

Machine Learning For Natural Language Processing

Abdelhak Mahmoudi
abdelhak.mahmoudi@um5.ac.ma

2020

Content

1. Introduction
2. Machine Learning
 1. Supervised Learning, 2. Unsupervised Learning
3. Natural Language Processing
 1. Regular Expressions, 2. Tokenization, 3. Character Encoding, 4. Part-of-Speech Tagging, 5. Chunking, 6. Stemming and Lemmatization, 7. Parsing, 8. Named Entity Recognition, 9. Topic Segmentation
4. Introduction to Deep Learning for NLP
 1. Sequence models, 2. Embeddings, 3. BERT models

Supervised Learning

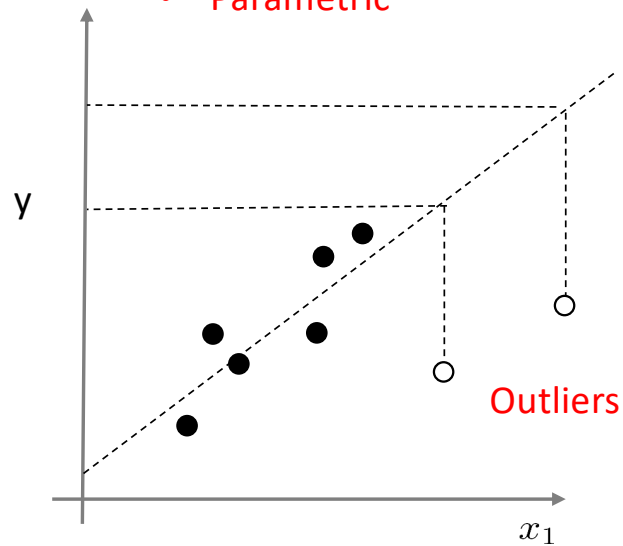
- Linear Regression
- Logistic Regression
- Support Vector Machines
- **Trees (Decision and Regression)**
- Random Forests
- Boosting
- Artificial Neural Networks

Classification and Regression Trees (CART)

Regression

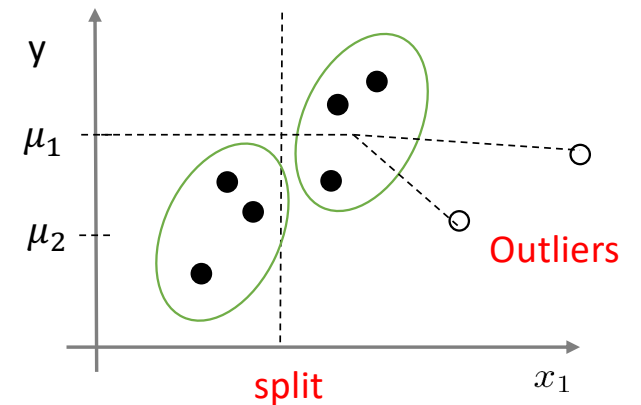
Linear regression

- Linear models
- Parametric



Regression trees

- Non Linear model
- Non-Parametric

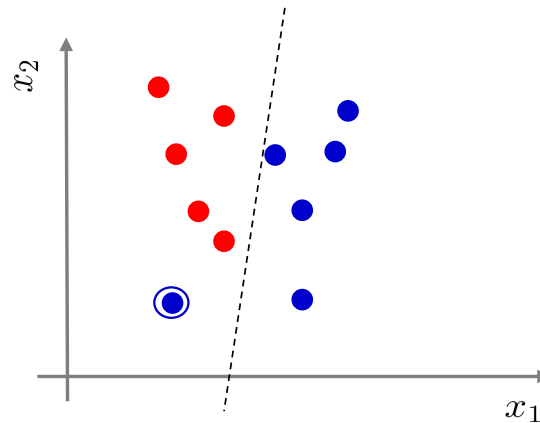


Classification and Regression Trees (CART)

Classification

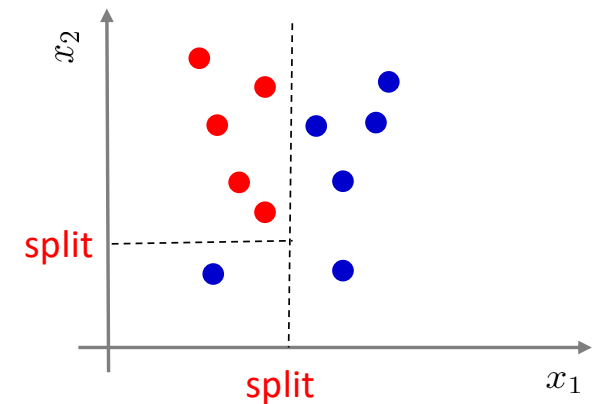
Logistic regression

- Linear models
- Parametric



Classification trees

- Non Linear model
- Non-Parametric



Classification Trees (aka Decision Trees)

Example of Restaurant Data

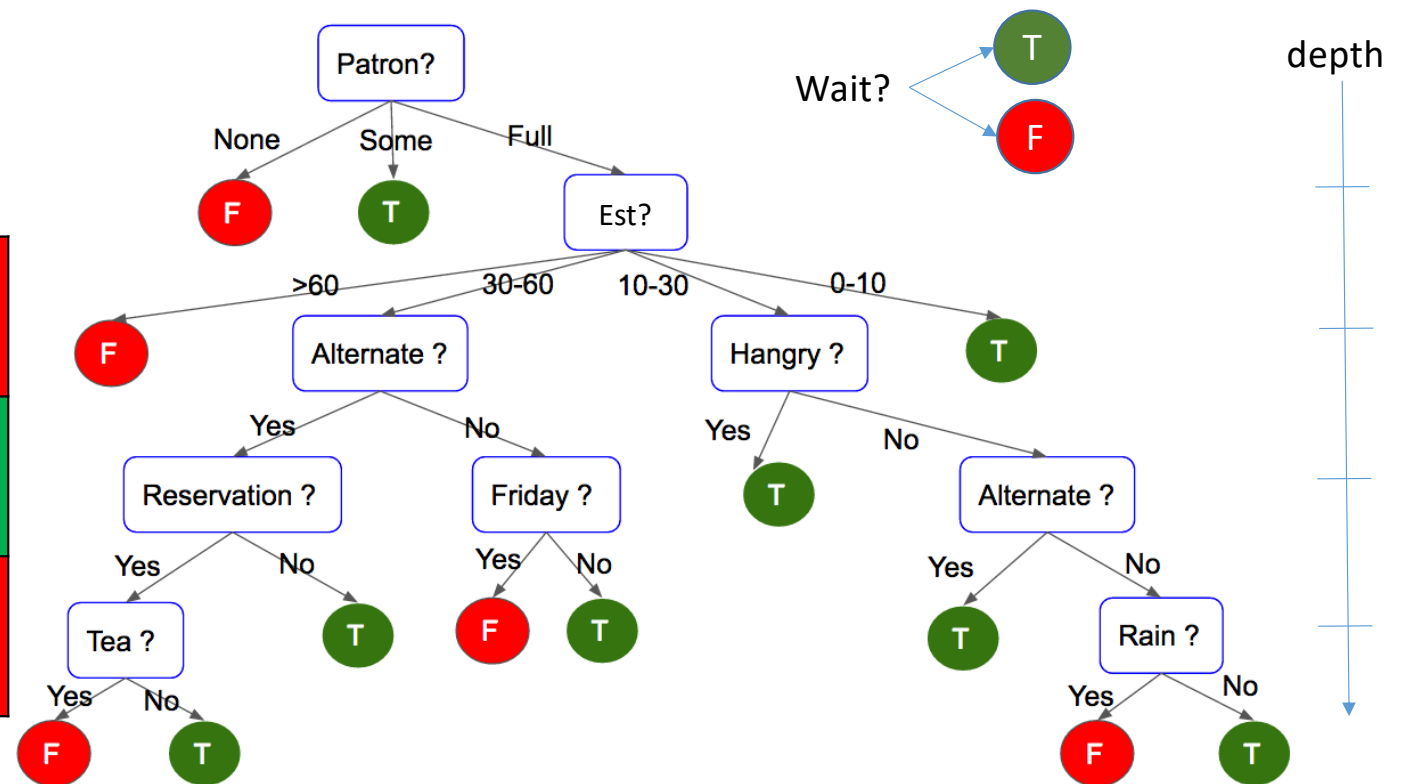
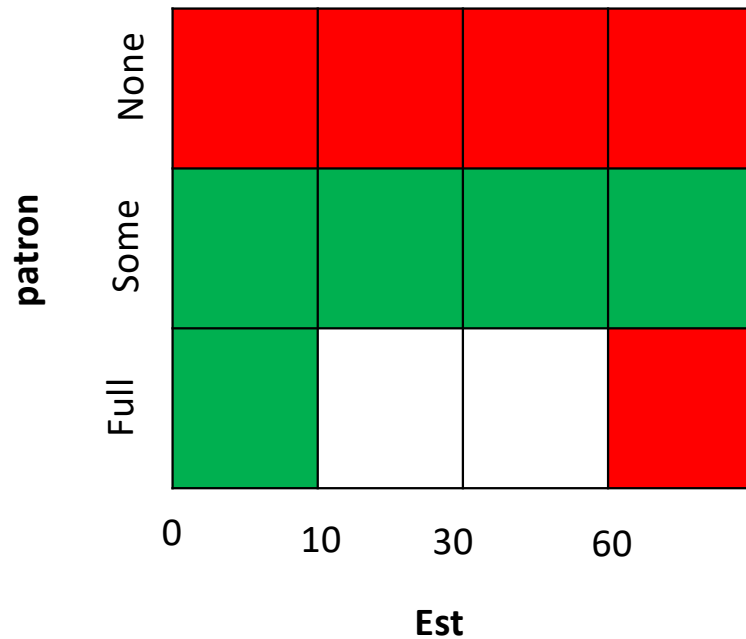
X	F										Y
Client	Alt	Tea	Fri	Hun	Patron	Price	Rain	Res	Type	Est	Wait
1	T	F	F	T	Some	\$\$\$	F	T	Moroccan	0-10	T
2	T	F	F	T	Full	\$	F	F	Chinese	30-60	F
3	F	T	F	F	Some	\$	F	F	Burger	0-10	T
4	T	F	T	T	Full	\$	F	F	Chinese	10-30	T
5	T	F	T	F	Full	\$\$\$	F	T	Moroccan	>60	F
6	F	T	F	T	Some	\$\$	T	T	Italian	0-10	T
7	F	T	F	F	None	\$	T	F	Burger	0-10	F
8	F	F	F	T	Some	\$\$	T	T	Chinese	0-10	T
9	F	T	T	F	Full	\$	T	F	Burger	>60	F
10	T	T	T	T	Full	\$\$\$	F	T	Italian	10-30	F
11	F	F	F	F	None	\$	F	F	Chinese	0-10	F
12	T	T	T	T	Full	\$	F	F	Burger	30-60	T

- Alt: is there any other alternative?
- Fri: is it Friday?
- Hun: is the client hungry?
- Patron: how many people are in the restaurant?
- Res: Restaurant
- Est: wait estimate

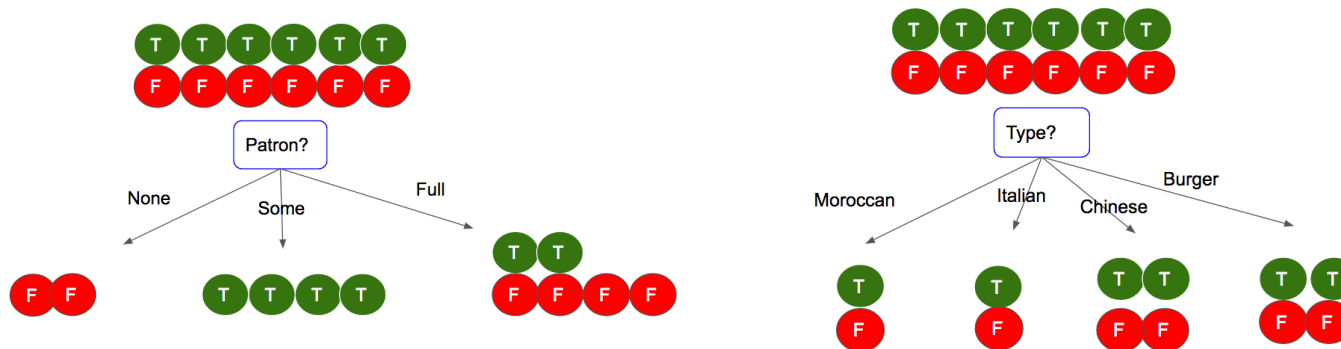
Most of the features are
Discrete (=categorical, =qualitative)

Classification Trees (aka Decision Trees)

Pick one feature at a time,
and make a binary decision



DT: Which feature to split with first?



Split with the feature F that **maximizes** the Information Gain

More informative
Less impurity

Less informative
More impurity

$$I(F) = H(S) - EH(F)$$

Information
Gain

Parent
entropy

Expected
entropy

S: subset = {p positives and n negatives}

F: Feature

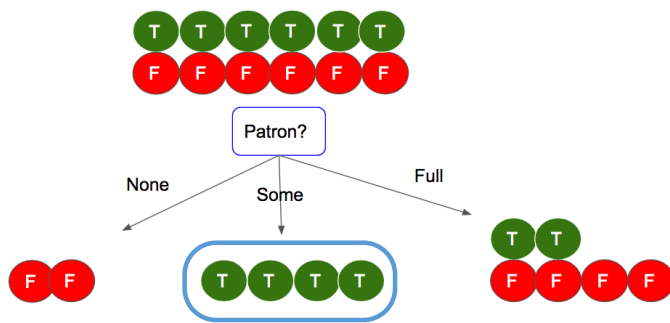
Entropy

$$H(S) = - \sum_c p_c \log_2(p_c)$$

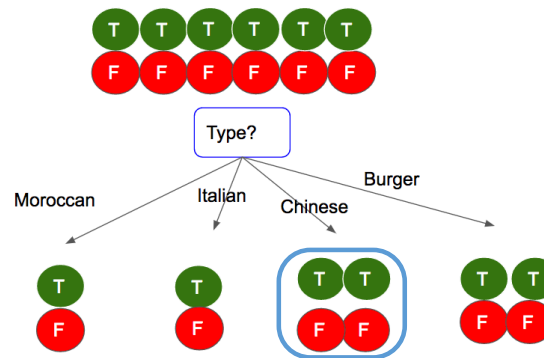
Python: `np.log2()`

p_c : probability of examples in class c
 S : subset of data examples

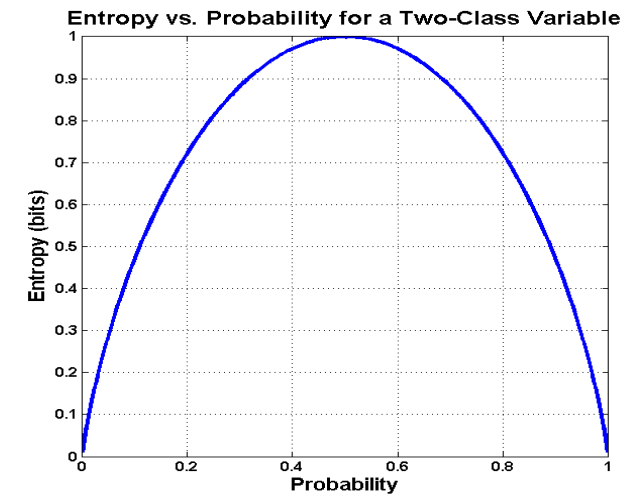
Interpretation: Measure of the **impurity** in a **subset** of examples



All examples in the **same** class
Entropy = 0



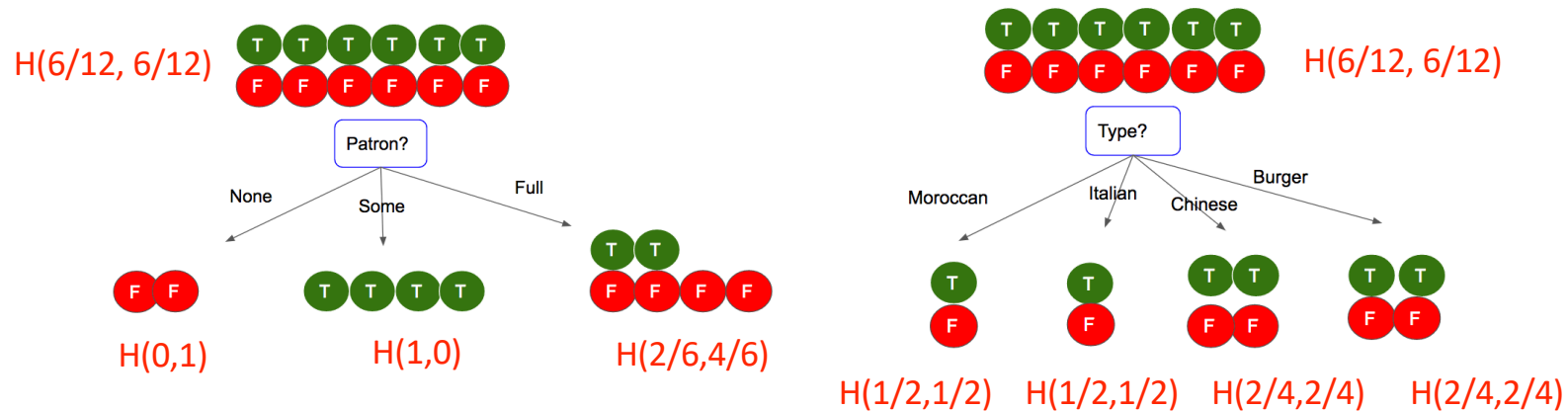
All examples **evenly split** between classes
Entropy = 1



Entropy (binary classification)

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

p: positive
n: negative

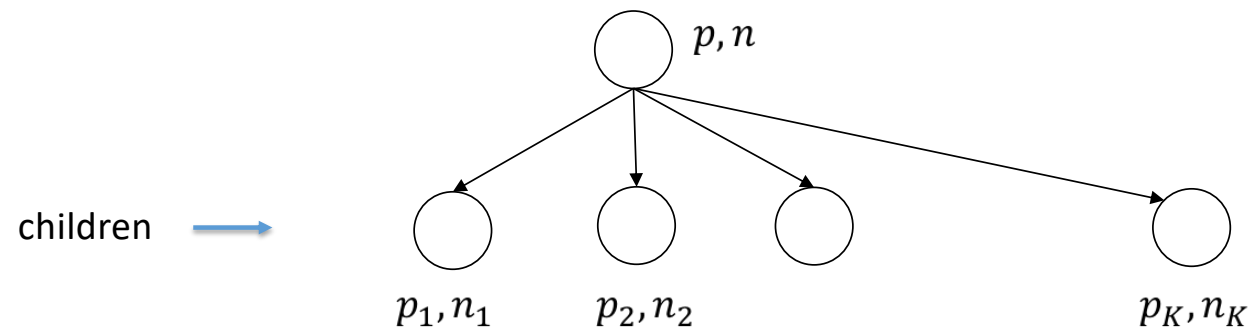


Expected Entropy

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

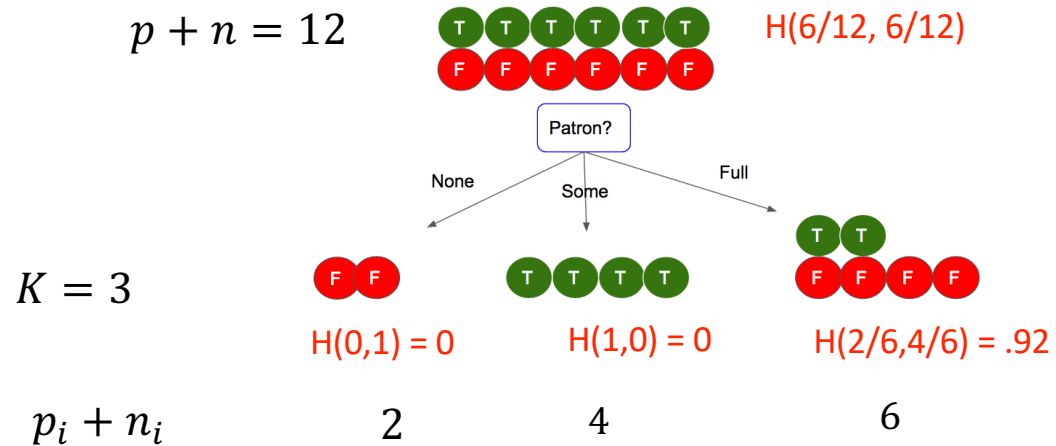
K = number of splits (regions) with Feature **F**
= number of **children nodes**

Expectation Entropy = weighted average of **children** entropy



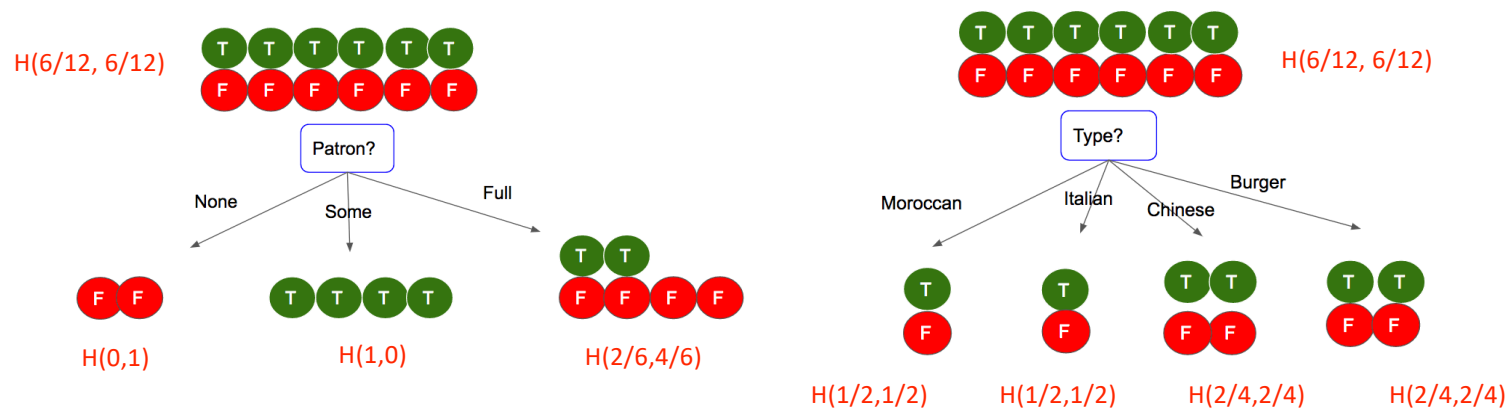
Expected Entropy (Example)

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



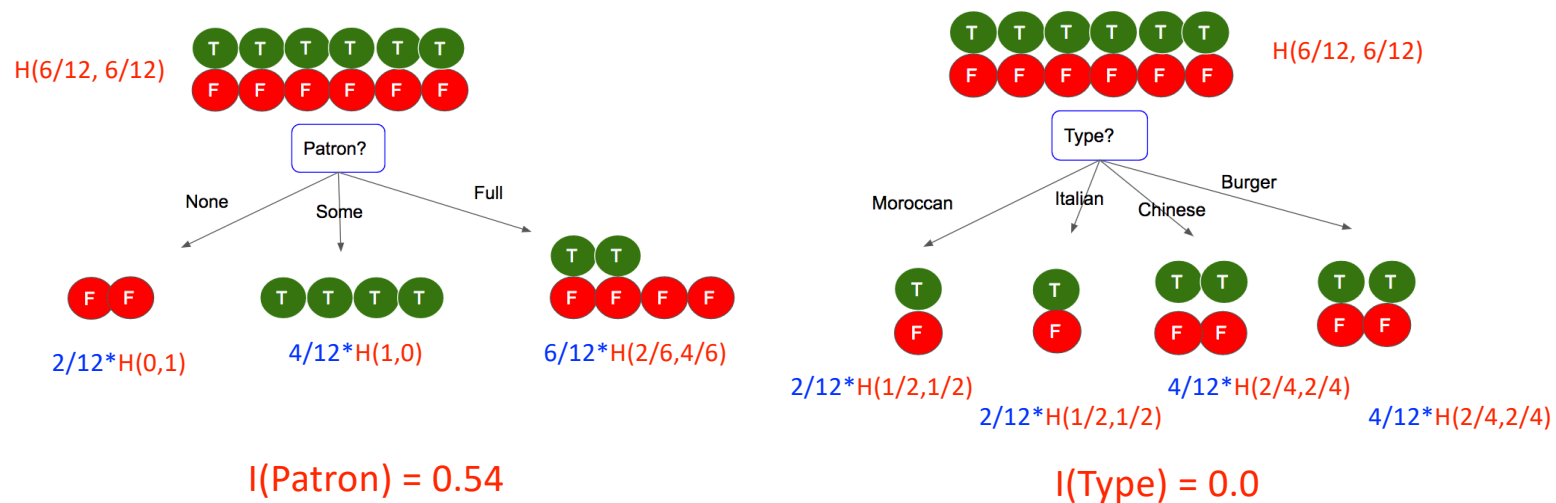
Information Gain

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



Information Gain

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



Other impurity measures

- p_c : probability of examples in class c
- S : subset of data examples

- **CART** algorithm uses the **Entropy**

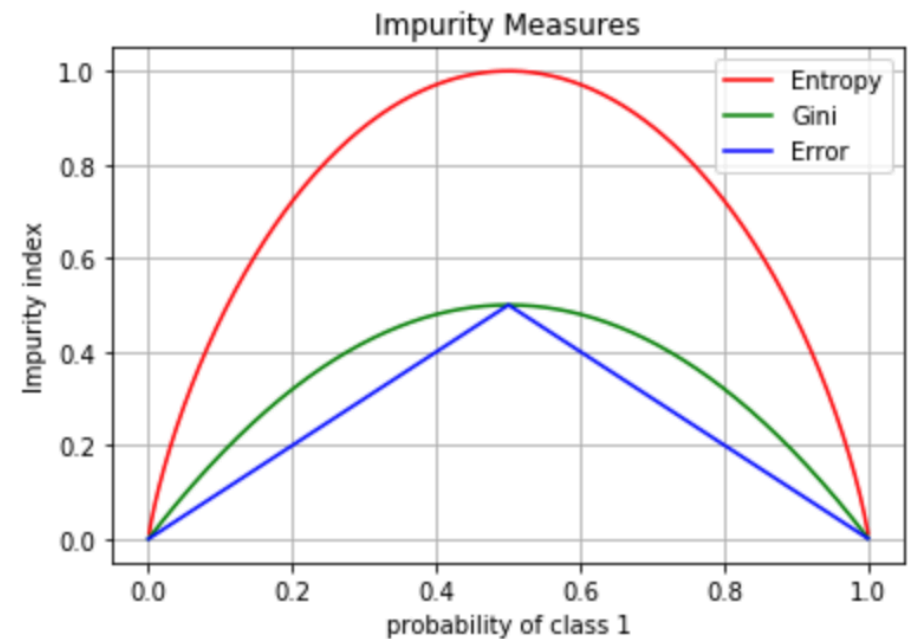
$$H(S) = - \sum_c p_c \log_2(p_c)$$

- Iterative **Dichotomiser ID3** and **C4.5** algorithms use **Gini Index**

$$G(S) = 1 - \sum_c p_c^2$$

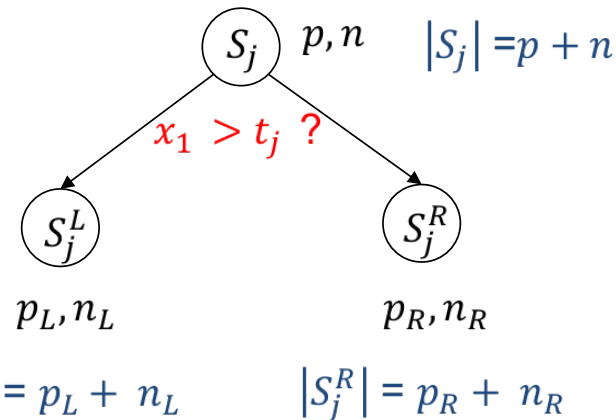
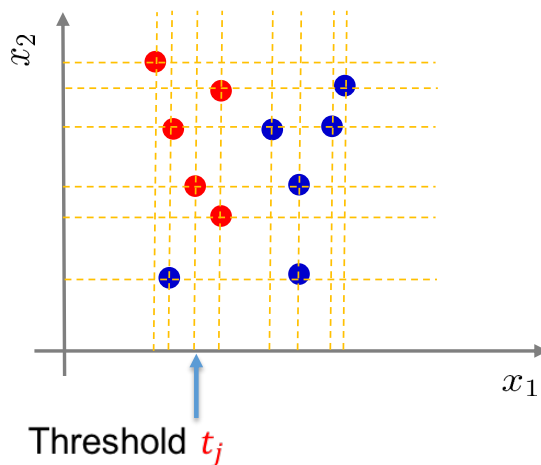
- One could also use the **Class Error**

$$E(S) = 1 - \max_c(p_c)$$



What if a feature is continuous (quantitative)?

$$I_j = H(S_j) - \sum_{i \in \{L,R\}} \frac{|S_j^i|}{|S_j|} H(S_j^i)$$



Note: Doing so, trees are almost **Binary**! Even if a feature is categorical (qualitative)

Decision Trees

- **Advantages**

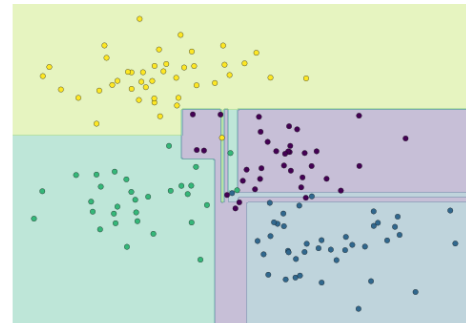
- Easy to interpret
- Deals with non linearity
- Handle qualitative features without the need to create fictive ones (one hot vector)
- Provide most important features (in terms of information gain)

Decision Trees

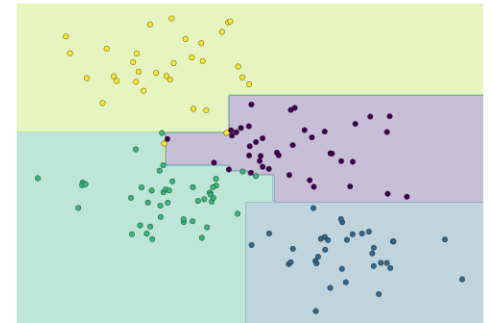
- **Disadvantages**

- Trees leads to **overfitting** (**high variance**): little change in little number of examples affect the whole tree.

DT on Data1



DT on Data2 = half of Data1



Decision Trees

- **Disadvantages**

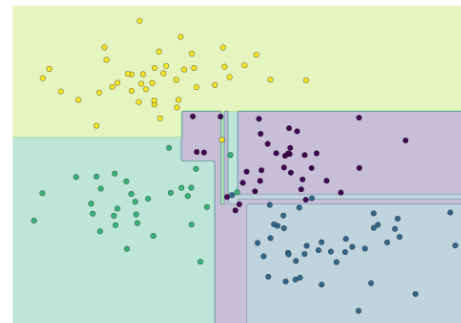
- Trees leads to **overfitting** (**high variance**): little change in little number of examples affect the whole tree.

- **Solution: Ensemble Methods**

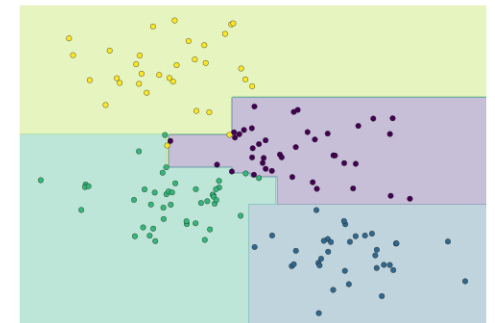
- **Bagging: Random Forest** (Leo Breiman 2001) consists of combining multiple **independent weak trees** to reduce variance.
- **Boosting** to reduce bias



DT on Data1



DT on Data2 = half of Data1



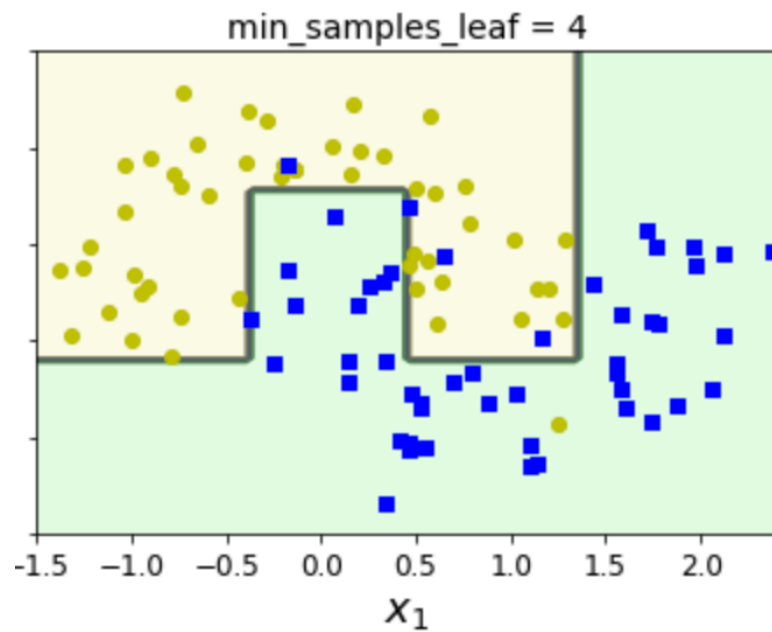
A tree **alone** will overfit.

However, it is clear that **in some places**, the two trees **together** produce consistent results

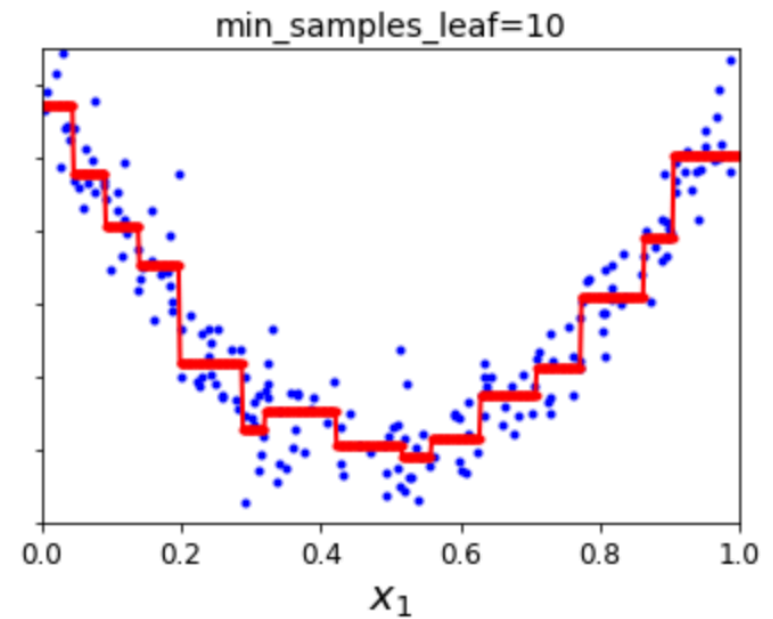
This idea comes from Bootstrapping (**Brad Efron 1979**): Given a set of **m** independent observations o_1, \dots, o_m , each with variance σ^2 , the variance of the mean **o** of the observations is given by σ^2/m .

Classification And Regression Trees

Classification (Decision) Trees



Regression Trees



Supervised Learning

- Linear Regression
- Logistic Regression
- Support Vector Machines
- Trees (Decision and Regression)
- **Random Forests**
- Boosting
- Artificial Neural Networks

Random Forest

Bagging

Bootstrap Aggregating

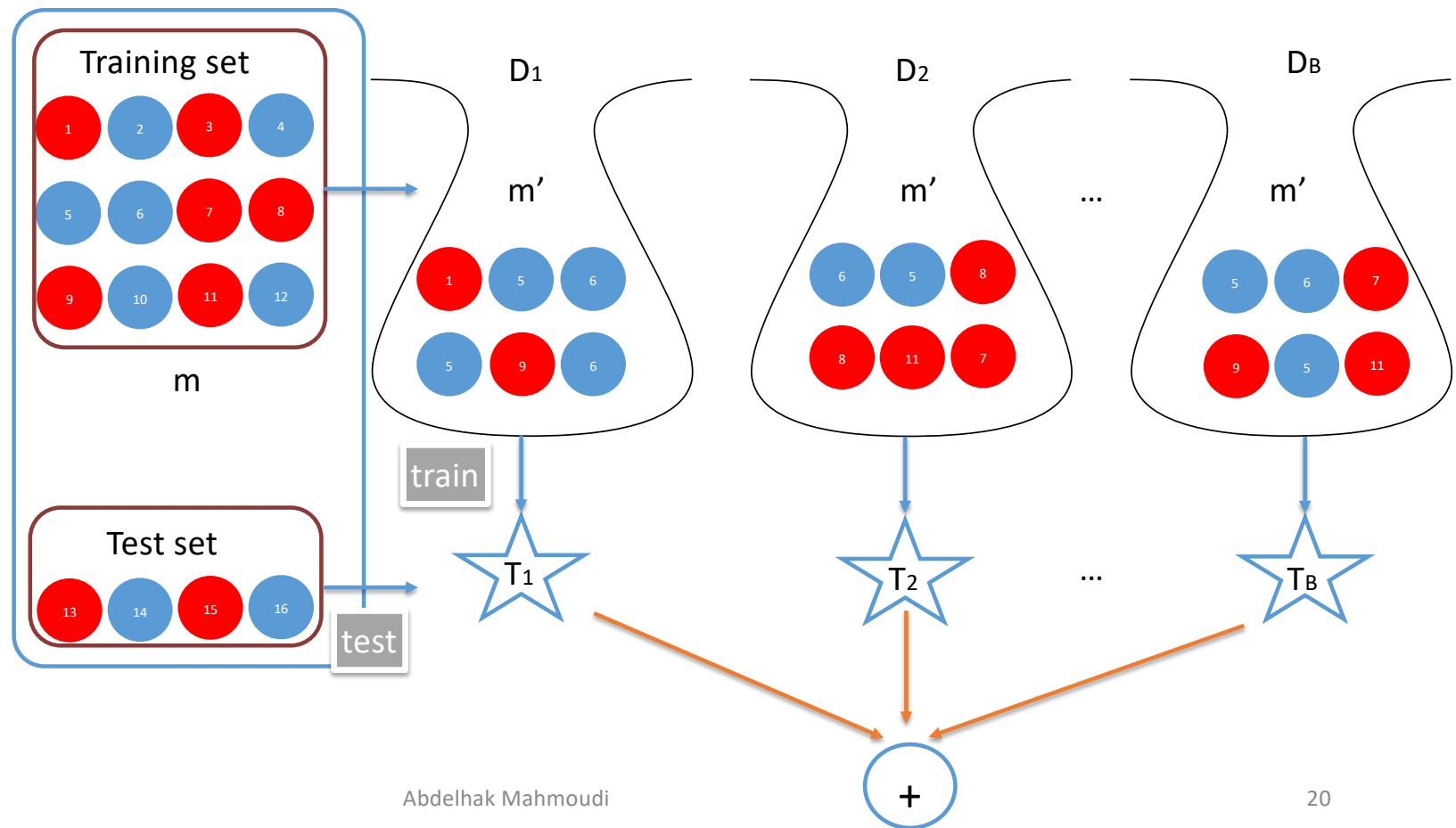
Training

Pick m' examples with replacement and train B trees

Testing

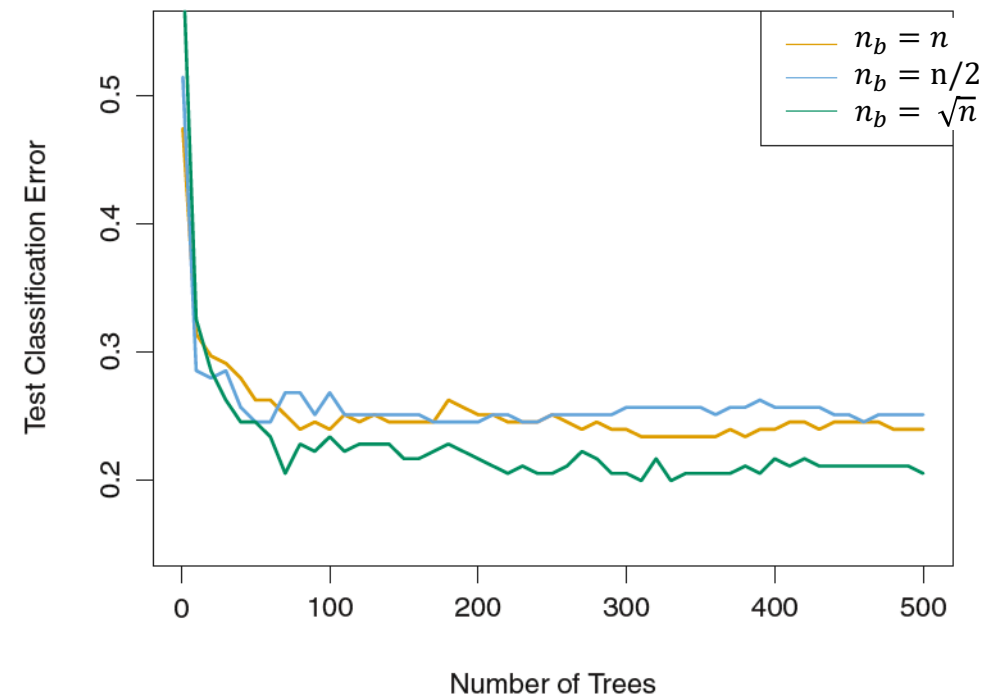
Regression: mean errors of all the B trees

Classification: vote



Random Forest

- **Problem:** Bagged trees will look quite similar to each other, so averaging them will not lead to much reduction of variance!
- **Solution:** Random Forest constructs multiple trees where each tree uses n_b random features from the n initial features (generally $n_b = \sqrt{n}$)
- $n_b = n \rightarrow$ Bagging case



Random Forest

- Both training and **prediction are very fast**, because of the simplicity of the underlying decision trees.
- Tasks can be straightforwardly **parallelized**, because the individual trees are entirely independent entities.
- The multiple trees allow for a **probabilistic** classification: a majority vote among estimators gives an estimate of the probability
- RF is a **Nonparametric** model, extremely flexible, and can thus perform well on tasks that are under-fit by other models.

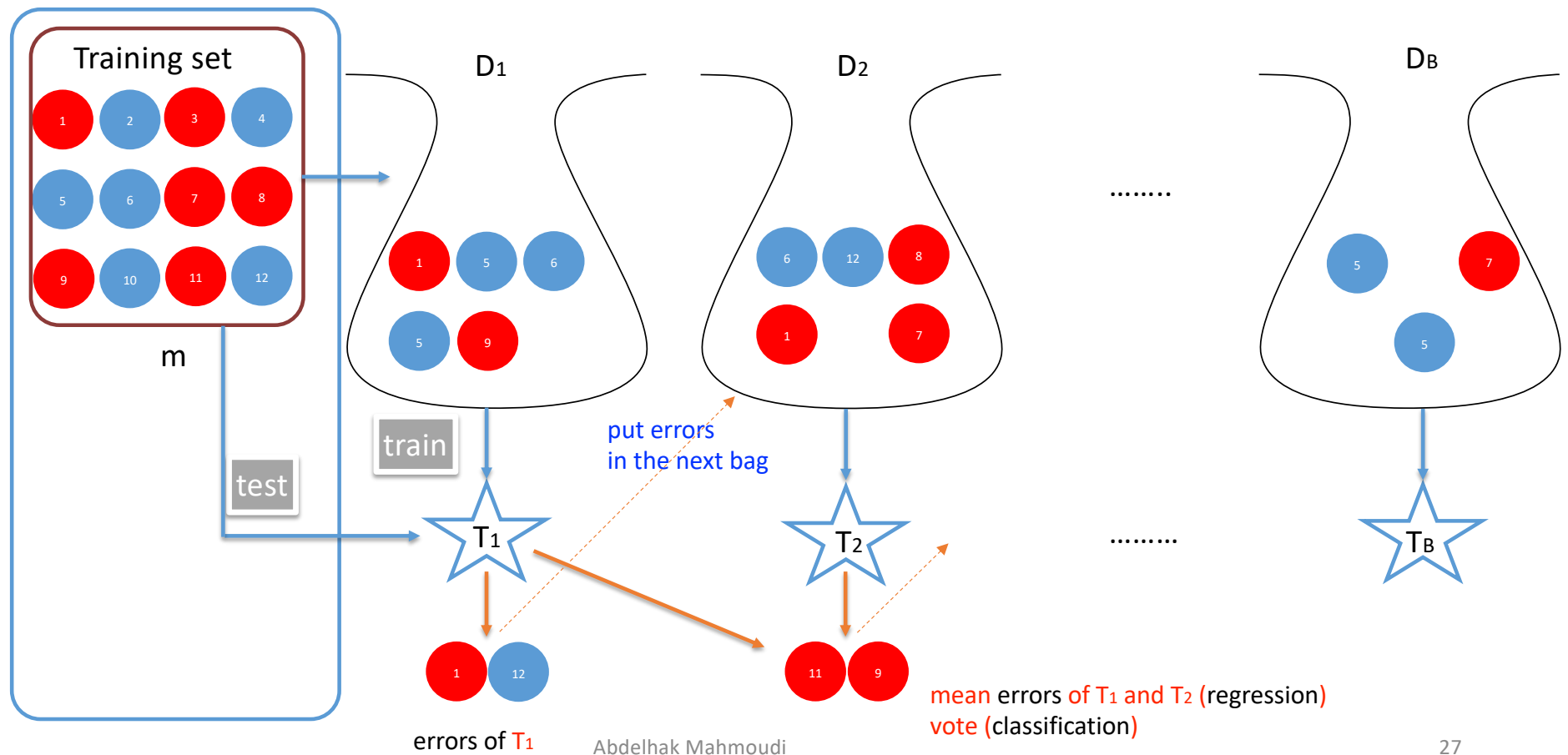
Random Forest

- Hyper-Parameters Tuning
 - d: Depth of the trees
 - B: number of Bags

Supervised Learning

- Linear Regression
- Logistic Regression
- Support Vector Machines
- Trees (Decision and Regression)
- Random Forests
- **Boosting**
- Artificial Neural Networks

Boosting



Boosting

- **Outperforms** RF
- **Smaller Trees** (depth = 1) are sufficient because the growth of a particular tree takes into account **preceding** trees.
- Smaller trees can aid in **interpretability**.
- **Boosting** (Freund & Schapire 1990)
- **Adaboost** (**A**daptive **B**oosting), 1996

