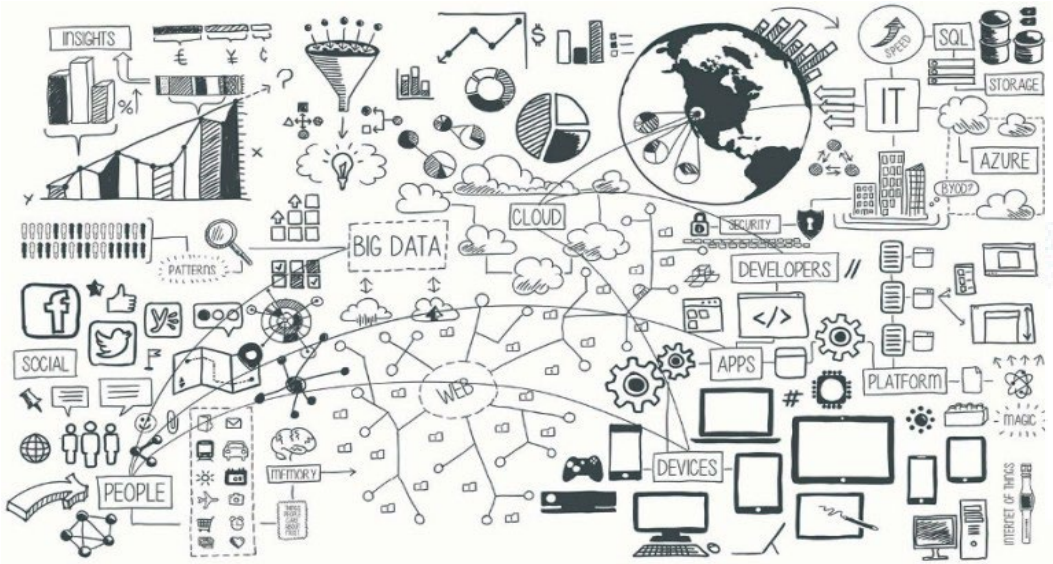
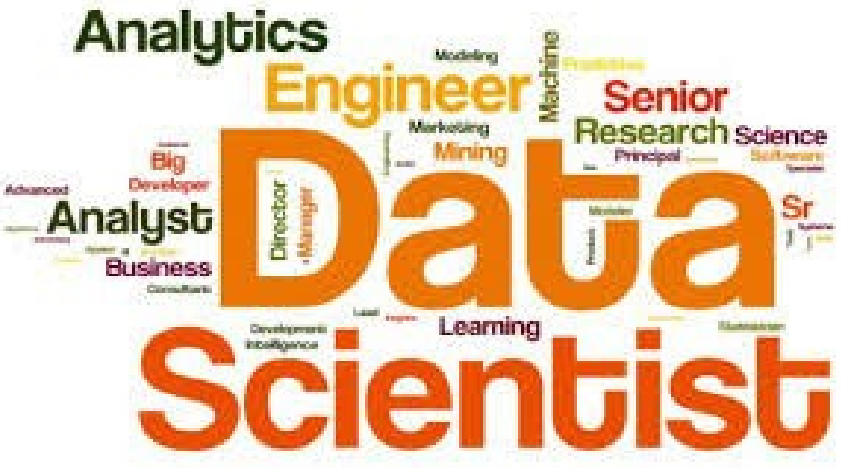
Data Mining (Minería de Datos)

Evaluación, sobreajuste y validación cruzada (cross-validation)





Sixto Herrera

Ana Casanueva

Grupo de Meteorología

Univ. de Cantabria – CSIC
MACC / IFCA

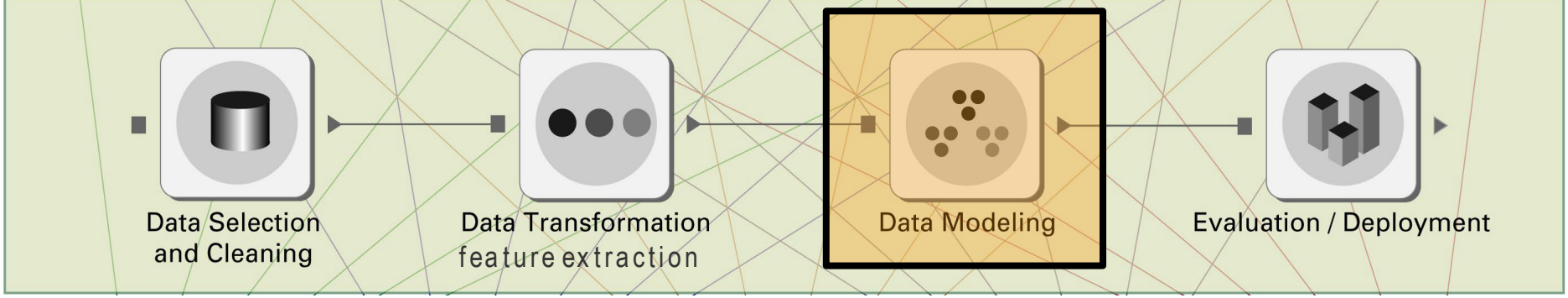
Oct	29	Presentación, introducción y perspectiva histórica
	30	Paradigmas, problemas canonicos y data challenges
	31	Reglas de asociación
Nov	4	Practica: Reglas de asociación
	6	**Evaluación, sobrejuste y crossvalidacion**
	11	**Practica: Crossvalidacion**
	13	Árboles de clasificacion y decision
	18	Practica: Árboles de clasificación
	20	Técnicas de vecinos cercano (k-NN)
	25	Práctica: Vecinos cercanos
	27	Comparación de Técnicas de Clasificación.
Dic	2	Árboles de clasificación y regresion (CART)
	4	Práctica: Árboles de clasificación y regresion (CART)
	9	Practica: El paquete CARET
	11	Ensembles: Bagging and Boosting
	13	Random Forests
	16	Gradient boosting
	18	Practica: XAI-Explainable Artificial Intelligence
Ene	8	Reducción de dimensión no lineal
	13	Reducción de dimensión no lineal
	15	Técnicas de agrupamiento
	20	Técnicas de agrupamiento
	22	Predicción Condicionada
	24	Sesión de refuerzo/repaso.
	29	Examen

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

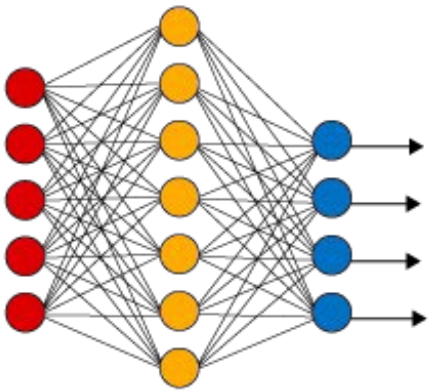
Sectores de aplicación

- Financiero Seguros
- Comercio y marketing
- Industria y empresarial
- Tecnologías información y comunicación
- Sanitario y farmacéutico
- Meteorología Medio Ambiente

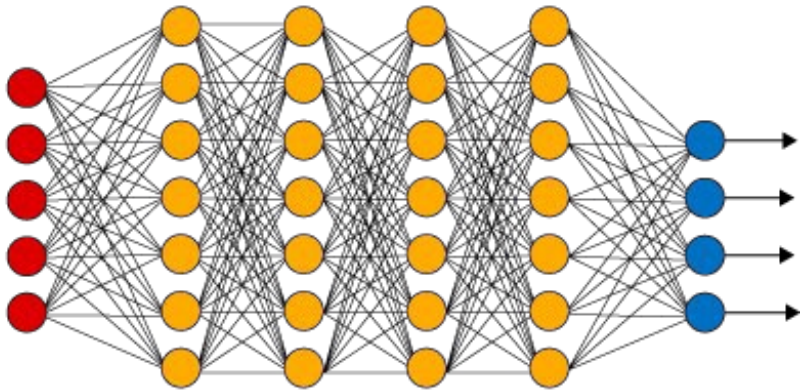
Proceso de Minería de Datos



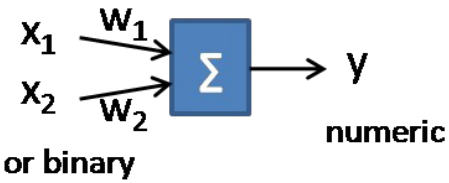
Simple Neural Network



Deep Learning Neural Network



Input Layer Hidden Layer Output Layer

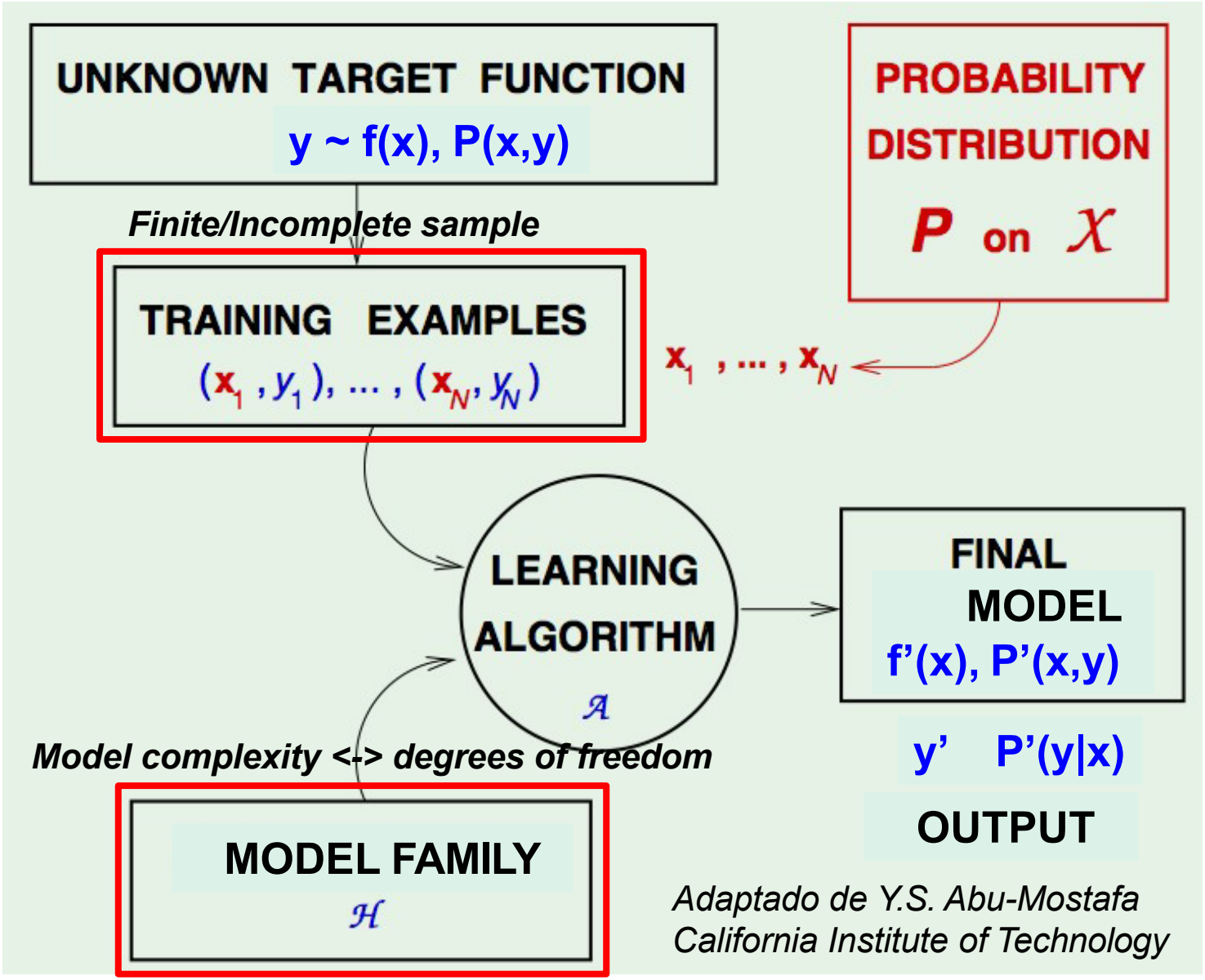


$$y = w_0 + w_1x_1 + w_2x_2$$

$$y = f(X,W) = X^T.W$$

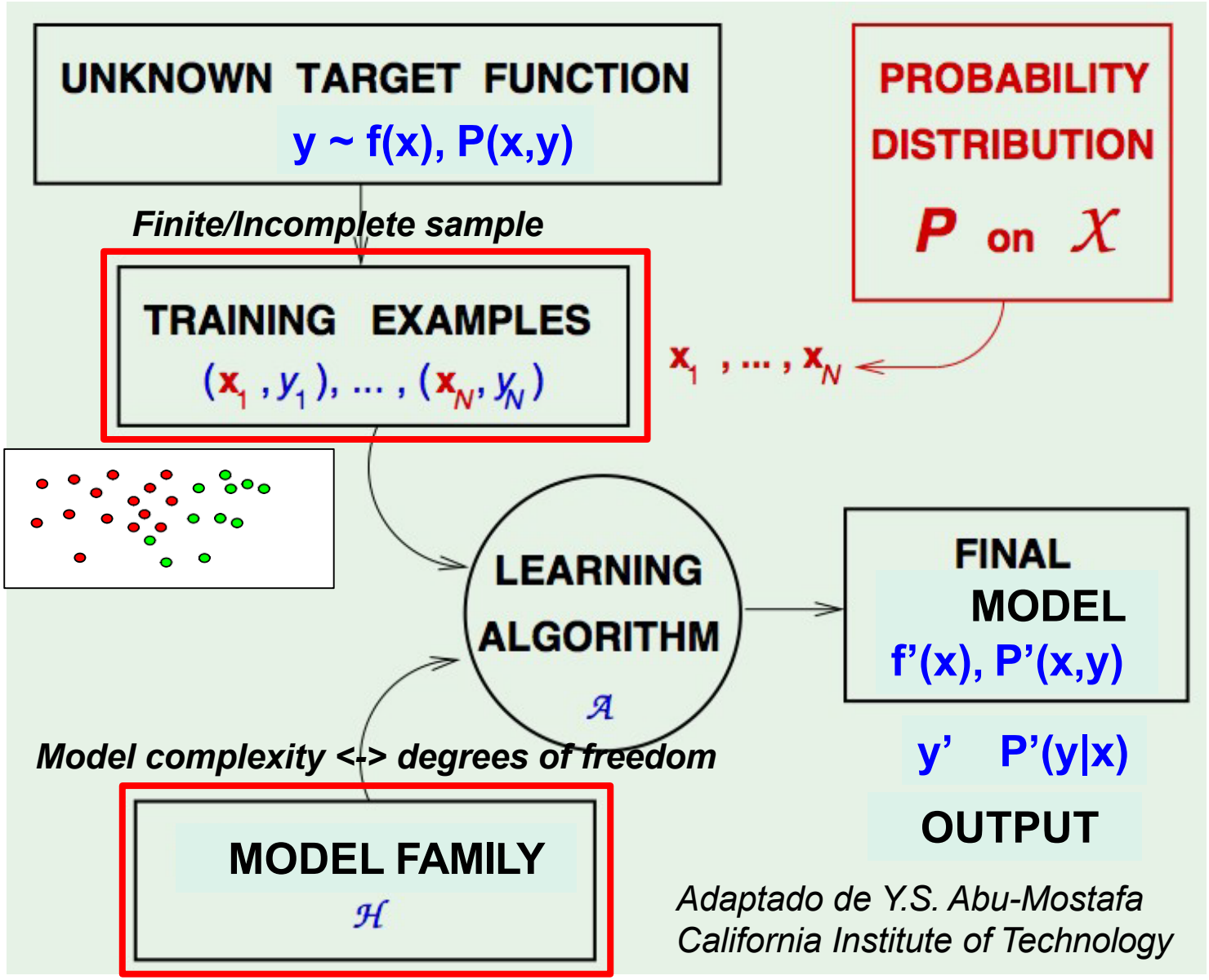
REGRESSION $W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ $X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$

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



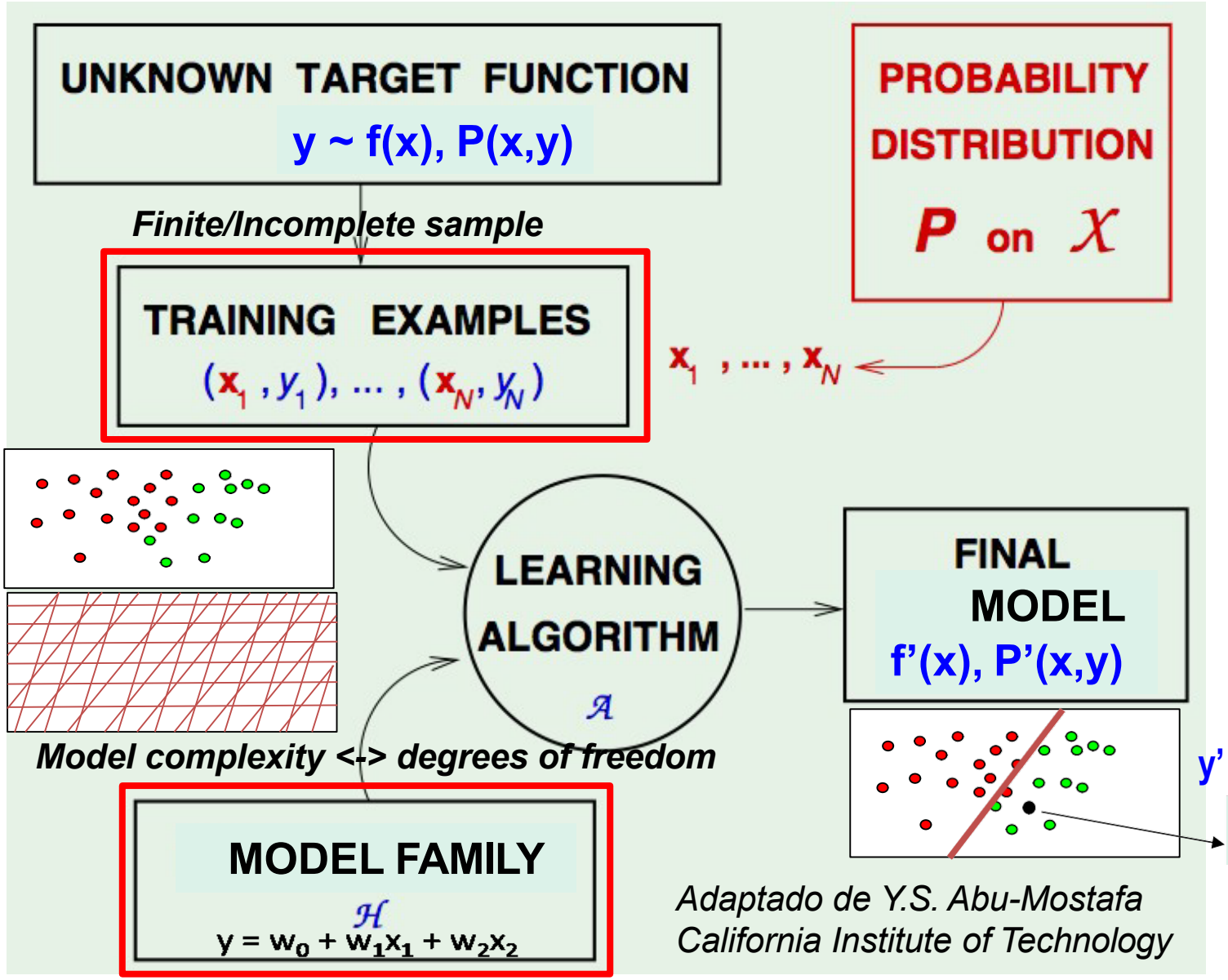
Adaptado de Y.S. Abu-Mostafa
California Institute of Technology

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



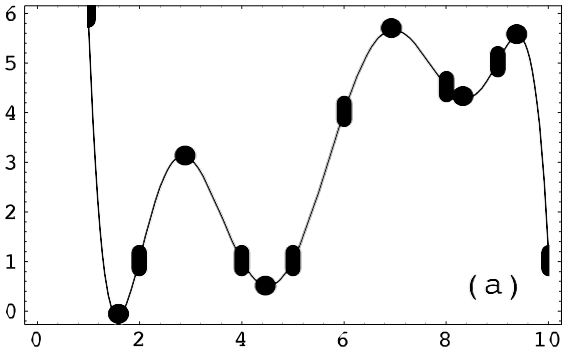
Adaptado de Y.S. Abu-Mostafa
California Institute of Technology

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



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

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



UNKNOWN TARGET FUNCTION

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

Finite/Incomplete sample

TRAINING EXAMPLES

$(x_1, y_1), \ldots, (x_N, y_N)$

PROBABILITY DISTRIBUTION

P on $\mathcal{X}$

$x_1, \ldots, x_N$

LEARNING ALGORITHM

$\mathcal{A}$

FINAL MODEL

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

y' $P'(y|x)$

OUTPUT

Model complexity <-> degrees of freedom

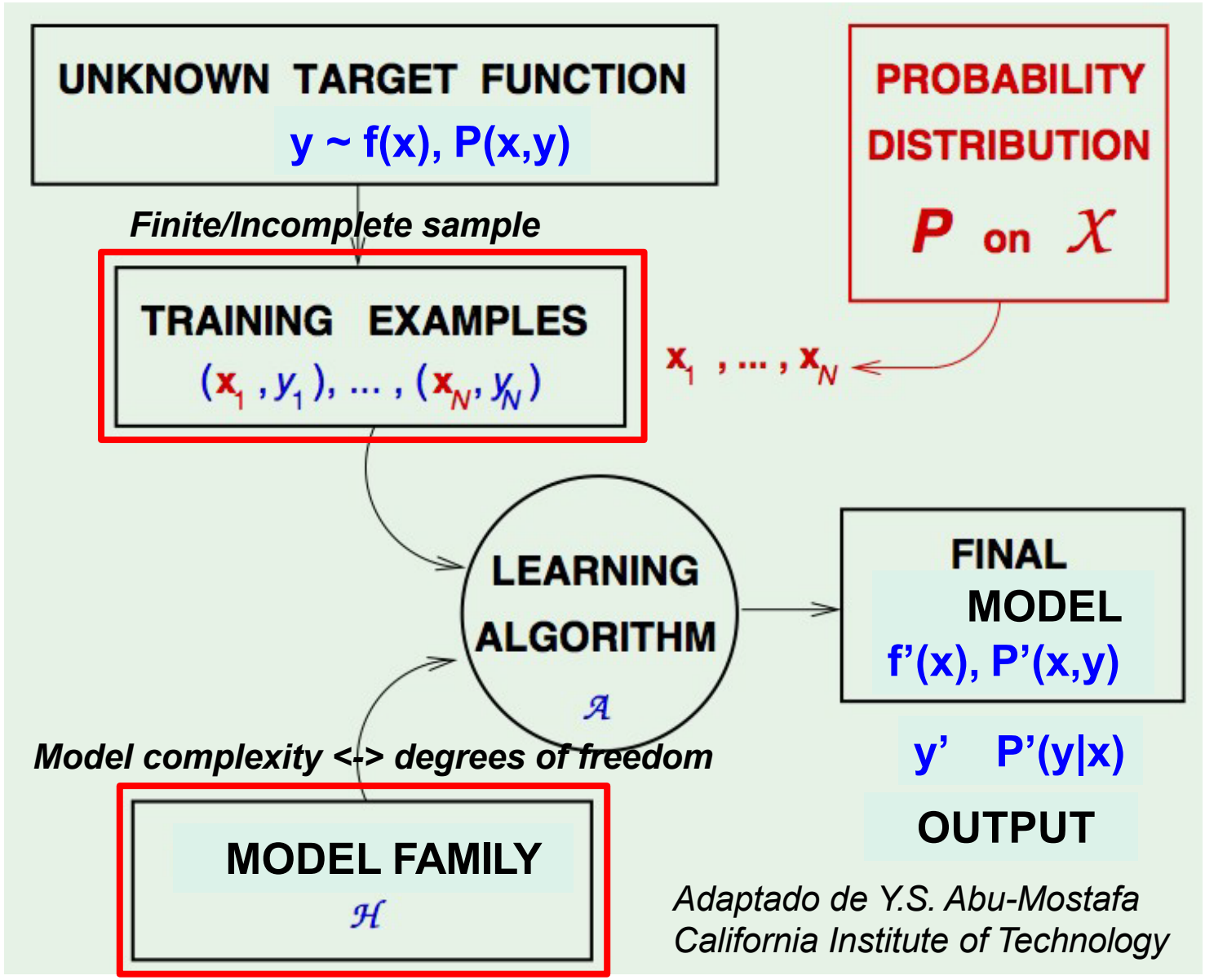
MODEL FAMILY

$\mathcal{H}$

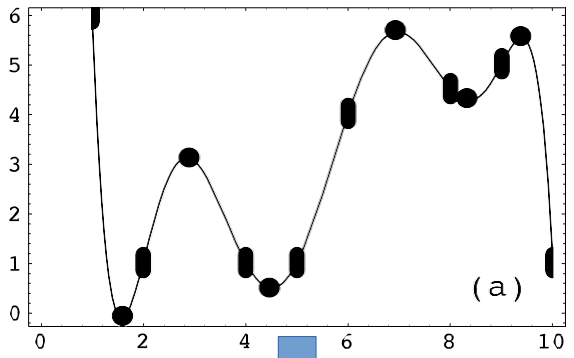
Adaptado de Y.S. Abu-Mostafa
California Institute of Technology

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

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

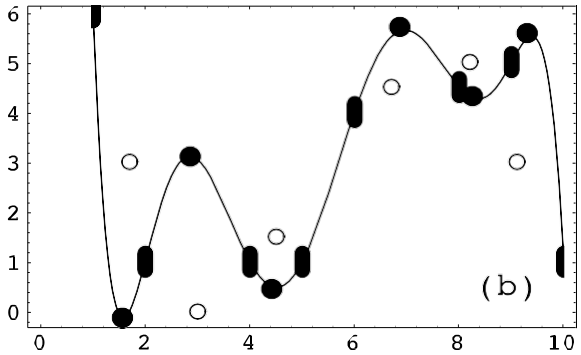


Adaptado de Y.S. Abu-Mostafa
California Institute of Technology



New sample

Overfitting



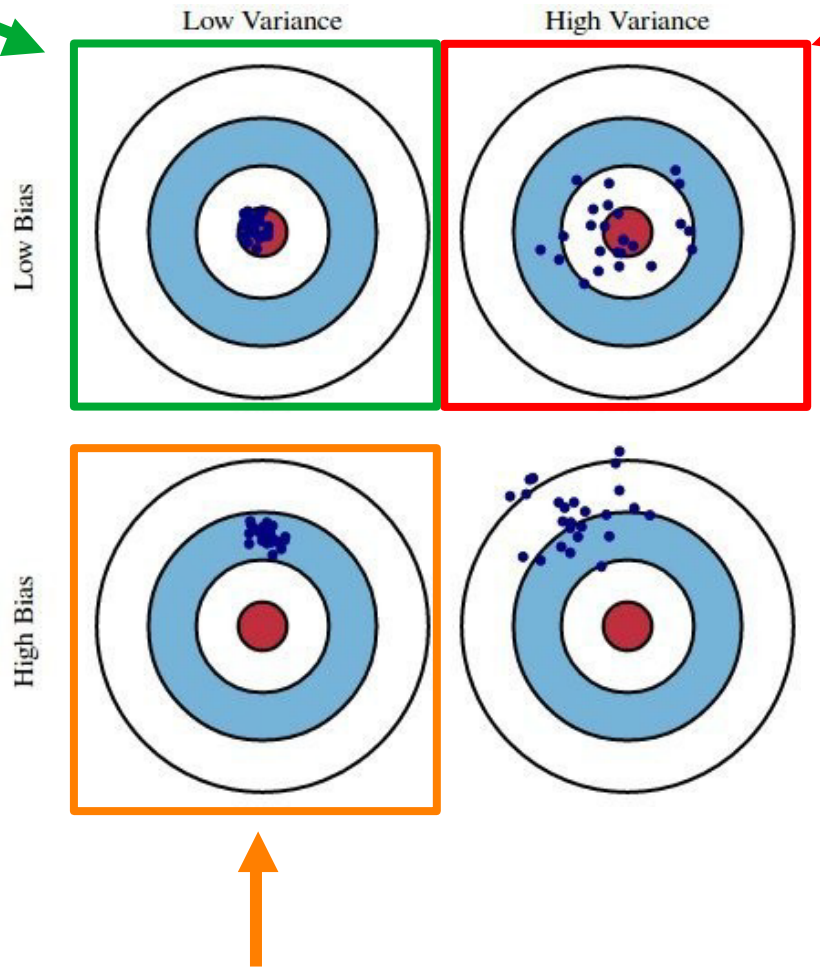
Trade-off between bias and variance

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

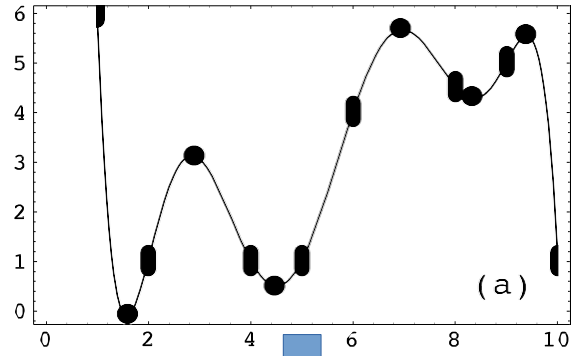
$$y = \sum_{i=0}^{N} (a_i * x^i)$$

Ajuste óptimo

Overfitting

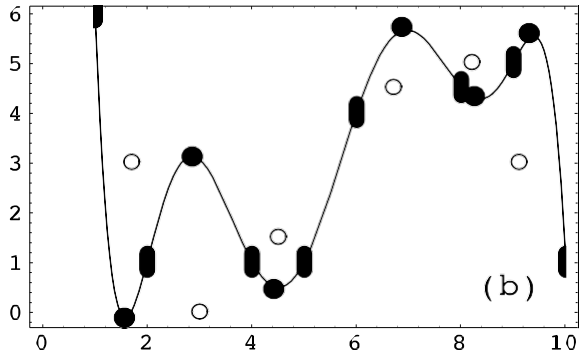


Underfitting



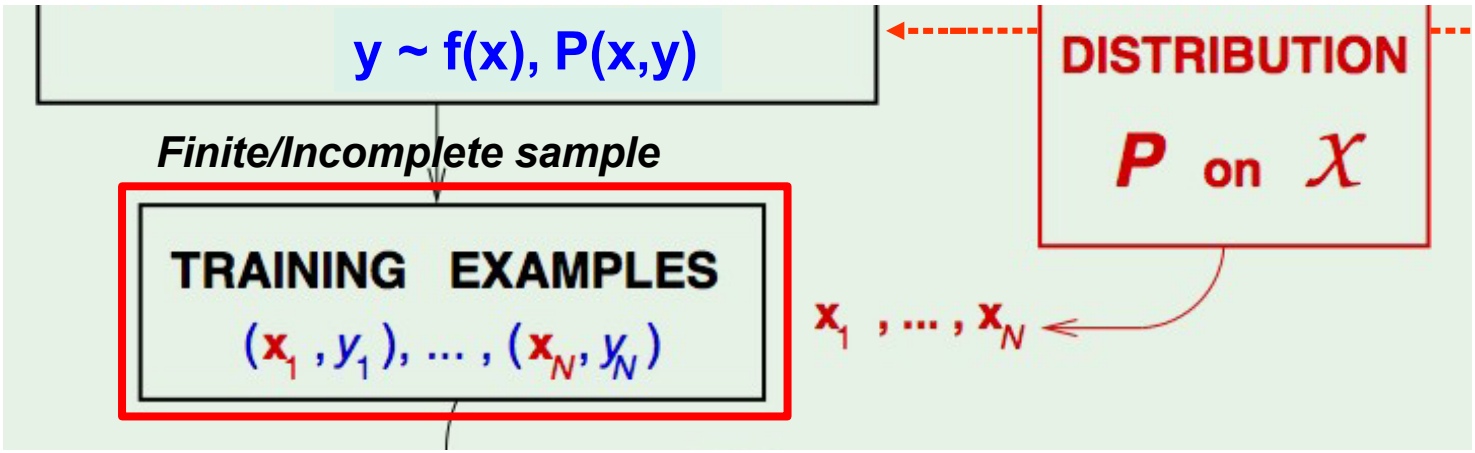
New sample

Overfitting



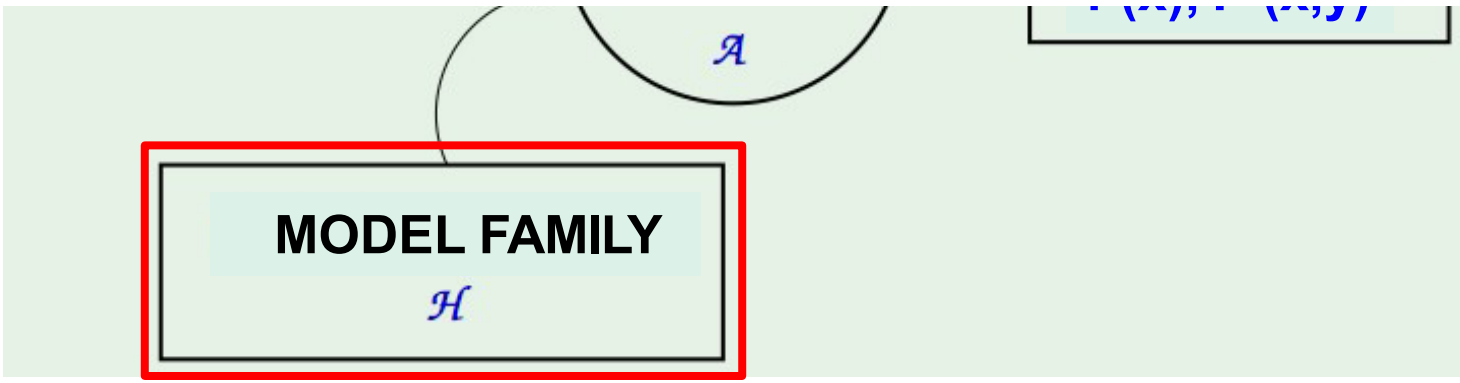
Trade-off between bias and variance

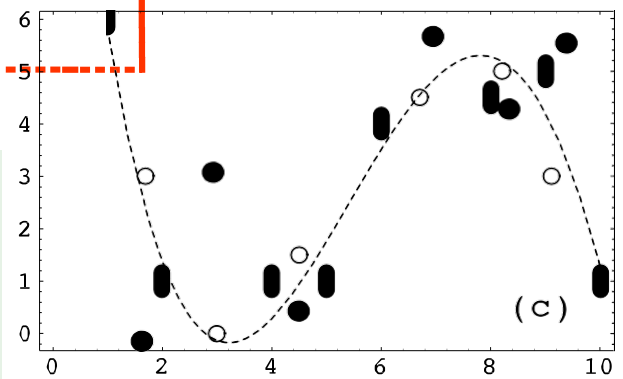
Generalization is the most important feature for data driven systems: They must perform "well" when applied to new data (**cross-validation**).



cross-validation

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

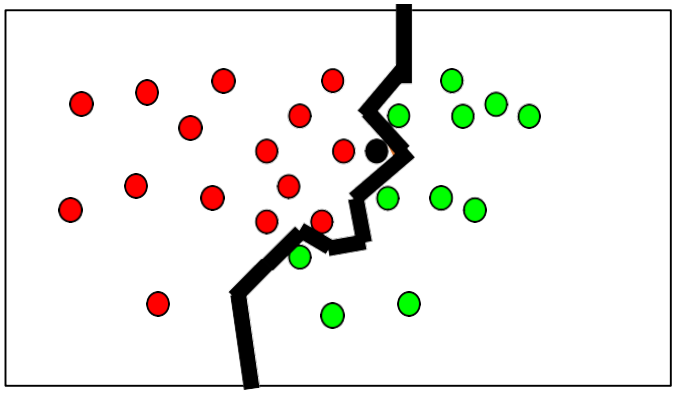


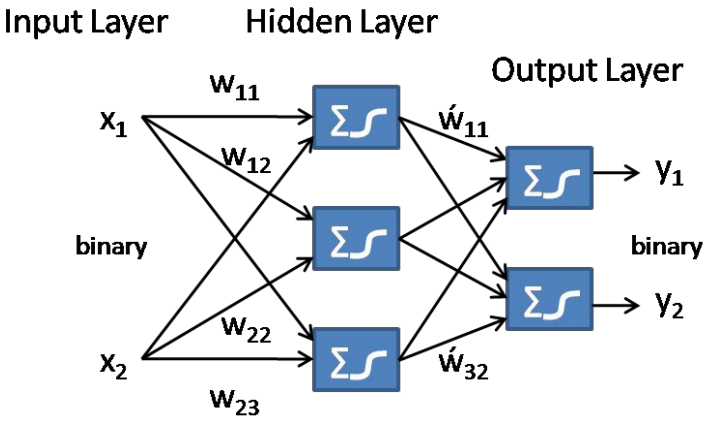


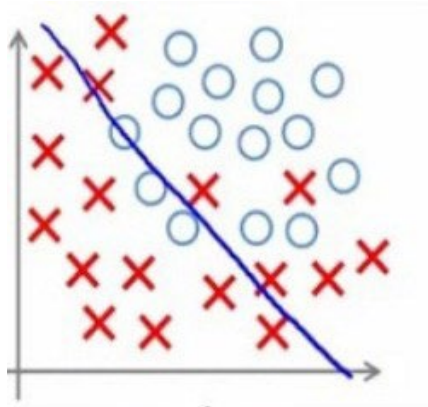
Trade-off between bias and variance

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

Increasing model complexity (e.g. number of parameters) can result in **overfitting** (lack of generalization).





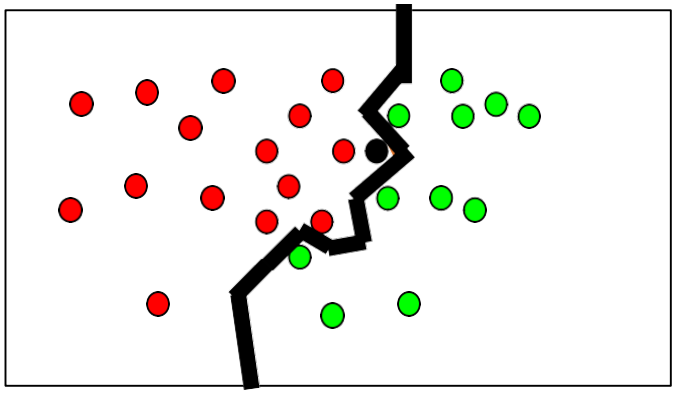


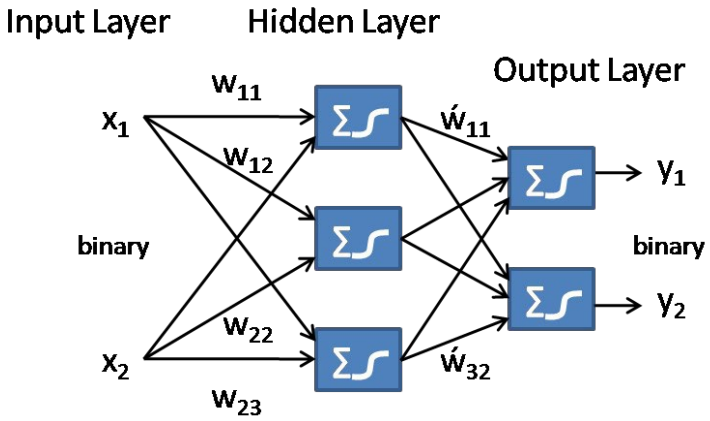
Under-fitting

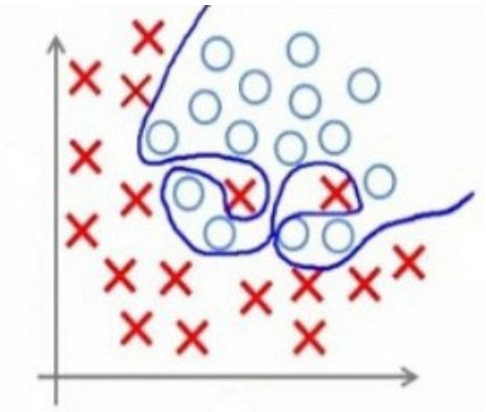
(too simple to explain the variance)

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

Increasing model complexity (e.g. number of parameters) can result in **overfitting** (lack of generalization).

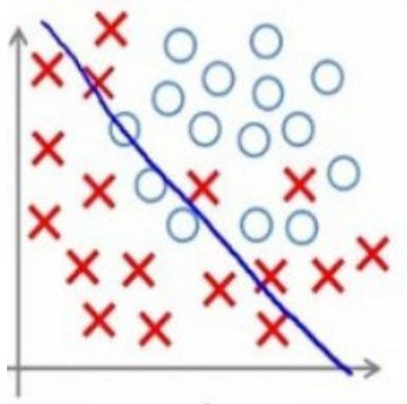






Over-fitting

(forcefitting -- too good to be true)

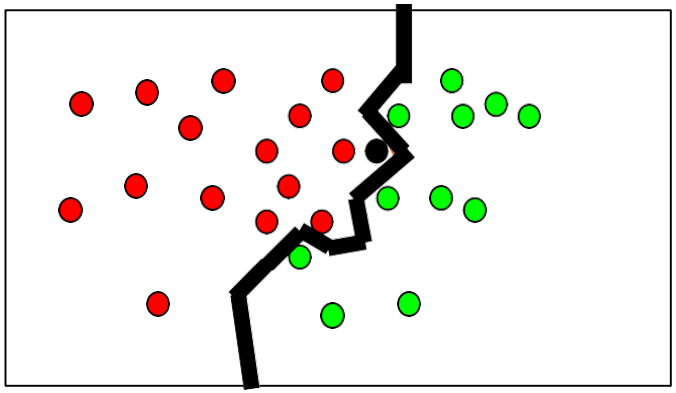


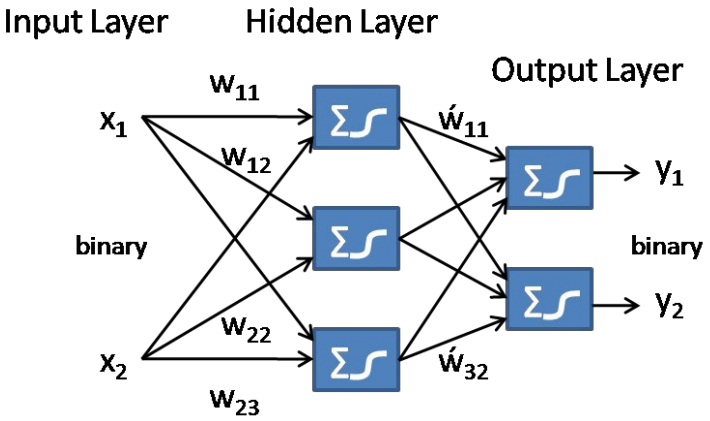
Under-fitting

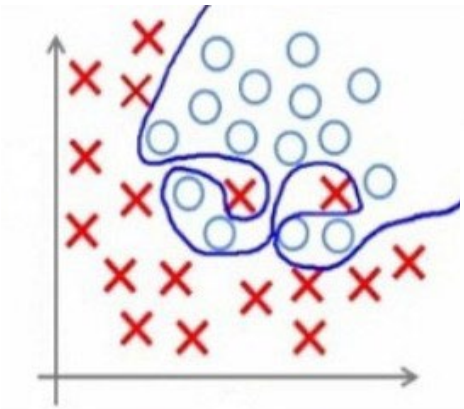
(too simple to explain the variance)

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

Increasing model complexity (e.g. number of parameters) can result in **overfitting** (lack of generalization).

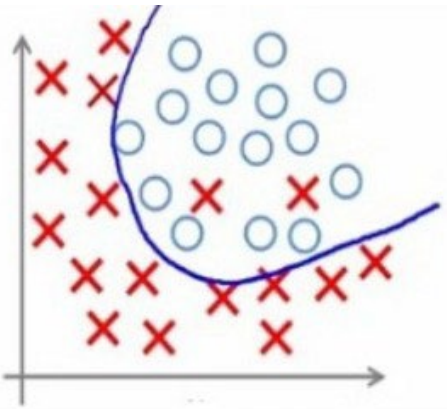




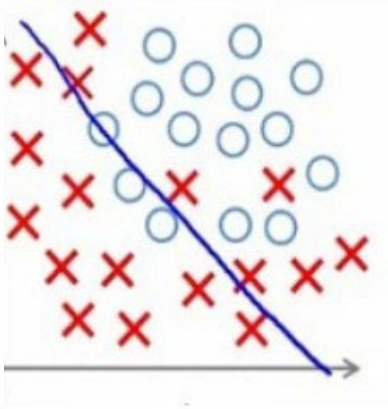


Over-fitting

(forcefitting -- too good to be true)



Appropriate-fitting



Under-fitting

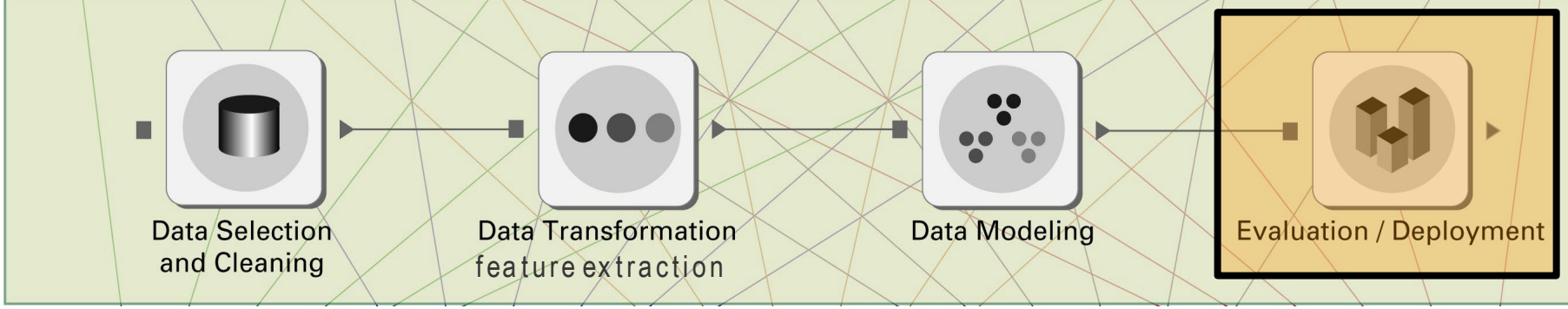
(too simple to explain the variance)

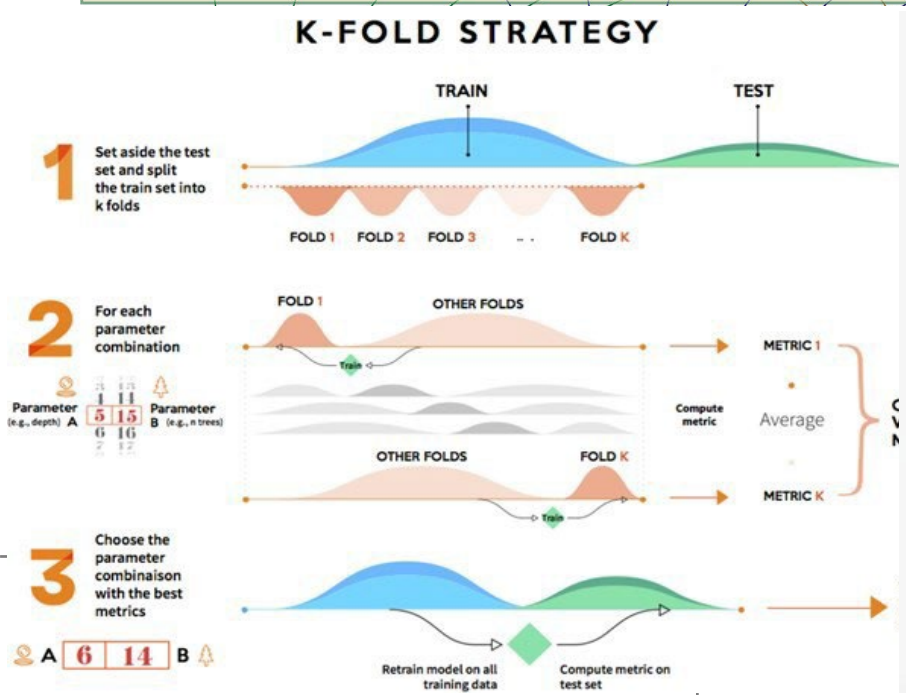
https://www.kdnuggets.com/2022/07/kfold-cross-validation.html

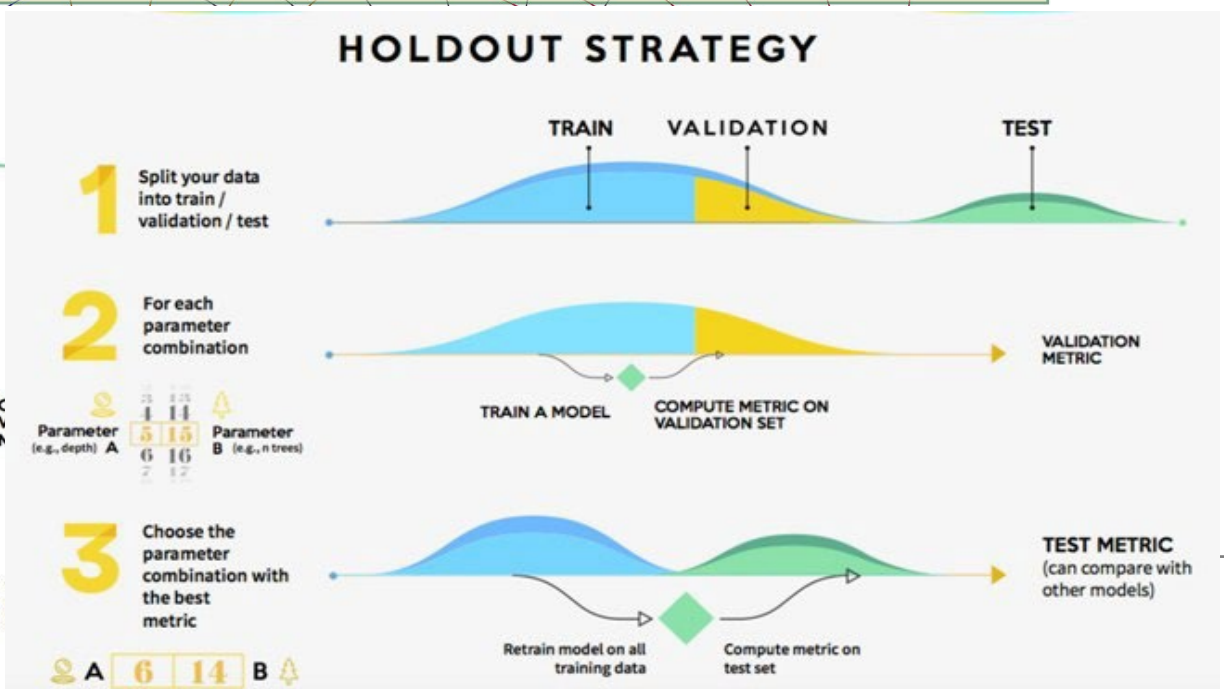
Sectores de aplicación

- Financiero Seguros
- Comercio y marketing
- Industria y empresarial
- Tecnologías información y comunicación
- Sanitario y farmacéutico
- Meteorología Medio Ambiente

Proceso de Minería de Datos







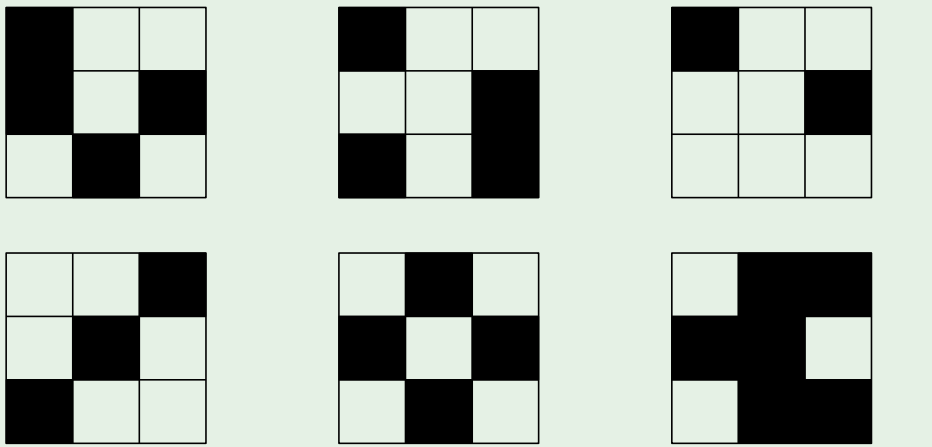
Generalization is the most important feature for data driven systems: They must perform "well" when applied to new data (**cross-validation**).

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?

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?

1



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

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

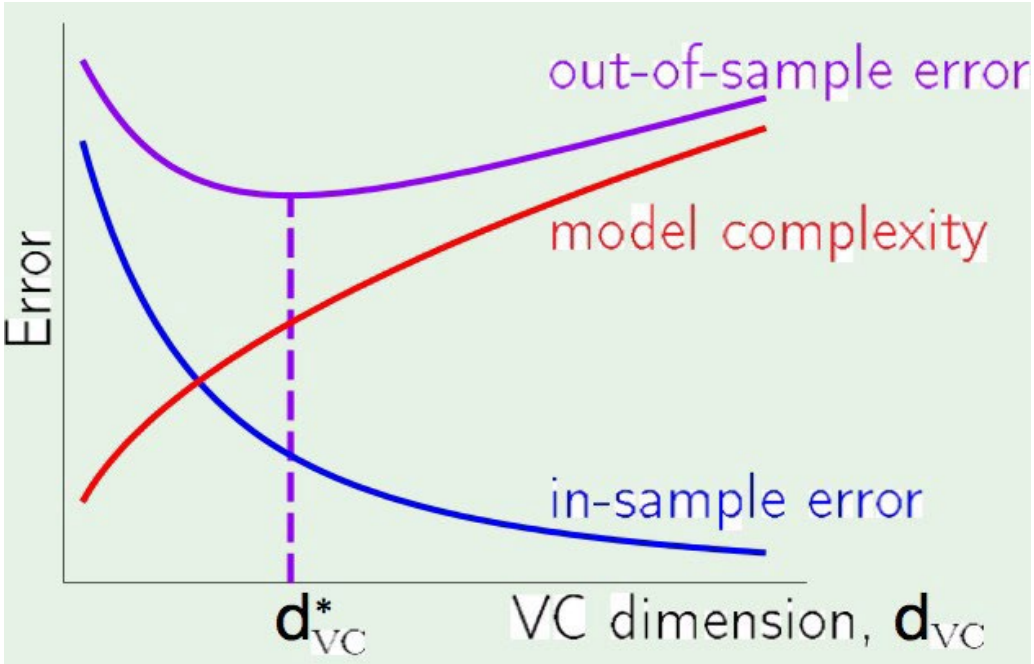
Generalization is the most important feature for data driven systems: They must perform "well" when applied to new data (**cross-validation**).

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

1

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

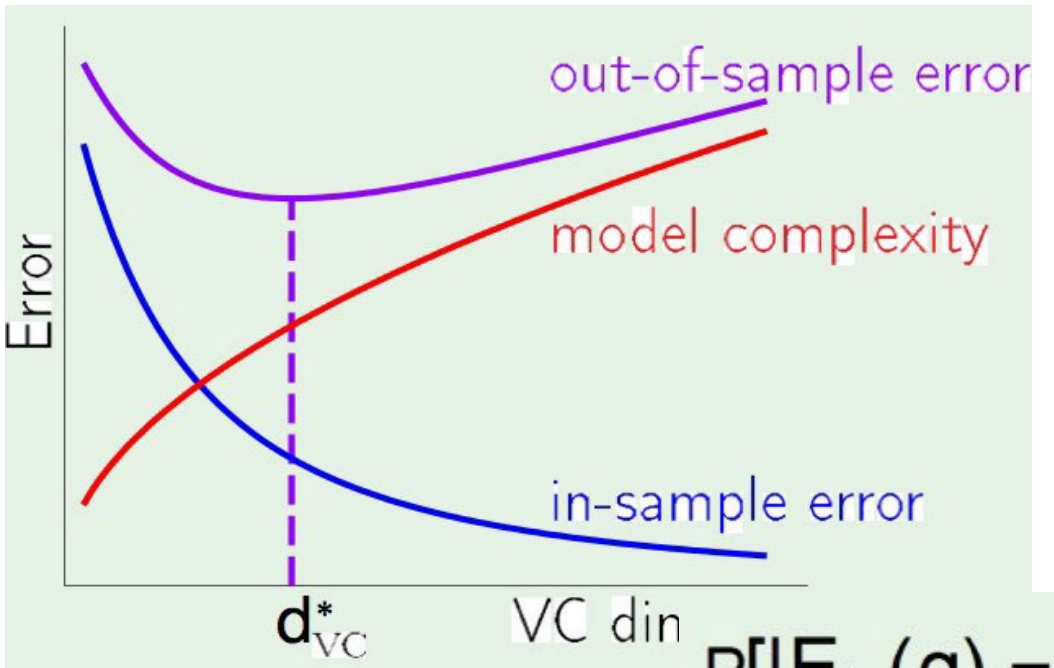
$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

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

$$\mathbb{P}[|v - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

N=sample size
M=model complexity

Vapnik-Chervonenkis (VC) Dimension

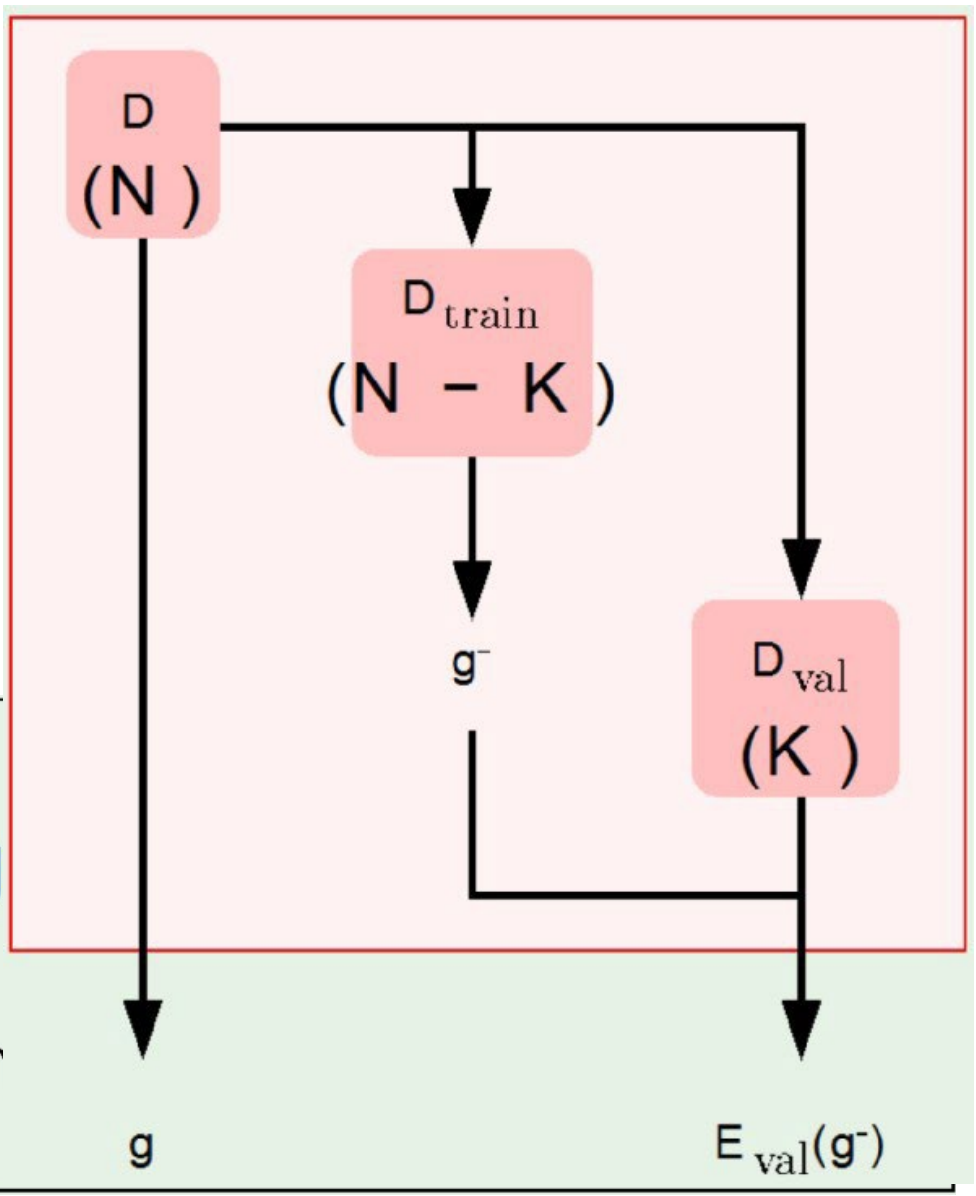


$$P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2M\, e^{-2\epsilon^2 N}$$

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 $E_{out}(g$
1
2. Can we make $E_{in}(g)$ small er

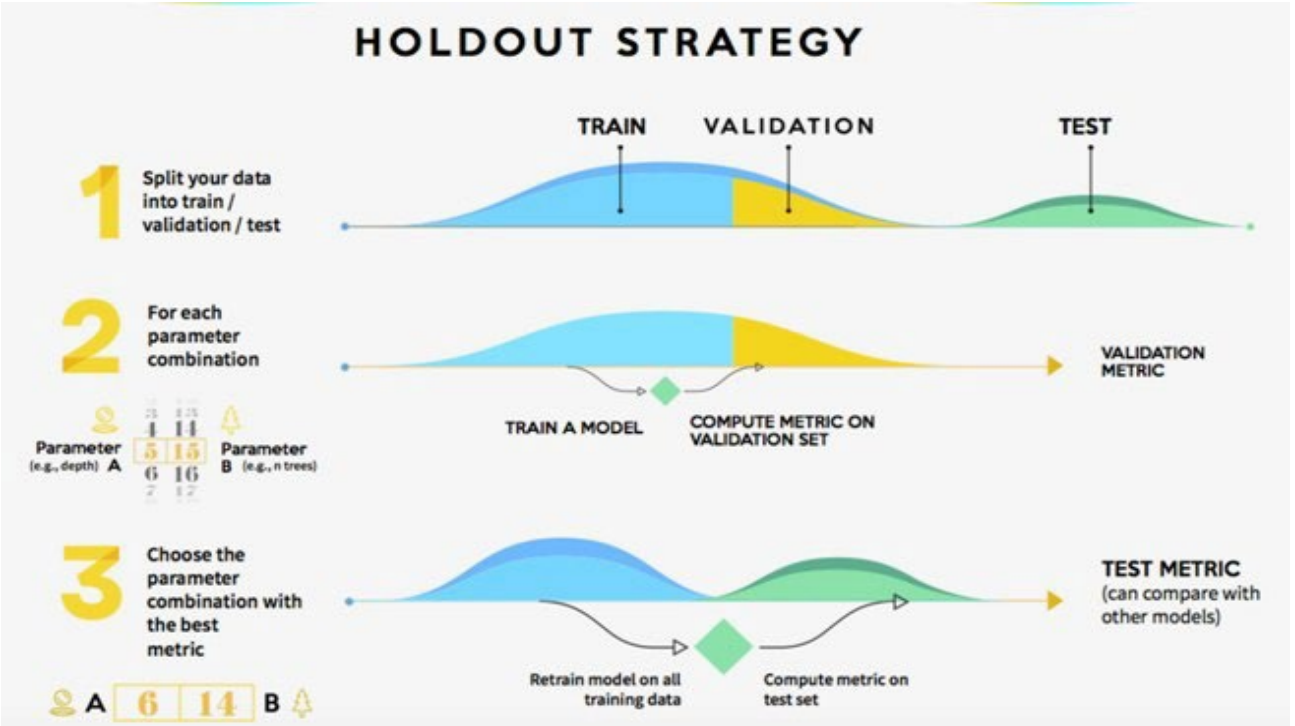
Generalization is the most important feature for data driven systems: They must perform "well" when applied to new data (**cross-validation**).The sample is divided in two subsets: **train** and **test**.

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

- **Modelo**:
1) Regresión
2) k-NN
3) Árboles de decisión
4) etc.

Parámetros:
1) Coeficientes
2) N.º vecinos
3) N.º árboles…
4) etc.

$$\vec{y}_{train} \simeq f(X_{train}, \vec{\theta})$$

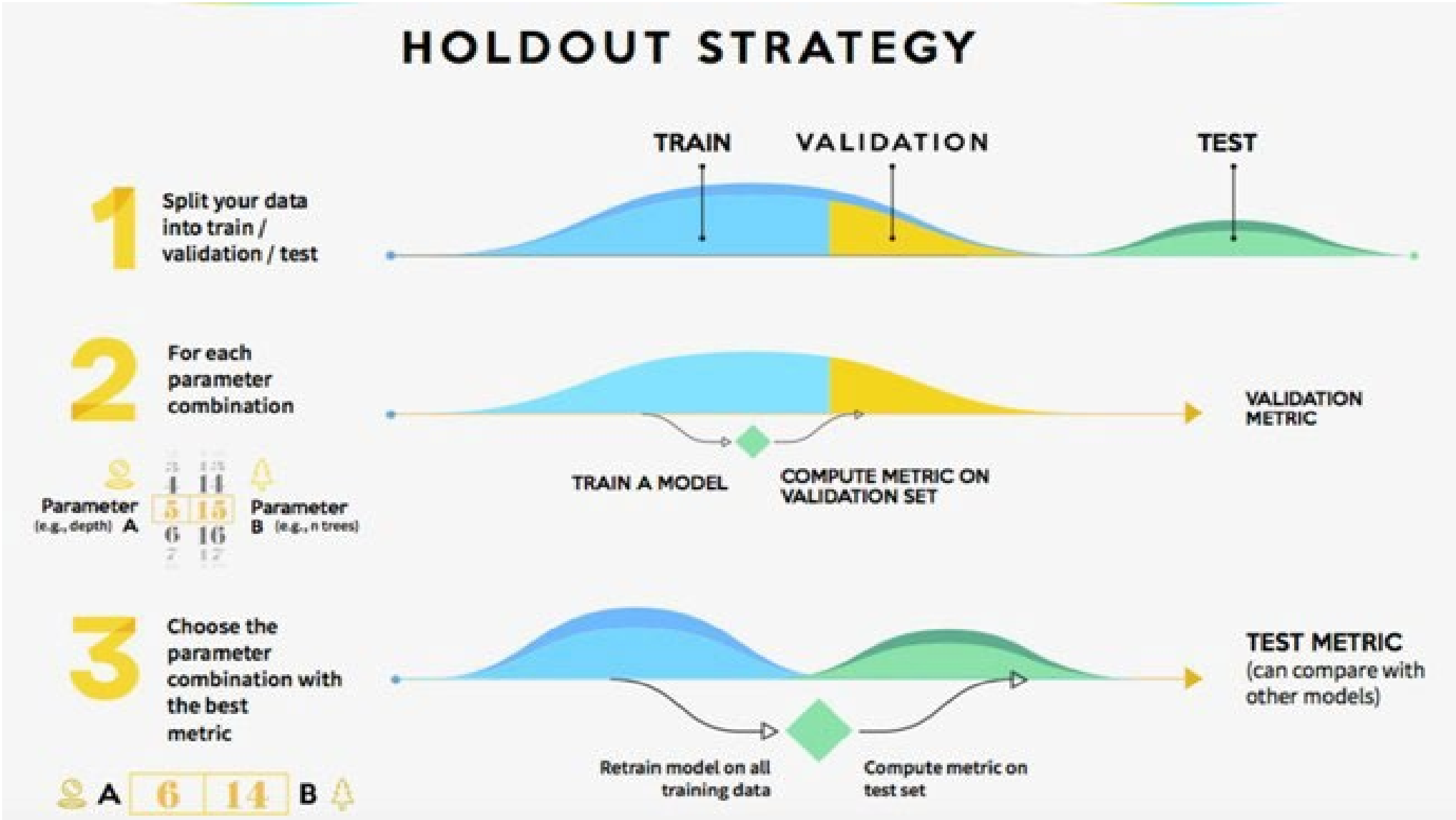


Source: Robert Kelley

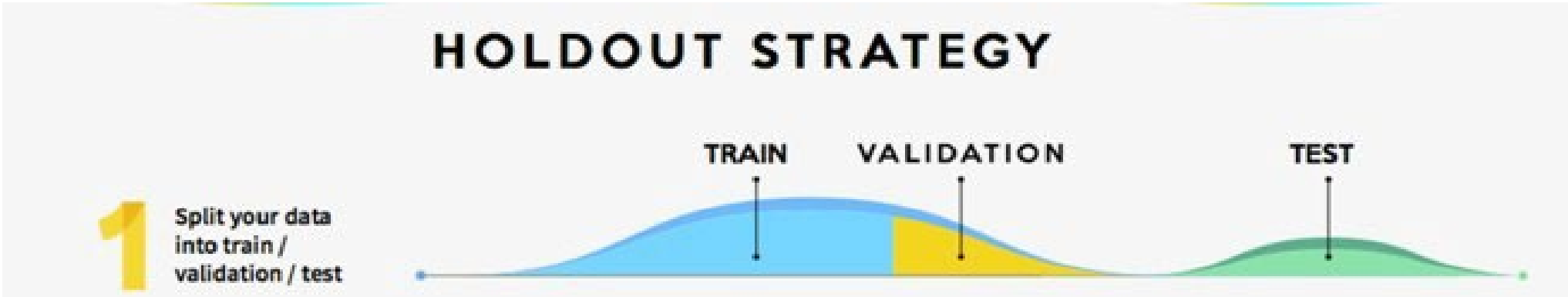
$$\vec{\hat{y}}_{val} = f(X_{val}, \vec{\theta})$$

$$\vec{\hat{y}}_{val} \text{ vs. } \vec{y}_{val} \Rightarrow E_{val} \simeq E_{test}$$

kaggle
What could the world's best analysts find in your data?



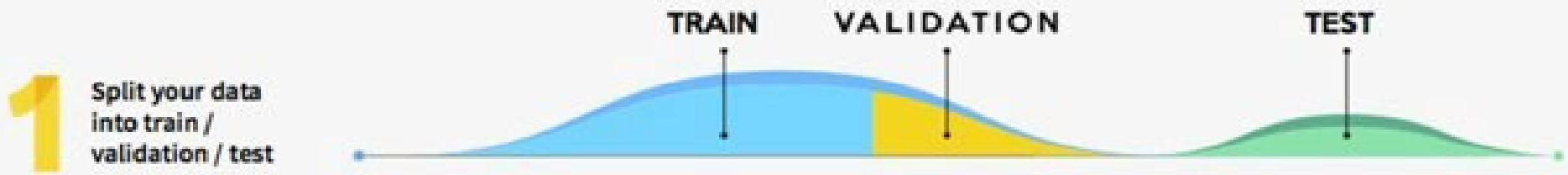
Source: [Robert Kelley]




```
plot(Altura, Peso)
train <- 1:ceiling(n/2)
order.index <- order(Peso)
Peso.sort <- Peso[order.index]
Altura.sort <- Altura[order.index]
points(Altura.sort[train], Peso.sort[train], pch=16, col="red")
mean.peso <- mean(Peso.sort[train])
abline(h=mean.peso)
# El error de test es mucho mayor ya que el modelo no generaliza.
mse.train <- mse(Peso.sort[train],mean.peso); mse.train
mse.test <- mse(Peso.sort[-train],mean.peso); mse.test
```

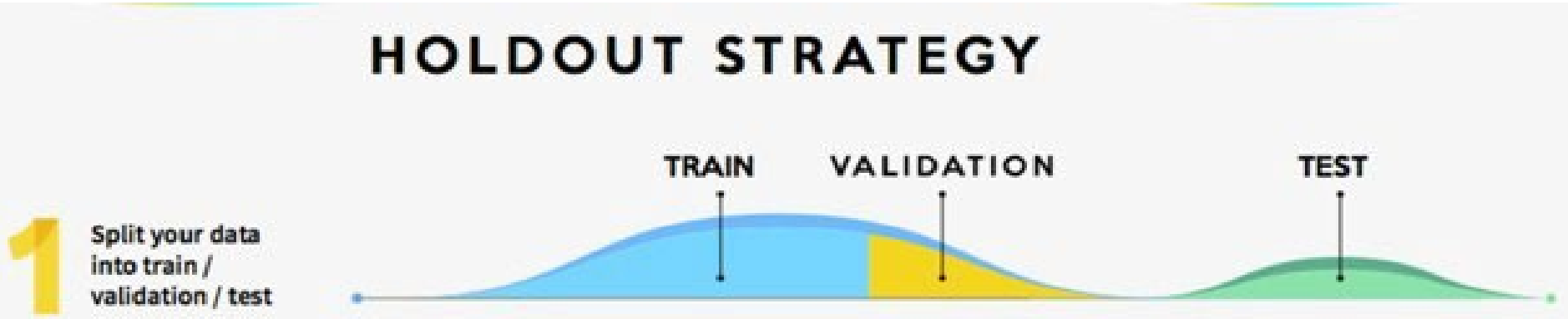
Source: Robert Kelley

HOLDOUT STRATEGY



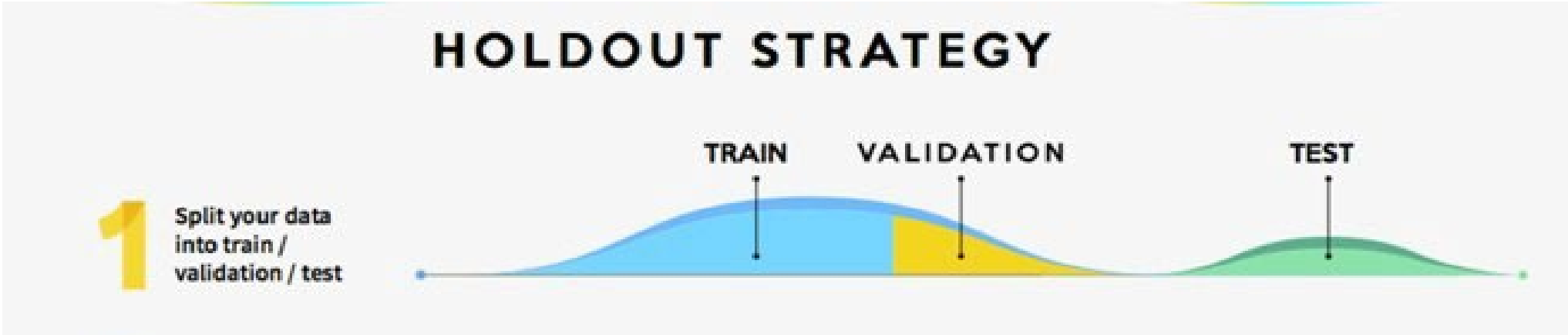

```
# Mejor si cogemos los datos aleatoriamente.
set.seed(1) # Para obtener el mismo valor fijamos la semilla
train <- sample(n,ceiling(n/2))
plot(Altura, Peso)
points(Altura[train], Peso[train], pch=16, col="red")
#y.est=cte esa cte es la media de la variable y selecionada en train
mean.peso <- mean(Peso[train])
abline(h=mean.peso)
mse.train <- mse(Peso[train],mean.peso); mse.train
mse.test <- mse(Peso[-train],mean.peso); mse.test
```

Source: Robert Kelley




```
# Sin embargo, hay una gran variabilidad respecto a la muestra
plot(Altura, Peso)
for (i in c(1:5)){
  train <- sample(n,ceiling(n/2))
  mean.peso <- mean(Peso[train])
  abline(h=mean.peso)
  print(mse(Peso[-train],mean.peso))
}
```

Source: Robert Kelley




```
# El problema se agudiza al incrementar la complejidad del modelo
set.seed(1)
train <- sample(n,ceiling(n/2))
plot(Altura, Peso)
points(Altura[train], Peso[train], pch=16, col="red")
Reg.2<-lm(Peso~Altura, data=Pulsaciones, subset=train)
yest.2 <- predict(Reg.2, data.frame(Altura=Altura[-train]))
mse.Reg.2<-mse(Peso[-train],yest.2); mse.Reg.2
yest.2.train <- predict(Reg.2, data.frame(Altura=Altura[train]))
mse.Reg.2.train<-mse(Peso[train],yest.2.train); mse.Reg.2.train
```

Bajo coste computacional

Parte de la muestra disponible para entrenar el modelo no se aprovecha

Estimación de E_{test} poco robusta (sensibilidad a la partición train/validation)

Source: Robert Kelley

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:**
  - Variability related with the train-test splitting.
  - The sample size of the test sample leads to conservative results.
  - The sample size of the train sample limits the complexity of the model.
- Leave-one-out
- K-fold

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?

1

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:
  - Variability related with the train-test splitting.
  - The sample size of the test sample leads to conservative results.
  - The sample size of the train sample limits the complexity of the model.
- **Leave-one-out**
- K-fold

$N - 1$ points for training, and 1 point for validation!

$$D_n = (x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), \cancel{(x_n, y_n)}, (x_{n+1}, y_{n+1}), \ldots, (x_N, y_N)$$

Final hypothesis learned from D_n is g_n^-

$$e_n = E_{\text{val}}(g_n^-) = e(g_n^-(x_n), y_n)$$

cross validation error: $$E_{\text{cv}} = \frac{1}{N} \sum_{n=1}^{N} e_n$$

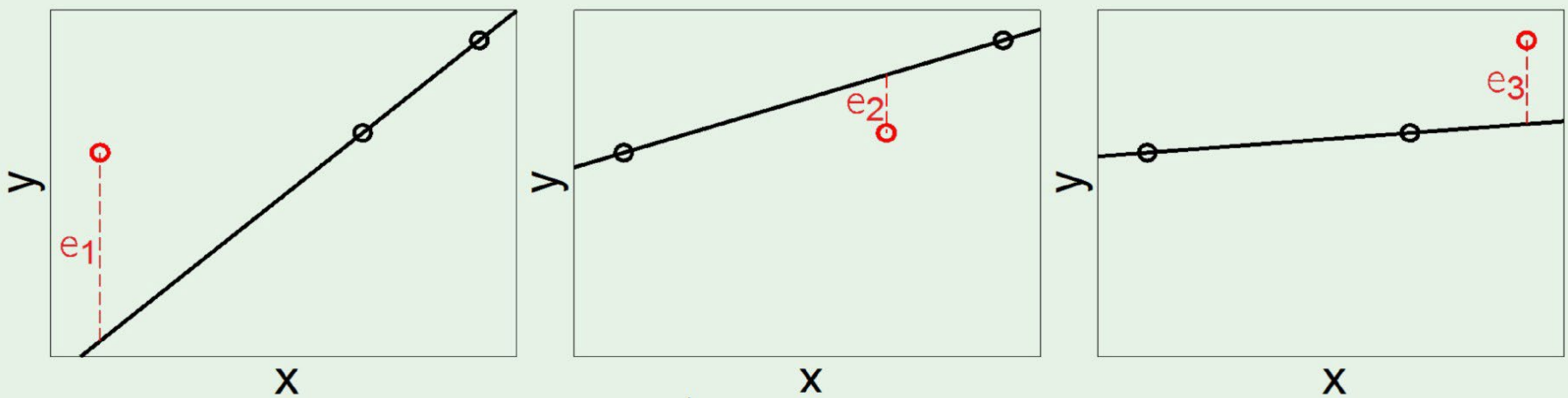
Generalization is the most important feature for data driven systems: They must perform "well" when applied to new data (**cross-validation**).

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

- Hold-out:
  - Variability related with the train-test splitting.
  - The sample size of the test sample leads to conservative results.
  - The sample size of the train sample limits the complexity of the model.
- **Leave-one-out**
- K-fold

$N - 1$ points for training, and 1 point for validation!

$$D_n = (x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), \cancel{(x_n, y_n)}, (x_{n+1}, y_{n+1}), \ldots, (x_N, y_N)$$



$$E_{cv} = \frac{1}{3}(e_1 + e_2 + e_3)$$

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:
  - Variability related with the train-test splitting.
  - The sample size of the test sample leads to conservative results.
  - The sample size of the train sample limits the complexity of the model.
- **Leave-one-out:**
  - High computational cost (**small samples**).
- K-fold

- Aprovechamiento máximo de la muestra disponible para entrenar
- Mejor estimación de E_{test} posible
- Muy Alto coste computacional

$$\vec{y}^1_{train} \simeq f(X^1_{train}, \vec{\theta}^1) \quad \vec{y}^2_{train} \simeq f(X^2_{train}, \vec{\theta}^2) \quad \vec{y}^3_{train} \simeq f(X^3_{train}, \vec{\theta}^3) \quad \cdots\cdots \quad \vec{y}^4_{train} \simeq f(X^4_{train}, \vec{\theta}^4)$$

$$\vec{y}^1_{val} - f(X^1_{val}, \vec{\theta}^1) \quad \vec{y}^2_{val} = f(X^2_{val}, \vec{\theta}^2) \quad \vec{y}^3_{val} = f(X^3_{val}, \vec{\theta}^3) \quad \cdots\cdots \quad \vec{y}^4_{val} = f(X^4_{val}, \vec{\theta}^4)$$

$$\vec{y}^1_{val} \text{ vs. } \vec{y}^1_{val} \Rightarrow E^1_{val} \quad \vec{y}^2_{val} \text{ vs. } \vec{y}^2_{val} \Rightarrow E^2_{val} \quad \vec{y}^3_{val} \text{ vs. } \vec{y}^3_{val} \Rightarrow E^3_{val} \quad \cdots\cdots \quad \vec{y}^4_{val} \text{ vs. } \vec{y}^4_{val} \Rightarrow E^4_{val}$$

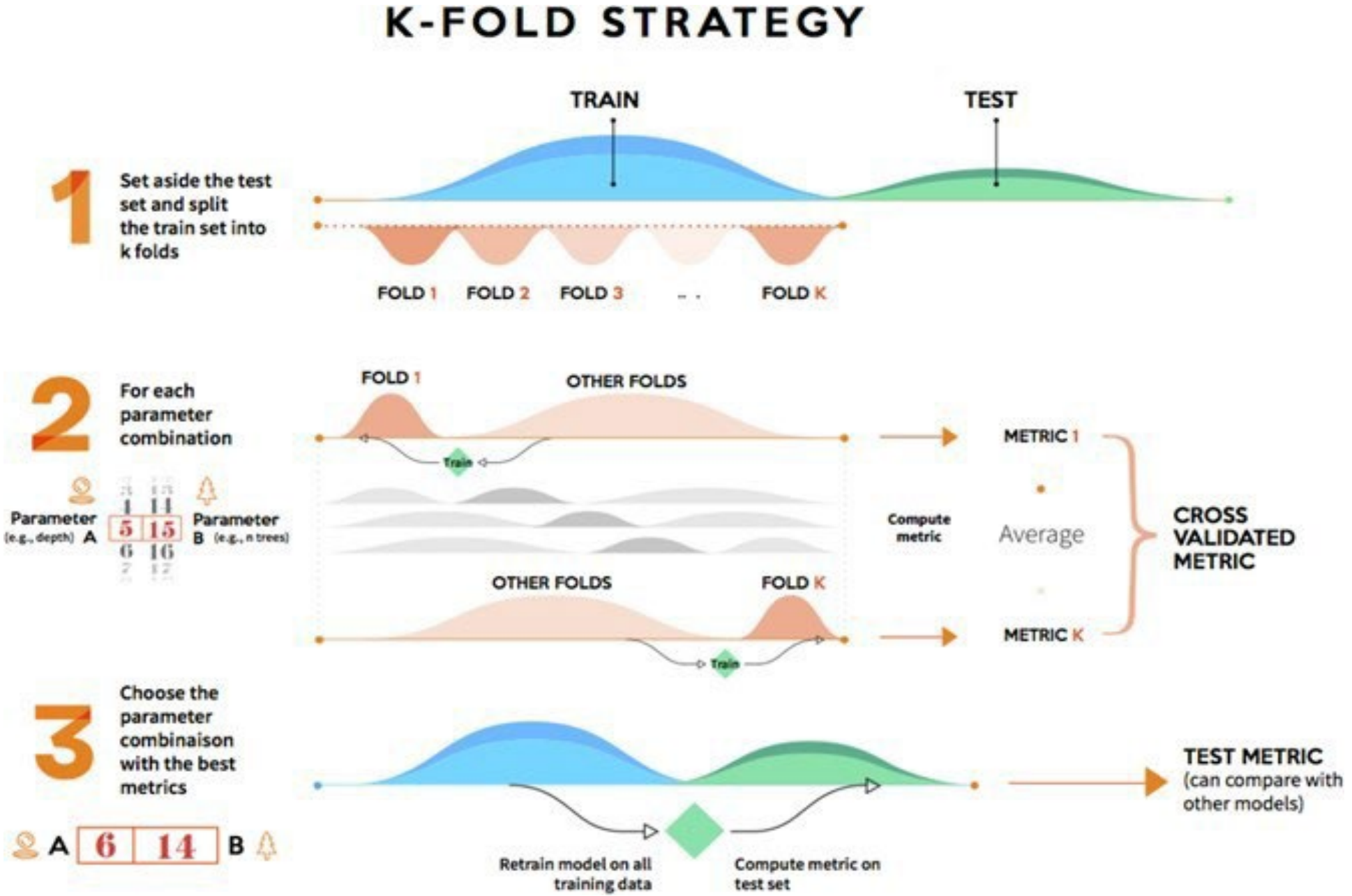
$$\frac{1}{k}\sum_{i=1}^{k} E^i_{val} \simeq E_{test}$$

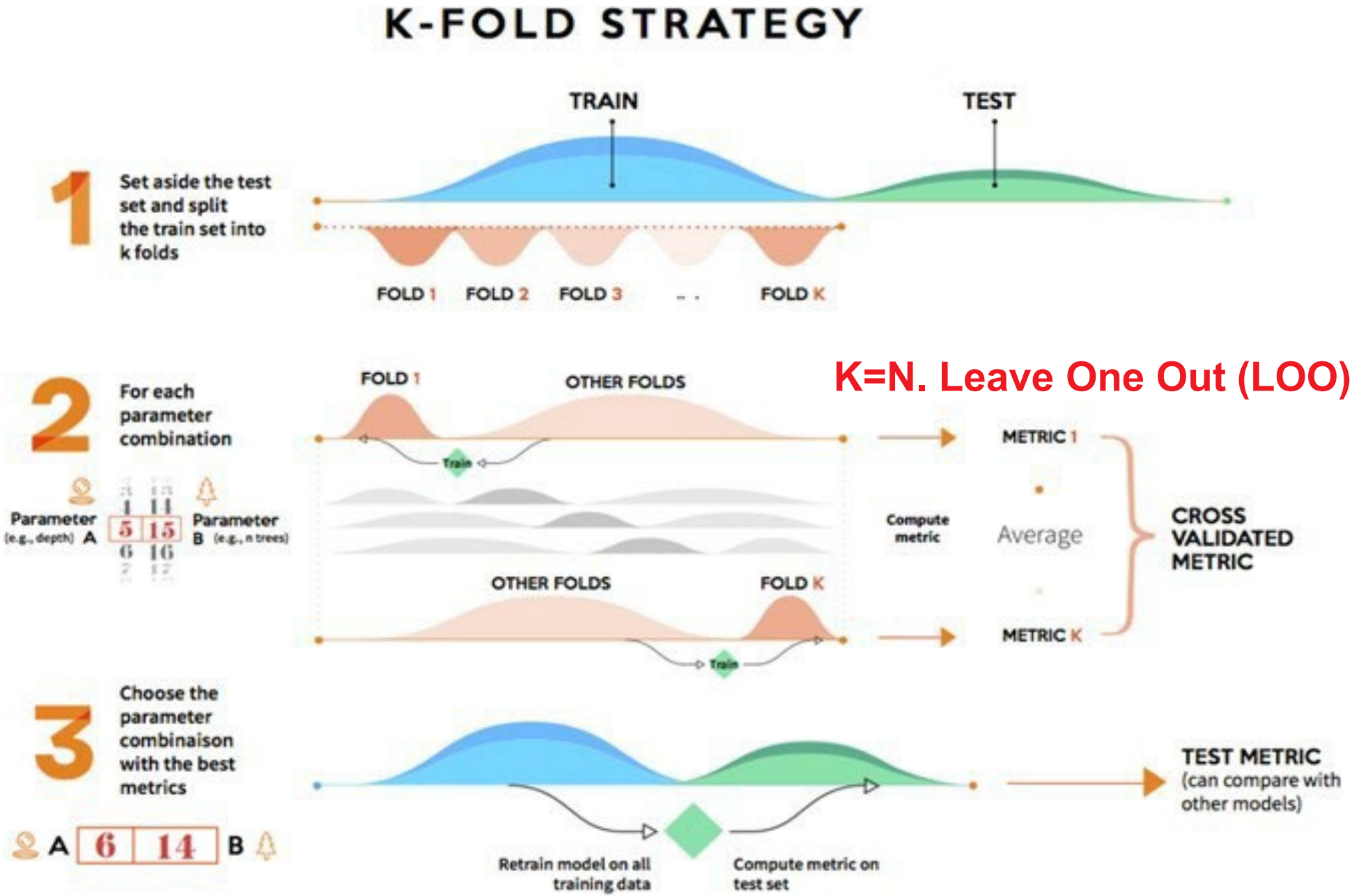
Generalization is the most important feature for data driven systems: They must perform “well” when applied to new data (**cross-validation**).

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

- Hold-out:
  - Variability related with the train-test splitting.
  - The sample size of the test sample leads to conservative results.
  - The sample size of the train sample limits the complexity of the model.
- **Leave-one-out:**
  - High computational cost (**small samples**).
- K-fold

```
# Leave-One-Out Cross-Validation
yest.3<-rep(NA, length(train)) # La actualización es ineficiente
train <- 1:n
for (i in train){
  Reg.i<-lm(Peso~Altura, data=Pulsaciones, subset=train[-i])
  yest.3[i]<-predict(Reg.i,data.frame(Altura=Altura[i]))
}
mse.Reg.3<-mse(Peso,yest.3); mse.Reg.3
```


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
Parameter (e.g., depth) A
5
15
Parameter B (e.g., n trees)
FOLD 1
OTHER FOLDS
Train
Compute metric
METRIC 1
Average
CROSS VALIDATED METRIC
OTHER FOLDS
FOLD K
Train
METRIC K
3
Choose the parameter combinaison with the best metrics
A
6
14
B
Retrain model on all training data
Compute metric on test set
TEST METRIC
(can compare with other models)


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
K=N. Leave One Out (LOO)
2
For each parameter combination
Parameter (e.g., depth) A
Parameter B (e.g., n trees)
FOLD 1
OTHER FOLDS
Train
Compute metric
METRIC 1
Average
CROSS VALIDATED METRIC
OTHER FOLDS
FOLD K
Train
METRIC K
3
Choose the parameter combinaison with the best metrics
A
6
14
B
Retrain model on all training data
Compute metric on test set
TEST METRIC
(can compare with other models)

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:
  - Variability related with the train-test splitting.
  - The sample size of the test sample leads to conservative results.
  - The sample size of the train sample limits the complexity of the model.
- Leave-one-out:
  - High computational cost (**small samples**).
- **K-fold:**
  - Symilar results than leave-one-out with low number of folds.
  - Statistical analysis of the validation measures.

Appears in the International Joint Conference on Artificial Intelligence (IJCAI), 1995

A Study of Cross-Validation and Bootstrap
for Accuracy Estimation and Model Selection

Ron Kohavi
Computer Science Department
Stanford University
Stanford, CA. 94305

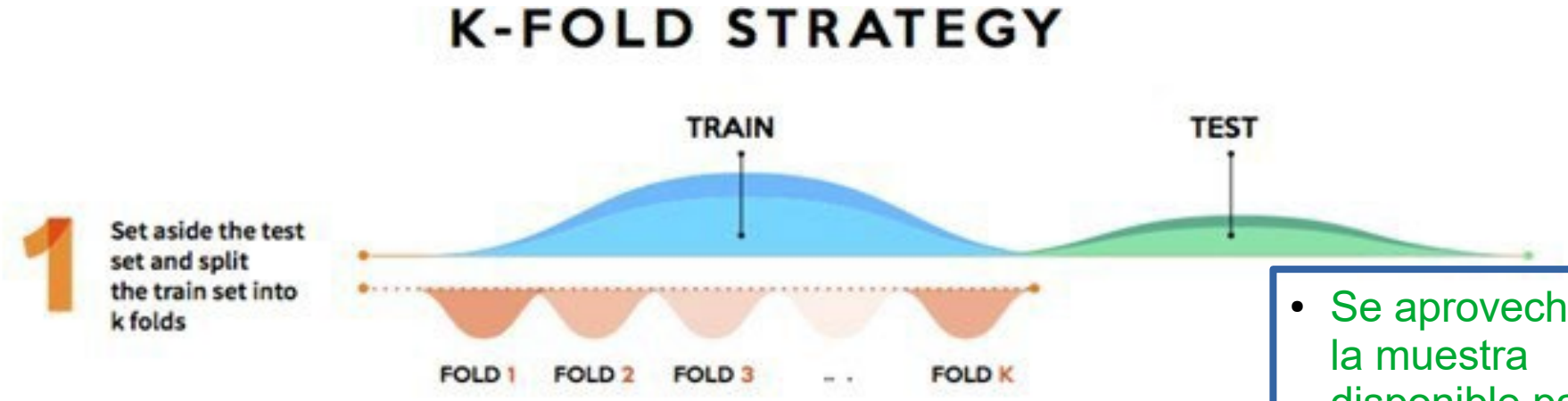
Over 3000 scitations

Reassessing Statistical Downscaling Techniques for Their Robust Application under Climate Change Conditions

J. M. Gutiérrez ✉; D. San-Martín; S. Brands; R. Manzanas; S. Herrera

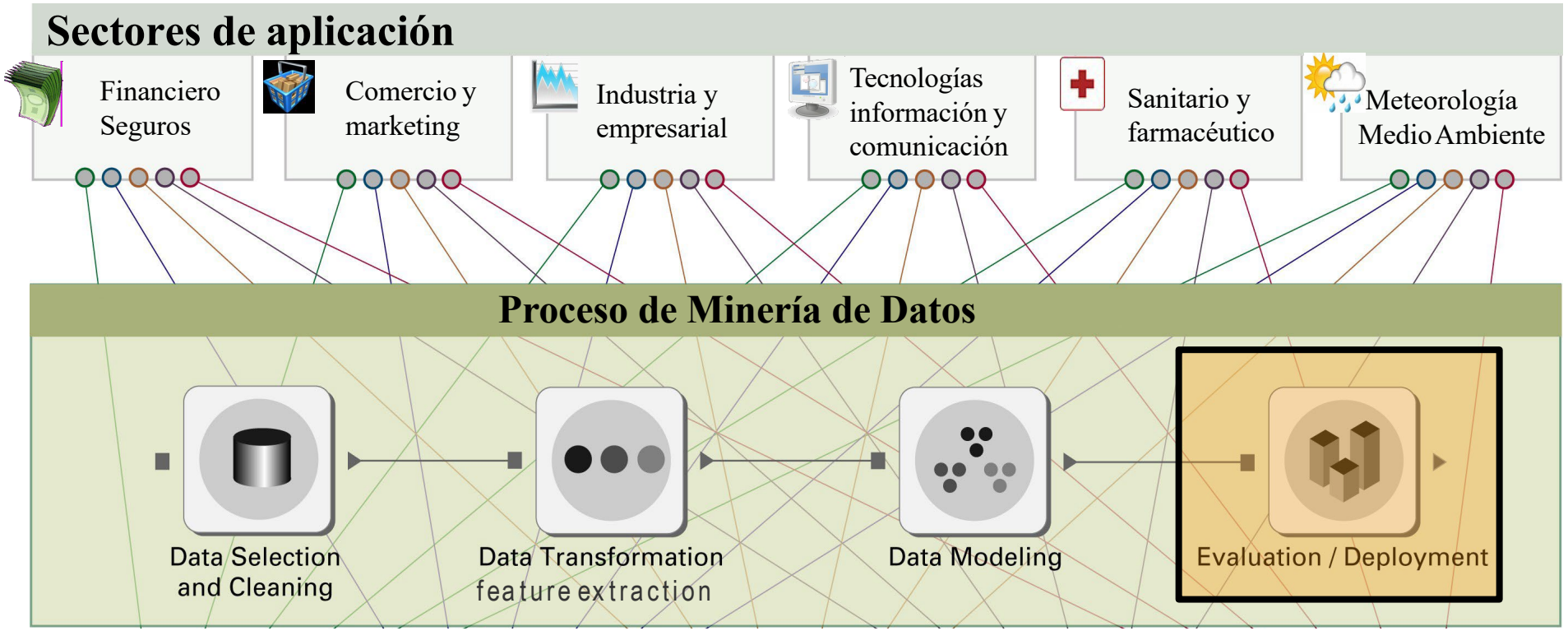
J. Climate (2013) **26** (1): 171–188.
https://doi.org/10.1175/JCLI-D-11-00687.1 Article history

Cros
Valida



- Se aprovecha toda la muestra disponible para entrenar
- Estimación de E_{test} más robusta
- Alto coste computacional

```
# 10-Fold Cross-Validation
idx.aleatorios <- sample(1:n,n,replace=F)
K <- 10
tam <- ceiling(n/K)
yest4 <- rep(NA, length(train)) # La actualización es ineficiente
for (i in 0:(K-1)){
  idx.test <- idx.aleatorios[(i*tam+1):((i+1)*tam)]
  idx.test <- idx.test[!is.na(idx.test)]
  lm4 <- lm(Peso~Altura, subset=-idx.test)
  yest4[idx.test] <- predict(lm4, data.frame(Altura=Altura[idx.test]))
}
mse4 <- mse(Peso,yest4); mse4
```

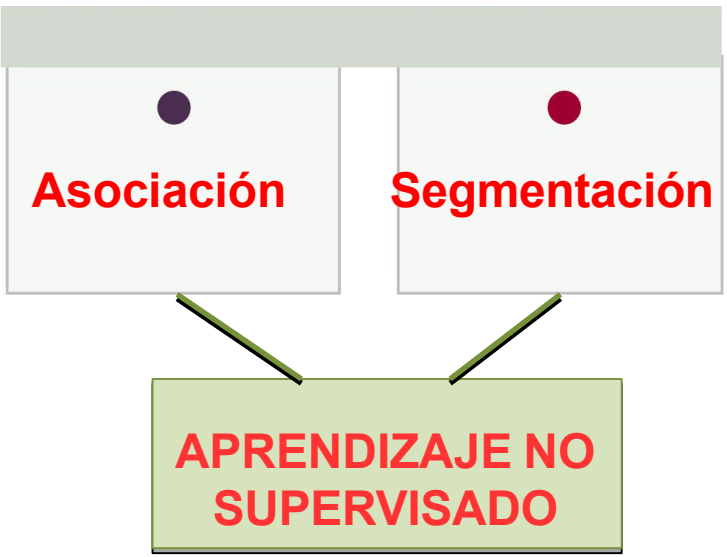

Sectores de aplicación
Financiero
Seguros
Comercio y
marketing
Industria y
empresarial
Tecnologías
información y
comunicación
Sanitario y
farmacéutico
Meteorología
Medio Ambiente
Proceso de Minería de Datos
Data Selection
and Cleaning
Data Transformation
feature extraction
Data Modeling
Evaluation /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

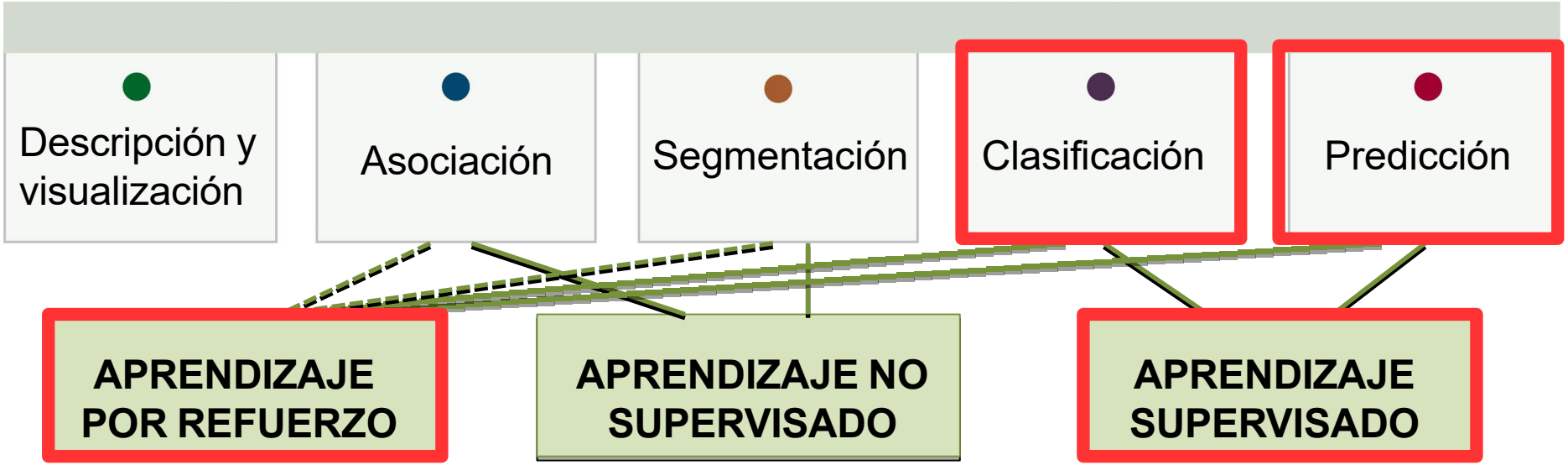
- Target Variable:
  - *Y (discrete/factor* or *continuous)*
- Predictive Model → $Y = f(X_1, X_2, \ldots, X_N)$

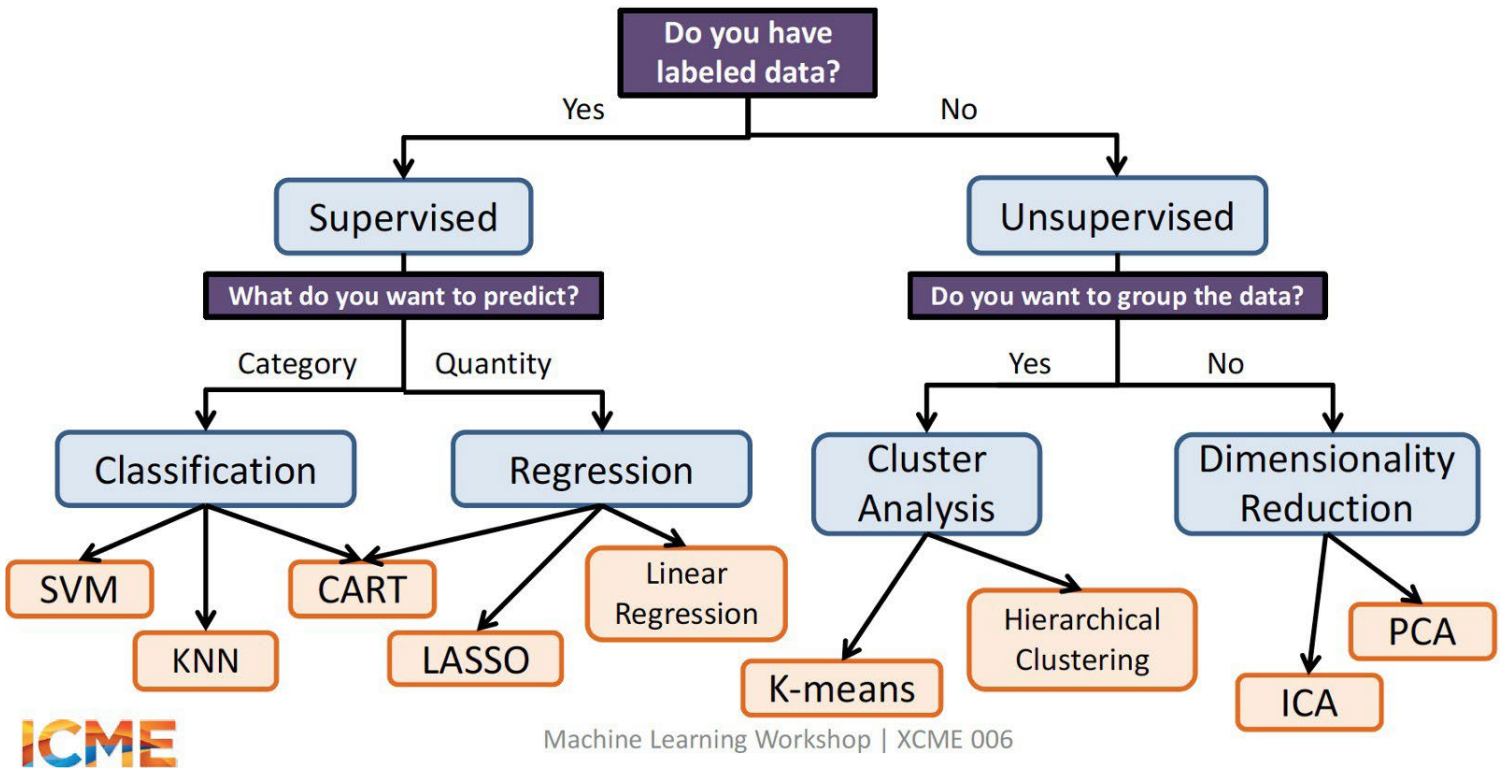
  Predictand → Y; Predictors → X_i

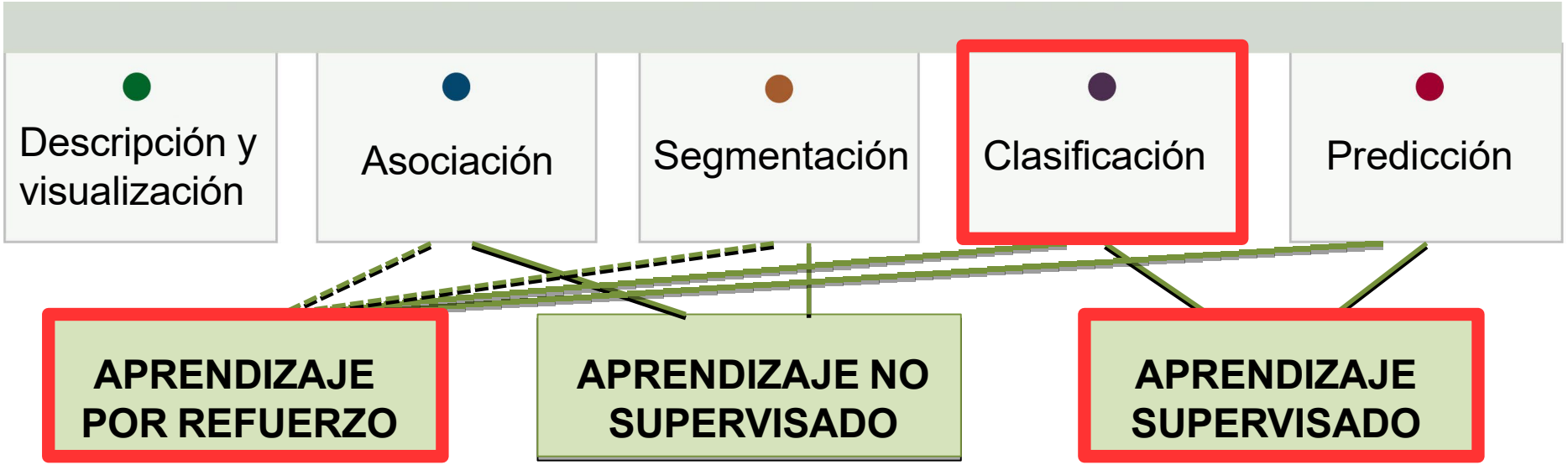
- There is no target variable:
  - *Association or segmentation*
- Predictive Model → Algorithmic

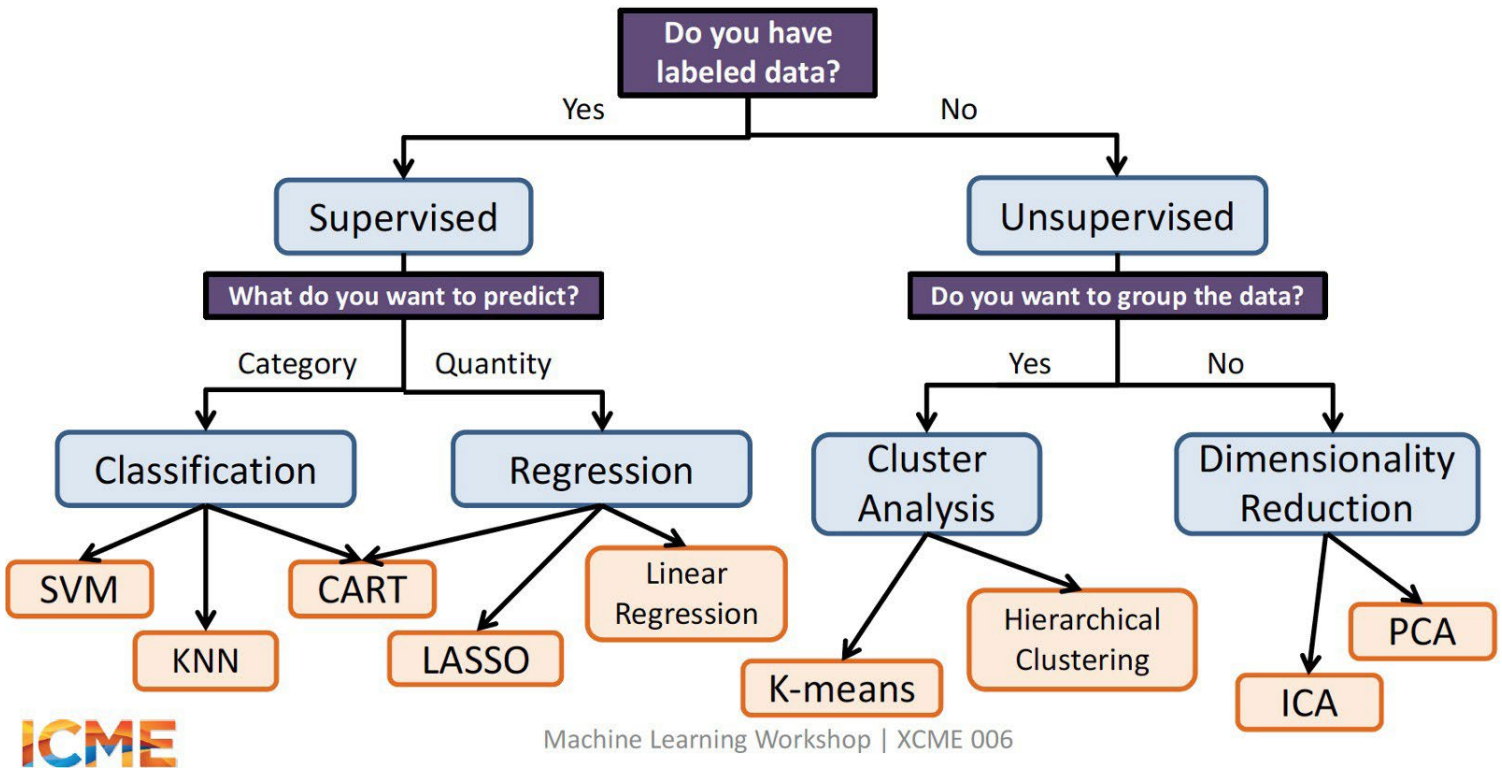
  Predictors → $(X_1, X_2, \ldots, X_N)$



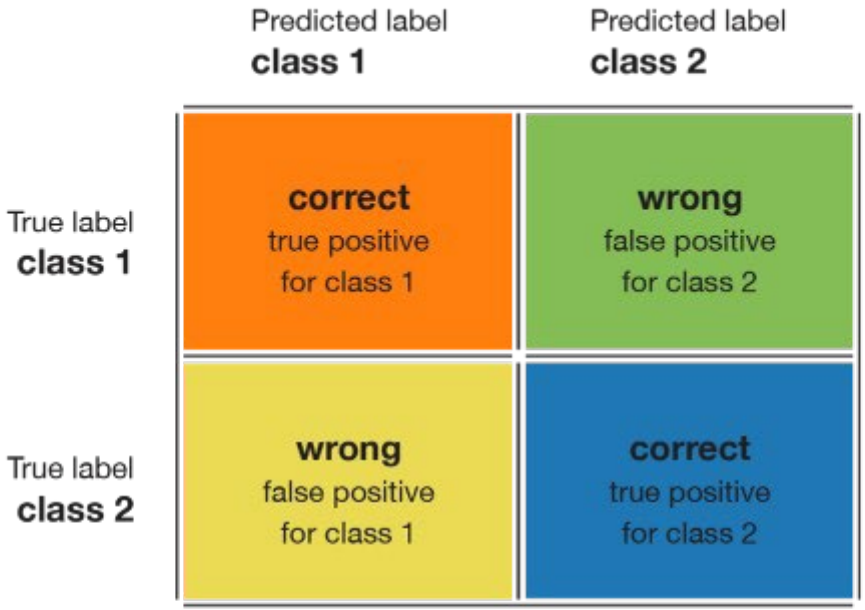

Descripción y visualización
Asociación
Segmentación
Clasificación
Predicción
APRENDIZAJE POR REFUERZO
APRENDIZAJE NO SUPERVISADO
APRENDIZAJE SUPERVISADO


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?
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

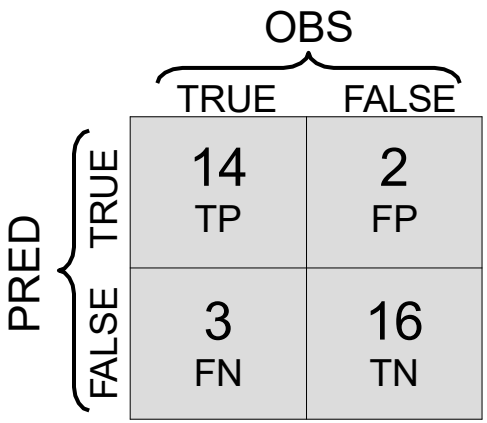
Confusion Matrix



$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Confusion Matrix - Accuracy



El tamaño muestral es:

$$n = 35$$

Accuracy: Tasa global de aciertos (considerando las dos clases)

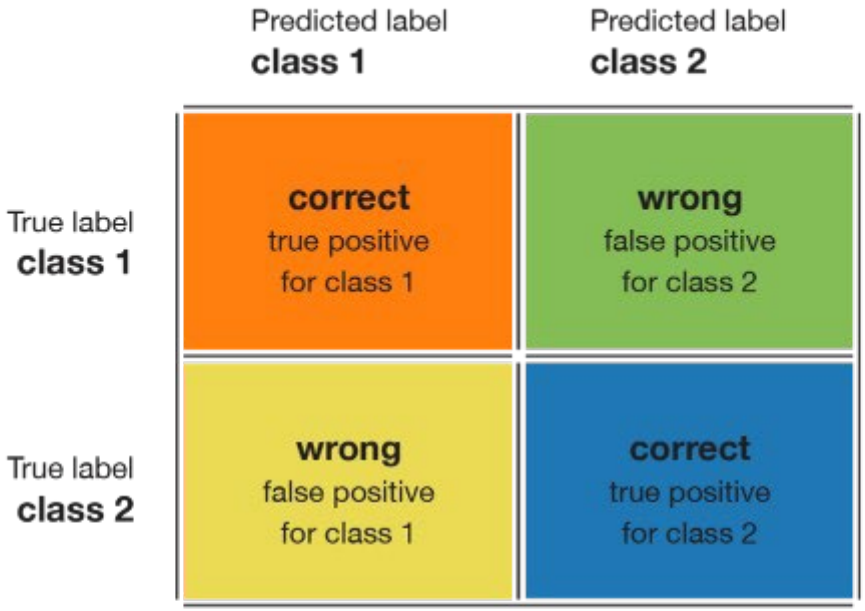
$$ACC = \frac{TP + TN}{n}, \quad 0 \leq ACC \leq 1$$

En este caso, ACC=30/35=0.86. Es decir, el modelo acierta en 86% de las veces que predice un clase y falla el 14% restante.

Es habitual acompañar el ACC con un p-valor que nos indica si el valor obtenido es estadísticamente significativo o no (H_0: el ACC es nulo). Este p-valor se obtiene a partir de M muestras aleatorias, extraídas, sin remplazamiento, de los datos originales con los que se calculó ACC. Se calculan M valores simulados de ACC y se cuenta cuántos de ellos son superiores al valor real, P.

$$p - val = \frac{P + 1}{M + 1}$$

Confusion Matrix



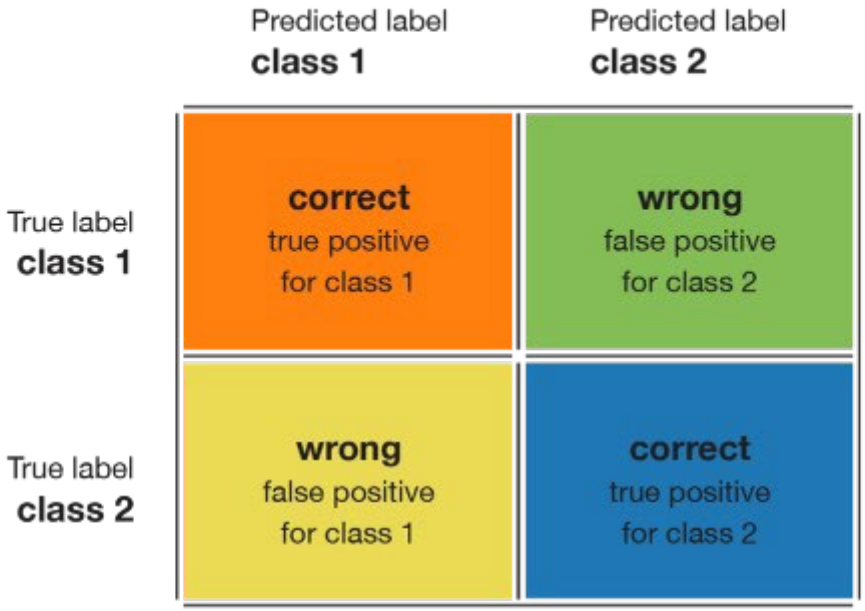
$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

Fatal Genetic Defect

10 out of every 100000 babies

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Confusion Matrix



Fatal Genetic Defect

10 out of every 100000 babies

Model predicting always No Defect

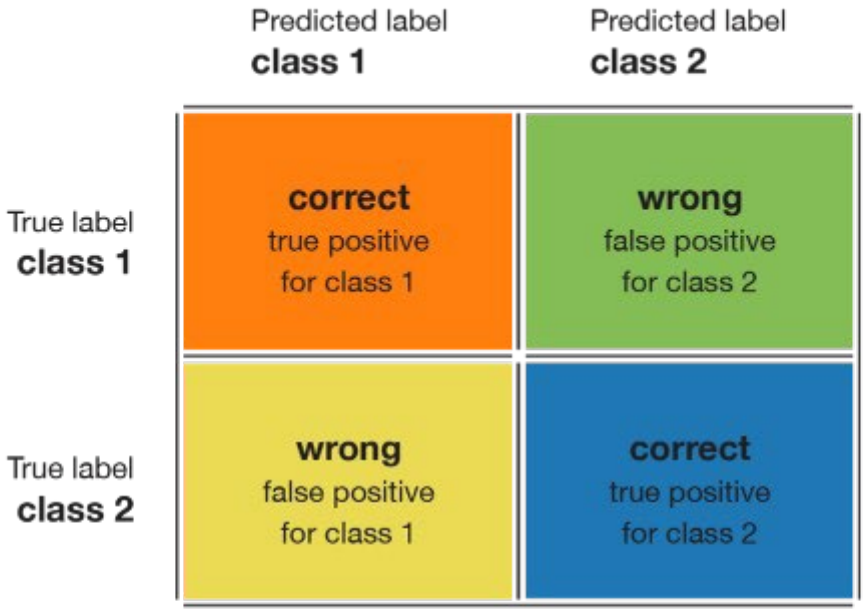
99.99% TP

100% FN

$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Confusion Matrix



$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

Fatal Genetic Defect

10 out of every 100000 babies

Model predicting always No Defect

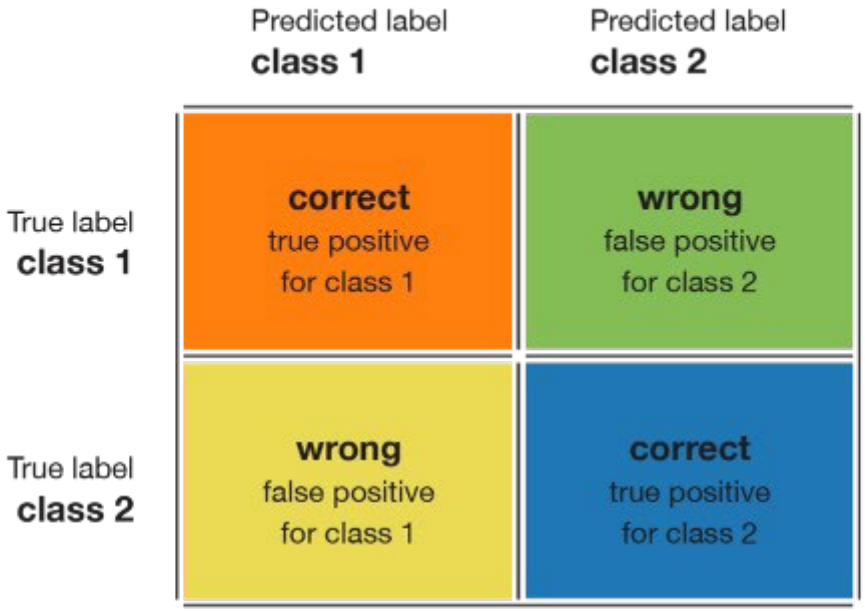
99.99% TP **100% FN**

All the new born babies with the fatal genetic defect are wrongly predicted.

Are there any other validation measure?

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Confusion Matrix



Fatal Genetic Defect

10 out of every 100000 babies

Model predicting always No Defect

99.99% TP **100% FN**

$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

$$\text{class 1 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare} \qquad \text{class 2 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare}$$

$$\text{class 1 recall} = \frac{\blacksquare}{\blacksquare + \blacksquare} \qquad \text{class 2 recall} = \frac{\blacksquare}{\blacksquare + \blacksquare}$$

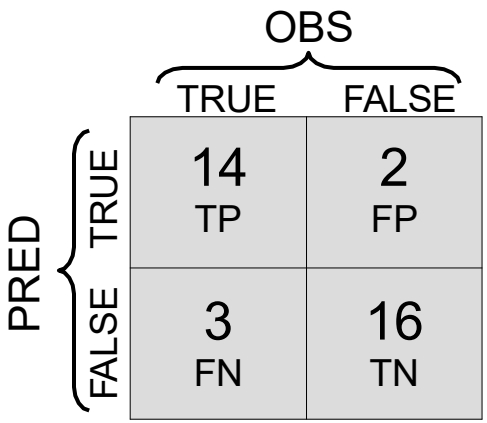
Precision: *define how trustable is the result*

Recall: *expresses how well the model is able to detect that class*

HP *-> priorizes the TP detection*

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Confusion Matrix - Accuracy



El tamaño muestral es:

$$n = 35$$

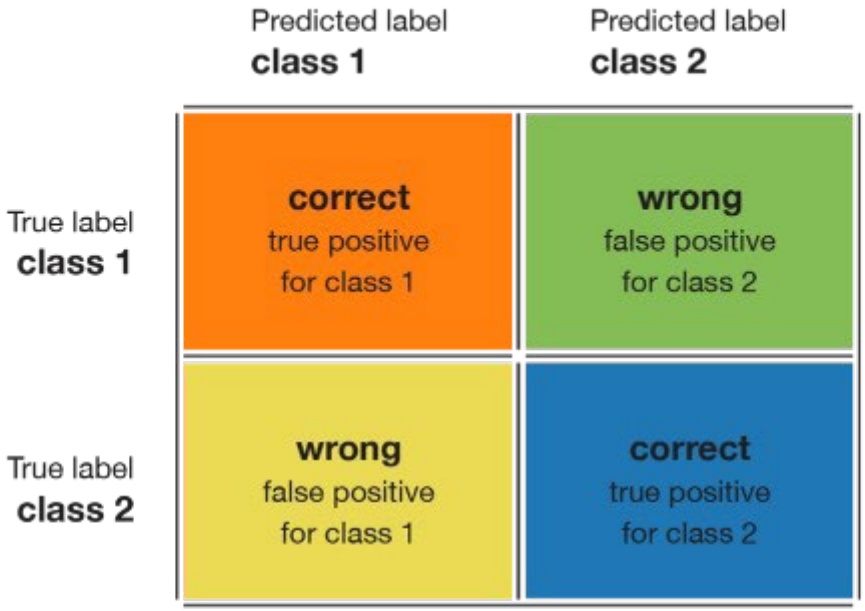
Accuracy: Tasa global de aciertos (considerando las dos clases)

$$ACC = \frac{TP + TN}{n}, \quad 0 \leq ACC \leq 1$$

En este caso, ACC=30/35=0.86. Es decir, el modelo acierta en 86% de las veces que predice un clase y falla el 14% restante.

¿Cuánto valen la precisión y el recall de ambas clases?

Confusion Matrix



Fatal Genetic Defect

10 out of every 100000 babies

Model predicting always No Defect

99.99% TP **100% FN**

$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

$$\text{class 1 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare} \qquad \text{class 2 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare}$$

$$\text{class 1 recall} = \frac{\blacksquare}{\blacksquare + \blacksquare} \qquad \text{class 2 recall} = \frac{\blacksquare}{\blacksquare + \blacksquare}$$

HR/HP: class is perfectly handled by the model

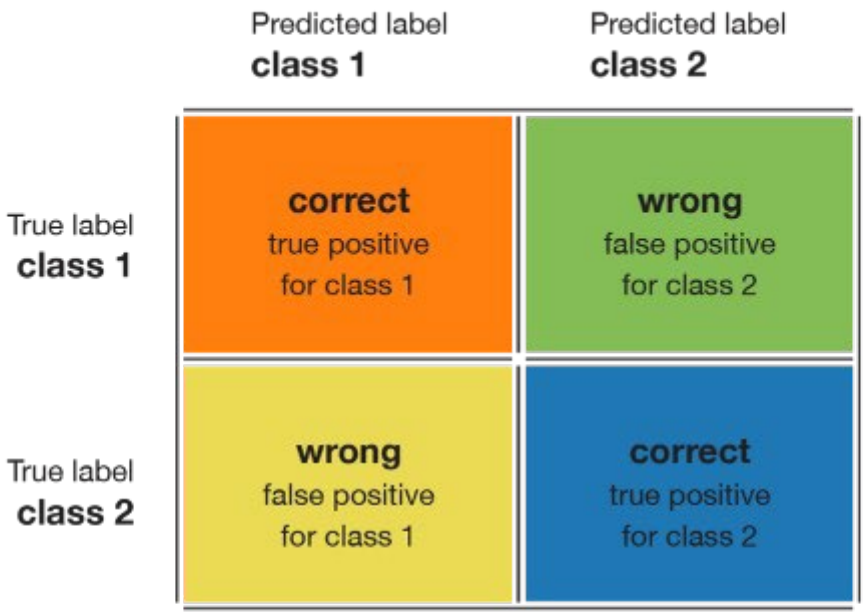
LR/HP: model can't detect the class well but is highly trustable when it does

HR/LP: class is well detected but the model include points of other classes in it

LR/LP: class is poorly handled by the model

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Confusion Matrix



Fatal Genetic Defect

10 out of every 100000 babies

Model predicting always No Defect

99.99% TP **100% FN**

$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

$$\text{class 1 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare} \qquad \text{class 2 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare}$$

$$\text{class 1 recall} = \frac{\blacksquare}{\blacksquare + \blacksquare} \qquad \text{class 2 recall} = \frac{\blacksquare}{\blacksquare + \blacksquare}$$

F1-Score:

$$\frac{2 * precision * recall}{(precision + recall)}$$

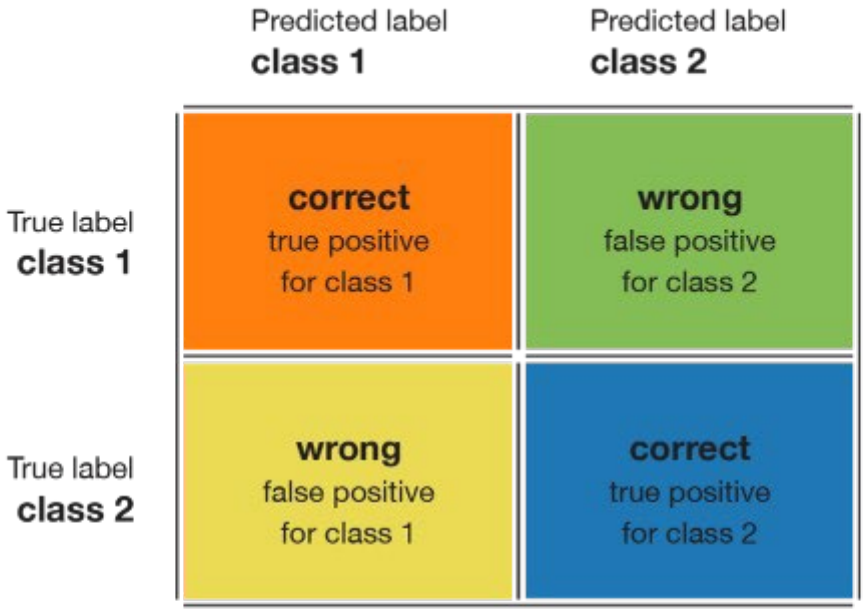
Precision: *define how trustable is the result*

Recall: *expresses how well the model is able to detect that class*

F1: *combines precision and recall of a class in one metric*

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Confusion Matrix



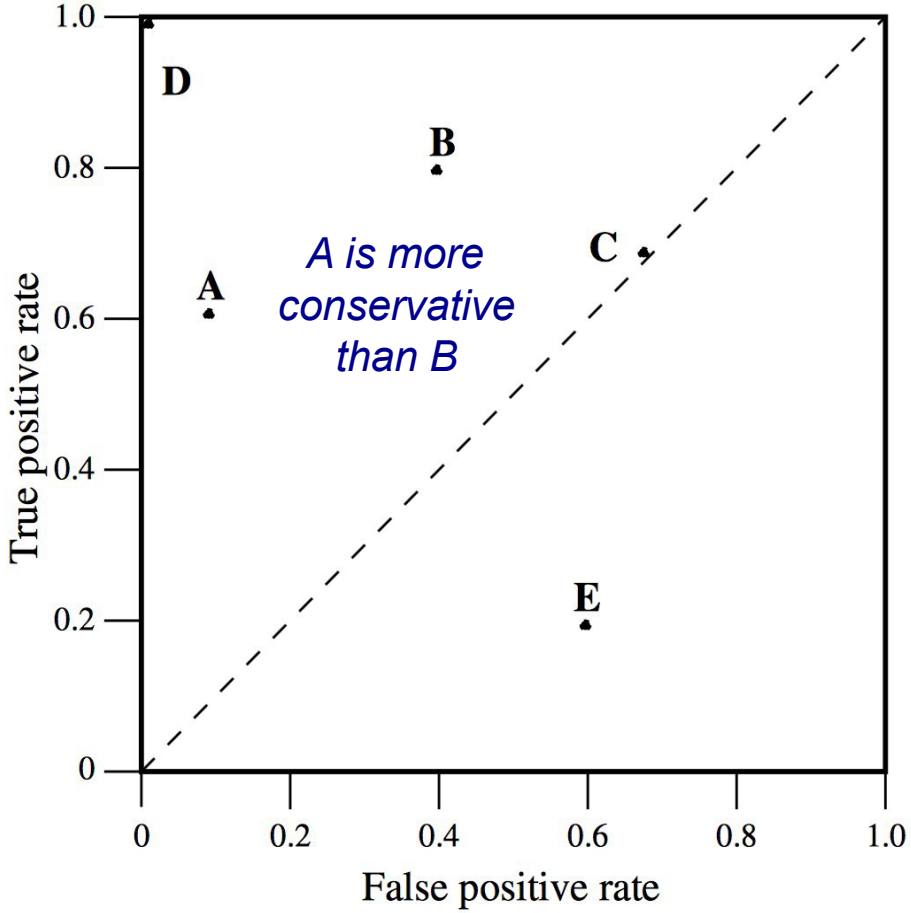
$$\text{accuracy} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare + \blacksquare + \blacksquare}$$

$$\text{class 1 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare} \qquad \text{class 2 precision} = \frac{\blacksquare}{\blacksquare + \blacksquare}$$

$$\text{class 1 recall} = \qquad \text{class 2 recall} =$$

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

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.

Curva ROC (Receiver Operating Characteristic)

OBS (bin)	PRED (prob)
1	0,926
1	0,680
0	0,556
1	0,742
0	0,058
0	0,471
0	0,866
1	0,443
0	0,075
1	0,685

Veamos cómo se construiría la curva ROC para este ejemplo

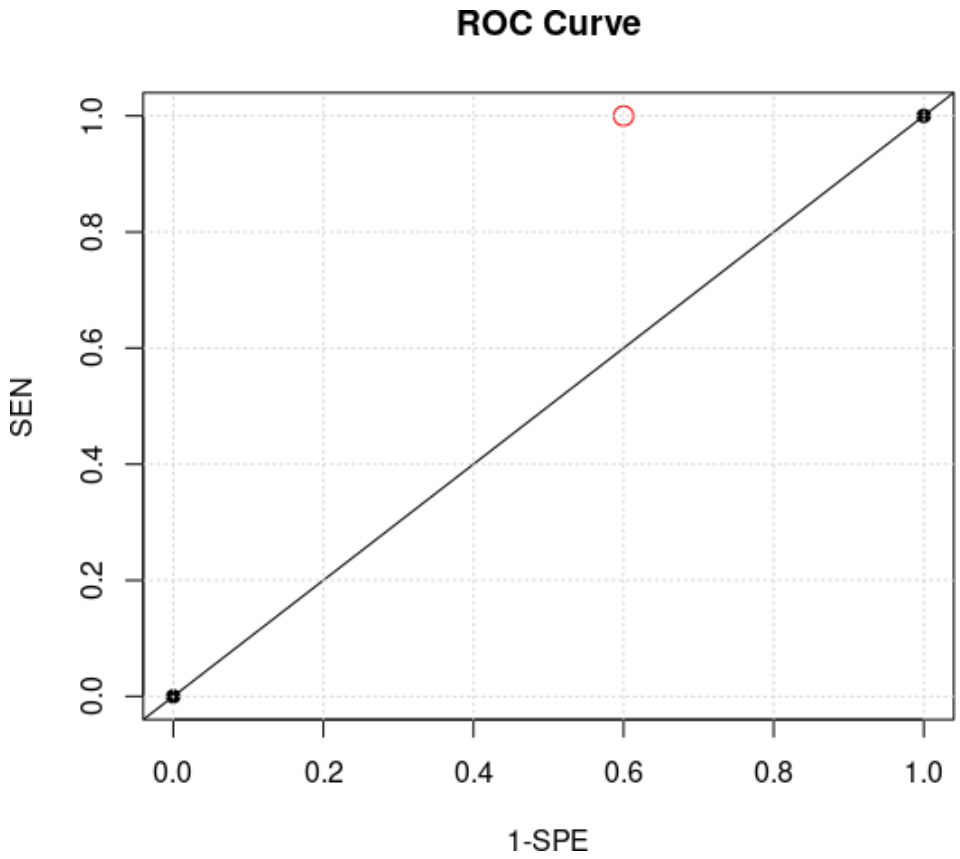
Curva ROC (Receiver Operating Characteristic)

OBS (bin)	PRED (prob)
1	0,926
1	0,680
0	0,556
1	0,742
0	0,058
0	0,471
0	0,866
1	0,443
0	0,075
1	0,685

P_{umb} = **0.25**

OBS (bin)	PRED (bin)
1	1
1	1
0	1
1	1
0	0
0	1
0	1
1	1
0	0
1	1

1-SPE = 0.6
HIR = 1



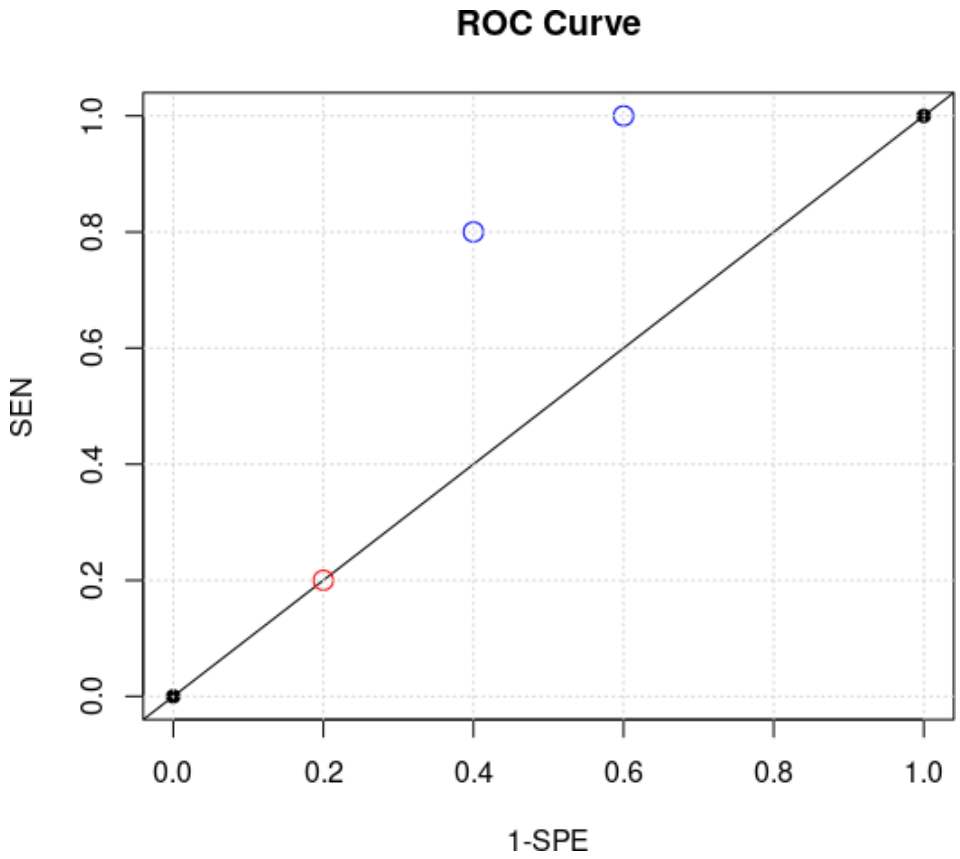
Curva ROC (Receiver Operating Characteristic)

OBS (bin)	PRED (prob)
1	0,926
1	0,680
0	0,556
1	0,742
0	0,058
0	0,471
0	0,866
1	0,443
0	0,075
1	0,685

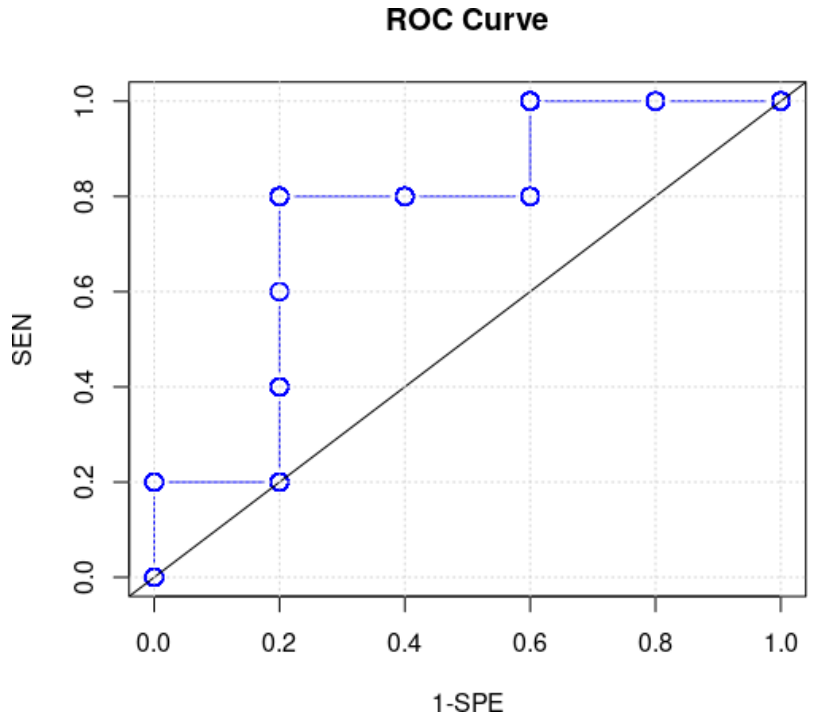
P_{umb} = **0.75**

OBS (bin)	PRED (bin)
1	1
1	0
0	0
1	0
0	0
0	0
0	1
1	0
0	0
1	0

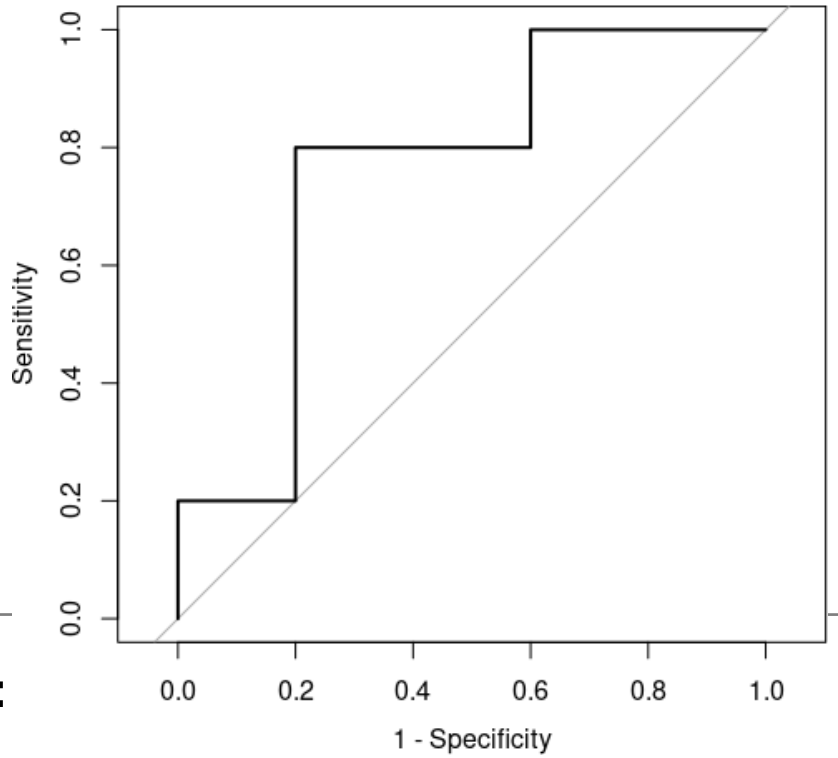
1-SPE = 0.2
HIR = 0.2



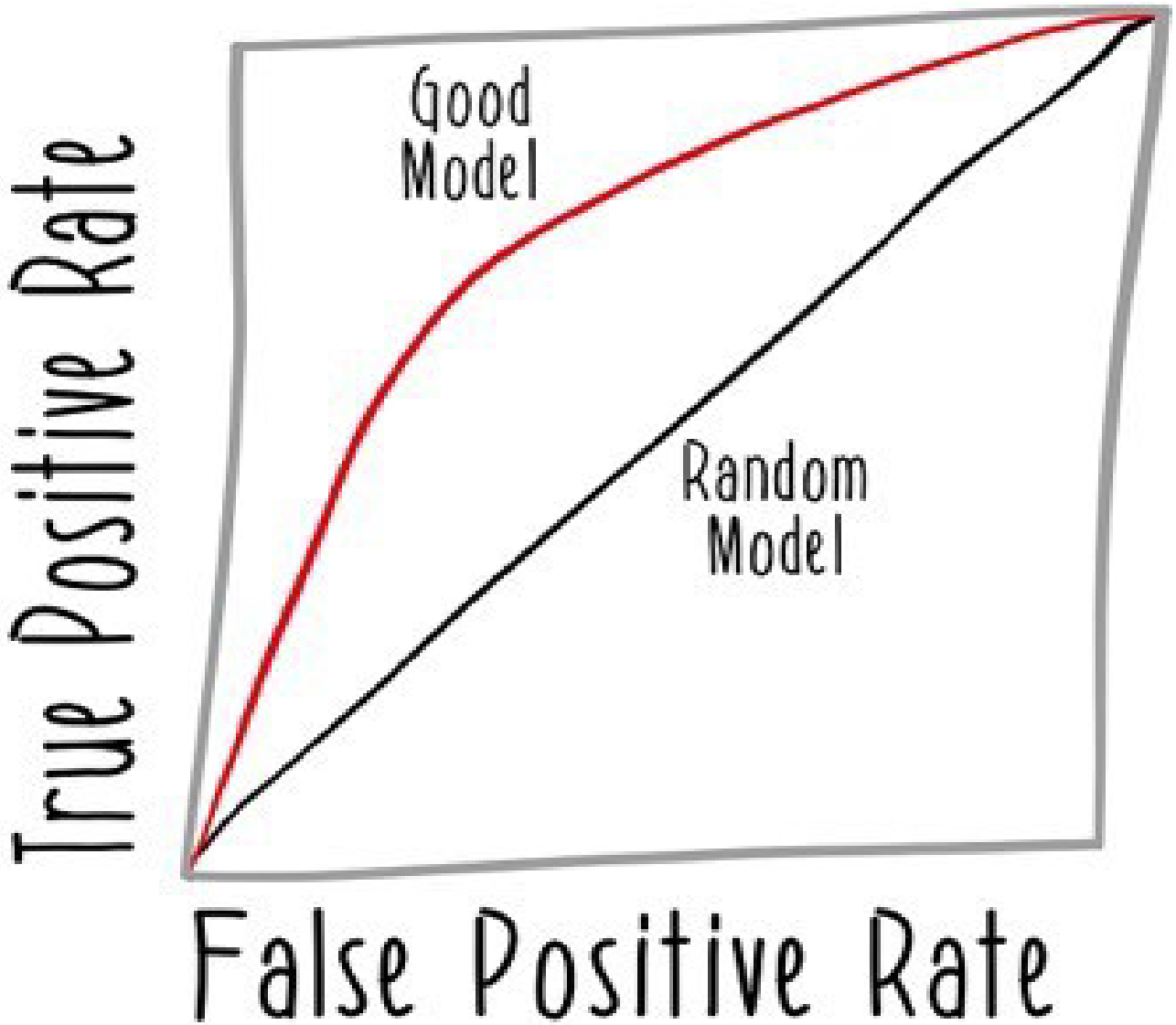
```
p.umb = seq(0, 1, 0.001)
SPE = rep(NA, length(p.umb))
SEN = rep(NA, length(p.umb))
for (p in p.umb) {
  prob.bin = pred.prob >= p;
  SEN[which(p.umb == p)] =
sum(prob.bin[obs.bin])/sum(obs.bin);
  SPE[which(p.umb == p)] = sum(!
prob.bin[!obs.bin])/sum(!obs.bin);
}
plot(1-SPE, SEN, type = "b", main =
"ROC Curve",
     xlim = c(0, 1), ylim = c(0, 1),
     xlab = "1-SPE", ylab = "SEN",
     pch = 1, col = "blue", cex =
1.5)
abline(coef = c(0, 1))
grid()
```




```
library(pROC)
plot(roc(obs.bin, pred.prob),
legacy.axes = T)
```

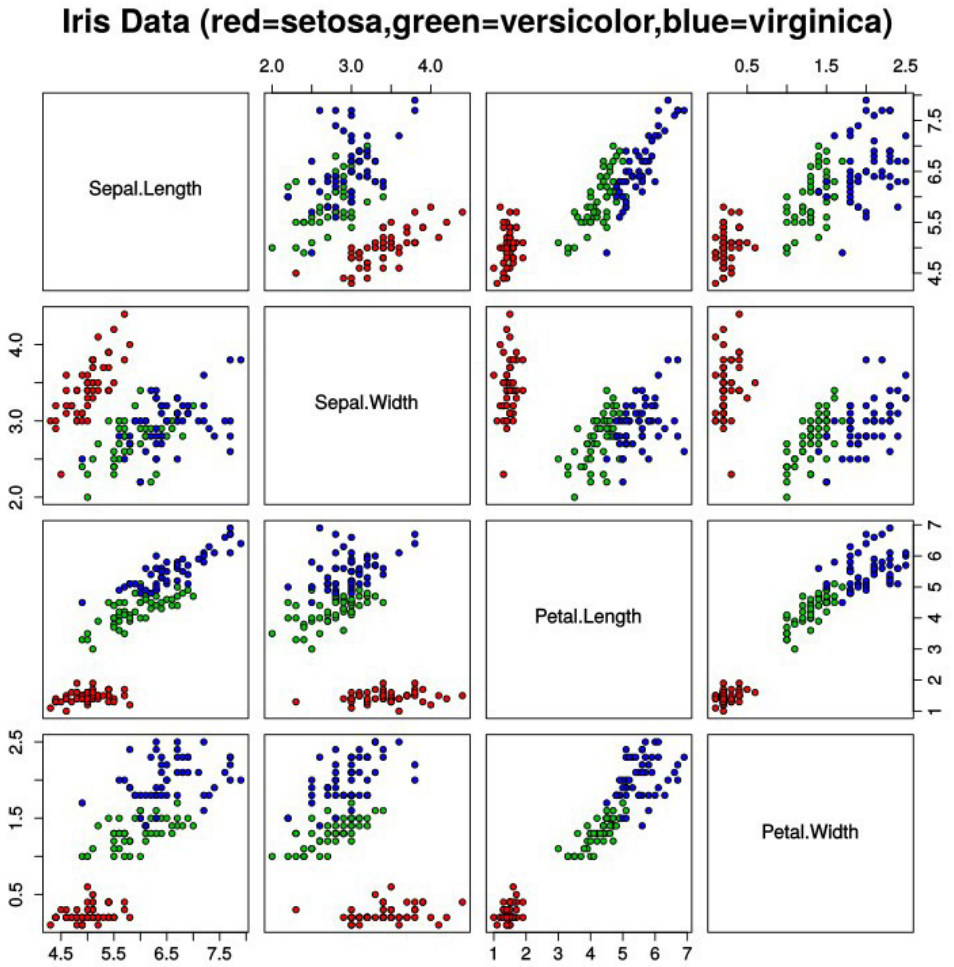


Summarizes the performance of the system over all possible probability thresholds.



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

```
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)
```



Existen varias alternativas:

Fatal Genetic Defect
10 out of every 100000 babies

Under-sampling: Consiste en quedarse únicamente con una parte de las observaciones correspondientes a la clase mayoritaria (seleccionadas aleatoriamente), para que la proporción de las dos sea similar.

Over-sampling: Consiste en replicar parte de las observaciones correspondientes a la clase minoritaria (seleccionadas aleatoriamente), para que la proporción de las dos sea similar.

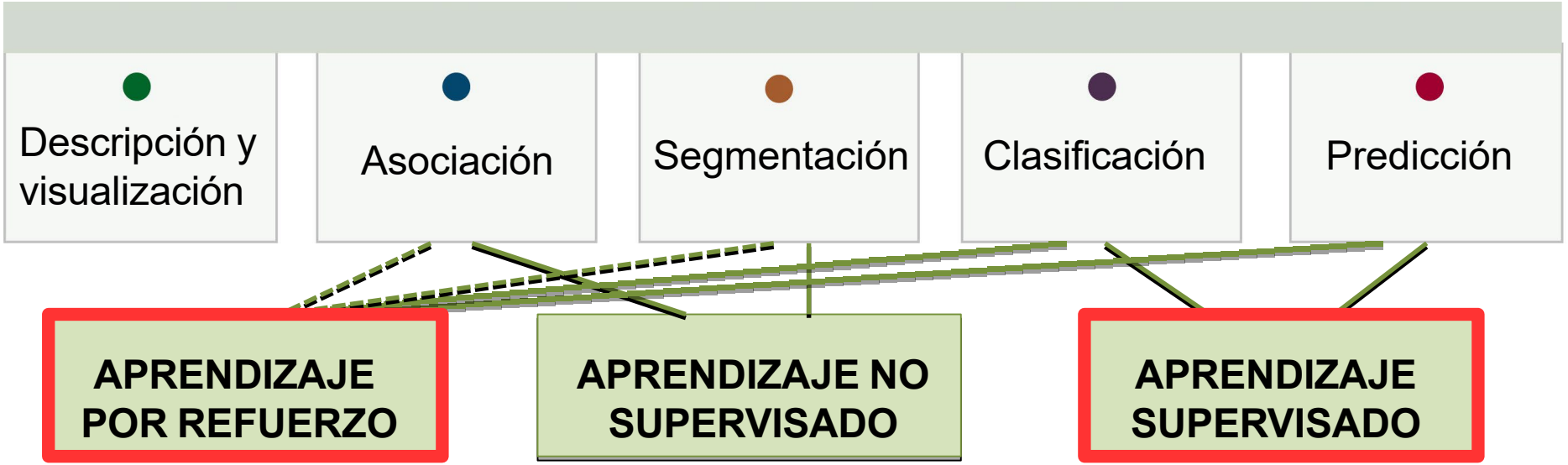
Generación de datos sintéticos: Consiste en generar nuevos datos para la clase minoritaria, a partir de los originales (por ejemplo, mediante interpolación). Los algoritmos más utilizados para ello son ROSE y SMOTE.

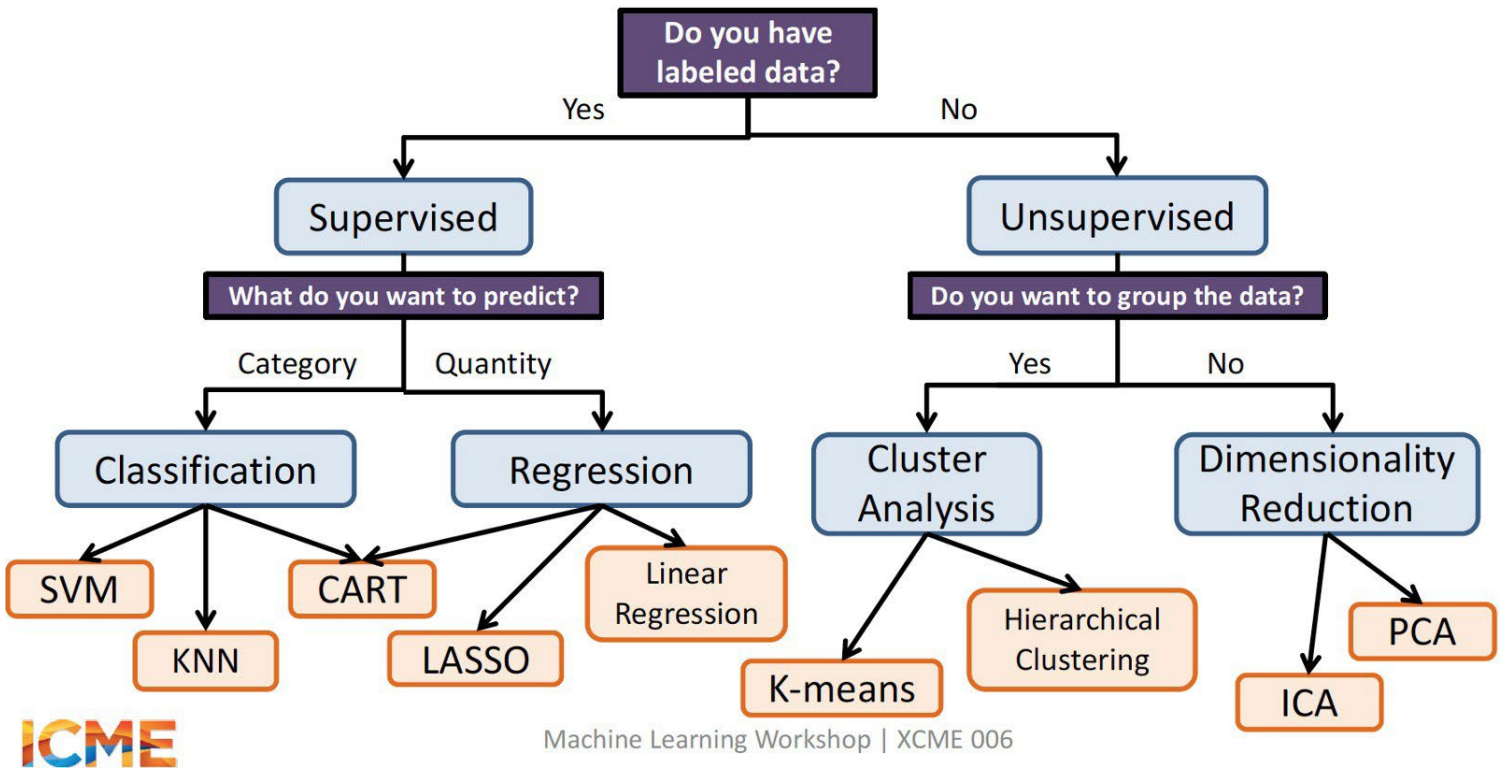
Modificación de la función de coste: En las técnicas que se entrenan para minimizar una función de coste, es posible incorporar pesos que den más importancia a los errores cometidos sobre la clase minoritaria.

Dealing with unbalanced data in machine learning

https://shiring.github.io/machine_learning/2017/04/02/unbalanced
https://www.kdnuggets.com/2023/07/overcoming-imbalanced-data-challenges-realworld-scenarios.html

Source: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28


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

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 (residual sum of squares)**.

$$RSS = \frac{1}{N}\sum_{i=1}(y_i - y^*_i)^2$$

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**.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y^*_i)^2}$$

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

Model performance: Pearons vs Spearman correlation

Pearson correlation: Indica la magnitud de la relación lineal existente entre dos variables, x e y, que deben seguir una **distribución normal**.

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \quad -1 \leq \rho_{xy} \leq 1$$

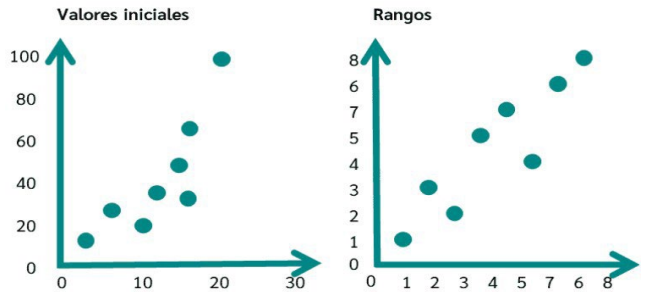
$$\rho_{xy} > 0 \Rightarrow \text{relación lineal directa}$$

$$\rho_{xy} < 0 \Rightarrow \text{relación lineal inversa}$$

Spearman correlation: Es la correlación de Pearson, pero calculada sobre **rangos** en lugar de sobre los datos originales. Se dice por tanto que es **no paramétrica**, puesto que no exige que las variables x e y sigan ninguna distribución particular. Se suele denotar con la letra r.

12	1
15	2
17	3
18	4
20	5
21	6
22	7
26	8

14	1
25	3
20	2
35	5
45	6
30	4
60	7
95	8

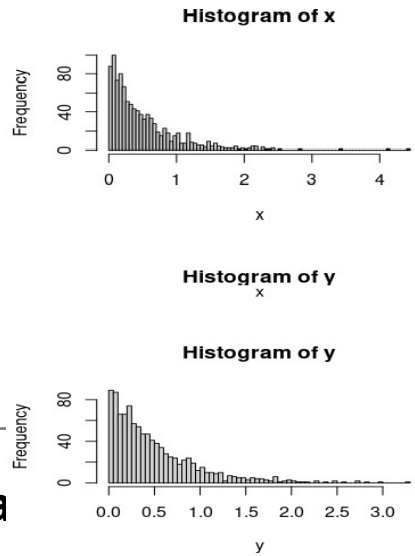



```
x = rgamma(1000, 1, 2)
y = rgamma(1000, 1, 2)

par(mfrow = c(2, 1))  hist(x,
100)
hist(y, 100)

cor(x, y, method = "spearman")
[1] 0.02812556

cor.test(x, y, method = "spearman")
```

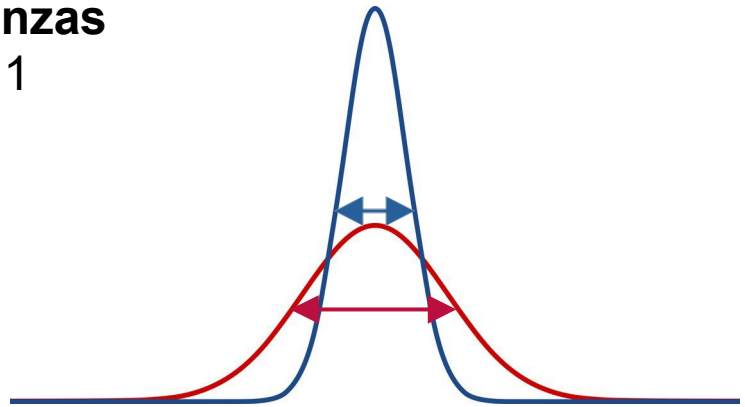


Distributional consistency: evaluates the model capability to reproduce the distribution of the observed data.

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

Ratio de varianzas
Valor perfecto: 1

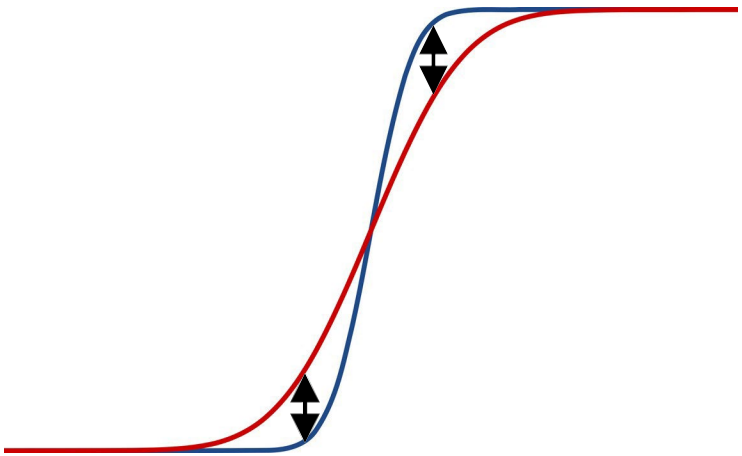
$$RV_{xy} = \frac{\sigma_x^2}{\sigma_y^2}$$



Kolmogorov-Smirnov score
Valor perfecto: 0

$$KS_{stat} = max\left[dist(CDF_x, CDF_y)\right], \quad 0 \leq KS_{stat} \leq 1$$

No paramétrico: Se aplica sobre las distribuciones empíricas
Detecta diferencias en cualquier parte de la distribución: media, varianza, skewness, kurtosis, colas….



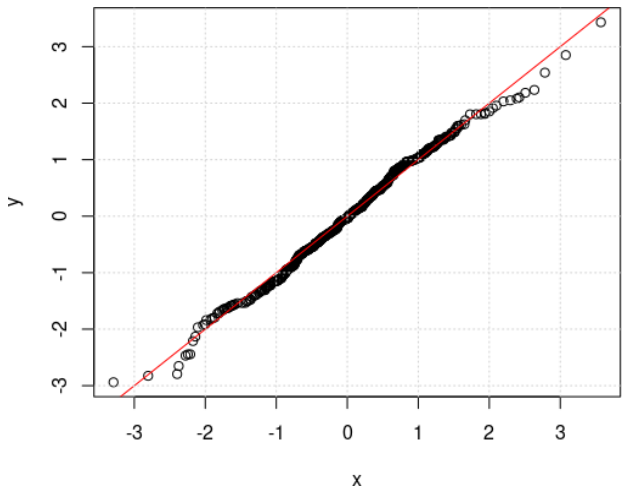
Distributional consistency: evaluates the model capability to reproduce the distribution of the observed data.

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

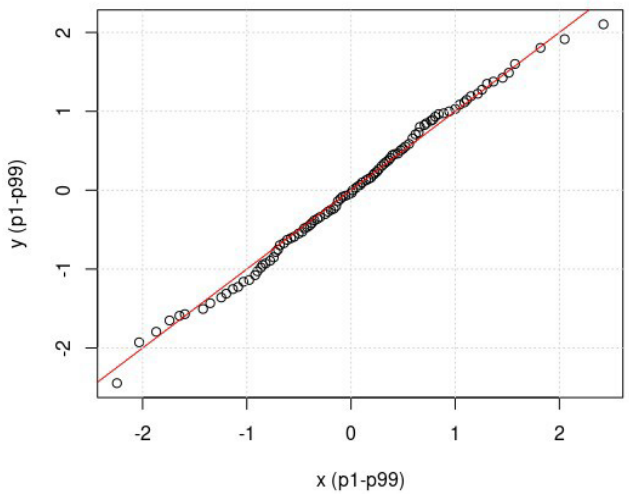
-The **quantile-quantile plot** is a typical tool to evaluate, in a graphical way, the distributional similarity of the order statistics (e.g. **percentiles**).

```
x = rnorm(1000)
y = rnorm(500)
```

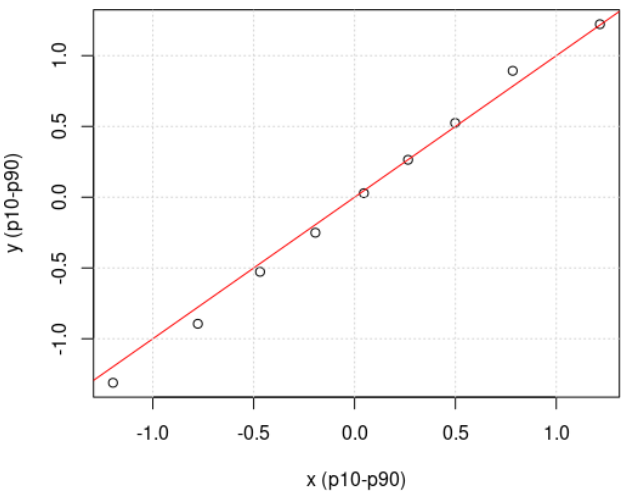
```
qqplot(x, y)
abline(coef = c(0, 1), col = "red")
grid()
```




```
plot(quantile(x,
   probs = seq(0.01, 0.99, 0.01)),
quantile(y,
   probs = seq(0.01, 0.99, 0.01)),
xlab = "x (p1-p99)", ylab = "y (p1-p99)")
abline(coef = c(0, 1), col = "red")
grid()
```



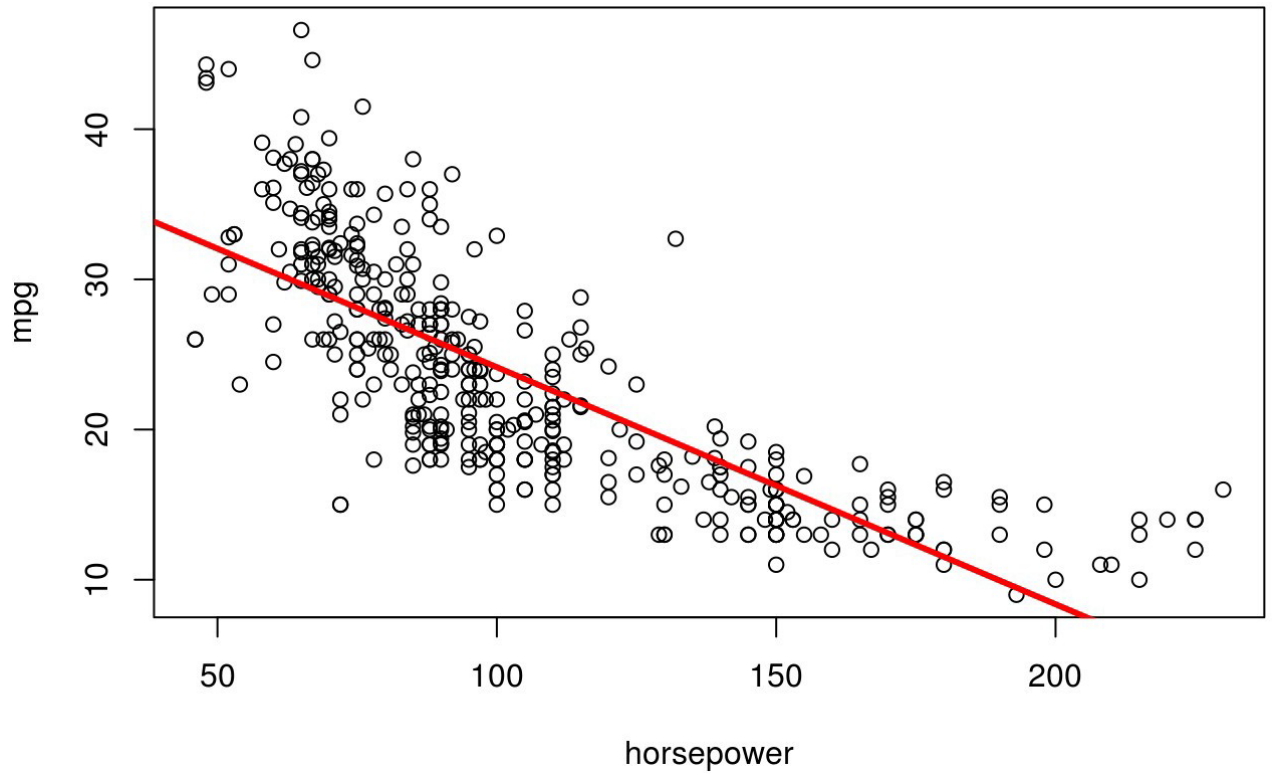

```
plot(quantile(x,
   probs = seq(0.1, 0.9, 0.1)),
quantile(y,
   probs = seq(0.1, 0.9, 0.1)),
xlab = "x (p10-p90)", ylab = "y (p10-p90)")
abline(coef = c(0, 1), col = "red")
grid()
```



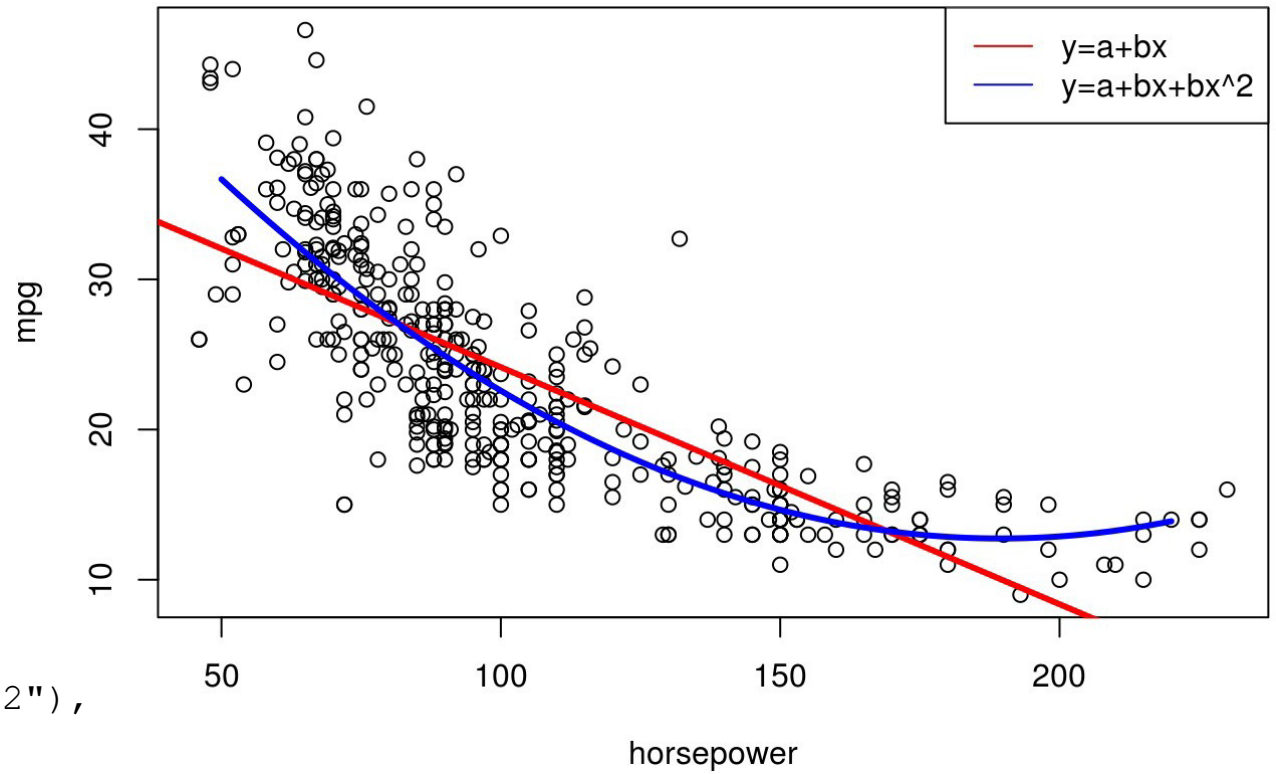
Different diagnostics for different fields.

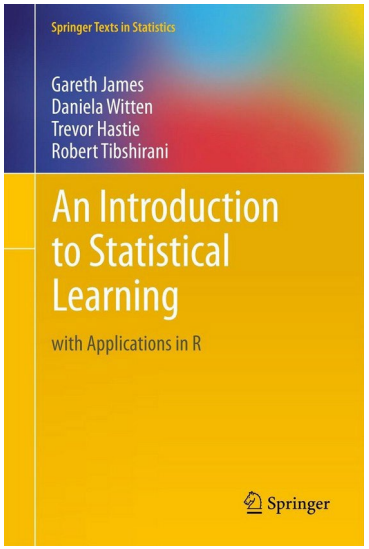
	Cross-Validation	**Continuous: Model Evaluation**

```
# install.packages('ISLR')
library(ISLR)
attach(Auto)
summary(Auto)
n <- length(mpg)
plot(horsepower, mpg)
lm1 <- lm(mpg~horsepower)
abline(lm1, col="red", lwd=3)
summary(lm1)
```



```
# install.packages('ISLR')
library(ISLR)
attach(Auto)
summary(Auto)
n <- length(mpg)
plot(horsepower, mpg)
lm1 <- lm(mpg~horsepower)
abline(lm1, col="red", lwd=3)
summary(lm1)
plot(horsepower, mpg)
abline(lm1, col="red", lwd=3)
lm2 <- lm(mpg~poly(horsepower,2))
xs <- seq(50,220,length=100)
ys <- predict(lm2,
    data.frame(horsepower=xs))
lines(xs,ys, type="l",
    lwd=3, col="blue")
legend("topright",
    legend=c("y=a+bx","y=a+bx+bx^2"),
    lty=1 , col=c("red", "blue"))
summary(lm2)
```





An Introduction to Statistical Learning: With Applications in R

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

Springer (2013)

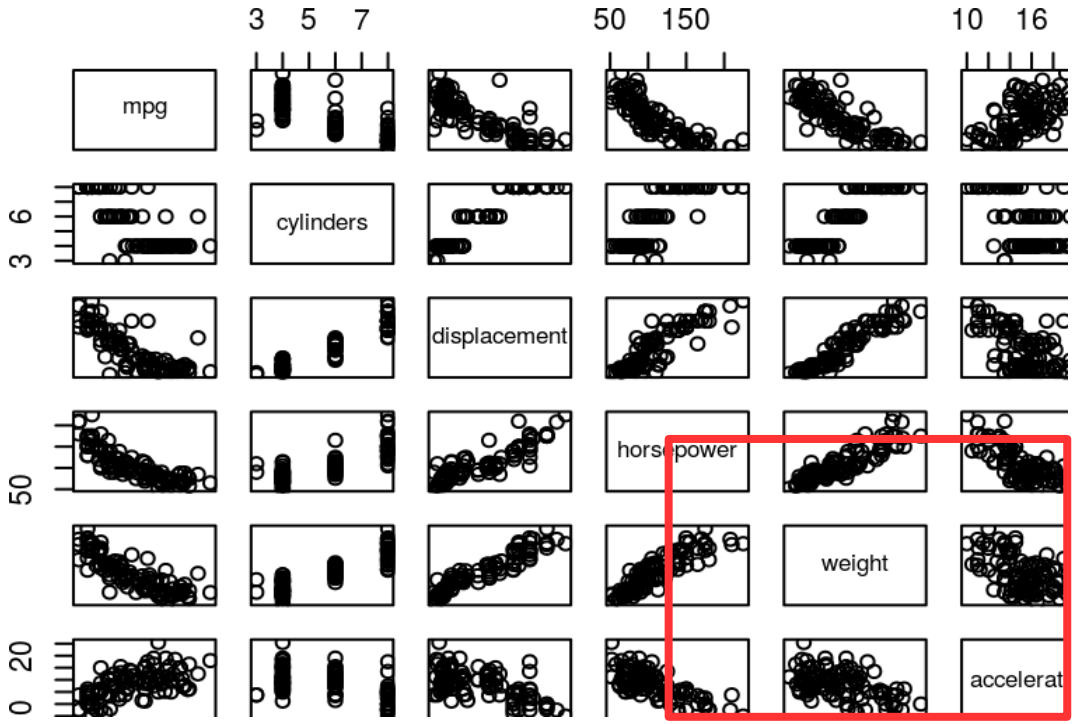
http://www-bcf.usc.edu/~gareth/ISL

```
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")
```