Report of Name Filtering Task

Objective

Objective is to prevent the occurrence of duplicate names within the relational data.

Work Flow

The data get loaded first. Then it was grouped by the similar columns. Then its undergoes through 3 functions. Filter_records_1, 2, 3. In this functions there are 3 matching algoritms were used. They are:

1. Token Set Ratio:

- The fuzz.token_set_ratio() function from RapidFuzz library computes the similarity between two strings based on the unique tokens (words) present in both strings.
- It tokenizes the input strings, removes duplicate tokens, and calculates the similarity ratio based on the intersection of tokens between the two strings.
- Higher ratios indicate greater similarity, considering the unordered nature of tokens. This metric is suitable for comparing strings with minor variations or reordering of words.
- In the program, token set ratio is used in the first filtering step (filter_records_1) to compare names and retain the most similar ones within each group.

2. Soundex:

- The jellyfish.soundex() function from the Jellyfish library computes the Soundex code for a given string, representing its phonetic pronunciation.
- Soundex is a phonetic algorithm that encodes strings into a small set of representative characters based on their pronunciation, facilitating approximate string matching.
- Strings with similar pronunciations have the same or similar Soundex codes, enabling efficient matching of names despite variations in spelling or pronunciation.
- In the program, Soundex is used in the second filtering step (filter_records_2_with_soundex) to compare names based on their phonetic representations and retain records with similar Soundex codes within each group.

3. Partial Ratio:

- The fuzz.partial_ratio() function from RapidFuzz library calculates the similarity between two strings based on the longest common substring.
- It considers partial matches between strings, allowing for comparisons where one string is a subset or contains a significant portion of the other.
- Partial ratio is useful for identifying similar strings even when they are not exact matches or have minor differences.
- In the program, partial ratio is used in the third filtering step (filter_records_3) to compare names and retain records with significant partial matches within each group.

In all function there were some thresholds is given. If the matching ratio is less than that they were taken as distinct records. If not the one with more characters is selected.
In first function the exactly similar names were filter out. In second one and third one the input is the names that are not exactly similar. The similarity between them is finded by using sound matching and partial matching.

Overall, by combining these string similarity metrics, the program effectively filters and retains the most relevant records within each group of the family tree dataset, ensuring accuracy and consistency in the data.

Demerits

The program does not effectively distinguish between nearly identical names of brothers, sisters, wives, or similar types of relationships, which can be multiple within a family tree dataset.

For example, a person, let's call them A, can have brothers named Vipin and Bipin. However, the program may not accurately identify and differentiate between these similar names, potentially leading to incorrect filtering or retention of records.