# A systematic comparison between different base learners in AdaBoosting model

**Mateus Maia Marques[1], Anderson Ara[2]**

Programa de Pós-Graduação em Matemática - UFBA

Instituto de Matemática e Estatística

[1]mateusmaia11@gmail.com ,[2]anderson.ara@ufba.br

## 1. Introduction

The Adaptative Boosting algorithm (AdaBoost), developed by Freund and Schapire (1997), has been showed to be a great statistical model that can outperform a lot of others statistical learning algorithms. Like all others ensemble methods, the AdaBoosting is built by the combination of several models that vote to classify and predict an observation. In AdaBoosting, these classifiers are modeled sequentially and each new model it's weighted considering the capacity to predict correctly the previous misclassified observations. In this process the type of each is fixed and is called as the base model. Generally, the base models are the decision-tree algorithm (C4.5) (Quilan,1993), however, any other weak leaner can be used in AdaBoosting. In order to explore the capacity of use a variety of base learners, this work presents a complete comparison between the most commons models used in the statistical learning tasks in their simplest form. The definition of a weak learner can be set as the model that classifies observations a little better than random guessing. Also, in the most cases, those weak models are associated with linear classification rules or decision boundaries. Many researches focus on enhancing the performance of AdaBoost, by choosing more discriminant classifier (Ratsch, 2001; Schwenk and Bengio, 2000; Li et al., 2008), so change the base learner it's way to emphasize this aspect.

The AdaBoost algorithm was modified to use the following models: K Nearest Neighbors, Linear Discriminant Analysis, Logistic Regression, Neural Networks, Support Vector Machines. All of them were applied in 10 datasets, and their accuracy were evaluated using a repeated holdout validation technique.

## 2. Methodology

Essentially, boosting consists of repeatedly uses a weak learning algorithm, on differently weighted versions of the training data, yielding a sequence of weak classifiers that are combined in a addiction function. The weighting of each model depends on the accuracy of the previous one. The ensemble prediction function of AdaBoost $H: X \to \{-1, 1\}$ is given by

$$H(\mathbf{x}) = sign\left(\sum_{m=1}^{M} \alpha_m h_m(\mathbf{x})\right) \quad (1)$$

where $\alpha_1, \ldots, \alpha_M$ is a set of weights from respective $h_1, \ldots, h_M$ models.

To build this model, we followed the pseudo-code below, varying the base models $h_i$ by those mentioned before

- Given $(x_1, y_1)$, where $x_i \in X, y_i \in \{-1, 1\}$
- Initialize: $D_1 = \frac{1}{n}$ for $i = 1, \ldots, n$
- For $m = 1, \ldots, M$
  - Train the weak learner using distribution $D_m$
  - Get the hypothesis $h_m : X \to \{-1, 1\}$
  - Aim: Select $h_m$ with lower weighted error.

$$\epsilon_m = Pr_i \sim D_t[h_m(x_i \neq y_i)]$$

  - Choose $\alpha_m = \frac{1}{2} \ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$
  - Update for $i = 1, \ldots, n$

$$D_{m+1} = \frac{D_m(i) exp(-\alpha_m y_i h_m)}{Z_m}$$

Where $Z_m$ is a normalization factor.

Then the output is given by the Equation (1).

Were choosen six type of statistical models to use as base learners in AdaBoosting, which follows:

- **K Nearest Neighbors (KNN)**, with the parameter k defined by tuning.
- **Linear Discriminant Analysis**.
- **Logistic Regression** in canonical form.
- **Neural Networks** with one perceptron.
- **Support Vector Machines** with the linear kernel.
- **Decision Trees** with just one split node (Stump Models).

Each model was applied to different datasets, that can be accessed in *UCI ML Data Repository*, to evaluate empirically the performance from each method. They were all a binary classification task, where $y_i \in \{-1, 1\}$. The validation technique used was the repeated holdout, with 30 repetitions and split ratio 70-30% of training-test.The performance metric obtained was the accuracy, once all datasets were balanced.

## 3. Results and Discussion

The main result is represented by the Figure 1, where it's possible to see a boxplot for the accuracy results from each round of the holdout split. To each AdaBoost model where generated 100 models of each classifier type.

As we can observe from Table 1, the best AdaBoost isn't always that which refers to the standard Stump Models, and specifically in the half of the cases he isn't the one with greatest accuracy. The Linear Discriminant Model as well the Logistic Regression preforms relativily well in some databeses with low dimensionality. However, the LDA for example, can't perform in some datasets where some covariates have an strong colinearity.

The base leaner that most appears, together with the Decision Tree Stumpo is the SVM, appearing 5 out of 10 times, followed by neural networks, LDA, and logistic that appear 4 times.

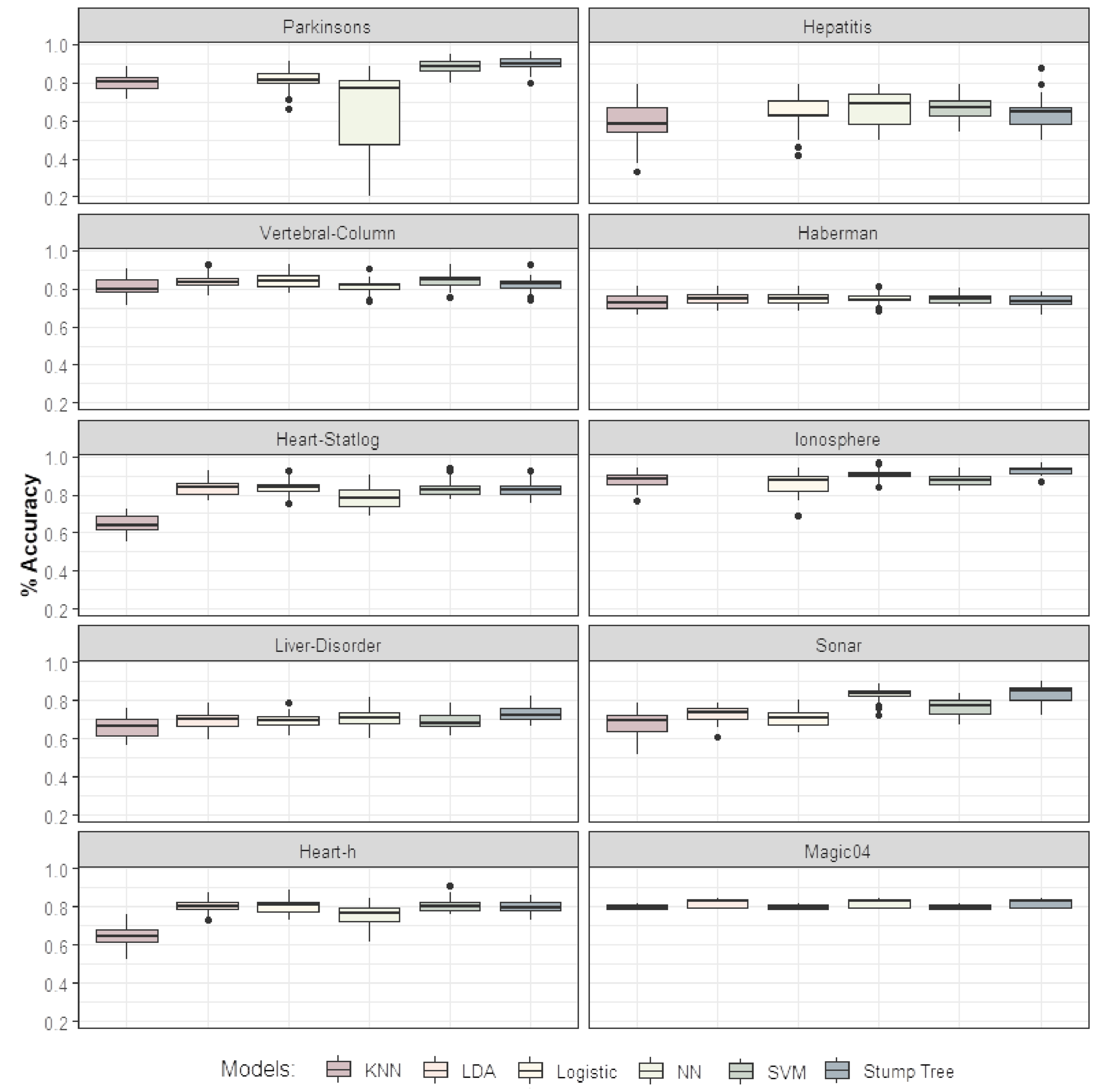


**Figure 1:** *AdaBoosting accuracy to all datasets and base learners.*

**Table 1:** *Table of Accuracy different AdaBoosting to all datasets*

| Datasets | KNN | LDA | Logistic | NN | SVM | Tree Stump |
|---|---|---|---|---|---|---|
| *hepatitis* | 0.58 ±0.11 | - | 0.64 ±0.09 | **0.66±0.08** | **0.66±0.07** | 0.65±0.09 |
| *parkinsons* | 0.80 ±0.11 | - | 0.81 ±0.06 | 0.65±0.24 | **0.89±0.04** | **0.89±0.04** |
| *sonar* | 0.68 ±0.16 | 0.73±0.04 | 0.71 ±0.05 | **0.83±0.04** | 0.78±0.05 | **0.83±0.04** |
| *heart-statlog* | 0.65 ±0.16 | **0.84±0.04** | **0.84 ±0.04** | 0.79±0.05 | 0.83±0.04 | 0.83±0.04 |
| *haberman* | 0.73 ±0.04 | **0.75±0.03** | **0.75 ±0.03** | **0.75±0.03** | **0.75±0.03** | 0.73±0.03 |
| *liver-disorder* | 0.66 ±0.05 | 0.70±0.05 | 0.69 ±0.04 | 0.71±0.04 | 0.68±0.04 | **0.73±0.04** |
| *ionosphere* | 0.88 ±0.04 | - | 0.86±0.05 | 0.90±0.03 | 0.87±0.03 | **0.93±0.02** |
| *vertebral-column* | 0.80 ±0.04 | **0.84 ±0.04** | **0.84±0.04** | 0.81±0.04 | **0.84±0.04** | 0.82±0.03 |
| *heart-h* | 0.64 ±0.05 | **0.80 ±0.04** | **0.80±0.04** | 0.76±0.04 | **0.80±0.03** | 0.79±0.03 |
| *magic04* | 0.81 ±0.01 | 0.79 ±0.01 | 0.79±0.01 | 0.83±0.01 | 0.79±0.01 | **0.83±0.01** |
| **All Datasets** | 0.72 ±0.10 | 0.78 ±0.06 | 0.78±0.08 | 0.77±0.11 | 0.79±0.08 | **0.81±0.09** |

## 4. Conclusion

The AdaBost can be defined as a powerful ensemble classifier formed by successively modelling a weak classifier to different weighted realizations of a data set. In this work we proposed a comparison between different base learners models used in AdaBoost, instead the standard Decision Tree Stumps, that already proof their predictive power in Schapire et. al, 2016, in order to study the efficiency of the others methods to predict correctly new observations, and create more discriminant classifiers to compose the AdaBoosting classifiers. We could observe that several methods improved or equated the standard AdaBoosting suggesting that's interesting to analyze in each situation what could be the better base learner to use. To futures works is important to try to combine multiples learners in a single AdaBoost model, and maybe change the hyperparameters iteratively in each model.

## References

FREUND, Yoav; SCHAPIRE, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, v. 55, n. 1, p. 119-139, 1997.

SALZBERG, Steven L. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. Machine Learning, v. 16, n. 3, p. 235-240, 199.

Rätsch, G., Onoda, T., & Müller, K. R. (2001). Soft margins for AdaBoost. Machine learning, 42(3), 287-320.

Schwenk, Holger, and Yoshua Bengio. "Boosting neural networks." Neural computation 12.8 (2000): 1869-1887.

Li, X., Wang, L., & Sung, E. (2008). AdaBoost with SVM-based component classifiers. Engineering Applications of Artificial Intelligence, 21(5), 785-795.

Reyzin, Lev, and Robert E. Schapire. "How boosting the margin can also boost classifier complexity." Proceedings of the 23rd international conference on Machine learning. ACM, 2006.