

# Machine learning y deep learning

Rolando Gonzales Martinez, PhD

Fellow postdoctoral Marie  
Skłodowska-Curie

Universidad de Groningen  
(Países Bajos)

Investigador (researcher)

Iniciativa de Pobreza y Desarrollo  
Humano de la Universidad de  
Oxford (UK)

# Contenido del curso

## **(2) Introducción a Machine Learning**

- Definición y tipos de aprendizaje: supervisado, no supervisado y por refuerzo.
- Herramientas y lenguajes de programación más utilizados.
- Laboratorio: prácticas en Python y R

# Machine learning

Dado un espacio de entrada  $X$  (espacio de características) y un espacio de salida  $Y$ , machine learning es un problema de optimización en el que el objetivo es encontrar una función  $f: X \rightarrow Y$  que predice la salida  $y \in Y$  dada una entrada  $x \in X$ , siendo  $f^*$  óptima en términos de **generalización y regularización**:

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad x_i \in \mathcal{X} \quad y_i \in \mathcal{Y}$$

$$f^* = \arg \min_{f \in \mathcal{F}} \mathbb{E}_{(x,y) \sim P} [L(f(x), y)]$$

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n L(f(x_i), y_i)$$

$$f^* = \arg \min_{f \in \mathcal{F}} \left[ \frac{1}{n} \sum_{i=1}^n L(f(x_i), y_i) + \lambda R(f) \right]$$

# Principales tipos de aprendizaje

- **Supervisado:** Modelo ML entrenado con datos etiquetados.

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n L(f(x_i), y_i)$$

- **Semi-supervisado:** Modelo ML entrenado con una combinación de datos etiquetados y no etiquetados.

$$f^* = \arg \min_{f \in \mathcal{F}} \left[ \frac{1}{|L|} \sum_{(x_i, y_i) \in \mathcal{D}_L} L(f(x_i), y_i) + \lambda R(f, \mathcal{D}_U) \right]$$

- **No supervisado:** Modelo ML entrenado con datos no etiquetados.

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n L(f(x_i))$$

# Otros tipos de aprendizaje

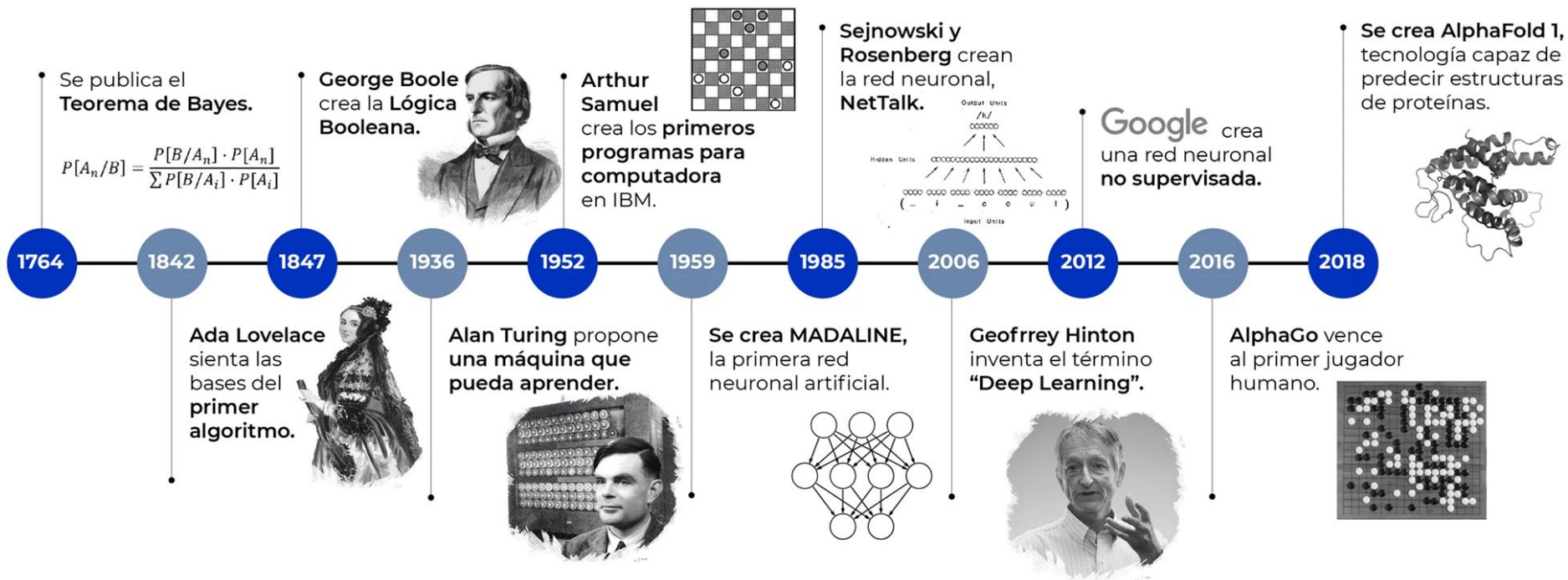
- **Reforzamiento:** Modelo ML se optimiza en base a un agente  $a$  y un conjunto de estados  $s$ ,  $\pi(a|s)$
- **Transferencia:** Modelo ML entrenado se re-utiliza y adapta a una nueva tarea
- **Meta-aprendizaje:** El modelo ML se adapta a nuevas tareas  $T$  en base a pocos ejemplos de entrenamiento

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^T \gamma^t r(s_t, a_t) \right]$$

$$f_T^* = \arg \min_{f \in \mathcal{F}_T} \frac{1}{m} \sum_{j=1}^m L(f(x_j), y_j)$$

$$f^* = \arg \min_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{T}_i \sim p(\mathcal{T})} \left[ \min_{\theta_i} L_{\mathcal{T}_i}(f(\theta_i)) \right]$$

# LÍNEA DEL TIEMPO DE MACHINE LEARNING



# Algoritmos y modelos

**Modelo:** Representación simplificada de un proceso, sistema o fenómeno del mundo real. Es una construcción matemática o estadística.

$$\text{salario} = f(\text{educación}, \text{edad}, \text{experiencia})$$

$$\text{salario} = f(\text{educación}, \text{edad}, \text{experiencia}; \Theta)$$

$$\Theta = [\theta_{educ}, \theta_{edad}, \theta_{expe}]$$

$$\text{salario} = f(\text{educación}, \text{edad}, \text{experiencia}; \theta_{educ}, \theta_{edad}, \theta_{expe})$$

Asumiendo que  $f(\cdot)$  es lineal:

$$\text{salario} = \theta_0 + \theta_{educ} \text{educación} + \theta_{edad} \text{edad} + \theta_{expe} \text{experiencia}$$

Añadiendo un término i.i.d.  $\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$ :

$$\text{salario} = \theta_0 + \theta_{educ} \text{educación} + \theta_{edad} \text{edad} + \theta_{expe} \text{experiencia} + \varepsilon$$

# Algoritmos y modelos

**Algoritmo:** Serie de pasos o instrucciones lógicas que se utilizan para realizar una tarea específica.

Output: **salario**

Input: **edad**, **experiencia**, **educación**

**if** **edad** > 6

**for** (**edad**) **t** = 1:n

**educación**(**t**) =  $f(\text{estudios})$

**while** (**educación** & **edad**) > **aceptable**

**for** (**años**) **e** = 1:m

**experiencia** = **e**

**end**

**salario**(**t**) =  $f(\text{educación}, \text{edad}, \text{experiencia}; \Theta)$

**end**

**end**

**end**



# Algoritmos y modelos

En el contexto de machine learning:

- **Modelo:** representación matemática/estadística que se ajusta a los datos y que se utiliza para hacer predicciones o tomar decisiones.
- **Algoritmo:** instrucciones que guían cómo se ajustan los parámetros de un modelo ML durante el proceso de entrenamiento.
- En el proceso de ML, primero se selecciona un algoritmo apropiado para el tipo de problema que se está abordando (supervisado, no supervisado, etc.). Luego, este algoritmo se utiliza para entrenar un modelo específico, ajustando sus parámetros para que el modelo pueda generalizar la tarea a datos nuevos sin sobreajuste.

# Algoritmos y modelos

$$\text{salario}_i = \theta_0 + \theta_{educ} \text{educación}_i + \theta_{edad} \text{edad}_i + \theta_{expe} \text{experiencia}_i + \varepsilon_i$$

## Algoritmo de estimación:

Input: datos de salarios, educación, edad, experiencia

Output: estimadores de  $\theta_{educ}, \theta_{edad}, \theta_{expe}$

1. Representación matricial:

$$\Theta = [\theta_{educ}, \theta_{edad}, \theta_{expe}]$$

$$\mathbf{X} = [\text{educación}_i, \text{edad}_i, \text{experiencia}_i]$$

$$\mathbf{y} = [\text{salario}_1, \text{salario}_2, \dots, \text{salario}_n]$$

$$\mathbf{y} = \mathbf{X} \Theta + \varepsilon$$

$$\mathbf{y} - \mathbf{X} \Theta = \varepsilon$$

2. Operaciones matriciales: cálculo de matrices transpuestas e inversas,  $\mathbf{X}'$ ,  $(\mathbf{X}'\mathbf{X})^{-1}$

3. Estimadores MCO:

$$\underset{\Theta \in \mathbb{R}}{\operatorname{argmin}} \varepsilon: \hat{\Theta} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}$$

$$i=1,2,\dots,n$$

4. Estimadores MCO:  $\hat{\theta}_{educ}, \hat{\theta}_{edad}, \hat{\theta}_{expe}$

# Particiones muestrales en machine learning

- **Partición en muestra train y test:** Método más básico y común. Se divide los datos en dos partes: un conjunto de entrenamiento para entrenar el modelo (estimar los parámetros del modelo) y un conjunto de prueba para evaluar el rendimiento del modelo en datos no vistos.
- **Hold-Out con Validación:** similar a la partición en entrenamiento y prueba, pero incluye un tercer conjunto de validación:
  - **Entrenamiento (Train):** Para ajustar los parámetros del modelo.
  - **Validación (Validation):** Para ajustar hiperparámetros y evitar sobreajuste.
  - **Prueba (Test):** Para la evaluación final del modelo.

# Particiones muestrales en machine learning

Partición de datos **en 4 grupos**:

Entrenamiento (train):  $i = 1, \dots, m$

persona	salario	educación	edad	experiencia
$i = 1$	2640	24	40	10
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i = 8$	6400	30	57	12
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i = m$	3555	29	32	9

**y train**      **X entrenamiento (train)**

El conjunto de entrenamiento se utiliza para entrenar el modelo (estimar los parámetros del modelo)

Evaluación (test):  $i = m, m+1, \dots, n$

persona	salario	educación	edad	experiencia
$i = m$	5656	27	38	5
$i = m+1$	3322	24	37	10
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i = n$	4300	32	40	13

**y test**      **X test**

El conjunto de prueba se emplea para evaluar el rendimiento del modelo en datos no vistos

# Particiones muestrales en machine learning

Partición de datos:

Entrenamiento (train):  $i = 1, \dots, m$

$$\text{salario}_i = \theta_0 + \theta_{educ} \text{educación}_i + \theta_{edad} \text{edad}_i + \theta_{expe} \text{experiencia}_i + \varepsilon_i$$

$$\boldsymbol{\theta} = [\theta_{educ}, \theta_{edad}, \theta_{expe}],$$

$$\mathbf{X}_{\text{train}} = [\text{educación}_i, \text{edad}_i, \text{experiencia}_i]$$

$$\mathbf{y}_{\text{train}} = [\text{salario}_1, \text{salario}_2, \dots, \text{salario}_m]$$

$$\mathbf{y}_{\text{train}} = \mathbf{X} \boldsymbol{\theta} + \boldsymbol{\varepsilon}$$

$$\underset{\substack{\boldsymbol{\theta} \in \mathbb{R} \\ i=1,2,\dots,m}}{\text{argmin } \boldsymbol{\varepsilon}}: \hat{\boldsymbol{\theta}} = (\mathbf{X}_{\text{train}}' \mathbf{X}_{\text{train}})^{-1} \mathbf{X}_{\text{train}}' \mathbf{y}_{\text{train}}$$

Evaluación (test):  $i = m, m+1, \dots, n$

$$\mathbf{y}_{\text{pronostico}} = \mathbf{X}_{\text{test}} \hat{\boldsymbol{\theta}}$$

Error cuadrático medio (MSE):

$$MSE = \frac{1}{m} \sum_{i=m}^n (\mathbf{y}_{\text{test},i} - \mathbf{y}_{\text{pronostico},i})^2$$

El conjunto de entrenamiento se utiliza para entrenar el modelo (estimar los parámetros del modelo)

El conjunto de prueba se emplea para evaluar el rendimiento del modelo en datos no vistos

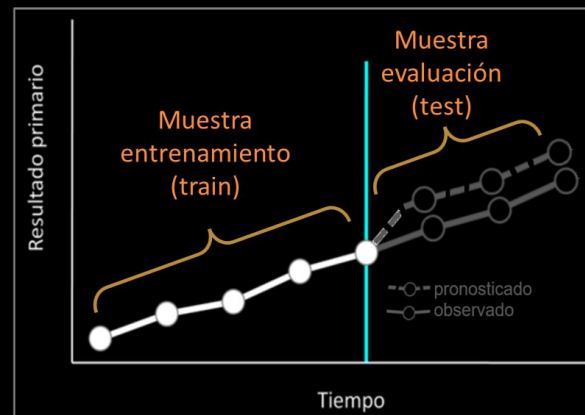
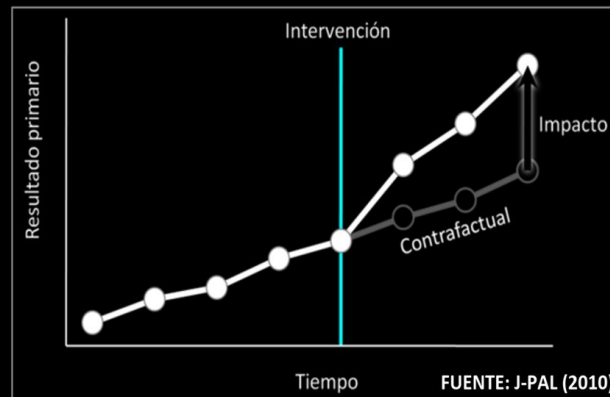
# Paralelismos en Econometría/Estadística y ML

Econometría y Estadística	Machine learning
Estimación	Aprendizaje, entrenamiento
Parámetros	Pesos
Variables explicativas (X)	Features/inputs
Variable dependiente (y)	Target/output
Estimación en muestra completa* (muestras limitadas)	Partición de datos (train/test) (big data)
Propósito: inferencia, predicción	Propósito: predicción
Maximización del ajuste-en-muestra	Minimización del error de predicción

(\*) En general, pero no siempre, e.g. rolling estimation

# Paralelismos en Econometría/Estadística y ML

- **Evaluación de impacto:** se compara la situación con un escenario contrafactual después de la intervención. El impacto es la diferencia entre lo observado y el contrafactual después de la intervención
- **Machine learning:** se particiona los datos en entrenamiento y evaluación, y la muestra de entrenamiento se utiliza para realizar un pronóstico que se compara con la muestra test. Se busca que los pronósticos sean los más cercanos a los datos observados en la muestra test



# Ejemplos de aplicaciones

- Bosques aleatorios se emplean para identificar qué programas sociales promueven un mejor desempeño financiero y una mejor situación social.
- Las predicciones de los árboles de decisión se combinan en una predicción final de los rendimientos financieros.
- El resultado combinado es el “bosque” aleatorio.

INTERNATIONAL JOURNAL OF SUSTAINABLE DEVELOPMENT & WORLD ECOLOGY  
2020, VOL. 27, NO. 5, 389–395  
<https://doi.org/10.1080/13504509.2019.1706059>



OPEN ACCESS Check for updates

## Which social program supports sustainable grass-root finance? Machine-learning evidence

R. Gonzales Martinez

Handelshøyskolen, University of Agder, Kristiansand, Norway

### ABSTRACT

Resources for development are used efficiently when social programs help to promote at the same time the sustainability of grass-root financial associations at the bottom of the pyramid. This study applies machine-learning to a worldwide database of grass-root associations in order to identify which social programs are good predictors of financial returns in the groups. The results indicate that education, income-generating activities and health programs are the most frequent programs provided by development agencies. Business training is not the most frequent intervention applied to grass-root associations, but it is in fact the most important social program to encourage financial sustainability, particularly after a development agency stops working with a group and leaves the community. Theoretical and practical implications of the findings are discussed.

### ARTICLE HISTORY

Received 7 November 2019  
Accepted 13 December 2019

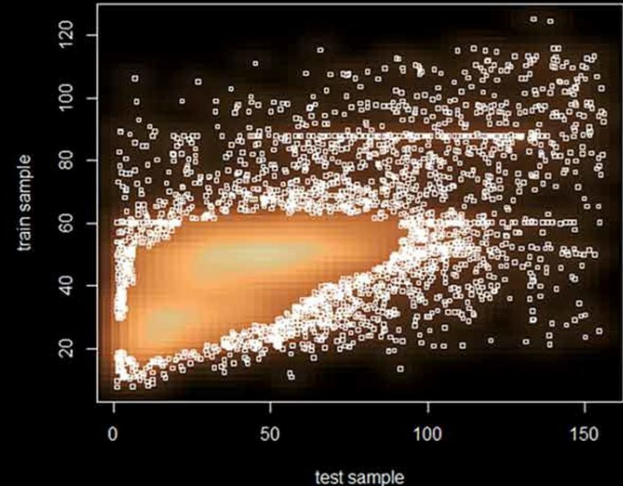
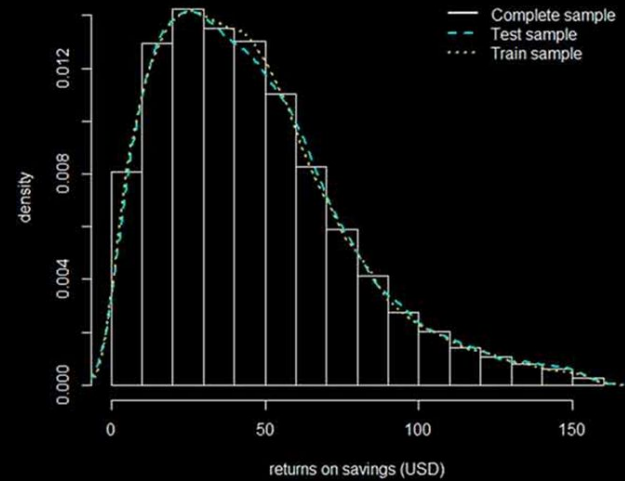
### KEYWORDS

Grass-root finance; health and business interventions; sustainable development; machine-learning

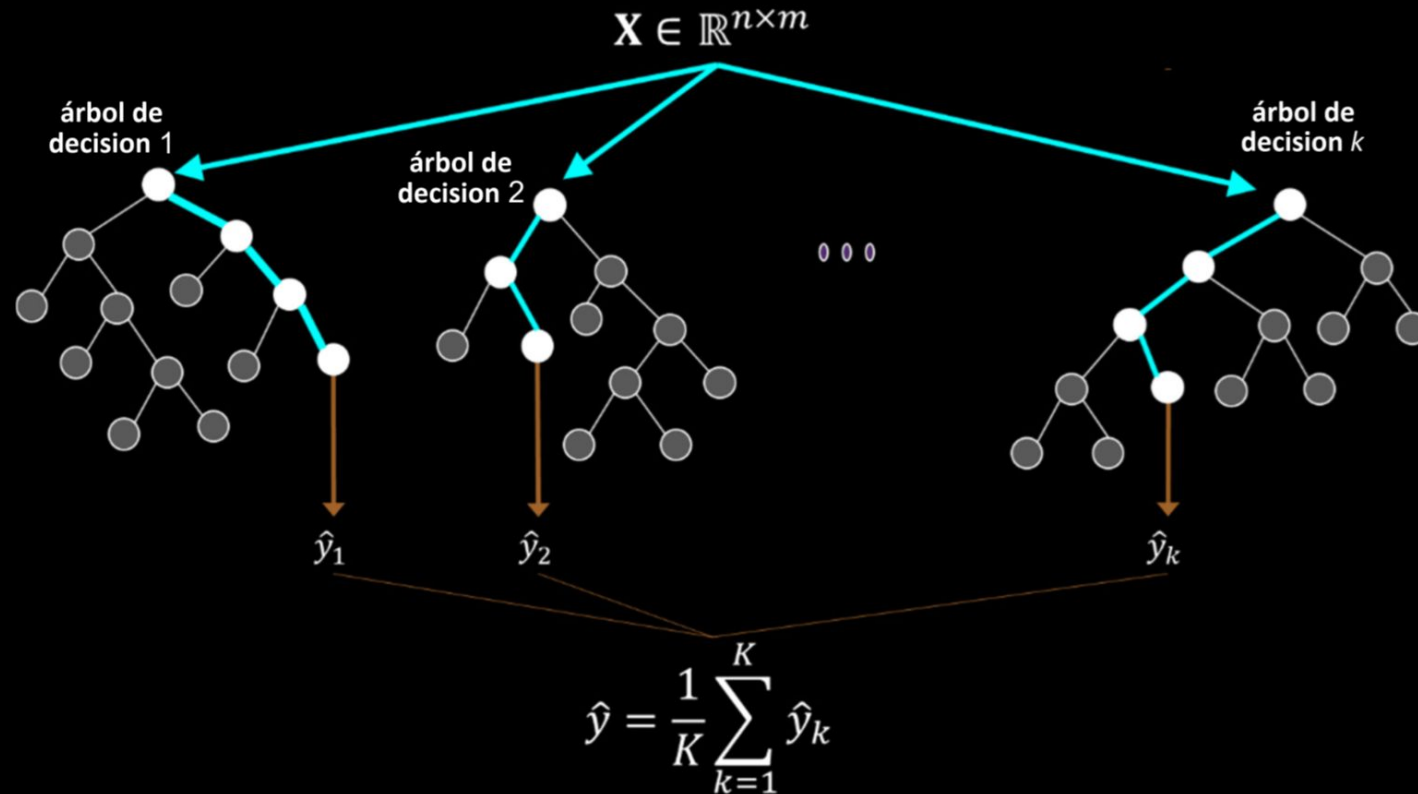


# Ejemplos de aplicaciones

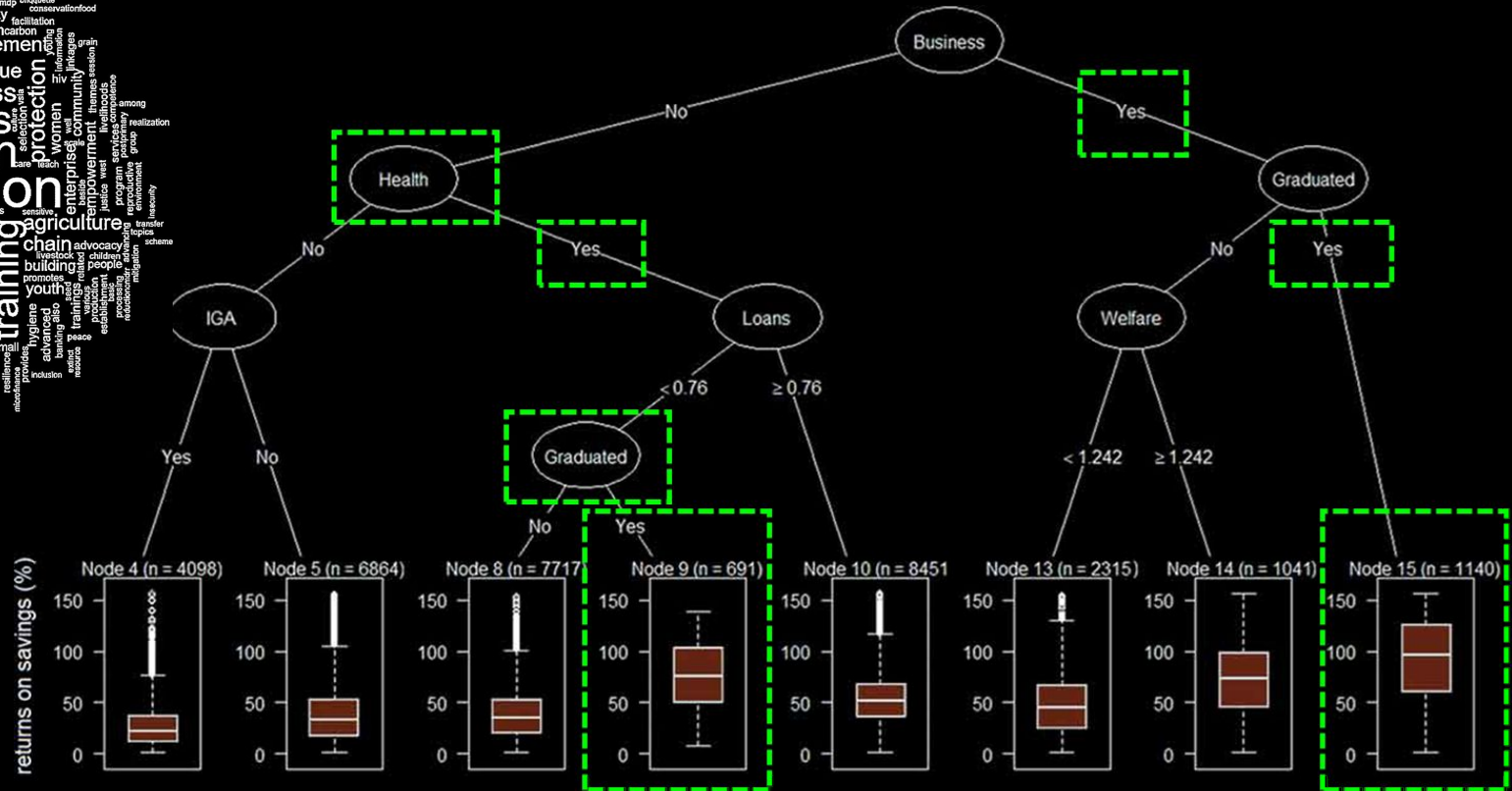
- Validación cruzada: la base de datos se dividió aleatoriamente en una muestra de entrenamiento y una muestra de validación
- 65% de los datos en el conjunto de entrenamiento.
- Se evaluó el error cuadrático medio (RMSE) y error de predicción absoluto medio (MAPE).



# Ejemplos de aplicaciones

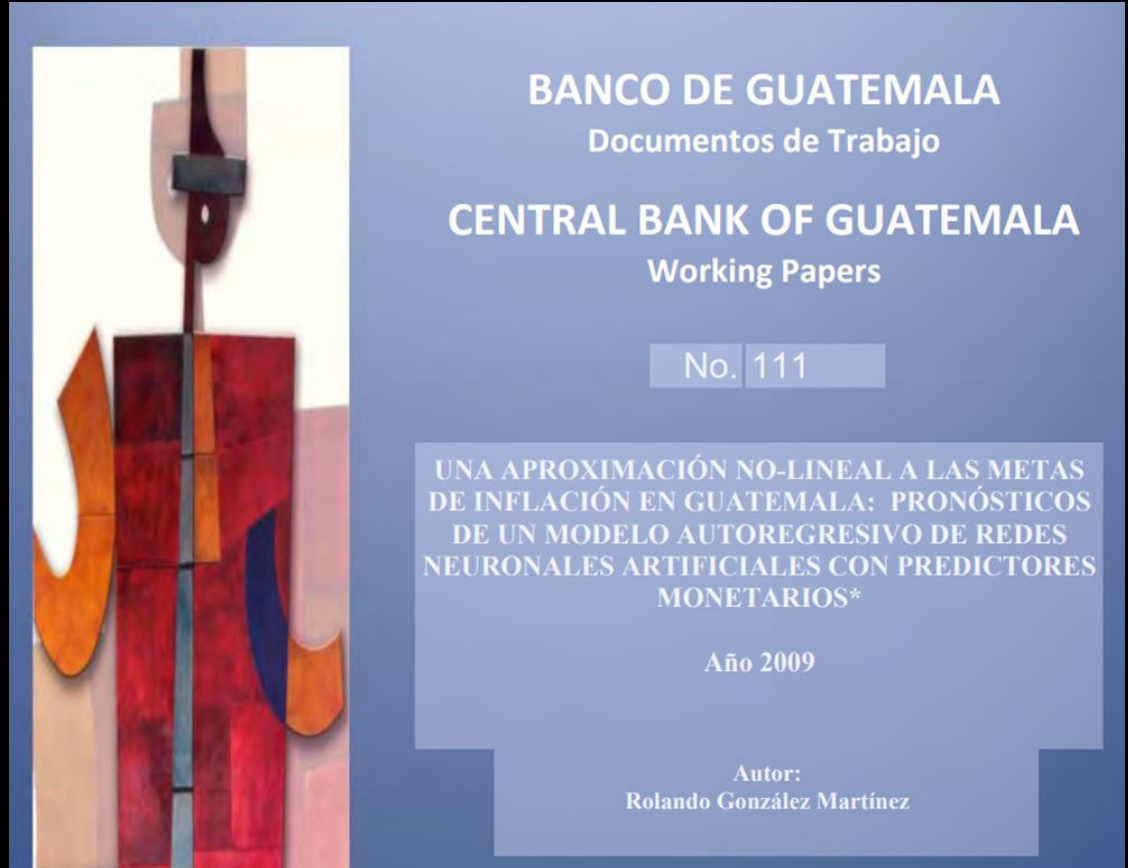


# Ejemplos de aplicaciones



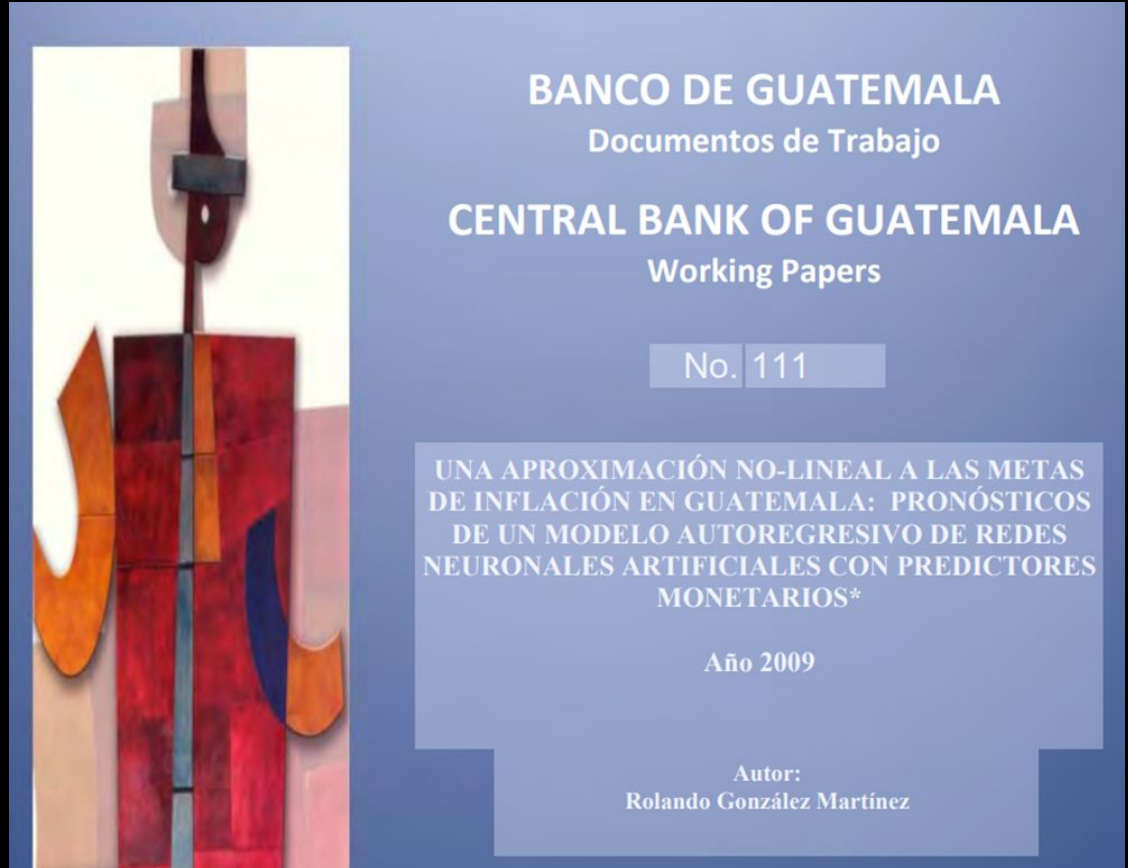
# Ejemplos de aplicaciones

- Pronóstico de la inflación con redes neuronales artificiales
- Pronóstico condicionado a la evolución de los agregados monetarios y la tasa de interés de referencia



# Ejemplos de aplicaciones

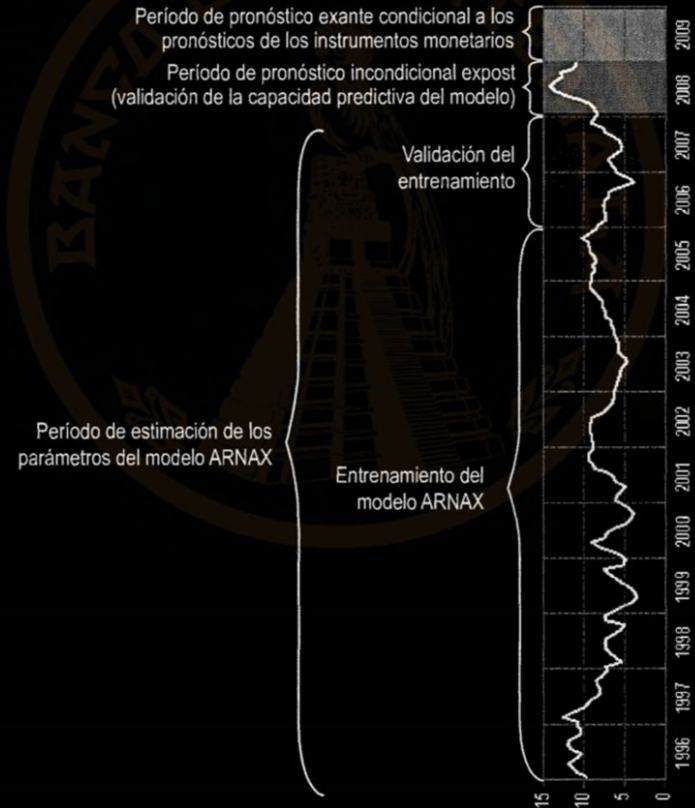
- Los bancos centrales tienen que cumplir la meta de inflación con cualquier instrumento
- Pronósticos precisos son necesarios y si la meta se cumple los bancos centrales aumentan su reputación ante los agentes económicos
- Modelos no-lineales pueden ser más precisos que modelos lineales



# Ejemplos de aplicaciones

Para ajustar y evaluar el poder predictivo del modelo ARNAX se dividió la serie de tiempo de la inflación en tres partes (véase también el gráfico 1):

Entrenamiento:	enero 1996 a diciembre 2007
Pronóstico expost:	enero 2008 a noviembre 2008
Pronóstico exante:	diciembre 2008 a diciembre 2009





# Ejemplos de aplicaciones

**Tabla 1. Estimación de los parámetros del modelo ARNAX y ajuste en el segmento de entrenamiento y validación**

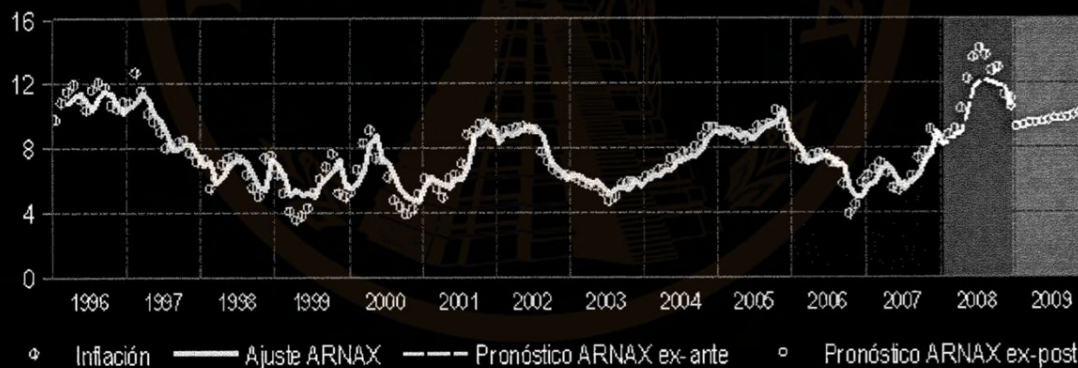
Capa de Salida		Capa oculta nodo I		Capa oculta nodo II	
Parámetros	Estimados	Parámetros	Estimados	Parámetros	Estimados
$\omega_0$	5.107	$\theta_{1,1}$	-5.179	$\theta_{2,1}$	0.345
$\omega_1$	7.384	$\theta_{1,2}$	0.595	$\theta_{2,2}$	-0.324
$\omega_2$	-1.231	$\theta_{1,3}$	-0.007	$\theta_{2,3}$	-0.086
		$\theta_{1,4}$	0.047	$\theta_{2,4}$	0.544
<b>Raíz de la suma de cuadrados de los errores</b>					
Segmento de entrenamiento			5.7198		
Segmento de validación			2.4917		

## Evaluación de los Pronósticos (expost)

Raíz del error cuadrático		
medio	3.5121	1.3182
Error promedio absoluto	2.9163	1.1426
Coefficiente de Theil	0.1724	0.0588

## Estadígrafo Diebold- Mariano para comparar la exactitud del pronóstico

Estadígrafo	2.713602	Valores críticos:	
p-value	0.003327	nivel del 1%	2.32
		nivel del 5%	1.64



# Ejemplos de aplicaciones

**Tabla 1. Estimación de los parámetros del modelo ARNAX y ajuste en el segmento de entrenamiento y validación**

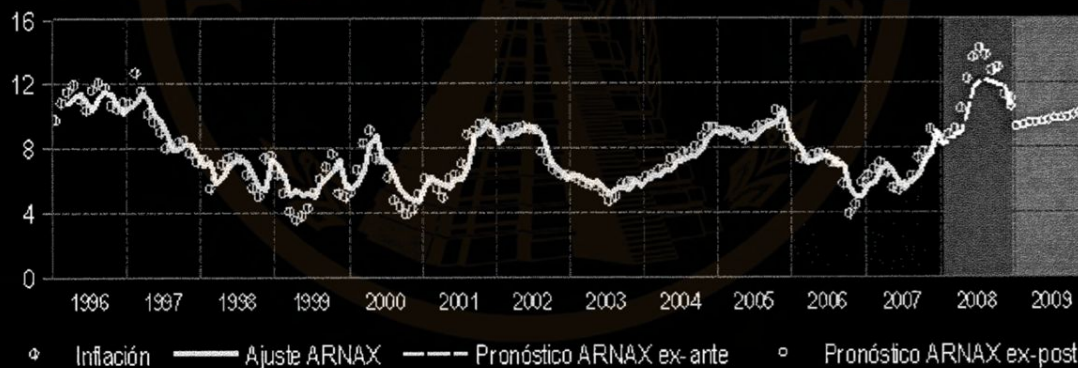
Capa de Salida		Capa oculta nodo I		Capa oculta nodo II	
Parámetros	Estimados	Parámetros	Estimados	Parámetros	Estimados
$\omega_0$	5.107	$\theta_{1,1}$	-5.179	$\theta_{2,1}$	0.345
$\omega_1$	7.384	$\theta_{1,2}$	0.595	$\theta_{2,2}$	-0.324
$\omega_2$	-1.231	$\theta_{1,3}$	-0.007	$\theta_{2,3}$	-0.086
		$\theta_{1,4}$	0.047	$\theta_{2,4}$	0.544
<b>Raíz de la suma de cuadrados de los errores</b>					
Segmento de entrenamiento			5.7198		
Segmento de validación			2.4917		

## Evaluación de los Pronósticos (expost)

Raíz del error cuadrático		
medio	3.5121	1.3182
Error promedio absoluto	2.9163	1.1426
Coefficiente de Theil	0.1724	0.0588

## Estadígrafo Diebold- Mariano para comparar la exactitud del pronóstico

Estadígrafo	2.713602	Valores críticos:	
p-value	0.003327	nivel del 1%	2.32
		nivel del 5%	1.64





# Ejemplos de aplicaciones

- **Problema:** en datos ruidosos, el ruido reduce la exactitud y precisión de los KPIs.
- **Objetivo:** metodología de dos pasos para calcular KPIs probabilísticos en conjuntos de datos ruidosos:
  - a. Optimización de enjambre para reducir el ruido
  - b. ML bayesiano para estimar KPI en datos con ruido reducido (relevance vector machines).

The screenshot shows the ScienceDirect website interface. At the top, the ScienceDirect logo and 'Journals & Books' are visible. Below the header, there are buttons for 'View PDF' and 'Download full issue'. The article title is 'How good is good? Bayesian machine-learning estimation of probabilistic benchmarks in noisy datasets and an application to nanofinance+'. The author is 'Rolando Gonzales Martinez'. The journal is 'Systems and Soft Computing', Volume 4, December 2022, 200036. The article is under a Creative Commons license and is open access. The left sidebar contains a table of contents with sections: Outline, Abstract, Keywords, Introduction, Multivariate probabilistic benchmarks, Estimation of multivariate probabilistic benchmarks, Empirical application: probabilistic benchmarks, Conclusion, Declaration of Competing Interest, References, and Figures (4).

ScienceDirect Journals & Books

View PDF Download full issue

Outline

Abstract

Keywords

1. Introduction

2. Multivariate probabilistic benchmarks

3. Estimation of multivariate probabilistic benchmarks

4. Empirical application: probabilistic benchmarks

5. Conclusion

Declaration of Competing Interest

References

Show full outline

Figures (4)

ELSEVIER

Systems and Soft Computing

Volume 4, December 2022, 200036

How good is good? Bayesian machine-learning estimation of probabilistic benchmarks in noisy datasets and an application to nanofinance+

Rolando Gonzales Martinez<sup>1</sup>

Show more

+ Add to Mendeley Share Cite

<https://doi.org/10.1016/j.sasc.2022.200036>

Get rights and content

Under a Creative Commons license

open access

# Ejemplos de aplicaciones

## Box 1. Pseudo-code of the swarm algorithm

**Data:**  $\{y_1, y_2, \dots, y_j\} \ni \mathbf{y}$

**Result:**  $\psi, \psi^\perp$

initialization;

$\delta, M, \theta_0, p_0, p_0^\perp, \zeta, \zeta^*$  ;

**while**  $m \in \mathbb{Z}_+$  **do**

$w_\delta = \delta \frac{\|p\|}{\|p^\perp\|}, \quad w_\delta^\perp = \delta \frac{\|p^\perp\|}{\|p\|} ;$

**for**  $m \leftarrow 1$  **do**

random exploration of hyperbola parameters;

$p_m = p_{m-1} + w_\delta \epsilon, \quad p_m^\perp = p_{m-1}^\perp + w_\delta^\perp \epsilon, \quad \epsilon \sim (0, 1) ;$

hyperbolic undersampling;

$y_h = f_h(p_m, \mathbf{y}), \quad y_h^\perp = f_h^\perp(p_m^\perp, \mathbf{y}), \quad \{y_h, y_h^\perp\} \ni \mathbf{y}_h ;$

copula dependence estimated with filtered  $m$ -samples;

$\hat{\theta}_m = C_\theta(\mathbf{y}_h) ;$

**end**

$\hat{\theta}^* = \max \{\hat{\theta}_i\}_{i=1}^m$  (optimal dependence);

$p^* = p(\hat{\theta}^*), \quad p^{\perp*} = p^\perp(\hat{\theta}^*)$  (optimal hyperbola parameters);

cohesion =  $\frac{1}{2} (\|p_m - p_m^*\| + \|p_m^\perp - p_m^{\perp*}\|) ;$

separation =  $\frac{1}{2} (\|p_m - \bar{p}_m^*\| + \|p_m^\perp - \bar{p}_m^{\perp*}\|) ;$

**if**  $\hat{\theta}^* > \hat{\theta}^{m-1}$  **then**

$\hat{\theta}_m = \hat{\theta}^* ;$

$p_m = p^*, \quad p_m^\perp = p^{\perp*} ;$

alignment;

$\delta_m = \delta_{m-1}(\zeta^*) ;$

$m = M - 1 ;$

**else**

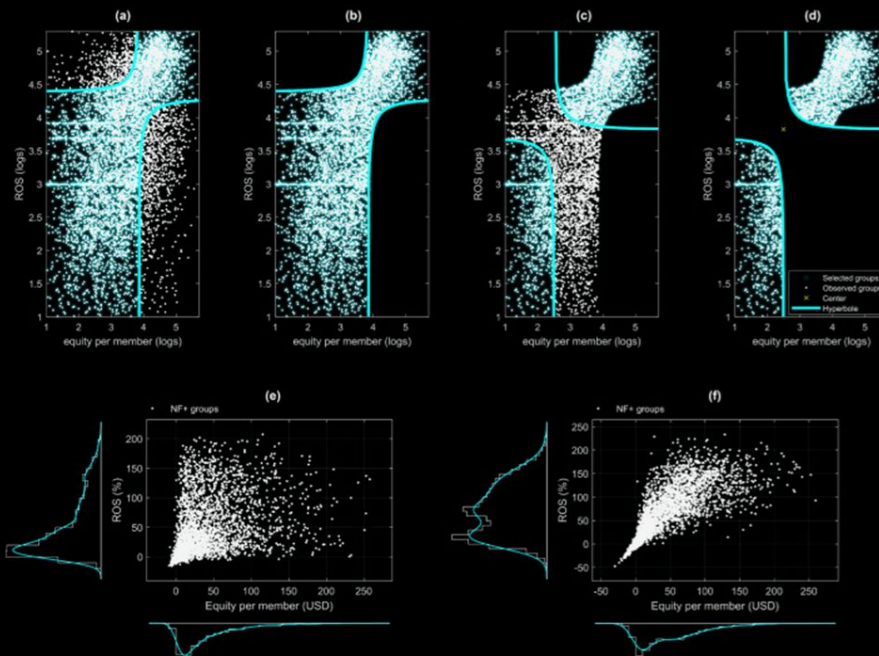
$\delta_m = \delta_{m-1}(\zeta) ;$

**end**

**end**

$$p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2) = \frac{p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\alpha, \sigma^2)},$$

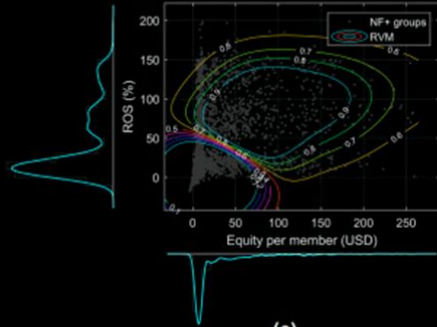
$$= (2\pi)^{-(k+1)/2} \left| \Sigma \right|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{w} - \mu)^\top \Sigma^{-1} (\mathbf{w} - \mu) \right\}^2$$



# Ejemplos de aplicaciones

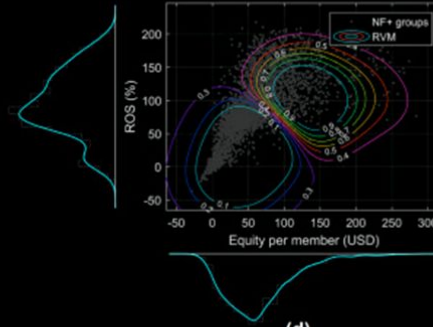
Rural nano-finance(+)

(a)

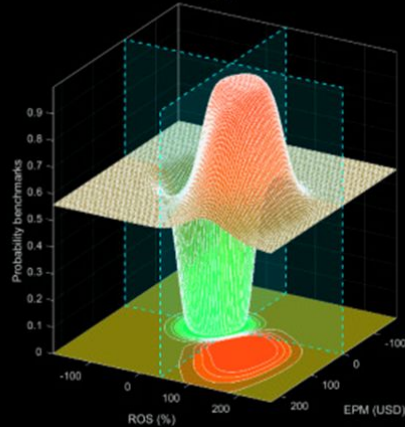


Urban nano-finance(+)

(b)



(c)



(d)

