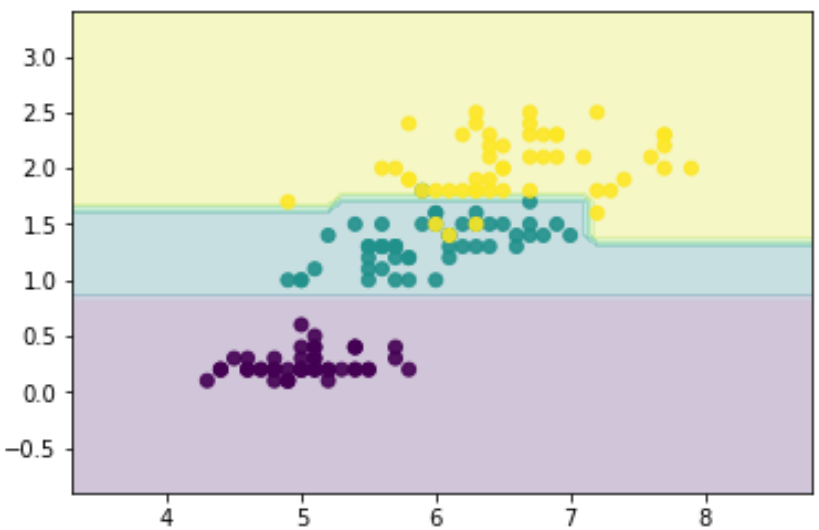
Análisis del Boosting en entornos con ruido de clase

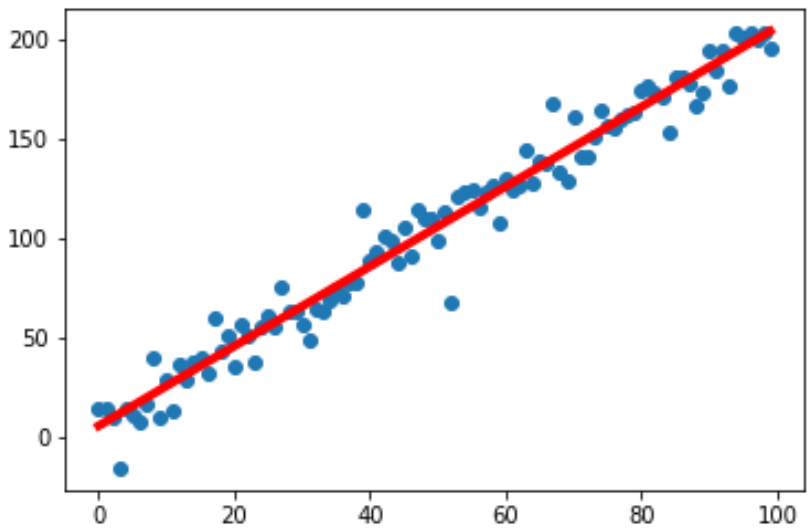
Rafael Nogales Vaquero

Índice

Problemas en Machine Learning

Clasificación es la tarea de asignar una clase a cada instancia.
La regresión tiene el objetivo de predecir valores continuos





Problema de Regresión

Fijar el precio de una vivienda basandonos en parámetros como:

- Cantidad de baños
- Metros cuadrados
- Nº de habitaciones
- Barrio
- Número de coches en parking de Ikea más cercano
- ...

Problema de Regresión

Heatmap GR

$$h : X \to \mathbb{R}$$

Clasificación clases de Iris

Figura: Clases de flores de Iris.

El problema de Fisher

Ronald Fisher construyó el dataset en 1936.
Consiste en 50 muestras de cada especie.

Iris Fisher Dataset

¿Quién es quién?

$$h : X \rightarrow Y$$

- h es el clasificador
- X es el espacio del que salen las instancias
- Y es el conjunto de todas las clases

De la realidad a los vectores

En los problemas reales tenemos objetos o sucesos que queremos clasificar. Pero no nos encontramos con vectores dirctamente.

$$M : \text{Realidad} \to X$$

De la realidad a los vectores

En los problemas reales tenemos objetos o sucesos que queremos clasificar. Pero no nos encontramos con vectores dirctamente.

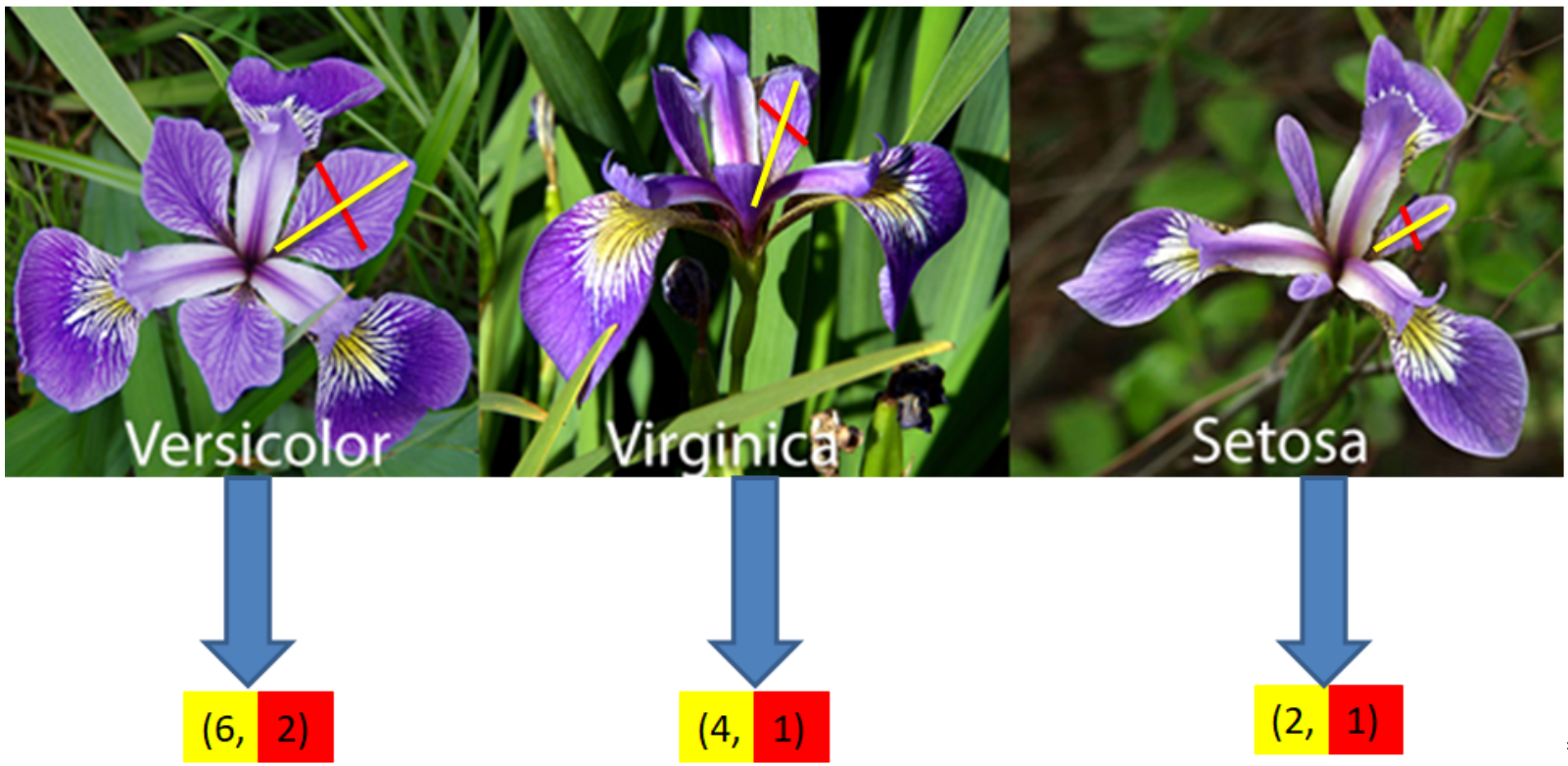
$$M : \text{Realidad} \rightarrow X$$

Transformamos la realidad a vectores porque la función h trabaja con vectores

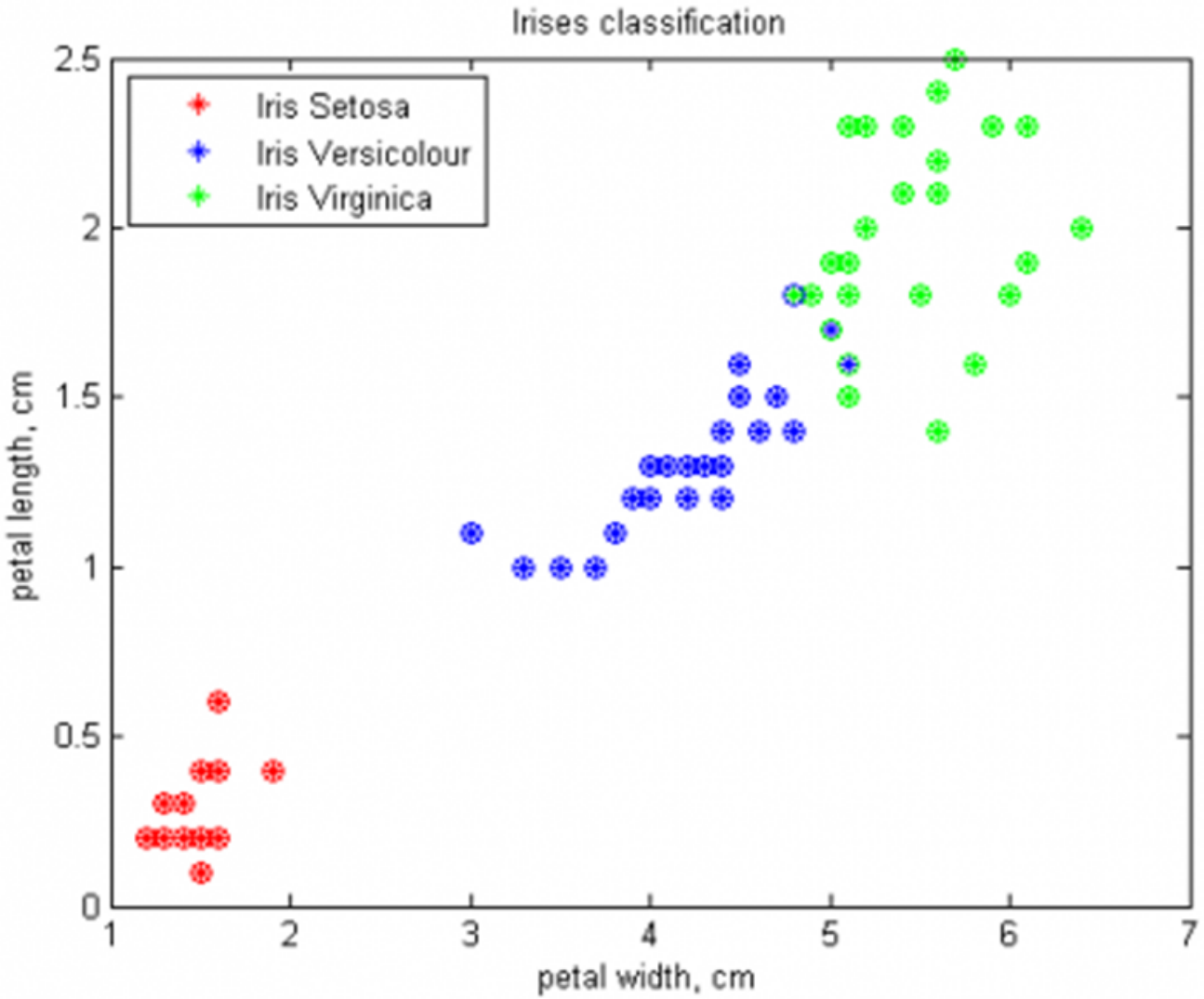
$$h : X \rightarrow Y$$

Realidad y vectores

Figura: De la realidad a los Vectores



El espacio X


Irises classification
Iris Setosa
Iris Versicolour
Iris Virginica
petal length, cm
petal width, cm
0
0.5
1
1.5
2
2.5
1
2
3
4
5
6
7

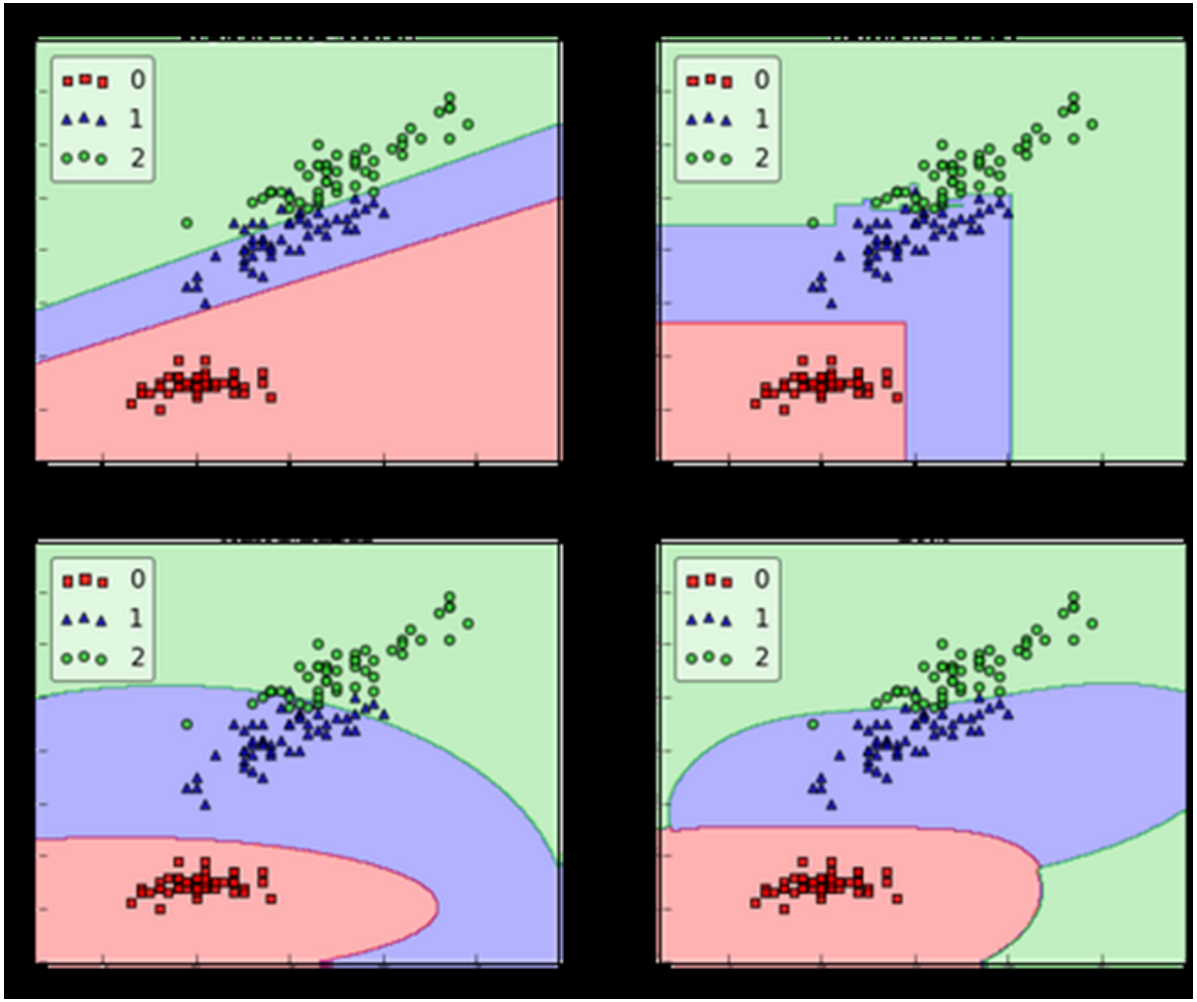
¿Qué es un clasificador?

Formalmente: $h : X \rightarrow Y$

Es una función que asigna a cada instancia una clase.

Gráficamente, es una regla para colorear el espacio en blanco.

Clasificadores hay muchos...



¿Con cuál me quedo?

Necesitamos saber cuándo un clasificador es mejor que otro.

Descripción del entorno

- Conjunto del que podemos obtener muestras X y clases de las muestras Y
- Conjunto de entrenamiento etiquetado $\{(x_1, y_1), ..., (x_m, y_m)\}$
- (x_i, y_i) son muestras i.i.d de $P(X, Y)$
- Buscamos h verificando $h(x_i) = y_i$
- P es desconocida

Concepto de Riesgo

Un clasificador es mejor que otro cuando su riesgo es menor.
Se define el riesgo como:

$$R(h) = \mathbf{E}[L(h(x), y)] = \int L(h(x), y) dP(x, y)$$

Dónde $L(h(x), y)$ es la función de pérdida, es decir, la medida del error entre nuestra predicción $h(x)$ y la realidad y

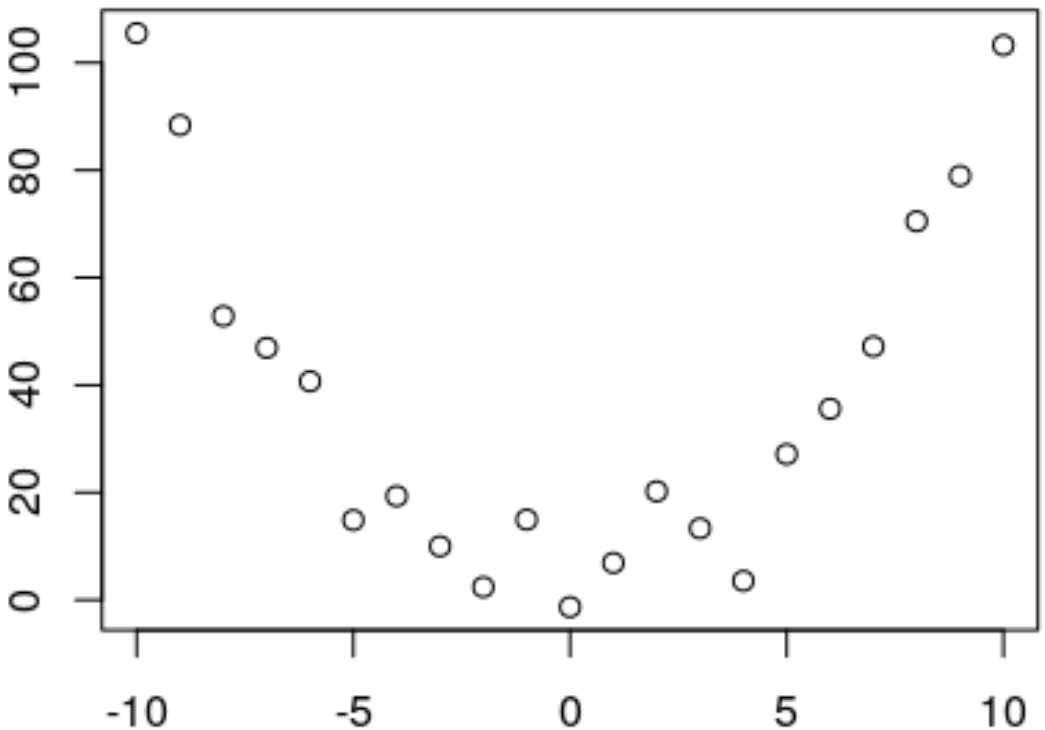
El mejor clasificador

$$R(h) = \mathbf{E}[L(h(x), y)] = \int L(h(x), y) dP(x, y)$$

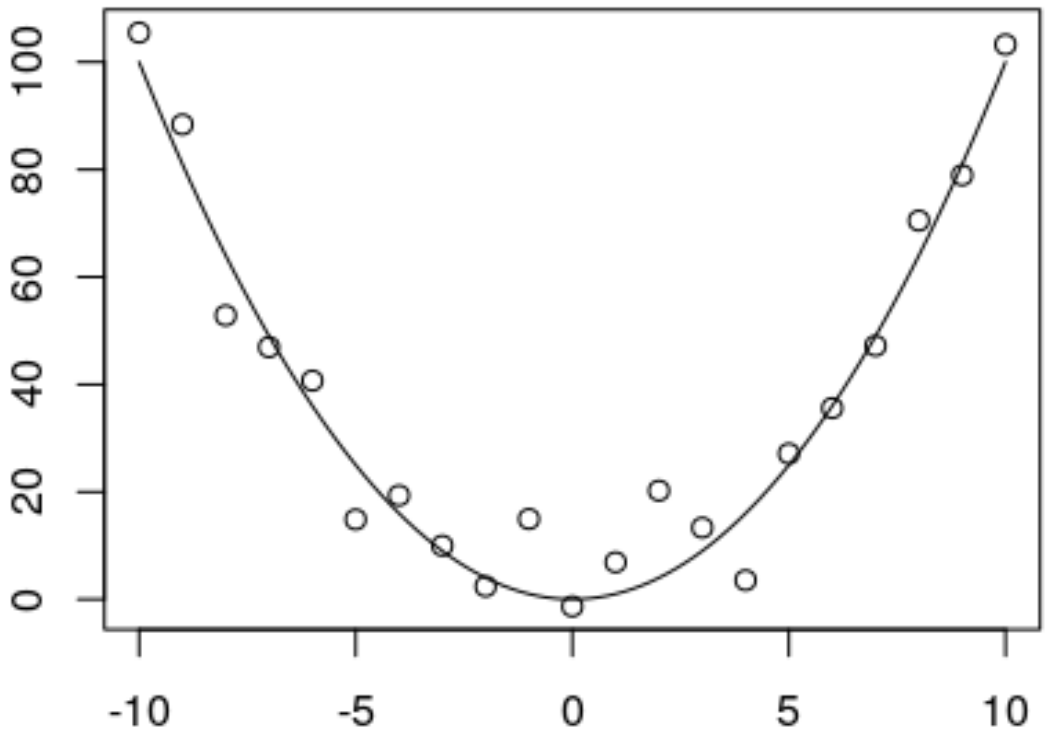
$$h^* = \arg\min_{h \in \mathcal{H}} R(h)$$

Dónde $\mathcal{H}$ es una famila de funciones definida a priori.

Distribución de probabilidad desconocida



Distribución de probabilidad desconocida



Vuelta a la realidad...

$$R(h) = \mathbf{E}[L(h(x), y)] = \int L(h(x), y) dP(x, y)$$

¿Pero cómo calcular $R(h)$ si no conocemos P ?

Vuelta a la realidad...

$$R(h) = \mathbf{E}[L(h(x), y)] = \int L(h(x), y) dP(x, y)$$

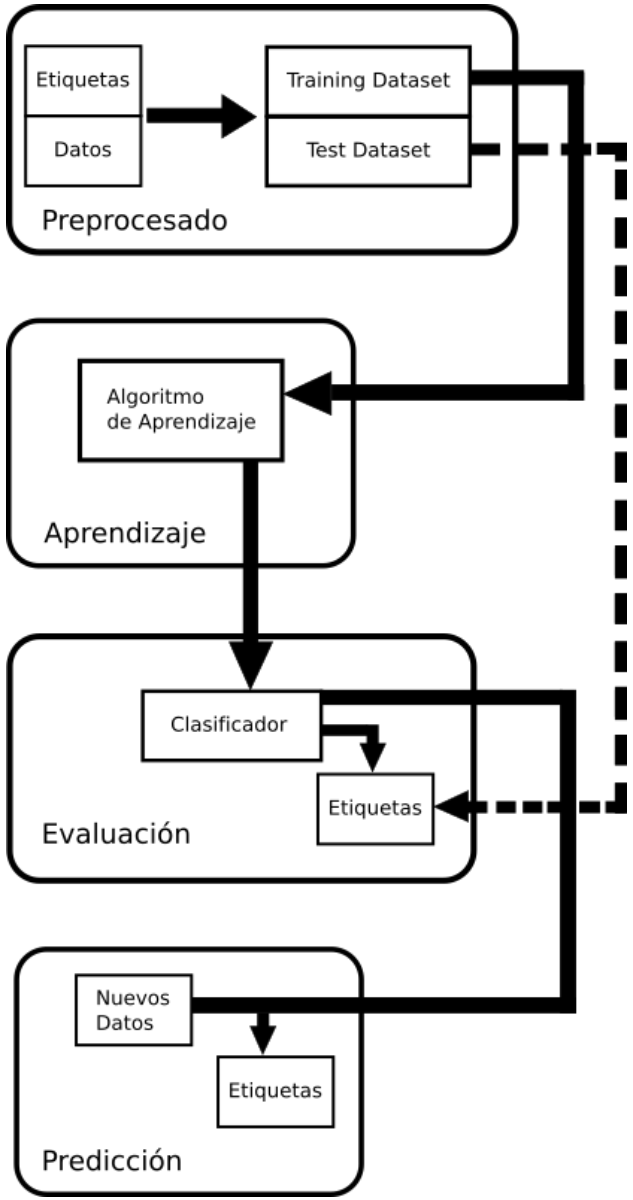
¿Pero cómo calcular $R(h)$ si no conocemos P ?

No podemos, pero podemos aproximar R utilizando lo que tenemos:
El conjunto de entrenamiento etiquetado $\{(x_1, y_1), ..., (x_m, y_m)\}$

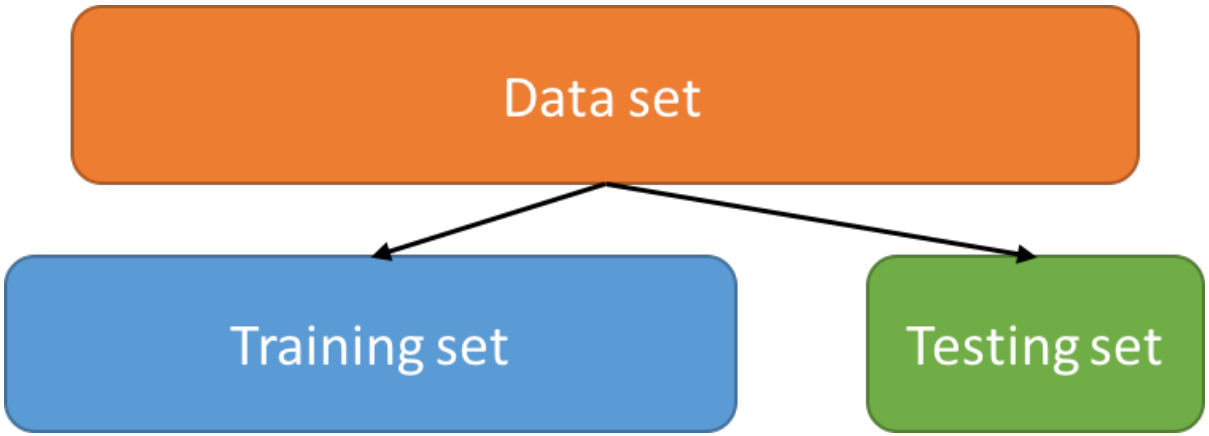
Concepto de Riesgo Empírico

$$R_{\text{emp}}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i)$$

$$\hat{h} = \arg\min_{h \in \mathcal{H}} R_{\text{emp}}(h)$$

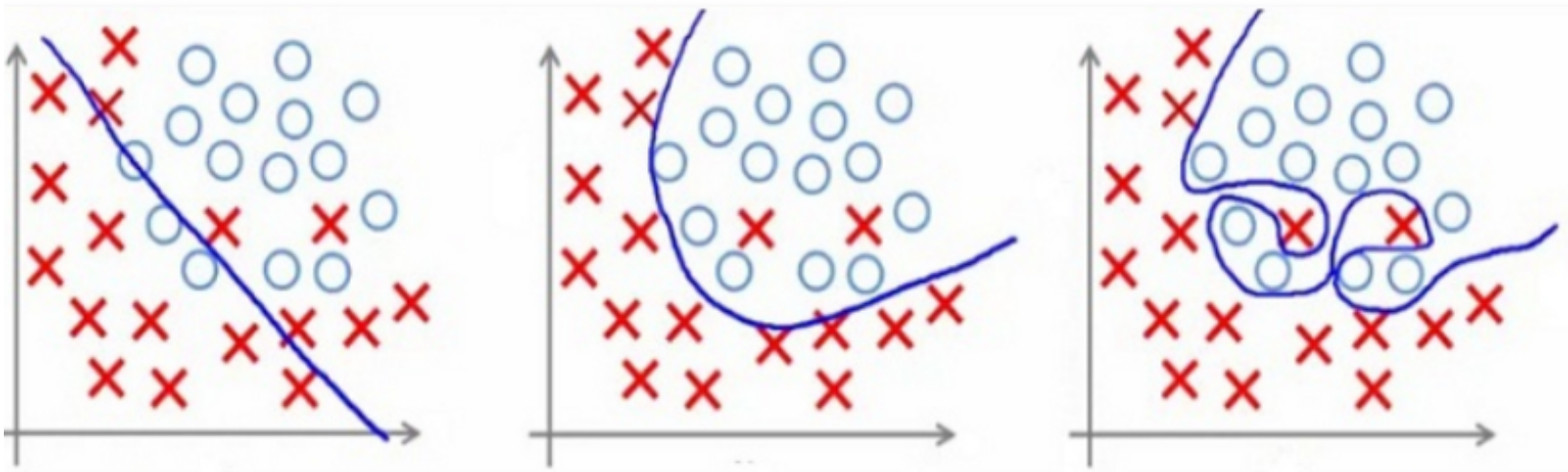

Etiquetas
Datos
Training Dataset
Test Dataset
Preprocesado
Algoritmo
de Aprendizaje
Aprendizaje
Clasificador
Etiquetas
Evaluación
Nuevos
Datos
Etiquetas
Predicción

Train-Test Split



Building the model

Underfitting-Overfitting



Under-fitting

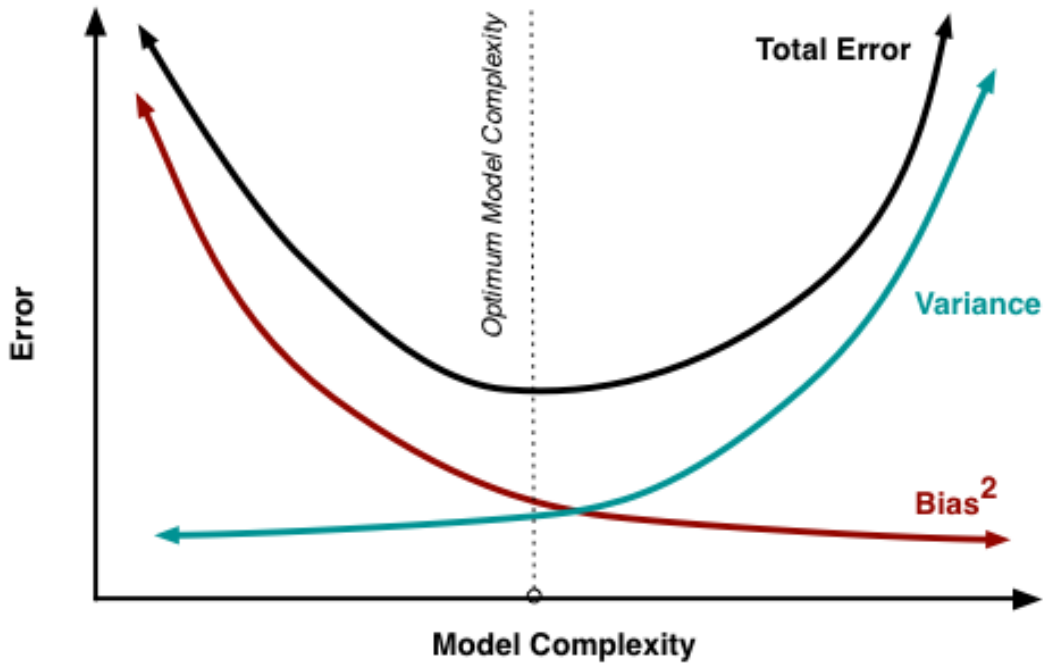
(too simple to explain the variance)

Appropriate-fitting

Over-fitting

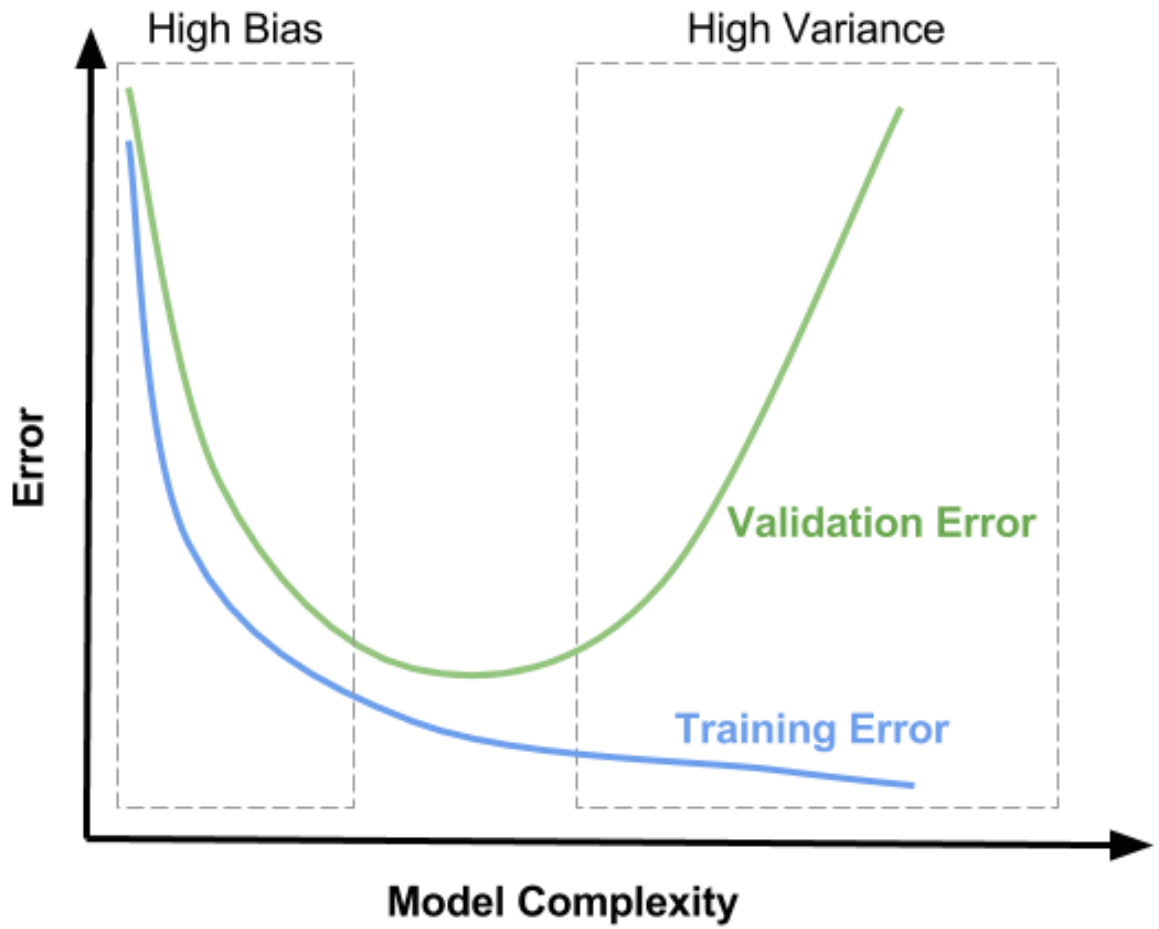
(forcefitting -- too good to be true)

Bias-Variance Tradeoff



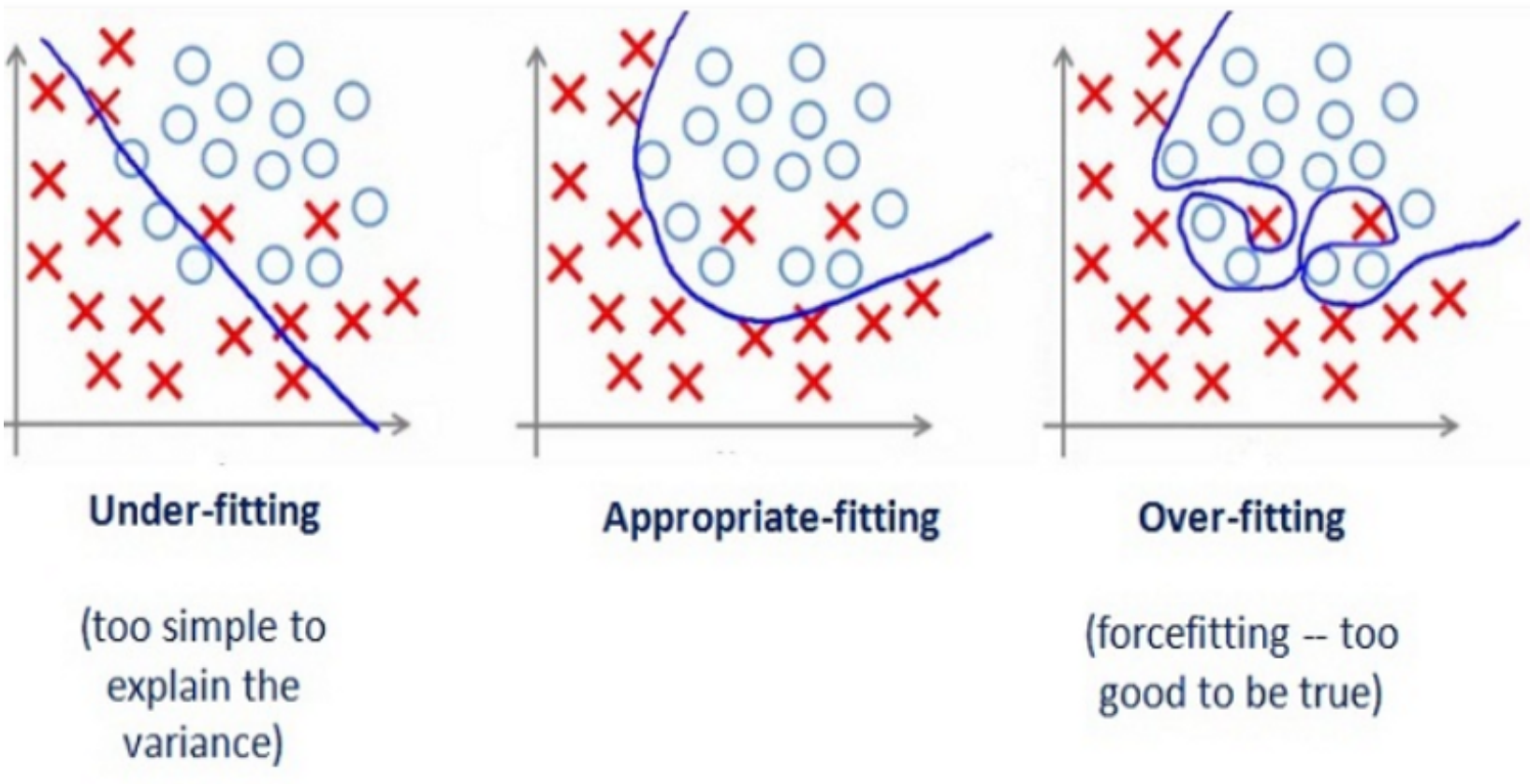
Bias-Variance Tradeoff

Bias-Variance Tradeoff en la práctica:



El ruido

El ruido son instancias mal clasificadas.



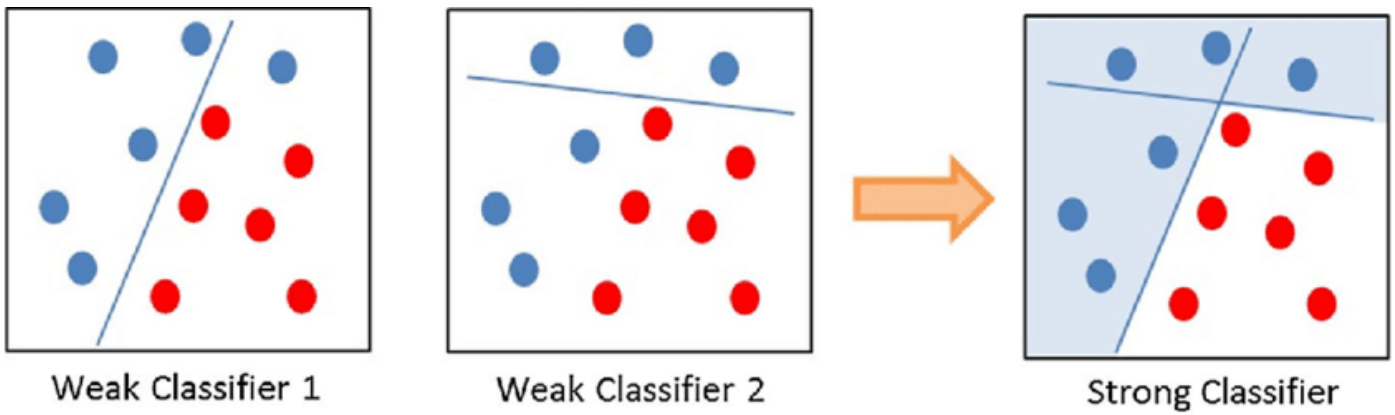
Ruido y overfitting

- El ruido en pequeña cantidad no es problematico generalmente
- Un buen modelo no debe aprender el ruido
- Un buen modelo generaliza

Métodos de ensemble

Figura: Parlamento de Budapest

Clasificadores débiles



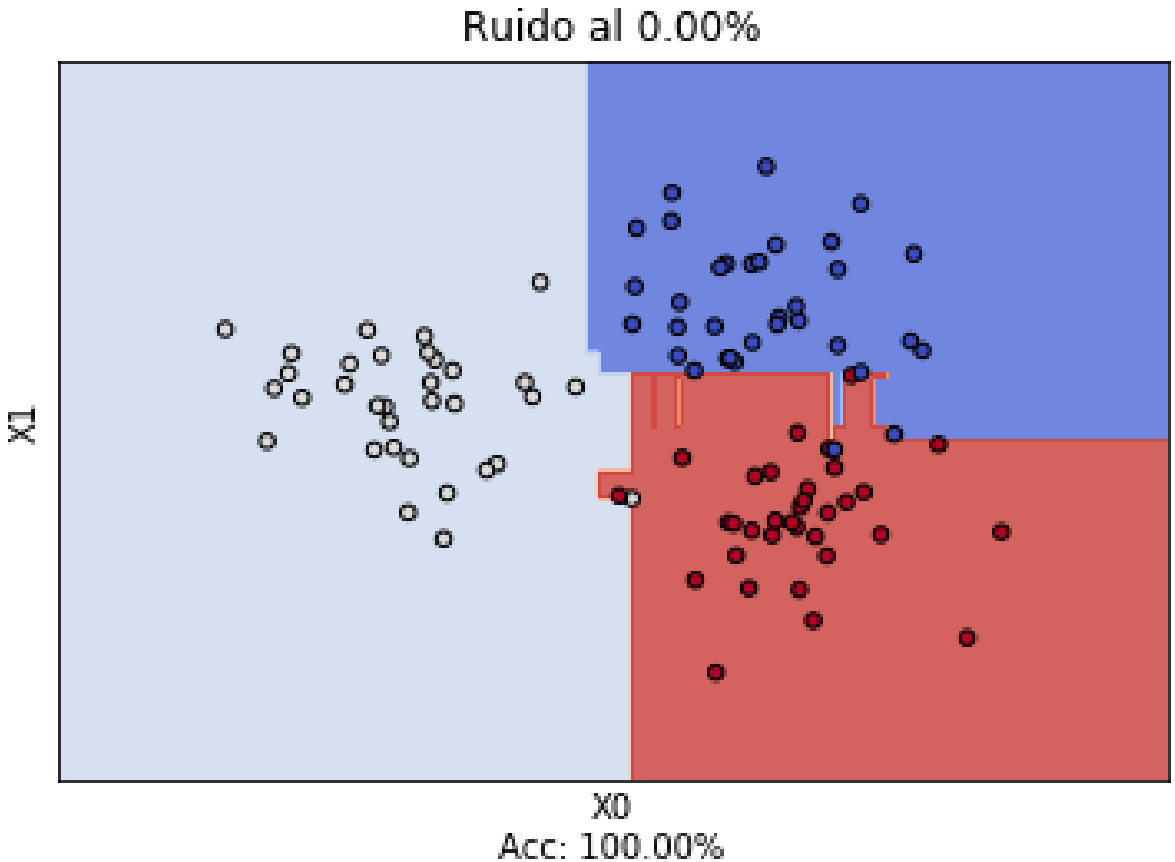
Shafique, Muhammad & Hato, Eiji. (2015). Use of acceleration data for transportation mode prediction. Transportation. 42. 163-188. 10.1007/s11116-014-9541-6.

XGBoost

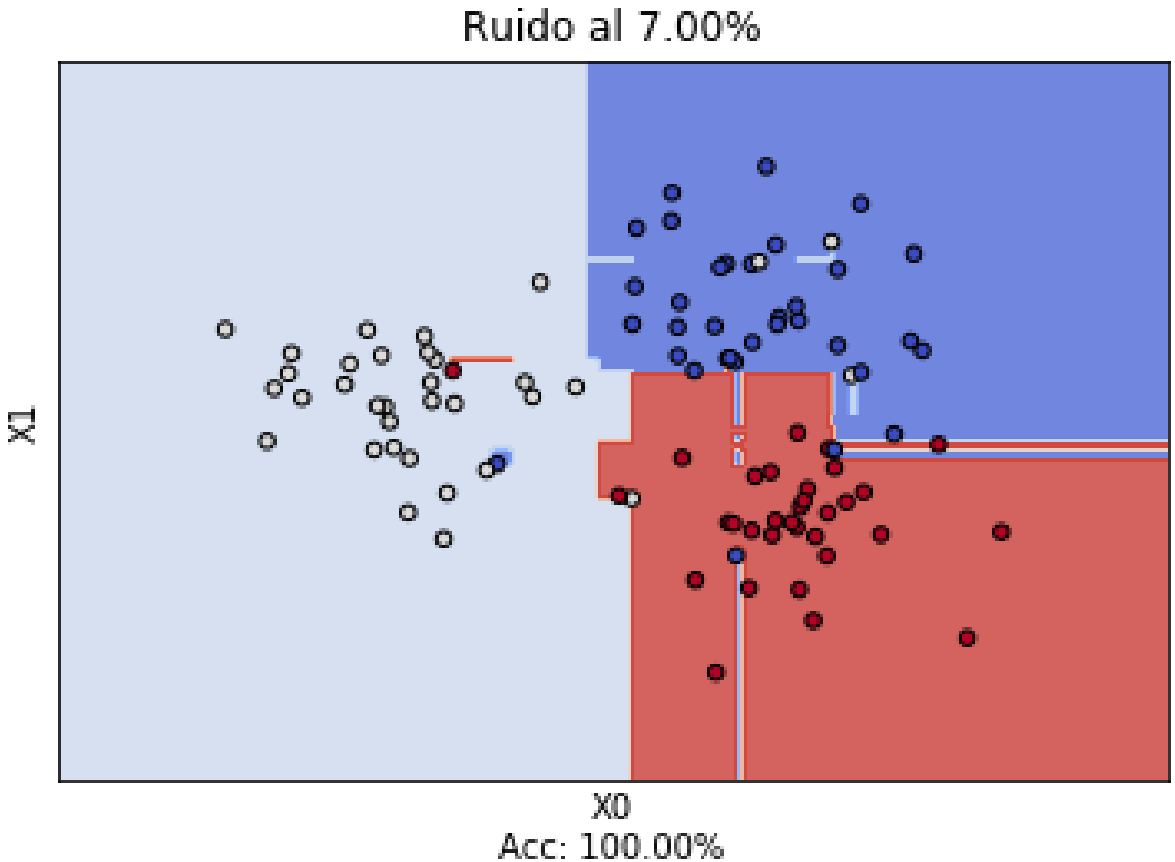
XGBoost es una implementación de Gradient Boosting que aprovecha al máximo los recursos hardware disponibles.

- Escrito en C++
- Soporte para sistemas distribuidos
- Compatible con scikit-learn (Python)
- Compatible con caret (R)
- Soporte para Julia
- Soporte para Java (Scala, Hadoop...)

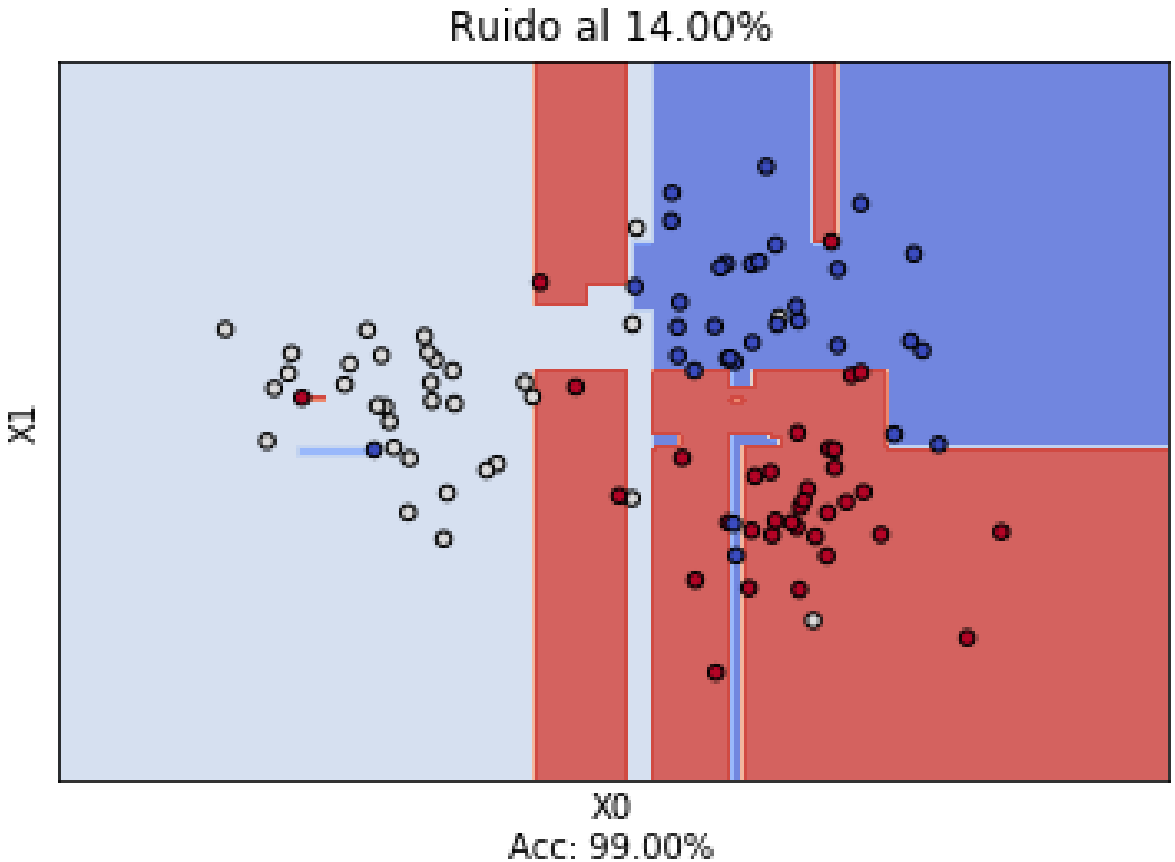
XGBoost en entornos Ruidosos



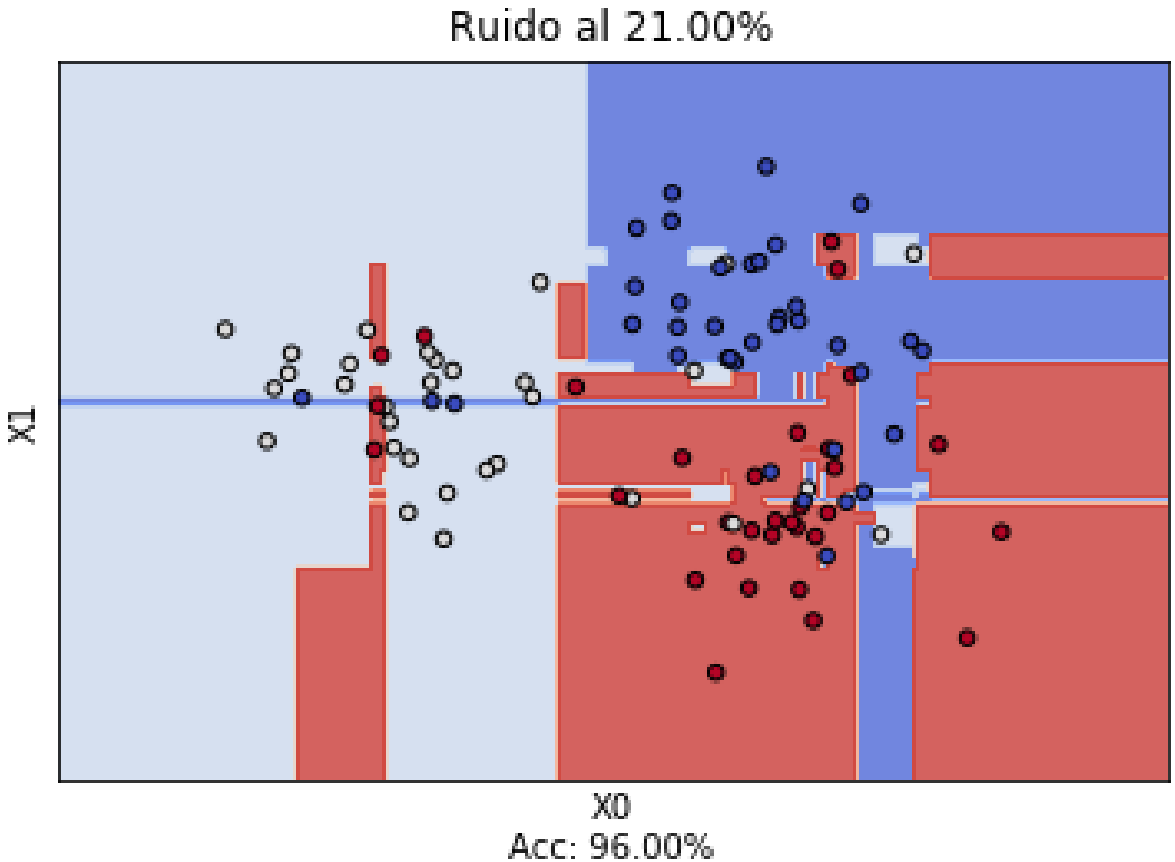
XGBoost en entornos Ruidosos



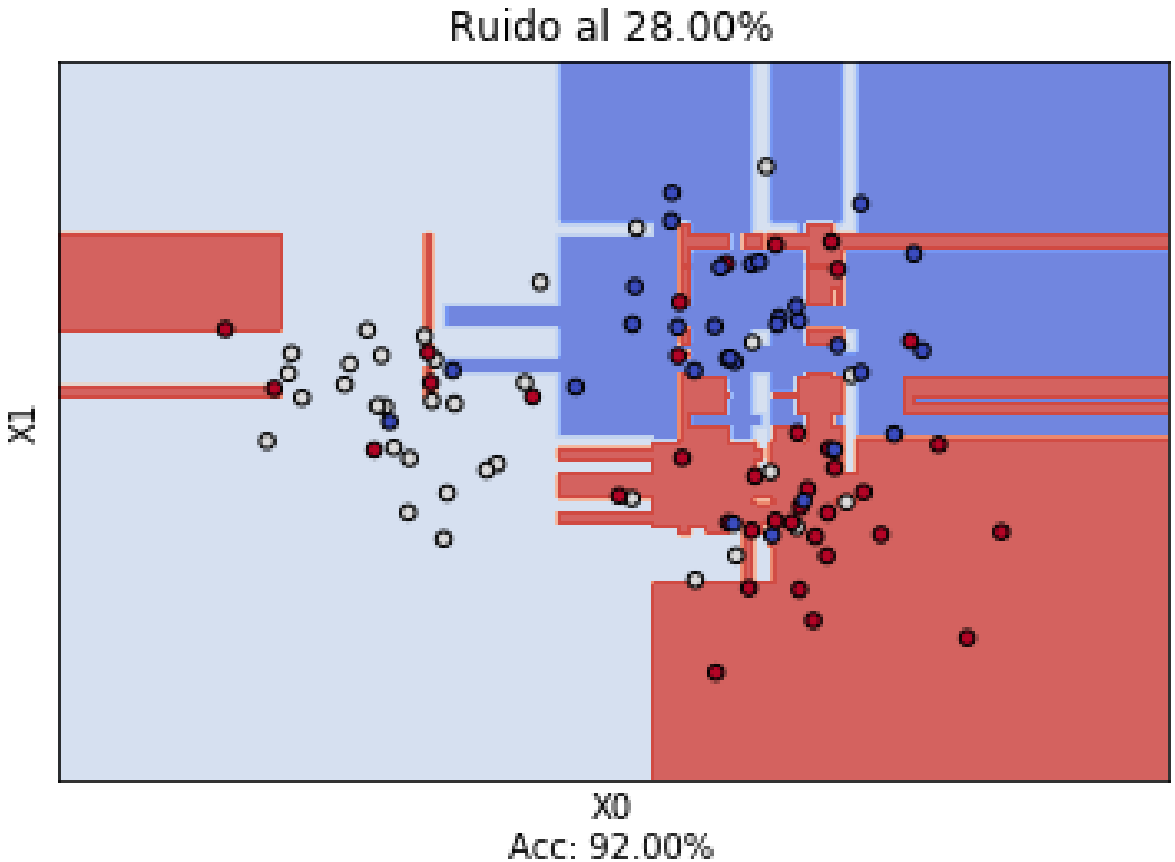
XGBoost en entornos Ruidosos



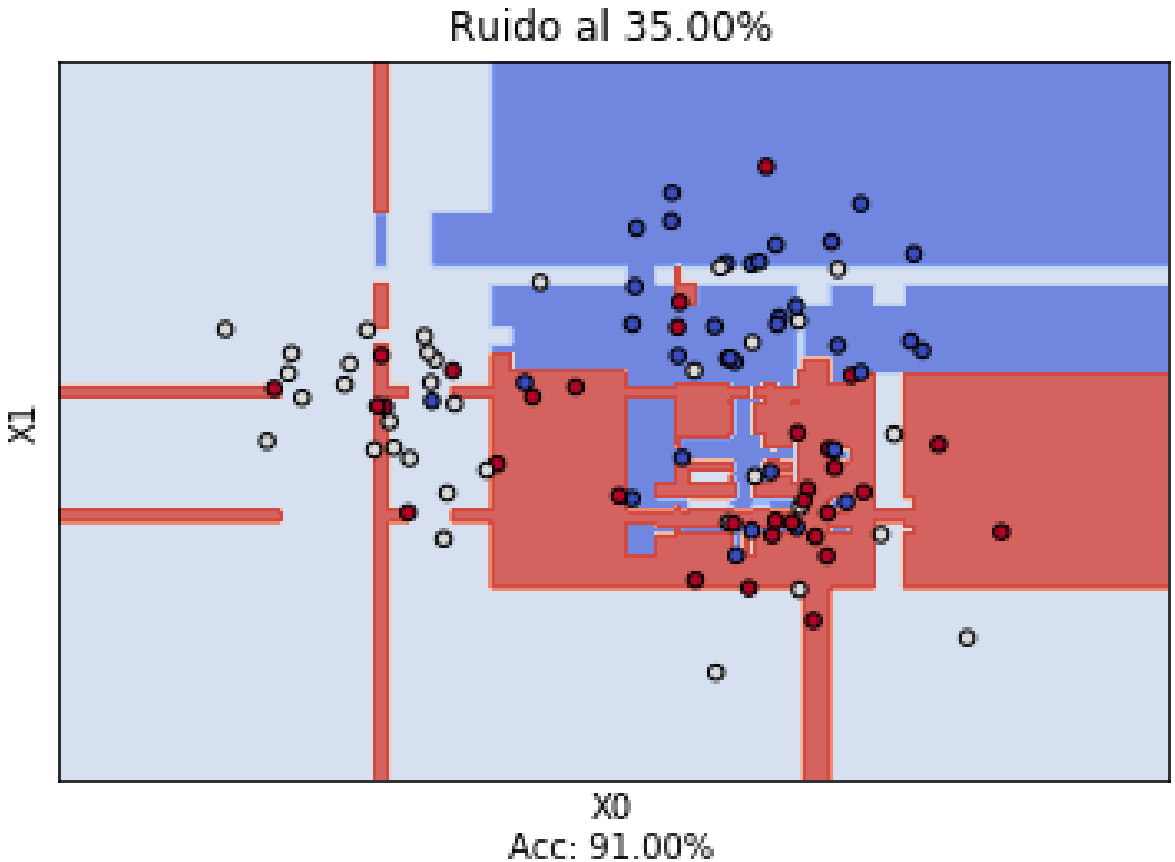
XGBoost en entornos Ruidosos



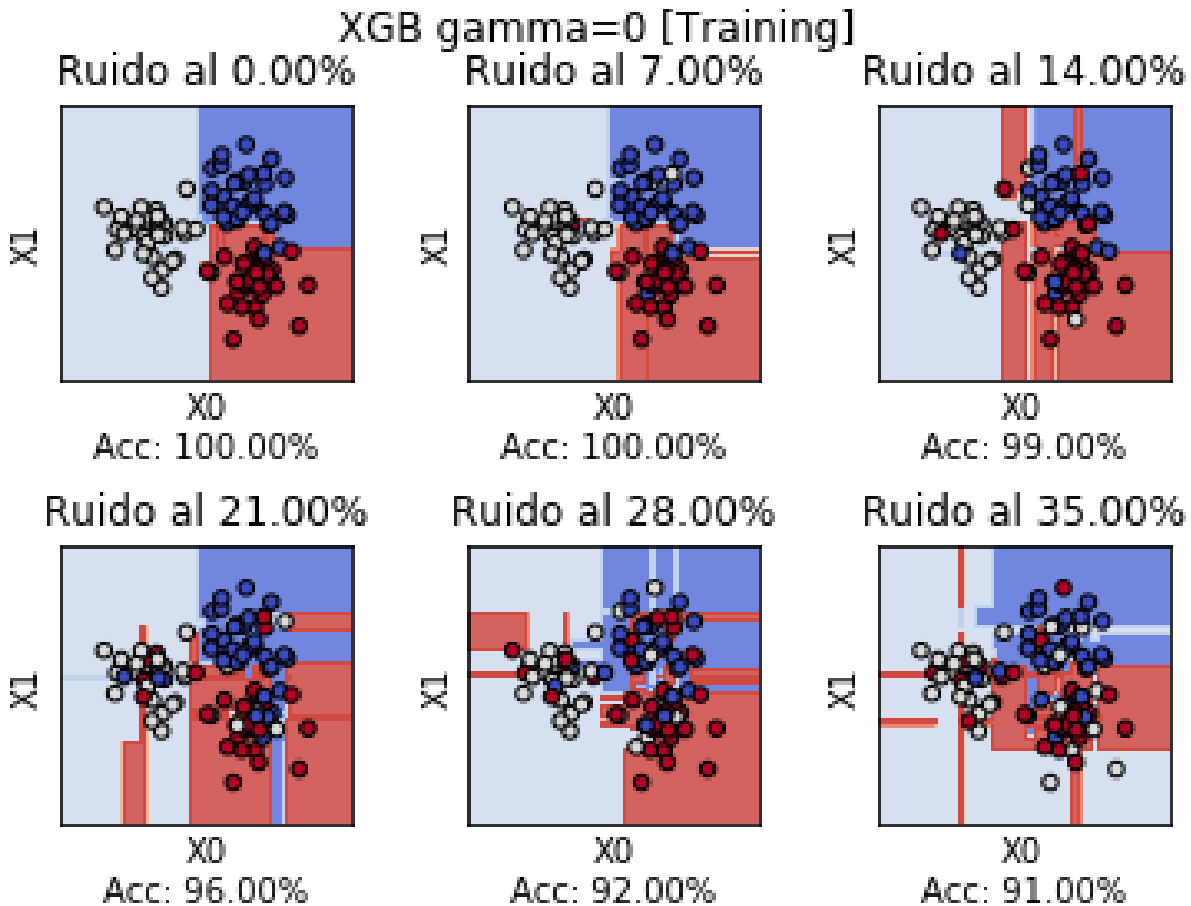
XGBoost en entornos Ruidosos



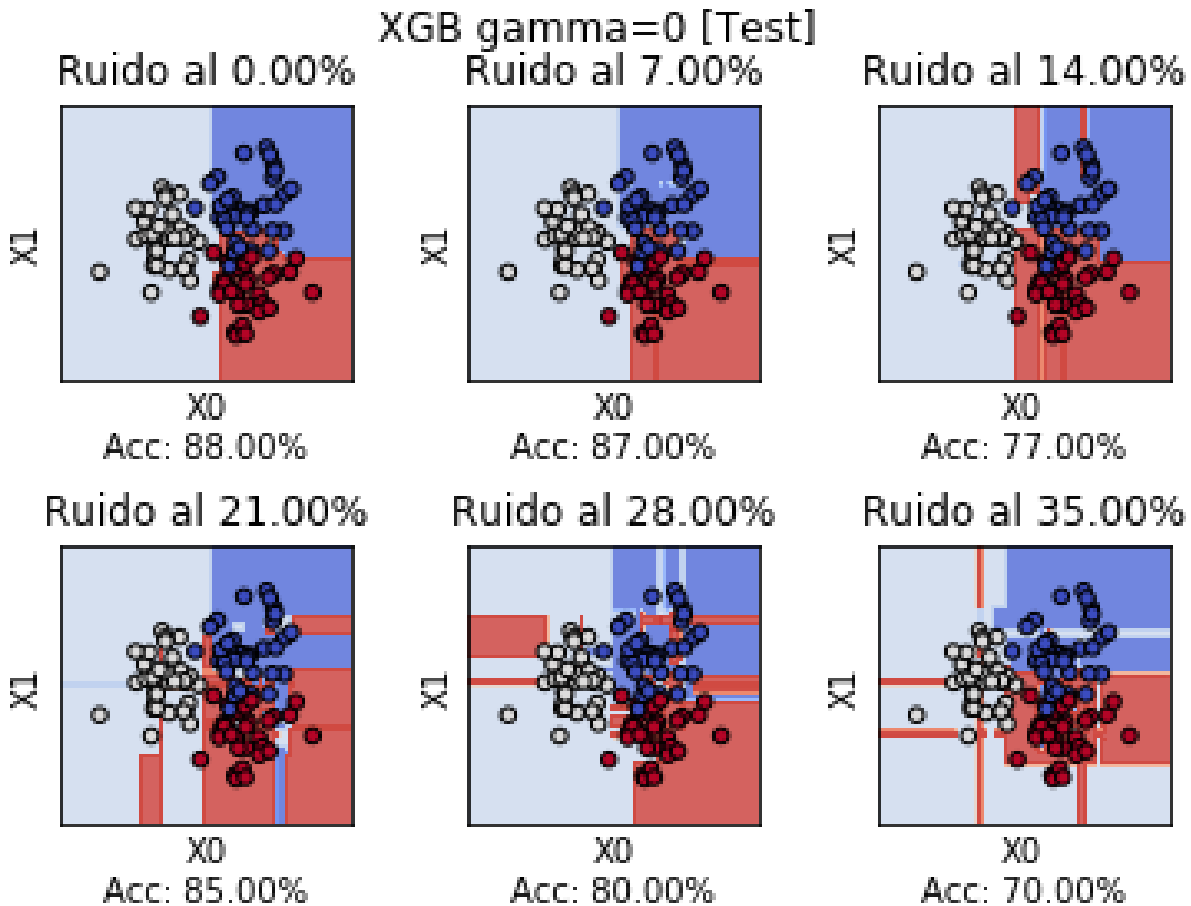
XGBoost en entornos Ruidosos



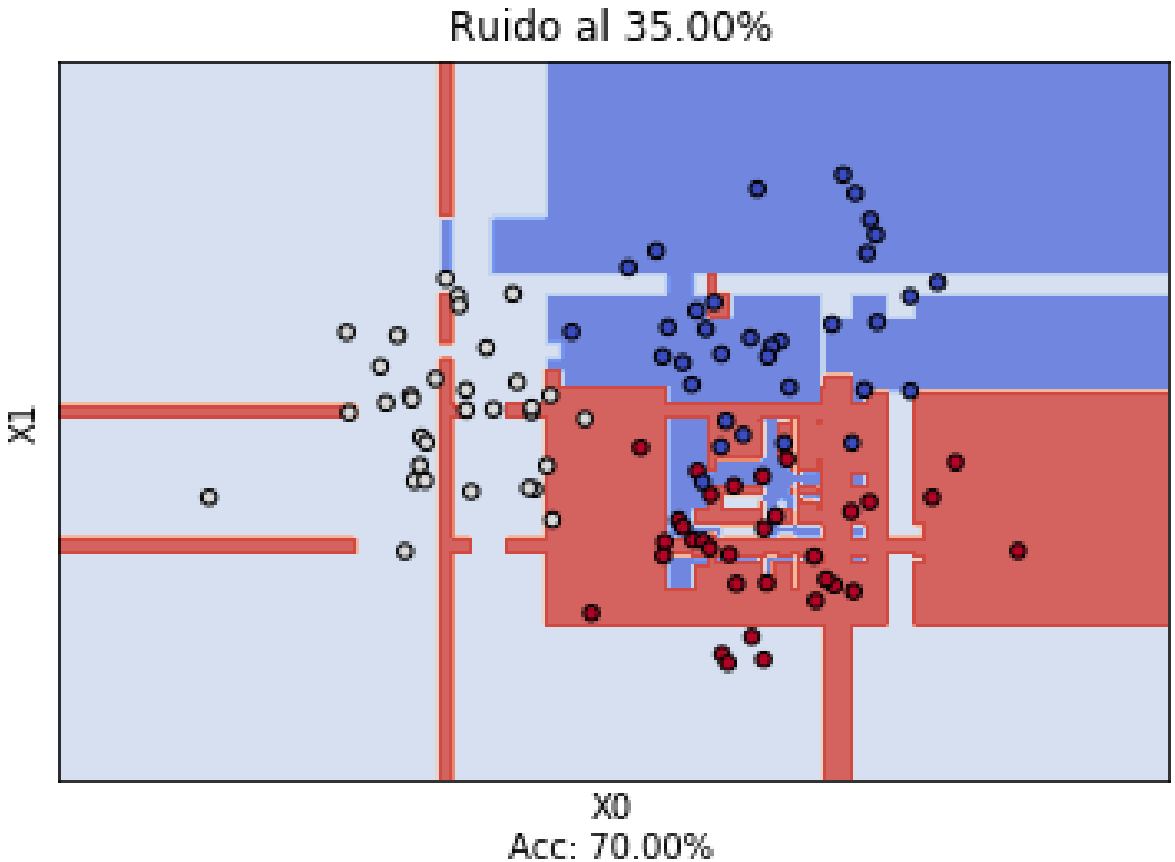
XGBoost resumen entornos ruidosos



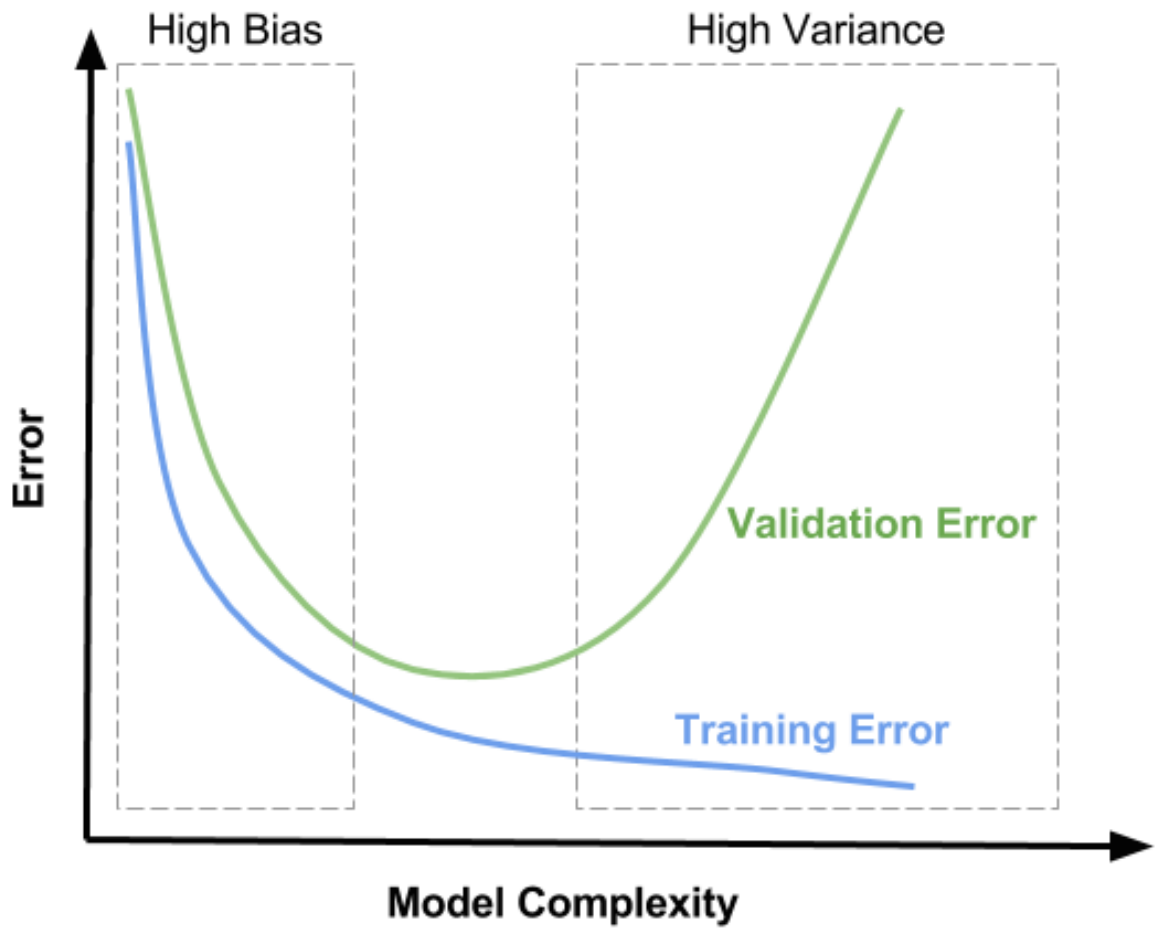
XGBoost resumen entornos ruidosos Test



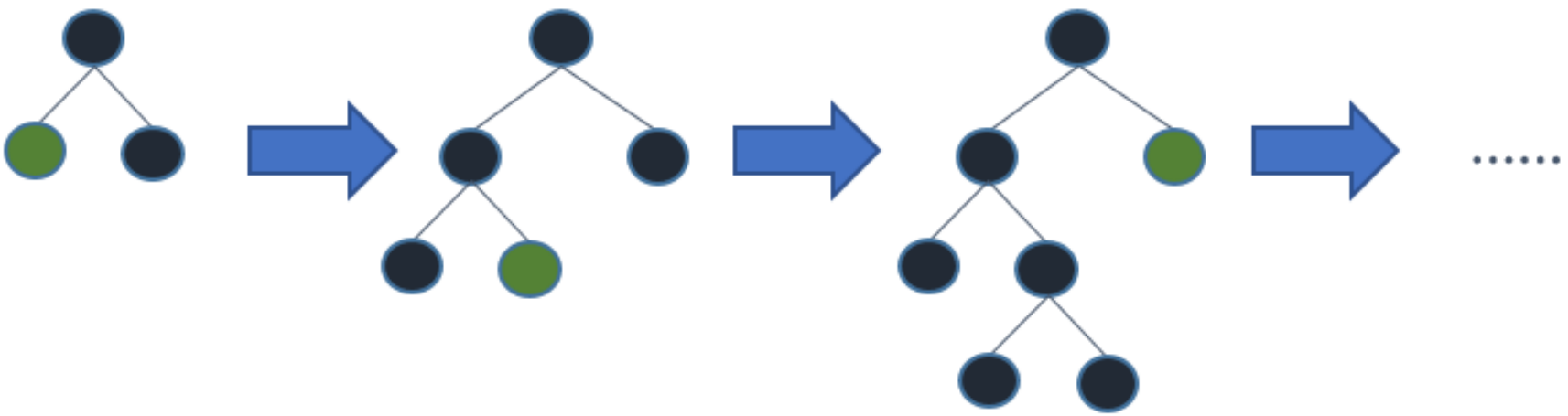
Zoom XGBoost en entornos Ruidosos (Test)



Regularización en Boosting



Regularización en Boosting



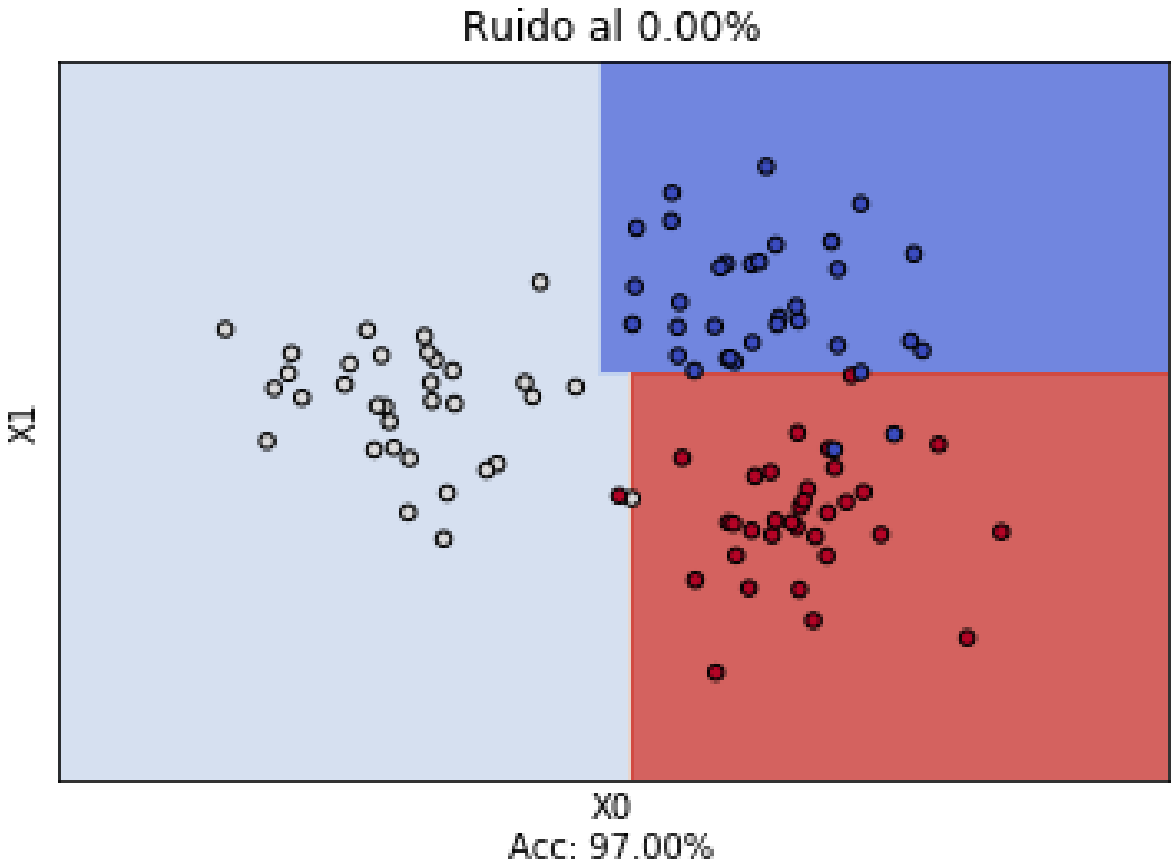
Leaf-wise tree growth

Source: www.analyticsvidhya.com

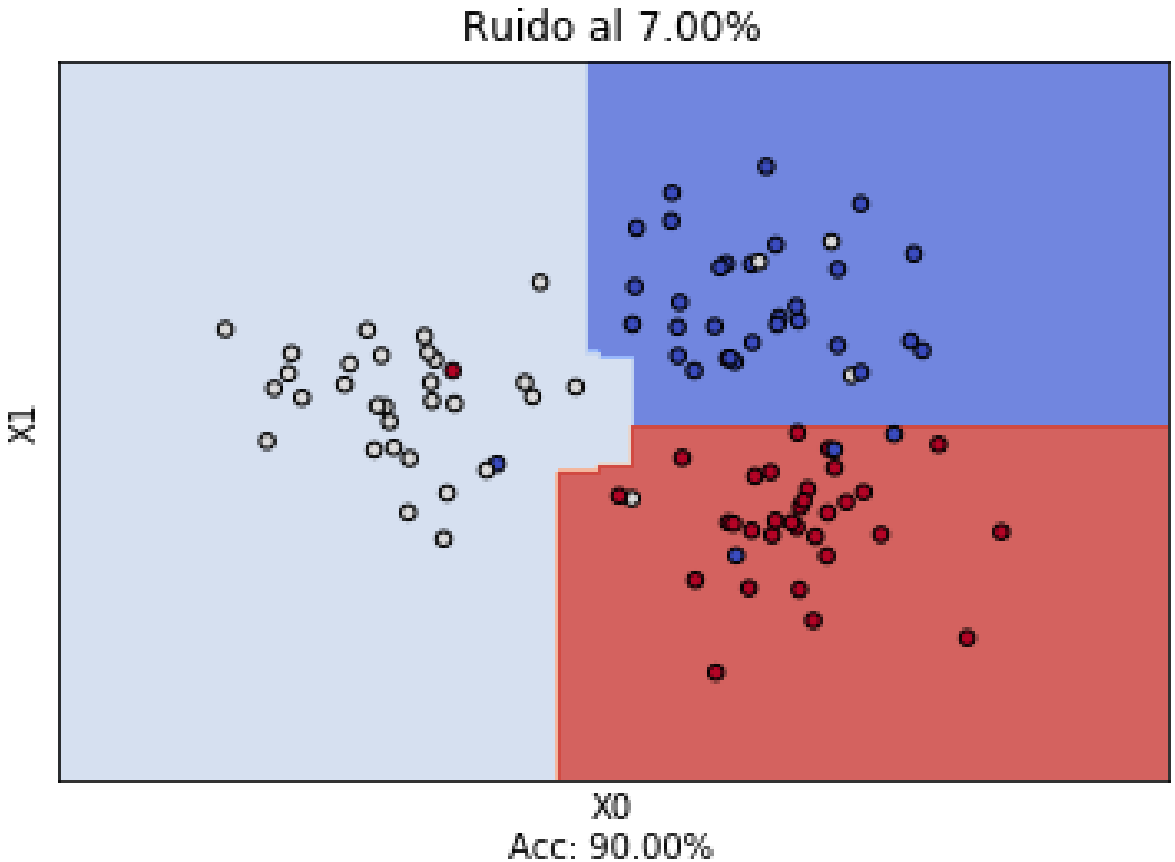
Regularización en Boosting

Parámetros de regularización

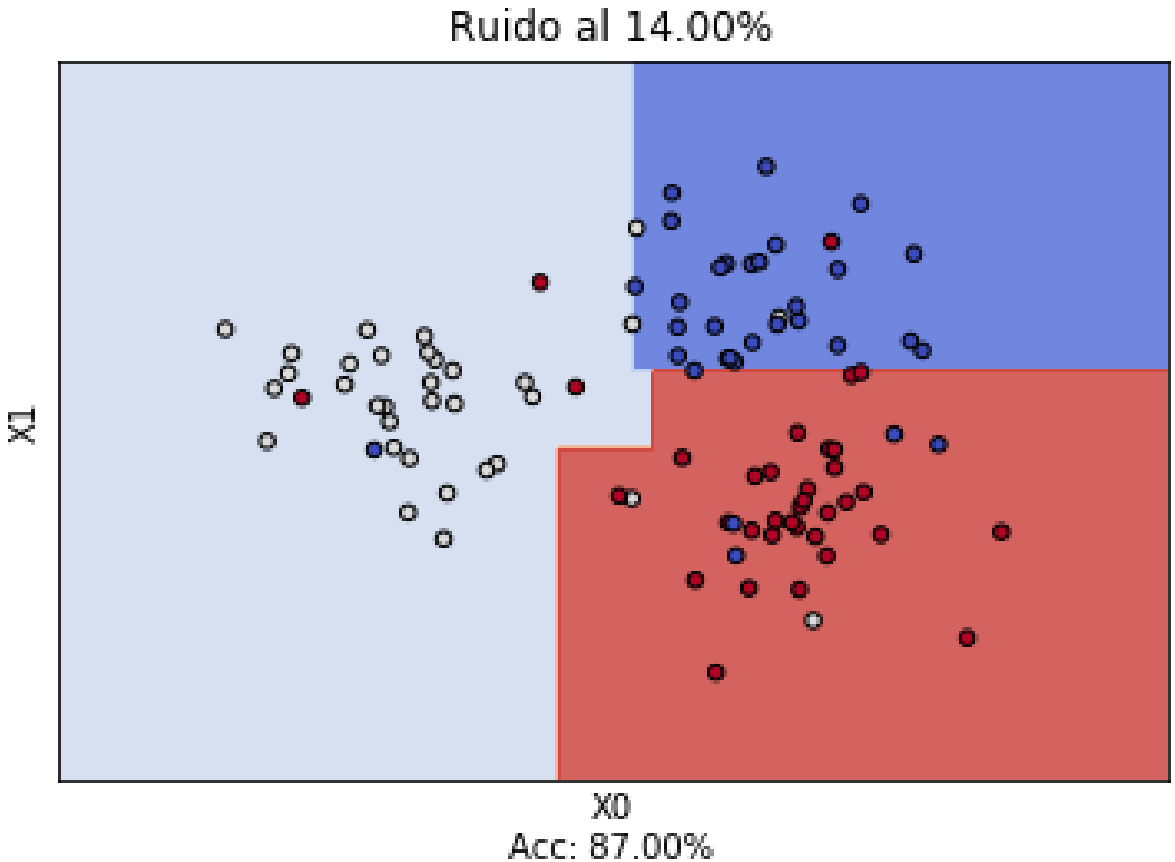
XGBoost regularizado en entornos Ruidosos



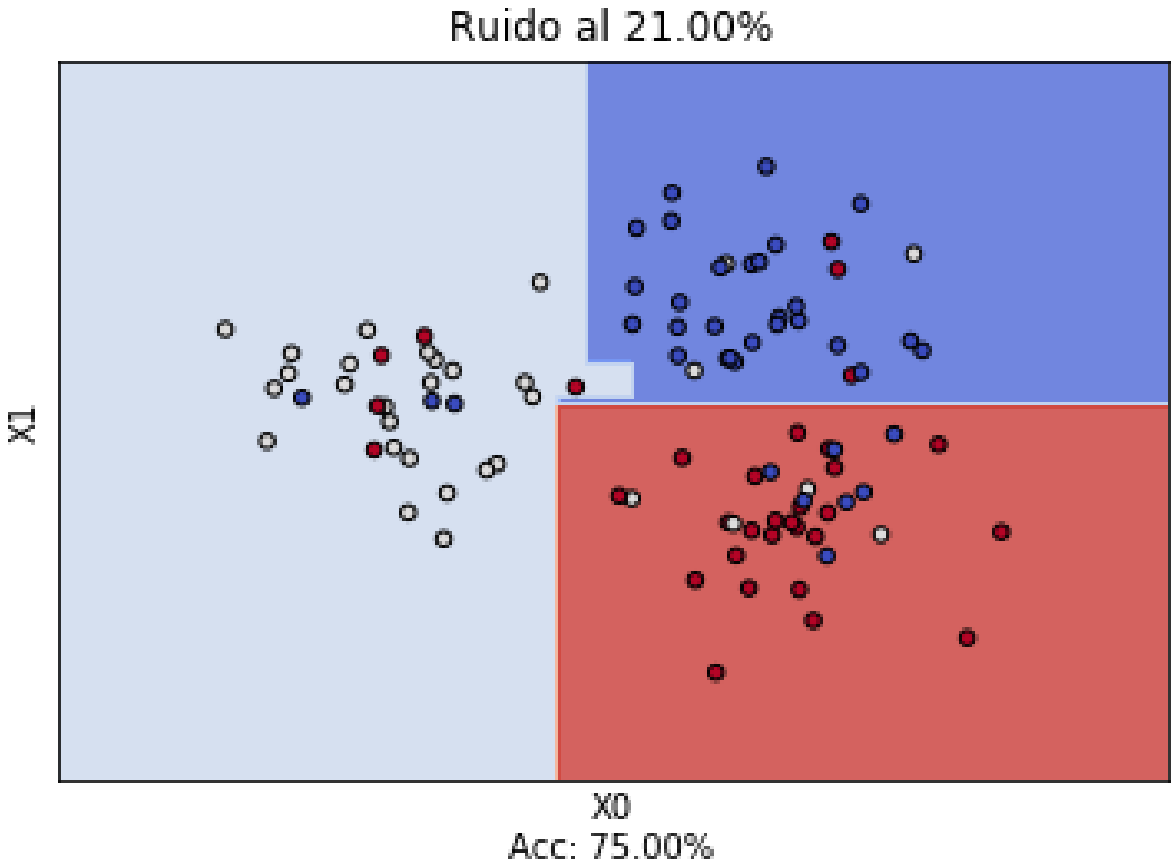
XGBoost regularizado en entornos Ruidosos



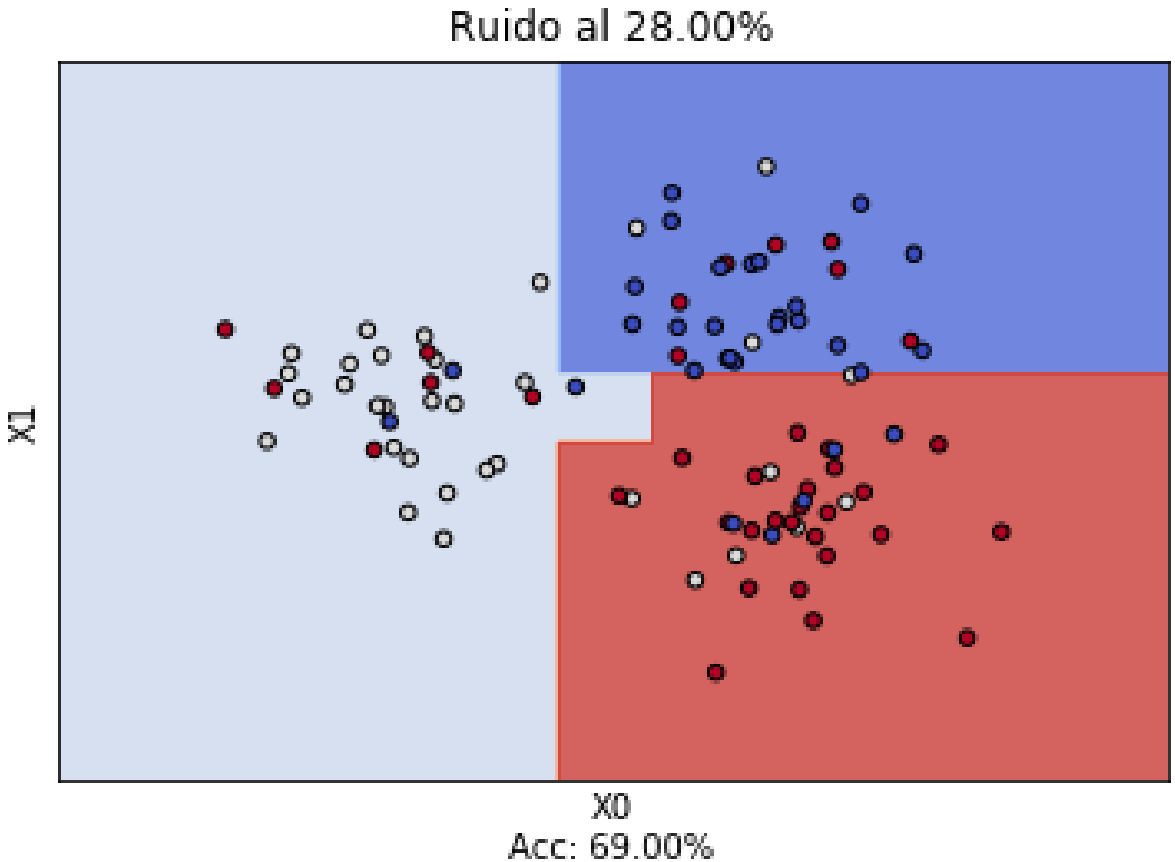
XGBoost regularizado en entornos Ruidosos



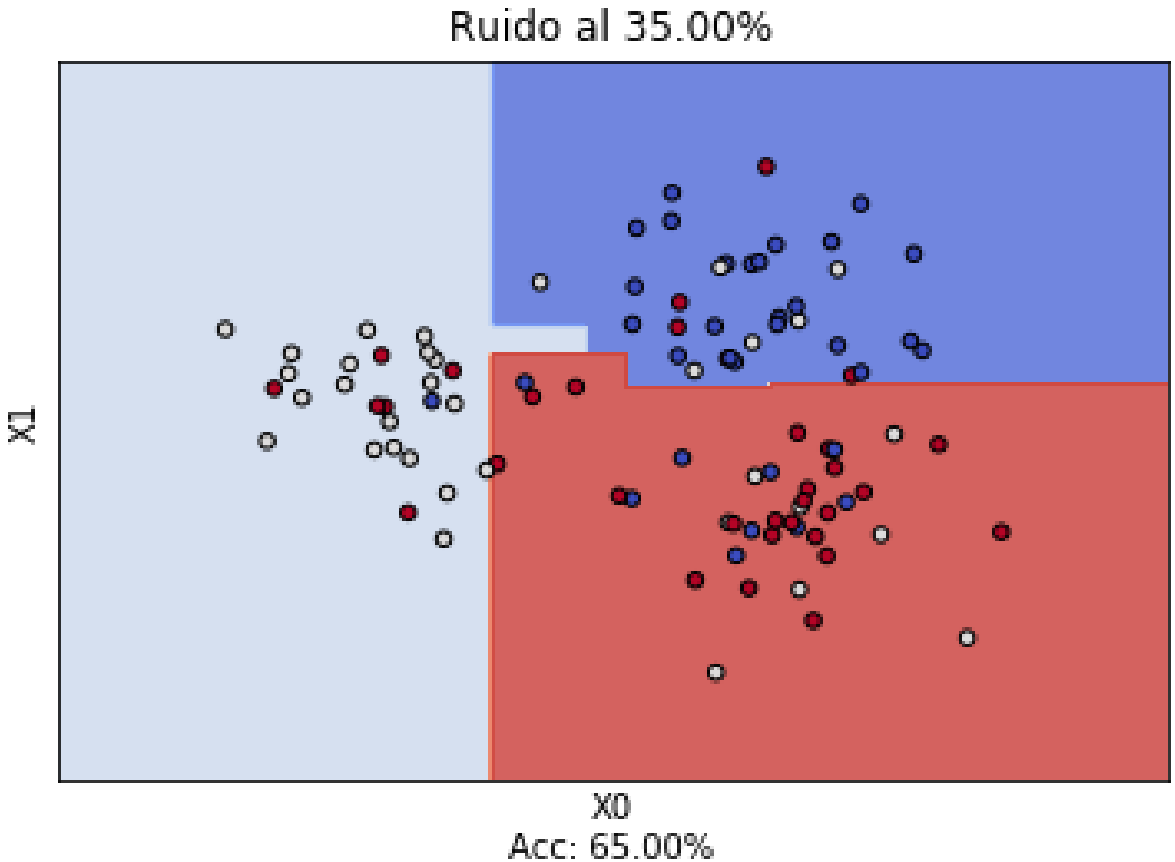
XGBoost regularizado en entornos Ruidosos



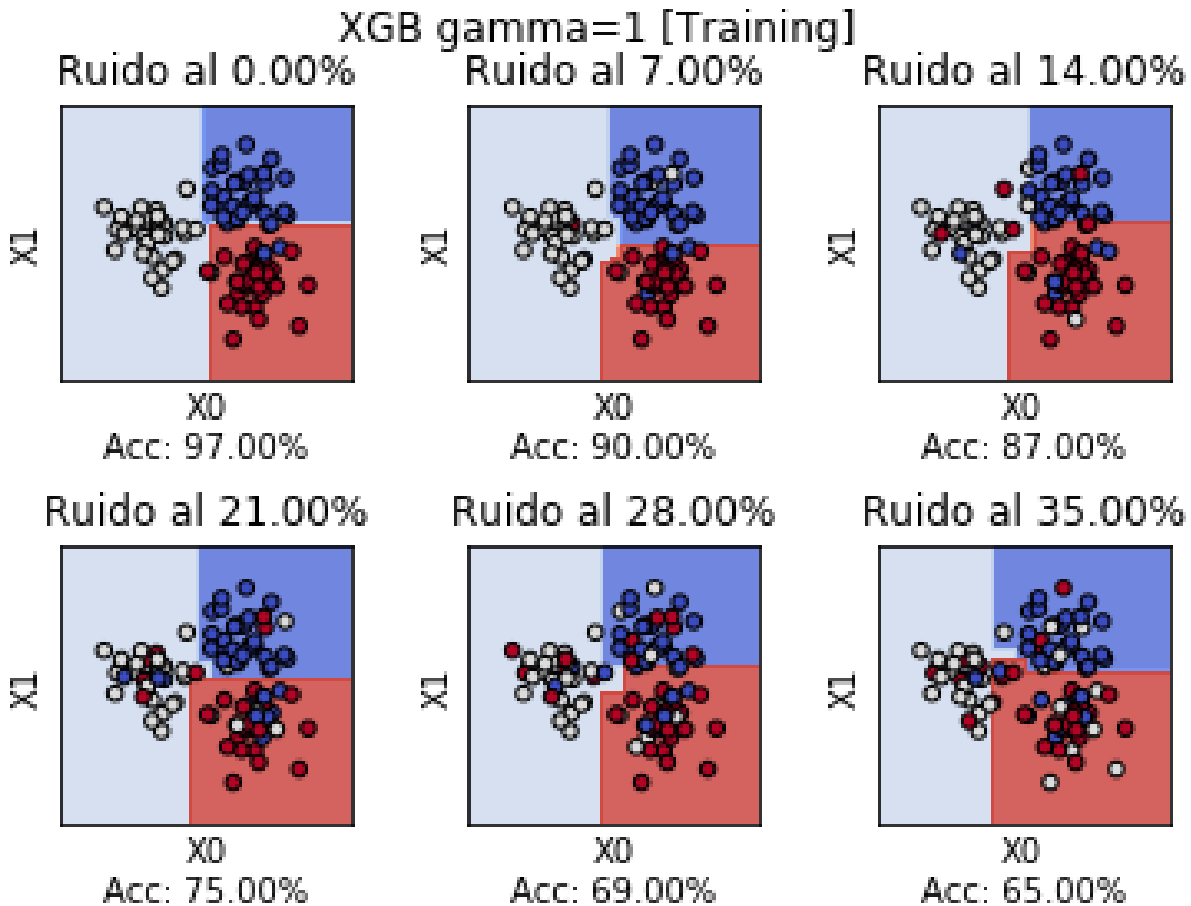
XGBoost regularizado en entornos Ruidosos



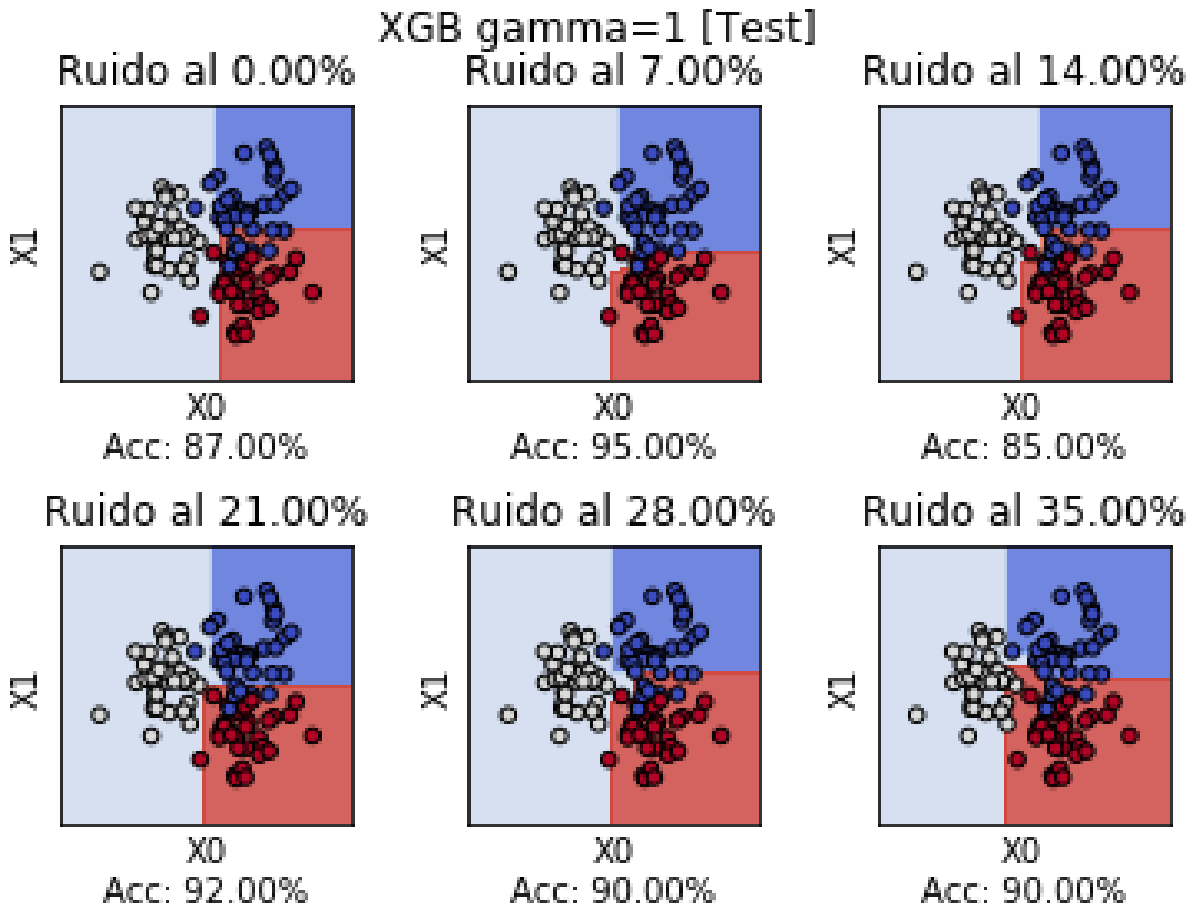
XGBoost regularizado en entornos Ruidosos



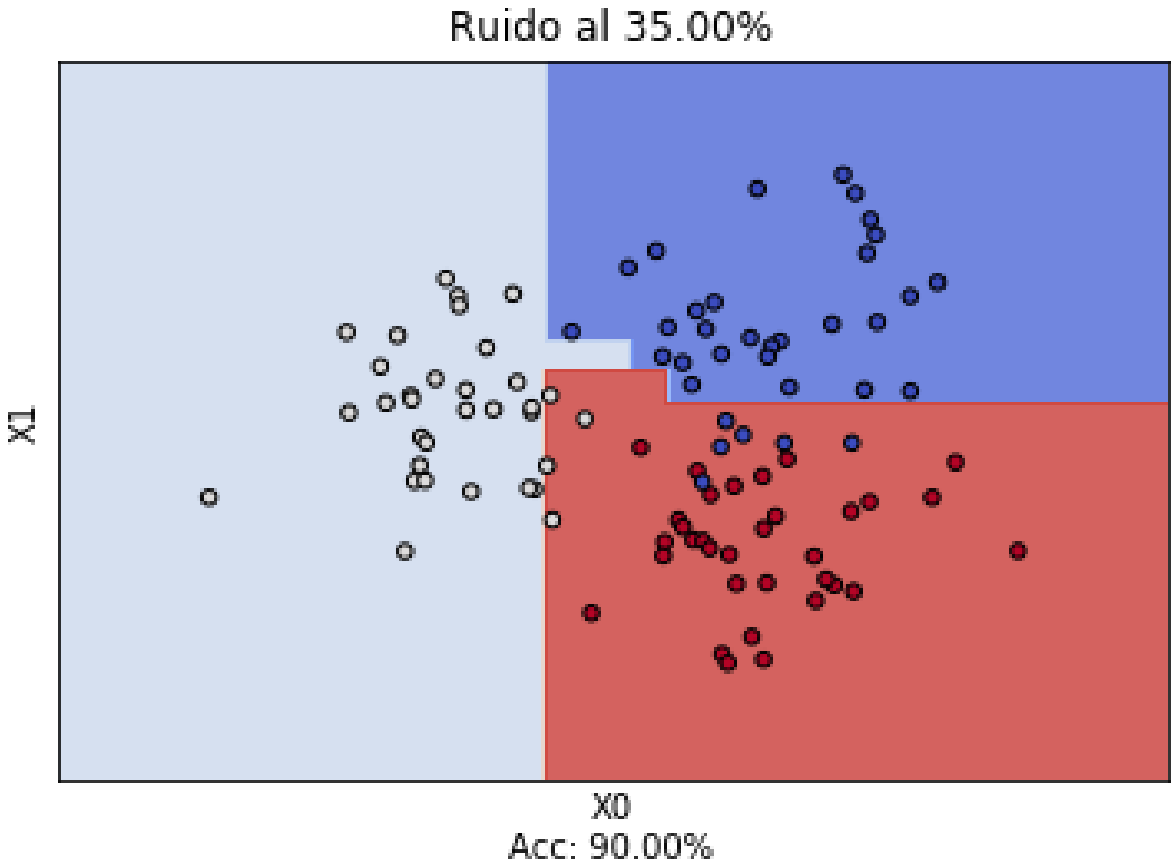
XGBoost regularizado resumen entornos ruidosos



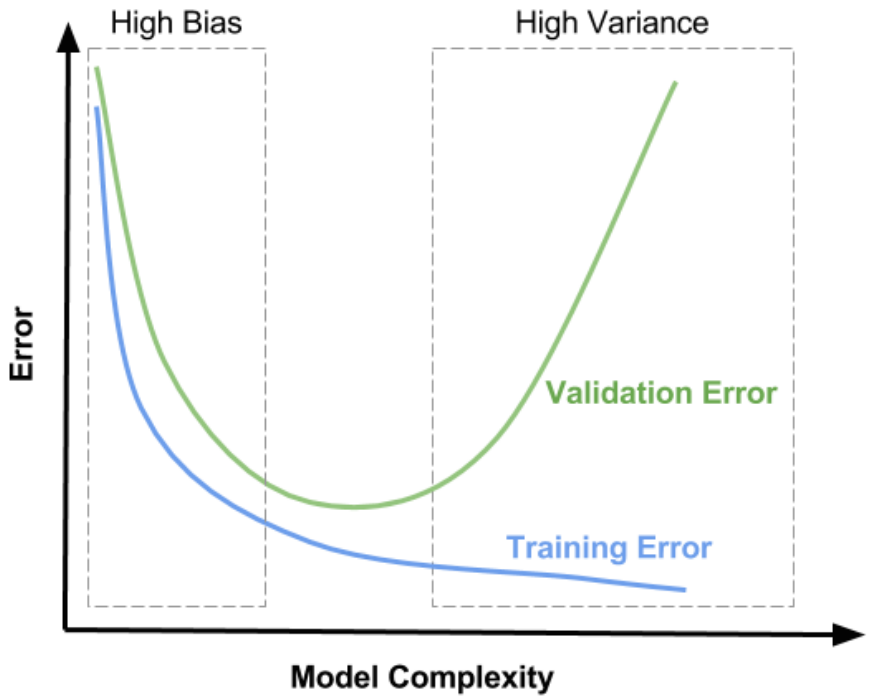
XGBoost regularizado resumen entornos ruidosos (Test)

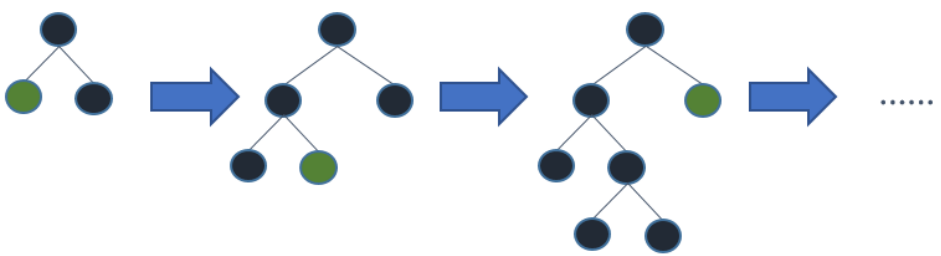


Zoom XGBoost regularizado en entornos Ruidosos Test



Conclusiones





Leaf-wise tree growth

Fin