Implementation of Adaptative Boosting (AdaBoost) for the detecting Alzheimer’s at early stage

Nithin Sai Jalukuru
School of Engineering and Applied Sciences
University at Buffalo, Buffalo, NY, USA
njalukur@buffalo.edu

Abstract : In this present work, using an adaptive boosting classifier with 10-fold cross validation and fine-tuned hyper-parameters based on the different attributes of factors involved in identifying the symptoms of Alzheimer's, it is possible to classify whether or not a person has the disease. The concept behinsd the machine learning strategy known as "boosting" is to combine a number of relatively weak and inaccurate prediction rules to produce a highly accurate prediction rule. The Freund and Schapire AdaBoost algorithm, which has applications in many different fields, was the first real-world boosting algorithm and is still one of the most popular and extensively studied.

Keywords : Ensemble classification, Supervised machine learning, AdaBoost classifiers,

1.INTRODUCTION

Ensemble Machine learning Algorithms has a huge potential in Medical Industry in detecting many diseases at an very early stage. One of the breakthroughs in machine learning models is the ability to detect Alzheimer's.

Alzheimer's disease is the most common cause of dementia and is a neurodegenerative disorder with an unknown cause and pathogenesis that primarily affects older adults. Selective memory impairment is the earliest clinical sign of AD, and while there are treatments to lessen some symptoms, there is currently no known cure. Let's find out more about the Ada boosting Model. Ada Boost is an example of an ensemble method, so let's define an *ensemble method*.

Ensemble methods are **techniques that create multiple models and then combine them to produce improved results**. Ensemble methods usually produces more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods.

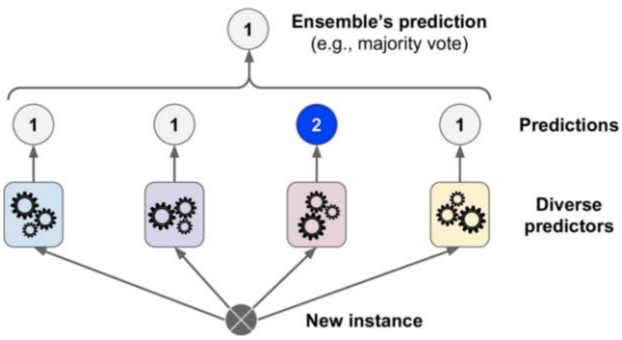

Fig 1: Representation of Ensemble Method (Src : Vitalflux)

Boosting is an **ensemble method** of converting weak learners into strong learners. Weak and strong refer to a measure how correlated are the learners to the actual target variable[^1]. Each training sample is used to train one decision tree unit in boosting and is selected using replacement over-weighted data. The trees will update the residuals error as they learn from their predecessors.

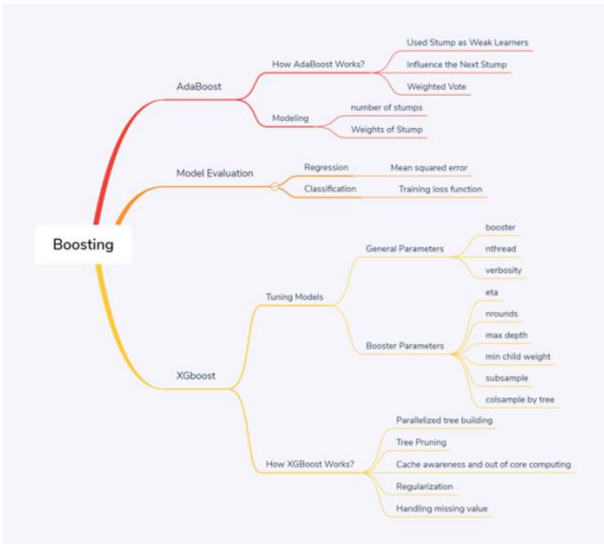

Fig 2: Boosting Overview (Source Algotech)

1.1 Adaptive Boosting

Adaboost (adaptive boosting) algorithm in 1995, which works through adjusting weight without the need of any priori knowledge on learner learning. Algorithms of AdaBoost.M1 、 AdaBoost.M2 、 AdaBoost.R, etc. were suggested based on later innovation. These algorithms could change voting-weights and solved many practical problems of the early Boosting algorithm.

Let’s look into Conceptual Working of AdaBoost. It is basically divided into 3 important concepts.

I. *Decision Stump as Weak Learners*

A classifier created by a weak learner is only marginally more accurate than random. a Weak classifier that succeeds in slightly exceeding 50% accuracy.

The term "weak classifier" can also refer to a "*weak learner*" or "base learner," and the idea is not limited to binary classification.

AdaBoost method employs one-level decision stumps for weak learners. Finding the best stump that can separate data by minimizing overall errors is the main goal when developing a weak classifier..

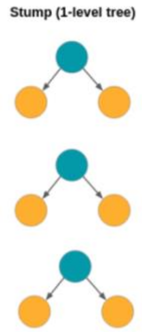

Fig 3. Rep of one level decision stump.

II. *Influence the next decision stump (Sequential)*

Contrary to bagging, sequential training is performed by boosting, and each weak learner is impacted by the previous decision stump. By assigning various weights to the data that will be used in the next stump-making process, Stump influences the subsequent stump. This weighting is based on calculations of error; if a data point was predicted incorrectly in the first stump, it will be given more weight in the subsequent stump-making process.

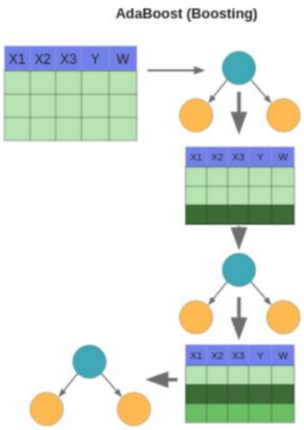

Fig 4: Sequential training in AdaBoost

III. *Weighted vote*

The weight assigned to each stump varies depending on the calculated error rate. The weight of a stump increases with the size of the errors it produces. Each stump's weight is taken into account when casting a vote, and the class with the highest total weight will be chosen as the winner.

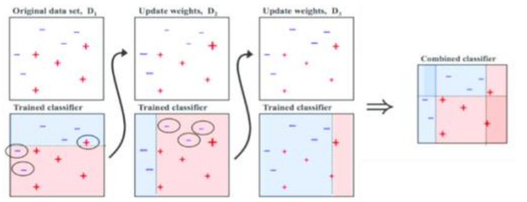

Fig 5: Rep of Ada Boost process (Src : Medium)

1.2 AdaBoost Algorithm

From above conceptual working of adaboost , let's see how the algorithm is built for Adaptive boosting.

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Fig 6: Boosting Algorithm AdaBoost

Pseudocode for AdaBoost is shown in Fig. 6. Here we are given $\mathcal{M}$ labeled training examples $(x_1, y_1) \ldots \ldots \ldots (x_m, y_m)$, where the x_i's are in some domain χ and the labels $y_i \in \{-1, +1\}$ 2. On each round$t = 1, \ldots \ldots, T$, a distribution D_t is computed as in the figure over the m training examples, and a given weak learner or weak learning algorithm is applied to find a weak, where the aim of the weak learner is to find a weak hypothesis with low weighted error relative to D_t . The final or combined hypothesis H computes the sign of a weighted combination of weak hypotheses.

1.3 Advantages of AdaBoost

I. Adaboost is less prone to overfitting as the input parameters are not jointly optimized
II. The accuracy of weak classifiers can be improved by using Adaboost
III. Instead of just binary classifications , Adaboost is used to classify text and images.
IV. There are many base estimators that can be used with AdaBoost (E.g: SVC)

1.4 Limitations of AdaBoost

I. AdaBoost needs the quality Dataset
II. This model is very sensitive to outliers and noisy data.

2. EXPERIMENTATION

For the Backpropagation model implementation, a data pipeline structure is constructed as illustrated in the accompanying figure 4.

Data collection and loading into the environment are both parts of data extraction After loading, the data has been labeled and cleaned so that the ML model can analyze it effectively. In addition, feature extraction a few features/attributes have been extracted, and the most pertinent ones have been chosen to train the model. The correct ML model has been selected, trained, and verified during the model selection, training, and validation processes by maximizing the more suitable assessment criteria.

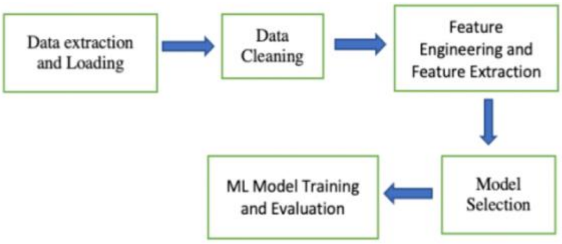

Fig 7: Data Pipeline

2.1 Data Extraction

Not a disease, **dementia** is a syndrome. A syndrome is a collection of symptoms for which there is no clear-cut cause. Memory and reasoning are two mental cognitive functions that are impacted by the symptoms of dementia. Alzheimer's disease may be classified as having dementia, which is a general term. It can be brought on by a number of illnesses, Alzheimer's disease being the most prevalent.

Longitudinal MRI data is collected from Kaggle. This data is a collection of 150 subjects aged 60 to 96.Each subject was scanned on two or more visits, separated by at least one year for a total of 373 imaging sessions.

Data consists of 15 attributes,

I. **Subject_ID**: Unique Id of the patient, Numerical Attribute
II. **MRI_ID:** Unique Id generated after conducting MRI on patient, Numerical Attribute
III. **Group**: Converted(Previously normal but developed dementia later), Demented and Non-Demented (Normal Patients) , categorical and target feature
IV. **Visit** : Number of visits to detect dementia status, Numerical attribute.
V. **MR_Delay**: numerical Attribute
VI. **M/F:** Represents Gender, M or F. Categorical variable.
VII. **Hand** : All the samples are right-handed (R), Categorical Variable.
VIII. **Age** : Age of the patient, ranging from 60 to 96
IX. **EDUC** : Years of Education, Numerical variable
X. **SES:** Socio Economic status, classified into categories from 1 to 5
XI. **MMSE** : Mini Mental state examination score, rages from 0 (Worst) to 30 (Best)
XII. **CDR** : Clinical Dementia Rating , 0 is no dementia, 0.5 is very mild, 1 is mild and 2 is moderate AD
XIII. **eTIV:** Estimated total intracranial volume, mm^3
XIV. **nWBV** : Normalized whole-brain volume.
XV. **ASF** : Atlas scaling factor (unitless)

2.2 Data Cleaning

I. Column names had been corrected with right *format* (As names were separated by a space)
II. Features :Subject_ID, MRI_ID, Hand are *dropped* (No useful information for model)
III. *Missing values* found in SES (19) and MMSE (2), These observations have been removed, since this is medical data , It's better to removed them.
IV. *Treated outliers* (Since AdaBoost is sensitive to noise and outliers).
V. Performed one hot encoding on M/F feature and created 2 features separately for Male and Female.
VI. Performed standardization on numerical features (To have mean =1 and Standard Deviation to 0)

2.3 Exploratory Data Analysis (EDA)

I. Distribution of Group (Target Variable):

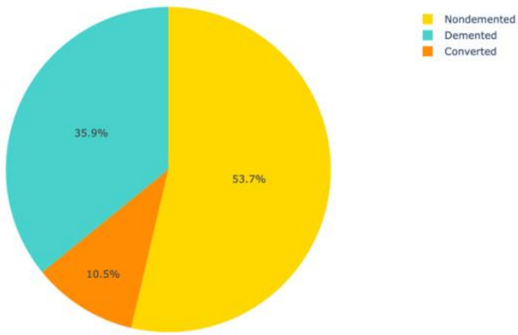

Fig 8: Distribution of Group feature

In the data, there are about 36% demented cases; the majority of the cases involve normal patients (non demented), and about 10.5% involve cases that have been converted.

II. Data Alignment:

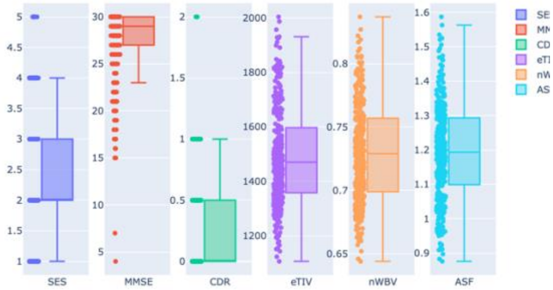

Fig 9: Data distribution

Box plots are plotted for identification of outliers and treating them.

III. MMSE Analysis.

From the fig , it is evident that majority of cases of *normal cognitive impairment* whereas very few cases of Mild, Moderate and Severe cognitive Impairment.

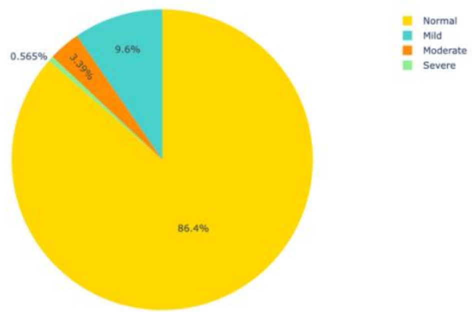

Fig 10: MMSE distribution

IV. Group with respect to Age Kde plot

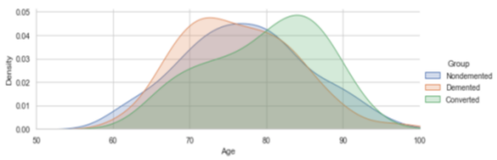

Fig 11 :Kde plot

From the fig 11, it can be concluded that age group from 70 to 85 are most prone to the AD symptoms.

V. Correlation Plot

	Visit	MR Delay	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	A
Visit	1.000000	0.922833	0.186306	0.013574	-0.051622	-0.029031	-0.015544	0.128745	-0.123453	-0.133
MR Delay	0.922833	1.000000	0.204313	0.044399	-0.030813	0.073640	-0.066391	0.119383	-0.102139	-0.125
Age	0.186306	0.204313	1.000000	-0.024977	-0.046857	0.055880	-0.022759	0.040182	-0.526316	-0.031
EDUC	0.013574	0.044399	-0.024977	1.000000	-0.722647	0.184459	-0.129440	0.267933	-0.019822	-0.251
SES	-0.051622	-0.030813	-0.046857	-0.722647	1.000000	-0.149219	0.076160	-0.261575	0.090095	0.255
MMSE	-0.029031	0.073640	0.055880	0.184459	-0.149219	1.000000	-0.705962	-0.019439	0.341381	0.027
CDR	-0.015544	-0.066391	-0.022759	-0.129440	0.076160	-0.705962	1.000000	0.052361	-0.350086	-0.063
eTIV	0.128745	0.119383	0.040182	0.267933	-0.261575	-0.019439	0.052361	1.000000	-0.206668	-0.989
nWBV	-0.123453	-0.102139	-0.526316	-0.019822	0.090095	0.341381	-0.350086	-0.206668	1.000000	0.211
ASF	-0.133897	-0.125088	-0.031783	-0.251677	0.255576	0.027745	-0.063413	-0.989030	0.211150	1.000

Fig 12 : Correlation plot

As seen in Fig. 12, there is a strong correlation between ASF and eTIV (-0.989), as well as between SES and EDUC.

The features M, F, Age, EDUC, SES, MMSE, eTIV, nWBV, and ASF are taken into account for data modeling, and the data is divided into 80 and 20 for training and testing.

2.4. AdaBoost Classifier Modelling

AdaBoost classifier has been fine-tuned with n_estimators ranging from 1 to 20 instep of 2 (combining 2 trees) and learning rate ranging from 0.0001 to 1 for training the classifier and performed the 10-fold cross validation on the training model . The best accuracy on validation set is around 75%, with Best M of 7 and learning rate of 1.

```
Best accuracy on validation set is: 0.7566502463054187
Best parameter of M is:  7
best parameter of LR is:  1
```

Fig 13: Validation and Tuning results

Finally, this classifier is ready for predicting testing data and see the testing results in following section

2.5 Model Evaluation Metrics

Overall accuracy, recall, precision, f1 score, and AUC of the ROC curve are the most common evaluation measures for classification models.

False positives (FP) and false negatives (FN) are outcomes that were mistakenly classified by the model, while true positives (TP) and true negatives (TN) are outcomes of the positive class and negative class, respectively.

a) *Overall Accuracy (OA):* This is defined by the following the equation

$$\mathrm{OA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}$$

The model could attain almost perfect overall accuracy if it consistently predicts the majority of classes.

b) The issue is more severe the more unbalanced the data. So, we require additional measurements. include Recall. It measures the proportion of accurately predicted positive classes to all positively categorized items.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Recall is important when we believe False Negatives are more important than False Positives

c) *Precision*: It is the ratio of correctly predicted positive classes to all items predicted to be positive

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

It tells us how correct or precise that our model's positive predictions are. When we think False Positives are more significant than False Negatives, precision is crucial.

d) *F1- Score* : The F1-score is a single performance statistic that considers both recall and precision. It is also frequently referred to as the F-Measure. It is calculated by averaging the two metrics harmonically.

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

with values closer to one indicating better performance, and values closer to zero indicating poorer performance

2.6 Test Results

Let's look at the classification report

	precision	recall	f1-score	support
Converted	0.33	0.10	0.15	10
Demented	0.78	0.78	0.78	23
Nondemented	0.71	0.84	0.77	38
accuracy			0.72	71
macro avg	0.61	0.57	0.57	71
weighted avg	0.68	0.72	0.69	71

Fig 14: Classification Report

The Model has an overall accuracy of 72% with a good tradeoff between precision (It talks about how many retrieved items are relevant) and recall (It talks about how many relevant items are retrieved).

Also, let's Look at Confusion Matrix,

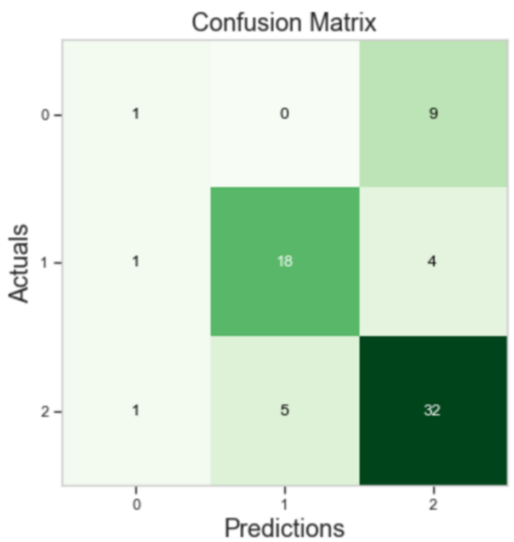

Fig 15: Confusion Matrix

One inference that can be noticed is that, for 5 cases the model has predicted that the patient has AD symptoms (Demented), but actually they don't have any symptoms of AD, similarly for 4 cases, model has predicted that the patient is normal, but patient is actually prone to AD symptoms (Demented), Which is a huge concern for the model.

Also implemented the AdaBoost classifier with SVC as a base estimator, and the results are even worse. *Model Accuracy with SVC base estimator is around 53%.*

3 FURTHER IMPROVEMENTS

This model can be further improved by more sophisticated EDA process with larger data samples. This score can be raised by experimenting with more fine-tuning the hyper parameters and with different base estimators.

4 REFERENCES

1. CSE 474/574 Lecture Slides, Prof Chen
2. AdaBoost Classifier Tutorial, Kaggle
3. *TU Chengsheng* at. El., " Adaboost is being used to classify text and images rather than binary classification problems"
4. ***Robert E. Schapire,*** "Explaining AdaBoost
5. Boosting algorithm (Algorithm and XGboost) , Algotech Blog