A Small Replication of CNM06 Supervised Machine Learning

Dorine V. Ernst
dvernst@ucsd.edu

Abstract

This research paper is a small subset replication of Caruana and Niculescu-Mizil 2006 paper. The aim of this research paper is to find out which algorithm does the best at the particular datasets. This paper will use four datasets from UCI Repository Classification. These datasets will be analyzed using three algorithms, which are: Logistic Regression, Decision Tree, and Random Forest. All the algorithms will be tuned using hyperparameter tuning to build a model and carry-on model comparison across the datasets and see the performance.

1. Introduction

This paper is a small-scale of CNM06, it will introduce three different algorithms such as Logistic Regression, Decision Tree, and Random Forest. Each of these algorithms will be trained and tuned using different hyperparameter tuning to choose the best parameters for each algorithm.

Cross validation will be introduced to maximize the use of each data to train the model and obtain the best result. Using five thousand data points to train the data and applying five cross validation to get the best model. The training will be repeated five times using different hold-out data from the training sets.

After obtaining the model, will be continuing the process by choosing the best parameters to train the five thousand data points and using the model to do the prediction in the test set and compare the result.

We will measure the performance of each results using five different metrics to accurately capture the differences of each model. The five metrics that will be introduced are accuracy, F1-score, precision, recall, and AUC.

All the datasets that are used in this paper are taken from UCI Repository Classification datasets. By using different metrics to measure the performance of the model, we can see understand which metrics perform the best with respect to different datasets.

The best empirical result across four datasets is logistic regression. While for decision tree and random forest tend to over fit in some datasets and can perform well in banking and default datasets.

2. Methodology

The algorithms that are used in this paper carefully following CNM06 paper with the same hyperparameter tuning. We will tune each algorithm using different parameters and train the data using the best parameters that we get from using the best model from GridSearchCV. To obtain the best result from each algorithm, five-fold cross validation will be applied here. The algorithms performance is different with CNM06 due to carrying the algorithms into four different datasets from CNM06 paper.

A Small Replication of CNM06 Supervised Machine Learning

Dorine V. Ernst
dvernst@ucsd.edu

2.1 Algorithm

Logistic Regression (LOGREG):
In this algorithm, we will train the model using regularized and unregularized models by tuning the kernel C using parameter from 10^{-8} to 10^{4} as well as infinity (for the unregularized parameter). Additionally, the model will be trained by two differ parameters which is l1 and l2 norm.

Decision Tree (DT): Using decision tree, the model will be trained using best splitter to choose the best split. We will also be varying the criterion to tune the hyperparameter using gini and entropy to gain the best model.

Random Forest (RF): We will be varying the criterion to tune the hyperparameters using gini and criterion. 1024 tress will be implemented for random forest algorithm to attain the best model. The size will be varied for each split into 1,2,4,6,8,12,16, or 20. For some datasets whose attributes is less than 10, we will vary the size into 1,2,4,6, or 8.

2.2 Performance Metrics

In this paper we will look into different metrics to catch distinct performance of each model, we will be using five different metrics to measure the model for every dataset. We will apply the metrics into GridSearchCV and applying it after the first training to determine the three best model and using the best metrics to measure.

2.3 Comparing Across Performance

Metrics: The performance metrics depends on the different datasets. For instance, AUC method cannot run in the Polish datasets. This is because the y prediction are all predicting 0 due to low value of percentage positive in the datasets.

Accuracy is not a really good performance in these datasets since the baseline for the accuracy is 0.86. Recall and AUC are pretty good metrics while it gives baseline 0.61 and 0.70 respectively.

The worst metrics to be applied in these datasets are F1 score whose baseline is 0.53 and followed by precision with 0.51 baseline.

2.4 Data Sets:

The datasets that will be used in this paper are taken from UCI Repository Classification. We will compare three different algorithms into four different binary classification datasets.

The datasets that are used in this paper are Avila, Bank Marketing, Default Credit Cards, and Polish Bankruptcy Dataset. To see all of the details of the dataset information, refer to Table 1 below

Dataset	#Attr	Train size	Test size	%Pos
Avila	11	5000	5430	41%
Marketing	17	5000	40,211	47%
Default	24	5000	25,000	22%
Polish	64	5000	2027	2%

A Small Replication of CNM06 Supervised Machine Learning

Dorine V. Ernst
dvernst@ucsd.edu

Avila dataset has 11 attributes and predicting class distribution. If the class is A then we will classify it as 1 and 0 for other class. We will scale the rest of the data to normalize the distribution of the data.

Bank Marketing dataset consists of 17 attributes to predict y (client term deposit). The y targets already a binary data. For this dataset, we will do one hot encoding for the datasets that are object. After doing a one hot encoding, the columns for this dataset become 53 columns. Scaling will be applied into the datasets to normalize the datasets.

Default Credit Card dataset has 24 attributes and predicting Y (whether the credit card is default or not). The Y data itself is a binary dataset. Scaling also has been applied to the dataset to normalize the data.

Polish Companies Bankruptcy Data has 64 number of attributes to predict the class. Since the y class is binary, we will scale the rest of the data (leaving the y out) to normalize the data.

3. Performances by Metric

For each test sets will be trained by the best three hyperparameter that has been tuned from randomly chosen 5000 training sets data points. Each model will be five-fold cross validated to ensure the best prediction.

After the prediction has been generated, we will compare it with the actual y and measure the performance of each model using different metrics.

To see the detailed result of each test set performance, please refer to table 2. In the table we can see by the algorithm-dataset combination. The performance of the best datasets is **boldfaced**. * in the table noted the algorithm that do not have the significance different from the best algorithm using the paired t-test.

From table 2, it shown that

4. Performances by problem

Every test sets that has been trained using the best model that has been tuned by the hyperparameter and cross-validated five times, we will see which algorithm-metric combo performs the best across four datasets.

Please refer to table 3 to see the detailed information. We can see that Random Forest in general perform a pretty good prediction. Random Forest -AUC performs the best combo by giving the accuracy rate 0.9816 though out the four datasets. Followed by Recall, Accuracy, F1, and Precision.

Followed by Decision Tree-Accuracy combo and AUC, Recall, F1, and Precision respectively.

A Small Replication of CNM06 Supervised Machine Learning

Dorine V. Ernst
dvernst@ucsd.edu

The last will be Logistic Regression (performs the worse compares to the rest algorithms) combo with Accuracy and followed by Recall, AUC, F1, and Precision respectively.

5. Conclusion

As we know we have no free lunch theorem where there is no best algorithm, it will perform good in some datasets and poorly in different datasets. From this paper, we can see that Random Forest performs the best across these datasets which generally are binary business classification. Followed by decision tree, and logistic regression.

From the table 3, we can see that Random Forest and Decision Tree have a stable performance across the datasets, while for Logistic Regression, the appropriate metrics will be accuracy metrics.

6. References

Caruana, R and Nisculuescu-Mizil, A., 2020. An Empirical Comparison of Supervised Learning Algorithms. (https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf)

Github Scikit-learn 2020. Scikit Model Selection Validation (https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/model_selection/_validation.py)

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011 (https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html) (https://scikit-learn.org/stable/modules/tree.html) (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html) (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html) (https://scikit-learn.org/stable/modules/grid_search.html)

A Small Replication of CNM06 Supervised Machine Learning

Dorine V. Ernst
dvernst@ucsd.edu

Table 2:

Datasets-Algo combo	Accuracy	AUC	Precision	Recall	F1 score	Mean
Avila - logreg	0.6888	0.6801	0.5111	0.6557	0.5741	0.6220
Avila - dt	0.9527	0.9516	0.9389	0.9455	0.9422	0.9462
Avila - rf	0.9816	0.9818	0.9719	0.9830	0.9774	0.9792
Bank marketing - logreg	0.8204	0.8209	0.7877	0.8254	0.8061	0.8121
Bank marketing - dt	0.7808	0.7803	0.7648	0.7708	0.7677	0.7729
Bank marketing - rf	0.8514	0.8522	0.8825	0.8183	0.8492	0.8507
Default - logreg	0.8076	0.7616	0.2236	0.7078	0.3362	0.5674
Default - dt	0.7271	0.6084	0.4049	0.3881	0.3962	0.5049
Default - rf	0.8128	0.7378	0.3584	0.6377	0.4576	0.6009
Bankrupt - logreg	0.9798	N/A	0.0000	0.0000	0.0000	0.1960
Bankrupt - dt	0.9695	0.6152	0.2426	0.2460	0.2436	0.4634
Bankrupt - rf	0.9795	0.6360	0.0111	0.2920	0.0212	0.3880

ttest	Acc	AUC	Precision	Recall	F1 score
logreg	0.16412356	0.25588737	0.08662897	0.13724093	0.09646465
dt	0.03493387	0.04198933	0.34907052	0.08099494	0.44303543

Since random forest appears to be the best algorithm, we will do t-test with alpha 0.05 to the random forest to the other algorithms.

Table 3:

Model	Accuracy	AUC	Precision	Recall	F1 Score
LOGREG	0.9798	0.8209	0.7877	0.8254	0.8061
DT	0.9695	0.9516	0.9389	0.9455	0.9422
RF	0.9816	0.9818	0.9719	0.9830	0.9774

Link to the table in excel :
https://drive.google.com/file/d/1ws19wLbK_8WO-5y3D3g_xA2wyvzWvLJV/view

A Small Replication of CNM06 Supervised Machine Learning

Dorine V. Ernst
dvernst@ucsd.edu