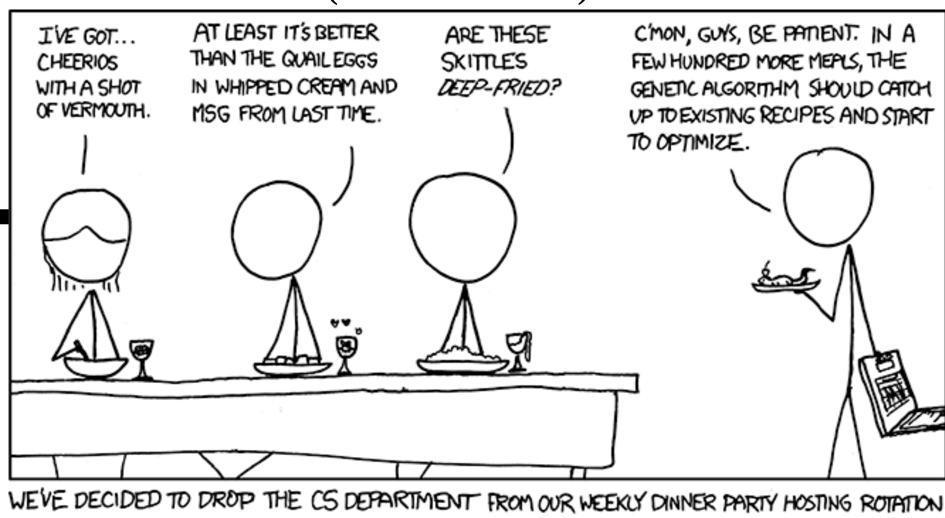
Machine Learning: Concepts

(Ch. 18.1–18.3)



PERMANENT LINK TO THIS COMIC: HTTPS://XKCD.COM/720/

1

Bookkeeping

- Today: ML 1
  - What is machine learning?
  - Classification
  - Intro to decision trees?
- Next class
  - In-class midterm review
  - A note about the midterm

2

Today's Class

- Machine learning
  - What is ML?
  - Inductive learning
- Decision trees and how to build them
- Information Gain
- Entropy
- Measuring success

Tell me about these examples

3

Why "Learn" ?

- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- There is no need to "learn" to calculate payroll
- Learning is used when:
  - Human expertise does not exist (navigating on Mars)
  - Humans are unable to explain their expertise (speech recognition)
  - Solution changes in time (routing on a computer network)
  - Solution needs to be adapted to particular cases (user biometrics)

4

What We Talk About When We Talk About "Learning"

- Learning general models from a dataset of particular examples
- **Data** is cheap and abundant (data warehouses, data marts); **knowledge** is expensive and scarce.
- Example in retail: Customer transactions to consumer behavior:
  - People who bought "Da Vinci Code" also bought "The Five People You Meet in Heaven" (www.amazon.com)
- Build a model that is a good and useful approximation to the data.

5

What is Machine Learning?

- Optimize a performance criterion using example data or past experience
- Role of Statistics: Inference from a sample
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Represent and evaluate the model for inference

6

Applications

- Association
- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
- Reinforcement Learning

7

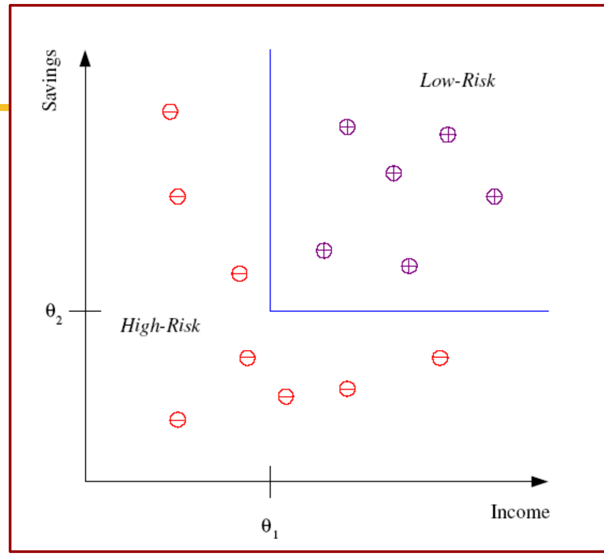
Learning Associations

- Basket analysis:
- *P (Y | X)* probability that somebody who buys *X* also buys *Y* where *X* and *Y* are products/services.
- Example: *P* (*chips* | *beer*) = 0.7

8

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their income and savings



Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

9

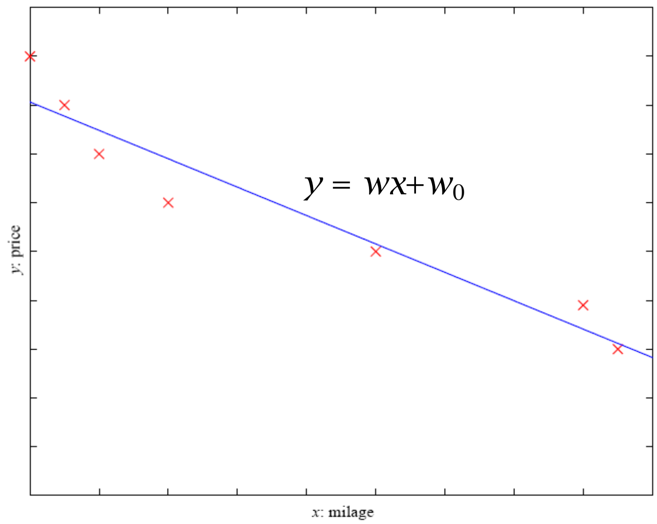
Classification: Applications

- AKA Pattern recognition
- **Face recognition:** Pose, lighting, occlusion (glasses, beard), make-up, hair style
- **Character recognition:** Different handwriting styles.
- **Speech recognition:** Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- **Medical diagnosis:** From symptoms to illnesses
- ...

10

Regression

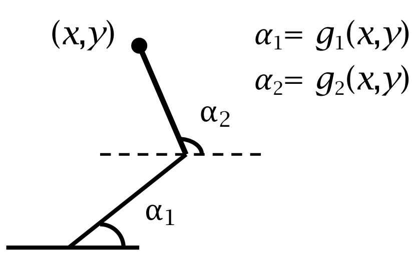
- Example: Price of a used car
  - x : car attributes
  - y : price
  - $y = g\ (x \mid \theta\ )$
  - g ( ) model,
  - θ parameters

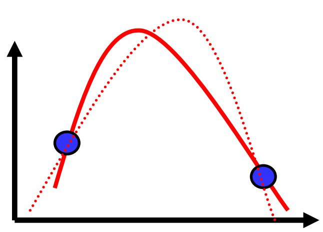


11

Regression Applications

- Navigating a car: Angle of the steering wheel (CMU NavLab)
- Kinematics of a robot arm





- Response surface design

12

Supervised Learning: Uses

- **Prediction of future cases:** Use the rule to predict the output for future inputs
- **Knowledge extraction:** The rule is easy to understand
- **Compression:** The rule is simpler than the data it explains
- **Outlier detection:** Exceptions that are not covered by the rule, e.g., fraud

13

Unsupervised Learning

- Learning "what normally happens"
- No output
- Clustering: Grouping similar instances
- Example applications
  - Customer segmentation in CRM
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

14

Reinforcement Learning

- Learning a policy: A sequence of outputs
- No supervised output but delayed reward
- Credit assignment problem
- Game playing
- Robot in a maze
- Multiple agents, partial observability, ...

15

So... What is Learning?

- "Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time."
  –Herbert Simon
- "Learning is constructing or modifying representations of what is being experienced."
  –Ryszard Michalski
- "Learning is making useful changes in our minds."
  –Marvin Minsky

16

Why Learn?

- Discover previously-unknown new things or structure
  - Data mining, scientific discovery
- Fill in skeletal or incomplete domain knowledge
- Build agents that can adapt to users or other agents
- Understand and improve efficiency of human learning
  - Use to improve methods for teaching and tutoring people (e.g., better computer-aided instruction)

17

Some Terminology

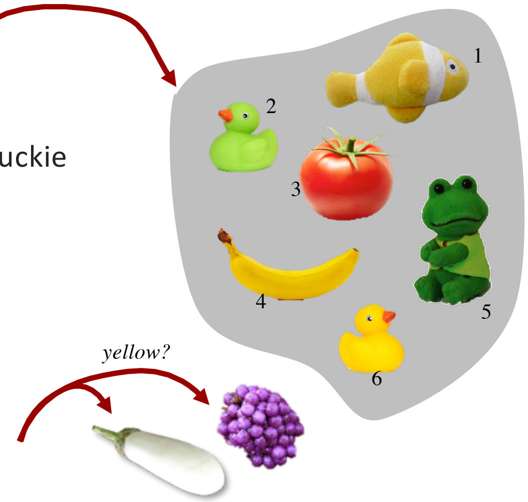
The Big Idea: given some data, you learn a model of how the world works that lets you predict new data.

- **Training Set:** Data from which you learn initially.
- **Model:** What you learn. A "model" of how inputs are associated with outputs.
- **Test set:** New data you test your model against.
- **Corpus:** A body of data. (pl.: corpora)
- **Representation:** The computational expression of data

19

ML Intro

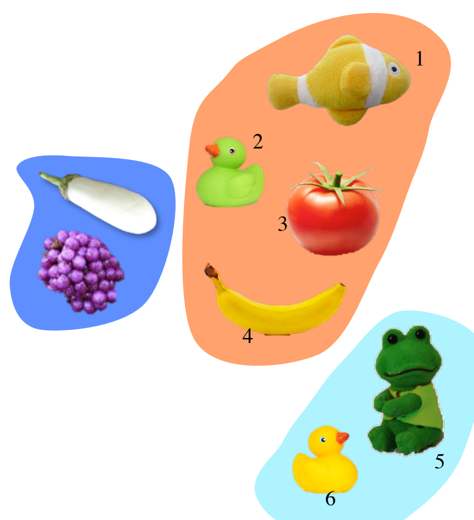
- What we have:
- **Data:** examples of our problem
  - Processed to produce **features**
    - Can't give a computer a rubber duckie
  - Turned into a feature **vector**
  - Sometimes labeled, sometimes not
- What we want:
- A **prediction** over new data



20

Learning Produces Models

- Trying to build a model of what it means to be, e.g., yellow
  - Train over data
  - Test on different data
  - Deploy: the real test
- Every step needs its own data
  - Split what we have into training data and test data to see if our learner is good



21

Questions

- What's supervised learning?
  - What's classification? What's regression?
  - What's a hypothesis? What's a hypothesis space?
  - What are the training set and test set?
  - What is Ockham's razor?
- What's unsupervised learning?

22

Machine Learning Problems

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

24

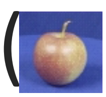
Machine Learning Problems

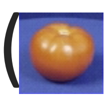
	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

27

The Machine Learning Framework

- Apply a prediction function to a feature representation of the data to get the desired output:

f() = "apple"

f() = "tomato"

f() = "cow"

Slide credit: Svetlana Lazebnik

28

The Machine Learning Framework

$$\mathbf{y} = f(\mathbf{x})$$

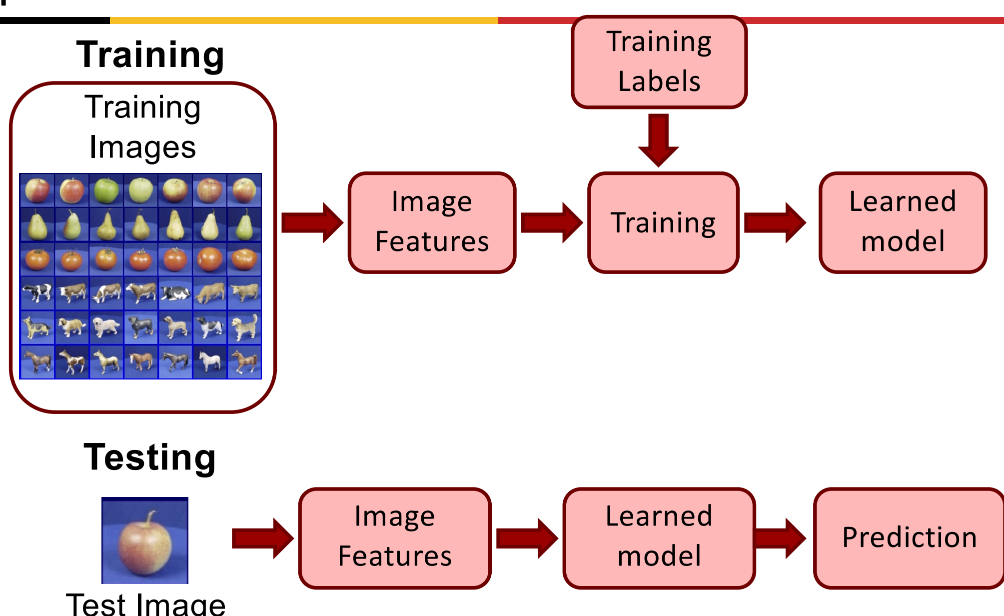
output — prediction function — feature(s) of input

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: Svetlana Lazebnik

29

Steps



Slide credit: Derek Hoiem and Svetlana Lazebnik

30

Many classifiers to choose from

- SVM
- Neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- RBMs
- Etc.

Which is the best one?

Slide credit: Derek Hoiem

31

Major Paradigms of ML (1)

- **Rote learning**: 1-1 mapping from inputs to stored representation, learning by memorization, association-based storage & retrieval
- **Induction**: Use specific examples to reach general conclusions
- **Clustering**: Unsupervised discovery of natural groups in data

32

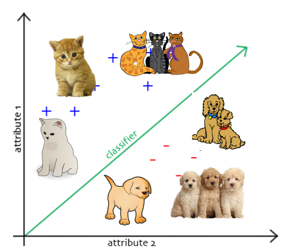
Major Paradigms of ML (2)

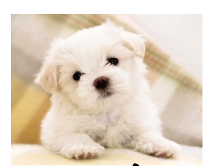
- **Analogy:** Find correspondences between different representations
- **Discovery:** Unsupervised, specific goal not given
- **Genetic algorithms:** Evolutionary search techniques, based on an analogy to survival of the fittest
- **Reinforcement:** Feedback (positive or negative reward) given at the end of a sequence of steps

33

Classification

- Classification or concept learning (aka “induction”)
  - Given a set of examples of some concept/class/category:
  - Determine if a given example is an instance of the concept (class member) or not
  - If it is: **positive example**
  - If it is not: **negative example**
  - Or we can make a probabilistic prediction (e.g., using a Bayes net)



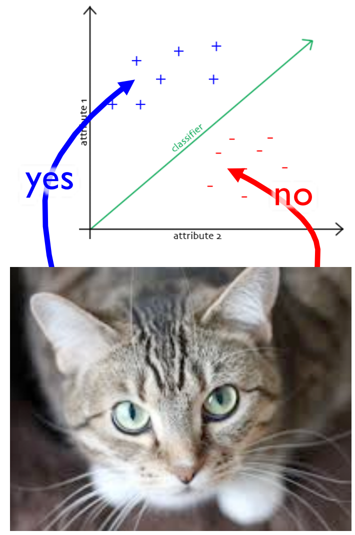


cat?

34

More on the Classification Problem

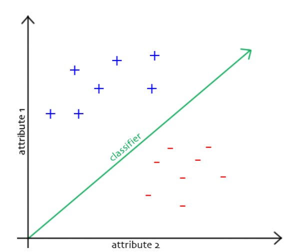
- Extrapolate from **examples** to make accurate **predictions** about future data points
  - Examples are called **training data**
- Predict into **classes**, based on attributes ("**features**")
  - Example: it has tomato sauce, cheese, and no bread. Is it pizza?
  - Example: does this image contain a cat?



35

Supervised

- Goal: Learn an unknown function $f(X) = Y$, where
  - X is an input example
  - Y is the desired output. (f is the..?)
- **Supervised learning:** given a training set of (X, Y) pairs by a "teacher"



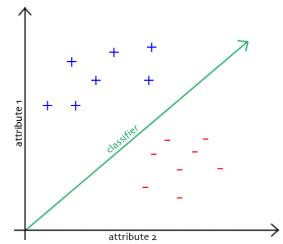
X			*Y*
bread	cheese	tomato sauce	**pizza**
¬ bread	¬ cheese	tomato sauce	**¬ not pizza**
bread	cheese	¬ tomato sauce	**pizza**
lots more rows…			

"class labels" provided

36

Unsupervised

- Goal: Learn an unknown function
  $f(X) = Y$, where
  - X is an input example
  - Y is the desired output. (f is the..?)
- **Unsupervised learning:** only given Xs and possibly some (eventual) feedback



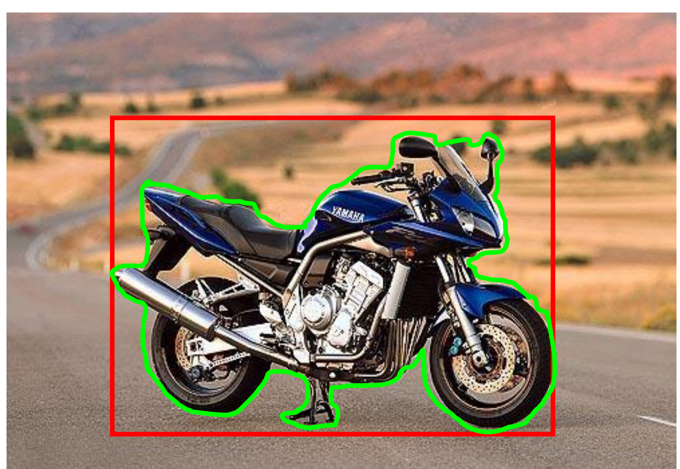
X		
bread	cheese	tomato sauce
¬ bread	¬ cheese	tomato sauce
bread	cheese	¬ tomato sauce
lots more rows…		

37

Recognition task and supervision

- Images in the training set must be annotated with the “correct answer” that the model is expected to produce

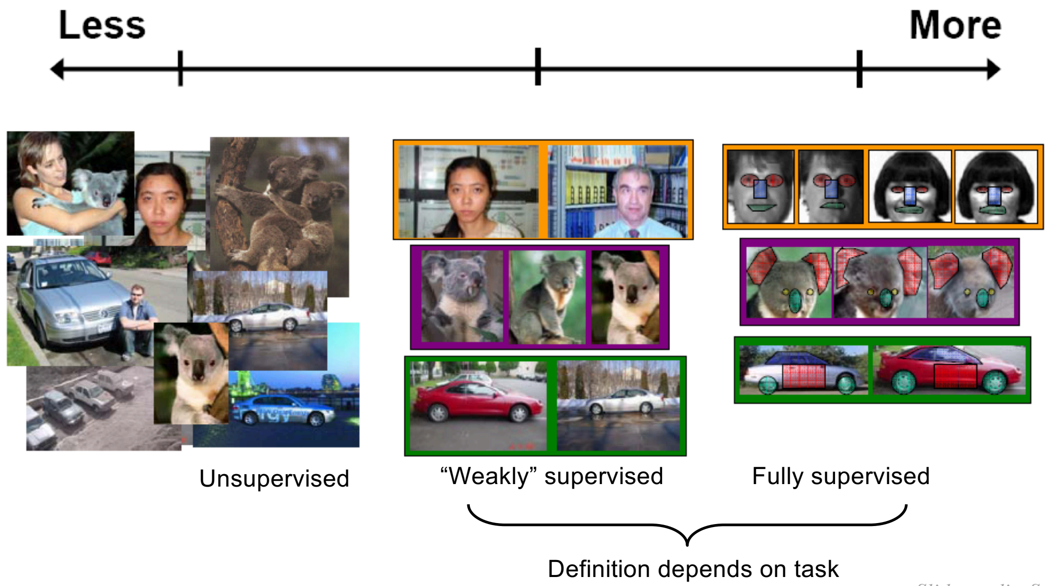
Contains a motorbike



Slide credit: Svetlana Lazebnik

38

Spectrum of Supervision



Slide credit: Svetlana Lazebnik

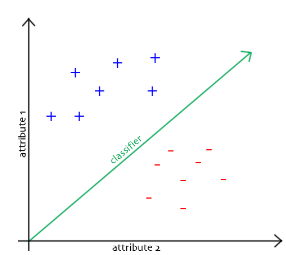
39

Supervised Concept Learning

- Given a training set of positive and negative examples of a concept
- Construct a description (model) that will accurately classify whether **future** examples are positive or negative
- I.e., learn estimate of function *f* given a training set:

  $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

  where each y_i is either + (positive) or - (negative), or a probability distribution over +/-



40

Supervised Learning

- Given training examples of inputs & outputs, produce "correct" outputs for new inputs
- Two main scenarios:
  - **Classification:** outputs whether something is in a **class** (goodRisk/badRisk, cat/notCat)
  - Learn a decision boundary that separates classes
  - **Regression** (aka "curve fitting" or "function approximation"): Learn a continuous input-output mapping from (possibly noisy) examples

41

Unsupervised Learning

Given only *unlabeled* data as input, learn some sort of structure, e.g.:

- Cluster your Facebook friends based on similarity of posts and friends
- Find sets of words whose meanings are related (e.g., doctor, hospital)
- Induce N topics and the words that are common in documents that are about that topic

42

Inductive Learning Framework

- Raw input data from sensors preprocessed to obtain **feature vector**, **X**, of **relevant** features for classifying examples
- Each **X** is a list of (attribute, value) pairs
- ***n*** attributes (a.k.a. features): fixed, positive, and finite
- Features have fixed, finite number # of possible values
  - Or continuous within some well-defined space, e.g., "age"
- Each example is a point in an *n*-dimensional feature space
  - X = [Name:Sue, EyeColor:Brown, Age:Young, Gender:Female]
  - X = [Cheese:*f*, Sauce:*t*, Bread:*t*]
  - X = [Texture:Fuzzy, Ears:Pointy, Purrs:Yes, Legs:4]

43

Inductive Learning as Search

- **Instance space, I,** is the set of all possible examples
  - Defines the **language** for the training and test instances
  - Usually each instance $i \in I$ is a **feature vector**
  - Features are also sometimes called *attributes* or *variables*

$$I: V_1 \times V_2 \times \ldots \times V_k,\ i = (v_1, v_2, \ldots, v_k)$$

- Class variable C gives an instance's class (to be predicted)

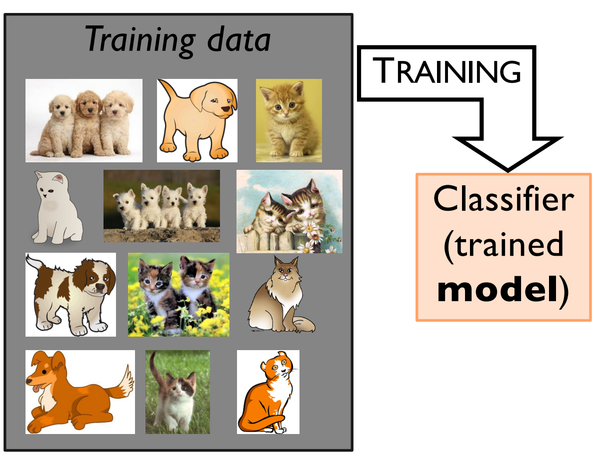
44

Inductive Learning as Search

- C gives an instance's class
- Model space M defines the possible **classifiers**
  - $M: I \rightarrow C$, $M = \{m_1, \ldots m_n\}$ (possibly infinite)
  - Model space is sometimes defined using same features as instance space (not always)
- Training data lets us search for a good (consistent, complete, simple) hypothesis in the model space
- The learned model is a *classifier*

45

Inductive Learning Pipeline

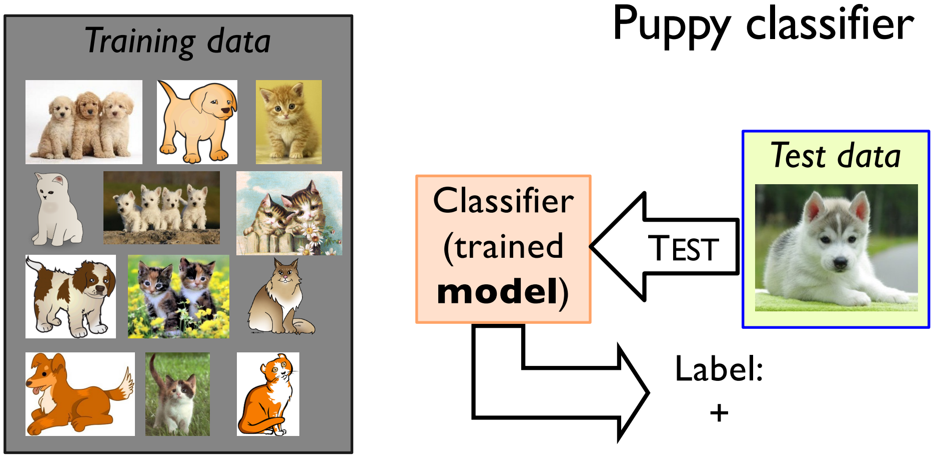


Puppy classifier

46

Inductive Learning Pipeline

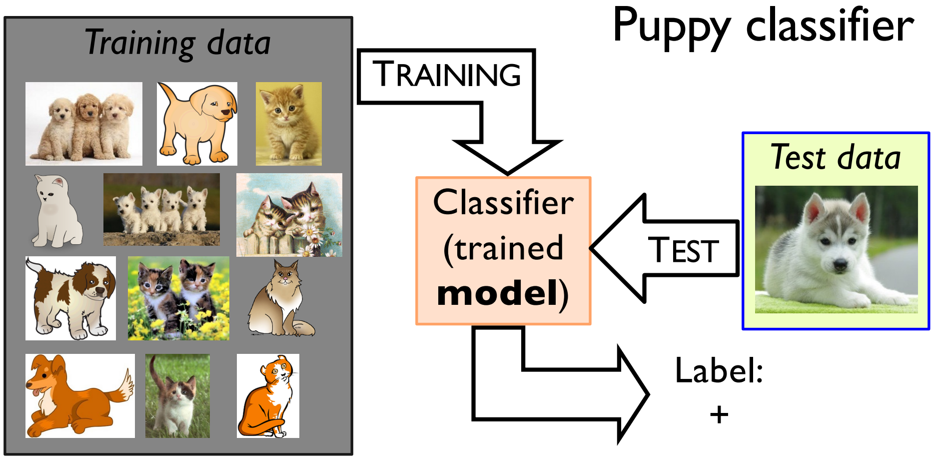
Puppy classifier



47

Inductive Learning Pipeline

Puppy classifier



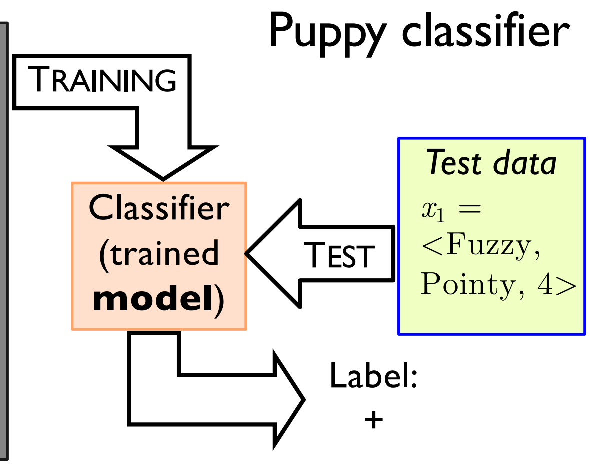
48

Inductive Learning Pipeline

Puppy classifier

Training data, X

Text-ure	*Ears*	*Legs*	*Class*
Fuzzy	Round	4	+
Slimy	Missing	4	-
Fuzzy	Pointy	4	-
Fuzzy	Round	4	+
Fuzzy	Pointy	4	+
…			



49

Model Spaces (1)

- Decision trees
  - Partition the instance space I into axis-parallel regions
  - Labeled with class value
- Nearest-neighbor classifiers
  - Partition the instance space I into regions defined by centroid instances (or cluster of *k* instances)
- Bayesian networks
  - Probabilistic dependencies of class on attributes
  - Naïve Bayes: special case of BNs where class → each attribute

50

Model Spaces (2)

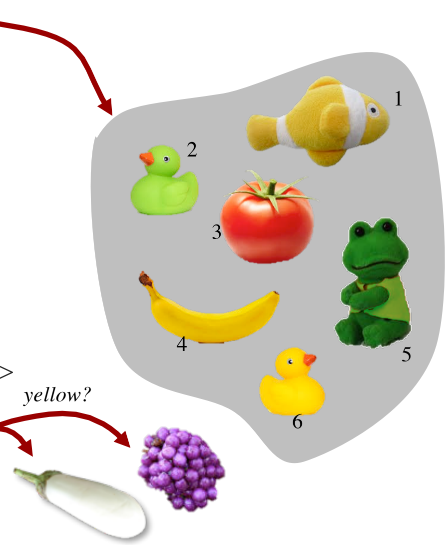
- Neural networks
  - Nonlinear feed-forward functions of attribute values
  - Can be ”deep”
  - Much learning today falls under neural approaches
- Support vector machines
  - Find a separating plane in a high-dimensional feature space
- Associative rules (feature values → class)
- First-order logical rules

51

Summary: ML Overview

What we have:

- **Data:** examples of our problem
  - Processed to produce **features**
    - Average R, G, B values of pixels
    - Fuzzy or not fuzzy
  - Turned into a **feature vector**
    - $X_1: <200, 200, 40, \text{yes}>$ …
    - $X_3: <220, 10, 22, \text{no}>$ …
  - Sometimes labeled, sometimes not
    - $X_1: <200, 200, 40, \text{yes}, \text{yellow}=\text{yes}>$



What we want:

- A prediction over new data
  - $X_7: <240, 240, 240, \text{no}, \text{yellow}=??>$

52

Summary: Machine Learning 1

- Core idea: given (possibly labeled) training data, learn a **model** of how the world works that lets you make **predictions** about new observations at test time
- Supervised vs. unsupervised, continuous vs. discrete
- Supervised learning over discrete data = classification
- Decision trees are one approach for discrete data

53

Decision Trees (DTs)

- A supervised learning method used for classification and regression
- Given a set of training tuples, learn model to predict one value from the others
  - Learned value typically a class (e.g. Puppy)
- Resulting model is simple to understand, interpret, visualize and apply

54

Decision Trees

- Goal: Build a tree to classify examples as positive or negative instances of a concept using supervised learning from a training set
- A decision tree is a tree where:
  - Each **non-leaf** node is an attribute (feature)
  - Each **leaf** node is a classification (+ or -)
    - Positive and negative data points
  - Each **arc** is one possible value of the attribute at the node from which the arc is directed
- Generalization: allow for >2 classes
  - e.g., {sell, hold, buy}

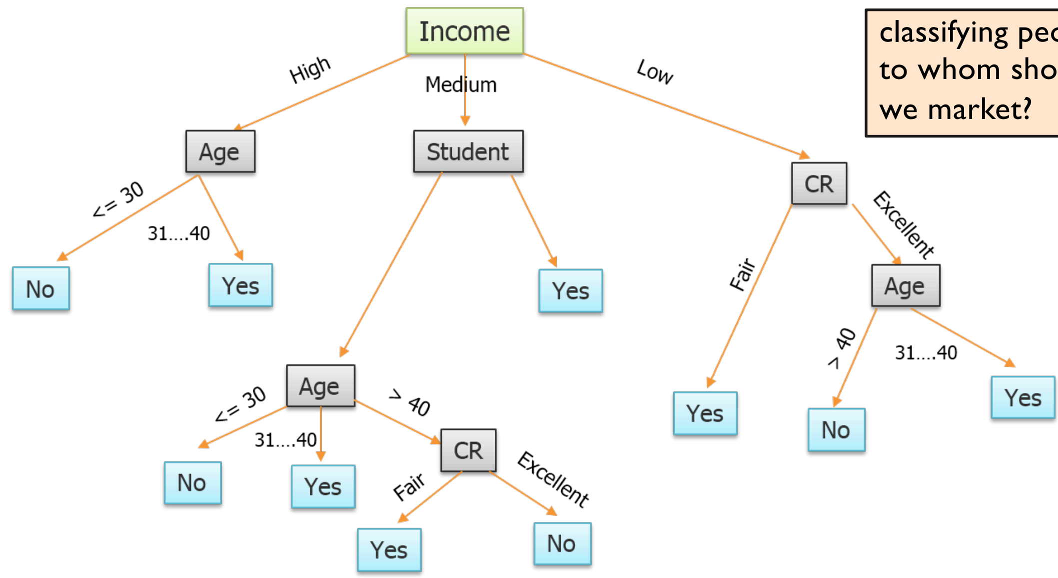
55

Decision Tree Induction

- The Big Idea: build a tree of **decisions**, each of which splits training data into smaller groups
  - Very common machine learning technique!
- At each split, an attribute of the training data – a **feature** – is chosen to divide data into classes
- Goal: each leaf group in the tree consists entirely of one class
- Learning: creating that tree

56

Will You Buy My Product?



classifying people: to whom should we market?

http://www.edureka.co/blog/decision-trees/

57

Let's Talk Features

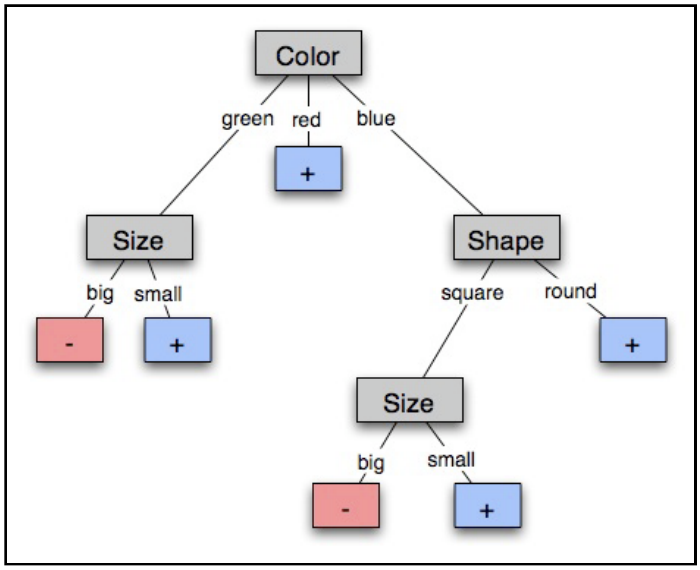
Class label

Object	Yellow?	RGB	Fuzzy?
Duckie1	N	0,255,0	N
Fish	Y	240,240,0	Y
Tomato	N	250,0,0	N
Banana	Y	255,230,0	hope not
Duckie2	Y	250,255,0	N
Frog	N	0,120,0	Y

58

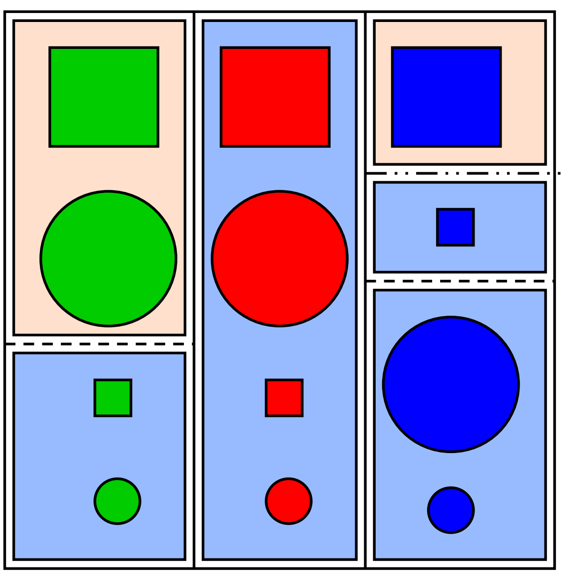
Learning Decision Trees

- Each **non-leaf** node is an attribute (feature)
- Each **arc** is one value of the attribute at the node it comes from
- Each **leaf** node is a classification (+ or -)



59

Learning a Concept



The red groups are negative examples, blue positive

Features

- Size: large, small
- Color: red, green, blue
- Shape: square, circle

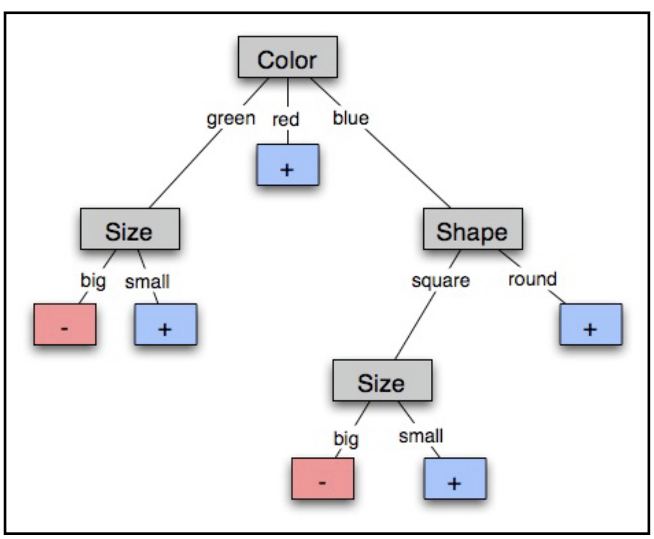
60

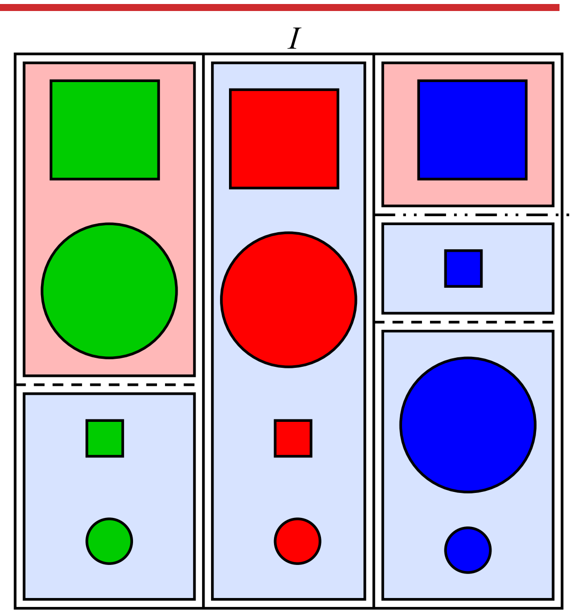
Training Data

Size	Color	Shape	class
Large	Green	Square	Negative
Large	Green	Circle	Negative
Small	Green	Square	Positive
Small	Green	Circle	positive
Large	Red	Square	Positive
Large	Red	Circle	Positive
Small	Red	Square	Positive
Small	Red	Circle	Positive
Large	Blue	Square	Negative
Small	Blue	Square	Positive
Large	Blue	Circle	Positive
Small	Blue	Circle	Positive

61

Decision Tree-Induced Partition – Example



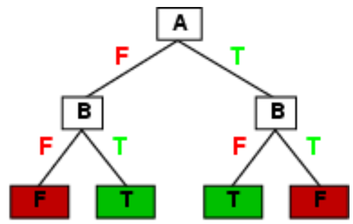


62

Expressiveness

- Decision trees can express any function of the input attributes.
- E.g., for Boolean functions, truth table row → path to leaf:

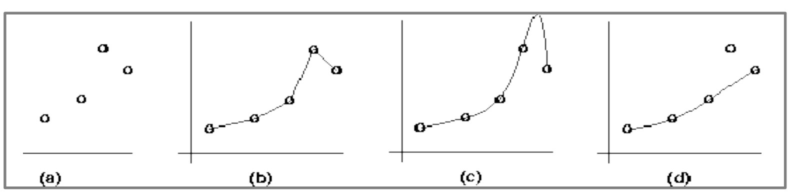
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



- Trivially, there is a consistent decision tree for any training set with one path to leaf for each example (unless f nondeterministic in x) but it probably won't generalize to new examples
- We prefer to find more compact decision trees!

63

Inductive Learning and Bias



- Want to learn a function f(x) = y
- Given sample (x,y) pairs, as in (a)
- There are several possible hypotheses (b-d)
- Preferring one shows the bias of our learning technique:
  - Prefer piece-wise functions? (b)
  - Prefer a smooth function? (c)
  - Prefer a simple function and treat outliers as noise? (d)

64

Preference Bias: Ockham's Razor

- A.k.a. Occam's Razor, Law of Economy, or Law of Parsimony
- Stated by William of Ockham (1285-1347/49):
  - "*Non sunt multiplicanda entia praeter necessitatem*"
  - "Entities are not to be multiplied beyond necessity"
- **"The simplest consistent explanation is the best."**
- Smallest decision tree that correctly classifies all training examples
- Finding the provably smallest decision tree is NP-hard!
- So, instead of constructing the absolute smallest tree consistent with the training examples, construct one that is "pretty small"

65

R&N's Restaurant Domain

- Model the decision a patron makes when deciding whether to wait for a table or leave the restaurant
  - Two classes (outcomes): **wait, leave**
  - Ten attributes:
    - Alternative available? ∃ Bar? Is it Friday? Hungry? How full is restaurant? How expensive? Is it raining? Do we have a reservation? What type of restaurant is it? What's purported waiting time?
- Training set of 12 examples
- ~ 7000 possible cases

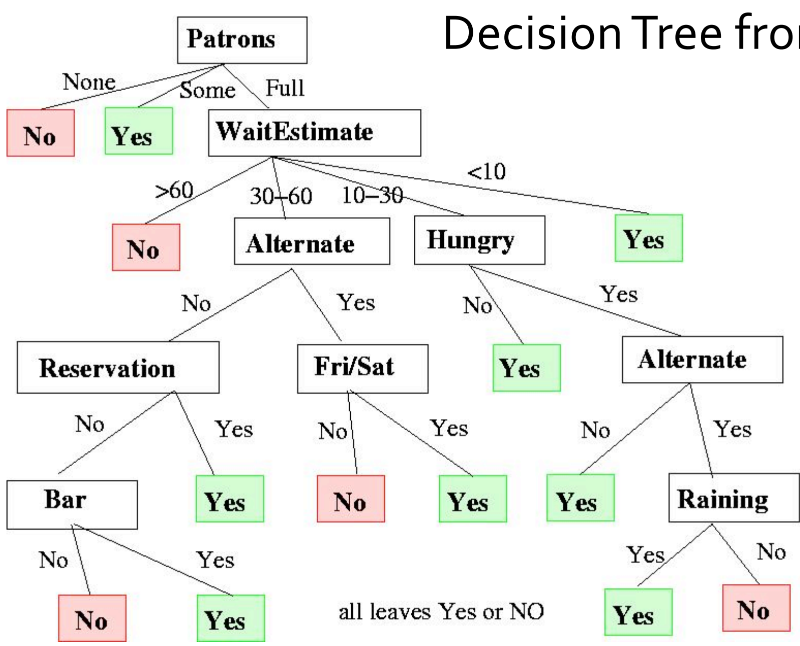
66

A Training Set

Datum	Attributes										Outcome (Label)
	altern-atives	bar	Friday	hungry	people	$	rain	reser-vation	type	wait time	Wait?
X_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
X_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
X_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
X_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	Yes
X_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
X_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
X_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
X_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	0-30	No
X_{11}	No	No	No	No	None	$	No	No	Thai	0-10	No
X_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

67

Decision Tree from Inspection



Problem from R&N, table from Dr. Manfred Kerber @ Birmingham, with thanks – www.cs.bham.ac.uk/~mmk/Teaching/AI/l3.html

68

Issues

- It’s like 20 questions:
- We can generate many decision trees depending on what attributes we ask about and in what order
- How do we decide?
- What makes one decision tree better than another: number of nodes? number of leaves? maximum depth?

69

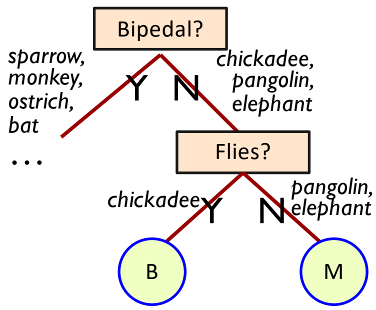
ID3/C4.5

- A **greedy** algorithm for decision tree construction
  - Ross Quinlan, 1987
- Construct decision tree top-down by recursively selecting the “best attribute” to use at current node
  - Select attribute for current node
  - Generate child nodes (one for each possible value of attribute)
  - Partition training data using attribute values
  - Assign subsets of examples to the appropriate child node
  - Repeat for each child node until all examples associated with a node are either all positive or all negative

70

Bird or Mammal?

1. Select attribute
2. Generate child nodes
3. Partition examples
4. Assign examples to child
5. Repeat until examples are +ve or -ve



Examples (training data)	Attributes			Outcome
	Bipedal	Flies	Feathers	
Sparrow	Y	Y	Y	B
Monkey	Y	N	N	M
Ostrich	Y	N	Y	B
Pangolin	N	N	N	M
Bat	Y	Y	N	M
Elephant	N	N	N	M
Chickadee	N	Y	Y	B

<u>Test</u>

mouse: <B:N, Fl:N, Fe:N>

71

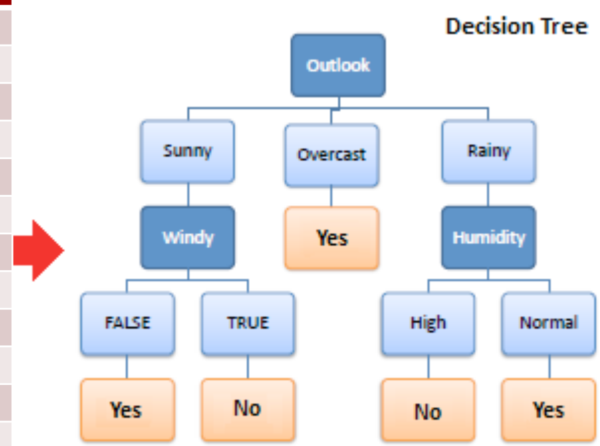
	Outlook	Temp	Humidity	Windy	Play golf?
1	Rainy	Hot	High	False	No
2	Rainy	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Sunny	Mild	High	False	Yes
5	Sunny	Cool	Normal	False	Yes
6	Sunny	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Rainy	Mild	High	False	No
9	Rainy	Cool	Normal	False	Yes
10	Sunny	Mild	Normal	False	Yes
11	Rainy	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Sunny	Mild	High	True	No

www.saedsayad.com/decision_tree.htm

72

Exercise: draw a decision tree

Outlook	Temp	Humidity	Windy	Play golf?
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



www.saedsayad.com/decision_tree.htm

73