FROM DECISION TREE TO ADABOOST

WHAT WE INTRODUCED ?

TREES & RANDOM FOREST

Characteristics and applications of trees and forest

CODE

How to predict Home Credit Default Risk using AdaBoost?

BACKGROUND

Can you predict how capable each applicant is of repaying a loan?

ADABOOST

What is AdaBoost?
What are the pros and cons of AdaBoost?

01 INTRODUCTION

- •Data chosen: Home credit default risk
- •File chosen: **application_{train|test}.csv**
- •Has 124 relevant rows in **HomeCredit_columns_description.csv**
- **link:https://www.kaggle.com/competitions/home-credit-default-risk/data?select=application_test.csv**

Home credit csv file

	Table	Row	Description	Special
1	application_{train|test}.csv	SK_ID_CURR	ID of loan in our sample	
2	application_{train|test}.csv	TARGET	Target variable (1 - client with payment difficulties: he/she had late payment r	
5	application_{train|test}.csv	NAME_CONTRACT_TYPE	Identification if loan is cash or revolving	
6	application_{train|test}.csv	CODE_GENDER	Gender of the client	
7	application_{train|test}.csv	FLAG_OWN_CAR	Flag if the client owns a car	
8	application_{train|test}.csv	FLAG_OWN_REALTY	Flag if client owns a house or flat	
9	application_{train|test}.csv	CNT_CHILDREN	Number of children the client has	
10	application_{train|test}.csv	AMT_INCOME_TOTAL	Income of the client	
11	application_{train|test}.csv	AMT_CREDIT	Credit amount of the loan	
12	application_{train|test}.csv	AMT_ANNUITY	Loan annuity	
13	application_{train|test}.csv	AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given	
14	application_{train|test}.csv	NAME_TYPE_SUITE	Who was accompanying client when he was applying for the loan	
15	application_{train|test}.csv	NAME_INCOME_TYPE	Clients income type (businessman, working, maternity leave,...)	
16	application_{train|test}.csv	NAME_EDUCATION_TYPE	Level of highest education the client achieved	
17	application_{train|test}.csv	NAME_FAMILY_STATUS	Family status of the client	
18	application_{train|test}.csv	NAME_HOUSING_TYPE	What is the housing situation of the client (renting, living with parents, ...)	
19	application_{train|test}.csv	REGION_POPULATION_RELATIVE	Normalized population of region where normalized	
20	application_{train|test}.csv	DAYS_BIRTH	Client's age in days at the time of applic time only relative to the application	
21	application_{train|test}.csv	DAYS_EMPLOYED	How many days before the application t time only relative to the application	

Home credit csv file

Also because, used in a number of research papers...

Financial fraud detection model: Based on random forest
C Liu, Y Chan, SH Alam Kazmi, H Fu - ... of economics and finance, 2015 - papers.ssrn.com
... In this study, we introduced Random Forest (RF) for financial ... partial correlation analysis and Multidimensional analysis. The ... models and concluded that Random Forest has the highest ...
Save Cite Cited by 119 Related articles All 11 versions

Forecasting stock index movement: A comparison of support vector machines and random forest
M Kumar, M Thenmozhi - Indian institute of capital markets 9th ..., 2006 - papers.ssrn.com
... Recently, a support vector machine (SVM), and random forest regression based ... in finance. There are few studies for the application of SVM and random forest regression in financial ...
Save Cite Cited by 204 Related articles

Online supply chain financial risk assessment based on improved random forest
H Zhang, Y Shi, J Tong - Journal of Data, Information and Management, 2021 - Springer
... This article applies the improved stochastic forest algorithm to online supply chain financial ... of the online supply chain financial risk assessment based on improved random forest. Data ...
Save Cite Cited by 10 Related articles

Predicting bank financial failures using random forest
Z Rustam, GS Saragih - ... Workshop on Big Data and Information ..., 2018 - ieeexplore.ieee.org
... introduce random forest to predict bank financial failures occurring as a result of the financial ... The aim of this research is to see how of the application and accuracy of random forest. All ...
Save Cite Cited by 14 Related articles All 2 versions

Page 2 of about 15,200 results (**0.06** sec)

[PDF] An **Adaboost** Algorithm Based Analysis Method of Nonstationary **Financial Data**

V Chang, T Li, Z Zeng - researchgate.net

... nonstationary **data**, and also demonstrate a feasible practice in **financial** trading. The **data** of future contract is used in our analysis. The future we test **Adaboost** algorithm is a contract ...

Save Cite Related articles

Financial prediction of real estate based on random forest

Z He, L Pan - ... Workshop on Advanced Algorithms and Control ..., 2022 - spiedigitallibrary.org

... , this algorithm has a big **advantage**, it combines the ability to ... **Adaboost** algorithm is implemented by changing the **data** ... This paper uses the **financial data** of Stock A real estate listed ...

Save Cite All 2 versions

AdaBoost models for corporate bankruptcy prediction with missing **data**

L Zhou, KK Lai - Computational Economics, 2017 - Springer

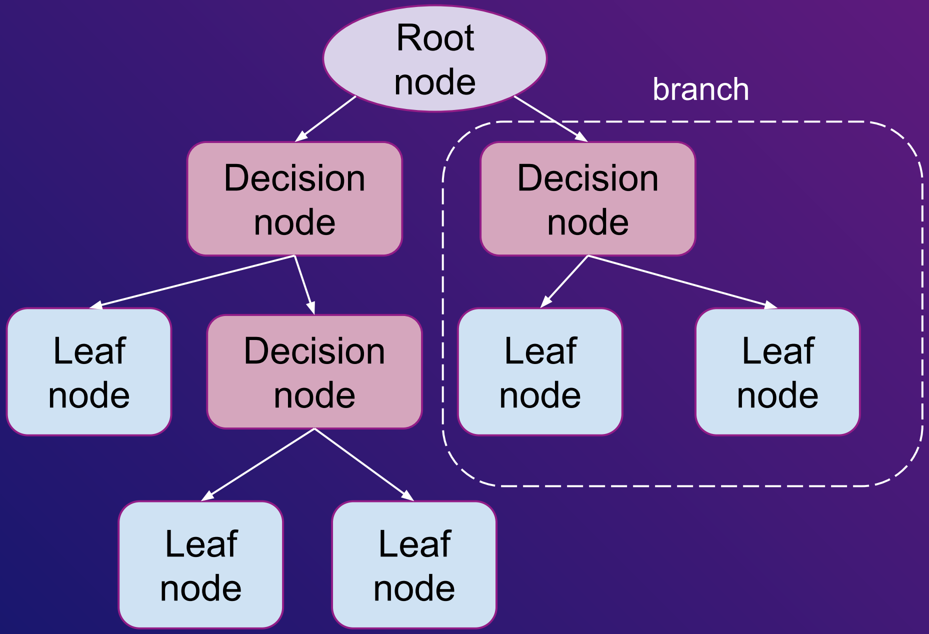
... -**financial** firms have been selected and bankruptcy prediction is based on the **financial data** ... How to take **advantage** of the whole **data** set in bankruptcy prediction. These two problems ...

Save Cite Cited by 33 Related articles All 6 versions

02

TREES AND RANDOM FOREST

DECISION TREES



In machine learning, a decision tree is a predictive model that represents a mapping relationship between attributes and values.

CHARACTERISTICS OF DECISION TREES

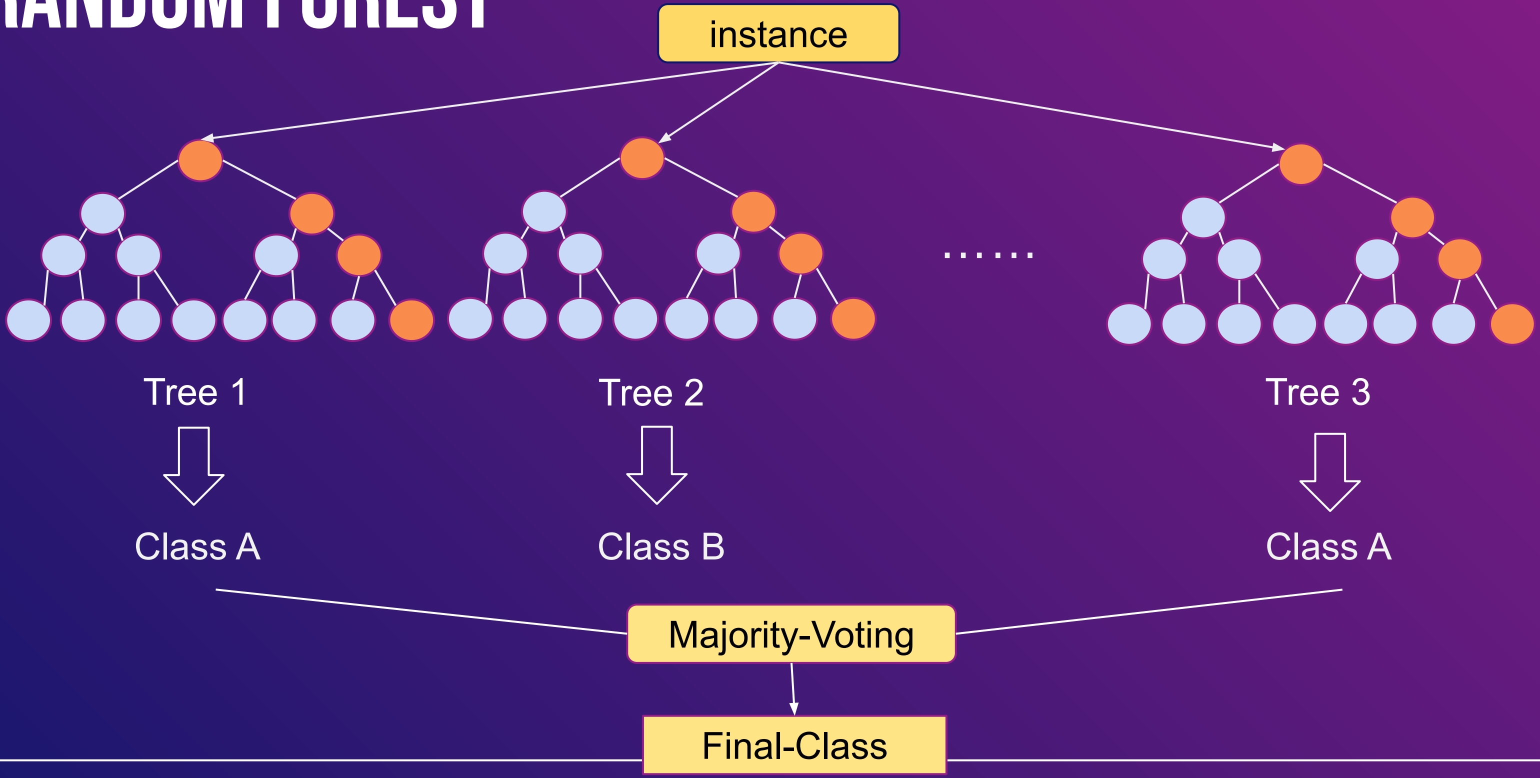
Advantage

- Simple to understand and to interpret
- Less Data Preparation
- Able to handle both numerical and categorical data.

Disadvantage

- Unstable
- Can cause overfitting

RANDOM FOREST



MAJORITY VOTING

The random forest classifier combines a number of decision trees to improve the accuracy of the classification.

The final output is determined by the mode of the classes of the individual tree output.

Majority rule is a principle that means the decision-making power belongs to the group that has the most members.

C1 C1 C1 C1 C1 C1 C1 C2 C2 C2 C3

APPLICATION OF RANDOM FOREST

Predict cardiovascular disease

Predict online buying behavior

Detect credit card fraud

...

CHARACTERISTICS OF RANDOM FOREST

Advantage

- Relatively high accuracy
- Stable
- Can process data with large number of features and samples
- Works well with both categorical and continuous variables
- Automatically handle missing values

Disadvantage

- Complexity
- Longer Training Period

03

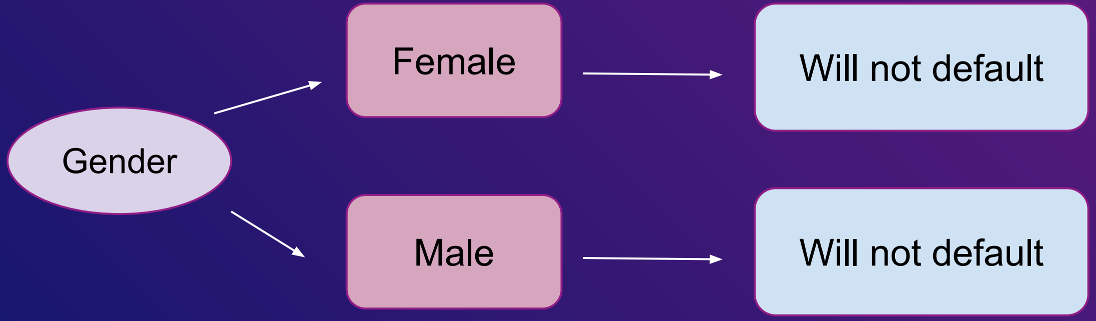
ADABOOST IMPROVMENT

SHORTCOMINGS OF TREE

$Default? \sim Gender$

```
> (tb.tree = table(tree.yhat, graph$TARGET))

tree.yhat      0      1
        0 282682  24825
        1      0      0
> Accuracy <- sum(diag(tb.tree))/sum(tb.tree)
> Accuracy
[1] 0.9192701
```

```
> table(graph$GENDER)

     F      M    XNA
202448 105059      4
```



BOOSTING - A BETTER ENSEMBLE LEARNING

Random Forest – Bagging

1. Choose a part of samples randomly

2. All the models are generate the same time

3. Take samples with same weights

4. Each model has the same weight

Adaboost

1. Can we use the whole data set?

2. Can we generate models based on the previous one?

3. Can we give more weights to difficult samples?

4. Can we give different weights to models?

BOOSTING - A BETTER ENSEMBLE LEARNING

Random Forest – Bagging

Adaboost

1. Choose a part of samples randomly

1. Can we use the whole data set?

In sample selection: Training set for adaboost is the same, only the weight of each sample is changing

2. All the models are generate the same time

2. Can we generate models based on the previous one?

In the order of calculation: The classify function for adaboost must be generated sequentially

3. Take samples with same weights

3. Can we give more weights to difficult samples?

In the sample weights: Adaboost adjusts the sample weights if error occurred in previous model

4. Each model has the same weight

4. Can we give different weights to models?

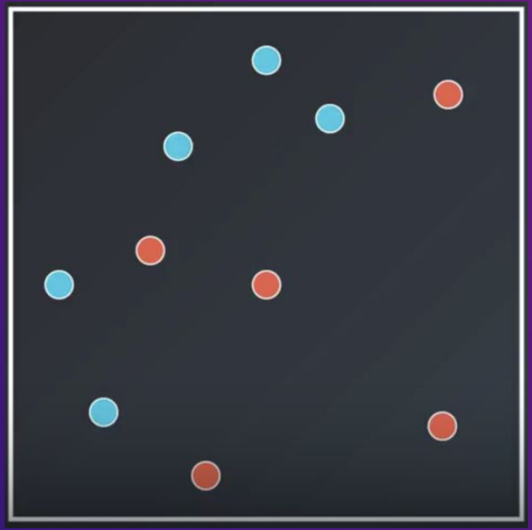
In the prediction function: The weights for predictor function in adaboost changed based on the error rate

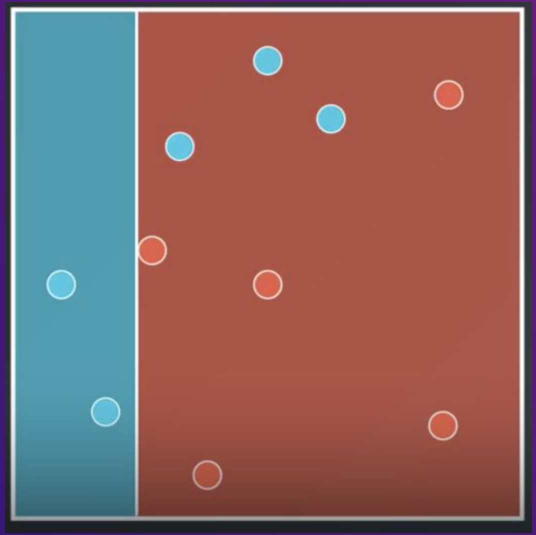
ADAPTIVE BOOSTING

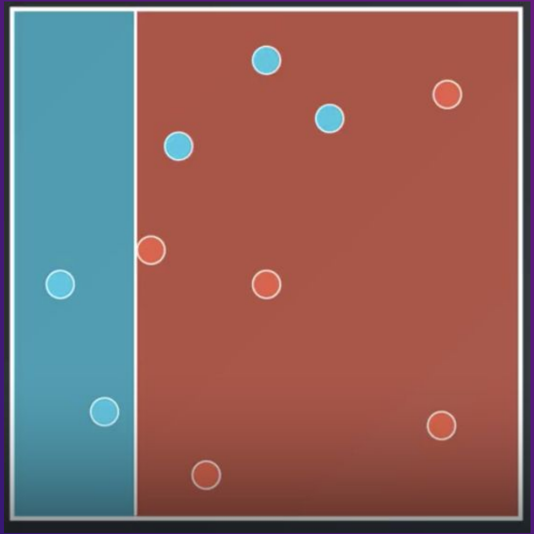
- The weak learners in AdaBoost are decision trees with a single split, called decision stumps.
- Each stump chooses a feature, say X2, and a threshold, T, and then splits the examples into the two groups on either side of the threshold.
- Sequential updating of weights on data points
- Form a final model from weak learners

BAGGING VS BOOSTING

Random Forest – Bagging	**Adaboost – Boosting**
Chose sample for each tree	All data set were trained More probability to drawn misclassification sample
Majority of trees leads to the answer	Use weights when combining trees





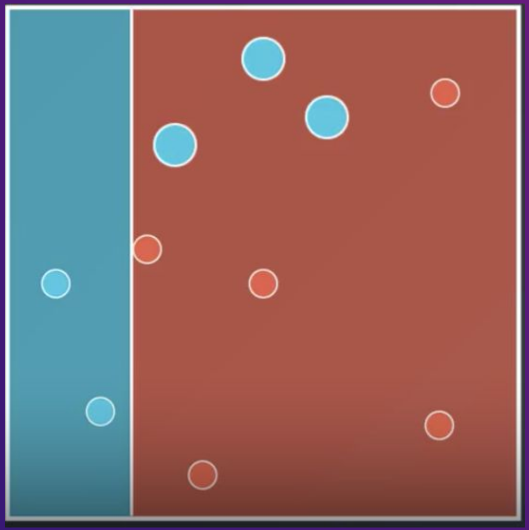


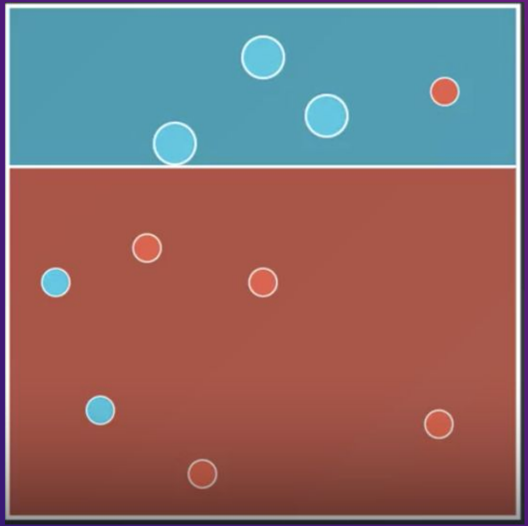
ϵ : *error rate in current model*

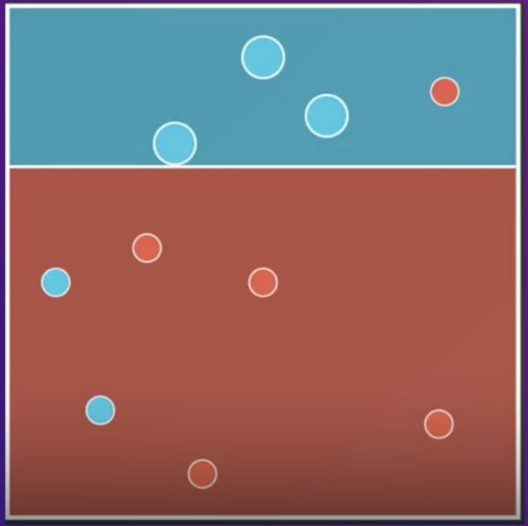
$$\omega_i^{new} = \omega_i^{old} \text{ for the correct points}$$

$$\omega_i^{new} = \frac{1-\epsilon}{\epsilon}\omega_i^{old} \text{ for the wrong points}$$

Then normalize ω_i





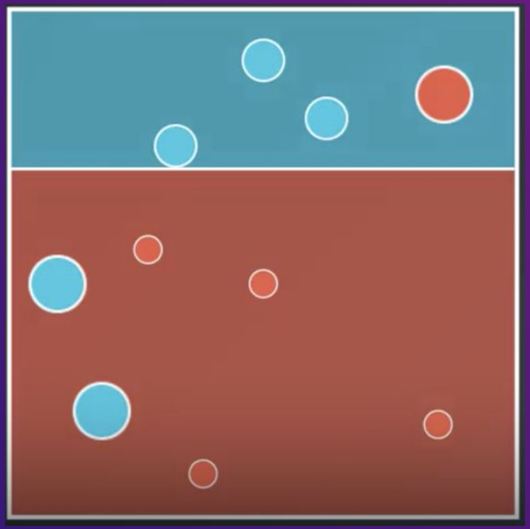


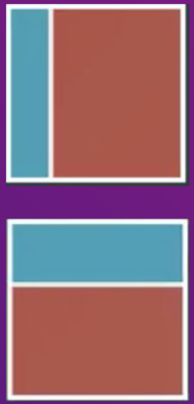
ϵ : *error rate in current model*

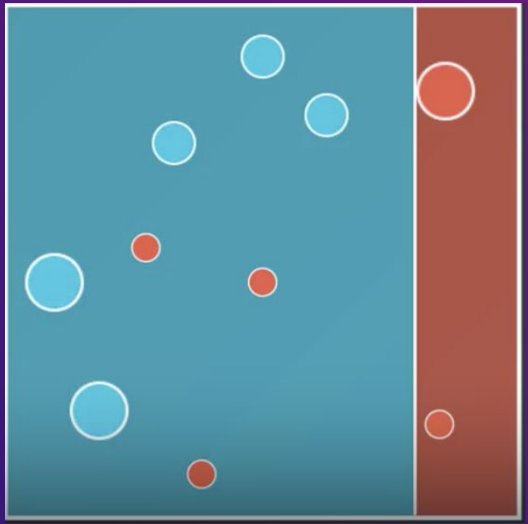
$\omega_i^{new} = \omega_i^{old}$ *for the correct points*

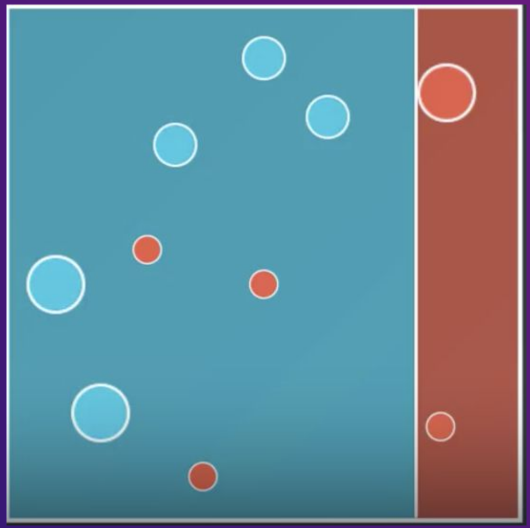
$\omega_i^{new} = \frac{1-\epsilon}{\epsilon}\omega_i^{old}$ *for the wrong points*

Then normalize ω_i





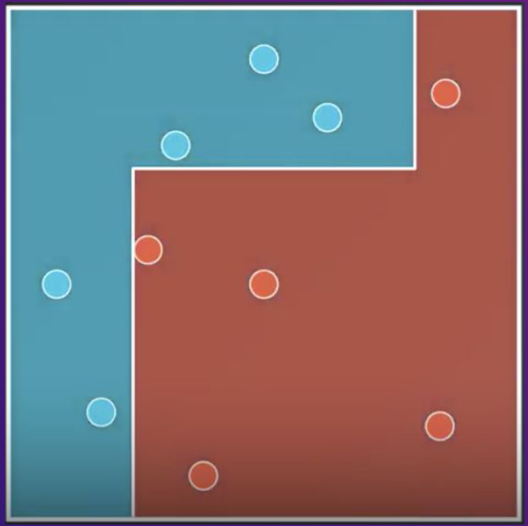


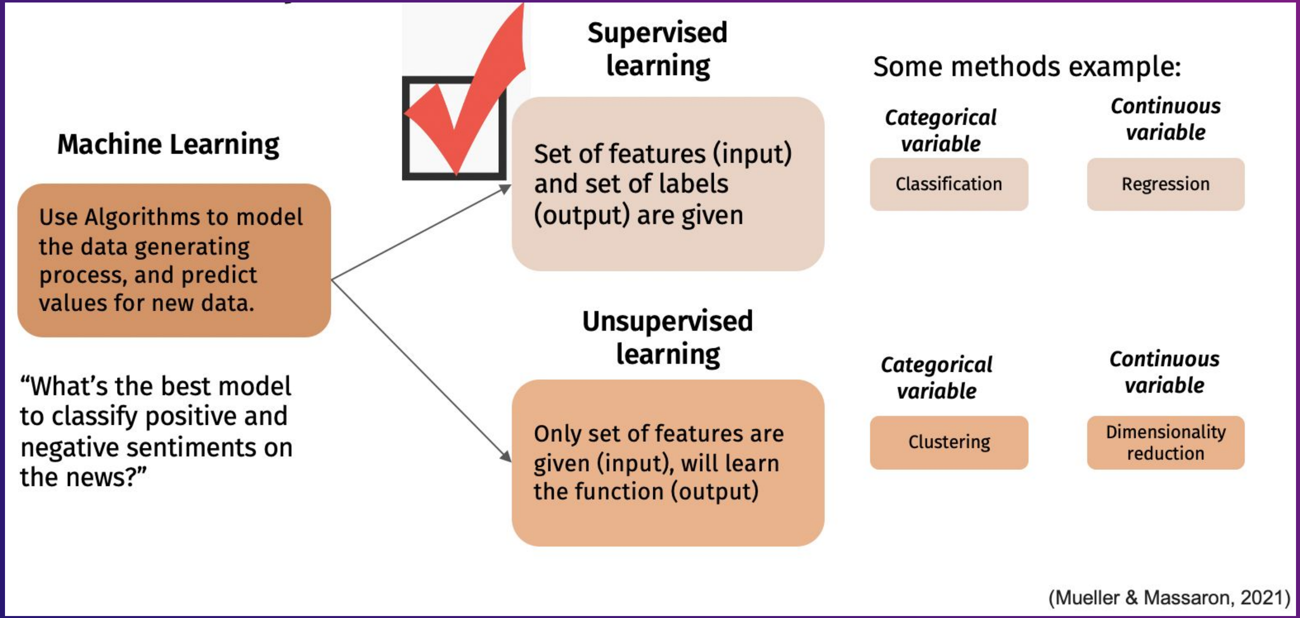


$\epsilon_k : error\ rate\ for\ k^{th} model$

$$\alpha_k = log(\frac{1-\epsilon_k}{\epsilon_k})$$

$$final\ model = \Sigma \alpha_i * weak\ learner_i$$





From presentation group 9

04 UNDERSTANDING ADABOOST

PSEUDOCODE

$(x_1, y_1), \dots, (x_n, y_n)$, x is predictor and $y \in \{-1,1\}$ is response

t is number of iteration

$C_t(x)$ is a weak classifier trained in iteration t

w_i is the weight of observation $i \in (1, \dots, n)$

$\boldsymbol{w}$ is the column vector $[w_1 \quad w_2 \quad \dots \quad w_{n-1} \quad w_n]^T$

α_t is the model $C_t(x)$ weighting

$I()$ is the indicator variable function (output vector for simplicity)

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125	M	Y	Y	-1
0.125	M	Y	N	1
0.125	M	Y	N	-1
0.125	M	N	Y	1
0.125	F	N	N	1
0.125	F	Y	Y	-1
0.125	F	Y	Y	-1
0.125	F	N	Y	-1

$\boxed{\text{Initialize } w_i = \frac{1}{n} \text{ for all } i \in \{1, \ldots, n\}}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T \, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

T
2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125	M	Y	Y	-1
0.125	M	Y	N	1
0.125	M	Y	N	-1
0.125	M	N	Y	1
0.125	F	N	N	1
0.125	F	Y	Y	-1
0.125	F	Y	Y	-1
0.125	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T\, I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it

t	T
1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125	M	Y	Y	-1
0.125	M	Y	N	1
0.125	M	Y	N	-1
0.125	M	N	Y	1
0.125	F	N	N	1
0.125	F	Y	Y	-1
0.125	F	Y	Y	-1
0.125	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T \, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

INFORMATION GAIN (ID3)	
GENDER	0.0484
CAR	**0.122**
HOUSE	0.122

t	T
1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125	M	Y	Y	-1
0.125	M	Y	N	1
0.125	M	Y	N	-1
0.125	M	N	Y	1
0.125	F	N	N	1
0.125	F	Y	Y	-1
0.125	F	Y	Y	-1
0.125	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

For $t = 1$ to T:

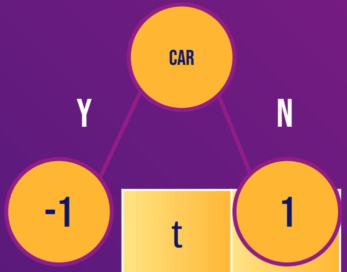
$\boxed{\text{Fit } C_t(x) \text{ and minimize error using weight } w_i}$

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

INFORMATION GAIN (ID3)	
GENDER	0.0484
CAR	**0.122**
HOUSE	0.122



t	
1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125	M	Y	Y	-1
0.125	M	Y	N	1
0.125	M	Y	N	-1
0.125	M	N	Y	1
0.125	F	N	N	1
0.125	F	Y	Y	-1
0.125	F	Y	Y	-1
0.125	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

For $t = 1$ to T:

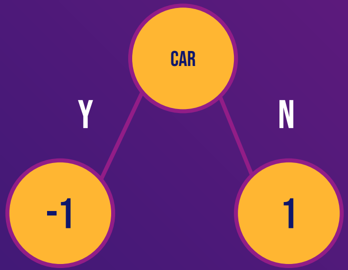
$\boxed{\text{Fit } C_t(x) \text{ and minimize error using weight } w_i}$

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T \, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

A1



t	T
1	2

EXAMPLE

W ERROR GENDER	CAR	HOUSE	TARGET
0.125 * 0 = 0	Y	Y	-1
0.125 * 1 = 0.125	Y	N	1
0.125 * 0 = 0	Y	N	-1
0.125 * 0 = 0	N	Y	1
0.125 * 0 = 0	N	N	1
0.125 * 0 = 0	Y	Y	-1
0.125 * 0 = 0	Y	Y	-1
0.125 * 1 = 0.125	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

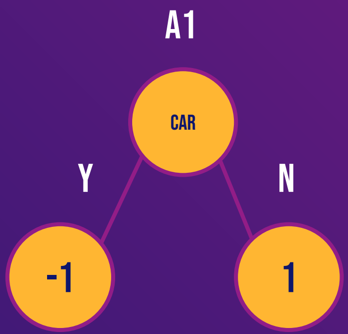
For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T \, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it



t	T
1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125	M	Y	Y	-1
0.125	M	Y	N	1
0.125	M	Y	N	-1
0.125	M	N	Y	1
0.125	F	N	N	1
0.125	F	Y	Y	-1
0.125	F	Y	Y	-1
0.125	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

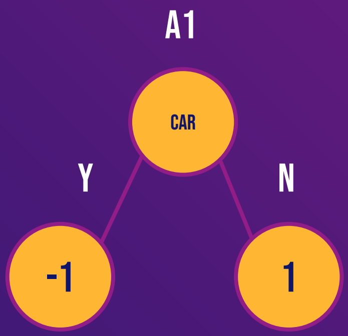
For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T\, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it



ERR	t	T
0.25	1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125	M	Y	Y	-1
0.125	M	Y	N	1
0.125	M	Y	N	-1
0.125	M	N	Y	1
0.125	F	N	N	1
0.125	F	Y	Y	-1
0.125	F	Y	Y	-1
0.125	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

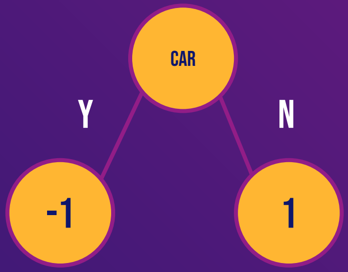
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it

A1



A1	t	T
ln(3)	1	2

EXAMPLE

W	GENDER CAR HOUSE W	TARGET
0.125	0.125 * 1 = 0.125	-1
0.125	0.125 * 3 = 0.375	1
0.125	0.125 * 1 = 0.125	-1
0.125	0.125 * 1 = 0.125	1
0.125	0.125 * 1 = 0.125	1
0.125	0.125 * 1 = 0.125	-1
0.125	0.125 * 1 = 0.125	-1
0.125	0.125 * 3 = 0.375	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

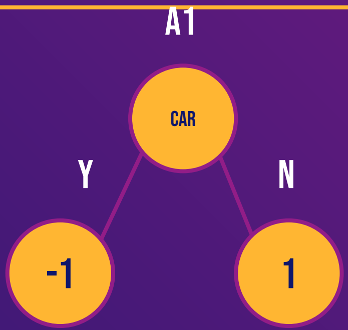
For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T\, I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

$\boxed{\text{Update } w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))} \text{ and normalize it}}$



A1	t	T
ln(3)	1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125				-1
0.125				1
0.125				-1
0.125				1
0.125				1
0.125				-1
0.125				-1
0.125				-1

W (SUM = 1.5)
0.125
0.375
0.125
0.125
0.125
0.125
0.125
0.375

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

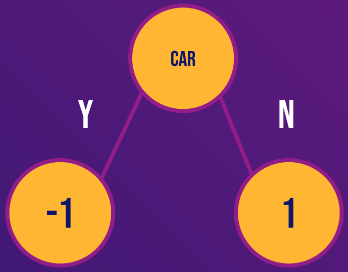
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T\, I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it

A1



A1	t	T
ln(3)	1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.125				-1
0.125				1
0.125				-1
0.125				1
0.125				1
0.125				-1
0.125				-1
0.125				-1

W
0.125/1.5 = 0.083
0.375/1.5 = 0.25
0.125/1.5 = 0.083
0.125/1.5 = 0.083
0.125/1.5 = 0.083
0.125/1.5 = 0.083
0.125/1.5 = 0.083
0.375/1.5 = 0.25

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

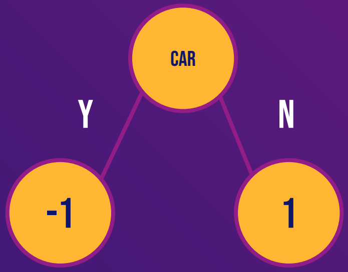
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T\, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

$\boxed{\text{Update } w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))} \text{ and normalize it}}$

A1



A1	t	T
ln(3)	1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.083	M	Y	Y	-1
0.25	M	Y	N	1
0.083	M	Y	N	-1
0.083	M	N	Y	1
0.083	F	N	N	1
0.083	F	Y	Y	-1
0.083	F	Y	Y	-1
0.25	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

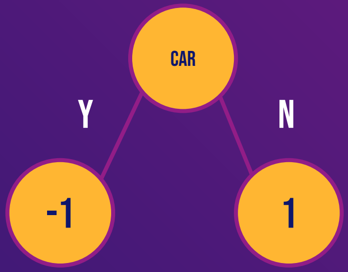
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it

A1



A1	t	T
ln(3)	1	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.083	M	Y	Y	-1
0.25	M	Y	N	1
0.083	M	Y	N	-1
0.083	M	N	Y	1
0.083	F	N	N	1
0.083	F	Y	Y	-1
0.083	F	Y	Y	-1
0.25	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

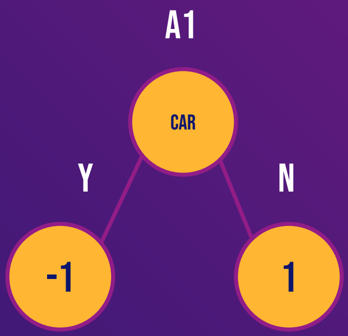
For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it



A1	t	T
ln(3)	2	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.083	M	Y	Y	-1
0.25	M	Y	N	1
0.083	M	Y	N	-1
0.083	M	N	Y	1
0.083	F	N	N	1
0.083	F	Y	Y	-1
0.083	F	Y	Y	-1
0.25	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T \, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(y \neq C_t(x))}$ and normalize it

INFORMATION GAIN (ID3)	
GENDER	0.244
CAR	0.049
HOUSE	**0.279**

A1	t	T
ln(3)	2	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.083	M	Y	Y	-1
0.25	M	Y	N	1
0.083	M	Y	N	-1
0.083	M	N	Y	1
0.083	F	N	N	1
0.083	F	Y	Y	-1
0.083	F	Y	Y	-1
0.25	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

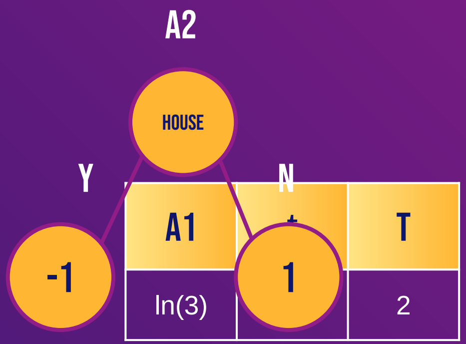
$\boxed{\text{Fit } C_t(x) \text{ and minimize error using weight } w_i}$

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(y \neq C_t(x))}$ and normalize it

INFORMATION GAIN (ID3)	
GENDER	0.244
CAR	0.049
HOUSE	**0.279**



EXAMPLE

W	GENDER ERROR CAR	HOUSE	TARGET
0.083	0.083 * 0 = 0	Y	-1
0.25	0.25 * 0 = 0	N	1
0.083	0.083 * 1 = 0.083	N	-1
0.083	0.083 * 0 = 0	Y	1
0.083	0.083 * 1 = 0.083	N	1
0.083	0.083 * 0 = 0	Y	-1
0.083	0.083 * 0 = 0	Y	-1
0.25	0.25 * 0 = 0	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

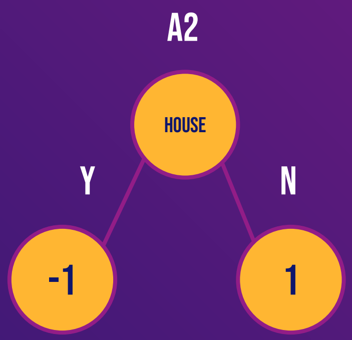
For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T\, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it



ERR	A1	t	T
0.17	ln(3)	2	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.083	M	Y	Y	-1
0.25	M	Y	N	1
0.083	M	Y	N	-1
0.083	M	N	Y	1
0.083	F	N	N	1
0.083	F	Y	Y	-1
0.083	F	Y	Y	-1
0.25	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

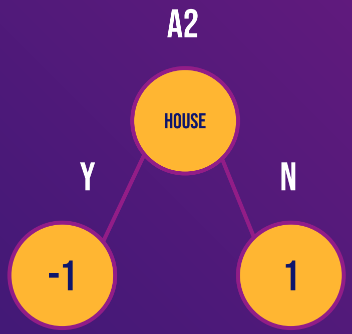
For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T\, I\left(\mathbf{y} \neq C_t(x)\right)$

$\boxed{\text{Compute } \alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)}$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it



A2	A1	t	T
ln(5)	ln(3)	2	2

EXAMPLE

W	GENDER CAR HOUSE W	TARGET
0.083	0.083 * 1 = 0.083	-1
0.25	0.25 * 1 = 0.25	1
0.083	0.083 * 5 = 0.417	-1
0.083	0.083 * 1 = 0.083	1
0.083	0.083 * 5 = 0.417	1
0.083	0.083 * 1 = 0.083	-1
0.083	0.083 * 1 = 0.083	-1
0.25	0.25 * 1 = 0.25	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

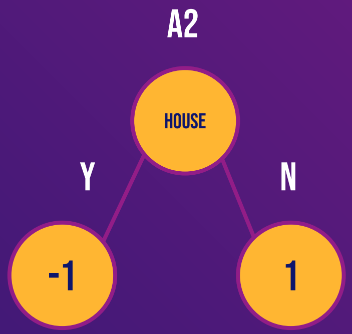
For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T\, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it



A2	A1	t	T
ln(5)	ln(3)	2	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
	W (SUM = 1.67)			
0.083	0.083 * 1 = 0.083			-1
0.25	0.25 * 1 = 0.25			1
0.083	0.083 * 5 = 0.417			-1
0.083	0.083 * 1 = 0.083			1
0.083	0.083 * 5 = 0.417			1
0.083	0.083 * 1 = 0.083			-1
0.083	0.083 * 1 = 0.083			-1
0.25	0.25 * 1 = 0.25			-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

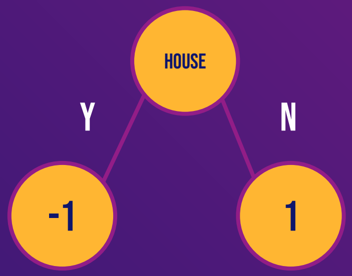
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T\, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

A2



A2	A1	t	T
ln(5)	ln(3)	2	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.083				-1
0.25				1
0.083				-1
0.083				1
0.083				1
0.083				-1
0.083				-1
0.25				-1

W (SUM = 1.67)
0.083
0.25
0.417
0.083
0.417
0.083
0.083
0.25

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

For $t = 1$ to T:

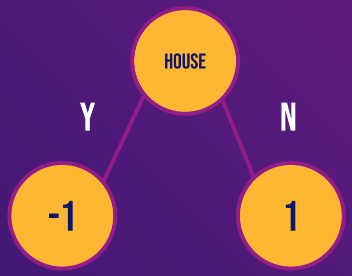
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T\, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

A2



A2	A1	t	T
ln(5)	ln(3)	2	2

EXAMPLE

W	GENDER CAR W HOUSE	TARGET
0.083	0.083/1.67 = 0.05	-1
0.25	0.25/1.67 = 0.15	1
0.083	0.417/1.67 = 0.25	-1
0.083	0.083/1.67 = 0.05	1
0.083	0.417/1.67 = 0.25	1
0.083	0.083/1.67 = 0.05	-1
0.083	0.083/1.67 = 0.05	-1
0.25	0.25/1.67 = 0.15	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

For $t = 1$ to T:

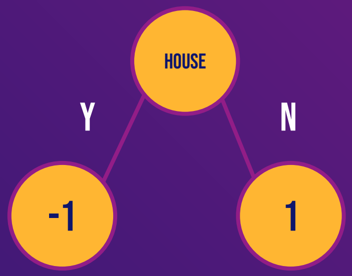
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it

A2



A2	A1	t	T
ln(5)	ln(3)	2	2

EXAMPLE

W	GENDER	CAR	HOUSE	TARGET
0.05	M	Y	Y	-1
0.15	M	Y	N	1
0.25	M	Y	N	-1
0.05	M	N	Y	1
0.25	F	N	N	1
0.05	F	Y	Y	-1
0.05	F	Y	Y	-1
0.15	F	N	Y	-1

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

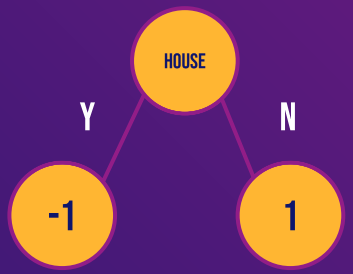
Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T \, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\boldsymbol{y} \neq C_t(x))}$ and normalize it

A2



A2	A1	t	T
ln(5)	ln(3)	2	2

PSEUDOCODE (TESTING)

$$C(x) = sign(\sum_{t=1}^{T} \alpha_t C_t(x))$$

EXAMPLE

GENDER	CAR	HOUSE	TARGET
F	Y	N	?

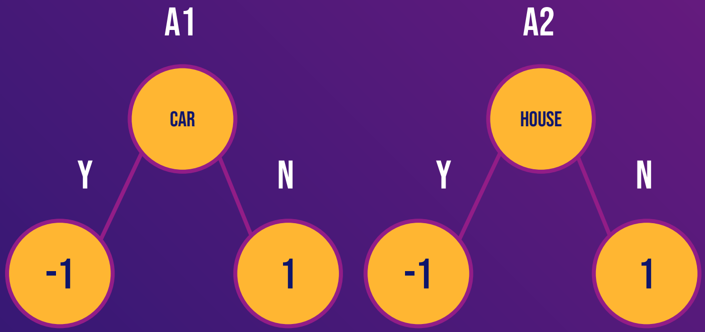
$$sign\big(\alpha_1 C_1(Gender = M, Car = Y, House = N) + \alpha_2 C_2(Gender = M, Car = Y, House = N)\big)$$

$$= sign\big(\ln(3)\,(-1) + \ln(5)\,(1)\big)$$

$$= sign(-1.09 + 1.61)$$

$$= sign(0.51)$$

$$= 1$$



A1	A2
ln(3)	ln(5)

ALGORITHM COMPLEXITY

- Given:
  - T(X) – complexity of training for weak learner
  - t(X) – complexity of testing for weak learner
  - τ – number of iteration
  - n – number of samples
  - p – number of predictors

- Training Phase for Adaboost: O(τT(X)+τn)
- Testing Phase for Adaboost: O(τt(X))

- Weak Learner of Decision Tree with depth = 1:
- Training Phase of weak learner: T(X) = O(np)
- Testing Phase of weak learner: t(X) = O(1)

- Training Phase = O(τnp) (why?)

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T\, I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it

$$C(x) = sign(\sum_{t=1}^{T} \alpha_t C_t(x))$$

05

OVERVIEW OF THE DATASET

FEATURES OF THE DATASET Home Credit Default Risk

- A Kaggle machine learning competition
- Behavioral Science related (predicting whether or not a client will repay a loan or have difficulty)
- Large sample size
- Imbalanced data
- Many predictor variables
- Adaboost performs better

06

SUMMARY OF ADABOOST

ADVANTAGES OF ADABOOST

High precision (greatly improve the accuracy of the decision tree, comparable to SVM).

The weight of each classifier fully considered by AdaBoost (relative to Bagging algorithm and Random Forest algorithm).

Various methods to build sub-classifiers (AdaBoost provides a framework).

Good use of weak classifiers for cascading.

Simple, efficient, easy to write and almost no overfitting.

No parameters to adjust during the training process.

LIMITATIONS OF ADABOOST

Training is time-consuming (reselect the best segmentation point for the current classifier each time).

Classification accuracy drops due to data imbalance.

The number of AdaBoost iterations (i.e.he number of weak classifiers) **is not easy to set. Cross-validation can be used to make the determination.**

Sensitive to noisy data and anomalous data.

APPLICATION SCENARIOS OF ADABOOST

For binary or multi-category scenarios

Baseline for classification tasks (simple, no overfitting, no need to adjust the classifier)

For feature selection (feature selection)

Correction the bad case (only need to add a new classifier, no need to change the original classifier)

07 UNDERSTANDING CODE

THE CODE

- Ensure that you have put the “application_train.csv” and “application_test.csv” in the same location as .ipynb
- The code run for too long (more than 3 seconds) → it is normal
- If the library cannot be used, use “pip install” on anaconda prompt

THE CODE

```python
#library
import math

import pandas as pd #pandas
import numpy as np #numpy
import matplotlib.pyplot as plt

from sklearn.ensemble import AdaBoostClassifier as ada #Adaboost
from sklearn.metrics import confusion_matrix as cf #confusion matrix
from sklearn.metrics import roc_curve, auc #ROC and AUC calculation

from scipy.stats import rankdata
```

USEFUL FOR DATA ANALYTICS

MACHINE LEARNING RELATED

THE CODE

```
#loading the data
trainingSet = pd.read_csv("application_train.csv")
testingSet = pd.read_csv("application_test.csv")
```

LOAD ALL DATA

THE CODE

```
#Data preprocessing

#Filling Missing Data
trainingSet = trainingSet.fillna(0)
testingSet = testingSet.fillna(0)

#Splitting training and testing X Y
trainY = trainingSet["TARGET"]
trainX = trainingSet.drop(columns = ["SK_ID_CURR", "TARGET"])
testX = testingSet.drop(columns = ["SK_ID_CURR"])

#dummy variables
trainX = pd.get_dummies(trainX)
testX = pd.get_dummies(testX)
trainX, testX = trainX.align(testX, join = "inner", axis = 1)
```

EXAMPLE

OCCUPATION
Driver
Staff
Manager
Staff
Staff
Manager
Staff
Driver

OCCUPATION_DRIVER	OCCUPATION_STAFF	OCCUPATION_MANAGER
T	F	F
F	T	F
F	F	T
F	T	F
F	T	F
F	F	T
F	T	F
T	F	F

THE CODE

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \mathbf{w}^T\ I(\mathbf{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(\mathbf{y} \neq C_t(x))}$ and normalize it

```
#Setting up a class for Adaboost
base_estimator = None #default is 1 level desicion tree, change it to any classifier if you wish to
n_estimators = 20 #the max number of n the boosting needs to stop
random_state = 938 #PS938, "seed"
```

PARA FOR THE MODEL

```
model = ada(base_estimator = base_estimator, n_estimators = n_estimators, random_state = random_state) #setting up adaboost
```

INITIALIZING MODEL

THE CODE

```
#fit the model
a = model.fit(trainX,trainY)
```

TRAINING PHRASE

```
#do the prediction
trainY_pred = model.predict_proba(trainX)[:, 1]
```

TESTING PHRASE

THE CODE

```
#output the prediction for the submission
test_Y = model.predict_proba(testX)[:, 1]
```

```
test_Y = result2 >= q
df1 = pd.DataFrame(test_Y, columns=["TARGET"])
df2 = testingSet['SK_ID_CURR']
```

```
df3 = pd.concat([df2,df ], axis=1)
df3.to_csv("result.csv",index=False)
```

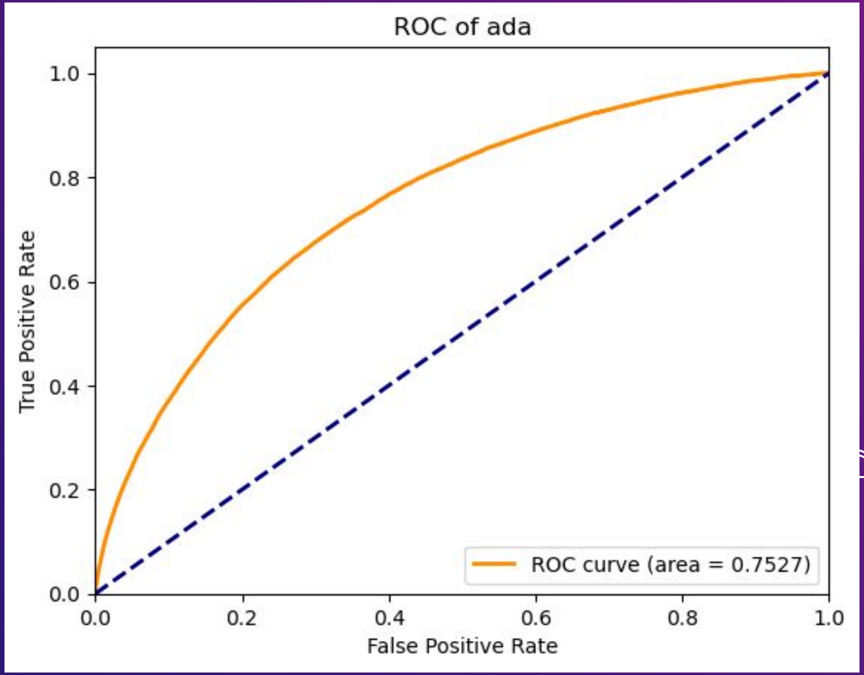
EXPORT RESULT TO THE COMPETITION FOR THE SCORE

THE CODE

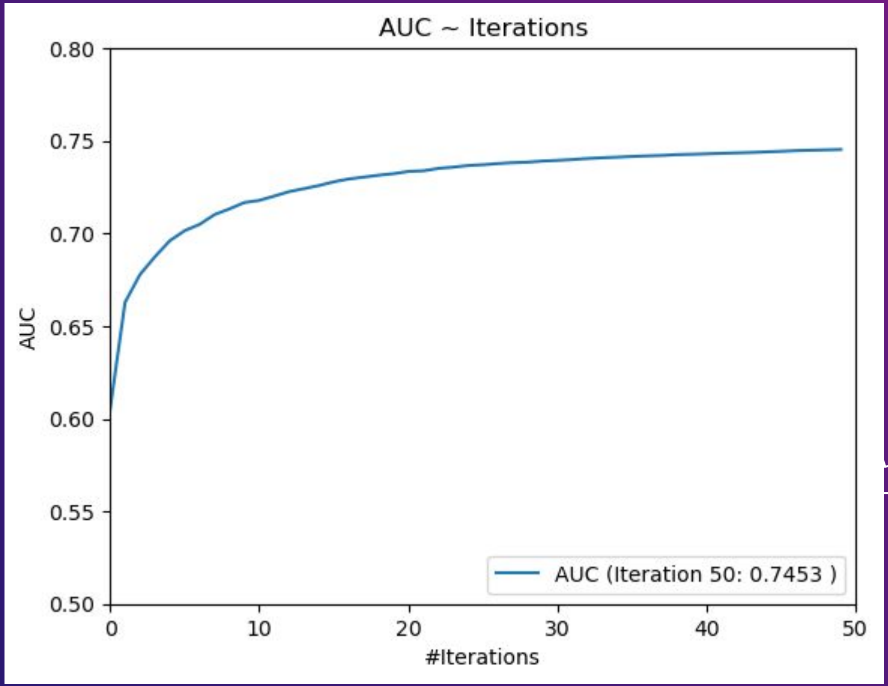
```
fpr, tpr, _ = roc_curve(trainY, trainY_pred)

plt.figure()
lw = 2
plt.plot(
    fpr,
    tpr,
    color="darkorange",
    lw=lw,
    label="ROC curve (area = %0.4f)" % roc_auc,
)
plt.plot([0, 1], [0, 1], color="navy", lw=lw, linestyle="--")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC of ada")
plt.legend(loc="lower right")
plt.show()
```

PLOTTING ROC CURVE

ROC CURVE


ROC of ada
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.7527)

CAN AUC BE IMPROVED WITH MORE ITERATIONS?



08 CONCLUSION

CONCLUSION

DATA SET

Home Credit Default Risk

TREES AND RANDOM FORESTS

Features and Applications

BOOSTING

Advantages over Trees and Random Forest

ADABOOST

Cons & Pros and Applications

THANKS