Introduction to Machine Learning

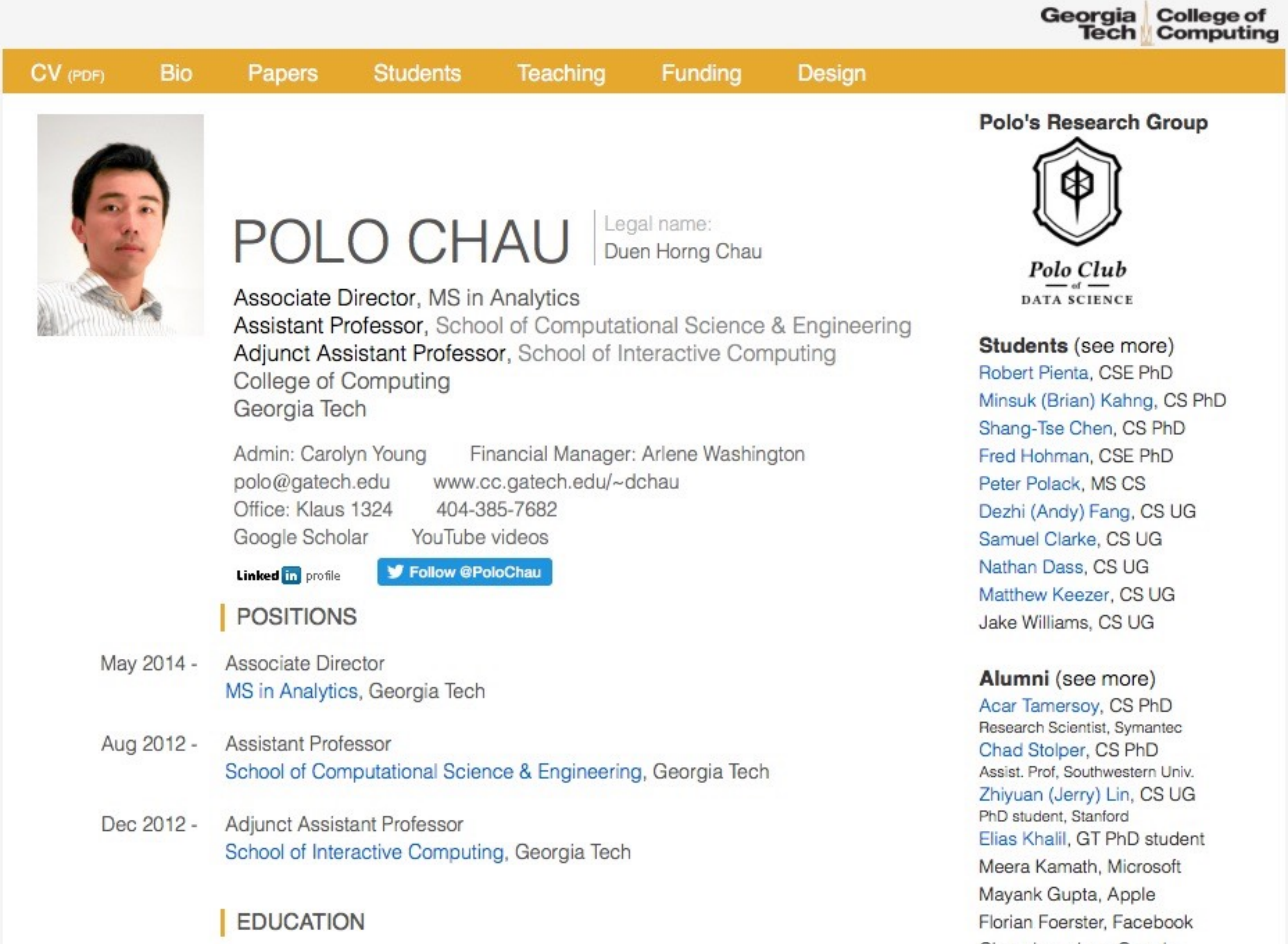
Duen Horng (Polo) Chau

Associate Director, MS Analytics

Assistant Professor, CSE, College of Computing

Georgia Tech

Google “Polo Chau” if interested in my professional life.



POLO CHAU

Legal name:
Duen Horng Chau

Associate Director, MS in Analytics
Assistant Professor, School of Computational Science & Engineering
Adjunct Assistant Professor, School of Interactive Computing
College of Computing
Georgia Tech

Admin: Carolyn Young	Financial Manager: Arlene Washington
polo@gatech.edu	www.cc.gatech.edu/~dchau
Office: Klaus 1324	404-385-7682
Google Scholar	YouTube videos

POSITIONS

May 2014 -	Associate Director MS in Analytics, Georgia Tech
Aug 2012 -	Assistant Professor School of Computational Science & Engineering, Georgia Tech
Dec 2012 -	Adjunct Assistant Professor School of Interactive Computing, Georgia Tech

EDUCATION

Polo's Research Group

Students (see more)

Robert Pienta, CSE PhD
Minsuk (Brian) Kahng, CS PhD
Shang-Tse Chen, CS PhD
Fred Hohman, CSE PhD
Peter Polack, MS CS
Dezhi (Andy) Fang, CS UG
Samuel Clarke, CS UG
Nathan Dass, CS UG
Matthew Keezer, CS UG
Jake Williams, CS UG

Alumni (see more)

Acar Tamersoy, CS PhD
Research Scientist, Symantec
Chad Stolper, CS PhD
Assist. Prof, Southwestern Univ.
Zhiyuan (Jerry) Lin, CS UG
PhD student, Stanford
Elias Khalil, GT PhD student
Meera Kamath, Microsoft
Mayank Gupta, Apple
Florian Foerster, Facebook

Every semester, Polo teaches…

CSE6242 / CX4242

Data & Visual Analytics

http://poloclub.gatech.edu/cse6242

(all lecture slides and homework assignments posted online)

Polo Club
of
DATA SCIENCE

Scalable. Interactive. Interpretable.

We bridge and innovate at the intersection of **data mining** and **human-computer interaction (HCI)** to synthesize **scalable, interactive, and interpretable** tools that amplify human's ability to understand and interact with big data.

What you will see next comes from:

1. **10 Lessons Learned from Working with Tech Companies**
   https://www.cc.gatech.edu/~dchau/slides/data-science-lessons-learned.pdf
2. CSE6242 "**Classification key concepts**"
   http://poloclub.gatech.edu/cse6242/2017spring/slides/CSE6242-13-Classification.pdf
3. CSE6242 "**Intro to clustering; DBSCAN**"
   http://poloclub.gatech.edu/cse6242/2017spring/slides/CSE6242-16-Classification-Vis.pdf

(**Lesson 1** from **“10 Lessons Learned from Working with Tech Companies”**)

Machine Learning is one of the **many things** you *should* learn.

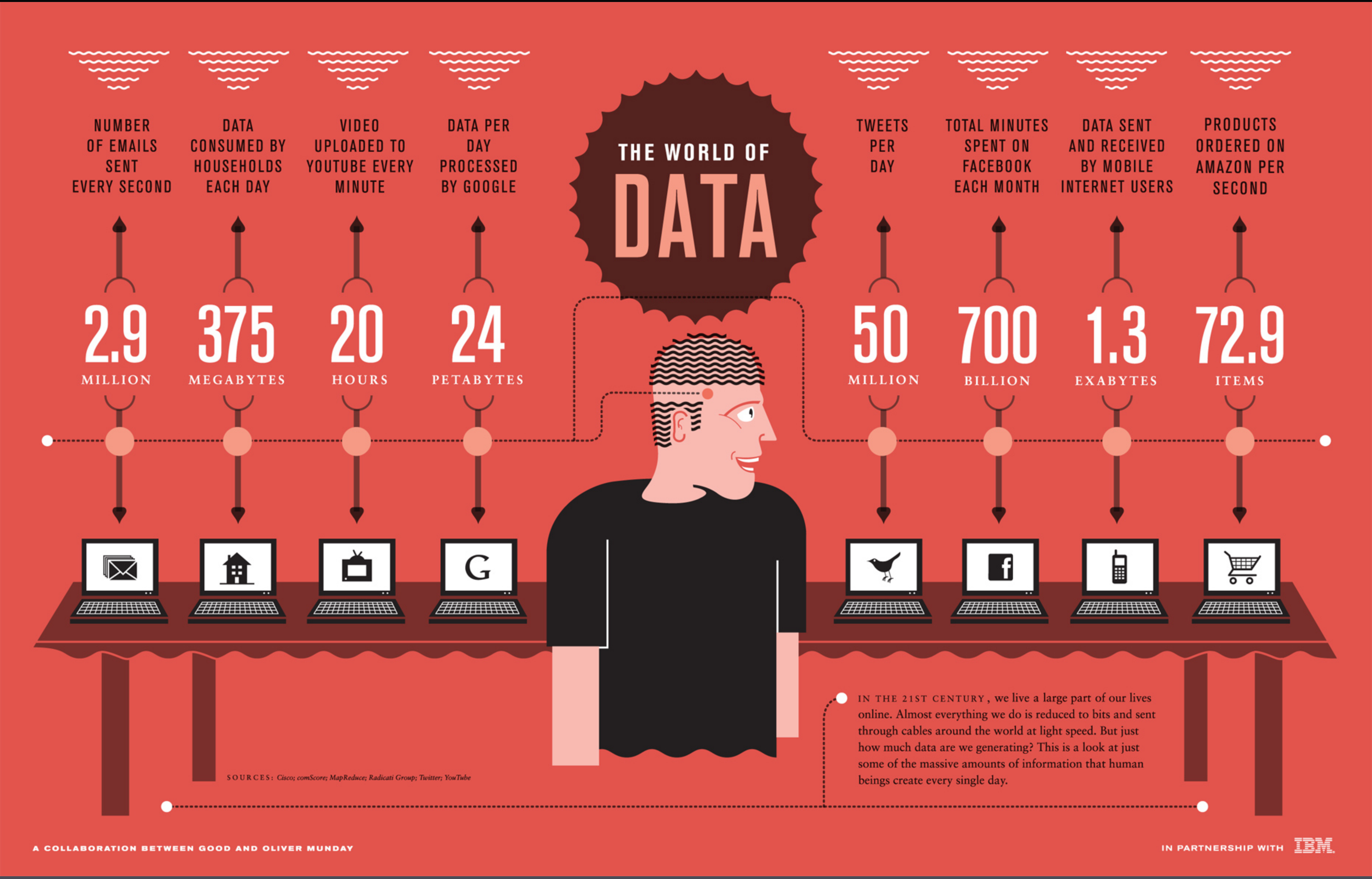
Many companies are looking for *data scientists*, *data analysts*, etc.

Good news! Many jobs!

Most companies looking for “data scientists”

> *The data scientist role is critical for organizations looking to extract insight from information assets for ‘big data’ initiatives and requires a **broad combination** of skills that may be fulfilled better as a team*
> - Gartner (http://www.gartner.com/it-glossary/data-scientist)

Breadth of knowledge is important.



THE WORLD OF DATA

NUMBER OF EMAILS SENT EVERY SECOND
2.9 MILLION

DATA CONSUMED BY HOUSEHOLDS EACH DAY
375 MEGABYTES

VIDEO UPLOADED TO YOUTUBE EVERY MINUTE
20 HOURS

DATA PER DAY PROCESSED BY GOOGLE
24 PETABYTES

TWEETS PER DAY
50 MILLION

TOTAL MINUTES SPENT ON FACEBOOK EACH MONTH
700 BILLION

DATA SENT AND RECEIVED BY MOBILE INTERNET USERS
1.3 EXABYTES

PRODUCTS ORDERED ON AMAZON PER SECOND
72.9 ITEMS

IN THE 21ST CENTURY, we live a large part of our lives online. Almost everything we do is reduced to bits and sent through cables around the world at light speed. But just how much data are we generating? This is a look at just some of the massive amounts of information that human beings create every single day.

SOURCES: *Cisco; comScore; MapReduce; Radicati Group; Twitter; YouTube*

A COLLABORATION BETWEEN GOOD AND OLIVER MUNDAY

IN PARTNERSHIP WITH IBM

http://spanning.com/blog/choosing-between-storage-based-and-unlimited-storage-for-cloud-data-backup/

What are the “ingredients”?

Need to think (a lot) about: storage, complex system design, scalability of algorithms, visualization techniques, interaction techniques, statistical tests, etc.

Analytics Building Blocks

Collection

Cleaning

Integration

Analysis

Visualization

Presentation

Dissemination

Building blocks, not “steps”

Collection

Cleaning

Integration

Analysis

Visualization

Presentation

Dissemination

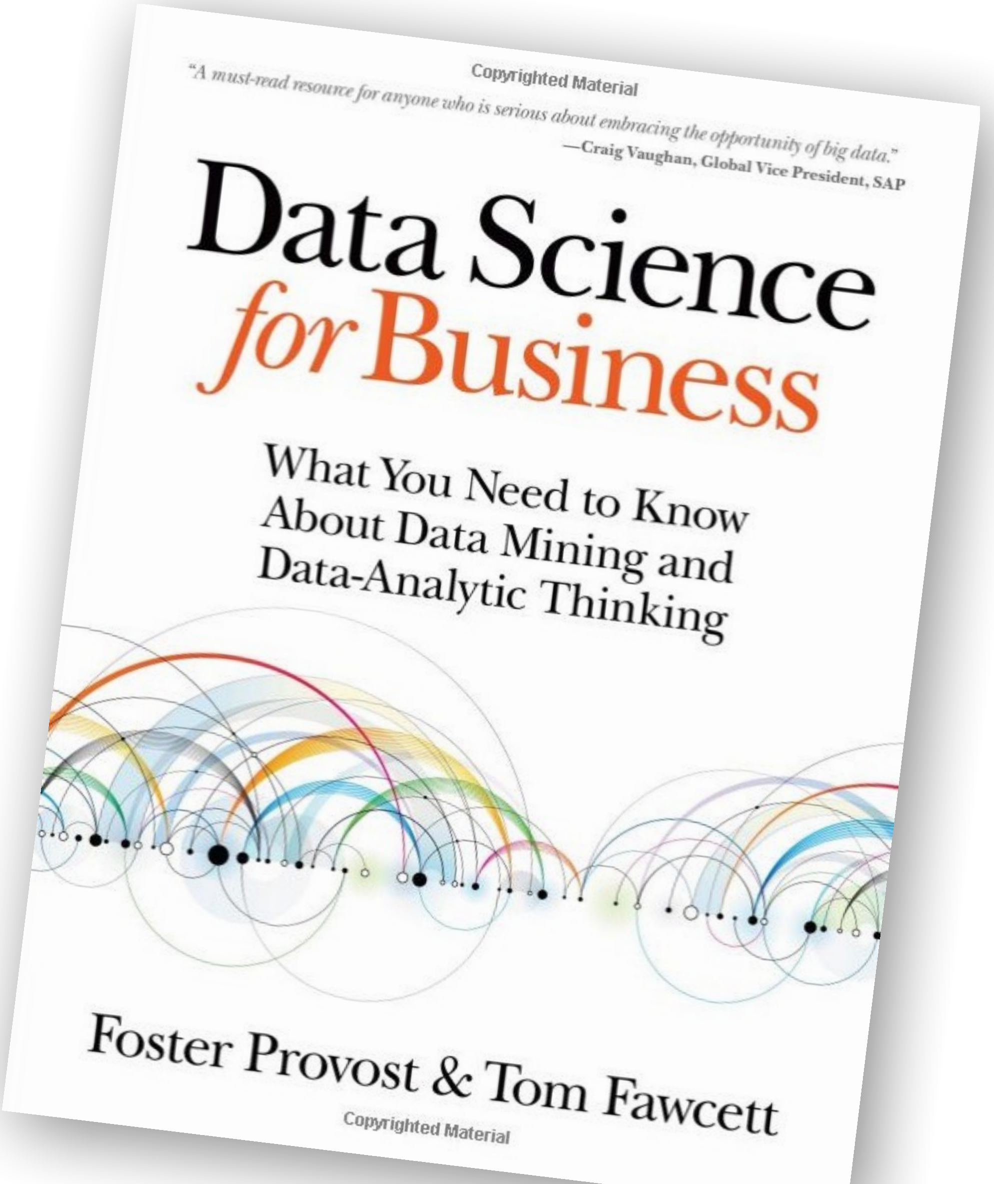
- Can skip some
- Can go back (two-way street)
- Examples
  - Data types inform visualization design
  - Data informs choice of algorithms
  - Visualization informs data cleaning (dirty data)
  - Visualization informs algorithm design (user finds that results don’t make sense)

(**Lesson 2** from **"10 Lessons Learned from Working with Tech Companies"**)

Learn **data science concepts** and key **generalizable techniques** to **future-proof** yourselves.

And here's a good book.

A critical skill in data science is the ability to decompose a data-analytics problem into pieces such that each piece matches a known task for which tools are available. Recognizing familiar problems and their solutions avoids wasting time and resources reinventing the wheel. It also allows people to focus attention on more interesting parts of the process that require human involvement—parts that have not been automated, so human creativity and intelligence must come into play.


Copyrighted Material
"A must-read resource for anyone who is serious about embracing the opportunity of big data."
—Craig Vaughan, Global Vice President, SAP
Data Science for Business
What You Need to Know About Data Mining and Data-Analytic Thinking
Foster Provost & Tom Fawcett
Copyrighted Material

http://www.amazon.com/Data-Science-Business-data-analytic-thinking/dp/1449361323

1. Classification

(or Probability Estimation)

Predict which of a (small) set of classes an entity belong to.

- email spam (y, n)
- sentiment analysis (+, -, neutral)
- news (politics, sports, …)
- medical diagnosis (cancer or not)
- face/cat detection
    - face detection (baby, middle-aged, etc)
- buy /not buy - commerce
- fraud detection

2. Regression ("value estimation")

Predict the **numerical value** of some variable for an entity.

- stock value
- real estate
- food/commodity
- sports betting
- movie ratings
- energy

3. Similarity Matching

Find similar entities (from a large dataset) based on what we know about them.

- price comparison (consumer, find similar priced)
- finding employees
- similar youtube videos (e.g., more cat videos)
- similar web pages (find near duplicates or representative sites) ~= clustering
- plagiarism detection

4. Clustering (unsupervised learning)

Group entities together by their similarity. (User provides # of clusters)

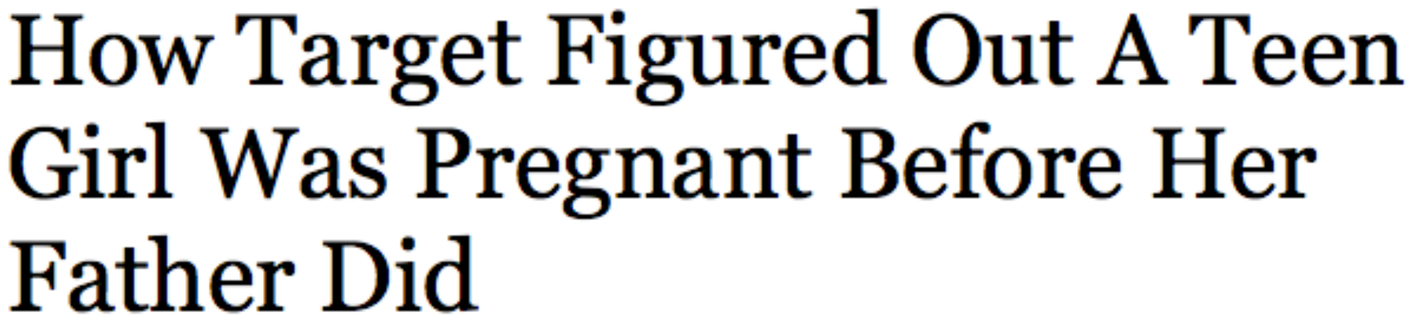
- groupings of similar bugs in code
- optical character recognition
    - unknown vocabulary
- topical analysis (tweets?)
- land cover: tree/road/…
- for advertising: grouping users for marketing purposes
- fireflies clustering
- speaker recognition (multiple people in same room)
- astronomical clustering

5. Co-occurrence grouping

(Many names: frequent itemset mining, association rule discovery, market-basket analysis)

Find associations between entities based on transactions that involve them

(e.g., bread and milk often bought together)



How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

http://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/

6. Profiling / Pattern Mining / Anomaly Detection (unsupervised)

Characterize **typical** behaviors of an entity (person, computer router, etc.) so you can find **trends** and **outliers**.

Examples?
computer instruction prediction
removing noise from experiment (data cleaning)
detect anomalies in network traffic
moneyball
weather anomalies (e.g., big storm)
google sign-in (alert)
smart security camera
embezzlement
trending articles

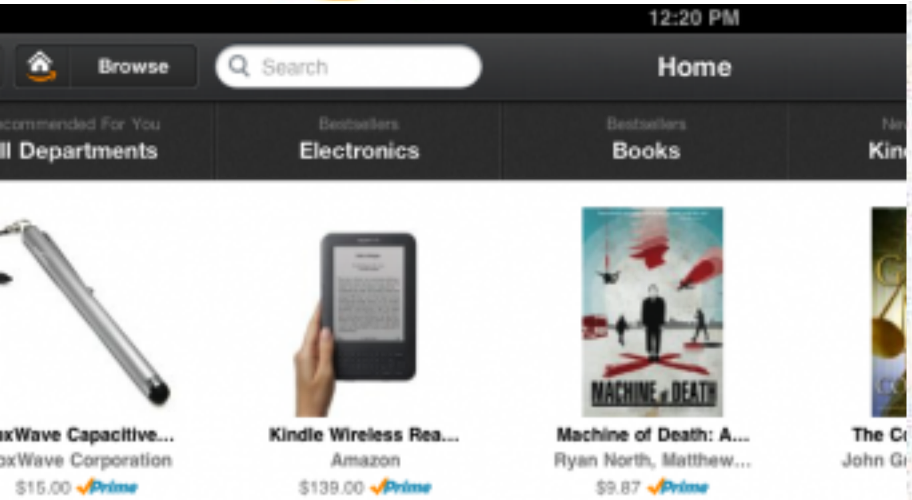
7. Link Prediction / Recommendation

Predict if two entities should be connected, and how strongly that link should be.

linkedin/facebook: people you may know

amazon/netflix: because you like terminator… suggest other movies you may also like

amazon.com



8. Data reduction (“dimensionality reduction”)

Shrink a large dataset into smaller one, with as little loss of information as possible

1. if you want to visualize the data (in 2D/3D)
2. faster computation/less storage
3. reduce noise

More examples

- **Similarity functions**: central to clustering algorithms, and some classification algorithms (e.g., k-NN, DBSCAN)
- **SVD** (singular value decomposition), for NLP (LSI), and for recommendation
- **PageRank** (and its personalized version)
- **Lag plots** for auto regression, and non-linear time series foresting

http://poloclub.gatech.edu/cse6242

CSE6242 / CX4242: Data & Visual Analytics

Classification Key Concepts

Duen Horng (Polo) Chau

Assistant Professor
Associate Director, MS Analytics
Georgia Tech

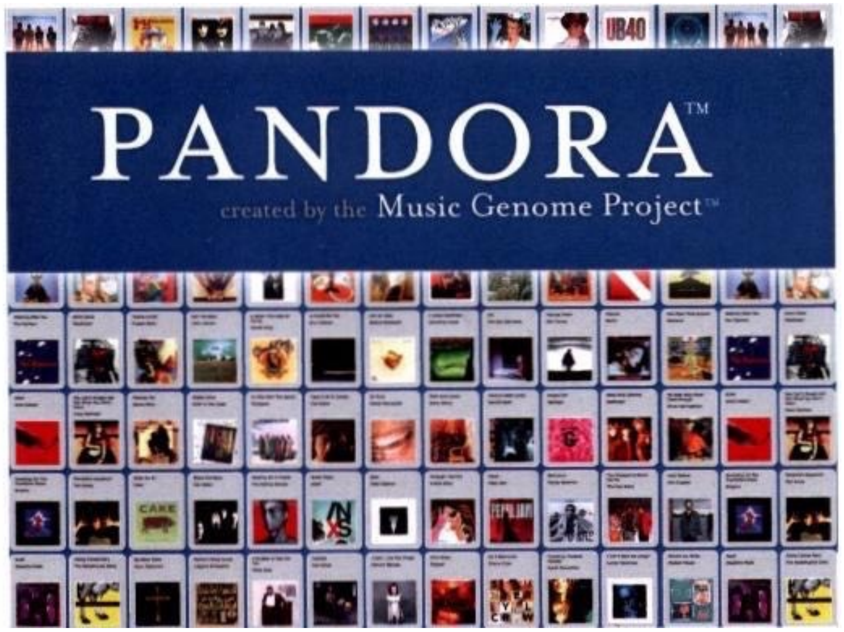
Parishit Ram

GT PhD alum; SkyTree

Partly based on materials by
Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos, Parishit Ram (GT PhD alum; SkyTree), Alex Gray

How will I rate "Chopin's 5th Symphony"?

🙂 ☹ 😐



Songs	**Like?**
Some nights	🙂
Skyfall	☹
Comfortably numb	😐
We are young	🙂
...	...
...	...
Chopin's 5th	???

Classification

What tools do you need for classification?

1. **Data** $S = \{(x_i, y_i)\}_{i = 1,...,n}$
    - ○ x_i : data example with d attributes
    - ○ y_i : label of example (what **you** care about)
2. Classification **model** $f_{(a,b,c,....)}$ with some parameters $a, b, c,...$
3. **Loss function** $L(y, f(x))$
    - ○ how to penalize mistakes

Terminology Explanation

data example = data instance

attribute = feature = dimension

label = target attribute

Data $S = \{(x_i, y_i)\}_{i = 1,\ldots,n}$

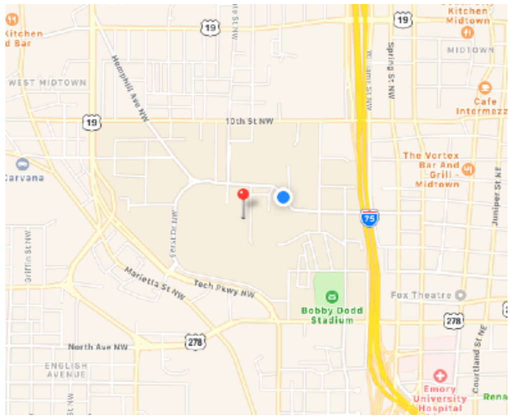
- x_i : data example with d attributes $x_i = (x_{i1}, \ldots, x_{id})$
- y_i : label of example

Song name	**Artist**	**Length**	**...**	**Like?**
Some nights	Fun	4:23	...	🙂
Skyfall	Adele	4:00	...	☹
Comf. numb	Pink Fl.	6:13	...	😐
We are young	Fun	3:50	...	🙂
...	...	...	...	...
...	...	...	...	...
Chopin's 5th	Chopin	5:32	...	??

What is a “model”?

“a simplified representation of reality created to serve a purpose” *Data Science for Business*

Example: maps are abstract models of the physical world



There can be many models!!

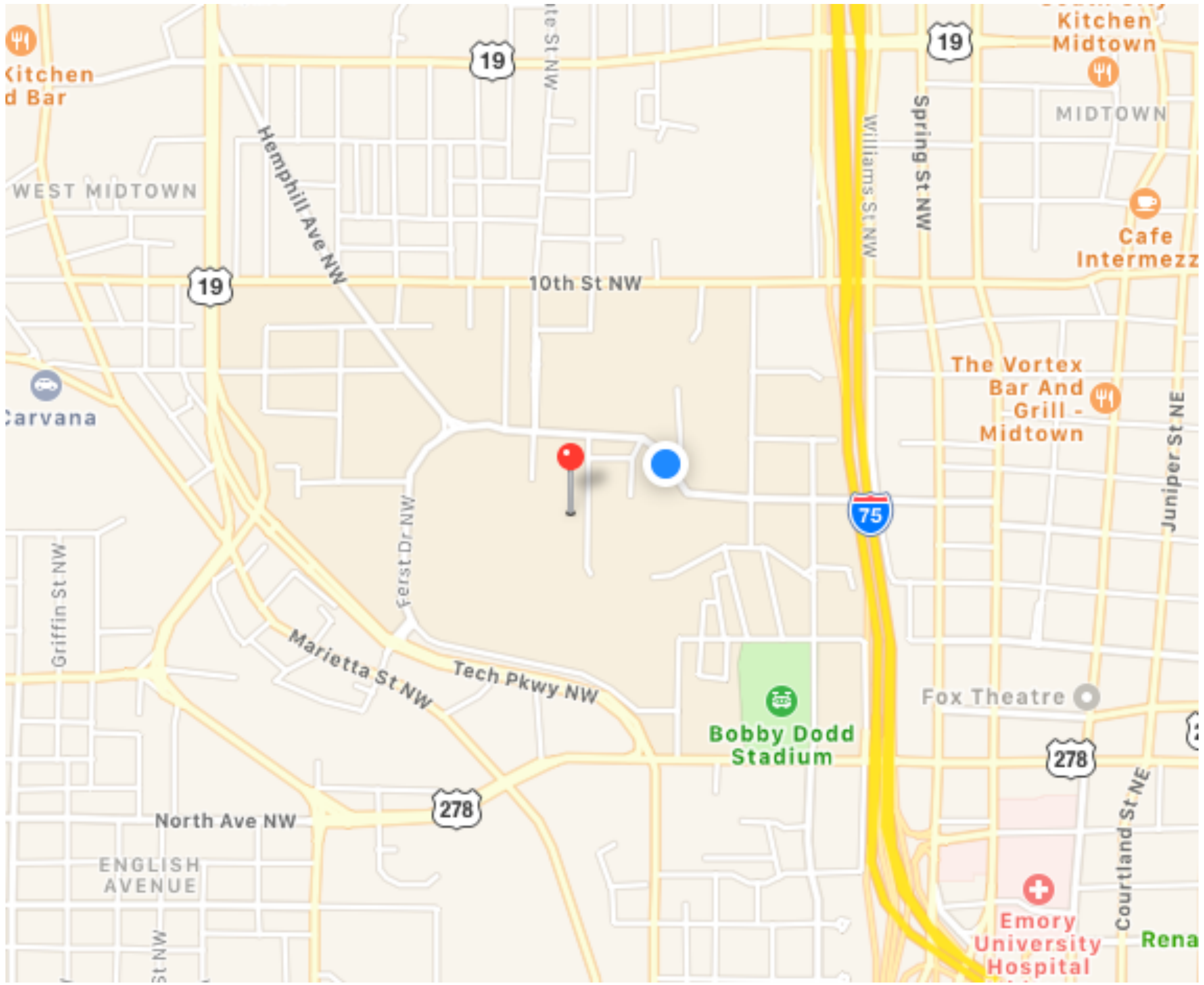
(Everyone sees the world differently, so each of us has a different model.)

In data science, a model is **formula to estimate what you care about**. The formula may be mathematical, a set of rules, a combination, etc.

Training a classifier = building the "model"

How do you learn appropriate values for parameters *a, b, c, ...* ?

Analogy: how do you know your map is a "good" map of the physical world?



Classification loss function

Most common loss: **0-1 loss function**

$$L_{0-1}(y, f(x)) = \mathbb{I}(y \neq f(x))$$

More general loss functions are defined by a *m x m* cost matrix *C* such that

$$L(y, f(x)) = C_{ab}$$

where $y = a$ and $f(x) = b$

Class	**T0**	**T1**
P0	0	C_{10}
P1	C_{01}	0

T0 (true class 0), **T1** (true class 1)

P0 (predicted class 0), **P1** (predicted class 1)

An ideal model should correctly estimate:

- known or seen data examples’ labels
- unknown or unseen data examples’ labels

Song name	Artist	Length	...	Like?
Some nights	Fun	4:23	...	🙂
Skyfall	Adele	4:00	...	☹
Comf. numb	Pink Fl.	6:13	...	😐
We are young	Fun	3:50	...	🙂
...	...	...	...	...
...	...	...	...	...
Chopin's 5th	Chopin	5:32	...	??

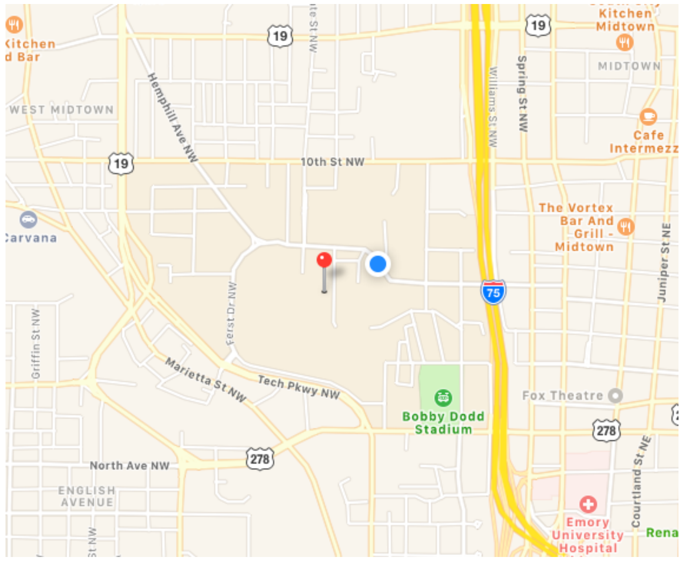
Training a classifier = building the "model"

Q: How do you learn appropriate values for parameters *a, b, c, ...* ?

(Analogy: how do you know your map is a "good" map?)

- $y_i = f_{(a,b,c,....)}(x_i), i = 1, ..., n$
  - Low/no error on training data ("seen" or "known")
- $y = f_{(a,b,c,....)}(x),$ for any new *x*
  - Low/no error on test data ("unseen" or "unknown")

It is very easy to achieve perfect classification on training/seen/known data. Why?



If your model works really well for *training* data, but poorly for *test* data, your model is “overfitting”.

How to avoid overfitting?

Example: one run of *5-fold* cross validation

You should do a **few runs** and **compute the average**
(e.g., error rates if that’s your evaluation metrics)

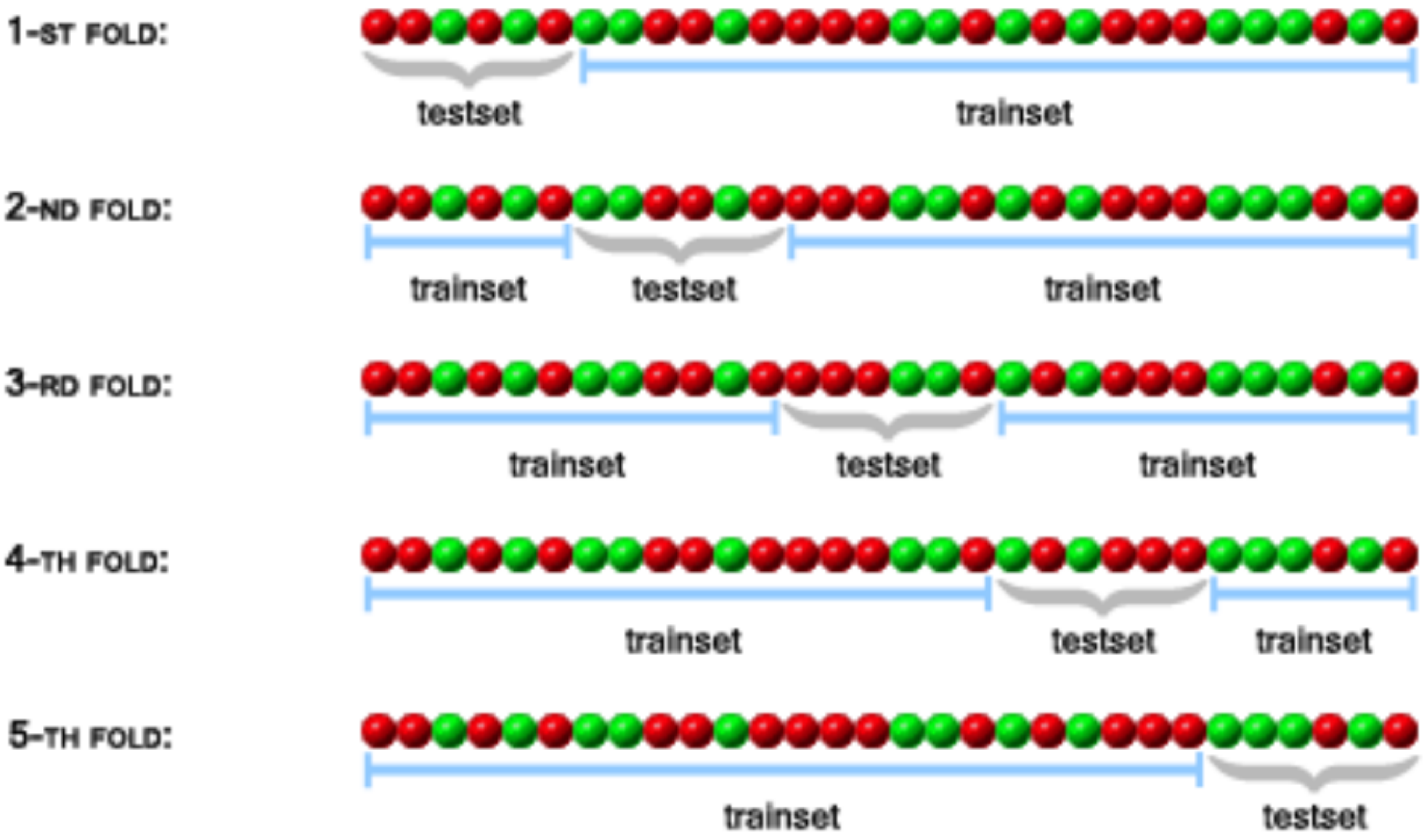


Image credit: http://stats.stackexchange.com/questions/1826/cross-validation-in-plain-english

Cross validation

1. Divide your data into n parts
2. Hold 1 part as “test set” or “hold out set”
3. Train classifier on remaining n-1 parts “training set”
4. Compute test error on test set
5. Repeat above steps n times, once for each n-th part
6. Compute the average test error over all n folds (i.e., cross-validation test error)

Cross-validation variations

Leave-one-out cross-validation (LOO-CV)

- test sets of size 1

K-fold cross-validation

- Test sets of size *(n / K)*
- K = 10 is most common (i.e., 10-fold CV)

Example:
k-Nearest-Neighbor classifier

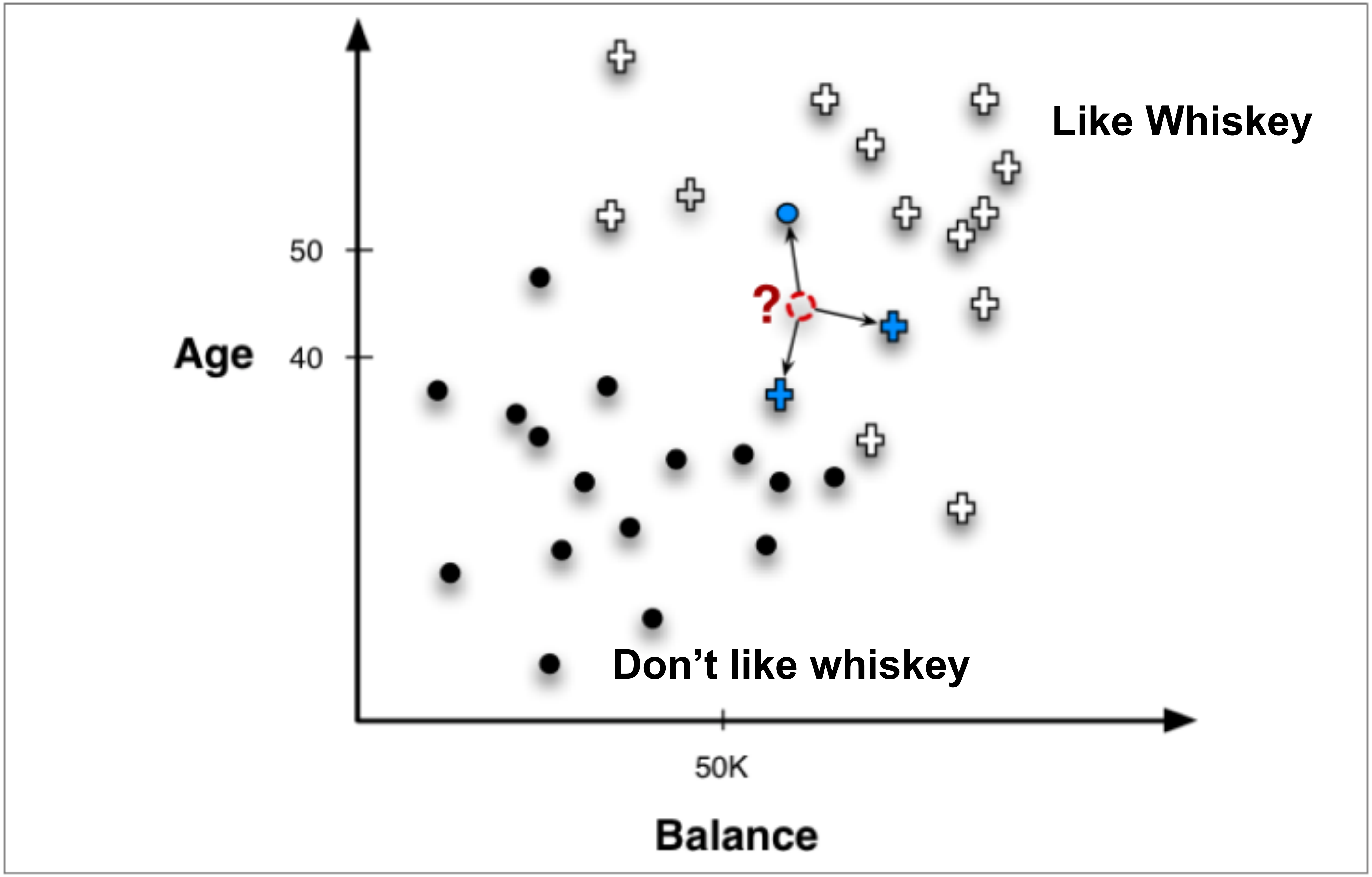
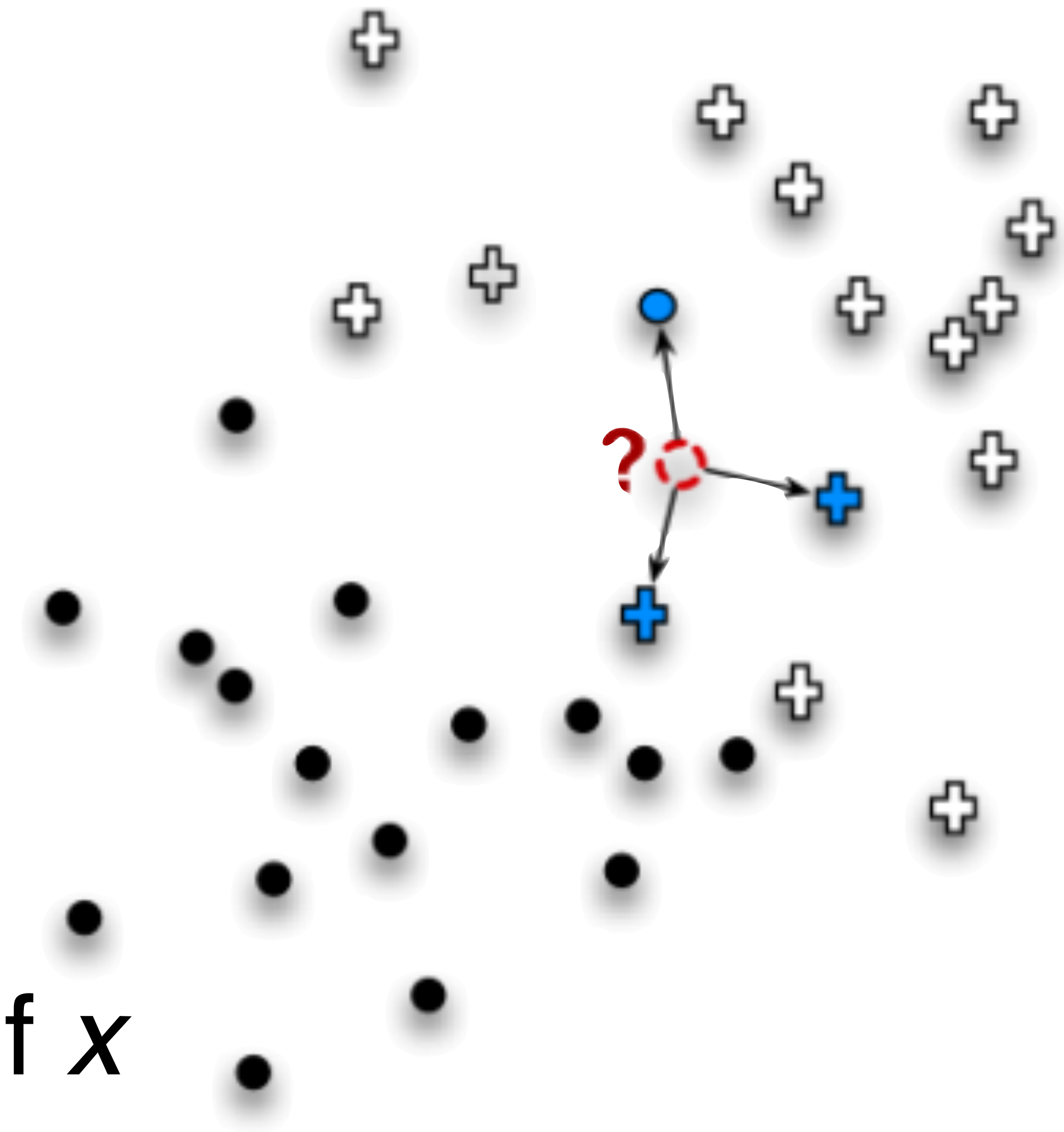


Figure 6-2. Nearest neighbor classification. The point to be classified, labeled with a question mark, would be classified + because the majority of its nearest (three) neighbors are +.

Image credit: Data Science for Business

k-Nearest-Neighbor Classifier



The classifier:

f(x) = majority label of the *k* nearest neighbors (NN) of *x*

Model parameters:

- Number of neighbors *k*
- Distance/similarity function *d(.,.)*

But k-NN is so simple!

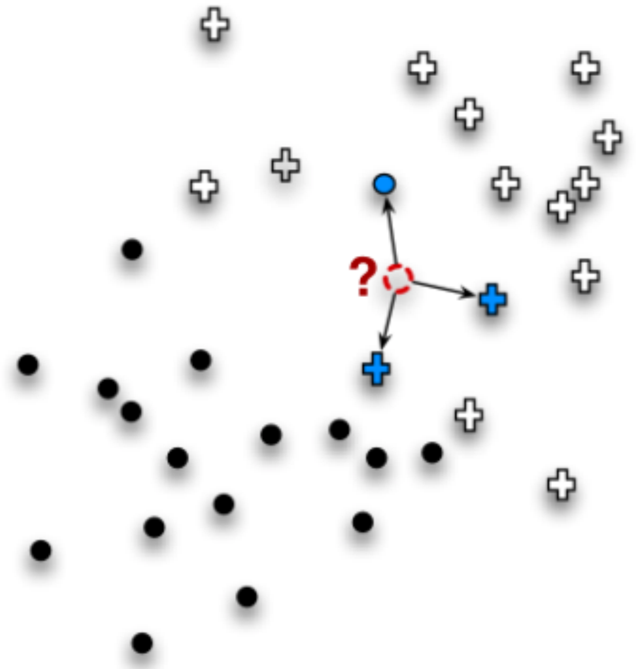
It can work really well! Pandora uses it or has used it: https://goo.gl/foLfMP
(from the book “Data Mining for Business Intelligence”)

Image credit: https://www.fool.com/investing/general/2015/03/16/will-the-music-industry-end-pandoras-business-mode.aspx

What are good models?

Simple (few parameters)	Effective	🤗
Complex (more parameters)	Effective (if significantly more so than simple methods)	🤔
Complex (many parameters)	Not-so-effective	😱

k-Nearest-Neighbor Classifier



If *k* and *d(.,.)* are fixed

Things to learn: ?

How to learn them: ?

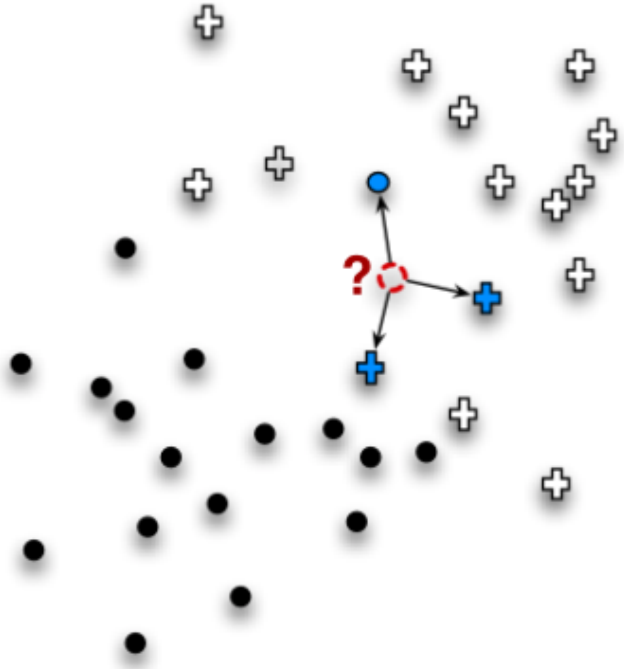
If *d(.,.)* is fixed, but you can change *k*

Things to learn: ?

How to learn them: ?

$x_i = (x_{i1}, \ldots, x_{id}); y_i = \{1, \ldots, m\}$

k-Nearest-Neighbor Classifier



If *k* and *d(.,.)* are fixed

Things to learn: Nothing

How to learn them: N/A

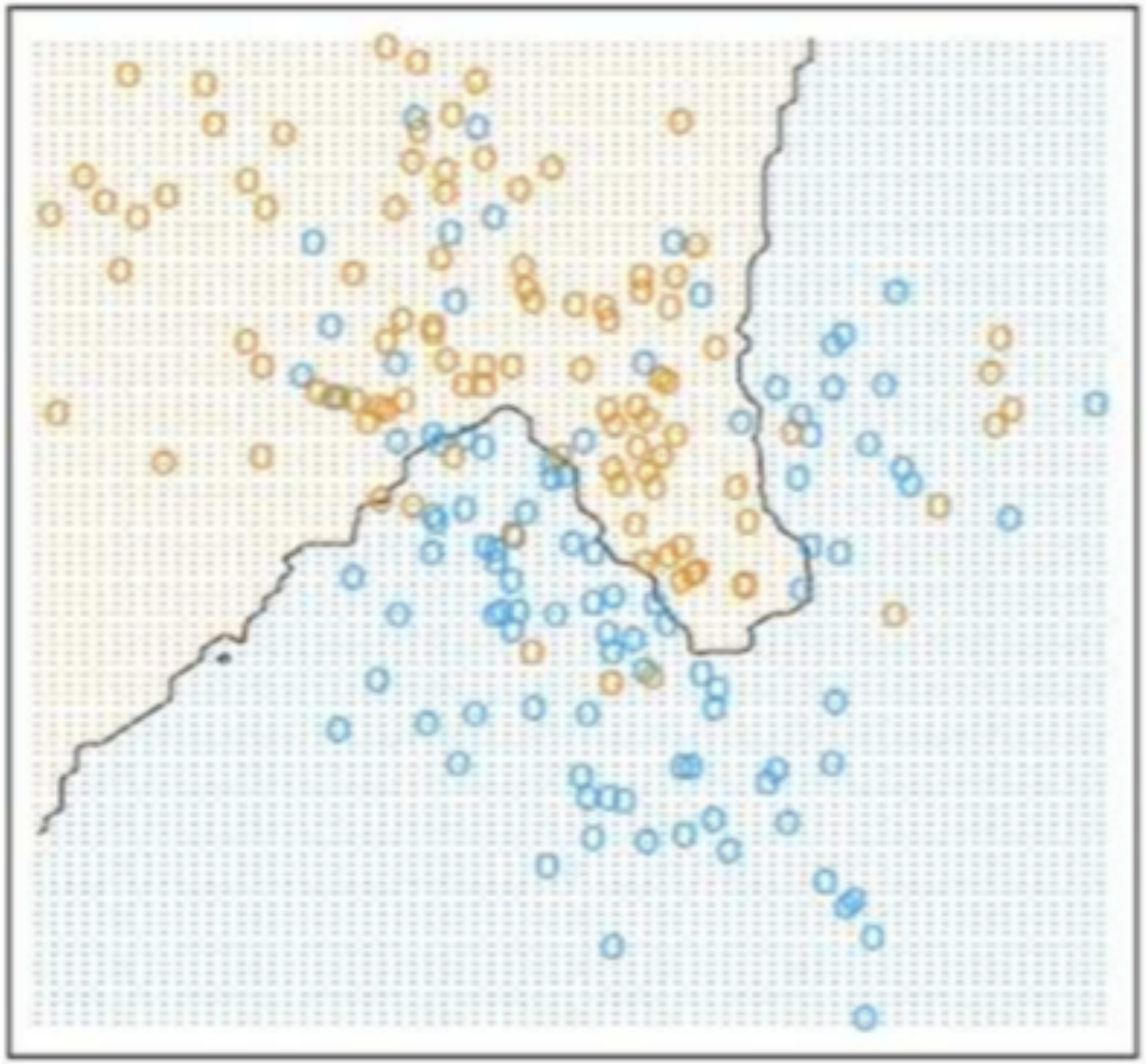
If *d(.,.)* is fixed, but you can change *k*

Selecting *k*: How?

How to find **best k** in k-NN?

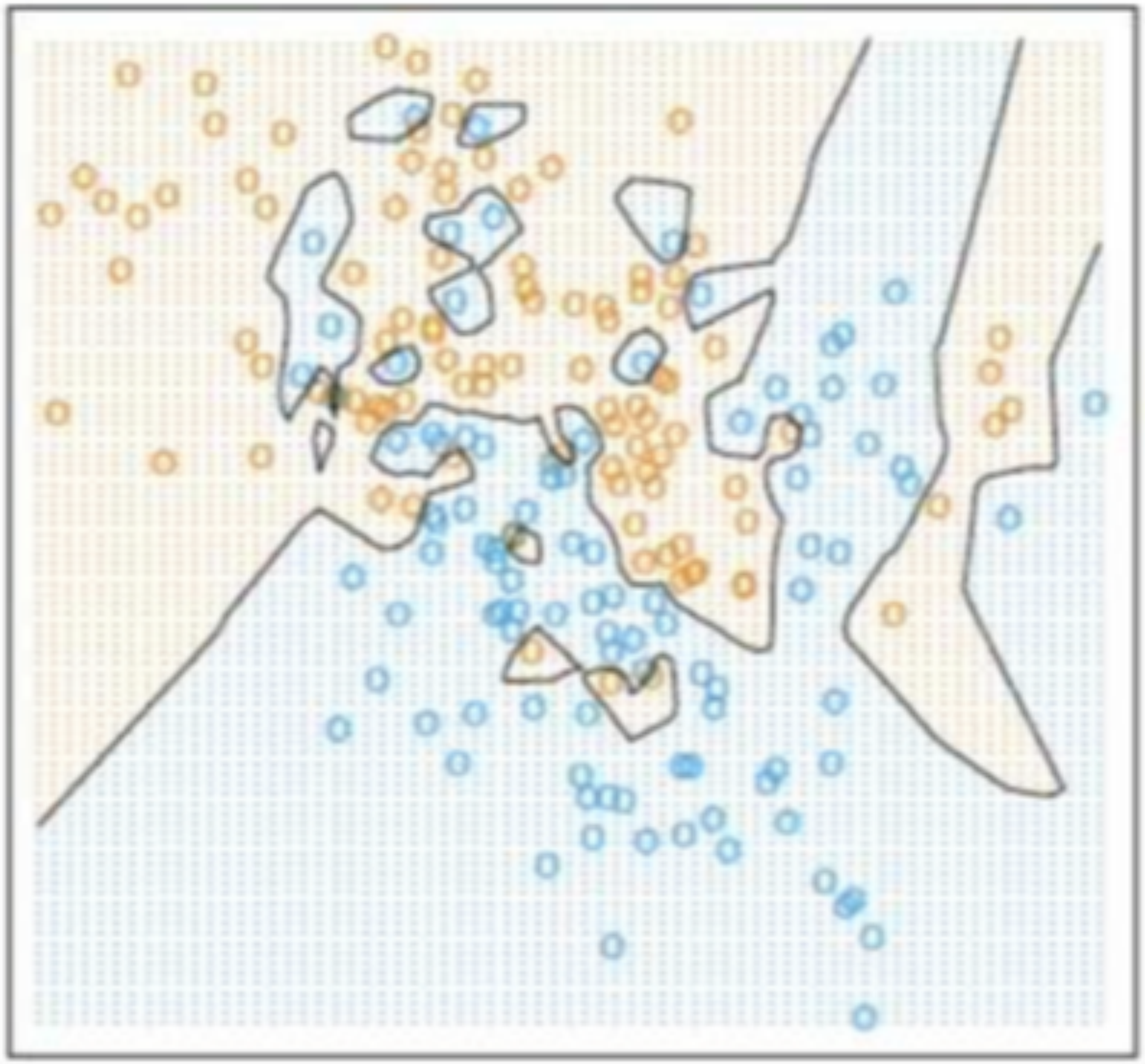
Use **cross validation (CV)**.

15-NN



Pretty good!

1-NN

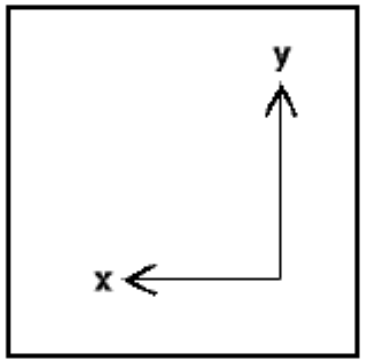


Overfitted

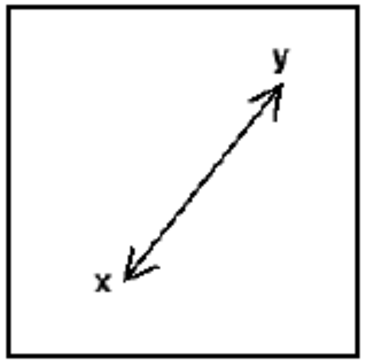
$x_i = (x_{i1}, \dots, x_{id}); y_i = \{1, \dots, m\}$

k-Nearest-Neighbor Classifier

If *k* is fixed, but you can change *d(.,.)*



Manhattan



Euclidean

Possible distance functions:

- Euclidean distance: $\|x_i - x_j\|_2 = \sqrt{(x_i - x_j)^\top (x_i - x_j)}$
- Manhattan distance: $\|x_i - x_j\|_1 = \sum_{l=1}^{d} |x_{il} - x_{jl}|$
- …

Summary on k-NN classifier

- Advantages
  - Little learning (unless you are learning the distance functions)
  - quite powerful in practice (and has theoretical guarantees as well)
- Caveats
  - Computationally expensive at test time

Reading material:

- ESL book, Chapter 13.3 http://www-stat.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf
- Le Song's slides on kNN classifier http://www.cc.gatech.edu/~lsong/teaching/CSE6740/lecture2.pdf

http://poloclub.gatech.edu/cse6242

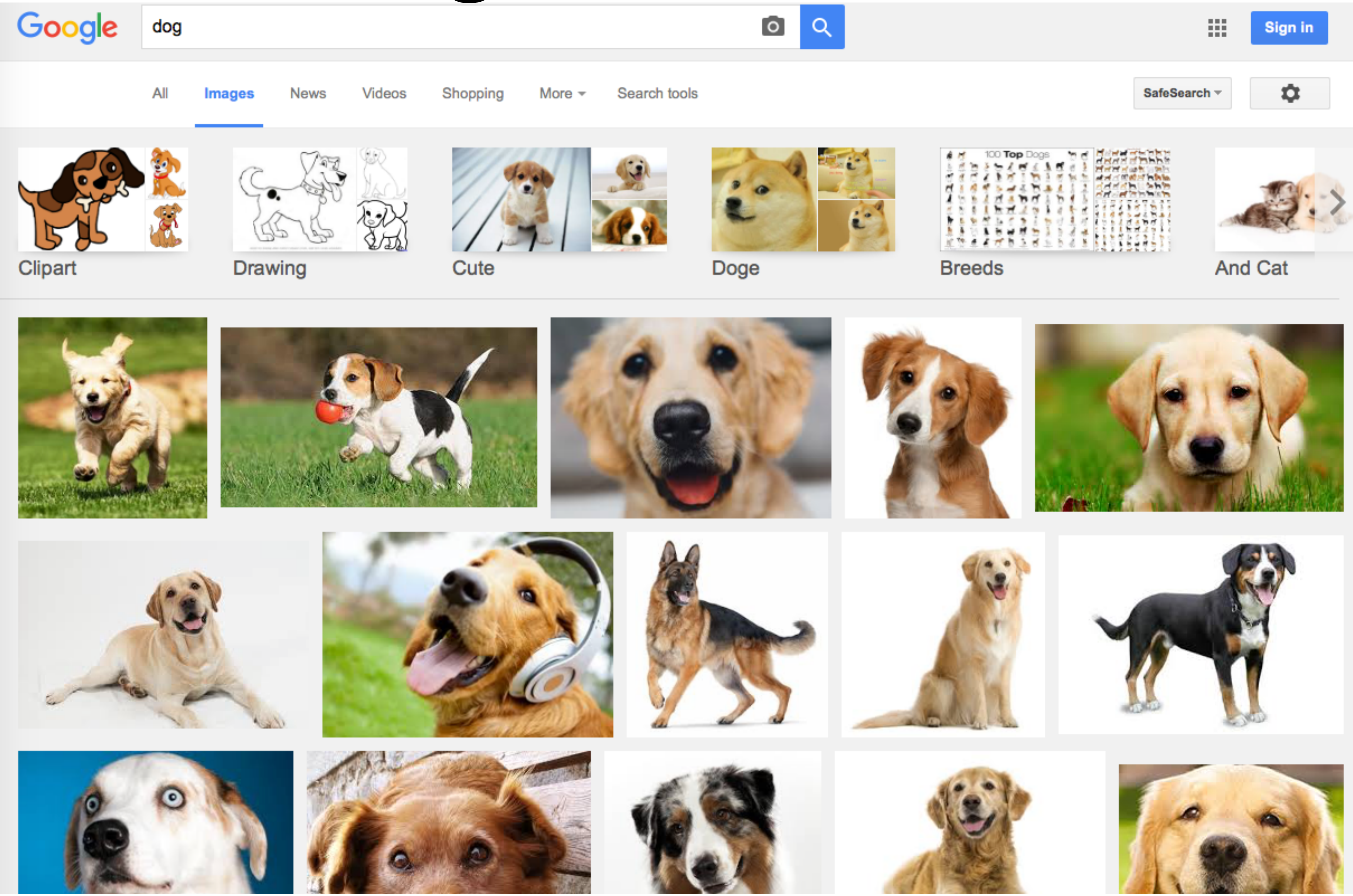
CSE6242 / CX4242: Data & Visual Analytics

Clustering

Duen Horng (Polo) Chau
Assistant Professor
Associate Director, MS Analytics
Georgia Tech

Partly based on materials by
Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos, Parishit Ram (GT PhD alum; SkyTree), Alex Gray

Clustering in Google Image Search



Video: http://youtu.be/WosBs0382SE

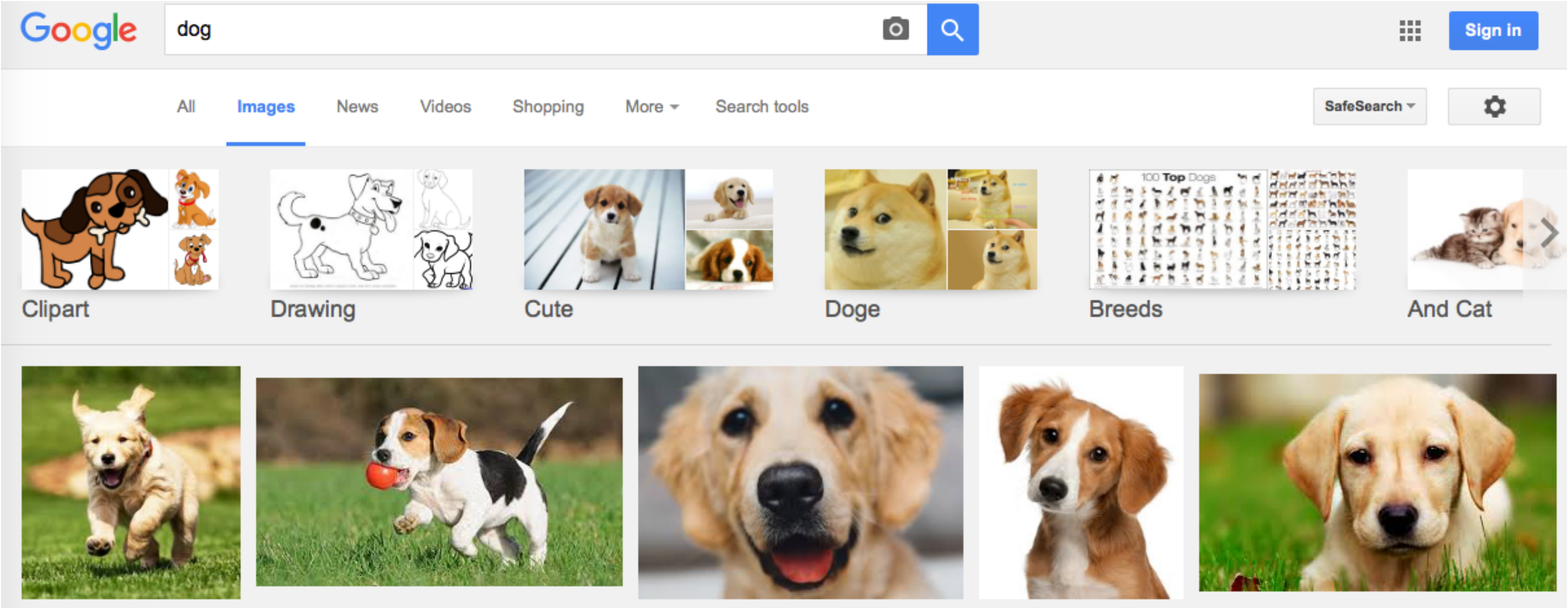
http://googlesystem.blogspot.com/2011/05/google-image-search-clustering.html

Clustering

The most common type of **unsupervised** learning

High-level idea: group **similar** things together

“**Unsupervised**” because clustering model is learned without any labeled examples



dog

Applications of Clustering

- Find similar patients subgroups
  - e.g., in healthcare
- Finding groups of similar text documents (topic modeling)
- …

Clustering techniques you’ve got to know

K-means

DBSCAN

(Hierarchical Clustering)

K-means (the “simplest” technique)

Best D3 demo Polo could find: http://tech.nitoyon.com/en/blog/2013/11/07/k-means/

Algorithm Summary

- We tell K-means the value of **k** (#clusters we want)
- **Randomly** initialize the k cluster “means” (“centroids”)
- **Assign** each item to the the cluster whose mean the item is closest to (so, we need a **similarity function**)
- **Update/recompute** the new “means” of all k clusters.
- If all items’ assignments do not change, **stop**.

YouTube video demo: https://youtu.be/IuRb3y8qKX4?t=3m4s

K-means What’s the catch?

http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

How to **decide k** (a hard problem)?

- A few ways; best way is to evaluate with real data (https://www.ee.columbia.edu/~dpwe/papers/PhamDN05-kmeans.pdf)

Only locally optimal (vs global)

- Different initialization gives different clusters
  - How to “fix” this?
- “Bad” starting points can cause algorithm to converge slowly

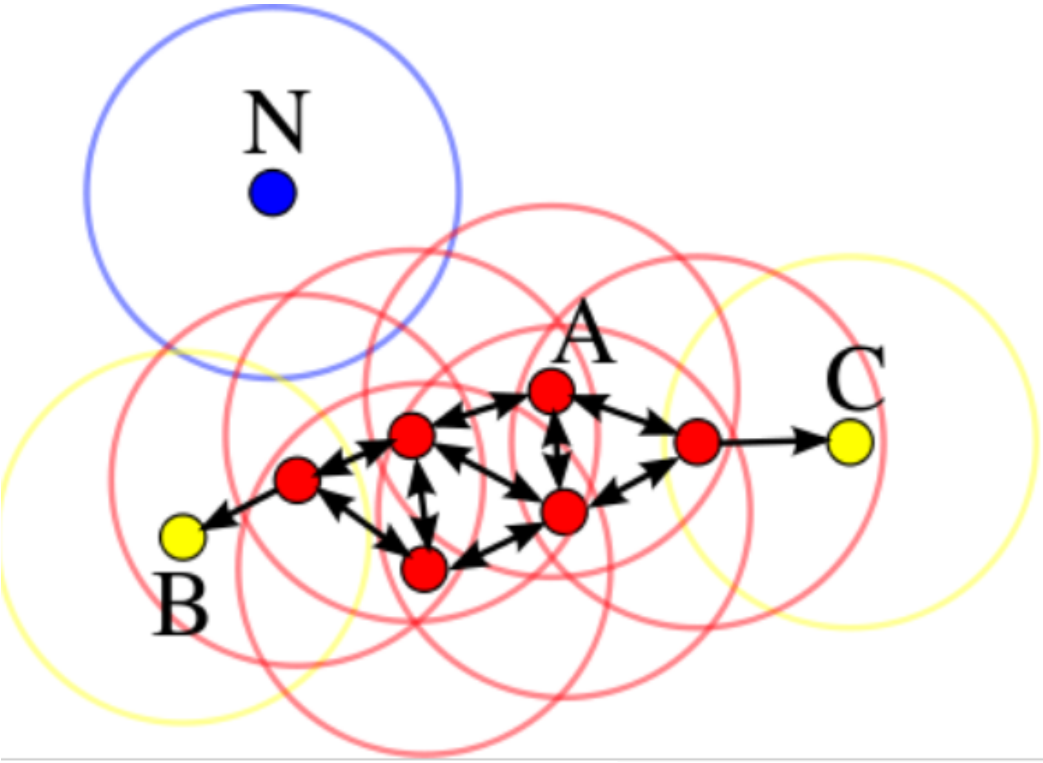
- Can work for relatively large dataset
  - Time complexity O(d n log n) per iteration (assumptions: n >> k, dimension d is small) http://www.cs.cmu.edu/~./dpelleg/download/kmeans.ps

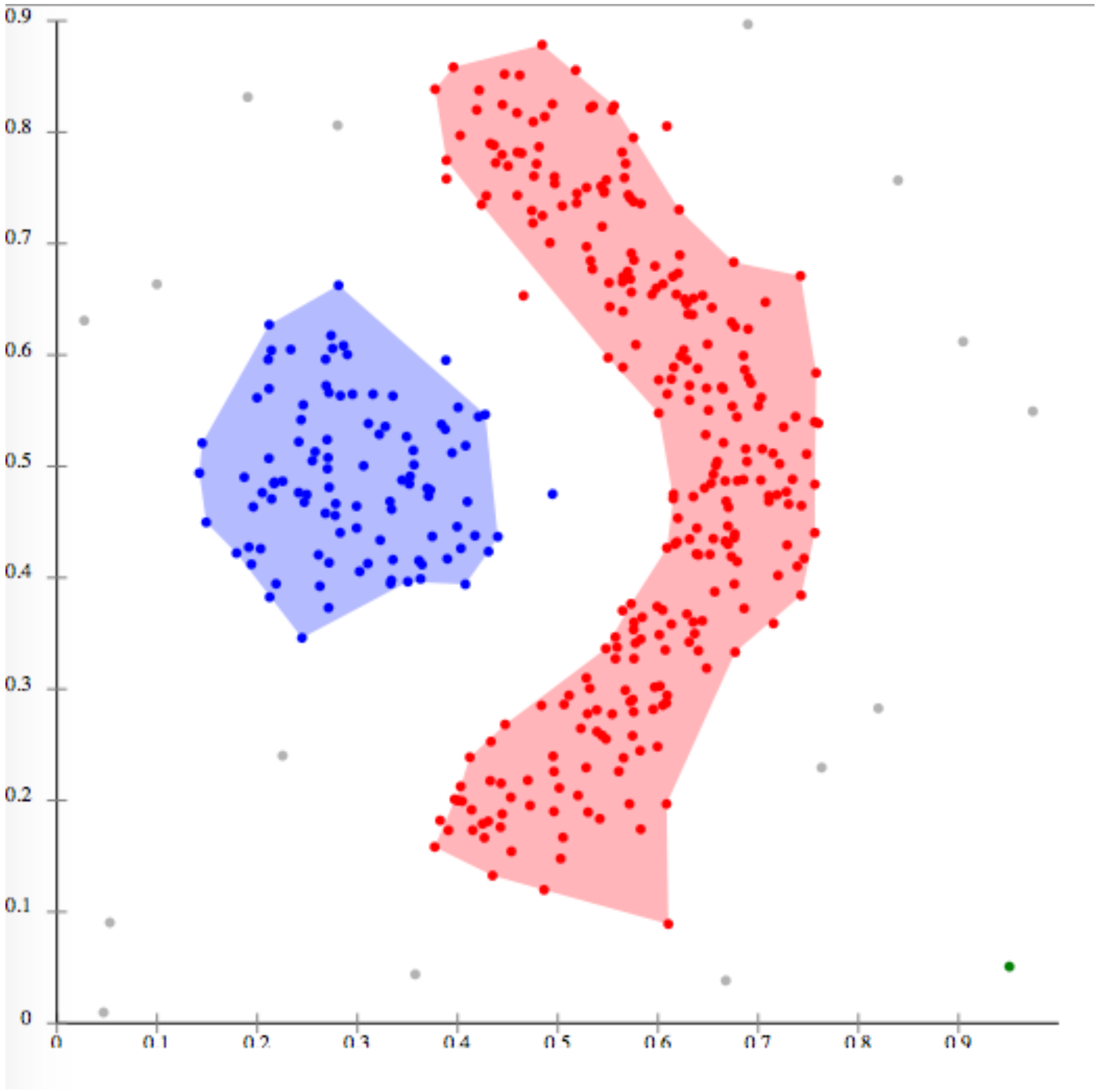
DBSCAN

"Density-based spatial clustering with noise"

https://en.wikipedia.org/wiki/DBSCAN

Received "test-of-time award" at KDD'14 — an extremely prestigious award.





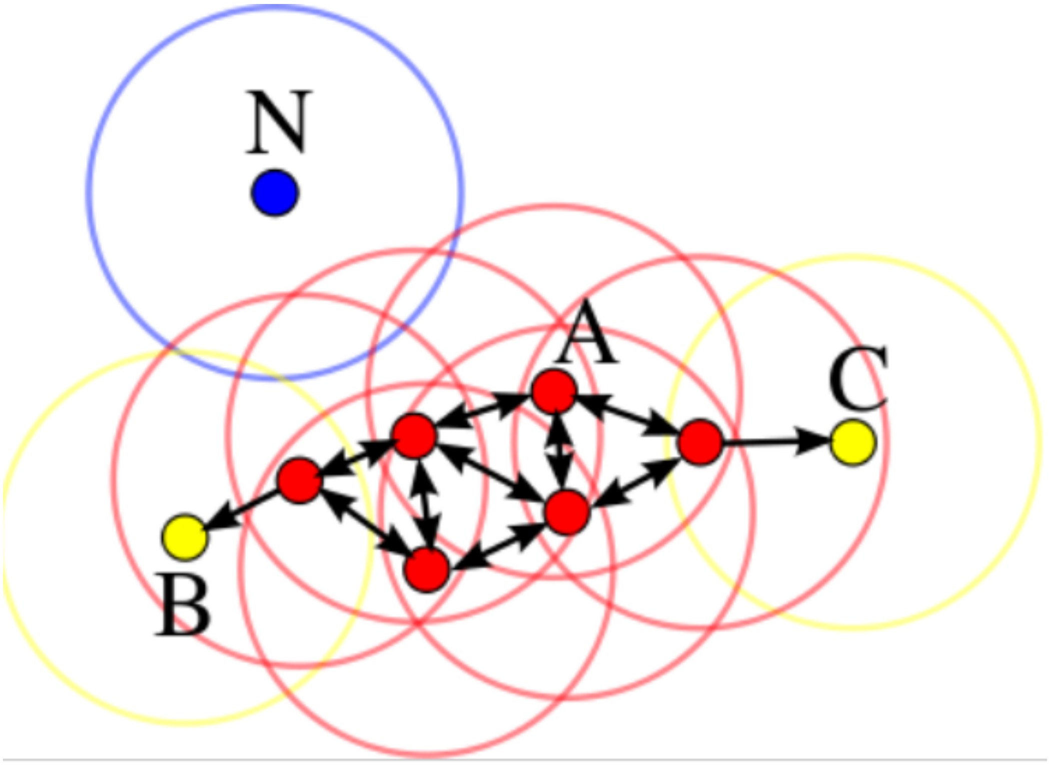
Only need two parameters:

1. "radius" epsilon
2. minimum number of points (e.g., 4) required to form a dense region

Yellow "border points" are **density-reachable** from red "core points", but not vice-versa.

Interactive DBSCAN Demo

https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/



Only need two parameters:
1. “radius” epsilon
2. minimum number of points (e.g., 4) required to form a dense region

Yellow “border points” are **density-reachable** from red “core points”, but not vice-versa.

You can use DBSCAN now.

http://scikit-learn.org/stable/auto_examples/cluster/plot_dbscan.html

To learn more…

- A great way is to try it out on **real data** (e.g., for your research), not just on toy datasets
- Courses at Georgia Tech
  - **CSE6740/ISYE6740/CS6741 Machine Learning** (course title may say "computational data analytics")
  - **CSE6242 Data & Visual Analytics** (Polo's class; more applied; ML is only part of the course)
  - Machine learning for trading, big data for healthcare, computer vision, natural language processing, deep learning, and many more!