Machine Learning

CS 6375.004

Final Project Report

Instructor: Anurag Nagar

Student: Guang Yang (UTD ID: gxy141230)

Date: 04/20/2016

1. Introduction

A blood transfusion is the transfer of blood or blood products from one person (donor) into another person's bloodstream (recipient). This is usually done as a lifesaving maneuver to replace blood cells or blood products lost through severe bleeding, during surgery when blood loss occurs or to increase the blood count in an anemic patient. In this project, the data were taken from Blood Transfusion Service Center in Taiwan. The center passes their blood transfusion service bus to one university to gather blood donated about every three months. To build the model, the dataset selected 748 donors at random from the database. In details of these 748 donors, data will be expanded in the Table1.

Table 1

Title	Blood Transfusion Service Center Data Set
Number of Attributes	5
Number of instances	748
Type of output variable	Binary variable representing whether he/she donated blood (1 stand for donating blood; 0 stands for not donating blood).
DataSet Characteristics	Multivariate
Attribute Characteristics	Real
Attribute Information: The order of this listing corresponds to the order of numerals along the rows of the database. R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation).	

2. Problem Definition and Algorithm

2.1 Task Definition

The problem we are addressing is based on the dataset of the Blood Transfusion Service Center, we implement the different classifier techniques (such as SVM, Bagging, Random Forest, AdaBoosting, and Gradient Boosting) to create models to predict whether a donor will donate blood.

2.2 Algorithm Definition

Table 2

Technique	**Method in sklearn class**
SVM	sklearn.svm.SVC
Bagging	sklearn.ensemble.BaggingClassifier
Random Forest	sklearn.ensemble.RandomForestClassifier
Ada Boost	sklearn.ensemble.AdaBoostClassifier
Gradient Boosting	sklearn.ensemble.GradientBoostingClassifier

1). C-Support Vector Classification: The implementation of our code is based on libsvm. The complexity of fit time is more than quadratic with the total number of samples and thus this shortcoming makes it impractical to scale to dataset with more than a couple of 10000 samples.

2). Bagging classifier: It is an ensemble meta-estimator which fits base classifiers each on random subset of the dataset at the first place and then aggregate each individual prediction to form a final prediction by voting or by averaging.

3). Random forest classifier: It is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

4). AdaBoost classifier: it is a classifier that begins by fitting a classifier on the original dataset and then fits additional classifier on the same dataset, however when the classified instances are incorrectly classified, the weights are adjusted in such a way that subsequent classifiers focus more on previous incorrect classified cases.

5). Gradient Boosting: this classification model is built in a forward stage-wise fashion which allows for the optimization of arbitrary differentiable loss functions. Each regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function.

3. Experimental Evaluation

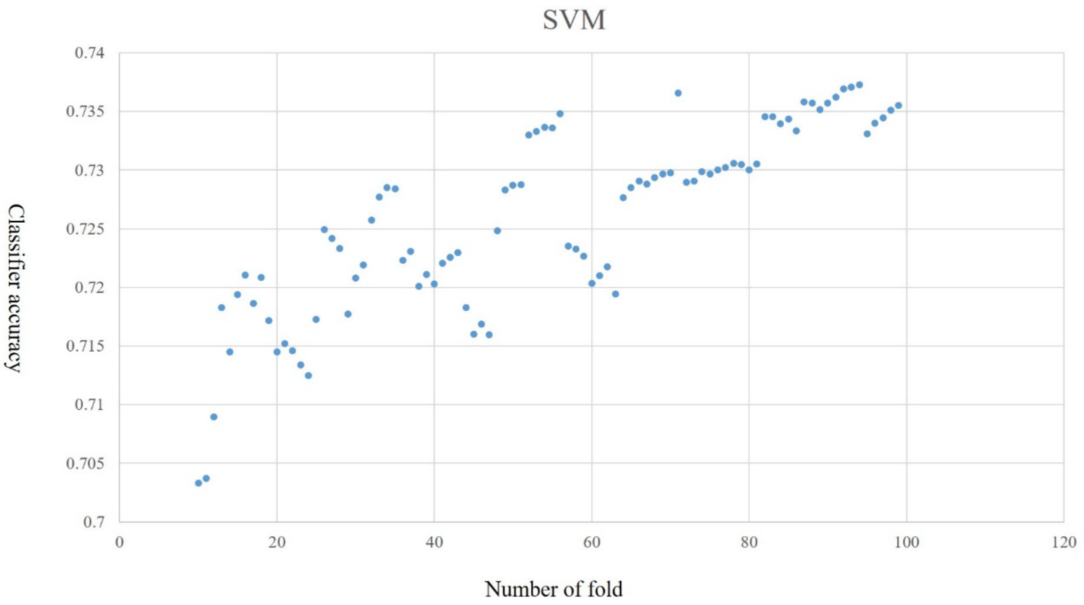
3.1 Methodology

In this project, we are using five techniques: support vector machines, bagging, random forest, AdaBoost and gradient boosting.
At first, we preprocess the blood-transfusion dataset, such as normalization. But we found that, before and after normalization, the accuracy did not change significantly. This means that the data in this dataset is in a valid range and it does not affect the model much. The goal of cross validation is to define a dataset to "test" the model in the training phase in order to limit problems like overfitting give an insight on how the model will generalize to an independent dataset. So we also used cross-validation with these five techniques by testing different folds (from 10 to 100), find the accuracy trend and choose the best fold.
We also adjusted some key parameters for each technique (kernel for SVM; max_samples and max_features for Bagging; n_estimators, min_samples_split, max_depth and max_features for Random Forest; n_estimators and learning rate for AdaBoost; n_estimators, learning_rate and max_depth for Gradient Boosting). After testing several choices during a big range, we chose the better one.

3.2 Results

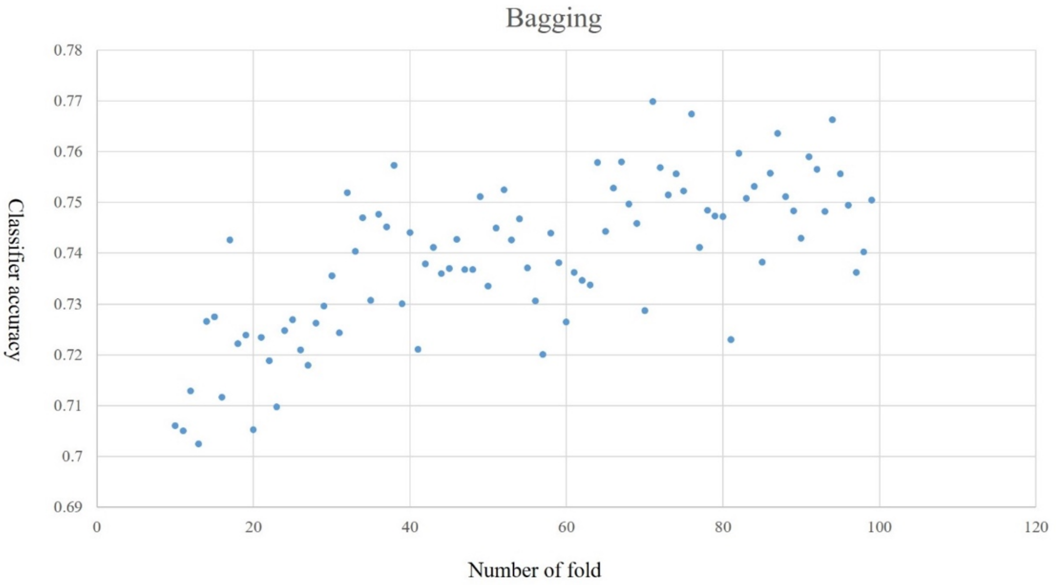
For each classifier technique, we test different fold with chosen parameters to get the trend of accuracy according to the number of fold.

3.2.1



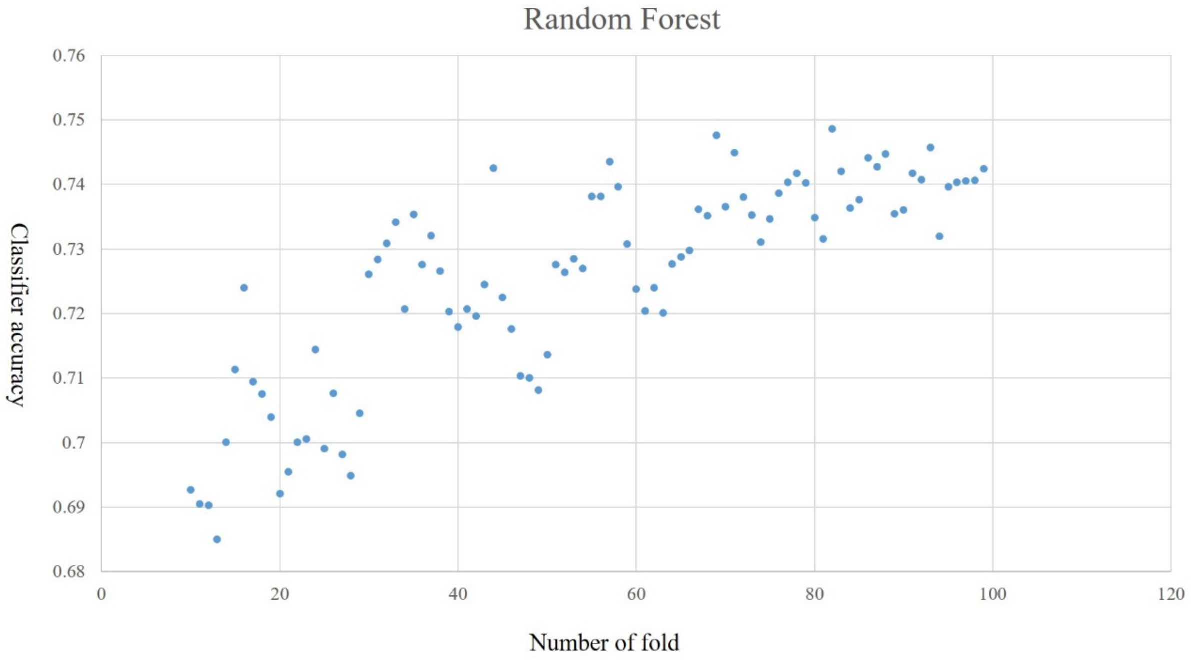
For SVM ('svm' method in package 'sklearn'), there is an ascending trend of accuracy as the number of fold increasing, but the trend will arrive at a constant accuracy and not increase while the number of fold increases to about 50-60. Also the accuracy varies strongly, that means SVM may not be continuous with the number of fold.

3.2.2



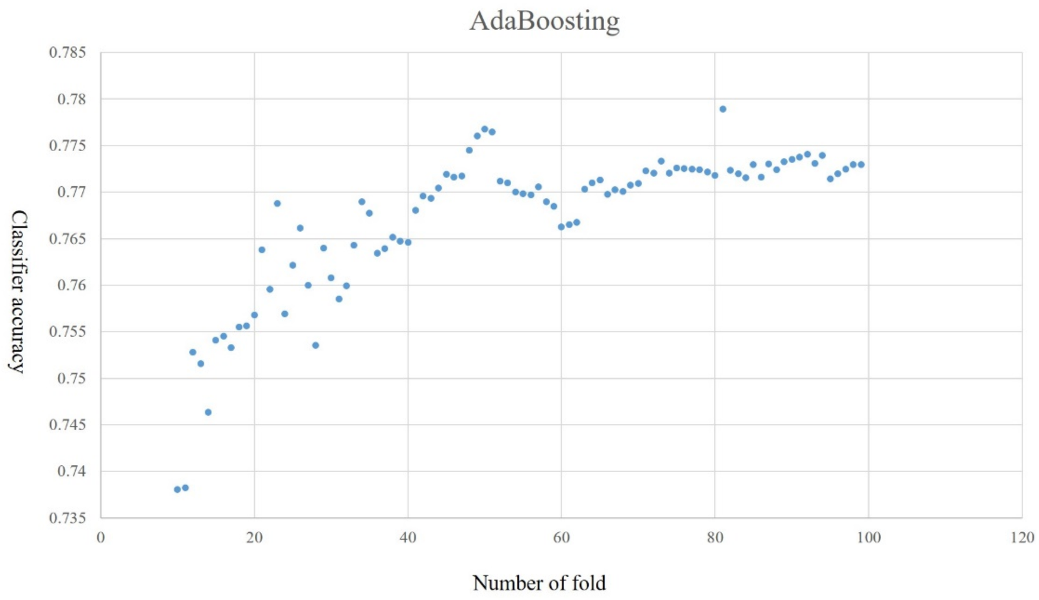
For Bagging, there is the same trend as the SVM. When the number of fold arrives to about 50-60, the accuracy will arrive to the best and not increases as the number of fold increases.

3.2.3



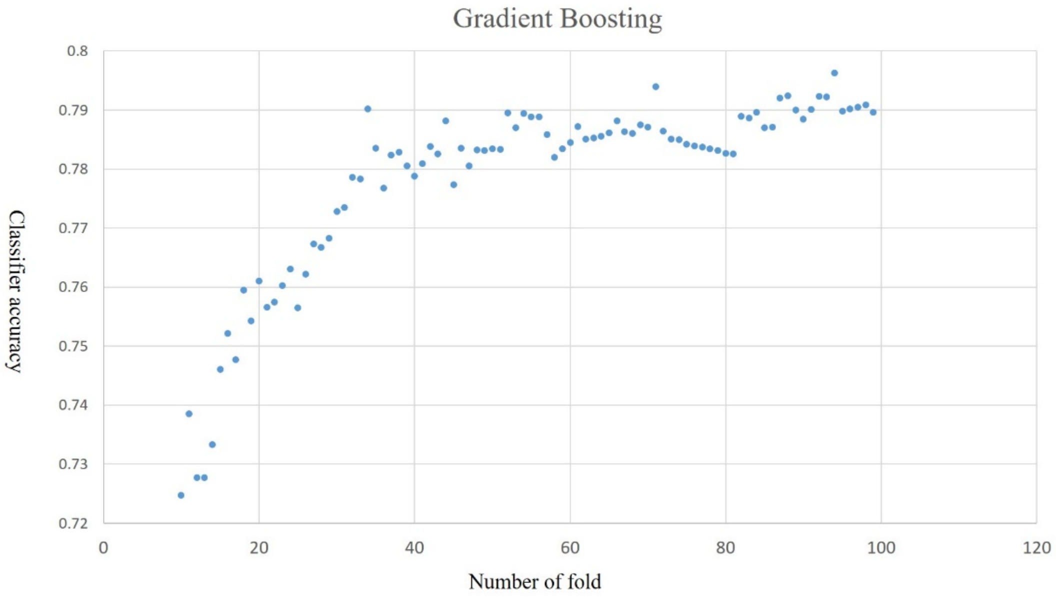
For Random Forest, there is the same trend as the SVM and Bagging, but it also has a zip shape. When the number of fold arrives to about 50-60, the accuracy will arrive to the best and not increases as the number of fold increases.

3.2.4



For AdaBoosting, there is the same trend as the previous classifiers, but it also has a less variance, that means AdaBoosting maybe more continuous. When the number of fold arrives to about 50-60, the accuracy will arrive to the best and not increases as the number of fold increases.

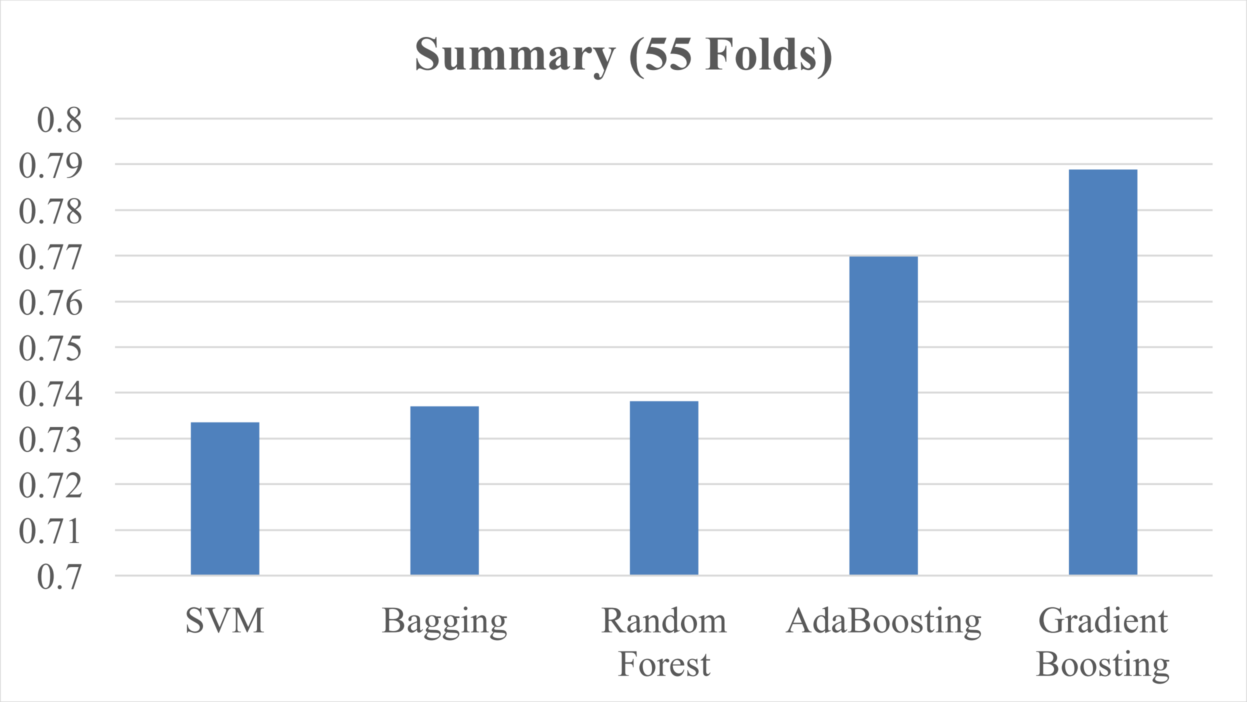
3.2.5



For Gradient Boosting, there is the same trend as the AdaBoosting. When the number of fold arrives to about 50-60, the accuracy will arrive to the best and not increases as the number of fold increases.

3.2.6

Table 3



As choosing 55 folds for crossing validation, the Gradient Boosting and AdaBoosting have the better accuracies. SVM, Bagging and Random Forest have lower accuracies.

3.3 Discussion

Support vector machines (SVM) could perform both linear classification and non-linear classification by using different kernels to find the optimal separation hyperplane. In this project, we choose non-linear kernel. As the little number of attributes, the accuracy is not very high.

Bagging (stands for Bootstrap Aggregation) is the way decrease the variance of your prediction by generating additional data for training from your original dataset using combinations with repetitions to produce multiset of the same cardinality/size as your original data. By increasing the size of your training set you can't improve the model predictive force, but just decrease the variance, narrowly tuning the prediction to expected outcome.

As Random Forest selected only a subset of features at random out of the total and the best split feature from the subset is used to split each node in a tree, unlike in bagging where all features are considered for splitting a node, Random Forest usually is better than bagging. In our project, Random Forest and bagging nearly have the same accuracy. This may be caused by the little number of attributes of the original data.

Boosting is an approach to calculate the output using several different models and then average the result using a weighted average approach. By combining the advantages and pitfalls of these approaches by varying your weighting formula you can come up with a good predictive force for a wider range of input data, using different narrowly tuned models. The aim for boosting is to decrease bias. Usually boosting is a good technique to construct models.

In our project, two boosting methods—AdaBoosting and Gradient Boosting get better accuracy than other methods. Therefore, for the blood-transfusion dataset, the boosting is a powerful technique with high prediction accuracy.

4. Other Researcher's Work

Paper title: A hybrid classification algorithm and its application on four real-world data sets
Journal: International Journal of Computer Science and Information Security (IJCSIS), Vol. 13, No. 10, October 2015

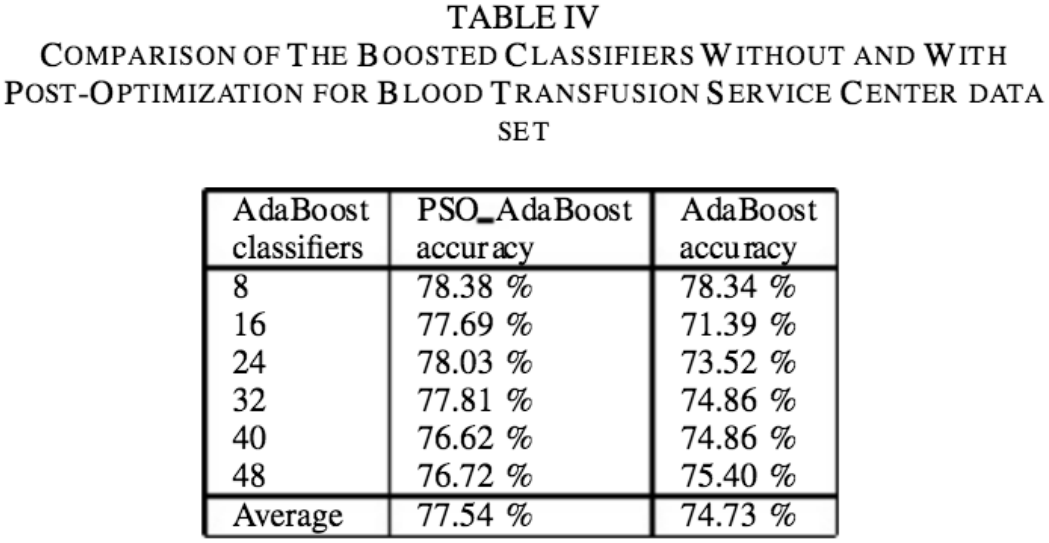


TABLE IV
COMPARISON OF THE BOOSTED CLASSIFIERS WITHOUT AND WITH POST-OPTIMIZATION FOR BLOOD TRANSFUSION SERVICE CENTER DATA SET

AdaBoost classifiers	PSO_AdaBoost accuracy	AdaBoost accuracy
8	78.38 %	78.34 %
16	77.69 %	71.39 %
24	78.03 %	73.52 %
32	77.81 %	74.86 %
40	76.62 %	74.86 %
48	76.72 %	75.40 %
Average	77.54 %	74.73 %

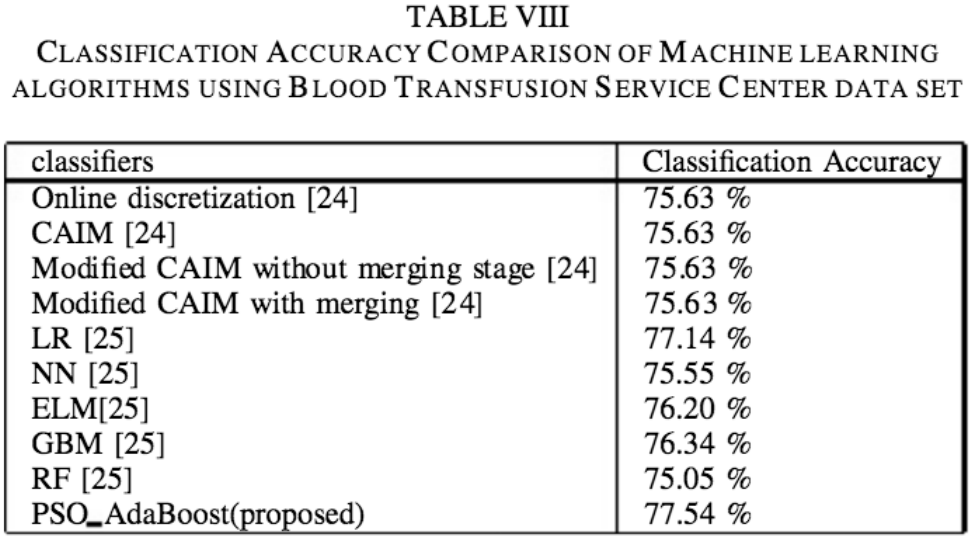


TABLE VIII
CLASSIFICATION ACCURACY COMPARISON OF MACHINE LEARNING ALGORITHMS USING BLOOD TRANSFUSION SERVICE CENTER DATA SET

classifiers	Classification Accuracy
Online discretization [24]	75.63 %
CAIM [24]	75.63 %
Modified CAIM without merging stage [24]	75.63 %
Modified CAIM with merging [24]	75.63 %
LR [25]	77.14 %
NN [25]	75.55 %
ELM[25]	76.20 %
GBM [25]	76.34 %
RF [25]	75.05 %
PSO_AdaBoost(proposed)	77.54 %

These two tables are cited from: International Journal of Computer Science and Information Security (IJCSIS), Vol. 13, No. 10, October 2015 p96

We compare our result with other machine learning researcher's work on this same dataset (Blood Transfusion Service Center data), as can be seen from the accuracy table, though we did not use the sophisticated data preprocessing and post optimization methods as Dr. Bakrawy and Dr. Desuky, we still achieve very similar good results due to parameter tuning.

5. Future Work

1. In our dataset, there are a few attributes in our dataset for our class label (whether he/she donated blood), in reality as we all know, there should be more attributes related to the class label, like donor's recent health situations, donor's occupations, schedule and so on. Therefore, predicting the outcome of our model simply based on just four attributes is far from enough, and the data collector would need more data points and class attributes to make our model more accurate and reliable.

2. In our dataset, the problem of class imbalance is very severe, since "real-world" problems does occur a lot of this problem, to preserve the integrity of the original data in our dataset, we did not choose to revise and using too many sampling method, instead we are focusing on tuning the parameter to select the best model just as other researchers do.

3. During parameter tuning process, some techniques we used have a lot of parameter to tune, due to the limit of calculation speed, it is not possible to tune all the parameter in order to find the best model, thus we only consider a few of the most significant parameter and choose the best case.

4. Due to the nature of class project, the professor only assign few classifier to us in our project, if possible, we would test more classifier learned in class , and test each of them to have a better grasp of all the techniques in our machine learning class.

6. Conclusion

In this report, we have shown that the techniques we have used to create the model and analyze our selected dataset. However, as presented in the part five, we believe that this is just a first step towards more systematically analyzed procedure on a more detailed dataset.

In our report, we have provided only a partial glimpse of results relating to the accuracy of our five algorithms. We use normalized data-preprocessing methods to adjust the range and optimize our data and use different approach of tuning techniques to tune for a more accurate model, and cross validation method is also used to prevent over-fitting, however, due to the nature of this dataset, there is a certain limit of our data. In the end, we compared our model and its corresponding result with other researcher's work on this same dataset, our result has close match which are the proofs of the correctness of our model and approach. Due to the nature of class project, we only adopt five classifiers in our project, if possible, it would be interesting in the future to test more classifier learned in class and have an overview and analysis of all the techniques we have learnt. During the five techniques we chose, we find that the boosting techniques have better results. As the boosting could decrease bias, this maybe the reason why it works better.