Project Draft

#Team: 15, Team ID: C5
Team members: Cristina Zenteno Garcia, Ian Shepherd
GT Usernames: cgarcia66, ishepherd3
GTIDs: 903647123, 903653735

1. Introduction

Record linkage is part of the data cleaning process as it is used to link data coming from different sources that belong to a same entity. One of the uses is in the healthcare setting were record linkage can be used to link medical records from different hospitals to create data that can be used to provide better healthcare services for the patient, track the spread of diseases and improve the quality of data for research.
On this project we will evaluate the reproducibility of the paper, the methods used and its results.

2. Scope of reproducibility

We will test whether an ensemble approach is a better way to address the complex problem of record linkage using three popular supervised machine learning methods: Support Vector Machine, Logistic Regression, Multi-layer Perceptron Neural Network. Then we will test whether the ensemble outperforms base learners in precision, sensitivity, f-score, and number of false matches for both the FEBRL and ePBRN datasets.

3. Methodology

a. Data descriptions

Two synthetic datasets are utilized, Freely Extensible Biomedical Record Linkage (FEBRL) and Electronic Practice Based Research Network (ePBRN). Within FEBRL, we will use FEBRL 3 for training and FEBRL 4 testing. The FEBRL 3 dataset consists of 5000 records, 2000 of which are original and 3000 are duplicate. There is a Zipf distribution of duplicate records ranging from 1 to 5. The FEBRL 4 dataset contains 5000 originals and 5000 duplicates with only 1 duplicate per original. The dataset is designed to be used for testing linkage procedures and thus is an ideal dataset for the study. ePBRN is based on the Australian UNSW Centre for Primary Health Care and Equity. The dataset was built on linkage errors

There are five steps in the pipeline: data cleaning and standardization, blocking, feature engineering, classification, and evaluation. While expansive data cleaning was not required due to some of the work already done on the datasets there are a couple steps. All names needed to be standardized, i.e. lower case and punctuation removed and then tokenized. Next, date of births need to be extracted to day, month, and year for those with a recognizable format with a similar process followed for addresses. Lastly, some basic steps such as establishing data types and addressing null values.

Blocking is then used on the clean dataset with the goal of eliminating pairs that are unlikely to be matched. While the study does not use any massive datasets, it correctly points out as the dataset grows there are exponentially more pairs to compare and thus increased computational costs. Essentially blocking divides the records into various blocks, or groups, that have a higher probability of being linked. The blocks used in the study were given name, surname, and postal code (see figure below). Lastly, feature engineering is utilized via phonetic algorithms to convert values into codes.

Number of True Matched Pairs		
Blocking Criteria	Scheme A	Scheme B
Given name	3287/5000	1567/2653
Surname	3325/5000	1480/2653
Postal code	4219/5000	2462/2653
1+ Match	4894/5000	2599/2653

Scheme A represents FEBRL dataset and Scheme B ePBRN

b. Model descriptions

The final output uses an ensemble classification approach utilizing support vector machines, logistic regression, and multi-layer perceptron neural network that then uses bagging and stacking. There is grid search performed for hyperparameter tuning on the C value for both SVM and logistic regression and alpha for the MLP. Once an appropriate set of C and alpha values are established, the algorithms are tuned on the kernel for SVM , penalty for logistic regression, and activation function for MLP.

Tuning parameters outside the default values as of scikit-learn version 1.1.3.
SVM: C values tested over a range from 0.001 to 5000, kernel=rbf and linear
Logistic Regression: C values tested over a range from 0.001 to 5000, penalty=l1 and l2, max_iter=5000, multi_class='ovr'
MLP: solver='lbfgs', alpha values tested over a range from 0.001 to 5000, hidden_layer_sizes=(256,), activation=relu and logistic, max_iter=10000

Next, we utilized bagging to address potential overfitting via 10 cross validation folds to lower generation errors on unseen data. The bagged results are then stacked and averaged across the three models to lower bias. The idea being this is one way to approach the bias-variance tradeoff.

c. Computational implementation

All algorithms are from the sklearn package on CPU

d. Code
https://github.com/ian-shepherd/CSE6250_BDH_Project

4. Results

After performing hyper-parameter tuning we got the best C for the SVM, MLP and LR models on both schemes.

	Scheme A		**Scheme B**	
Model	**hyper-parameter**	**f-score**	**hyper-parameter**	**f-score**
SVM	Linear kernel with C = 0.001	95.36	Linear kernel with C = 0.001	80.47
MLP	Relu activation with C = 1000	98.87	Relu activation with c = 2000	81.17
LR	Regularization L2 with C = 0.001	98.71	Regularization L2 with C = 0.001	79.32

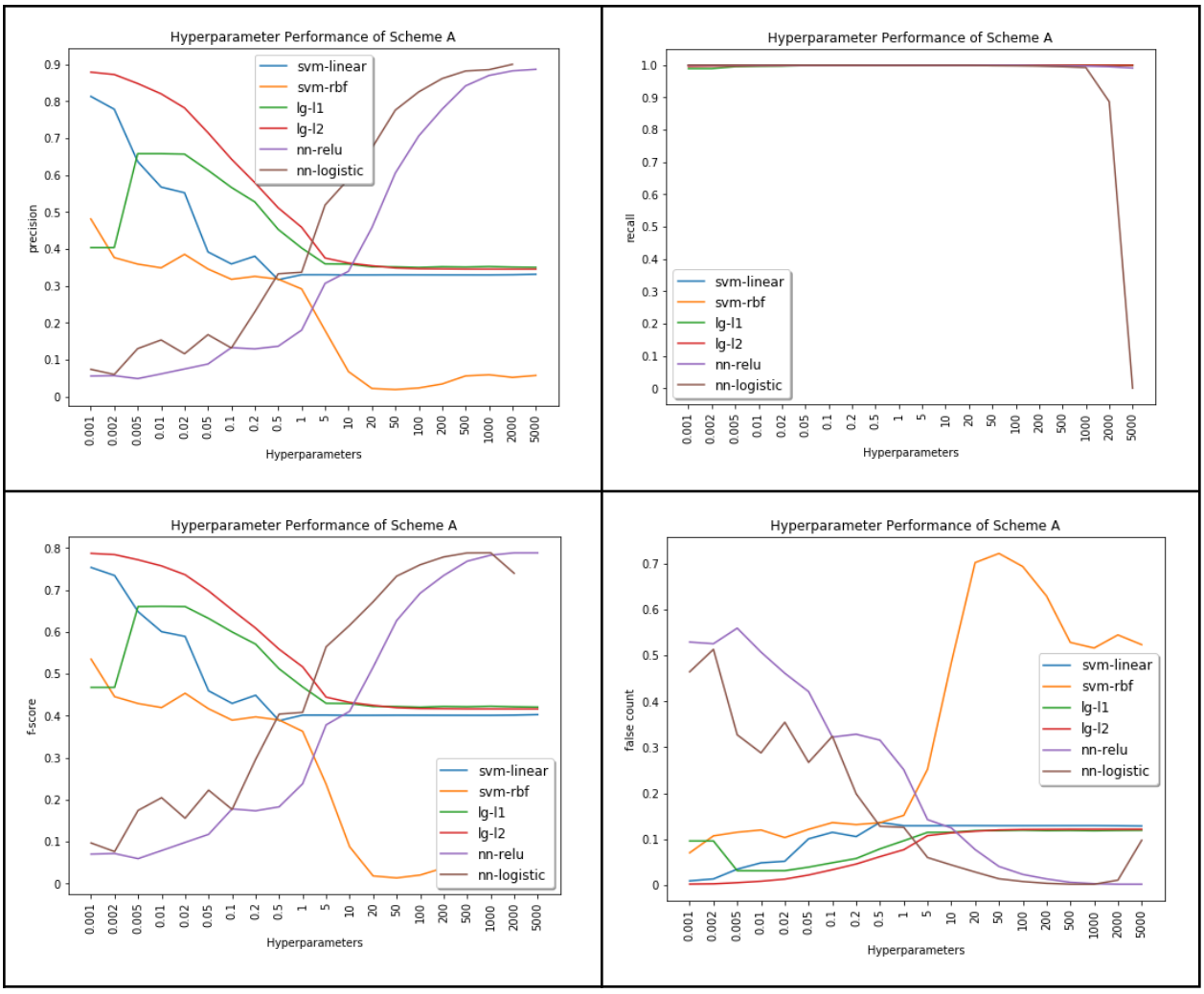
Then we use those C to run the models by themselves and as base models for bagging and stacking to compare which one achieved a highest matching or lowest false count(false positive + false negative)

	Scheme A				**Scheme B**			
	precision	**recall**	**f-score**	**false count**	**precision**	**recall**	**f-score**	**false count**
SVM	91.32	99.78	95.36	475	69.13	96.25	80.47	1220
SVM-bag	92.99	99.75	96.25	380	69.13	96.25	80.47	1220
MLP	95.56	99.18	98.87	111	70.08	96.44	81.17	1168
MLP-bag	98.64	99.06	98.85	113	72.25	96.4	82.59	1061
LR	97.89	99.55	98.71	127	67.33	96.51	79.32	1314
LR-bag	98.07	99.51	98.78	120	67.68	96.48	79.55	1295
Meta-ensemble	98.76	98.96	98.86	112	76.16	96.17	85	886

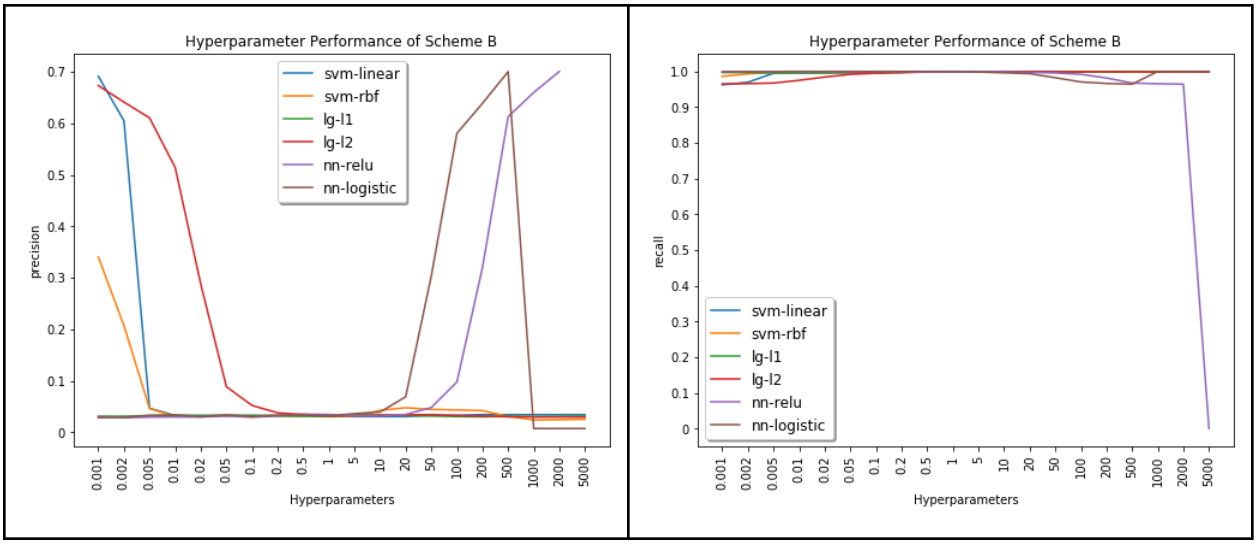
As result we can see that when using Scheme A, the MLP models perform almost the same as Stack and Bag, but when using Scheme B, the latter outperforms all the other models.

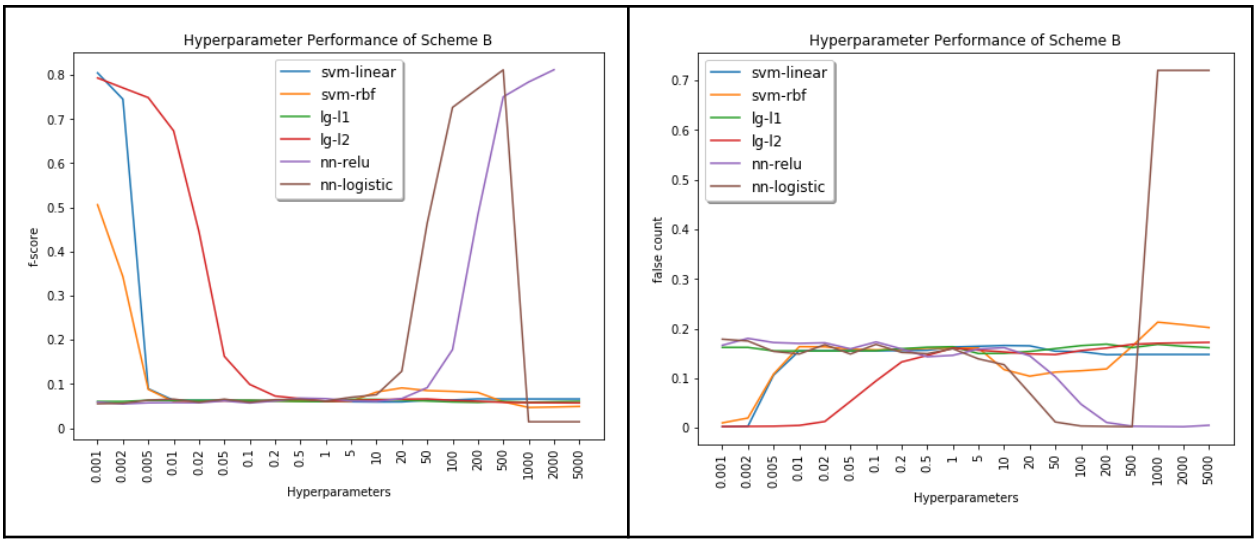
This partially contradicts the result of the paper which concluded that the meta-ensemble algorithm(bagging and stacking) performed significantly better on Scheme A and similar to the best base model SVM in Scheme B.

Scheme A


Hyperparameter Performance of Scheme A
precision
Hyperparameters
svm-linear
svm-rbf
lg-l1
lg-l2
nn-relu
nn-logistic
Hyperparameter Performance of Scheme A
recall
Hyperparameters
Hyperparameter Performance of Scheme A
f-score
Hyperparameters
Hyperparameter Performance of Scheme A
false count
Hyperparameters

Scheme B


Hyperparameter Performance of Scheme B
precision
Hyperparameters
svm-linear
svm-rbf
lg-l1
lg-l2
nn-relu
nn-logistic
Hyperparameter Performance of Scheme B
recall
Hyperparameters



5. Discussion

Blocking does a very good job matching pairs between both scheme A and scheme B. This significantly decreases the computational power needed to train the models and a good sign for scaling the implementation in the future. As for the implementations themselves, scheme A appears to get the best results when C/alpha is between 0.2 and 5.0. The plots above illustrate the advantage of an ensemble approach. At various points along that range, different implementations are better than others with all of them fairly performant. By combining them all together we are more robust to overfitting. Scheme B on the other hand, very clearly shows it has poor precision and thus f-score when C/alpha is between 0.1 and 50.0. However, by bagging and stacking we are still able to extract good scores even from some imperfect implementations achieving 0.9645 precision, 0.9949 recall, and 0.9795 f-score for scheme A and 0.7278, 0.9638, and 0.8293 for scheme B.

The remaining plan includes trying to improve the code base to be more efficient and to allow for testing of a more expansive grid search. Time permitting, we can also test adding another implementation to the ensemble like a random forest classifier or a gaussian naive Bayes.

6. References

[1] Kha Vo, Jitendra Jonnagaddala, Siaw-Teng Liaw, Statistical supervised meta-ensemble algorithm for medical record linkage, Journal of Biomedical Informatics, Volume 95, 2019, 103220, ISSN 1532-0464, https://doi.org/10.1016/j.jbi.2019.103220.

[2] Kevin O'Hare, Anna Jurek-Loughrey, Cassio de Campos, An unsupervised blocking technique for more efficient record linkage, Data & Knowledge Engineering, Volume 122, 2019, Pages 181-195, ISSN 0169-023X, https://doi.org/10.1016/j.datak.2019.06.005.

[3] P. Christen, T. Churches, FEBRL - Freely extensible biomedical record linkage, Available at Febrl - Freely extensible biomedical record linkage (anu.edu.au)

[4] UNSW, The Electronic Practice Based Research Network(ePBRN). Available at The Electronic Practice Based Research Network | Centre for Primary Health Care and Equity (unsw.edu.au)

[5] ReadTheDocs, Python Record Linkage Toolkit. Available at About — Python Record Linkage Toolkit 0.15 documentation