Brandon Palonis

Dr. Garrison

CS1699

4/20/2019

Project 3

**W0:** The dataset I chose was one of Pokémon, obtainable from <https://www.kaggle.com/rounakbanik/pokemon> . This dataset contains information on all 802 Pokémon that currently exist, including their typing, all six of their stats and the total, and their possible abilities. To make this dataset more akin to the kind that are necessary for this project, I had to remove several columns from the original dataset that would not work for a public dataset. This was done to treat each Pokémon like a person and make this dataset innocuous enough to be released, but not so innocuous as to not be able to be further anonymized for this project. I removed columns containing: a Pokémon’s name in English, a Pokémon’s Pokédex number, A Pokémon’s name in Japanese, and a Pokémon’s classification. My assumption is that if this dataset were to be publicly released by a company, these would be direct identifiers as to the occupants of the dataset. A Pokémon’s English and Japanese name would not be released in a public dataset, just as a person’s name would not be released in a public dataset. The Pokédex number is also unique to each Pokémon, so it can be immediately linked to a Pokémon and therefore reveals too much information to resemble a publicly available dataset. Lastly, while there is some overlap on Pokémon classification, for instance there are four Pokémon classified as the Seed Pokémon, but there is only one Pokémon classified as the Moonlight Pokémon. Due to this, I felt classification would still be too direct of an identifier for some Pokémon, and therefore it was removed from this dataset. The removal of these fields does not remove utility from the set, there is still useful information to be gained by what is left, this data can still be used to analyze Pokémon as a whole. For instance, this still contains information on the typing of Pokémon and their stats, and one could use this dataset to determine the percentage of fire type Pokémon that have less than 50 base speed. The original dataset is included here as pokemon\_clean.csv, and the dataset with removed fields used for anonymization is pokemon.csv.

**W1:** In its current form, this dataset leaks potentially damaging information on the Pokémon contained within. All Pokémon have 6 total statistics associated with them, hit points (HP), attack, defense, special attack, special defense, and speed. If all 6 stats appeared in the table as they do now, all one would have to do is search for a Pokémon with the exact spread of stats to link their identity. For instance, Bulbasaur’s stats are 45, 49, 49, 65, 65, 45 for hp, attack, defense, special attack, special defense, and speed respectively. Bulbasaur is the only Pokémon with these exact numbers in those exact stats, so the statistics being in the dataset is potentially problematic. All an attacker has to do is cross reference all six stats of a row in the database and they can find the exact Pokémon to which those stats pair.

**C2:** To transform the dataset to some value of *k* at your discretion, run **anonymizer.py** and enter a value of *k*. The quasi-identifiers I chose for this were a Pokémon’s 6 stats as described above: HP, attack, defense, special attack, special defense, and speed. **anonymizer.py** generalizes *k* values for each stat and hides each stat among several.

**anonymizer.py outputs the anonymized dataset as pokeanon.csv, the original dataset used for anonymization is pokemon.csv**

**W3:** When comparing the original dataset used for anonymization, **pokemon.csv,** and the anonymized set, **pokeanon.csv**, less information is revealed in the anonymization process. Looking at Bulbasaur as example, in the original table his stats are listed as their exact numbers:

pokemon.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HP | Attack | Defense | Sp\_Attack | Sp\_Defense | Speed |
| 45 | 49 | 49 | 65 | 65 | 45 |

Here are the same values for a 4-anonymity set contained in pokeanon.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HP | Attack | Defense | Sp\_Attack | Sp\_Defense | Speed |
| [45, 60, 80, 39] | [49, 62, 100, 52] | [49, 63, 123, 43] | [65, 80, 122, 60] | [65, 80, 120, 50] | [45, 60, 80, 65] |

Bulbasaur’s stats are now generalized among the values of the stats for other Pokémon, which make finding Bulbasaur from afar much more difficult. The only information that may be revealed about a Pokémon are its abilities. Every Pokémon has a list of one or more abilities they have, and these abilities activate in battle in some way, i.e. “Run Away” is an ability that, if the Pokémon who has it is out during battle, allows the trainer to leave the fight without fail. All Pokémon have abilities, but no Pokémon have an ability so unique that it identifies them immediately, Bulbasaur’s abilities alone are shared by 3 Pokémon. All abilities in the csv file are still information being given away, however. The utility of this dataset does lessen slightly as queries cannot be run on a Pokémon’s stats, but with the values still left, i.e. Abilities, stat totals, and typing are innocuous fields that don’t identify a Pokémon and can still be queried.

**C4 & C6:** Run **load\_database.py** and type in 1 when prompted to perform Task C4, type in 2 to perform Task C6. Pokeano.csv is loaded into this program and then put into a database to make querying easier.

**W5:** The insight used for this task was the count of all Pokémon who are primarily fire type, with a secondary fighting type whose stat total was greater than 300. This returns 6, the total number of Pokémon who satisfy the above query. This does not satisfy differential privacy because if a neighboring dataset changes by one of the following: if a Pokémon is added that satisfies the query where the previous row did not, if a Pokémon is removed that satisfied the query when the previous row did, then the count would change, and an attacker would know whether it is from our dataset or from the neighboring dataset. If a row was swapped and the previous row satisfied the query or not, then that is not detectable.

**W7:**  The insight used for this task was the count of all Pokémon that are primarily water type and are legendary. This method uses Laplacian sampling to create noise that masks the metric given by the query. This satisfies differential privacy because in any given neighboring dataset where a single Pokémon’s attributes have been changed is hidden by the Laplacian noise. We know that they might be in the table, but there is no guarantee that that actually are.