]"
34,Privy-small-business,Bachelors,Married-civ-spouse,Tech-support,In-family,Male,USA,"(75K, 87.5K]"
34,Privy-corporate,College-1yr+,Engaged,Other-service,In-family,Female,USA,"[0, 12.5K]"
```

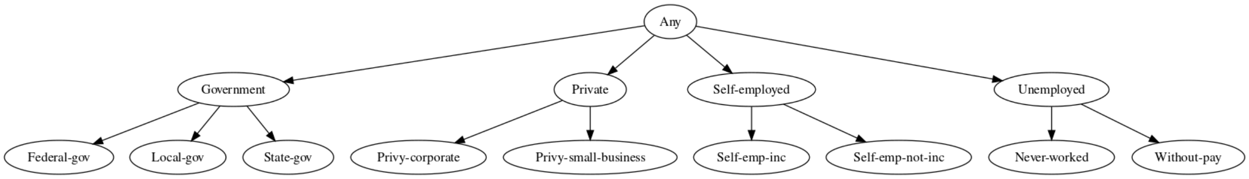
In addition to the **Adult** dataset, you are also given a folder named **DGHs**. Each file in the DGHs folder contains the domain generalization hierarchy of one of the QI attributes, e.g., **age.txt** contains the DGH for the **age** attribute, **education.txt** contains the DGH for the **education** attribute, and so forth.

When you study the contents of the DGH files, you will see that they follow a pattern such that the tabs indicate parent-child relationships in a DGH. For example, here is how **workclass.txt** represents the visual DGH of the **workclass** attribute:



```
Any
├── Government
│   ├── Federal-gov
│   ├── Local-gov
│   └── State-gov
├── Private
│   ├── Privy-corporate
│   └── Privy-small-business
├── Self-employed
│   ├── Self-emp-inc
│   └── Self-emp-not-inc
└── Unemployed
    ├── Never-worked
    └── Without-pay
```

This **workclass.txt** can be visualized as a tree shown below:



Your first goal should be to read datasets and DGHs into your program's memory. It is important to note that the **Adult** dataset and its DGHs are provided to you as samples, to help you in development and testing. Your code should not be specific to these inputs – **it should work for any dataset and any set of DGHs.** Hence, do not hardcode any paths, filenames, DGHs, etc. We may test your code with arbitrary datasets and DGHs, and your code should still work.

You can assume that the following conditions will always hold:

- The dataset will always be a csv file and its first row will contain the names of attributes.
- For a dataset containing N columns, 1 of those columns will be the SA and the rest will be QIs.
- All DGHs will be given to you in one folder containing multiple files (one file for each DGH).
- All necessary DGHs will be present and all DGHs will be complete (i.e., no need to check if there is a missing DGH file or if the DGH covers all possible values for an attribute – it does).

The homework assignment consists of several parts. In each part, we give you the names and parameters of the functions you need to implement. **Do not modify function names or parameters (e.g., do not add or remove parameters)!** When grading, we will call exactly these functions with different inputs – if you modify function names or parameters, we won't be able to call your functions the way we are expecting to, and you will lose points.

In addition to the functions you are asked to implement, you are welcome to implement as many additional or helper functions as you like. Try to make your code modular, organized and easy-to-follow. This may help us in giving partial credit.

As you go through the assignment, you will find out that there are multiple ways to implement a certain operation. We strongly recommend that you take into account **efficiency** (both time-efficiency and memory-efficiency). An inefficient implementation can take several hours on a small dataset, whereas an efficient implementation can finish in a few seconds.

GOOD LUCK!

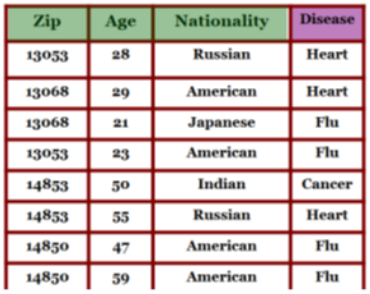
<u>PART 1: ANONYMIZATION COSTS</u> (20 pts)

In the lectures, we covered two metrics to measure costs of anonymization: Distortion Metric (MD) and Loss Metric (LM). In this part, you will implement these metrics.

cost_MD (*raw_dataset_file*, *anonymized_dataset_file*, *DGH_folder*)

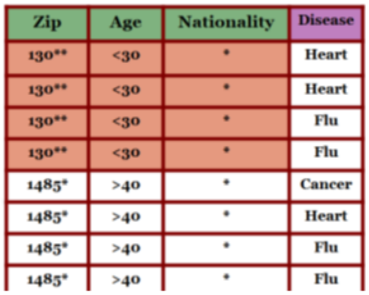
- *raw_dataset_file* is a string containing the path of the raw dataset file, e.g., *“adult-hw1.csv”.*
- *anonymized_dataset_file* is the path of the anonymized dataset, e.g., *“adult-anonymized.csv”.*
- *DGH_folder* is the directory containing all DGH files.

This function should read the raw dataset and anonymized dataset from the corresponding files and calculate the cost of anonymization using the Distortion Metric. You can assume that the order of the rows and columns are the same in the raw and anonymized datasets. That is: (i) both datasets contain the same attributes in the same order, (ii) N’th row in the anonymized dataset is the generalized version of the N’th row in the raw dataset. For example:



Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Flu
13053	23	American	Flu
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Flu
14850	59	American	Flu

Raw dataset



Zip	Age	Nationality	Disease
130**	<30	*	Heart
130**	<30	*	Heart
130**	<30	*	Flu
130**	<30	*	Flu
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Flu
1485*	>40	*	Flu

Anonymized dataset

In this example, it also happens to be that consecutive records belong to the same equivalence class. This need not always be the case. For example, in the files given to you, rows 1-3-8 could constitute one equivalence class, rows 2-4-5 could constitute another equivalence class, etc.

The return value of the **cost_MD** function should be the MD cost.

cost_LM(*raw_dataset_file*, *anonymized_dataset_file*, *DGH_folder*):

This function follows the same parameters and assumptions as **cost_MD**. But instead of calculating MD cost, it calculates and returns the Loss Metric (LM) cost. When calculating LM cost, assume that the weights of all attributes in the dataset are identical, i.e., for a dataset with M quasi-identifier attributes, $w_1 = w_2 = \ldots = w_M = 1/M$.

PART 2: RANDOM ANONYMIZER (20 pts)

In this part, you will implement a randomized algorithm for k-anonymizing a dataset. The algorithm works as follows:

1. Given dataset ***D*** and k-anonymity parameter ***k***, the algorithm randomly shuffles the dataset to randomize the order of the rows.
2. The first k records are put into EC1, the next k records are put into EC2, the next k records are put into EC3, … and so forth.
    a. If the size of the dataset is a perfect multiple of ***k***, you'll have exactly ***|D|/k*** ECs.
    b. Otherwise, the very last EC will contain between ***k+1*** and ***2k-1*** records, so that all records can end up belonging to an EC that contains at least ***k*** records.
3. For each EC, make sure that QI-wise equality is achieved through generalizations. While doing so, perform the minimum amount of generalization necessary to achieve k-anonymity for that EC. **No redundant generalizations or over-generalizations!**
4. Finally, the impact of the shuffling performed in step #1 is reversed, so that the Nth row in the anonymized dataset will be the generalized version of the Nth row in the raw dataset when we write the anonymized dataset to an output file.

In the skeleton code provided to you, steps 1 and 4 are already taken care of. Do not modify these parts. We did this so that when we call your function with a fixed seed (for the random shuffler), we'll get the same results each time. This makes grading and debugging easier on our end. If you modify these parts, your output may not match what we are expecting, and you may lose points.

Your task is to add the code necessary to perform steps 2 and 3.

random_anonymizer (*raw_dataset_file*, *DGH_folder, k, s, output_file*):

- *raw_dataset_file* is a string containing the path of the raw dataset file, e.g., *"adult-hw1.csv".*
- *DGH_folder* is the directory containing all DGH files.
- *k* is the k-anonymity parameter.
- *s* is the seed for the random shuffler. You can call your function with different seeds to observe different results.
- The function has no return value, but it should write the anonymized dataset to a file. The name of this file is passed in a parameter called *output_file*, e.g., *output_file* = *"anon.csv"*. Nth row in the anonymized dataset will be the generalized version of the Nth row in the raw dataset.

PART 3: CLUSTERING-BASED ANONYMIZER (25 pts)

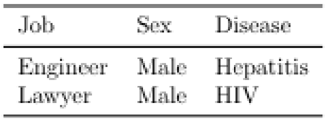
Now, let us implement a more intelligent k-anonymization algorithm based on the idea of clustering similar records together. So far we have seen 2 metrics that can calculate the distance between two records: LM cost and MD cost. Since they both measure distance from different points of view, to make a more robust distance metric, we will use both of them together.

We define a distance metric based on LM cost **LM_dist(r1,r2)** between two records r1 and r2 as the **LM cost of hypothetically placing those two records in one equivalence class (EC) with the minimum amount of generalization necessary**. When calculating the LM cost, assume that the weights of all attributes are identical, i.e., for a dataset with M quasi-identifier attributes, $w_1 = w_2 = \ldots = w_M = 1/M$. Note that the range of LM_dist is between 0 and 1.

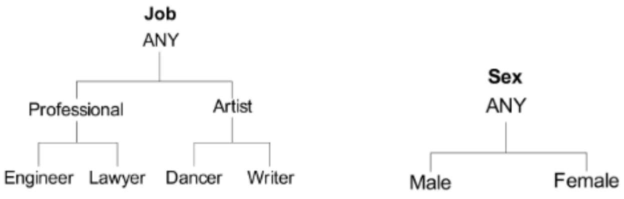
Similarly, we define a distance metric based on MD cost **MD_dist(r1,r2)** between two records as the **MD cost of hypothetically placing those two records in one equivalence class (EC) with minimum amount of generalization necessary**. However, the range of MD_dist is different from LM_dist. Therefore, we are going to normalize MD_dist by dividing it by the maximum possible MD dist. The total distance **dist(r1, r2)** is then calculated by:

dist(r1, r2) = LM_dist(r1, r2) + normalized_MD_dist(r1, r2)

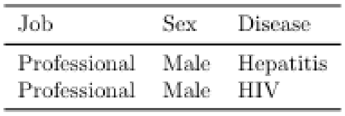
Let us give an example of calculating this distance. Consider the following two records on the left-hand side. Assume Job and Sex are the QIs, Disease is the sensitive attribute:



Job	Sex	Disease
Engineer	Male	Hepatitis
Lawyer	Male	HIV



The minimum amount of generalization to put these two records in one EC is:



Job	Sex	Disease
Professional	Male	Hepatitis
Professional	Male	HIV

Then: LM_dist(r1,r2) = 1/6 because LM(Professional) = 1/3, LM(Male) = 0

MD_dist(r1,r2) = 2 because Engineer->Professional in two records

Maximum possible MD dist = 6 because Engineer->Any in two records (2*2)

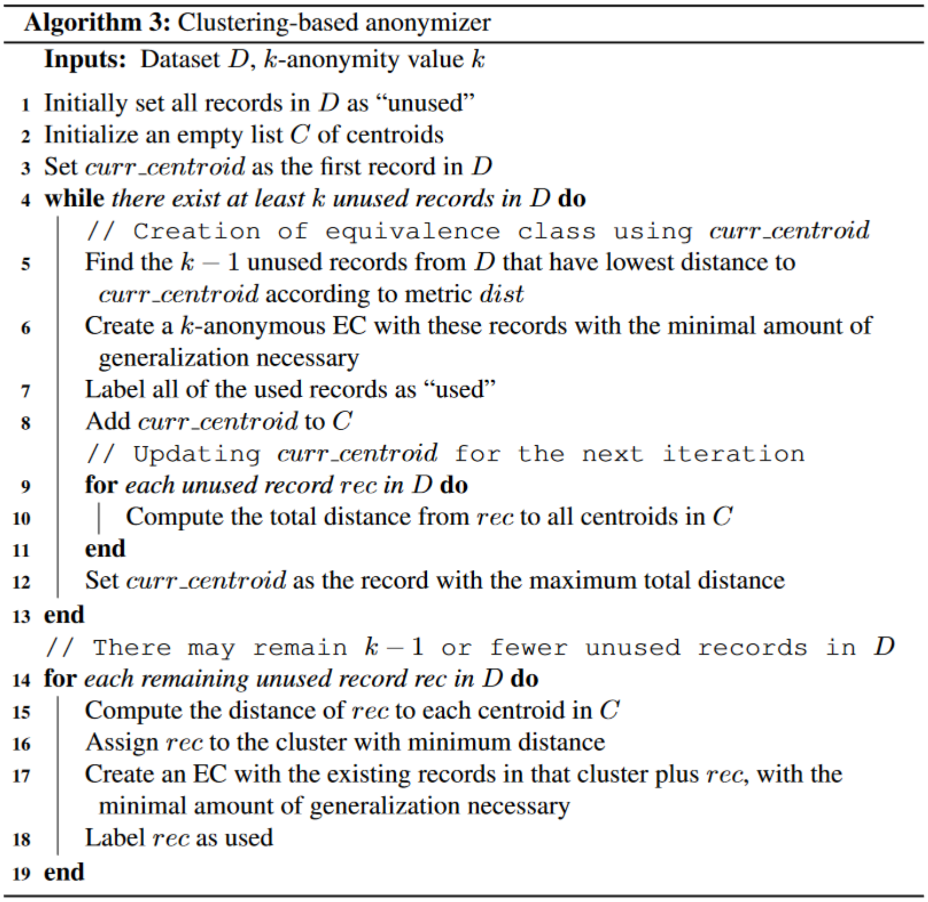
+ Male->Any in two records (1*2)

normalized_MD_dist(r1,r2) = 2/6

dist(r1, r2) = LM_dist(r1, r2) + normalized_MD_dist(r1, r2) = 1/6 + 2/6 = 1/2

[Hint: For these computations, you may want to re-use some code that you wrote for the cost_LM and cost_MD functions in Part1.]

Given dataset ***D*** and parameter ***k***, your clustering-based anonymizer should work as follows:



```
Algorithm 3: Clustering-based anonymizer
   Inputs: Dataset $D$, $k$-anonymity value $k$
1  Initially set all records in $D$ as "unused"
2  Initialize an empty list $C$ of centroids
3  Set $curr\_centroid$ as the first record in $D$
4  while there exist at least $k$ unused records in $D$ do
       // Creation of equivalence class using $curr\_centroid$
5      Find the $k-1$ unused records from $D$ that have lowest distance to
         $curr\_centroid$ according to metric $dist$
6      Create a $k$-anonymous EC with these records with the minimal amount of
         generalization necessary
7      Label all of the used records as "used"
8      Add $curr\_centroid$ to $C$
       // Updating $curr\_centroid$ for the next iteration
9      for each unused record rec in $D$ do
10         Compute the total distance from $rec$ to all centroids in $C$
11     end
12     Set $curr\_centroid$ as the record with the maximum total distance
13 end
   // There may remain $k-1$ or fewer unused records in $D$
14 for each remaining unused record rec in $D$ do
15     Compute the distance of $rec$ to each centroid in $C$
16     Assign $rec$ to the cluster with minimum distance
17     Create an EC with the existing records in that cluster plus $rec$, with the
         minimal amount of generalization necessary
18     Label $rec$ as used
19 end
```

Some notes and clarifications:

- The terminology "used" and "unused" records indicate whether that record was previously used in the construction of an EC. If a record has previously been used to construct one EC, we label it as "used" so that it is not re-used in a later iteration in another EC.
- The purpose of lines 14-18 is to handle datasets with cardinality other than perfect multiple of **k**.
- In some cases (e.g., lines 5, 12, 16), you may need to perform tie-breaking when there are multiple records or clusters with the exact same distance. In such cases, always prefer the record/cluster with the lowest index.

The signature of the function you need to implement is:

clustering_anonymizer *(raw_dataset_file, DGH_folder, k, output_file)*

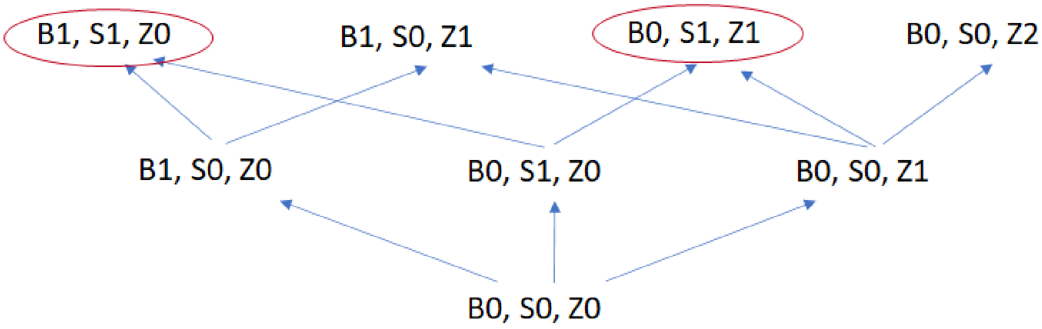
The parameters have the same meaning as in **random_anonymizer**. Similar to random_anonymizer, clustering_anonymizer does not have a return value. Instead, it should write the anonymized dataset to *output_file*. N'th row in the anonymized dataset should be the generalized version of the N'th row in the raw dataset.

PART 4: BOTTOM-UP ANONYMIZER (25 pts)

Recall the bottom-up anonymization approach from the lectures. In this part, you will implement a simplified variant of this approach that satisfies k-anonymity and distinct l-diversity together:

1. Initially, set the current level as the bottom-most level of the generalization lattice.
2. Explore all nodes in the current level.
    a. If one or more nodes satisfy k-anonymity and distinct l-diversity for the given k and l, pick the one with lowest LM cost and terminate.
    b. If no node satisfies both k-anonymity and distinct l-diversity together, continue with the next level of the generalization lattice (one level up).

As an example, consider the scenario below. We start with the bottom-most level and find that B0,S0,Z0 does not satisfy k-anonymity and distinct l-diversity together. Then, we move to the next level and explore the three nodes: B1,S0,Z0 and B0,S1,Z0 and B0,S0,Z1. Say that none of them satisfy k-anonymity and distinct l-diversity. We move to the next level and find four nodes: B1,S1,Z0 and B1,S0,Z1 and B0,S1,Z1 and B0,S0,Z2. Among them, two nodes satisfy k-anonymity and distinct l-diversity: B1,S1,Z0 and B0,S1,Z1. We calculate the LM cost for both of these nodes and pick the one which has lower LM cost.



Hint: Before jumping into code, think about how you can implement this easily and effectively. Considering that this is a simplified variant of the bottom-up approach, you may not necessarily need to organize nodes in a lattice structure with links, etc. What matters is the correctness of your end result.

The signature of the function you need to implement is:

bottomup_anonymizer *(raw_dataset_file, DGH_folder, k, l, output_file)*

The parameters have the same meaning as the previous two parts, with the addition of l for l-diversity. Write the anonymized dataset to *output_file*. N'th row in the anonymized dataset should be the generalized version of the N'th row in the raw dataset.

PART 5: MINI REPORT (10 pts)

Submit a **max one-page report (hard limit!)** containing your experiments and analysis of the different anonymization algorithms you implemented in Parts 2, 3 and 4. Your report should be in PDF format.

Impact of k in k-anonymity

Run the 3 anonymizers on the Adult dataset for different values of k: k = 4, 8, 16, 32, 64, 128, 256. For the bottom-up anonymizer, fix l = 1. In each run, measure the anonymizer's time cost (how long does it take to execute), LM cost, and MD cost.

- For the random anonymizer, we recommend that you repeat each experiment a few times using different seeds and average the results.
- For the clustering-based anonymizer, we recommend that you shuffle the data (do it before calling clustering_anonymizer) to achieve randomization. You can repeat each experiment a few times using random shuffling and average the results.

Impact of l in l-diversity

Recall that l-diversity is used only in the bottom-up anonymizer. Fix k = 16, and run the bottom-up anonymizer with different values of l: l = 1, 2, 3, 4, 5, 6, 7, 8. In each run, measure the anonymizer's time cost (how long does it take to execute), LM cost, and MD cost.

Contents of your report

Report your experiment results in tables and/or graphs: LM cost vs k, LM cost vs l, time vs k, time vs l, MD cost vs k, MD cost vs l. You can choose to draw them as tables or graphs, whichever you think is better.

Briefly discuss your observations and take-away messages. For example, what trade-offs do you observe? Which anonymizer is fastest? Which one has the lowest utility loss? Which anonymizer would you prefer under what settings? Do the results fit your expectations?

When grading your report, we will pay attention to how you constructed your tables/graphs, their readability and quality, as well as the quality of your analysis and discussion.

SUBMISSION

When you are finished, submit your assignment via LearnHub:

- Move all of your Python files and your report into a folder named **`your KUNet ID`.**
- **Compress this folder into a single zip file**. (Don't use compression methods other than zip.)
- Upload your zip file to LearnHub.

Notes and reminders:

- After submitting, download your submission and double-check that: (i) your files are not corrupted, (ii) your submission contains all the files you intended to submit, including all of your source code and your report. If we cannot run your code because some of your code files are missing, we cannot give you credit!
- You must upload your code in py files**, we do not accept Python notebooks (.ipynb extension)**.
- This homework is an **individual assignment**. All work needs to be your own. Submissions will be checked for plagiarism (including comparing to previous years' assignments).
- **Your report should be a PDF file.** Do not submit Word files or others which may only be opened on Windows or Mac (or opening them may remove table/figure formatting).
- Only LearnHub submissions are allowed. Do not email your assignment to the instructor or TAs.
- **Do not submit any data files** (such as the Adult dataset, DGHs, or your anonymized datasets).
- Do not change the names or parameters of the functions that we will grade.
- If your code does not run (e.g., syntax errors) or takes an extremely long amount of time (e.g., it takes multiple hours whereas our reference implementation takes 1-2 minutes), you may get 0 for the corresponding part.