

Decision Trees

*** *More slides here*

https://github.com/gSchool/DSI_Lectures/tree/master/non-parametric-learners

Parametric vs NonParametric

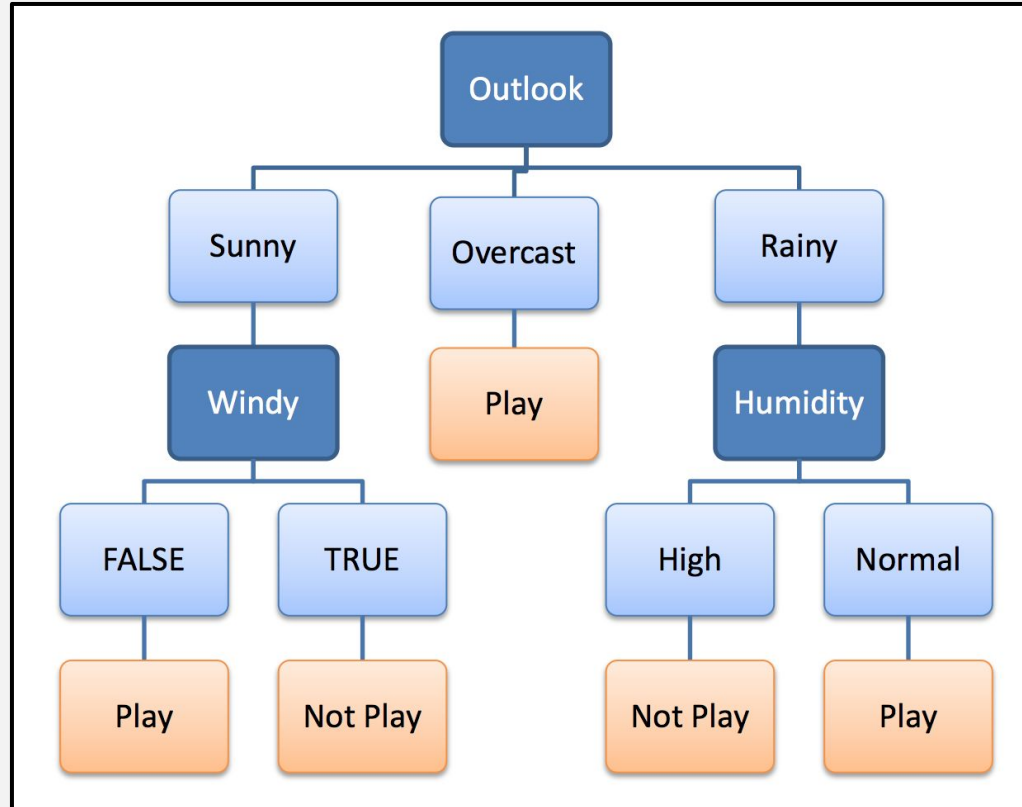
Parametric models - models that simplify the function to a known form are called parametric machine learning models.

Non-parametric models - models that do not make strong assumptions about the form of the mapping function are called nonparametric machine learning models. By not making assumptions, they are free to learn any functional form from the training data.

Will I play golf today?

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Decision Tree



Why Decision Trees

- Easily interpretable
- Handles missing values and outliers
- Non-parametric/non-linear/discontinuity/
model complex phenomenon
- Computationally *cheap* to *predict*
- Can handle irrelevant features
- Mixed data (nominal and continuous)

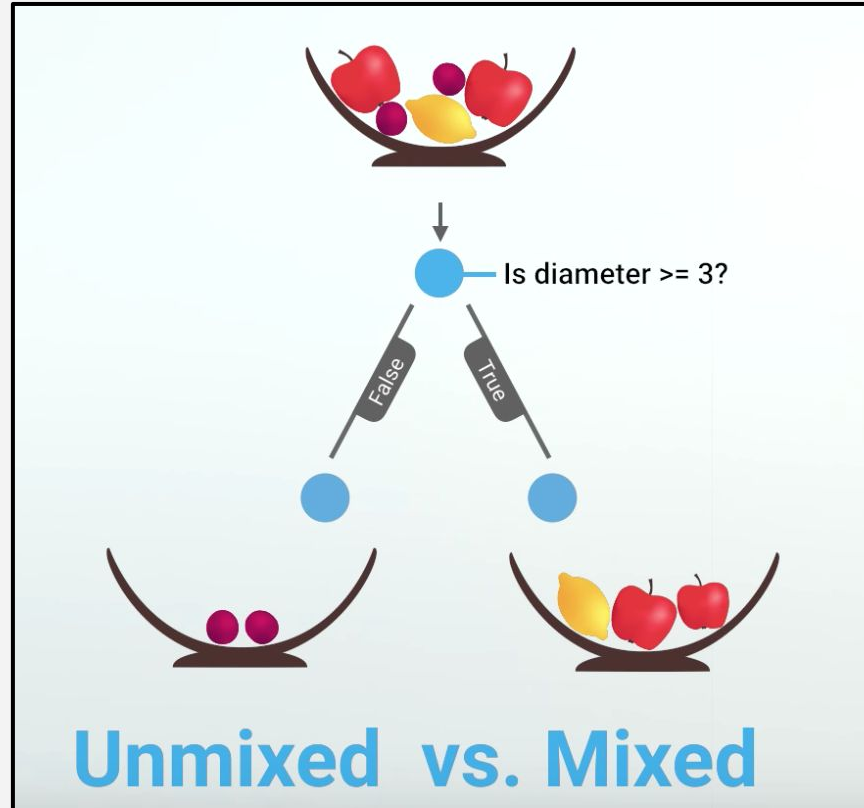
Why not Decision Trees

- Computationally *expensive* to *train*
- Very easy to overfit

How to build a Decision Tree

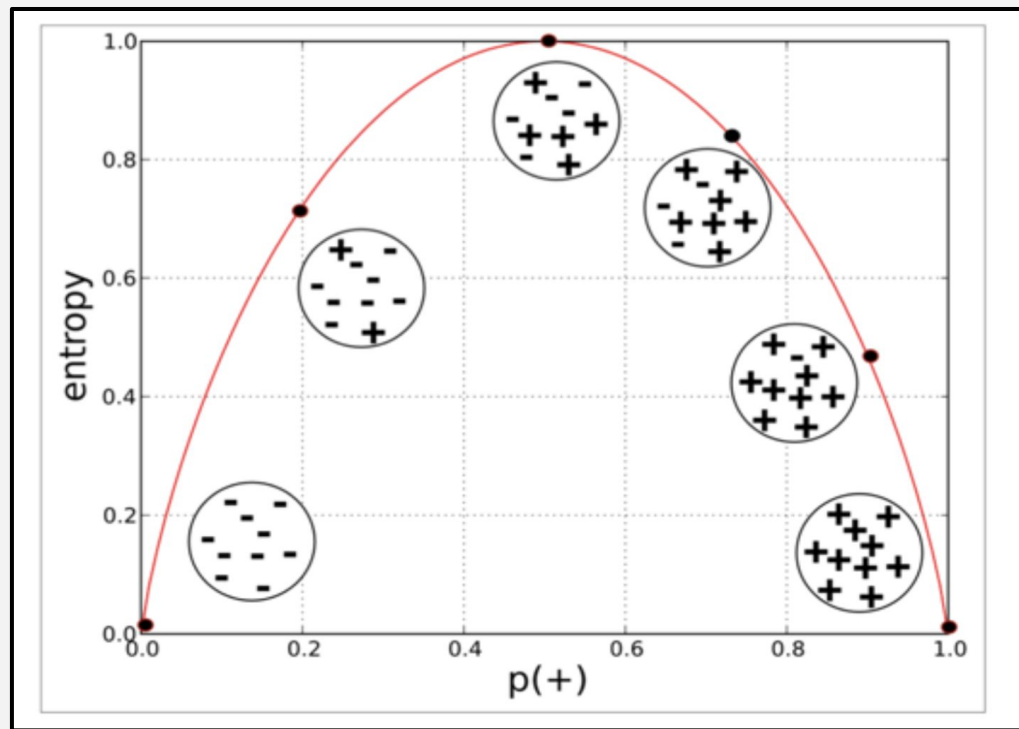
- For binary split tree:
 - For a categorical variable, choose either value or not value (e.g. sunny or not sunny)
 - For a continuous variable, choose a threshold and do $>$ or \leq the value (e.g. temperature < 75 or ≥ 75)
- To measure how good the split is:
 - Information gain
 - Gini impurity

Splitting the data






Entropy

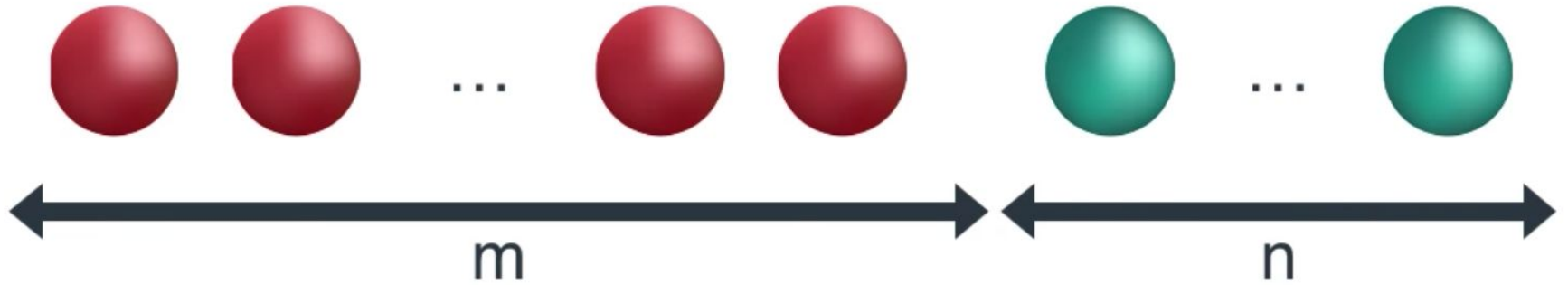
- A measure of disorder
- An indicator of how messy your data is
- The amount of uncertainty in a system



Probabilities & Entropy

	P(red)	P(blue)	P(winning)	$-\log_2(P(\text{winning}))$	Entropy
	1	0	$1 \times 1 \times 1 \times 1 = 1$	$0+0+0+0$	0
	0.75	0.25	$0.75 \times 0.75 \times 0.75 \times 0.25 = 0.105$	$0.415+0.415+0.415+2$	0.81
	0.5	0.5	$0.5 \times 0.5 \times 0.5 \times 0.5 = 0.0625$	$1+1+1+1$	1

Generalized form for Entropy



$$\text{Entropy} = -\frac{m}{m+n} \log_2 \left(\frac{m}{m+n} \right) - \frac{n}{m+n} \log_2 \left(\frac{n}{m+n} \right)$$

Comparing Information Gain Types

Shannon Entropy - Measures the **diversity** of a sample

$$H(X) = - \sum_i p_i \log_2(p_i)$$

Information Gain Using Shannon Entropy

$$\text{IG}(S, C) = H(S) - \sum_{C_i \in C} \frac{|C_i|}{|S|} H(C_i)$$

Gini Index - Measures the **probability of misclassifying** a single element if it was randomly labeled according to the distribution of classes in the sample

$$\text{Gini}(S) = 1 - \sum_{i \in S} p_i^2$$

Information Gain Using Gini Index

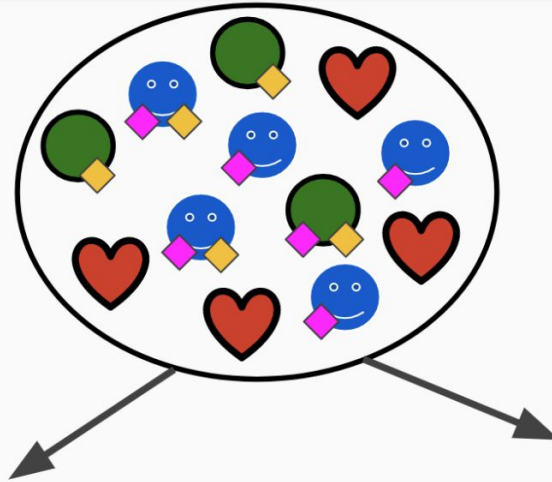
$$\text{IG}(S, C) = \text{Gini}(S) - \sum_{C_i \in C} \frac{|C_i|}{|S|} \text{Gini}(C_i)$$

Splitting 1

Features:

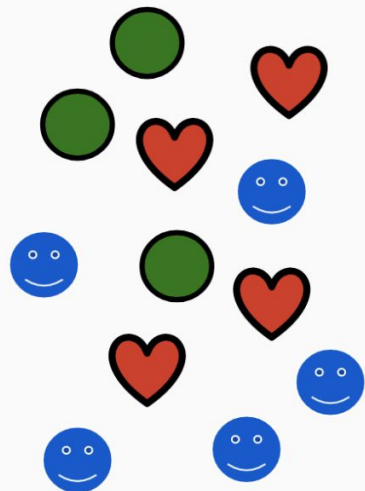


Target Variable
Categories:



What feature can we split on to maximize information gain?

Splitting 2



Entropy Equation:

Proportion of class i
in the sample

$$H(X) = - \sum_i p_i \log_2(p_i)$$

Estimate:

$$P(\text{green circle}) = 3/12 = 0.25$$

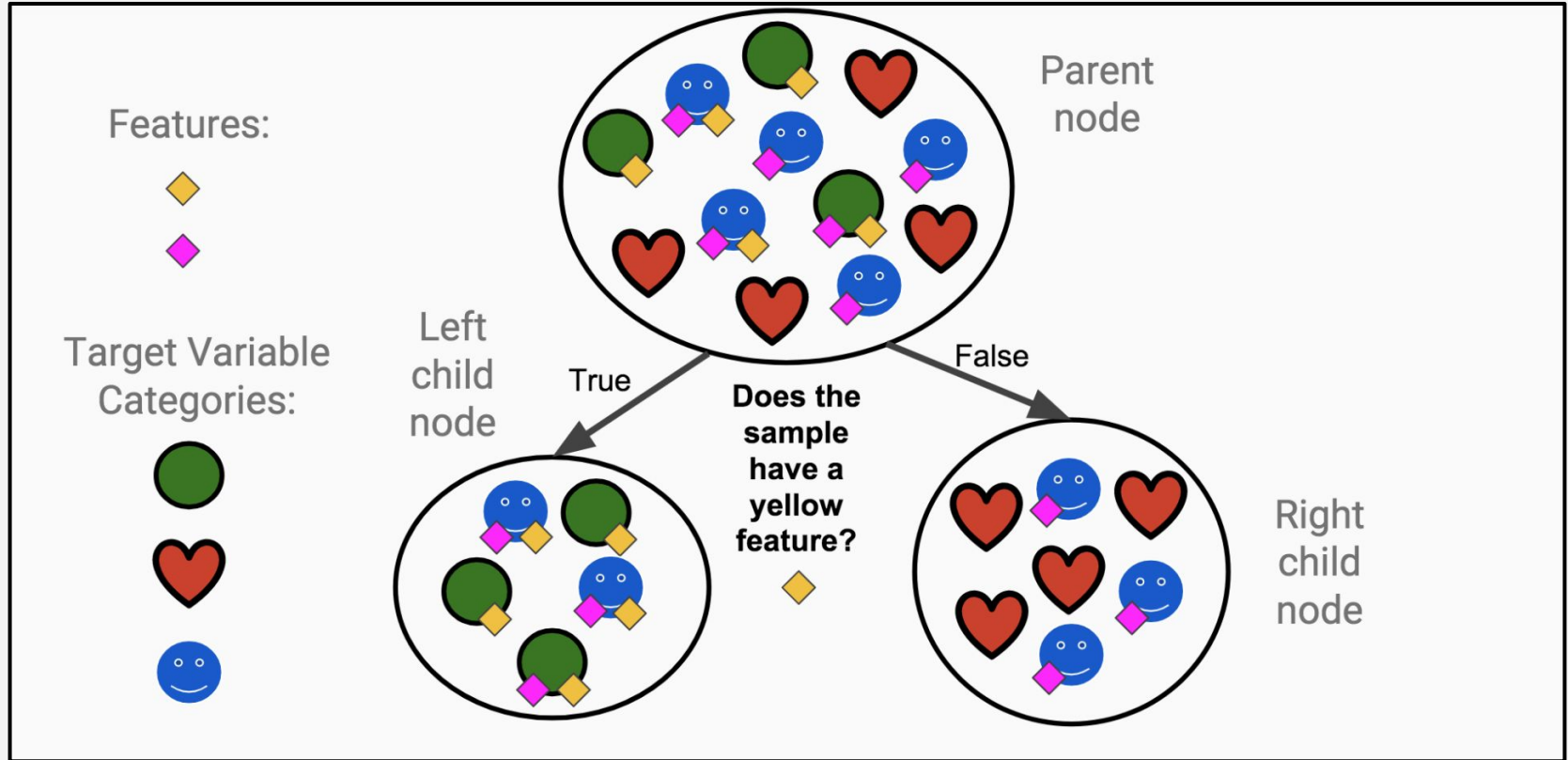
$$P(\text{red heart}) = 4/12 = 0.33$$

$$P(\text{blue smiley}) = 5/12 = 0.42$$

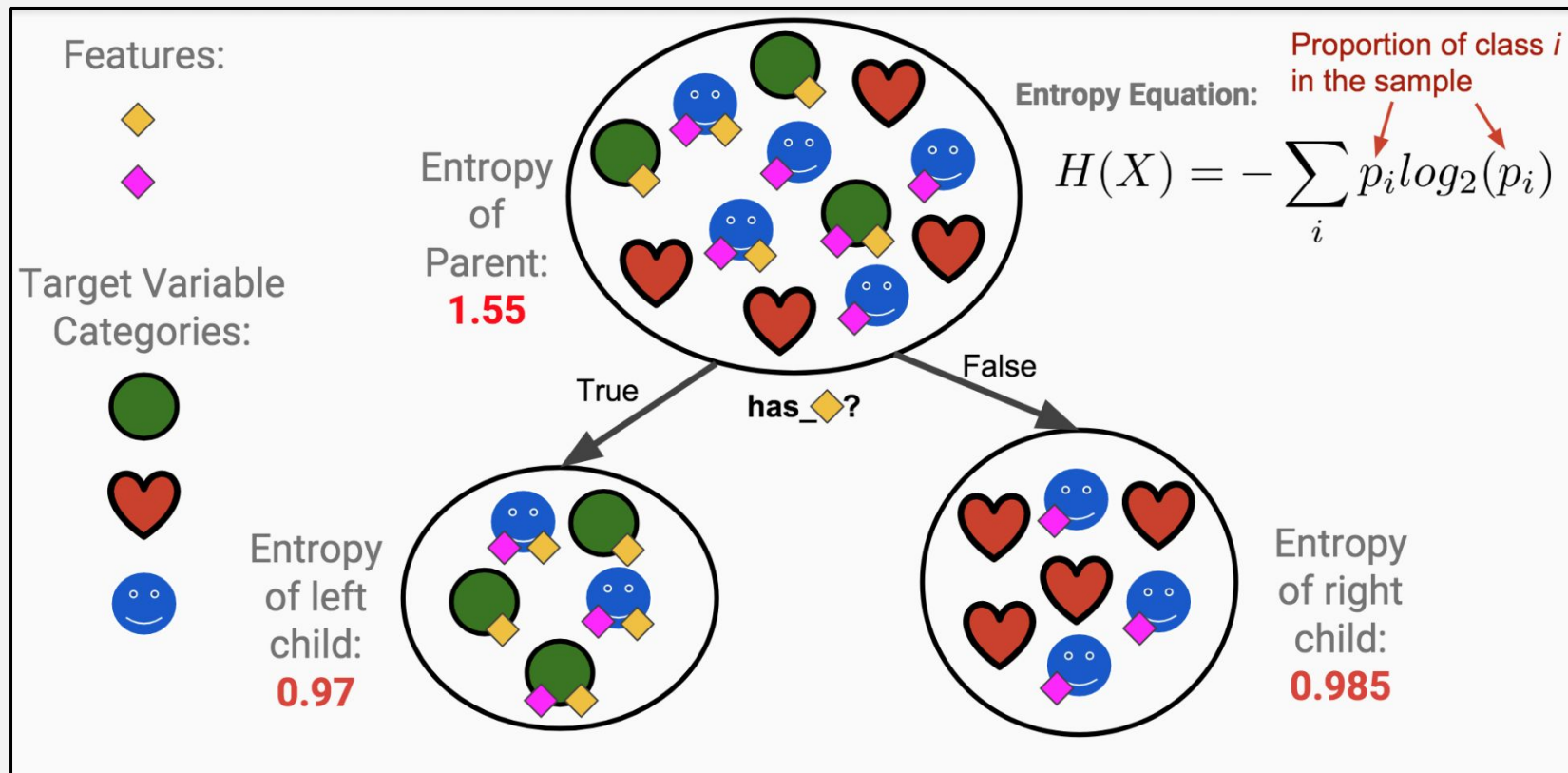
$$\begin{aligned} H &= -0.25 \log_2(0.25) + \\ &\quad -0.33 \log_2(0.33) + \\ &\quad -0.42 \log_2(0.42) \end{aligned}$$

$$H = 1.55$$

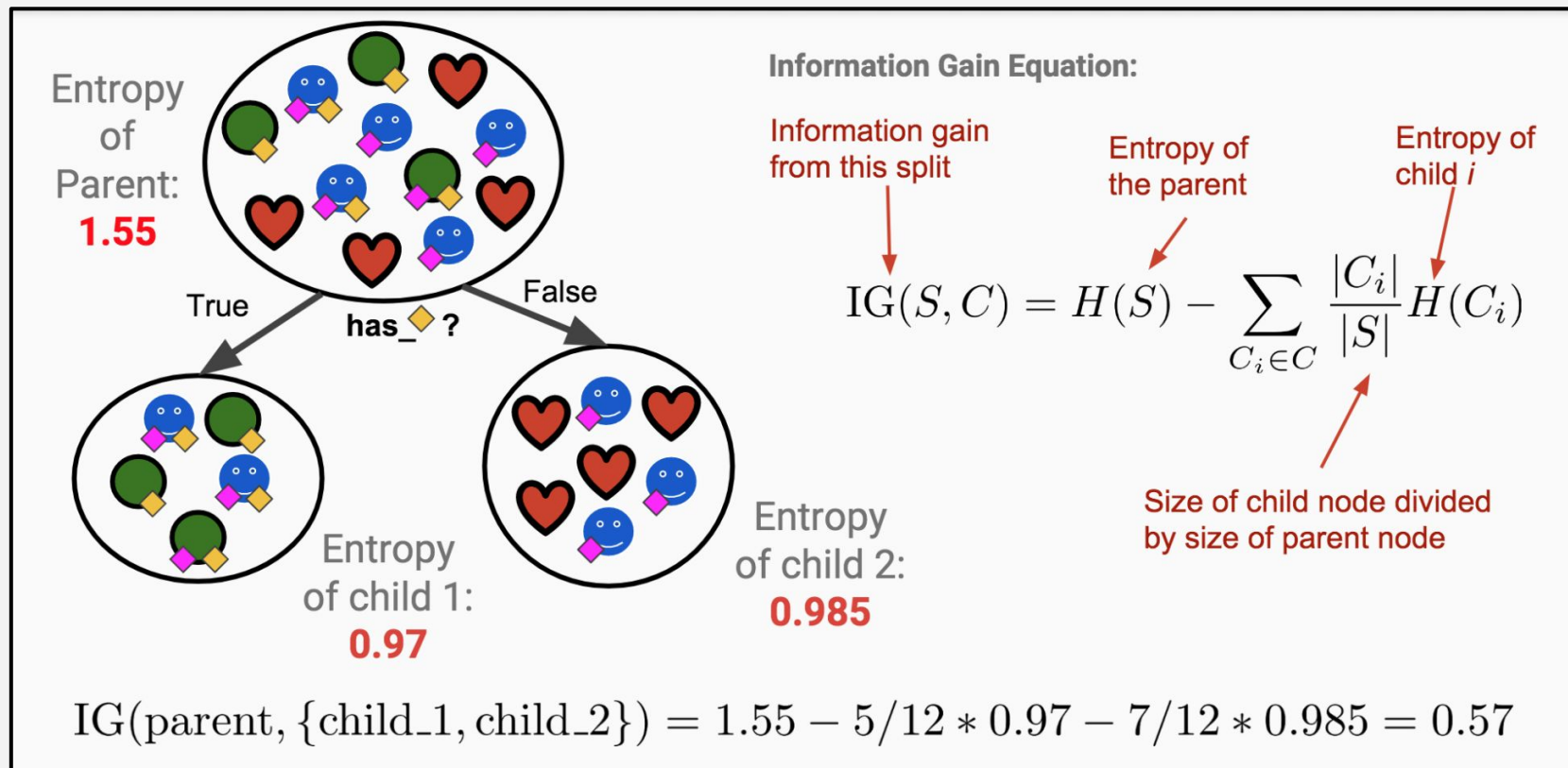
Splitting 3



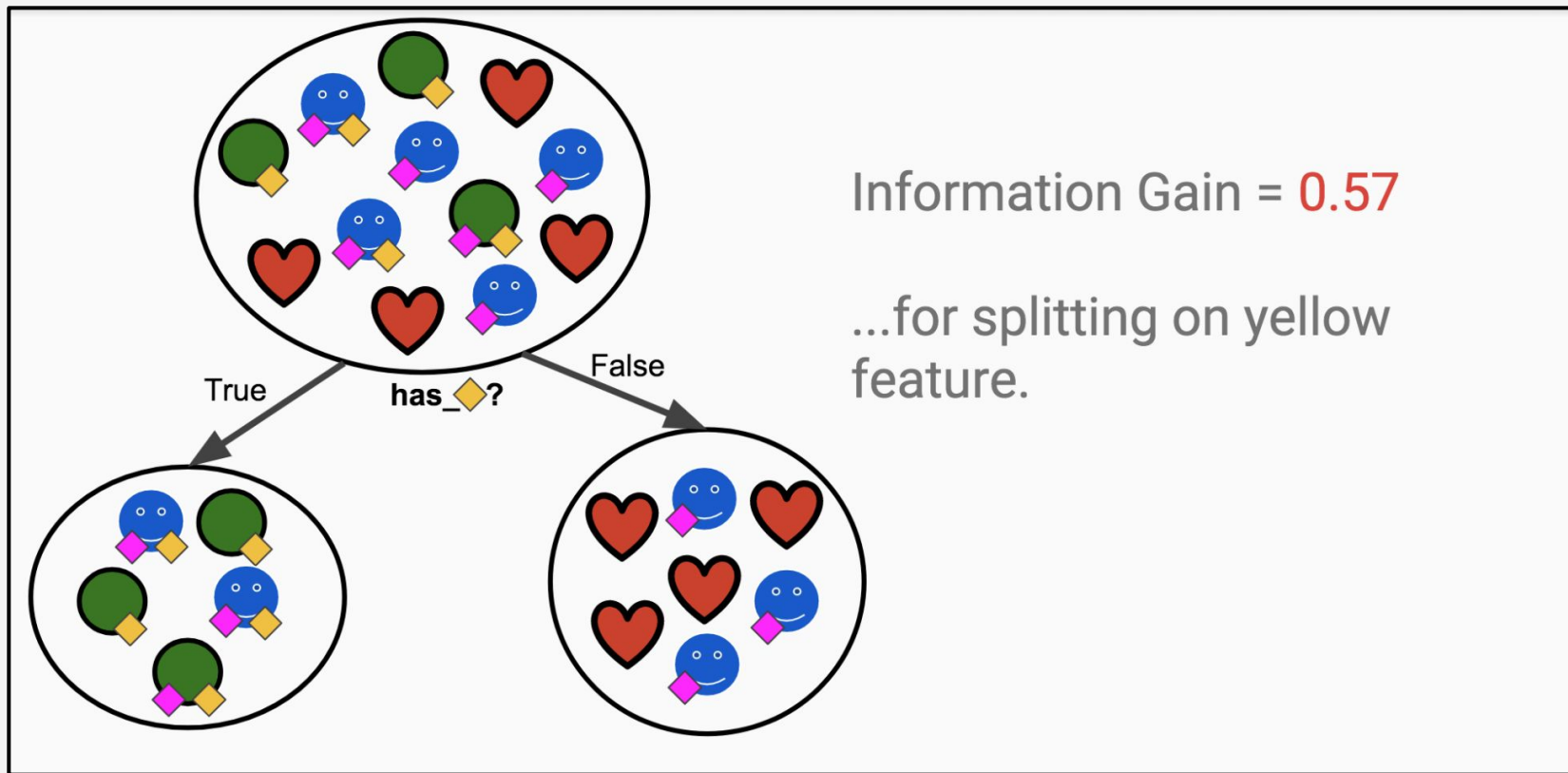
Splitting 4



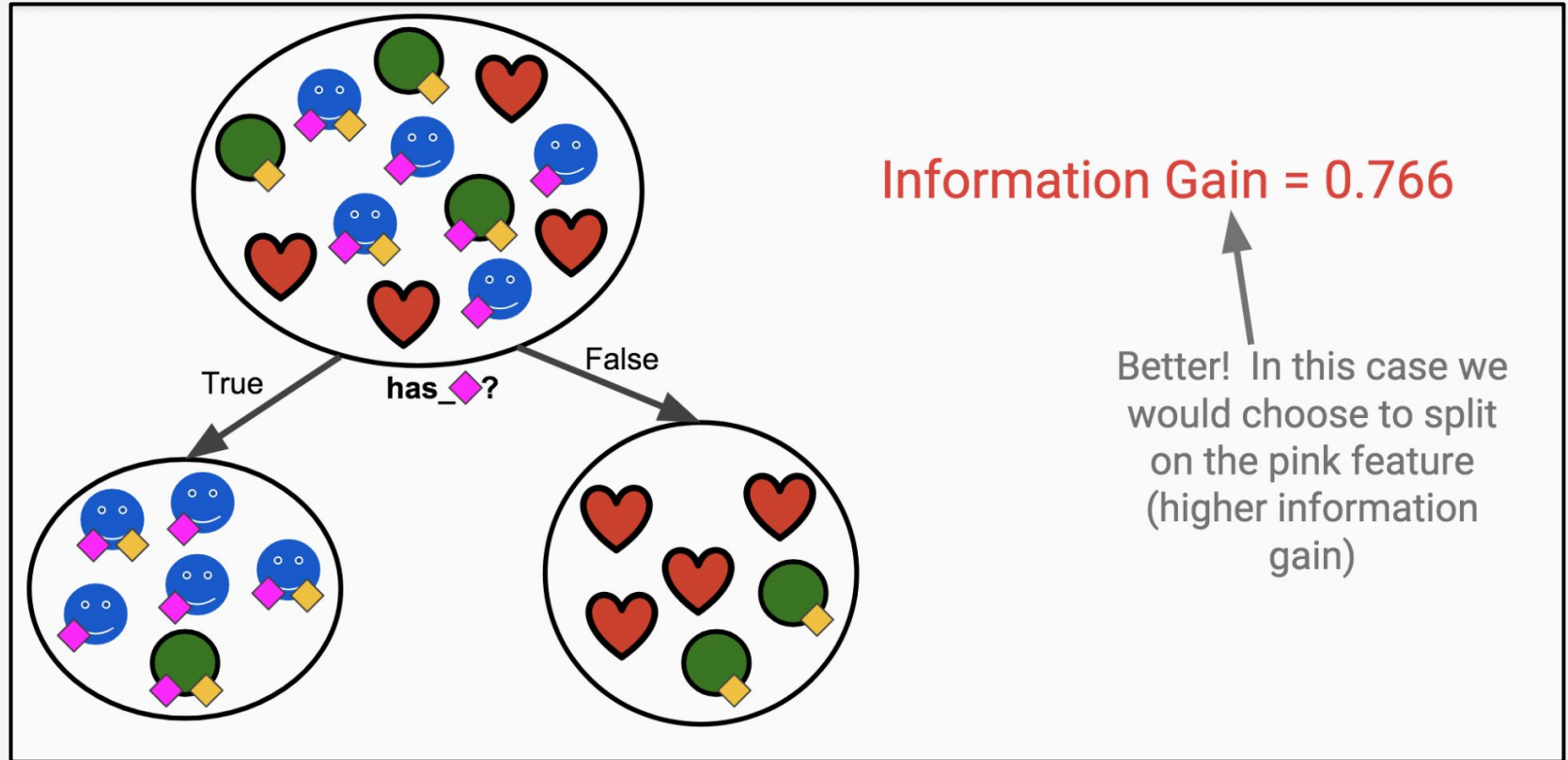
Splitting 5



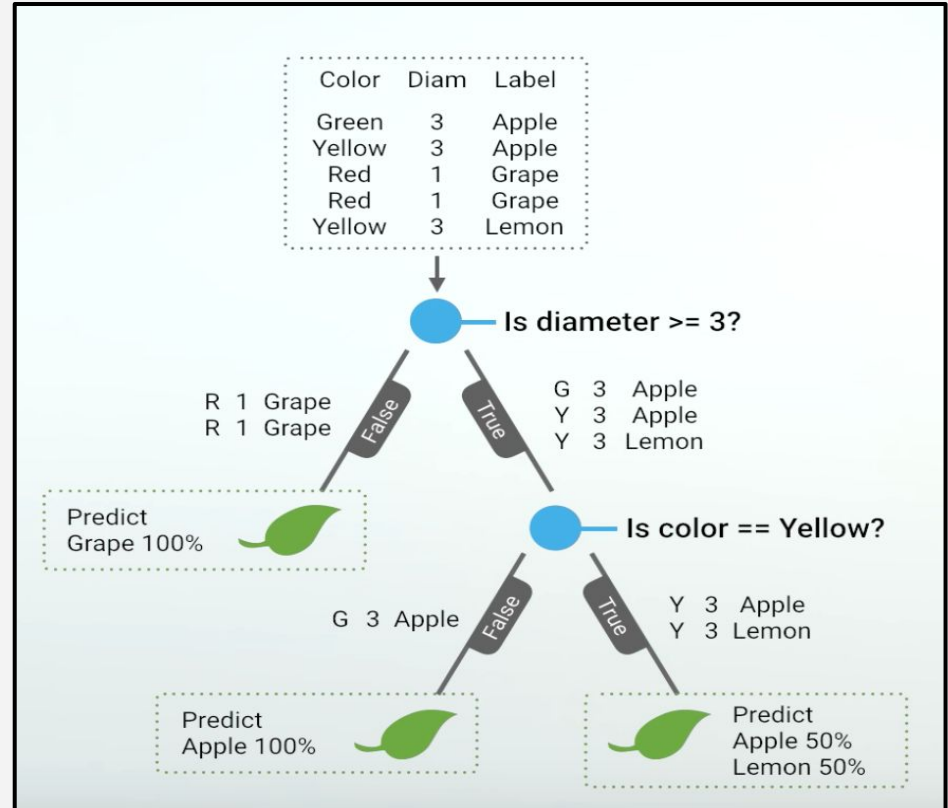
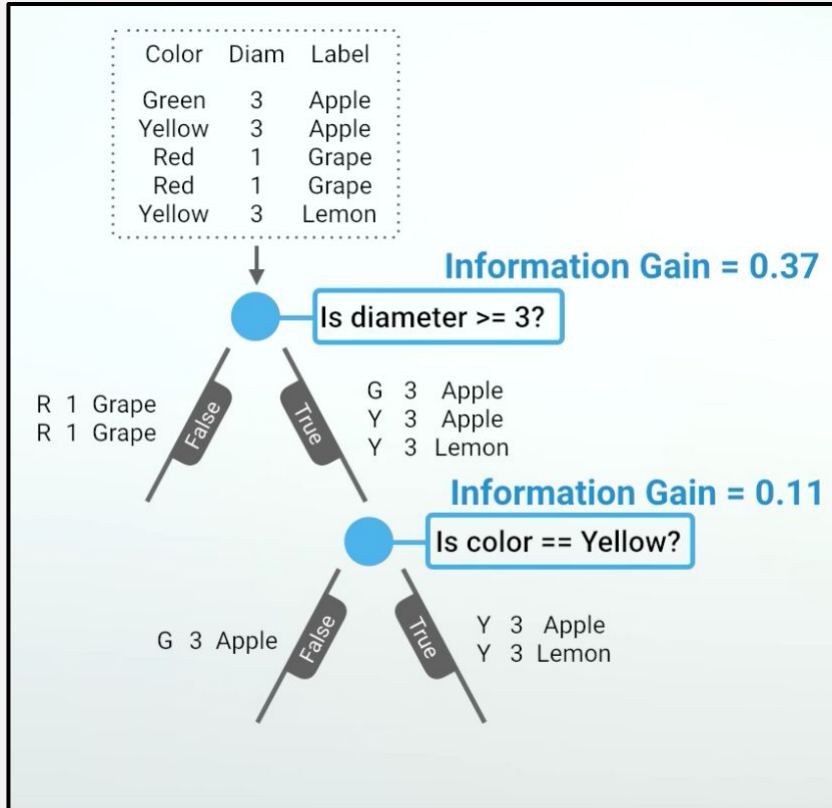
Splitting 6



Splitting 7



Splitting based on information gain



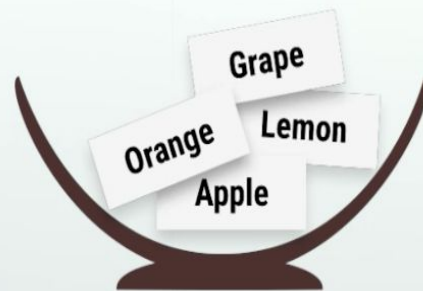
Gini Impurity

Impurity = 0



Impurity = 0.8

$$1 - \frac{1}{5} = 0.8$$



Tree Pruning

- Making the algorithm stop early
 - Leaf size: stop when the number of data points for a leaf gets below a threshold
 - Depth: stop when the depth of the tree (distance from root to leaf) reaches a threshold
 - Mostly the same: stop when some percent of the data points are the same (rather than all the same)
 - Error threshold: stop when the error reduction (information gain) is not improved significantly

sklearn

`sklearn.tree`.DecisionTreeClassifier

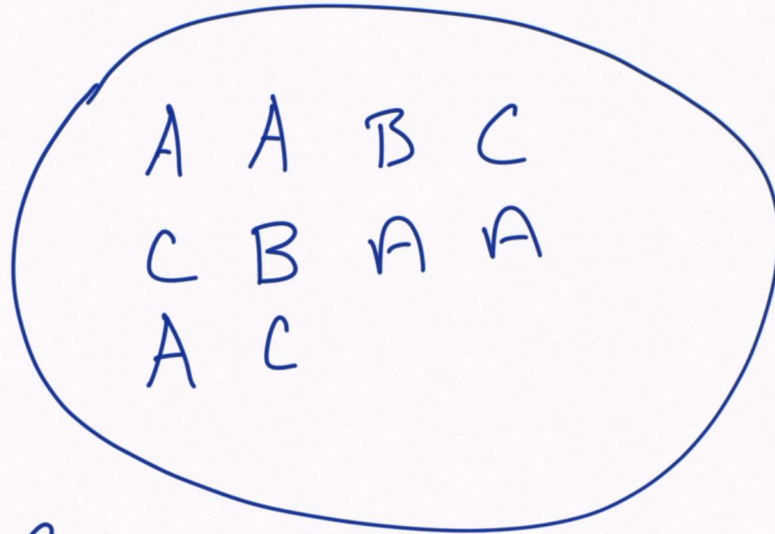
```
class sklearn.tree. DecisionTreeClassifier (criterion='gini', splitter='best', max_depth=None,  
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None,  
max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False)
```

[\[source\]](#)

Hands On



Calculate Entropy & Gini

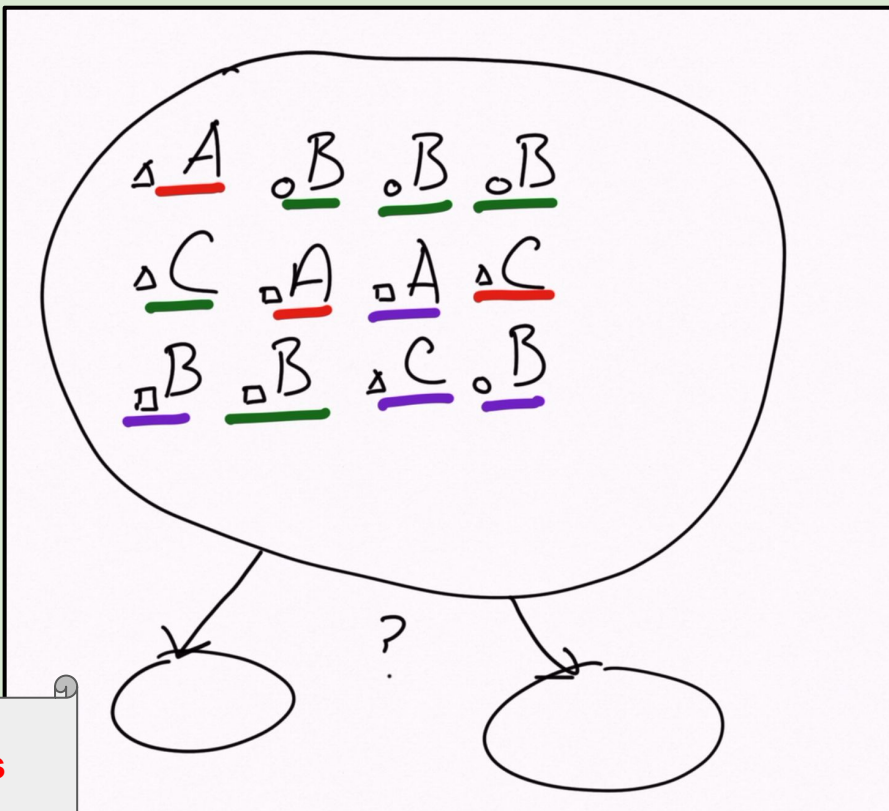


1. Entropy?
2. Gini?

Show results on mini-whiteboards

Information Gain - with classes A, B & C

- Feature 1
 - Shape
 - Circle
 - Square
 - Triangle
- Feature 2
 - Color
 - Red
 - Green
 - Purple
- Choose a Feature/Value to split
- Calculate the Information gain of that split
 - Calculate using Entropy
 - Calculate using Gini



Show results on mini-whiteboards

Use sklearn DecisionTreeClassifier

Things to try

- Fit
- Feature Importance
- Score
- Predict

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
# conda install python-graphviz
```

```
df = pd.read_csv('golf.csv')
y = df.Result
X = df.drop(['Result'], axis=1)
X = pd.get_dummies(X)
```

Let's code!

- Create main.py
 - Import disorder (see below)
 - Use this file to test out your functions
- Create disorder.py
 - Write entropy function
 - Write gini function
 - Write information_gain function

```
from disorder import gini, entropy, information_gain

p = 'aaaaabbbbccc'
c1 = 'aacc'
c2 = 'aaabbbb'
z1 = entropy(p)
z2 = gini(p)
z3 = information_gain(p, c1, c2)
```