CS 383: Machine Learning

Prof Adam Poliak

Fall 2024

09/10/2024

Lecture 02

Outline

- Reading Quiz 1
- K-nearest neighbors
- Featurization (intro)
- Entropy & Decision Trees

Outline

1) *Generalization*: ability to answer new questions related to the topic studied

2) No! If we look at the test data (either the *features* or the *labels*), then any measurement of the performance of our algorithm becomes inaccurate

3) *Multiclass classification*

4) *Regression*

Outline

- Reading Quiz 1
- K-nearest neighbors
- Featurization (intro)
- Entropy & Decision Trees

Typical Supervised Learning setup

- X: Input matrix
  - n rows (examples/instances/individuals)
  - m columns (features/attributes)
- y: Label vector
- $\hat{y}$: Prediction vector

Simple Evaluation

How close is $\hat{y}$ to y?

Indicator variable

Accuracy: $\frac{1}{n}\sum_i \mathbb{1}(\hat{y}_i, y_i)$

K-nearest neighbors

Problem: Given a collection of labeled examples, determine the class of a new unlabeled example

Solution: assign the same class as the ***most similar*** examples

Key idea: whats ***most similar***?

K-NN creates implicit decision boundaries

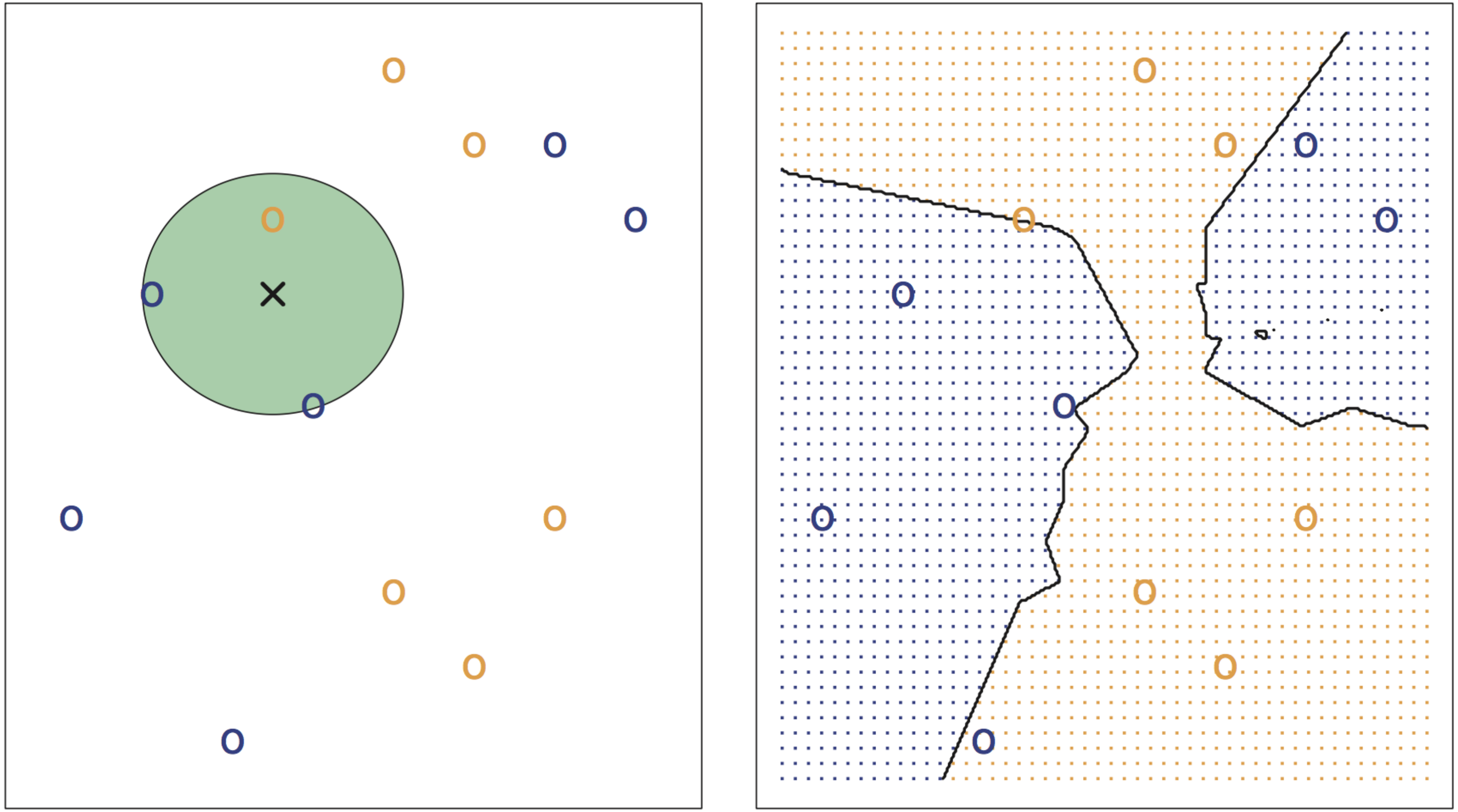


Figure 2.14 from ISL book, KNN with two classes (C=2), and K=3

HW01 – K-Nearest Neighbor

Implement KNN

Given:

- starter code, unit tests
- Train data with labels, dev data with labels, test data w/o labels

Deliverables:

- Implemented code
- Predictions on test data

Outline

- Reading Quiz 1
- K-nearest neighbors
- Featurization (intro)
- Entropy & Decision Trees

Terminology

Features: feature names

- e.g. shape

Feature values: what values are possible

- e.g. {circle, square, triangle}

Feature vector: values for a particular example

- e.g. $\boldsymbol{x} = [x_1, x_2, x_3, \ldots, x_p]$

Terminology

Decision boundary:

separates regions of the feature space that would be classified as positive or negative (or multiclass)

Underfitting:

“had the opportunity to learn something but didn’t” (Duame)

Overfitting:

memorized individual training examples (fit to noise) and can’t generalize

Handout 2
(find and work with a partner)

Comparing Decision Boundaries

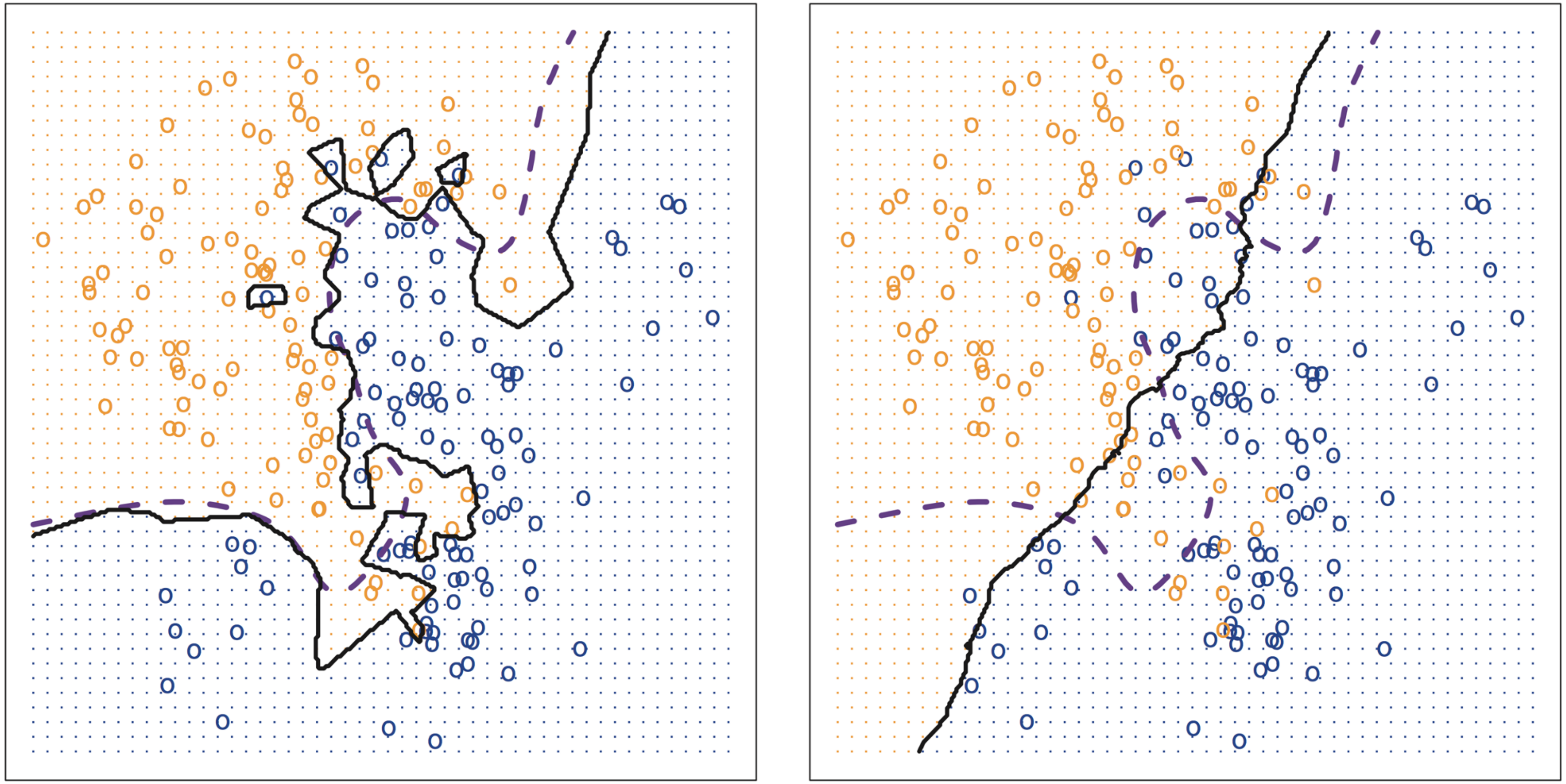


Figure 2.16 from ISL book (dashed line is “ideal” boundary)

Comparing Decision Boundaries

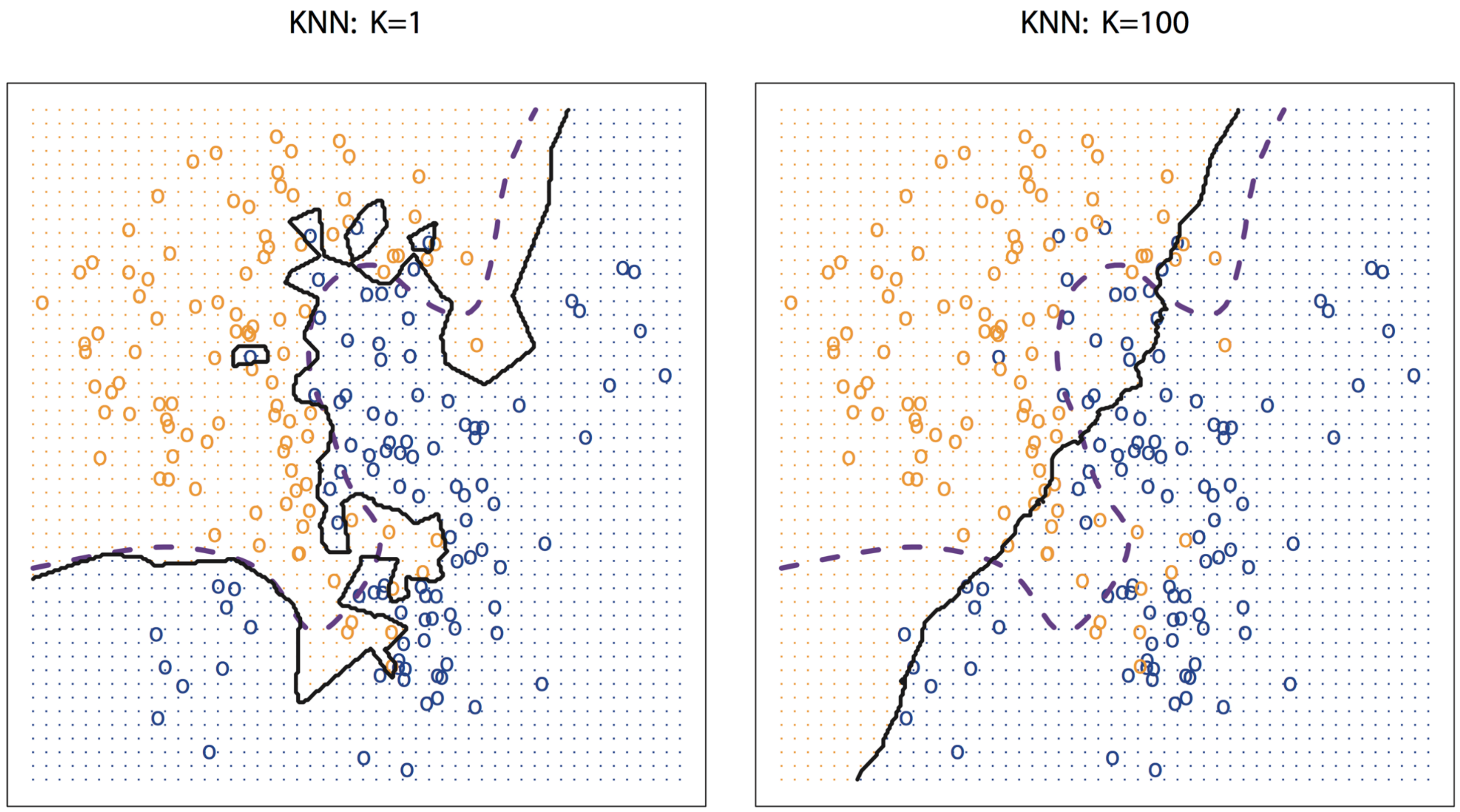


Figure 2.16 from ISL book (dashed line is “ideal” boundary)

Comparing Decision Boundaries

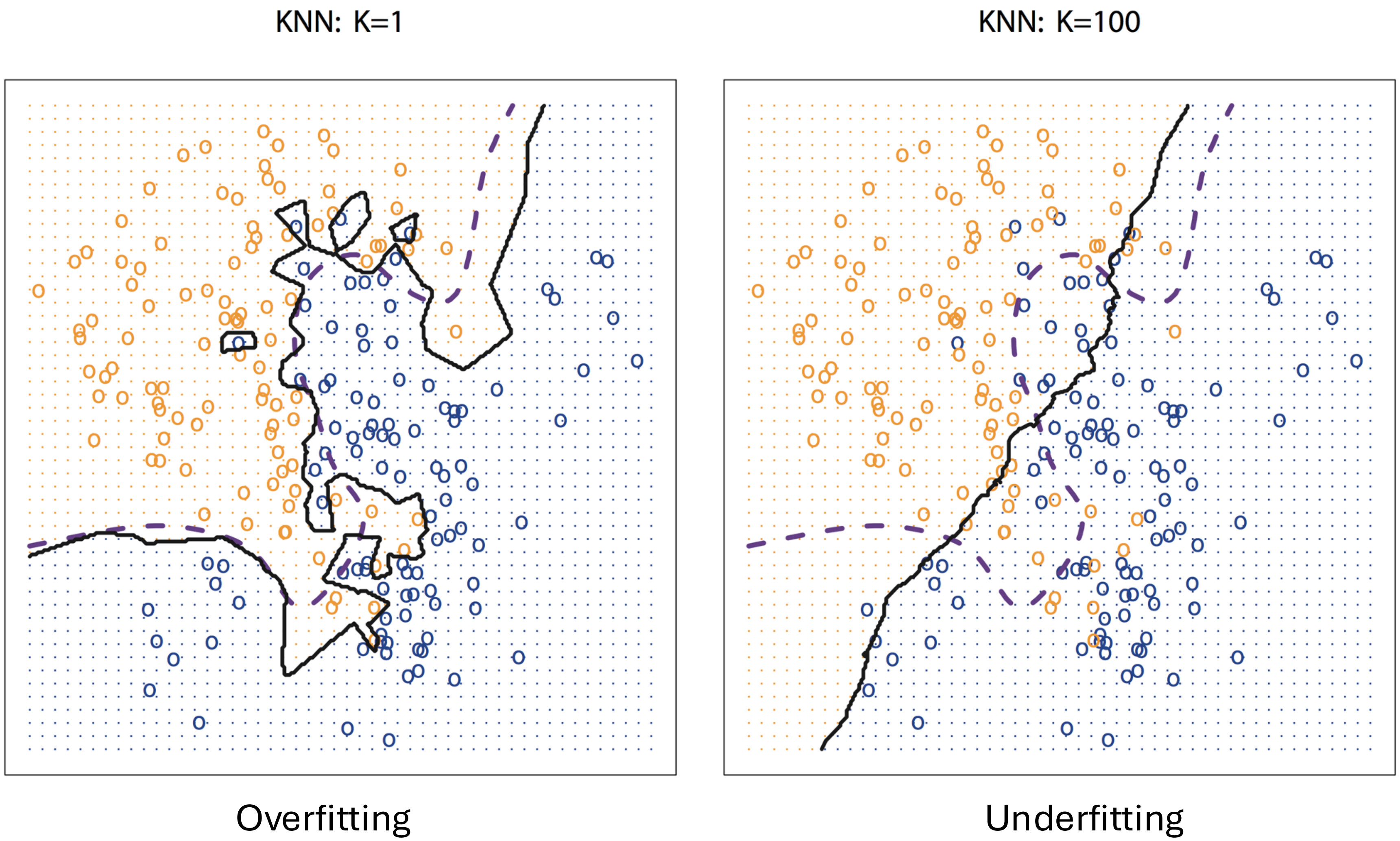


Figure 2.16 from ISL book (dashed line is “ideal” boundary)

Featurization (rule of thumb)

Duame, Chap 3

- Real-valued features get copied directly.
- Binary features become 0 (for false) or 1 (for true).
- Categorical features with V possible values get mapped to V-many binary indicator features.

Haven't discussed:

-normalization

-categorical variables on a spectrum

Handout

4. Using your response from the previous question, what would the *feature vector* become for $\boldsymbol{x}_1$?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes

Data from Machine Learning by Tom Mitchell (Table 3.2)

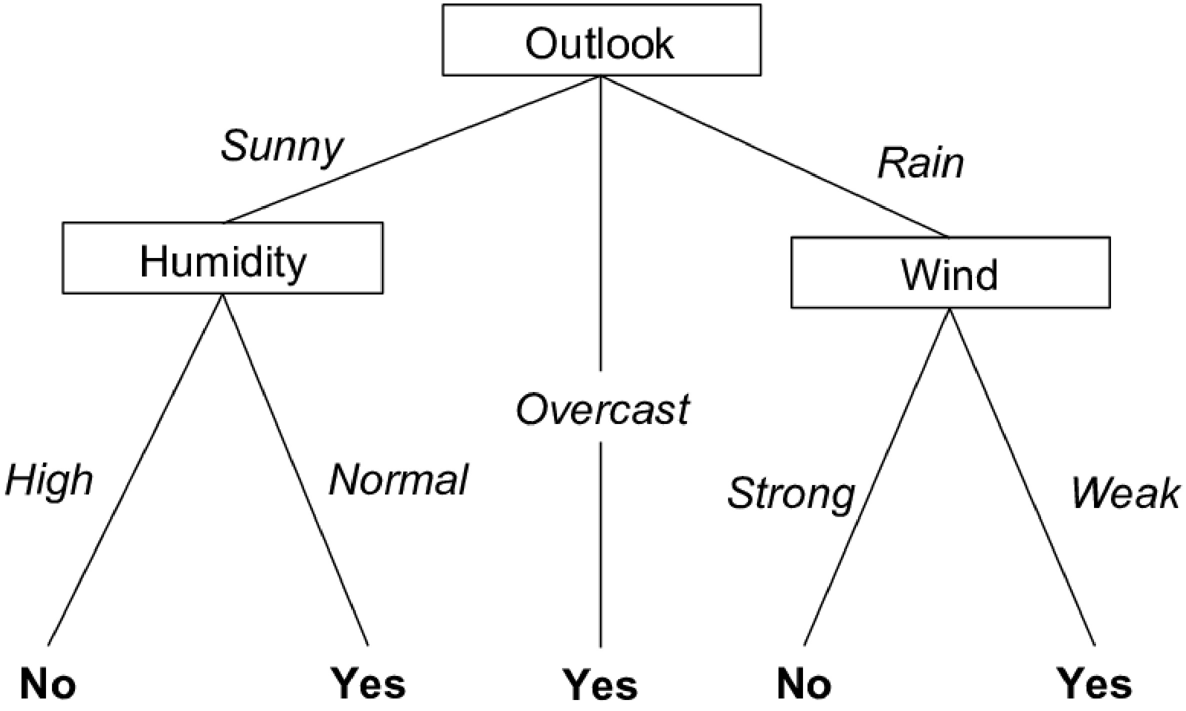
Sunny:	{0,1}	
Overcast:	{0,1}	
Rain:	{0,1}	
Temperature:	{0, 1, 2}	(Cool, Mild, Hot)
Humidity:	{0,1}	(Normal, High)
Wind	{0,1}	(Weak, Strong)

	Sunny	Overcast	Rain	Temp	Humidity	Wind
$\boldsymbol{x}_1$	1	0	0	2	1	0

Outline

- Reading Quiz 1
- K-nearest neighbors
- Featurization (intro)
- Entropy & Decision Trees

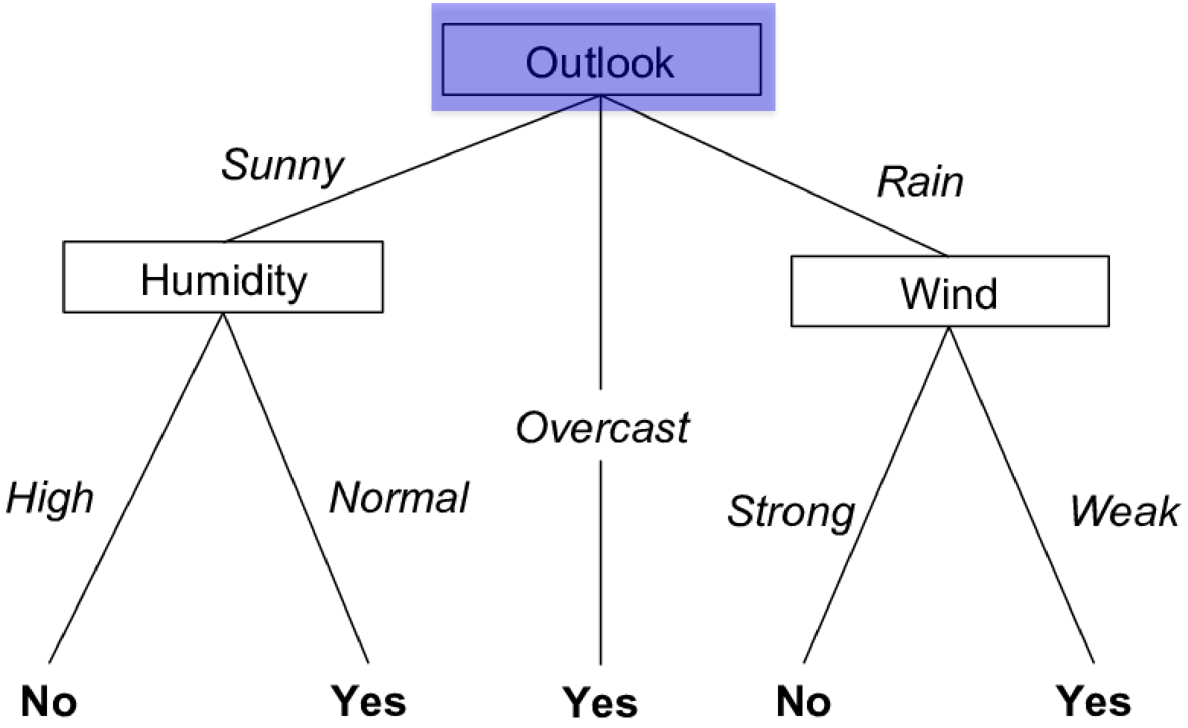
Decision Tree example (tennis)



Each internal node: test one feature
Each branch from node: selects one value of the feature
Each leaf node: predict *y*

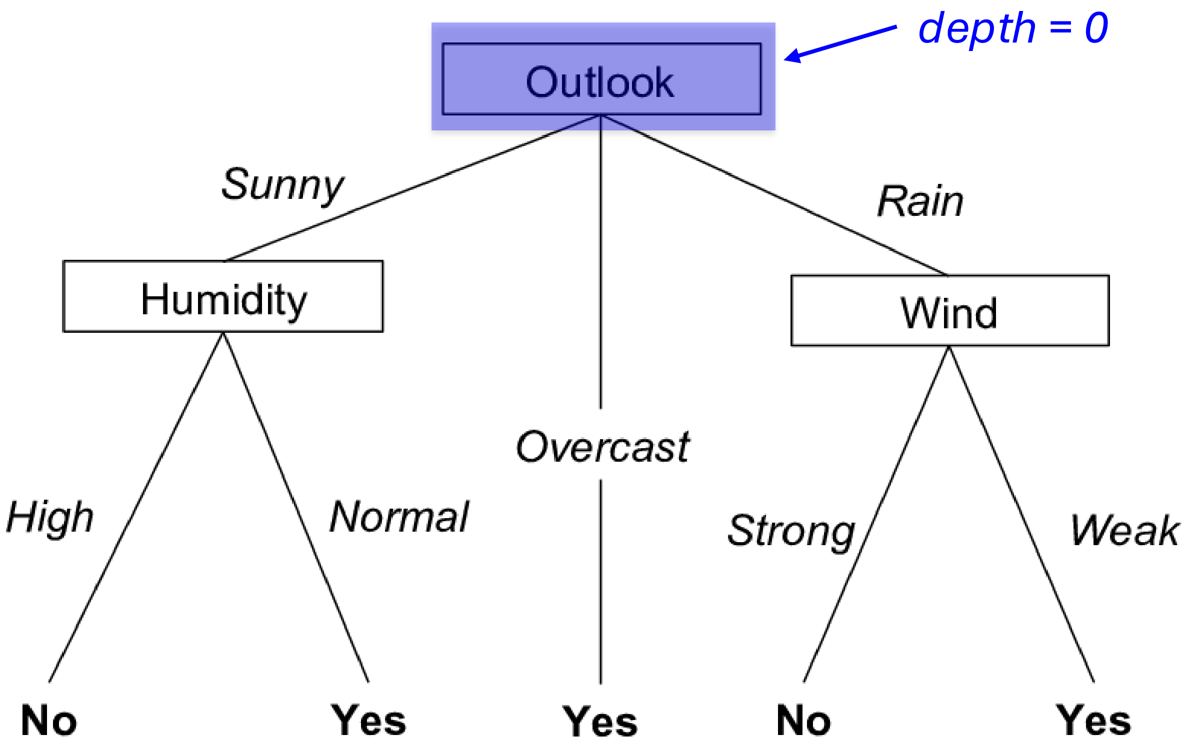
Based on slides by Jessica Wu and Eric Eaton [originally by Tom Mitchell]

Decision Tree example (tennis)



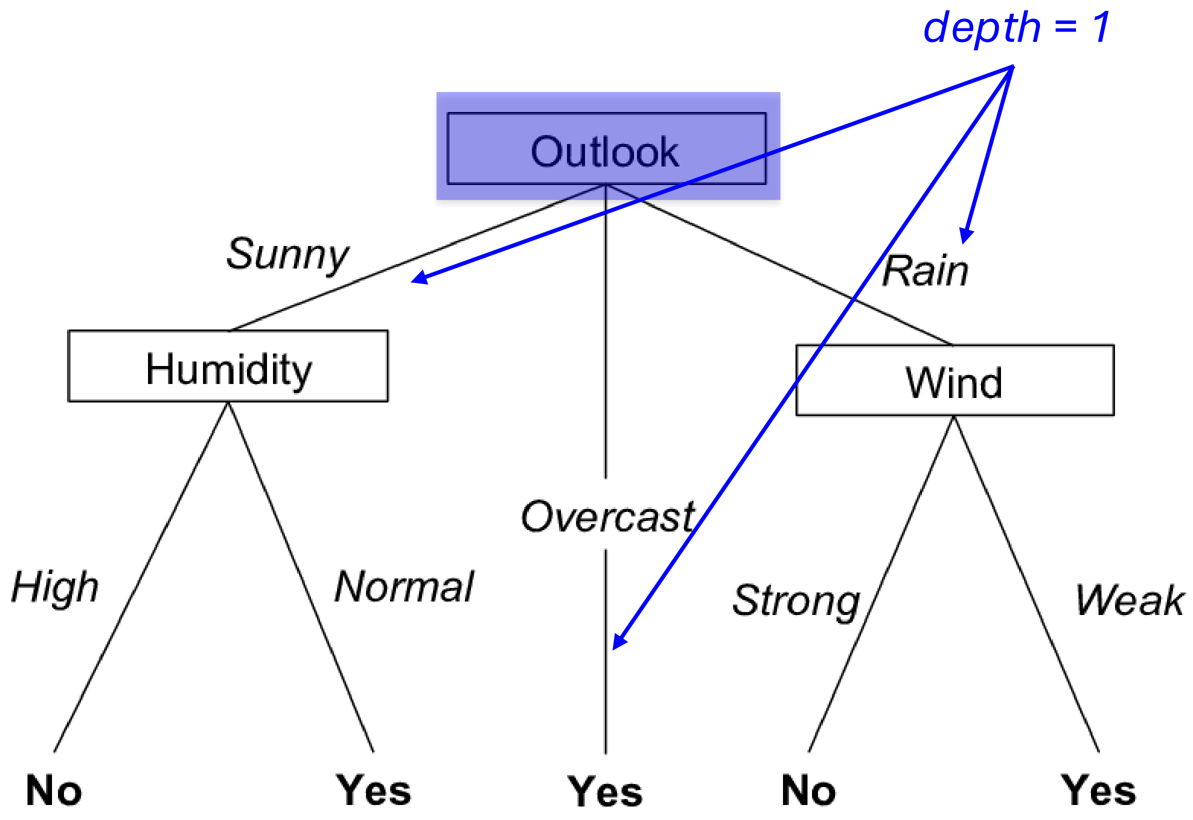
Key term: *depth*

Decision Tree example (tennis)



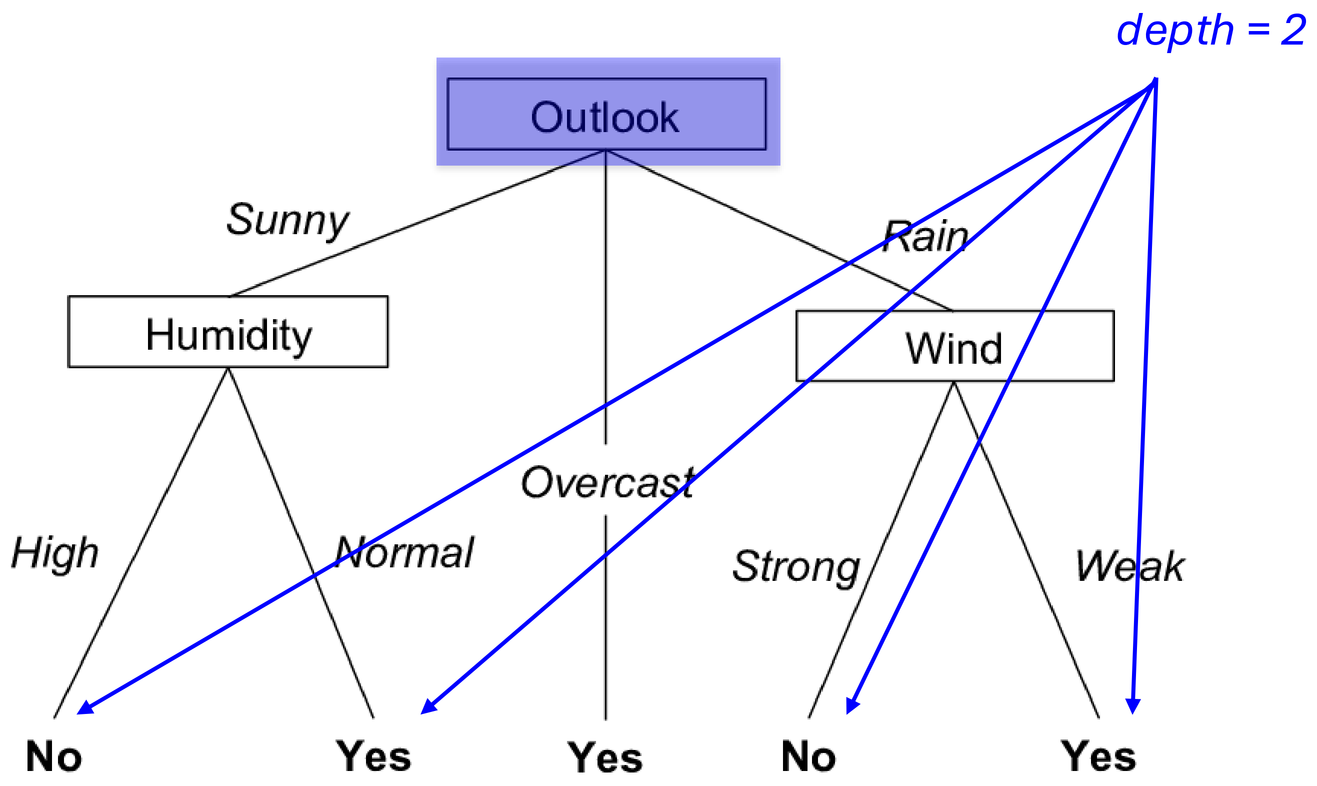
Key term: *depth*

Decision Tree example (tennis)



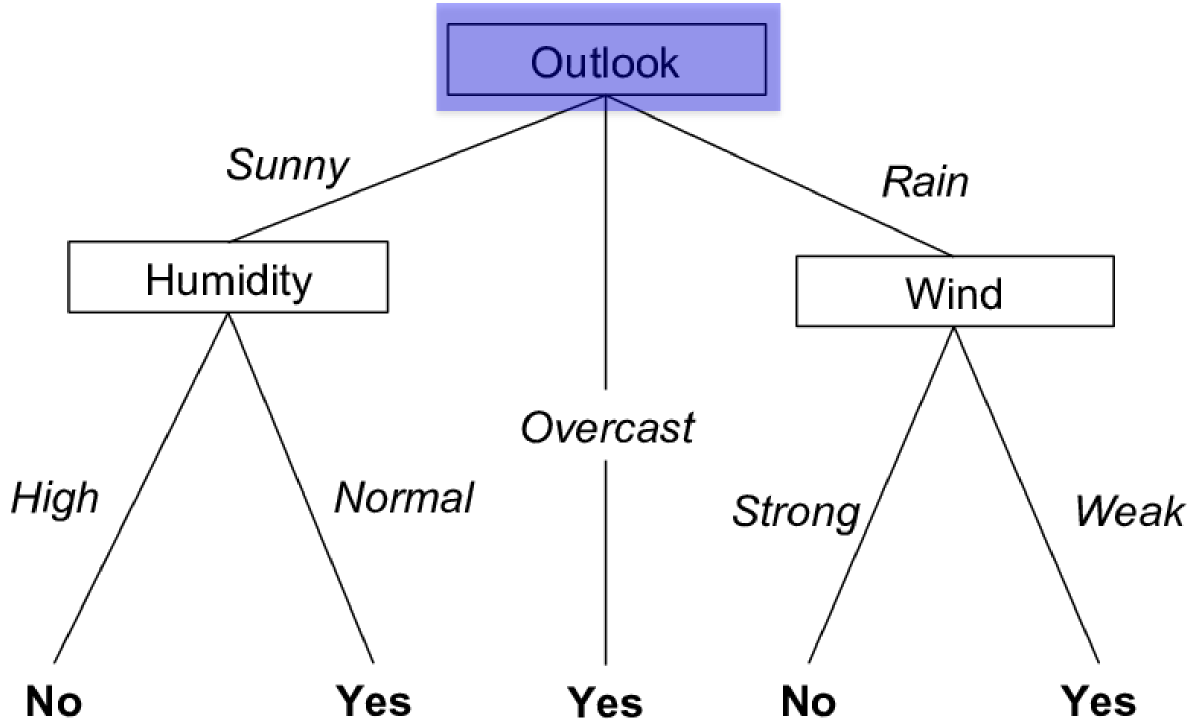
Key term: *depth*

Decision Tree example (tennis)



Key term: *depth*

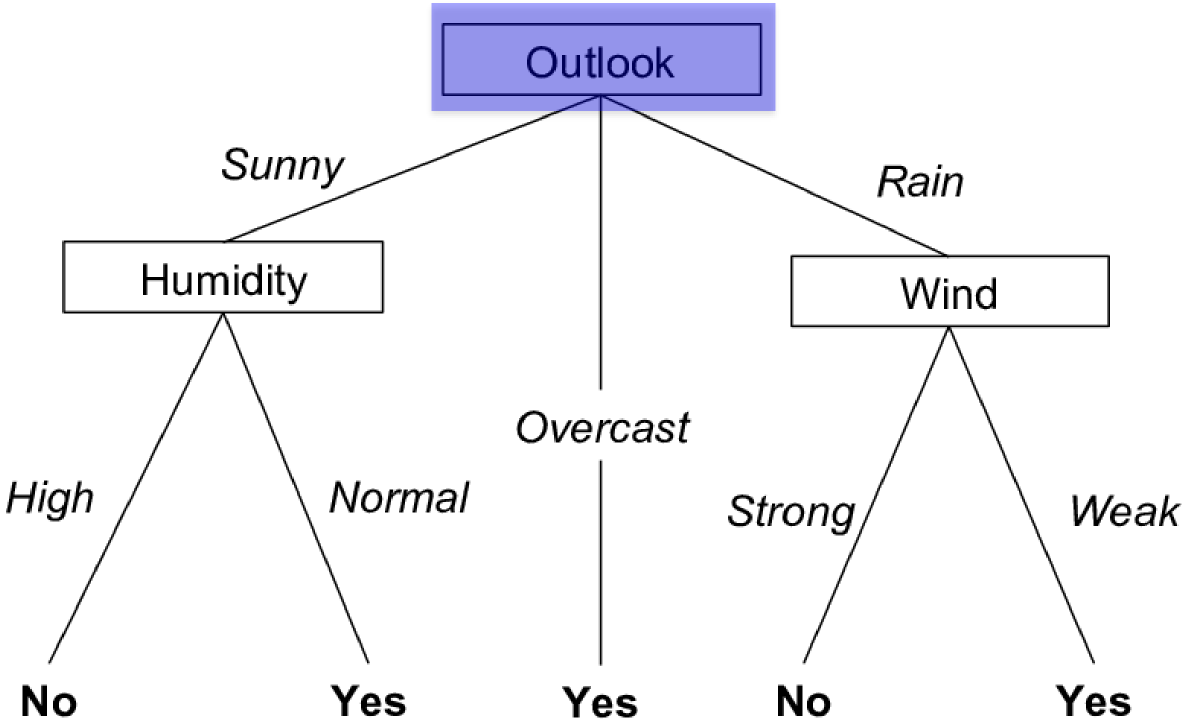
Decision Tree example (tennis)



(test example) $\boldsymbol{x}$ =

Outlook	Temp	Humidity	Wind
Rain	Hot	High	Strong

Decision Tree example (tennis)

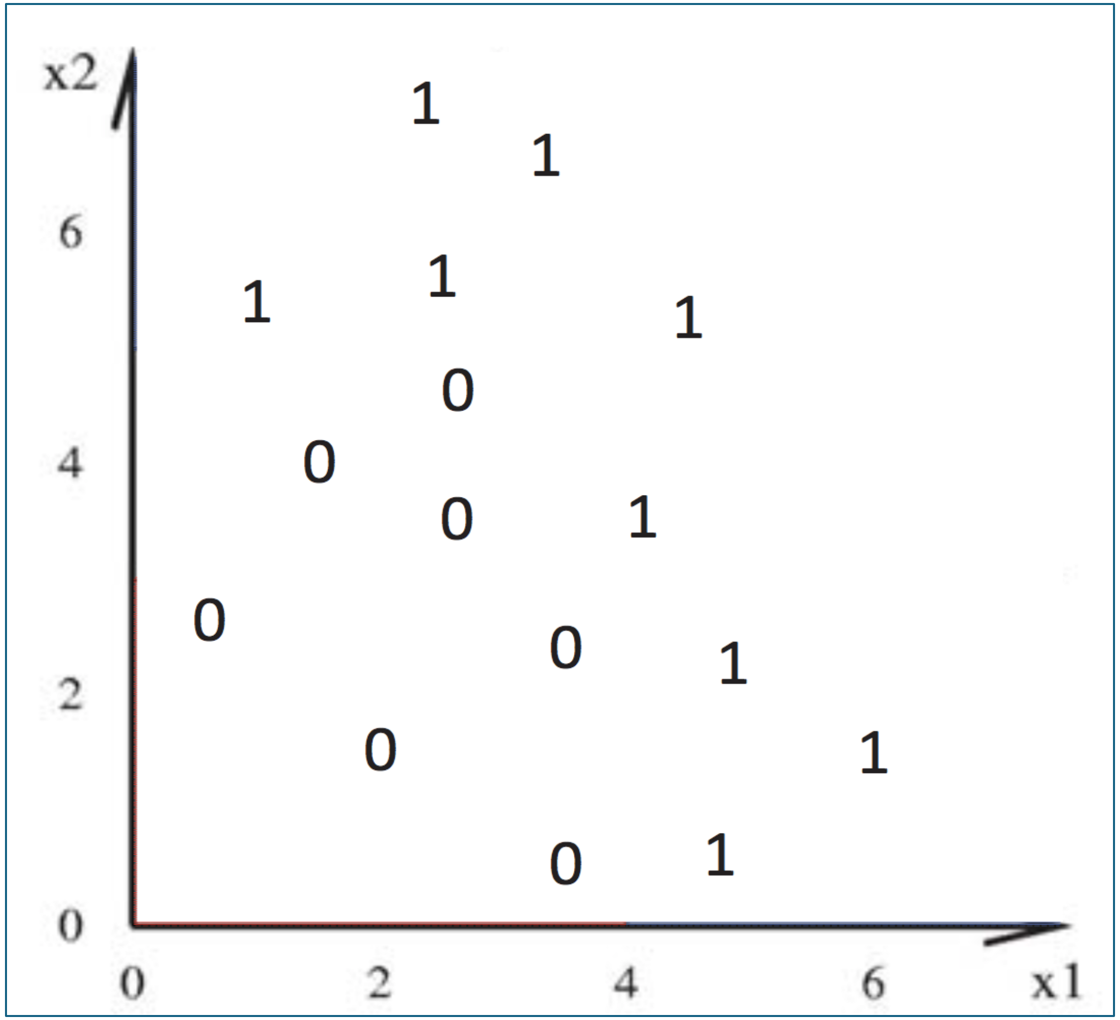


(test example) $\boldsymbol{x}$ =

Outlook	Temp	Humidity	Wind
Rain	Hot	High	Strong

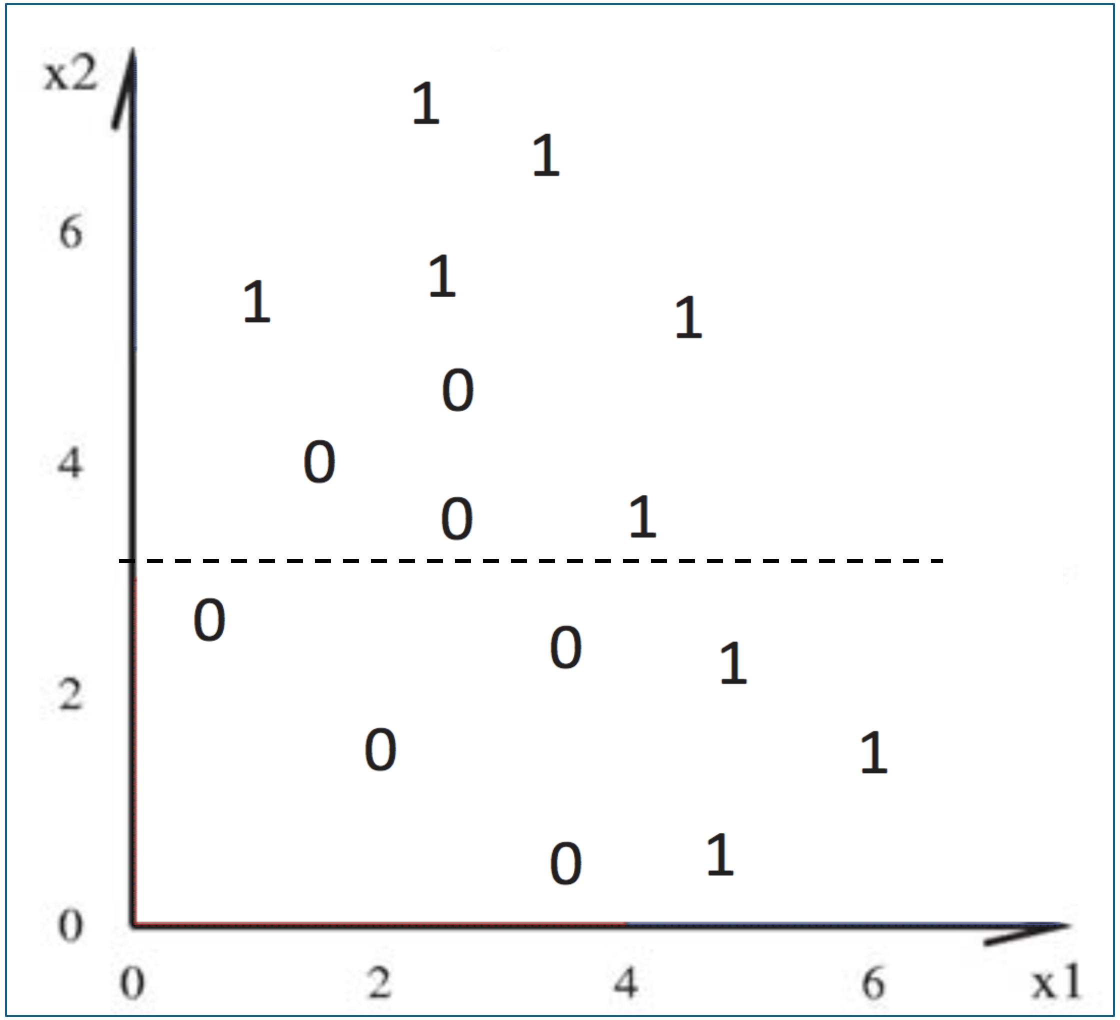
$\hat{y}$ = No

Continuous Features



Example by: Eric Eaton

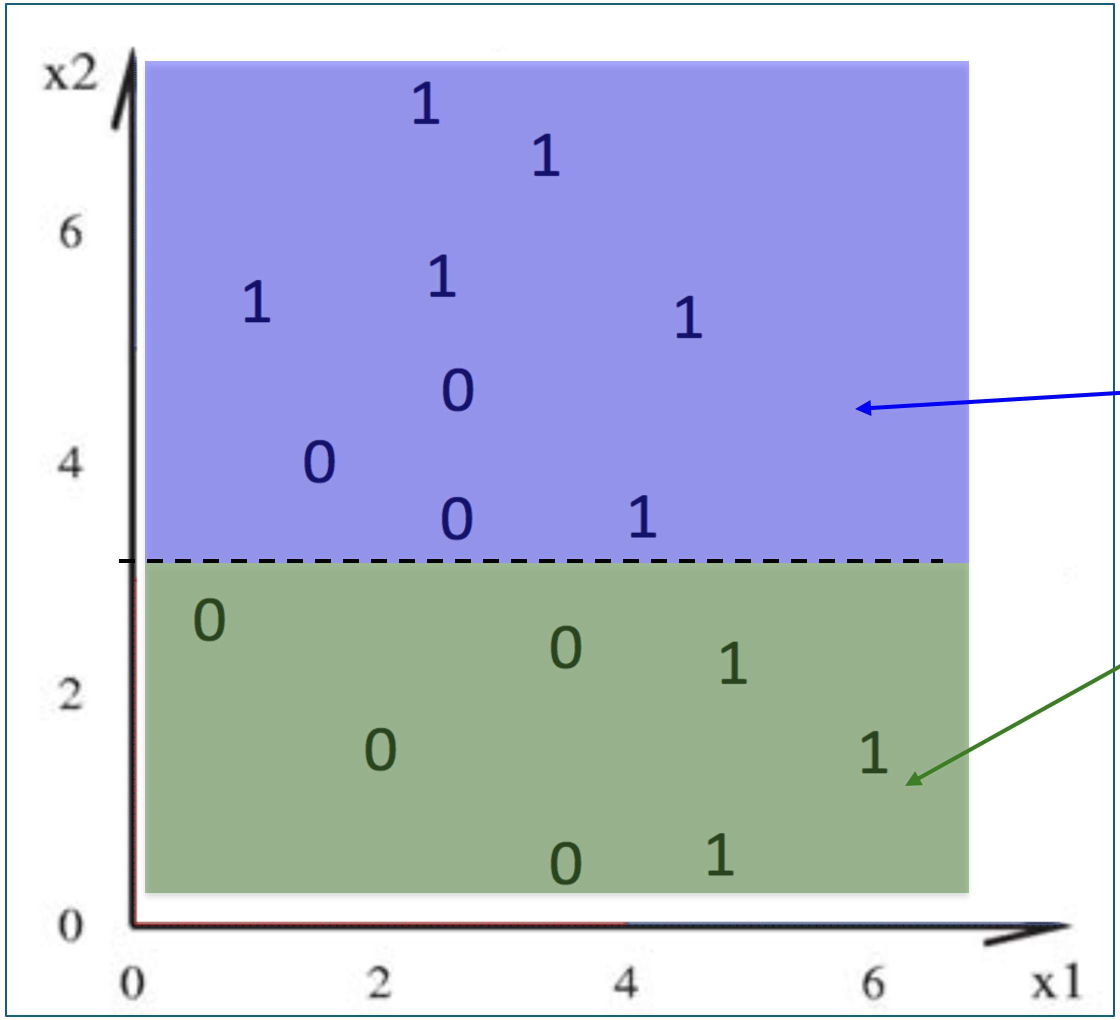
Continuous Features

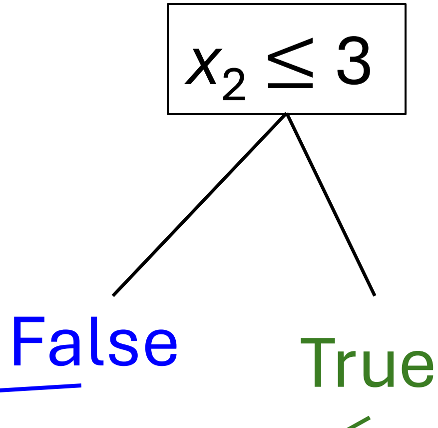


$x_2 \leq 3$

Example by: Eric Eaton

Continuous Features





Correction!

Example by: Eric Eaton

Decision Trees: Pros vs Cons

Discuss with a partner! Think about:

* training and testing
* featurization
* runtime
* human factors

Decision Trees: Pros vs Cons

- Very interpretable! Easy to say *why* we made a classification (can point to which features)

- Compact representation and fast predictions

- Can be brittle (not looking at each example holistically)

- Featurization and implementation difficulties

Next class

- Algorithm to create decision trees
- HW01 due Friday night
- Wednesday’s Lab will be a lecture
- Thursday lecture back on schedule
  - Reading quiz on Thursday