Executive Summary

A dataset composed of 250,000 rows of provider data was examined. The purpose of the analysis was to determine which providers were unique and create a Provider Entity Id to group together all like providers. The threshold for an approximate match was 75; while this is not statistically significant it allowed for a margin of error to account for the number of uncleaned fields. The specific field of interest was Service Provider Name.

Methodology

The approach utilized a python library Fuzzy Wuzzy. Fuzzy Wuzzy is a string matching library that easily compares the distance of sequences using the Levenshtein Distance formula.

Further requirements of the analysis was heavy data cleaning to produce a master list of unique providers. Because of the volume of dirty data specific manual steps were taken to reduce the number of repeated provider entries, this included removing specific prefixes such as MD, DO, CHRIO, etc. It is important to note this step was not taken into account for the filtered list, but other minor data cleaning techniques were employed such as removing unneeded white space, blank fields in the Service Provider Name column.

Executing the fuzzy wuzzy package on the master list and filtered list produces a probability score of the two string comparisons matching.

Hypothesis

The expectation with reasonable data wrangling will result in the unique list matching the non unique data list at a certain threshold (in this case 75), however the threshold can be adjusted in other cases to gain a higher confidence.

Challenges

* Excessive duplicated entries.
* Missing values in multiple columns.
* Defining master list where ambiguity exists.
* Computationally expensive.

Results

Unique Service Provider Name and Provider Entity Id matched to filtered list. This solution could be used to group all un scrubbed data into a unique group or it could be used to put as a flag for a current cleaning model. E.g. probability score > 75, etc.

Further consideration

* Whereas this solution works, heavy data cleaning via python, R, and/or SQL would be needed to adapt this for production.
* More time would allow further scrubbing for a master list.

In order to fully productionalize this process three things would need to be considered:

1. Scalability. Pyspark should be explored depending on the volume of data. E.g. 10 gigs, 1 terabyte, etc.
2. Processing time of Fuzzy Wuzzy. For instance, the master list maybe 200,000 unique providers but the list to match against maybe much larger. Many more combinations of sequences and difference in sequences may make the process run longer.
   * Currently the MVP (Minimal Viable Product) I have developed uses approximately 6000 unique providers for a master list and 75000 for a comparable list. This was due to run times of greater than 2 hours if it was increased.
3. Data cleaning for the master list.