AdaBoost Classifier

Md. Golam Rabiul Alam

Boosting in ML

Boosting is an ensemble modeling technique which attempts to build a strong classifier from the number of weak classifiers.

The term 'Boosting' refers to a family of algorithms which converts weak learner to strong learners.

Types of Boosting Algorithms

- AdaBoost (**Ada**ptive **Boost**ing)
- Gradient Tree Boosting
- XGBoost

Boosting Example

How would you classify an email as SPAM or not?

1) Email has promotional image file, It’s a SPAM
2) Email has link(s), It’s a SPAM
3) Email body consist of sentence like “You won a prize money of $ ….”, It’s a SPAM
4) Email from our official domain “bracu.com” , Not a SPAM
5) Email from known source, Not a SPAM

Do you think these rules individually are strong enough to successfully classify an email?

To convert weak learner to strong learner, we can combine the prediction of each weak learner using methods like:

- Using average/ weighted average
- Considering prediction has higher vote

AdaBoost (Adaptive Boosting)

AdaBoost was the first really successful boosting algorithm developed for the purpose of binary classification.

B3

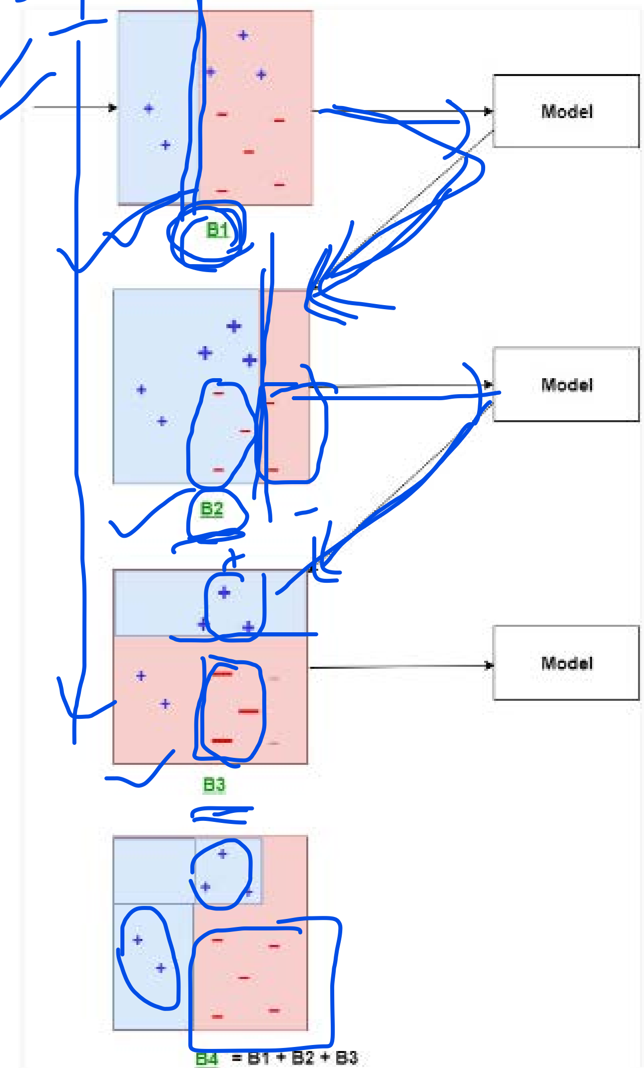
AdaBoost is short for Adaptive Boosting and is a very popular boosting technique which combines multiple “weak classifiers” into a single “strong classifier”.

B3

It was formulated by Yoav Freund and Robert Schapire.

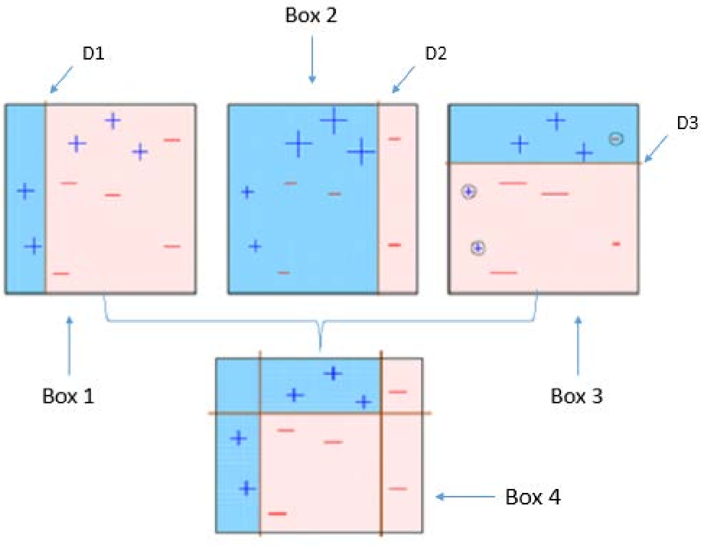
B1

B2



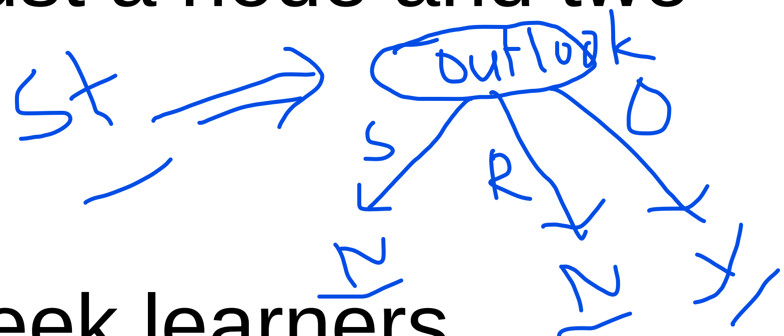
AdaBoosting Procedure

- *1. Initialize the dataset and assign equal weight (attention) to each of the data point.*
- *2. Provide this as input to the model and identify the wrongly classified data points.*
- *3. Increase the weight (attention) of the wrongly classified data points.*
- *4. if (got required results)*
  Goto step 5
  else
  Goto step 2
- *5. End*



AdaBoost with DT and RF

- In AdaBoost, we have to construct a forest of trees $\boxed{\text{(Stumps)}}$ with just a node and two leaves.



- Here, stumps are the week learners. Unlike RF, the weighted voting approach of stumps are used in bagging process. The order of stumps construction is also a matter in AdaBoost.

The AdaBoost Algorithm

(Freund and Schapire, 1996)

Given data: $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

1. Initialize weights $w_i = 1/N, i = 1, \ldots, N$
2. For $m = 1 : M$ 100
   a) Fit classifier $G_m(\mathbf{x}) \in \{-1, 1\}$ to data using weights w_i
   b) Compute

$$err_m = \frac{\sum_{i=1}^{N} w_i I\left(y_i \neq G_m(\mathbf{x}_i)\right)}{\sum_{i=1}^{N} w_i}$$

   c) Compute $\alpha_m = \log\left(\left(1 - err_m\right)/err_m\right)$
   d) Set $w_i \leftarrow w_i \exp\left[\alpha_m I\left(y_i \neq G_m(\mathbf{x}_i)\right)\right], \quad i = 1, \ldots, N$

AdaBoost Example

Outlook

Sunny

Overcast

Rain

Day	Outlook	Temperature	Humidity	Wind	Play Tennis	Sample Weight
Day1	Sunny	Hot	High	Weak	No	1/14
Day2	Sunny	Hot	High	Strong	No	1/14
Day3	Overcast	Hot	High	Weak	Yes	1/14
Day4	Rain	Mild	High	Weak	Yes	1/14
Day5	Rain	Cool	Normal	Weak	Yes	1/14
Day6	Rain	Cool	Normal	Strong	No	1/14
Day7	Overcast	Cool	Normal	Strong	Yes	1/14
Day8	Sunny	Mild	High	Weak	No	1/14
Day9	Sunny	Cool	Normal	Weak	Yes	1/14
Day10	Rain	Mild	Normal	Weak	Yes	1/14
Day11	Sunny	Mild	Normal	Strong	Yes	1/14
Day12	Overcast	Mild	High	Strong	Yes	1/14
Day13	Overcast	Hot	Normal	Weak	Yes	1/14
Day14	Rain	Mild	High	Strong	No	1/14

$N = 14$

x_N

y_n

$4 \times \frac{1}{14} =$

$Y = +1$

$N = -1$

AdaBoost

GINI Impurity:

The GINI Impurity of a node is the probability that a randomly chosen sample in a node would be incorrectly labeled if it was labeled by the distribution of samples in the node.

The GINI impurity can be computed by summing the probability $\boldsymbol{p_i}$ of an item with label ***i*** being chosen times the probability $\sum_{k \neq i} p_k = 1 - p_i$ of a mistake in categorizing that item.

$$\mathrm{I}_G(p) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = \sum_{i=1}^{J} p_i (1 - p_i) = \sum_{i=1}^{J} (p_i - {p_i}^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} {p_i}^2 = 1 - \sum_{i=1}^{J} {p_i}^2$$

It reaches its minimum (zero) when all cases in the node fall into a single target category.

AdaBoost

- [Outlook=Sunny(5): Yes-2 and No-3]
- [Outlook=Overcast(4): Yes-4 and No-0]
- [Outlook=Rain(5): Yes-3 and No-2]

GINI(Outlook=sunny) = 1 – (2/5)^2-(3/5)^2 = 1 – 0.16 – 0.36 = 0.48

GINI(Outlook= Overcast) = 1 - (4/4)^2-(0/4)^2 = 1 - 1 - 0 = 0

GINI(Outlook= Rain) = 1 - (3/5)^2-(2/5)^2 = 1 – 0.36 – 0.16 = 0.48

Now,

GINI impurity of parent node = weighted average of Gini impurities of leaf nodes

GINI(Outlook) = (5/14)*0.48 + (4/14)*0 + (5/14)*0.48 = 0.343

AdaBoost

- [Temperature=Hot(4): Yes-2 and No-2]
- [Temperature=Mild(6): Yes-4 and No-2]
- [Temperature=Cool(4): Yes-3 and No-1]

GINI(Temp=Hot) = 1 – (2/4)^2-(2/4)^2 = 0.5

GINI(Temp= Mild) = 1 - (4/6)^2-(2/6)^2 = 0.445

GINI(Temp= Cool) = 1 - (3/4)^2-(1/4)^2 = 0.375

Now,

GINI impurity of parent node = weighted average of Gini impurities of leaf nodes

GINI(Temperature) = (5/14)*0.5 + (4/14)*0.44 + (5/14)*0.375 = 0.441

AdaBoost

- [Humidity=High(7): Yes-3 and No-4]
- [Humidity=Normal(7): Yes-6 and No-1]

GINI(Hum=High) = 1 – (3/7)^2-(4/7)^2 = 0.49

GINI(Hum= Normal) = 1 - (6/7)^2-(1/7)^2 = 0.25

Now,

GINI impurity of parent node = weighted average of Gini impurities of leaf nodes

GINI(Humidity) = (7/14)*0.49 + (7/14)*0.25 = 0.367

AdaBoost

- [Wind=Strong(6): Yes-3 and No-3]
- [Wind=Weak(8): Yes-6 and No-2]

GINI(Wind=Strong) = 1 – (3/6)^2-(3/6)^2 = 0.5

GINI(Wind= Weak) = 1 - (6/8)^2-(2/8)^2 = 0.375

Now,

GINI impurity of parent node = weighted average of Gini impurities of leaf nodes

GINI(Wind) = (6/14)*0.5 + (8/14)*0.375 = 0.429

GINI comparison

0.459 –

GINI(Outlook) = (5/14)*0.48 + (4/14)*0 + (5/14)*0.48 = 0.343

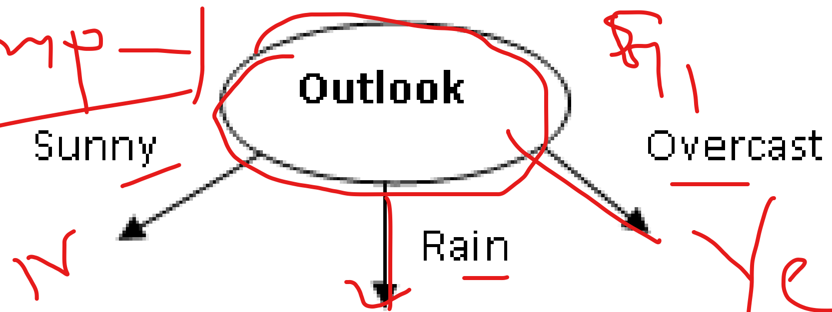
GINI(Temperature) = (5/14)*0.5 + (4/14)*0.44 + (5/14)*0.375 = 0.441

GINI(Humidity) = (7/14)*0.49 + (7/14)*0.25 = 0.367

GINI(Wind) = (6/14)*0.5 + (8/14)*0.375 = 0.429

GINI(Outlook) = (5/14)*0.48 + (4/14)*0 + (5/14)*0.48 = 0.34 is the lowest. So, outlook is the first stump.

stump



[Outlook=Sunny(5): Yes-2 and No-3]→
[Outlook=Overcast(4): Yes-4 and No-0]
[Outlook=Rain(5): Yes-3 and No-2]

GINI comparison for stump selection

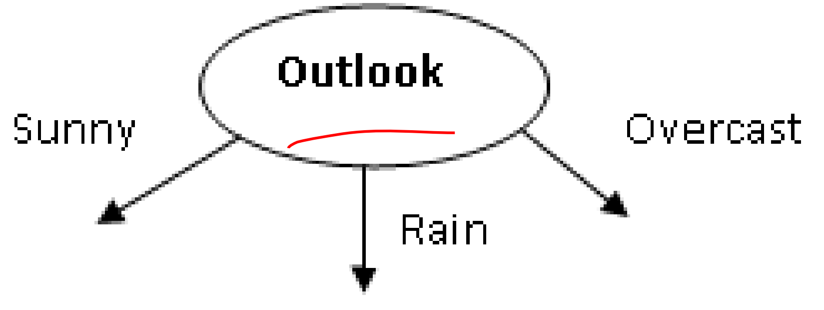
GINI(Outlook) = (5/14)*0.48 + (4/14)*0 + (5/14)*0.48 = 0.343

GINI(Temperature) = (5/14)*0.5 + (4/14)*0.44 + (5/14)*0.375 = 0.441

GINI(Humidity) = (7/14)*0.49 + (7/14)*0.25 = 0.367

GINI(Wind) = (6/14)*0.5 + (8/14)*0.375 = 0.429

GINI(Outlook) = (5/14)*0.48 + (4/14)*0 + (5/14)*0.48 = 0.34 is the lowest. So, outlook is the first stump.



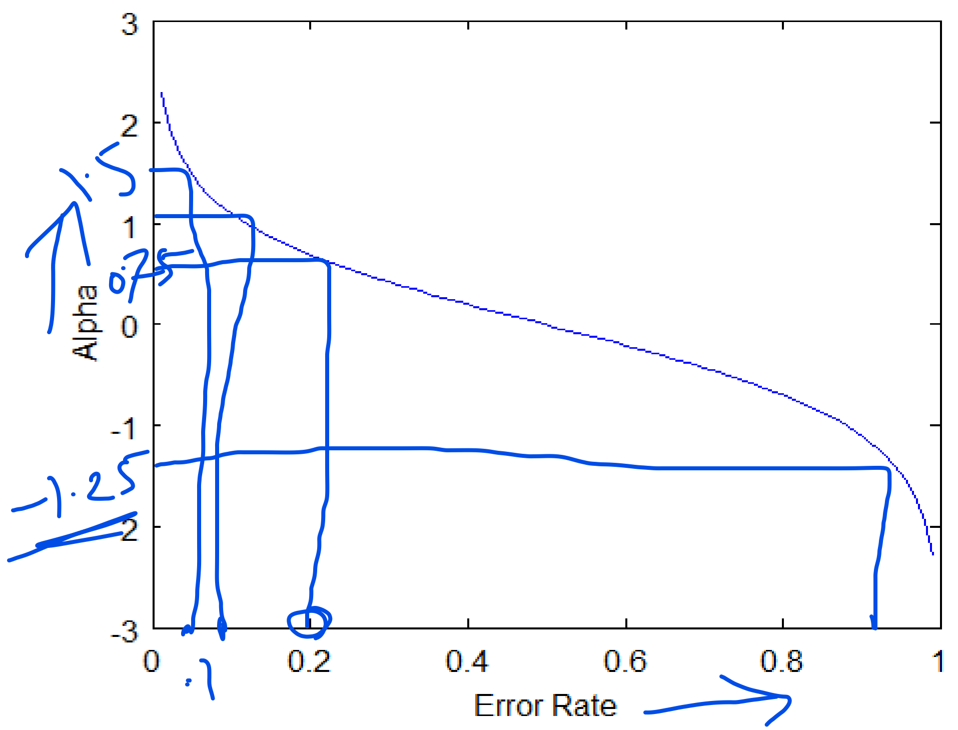
[Outlook=Sunny(5): Yes-2 and No-3]
[Outlook=Overcast(4): Yes-4 and No-0]
[Outlook=Rain(5): Yes-3 and No-2]

Amount of say determination

Amount of say or α = ½ ln ((1-Total error)/Total error)

Here, amount of say is the measurement of how well it classified the samples in final classification.

Total error is the sum of the weights associated with the incorrectly classified samples.



Total Error

[Outlook=Sunny(5): Yes-2 and No-3]
[Outlook=Overcast(4): Yes-4 and No-0]
[Outlook=Rain(5): Yes-3 and No-2]

Day	Outlook	Temperature	Humidity	Wind	Play Tennis	Sample Weight
Day1	Sunny	Hot	High	Weak	*No*	1/14
Day2	Sunny	Hot	High	Strong	*No*	1/14
Day3	Overcast	Hot	High	Weak	Yes	1/14
Day4	Rain	Mild	High	Weak	Yes	1/14
Day5	Rain	Cool	Normal	Weak	Yes	1/14
Day6	Rain	Cool	Normal	Strong	*No*	1/14
Day7	Overcast	Cool	Normal	Strong	Yes	1/14
Day8	Sunny	Mild	High	Weak	*No*	1/14
Day9	Sunny	Cool	Normal	Weak	Yes	1/14
Day10	Rain	Mild	Normal	Weak	Yes	1/14
Day11	Sunny	Mild	Normal	Strong	Yes	1/14
Day12	Overcast	Mild	High	Strong	Yes	1/14
Day13	Overcast	Hot	Normal	Weak	Yes	1/14
Day14	Rain	Mild	High	Strong	*No*	1/14

Total Error= 2* (1/14) + 0+ 2* (1/14)=2/7=$\boxed{0.29}$

Amount of say or α = ½ ln ((1-Total error)/Total error)= $\boxed{0.45}$

Total Error

[Outlook=Sunny(5): Yes-2 and No-3]
[Outlook=Overcast(4): Yes-4 and No-0]
[Outlook=Rain(5): Yes-3 and No-2]

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Sample Weight
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14

Total Error= 2* (1/14) + 0+ 2* (1/14)=2/7=0.29

Amount of say or α = ½ ln ((1-Total error)/Total error)= 0.45

New sample Weight

New Sample Weight for incorrectly classified sample
= sample weight * e ^{alpha}
= 1/14 * e $^{0.45}$
= 0.11

New Sample Weight for correctly classified sample
= sample weight * e $^{–alpha}$
= 1/14* e $^{–alpha}$
= 0.046

Updated sample weight

Day	Outlook	Temperature	Humidity	Wind	Play Tennis	Sample Weight	Normalized Sample Weight	Cumulative Normalized Sample Weight
Day1	Sunny	Hot	High	Weak	No	0.046	0.051	0.051
Day2	Sunny	Hot	High	Strong	No	0.046	0.051	0.102
Day3	Overcast	Hot	High	Weak	Yes	0.046	0.051	0.153
Day4	Rain	Mild	High	Weak	Yes	0.046	0.051	0.204
Day5	Rain	Cool	Normal	Weak	Yes	0.046	0.051	0.255
Day6	Rain	Cool	Normal	Strong	No	0.11	0.122	0.377
Day7	Overcast	Cool	Normal	Strong	Yes	0.046	0.051	0.428
Day8	Sunny	Mild	High	Weak	No	0.046	0.051	0.479
Day9	Sunny	Cool	Normal	Weak	Yes	0.11	0.122	0.601
Day10	Rain	Mild	Normal	Weak	Yes	0.046	0.051	0.652
Day11	Sunny	Mild	Normal	Strong	Yes	0.11	0.122	0.774
Day12	Overcast	Mild	High	Strong	Yes	0.046	0.051	0.825
Day13	Overcast	Hot	Normal	Weak	Yes	0.046	0.051	0.876
Day14	Rain	Mild	High	Strong	No	0.11	0.122	0.998

“Normalized sample weight” = “Sample Weight” / “Summation of all of the sample weights”

$NSW_i = SW_i / \Sigma SW_i$

100

0 ~ .1 = 10

.1 ~ .2 = 10

.9 ~ 1 = 10

0 ~ .15 = 15

Updated sample weight

Day	Outlook	Temperature	Humidity	Wind	Play Tennis	Cumulative Normalized Sample Weight	Generated Random Number
Day1	Sunny	Hot	High	Weak	No	0.051	0.040
Day2	Sunny	Hot	High	Strong	No	0.102	0.100
Day3	Overcast	Hot	High	Weak	Yes	0.153	0.151
Day4	Rain	Mild	High	Weak	Yes	0.204	0.200
Day5	Rain	Cool	Normal	Weak	Yes	0.255	0.250
Day6	Rain	Cool	Normal	Strong	No	0.377	0.267
Day7	Overcast	Cool	Normal	Strong	Yes	0.428	0.500
Day8	Sunny	Mild	High	Weak	No	0.479	0.700
Day9	Sunny	Cool	Normal	Weak	Yes	0.601	0.990
Day10	Rain	Mild	Normal	Weak	Yes	0.652	0.370
Day11	Sunny	Mild	Normal	Strong	Yes	0.774	0.600
Day12	Overcast	Mild	High	Strong	Yes	0.825	0.682
Day13	Overcast	Hot	Normal	Weak	Yes	0.876	0.886
Day14	Rain	Mild	High	Strong	No	0.998	0.980

0.05

0.122

0.122

0.051

New Dataset Creation with Random Sampling

Day	Outlook	Temperature	Humidity	Wind	Play Tennis	Sample Weight	Normalized Sample Weight	Cumulative Normalized Sample Weight	Generated Random Number
Day1	Sunny	Hot	High	Weak	No	0.046	0.051	0.051	0.040
Day2	Sunny	Hot	High	Strong	No	0.046	0.051	0.102	0.100
Day3	Overcast	Hot	High	Weak	Yes	0.046	0.051	0.153	0.151
Day4	Rain	Mild	High	Weak	Yes	0.046	0.051	0.204	0.200
Day5	Rain	Cool	Normal	Weak	Yes	0.046	0.051	0.255	0.250
Day6	Rain	Cool	Normal	Strong	No	0.11	0.122	0.377	0.267
Day9	Sunny	Cool	Normal	Weak	Yes	0.11	0.122	0.428	0.500
Day11	Sunny	Mild	Normal	Strong	Yes	0.11	0.122	0.479	0.700
Day14	Rain	Mild	High	Strong	No	0.11	0.122	0.601	0.990
Day6	Rain	Cool	Normal	Strong	No	0.11	0.122	0.652	0.370
Day9	Sunny	Cool	Normal	Weak	Yes	0.11	0.122	0.774	0.600
Day11	Sunny	Mild	Normal	Strong	Yes	0.11	0.122	0.825	0.682
Day14	Rain	Mild	High	Strong	No	0.11	0.122	0.876	0.886
Day14	Rain	Mild	High	Strong	No	0.11	0.122	0.998	0.980

G_2

New Dataset Creation with Sample weight

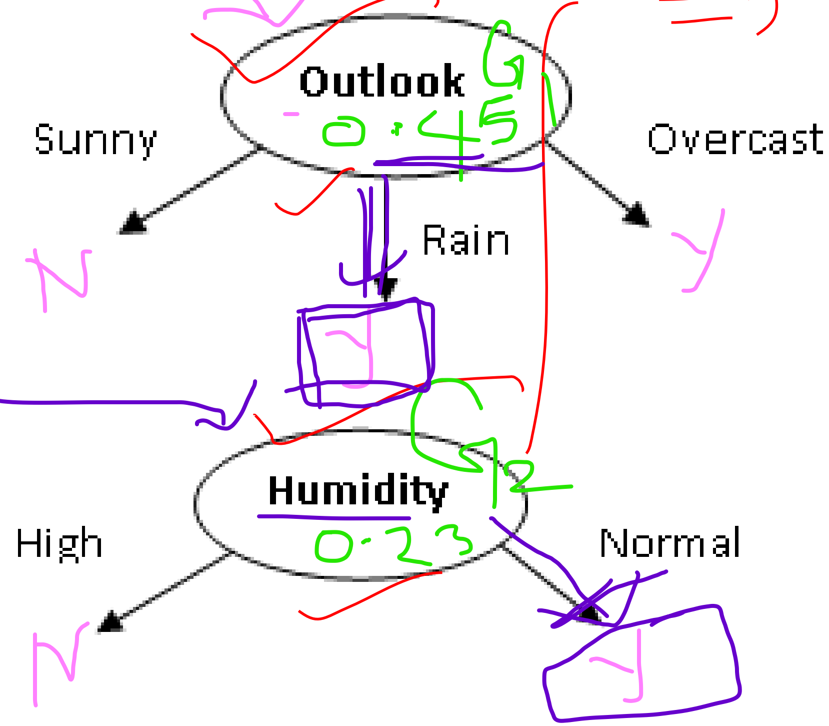
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day14	Rain	Mild	High	Strong	No
Day6	Rain	Cool	Normal	Strong	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day14	Rain	Mild	High	Strong	No
Day14	Rain	Mild	High	Strong	No

Sample Weight
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14
1/14

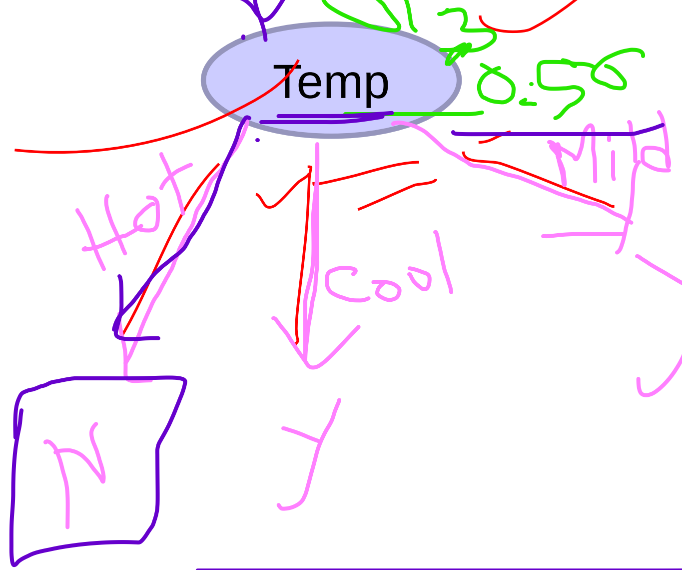
Day 30 | Outlook: Rain | Temp: Hot | Hum: Nor | Wind: St | PL: Y

Bagging

Total Amount of say for YES stumps= 0.45+0.23=0.68



Total Amount of say for NO stumps= 0.56



YES will be the classification

Y : 0.45 + 0.23 = 0.68

N : 0.56

Conclusion

- We can use AdaBoost algorithms for both classification and regression problem.

References

- 1. Boosting : foundations and algorithms by Robert E. Schapire and Yoav Freund.
- **MIT press**
- https://mitpress.mit.edu/sites/default/files/titles/content/boosting_foundations_algorithms/toc.html#indx-1

- **2. Boosting and AdaBoost for Machine Learning**
- https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/

- **3. Quick Introduction to Boosting Algorithms in Machine Learning**
- https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/

- **4. An Introduction to Statistical Learning: with Applications in R by Gareth James et al.**
- **http://www.amazon.com/dp/1461471370?tag=inspiredalgor-20**

- **5. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, by Trevor Hastie et al.**
- **http://www.amazon.com/dp/0387848576?tag=inspiredalgor-20**
- **6. Applied Predictive Modeling by Max Kuhn**
- https://www.amazon.com/dp/1461468485?tag=inspiredalgor-20

- **7.** AdaBoost.SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function)
- https://web.stanford.edu/~hastie/Papers/SII-2-3-A8-Zhu.pdf