Foundations of Machine Learning (ECE 5984)

- Decision Trees and Random Forest -

Eunbyung Park

Assistant Professor

School of Electronic and Electrical Engineering

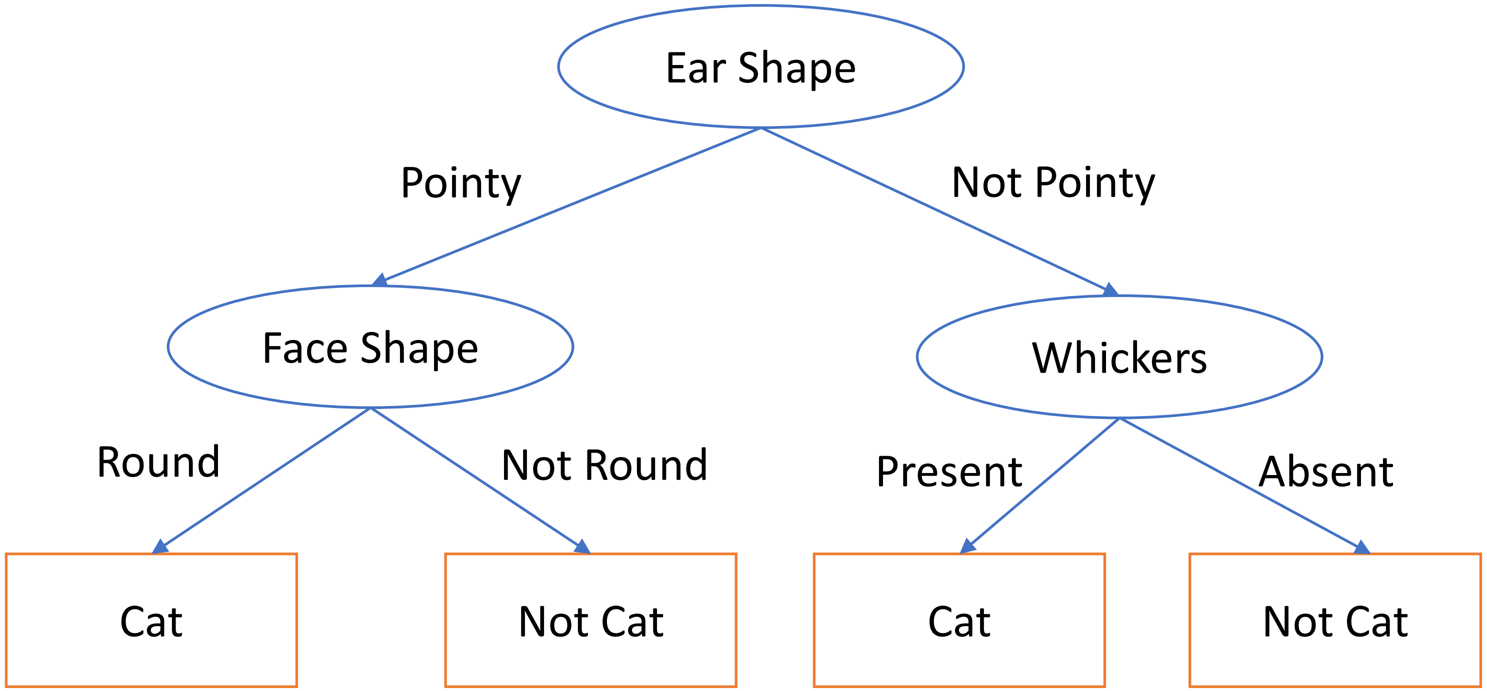
Eunbyung Park (silverbottlep.github.io)

Machine Learning | Coursera

Cat Classification Example

Ear shape (x_1)	Face shape (x_2)	Whiskers (x_3)	Cat
Pointy	Round	Present	1
Floppy	Not round	Present	1
Floppy	Round	Absent	0
Pointy	Not round	Present	0
Pointy	Round	Present	1
Pointy	Round	Absent	1
Floppy	Not round	Absent	0
Pointy	Round	Absent	1
Floppy	Round	Absent	0
Floppy	Round	Absent	0

An Example of Decision Tree



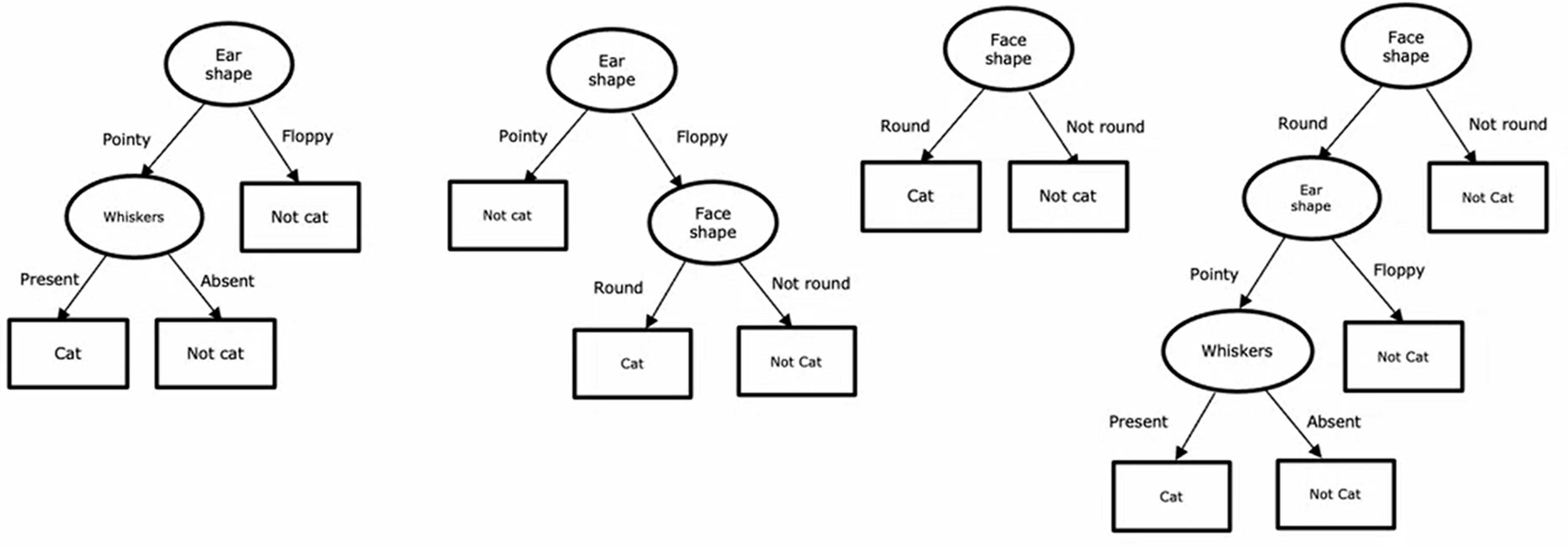
New test example

Ear shape: Pointy

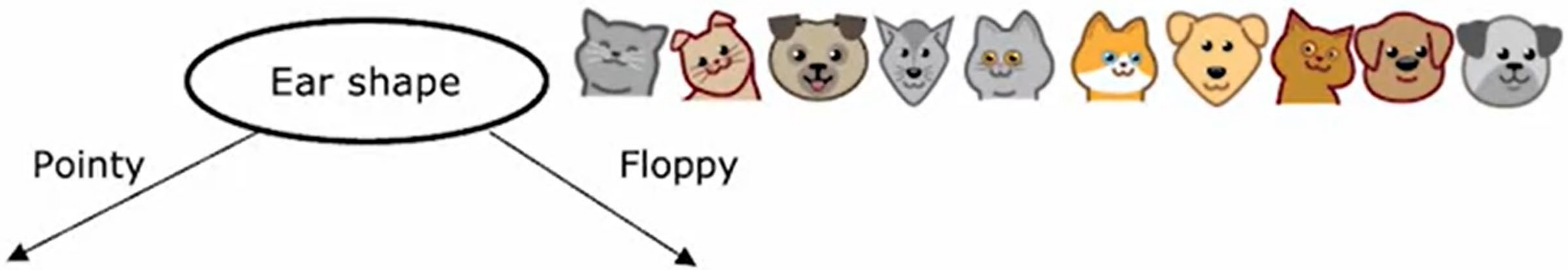
Face shape: Round

Whiskers: Present

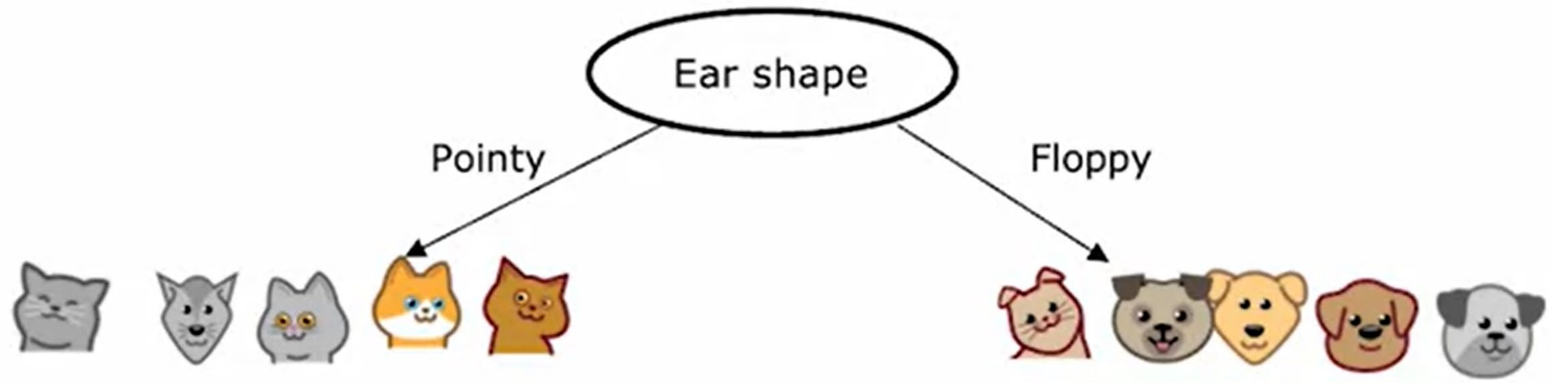
Various Trees



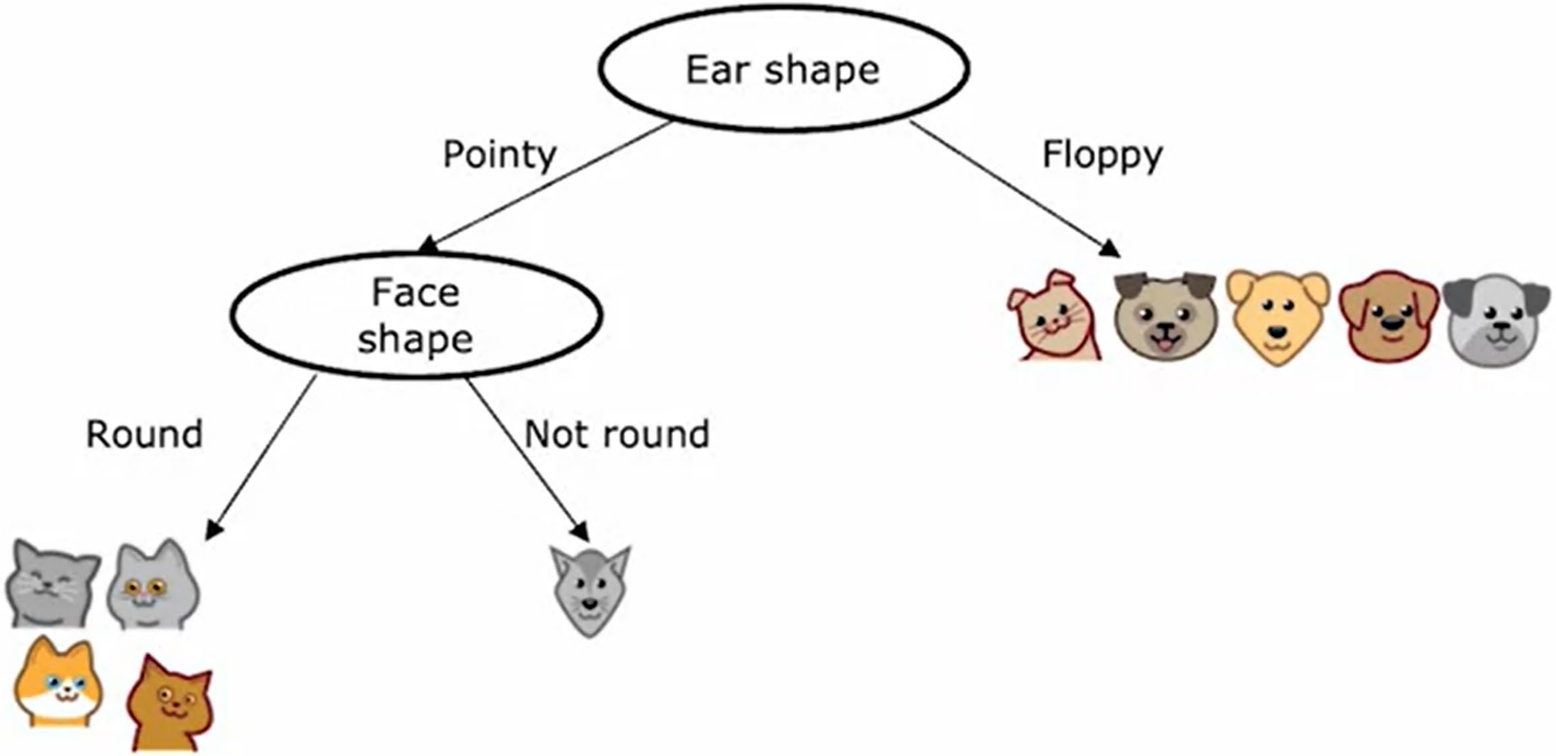
Decision Tree Learning



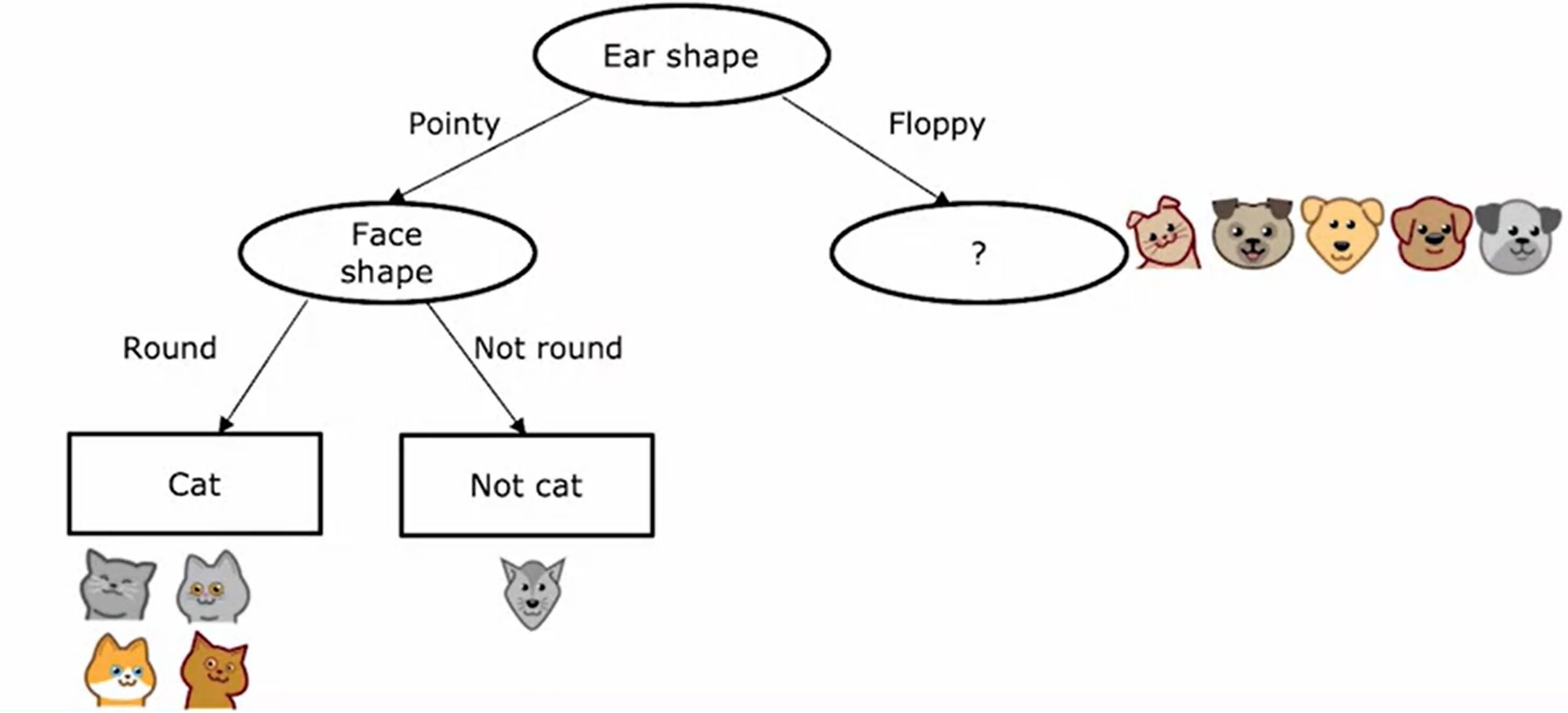
Decision Tree Learning



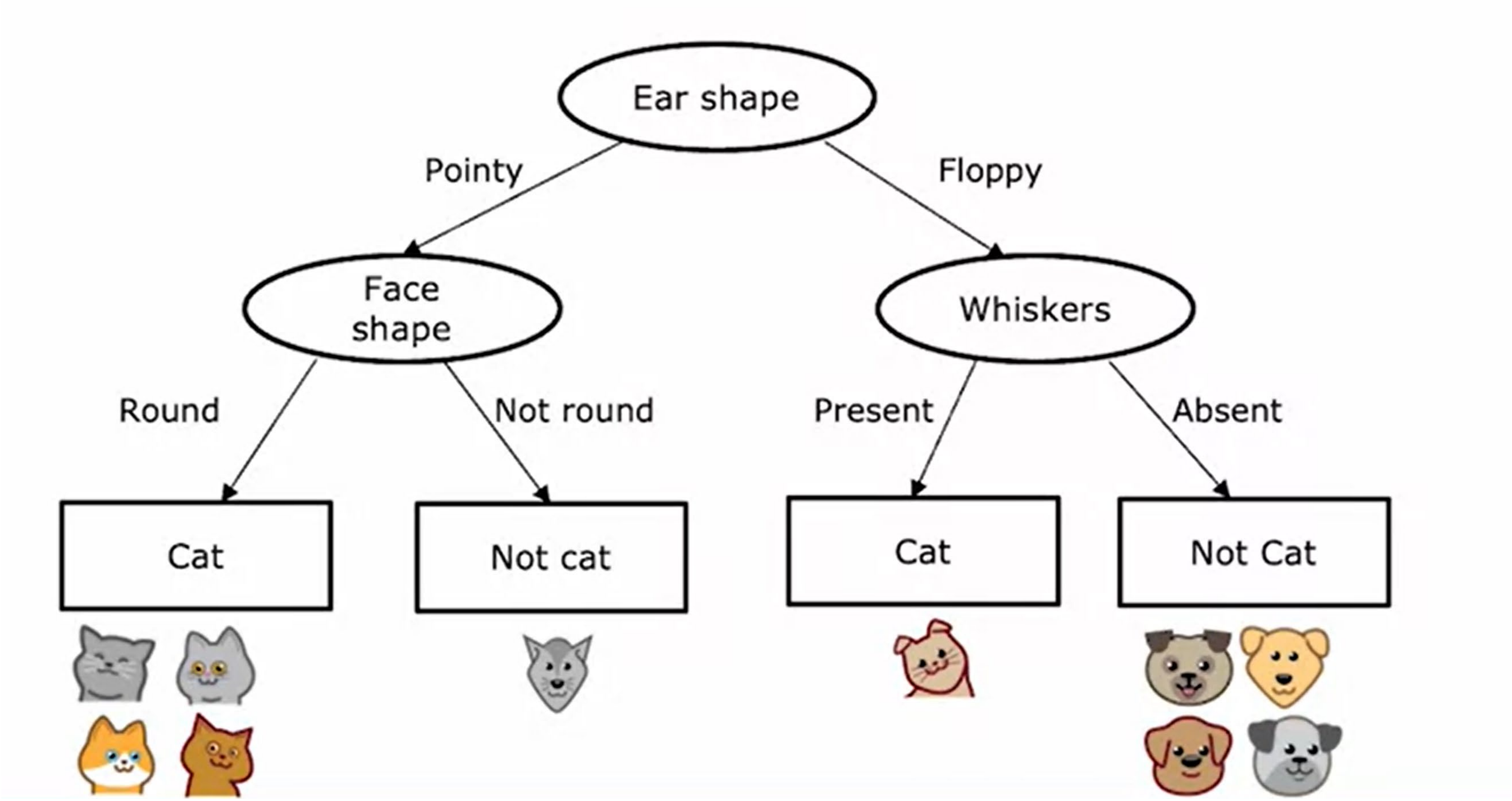
Decision Tree Learning



Decision Tree Learning

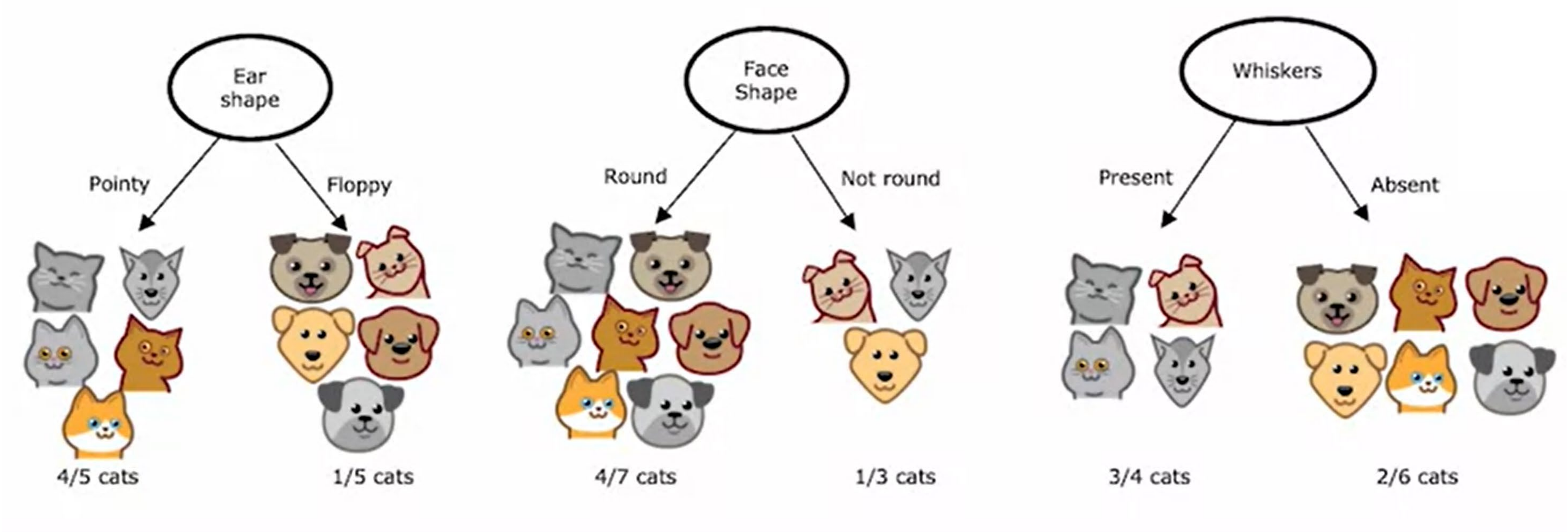


Decision Tree Learning



Decision Tree Learning

- Decision 1: How to choose what feature to split on at each nod?
- Maximize Purity (or minimize impurity)



Decision Tree Learning

- Decision 2: When do you stop splitting?
- When a node is 100% one class
- A maximum depth thresholds
- When improvements in purity score are below a threshold
- When number of examples in a node is below a threshold
- …

Bias and Variance

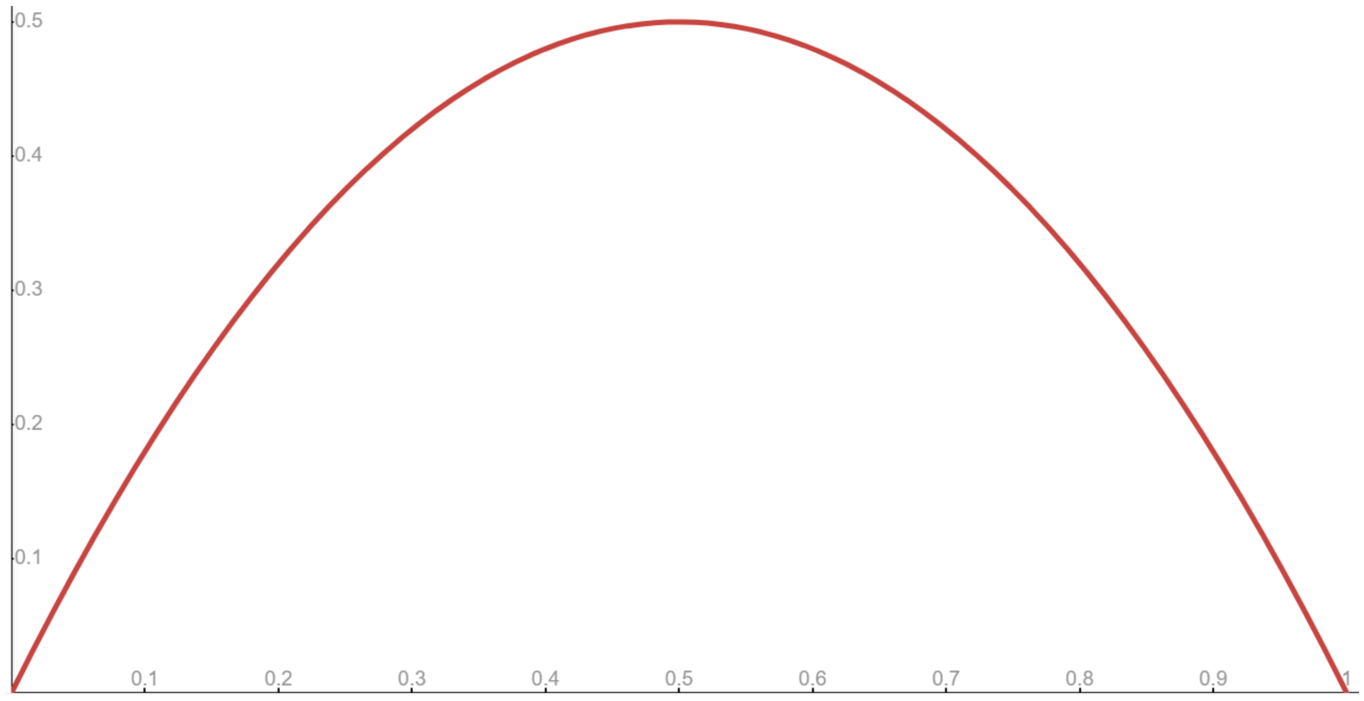
- Large Tree – High Variance (Overfitting)
- Small Tree – High Bias (Underfitting)
- We want to find the smallest tree with high accuracy
- NP Hard!

Measuring Purity – Gini Impurity

$$D = \{(x_1, y_1), \dots, (x_N, y_N)\} \qquad y_i \in \{1, \dots, K\}$$

$$p_k = \frac{|D_k|}{|D|}$$

$$G(D) = \sum_{k=1}^{K} p_k(1 - p_k)$$

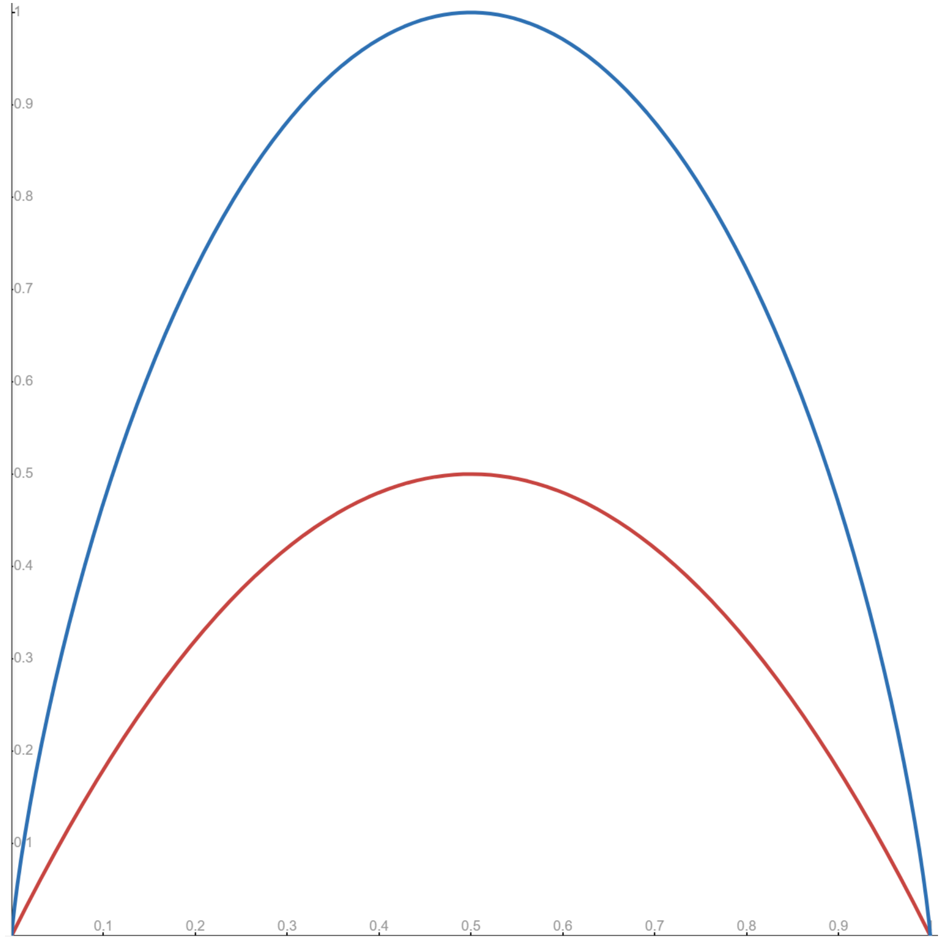


Measuring Purity – Entropy

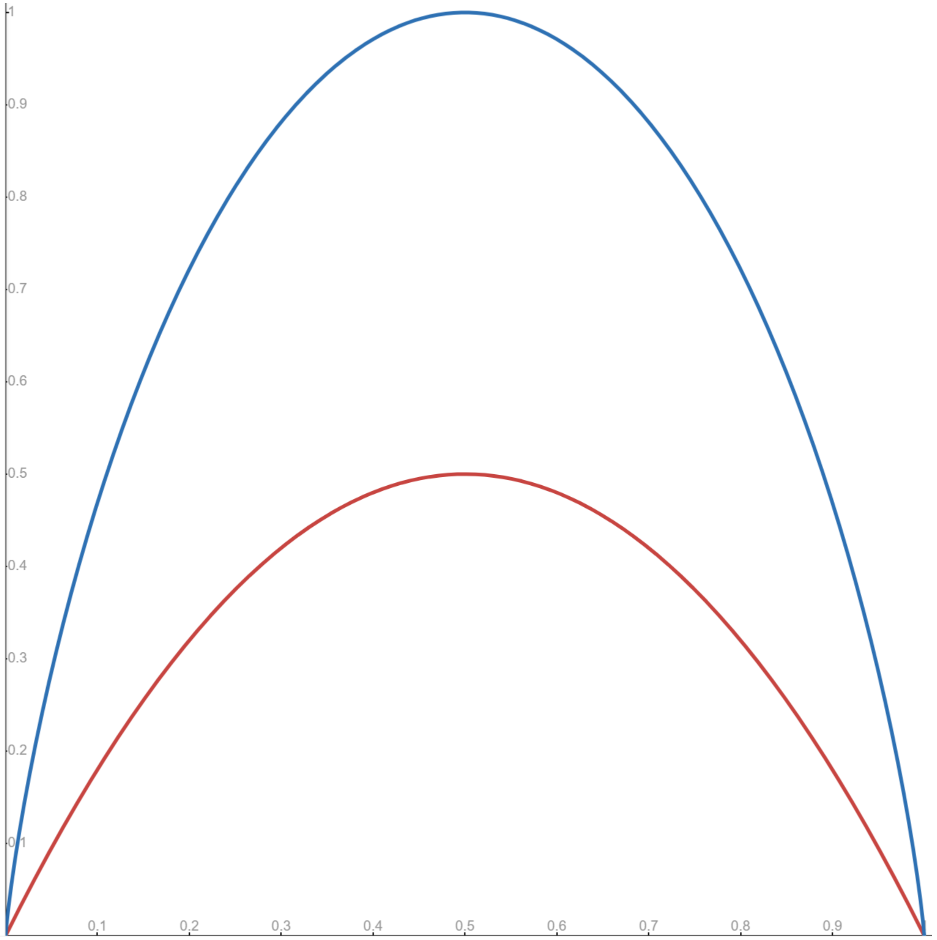
$$D = \{(x_1, y_1), \dots, (x_N, y_N)\} \qquad y_i \in \{1, \dots, K\}$$

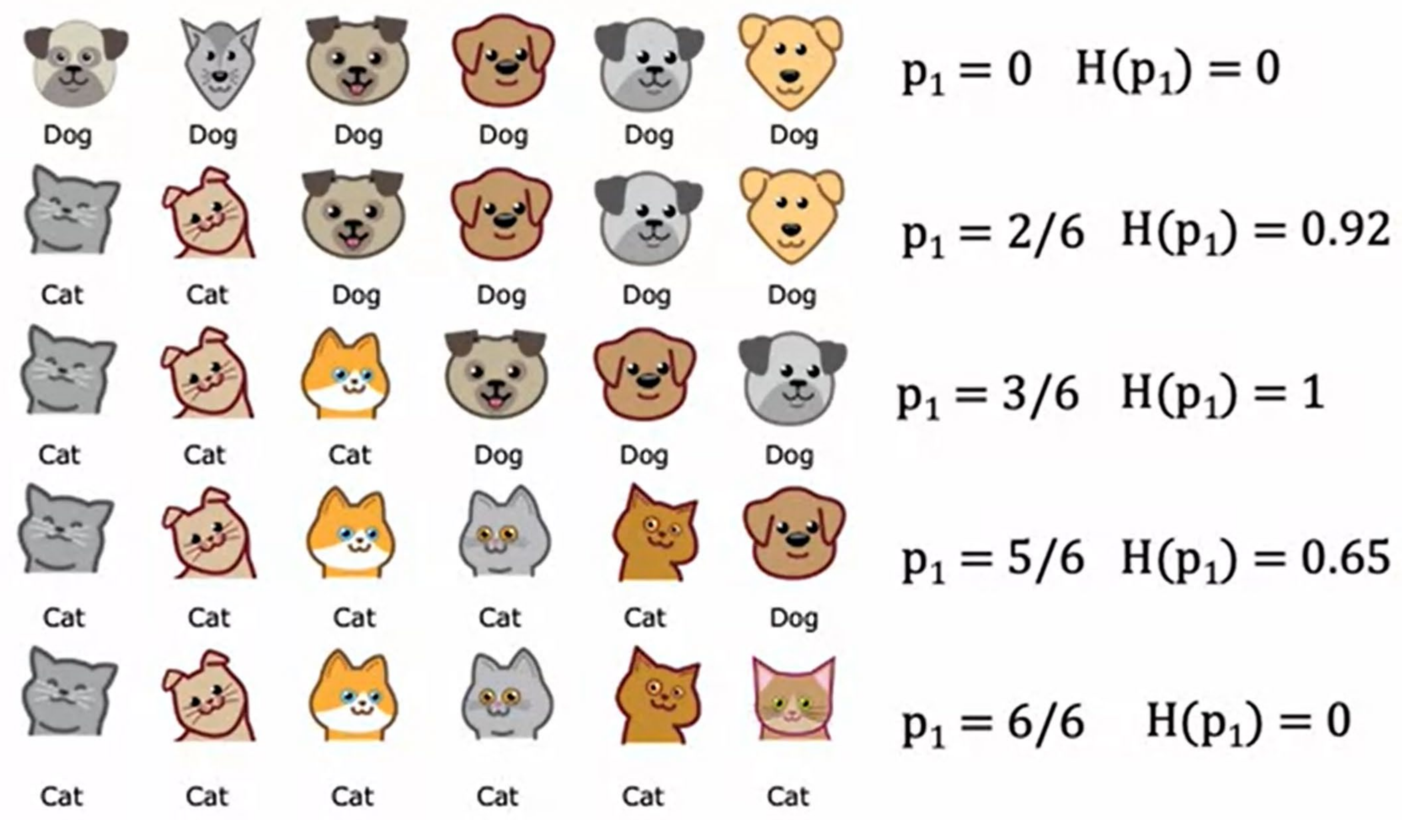
$$p_k = \frac{|D_k|}{|D|}$$

$$H(D) = -\sum_{k=1}^{K} p_k \log p_k$$



Measuring Purity – Entropy

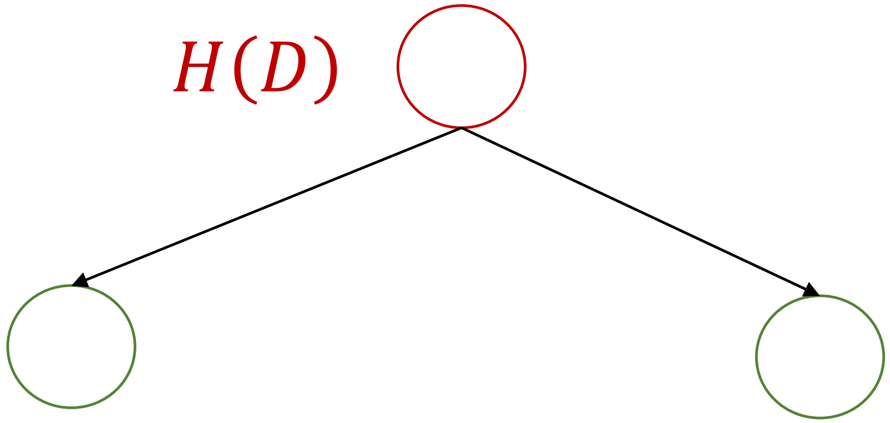




Measuring Purity over Tree

$H(D)$

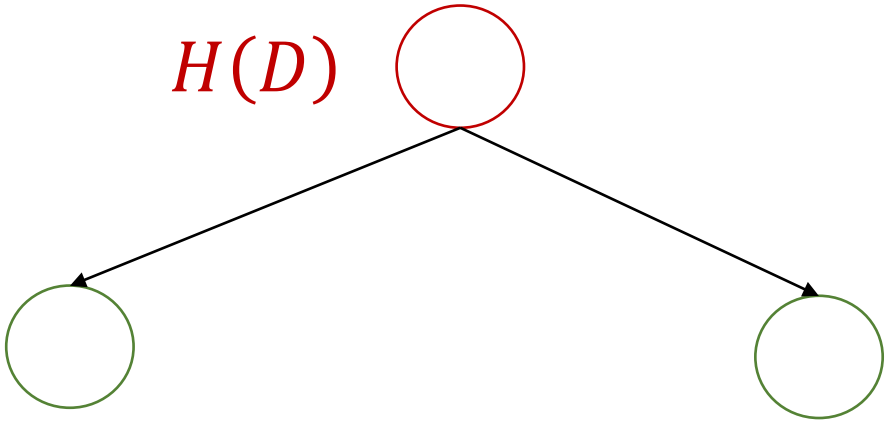
$$\frac{|D^L|}{|D|}H(D^L) + \frac{|D^R|}{|D|}H(D^R)$$



Information Gain

- We want to find the split that maximize the information gain

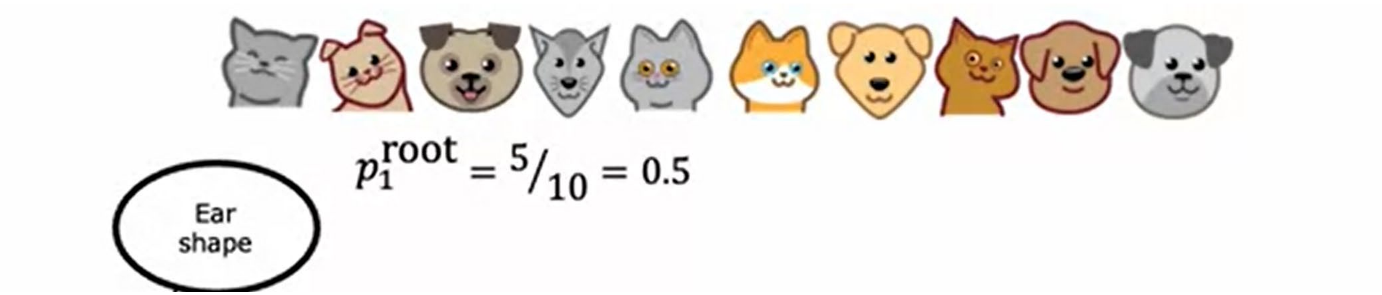
$$H(D)$$



$$\frac{|D^L|}{|D|}H(D^L) + \frac{|D^R|}{|D|}H(D^R)$$

$$\text{Information Gain} = H(D) - \left(\frac{|D^L|}{|D|}H(D^L) + \frac{|D^R|}{|D|}H(D^R)\right)$$

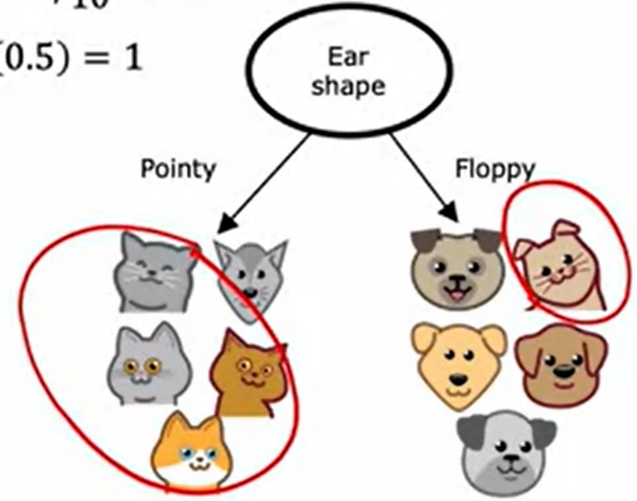
Information Gain Example



Information Gain Example

$p_1 = {}^{5}/_{10} = 0.5$

$H(0.5) = 1$



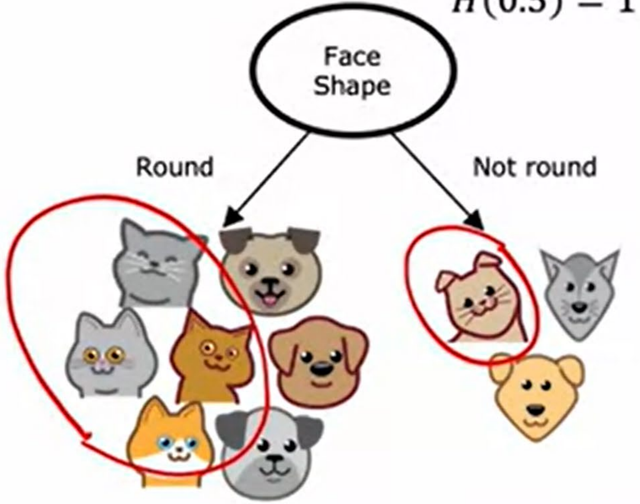
$p_1 = {}^{4}/_{5} = 0.8$ $\quad$ $p_1 = {}^{1}/_{5} = 0.2$

$H(0.8) = 0.72$ $\quad$ $H(0.2) = 0.72$

$$H(0.5) - \left(\frac{5}{10}H(0.8) + \frac{5}{10}H(0.2)\right)$$

$= 0.28$

$H(0.5) = 1$



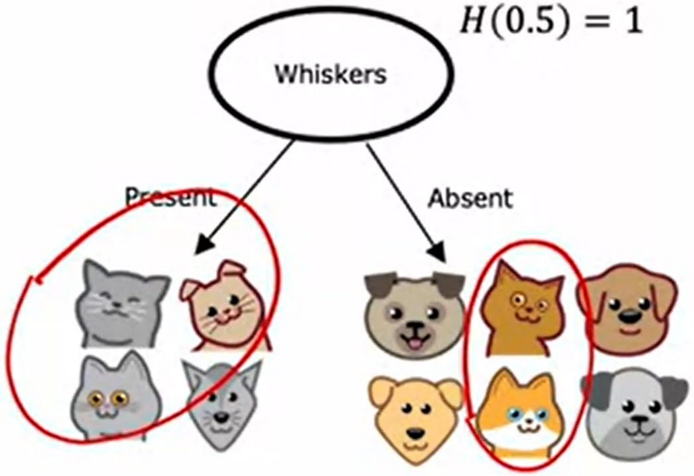
$p_1 = {}^{4}/_{7} = 0.57$ $\quad$ $p_1 = {}^{1}/_{3} = 0.33$

$H(0.57) = 0.99$ $\quad$ $H(0.33) = 0.92$

$$H(0.5) - \left(\frac{7}{10}H(0.57) + \frac{3}{10}H(0.33)\right)$$

$= 0.03$

$H(0.5) = 1$



$p_1 = {}^{3}/_{4} = 0.75$ $\quad$ $p_1 = {}^{2}/_{6} = 0.33$

$H(0.75) = 0.81$ $\quad$ $H(0.33) = 0.92$

$$H(0.5) - \left(\frac{4}{10}H(0.75) + \frac{6}{10}H(0.33)\right)$$

$= 0.12$

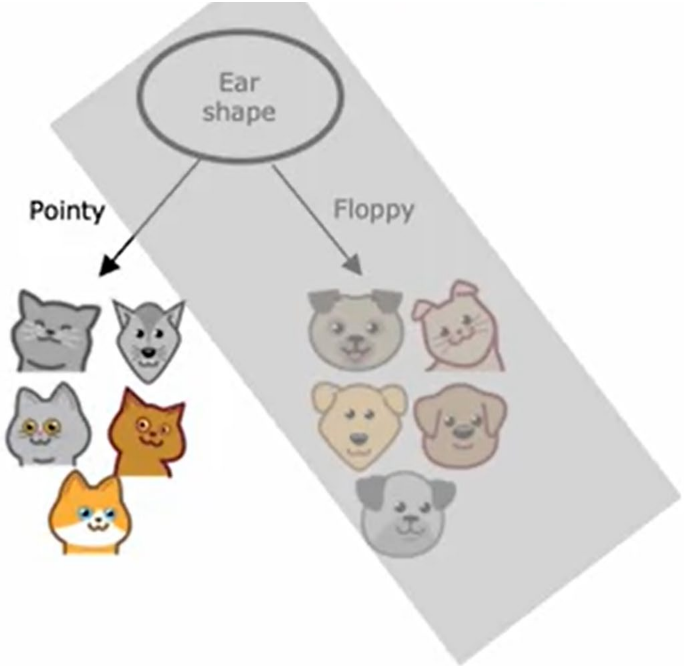
Information gain

Decision Tree Learning

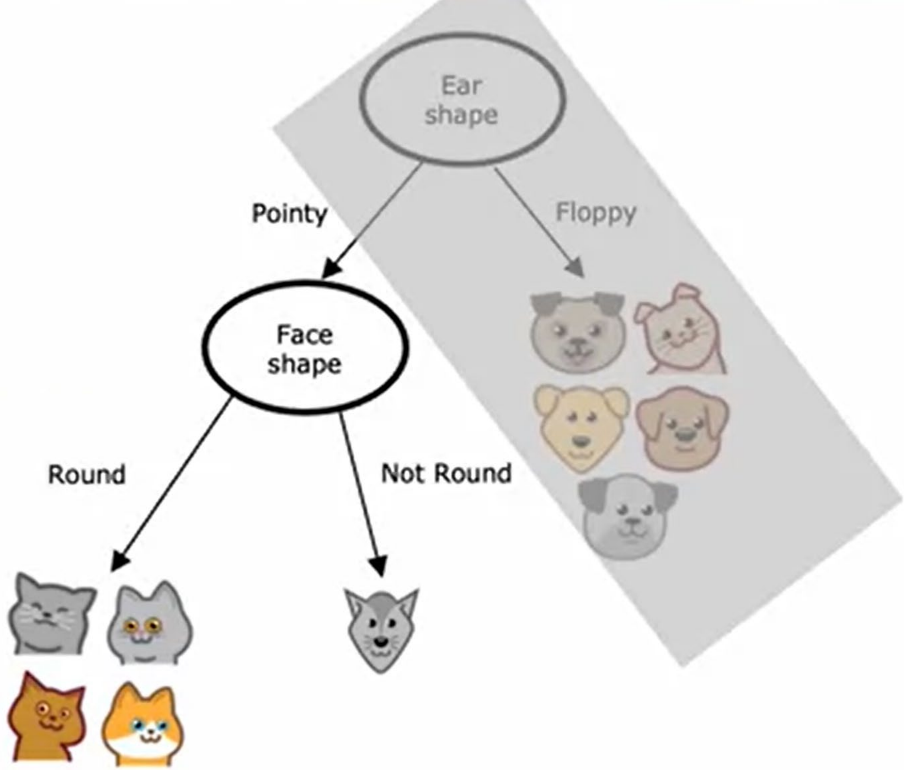
- Start with all examples at the root node
- Calculate information gain for all possible features, and pick the one with the highest information gain
- Split dataset according to selected features, and create left and right branches of the tree
- Keep repeating splitting process until stopping criteria is met:
  - When a node is 100% one class
  - When splitting a node will result in the tree exceeding a maximum depth
  - Information gain from additional splits is less than threshold
  - When number of examples in a node is below a threshold

Recursive Splitting

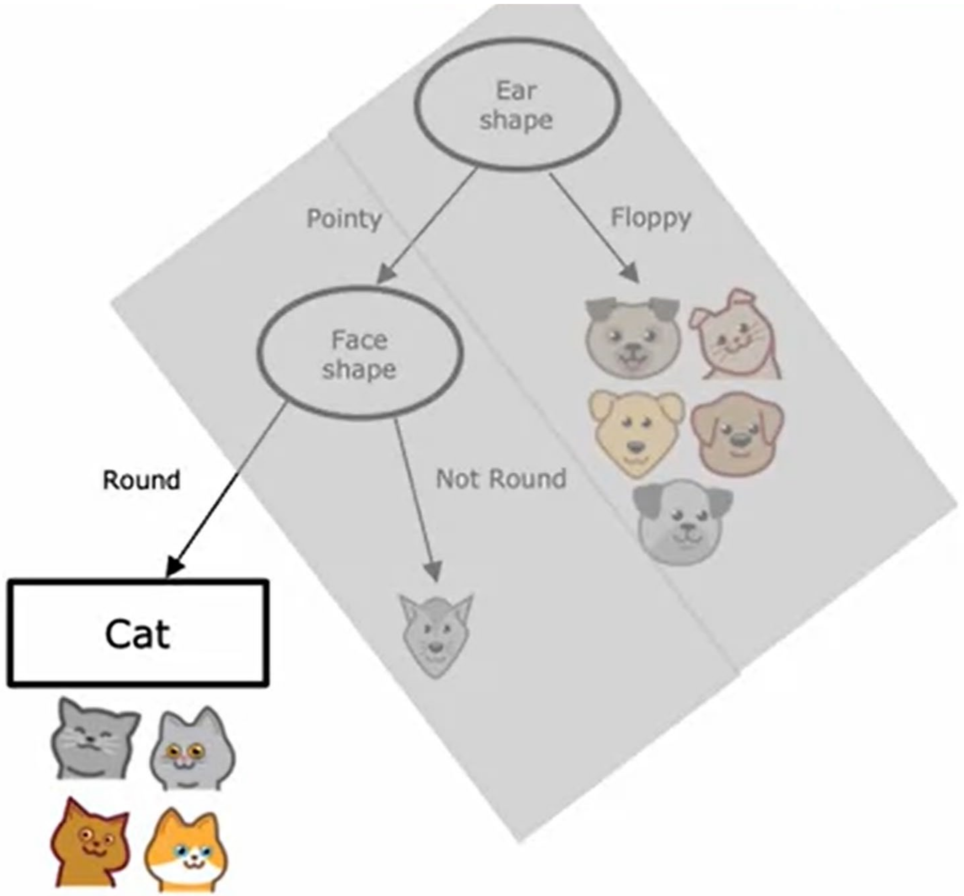
Recursive Splitting



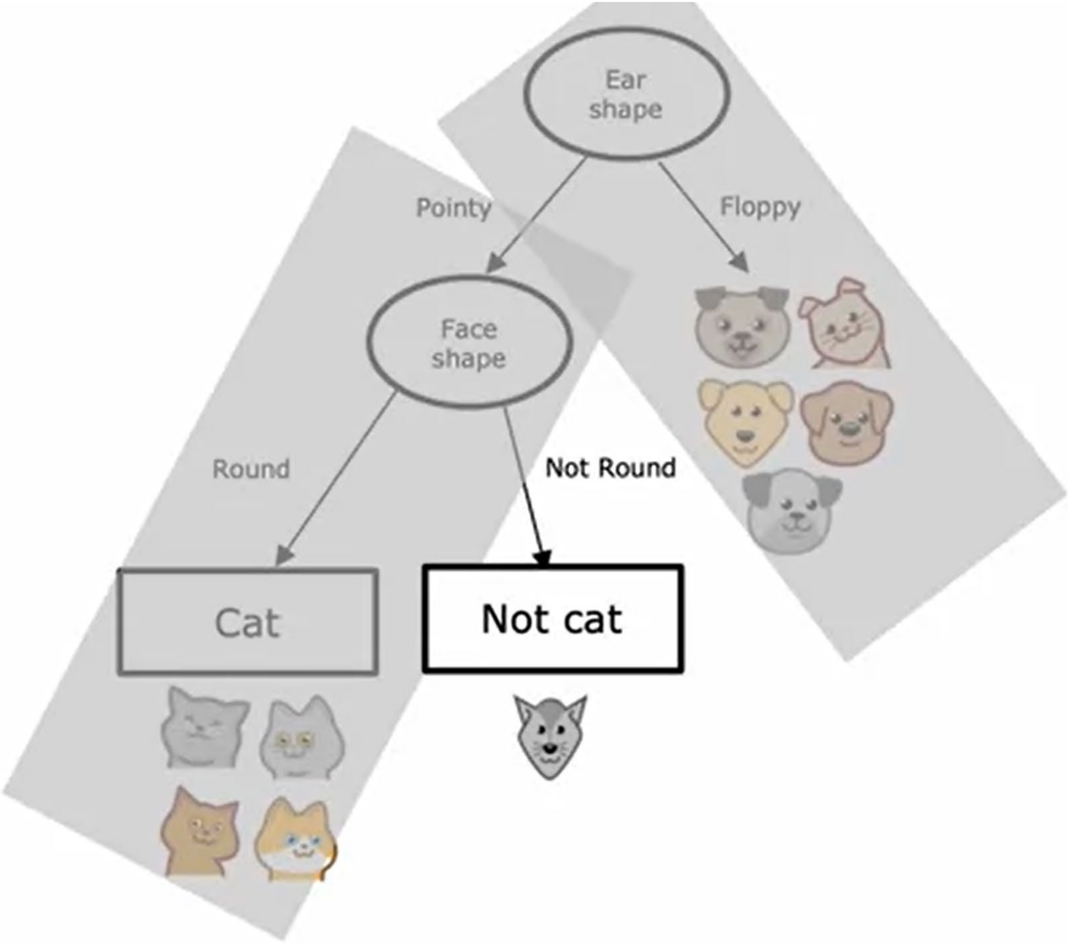
Recursive Splitting



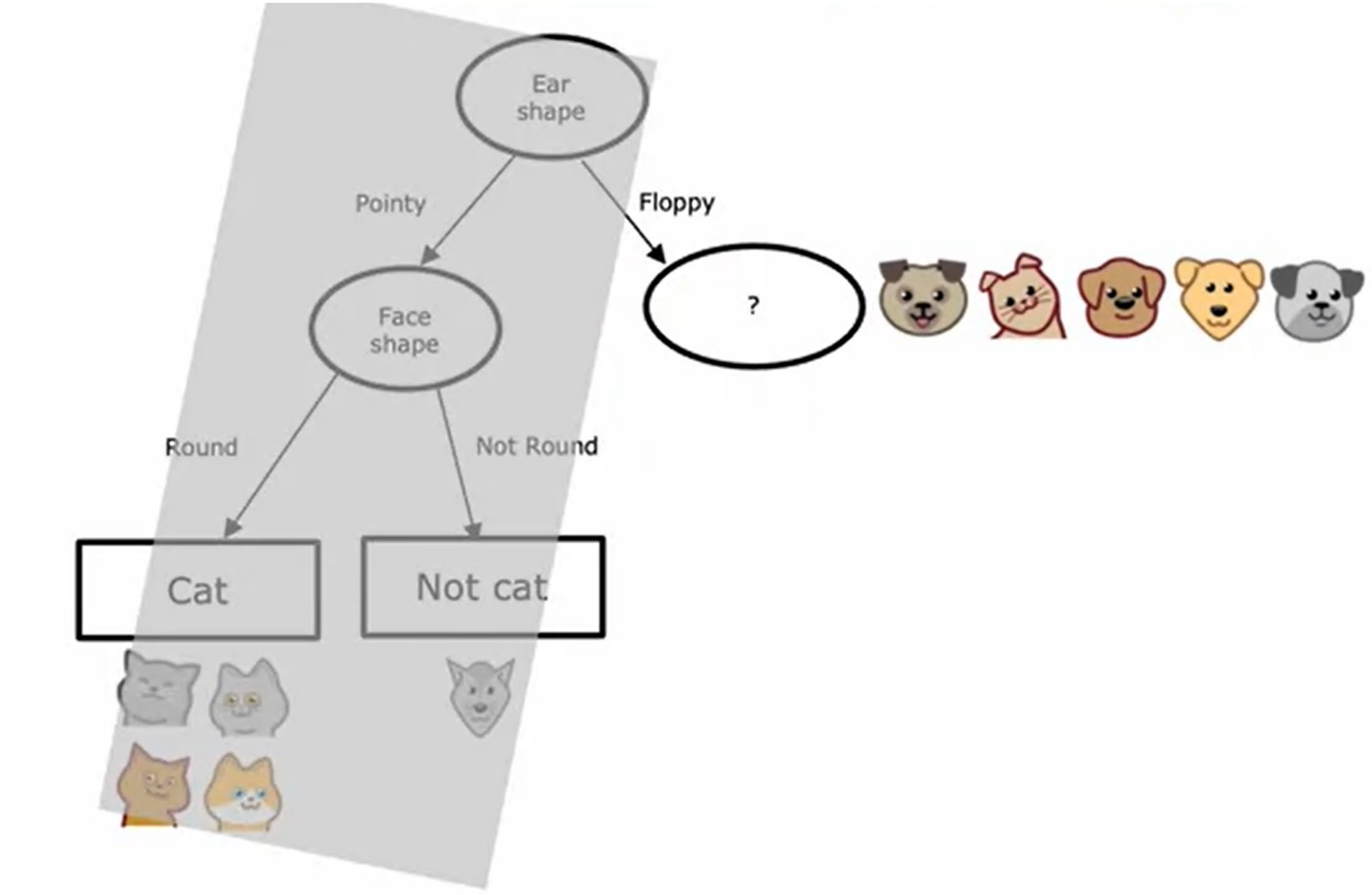
Recursive Splitting



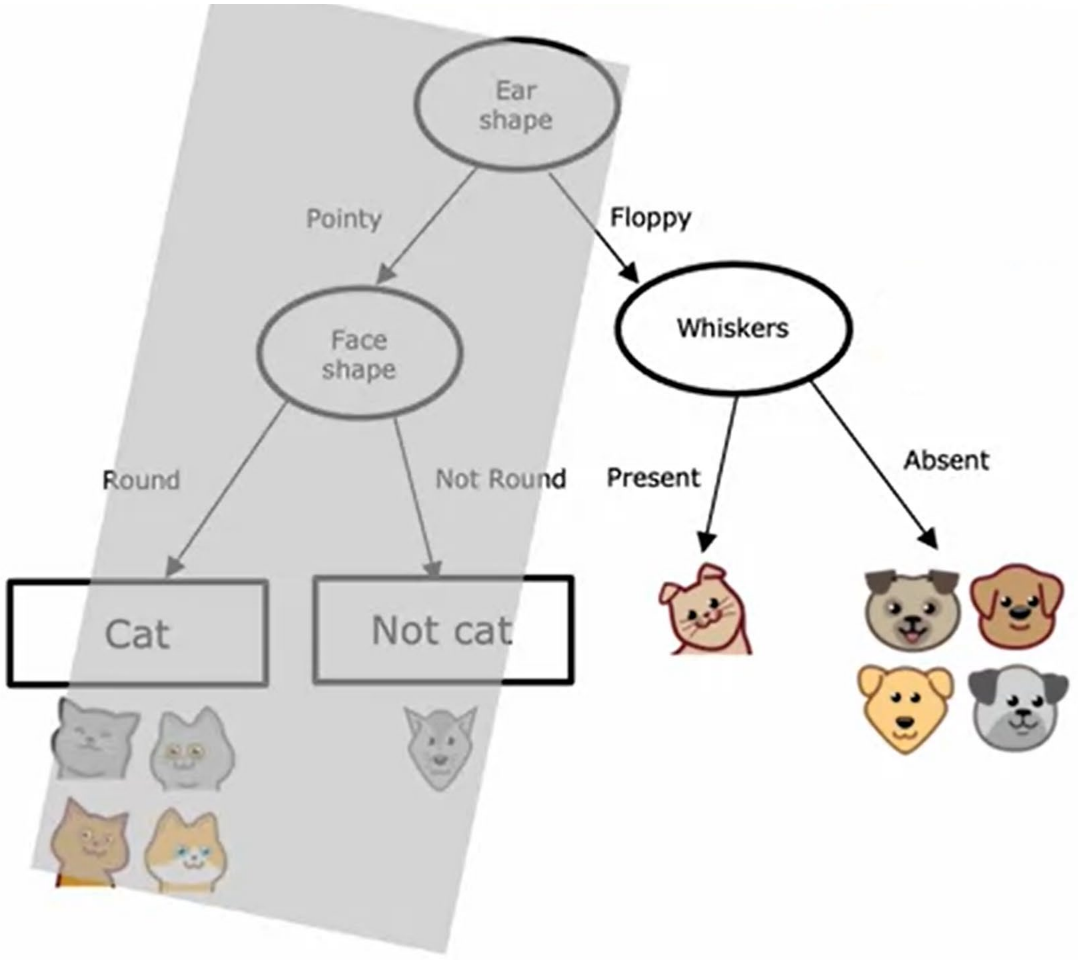
Recursive Splitting



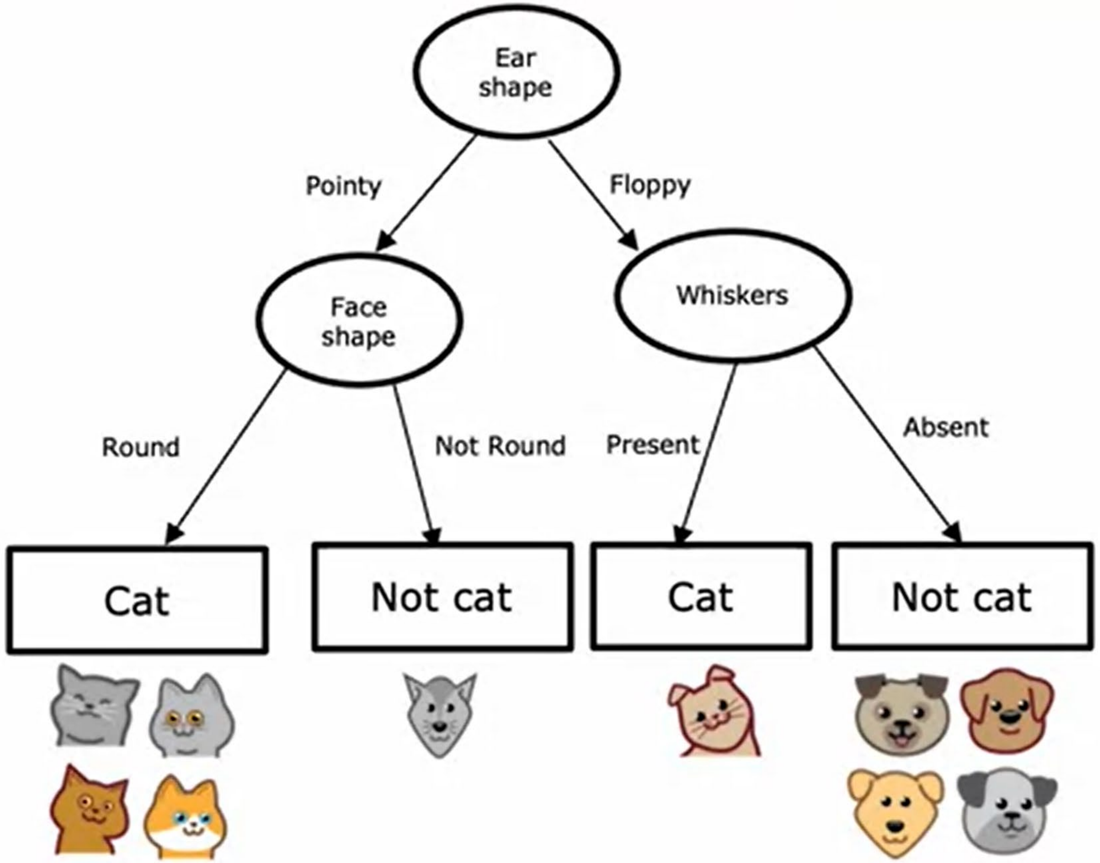
Recursive Splitting



Recursive Splitting



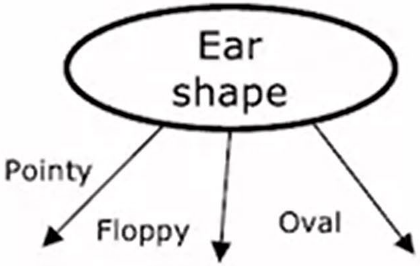
Recursive Splitting



Categorical Features

Ear shape (x_1)	Face shape (x_2)	Whiskers (x_3)	Cat (y)
Pointy	Round	Present	1
Oval	Not round	Present	1
Oval	Round	Absent	0
Pointy	Not round	Present	0
Oval	Round	Present	1
Pointy	Round	Absent	1
Floppy	Not round	Absent	0
Oval	Round	Absent	1
Floppy	Round	Absent	0
Floppy	Round	Absent	0

3 possible values



One Hot Encoding

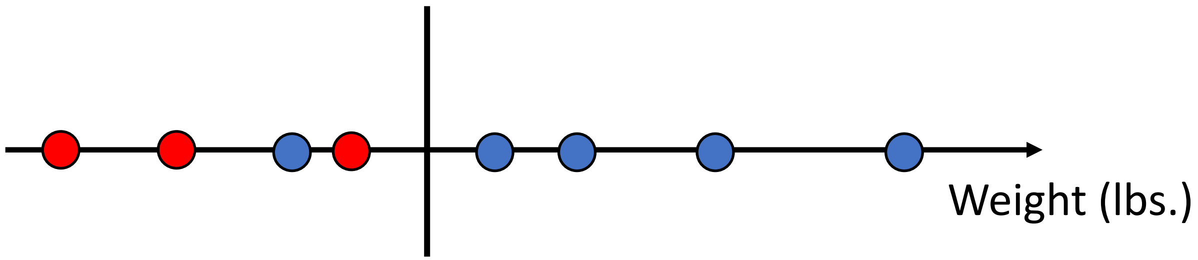
	~~Ear shape~~	Pointy ears	Floppy ears	Oval ears	Face shape	Whiskers	Cat
	~~Pointy~~	1	0	0	Round	Present	1
	~~Oval~~	0	0	1	Not round	Present	1
	~~Oval~~	0	0	1	Round	Absent	0
	~~Pointy~~	1	0	0	Not round	Present	0
	~~Oval~~	0	0	1	Round	Present	1
	~~Pointy~~	1	0	0	Round	Absent	1
	~~Floppy~~	0	1	0	Not round	Absent	0
	~~Oval~~	0	0	1	Round	Absent	1
	~~Floppy~~	0	1	0	Round	Absent	0
	~~Floppy~~	0	1	0	Round	Absent	0

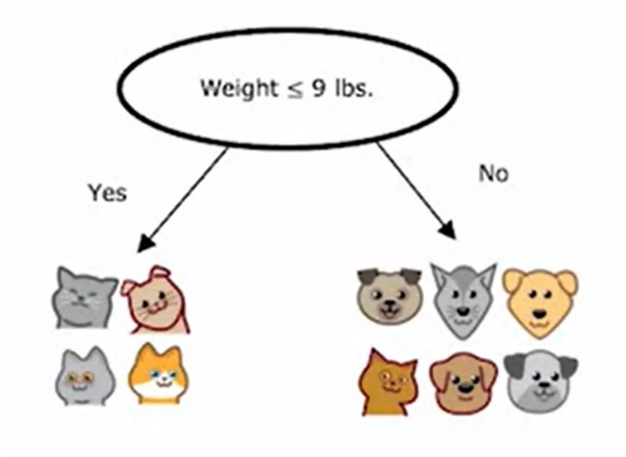
Continuous Features

	Ear shape	Face shape	Whiskers	Weight (lbs.)	Cat
	Pointy	Round	Present	7.2	1
	Floppy	Not round	Present	8.8	1
	Floppy	Round	Absent	15	0
	Pointy	Not round	Present	9.2	0
	Pointy	Round	Present	8.4	1
	Pointy	Round	Absent	7.6	1
	Floppy	Not round	Absent	11	0
	Pointy	Round	Absent	10.2	1
	Floppy	Round	Absent	18	0
	Floppy	Round	Absent	20	0

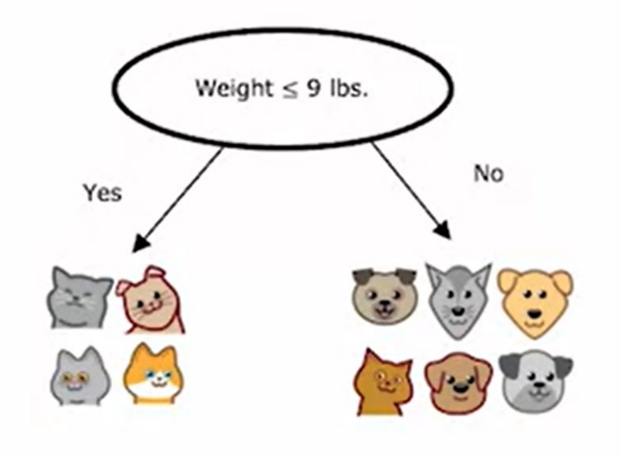
Splitting on a Continuous Variable

1. What is the best threshold?
2. How can we find?

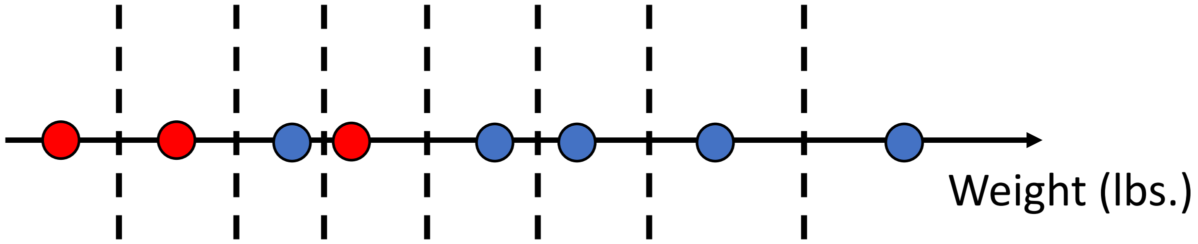




Splitting on a Continuous Variable



Try all possible thresholds.
Computational Complexity?



N: # training data
K: # classes
d: # feature dim

$$\left(\frac{|D^L|}{|D|}H(D^L) + \frac{|D^R|}{|D|}H(D^R)\right)$$

$$H(D) = -\sum_{k=1}^{K} p_k \log p_k$$

Regression Trees

Ear shape	Face shape	Whiskers	Weight (lbs.)
Pointy	Round	Present	7.2
Floppy	Not round	Present	8.8
Floppy	Round	Absent	15
Pointy	Not round	Present	9.2
Pointy	Round	Present	8.4
Pointy	Round	Absent	7.6
Floppy	Not round	Absent	11
Pointy	Round	Absent	10.2
Floppy	Round	Absent	18
Floppy	Round	Absent	20

x y

Regression Trees

$$D = \{(x_1, y_1), \dots, (x_N, y_N)\} \qquad y_i \in \mathbb{R}$$

$$H(D) = -\sum_{k=1}^{K} p_k \log p_k \longrightarrow$$

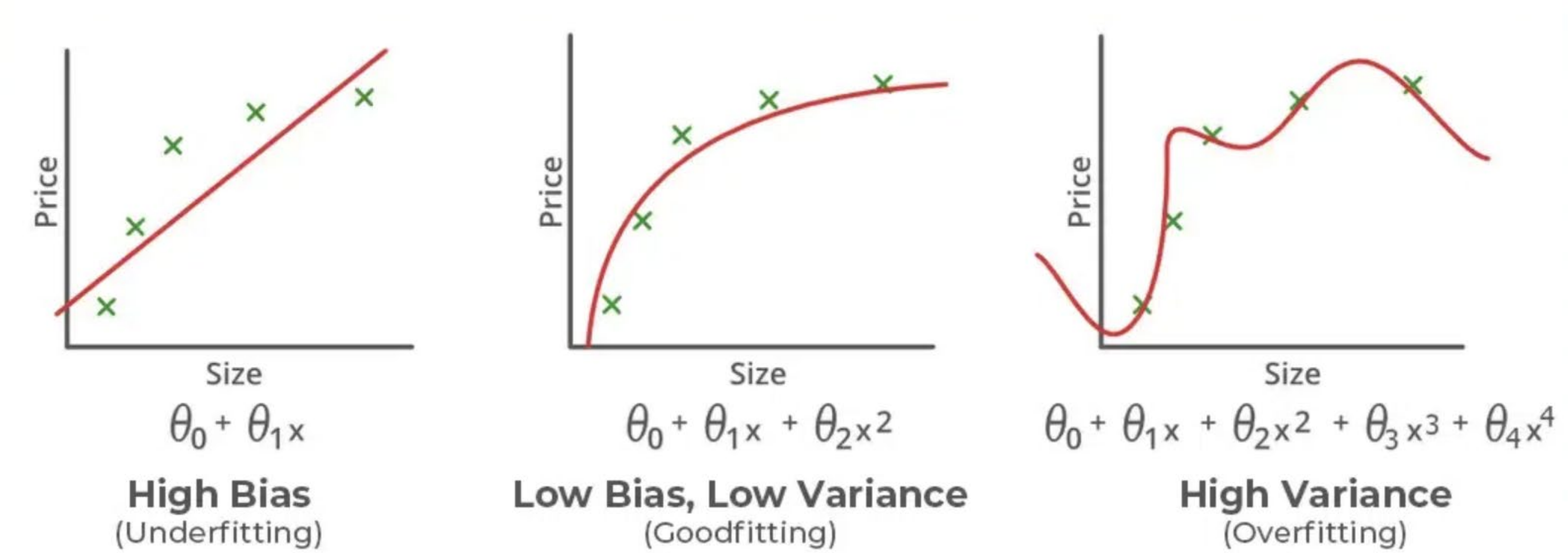
$$L(D) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)^2$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} y_i$$

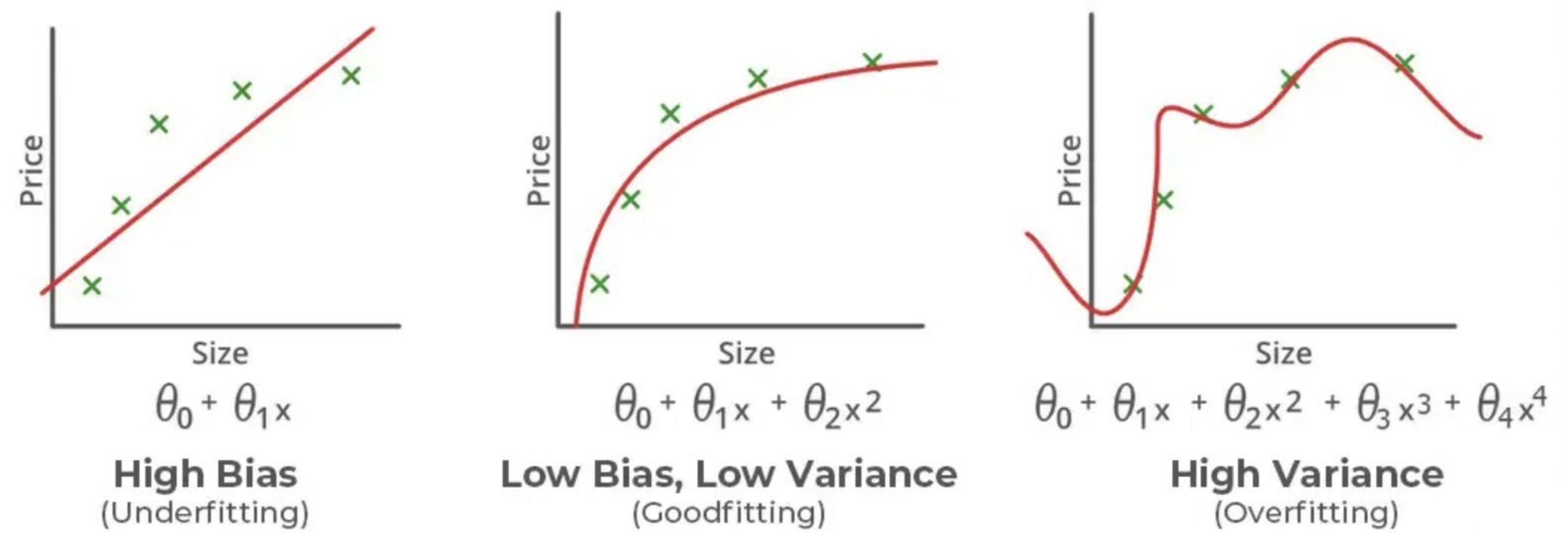
Bagging

(Bootstrap Aggregating)

Bias and Variance



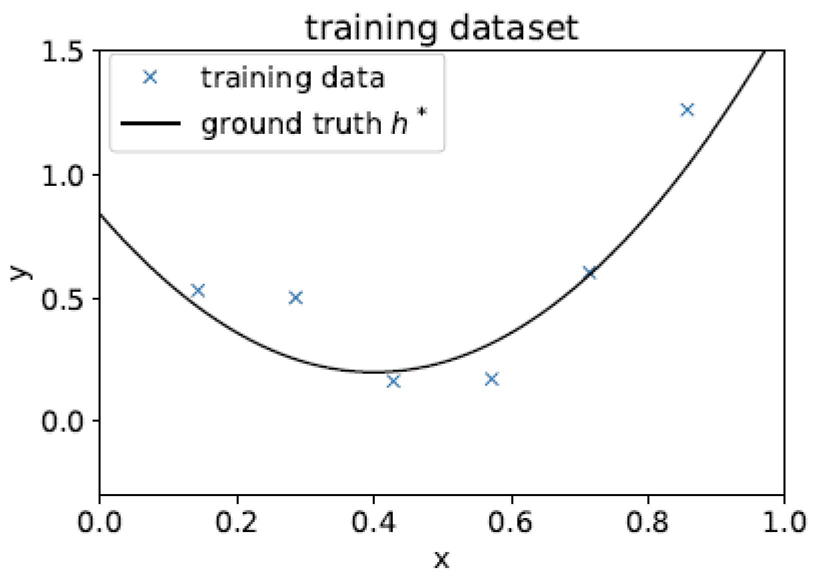
Bias and Variance

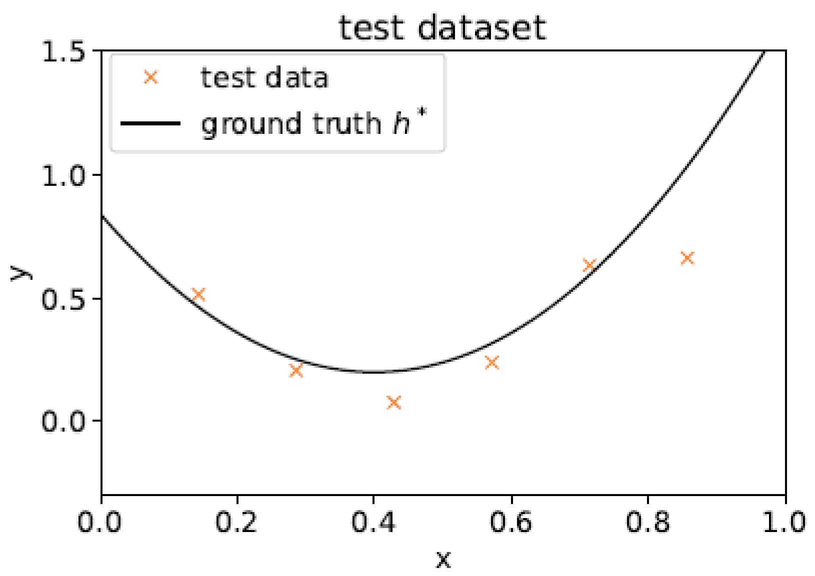


Is a decision tree high bias or high variance?

Bias and Variance Tradeoff

- Ground truth data

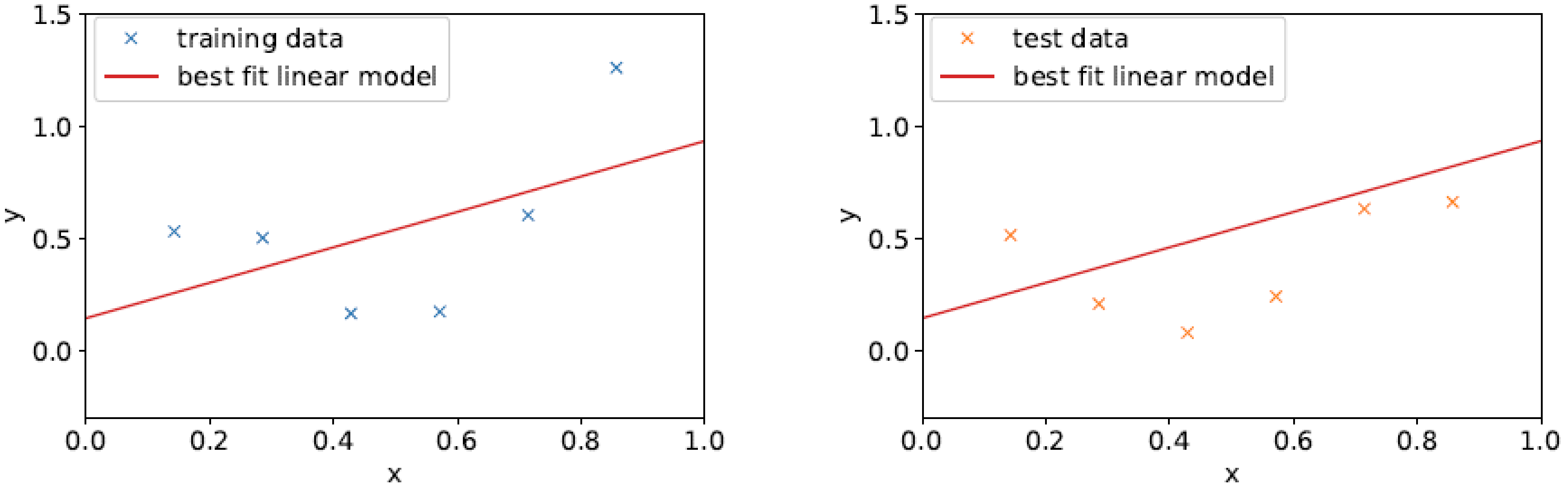




$$y^{(i)} = h^*\left(x^{(i)}\right) + \xi^{(i)} \qquad \xi^{(i)} \sim N(0, \sigma^2)$$

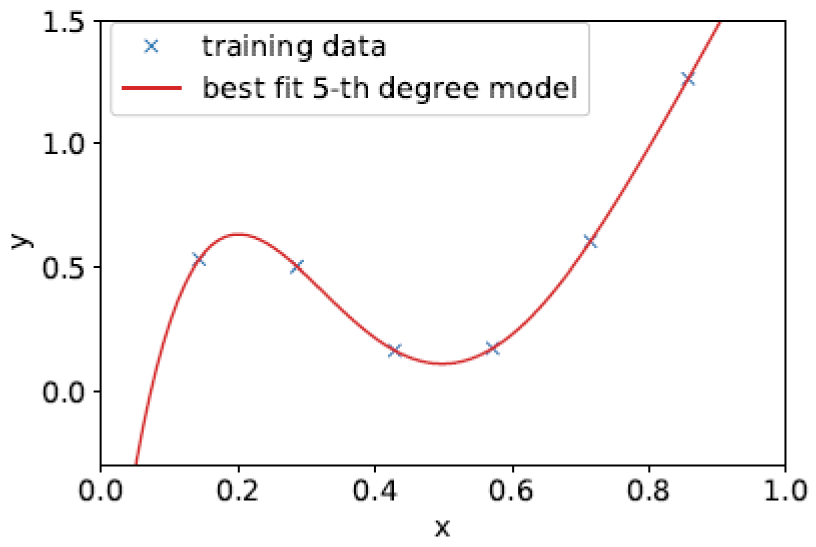
Bias and Variance Tradeoff

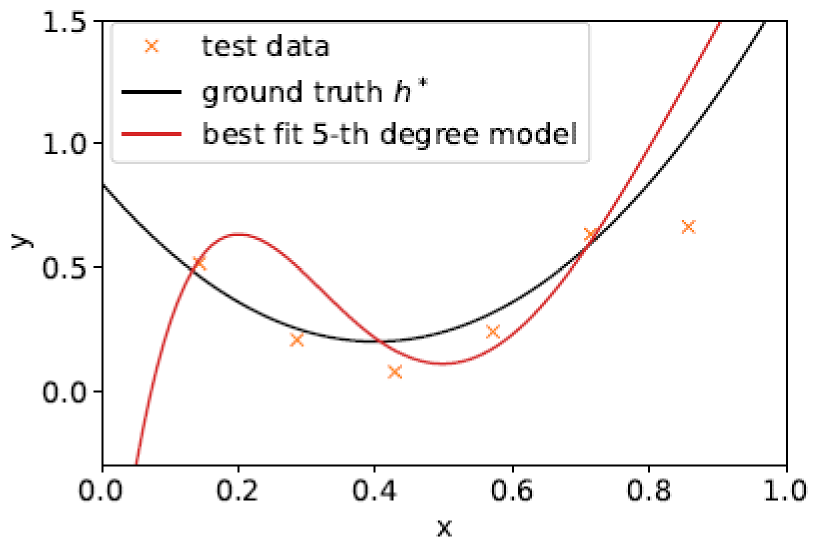
- Fitting a linear model to this data -> underfitting
- Linear model's inability
- More data doesn't really help
- High Bias



Bias and Variance Tradeoff

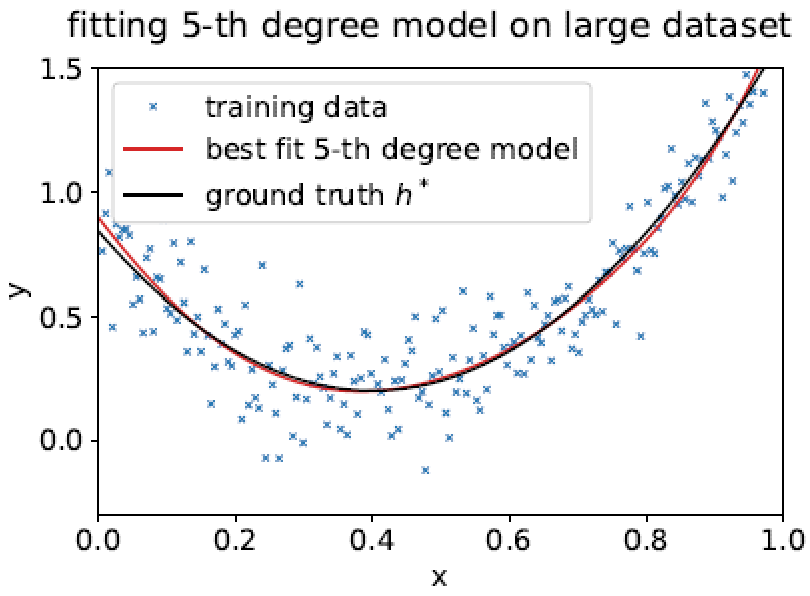
- Fitting a 5^{th} degree polynomial model to this data -> overfitting
- Too powerful





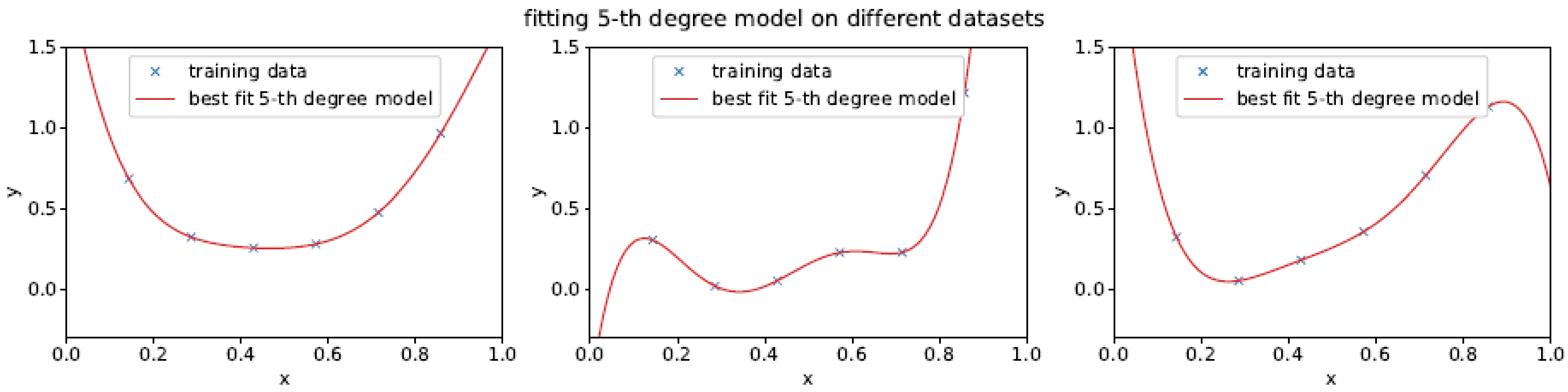
Bias and Variance Tradeoff

- If we had a very large dataset, then it would have been fine
  - Usually we don't have access to this data



Bias and Variance Tradeoff

- If we trained various models on different *'small'* and 'finite' training sets, we have a very different models for each dataset
- High variance



Mathematical Decomposition

- Draw a training dataset

$$S = \{x^{(i)}, y^{(i)}\}_{i=1}^{N} \qquad y^{(i)} = h^*\left(x^{(i)}\right) + \xi^{(i)} \qquad \xi^{(i)} \sim N(0, \sigma^2)$$

- Train a model on S, denoted by $\hat{h}_S$
- Take a test example (x, y), $y = h^*(x) + \xi$, and measure the expected error for the text example

$$MSE(x) = \mathbb{E}_{S,\xi}\left[\left(y - h_S(x)\right)^2\right]$$

Mathematical Decomposition

- Claim

$$\mathbb{E}[(A+B)^2] = \mathbb{E}[A^2] + \mathbb{E}[B^2]$$

if $\mathbb{E}[A] = 0$, and

A and B are independent

- Proof

$$\mathbb{E}[(A+B)^2] = \mathbb{E}[A^2] + \mathbb{E}[B^2] + 2\mathbb{E}[AB] = \mathbb{E}[A^2] + \mathbb{E}[B^2]$$

Mathematical Decomposition

$$MSE(x) = \mathbb{E}_{S,\xi}\left[\left(y - h_S(x)\right)^2\right] = \mathbb{E}_{S,\xi}[(\xi + h^*(x) - h_S(x))^2]$$

$$= \mathbb{E}_{\xi}[\xi^2] + \mathbb{E}_{S}[(h^*(x) - h_S(x))^2]$$

$$= \sigma^2 + \mathbb{E}_{S}[(h^*(x) - h_S(x))^2]$$

Mathematical Decomposition

$$MSE(x) = \mathbb{E}_{S,\xi}\left[\left(y - h_S(x)\right)^2\right] = \mathbb{E}_{S,\xi}[(\xi + h^*(x) - h_S(x))^2]$$

$$= \mathbb{E}_{\xi}[\xi^2] + \mathbb{E}_{S}[(h^*(x) - h_S(x))^2]$$

$$h_{avg} = \mathbb{E}_S[h_S(x)]$$

$$= \sigma^2 + \mathbb{E}_{S}[(h^*(x) - h_S(x))^2]$$

Model trained on infinite datasets

$$= \sigma^2 + \mathbb{E}_{S}\left[\left(h^*(x) - h_{avg}(x) + h_{avg}(x) - h_S(x)\right)^2\right]$$

Mathematical Decomposition

$$
\begin{aligned}
MSE(x) &= \mathbb{E}_{S,\xi}\left[\left(y - h_S(x)\right)^2\right] = \mathbb{E}_{S,\xi}[(\xi + h^*(x) - h_S(x))^2] \\
&= \mathbb{E}_{\xi}[\xi^2] + \mathbb{E}_{S}[(h^*(x) - h_S(x))^2] \\
&= \sigma^2 + \mathbb{E}_{S}[(h^*(x) - h_S(x))^2] \\
&= \sigma^2 + \mathbb{E}_{S}\left[\left(h^*(x) - h_{avg}(x) + h_{avg}(x) - h_S(x)\right)^2\right] \\
&= \sigma^2 + \left(h^*(x) - h_{avg}(x)\right)^2 + \mathbb{E}_{S}\left[\left(h_{avg}(x) - h_S(x)\right)^2\right]
\end{aligned}
$$

Mathematical Decomposition

$$MSE(x) = \mathbb{E}_{S,\xi}\left[\left(y - h_S(x)\right)^2\right] = \mathbb{E}_{S,\xi}[(\xi + h^*(x) - h_S(x))^2]$$

$$= \mathbb{E}_{\xi}[\xi^2] + \mathbb{E}_S[(h^*(x) - h_S(x))^2]$$

$$= \sigma^2 + \mathbb{E}_S[(h^*(x) - h_S(x))^2]$$

$$= \sigma^2 + \mathbb{E}_S\left[\left(h^*(x) - h_{avg}(x) + h_{avg}(x) - h_S(x)\right)^2\right]$$

$$= \sigma^2 + \left(h^*(x) - h_{avg}(x)\right)^2 + \mathbb{E}_S\left[\left(h_{avg}(x) - h_S(x)\right)^2\right]$$

$$= \sigma^2 + \left(h^*(x) - h_{avg}(x)\right)^2 + var(h_S(x))$$

Mathematical Decomposition

$$MSE(x) = \mathbb{E}_{S,\xi}\left[\left(y - h_S(x)\right)^2\right] = \mathbb{E}_{S,\xi}[(\xi + h^*(x) - h_S(x))^2]$$

$$= \mathbb{E}_{\xi}[\xi^2] + \mathbb{E}_S[(h^*(x) - h_S(x))^2]$$

$$= \sigma^2 + \mathbb{E}_S[(h^*(x) - h_S(x))^2]$$

$$= \sigma^2 + \mathbb{E}_S\left[\left(h^*(x) - h_{avg}(x) + h_{avg}(x) - h_S(x)\right)^2\right]$$

$$= \sigma^2 + \left(h^*(x) - h_{avg}(x)\right)^2 + \mathbb{E}_S\left[\left(h_{avg}(x) - h_S(x)\right)^2\right]$$

$$= \sigma^2 + \left(h^*(x) - h_{avg}(x)\right)^2 + var(h_S(x))$$

Bias Variance

Mathematical Decomposition

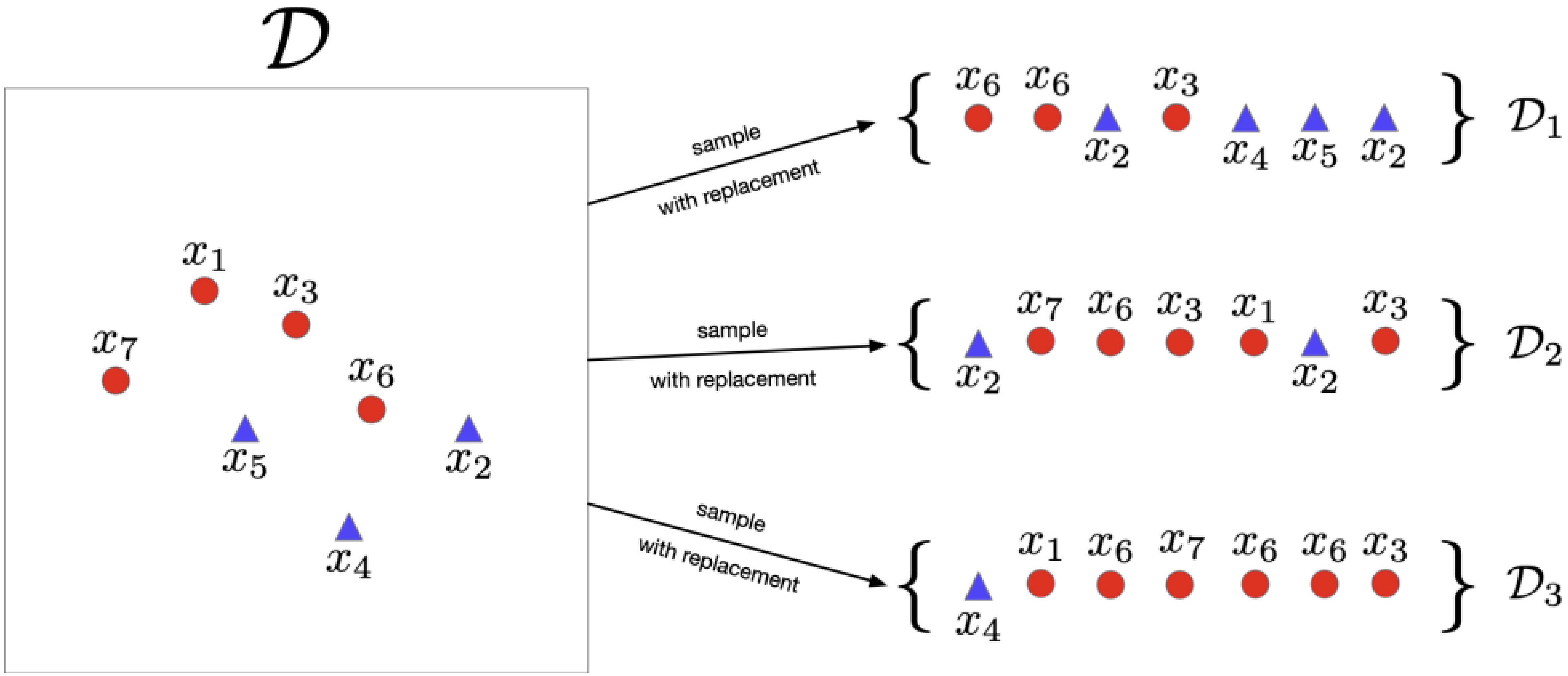
$$MSE(x) = \sigma^2 + \underbrace{\left(h^*(x) - h_{avg}(x)\right)^2}_{\text{Bias}} + \underbrace{\mathbb{E}_S\left[\left(h_{avg}(x) - h_S(x)\right)^2\right]}_{\text{Variance}}$$

- What if we can access to $h_{avg}(x) = \mathbb{E}_S[h_S(x)]$?
- Then, Bias terms stays the same, how about ‘variance’?

Bagging

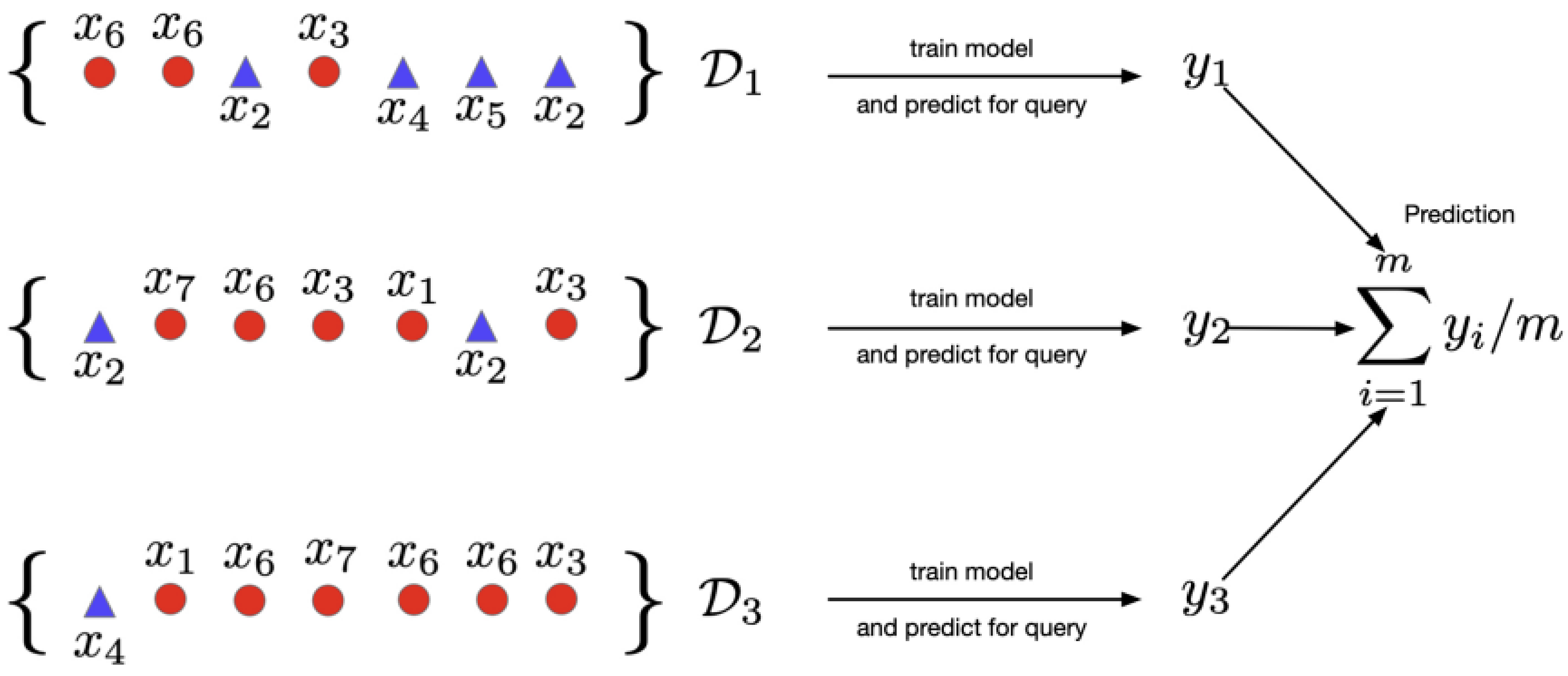
- In practice, the dataset is often finite and expensive to collect
- So, training separate models on independently sampled datasets is not feasible
- Solution: Given a training dataset D
  - Take a single dataset D with N examples
  - Generate M new datasets each by sampling N examples from D with replacement
  - Average the predictions of models trained on each of these datasets

Bagging



in this example $n = 7$, $m = 3$

Bagging



predicting on a query point x

Bagging

- Variance reduction

$$var\left[\frac{1}{M}\sum_{i=1}^{M} h_{D_i}(x)\right] = \frac{1}{M}(1-\rho)\sigma^2 + \rho\sigma^2$$

$var\left[h_{D_i}(x)\right] = \sigma^2$

$\rho =$ correlation factor

What happen if ρ goes to zero?

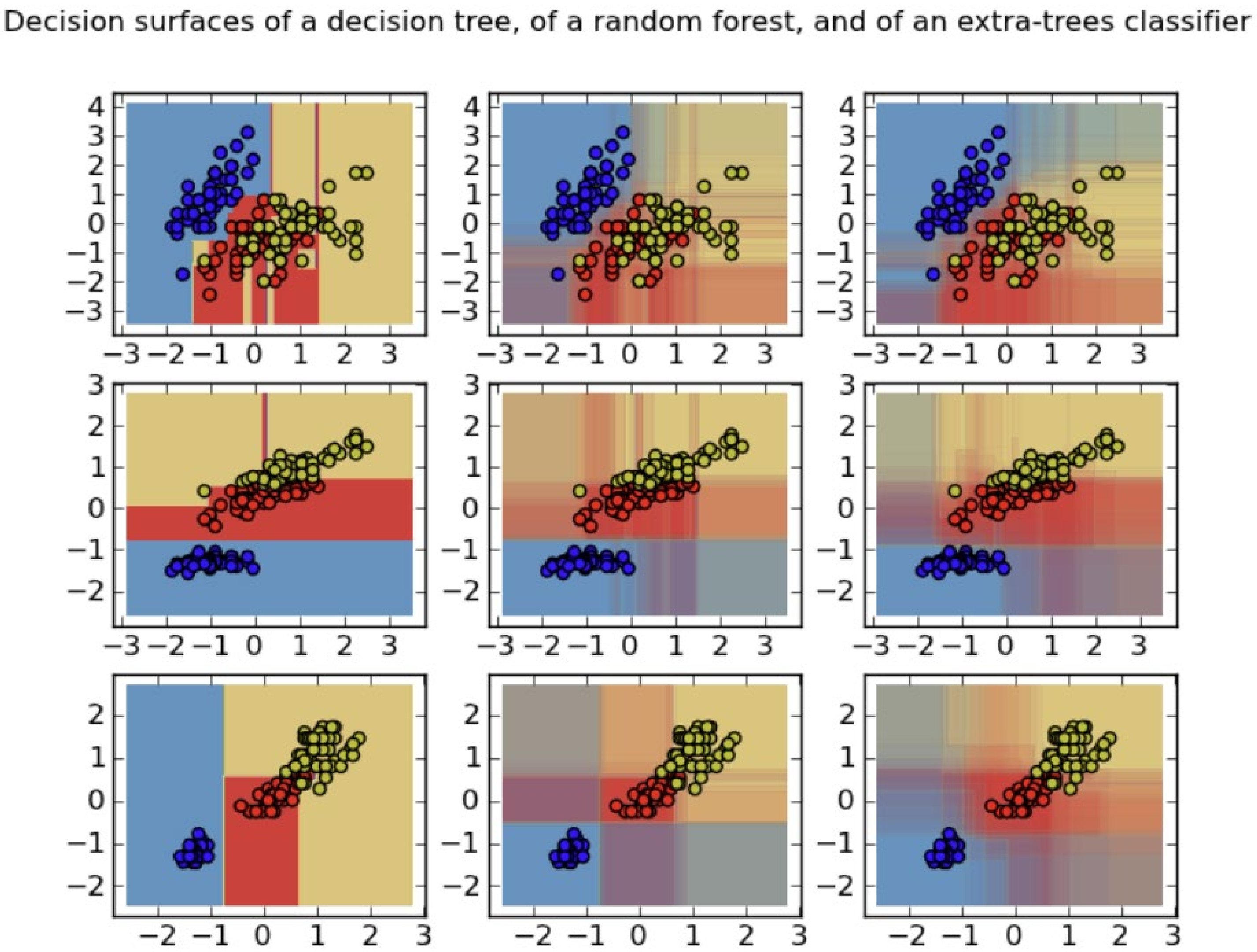
Random Forest

- Bagged decision trees, with one extra trick to decorrelate the predictions

- When choosing each node of the decision tree, choose a random set of d input features, and only consider splits on those features

$$k = \sqrt{d}$$

Random Forest



[Plot the decision surfaces of ensembles of trees on the iris dataset — scikit-learn 0.11-git documentation (ogrisel.github.io)]