Machine Learning methods:

1. “SVM”: Support Vector Machines
   * Largely used for regression analysis and classification models, so this will be a good candidate for Record Linkage
   * For supervised approach: The training data set should include two definitive categories from which the model can differentiate (so for Record Linkage, we would want records labeled as “links” and “nonlinks” after being put through the emClassify or epiClassify functions and those labeled as
   * For unsupervised approach: Also possible, the algorithm just identifies natural clustering of data
   * Pretty good explanation of the methodology behind this approach in R, though this article discusses it in the context of a different package. <https://www.r-bloggers.com/machine-learning-using-support-vector-machines/>
2. “Rpart”: Recursive Partitioning
   * Essentially follows the decision tree format where records are split into sub-populations based on the values of their variables.
   * A number of research papers and past studies have used rpart as the method for record deduplication, so this seems to also be a promising option.
   * Essentially this algorithm works by creating an overall classification tree or hierarchical structure from the training dataset. A really simple basic example of the first couple nodes (to my understanding of how this model works) is below

nonlink

DOB doesn’t Match

All Pairs

nonlink

SSN doesn’t match

DOB Matches

SSN matches

1. "ada" Stochastic boosting model:
   * Synthesizes multiple “weak” learning models in order to limit correlation in results and optimize residual loss functions
   * A “weak” learning model is one that is only slightly better than random chance
   * Ada boosting models specifically use a decision tree with a single split. It then weighs observations and puts more weight on difficult to classify instances.
   * Training data: definitive link and non-link data
   * Great website that goes into more depth: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/
2. "Bagging" Bagging with classification trees (i.e. booststrap aggregating)
   * Fits multiple large trees to bootstrap-resampled (aka random sampling with replacement) versions of the training data. Then it chooses the best model by “majority vote” or unweighted average.
   * Used to improve stability and accuracy
   * Training dataset: same as ADA
3. "Nnet" Single-hidden-layer neural network:
   * How neural networks work is less intuitive than the other methods of machine learning.
   * Short process description:
     1. Each variable in the training/ testing datasets (so in our case DOB, SSN, etc) is treated as an input with a certain weight. The weights start off with random initialization values.
     2. Then inputs are put through an activation function, which is essentially the black box where the inputs, weights, and an assigned bias are processed into an output.
     3. Then then output is compared to a known value, and the algorithm recalibrates with the result of the comparison.
     4. This process is repeated until the maximum number of allowed iterations or the acceptable error rate is reached.
   * Data needs to be normalized, which can be accomplished using the scale function in R.
   * It seems like the split for the training and testing dataset can be fairly random.
   * Great website with a pretty concise and clear explanation of how this works in R: http://www.kdnuggets.com/2016/08/begineers-guide-neural-networks-r.html/2
4. "bumping" A bootstrap based method using classification trees, very similar to bagging in terms of results and methodology.