3000788 Intro to Comp Molec Biol

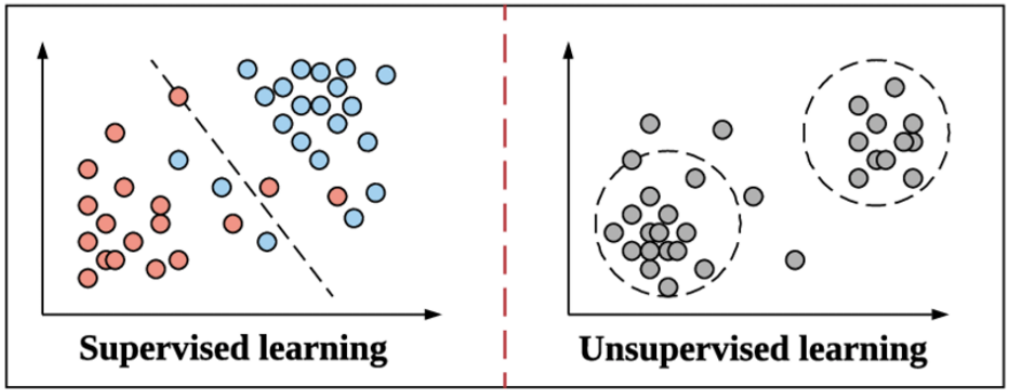
Lecture 27: Principles of machine learning

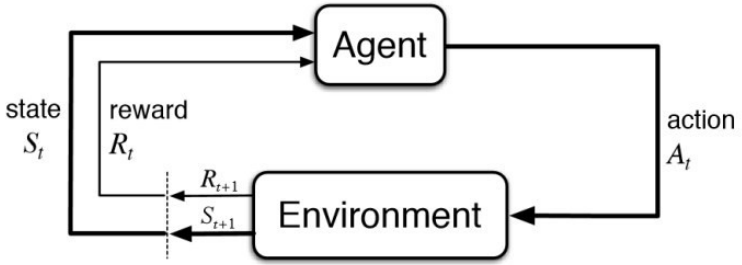
November 20, 2023

Sira Sriswasdi, PhD

- Research Affairs
- Center of Excellence in Computational Molecular Biology (CMB)
- Center for Artificial Intelligence in Medicine (CU-AIM)

Machine learning paradigms





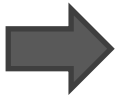
Reinforcement learning

Qian, B. et al. "Orchestrating the Development Lifecycle of Machine Learning-Based IoT Applications: A Taxonomy and Survey"

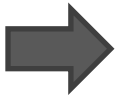
- **Supervised**: Model learns from a dataset (*x*, *y*) to predict *y* from *x*
- **Unsupervised**: Pattern recognition with no target output (only *x*)
- **Reinforcement**: Model learns by interacting and receiving feedbacks from the environment (dynamic data)

Machine learning versus human's way of thinking

Data
+
Hypotheses
Knowledge-based

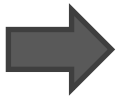


Statistics
Likelihood, goodness-of-fit

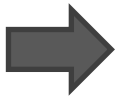


Model that best explains the data

Data
+
Algorithms



Machine Learning
Performance evaluation on unseen data points

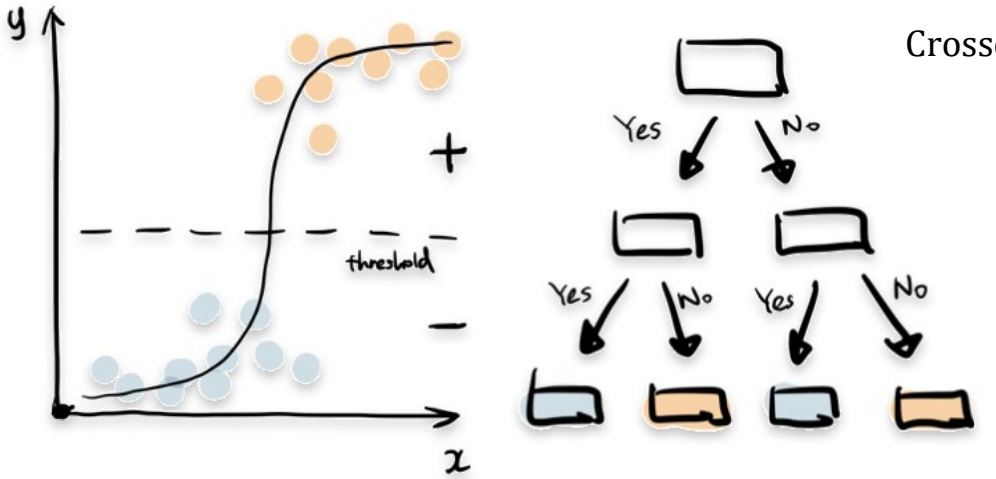


Model that best predicts new data

Supervised learning

The cores of supervised learning

Model



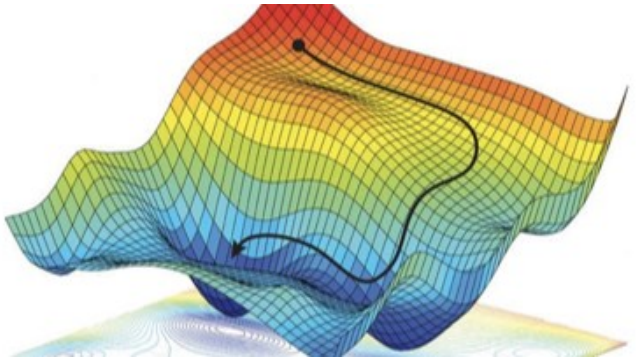
https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501

Objective / Loss Function

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad \text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100$$

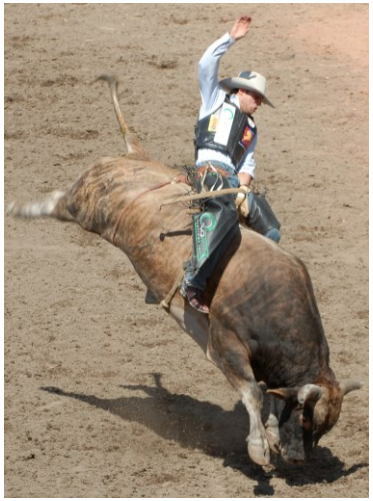
$$\text{Crossentropy} = -\frac{1}{n}\sum_{i=1}^{n} y_i \ln(\hat{y}_i) + (1 - y_i)\ln(1 - \hat{y}_i)$$

Optimization Algorithm

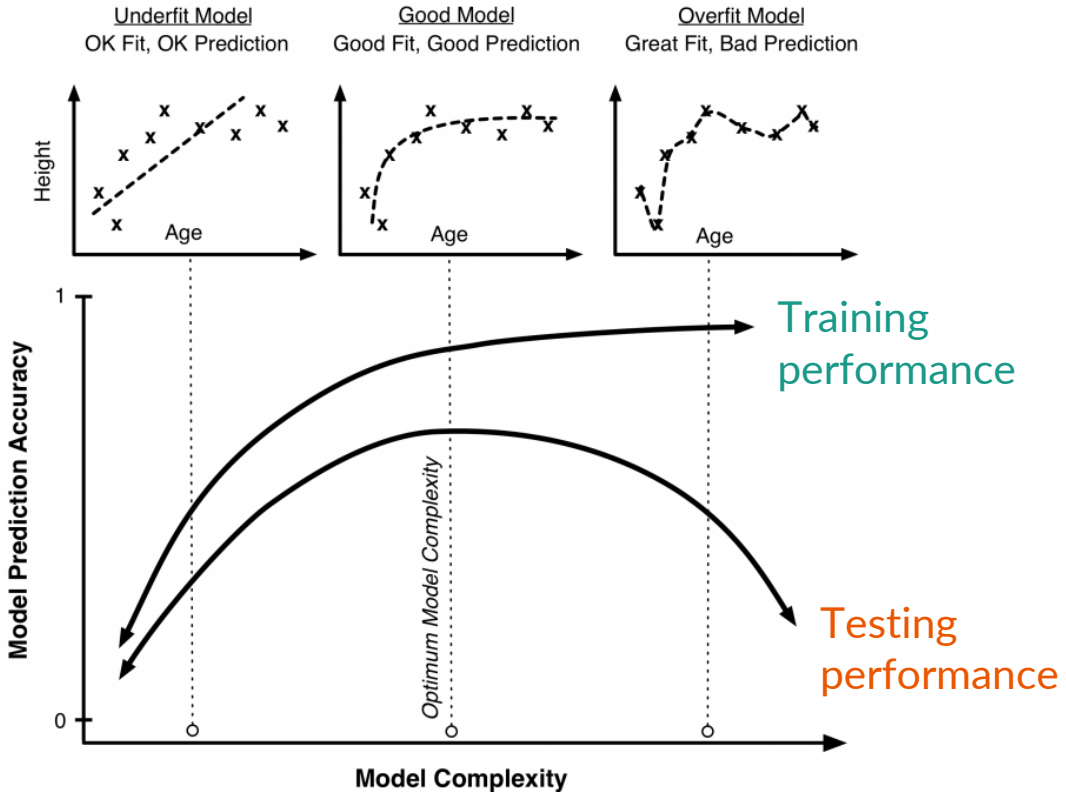


https://medium.com/analytics-vidhya/gradient-descent-b0dc1af33517

Supervised learning is all about control



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



https://realkm.com/2018/04/23/optimization-and-complexity-the-cost-of-complexity-systems-thinking-modelling-series/

Statistical control of overfitting

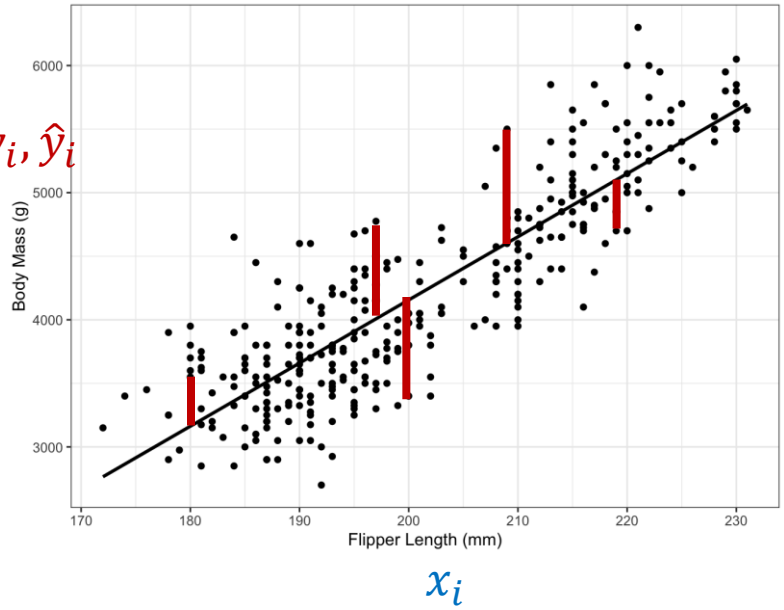
- Better model achieves higher likelihood
- Complex model has more parameters

- **Information Criterion**
  - Akaike (AIC) $= 2k - 2 \cdot \ln(\hat{L})$, where $\hat{L}$ is the likelihood
  - Bayesian (BIC) $= \ln(n)\, k - 2 \cdot \ln(\hat{L})$, where n is the sample size

- **Nested model testing**
  - Simple model has n parameters, fit the data with likelihood $\widehat{L_1}$
  - Complex model has $m > n$ parameters, fit the data with likelihood $\widehat{L_2} > \widehat{L_1}$
  - Is the improvement $\frac{\widehat{L_2}}{\widehat{L_1}}$ worth the increase in $m - n$ parameters?

Linear and logistic regression

Linear regression (Ordinary Least Square)



- Model: $\hat{y_i} = b_0 + b_1 x_i$
- Minimize MSE: $\frac{1}{n}\sum_i (y_i - [b_0 + b_1 x_i])^2$

- $\frac{\delta MSE}{\delta b_o} = -2\sum_i y_i - 2b_1 \sum_i x_i - 2nb_0$
- $\frac{\delta MSE}{\delta b_1} = -2\sum_i x_i y_i - 2b_1 \sum_i x_i^2 - 2b_0 \sum_i x_i$

- $b_0 = \frac{\sum xy \sum x - \sum x^2 \sum y}{(\sum x)^2 - n\sum x^2}$
- $b_1 = \frac{\sum y \sum x - n\sum xy}{(\sum x)^2 - n\sum x^2}$

Ordinary Least Square interpretation

- Observed value = True value + Normally-distributed noise
- **Assumption**: Noises are identical and independent across samples

- Model: $(y_i - \hat{y_i}) \sim N(0, \sigma^2)$
- Density: $\mathrm{P}(y_i - \hat{y_i} = \varepsilon_i \mid \sigma^2) \propto e^{\frac{-{\varepsilon_i}^2}{2\sigma^2}}$
- Likelihood: $\prod_i \mathrm{P}(y_i - \hat{y_i} = \varepsilon_i \mid \sigma^2) \propto e^{\frac{-\sum_i {\varepsilon_i}^2}{2\sigma^2}}$

- MSE: $\frac{1}{n}\sum_i (y_i - \hat{y_i})^2 = \frac{1}{n}\sum_i {\varepsilon_i}^2$
- Minimizing MSE is the same as maximizing likelihood

Logistic regression

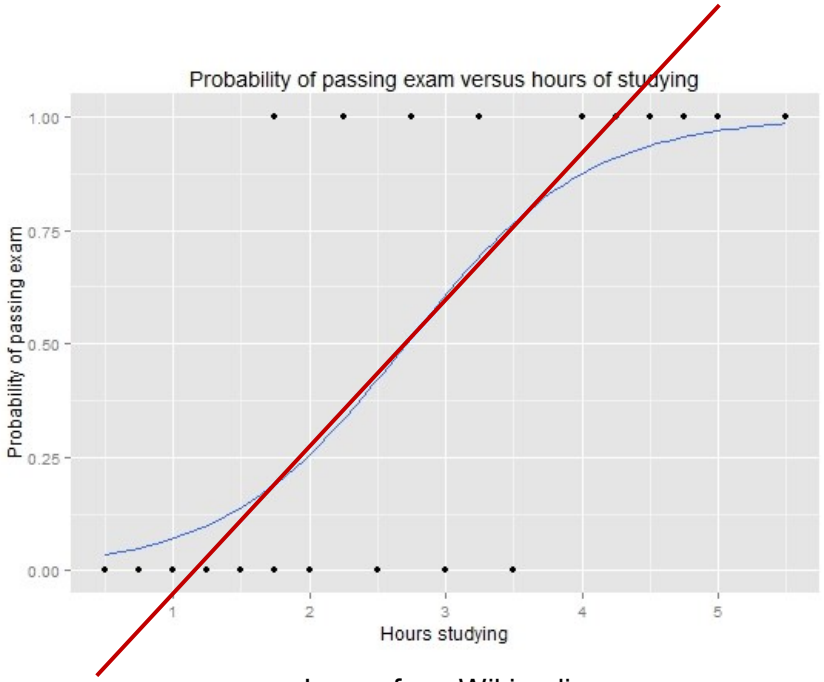


Image from Wikipedia

- Classification output = 0 or 1
- Linear regression outputs $-\infty$ to ∞

- Probability of success p
- Log odd: $\ln\left(\frac{p}{1-p}\right)$
  - $\ln\left(\frac{p}{1-p}\right) \to -\infty$ as $p \to 0$
  - $\ln\left(\frac{p}{1-p}\right) \to \infty$ as $p \to 1$

- Transform linear regression output with log odd!

Logistic regression

- Model: $\ln\left(\frac{\widehat{y_i}}{1-\widehat{y_i}}\right) = f(x_i) = b_0 + b_1 x_{i,1} + \cdots + b_n x_{i,n}$
- $\widehat{y_i} = \frac{e^{b_0+b_1 x_{i,1}+\cdots+b_n x_{i,n}}}{1+e^{b_0+b_1 x_{i,1}+\cdots+b_n x_{i,n}}}$
  - When $f(x_i) \to \infty$, $\widehat{y_i} \to 1$
  - When $f(x_i) \to -\infty$, $\widehat{y_i} \to 0$

- Can we keep using MSE as the loss function?
  - Brier score $= \frac{1}{N}\sum_i (y_i - \widehat{y_i})^2$
  - But this does not interpret logistic output as probability

Likelihood for logistic regression

- Likelihood: $\mathrm{P}(y_i \mid x_i) = \hat{y}_i^{\,y_i}(1-\hat{y}_i)^{1-y_i}$
  - y_i is either 0 or 1
  - When y_i is 0, the likelihood is $1-\hat{y}_i$
  - When y_i is 1, the likelihood is $\hat{y}_i$

- Log likelihood: $y_i \ln(\hat{y}_i) + (1-y_i)\ln(1-\hat{y}_i)$
  - This is the cross-entropy loss function!
  - Maximizing likelihood is the same as minimizing cross-entropy

Regularization of linear model

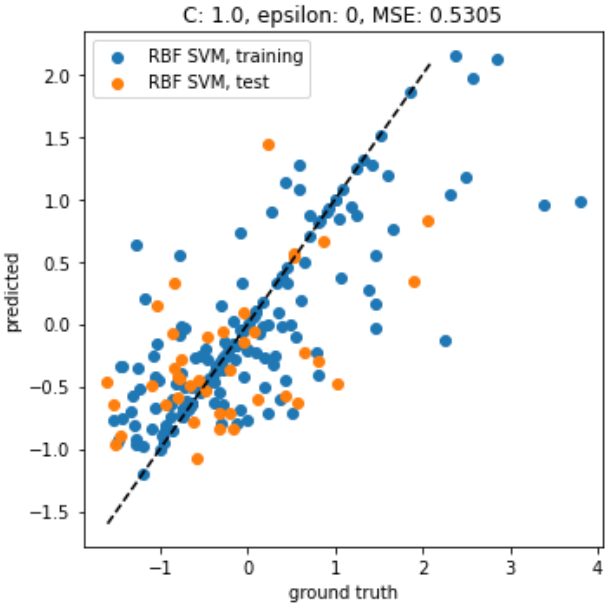
- L1 regularization (LASSO): MSE + $\alpha \sum_k |b_k|$
- L2 regularization (Ridge): MSE + $\alpha \sum_k {b_k}^2$

- α is the hyperparameter that controls the regularization strength

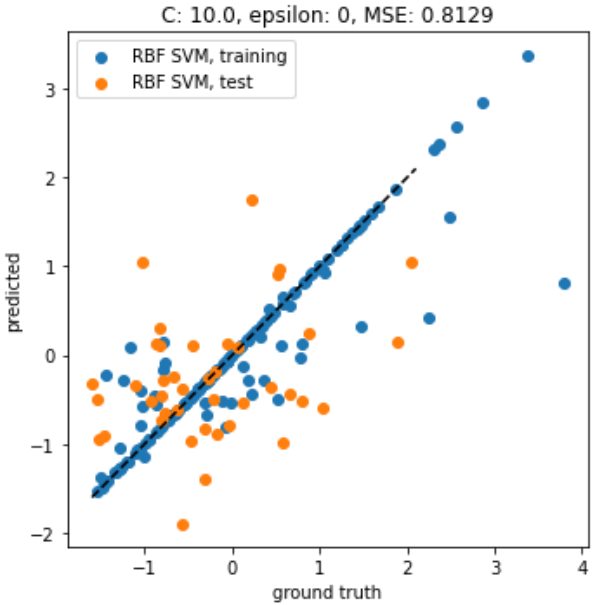
- Hyperparameter must be tuned for every dataset!

Tuning regularization strength

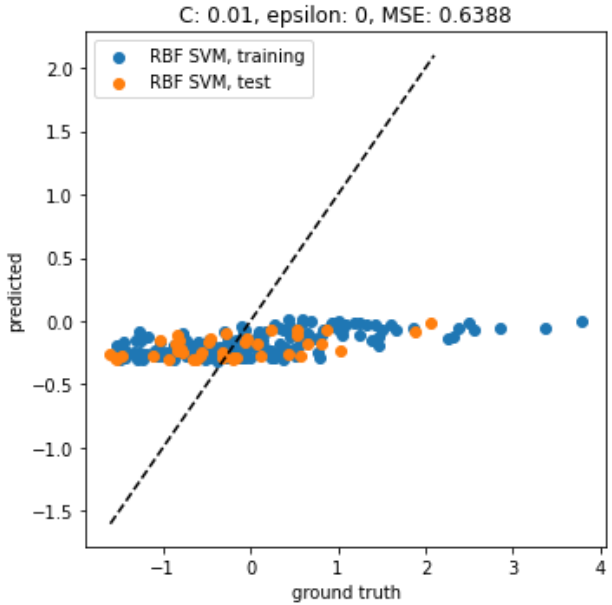
Just right



Too weak

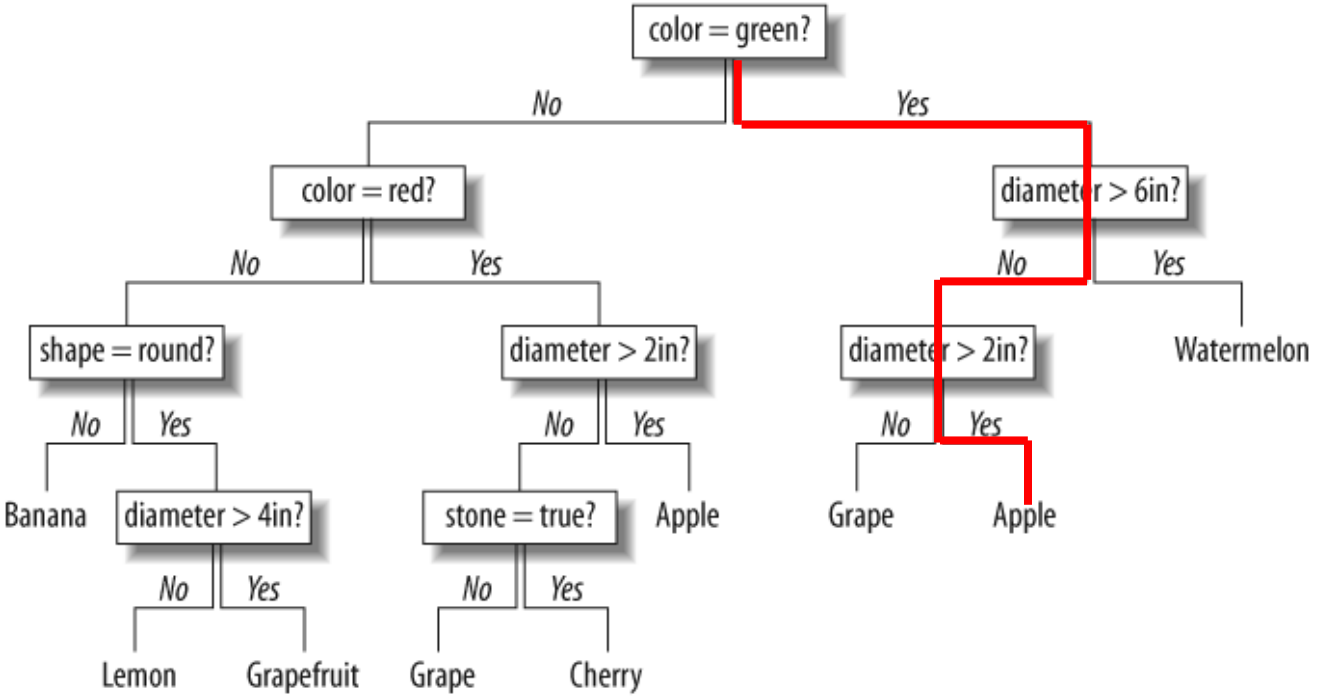


Too strong



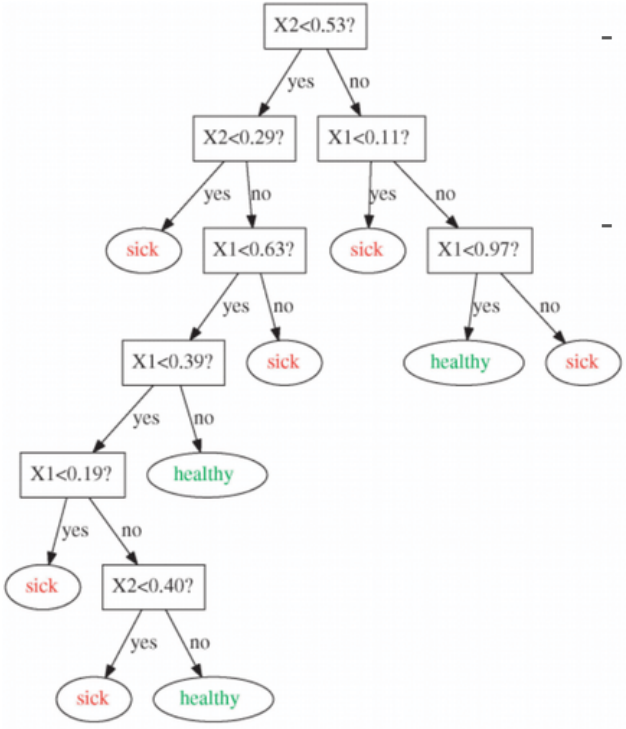
Decision tree

Decision tree

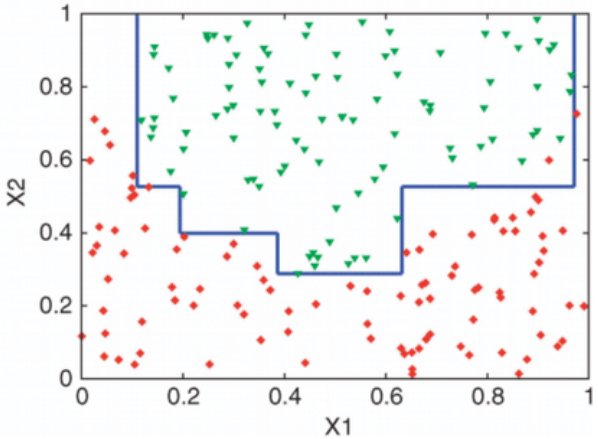


Source: Programming collective intelligence by Toby Segaran

Decision tree behaviors



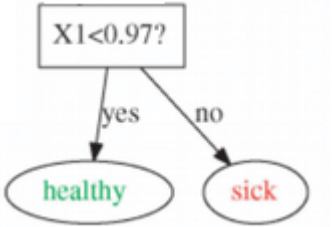
- Each decision is a threshold on each feature
  - Piecewise linear
  - Parallel to an axis
- Good for criteria-based classification



Miller, C. “Screening meter data: Characterization of temporal energy data from large groups of non-residential buildings”

Splitting quality

100 healthy, 100 sick



85 healthy, 5 sick

15 healthy, 95 sick

45 healthy, 51 sick

55 healthy, 49 sick

- Gini impurity: $\sum p(1-p)$
- Entropy: $-\sum p \ln(p)$
  - Minimal at p = 0 or 1 → Perfect split
  - Maximal at p = 0.5 → 50-50 split
- Search for feature and cutoff that yield lowest impurity or entropy

Control mechanisms for tree building

1. Too few samples to make a split

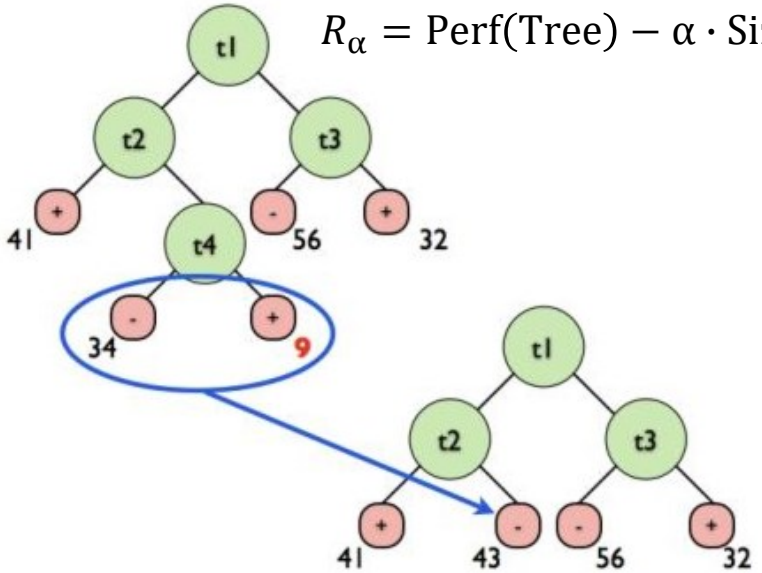
3. Impurity or entropy does not change much after the split

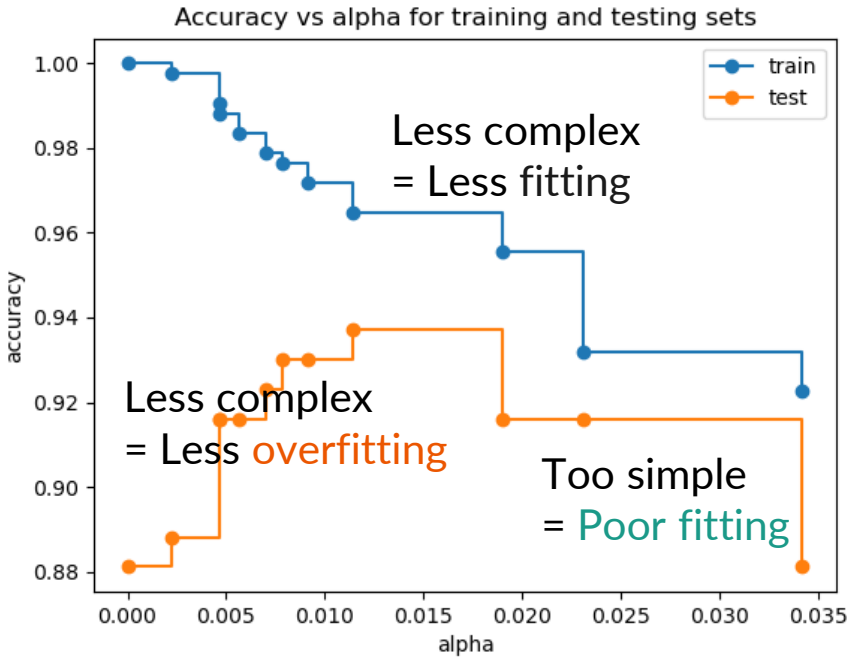
2. Too few samples on either branch

- Limit the tree size
- Limit the improvement in quality
- Limit the number of samples that support a split

Tree pruning (post-processing)

$$R_\alpha = \text{Perf}(\text{Tree}) - \alpha \cdot \text{Size}$$





Patel, N. and Upadhyay, S. “Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA” IJCA 2012

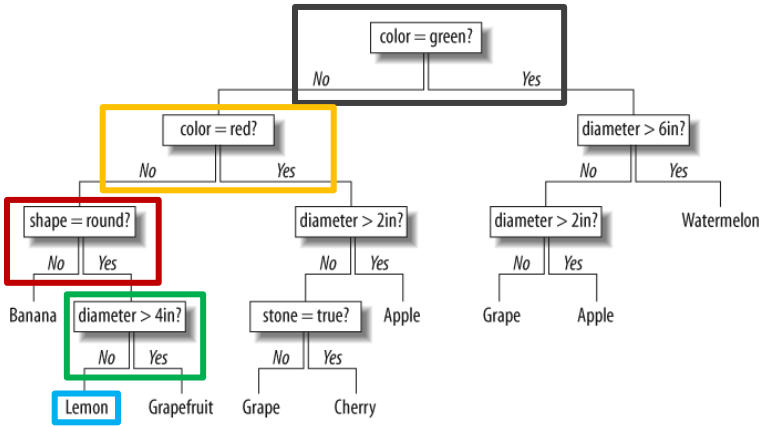
Image from scikit-learn,org

Regularization on features

- **Linear model:** $\hat{y}_i = b_0 + b_1 x_{i,1} + \cdots + b_n x_{i,n}$
  - LASSO

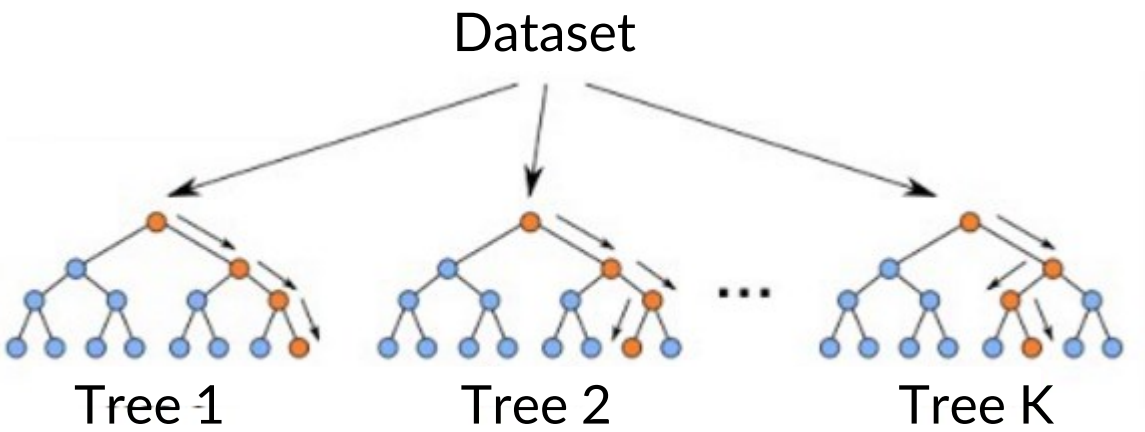
- **Tree model:**
  - Repeatedly using the same feature
  - Early decision affects the rest

- Feature bagging
  - Look at only *N* features at each step
  - Force model to use diverse features



Ensemble approaches

Bagging: Random forest



- Sample 80% of the dataset to train each decision tree
- Each tree may overfit to different part of the dataset
- But the consensus should be correct

Boosting

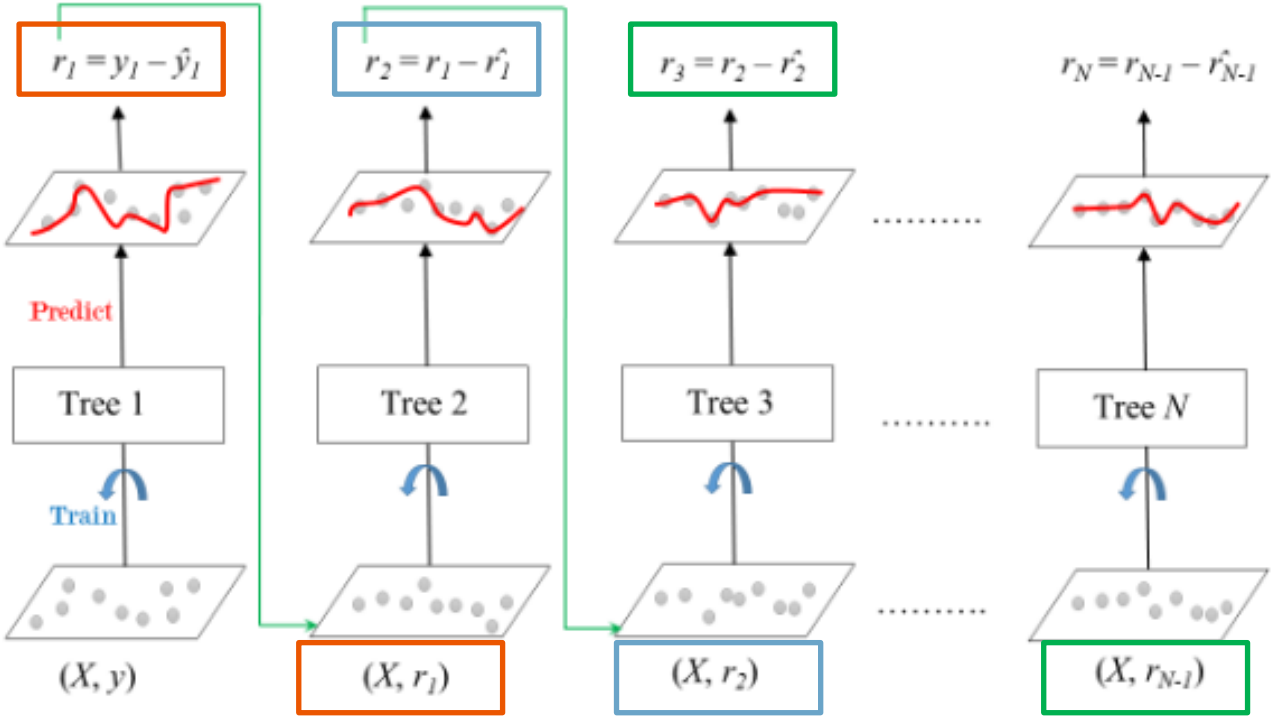
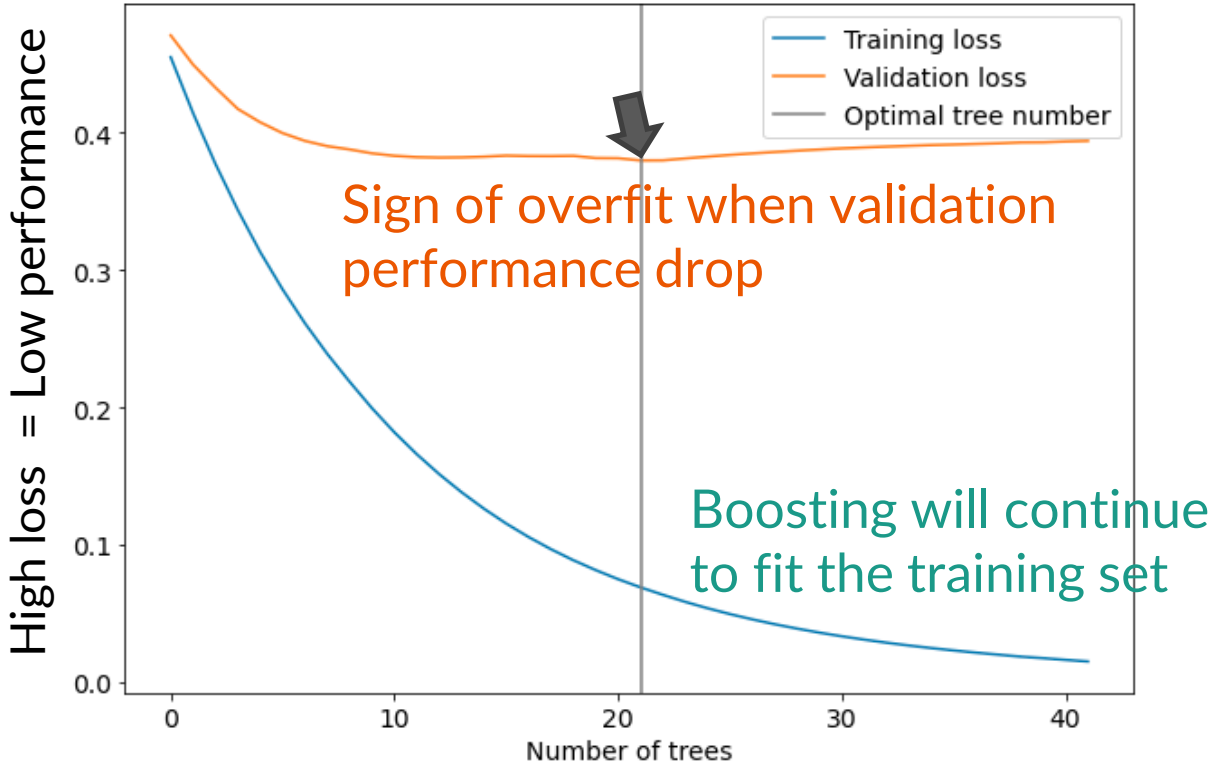


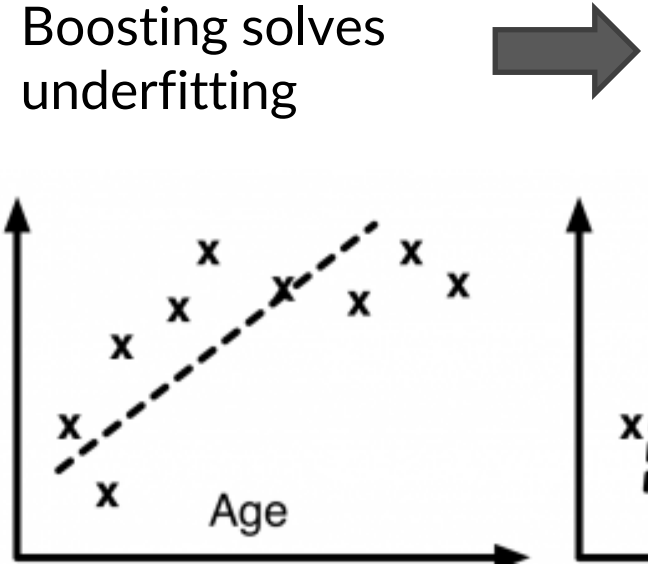
Image from geeksforgeeks.org

Controlling the boosting process



Impact of ensemble

Boosting solves underfitting

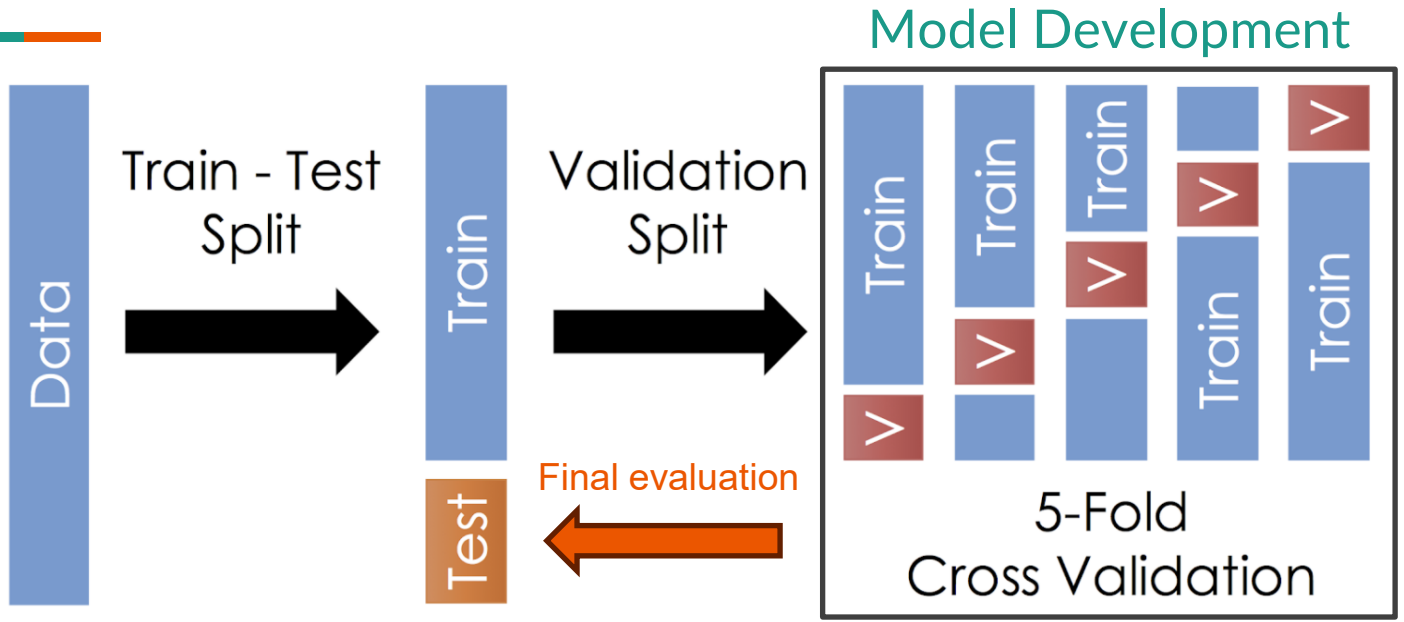


Bagging prevent overfitting

https://realkm.com/2018/04/23/optimization-and-complexity-the-cost-of-complexity-systems-thinking-modelling-series/

Model evaluation

Train-Val-Test

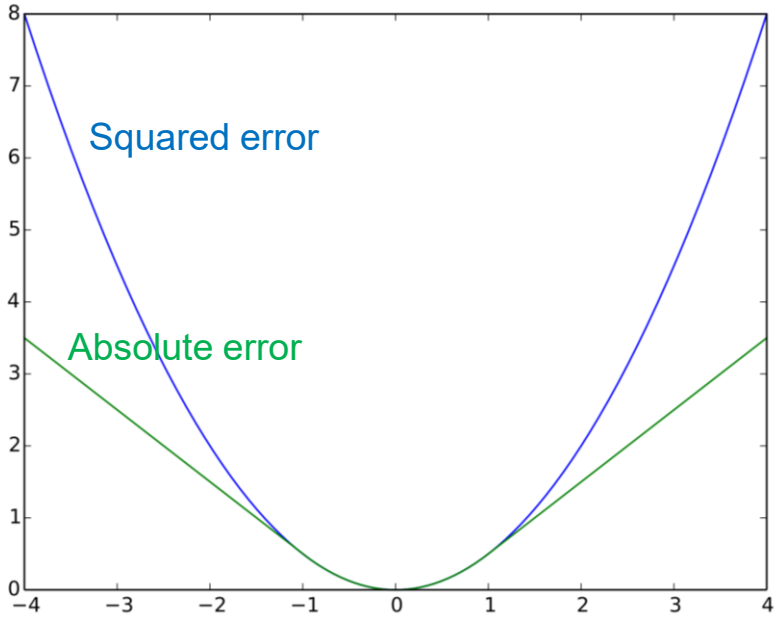


Source: medium.com

- **Training** data determines the best coefficients / weights
- **Validation** data determine the best hyperparameters
- **Test** data determine performance on new datasets

Performance metrics: regression

- Mean Square Error
- Mean Absolute Error
- Mean Absolute Percentage Error
- R^2 (Coefficient of Determination)



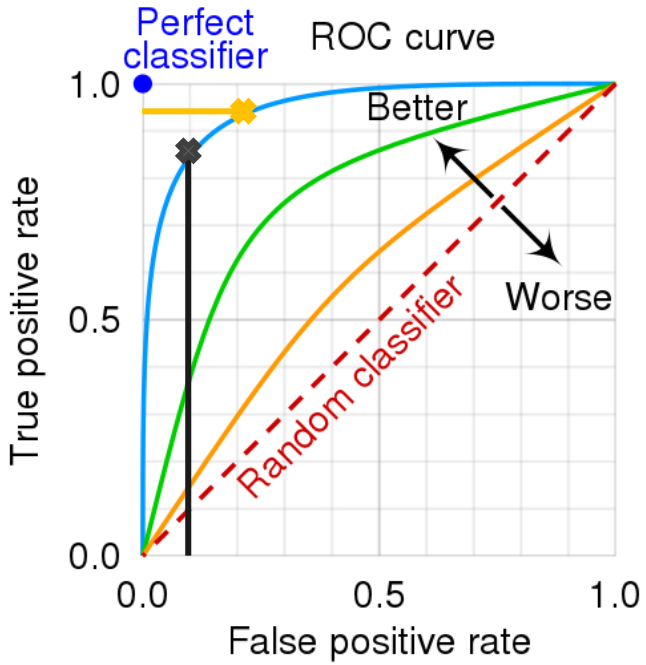
Performance metrics: classification

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive
		Predicted < 0.5	Predicted > 0.5

- Accuracy = (TN + TP) / total
- Precision = TP / (TP + FP) = Positive predictive value

- Recall = TP / (TP + FN) = Sensitivity
- Specificity = TN / (TN + FP)

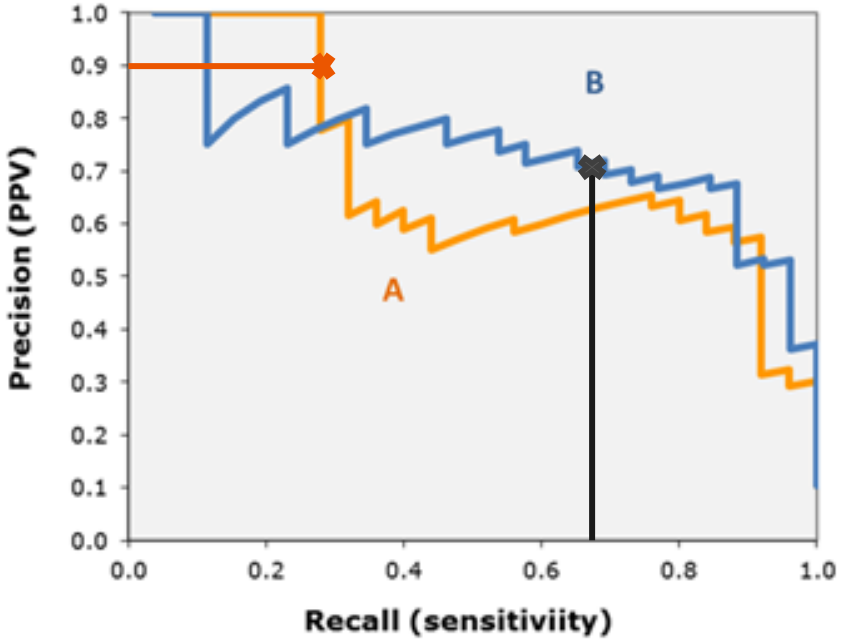
Threshold-free metrics



https://commons.wikimedia.org/wiki/File:Roc_curve.svg

- Sensitivity-specificity at every output threshold
- Area under the ROC curve (AUROC, AUC)
  - Random guess = 0.5
  - Perfect model = 1.0
- Pick threshold based on use case
  - Specificity >0.9
  - Sensitivity >0.9

Precision-Recall curve

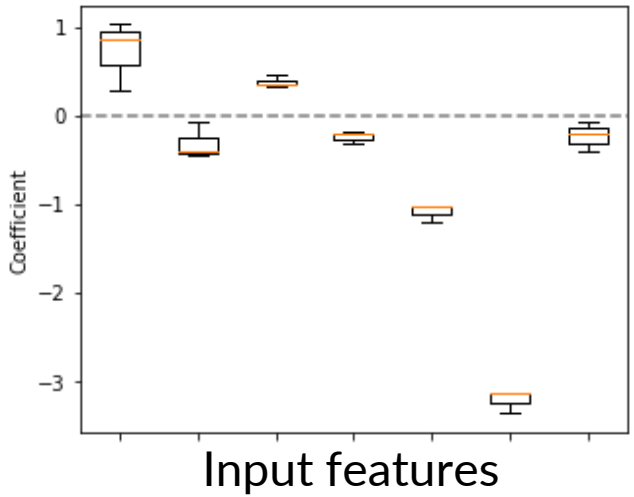


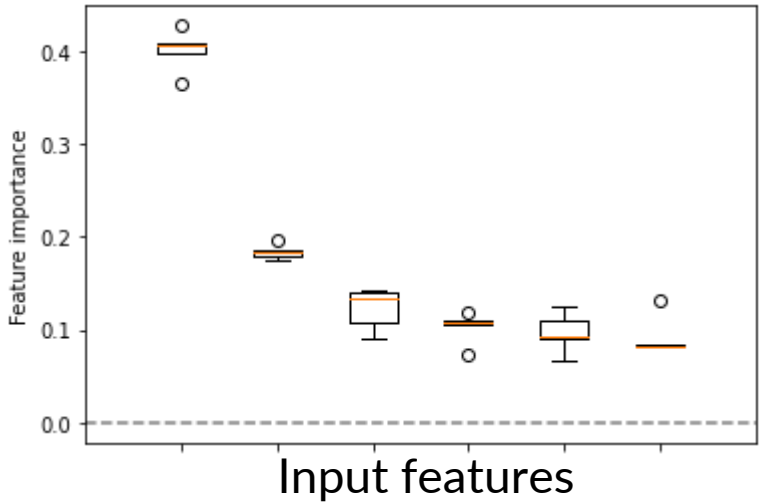
https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used

- The best model can depend on use case

Explainability

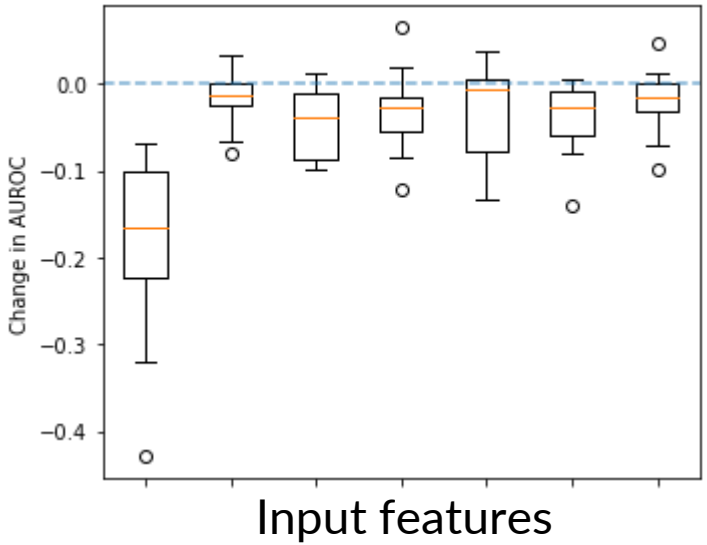
Feature importance





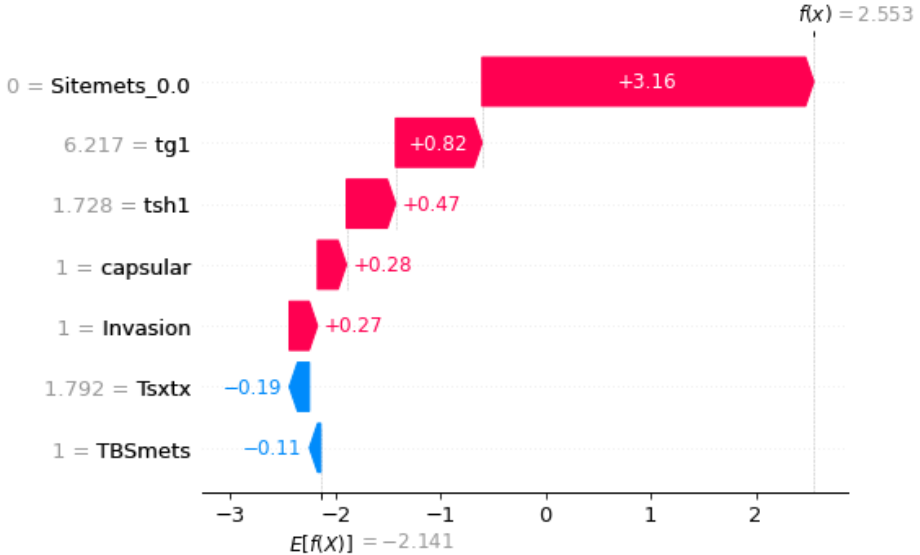
- Coefficients of linear, logistic, and SVM models
- Average improvement in impurity or entropy in tree models
- Model-level explanation

Change in performance after dropping a feature



- Compare performance with and without each input feature
- Big drop = important

Shapley value



- Change in predicted value due to the addition of a feature *i*
  - $\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!} [v(S \cup \{i\}) - v(S)]$
- Sample-level explanation

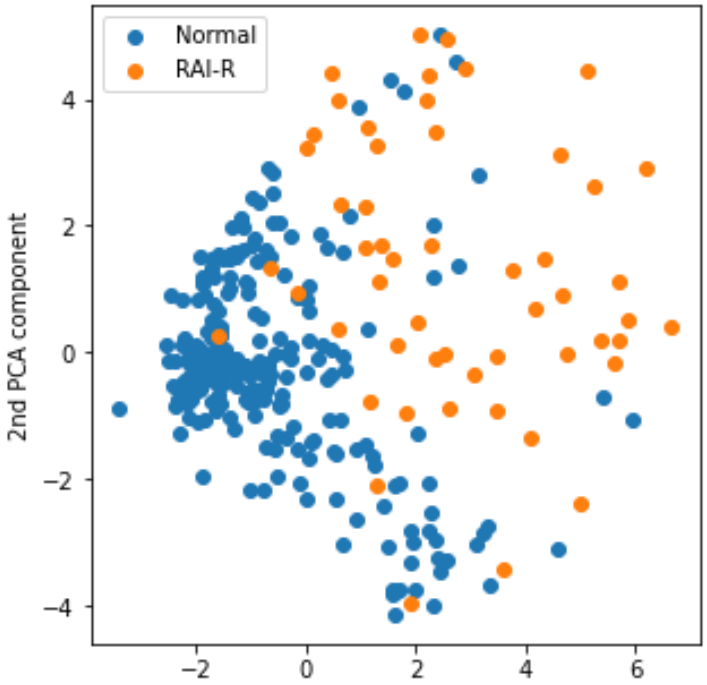
Key points

- Machine learning vs statistics = data driven vs hypothesis driven
- ML components = **model architecture** + **objective** + **learning algorithm**
- Heart of ML = balancing between data fitting and generalization
- Explainability ensures that the model learns the right knowledge

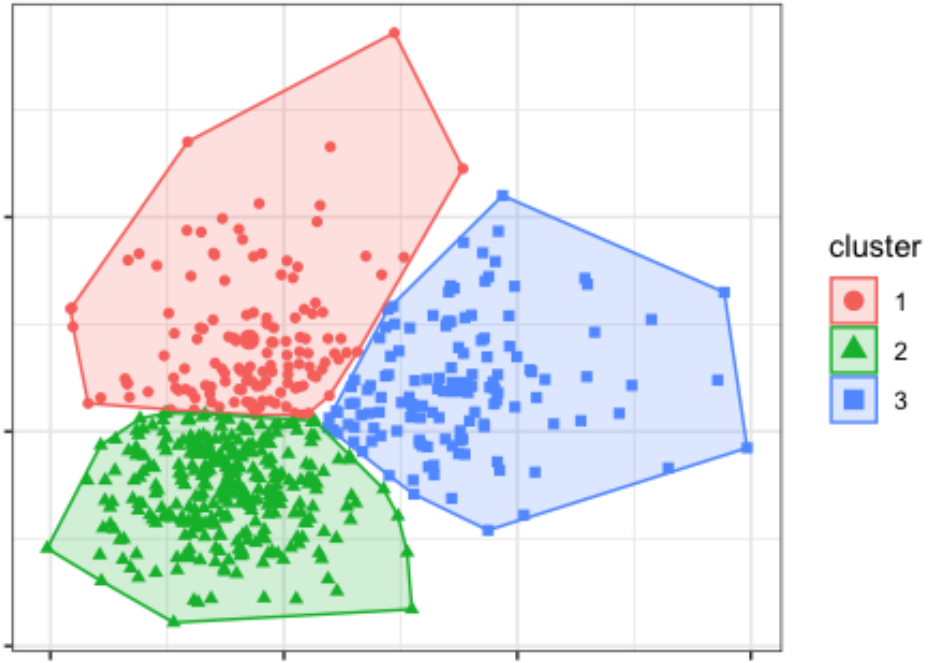
Unsupervised learning

Two primary branches of unsupervised learning

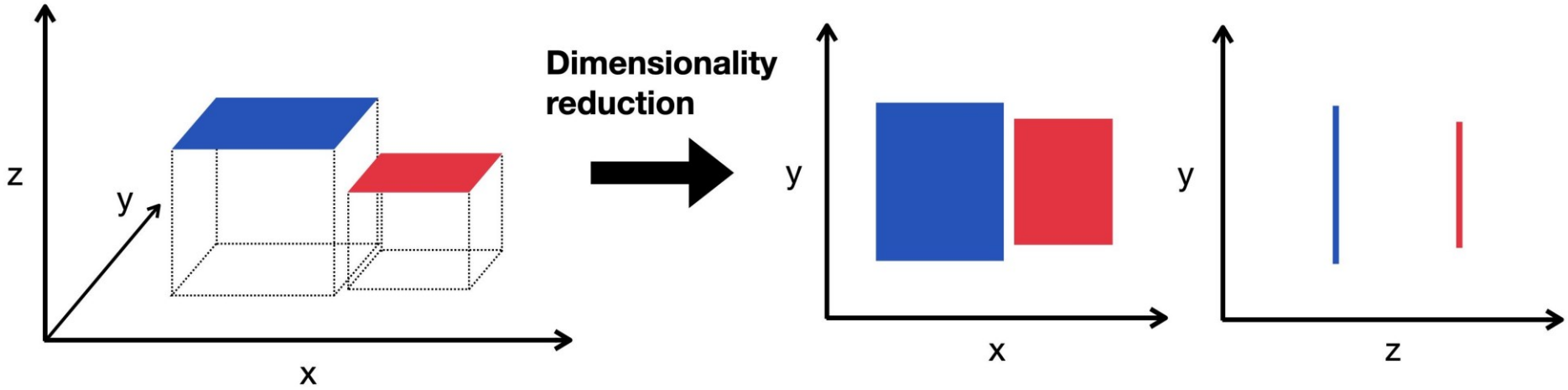
Dimensionality Reduction



Clustering



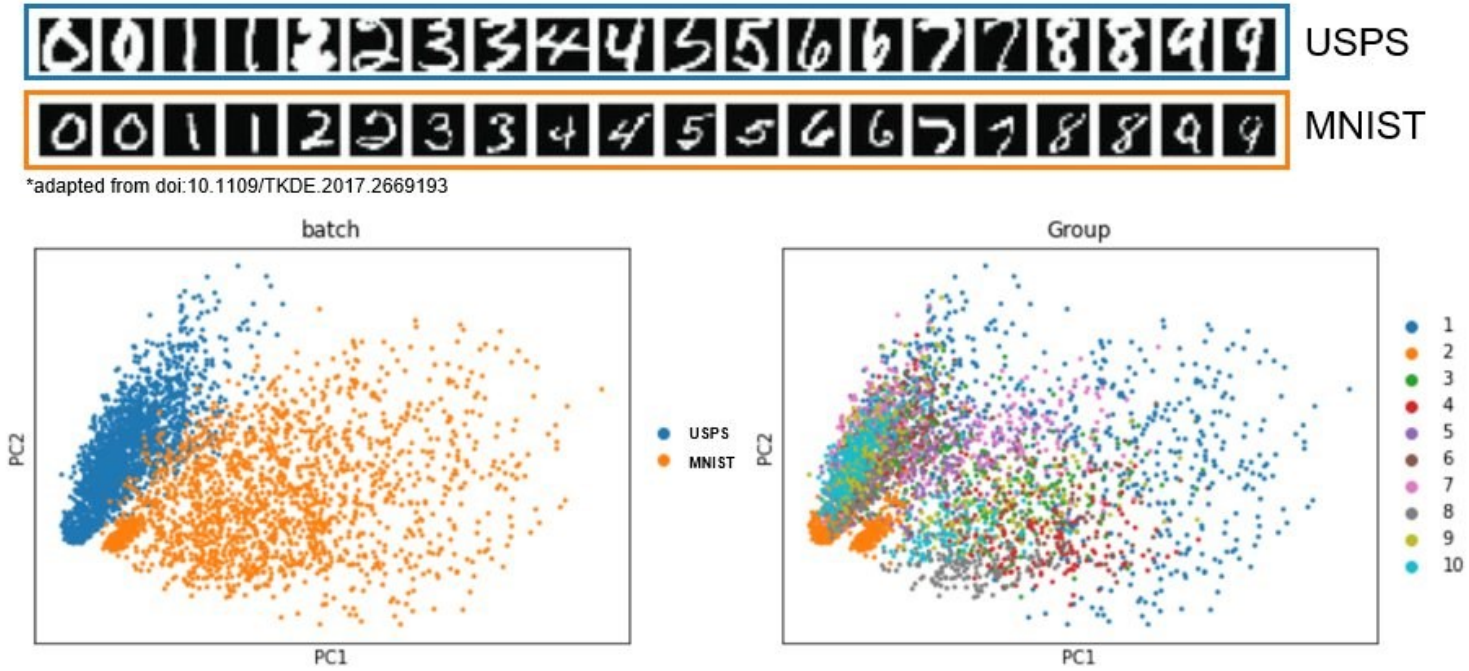
Dimensionality reduction



https://www.sc-best-practices.org/preprocessing_visualization/dimensionality_reduction.html

- Reduce dimension (number of features) while maintaining information
- Patient with similar symptoms also exhibit similar lab tests or have similar demographics or similar medical history

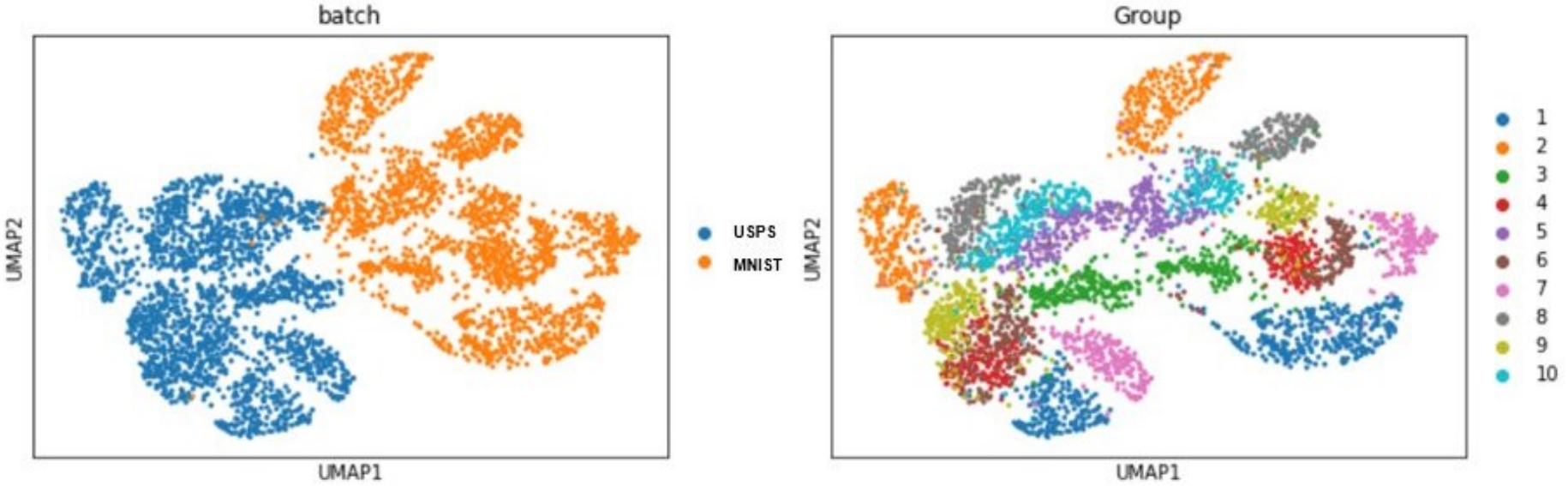
Digit datasets



*adapted from doi:10.1109/TKDE.2017.2669193

https://twitter.com/lkmklsmn/status/1436357177887895555

UMAP captures every group in 2D

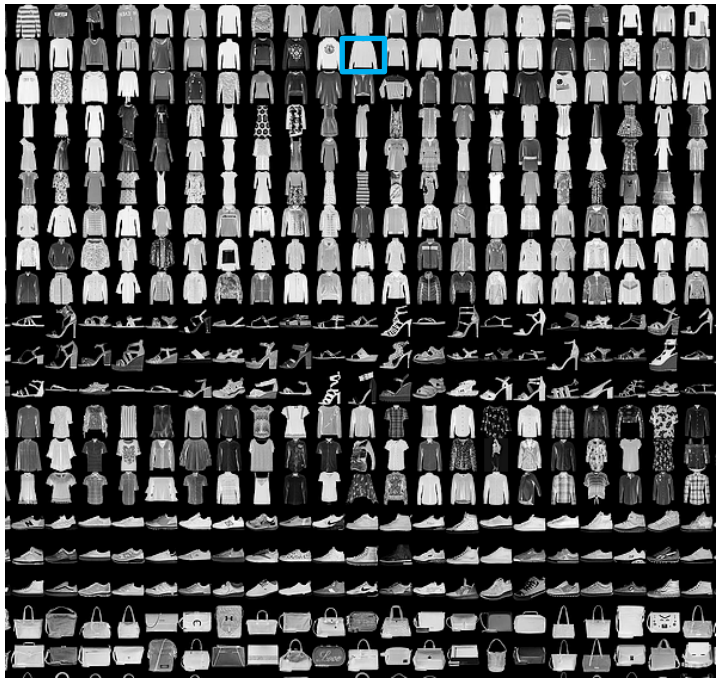


https://twitter.com/lkmklsmn/status/1436357177887895555

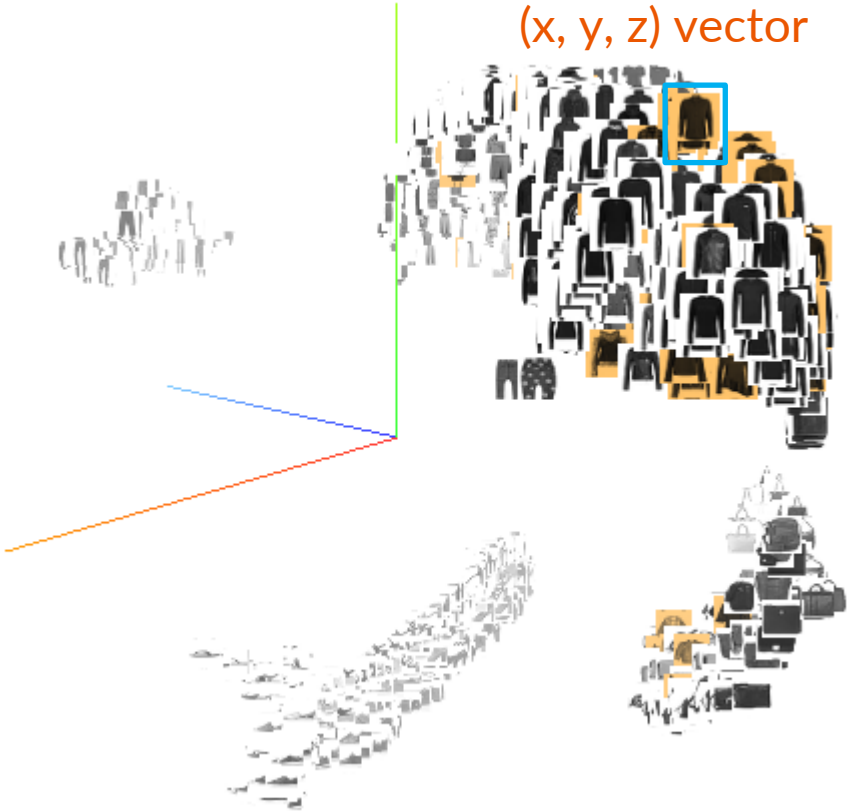
- Both data source and digit identity can be distinguished

Fashion MNIST

28x28 pixel image



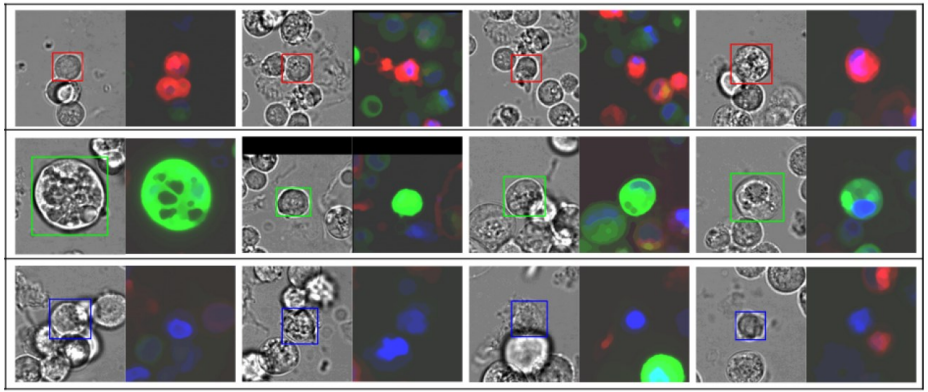
(x, y, z) vector

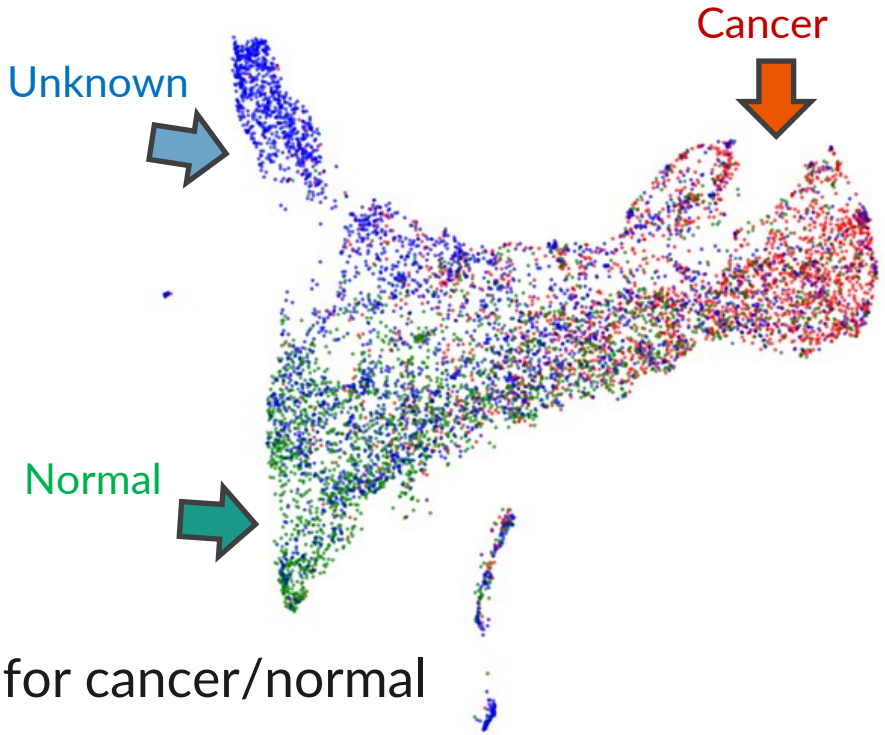


https://github.com/zalandoresearch/fashion-mnist

https://pair-code.github.io/understanding-umap/

2D visualization for cell images

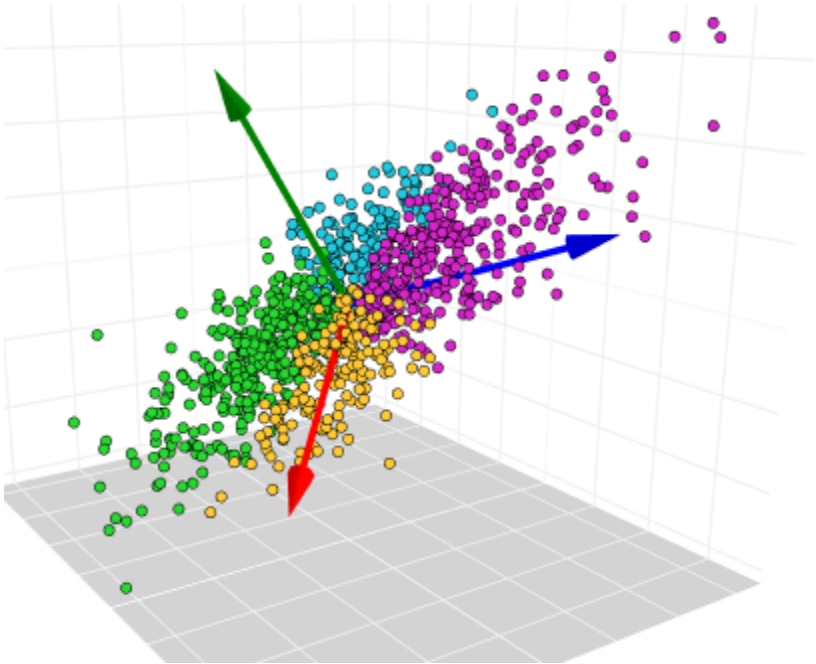




- Extracted from deep learning model for cancer/normal cell type classification

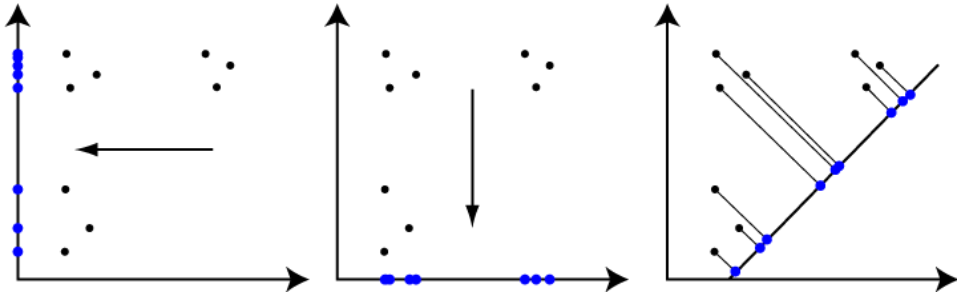
Principal component analysis (PCA)

Variance is information



https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d

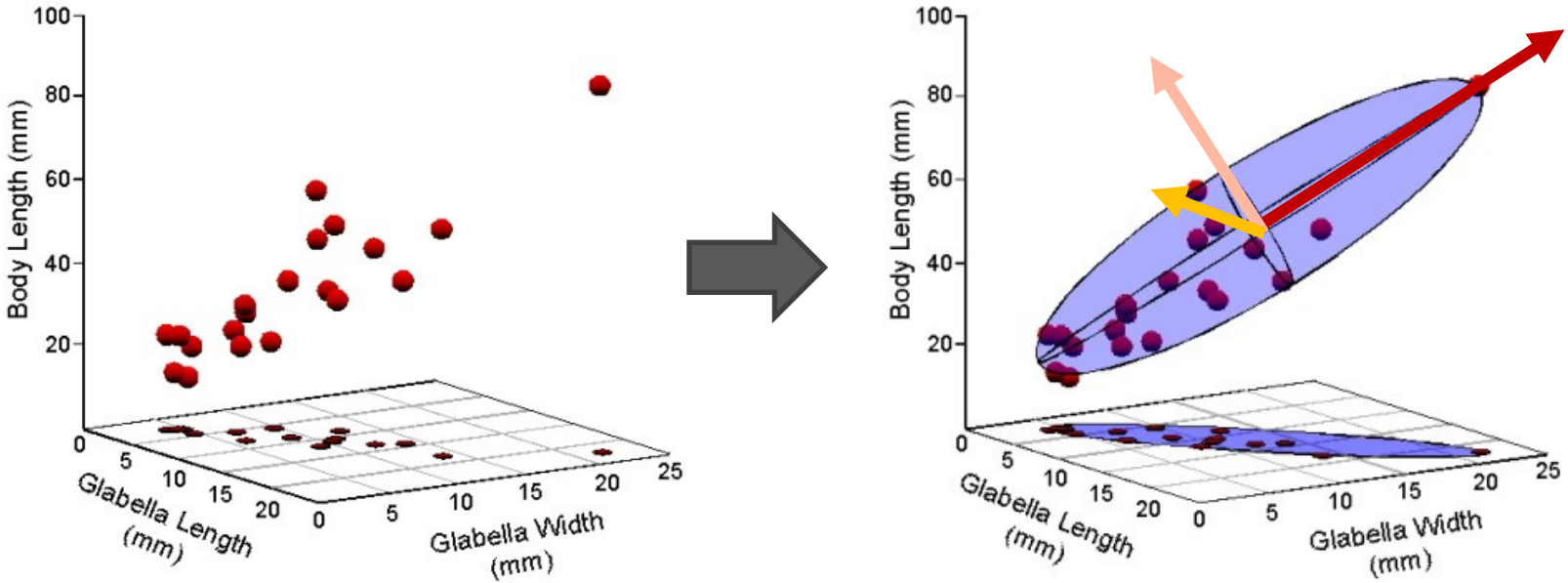
Projection



https://shapeofdata.wordpress.com/2013/04/16/visualization-and-projection/

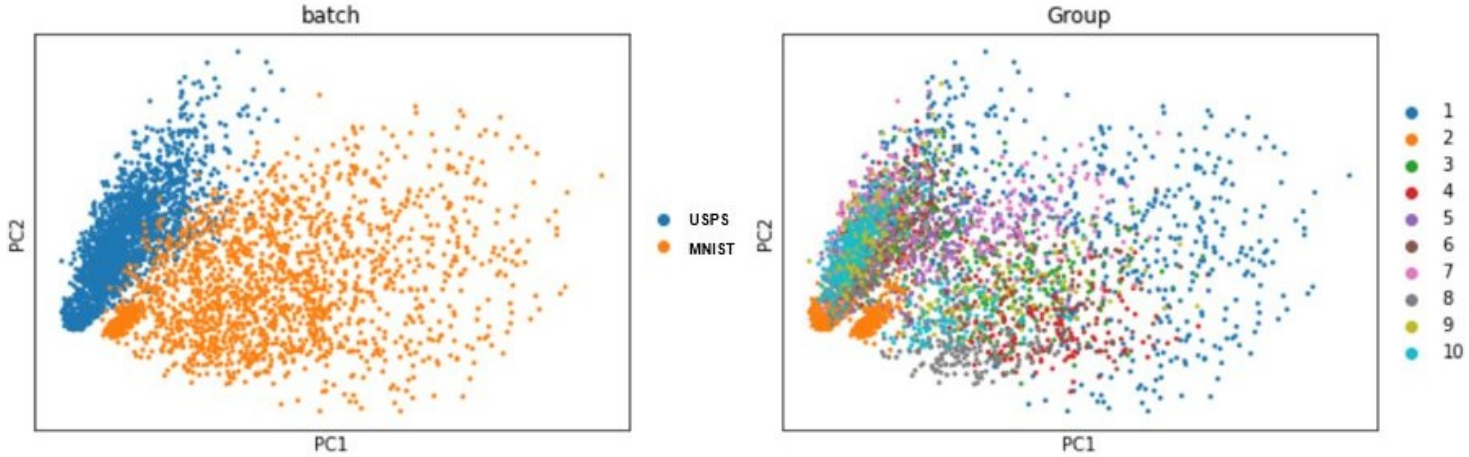
- High variances = more power to distinguish groups of data points

PCA prioritizes directions with high variances



Source: the paleontological association

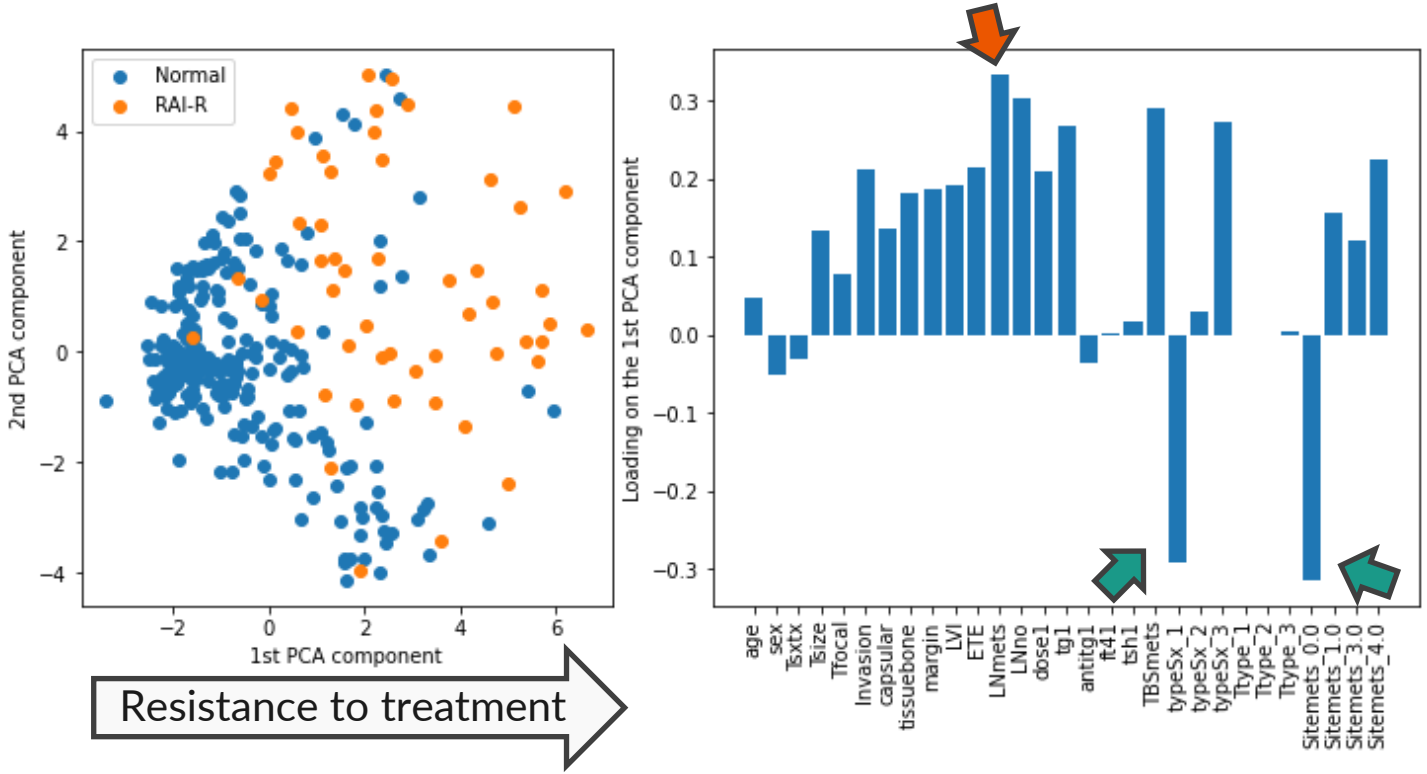
Interpretation of PCA result



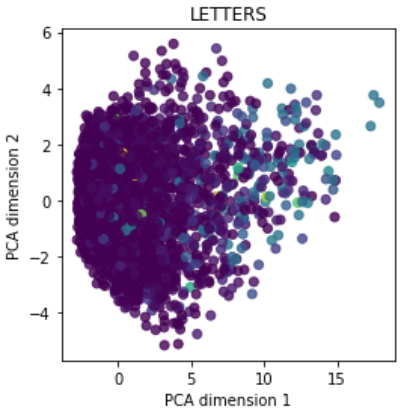
https://twitter.com/lkmklsmn/status/1436357177887895555

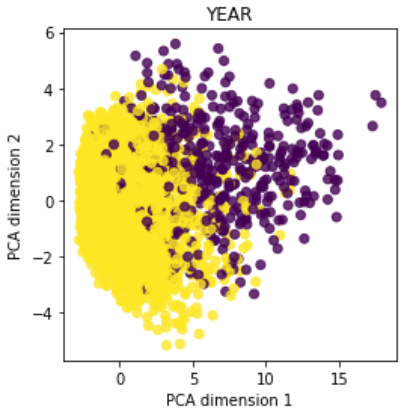
- PC1 captures the variance between data sources
- PC2 somewhat captures the variance between digit identity

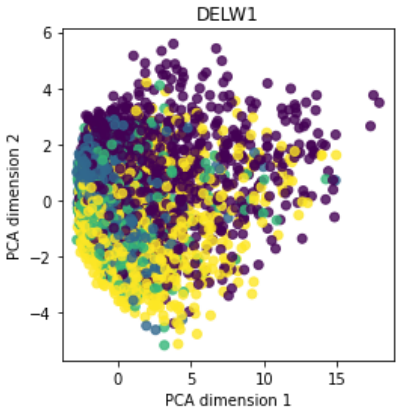
Interpreting loadings on individual PC

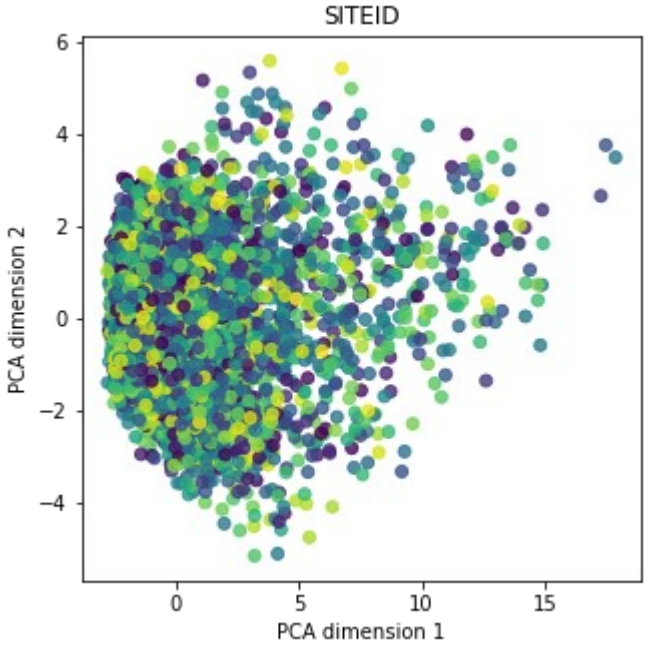


Exploring PCA results



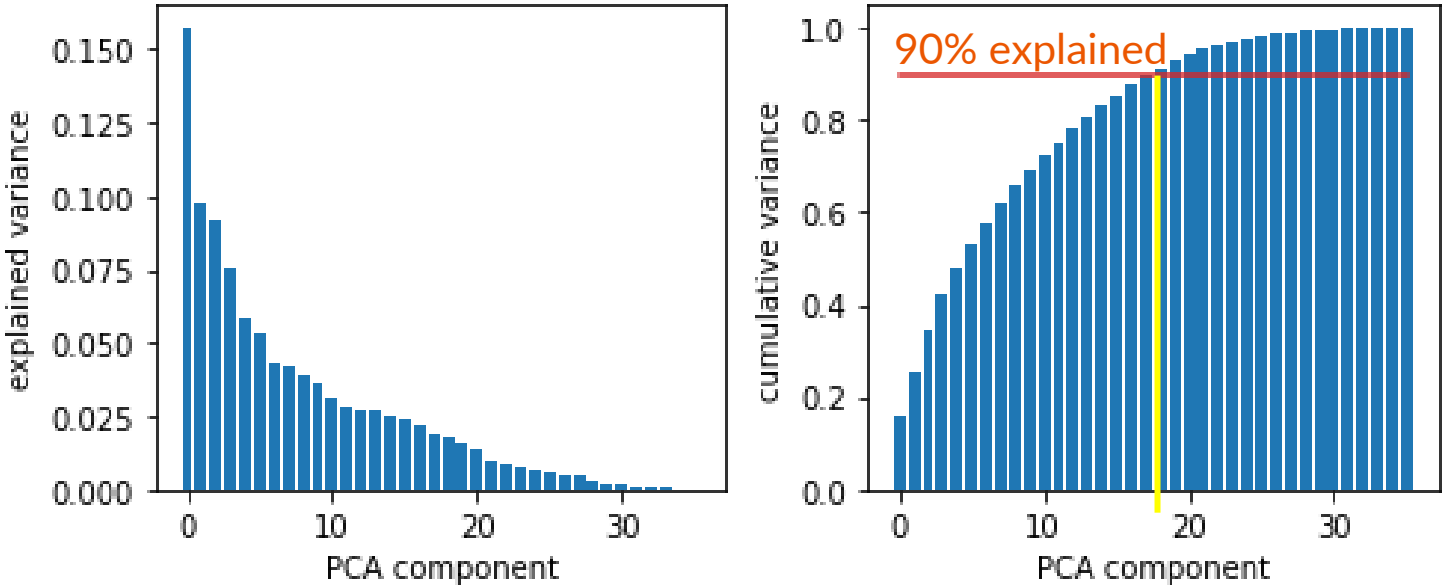






- Color by feature values to understand how PCA group data points
- Color by potential confounding factors

PCA for dimensionality reduction

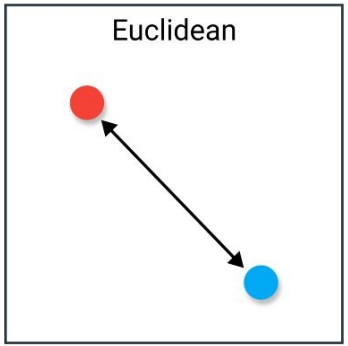


- By default, PCA retains the number of dimensions
- We can select only the first *k* PC for downstream analyses

Pros and cons of PCA

- Each PC can be interpreted from the loadings
- Highly correlated features tend to be grouped into the same PC
- PCA is a good initial dimensionality reduction step

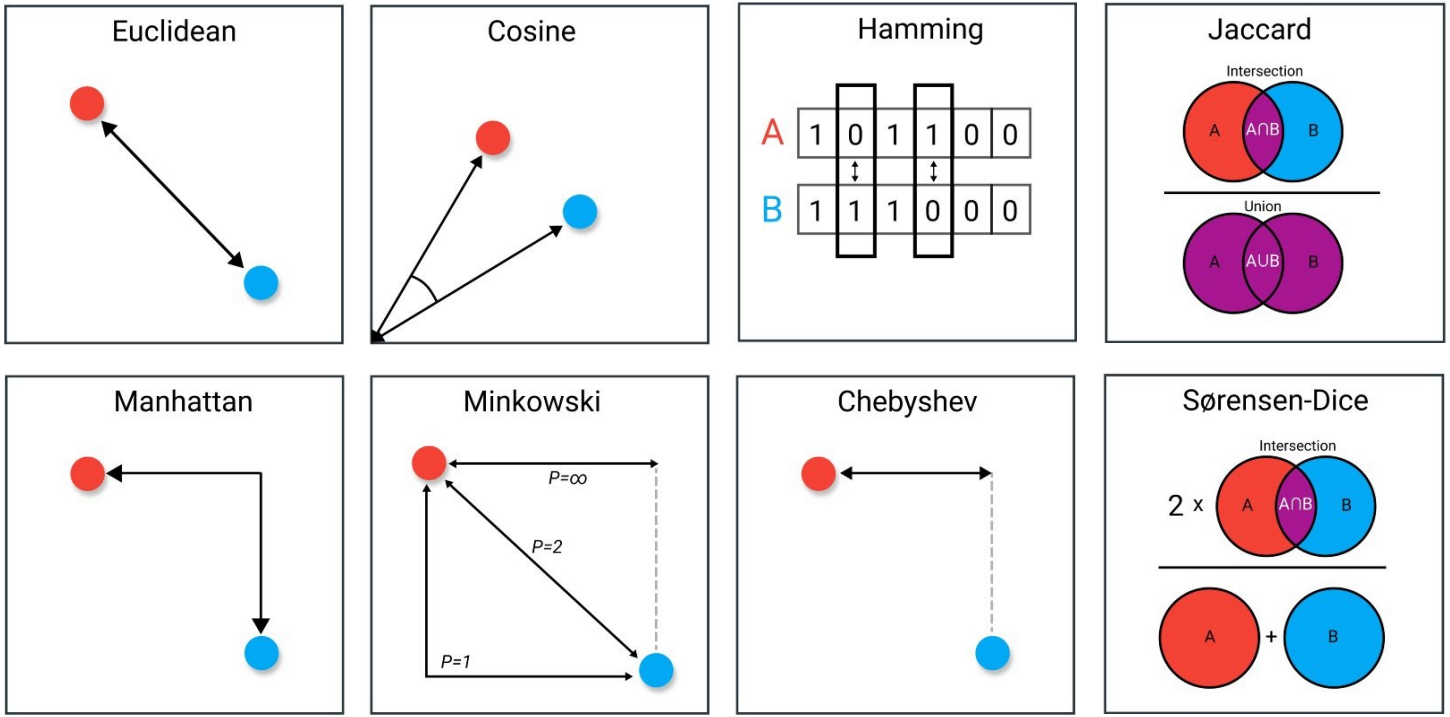
- PCA strictly preserves Euclidean distance
  - But some datasets require different distance metric!



https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa

Multidimensional Scaling (MDS)

Distances



https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa

Pairwise distance matrix

	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

http://www.slimsuite.unsw.edu.au/teaching/upgma/

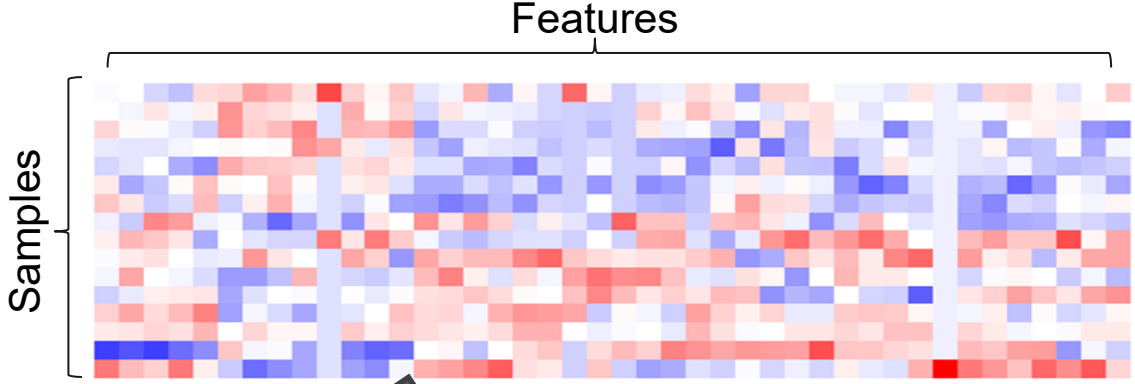
Metric properties

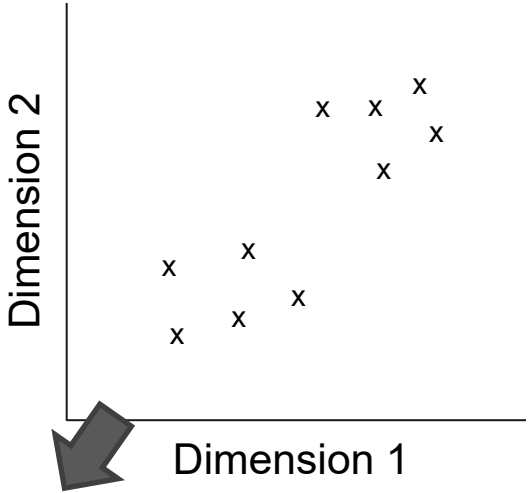
- D(*i*, *j*) = distance between sample *i* and sample *j*
- D(*i*, *i*) = 0
- D(*i*, *j*) = 0 iff *i* = *j*
- D(*i*, *j*) = D(*j*, *i*)
- D(*i*, *j*) + D(*j*, *k*) ≥ D(*i*, *k*)

Non-metric

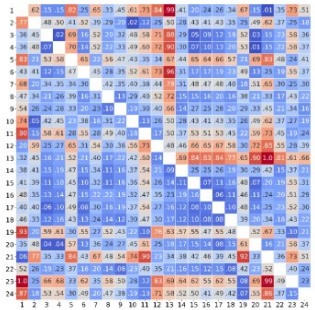
- Any user-defined dissimilarity
- D(*i*, *j*) != D(*j*, *i*)

Principal Coordinate Analysis (PCoA)

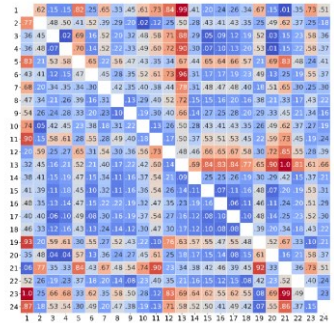




Non-Euclidean distance

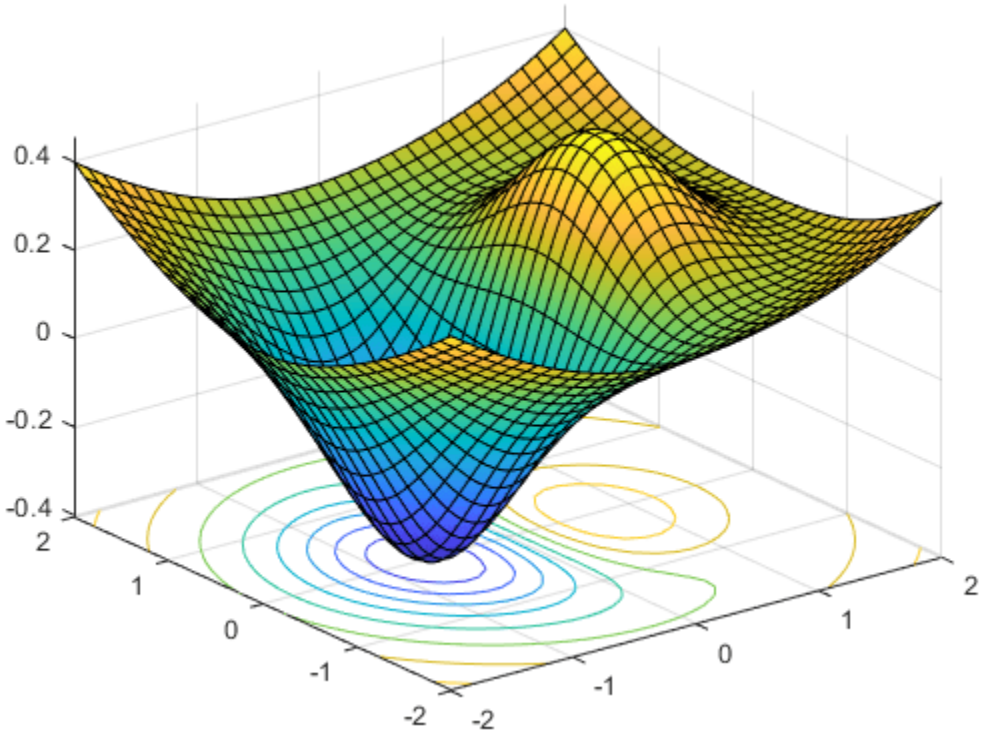


Maximize correlation



Euclidean distance

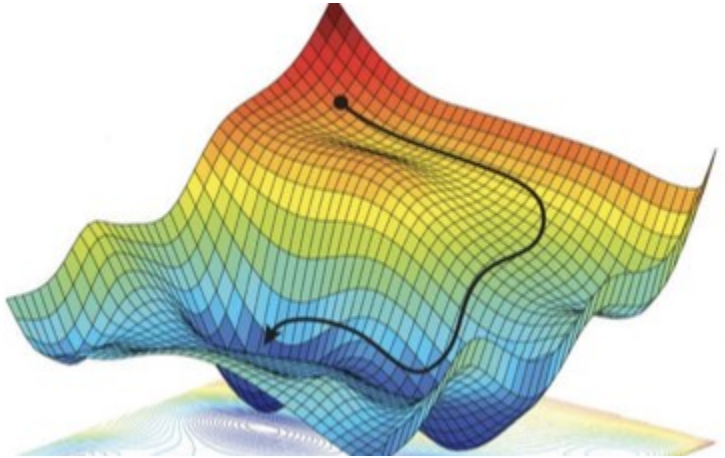
How to optimize a function?



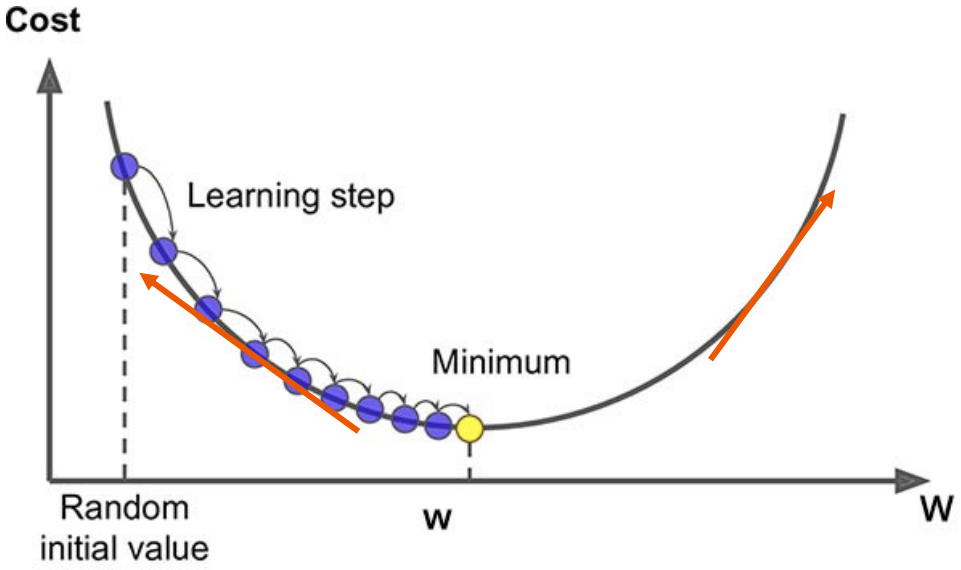
https://es.mathworks.com/help/optim/ug/optimization-toolbox-tutorial.html

- Find (x_1, x_2, ..., x_n) that minimize $f(x_1, x_2, ..., x_n)$
- At minimum, the slope is zero in all directions
- Take derivative of each variable and set to zero
  - $\frac{\delta f}{\delta x_1} = 0$
  - $\frac{\delta f}{\delta x_2} = 0$
  - *n* equations with *n* variables

Gradient descent

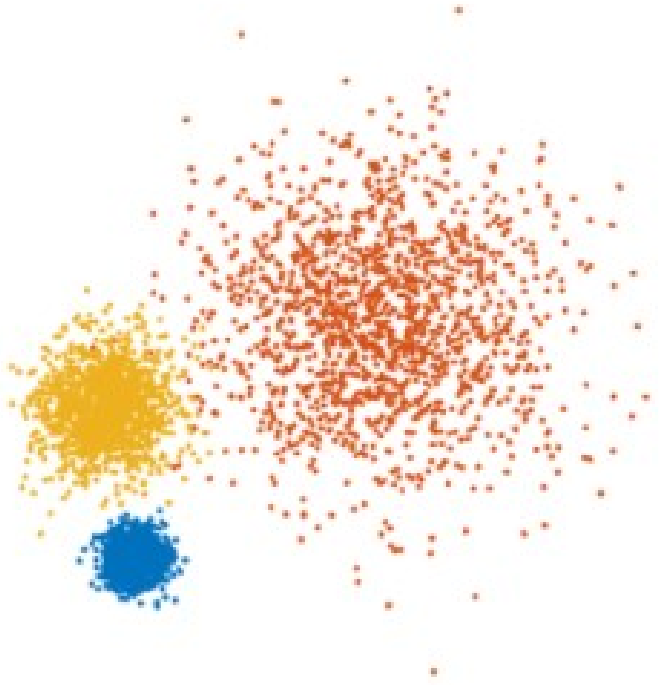


https://medium.com/analytics-vidhya/gradient-descent-b0dc1af33517



- Slope tells us if the function is increasing or decreasing if we increase x_i
  - So, we can update x_i accordingly

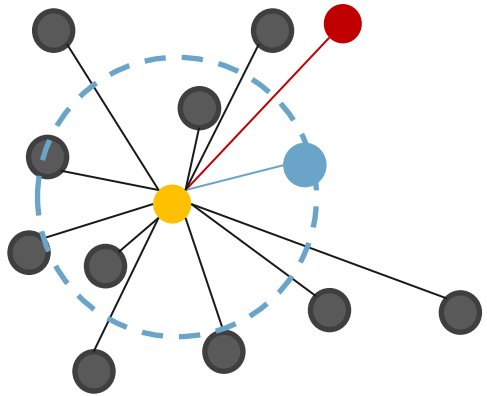
Limitation of PCA and MDS

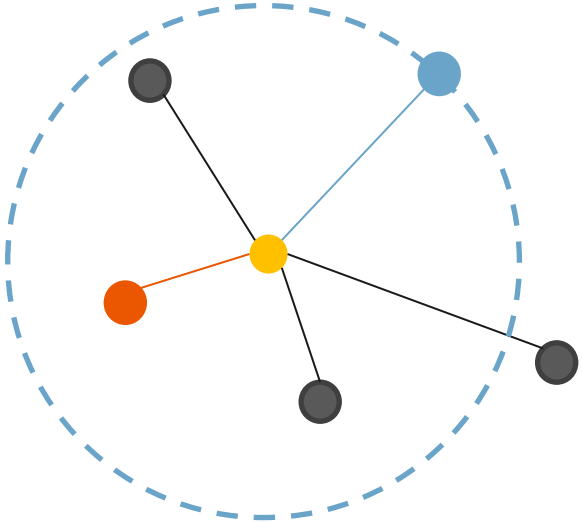


- A single definition of distance is used throughout the data space
- What if some data groups are noisier than the others?
  - Difference in data density

t-distributed stochastic neighbor embedding (*t*-SNE)

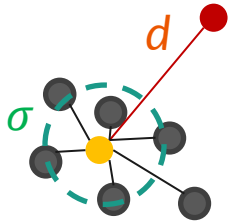
Measuring data density





- Distance to the *k*-th nearest neighbor reflects data density
  - Small distance in dense area
  - Large distance in sparse area

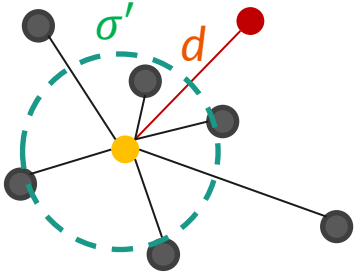
Probability of being a neighbor



score(o | o) = probability that o would pick o as neighbor under a **normal distribution** center at o

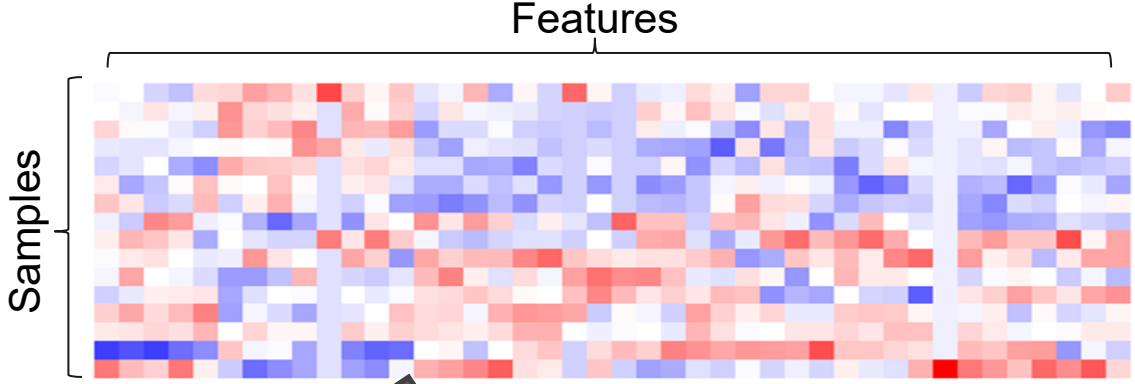
$$= \frac{e^{-\frac{d^2}{2\sigma^2}}/\sigma}{\sum e^{-\frac{(\text{dist}(o,\,o))^2}{2\sigma^2}}/\sigma}$$

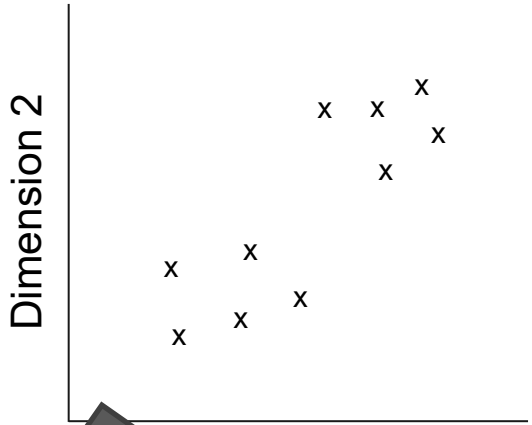
o = other data points



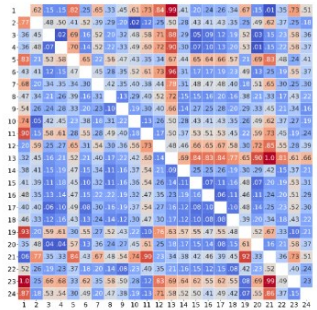
- Same distance d normalized against density σ and distances to other nearby data points o

Finding the optimal projection for *t*-SNE

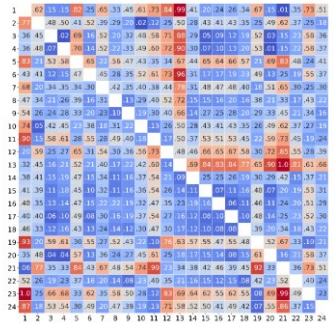




Probability of being a neighbor (Normal) (σ depend on density)



Maximize similarity



Probability of being a neighbor (*t*-distribution) (σ = 1)

Why *t*-distribution for the projection?

- *t*-distribution has fatter tails
- Allow data points to be projected far away from each other

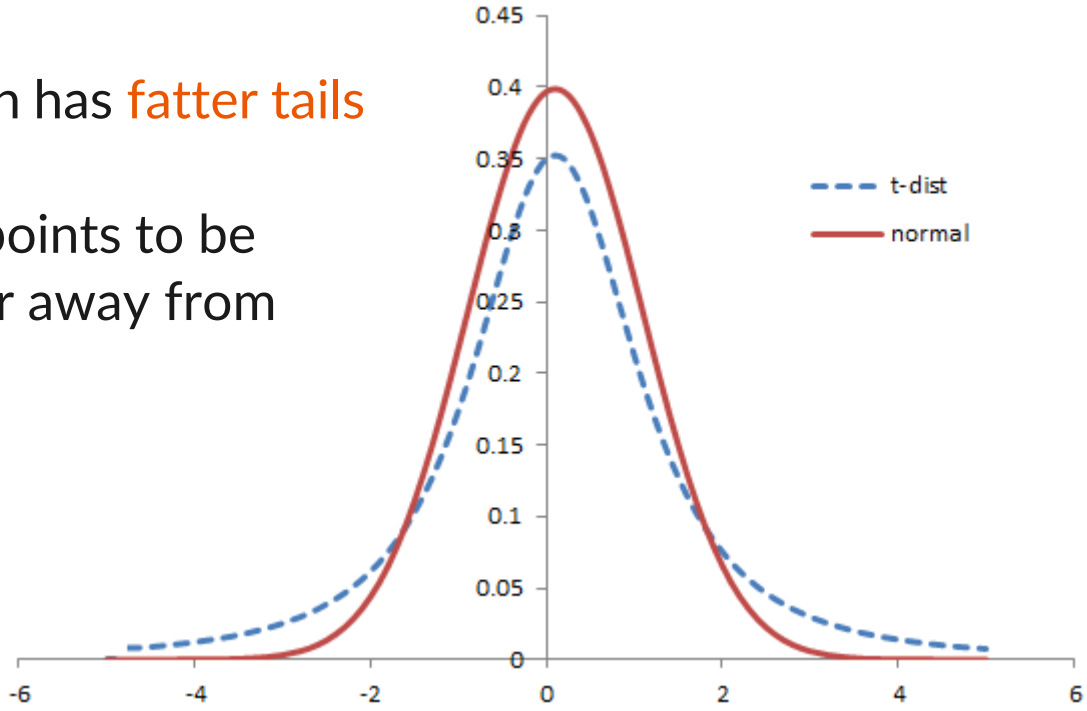
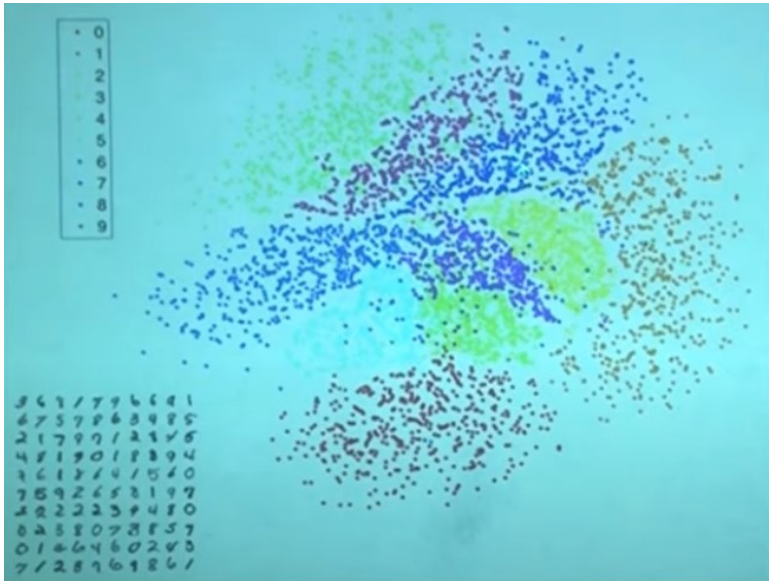
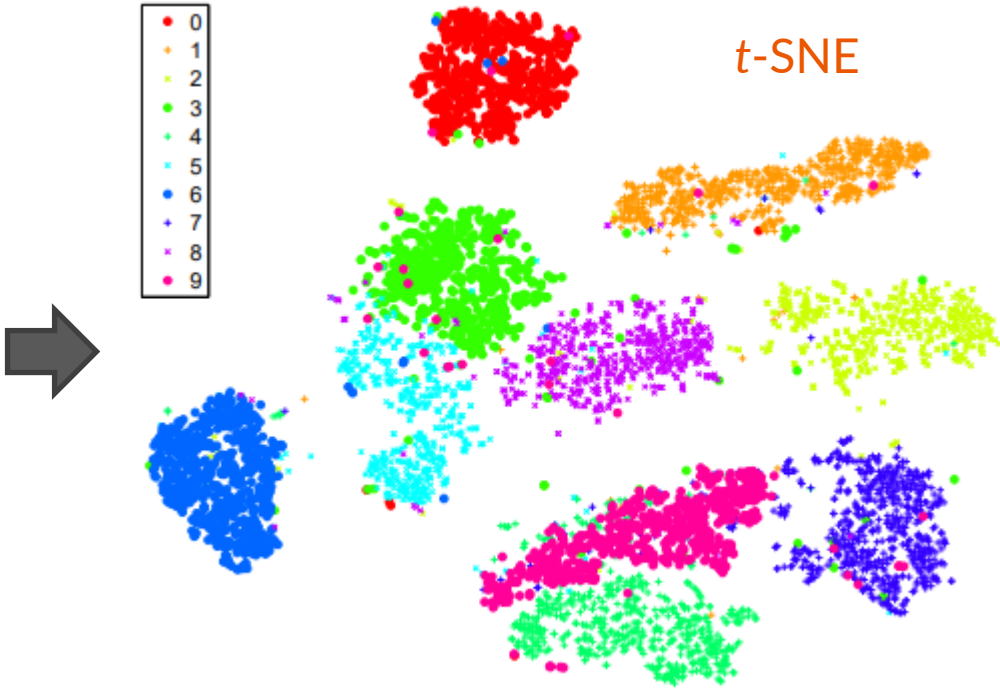


Image from riskprep.com

Impact of *t*-distribution

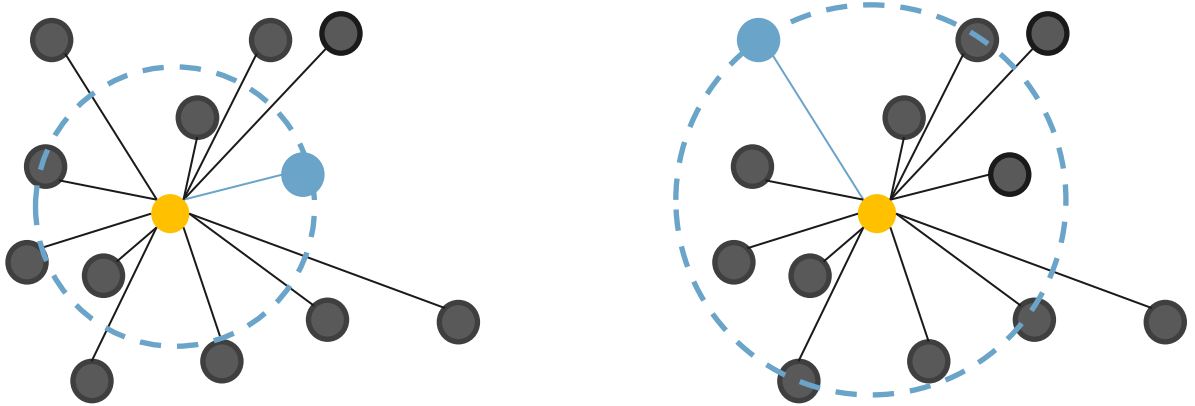


SNE (Normal → Normal)



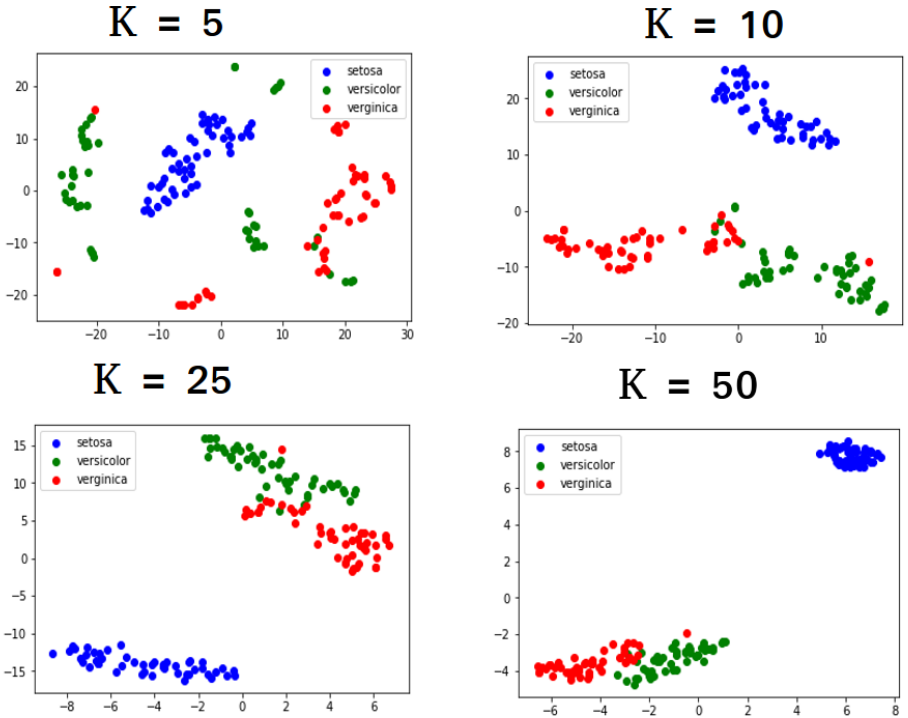
Maaten, L. and Hinton, G. J of Machine Learning Research 9:2579-2605 (2008)

Perplexity



- How many nearest neighbors to consider to normalize data density?
  - Perplexity parameters

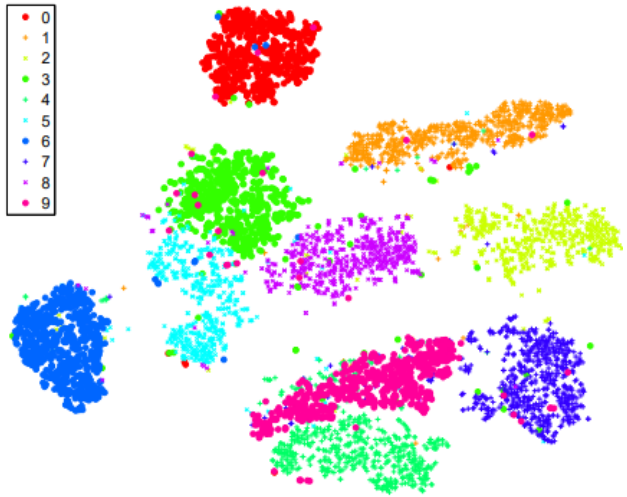
Impact of perplexity



- Too small perplexity = a lot of scatted data groups
- Try varying the perplexity and identify patterns that consistently appear

Source: blog.paperspace.com/dimension-reduction-with-t-sne/

Pros and cons of *t*-SNE



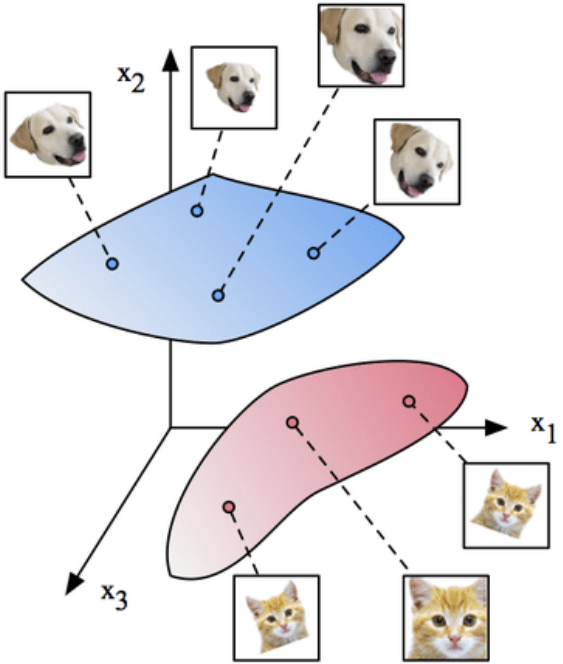
- Capture neighbor relationship
- Normalize data density

- Recompute every time new data is added
- Lose long-range relationship
- Axes of the resulting projection have no meaning
  - Don't use *t*-SNE coordinates for clustering or interpretation

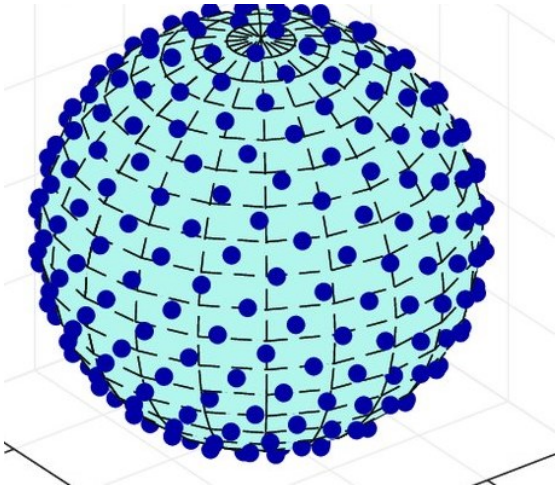
Maaten, L. and Hinton, G. J of Machine Learning Research 9:2579-2605 (2008)

Uniform manifold approximation and projection (UMAP)

Two key assumptions



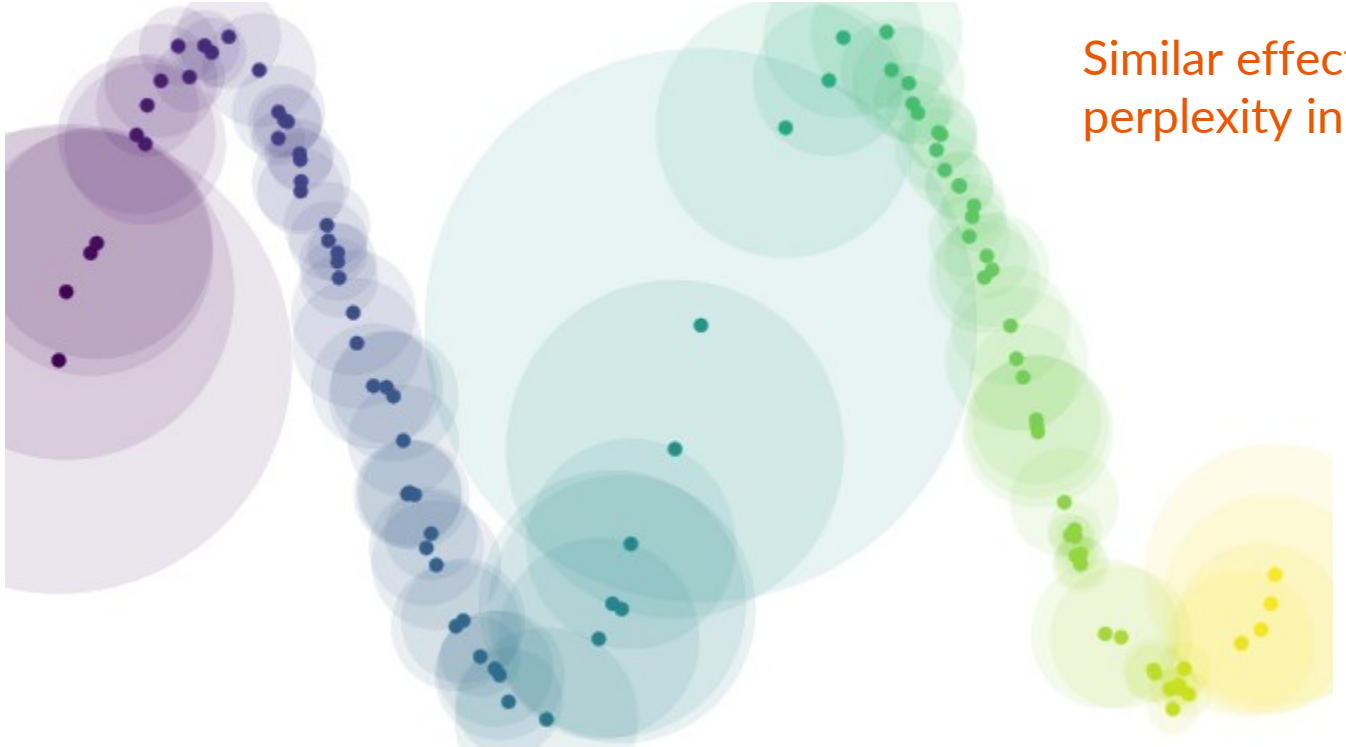
Chung, S. et al. “Classification and Geometry of General Perceptual Manifolds”



Ali, A. et al. IEEE Access PP(99):1 (2021)

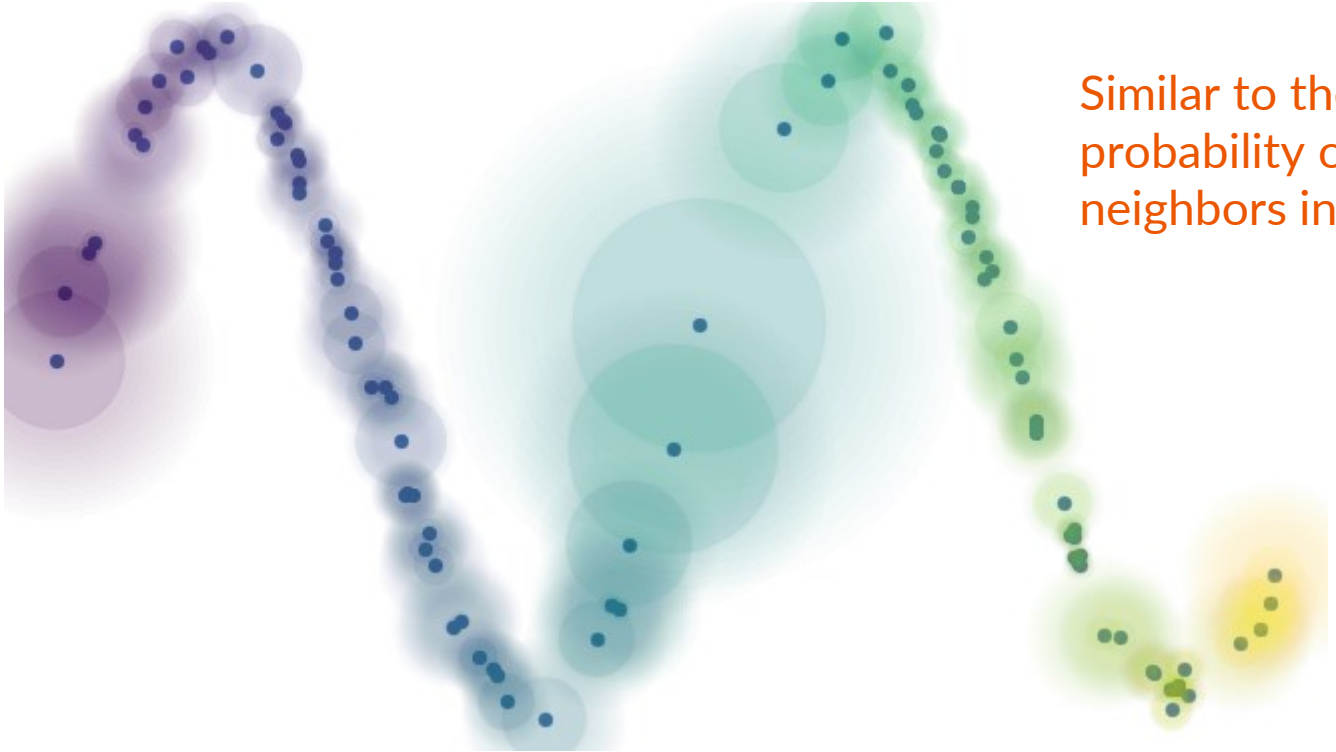
- Data came from multiple manifolds
- Data points were sampled uniformly

Uniform sampling = similar distance to *k*-th neighbor



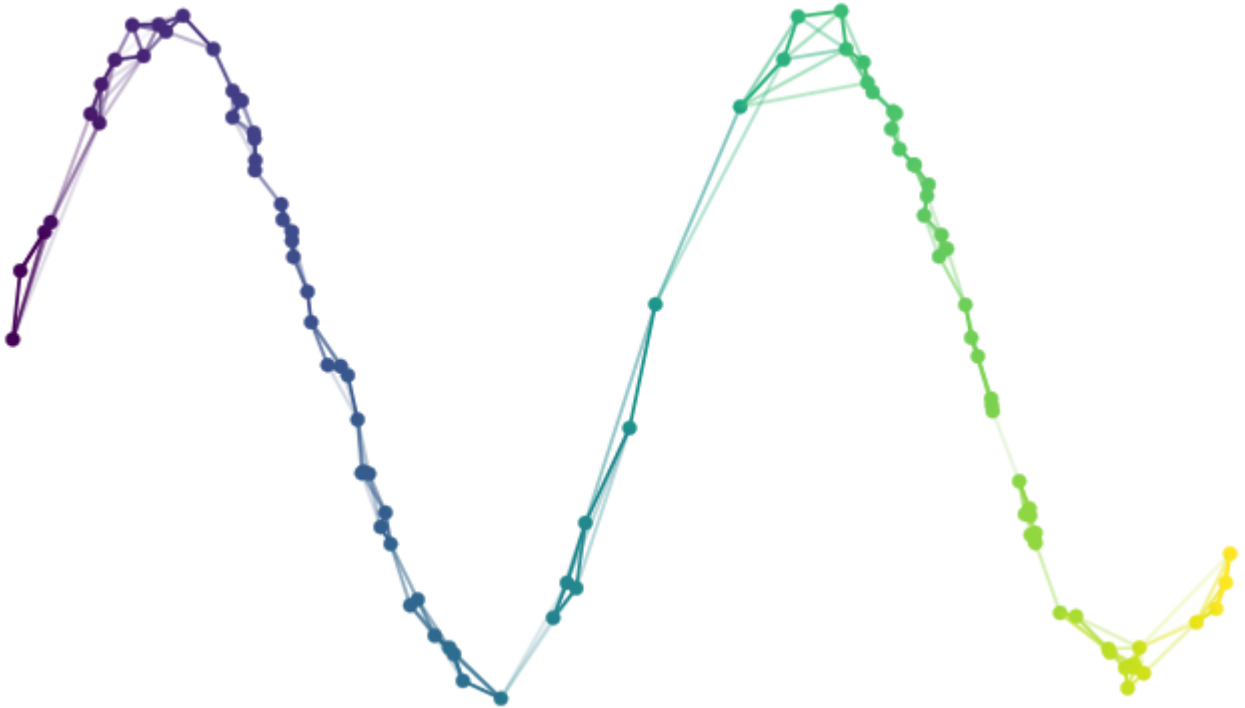
Similar effect as perplexity in t-SNE

Adding uncertainty between faraway data points

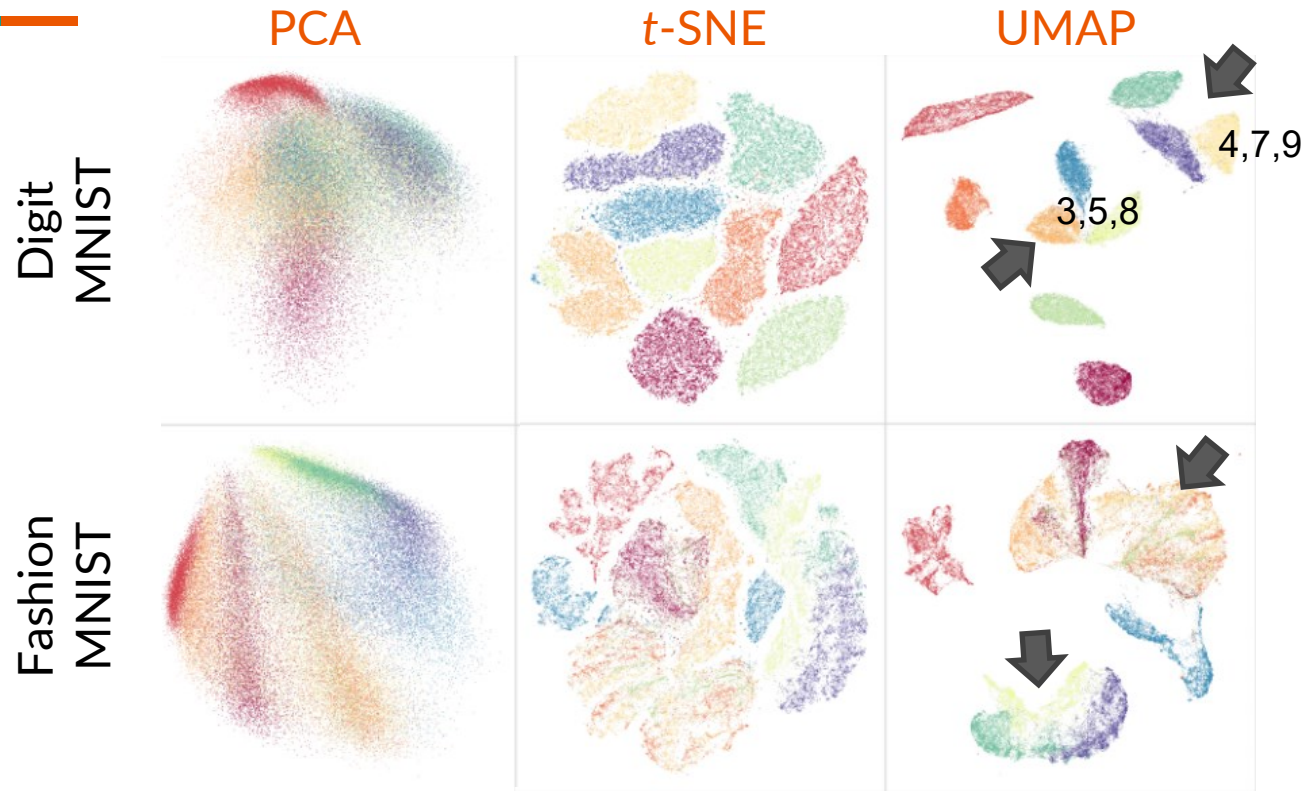


Similar to the probability of being neighbors in *t*-SNE

Network representation of neighbor relationship

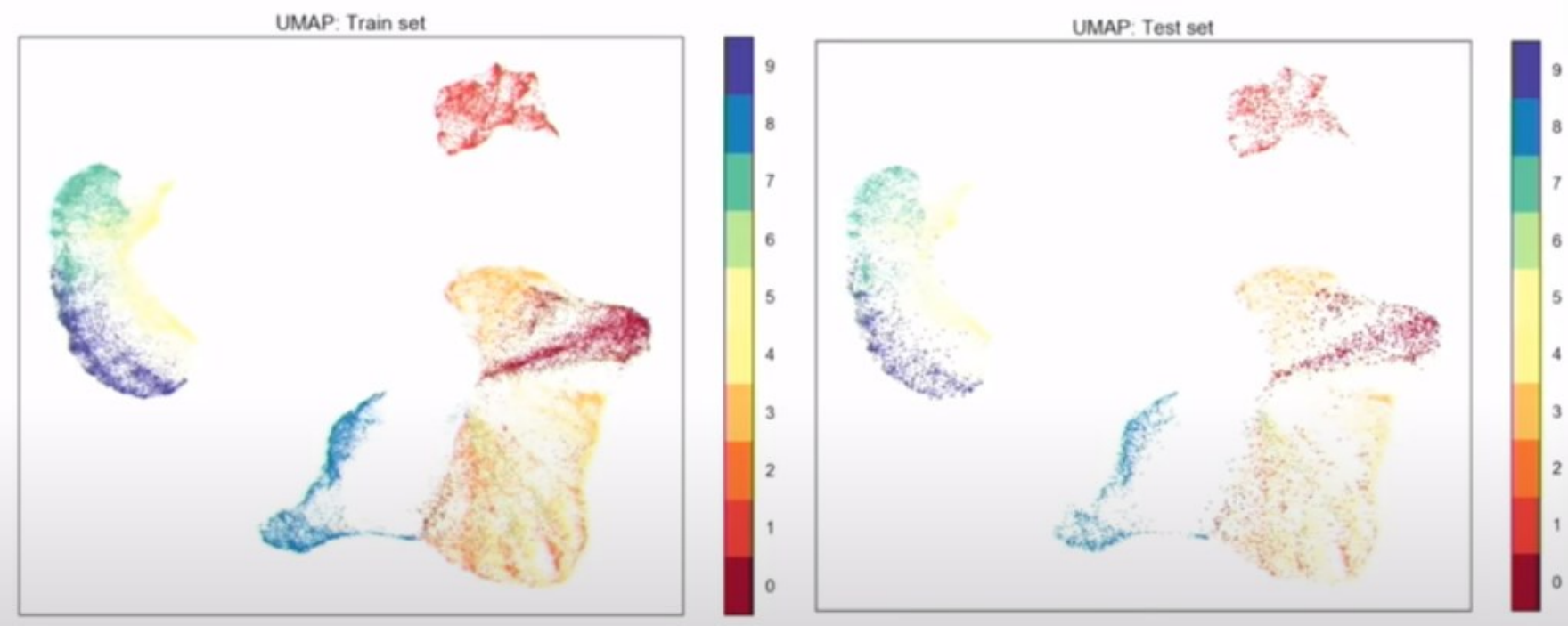


UMAP can capture long-range relationship



McInnes, L., Healy, J. and Melville, J. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”

UMAP can transform new data points



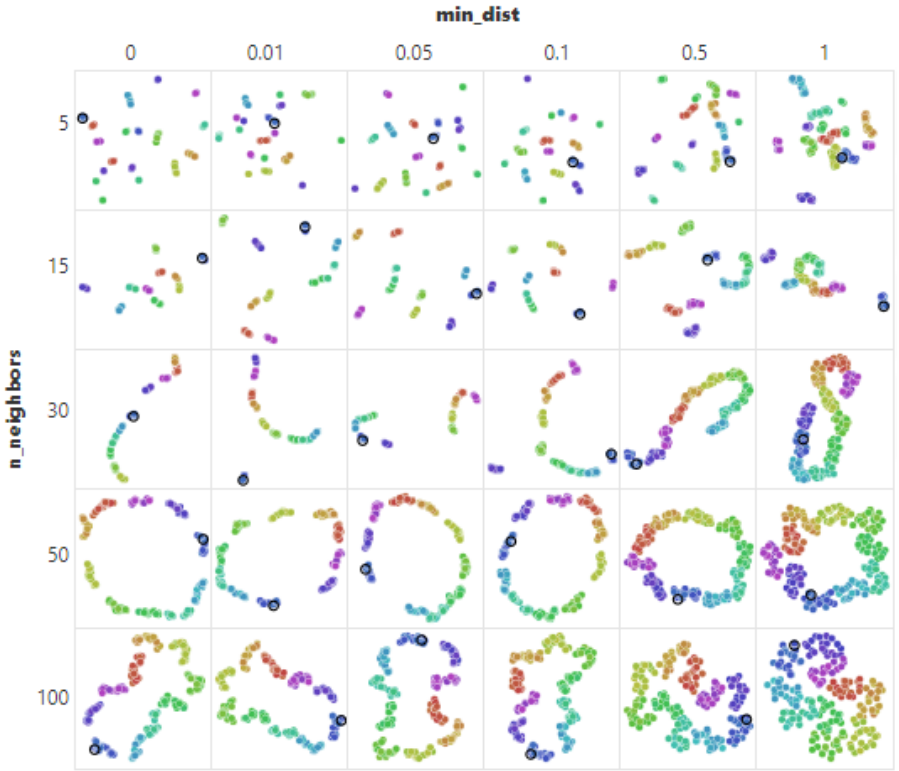
UMAP presentation by Dr. McInnes

Pros and cons of UMAP

- Can capture long-range relationship
- Can be applied to new data points without recomputing

- Require strong assumptions

Customizing UMAP outputs



- Number of neighbors (n_neighbors) is perplexity
- Minimum distant for placing similar data point (min_dist) is for adjusting the scale of visualization

Source: https://pair-code.github.io/understanding-umap/

Clustering

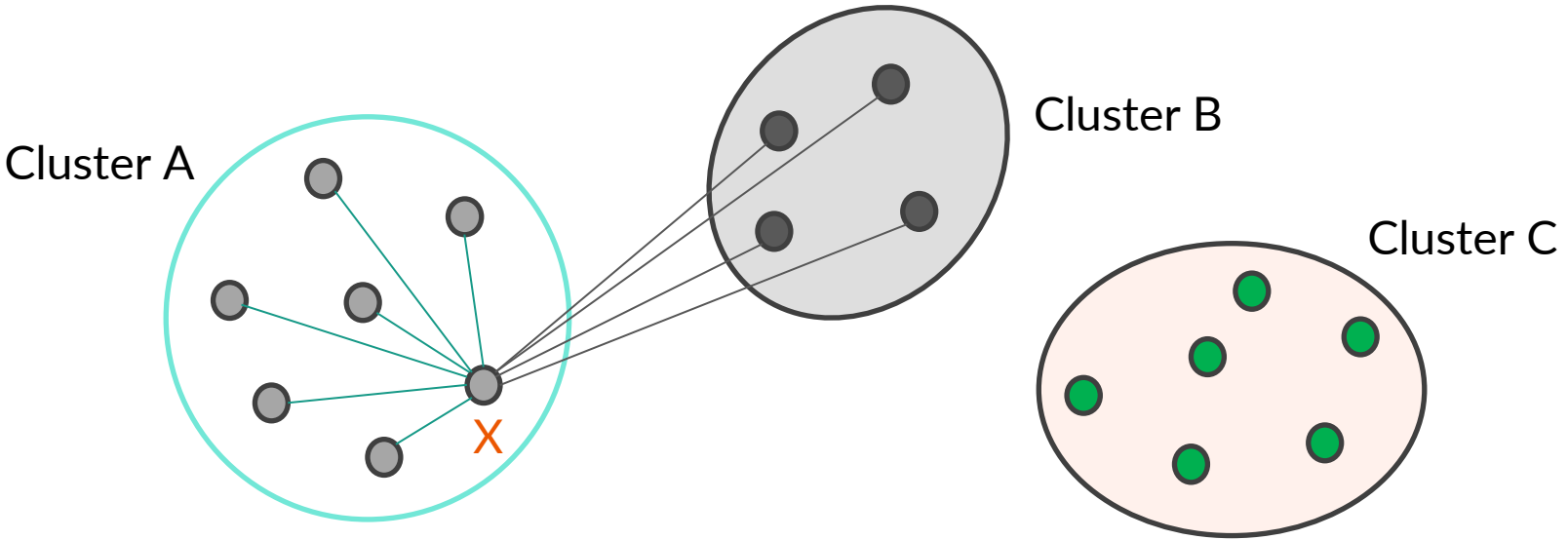
The heart of clustering

- **Goal**: Group similar data point together

- How to define similarity?
  - Distance: Between two data points
  - Linkage: Between groups of data points

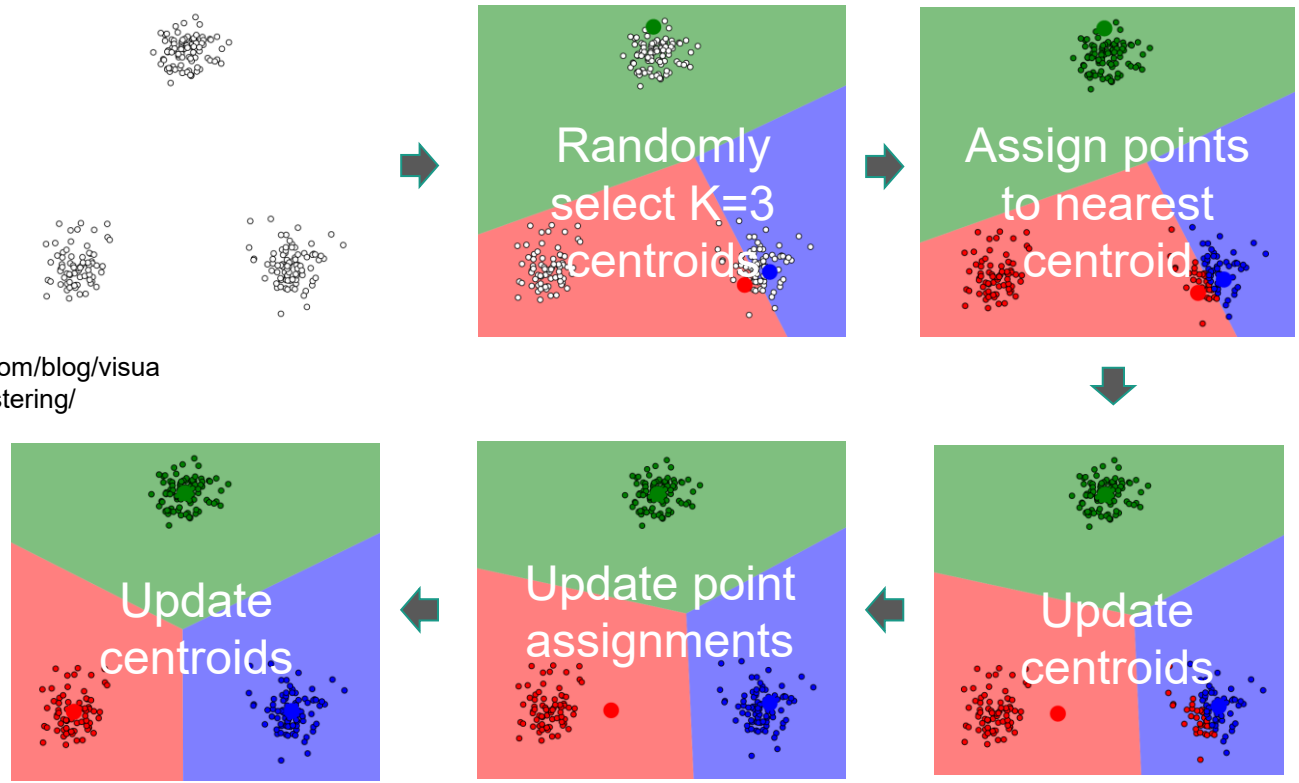
- How many clusters is appropriate?
  - Within-cluster (small) versus between-cluster (large) distance

Silhouette score



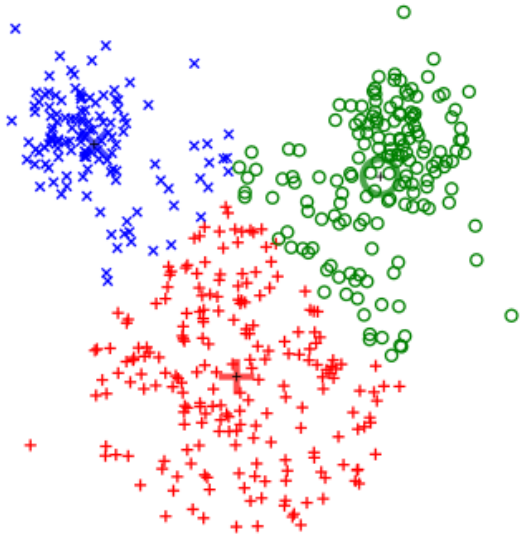
- Compare distances from X to other members of cluster A versus distances from X to members of cluster B (the closest cluster from A)

k-mean clustering



www.naftaliharris.com/blog/visualizing-k-means-clustering/

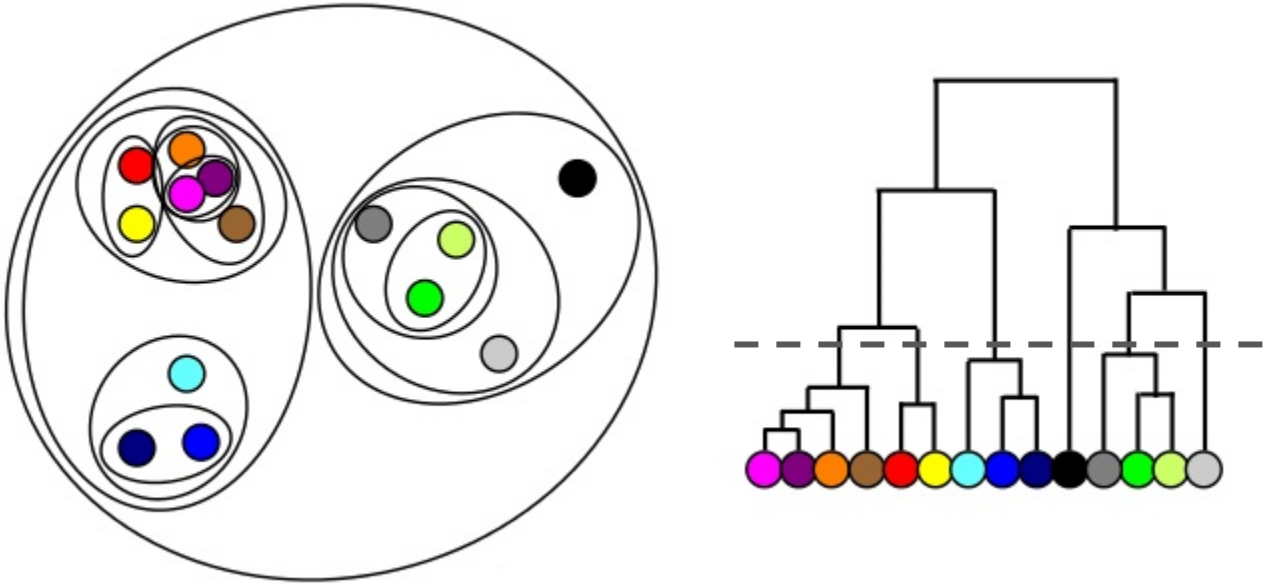
Limitation of *k*-mean



- Assume Euclidean distance
- Assume that clusters are of equal radius
- The initial guess of the locations of *k* means can affect the final clusters
  - Repeat multiple times

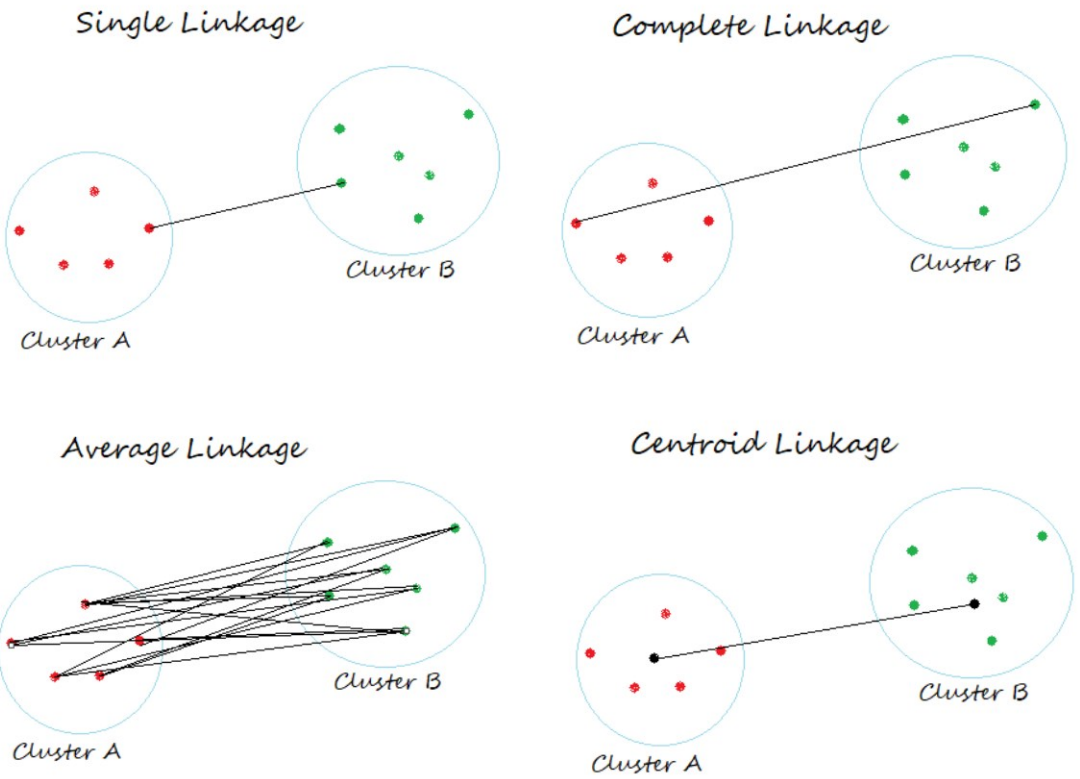
https://en.wikipedia.org/wiki/K-means_clustering

Agglomerative / Hierarchical clustering



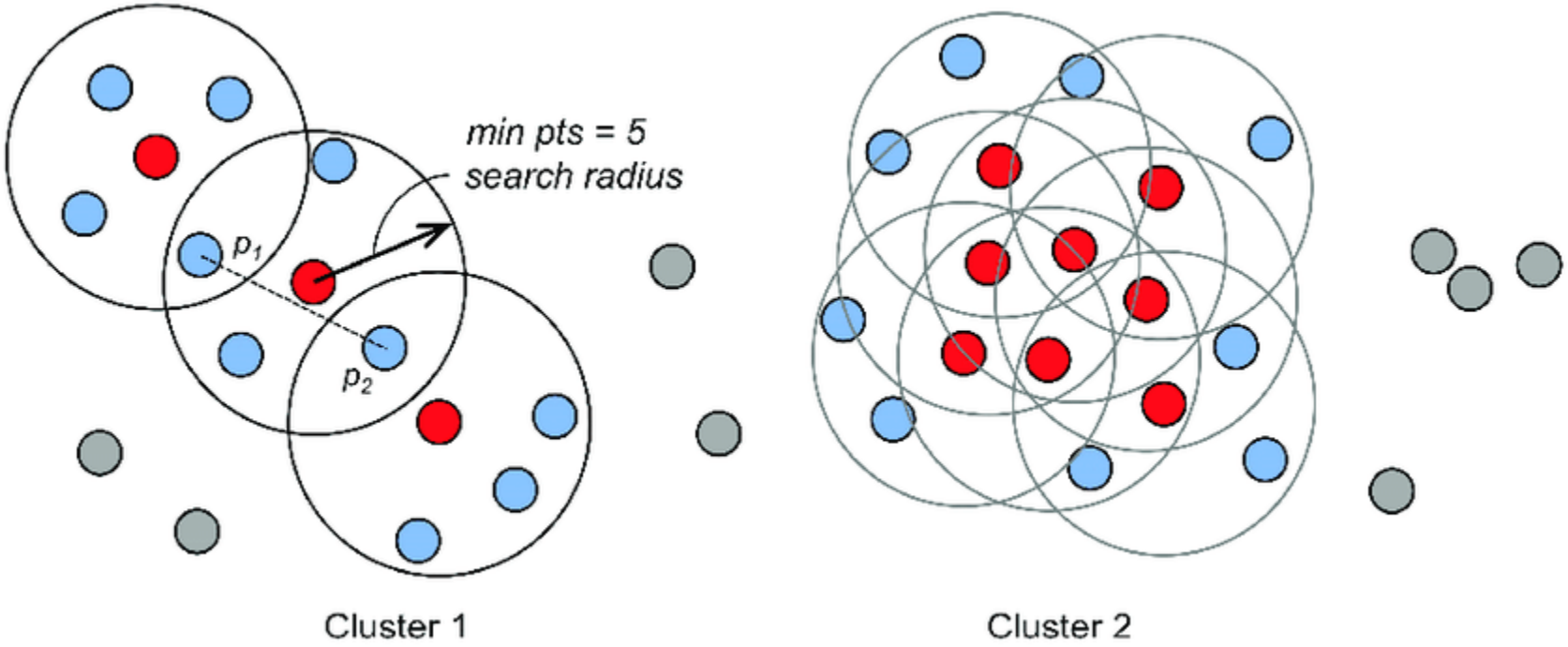
Source: www.slideshare.net/ElenaSgis/data-preprocessing-and-unsupervised-learning-methods-in-bioinformatics

Linkage = distance metric for groups of data points



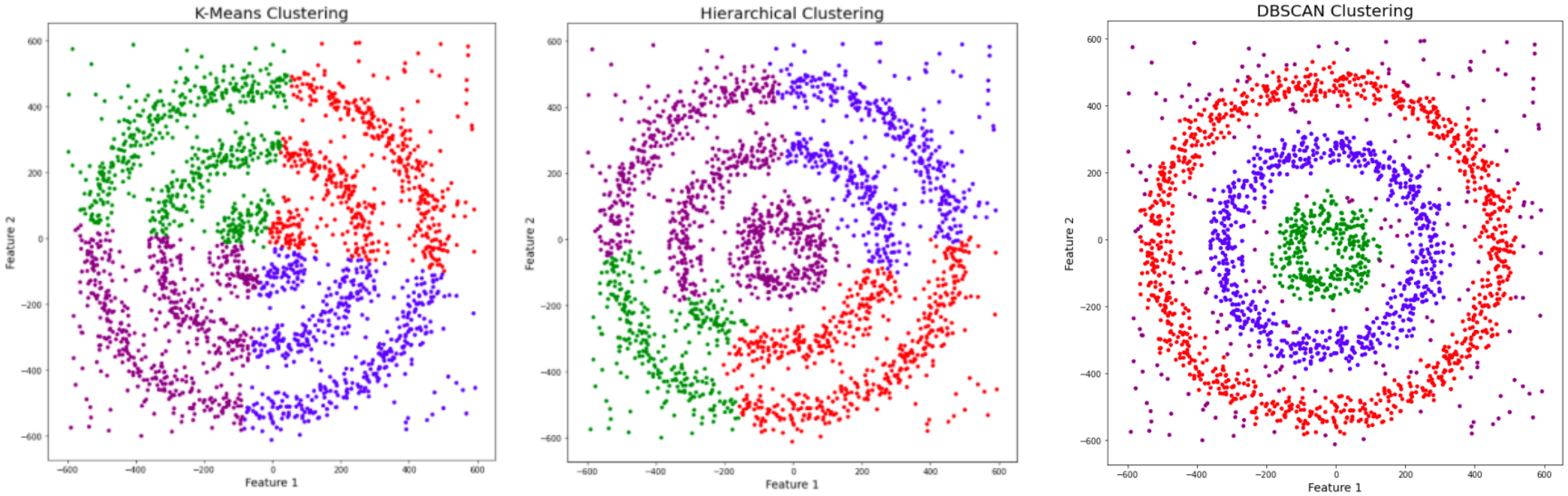
https://www.analyticsvidhya.com/blog/2021/06/single-link-hierarchical-clustering-clearly-explained/

DBSCAN: A density-based technique



Difrancesco, P.-M. Remote Sensing 12:1885 (2020)

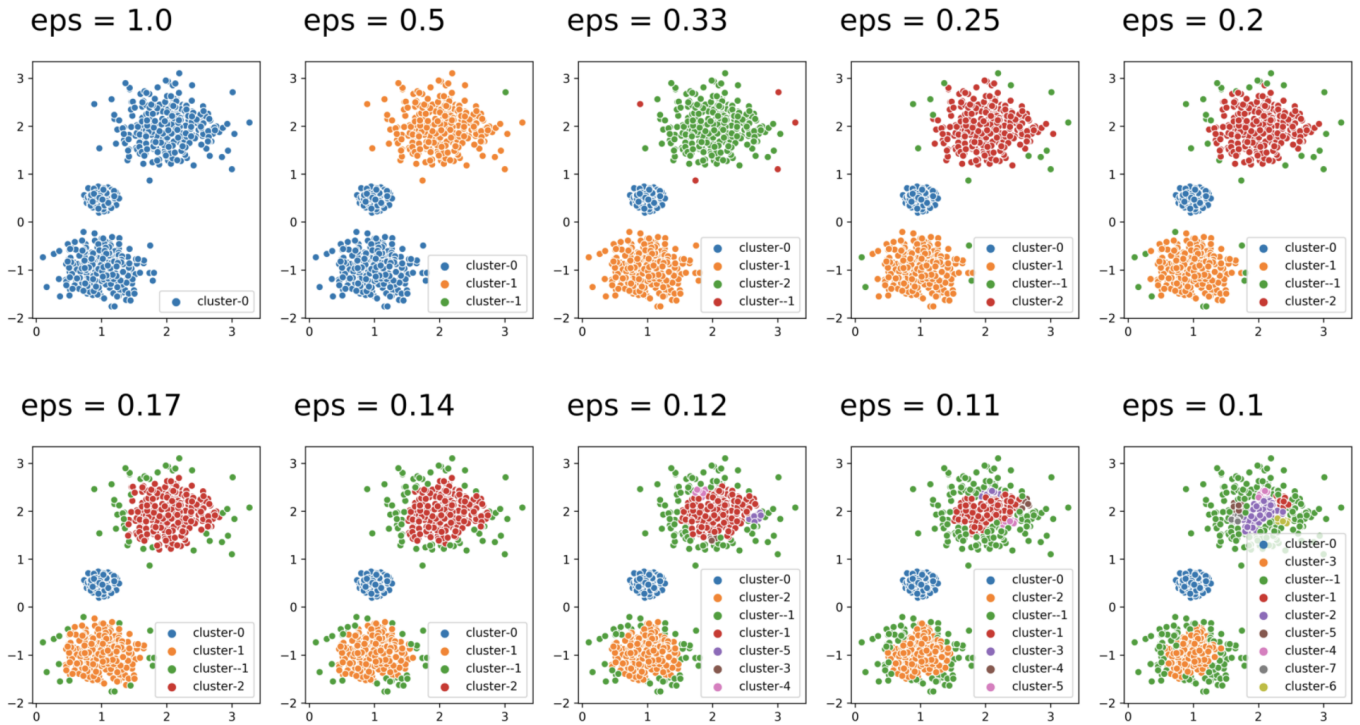
DBSCAN can handle complex cluster shape



https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/

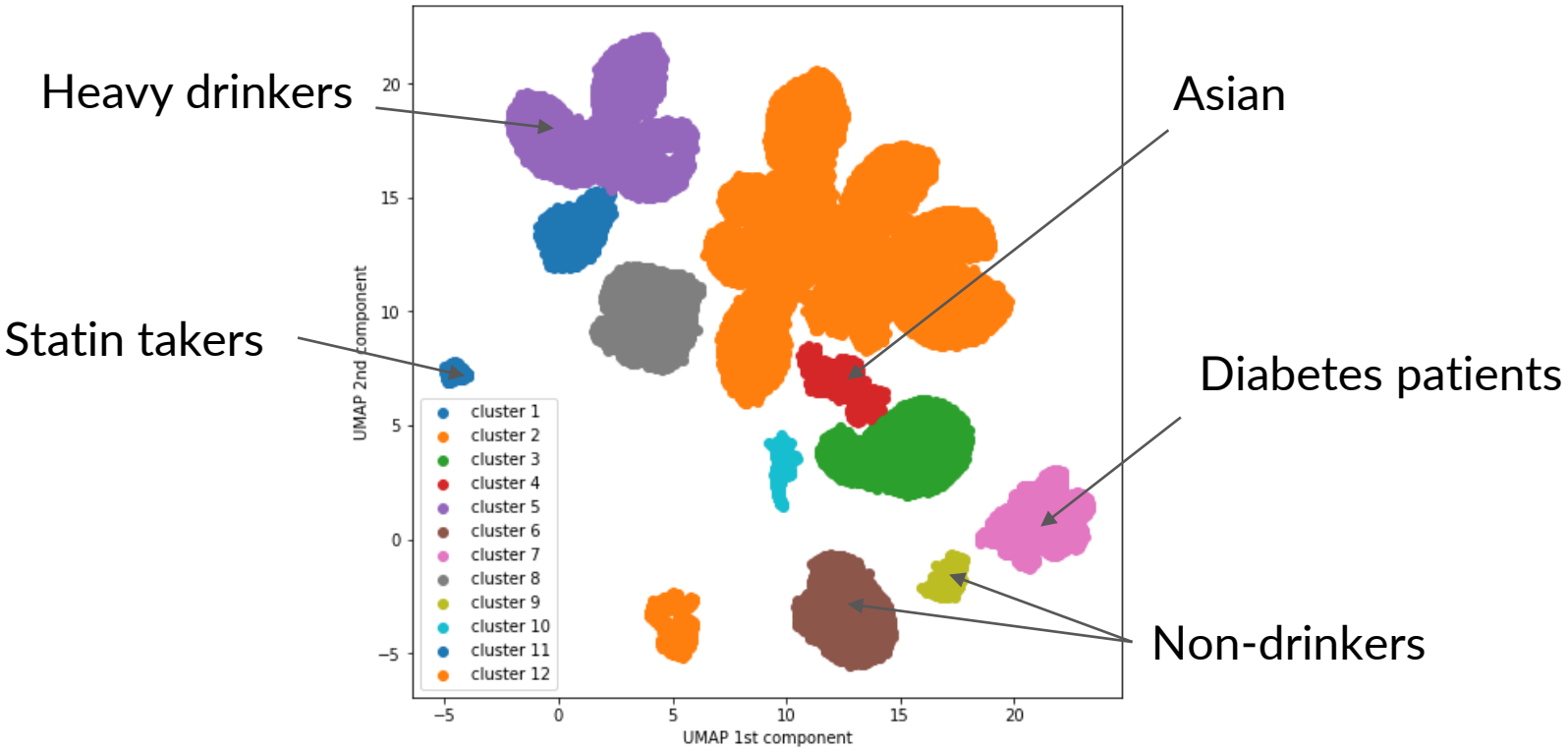
- Distance-based techniques assume that data are spread in all directions

But may be difficult to tune

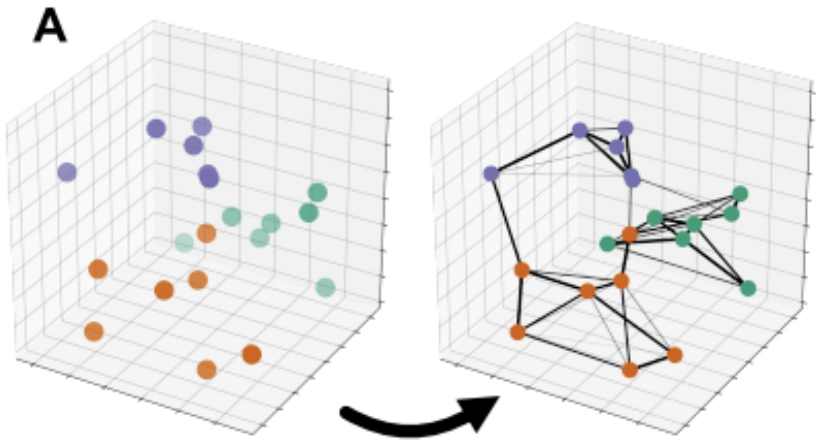


https://towardsdatascience.com/

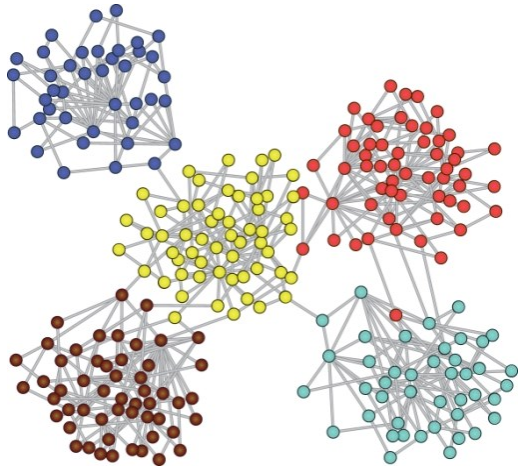
DBSCAN on patient data



Spectral clustering and network clustering



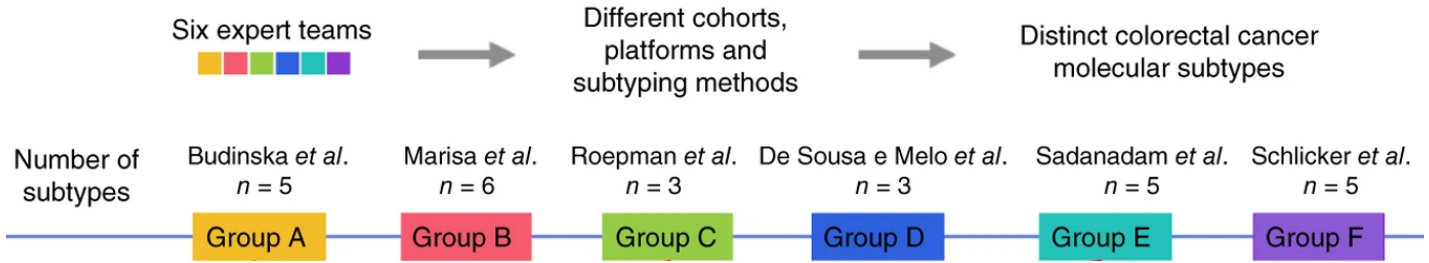
Sainburg, T. *et al.*, Neural Comput 33(11):2881-2907 (2021)

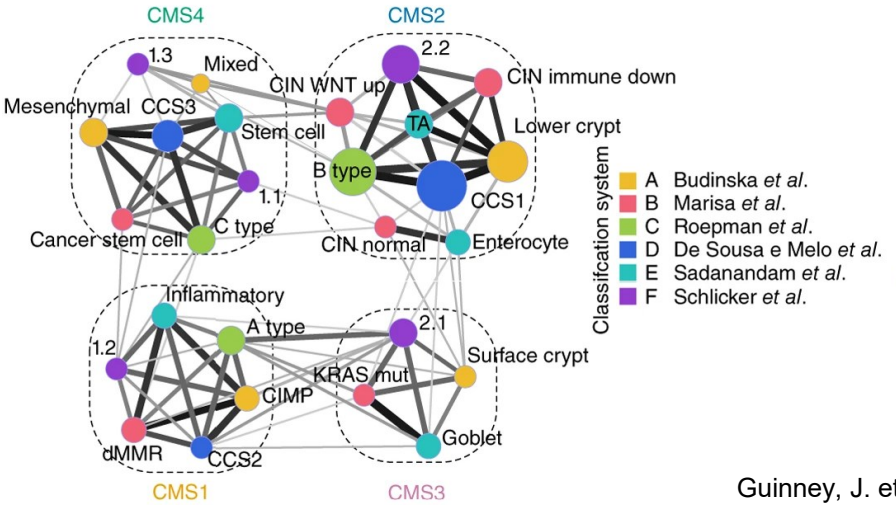


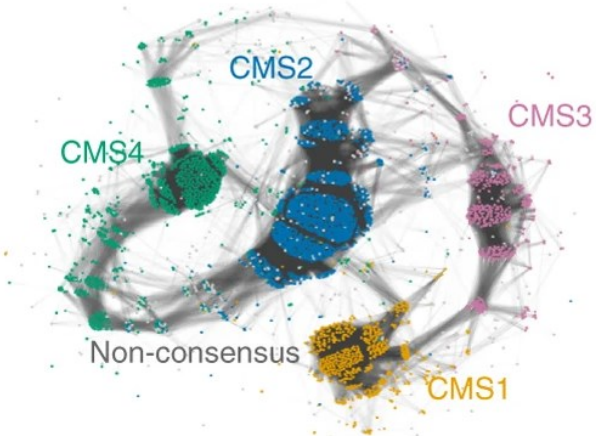
https://github.com/topics/graph-clustering

- View distance matrix as network
- Apply some threshold on the distance to create sparse network
- Split network into modules with dense edges

Colorectal cancer subtyping

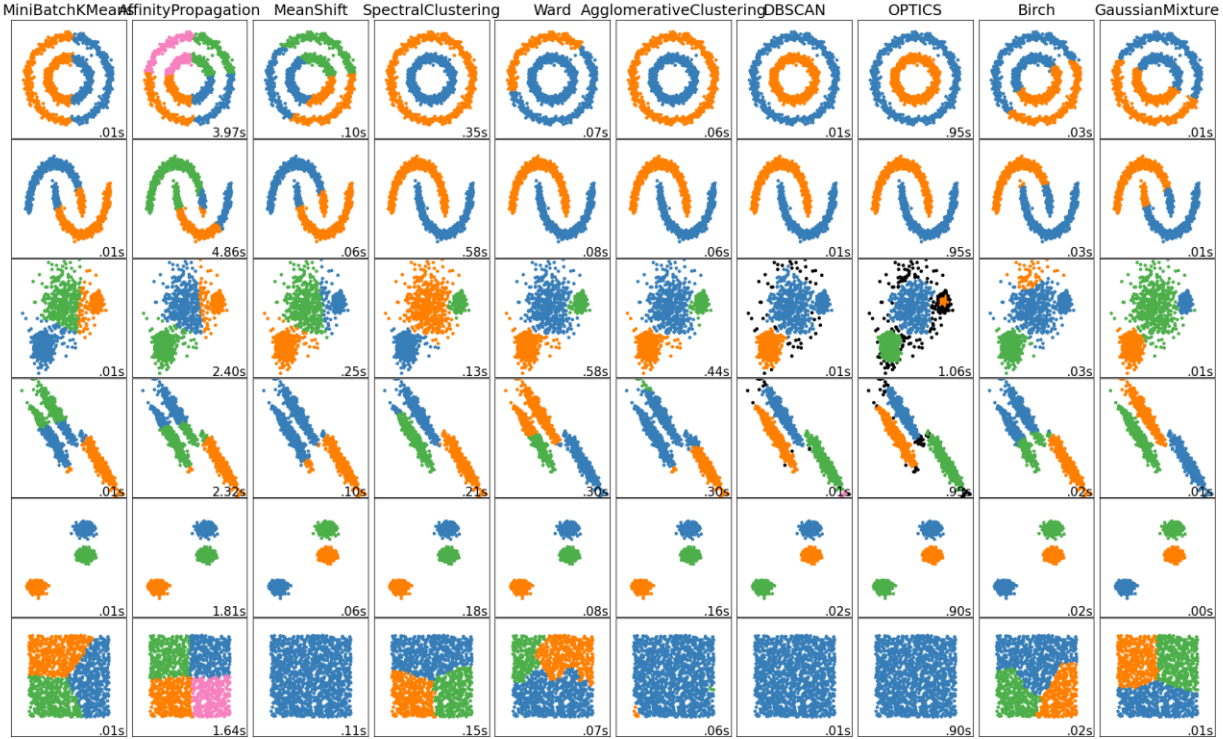






Guinney, J. et al. Nature Medicine 21:1350-1356 (2015)

No universal solution



https://scikit-learn.org/0.23/auto_examples/cluster/plot_cluster_comparison.html

Key points

- Explore distribution of feature values on the 2D/3D scatter plot

- Distance calculation influences everything!
  - Using *all* features vs *informative* features
    - $\text{Euclidean}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$
  - Pick the appropriate distance metric for the data type
    - Correlation vs Euclidean for gene expression

- Unsupervised learning needs domain knowledge

Any question?

- See you on November 23