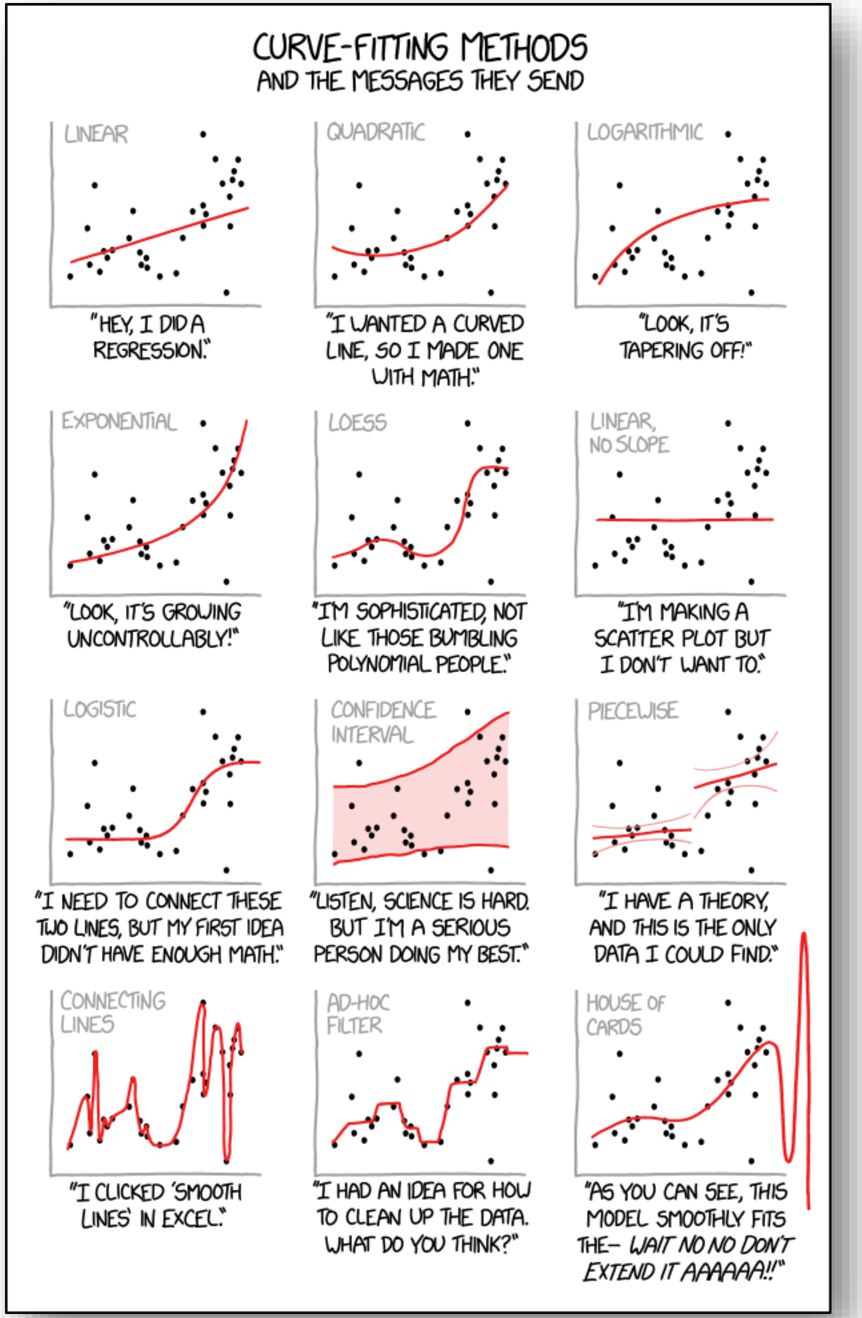
https://xkcd.com/1838/



https://xkcd.com/2048/

Classification and Supervised Learning

MACHINE LEARNING

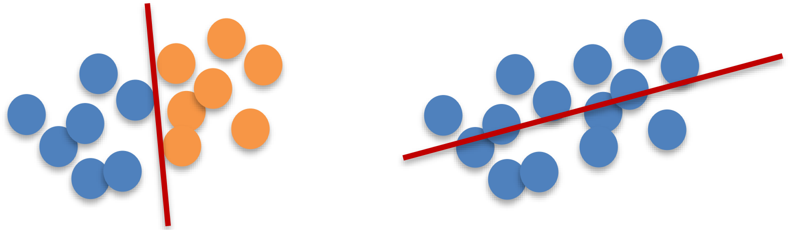
DEPARTMENT OF BIOLOGICAL PSYCHOLOGY AND NEUROERGONOMICS

Topics

- Introduction: GUI and basic calculations
- Coding 1: Scripts, style, and variable classes
- Coding 2: Control statements and loops
- Visualization 1: Basics, subplots, get and set
- Coding 3: Functions
- Visualization 2: Descriptive plots
- Coding 4: Basic input and output
- Visualization 3: Distribution and 3D plots
- Coding 5: Input and output specials – last lecture before holidays
- Machine Learning 1: Introduction and dimension reduction
- Machine Learning 2: Clustering
- Machine Learning 3: Classification
- Coding 6: Efficiency and debugging basics
- Coding 7: Advanced functions and debugging

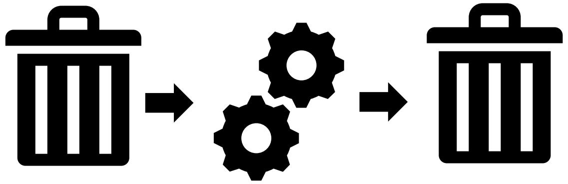
Supervised Learning

- Learn a function that maps an input to an output based on training data by minimizing the output error
- Each training data point consists of an input vector and an associated output (supervisory signal)
- The learned function can be tested using a new test dataset (test whether the classifier generalizes well)
- If the output is continuous, it‘s a regression task, if it‘s categorical, it‘s a classification task



Considerations

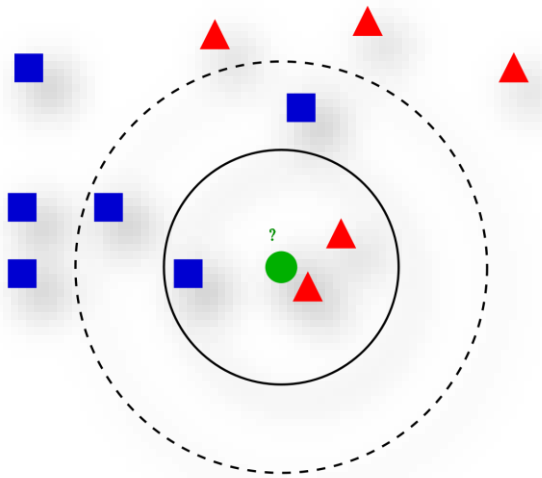
- Bias-Variance tradeoff
  - Do different, but equally good training datasets lead to equal results?
  - An algorithm must be flexible, but should not overfit
- Noise in the data
  - Noisy data with potentially inaccurate output can only lead to bad classification results (GIGO: Garbage in – Garbage out)

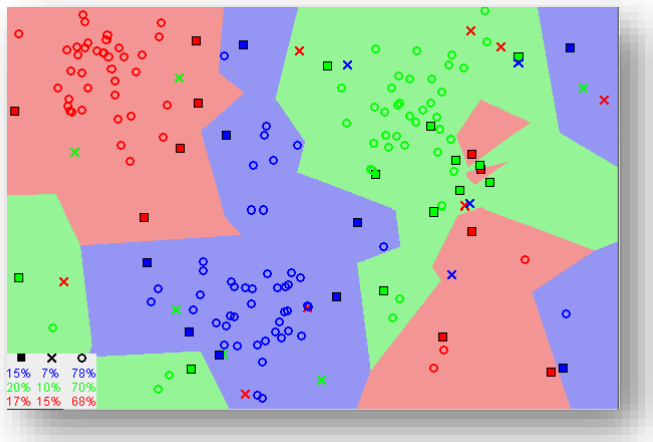


- Dimensionality and classifier complexity
  - Less is less risky (same as with clustering)
- Heterogeneity, Redundancy, Non-Linearities

K-Nearest Neighbors

- Maybe the simplest ML algorithm
- Lazy Learning: Computation is actually done only upon classification, and only locally
- Simplest case: Classification result is just the same as the class of the nearest neighbor
- Parameters
  - The number of neighbors (k)
  - Distance function
  - Weights for the k neighbors
- Nonlinear





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

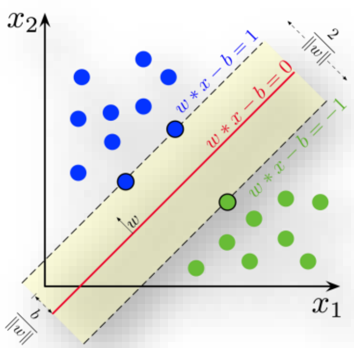
Linear Discriminant Analysis

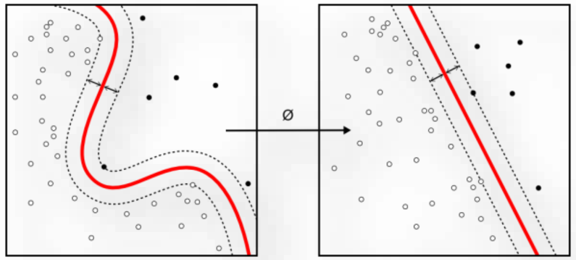
- Finds a linear combination of features (weights, like PCA) that separates known classes
- Developed by Sir Ronald Fisher (1890 – 1962)
  - The guy with the flowers (the iris dataset was an example dataset for LDA!)
  - Statistics genius, also created ANOVA, F(isher)-distribution, Student‘s-t distribution
  - Kind of a racist though (eugenics…)
- Assumptions
  - Multivariate normality
  - Homoscedasticity (all features have the same variance)
  - Independence (data points are randomly sampled)
- Can also be used for dimensionality reduction

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

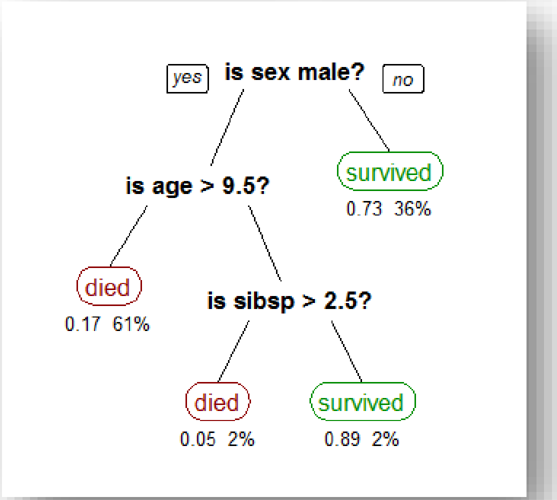
Other Algorithms

- Support Vector Machines
  - Are no machines.
  - Finds a set of support data points that define the decision boundary between two classes with a maximal margin between them
  - By using a mathematical trick (nonlinear kernels) it's possible to have them work nonlinearly
- Decision Trees
  - Splits the data into subsections based on decisions about their features
  - Can be interpreted by humans
  - Can be very non-robust to changes in the data
  - Learning such a tree takes a while so heuristcs are used
- And a bazillion more.





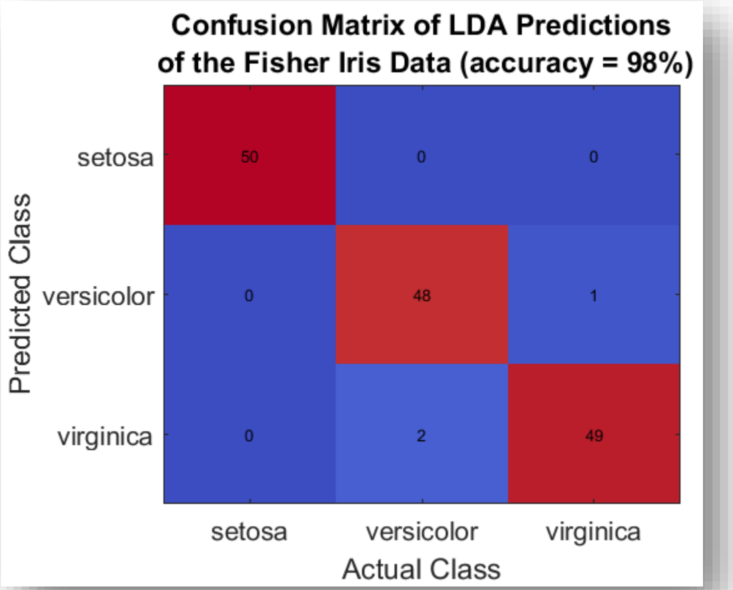
https://en.wikipedia.org/wiki/Support-vector_machine



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

Confusion Matrix

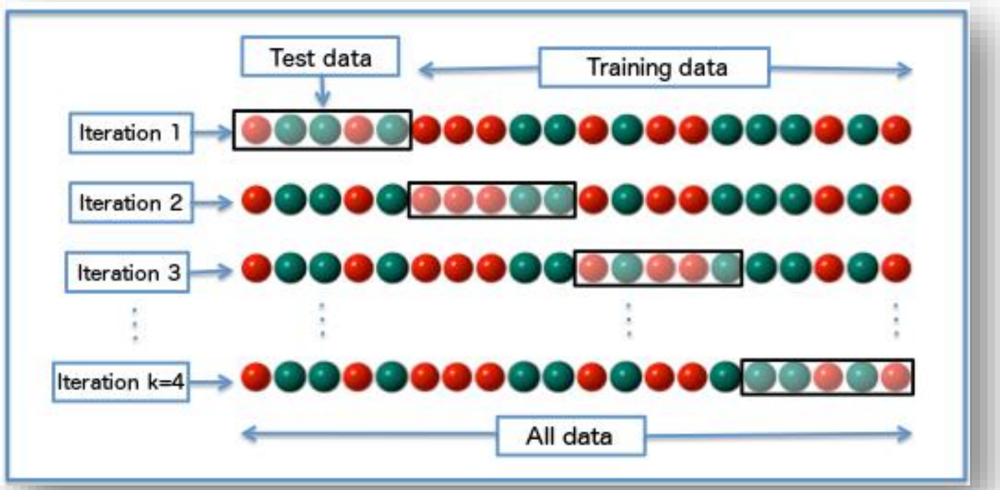
- Table that visualizes the prediction in contrast to the actual class
- Rows = Predictions, Columns = Actual Class
  - Well, it doesn‘t really matter, but stick to the convention… you can remember it by the y-axis being dependent on the x-axis, like in a function
- Signal-detection-theory can be applied
  - True-positive (sensitivity of a class, e.g. „versicolor“) = #hits (48)/#data (50) = 0,96
  - Accuracy (overall) = summed true-positive (147)/ # all data (150) = 0,98
- Very simple tool, but powerful and useful for visualization



DEPARTMENT OF BIOLOGICAL PSYCHOLOGY AND NEUROERGONOMICS

Cross Validation

- Try to asses the performance of a classifier and how well a classifier generalizes (bias vs variance)
  - In practice you don‘t just have a test dataset…
- Split the training data into subsets (folds, often 5 <= k <= 10)
- Use one of the folds as test dataset, repeat k times
- Compute mean accuracy over all folds



https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Chance Level

- Chance level is not just 1/k (e.g. 50% with two classes)!
  - This is the case only with infinite amount of data
  - It‘s also different if the classes do not have the same amount of data
    - E.g. class 1 has 10 data points, class 2 has 500 -> a classifier that always predicts class 2 has high accuracy but is useless
- Permutation-based (or binomial-distribution) statistics can be used to compute an actual statistical significance level
  - Shuffle the data around very often and see how often you get accurate by chance