Nearest Neighbor, Decision Tree

Danna Gurari
University of Texas at Austin
Spring 2021

https://www.ischool.utexas.edu/~dannag/Courses/IntroToMachineLearning/CourseContent.html

Review

- Last week:
  - Binary classification applications
  - Evaluating classification models
  - Biological neurons: inspiration
  - Artificial neurons: Perceptron & Adaline
  - Gradient descent
- Assignments (Canvas)
  - Lab assignment 1 due tonight
  - Problem set 3 due next week
- Questions?

Today’s Topics

- Multiclass classification applications and evaluating models
- Motivation for new ML era: need non-linear models
- Nearest neighbor classification
- Decision tree classification
- Parametric versus non-parametric models

Today’s Topics

- Multiclass classification applications and evaluating models
- Motivation for new ML era: need non-linear models
- Nearest neighbor classification
- Decision tree classification
- Parametric versus non-parametric models

Today’s Focus: Multiclass Classification

Predict 3+ classes

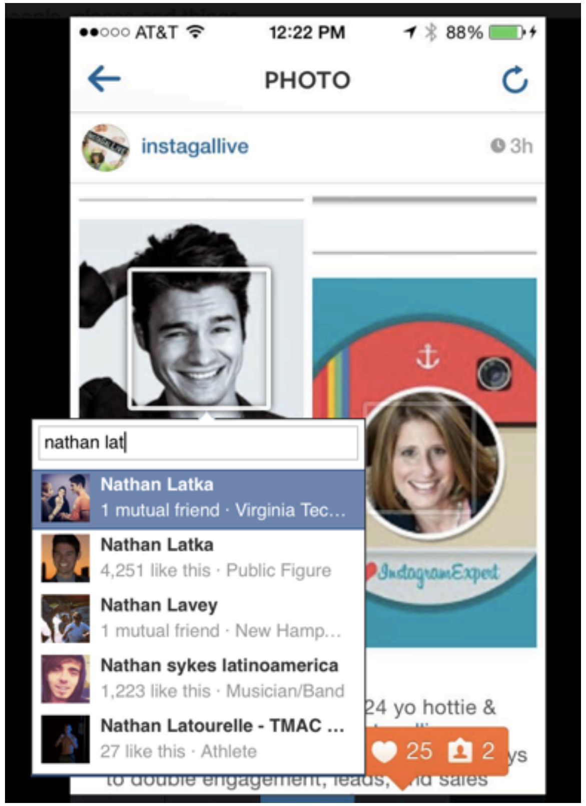
Multiclass Classification: Cancer Diagnosis

Doctor-level performance in recognizing 2,032 diseases

https://www.nature.com/articles/nature21056

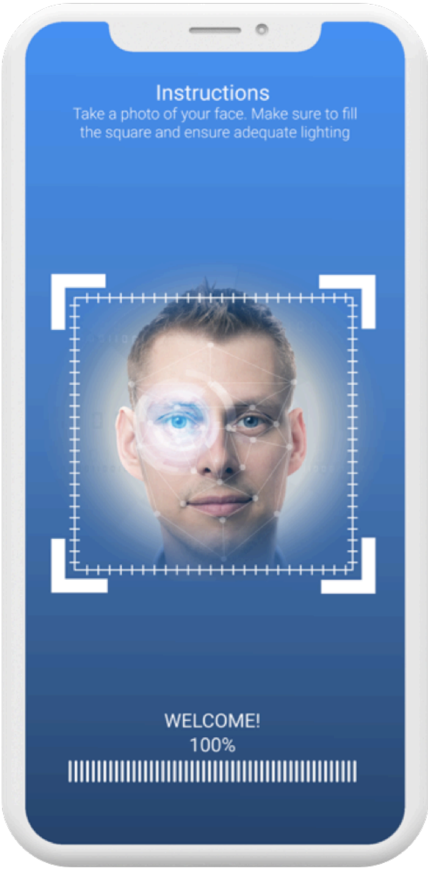
https://news.stanford.edu/2017/01/25/artificial-intelligence-used-identify-skin-cancer/

Multiclass Classification: Face Recognition



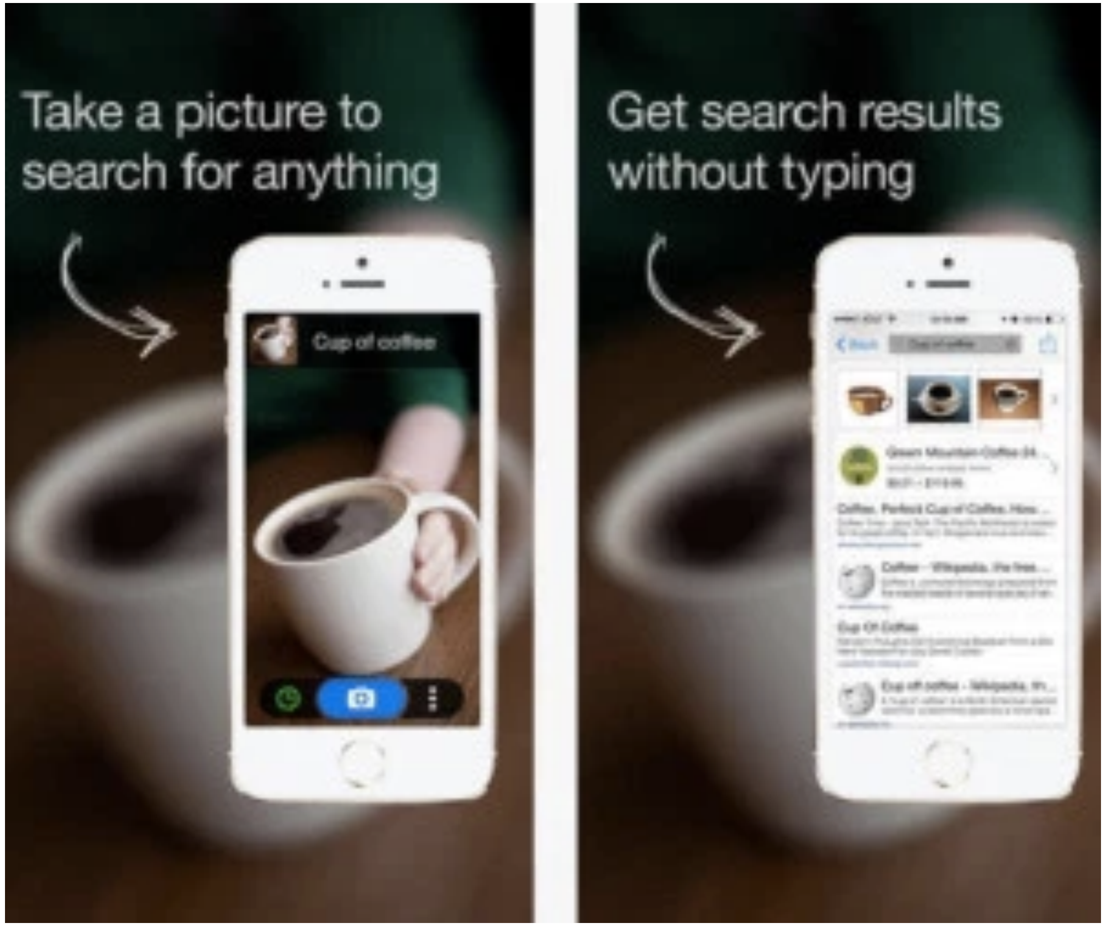
Social media

Visual assistance for people with vision impairments



Security
https://www.anyvision.co/

Multiclass Classification: Shopping

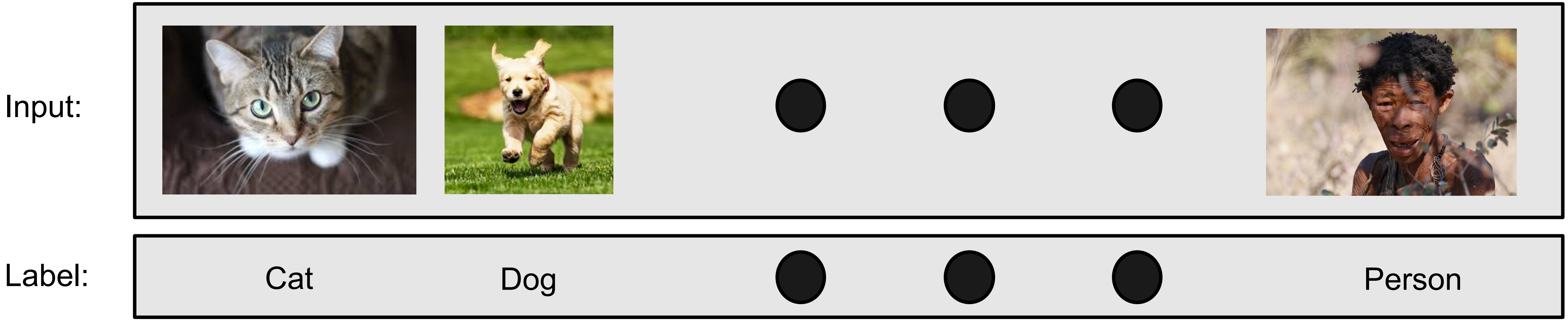


Camfind (https://venturebeat.com/2014/09/24/camfind-app-brings-accurate-visual-search-to-google-glass-exclusive/)

Multiclass Classification: Song Recognition

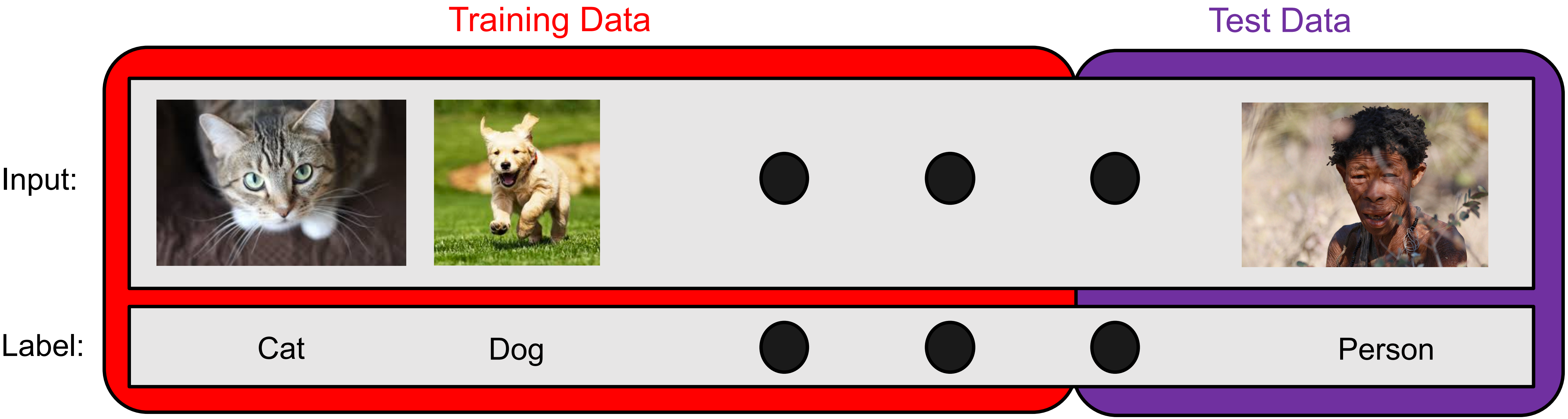
https://gbksoft.com/blog/how-to-make-a-shazam-like-app/

Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples



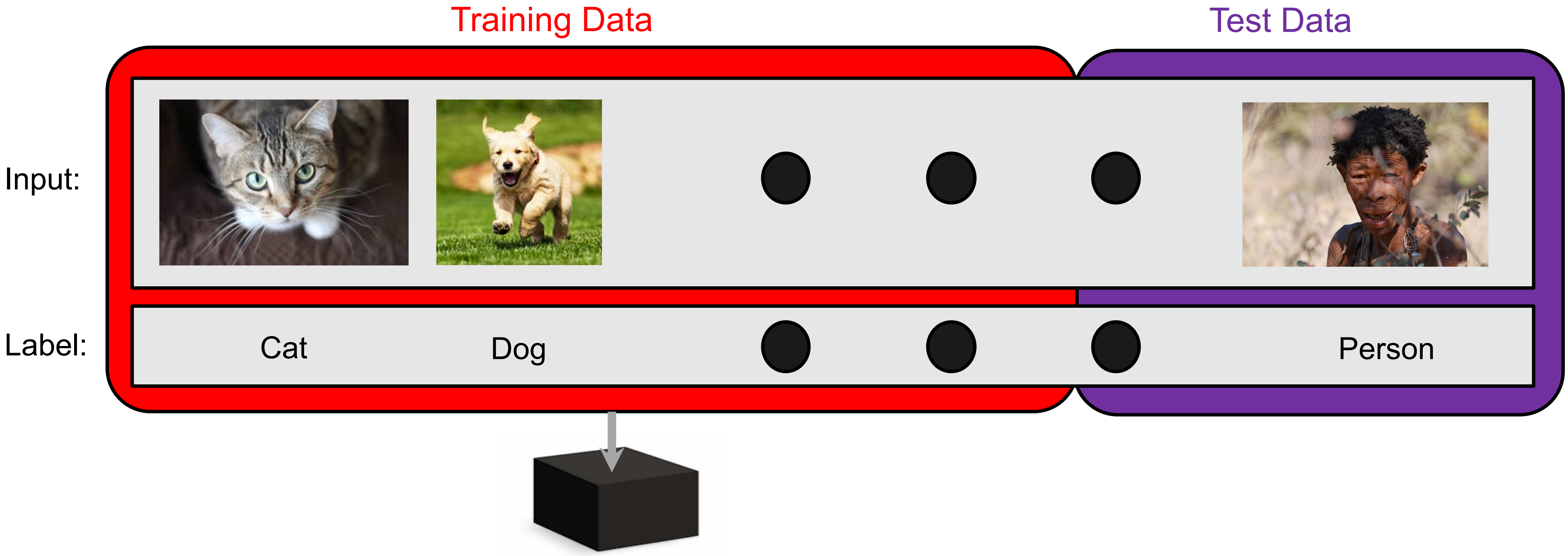
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

1. Split data into a "training set" and "test set"



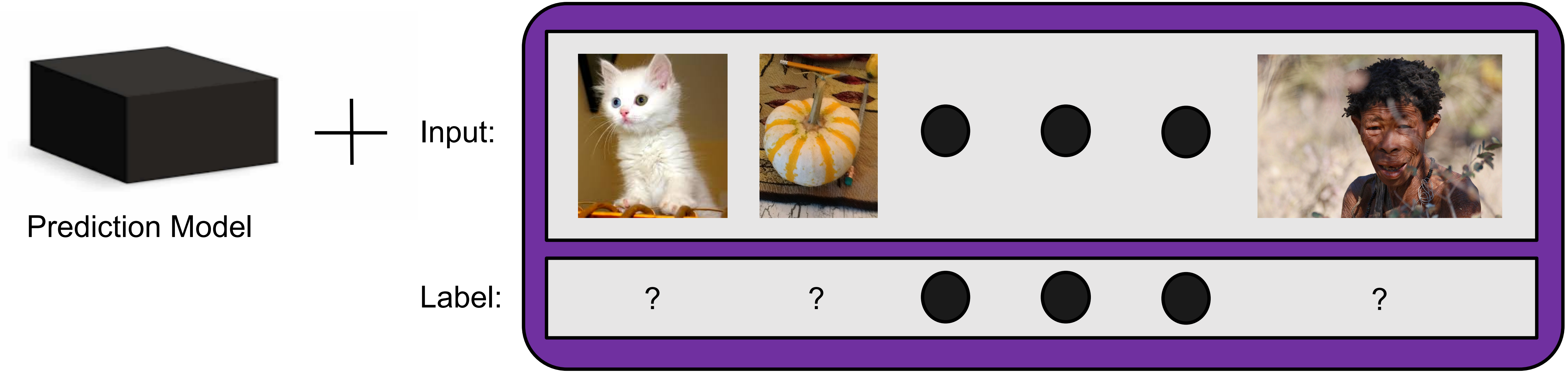
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

2. Train model on "training set" to try to minimize prediction error on it



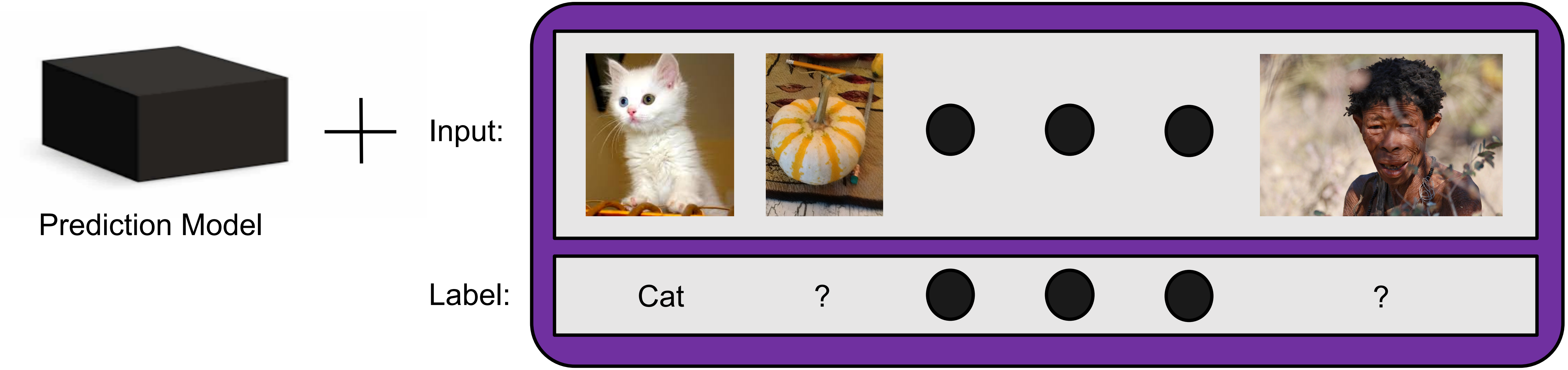
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error



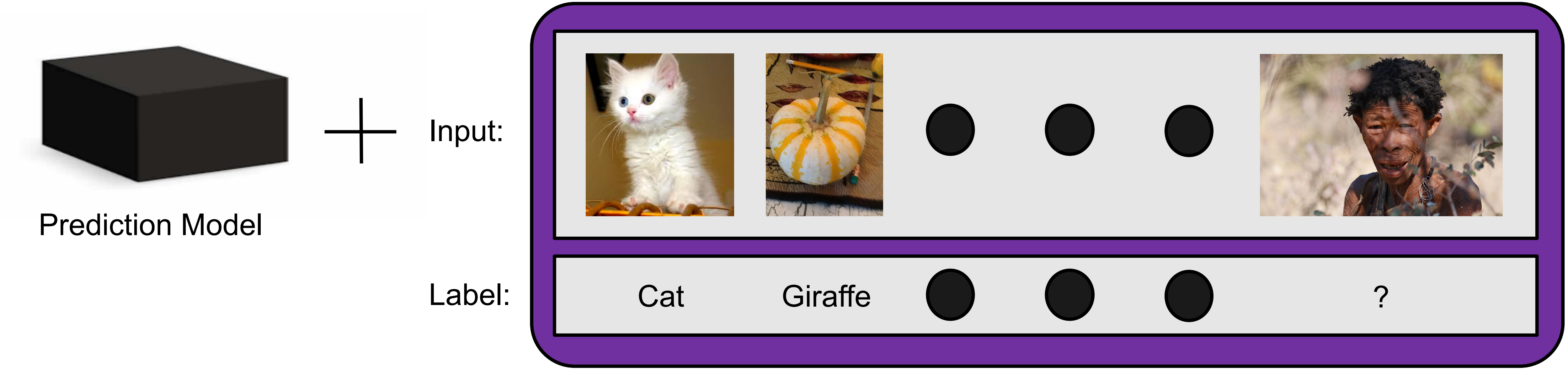
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error



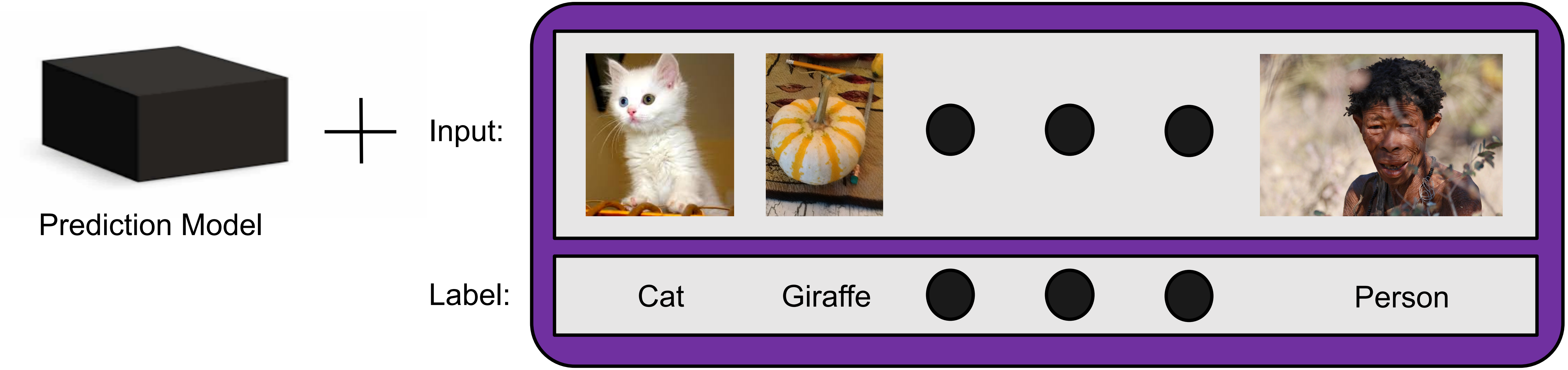
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on “test set” to measure generalization error



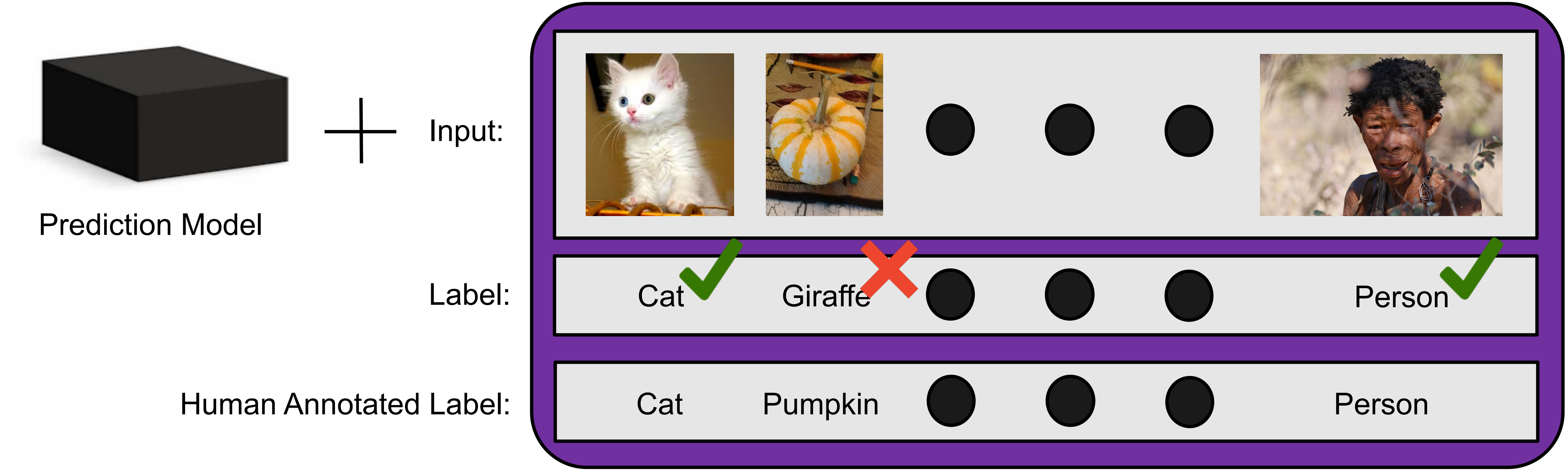
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on “test set” to measure generalization error



Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error



Evaluation Methods

- Confusion matrix; e.g.,

		Predicted			
		A	B	C	
True labels	A	2	2	0	4
	B	1	2	0	3
	C	0	0	3	3
		3	4	3	Total

http://gabrielelanaro.github.io/blog/2016/02/03/multiclass-evaluation-measures.html

- Accuracy: percentage correct
- Precision: $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
- Recall: $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$

Today’s Topics

- Multiclass classification applications and evaluating models
- Motivation for new ML era: need non-linear models
- Nearest neighbor classification
- Decision tree classification
- Parametric versus non-parametric models

Recall: Historical Context of ML Models

Recall: Vision for Perceptron

Frank Rosenblatt
(Psychologist)

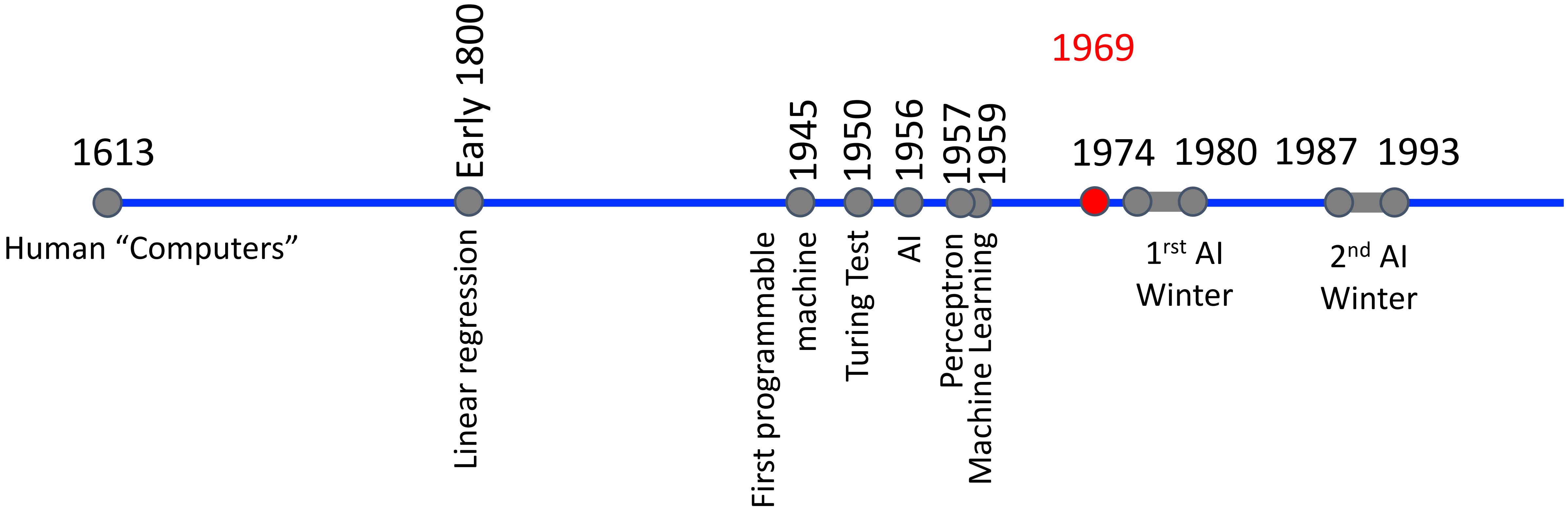
“[The perceptron is] the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence…. [It] is expected to be finished in about a year at a cost of $100,000.”

1958 New York Times article: https://www.nytimes.com/1958/07/08/archives/new-navy-device-learns-by-doing-psychologist-shows-embryo-of.html

https://en.wikipedia.org/wiki/Frank_Rosenblatt

“Perceptrons” Book: Instigator for “AI Winter”

Minsky & Papert publish a book called “Perceptrons” to discuss its limitations



Perceptron Limitation: XOR Problem

XOR = “Exclusive Or”
- Input: two binary values x_1 and x_2
- Output:
    - 1, when exactly one input equals 1
    - 0, otherwise

x_1	x_2	x_1 XOR x_2
0	0	?
0	1	?
1	0	?
1	1	?

Perceptron Limitation: XOR Problem

XOR = "Exclusive Or"
- Input: two binary values x_1 and x_2
- Output:
    - 1, when exactly one input equals 1
    - 0, otherwise

x_1	x_2	x_1 XOR x_2
0	0	?
0	1	?
1	0	?
1	1	?

Perceptron Limitation: XOR Problem

XOR = “Exclusive Or”
- Input: two binary values x_1 and x_2
- Output:
  - 1, when exactly one input equals 1
  - 0, otherwise

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	?
1	0	?
1	1	?

Perceptron Limitation: XOR Problem

XOR = “Exclusive Or”

- Input: two binary values x_1 and x_2
- Output:
  - 1, when exactly one input equals 1
  - 0, otherwise

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	?
1	1	?

Perceptron Limitation: XOR Problem

XOR = "Exclusive Or"

- Input: two binary values x_1 and x_2
- Output:
    - 1, when exactly one input equals 1
    - 0, otherwise

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	?

Perceptron Limitation: XOR Problem

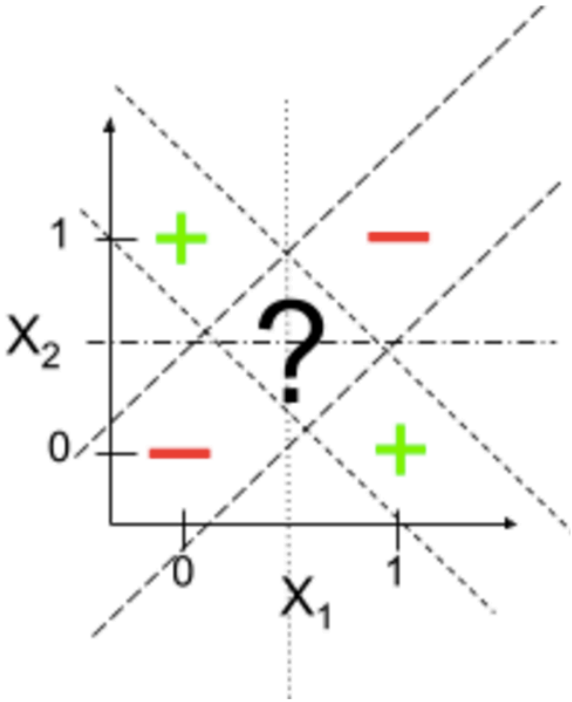
XOR = "Exclusive Or"

- Input: two binary values x_1 and x_2
- Output:
  - 1, when exactly one input equals 1
  - 0, otherwise

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptron Limitation: XOR Problem

How to separate 1s from 0s with a perceptron (linear function)?



x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptron Limitation: XOR Problem

Frank Rosenblatt
(Psychologist)

“[The perceptron is] the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its

How can a machine be “conscious” when it can’t solve the XOR problem?

navy-device-learns-by-doing-psychologist-shows-embryo-of.html

How to Overcome Limitation?

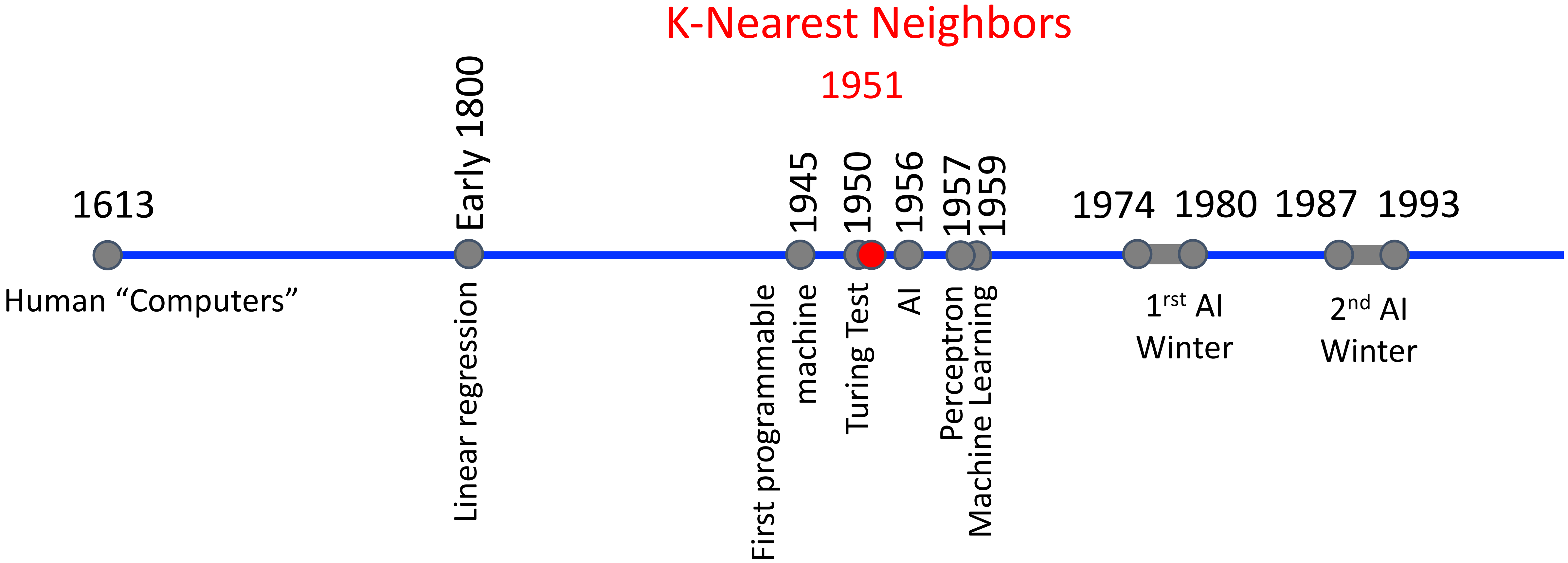
Non-linear models: e.g.,

- Linear regression: perform non-linear transformation of input features
- K-nearest neighbors
- Decision trees
- And many more to follow…

Today's Topics

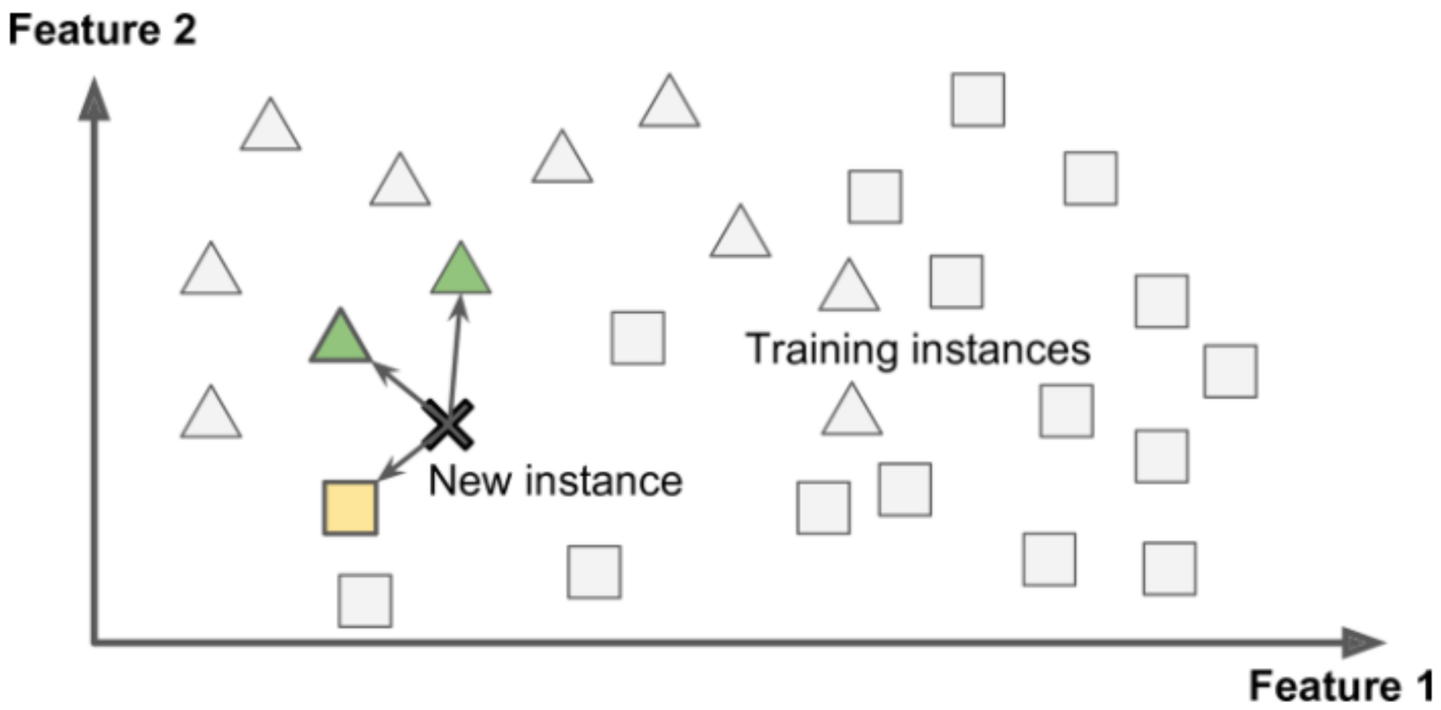
- Multiclass classification applications and evaluating models
- Motivation for new ML era: need non-linear models
- Nearest neighbor classification
- Decision tree classification
- Parametric versus non-parametric model

Historical Context of ML Models



E. Fix and J.L. Hodges. Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, 1951

Recall: Instance-Based Learning



Memorizes examples and uses a similarity measure to those examples to make predictions

Figure Source: Hands-on Machine Learning with Scikit-Learn & TensorFlow, Aurelien Geron

e.g., Predict What Scene The Image Shows

Input

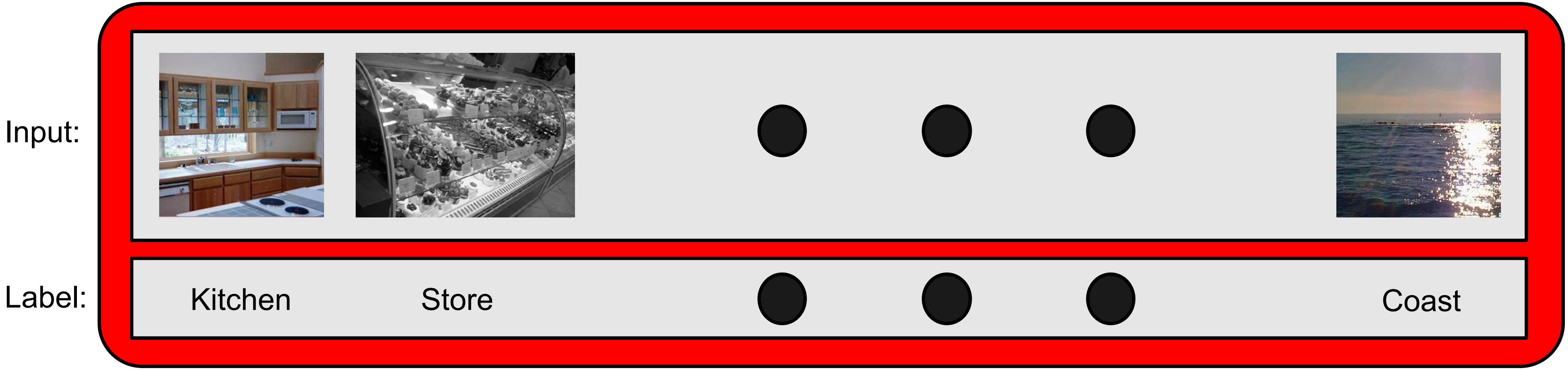
Output

Database Search

Kitchen

e.g., Predict What Scene The Image Shows

1. Create Large Database



e.g., Predict What Scene The Image Shows

2. Organize Database so Visually Similar Examples Neighbor Each Other

Adapted from slides by Antonio Torralba

e.g., Predict What Scene The Image Shows

3. Predict Class Using Label of Most Similar Example(s) in the Database

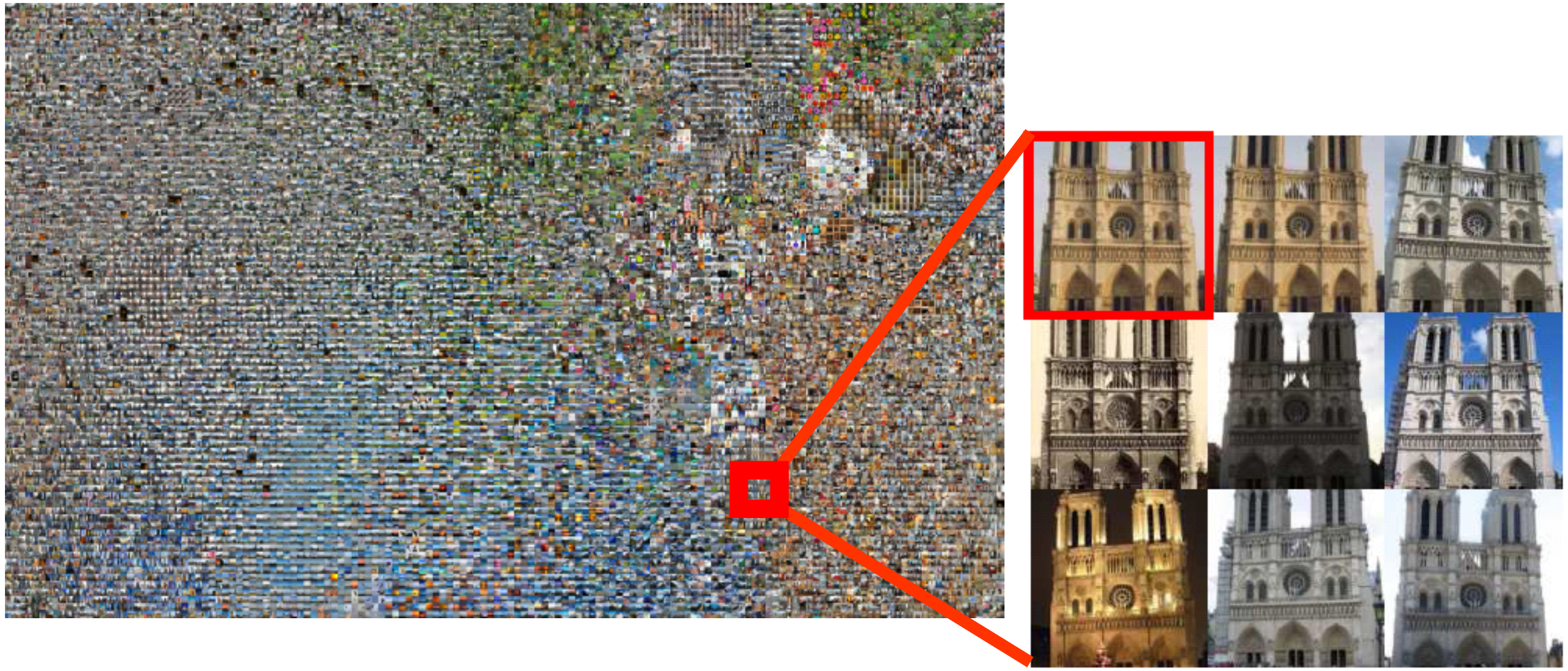
Input:

Label: ?

Adapted from slides by Antonio Torralba

e.g., Predict What Scene The Image Shows

3. Predict Class Using Label of Most Similar Example(s) in the Database



Input:

Label: Cathedral

Adapted from slides by Antonio Torralba

e.g., Predict What Scene The Image Shows

3. Predict Class Using Label of Most Similar Example(s) in the Database

Input: 

Label: ?

Adapted from slides by Antonio Torralba

e.g., Predict What Scene The Image Shows

3. Predict Class Using Label of Most Similar Example(s) in the Database

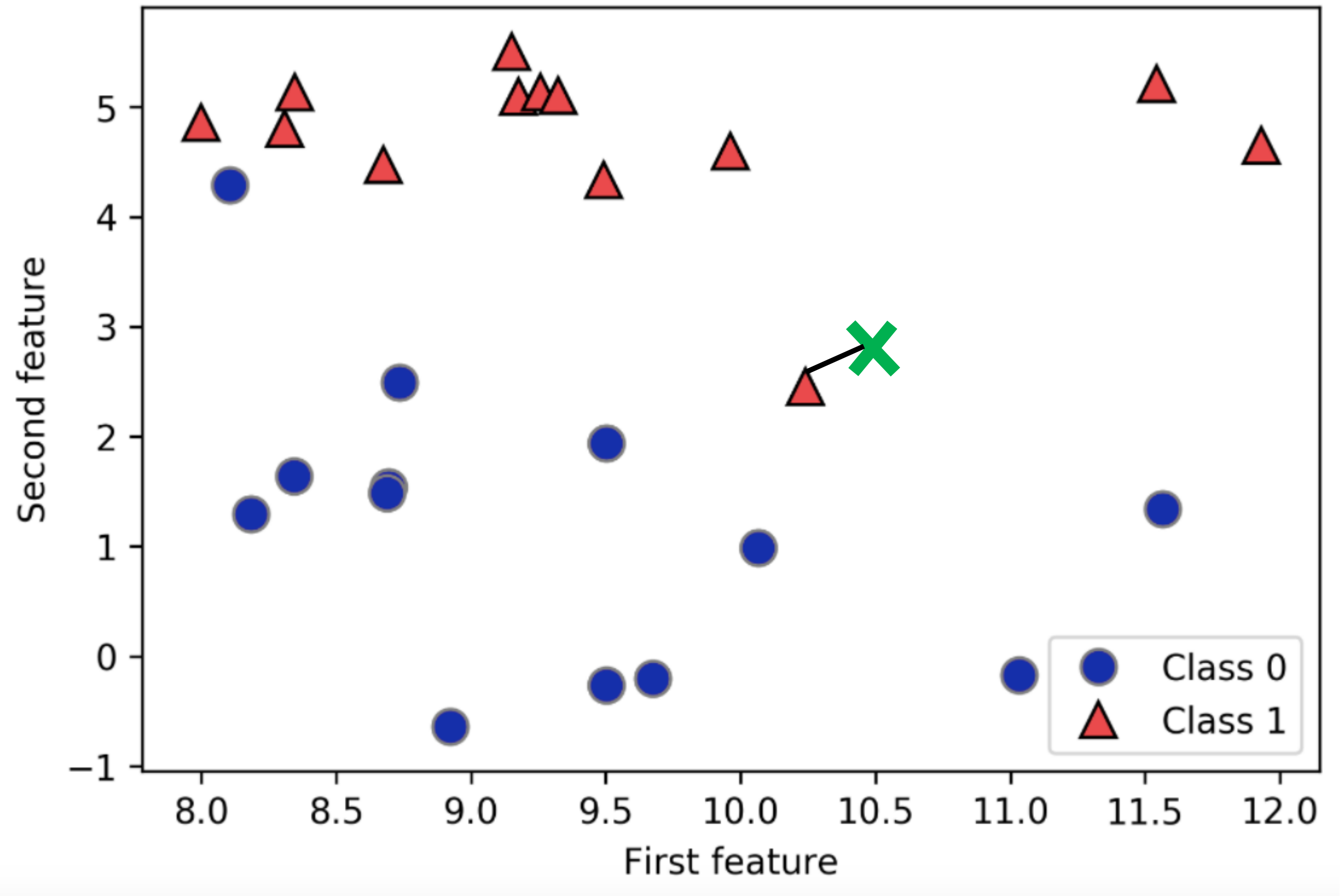
Input:

Label: Highway

Adapted from slides by Antonio Torralba

K-Nearest Neighbor Classification

Training Data:



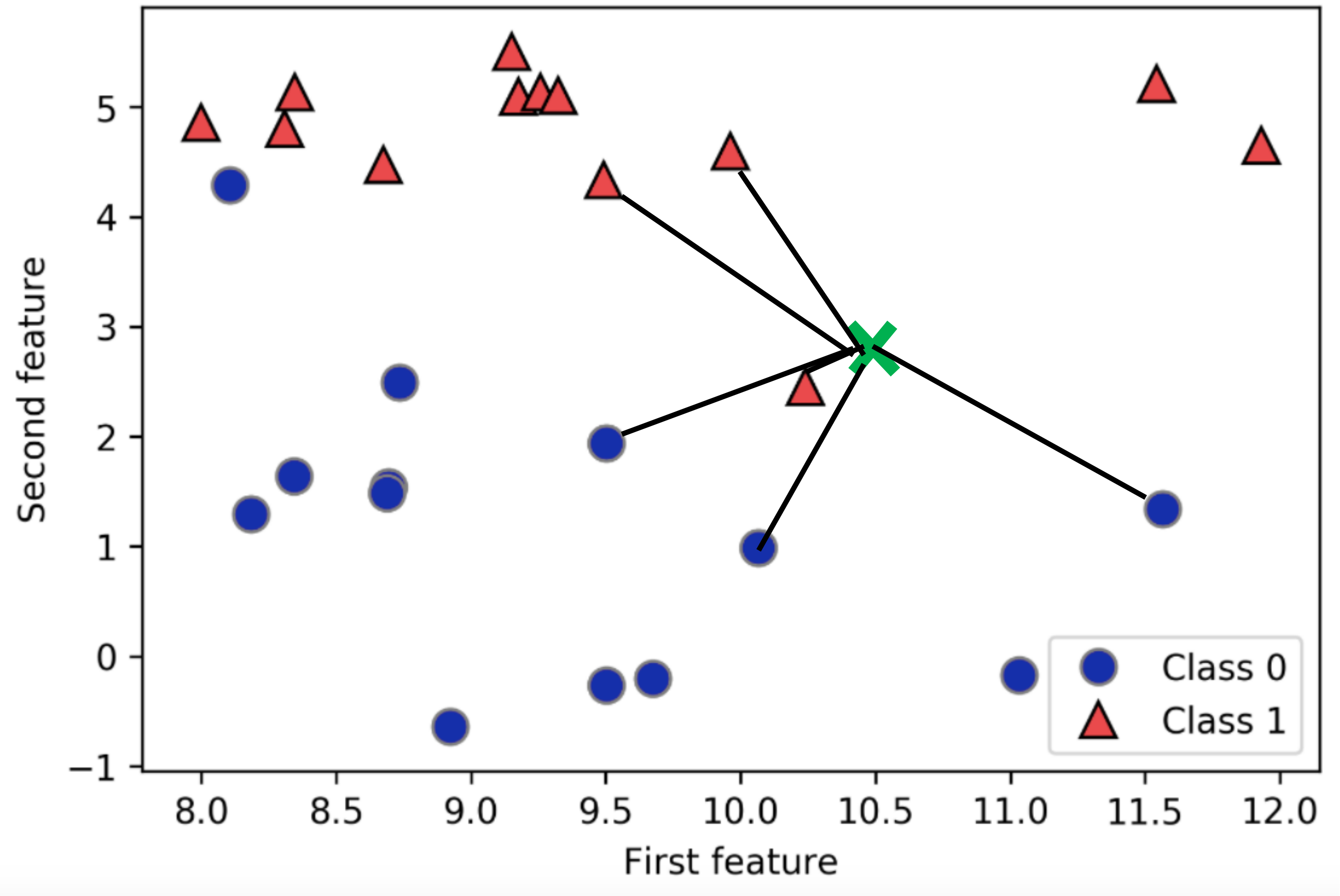
Novel Examples:

- Given:
  - **x = {10.5, 3}**
- Predict:
  - When k = **1:**

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification

Training Data:



Novel Examples:

- Given:
  - **x = {10.5, 3}**
- Predict:
  - When k = **1:**
    - Class 1
  - When k = **6:**
    - How to avoid ties?
      - Set “k” to odd value for binary problems
      - Prefer “closer” neighbors

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

How to measure distance between a novel example and test example?

- Commonly use, Minkowski distance:

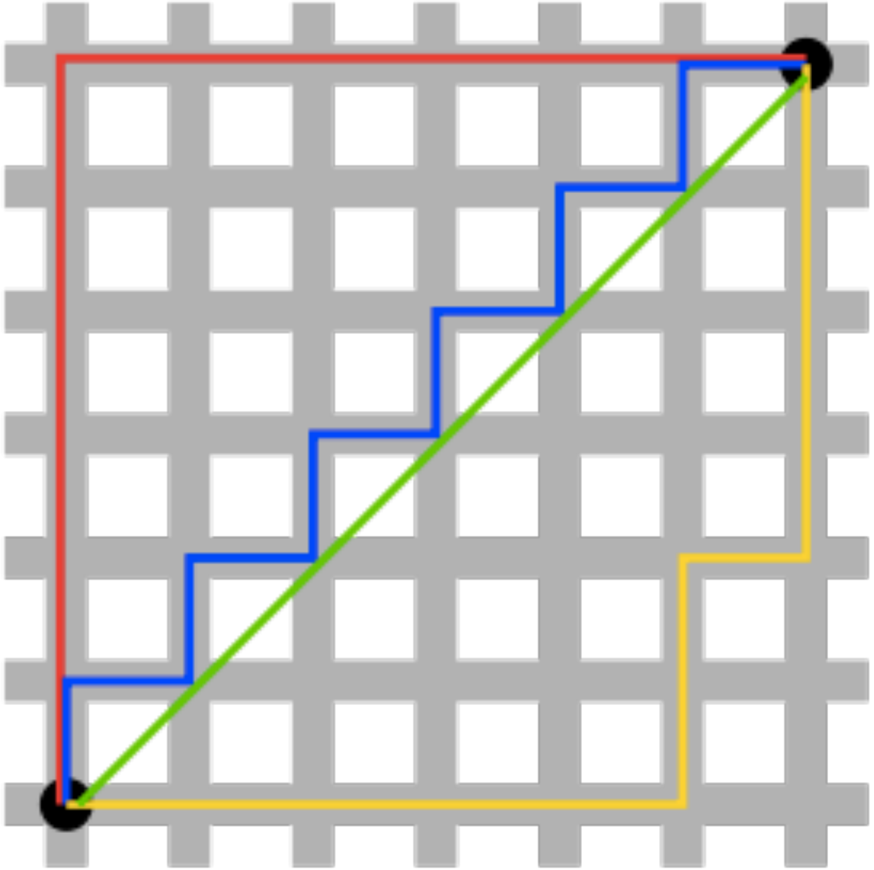
$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

  - When p = 2, Euclidean distance:

$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

  - When p = 1, Manhattan distance:

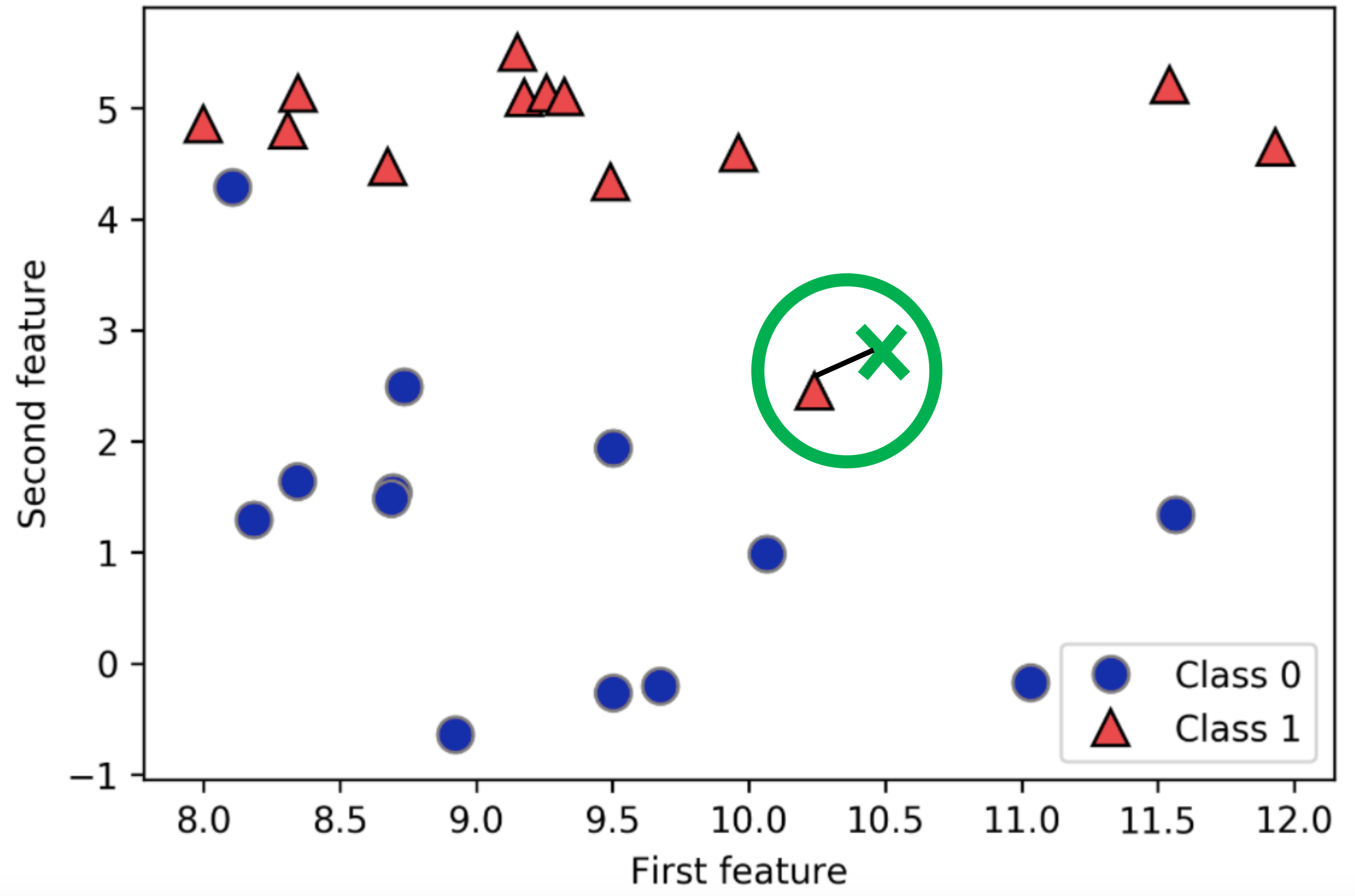
$$= \sum_{i=1}^{n} |x_i - y_i|$$



https://en.wikipedia.org/wiki/Taxicab_geometry

K-Nearest Neighbors: Measuring Distance

Training Data:



Euclidean Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

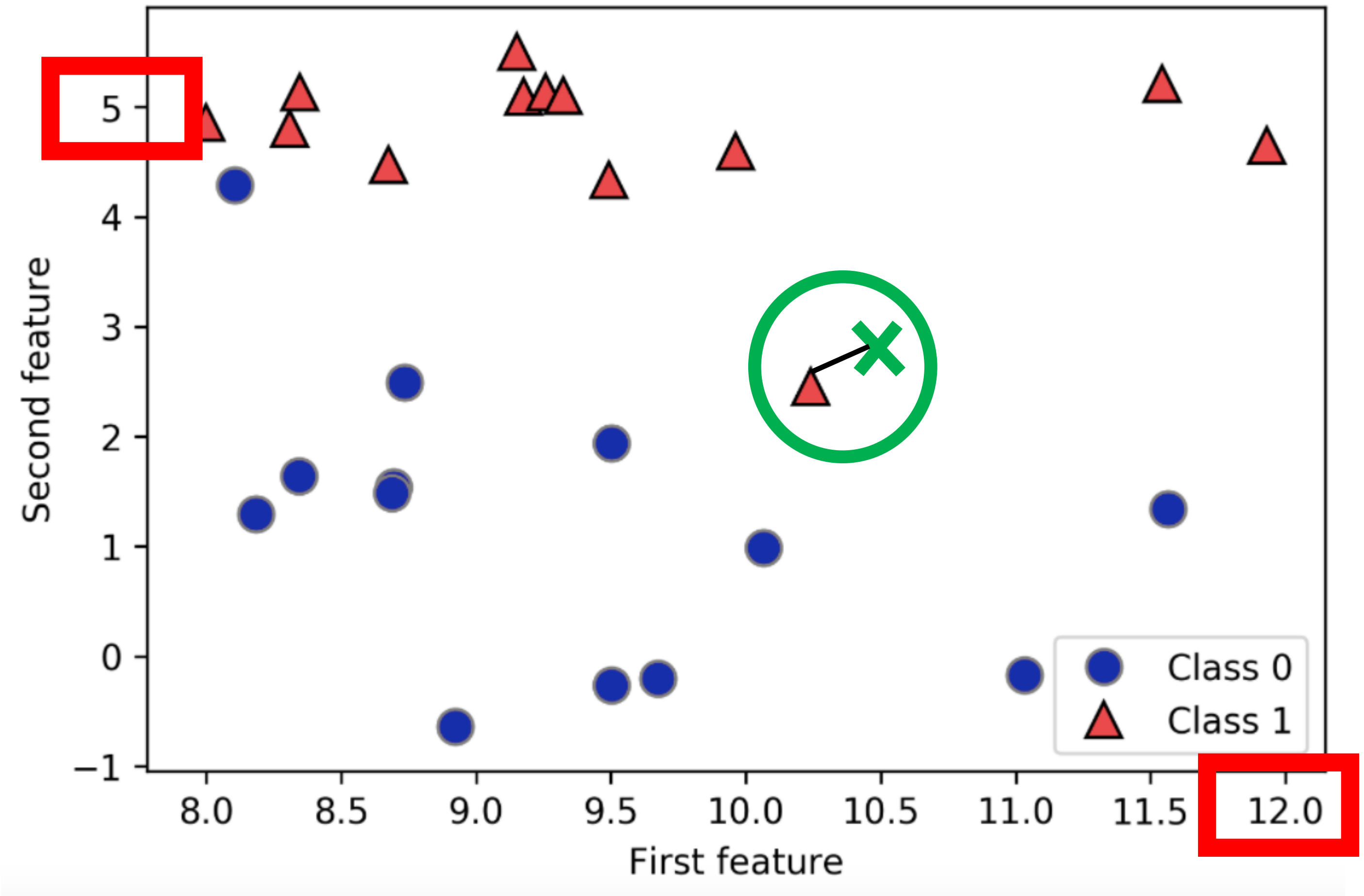
$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$dist = \sqrt{(10.5 - 10.1)^2 + (3 - 2.3)^2}$$
$$dist = \sqrt{0.4^2 + 0.7^2}$$
$$dist = \sqrt{0.16 + 0.49}$$
$$dist = \sqrt{0.65}$$
$$dist = 0.81$$

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

Training Data:



Euclidean Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$dist = \sqrt{(10.5 - 10.1)^2 + (3 - 2.3)^2}$$

$$dist = \sqrt{0.16 + 0.49}$$

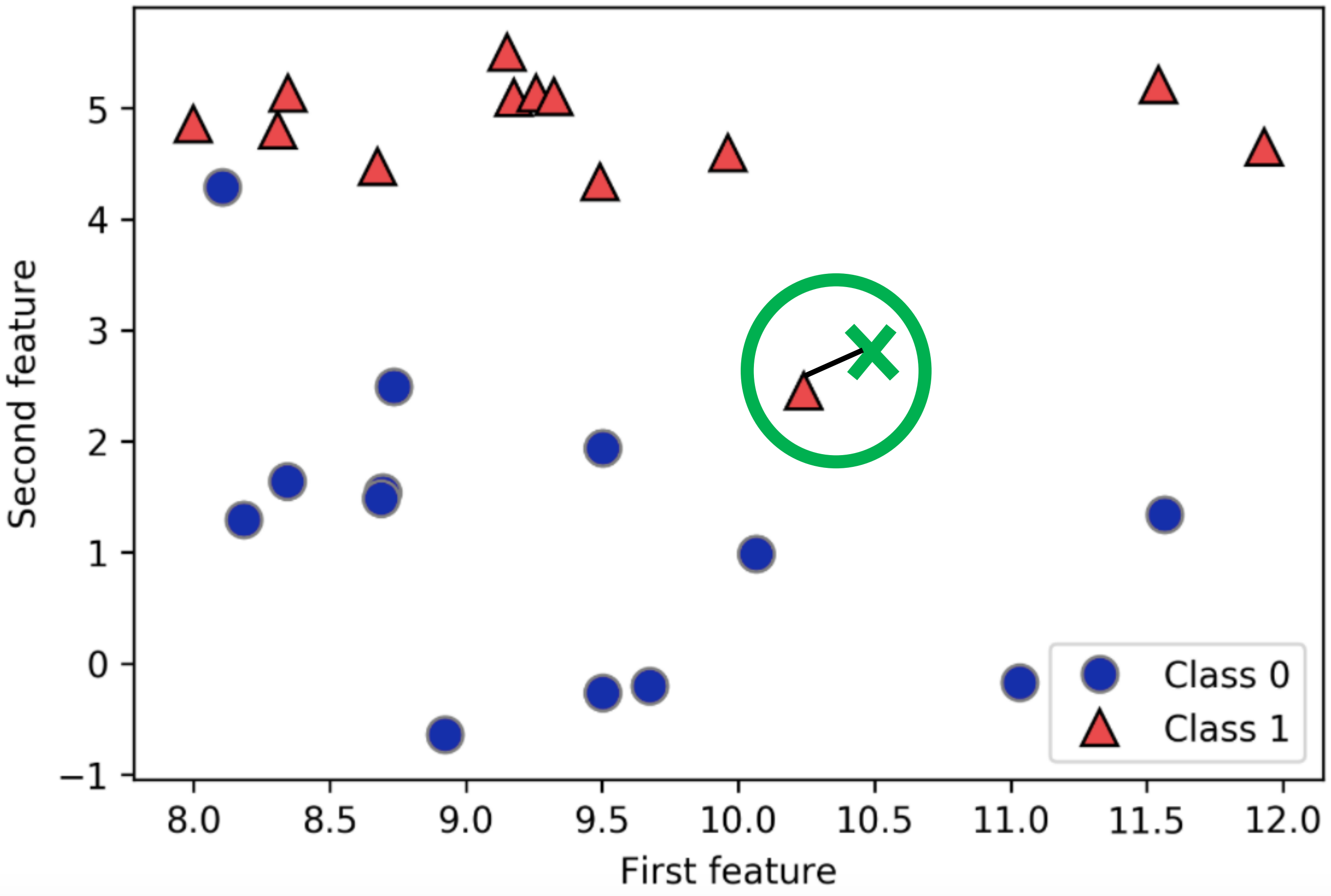
$$dist = 0.81$$

Note: May want to scale the data to the same range first

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

Training Data:



Manhattan Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

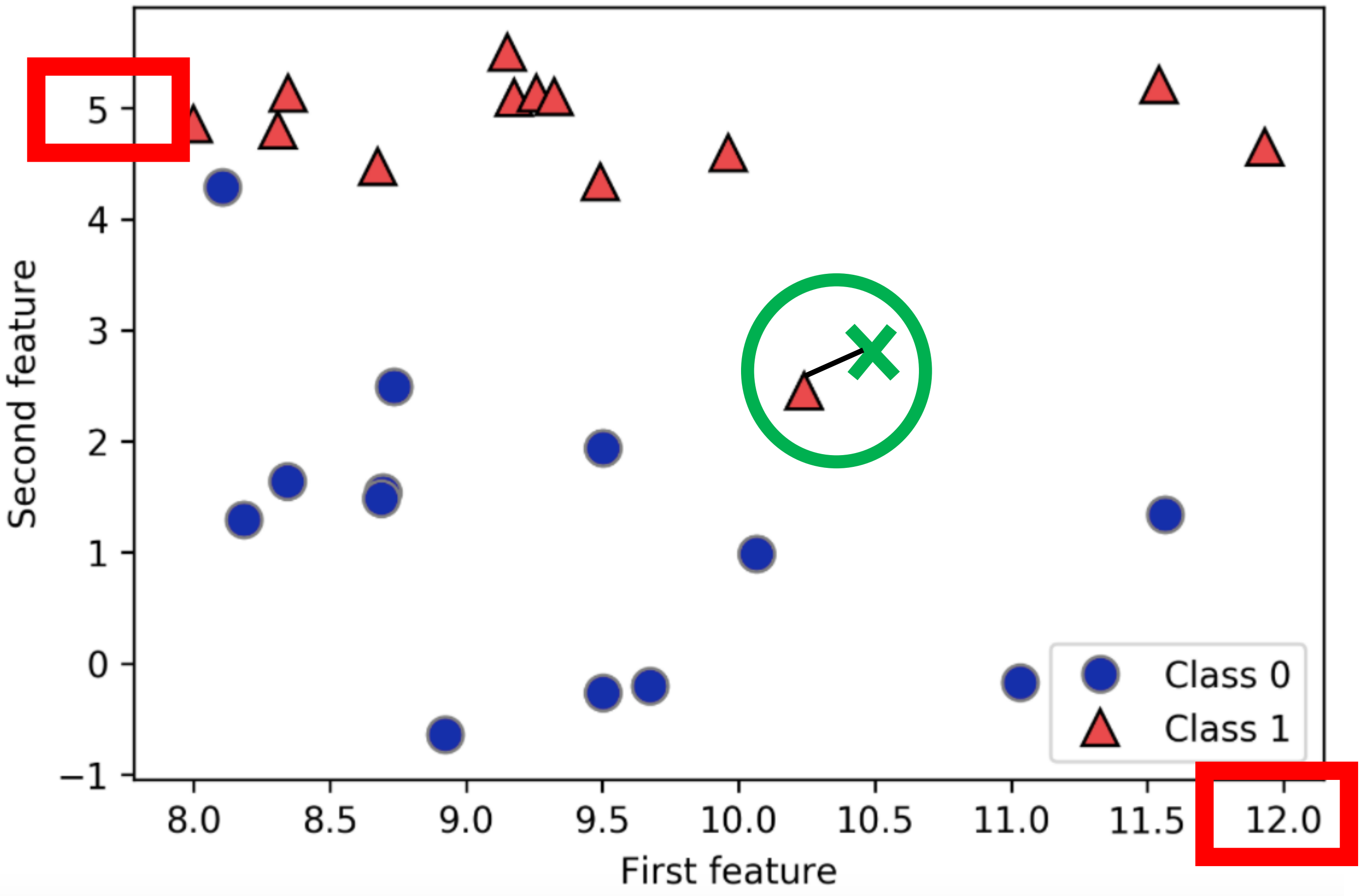
$$= \sum_{i=1}^{n} |x_i - y_i|$$

$$dist = |10.5 - 10.1| + |3 - 2.3|$$
$$dist = 0.4 + 0.7$$
$$dist = 1.1$$

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

Training Data:



Manhattan Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

$$= \sum_{i=1}^{n} |x_i - y_i|$$

$$dist = |10.5 - 10.1| + |3 - 2.3|$$

$$dist = 0.4 + 0.7$$

$$dist = 1.1$$

Note: May want to scale the data to the same range first

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

How to measure distance for novel categorical test data?

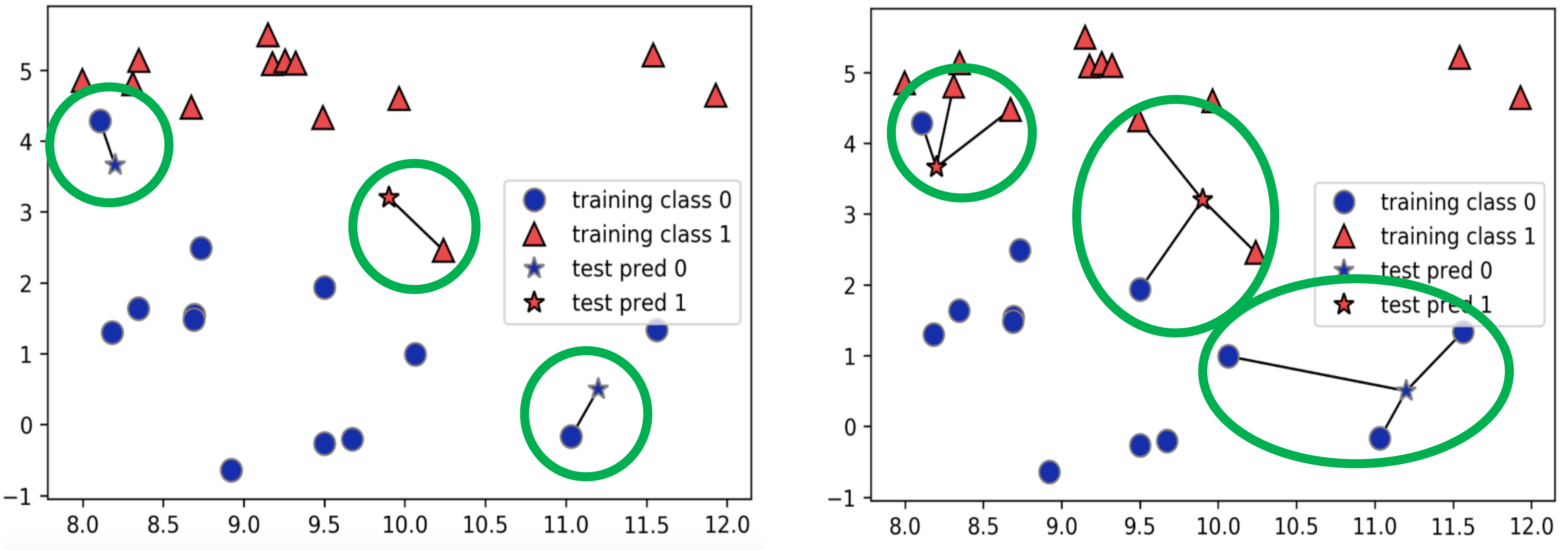
- e.g., Train = blue
- e.g., Test = blue; identical values so assign distance 0
- **e.g., Test = white; different values so assign distance 1**

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?

When K=1:

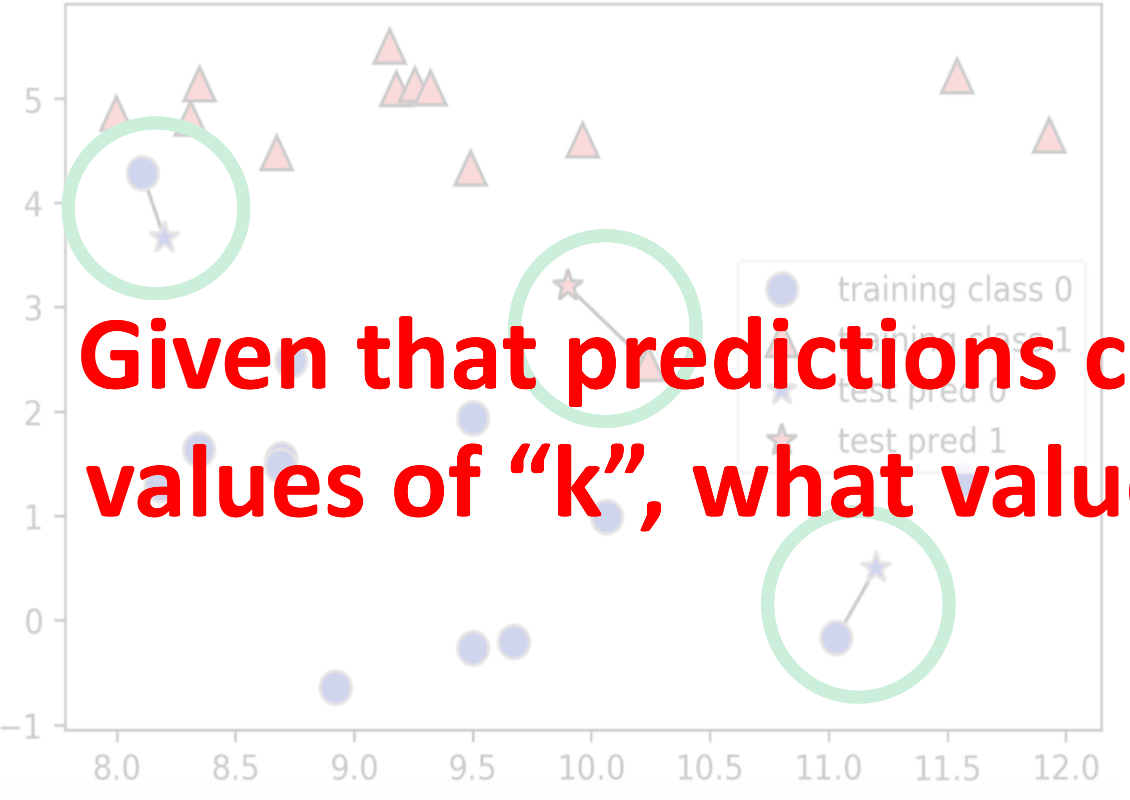
When K=3:



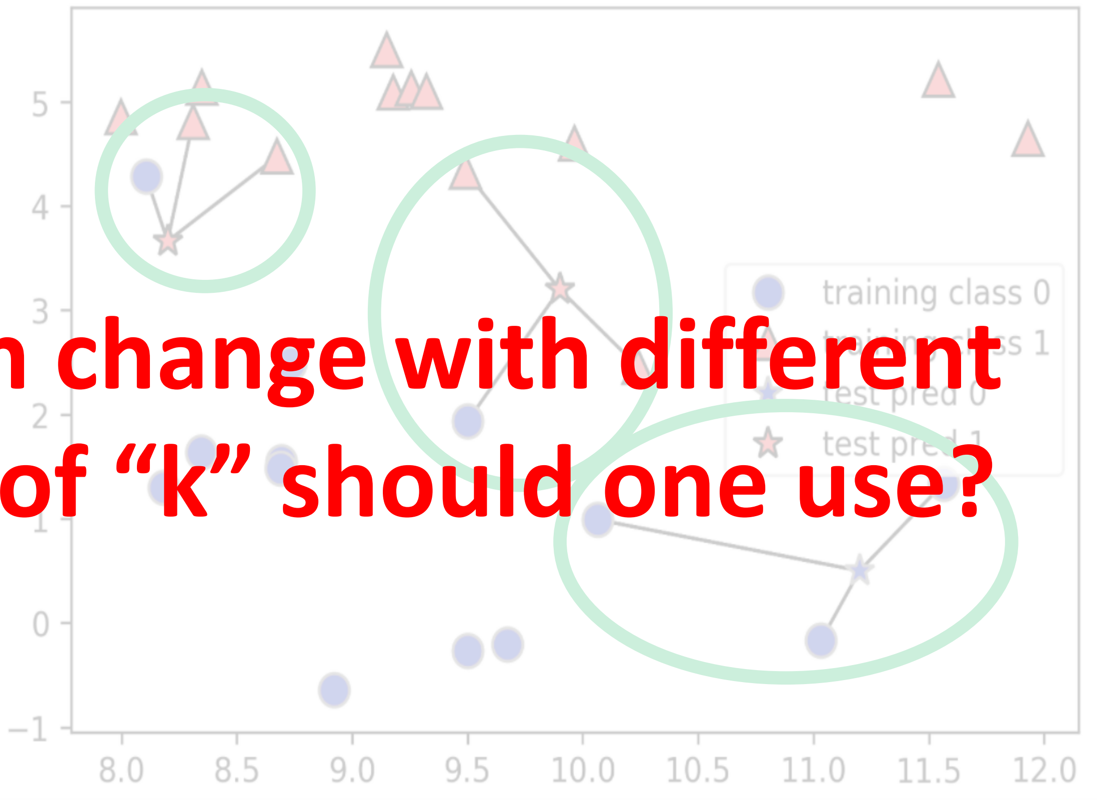
https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?

When K=1:



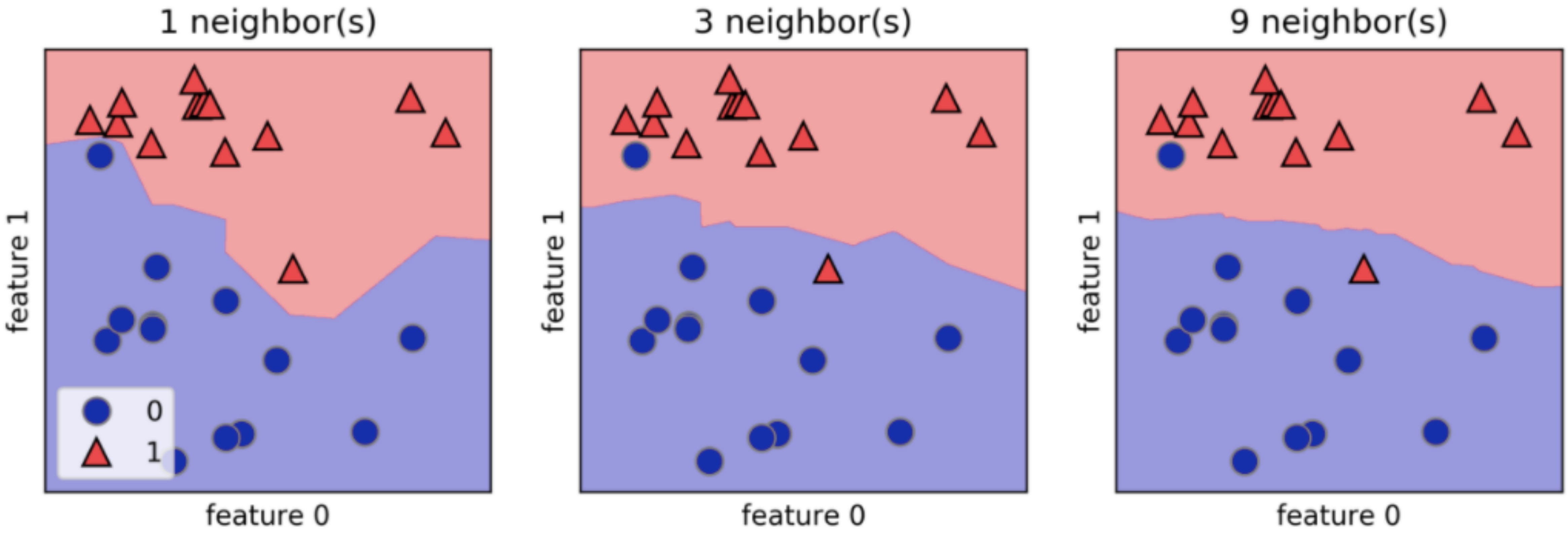
When K=3:



Given that predictions can change with different values of “k”, what value of “k” should one use?

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

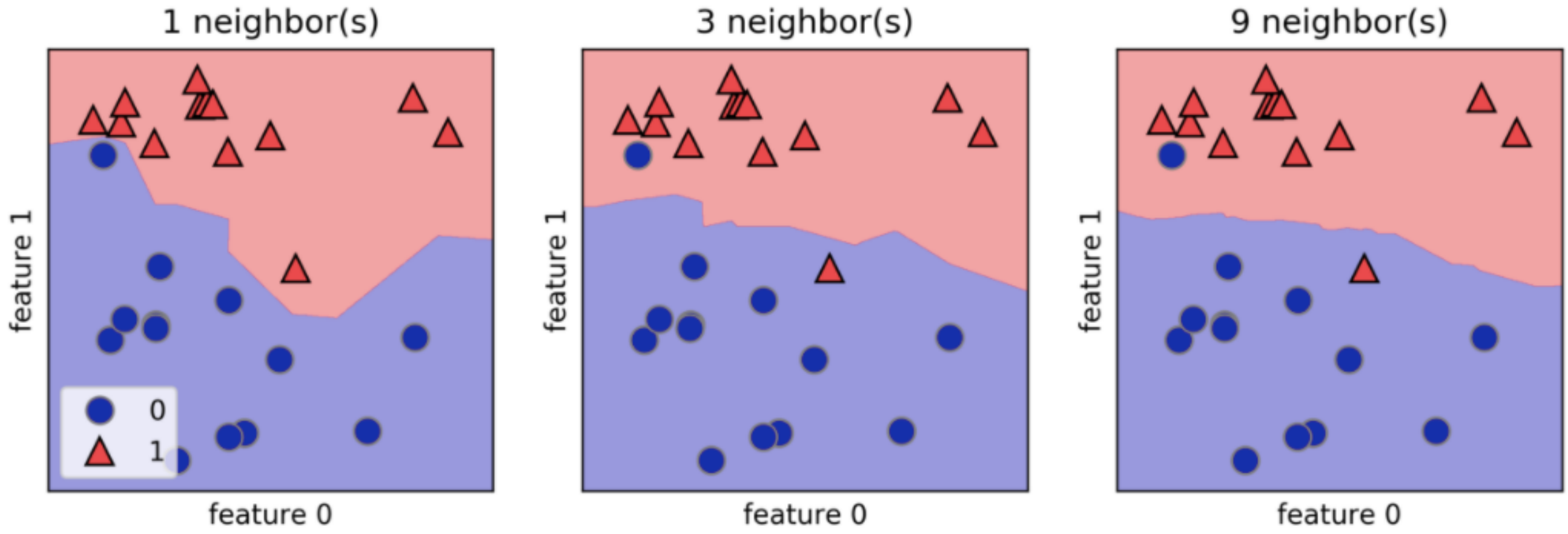
K-Nearest Neighbor Classification: What "K"?



What happens to the decision boundary as "k" grows?

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

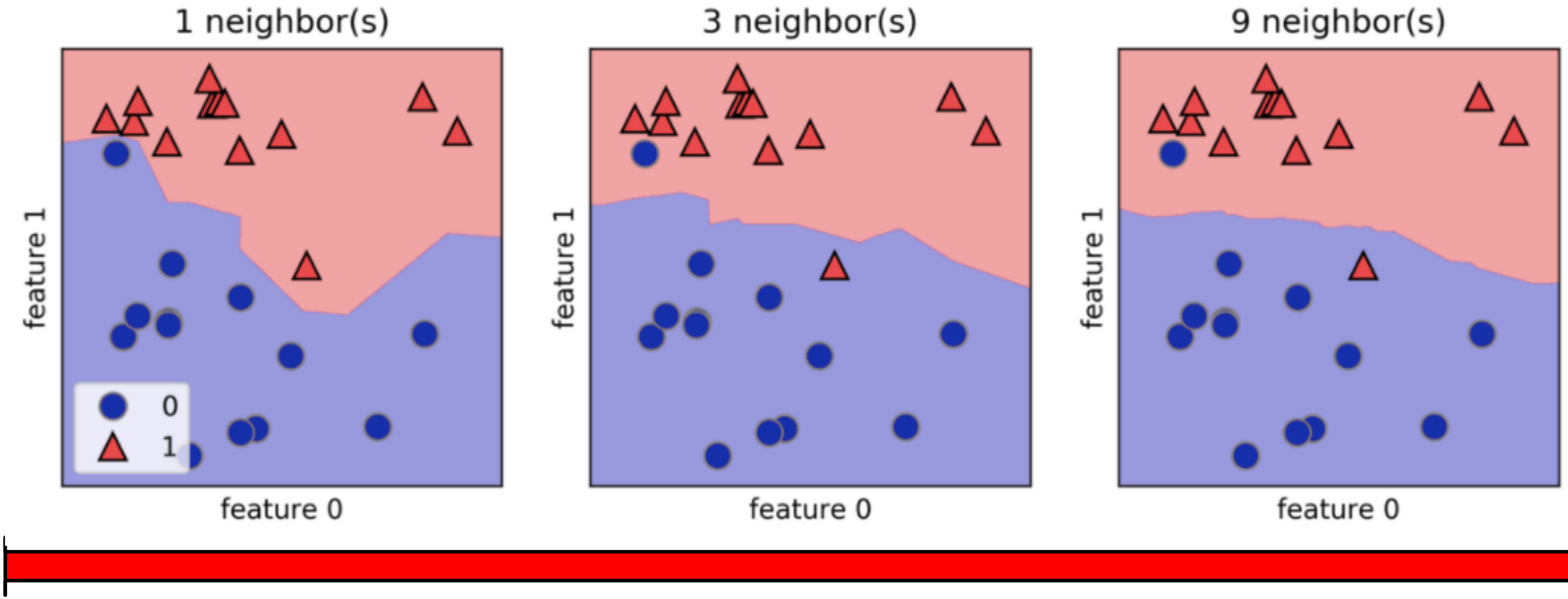
K-Nearest Neighbor Classification: What "K"?



What happens when "k" equals the training data size?

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What "K"?

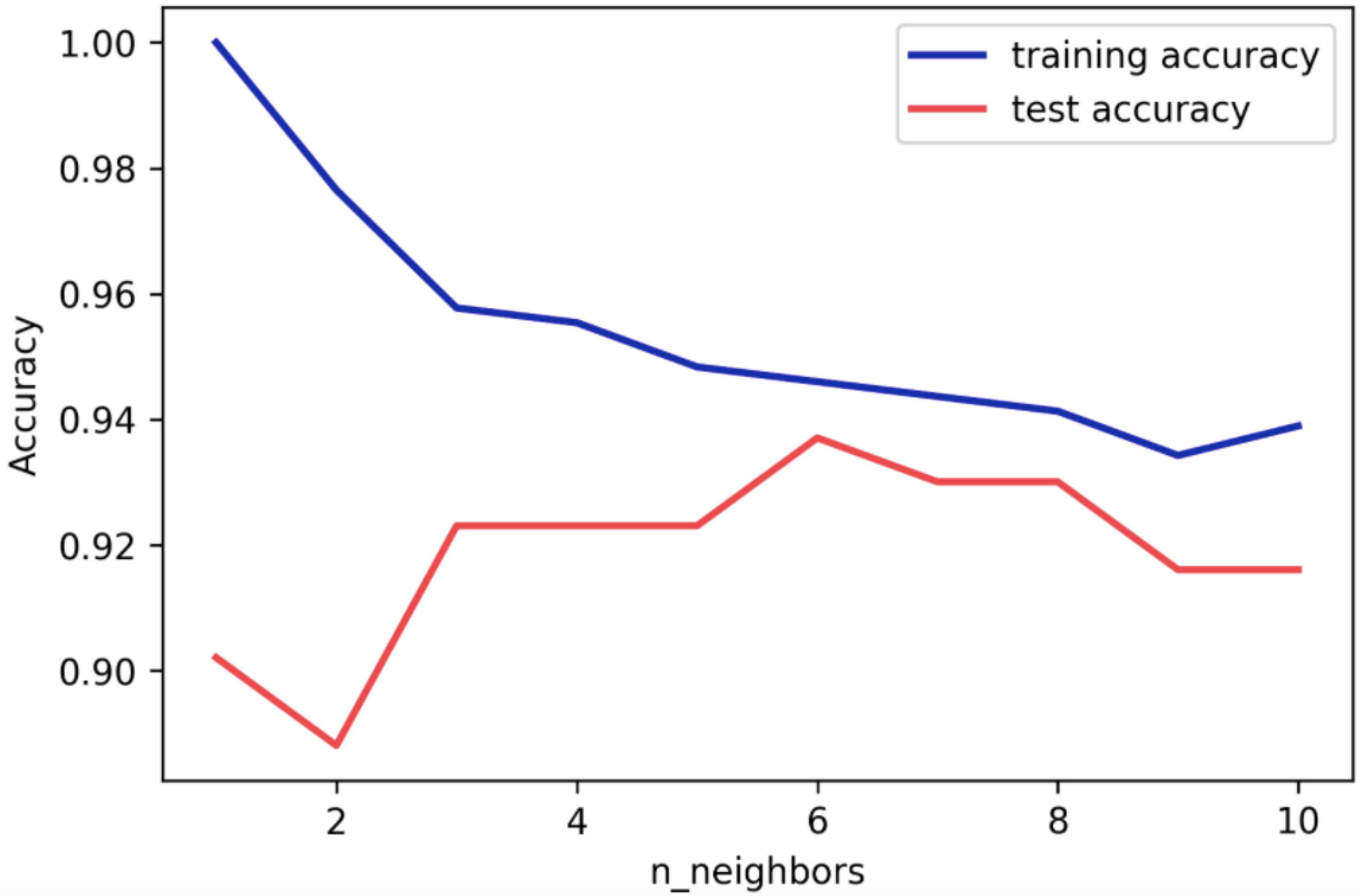


(Higher Model Complexity)

(Lower Model Complexity)

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What "K"?

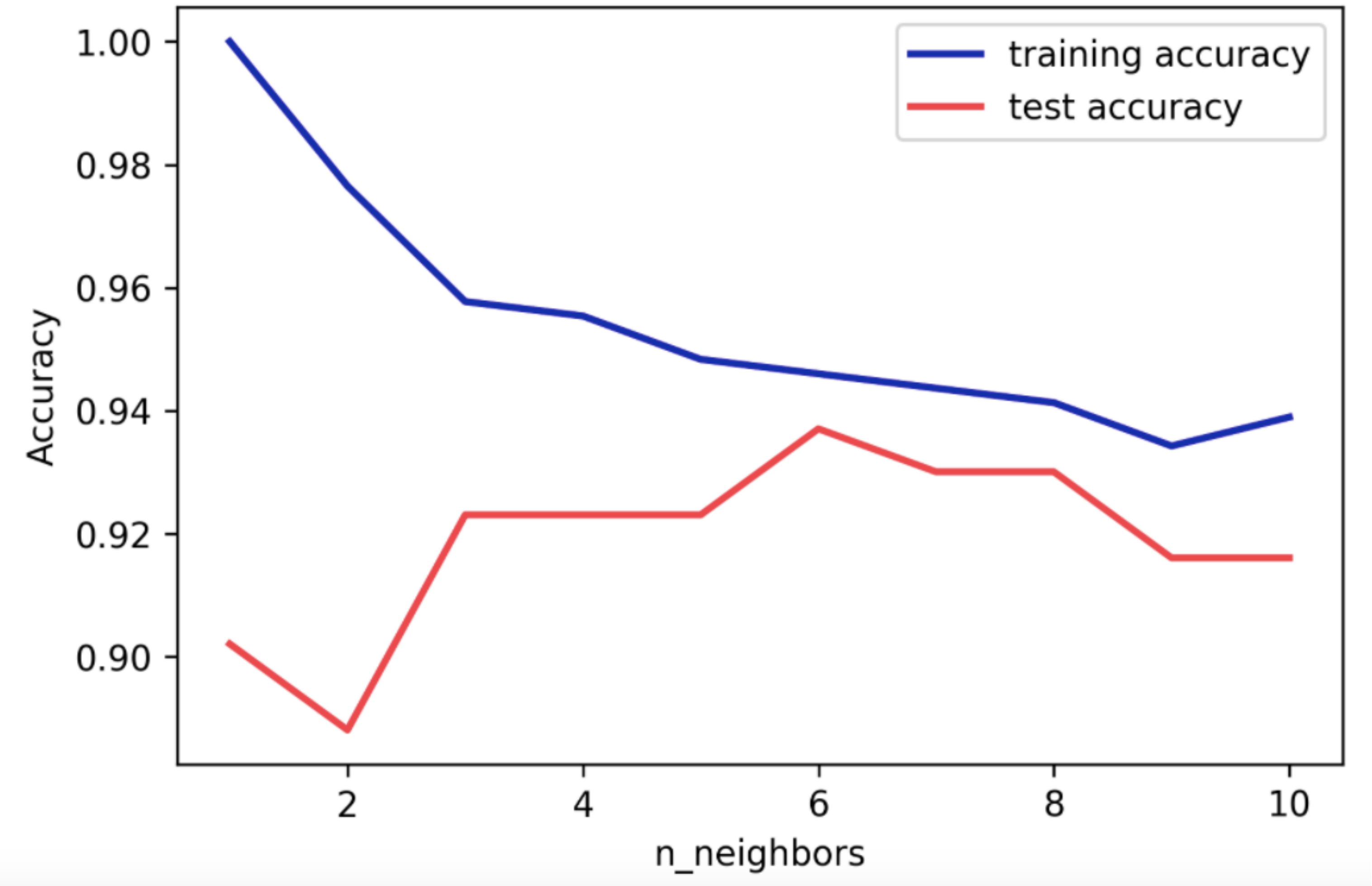


At what value for "k" is model **overfitting** the most?

k = 1

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?



What is the best value for “k”?

k = 6

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor: How to Use to Predict More than Two Classes?

- Tally number of examples belonging to each class and again choose the majority vote winners

What are Strengths of KNN?

- Adapts as new data is added
- Training is relatively fast
- Easy to understand

What are Weaknesses of KNN?

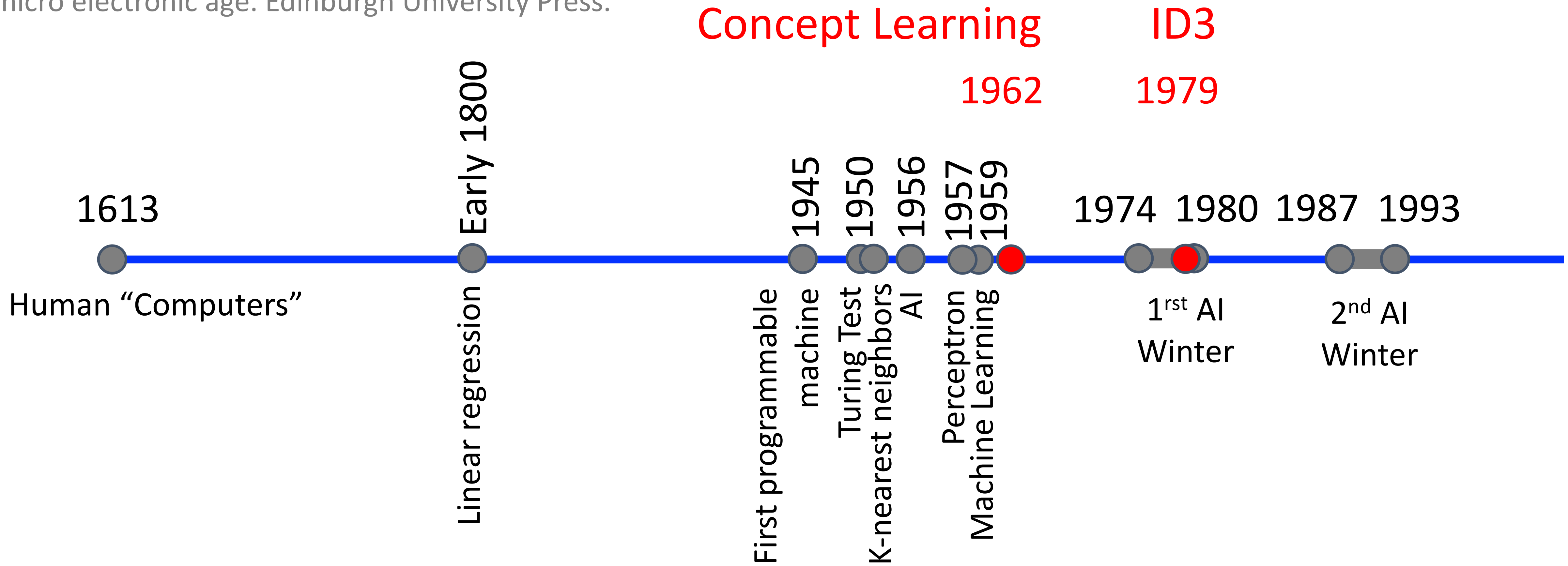
- For large datasets, requires large storage space
- For large datasets, this approach can be very slow or infeasible
  - Note: can improve speed with efficient data structures such as KD-trees
- Vulnerable to noisy/irrelevant examples
- Sensitive to imbalanced datasets where more frequent class will dominate majority voting

Today’s Topics

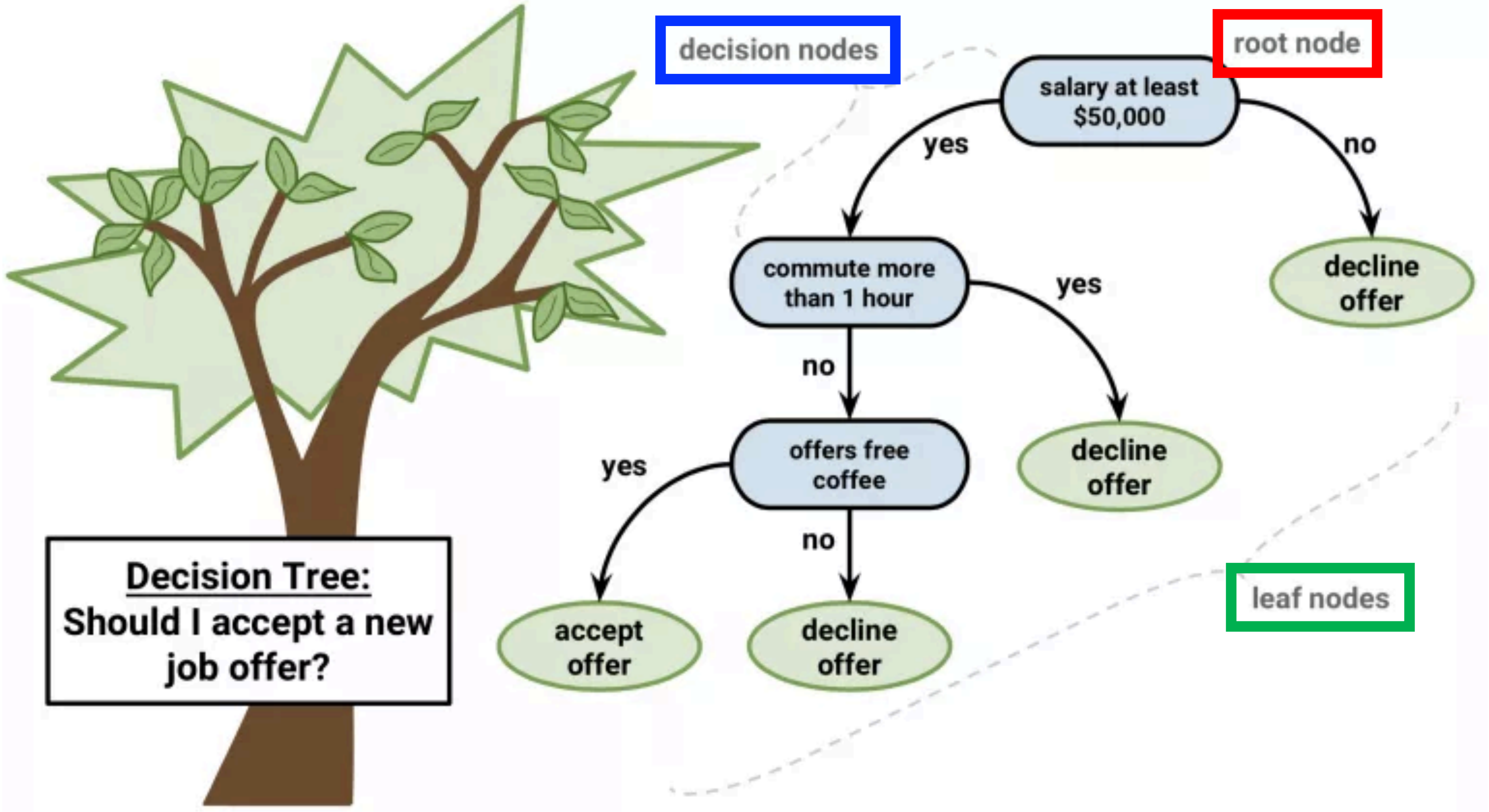
- Multiclass classification applications and evaluating models
- Motivation for new ML era: need non-linear models
- Nearest neighbor classification
- Decision tree classification
- Parametric versus non-parametric models

Historical Context of ML Models

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106.
Hunt, E.B. (1962). Concept learning: An information processing problem. New York: Wiley.
Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples. In D. Michie (Ed.), Expert systems in the micro electronic age. Edinburgh University Press.



Example: Decision Tree

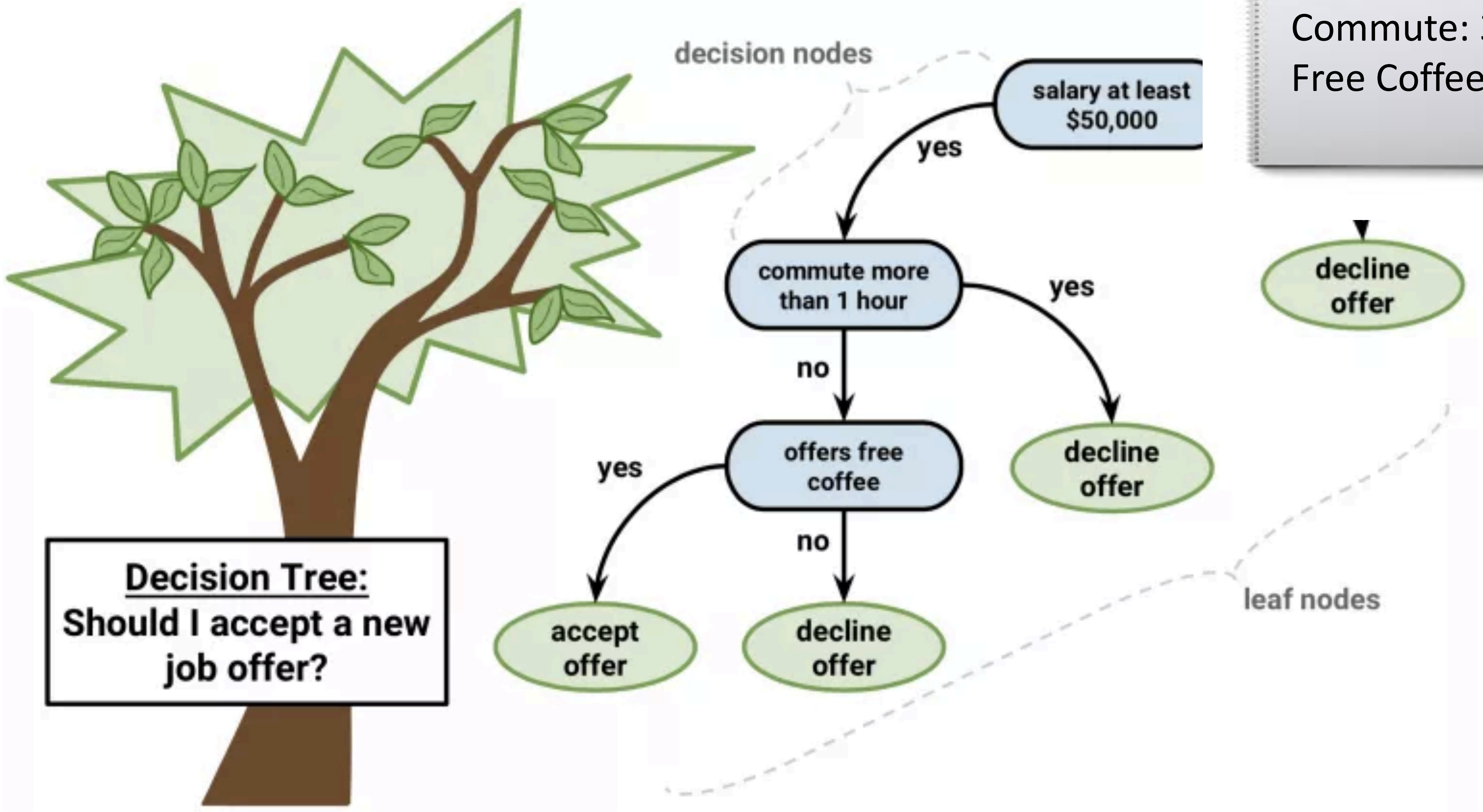


http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

Example: Decision Tree

Test Example

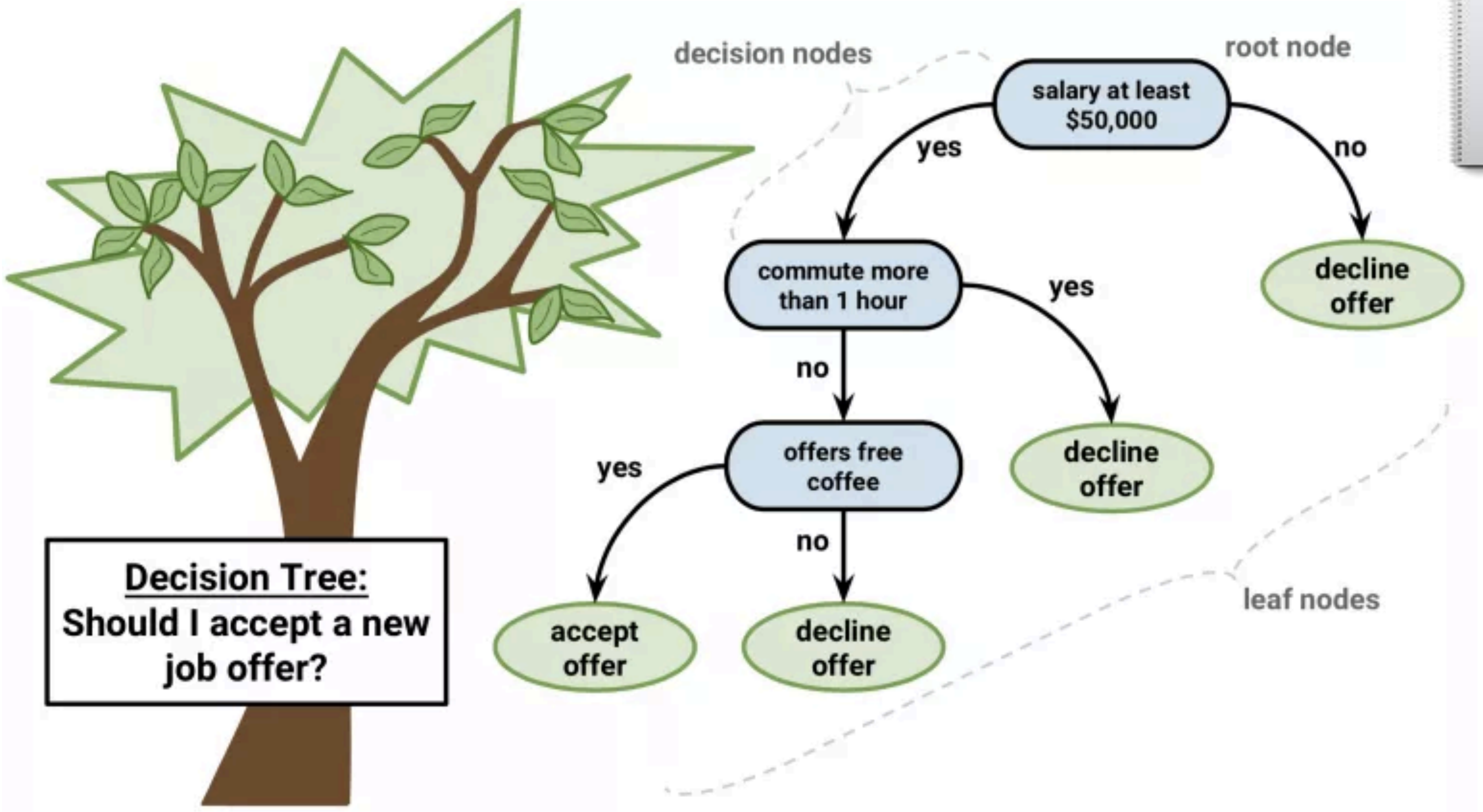
Salary: $44,869
Commute: 35 min
Free Coffee: Yes



http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

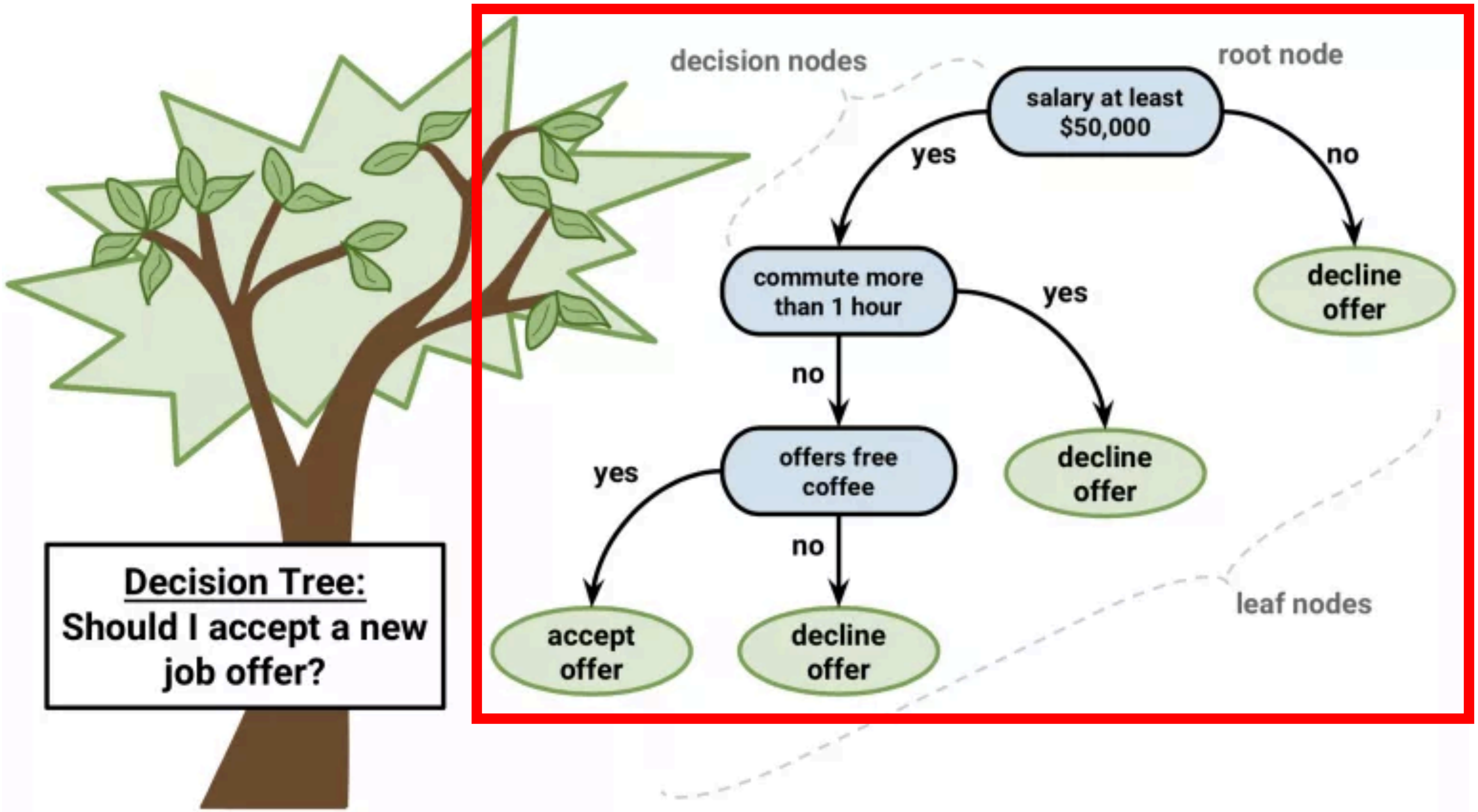
Test Example

Example: Decision Tree



http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

Example: Decision Tree

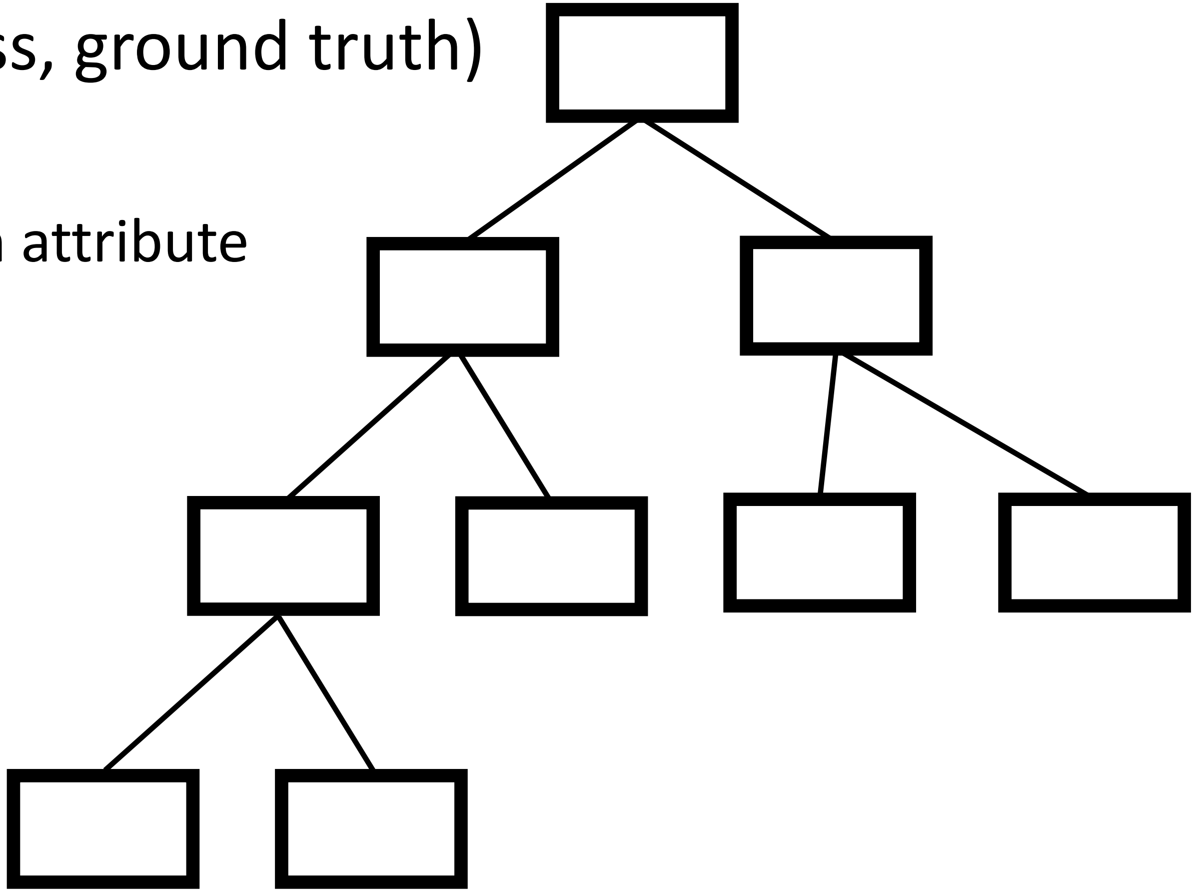


Machine must learn to create a decision tree

http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

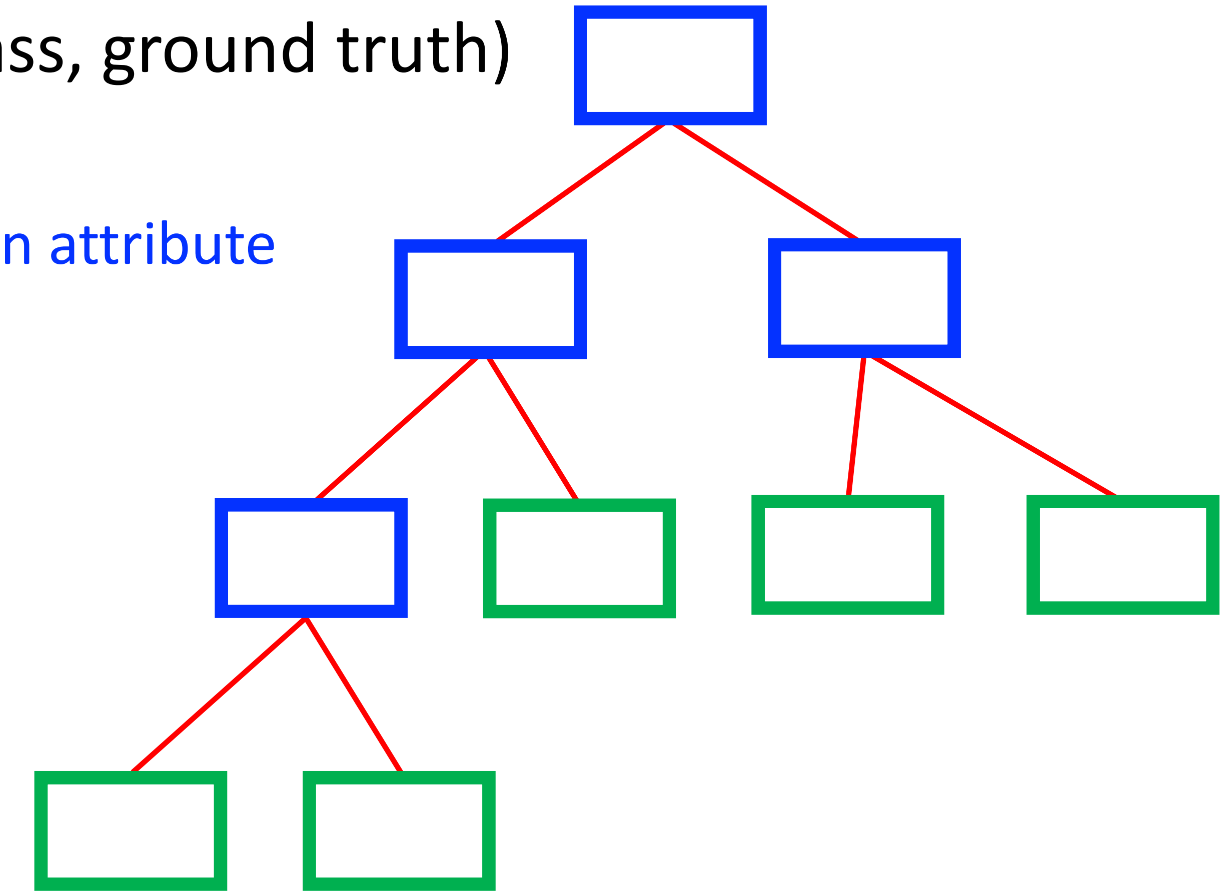
Decision Tree: Generic Structure

- Goal: predict class label (aka: class, ground truth)
- Representation: Tree
  - Internal (non-leaf) nodes = tests an attribute
  - Branches = attribute value
  - Leaf = classification label



Decision Tree: Generic Structure

- Goal: predict class label (aka: class, ground truth)
- Representation: Tree
  - Internal (non-leaf) nodes = tests an attribute
  - Branches = attribute value
  - Leaf = classification label



Decision Tree: Generic Structure

- Goal: predict class label
- Representation: Tree
  - Internal (non-leaf) nodes = tests an attribute
  - Branches = attribute value
  - Leaf = classification label

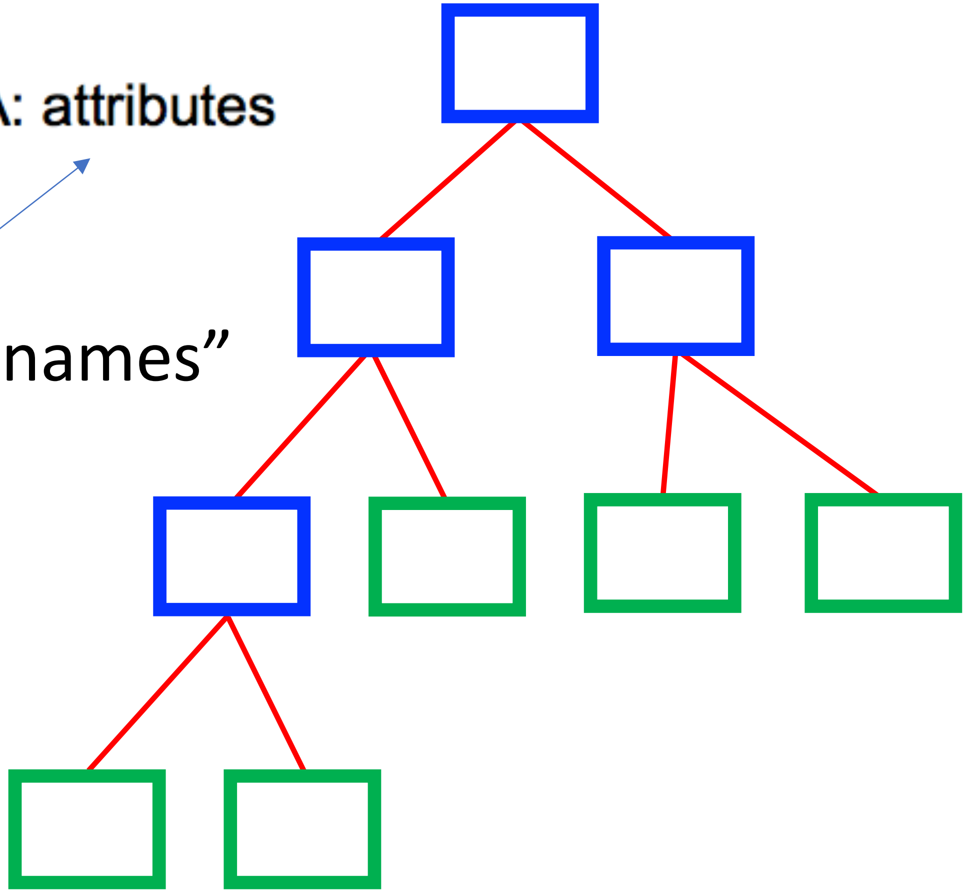
How can a machine learn a decision tree?

Decision Tree: Generic Learning Algorithm

- Greedy approach (NP complete problem)

Function BuildTree(n,A)   // n: samples (rows), A: attributes

aka – "data"   aka – "feature names"



Decision Tree: Generic Learning Algorithm

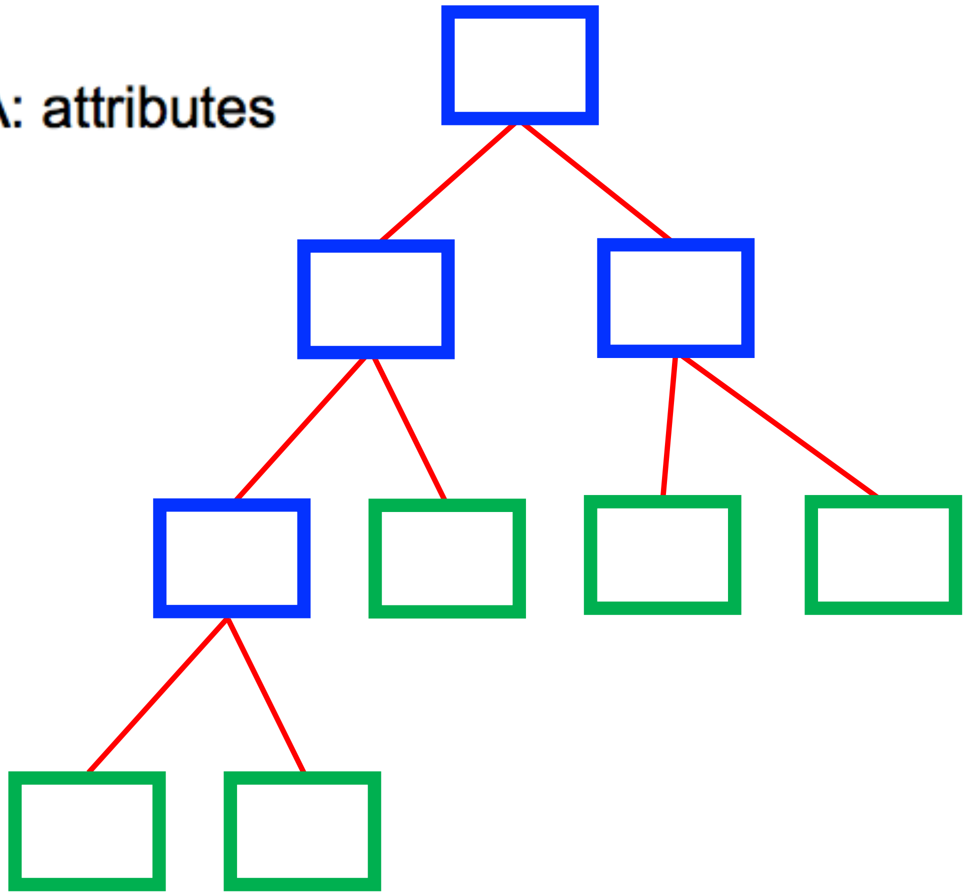
- Greedy approach (NP complete problem)

Function BuildTree(n,A) // n: samples (rows), A: attributes

If empty(A) or all n(L) are the same

status = leaf

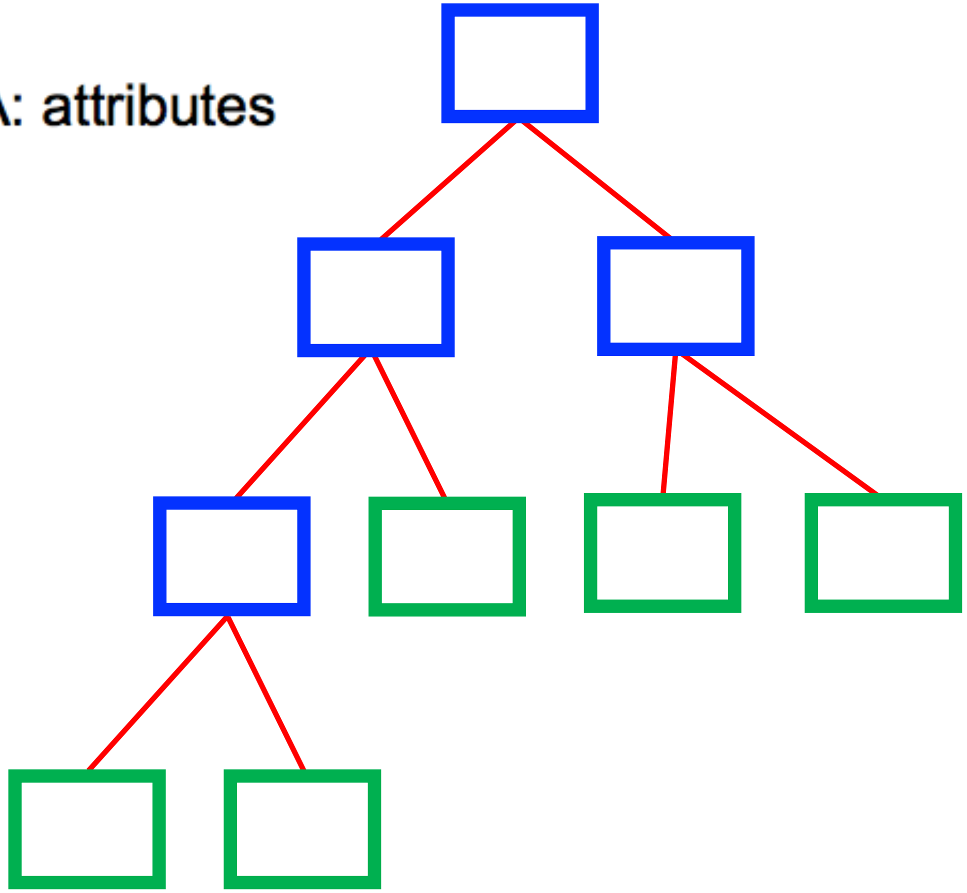
class = most common class in n(L)



Decision Tree: Generic Learning Algorithm

- Greedy approach (NP complete problem)

```
Function BuildTree(n,A)    // n: samples (rows), A: attributes
  If empty(A) or all n(L) are the same
     status = leaf
     class = most common class in n(L)
  else
     status = internal
     a ⇐ bestAttribute(n,A)    Key Decision
     LeftNode = BuildTree(n(a=1), A \ {a})
     RightNode = BuildTree(n(a=0), A \ {a})
  end
end
```



Next “Best” Attribute: Use Entropy

Number of classes

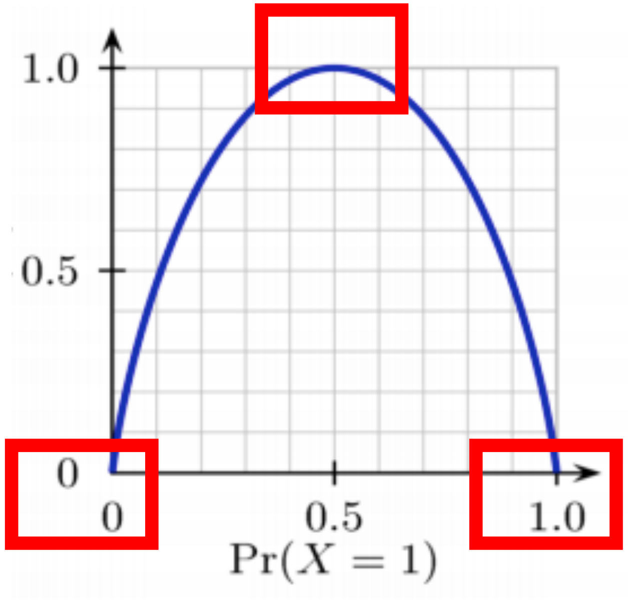
Encodes in bits

$$Entropy = -\sum_{i=1}^{\boxed{n}} \boxed{p_i} \log_{\boxed{2}} p_i$$

Fraction of examples belonging to class *i*

In a binary setting,

- Entropy is 0 when fraction of examples belonging to a class is 0 or 1
- Entropy is 1 when fraction of examples belonging to each class is 0.5



https://www.logcalculator.net/log-2

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Type	Length	IMDb Rating	Liked?
m1	Comedy	Short	7.2	$\boxed{\text{Yes}}$
m2	Drama	Medium	9.3	$\boxed{\text{Yes}}$
m3	Comedy	Medium	5.1	No
m4	Drama	Long	6.9	No
m5	Drama	Medium	8.3	$\boxed{\text{Yes}}$
m6	Drama	Short	4.5	No
m7	Comedy	Short	8.0	$\boxed{\text{Yes}}$
m8	Drama	Medium	7.5	$\boxed{\text{Yes}}$

- Let $\boxed{\text{C1 = "Yes"}}$ and C2 = "No"
- Current entropy?

$$Entropy = -\left(\boxed{\tfrac{5}{8}\log_2\tfrac{5}{8}} + \qquad\qquad \right)$$

Next “Best” Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

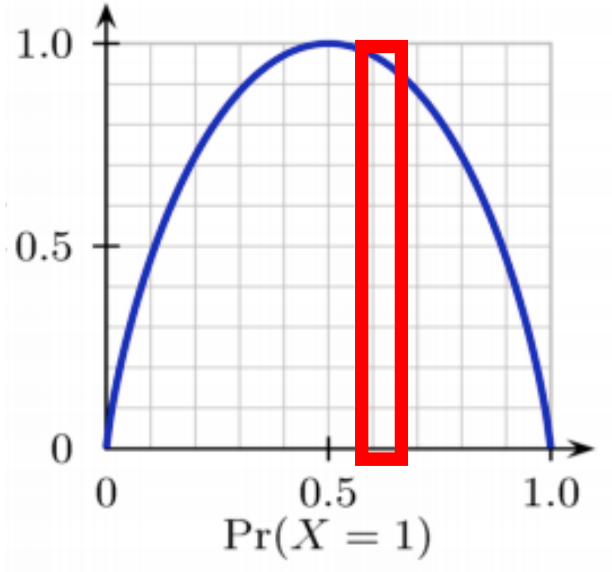
e.g., Will you like a movie?

Movie	Type	Length	IMDb Rating	Liked?
m1	Comedy	Short	7.2	Yes
m2	Drama	Medium	9.3	Yes
m3	Comedy	Medium	5.1	No
m4	Drama	Long	6.9	No
m5	Drama	Medium	8.3	Yes
m6	Drama	Short	4.5	No
m7	Comedy	Short	8.0	Yes
m8	Drama	Medium	7.5	Yes

- Let C1 = “Yes” and C2 = “No”
- Current entropy?

$$Entropy = -(\frac{5}{8}\log_2\frac{5}{8} + \boxed{\frac{3}{8}\log_2\frac{3}{8}})$$

Next “Best” Attribute: Use Entropy



$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Type	Length	IMDb Rating	Liked?
m1	Comedy	Short	7.2	Yes
m2	Drama	Medium	9.3	Yes
m3	Comedy	Medium	5.1	No
m4	Drama	Long	6.9	No
m5	Drama	Medium	8.3	Yes
m6	Drama	Short	4.5	No
m7	Comedy	Short	8.0	Yes
m8	Drama	Medium	7.5	Yes

- Let C1 = “Yes” and C2 = “No”
- Current entropy?

$$Entropy = -(\tfrac{5}{8}\log_2\tfrac{5}{8} + \tfrac{3}{8}\log_2\tfrac{3}{8})$$

$$Entropy = -(-0.42 - 0.53) = \boxed{0.95}$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Type"?
  - Left tree: "Comedy" = ?

$$Entropy = -(\tfrac{2}{3}\log_2\tfrac{2}{3} + \tfrac{1}{3}\log_2\tfrac{1}{3})$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Type"?
  - Left tree: "Comedy" = ?

$$Entropy = -(\tfrac{2}{3}\log_2 \tfrac{2}{3} + \tfrac{1}{3}\log_2 \tfrac{1}{3})$$

$$Entropy = -(-0.53 - 0.39) = 0.92$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	$\boxed{\text{Drama}}$	$\boxed{\text{Yes}}$
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	$\boxed{\text{Drama}}$	$\boxed{\text{Yes}}$

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Type"?
  - Left tree: "Comedy" = 0.92
  - Right tree: "Drama" = ?

$$Entropy = -(\tfrac{3}{5}\log_2\tfrac{3}{5} + \tfrac{2}{5}\log_2\tfrac{2}{5})$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Type"?
  - Left tree: "Comedy" = 0.92
  - Right tree: "Drama" = ?

$$Entropy = -(\tfrac{3}{5}\log_2\tfrac{3}{5} + \tfrac{2}{5}\log_2\tfrac{2}{5})$$

$$Entropy = -(-0.44 - 0.53) = 0.97$$

Next “Best” Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “Type”?
  - Left tree: “Comedy” = 0.92
  - Right tree: “Drama” = 0.97
- Information gain by split on “Type”?

$$IG = 0.95 - (\tfrac{3}{8} * 0.92 + \tfrac{5}{8} * 0.97)$$
$$IG = 0$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Length"?
  - Left tree: "Short" = ?

$$Entropy = -(\tfrac{2}{3}\log_2 \tfrac{2}{3} + \tfrac{1}{3}\log_2 \tfrac{1}{3})$$

$$Entropy = -(-0.53 - 0.39) = 0.92$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Length"?
  - Left tree: "Short" = 0.92
  - Middle tree: "Medium" = ?

$$Entropy = -(\tfrac{3}{4}\log_2\tfrac{3}{4} + \tfrac{1}{4}\log_2\tfrac{1}{4})$$

$$Entropy = -(-0.32 - 0.5) = 0.82$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Length"?
  - Left tree: "Short" = 0.92
  - Middle tree: "Medium" = 0.82
  - Right tree: "Long" = ?

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Length"?
  - Left tree: "Short" = 0.92
  - Middle tree: "Medium" = 0.82
  - Right tree: "Long" = 0
- Information gain by split on "Length"?

$$IG = 0.95 - (\tfrac{3}{8} * 0.92 + \tfrac{4}{8} * 0.82 + \tfrac{1}{8} * 0)$$
$$IG = 0.19$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

$$IG = 0.95 - \left(\tfrac{5}{8} * \left(\tfrac{5}{5}\log_2\tfrac{5}{5}\right) + \tfrac{3}{8} * \left(\tfrac{3}{3}\log_2\tfrac{3}{3}\right)\right)$$
$$IG = 0.95$$

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next "Best" Attribute: Use Entropy

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

    • • •

Split at midpoint and measure entropy

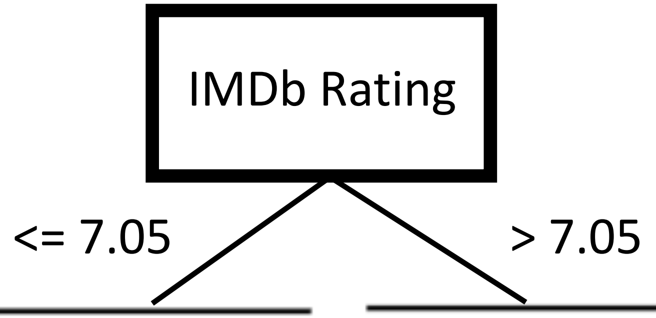
Decision Tree: What is Our First Split?

- Greedy approach (NP complete problem)

```
Function BuildTree(n,A)    // n: samples (rows), A: attributes
  If empty(A) or all n(L) are the same
     status = leaf
     class = most common class in n(L)
  else
     status = internal
     a ⇐ bestAttribute(n,A)
     LeftNode = BuildTree(n(a=1), A
     RightNode = BuildTree(n(a=0),
   end
end
```

		IG = 0	IG = 0.19	IG = 0.95	
	Movie	Type	Length	IMDb Rating	Liked?
	m1	Comedy	Short	7.2	Yes
	m2	Drama	Medium	9.3	Yes
	m3	Comedy	Medium	5.1	No
	m4	Drama	Long	6.9	No
	m5	Drama	Medium	8.3	Yes
	m6	Drama	Short	4.5	No
	m7	Comedy	Short	8.0	Yes
	m8	Drama	Medium	7.5	Yes

Decision Tree: What Tree Results?

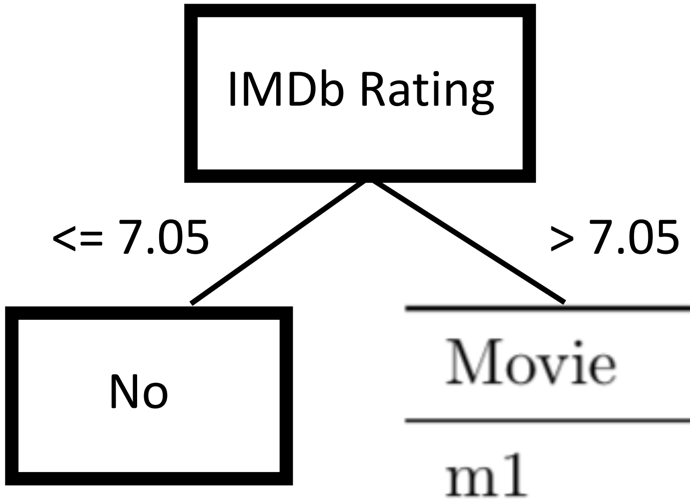


Movie	Type	Length	Liked?
m3	Comedy	Medium	No
m4	Drama	Long	No
m6	Drama	Short	No

Recurse on this tree?

Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m5	Drama	Medium	Yes
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

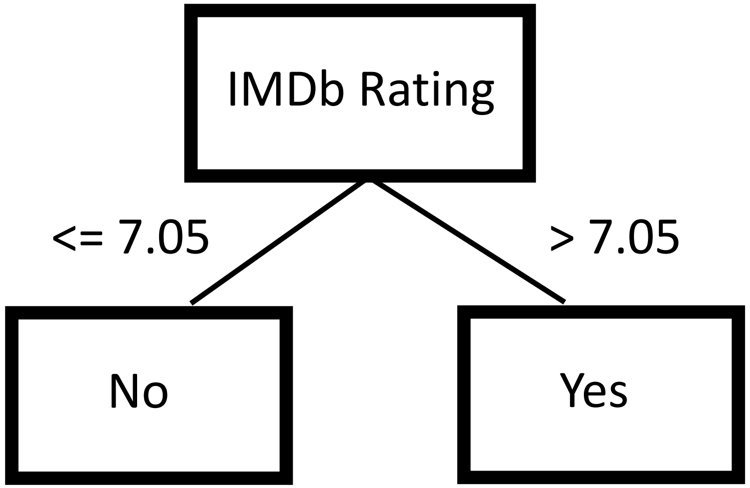
Decision Tree: What Tree Results?



Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m5	Drama	Medium	Yes
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

Recurse on this tree?

Decision Tree: What Tree Results?



Decision Tree: Generic Learning Algorithm

- Greedy approach (NP complete problem)

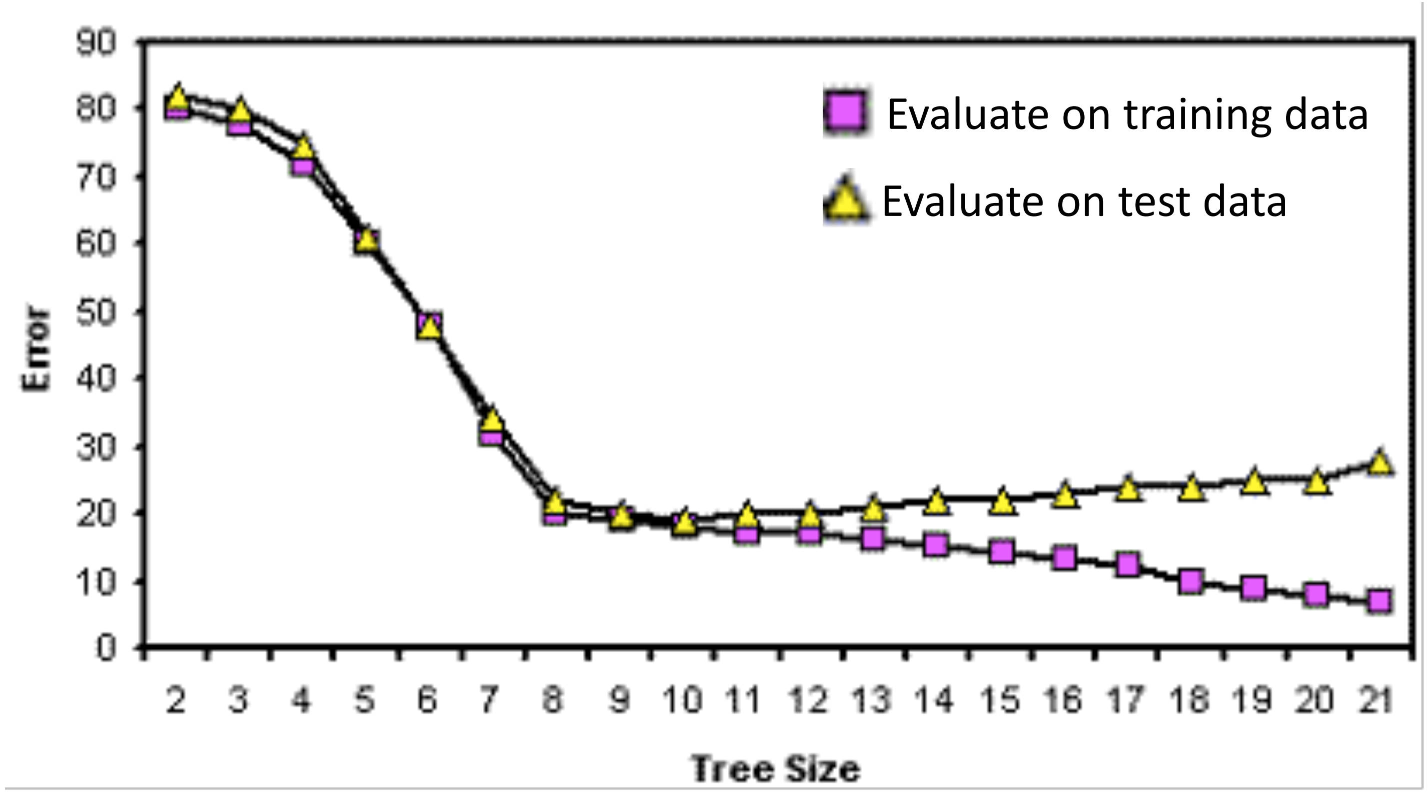
```
Function BuildTree(n,A)    // n: samples (rows), A: attributes
  If empty(A) or all n(L) are the same
    status = leaf
    class = most common class in n(L)
  else
    status = internal
    a ⇐ bestAttribute(n,A)
    LeftNode = BuildTree(n(a=1), A \ {a})
    RightNode = BuildTree(n(a=0), A \ {a})
  end
end
```

Key Decision

- Entropy (maximize information gain)
- Gini Index (used in CART algorithm)
- Gain ratio (used in C4.5 algorithm)
- Mean squared error
- …

Overfitting

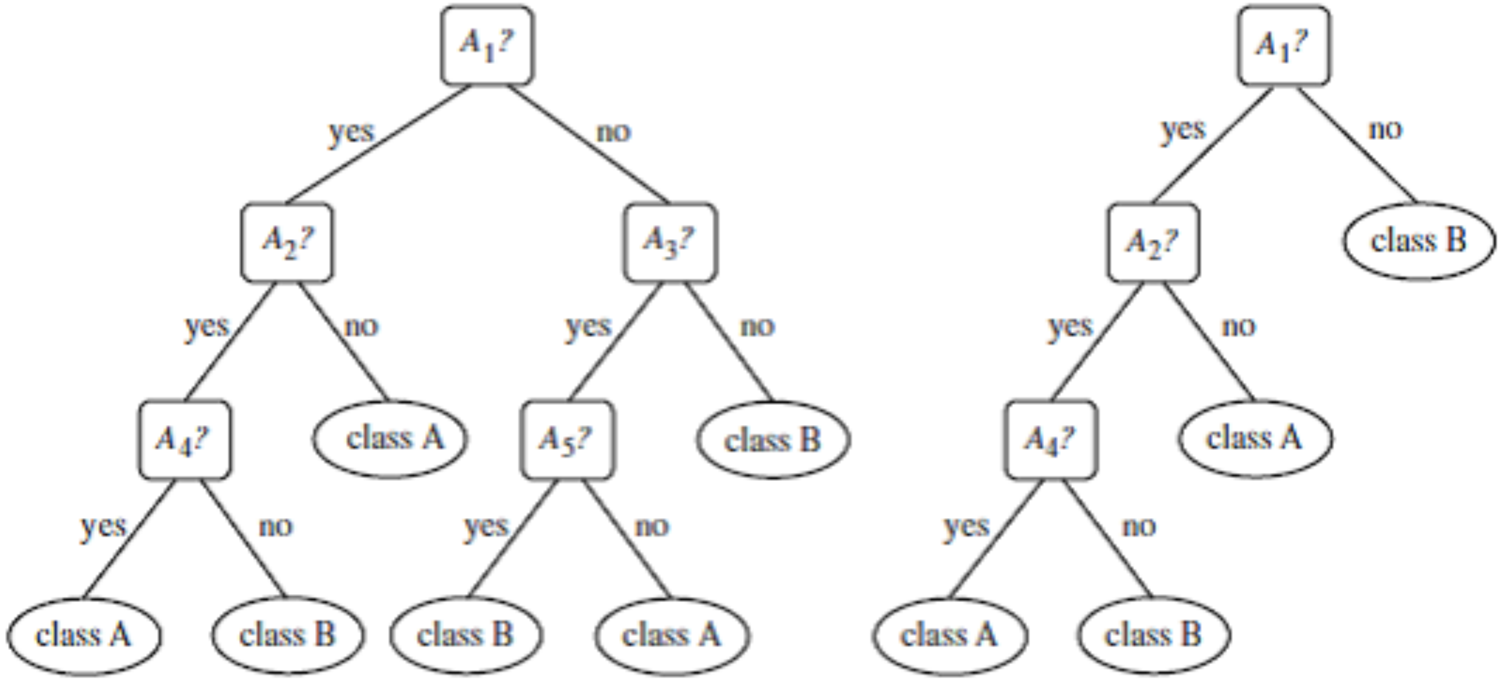
- At what tree size, does overfitting begin?



http://www.alanfielding.co.uk/multivar/crt/dt_example_04.htm

Regularization to Avoid Overfitting

- Pruning
  - Pre-pruning: stop tree growth earlier
  - Post-pruning: prune tree afterwards



https://www.clariba.com/blog/tech-20140811-decision-trees-with-sap-predictive-analytics-and-sap-hana-emilio-nieto

Today’s Topics

- Multiclass classification applications and evaluating models
- Motivation for new ML era: need non-linear models
- Nearest neighbor classification
- Decision tree classification
- Parametric versus non-parametric models

Machine Learning Goal

- Learn function that maps input features (X) to an output prediction (Y)

$$Y = f(x)$$

Machine Learning Goal

- Learn function that maps input features (X) to an output prediction (Y)

$$Y = f(x)$$

- Parametric model: has a fixed number of parameters

Machine Learning Goal

- Learn function that maps input features (X) to an output prediction (Y)

$$Y = f(x)$$

- Parametric model: has a fixed number of parameters
- Non-parametric model: does not specify the number of parameters

Class Discussion

- For each model, is it parametric or non-parametric?
  - Linear regression
  - K-nearest neighbors

- What are advantages and disadvantages of parametric models?

- What are advantages and disadvantages of non-parametric models?

Today’s Topics

- Multiclass classification applications and evaluating models
- Motivation for new ML era: need non-linear models
- Nearest neighbor classification
- Decision tree classification
- Parametric versus non-parametric models

References Used for Today's Material

- Chapter 3 of Deep Learning book by Goodfellow et al.
- http://www.cs.utoronto.ca/~fidler/teaching/2015/slides/CSC411/06_trees.pdf
- http://www.cs.utoronto.ca/~fidler/teaching/2015/slides/CSC411/tutorial3_CrossVal-DTs.pdf
- http://www.cs.cmu.edu/~epxing/Class/10701/slides/classification15.pdf