Lecture 11: Exploring Data Through Preprocessing and Unsupervised ML Part 1

Feb 20, 2023

CIS 4190/5190

Spring 2023

Administrivia

- Project proposal-related deadlines coming up!
  - Team info submission tonight at 8 p.m.
  - Project proposal due Mar 1 at 8 p.m.

- HW3 due Wed at 8 p.m.

Our Machine Learning Toolkit in Practice

So far, we've assumed the data is ready for us to apply ML techniques

1. Available as an "**X** matrix": instances as rows, features as columns
2. We know the classes we want to predict
3. We are given training labels for the classes

But this isn't always the case!

- Data may need to be wrangled and integrated
- We may need to understand our data and tasks
- We may need to understand what the data "tells us"

For more depth, see CIS 5450 – but here we'll briefly discuss some of the major techniques and ideas

https://www.base-4.com/open-apartments-faster-with-modular/

https://dreamsmeaning.site/house-under-construction/

ML “Workflow”

- Understanding the domain, framing the ML problem — Project
- Data acquisition + curation (sourcing, scraping, collection, labeling)
- Data analysis / visualization

This module

- ML Design (hypothesis class, loss function, optimizer, hyperparameters, features)
- Train model
- Validate / Evaluate

Main focus of this class

- Deploy (and generate new data)
- Monitor performance on new data

End of semester

Need to Do Work to Prepare Data for ML

Data is rarely "clean": Real data is messy, fragmented

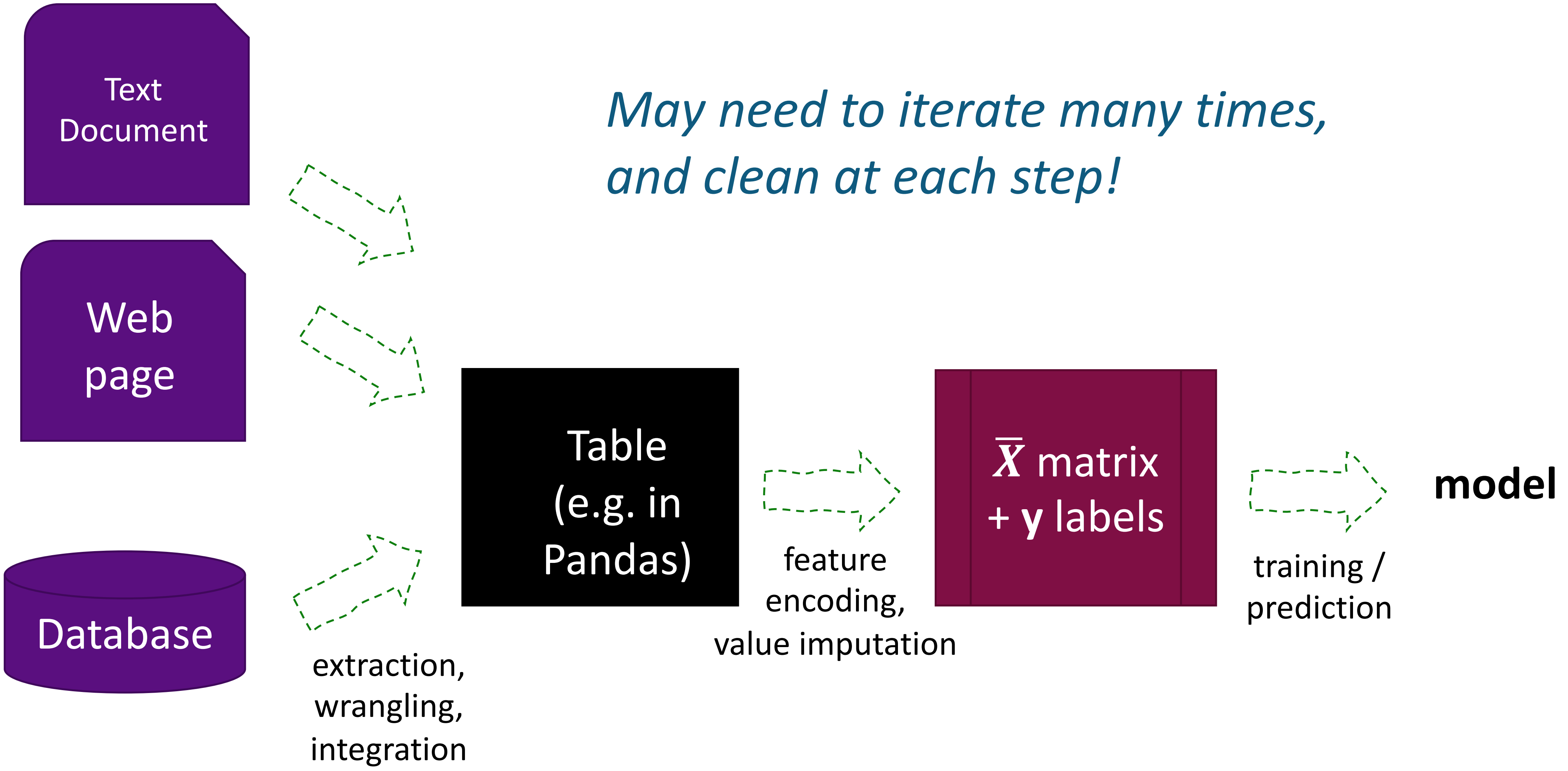
$80\% +$ of human time is spent on data wrangling, according to practitioners

- Depends on who you ask, the application, the data source, etc...
- And "80%+" can be an underestimate!

Having good data with appropriate features is absolutely critical to success

Acquiring Data

Data through the ML Pipeline



Challenges of Acquiring Relevant Data

https://www.drivethrurpg.com/product/190343/Fiefdom-Manors-Ruling-manorial-fiefs

In many data science/ML applications: identifying and *getting access to data sources* is a challenge itself

- Lack of infrastructure for data sharing
- Lack of clarity on what is needed / need for data discovery
- Proprietary data
- Privacy constraints on access: HIPAA, FERPA, GDPR, …

Once you have permissions, may need to *wrangle* the data into forms ready for applying ML algorithms, such as Pandas tables

- From popular data sharing formats: CSV, JSON, XML
- From HTML tables
- From structured files: DICOM, HDF5, Excel, MatLab, …
- Text → information extraction and NLP
- Database connections: SQLAlchemy etc

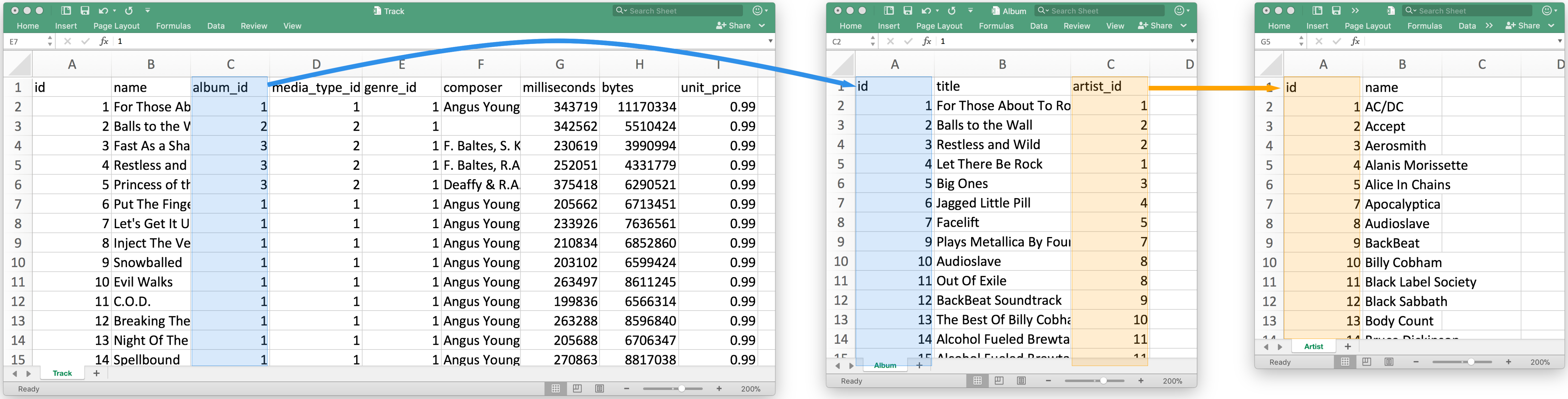
Integrating Multiple Data Sources

Merging Data: Pandas or SQL merge / join

Data may be split across different files

Requires doing a join based on a key to combine data into one table

tracks albums artists



tracks

	A	B	C	D	E	F	G	H	I
1	id	name	album_id	media_type_id	genre_id	composer	milliseconds	bytes	unit_price
2	1	For Those Ab	1	1	1	Angus Young	343719	11170334	0.99
3	2	Balls to the V	2	2	1		342562	5510424	0.99
4	3	Fast As a Sha	3	2	1	F. Baltes, S. K	230619	3990994	0.99
5	4	Restless and	3	2	1	F. Baltes, R.A	252051	4331779	0.99
6	5	Princess of th	3	2	1	Deaffy & R.A.	375418	6290521	0.99
7	6	Put The Finge	1	1	1	Angus Young	205662	6713451	0.99
8	7	Let's Get It U	1	1	1	Angus Young	233926	7636561	0.99
9	8	Inject The Ve	1	1	1	Angus Young	210834	6852860	0.99
10	9	Snowballed	1	1	1	Angus Young	203102	6599424	0.99
11	10	Evil Walks	1	1	1	Angus Young	263497	8611245	0.99
12	11	C.O.D.	1	1	1	Angus Young	199836	6566314	0.99
13	12	Breaking The	1	1	1	Angus Young	263288	8596840	0.99
14	13	Night Of The	1	1	1	Angus Young	205688	6706347	0.99
15	14	Spellbound	1	1	1	Angus Young	270863	8817038	0.99

albums

	A	B	C
1	id	title	artist_id
2	1	For Those About To Ro	1
3	2	Balls to the Wall	2
4	3	Restless and Wild	2
5	4	Let There Be Rock	1
6	5	Big Ones	3
7	6	Jagged Little Pill	4
8	7	Facelift	5
9	9	Plays Metallica By Fou	7
10	10	Audioslave	8
11	11	Out Of Exile	8
12	12	BackBeat Soundtrack	9
13	13	The Best Of Billy Cobha	10
14	14	Alcohol Fueled Brewta	11

artists

	A	B
1	id	name
2	1	AC/DC
3	2	Accept
4	3	Aerosmith
5	4	Alanis Morissette
6	5	Alice In Chains
7	7	Apocalyptica
8	8	Audioslave
9	9	BackBeat
10	10	Billy Cobham
11	11	Black Label Society
12	12	Black Sabbath
13	13	Body Count

Image: https://dataschool.com/learn-sql/joins/

Integration May be Hard

Encoding issues

- Inconsistent data formats or terminology
- Key aspects mentioned in cell comments or auxiliary files

Merged table may be too large for memory

- Incrementally load and join data, using SGD or mini-batches
- Use online learning techniques

Record Linkage problem (next)

The Record Linkage Problem

Ins ID	Name
203342	J Smith
123452	Mao Y

Student ID	Name
3432432	Jon Smithee
9734783	Jane Smyth
8273737	Ying Mao

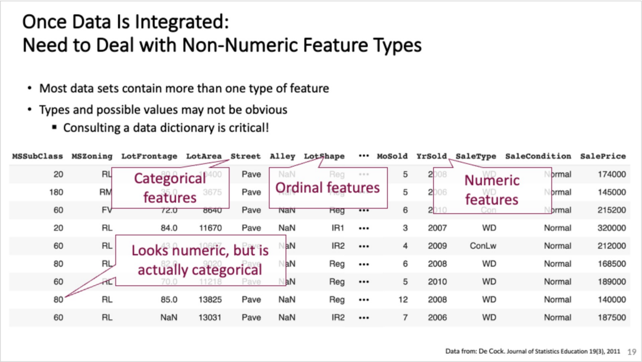
Huge literature. Some popular ideas:

- String similarity above a threshold
  - String edit distance ("J Smith" → "Jon Smithee" with 4 edits)
  - String overlap (n-grams)
- May want to tokenize and compare tokens, not just strings
  - Or consider how they sound (e.g. "soundex"), common mis-substitutions, …
- Often combine similarities of multiple fields (e.g., addresses, employer)

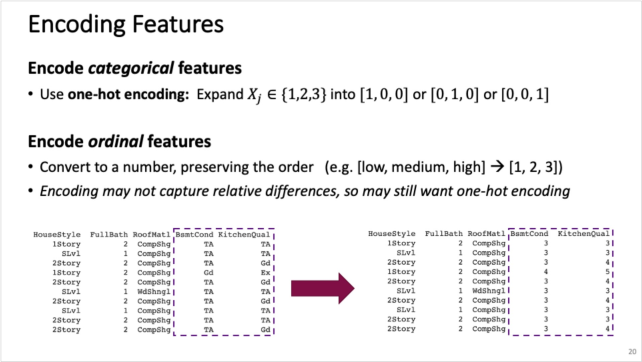
Exploring and Fixing Data Quality Issues

Recap: Some Ideas We've Encountered Before

Resolving feature datatypes etc. with a data dictionary

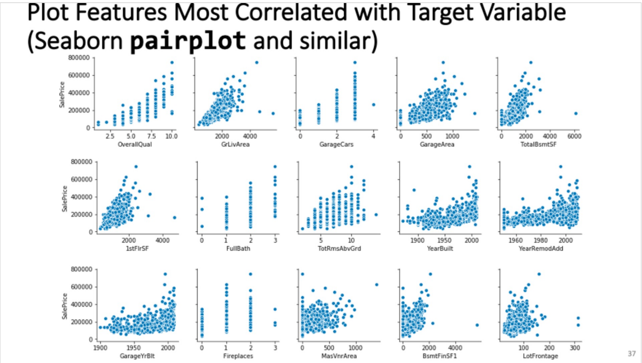


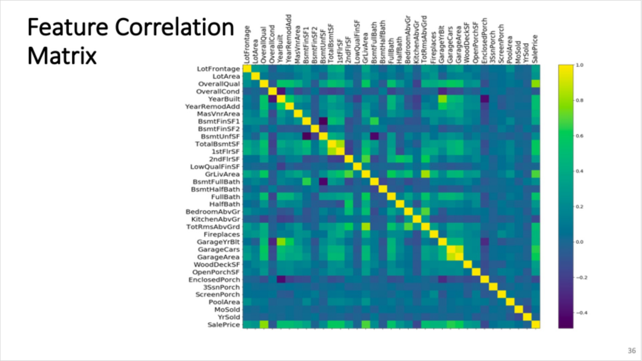
Convert categorical / ordinal data to numerical



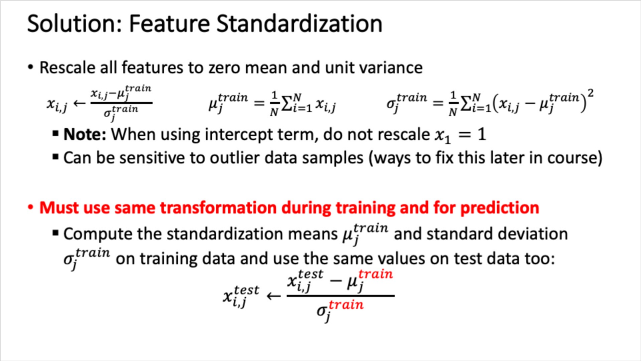
Recap: Some Ideas We've Encountered Before

Feature-feature and feature-label correlations to inform feature selection





Scaling features to prepare for non-scaling-invariant ML approaches



Solution: Feature Standardization

- Rescale all features to zero mean and unit variance

$$x_{i,j} \leftarrow \frac{x_{i,j}-\mu_j^{train}}{\sigma_j^{train}} \qquad \mu_j^{train} = \frac{1}{N}\sum_{i=1}^{N} x_{i,j} \qquad \sigma_j^{train} = \frac{1}{N}\sum_{i=1}^{N}\left(x_{i,j}-\mu_j^{train}\right)^2$$

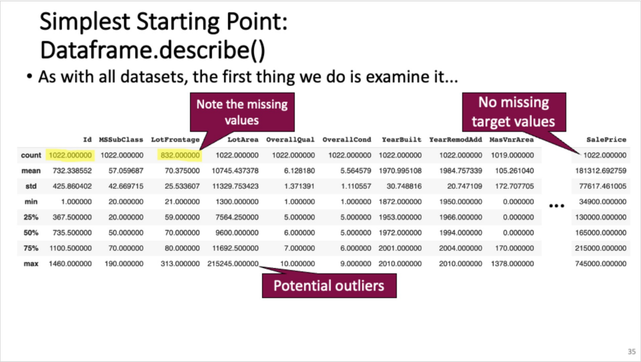
  - **Note:** When using intercept term, do not rescale $x_1 = 1$
  - Can be sensitive to outlier data samples (ways to fix this later in course)

- **Must use same transformation during training and for prediction**
  - Compute the standardization means μ_j^{train} and standard deviation σ_j^{train} on training data and use the same values on test data too:

$$x_{i,j}^{test} \leftarrow \frac{x_{i,j}^{test}-\mu_j^{train}}{\sigma_j^{train}}$$

Recap: Some Ideas We've Encountered Before

Find missing values and outliers



Simplest Starting Point: Dataframe.describe()

- As with all datasets, the first thing we do is examine it…

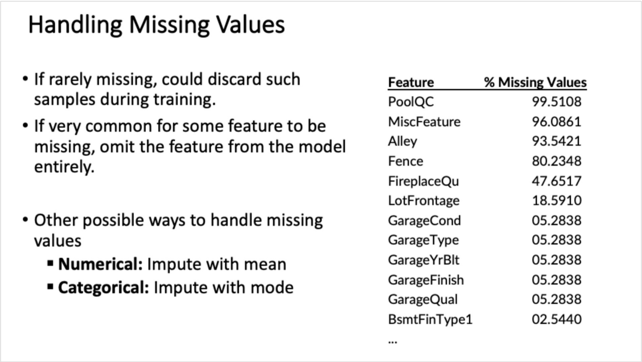
Note the missing values

No missing target values

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	…	SalePrice
count	1022.000000	1022.000000	832.000000	1022.000000	1022.000000	1022.000000	1022.000000	1022.000000	1019.000000		1022.000000
mean	732.338552	57.059687	70.375000	10745.437378	6.128180	5.564579	1970.995108	1984.757339	105.261040		181312.692759
std	425.860402	42.669715	25.533607	11329.753423	1.371391	1.110557	30.748816	20.747109	172.707705		77617.461005
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000		34900.000000
25%	367.500000	20.000000	59.000000	7564.250000	5.000000	5.000000	1953.000000	1966.000000	0.000000		130000.000000
50%	735.500000	50.000000	70.000000	9600.000000	6.000000	5.000000	1972.000000	1994.000000	0.000000		165000.000000
75%	1100.500000	70.000000	80.000000	11692.500000	7.000000	6.000000	2001.000000	2004.000000	170.000000		215000.000000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1378.000000		745000.000000

Potential outliers

Handling missing values



Handling Missing Values

- If rarely missing, could discard such samples during training.
- If very common for some feature to be missing, omit the feature from the model entirely.
- Other possible ways to handle missing values
  - **Numerical:** Impute with mean
  - **Categorical:** Impute with mode

Feature	% Missing Values
PoolQC	99.5108
MiscFeature	96.0861
Alley	93.5421
Fence	80.2348
FireplaceQu	47.6517
LotFrontage	18.5910
GarageCond	05.2838
GarageType	05.2838
GarageYrBlt	05.2838
GarageFinish	05.2838
GarageQual	05.2838
BsmtFinType1	02.5440
...	

continued

Missing Feature Values

- **Delete features** with mostly missing values
- **Delete instances** with (many) missing features, if rare
- **Impute** via mean (numeric) or mode (categorical or ordinal)

A couple more sophisticated solutions:

- Train an ML model to **predict the missing values** (i.e., a kind of model "stacking")
- Flag missing values using **binary variables**

Missing Values

Data might not be “missing at random”. It might be meaningful that instances have missing features!
e.g., history might be missing from an unconscious trauma patient

ID	Last_Visit
1234	2018-03-05
4567	0
8910	2019-12-12

Rather than removing the case where Last_Visit is unknown – flag it with a separate feature!

We will learn a weight for the feature if it’s there, and also a different weight if it isn’t!

Other Data Quality Issues: Incorrect Feature Values

- Typos: e.g., color = {“bleu”, “green”, “gren”, “red”}
- Inconsistent spelling (e.g., “color”, “colour”)
- Inconsistent abbreviations (e.g., “Oak St.”, “Oak Street”)
- Garbage: e.g., color = “w┐r--śïį”

Often can be identified by comparing against a dictionary (~ spell-check) to identify data-entry or encoding issues. Sometimes easy to fix once identified.

Data Outliers

Errors

- Human error in data collection or data entry
- Measurement/instrumentation errors
- Experimental errors
- Data merge errors
  - e.g., merging datasets with different scales
- Data preprocessing errors

Natural

- Could also just be real outliers in the data – not mistakes!

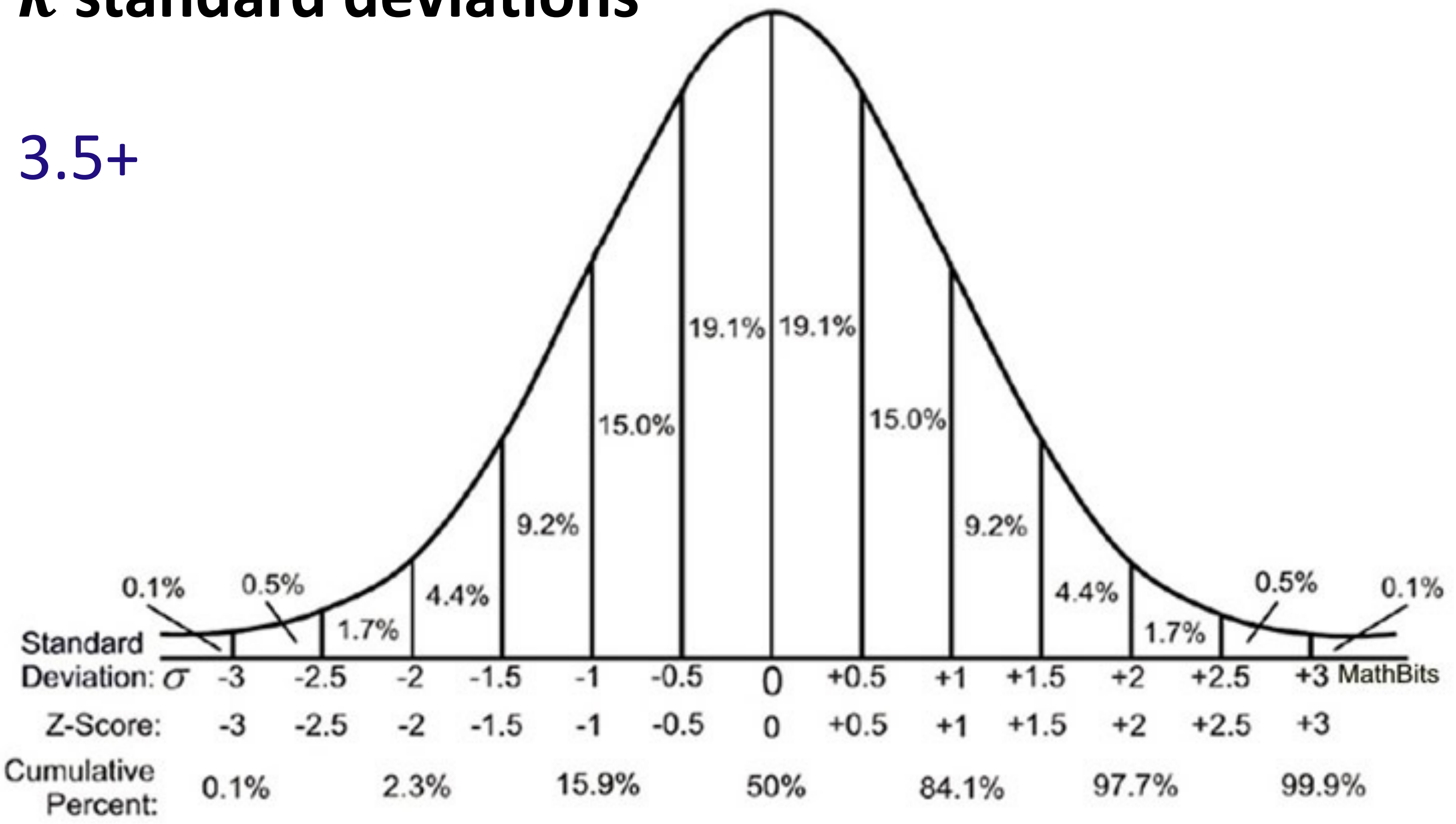
Outlier Detection: Z-score

- Assume feature values are **Gaussian-distributed**
- Discard points more than k **standard deviations** away from the mean

  Good values for k: 2.5, 3, 3.5+

Cautions:

- Mostly for low-*d* feature spaces on reasonably small-to-medium data sets
- Incorrect if parametric assumption doesn't hold



https://mathbitsnotebook.com/Algebra2/Statistics/STzScores.html

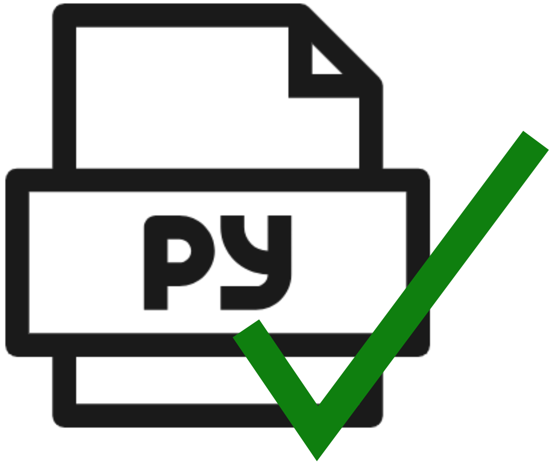
Best Practice: Script All Data Exploration & Preprocessing!

Don't manually edit via a spreadsheet program

- No history of changes
- Very easy to introduce mistakes
- Hard to correct earlier decisions

Instead, write a script that loads the raw data and does all preprocessing

- Documents all steps
- Incremental debugging
- Easy to make changes to earlier steps
- Repeatable



Optional readings: Data Wrangling

- https://hbr.org/2018/08/what-data-scientists-really-do-according-to-35-data-scientists
- https://sites.google.com/seas.upenn.edu/cis545
- https://dcl-wrangle.stanford.edu/

Unsupervised Machine Learning

Understanding the *Structure* of a Dataset

- Our understanding of the distribution of the data thus far has been quite simplistic, restricted to univariate and bivariate distributions.

- Data can often have much more complex structure that is useful to understand.

- E.g., suppose the data naturally falls into different groupings – perhaps even suggesting which *classes* we would like to learn?

➢ This motivates a study of additional techniques for extracting structure present in the data

Types of Learning

Supervised learning

- Given: training data + desired outputs (labels)

Unsupervised learning

- Given: training data (without labels)

Semi-supervised learning

- Given: training data + some labels

Reinforcement learning

- Given: observations and occasional rewards as the agent performs sequential actions in an environment

“Here’s some data, could you do something with it?”

One common use of unsupervised learning is to identify use cases for data:

- Visualize the data, find clusters
  e.g. “based on our polling data, there are three main voting blocs, based on age, race, education level, income, political beliefs, and home-ownership. Features like marital status and # children are irrelevant.”
- Identify interesting supervised learning problems within your dataset
  e.g. “do our company’s profits y_i actually correlate with the weather $\boldsymbol{x}_i$?”
- Generate new data
  e.g. “given all of Bach’s work, I could generate new music that would sound like Bach.”
- Identify important features in the dataset
  e.g. “Most of the variation between our customers is explained by their age, location, and education level.”

Unsupervised ML in the ML Workflow

- Framing an ML problem
- Data curation (sourcing, scraping, collection, labeling)
- Data analysis / visualization
- ML Design (hypothesis class, loss function, optimizer, hyperparameters, features)
- Train model
- Validate / Evaluate
- Deploy (and generate new data)
- Monitor performance on new data

A Popular Clustering Algorithm: K-Means Clustering

In unsupervised learning, you see,

There's a technique called clustering, that be,

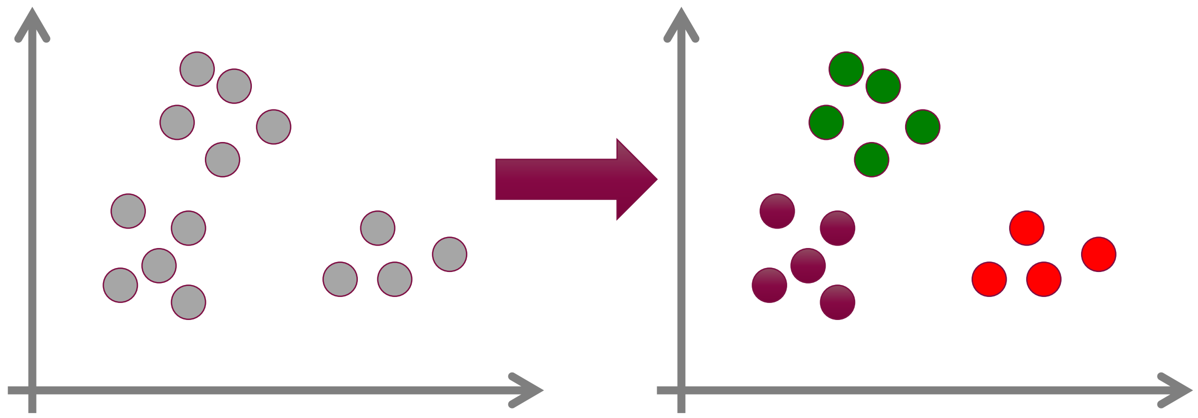
It groups data points alike,

With algorithms that strike,

Patterns hidden from you and me!

Clustering

What natural groupings exist in this data?



The Clustering Setting

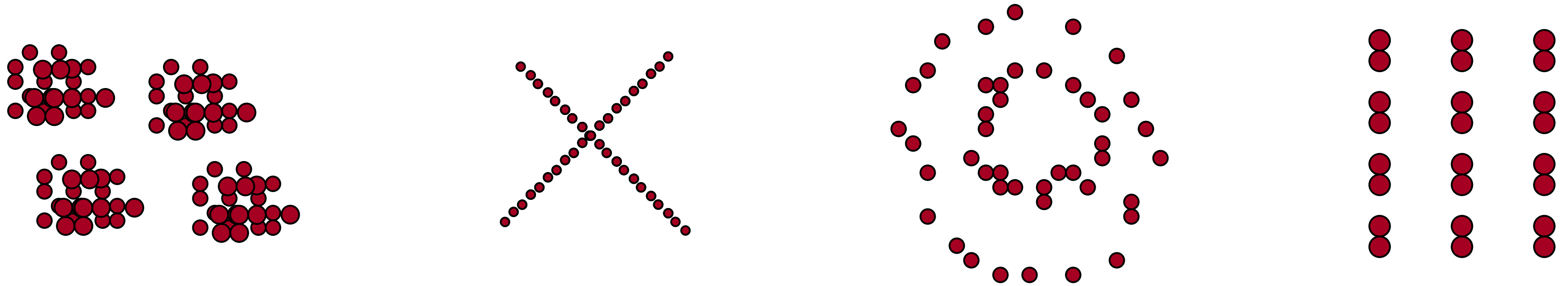
Task:

Input: $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$

Want to discover a mapping $f(\boldsymbol{x}_i) \in \{1, 2, 3, \dots, K\}$ that discovers natural groupings in the data.

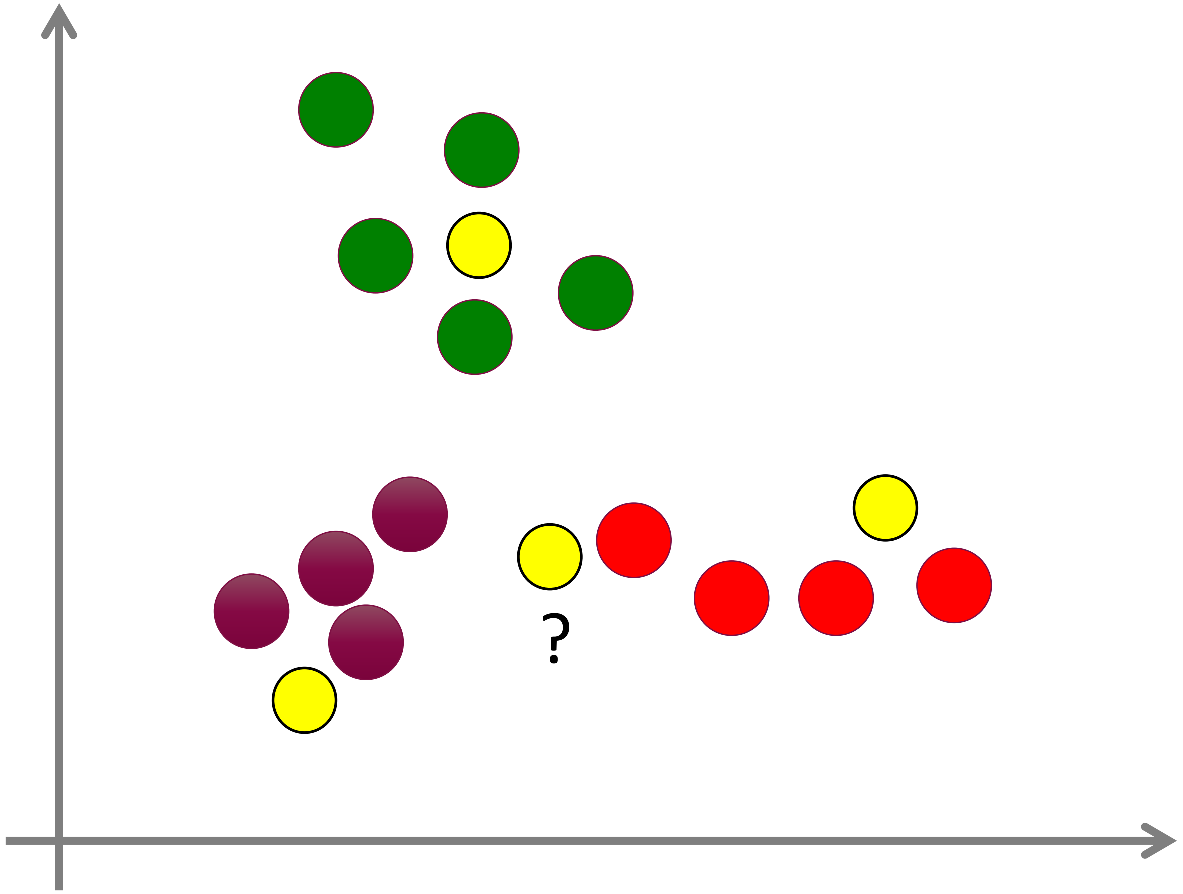
Performance Metric / Objective Function: What is a good mapping $f(.)$?

Somewhat loosely defined, and different clustering algorithms differ in their definition of a good clustering $f(.)$



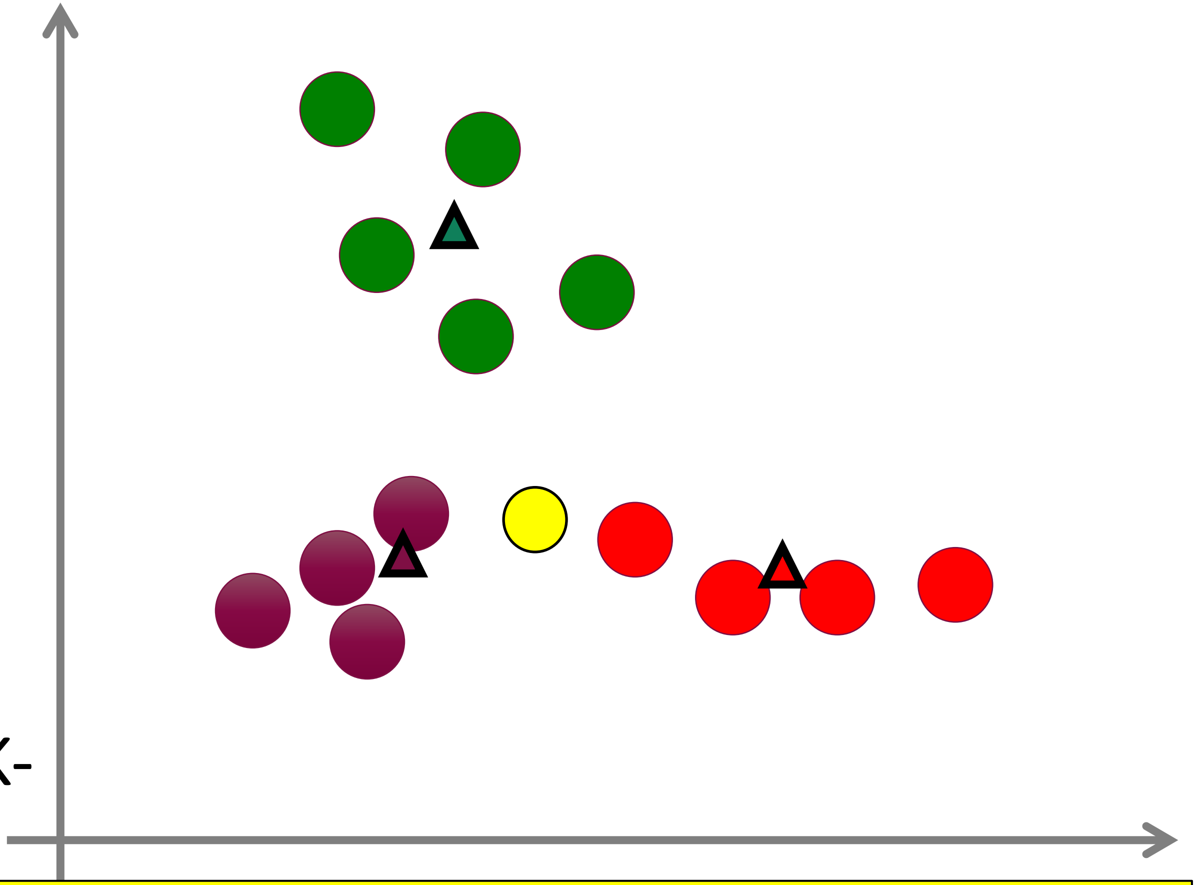
Figures: Dan Roth

Guess The Cluster Assignment



Cluster Assignment in K-Means

- Define a centroid μ_k for each cluster k
- For any new sample, assign the cluster whose centroid/mean is closest!

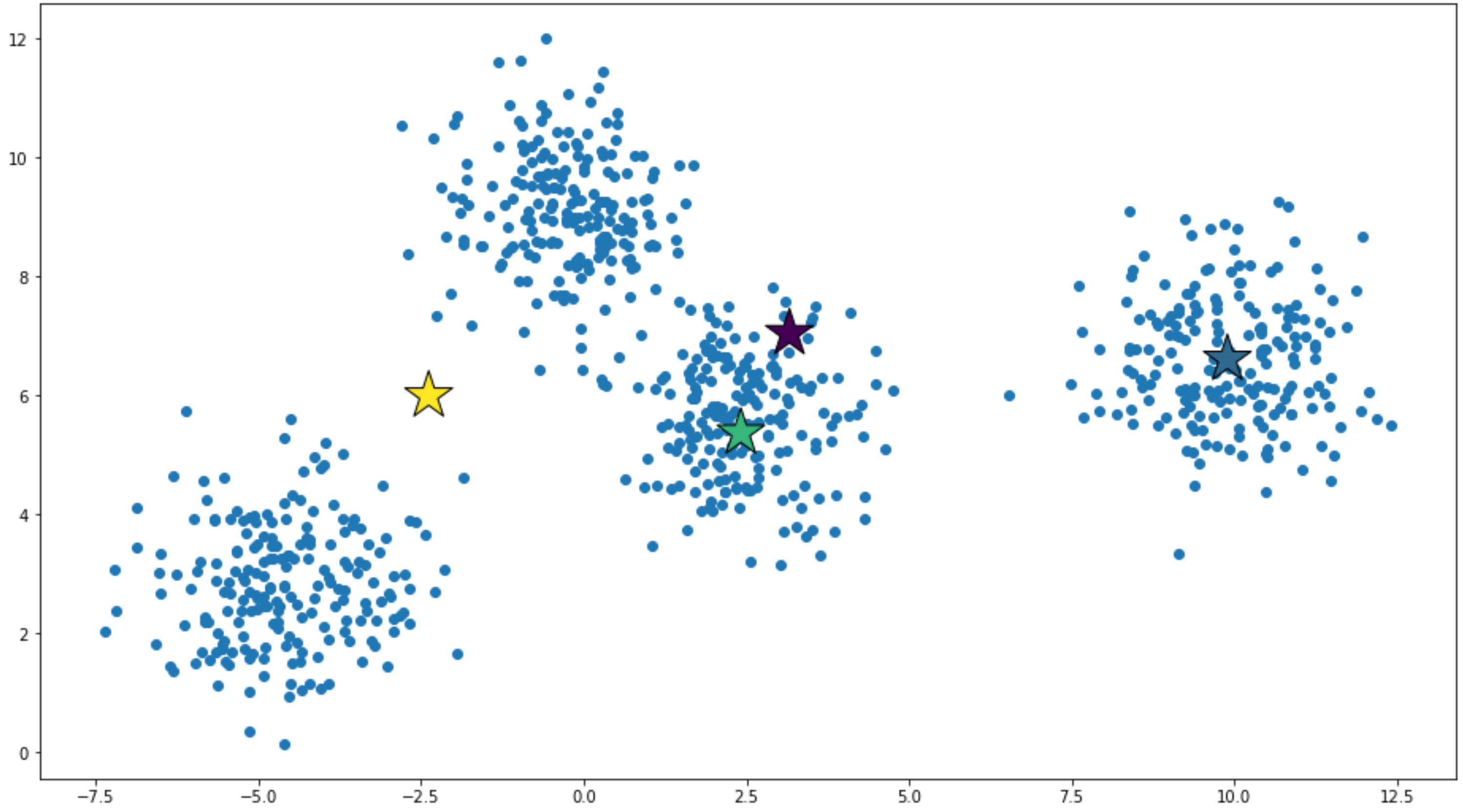


Equivalent to 1-nearest neighbor classification over the cluster means.

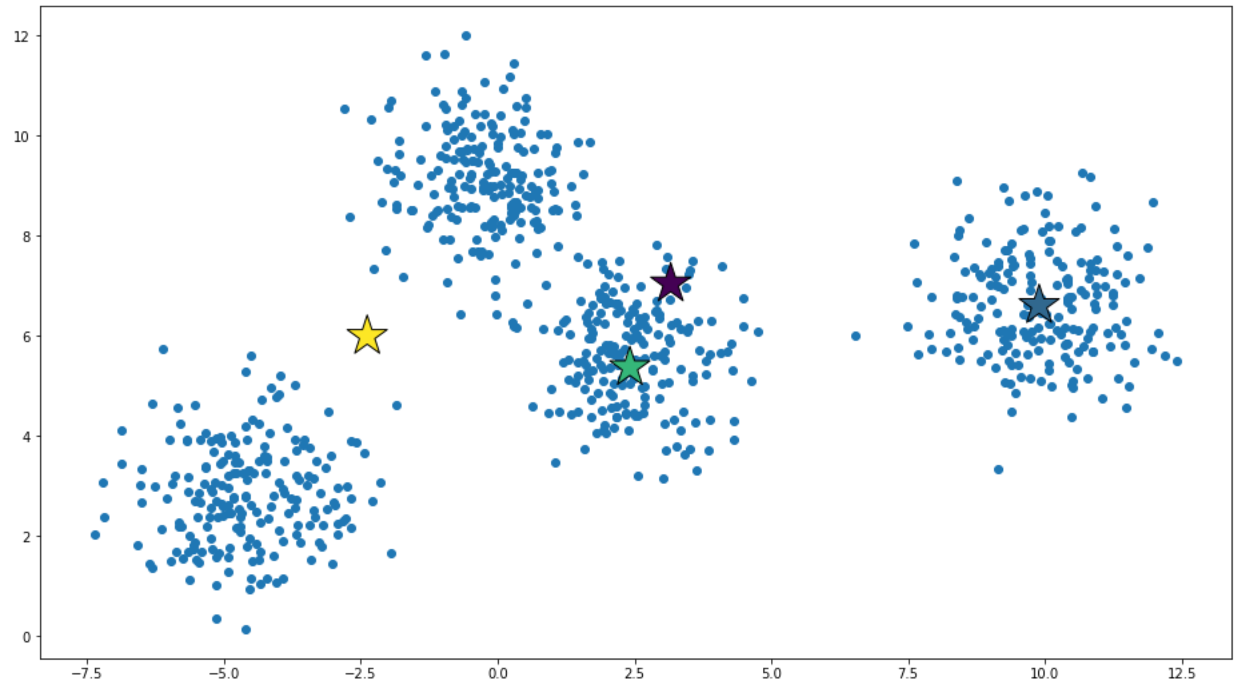
Note: Certainly not the only answer we could have come up with, this is just the K-Means way to cluster!

Can we expand this into a full, consistent clustering algorithm?

Clustering Data



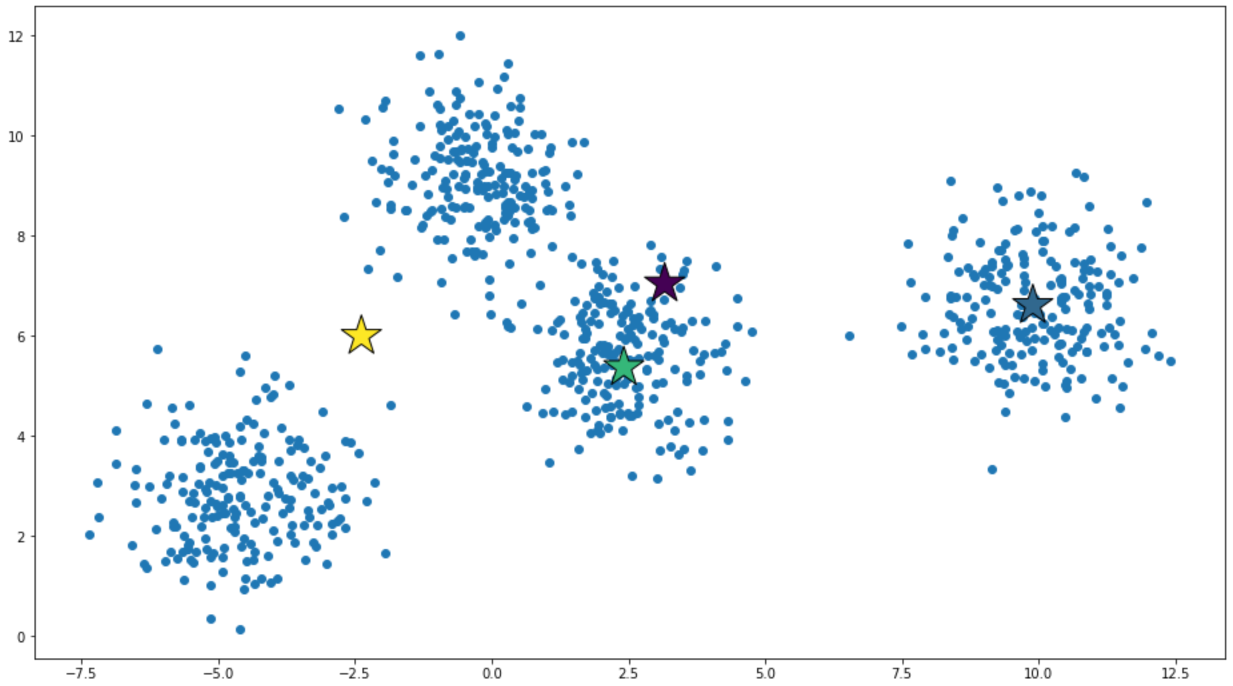
Clustering Data



K-Means Clustering

K-Means (K, X)

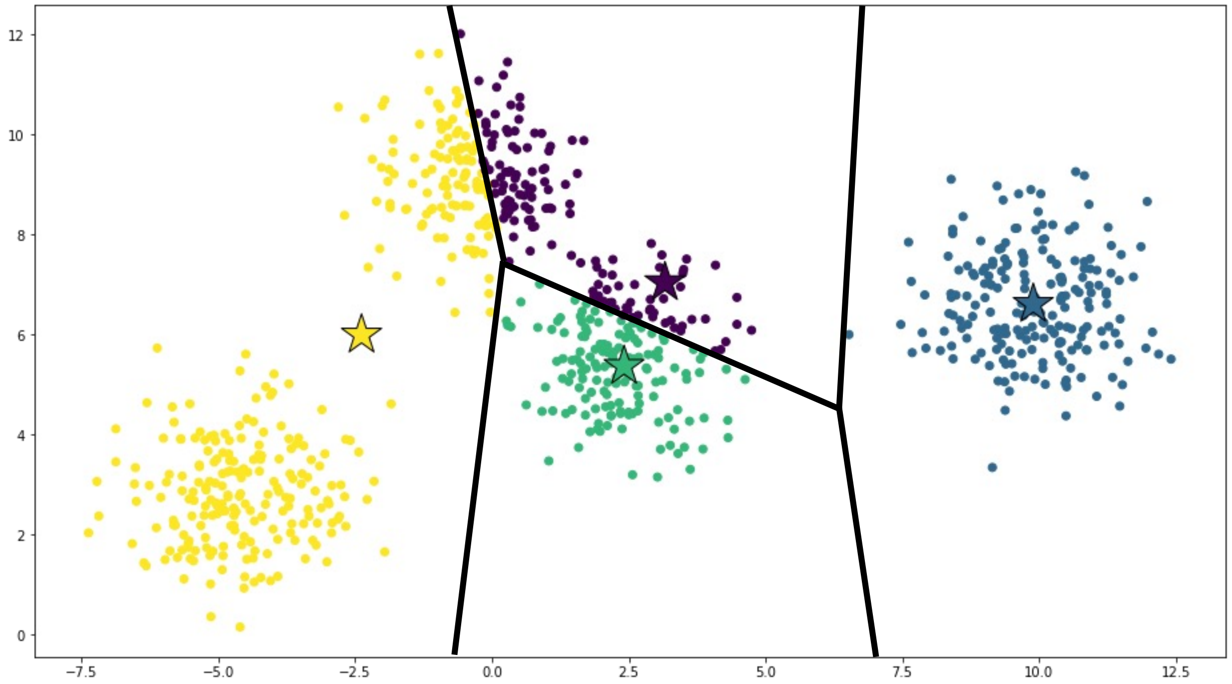
- Randomly choose K cluster means
- Loop until convergence, do:
  - Assign each point to the cluster of the closest centroid
  - Re-estimate the cluster centroids based on the data assigned to each cluster



K-Means Clustering

K-Means (K, X)

- Randomly choose K cluster center locations (centroids)
- Loop until convergence, do:
    - Assign each point to the cluster of the closest centroid
    - Re-estimate the cluster centroids based on the data assigned to each cluster

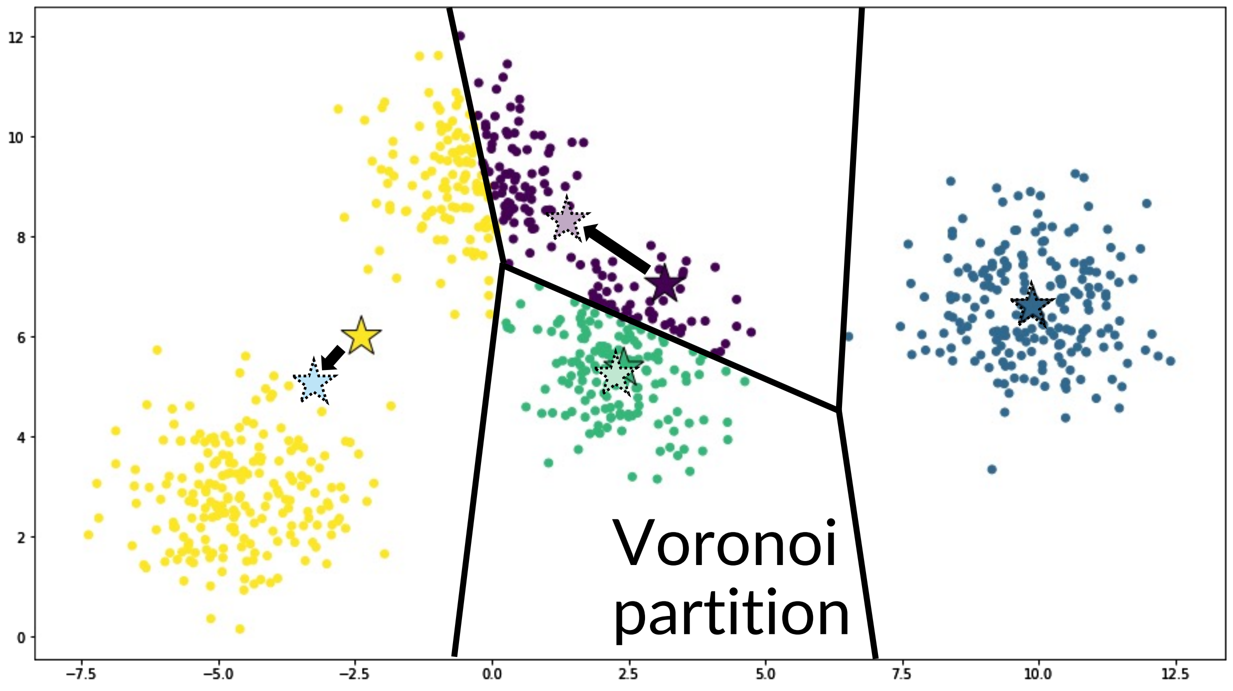


Voronoi partition

K-Means Clustering

K-Means (K, X)

- Randomly choose K cluster center locations (centroids)
- Loop until convergence, do:
  - Assign each point to the cluster of the closest centroid
  - Re-estimate the cluster centroids based on the data assigned to each cluster



K-Means Clustering

K-Means (K, X)

- Randomly choose K cluster center locations (centroids)
- Loop until convergence, do:
  - Assign each point to the cluster of the closest centroid
  - Re-estimate the cluster centroids based on the data assigned to each cluster

Optimizer: "Alternating Minimization"

K-means finds a local optimum of the following objective function:

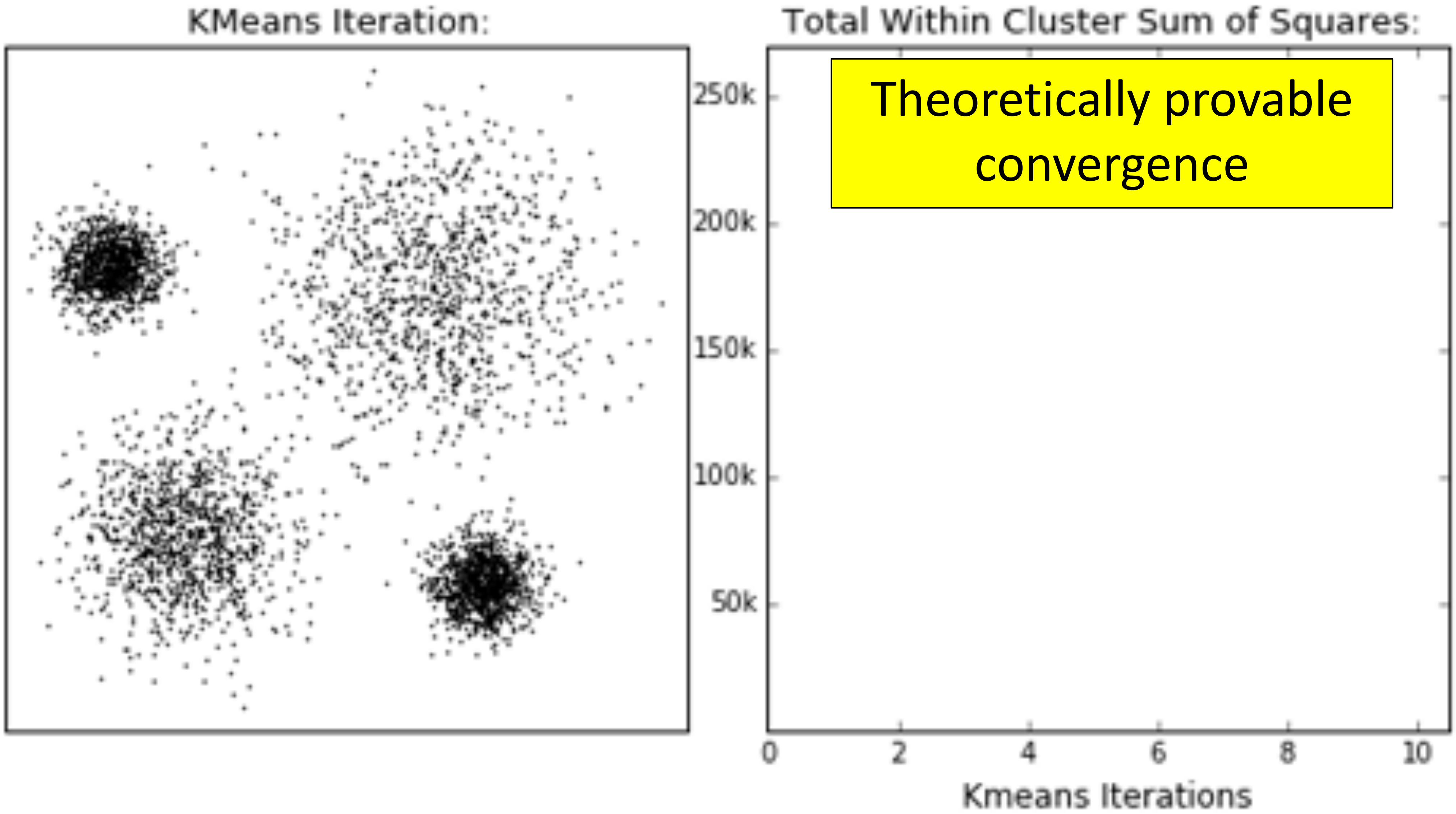
"Sum of squared distances" loss function

$$\arg\min_{\boldsymbol{S}} \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in S_k} \boxed{\|\boldsymbol{x} - \boldsymbol{\mu}_K\|_2^2}$$

where $\boldsymbol{S} = \{S_1, \ldots, S_K\}$ are sets corresponding to disjoint clusters, and the clusters together include all samples.

K-Means Clustering Convergence

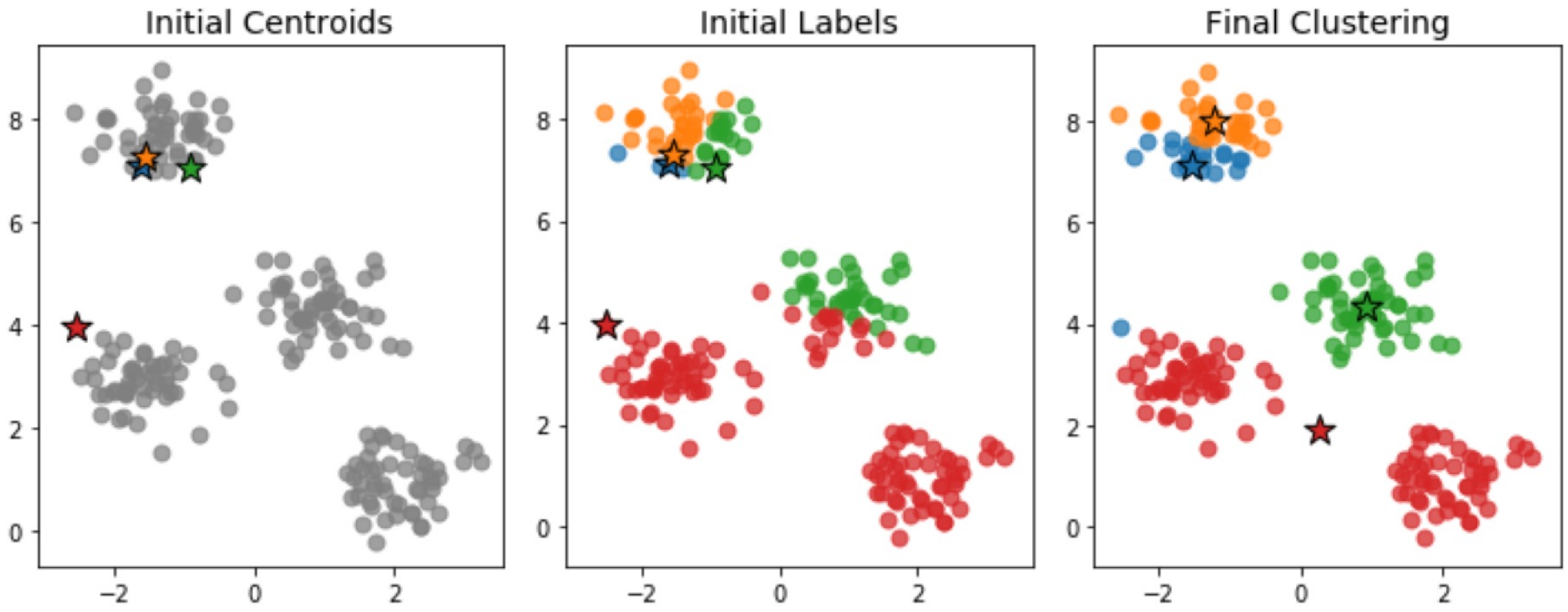
$$\sum_{k=1}^{K} \sum_{x \in S_k} \|\boldsymbol{x} - \boldsymbol{\mu}_K\|_2^2$$



https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/

Initializing K-Means

K-Means is Too Sensitive to Initialization



http://www.salientiastuff.com/k-means-clustering-part-2.html

Different local minima each time (sometimes bad) based on the initialization.

Can take long to converge with bad initializations.

K-Means is Too Sensitive to Initialization

Alternative strategies:

1. Do many runs of K-Means, each with different initial centroids, and pick the best
2. Pick initial centroids using a better method than random choice

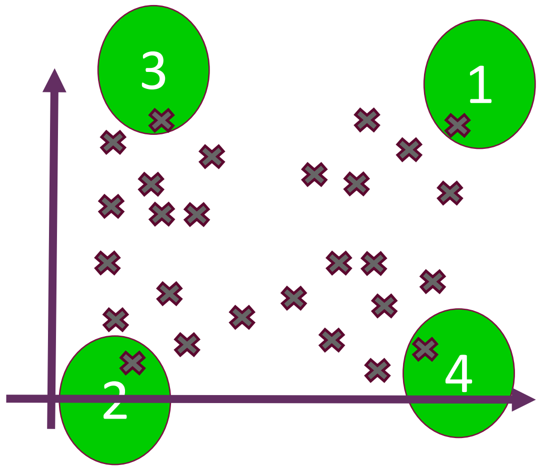
K-means+ + initialization

- Choose a data point uniformly at random as the first centroid
- Loop for $2{:}K$, do:
  - Let $D(\boldsymbol{x})$ be the distance from each point $\boldsymbol{x}$ to the closest centroid
  - For sampling the next candidate centroid from among the remaining points, assign probability weights $w(\boldsymbol{x}) \propto D(\boldsymbol{x})^2$ to each point, i.e., higher chance to pick points that are far from previous centroids.

K-means++ Illustrated

Place the initial centroids **far away from one another**:

1. Initialize an empty set M (for storing selected centroids); **Randomly select** the first centroid from the input sample and assign it to M
2. For each x_i that is not in M, find the distance $D(x_i)$ to the closest centroid in M
3. Choose one new data point at random as a new centroid **using probability distribution ~ $D(x)^2$**
4. Repeat (2) and (3) until K centroids have been chosen



Then do "classic" k-means

How Many Clusters K?

How can we evaluate how good our clustering is?

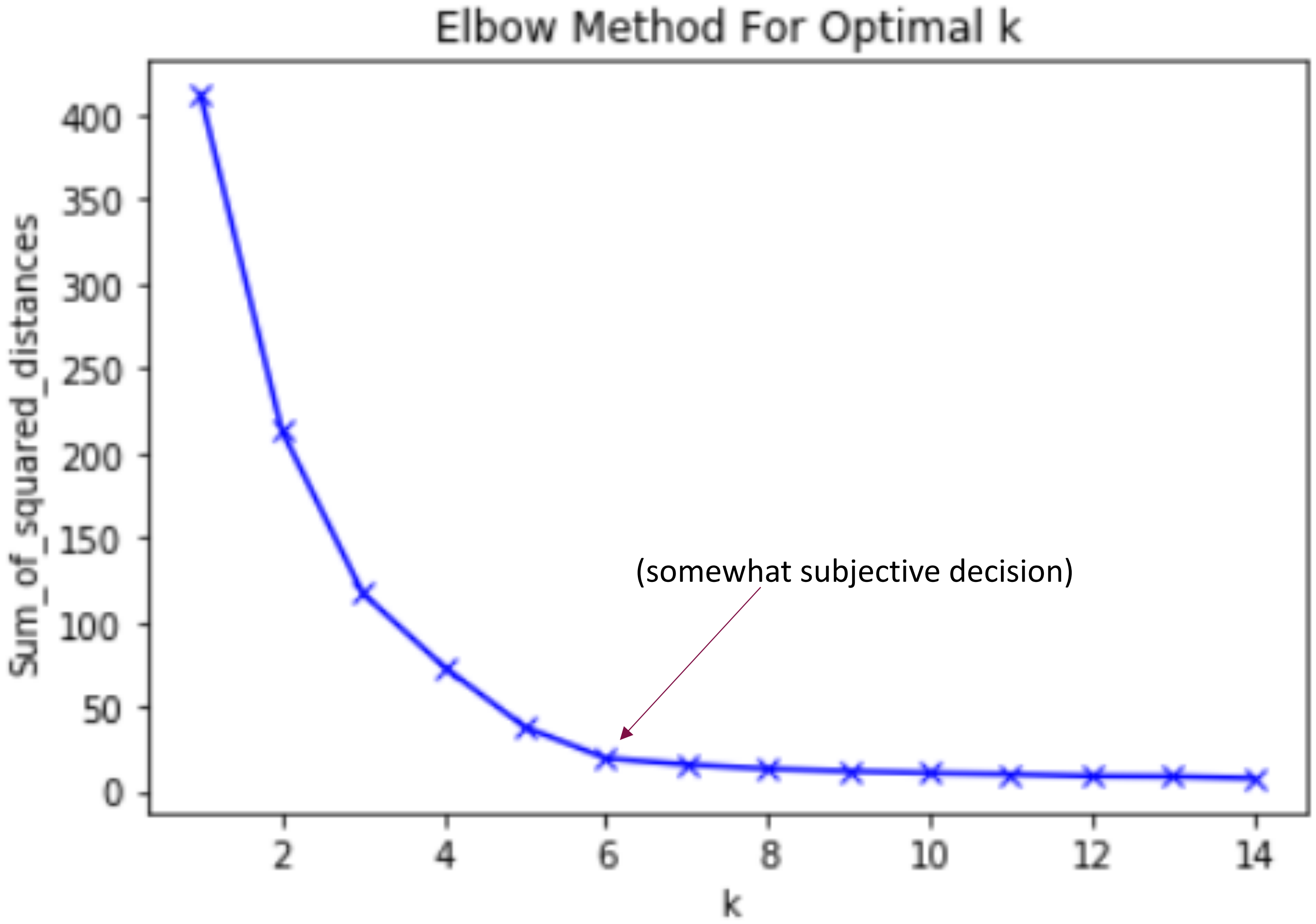
Some options:

- Evaluation using the k-means objective itself
- Comparing to class labels (for a subset of data)

  Sometimes possible
- Subjective evaluation by a human domain expert

...

"Knee Point" For Selecting K



https://blog.cambridgespark.com/how-to-determine-the-optimal-number-of-clusters-for-k-means-clustering-14f27070048f

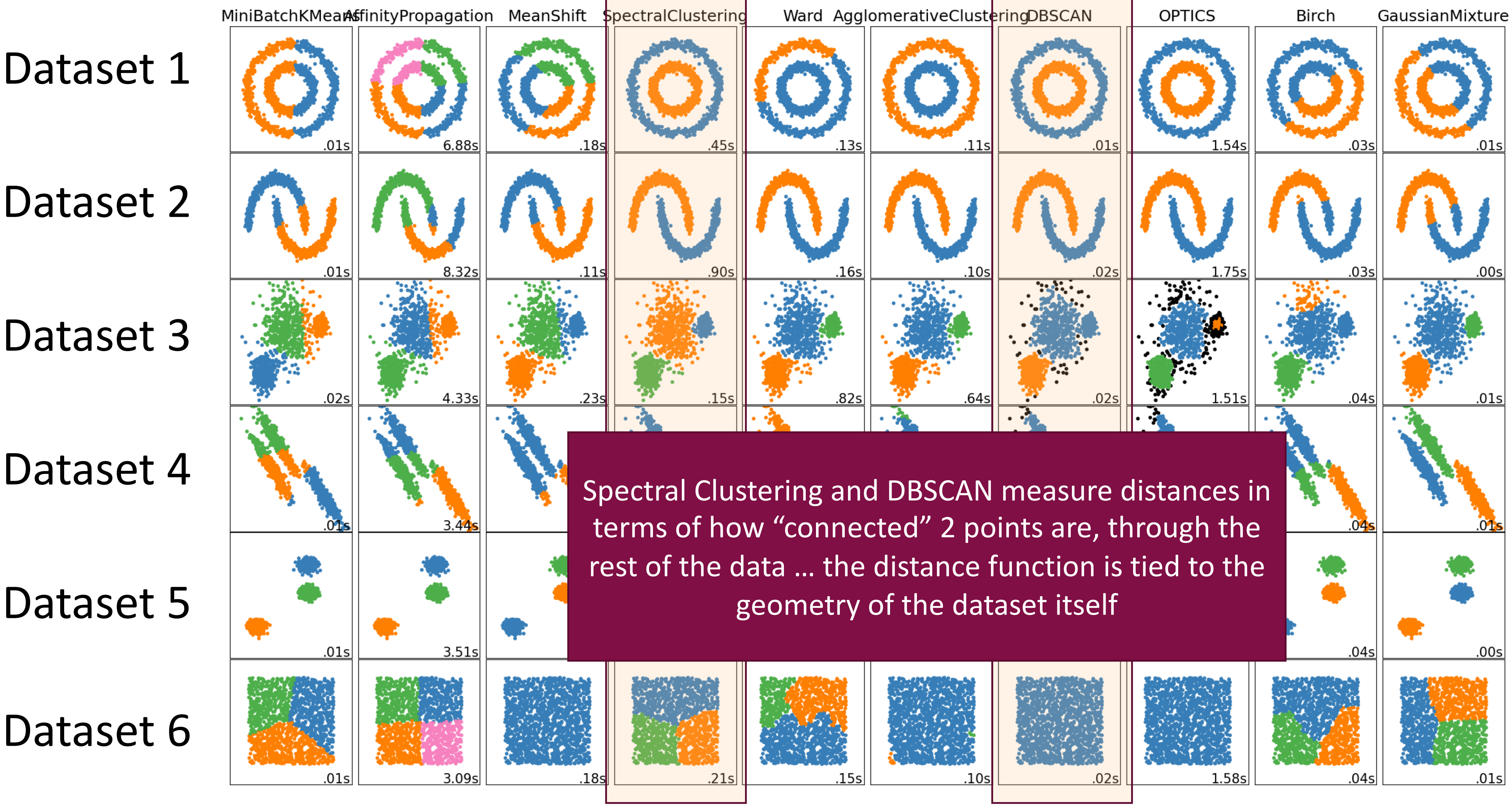
k-Means Clustering

- Non-deterministic (may get different outputs based on initialization) but guaranteed to converge.
- Iterative algorithm with two sub-steps (after random cluster centroids chosen):
    1. Assign points to nearest cluster
    2. Recompute cluster centroid

- Select number of clusters by exploring error (distortion)

There are many other methods of clustering.

Some common clustering algorithms



Spectral Clustering and DBSCAN measure distances in terms of how "connected" 2 points are, through the rest of the data … the distance function is tied to the geometry of the dataset itself

https://scikit-learn.org/stable/modules/clustering.html#clustering

The Choice of Distance Function

Generalizing K-Means to Other Distances

K-Means Objective Function:

$$\arg\min_{\boldsymbol{S}} \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in S_k} \|\boldsymbol{x} - \boldsymbol{\mu}_K\|_2^2$$

But it is possible to define k-means with other notions of pairwise distance between samples too. For example:

$$\left(\sum_d |x_{1d} - x_{2d}|^1 \right)^{\frac{1}{1}}$$

$\boxed{\text{“K-medoids” clustering}}$

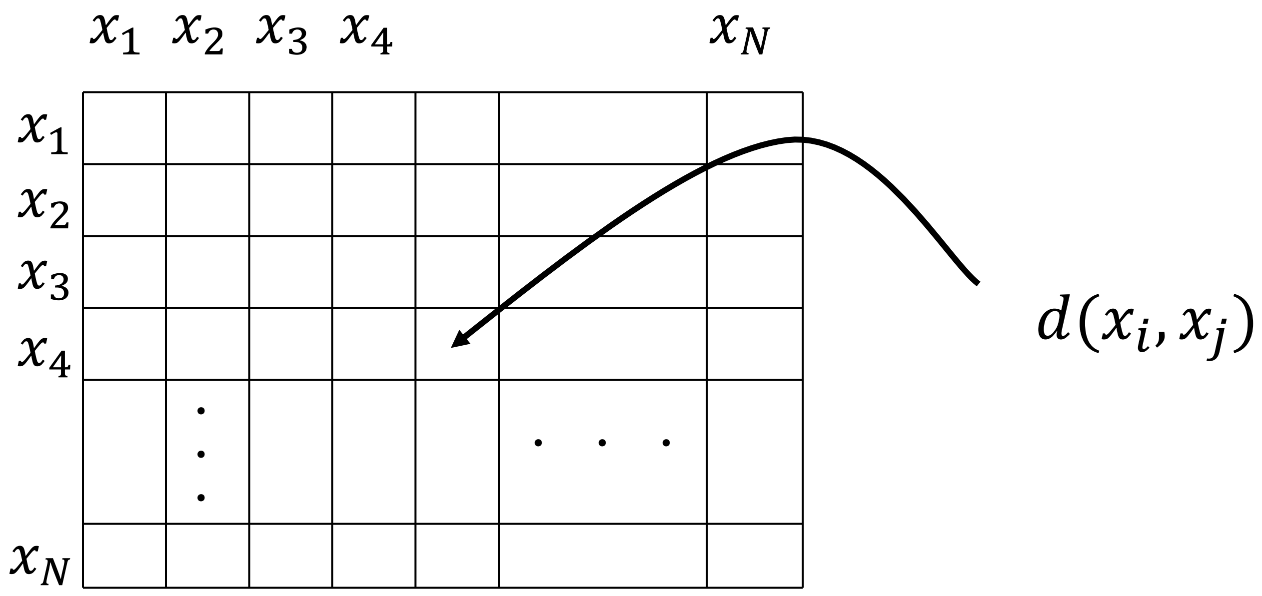
Requires only a small change to the algorithm,
Use medioids instead of centroids.

ℓ_1 distance

$$\sum_d |x_{1d} - x_{2d}|$$

Distance Measures

- More broadly, most clustering techniques can work given an NxN matrix of distances between all pairs of data points.



Distance Measures

Examples:

- Euclidean Distance:

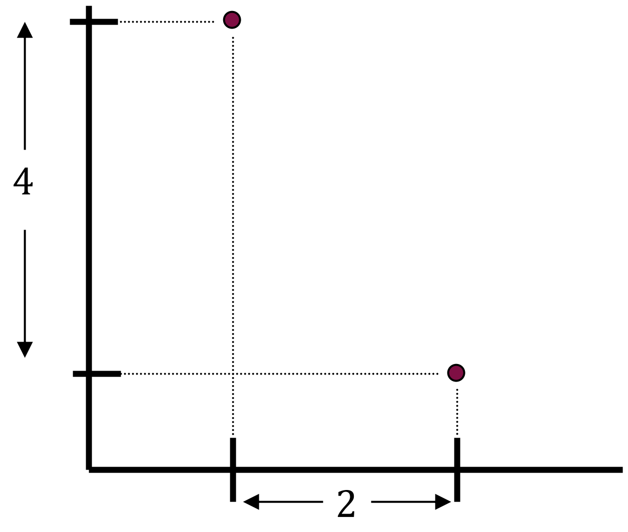
$$d(x,y) = \sqrt{(x-y)^2} = \sqrt{(x-y)^T\,(x-y)} = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

- Manhattan Distance:

$$d(x,y) = |x-y| = \sum_{i=1}^{d}|x_i - y_i|$$

- Infinity (Sup) Distance:

$$d(x,y) = \max_{1\le i\le d}|x_i - y_i|$$



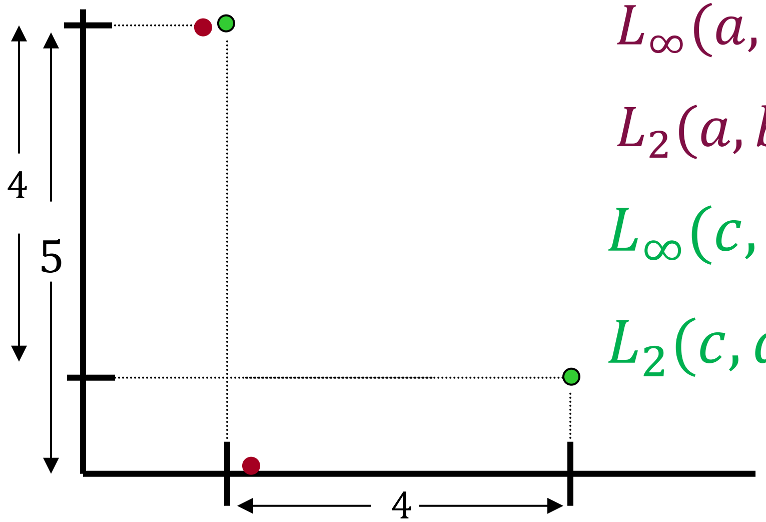
Euclidean: $(4^2 + 2^2)^{\frac{1}{2}} = 4.47$
Manhattan: $4 + 2 = 6$
Sup: $Max(4,2) = 4$

Distance Measures

- For two points $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{R}^{d}$:
  - Infinity (Sup) Distance < Euclidean Distance < Manhattan Distance:

$$L_\infty = \max_{1 \le i \le d} |x_i - y_i| \quad L_2 = \sqrt{(x-y)^2} = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \quad L_1 = |x - y| = \sum_{i=1}^{d} |x_i - y_i|$$

- But different distances do not induce same order on pairs of points



$$L_\infty(a,b) = 5$$

$$L_2(a,b) = (5^2 + \varepsilon^2)^{\frac{1}{2}} = 5 + \varepsilon$$

$$L_\infty(c,d) = 4$$

$$L_2(c,d) = (4^2 + 4^2)^{\frac{1}{2}} = 4\sqrt{2} = 5.66$$

$$L_\infty(c,d) < L_\infty(a,b)$$
$$L_2(c,d) > L_2(a,b)$$

Distance Measures

- Since the discovered clusters depend on the distance measure, one common choice is to use a measure that is invariant to some of the transformations that are natural to the problem.
- **Mahalanobis Distance**: where Σ is a symmetric matrix.

$$d(x, y) = \sqrt{(x - y)^T \, \Sigma(x - y)}$$

- **Covariance Matrix**: Translates all the axes so that they have
  - $Mean = 0$ and $Variance = 1$ (Shift and Scale invariance)

$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$	a column vector, average of the data
$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x - \mu)(x - \mu)^T$	a matrix of size $m \times m$

Summary of Clustering

- Critical to understanding the structure of our data
- Often useful for creating high-level features useful for supervised learning
- We saw two approaches: k-Means vs hierarchical clustering

Optional readings: Clustering

- Bishop Ch 9.1 on K-Means Clustering: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf
- Hands-On ML Unsupervised ML: https://github.com/ageron/handson-ml2/blob/master/09_unsupervised_learning.ipynb (Play with lots of clustering approaches, including K-Means in detail)
- Scikit-Learn documentation of clustering approaches: https://scikit-learn.org/stable/modules/clustering.html#clustering