

**POLIMI** GRADUATE  
SCHOOL OF **MANAGEMENT**

# EXECUTIVE PROGRAM IN DIGITAL TRANSFORMATION FLEX

BIG DATA, MACHINE LEARNING AND AI

# ABOUT ME

- ▶ Researcher at DI-UNIMI
  - [mauricioabel.soto@polimi.it](mailto:mauricioabel.soto@polimi.it)
- ▶ Previous
  - Mathematical Engineering - U. of Chile
  - PhD. Computer Science - U. of Paris
  - Researcher - U. of Orléans, U. of Chile, UNIMIB
  - Researcher at DIG-POLIMI
- ▶ Research interests
  - Graph theory
  - Evolutionary networks
  - Data representation (NLP)

# AGENDA

## 1. Day 1

- Pre-processing
- Classification
- Assignment

## 2. Day 2

- Discussion
- Assignment
- Regression

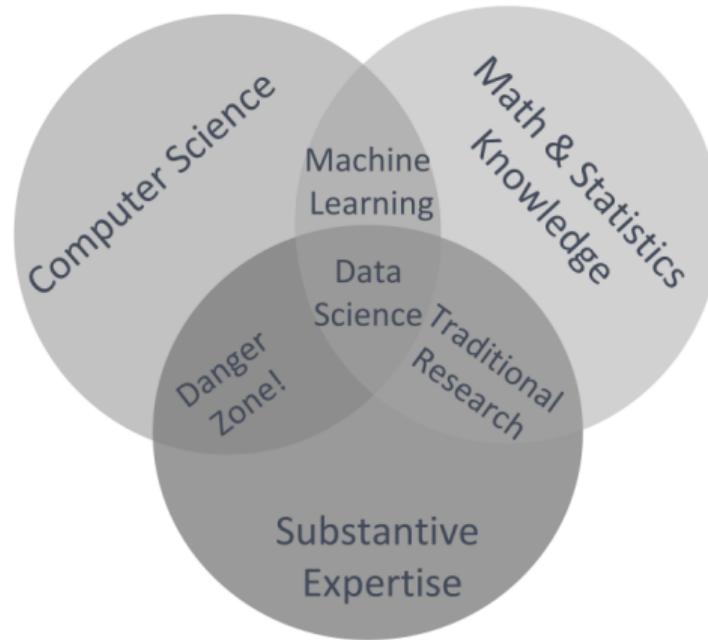
# QUESTION TIME

Go to <https://pollev.com/mauriciosoto>



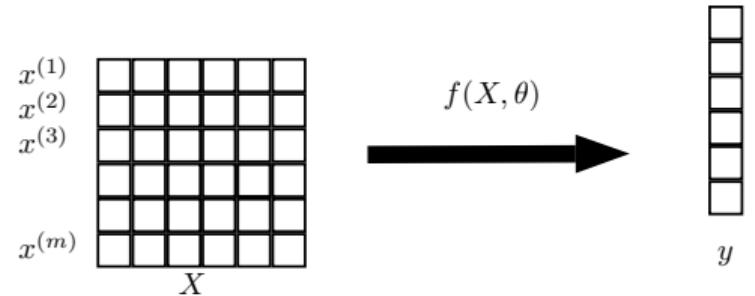
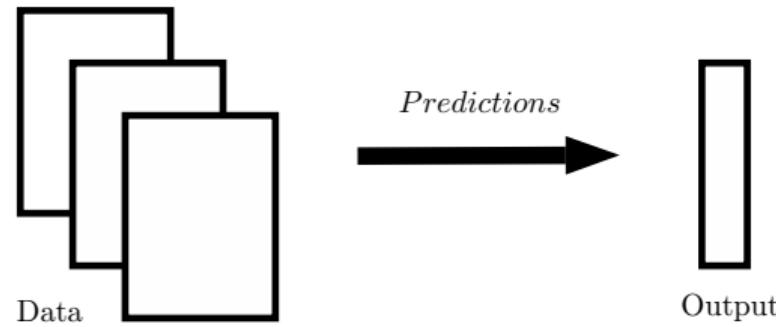
# DATA SCIENCE

- Drew Conway, 2010



@manudellavalle - <http://emanueledellavalle.org>

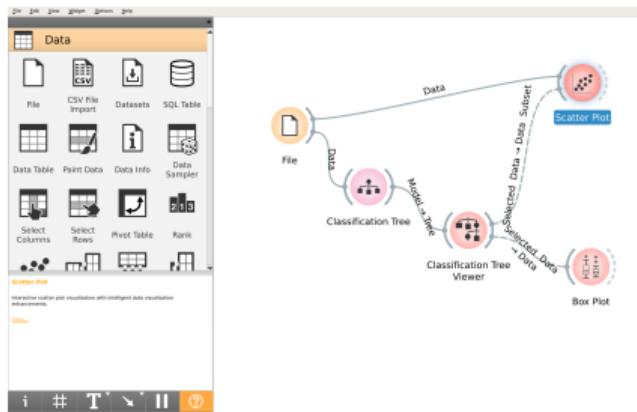
# THE OBJECTIVE



# PROGRAMMING TOOLS

## 1. Orange

<https://orange.biolab.si/>



A library featuring various ML algorithms designed to inter-operate with the Python numerical and scientific libraries e.g. NumPy, Pandas.

<https://scikit-learn.org/stable/>

## 2. Jupyter-Notebook (Anaconda)

<https://www.anaconda.com/>

```
#PCA
#CALCULATE the percentage of variance explained by each of the selected components.
explained_var=pd.DataFrame(pca2.explained_variance_ratio_.transpose())
explained_var

Out[12]:
0    0.367959
1    0.366921
2    0.112784
3    0.089429
4    0.034234
5    0.018437

In [13]:
#CALCULATE the cumulative percentage of explained variance
cum_explained_var=cumsum(explained_var)
pd.DataFrame(cum_explained_var).transpose()

Out[13]:
0    0.367959
1    0.75417
2    0.866903
3    0.947326
4    0.981563
5    1.0

In [14]:
%matplotlib inline
import seaborn as sns
ax = sns.barplot(x=range(explained_var.shape[0]), y=explained_var)
```

# AN EXAMPLE: BANK TELEMARKETING

Attribute	Type	Description/Values	
Personal	age	num	Age of the potential client
	job	cat	admin., blue- collar, entrepreneur, housemaid, ... ,unknown
	marital_status	cat	divorced, married, single, unknown
	education	cat	basic.4y, basic.6y, basic.9y, high.school, ... unknown
Bank	default	cat	The client has credit in default: no, yes, unknown
	housing	cat	The client has a housing loan contract: no, yes, unknown
	loan	cat	The client has a personal loan: no, yes, unknown
Campaign	contact	cat	Communication type: cellular, telephone
	month	cat	Last month contacted: jan, feb, ..., dec
	day_of_week	cat	Last contact day: mon, tue, ..., fri
	duration	num	Last contact duration (in seconds)
	campaign	num	Number of contacts performed during this campaign
	pdays	num	Number of days that passed by after last contact
	previous	num	Number of contacts performed before this campaign
	poutcome	cat	Outcome of the previous marketing campaign: failure, nonexistent, success
Economical	emp.var.rate	num	Employment variation rate in the last quarter
	cons.price.idx	num	Consumer price index in the last month
	cons.conf.idx	num	Monthly consumer confidence index
	euribor3m	num	Dayly Euro Interbank Offered Rate
	nr.employed	num	Number of employed citizens in the last quarter (thousands)
Target	success	target	0: no, 1: yes

<sup>1</sup> A data-driven approach to predict the success of bank telemarketing. S. Moro, P. Cortez, P. Rita. Decision Support Systems, 62:22-31, 2014.

# SUPERVISED LEARNING EXAMPLE

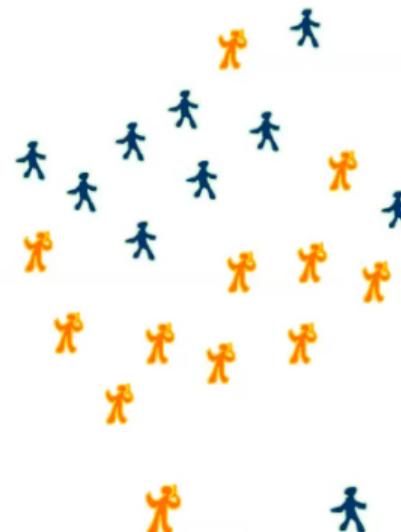
Observations/Clients

Features

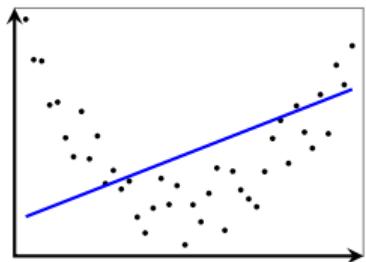
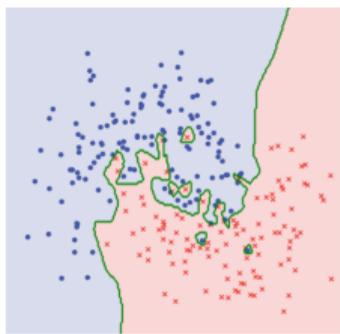
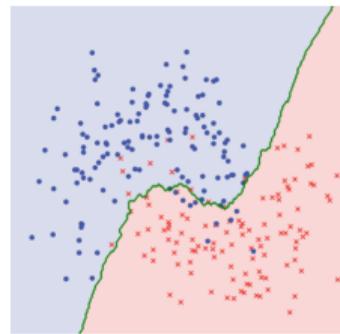
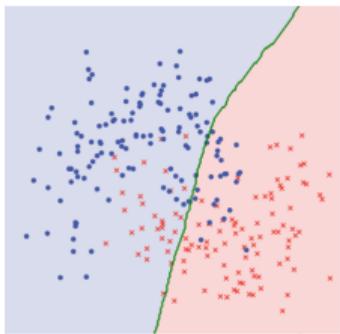
Target

			...			
	id	Age	Job	...	nr.employed	success
1	1	50	admin.	...	0.31	1
2	2	22	housemaid	...	0.23	0
3	3	34	unknown	...	0.16	1
...	...	...	...	...	...	...
$n$	$n$	20	blue-collar	...	0.65	1

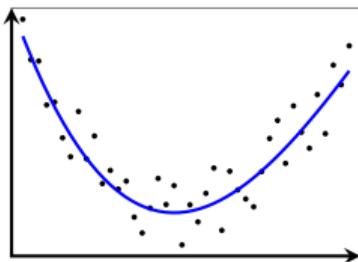
	id	Age	Job	...	nr.employed	success
	$n + 1$	30	blue-collar.	...	0.61	?
	$n + 22$	21	unknown	...	0.16	?
	$n + 33$	66	housemaid	...	0.45	?
	...	...	...	...	...	...
	$n + k$	28	blue-admin.	...	0.35	?



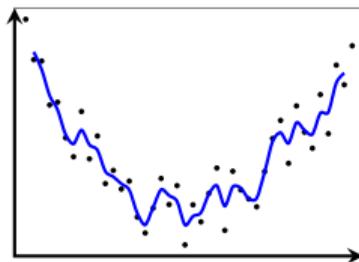
# SUPERVISED LEARNING GOAL



Underfitting



Balance

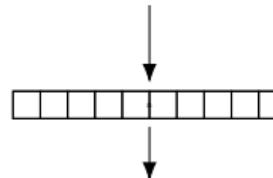


Overfitting

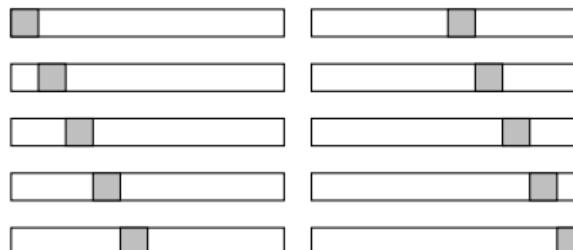
# HOLD OUT METHOD / CROSS VALIDATION



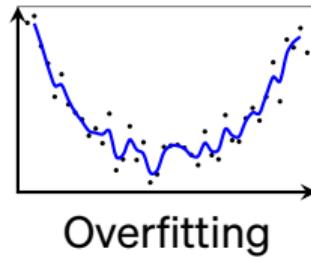
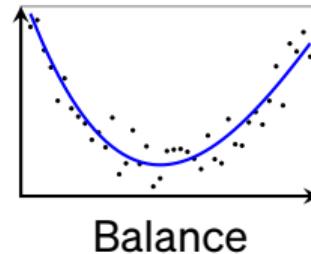
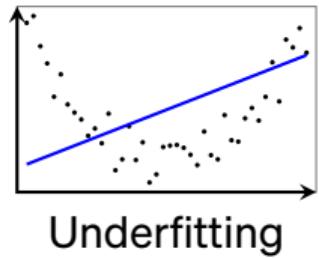
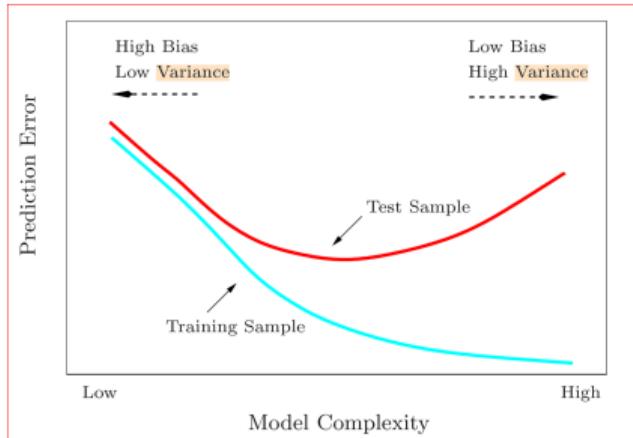
divide into 10 folds of equal size



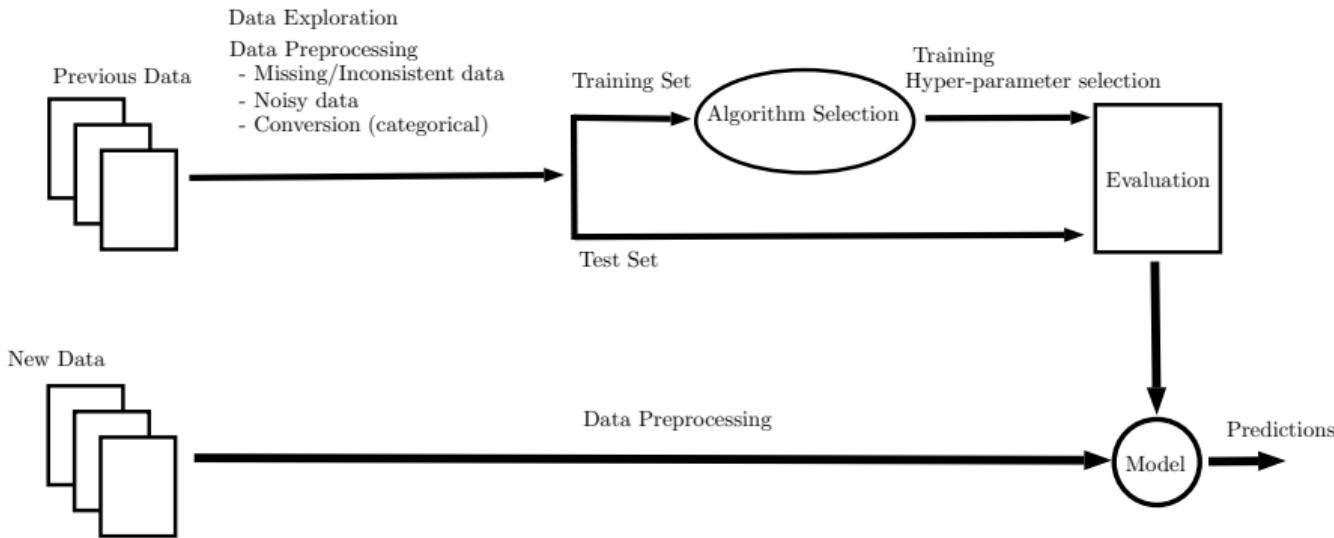
run experiments  
using 10 different partitionings



# UNDER/OVER-FITTING

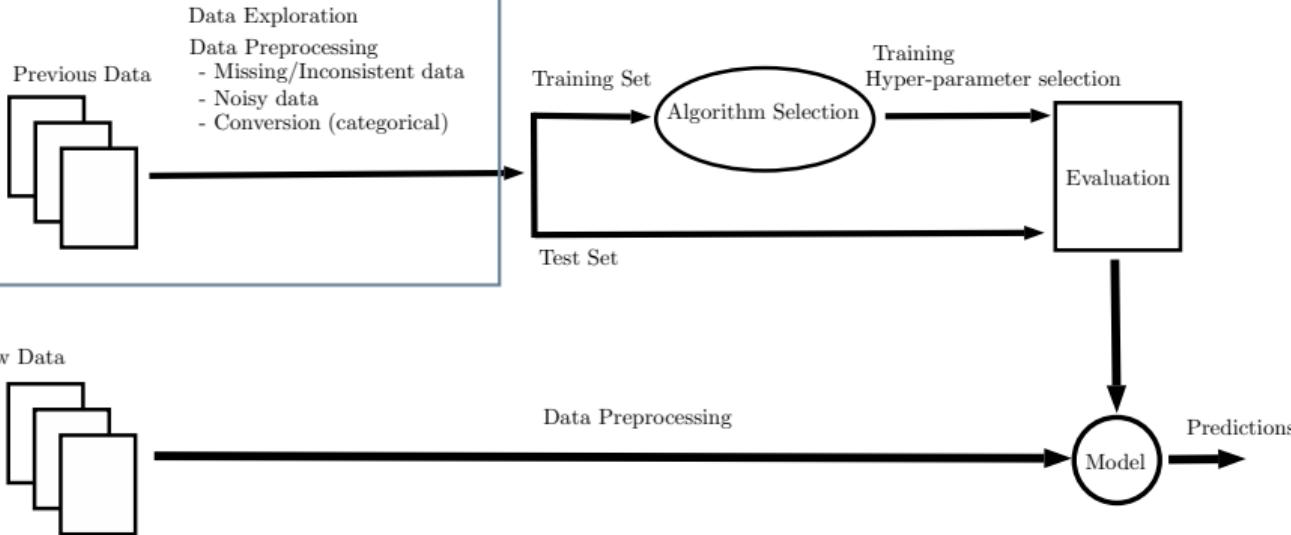


# WORKFLOW



# WORKFLOW

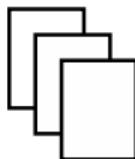
## 1. Data Exploration/Analysis



# WORKFLOW

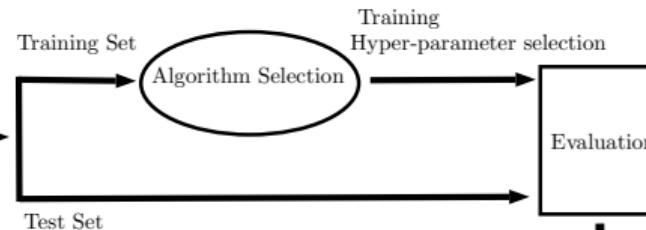
## 1. Data Exploration/Analysis

Previous Data

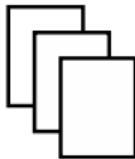


Data Exploration  
Data Preprocessing  
- Missing/Inconsistent data  
- Noisy data  
- Conversion (categorical)

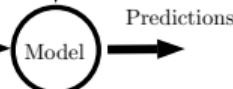
## 2. Model Creation



New Data



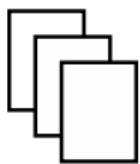
Data Preprocessing



# WORKFLOW

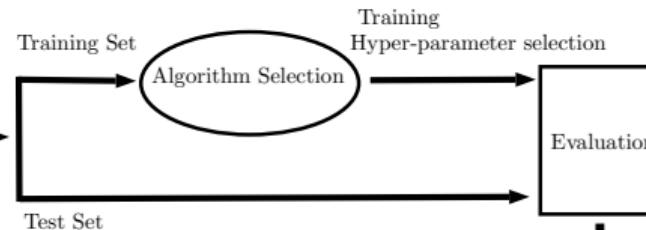
## 1. Data Exploration/Analysis

Previous Data

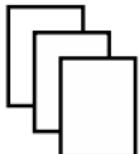


Data Exploration  
Data Preprocessing  
- Missing/Inconsistent data  
- Noisy data  
- Conversion (categorical)

## 2. Model Creation

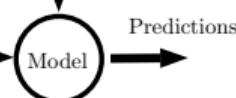


New Data



Data Preprocessing

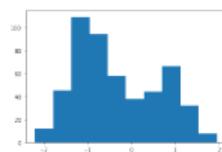
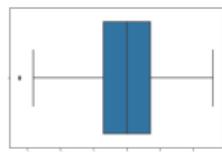
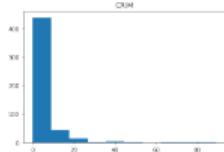
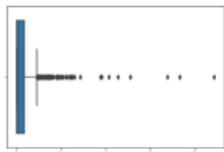
## 3. Predictions



# DATA EXPLORATION/ANALYSIS

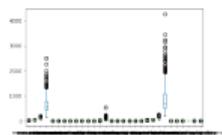
## 1. Data Validation

- Incomplete data (identify, drop, replace)
- Noisy data (Outliers)

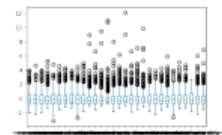


## 2. Data transformation

- Standardization
- Discretization
- Dummy variables
- Feature construction



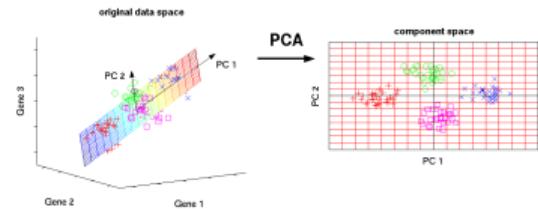
	type_of_food
0	fruit
1	vegetable
2	fruit
3	meat
4	fruit
5	vegetable



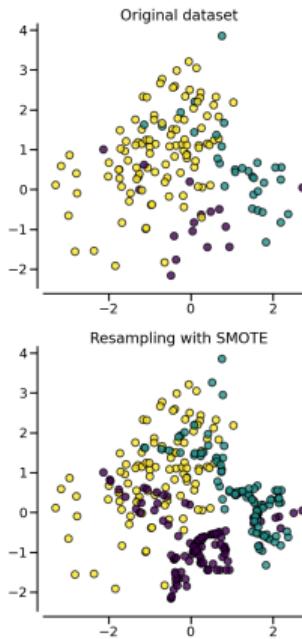
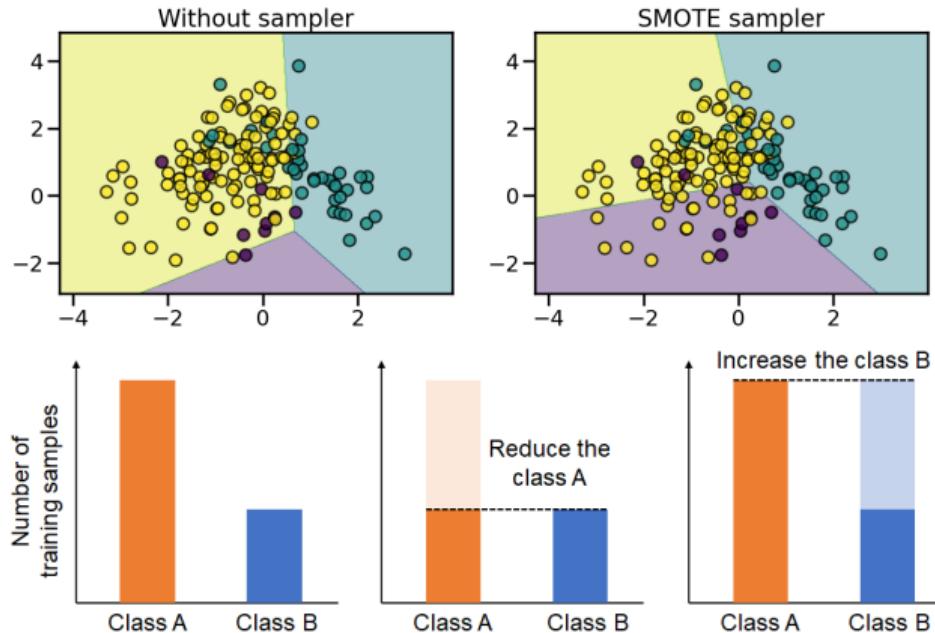
	fruit	meat	vegetable
0	1	0	0
1	0	0	1
2	1	0	0
3	0	1	0
4	1	0	0
5	0	0	1

## 3. Data reduction

- Sampling
- Discretization
- Feature/Dimensionality Reduction (PCA)



# DATA UNBALANCE



<https://imbalanced-learn.org/>

# FEATURE SELECTION

## 1. Filter Methods

- ▶ Model independent
- ▶ Simple and Fast
- ▶ Ex. Uni/Multi-variate correlation

## 2. Wrapper methods

- ▶ Model determine the variable set quality
- ▶ Slow
- ▶ Ex. Greedy Forward

## 3. Embedded Methods

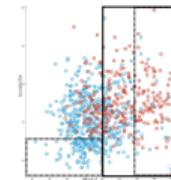
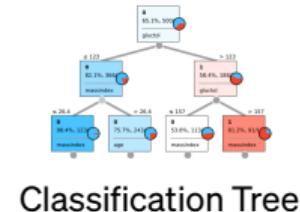
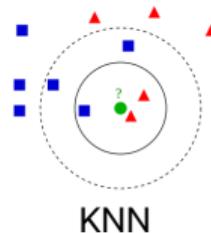
- ▶ Model determine the variable set
- ▶ Slow
- ▶ Ex. Tree models

# SUPERVISED LEARNING: CLASSIFICATION

# CLASSIFICATION MODELS

## ► Heuristics Methods

- Nearest Neighbours
- Classification Trees

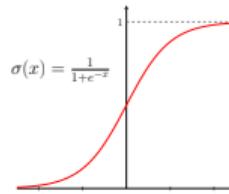


## ► Probabilistic Methods

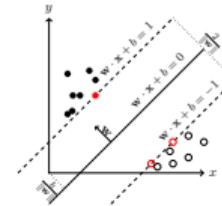
- Bayesian Methods

## ► Regression Methods

- Logistic regression

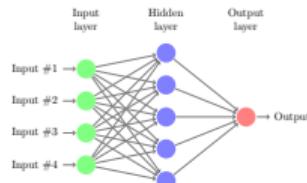


Logistic Regression



## ► Separation Methods

- Support vector machine
- Perceptron
- Neural Networks



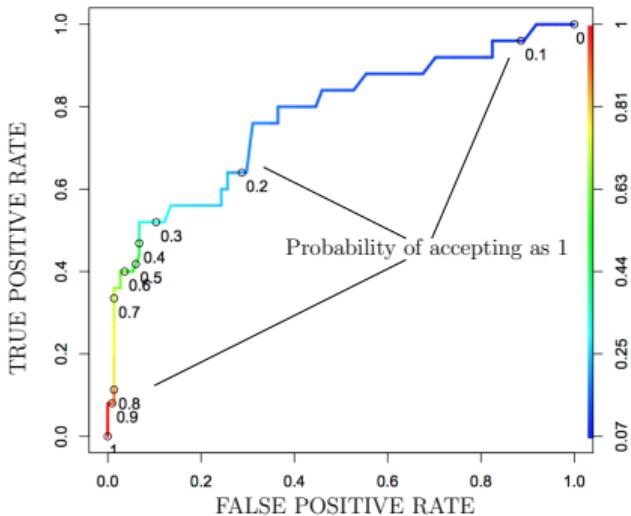
Neural Network

# QUALITY MEASURES - CONFUSION MATRIX

Prediction outcome			
		0	1
Actual value	0	True Negative	False Positive
	1	False Negative	True Positive

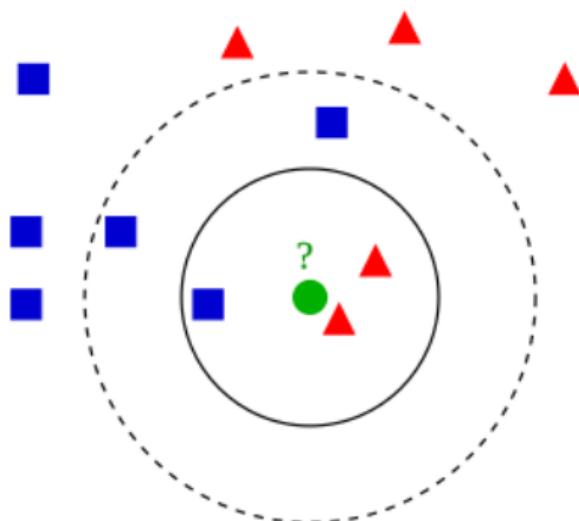
- ▶ Precision =  $\frac{TP}{TP+FP}$   
*``proportion of true positives among **positive predictions**''*
- ▶ False Positive rate =  $\frac{FP}{FP+TN}$   
*``proportion of false positives among **actual negatives**''*
- ▶ Recall (True Positive rate) =  $\frac{TP}{FN+TP}$   
*``proportion of true positives among **actual positive**''*
- ▶ F-score =  $(\beta^2 + 1) \frac{1}{\frac{\beta^2}{\text{recall}} + \frac{1}{\text{precision}}}$
- ▶ Geom. mean =  $\sqrt{\text{Precision} \times \text{Recall}}$

# QUALITY MEASURES - ROC CURVE & AUC



- ▶ If we accepting even with small probability then  $TPR = FPR = 1$
  - ▶ If we accepting just with high probability then  $TPR = FPR = 0$
  - ▶ The perfect classifier is the the point  $(0, 1)$
  - ▶  $AUC \in [0.5, 1]$  area under the curve is a quality measure of our algorithm.

# KNN K-NEAREST NEIGHBOURS



## Main Parameters

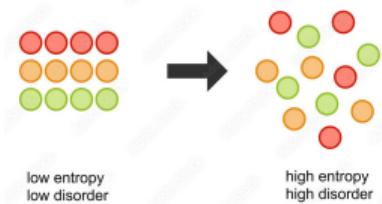
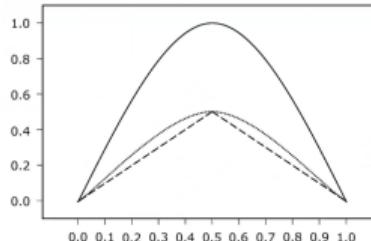
- ▶  $k$  : number of neighbours
- ▶ neighbour weights
- ▶ distances

# CLASSIFICATION TREE

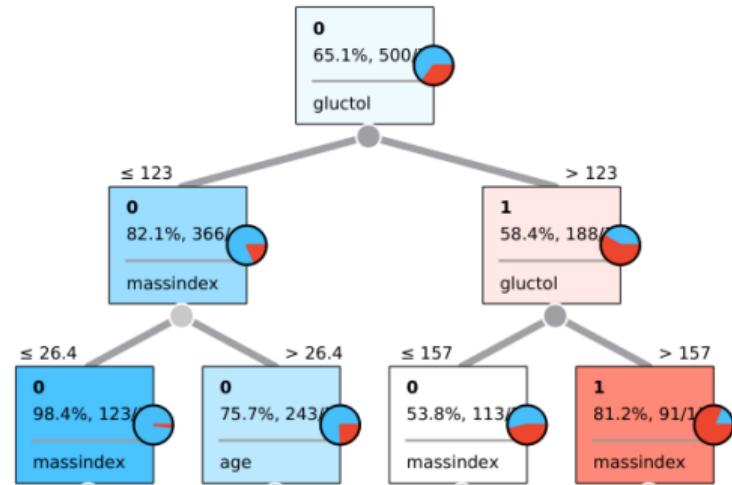
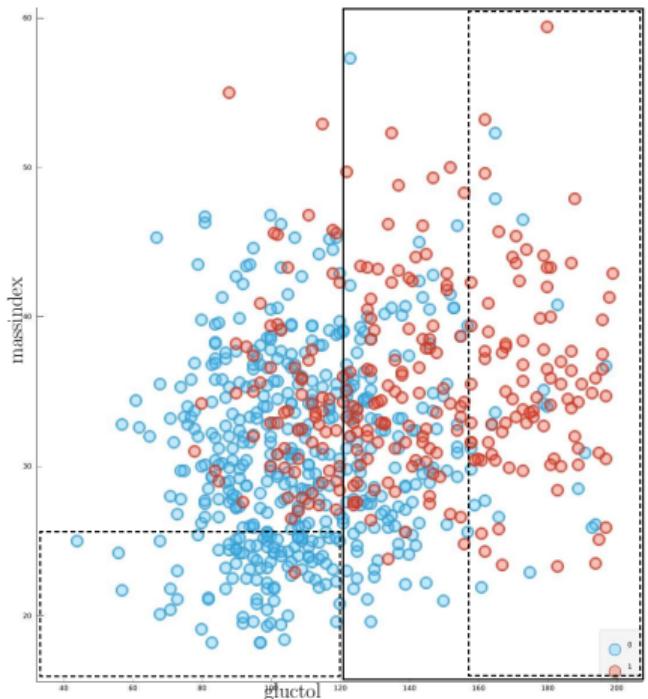


## Split criteria

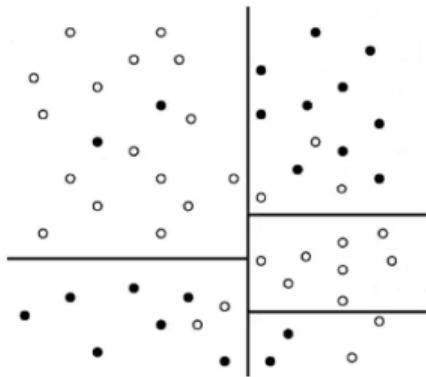
- ▶ Entropy index
- ▶ Gini index
- ▶ Miss-classification index



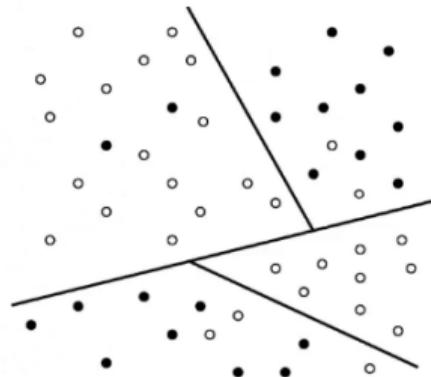
# CLASSIFICATION TREE



# UNI/MULTI-VARIATE CLASSIFICATION TREE



classification by an  
axis parallel tree



classification by an  
oblique tree

# NAIVE BAYESIAN CLASSIFIER

- ▶ Bayes Theorem ``**Predict future probabilities based on history**''

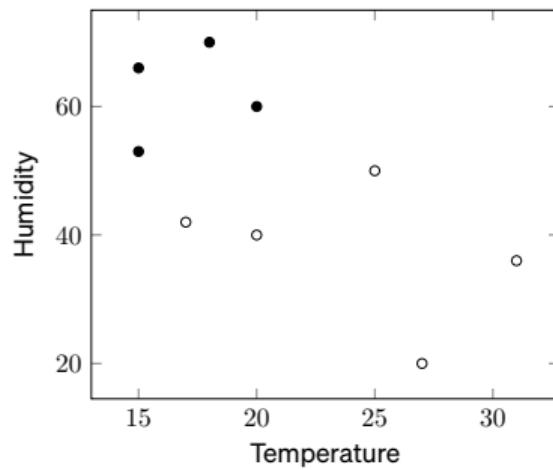
$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\sum_{l=1}^H P(\mathbf{x}|y)P(y)}$$

- ▶ Naive (Independence) - **Strong hypothesis**

$$P(\mathbf{x}|y) = P(x_1|y) \times P(x_2|y) \times \cdots \times P(x_n|y) = \prod_{j=1}^n P(x_j|y)$$

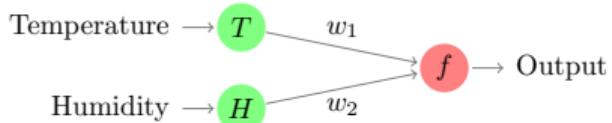
# LOGISTIC REGRESSION - EXAMPLE

Temp. [C]	20	31	15	18	27	15	20	25	17
Humidity [%]	60	36	53	70	20	66	40	50	42
Rain	1	-1	1	1	-1	1	-1	-1	-1



# LOGISTIC REGRESSION - EXAMPLE

Temp. [C]	20	31	15	18	27	15	20	25	17
Humidity [%]	60	36	53	70	20	66	40	50	42
Rain	1	-1	1	1	-1	1	-1	-1	-1

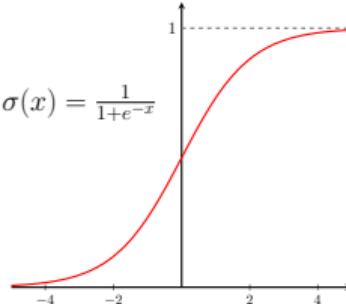
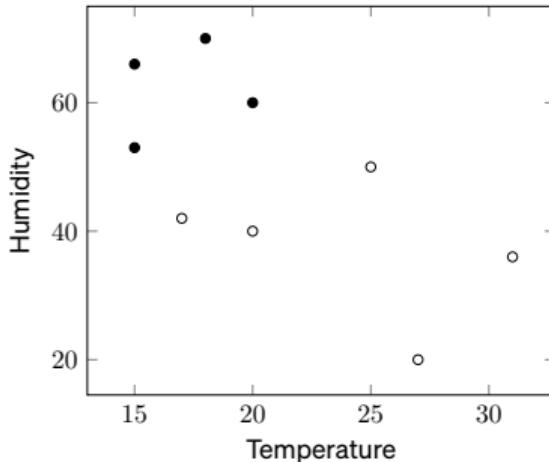


$$P(y = 1|T, H; w) = \sigma(w_0 + w_1 \cdot T + w_2 \cdot H)$$

sigmoid function

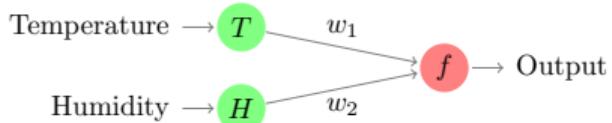
**Maximize Likelihood**

$$\max_w L(w | y; x) = \Pr(Y | X; w) = \prod_i \Pr(y_i | x_i; w)$$



# LOGISTIC REGRESSION - EXAMPLE

Temp. [C]	20	31	15	18	27	15	20	25	17
Humidity [%]	60	36	53	70	20	66	40	50	42
Rain	1	-1	1	1	-1	1	-1	-1	-1



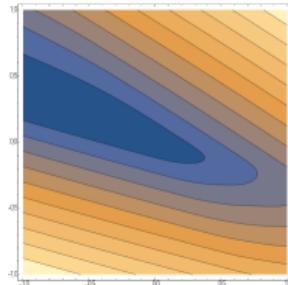
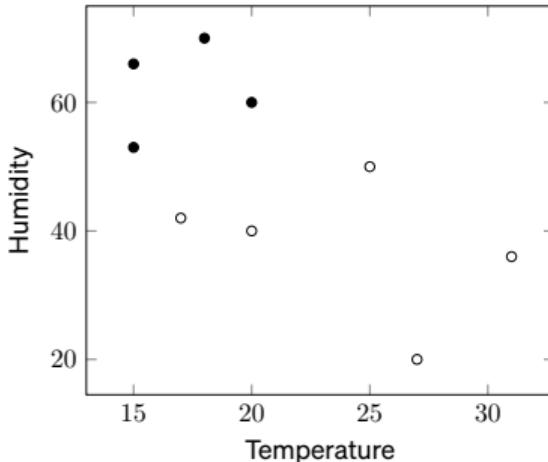
$$P(y = 1|T, H; w) = \sigma(w_0 + w_1 \cdot T + w_2 \cdot H)$$

sigmoid function

## Maximize Likelihood

$$\max_w L(w | y; x) = \Pr(Y | X; w) = \prod_i \Pr(y_i | x_i; w)$$

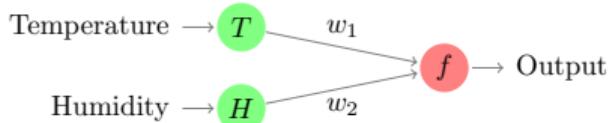
$$P(\text{rain}) = \sigma(-0.044 - 0.633 \times T + 0.235 \times H)$$



$$w^* = (-0.044, -0.633, 0.235)$$

# LOGISTIC REGRESSION - EXAMPLE

Temp. [C]	20	31	15	18	27	15	20	25	17
Humidity [%]	60	36	53	70	20	66	40	50	42
Rain	1	-1	1	1	-1	1	-1	-1	-1



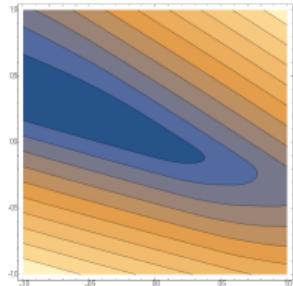
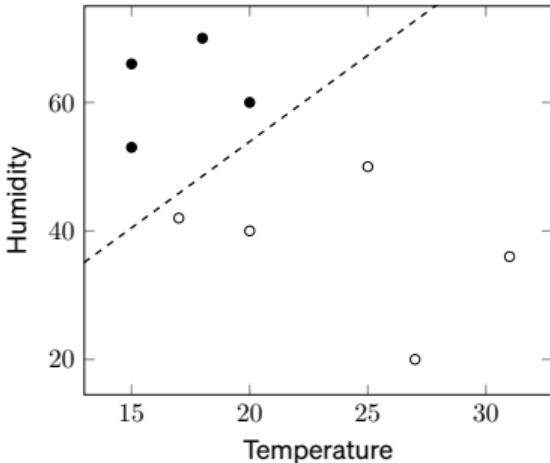
$$P(y = 1|T, H; w) = \sigma(w_0 + w_1 \cdot T + w_2 \cdot H)$$

sigmoid function

**Maximize Likelihood**

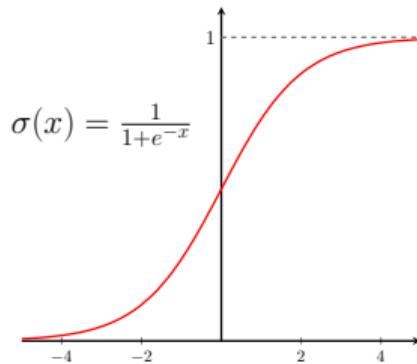
$$\max_w L(w | y; x) = \Pr(Y | X; w) = \prod_i \Pr(y_i | x_i; w)$$

$$P(\text{rain}) = \sigma(-0.044 - 0.633 \times T + 0.235 \times H)$$



$$w^* = (-0.044, -0.633, 0.235)$$

# LOGISTIC REGRESSION



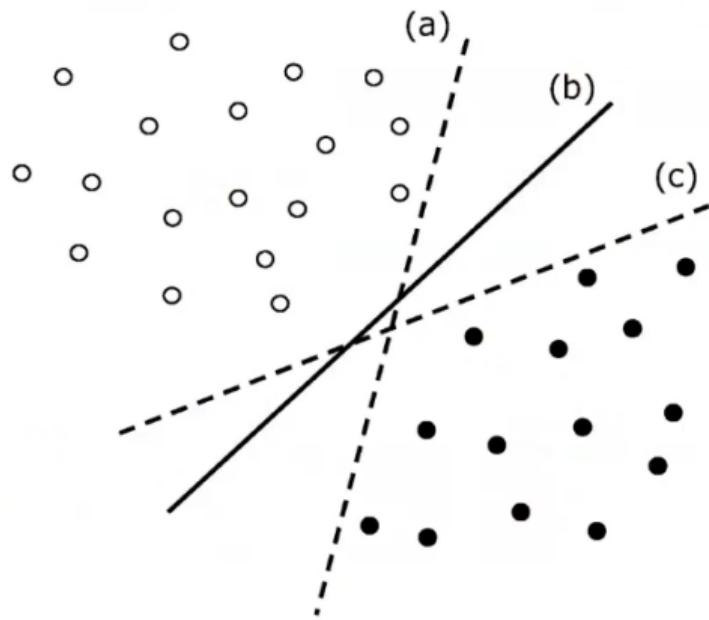
$$\min_w \underbrace{\frac{1}{2} ||w||^2}_{\text{regularization}} + C \underbrace{\sum_{i=1}^n \log(1 + \exp(-y_i (w^T x_i)))}_{\text{error}}$$

## Main Parameters

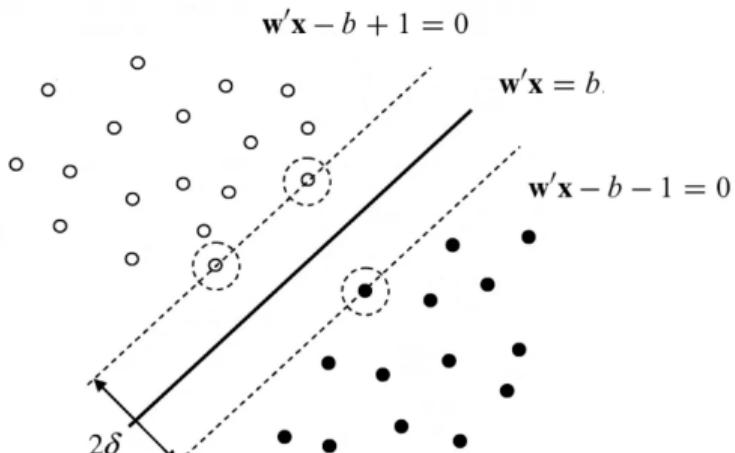
- ▶  $C$ : Inverse of regularization strength

$$P(y = 1|x; w) = \sigma(w_0 + w_1 x_1 + \dots + w_n x_n)$$

# SUPPORT VECTOR MACHINE - LINEARLY SEPARABLE



# SVM - LINEARLY SEPARABLE



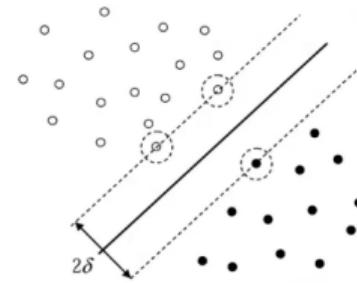
$$\delta = \frac{2}{\|w\|},$$

$$\|w\| = \sqrt{\sum_{j \in \mathcal{N}} w_j^2}$$

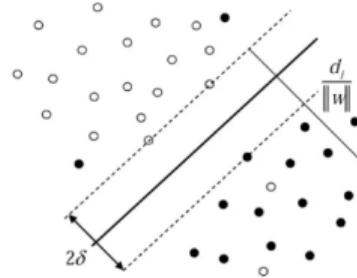
$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|_2^2 \\ \text{s. t.} \quad & y_i(w'x_i - b) \geq 1 \quad i \in \mathcal{M} \end{aligned}$$

# SVM - GENERAL CASE

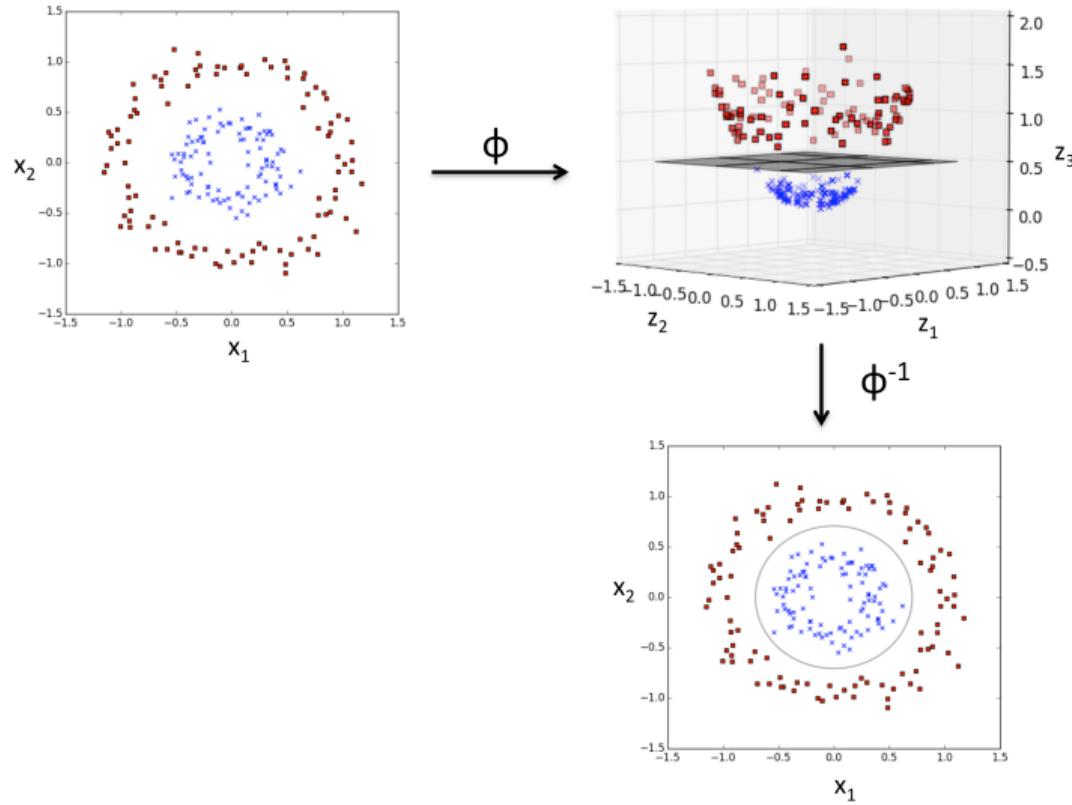
$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 \\ \text{s. t.} \quad & y_i(\mathbf{w}' \mathbf{x}_i - b) \geq 1 \quad i \in \mathcal{M} \end{aligned}$$



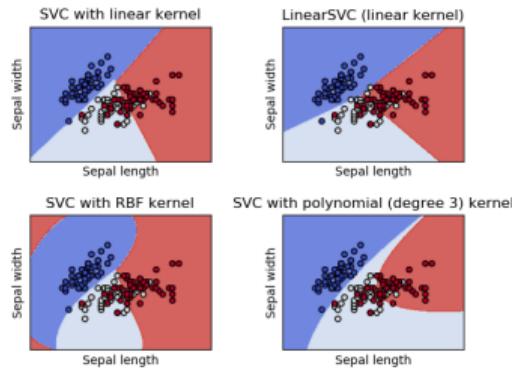
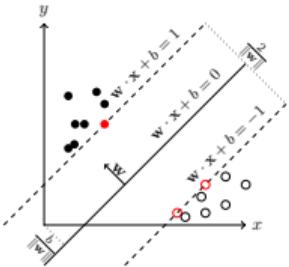
$$\begin{aligned} \min_{\mathbf{w}, b, d} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^m d_i \\ \text{s. t.} \quad & y_i(\mathbf{w}' \mathbf{x}_i - b) \geq 1 - d_i \quad i \in \mathcal{M} \\ & d_i \geq 0 \quad i \in \mathcal{M} \end{aligned}$$



# SVM - KERNELS



# SVM - GENERAL CASE



$$\min_{w, b, d} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m d_i$$

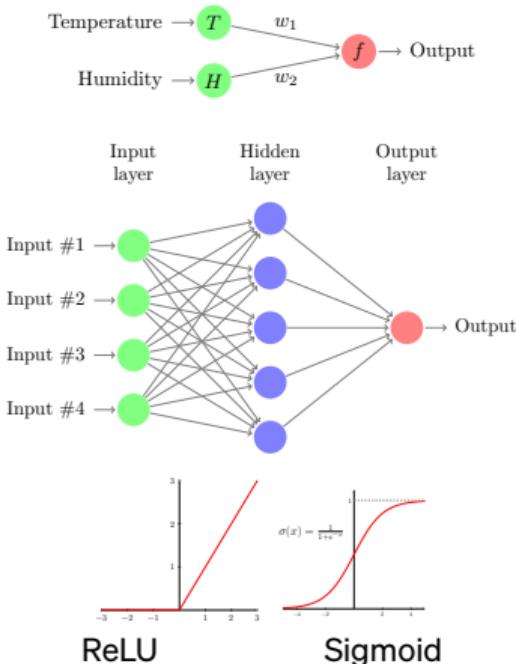
subject to  $y_i (\underbrace{w^T \phi(x_i) - b}_{\text{kernel}}) \geq 1 - d_i$ ,

$$d_i \geq 0$$

## Main Parameters

- ▶  $C$ : Inverse of regularization strength
- ▶ kernel:
  - linear:  $x'x$
  - poly:  $(\gamma x'x + r)^d$
  - rbf:  $\exp(-\gamma \|x - x'\|^2)$
  - sigmoid:  $\tanh(\gamma x'x + r)$

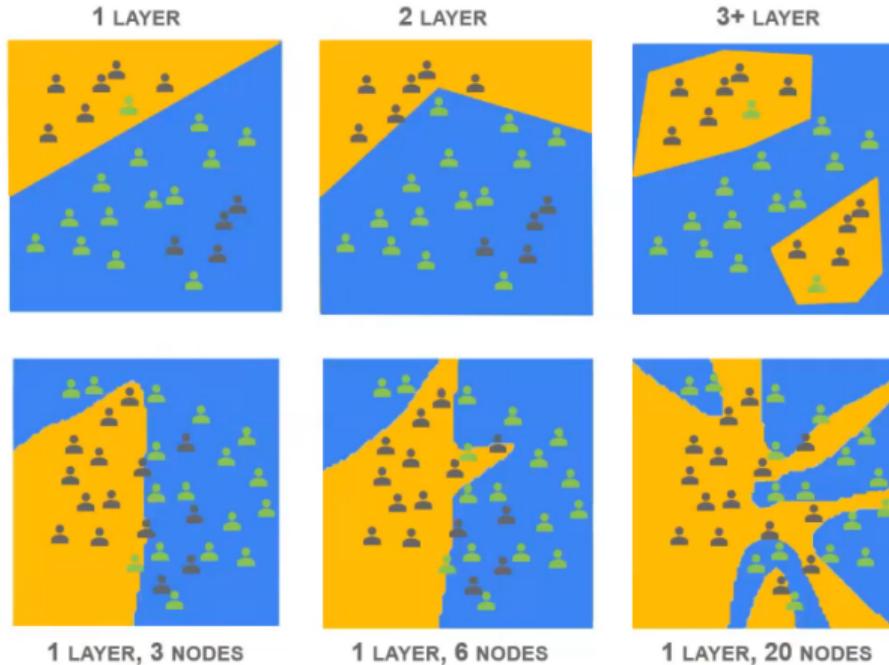
# MULTI-LAYER PERCEPTRON



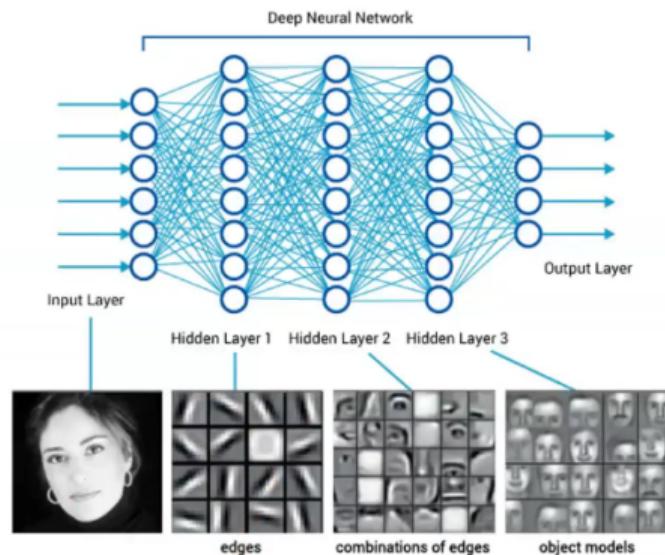
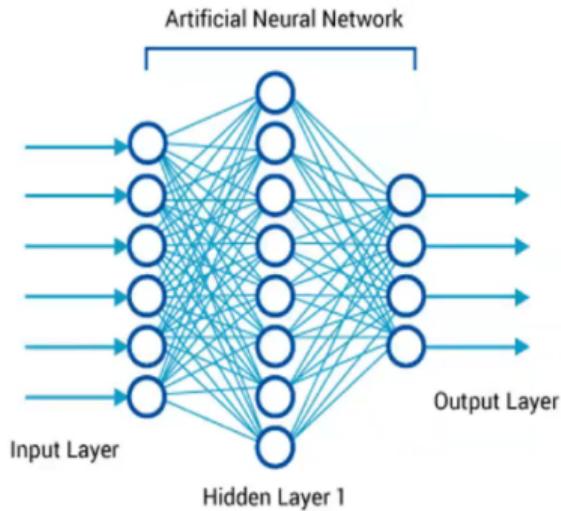
## Main Parameters

- ▶ `hidden_layer_sizes`:  $(n_1, n_2, \dots, n_L)$
- ▶ `activation`: identity, logistic, tanh, relu
- ▶ `alpha` regularization term parameter
- ▶ Resolution algorithm parameters: `tol`, `max_iter`.

# NEURAL NETWORK



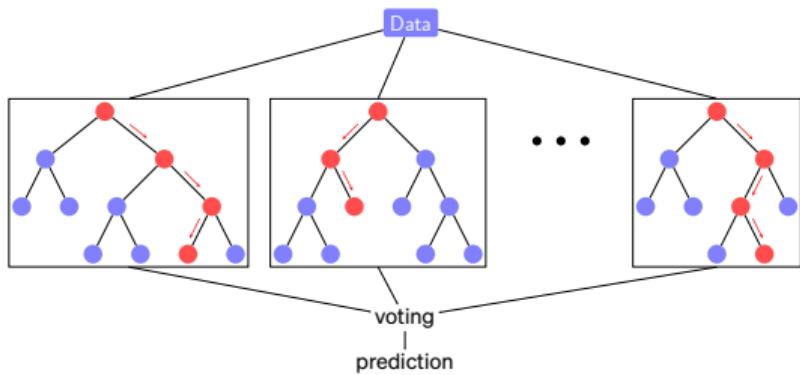
# DEEP LEARNING



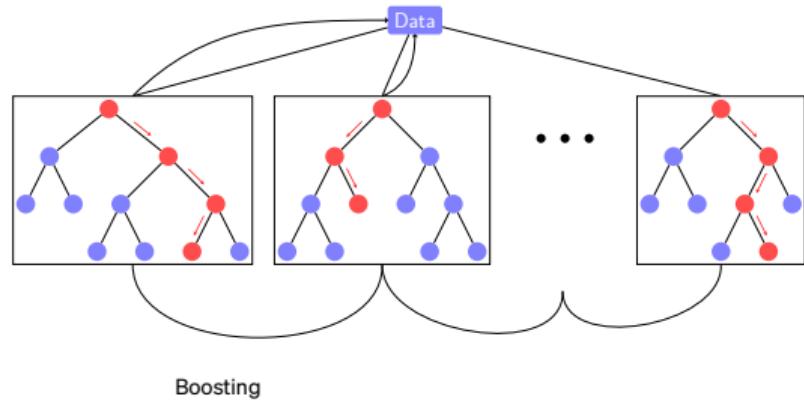
Try yourself: [https://adamharley.com/nn\\_vis/](https://adamharley.com/nn_vis/)

# ENSEMBLE METHODS

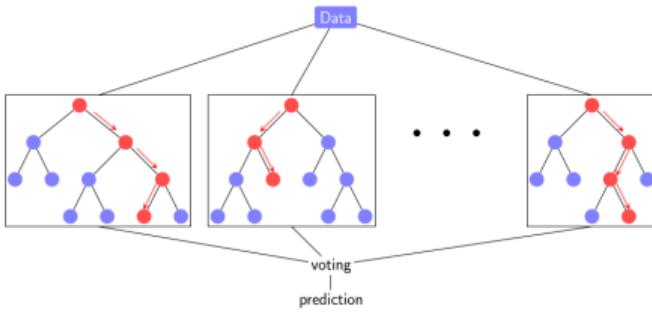
Bagging



Boosting



# RANDOM FOREST

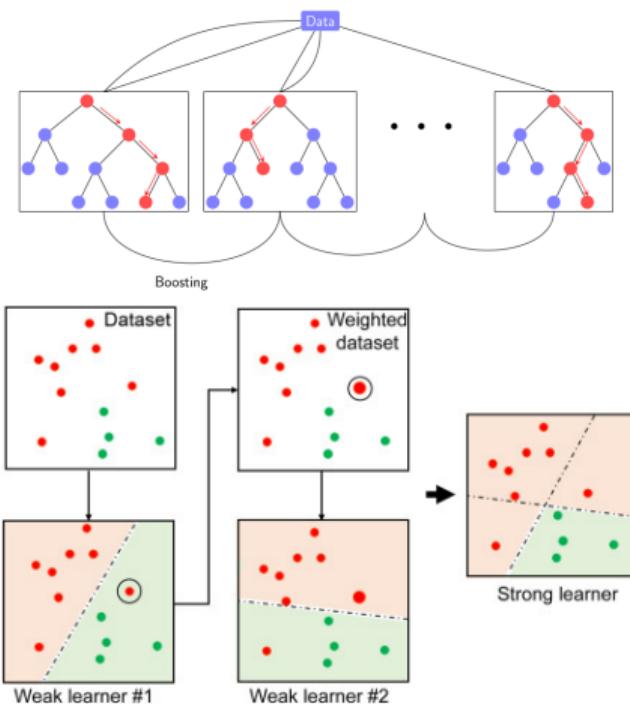


1. Create different (simple) tree models (stumps)
2. Each model is created with a subset of observation/features ( $\sim 2m/3$ )
3. We combine the prediction of all trees

## Main Parameters

- ▶ `n_estimators`: Number of trees
- ▶ `max_features`: Number of features selected for the split
- ▶ `bootstrap=False`: Use all samples
- ▶ Tree parameters

# ADABOOST



1. Assign equal weights to observations
2. For  $k = 1, \dots, K$ 
  - Select a sample of observations based on the weights.
  - Create the  $k$ -th weak learner and compute predictions  $x^{(k)}$
  - Compute the model **weighted** error and assign its coefficient according to its error.
  - Update sample weights
3. Final weighted prediction

## Main Parameters

- ▶ `n_estimators`: Number of estimators ( $K$ )
- ▶ `base_estimator`: Weak estimator type
- ▶ `learning_rate`: weights of estimator in final decision ( $\lambda$ )

# BANK TELEMARKETING RESULTS

Model	AUC	CA	F1	Precision	Recall
AdaBoost	1.000	1.000	1.000	1.000	1.000
kNN	1.000	1.000	1.000	1.000	1.000
Logistic Regression	0.937	0.875	0.712	0.756	0.672
Naive Bayes	0.834	0.791	0.599	0.535	0.681
Neural Network	0.956	0.899	0.793	0.747	0.844
Random Forest	0.969	0.915	0.808	0.844	0.775
SVM	0.932	0.852	0.630	0.738	0.549
Tree	0.945	0.884	0.750	0.744	0.757

Model	AUC	CA	F1	Precision	Recall
AdaBoost	0.785	0.846	0.668	0.663	0.673
kNN	0.741	0.825	0.607	0.630	0.586
Logistic Regression	0.938	0.877	0.714	0.767	0.668
Naive Bayes	0.847	0.807	0.627	0.566	0.703
Neural Network	0.935	0.879	0.748	0.716	0.784
Random Forest	0.937	0.877	0.720	0.758	0.686
SVM	0.931	0.854	0.636	0.745	0.554
Tree	0.932	0.880	0.740	0.739	0.741

# MODEL EXPLAINABILITY



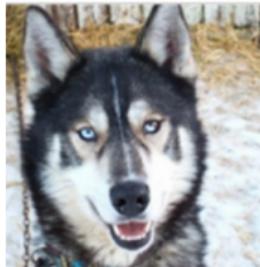
Predicted: **wolf**  
True: **wolf**



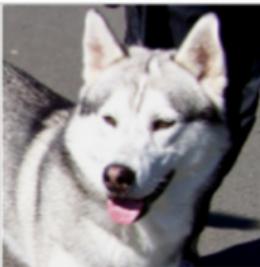
Predicted: **husky**  
True: **husky**



Predicted: **wolf**  
True: **wolf**



Predicted: **wolf**  
True: **husky**



Predicted: **husky**  
True: **husky**



Predicted: **wolf**  
True: **wolf**

# MODEL EXPLAINABILITY



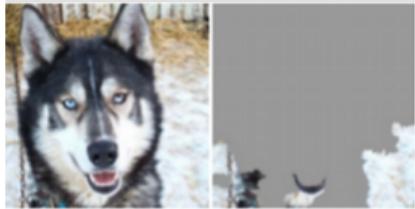
Predicted: **wolf**  
True: **wolf**



Predicted: **husky**  
True: **husky**



Predicted: **wolf**  
True: **wolf**



Predicted: **wolf**  
True: **husky**



Predicted: **husky**  
True: **husky**

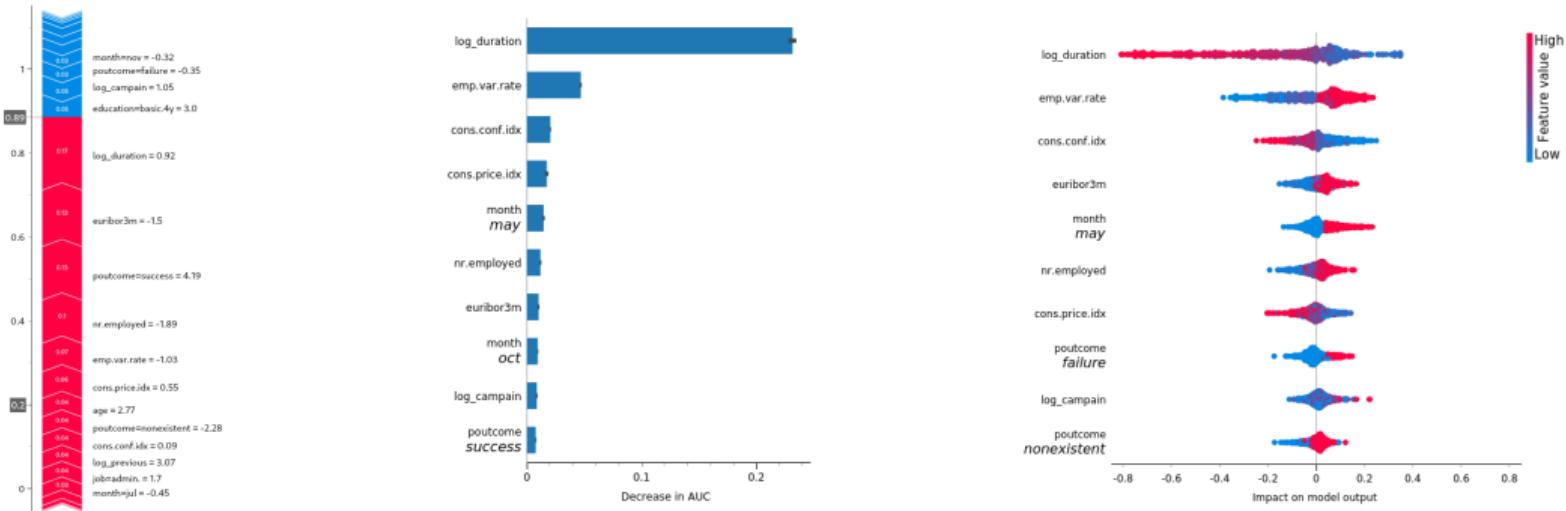


Predicted: **wolf**  
True: **wolf**

# MODEL EXPLAINABILITY

## SHAP: SHapley Additive exPlanation

`` Marginal contribution of the feature in the prediction''



$$\phi_A = \sum_{S \subseteq F \setminus \{A\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f(S \cup \{A\}) - f(S)]$$

# CASE STUDY - HEALTH INSURANCE CROSS SELLING

Variable	Definition
id	Unique ID for the customer
Gender	Gender of the customer
Age	Age of the customer
Driving_License	0 : Customer does not have DL, 1: Customer already has DL
Region_Code	Unique code for the region of the customer
Previously_Insured	1: Customer already has Vehicle Insurance, 0 : Customer doesn't have Vehicle Insurance
Vehicle_Age	Age of the Vehicle
Vehicle_Damage	1: Customer got his/her vehicle damaged in the past. 0 : Customer didn't get his/her vehicle damaged in the past.
Annual_Premium	The amount customer needs to pay as premium in the year
PolicySalesChannel	Anonymized Code for the channel of outreaching to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.
Vintage	Number of Days, Customer has been associated with the company
Response	<b>1: Customer is interested, 0 : Customer is not interested</b>

# CASE STUDY - INSURANCE FRAUD

- ▶ months\_as\_customer
- ▶ insured\_relationship
- ▶ age
- ▶ capital-gains
- ▶ policy\_number
- ▶ capital-loss
- ▶ policy\_bind\_date
- ▶ incident\_date
- ▶ policy\_state
- ▶ incident\_type
- ▶ policy\_csl
- ▶ collision\_type
- ▶ policy\_deductable
- ▶ incident\_severity
- ▶ policy\_annual\_premium
- ▶ authorities\_contacted
- ▶ umbrella\_limit
- ▶ incident\_state
- ▶ insured\_zip
- ▶ incident\_city
- ▶ insured\_sex
- ▶ incident\_location
- ▶ insured\_education\_level
- ▶ incident\_hour\_of\_the\_day
- ▶ insured\_occupation
- ▶ number\_of\_vehicles\_involved
- ▶ insured\_hobbies
- ▶ property\_damage
- ▶ bodily\_injuries
- ▶ witnesses
- ▶ police\_report\_available
- ▶ total\_claim\_amount
- ▶ injury\_claim
- ▶ property\_claim
- ▶ vehicle\_claim
- ▶ auto\_make
- ▶ auto\_model
- ▶ auto\_year
- ▶ fraud\_reported
- ▶ \_c39

# ASSIGNMENT: THE \$50K/YR BASED ON CENSUS DATA

The prediction task is to determine whether a person makes over \$50K a year from the data in 1994 Census bureau database.

- ▶ age: continuous.
- ▶ workclass: Private, Self-emp-not-inc, Federal-gov, etc.
- ▶ final weight<sup>a</sup>: continuous.
- ▶ education level: Bachelors, Some-college, 11th, etc.
- ▶ education-num: continuous.
- ▶ marital-status: Married-civ-spouse, Divorced, etc.
- ▶ occupation: Tech-support, Craft-repair,etc.
- ▶ relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- ▶ ethnic : White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- ▶ gender: Female, Male.
- ▶ capital-gain/capital-loss: continuous.
- ▶ hours-per-week: continuous.
- ▶ origin-country: United-States, Cambodia, England, etc.

---

<sup>a</sup>The weights on the Current Population Survey (CPS) files are controlled to independent estimates of the civilian noninstitutional population of the US.

Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996.

# ASSIGNMENT: THE \$50K/YR BASED ON CENSUS DATA

- ▶ `adult_data_model.csv`: il dataset contiene le informazioni di circa 25000 osservazioni con le rispettive variabili esplicative. Dovete usare questi dati per creare e valutare il vostro modello.
- ▶ `adult_data_future_with_target.csv`: il dataset contiene le informazioni di 7000 records. Vi viene richiesto di fornire le previsioni per questo set di record.

L'obiettivo è quello di mazimizzare la qualità delle previsioni del modello, misurata in base alla metrica F1.

# SUPERVISED LEARNING: REGRESSION

# CASE STUDY - HOUSE PRICING

Variable	Definition
CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per \$10,000
PTRATIO	pupil-teacher ratio by town
B	$1000(Bk - 0.63)2$ where $Bk$ is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in \$1000's

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', *J. Environ. Economics & Management*, vol.5, 81-102, 1978.

# QUALITY MEASURES - REGRESSION

- ▶ Coefficient of determination

$$R^2 = 1 - \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$$

- ▶ Mean Absolute Error :

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- ▶ Mean Squared Error :

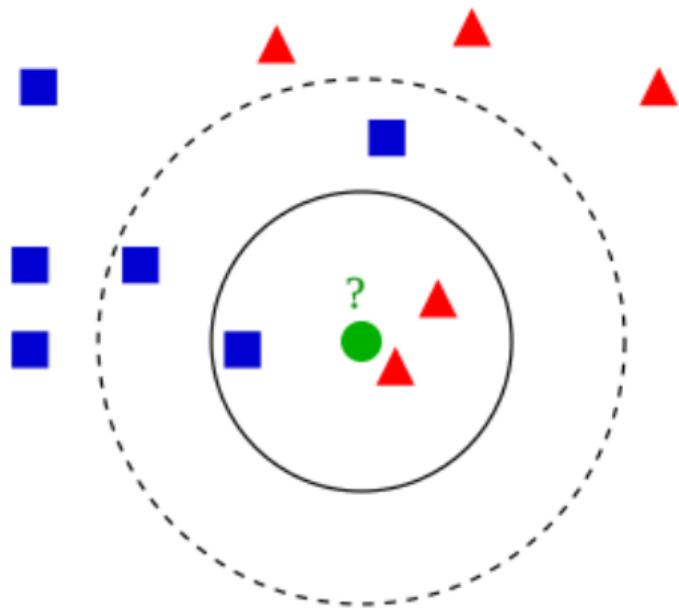
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- ▶ Root Mean Squared Error :  $RMSE = \sqrt{MSE}$

# REGRESSION MODELS

- ▶ Heuristics Methods
  - Nearest Neighbours
  - Regression Trees
- ▶ Optimization based Methods
  - Linear models
  - Support vector machine
  - Neural Networks

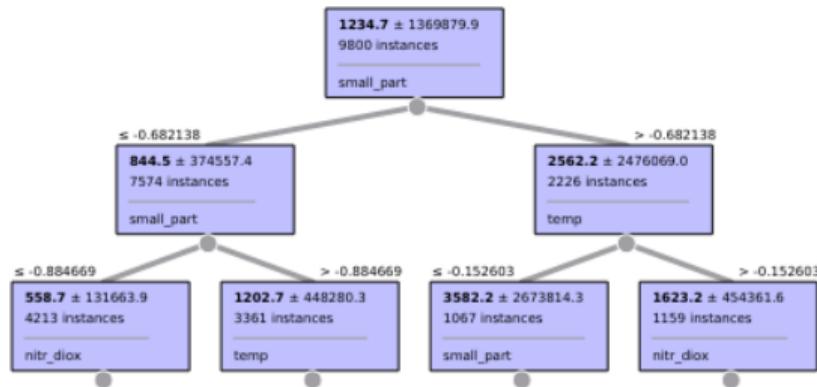
# KNN K-NEAREST NEIGHBOURS



## Main Parameters

- ▶  $k$  : number of neighbours
- ▶ neighbour weights
- ▶ distances

# REGRESSION TREE



## Main Parameters

- ▶ variability measure: mse (variance from mean), mae (error from median)
- ▶ max\_depth
- ▶ min\_samples\_split: minimum number of samples to split an internal node
- ▶ min\_sample\_leaf: minimum number of samples required to be at a leaf node

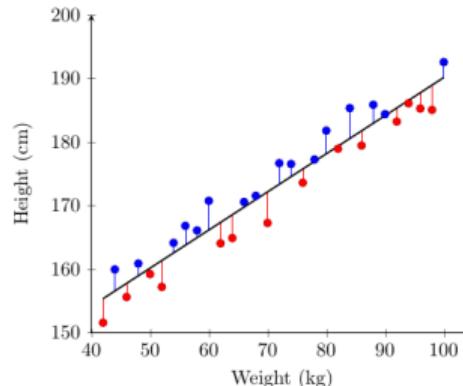
# SIMPLE LINEAR REGRESSION



$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

## ► Least square regression

$$SSE = \sum_{i=1}^m e_i^2 = \sum_{i=1}^m [y_i - (w\mathbf{x}_i + b)]^2$$



# LINEAR MODELS REGULARIZATION

► Ridge:

$$\min_w \lambda ||w||^2 + ||e||^2 = \min_w \lambda ||w||^2 + (y - Xw)^\top (y - Xw)$$

► Lasso:

$$\min_w \lambda |w| + ||e||^2 = \min_w \lambda |w| + (y - Xw)^\top (y - Xw)$$

# GENERAL LINEAR MODELS

- ▶ We consider a set of bases functions: polynomials, kernels, etc.

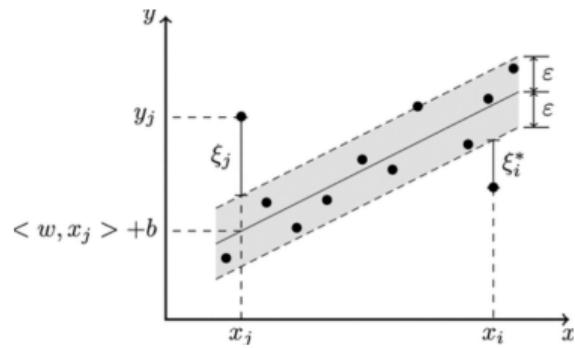
$$Y = \sum_h w_h g_h(X_1, X_2, \dots, X_n) + b$$

- ▶ For example,
  - **quadratic**

$$Y = wX + b = X_1 w_1 + X_2 w_2 + X_1^2 w_3 + X_2^2 w_4 + X_1 X_2 w_5 + b$$

- **exponential**

$$\log Y = b + wX \quad \Rightarrow \quad Y = e^{b+wX}$$



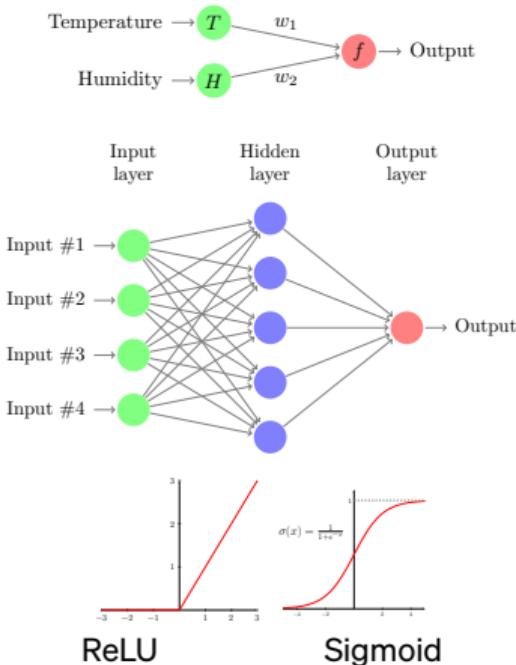
$$\min_{w, b, \zeta, \zeta^*} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\zeta_i + \zeta_i^*)$$

subject to  $y_i - w^T \phi(x_i) - b \leq \varepsilon + \zeta_i$ ,  
 $w^T \phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*$ ,  
 $\zeta_i, \zeta_i^* \geq 0, i = 1, \dots, n$

## Main Parameters

- ▶  $C$ : inverse of regularization strength
- ▶  $\varepsilon$ : tolerance
- ▶ kernel
- ▶ Resolution algorithm parameters

# MULTI-LAYER PERCEPTRON

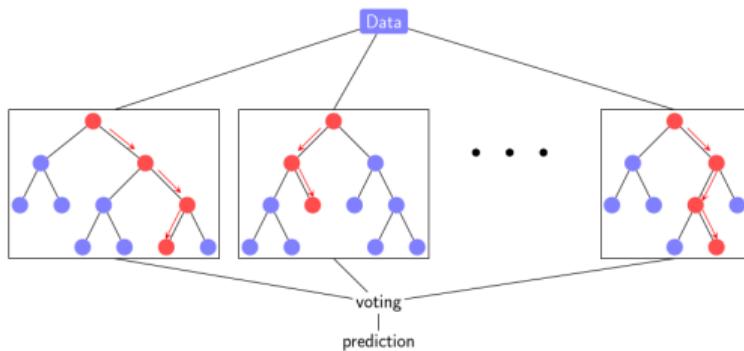


## Main Parameters

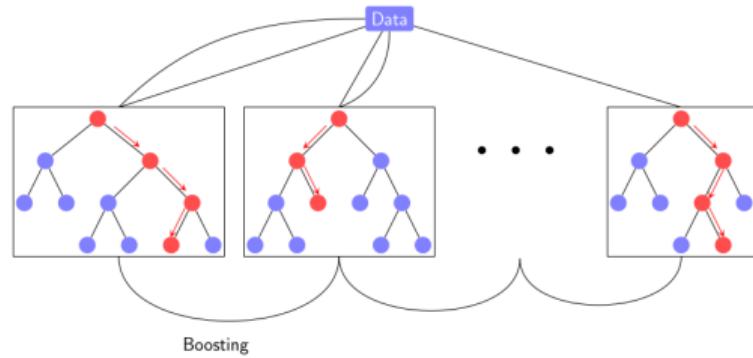
- ▶ `hidden_layer_sizes`:  $(n_1, n_2, \dots, n_L)$
- ▶ `activation`: identity, logistic, tanh, relu
- ▶ `alpha` regularization term parameter
- ▶ Resolution algorithm parameters: `solver`, `tol`, `batch_size`, `learning_rate`, `max_iter`.

# ENSEMBLE METHODS

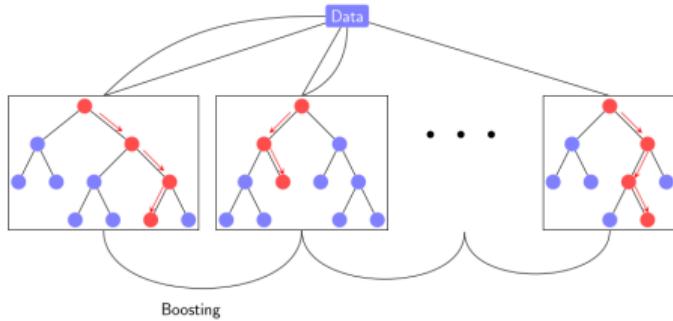
Bagging



Boosting



# GRADIENT BOOST



1. Train a weak learner  $F_0$  and compute predictions  $x^{(k)}$
2. For  $k = 1, \dots, K$ 
  - Compute the difference between the target  $y$  and the predictions of the current learner

$$\hat{y}_{k-1} = F_{k-1}(x_i)$$

- Train a weak learner that approximate this difference (error)

$$f_k = \arg \min_f L_m = \arg \min_f \sum_{i=1}^n l(y_i, F_{m-1}(x_i) + f(x_i))$$

- $F_k = F_{k-1} + \lambda f_k$

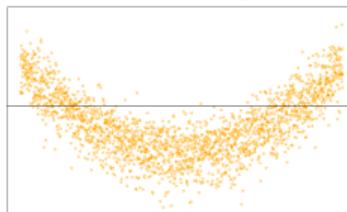
## Main Parameters

- ▶ `n_estimators`: Number of estimators ( $K$ )
- ▶ `base_estimator`: Weak estimator type
- ▶ `learning_rate`: weights of estimator in final decision ( $\lambda$ )

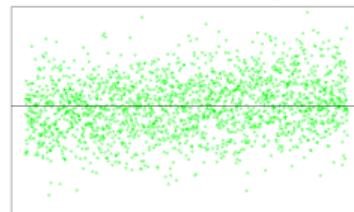
# RESIDUAL ASSUMPTIONS

Independence,  $E(\varepsilon_i | \mathbf{x}_i) = 0$ ,  $Var(\varepsilon_i | \mathbf{x}_i) = \sigma^2$

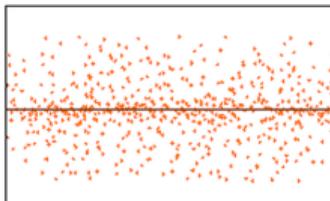
Pattern in Relationship



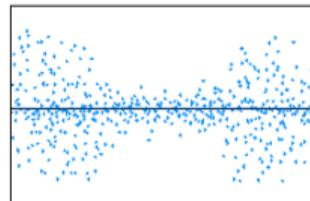
No Pattern in Relationship



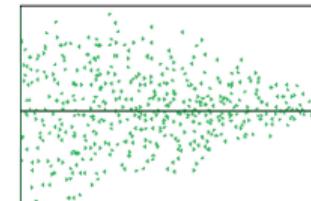
Homoscedasticity



Heteroscedasticity



Heteroscedasticity



Random Cloud (No Discernible Pattern)

Bow Tie Shape (Pattern)

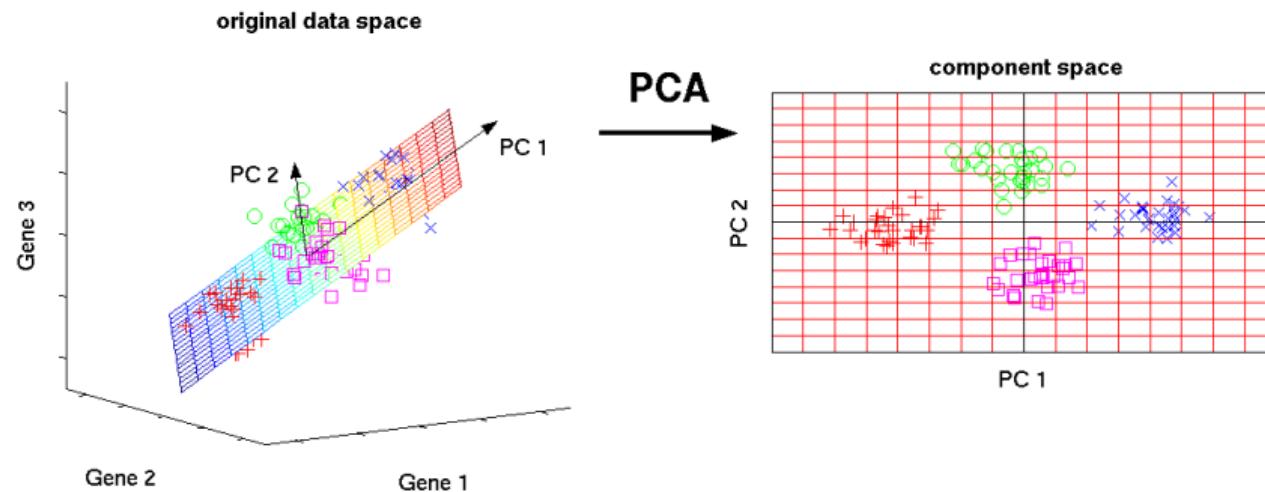
Fan Shape (Pattern)

# CASE STUDY - MEDICAL INSURANCE COSTS

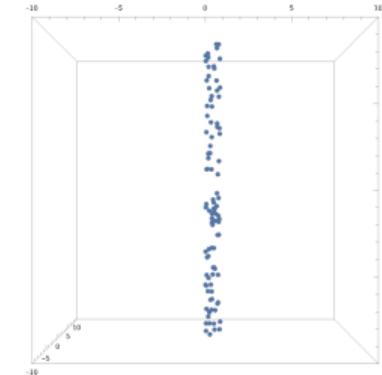
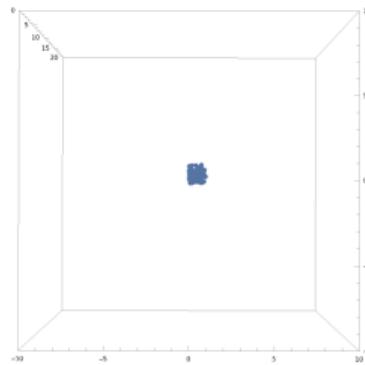
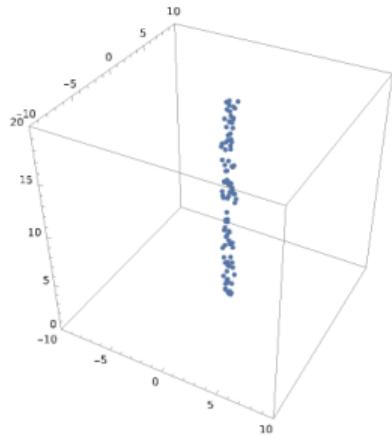
Variable	Definition
age	age of primary beneficiary
sex	insurance contractor gender, female, male
bmi	Body mass index - 18.5 to 24.9
children	Number of children covered by health insurance / Number of dependents
smoker	Smoking
region	the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
charges	Individual medical costs billed by health insurance

# UNSUPERVISED LEARNING

# PCA: PRINCIPAL COMPONENT ANALYSIS



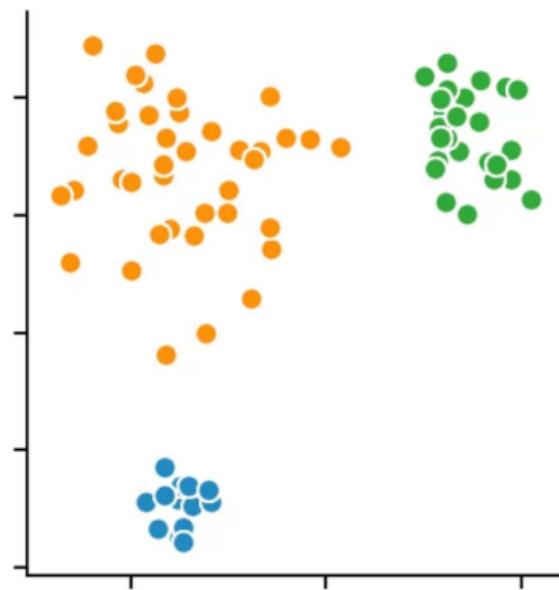
# PCA-INTUITION



Compute the projection that maximizes the **variance**

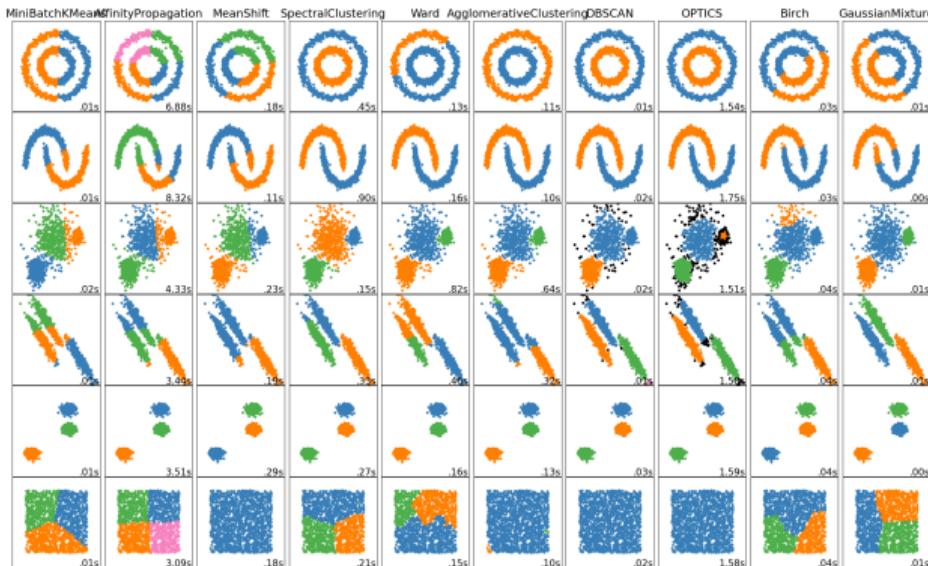
# CLUSTERING PROBLEM

- ▶ ``Homogeneous" groups of observations
  - observations in the same group should be similar
  - observations in different groups should be dissimilar
- ▶ Applications
  - Marketing segmentation
  - Social network analysis
  - Social Science
  - etc



# CLUSTERING METHODS

- ▶ Strategies
  - Partition
  - Hierarchical
  - Density based
  - Grid
- ▶ Classification
  - Exclusive
  - Fuzzy
  - Complete
  - Partial



# AFFINITY MEASURES

Based on distances

- ▶ Euclidean distance
- ▶ Minkowski distance

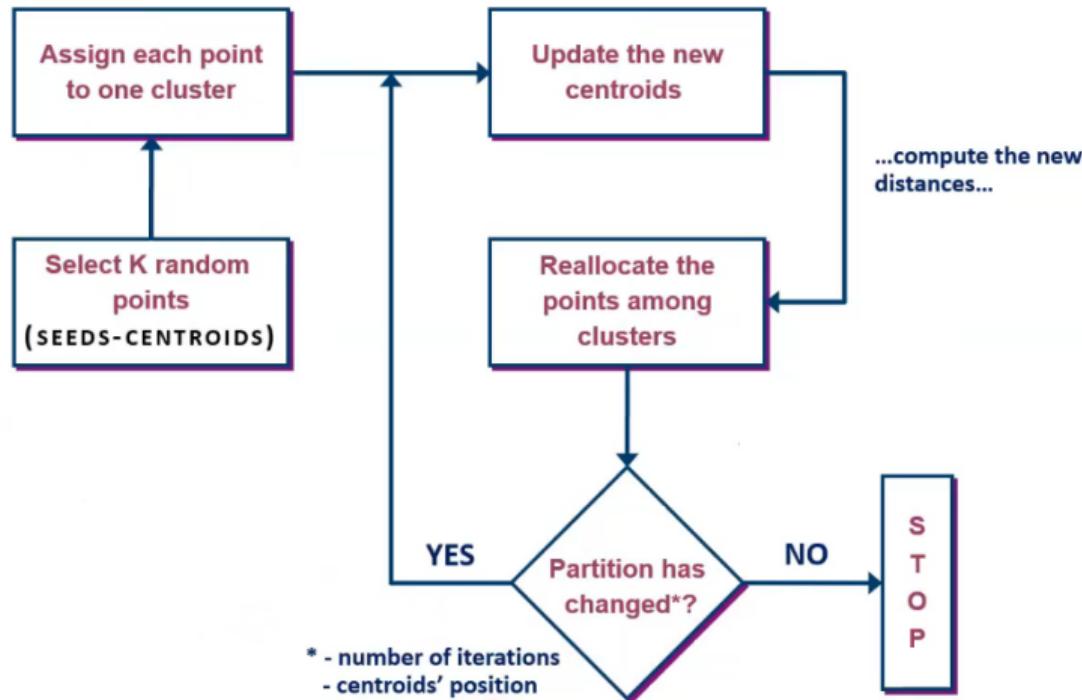
$$\sqrt[q]{\sum_{i=i}^n |x_{ij} - x_{kj}|^q}$$

- ▶ Mahalanobis distance (consider the covariance)

$$\sqrt{(x_i - x_k)^\top V^{-1} (x_i - x_k)}$$

- ▶ Cosine distance
- ▶ etc.

# PARTITION METHODS: K-MEANS

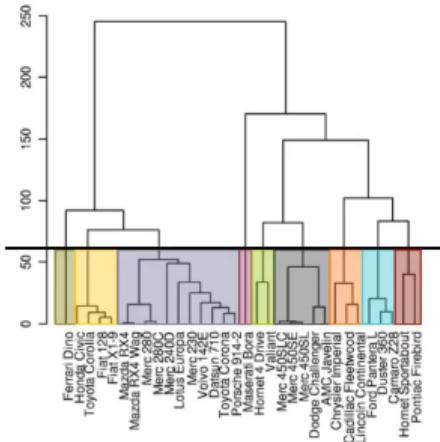


# HIERARCHICAL METHODS

- ▶ Tree structure
- ▶ Based in distances inter/intra-clusters
- ▶ No fixed number of clusters

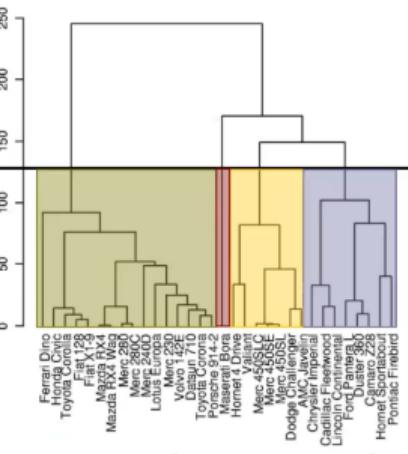
## 1. Agglomerative algorithms (bottom-up)

- ▶ starts with one cluster per observation,
- ▶ pairs of closest clusters are merged as one moves up the hierarchy.



## 2. Divisive algorithms (top-down)

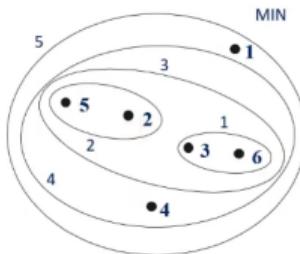
- ▶ start with a single cluster
- ▶ splits recursively as one moves down the hierarchy.



# HIERARCHICAL METHODS - INTERCLUSTER DISTANCE

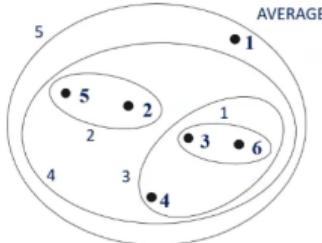
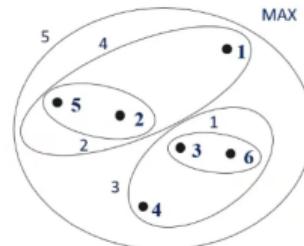
- ▶ **Minimum distance:** (Single Linkage)

$$dist(C_a, C_b) = \min_{x_i \in C_a, x_j \in C_b} d(x_i, x_j)$$



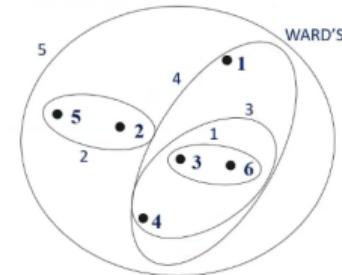
- ▶ **Maximum distance:** (Complete Linkage)

$$dist(C_a, C_b) = \max_{x_i \in C_a, x_j \in C_b} d(x_i, x_j)$$



- ▶ **Average distance**

- ▶ **Ward distance:** Minimizes the within cluster variance



# QUALITY MEASURES

► **Inertia**  $\sum_{i=1}^N (x_i - C_k)^2$

► **Cohesion** of a cluster

$$coes(C) = \sum_{x_i, x_j \in C} d(x_i, x_j)$$

► **Separation** between clusters

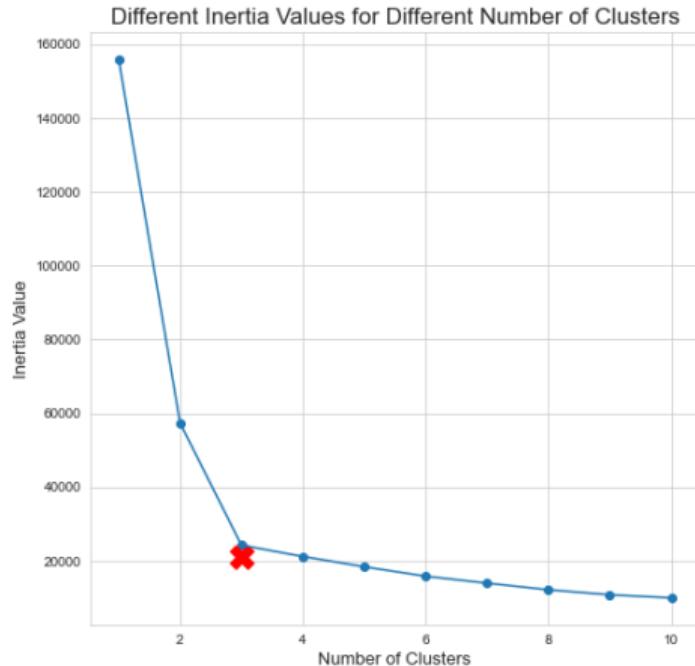
$$sep(C_a, C_b) = \sum_{x_i \in C_a, x_j \in C_b} d(x_i, x_j)$$

► **Overall Cohesion**

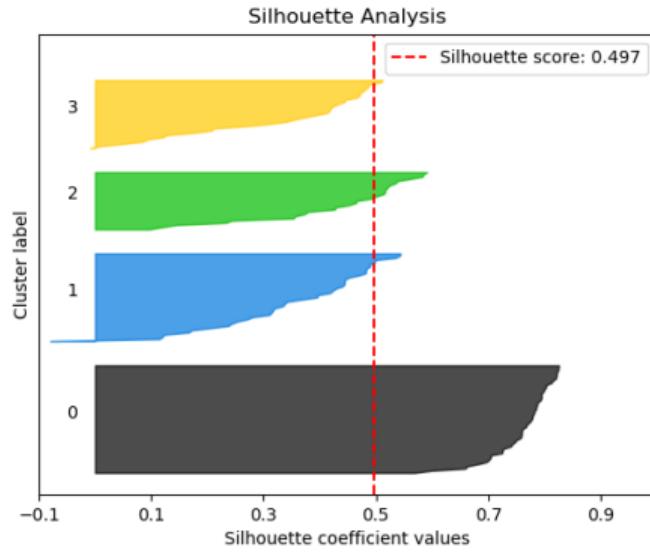
$$coes(\mathcal{C}) = \sum_{C \in \mathcal{C}} coes(C)$$

► **Overall Separation**

$$sep(\mathcal{C}) = \sum_{C_a, C_b \in \mathcal{C}} sep(C_a, C_b)$$



# SILHOUETTE

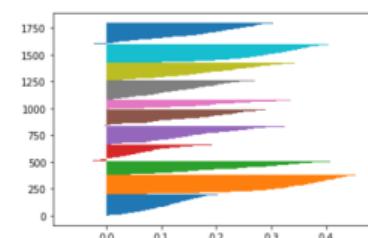
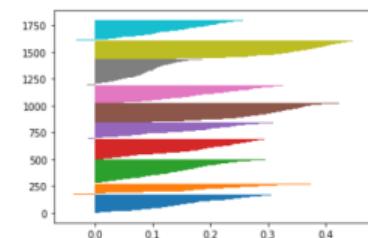
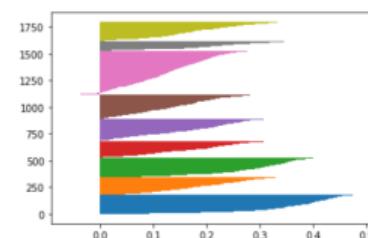
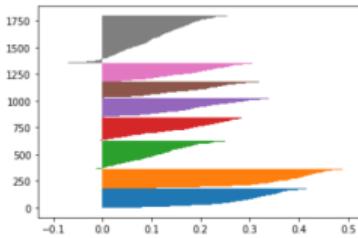
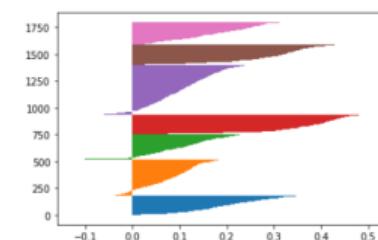
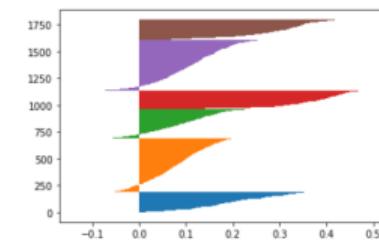
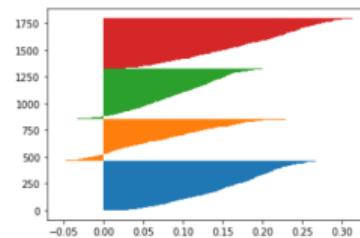
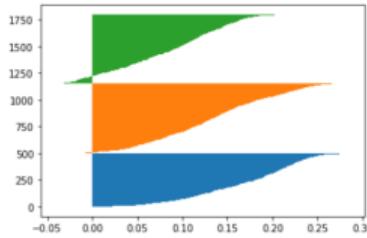


For an observation  $x_i$

- ▶ Compute the average distance  $u_i$  of  $x_i$  to all the others point in the **same** cluster
- ▶ For each of the other clusters  $C$ , compute the average distance from  $x_i$  to the elements of  $C$
- ▶ Compute the minimum of the previous distances  $v_i$

$$silh(x_i) = \frac{v_i - u_i}{\max(u_i, v_i)} \in [-1, 1]$$

# A CLUSTERING EXAMPLE



# MNIST DATASET

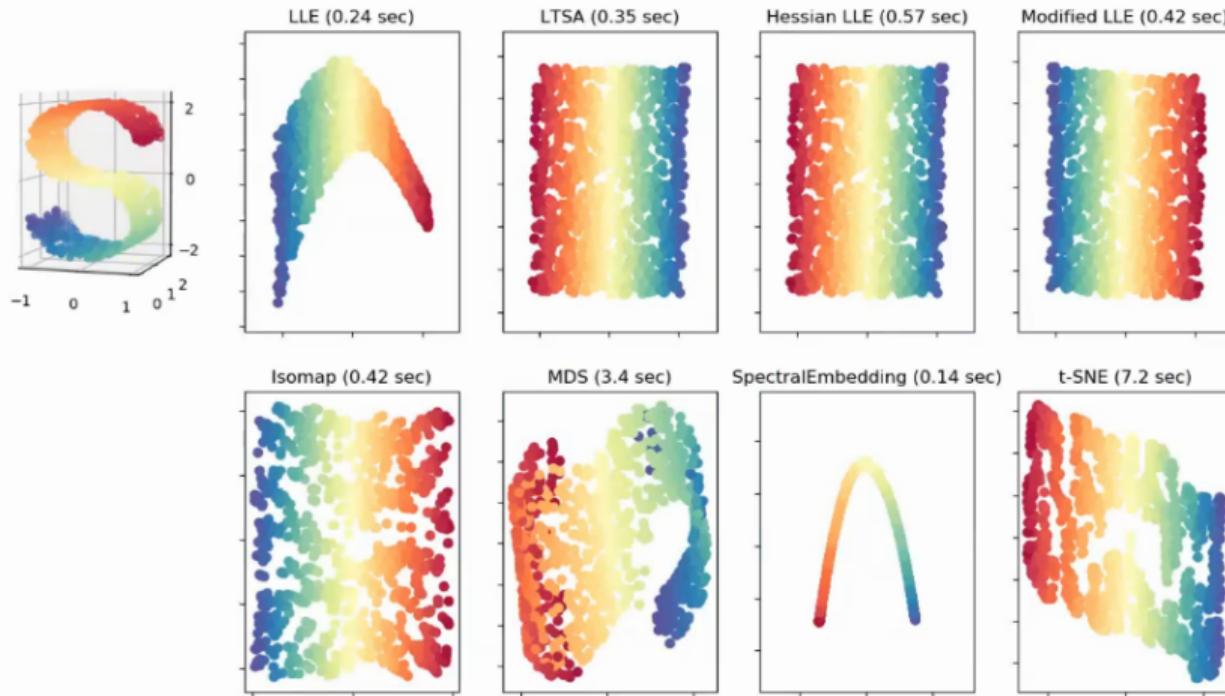


0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29	
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0	
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1	
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49	
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36	
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62	
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0	
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0	
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19	
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0	
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0	
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4	
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0	
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0	
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3	
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0	
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4	
0	18	146	250	255	247	255	255	249	255	240	255	129	0	5	0	
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0	
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1	
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0	

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29	
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0	
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1	
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49	
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36	
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62	
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0	
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0	
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19	
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0	
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0	
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4	
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0	
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0	
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3	
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0	
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4	
0	18	146	250	255	247	255	255	249	255	240	255	129	0	5	0	
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0	
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1	
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0	

# NON-LINEAR REDUCTION

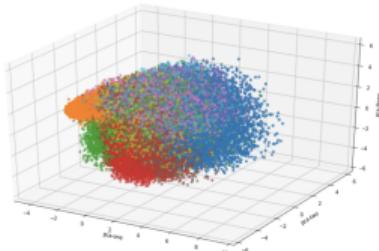
Manifold Learning with 1000 points, 10 neighbors



# T-SNE: T-DISTRIBUTED STOCHASTIC NEIGHBOUR EMBEDDING

We aim to project observations preserving observation distance.

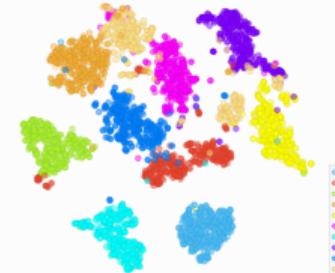
High-Dimensional Space



Distance as Normal Distribution

$$p_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq l} \exp\left(-\|x_k - x_l\|^2 / 2\sigma_i^2\right)}$$

Low-Dimensional Space



Distance as t-Students Distribution

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

We try to minimize the divergence between the distributions

$$KL(P\|Q) = \sum_i \sum_{j \neq i} p_{ij} \log \left( \frac{p_{ij}}{q_{ij}} \right) \quad (\text{Kullback-Leibler})$$

# MNIST FASHION (ZALANDO)

Image 1: Boot

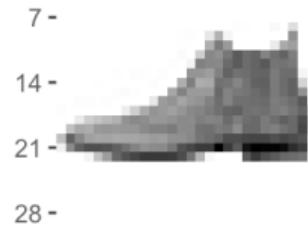


Image 2: Pullover



Image 3: Trouser



Image 4: Trouser



Image 5: Shirt



Image 6: Trouser



Image 7: Coat

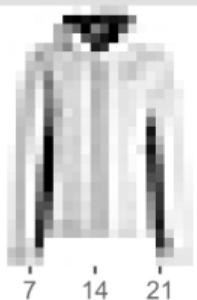


Image 8: Shirt

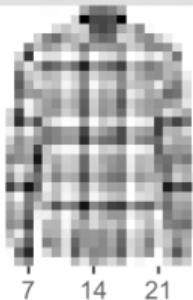


Image 9: Sandal

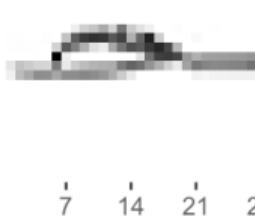
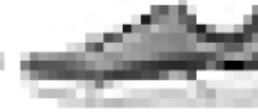


Image 10: Sneaker



# THANK YOU