Universität Leipzig
Institut für Informatik lecturer: Dr. Victor Christen
Abteilung Datenbanken Deadline: 05.06.2024

Data Preparation & Cleaning

Summer semester 2024

—

Practical 1: Blocking

Preliminaries

To complete the tasks, you should check out the corresponding git repository `https://git.informatik.uni-leipzig.de/dbs/dpc_practicals` This contains the code skeleton and data sets for using the record linkage system. You should integrate the modules and data sets into your Python project.

Before implementation, you should familiarize yourself with the individual modules and the structure for the various tasks. This exercise only covers completing the modules within the `blocking` package. For a more detailed understanding, you can run `recordLinkage.py`. It uses the provided data records and already implemented functions. This makes the output of the different modules comprehensible. To test the program for correctness, it is recommended to use the data records without impurities, the `clean-A-1000.csv` and `clean-B-1000.csv`. The other data records can then be used.

Task 1: Soundex-Blocking (3 points)

Implement the Soundex phonetic encoding function, and use Soundex values as the blocking key values. In other words, all values with the same Soundex code should be placed into the same block. Note: You should follow the algorithm as described in the lecture.

(a) (1 point) Using your Python Soundex function, generate the Soundex codes for the following name strings: *christina*, *kirstyn*, *allyson*, *alisen*.

(b) (1 point) On the attribute(s) you think are suitable for blocking, run the simpleBlocking and your Soundex based blocking function, and for both write down the number of blocks generated, as well as their minimum, average, and maximum sizes, when you apply blocking on the clean-A-1000.csv and little-dirty-A-1000.csv data sets.

(c) (1 point) Describe your results in a few sentences, and also explain why you selected certain attribute(s) for blocking.

For question 1 you will receive 1 point if all four encodings are correct while if not all or none are correct we will assess your submitted Python code and award points depending upon what you have implemented. For questions 2 and 3 you will receive 1 point each for reasonable answers and appropriate explanations of your choice of attributes for blocking.

Task 2: SLK-581 (2 points)

The Statistical Linkage Key SLK-581 is an identifier that can be used to identify records that belong to the same person if they have the same SLK-581. As shown in the figure, SLK-581 is made up of four elements, including three letters from family name (surname or last name), two letters from given name (first name), date of birth, and gender.

In the blocking.py module, implement SLK-581 as a blocking key where all records with the same SLK-581 identifier should be inserted into the same block. We cover SLK-581 in more detail in the chapter "Record linkage - Blocking and Indexing".

(a) (1 point) As we asked in the previous task on Soundex based blocking, write down the number of blocks generated, as well as their minimum, average, and maximum sizes, when you apply SLK-581 as a blocking function on the clean-A-1000.csv and little-dirty-A-1000.csv data sets.

(b) (1 point) SLK-581 was developed in Australia for modern Australian health databases. Discuss what aspects could be problematic with SLK-581 when used in Germany, and how could SLK-581 be modified so it works better in Germany?

Task 3: Automatic Blocking Key Selection (3 points)

The selection of blocking keys is an essential and non-trivial step. The Fisher score can be used to select blocking key candidates for a disjunctive blocking plan. Complete the functions `compute_fisher_score` and the necessary `generate_feature_vectors` Evaluate your completed method by executing the selection method `select_blocking_keys` with all attributes (1-11) and the blocking functions Soundex and SimpleBlocking as blocking key candidates. Use 200 as the training size for the number of matches.

(a) (2 points) What influence do the parameters ϵ and the maximum block size ratio have on the distribution of the block size and the number of blocks? Test different configurations for the little dirty data set with 1000 and 10000 records by using the `disjunctive_block` function.

(b) (1 point) What is the difference to the conjunctive blocking scheme regarding the number of blocks, average/maximum/minimum number of elements per block?

Task 4: Performance Assessment (2 points)

Evaluate the implemented blocking methods from tasks 1, 2 and 3 for the clean and slightly dirty data with 1000 and 10000 records respectively. Describe your results. Evaluate the results in terms of the number of blocks, the distribution of block sizes (minimum, maximum, average and median) and the runtimes for the different data sets.

In Moodle, please enter the numerical results in the first text field and the description and explanation of the results in the second text field. Please address the following points:

- Which blocking functions and attributes would you use and why?
- Which attributes are not suitable for blocking in isolation, but lead to an improvement in combination?
- Define a set of criteria that a good blocking key or a set of blocking keys should fulfill. Justify your answer using the results of your experiments.

You receive 1 point per answer, whereby the completeness of the numerical values is relevant for the first part and appropriate reasons and explanations for the second part.