



THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學



PolyU 理大商學院
Business School
Innovation-driven Education and Scholarship

School of
**ACCOUNTING
& FINANCE**
會計及金融學院

Week 9: Introduction to Machine Learning in Accounting and Finance

AF3214 Python Programming for Accounting and Finance

Vincent Y. Zhuang, Ph.D.
vincent.zhuang@polyu.edu.hk

School of Accounting and Finance
The Hong Kong Polytechnic University

R508, 8:30 am – 11:20 am, Wednesdays, Semester 2, AY 2024-25


Motivation

- Humans learn in two ways—**memorization** and **generalization**. We use **memorization** to accumulate individual facts. We use **generalization** to deduce new facts from old facts.
- While machine learning is hard to define. In some sense, every useful program **learns** something.
- When scientists speak about machine learning, often mean the discipline of writing programs that automatically learn to make useful **inferences** from **implicit patterns** in data.
- Machine learning is a huge topic - with whole courses devoted to it.
 - **Topics include** natural language processing (NLP), computational biology, computer vision, robotics, and in accounting and finance (i.e., Dynamic Pricing, big data analysis in risk management, trading strategies, financial econometrics, statistical computing, probabilistic programming).


Motivation

Today, we will:

- Give you the basic introduction.
- Start by talking about the basic concepts of machine learning.
 1. The idea of having examples, how do we talk about features representing those examples.
 2. How do we measure distances between examples, and
 3. Use the notion of distance to try and group similar things together as a way of doing machine learning.
 4. Look two different standard ways of doing learning:
 - classification method, such as “K-nearest-neighbour”
 - clustering method, such as “K-means”



works well when I have labeled data on my examples and use them to define classes that I can learn



works well when I don't have labeled data on my examples

A Little History

In 1955, John McCarthy, Assistant Professor of Mathematics, at Dartmouth College submitted a proposal* with Marvin Minsky, Nathaniel Rochester, & Claude Shannon for the *Dartmouth Summer Research Project* on Artificial Intelligence. These organizers were joined in the summer of 1956 by Trenchard More, Oliver Selfridge, Herbert Simon, Ray Solomonoff, among others. The stated goal was ambitious:

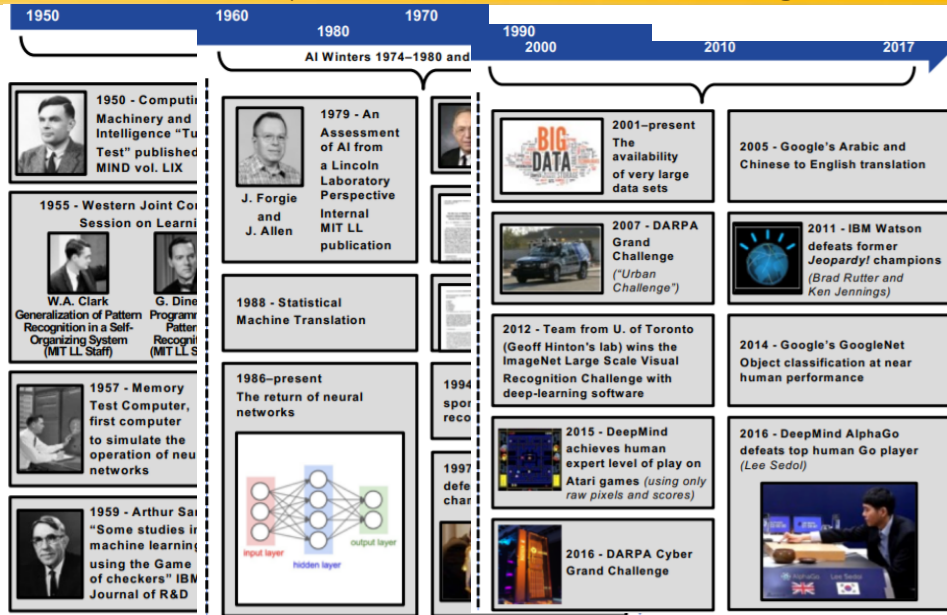
“The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.”

The field of Artificial Intelligence was born.

*McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1955, August). A proposal for the Dartmouth summer research project on artificial intelligence.

<http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>

Selected History of AI and Machine Learning



Machine Learning is Everywhere



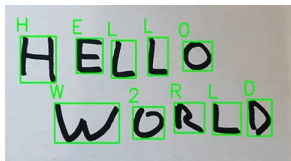
AlphaGo



Recommendation Systems



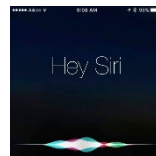
Drug Discovery



Pattern/character recognition



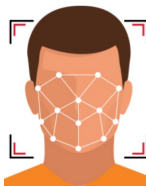
Stock prediction by Hedge Fund



Voice assistant



Assisted driving

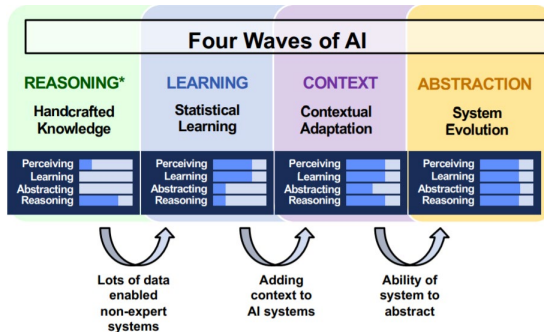


Face detection



Medical diagnostic

AI and Machine Learning Evolution

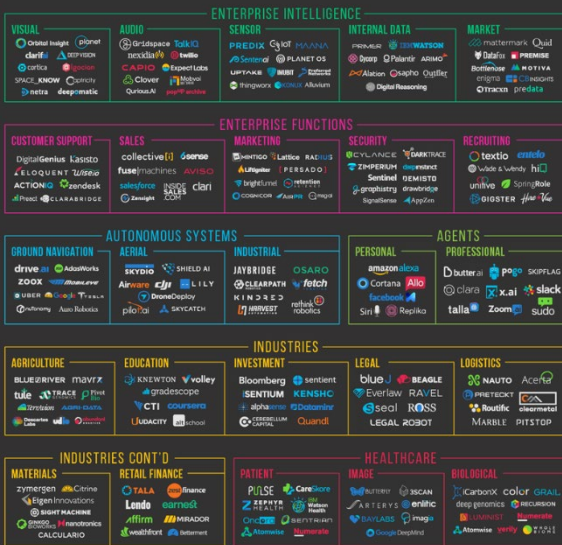


	First Wave	Second Wave	Third Wave	Fourth Wave
Timeline	c. 1970 – 1990	c. 2000 – present	est. 2020 – 2040	est. 2040 –
Main Feature	Reasoning	Learning	Explaining	Critical Thinking
Method	Rules	Statistics	Non-Statistical	???
Outcomes	Predictable	Probabilistic	Guaranteed	???
Example	Chess Tic-tac-toe	Medical Imaging Stratification	Biomarker Ranking	Human Intelligence

As defined by the U.S. Defense Advanced Research Projects Agency (DARPA), the four waves of AI refer to the state of AI capabilities past, present, and future.

Spectrum of Commercials in Machine Intelligence

MACHINE INTELLIGENCE 3.0



shivonzills.com/MACHINEINTELLIGENCE · Bloomberg BETA

TECHNOLOGY STACK



The Current State of Machine Intelligence

Machine Intelligence LANDSCAPE

CORE TECHNOLOGIES

ARTIFICIAL INTELLIGENCE

IBM Watson
Numenta
Cycorp
Reactor
ai-one
Research non
KAD
KAD

DEEP LEARNING

vicarious
facebook
Google
SKYND
Vision Factory
erschl
SignalSense

MACHINE LEARNING

rapidminer
context
DATA
Lumigo
Adaptix
Alpine
AYASDI

NLP PLATFORMS

cortical.io
idibon
wit.ai
LUMINO
Malaya

PREDICTIVE APIS

AlchemyAPI
Google
MINDOS
big
indico
ALGORITHMS
Expect
PredictionIO
Labs

IMAGE RECOGNITION

clarifai
MADBITS
DNNresearch
VISENZE
lookflow

SPEECH RECOGNITION

GRIDSPACE
popUP archive
NUANCE

RETHINKING ENTERPRISE

SALES

Preact
RelateIO
infer
AVISO
NGRDATA
FRAMED
cassia

SECURITY / AUTHENTICATION

CROSSMATCH
CYCLANCE
c-njur
BITSIGHT
bionym

FRAUD DETECTION

sift science
ThreatMetrix
Brighterion
secure
feedzai
verafin

HR / RECRUITING

TalentBin
predikt
gild
connect
hire

MARKETING

brightfuel
CommandIQ
RADIUS
Teqpart
bloomreach
AIRPR
people
pattern
fusion

PERSONAL ASSISTANT

Siri
Cortana
tempo
KASIST
VIV
Google now
cleversense
Rebinlabs
fuse
CLARA LABS

INTELLIGENCE TOOLS

ADATAD
Palantir
Quid
FirstRain
Digital Reasoning

RETHINKING INDUSTRIES

ADTECH

METAMARKETS
dStillery
rocketfuel
YieldMo
ADBRAIN

AGRICULTURE

BLUE RIVER
censmaging
tule
Terraviva
KONIGSBERG
tule

EDUCATION

edexra
coursera
KNEWTON
KIDaptive

FINANCE

Bloomberg
alphasense
Dataminr
KENSHC
minibrock
BINATIX

LEGAL

Lex Machina
brightleaf
COUNSELLYTICS
JUDICATA
Brevia
Biligence

MANUFACTURING

SIGHT MACHINE
MICROSCAN
IVISYS
BROOKS
MAGNET

MEDICAL

Parzival
Genescent
groundtable
transcriptic
ZEPHYR
bina
TUTE

OIL AND GAS

kaggle
bieta
TACHYUS
Rutur

MEDIA / CONTENT

Outbrain
newsie
SAILTHRU
wovii
Prismatic
ARRIA
Chorus
wovii
Summy

CONSUMER FINANCE

affirm
venture
finance
GUARD
LendUp
LendingClub
Kabbage

PHILANTHROPIES

DataKind
thorn
DATA
GUIDE

AUTOMOTIVE

Google
Quintient
Cruise

DIAGNOSTICS

enlitic
lumiat
3SCAN
lumiat

RETAIL

BAY SENSORS
PRISM SKYLARS
select
euclid

RETHINKING HUMANS / HCI

AUGMENTED REALITY

THALMIC LABS
APX
blippar
Picta
layor

GESTURAL COMPUTING

THALMIC LABS
Leap
Leap
Leap
Leap

ROBOTICS

Intel
Liqui Robotics
SoftBank
Robot
Andri

EMOTIONAL RECOGNITION

affectiva
BEYOND VERBAL
EMOTION
logito

SUPPORTING TECHNOLOGIES

HARDWARE

NVIDIA
XILINX
QUALCOMM
NVIDIA
rigit

DATA PREP

TRIFACTA
tamr
Alation

DATA COLLECTION

diffbot
kimono
CrowdFlower
Import

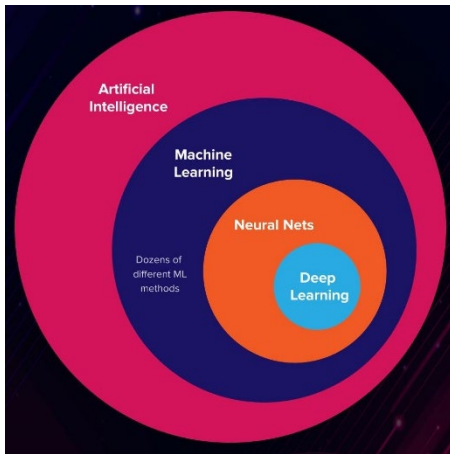
Breakthroughs in AI and Machine Learning

Year	Breakthroughs in AI & ML	Datasets (First Available)	Algorithms (First Proposed)
1994	Human-level spontaneous speech recognition	Spoken Wall Street Journal articles and other texts (1991)	Hidden Markov Model (1984)
1997	IBM Deep Blue defeated Garry Kasparov	700,000 Grandmaster chess games, aka "The Extended Book" (1991)	Negascout planning algorithm (1983)
2005	Google's Arabic-and Chinese-to-English translation	1.8 trillion tokens from Google Web and News pages (collected in 2005)	Statistical machine translation algorithm (1988)
2011	IBM Watson became the world Jeopardy! champion	8.6 million documents from Wikipedia, Wiktionary, and Project Gutenberg (updated in 2010)	Mixture-of-Experts (1991)
2014	Google's GoogLeNet object classification at near-human performance	ImageNet corpus of 1.5 million labeled images and 1,000 object categories (2010)	Convolutional Neural Networks (1989)
2015	Google's DeepMind achieved human parity in playing 29 Atari games by learning general control from video	Arcade Learning Environment dataset of over 50 Atari games (2013)	Q-learning (1992)
Average # of Years to Breakthrough:		3 years	18 years

The average elapsed time between key algorithm proposals and corresponding advances was about 18 years, whereas the average elapsed time between key dataset availabilities and corresponding advances was less than 3 years, or about 6 times faster.

AI and Machine Learning?

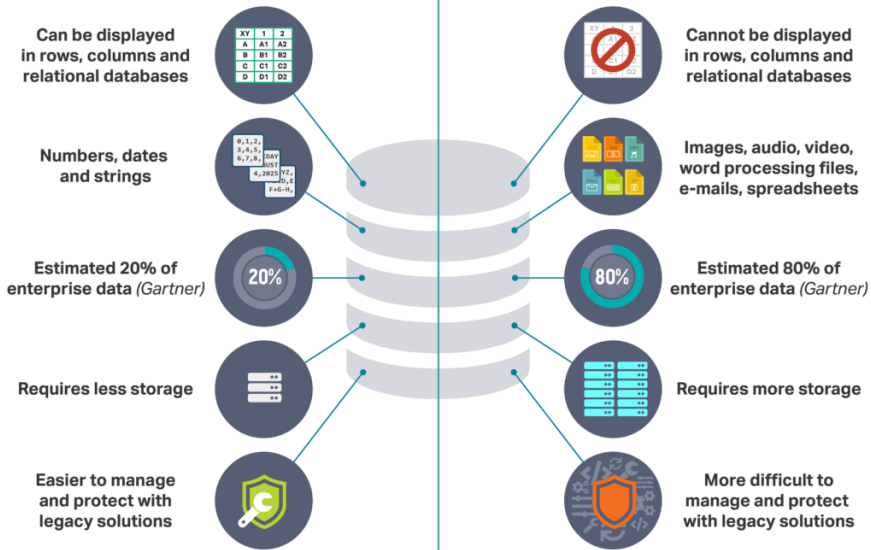
What's the difference between AI and Machine Learning?



Artificial intelligence is science. It studies ways to build intelligent programs and machines that can creatively solve problems. **Machine learning** is a subset of artificial intelligence and **Deep learning** is a subset of machine learning.

Unstructured and Structured Data

Structured Data vs Unstructured Data

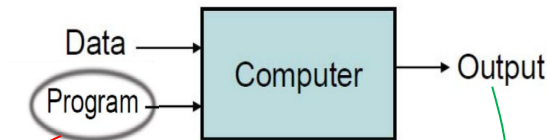


What Is Machine Learning?

- All useful programs "learn" something
 - e.g., *You are designing an algorithm to*
 - *calculate your portfolio return;*
 - *find optimal portfolio;*
- Early definition of machine learning:
 - *"Field of study that gives computers the ability to learn without being explicitly programmed."* Arthur Samuel (1959)
 - A computer pioneer who wrote first self-learning program, which played checkers - learned from "experience"
 - Invented alpha-beta pruning - widely used in decision tree searching
 - The idea is, how can we have the computer learn without being explicitly programmed

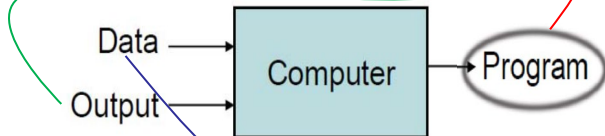
What is Machine Learning? - Cont'd

Traditional Programming



Square root finder

Machine Learning



Loop-to solve some other problems

Curve fitting by linear regression and non-linear regression (Linear, Quadratic, Cubic)

labels on data, characterizations of different classes of things.

My expectation from computer:

Given characterizations of output and data, ML algorithm to produce for me a program to infer new information about things.

How Are Things Learned?

If we want to learn things, we could also ask: How does a human learn?


- **Memorization** [having Wikipedia in your back pocket, naïve & old school?]
 - Accumulation of individual facts
 - Memorize as many as you can
 - Limited by
 - Time to observe facts
 - Memory to store facts
- **Generalization** [maybe better, be able to infer new info]
 - Ways to Deduce new facts from old facts
 - Limited by accuracy of deduction process
 - Essentially a predictive activity
 - Assumes that the past predicts the future

Declarative knowledge

Imperative knowledge



Interested in extending to programs that can infer useful information from **implicit patterns** in data

So idea? Basic Paradigm for Computer to Learn

- Observe set of examples: **training data (some obs)**  feed the system

Past stock performance
- Infer something (**infers engine**) about process that generated data

Write a program to find the best portfolio using PCA and backtesting
- Use inference to make predictions about previously unseen data: **test data**

Predict portfolio performance
- Variations on paradigm  How to do inference on labeling new things.
 - Supervised:** given a set of feature/label pairs, find a rule that predicts the label associated with a previously unseen input
 - Unsupervised:** given a set of feature vectors (without labels)
 -  group them into "natural clusters" (or create labels for groups)
 - find the natural ways to group those examples together into different models.

Some Examples of Classifying and Clustering

- Here are some data on some student basketball players:

Data: Name, height (inches), weight (lb)

Label: by the position type: point guard (PG), the shooting guard (SG), the small forward (SF), the power forward (PF), and the center (C)

- Group I:

Alice = ['point guard', 70, 200]

Bob = ['shooting guard', 73, 210]

Carmen = ['small forward', 78, 265]

Damon = ['power forward', 71, 190]

Elliot = ['center', 78, 275]

What do we want to do with this?

1. Come up a way of characterizing the implicit pattern of how does height and weight predict the position for this player

- Group II:

Frankie = ['point guard ', 77, 335]

George = ['shooting guard ', 80, 325]

Hannah = ['small forward ', 73, 310]

Iris = ['power forward', 77, 305]

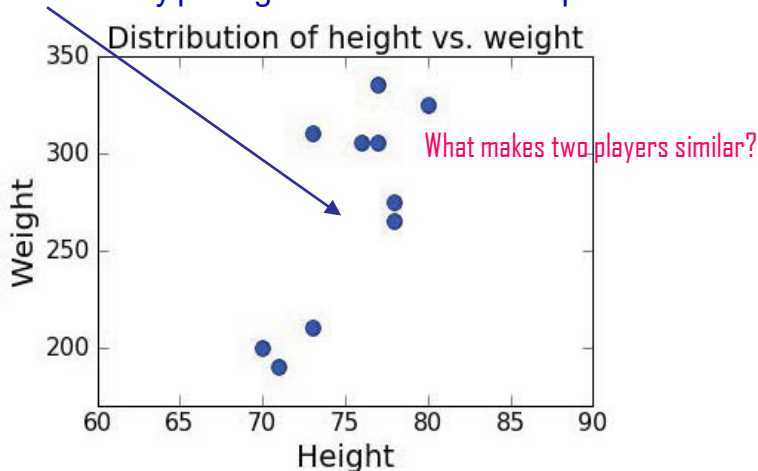
Jimmy = ['center ', 76, 305]

2. Come up with an algorithm that will predict the position of new players

The paradigm: set of obs, labeled or not, to infer a model, use the model for prediction

Some Examples of Classifying and Clustering - Cont'd

Unlabeled data by plotting on a two dimensional plot



I am trying to learn...

1. Are their characteristics that distinguish two classes from one another?
2. Can I separate this distribution into two or more natural groups?

Some Examples of Classifying and Clustering - Cont'd

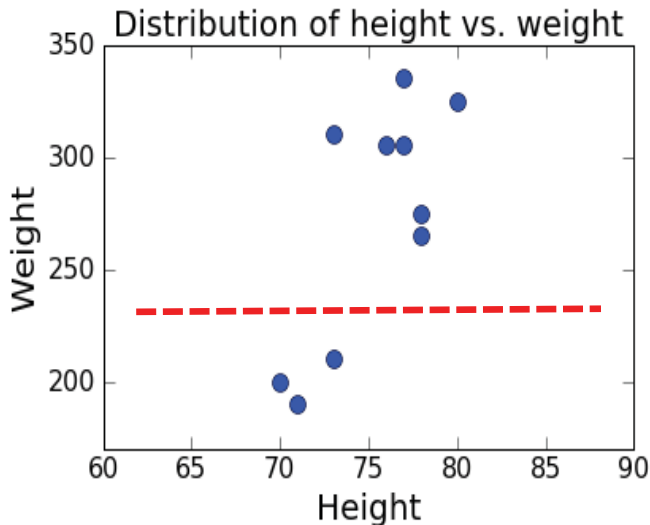
Clustering examples into groups

- Want to decide on "similarity" of examples, with goal of separating into distinct, "natural", groups
 - Similarity is a **distance measure**
- 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
 - Create clusters with the property that the distances between all of the examples of that cluster are small.
 - Find clusters the average distance for both clusters as small as possible.
 - Cluster remaining samples by minimizing distance between samples in same cluster(**objective function**)-put sample in group with closest exemplar
 - Once clusters fixed, find median element/example in each cluster as new exemplar (the one closest to the center, and treat it as exemplars)
 - Repeat above steps until no change in the process

clustering based on distance

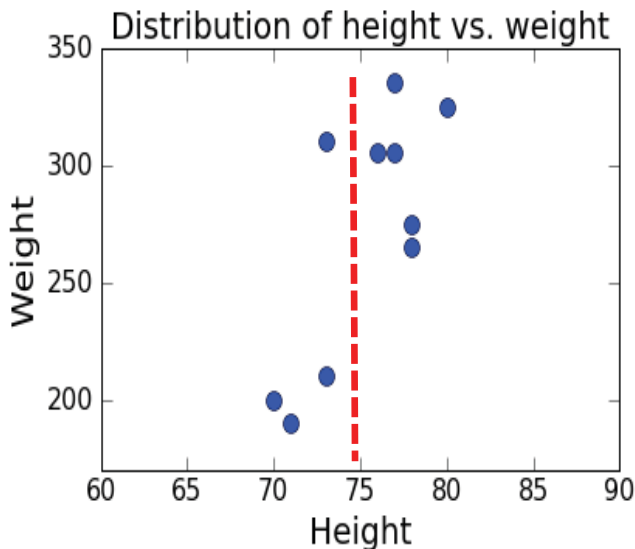
Some Examples of Classifying and Clustering - Cont'd

Similarity Based on Weight



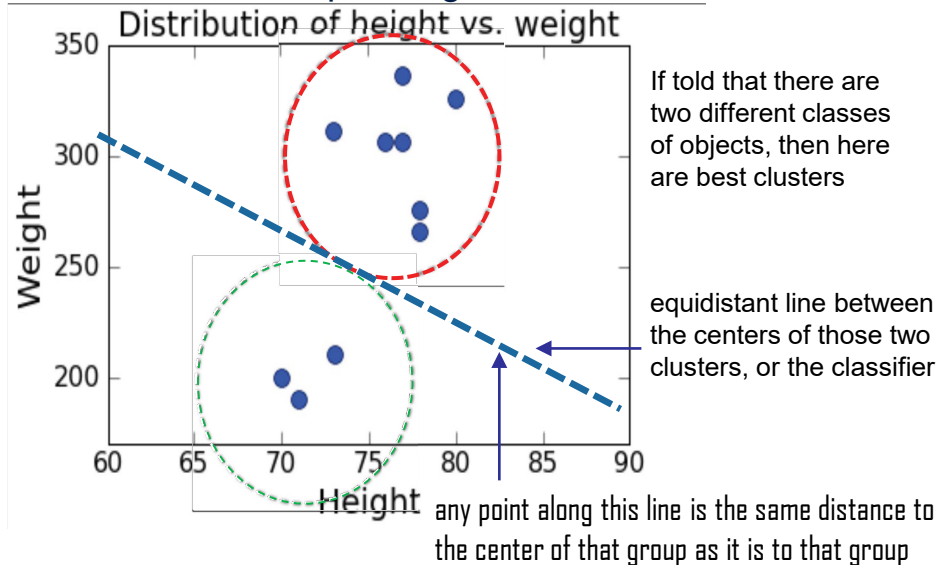
Some Examples of Classifying and Clustering - Cont'd

Similarity Based on Height



Some Examples of Classifying and Clustering - Cont'd

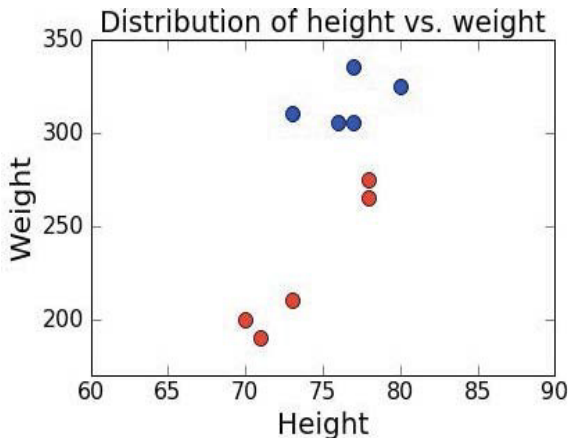
Cluster into Two Groups Using Both Attributes



Some Examples of Classifying and Clustering - Cont'd

Suppose Data Was Labeled

- Red-group I, Blue-group II



If I could take advantage of knowing the labels, how would I divide these groups up?

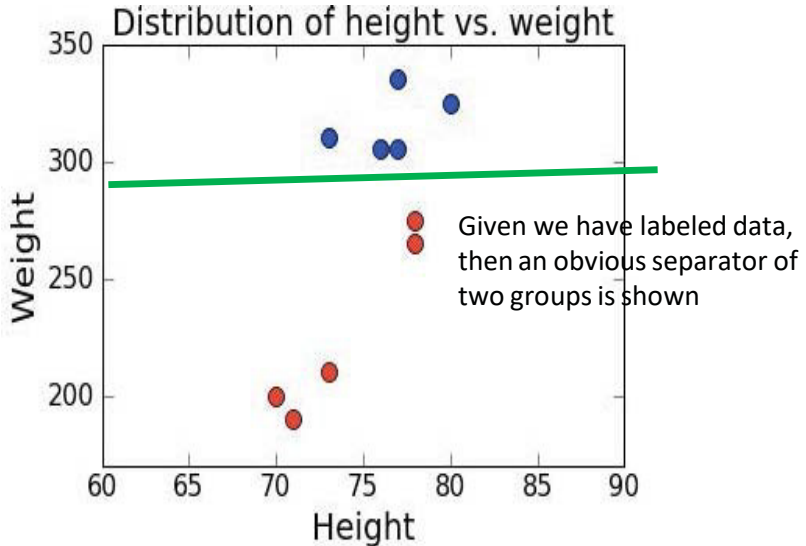
Some Examples of Classifying and Clustering - Cont'd

Finding Classifier Surfaces

- Given labeled groups in feature space, want to find subsurface in that space that naturally separates the groups
 - Subject to constraints on complexity of subsurface
 - subsurface is, in the two-dimensional case, I want to know what's the best line that separates all the examples with one label from all the examples of the second label.
- If examples are well separated, this is easy to do. But in some cases, it's going to be more complicated because some of the examples may be very close to one another. To avoid overfitting.
- 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

Some Examples of Classifying and Clustering - Cont'd

Suppose Data Was Labeled



Some Examples of Classifying and Clustering - Cont'd

Adding Some New Data

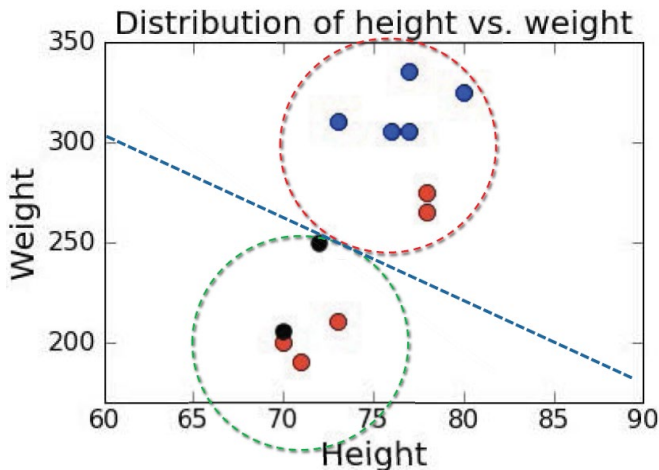
- Suppose we have learned to separate Group I versus Group II
- Now we are given some running backs, and want to use model to decide if they are more like Group I or Group II:

Catherine = ['point guard ', 72, 250]

James = ['power forward ', 70, 205]

Some Examples of Classifying and Clustering - Cont'd

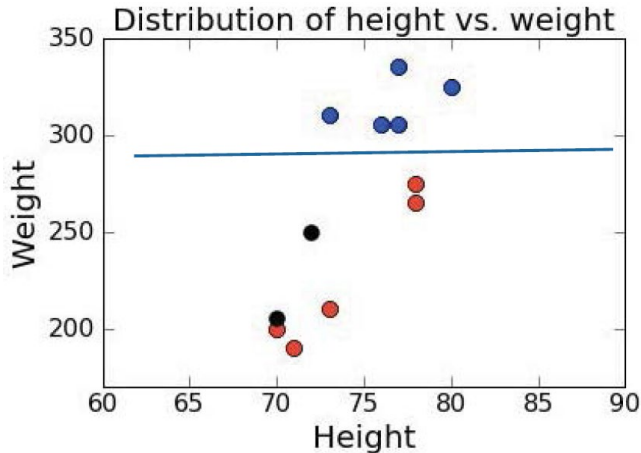
Adding Some New Data – unlabeled data



Maybe clustering is wrong? Or maybe not two clusters, maybe three?

Some Examples of Classifying and Clustering - Cont'd

Adding Some New Data – Classified using Labeled Data



Both of those new examples are clearly below the dividing line. They are clearly examples that I would categorize as being more like Group I than they are like Group II

Machine Learning Methods

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

Machine Learning Methods - Cont'd

All Machine Learning Methods Require 5 essential components :

- Choosing **training data** and evaluation method
 - Representation of the features (e.g., height, weight)
 - Distance metric for feature vectors (how to measure distance)
 - Objective function and constraints
 - Optimization method for learning the model
- } More advanced topics, not covered in this course

Feature Representation

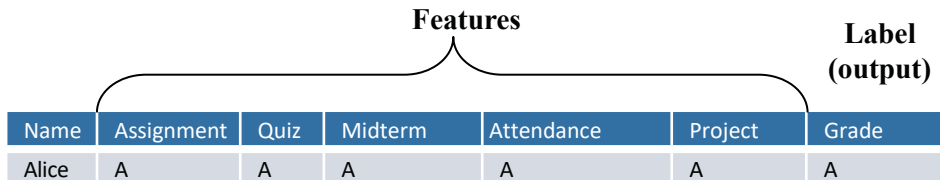


- Let's start talking about features.
- Features never fully describe the situation
 - "All models are wrong, but some are useful."
 - George Box (1919-2013) British statistician
 - "one of the great statistical minds of the 20th century"
- Feature engineering – which features are useful
 - I've got a set of examples, labeled or not. I need to decide what is it about those examples that's useful to use
 - 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 in AF3214
 - Let's just build a little learning algorithm that takes a set of data and predicts your final grade.
 - Some features surely helpful, e.g., GPA, prior Python experience
 - Others might cause me to overfit, corr between these, e.g., birth date, gender
 - Want to maximize ratio of useful input to irrelevant input:
Signal-to-Noise Ratio (SNR)

Feature Representation

- Overfitting - occurs **when a statistical model fits exactly against its training data**. When the model memorizes the noise and fits too closely to the training set, the model becomes “overfitted”
- Not to create a really complicated surface to separate things
- So we may have to tolerate a few incorrectly labeled things
- And it is unable to generalize well to new data, or the test data
- See example later

An Example - Course Evaluation



Features						Label (output)
Name	Assignment	Quiz	Midterm	Attendance	Project	Grade
Alice	A	A	A	A	A	A


Initial model:

- Not enough information to generalize

An Example - Course Evaluation - Cont'd


Features

**Label
(output)**



Name	Assignment	Quiz	Midterm	Attendance	Project	Grade
Alice	A	A	A	A	A	A
Bob	A	A	A	A	A	A

An Example - Course Evaluation - Cont'd



Features						Label (output)
Name	Assignment	Quiz	Midterm	Attendance	Project	Grade
Alice	A	A	A	A	A	A
Bob	A	A	A	A	A	A
Carmen	B	A	C	B	A	A

Carmen doesn't fit model, but is labeled as A. Need to refine model

An Example - Course Evaluation - Cont'd

Features

Label (output)

Name	Assignment	Quiz	Midterm	Attendance	Project	Grade
Alice	A	A	A	A	A	A
Bob	A	A	A	A	A	A
Carmen	B	A	C	B	A	A
Damon	A	C	B	A	B	B

An Example - Course Evaluation - Cont'd

Features

Label (output)

Name	Assignment	Quiz	Midterm	Attendance	Project	Grade
Alice	A	A	A	A	A	A
Bob	A	A	A	A	A	A
Carmen	B	A	C	B	A	A
Damon	A	C	B	A	B	B
Elliot	A	B	C	B	B	A
Frankie	C	C	C	B	C	C
George	A	A	A	B	A	A
Hannah	A	A	A	B	A	B

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

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

Supervised versus Unsupervised Learning

In the next few slides, we will see examples of learning algorithms:

1. 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
2. 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

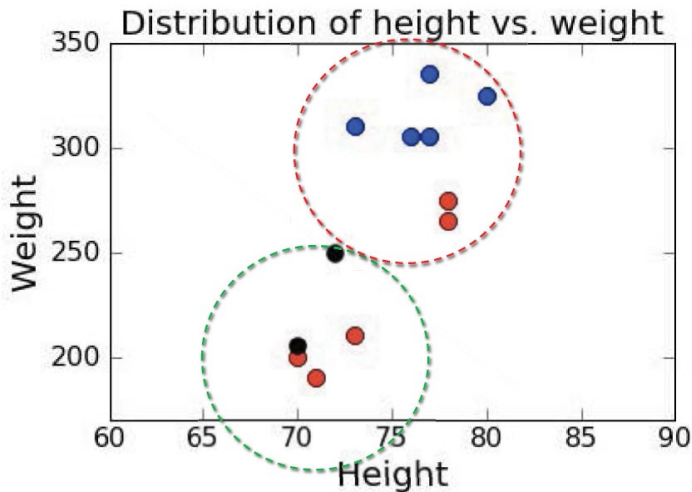
Learned models will depend on:

- Distance metric between examples
 - How do I measure distance between examples
- Choice of feature vectors
 - What is the right set of features to use in that vector?
- Constraints on complexity of model
 - What constraints do I want to put on the model
 - Specified number of clusters in case of unlabeled data
 - Complexity of separating surface in case of labeled data
 - Want to avoid over fitting problem (each example is its own cluster, or a complex separating surface)

Clustering Approaches

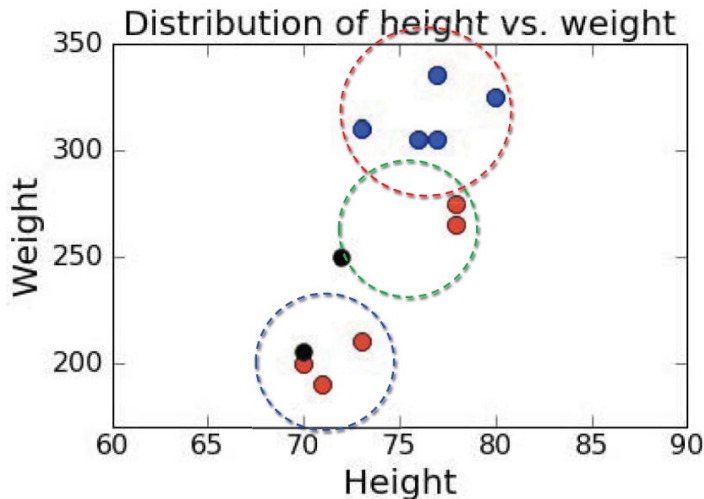
- Suppose we know that there are k different groups in our training data, but don't know labels
 - Pick k samples (*at random?*) as exemplars
 - Create clusters with the property that the distances between all of the examples of that cluster are small.
 - Find clusters the average distance for both clusters as small as possible.
 - Cluster remaining samples by minimizing distance between samples in same cluster(**objective function**)-put sample in group with closest exemplar
 - Once clusters fixed, find median element/example in each cluster as new exemplar (the one closest to the center, and treat it as exemplars)
 - Repeat above steps until no change in the process
- 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



Black dots are too close. What about three clusters?

Fitting Three Clusters Unsupervised



Nice grouping of 3, 4, and 5

The average distance between examples in each of these clusters, it is much tighter than in previous example.

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. A sequence of line segments that separates them out. Because there's not just one line that does the separation
 - ✓ Use voting schemes
 - Find k nearest training examples, use majority vote to select label. For every new example, the five closest labeled examples. And take a vote. If 3 out of 5, 4 out of 5, or 5 out of 5 of those labels are the same, I'm going to say it's part of that group. If I have less than that, I'm going to leave it as unclassified
- Issues:
 - ✓ How do we avoid over-fitting to data?
 - ✓ How do we measure performance?
 - ✓ How do we select best features?

Classification

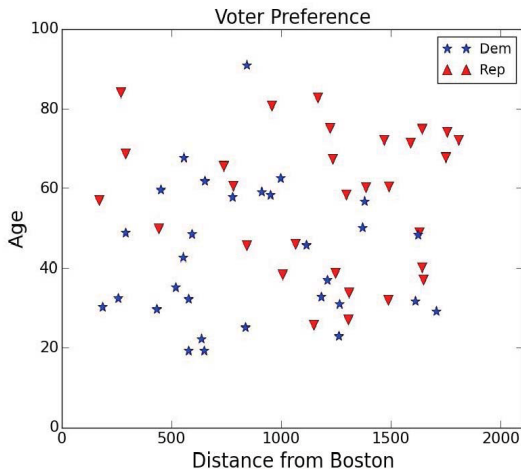
- Attempt to minimize error on training data
Similar to fitting a curve to data
- Evaluate on training data

These are a set of voters in the United States with their preference. They tend to vote Republican. They tend to vote Democrat. And the two categories are their age and how far away they live from Boston

How would I fit a curve to
separate those two classes?

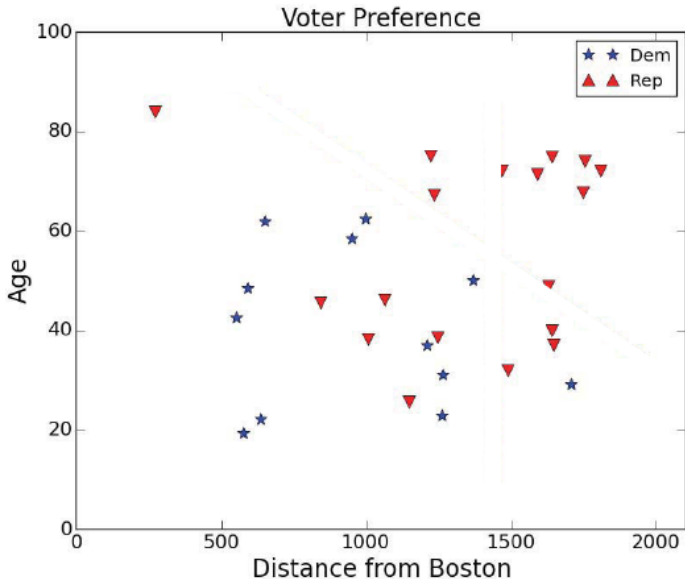
half data to test, half data to train

Voter preference, by age
and distance from Boston



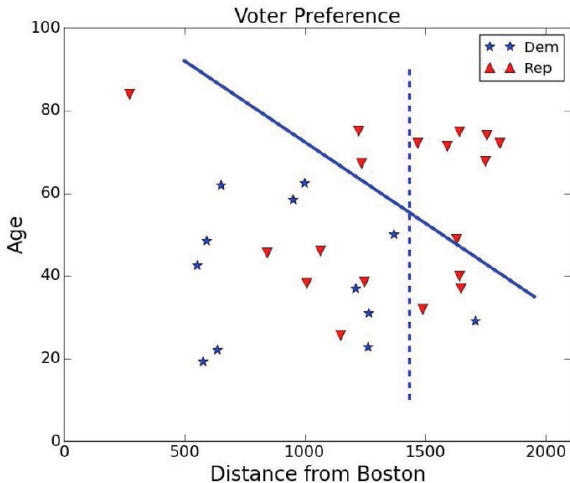
Randomly Divide Data into Training Set

Training data



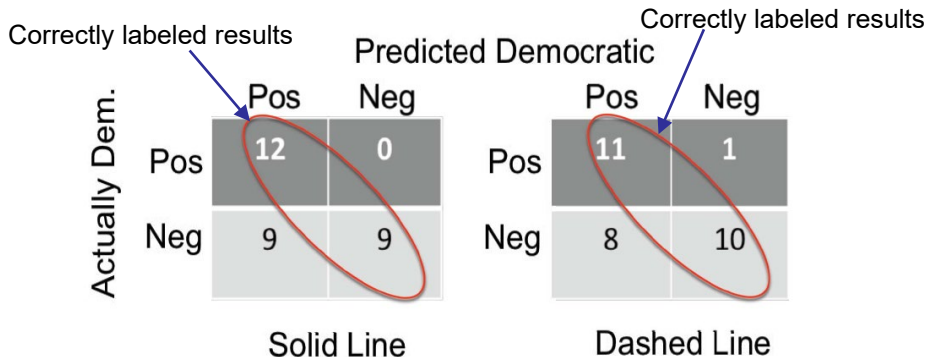
Two Possible Models for a Training Set

The fact that left and right correlates with distance from Boston is completely irrelevant here. But it has a nice punch to it.



So now the question is, how would I evaluate these?
How do I decide which one is better?

Confusion Matrices (Training Error)



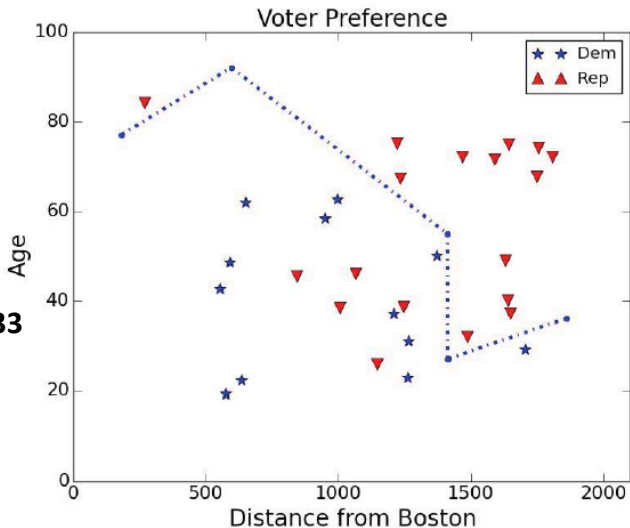
$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}$$

0.7 for both models

- Which one is better?
- Can we find a model with less training error?

A More Complex Model

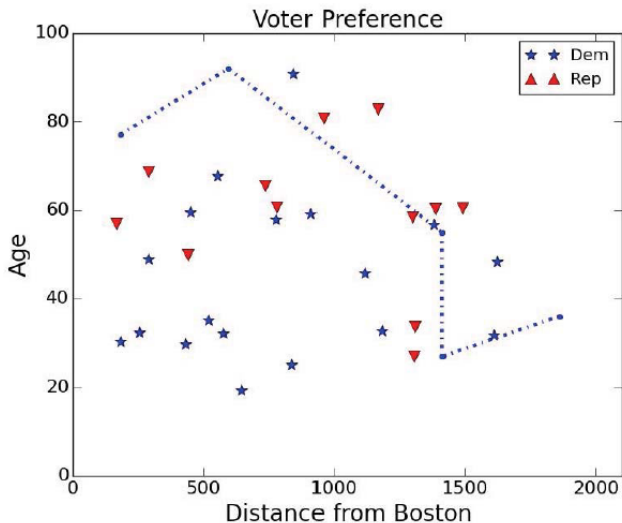
True Pos(Rep) = 12,
False Pos(Rep) = 5,
True Neg(Dem) = 13,
False Neg(Dem) = 0,
Accuracy = $25/30 = 0.833$



Applying Model to Test Data

Test data

True Pos = 14,
False Pos = 4,
True Neg = 4,
False Neg = 8,
Accuracy = $18/30 = 0.6$



Other Statistical Measures

PPV, Positive Predictive Value, which is how many true positives do I come up with out of all the things I labeled positively.

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

- Solid line model: 0.57
- Dashed line model: 0.58
- Complex model, training: 0.71
- Complex model, testing: 0.78
- You will also see "**sensitivity**" versus "**specificity**"

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

Percentage
correctly found

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

Percentage
correctly rejected

This is where the trade-off comes in, where I can increase specificity at the cost of sensitivity or vice versa..

Short 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

The End