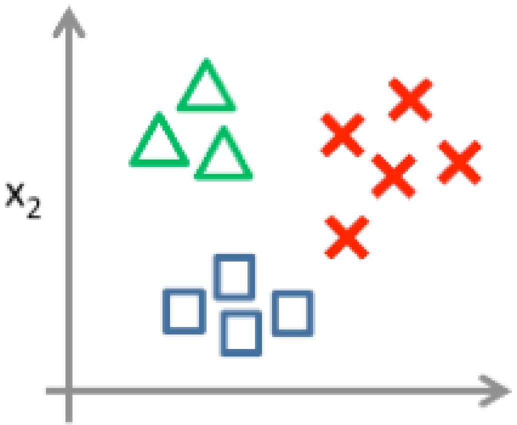
ML-101:
Classification, Random Forest Algorithm

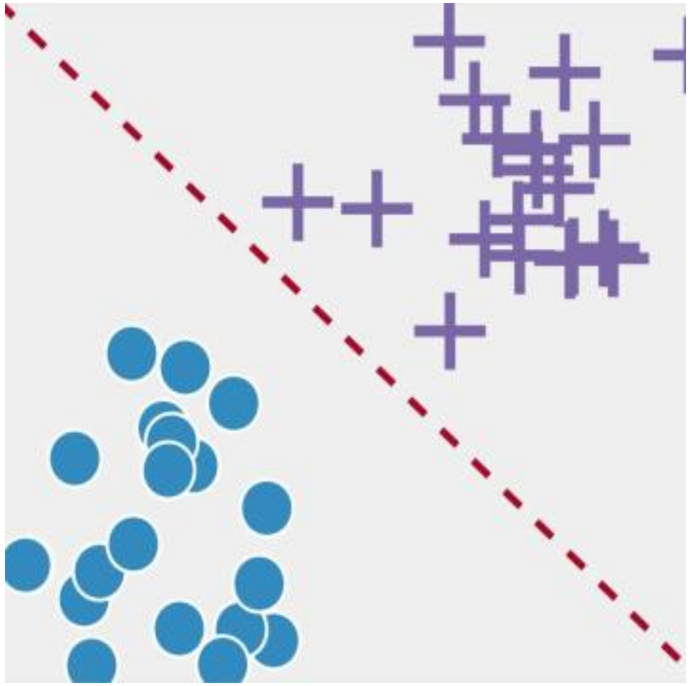
BY SARTHAK CONSUL

Classification

- Y={0,1} – Binary
- Y={0,1,…k} – Multiclass

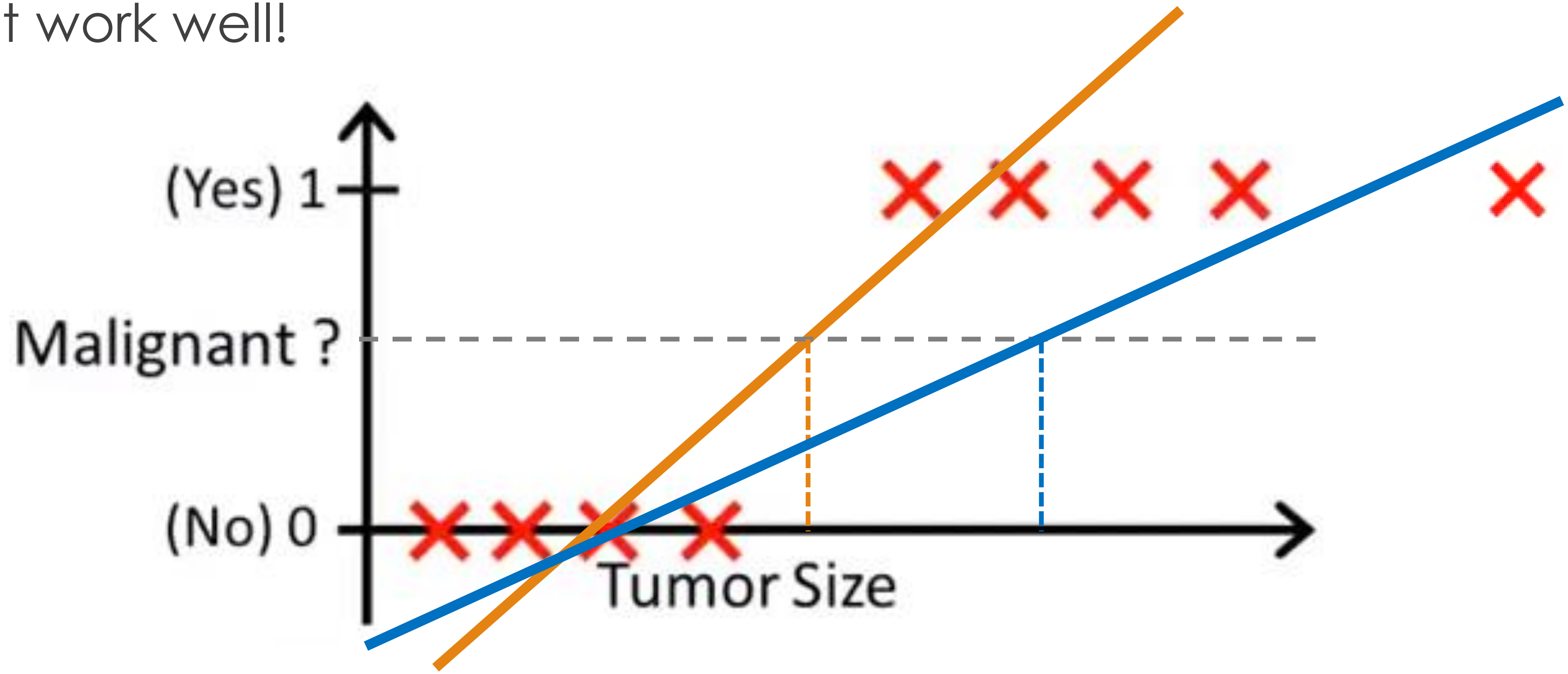
Eg. Spam filters, Image Recognition, Tumour type





Attempt 1:
Linear Regression

- Won’t work well!

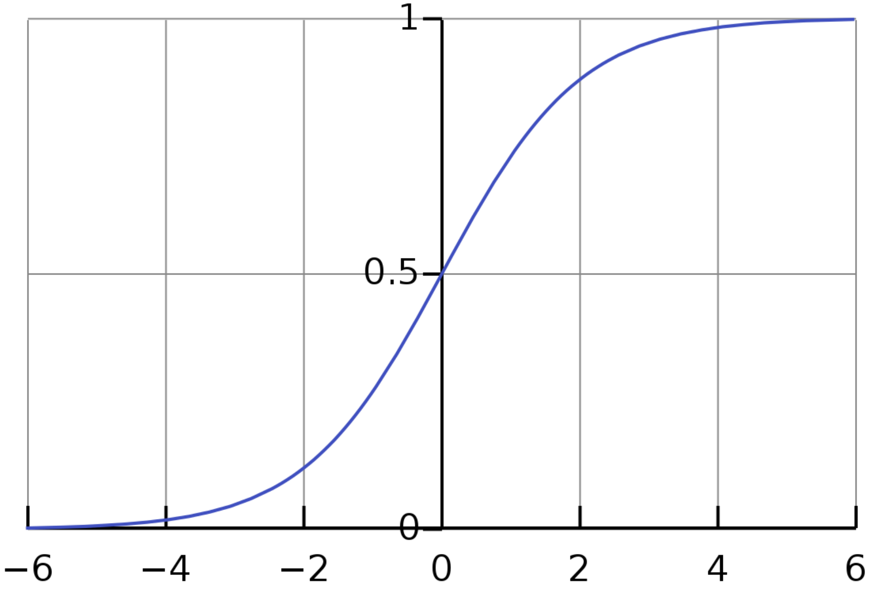


Source: Coursera - Machine Learning by Andrew Ng

A Better Way:

Logistic Regression

- The sigmoid Function: $$g(z) = \frac{1}{1 + e^{-z}}$$



$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Classification on the basis of $h_\theta(x)$ compared to 0.5
- Multiclass extension: One-vs-All

Cost Function for Logistic Regression

- Loss should be of the form that penalizes wring sign of prediction and actual value i.e.

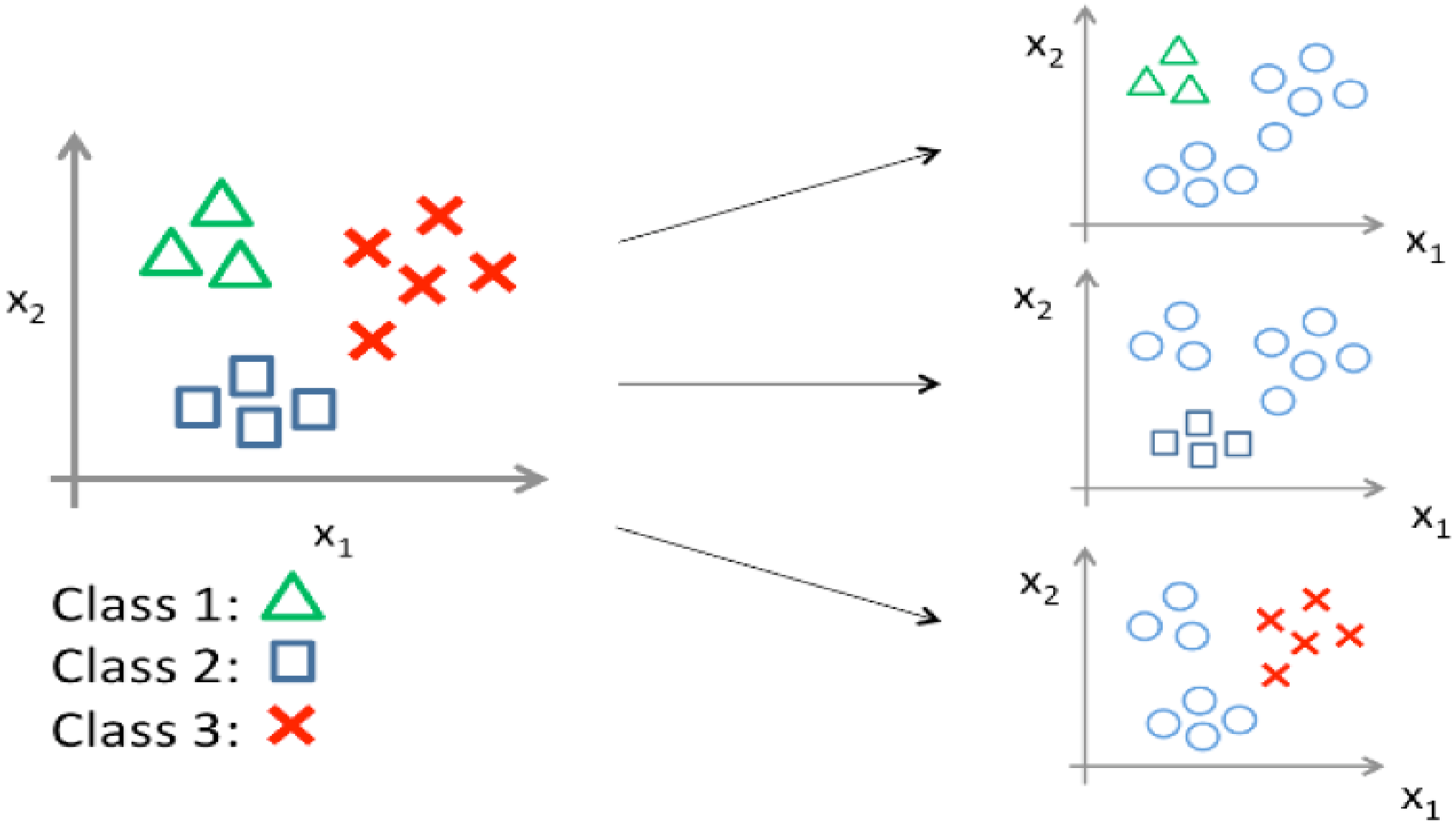
$$L(y\hat{y}) = (1 - sign(y\hat{y}))/2$$

- MSE wont work $\rightarrow$ non convex
- Instead objective function (to minimize) is:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)} \log h_\theta(x^{(i)}) - (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)$$

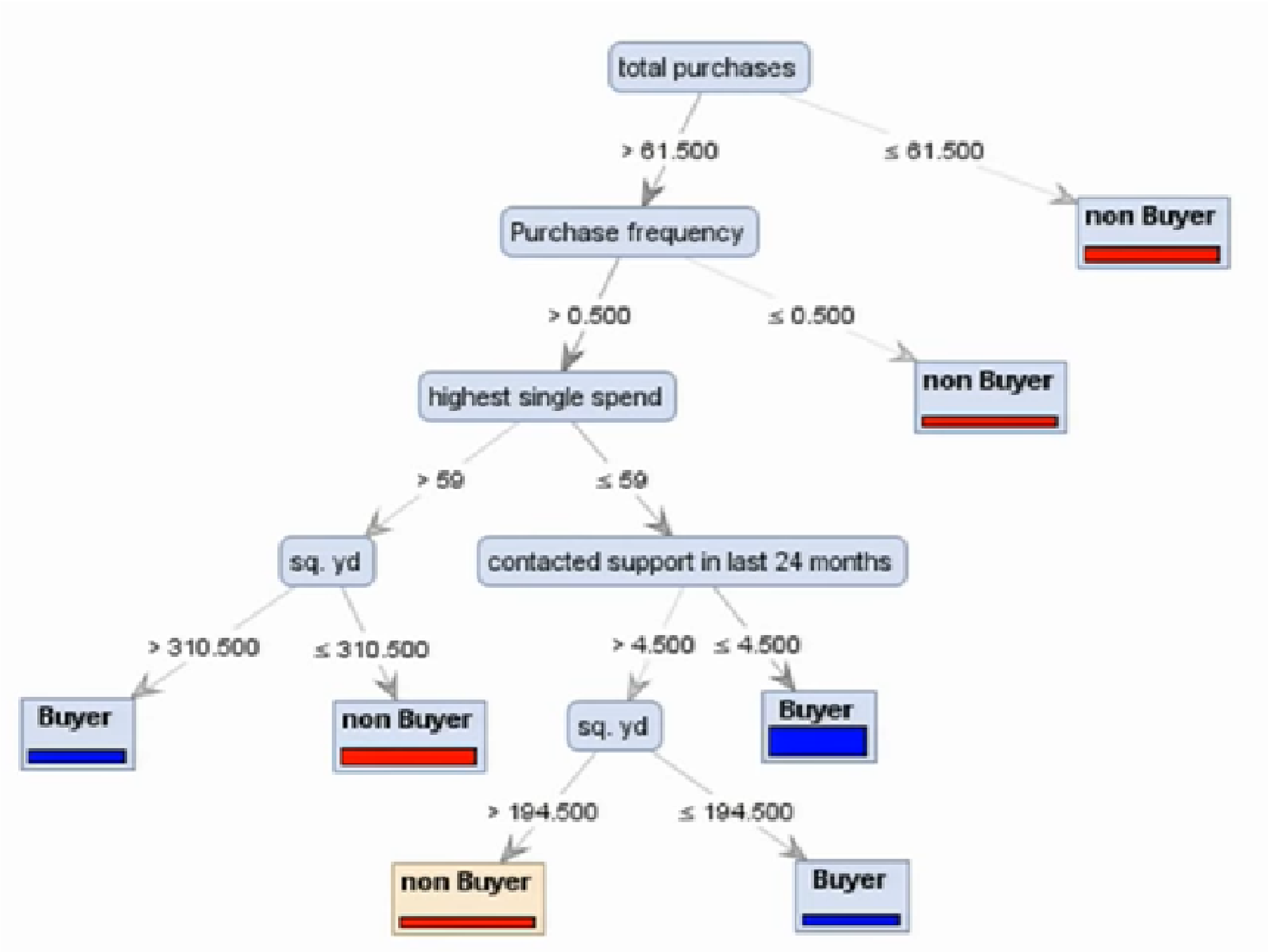
- Gradient Descent $\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$

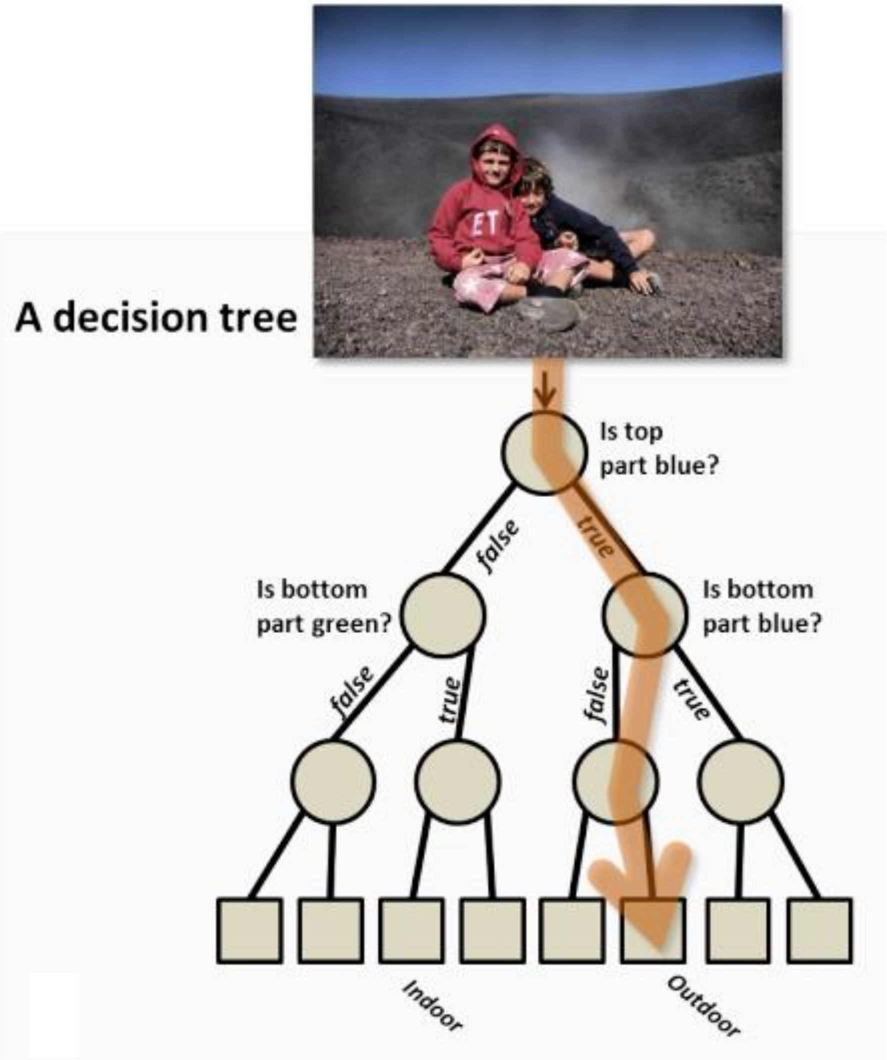
Multiclass Classification: One-vs-All



Decision Tree

- Used a lot 2-3 decades ago
- Fell out of fashion as they tend to not generalize well
- Trees have large variance, averaging out many trees reduces the varies
- Modified to make very powerful algorithms (eg. Random Forest/ Decision Forest)





Source: MSR Tutorial on decision forests by Criminisi et al, 2011

Constructing a Decision Tree

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Source: Artificial Intelligence: A Modern Approach by Stuart Russel and Peter Norvig

Entropy and Information Gain

- For a training set containing p positive and n negative examples,

$$H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

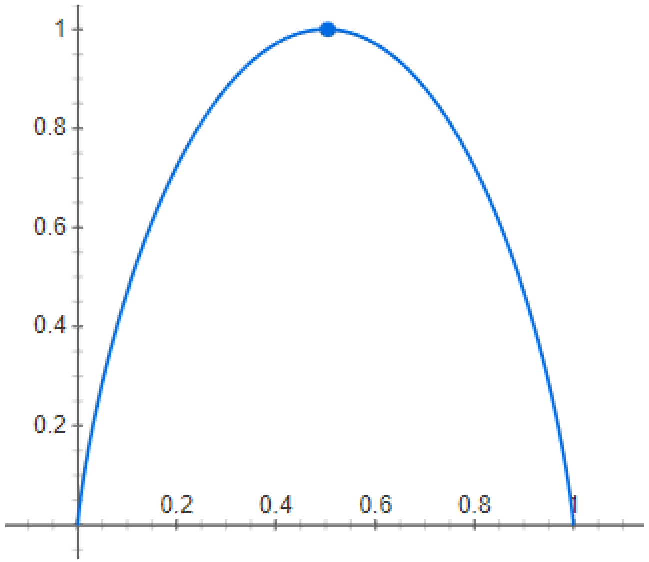
- Attribute A divides the training set into K sunsets
  - Expected Entropy remaining after A, EH

$$EH(A) = \sum_{i=1}^{K} \frac{p_i + n_i}{p+n} H\left(\frac{p_i}{p_i + n_i}\right)$$

  - Information Gain (reduction in entropy), I

$$I(A) = H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - EH(A)$$

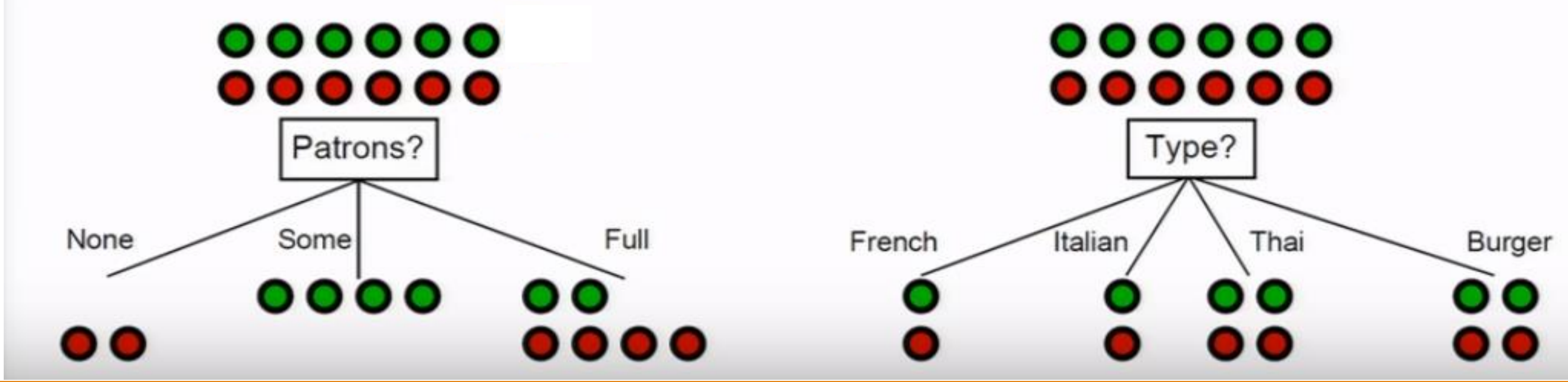
  - Advanced Idea: Gaussians to decide



Choosing the attributes

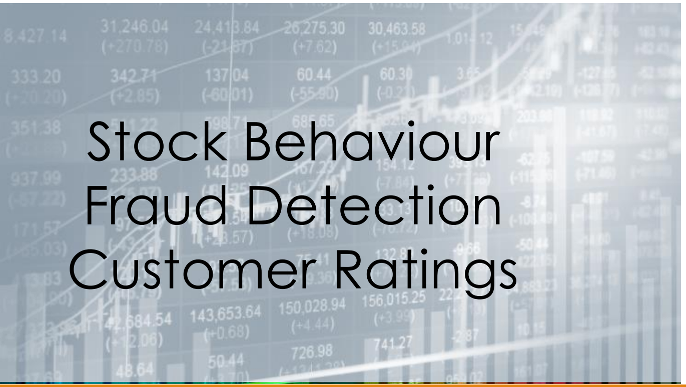
$$I(Patrons) = 1 - [\frac{2}{12}H(0,1) + \frac{4}{12}H(1,0) + \frac{6}{12}H(\frac{2}{6},\frac{4}{6})] = .0541 \text{ bits}$$

$$I(Type) = 1 - [\frac{2}{12}H(\frac{1}{2},\frac{1}{2}) + \frac{2}{12}H(\frac{1}{2},\frac{1}{2}) + \frac{4}{12}H(\frac{2}{4},\frac{2}{4}) + \frac{4}{12}H(\frac{2}{4},\frac{2}{4})] = 0 \text{ bits}$$



Random Forest Algorithm

- One of the most popular powerful supervised learning algorithm
- Can perform both regression and classification tasks
- It builds multiple decision trees and merges them together to get a more accurate and stable prediction
- Uses:



Growing a Forest from Trees

- d=#of Features, n=#of data points
- Pick any 2 features at **RANDOM,** split according to it-> RANDOM TREE
  - Reduces computation time
  - Randomness in introduced
- Take trees and average them to make a FOREST

Growing a Forest from Trees: the Algo

(Bagging) **B**ootstrap **Agg**regat**ing**

For b=1 to T:

a. Draw a *bootstrap sample, Z** of size N from training data
b. Grow a random-forest tree, T_b from Z*

   (via recursion – till min node size is reached)

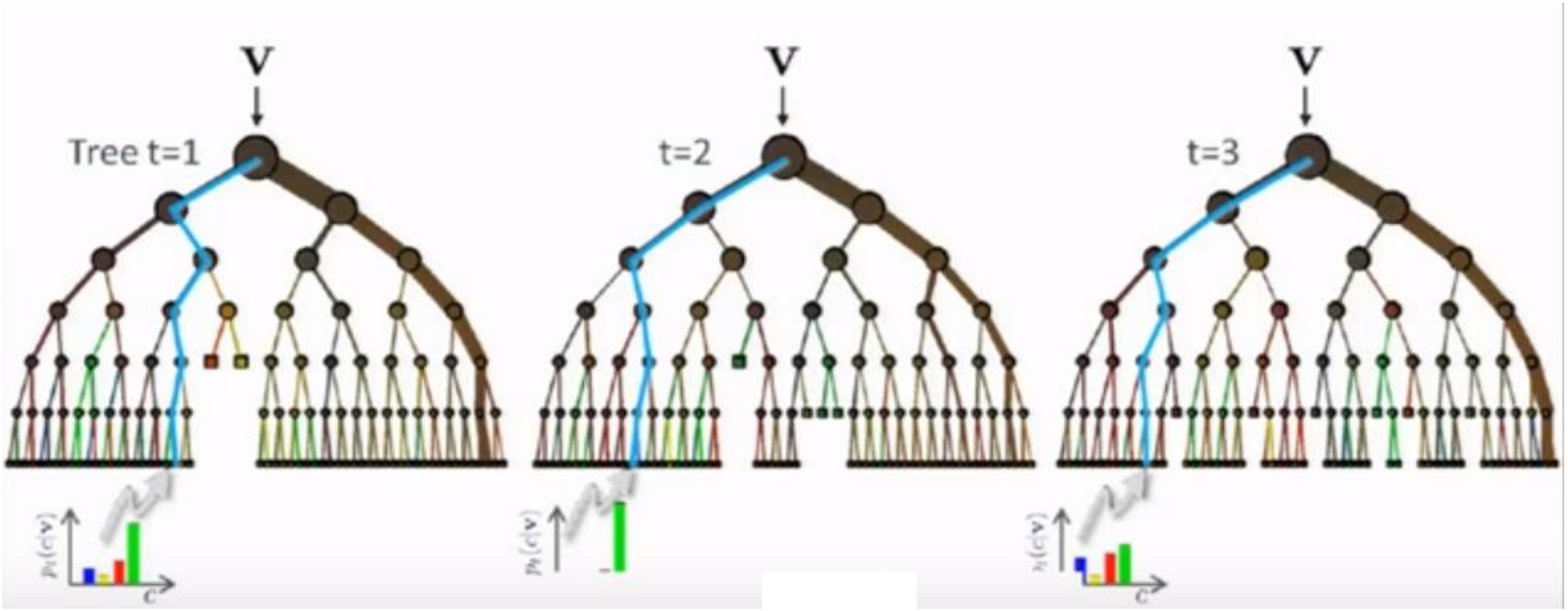
Ensemble of trees $\{T_b\}$

Word of the day: *Bootstrap Sample*
Sampling uniformly at random, with replacement

During testing,

i. each point is passed through all trees
ii. Average out the result (AM or GM), Max votes, etc.

Source: The Elements of Statistical Learning by Friedman, Tibshirani, and Hastie



$$p(c|\boldsymbol{v}) = \frac{1}{T}\sum_{t=1}^{T} p_t(c|v)$$

Source: Criminisi et al, 2011