Decision Trees

Blake VanBerlo

Lecture 13

Readings: RN 19.3, PM 7.3.1.

Outline

Learning Goals

Examples of Decision Trees

Definition and Classifying an Example

Grow a Full Tree

Determine the Order of Testing Features

Real-Valued Features

Overfitting

Revisiting the Learning goals

Learning Goals

By the end of the lecture, you should be able to

- Describe the components of a decision tree.
- Construct a decision tree given an order of testing the features.
- Determine the prediction accuracy of a decision tree on a test set.
- Compute the entropy of a probability distribution.
- Compute the expected information gain for selecting a feature.
- Trace the execution of and implement the algorithm for learning a decision tree.
- Construct decision trees with real-valued features.
- Construct a decision tree to prevent over-fitting.

Learning Goals

Examples of Decision Trees

Definition and Classifying an Example

Grow a Full Tree

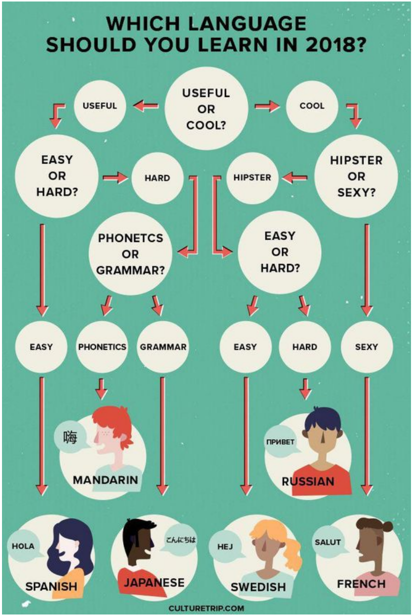
Determine the Order of Testing Features

Real-Valued Features

Overfitting

Revisiting the Learning goals

Which language should you learn?



WHICH LANGUAGE SHOULD YOU LEARN IN 2018?

USEFUL OR COOL?

USEFUL

COOL

EASY OR HARD?

HARD

HIPSTER

HIPSTER OR SEXY?

PHONETCS OR GRAMMAR?

EASY OR HARD?

EASY

PHONETICS

GRAMMAR

EASY

HARD

SEXY

MANDARIN

RUSSIAN

SPANISH

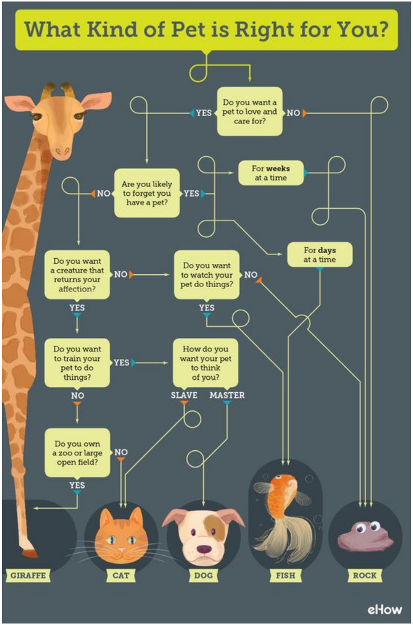
JAPANESE

SWEDISH

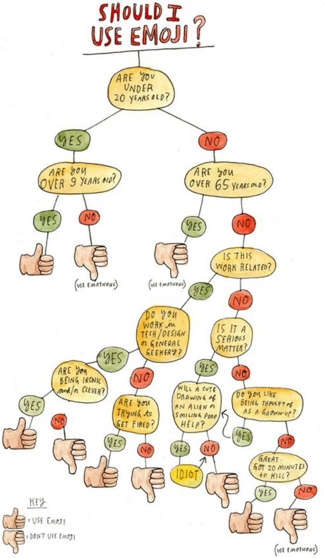
FRENCH

CULTURETRIP.COM

What kind of pet is right for you?


What Kind of Pet is Right for You?
Do you want a pet to love and care for?
YES
NO
Are you likely to forget you have a pet?
NO
YES
For weeks at a time
For days at a time
Do you want a creature that returns your affection?
NO
YES
Do you want to watch your pet do things?
NO
YES
Do you want to train your pet to do things?
YES
NO
How do you want your pet to think of you?
SLAVE
MASTER
Do you own a zoo or large open field?
NO
YES
GIRAFFE
CAT
DOG
FISH
ROCK
eHow

Should I use an emoji?


SHOULD I USE EMOJI?
ARE YOU UNDER 20 YEARS OLD?
YES
NO
ARE YOU OVER 9 YEARS OLD?
YES
NO
(USE EMOTICONS)
ARE YOU OVER 65 YEARS OLD?
YES
(USE EMOTICONS)
NO
IS THIS WORK RELATED?
YES
NO
DO YOU WORK IN TECH/DESIGN OR GENERAL GEEKERY?
YES
ARE YOU BEING IRONIC AND/OR CLEVER?
YES
NO
NO
ARE YOU TRYING TO GET FIRED?
YES
NO
IS IT A SERIOUS MATTER?
YES
WILL A CUTE DRAWING OF AN ALIEN OR SMILING POOP HELP?
YES
IDIOT
NO
NO
DO YOU LIKE BEING THOUGHT OF AS A GROWN-UP?
YES
NO
GREAT. GOT 20 MINUTES TO KILL?
YES
NO
(USE EMOTICONS)
KEY
👍 = USE EMOJI
👎 = DON'T USE EMOJI

Running example: Jeeves the valet

Jeeves is a valet to Bertie Wooster. On some days, Bertie likes to play tennis and asks Jeeves to lay out his tennis things and book the court. Jeeves would like to predict whether Bertie will play tennis (and so be a better valet). Each morning over the last two weeks, Jeeves has recorded whether Bertie played tennis on that day and various attributes of the weather (training set).

Jeeves would like to evaluate the classifier he has come up with for predicting whether Bertie will play tennis. Each morning over the next two weeks, Jeeves records more data (test set).

Jeeves the valet - training set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Jeeves the valet - test set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Mild	High	Strong	No
2	Rain	Hot	Normal	Strong	No
3	Rain	Cool	High	Strong	No
4	Overcast	Hot	High	Strong	Yes
5	Overcast	Cool	Normal	Weak	Yes
6	Rain	Hot	High	Weak	Yes
7	Overcast	Mild	Normal	Weak	Yes
8	Overcast	Cool	High	Weak	Yes
9	Rain	Cool	High	Weak	Yes
10	Rain	Mild	Normal	Strong	No
11	Overcast	Mild	High	Weak	Yes
12	Sunny	Mild	Normal	Weak	Yes
13	Sunny	Cool	High	Strong	No
14	Sunny	Cool	High	Weak	No

Learning Goals

Examples of Decision Trees

Definition and Classifying an Example

Grow a Full Tree

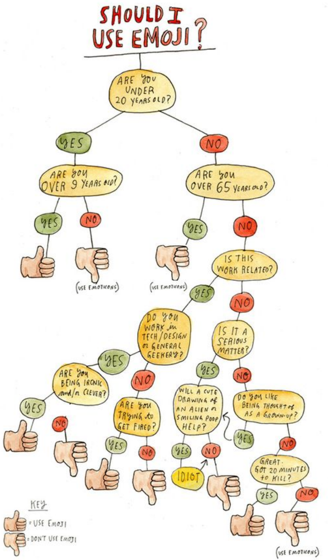
Determine the Order of Testing Features

Real-Valued Features

Overfitting

Revisiting the Learning goals

What is a decision tree?


SHOULD I USE EMOJI?
ARE YOU UNDER 20 YEARS OLD?
YES
NO
ARE YOU OVER 9 YEARS OLD?
ARE YOU OVER 65 YEARS OLD?
YES
NO
YES
NO
(USE EMOTICONS)
(USE EMOTICONS)
IS THIS WORK RELATED?
YES
NO
DO YOU WORK IN TECH/DESIGN OR GENERAL GEEKERY?
IS IT A SERIOUS MATTER?
YES
NO
YES
NO
ARE YOU BEING IRONIC AND/OR CLEVER?
ARE YOU TRYING TO GET FIRED?
WILL A CUTE DRAWING OF AN ALIEN OR SMILING POOP HELP?
DO YOU LIKE BEING THOUGHT OF AS A GROWN-UP?
YES
NO
YES
NO
YES
NO
YES
NO
IDIOT
GREAT. GOT 20 MINUTES TO KILL?
YES
NO
(USE EMOTICONS)
KEY
= USE EMOJI
= DON'T USE EMOJI

What is a decision tree?

- A simple model for supervised classification.

  → One of the simplest, yet useful and widely successful ML algorithms.

- A single discrete target feature.

  → Each value in the domain of the target feature is a class.

- Each internal node performs a Boolean test on an input feature.

- The edges are labeled with values of the input feature.

  → yes/no. true/false.

- Each leaf node specifies a value for the target feature.

  → specifies a class.

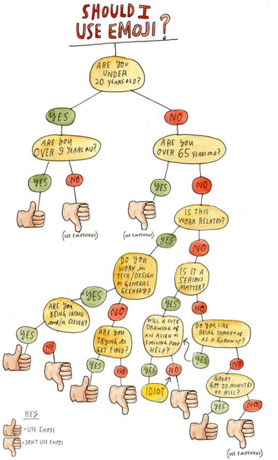
Classify an example using a decision tree

- Go down the tree, evaluate each test and follow the corresponding edge.
- When a leaf is reached, return the classification on that leaf.

E.g. Should I use an emoji?

- I am 30 years old.
- This is work related.
- I am an accountant.
- I am not trying to get fired.

→ No.


SHOULD I USE EMOJI?
ARE YOU UNDER 20 YEARS OLD?
YES
NO
ARE YOU OVER 9 YEARS OLD?
YES
NO
(USE EMOTICONS)
ARE YOU OVER 65 YEARS OLD?
YES
(USE EMOTICONS)
NO
IS THIS WORK RELATED?
YES
NO
DO YOU WORK IN TECH/DESIGN OR GENERAL GEEKERY?
YES
ARE YOU BEING IRONIC AND/OR CLEVER?
YES
NO
NO
ARE YOU TRYING TO GET FIRED?
YES
NO
IS IT A SERIOUS MATTER?
YES
WILL A CUTE DRAWING OF AN ALIEN OR SMILING POOP HELP?
YES
IDIOT
NO
NO
DO YOU LIKE BEING THOUGHT OF AS A GROWN-UP?
YES
NO
GREAT. GOT 20 MINUTES TO KILL?
YES
NO
(USE EMOTICONS)
KEY
= USE EMOJI
= DON'T USE EMOJI

If we convert a decision tree to a program, what does it look like?

→ A nested if-then-else structure.

Issues in learning a decision tree

How should we build a decision tree?

- We need to determine an order of testing the input features.
- Given an order of testing the input features,
  we can build a decision tree by splitting the examples.

Issues in learning a decision tree

How should we build a decision tree?

- We need to determine an order of testing the input features.
- Given an order of testing the input features, we can build a decision tree by splitting the examples.

Which decision tree should we generate?

- Which order of testing the input features should we use?
  - The search space is too big for systematic search.
  - Solution: greedy (myopic) search.
- Should we grow a full tree or not?
  - A decision tree can represent any discrete function of input features.
  - Need a bias. For example, prefer the smallest tree. (Least depth? Fewest nodes?)

Learning Goals

Examples of Decision Trees

Definition and Classifying an Example

Grow a Full Tree

Determine the Order of Testing Features

Real-Valued Features

Overfitting

Revisiting the Learning goals

Construct a Decision Tree
Given an Order of Testing Features

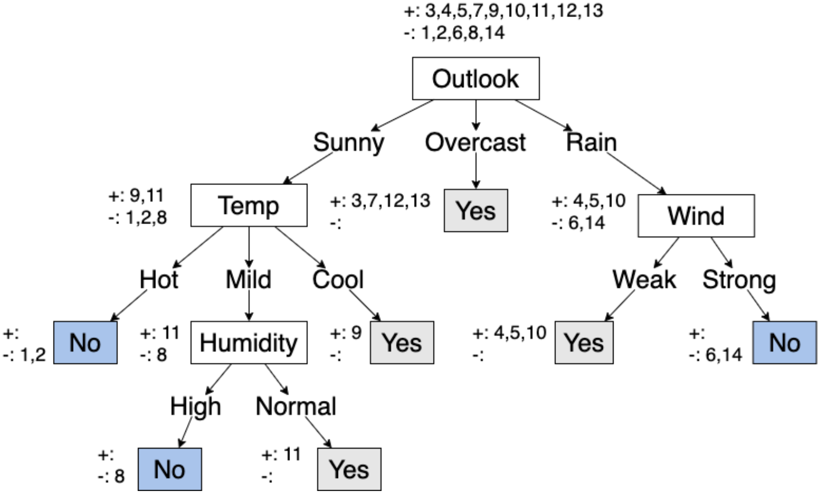
→ Given an order of testing features, it's straightforward to generate a decision tree.

Construct a decision tree
using the following order of testing features.

- First, test Outlook.
- For Outlook = Sunny, test Temp.
- For Outlook = Rain, test Wind.
- For other branches, test Humidity before testing Wind.

Let's draw the decision tree.

Example: Decision tree



When do we stop?

We stop in any of the following scenarios:

- All the examples belong to the same class.
- There are no more features to test.
- There are no more examples.

No features left

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No
15	Sunny	Mild	High	Weak	No
16	Sunny	Mild	High	Weak	Yes
17	Sunny	Mild	High	Strong	Yes

No features left

Complete the branch of the tree where

Outlook = Sunny, Temp = Mild, and Humidity = High.

No features left

Why do we have this special case?

→ The data is noisy. Even if we know the values of all the features, we still cannot make a deterministic decision. Perhaps, the outcome is influenced by a feature that we don't observe.

What are some possible ways of handling this case?

→

- Predict the majority class.
- Predict the labels probabilistically.

No examples left

Consider the revised training set below.

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No
15	Sunny	Hot	High	Weak	No

No examples left

Complete the branch of the tree where

Temp = Hot, Wind = Weak, and Humidity = High.

No examples left

Why do we have this special case?

→ A combination of features is not present in the training set. If we never observe this combination, we don't know how to predict it.

What are some possible ways of handling this case?

→ Use the majority decision in the examples at the parent node.

The Decision Tree Learner algorithm

Algorithm 1 Decision Tree Learner (examples, features)

```
if all examples are in the same class then
    return the class label.
else if no features left then
    return the majority decision.
else if no examples left then
    return the majority decision at the parent node.
else
    choose a feature $f$.
    for each value $v$ of feature $f$ do
        build edge with label $v$.
        build sub-tree using examples where the value of $f$ is $v$.
```

Which feature should we test at each step?

Learning the "best" decision tree

- Each order of testing features leads to a different decision tree. Many decision trees can predict the training examples perfectly. Which tree do we prefer?

  → We prefer the smallest tree, since a simple model is more likely to predict unseen data well.

Learning the "best" decision tree

- Each order of testing features leads to a different decision tree. Many decision trees can predict the training examples perfectly. Which tree do we prefer?

  → We prefer the smallest tree, since a simple model is more likely to predict unseen data well.

- How many possible decision trees are there?

  How do we find the smallest decision tree?

  → A lot. Can represent all discrete functions of features. Cannot do systematic search. Intractable to find the optimal order of testing features.

  Do greedy search.

  Greedy search: make the myopic best decision at each step. (Best for that step, may not be best for minimizing the tree)

How many functions can we encode with decision trees?

Each decision tree encodes a propositional formula.

$\rightarrow$ The smaller tree is equivalent to $(Outlook = Sunny \wedge Humidity = Normal) \vee (Outlook = Overcast) \vee (Outlook = Rain \wedge Wind = Weak)$

If we have n features, how many different functions can we encode with decisions trees?

$\rightarrow$ Each function corresponds to a truth table. Each truth table has 2^n rows. There are 2^{2^n} possible truth tables.

With $n = 10$, $2^{1024} \approx 10^{308}$

Learning Goals

Examples of Decision Trees

Definition and Classifying an Example

Grow a Full Tree

Determine the Order of Testing Features

Real-Valued Features

Overfitting

Revisiting the Learning goals

Which feature should we test at each step?

- How can we create a shallow/small tree?
  Need to minimize the number of tests.

- Finding the optimal order of testing features is difficult.

  Greedy search — make the best myopic choice at each step.

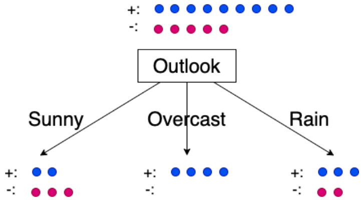
- At each step, test a feature that
  makes the biggest difference to the classification.

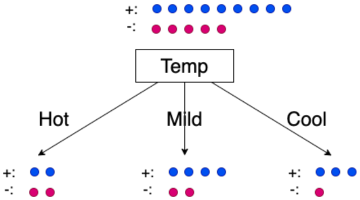
Which feature should we test?

Which of the two features below should we test?

(A) Outlook

(B) Temp



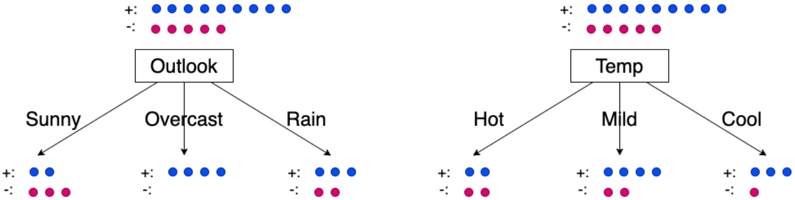


Which feature should we test?

Which of the two features below should we test?

(A) Outlook

(B) Temp



→ (A) We should test Outlook since it reduces uncertainty more than Temp.

Identifying the most important feature

- We want to make a decision as soon as possible
  → reduce uncertainty as much as possible
- Information content of a feature
  = uncertainty before testing the feature − uncertainty after testing the feature
- Measure uncertainty using the notion of entropy.

Given a distribution $P(c_1), ..., P(c_k)$ over k outcomes $c_1, ..., c_k$, the **entropy** of the distribution is

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

→ Base 2 since we measure info content in bits.

Q: Entropy of a distribution over two outcomes

Q #1: What is the entropy of the distribution $(0.5, 0.5)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.2

(B) 0.4

(C) 0.6

(D) 0.8

(E) 1

Q: Entropy of a distribution over two outcomes

Q #1: What is the entropy of the distribution $(0.5, 0.5)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.2

(B) 0.4

(C) 0.6

(D) 0.8

(E) 1

$\rightarrow I(0.5, 0.5) = -\frac{1}{2} log_2(1/2) - \frac{1}{2} log_2(1/2) = 1.$

There is one bit of uncertainty in this distribution.

There is a lot of uncertainty in this distribution.

Q: Entropy of a distribution over two outcomes

Q #2: What is the entropy of the distribution $(0.01, 0.99)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.02

(B) 0.04

(C) 0.06

(D) 0.08

(E) 0.1

Q: Entropy of a distribution over two outcomes

Q #2: What is the entropy of the distribution $(0.01, 0.99)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.02

(B) 0.04

(C) 0.06

(D) 0.08

(E) 0.1

$\rightarrow I(0.99, 0.01) = -0.99 log_2(0.99) - 0.01 log_2(0.01) = 0.08.$

There is 0.08 bits of uncertainty in this distribution.

There is very little uncertainty in this distribution. We almost know for sure that the outcome will be the second one.

Entropy of a distribution over two outcomes

Consider a distribution $(p, 1-p)$ where $0 \leq p \leq 1$.

- What is the maximum entropy of this distribution?

  $\rightarrow$ maximum is at $p = 1/2$.

- What is the minimum entropy of this distribution?

  $\rightarrow$ minimum is at $p = 0$ and $p = 1$.

  By definition $I(1,0) = 0$ and $I(0,1) = 0$.

- Plot the entropy of the distribution $(p, 1-p)$ with respect to p.

Expected information gain of testing a feature

- The feature has k values $v_1, ..., v_k$.
- Before testing the feature,
  we have p positive and n negative examples.
- After testing the feature,
  for each value v_i of the feature,
  we have p_i positive and n_i negative examples.

Let's calculate the expected information gain of testing this feature.

Expected information gain of testing a feature

Entropy before testing the feature:

→

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

Expected entropy after testing the feature:

→

$$H_{after} = \sum_{i=1}^{k} \frac{p_i + n_i}{p+n} * I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

Expected information gain (entropy reduction) is

→

$$InformationGain = H_{before} - H_{after}$$

→ can be generalized to more than two classes as well.

Q: Entropy and Information Gain

Q #3: What is the entropy of the examples before we select a feature for the root node of the tree?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

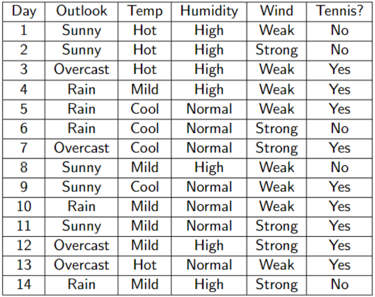
(A) 0.54

(B) 0.64

(C) 0.74

(D) 0.84

(E) 0.94



Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Q: Entropy and Information Gain

Q #3: What is the entropy of the examples before we select a feature for the root node of the tree?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i)\log_2(P(c_i))$$

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

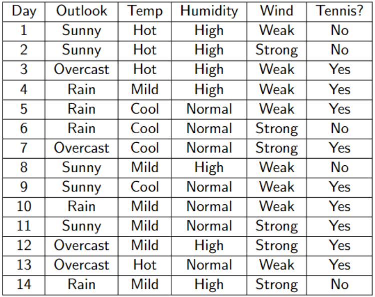
(A) 0.54

(B) 0.64

(C) 0.74

(D) 0.84

(E) 0.94



Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

→ Correct answer is 0.940 (E).

Q: Entropy and Information Gain

Q #4: What is the expected information gain if we select **Outlook** as the root node of the tree?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i)\log_2(P(c_i))$$

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

$$H_{after} = \sum_{i=1}^{k} \frac{p_i+n_i}{p+n} * I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

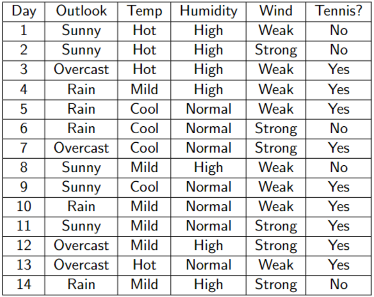
(A) 0.237

(B) 0.247

(C) 0.257

(D) 0.267

(E) 0.277



Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Q: Entropy and Information Gain

Q #4: What is the expected information gain
if we select **Outlook** as the root node of the tree?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i)\log_2(P(c_i))$$

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

$$H_{after} = \sum_{i=1}^{k} \frac{p_i+n_i}{p+n} * I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

(A) 0.237

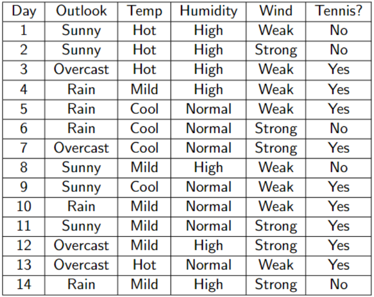
(B) 0.247

(C) 0.257

(D) 0.267

(E) 0.277

→ Correct answer is 0.247 (B).



Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Q: Entropy and Information Gain

Q #5: What is the expected information gain if we select **Humidity** as the root node of the tree?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i)\log_2(P(c_i))$$

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

$$H_{after} = \sum_{i=1}^{k} \frac{p_i+n_i}{p+n} * I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

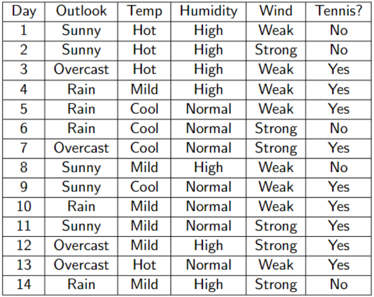
(A) 0.151

(B) 0.251

(C) 0.351

(D) 0.451

(E) 0.551



Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Q: Entropy and Information Gain

Q #5: What is the expected information gain if we select **Humidity** as the root node of the tree?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i)\log_2(P(c_i))$$

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

$$H_{after} = \sum_{i=1}^{k} \frac{p_i+n_i}{p+n} * I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

(A) 0.151

(B) 0.251

(C) 0.351

(D) 0.451

(E) 0.551

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

→ Correct answer is 0.151 (A).

Learning Goals

Examples of Decision Trees

Definition and Classifying an Example

Grow a Full Tree

Determine the Order of Testing Features

Real-Valued Features

Overfitting

Revisiting the Learning goals

Jeeves dataset with real-valued temperatures

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	29.4	High	Weak	No
2	Sunny	26.6	High	Strong	No
3	Overcast	28.3	High	Weak	Yes
4	Rain	21.1	High	Weak	Yes
5	Rain	20.0	Normal	Weak	Yes
6	Rain	18.3	Normal	Strong	No
7	Overcast	17.7	Normal	Strong	Yes
8	Sunny	22.2	High	Weak	No
9	Sunny	20.6	Normal	Weak	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
12	Overcast	22.2	High	Strong	Yes
13	Overcast	27.2	Normal	Weak	Yes
14	Rain	21.7	High	Strong	No

Jeeves dataset ordered by temperatures

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

Handling a discrete feature

- Allow multi-way splits.

  → Multiway split: tree becomes more complex than if-then-else. tree tends to be shallower. Info gain prefers a variable with a larger domain.

- Restrict to binary splits.

  → Binary split: tree is simpler and more compact. Tree tends to be deeper.

→ Trade-off: complexity of the tree. Depth of the tree.

Draw diagram: 4-way split is equivalent to 3 binary splits.

Handling a real-valued feature

- Discretize the feature.

  → Advantage: easy to do.

  Disadvantage: lose valuable information. The discretization might make the decision tree much more complicated.

- Allow multi-way splits.

- Restrict to binary splits.

  → Cannot do multiway split. Domain could be unbounded.

  If we do binary split, we may test the feature many times. Tree size may blow up.

Q: Testing a discrete feature

Q #6: Suppose that feature X has **discrete** values (e.g. Temp is Cool, Mild, or Hot.) On any path from the root to a leaf, how many times can we test feature X?

(A) 0 times

(B) 1 time

(C) > 1 time

(D) Two of (A), (B), and (C) are correct.

(E) All of (A), (B), and (C) are correct.

Q: Testing a discrete feature

Q #6: Suppose that feature X has **discrete** values (e.g. Temp is Cool, Mild, or Hot.) On any path from the root to a leaf, how many times can we test feature X?

(A) 0 times

(B) 1 time

(C) > 1 time

(D) Two of (A), (B), and (C) are correct.

(E) All of (A), (B), and (C) are correct.

→ Correct answer is (D). A discrete feature is tested at most once along any path.

Q: Testing a real-valued feature

Q #7: Assume that we will do binary tests at each node in a decision tree. Suppose that feature X has **real** values (e.g. Temp ranges from 17.7 to 29.4.) On any path from the root to a leaf, how many times can we test feature X?

(A) 0 times

(B) 1 time

(C) > 1 time

(D) Two of (A), (B), and (C) are correct.

(E) All of (A), (B), and (C) are correct.

Q: Testing a real-valued feature

Q #7: Assume that we will do binary tests at each node in a decision tree. Suppose that feature X has **real** values (e.g. Temp ranges from 17.7 to 29.4.) On any path from the root to a leaf, how many times can we test feature X?

(A) 0 times

(B) 1 time

(C) > 1 time

(D) Two of (A), (B), and (C) are correct.

(E) All of (A), (B), and (C) are correct.

→ (E). A real-valued feature can be tested many times.

On any path from the root to a leaf, a real-valued feature can be tested 0 times, 1 time, or more than 1 time.

Problem: The tree may be larger and difficult to understand. The depth of the tree is still bounded by the number of data points.

Choosing a split point for a real-valued feature

1. Sort the instances according to the real-valued feature
2. Possible split points are values that are midway between two different and adjacent values.
3. Suppose that the feature changes from X to Y. Should we consider $(X+Y)/2$ as a possible split point?
4. Let L_X be all the labels for the examples where the feature takes the value X.
5. Let L_Y be all the labels for the examples where the feature takes the value Y.
6. If there exists a label $a \in L_X$ and a label $b \in L_Y$ such that $a \neq b$, then $(X+Y)/2$ is a possible split point.
7. Determine the expected information gain for each possible split point and choose the split point with the largest gain.

Q: Is this a possible split point?

Q #8: For the Jeeves training set, is the midpoint between 20.0 and 20.6 a possible split point?

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

(A) Yes.

(B) No.

Q: Is this a possible split point?

Q #8: For the Jeeves training set, is the midpoint between 20.0 and 20.6 a possible split point?

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

(A) Yes.

(B) No.

→ Answer: No. The two labels are the same.

Q: Is this a possible split point?

Q #9: For the Jeeves training set, is the midpoint between 21.1 and 21.7 a possible split point?

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

(A) Yes.

(B) No.

Q: Is this a possible split point?

Q #9: For the Jeeves training set, is the midpoint between 21.1 and 21.7 a possible split point?

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

(A) Yes.

(B) No.

→ Answer: Yes. The two labels are different.

Q: Is this a possible split point?

Q #10: For the Jeeves training set, is the midpoint between 21.7 and 22.2 a possible split point?

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

(A) Yes.

(B) No.

Q: Is this a possible split point?

Q #10: For the Jeeves training set, is the midpoint between 21.7 and 22.2 a possible split point?

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

(A) Yes.

(B) No.

→ Answer: Yes. The label for example 14 (21.7) is different from the label for example 12 (22.2).

Learning Goals

Examples of Decision Trees

Definition and Classifying an Example

Grow a Full Tree

Determine the Order of Testing Features

Real-Valued Features

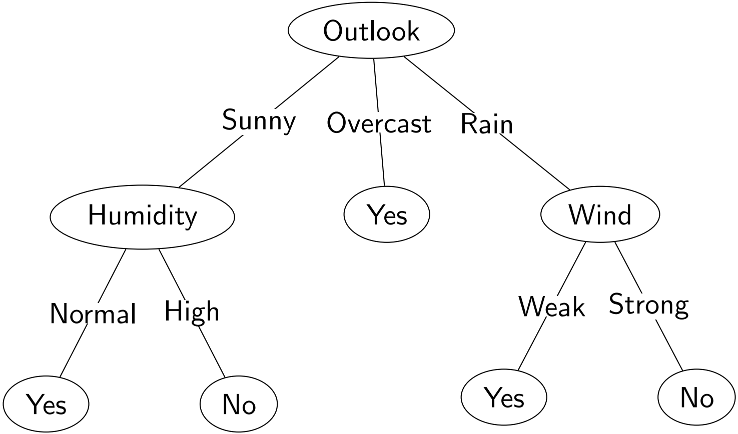
Overfitting

Revisiting the Learning goals

Jeeves the valet - training set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Decision tree generated by the learner algorithm

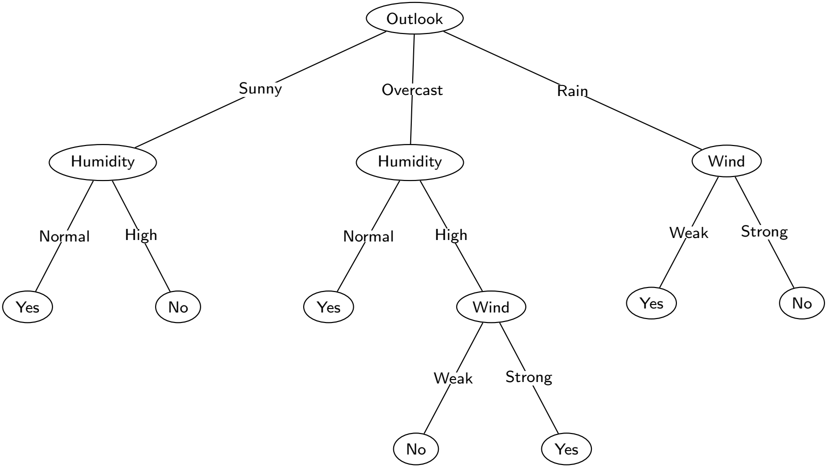


Test error is 0/14.

Jeeves the valet - a corrupted training set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	**No**
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Decision tree for the corrupted data set



Test error is 2/14.

Dealing with Overfitting

Problem: Growing a full tree is likely to lead to overfitting.

Strategies to prevent overfitting:

- Pre-pruning
  - Maximum depth
  - Minimum examples at a leaf node
  - Minimum information gain
  - Reduction in training error
- Post-pruning (using the criteria above)

  → It's possible that two features together predict well but one of them by itself, is not very useful. XOR example. One bit provides no information. Two bits predict the result perfectly.

Revisiting the Learning Goals

By the end of the lecture, you should be able to

- Describe the components of a decision tree.
- Construct a decision tree given an order of testing the features.
- Determine the prediction accuracy of a decision tree on a test set.
- Compute the entropy of a probability distribution.
- Compute the expected information gain for selecting a feature.
- Trace the execution of and implement the algorithm for learning a decision tree.
- Construct decision trees with real-valued features.
- Construct a decision tree to prevent over-fitting.