

# **Distancia de Fermat en Clasificadores de Densidad por Núcleos**

---

Lic. Gonzalo Barrera Borla, Dr. Pablo Groisman - FCEyN, UBA

# **El problema de clasificación**

---

# Definición y vocabulario

[ESL §2.2]

- El *aprendizaje estadístico supervisado* busca estimar (aprender) una variable *respuesta* a partir de cierta(s) variable(s) *predictora(s)*.
- Cuando la *respuesta* es una variable *cualitativa*, el problema de asignar cada observación  $x$  a una clase  $G \in \mathcal{G} = \{g^1, \dots, g^K\}$  se denomina *de clasificación*.
- Un *clasificador* es una función  $\hat{G}(x)$  que para cada observación  $x$ , intenta aproximar su verdadera clase  $g$  por  $\hat{g}$  («ge sombrero»).
- Para construir  $\hat{G}$ , contamos con un *conjunto de entrenamiento* de pares  $(x_i, g_i), i \in \{1, \dots, N\}$  conocidos. Típicamente, las clases serán MECE, y las observaciones  $X \in \mathbb{R}^p$ .

# Clasificador de Bayes

Una posible estrategia de clasificación consiste en asignarle a cada observación  $x_0$ , la clase más probable en ese punto, dada la información disponible.

$$\hat{G}(x) = \arg \max_{g \in \mathcal{G}} \Pr(G = g | X = x) \quad (1)$$

Esta razonable solución es conocida como el *clasificador de Bayes*, y se puede reescribir usando la regla homónima como

$$\begin{aligned} \hat{G}(x) = g_i &\Leftrightarrow \Pr(g_i | X = x) = \max_{g \in \mathcal{G}} \Pr(G = g | X = x) \\ &= \max_{g \in \mathcal{G}} \Pr(X = x | G = g) \times \Pr(G = g) \end{aligned} \quad (2)$$

# Clasificadores «suaves» y «duros»

- Un clasificador que responda «¿qué clase es la que más probablemente contenga esta observación» es un clasificador «duro».
- Un clasificador que además puede responder «¿cuán probable es que esta observación pertenezca a cada clase  $g_j$ ?» es un clasificador «suave».
- La regla de Bayes para clasificación nos puede dar un clasificador duro al maximizar la probabilidad; más aún, también puede construir un clasificador suave:

$$\begin{aligned}\widehat{\Pr}(G = g_i | X = x) &= \frac{\widehat{\Pr}(x|G = g_i) \times \widehat{\Pr}(G = g_i)}{\widehat{\Pr}(X = x)} \\ &= \frac{\widehat{\Pr}(x|G = g_i) \times \widehat{\Pr}(G = g_i)}{\sum_{k \in [K]} \widehat{\Pr}(X = x, G = g_k)}\end{aligned}\tag{3}$$

# **Estimación de Densidad por Nú- cleos**

---

# Clasificador de Bayes empírico

- Si el conjunto de entrenamiento  $\{(x_1, g_1), \dots, (x_N, g_N)\}$  proviene de un muestreo aleatorio uniforme, las probabilidades de clase  $\pi_i = \Pr(G = g^{(i)})$  se pueden aproximar razonablemente por las proporciones muestrales  $\hat{\pi}_i = \#\{g_j : g_j = g^{(i)}\}/N$
- Resta hallar una aproximación  $\Pr(x|G = g)$  para cada clase, ya sea a través de una función de densidad, de distribución, u otra manera.

# Estimación unidimensional

[ESL §6.6, Parzen 1962]

Para fijar ideas, asumamos que  $X \in \mathbb{R}$  y consideremos la estimación de densidad en una única clase para la que contamos con  $N$  ejemplos  $\{x_1, \dots, x_N\}$ . Una aproximación  $\hat{f}$  directa sería (1)

$$\hat{f}(x_0) = \frac{\#\{x_i \in \mathcal{N}(x_0)\}}{N \times h} \quad (4)$$

donde  $\mathcal{N}$  es un vecindario métrico de  $x_0$  de diámetro  $h$ . Esta estimación es irregular, con saltos discretos en el numerador, por lo que se prefiere el estimador suavizado por núcleos de Parzen-Rosenblatt

$$\hat{f}(x_0) = \frac{1}{N} \sum_{i=1}^N K(x_0, x_i) \quad (5)$$

# Función núcleo o «kernel»

Se dice que  $K(x) : \mathbb{R} \rightarrow \mathbb{R}$  es una *función núcleo* si

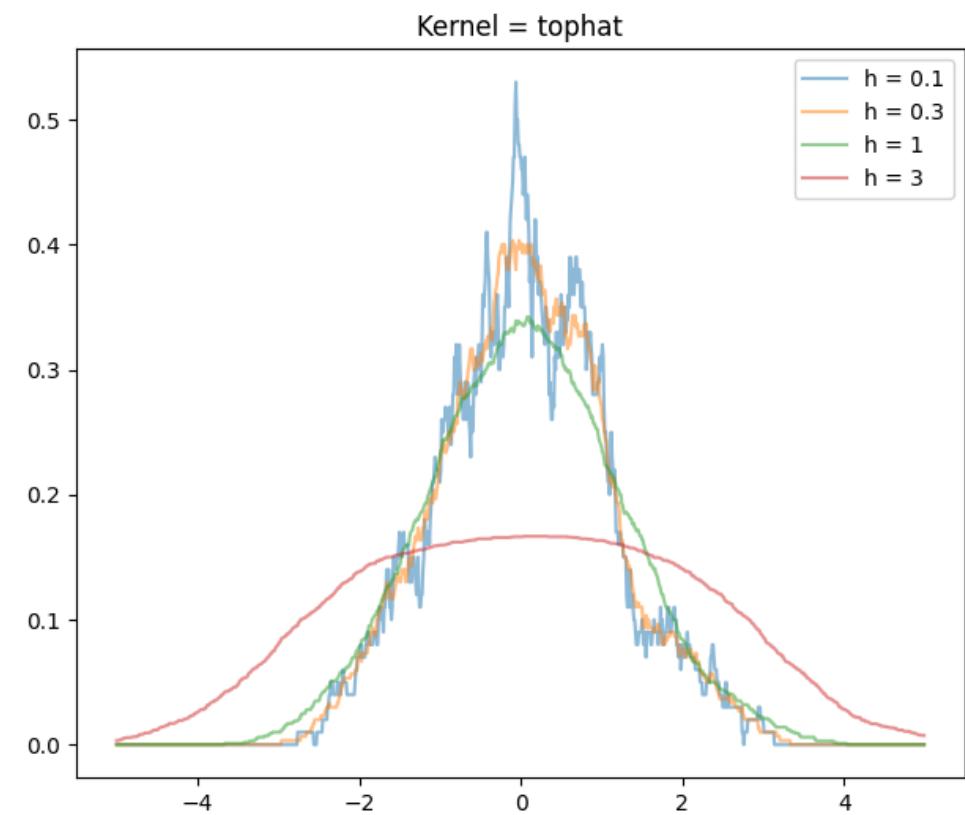
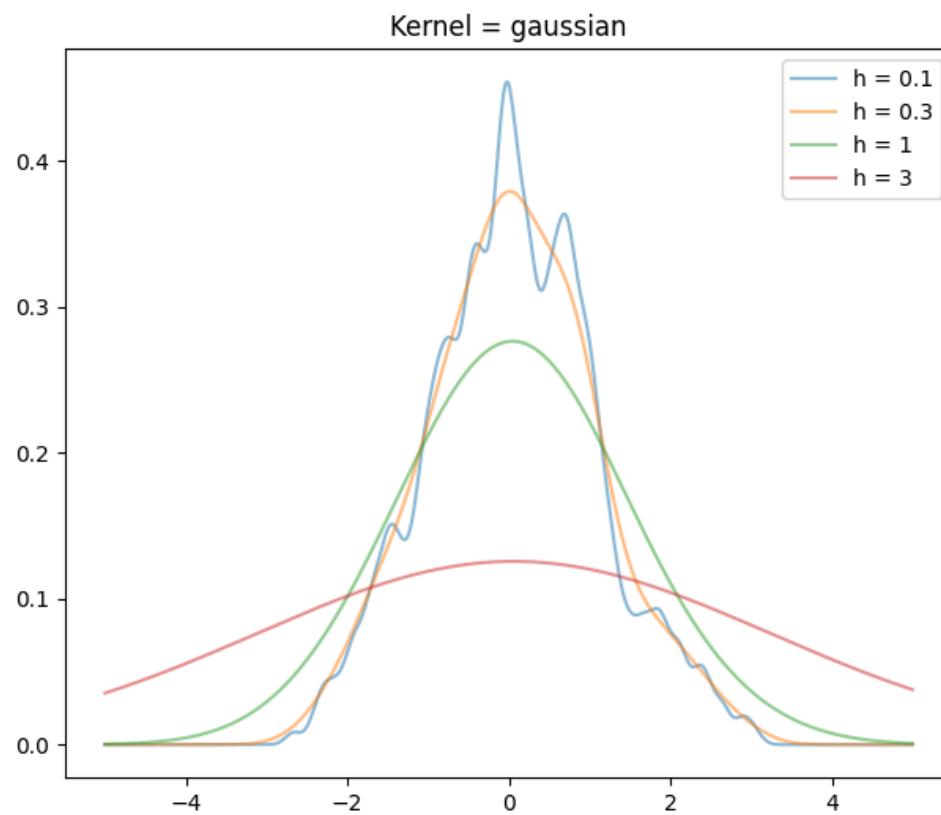
- toma valores reales no negativos:  $K(u) \geq 0 \forall u \in \text{sop } K$ ,
- está normalizada:  $\int_{-\infty}^{+\infty} K(u)du = 1$ ,
- es simétrica:  $K(u) = K(-u)$  y
- alcanza su máximo en el centro:  $\max_u K(u) = K(0)$

Observación 1: Todas las funciones de densidad simétricas centradas en 0 son núcleos; en particular, la densidad «normal estándar»  $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$  lo es.

Observación 2: Si  $K(u)$  es un núcleo, entonces  $K_h(u) = \frac{1}{h} K\left(\frac{u}{h}\right)$  también lo es.

Observación 3: Si  $\mathbb{1}(\cdot)$  es la función indicadora, resulta que  $U_h(x) = \frac{1}{h} \mathbb{1}\left(-\frac{h}{2} < x < \frac{h}{2}\right)$  es un núcleo válido, y el estimador de Ecuación 5 con núcleo  $U_h$  devuelve el estimador Ecuación 4

# Núcleo uniforme



# Clasificador de densidad por núcleos

[ESL §6.6.2]

Si  $\hat{f}_k, k \in 1, \dots, K$  son estimadores de densidad por núcleos<sup>1</sup> según Ecuación 5, la regla de Bayes nos provee un clasificador suave

$$\begin{aligned}\widehat{\Pr}(G = g_i | X = x) &= \frac{\widehat{\Pr}(x|G = g_i) \times \widehat{\Pr}(G = g_i)}{\widehat{\Pr}(X = x)} \\ &= \frac{\hat{\pi}_i \hat{f}_i(x)}{\sum_{k=1}^K \hat{\pi}_k \hat{f}_k(x)}\end{aligned}\tag{6}$$

---

<sup>1</sup>KDEs ó *Kernel Density Estimators*, por sus siglas en inglés

## Interludio: Naive Bayes

[ESL §6.6.3]

¿Y si las  $X$  son multivariadas ( $X \in \mathbb{R}^d, d \geq 2$ )? ¿Se puede adaptar el clasificador?

Sí, pero es complejo. Un camino sencillo: asumir que condicional a cada clase  $G = j$ , los predictores  $X_1, X_2, \dots, X_p$  se distribuyen independientemente entre sí.

$$f_j(X) = \prod_{i=1}^p f_{j,i}(X_i) \tag{7}$$

Cada densidad marginal  $f_{j,i}$  condicional a la clase se puede estimar usando KDE univariado, y hasta se puede aplicar - usando histogramas - cuando algunas componentes  $X_i$  son discretas.

A este procedimiento, se lo conoce como «Naive Bayes».

# KDE multivariado

[Wand & Jones 1995 §4]

En su forma más general, estimador de densidad por núcleos  $d$ -variado es

$$\hat{f}(x; \mathbf{H}) = N^{-1} \sum_{i=1}^N K_{\mathbf{H}}(x - x_i) \quad (8)$$

donde

- $\mathbf{H} \in \mathbb{R}^{d \times d}$  es una matriz simétrica def. pos. análoga a la ventana  $h \in \mathbb{R}$  para  $d = 1$ ,
- $K_{\mathbf{H}}(t) = |\det \mathbf{H}|^{-\frac{1}{2}} K(\mathbf{H}^{-\frac{1}{2}} t)$
- $K$  es una función núcleo  $d$ -variada tal que  $\int K(\mathbf{x}) d\mathbf{x} = 1$

Típicamente,  $K$  es la densidad normal multivariada

$$\Phi(x) : \mathbb{R}^d \rightarrow \mathbb{R} = (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{\|x\|^2}{2}\right) \quad (9)$$

# Dificultades: elección de $\mathbf{H}$

Sean las clases de matrices pertenecientes a  $\mathbb{R}^{d \times d}$  ...

- $\mathcal{F}$ , de matrices simétricas definidas positivas,
- $\mathcal{D}$ , de matrices diagonales definidas positivas ( $\mathcal{D} \subseteq \mathcal{F}$ ) y
- $\mathcal{S}$ , de múltiplos escalares de la identidad:  $\mathcal{S} = \{h^2\mathbf{I} : h > 0\} \subseteq \mathcal{D}$

Aún tomando una única  $\mathbf{H}$  para toda la muestra,  $\mathbf{H} \in \dots$

- $\mathcal{F}$ , requiere definir  $\binom{d}{2} = \frac{d(d-1)}{2}$  parámetros de ventana,
- $\mathcal{D}$  requiere  $d$  parámetros, y
- $\mathcal{S}$  tiene un único parámetro  $h$ .

A priori no es posible saber qué parametrización conviene, pero en general  $\mathbf{H} \in \mathcal{D}$  parece un compromiso razonable: no se pierde demasiado contra  $\mathcal{F}$ , pero tampoco se padece la «rigidez» de  $\mathbf{H} \in \mathcal{S}$ .

# Dificultades: La maldición de la dimensionalidad

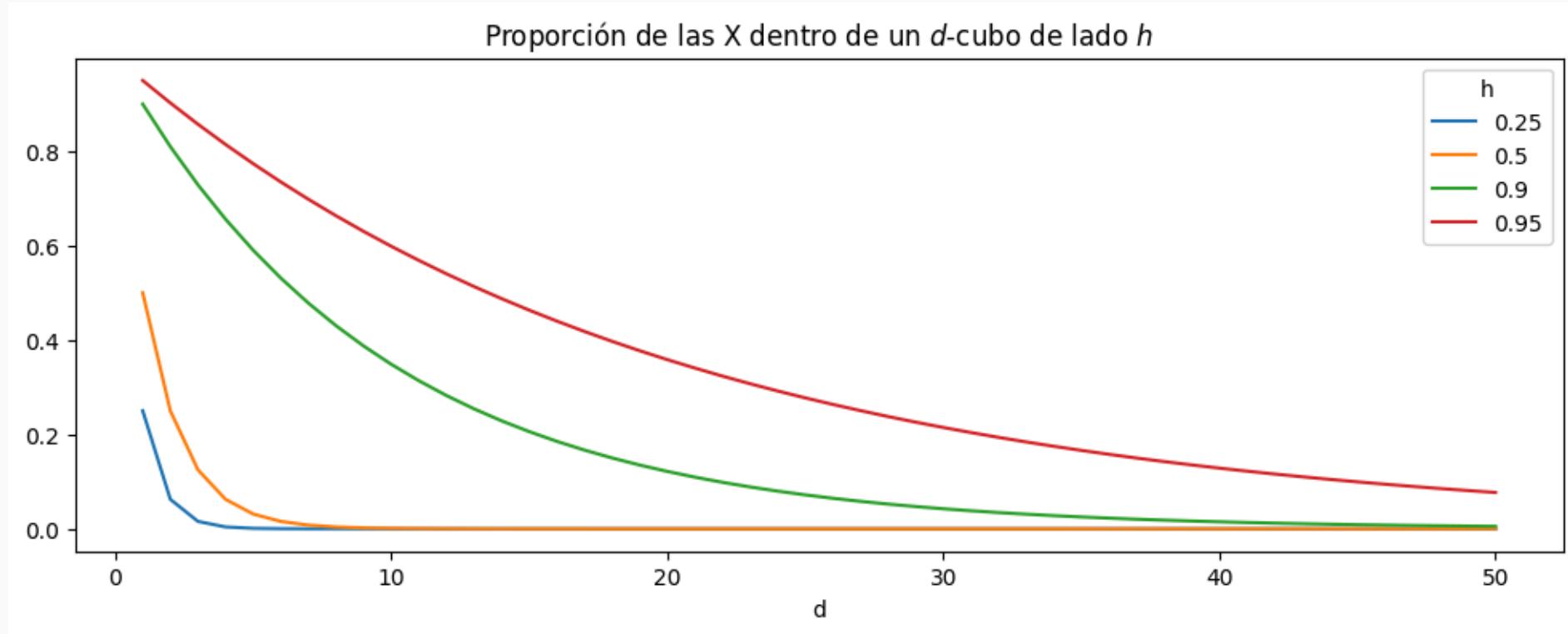
[ESL §2.5, Wand & Jones 1995 §4.9 ej 4.1]

Sean  $X_i \stackrel{\text{iid}}{\sim} \text{Uniforme}([-1, 1]^d), i \in \{1, \dots, N\}$ , y consideremos la estimación de la densidad en el origen,  $f(\mathbf{0})$ . Suponga que el núcleo  $K_{\mathbf{H}}$  es un «núcleo producto» basado en la distribución univariada Uniforme(-1, 1), y  $\mathbf{H} = h^2 \mathbf{I}$ . Derive una expresión para la proporción esperada de puntos incluidos dentro del soporte del núcleo  $K_{\mathbf{H}}$  para  $h, d$  arbitrarios.

(... interludio de pizarrón ...)

$$\begin{aligned}\Pr(X \in [-h, h]^d) &= \Pr(\cap_{i=1}^d |X_i| \leq h) \\ \Pr(X \in [-0.95, 0.95]^{50}) &\approx 0.0077\end{aligned}\tag{10}$$

# Dificultades: La maldición de la dimensionalidad



Para  $h \leq 0.5$ ,  $\Pr(\cdot) < 1 \times 10^{-15}$ . Aún para  $h = 0.95$ ,  $\Pr(\cdot) \approx 0.0077$  😱

## **Clasificación en variedades**

---

# La hipótesis de la variedad («manifold hypothesis»)

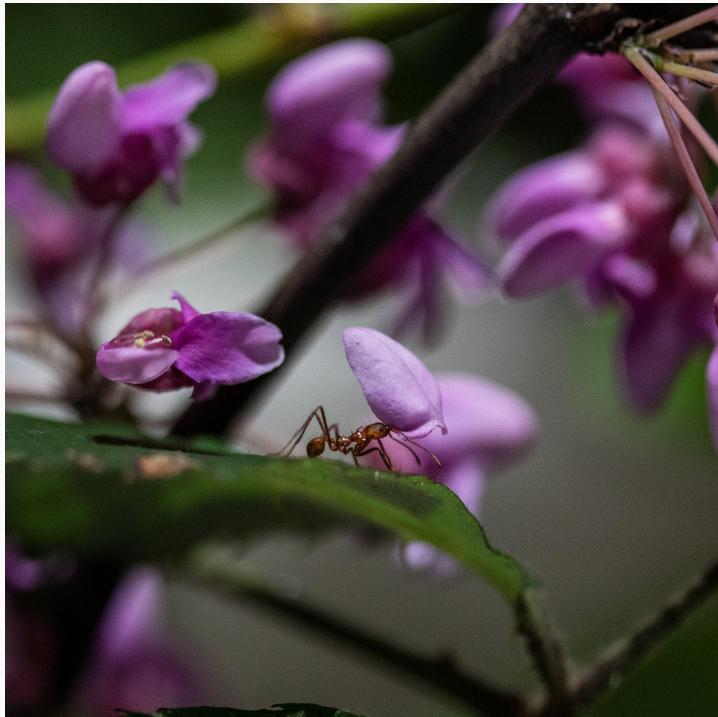
[Bengio Repr learning] [[Bengio en Reddit](#) ]

La hipótesis de la variedad postula que los datos  $X \in \mathbb{R}^{d_X}$  muestreados soportados en un espacio de alta dimensionalidad<sup>1</sup>. tenderán a concentrarse sobre una *variedad*  $\mathcal{M}$ , potencialmente de mucha menor dimensión  $d_{\mathcal{M}} \ll d_X$ , embebida en el espacio original  $\mathcal{M} \subseteq \mathbb{R}^{d_X}$ .

- Well suited for AI tasks such as those involving images, sounds or text, for which most uniformly sampled input configurations are unlike natural stimuli.
- archetypal manifold modeling algorithm is, not surprisingly, also the archetypal low dimensional representation learning algorithm: Principal Component Analysis, which models a linear manifold.
- Data manifold for complex real world domains are however expected to be strongly nonlinear.

---

<sup>1</sup>E.g.: imágenes, audio, video, secuencias de nucleótidos



Pero: ¿en qué variedad vive un dígito, o su trazo, o una canción? 🎵

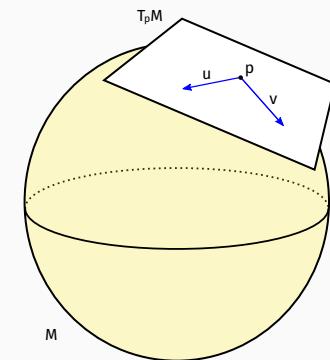
# Interludio: Variedades de Riemann [Wikipedia]

Una variedad  $d$ -dimensional  $\mathcal{M}$  es un espacio topológico tal que cada punto  $p \in \mathcal{M}$  tiene un vecindario  $U$  que resulta homeomórfico a un conjunto abierto en  $\mathbb{R}^d$

- topológico: se puede definir cercanía (pero no necesariamente distancia), permite definir funciones continuas y límites
- homeomórfico a  $\mathbb{R}^d$ : para cada punto  $p \in \mathcal{M}$ , existe un mapa biyectivo y suave entre el vecindario de  $p$  y  $\mathbb{R}^d$ . El conjunto de tales mapas se denomina *atlas*.

Sea  $T_p\mathcal{M}$  el espacio tangente a un punto  $p \in \mathcal{M}$ , y  $g_p : T_p\mathcal{M} \times T_p\mathcal{M} \rightarrow \mathbb{R}$  una forma bilinear pos. def. para cada  $p$  que induce una norma  $\|v\|_p = \sqrt{g_p(v, v)}$ .

Decimos entonces que  $g_p$  es una métrica Riemanniana y el par  $(\mathcal{M}, g)$  es una variedad de Riemann, donde las nociones de *distancia*, *ángulo* y *geodésica* están bien definidas.



# KDE en variedades de Riemann [Pelletier 2005]

- Sea  $(\mathcal{M}, g)$  una variedad de Riemann compacta y sin frontera de dimensión  $d$ , y usemos  $d_g$  para denotar la distancia de Riemann.
- Sea  $K$  un *núcleo isotrópico en  $\mathcal{M}$  soportado en la bola unitaria* (cf. cond. (i)-(v))
- Sean  $p, q \in \mathcal{M}$ , y  $\theta_p(q)$  la *función de densidad de volumen en  $\mathcal{M}$* <sup>1</sup>

Luego, el estimador de densidad para  $X_i \stackrel{\text{iid}}{\sim} f$  es  $f_{N,K} : \mathcal{M} \rightarrow \mathbb{R}$  que a cada  $p \in \mathcal{M}$  le asocia el valor

$$f_{N,K}(p) = N^{-1} \sum_{i=1}^N K_h(p, X_i) = N^{-1} \sum_{i=1}^N \frac{1}{h^d} \frac{1}{\theta_{X_i}(p)} K\left(\frac{d_g(p, X_i)}{h}\right) \quad (11)$$

con la restricción de que la ventana  $h \leq h_0 \leq \text{inj}(\mathcal{M})$ , el *radio de inyectividad* de  $\mathcal{M}$ <sup>2</sup>

---

<sup>1</sup>;Ardua definición! Algo así como el cociente entre las medida de volumen en  $\mathcal{M}$ , y su transformación via el mapa local a  $\mathbb{R}^d$

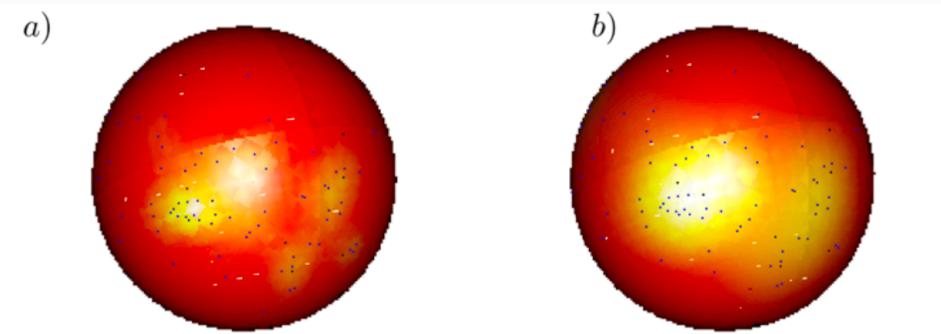
<sup>2</sup>el ínfimo entre el supremo del radio de una bola en cada  $p$  tal que su mapa es un difeomorfismo

# Interludio: densidad de volumen en la esfera [Henry y Rodríguez, 2009]

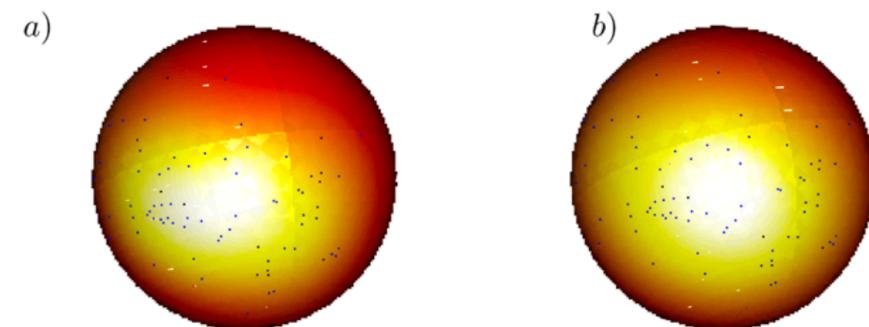
En «*Kernel Density Estimation on Riemannian Manifolds: Asymptotic Results*» (2009), Guillermo Henry y Daniela Rodriguez estudian algunas propiedades asintótica de este estimador, y las ejemplifican con datos de sitios volcánicos en la superficie terrestre. En particular, calculan la densidad de volumen  $\theta_{p(q)}$

$$\theta_p(q) = R \frac{|\operatorname{sen}(d_g(p, q)/R)|}{d_g(p, q)} \quad \text{if } q \neq p, -p \quad \text{and}$$

$$\theta_p(p) = 1.$$



**Fig. 1** The nonparametric density estimator using different bandwidth, **a**  $h = 1500$  and **b**  $h = 3000$



**Fig. 2** The nonparametric density estimator using different bandwidth, **a**  $h = 5000$  and **b**  $h = 7000$

# Clasificación en variedades [Loubes y Pelletier 2008]

¡Clasificador de Bayes + KDE en Variedades = Clasificación (suave o dura) en variedades!

Plantean una regla de clasificación  $\hat{G}$  para 2 clases adaptable a K clases de forma directa. Sea  $p \in \mathcal{M}$  una variedad riemanniana como antes, y  $\{(x_1, g_1), \dots, (x_N, g_N)\}$  nuestras observaciones y sus clases. Luego,

$$\hat{G}(p) = \arg \max_{g \in \mathcal{G}} \sum_{i=1}^N \mathbb{1}(g_i = g) K_h(p, X_i) \quad (12)$$

# Clasificación en variedades [Loubes y Pelletier 2008]

¡Clasificador de Bayes + KDE en Variedades = Clasificación (suave o dura) en variedades!

Plantean una regla de clasificación  $\hat{G}$  para 2 clases adaptable a K clases de forma directa. Sea  $p \in \mathcal{M}$  una variedad riemanniana como antes, y  $\{(x_1, g_1), \dots, (x_N, g_N)\}$  nuestras observaciones y sus clases. Luego,

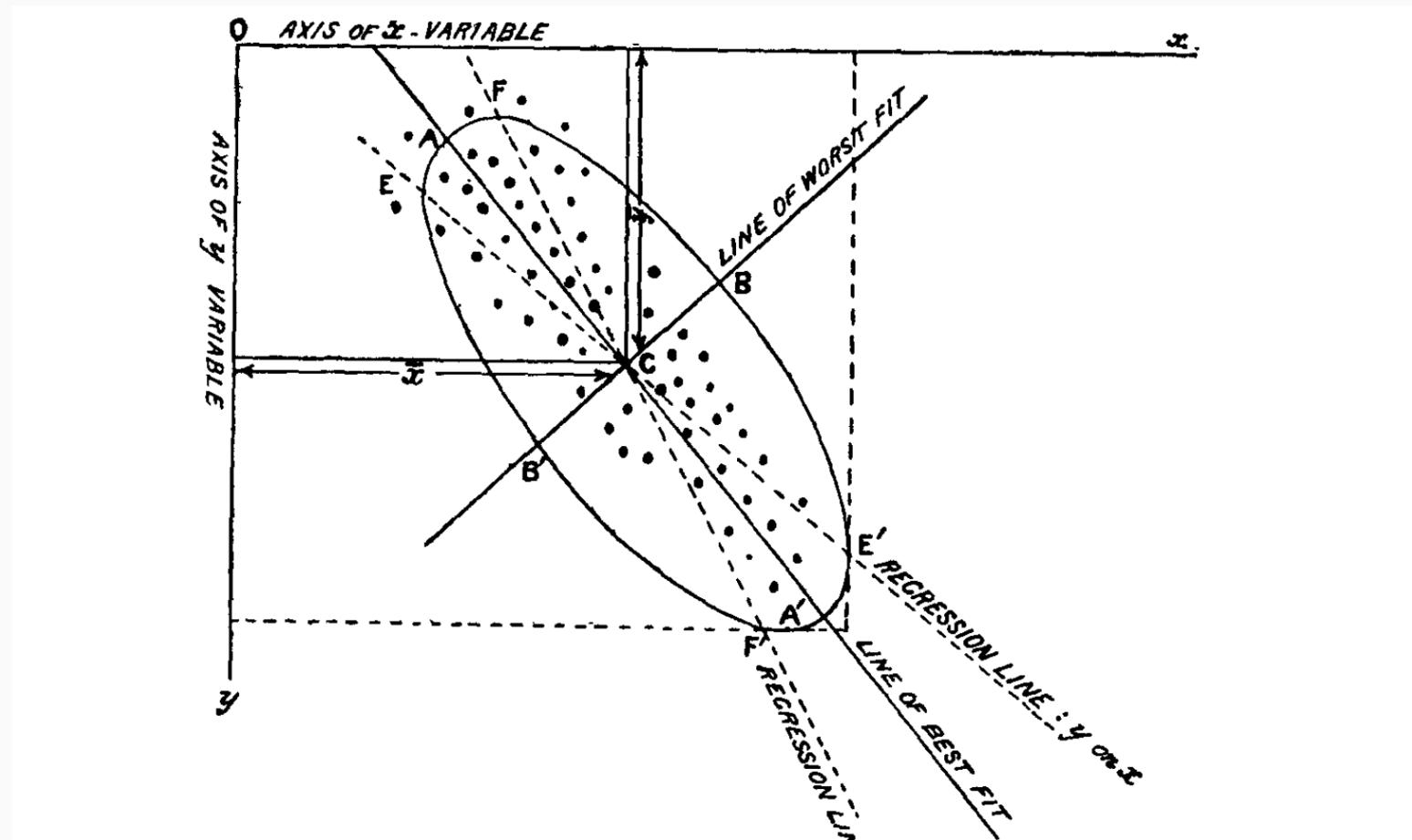
$$\hat{G}(p) = \arg \max_{g \in \mathcal{G}} \sum_{i=1}^N \mathbb{1}(g_i = g) K_h(p, X_i) \quad (12)$$

Pero... ¿y si la variedad es desconocida?

# Aprendizaje de distancias

---

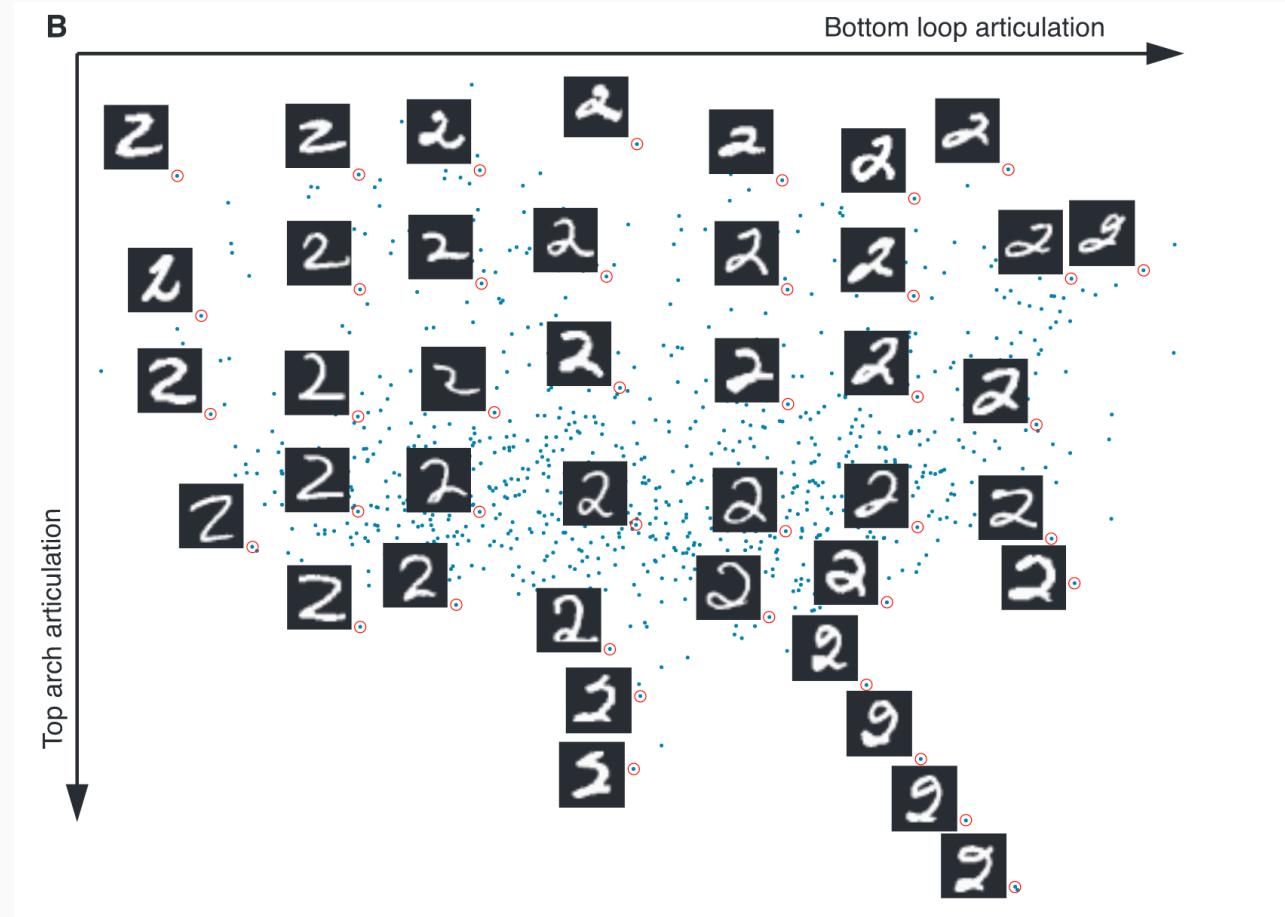
# El ejemplo canónica: Análisis de Componentes Principales (PCA)



Karl Pearson (1901), «LIII. On lines and planes of closest fit to systems of points in space.»

# El algoritmo más cool: Isomap

1. Construya el grafo de  $k, \varepsilon$ -vecinos,  $\text{NN} = (\mathbf{X}, E)$
2. Compute los caminos mínimos - las geodésicas entre observaciones,  $d_{\text{NN}}(x, y)$ .
3. Construya una representación («embedding»)  $d^*$ -dimensional que minimice la discrepancia («stress») entre  $d_{\text{NN}}$  y la distancia euclídea en  $\mathbb{R}^{d^*}$



Tenenbaum et al (2000), «A Global Geometric Framework for Nonlinear Dimensionality Reduction»

We tackle the problem of learning a distance between points, able to capture both the geometry of the manifold and the underlying density. We define such a sample distance and prove the convergence, as the sample size goes to infinity, to a macroscopic one that we call Fermat distance as it minimizes a path functional, resembling Fermat principle in optics.

— P. Groisman et al (2019)

Sea  $f$  una función continua y positiva,  $\beta \geq 0$  y  $x, y \in S \subseteq \mathbb{R}^d$ . Definimos la *Distancia de Fermat*  $\mathcal{D}_{f,\beta}(x, y)$  como:

$$\mathcal{T}_{f,\beta}(\gamma) = \int_{\gamma} f^{-\beta}, \quad \mathcal{D}_{f,\beta}(x, y) = \inf_{\gamma \in \Gamma} \mathcal{T}_{f,\beta}(\gamma) \quad (13)$$

We tackle the problem of learning a distance between points, able to capture both the geometry of the manifold and the underlying density. We define such a sample distance and prove the convergence, as the sample size goes to infinity, to a macroscopic one that we call Fermat distance as it minimizes a path functional, resembling Fermat principle in optics.

— P. Groisman et al (2019)

Sea  $f$  una función continua y positiva,  $\beta \geq 0$  y  $x, y \in S \subseteq \mathbb{R}^d$ . Definimos la *Distancia de Fermat*  $\mathcal{D}_{f,\beta}(x, y)$  como:

$$\mathcal{T}_{f,\beta}(\gamma) = \int_{\gamma} f^{-\beta}, \quad \mathcal{D}_{f,\beta}(x, y) = \inf_{\gamma \in \Gamma} \mathcal{T}_{f,\beta}(\gamma) \quad \text{😱} \quad (13)$$

We tackle the problem of learning a distance between points, able to capture both the geometry of the manifold and the underlying density. We define such a sample distance and prove the convergence, as the sample size goes to infinity, to a macroscopic one that we call Fermat distance as it minimizes a path functional, resembling Fermat principle in optics.

— P. Groisman et al (2019)

Sea  $f$  una función continua y positiva,  $\beta \geq 0$  y  $x, y \in S \subseteq \mathbb{R}^d$ . Definimos la *Distancia de Fermat*  $\mathcal{D}_{f,\beta}(x, y)$  como:

$$\mathcal{T}_{f,\beta}(\gamma) = \int_{\gamma} f^{-\beta}, \quad \mathcal{D}_{f,\beta}(x, y) = \inf_{\gamma \in \Gamma} \mathcal{T}_{f,\beta}(\gamma) \quad \text{😱} \quad (13)$$

... donde el ínfimo se toma sobre el conjunto  $\Gamma$  de todos los caminos rectificables entre  $x$  e  $y$  contenidos en  $\bar{S}$ , la clausura de  $S$ , y la integral es entendida con respecto a la longitud de arco dada por la distancia euclídea.

# Distancia de Fermat muestral

Para  $\alpha \geq 1$  y  $x, y \in \mathbb{R}^d$ , la *Distancia Muestral de Fermat* se define como

$$D_{\mathbf{X}, \alpha} = \inf \left\{ \sum_{j=1}^{K-1} \|q_{j+1} - q_j\|^\alpha : (q_1, \dots, q_K) \text{ es un camino de } x \text{ a } y, K \geq 1 \right\} \quad (14)$$

donde los  $q_j$  son elementos de la muestra  $\mathbf{X}$ . Nótese que  $D_{\mathbf{X}, \alpha}$  satisface la desigualdad triangular, define una métrica sobre  $\mathbf{X}$  y una pseudo-métrica sobre  $\mathbb{R}^d$ .

En su paper, Groisman et al. muestran que

$$\lim_{N \rightarrow \infty} n^\beta D_{\mathbf{X}_n, \alpha}(x, y) = \mu \mathcal{D}_{f, \beta}(x, y) \quad (15)$$

donde  $\beta = (a - 1)/d$ ,  $n \geq n_0$  y  $\mu$  es una constante adecuada.

# Distancia de Fermat muestral

Para  $\alpha \geq 1$  y  $x, y \in \mathbb{R}^d$ , la *Distancia Muestral de Fermat* se define como

$$D_{\mathbf{X}, \alpha} = \inf \left\{ \sum_{j=1}^{K-1} \|q_{j+1} - q_j\|^\alpha : (q_1, \dots, q_K) \text{ es un camino de } x \text{ a } y, K \geq 1 \right\} \quad (14)$$

donde los  $q_j$  son elementos de la muestra  $\mathbf{X}$ . Nótese que  $D_{\mathbf{X}, \alpha}$  satisface la desigualdad triangular, define una métrica sobre  $\mathbf{X}$  y una pseudo-métrica sobre  $\mathbb{R}^d$ .

En su paper, Groisman et al. muestran que

$$\lim_{N \rightarrow \infty} n^\beta D_{\mathbf{X}_n, \alpha}(x, y) = \mu \mathcal{D}_{f, \beta}(x, y) \quad (15)$$

donde  $\beta = (a - 1)/d$ ,  $n \geq n_0$  y  $\mu$  es una constante adecuada.

¡Esta sí la podemos aprender de los datos! 💪

## **Todo junto:**

---

Clasificación en variedades desconocidas  
por estimación de densidad por núcleos  
con Distancia de Fermat Muestral

# Algunas dudas

- Entrenar el clasificador por validación cruzada está OK: como  $\mathbf{X}_{\text{train}} \subseteq \mathbf{X}$  y  $\mathbf{X}_{\text{test}} \subseteq \mathbf{X}$ , se sigue que  $\forall (a, b) \in \{\mathbf{X}_{\text{train}} \times \mathbf{X}_{\text{test}}\} \subseteq \{\mathbf{X} \times \mathbf{X}\}$  y  $D_{\mathbf{X}, \alpha}(a, b)$  está bien definida.

# Algunas dudas

- Entrenar el clasificador por validación cruzada está OK: como  $\mathbf{X}_{\text{train}} \subseteq \mathbf{X}$  y  $\mathbf{X}_{\text{test}} \subseteq \mathbf{X}$ , se sigue que  $\forall (a, b) \in \{\mathbf{X}_{\text{train}} \times \mathbf{X}_{\text{test}}\} \subseteq \{\mathbf{X} \times \mathbf{X}\}$  y  $D_{\mathbf{X}, \alpha}(a, b)$  está bien definida. ¿Cómo sé la distancia muestral de una *nueva* observación  $x_0$ , a los elementos de cada clase?

# Algunas dudas

- Entrenar el clasificador por validación cruzada está OK: como  $\mathbf{X}_{\text{train}} \subseteq \mathbf{X}$  y  $\mathbf{X}_{\text{test}} \subseteq \mathbf{X}$ , se sigue que  $\forall (a, b) \in \{\mathbf{X}_{\text{train}} \times \mathbf{X}_{\text{test}}\} \subseteq \{\mathbf{X} \times \mathbf{X}\}$  y  $D_{\mathbf{X}, \alpha}(a, b)$  está bien definida. ¿Cómo sé la distancia muestral de una *nueva* observación  $x_0$ , a los elementos de cada clase?

Para cada una de las  $g_i \in \mathcal{G}$  clases, definimos el conjunto

$$Q_i = \{x_0\} \cup \{x_j : x_j \in \mathbf{X}, g_j = g_i, j \in \{1, \dots, N\}\} \quad (17)$$

y calculamos  $D_{Q_i, \alpha}(x_0, \cdot)$

# Algunas dudas

- El clasificador de Loubes & Pelletier asume que todas las clases están soportadas en la misma variedad  $\mathcal{M}$ . ¿Quién dice que ello vale para las diferentes clases?

# Algunas dudas

- El clasificador de Loubes & Pelletier asume que todas las clases están soportadas en la misma variedad  $\mathcal{M}$ . ¿Quién dice que ello vale para las diferentes clases?

¡Nadie! Pero

1. No hace falta dicho supuesto, y en el peor de los casos, podemos asumir que la unión de las clases está soportada en *cierta* variedad de Riemman, que resulta de (¿la clausura de?) la unión de sus soportes individuales.

# Algunas dudas

- El clasificador de Loubes & Pelletier asume que todas las clases están soportadas en la misma variedad  $\mathcal{M}$ . ¿Quién dice que ello vale para las diferentes clases?

¡Nadie! Pero

1. No hace falta dicho supuesto, y en el peor de los casos, podemos asumir que la unión de las clases está soportada en cierta variedad de Riemann, que resulta de (¿la clausura de?) la unión de sus soportes individuales.
2. Sí es cierto que si las variedades (y las densidades que soportan) difieren, tanto el  $\alpha_i^*$  como el  $h_i$  \* «óptimos» para los estimadores de densidad individuales no tienen por qué coincidir.

# Algunas dudas

- El clasificador de Loubes & Pelletier asume que todas las clases están soportadas en la misma variedad  $\mathcal{M}$ . ¿Quién dice que ello vale para las diferentes clases?

¡Nadie! Pero

1. No hace falta dicho supuesto, y en el peor de los casos, podemos asumir que la unión de las clases está soportada en cierta variedad de Riemann, que resulta de (¿la clausura de?) la unión de sus soportes individuales.
2. Sí es cierto que si las variedades (y las densidades que soportan) difieren, tanto el  $\alpha_i^*$  como el  $h_i$  \* «óptimos» para los estimadores de densidad individuales no tienen por qué coincidir.
3. Aunque las densidades individuales  $f_i$  estén bien estimadas, el clasificador resultante puede ser mal(ard)o si no diferencia bien «en las fronteras». Por simplicidad, además, decidimos parametrizar el clasificador con dos únicos hiperparámetros globales:  $\alpha, h$ .

# Diseño experimental

1. Desarrollamos un clasificador compatible con el *framework* de [scikit-learn](#) según los lineamientos de Loubes & Pelleteir, que apodamos KDC.

---

<sup>1</sup>sólo se consideró su exactitud. ya que no es un clasificador suave

# Diseño experimental

1. Desarrollamos un clasificador compatible con el *framework* de [scikit-learn](#) según los lineamientos de Loubes & Pelleteir, que apodamos KDC.
2. Implementamos el estimador de la distancia muestral de Fermat, y combinándolo con KDC, obtenemos la titular «Clasificación por KDE con Distancia de Fermat», FKDC.

---

<sup>1</sup>sólo se consideró su exactitud, ya que no es un clasificador suave

# Diseño experimental

1. Desarrollamos un clasificador compatible con el *framework* de [scikit-learn](#) según los lineamientos de Loubes & Pelleteir, que apodamos KDC.
2. Implementamos el estimador de la distancia muestral de Fermat, y combinándolo con KDC, obtenemos la titular «Clasificación por KDE con Distancia de Fermat», FKDC.
3. Evaluamos el *pseudo- $R^2$*  y la *exactitud* («accuracy») de los clasificadores propuestos en diferentes *datasets*, relativa a técnicas bien establecidas:

---

<sup>1</sup>sólo se consideró su exactitud, ya que no es un clasificador suave

# Diseño experimental

1. Desarrollamos un clasificador compatible con el *framework* de [scikit-learn](#) según los lineamientos de Loubes & Pelleteir, que apodamos KDC.
2. Implementamos el estimador de la distancia muestral de Fermat, y combinándolo con KDC, obtenemos la titular «Clasificación por KDE con Distancia de Fermat», FKDC.
3. Evaluamos el *pseudo-R<sup>2</sup>* y la *exactitud* («accuracy») de los clasificadores propuestos en diferentes *datasets*, relativa a técnicas bien establecidas:
  - regresión logística (LR)
  - clasificador de soporte vectorial (svc)<sup>1</sup>
  - *gradient boosting trees* (GBT)
  - k-vecinos-más-cercanos (KN)
  - Naive Bayes Gaussiano (GNB)

---

<sup>1</sup>sólo se consideró su exactitud. ya que no es un clasificador suave

# Diseño experimental

- La implementación de `KNeighbors` de referencia acepta distancias precomputadas, así que incluimos una versión con distancia de Fermat, que apodamos `F(ermat)KN`.

# Diseño experimental

- La implementación de **KNeighbors** de referencia acepta distancias precomputadas, así que incluimos una versión con distancia de Fermat, que apodamos **F(ermat)KN**.
- Para ser «justos», se reservó una porción de los datos para la evaluación comparada, y del resto, cada algoritmo fue entrenado repetidas veces por validación cruzada de 5 pliegos, en una extensísima grilla de hiperparametrizaciones. Este procedimiento **se repitió 25 veces en cada dataset**.

# Diseño experimental

- La implementación de `KNeighbors` de referencia acepta distancias precomputadas, así que incluimos una versión con distancia de Fermat, que apodamos `F(ermat)KN`.
- Para ser «justos», se reservó una porción de los datos para la evaluación comparada, y del resto, cada algoritmo fue entrenado repetidas veces por validación cruzada de 5 pliegos, en una extensísima grilla de hiperparametrizaciones. Este procedimiento **se repitió 25 veces en cada dataset**.
- La función de score elegida fue `neg_log_loss` ( $= \ell$ ) para los clasificadores suaves, y `accuracy` para los duros.

# Diseño experimental

- Para tener una idea «sistémica» de la performance de los algoritmos, evaluaremos su performance con *datasets* que varíen en el tamaño muestral  $N$ , la dimensión  $p$  de las  $X_i$ , el nro. de clases  $k$  y su origen («real» o «sintético»).

# Diseño experimental

- Para tener una idea «sistémica» de la performance de los algoritmos, evaluaremos su performance con *datasets* que varíen en el tamaño muestral  $N$ , la dimensión  $p$  de las  $X_i$ , el nro. de clases  $k$  y su origen («real» o «sintético»).
- Cuando creamos datos sintéticos en variedades con dimensión intrínseca menor a la ambiente, (casi) cualquier clasificador competente alcanza exactitud perfecta; para complejizar la tarea, agregamos un poco de «ruido» a las observaciones, y también analizamos sus efectos.

# Regla de Parsimonia

- ¿Qué parametrización elegir cuando «en test da todo igual»?

# Regla de Parsimonia

- ¿Qué parametrización elegir cuando «en test da todo igual»?  
 de Occam: la más «sencilla» (TBD)

# Regla de Parsimonia

- ¿Qué parametrización elegir cuando «en test da todo igual»?
  - de Occam: la más «sencilla» (TBD)
- ¿Qué parametrización elegir cuando «en test da **casi** todo igual»?

# Regla de Parsimonia

- ¿Qué parametrización elegir cuando «en test da todo igual»?
  - Knife icon of Occam: la más «sencilla» (TBD)
- ¿Qué parametrización elegir cuando «en test da **casi** todo igual»?

**Regla de  $1\sigma$** : De las que estén a  $1\sigma$  de la mejor, la más sencilla.

# Regla de Parsimonia

- ¿Qué parametrización elegir cuando «en test da todo igual»?



de Occam: la más «sencilla» (TBD)

- ¿Qué parametrización elegir cuando «en test da **casi** todo igual»?

**Regla de  $1\sigma$** : De las que estén a  $1\sigma$  de la mejor, la más sencilla.

¿Sabemos cuánto vale  $\sigma$ ?

## $R^2$ de McFadden

Sea  $\mathcal{C}_0$  el clasificador «base», que asigna a cada observación y posible clase, la frecuencia empírica de clase encontrada en la muestra  $\mathbf{X}$ . Para todo clasificador suave  $\mathcal{C}$ , definimos el  $R^2$  de McFadden como

$$R^2(\mathcal{C} \mid \mathbf{X}) = 1 - \frac{\ell(\mathcal{C})}{\ell(\mathcal{C}_0)} \quad (24)$$

donde  $\ell(\cdot)$  es la log-verosimilitud clásica. Nótese que  $R^2(\mathcal{C}_0) = 0$ .

## $R^2$ de McFadden

Sea  $\mathcal{C}_0$  el clasificador «base», que asigna a cada observación y posible clase, la frecuencia empírica de clase encontrada en la muestra  $\mathbf{X}$ . Para todo clasificador suave  $\mathcal{C}$ , definimos el  $R^2$  de McFadden como

$$R^2(\mathcal{C} \mid \mathbf{X}) = 1 - \frac{\ell(\mathcal{C})}{\ell(\mathcal{C}_0)} \quad (24)$$

donde  $\ell(\cdot)$  es la log-verosimilitud clásica. Nótese que  $R^2(\mathcal{C}_0) = 0$ . A su vez, para un clasificador perfecto  $\mathcal{C}^*$  que otorgue toda la masa de probabilidad a la clase correcta, tendrá  $L(\mathcal{C}^*) = 1$  y log-verosimilitud igual a 0, de manera que  $R^2(\mathcal{C}^*) = 1 - 0 = 1$ .

## $R^2$ de McFadden

Sea  $\mathcal{C}_0$  el clasificador «base», que asigna a cada observación y posible clase, la frecuencia empírica de clase encontrada en la muestra  $\mathbf{X}$ . Para todo clasificador suave  $\mathcal{C}$ , definimos el  $R^2$  de McFadden como

$$R^2(\mathcal{C} \mid \mathbf{X}) = 1 - \frac{\ell(\mathcal{C})}{\ell(\mathcal{C}_0)} \quad (24)$$

donde  $\ell(\cdot)$  es la log-verosimilitud clásica. Nótese que  $R^2(\mathcal{C}_0) = 0$ . A su vez, para un clasificador perfecto  $\mathcal{C}^*$  que otorgue toda la masa de probabilidad a la clase correcta, tendrá  $L(\mathcal{C}^*) = 1$  y log-verosimilitud igual a 0, de manera que  $R^2(\mathcal{C}^*) = 1 - 0 = 1$ .

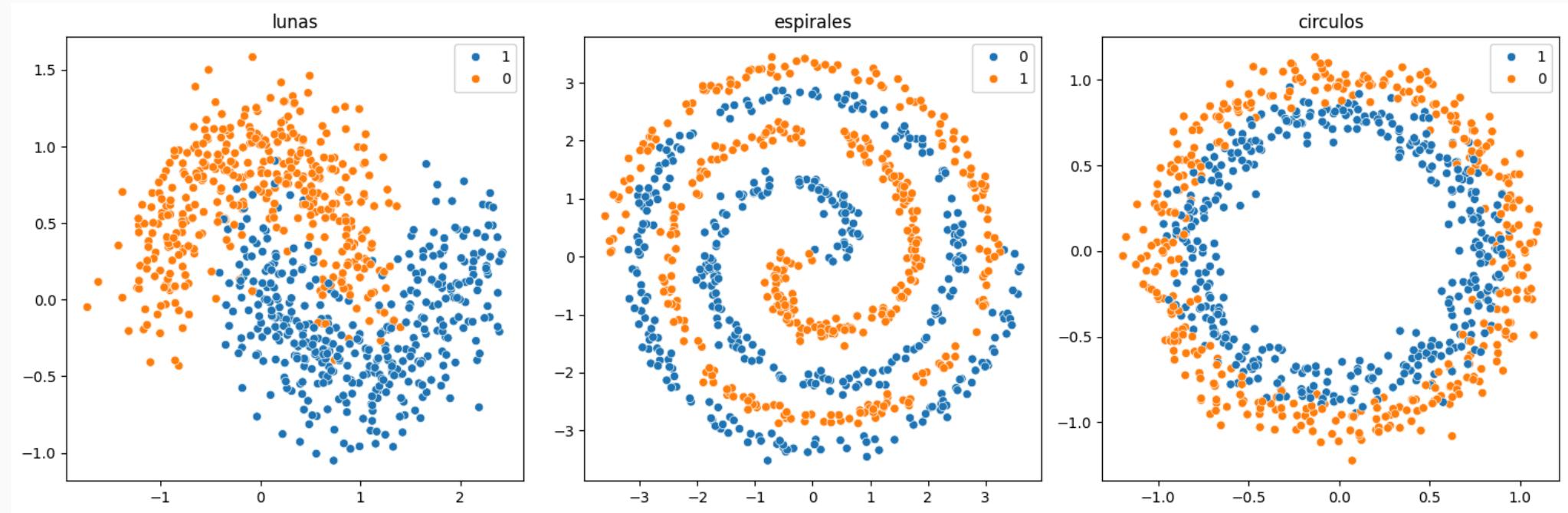
Sin embargo, un clasificador *peor* que  $\mathcal{C}_0$  en tanto asigne bajas probabilidades ( $\approx 0$ ) a las clases correctas, puede tener un  $R^2$  infinitamente negativo.

# Resultados

---

# 2D, 2 clases: excelente $R^2$ con exactitud competitiva

## Con Bajo Ruido



# 2D, 2 clases: excelente $R^2$ con exactitud competitiva

lunas\_lo (acc: 93.66%)

	delta_acc	r2
clf		
fkdc	-0.29	75.58
fkn	-0.04	74.43
kn	0.00	74.30
kdc	-0.33	73.42
gbt	-0.67	70.14
lr	-8.71	49.90
slr	-7.93	49.62
gnb	-10.12	48.83
base	-45.12	0.00
svc	-0.98	NaN

circulos\_lo (acc: 88.15%)

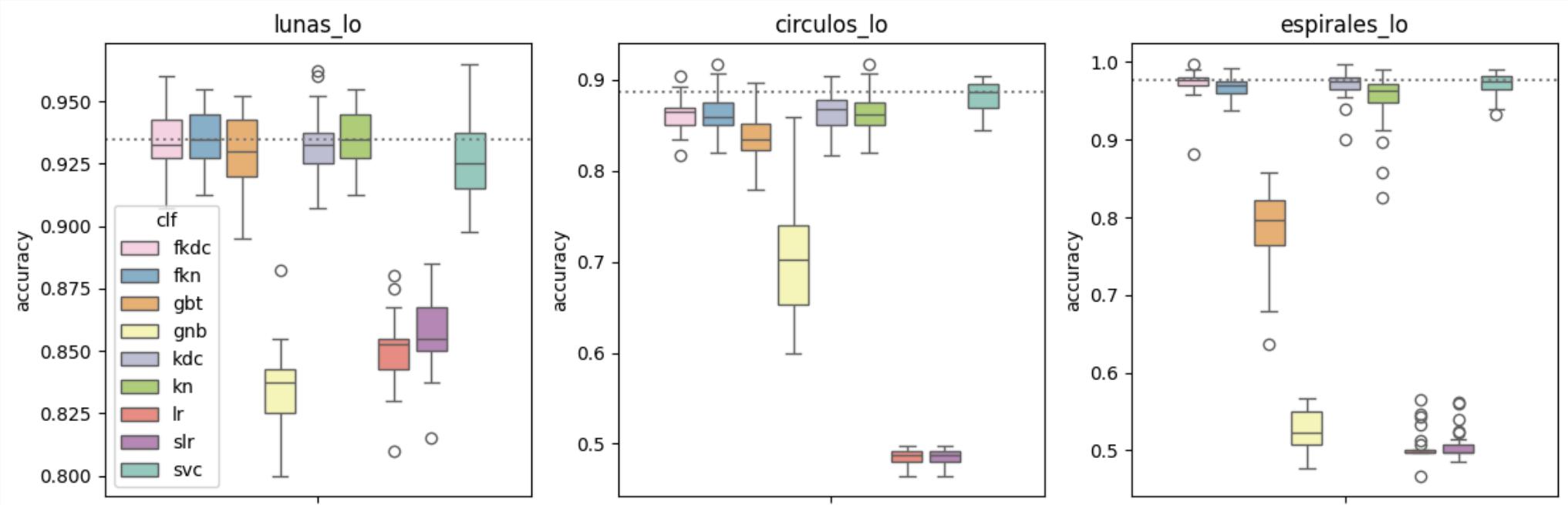
	delta_acc	r2
clf		
fkdc	-1.91	49.25
kdc	-1.71	48.58
fkn	-1.82	45.13
kn	-1.82	44.92
gbt	-4.64	43.42
gnb	-17.43	5.13
base	-39.61	0.00
lr	-39.61	-0.00
slr	-39.61	-0.00
svc	0.00	NaN

espirales\_lo (acc: 97.27%)

	delta_acc	r2
clf		
fkdc	0.00	84.66
kdc	-0.14	82.36
kn	-2.29	71.47
fkn	-0.50	70.64
gbt	-18.68	28.82
gnb	-44.59	0.35
lr	-46.74	0.17
slr	-46.56	0.17
base	-47.52	0.00
svc	-0.18	NaN

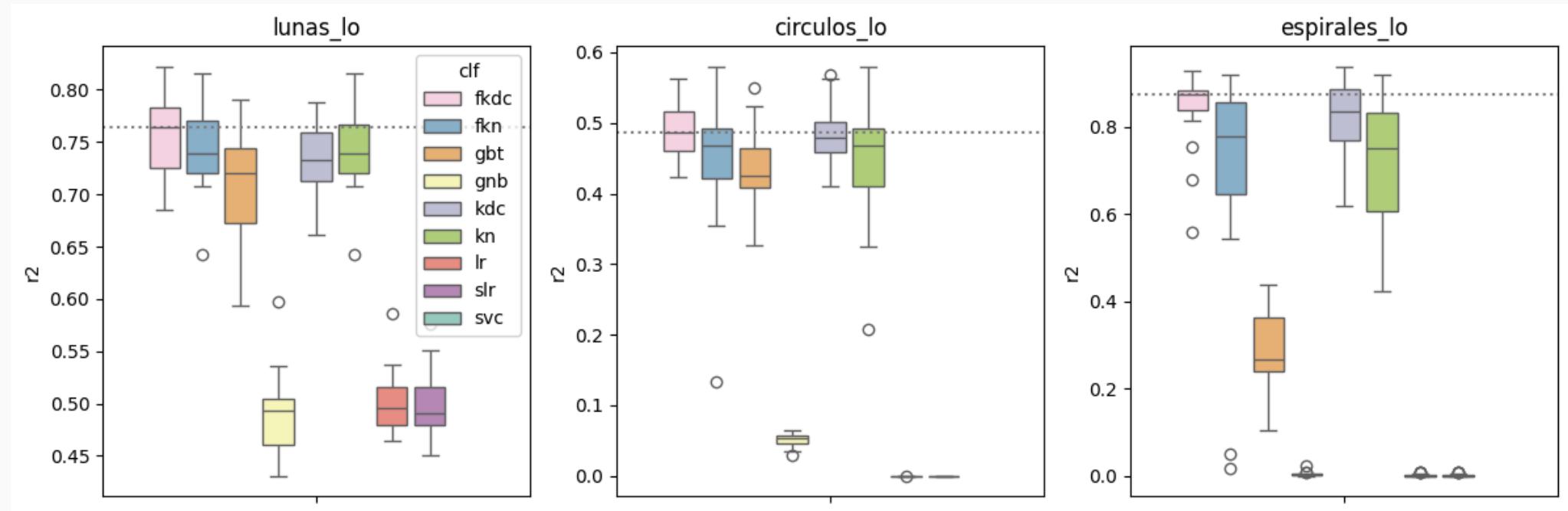
# 2D, 2 clases: excelente $R^2$ con exactitud competitiva

## Boxplot Accuracy



# 2D, 2 clases: excelente $R^2$ con exactitud competitiva

## Boxplot $R^2$



## Superposición de parámetros: $\alpha$ y $h$

- El uso de la distancia de Fermat muestral no hiere la performance, pero las mejoras son nulas o marginales. ¿Por qué?

## Superposición de parámetros: $\alpha$ y $h$

- El uso de la distancia de Fermat muestral no hiere la performance, pero las mejoras son nulas o marginales. ¿Por qué?

Si recordamos  $\hat{f}_{K,N}$  según Loubes & Pelletier, al núcleo  $K$  se lo evalúa sobre

$$\frac{d(x_0, X_i)}{h}, \quad d = D_{Q_i, \alpha} \tag{25}$$

## Superposición de parámetros: $\alpha$ y $h$

- El uso de la distancia de Fermat muestral no hiere la performance, pero las mejoras son nulas o marginales. ¿Por qué?

Si recordamos  $\hat{f}_{K,N}$  según Loubes & Pelletier, al núcleo  $K$  se lo evalúa sobre

$$\frac{d(x_0, X_i)}{h}, \quad d = D_{Q_i, \alpha} \tag{25}$$

Lo que  $\alpha$  afecta a  $\hat{f}$  vía  $d$ , también se puede conseguir vía  $h$ .

## Superposición de parámetros: $\alpha$ y $h$

- El uso de la distancia de Fermat muestral no hiere la performance, pero las mejoras son nulas o marginales. ¿Por qué?

Si recordamos  $\hat{f}_{K,N}$  según Loubes & Pelletier, al núcleo  $K$  se lo evalúa sobre

$$\frac{d(x_0, X_i)}{h}, \quad d = D_{Q_i, \alpha} \tag{25}$$

Lo que  $\alpha$  afecta a  $\hat{f}$  vía  $d$ , también se puede conseguir vía  $h$ . Si  $D_{Q_i, \alpha} \propto \|\cdot\|$  (la distancia de fermat es proporcional a la euclídea), los efectos de  $\alpha$  y  $h$  se «solapan»

## Superposición de parámetros: $\alpha$ y $h$

- El uso de la distancia de Fermat muestral no hiere la performance, pero las mejoras son nulas o marginales. ¿Por qué?

Si recordamos  $\hat{f}_{K,N}$  según Loubes & Pelletier, al núcleo  $K$  se lo evalúa sobre

$$\frac{d(x_0, X_i)}{h}, \quad d = D_{Q_i, \alpha} \tag{25}$$

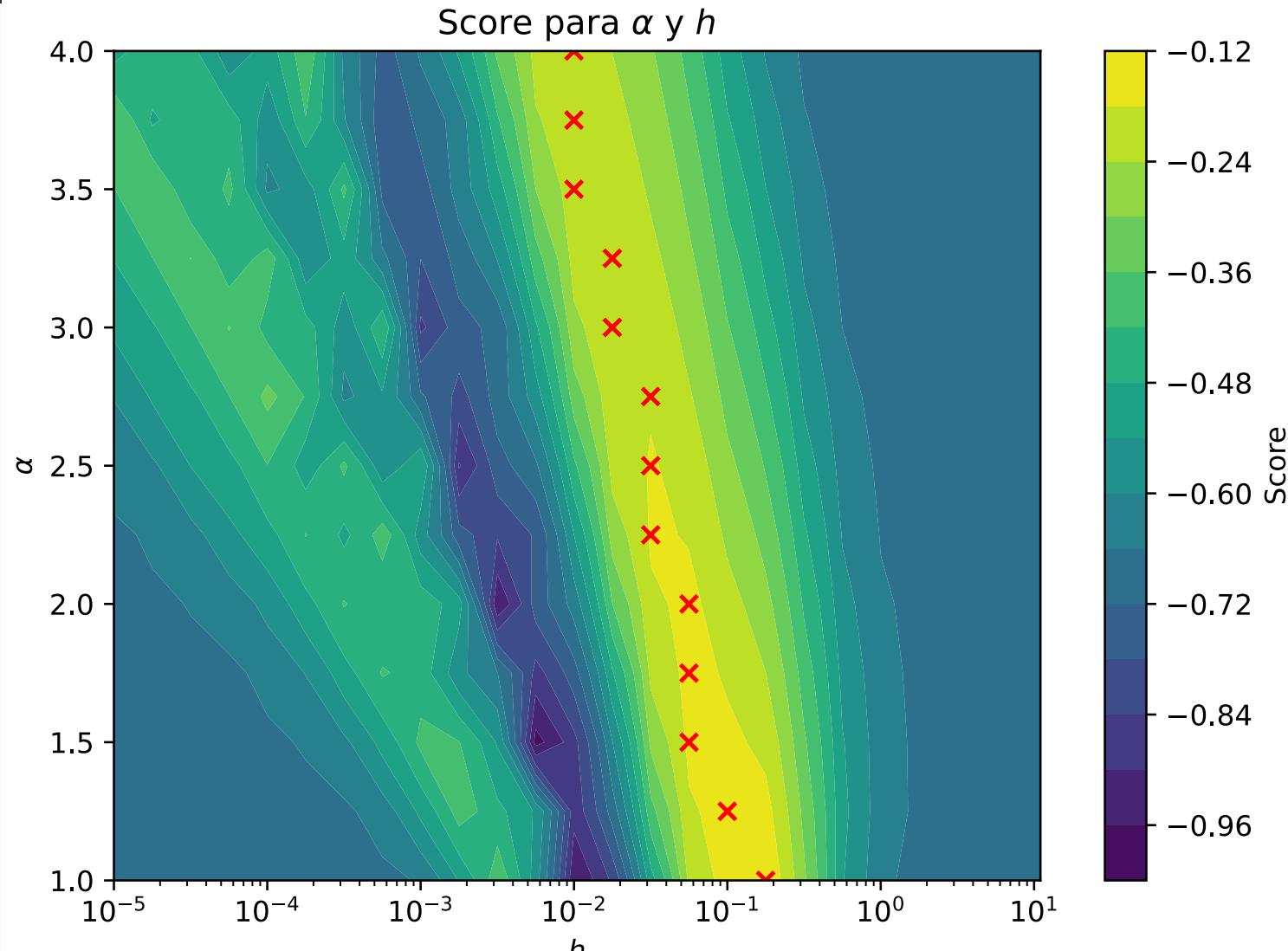
Lo que  $\alpha$  afecta a  $\hat{f}$  vía  $d$ , también se puede conseguir vía  $h$ . Si  $D_{Q_i, \alpha} \propto \|\cdot\|$  (la distancia de fermat es proporcional a la euclídea), los efectos de  $\alpha$  y  $h$  se «solapan»

... y sabemos que localmente, eso es cierto 😢

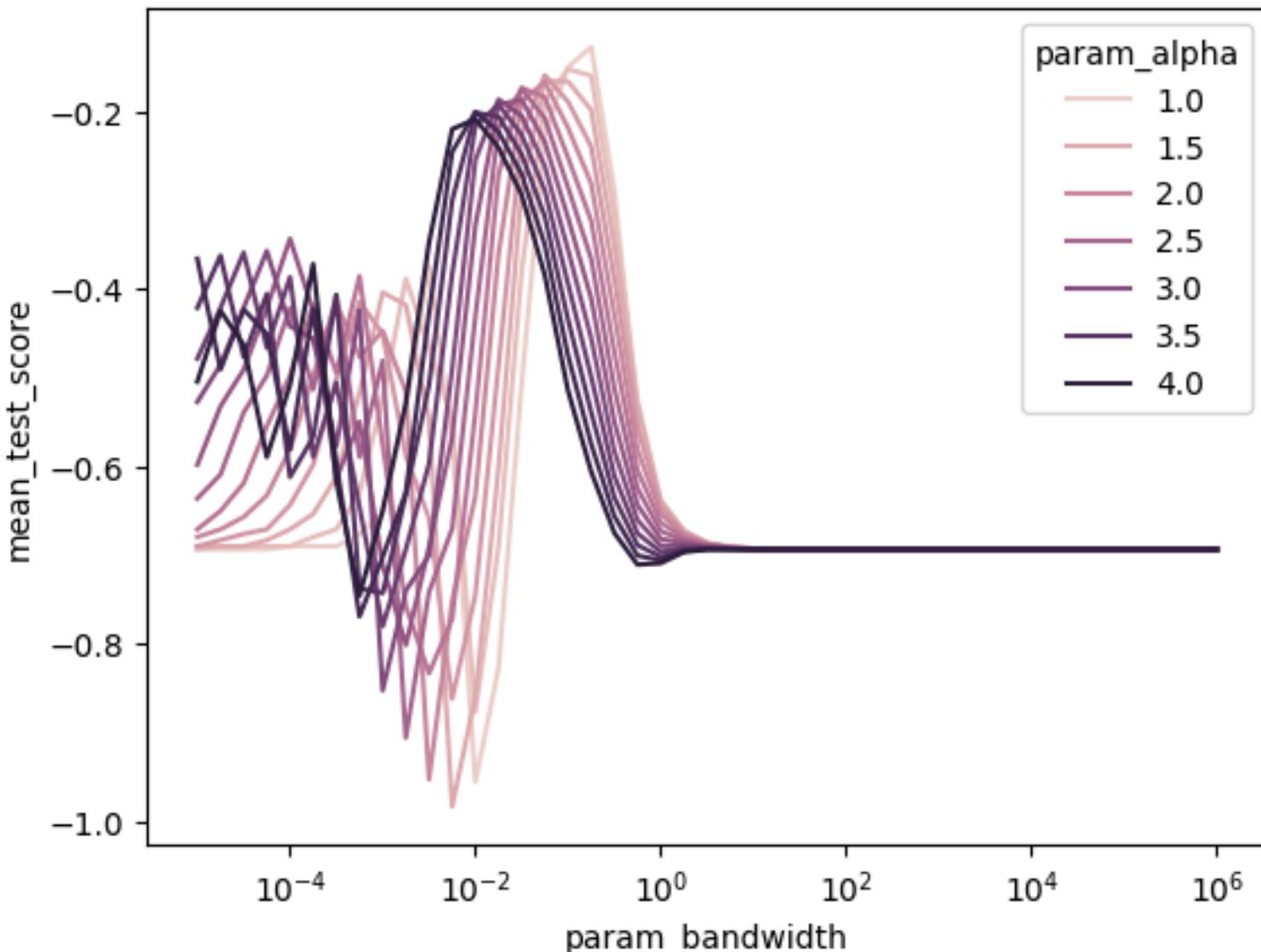
## Parámetros óptimos para (F)KDC en espirales\_lo

(fkdc, alpha)	(fkdc, bandwidth)	(kdc, bandwidth)	count
1.0	0.1000	0.1431	1
1.0	0.1778	0.1726	8
1.0	0.1778	0.2082	7
1.0	0.1778	0.2512	7
1.0	0.3162	0.3030	1
1.5	0.0032	0.2082	1

# Superficies (o paisajes) de score para (espirales\_lo, 1434)

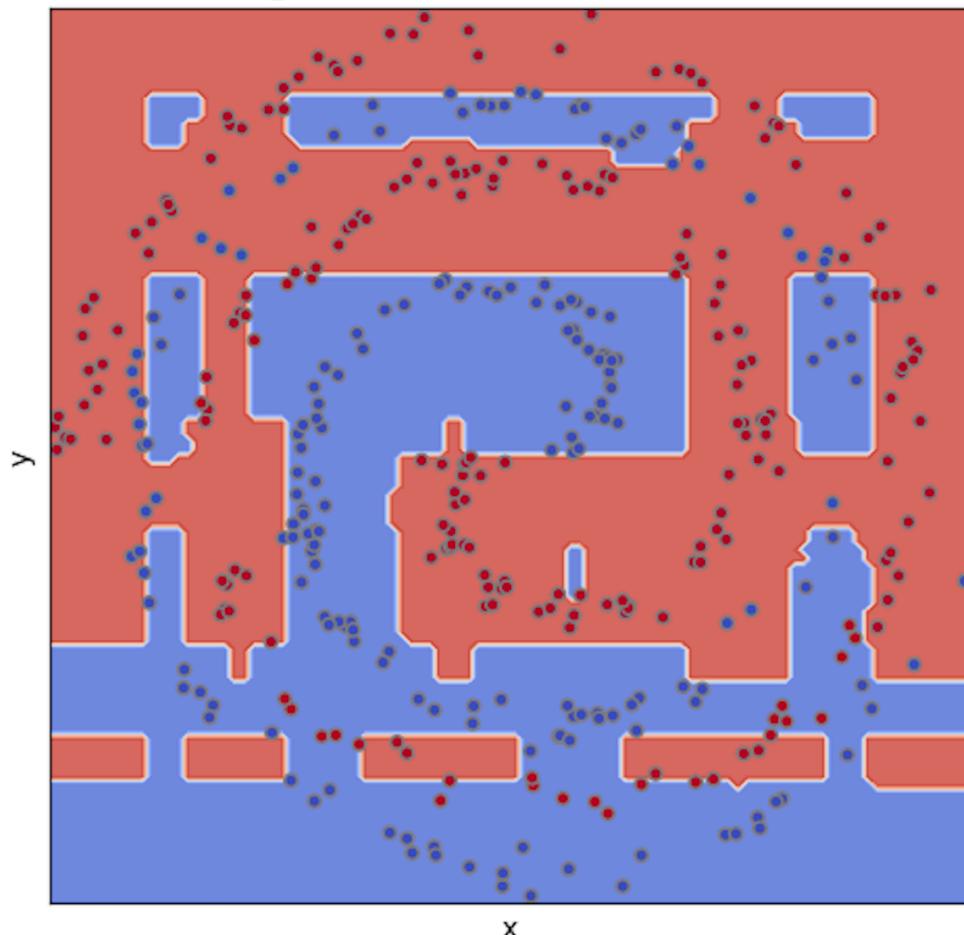


# Alt-viz: Perfiles de pérdida para (espirales\_lo, 1434)

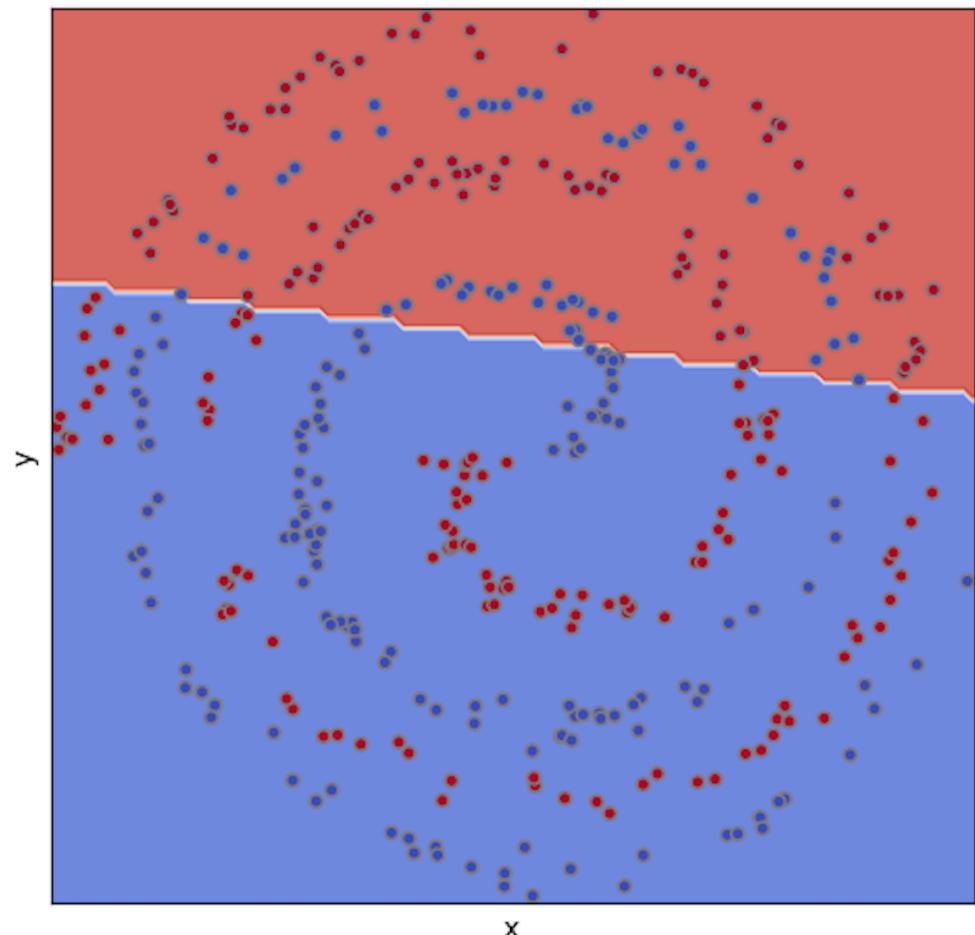


# Fronteras de decisión para (espirales\_lo, 1434)

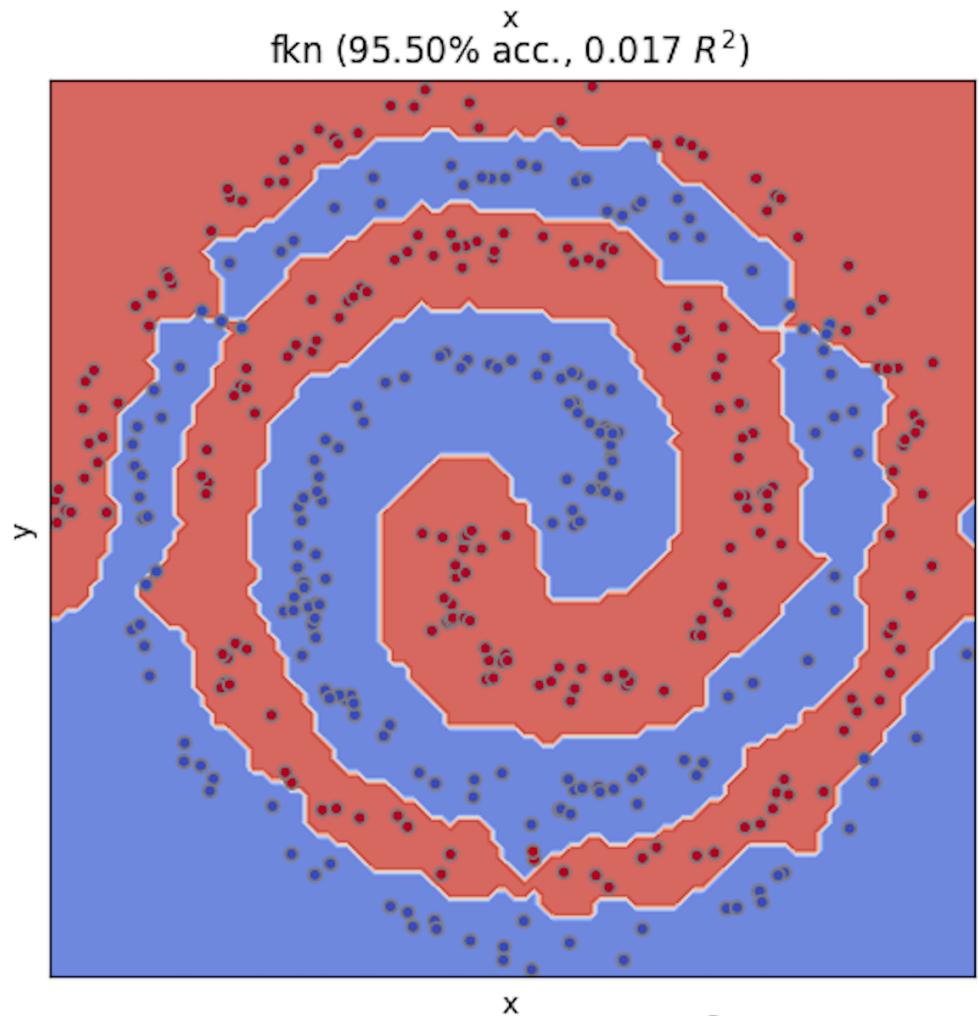
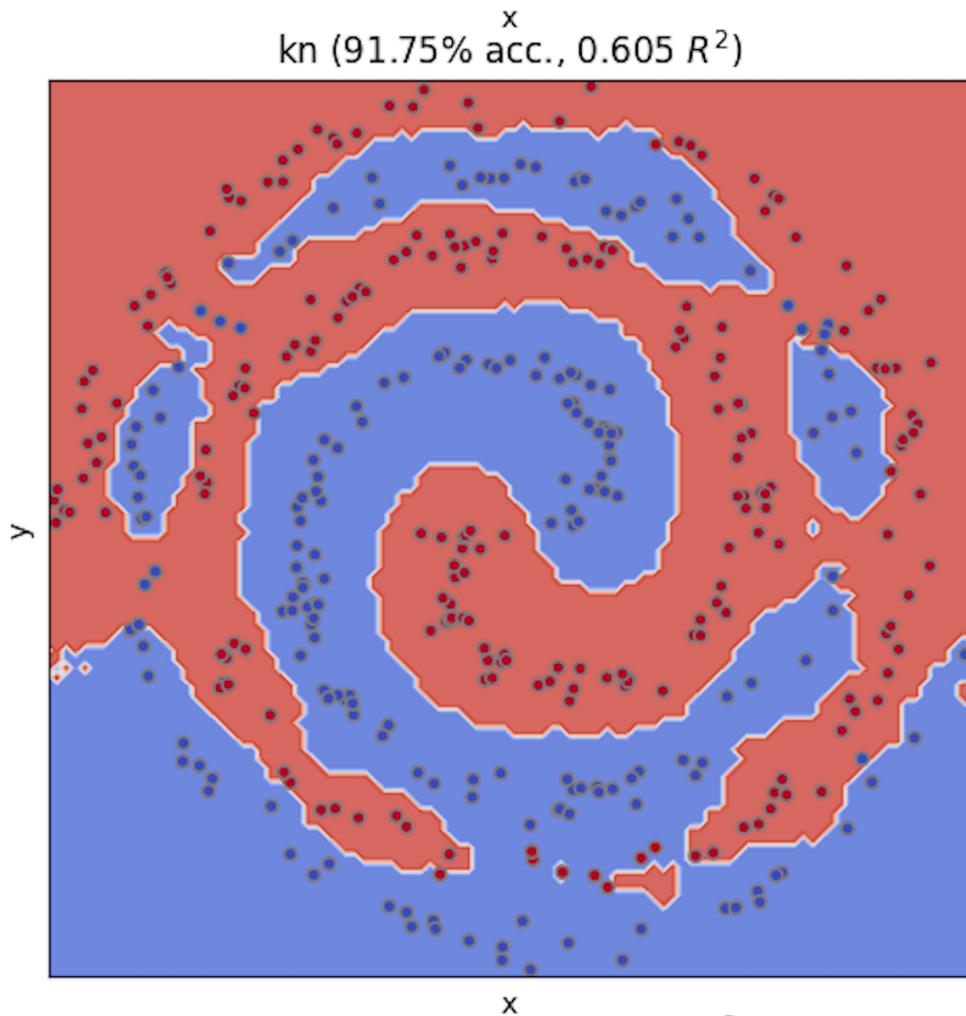
gbt (83.00% acc., 0.320  $R^2$ )



lr (56.50% acc., 0.003  $R^2$ )

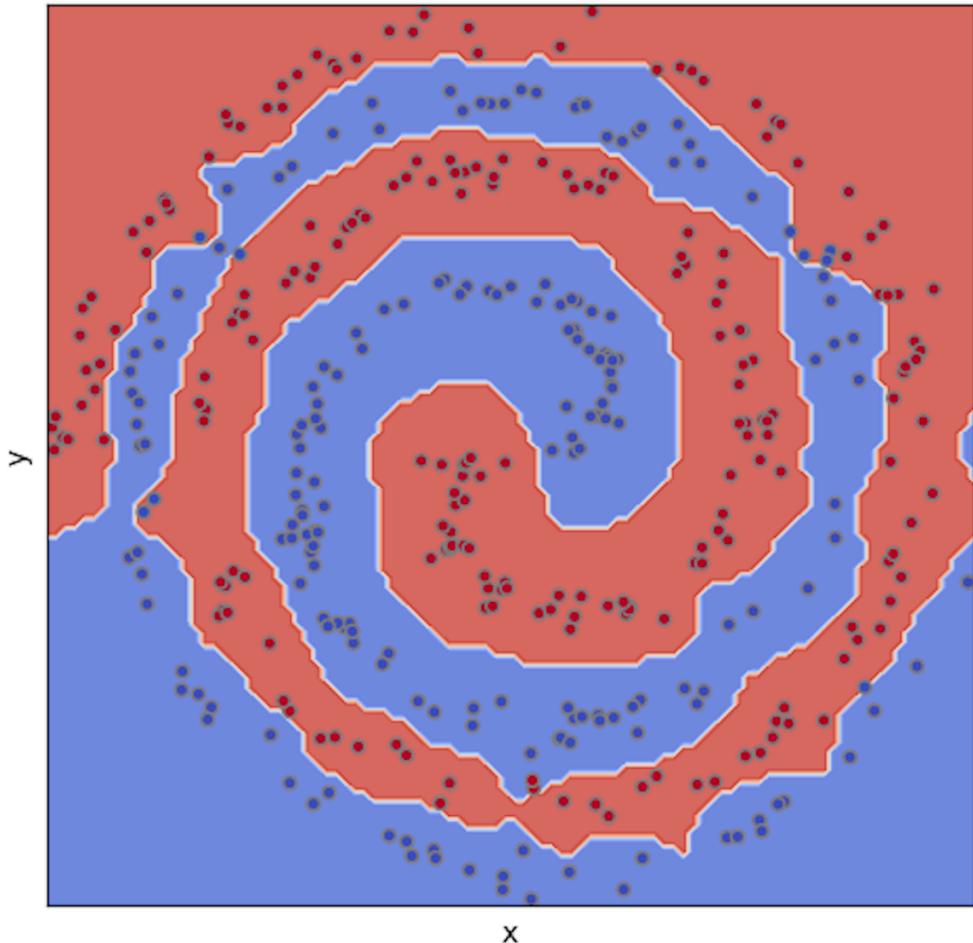


# Fronteras de decisión para (espirales\_lo, 1434)

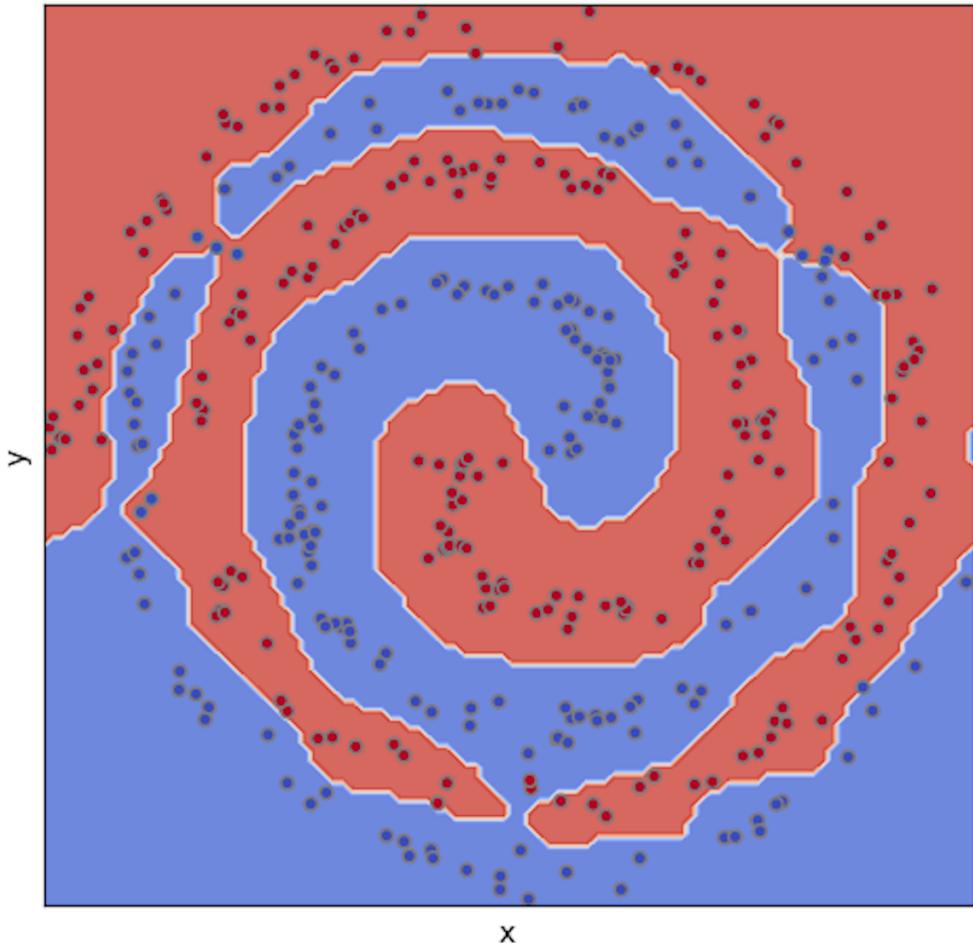


# Fronteras de decisión para (espirales\_lo, 1434)

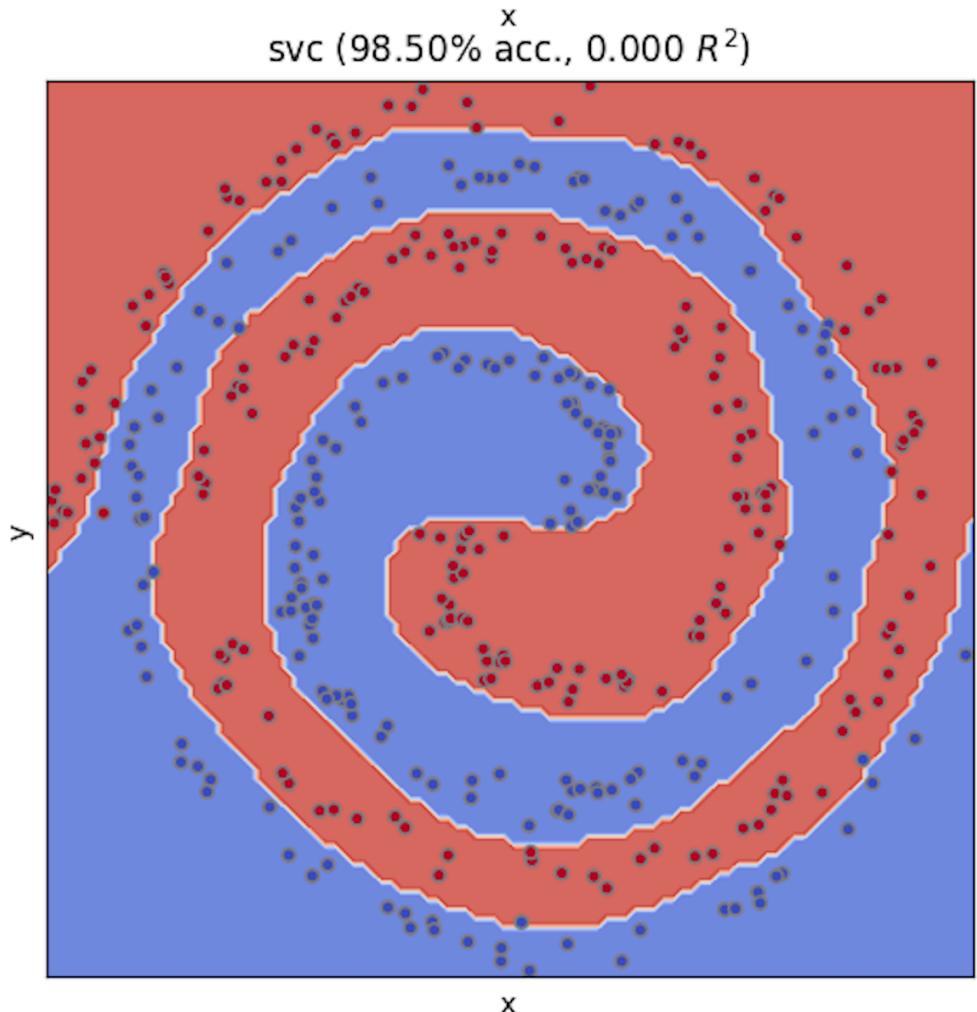
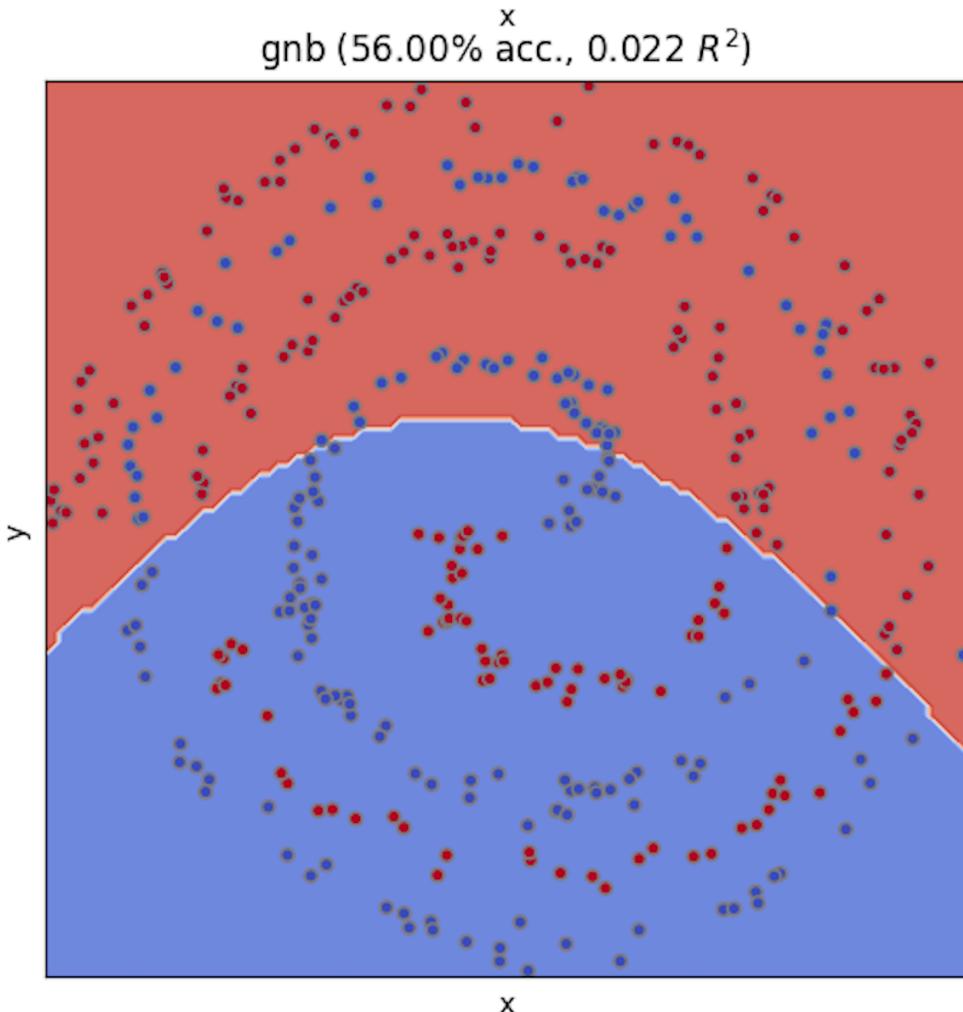
fkdc (96.00% acc., 0.848  $R^2$ )



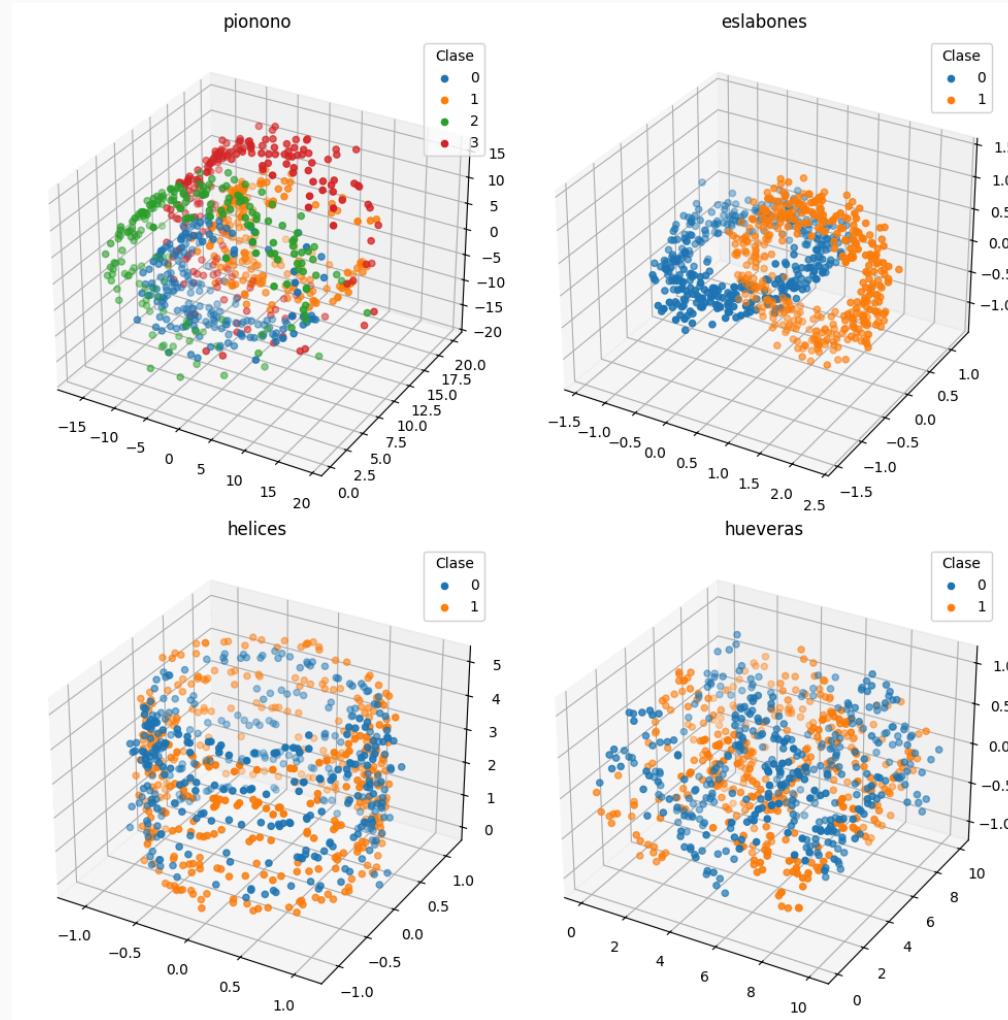
kdc (95.50% acc., 0.745  $R^2$ )



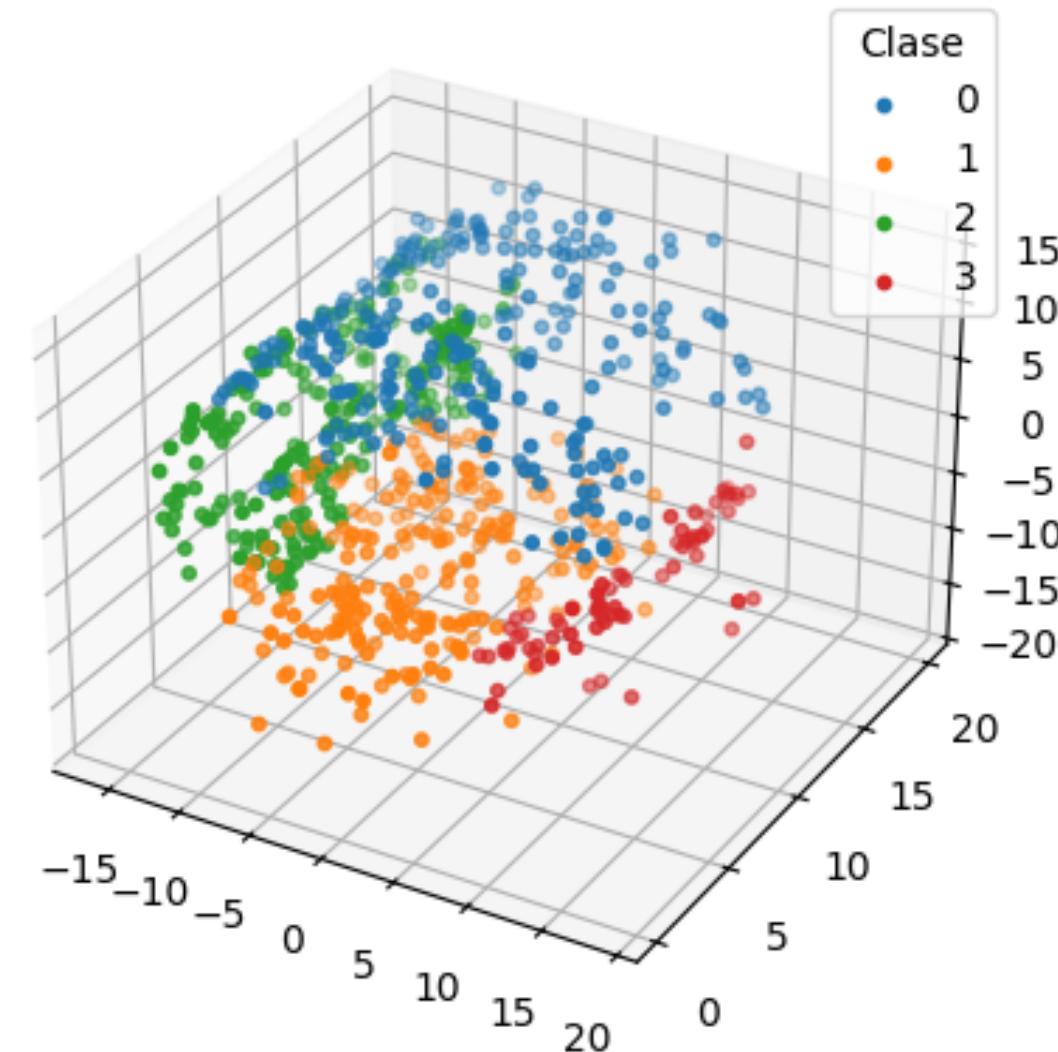
# Fronteras de decisión para (espirales\_lo, 1434)



# 3D, 2 clases + piononos



# 3D, 2 clases + piononos



# 3D, 2 clases + piononos

pionono\_0 (acc: 94.19%)

	delta_acc	r2
clf		
fkdc	-1.06	81.04
gbt	-2.37	81.02
kdc	-0.83	79.89
fkn	-2.65	72.55
kn	-3.57	70.62
gnb	-20.40	58.65
slr	-29.69	47.45
lr	-29.72	47.44
base	-71.44	0.00
svc	0.00	NaN

eslabones\_0 (acc: 99.9%)

	delta_acc	r2
clf		
kdc	-0.10	97.75
kn	-0.02	96.51
fkdc	-0.06	96.44
fkn	-0.02	96.04
gbt	-1.14	91.97
gnb	-10.68	75.16
lr	-33.32	22.04
slr	-33.30	21.87
base	-50.15	0.00
svc	0.00	NaN

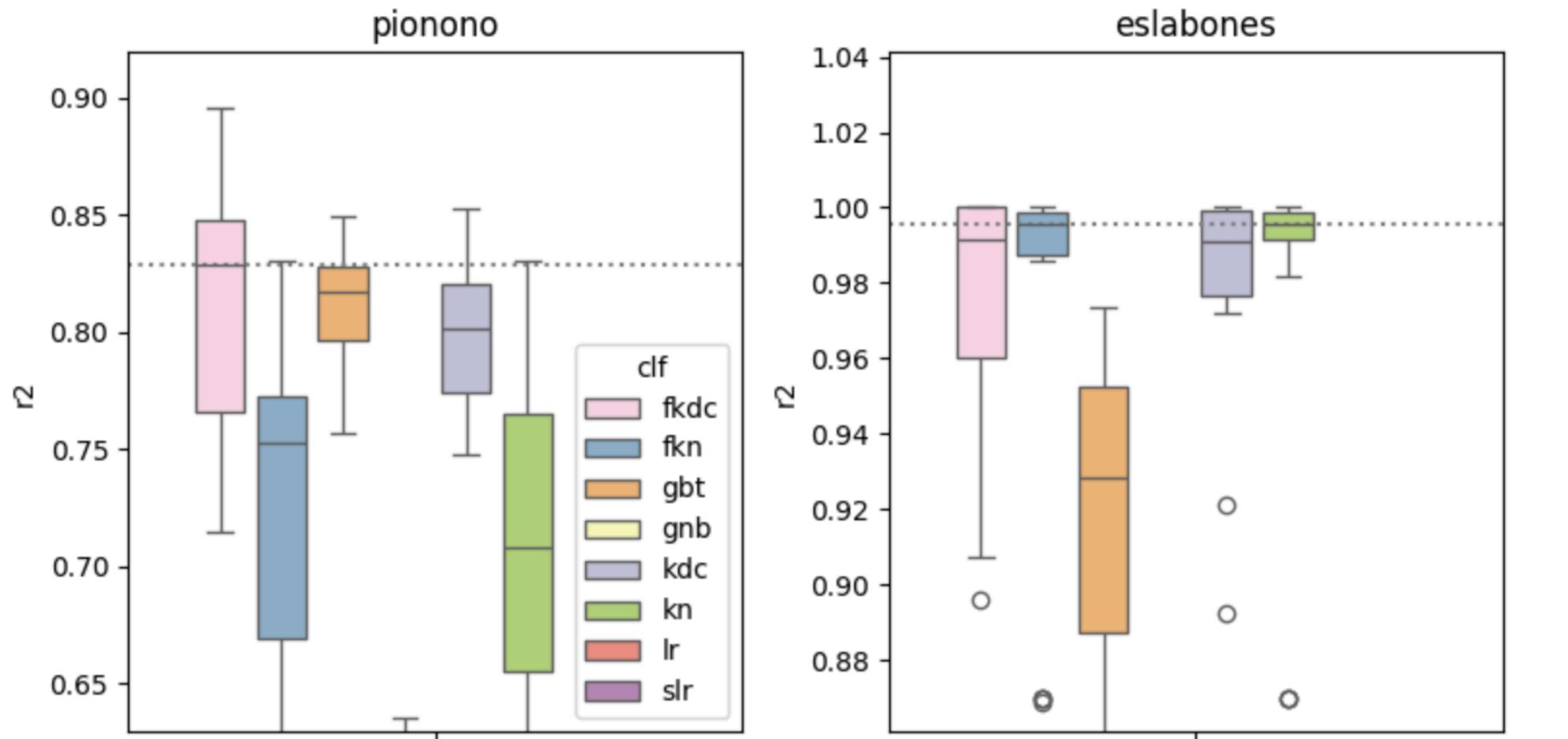
helices\_0 (acc: 99.16%)

	delta_acc	r2
clf		
fkdc	0.00	94.41
kdc	-0.06	86.10
fkn	-0.76	83.45
kn	-4.90	62.81
gbt	-9.48	57.56
gnb	-49.29	0.01
base	-49.41	0.00
lr	-49.41	0.00
slr	-49.41	0.00
svc	-24.03	NaN

hueveras\_0 (acc: 77.88%)

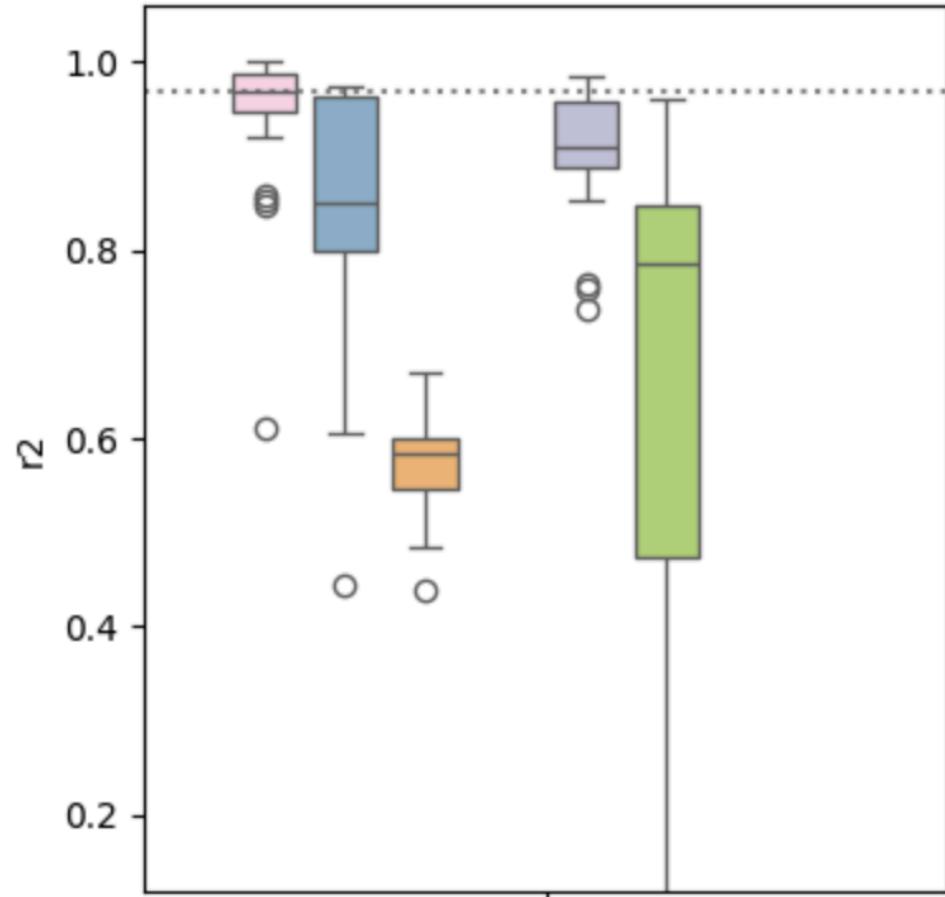
	delta_acc	r2
clf		
fkn	-1.60	30.20
fkdc	-2.57	29.40
kdc	-3.36	26.62
kn	-2.71	26.43
gbt	-19.90	3.44
gnb	-27.43	0.01
base	-28.13	0.00
slr	-28.13	0.00
lr	-28.15	-0.01
svc	0.00	NaN

# 3D, 2 clases + piononos

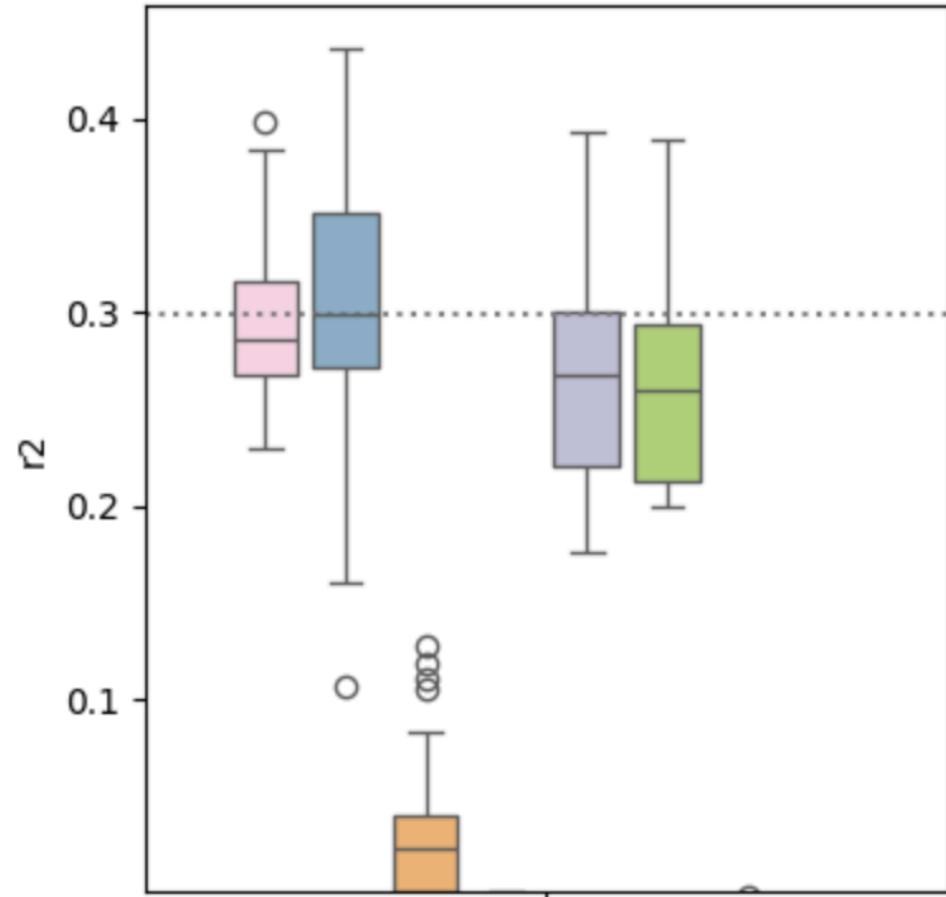


# 3D, 2 clases + piononos

helices



hueveras



# Parámetros óptimos para (F)KDC en helices\_0

(fkdc, alpha)	(fkdc, bandwidth)	(kdc, bandwidth)	count
1.00	0.1000	0.1431	6
1.00	0.0100	0.1431	3
1.25	0.0056	0.1726	3
1.00	0.0100	0.1726	2
1.00	0.1000	0.1186	1
1.25	0.0056	0.1431	1
1.25	0.0056	0.2082	1
1.25	0.0100	0.1431	1
1.50	0.0056	0.1726	1
1.75	0.0032	0.1431	1
1.75	0.0032	0.1726	1
2.00	0.0032	0.1431	1
2.00	0.0032	0.1726	1
2.50	0.0010	0.2082	1
2.50	0.0018	0.1726	1

# Microindiferencia, macrodiferencia

- En zonas con muchas observaciones (por tener alta  $f$  o alto  $N$ ) sampleadas, la distancia de Fermat y la euclídea coinciden.

# Microindiferencia, macrodiferencia

- En zonas con muchas observaciones (por tener alta  $f$  o alto  $N$ ) sampleadas, la distancia de Fermat y la euclídea coinciden.
- «Localmente», siempre van a coincidir, aunque sea en un vecindario muy pequeño.

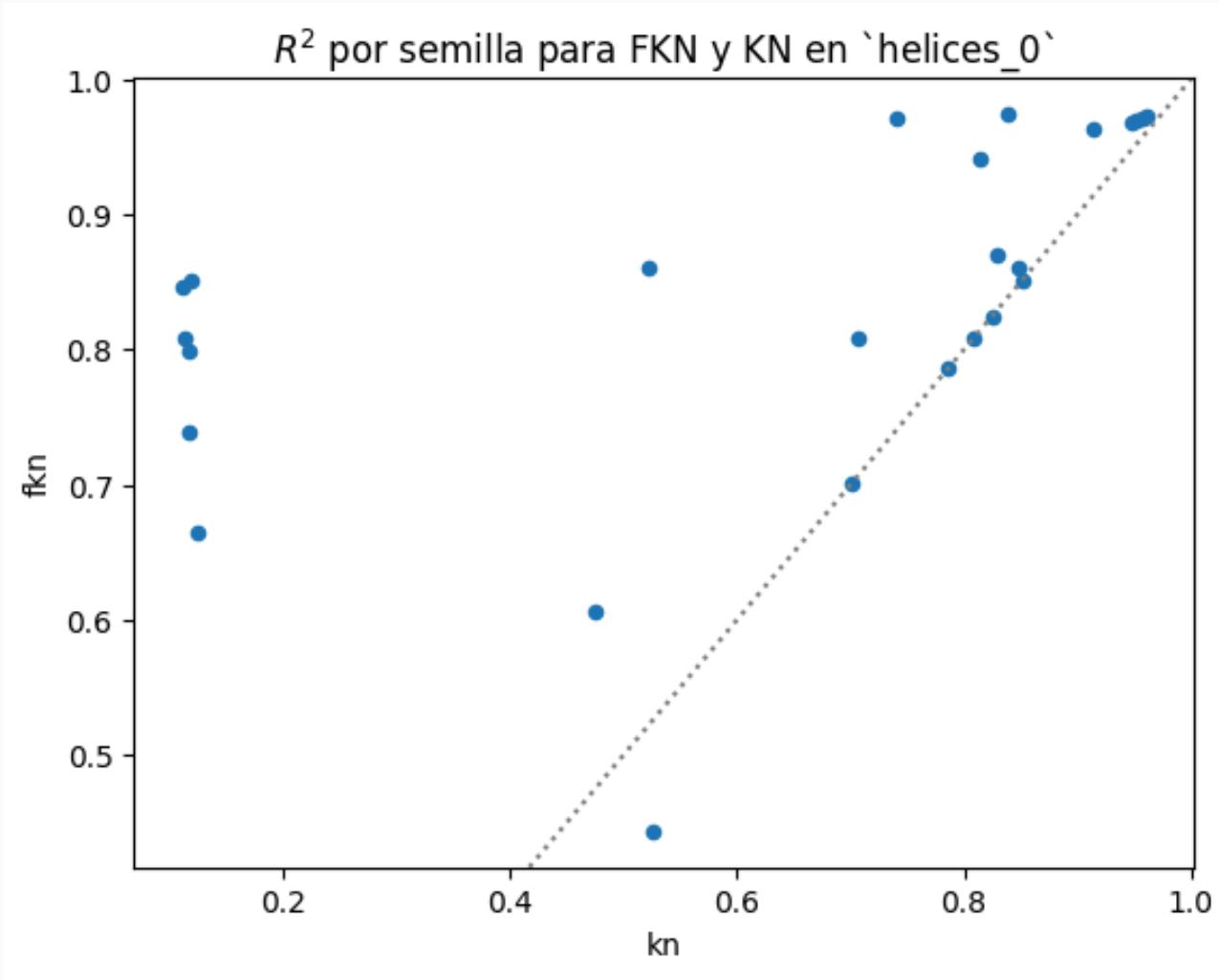
# Microindiferencia, macrodiferencia

- En zonas con muchas observaciones (por tener alta  $f$  o alto  $N$ ) sampleadas, la distancia de Fermat y la euclídea coinciden.
- «Localmente», siempre van a coincidir, aunque sea en un vecindario muy pequeño.
- Si el algoritmo de clasificación sólo depende de ese vecindario local para clasificar, no hay ganancia en la distancia de Fermat.

# Microindiferencia, macrodiferencia

- En zonas con muchas observaciones (por tener alta  $f$  o alto  $N$ ) sampleadas, la distancia de Fermat y la euclídea coinciden.
- «Localmente», siempre van a coincidir, aunque sea en un vecindario muy pequeño.
- Si el algoritmo de clasificación sólo depende de ese vecindario local para clasificar, no hay ganancia en la distancia de Fermat.
- ¡Pero tampoco hay pérdida si se elige mal `n_neighbors`! 🤷

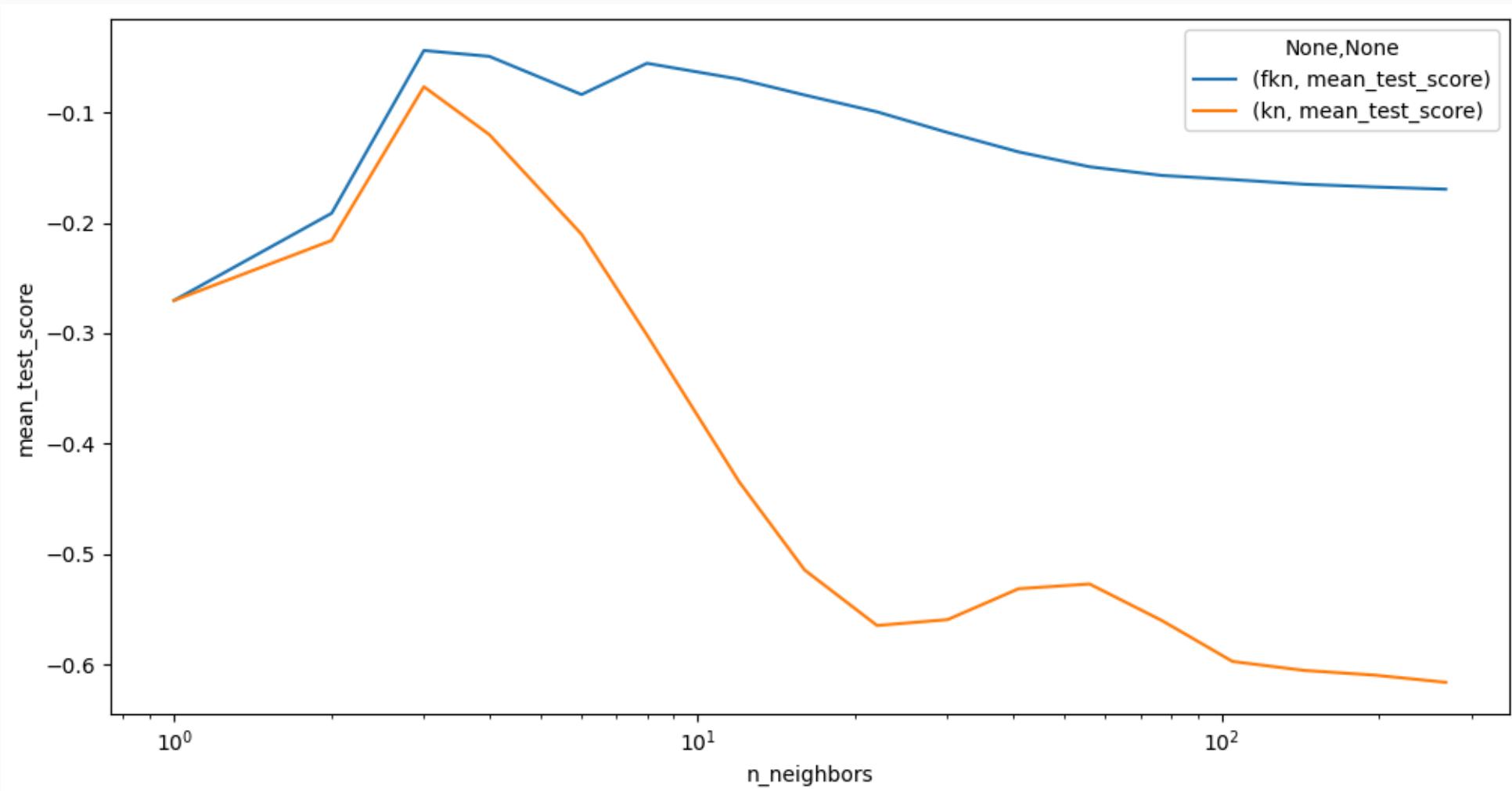
# $R^2$ por semilla para (F)KN en helices\_0



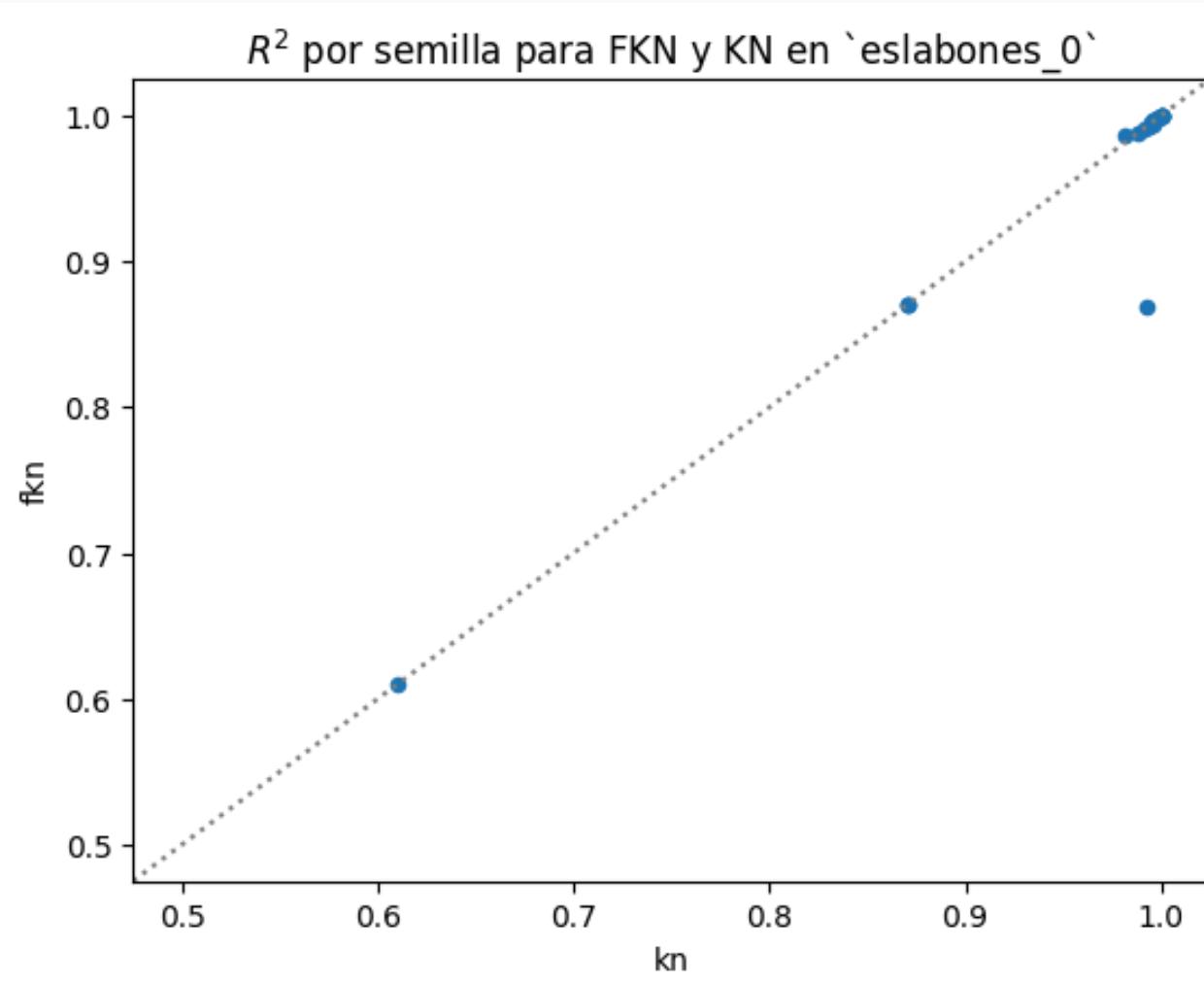
## $R^2$ y $\alpha^*$ para (F)KN en helices\_0, n\_neighbors seleccionados

	fkn	kn	
	param_alpha	mean_test_score	mean_test_score
param_n_neighbors			
1	3.50	-0.270327	-0.270327
3	1.75	-0.043833	-0.076631
12	4.00	-0.069771	-0.434716
41	4.00	-0.135744	-0.531532
144	4.00	-0.165027	-0.605601

# Mejor $R^2$ para (F)KN en helices\_0, en función de n\_neighbors



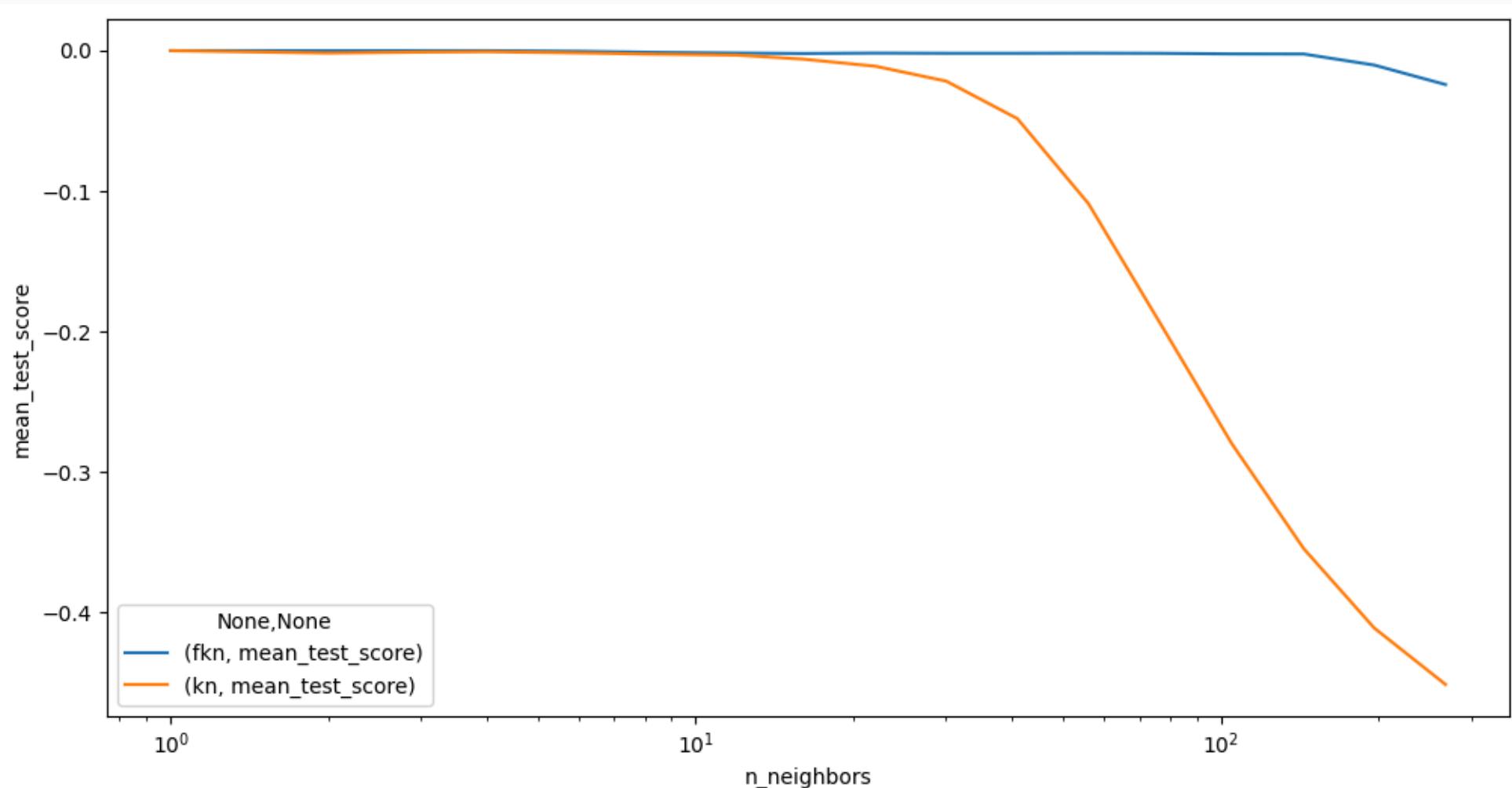
## $R^2$ por semilla para (F)KN en eslabones\_0



## $R^2$ y $\alpha^*$ para (F)KN en eslabones\_0, n\_neighbors seleccionados

	fkn	kn	
	param_alpha	mean_test_score	mean_test_score
param_n_neighbors			
1	1.00	-0.000	-0.000
3	3.75	-0.000	-0.001
12	3.75	-0.002	-0.003
41	3.00	-0.002	-0.048
144	4.00	-0.002	-0.355

# Mejor $R^2$ para (F)KN en eslabones\_0, en función de n\_neighbors



# Otros datasets: 2D mucho ruido

lunas\_hi (acc: 81.44%)

	delta_acc	r2
clf		
gbt	-0.21	38.14
fkdc	-0.33	37.46
fkn	0.00	36.57
kn	-0.02	36.50
kdc	-0.47	35.67
slr	-1.21	33.75
lr	-1.73	33.70
gnb	-1.98	33.34
base	-32.90	0.00
svc	-1.18	NaN

circulos\_hi (acc: 65.93%)

	delta_acc	r2
clf		
gbt	-0.23	9.49
fkdc	-3.82	6.98
kdc	-4.78	5.54
kn	-4.87	5.14
fkn	-5.41	5.12
gnb	-5.34	3.33
base	-17.39	0.00
lr	-17.39	-0.00
slr	-17.39	-0.00
svc	0.00	NaN

espirales\_hi (acc: 85.54%)

	delta_acc	r2
clf		
fkdc	-1.86	44.41
kdc	-1.98	43.45
fkn	-2.57	40.36
kn	-3.06	39.93
gbt	-14.68	15.22
gnb	-32.64	0.33
lr	-34.95	0.17
slr	-34.97	0.17
base	-35.79	0.00
svc	0.00	NaN

# Otros datasets: 15D

pionono\_12 (acc: 91.07%)

	delta_acc	r2
clf		
gbt	0.00	78.96
gnb	-20.78	54.30
slr	-28.42	42.42
lr	-28.56	42.22
fkdc	-46.30	10.18
kdc	-41.83	10.05
fkn	-44.47	9.99
kn	-44.15	9.95
base	-68.32	0.00
svc	-29.07	NaN

eslabones\_12 (acc: 98.48%)

	delta_acc	r2
clf		
gbt	0.00	91.66
gnb	-8.67	75.24
fkdc	-22.66	24.58
fkn	-21.98	24.43
kdc	-22.40	24.32
kn	-22.52	24.11
lr	-31.16	21.01
slr	-30.98	20.69
base	-48.73	0.00
svc	-19.20	NaN

helices\_12 (acc: 53.15%)

	delta_acc	r2
clf		
fkn	-0.99	0.20
kn	-0.99	0.20
gnb	-1.01	0.13
base	-3.40	0.00
lr	-3.40	0.00
slr	-3.40	0.00
gbt	0.00	-0.37
fkdc	-3.41	-2.69
kdc	-3.45	-6.01
svc	-2.92	NaN

hueveras\_12 (acc: 53.55%)

	delta_acc	r2
clf		
kn	-0.87	0.20
fkn	-0.99	0.17
gnb	-1.30	0.14
lr	-3.81	0.02
base	-3.80	0.00
slr	-3.80	0.00
gbt	0.00	-0.62
fkdc	-3.74	-10.75
kdc	-3.97	-15.08
svc	-3.69	NaN

# Otros datasets: multiclase

iris (acc: 96.27%)

	delta_acc	r2
clf		
lr	0.00	88.64
slr	-0.53	87.99
gbt	-1.49	85.88
gnb	-4.05	80.93
fkdc	-0.96	79.30
kdc	-1.17	78.02
kn	-0.48	74.39
fkn	-0.64	73.21
base	-65.87	0.00
svc	-1.65	NaN

vino (acc: 97.3%)

	delta_acc	r2
clf		
gbt	-0.85	89.80
slr	0.00	89.72
gnb	-2.70	84.59
lr	-10.92	67.06
fkn	-27.55	45.22
kn	-27.55	45.22
fkdc	-30.56	42.44
kdc	-30.92	39.42
base	-58.83	0.00
svc	-8.94	NaN

pinguinos (acc: 99.13%)

	delta_acc	r2
clf		
slr	0.00	96.22
lr	-0.77	95.51
gbt	-1.80	91.85
gnb	-4.68	84.81
fkn	-22.67	49.42
kn	-22.67	49.36
fkdc	-26.55	41.60
kdc	-26.50	40.49
base	-56.16	0.00
svc	-3.67	NaN

anteojos (acc: 97.68%)

	delta_acc	r2
clf		
fkdc	-0.08	92.89
kdc	0.00	91.84
kn	-0.16	89.32
fkn	-0.17	87.08
gbt	-1.36	86.12
gnb	-0.62	85.17
lr	-55.51	27.80
slr	-56.00	27.60
base	-48.34	0.00
svc	-0.26	NaN

# Otros datasets: dígitos y mnist

dígitos (acc: 98.41%)

	delta_acc	r2
clf		
fkdc	0.00	97.67
kdc	-0.10	96.80
gbt	-2.43	94.17
lr	-2.24	93.38
slr	-2.33	93.10
fkn	-1.98	92.22
kn	-2.91	90.62
gnb	-8.42	85.67
base	-89.43	0.00
svc	-0.16	NaN

mnist (acc: 87.07%)

	delta_acc	r2
clf		
kdc	-4.04	76.38
lr	-4.32	76.10
fkdc	-4.38	73.38
gbt	-10.11	66.57
slr	-12.33	61.43
gnb	-19.10	56.91
fkn	-16.28	54.02
kn	-19.10	50.40
base	-76.57	0.00
svc	0.00	NaN