Reproducing the paper "An Empirical Comparison of Supervised Learning Algorithms" by R. Caruana and A. Niculescu-Mizil

Alena Sorokina

Department of Mathematics, Nazarbayev University
alena.sorokina@nu.edu.kz

1 Executive overview

The main aim of the following project is to reproduce a scientific paper, which use several machine learning algorithms. I chooses the paper by R. Caruana and A. Niculescu-Mizil entitled "An Empirical Comparison of Supervised Learning Algorithms". Originally the paper used 10 supervised classification algorithms, evaluated by 10 different metrics on 11 data sets. However, it would be computationally expensive to reproduce all the results. Therefore, I decided to focus on the main results, and reproduce 9 algorithms (as the main focus of the course was machine learning algorithms), but evaluate them on 2 metrics (accuracy and F1 score, which we covered in the class) and use only one data set. The evaluation on the other data sets is not a complex work, given the reproduced experimental set-up, but it will require more time and will be taken as a future work.

I decided to reproduce the models on Adult data set, which is used to predict whether income exceeds 50K/yr based on census data. It contains person's annual income as an output variable (**2 classes:** $< 50K$**,** $\geq 50K$) and 14 features as input variables (**age, work class, sampling weight, education, education-num, marital status, occupation, relationship, race, sex, capital gain, capital loss, hours per week, native country**).

The following classification models were reproduced: Support Vector Machines (**SVMs**), Artificial Neural Network (**ANN**), Logistic Regression (**LogReg**), Naive Bayes (**NB**), k-Nearest Neighbours (**KNN**), Random Forest (**RF**), Decision Trees (**DT**), Bagged trees (**BAG-DT**), boosted trees (**BST-DT**).

The experimental set-up was the same as in the original paper, except from Decision Trees algorithm. In total, I have trained about **200** different models in each trial on Adult dataset. The results appeared to be comparable with the original paper (by Accuracy and F1 metrics).

2 Overview of the data and its pre-processing

Adult data-set was taken from UCI Machine Learning Repository. It contains 14 attributes (both categorical and continuous variables) with 48842 observations per each attribute. Prediction task is to determine whether a person makes over 50K a year.

The following pre-processing was done:

- The categorical columns were set to type **category**.
- Multi-class categorical features were represented as a binary features, using one-hot encoding.
- Train-Test split was done using **sklearn "Model Selection"** package. In the oiginal paper, the Train size was equal 5000, and the rest was treated as a Test data, and I have done the same set-up.
- Normalization using **sklearn "Standard Scaler"** was applied, firstly fitting only on training data and then applying same transformation to test data

3 Models set-up and Results

3.1 Experimental set-up

- **SVMs**
  I used the following kernels in SVM (sklearn): linear, polynomial degree 2 and 3, radial with width $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$. I also varied the regularization parameter by factors of ten from 10^7 to 10^3 for each kernel.

- **ANN**
  I trained neural nets with gradient descent backpropogation and vary the number of hidden units $\{1, 2, 4, 8, 32, 128\}$ and the momentum $\{0, 0.2, 0.5, 0.9\}$.

- **LogReg**
  I trained regularized models, varying the regularization parameter by factors of 10 from 10^8 to 10^4.

- **NB**
  I used sklearn package with single normal modeling and discretizied using supervised discretization modeling.

- **KNN**
  I used 26 values of K ranging from $K = 1$ to $K = \|trainset\|$ for KNN with Euclidean distance.

- **RF**
  I used sklearn implementation. The forests have 1024 trees. The size of the feature set considered at each split is $\{1, 2, 4, 6, 8, 12, 16, 20\}$

– **DT**
As there was not an open-source implementations of the trees described in the original paper, I decided to use sklearn implementation of the DTs, varying the informational criterion: gini vs entropy.

– **BAG-DT**
I bagged 100 trees of each type described above.

– **BST-DT**
I there was not an open source implementation for the Boosted tree algorithm, I decided to use modern boosting algorithm ADA boost from sklearn for each tree type.

3.2 Results

The results of evaluation are provided in Table 1, in which the Accuracy score and F1 score are presented for each of the models.

Table 1. Results

Supervised learning algorithm	Average Accuracy Score	Average F1 score
SVMs	0.791907	0.436243
Neural Network	0.818805	0.562039
Logistic Regression	0.817635	0.649641
Naive Bayes	0.781054	0.632630
K-Nearest Neighbours	0.767873	0.284860
Random Forest	0.846146	0.651944
Decision Tree	0.808631	0.606378
Bagged trees	0.848501	0.659609
ADA boost	0.808745	0.606701

4 Discussion

Generally, my results are comparable with ones in the original paper, because I tried to maintain similar experimental set-up. However, due to some limitations

of my work, e.g. using only one data-set, some of the results are different (I compared my results only with a non-scaled results from the original paper).

In some cases, namely **LogReg**, **NB**, **DT**, **ANN** (Accuracy metric) and **KNN** (Accuracy metric), my models outperform the original models, and the reason could be that I used the **sklearn** package, which is newer and more optimized then the implementations discussed in the paper (the year of publication is 2006).

Another problem was the absence of the open-source implementation for the Decision tree based algorithms. I used slightly different implementation of DT from **sklearn**, and this led to the dissimilar performance.

Due to the time constrains, the evaluation of these algorithms on the other data sets and metrics is taken as a future work.

5 Conclusion

In this project, I tried to reproduce 9 supervised learning algorithms and evaluate them on 2 metrics. Generally, the experimental set up in the original paper was clear, thus, the results were reproducible. However, due to the fact, that in the most cases I used the sklearn package, some of the results are slightly differ. During the working process, I gained a valuable knowledge, namely how to preprocess the data, how to implement supervised learning algorithms with a given experimental set-up and build an optimized code, which gives the results in the necessary form (output of each model's function is an average accuracy and average F1 score). Moreover, my skills of the fast typing in Overleaf were also improved.