Grau de Ciències Ambientals Anàlisi de Dades

Data science: maneig i anàlisi de dades (BLOC MULTIVARIANT)

Toni Monleón-Getino (basat en diversos materials lliures)

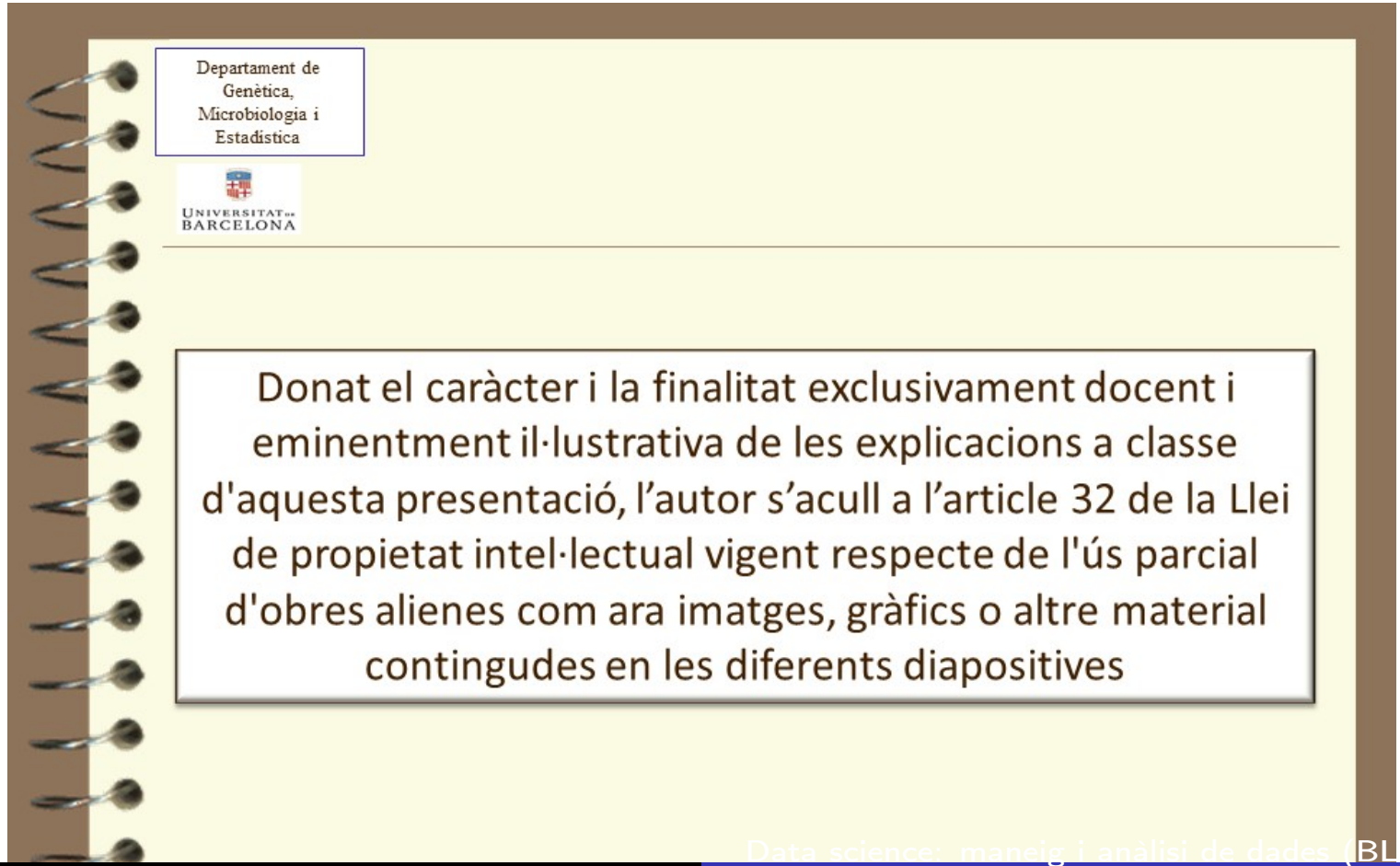
Novembre 2020

Secció d'Estadística
Facultat de Biologia

UNIVERSITAT DE BARCELONA

Continguts

Nota aclaridora



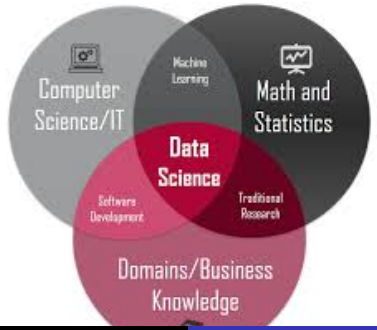
Introducció

Toni Monleón-Getino (basat en diversos materials lliures)

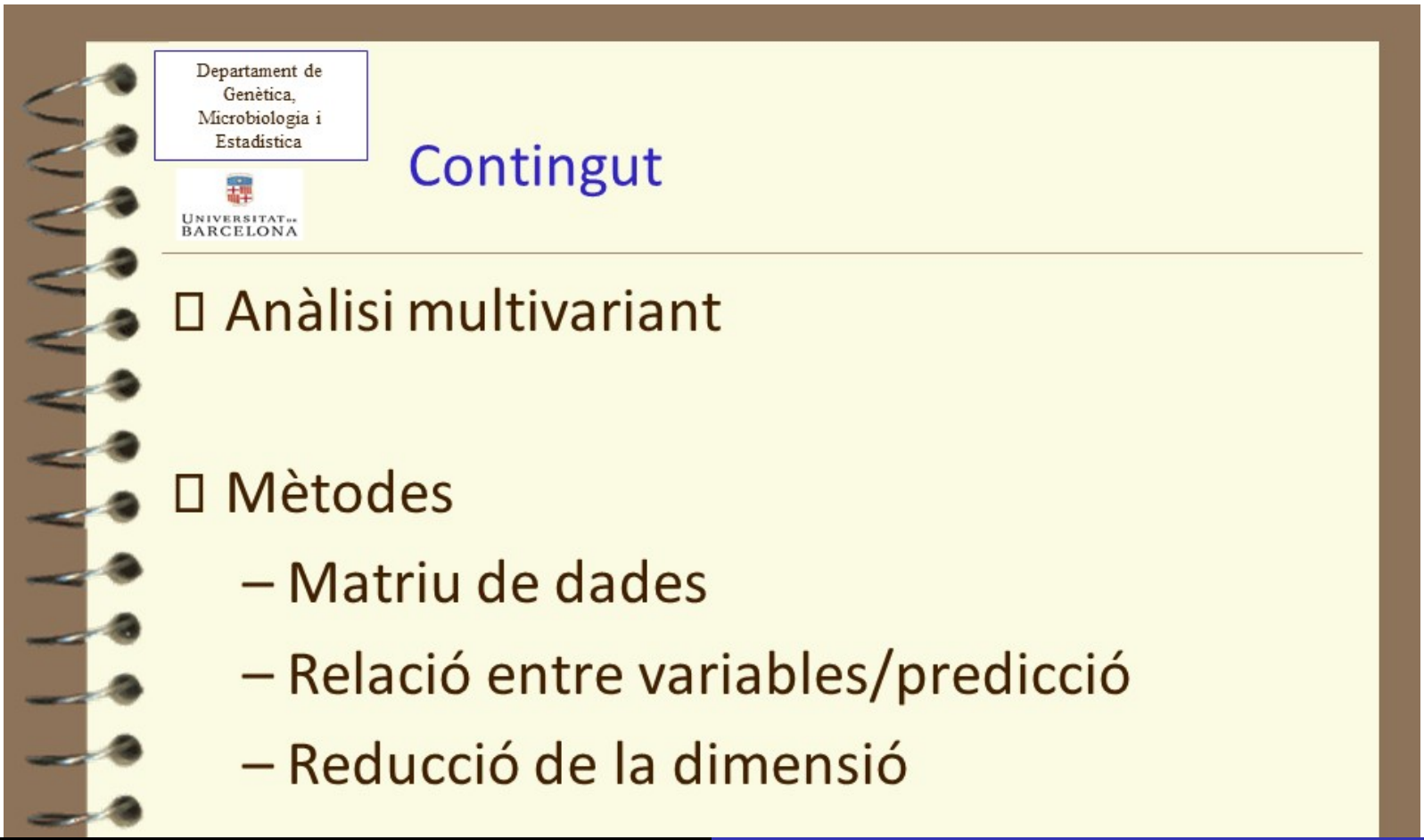
Novembre 2020

Concepte de Data Science

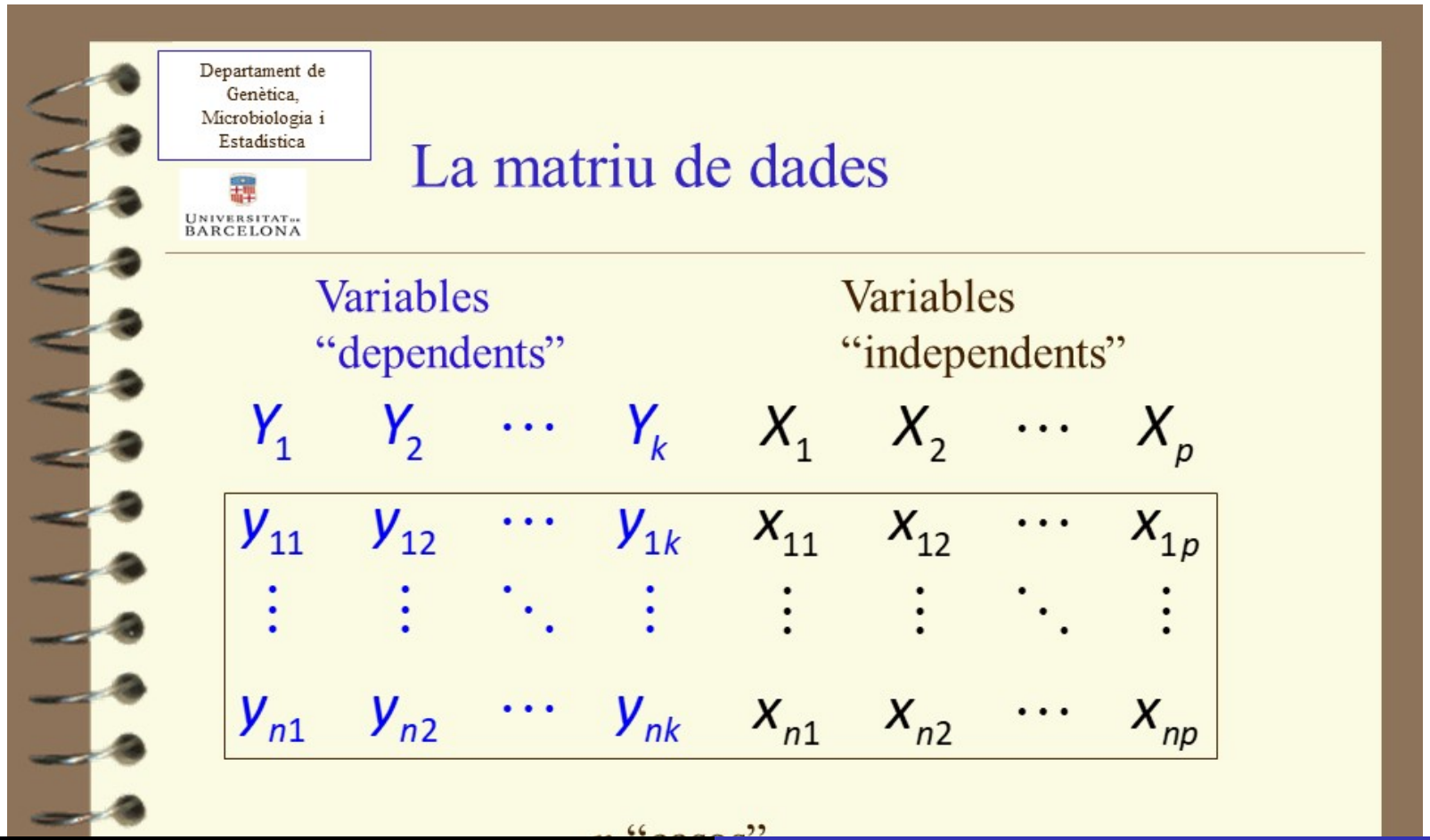
Es pot definir com a: La ciencia de datos es un campo interdisciplinario que involucra métodos científicos, procesos y sistemas para extraer conocimiento o un mejor entendimiento de datos en sus diferentes formas, ya sea estructurados o no estructurados, lo cual es una continuación de algunos campos de análisis de datos como la estadística, la minería de datos, el aprendizaje automático y la analítica predictiva (Wikipedia). Nosaltres l' abordarem amb una perspectiva quelcom més clàssica.



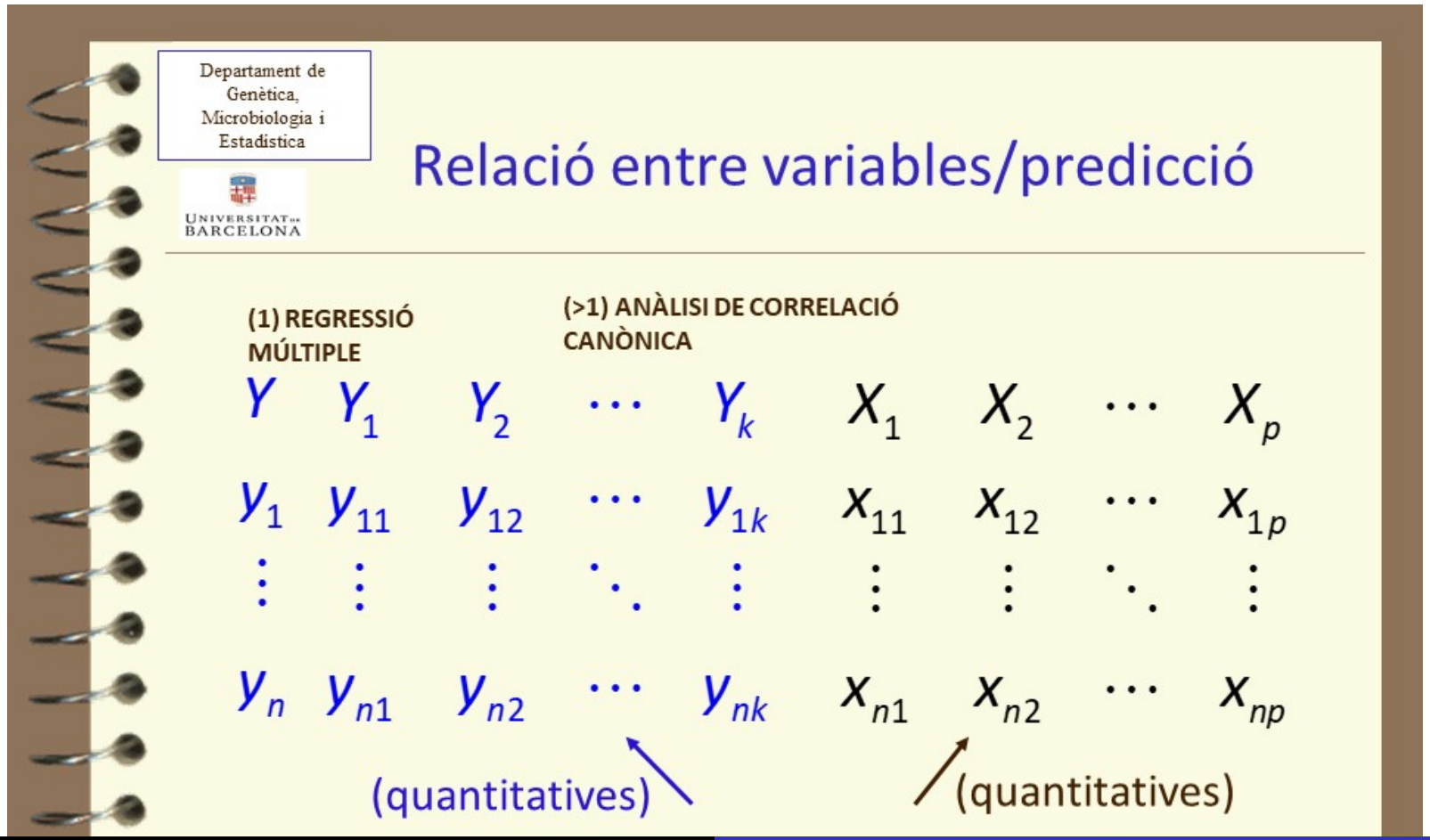
Multivariant: perspectiva clàssica



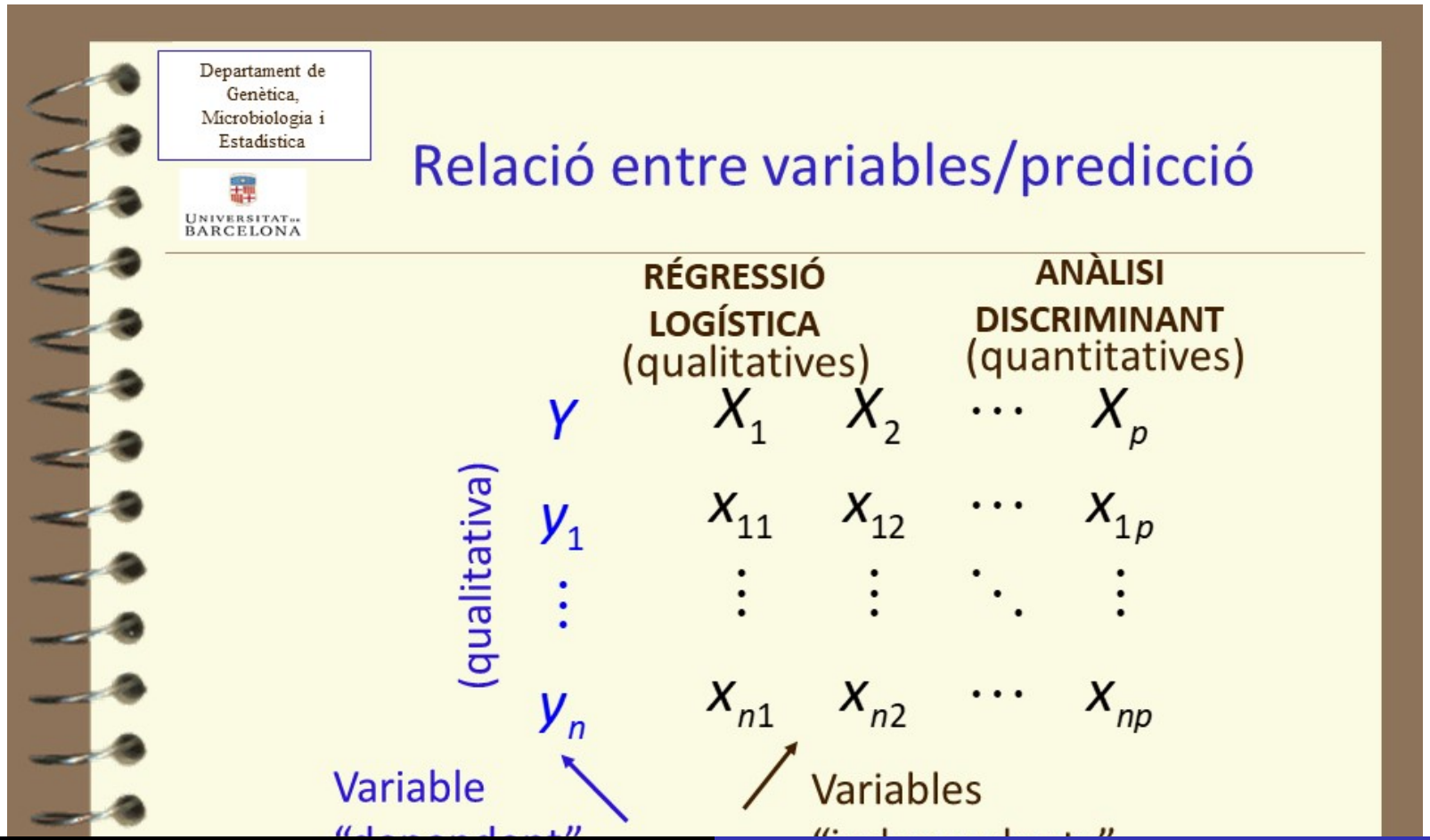
Multivariant



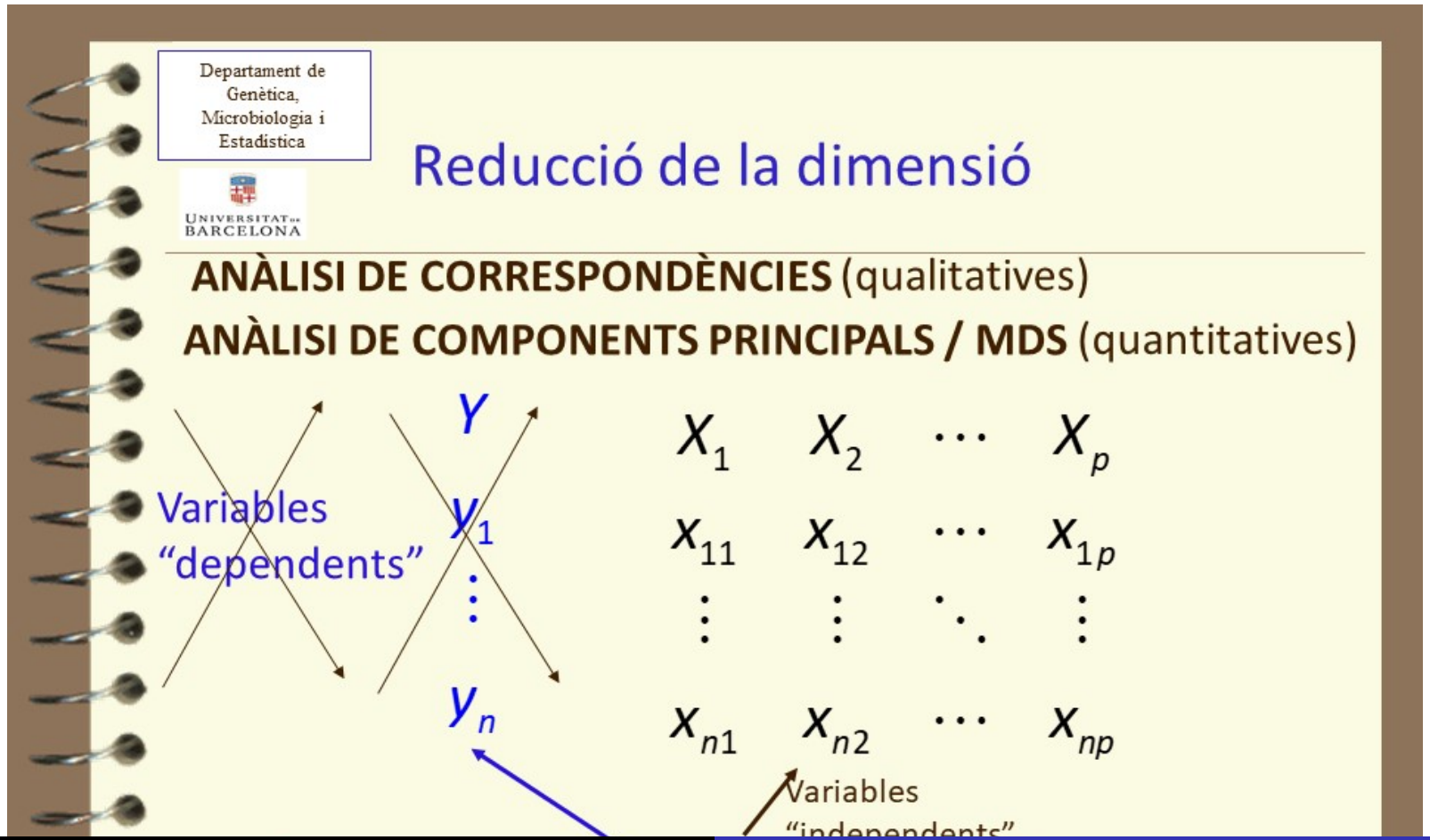
Multivariant



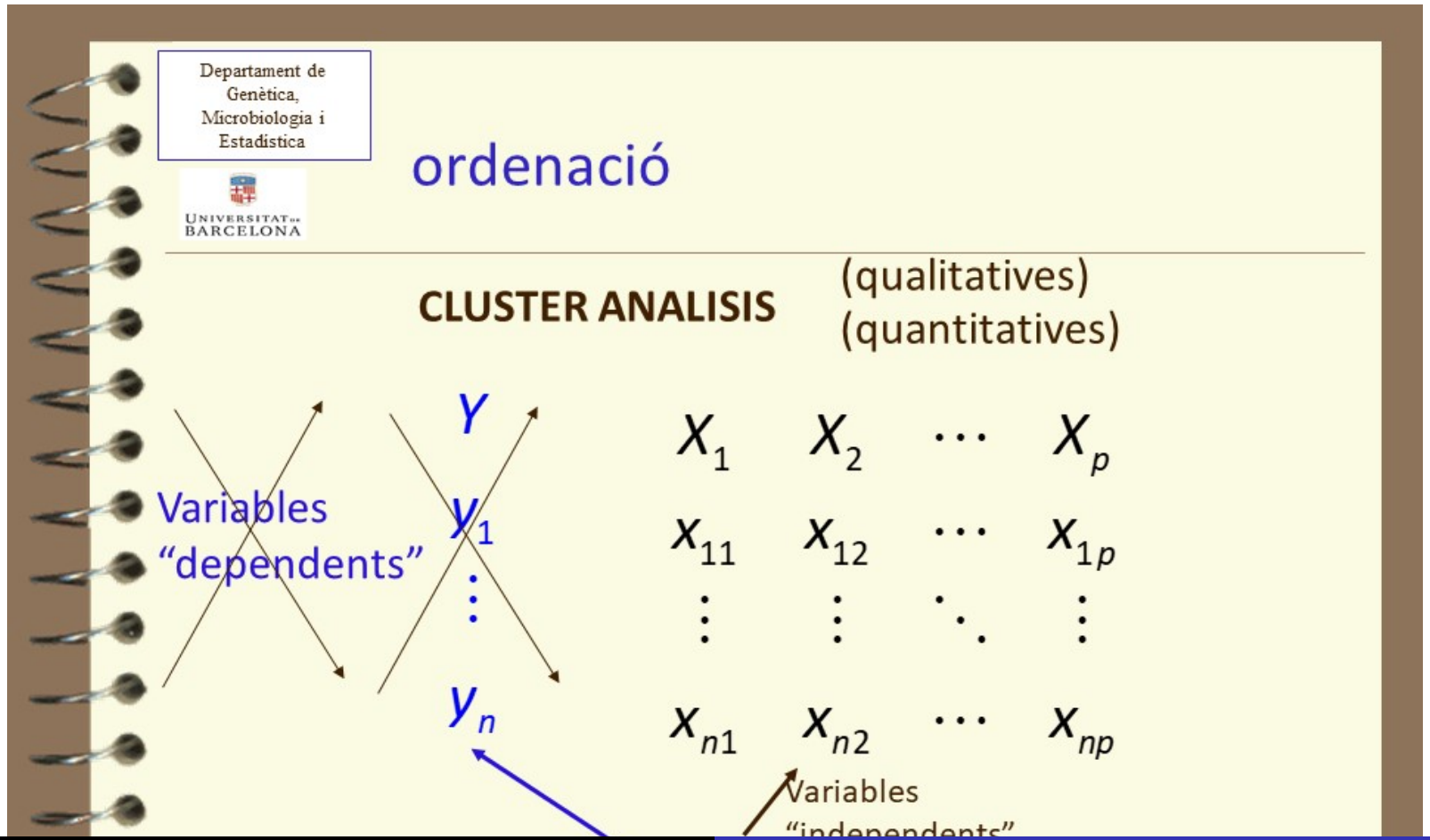
Multivariant



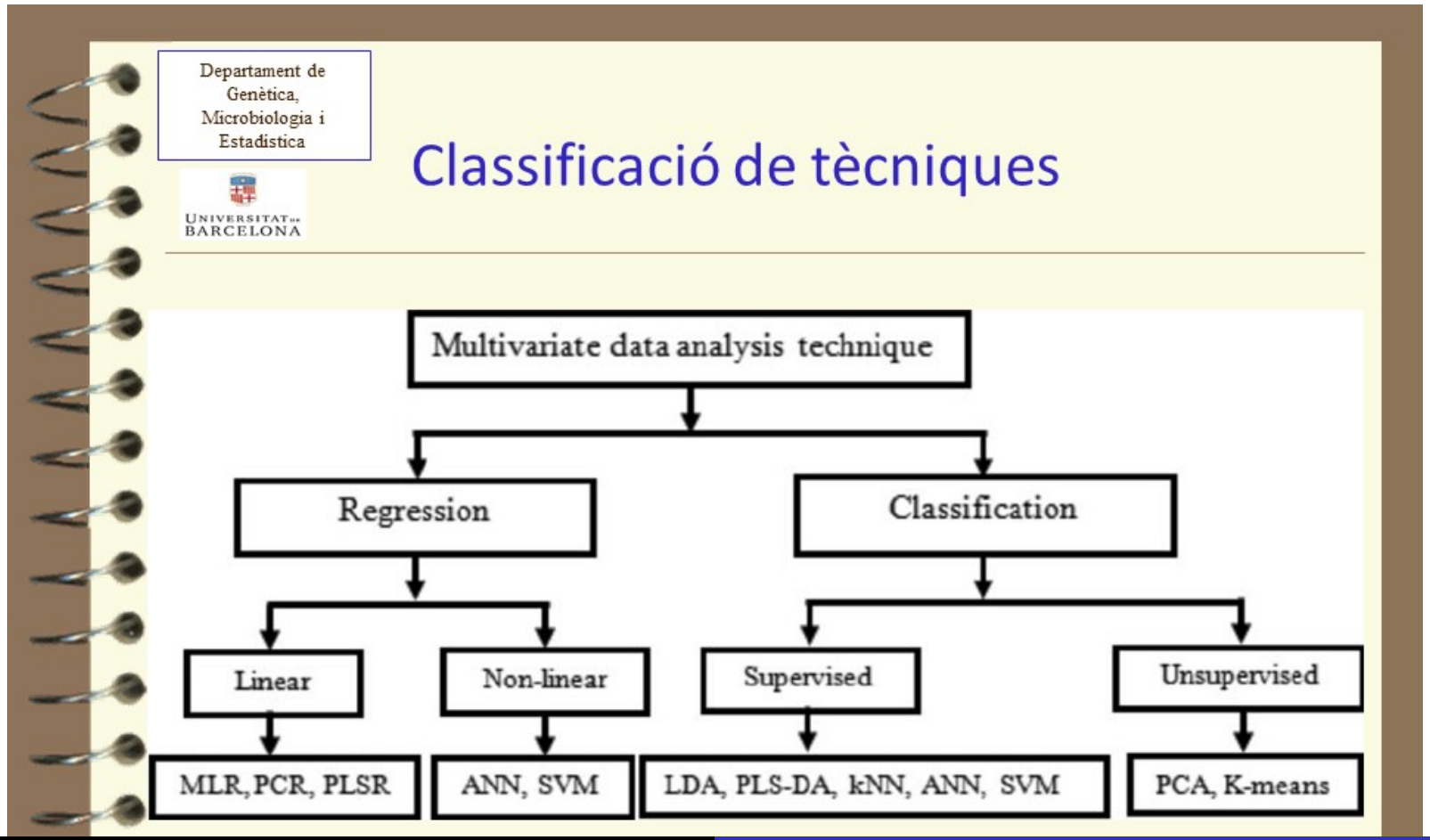
Multivariant



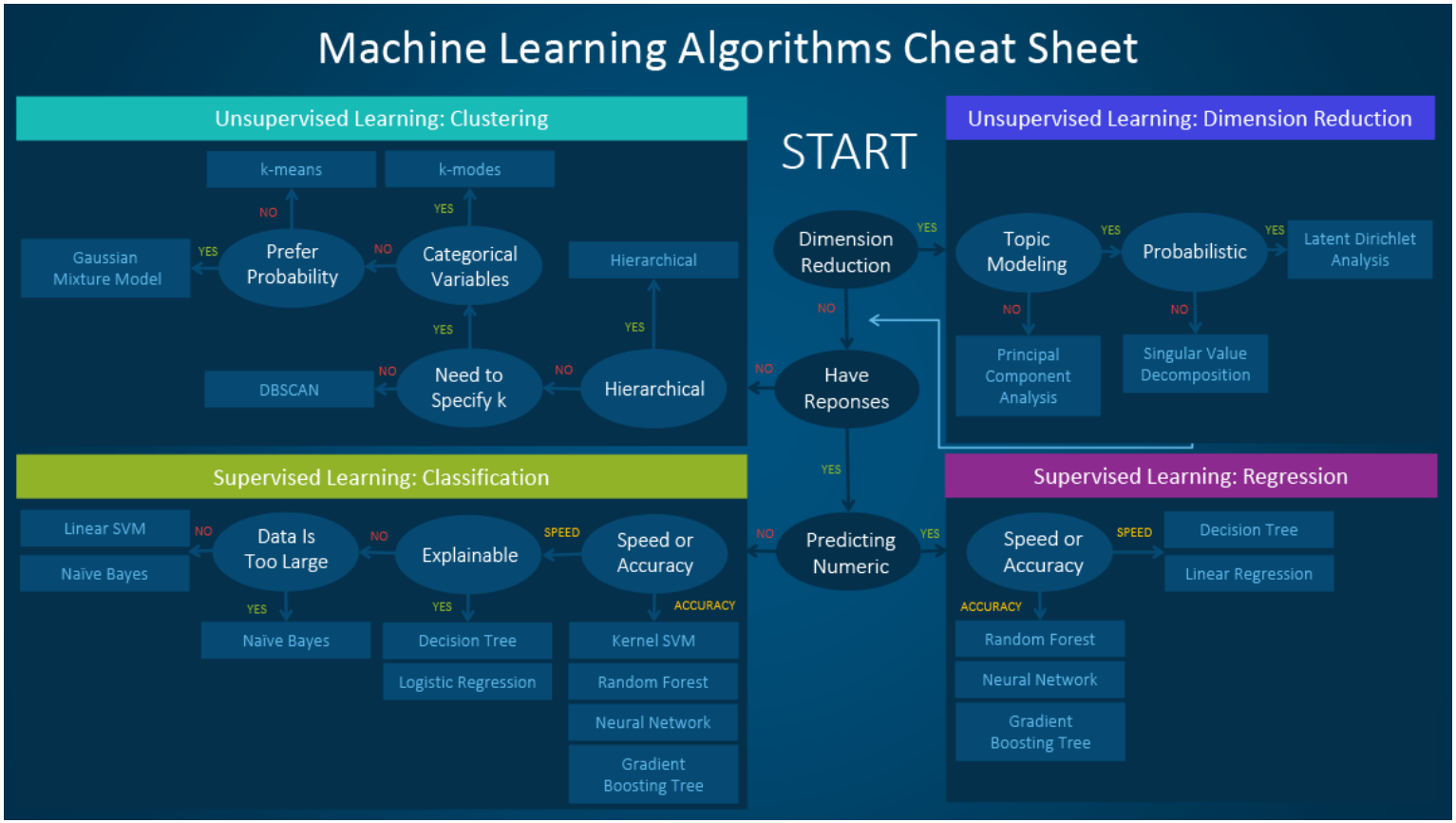
Multivariant



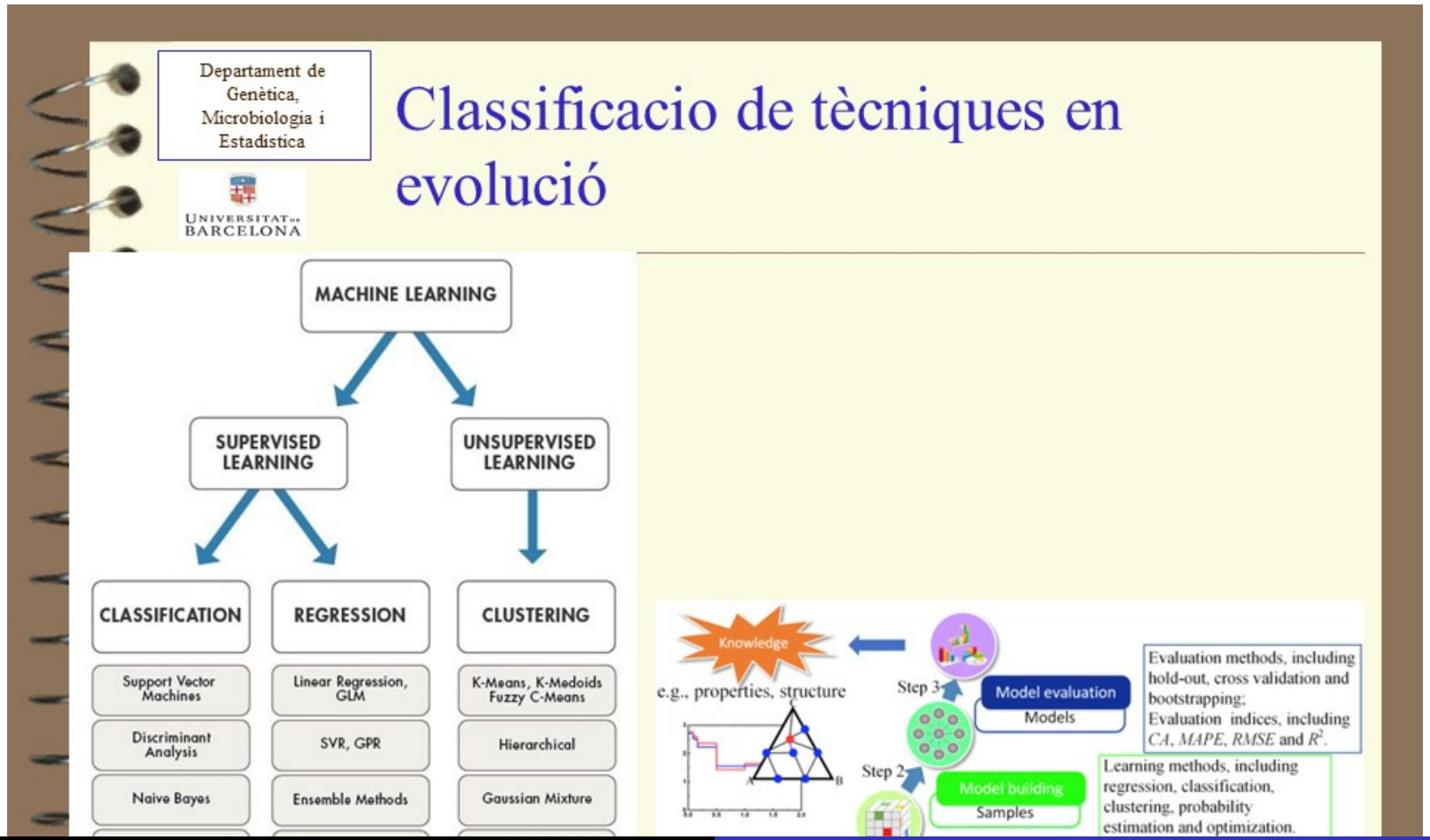
Multivariant



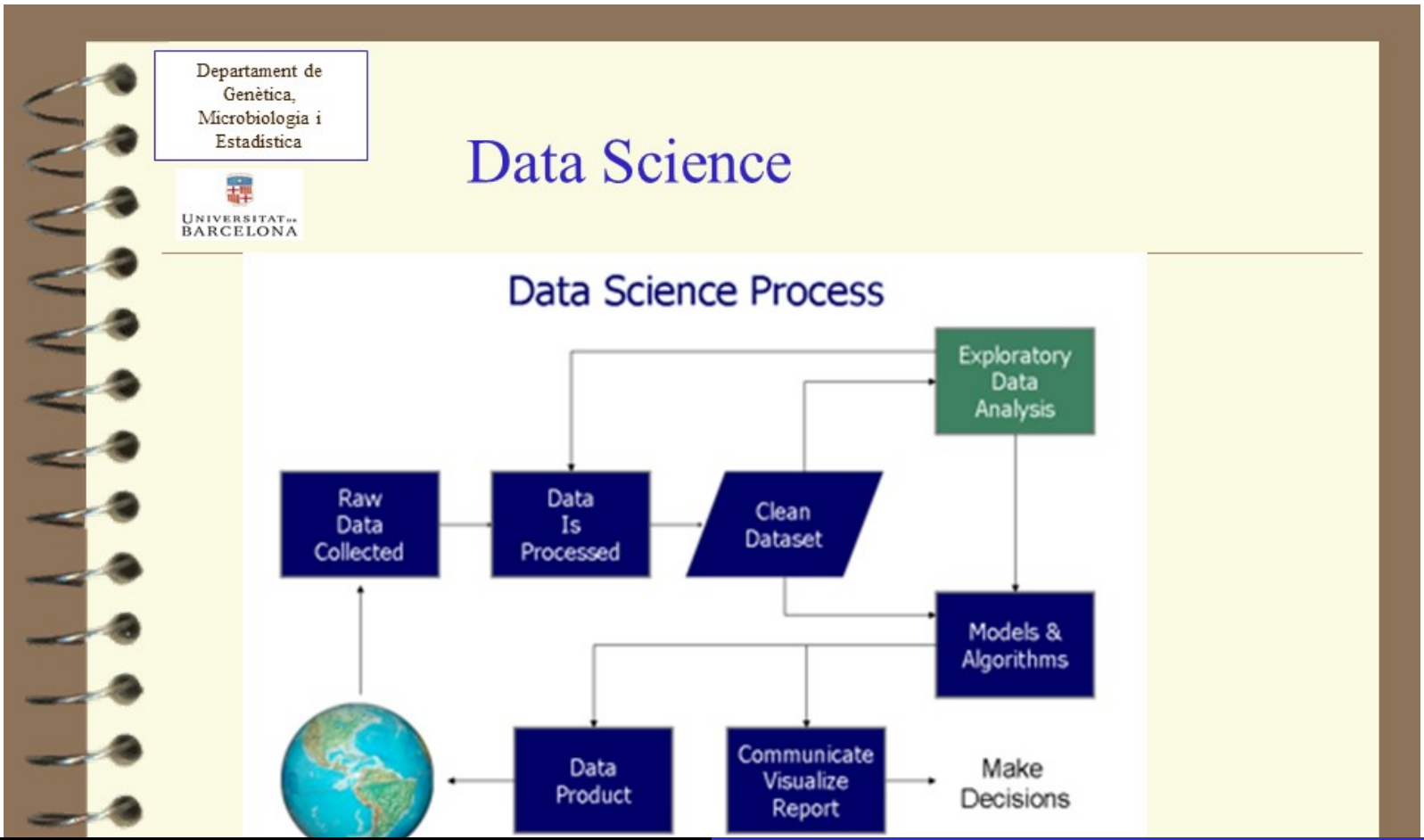
Multivariant



Multivariant



Multivariant



Data Science problems...

- **Supervised learning problem**: we have training data $(x_i, y_i)_{i=1}^N$ and we would like to 1) accurately predict unseen test cases, 2) understand which inputs affect the outcome and how, and 3) assess the quality of our predictions and inferences.

- **Unsupervised learning**: 1) no outcome variable, just a set of predictors (features) measured on a set of samples, 2) objective is more fuzzy, 3) difficulty to know how well you are doing, and 4) can be useful as a pre-processing step for supervised learning.

- **Statistical Learning versus Machine Learning**: Machine learning has a greater emphasis on *large scale* applications and *prediction accuracy*; statistical learning emphasizes *models* and their interpretability, and *precision* and *uncertainty*. Much overlap, much cross-fertilization.

Link amb molta informació i exemples a:

- Supervised and unsupervised learning

Material de referència:...

Aquest material està basat en:

Statistical Learning, 2009.

Lectures: Niels Richard Hansen

- Homepage

Statistical Learning

What is Statistical Learning?

Old wine on new bottles? Is it not just plain statistical inference and regression theory?

New(ish) field on how to use statistics to make the computer "learn"?

A merger of classical disciplines in statistics with methodology from areas known as machine learning, pattern recognition and artificial neural networks.

Major purpose: Prediction – as opposed to .... truth!?
Major point of view: Function approximation, solution of a mathematically formulated estimation problem – as opposed to algorithms.

- The areas mentioned above, machine learning, pattern recognition and artificial neural networks have lived their lifes mostly in the non-statistical literature.
- The theories for *learning* – what would be called estimation in the statistical jargon – have been developed mostly by computer scientists, engineers,physicists and others.

- The quite typical approach of statistics to the problem of inductive inference – the learning from data – is to formulate the problem as a mathematical problem. Then learning means that we want to find one mathematical model for data generation among a set of candidate models, and the one found is almost always found as a solution to an estimation equation or an optimization problems.

- A typical alternative approach to learning is algorithmic, and a lot of the algorithms are thought up with the behavior of human beings in mind. Hence the term "learning" – and hence the widespread use of terminology such as "training data" and "supervised learning" in machine learning.

Iris data

A classical dataset collected by the botanist Edgar Anderson, 1935, *The irises of the Gaspe Peninsula* and studied by statistician R. A. Fisher, 1936 *The use of multiple measurements in taxonomic problems*. Available as the `iris` dataset in the `MASS` library in `R`.

iris

Sepal		Petal		
Length	Width	Length	Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica

Curs de machine learning

Aqui podeu trobar un curs sobre machine learning amb exercicis i teoria:
Machine-Learning (ML)

Un llibre de referencia escrit per Kevin P. Murphy
Machine Learning, A Probabilistic Perspective

Regressió multivariant

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

Referències

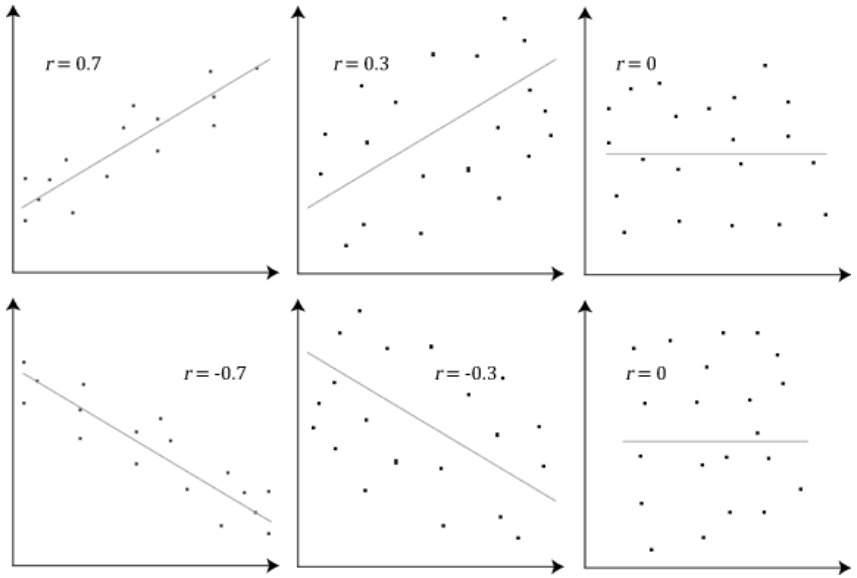
Aquest apartat es basa parcialment en material lliure procedent de:

- Applied linear statistical models. Professor Gunnar Stefansson. Dept. of Mathematics, University of Iceland
- Multiple regression analysis Thomas Alexander GerdsDepartment of Biostatistics, University of Copenhagen

Altres fonts

Covariance, Correlation and dependence

Correlation and dependence between X and Y



Linear Models

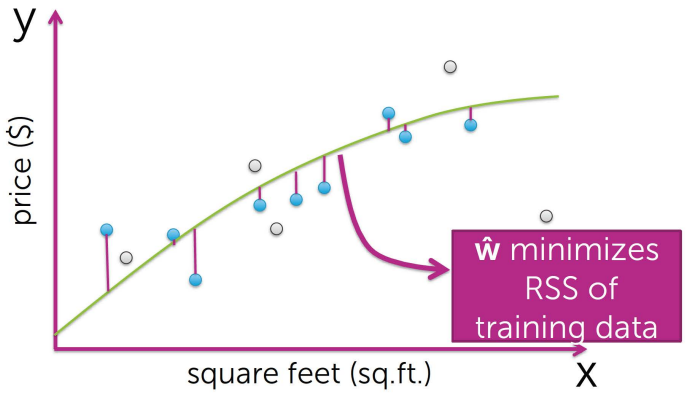
Linear models are defined as $h : \mathbf{R}^n \to \mathbf{R}$ so that $h(x) = a^T x + b \qquad a, x \in \mathbf{R}^n \qquad b \in \mathbf{R}$

Compute training error

1. Define a loss function $L(y, f_{\hat{w}}(\mathbf{x}))$
   - E.g., squared error, absolute error,...
2. Training error
   = avg. loss on houses in training set
   $= \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_{\hat{w}}(\mathbf{x}_i))$

fit using training data

Example:
Fit quadratic to minimize RSS



$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Regression models

Statistical modeling: Regression models

Set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates' or 'features').

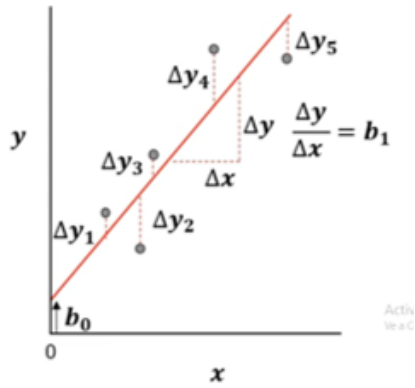
- Ordinary Least Squares (OLS) regression is a frequentist approach to modeling
- The idea is minimize a loss function (L^2), based only on training data
- Estimation of model parameters:

  Model: $y = \beta_0 + \beta_1 x$
  where y is the outcome and x the predictor, $\beta_0 and \beta_1$ are the model coefficients

Simple Regression model

Model parameters set to minimize mismatch at with training data locations.
Model: $y = \beta_0 + \beta x$

- Objective: Find β_0, β_0, fit a linear function, to:
  - Minimize Δy_i over all the data with the L^2 Norm
  - Δy_i is the prediction error:
    $\Delta y_i = y_i - y_{est}$



- Minimize cost function:
  $\sum_{i=1}^{n} (\Delta y_i)^2 = \sum_{i=1}^{n} (y_i - (\beta_o + \beta_1 x))^2$

Simple Linear Regression (SLR) model

Simple Linear regression fits the function:

$y = \beta_0 + \beta_1 x_1$
or $y = \alpha + \beta x_1$ (in old times)
where x_1 is the predictor feature, y the response feature and β_0, β_1 the model parameters

Under the constraint:
$RSS = \sum_{i=1}^{n} (y_i - (\beta_o + \beta_j x_{1i})^2$
minimize the residual sum of squares (RSS) over the training data

The models is composed by:

Fixed numbers, x_i and Random variables: $Y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$ or:
$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$
with $\epsilon_i \sim N(0, \sigma^2)$ independent and identically distributed (i.i.d.)

So: y_i-values are outcomes of the random variable Y_i, but x_i-values are constants.

Simple linear regression: Have n pairs $(x_1, y_1), \ldots, (x_n, y_n)$ and want "best"fitting line.
Estimation (OLS):

$$S(\beta_0, \beta_1) = \sum_i (y_i - (\beta_0 + \beta_1 x_i))^2$$

Minimize S over β, β_1 to get

$$\beta_0 = \bar{y} - b\bar{x} = intercept$$

$$\beta_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = slope$$

Usual regression assumptions

The assumptions (which may all fail) are:

- x-values are constants (no error)
- Linearity
- Constant variance
- Gaussian
- Independence

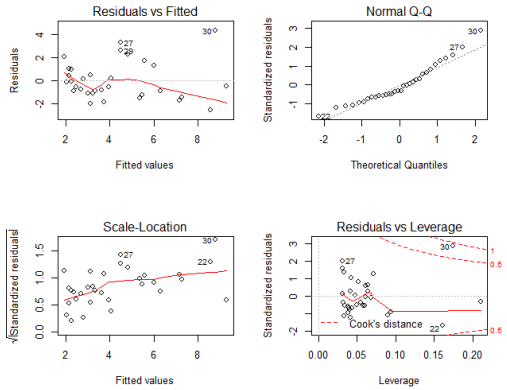
Will test these and modify accordingly (use $plot(obj < -lm(y\ x))$ or hypothesis test)

Examples of problems: Fish growth (nonlinear); Bird counts (nonnormal); fuel consumption (heteroscedastic); stock prices (autocorrelated)

Goodness of fit and diagnostics

Verifying Simple Linear Regression assumptions...

Will derive tests for nonlinearity (lack-of-fit), normality, homoscedasticity, independence, outliers, etc etc. This is (mainly) based on residuals $e_i = y_i - \hat{y}_i$ or variations thereof
Some concepts: Standardized residuals, studentized residuals, deleted residuals etc etc.

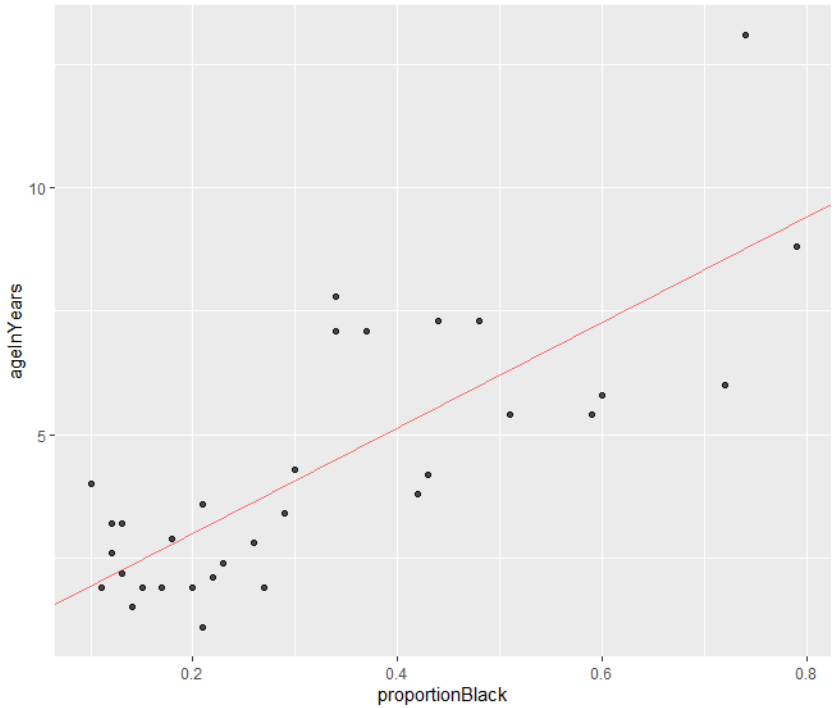


Simplest diagnostics: Plot residuals in all possible ways
Note: It is never enough to fit a model or use it for predictions. One must always also verify whether the model is adequate.

Simple Linear Regression (SLR) example

See a simple regression example: bivariate regression with the lion nose pigmentation data set

Simple Linear Regression (SLR) example



Lion nose pigmentation data set example: y=age, x= proportion black

Multiple regression analysis

The distribution of a single response variable (Y) is related to several explanatory variables, X_1, X_2, ..., by a mathematical function, f:

$$Y \sim f(X_1, X_2, X_3, ...)$$

Examples:

- multiple linear regression (Y: continuous)
- multiple logistic regression (Y: binary)
- multiple Poisson regression (Y: count)
- multiple Cox regression (Y: survival)

Typical data structure for regression analysis

For each of n subjects, measurements of the response Y_i and $p-$many covariates $X_{11}, \ldots, X_{1p}$:

Subject id	X_1	$\ldots$	X_p	Y
1	X_{11}	$\ldots$	X_{1p}	Y_1
2	X_{21}	$\ldots$	X_{2p}	Y_2
3	X_{31}	$\ldots$	X_{3p}	Y_3
.	.	$\ldots$	.	.
n	X_{n1}	$\ldots$	X_{np}	Y_n

No missing values

Exemple guia de regressió multivariant: Air polution

The dataset contains hourly measurements of air pollutant concentrations, wind speed and wind direction collected at the Marylebone (London) air quality monitoring supersite between 1st January 1998 and 23rd June 2005.

```
library(openair)
#see the data at ??mydata and copy in quality_air
quality_air<-mydata #data wrangling
toSeason <- function(dat) {
  scalarCheck <- function(dat) {
    m <- as.POSIXlt(dat)$mon + 1       # correct for 0:11 range
    d <- as.POSIXlt(dat)$mday          # correct for 0:11 range
    if ((m == 3 & d >= 21) | (m == 4) | (m == 5) | (m == 6 & d < 21)) {
      r <- 1
    } else if ((m == 6 & d >= 21) | (m == 7) | (m == 8) | (m == 9 & d < 21)) {
      r <- 2
    } else if ((m == 9 & d >= 21) | (m == 10) | (m == 11) | (m == 12 & d < 21)) {
      r <- 3
    } else {
      r <- 4
    }
    r  }
  res <- sapply(dat, scalarCheck)
  res <- ordered(res, labels=c("Spring", "Summer", "Fall", "Winter"))
  invisible(res)}
```

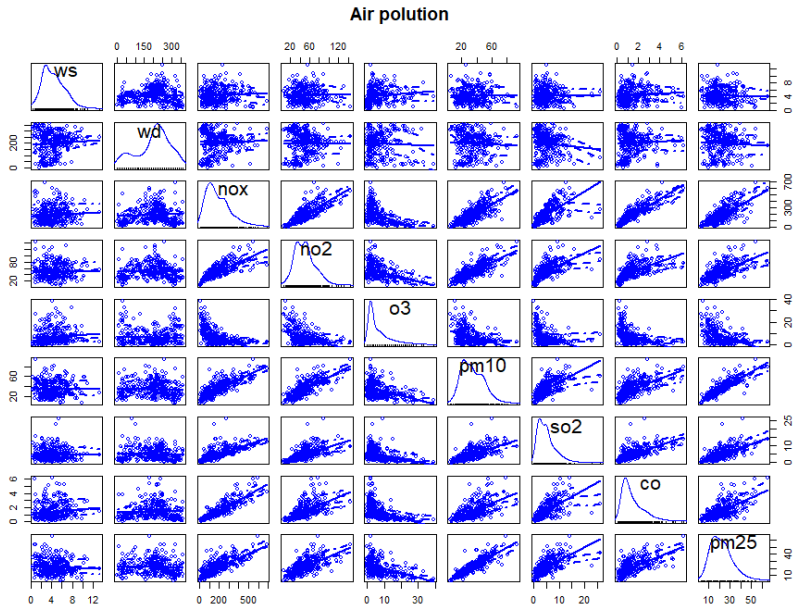
Exemple guia de regressió multivariant: Air polution

The dataset contains hourly measurements of air pollutant concentrations, wind speed and wind direction collected at the Marylebone (London) air quality monitoring supersite between 1st January 1998 and 23rd June 2005.

```
#select same part of data
res<-toSeason(dat=quality_air$date)
quality_air$season <- res

#select only few data without NA (a random sample)
set.seed(197) #fix the seed
quality_air1<- na.omit(quality_air[sample(nrow(quality_air), 500), ])

#see the variable names
names(quality_air)
#"date"   "ws"    "wd"    "nox"   "no2"   "o3"    "pm10"  "so2"   "co"    "pm25"  "
      season"

#observe the data set
??mydata
```

Carrega i visualitzacio de les dades

```
#Carrega i visualitzaci de les dades
library(car)
scatterplotMatrix(quality_air1[,2:10], diag='boxplot',main = "Air polution")
```



Linear regression theory

Models the distribution of a continuous-type/quantitative response variable Y_i of subject i in relation to one or more subject specific explanatory variables $X_{i1}, \ldots, X_{ip}$ as follows (additive model):

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \epsilon_i.$$

- Intercept: β_0
- Regression coefficients: $\beta_1, \ldots, \beta_p$
- Error term ϵ_i has mean zero, it captures the residual variability

A typical aim is to study changes of the mean of Y_i under changes of $X_{i1}, \ldots, X_{ip}$.

SLR in matrix form

Convert the SLR in a matrix form to generalize to Multiple regression
$\mathbf{y} \in R^n$ = vector of measurements

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

the "$\mathbf{X}$-matrix"
min $\sum(y_i - (\beta_0 + \beta_1 x_i))^2$ is equivalent to finding

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

Point estimate as a projection

to mininmize the cost function $||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2$
Number notation: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$

Multiple regression

The model can be written in matricial form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$

using,

$$X = \begin{pmatrix} 1 & X_{11} & X_{12} & \dots & X_{1p} \\ 1 & X_{21} & X_{22} & \dots & X_{2p} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & X_{n1} & X_{n2} & \dots & X_{np} \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \dots \\ \beta_p \end{pmatrix} \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \dots \\ \epsilon_n \end{pmatrix} \quad Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_n \end{pmatrix}$$

The coefficients β_i are estimated using the OLS criteria (Ordirary Least Squares):

$$\min \sum_{i=1}^{n} (Y_i - (\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip}))^2$$

to obtain: $\hat{\beta} = (X^T X)^{-1} X^T Y$

Multiple regression

The model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$

where $\mathbf{X}$ in an $n \times p$ matrix.

Example: multiple linear regression model: $y_i = \beta_0 + \beta_1 x_i + \beta_2 w_i + \beta_3 x_i$.

More examples: Estimate single intercept, many slopes; Test whether multiple lines are all parallel; ...

Used in all fields of biology (ecology, genetics, ...), medicine, etc, ...
Need to develop point estimates, methods of validation and testing.

Matrix solution: Normal equations

The point estimate is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

Is named as Normal equations
which is always unbiased (if the mean of the Y-s is correct)

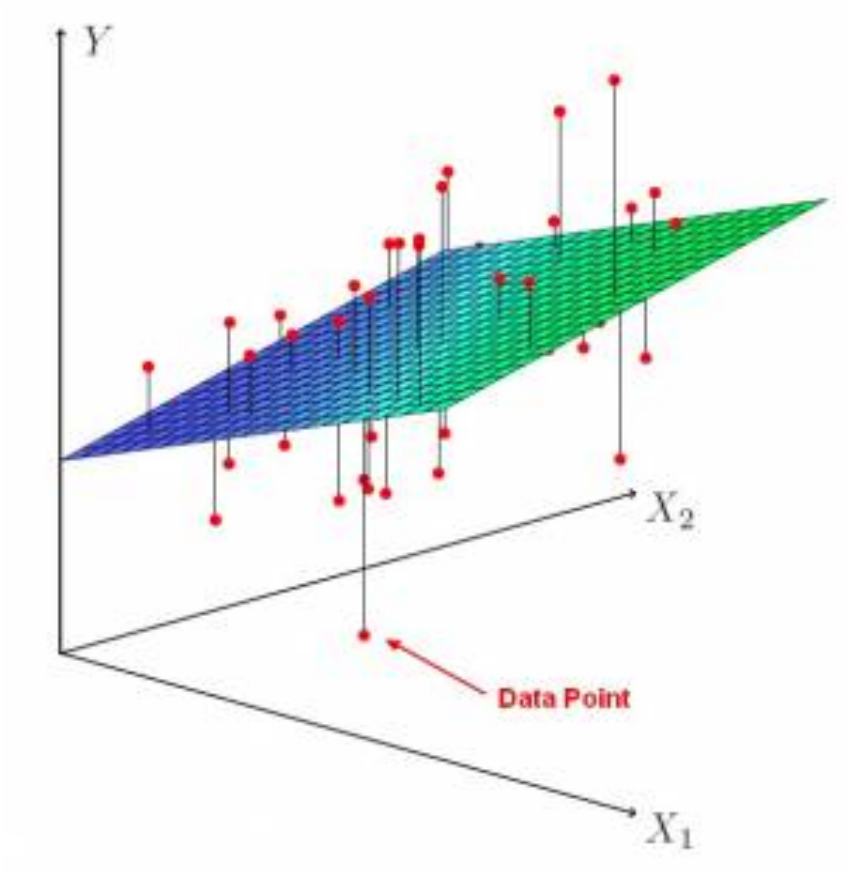
$$E\left[\hat{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}$$

and has variance-covariance matrix

$$V\left[\hat{\boldsymbol{\beta}}\right] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}.$$

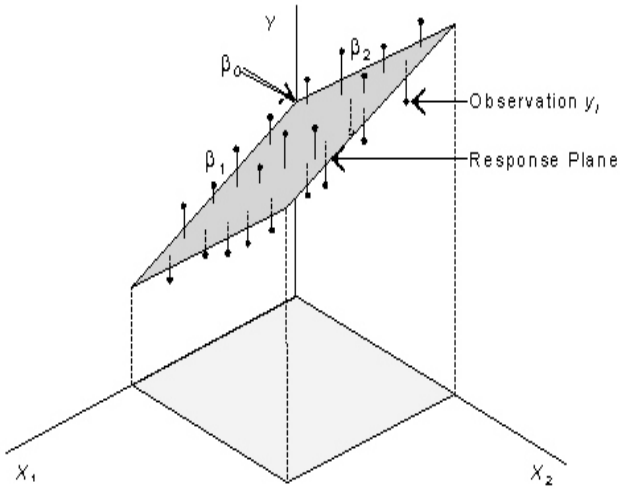
(if the variance assumptions are correct)
and is multivariate Gaussian (if the Y-values are Gaussian).

Multiple regression: hyperplan



Multiple regression

Interpretation of estimation in the hyperplan



See image at:
https://www.ck12.org/c/statistics/multiple-regression/lesson/Multiple-Regression-ADV-PST/

Fit a Linear model for pm25

```
mod1<- lm(pm25~ ws + wd + nox + no2 + o3 + so2 + co, data= quality_air1)
summary(mod1)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.886294   1.501952   7.914 4.18e-14 ***
ws          -0.307276   0.154734  -1.986  0.04791 *
wd          -0.011509   0.004295  -2.680  0.00775 **
nox          0.060273   0.008184   7.365 1.54e-12 ***
no2         -0.002908   0.028224  -0.103  0.91800
o3           0.045912   0.064177   0.715  0.47489
so2          1.017520   0.166775   6.101 3.06e-09 ***
co          -1.641455   0.656728  -2.499  0.01294 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.755 on 318 degrees of freedom
Multiple R-squared:  0.6248,        Adjusted R-squared:  0.6166
F-statistic: 75.65 on 7 and 318 DF,  p-value: < 2.2e-16
```

Multiple regression: Interpretation of model coefficients

The estimated model for pm25 is:

pm25 = 11.89 -0.307 ws + ... -1.64 co

Parameters are interpreted as the increase in the response variable when the corresponding predictive variable is incremented by one unit.

Eg. Interpretation for ws (in function of unities of ws):

- The estimated value of the (pending) parameter is -0.307.
- This means that we expect an increase / decrease of -0.307 units of pm25 (Y) for every unities of increase in the unities of ws if the other variables remain constant.
- In regression, the number of observations must be greater than the number of predictive variables or the previous matrix calculations cannot be performed.

Multiple regression: ANOVA

As we did at ANOVA we can divide the total observed variability (SS_Total) into two additive components:

- SSRegression = Variability in Y explained by the linear relationship with $X1, \ldots, Xp$:
- SSResidual = Variability in Y not explained by the linear relationship with $X1, \ldots, Xp$ measured as the difference between each Yi observed and the value predicted by the regression model.

Table 6.1 Analysis of variance table for a multiple linear regression model with an intercept, p predictor variables and n observations

Source of variation	SS	df	MS
Regression	$\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2$	p	$\frac{\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2}{p}$
Residual	$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$	$n-p-1$	$\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{n-p-1}$
Total	$\sum_{i=1}^{n}(y_i-\bar{y})^2$	$n-1$	

Multiple regression: ANOVA

ANOVA table for regression:

```
anova(mod1)
```

```
Analysis of Variance Table

Response: pm25
           Df  Sum Sq Mean Sq  F value    Pr(>F)
ws          1    27.4    27.4   0.5996   0.43929
wd          1    36.4    36.4   0.7977   0.37246
nox         1 22051.7 22051.7 483.2659 < 2.2e-16 ***
no2         1   196.8   196.8   4.3135   0.03861 *
o3          1   109.0   109.0   2.3891   0.12318
so2         1  1458.6  1458.6  31.9655 3.494e-08 ***
co          1   285.1   285.1   6.2472   0.01294 *
Residuals 318 14510.5    45.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The value that appears in the summary of the linear model as Residual standard error corresponds to the square root of the MSE (estimation of $\sigma^2 = 45.6$). This allows us to construct a global contrast on the regression (based on a F test)

Global hypothesis on regression

$H_0 : \beta_1 = \beta_1 == \cdots = \beta_p = 0$
$H_1 : \beta_i \neq 0$

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.886294   1.501952   7.914 4.18e-14 ***
ws          -0.307276   0.154734  -1.986  0.04791 *
wd          -0.011509   0.004295  -2.680  0.00775 **
nox          0.060273   0.008184   7.365 1.54e-12 ***
no2         -0.002908   0.028224  -0.103  0.91800
o3           0.045912   0.064177   0.715  0.47489
so2          1.017520   0.166775   6.101 3.06e-09 ***
co          -1.641455   0.656728  -2.499  0.01294 *
---
Residual standard error: 6.67 on 300 degrees of freedom
Multiple R-squared:  0.6061,        Adjusted R-squared:  0.5969
F-statistic: 65.96 on 7 and 300 DF,  p-value: < 2.2e-16
```

In our case p=2.2e-16, so we can reject the null hyphotesis $H_1 : \beta_i \neq 0$

Hypothesis on regression coefficients

To contrast the hypothesis on regression coefficients: $H_0 : \beta_i = 0$ vs $H_1 : \beta_i \neq 0$ we can use the statisic:

$$t = \frac{b_i}{se(b_i)}$$

where $se(b_i)$ is the estimated standard error of the estimator for the ith coefficient. Which under H_0 follows a Student's t distribution with $n - (p + 1)$ degrees of freedom.

```
summary(mod1)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.886294   1.501952   7.914 4.18e-14 ***
ws          -0.307276   0.154734  -1.986  0.04791 *
wd          -0.011509   0.004295  -2.680  0.00775 **
nox          0.060273   0.008184   7.365 1.54e-12 ***
no2         -0.002908   0.028224  -0.103  0.91800
o3           0.045912   0.064177   0.715  0.47489
so2          1.017520   0.166775   6.101 3.06e-09 ***
co          -1.641455   0.656728  -2.499  0.01294 *
```

The coefficients with * (p<0.05) are significatives and we can reject H_0.

Is equivalent to comparing the increase that supposes, in the explained variability, to introduce the term in question in front of not introducing it.

Explained variance R^2 (Coefficient of determination)

- Individually an explanatory variable may be significantly related to the response variable, but not be a significant predictor in the multiple linear regression model.
- An individual variable may NOT be significantly related to the response variable, but in a multiple linear regression model it may be
- The slope of a response variable can change sign if we go from a simple regression to a multiple
- The proportion of the total variability of Y explained by the regression model is: $R^2 = \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{SS_{Residual}}{SS_{Total}}$
- The closer R2 to 1 the more suitable the model will be.
- The closer to 0 the worse. The response variable does not fit linearly with the predictive variables.

```
Residual standard error: 6.67 on 300 degrees of freedom Multiple R-squared: 0.6061, Adjusted
R-squared: 0.5969 F-statistic: 65.96 on 7 and 300 DF, p-value: < 2.2e-16
```

In our case $R^2 = 0.6061$

Explained variance R^2

But ...

- R^2 is not a good measure to compare models with different number of explanatory variables.
- As we add more predictors R^2 always increases.
- There is a correction that takes into account the number of predictors (adjusted R^2).

```
Residual standard error: 6.67 on 300 degrees of freedom Multiple R-squared: 0.6061, Adjusted
R-squared: 0.5969 F-statistic: 65.96 on 7 and 300 DF, p-value: < 2.2e-16
```

In our case Adjusted R-squared: 0.5969

Prediction and confidence intervals

We can obtain confidence intervals by estimating the model parameters.

```
confint(mod1,level=0.95)
```

```
                 2.5 %       97.5 %
(Intercept)  8.93127596 14.841312873
ws          -0.61170879 -0.002844059
wd          -0.01995865 -0.003059623
nox          0.04417077  0.076374709
no2         -0.05843736  0.052621170
o3          -0.08035312  0.172177068
so2          0.68939680  1.345642287
co          -2.93353552 -0.349374421
```

Make predictions

```
predict(mod1,as.data.frame(quality_air1[1,2:9]),interval='confidence')
predict(mod1,as.data.frame(quality_air1[1,2:9]),interval='prediction')
```

```
> predict(mod1,as.data.frame(quality_air1[1,2:9]),interval='confidence')
       fit     lwr      upr
1 19.18804 18.0208 20.35527
> predict(mod1,as.data.frame(quality_air1[1,2:9]),interval='prediction')
       fit      lwr      upr
1 19.18804 5.846655 32.52942
```

Model validation

- Validate mainly using various residuals
- Investigate assumptions, but now also investigate influential observations
- DFFITS etc
- Hat matrix H, particularly the leverage values h_{ii}

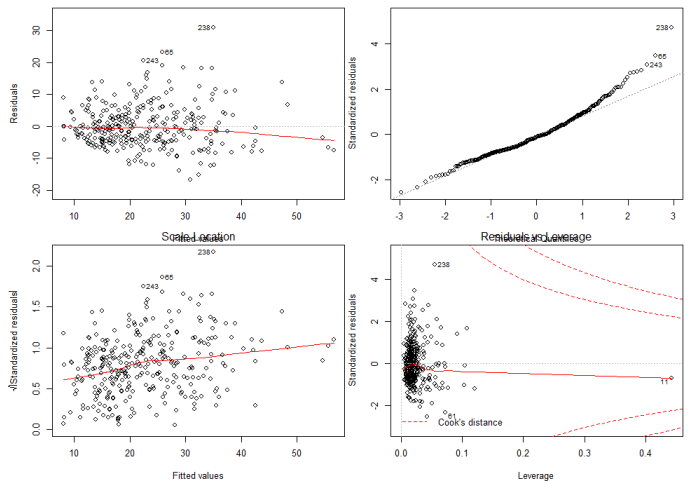
$$H = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$$

- projects $\mathbf{y}$ to $\hat{\mathbf{y}}$
- Investigate collinearity

Model diagnosis

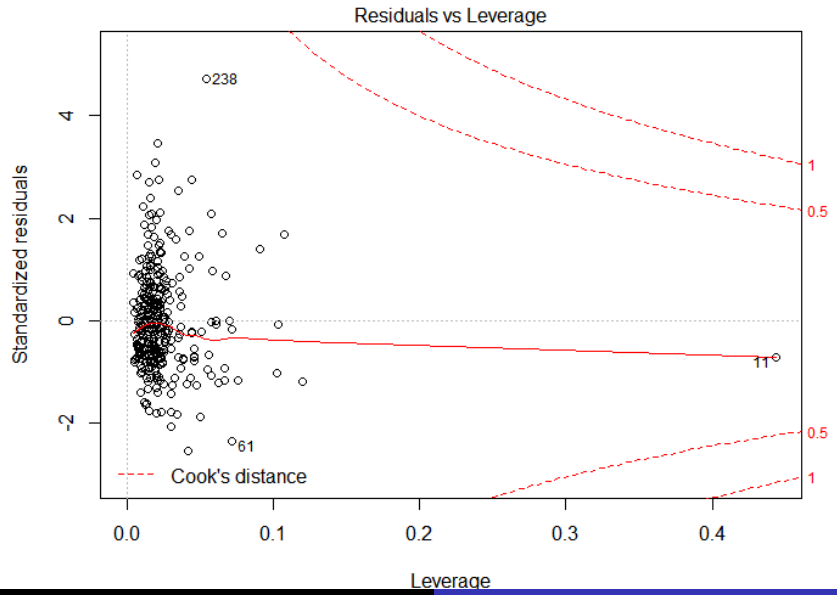
Interval estimation and hypothesis testing require a set of assumptions about model error terms. See: Normality, Homocedasticity and Independence. These assumptions can be verified through the usual graphs on the model residues. See information about interpretation of Leverage, Cook's distance and other dignosis at:REGRESSION DIAGNOSIS and Leverage, Cook's distance, Leverage theory and Cook's distance theory

```
par(mfrow=c(2,2))
plot(mod1)
```



Model diagnosis Leverage and Cook's distance

- Cook's distance: Data points with large residuals (outliers) and/or high leverage may distort the outcome and accuracy of a regression. Cook's distance measures the effect of deleting a given observation. Points with a large Cook's distance are considered to merit closer examination in the analysis.
- High-leverage points: High-leverage points are those observations, if any, made at extreme or outlying values of the independent variables such that the lack of neighboring observations means that the fitted regression model will pass close to that particular observation.



Multicolineality

An additional problem is that the predictive variables are correlated with each other. If this fact is given we speak of Multicolineality and it can influence the results of the regression in two main ways:

- Makes model coefficient estimates unstable.
- It increases the standard errors of the estimated slopes and therefore the
- confidence intervals are more inaccurate.

To detect multicollinearity, the matrix of correlations between the predictive variables can be analyzed (see above)

One possibility to determine if multicollinearity is a problem is to calculate the VIF (variance inflation) corresponding to each variable. A value greater than 5 and especially a value greater than 10 is a serious multicollinearity problem.
The calculation of the VIF on a predictive variable X_i
is performed from the formula: $VIF = \frac{1}{1-R_i^2}$

on R_i^2 is the coefficient of determination of the regression where X_i is the dependent variable and the other predictors act as independent.

```
1 vif(mod1)
```

```
> vif(mod1)
      ws       wd      nox      no2       o3      so2       co
1.127147 1.045068 7.416214 2.381755 1.433830 2.625194 4.837558
```

Model selection

- Many x-variables?
- Need to choose subset for inclusion
- Look at all subsets?
- How should quality of fit be measured?
- Forward and backwards stepwise regression.

Variable selection

There is controversy among statisticians about the appropriateness of applying techniques for automatically selecting (forward, backward, etc.) subsets of predictive variables. Problems may arise due to the large number of tests performed. There are many methods, but one very often used is the measure used to quantify the goodness of the model is the Akaike information criterion (AIC). Given a set of candidate models, the preferred model is the lowest AIC.

```
step(mod1) #Using stepwise regression:
```

```
lm(formula = pm25 ~ ws + wd + nox + so2 + co, data = quality_air1)

    Coefficients:
    (Intercept)           ws           wd          nox
       12.17435     -0.27982     -0.01128      0.05788
            so2           co
        1.03070     -1.58302
```

And now we have a 5 variables model: pm25 ws + wd + nox + so2 + co

Variable selection

It is possible that the selection is easy to introduce variables in it to eliminate them. Or compare previous model with final model.

```
#forward regression
step(mod1,direction='forward')
#other
one.lm.full<-lm(pm25~ ws + wd + nox + no2 + o3 + so2 + co,data=quality_air1)
two.lm.null<-lm(pm25~1,data=quality_air1)
pm25.step<-step(two.lm.null,scope=list(upper=one.lm.full),direction='forward')
summary(pm25.step)
```

```
#final model selected pm25.step
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.174351   1.210539  10.057  < 2e-16 ***
nox          0.057880   0.006570   8.809  < 2e-16 ***
so2          1.030705   0.157236   6.555 2.23e-10 ***
co          -1.583017   0.622413  -2.543  0.01145 *
wd          -0.011285   0.004252  -2.654  0.00836 **
ws          -0.279816   0.147824  -1.893  0.05927 .
Residual standard error: 6.739 on 320 degrees of freedom
Multiple R-squared:  0.6242,        Adjusted R-squared:  0.6183
F-statistic: 106.3 on 5 and 320 DF,  p-value: < 2.2e-16
```

Variable selection: Lasso regression and others

LASSO regression stands for Least Absolute Shrinkage and Selection Operator. The algorithm is another variation of linear regression, just like ridge regression. We use lasso regression when we have a large number of predictor variables. LASSO is complex to carry out.See at:
LASSO
Finally we present other method to try to check all possible models perfomance using the library(olsrr). See at VARIABLE SELECTION

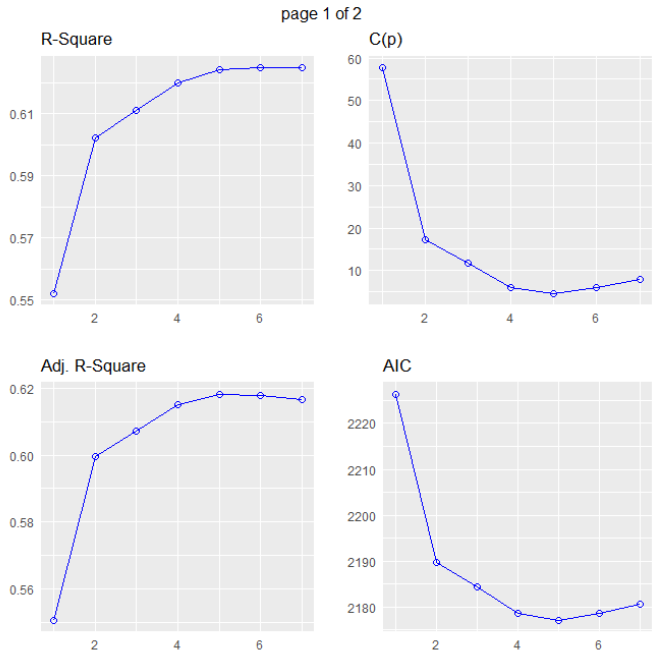
```
mod1<- lm(pm25~ ws + wd + nox + no2 + o3 + so2 + co, data= quality_air1
    )
summary(mod1)
library(olsrr)
ols_step_all_possible(mod1)
ols_step_best_subset(mod1)
```

```
> ols_step_all_possible(mod1) #127 models
   Index N              Predictors     R-Square Adj. R-Square Mallow's Cp
3      1 1                     nox 0.5519361750   0.550553262   57.769303
6      2 1                     so2 0.4920738740   0.490506201  108.507308

...
```

Variable selection: Lasso regression and others

Finally using library(olsrr) a plot to select the different models.

```
k <- ols_step_best_subset(mod1)
plot(k)
```



Altres Exemples de regressió múltivariant

Trobareu diferents exemples de regressió multivariant per fer a:

- Exploration of the trees data set
- Exemple medi ambient
- Exemple crims i variables relacionades

Kriging regression

Toni Monleón-Getino (basat en diversos materials lliures)

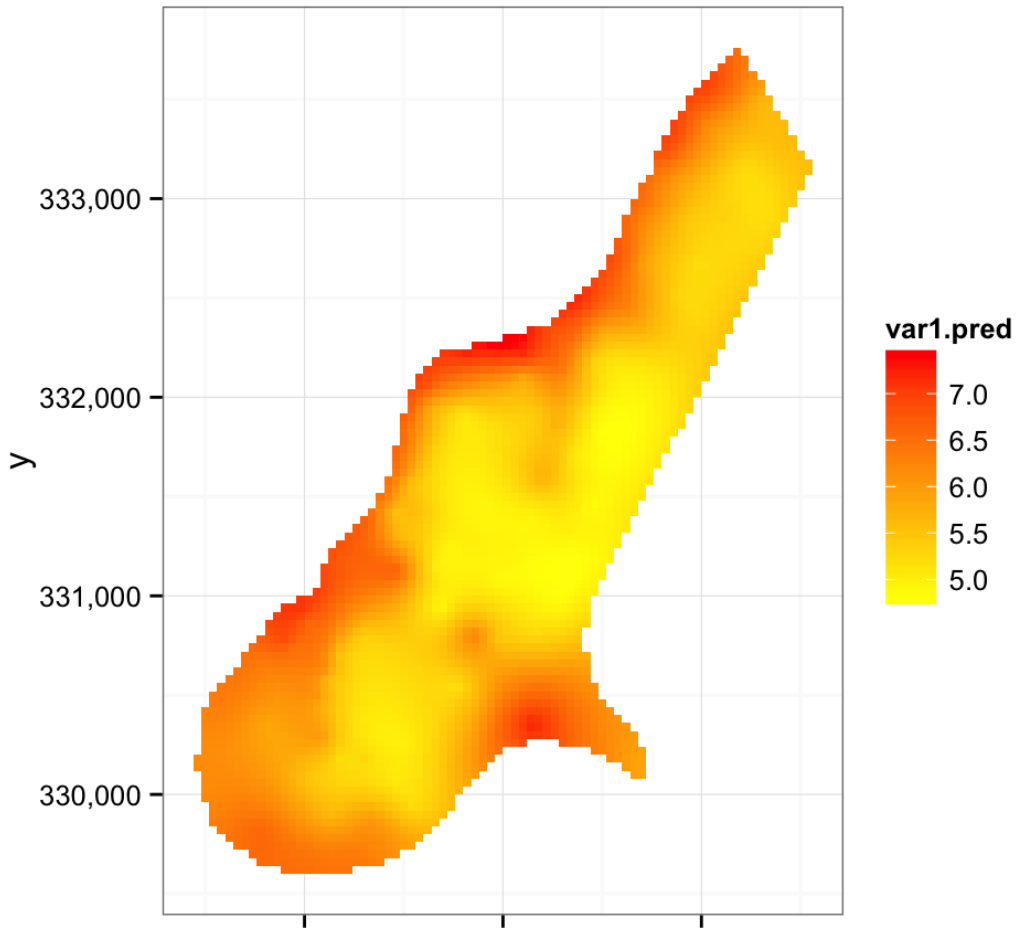
Novembre 2020

Regression-kriging is a spatial interpolation technique that combines a regression of the dependent variable (target variable) on predictors (i.e. the environmental covariates) with kriging of the prediction residuals.

In other words, Regression-Kriging is a hybrid method that combines either a simple or a multiple-linear regression model with ordinary kriging of the prediction residuals.

A little bit of information:
Geostatistics:
http://snobear.colorado.edu/Markw/BioMath/Geostats/Spatial_geostat.p
Theory
https://www.nersc.no/sites/www.nersc.no/files/Basics2kriging.pdf

Heavy metal example (spatial distribution) in Holland



Kriging regression

See this example in: Kriging regressio example

The meuse dataset contains concentration measurements for a number of chemical elements taken from the Meuse river in the Netherlands. More information can be found by checking the help page via ?meuse.

Of particular interest is that each value/measurement is associated with geographic coordinates, namely the x- and y- columns. A priori, given just the dataframe and no additional information, it might not be clear that those two columns indicate locations (I, at least, had never heard of RDH coordinates before).

And that's what the motivation for SPDF's was: to provide a structure which allows for coordinates to clearly be associated with corresponding

Unsupervised Learning

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

Unsupervised learning

Unsupervised learning:

1) no outcome variable, just a set ofpredictors (features) measured on a set of samples,

2) objective is more fuzzy,

3) difficulty to know how well you are doing, and

4) can be useful as a pre-processing step for supervised learning

Principal component analysis

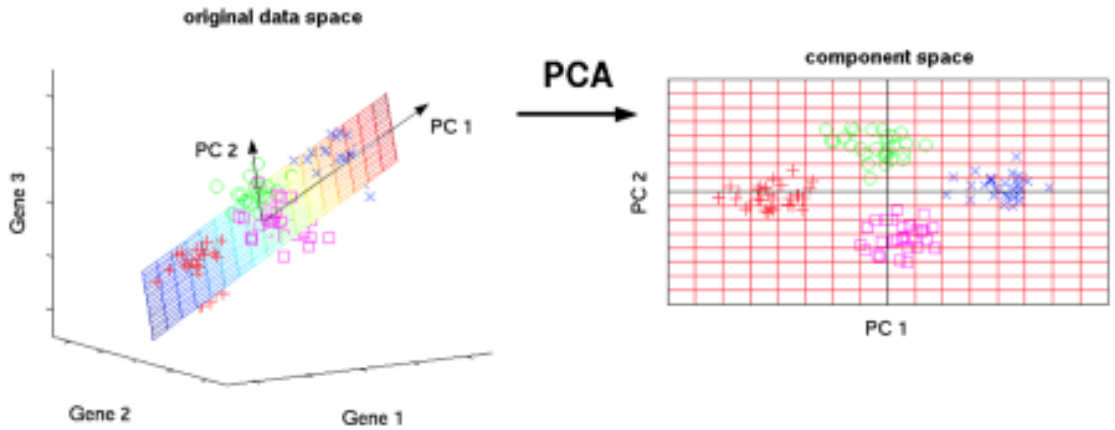
Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

The problem of High-Dimensional Data

Reduction of dimension. Summarization of data with many (p) variables by a smaller set of (k) derived (synthetic, composite) variables

The image below shows the transformation of a high dimensional data (3 dimension) to low dimensional data (2 dimension) using PCA. Not to forget, each resultant dimension is a linear combination of *p* features

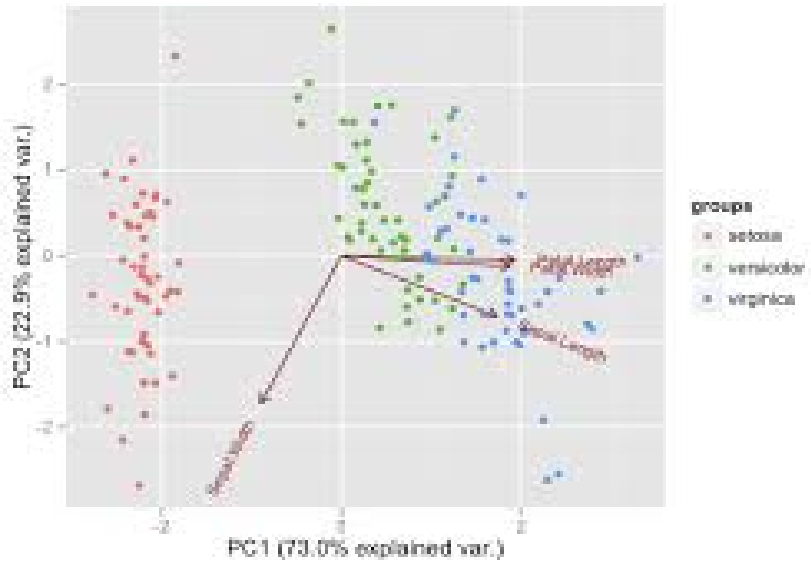


Source: nlpca

See the presentation in: PCA introduction Princeton

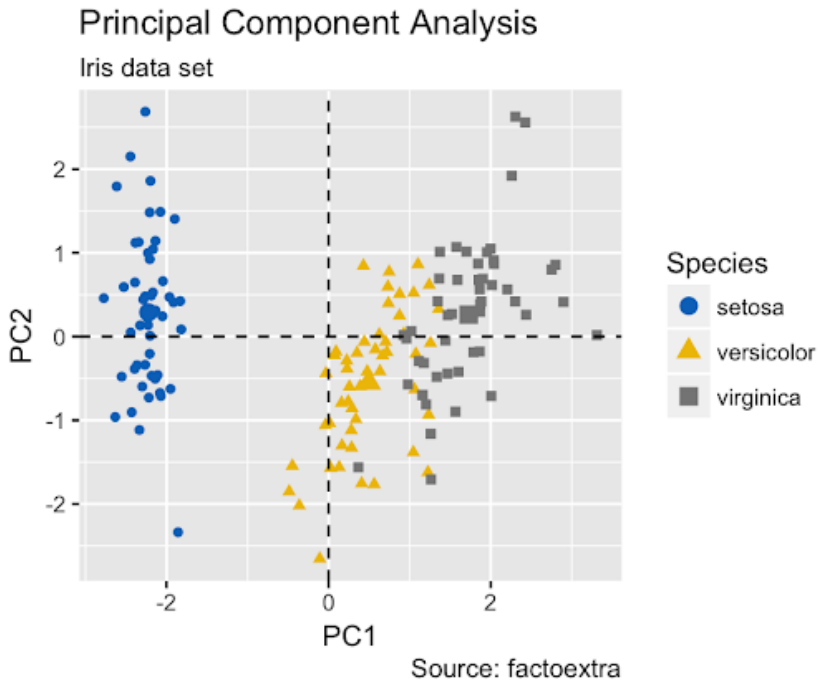
PCA: A small introduction

See other presentation in: PCA introduction



PCA example in R: Iris data set

See a biological example (Iris measures) in: example 1 of PCA



Other very complete example at:
example 2 of PCA

Principal component analysis

Principal component analysis can be used to analyze the structure of a data set or allow the representation of the data in a lower dimensional dataset (as well as many other applications).
Let $\{\vec{x}_i\}$ be a set of N column vectors of dimension D. Define the scatter matrix S_x of the data set as

$$\mathrm{S}_x = \sum_{i=1}^{N} (\vec{x}_i - \vec{\mu}_x)(\vec{x}_i - \vec{\mu}_x)^T$$

where $\vec{\mu}_x$ is the mean of the dataset

$$\vec{\mu}_x = \frac{1}{N} \sum_{i=1}^{N} \vec{x}_i$$

PCA

The d largest principle components are the eigenvectors $\vec{w}_i$ corresponding to the d largest eigenvalues. d can be chosen arbitrarily with $d < D$. The eigenvectors of $\mathbf{S}$ can usually be found by using singular value decomposition.

The dominant eigenvectors describe the main directions of variation of the data. For example, if a dataset had 2 large eigenvalues, then the data variation is described largely by linear combinations of the 2 corresponding eigenvectors (ie. the data is largely coplanar).

PCA

The d eigenvectors can also be used to project the data into a d dimensional space. Define

$$\mathrm{W} = [\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_d]$$

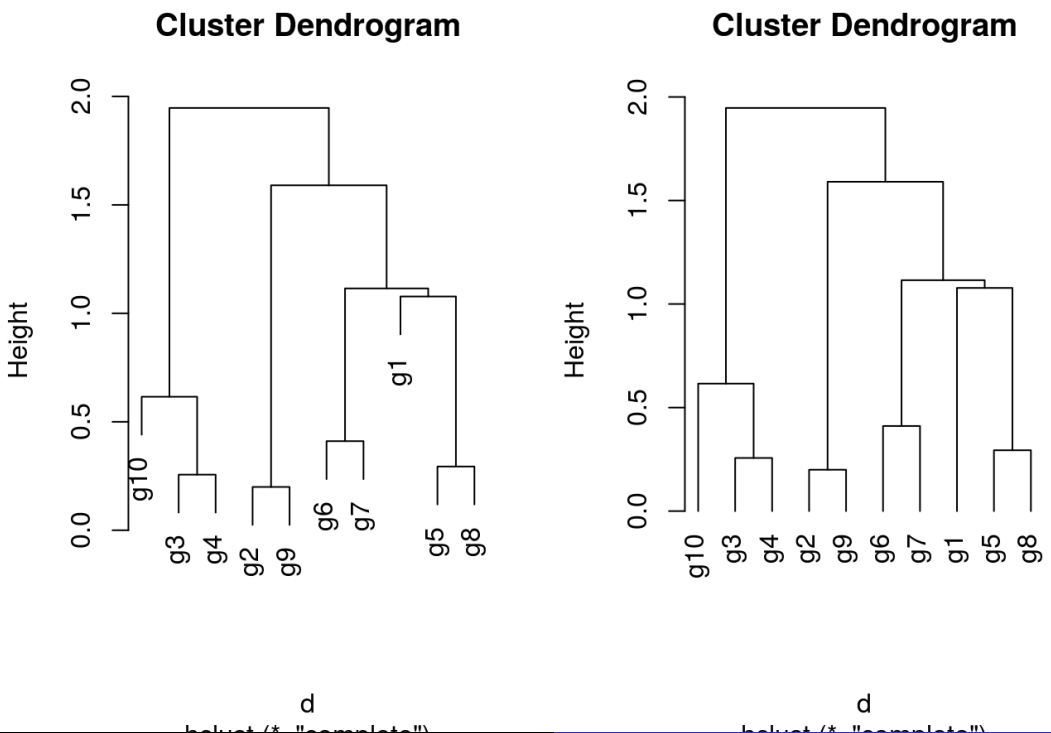
The projection of vector $\vec{x}$ is $\vec{y} = \mathrm{W}^T \vec{x}$. The corresponding scatter matrix S_y of the vectors $\{\vec{y}_i\}$ is:

$$\mathrm{S}_y = \mathrm{W}^T \mathrm{S}_x \mathrm{W}$$

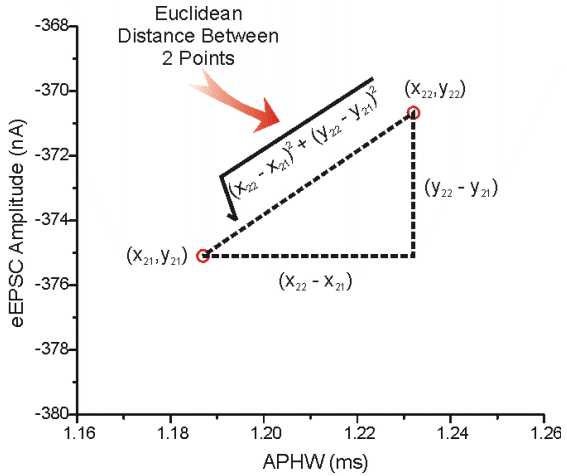
The matrix W maximizes the determinant of S_y for a given d.

Classification methods

Aqui podeu trobar un curs sobre mètodes de classificació en R:
Cluster Analysis in R



Concepts in Classification methods



- Algunes distancies estadístiques (distància euclidiana, Manhattan, etc): Distances
- Conceptes sobre distàncies euclidiana: Euclidiana
- Conceptes sobre distàncies /similaritats no euclidianes: no Euclidiana

MULTIDIMENSIONAL SCALING

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

MDS

If you are interested in how certain objects relate to each other . . . and if you would like to present these relationships in the form of a map then MDS is the technique you need"

"Multidimensional scaling (MDS) is a means of visualizing the level of similarity of individual cases of a dataset. MDS is used to translate ïnformation about the pairwise 'distances' among a set of n objects or individualsïnto a configuration of n points mapped into an abstract Cartesian space"

See this link about MDS in: MDS

MDS: examples

Exercisi a library(BDSbiost3)

funció: LinesMDS(matriu, use.conditions = F, distance = "BT", OTU = T, label.yes = F, vector.labels, nrows = 50)

Library in:

https://github.com/amonleong/BDSbiost3

hierarchical cluster analysis

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

hierarchical cluster analysis

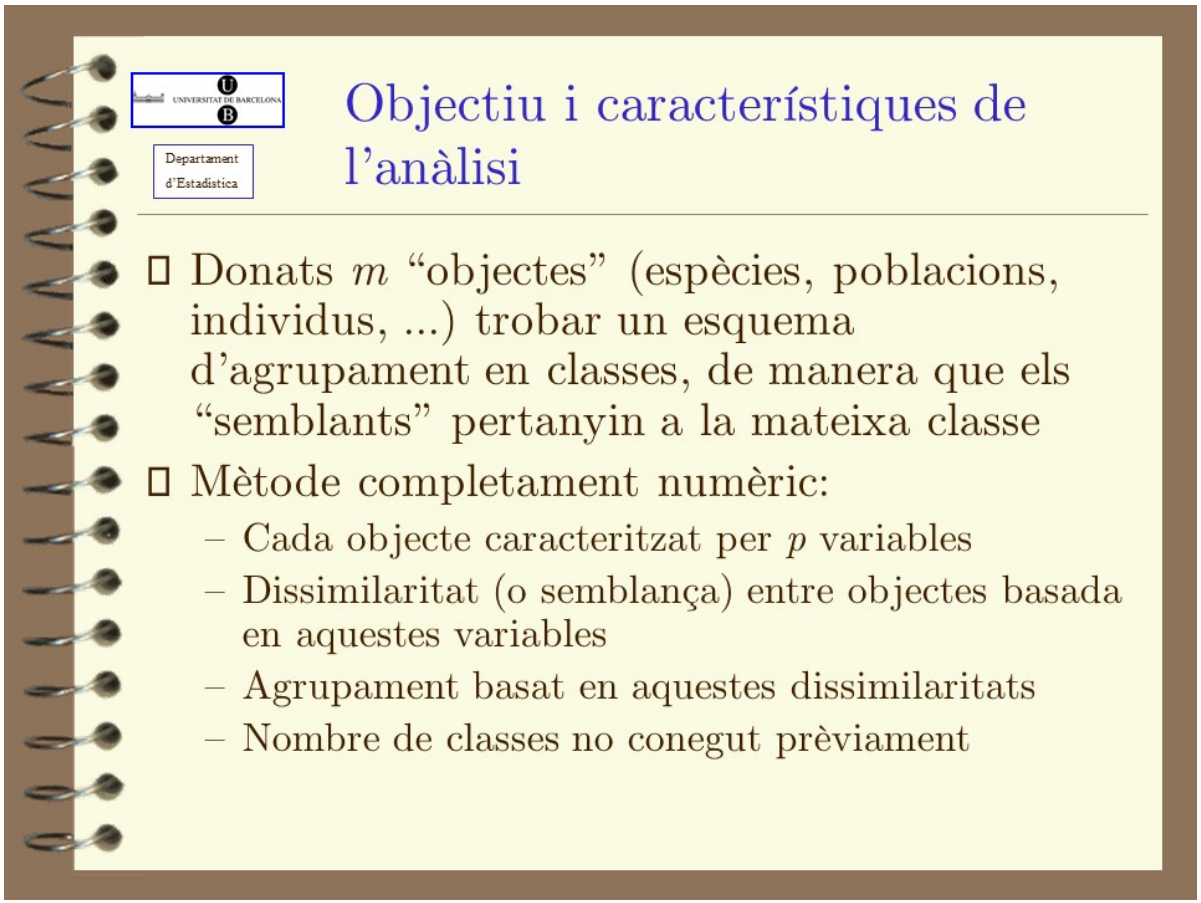
See introduction in:
Hierarquical cluster analysis

Example in this link ierarquical cluster analysis example and little introduction seeds dataset

More presentations: Hierarquical cluster analysis example

Hierarquical cluster analysis example

Antigua presentacion HC



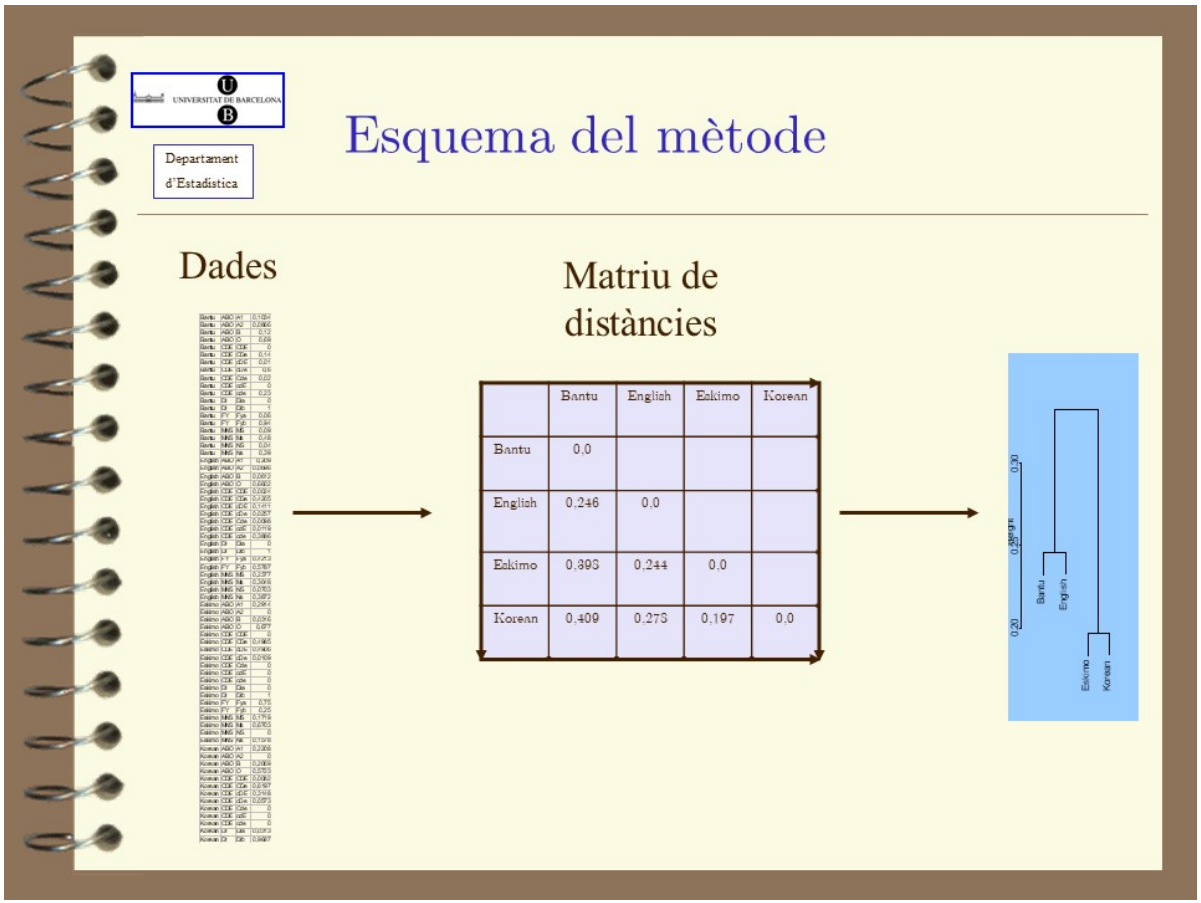
UNIVERSITAT DE BARCELONA

Departament d'Estadística

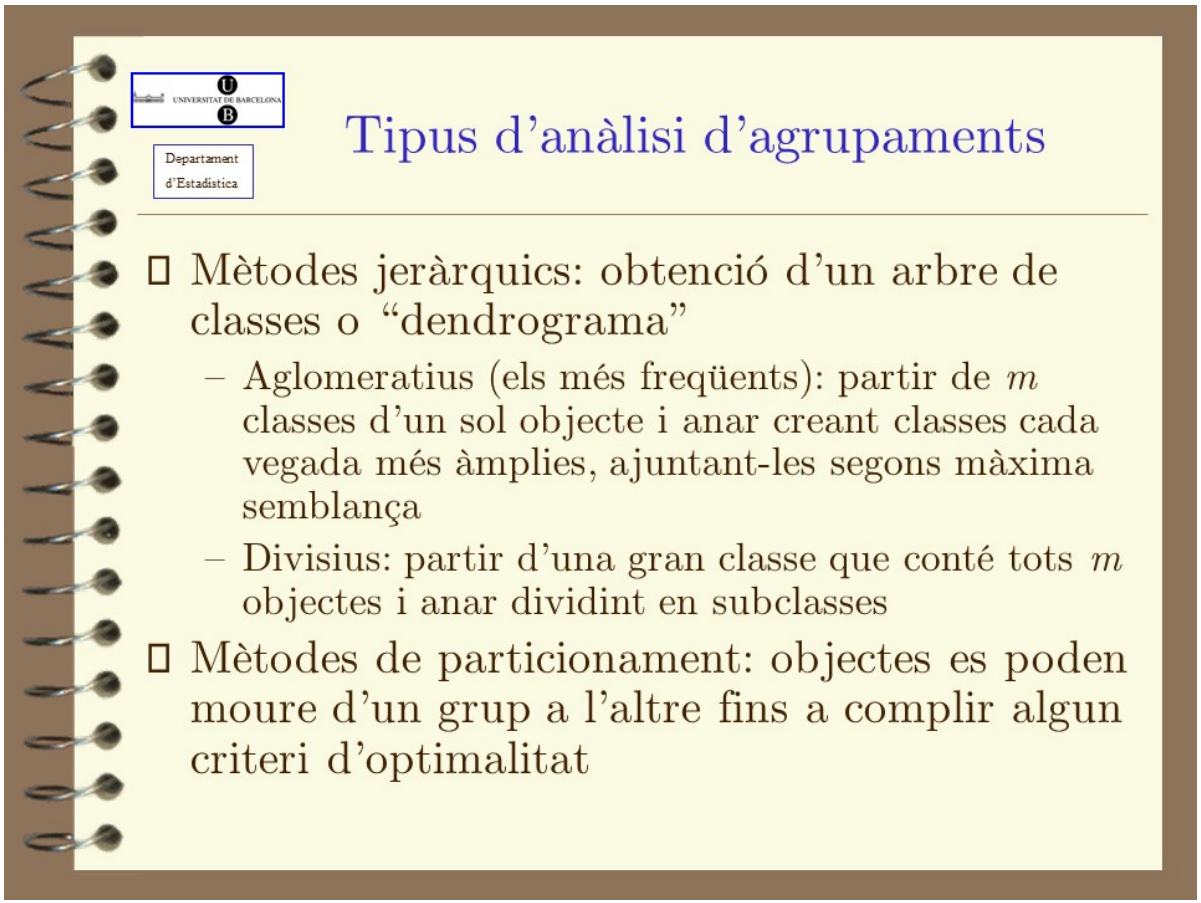
Objectiu i característiques de l'anàlisi

- Donats m "objectes" (espècies, poblacions, individus, ...) trobar un esquema d'agrupament en classes, de manera que els "semblants" pertanyin a la mateixa classe
- Mètode completament numèric:
  - Cada objecte caracteritzat per p variables
  - Dissimilaritat (o semblança) entre objectes basada en aquestes variables
  - Agrupament basat en aquestes dissimilaritats
  - Nombre de classes no conegut prèviament

Antigua presentacion HC



Antigua presentacion HC



Antigua presentacion HC

UNIVERSITAT DE BARCELONA

Departament d'Estadística

Dades de grups sanguinis per diverses (m=4) poblacions

- Freqüències relatives dels al·lels possibles per a $s = 5$ sistemes de grups sanguinis: AB0, CDE, DI, FY i MNS
- En sengles mostres de $m = 4$ poblacions: Bantús, Anglesos, Esquimals i Coreans.

Bantu	ABO	A1	0,1034
Bantu	ABO	A2	0,0866
Bantu	ABO	B	0,12
Bantu	ABO	O	0,69
Bantu	CDE	CDE	0
Bantu	CDE	CDe	0,14
Bantu	CDE	cDE	0,01
Bantu	CDE	cDe	0,6
Bantu	CDE	Cde	0,02
Bantu	CDE	cdE	0
Bantu	CDE	cde	0,23
Bantu	DI	Dia	0
Bantu	DI	Dib	1
Bantu	FY	Fya	0,06
Bantu	FY	Fyb	0,94
Bantu	MNS	MS	0,09
Bantu	MNS	Ms	0,48
Bantu	MNS	NS	0,04
Bantu	MNS	Ns	0,39
English	ABO	A1	0,209
English	ABO	A2	0,0696
English	ABO	B	0,0612
English	ABO	O	0,6602
English	CDE	CDE	0,0024

etc...

Antigua presentacion HC

UNIVERSITAT DE BARCELONA

Departament d'Estadística

Adequada per aquestes dades: distància de Prevosti

$s = 5$ sistemes de grups sanguinis, amb

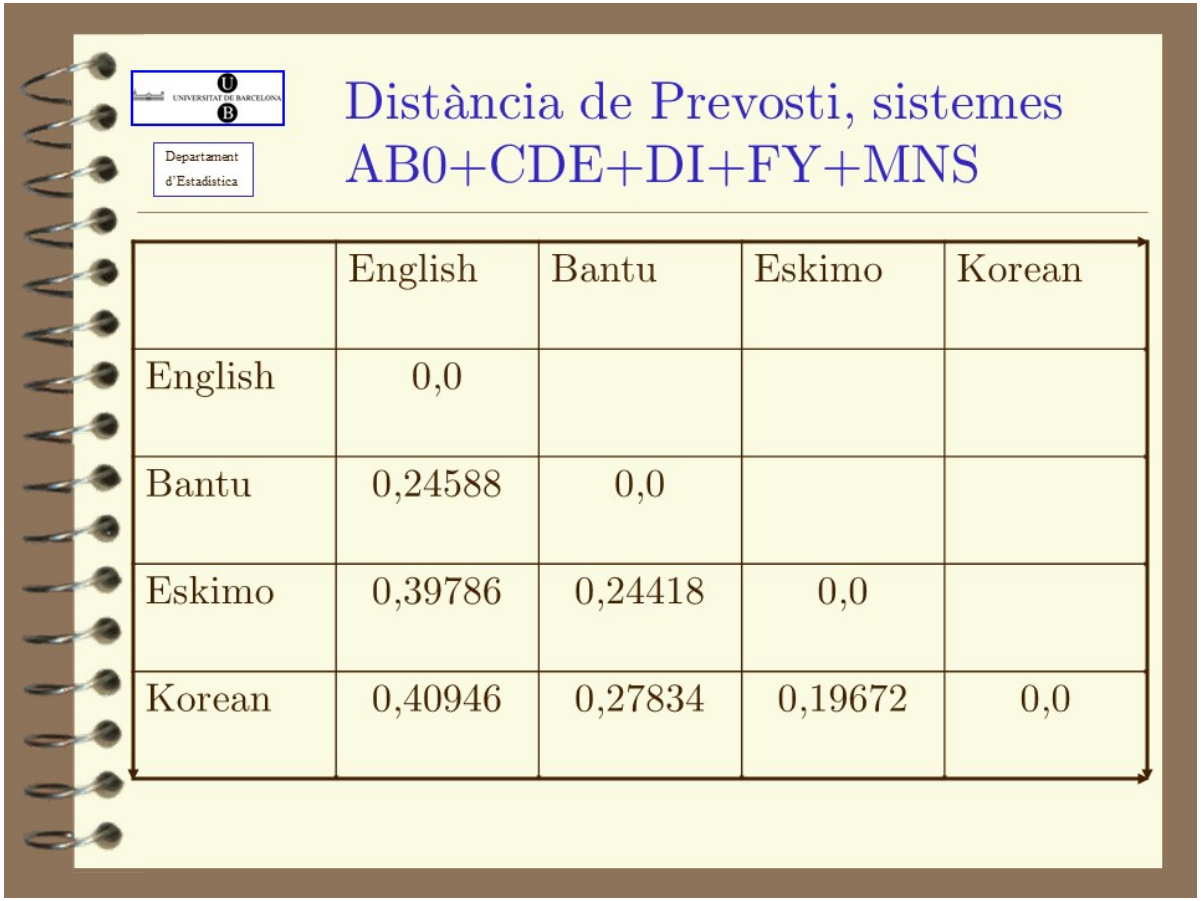
$a_1 = 4, a_2 = 7, a_3 = 2, a_4 = 2, a_5 = 4$

al·lels possibles, respectivament.

p_{ij} proporció de l'al·lel j del sistema i a la població P (igualment per Q)

$$D(P,Q) = \frac{1}{2s} \sum_{i=1}^{s} \sum_{j=1}^{a_i} |p_{ij} - q_{ij}|$$

Antigua presentacion HC



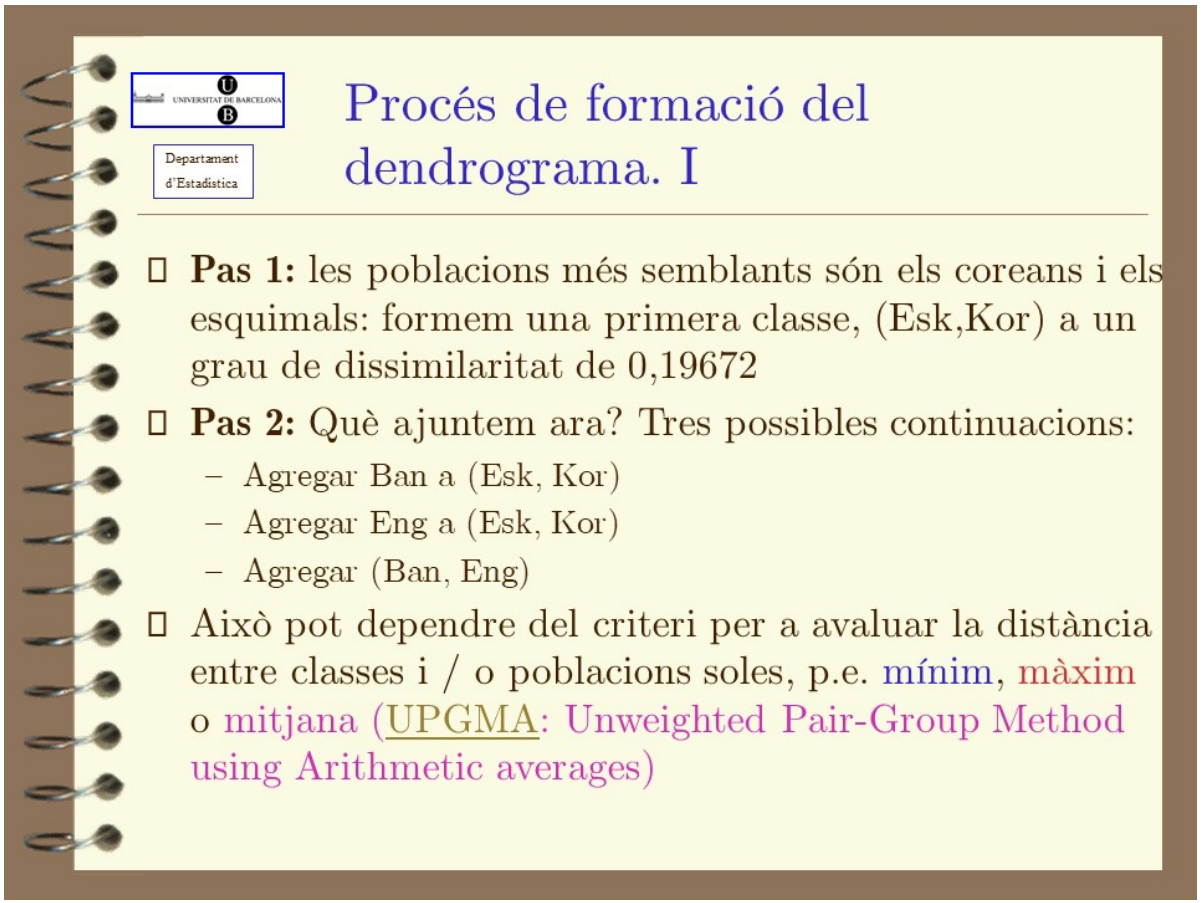
UNIVERSITAT DE BARCELONA

Departament d'Estadística

Distància de Prevosti, sistemes AB0+CDE+DI+FY+MNS

	English	Bantu	Eskimo	Korean
English	0,0			
Bantu	0,24588	0,0		
Eskimo	0,39786	0,24418	0,0	
Korean	0,40946	0,27834	0,19672	0,0

Antigua presentacion HC



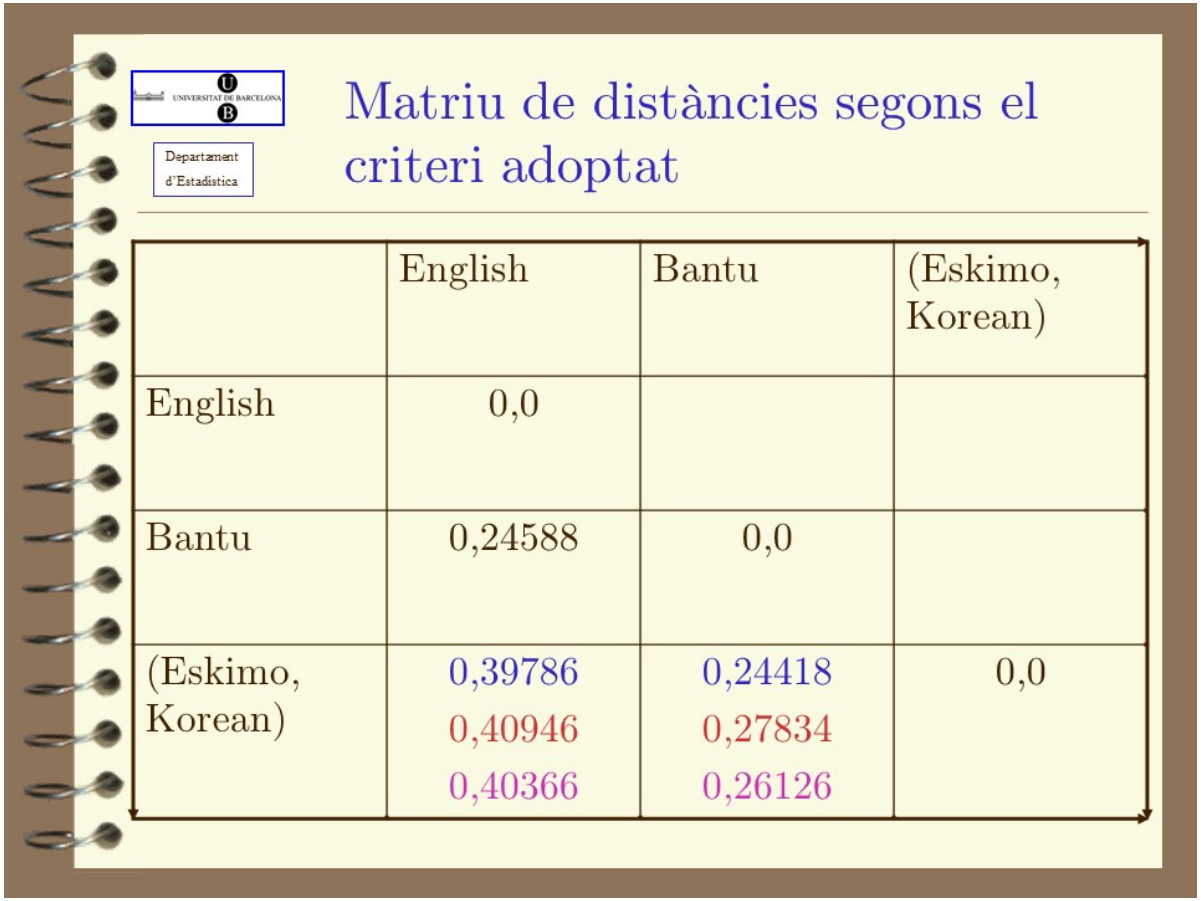
UNIVERSITAT DE BARCELONA

Departament d'Estadística

Procés de formació del dendrograma. I

- **Pas 1:** les poblacions més semblants són els coreans i els esquimals: formem una primera classe, (Esk,Kor) a un grau de dissimilaritat de 0,19672
- **Pas 2:** Què ajuntem ara? Tres possibles continuacions:
  - Agregar Ban a (Esk, Kor)
  - Agregar Eng a (Esk, Kor)
  - Agregar (Ban, Eng)
- Això pot dependre del criteri per a avaluar la distància entre classes i / o poblacions soles, p.e. mínim, màxim o mitjana (UPGMA: Unweighted Pair-Group Method using Arithmetic averages)

Antigua presentacion HC



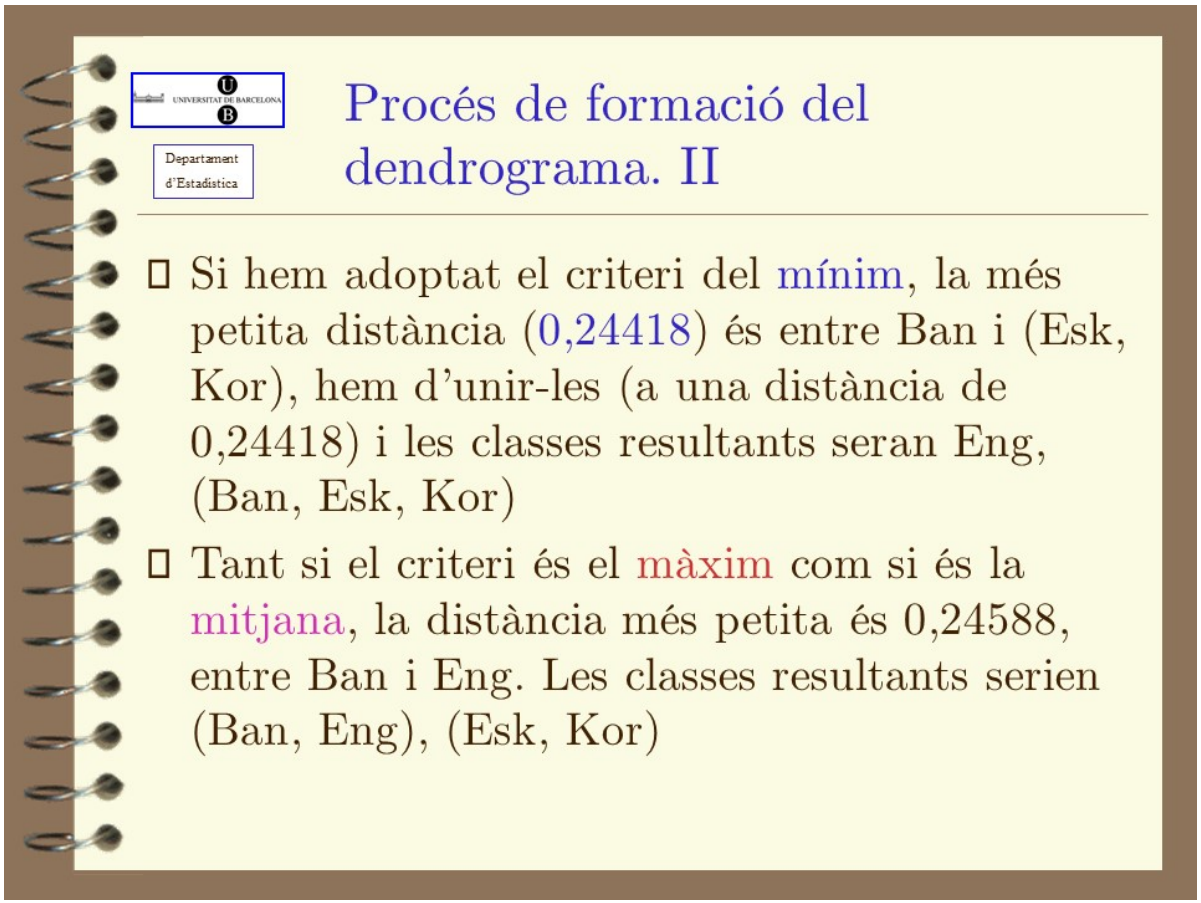
UNIVERSITAT DE BARCELONA

Departament d'Estadística

Matriu de distàncies segons el criteri adoptat

	English	Bantu	(Eskimo, Korean)
English	0,0		
Bantu	0,24588	0,0	
(Eskimo, Korean)	0,39786 0,40946 0,40366	0,24418 0,27834 0,26126	0,0

Antigua presentacion HC



UNIVERSITAT DE BARCELONA

Departament d'Estadística

Procés de formació del dendrograma. II

- Si hem adoptat el criteri del mínim, la més petita distància (0,24418) és entre Ban i (Esk, Kor), hem d'unir-les (a una distància de 0,24418) i les classes resultants seran Eng, (Ban, Esk, Kor)
- Tant si el criteri és el màxim com si és la mitjana, la distància més petita és 0,24588, entre Ban i Eng. Les classes resultants serien (Ban, Eng), (Esk, Kor)

Partitional cluster: KMEANS PAM CLARA

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

PAM KMEANS CLARA

See introduction in:

PARTITIONAL CLUSTER INTRODUCTION 1

PARTITIONAL CLUSTER INTRODUCTION 2

PARTITIONAL CLUSTER INTRODUCTION 3

Supervised Learning

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

Supervised learning

Supervised learning problem:

we have training data $(x_i, y_i)_{i=1}^N$ and we would like to:

- 1) accurately predict unseen test cases
- 2) understand which inputs affect the outcome and how
- 3) assess the quality of our predictions and inferences

Classification and discrimination

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

Classification Discrimination problems

- The objective in a classification problem is to be able to classify an object
- into a finite number of distinct groups based on observed quantities.

- Classification is a problem that can be supervised or unsupervised learning method in function of the objective. In this case we study the case of the supervised learning.

- In simple terms, discriminant function analysis is classification when we try to separate known groups.

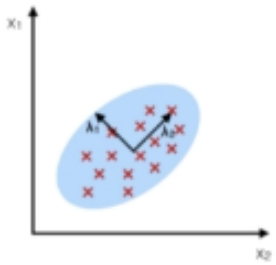
- Discrimination attempts to separate distinct sets of objects, and classification attempts to allocatenew objects to predefined group.

- There are many methods to do discrimination (LDA, SVM, etc) and supervised classification (KMEANS, PAM).

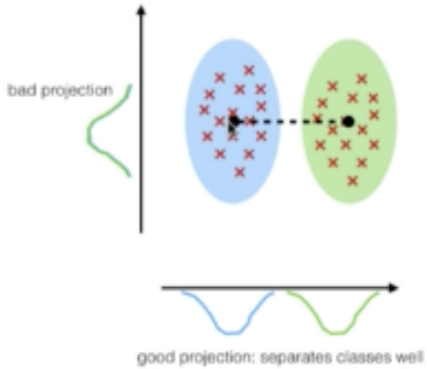
Classification Discrimination problems

Linear Discriminant Analysis - LDA

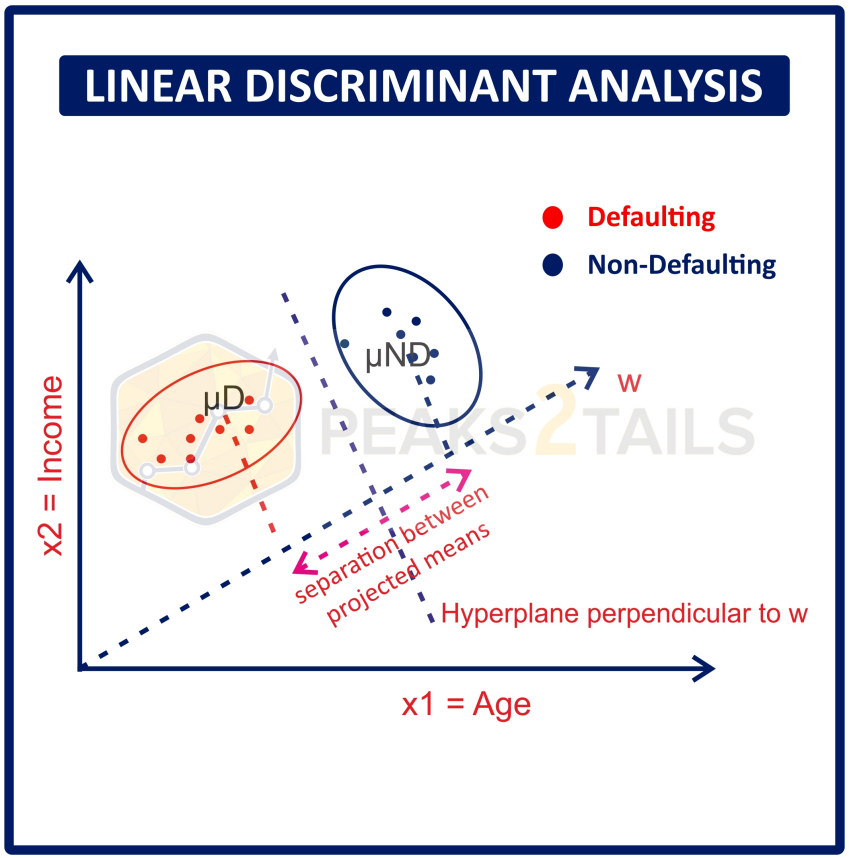
PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



Classification Discrimination problems



Linear Discriminant Analysis (LDA)

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

Iris data. A classical example of LDA

A classical dataset collected by the botanist Edgar Anderson, 1935, *The irises of the Gaspe Peninsula* and studied by statistician R. A. Fisher, 1936 *The use of multiple measurements in taxonomic problems*. Available as the `iris` dataset in the `MASS` library in `R`.

iris

Sepal		Petal		
Length	Width	Length	Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

LDA: linear discriminant analysis

See introduction in:

- LDA-Introduction: theory I
- LDA-Introduction: theory II

- iris example in R - LDA
- iris example in R(1) - LDA

- Nathaniel E. Helwig: Classification and discrimination with iris data
- Nathaniel E. Helwig: iris example in R(2) - ADVANCED LDA

More theory and R:

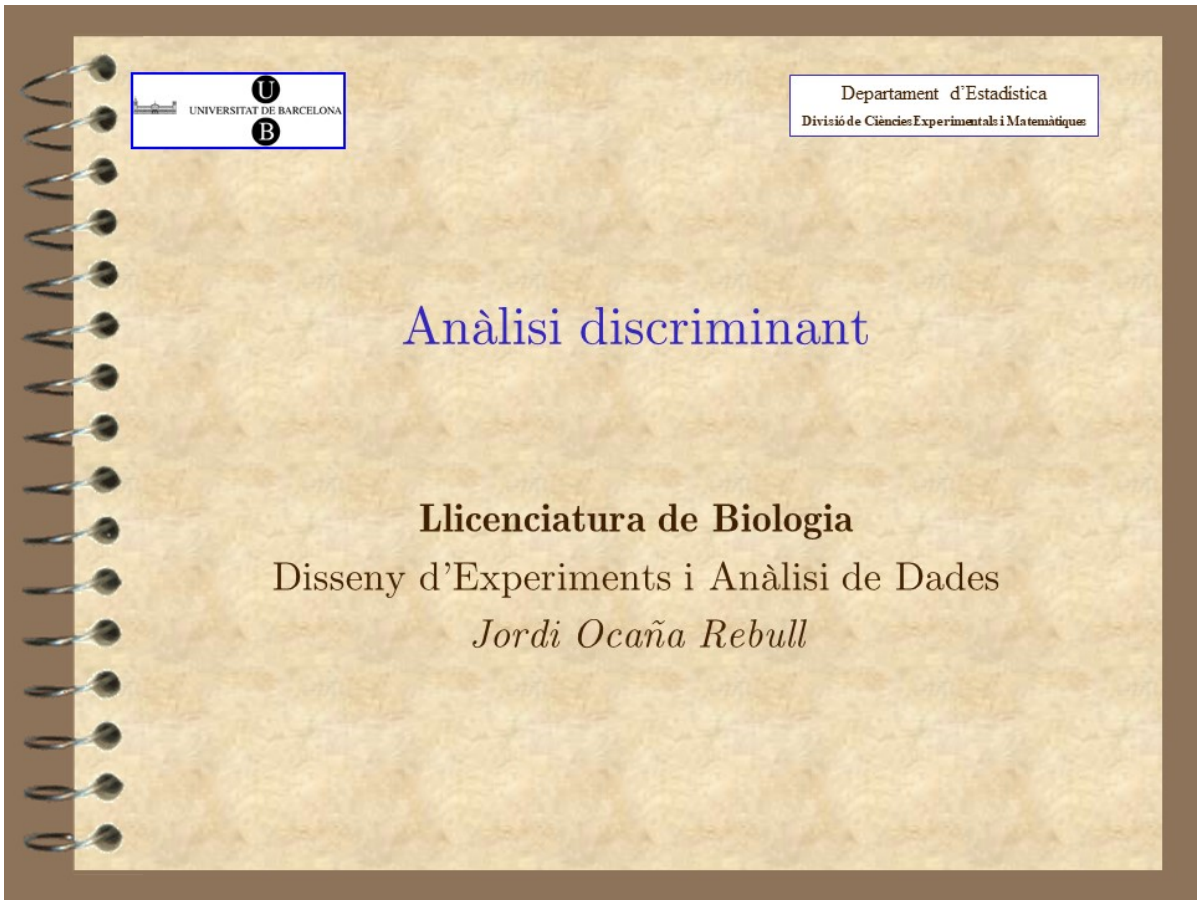
- LDA-Introduction by Xuelian Wei
- Compare LDA and PCA step by step

Exemple simulat

Script R per obtenir matriu de confussió en LDA

```
data(iris)

da <- lda(Species ~ ., data = iris)

pred <- predict(da, dimen = 1)

#confussion matrix (accuracy of the method)
table(test$lda,test$Species)

library(caret) # little bit better

confusionMatrix(iris$Species, pred$class)
```

Antigua presentacion LDA



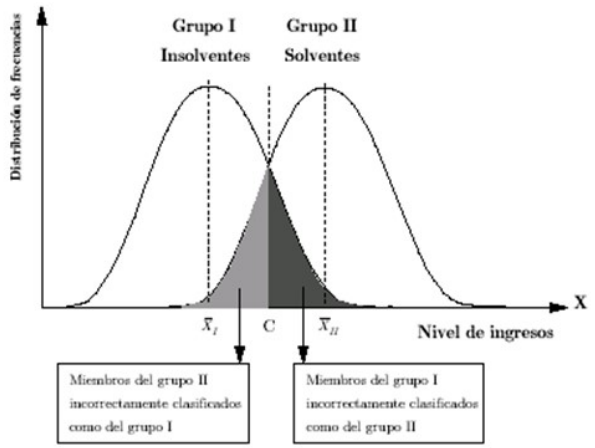
Antigua presentacion LDA

UNIVERSITAT DE BARCELONA

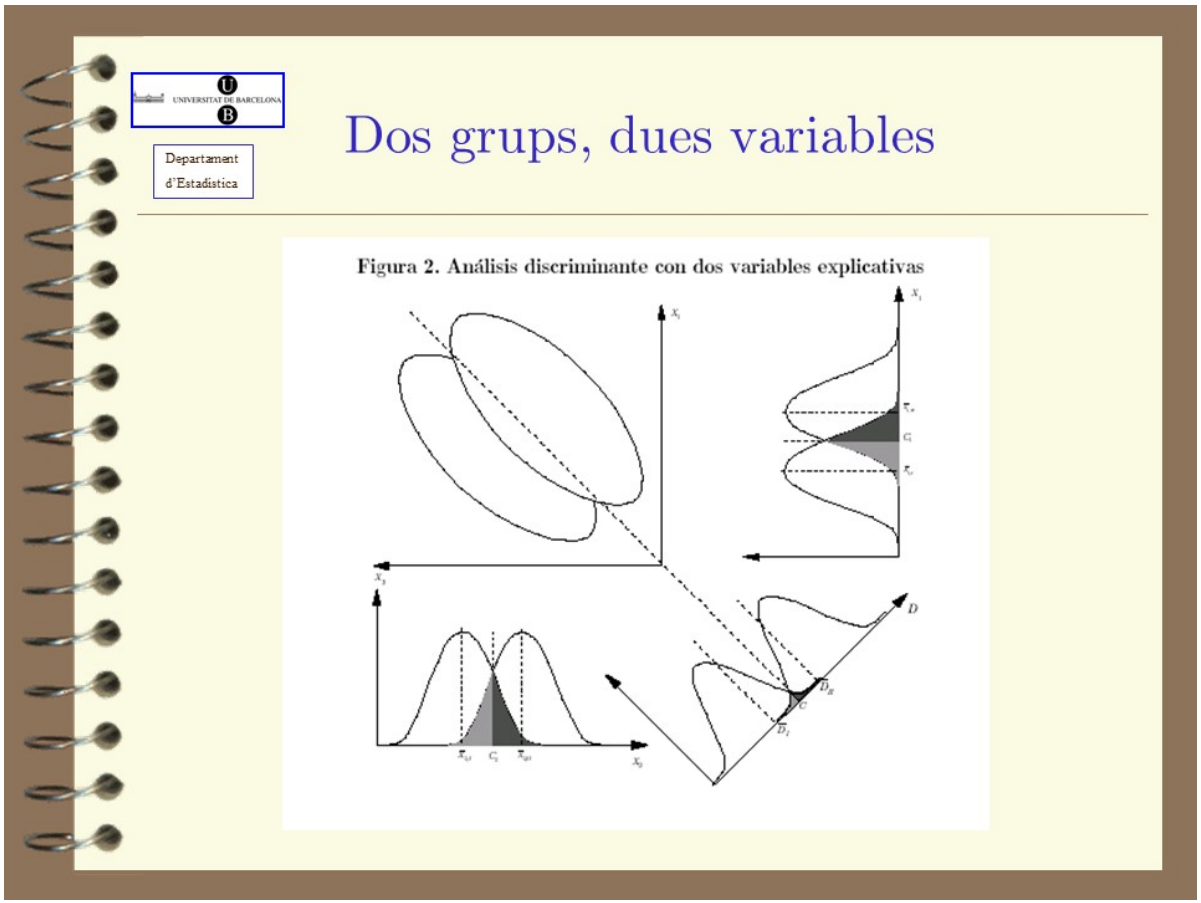
Departament d'Estadística

Discriminació entre dues poblacions amb una variable

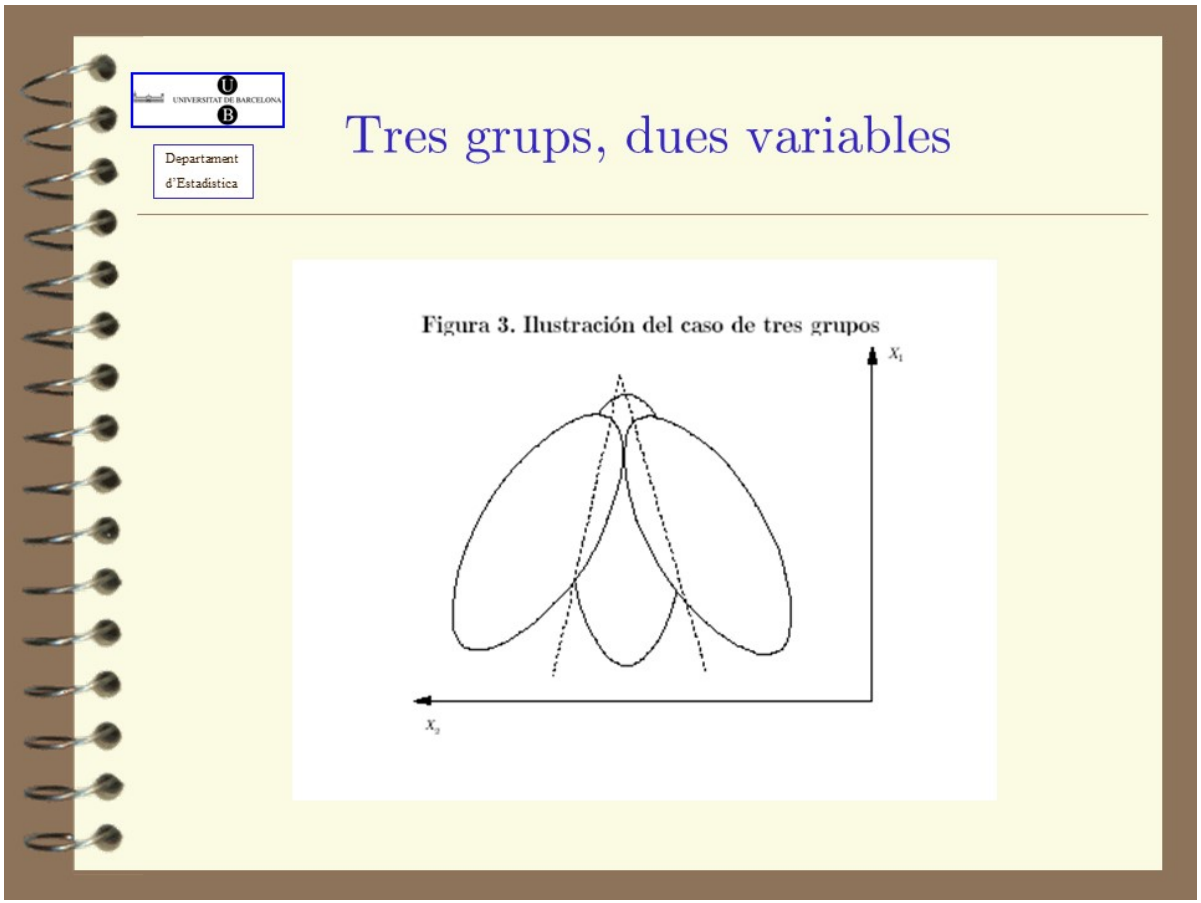
Figura 1. Funciones de distribución hipotéticas de dos grupos



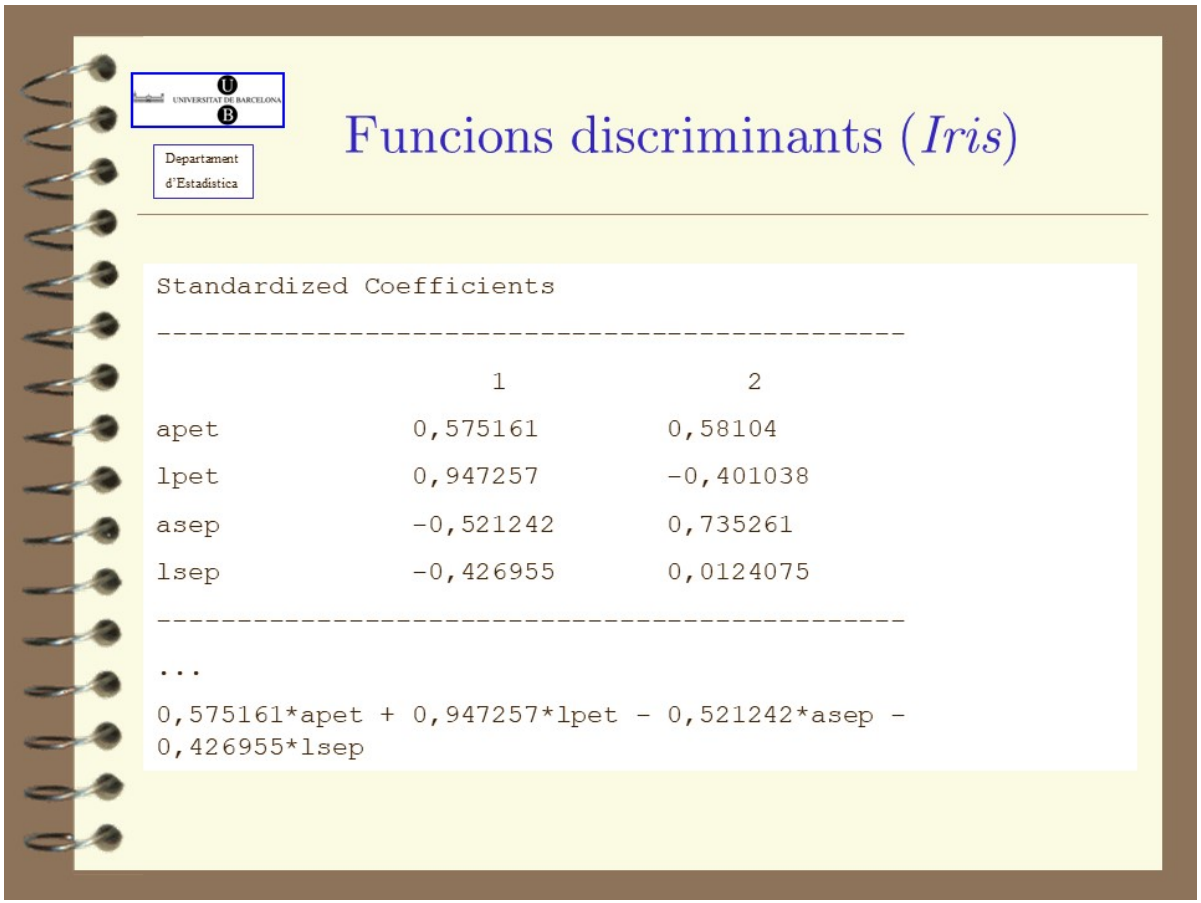
Antigua presentacion LDA



Antigua presentacion LDA



Antigua presentacion LDA



UNIVERSITAT DE BARCELONA

Departament d'Estadística

Funcions discriminants (*Iris*)

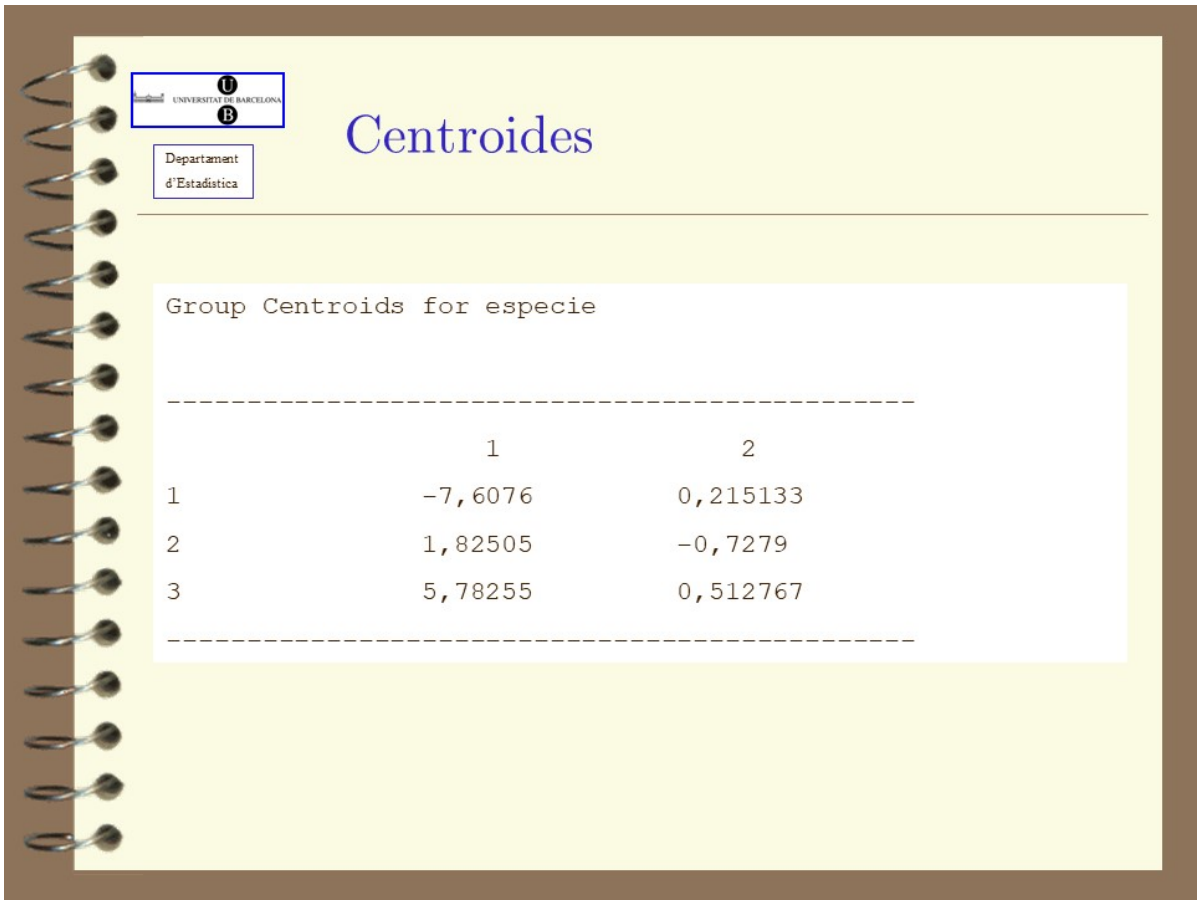
Standardized Coefficients

	1	2
apet	0,575161	0,58104
lpet	0,947257	-0,401038
asep	-0,521242	0,735261
lsep	-0,426955	0,0124075

...

0,575161*apet + 0,947257*lpet - 0,521242*asep - 0,426955*lsep

Antigua presentacion LDA



UNIVERSITAT DE BARCELONA

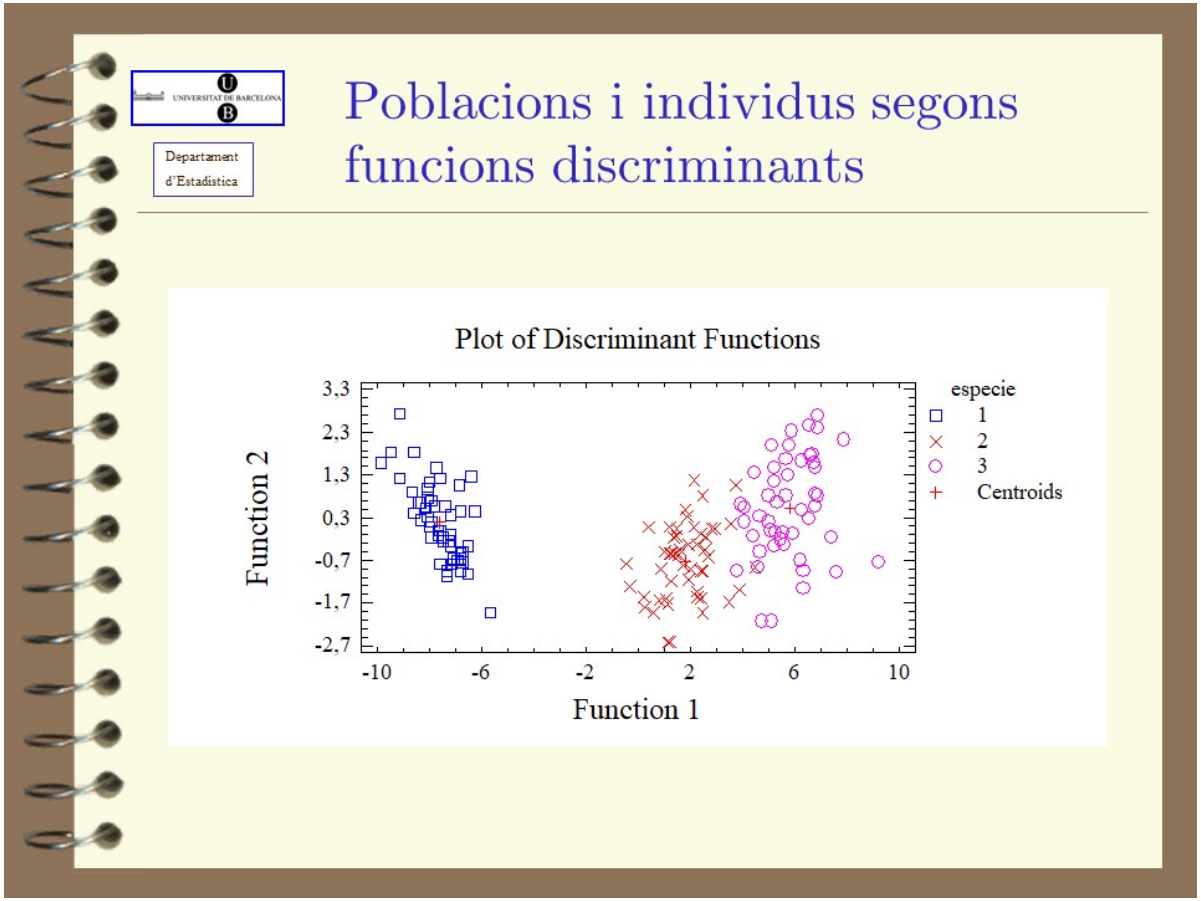
Departament d'Estadística

Centroides

Group Centroids for especie

	1	2
1	-7,6076	0,215133
2	1,82505	-0,7279
3	5,78255	0,512767

Antigua presentacion LDA



Classification and discrimination theory

We have observations $(x_1, y_1), \ldots, (x_N, y_N)$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$. We assume that the data arose as independent and identically distributed samples of a pair (X, Y) of random variables.

Assume $X = x_0 \in \mathbb{R}^p$ what is Y? Let

$$N_k(x_0) = \{i \mid x_i \text{ is one of the } k\text{'th nearest observations}\}.$$

Define

$$\hat{f}(x_0) = \frac{1}{k} \sum_{i \in N_k(x_0)} y_i \in [0, 1]$$

and classify using majority rules

$$\hat{y} = \begin{cases} 1 & \text{if } \hat{f}(x_0) \geq 1/2 \\ 0 & \text{if } \hat{f}(x_0) < 1/2 \end{cases}$$

In generality we study problems where $x_i \in E$ and $y_i \in F$ and where we want to understand the relation between the two variables. When $F = \mathbb{R}$ we mostly talk about regression and when F is discrete we talk about classification.

Sometimes the assumption of independence can be relaxed without harming the methods used too seriously, and in other cases – in designed experiments – we can hardly think of the x_i's as random, in which case we will regard only the y_i's as (conditionally) independent given the x_i's.

Linear Classifiers

A classifier is called linear if there is an affine function

$$x \mapsto x^T \beta + \beta_0$$

with the classifier at x_0

$$f(x) = \begin{cases} 1 & \text{if } x^T \beta + \beta_0 \geq 0 \\ 0 & \text{if } x^T \beta + \beta_0 < 0 \end{cases}$$

There are several examples of important linear classifiers. We encounter

- Linear discriminant analysis (LDA).
- Logistic regression.
- Support vector machines.

Tree based methods is a fourth method that relies on locally linear classifiers.

LDA

The Fisher Linear Discriminant (FLD) gives a projection matrix W that reshapes the scatter of a data set to maximize class separability, defined as the ratio of the between-class scatter matrix to the within-class scatter matrix.

This projection defines features that are optimally discriminating.
Let $\{\vec{x}_i\}$ be a set of N column vectors of dimension D. The mean of the dataset is

$$\vec{\mu}_x = \frac{1}{N} \sum_{i=1}^{N} \vec{x}_i$$

LDA

There are K classes $\{C_1, C_2, \ldots, C_K\}$. The mean of class k containing N_k members is:

$$\vec{\mu}_{xk} = \frac{1}{N_k} \sum_{\vec{x}_i \in C_k} \vec{x}_i$$

The between class scatter matrix is

$$S_B = \sum_{k=1}^{K} N_k (\vec{\mu}_{xk} - \vec{\mu}_x)(\vec{\mu}_{xk} - \vec{\mu}_x)^T$$

The within class scatter matrix is

$$S_W = \sum_{k=1}^{K} \sum_{\vec{x}_i \in C_k} (\vec{x}_i - \vec{\mu}_{xk})(\vec{x}_i - \vec{\mu}_{xk})^T$$

LDA

The transformation matrix that repositions the data to be most separable is the matrix W that maximizes

$$\frac{det(\mathrm{W}^T \mathrm{S}_B \mathrm{W})}{det(\mathrm{W}^T \mathrm{S}_W \mathrm{W})}$$

Let $\{\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_D\}$ be the generalized eigenvectors of S_B and S_W. Then $\mathbf{W} = [\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_D]$. This gives a projection space of dimension D. A projection space of dimension $d < D$ can be defined by using the generalized eigenvectors with the largest d eigenvalues to give $\mathbf{W}_d = [\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_d]$.

The projection of vector $\vec{x}$ into a subspace of dimension d is $\vec{y} = \mathrm{W}_d^T \vec{x}$.
The generalized eigenvectors are eigenvectors of

$$\mathrm{S}_B \mathrm{S}_W^{-1}$$

Mahalanobis distance

The distance between two N dimensional points scaled by the statistical variation in each component of the point. For example, if $\vec{x}$ and $\vec{y}$ are two points from the same distribution which has covariance matrix $\mathbf{C}$, then the Mahalanobis distance is given by

$$((\vec{x} - \vec{y})' \mathbf{C}^{-1} (\vec{x} - \vec{y}))^{\frac{1}{2}}$$

The Mahalanobis distance is the same as the Euclidean distance if the covariance matrix is the identity matrix.

A common usage in computer vision systems is for comparing feature vectors whose elements are quantities having different ranges and amounts of variation, such as a 2-vector recording the properties of area and perimeter.

SVM: Support vector machine

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

SVM: Support vector machine

See introduction in: svm

Boosting

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

Boosting

See introduction in: boosting

KERNEL METHODS

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

KERNEL METHODS

See introduction in: kernel methods

ANN

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

ANN

See introduction in: ann

See another good introduction in:
ann

Bibliografia - links

http://statweb.stanford.edu/ tibs/sta306bfiles/ cluster - svm - etc Rob Tibshirani

https://cs.fit.edu/ dmitra/ArtInt/ Artificial Intelligence Florida Institute of Technology Instructor: Debasis Mitra

http://www.sci.utah.edu/ shireen/pdfs/tutorials/Elhabian$_L$*DA*09.*pdf*

http://users.stat.umn.edu/ helwig/teaching.html Nathaniel E. Helwig University of Minesota (USA)

http://www.utstat.toronto.edu/ brunner/oldclass/302f14/

REAL DATA CASES

Toni Monleón-Getino (basat en diversos materials lliures)

Novembre 2020

Examples to work 1

PROBLEMA DEL DOSIER DE PROBLEMAS DE LOS PROFESORES DR MIQUEL SALICRU - TONI ARCAS. Los datos de este ejercicio se corresponden con un estudio en que se pretendía analizar la relación entre la biomasa de Spartina alterniflora (una alga de humedales) y cinco variables ambientales: X1 = Salinidad (%), X2 = Acidez (pH), X3 = Potasio (ppm), X4 = Sodio (ppm), X5 = Zinc (ppm), Y = Biomasa (g/m2)

Fuente: Rawlings, J.O., Applied Regression Analysis. A research tool. (Cap. 5). Wadsworth Brooks.

- Obtener el modelo de regresión lineal múltiple (Y=variable dependiente).
- Estudiar la significación de los coeficientes de regresión y la significación global del modelo.
- Obtener intervalos de confianza para los coeficientes de regresión.
- Representar los datos en un espacio de dimensión reducida.
- Agrupar los puntos utilizando sólo las variables de tipo X mediante un cluster jerárquico u otra técnica de agrupación.
- ¿Puede aplicarse un análisis discriminante LDA? ¿Por qué?.

Examples to work 1

Copy this data to the text file "Spartina_alterniflora.txt"and do the analysis required. Interpret results.

Dato	Salinidad	Ph	Potasio	Sodio	Zinc	Biomasa
1	33	5.00	1441.67	35184.5	16.4524	676
2	35	4.75	1299.19	28170.4	13.9852	516
3	32	4.20	1154.27	26455.0	15.3276	1052
4	30	4.40	1045.15	25072.9	17.3128	868
5	33	5.55	521.62	31664.2	22.3312	1008
6	33	5.05	1273.02	25491.7	12.2778	436
7	36	4.25	1346.35	20877.3	17.8225	544
8	30	4.45	1253.88	25621.3	14.3516	680
9	38	4.75	1242.65	27587.3	13.6826	640
10	30	4.60	1282.95	26511.7	11.7566	492
11	30	4.10	553.69	7886.5	9.8820	984
12	37	3.45	494.74	14596.0	16.6752	1400
13	33	3.45	526.97	9826.8	12.3730	1276
14	36	4.10	571.14	11978.4	9.4058	1736
15	30	3.50	408.64	10368.6	14.9302	1004
16	30	3.25	646.65	17307.4	31.2865	396
17	27	3.35	514.03	12822.0	30.1652	352
18	29	3.20	350.73	8582.6	28.5901	328
19	34	3.35	496.29	12369.5	19.8795	392
20	36	3.30	580.92	14731.9	18.5056	236
21	30	3.25	535.82	15060.6	22.1344	392

Examples to work 1

22 28 3.25 490.34 11056.3 28.6101 268
23 31 3.20 552.39 8118.9 23.1908 252
24 31 3.20 661.32 13009.5 24.6917 236
25 35 3.35 672.15 15003.7 22.6758 340
26 29 7.10 525.65 10225.0 0.3729 2436
27 35 7.35 563.13 8024.2 0.2703 2216
28 35 7.45 497.96 10393.0 0.3205 2096
29 30 7.45 458.38 8711.6 0.2648 1660
30 30 7.40 498.25 10239.6 0.2105 2272
31 26 4.85 936.26 20436.0 18.9875 824
32 29 4.60 894.79 12519.9 20.9687 1196
33 25 5.20 941.36 18979.0 23.9841 1960
34 26 4.75 1038.79 22986.1 19.9727 2080
35 26 5.20 898.05 11704.5 31.3864 1764
36 25 4.55 989.87 17721.0 23.7063 412
37 26 3.95 951.28 16485.2 30.5589 416
38 26 3.70 939.83 17101.3 26.8415 504
39 27 3.75 925.42 17849.0 27.7292 492
40 27 4.15 954.11 16949.6 21.5699 636
41 24 5.60 720.72 11344.6 19.6531 1756
42 27 5.35 782.09 14752.4 20.3295 1232
43 26 5.50 773.30 13649.8 19.5880 1400
44 28 5.50 829.26 14533.0 20.1328 1620
45 28 5.40 856.96 16892.2 19.2420 1560

Llegir importar dades

Llegiu dades i importeu fitxer

```
#llegir les dades de l exemple (copiar los datos en un fichero y importadlo en RSTUDIO)
#maneig de dades (DATA-WRANGLING)

getwd()
#setwd() #mirar directorio de trabajo

#importeu el fitxer utilitzant Import_Dataset a RSTUDIO (opcions...), per exemple
library(readr)

Spartina_alterniflora <- read_table2("Spartina_alterniflora.txt")
View(Spartina_alterniflora) #mireu el data frame que esta correctament importat

#treure primera columna, no serveix
Spartina_alterniflora<-Spartina_alterniflora[,-1]
```

Exemple 1: Regressió múltiple

Llegiu dades i feu una regressió múltiple. Analitzeu els resultats i feu-ne diagnòstic

```
#llegir les dades de l exemple (copiar los datos en un fichero y importadlo en RSTUDIO)
setwd(wk) #mirar directorio de trabajo
Spartina_alterniflora <- read.csv2("Spartina_alterniflora.txt", header=T)
Spartina_alterniflora<-Spartina_alterniflora[,-1]
#Model linial de regressi[U+FFFD] associat:
mod_spar <- lm(formula = Biomasa ~ ., data = Spartina_alterniflora)
#Obtenim el summary del model: Multiple R-squared; Adjusted R-squared i F-statistic
summary(mod_spar)
anova(mod_spar)
#Podem fer un grafic bivariant entre totes les variables per tal d observar
# si existeix rlacio entre les variables "visualment" i intentar triar un model "millor" sols amb
      variables
#que estiguin "realment" relacionades amb la biomasa
plot(Spartina_alterniflora)
#i tambe:
#Coeficient de Correlacio de Pearson entre les variables:
cor(Spartina_alterniflora) # Matriu de correlacions
mod_spar <- lm(formula = Biomasa ~ ., data = Spartina_alterniflora)
summary(mod_spar)
#diagnosi
plot(mod_spar)
```

Exemple 1: Regressió 2

```
# interval de confianca dels coef.
confint(mod_spar)

#Es recomanable utilitzar un metode de seleccio de variables:
g <- lm(Biomasa ~ Salinidad+Ph+Potasio+Sodio+Zinc, data = Spartina_alterniflora)
#model amb totes les variables independents
summary(g)

#Utilitzem un metode de seleccio de variables Backward
step(g, direction="backward") # Backward selection (si no hi ha scope )
step(g, direction="forward")

#diagnosi
plot(mod_spar)
```

Exemple 1: PCA

```
#If we use prcomp() function, we indicate 'scale=TRUE' to use correlation matrix
pca <- prcomp(Spartina_alterniflora,scale=T)
pca

summary(pca)

#This gives us the standard deviation of each component, and the proportion of variance
      explained by each component.
#The standard deviation is stored in (see 'str(pca)'):
pca$sdev

#plot of variance of each PCA.
#It will be useful to decide how many principal components should be retained.
screeplot(pca, type="lines",col=3)

#The loadings for the principal components are stored in:
pca$rotation # with princomp(): pca$loadings

#biplot of first two principal components
biplot(pca,cex=0.8)
abline(h = 0, v = 0, lty = 2, col = 8)
```

Exemple 1: HIERARCHICAL CLUSTER

```
#scale the date
Spartina_alterniflora_sc <- as.data.frame(scale(Spartina_alterniflora[,1:5]))
summary(Spartina_alterniflora_sc)
#compute the euclidean distance
dist_mat <- dist(Spartina_alterniflora_sc, method = 'euclidean')
#do the cluster
hclust_avg <- hclust(dist_mat, method = 'average')
plot(hclust_avg)
#work with the cluster to obtain 3 groups
cut_avg <- cutree(hclust_avg, k = 3)

plot(hclust_avg)
rect.hclust(hclust_avg , k = 3, border = 2:6)
abline(h = 3, col = 'red')

#work with the cluster to obtain 3 groups
suppressPackageStartupMessages(library(dendextend))
avg_dend_obj <- as.dendrogram(hclust_avg)
avg_col_dend <- color_branches(avg_dend_obj, h = 3)
plot(avg_col_dend)
```

Exemple 1: LDA

```
#do 3 groups based on Biomasa
Spartina_alterniflora$group<-cut(Spartina_alterniflora$Biomasa,breaks = c
      (0,400,1500,5000000))

#do a LDA analysis and check it
library(MASS) #Load package 'MASS' to perform LDA
fit.LDA = lda( group ~ Salinidad + Ph + Potasio + Sodio + Zinc, Spartina_alterniflora)
fit.LDA

#represent LDA functions
plot(fit.LDA, col = as.integer(Spartina_alterniflora$group))


#Perform classification
fit.LDA.C = predict(fit.LDA, newdata=Spartina_alterniflora[,c(1:5)])$class
fit.LDA.C

#Determine misclassification with confussion matrix and accuracy
table(as.matrix(Spartina_alterniflora[,7]),as.matrix(fit.LDA.C))
library(caret)
confusionMatrix(unlist(c(Spartina_alterniflora[7])),fit.LDA.C)
```

Example 2: Wines

Intenteu reproduir l' exemple d' uns vins de diferents cultivars caracteritzats químicament que trobareu a:
Example wines: multivariate analysis in R from Avril Coghlan

Other examples and dataframes

- More data-frames: Extra data
- Other examples: Exploratory Data with R from Roger D. Peng
- Examples from Stanford University
- Multivariate Analysis with R from Edward R. Tufte

Data Science: principis

Supervised Learning

Hansel Gómez, Ph.D.

IRB-BCN

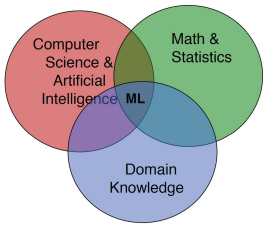
Data Science: principis

Introduction to Supervised Classification Methods

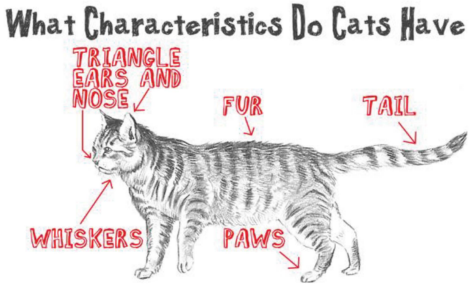
A practical approach to machine learning

1. What is Machine Learning (ML)
2. Modeling the ML problem
3. ML Pipeline
4. Learning concepts and theory
5. Models in Machine Learning
6. About the software

Data Science: principis



1. What is Machine Learning?



Machine Learning (ML) is about coding programs that automatically adjust their performance from exposure to information encoded in data. This learning is achieved via a parameterized model with tunable parameters automatically adjusted according to a performance criteria.

There are three major classes of ML:

Beer-diaper syndrome

1. **Unsupervised learning** : Algorithms which learn from a training set of unlabeled examples, using the features of the inputs to categorize inputs together according to some statistical criteria. Clusterization, **Association Analysis**.
2. **Supervised learning** : Algorithms which learn from a training set of labeled examples to generalize to the set of all possible inputs.
3. **Reinforcement learning** : Algorithms that learn via reinforcement from a critic that provides information on the quality of a solution, but not on how to improve it. Improved solutions are achieved by iteratively exploring the solution space.

Data Science: principis

2. Modeling the machine learning problem

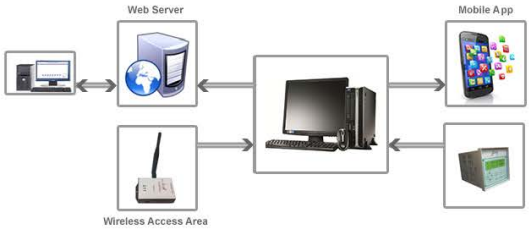
The first step to apply data science and machine learning is identifying an interesting question to answer. According to the type of answer we are seeking we are directly aiming for a certain set of techniques.

- If our question has a discrete set of answers, this is a **classification** problem (binary/multiclass).
  - Given a client profile and past activity, which are the financial products she would be most interested in?
  - Given an Magnetic Resonance Image, is there a tumor in it?
  - Given the past activity associated to a credit card, is the current operation a fraud?
- If our question is a prediction of a continous quantity, we are in front of a **regression** problem.
  - Given the description of an appartment, which is the expected market value of the flat?
  - Given the past records of user activities on Apps, how long is a certain client be hocked to our App?
  - Given my skills and marks in computer science and maths, what mark will I achive?

Data Science: principis

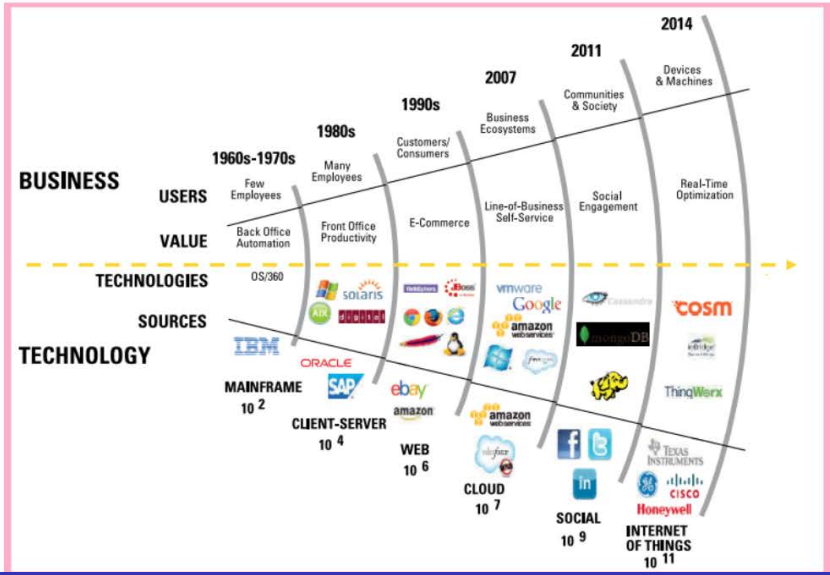
3. Machine Learning Pipeline

ACQUIRE | PREPARE | ANALIZE | REPORT | ACT



Data Acquisition

- ✓ Identify Data Sources (sensors, web, social media etc ...)
- ✓ Collect Data
- ✓ Integrate Data
- ✓ Static or dynamic study ?



Data Science: principis

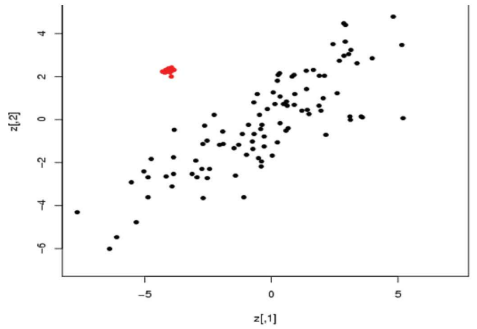
3. Machine Learning Pipeline

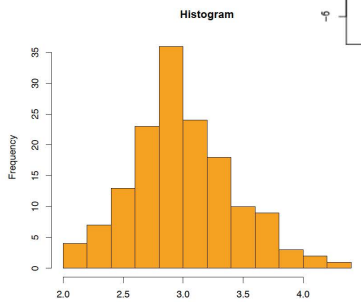
ACQUIRE → **PREPARE** → ANALIZE → REPORT → ACT

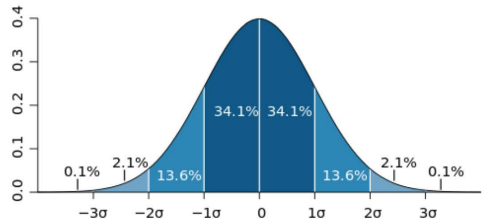
1. EXPLORE (EDA)

GOAL (Understand data*)

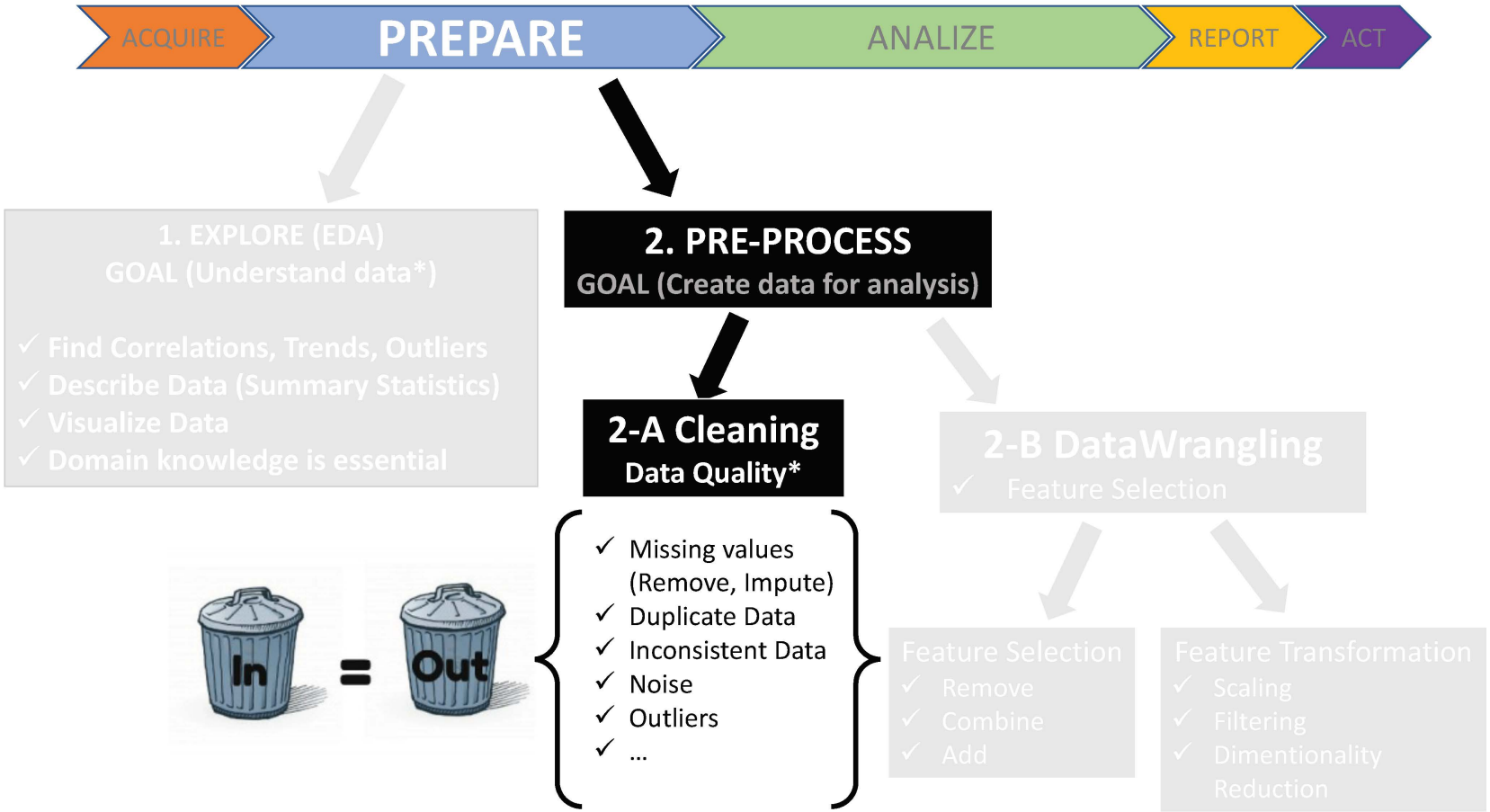
- ✓ Find Correlations, Trends, Outliers
- ✓ Describe Data (Summary Statistics)
- ✓ Visualize Data
- ✓ Domain knowledge is essential





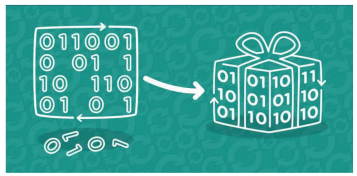


Data Science: principis



Data Science: principis

2. PRE-PROCESS
GOAL (Create data for analysis)



Raw data:

Advantages: No domain specific knowledge is required.

Drawbacks: Highly redundant in many cases and usually span very large dimensional spaces. Unknown discriminability.

Feature selection:

Advantages: Attempt to capture discriminant information in the data. Lower dimensionality and complexity.

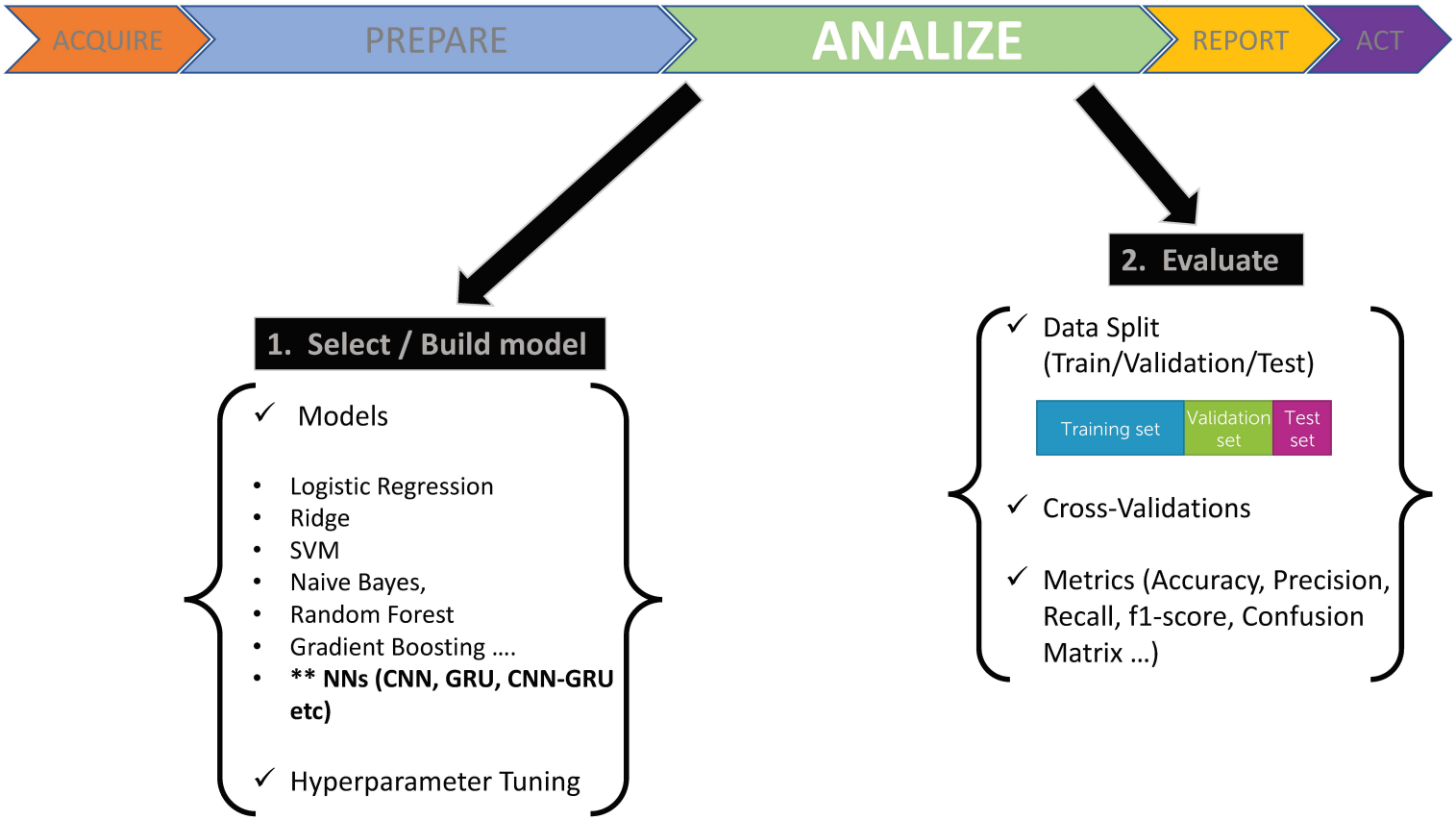
Drawbacks: Domain specific knowledge is required.

2-B DataWrangling
- ✓ Feature Selection

Feature Selection
- ✓ Remove
- ✓ Combine
- ✓ Add

Feature Transformation
- ✓ Scaling
- ✓ Filtering
- ✓ Dimentionality Reduction

Data Science: principis



Data Science: principis

Split of the data

As we have seen the process of assessing the performance of the classifier by estimating the **generalization error** is called **testing**. And the process of selecting a model using the estimation of the generalization error is called **validation**. There is a subtle but critical difference in both and we have to be aware of it when dealing with our problem.

- **Testing data** is used only for assessing performance and will never be used in the learning process.
- **Validation data** is used to explicitly select the parameter with best performance according to an estimation of the generalization error. This is a form of learning.
- **Training data** is used for learning the model instance from a model class.

➢ **A practical issue:** once selected the model we use the complete training set to train the final model.

Data Science: principis

Split of the data

- If we want to know the performance of our model we have to use unseen data. Thus, we may proceed in the following way:
  - Split the training set in training and testing data. For example, use 30% of the training set for testing purposes. This data is hold out and will only be used to assess the performance of the method.
  - Use the remaining training data for selecting the hyper-parameters by means of **cross-validation**.
  - Train the model with the selected parameter and assess the performance using the testing data set.

➢ **A practical issue:** Observe that by splitting in three sets the classifier is trained with a smaller fraction of the data.

Data Science: principis

Is a classifier with 90% accuracy good? Depends...

2010 data shows:
"90% emails sent are spam!"

Predicting every email is spam gets you 90% accuracy!!!

Majority class prediction

Amazing performance when there is class imbalance
(but silly approach)
- One class is more common than others
- Beats random (if you know the majority class)

Data Science: principis

Confusion Matrix and Common Metrics in Classification

		"Golden Standard" (Real Truth Values)		
		Positive	Negative	
Observed	Predicted positive	True Positive	False Positive (Type 1 error)	Precision
	Predicted Negative	False Negative (Type 2 error)	True Negative	
		Recall/ Sensitivity	(Specificity)	

$$\text{Accuracy} = \frac{TP + TN}{TN + FP + FN + TP}$$

$$\text{Precision} = \frac{TP}{FP + TP}$$

$$\text{Recall} = \frac{TP}{FN + TP}$$

$$\text{F1-score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Data Science: principis

So, always be digging in and asking the hard questions about reported accuracies

- Is there class imbalance?
- How does it compare to a simple, baseline approach?
  - Random guessing
  - Majority class
  - ...
- Most importantly: *what accuracy does my application need?*
  - What is good enough for my user's experience?
  - What is the impact of the mistakes we make?

Data Science: principis

Some notes on the learning process

The main goal of any learning process is to achieve the maximum predictive power (*accuracy*). This is minimize the error. However, there are three other important properties we usually desire our models to have:

- ✓ **Simplicity** - how much fiddling do we need for the method to work? Can I modify it to handle the particularities of my problem?
- ✓ **Speed** - How long does it take to train a reliable model? (training time) Can I use it in embedded and real time applications? (testing time), How long do I have to wait for processing my 1YB (yottabyte - 1e24 Bytes) dataset?
- ✓ **Interpretability** - Why did it make these predictions? It happens that accuracy trades off with all the rest of the desirable properties.

Data Science: principis

4. Learning concepts and theory

4.1. What is learning?

Training error or *in-sample error*, E_{in} refers to the error measured over all the observed data samples in the training set.

$$E_{\text{in}} = \frac{1}{N} \sum_{i=1}^{N} e(x_i, y_i)$$

Testing error or *generalization error*, E_{out}, refers to the expected error on unseen data.

$$E_{\text{out}} = \mathbb{E}_{x,y}(e(x, y))$$

Common **instantaneous error** in classification :

$$e(x_i, y_i) = I[h(x_i) = y_i] = \begin{cases} 0 \text{ if } h(x_i) = y_i \\ 1 \text{ otherwise} \end{cases}$$

We can empirically estimate the generalization error by means of **cross-validation** techniques and observe that

$$E_{out} \geq E_{in}$$

Data Science: principis

4. Learning concepts and theory

4.1. What is learning?

The goal of learning is to ***minimize the generalization error***, but how can we guarantee this minimization only using training data?

From the above inequality it is easy to derive a couple of very intuitive ideas:

- Because E_{out} is greater than or equal to E_{in}, it is desirable to have $E_{in} \rightarrow 0$
- Additionally, we also want the training error behavior to track the generalization error, i.e. $E_{out} \approx E_{in}$.

We can rewrite this second condition as

$$E_{in} \leq E_{out} \leq E_{in} + \Omega,$$

with $\Omega \rightarrow 0$.

Data Science: principis

Probably approximately correct learning

We would like to characterize Ω in terms of our problem parameters, i.e. number of samples (***N***), dimensionality of the problem (***d***), etc.

Statistic analysis offers an interesting characterization of this quantity

$$E_{\text{out}} \leq E_{\text{in}} + \mathcal{O}\left(\sqrt{\frac{\log C}{N}}\right)$$

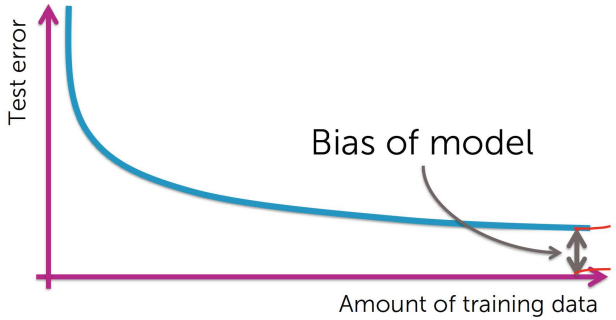
where ***C*** is a measure of complexity of the *model class* we are using. Technically, we may refer to this model class also as the hypothesis space.

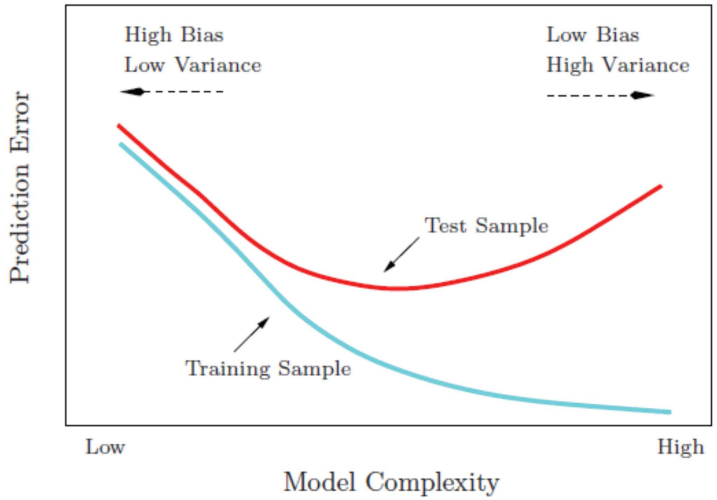
- ✓ **Which will be the effect of having a large number of data ?**
- ✓ **Will selecting a model with small complexity reduce the out of sample error ?**

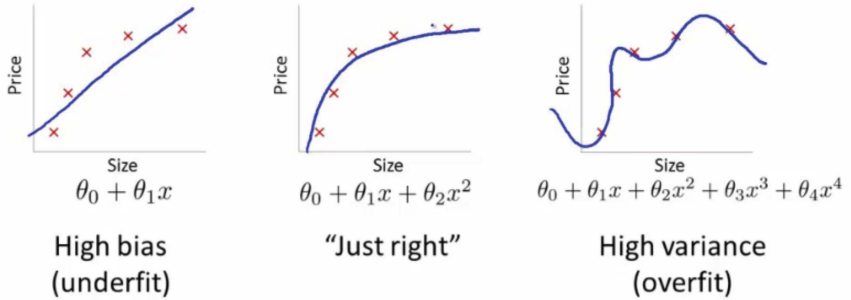
Data Science: principis

Bias-variance tradeoff

Is there a limit?
Yes, for most models...







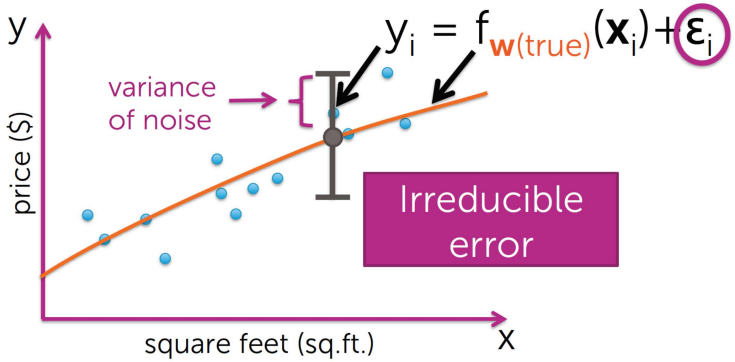
Data Science: principis

3 sources of error

In forming predictions, there are 3 sources of error:

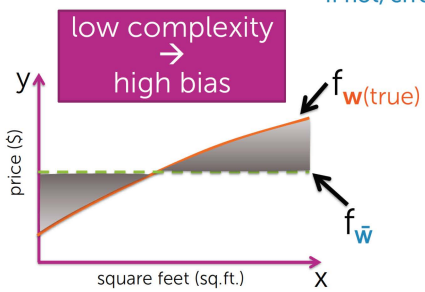
1. Noise
2. Bias
3. Variance

Data inherently noisy



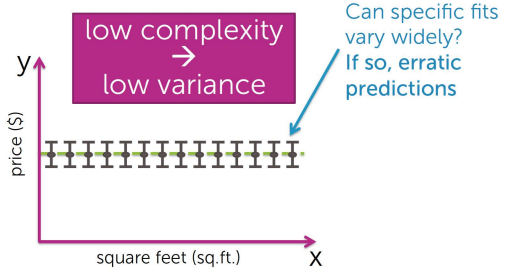
Bias contribution

$Bias(x) = f_{w(true)}(x) - f_{\bar{w}}(x)$ ← Is our approach flexible enough to capture $f_{w(true)}$? If not, error in predictions.

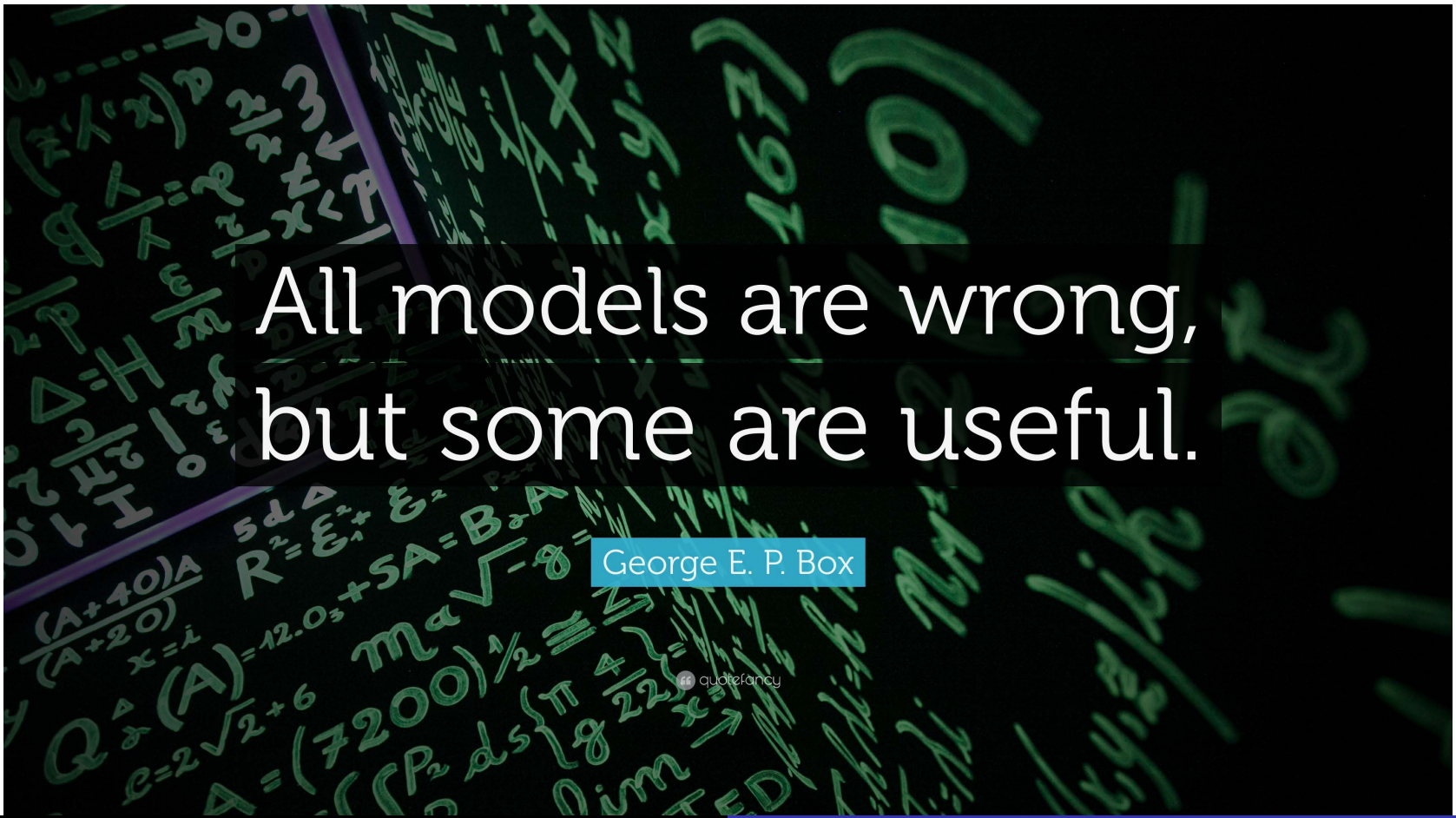


Variance contribution

How much do specific fits vary from the expected fit?



Data Science: principis



Data Science: principis

Components of the ML algorithms

1. The model class/hypothesis space defines the family of mathematical models that will be used. The target decision boundary will be approximated from one element of this space. For example, all possible lines in $\mathbf{R}^2$ for a lineal model. Model classes define the geometric properties of the decision function.

There are different taxonomies but the most well-known are the *families* of **linear** and **non-linear** models. These families usually depend on some parameters. And the solution to a learning problem is the selection of a particular set of parameters, i.e. the selection of an instance model from the model class space. The model class space is also called **hypothesis space**.

The selection of the best model will depend on our problem and what we want to obtain from the problem. The primary goal in learning is usually achieving the minimum error/maximum performance. But according to what else we want from the algorithm we will find different algorithms. Other common desirable properties are **interpretability**, **behavior** in front of missing data, **fast** training, etc.

Data Science: principis

Components of the ML algorithms

2. The problem model formalizes and encodes the desired properties of the solution. In many cases this formalization takes the form of an **optimization** problem. In it most basic instantiation, the problem model can be the **minimization of an error function**. The error function measures the difference between our model and the target one. For example, in classification the ideal error function is the **0-1 loss**. This function takes value 1 when we incorrectly classify a training sample and zero otherwise. For regression problems is commonly **RSS**. Problem model can also be used to impose other constraints on our solution, such as finding a smooth approximation, small complexity model, sparse solution, etc.

Data Science: principis

Components of the ML algorithms

3. The learning algorithm is an **optimization/search method** or algorithm that given a model class fits it to the training data according to the error function. According to the nature of our problem there are many different algorithms. In general, we are talking about finding the *minimum error* approximation or *maximum probable model*. In those cases, if the problem is convex/quasi-convex we will typically use first or second order methods (i.e. ***gradient descent***, *coordinate descent, Newton's method, Interior Point methods*, etc). Other searching techniques such as genetic algorithms or monte-carlo techniques can be used if we do not have access to the derivatives of the objective function.

Data Science: principis

5. Models in Machine Learning

Generative and discriminative models.

There are two complementary visions of the learning problem according to the problem they solve, generative vs discriminative models.

- ✓ **Generative models** goal is modeling the data. This consists of estimating the joint probability density function of x, y, $P(x,y)$. With this description, the problem of classification is selecting the model that maximizes the posterior probability of the labels given the data, $P(y|x)$. This is done by applying Bayes rule to relate the posterior distribution to the likelihood and the priors.

$$\underset{w}{\text{maximize}}\ P_w(y|x)$$

- ✓ **Discriminative models** are concerned in finding a good approximation of the decision function even if it means losing the information about the concrete description of the data. In this setting we may find *Maximum Likelihood Estimated* methods such as *logistic regression* and other explicit function models such as *SVM*.

Data Science: principis

Linear Models

Linear models are defined as $h : \mathbf{R}^n \to \mathbf{R}$ so that $h(x) = a^T x + b \qquad a, x \in \mathbf{R}^n \qquad b \in \mathbf{R}$

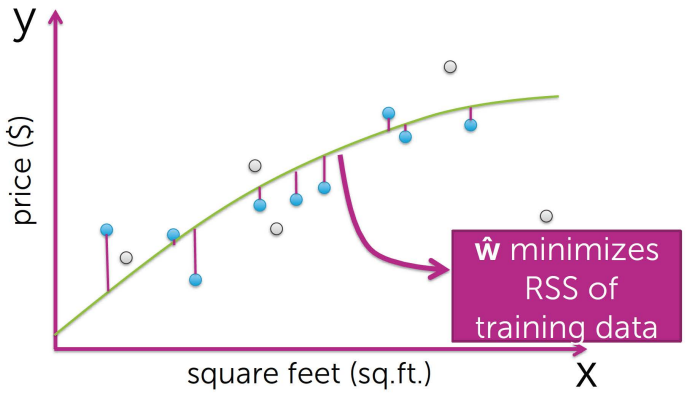
Compute training error

1. Define a loss function $L(y, f_{\hat{w}}(\mathbf{x}))$
   - E.g., squared error, absolute error,...

2. Training error
   = avg. loss on houses in training set
   $= \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_{\hat{w}}(\mathbf{x}_i))$

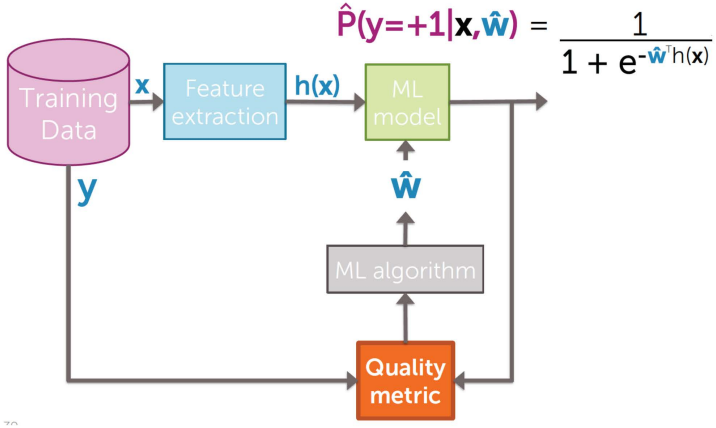
fit using training data

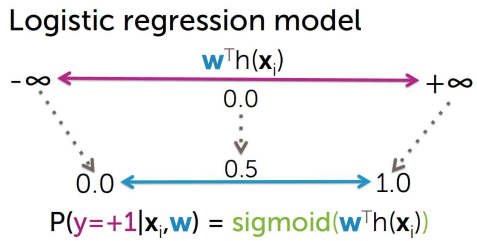
Example:
Fit quadratic to minimize RSS



$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Data Science: principis





Maximum likelihood estimation (MLE):
Measure of fit = Data likelihood

- Choose coefficients **w** that maximize likelihood:

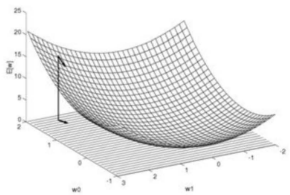
$$\prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

- Typically, we use the log of likelihood function
(simplifies math and has better convergence properties)

$$\ell(\mathbf{w}) = \ln \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave ! Find optimum with gradient ascent



Gradient: $\nabla_{\mathbf{w}} l(\mathbf{w}) = [\frac{\partial l(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial l(\mathbf{w})}{\partial w_n}]'$

Update rule: $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$ (Learning rate, η>0)

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent much better (see reading)

Data Science: principis

The Naive Bayes classifier

Naive Bayes is an instance of a Bayessian classifier. In this framework, the problem of classification consists of selecting the class with *Maximum A Posteriori (MAP)* probability, i.e.

$$\hat{y} = \arg\max_y p(y|x).$$

In order to find this quantity we use the Bayes equation,

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x),$$

and

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}.$$

In order to compute the MAP the quantities *p(x|y), p(y), p(x)* have to be estimated from observed data.

Note that p(x) is a constant value and it does not affect the decision, thus we just need to compute

$$P(y|x) \propto P(y)P(x|y)$$

Up to this point, the description of the classifier is general for any Bayessian classifier. **Naive Bayes** additionally assumes that *x* is composed of a set of *d* independent variables. This allows to rewrite the likelihood term as

$$p(x_1, x_2, ..., x_N|y) = p(x_1|y)p(x_2|y)...p(x_N|y) = \prod_{i=1}^{N} p(x_i|y)$$

In the end, the Naive Bayes classifier has the following form,

$$p(y|x) \propto p(y) \prod_{i=1}^{N} p(x_i|y)$$

Data Science: principis

The Naive Bayes classifier

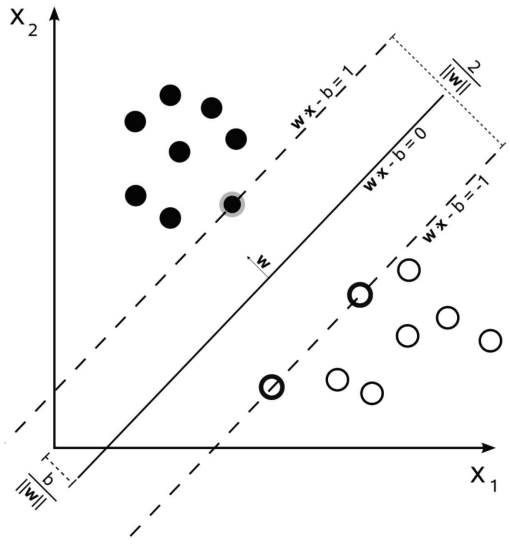
In many cases the prior *p(y)* is unknown or simply we prefer to use a non-informative prior In that case the formulation is simplified to the **Maximum Likelihood Estimate**.

The final step is to to estimate the conditioned probabilities. There are two classical variants the **Multinomial Naive Bayes** and the **Bernoulli Naive Bayes**. The difference between both lies in the goal of what they are modeling.

Data Science: principis

Linear Support Vector Machine (SVM)

A hyperplane in R^d is defined as an affine combination of the variables: $\pi \equiv a^T x + b = 0$



Features:

- A hyperplane splits the space in two half-spaces. The evaluation of the equation of the hyperplane on any element of one of the half-space is a positive value. It is a negative value for all the elements in the other half-space.
- The distance of a point $x \in R^d$ to the hyperplane π is $d(x, \pi) = \frac{a^T x + b}{\|a\|_2}$
- **C** is the trade-off parameter that roughly balances margin and misclassification rate. This formulation is also called **soft-margin SVM.**

SVM finds the boundary with maximum distance/**margin** to both classes

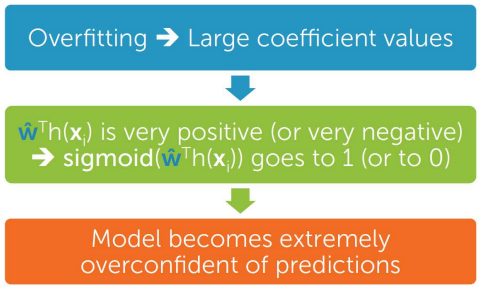
Data Science: principis

Regularization

Regularization accounts for estimating the value of Ω in our out-of-sample inequality. In other words, it models the complexity of the technique. This usually becomes implicit in the algorithm but has huge consequences in real applications. There are two kinds of standard regularization strategies:+

- ✓ **L2 regularization**: Intuitively, L2 regularization is in many cases a surrogate of the notion of smoothness. In this sense, low complexity means smooth models.
- ✓ **L1 regularization**: L1 regularization force **sparse solution**. This is useful for interpretability or when the number of parameters is so large that we only want a few active ones for computational issues. Although they are used to deal with overfitting, they trade-off with the error function in the objective and are governed by a **hyper-parameter**. Thus, we still have to select this parameter by means of model selection.

The subtle (negative) consequence of overfitting in logistic regression



Consider resulting objective

What if $\hat{\mathbf{w}}$ selected to minimize

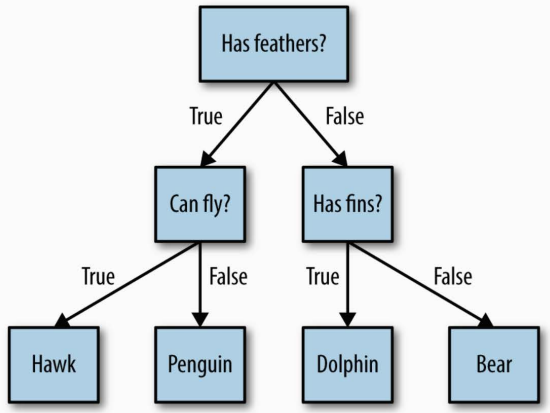
$$\ell(\mathbf{w}) - \lambda||\mathbf{w}||_2^2$$

tuning parameter = balance of fit and magnitude

Ridge / Lasso Regression

Data Science: principis

Decision Trees



Decision trees are another kind of intuitive classification strategy based on the *divide and conquer paradigm*.

The basic idea in decision trees is to partition the space in patches and fit a model in that patch. There are two questions to answer in order to implement this solution:

1. How do we partition the space?

2. What model to use in each patch?

In classification trees the second question is straight forward, each patch is given the value of a label and all data falling in that part of the space will be predicted as such.

Data Science: principis

Decision Trees

Modeling :

- ✓ Splits using axis-orthogonal hyperplanes. This is the key that allows interpretability of the results.
- ✓ At each internal node we test a value of a feature. A feature and a threshold are stored for each internal node.
- ✓ Leaves makes the class prediction. If leaves are pure, we have to store the class label. If leaves are impure, then the fraction of samples for each class is stored and its frequency is returned when queried.

Data Science: principis

Decision Trees

What is great about decision trees?

- ✓ Trees are easy for humans to interpret. It can be seen as a set of rules. Each path from root to one leaf of the tree is an AND combination of the thresholded features.
- ✓ Given a finite data set, decision trees can express any function of the input attributes. In R^d we can isolate every point in the data set by constructing a box around each of them.
- ✓ There can be more than one tree that fits the same data.

Data Science: principis

Decision Trees

Learning the Tree:

Greedy algorithms choose the current best binary partition without taking into account its impact on the quality of subsequent splits.

The algorithm idea is as follows:

1. Initialize the algorithm with a node associated to the full data set.

   While the list is not empty:

   1. Retrieve the first node from the list.
   2. Find the data associated to that node.
   3. Find a splitting point.
   4. If the node is splittable, create the nodes linked to the parent node and put them in the exploration list.

Data Science: principis

Decision Trees

Splitting Criterion:

There are many different splitting criteria. The most common ones are

- ✓ Misclassification error
- ✓ Gini index
- ✓ Cross-entropy/Information gain/Mutual information

Misclassification error splits greedily select the split that corrects more data at each point.

Data Science: principis

Decision Trees

Splitting Criterion:

There are many different splitting criteria. The most common ones are

- ✓ Misclassification error
- ✓ Gini index
- ✓ Cross-entropy/Information gain/Mutual information

Gini index and cross-entropy probabilistically model the notion of *impurity* of a node. The split is chosen so that the average purity of the new nodes is maximized. As we descend in the tree the purity increases and eventually converge to pure leaves.

Data Science: principis

Decision Trees

Trees and overfitting:

Because trees are very expressive models they can model any training set perfectly and easily overfit.

There are two ways of avoiding overfitting in trees:

1. Stop growing the tree when the split is not statistically significant.
2. Grow a full tree and post-prune.

One of the simplest ways of post pruning is **"reduced error prunning".** It goes like this:

1. Split data into training and validation
2. Create a candidate tree on the training set
3. Do until further pruning is harmful
    1. Evaluate impact on the validation set of removing each posible node (with descendants)
    2. Greedily remove the node that improves the performance the most.

Pruning is not implemented in sklearn at this moment

Data Science: principis

Extending Support Vector Machines to the Non-Linear Case. A very brief introduction to kernels.

Linear models can be quite limiting in low-dimensional spaces, as lines and hyperplanes have limited flexibility. One way to make a linear model more flexible is by adding more features—for example, by adding interactions or polynomials of the input features.

Moreover, a linear model can model non-linear boundaries provided if we **explicitly** map original data non-linearly. For example, we can create a linear model with features squared. This will lead to a quadratic boundary with respect to the original space.

There is another way of **implicitly** encoding non-linearities by means of **kernels**.

The kernel encodes the notion of similarity between two data points. The change in the formulation involve the introduction of several concepts from mathematical analysis

As a result, any regularized cost function optimization problem such as SVM has a solution of following form,

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

Data Science: principis

Extending Support Vector Machines to the Non-Linear Case. A very brief introduction to kernels.

Functional Analysis

- ✓ Brach of mathematical analyisis that deals with spaces of functions.
- ✓ Hilbert space must be introduced so that similarity and distance among functions can be measured.
- ✓ Intuitively, it generalizes the classical Euclidean space to infinite dimensions, and thus, to functional spaces.

One particular functional space is the **Reproducing Kernel Hilbert Space (RKHS).** In this space a function evaluated on a point x is defined by the inner product of the function and the kernel evaluated on that point, i.e. $f(x)=\langle f(\cdot),K(x,\cdot)\rangle$, where K is the kernel. This is called *Riesz representation* and it is the key for showing the most important result for our problems, *The Representer's theorem*.

Kernel trick works by directly computing the distance (more precisely, the scalar products) of the data points for the expanded feature representation, without ever actually computing the expansion.

Data Science: principis

Extending Support Vector Machines to the Non-Linear Case. A very brief introduction to kernels.

Functional Analysis

The kernel has to be a positive semi-definite function, such as:

- ✓ Linear kernel: $k(x_i, x_j) = x_i^T x_j$
- ✓ Polynomial kernel: $k(x_i, x_j) = (1 + x_i^T x_j)^p$
- ✓ Radial Basis Function kernel: $k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$

On a practical side you can define a kernel by using your favorite distance $d(x_i, x_j)$ and defining the kernel as

$$k(x_i, x_j) = e^{-\gamma d(x_i, x_j)}, \quad \gamma > 0$$

where γ is a hyper-parameter that controls the decay of the exponential.

Data Science: principis

Extending Support Vector Machines to the Non-Linear Case. A very brief introduction to kernels.

Kernels **implicitly** encode a non-linear transformation and SVM finds the optimal hyperplane. By combining both concepts we have a linear method applied on a data on a transformed space, but we do not have to provide the explicit transformation.

This becomes incredibly useful since feature mapping from a radial basis function kernel is a mapping into a ∞-dimensional space.

Gain some intuition about how kernels work with the following video:

https://www.youtube.com/watch?v=3liCbRZPrZA

Data Science: principis

Ensemble learning

- When we want to purchase a product we usually read user's reviews.
- Before undergoing a major surgery procedure we seek the opinion of different experts.

Ensemble learning mimics one of the human *uncertainty reduction mechanism*, seeking additional opinions before making a major decision.

Ensemble learning is divided in two steps:

1. **Train a set** of classifiers
2. **Aggregate** their results.

Data Science: principis

Ensemble learning

There are different reasons for using ensemble learning in practice:

- ✓ **Statistical reasons**: The combination of outputs of different classifiers may reduce the risk of an unfortunate selection of a poorly performing classifier.
- ✓ **Large scale data sets**: It makes little sense to only have one classifier on very large sets of data. Partition data in smaller subsets and aggregate seems like a good idea.
- ✓ **Divide and conquer**: Some problems too difficult for a single classifier to solve. The decision boundary may be too complex or lie outside the space of functions of the classifier.
- ✓ **Data fusion**: Different source fusion is usually a problem. One usually faces data coming from heterogeneous sources and the question is how to fuse these data. One solution is to train one classifier per source and the fuse the decision of those experts.

Data Science: principis

Ensemble learning

Diversity

One condition required for the system to work is that errors on different classifiers should be made on different samples in order for the strategic combination of the classifiers to correct possible errors in the judgement of the class o a particular instance. This effect has been called diversity.

Diversity can be obtained in different ways:

- ✓ Using different training sets. Use resampling strategies to obtain different optimal classifiers. This effect is correlated with the notion of stability of the classifier and the concept of bias and variance of the classifier.
- ✓ Using different training parameters for different classifiers.
- ✓ Combining different architectures. (i.e. svm, decission trees, ...).
- ✓ Training on different features. (i.e. random subspaces or random projections)

Data Science: principis

Ensemble learning

Bootstrapping aggregation:

Bootstrapping means resampling the training data set with replacement. Usually the same number of data as the original data set is used.

Bootstrapping aggregation (aka. Bagging) is a ensemble technique that uses multiple bootstrapped copies of the training set to build a set of classifiers. One classifier for each bootstrapped training copy. And then, use a combination technique, such as majority voting, in order to take the final decision.

Bagging performance improvement is due to the reduction of the variance of the classifier while maintaining its bias.

Random Forest:

Random Forest technique introduces a randomization over the feature selected for building each tree in the ensemble in order to improve diversity in an attempt to reduce variance even more. Can be seen as a variant of bagging.

Data Science: principis

Ensemble learning

Gradient Boosting:

In contrast to the random forest approach, gradient boosting works by building trees in a serial manner, where each tree tries to correct the mistakes of the previous one. By default, there is no randomization in gradient boosted regression trees; instead, strong pre-pruning is used.

Gradient boosted trees are frequently the winning entries in machine learning competitions, and are widely used in industry.

Kaggle : https://www.kaggle.com/