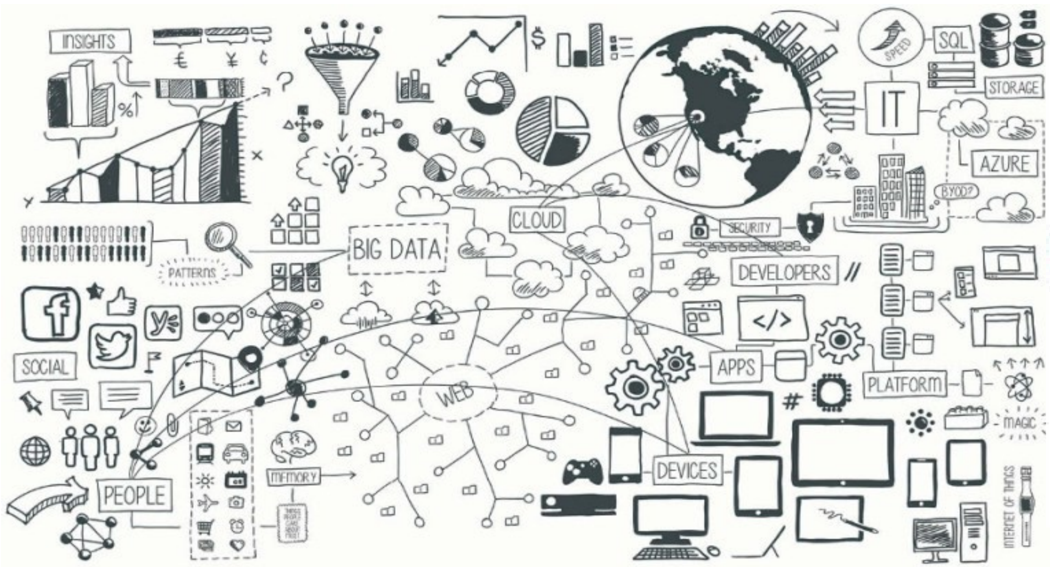
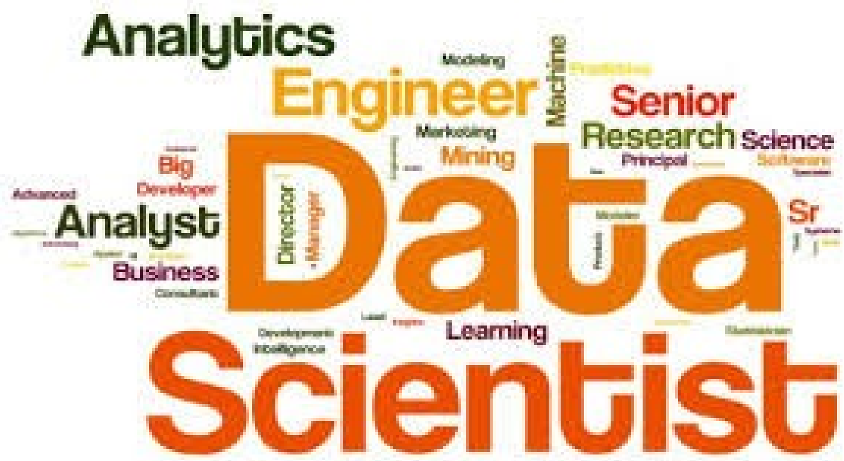
Data Mining (Minería de Datos)

Evaluación, sobreajuste y validación cruzada (cross-validation)





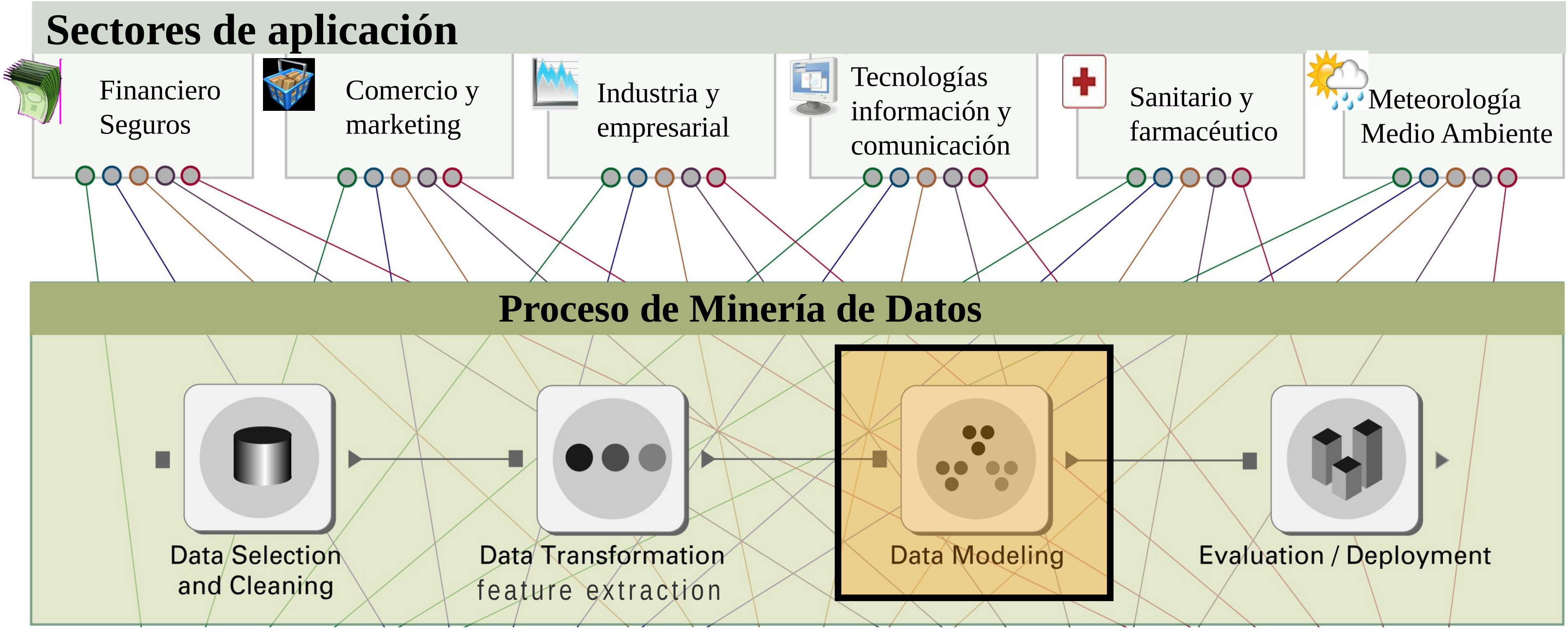
Sixto Herrera

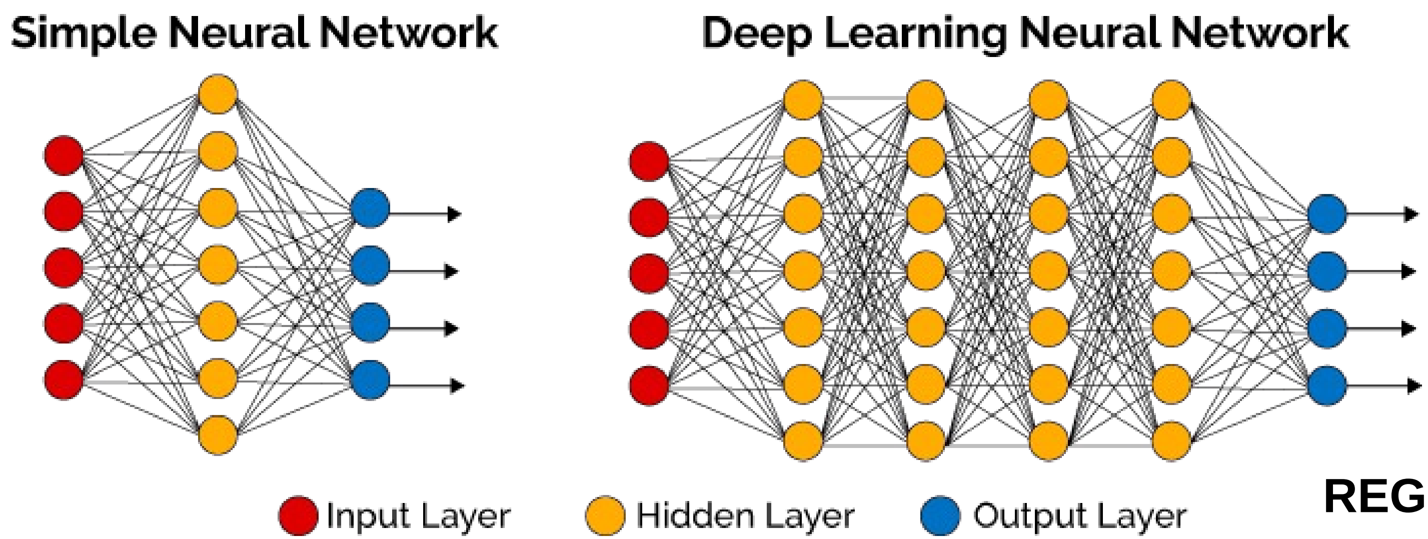
Grupo de Meteorología

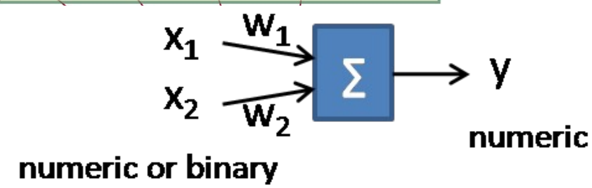
Univ. de Cantabria – CSIC
MACC / IFCA

Oct	29	Aplazada (sesión de refuezo)
	31	Presentación, introducción y perspectiva histórica
Nov	5	Paradigmas, problemas canonicos y data challenges
	7	Reglas de asociación
	12	Práctica: Reglas de asociación
	14	**Evaluación, sobrejuste y crossvalidacion**
	19	**Práctica: Cross-validación**
	21	Árboles de clasificacion y decision
	26	Practica: Árboles de clasificación
		T01. Datos discretos
	28	Técnicas de vecinos cercano (k-NN)
Dic	3	Práctica: Vecinos cercanos
	5	Reducción de dimensión no lineal
	10	Práctica: Reducción de dimensión no lineal
		T02. Clasificación
	12	Árboles de clasificación y regresion (CART)
	17	Práctica: Árboles de clasificación y regresion (CART)
	19	Ensembles: Bagging and Boosting
Ene	7	Práctica Random Forests
	9	Práctica Gradient boosting
		T03. Prediccion
	14	Técnicas de agrupamiento
	16	Practica: Técnicas de agrupamiento
	21a	Practica: El paquete CARET
	21b	Examen

NOTA: Las líneas de código de R en esta presentación se muestran sobre un fondo gris.





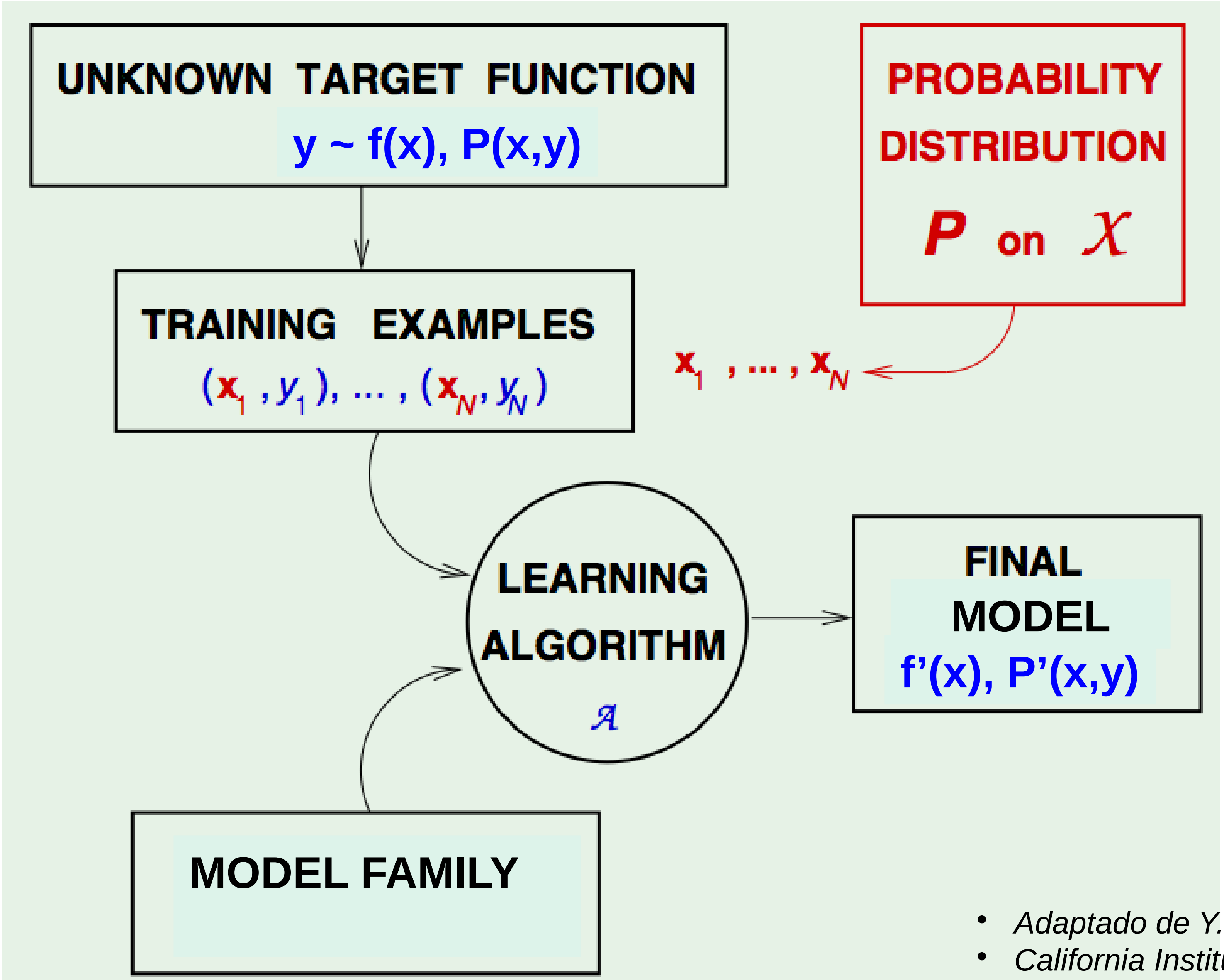


$$y = w_0 + w_1x_1 + w_2x_2$$

$$y = f(\mathbf{X},\mathbf{W}) = \mathbf{X}^T.\mathbf{W}$$

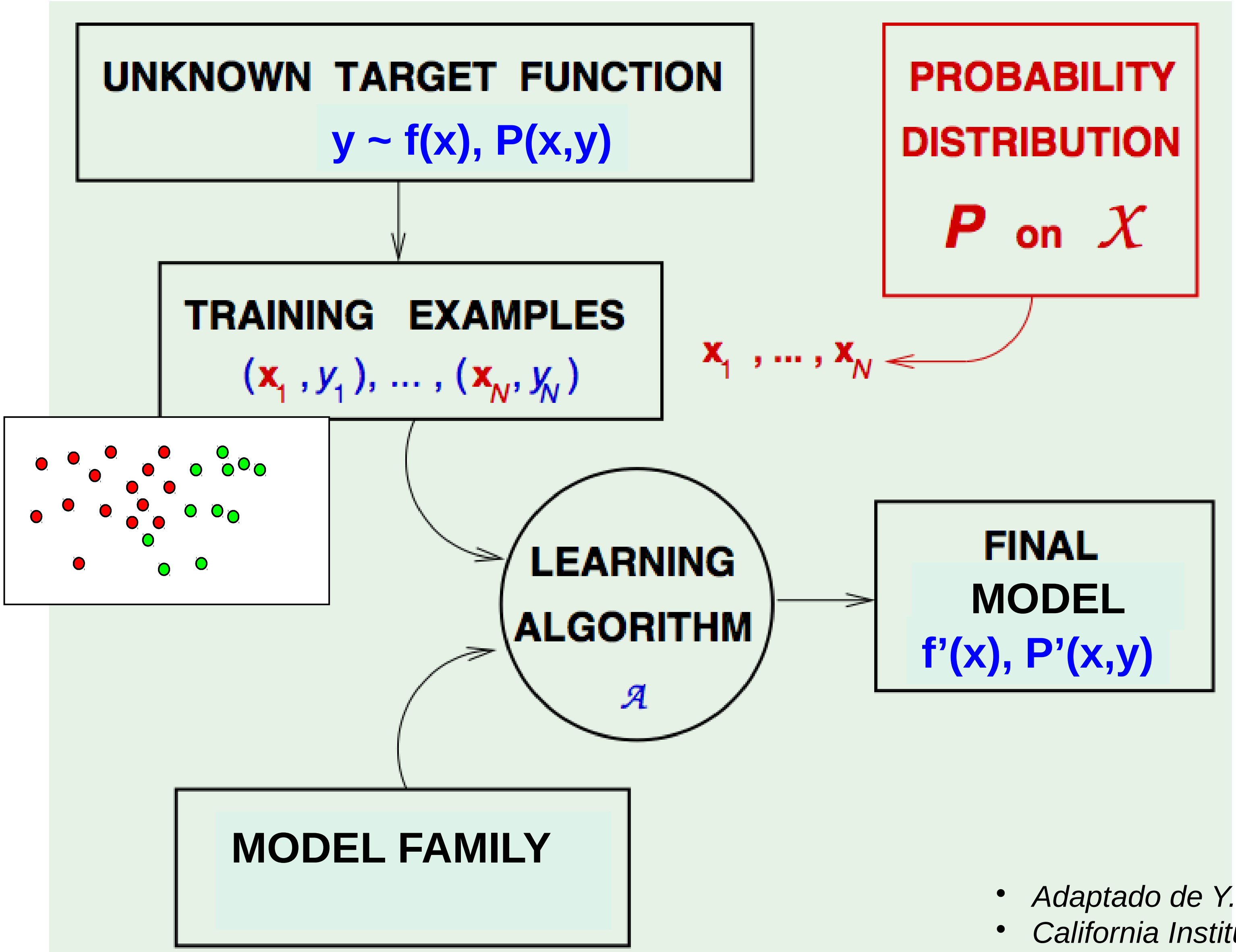
REGRESSION $\mathbf{W} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$ $\mathbf{X} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$

Learning is the automatic process of building (adjusting) a model from a data set which is representative from the full population.



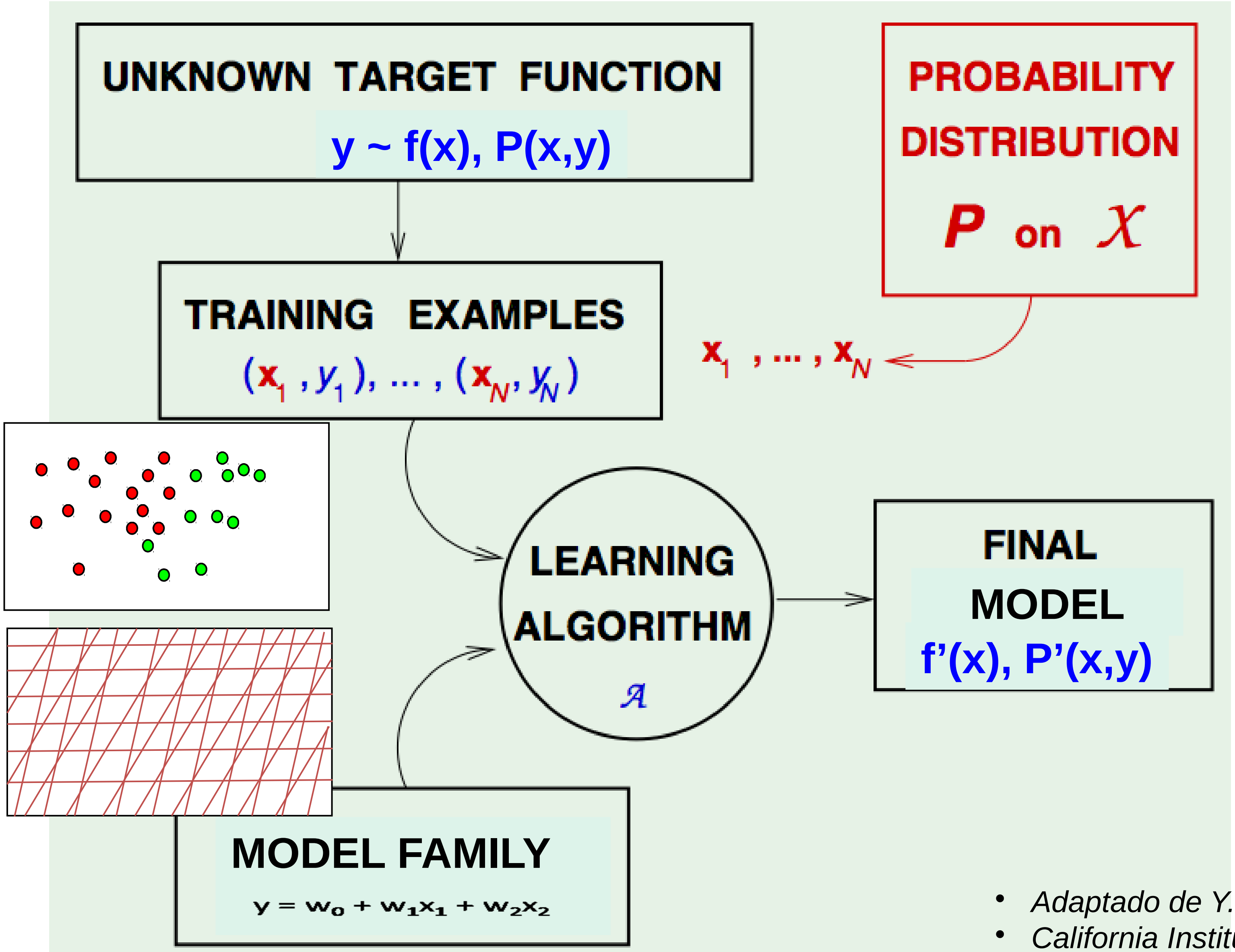
- *Adaptado de Y.S. Abu-Mostafa*
- *California Institute of Technology*

Learning is the automatic process of building (adjusting) a model from a data set which is representative from the full population.



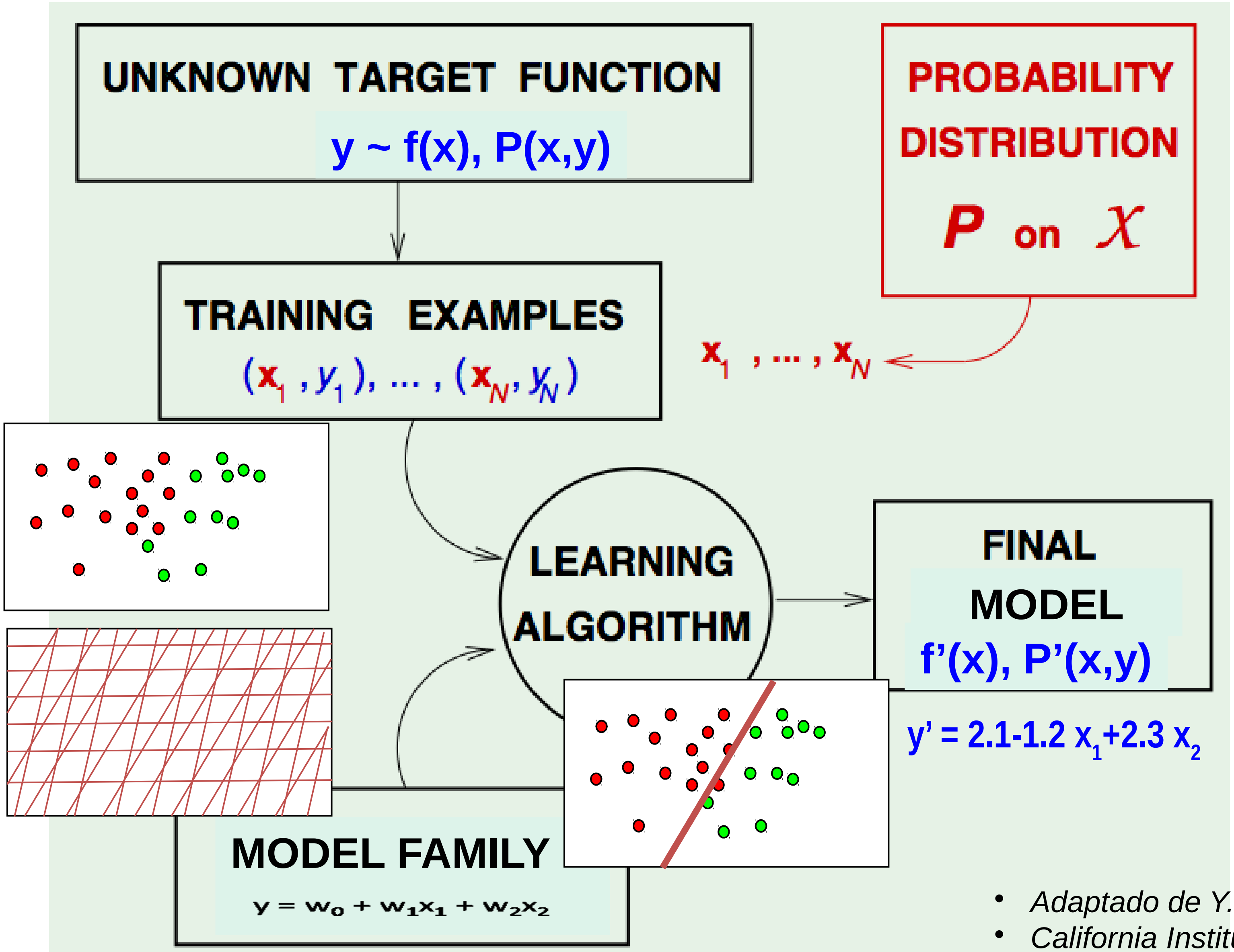
- *Adaptado de Y.S. Abu-Mostafa*
- *California Institute of Technology*

Learning is the automatic process of building (adjusting) a model from a data set which is representative from the full population.



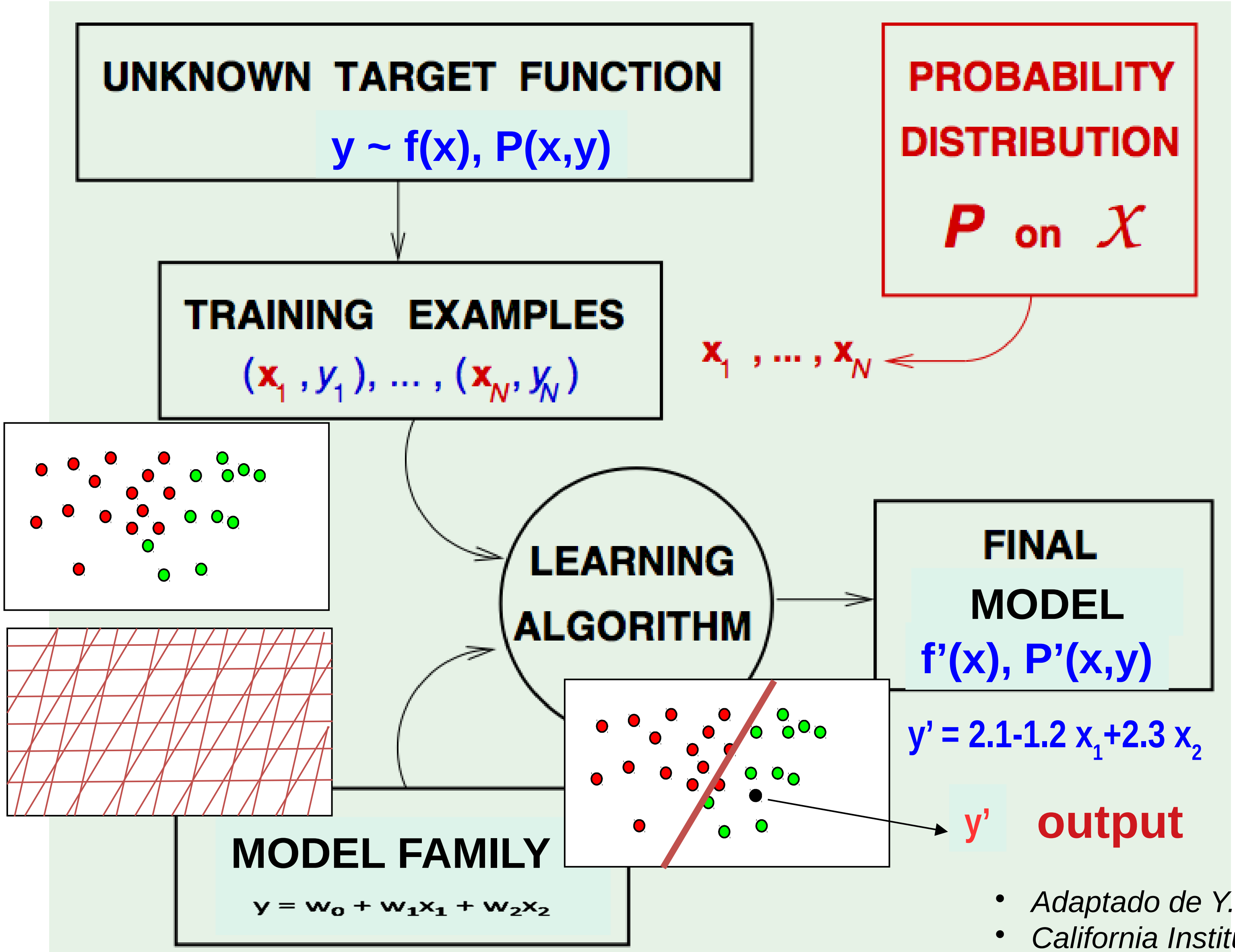
- *Adaptado de Y.S. Abu-Mostafa*
- *California Institute of Technology*

Learning is the automatic process of building (adjusting) a model from a data set which is representative from the full population.



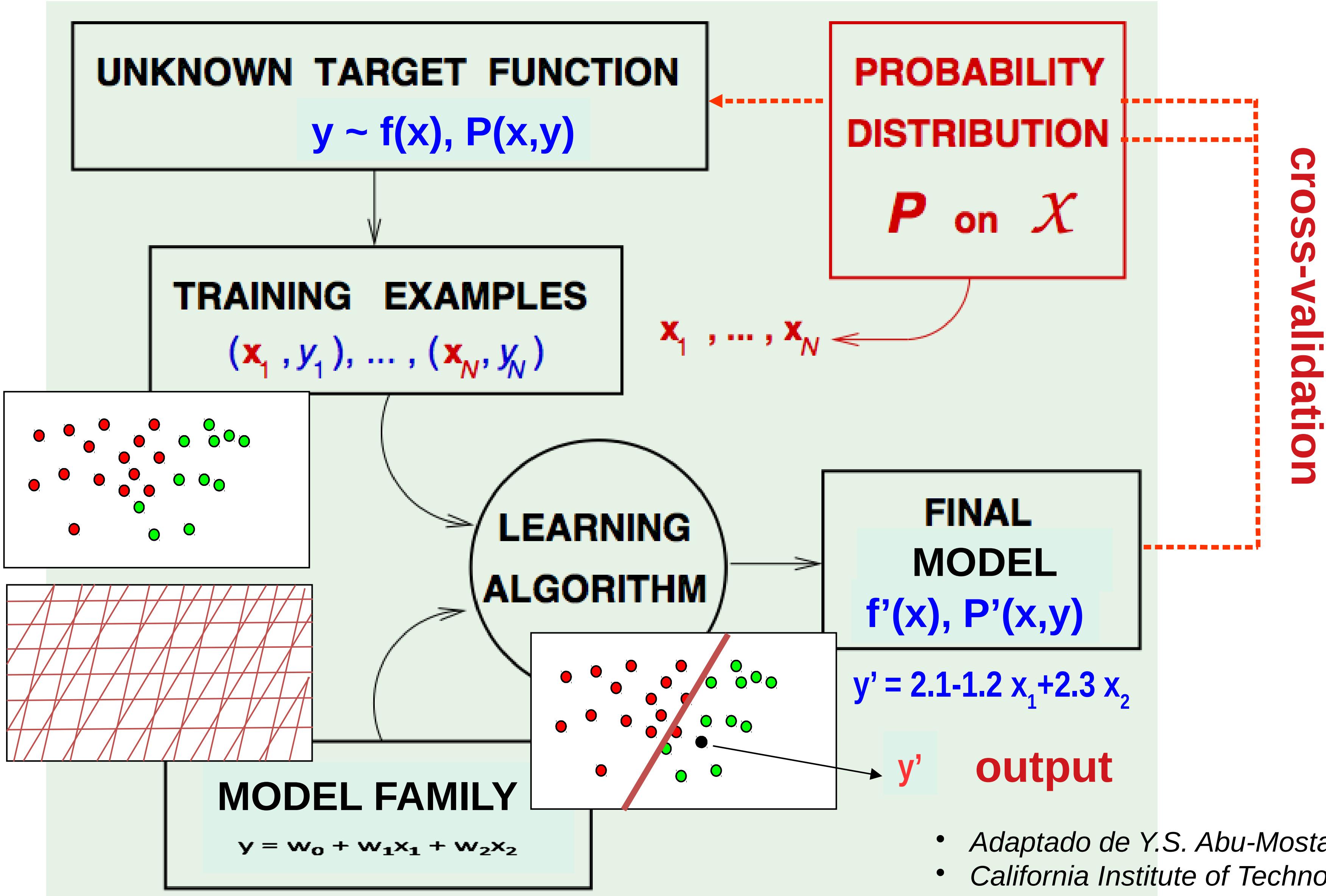
- *Adaptado de Y.S. Abu-Mostafa*
- *California Institute of Technology*

Learning is the automatic process of building (adjusting) a model from a data set which is representative from the full population.



- *Adaptado de Y.S. Abu-Mostafa*
- *California Institute of Technology*

Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).



- *Adaptado de Y.S. Abu-Mostafa*
- *California Institute of Technology*

UNKNOWN TARGET FUNCTION

$y \sim f(x), P(x,y)$

Finite/Incomplete sample

TRAINING EXAMPLES

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

PROBABILITY DISTRIBUTION

P on $\mathcal{X}$

$\mathbf{x}_1, \ldots, \mathbf{x}_N$

LEARNING ALGORITHM

$\mathcal{A}$

FINAL MODEL

$f'(x), P'(x,y)$

y' $P'(y|x)$

OUTPUT

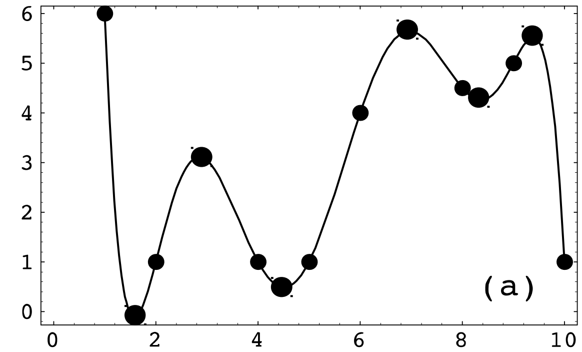
MODEL FAMILY

$\mathcal{H}$

Adaptado de Y.S. Abu-Mostafa
California Institute of Technology

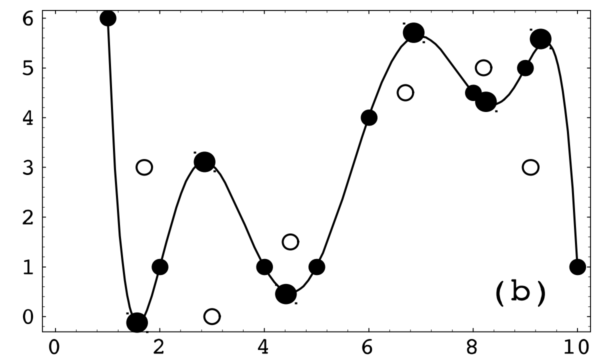
Model complexity <-> degrees of freedom

$$y = \sum_{i=0}^{N} (a_i * x^i)$$



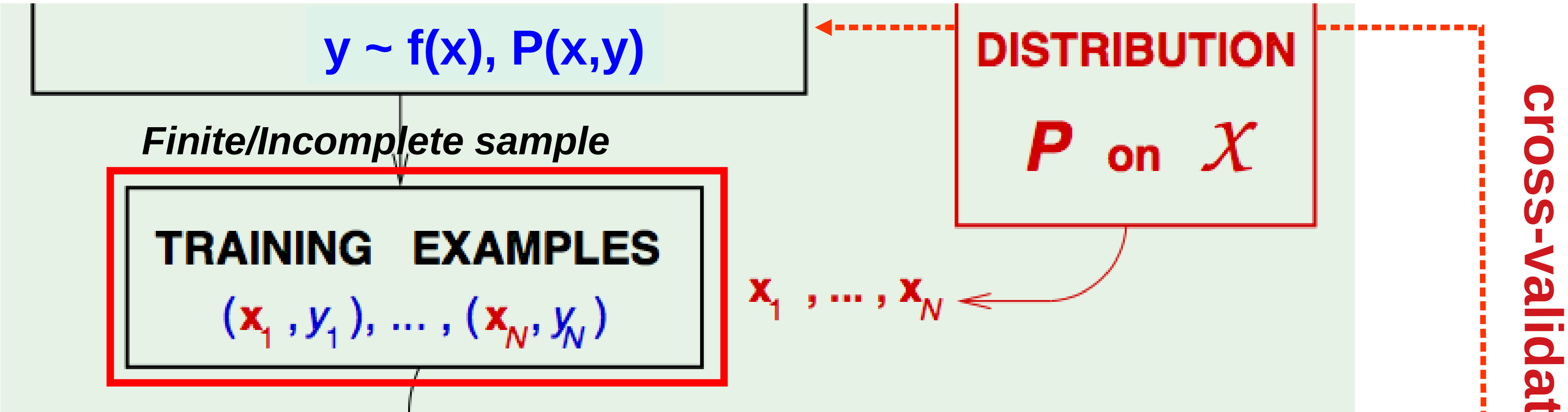
New sample

Overfitting

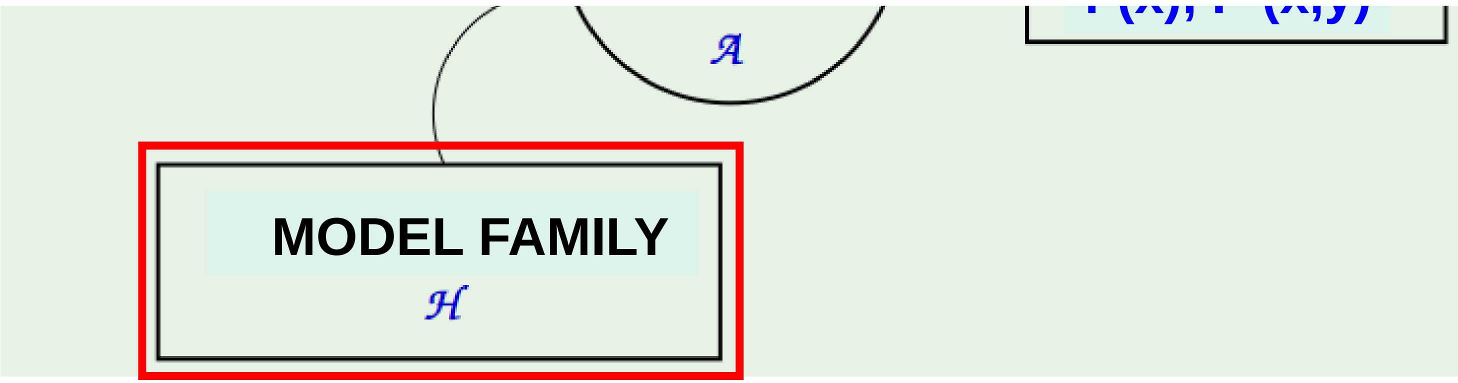


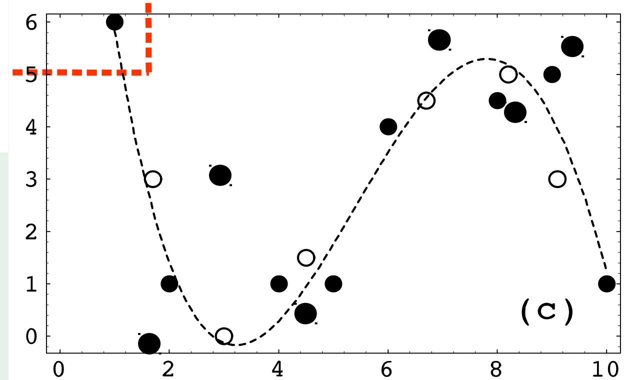
Trade-off between bias and variance

Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).



Simplest models have better generalization properties and avoid the **overfitting** removing parameters/degree of freedom.





Trade-off between bias and variance

Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).

1. Can we make sure that $E_{\mathrm{out}}(g)$ is close enough to $E_{\mathrm{in}}(g)$?

1

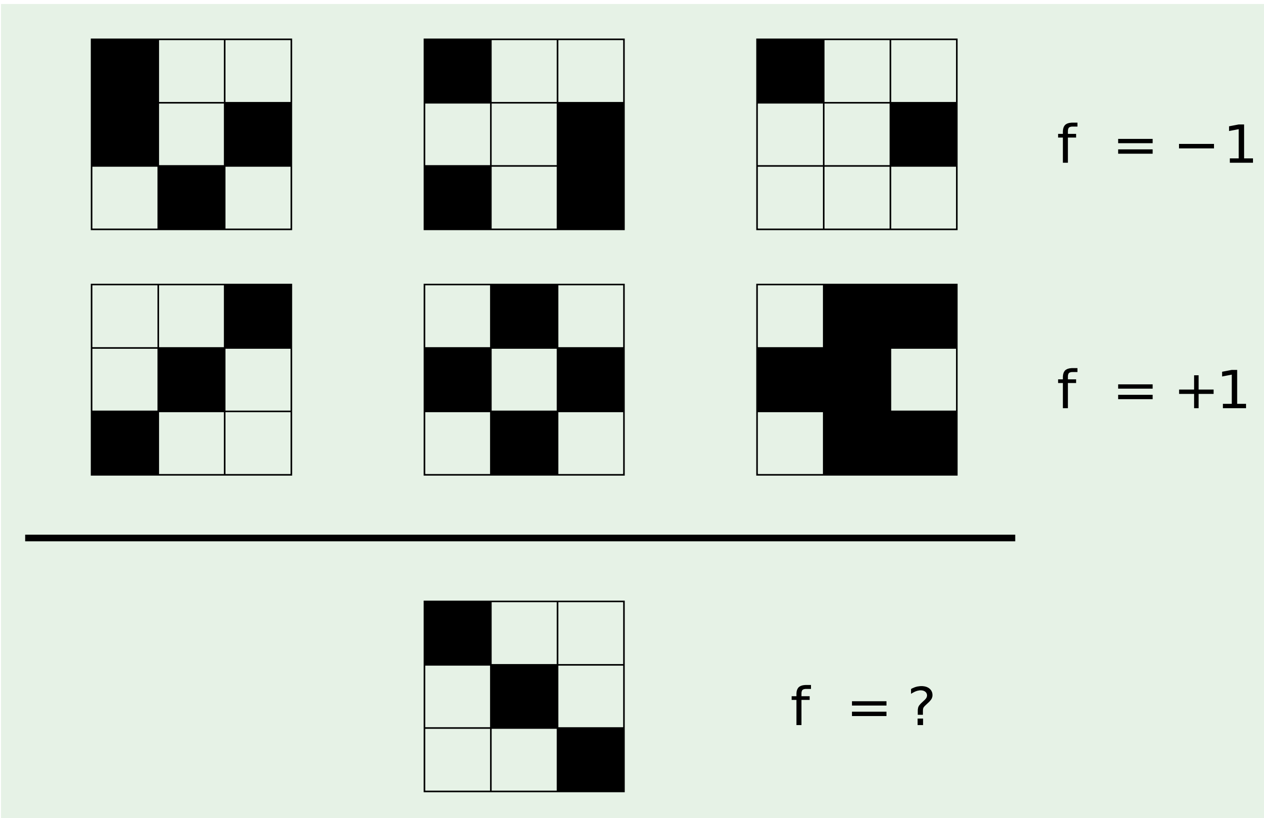
2. Can we make $E_{\mathrm{in}}(g)$ small enough?

Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).

1. Can we make sure that $E_{out}(g)$ is close enough to $E_{in}(g)$?
   1
2. Can we make $E_{in}(g)$ small enough?

The (***in-sample***) error is the unique which can be estimated:

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$



		f	
		+1	−1
h	+1	no error	*false accept*
	−1	*false reject*	no error

$$E_{out}(h)=E(f,h)$$

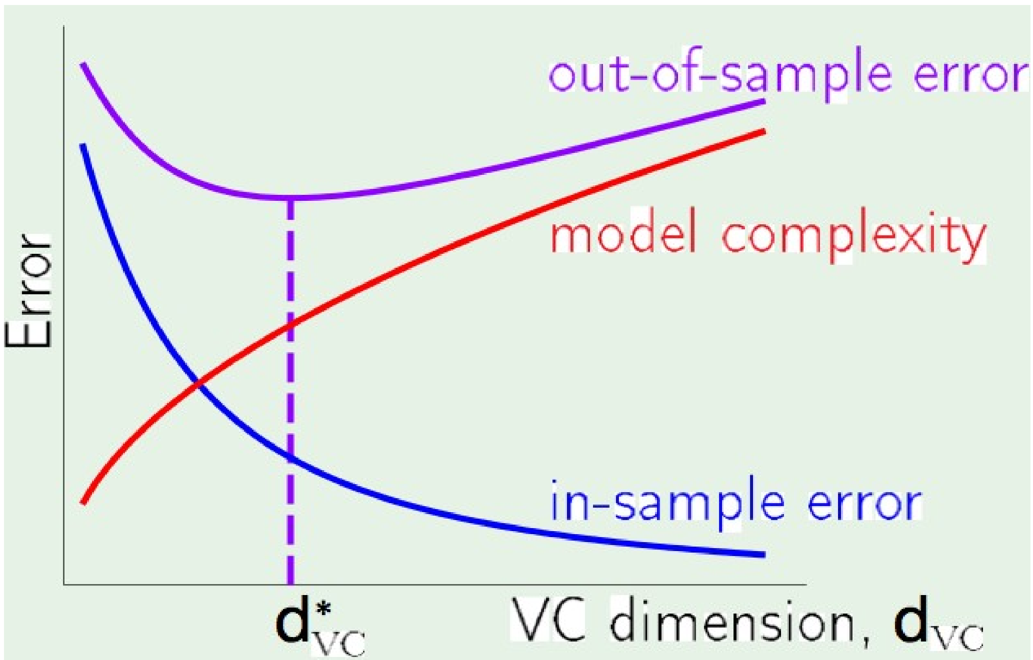
Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).

1. Can we make sure that $E_{out}(g)$ is close enough to $E_{in}(g)$?
1
2. Can we make $E_{in}(g)$ small enough?

The (***in-sample***) error is the unique which can be estimated:

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

Vapnik-Chervonenkis (VC) Dimension



		f	
		+1	−1
h	+1	no error	*false accept*
	−1	*false reject*	no error

$$E_{out}(h)=E(f,h)$$

Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).

1. Can we make sure that $E_{out}(g)$ is close enough to $E_{in}(g)$?
2. Can we make $E_{in}(g)$ small enough?

The (***in-sample***) error is the unique which can be estimated:

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

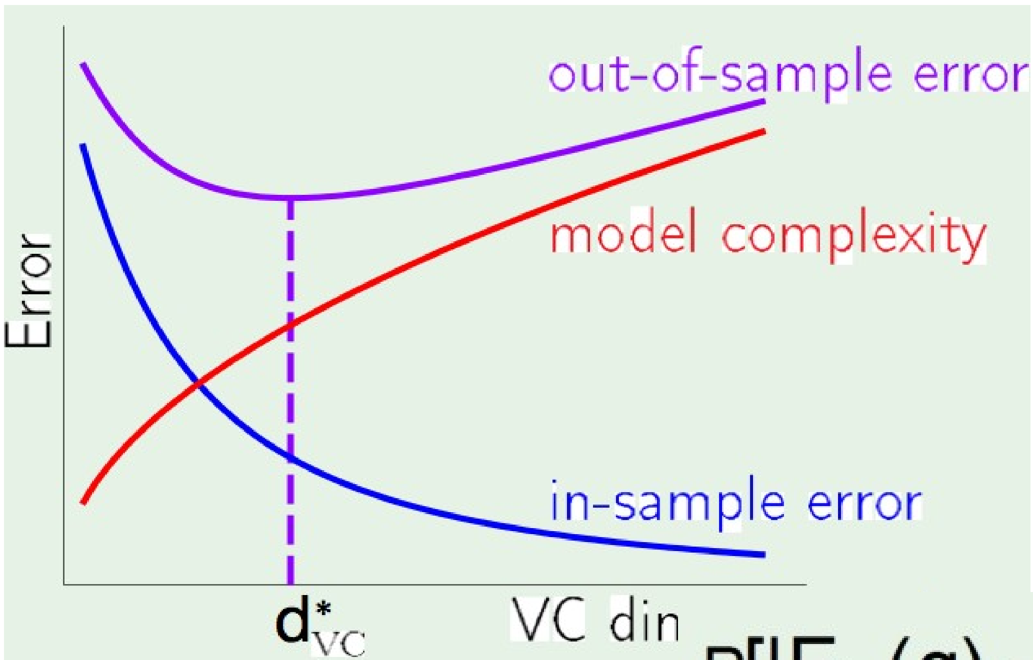
$$E_{out}(h)=E(f,h)$$

$$\mathbb{P}\left[|\nu - \mu| > \epsilon\right] \le 2e^{-2\epsilon^2 N}$$

N=sample size

M=model complexity

Vapnik-Chervonenkis (VC) Dimension

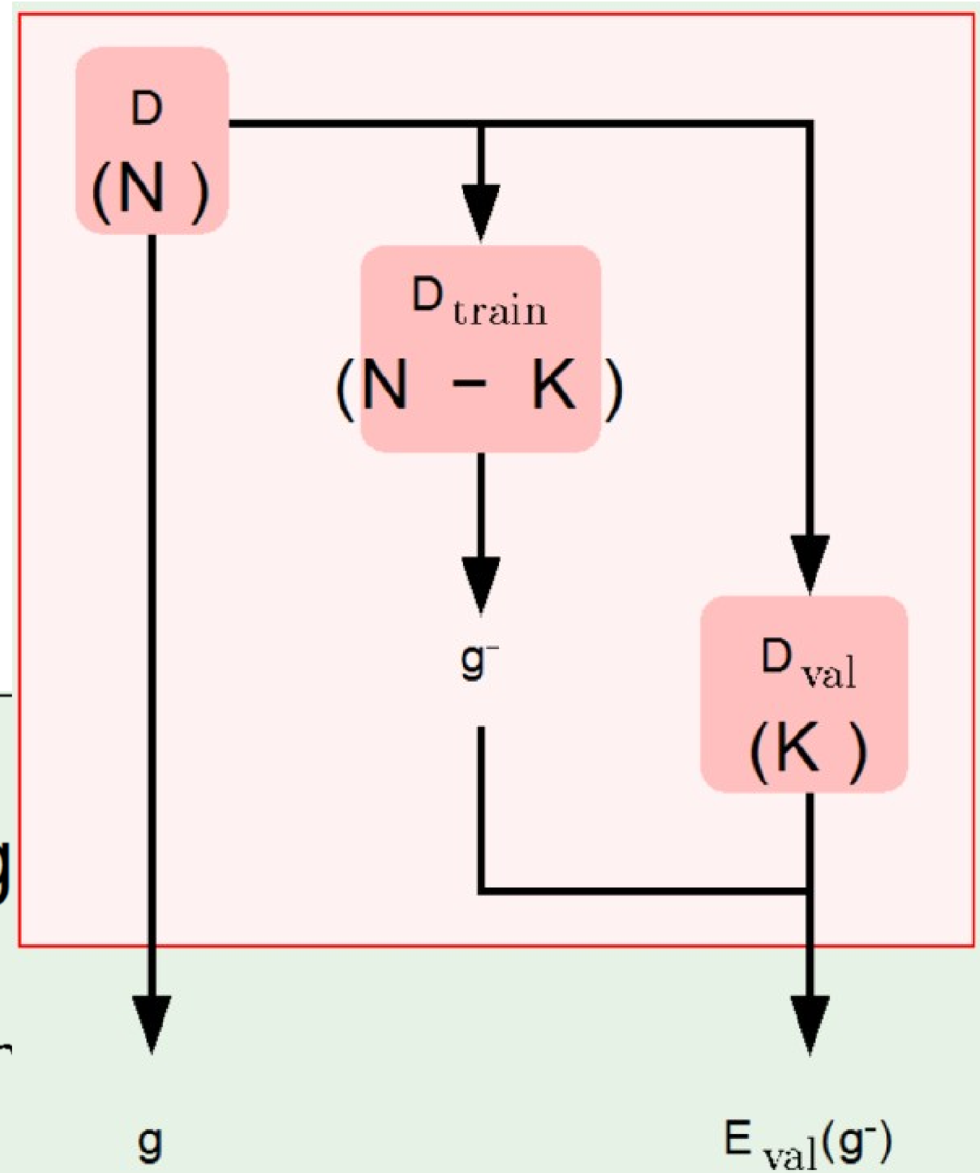


$$P\left[|E_{in}(g) - E_{out}(g)| > \epsilon\right] \le 2M\, e^{-2\epsilon^2 N}$$

Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).

The sample is divided in two subsets: **train** and **test**.

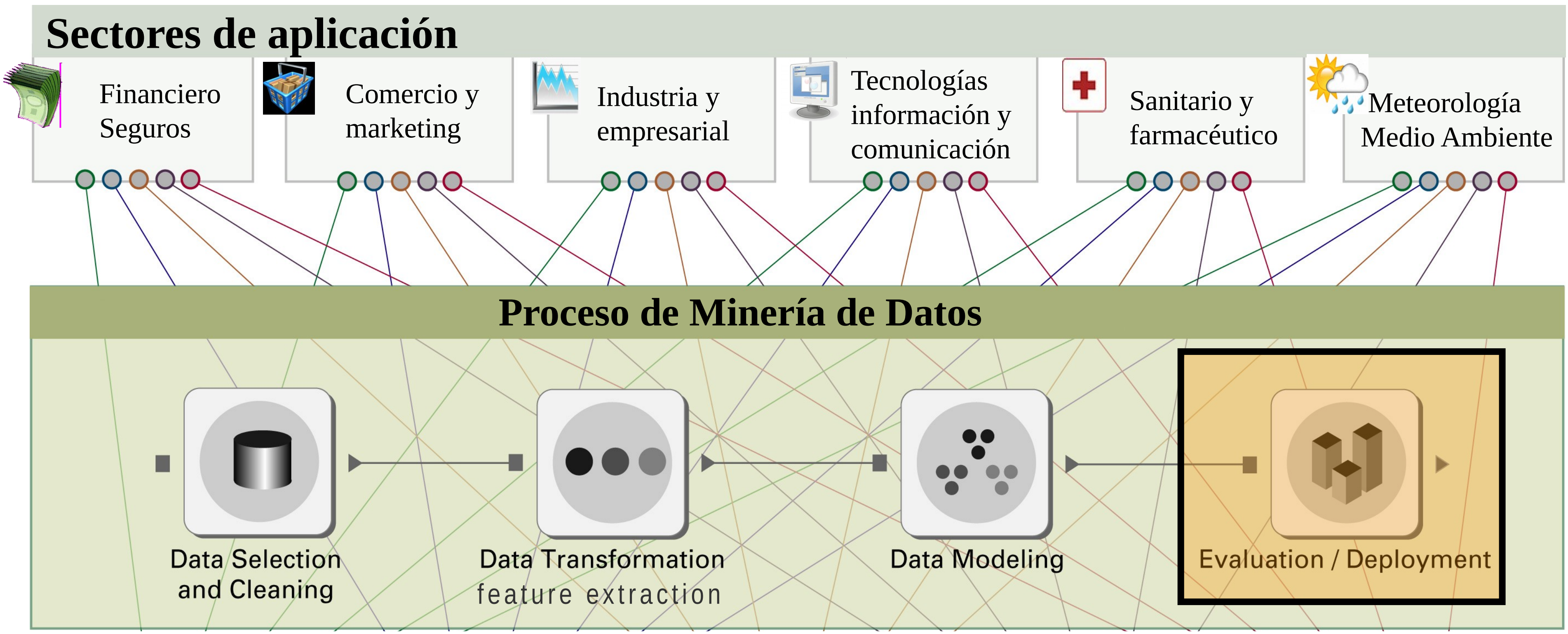
- Hold-out
- Leave-one-out
- K-fold

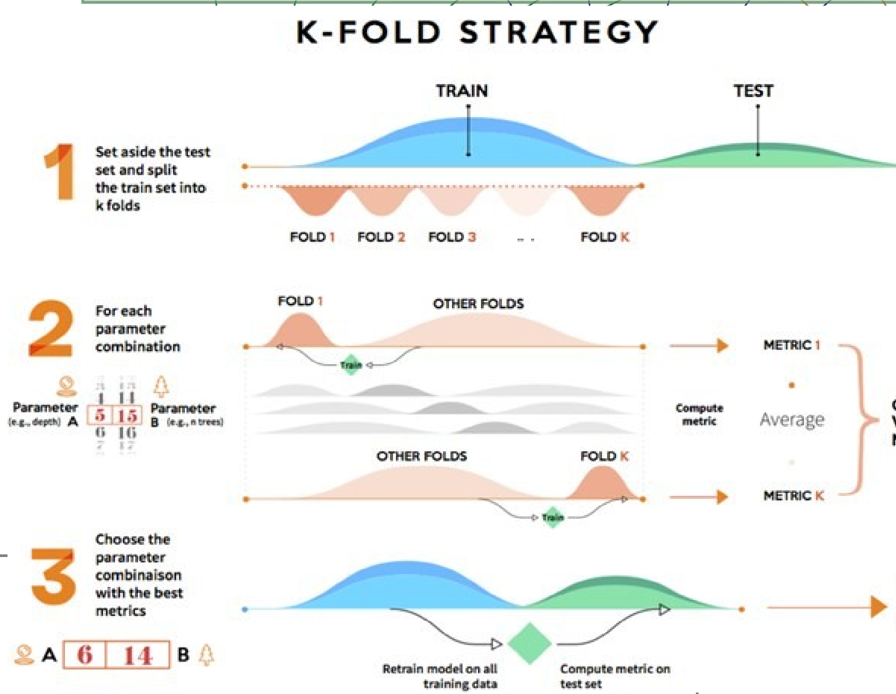


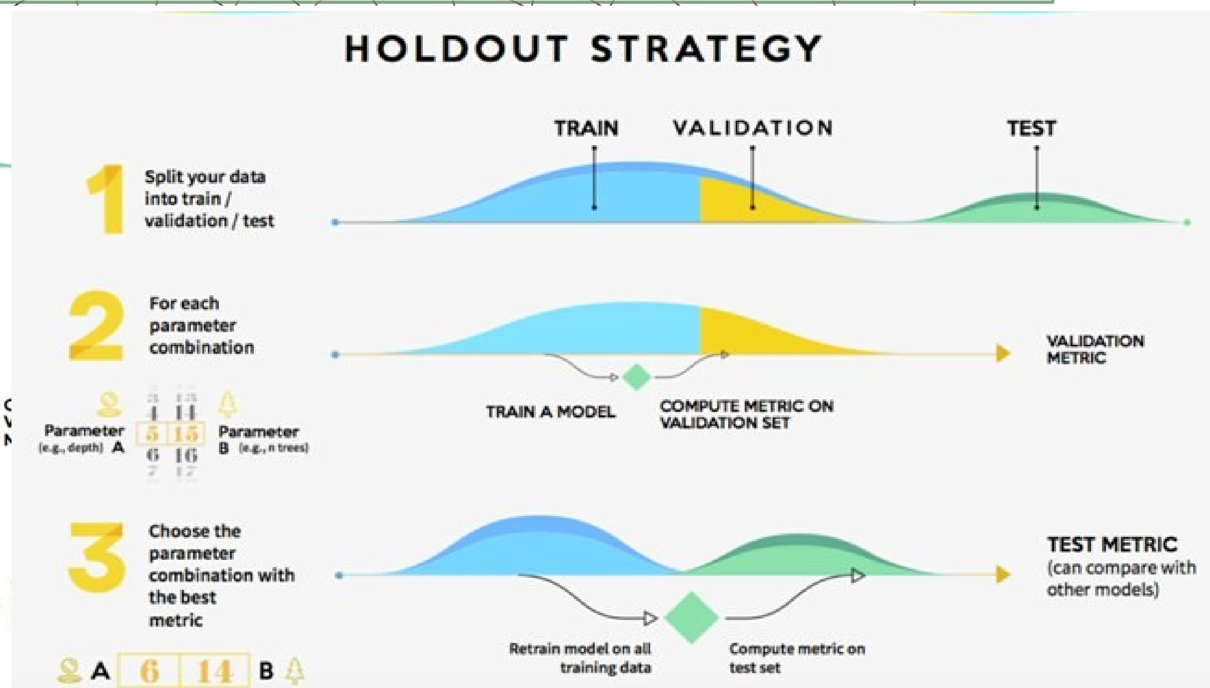
1. Can we make sure that $E_{\text{out}}(g$

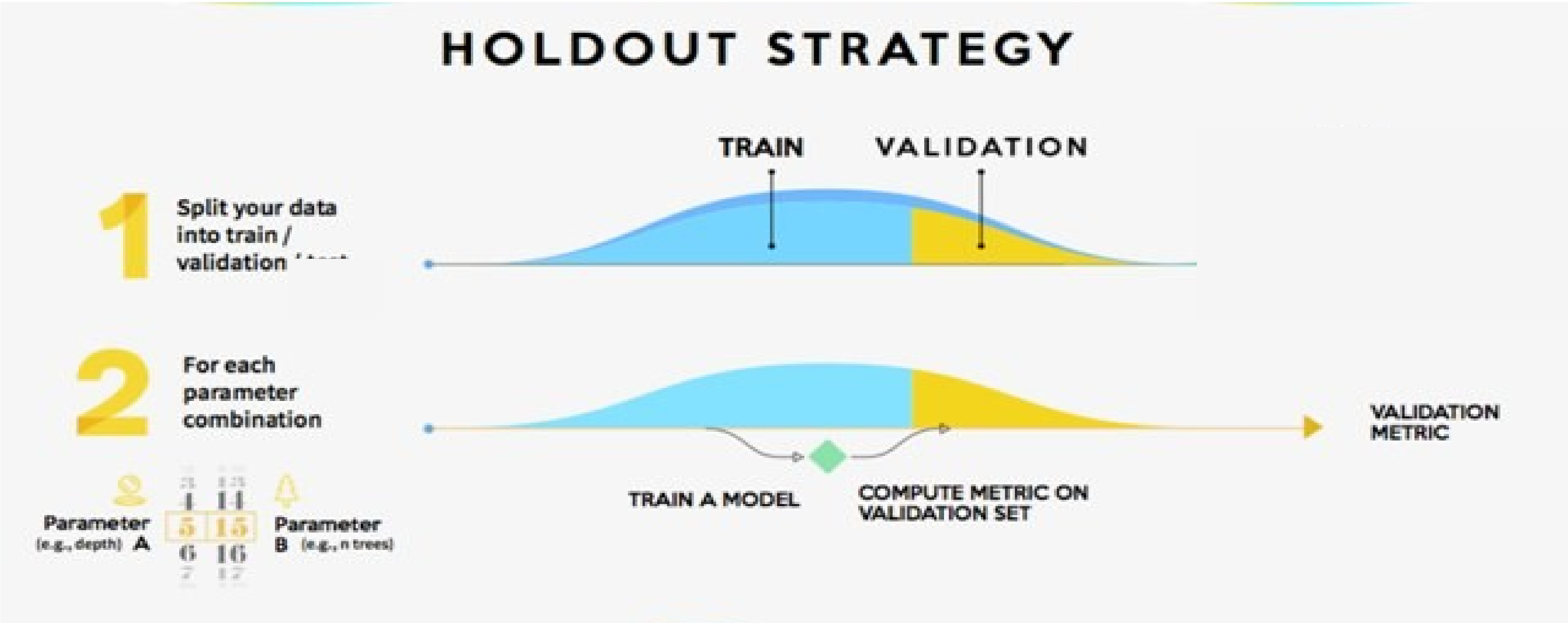
1

2. Can we make $E_{\text{in}}(g)$ small er

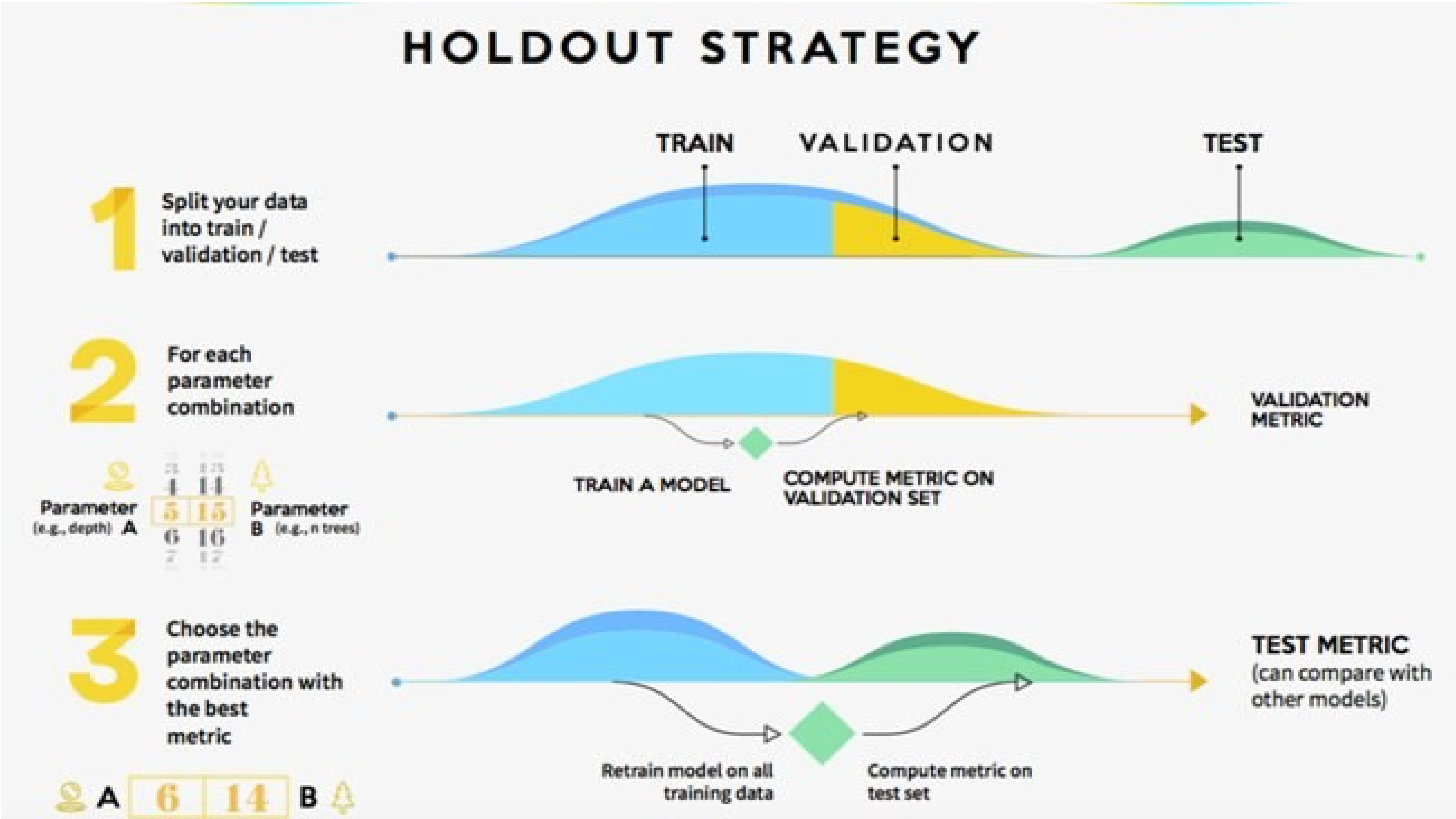

Sectores de aplicación
Financiero Seguros
Comercio y marketing
Industria y empresarial
Tecnologías información y comunicación
Sanitario y farmacéutico
Meteorología Medio Ambiente
Proceso de Minería de Datos
Data Selection and Cleaning
Data Transformation
feature extraction
Data Modeling
Evaluation / Deployment


K-FOLD STRATEGY
TRAIN
TEST
1
Set aside the test set and split the train set into k folds
FOLD 1
FOLD 2
FOLD 3
FOLD K
2
For each parameter combination
FOLD 1
OTHER FOLDS
Train
METRIC 1
Compute metric
Average
OTHER FOLDS
FOLD K
METRIC K
Parameter A (e.g., depth)
Parameter B (e.g., n trees)
3
Choose the parameter combinaison with the best metrics
A 6 14 B
Retrain model on all training data
Compute metric on test set


HOLDOUT STRATEGY
TRAIN
VALIDATION
TEST
1
Split your data into train / validation / test
2
For each parameter combination
VALIDATION METRIC
TRAIN A MODEL
COMPUTE METRIC ON VALIDATION SET
Parameter A (e.g., depth)
Parameter B (e.g., n trees)
3
Choose the parameter combination with the best metric
TEST METRIC
(can compare with other models)
Retrain model on all training data
Compute metric on test set
A 6 14 B

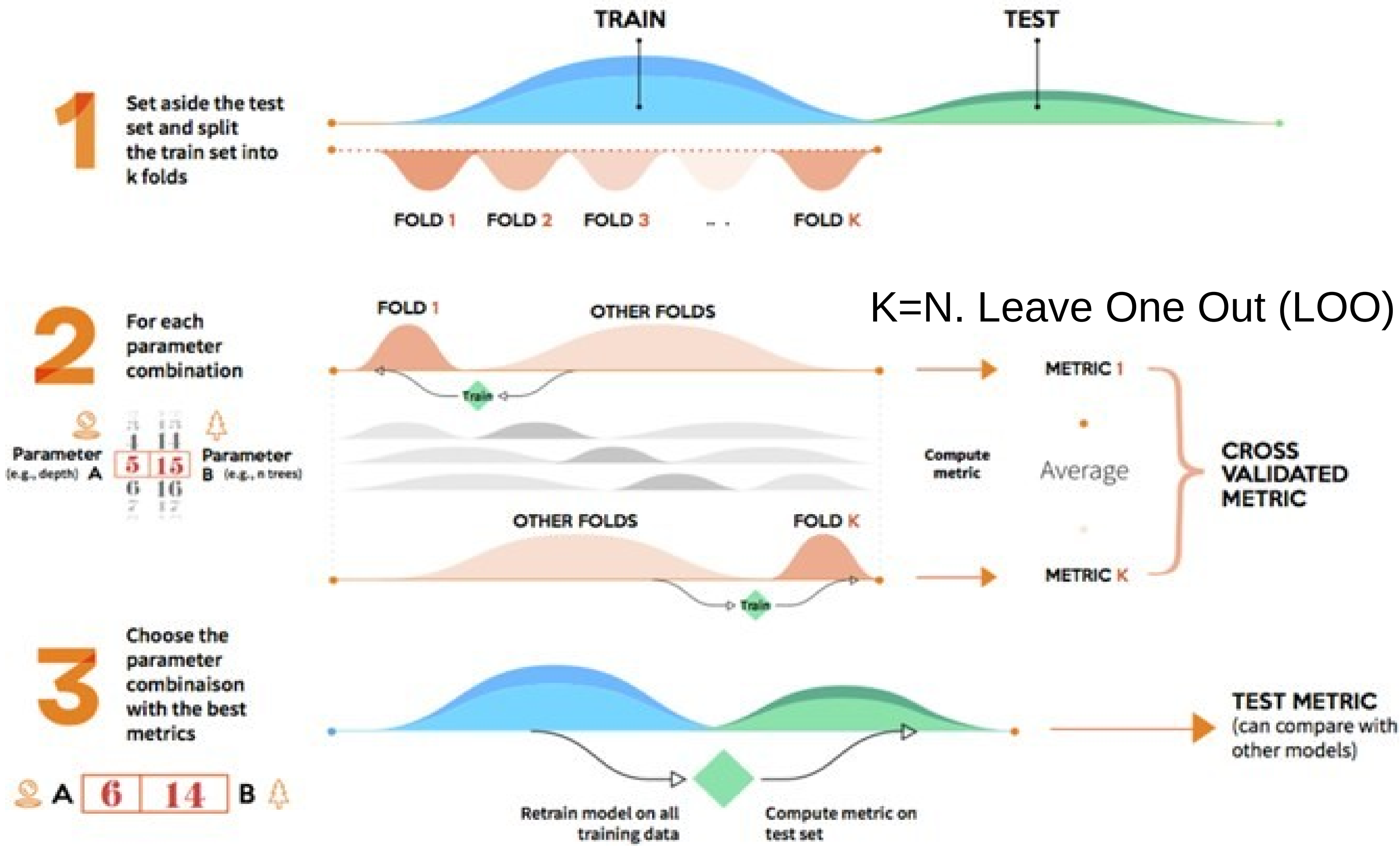


Source: Robert Kelley

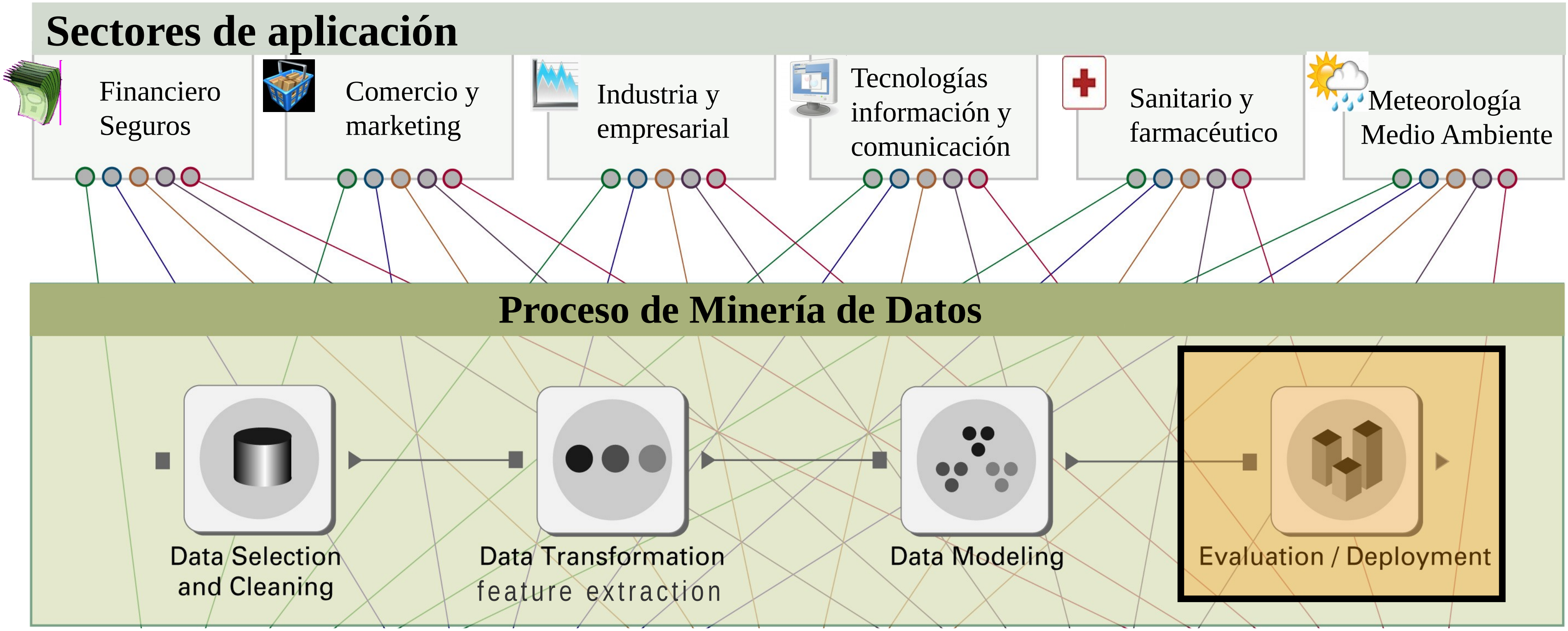


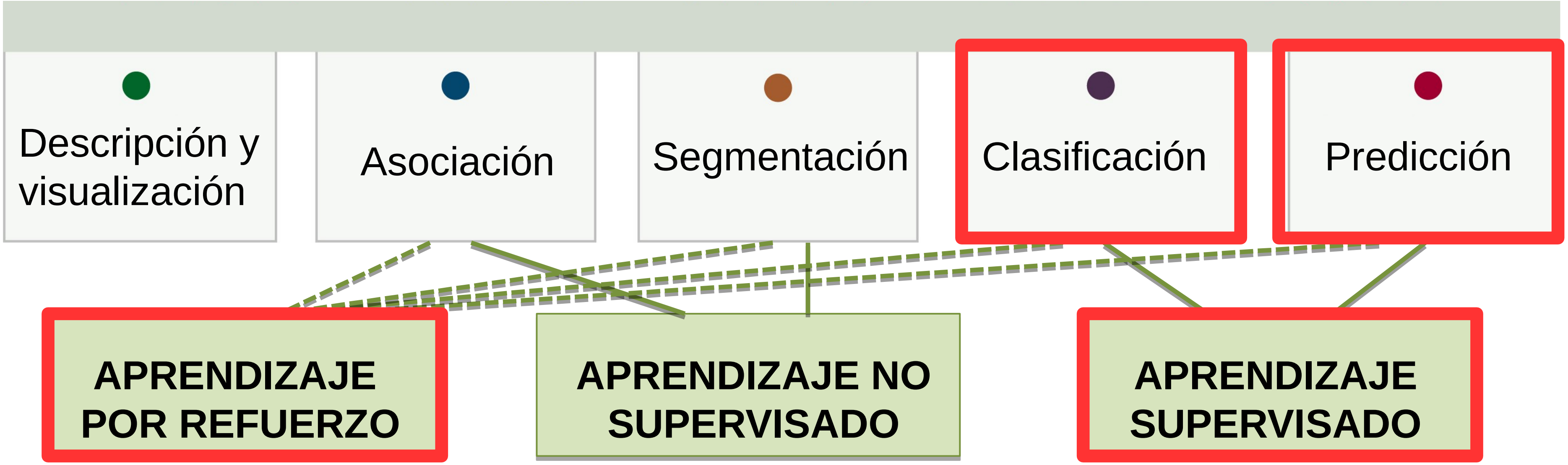
Source: [Robert Kelley]

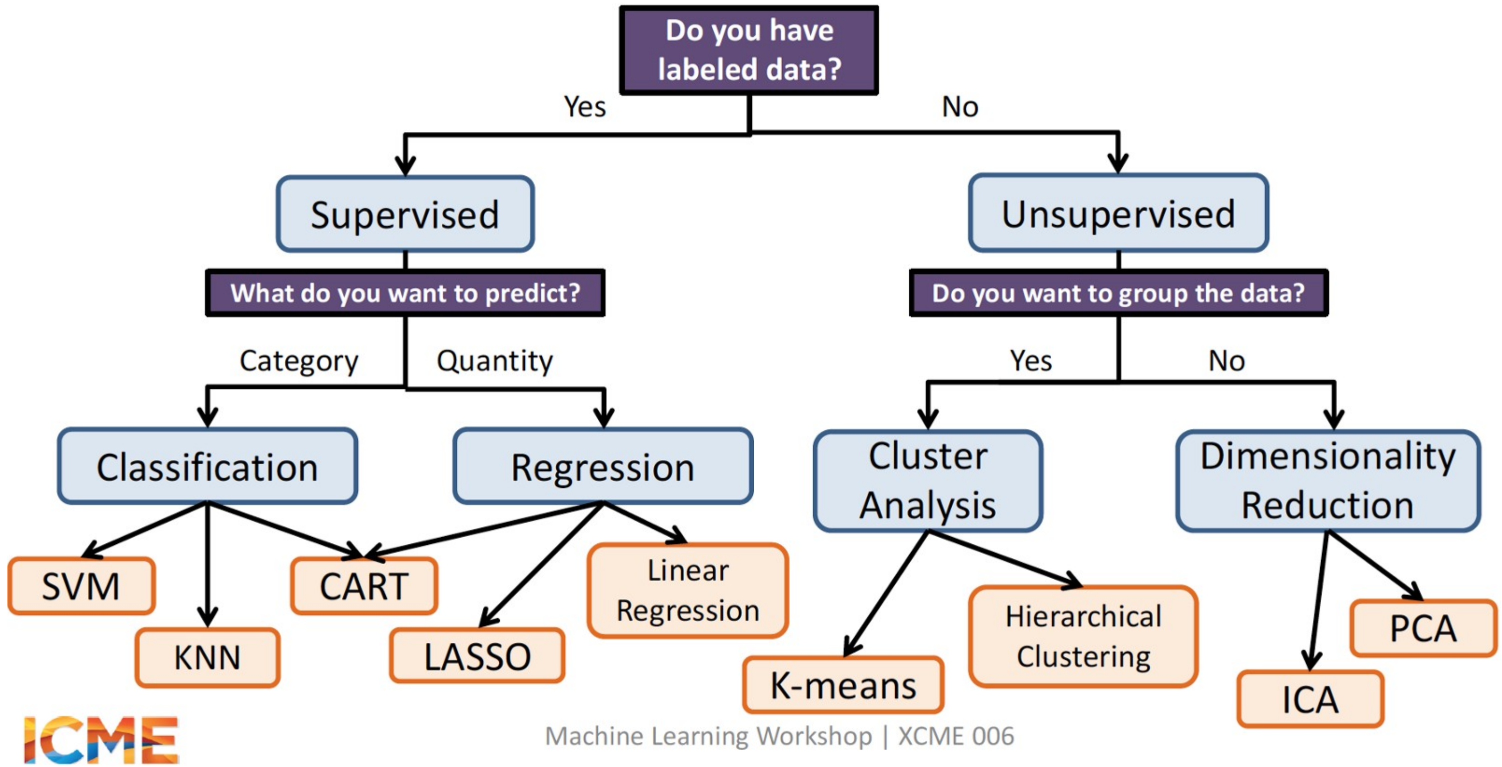
K-FOLD STRATEGY



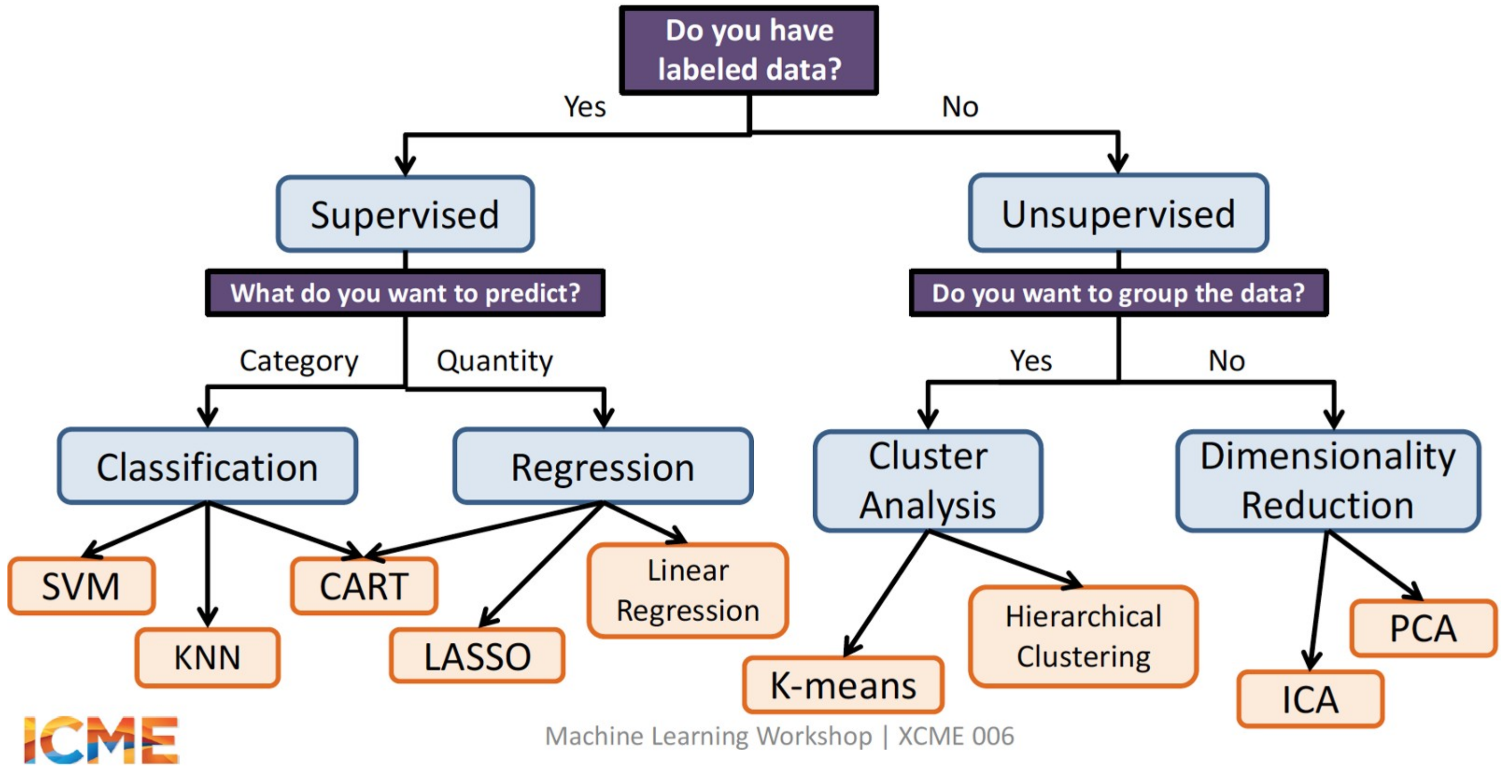
K=N. Leave One Out (LOO)


Sectores de aplicación
Financiero Seguros
Comercio y marketing
Industria y empresarial
Tecnologías información y comunicación
Sanitario y farmacéutico
Meteorología Medio Ambiente
Proceso de Minería de Datos
Data Selection and Cleaning
Data Transformation
feature extraction
Data Modeling
Evaluation / Deployment


Descripción y visualización
Asociación
Segmentación
Clasificación
Predicción
APRENDIZAJE POR REFUERZO
APRENDIZAJE NO SUPERVISADO
APRENDIZAJE SUPERVISADO


Do you have labeled data?
Yes
No
Supervised
Unsupervised
What do you want to predict?
Do you want to group the data?
Category
Quantity
Yes
No
Classification
Regression
Cluster Analysis
Dimensionality Reduction
SVM
KNN
CART
LASSO
Linear Regression
K-means
Hierarchical Clustering
PCA
ICA
ICME
Machine Learning Workshop | XCME 006


Descripción y visualización
Asociación
Segmentación
Clasificación
Predicción
APRENDIZAJE POR REFUERZO
APRENDIZAJE NO SUPERVISADO
APRENDIZAJE SUPERVISADO


Do you have labeled data?
Yes
No
Supervised
Unsupervised
What do you want to predict?
Category
Quantity
Do you want to group the data?
Yes
No
Classification
Regression
Cluster Analysis
Dimensionality Reduction
SVM
KNN
CART
LASSO
Linear Regression
K-means
Hierarchical Clustering
PCA
ICA
ICME
Machine Learning Workshop | XCME 006

		True class: p	True class: n
Predicted class	Y	**True Positives**	**False Positives**
	N	**False Negatives**	**True Negatives**
Column totals:		**P**	**N**

False Alarm Rate (**FAR**) Hit Rate (**HIR**)

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

ACCURACY FUNCTION	(TP + TN) / (P + N)
PRECISION FUNCTION	TP / (TP + FP)
SPECIFICITY FUNCTION	TN / (FP + TN)
SENSITIVITY FUNCTION	TP / (TP + FN)

FAR = 1-*specificity*

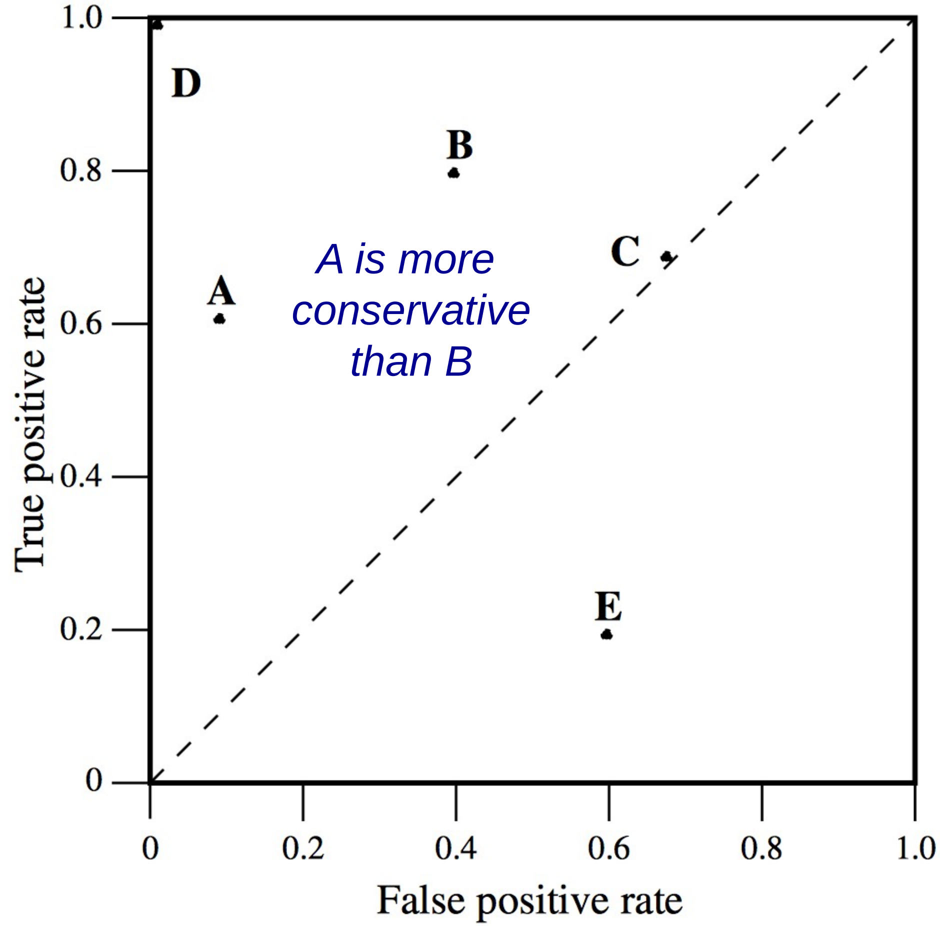
HIR = *sensivity*

Fawcett, T. (2006) An introduction to ROC analysis, In Pattern Recognition Letters, 27, 861-874, https://doi.org/10.1016/j.patrec.2005.10.010.

		True class: p	True class: n
Predicted class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Column totals:		P	N

False Alarm Rate (**FAR**) Hit Rate (**HIR**)

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$



Which systems yield?

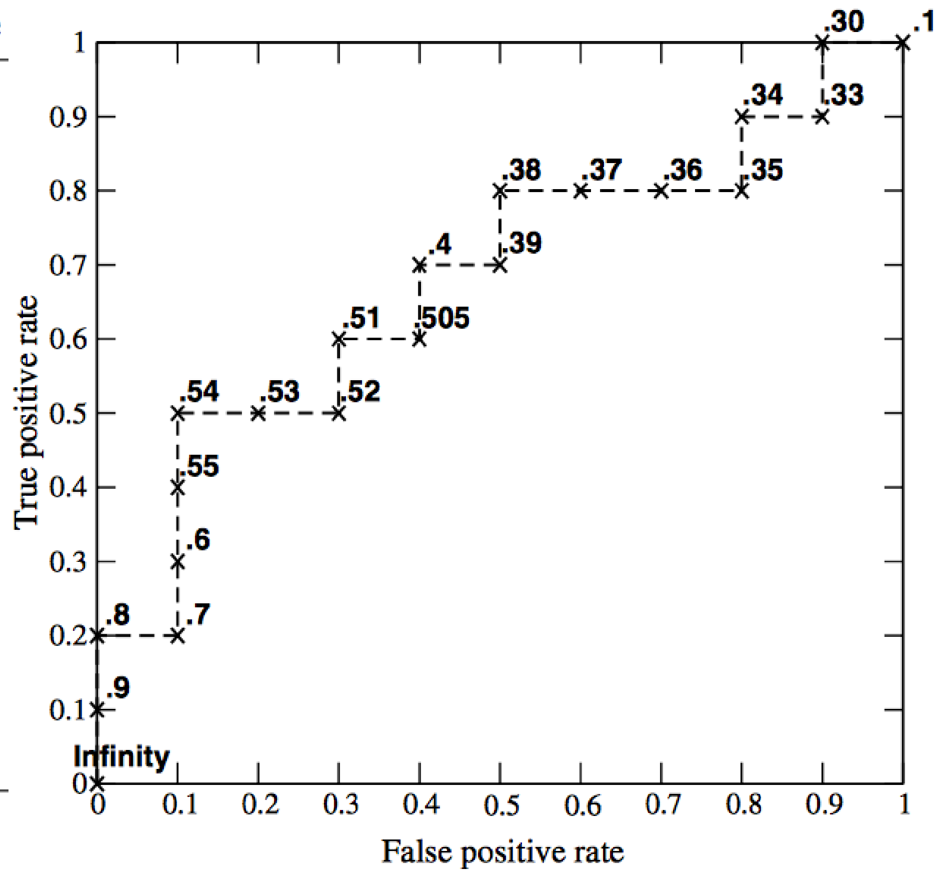
HIR = FAR = 0 → Never predicting

HIR = FAR = 1 → Always predicting

Summarizes the performance of the system over all possible probability thresholds.

```
library(pROC)
obs<-c(rep(0,50),rep(1,50));
prd<-obs+2*(runif(100)-0.5);
prd[which(prd<0)]<-0; prd[which(prd>1)]<-1;
plot(roc(obs,prd), print.auc=TRUE)
hist(prd)
```

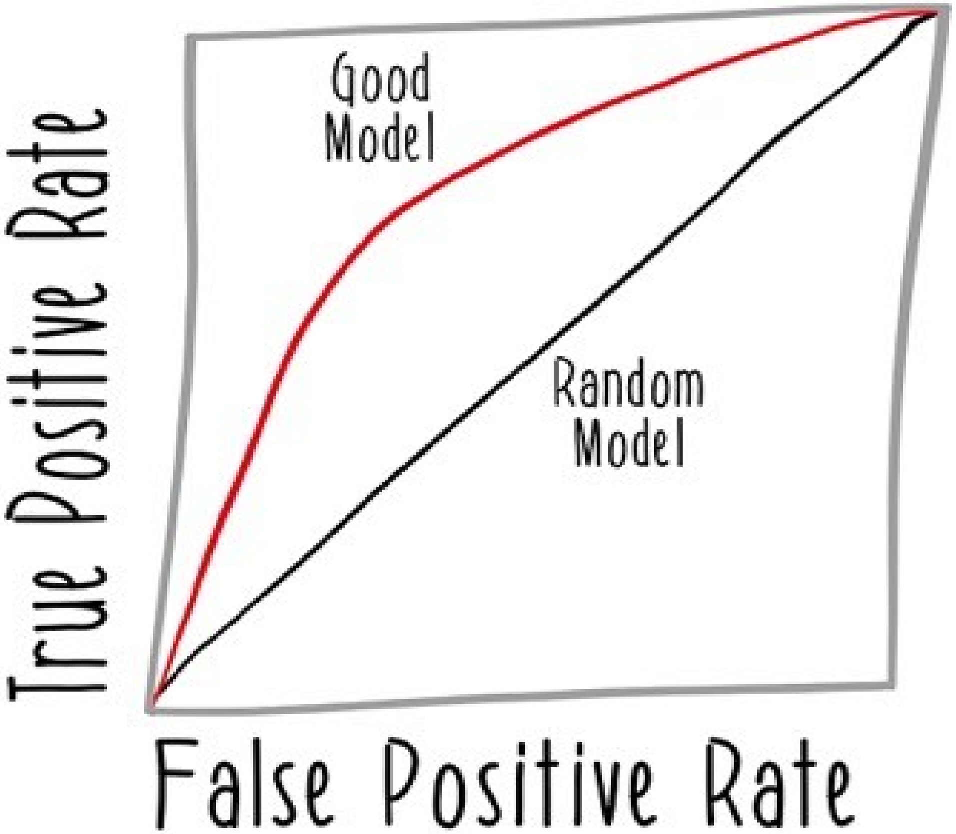
Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



Summarizes the performance of the system over all possible probability thresholds.

```
library(pROC)
obs<-c(rep(0,50),rep(1,50));
prd<-obs+2*(runif(100)-0.5);
prd[which(prd<0)]<-0; prd[which(prd>1)]<-1;
plot(roc(obs,prd), print.auc=TRUE)
hist(prd)
```

Inst#	Class	Score	Inst#	Class	Score
1	**p**	.9	11	**p**	.4
2	**p**	.8	12	**n**	.39
3	**n**	.7	13	**p**	.38
4	**p**	.6	14	**n**	.37
5	**p**	.55	15	**n**	.36
6	**p**	.54	16	**n**	.35
7	**n**	.53	17	**p**	.34
8	**n**	.52	18	**n**	.33
9	**p**	.51	19	**p**	.30
10	**n**	.505	20	**n**	.1



https://www.kdnuggets.com/2018/01/machine-learning-model-metrics.html

- **Accuracy:** Overall, how often is the classifier correct?
    - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
    - (FP+FN)/total = (10+5)/165 = 0.09
    - equivalent to 1 minus Accuracy
    - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
    - TP/actual yes = 100/105 = 0.95
    - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
    - FP/actual no = 10/60 = 0.17
- **Specificity:** When it's actually no, how often does it predict no?
    - TN/actual no = 50/60 = 0.83
    - equivalent to 1 minus False Positive Rate
- **Precision:** When it predicts yes, how often is it correct?
    - TP/predicted yes = 100/110 = 0.91
- **Prevalence:** How often does the yes condition actually occur in our sample?
    - actual yes/total = 105/165 = 0.64

HIR
(Hit rate)

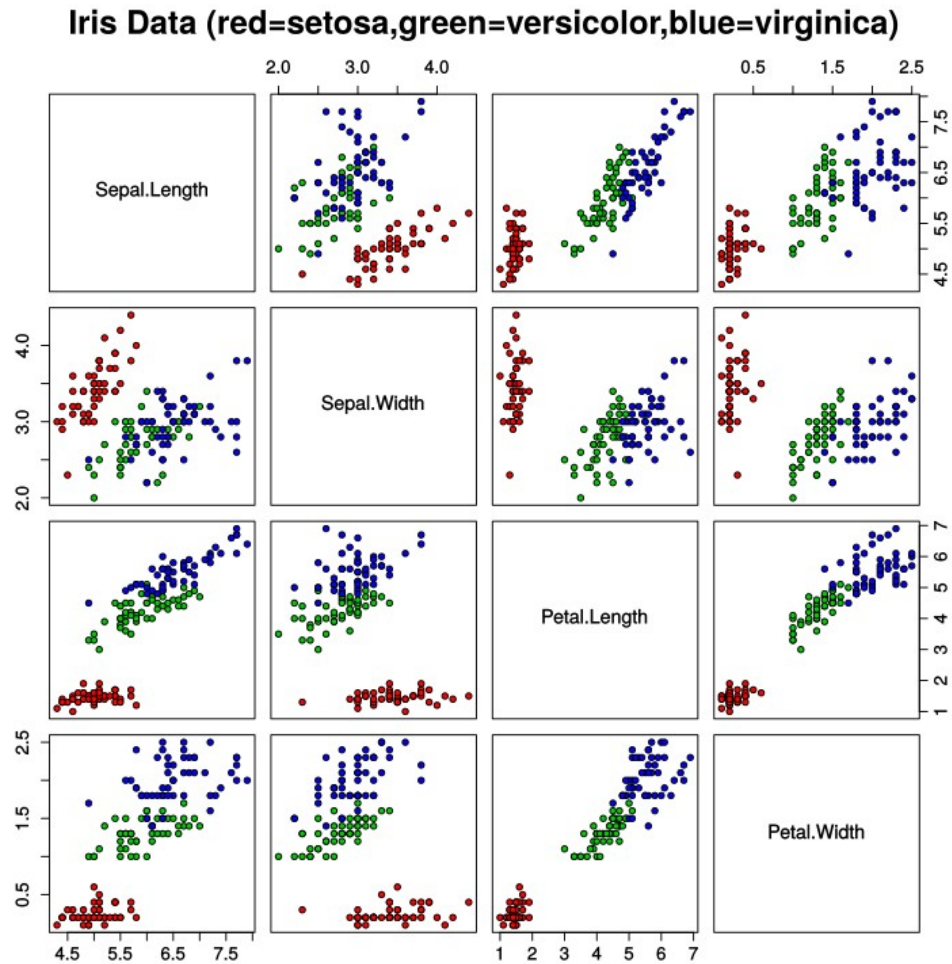
FAR
(False alarm rate)

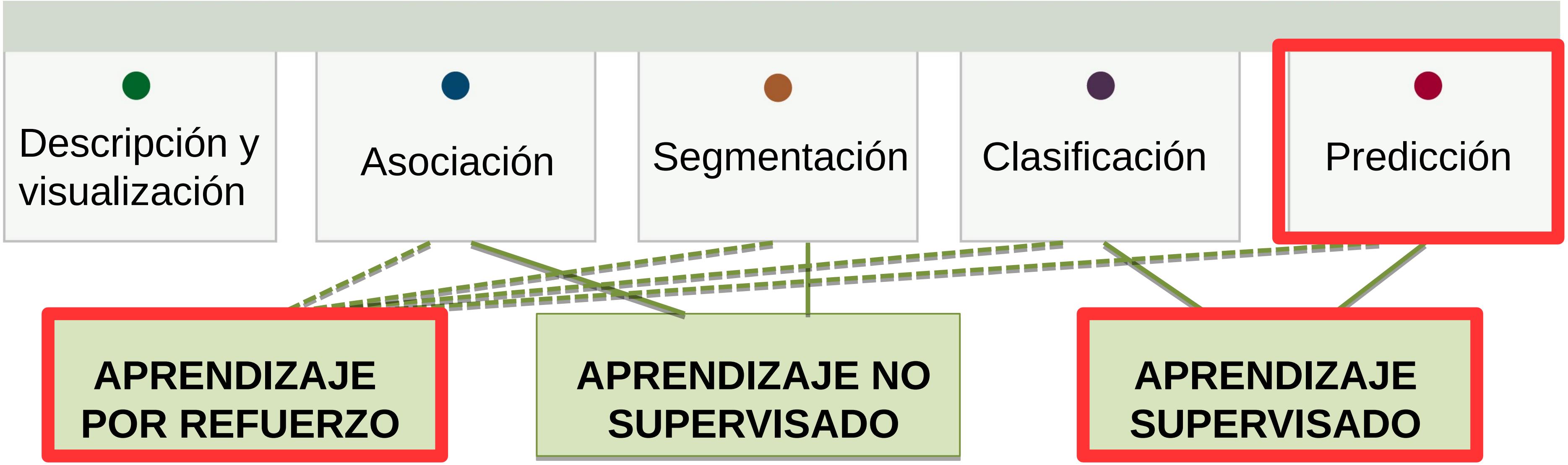
For which systems do the following equalities hold?:

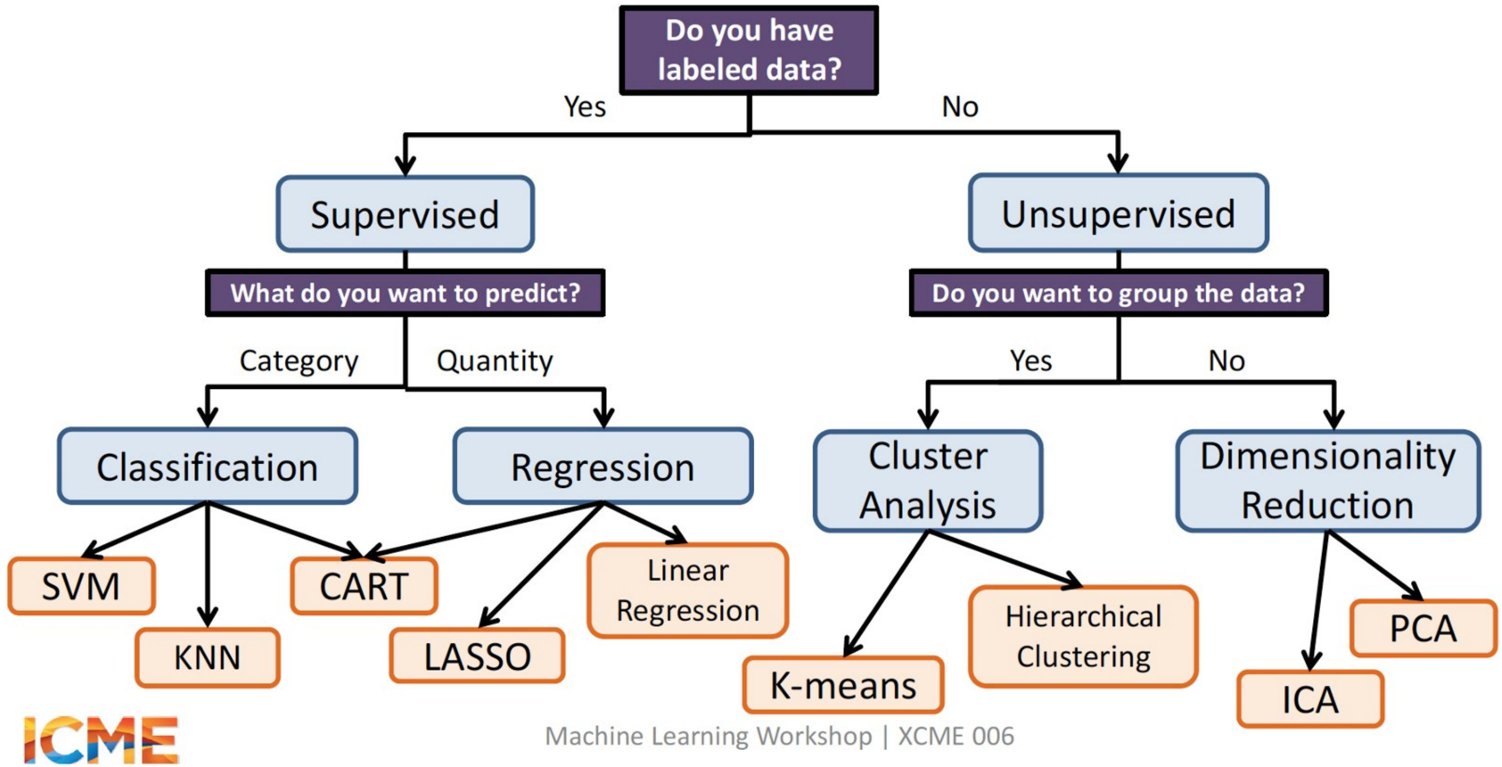
HIR = FAR = 0

HIR = FAR = 1

```
data(iris)
fitControl <- trainControl(method="none",
                           number=1,
                           repeats=1,
                           verboseIter=TRUE)
modelFit <- train(Species ~ ., data=iris, method="knn", trControl=fitControl)
pred <- predict(modelFit, newdata = iris[,-5])
acc<-confusionMatrix(iris$Species,pred)
print(acc)
```




Descripción y visualización
Asociación
Segmentación
Clasificación
Predicción
APRENDIZAJE POR REFUERZO
APRENDIZAJE NO SUPERVISADO
APRENDIZAJE SUPERVISADO


Do you have labeled data?
Yes
No
Supervised
Unsupervised
What do you want to predict?
Do you want to group the data?
Category
Quantity
Yes
No
Classification
Regression
Cluster Analysis
Dimensionality Reduction
SVM
KNN
CART
LASSO
Linear Regression
K-means
Hierarchical Clustering
PCA
ICA
ICME
Machine Learning Workshop | XCME 006

Model accuracy (training and validation).

Some models are trained using an **empirical error (cost) function**, which measures **model accuracy** as the difference between the predicted and the actual value. In this cas, this a natural **validation measure**.

Correlation (Pearson, Speman)

- This cost function could be anything:
  - Sum of absolute errors: $J = \sum |y - u|$.
  - Sum of square errors: $J = \sum (y - u)^2$.
  - As long as the minimum occurs when the distributions are the same, in theory it would work.
- One good idea is that u represents the parameters of the distribution of y.
  - Rationale: often natural processes are fuzzy, and any input might have a range of outputs.
  - This approach also gives a smooth measure of how accurate we are.
  - The maximum likelihood principle says that: $\theta_{\mathrm{ML}} = \arg\max_\theta p(y; u)$
  - Thus we want to minimize: $J = -p(y; u)$
  - For i samples: $J = -\prod_i p(y_i; u)$
  - Taking log both sides: $J' = -\sum_i \log p(y_i; u)$.
  - This is called cross-entropy.
- Applying the idea for: $y \sim \mathrm{Gaussian}(\mathrm{center} = u)$:
  - $p(y; u) = e^{-(y-u)^2}$.
  - $J = -\sum \log e^{-(y-u)^2} = \sum (y - u)^2$
  - This motivates sum of squares as a good choice.

Model performance: Validation diagnostics and metrics.

There are several domain-dependent diagnostics (computed separately for prediction ‘p’ and observation ‘o’) and metrics/errors for validating model performance.

Distributional consistency: evaluates the model capability to reproduce the distribution of the observed data.

- **Bias** = mean p – mean o
- **Variance ratio** = var p / var o
- **Distributional similarity:** ks-score, Von Misses, pdf-score, etc.

The **quantile-quantile plot** is a typical tool to evaluate, in a graphical way, the distributional similarity of the order statistics (e.g. **percentiles**).

Different diagnostics for different fields.

Accuracy: assess the correspondence of the simulated and observed sequences. Two typical scores are usally used: Root Mean Square Error (**RMSE**) and the (Pearson/Spearman/Kendall) **Correlation**.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i)^2} \qquad r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Distributional consistency: evaluates the model capability to reproduce the distribution of the observed data. The most popular are the **bias** (mean difference) or the **ratio of variances/standard deviation**. In addition, there are hypothesis tests to evaluate in a global way the similarity of the observed and simulated series (e.g. **Kolmogorov-Smirnov**, **Perkins**, **Von Misses**, etc).

The **quantile-quantile plot** is a typical tool to evaluate, in a graphical way, the distributional similarity of the order statistics (e.g. **percentiles**).

```
## Example with R:
?qqplot
require(graphics)
y<-rt(200,df=5)
qqnorm(y)
qqline(y,col=2)
qqplot(y,rt(300,df=5))
```

Accuracy: assess the correspondence of the simulated and observed sequences. Two typical scores are usally used: Root Mean Square Error (**RMSE**) and the (Pearson/Spearman/Kendall) **Correlation**.

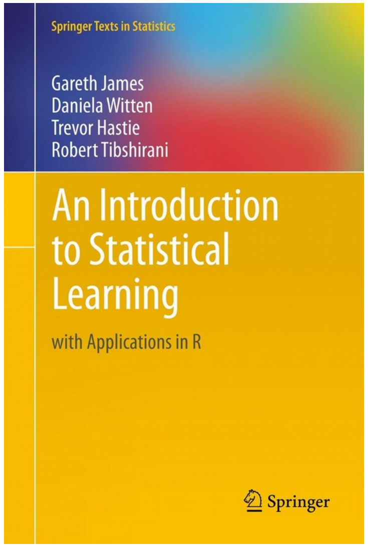
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i)^2} \qquad r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Distributional consistency: evaluates the model capability to reproduce the distribution of the observed data. The most popular are the **bias** (mean difference) or the **ratio of variances/standard deviation**. In addition, there are hypothesis tests to evaluate in a global way the similarity of the observed and simulated series (e.g. **Kolmogorov-Smirnov**, **Perkins**, **Von Misses**, etc).

The **quantile-quantile plot** is a typical tool to evaluate, in a graphical way, the distributional similarity of the order statistics (e.g. **percentiles**).

```
## Example with R:
?qqplot
require(graphics)
y<-rt(200,df=5)
qqnorm(y)
qqline(y,col=2)
qqplot(y,rt(300,df=5))
```

How to create and use our own functions, including validation measures in R?



An Introduction to Statistical Learning: With Applications in R

James, G., Witten, D., Hastie, T., Tibshirani, R.

Springer (2013)

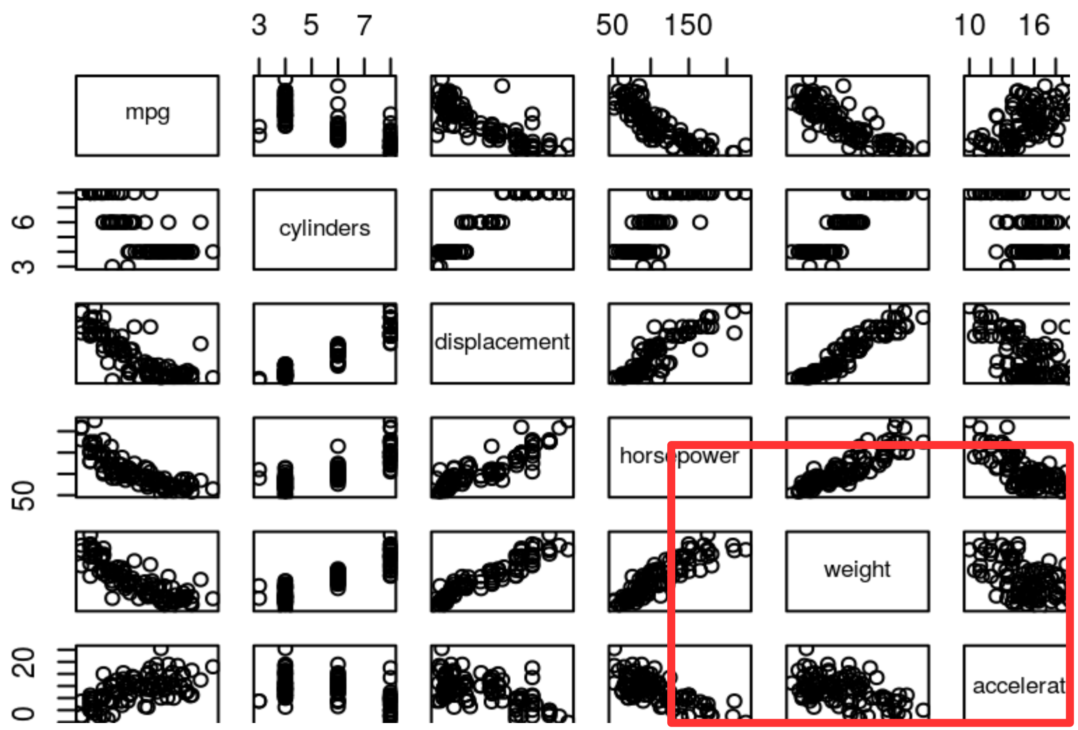
http://www-bcf.usc.edu/~gareth/ISL

```
install.packages(“ISLR”)
library(“ISLR”)
library(help = "ISLR")
```

```
> data(Auto)
> str(Auto)
```

```
'data.frame':    392 obs. of  9 variables:
 $ mpg         : num  18 15 18 16 17  ...
 $ cylinders   : num  8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 ...
 $ horsepower  : num  130 165 150 150 ...
 $ weight      : num  3504 3693 3436  ...
 $ acceleration: num  12 11.5 11 12   ...
 $ year        : num  70 70 70 70 70  ...
 $ origin      : num  1 1 1 1 1 1 1 1 ...
 $ name        : Factor w/ 304 levels ...
```

```
> pairs(Auto)
```



CARET (ClAssification and REgresion Training) is a wrapper of a number of standard machine learning packages which performs model tunning (optimization of the model parameters) and cross-validation strategies.
http://topepo.github.io/caret/index.html

```
> modelLookup(model = "lm")
  model parameter     label forReg forClass probModel
     lm intercept intercept   TRUE    FALSE     FALSE
```

```
trainControl(method , number, …)
    method: "none", "cv", "LOOCV"
    number: For "cv" (2 => hold-out, 10 => 10-fold)
```

```
> ctrl <- trainControl(method = "LOOCV")
> mod <- train(weight ~ horsepower,
             data = Auto,
             method = "lm",
             trControl = ctrl)
        # metric="RMSE",
        # preProc = c("center", "scale")
```

```
> mod
```

```
Linear Regression | 392 samples | 1 predictor | No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 391, 391, 391, 391, 391, 391, ...
Resampling results:
  RMSE      Rsquared   MAE
  429.5254  0.7436498  347.5039
```

```
> str(model$control$index$Fold001)
 int [1:391] 2 3 4 5 6 7 8 9 10 11 ...
```

```
> plot(mod$pred$obs, type="l");
      lines(1:392,mod$pred$pred,col="red")
```