NEW ROLE SCHOOL

DATA SCIENTIST

F2 | Machine Learning Lab

YOUR WAY TO THE FUTURE

GENERALI | We LEARN

AGENDA

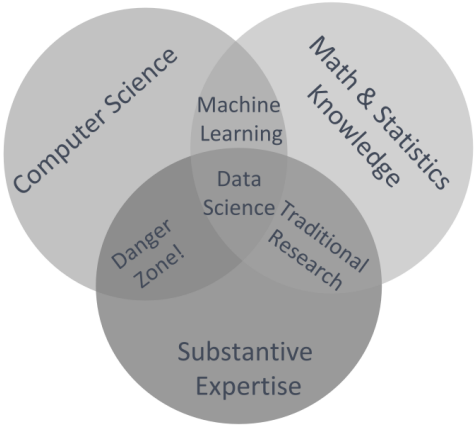
1. Python Intro
2. Exploratory Data Analysis
3. Supervised Learning
   - Classification
   - Regression

Question time

Go to https://pollev.com/mauriciosoto

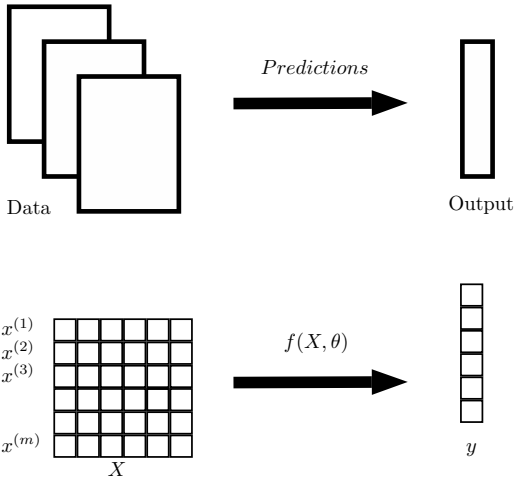
Data Science

- Drew Conway, 2010



@manudellavalle - http://emanueledellavalle.org

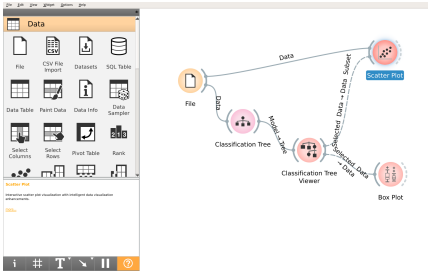
THE OBJECTIVE



Programming Tools

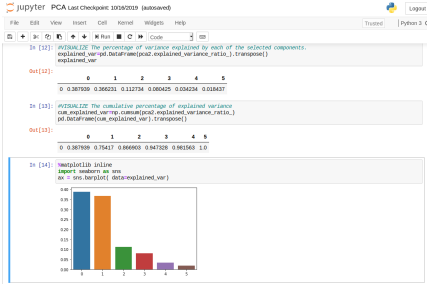
1. Orange

https://orange.biolab.si/



2. Jupyter-Notebook (Anaconda)

https://www.anaconda.com/



scikit learn

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/

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
Campain	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)
	campain	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**

[1] A data-driven approach to predict the success of bank telemarketing. S. Moroa, P. Cortez, P. Rita.Decision Support Systems, 62:22-31, 2014.

Supervised Learning example

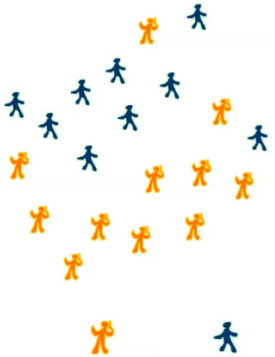
Features

Target

Observations/Clients

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

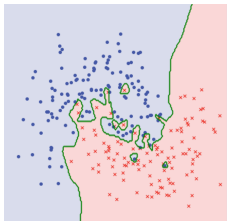
id	Age	Job	⋯	nr.employed	success
$n+1$	30	blue-collar.	⋯	0.61	**?**
$n+22$	21	unknown	⋯	0.16	**?**
$n+33$	66	housemaid	⋯	0.45	**?**
⋮	⋮	⋮	⋱	⋮	⋮
$n+k$	28	blue-admin.	⋯	0.35	**?**

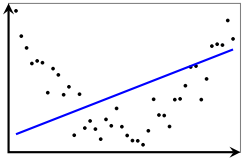


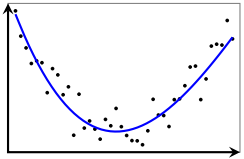
Supervised Learning - Problem Formulation

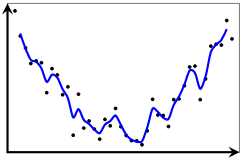
- **Training (observations) set:** $\mathcal{S}_m = \{(x_i, y_i),\, i \in \mathcal{M}\}$
  $x_i \in \mathbb{R}^n$ are the **independent features** while $y_i \in \mathcal{D}$ is the **target**.
- **Hypothesis space:** $\mathcal{H}$ space of functions (models) $f(x) : \mathbb{R}^n \to \mathcal{D}$
  In classification $\mathcal{D}$ is a discrete set (e.x. $\{0, 1\}$), in regression is a continuous set (e.x. $\mathbb{R}^+$)
- In **Supervise Learning** problems we attempt to define a hypothesis space $\mathcal{H}$ and a function $f^* \in H$ which optimally describe the **relation** between x_i and y_i allowing to predict correctly **future observations**.

Supervised Learning goal

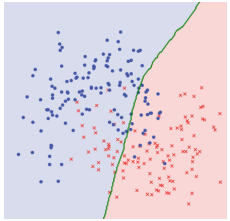


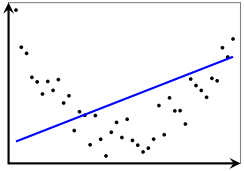




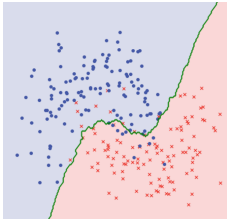


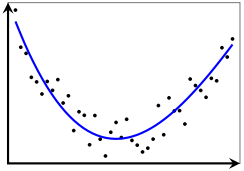
Supervised Learning goal



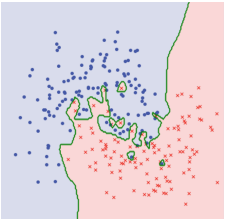


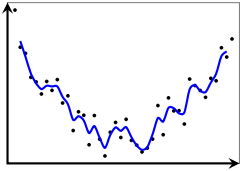
Underfitting





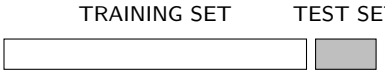
Balance

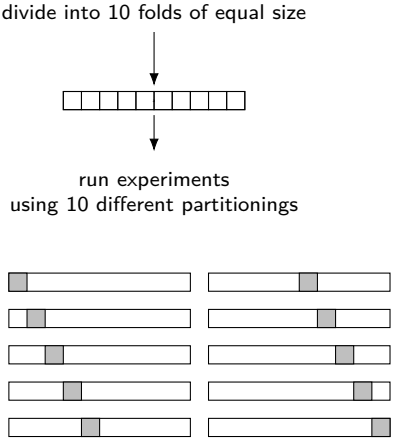




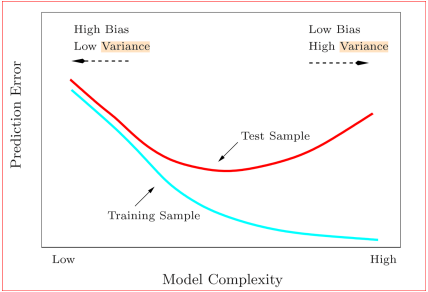
Overfitting

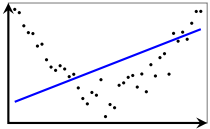
Hold out method / Cross validation



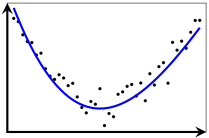


Under/Over-fitting

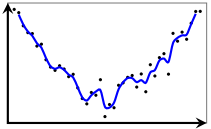




Underfitting

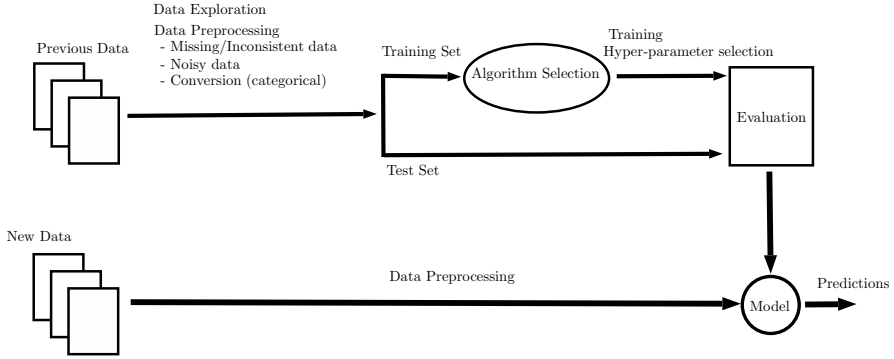


Balance



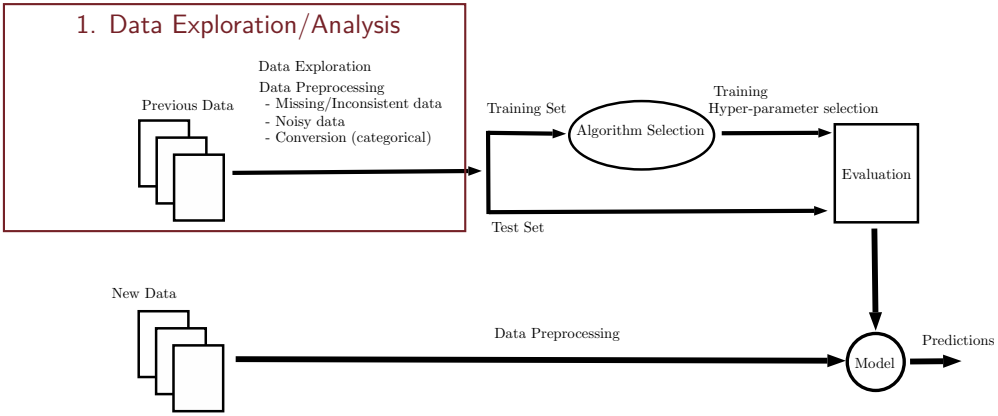
Overfitting

Workflow

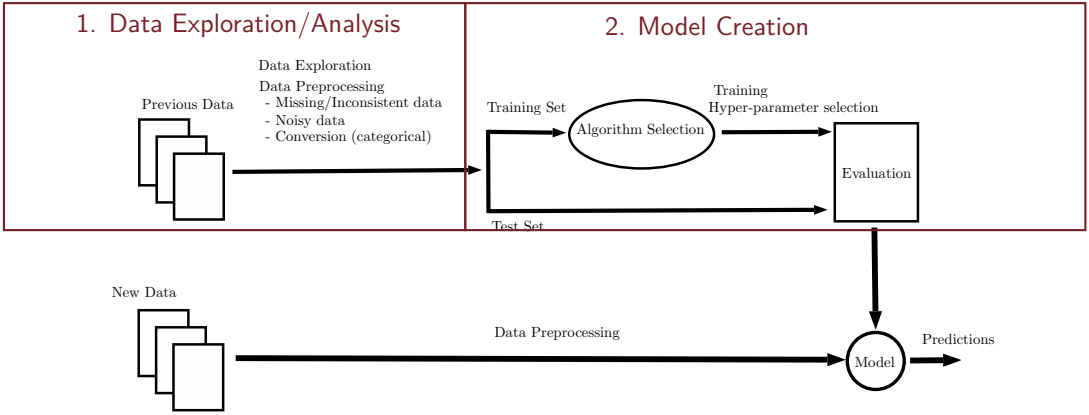

Data Exploration
Data Preprocessing
- Missing/Inconsistent data
- Noisy data
- Conversion (categorical)
Previous Data
Training Set
Algorithm Selection
Training
Hyper-parameter selection
Evaluation
Test Set
New Data
Data Preprocessing
Model
Predictions

Workflow

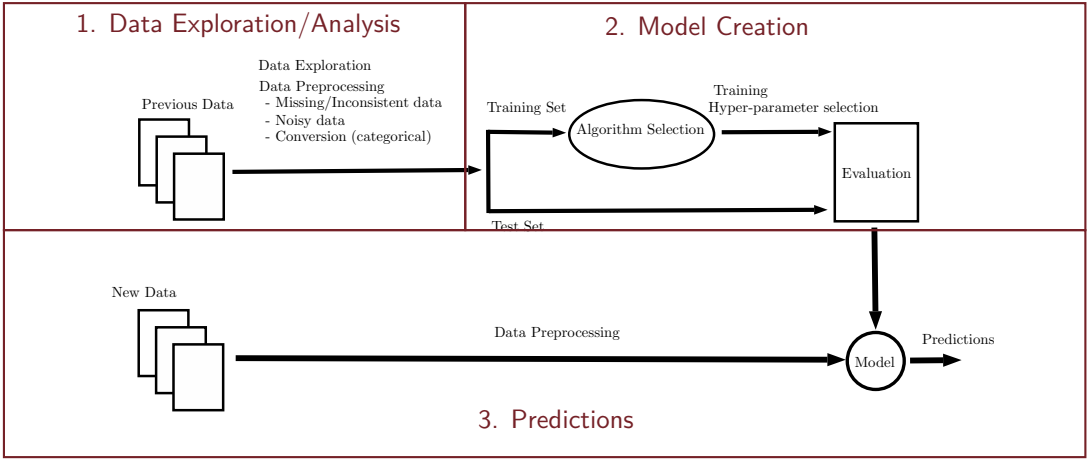
1. Data Exploration/Analysis



Workflow


1. Data Exploration/Analysis
2. Model Creation
Data Exploration
Data Preprocessing
- Missing/Inconsistent data
- Noisy data
- Conversion (categorical)
Previous Data
Training Set
Algorithm Selection
Training
Hyper-parameter selection
Evaluation
Test Set
New Data
Data Preprocessing
Model
Predictions

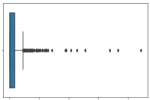
Workflow

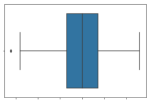

1. Data Exploration/Analysis
Data Exploration
Data Preprocessing
- Missing/Inconsistent data
- Noisy data
- Conversion (categorical)
Previous Data
2. Model Creation
Training Set
Algorithm Selection
Training
Hyper-parameter selection
Evaluation
Test Set
New Data
Data Preprocessing
Model
Predictions
3. Predictions

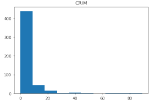
Data Exploration/Analysis

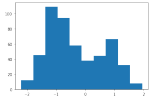
1. Data Validation

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



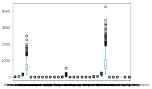


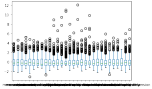




2. Data transformation

- Standardization
- Discretization
- Dummy variables
- Feature construction



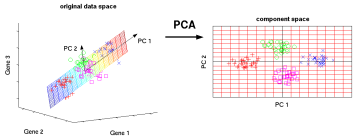


	type_of_food
0	fruit
1	vegetable
2	fruit
3	meat
4	fruit
5	vegetable

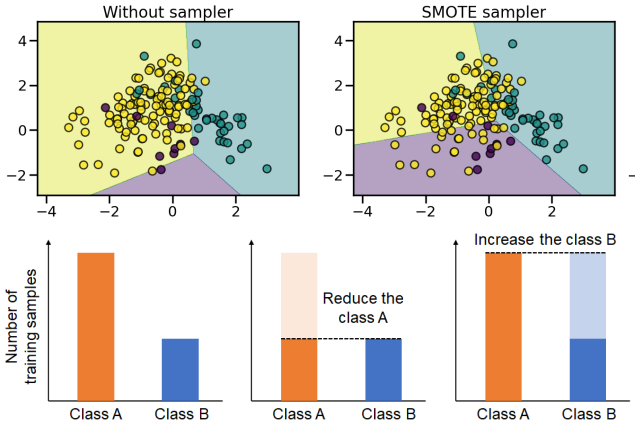
	fruit	meat	vegetable
0	1	0	0
1	0	0	1
2	1	0	0
3	0	1	0
4	1	0	0
5	0	0	1

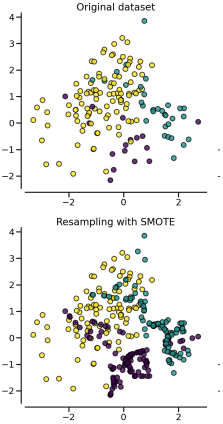
3. Data reduction

- 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

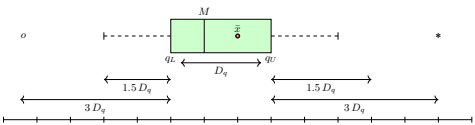
- Model determine the variable set quality
- Slow
- Ex. Greedy Forward

3. Embbeded Methods

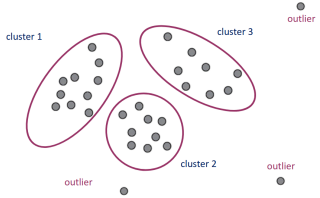
- Model determine the variable set
- Slow
- Ex. Tree models

Noisy Data

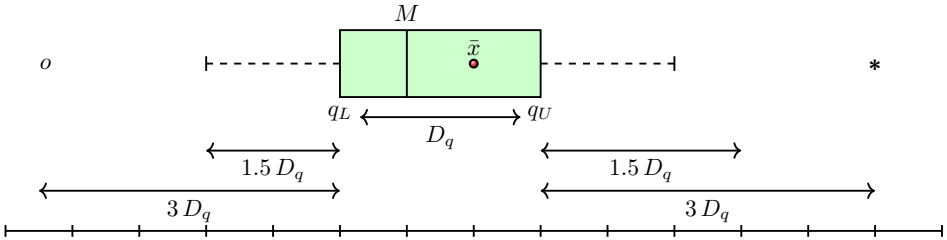
- Univariate
  - Normal-like distribution $[\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma}]$ contains about 96% of the data



  - In the general case, Tchebysheff theorem states taht for $\gamma > 1$ $[\bar{\mu} - \gamma\bar{\sigma}, \bar{\mu} + \gamma\bar{\sigma}]$ contains $1 - 1/\gamma^2$ proportion of the observations
- Multi variate
  - Clustering techniques



Box-plot



- $D_q = q_U - q_L = q_{0.75} - q_{0.25}$
- internal lower edge$= q_L - 1.5\,D_q$
- external lower edge$= q_L - 3\,D_q$

Data transformation

- **Decimal Scaling**

$$x'_{ij} = \frac{x_{ij}}{10^k}$$

- **Min-Max** in the interval $[x'_{\min,j}, x'_{\max,j}]$

$$x'_{ij} = \frac{x_{ij} - x_{\min,j}}{x_{\max,j} - x_{\min,j}}(x'_{\max,j} - x'_{\min,j}) + x'_{\min,j}$$

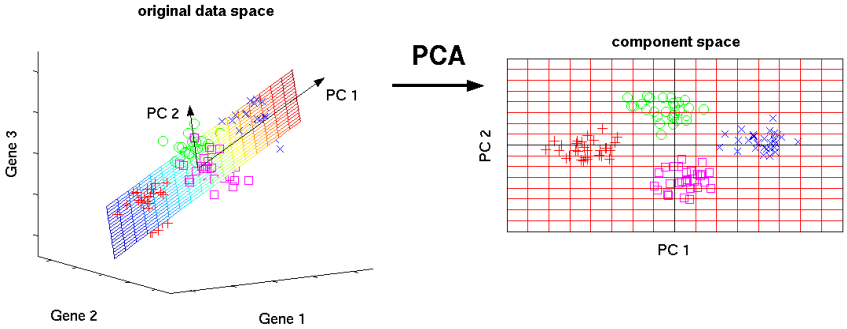
- z-**index**

$$x'_{ij} = \frac{x_{ij} - \bar{\mu}_j}{\bar{\sigma}_j}$$

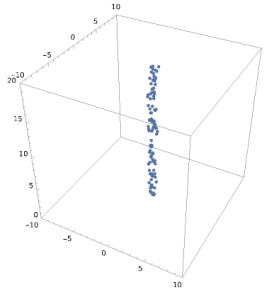
Data reduction

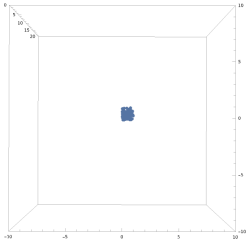
- **Sampling**
  - Simple sampling
  - Stratified sampling
- **Selection**
  - Filter methods
  - Wrapper methods
  - Embedded methods
- **Discretization, Aggregation**
- **Projection** (ex. PCA)

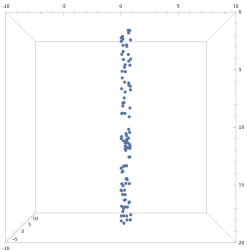
PCA: Principal Component Analysis



PCA-intuition

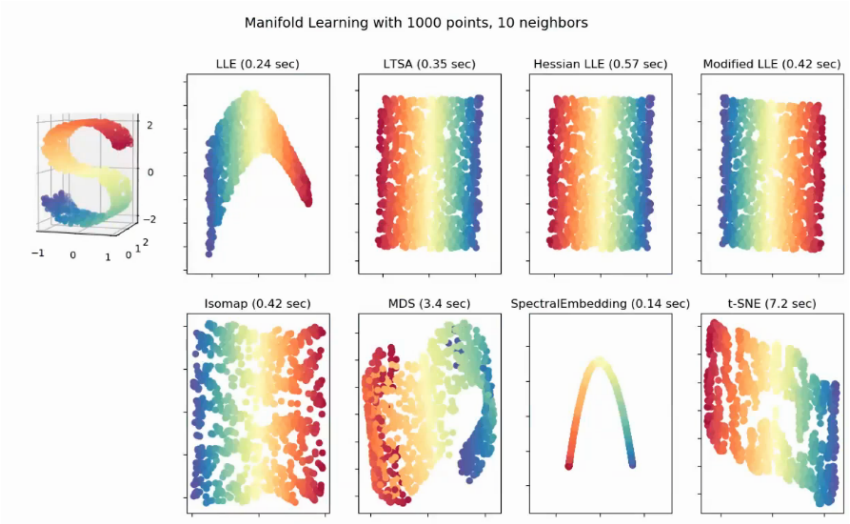






Compute the projection that maximizes the **variance**

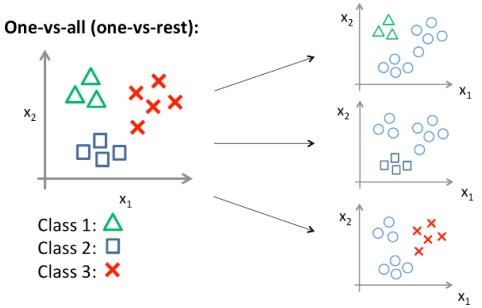
Nonlinear reduction



Supervised Learning: Classification

Multi-class classification

1. **One-vs-Rest** One binary classifications for every class.



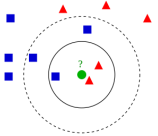
[1]

2. **One-vs-One** One binary classifications for every pair of classes.

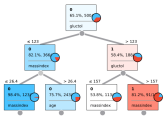
[1] Image via www.cc.gatech.edu

Classification Models

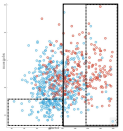
- Heuristics Methods
  - Nearest Neighbours
  - Classification Trees
- Probabilistic Methods
  - Bayesian Methods
- Regression Methods
  - Logistic regression
- Separation Methods
  - Support vector machine
  - Perceptron
  - Neural Networks

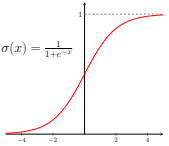


KNN

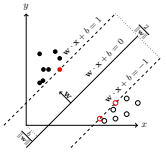


Classification Tree

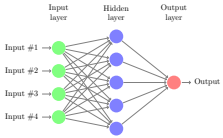




Logistic Regression



SVM



Neural Network

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
Campain	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)
	campain	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**

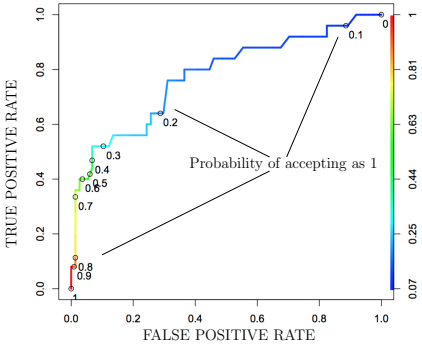
[1] A data-driven approach to predict the success of bank telemarketing. S. Moroa, P. Cortez, P. Rita.Decision Support Systems, 62:22-31, 2014.

Quality measures - Confusion Matrix

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

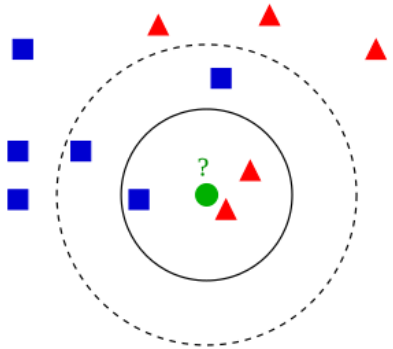
- Precision $= \frac{TP}{TP+FP}$
  *"proportion of true positives among **positive predictions**"*
- False Positive rate$=\frac{FP}{FP+TN}$
  *"proportion of false positives among **actual negatives**"*
- Recall (True Positive rate)$=\frac{TP}{FN+TP}$
  *"proportion of true positives among **actual positive**"*
- 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 classificator is the the point $(0, 1)$
- $AUC \in [0.5, 1]$ area under the curve is a quality measure of our algorithm.

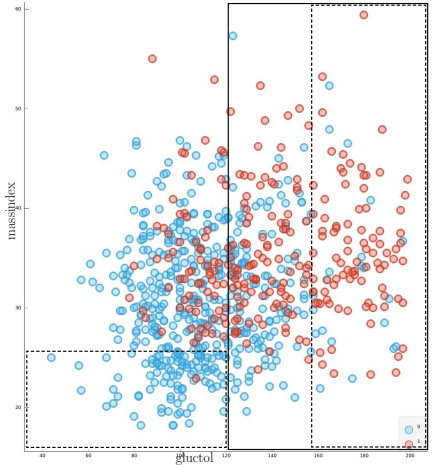
KNN K-nearest Neighbours

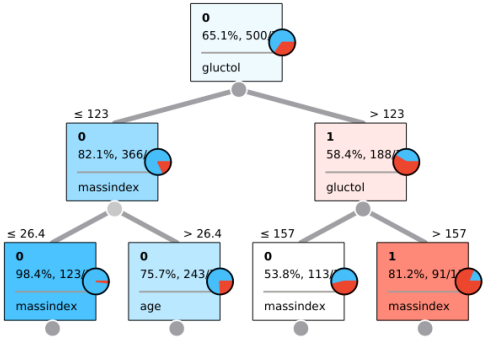


Main Parameters

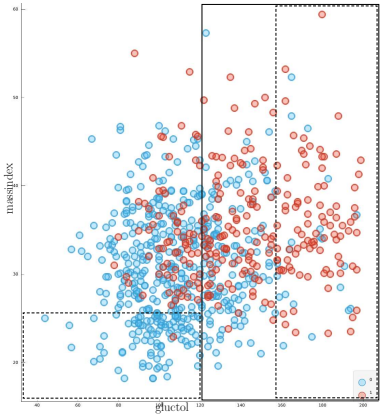
- k : number of neighbours
- neighbour weights
- distances

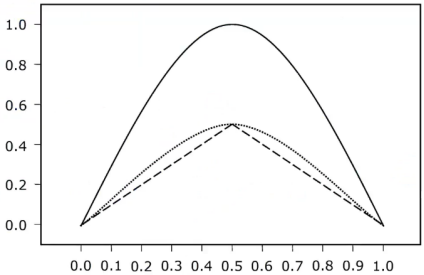
Classification tree





Classification tree





- **Entropy index:**
- **Gini index:**
- **Miss-classification index:**

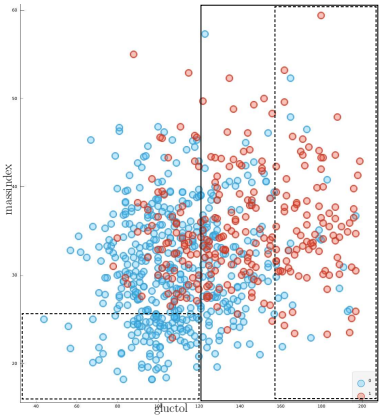
Split criteria

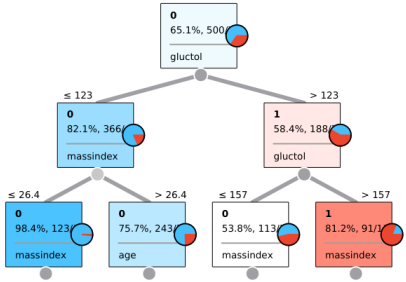
$$-\sum_{h=1}^{H} f_h \log_2 f_h$$

$$1 - \sum_{h=1}^{H} f_h^2$$

$$1 - \max_h f_h$$

Classification tree

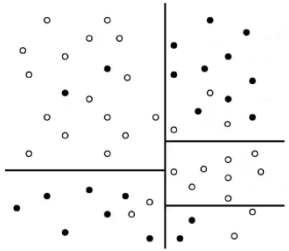




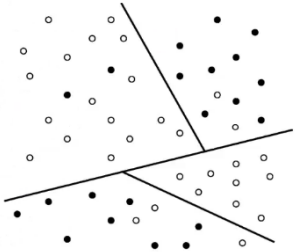
Main Parameters

- impurity measure: “gini”, “entropy”
- 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

Uni/Multi-variate classification tree



classification by an
axis parallel tree



classification by an
oblique tree

Naive Bayesian Classifier

- Bayes Theorem

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

- Maximum a posteriori hypothesis

$$y_{MAP} = \arg\max_{y \in \mathcal{H}} P(y|\mathbf{x}) = \arg\max \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

- Independence (Naive)

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

Naive Bayesian Classifier

- Categorical/discrete attributes

$$P(x_j|y) = P(x_j = r_{jk}|y = v_h)$$

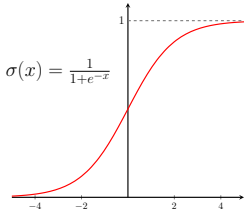
$\rightarrow$ empirical frequency of the observed value on the class v_h

- Numerical attribute

$$P(x_j|y) \sim N(\mu_{jh}, \sigma_{jh})$$

$\rightarrow$ assuming Gaussian density with empirical parameters

Logistic regression



$$\log \frac{P(y=1|x)}{P(y=0|x)} = w_0 + w_1 x_1 + \cdots + w_n x_n$$
$$= w^\top x$$

$$P(y=0|x) = \frac{1}{1+e^{w^\top x}}, P(y=1|x) = \frac{1}{1+e^{-w^\top x}}$$

$$P(y|x) = (1+\exp(-y\,(w^\top x)))^{-1} \quad (y = \pm 1)$$

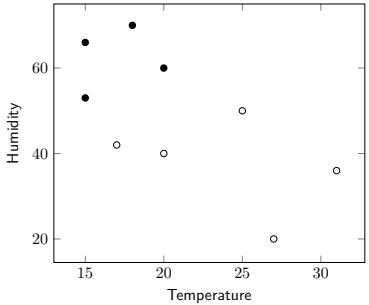
Likelihood

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

$$\max \log L(\theta \mid y; x) = \max \sum_{i=1}^{n} \log \Pr(y_i \mid x_i; w)$$
$$= \max - \sum_{i=1}^{n} \log(1 + \exp(-y\,(w^\top x))$$

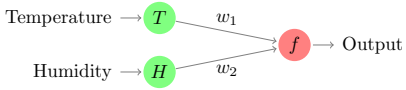
Logistic Regression - example

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

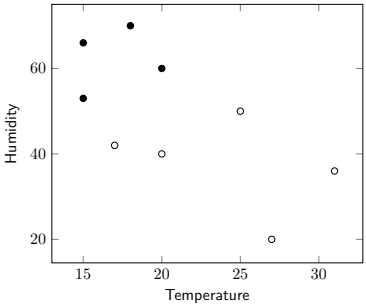


Logistic Regression - example

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

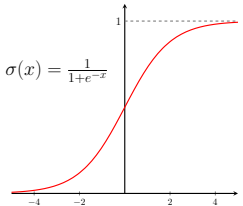


$$P(y = 1|T, H; w) = \underbrace{\sigma(w_0 + w_1 \cdot T + w_2 \cdot H)}_{\text{sigmoid function}}$$



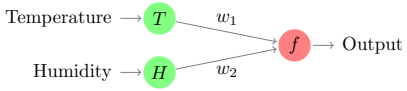
Maximize Likelihood

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

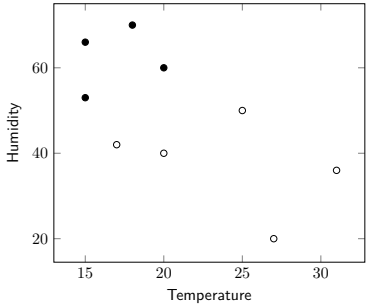


Logistic Regression - example

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



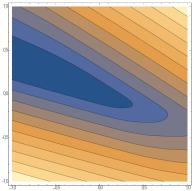
$$P(y = 1|T, H; w) = \underbrace{\sigma}_{\text{sigmoid function}}(w_0 + w_1 \cdot T + w_2 \cdot H)$$



Maximize Likelihood

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

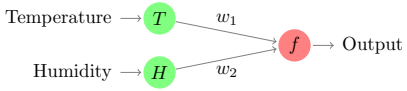
$$P(\text{rain}) = \sigma(-0.044 - 0.633 \times T + 0.235 \times H)$$



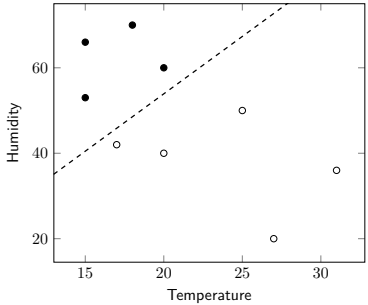
$$w^* = (-0.044, -0.633, 0.235)$$

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



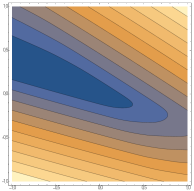
$$P(y = 1|T, H; w) = \underbrace{\sigma}_{\text{sigmoid function}}(w_0 + w_1 \cdot T + w_2 \cdot H)$$



Maximize Likelihood

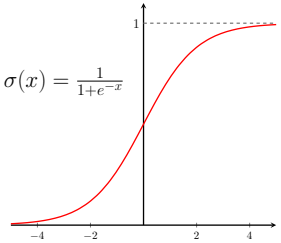
$$\max_w L(w \mid y; x) = \Pr(Y \mid X; w) = \prod_i \Pr(y_i \mid x_i; w)$$

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



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

Logistic regression



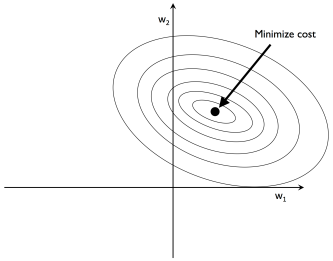
$$\min_{w} \underbrace{\frac{1}{2}||w||^2}_{\text{regularization}} + C \underbrace{\sum_{i=1}^{n} \log(1 + \exp(-y_i\,(w^T x_i)))}_{\text{error}}$$

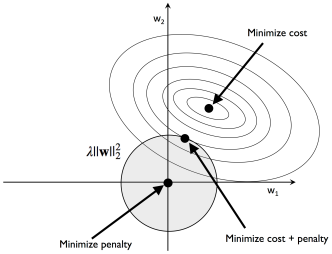
Main Parameters

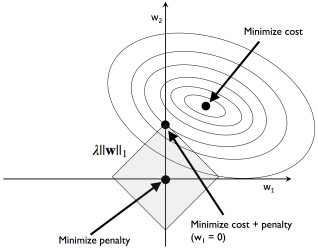
- C: Inverse of regularization strength

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

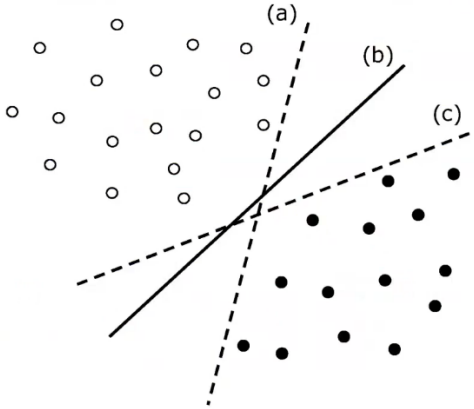
Regularization



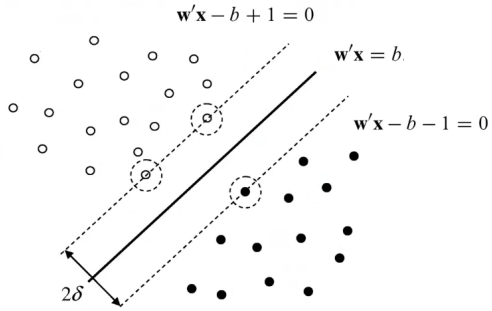




Support Vector Machine - linearly separable



SVM - linearly separable



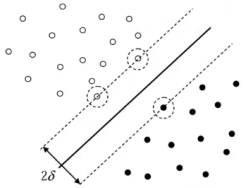
$$\delta = \frac{2}{\|w\|}, \qquad \|w\| = \sqrt{\sum_{j \in \mathcal{N}} w_j^2}$$

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2$$

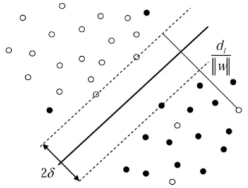
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 \quad i \in \mathcal{M}$$

SVM - general case

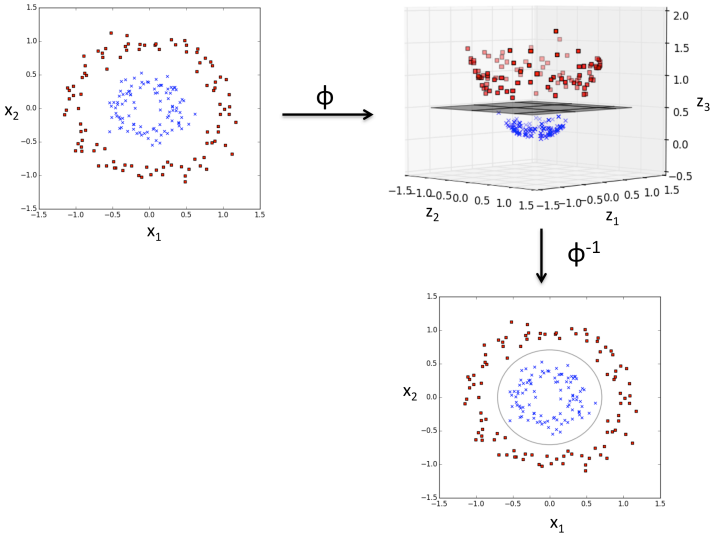
$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 \quad i \in \mathcal{M}$$



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

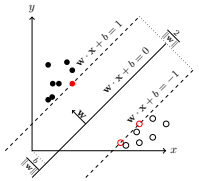


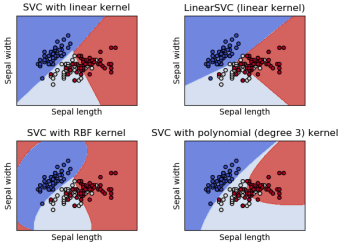
SVM - kernels



Try yourself: https://dash.gallery/dash-svm/

SVM - general case





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

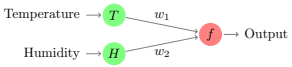
$$\text{subject to } y_i(w^T \underbrace{\phi(x_i)}_{\text{kernel}} - b) \geq 1 - d_i,$$

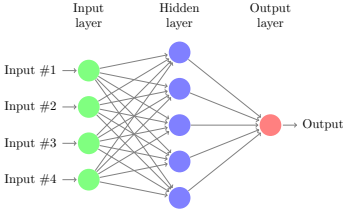
$$d_i \geq 0$$

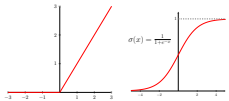
Main Parameters

- 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







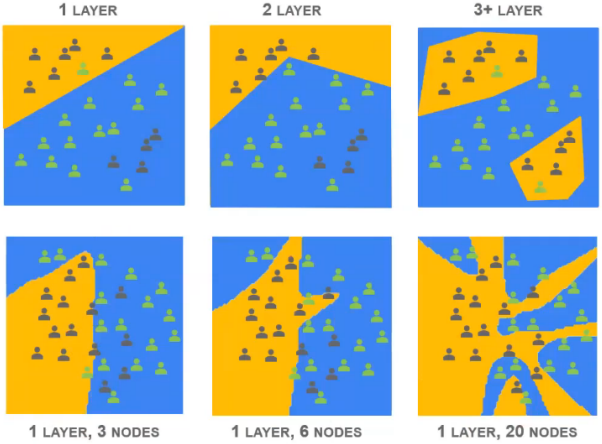
ReLU

Sigmoid

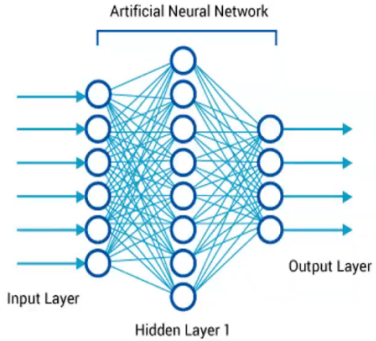
Main Parameters

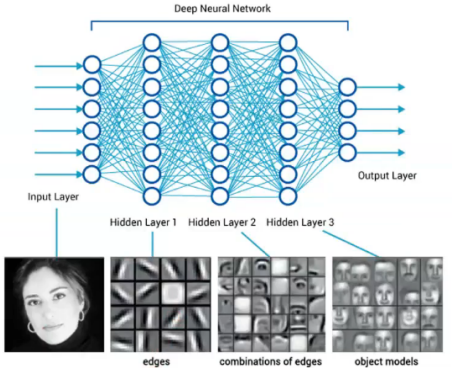
- hidden_layer_sizes: $(n_1, n_2, \ldots, n_L)$
- activation: identity, logistic, tanh, relu
- alpha regularization term parameter
- Resolution algorithm parameters: tol, max_iter.

Neural Network



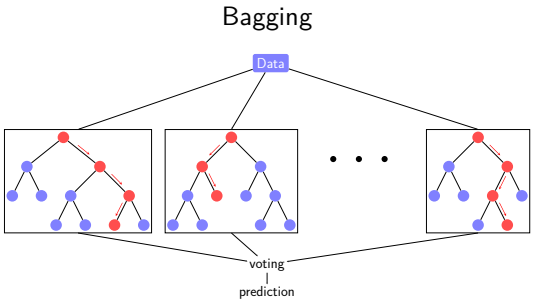
Deep Learning

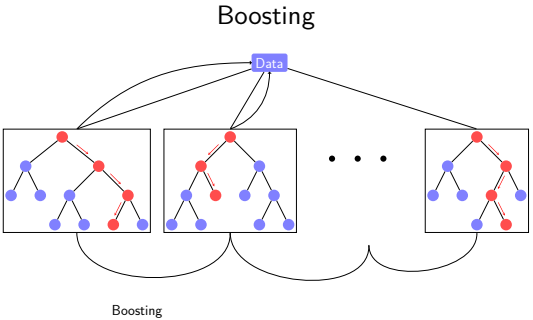




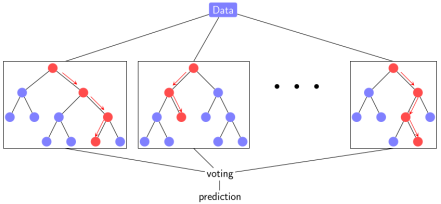
Try yourself: https://adamharley.com/nn_vis/

Ensemble Methods





Random Forest

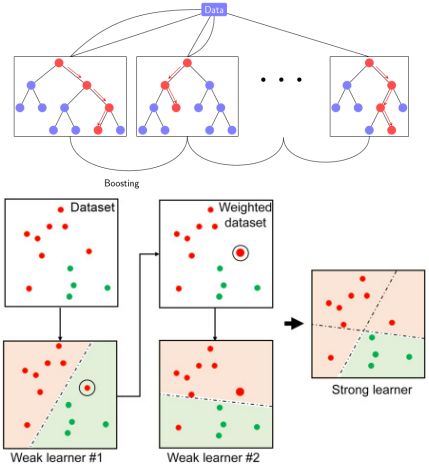


1. Create different (simple) tree models (stumps)
2. Each model is created with a subset of observation/features ($\sim 2m/3$)
3. We combine the prediction of all trees

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 $k = 1, \ldots, K$
   - 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 (λ)

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

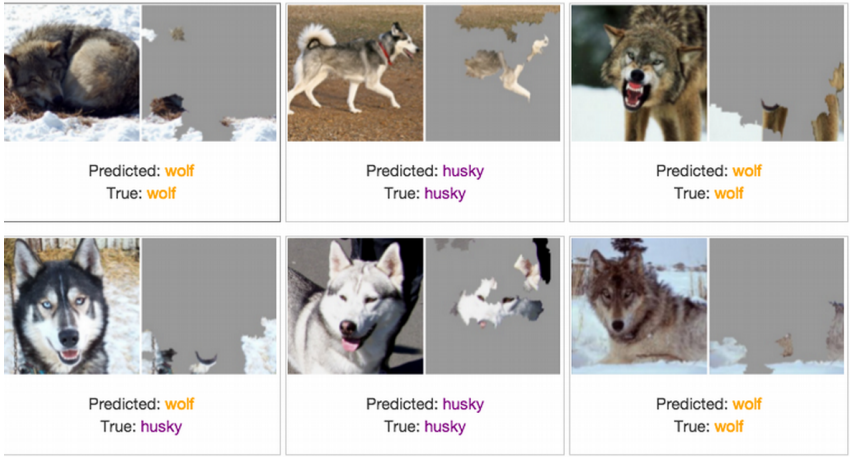
Predicted: wolf
True: wolf

Predicted: wolf
True: husky

Predicted: husky
True: husky

Predicted: wolf
True: wolf

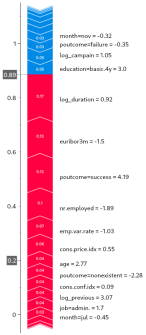
Model Explainability

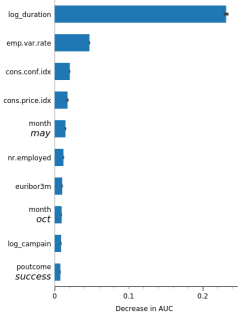


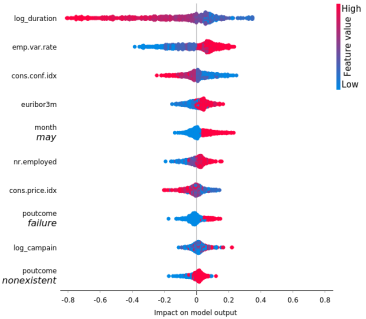
Model Explainability

SHAP: SHapley Additive exPlanation

"Marginal contribution of the feature in the prediction"







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

Supervised Learning: Regression

Quality measures - Regression

- Coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{SS_{res}}{SS_{tot}}$$

- 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: $RMSE = \sqrt{MSE}$
- Mean Absolute Percentage Error:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

Regression Models

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

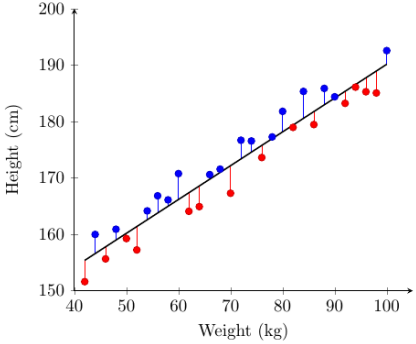
Simple linear regression

- Deterministic model

$$Y = w\,X + b$$

- Probabilistic model

$$Y = w\,X + b + \varepsilon$$



Regression models (n=1)

- Linear

$$Y = b + \sum_{j=1}^{n} w_j X_j \; = \; b + w_1 X_1 + w_2 X_2 + \cdots + w_n X_n \; = \; b + Xw$$

- Quadratic

$$\begin{aligned} Y &= b + Xw + X^2 d \qquad Z = X^2 \\ &= b + Xw + Zd \end{aligned}$$

- Exponential

$$\begin{aligned} Y &= e^{b+Xw} \qquad Z = logY \\ &= b + Xw \end{aligned}$$

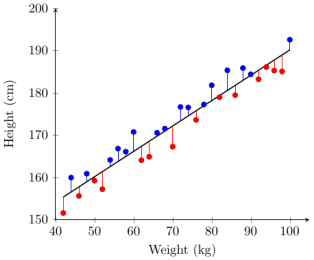
Simple linear regression

- Residuals

$$e_i = y_i - f(x_i) = y_i - wx_i - b \qquad i \in \mathcal{M}$$

- Least square regression

$$SSE = \sum_{i=1}^{m} e_i^2 = \sum_{i=1}^{m} [y_i - wx_i - b]^2$$



General Linear Models

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

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

- For example, for $n = 2$

$$Y = X_1 w_1 + X_2 w_2 + X_1^2 w_3 + X_2^2 w_4 + [X_1 X_2] w_5 + b + \varepsilon$$

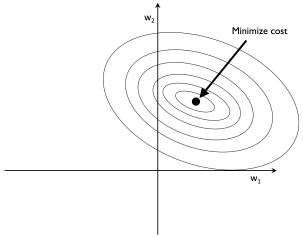
Linear Models Regularization

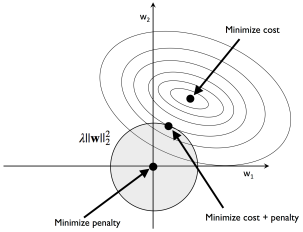
- Ridge:

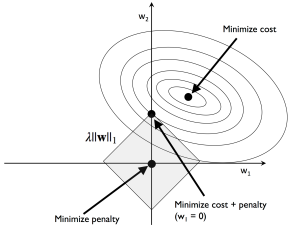
$$\min_w \lambda||w||^2 + ||e||^2 = \min_w \lambda||w||^2 + (y - Xw)^\top (y - Xw)$$

- Lasso:

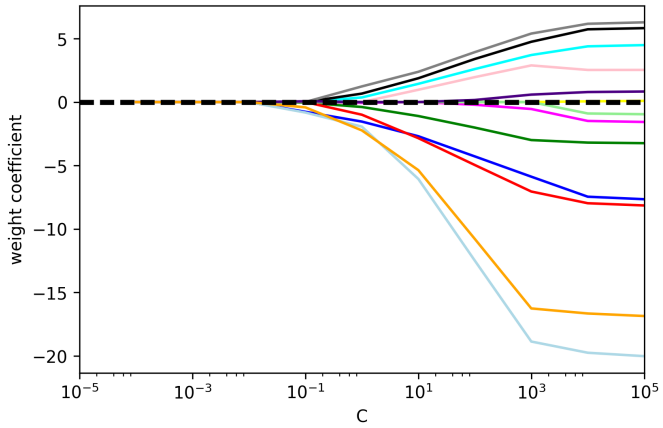
$$\min_w \lambda|w| + ||e||^2 = \min_w \lambda|w| + (y - Xw)^\top (y - Xw)$$







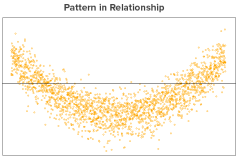
Regularization effect

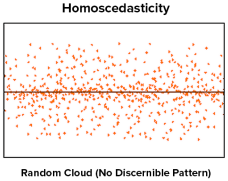


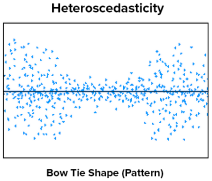
$$C = 1/\lambda$$

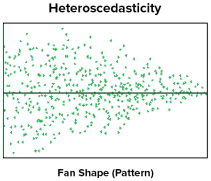
Residual assumptions

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



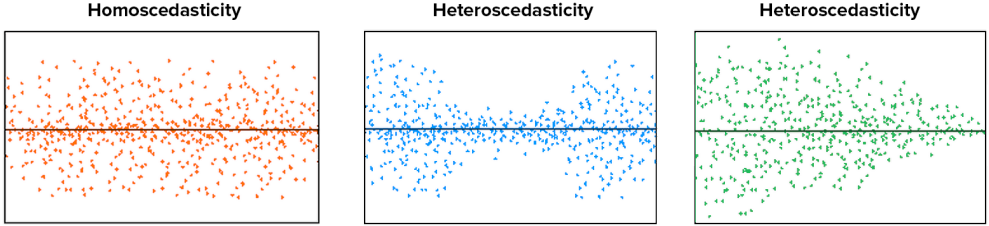






Normal residual assumption

- Graphical distribution



- Graphically compare error distribution against a normal distribution with QQ-plots
- Apply an hypothesis test to check the normality of the errors (Kolmogorov–Smirnov, D'Agostino, etc.)

Linear models - Significance of coefficients

- By assuming residuals independent and normal distribution
- Variance of coefficients

$$Var(\hat{w}) = (X'X)^{-1}\sigma^2 \quad \hat{w} \sim \mathcal{N}(w, (X'X)^{-1}\sigma^2)$$

- Empirical Variance

$$\hat{\sigma} = \frac{SSE}{m-n-1} = \frac{\sum_{i=1}^{m}(y_i - \mathbf{w}'\mathbf{x_i})^2}{m-n-1}$$

-

$$(m-n-1)\,\hat{\sigma}^2 \sim \sigma^2 \chi^2_{m-n-1}$$

- Under the null hypothesis $w_i = 0$ then

$$\frac{\hat{w}_i}{\hat{\sigma}\sqrt{(X'X)_{ii}}} \sim t_{m-n-1}$$

Linear models - Significance of coefficients

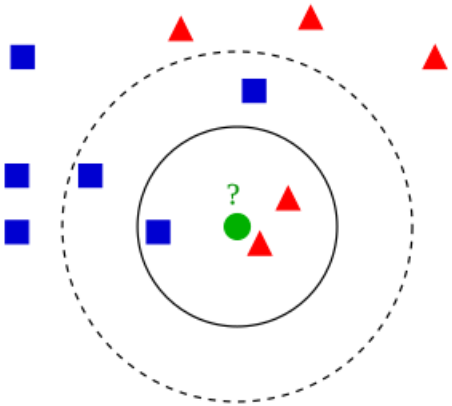
	coef	std err	t	P>|t|	[0.025	0.975]
const	22.5693	0.245	92.144	0.000	22.088	23.051
CRIM	-0.8678	0.298	-2.909	0.004	-1.455	-0.281
ZN	0.9310	0.365	2.551	0.011	0.213	1.649
INDUS	0.5166	0.494	1.045	0.297	-0.456	1.489
CHAS	0.0671	0.270	0.249	0.804	-0.463	0.598
NOX	-1.6601	0.532	-3.121	0.002	-2.706	-0.614
RM	3.3925	0.340	9.971	0.000	2.723	4.062
AGE	-0.2093	0.429	-0.488	0.626	-1.052	0.634
DIS	-2.7910	0.475	-5.879	0.000	-3.725	-1.857
RAD	2.3790	0.650	3.660	0.000	1.100	3.658
TAX	-2.1962	0.718	-3.059	0.002	-3.608	-0.784
PTRATIO	-2.0690	0.325	-6.372	0.000	-2.708	-1.430
B	0.5860	0.298	1.965	0.050	-0.001	1.173
LSTAT	-3.4712	0.432	-8.032	0.000	-4.321	-2.621

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.

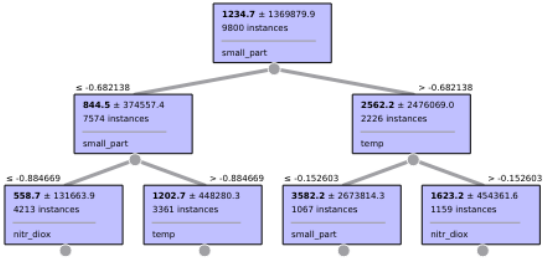
KNN K-nearest Neighbours



Main Parameters

- k : number of neighbours
- neighbour weights
- distances

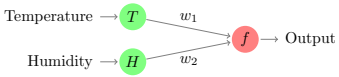
Regression tree

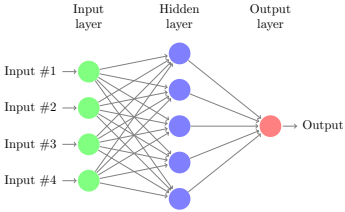


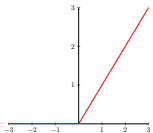
Main Parameters

- variability measure: mse (i.e., reduction in variance), mae, ...
- 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

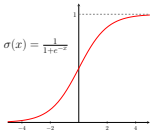
Multi-Layer Perceptron







ReLU

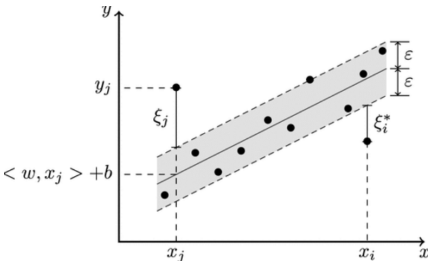


Sigmoid

Main Parameters

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

SVR



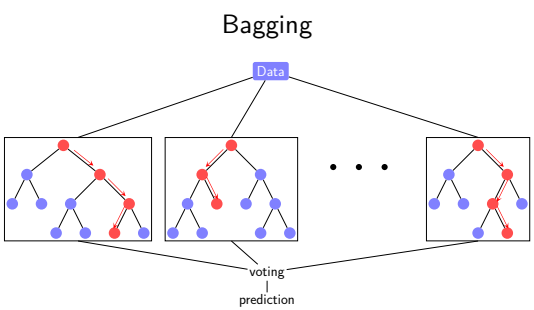
$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2}\|w\| + C\sum_{i=1}^{n}(\zeta_i + \zeta_i^*)$$

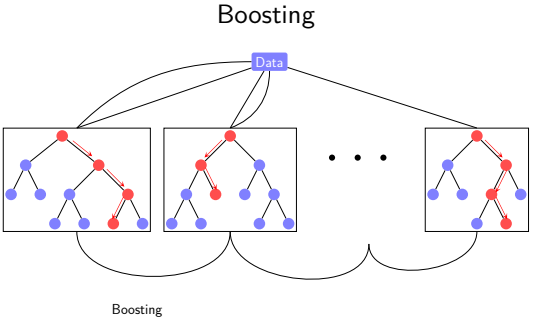
$$\text{subject to } y_i - w^T\phi(x_i) - b \leq \varepsilon + \zeta_i,$$
$$w^T\phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*,$$
$$\zeta_i, \zeta_i^* \geq 0, i = 1, ..., n$$

Main Parameters

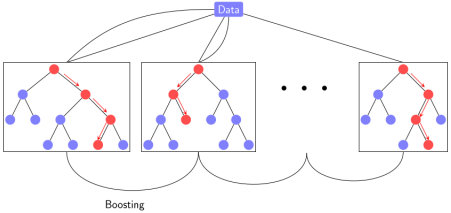
- C: inverse of regularization strength
- ε: tolerance
- kernel
- Resolution algorithm parameters

Ensemble Methods





Gradient Boost



Motivation:

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(a)}{3!}(x - x_0)^3 + \cdots$$

1. Train a weak learner F_0 and compute predictions $x^{(k)}$
2. For $k = 1, \ldots, K$
   - 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 minimize the loss function (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 (λ)