Notes machine learning

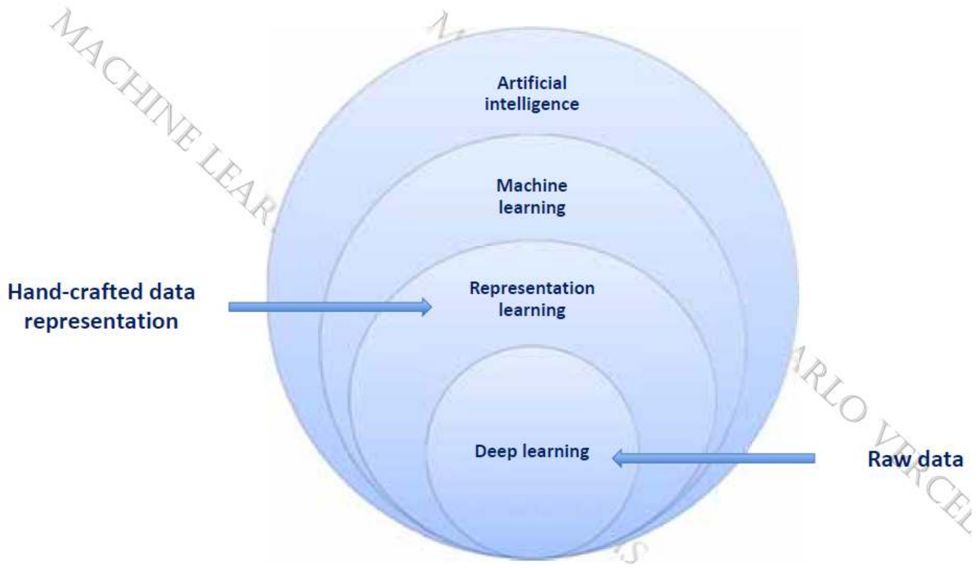
Machine Learning (Politecnico di Milano)

Machine learning

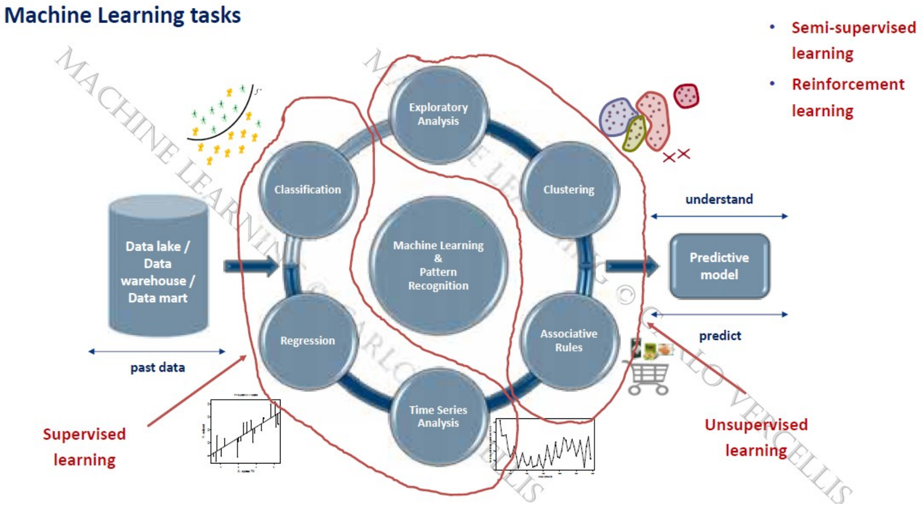
1. Machine learning introduction

- We are surrounded by machine learning : smarter healthcare, Multi-channel, Finance, Log analysis, Homeland Security, Traffic Control, Telecom, Search Quality, Manufacturing, Trading Analytics, Fraud and Risk, Retail: Churn, NBO
- Big Data come on stage, it have an exponential growth, people are more and more connected, internet of things fueled the flow of information and made the Bid data bigger and bigger
- From data insight to data driven strategy
  - Descriptive Analytics
  - Predictive Analytics
  - Perspective Analytics
  - Automated analytics
- Difference between AI and Machine learning :
  - AI :
  - Emulate human behavior : Image video, Sound, Smell & Speech processing
  - Machine learning & Optimization
  - Interacting physically and Socially
  - Machine Learning :
  - Machine learning is centrale in AI : Speech Recognition, Data analysis, Robotics, Computer vision text Analysis
  - Machine learning is a major component in the AI :

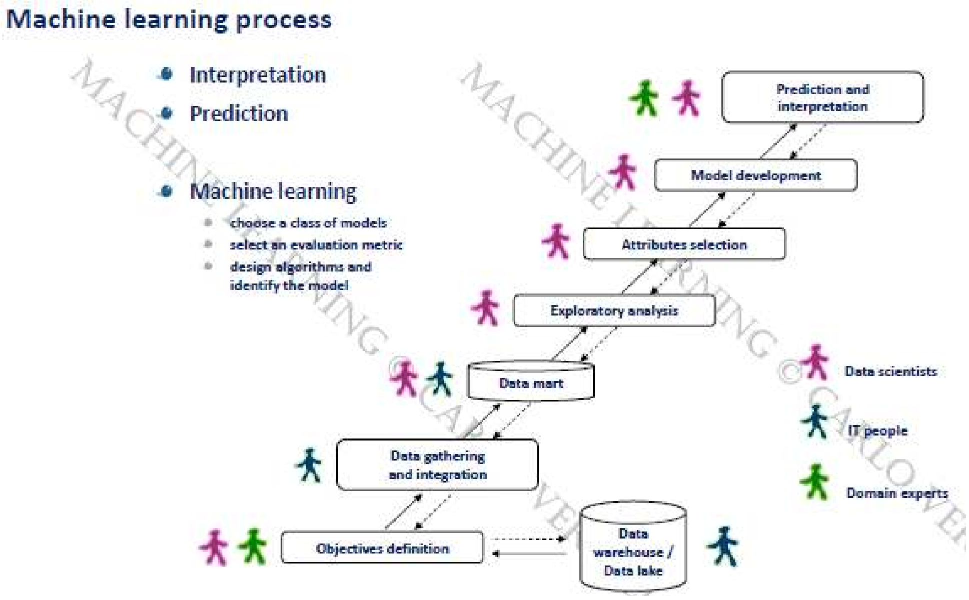
Machine learning is the major component in AI



- Tasks in machine learning :
    - Search of hidden patterns how ?



- Tasks in machine Process :


Machine learning process

- Interpretation
- Prediction
- Machine learning
    - choose a class of models
    - select an evaluation metric
    - design algorithms and identify the model

- Dataset: Two dimensional tables :
    - Rows are the observations
    - Columns are the data that characterize each observations
- Attribute type :
    - Categorical : Binary – True/False or 0/1, Nominal without a natural sorting, Ordinal attribute with a natural sorting
    - Numerical attribute : Discrete or Continuous

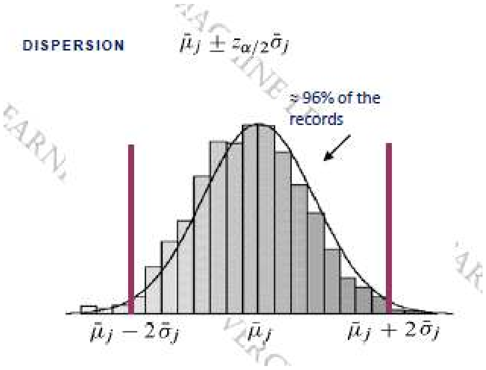
2. Machine learning Data preparation

a) Incomplete data :

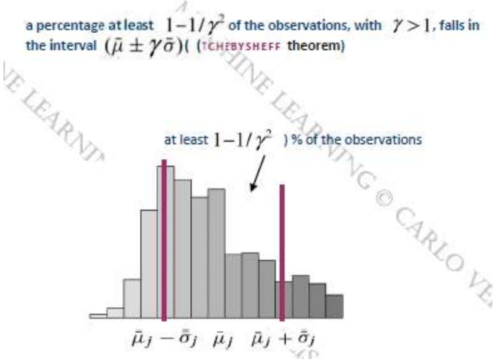
- Elimination (Eliminate rows or colums)
- Inspection
- Identification
- Replacement
    - Mean value of the attribute
    - Mean value of the attribute for the same target class
    - Value estimated using statistical inference

b) Noisy data

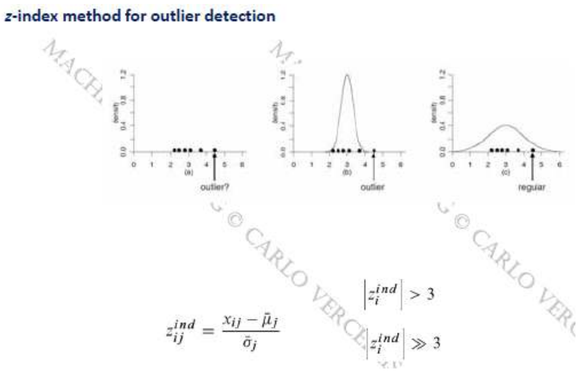
- In case the data follows normal distribution :



- Tchebysheff theorem (General cases) :


a percentage at least $1-1/\gamma^2$ of the observations, with $\gamma > 1$, falls in the interval $(\bar{\mu} \pm \gamma\bar{\sigma})$((TCHEBYSHEFF theorem)

- Z-index for outlier detection :



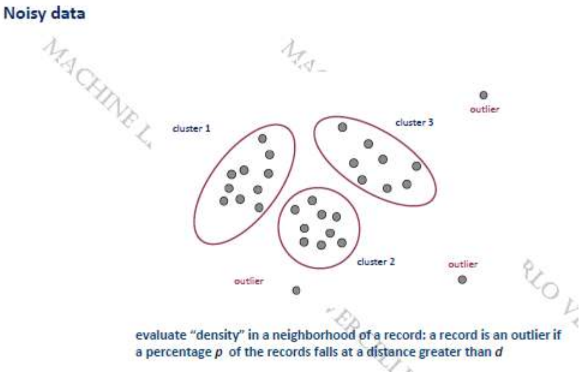
***z*-index method for outlier detection**

$$z_{ij}^{ind} = \frac{x_{ij} - \bar{\mu}_j}{\bar{\sigma}_j}$$

$$\left|z_i^{ind}\right| > 3$$

$$\left|z_i^{ind}\right| \gg 3$$

- Clustering :

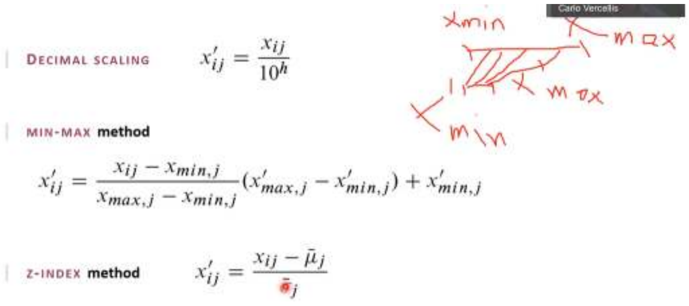


Noisy data

evaluate "density" in a neighborhood of a record: a record is an outlier if a percentage p of the records falls at a distance greater than d

c) Data transformation

Having all the numerical data in the same scale :



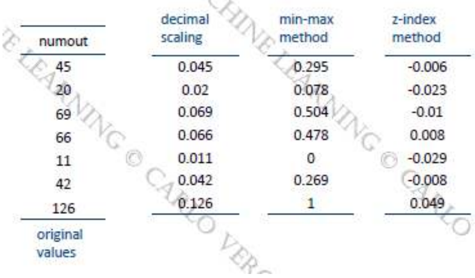
DECIMAL SCALING $$x'_{ij} = \frac{x_{ij}}{10^h}$$

MIN-MAX **method**

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

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

Example :



numout	decimal scaling	min-max method	z-index method
45	0.045	0.295	-0.006
20	0.02	0.078	-0.023
69	0.069	0.504	-0.01
66	0.066	0.478	0.008
11	0.011	0	-0.029
42	0.042	0.269	-0.008
126	0.126	1	0.049
original values			

in the Min Max method the new x'min=0 and x'max=1

d) Data reduction

- improve efficiency in model identification
- preserve model accuracy
- achieve simpler models

How ?

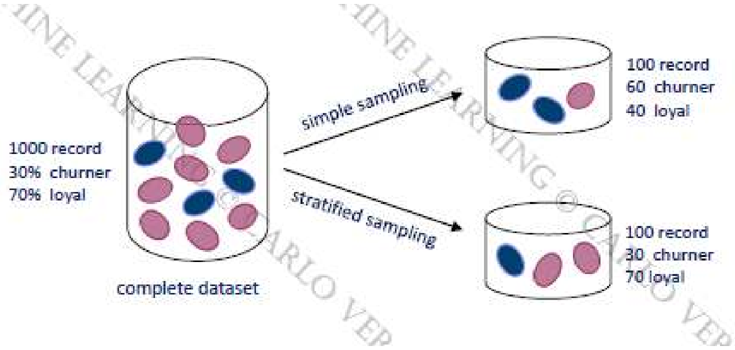
- record reduction by SAMPLING
- attribute reduction by SELECTION or PROJECTION
- values reduction by DISCRETIZATION or AGGREGATIOI

Operate in rows

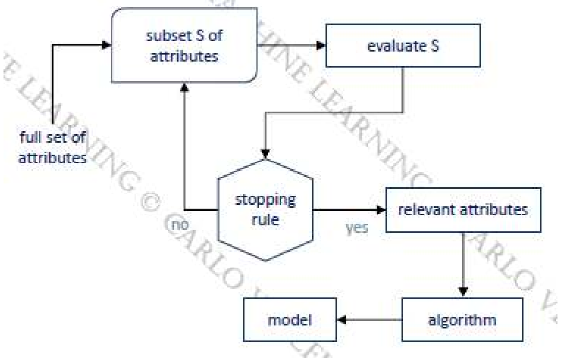
Operate in columns

(numerical) or (cathegories)

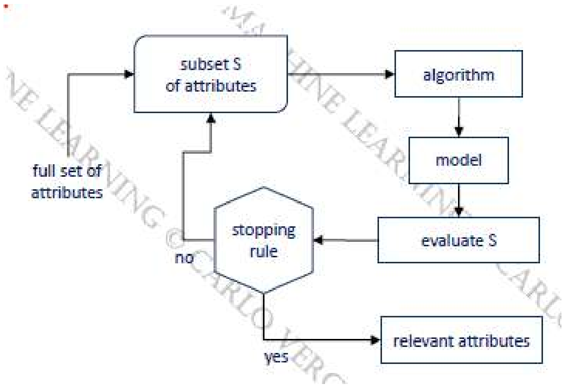
Example of sampling :



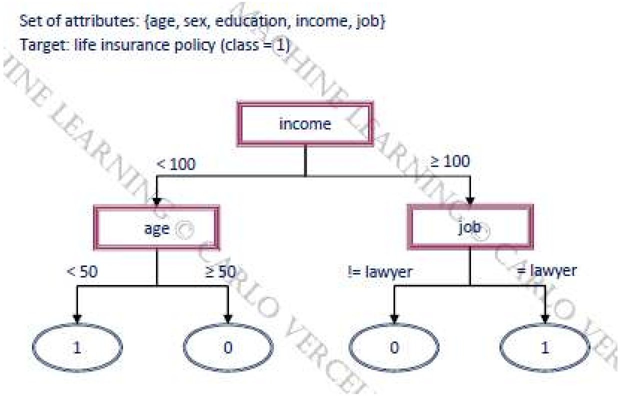
Example of attribute selection : filter methods



Example of attribute selection : wrapper methods



Example of attribute selection : embedded methods classification trees:

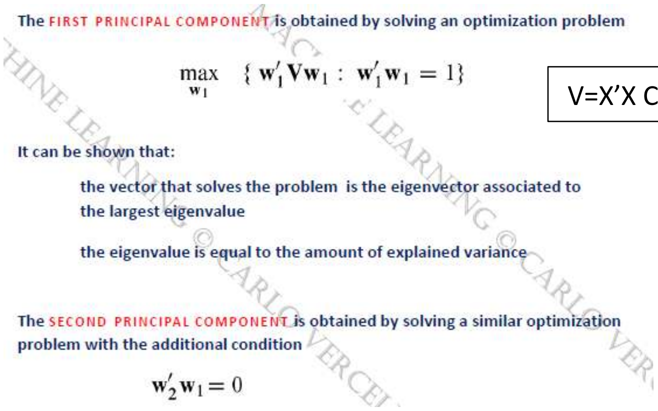


Set of attributes: {age, sex, education, income, job}
Target: life insurance policy (class = 1)

For n attribute => 2^n subset. Greedy search scheme: forward, backward, forward- backward
Data reduction : principal component analysis :

Principal component analysis

- First goal : Changing the axes in order to achieve a better representation of your data
- Second goal: Lower the number of variables



The FIRST PRINCIPAL COMPONENT is obtained by solving an optimization problem

$$\max_{\mathbf{w}_1} \quad \{ \mathbf{w}_1' \mathbf{V} \mathbf{w}_1 : \ \mathbf{w}_1' \mathbf{w}_1 = 1 \}$$

It can be shown that:

the vector that solves the problem is the eigenvector associated to the largest eigenvalue

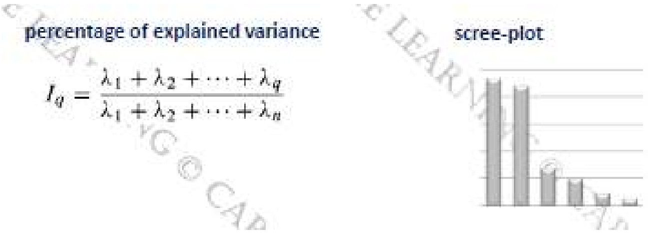
the eigenvalue is equal to the amount of explained variance

The SECOND PRINCIPAL COMPONENT is obtained by solving a similar optimization problem with the additional condition

$$\mathbf{w}_2' \mathbf{w}_1 = 0$$

V=X’X COVARIANCE MATRIX

- How many to consider ?

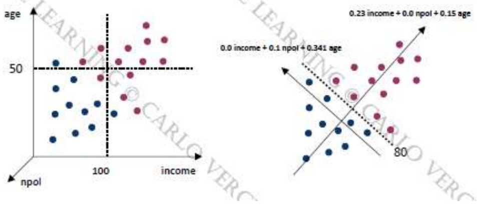


percentage of explained variance

$$I_q = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_q}{\lambda_1 + \lambda_2 + \cdots + \lambda_n}$$

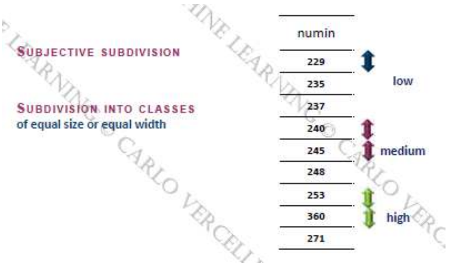
Search the gap

- Example :

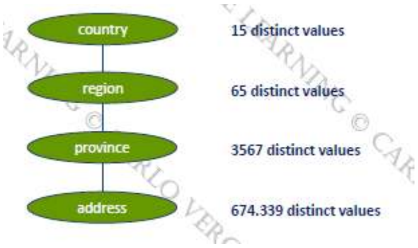


Data discretization

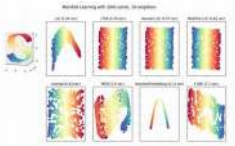
For numerical attributes : Subdivision into classes



Categorical attribute hierarchical discretization



Nonlinear dimensionality reduction – embedding

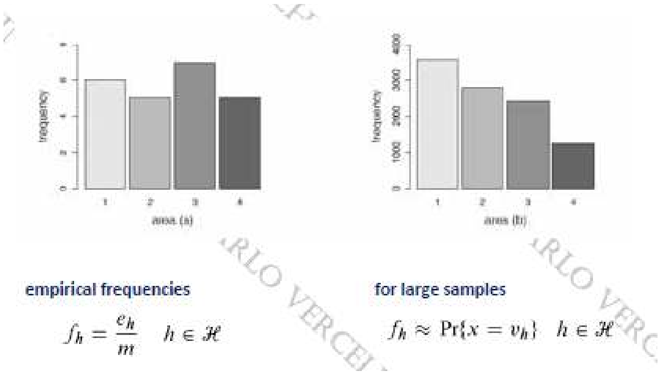


3. Exploratory data Analysis

Single variate and multiple variate analysis :

I. Single variate analysis

a) <u>Graphical analysis of categorical attributes and numerical attributes:</u>



- When using categorical attribute we rely mainly on frequencies (counting the attributes). In the case of large samples we can use frequencies as probability of occurrence.

The other thing we can do with categorical attribute is to estimate how regular are they.

GINI INDEX

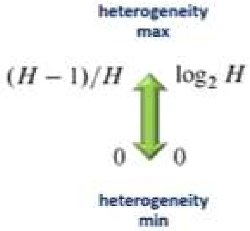
$$G = 1 - \sum_{h=1}^{H} f_h^2$$

ENTROPY INDEX

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

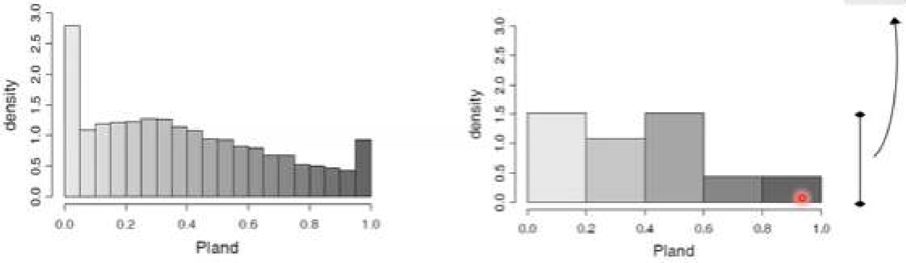
In the max case all the fh are equal to 1/H

In the min all are equal to 0 except 1 it is equal to 1



They are used a lot in the classification trees

- For numerical attributes we can divide our data into small interval to create histograms, the shape of it depends heavily on the width of the interval and the number of observations



$$p_r = \frac{e_r}{m l_r}$$

The probability =
er: nember of observations falling in the interval

m:Number of observations

l : length of interval

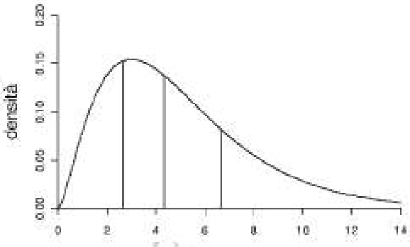
b) Measures of central tendency for numerical attributes

SAMPLE ARITHMETIC MEAN:

$$\bar{\mu} = \frac{x_1 + x_2 + \cdots + x_m}{m} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

MEDIAN: central value of the observations

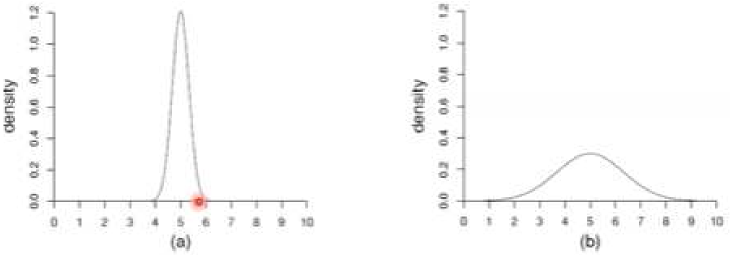
MODE: value that corresponds to the peak of the empirical density curve



The mean is more sensible to outliers

The Mode is not properly defined, when we have more than one peak

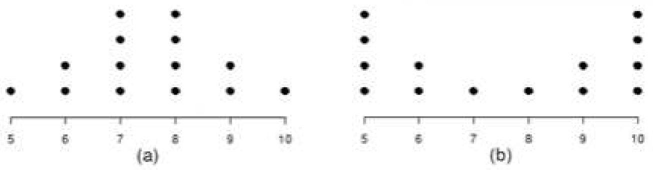
c) Measures of dispersion for numerical attributes



If these two were errors on a process ,the curve (a) is better because values are more concentrated around 5

Range : We can see in this ex the data have the same mean and centrale tendencies and also range

$$x^{range} = x^{max} - x^{min} \qquad x^{max} = \max_i x_i \qquad x^{min} = \min_i x_i$$



Standard deviation :

MEAN ABSOLUTE DEVIATION:

$$\text{MAD} = \frac{1}{m}\sum_{i=1}^{m} |x_i - \bar{\mu}|$$

SAMPLE STANDARD DEVIATION:

$$\bar{\sigma} = \sqrt{\bar{\sigma}^2}$$

SAMPLE VARIANCE:

$$\bar{\sigma}^2 = \frac{1}{m-1}\sum_{i=1}^{m} (x_i - \bar{\mu})^2$$

COEFFICIENT OF VARIATION:

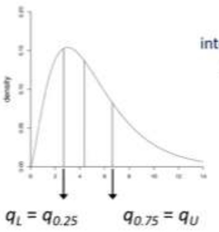
$$\text{CV} = 100\frac{\bar{\sigma}}{\bar{\mu}}$$

CV to evaluate the values of sigma. CV must not be higher than 5%

-

d) Measures of relative location for numerical attributes

p ORDER QUANTILE q_p



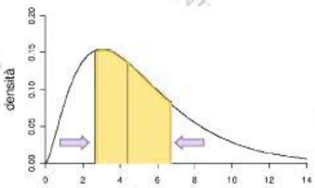
interquartile distance

$D_q = q_U - q_L$

MEAD-MEAN: mean of the values between q_L e q_U

TRIMMED-MEAN: mean of the values between q_p e q_{1-p}

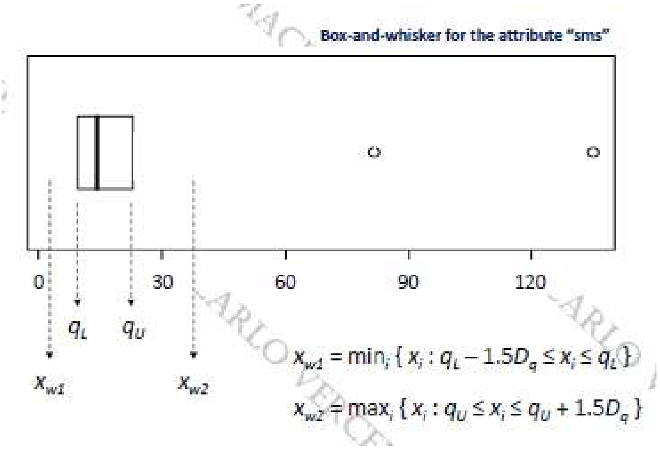
WINSORIZED-MEAN: increase (decrease) values less (greater) than the quantile q_p (q_{1-p})



A typical value of p is 5%

For the **Winsorized-mean**: we substitute all the values less than Qp by Qp and all the values greater than Q1-p by the values of Q1-p

e) Box and whisker

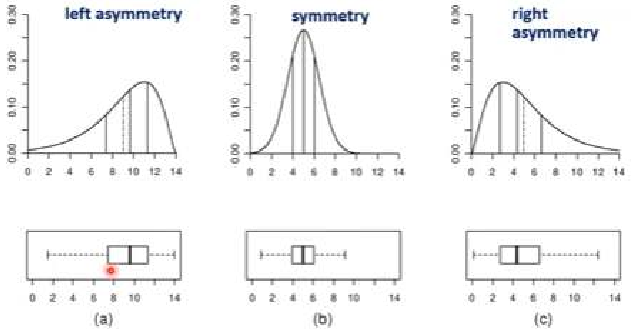


$$x_{w1} = \min_i \{ x_i : q_L - 1.5D_q \le x_i \le q_L \}$$

$$x_{w2} = \max_i \{ x_i : q_U \le x_i \le q_U + 1.5D_q \}$$

The values that are outside the whisker are outliers

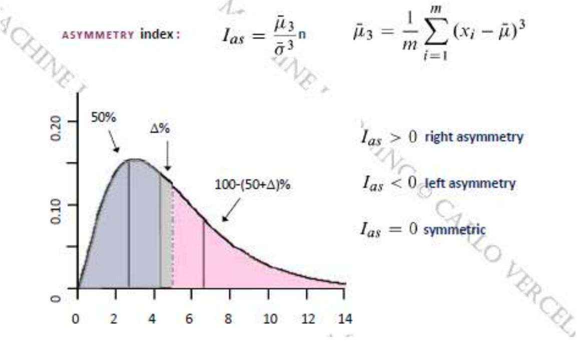
The 1.5 is empirical.



The box and whisker is also used to determine asymmetry in data

f) Analysis of empirical density

- Asymmetry index :



$$I_{as} = \frac{\bar{\mu}_3}{\bar{\sigma}^3} \text{n} \qquad \bar{\mu}_3 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{\mu})^3$$

$I_{as} > 0$ right asymmetry

$I_{as} < 0$ left asymmetry

$I_{as} = 0$ symmetric

- Determining normality :

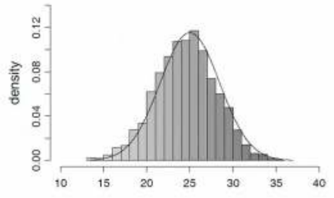
$$I_{\text{curt}} = \frac{\bar{\mu}_4}{\bar{\sigma}^4} - 3 \qquad \bar{\mu}_4 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{\mu})^4.$$

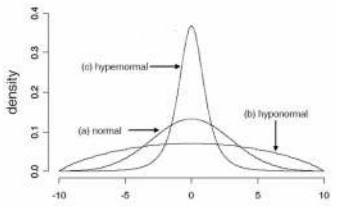
KURTOSIS Index :

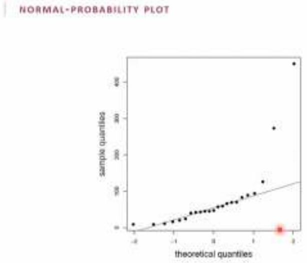
I=0 Normal

I>0 Hypernormal

I<0 Hyponormal





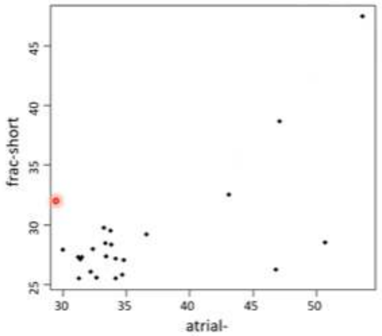


Graphical method to see the normality :

The points should be aligned with the theoretical line of the normal distribution

II. Bivariate analysis

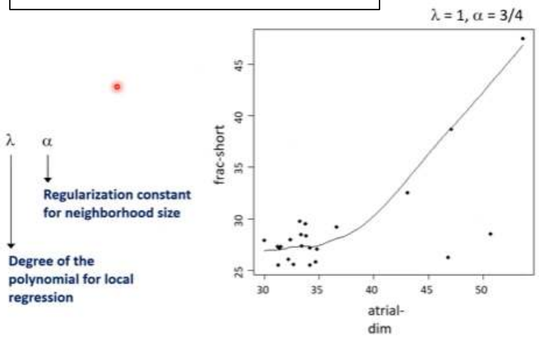
a) Scatterplot

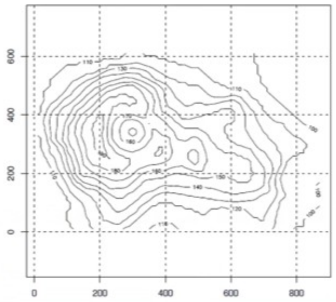


When analyzing scatterplots, we must be very careful from outliers, in the example, the two dimension are not related if we eliminate the outliers

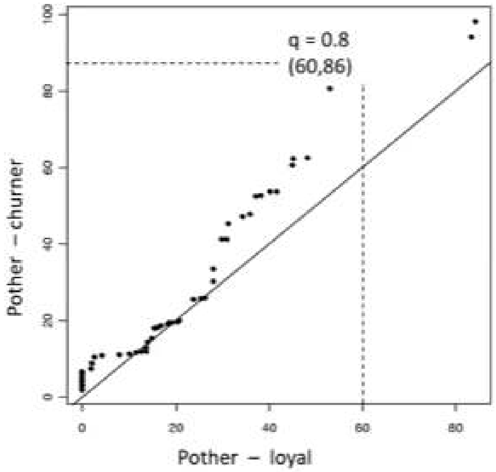
b) Local regression loess

c) Contour lines representing 3



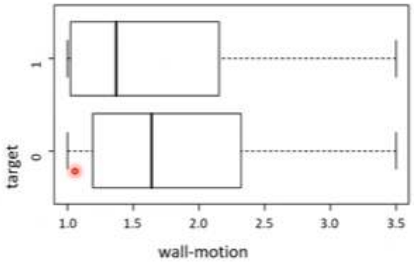


Quantile-Quantile plot (To analyses the densities



d) Quantile-Quantile plot

We use it to analyses the densities, every point has the same quantile for the two variable. For example q=0.8 the value for churner is 86 and the value for the Loyal is 60



e) Box and whisker

As we can see from the example the two attributes have different

SAMPLE COVARIANCE:

$$v_{jk} = \text{cov}(\mathbf{a}_j, \mathbf{a}_k) = \frac{1}{m-2} \sum_{i=1}^{m} (x_{ij} - \bar{\mu}_j)(x_{ik} - \bar{\mu}_k)$$

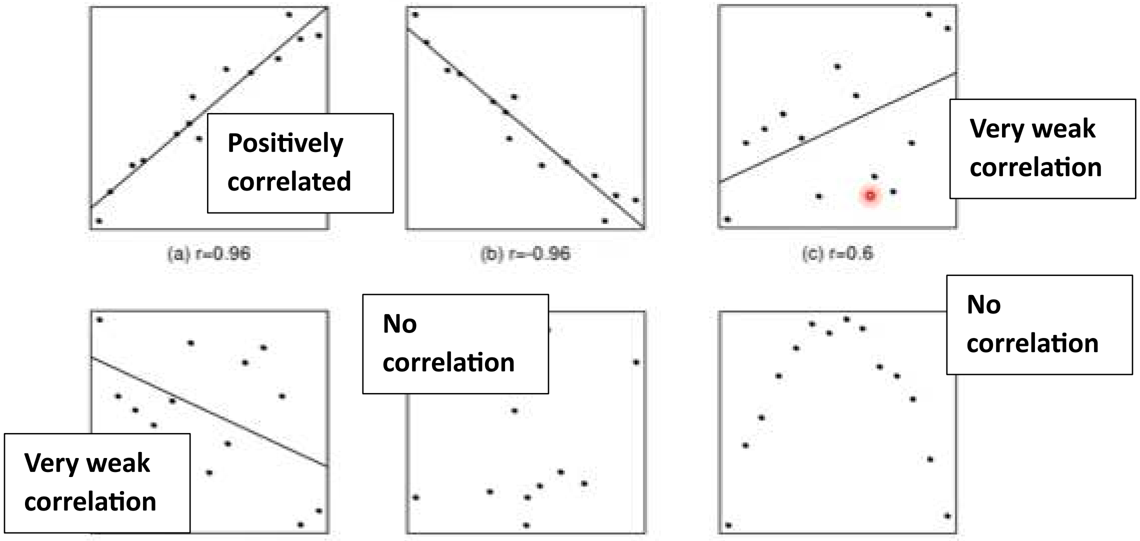
SAMPLE LINEAR CORRELATION :

$$r_{jk} = \text{corr}(\mathbf{a}_j, \mathbf{a}_k) = \frac{v_{jk}}{\bar{\sigma}_j \bar{\sigma}_k} \qquad r_{jk} \in [-1, 1]$$

f) Measure of correlation for numerical attributes

Example of sample linear correlation coefficient :

Negatively correlated



Positively correlated

(a) r=0.96

(b) r=-0.96

Very weak correlation

(c) r=0.6

No correlation

Very weak correlation

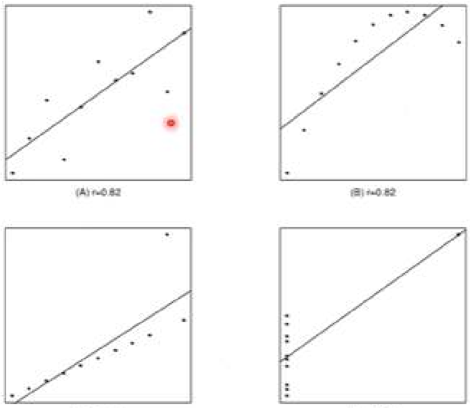
(d) r=-0.6

(e) r=0.0

No correlation

(f) r=0.0

Example of ANSCUMBE DATA SET: same mean same STD, same statistic carrier, same correlation coefficient :



(A) r=0.82

(B) r=0.82

(C) r=0.82

(D) r=0.82

Points are very scattered. So the value of the linear correlation coefficient even if it's high (0.82) it does not mean necessarily there is a correlation. Other things should be taken into consideration

	family		
area	0	1	totale
1	2	4	6 (f_1)
2	4	2	6
3	2	5	7
4	3	3	6
totale	11 (g_1)	14 (g_2)	25

g) Contingency table

We say two variable are independent if the formula is verified example :

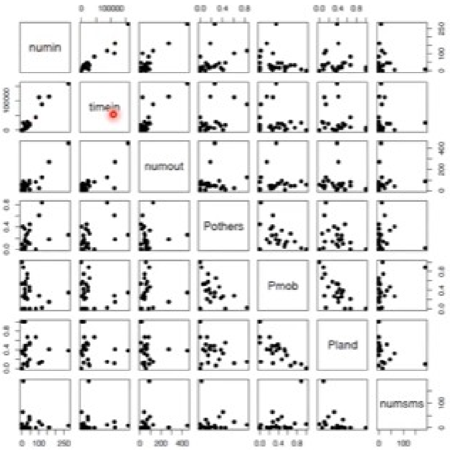
2/11 is diff than 4/14

Two attributes are independent if

$$\frac{t_{r1}}{g_1} = \frac{t_{r2}}{g_2} \qquad r = 1, 2, \ldots, J$$

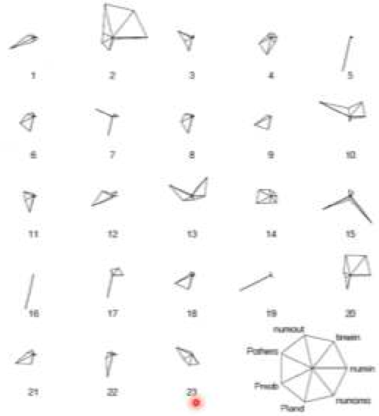
III. Multivariate analysis

a) Scatter plot matrix



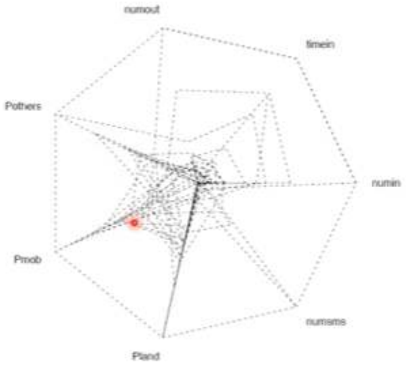
We represent every couple in a plot to see if there is any correlation.

b) Star plot



We say that there is a correlation when there is a recurrent pattern

c) Spider Web chart



Each observation is represented as a trajectory. We can see in the example that the area between Pothers and Pmob is more populated than the others

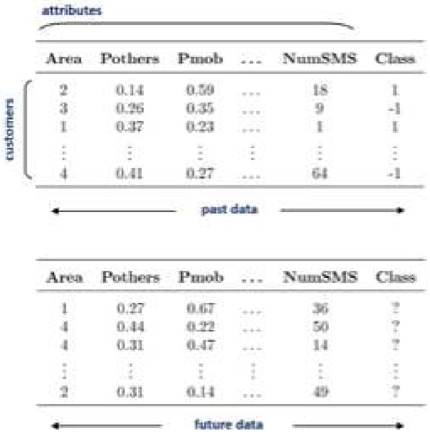
4. Classification

Classification Part 1 :

https://politecnicomilano.webex.com/recordingservice/sites/politecnicomilano/recording/playback/c8fcdd340dc24115affe65a49bdce89d

We are going to see the supervised learning :

A marketing example for telecommunication :



attributes

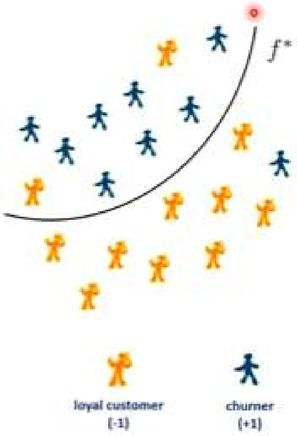
customers

Area	Pothers	Pmob	...	NumSMS	Class
2	0.14	0.59	...	18	1
3	0.26	0.35	...	9	-1
1	0.37	0.23	...	1	1
⋮	⋮	⋮	⋮	⋮	⋮
4	0.41	0.27	...	64	-1

past data

Area	Pothers	Pmob	...	NumSMS	Class
1	0.27	0.67	...	36	?
4	0.44	0.22	...	50	?
4	0.31	0.47	...	14	?
⋮	⋮	⋮	⋮	⋮	⋮
2	0.31	0.14	...	49	?

future data



We want to separate the group of customer into two : loyal and churners based on the data we have.

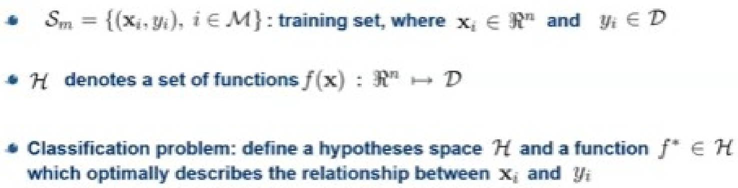
Basically we are going to classify the group in two classes.

In this example we separated the blues from the orange ones by this curve f*. We used majority voting to say that the upper part is for the blue ones. There are 3 misclassification in this example (2 orange in the blues and one blue in the oranges). In the future if a value falls in the blues it will be predicted churner and vis versa

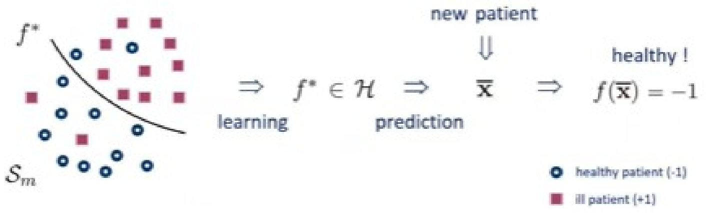
The logic is the following : apply the algorithm for the past data and drive a prediction for the future. We can do this by dividing the past data in two portions, one used for performing the algorithm and the other for tests

a) Classification as a learning task

The classification helps understand the patterns in data and predict future behavior

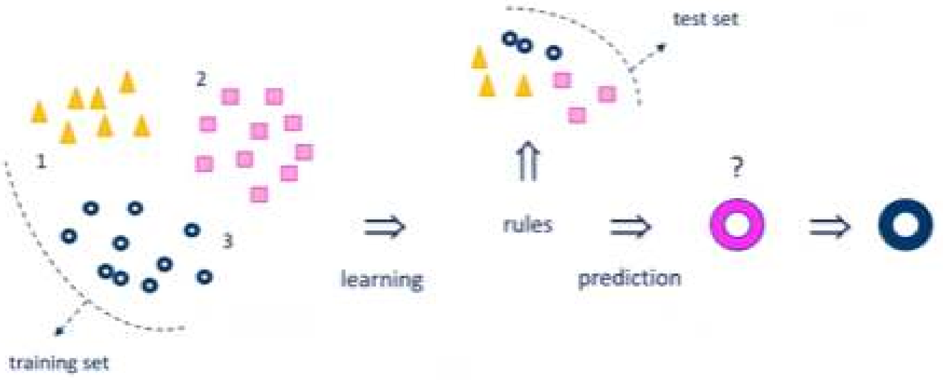


- $\mathcal{S}_m = \{(\mathbf{x}_i, y_i),\ i \in \mathcal{M}\}$: training set, where $\mathbf{x}_i \in \Re^n$ and $y_i \in \mathcal{D}$
- $\mathcal{H}$ denotes a set of functions $f(\mathbf{x}) : \Re^n \mapsto \mathcal{D}$
- Classification problem: define a hypotheses space $\mathcal{H}$ and a function $f^* \in \mathcal{H}$ which optimally describes the relationship between $\mathbf{x}_i$ and y_i

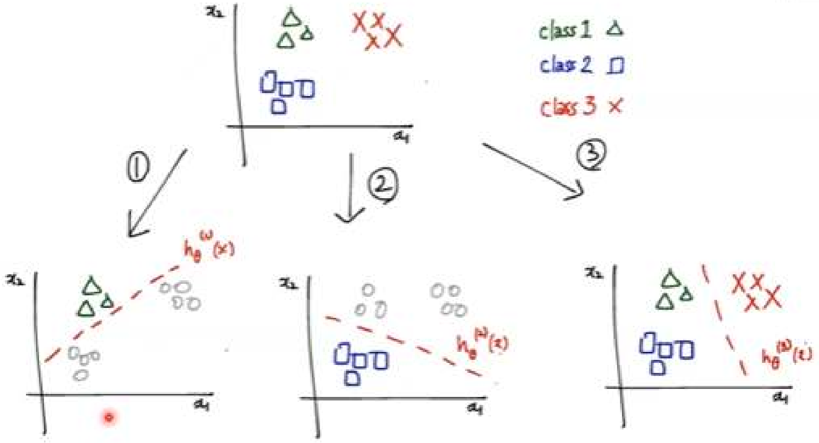


Most of the cases we have more than two categories

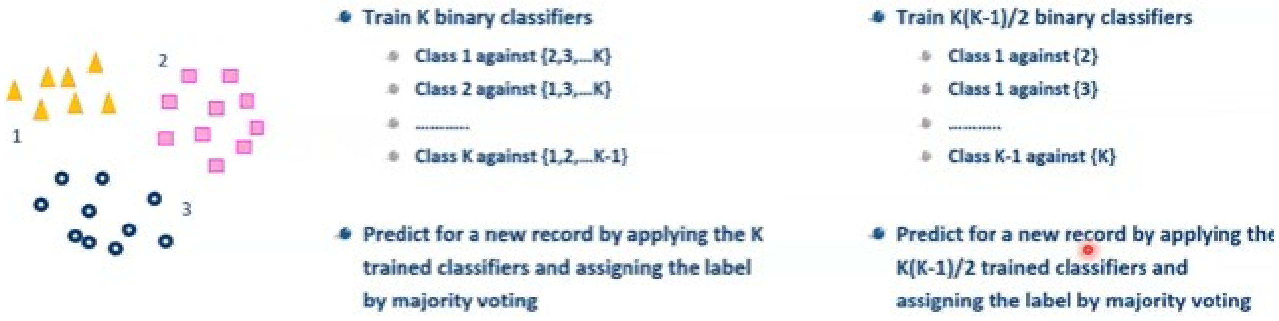
b) Classification models



- **Multi-class Approaches : 1-against-all**



Using this method we go back to binary separation. The price we paid is solving tree problems instead of one. The advantage we solve simple binary problem



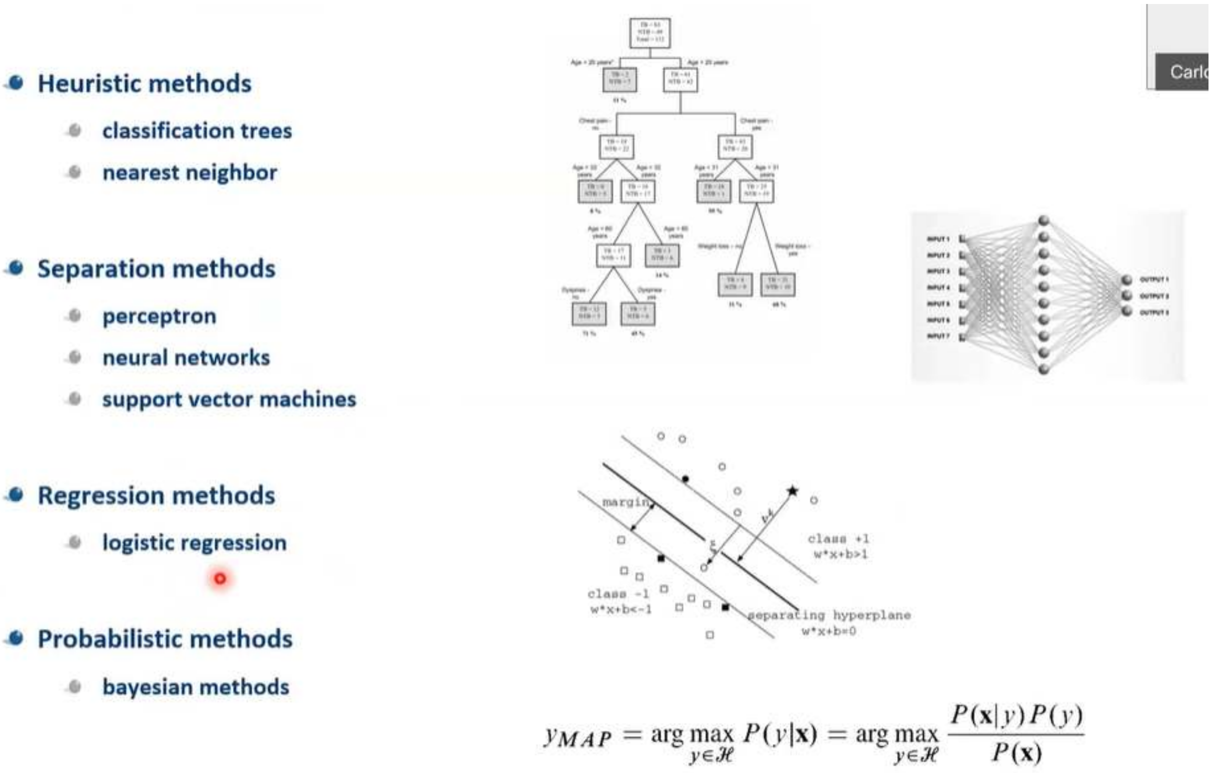
- Train K binary classifiers
  - Class 1 against {2,3,...K}
  - Class 2 against {1,3,...K}
  - ....
  - Class K against {1,2,...K-1}
- Predict for a new record by applying the K trained classifiers and assigning the label by majority voting

- Train K(K-1)/2 binary classifiers
  - Class 1 against {2}
  - Class 1 against {3}
  - ....
  - Class K-1 against {K}
- Predict for a new record by applying the K(K-1)/2 trained classifiers and assigning the label by majority voting

In general, training one against one can lead to better results. But it depends on the pb of course.

To sum up all multivariable problems can be transformed into binary problems.

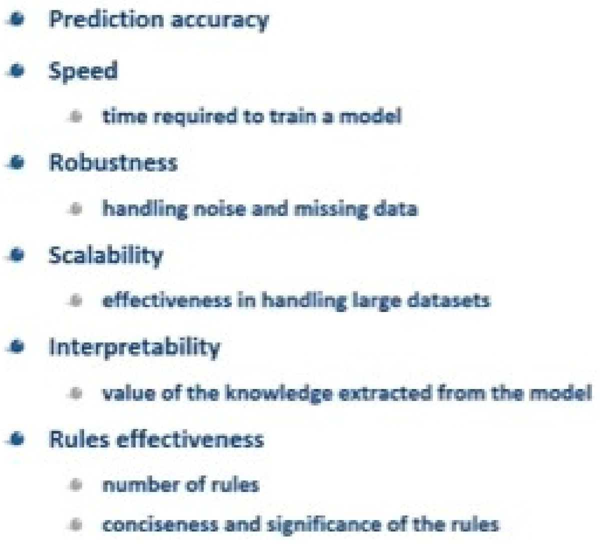
c) Taxonomy of classification methods



- Heuristic methods
  - classification trees
  - nearest neighbor
- Separation methods
  - perceptron
  - neural networks
  - support vector machines
- Regression methods
  - logistic regression
- Probabilistic methods
  - bayesian methods

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

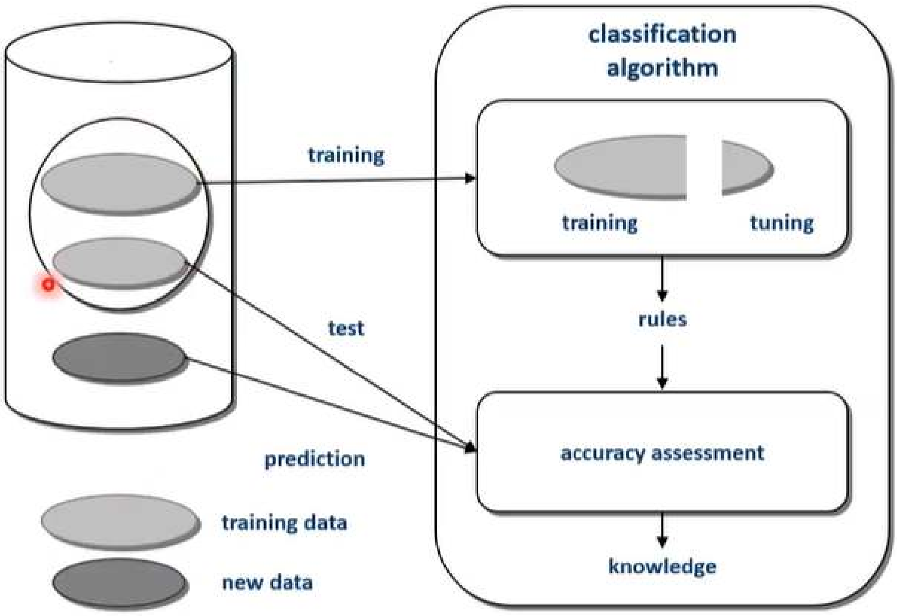
d) Assessing classification methods



- Prediction accuracy
- Speed
  - time required to train a model
- Robustness
  - handling noise and missing data
- Scalability
  - effectiveness in handling large datasets
- Interpretability
  - value of the knowledge extracted from the model
- Rules effectiveness
  - number of rules
  - conciseness and significance of the rules

It’s difficult to say that a method is better than the others that why there are some criteria that can be used to compare each method. More details will be given in the next pages

e) Prediction accuracy



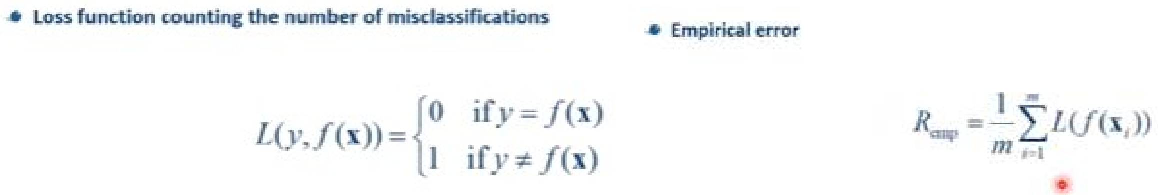
Source: C. Vercellis, Business Intelligence. Data Mining and Optimization for Decision Making, Wiley, 2009.

Starting from past data, we devise it into two parts : training data and test. The training data is given to the algorithm in order to learn the hidden rules. The training set is sometimes devised in the algorithm in two parts training and tuning. The accuracy of the model is determined based on the past test data. Then we assume that this accuracy will be the same for the future data. This means that we assume past data and future data will have the same distribution (it’s an assumption).

We have a set of different algorithms depending on the parameters we chose, once we take the best based on it accuracy in predicting the past test data, we need to evaluate the accuracy of the algorithm using future data that become past data (we made prediction for a month, after a month we collect the real data) If the accuracy is still good we keep our model otherwise we need to train it again.

How ?

We count the numbers of misclassifications :



- Loss function counting the number of misclassifications

$$L(y, f(\mathbf{x})) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}) \\ 1 & \text{if } y \neq f(\mathbf{x}) \end{cases}$$

- Empirical error

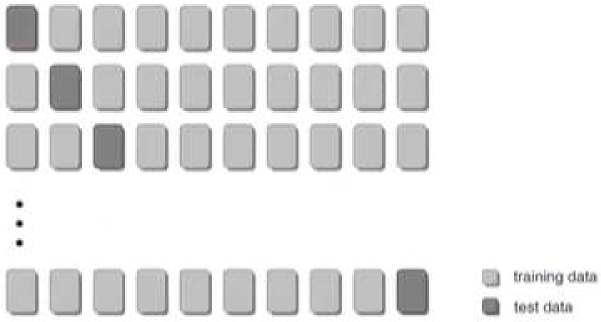
$$R_{emp} = \frac{1}{m}\sum_{i=1}^{m} L(f(\mathbf{x}_i))$$

Empirical error: Remp = Total number of misclassification/total number of observations

Accuracy = 1-Empirical error

How to separate training data and test data ?

- **Holdout method :** Randomly selecting a portion of the data (the most basic) for example selecting 2/3 as training set and 1/3 as test set. The drawback of this method is that the accuracy depend a lot on the random instruction of the test set, given a different test set you will get a new accuracy and this is not robust.
- **Repeated random sampling method:** repeated applications of holdout method and then average the accuracy over all the test sets
- **K-fold Cross-validation :** you divide the data set into “k” group one is used as test the rest for training and you repeat this “k” time. The accuracy is the average (advantage : systematic procedure every observation belongs just once to a test set and k-1 to the training set, less depending from the random instruction.

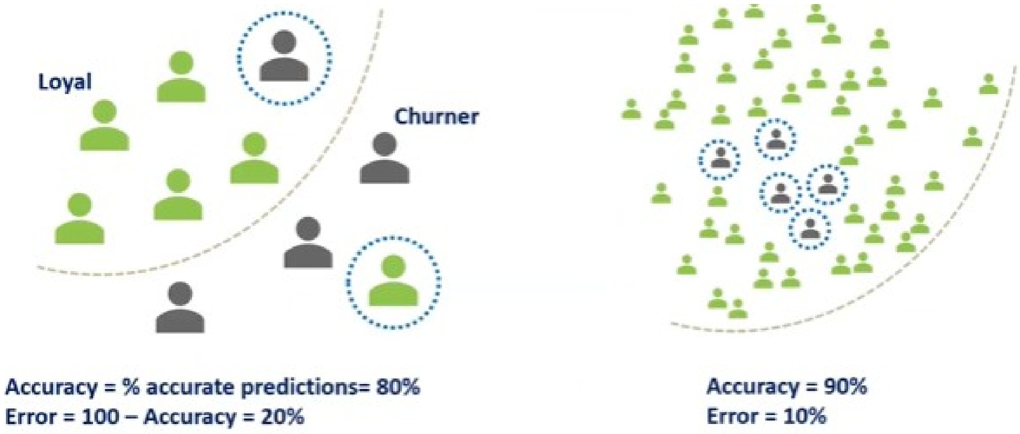


- **Leave one out method :** K-fold with k= m number of observations

Cross-validation : best for small observation 0 – 10000

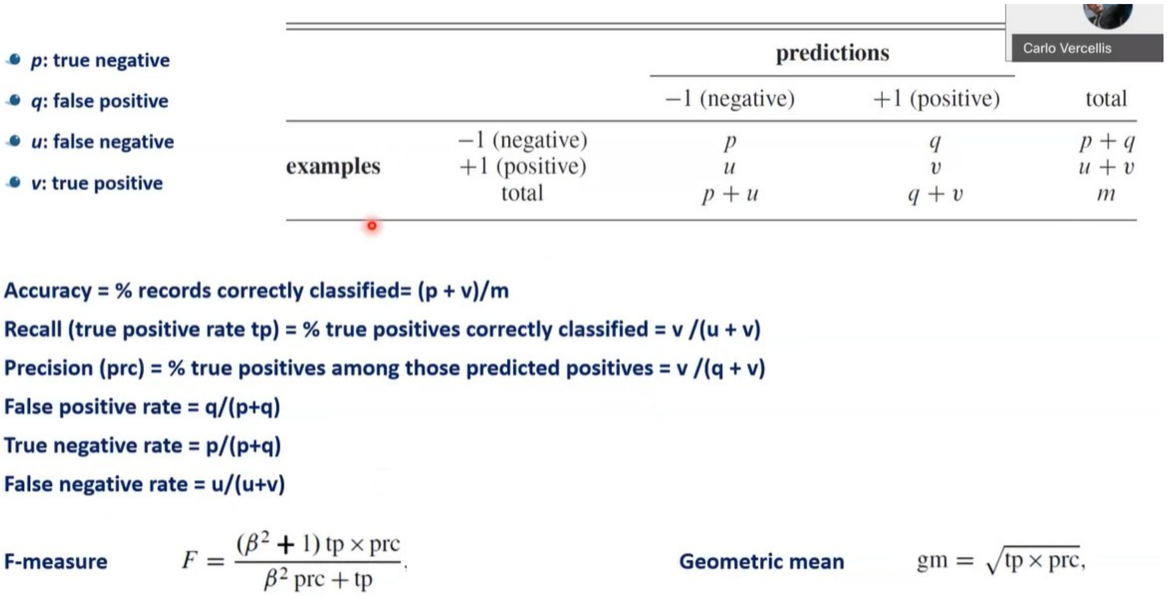
When dealing with more than 100 000 when can use holdout method. If the test set is very large there is no big risk of fluctuation.

f) Model evaluation


Loyal
Churner
Accuracy = % accurate predictions= 80%
Error = 100 – Accuracy = 20%
Accuracy = 90%
Error = 10%

Accuracy is not enough because if we consider the second model with 90% accuracy we will consider all the churners as loyal and we can't do the distinction

g) Confusion matrix



- *p*: true negative
- *q*: false positive
- *u*: false negative
- *v*: true positive

		predictions		
		−1 (negative)	+1 (positive)	total
examples	−1 (negative)	p	q	$p+q$
	+1 (positive)	u	v	$u+v$
	total	$p+u$	$q+v$	m

Accuracy = % records correctly classified= (p + v)/m

Recall (true positive rate tp) = % true positives correctly classified = v /(u + v)

Precision (prc) = % true positives among those predicted positives = v /(q + v)

False positive rate = q/(p+q)

True negative rate = p/(p+q)

False negative rate = u/(u+v)

F-measure $F = \frac{(\beta^2 + 1)\,\mathrm{tp} \times \mathrm{prc}}{\beta^2\,\mathrm{prc} + \mathrm{tp}}.$

Geometric mean $\mathrm{gm} = \sqrt{\mathrm{tp} \times \mathrm{prc}},$

True positive are the once we are interested in (the churners, the ill cells …)

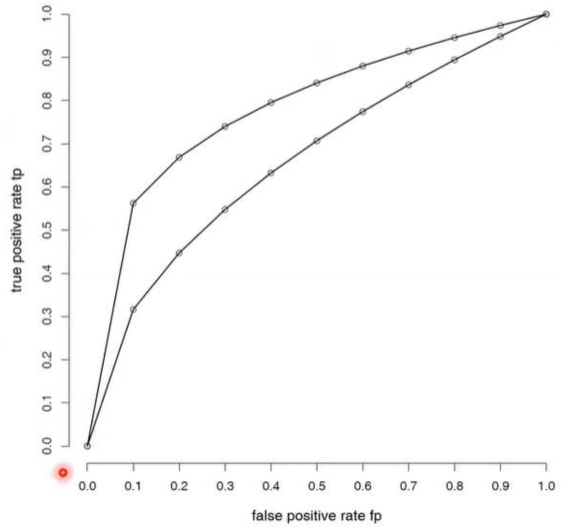
The recall must be close to one, the higher the better

False positive is less costly than false negative (false positive : the value is negative (not sick) and we predicted it as (sick))

In the F-measure formula Beta is a positive parameter, tp : true positive rate recall, prc : precision. It's an harmonic mean between recall and precision. Decreasing or increasing beta means putting more focuss on the precision.

A different way of balancing the two parameter is the Geometric mean

h) ROC curves

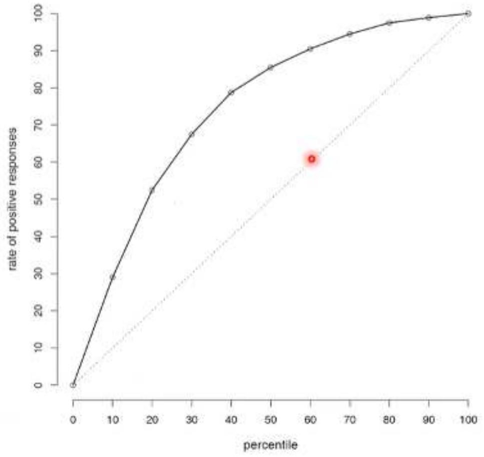


Receiving Operating Characteristic curves

Every observation is associated to a score

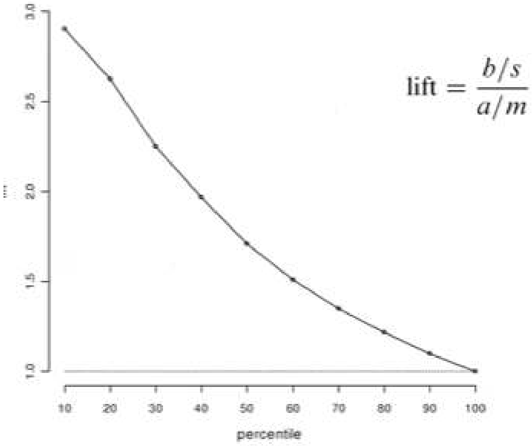
(more research in the subject, prof bouteftaf)

i) Cumulative gain



(more research in the subject, prof bouteftaf)

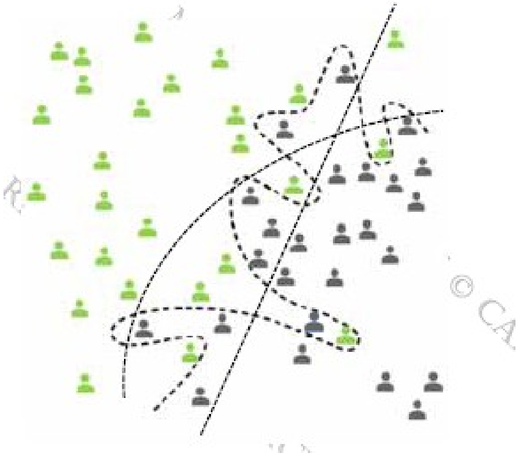
j) lift



$$\text{lift} = \frac{b/s}{a/m}$$

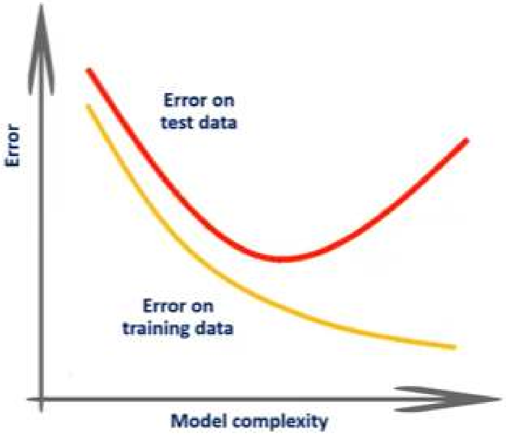
(more research in the subject, prof bouteftaf)

k) Overfitting



High accuracy but poor generalization capability

l) Bias-variance trade-off



Model complexity depends on the type of the model (eg artificial network: number of nods and number of layer, Tree : over ramification)

The work of a machine learner is detecting the point after which the error starts to increase when the model complexity increase

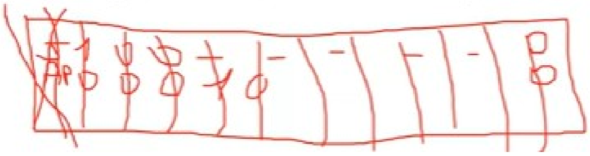
m) Numerical representation of categorical variables

- variable X_j assumes values in the set $V = \{v_1, v_2, \ldots, v_H\}$
- H-1 dummy variables

$$D_{j1}, D_{j2}, \ldots, D_{j,H-1}$$

- For example, to represent the month associated to an observation

$$\{D_{gen}, D_{feb}, D_{mar}, D_{apr}, D_{mag}, D_{giu}, D_{lug}, D_{ago}, D_{set}, D_{ott}, D_{nov}\}.$$



Jan : 1 0 0 0 0 0 0 0 0 0 0 0 0

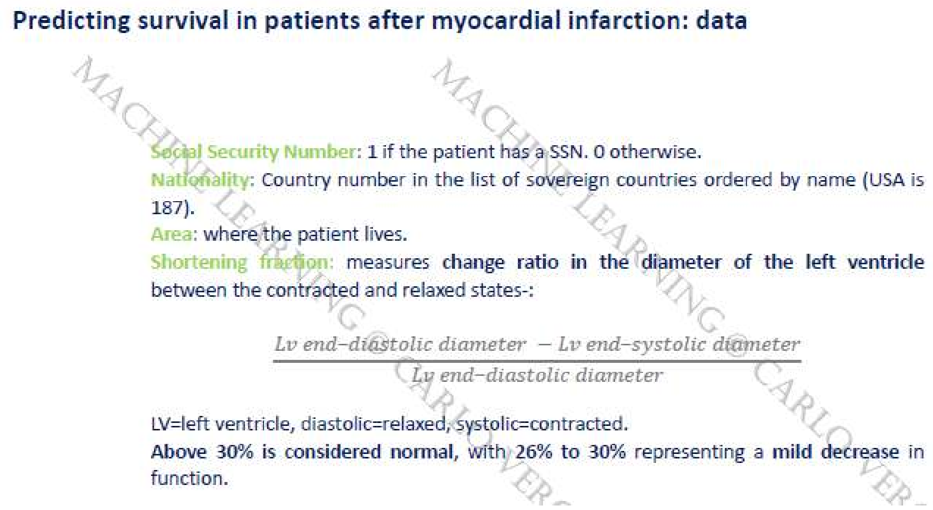
Feb : 0 1 0 0 0 0 0 0 0 0 0 0 0

Mar : 0 0 1 0 0 0 0 0 0 0 0 0 0

We stopped at Dj,H-1 because Dj,H depends on the others

If there is 1 on the first the others are 0

Example of predicting survival in patients after myocardial infarction :



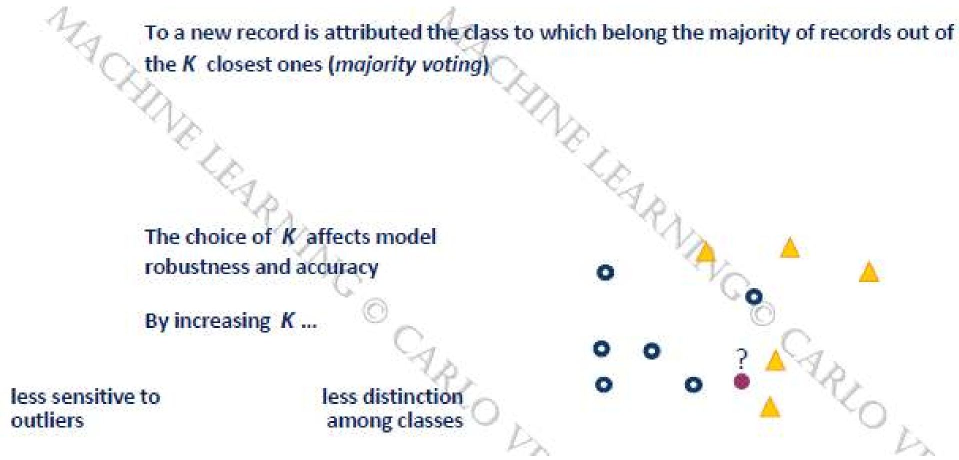
Predicting survival in patients after myocardial infarction: data

Social Security Number: 1 if the patient has a SSN. 0 otherwise.
Nationality: Country number in the list of sovereign countries ordered by name (USA is 187).
Area: where the patient lives.
Shortening fraction: measures change ratio in the diameter of the left ventricle between the contracted and relaxed states-:

$$\frac{Lv\ end\text{–}diastolic\ diameter\ - Lv\ end\text{–}systolic\ diameter}{Lv\ end\text{–}diastolic\ diameter}$$

LV=left ventricle, diastolic=relaxed, systolic=contracted.
Above 30% is considered normal, with 26% to 30% representing a mild decrease in function.

First classification K-nearest neighbor (K-NN)



To a new record is attributed the class to which belong the majority of records out of the K closest ones (*majority voting*)

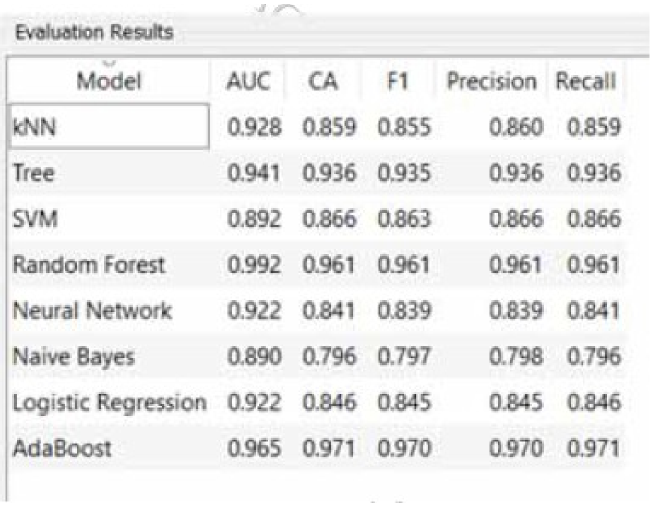
The choice of K affects model robustness and accuracy

By increasing K ...

less sensitive to outliers

less distinction among classes

Cross-validation for different methods on Heart dataset

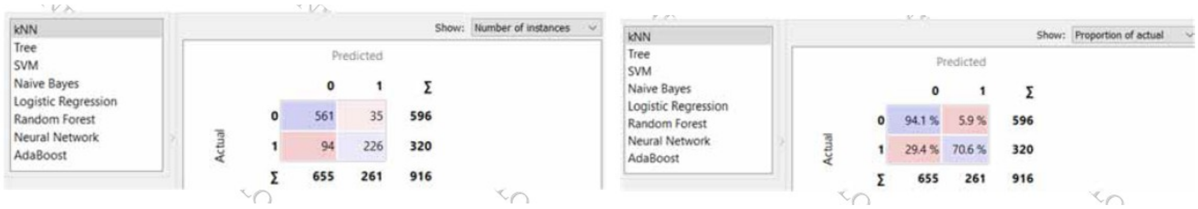


Evaluation Results

Model	AUC	CA	F1	Precision	Recall
kNN	0.928	0.859	0.855	0.860	0.859
Tree	0.941	0.936	0.935	0.936	0.936
SVM	0.892	0.866	0.863	0.866	0.866
Random Forest	0.992	0.961	0.961	0.961	0.961
Neural Network	0.922	0.841	0.839	0.839	0.841
Naive Bayes	0.890	0.796	0.797	0.798	0.796
Logistic Regression	0.922	0.846	0.845	0.845	0.846
AdaBoost	0.965	0.971	0.970	0.970	0.971

We can see from the different parameters that Random Forest give better results than the others

Confusion matrix for different methods on Herat dataset



kNN
Tree
SVM
Naive Bayes
Logistic Regression
Random Forest
Neural Network
AdaBoost

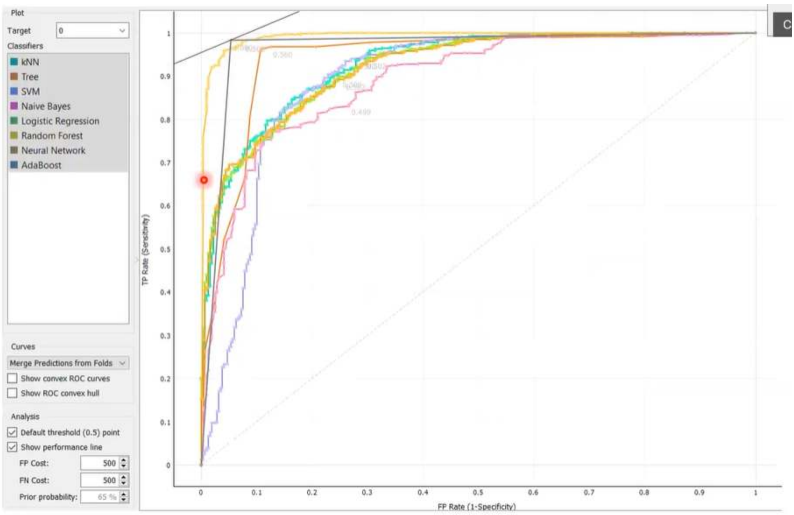
Show: Number of instances

Actual \ Predicted	0	1	Σ
0	561	35	596
1	94	226	320
Σ	655	261	916

Show: Proportion of actual

Actual \ Predicted	0	1	Σ
0	94.1 %	5.9 %	596
1	29.4 %	70.6 %	320
Σ	655	261	916

ROC curve of different methods used on the Heart Dataset



Plot
Target: 0
Classifiers: kNN, Tree, SVM, Naive Bayes, Logistic Regression, Random Forest, Neural Network, AdaBoost
Curves: Merge Predictions from Folds
☐ Show convex ROC curves
☐ Show ROC convex hull
Analysis
☑ Default threshold (0.5) point
☑ Show performance line
FP Cost: 500
FN Cost: 500
Prior probability: 65 %

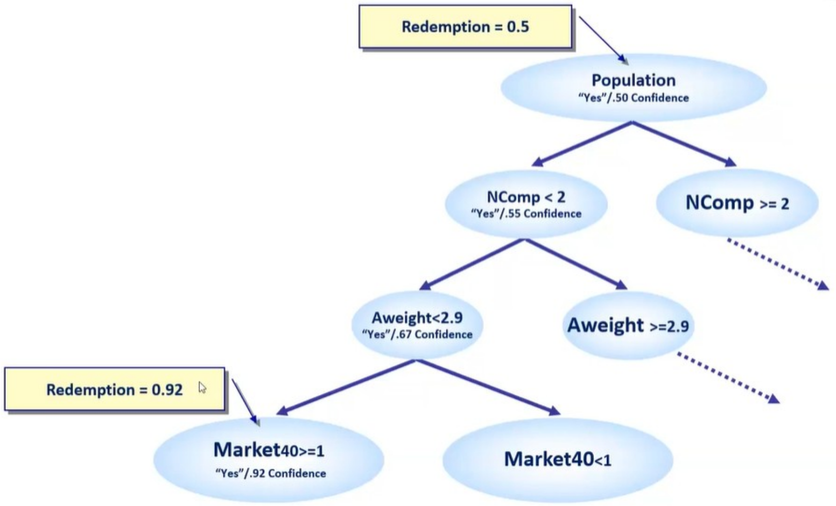
The more area there is between the curve and the straight line the best the classification method. In our example we can see that the random forest is far away from most of the other curves

Classification Part 2 :

https://politecnicomilano.webex.com/recordingservice/sites/politecnicomilano/recording/playback/c826eecf5c5740b289fbae48155392b6

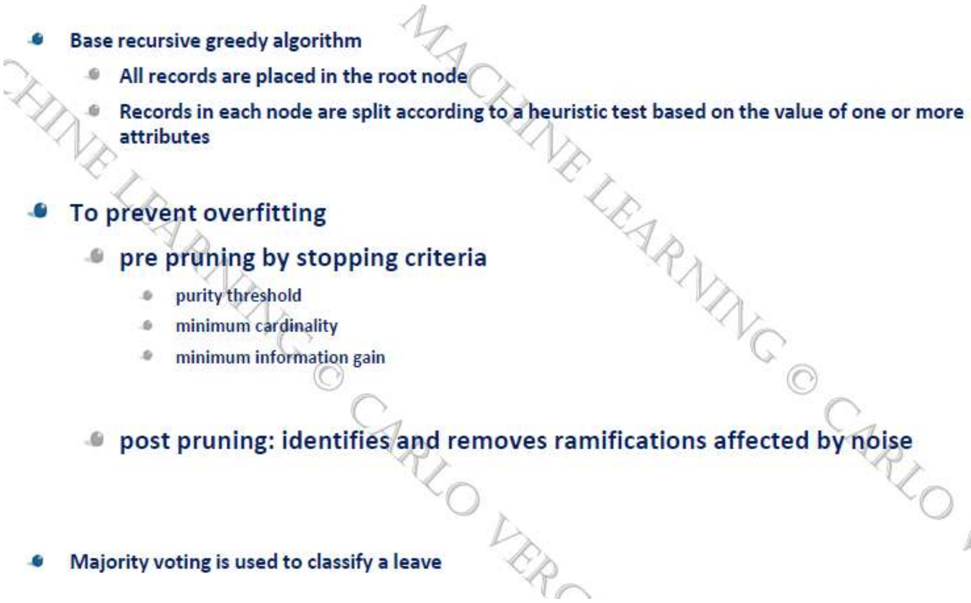
Classification tree

a) Example :



Maximum heterogeneity corresponds to max Gini or Entropy index. So we start with maximum heterogeneity (or an index expressing confusion) and we try to decrease it, they analyze all the variables, all the splitting of the data set, and they select the variable and splitting point that correspond to the best ordering that decreases homogeinity

b) Characteristics



- Base recursive greedy algorithm
  - All records are placed in the root node
  - Records in each node are split according to a heuristic test based on the value of one or more attributes
- To prevent overfitting
  - pre pruning by stopping criteria
    - purity threshold
    - minimum cardinality
    - minimum information gain
  - post pruning: identifies and removes ramifications affected by noise
- Majority voting is used to classify a leave

The algorithm is recursive : it repeat the same operations.

The split does not guarantee that it's the best.

Pruning : cutting branches in botanic science

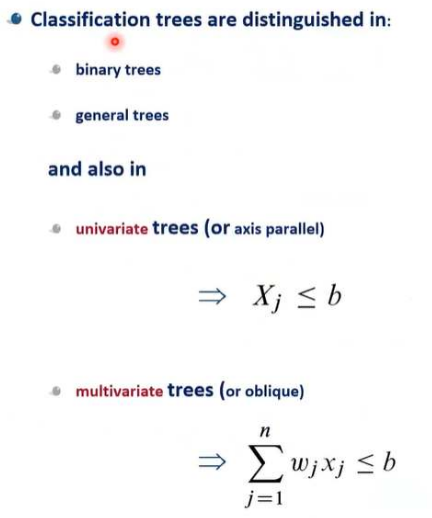
Purity threshold : hyper parameter, a limit that we fix after which we stop splitting (last example: 90%)

Minimum cardinality : minimum number of observations to have in a split

Minimum information gain : is the gain in the percentage index (GINI , entropy …) which translate into a gain in the information (last example in the first split we passed from .50 to 0.55

For post pruning : the algorithm build the classification based on the data of training that is split into two (training and tuning(tuning is different than the test data, it's kept internally))**.** Once the data of training if used, the algorithm uses the tuning data as a self-test and then evaluate if it's necessary to keep the branch or remove it (easy to say very difficult to implement a lot of coding in the algorithm)

c) Properties of classification trees



- Classification trees are distinguished in:
  - binary trees
  - general trees

  and also in
  - univariate trees (or axis parallel)

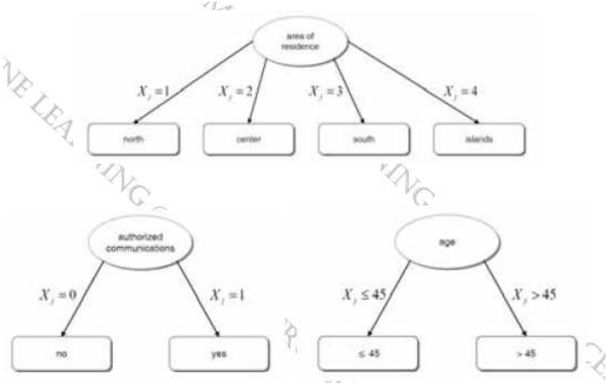
    $$\Rightarrow \quad X_j \leq b$$

  - multivariate trees (or oblique)

    $$\Rightarrow \sum_{j=1}^{n} w_j x_j \leq b$$

Binary trees : 2 splits

General trees : more than 2

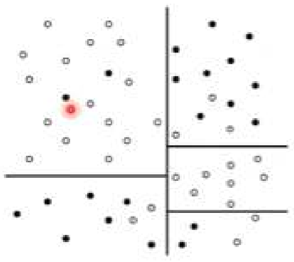


Univariate tree : the splitting rule is based on a single attribute at a time (also called axis parallel) advantage : easy to interpreted ex if sex gender is Male, in the age is under 30

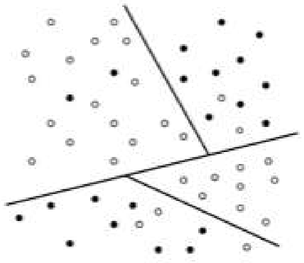
Multivariate tree : the splitting rule is based on a multi attribute at a time (also called oblique)

Advantage : more powerful

Example :



classification by an axis parallel tree

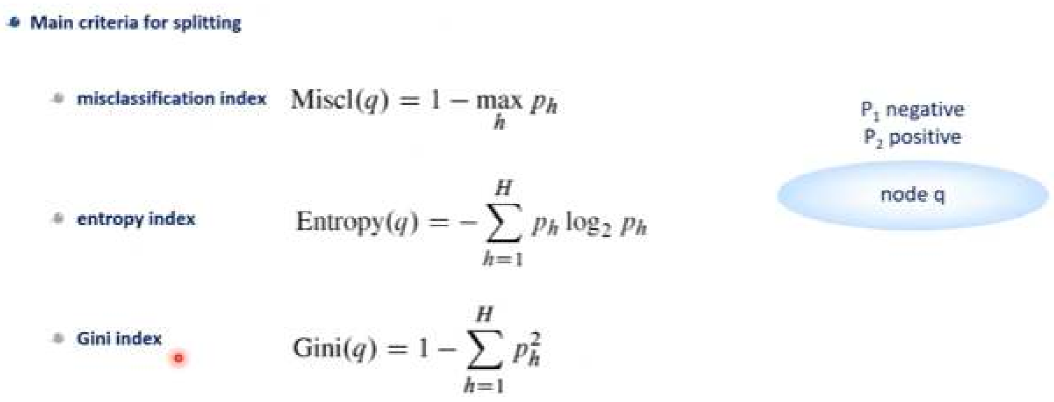


classification by an oblique tree

We obtained the axis parallel classification by 4 nods (4 lines)

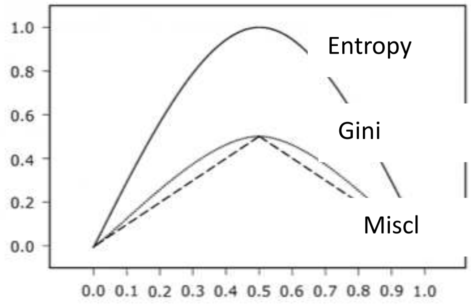
We obtained the oblique classification by 3 nods (3 lines) we have less errors but it’s less interpretable

d) Criteria used in classification trees

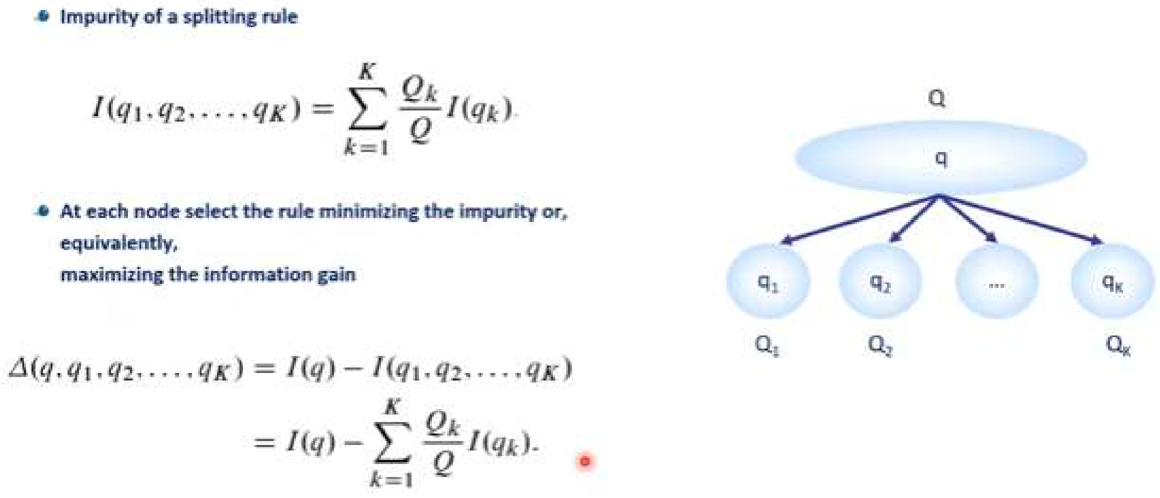


- Main criteria for splitting
  - misclassification index $\mathrm{Miscl}(q) = 1 - \max_h p_h$
  - entropy index $\mathrm{Entropy}(q) = -\sum_{h=1}^{H} p_h \log_2 p_h$
  - Gini index $\mathrm{Gini}(q) = 1 - \sum_{h=1}^{H} p_h^2$

The three of them are heterogeneity indexes, below we can see there graphical representation :



Using one or the other depends on the imperial approach, we need to test each one of them and see the results



- Impurity of a splitting rule

$$I(q_1, q_2, \dots, q_K) = \sum_{k=1}^{K} \frac{Q_k}{Q} I(q_k)$$

- At each node select the rule minimizing the impurity or, equivalently, maximizing the information gain

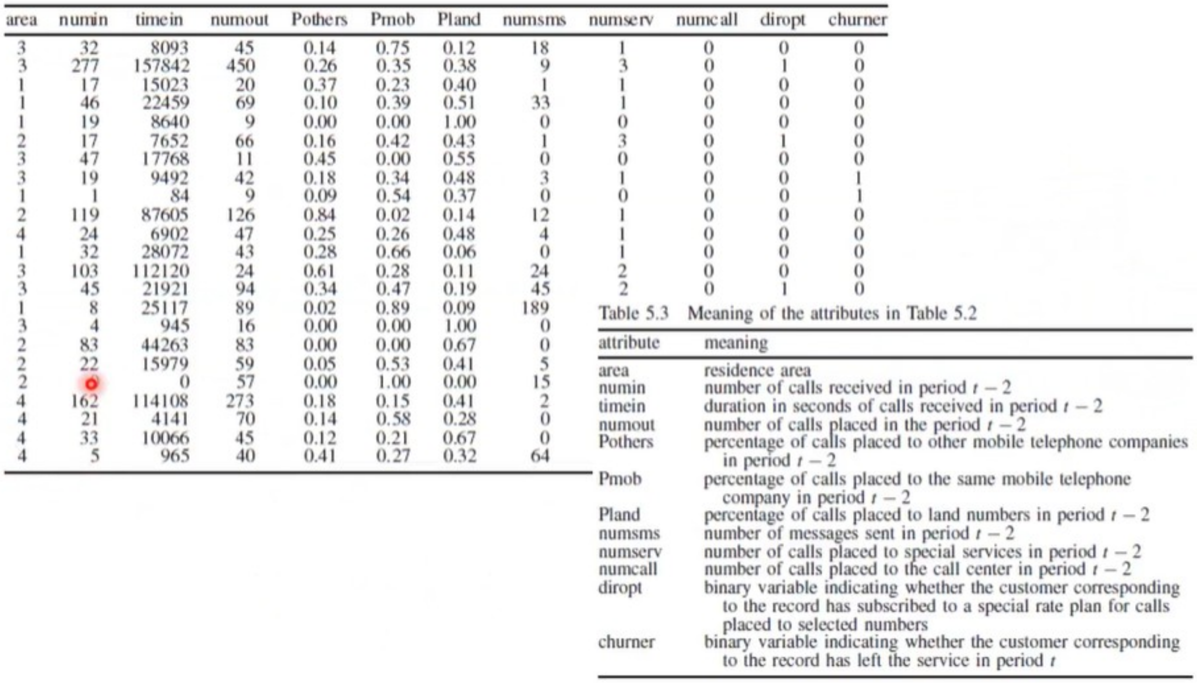
$$\Delta(q, q_1, q_2, \dots, q_K) = I(q) - I(q_1, q_2, \dots, q_K) = I(q) - \sum_{k=1}^{K} \frac{Q_k}{Q} I(q_k).$$

“I” is the impurity (or heterogeneity) index that can be either Gini, entropy, miscal.. q are the nods and Q are the number of attributes in each nod.

We calculate the impurity index by the weighted average of all the impurities of the nods

We calculate the information gain by subtraction the Impurity of the Father I(q) and the impurity of a splitting rule

Example :

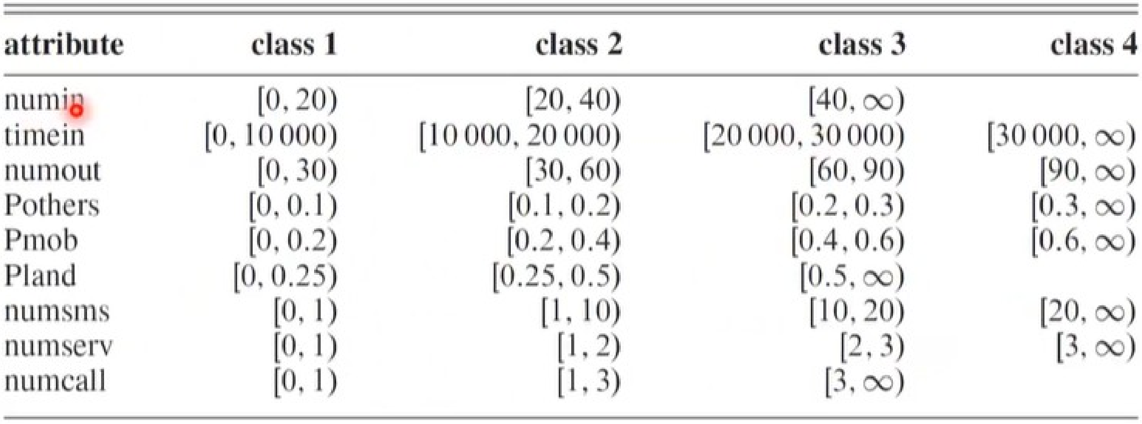


area	numin	timein	numout	Pothers	Pmob	Pland	numsms	numserv	numcall	diropt	churner
3	32	8093	45	0.14	0.75	0.12	18	1	0	0	0
3	277	157842	450	0.26	0.35	0.38	9	3	0	1	0
1	17	15023	20	0.37	0.23	0.40	1	1	0	0	0
1	46	22459	69	0.10	0.39	0.51	33	1	0	0	0
1	19	8640	9	0.00	0.00	1.00	0	0	0	0	0
2	17	7652	66	0.16	0.42	0.43	1	3	0	1	0
3	47	17768	11	0.45	0.00	0.55	0	0	0	0	0
3	19	9492	42	0.18	0.34	0.48	3	1	0	0	1
1	1	84	9	0.09	0.54	0.37	0	0	0	0	1
2	119	87605	126	0.84	0.02	0.14	12	1	0	0	0
4	24	6902	47	0.25	0.26	0.48	4	1	0	0	0
1	32	28072	43	0.28	0.66	0.06	0	1	0	0	0
3	103	112120	24	0.61	0.28	0.11	24	2	0	0	0
3	45	21921	94	0.34	0.47	0.19	45	2	0	1	0
1	8	25117	89	0.02	0.89	0.09	189				
3	4	945	16	0.00	0.00	1.00	0				
2	83	44263	83	0.00	0.00	0.67	0				
2	22	15979	59	0.05	0.53	0.41	5				
2	0	0	57	0.00	1.00	0.00	15				
4	162	114108	273	0.18	0.15	0.41	2				
4	21	4141	70	0.14	0.58	0.28	0				
4	33	10066	45	0.12	0.21	0.67	0				
4	5	965	40	0.41	0.27	0.32	64				

Table 5.3 Meaning of the attributes in Table 5.2

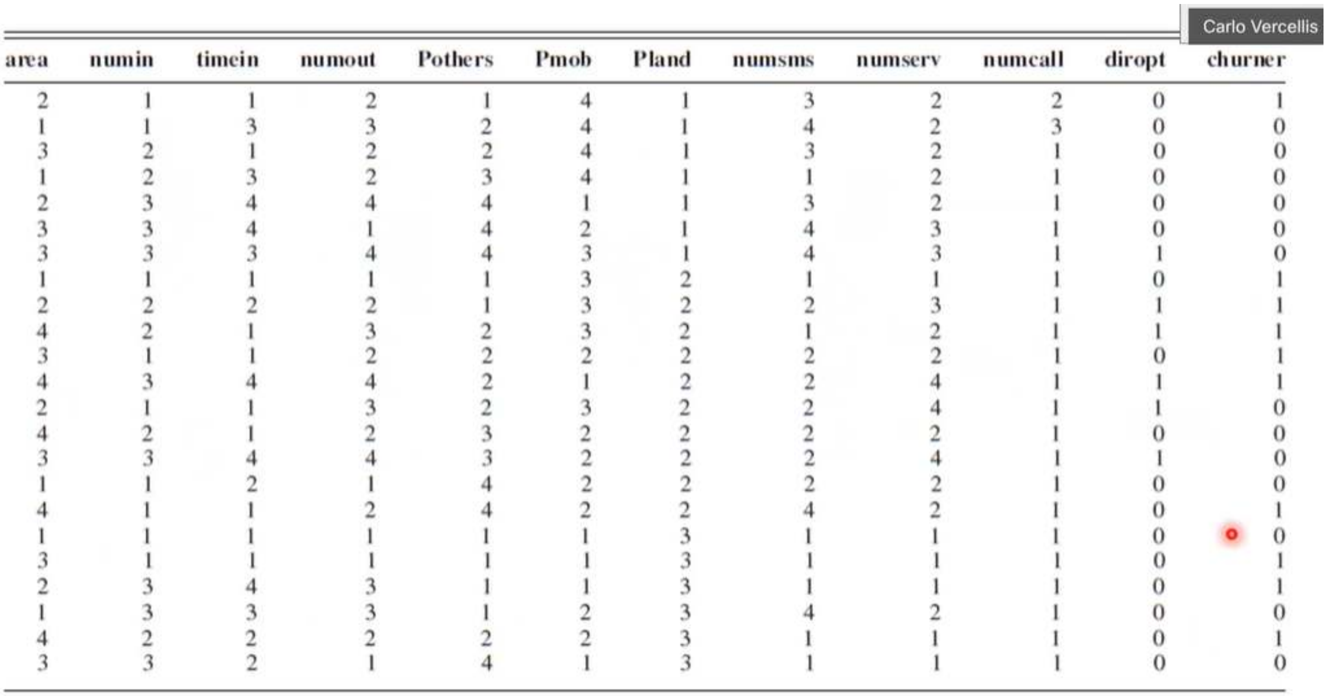
attribute	meaning
area	residence area
numin	number of calls received in period $t-2$
timein	duration in seconds of calls received in period $t-2$
numout	number of calls placed in the period $t-2$
Pothers	percentage of calls placed to other mobile telephone companies in period $t-2$
Pmob	percentage of calls placed to the same mobile telephone company in period $t-2$
Pland	percentage of calls placed to land numbers in period $t-2$
numsms	number of messages sent in period $t-2$
numserv	number of calls placed to special services in period $t-2$
numcall	number of calls placed to the call center in period $t-2$
diropt	binary variable indicating whether the customer corresponding to the record has subscribed to a special rate plan for calls placed to selected numbers
churner	binary variable indicating whether the customer corresponding to the record has left the service in period t

The numerical data is going to be transformed to categorical data by gathering them (bin them)



attribute	class 1	class 2	class 3	class 4
numin	$[0, 20)$	$[20, 40)$	$[40, \infty)$	
timein	$[0, 10\,000)$	$[10\,000, 20\,000)$	$[20\,000, 30\,000)$	$[30\,000, \infty)$
numout	$[0, 30)$	$[30, 60)$	$[60, 90)$	$[90, \infty)$
Pothers	$[0, 0.1)$	$[0.1, 0.2)$	$[0.2, 0.3)$	$[0.3, \infty)$
Pmob	$[0, 0.2)$	$[0.2, 0.4)$	$[0.4, 0.6)$	$[0.6, \infty)$
Pland	$[0, 0.25)$	$[0.25, 0.5)$	$[0.5, \infty)$	
numsms	$[0, 1)$	$[1, 10)$	$[10, 20)$	$[20, \infty)$
numserv	$[0, 1)$	$[1, 2)$	$[2, 3)$	$[3, \infty)$
numcall	$[0, 1)$	$[1, 3)$	$[3, \infty)$	

In the end this is the data set : (this step is not necessary for classification trees)



Carlo Vercellis

area	numin	timein	numout	Pothers	Pmob	Pland	numsms	numserv	numcall	diropt	churner
2	1	1	2	1	4	1	3	2	2	0	1
1	1	3	3	2	4	1	4	2	3	0	0
3	2	1	2	2	4	1	3	2	1	0	0
1	2	3	2	3	4	1	1	2	1	0	0
2	3	4	4	4	1	1	3	2	1	0	0
3	3	4	1	4	2	1	4	3	1	0	0
3	3	3	4	4	3	1	4	3	1	1	0
1	1	1	1	1	3	2	1	1	1	0	1
2	2	2	2	1	3	2	2	3	1	1	1
4	2	1	3	2	3	2	1	2	1	1	1
3	1	1	2	2	2	2	2	2	1	0	1
4	3	4	4	2	1	2	2	4	1	1	1
2	1	1	3	2	3	2	2	4	1	1	0
4	2	1	2	3	2	2	2	2	1	0	0
3	3	4	4	3	2	2	2	4	1	1	0
1	1	2	1	4	2	2	2	2	1	0	0
4	1	1	2	4	2	2	4	2	1	0	1
1	1	1	1	1	1	3	1	1	1	0	0
3	1	1	1	1	1	3	1	1	1	0	1
2	3	4	3	1	1	3	1	1	1	0	1
1	3	3	3	1	2	3	4	2	1	0	0
4	2	2	2	2	2	3	1	1	1	0	1
3	3	2	1	4	1	3	1	1	1	0	0

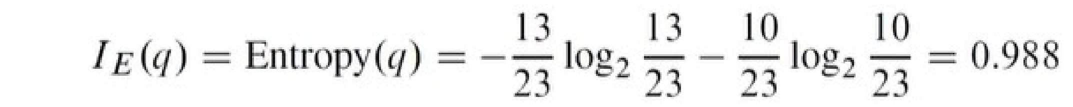
We have 23 observations, 13 positive (13 cherner) and 10 negative (loyal)

We have 4 areas.

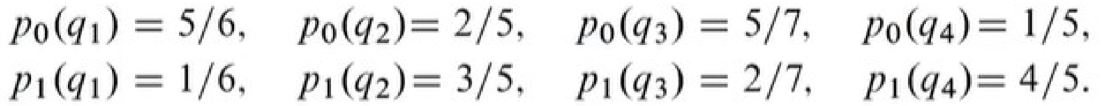
6 observations in area 1 , 5 are loyal and 1 is a churner

5 observations in area 2, 2 are loyal and 3 are churners

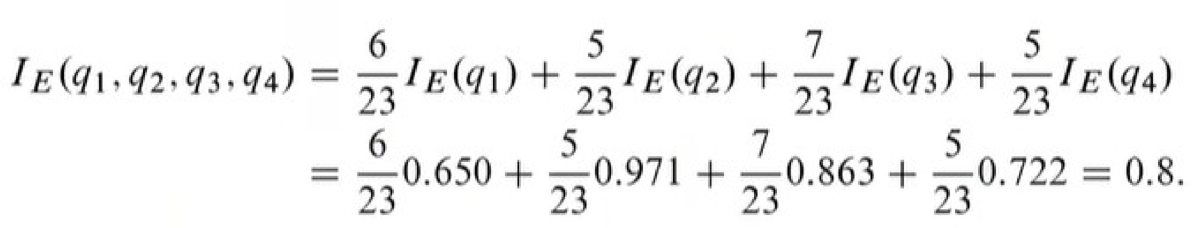
We start by calculating the entropy :



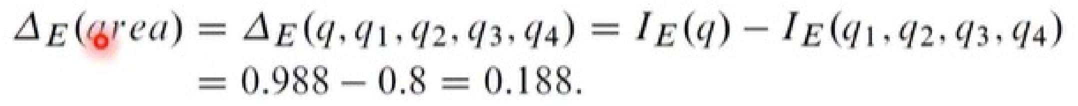
$$I_E(q) = \text{Entropy}(q) = -\frac{13}{23}\log_2\frac{13}{23} - \frac{10}{23}\log_2\frac{10}{23} = 0.988$$



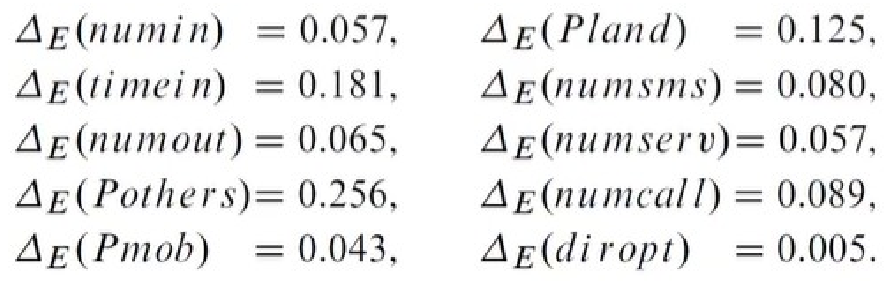
$$\begin{aligned} &p_0(q_1) = 5/6, \quad p_0(q_2) = 2/5, \quad p_0(q_3) = 5/7, \quad p_0(q_4) = 1/5,\\ &p_1(q_1) = 1/6, \quad p_1(q_2) = 3/5, \quad p_1(q_3) = 2/7, \quad p_1(q_4) = 4/5. \end{aligned}$$



$$\begin{aligned} I_E(q_1, q_2, q_3, q_4) &= \frac{6}{23}I_E(q_1) + \frac{5}{23}I_E(q_2) + \frac{7}{23}I_E(q_3) + \frac{5}{23}I_E(q_4)\\ &= \frac{6}{23}0.650 + \frac{5}{23}0.971 + \frac{7}{23}0.863 + \frac{5}{23}0.722 = 0.8. \end{aligned}$$



$$\begin{aligned} \Delta_E(area) &= \Delta_E(q, q_1, q_2, q_3, q_4) = I_E(q) - I_E(q_1, q_2, q_3, q_4)\\ &= 0.988 - 0.8 = 0.188. \end{aligned}$$



$$\begin{aligned} &\Delta_E(numin) = 0.057, && \Delta_E(Pland) = 0.125,\\ &\Delta_E(timein) = 0.181, && \Delta_E(numsms) = 0.080,\\ &\Delta_E(numout) = 0.065, && \Delta_E(numserv) = 0.057,\\ &\Delta_E(Pothers) = 0.256, && \Delta_E(numcall) = 0.089,\\ &\Delta_E(Pmob) = 0.043, && \Delta_E(diropt) = 0.005. \end{aligned}$$

The same calculating done to the area is done to the other attributes, the max information gain is Pothers : percentage toward other operators

With the Gini index we can have the same calculation :

$$I_G(q) = \text{Gini}(q) = 1 - \left(\frac{13}{23}\right)^2 - \left(\frac{10}{23}\right)^2 = 0.491.$$

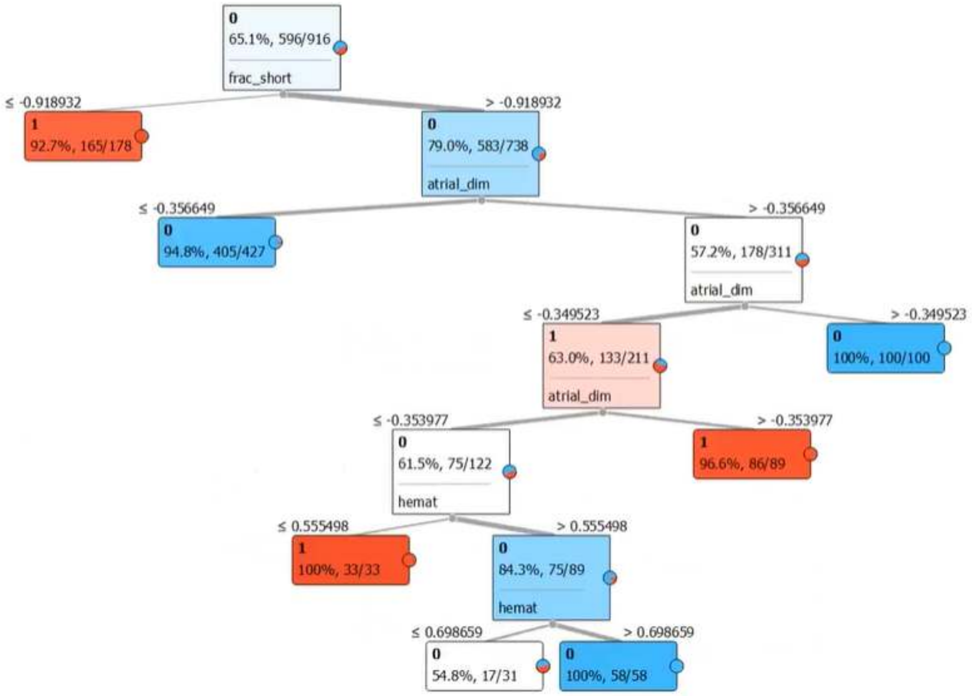
$$\begin{aligned} I_G(q_1, q_2, q_3, q_4) &= \frac{6}{23} I_G(q_1) + \frac{5}{23} I_G(q_2) + \frac{7}{23} I_G(q_3) + \frac{5}{23} I_G(q_4) \\ &= \frac{6}{23} 0.278 + \frac{5}{23} 0.638 + \frac{7}{23} 0.194 + \frac{5}{23} 0.528 = 0.370. \end{aligned}$$

$$\begin{aligned} \Delta_G(area) &= \Delta_G(q, q_1, q_2, q_3, q_4) = I_G(q) - I_G(q_1, q_2, q_3, q_4) \\ &= 0.491 - 0.370 = 0.121. \end{aligned}$$

$$\begin{array}{llll} \Delta_G(numin) & = 0.037, & \Delta_G(Pland) & = 0.078, \\ \Delta_G(timein) & = 0.091, & \Delta_G(numsms) & = 0.052, \\ \Delta_G(numout) & = 0.043, & \Delta_G(numserv) & = 0.038, \\ \Delta_G(Pothers) & = 0.146, & \Delta_G(numcall) & = 0.044, \\ \Delta_G(Pmob) & = 0.028, & \Delta_G(diropt) & = 0.003. \end{array}$$

With the Gini index also, Pother is also the one with the max gain of information.

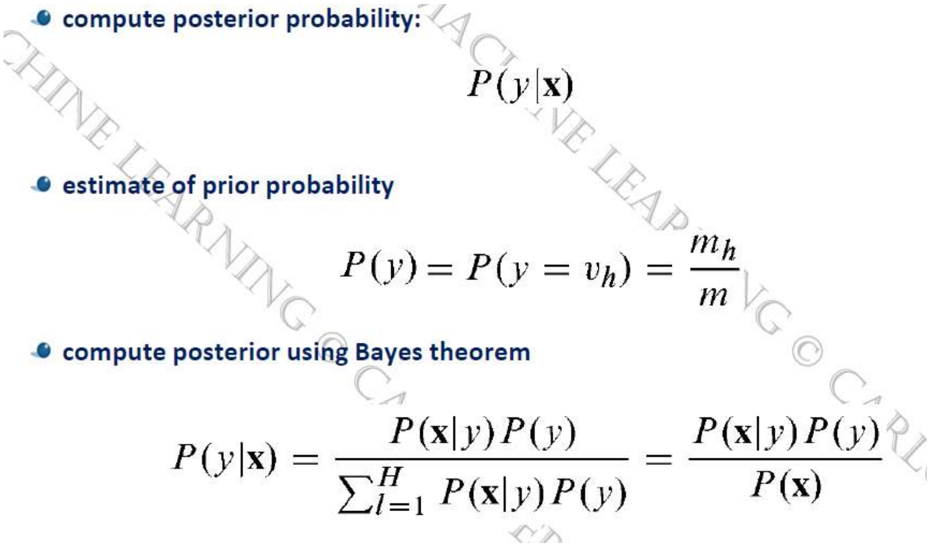
The output of the final classification tree :



Classification Part 3 :

https://politecnicomilano.webex.com/recordingservice/sites/politecnicomilano/recording/playback/c826eecf5c5740b289fbae48155392b6

Bayesian methods



- compute posterior probability:

$$P(y|\mathbf{x})$$

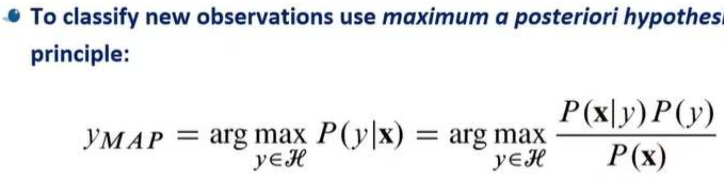
- estimate of prior probability

$$P(y) = P(y = v_h) = \frac{m_h}{m}$$

- compute posterior using Bayes theorem

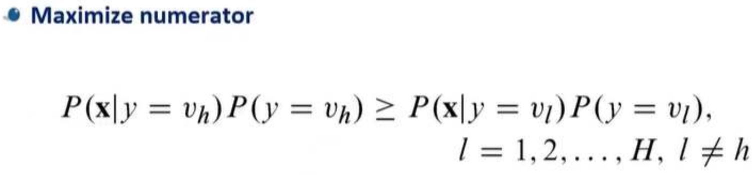
$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{\sum_{l=1}^{H} P(\mathbf{x}|y)P(y)} = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

P(x|y) can be estimated using past experiences data. For example, observing the state of a patient after applying the test and so you know how often it based on relative frequencies you can estimate this probability. The probability of X a given Y can be estimated empirically based on your past experience this is the real strength of bayes theorem middle expression.



- **To classify new observations use *maximum a posteriori hypothesis* (MAP) principle:**

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



- **Maximize numerator**

$$P(\mathbf{x}|y = v_h)P(y = v_h) \geq P(\mathbf{x}|y = v_l)P(y = v_l), \quad l = 1, 2, \ldots, H, \ l \neq h$$

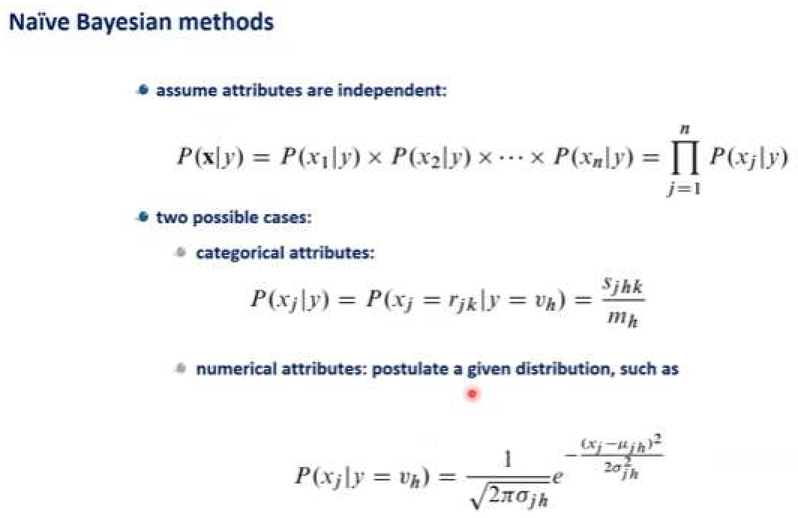
Suppose I have a new observation, I want to predict the label we apply bayes theorem we calculate the expression for all the possible values of Y and which select we choose that class label for the target variable which corresponds to the maximum of these expressions so this called maximum a posteriori hypothesis (MAP). So we maximize the posterior probability by using the bayes theorem.

Since the denominator P(x) is independent of y, in order to maximize the posterior probability it is enough to maximize the numerator of the expression Ymap.

We cannot apply this approach in practical situations quite because the medical test diagnosis I mentioned earlier I suppose that the state of the patient is represented by a single variable categorical in that case, binary being suffering a from of illness or not, and in that case the empirical estimation make sense.

consider my our situation the applications I presented to you of machine learning of classification X is a very high dimensional vector for the example of X being the vector of characteristics of a customer in the churn example not in a small churn example used for teaching purpose but in the real world situation how many components it has as well 100,000 actually of components how often do you think that you observe the outcome why being a churner or not

if X has a very complex combination, the probability of observing the same X in the past is very low. That’s why sample estimation P(X|y) cannot be done in practice, because we have a very complex input. Solution, two other types: **Naive Bayesian classifiers** and **Bayesian networks.**



Naïve Bayesian methods

- assume attributes are independent:

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

- two possible cases:
  - categorical attributes:

$$P(x_j|y) = P(x_j = r_{jk}|y = v_h) = \frac{s_{jhk}}{m_h}$$

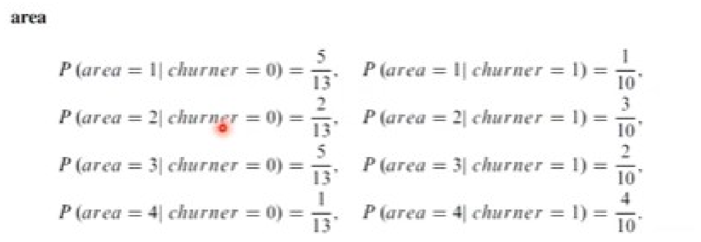
  - numerical attributes: postulate a given distribution, such as

$$P(x_j|y = v_h) = \frac{1}{\sqrt{2\pi\sigma_{jh}}} e^{-\frac{(x_j-\mu_{jh})^2}{2\sigma_{jh}^2}}$$

We suppose that the attributes are independent

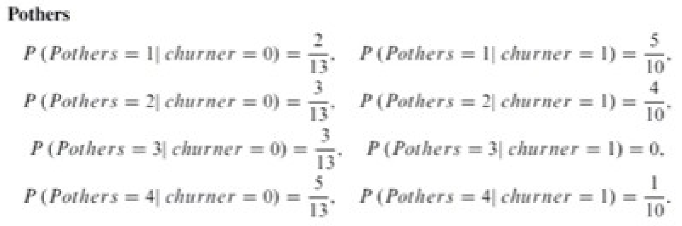
We use Relative frequencies in case of categorical attributes (first formula), but when it’s continuous we make an assumption if it’s normally distributed we use the (second formula)

Example :



area

$$P(area = 1 | churner = 0) = \frac{5}{13}, \quad P(area = 1 | churner = 1) = \frac{1}{10},$$
$$P(area = 2 | churner = 0) = \frac{2}{13}, \quad P(area = 2 | churner = 1) = \frac{3}{10},$$
$$P(area = 3 | churner = 0) = \frac{5}{13}, \quad P(area = 3 | churner = 1) = \frac{2}{10},$$
$$P(area = 4 | churner = 0) = \frac{1}{13}, \quad P(area = 4 | churner = 1) = \frac{4}{10}.$$

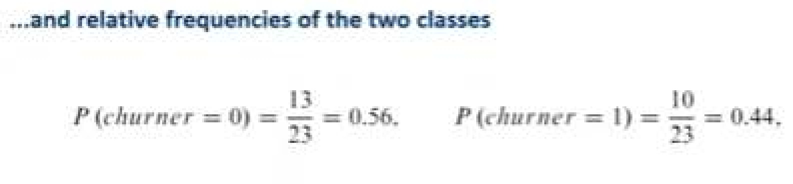


Pothers

$$P(Pothers = 1 | churner = 0) = \frac{2}{13}, \quad P(Pothers = 1 | churner = 1) = \frac{5}{10},$$
$$P(Pothers = 2 | churner = 0) = \frac{3}{13}, \quad P(Pothers = 2 | churner = 1) = \frac{4}{10},$$
$$P(Pothers = 3 | churner = 0) = \frac{3}{13}, \quad P(Pothers = 3 | churner = 1) = 0,$$
$$P(Pothers = 4 | churner = 0) = \frac{5}{13}, \quad P(Pothers = 4 | churner = 1) = \frac{1}{10}.$$

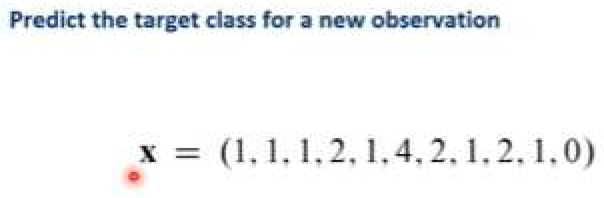
Bayesian method works better for categorical variables.

The rest of the attributes is in the book



...and relative frequencies of the two classes

$$P(churner = 0) = \frac{13}{23} = 0.56, \qquad P(churner = 1) = \frac{10}{23} = 0.44.$$



Predict the target class for a new observation

$$\mathbf{x} = (1, 1, 1, 2, 1, 4, 2, 1, 2, 1, 0)$$

With this aim in mind, we compute the posterior probabilities P(x|0) and P(x|1):

$$P(\mathbf{x}|0) = \frac{5}{13} \cdot \frac{4}{13} \cdot \frac{4}{13} \cdot \frac{3}{13} \cdot \frac{2}{13} \cdot \frac{3}{13} \cdot \frac{4}{13} \cdot \frac{3}{13} \cdot \frac{7}{13} \cdot \frac{12}{13} \cdot \frac{10}{13} = 0.81 \cdot 10^{-5},$$

$$P(\mathbf{x}|1) = \frac{1}{10} \cdot \frac{5}{10} \cdot \frac{6}{10} \cdot \frac{5}{10} \cdot \frac{5}{10} \cdot \frac{1}{10} \cdot \frac{6}{10} \cdot \frac{5}{10} \cdot \frac{4}{10} \cdot \frac{9}{10} \cdot \frac{7}{10} = 5.67 \cdot 10^{-5}.$$

The 5/13=p(area=1| churner=0)

4/13=p(numin=1| churner=0)

And so on…

Since the relative frequencies of the two classes are given by

$$P(churner = 0) = \frac{13}{23} = 0.56, \qquad P(churner = 1) = \frac{10}{23} = 0.44,$$

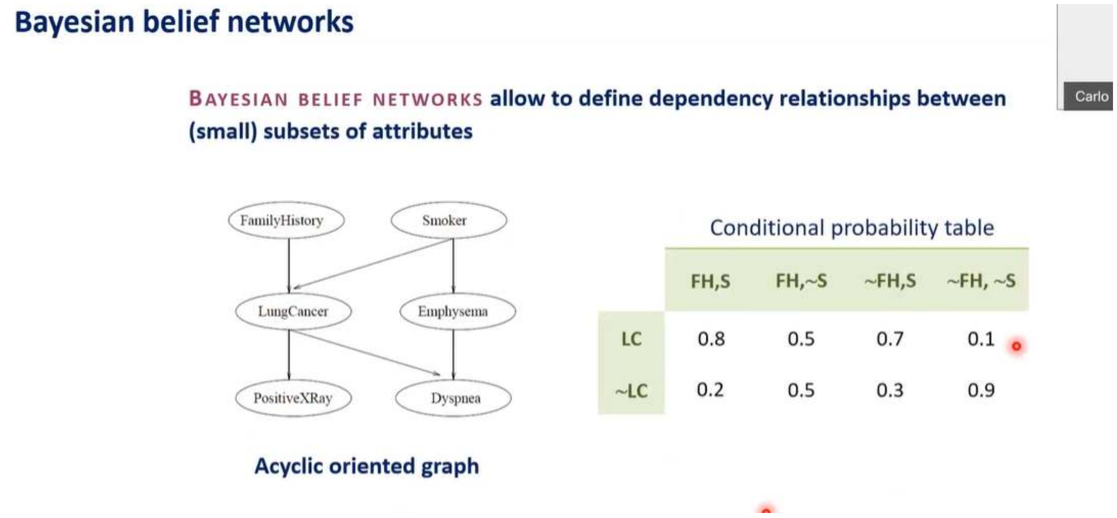
we have

$$\begin{aligned} P(churner = 0|\mathbf{x}) &= P(\mathbf{x}|0)\,P(churner = 0) \\ &= 0.81 \cdot 10^{-5} \cdot 0.56 = 0.46 \cdot 10^{-5} \quad /p(x) \\ P(churner = 1|\mathbf{x}) &= P(\mathbf{x}|1)\,P(churner = 1) \\ &= 5.67 \cdot 10^{-5} \cdot 0.44 = 2.495 \cdot 10^{-5} \quad /p(x) \end{aligned}$$

We can see that both probabilities don't add up to 100%, this is because we didn't divide them by p(x)

The new example x is then labeled with the class value {1}, since this is associated with the maximum a posteriori probability.

The other method used that can allow a relationship between attributes is **Bayesian networks,** it's used a lot in the medical field



Bayesian belief networks

BAYESIAN BELIEF NETWORKS allow to define dependency relationships between (small) subsets of attributes

Acyclic oriented graph

Conditional probability table

	FH,S	FH,~S	~FH,S	~FH, ~S
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

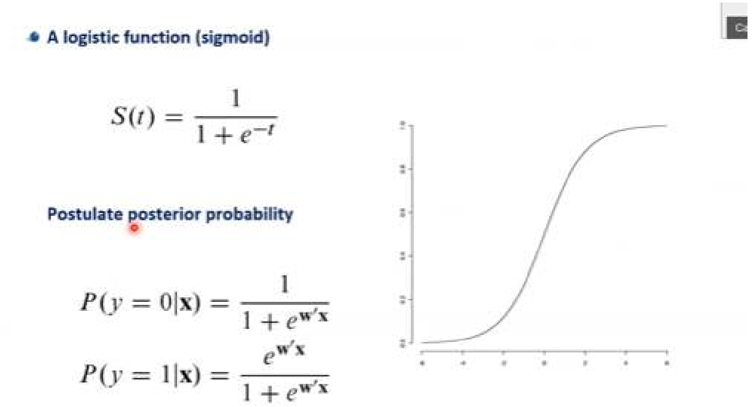
We can see that having a lung cancer LC is strongly related to Family history (FH) and being a Smoker (S) The probability is 0.8 and vis versa, ~FH (Not related to family history) and ~S have a small probability 0.1.

In this case wd can use conditional probabilities with their exact value in the calculation instead of using the multiplication formula used in the Naive method.

Bayesian methods are parametric methods.

Logistic regression methods

Discrimination analysis the term used in the past for classification analysis. The logistic regression is also a parametric approach because it’s based on a postulate



- A logistic function (sigmoid)

$$S(t) = \frac{1}{1+e^{-t}}$$

Postulate posterior probability

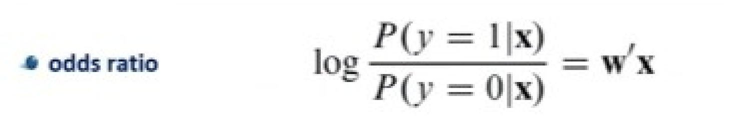
$$P(y=0|\mathbf{x}) = \frac{1}{1+e^{\mathbf{w}'\mathbf{x}}}$$

$$P(y=1|\mathbf{x}) = \frac{e^{\mathbf{w}'\mathbf{x}}}{1+e^{\mathbf{w}'\mathbf{x}}}$$

W in the formula is a vector of coefficient that needs to be estimated.

The posterior probability follows a sigmoid function

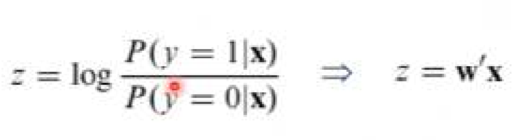
The two terms add to one, one is the complement of the other. If we dived one by the other and introduce the logarithm we end up with :



- odds ratio

$$\log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = \mathbf{w}'\mathbf{x}$$

And then the problem can be transformed to linear regression model :

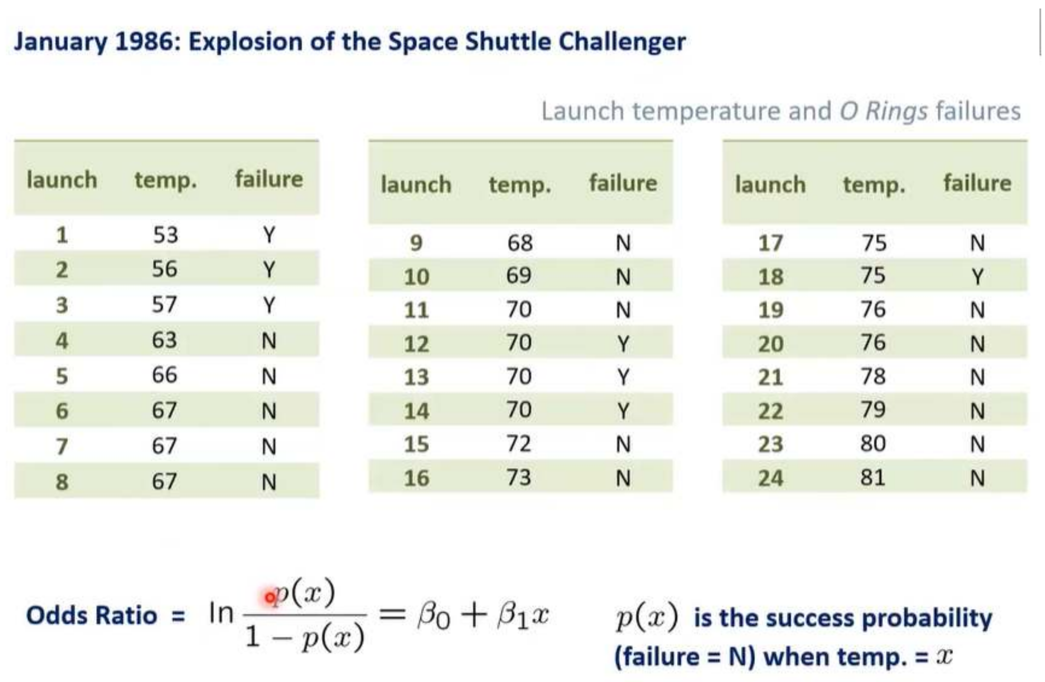


$$z = \log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} \quad \Rightarrow \quad z = \mathbf{w}'\mathbf{x}$$

Advantage : easy to develop

Problems : works very poorly with big numbers of attributes when you increase the vector x, very limited potential.

Example :



January 1986: Explosion of the Space Shuttle Challenger

Launch temperature and *O Rings* failures

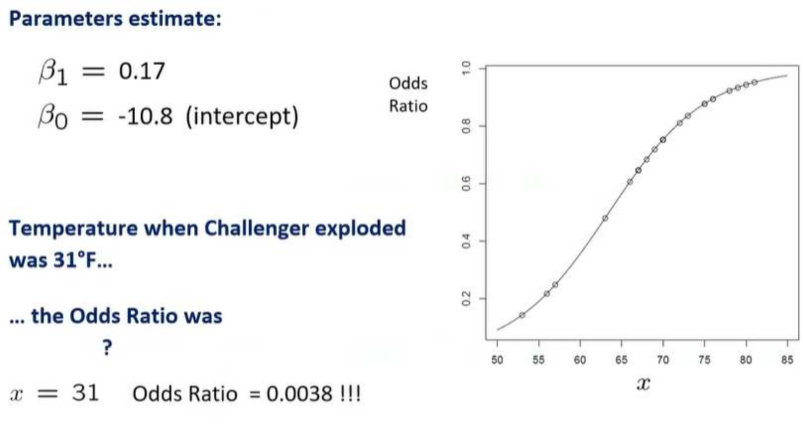
launch	temp.	failure
1	53	Y
2	56	Y
3	57	Y
4	63	N
5	66	N
6	67	N
7	67	N
8	67	N

launch	temp.	failure
9	68	N
10	69	N
11	70	N
12	70	Y
13	70	Y
14	70	Y
15	72	N
16	73	N

launch	temp.	failure
17	75	N
18	75	Y
19	76	N
20	76	N
21	78	N
22	79	N
23	80	N
24	81	N

Odds Ratio = $$\ln \frac{p(x)}{1-p(x)} = \beta_0 + \beta_1 x$$

$p(x)$ **is the success probability (failure = N) when temp. =** x



Parameters estimate:

$\beta_1 =$ 0.17

$\beta_0 =$ -10.8 (intercept)

Temperature when Challenger exploded was 31°F...

... the Odds Ratio was

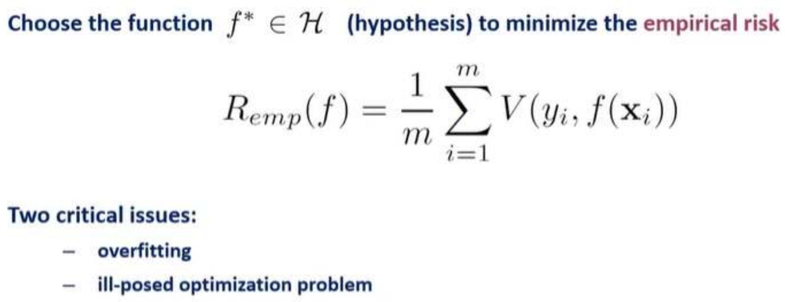
?

$x = 31$ Odds Ratio = 0.0038 !!!

We can see very good fitting of the points in the curve.

The odds ratio is very low so the failure of the rings is almost certain

n) Classification and statistical learning theory

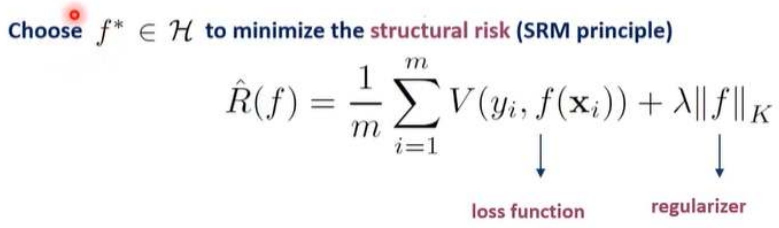


Choose the function $f^* \in \mathcal{H}$ (hypothesis) to minimize the empirical risk

$$R_{emp}(f) = \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i))$$

Two critical issues:

- overfitting
- ill-posed optimization problem

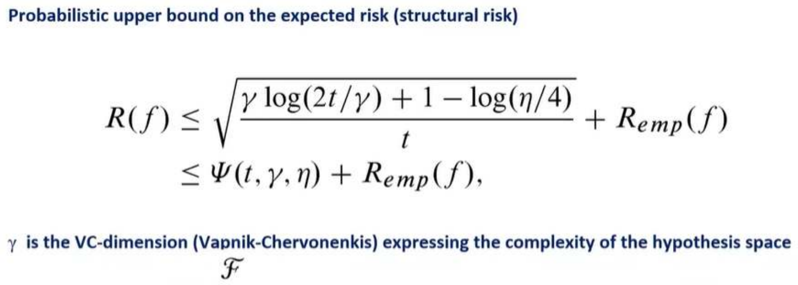


Choose $f^* \in \mathcal{H}$ to minimize the structural risk (SRM principle)

$$\hat{R}(f) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{V(y_i, f(\mathbf{x}_i))}_{\text{loss function}} + \underbrace{\lambda\|f\|_K}_{\text{regularizer}}$$

First method used to choose f* was based on minimizing the empirical risk (sum of misclassified points divided by the number m⇔ error) but it leads to overfitting and the second problem is ill-posed optimization problem (small changes in the input data correspond to big changes in the solution the solution is not robust)

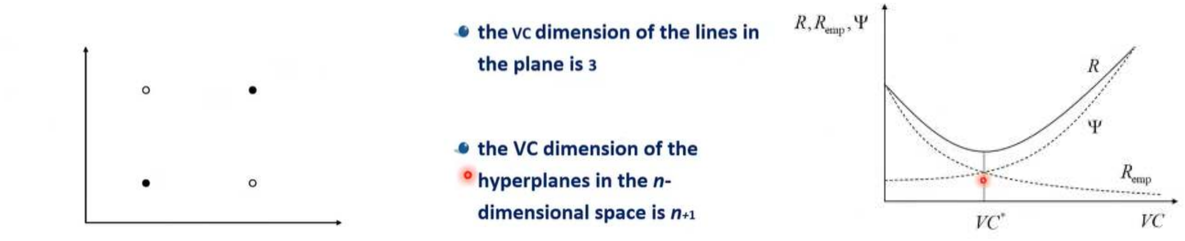
Then they introduced the structural risk that have two terms: the fist one is the empirical risk and the second is a factor to keep generalization capability.



Probabilistic upper bound on the expected risk (structural risk)

$$\begin{aligned} R(f) &\leq \sqrt{\frac{\gamma \log(2t/\gamma) + 1 - \log(\eta/4)}{t}} + R_{emp}(f) \\ &\leq \Psi(t, \gamma, \eta) + R_{emp}(f), \end{aligned}$$

γ is the VC-dimension (Vapnik-Chervonenkis) expressing the complexity of the hypothesis space $\mathcal{F}$



- the VC dimension of the lines in the plane is 3
- the VC dimension of the hyperplanes in the *n*-dimensional space is *n*+1

t: number of observations in the training set

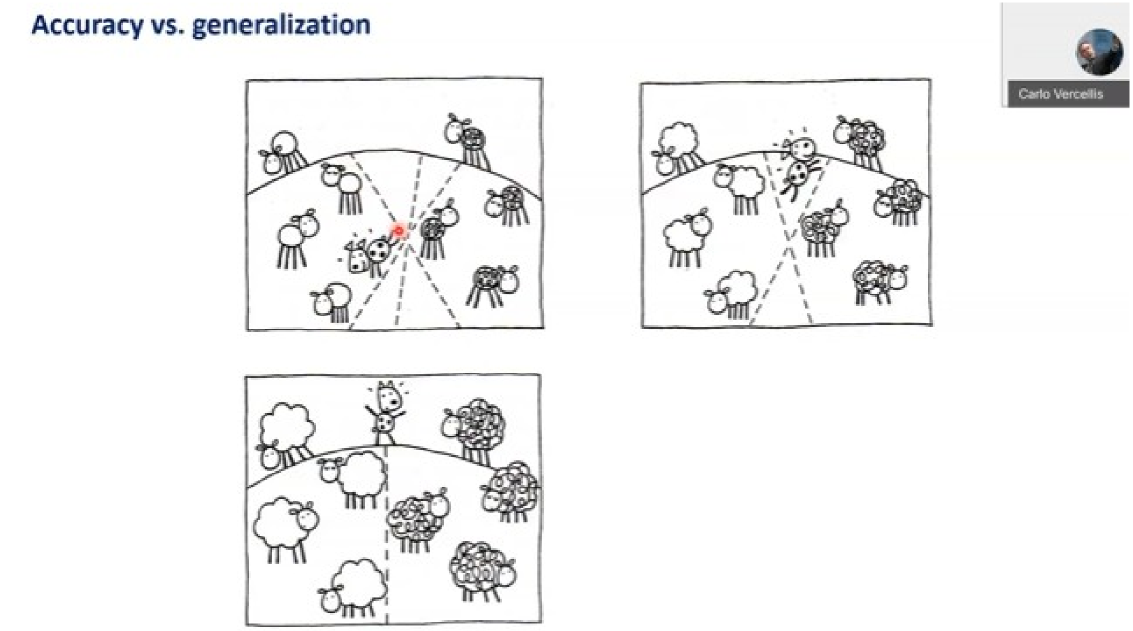
Gama: complexity of the hypotheses space (the more complex the higher of over fitting) also known as VC dimension : number of point that you can separate no matter how you place them in the space

Eta : the confidence level : threshold

Classification part 4 :

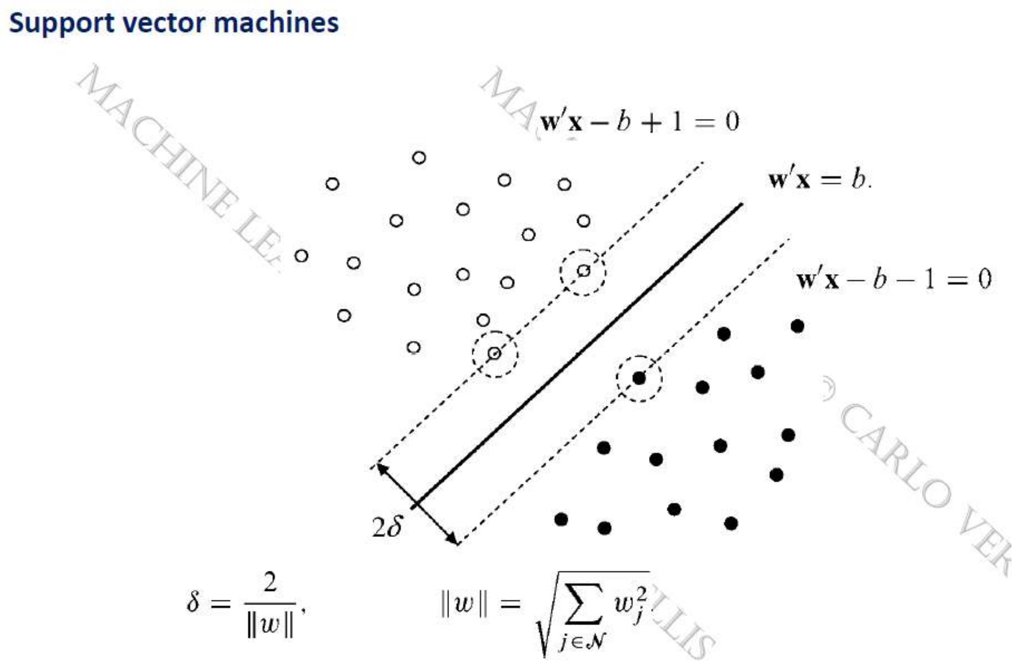
https://politecnicomilano.webex.com/recordingservice/sites/politecnicomilano/recording/playback/55fbe709962041c39c70f2d0ed388757

Support vector machines



Accuracy vs. generalization

Maximum margin of separation

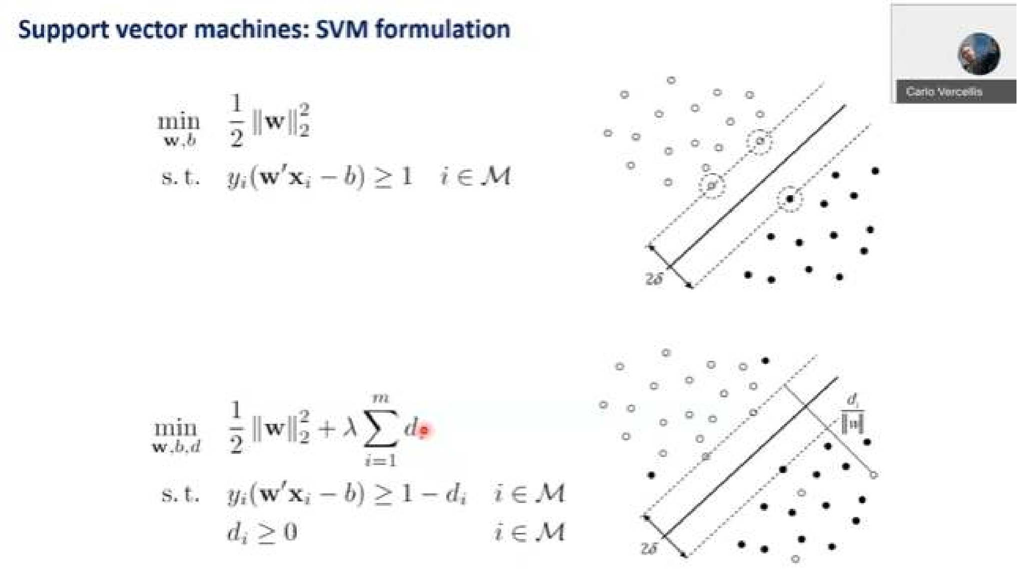


Support vector machines

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

Why exactly -1 and +1 : because any coefficient can be divided by it's value and brought back to one. The marginal separation is 2*delta. It can be shown that delta is 2/ the norm of w. We want to maximize delta which mean minimize w.

So we need to solve this problem with conditions :



Support vector machines: SVM formulation

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

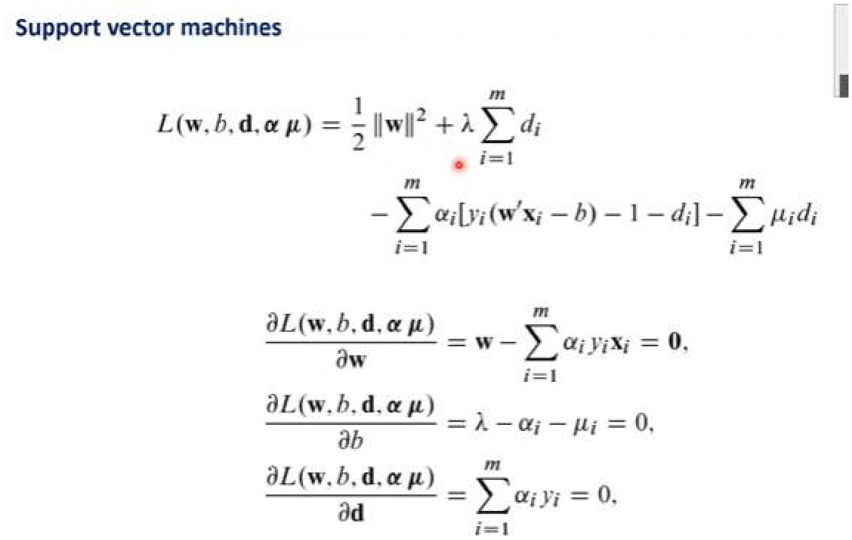
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 \quad i \in \mathcal{M}$$

$$\min_{\mathbf{w},b,d} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^{m} d_i$$

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

$$d_i \geq 0 \quad i \in \mathcal{M}$$

We can introduce the LaGrange to this problem :



Support vector machines

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{m} d_i - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 - d_i] - \sum_{i=1}^{m} \mu_i d_i$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = \mathbf{0},$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial b} = \lambda - \alpha_i - \mu_i = 0,$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial \mathbf{d}} = \sum_{i=1}^{m} \alpha_i y_i = 0,$$

Landa is a hyper parameter used to evaluate our choice . we solve this problem by calculating the partial derivatives

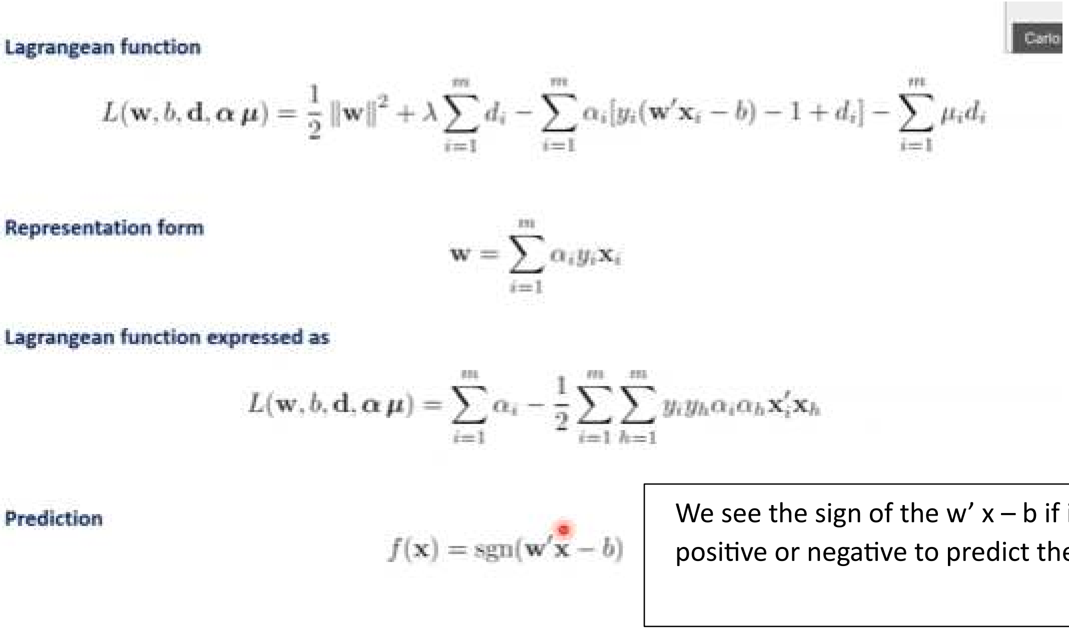
$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i,$$

$$\lambda = \alpha_i + \mu_i,$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{k=1}^{m} y_i y_k \alpha_i \alpha_k \mathbf{x}_i' \mathbf{x}_k$$

We can use duality to solve this problem. It’s a quadratic problem (see chapter 30 in the book optimisation).



Lagrangean function

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{m} d_i - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 + d_i] - \sum_{i=1}^{m} \mu_i d_i$$

Representation form

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

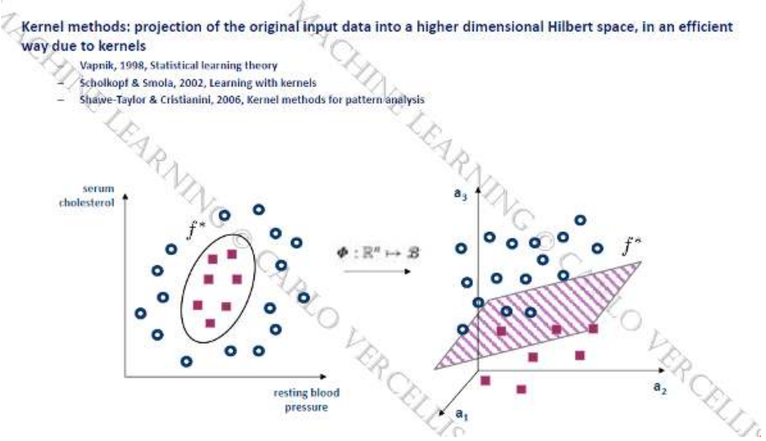
Lagrangean function expressed as

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{h=1}^{m} y_i y_h \alpha_i \alpha_h \mathbf{x}_i' \mathbf{x}_h$$

Prediction

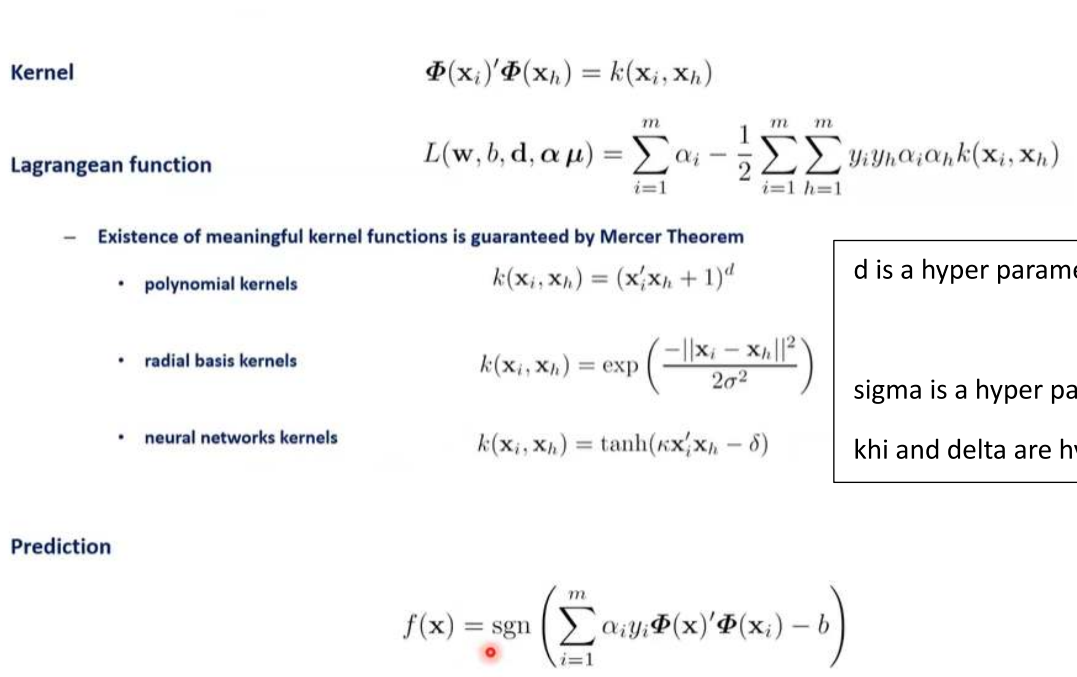
$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}'\mathbf{x} - b)$$

We see the sign of the w’ x – b if it’s positive or negative to predict the output



Kernel methods: projection of the original input data into a higher dimensional Hilbert space, in an efficient way due to kernels

- Vapnik, 1998, Statistical learning theory
- Scholkopf & Smola, 2002, Learning with kernels
- Shawe-Taylor & Cristianini, 2006, Kernel methods for pattern analysis



Kernel $\boldsymbol{\Phi}(\mathbf{x}_i)'\boldsymbol{\Phi}(\mathbf{x}_h) = k(\mathbf{x}_i, \mathbf{x}_h)$

Lagrangean function $L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\,\boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{h=1}^{m} y_i y_h \alpha_i \alpha_h k(\mathbf{x}_i, \mathbf{x}_h)$

- **Existence of meaningful kernel functions is guaranteed by Mercer Theorem**
  - **polynomial kernels** $k(\mathbf{x}_i, \mathbf{x}_h) = (\mathbf{x}_i'\mathbf{x}_h + 1)^d$
  - **radial basis kernels** $k(\mathbf{x}_i, \mathbf{x}_h) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_h||^2}{2\sigma^2}\right)$
  - **neural networks kernels** $k(\mathbf{x}_i, \mathbf{x}_h) = \tanh(\kappa\mathbf{x}_i'\mathbf{x}_h - \delta)$

Prediction

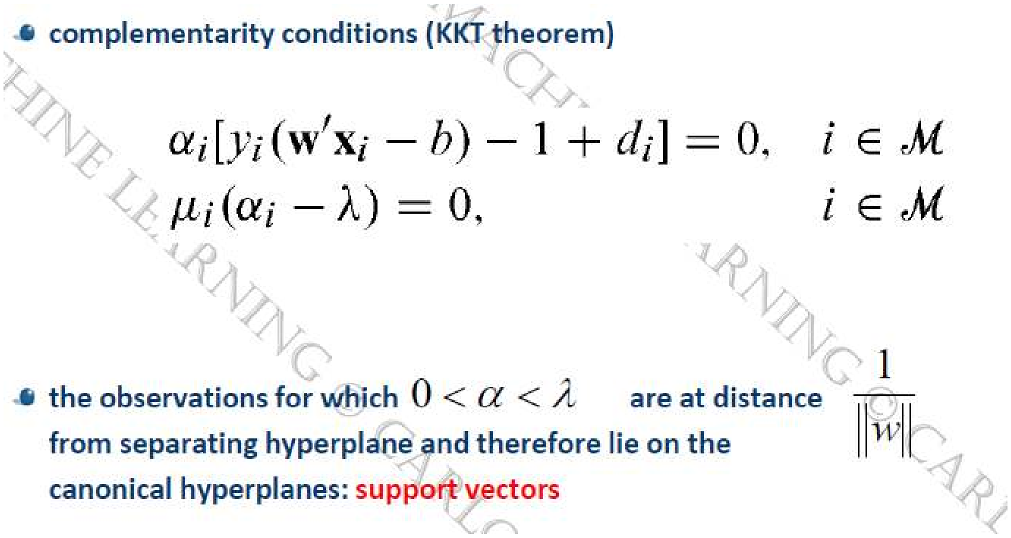
$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{\Phi}(\mathbf{x})'\boldsymbol{\Phi}(\mathbf{x}_i) - b\right)$$

d is a hyper parameter to define

sigma is a hyper parameter

khi and delta are hyper parameters

Kernal : (to be search exactly) veeeeery important

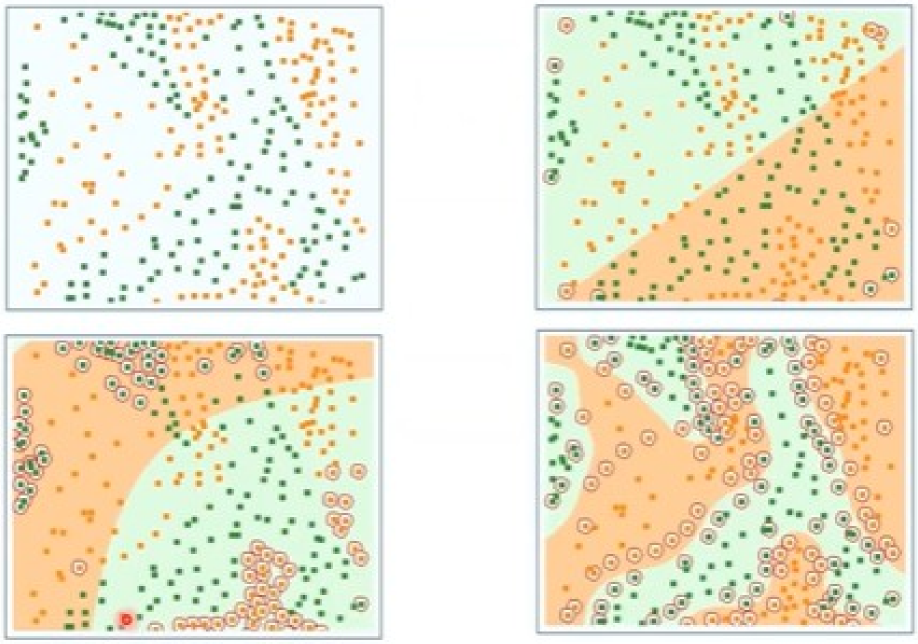


- complementarity conditions (KKT theorem)

$$\begin{aligned} &\alpha_i[y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 + d_i] = 0, && i \in \mathcal{M} \\ &\mu_i(\alpha_i - \lambda) = 0, && i \in \mathcal{M} \end{aligned}$$

- the observations for which $0 < \alpha < \lambda$ are at distance $\frac{1}{\|w\|}$ from separating hyperplane and therefore lie on the canonical hyperplanes: support vectors

SVM: linear and nonlinear learning with kernels



Neural networks

The Rosenblatt perceptron:

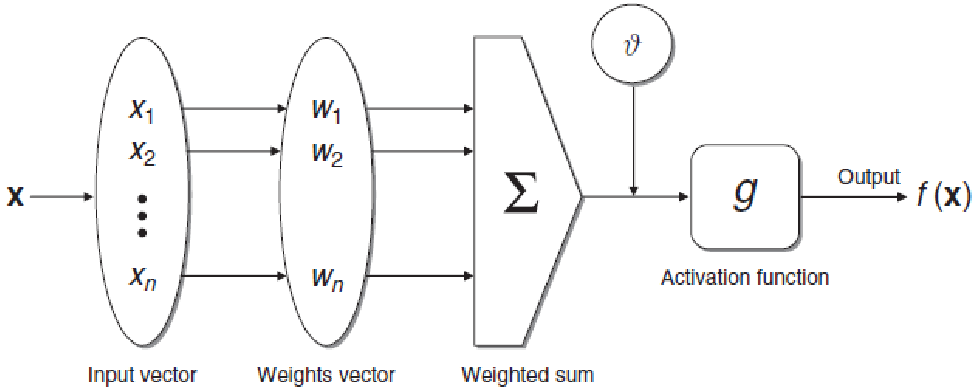
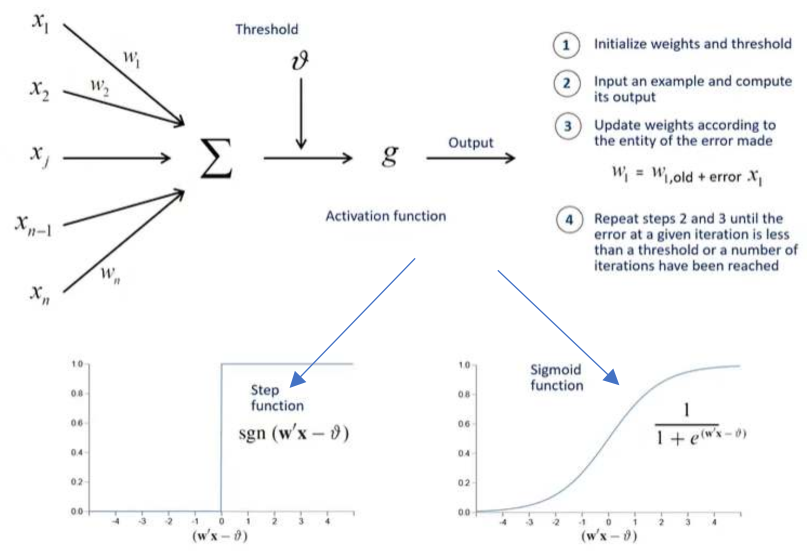


Figure 10.16 Operation of a single unit in a neural network

Rosenblatt's Perceptron



1. Initialize weights and threshold
2. Input an example and compute its output
3. Update weights according to the entity of the error made

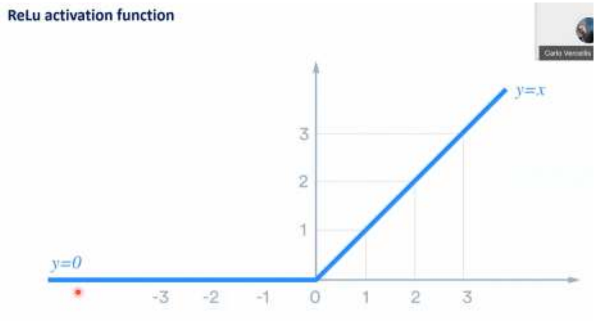
$$w_1 = w_{1,\text{old}} + \text{error}\ x_1$$

4. Repeat steps 2 and 3 until the error at a given iteration is less than a threshold or a number of iterations have been reached

Steps of the algorithm

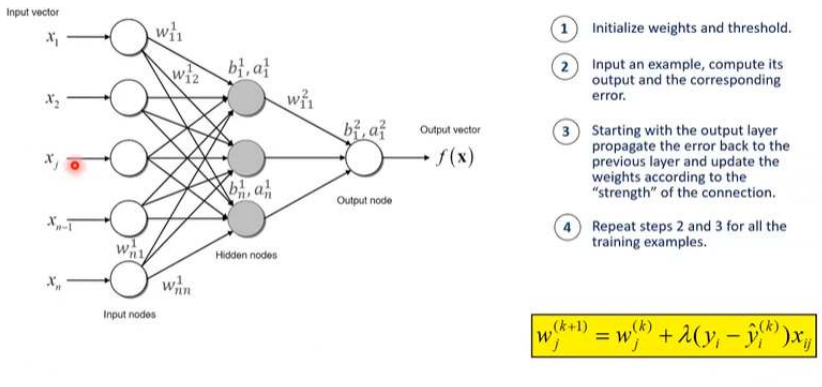
The stop condition can be a number of iterations (2000) or a threshold for the error

Rectified linear unite as activation function : used mostly in internal noads


ReLu activation function

Multi-layer feed-forward neural networks

Each hidden unit can be considered as a single perceptron.



1. Initialize weights and threshold.
2. Input an example, compute its output and the corresponding error.
3. Starting with the output layer propagate the error back to the previous layer and update the weights according to the "strength" of the connection.
4. Repeat steps 2 and 3 for all the training examples.

$$w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$$

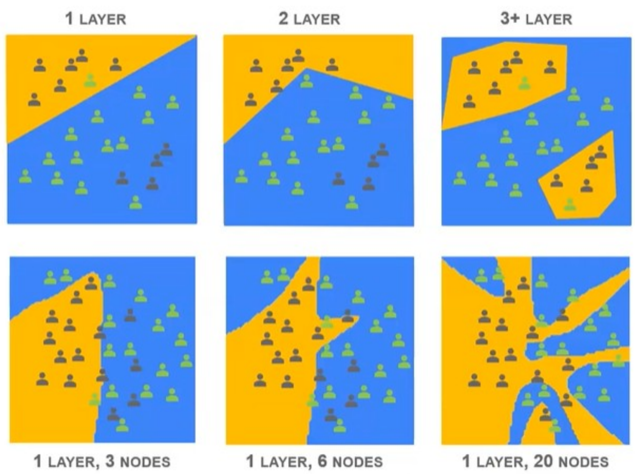
$a_i^l = \phi(\sum_n w_{ni}^l a_n^{l-1} + b_i^l)$, where ϕ is the element-wise activation function

Backpropagation: (to see what does it mean)

2 layers, 1 node : non -linear separation

From neural Networks to deep learning

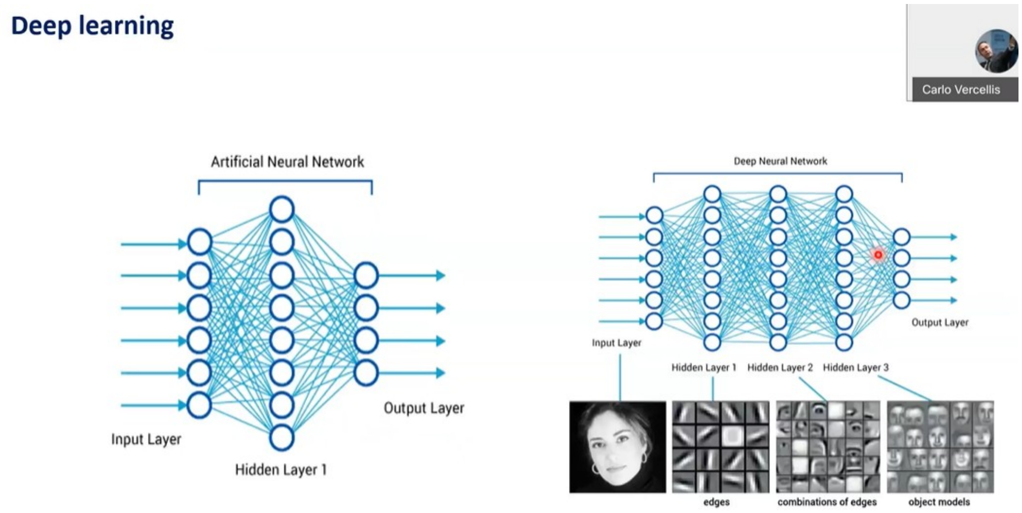
One layer – one node : linear separation



3 layers, 1 node : Polygonal separation

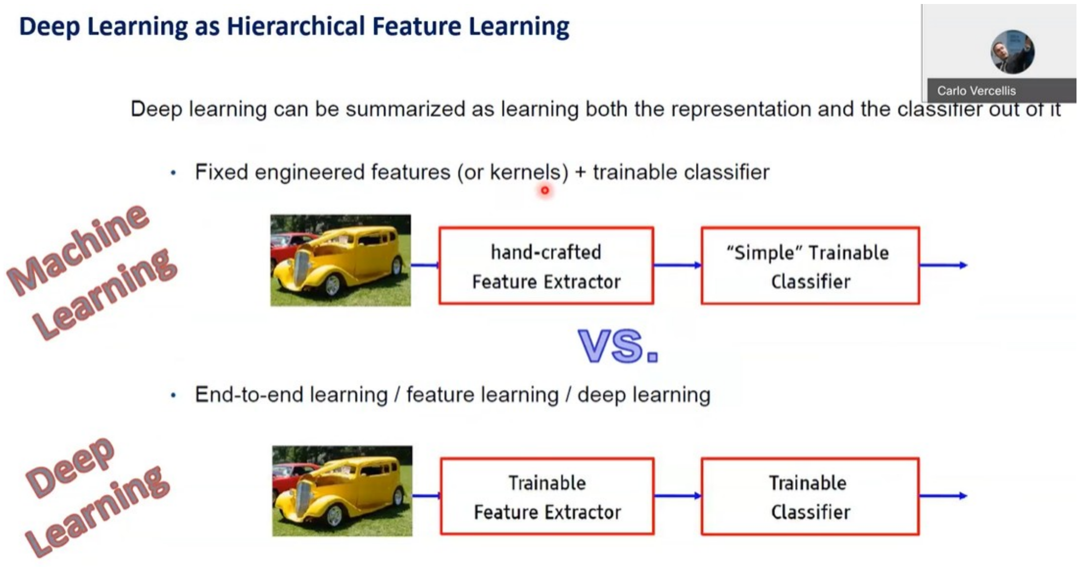
More complex hypotheses result to over fitting

When we increase the number of layers we are in deep learning



Deep learning

In machine learning we define features (kernals, PCA eg.) defined by Humans based on their experience whereases in deep learning it's learning the presentation and the classifier



Deep Learning as Hierarchical Feature Learning

Deep learning can be summarized as learning both the representation and the classifier out of it

- Fixed engineered features (or kernels) + trainable classifier

Machine Learning

hand-crafted Feature Extractor → "Simple" Trainable Classifier →

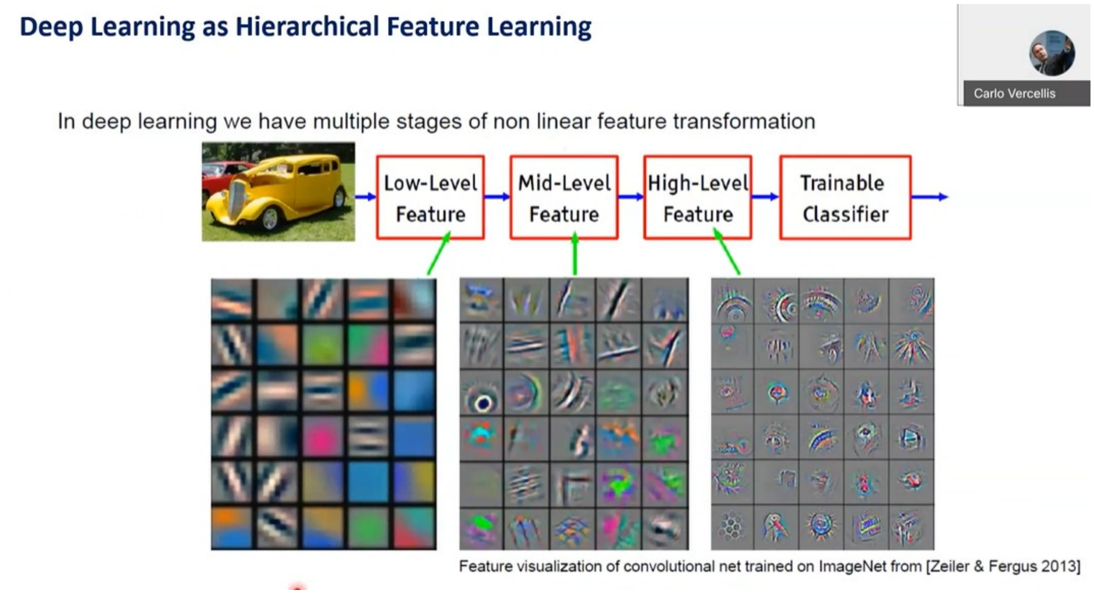
VS.

- End-to-end learning / feature learning / deep learning

Deep Learning

Trainable Feature Extractor → Trainable Classifier →

This technique is more used in imagery data but when using structured data it's not very different from traditional techniques.



Deep Learning as Hierarchical Feature Learning

In deep learning we have multiple stages of non linear feature transformation

Low-Level Feature → Mid-Level Feature → High-Level Feature → Trainable Classifier →

Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

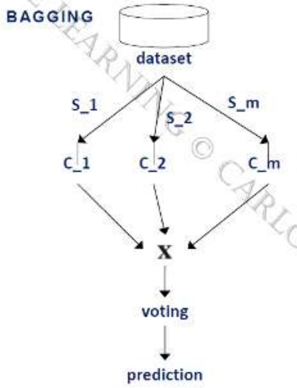
Last idea :

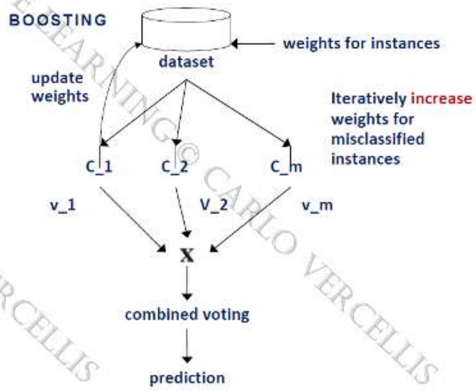
Combining many classifiers and relying on majority voting. The idea is selecting different classifiers and fitting a randomly selected portion of data to them. We get many algorithm by this method. Once a new observation comes, we get different result and we select one by majority voting.

In boosting we use a weight to every classifier to reduce errors every time we train the model and in the end we choose the result value by voting

Ensemble classifiers

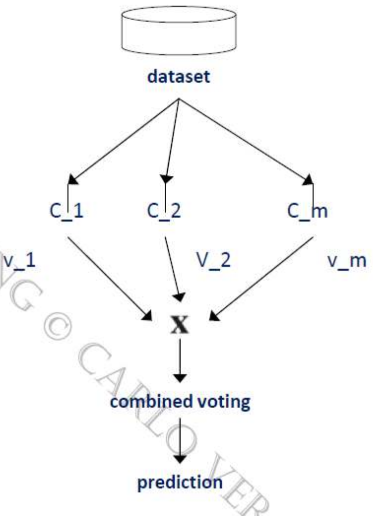
ENSEMBLE CLASSIFIERS are aimed at combining predictions obtained from different classifiers

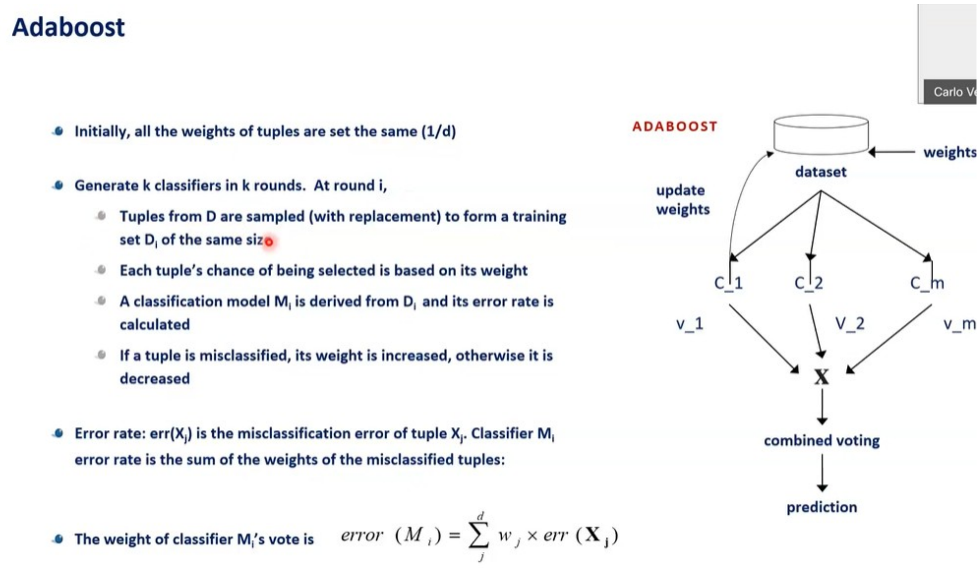




Random Forest

- **Each classifier in the ensemble is a classification tree and is generated using a random selection of attributes at each node to determine the split.**
- **Forest-RI (random input selection): Randomly select, at each node, F attributes as candidates for the split at the node.**
- **Forest-RC (random linear combinations): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers).**





Adaboost

- Initially, all the weights of tuples are set the same (1/d)
- Generate k classifiers in k rounds. At round i,
  - Tuples from D are sampled (with replacement) to form a training set D_i of the same size
  - Each tuple's chance of being selected is based on its weight
  - A classification model M_i is derived from D_i and its error rate is calculated
  - If a tuple is misclassified, its weight is increased, otherwise it is decreased
- Error rate: $err(X_j)$ is the misclassification error of tuple X_j. Classifier M_i error rate is the sum of the weights of the misclassified tuples:
- The weight of classifier M_i's vote is $error(M_i) = \sum_j^d w_j \times err(\mathbf{X}_j)$

5. Regression

https://politecnicomilano.webex.com/recordingservice/sites/politecnicomilano/recording/95d5ba59ee244c49891b25cf6e7af999/playback

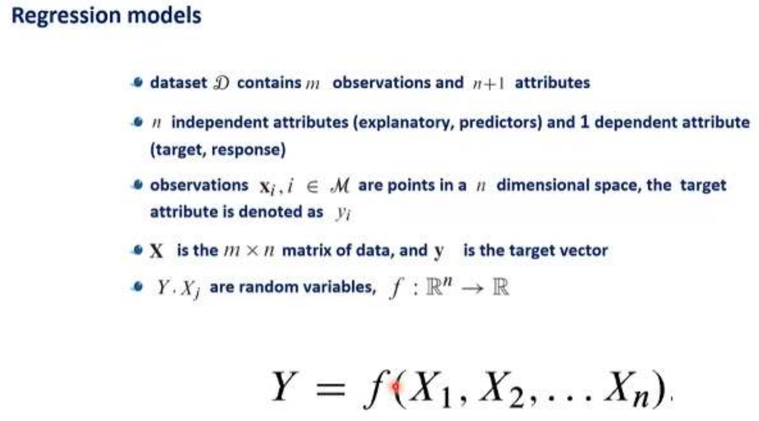
Is the numerical part of classification (it's a part of supervised learning). Most of people when hearing regression they think about least square regression (which is used a lot in statistics).

Consider a data set with m observation and n+1 attribute. n are the explanatory variables and 1 is the target value. The target attribute is also called the **dependent variable**, **response** or **output**, while the explanatory variables are also termed **independent variables** or **predictors**.

The target value is categorical or binary for classification and numerical in case of regression.

All the variables must be numerical (categorical attributes are transformed to dummy variables)

We want to explain Y based on the best function f. Regression models conjecture the existence of a function f : Rn → R that expresses the relationship between the dependent variable Y and the n explanatory variables Xj

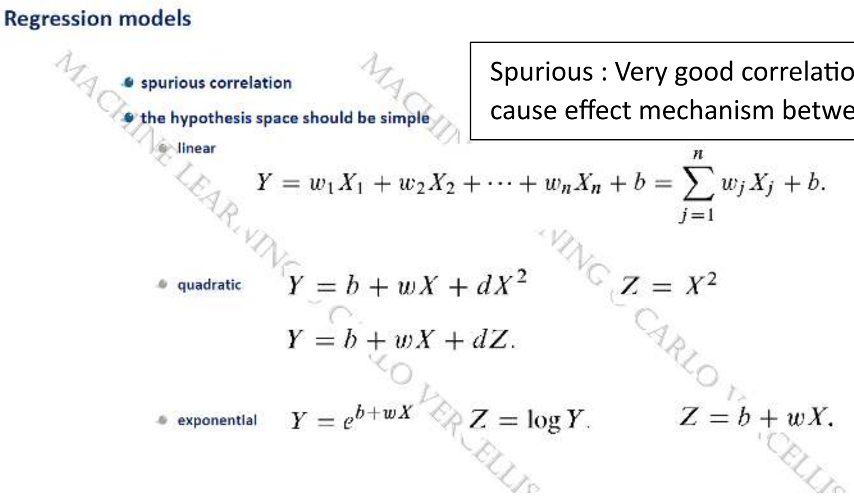


Regression models

- dataset $\mathcal{D}$ contains m observations and $n+1$ attributes
- n independent attributes (explanatory, predictors) and 1 dependent attribute (target, response)
- observations $\mathbf{x}_i, i \in \mathcal{M}$ are points in a n dimensional space, the target attribute is denoted as y_i
- $\mathbf{X}$ is the $m \times n$ matrix of data, and $\mathbf{y}$ is the target vector
- Y, X_j are random variables, $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$Y = f(X_1, X_2, \ldots X_n).$$

Thus, their goal is twofold. On one hand, regression models serve to highlight and interpret the dependency of the target variable on the other variables. On the other hand, they are used to predict the future value of the target attribute, based upon the functional relationship identified and the future value of the explanatory attributes. This opportunity is of great interest, particularly for analyzing those attributes that are **control levers** available to decision makers.



Regression models

- spurious correlation
- the hypothesis space should be simple
  - linear

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

  - quadratic

$$Y = b + wX + dX^2 \qquad Z = X^2$$

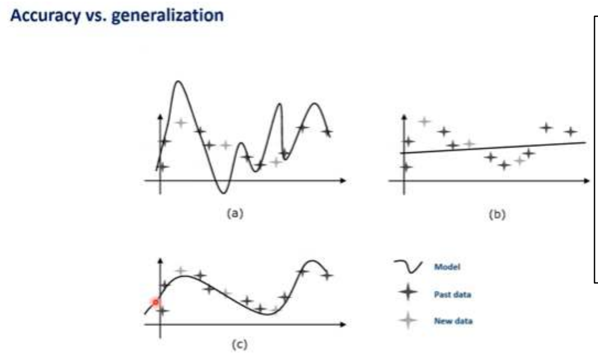
$$Y = b + wX + dZ.$$

  - exponential

$$Y = e^{b+wX} \qquad Z = \log Y. \qquad Z = b + wX.$$

Spurious : Very good correlation but you don't see any cause effect mechanism between X and Y.

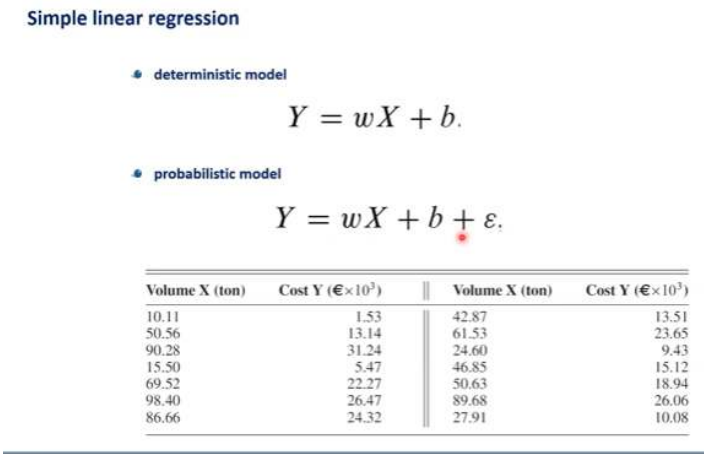
W and b are parameters to be determined by the algorithm itself. Z can be introduced to facilitate the model and transformed it to an easy one.



Accuracy vs. generalization

Overfitting : we can have a model with very high accuracy in the past data but then the error will be very high the new data.

Simple linear regression :



Simple linear regression

- deterministic model

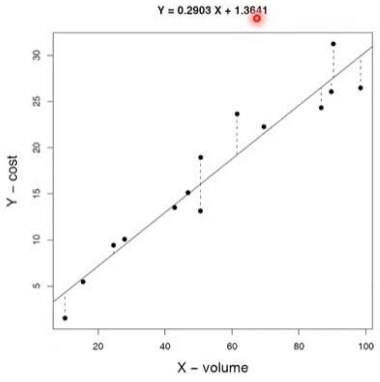
$$Y = wX + b.$$

- probabilistic model

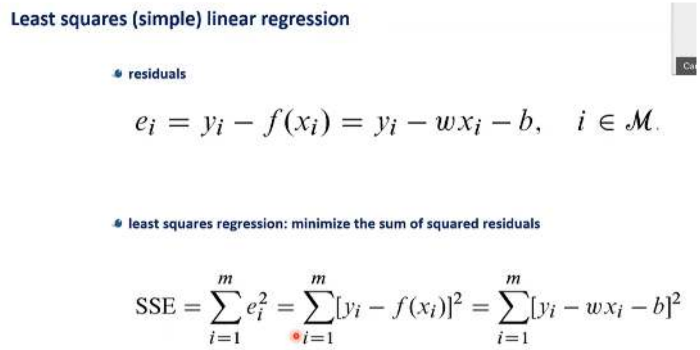
$$Y = wX + b + \varepsilon.$$

Volume X (ton)	Cost Y (€×10^3)	Volume X (ton)	Cost Y (€×10^3)
10.11	1.53	42.87	13.51
50.56	13.14	61.53	23.65
90.28	31.24	24.60	9.43
15.50	5.47	46.85	15.12
69.52	22.27	50.63	18.94
98.40	26.47	89.68	26.06
86.66	24.32	27.91	10.08

Example of least square regression



In the example we determined w and b. w=0.2903 and b=1.3641 we can say for example that 1.3641 is the fixe cost payed by the company. How to obtained this coefficients :



Least squares (simple) linear regression

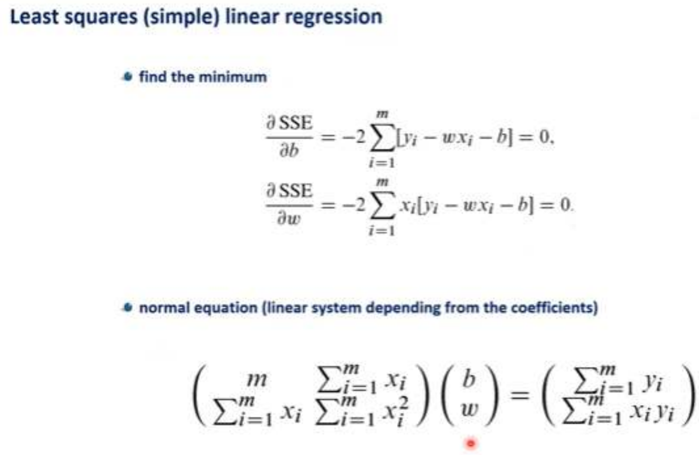
- residuals

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

- least squares regression: minimize the sum of squared residuals

$$\text{SSE} = \sum_{i=1}^{m} e_i^2 = \sum_{i=1}^{m} [y_i - f(x_i)]^2 = \sum_{i=1}^{m} [y_i - wx_i - b]^2$$

The technique consist of minimizing the squares of errors. It’s a Non constrained quadratic optimization. We can solve this problem by using partial derivatives :



Least squares (simple) linear regression

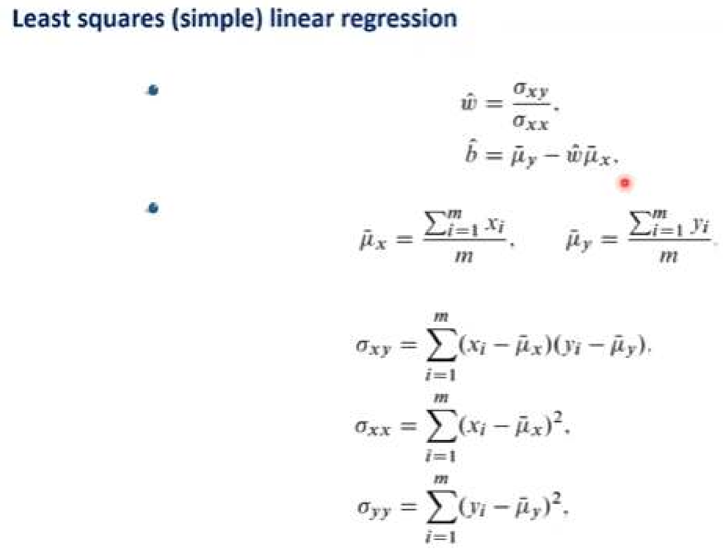
- find the minimum

$$\frac{\partial\, \text{SSE}}{\partial b} = -2\sum_{i=1}^{m} [y_i - wx_i - b] = 0,$$
$$\frac{\partial\, \text{SSE}}{\partial w} = -2\sum_{i=1}^{m} x_i[y_i - wx_i - b] = 0.$$

- normal equation (linear system depending from the coefficients)

$$\begin{pmatrix} m & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 \end{pmatrix} \begin{pmatrix} b \\ w \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i y_i \end{pmatrix}$$

The solution of this problem :



Least squares (simple) linear regression

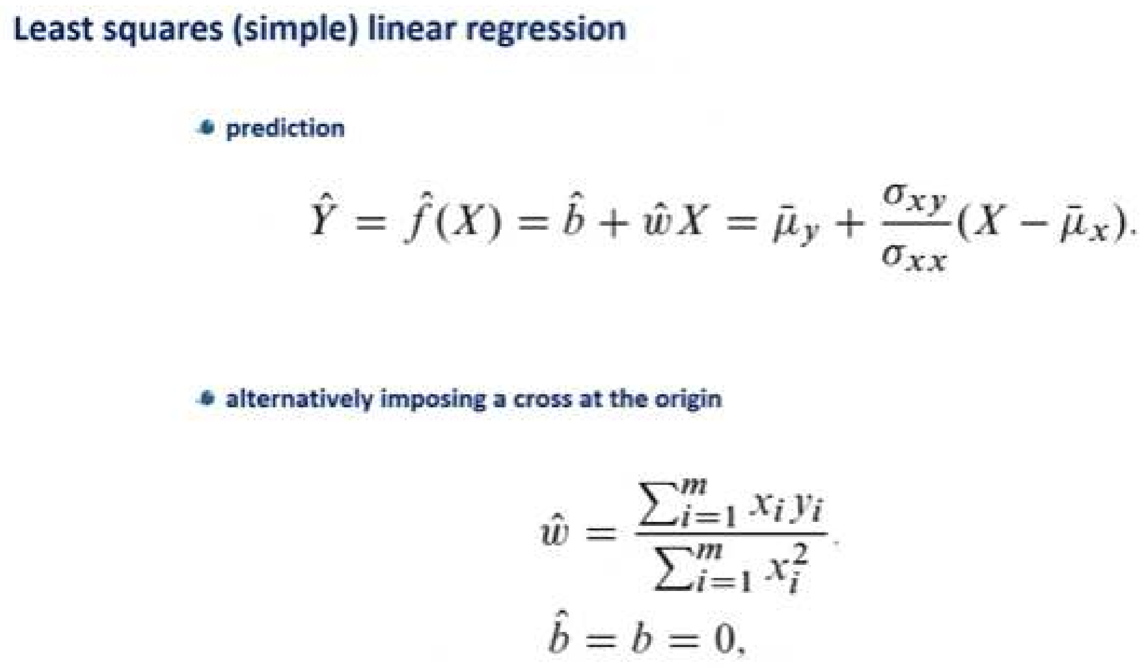
- $$\hat{w} = \frac{\sigma_{xy}}{\sigma_{xx}},$$
$$\hat{b} = \bar{\mu}_y - \hat{w}\bar{\mu}_x.$$

- $$\bar{\mu}_x = \frac{\sum_{i=1}^{m} x_i}{m}, \qquad \bar{\mu}_y = \frac{\sum_{i=1}^{m} y_i}{m}.$$

$$\sigma_{xy} = \sum_{i=1}^{m} (x_i - \bar{\mu}_x)(y_i - \bar{\mu}_y),$$
$$\sigma_{xx} = \sum_{i=1}^{m} (x_i - \bar{\mu}_x)^2,$$
$$\sigma_{yy} = \sum_{i=1}^{m} (y_i - \bar{\mu}_y)^2.$$

We are using B hate and w hate because it’s an estimation

And for the prediction of values we use :



Least squares (simple) linear regression

- prediction

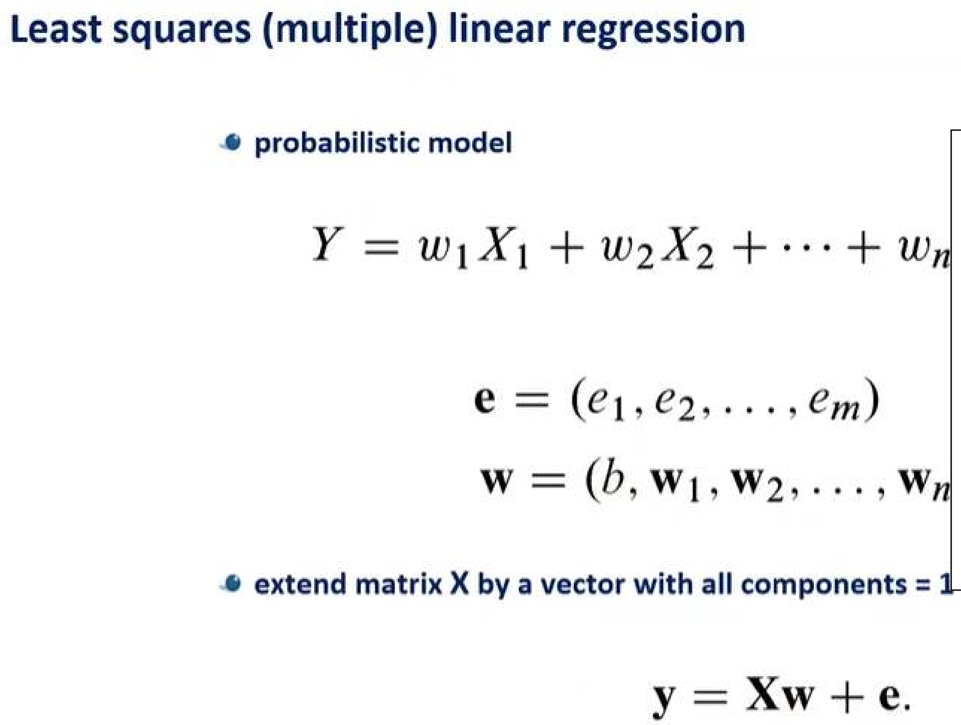
$$\hat{Y} = \hat{f}(X) = \hat{b} + \hat{w}X = \bar{\mu}_y + \frac{\sigma_{xy}}{\sigma_{xx}}(X - \bar{\mu}_x).$$

- alternatively imposing a cross at the origin

$$\hat{w} = \frac{\sum_{i=1}^{m} x_i y_i}{\sum_{i=1}^{m} x_i^2}.$$

$$\hat{b} = b = 0,$$

In the general case where we have multiple linear regression :



Least squares (multiple) linear regression

- probabilistic model

$$Y = w_1 X_1 + w_2 X_2 + \cdots + w_n$$

$$\mathbf{e} = (e_1, e_2, \ldots, e_m)$$

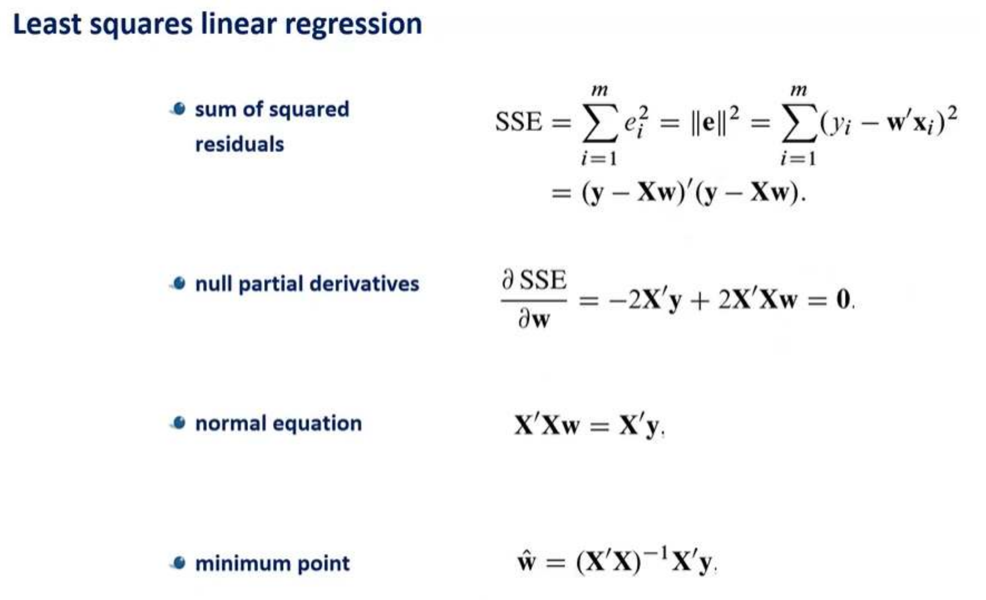
$$\mathbf{w} = (b, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$$

- extend matrix X by a vector with all components = 1

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e}.$$

The vector e is the vector of errors. In the vector w we also introduce b within its values. We extend X by a vector with 1s.

Then the problem is formulated as the following :



Least squares linear regression

- sum of squared residuals

$$\mathrm{SSE} = \sum_{i=1}^{m} e_i^2 = \|\mathbf{e}\|^2 = \sum_{i=1}^{m}(y_i - \mathbf{w}'\mathbf{x}_i)^2$$
$$= (\mathbf{y} - \mathbf{X}\mathbf{w})'(\mathbf{y} - \mathbf{X}\mathbf{w}).$$

- null partial derivatives

$$\frac{\partial\,\mathrm{SSE}}{\partial \mathbf{w}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\mathbf{w} = \mathbf{0}.$$

- normal equation

$$\mathbf{X}'\mathbf{X}\mathbf{w} = \mathbf{X}'\mathbf{y}.$$

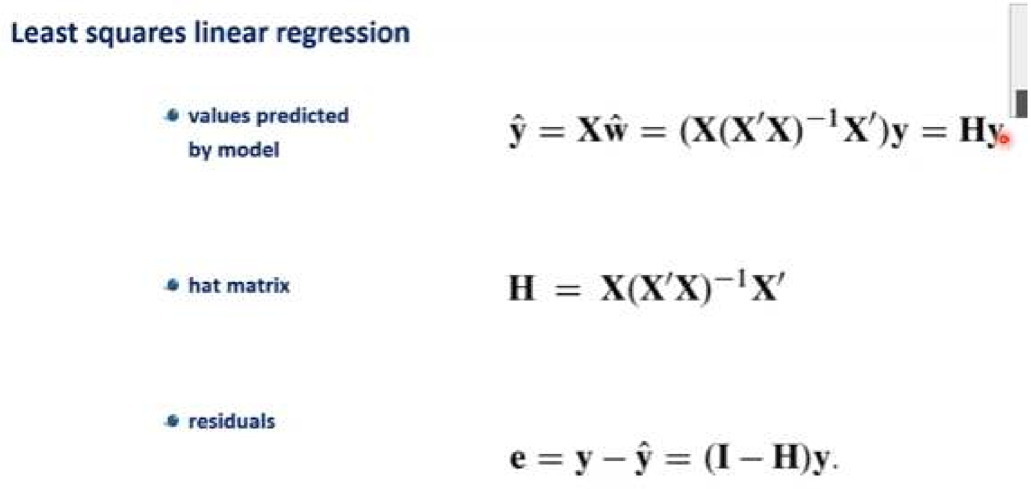
- minimum point

$$\hat{\mathbf{w}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

The last problem is ill-posed which mean a small change in the input result to a large change in the solution (the problem is not robust).

From the numerical POV inverting these metrics can be very problematic in some cases because the inverse is critical. Do we know exactly when this happens to be critical ? yes when the range (the difference between the largest eigenvalue and the smallest eigenvalue) in which the eigenvalues of these covariance is pretty high it means the eigenvalues are very spread much spread around the straight line then the calculation of inverse is very critical from the numerical point of view.

Once the previous values are calculated we can then predict new values by using :



Least squares linear regression

- values predicted by model

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}} = (\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{y} = \mathbf{H}\mathbf{y}$$

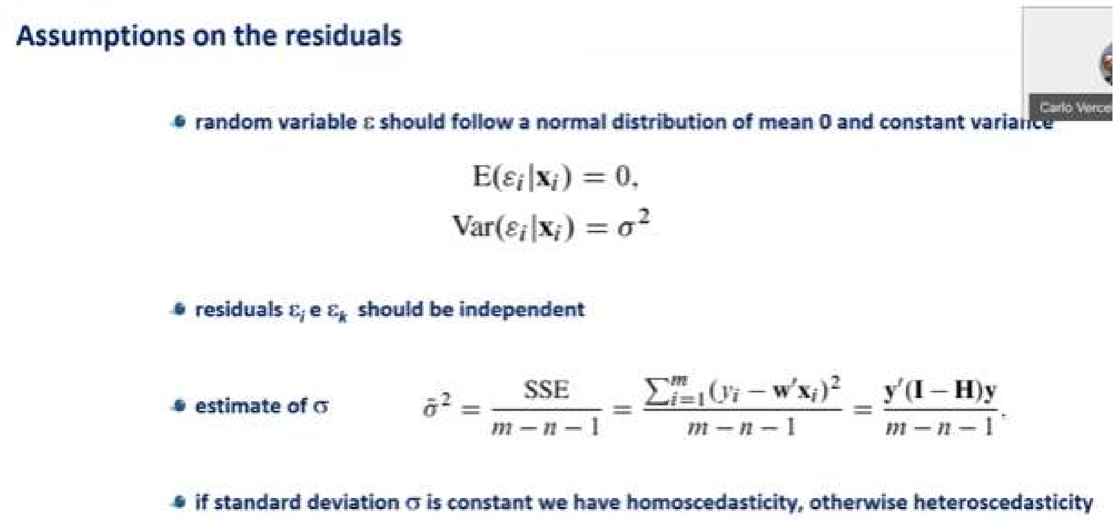
- hat matrix

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

- residuals

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}.$$

The assumptions we have on residuals :



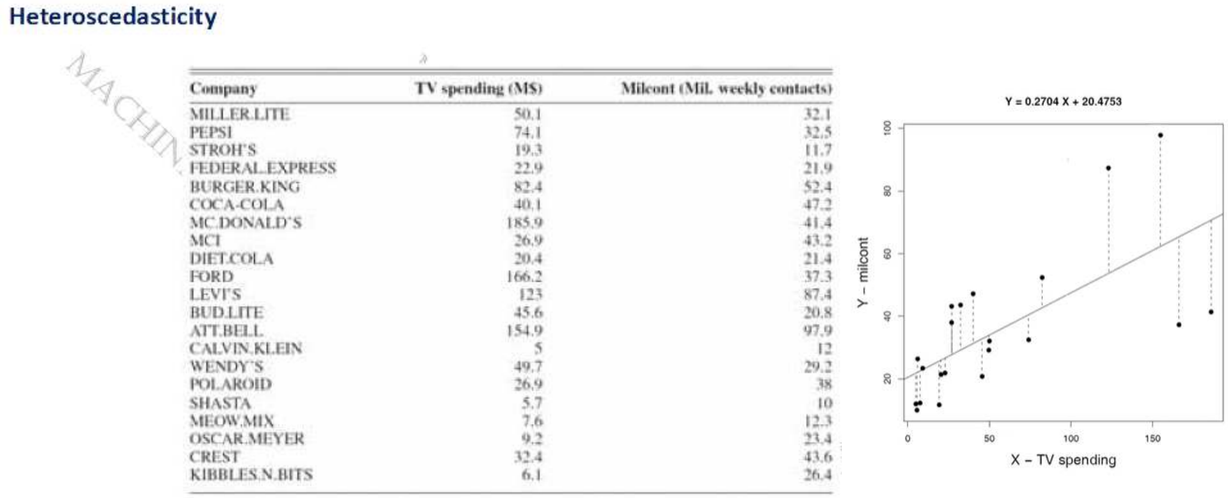
Assumptions on the residuals

- random variable ε should follow a normal distribution of mean 0 and constant variance

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

- residuals ε_j e ε_k should be independent
- estimate of σ $\quad \tilde{\sigma}^2 = \frac{\text{SSE}}{m-n-1} = \frac{\sum_{i=1}^{m}(y_i - \mathbf{w}'\mathbf{x}_i)^2}{m-n-1} = \frac{\mathbf{y}'(\mathbf{I}-\mathbf{H})\mathbf{y}}{m-n-1}$.
- if standard deviation σ is constant we have homoscedasticity, otherwise heteroscedasticity

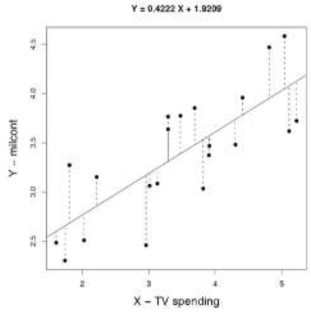
Heteroscedasticity example of Tv spending by brands :



Heteroscedasticity

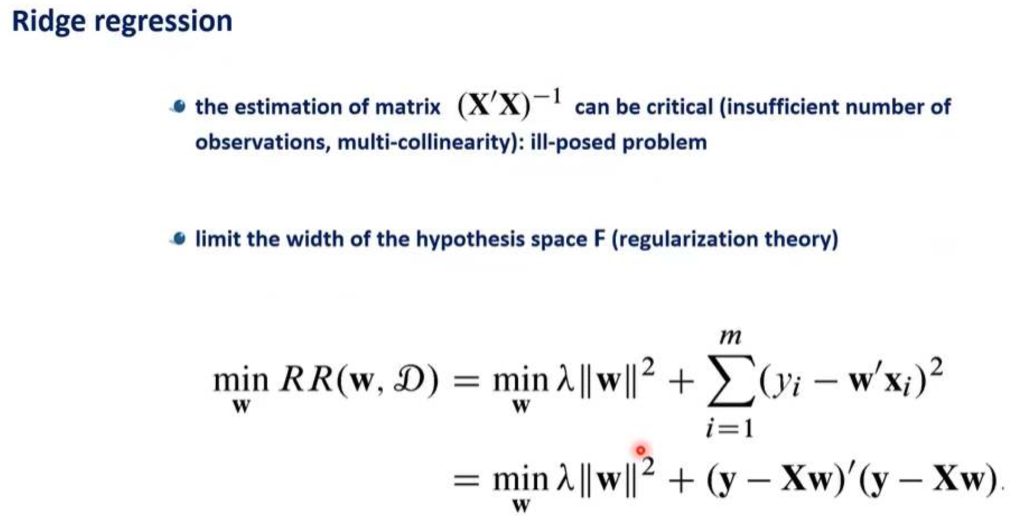
Company	TV spending (M$)	Milcont (Mil. weekly contacts)
MILLER.LITE	50.1	32.1
PEPSI	74.1	32.5
STROH'S	19.3	11.7
FEDERAL.EXPRESS	22.9	21.9
BURGER.KING	82.4	52.4
COCA-COLA	40.1	47.2
MC.DONALD'S	185.9	41.4
MCI	26.9	43.2
DIET.COLA	20.4	21.4
FORD	166.2	37.3
LEVI'S	123	87.4
BUD.LITE	45.6	20.8
ATT.BELL	154.9	97.9
CALVIN.KLEIN	5	12
WENDY'S	49.7	29.2
POLAROID	26.9	38
SHASTA	5.7	10
MEOW.MIX	7.6	12.3
OSCAR.MEYER	9.2	23.4
CREST	32.4	43.6
KIBBLES.N.BITS	6.1	26.4

We can see from the graph that in the lower part, the variance between values is small whereas in the higher part, the variance between the values is very high. We have small errors corresponding to small values of X and large errors corresponding to large values of X 5 this is very common). This situation is not acceptable because it violate one of the assumption, that's why for example we can use the Log in both axes :



So we can solve the Heteroscedasticity problem by going throw transformations in the axes

An alternative technic to linear regression appropriate to avoid overfitting :



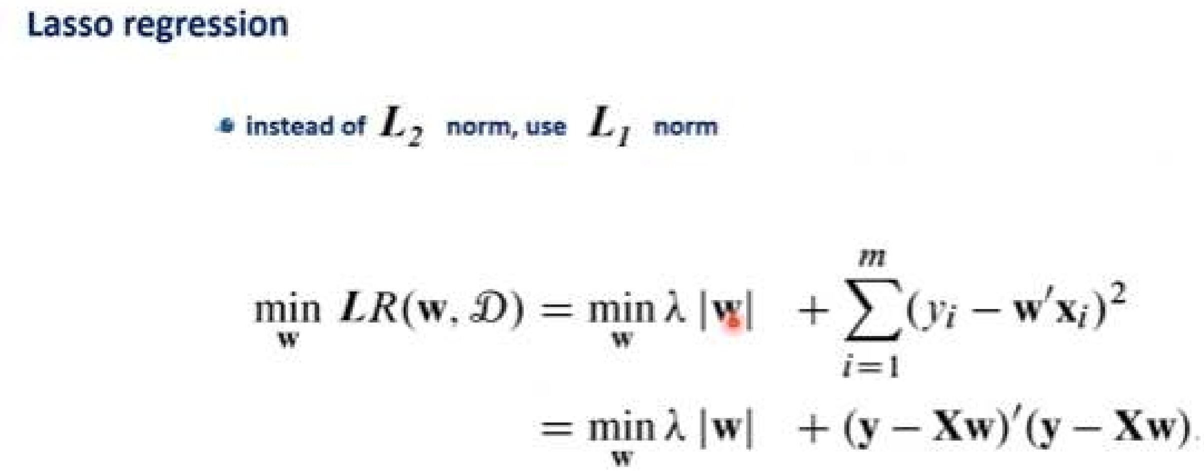
Ridge regression

- the estimation of matrix $(\mathbf{X}'\mathbf{X})^{-1}$ can be critical (insufficient number of observations, multi-collinearity): ill-posed problem
- limit the width of the hypothesis space F (regularization theory)

$$\begin{aligned}\min_{\mathbf{w}} RR(\mathbf{w}, \mathcal{D}) &= \min_{\mathbf{w}} \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{m} (y_i - \mathbf{w}'\mathbf{x}_i)^2 \\ &= \min_{\mathbf{w}} \lambda \|\mathbf{w}\|^2 + (\mathbf{y} - \mathbf{X}\mathbf{w})'(\mathbf{y} - \mathbf{X}\mathbf{w}).\end{aligned}$$

We introduce to the quadratic term the norm of W, the idea is similar to support vector machine. So we have the error of past data and we are taking into account the generalization capability by the quadratic term of the norm of w. The lambda plays the role of regulating the tradeoff between these two terms (accuracy and generalization).

We can introduce non differentiability by using L1 norm instead of L2 norm, this is what is called lasso regression :

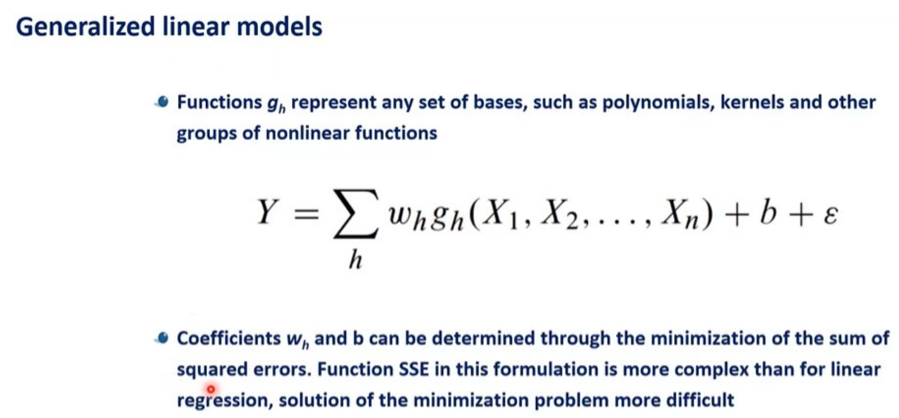


Lasso regression

- instead of L_2 norm, use L_1 norm

$$\begin{aligned}\min_{\mathbf{w}} LR(\mathbf{w}, \mathcal{D}) &= \min_{\mathbf{w}} \lambda \, |\mathbf{w}| \quad + \sum_{i=1}^{m} (y_i - \mathbf{w}'\mathbf{x}_i)^2 \\ &= \min_{\mathbf{w}} \lambda \, |\mathbf{w}| \quad + (\mathbf{y} - \mathbf{X}\mathbf{w})'(\mathbf{y} - \mathbf{X}\mathbf{w}).\end{aligned}$$

A third alternative is the Generalized linear models :



Generalized linear models

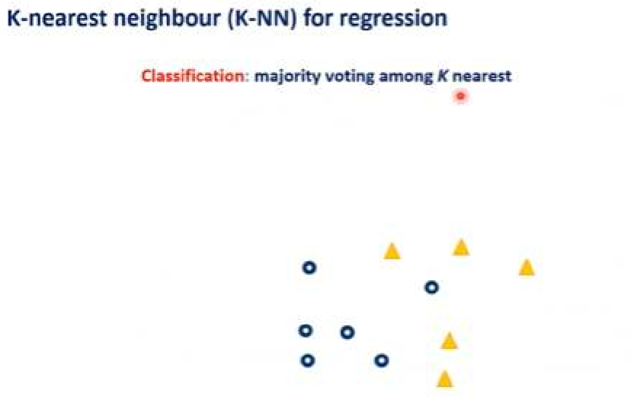
- Functions g_h represent any set of bases, such as polynomials, kernels and other groups of nonlinear functions

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

- Coefficients w_h and b can be determined through the minimization of the sum of squared errors. Function SSE in this formulation is more complex than for linear regression, solution of the minimization problem more difficult

The idea is to introduce non linear relationship by mean of this bases function in the space.

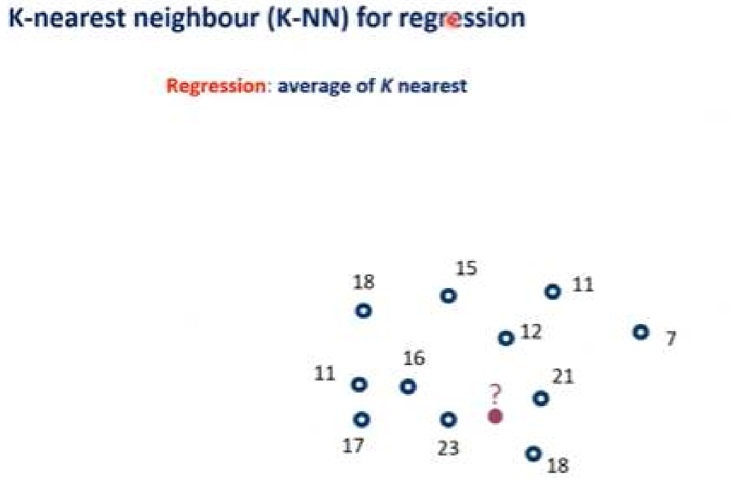
How to extend other technics used in the classification problems to regression :



K-nearest neighbour (K-NN) for regression

Classification: majority voting among K nearest

K-NN Heuristic method using the majority voting to determine the class and the number considering the majority is K.

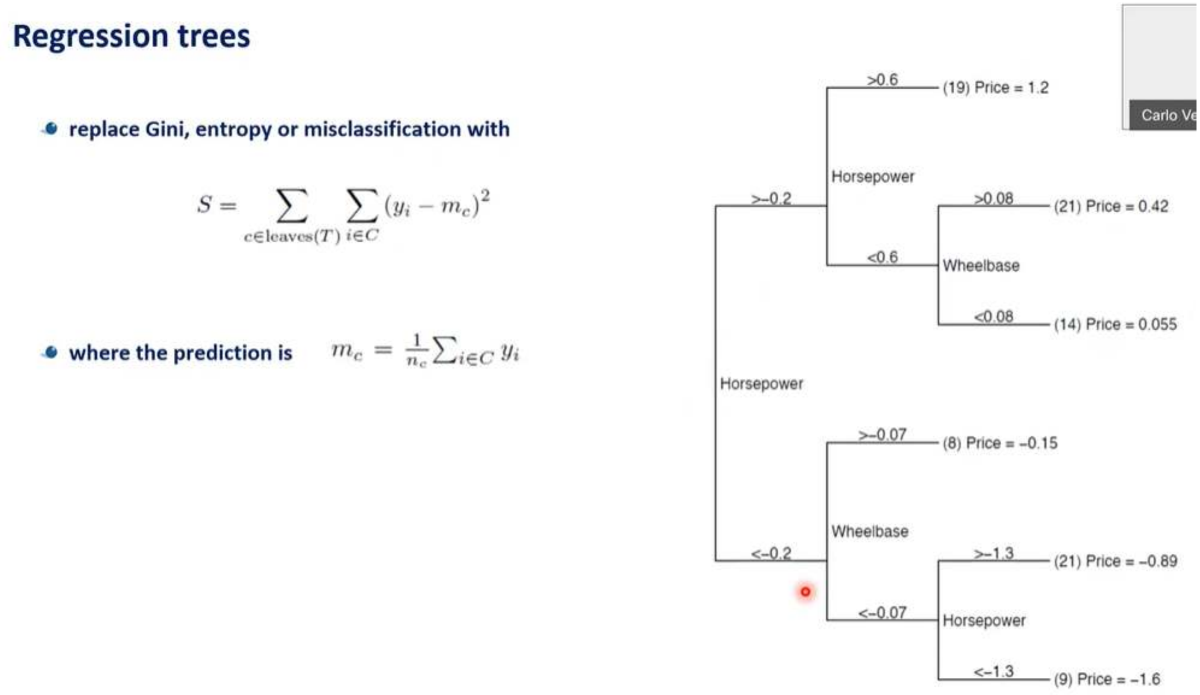


K-nearest neighbour (K-NN) for regression

Regression: average of K nearest

In the regression case we can use a centrale tendency (for example the mean) to determine the missing value relying on the K-neighbors.

We can also use regression trees in the same way :



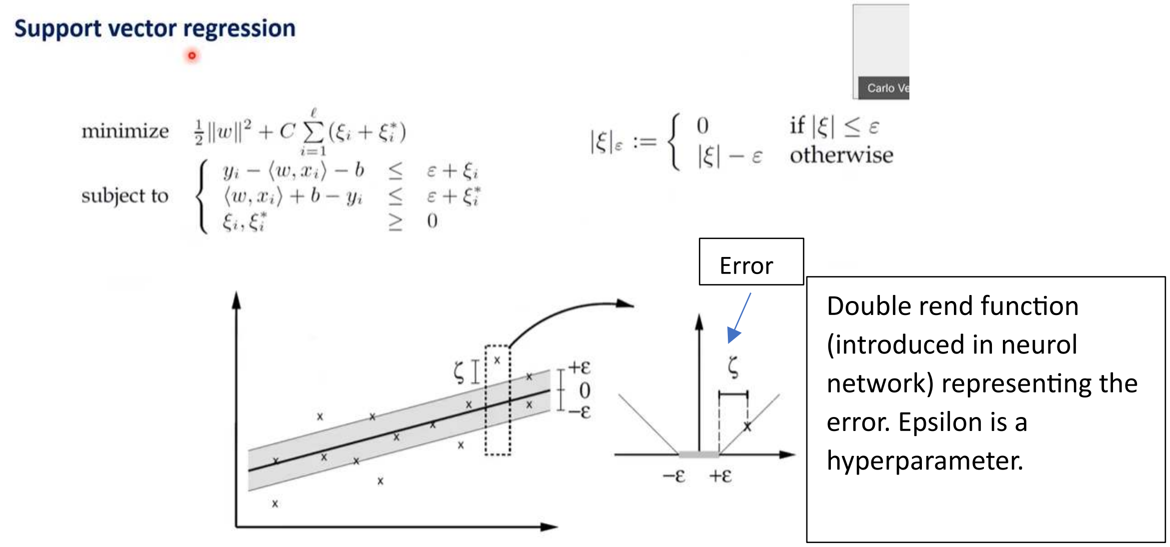
Regression trees

- replace Gini, entropy or misclassification with

$$S = \sum_{c \in \text{leaves}(T)} \sum_{i \in C} (y_i - m_c)^2$$

- where the prediction is $m_c = \frac{1}{n_c} \sum_{i \in C} y_i$

We are using the average of each leaf and we replace the Gini index and entropy by the variance. So we want to minimize the variance in each nod.



Support vector regression

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*)$$

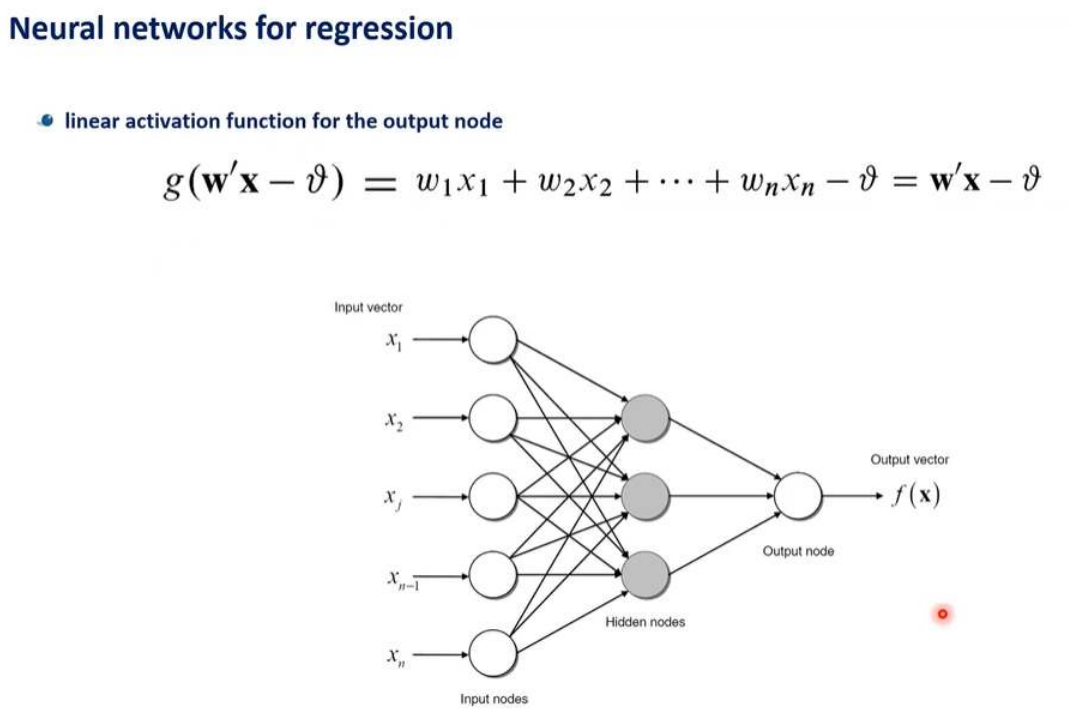
$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b & \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i & \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases}$$

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases}$$

Error

Double rend function (introduced in neurol network) representing the error. Epsilon is a hyperparameter.

In the minimization problem, we have the generalization term which is the norm of w and the coefficient C which is regulation the tradeoff between the two terms and finally the precision term.



Neural networks for regression

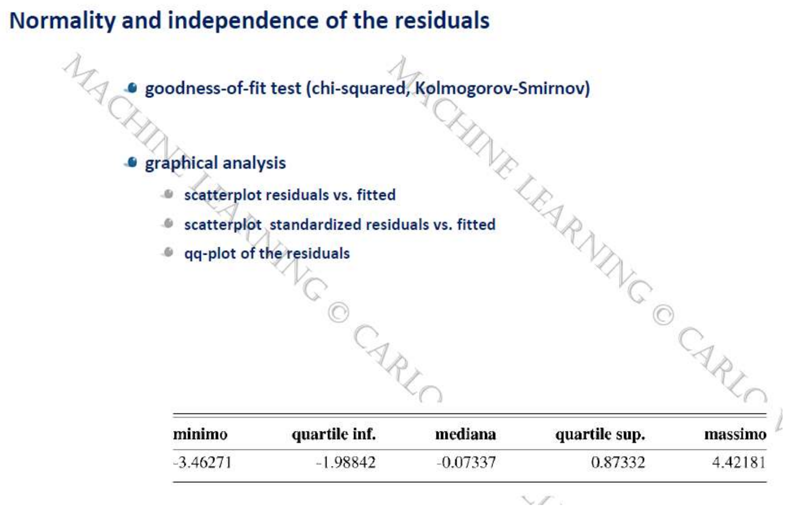
- linear activation function for the output node

$$g(\mathbf{w}'\mathbf{x} - \vartheta) = w_1x_1 + w_2x_2 + \cdots + w_nx_n - \vartheta = \mathbf{w}'\mathbf{x} - \vartheta$$

The only different is the activation function. In the case of classification it was either the activation function was -1,1 or the sigmoid function. In this case we use the W'x-v term. If we have a single variable the la last node is just one. The output nodes are as many as the number of the component of the vector to be predicted

Techniques to check our models :

https://politecnicomilano.webex.com/recordingservice/sites/politecnicomilano/recording/76a610e89c134a31af872b0da94c039a/playback



Normality and independence of the residuals

- goodness-of-fit test (chi-squared, Kolmogorov-Smirnov)
- graphical analysis
  - scatterplot residuals vs. fitted
  - scatterplot standardized residuals vs. fitted
  - qq-plot of the residuals

minimo	quartile inf.	mediana	quartile sup.	massimo
-3.46271	-1.98842	-0.07337	0.87332	4.42181

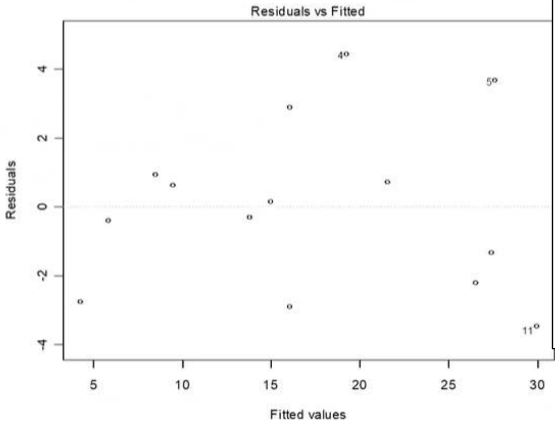
These techniques apply to all models and we check them after we apply the model.

Chi squared and Kormogorov are test to accept or reject the normality of residuals.

Before rejecting the model if it's not normally distributed we need to check the errors, this means that there is something that can be explained. (maybe you don't know much about explanatory factors, or maybe you are using the wrong space of hypotheses (linear explanation vs quadratic cubic. .)

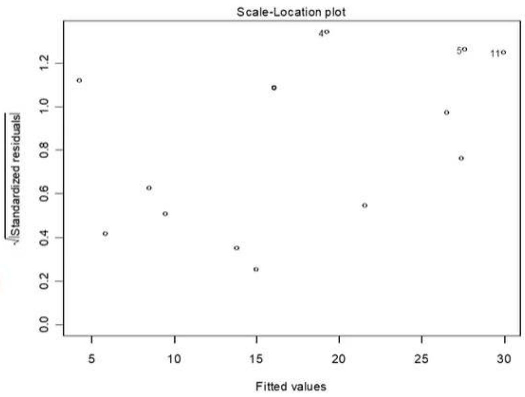
Graphical tools are complements to tests

Scatterplot residuals vs. fitted



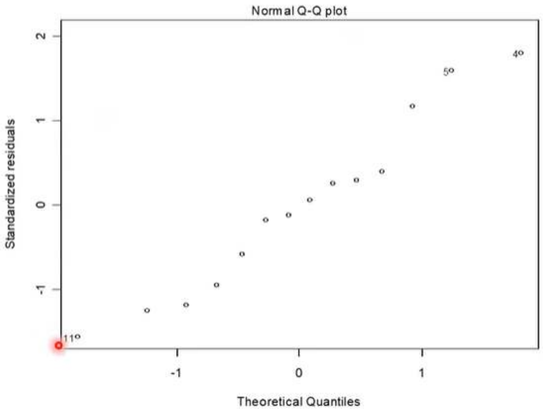
In this graph we plot the residuals vs fitted values, from this example we can say that the distribution of residuals appear randomly

Scatterplot standardized residuals vs. fitted

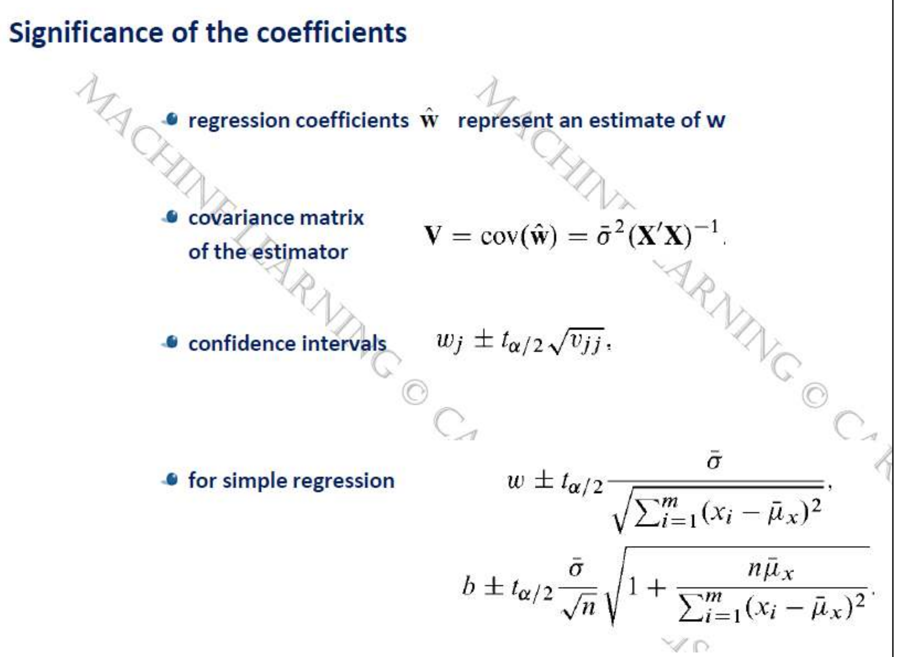


Standardized = (x-average)/variance

qq-plot of the residuals



Quantile-Quantile plot, in this case we can see that the distribution is more or less normally distributed



Significance of the coefficients

- regression coefficients $\hat{\mathbf{w}}$ represent an estimate of w
- covariance matrix of the estimator $\mathbf{V} = \mathrm{cov}(\hat{\mathbf{w}}) = \bar{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$.
- confidence intervals $w_j \pm t_{\alpha/2}\sqrt{v_{jj}}$,
- for simple regression

$$w \pm t_{\alpha/2}\frac{\bar{\sigma}}{\sqrt{\sum_{i=1}^{m}(x_i-\bar{\mu}_x)^2}},$$

$$b \pm t_{\alpha/2}\frac{\bar{\sigma}}{\sqrt{n}}\sqrt{1+\frac{n\bar{\mu}_x}{\sum_{i=1}^{m}(x_i-\bar{\mu}_x)^2}}.$$

Criterion meaningful only for least square regression since we speak about linear coefficient.

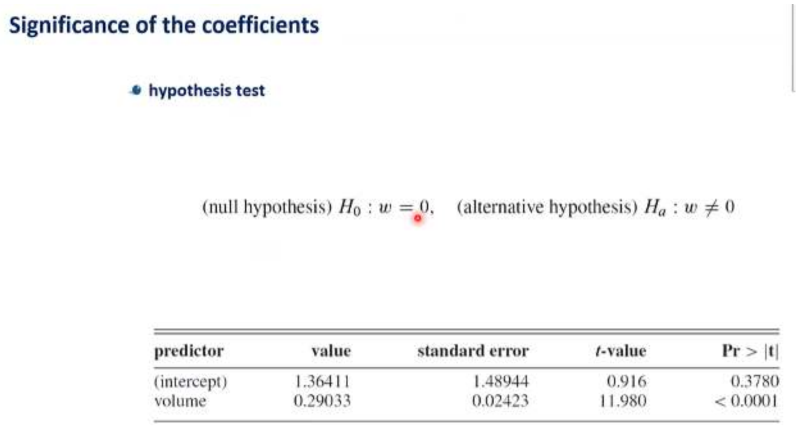
We are talking about estimation because we are analyzing only a sample.

So the qs is how good is this estimation.

We calculate the covariance matrix of the estimator and conclude the confidence intervals, the intervals are calculated along the diagonal of the matrix of variance

We are using t student quantile because we are assuming it could be a small sample.

We are using confidence intervals be cause the coefficients alone are not significant for example if the regression is Y=0.235x+20.45 the coefficient w=0.235 if it has an interval of confidence including 0 it means we can have negative tendency



Significance of the coefficients

- hypothesis test

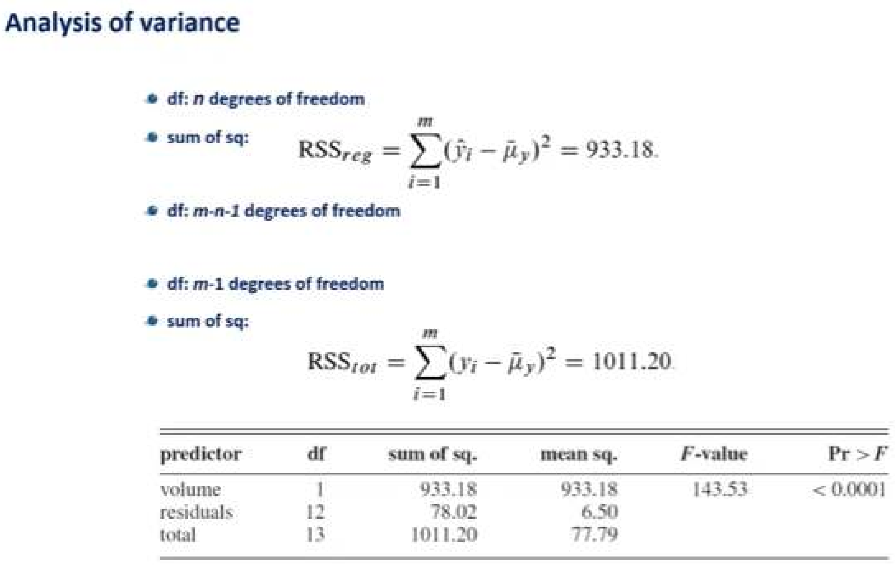
(null hypothesis) $H_0 : w = 0$, (alternative hypothesis) $H_a : w \neq 0$

predictor	value	standard error	t-value	Pr > |t|
(intercept)	1.36411	1.48944	0.916	0.3780
volume	0.29033	0.02423	11.980	< 0.0001

We have to guarantee that "w" does not include 0, it must be either all positive or all negative. This should be true in a multivariate analysis for all the coefficient.

Null hypotheses is the situation we want to guarantee that's not true. W=0 means there is no relationship between the independent and dependent variable.

We don't care a lot about the intercept, we focus more on the coefficient. The standard error 0.02 is very small. T-value is the value/standard error which means in this case that we need almost 12 standard error to move from the origin to the value which means it's an extreme outlier (very far from the tail) .and the probability to fall within that tail is less than 0.0001. to calculate the confidence interval : 0.29033+-2*0.02423, (2 is the value of student) in this case 0 is not part of the interval. An other way to do it is to compare t-value and 2, is it's lower than 2 or greater than -2 we have to reject the model. Or Pr > 0.05 third method.



Analysis of variance

- df: *n* degrees of freedom
- sum of sq: $$\mathrm{RSS}_{reg} = \sum_{i=1}^{m} (\hat{y}_i - \bar{\mu}_y)^2 = 933.18.$$
- df: *m-n-1* degrees of freedom
- df: *m*-1 degrees of freedom
- sum of sq: $$\mathrm{RSS}_{tot} = \sum_{i=1}^{m} (y_i - \bar{\mu}_y)^2 = 1011.20.$$

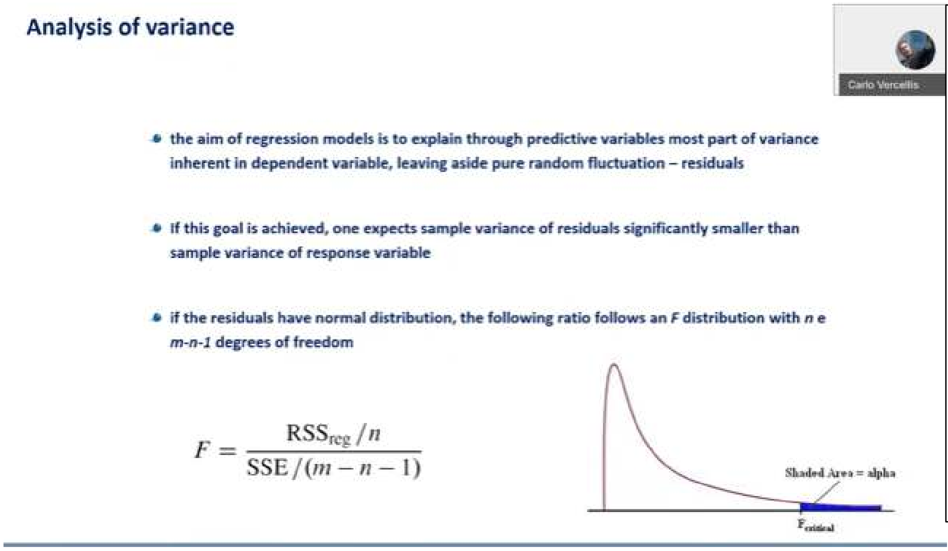
predictor	df	sum of sq.	mean sq.	*F*-value	Pr > *F*
volume	1	933.18	933.18	143.53	< 0.0001
residuals	12	78.02	6.50		
total	13	1011.20	77.79		

The analysis of variance is based on the F-value.

df is the degree of freedom. For volume we have one independent variable and for the residuals we have 12.

The mean sq = sum sq/df

Here we calculate the variance related to the regression RSSreg, and the variance related to the actual values RSStot



Analysis of variance

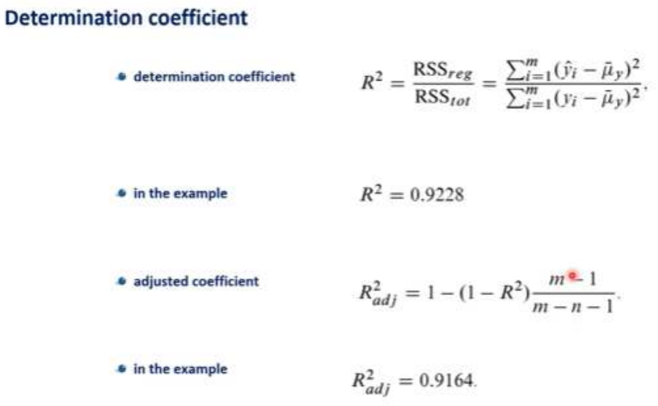
- the aim of regression models is to explain through predictive variables most part of variance inherent in dependent variable, leaving aside pure random fluctuation – residuals
- If this goal is achieved, one expects sample variance of residuals significantly smaller than sample variance of response variable
- if the residuals have normal distribution, the following ratio follows an *F* distribution with *n* e *m-n-1* degrees of freedom

$$F = \frac{\mathrm{RSS}_{\mathrm{reg}}/n}{\mathrm{SSE}/(m-n-1)}$$

The F distribution is one tailed distribution more adapted to describing random variables that are positive.

We calculate the F-value and compare it to the critical value alpha. If the value is greater than 0.05 the test is failed (the area is going to the left of the critical value

Now we move to a criterium applied to all type of technics

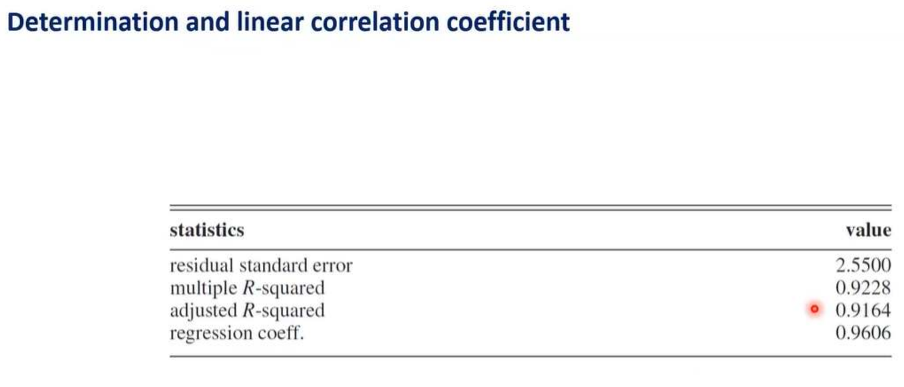


Determination coefficient

- determination coefficient $$R^2 = \frac{\mathrm{RSS}_{reg}}{\mathrm{RSS}_{tot}} = \frac{\sum_{i=1}^{m}(\hat{y}_i - \bar{\mu}_y)^2}{\sum_{i=1}^{m}(y_i - \bar{\mu}_y)^2}.$$
- in the example $$R^2 = 0.9228$$
- adjusted coefficient $$R^2_{adj} = 1 - (1 - R^2)\frac{m-1}{m-n-1}.$$
- in the example $$R^2_{adj} = 0.9164.$$

For R² : The closer to 1 the better

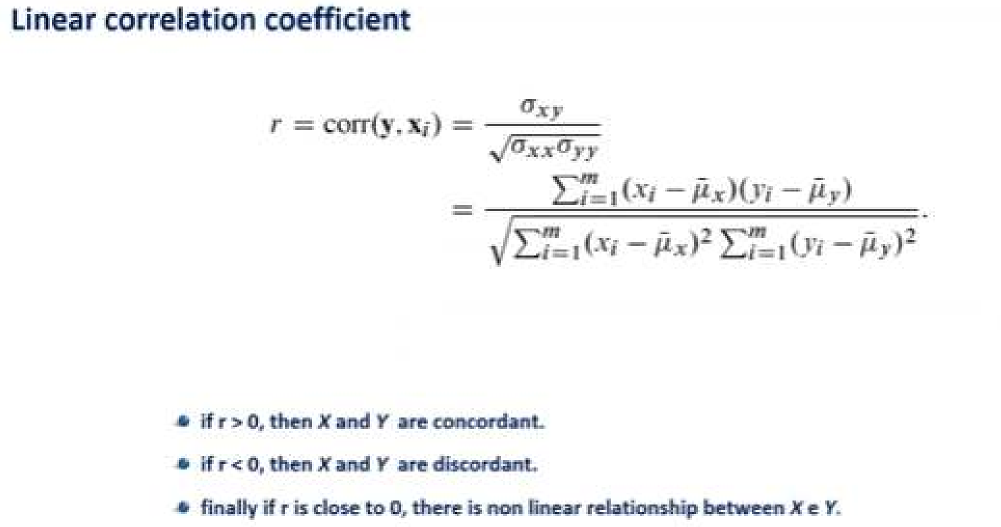
The adjustment is done to take into consideration the degree of freedom. This is done to also avoid overfitting in small samples. If we have a lot of observations m>>>n the adjustment has no effect



Determination and linear correlation coefficient

statistics	value
residual standard error	2.5500
multiple R-squared	0.9228
adjusted R-squared	0.9164
regression coeff.	0.9606

The regression coefficient is the square root of R-squared



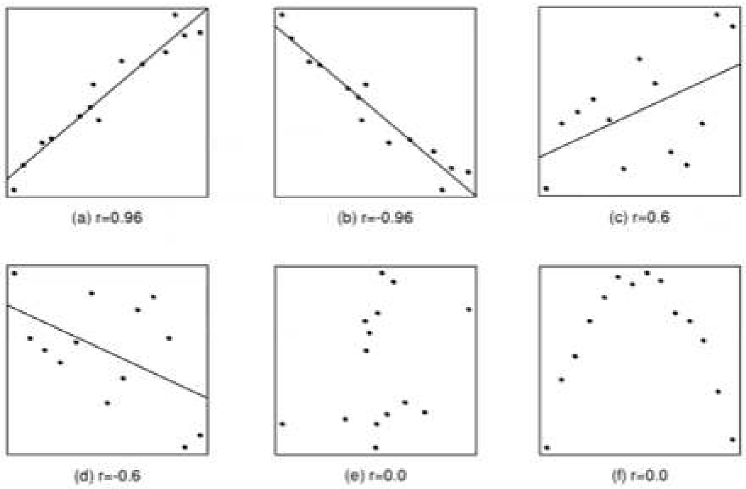
Linear correlation coefficient

$$r = \mathrm{corr}(\mathbf{y}, \mathbf{x}_i) = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}\sigma_{yy}}}$$
$$= \frac{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)(y_i - \bar{\mu}_y)}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2 \sum_{i=1}^{m}(y_i - \bar{\mu}_y)^2}}.$$

- if r > 0, then *X* and *Y* are concordant.
- if r < 0, then *X* and *Y* are discordant.
- finally if r is close to 0, there is non linear relationship between *X* e *Y*.

Example :

Linear correlation coefficient



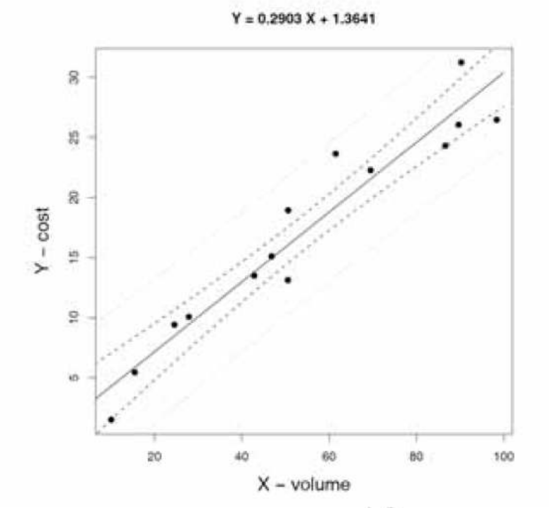
In the last graph r=0 means there is not a linear regression. But in that case we have another type of correlation

Multi-collinearity of the independent variables

- variance inflation factor:
  values greater than 5 point to
  the existence of multi-collinearity

$$\mathrm{VIF}_j = \frac{1}{1 - R_j^2}$$

Confidence and prediction limits



Confidence and prediction limits

- prediction associated to a new observation $\hat{y} = \hat{\mathbf{w}}\mathbf{x}$
- its variance is $\mathrm{Var}(\hat{y}) = \mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}\bar{\sigma}^2,$
- for simple regression the confidence limit for E[*Y*] is

$$\hat{y} \pm t_{\alpha/2}\bar{\sigma}\sqrt{\frac{1}{m} + \frac{x - \bar{\mu}_x}{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2}},$$

- whereas the prediction interval for *Y* is

$$\hat{y} \pm t_{\alpha/2}\bar{\sigma}\sqrt{1 + \frac{1}{m} + \frac{x - \bar{\mu}_x}{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2}}.$$