MALTAOMICS SUMMER SCHOOL

INTRODUCTION TO RANDOM FORESTS FOR VS

DAY 4 – 11:40-13:00

Dr Jean-Paul Ebejer
jean.p.ebejer@um.edu.mt

A MANUAL DATA MINING EXERCISE

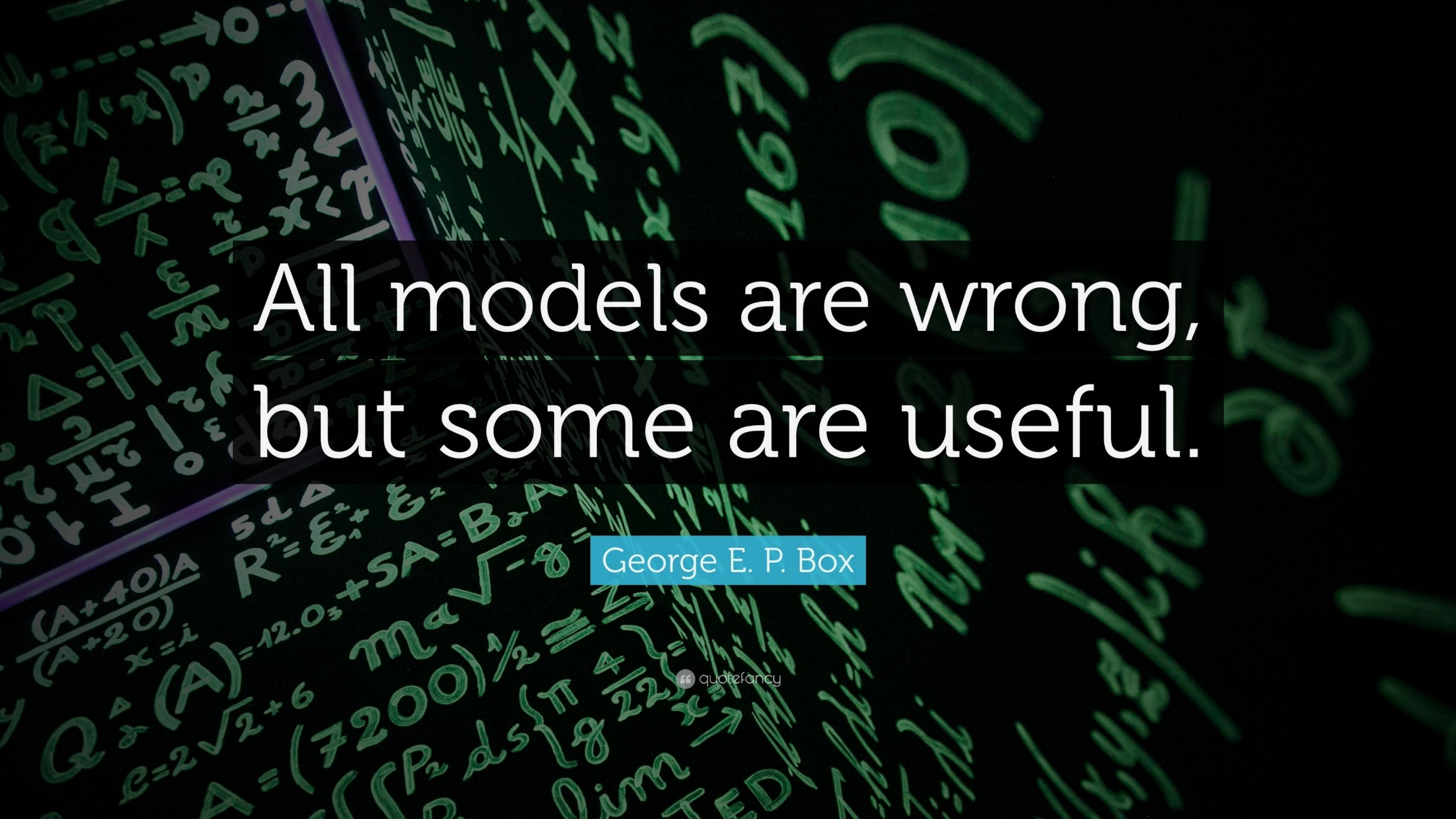
	$Feature_1$	$Feature_2$	$Feature_3$	$Feature_4$	$Feature_5$	$Feature_6$	$Feature_7$	$Feature_8$
Mol_1	1	1	0	1	1	0	1	0
Mol_2	1	1	0	0	1	0	1	0
Mol_3	0	1	1	1	0	0	1	1
Mol_4	0	0	0	1	0	0	0	0
Mol_5	0	1	1	0	1	0	1	1
Mol_6	0	0	0	0	1	1	1	1
Mol_7	1	0	0	1	0	1	1	0

Note: Can get better results using proper ML and data mining techniques (e.g. Random Forests, Artificial Neural Networks etc.)

BUILDING MODELS

- Models are simplified representations of reality
- Used for explanation and/or prediction
- Takes a set of inputs, gives you a set of outputs
- Could be treated as a black box. **DANGER**

Green? **Yes**
Tastes like Ice-Cream? **No**
Is it hot? **Yes**


All models are wrong,
but some are useful.
George E. P. Box
quotefancy

DECISION TREES

Root node

Is it sunny?

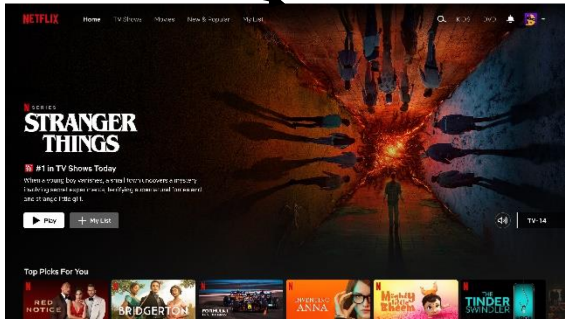
Sub-tree / branch

Yes

No

Are there sharks?

Decision node



Leaf/Terminal node

No

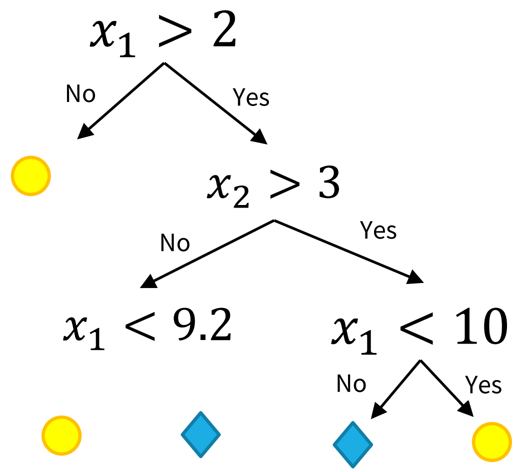
Yes

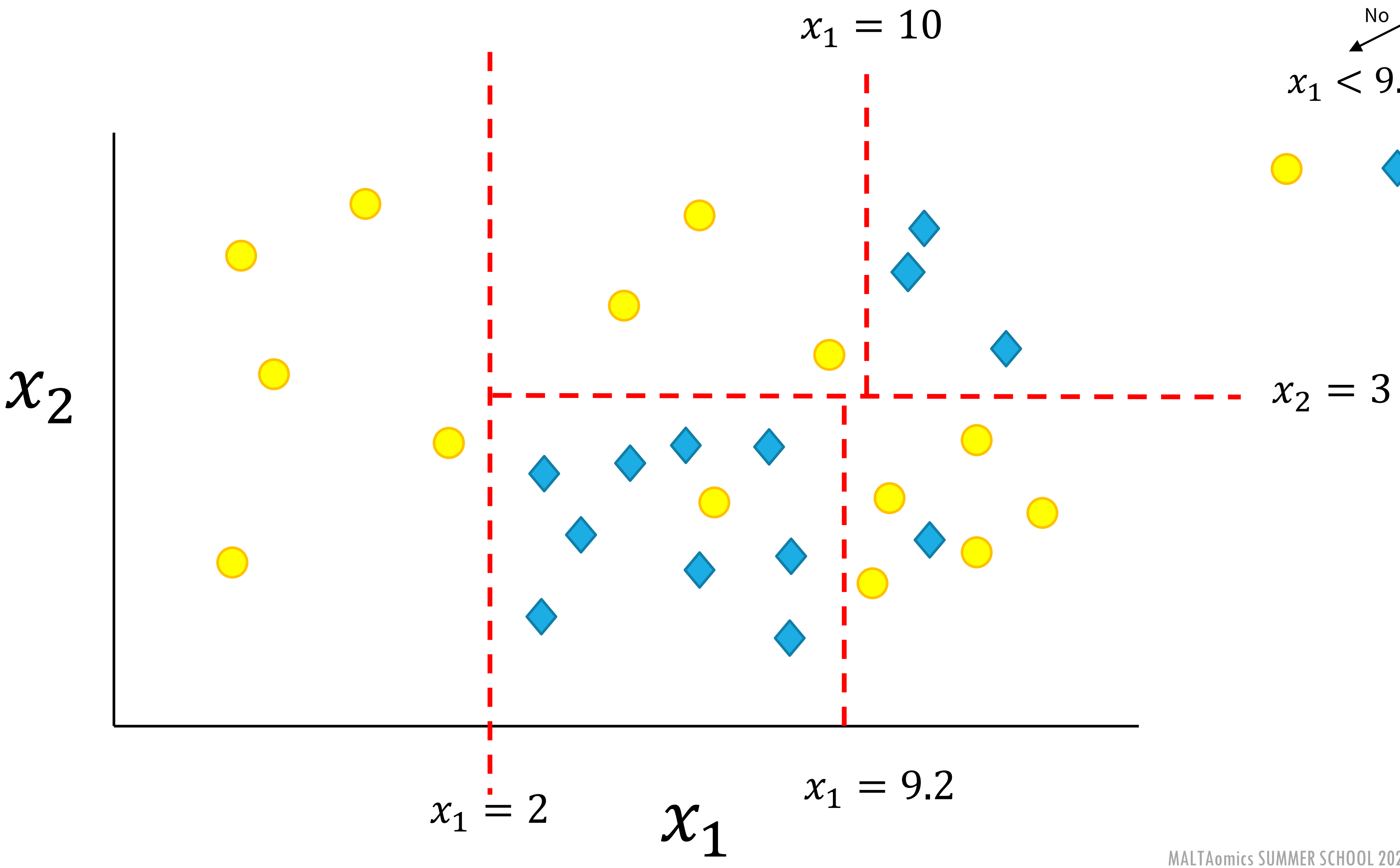
Splitting, Pruning

SOME QUESTIONS

- Which feature should we split on (at a specific height in the tree)?
  - How to compare different features?
- Which is the splitting criteria to use?
- How many child nodes should we employ?

AN ABSTRACTION OF A DECISION TREE





Note some impure nodes may exist!

HOW TO BUILD A TREE?

- Many algorithms exist (ID3, CART, C4.5, C5.0)
- All differ in detail, but similar in spirit (e.g., CART is binary tree)
- Recursive
- Basically (at each step) we need to select two things
  - A dimension (feature)
  - A split

WORKED EXAMPLE – DATASET

Features

Target

Observations (rows)

Weather Outlook	Water Temp (C)	Sun Peak Hours (11-3pm)	Swim?
Sunny	25	Peak	No
Sunny	26	Not Peak	Yes
Sunny	18	Not Peak	No
Sunny	20	Peak	No
Cloudy	23	Peak	No
Cloudy	24	Not Peak	Yes
Cloudy	21	Not Peak	No
Rainy	16	Peak	No
Rainy	23	Not Peak	No
Rainy	20	Not Peak	No

ALGORITHM

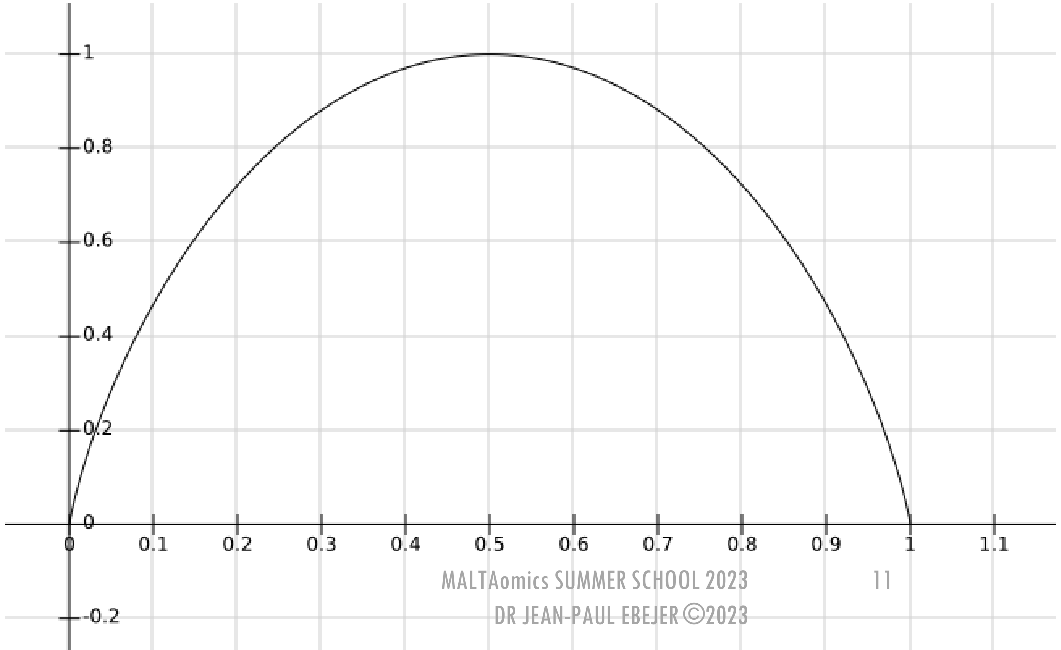
- Decision rules to be applied found using:
  - Entropy
  - Information Gain
- At each level of the tree, feature with maximum "gain ratio" will be the decision rule

BUT HOW TO SPLIT THE TREE? (WHAT IS ENTROPY?)

- Use Entropy (or Gini or …)
  - $E(S) = -p_{+}log_{2}p_{+} - p_{-}log_{2}p_{-}$

$$\mathrm{H}(T) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

- p_{+} proportion of positive cases in collection S
  - p_{-} -ve proportion
- When is entropy at its highest?



BUT HOW TO SPLIT THE TREE? (II)

- Information Gain (expected reduction in entropy)
- Decides which feature to pick
- The feature with **most** entropy reduction is best choice, i.e., most gain
- $IG(S,F) = Entropy(S) - \sum_{v \in F} \frac{|S_v|}{|S|}.Entropy(S_v)$

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|a)}^{\text{Weighted Sum of Entropy (Children)}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_a p(a) \sum_{i=1}^{J} - \Pr(i|a) \log_2 \Pr(i|a)$$

WORKED EXAMPLE (CONTD.)

Weather Outlook	Water Temp (C)	Sun Peak Hours (11-3pm)	Swim?
Sunny	25	Peak	No
Sunny	26	Not Peak	Yes
Sunny	18	Not Peak	No
Sunny	20	Peak	No
Cloudy	23	Peak	No
Cloudy	24	Not Peak	Yes
Cloudy	21	Not Peak	No
Rainy	16	Peak	No
Rainy	23	Not Peak	No
Rainy	20	Not Peak	No

- In our dataset, 2 out of 10 times we decide to **swim**, while 8 out of 10 times we decide **not to swim**
- $Entropy(Decision) = \sum -p(I).\log_2 p(I) = - p(Yes).\log_2 p(Yes) - p(No).\log_2 p(No)$
  - Note “**Decision**” here refers to the decision of whether to swim or not
- $Entropy(Decision) = -{}^{2}/_{10} \log_2 {}^{2}/_{10} - {}^{8}/_{10} \log_2 {}^{8}/_{10} = 0.722$

WORKED EXAMPLE – SUN PEAK HOURS (11-15)

Weather Outlook	Water Temp (C)	Sun Peak Hours (11-3pm)	Swim?
Sunny	25	Peak	No
Sunny	26	Not Peak	Yes
Sunny	18	Not Peak	No
Sunny	20	Peak	No
Cloudy	23	Peak	No
Cloudy	24	Not Peak	Yes
Cloudy	21	Not Peak	No
Rainy	16	Peak	No
Rainy	23	Not Peak	No
Rainy	20	Not Peak	No

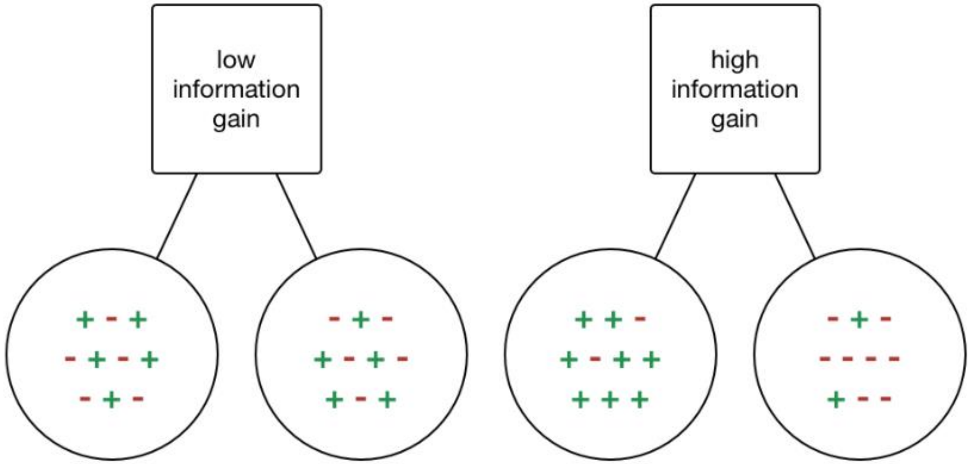
- "Sun Peak Hours" is a (dichotomous) nominal value
  - Possible values are Yes (Peak)/No (Not Peak)
- 6 "Not peak", 4 "Peak" instances
- Gain(Decision, PeakHrs)=$E(Decision) - \sum(p(Decision|PeakHrs).E(Decision|PeakHrs)$
- $\sum(p(Decision|PeakHrs).E(Decision|PeakHrs) = p(Decision|PeakHrs = Not\ peak).E(Decision|PeakHrs = Not\ peak) + p(Decision|PeakHrs = Peak).E(Decision|PeakHrs = Peak)$
- When Peak (4 instances), in 0 instances I **swim**, and 4 instances I do **not swim**
- When Not Peak (6 instances), in 2 instances I **swim**, and 4 instances I do **not swim**

WORKED EXAMPLE – SUN PEAK HOURS (11-15)

- $E(Decision|Peak) = -p(No).\log_2 p(No) - p(Yes).\log_2 p(Yes) = -\left(\frac{4}{4}\right)\log_2\left(\frac{4}{4}\right) - \left(\frac{0}{4}\right)\log_2\left(\frac{0}{4}\right) = 0.000$
- $E(Decision|Not\ Peak) = -p(No).\log_2 p(No) - p(Yes).\log_2 p(Yes) = -\left(\frac{4}{6}\right)\log_2\left(\frac{4}{6}\right) - \left(\frac{2}{6}\right)\log_2\left(\frac{2}{6}\right) = 0.918$

WORKED EXAMPLE – SUN PEAK HOURS INFORMATION GAIN

- Compute Gain(Decision, PeakHrs)
- $Information\ Gain(Decision, PeakHrs) = E(Decision) - \sum(p(Decision|PeakHrs).E(Decision|PeakHrs))$
- $Gain(Decision, PeakHrs) = 0.722 - \frac{4}{10}.0.000 - \frac{6}{10}.0.918 = 0.171$



We want to split on the feature which gives us the maximum information gain!

WORKED EXAMPLE – OUTLOOK

Weather Outlook	Water Temp (C)	Sun Peak Hours (11-3pm)	Swim?
Sunny	25	Peak	No
Sunny	26	Not Peak	Yes
Sunny	18	Not Peak	No
Sunny	20	Peak	No
Cloudy	23	Peak	No
Cloudy	24	Not Peak	Yes
Cloudy	21	Not Peak	No
Rainy	16	Peak	No
Rainy	23	Not Peak	No
Rainy	20	Not Peak	No

- "Outlook" is a nominal variable
  - Possible values are Sunny/Cloudy/Rainy
- 4 instances of Sunny, 3 instances of Cloudy and Rainy outlook
- Gain(Decision, Outlook)=$E(Decision) - \sum(p(Decision|Outlook).E(Decision|Outlook)$
- $p(Decision|Outlook = Sunny).E(Decision|Outlook = Sunny) + p(Decision|Outlook = Rainy).E(Decision|Outlook = Rainy) + p(Decision|Outlook = Cloudy).E(Decision|Outlook = Cloudy)$
- When Sunny (4 instances), in 1 instance I **swim**, and 3 instances I do **not swim**
- When Rainy (3 instances), in 0 instances I **swim**, and 3 instances I do **not swim**
- When Cloudy (3 instances), in 1 instances I swim, and in 2 instances I do **not** swim

WORKED EXAMPLE – OUTLOOK

- $E(Decision|Sunny) =$
  $- p(No).\log_2 p(No) - p(Yes).\log_2 p(Yes) = -\left(\frac{3}{4}\right)\log_2\left(\frac{3}{4}\right) - \left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right) = 0.811$
- $E(Decision|Rainy) =$
  $- p(No).\log_2 p(No) - p(Yes).\log_2 p(Yes) = -\left(\frac{3}{3}\right)\log_2\left(\frac{3}{3}\right) - \left(\frac{0}{3}\right)\log_2\left(\frac{0}{3}\right) = 0.000$
- $E(Decision|Cloudy) =$
  $- p(No).\log_2 p(No) - p(Yes).\log_2 p(Yes) = -\left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) - \left(\frac{1}{3}\right)\log_2\left(\frac{1}{3}\right) = 0.918$
- $Gain(Decision, Outlook) = 0.722 - \frac{4}{10}.0.811 - \frac{3}{10}.0 - \frac{3}{10}.0.918 = 0.122$

WORKED EXAMPLE – WATER TEMP

- "Water Temp." is a continuous variable
  - Possible values range from 16 to 26
- Need to convert to nominal variable
  - How? C4.5 suggests a binary split on a threshold value
- But how to find this threshold value?
  - Try many values, choose one that maximizes Gain
- Let us compute information gain for all thresholds from 16.5 to 25.5

water_temp	swim
16	No
18	No
20	No
20	No
21	No
23	No
23	No
24	Yes
25	No
26	Yes

A SPECIFIC EXAMPLE

water_temp	swim
16	No
18	No
20	No
20	No
21	No
23	No
23	No
24	Yes
25	No
26	Yes

- $E(Decision|WaterTemp < 16.5) =$
  $-\,p(No).\log_2 p(No) - p(Yes).\log_2 p(Yes) = -\left(\frac{1}{1}\right)\log_2\left(\frac{1}{1}\right) - \left(\frac{0}{1}\right)\log_2\left(\frac{0}{1}\right) = 0.000$
- $E(Decision|WaterTemp > 16.5) =$
  $-\,p(No).\log_2 p(No) - p(Yes).\log_2 p(Yes) = -\left(\frac{7}{9}\right)\log_2\left(\frac{7}{9}\right) - \left(\frac{2}{9}\right)\log_2\left(\frac{2}{9}\right) = 0.764$
- $Gain(Decision, WaterTemp) = 0.722 - \frac{9}{10}.0.764 - \frac{1}{10}.0.000 = 0.034$
- Repeat this for water temp. of 17.5, 18.5, 19.5, 20.5, 21.5, etc.
- Max gain is at 23.5 C

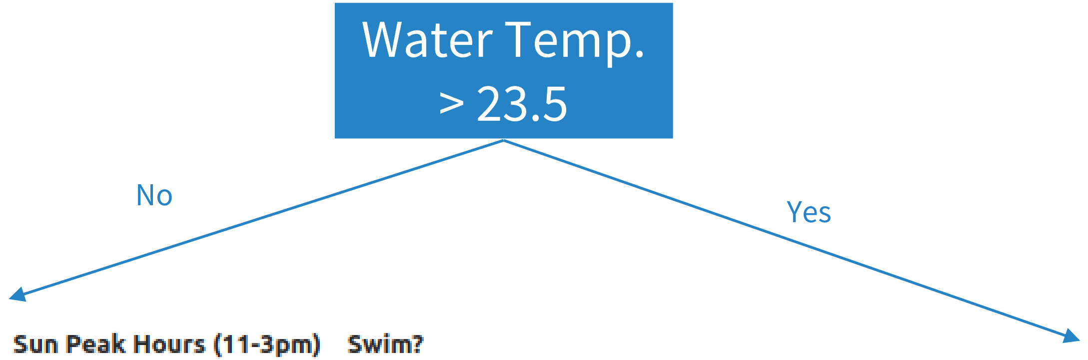
Water Temp	**Gain**
16.5	**0.034**
17.5	**0.034**
18.5	**0.073**
19.5	**0.073**
20.5	**0.171**
21.5	**0.236**
22.5	**0.236**
23.5	**0.446**
24.5	**0.087**
25.5	**0.269**

INFORMATION GAIN SUMMARY

Attribute	Gain
Outlook	0.122
Water Temp. (23.5)	**0.446**
Sun Peak Hours	0.171

- Water Temp 23.5 will be the root decision node
- Many more metrics exist (SplitInfo, Gain Ratio, Gini index etc)

STEP FORWARD

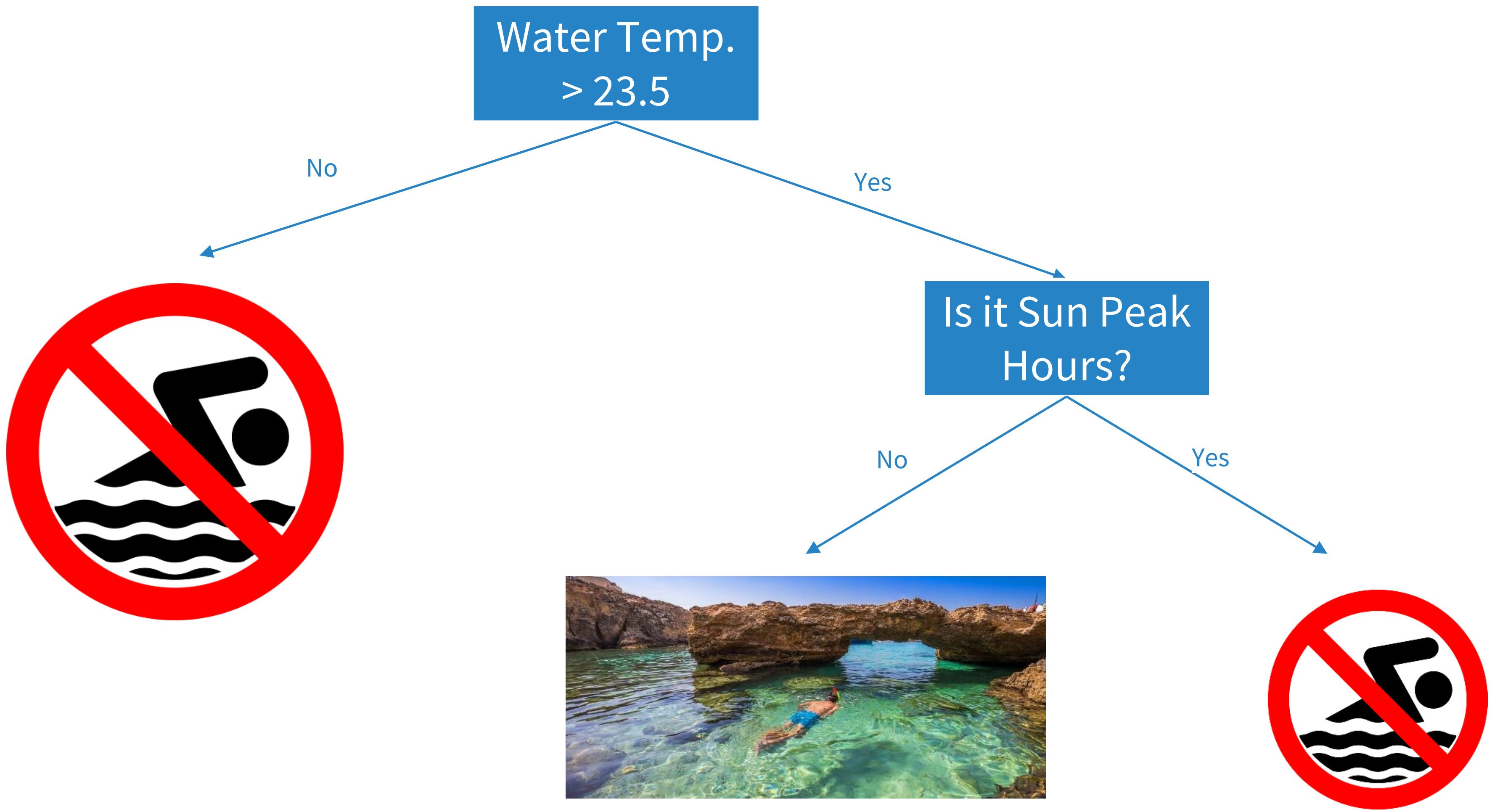


Weather Outlook	Water Temp (C)	Sun Peak Hours (11-3pm)	Swim?
Sunny	18	Not Peak	No
Sunny	20	Peak	No
Cloudy	23	Peak	No
Cloudy	21	Not Peak	No
Rainy	16	Peak	No
Rainy	23	Not Peak	No
Rainy	20	Not Peak	No

Weather Outlook	Water Temp (C)	Sun Peak Hours (11-3pm)	Swim?
Sunny	25	Peak	No
Sunny	26	Not Peak	Yes
Cloudy	24	Not Peak	Yes

- Redo the entropy and information gain calculations on this subset of the data
- In ID3 you have as many children as values (e.g. outlook – sunny, cloudy, rain)
- Keep on splitting until all leaves are pure

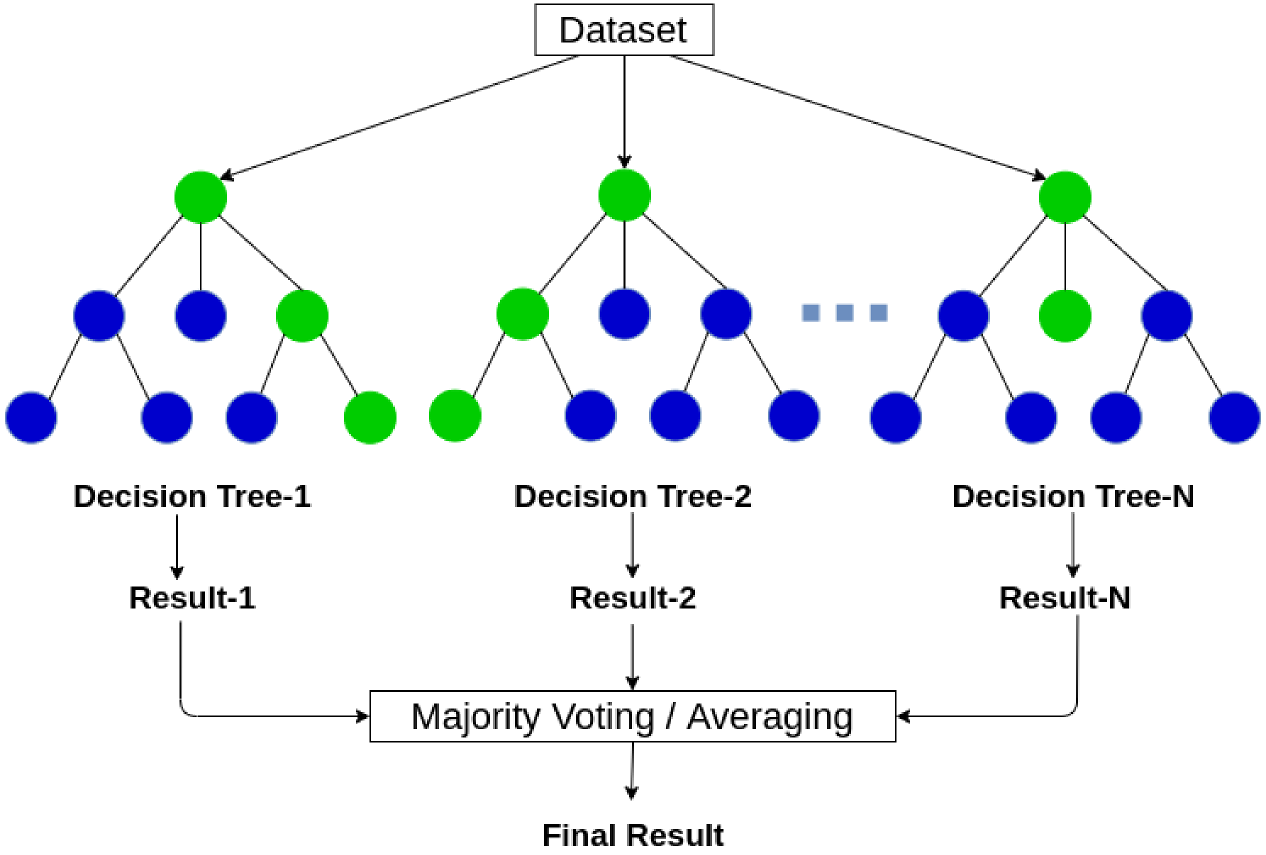
FINAL DECISION TREE



RANDOM FOREST

- Made up of a set of (uncorrelated) decision trees
  - Hence "ensemble" method
- Each Decision Tree trained on (random with replacement) subset of training data
  - Bagging (bootstrap aggregating)
  - Only a random subset of features are considered for splitting nodes
- Many predictions (one per tree) are then aggregated into a single result
  - E.g. Majority class

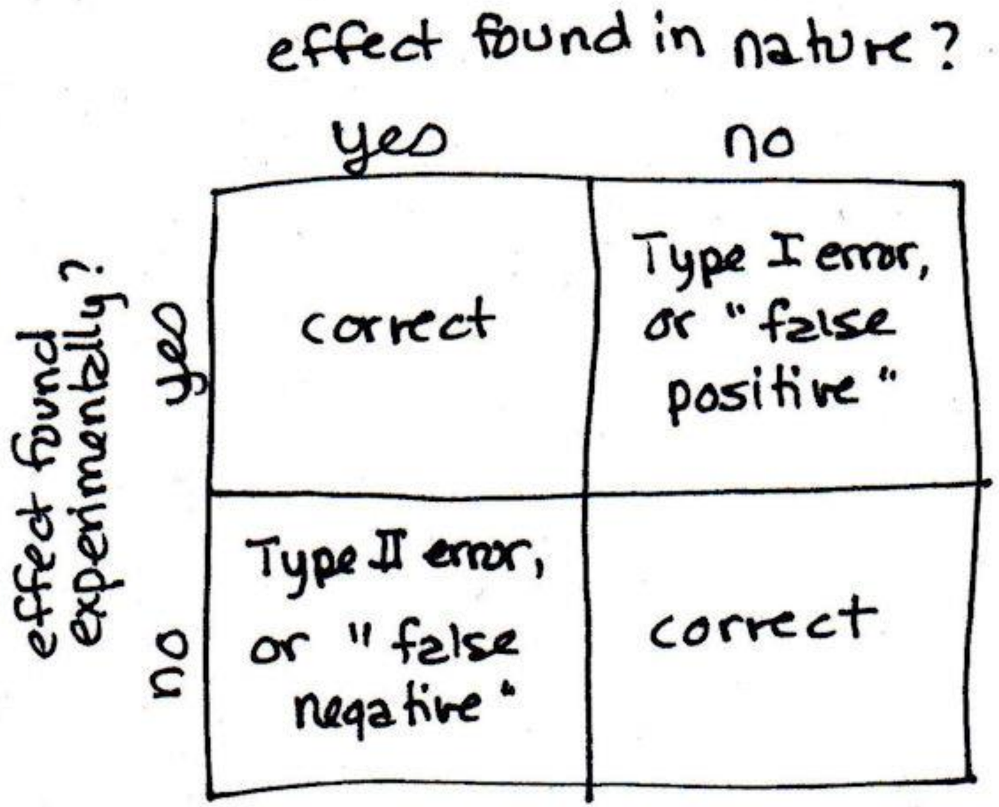
RANDOM FOREST – VISUALIZATION



HOW TO EVALUATE THE MODEL

- Split your dataset in two (three) sets
  - Training
  - (Validation)
  - Testing
- Usual split is 80% for training (i.e. building the trees) and 20% of testing
- Golden Machine Learning rule: **the testing data must be unseen during training**
- Variations exist; cross-validation

CLASSIFYING ERRORS



		effect found in nature? yes	effect found in nature? no
effect found experimentally?	yes	correct	Type I error, or "false positive"
effect found experimentally?	no	Type II error, or "false negative"	correct

Type I Error
False Positive

Type II Error
False Negative

EVALUATION OF THE MODEL

- Contingency Table (actual vs predicted)
- 100 test set messages to classify as spam/ham
  - We know the real class of these messages (50/50)

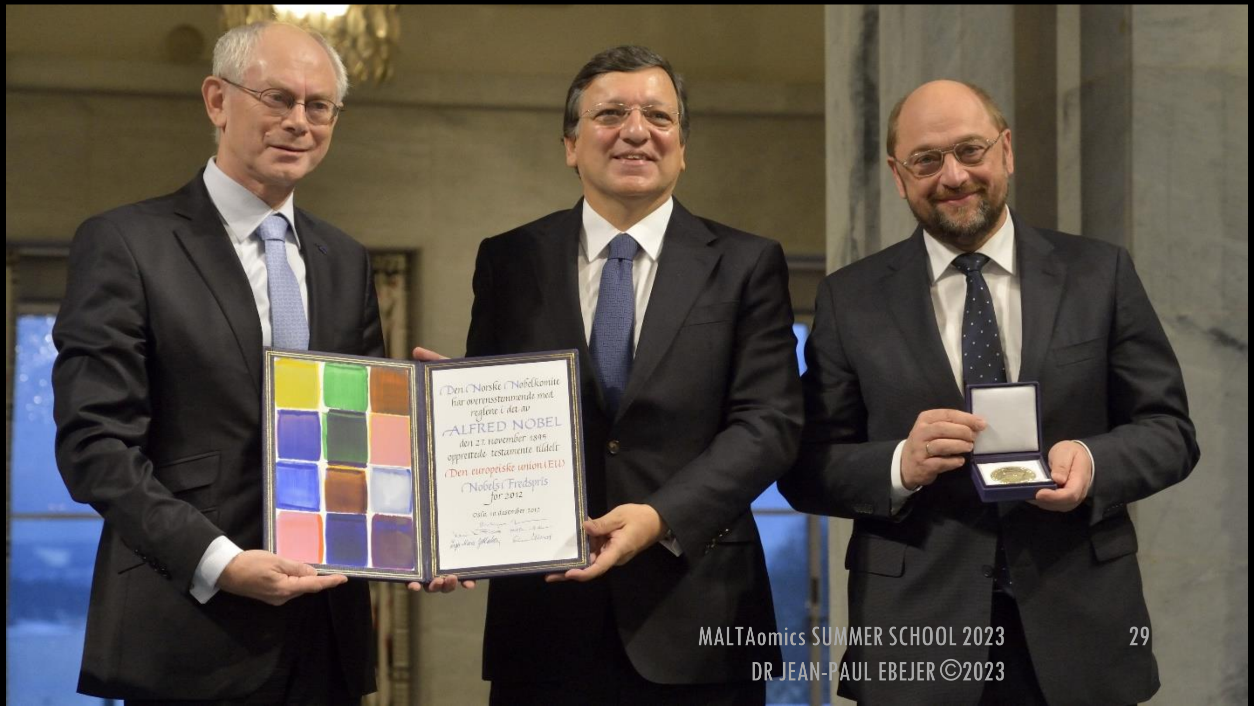
	Actual SPAM	Actual ~SPAM	
Predicted SPAM	46 (TP)	5 (FP)	51
Predicted ~SPAM	10 (FN)	39 (TN)	49
	56	44	100

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = 0.85$
- Accuracy is, counterintuitively, a bad idea

WRITE A PREDICTOR FOR NOBEL PRIZE WINNERS


```python
def predict_nobel_winner(name):
        return False
```

99.999987% Accurate



PROBLEM WITH ACCURACY

- Terrible at unbalance sets (one class is much larger or smaller than the rest)
- Use Precision and Recall instead
- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- Use F-measure to combine the two in a single metric

F SCORE

- $F_{\beta} = \frac{(\beta^2+1)PR}{\beta^2 P+1}$
- Most frequent use is when β=1, and Precision and Recall are equally balanced
  - Called F_1 measure

RECAP

- What is a model
- How a decision tree works
- What is a random forest
- Evaluation of a Machine Learning Model

PRACTICAL