Computer Science

CSC380: Principles of Data Science

Predictive Modeling and Classification 1

Intro to Decision trees

Credit:
- Jason Pacheco,
- Kwang-Sung Jun,
- Chicheng Zhang
- Xinchen yu

1

3

Introduction to Machine Learning

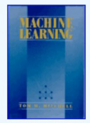
3

What is machine learning?

4

- **Tom Mitchell** established Machine Learning Department at CMU (2006).

Machine Learning, Tom Mitchell, McGraw Hill, 1997. “through experience”



Machine Learning is the study of computer algorithms that improve automatically through experience. Applications range from datamining programs that discover general rules in large data sets, to information filtering systems that automatically learn users' interests.

This book provides a single source introduction to the field. It is written for advanced undergraduate and graduate students, and for developers and researchers in the field. No prior background in artificial intelligence or statistics is assumed.

- A bit outdated with recent trends, but still has interesting discussion (and easy to read).
- A subfield of **Artificial Intelligence** – you want to perform nontrivial, smart tasks. The difference from the traditional AI is “**how**” you build a computer program to do it.

4

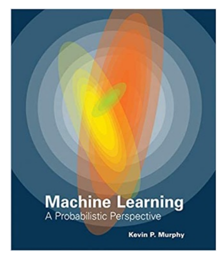
Textbooks

5

We will use a more recent textbook for readings

*Takes a **probabilistic approach** to machine learning*

Consistent with the goals of data science in this class



Murphy, K. "Machine Learning: A Probabilistic Perspective." MIT press, 2012

(UA Library)

5

AI Task 1: Image classification

6

- Predefined categories: C = {cat, dog, lion, …}
- Given an image, classify it as one of the categories $c \in C$ with the highest accuracy.
- **Use**: sorting/searching images by category.
- **Other example**: categorize types of stars/events in the Universe (images taken from large surveying telescopes)

6

AI Task 2: Recommender systems

7

- Predict how user would rate a movie
- **Use**: For each user, pick an unwatched movie with the high predicted ratings.
- **Idea**: compute user-user similarity or movie-movie similarity, then compute a weighted average.

	User 1	User 2	User 3
Movie 1	1	2	1
Movie 2	?	3	1
Movie 3	2	5	2
Movie 4	4	?	5
Movie 5	?	4	5

"collaborative filtering"

7

AI Task 3: Machine translation

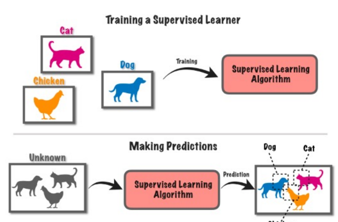
8

- No need to explain how useful it is.

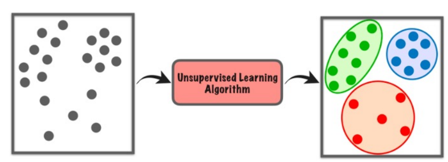
8

Supervised vs Unsupervised Learning

- **Supervised Learning** - Training data consist of inputs and outputs
  - Classification, regression, translation, …



- **Unsupervised Learning** – Training data only contain inputs
  - Clustering, dimensionality reduction, segmentation, k-nearest neighbors …



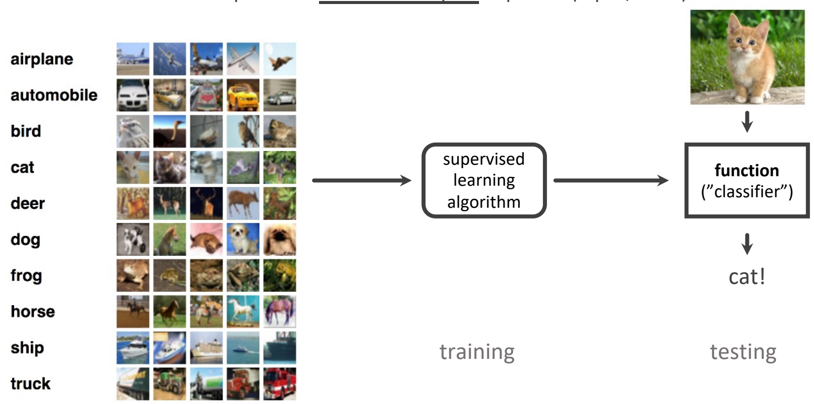
9

Supervised learning

example = data point
labeled = categorized

10

- Train data: dataset comprised of *labeled examples*: a pair of (input, label)

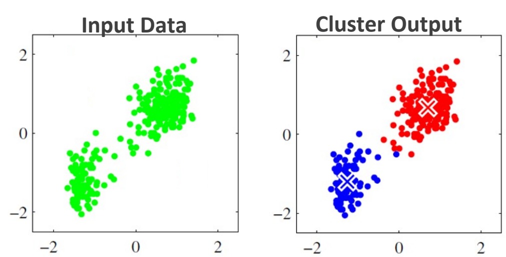


10

Unsupervised learning: Clustering

11

Identify groups (clusters) of similar data



Useful for interpreting large datasets

Clusters are assigned arbitrary labels (e.g. 1, 2, …, K).
=> afterwards, you may look at the data and name **each group**.
(no need to name each sample)

Common clustering algorithms: K-means, Expectation Maximization (EM)

11

Decision Trees

12

Majority Vote Classifier

The most basic classifier you can think of.

How to train:

- Given: A (train) dataset with m data points $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ with C classes.
- Compute the most common class c^* in the dataset.

$$c^* = \arg\max_{c \in \{1,\ldots,C\}} \sum_{i=1}^{m} \mathbf{I}\{y^{(i)} = c\}$$

- Output a classifier $f(x) = c^*$.

Stupid enough classifier! Always try to beat this classifier.

Often, state-of-the-art ML algorithms perform barely better than the majority vote classifier..
⇒ happens when there is no association between features and labels in the dataset

13

Train set accuracy

- Suppose the ML algorithm has trained a function f using the dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ where $x^{(i)}$ is input and $y^{(i)}$ is label.
- Train set accuracy:

This is 1 if f returns the correct label on $x^{\{(i)\}}$

$$\widehat{acc}(f) := \frac{1}{m}\sum_{i=1}^{m} \mathbf{I}\{f(x^{(i)}) = y^{(i)}\}$$

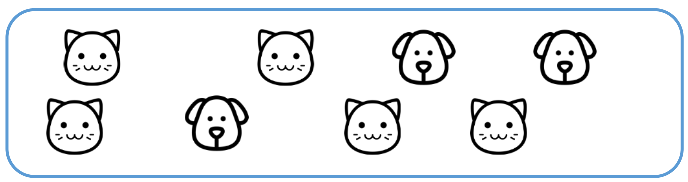
- Q: We have 100 data points (images) with 5 cats, 80 dogs, and 15 lions. What is the train set accuracy of the majority vote classifier?

quiz candidate

For each of 100 data points, the predicted label is dog: 80/100=.80

Train set accuracy

If the model is majority vote classifier..



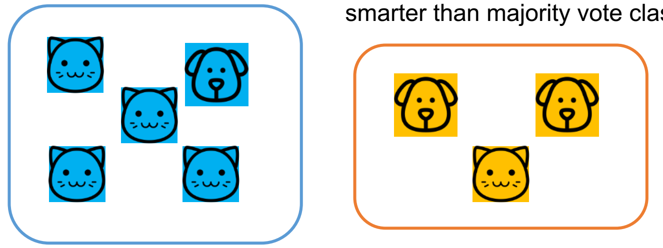
Majority vote: cat

Q: what is the accuracy? $\frac{5}{8}$

Train set accuracy

Given query, first find the DataSet containing it, and then use majority

Suppose the model is a little bit smarter than majority vote classifier..



Majority vote: cat

Majority vote: dog

Q: what is the accuracy? $\frac{6}{8} = \frac{5}{8} \cdot \frac{4}{5} + \frac{3}{8} \cdot \frac{2}{3} = \frac{3}{4}$

16

Example: course recommendation

17

- Data: Set of students that took a course, each with their preferences, and whether the have liked the course, build a decision tree such that
- given the preferences of a new student, predict if she/he would like the course.

is it a systems course?
is it an application course?
who is the instructor?

course description
student info → function → rating ∈ {+, −}

what courses have you taken?
do you like morning class?

17

18

HasTakenPrereqs (=: Prereq)
HasTakenACourseFromTheSameLecturer (=: Lecturer)
HasLabs

y_i= Label

consider it to be 'like'

consider it to be 'dislike'

Rating	Easy?				Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

student i

For example, this table is data D;
Each row is a course you have rated;
$x^{(i)}$ is a sequence of 5 yes/no for the i-th course;
$y^{(i)}$ is the sign of rating for the i-th course.

Define the data $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$

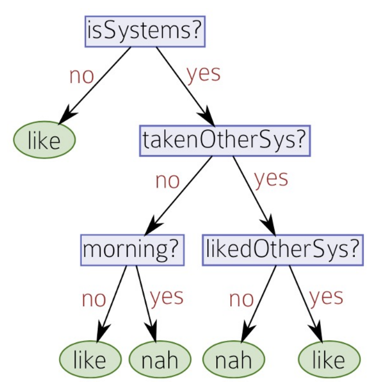
$\in \{y, n\}^d$

$\in \{+, -\}$

Each dimension of $x^{(i)}$ is called a **feature**.
$x^{(i)}$ is called a **feature vector**.

18

19



Wouldn't it be nice to construct such a tree automatically by a computer algorithm?

Wouldn't it be nice if it accurately predicts?

You can, if you have data!

- Many algorithms for building decisions trees
- In general, building a tree with minimum prediction error is NP-hard
- Our approach: Greedy:
- Pick the feature that maximize accuracy, under the assumption that after this feature we only use Majority vote.
- In other words: If we could ask a new student only a **single** y/n question, what should we ask.

19

HasTakenPrereqs (=: Prereq)
HasTakenACourseFromTheSameLecturer (=: Lecturer)
HasLabs

y_i= Label

consider it to be 'like'

consider it to be 'dislike'

Rating	Easy?				Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

For example, this table is data D;
Each row is a course you have rated;
$x^{(i)}$ is a sequence of 5 yes/no for the i-th course;
$y^{(i)}$ is the sign of rating for the i-th course.

Define the data $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$

$\in \{y, n\}^d$

$\in \{+, -\}$

Each dimension of $x^{(i)}$ is called a **feature**.
$x^{(i)}$ is called a **feature vector**.

20

How to Train a Tree

- Main principle: Find a tree that has a high train set accuracy

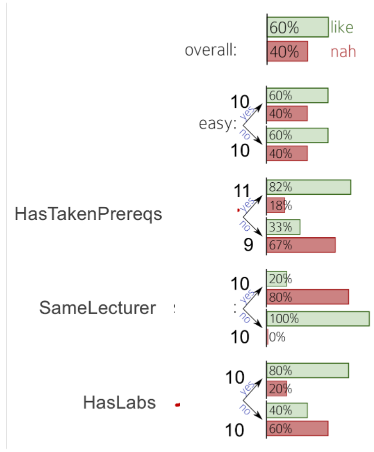
$$\widehat{acc}(f) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{I}\{f(x^{(i)}) = y^{(i)}\}$$

- This is essentially the main principle governing pretty much all the machine learning algorithms!
  - "Empirical risk minimization" principle
    (empirical risk := 1 – train_accuracy)

21

How to construct a tree: choosing root

22



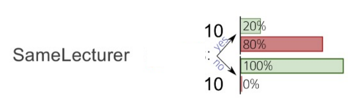
Prereqs Lecturer HasLabs

Rating	Easy?				Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

22

How to construct a tree

23



Prereqs Lecturer HasLabs

Rating	Easy?				Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

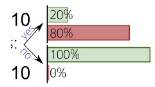
23

How to construct a tree

24

Rating	Easy?	Prereqs	Lecturer	HasLabs	Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

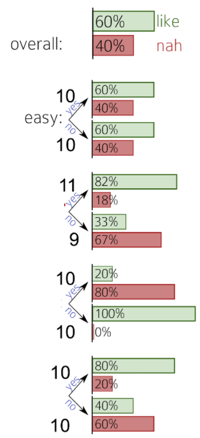
SameLecturer



24

How to construct a tree: choosing root

25



HasTakenPrereqs

SameLecturer

HasLabs

Baseline: majority vote classifier

Q: What is the train set accuracy? 12/20 = 0.60

Suppose we place the node HasTakenPrereqs at the root.
Set the prediction at each leaf node as the majority vote.

25

How to construct a tree

26

What is the train set accuracy now?

$$\frac{9}{20}\cdot\frac{6}{9}+\boxed{\frac{11}{20}}\cdot\frac{9}{11}=\frac{15}{20}=0.75$$

Accuracy for two groups:

- Prereqs = yes (11): 9/11
- Prereqs = no (9): 6/9

For the 11 people prereqs = y, use the majority vote label **like** (9 like, 2 dislike).

Predicted label for 11 people is **like**, 9 people are correctly predicted.

consider it to be 'like'

consider it to be 'dislike'

Rating	Easy?	Prereqs	Lecturer	HasLabs	Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

26

HasTakenPrereqs

N

Y

9 students did not take prereq.
6 of them did not like the course.

11 students took prereq.
9 of them liked the course.

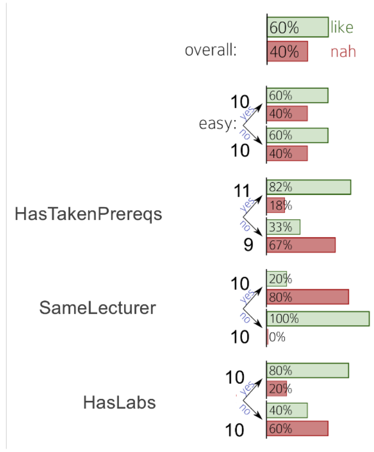
True label: +, +, +, -, -, -, -, -, -   +, +, +, +, +, +, +, +, +, -, -

Predicted label: -, -, -, -, -, -, -, -, -   +, +, +, +, +, +, +, +, +, +, +

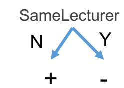
Accuracy: $\frac{9}{20}\cdot\frac{6}{9}+\frac{11}{20}\cdot\frac{9}{11}=\frac{15}{20}=0.75$

27

How to construct a tree



Suppose placing the node SameLecturer at the root.



What is the train set accuracy now?

$$\frac{10}{20}\cdot\frac{10}{10}+\frac{10}{20}\cdot\frac{8}{10}=\frac{18}{20}=0.9$$ even better!

What would you do to build a depth-1 tree?

try out each feature and choose the one that leads to the largest accuracy!

How to construct a tree

What is the train set accuracy now?

$$\boxed{\frac{10}{20}}\frac{10}{10}+\frac{10}{20}\cdot\frac{8}{10}=\frac{18}{20}=0.9$$

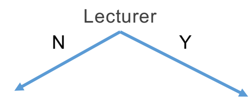
Prereqs Lecturer HasLabs

consider it to be 'like'

consider it to be 'dislike'

Rating	Easy?				Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

How to construct a tree



True label: +, +, +, +, +, +, +, +, +, +   +, +, -, -, -, -, -, -, -, -

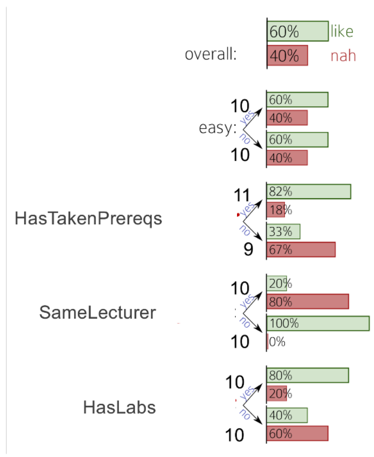
Predicted label: +, +, +, +, +, +, +, +, +, +   -, -, -, -, -, -, -, -, -, -

Accuracy: $\frac{10}{20}\cdot\frac{10}{10}+\frac{10}{20}\cdot\frac{8}{10}=\frac{18}{20}=0.9$

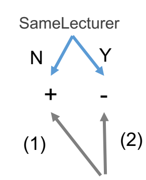
30

How to construct a tree

31



What about depth 2?



Which nodes to put at each leaf node?

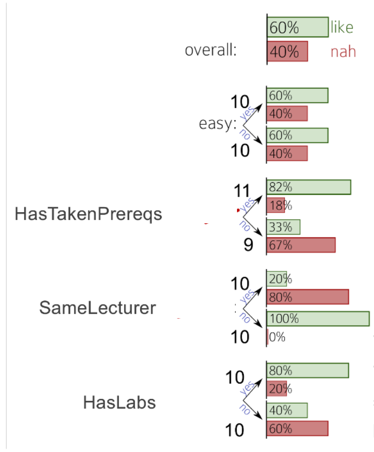
Focus on (2). Try placing HasTakenPrereqs

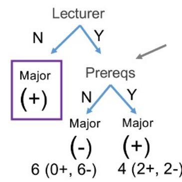
All the data on (1) has same label “like”, no need to do further splitting.

31

How to construct a tree

32





Q: How many training data points fall here? 10

Q: How many training data points arrive at these two leaves? How many for each label?

Q: what prediction should we use for each leaf?

Q: What is the train set accuracy, conditioning on SameLecturer=Y?

'local' train set accuracy

$$\frac{6}{10}\cdot\frac{6}{6}+\frac{4}{10}\cdot\frac{2}{4}=\frac{8}{10}$$

Try all the other nodes and pick the one with the largest acc.!

Then, repeat the same for SameLecturer=N branch!

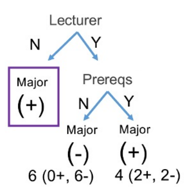
=> but this has 1 local train set acc. So leave it be!

Move onto expanding nodes at depth 2!

32

How to construct a tree

33



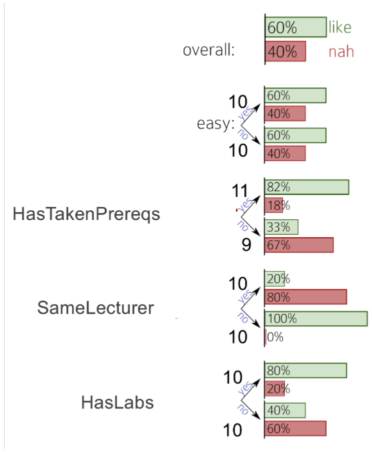
Q: What is the train set accuracy, conditioning on SameLecturer=Y?

$$\boxed{\frac{6}{10}}\cdot\frac{6}{6}+\boxed{\frac{4}{10}}\cdot\frac{2}{4}=\frac{8}{10}$$

Rating	Easy?	Prereqs	Lecturer	HasLabs	Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

33

How to construct a tree



Overall idea:

1. Set the root node as a leaf node.
2. Grab a leaf node for which its ‘local’ train accuracy is not 1.
3. Find a feature that maximizes the ’local’ train accuracy and replace the leaf node with a node with that feature; add leaf nodes and set their predictions by majority vote.
4. If local accuracy is 1, no need to split the leaf node.
5. Repeat 2-3.

34

Algorithm 1 DecisionTreeTrain(*data*, *remaining features*)

```
guess ← most frequent answer in data                    // default answer for this data
if the labels in data are unambiguous then               <= i.e., all data points have the same label
    return Leaf(guess)                                   // base case: no need to split further
else if remaining features is empty then
    return Leaf(guess)                                   // base case: cannot split further
else                                                     // we need to query more features
    for all f ∈ remaining features do                    <= there is no point in adding a feature that
                                                            appeared in its parent!
        NO ← the subset of data on which f=no
        YES ← the subset of data on which f=yes
        score[f] ← ( # of majority vote answers in NO    <= answer = label
                    + # of majority vote answers in YES ) /
                   size(data)
    end for
    f ← the feature with maximal score(f)
    NO ← the subset of data on which f=no
    YES ← the subset of data on which f=yes
    left ← DecisionTreeTrain(NO, remaining features \ {f})
    right ← DecisionTreeTrain(YES, remaining features \ {f})
    return Node(f, left, right)
end if
```

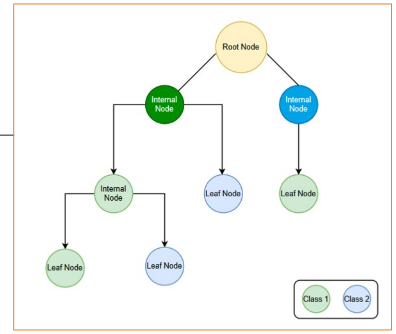
e.g {Prereq, hasLecturer…} etc

35

36

Algorithm 2 DecisionTreeTest(*tree*, *test point*)

if *tree* is of the form Leaf(*guess*) **then**
    return *guess*
else if *tree* is of the form Node(*f*, *left*, *right*) **then**
    if $f = no$ in *test point* **then**
        return DecisionTreeTest(*left*, *test point*)
    else
        return DecisionTreeTest(*right*, *test point*)
    end if
end if

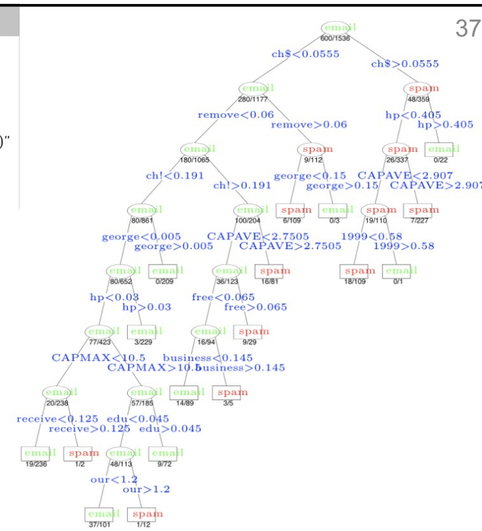


36

Example: spam filtering I

37

- Spam dataset
- 4601 email messages, about 39% are spam
- Classify message by spam and not-spam
- 57 features
    - 48 are of the form "percentage of email words that is (WORD)"
    - 6 are of the form "percentage of email characters is (CHAR)"
    - 3 other features (e.g., "longest sequence of all-caps")
- Final tree after pruning has 17 leaves, 9.3% test error rate

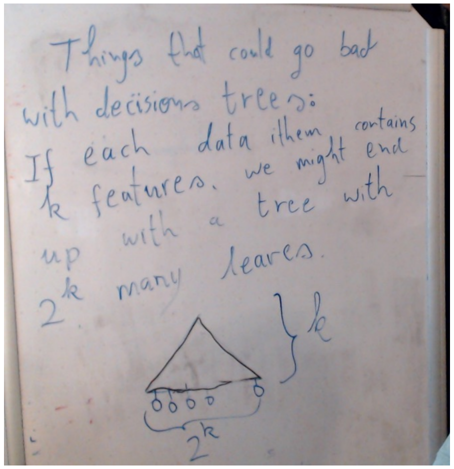


37

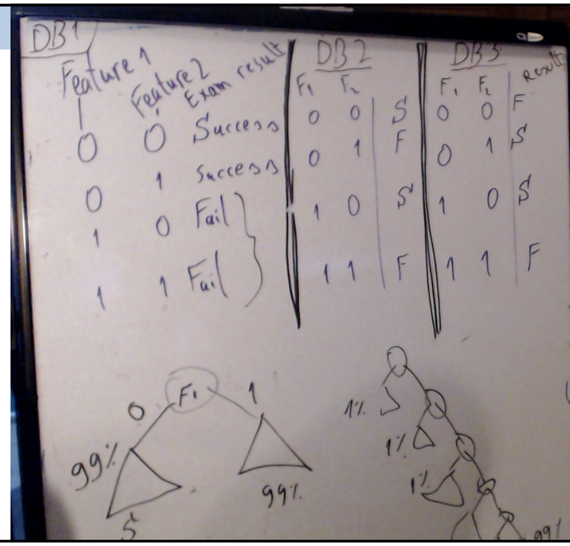
Things that could go wrong with decision trees

38

The video of the lecture for this part is in this lecture (last part)



39


DB1
Feature 1
Feature 2
Exam result
Success
Success
Fail
Fail
DB2
DB3
99%
99%
1%

40