1. Machine Learning: general principles

Laerte Sodré Jr.
IAG – Universidade de São Paulo

IX La Plata International School (LAPIS) on Astronomy and Geophysics
S-PLUS: The Universe in True Colors
17-21 February 2020, La Plata, Argentina

program

Classes:

- 1. Introduction to ML
- 2. Regression & Classification
- 3. Deep Learning

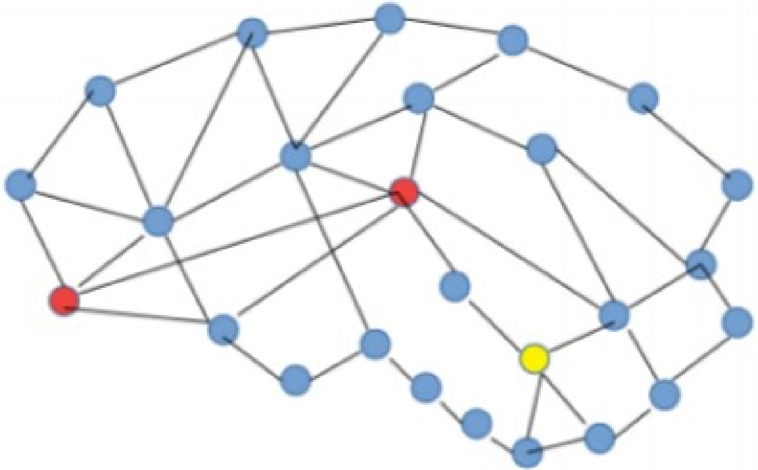
practical activities:

- **descriptive statistics, density estimation, cluster analysis, dimensionality reduction**
- **regression & classification with ML & DL**

team:

- **Laerte Sodré Jr.**
- **Maria Luísa Buzzo**
- **Lilianne Nakazono**
- **Erik Vinicius**

astronomer’s life

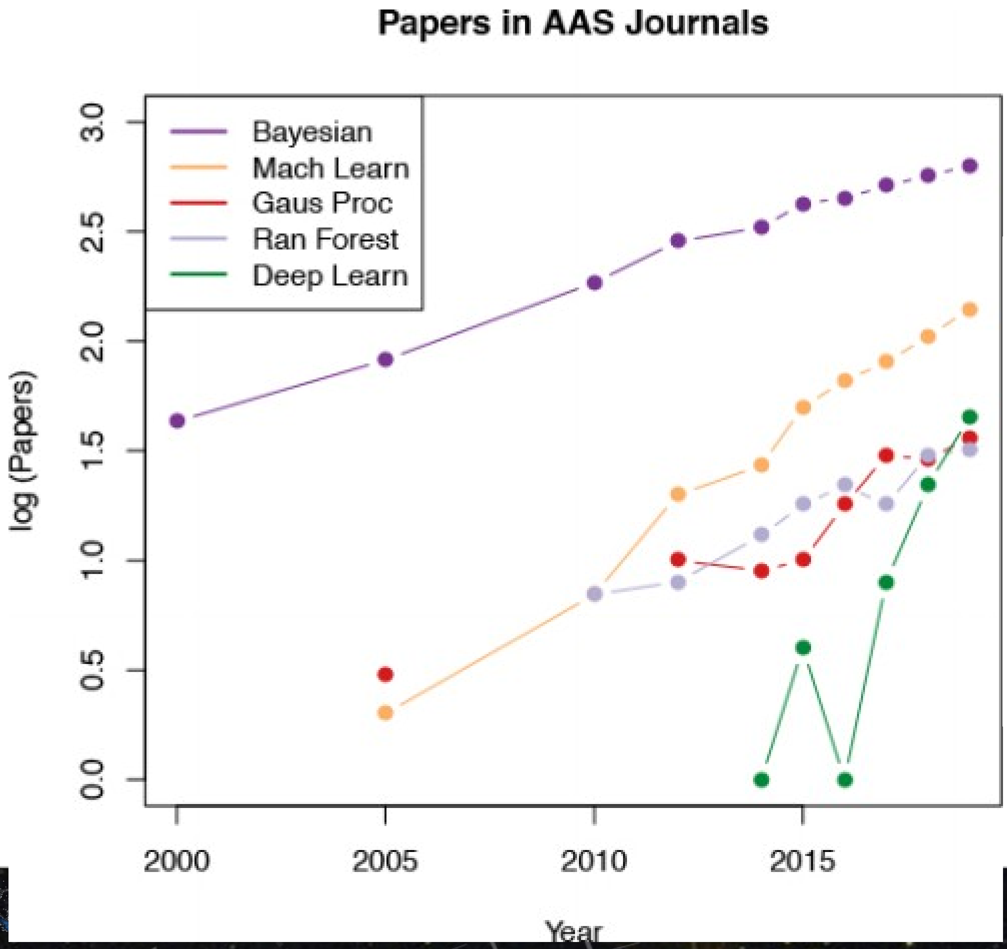


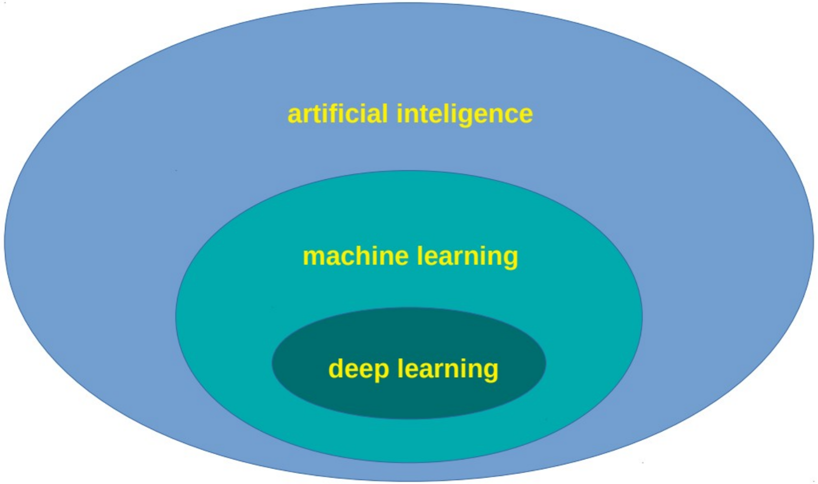
The morphology of Hα emission in CALIFA galaxies

P. M. Novais and L. Sodré, Jr.

we want to extract knowledge from the data!

machine learning in Astronomy





Classification of TNOs detected by stellar occultation using a SVM

Joel H. Castro-Chacón (1), Benjamín Hernández (2), Mauricio Reyes-Ruiz (2), José Silva (1), Bosco Hernández (2), Fernando Alvarez (2), Matthew Lehner (3,4,5) and the Team-TAOS II.
(1) CONACYT - Instituto de Astronomía, Universidad Nacional Autónoma de México, México, (2) Instituto de Astronomía, Universidad Nacional Autónoma de México, México, (3) Institute for Astronomy and Astrophysics, Academia Sinica,

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY
MNRAS 492, 5023–5029 (2020) doi:10.1093/mnras/staa127
Advance Access publication 2020 January 16

Deep learning dark matter map reconstructions from DES SV weak lensing data

Niall Jeffrey,[1★] François Lanusse,[2] Ofer Lahav[1] and Jean-Luc Starck[3]

Identifying new X-ray binary candidates in M31 using random forest classification

R. M. Arnason,[1★] P. Barmby[1,2★] and N. Vulic[1,3,4]
[1]Department of Physics and Astronomy, University of Western Ontario, 1151 Richmond Street, London ON N6A 3K7, Canada
[2]Institute for Earth and Space Exploration, University of Western Ontario, 1151 Richmond Street, London ON N6A 3K7, Canada

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY
MNRAS 489, 941–950 (2019) doi:10.1093/mnras/stz2226
Advance Access publication 2019 August 12

Estimating dayside effective temperatures of hot Jupiters and associated uncertainties through Gaussian process regression

Emily K. Pass,[1,2,3★] Nicolas B. Cowan,[1,2,4,5] Patricio E. Cubillos[6] and Jack G. Sklar[4,7]
[1]McGill Space Institute, 3550 rue University, Montreal, QC H3A 2A7, Canada
[2]Institut de Recherche sur les Exoplanètes, Université de Montreal, C.P. 6128, Succ. Centre-ville, Montreal, QC H3C 3J7, Canada

Classification of pulsars with Dirichlet process Gaussian mixture model

Fahrettin Ay,[1★] Gökhan İnce,[1] Mustafa E. Kamaşak[1] and K. Yavuz Ekşi[2†]
[1]Istanbul Technical University, Faculty of Computer and Informatics, Computer Engineering Department, 34469, İstanbul, Turkey

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY
MNRAS 492, 2236–2240 (2020) doi:10.1093/mnras/stz3621
Advance Access publication 2020 January 3

Characterization of unresolved and unclassified sources detected in radio continuum surveys of the Galactic plane

Arnab Chakraborty,[1★] Nirupam Roy,[2] Y. Wang,[3] Abhirup Datta,[1] H. Beuther,[3] S.-N. X. Medina,[4] K. M. Menten,[4] J. S. Urquhart,[5] A. Brunthaler[4] and S. A. Dzib[4]
[1]Discipline of Astronomy, Astrophysics and Space Engineering, Indian Institute of Technology Indore, Indore 453552, India

J-PLUS: Identification of low-metallicity stars with artificial neural networks using SPHINX

D. D. Whitten[1,2], V. M. Placco[1,2], T. C. Beers[1,2], A. L. Chies-Santos[3], C. Bonatto[3], J. Varela[4], D. Cristóbal-Hornillos[4], A. Ederoclite[4], T. Masseron[5,6], Y. S. Lee[7], S. Akras[8], M. Borges Fernandes[8], J. A. Caballero[9], A. J. Cenarro[4], P. Coelho[10], M. V. Costa-Duarte[10], S. Daflon[8], R. A. Dupke[8,13,14], R. Lopes de Oliveira[15,16,10,8], C. López-Sanjuan[4], A. Marín-Franch[4], C. Mendes de Oliveira[10], M. Moles[12], A. A. Orsi[12], S. Rossi[10], L. Sodré[10], and H. Vázquez Ramió[12]

The Astrophysical Journal, 715:823–832, 2010 June 1 doi:10.1088/0004-637X/715/2/823

ArborZ: PHOTOMETRIC REDSHIFTS USING BOOSTED DECISION TREES

David W. Gerdes[1], Adam J. Sypniewski[1], Timothy A. McKay[1], Jiangang Hao[1,3], Matthew R. Weis[1], Risa H. Wechsler[2], and Michael T. Busha[2]
[1] Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA; gerdes@umich.edu

machine learning in Astronomy

some data analysis applications

- density estimation
- cluster analysis
- dimensionality reduction
- regression
- classification
- time series analysis

tools/algorithms: the fauna

- decision trees
- naive bayes
- k-nn
- support vector machines- SVM
- random forest & bagging
- boosting
- deep learning
- ...

useful bibliography

- Statistics, Data Mining, and Machine Learning in Astronomy - Ivezić, Connolly, VanderPlas & Gray, 2014
- Modern Statistical Methods for Astronomy: With R Applications- Feigelson & Babu, 2012
- Machine Learning in Astronomy: a Practical Overview - D. Baron, arXiv:1904.07248
- Deep Learning with Python- F. Chollet & J. J. Allaire, 2018
- Deep Learning- I. Goodfellow, Y. Bengio & A. Courville

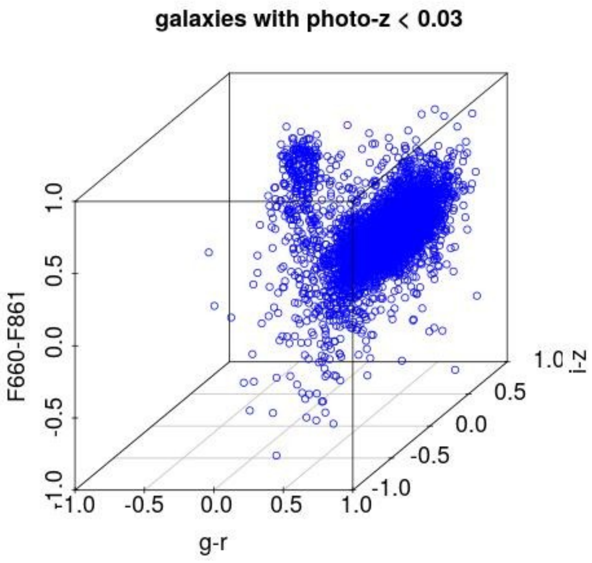
data in ML

- in many cases we work with tables (more generally: vectors, matrices, tensors)
- tables: rows, columns, cells
- rows: objects
- columns: properties/measurements of objects: ***features***
- cell: value of a given feature for a given object

Galaxy	M_r	$u-r$	$\log(t^*/\mathrm{Gyr})$	$\log(Z^*/Z_\odot)$	$\log(M_*/M_\odot)$	H. type	M. class	H α profile	c_r	c_e (H α)	c_c (H α)	ε
IC 0776	−18.69	1.94	8.34	−0.27	9.59	Sd	Slate	CL	1.95	0.55	0.20	0.28
IC 1256	−20.81	2.21	9.08	−0.29	10.72	Sb	Searly	CL	2.19	0.57	0.22	0.20
IC 1683	−20.75	2.54	9.31	−0.24	10.76	Sb	Searly	CL	2.59	0.30	0.77	0.28
IC 4566	−21.51	2.88	9.30	−0.26	11.01	Sb	Searly	E	2.24	0.63	0.08	0.26
NGC 0001	−21.11	2.24	8.89	−0.35	10.82	Sbc	Slate	CE	3.04	0.52	0.40	0.25
NGC 0036	−21.86	2.48	9.30	−0.34	11.22	Sb	Searly	E	2.49	0.72	0.05	0.25

data space

- ***a datum is a point in the data space***



There are *regularities* in the data space!

how to solve a problem with a computer?

example: star or galaxy?

- classic approach: a program with a set of rules

- machine learning:
- systems that learn from examples
- learning by trial and error, ‘correcting’ the errors

what is learning?

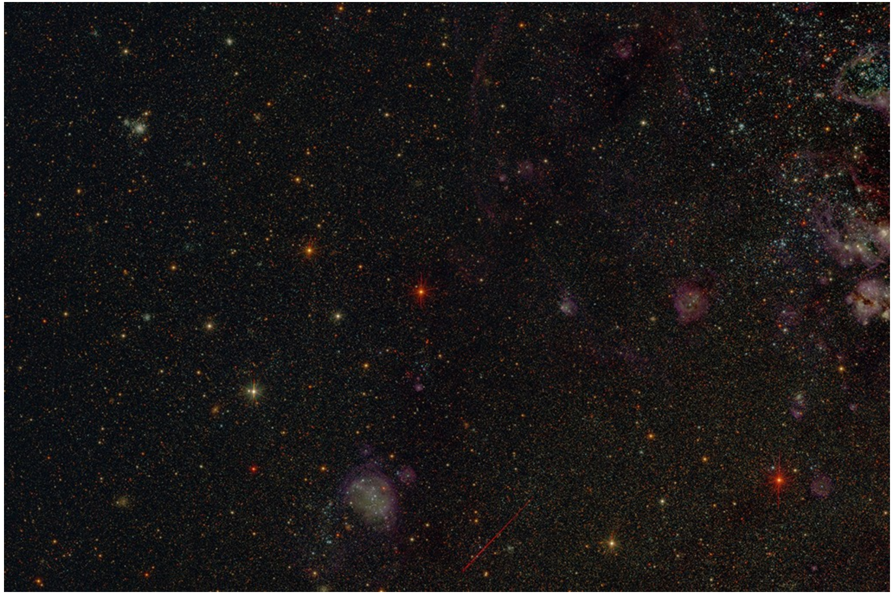
example: star or galaxy?

- system that ***learns*** from examples:

• the *algorithm* has a set of parameters which are adjusted as long as new examples are presented

• the adjustments aim improving the learning: e.g., reducing the wrong classifications

- learning = *model optimization*



learning in ML

- a common problem: estimate a variable y from a set of other variables, x:

  $$y = f(x;w)$$

  x and y can be scalars, vectors, tensors…

- a ML algorithm promotes a mapping from x to y that is equivalent to a *function* with parameters w

- *learning*: the procedure to determine w

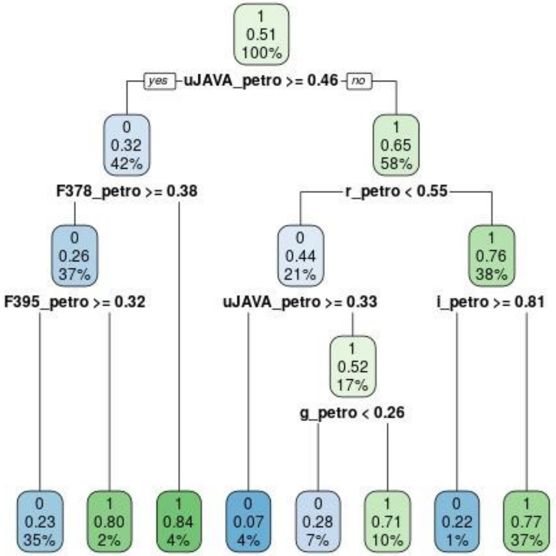
  learning = model otimization

- different ML algorithms implement different functions and learning strategies

- ML is useful for predictive modeling:

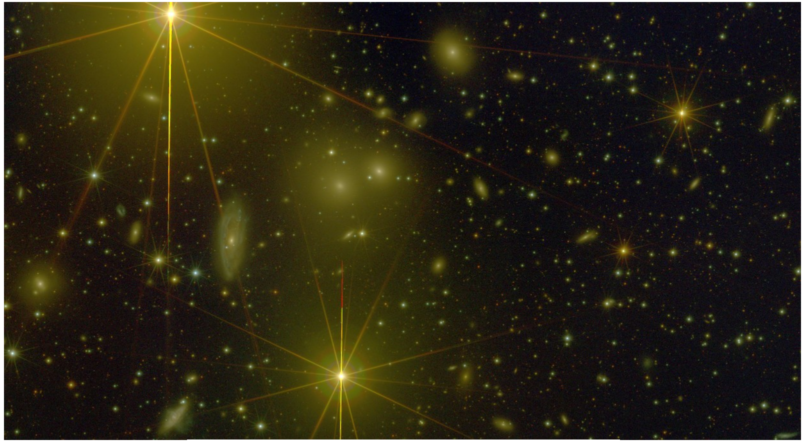
  $$y = f(x;w) + \varepsilon$$

  predictions have errors: from the data and from the model!



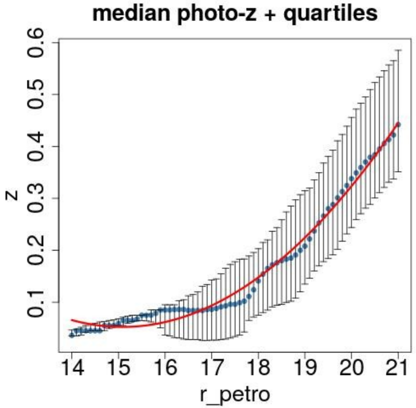
types of learning: *supervised learning*

- we have input and output variables, **x** and **y**, and the algorithm learns to map **x** on to **y** from examples **x** in a certain training set where the values of **y** are known (‘targets’)



- *classification*: **y** is a categorical variable: 'star', ‘galaxy', 'planet'

  common algorithms: random forest, k-NN, SVM, ANN, logistic regression



- *regression*: **y** is a real number, like redshift, stellar metallicity

  common algorithms: linear regression, k-NN, Gaussian Process, random forest, ANN

types of learning: *unsupervised learning*

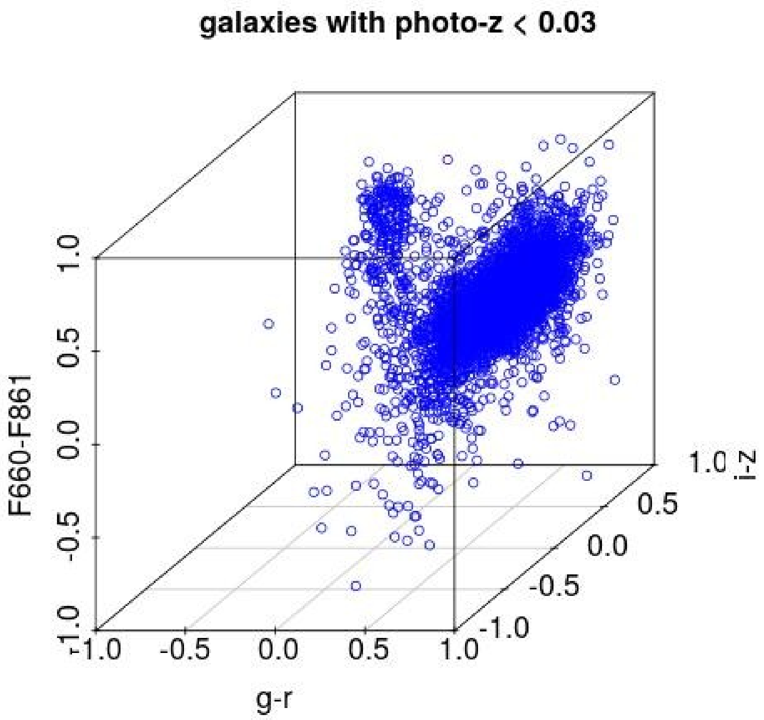
- we have the data **x** and the objective is to learn structures or interesting properties in the data

- cluster analysis (k-means, k-NN)

- density estimation (KDE)

- dimensionality reduction (pca, isomap)

- association analysis: learn the rules that describe subsets of the data, e.g., ``galaxies without star formation are red'' (Apriori algorithm)

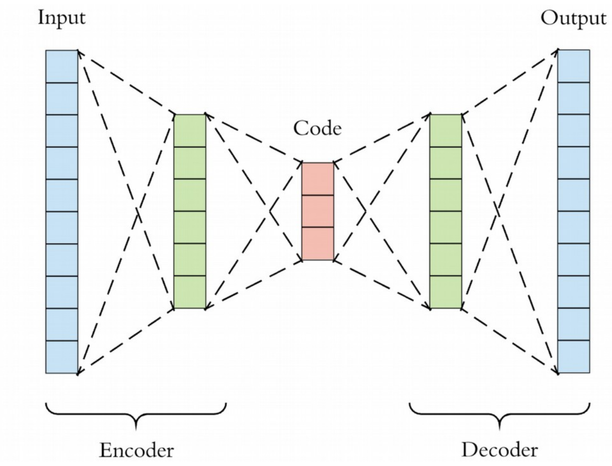


types of learning:

self-supervised learning

- kind of supervised learning without ”human intervention”
(= non supervised learning?)

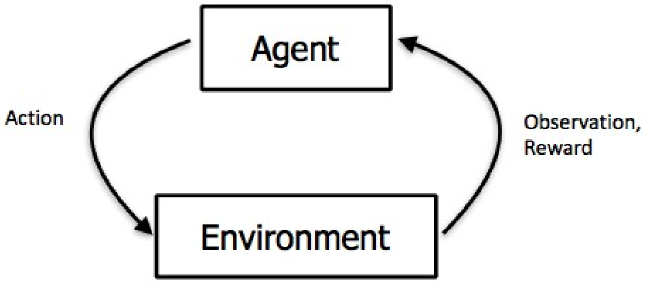
autoencoders: the algorithm learns the input

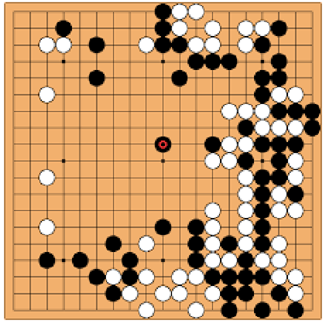


reinforcement learning

- an agent learns actions which maximize success/reward:

self-driven cars, game of Go





parametric and non-parametric models

- **parametric models:**
  a functional form with a finite set of parameters **w**

- the prediction of a new observation, **x**, depends only on **w**, not on the data:

  $P(x|w,D) = P(x|w)$

  example: linear regression

- model complexity limited;
  models of ``low flexibility''

- **non-parametric models:**
  "no-suposition" on the nature of the data

- do not assume any function:
  the mapping depends only on the data
  example: k-NN

- "flexible methods"

- *non-parametric models do have parameters!*

generalization

- models are trained with data in a training set
- how well a model behaves with new data?
- good models should ***generalize*** well!

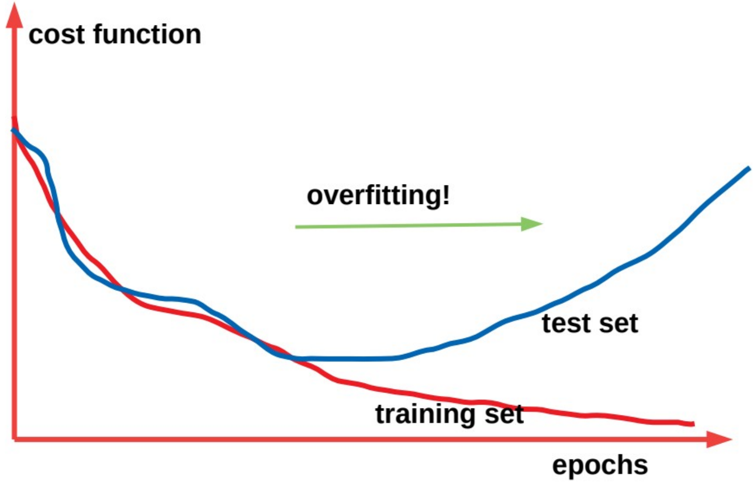
- data in ML:
- in many cases it is advisable to divide the data in ***training***, ***validation***, and ***test*** data sets

- training set: used to fit the model parameters

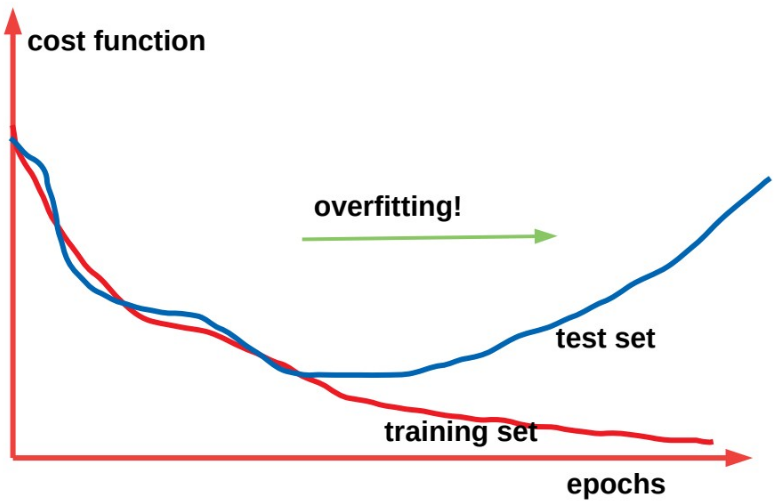
- validation set: to monitor the learning of the training set

- test set: to evaluate the model performance- ***it should not have been used in training or validation***

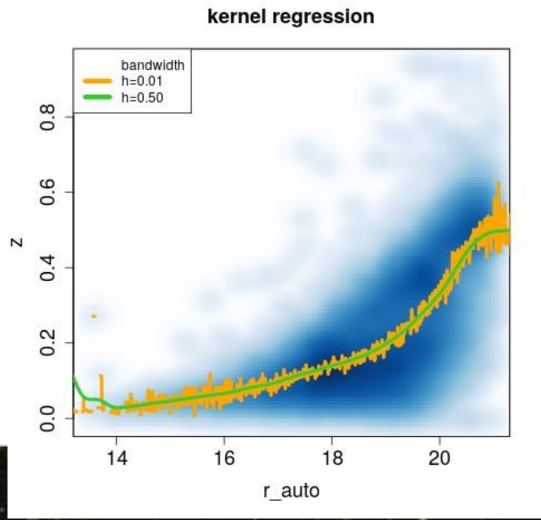
- as the algorithm learns, the error in the training and validation sets decreases
- after some optimal point, the error of the training set continues to decrease but the error of the validation set starts to increase



generalization

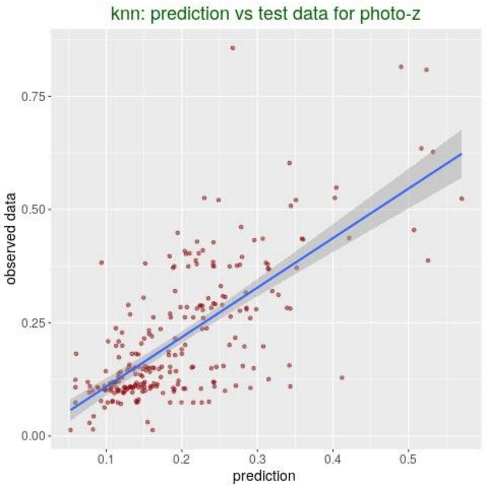


- this optimal point maximizes the generalization
- stopping the learning before or after this point leads to:
  - underfitting: the algorithm didn't learn enough, or
  - overfitting: the algorithm is learning the noise of the training set

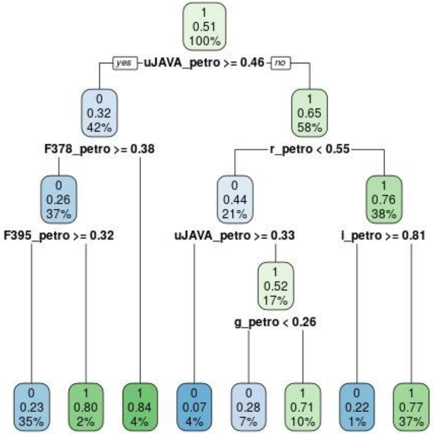


applications of ML

- **regression:** y = f(x;w)
- **y is a continuous variable, we want to <u>estimate a continuous value</u>**
- **examples: multi-linear regression, neural networks, ...**

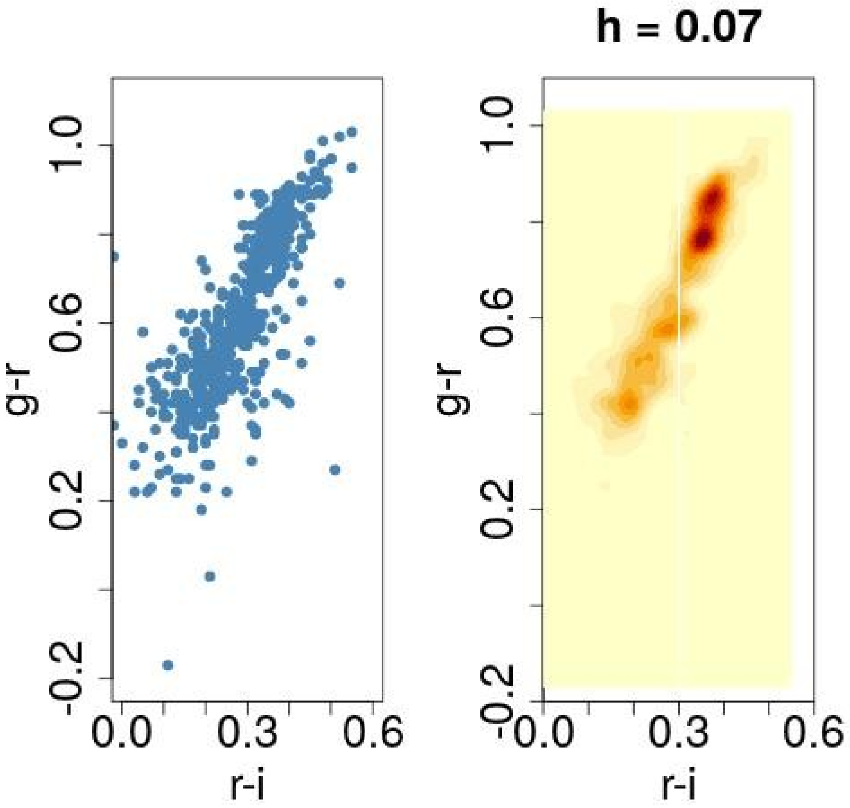


- **classification:** y = f(x;w)
- **y is a categorical/discrete variable, we want to <u>estimate classes</u> (binary or multiclass)**
- **examples: logistic regression, decision trees, k-nn, neural networks, ...**



applications of ML: density estimation

- modeling the data distribution in a data space
- inference of a probability distribution function (PDF)
- parametric methods: Gaussian Mixture Models (GMM), …
- non-parametric methods: Kernel Density Estimators (KDE), k-NN, ...

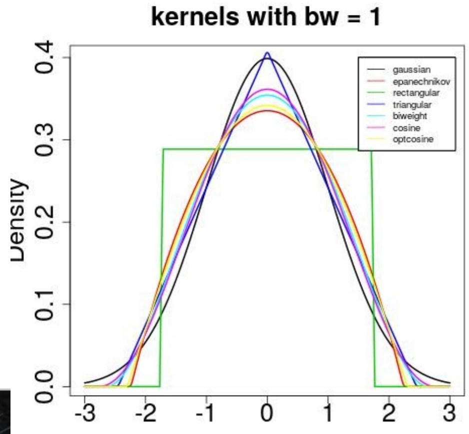


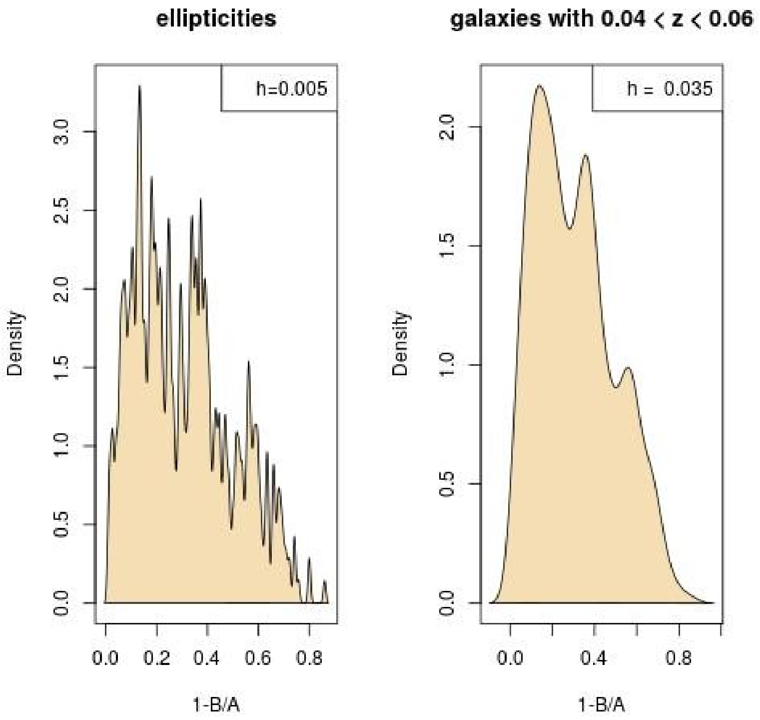
applications of ML: kernel density estimation (KDE)

- for data in any dimension D
- density (pdf) at a point x:

$$\hat{f}_N(\mathbf{x}) = \frac{1}{Nh^D}\sum_{i=1}^{N} K\left(\frac{d(\mathbf{x},\mathbf{x}_i)}{h}\right)$$

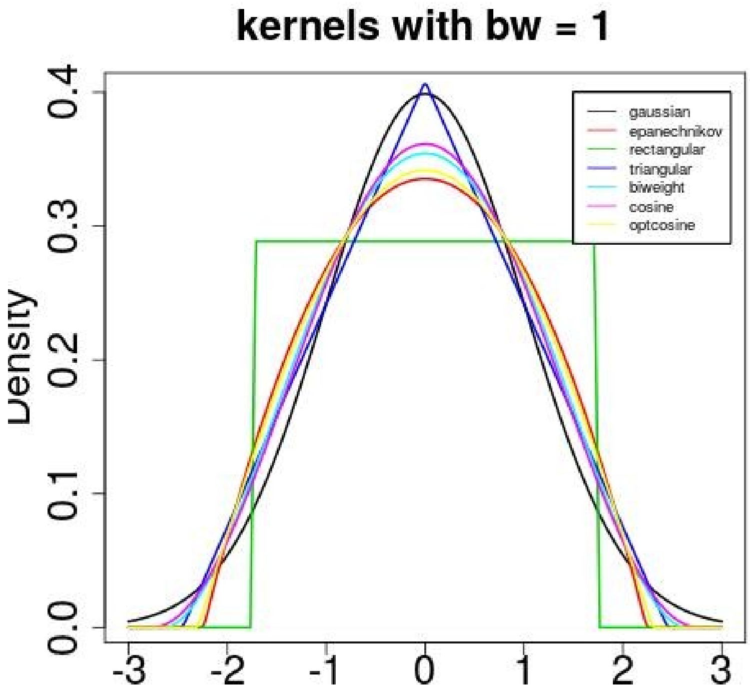
- $K(u)$: kernel function
- $d(x_1, x_2)$: "distance" between x_1 and x_2
- h: band-width

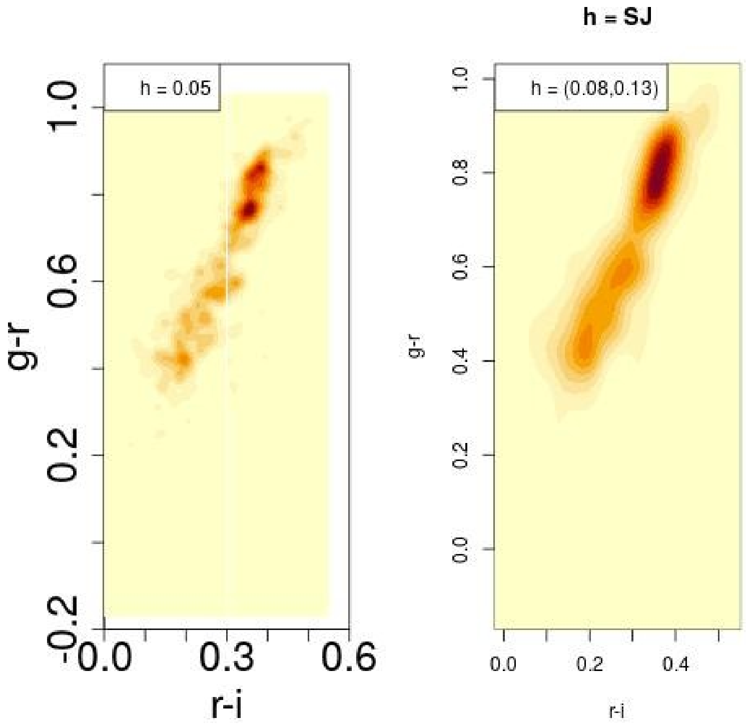




- estimation of h: cross-validation

applications of ML: kernel density estimation (KDE)





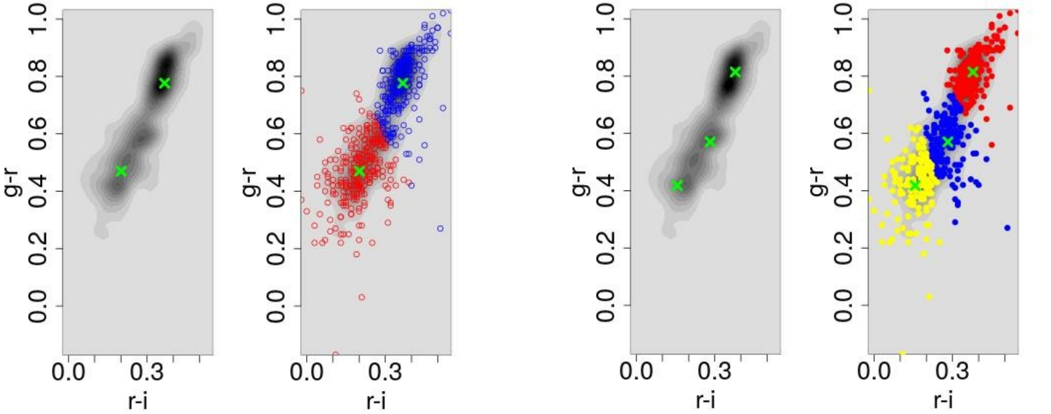
- estimation of h: cross-validation

applications of ML: cluster analysis

- objective: identification of groups/clusters in data space
- clusters: objects with similar properties
- unsupervised technique: the clusters are not known in advance
- different from classification: supervised approach- the objects are associated to pre-defined classes

- types of distances between two objects (x_i and x_j):

- Euclidian: $\mathrm{d_{ij}} = \sum_{\mathrm{i=1}}^{\mathrm{D}} |\mathrm{x_i} - \mathrm{x_j}|$
- Manhattan: $\mathrm{d_{ij}} = \left[\sum_{\mathrm{i=1}}^{\mathrm{D}} |\mathrm{x_i} - \mathrm{x_j}|^2\right]^{1/2}$
- Mahalanobis: $\mathrm{d_{ij}} = \left[\sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{x_i} - \mu).\mathrm{C}^{-1}.(\mathrm{x_j} - \mu)\right]^{1/2}$



applications of ML: dimensionality reduction

- X: data in D dimensions

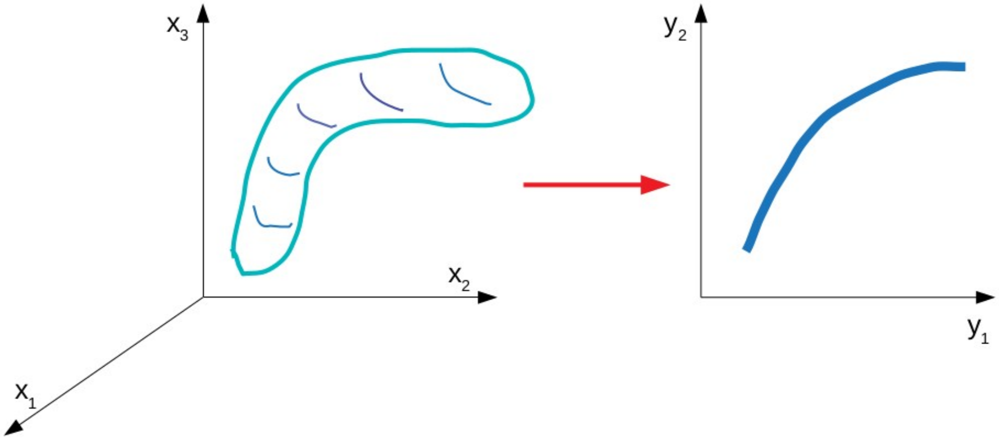
- we want a new representation of X, which we will call Y, in d << D dimensions

  $X = \{x_1, x_2, \ldots, x_D\} \rightarrow Y = \{ y_1, y_2, \ldots, y_d\}$

- embedding: one mathematical structure is contained within another
- useful for data compression an visualization

- linear methods: PCA

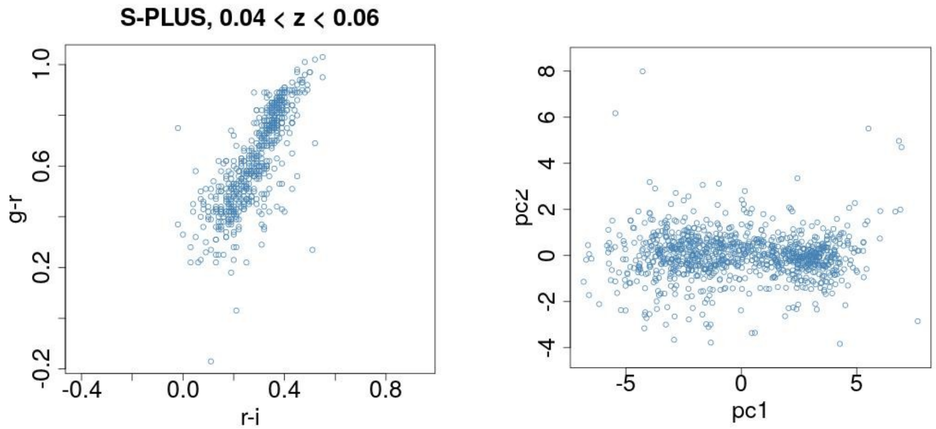
- non-linear methods:
  LLE (locally linear embedding), IsoMap, t-SNE (t-distributed stochastic neighbor embedding)



applications of ML: dimensionality reduction with PCA

- data space *D*:

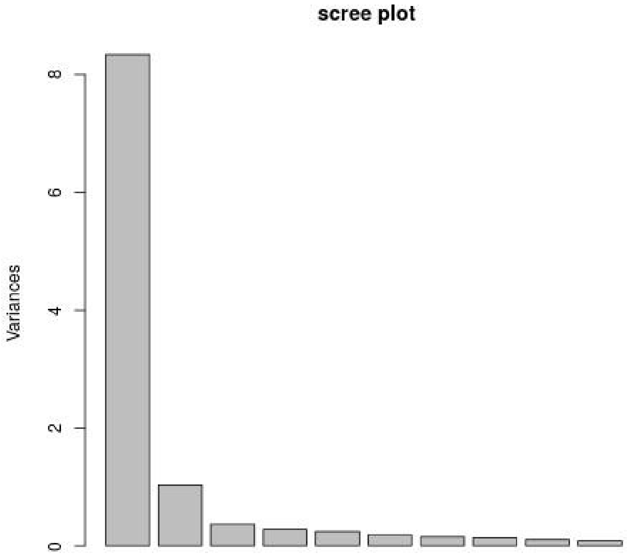
- D-dimensional space where each coordinate is a feature
- an object is a point in *D*
- PCA helps to 'visualize' *D*
- PCA performs a rotation in *D* coordinates

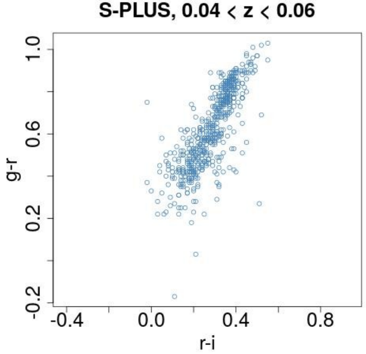


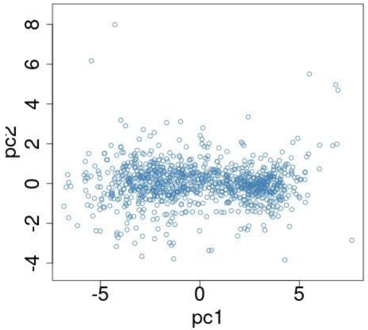
- new axes: principal components
- PC1: direction of maximum variance in *D*
- PC2: direction of maximum variance in the sub-space perpendicular to PC1
- PC3: direction of maximum variance in the sub-space perpendicular to PC1 and PC2
- ...

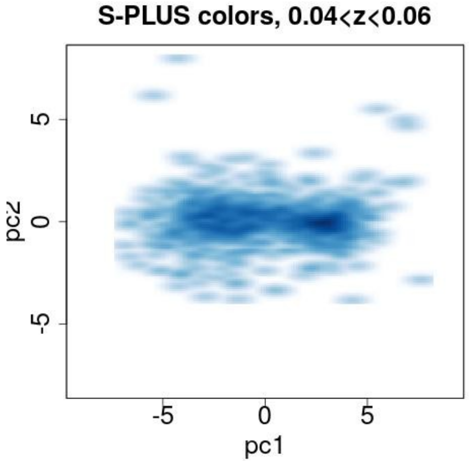
applications of ML: dimensionality reduction with PCA

- Scree plot: variance explained by each component







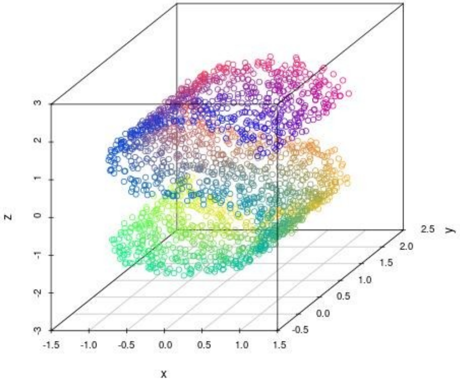


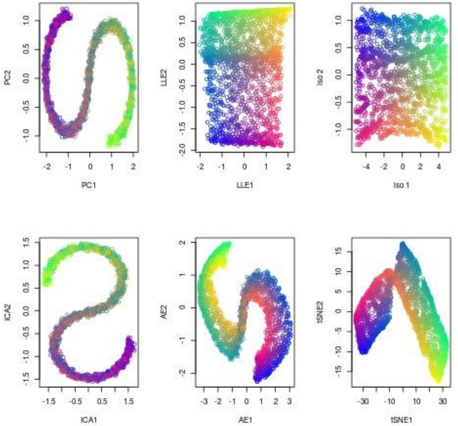
applications of ML: dimensionality reduction with non-linear methods

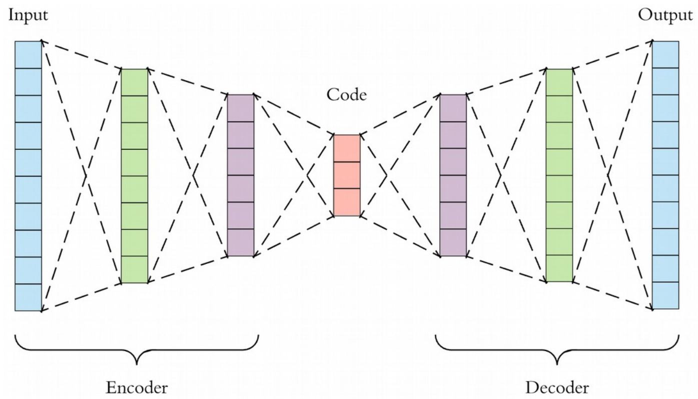
- **LLE: locally linear embbeding:**
  - **represents the data in a low-dimensional space by preserving the neighborhood of each point**
  - **'neighborhood': k-th nearest neighbor of each point**

- **isomap: preserves the geodesic distance between points**

- **autoencoders: neural nets trained with the output equal to the input**







Machine learning

where, in your work, non-parametric methods could be interesting?

applications of ML: dimensionality reduction with PCA

- data space *D*:
  $Y = N \times D$ matrix
- In many ML problems most of the time is spent on data cleaning and normalization
- PCA: data standardization- for each feature we remove its mean and divide by its standard deviation:
  $X = (Y - \langle Y \rangle)/\sigma_y$
- covariance matrix: $C = X^T X$ (D x D)
- eigenvectors V and eigenvalues λ of C:
  $CV = \lambda V$
- W_k: matrix D x k with k eigenvectors
- approximation: $Y \approx X W_k$ (N x k matrix)

S-PLUS, 0.04 < z < 0.06