Documentation

It is a tricky situation when you have 2 data sets and want to find duplicates in both but the problem is that the rows are not in same positions. And looping on the entire datasets will be time taking.

The method I have used for the solution of Finding duplicates problem is given below:

Tis method uses Dedupe python package to solve the problem. The library provides required functions and techniques. The technique Used is called regularized logistic regression. If we supply pairs of records that we label as either being duplicates or distinct, then Dedupe will learn a set of weights such that the record distance can easily be transformed into our best estimate of the probability that a pair of records are duplicates. Once we have learned these good weights, we want to use them to find which records are duplicates.

In order to learn those weights, Dedupe needs example pairs with labels. Most of the time, we will need people to supply those labels. But the whole point of Dedupe is to save people’s time, and that includes making good use of your labeling time so they use an approach called Active Learning. Dedupe picks, at random from this disagreement set, a pair of records and asks the user to decide. Once it gets this label, it relearns the weights and blocking rules. We then recalculate the disagreement set.

The algorithm uses hierarchical clustering with centroid linkage gave us the best results. What this algorithm does is say that all points within some distance of centroid are part of the same group. In this way the duplicates are part of one cluster.

The end result is a dataset showing cluster ID and Scores of similarity. You can sort on Cluster ID and see the duplicate rows in cluster of same ID.