Biostat 202: Opportunities and Challenges of "Big Data": Machine Learning 5: Clustering and Data Reduction

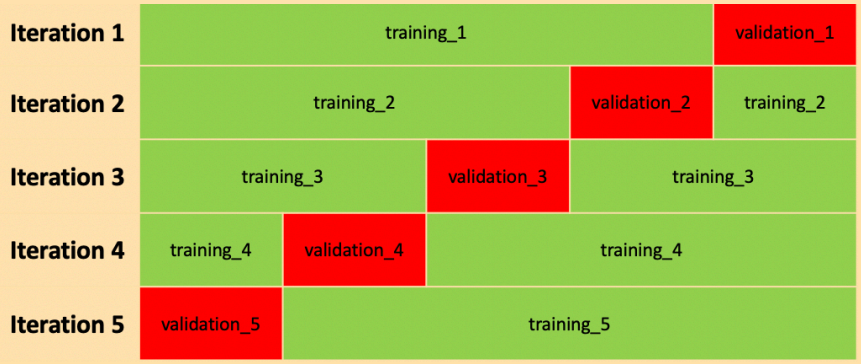
Aaron Wolfe Scheffler

Division of Biostatistics

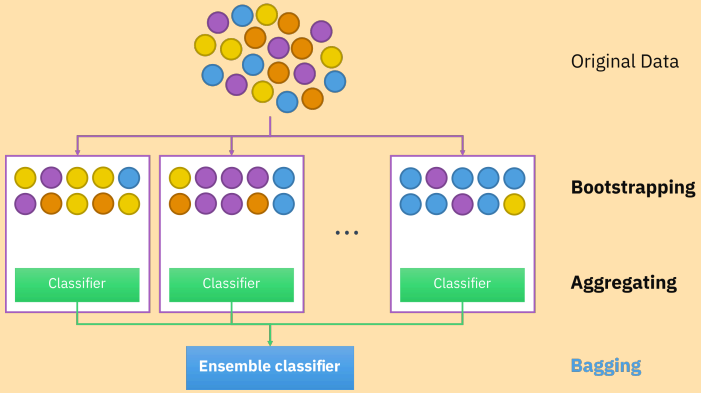
Department of Epidemiology and Biostatistics

Review of last lecture

1. Cross-validation



2. Bagging



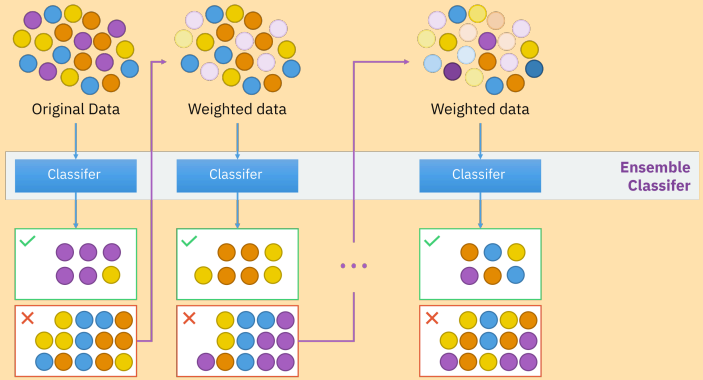
Original Data

Bootstrapping

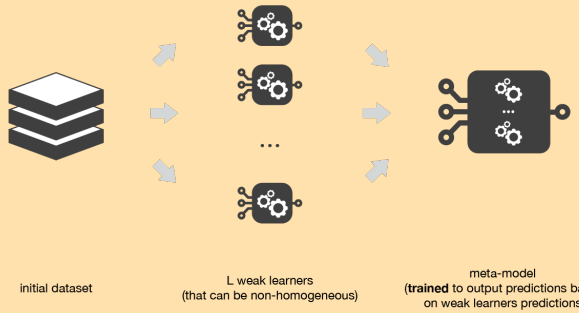
Aggregating

Bagging

3. Boosting



4. Stacking



initial dataset

L weak learners (that can be non-homogeneous)

meta-model (**trained** to output predictions based on weak learners predictions)

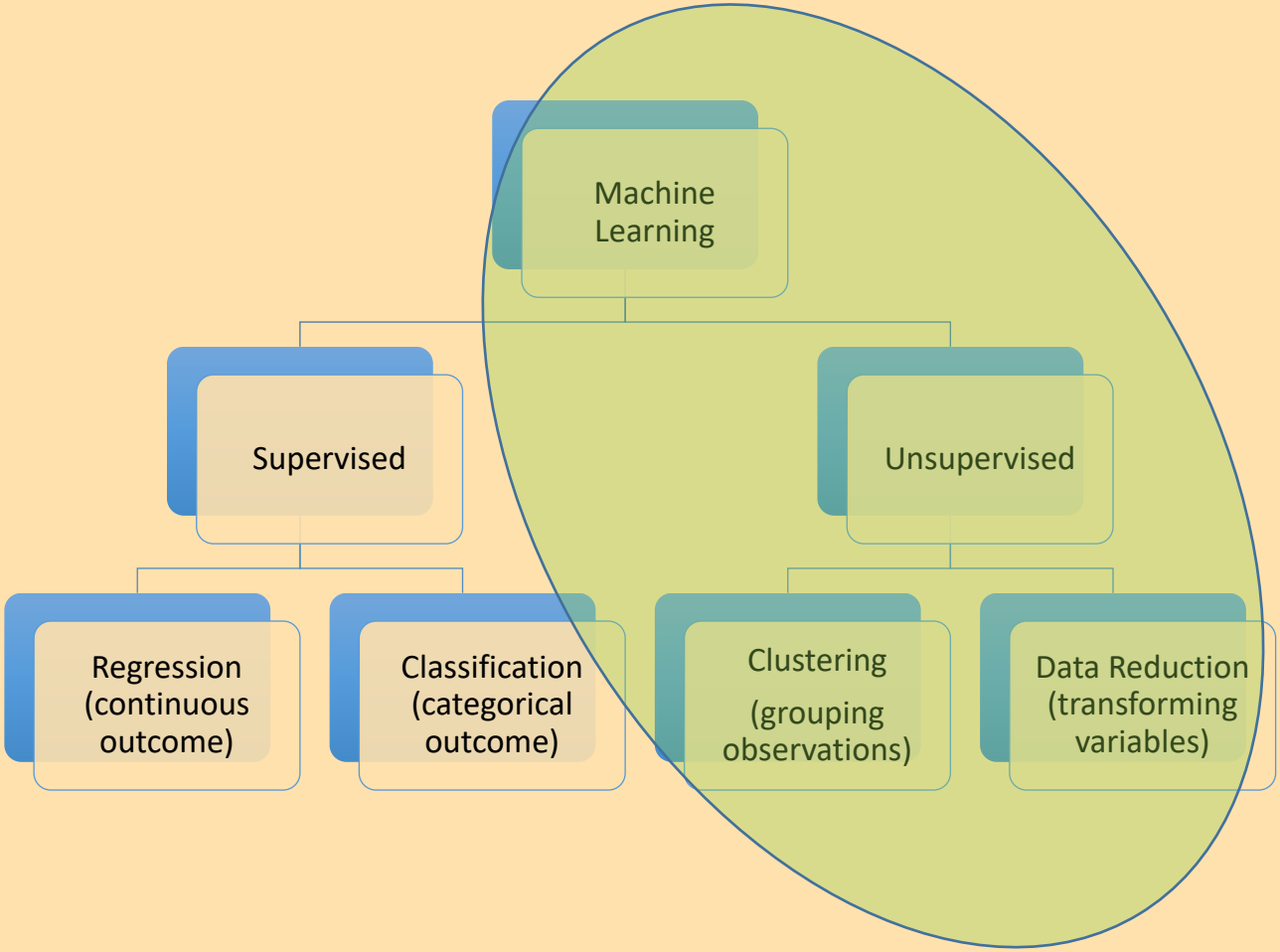
5. Feature Importance

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks
182	155	...	
175	147	...	
...	...	...	
156	142	...	
153	130	...	

Overview of this lecture

- Clustering
  - k-means clustering
  - Hierarchical clustering
- Data reduction
  - Principal components analysis (PCA)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)

Supervised vs. unsupervised



Supervised vs unsupervised

- **Supervised Learning**
- Have data where you know outcome in order to *train* the model
- Train the model to predict future outcomes from sets of predictors

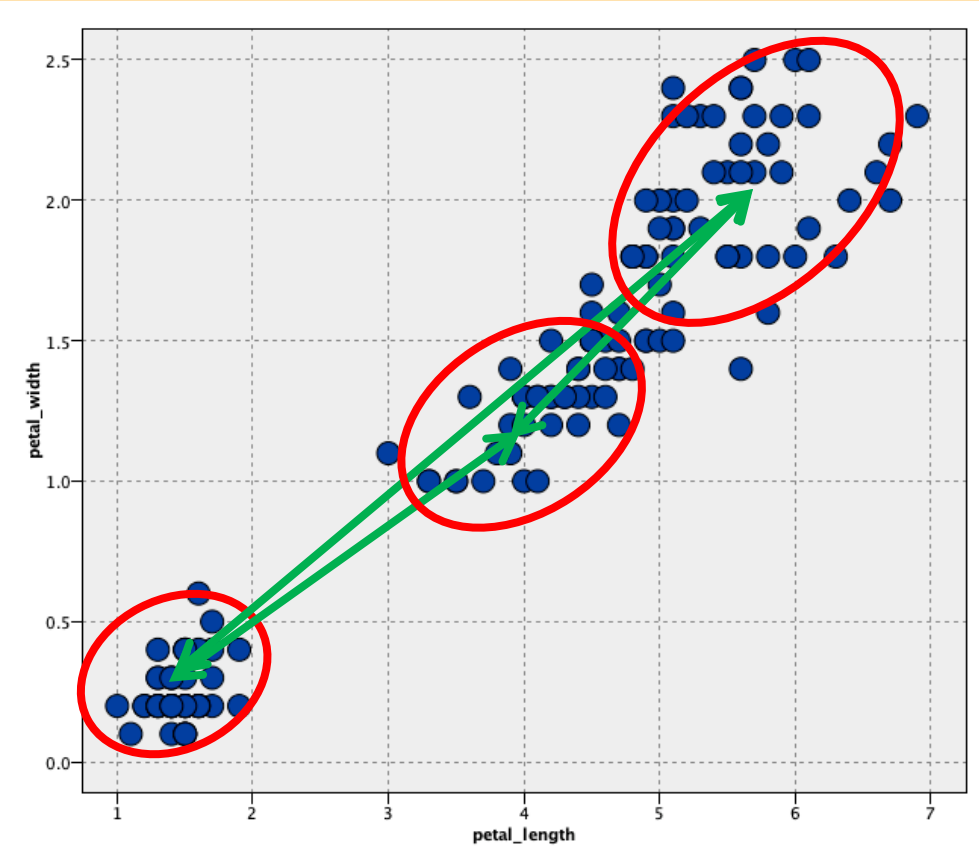
Prediction: Regression (continuous outcomes) and classification (categorical outcomes)

- **Unsupervised Learning**
- No desired outcomes
- Separate data points into groups with "similar" properties (clustering)
- Or reduce data to fewer variables in some way that still captures important structure of the data

Clustering and data reduction

Clustering

2D clustering: Iris example



When determining clusters

We want long green lines (large distance between cluster centers)

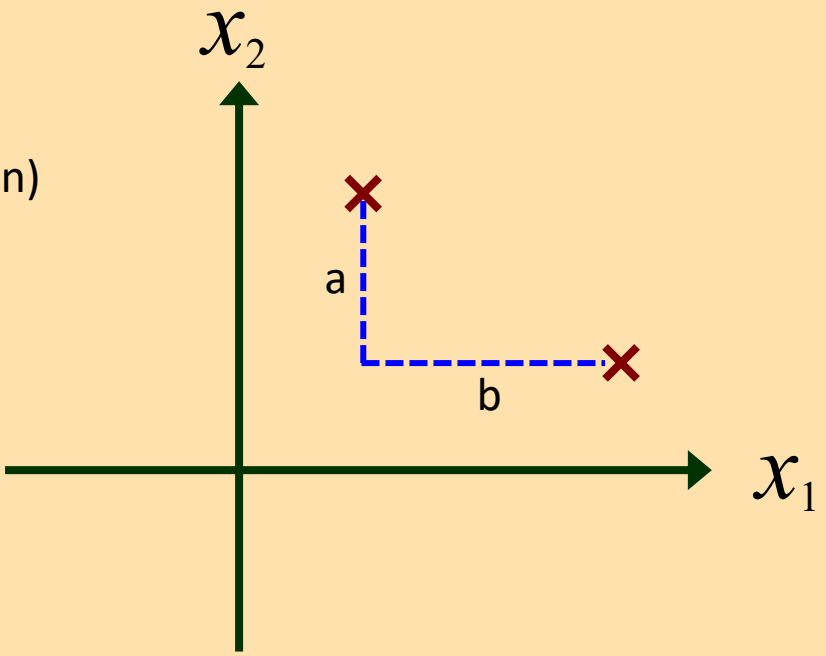
We want small ellipses (points within clusters grouped tightly together)

Goal: maximize similarity within clusters and minimize similarity between clusters

Measures of similarity/dissimilarity

- Need to define similarity (or dissimilarity) and we need to be able to measure it
- Consider 2 observations with only 2 variables x_1 and x_2 – dissimilarities could be

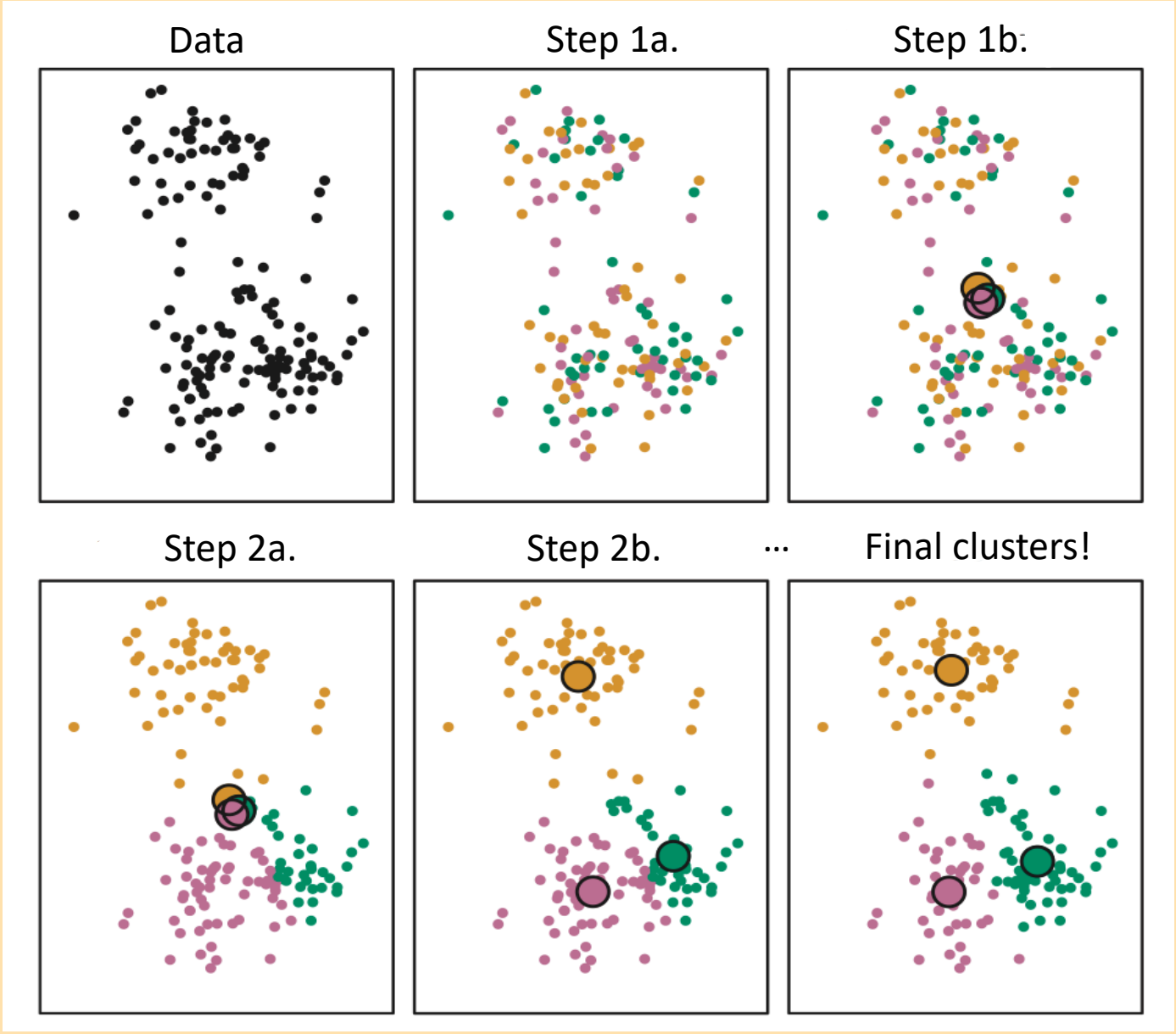
- Euclidean = $\sqrt{a^2 + b^2}$ (most common)
- Absolute distance = $|a| + |b|$
- Can also differentially weight attributes e.g. $\sqrt{ka^2 + b^2}$



Extension to 3D for Euclidean distance would be e.g. $\sqrt{a^2 + b^2 + c^2}$, and so on for higher dimensions

k-means clustering

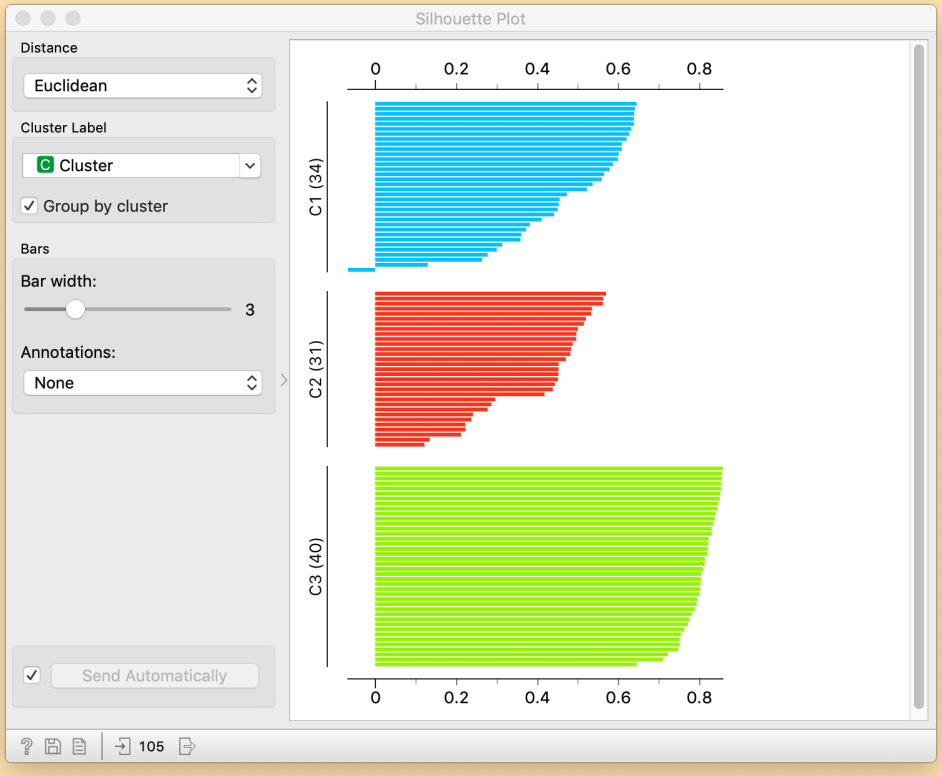
k-means algorithm (k = 3)



Adapted from ISLR, James et al., 2013 p. 389

Silhouette Measure

- Measures the degree of similarity (**cohesion**) of observations within a cluster **compared to** other clusters (**separation**)
- [-1 (dissimilar), 1 (similar)]
- An evaluation metric for cluster assignment
- *Goal*: high average silhouette via clustering



Example silhouette measures for three clusters of individuals.

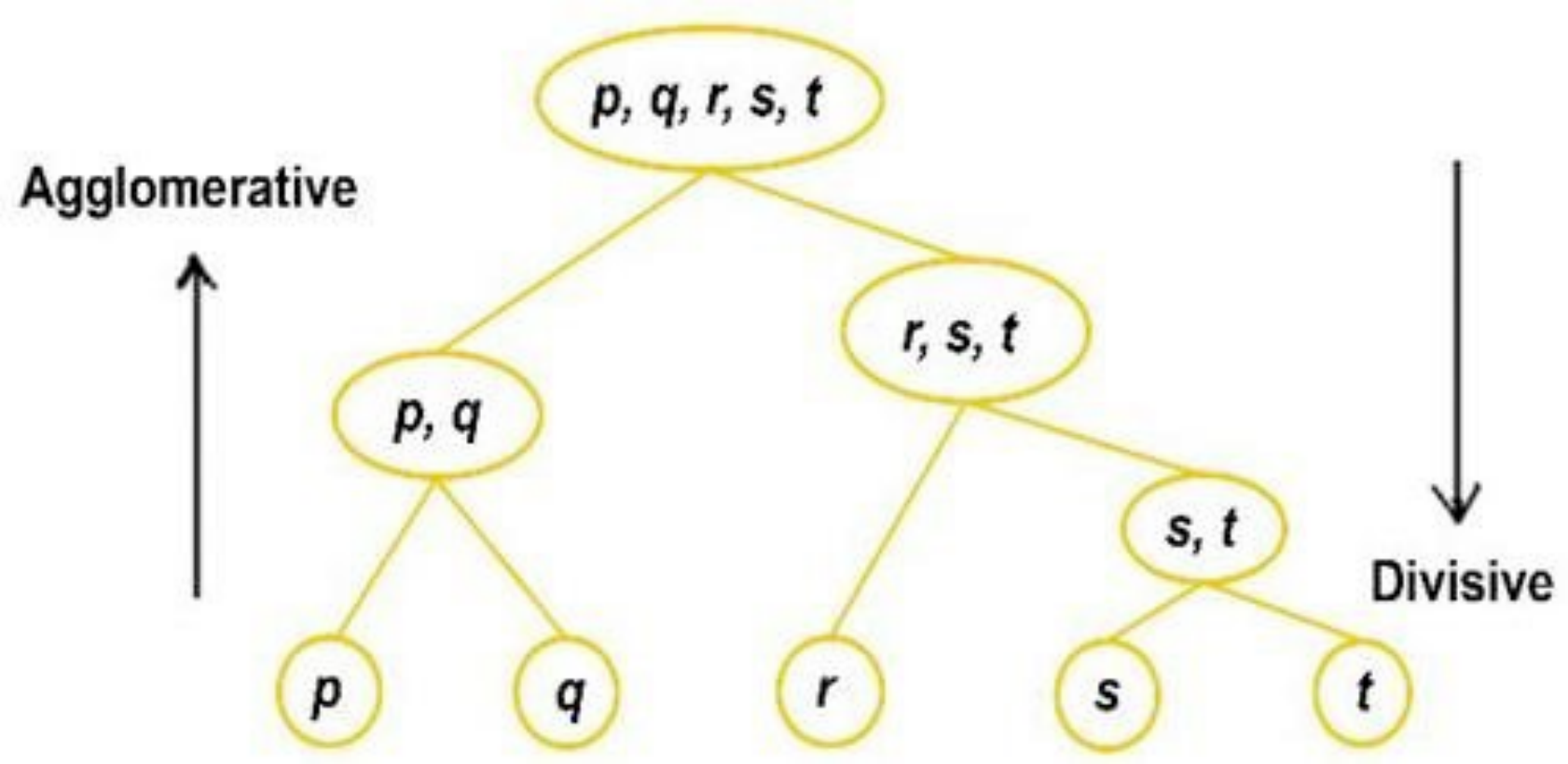
Clustering vs. classification

- Cluster analysis is NOT classification!
- In our Iris example, the clusters correspond to particular species but this comparison is provided for purely expository reasons
- Cluster analyses hypothesizes there is some latent characteristic which generates these clusters, something we might need to construct a label for after our cluster analysis (e.g. Netflix movie genres)
- So we do not wish to predict a label but rather potentially reveal a label or grouping and interpret

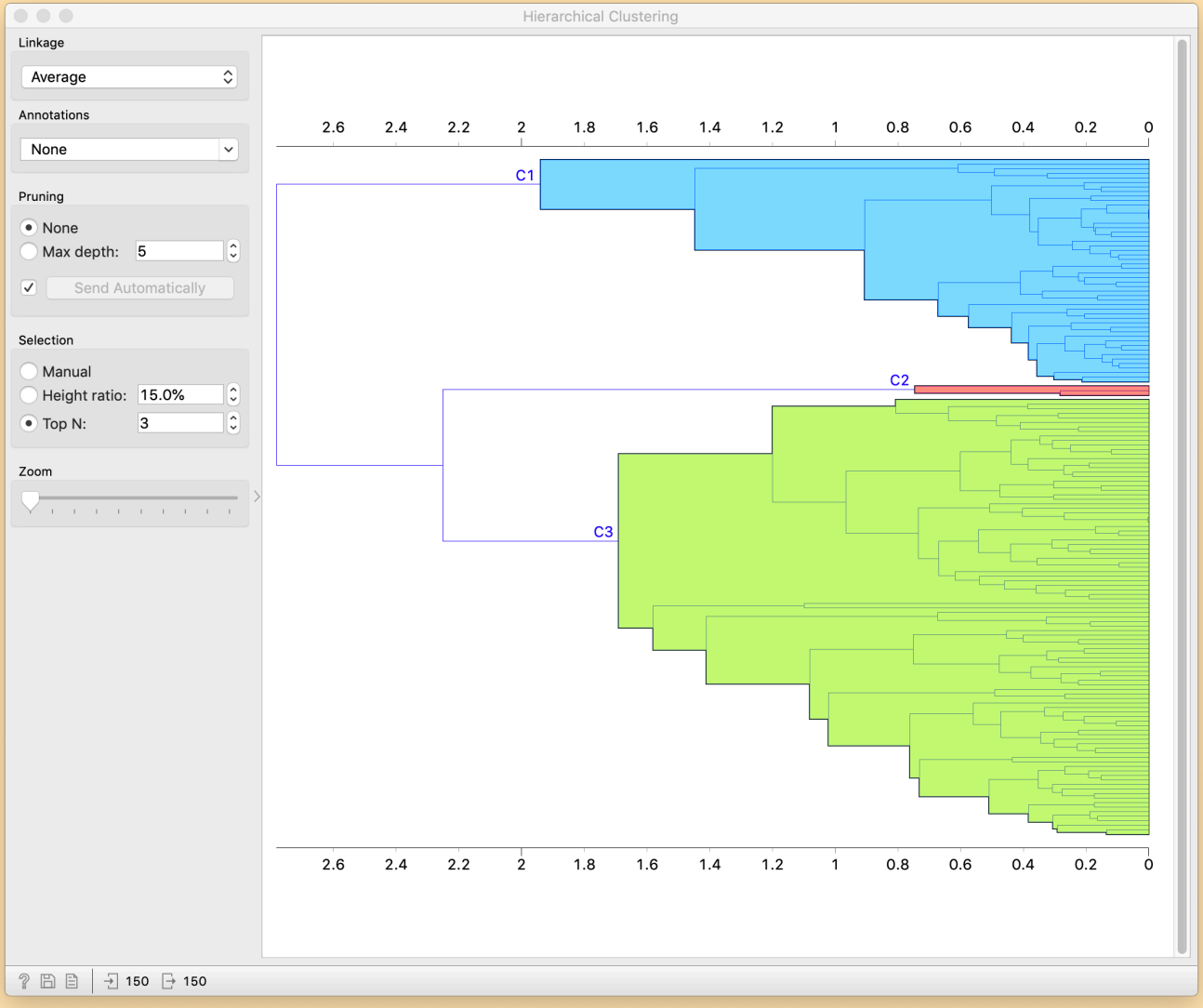
Other clustering algorithms

Hierarchical clustering method

1. Joins (**agglomerative**) or divides (**divisive**) observations
2. Hierarchically joins/divides similar/dissimilar subclusters together based on distance



Iris example – agglomerative hierarchical clustering

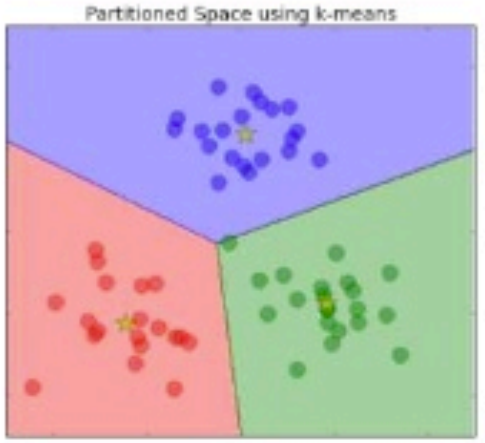


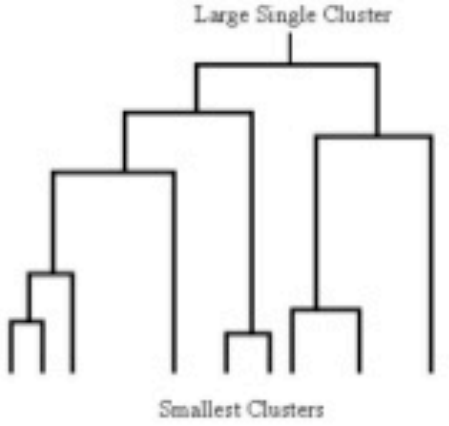
Partitional vs. Hierarchical

1. **Partitional algorithms (k-means)**
   Partition the data space
   Finds all clusters simultaneously

2. **Hierarchical algorithms**
   Generate nested cluster hierarchy
   Agglomerative (bottom-up)
   Divisive (top-down)

   Distance between clusters:
   Single-linkage, complete linkage, average-linkage





Data Reduction Methods

Shadow Analogy

Why data reduction?

- Interpretation:
  - Wish to **visualize** high dimensional data in a 2D/3D space
- Memory/storage:
  - Cannot store raw data, need to reduce to store/analyze
  - E.g. summarize step counts by minute rather than second
- Pre-processing:
  - Reduce dimension of data to perform an analysis
  - E.g. for k-nearest neighbors to avoid the “curse of dimensionality”

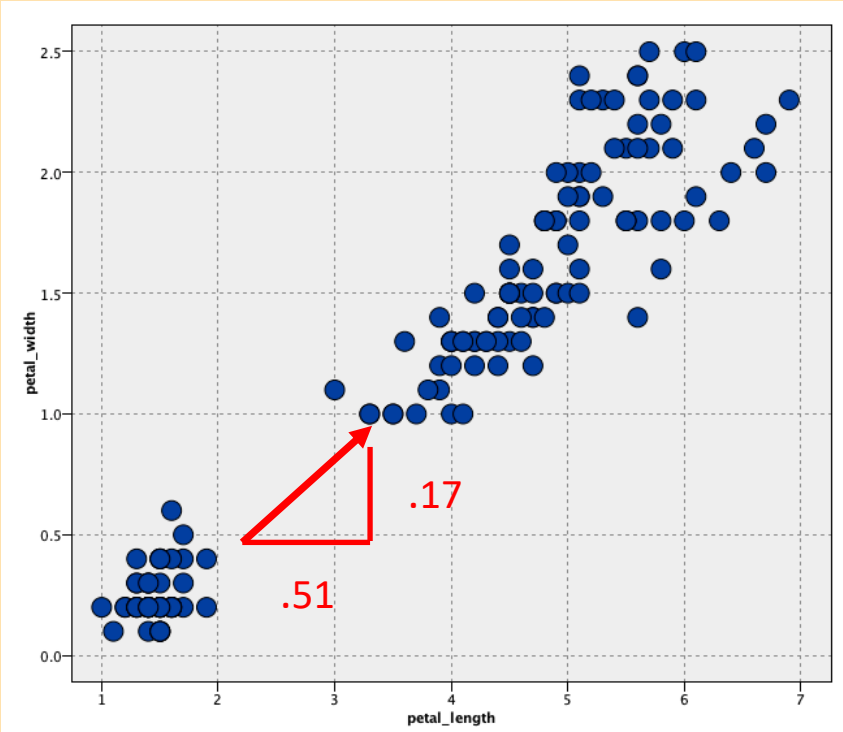
Principal components analysis (PCA)

Principal components analysis

- How does this new mathematical vocabulary fit into “recipe” framework?

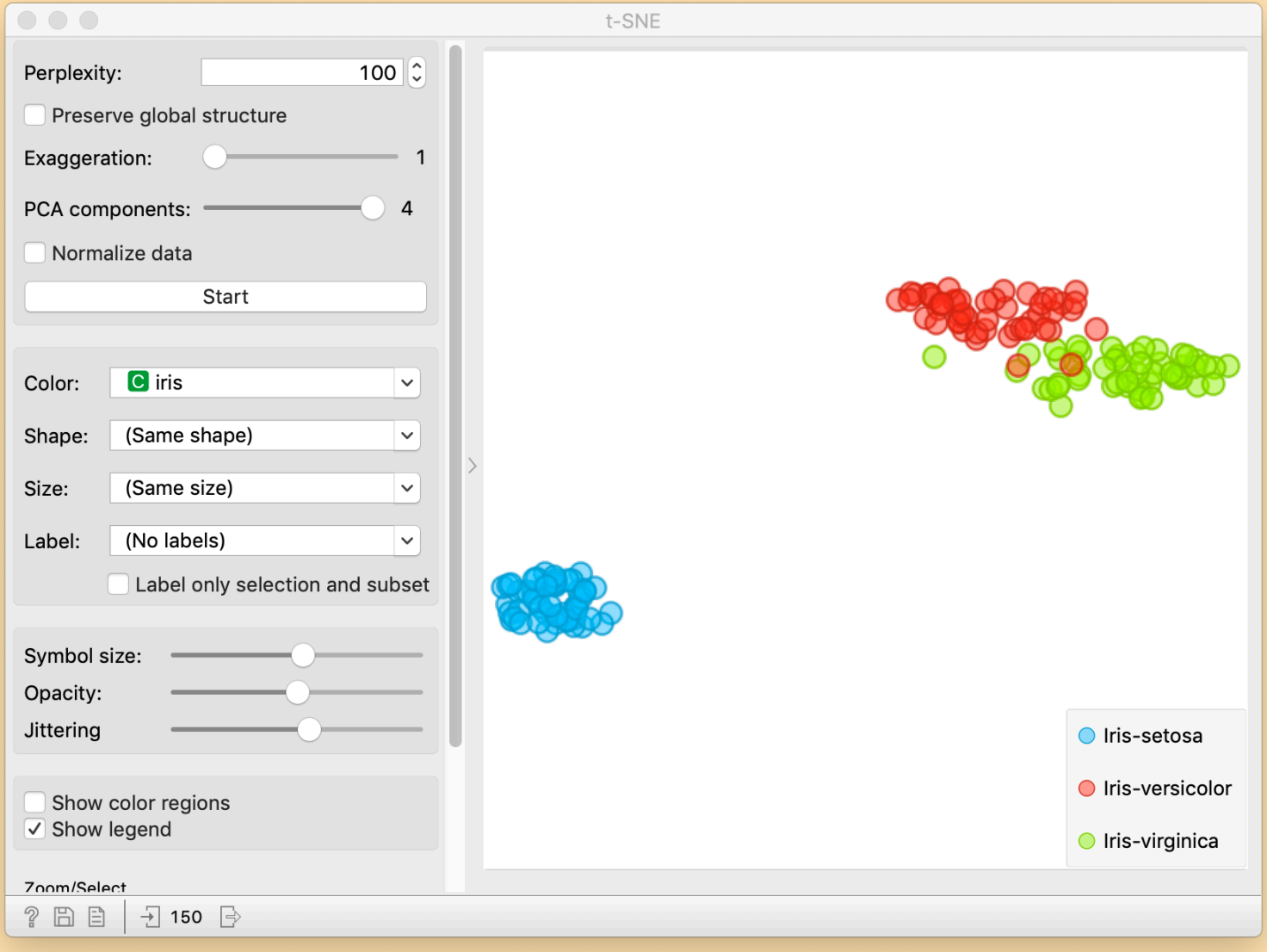
		Principal Components			
Flower 1		sepal_length	sepal_width	petal_length	petal_width
Principal Component Scores	5.9*	.75	.38	.51	.17
+	2.3*	-.29	-.54	.71	.35
		5.09	3.5	1.4	0.2

- Principal components identify the directions of “maximal spread (variation)”



Other dimension reduction methods

t-SNE for Iris Data



Clustering vs Dimension Reduction

Clustering

- Discover latent classes or structure in data
- Maximizes within cluster similarity
- Minimize similarity between clusters
- Cluster labels can be assigned

Dimension Reduction

- Reduces feature dimension to lower dimension
- Can capture leading directions of variation
- Does not assign a label
- Subsequent machine learning can be used on reduced dimension data

Choosing a Machine Learning Method

Choosing a Machine Learning Method

- Common question: “Which machine learning algorithm should I use?”
- Answer: It depends,
  1. Do you have supervised or unsupervised data?
  2. If supervised, do you have continuous, binary class, or multi-class outcome(s)?
  3. If unsupervised, are you interested in looking for latent groupings or data reduction?
  4. Do you have predictors that are continuous, categorical, ordinal, text-based, or some combination of the above?
  5. How large is your training set?
  6. Do you need a fast algorithm?
  7. Do you have time to explore multiple approaches?
  8. Do you need an interpretable model?
  9. For classification, do you need class probabilities, or will “hard” classes suffice?

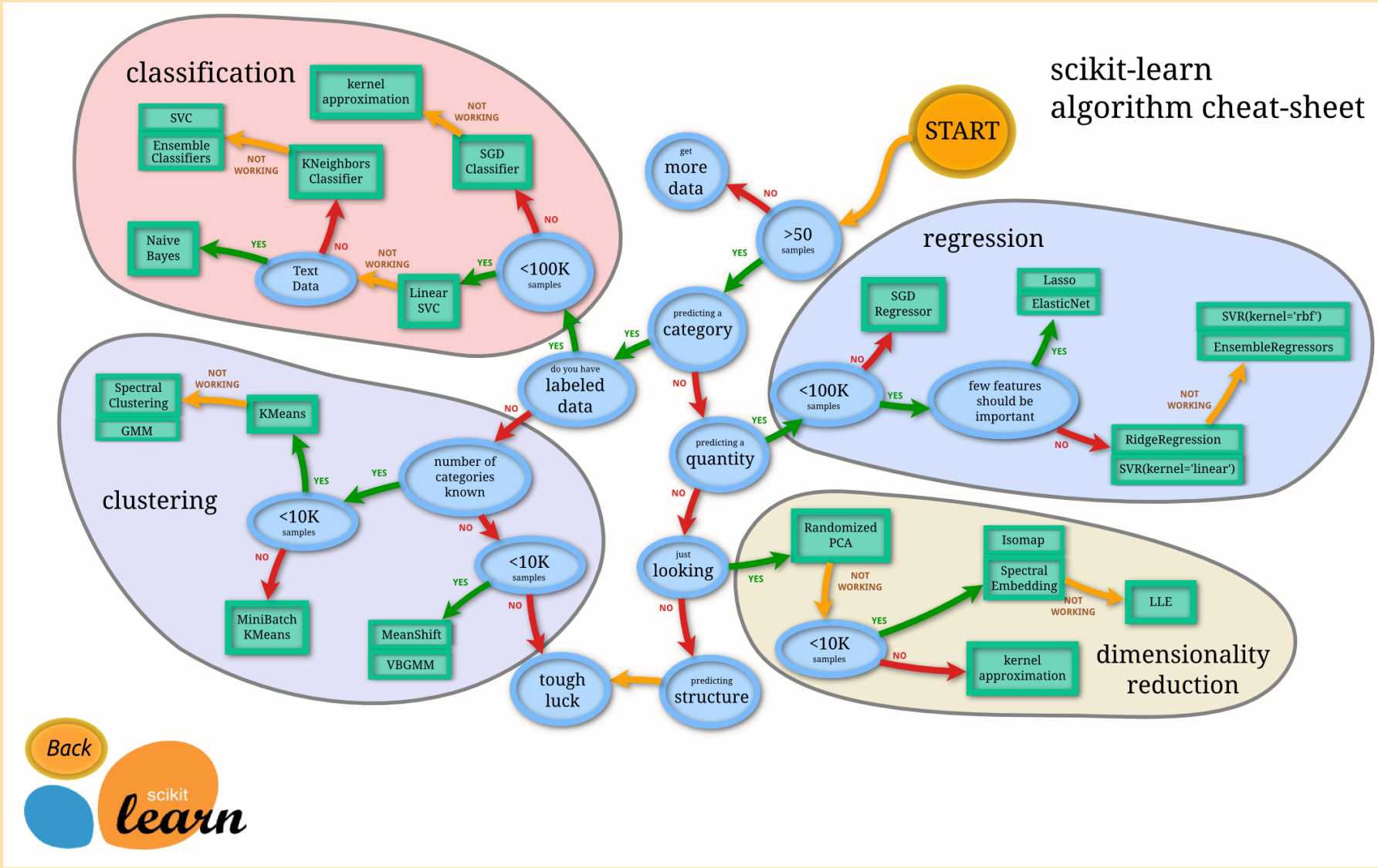
1. Do you have supervised or unsupervised data?
   - Supervised data → use classification or regression methods
   - Unsupervised data → use clustering or data reduction algorithms
2. If supervised, do you have continuous, binary class, or multi-class outcomes?
   - Continuous data → use regression methods (e.g. multiple linear regression)
   - Class data → use classification methods (e.g. classification trees

- If unsupervised, are you interested in looking for natural groupings, or data reduction/simplification?
  - To group data into similar sets → clustering algorithms, e.g. k-means
  - To reduce data → use data reduction algorithms (e.g. principal components analysis, t-SNE)

4. Do you have predictors that are continuous, categorical, ordinal, text-based, or some combination of the above?
   - Some algorithms will not work with certain types of predictors
5. How large is your training set?
   - Small data → "high bias/low variance" methods (e.g. linear regression, logistic regression, SVM)
   - Large data → "low bias/high variance" methods (e.g. decision trees, KNN); lower level of assumptions – need a lot of data to train
6. Do you need a fast algorithm? (e.g. for real-time work)
   - More complex algorithms can be relatively slow, especially if there are algorithm parameters to be optimized (e.g. neural nets) – not likely to be an issue in most medicine datasets
7. Do you have time to explore multiple approaches?
   - If so, you may as well try many and see which does "best" for your data (making sure you are dutifully evaluating on independent data) or use ensemble methods

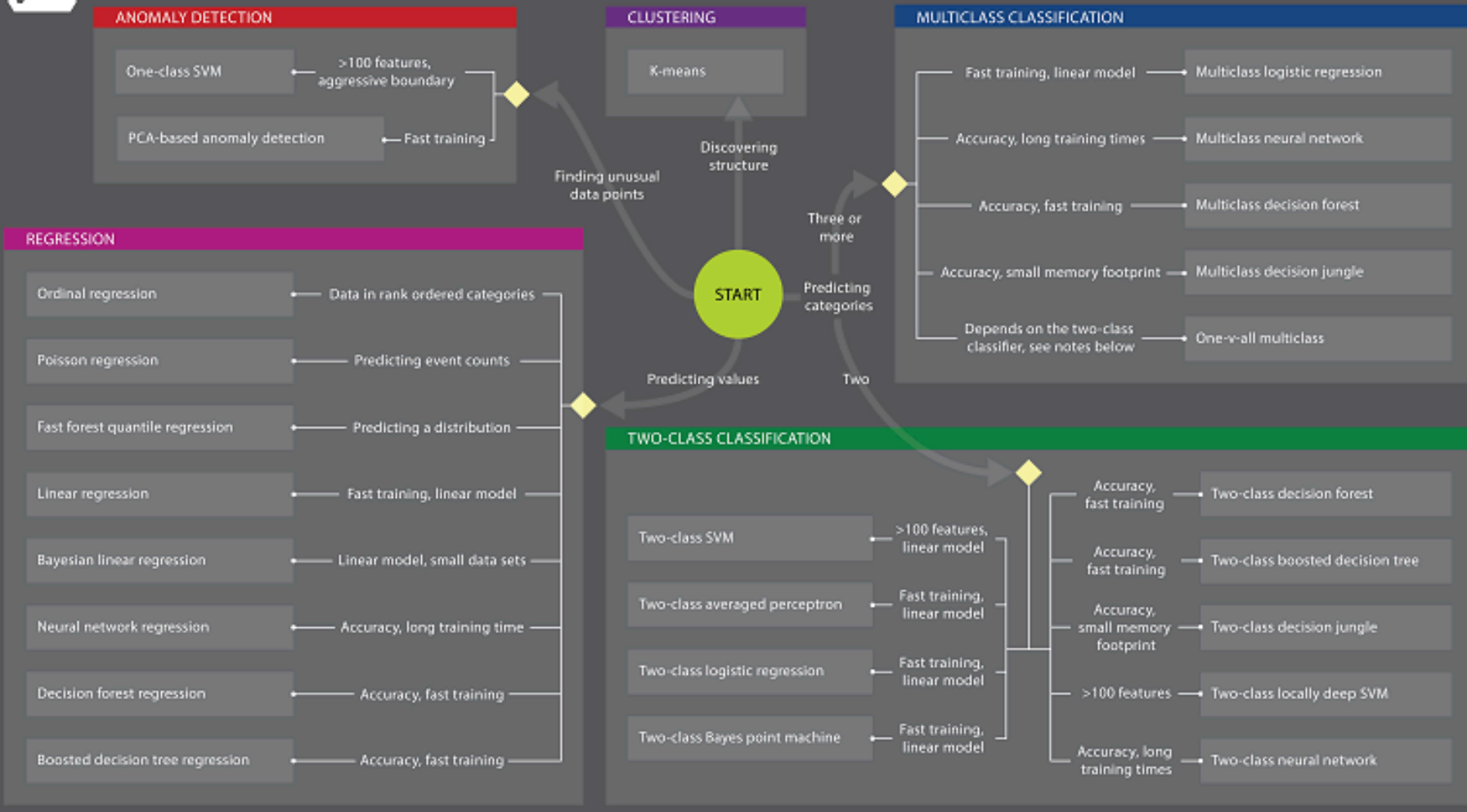
8. Do you need an interpretable model?
   - Consider whether you'll need to understand the role your predictors play in prediction
9. For classification, do you need class probabilities, or will "hard" classes suffice?
   - Some classification methods don't provide class probabilities, so if you want probabilities your options are less

There are also many "cheat sheets" for choosing machine learning algorithms on the internet, but you can see there is a lot of variability…


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
get more data
NO
>50 samples
YES
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
RidgeRegression
NO
SVR(kernel='linear')
predicting a category
YES
NO
do you have labeled data
YES
NO
clustering
Spectral Clustering
NOT WORKING
GMM
KMeans
YES
number of categories known
YES
<10K samples
NO
NO
<10K samples
MiniBatch KMeans
YES
MeanShift
VBGMM
NO
predicting a quantity
YES
NO
just looking
YES
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck
Back
scikit learn

Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.

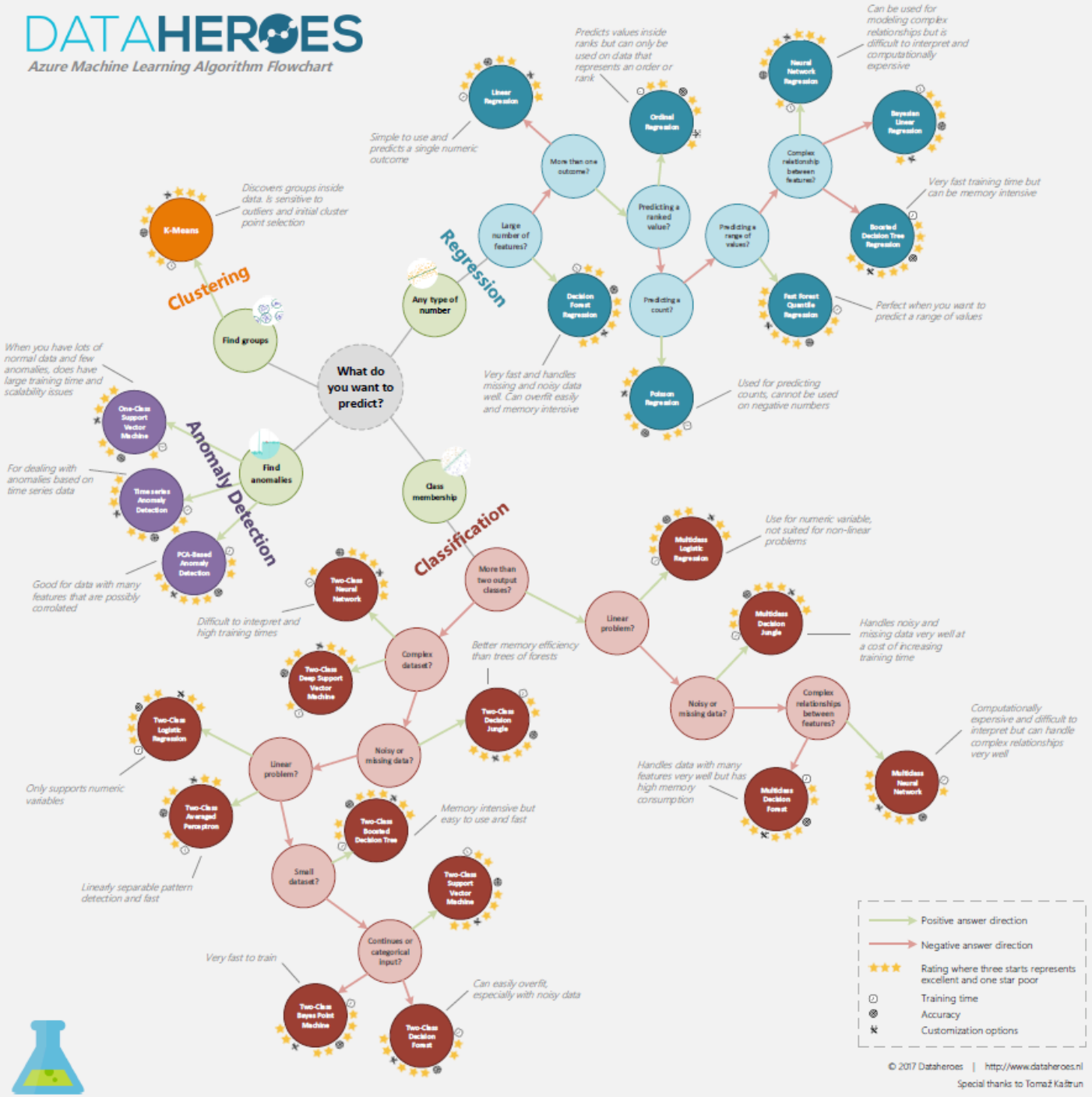


Created by the Azure Machine Learning Team Email: AzurePoster@microsoft.com
Download this poster: http://aka.ms/MLCheatSheet

Microsoft

DATAHEROES

Azure Machine Learning Algorithm Flowchart



Special thanks to Tomaž Kaštrun

Machine Learning Computational Tools

- We have focused on Orange but there are other tools/software/programming languages if you want to dig further. Here are some free ones
- Weka -- http://www.cs.waikato.ac.nz/ml/weka/ - Java based machine learning platform - Data Mining: Practical Machine Learning Tools and Techniques 3rd edition, Witten, Frank and Hall, 2011
- R -- https://www.r-project.org/ -- R has many machine learning libraries and tools. They have a machine learning and statistical learning "Task View" page which much of what is available in R (including an R interface to the Weka package) – packages mlr and caret are wrappers for calling multiple machine learning algorithms
- Python -- https://www.python.org/ -- if you want to go even further down the programming line – many numerical and scientific libraries – see e.g. python resources to get started

Review of this lecture

- Clustering
  - k-means clustering
  - Hierarchical clustering
- Data reduction
  - Principal components analysis (PCA)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)