Machine Learning
Lecture 1

Sardar Hamidian

The George Washington University
Department of Computer Science

Logistics

Meeting time:	**Fridays 4:10-6:40**
Location:	Tompkins 201
Instructors:	**Sardar Hamidian**
E-mails:	sardar@gwu.edu
Office hours:	By appointment.

Syllabus

- **No cheating!**

- **Religious holidays**
  - Tell us in advance when you will be off.

- **Disability**
  - Please bring the written statement on how to accommodate your needs.

Syllabus

- **Course schedule and organization:**

  40% 4 Projects (every 2 weeks)

  40% 2 Midterm Exams

  20% Final Project (Groups of up to 3)

  Class activity is important
- **Final grade criteria will not be harsher than:**

  90% --> A- or better

  80% --> B- or better

  70% --> C- or better


machine
learning
data
algorithms
training
edit
model
article
main
systems
representation
performance
show
inputs
methods
knowledge
examples
algorithm
mining
statistical
network
dictionary
set
theory
field
used
sparse
programming
computer
method
tasks
biases
also
example
reinforcement
unsupervised
artificial
neural
known
problem
machines
models
classification
system
networks
statistics
supervised
often
learn
ability
trained
rule
given
logic
learned
time
language
function
related
computational
two
analysis
predictions
bias
problems
one
use
previously
distribution
new
ai
many
recognition
complexity
different
vector
find
approach
represented
health
learner
fields
bayesian
unknown
discovery
intelligence

What is machine learning?

- “Learning is any process by which a system improves performance from experience.” - HS

- “Machine learning is concerned with computer programs which improve performance through experience.” - HS

Herbert Simon

Turing Award - 1975

Nobel Prize in Economics - 1978

What is machine learning?

- Machine learning is a huge topic
  - Computational biology
  - Computer vision
  - Natural language processing
  - Robotics

- In this course we will study:
  - Supervised and unsupervised learning algorithms
  - Deep learning
  - Data preprocessing and feature selection
  - Parameters and similarity metrics
  - Kernels

Why is machine learning so popular?

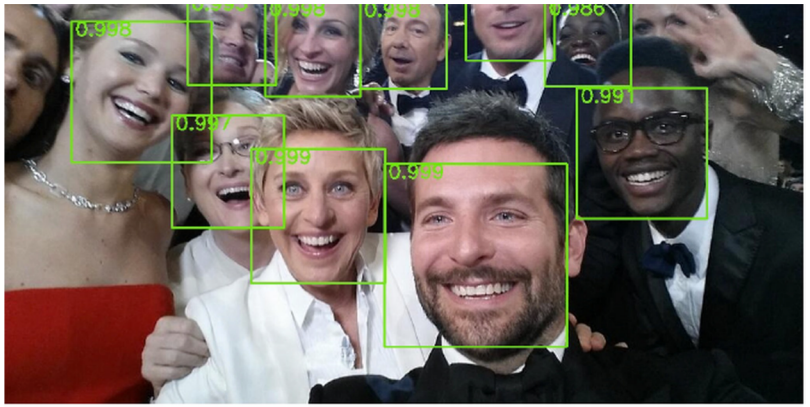
- Flood of available data
- Computational power
- Advances in algorithms and theory
- Support from industry and research foundations

Machine learning applications - examples

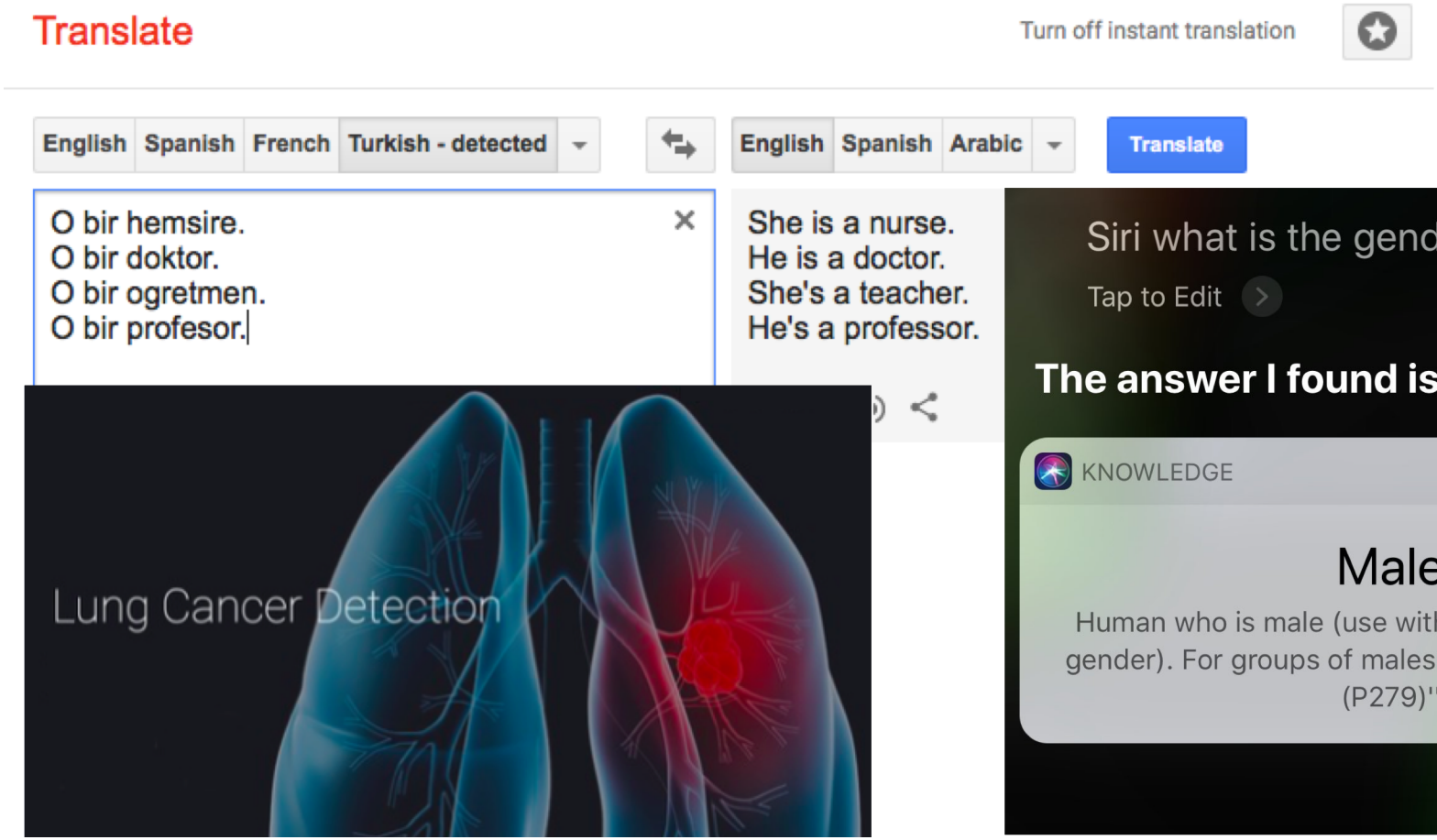
NETFLIX

2σ

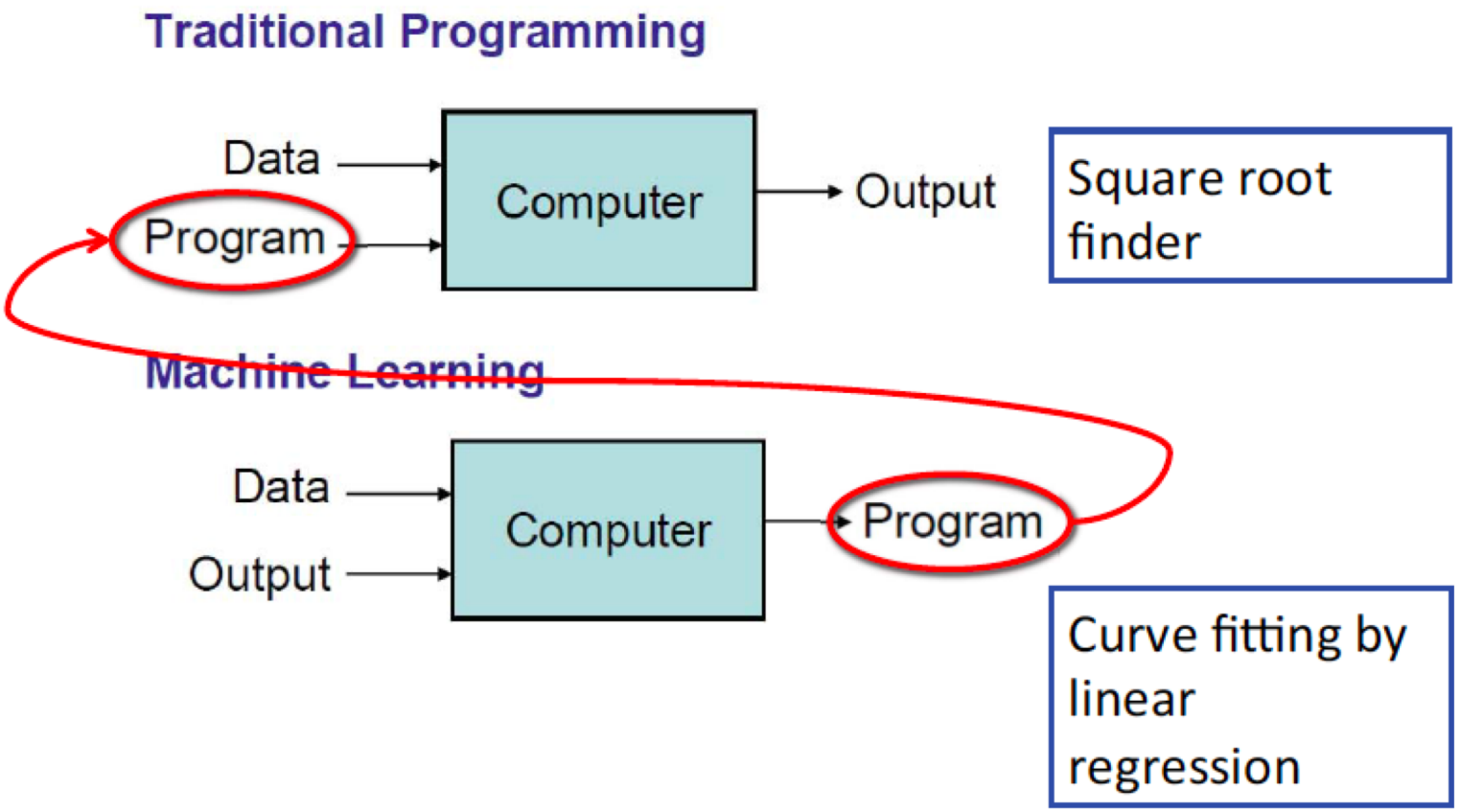
TWO SIGMA
VENTURES



Machine learning applications - examples



What is machine learning?



How are things learned?

- **Memorization**
  - Declarative knowledge
    - Accumulation of individual facts
    - Limitation: time to observe facts and memory for fact storage
- **Generalization**
  - Imperative knowledge
    - Deduce new facts from old facts
    - Assumption: past predicts the future
    - Limitation: accuracy of deduction process
- **Goal**
  - Extend to programs that can infer useful information from **implicit** patterns in data

Machine learning setup

Let us formalize the supervised machine learning setup. Our training data comes in pairs of inputs ($\mathbf{x}$, y), where $\mathbf{x} \in \boldsymbol{R}^{\mathbf{d}}$ is the input instance and y its label. The entire training data is denoted as:

$$\mathbf{D}=\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_n,y_n)\} \subseteq \boldsymbol{R}^{\mathbf{d}} \times \boldsymbol{C}$$

where:

$\boldsymbol{R}^{\mathbf{d}}$:	d-dimensional feature space
$\mathbf{x}_i$:	input vector of the i^{th} sample
y_i:	label of the i^{th} sample
$\boldsymbol{C}$:	label space

Machine learning setup

- Binary classification, i.e., $\boldsymbol{C} = \{0, 1\}$ or $\boldsymbol{C} = \{-1, +1\}$.
- Multi-class classification, i.e., $\boldsymbol{C} = \{1, 2, \cdots, K\}$ ($K \geq 2$).
- Regression, i.e., $\boldsymbol{C} = \mathbb{R}$ (Regression can be used for prediction and both use continuous real numbers instead of discrete classes.)

Machine learning setup

There are multiple scenarios for the label space $\boldsymbol{C}$:

Example-1: (Two-class (binary) classification) - Spam filtering.

An email is either a spam (+1), or not (−1).

Example-2: (Multi-class classification) - Face classification.

A person can be exactly one of $\boldsymbol{K}$ identities (e.g., 1="Barack Obama", 2="Donald Trump", …, *K*=”Sardar Hamidian”).

Example-3: (Prediction (for continuous variables)

Predict future temperature or the height of a person.

The goal of **supervised learning** is to find a function h: $\boldsymbol{R}^{\mathbf{d}} \rightarrow \boldsymbol{C}$, such that $h(\mathbf{x}_i) \approx y_i$ for all $(\mathbf{x}_i , y_i) \in \mathbf{D}$ (training); $h(\mathbf{x}_i) \approx y_i$ for all $(\mathbf{x}_i , y_i) \notin \mathbf{D}$ (testing).

Basic paradigm

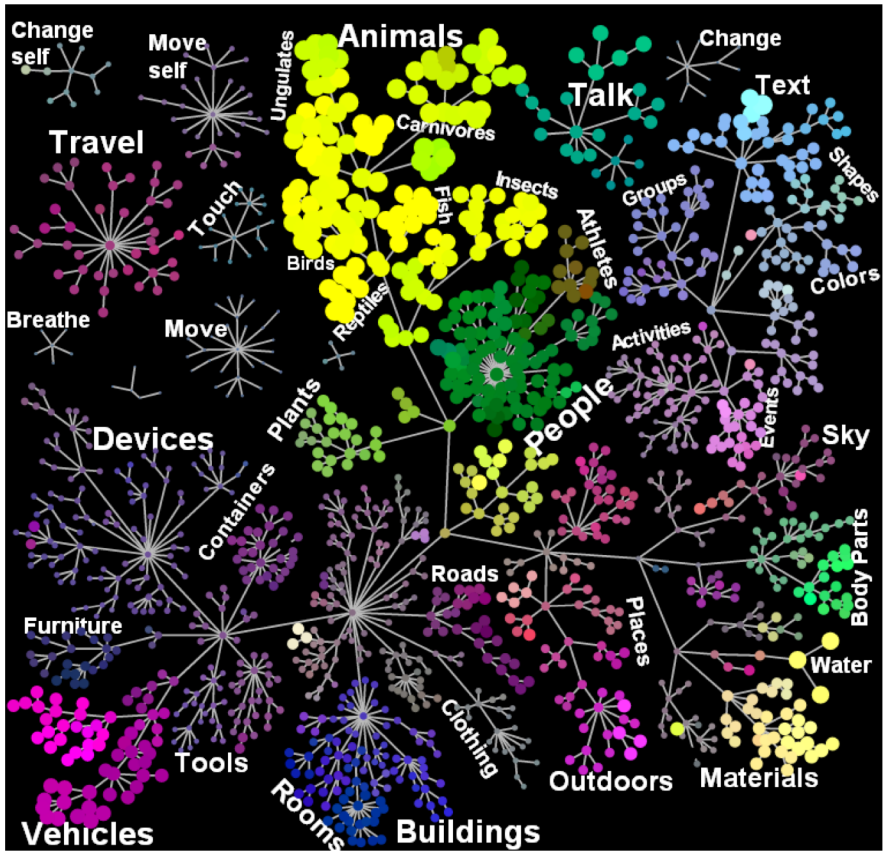
- Training data: Observe set of examples
- Train a model: Infer something about process that generated that training data
- Test data: Use inference to make predictions about previously unseen data

Basic paradigm variations

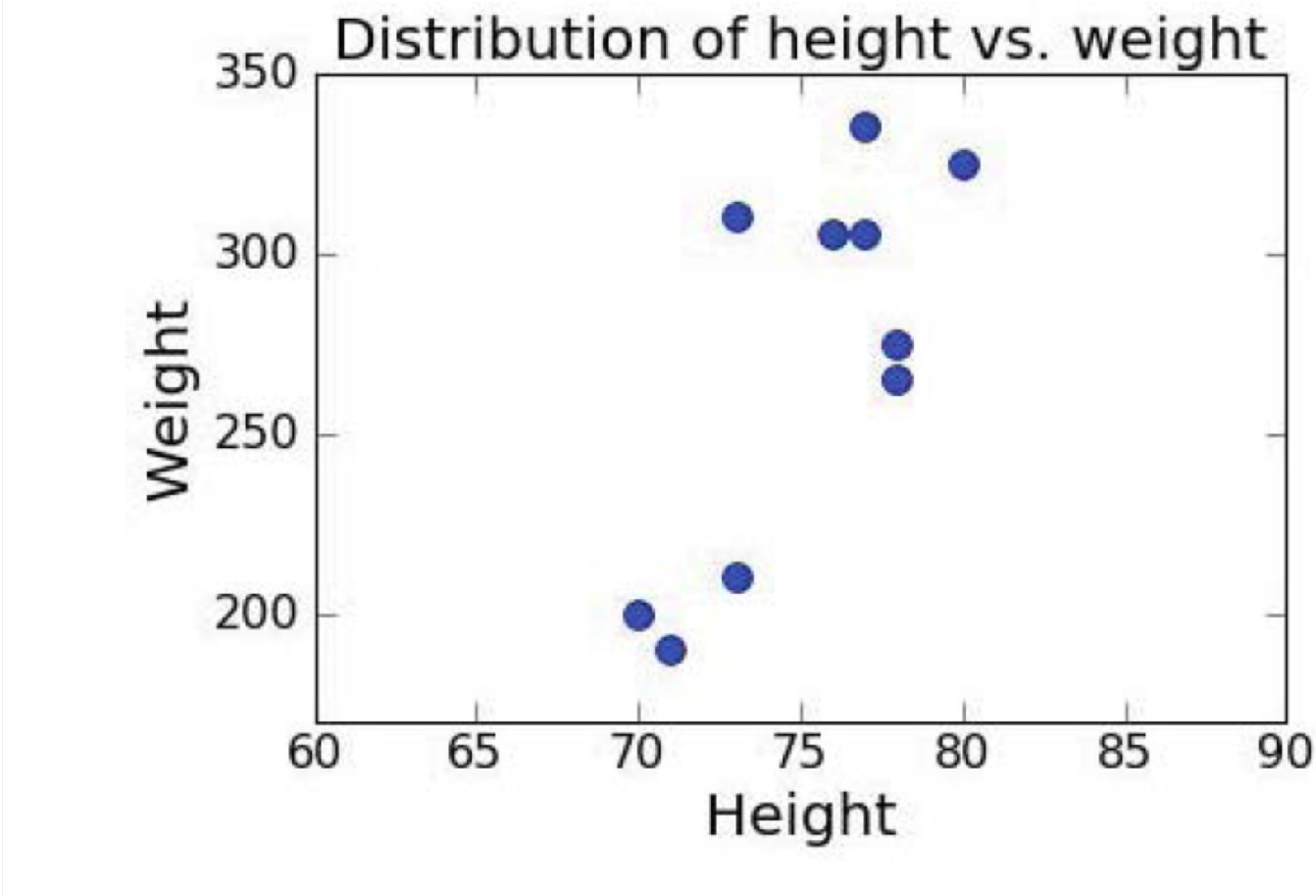
- Supervised learning: given a set of feature/label pairs, find a rule that predicts the label associated with a previously unseen input
- Unsupervised learning: given a set of feature vectors (without labels) group them into “natural clusters” (or create labels for groups)

Classification example: object recognition

Clustering example: semantic space



Unlabeled data



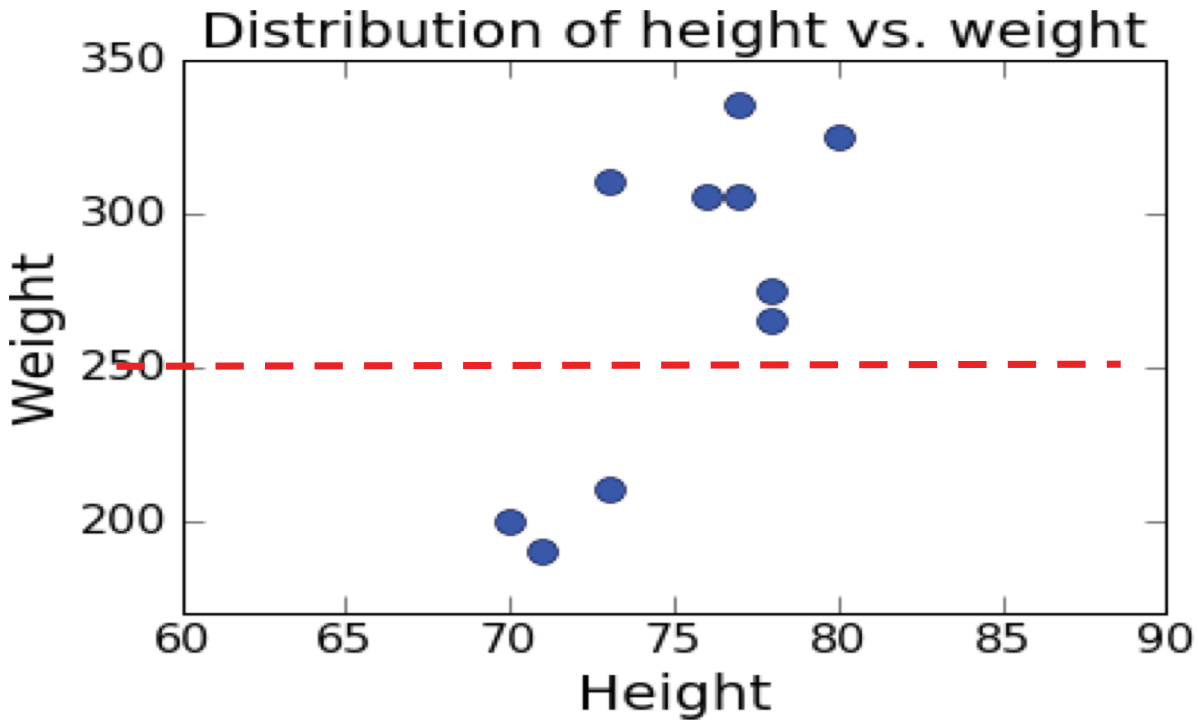
Clustering examples into groups

- Want to decide on “similarity” of examples, with goal of separating into distinct, “natural”, groups
  - Similarity is a distance measure

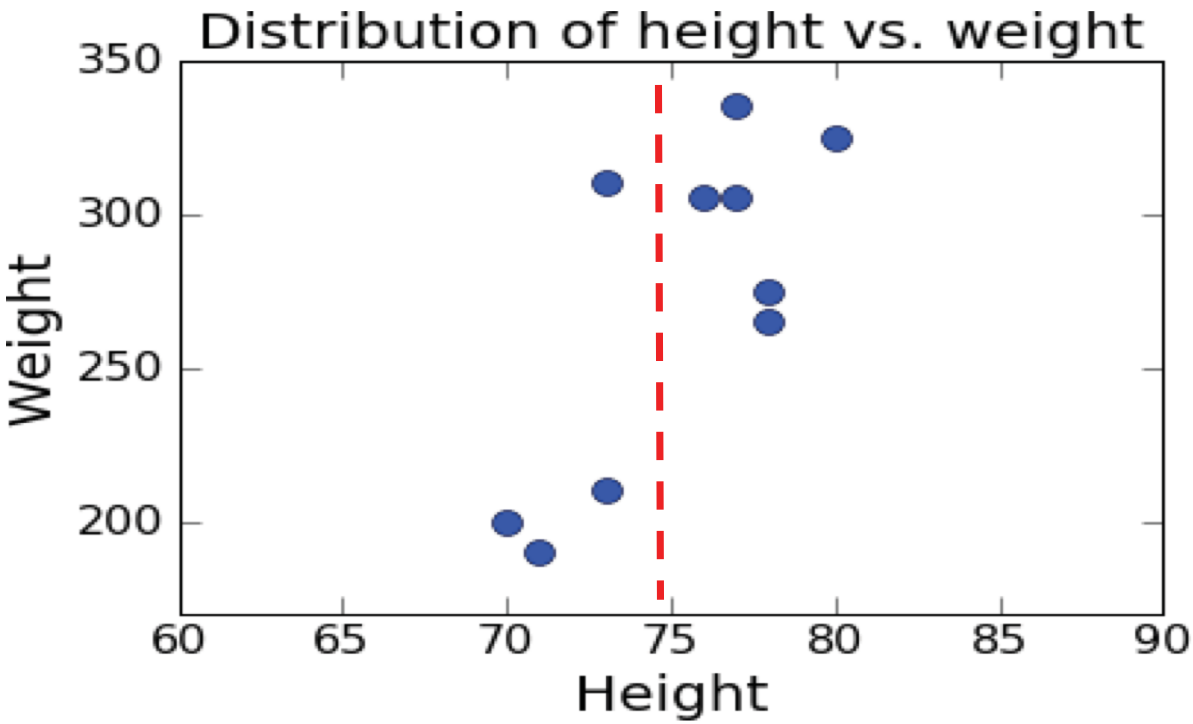
Clustering examples into groups

- Suppose we know that there are k different groups in our training data, but don't know labels (here k = 2)
  - Pick k samples (at random?) as exemplars
  - Cluster remaining samples by minimizing distance between samples in same cluster (objective function) – put sample in group with closest exemplar
  - Find median example in each cluster as new exemplar
  - Repeat until no change

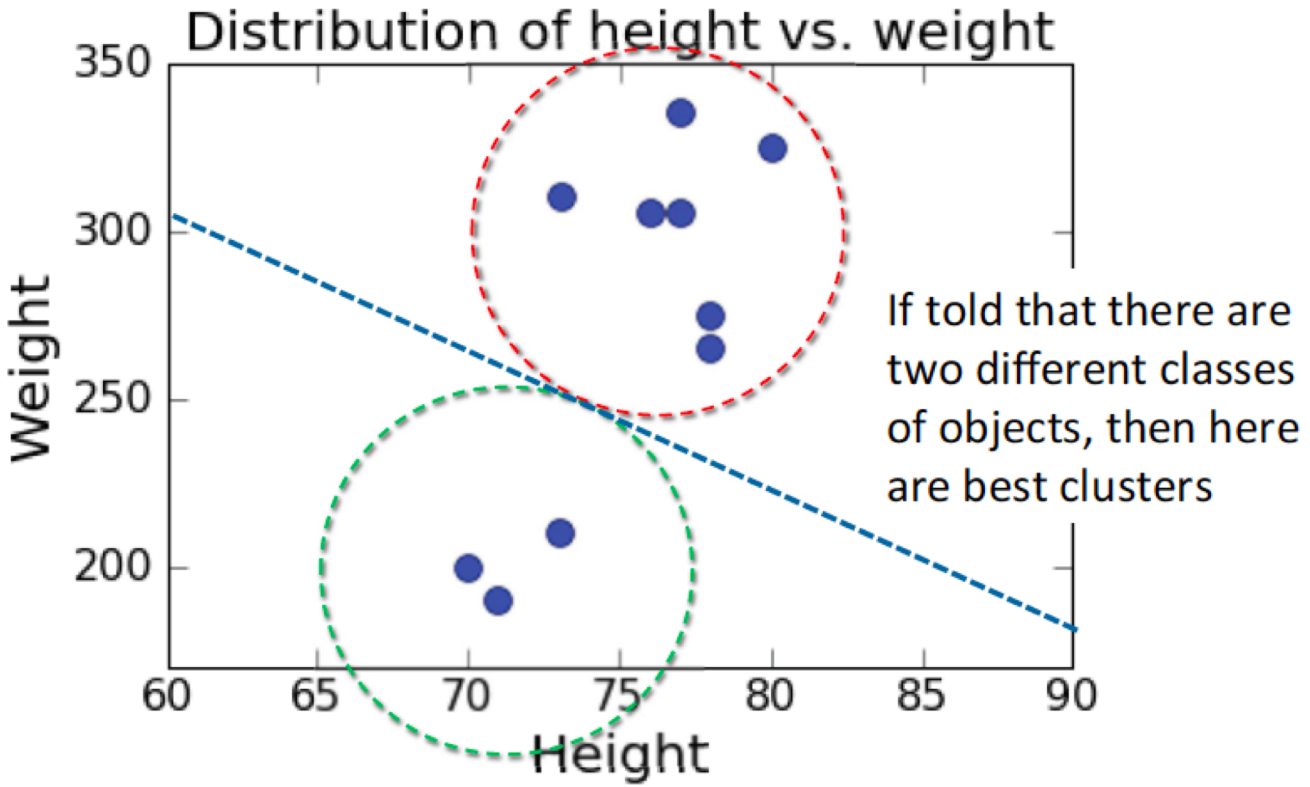
Similarity based on weight



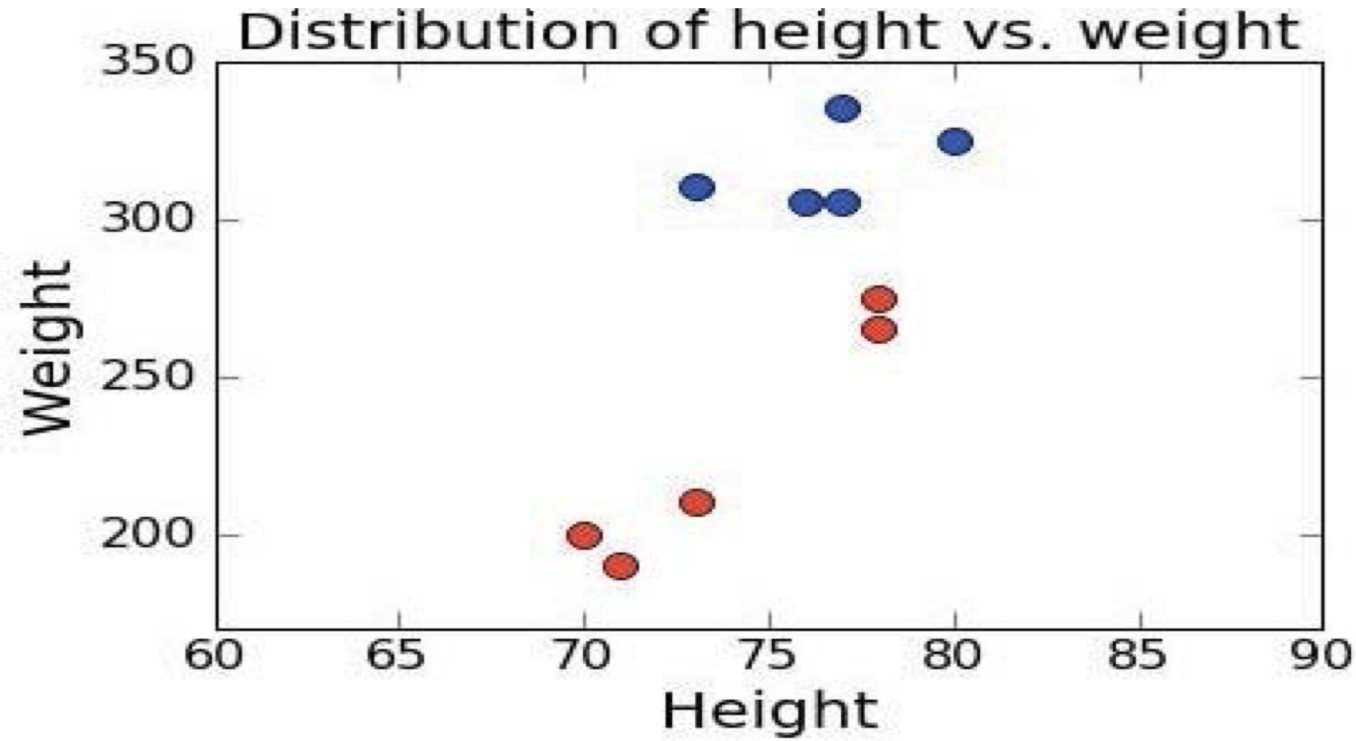
Similarity based on height



Cluster into Two Groups Using Both Attributes



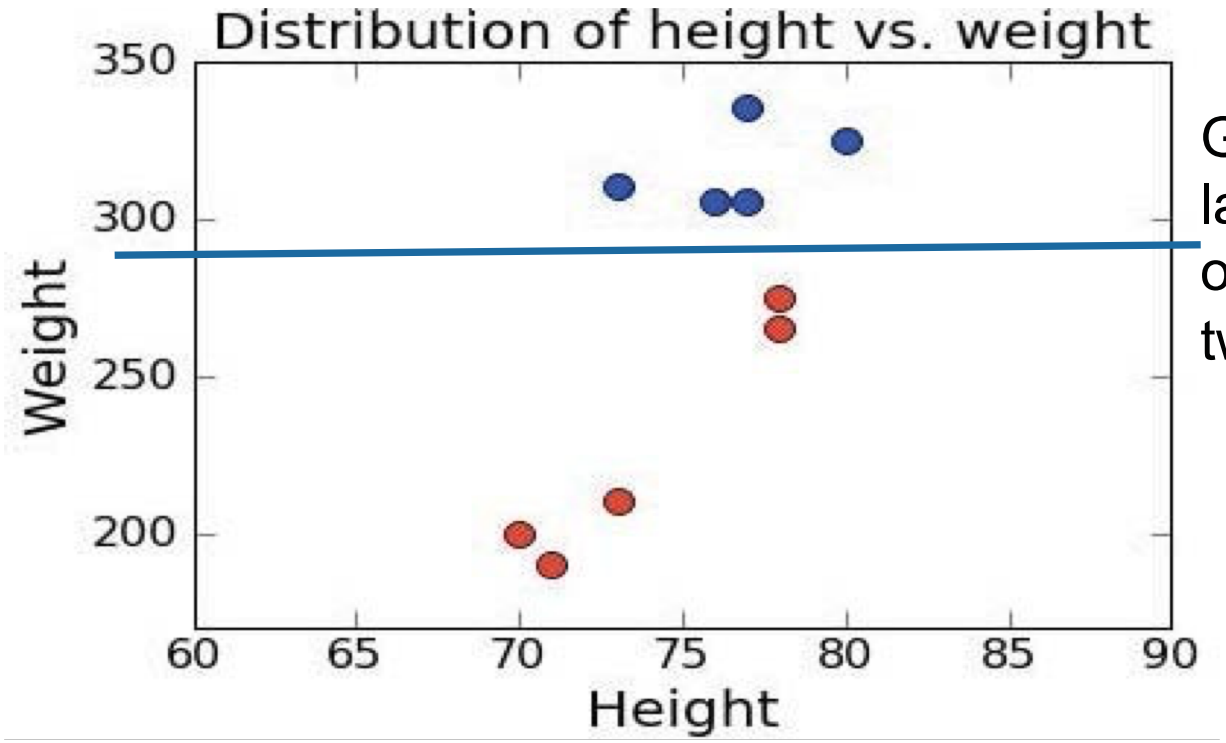
Suppose Data Was Labeled



Finding Classifier Surfaces

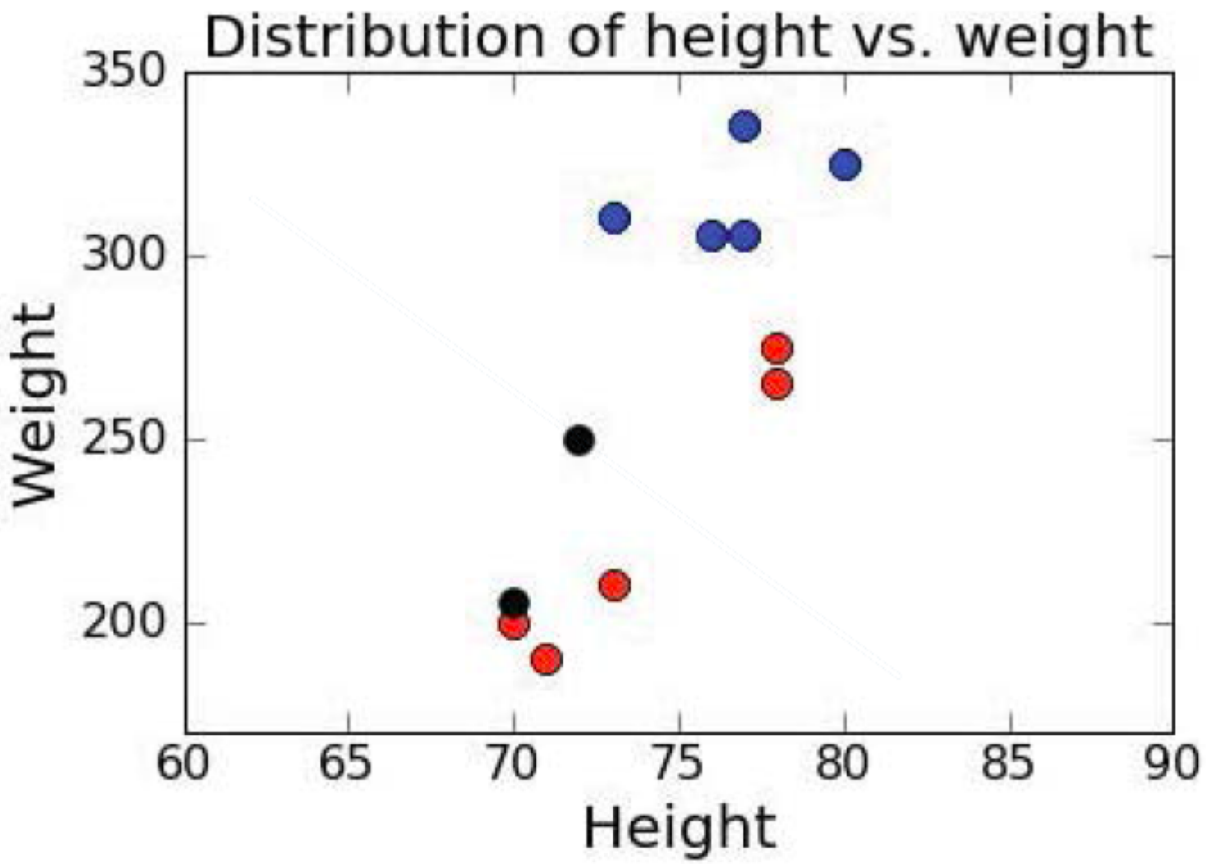
- Given labeled groups in feature space, want to find subsurface in that space that separates the groups
  Subject to constraints on complexity of subsurface
- In this example, have 2D space, so find line (or connected set of line segments) that best separates the two groups
- When examples well separated, this is straightforward
- When examples in labeled groups overlap, may have to trade off false positives and false negatives

Suppose Data Was Labeled

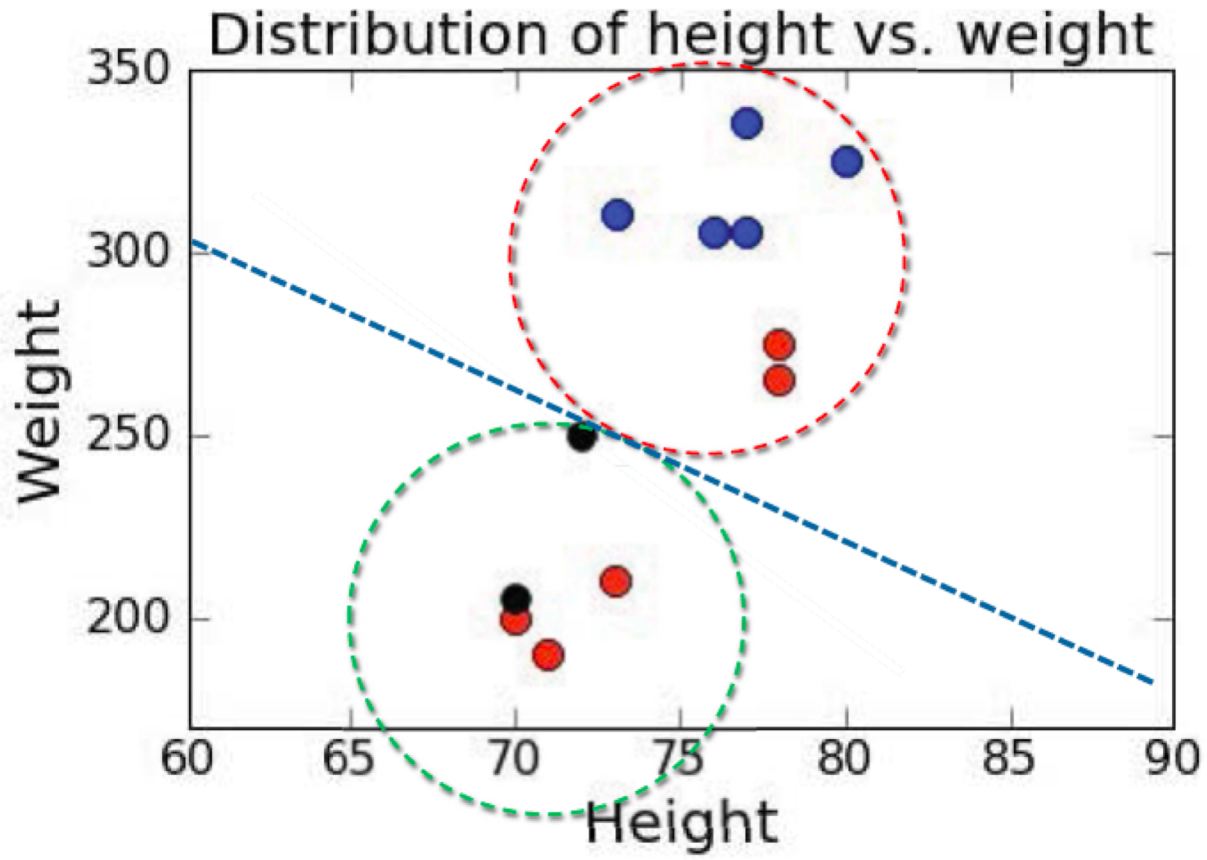


Given we have labeled data, then an obvious separator of two groups is shown

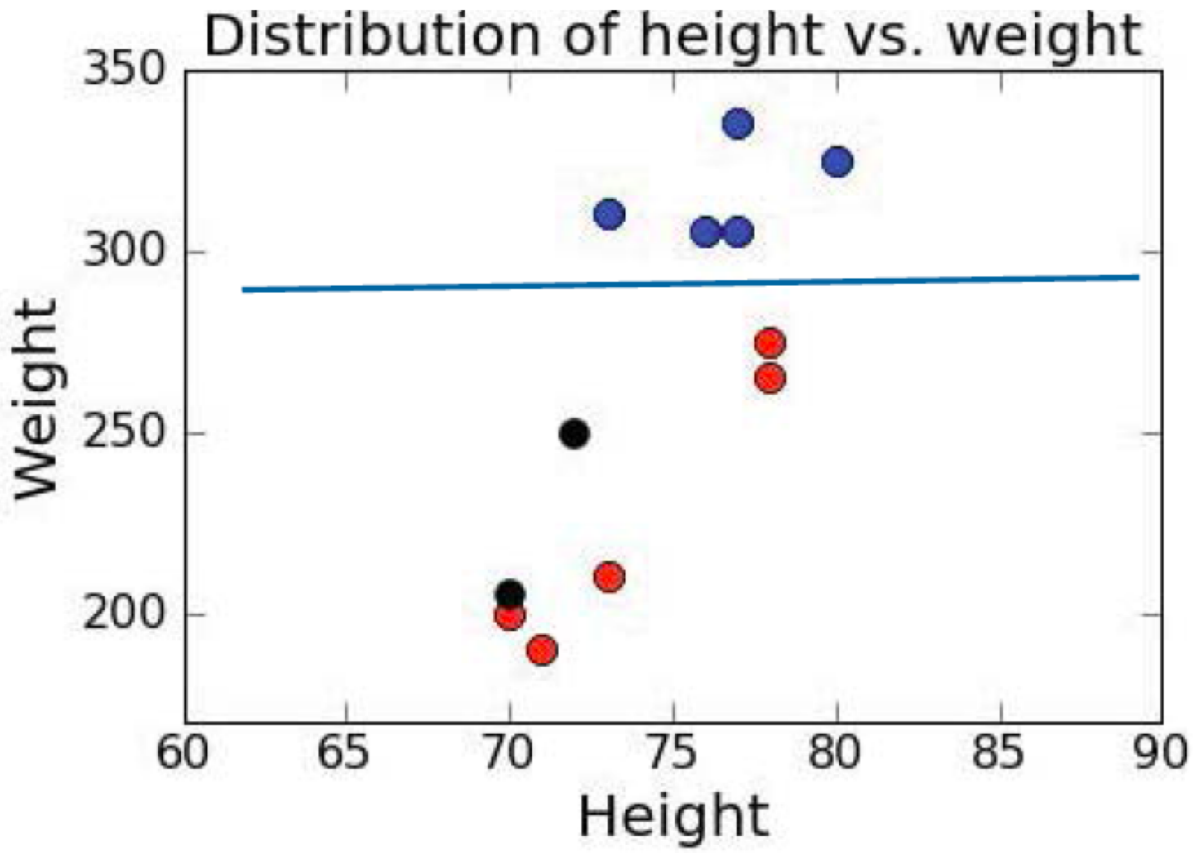
Adding Some New Data



Clustering using Unlabeled Data



Classified using Labeled Data



Machine Learning Methods

- We will see some examples of machine learning methods:
- Learn models based on unlabeled data, by clustering training data into groups of nearby points
  - Resulting clusters can assign labels to new data
- Learn models that separate labeled groups of similar data from other groups
  - May not be possible to perfectly separate groups, without "overfitting"
  - But can make decisions with respect to trading off "false positives" versus "false negatives"
  - Resulting classifiers can assign labels to new data

All ML Methods Require:

- Choosing training data and evaluation method
- Representation of the features
- Distance metric for feature vectors
- Objective function and constraints
- Optimization method for learning the model

Feature Representation

- Features never fully describe the situation
  - ”All models are wrong, but some are useful.” – George Box
- Feature engineering
  - Represent examples by feature vectors that will facilitate generalization
  - Suppose I want to use 100 examples from past to predict, at the start of the subject, which students will get an A
    - Some features surely helpful, e.g., GPA, prior programming experience (not a perfect predictor)
  - Others might cause me to overfit, e.g., birth month, eye color
- Want to maximize ratio of useful input to irrelevant input
  - Signal–to–Noise Ratio (SNR)

An Example

Features | **Label**

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes

Initial model:

- Not enough information to generalize

An Example

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes

Initial model:

- Egg laying
- Has scales
- Is poisonous
- Cold blooded
- No legs

An Example

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes

Initial model:

- Egg laying
- Has scales
- Is poisonous
- Cold blooded
- No legs

Current model:

- Has scales
- Cold blooded
- No legs

Boa doesn't fit model, but is labeled as reptile.
Need to refine model

An Example

Features | **Label**

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No

Current model:

- Has scales
- Cold blooded
- No legs

Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

An Example

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes

Current model:

- Has scales
- Cold blooded
- Has 0 or 4 legs

Current model:

- Has scales
- Cold blooded
- No legs

Alligator doesn't fit model, but is labeled as reptile.
Need to refine model

An Example

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No

Current model:

- Has scales
- Cold blooded
- Has 0 or 4 legs

An Example

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

Current model:

- Has scales
- Cold blooded
- Has 0 or 4 legs

No (easy) way to add to rule that will correctly classify salmon and python (since identical feature values)

An Example

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

Good model:

- Has scales
- Cold blooded

Not perfect, but no false negatives (anything classified as not reptile is correctly labeled); some false positives (may incorrectly label some animals as reptile)

Need to Measure Distances between Features

- Feature engineering:
  - Deciding which features to include and which are merely adding noise to classifier
  - Defining how to measure distances between training examples (and ultimately between classifiers and new instances)
  - Deciding how to weight relative importance of different dimensions of feature vector, which impacts definition of distance

Measuring Distance Between Animals

- We can think of our animal examples as consisting of four binary features and one integer feature
- One way to learn to separate reptiles from non− reptiles is to measure the distance between pairs of examples, and use that:
  - To cluster nearby examples into a common class (unlabeled data), or
  - To find a classifier surface in space of examples that optimally separates different (labeled) collections of examples from other collections

rattLesnake = [1,1,1,1,0]
Boa constrictor = [0,1,0,1,0]
Dart Frog = [1,0,1,0,4]

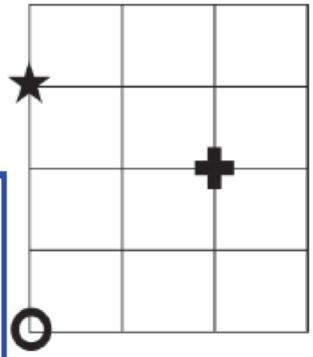
Can convert examples into feature vectors

Minkowski Metric

$$dist(X1, X2, p) = \left(\sum_{k=1}^{len} abs(X1_k - X2_k)^p\right)^{1/p}$$

p = 1: Manhattan Distance
p = 2: Euclidean Distance

Need to measure distances between feature vectors



Typically use Euclidean metric; Manhattan may be appropriate if different dimensions are not comparable

Is circle closer to star or cross?

- Euclidean distance
  - Cross – 2.8
  - Star – 3
- Manhattan Distance
  - Cross – 4
  - Star - 3

Euclidean Distance Between Animals

rattLesnake = [1,1,1,1,0]
Boa constrictor = [0,1,0,1,0]
dartFrog = [1,0,1,0,4]

Euclidean Distance Between Animals

rattLesnake = [1,1,1,1,0]
Boa constrictor = [0,1,0,1,0]
dartFrog = [1,0,1,0,4]

	rattlesnake	boa constrictor	dart frog
rattlesnake	--	1.414	4.243
boa constrictor	1.414	--	4.472
dart frog	4.243	4.472	--

Using Euclidean distance, rattlesnake and boa constrictor are much closer to each other, than they are to the dart frog

Add an Alligator

- alligator = Animal('alligator', [1,1,0,1,4])
- animals.append(alligator)
- compareAnimals(animals, 3)

Add an Alligator

- alligator =Animal('alligator',[1,1,0,1,4])
- animals.append(alligator)
- compareAnimals(animals, 3)

RattLesnake=[1,1,1,1,0]
Boa constrictor =[0,1,0,1,0]
Dart Frog=[1,0,1,0,4]

	rattlesnake	boa constrictor	dart frog	alligator
rattlesnake	--	1.414	4.243	4.123
boa constrictor	1.414	--	4.472	4.123
dart frog	4.243	4.472	--	1.732
alligator	4.123	4.123	1.732	--

Alligator is closer to dart frog than to snakes − why?

- Alligator differs from frog in 3 features, from boa in only 2 features
- But scale on “legs” is from 0 to 4, on other features is 0 to 1
- “legs” dimension is disproportionately large

Using Binary Features

rattLesnake = [1,1,1,1,0]
Boa constrictor = [0,1,0,1,0]
dartFrog = [1,0,1,0,1]
Alligator = [1,1,0,1,1]

	rattlesnake	boa constrictor	dart frog	alligator
rattlesnake	--	1.414	1.732	1.414
boa constrictor	1.414	--	2.236	1.414
dart frog	1.732	2.236	--	1.732
alligator	1.414	1.414	1.732	--

Now alligator is closer to snakes than it is to dart frog

– makes more sense

Feature Engineering Matters

Supervised versus Unsupervised Learning

- In the next few lectures, we will see examples of learning algorithms:
- When given unlabeled data, try to find clusters of examples near each other
  - Use centroids of clusters as definition of each learned class
  - New data assigned to closest cluster
- When given labeled data, learn mathematical surface that "best" separates labeled examples, subject to constraints on complexity of surface (don't over fit)
  - New data assigned to class based on portion of feature space carved out by classifier surface in which it lies

Issues of Concern When Learning Models

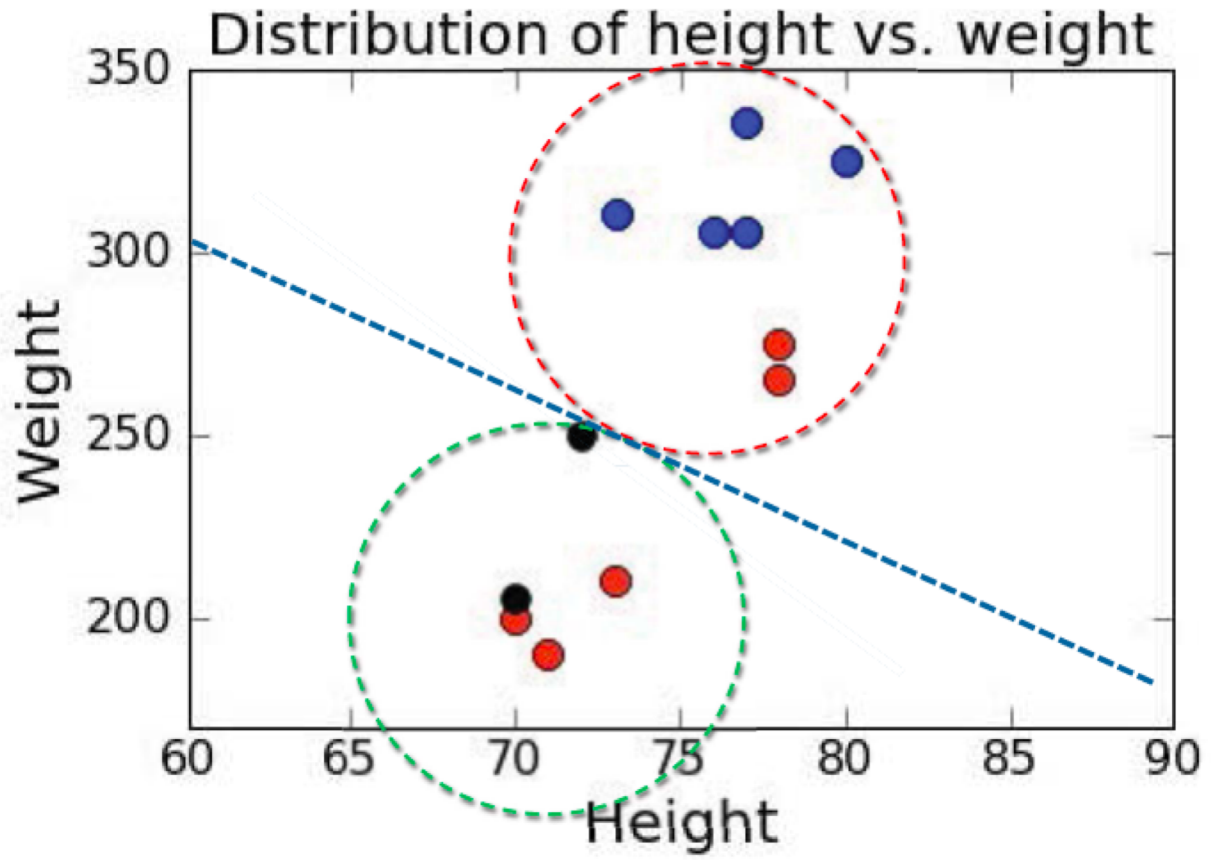
Learned models will depend on:

- Distance metric between examples
- Choice of feature vectors
- Constraints on complexity of model
  - Specified number of clusters
  - Complexity of separating surface
  - Want to avoid overfitting problem (each example is its own cluster, or a complex separating surface)

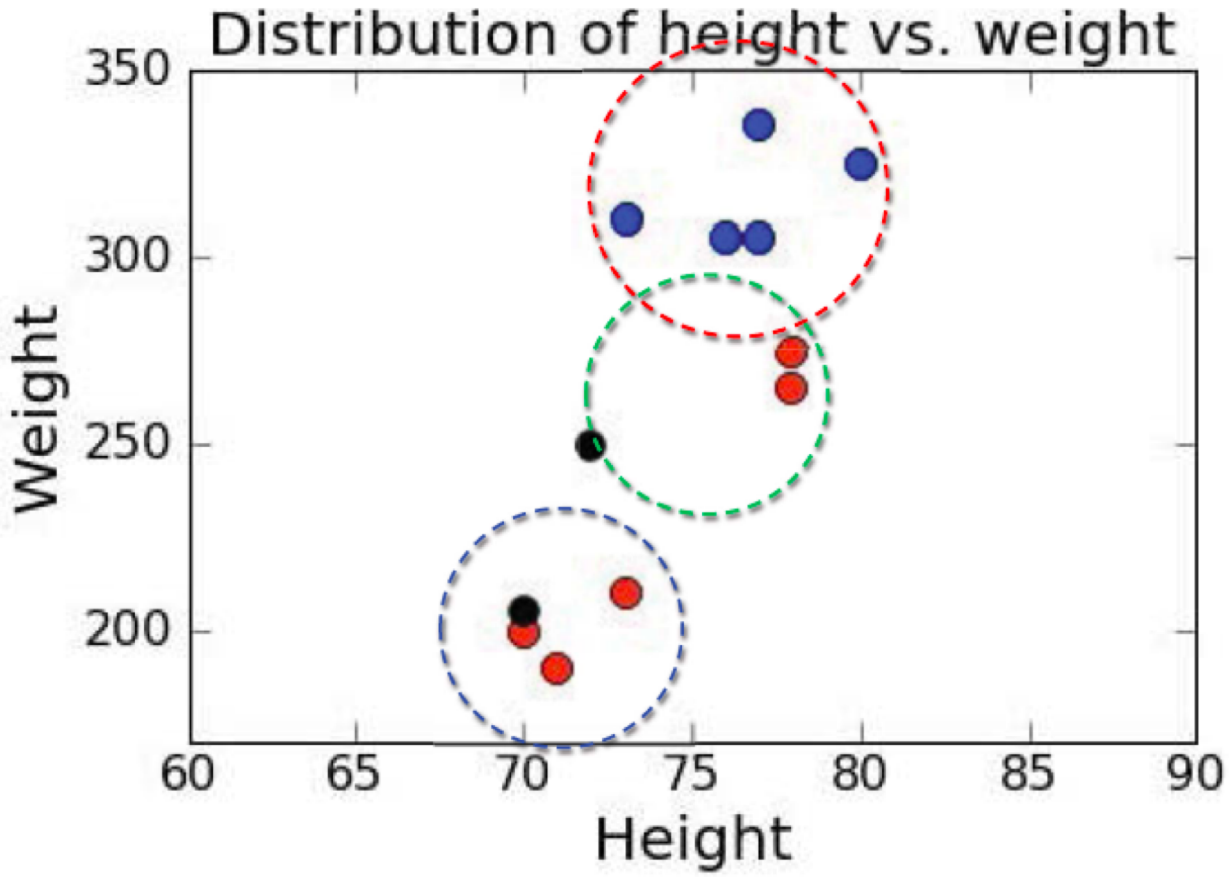
Clustering approaches

- Suppose we know that there are k different groups in our training data, but don't know labels
  - Pick k samples (at random?) as exemplars
  - Cluster remaining samples by minimizing distance between samples in same cluster (objective function) – put sample in group with closest exemplar
  - Find median example in each cluster as new exemplar
  - Repeat until no change
- Issues:
  - How do we decide on the best number of clusters?
  - How do we select the best features, the best distance metric?

Clustering using Unlabeled Data



Fitting Three Clusters Unsupervised



Classification approaches

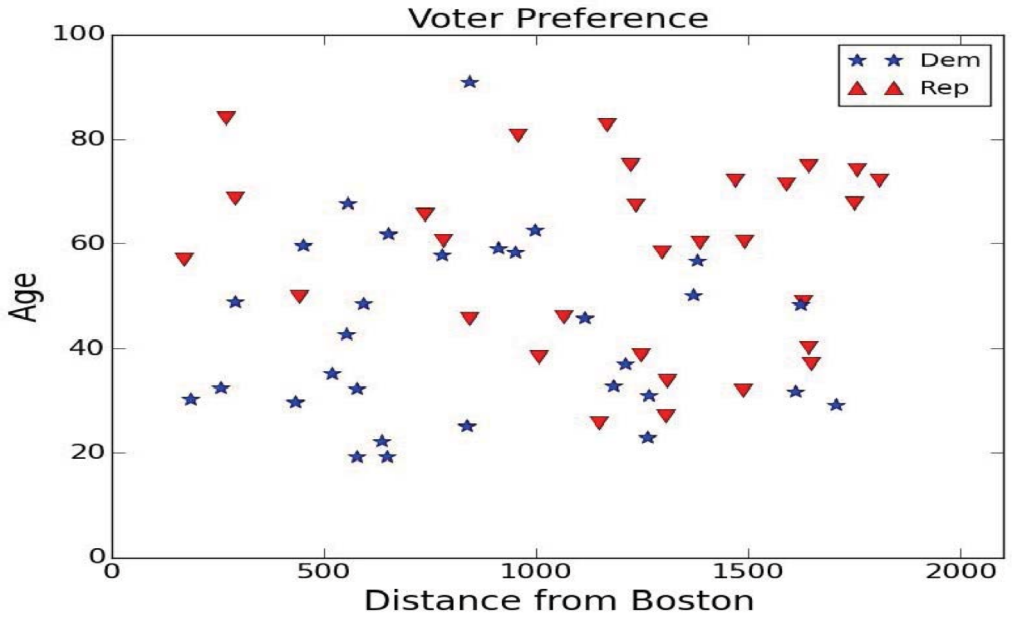
- Want to find boundaries in feature space that separate different classes of labeled examples
  - Look for simple surface (e.g. best line or plane) that separates classes
  - Look for more complex surfaces (subject to constraints) that separate classes
  - Use voting schemes
    - Find k-nearest training examples, use majority vote to select label

- Issues:
  - How do we avoid over−fitting to data?
  - How do we measure performance?
  - How do we select best features?

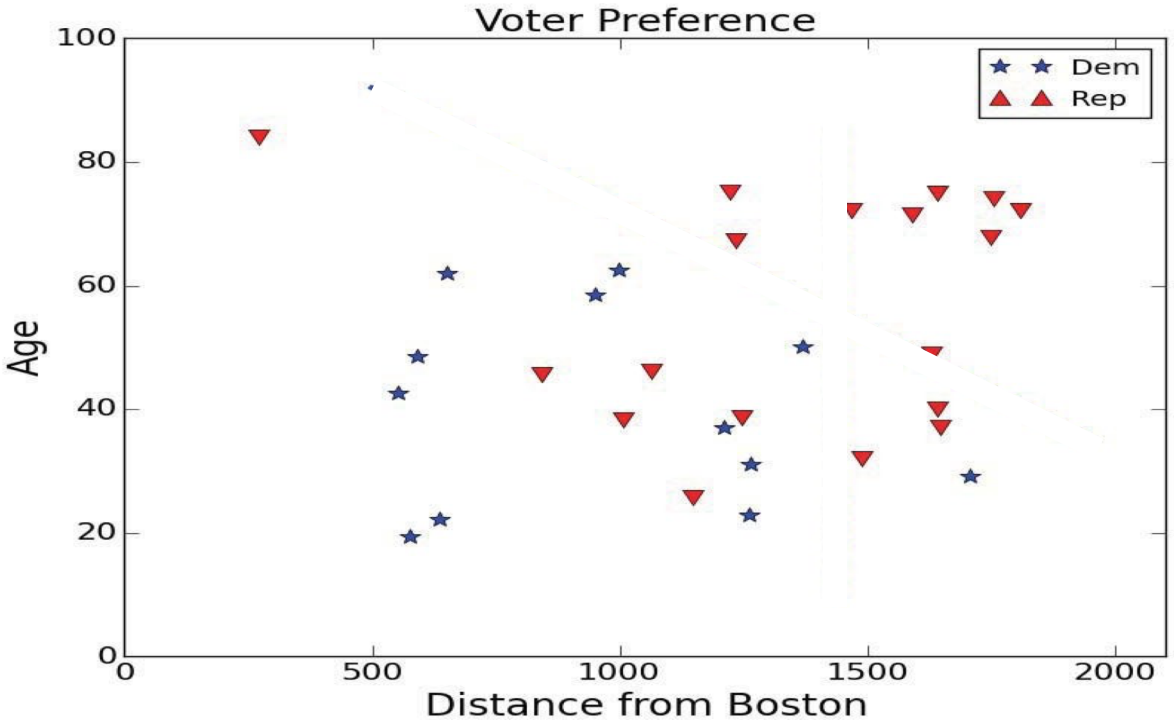
Classification

- Attempt to minimize error on training data
  - Similar to fitting a curve to data
- Evaluate on training data

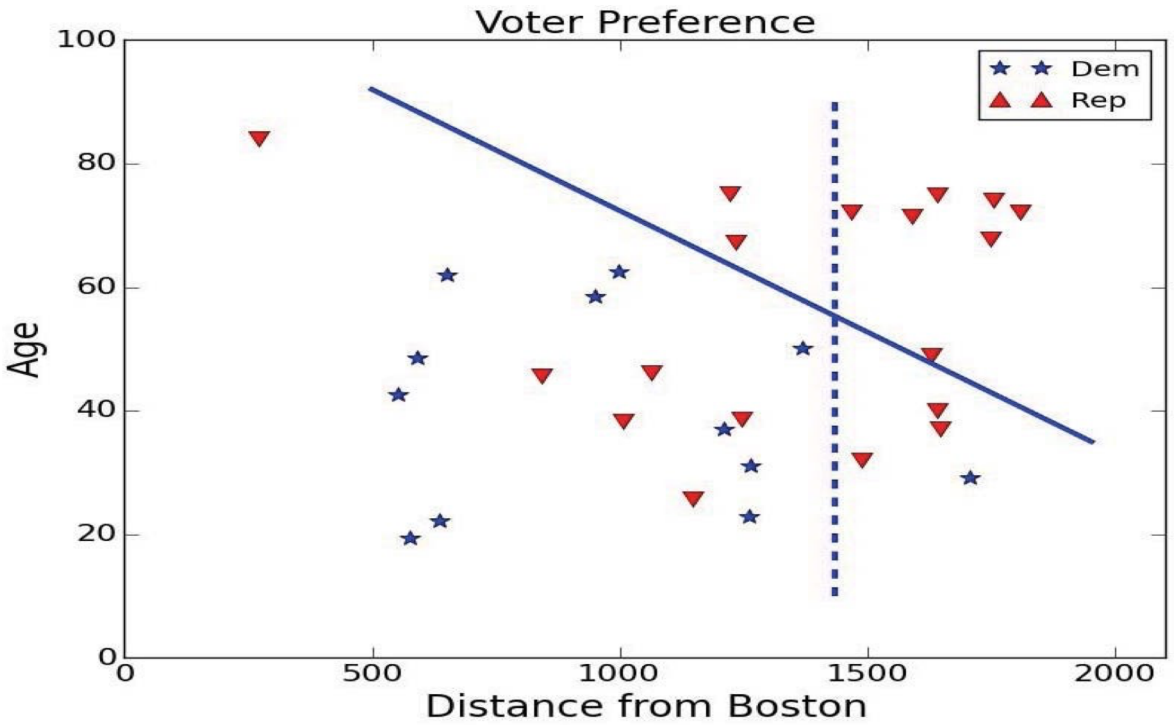
Voter preference, by age and distance from Boston



Randomly Divide Data into Training and Test Set



Two Possible Models for a Training Set



Confusion Matrices (Training Error)

Predicted Democratic

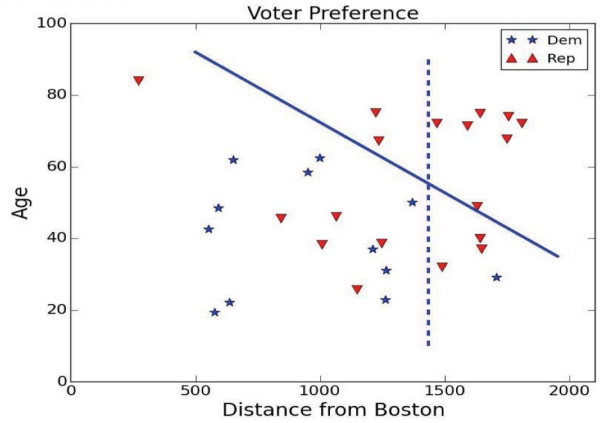
Actually Dem.

	Pos	Neg
Pos	12	0
Neg	9	9

Solid Line

	Pos	Neg
Pos	11	1
Neg	8	10

Dashed Line

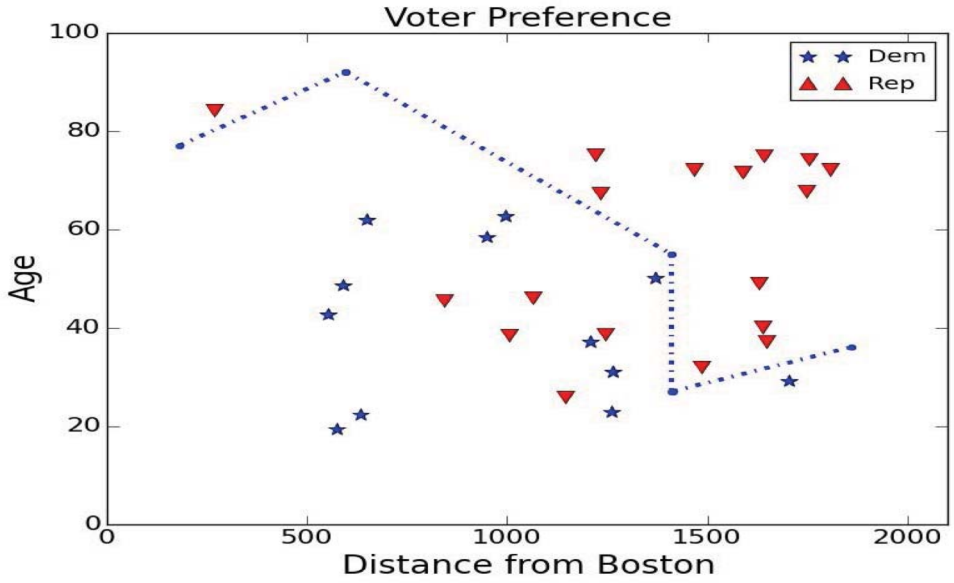


Training Accuracy of Models

$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}$$

- 0.7 for both models
  - Which is better?
- Can we find a model with less training error?

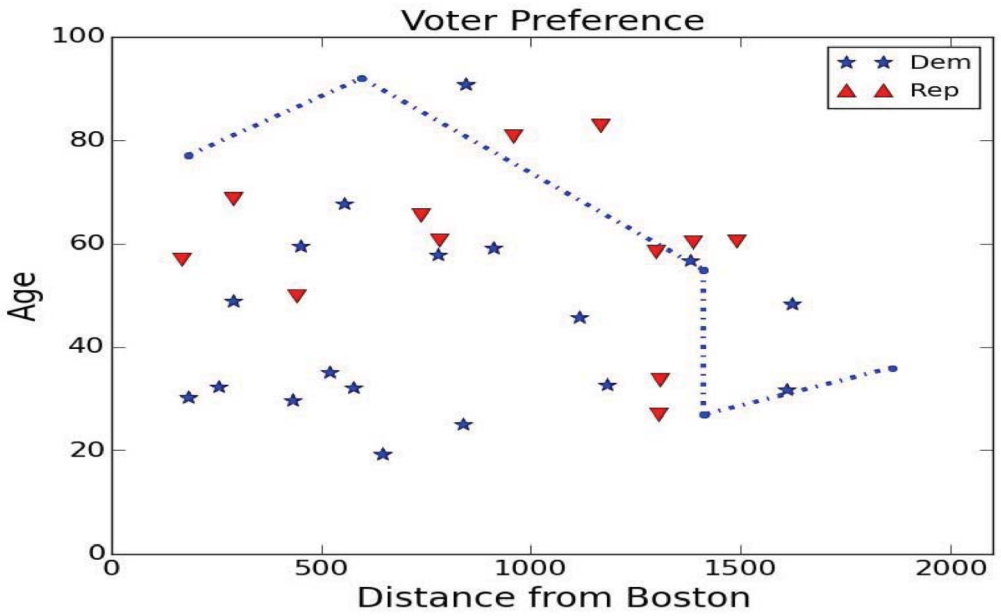
A More Complex Model



TP = 12, FP = 5, TN = 13, FN = 0

Accuracy = 25/30 = 0.833

Applying Model to Test Data



TP = 14, fP = 4, TN = 4, fN = 8

Accuracy = 18/30 = 0.6

Other statistical measures:

$$positive\ predictive\ value = \frac{true\ positive}{true\ positive + false\ positive}$$

- Solid line model: .57
- Dashed line model: .58
- Complex model, training: .71
- Complex model, testing: .78

- You will also see "sensitivity" versus "specificity"

precision = $sensitivity = \frac{true\ positive}{true\ positive + false\ negative}$

$$specificity = \frac{true\ negative}{true\ negative + false\ positive}$$

Percentage correctly found
Percentage correctly rejected

Summary

- Machine learning methods provide a way of building models of processes from data sets
  - Supervised learning uses labeled data, and creates classifiers that optimally separate data into known classes
  - Unsupervised learning tries to infer latent variables by clustering training examples into nearby groups
- Choice of features influences results
- Choice of distance measurement between examples influences results
- We will see some examples of clustering methods, such as k−means
- We will see some examples of classifiers, such as k-nearest neighbor methods