TALLER · DATOSTADA 2026

Buscador de imágenes con IA

Embeddings, similitud y clasificación con pocos datos



Rodolfo Ferro · rodolfoferro.xyz · @rodo_ferro



QUIÉN IMPARTE

Rodolfo Ferro

- Head of AI en Vira Doctores
- Docente de IA en EdgeHub y en el Diplomado en Ciencia de Datos de la ENES UNAM León
- Especialista en Métodos Estadísticos (CIMAT); Maestría en IA Aplicada (Tec de Monterrey)
- Charlas en PyCon Colombia, Talent Land y TEDxUNAMLeón

rodolfoferro.xyz · @rodo_ferro · github.com/RodolfoFerro

ABRE EL NOTEBOOK AHORA

rodolfoferro.xyz/buscador-imagenes-notebook

Se abre directo en Google Colab. Funciona con CPU; no necesitas instalar nada.

MIENTRAS PLATICAMOS

Ejecuta el bloque 0 y la celda de descarga del bloque 1 (Shift + Enter). Así EuroSAT y CLIP se descargan mientras avanzamos.

LA PREGUNTA GUÍA

¿Basta un modelo de IA general para entender imágenes de un dominio específico, o necesitamos nuestros propios datos?

Vota: **A)** basta un modelo general · **B)** necesitamos datos propios

LO QUE VAMOS A CONSTRUIR

Un sistema de visión por computadora, sin entrenar ninguna red neuronal

01

Ver

Un mapa del dataset completo en dos dimensiones.

02

Buscar

Imágenes parecidas a una imagen o a una frase.

03

Clasificar

Solo con texto (*zero-shot*), y medir hasta dónde llega.

04

Mejorar

Con muy pocos ejemplos etiquetados.

AGENDA · 2 HORAS

Siete bloques, tres ejercicios

Bloque	Tema	Tiempo	Ejercicio
1	La imagen como dato	10 min	
2	¿Qué es un embedding?	15 min	
3	Un mapa del dataset: PCA y UMAP	15 min	
4	Buscador de imágenes similares	15 min	Ejercicio 1
5	Clasificación zero-shot con texto	15 min	Ejercicio 2
6	Clasificación con pocos datos	20 min	Ejercicio 3
7	Búsqueda con texto y cierre	10 min	

CÓMO LEER EL NOTEBOOK

Cuatro tipos de recuadro para repasar en casa

Definición

Qué es un concepto nuevo.

Cómo funciona

El proceso detrás de un método, paso a paso.

Intuición

Una analogía para fijar la idea.

Cuidado

Errores comunes al interpretar un resultado.

Además: 11 diagramas, tres **ejercicios** con solución propuesta y un glosario al final.

01

La imagen como dato

Qué es una imagen satelital, qué ve la computadora y por qué los pixeles no bastan.

EL DATASET · EUROSAT

Imágenes del satélite Sentinel-2, del programa europeo Copernicus

27,000
imágenes etiquetadas

10
clases de uso de suelo

64 px
por lado: 64 × 64 pixeles

10 m
de terreno por cada pixel

Cada imagen cubre 640 × 640 m, unas 57 canchas de futbol. Usaremos una muestra balanceada: 200 imágenes por clase.

Helber et al. (2019), EuroSAT · licencia MIT

DEFINICIÓN

Un satélite no toma fotos: mide bandas espectrales

Una **banda espectral** es un rango de longitudes de onda de la luz. El sensor mide cuánta luz refleja la superficie en cada banda.

Sentinel-2 mide **13 bandas**: las tres que vemos (rojo, verde y azul) y otras en el infrarrojo, que revelan, por ejemplo, qué tan sana está la vegetación.

IMAGEN RGB

La combinación de solo las tres bandas visibles. Por eso se parece a una foto común. Es la versión que usaremos.

PERCEPCIÓN REMOTA

La disciplina que analiza la Tierra con sensores a distancia.

LAS 10 CLASES

¿Cuáles te costaría distinguir a ti?


Cultivo anual


Bosque


Vegetación herbácea


Autopista


Industrial


Pastizal


Cultivo permanente


Residencial


Río


Mar o lago

DEFINICIÓN

Una imagen es un arreglo de números

```
img = np.array(Image.open(ruta))
img.shape # (64, 64, 3)
img.dtype # uint8
# 64 × 64 × 3 = 12,288 números
```

Alto × ancho × 3 canales. Cada valor va de **0** (sin intensidad) a **255** (intensidad máxima).

PIXEL

El punto más pequeño de la imagen. Aquí, un cuadro de 10 × 10 m de terreno.

¿POR QUÉ 0 A 255?

Cada valor ocupa 8 bits: $2^8 = 256$ valores posibles (uint8).

LOS TRES CANALES

Cada canal aporta solo la luz de un color

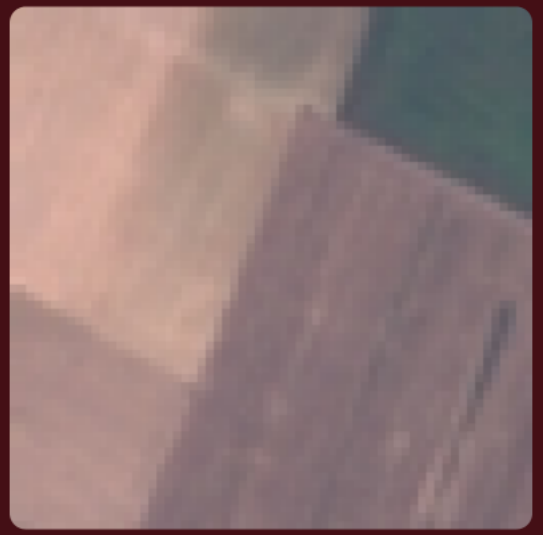
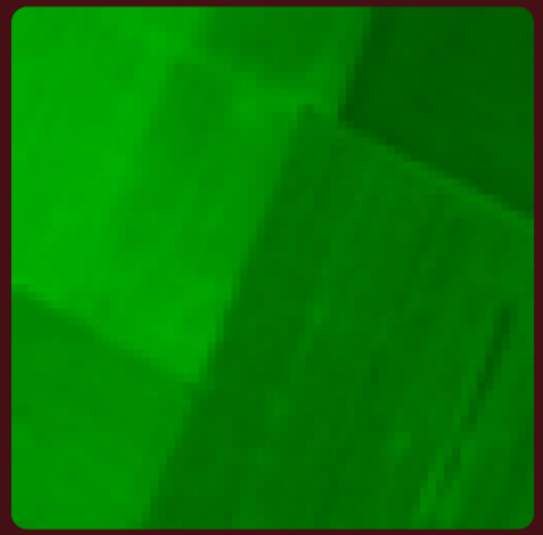


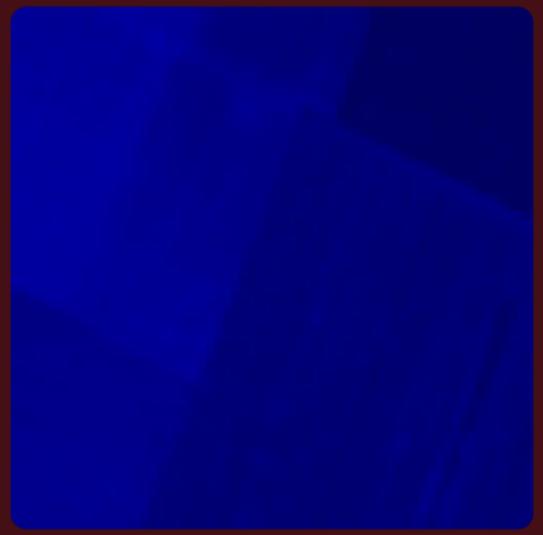
Imagen RGB



Solo rojo · img[:, :, 0]



Solo verde · img[:, :, 1]



Solo azul · img[:, :, 2]

Los otros dos canales se ponen en cero. Al sumar las tres capas recuperamos **exactamente** la imagen original.

EL PROBLEMA

Si movemos la imagen un solo pixel, casi todos sus 12,288 números cambian, aunque el terreno sea el mismo.

Necesitamos una representación del **contenido**, no del valor exacto de cada pixel. Eso es un *embedding*.

02

¿Qué es un *embedding*?

Cómo un modelo preentrenado resume una imagen en 512 números, y cómo medimos qué tan parecidas son dos imágenes.

DEFINICIÓN · EMBEDDING

Un vector que resume el contenido

LA IDEA CLAVE

Datos con contenido parecido tienen vectores cercanos.

Para cualquier dato

Imágenes, textos, audio: todo se puede convertir en un vector.

Lo aprende un modelo

Nadie define a mano qué significa cada coordenada.

Importa la distancia

Los números sueltos no se interpretan; la cercanía entre vectores sí.

INTUICIÓN

Un *embedding* es como un mapa

Las ciudades cercanas tienen latitud y longitud parecidas. Un embedding hace lo mismo con cientos de coordenadas, y para **contenido** en lugar de ubicación.

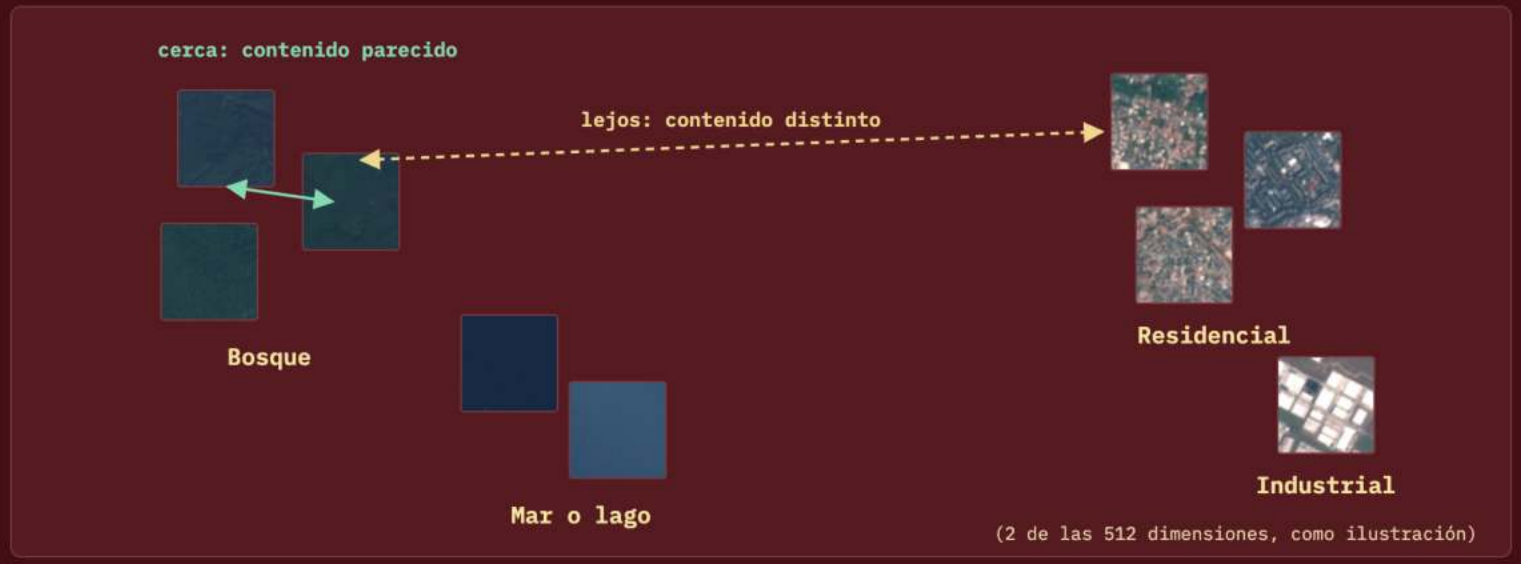


Ilustración en 2 de las 512 dimensiones

DE IMAGEN A VECTOR

12,288 pixeles entran, 512 números salen



Imagen: 64 × 64 × 3
12,288 números (pixeles)

512 números que resumen
el contenido de la imagen

Una reducción de 24 veces. Los números sueltos no se interpretan: no hay una “coordenada de bosque”.

DEFINICIÓN

Aprendizaje por transferencia

Reutilizar lo que aprendió un modelo en una tarea para resolver otra distinta.

Entrenar un modelo como CLIP requiere cientos de millones de ejemplos y semanas de cómputo en cientos de GPUs. En la práctica casi nunca se entrena desde cero.

LO QUE HAREMOS HOY

No vamos a entrenar CLIP. Lo descargamos ya entrenado y solo lo usamos para **transformar** nuestras imágenes, igual que un StandardScaler ya ajustado al que solo le llamamos transform.

EL MODELO · CLIP

Imágenes y textos en el mismo espacio

CLIP (*Contrastive Language-Image Pre-training*, OpenAI, 2021) tiene dos codificadores que producen vectores comparables entre sí.

Imagen → vector

Codificador de imágenes

Convierte cada foto en un vector de 512 números.

Texto → vector

Codificador de texto

Convierte una frase en otro vector de 512.

Consecuencia

Comparables

La foto de un bosque queda cerca de *“a photo of a forest”*.

`openai/clip-vit-base-patch32 · 151 millones de parámetros · Hugging Face`

Aprendizaje contrastivo

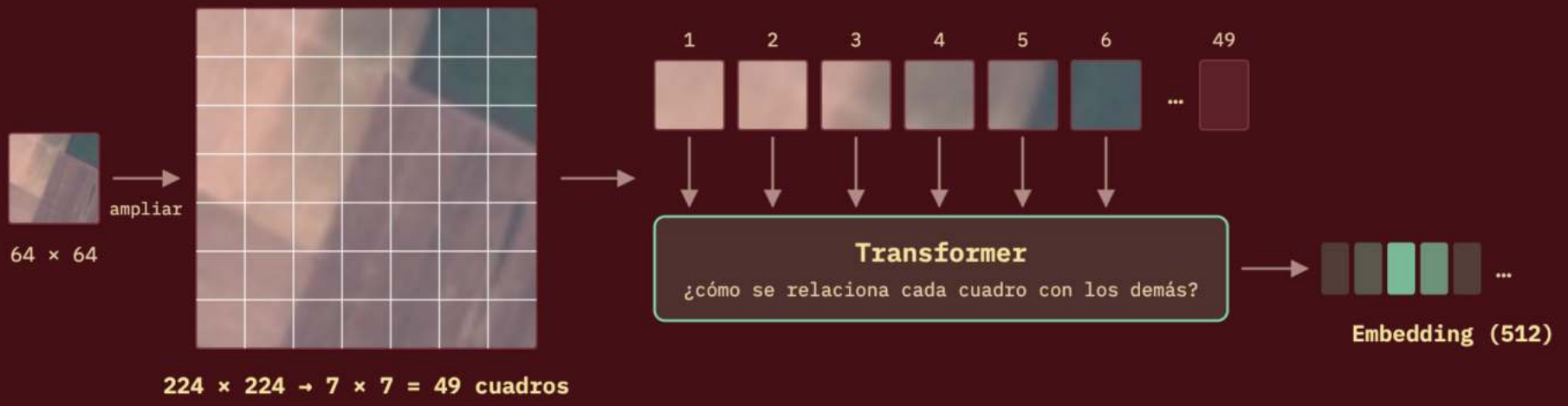
1. Un lote de pares (imagen, texto) de internet, por ejemplo una foto y su pie de foto.
2. Cada codificador convierte su parte en un vector.
3. Se ajusta el modelo para **acercar** cada foto a su texto y **alejarla** de los demás.



Entrenado con unos 400 millones de pares imagen-texto

DEFINICIÓN · CLIP-VIT-BASE-PATCH32

Así ve una imagen el *Vision Transformer*



ViT: corta la imagen en cuadros (*patches*) y analiza cómo se relacionan.

base: el tamaño del modelo; hay versiones más pequeñas y más grandes.

patch32: cuadros de 32 × 32; 224 × 224 da 7 × 7 = 49.

Dosovitskiy et al. (2021), An Image is Worth 16x16 Words

EN EL NOTEBOOK

Todo el dataset a vectores, en unas líneas

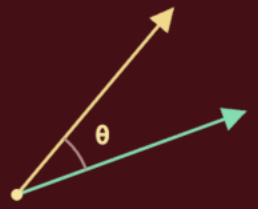
```
modelo = CLIPModel.from_pretrained("openai/clip-vit-base-patch32").eval()
procesador = CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32")
@torch.no_grad() # no vamos a entrenar: no necesitamos gradientes
def embeddings_de_imagenes(rutas):
    imagenes = [Image.open(r).convert("RGB") for r in rutas]
    entrada = procesador(images=imagenes, return_tensors="pt")
    v = modelo.get_image_features(**entrada)
    return torch.nn.functional.normalize(v, dim=-1).numpy()
X = embeddings_de_imagenes(df["ruta"]) # (2000, 512)
```

El paso más lento del taller: unos **30 segundos** en CPU para las 2,000 imágenes. Normalizamos cada vector a longitud 1.

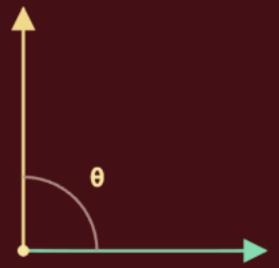
DEFINICIÓN · SIMILITUD COSENO

Lo que importa es la dirección, no la longitud

$$\text{sim}(a, b) = (a \cdot b) / (\|a\| \, \|b\|) = \cos \theta$$



θ pequeño
sim ≈ 0.9
parecidos



θ = 90°
sim = 0
sin relación



θ = 180°
sim = −1
opuestos

Con vectores de longitud 1 basta el **producto punto**: X @ X.T da la similitud entre todos los pares a la vez.

DEFINICIÓN · MEDIDAS DE SIMILITUD

Tres formas de comparar dos vectores

Medida	Fórmula	Qué mide	Parecidos si…
Producto punto	$a \cdot b$	Dirección y longitud	es grande
Similitud coseno	$a \cdot b / (\|a\| \|b\|)$	Solo la dirección	se acerca a 1
Distancia euclidiana	$\|a - b\|$	Distancia en línea recta	se acerca a 0

¿POR QUÉ COSENO?

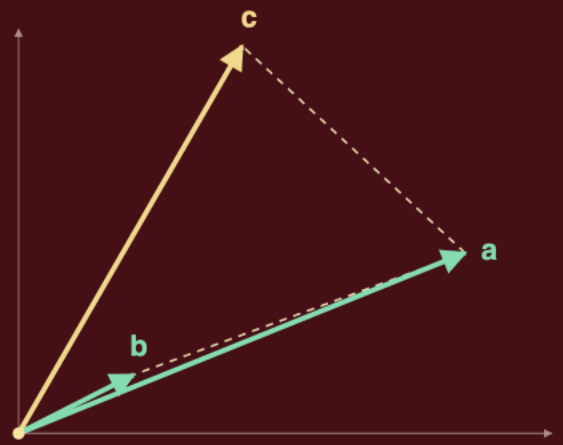
En un embedding, la longitud del vector aporta poco: lo que describe el contenido es hacia **dónde apunta**. Por eso normalizamos y comparamos ángulos.

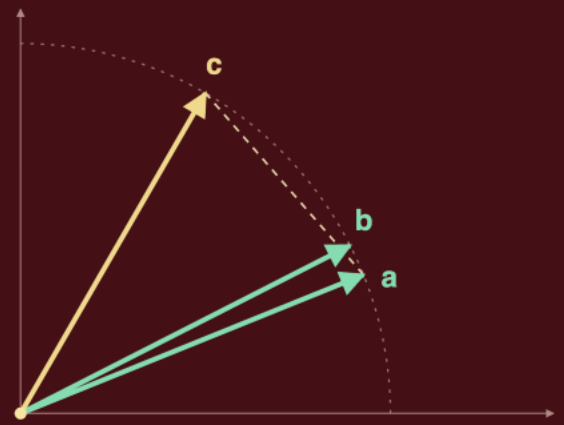
La similitud es lo opuesto de una distancia: más similitud, menos distancia

Normalizar pone todo de acuerdo

Sin normalizar: la euclidiana ve a a más cerca de c; el coseno, de b.

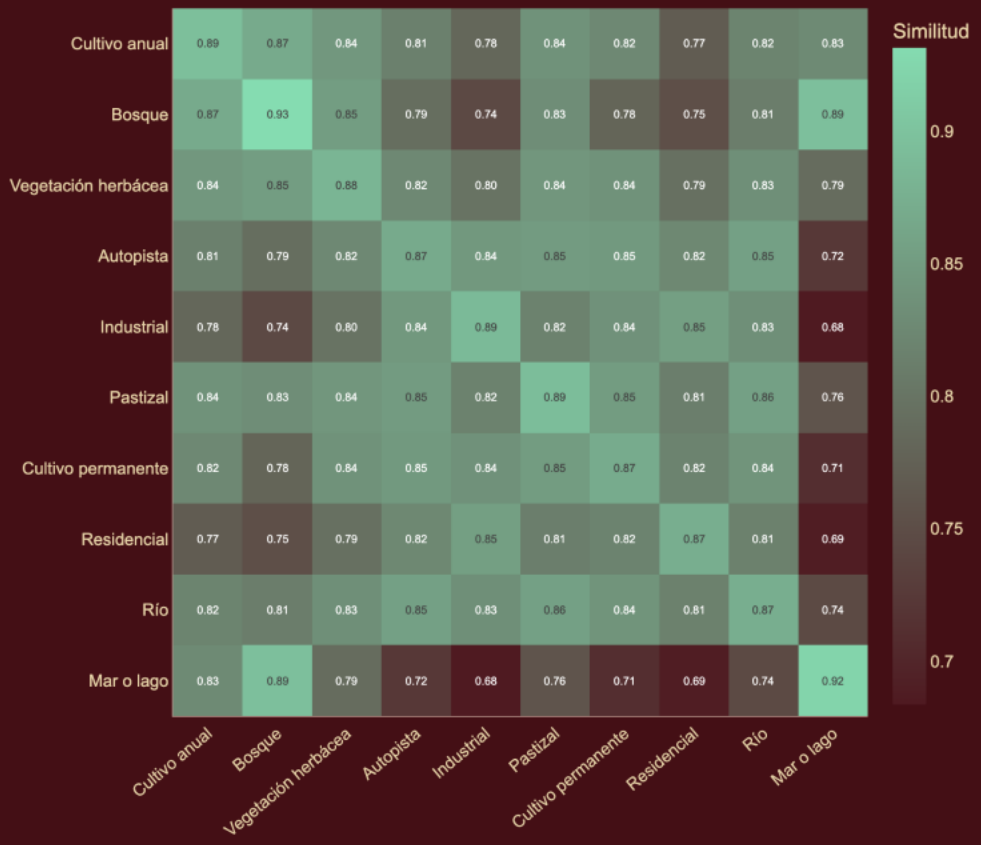
Normalizados (longitud 1): las dos eligen a b.





$$\|a - b\|^2 = 2 - 2 \cdot \cos \theta$$

Con vectores de longitud 1, menor distancia equivale a mayor coseno



RESULTADO

Las clases se ordenan solas

0.889
misma clase

0.809
clase distinta

- El máximo de cada renglón está en la diagonal.
- Todo es alto: importa lo **relativo**.
- Bosque y Mar o lago: 0.89.

Similitud coseno promedio entre clases · 2,000 imágenes

03

Un mapa del dataset: PCA y UMAP

No podemos dibujar 512 dimensiones, pero sí proyectarlas a 2 y ver si las clases se separan solas.

DEFINICIÓN

Reducción de dimensionalidad

De muchas columnas a muy pocas, perdiendo lo menos posible.

Lineal

PCA

Conserva las direcciones en las que los datos varían más. Como la sombra de un objeto sobre una pared.

No lineal

UMAP

Conserva quién es vecino de quién. Mejor para ver grupos, pero las distancias lejanas no significan mucho.

> Siempre se pierde algo: ninguna proyección a 2 dimensiones conserva todo lo que hay en 512.

CÓMO FUNCIONA · PCA

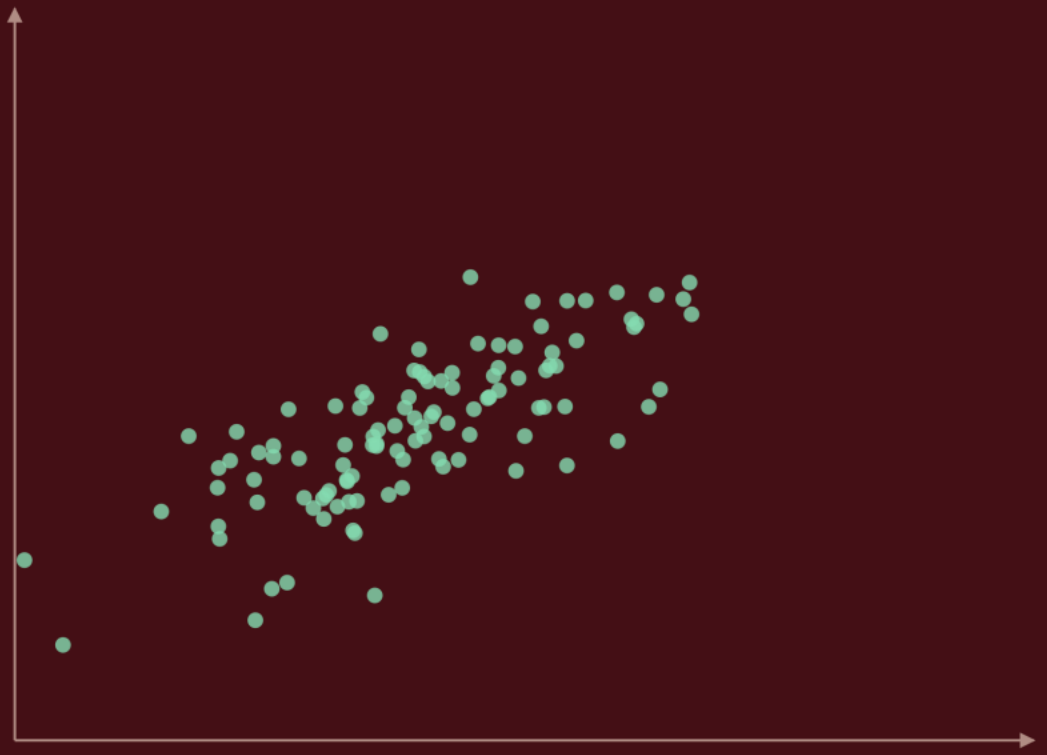
Paso a paso

1. Partimos de una nube de puntos

2. Centrar: restar el promedio

3. Buscar las direcciones de mayor dispersión

4. Proyectar sobre ellas



CÓMO FUNCIONA · PCA

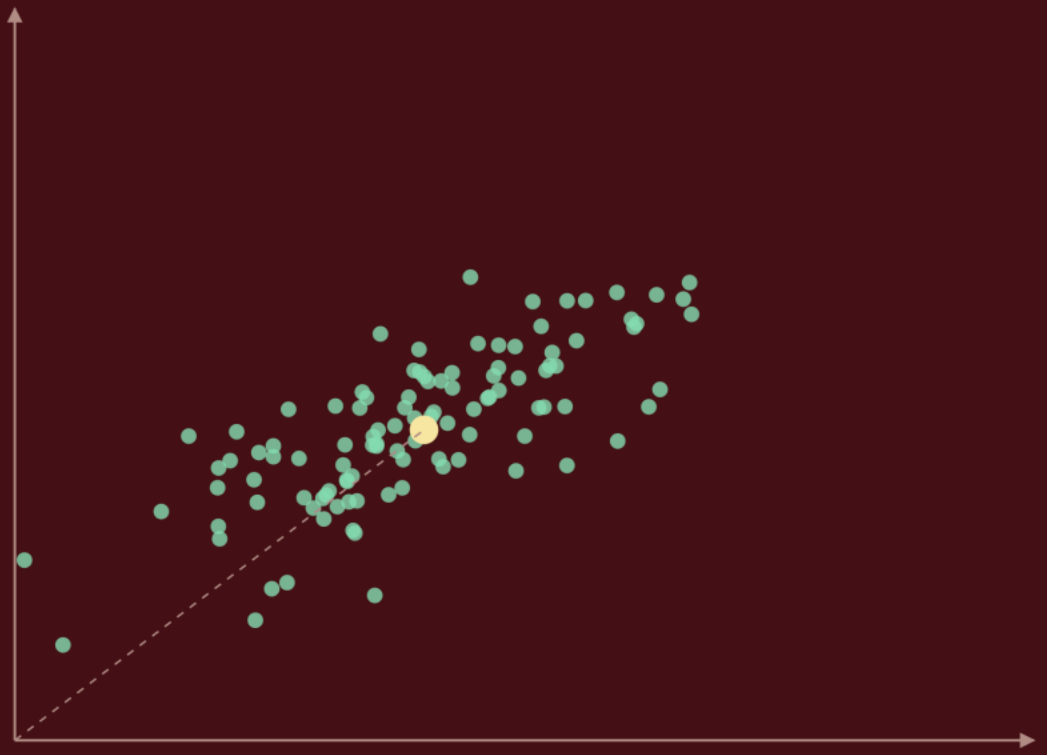
Paso a paso

1. Partimos de una nube de puntos

2. Centrar: restar el promedio

3. Buscar las direcciones de mayor dispersión

4. Proyectar sobre ellas

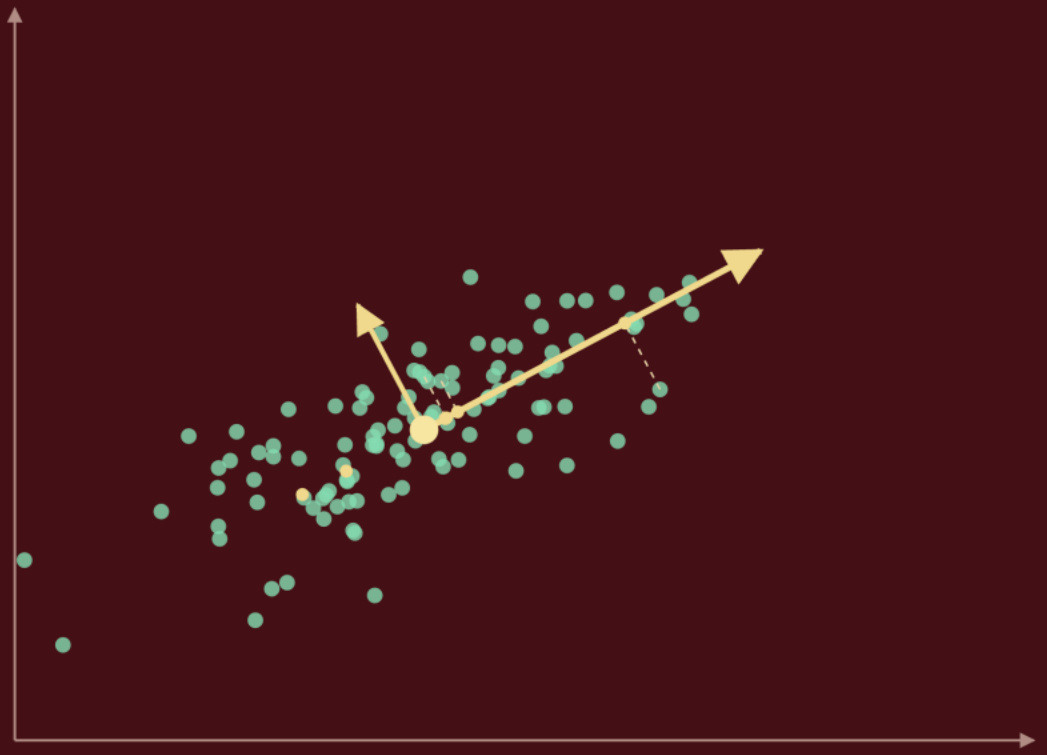


CÓMO FUNCIONA · PCA

Paso a paso

1. Partimos de una nube de puntos
2. Centrar: restar el promedio
3. **Buscar las direcciones de mayor dispersión**
4. Proyectar sobre ellas

PC1 es la dirección en que la nube se estira más; PC2, la siguiente, **perpendicular**.

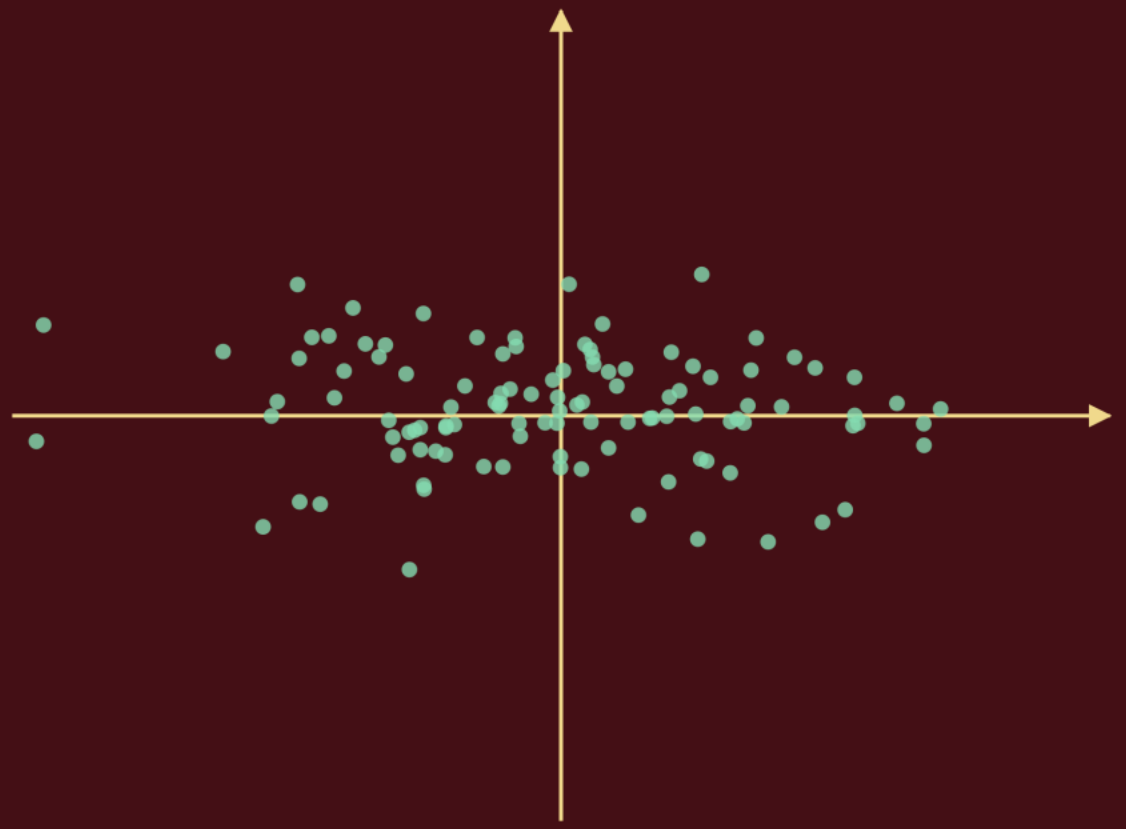


CÓMO FUNCIONA · PCA

Paso a paso

1. Partimos de una nube de puntos
2. Centrar: restar el promedio
3. Buscar las direcciones de mayor dispersión
4. **Proyectar sobre ellas**

La **varianza explicada** dice qué porcentaje de la dispersión conservan los componentes elegidos.

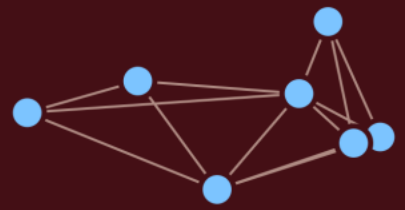


Paso a paso

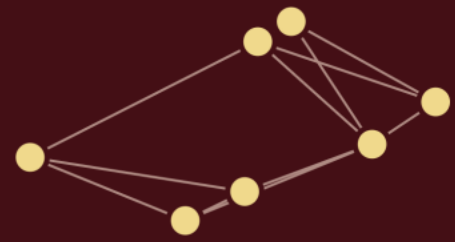
1. Conectar a cada punto con sus vecinos más cercanos (en 512-D)

2. Colocar los puntos al azar en 2-D

3. Acercar vecinos y alejar no vecinos







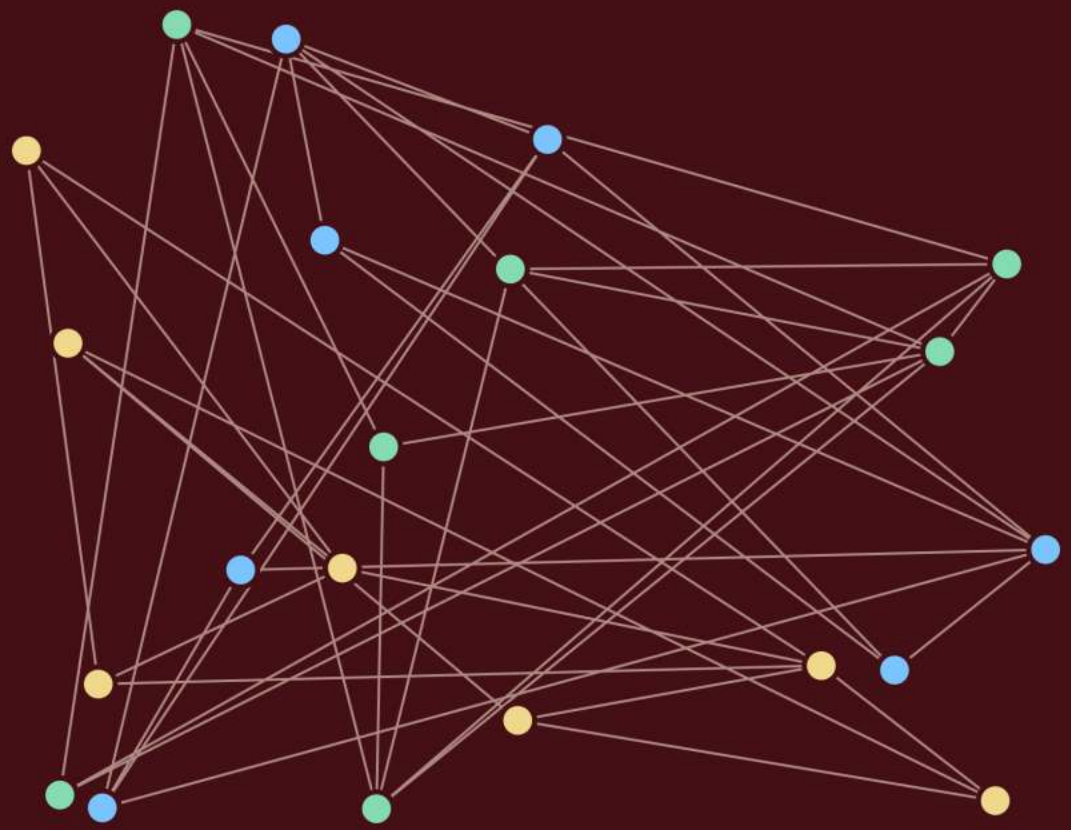
CÓMO FUNCIONA · UMAP

Paso a paso

1. Conectar a cada punto con sus vecinos más cercanos (en 512-D)

2. Colocar los puntos al azar en 2-D

3. Acercar vecinos y alejar no vecinos



CÓMO FUNCIONA · UMAP

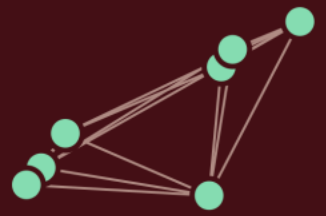
Paso a paso

1. Conectar a cada punto con sus vecinos más cercanos (en 512-D)

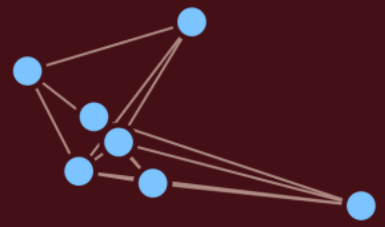
2. Colocar los puntos al azar en 2-D

3. Acercar vecinos y alejar no vecinos

Parámetros: n_neighbors (cuántos vecinos) y min_dist (qué tan juntos).







CUIDADO

Cómo leer un mapa de UMAP

Confiable

Qué puntos quedan juntos

Los vecinos en el mapa también lo son en 512-D.

No confiable

El tamaño de un grupo

Un grupo grande no es necesariamente más variado.

No confiable

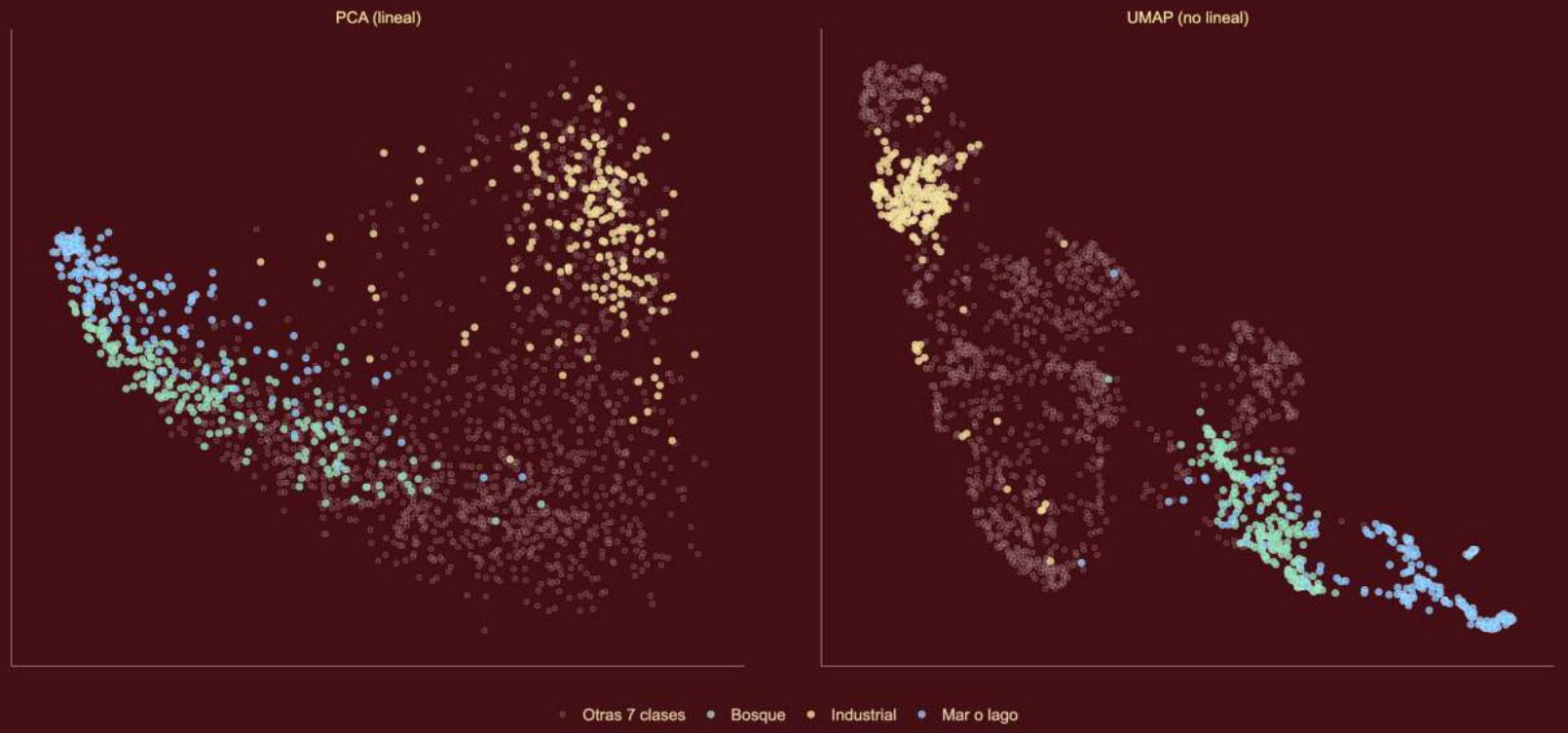
La distancia entre grupos lejanos

Dos islas lejanas no son “más distintas” que dos cercanas.

UMAP tiene una parte aleatoria: con random_state=42 es reproducible. Una técnica parecida y muy usada es **t-SNE**.

RESULTADO · LAS 2,000 IMÁGENES EN 2-D

UMAP separa los grupos mucho mejor que PCA



PCA conserva el 41.7% de la varianza · 3 de 10 clases resaltadas

RESULTADO · UNA CLASE POR PANEL

Autopista, Río y Pastizal quedan dispersos



Mapa UMAP · small multiples

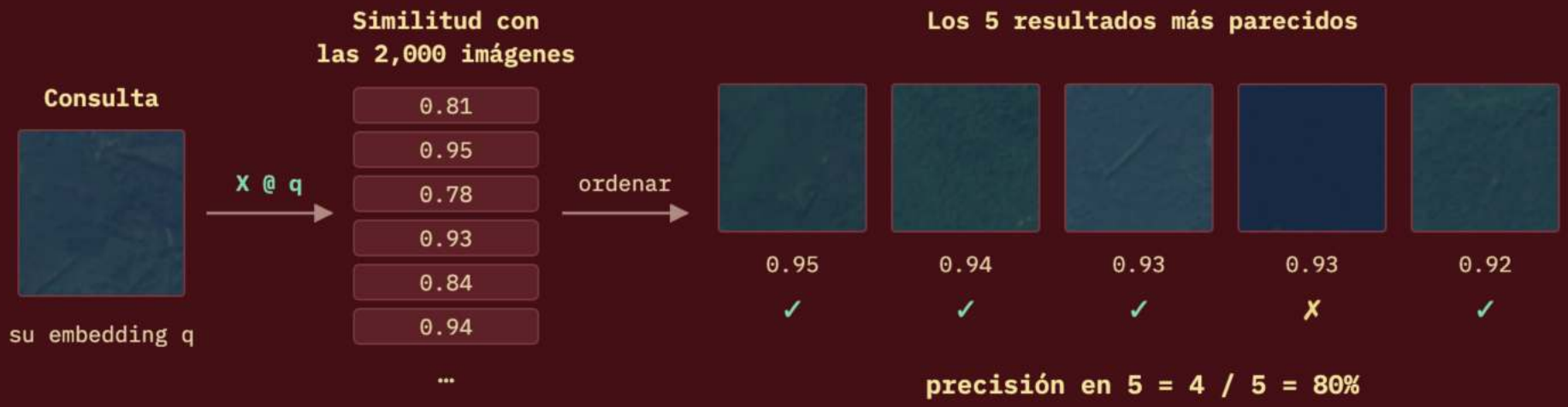
04

Un buscador de imágenes similares

Si las imágenes parecidas quedan cerca, buscar es encontrar a los vecinos más cercanos.

DEFINICIÓN · BÚSQUEDA POR VECINOS MÁS CERCANOS

Buscar es comparar y ordenar




```
similitudes = X @ X[i] · similitudes[i] = -np.inf · vecinos = np.argsort(-similitudes)[:5]
```

Esto es todo el buscador: un producto punto y un ordenamiento

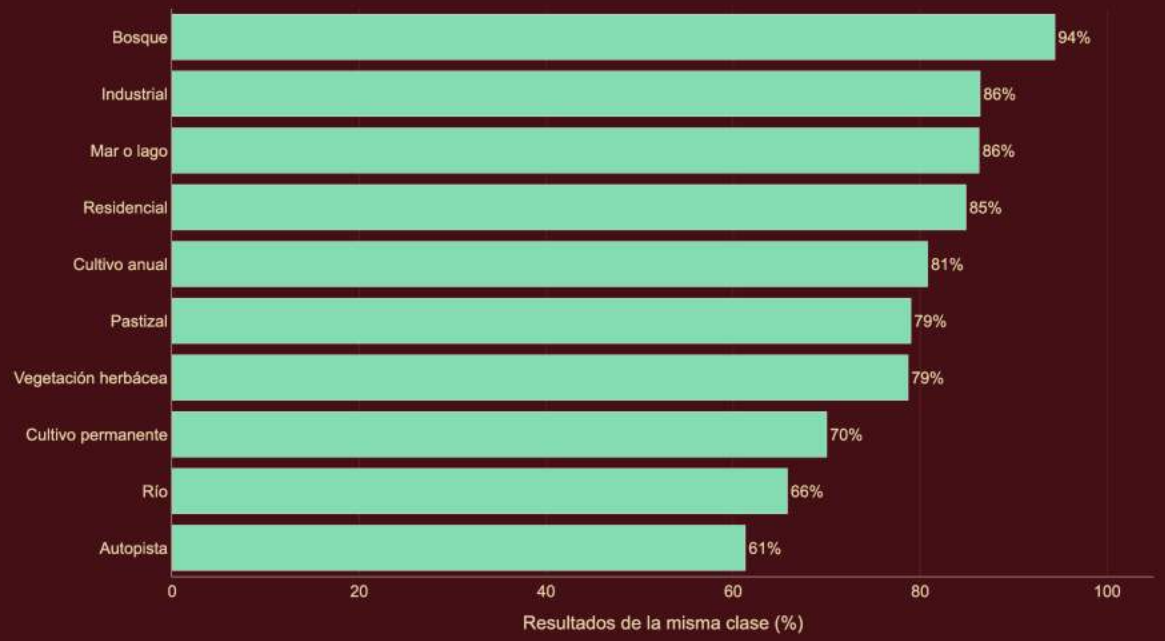
Funciona, y sus errores tienen sentido



Arriba: un río, 1 de 5 correctos · Abajo: residencial, 5 de 5 · la marca indica otra clase

DEFINICIÓN · PRECISIÓN EN K

¿Qué tan bueno es el buscador?



78.8%

de los 5 resultados son de la clase correcta (al azar: 10%)

EJEMPLO

Si 4 de los 5 resultados para un bosque son bosques, la precisión en 5 es 4 / 5 = 80%.

Promedio sobre las 2,000 consultas

¿DÓNDE SE USA?

La misma idea, a gran escala

Búsqueda por imagen

Sube una foto y encuentra otras parecidas.

Productos similares

Recomendaciones en tiendas en línea.

Galería de fotos

Escribe “playa” y aparecen tus fotos de playa.

Bases vectoriales

Asistentes de IA que buscan en documentos.

CON MILLONES DE VECTORES

Ya no se compara contra todo: se usan índices de **búsqueda aproximada** (por ejemplo, FAISS), que cambian un poco de precisión por mucha velocidad.

EJERCICIO 1 · 5 MINUTOS

¿Los errores del buscador tienen sentido?

Elige una imagen al azar y busca sus **8** imágenes más parecidas.

```
mostrar_busqueda(...)
# i: un índice entre 0 y 1,999
```

PARA PENSAR

¿Confunde clases que a ti también te costaría distinguir? Encuentra un error que te parezca “razonable”.

Solución propuesta en el notebook

EJERCICIO 1 · 5 MINUTOS

¿Los errores del buscador tienen sentido?

Elige una imagen al azar y busca sus **8** imágenes más parecidas.

```
mostrar_busqueda(i, k=8)
# i: un índice entre 0 y 1,999
```

PARA PENSAR

¿Confunde clases que a ti también te costaría distinguir? Comparte un error que te parezca “razonable”.

Solución propuesta en el notebook

05

Clasificación *zero-shot* con texto

Ya sabemos buscar; ahora vamos a ponerle nombre a lo que vemos, sin ningún ejemplo etiquetado.

DEFINICIÓN

Zero-shot, *few-shot* y supervisado

Los tres escenarios que vamos a comparar, según cuántos ejemplos etiquetados necesita cada uno.

Término	Ejemplos etiquetados	En este taller
Zero-shot	Ninguno: solo una descripción de cada clase	Clasificar con frases (bloque 5)
Few-shot	Unos cuantos por clase (1, 5, 10…)	La curva de aprendizaje (bloque 6)
Supervisado	Muchos por clase	150 imágenes por clase (bloque 6)

DEFINICIÓN · ENTRENAMIENTO Y PRUEBA

Evaluar con imágenes que el modelo no vio

1,500

entrenamiento: 150 por clase, para aprender

500

prueba: 50 por clase, solo para medir

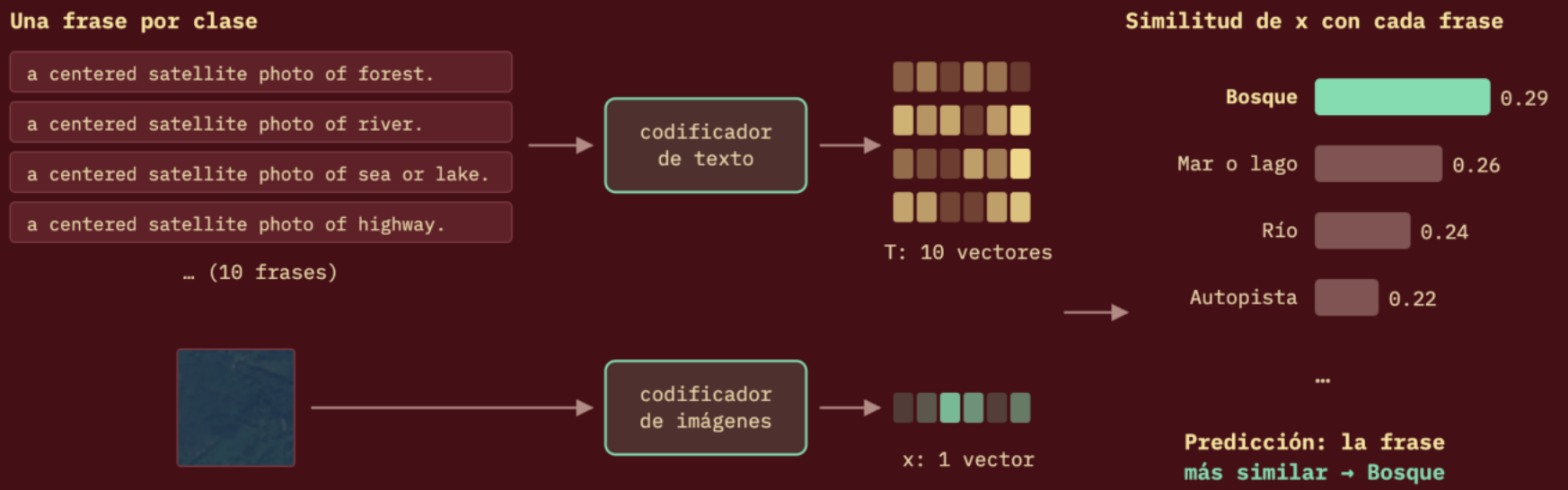
Si evaluamos con datos que el modelo ya vio, medimos qué tan bien **memoriza**, no qué tan bien **generaliza**.

La división es **estratificada**: cada clase conserva la misma proporción en ambos conjuntos.

```
train_test_split(…, test_size=0.25, stratify=y, random_state=42)
```

CÓMO FUNCIONA · ZERO-SHOT

Gana la frase más parecida



1. Una frase por clase · **2.** El embedding de cada frase · **3.** A cada imagen, la clase de la frase más similar

Las frases van en inglés: CLIP se entrenó casi solo con texto en inglés

DEFINICIÓN · PROMPT

La forma de escribir la frase importa

Un **prompt** es el texto que le damos al modelo; una **plantilla** tiene un hueco {} para el nombre de la clase. Antes de ver los resultados: **¿qué exactitud esperan?**

Plantilla	Exactitud
{} (solo el nombre de la clase)	36.6%
a satellite photo of {}.	42.0%
a centered satellite photo of {}.	44.8%

Sin cambiar el modelo ni los datos, **+8 puntos**. Es mejor que el azar (10%), pero falla en más de la mitad.

Exactitud en las 500 imágenes de prueba

DEFINICIÓN · MÉTRICAS DE CLASIFICACIÓN

Cuatro formas de medir un clasificador

Exactitud

Proporción de predicciones correctas sobre el total.

Matriz de confusión

Clase real contra predicha. Fuera de la diagonal: qué se confunde con qué.

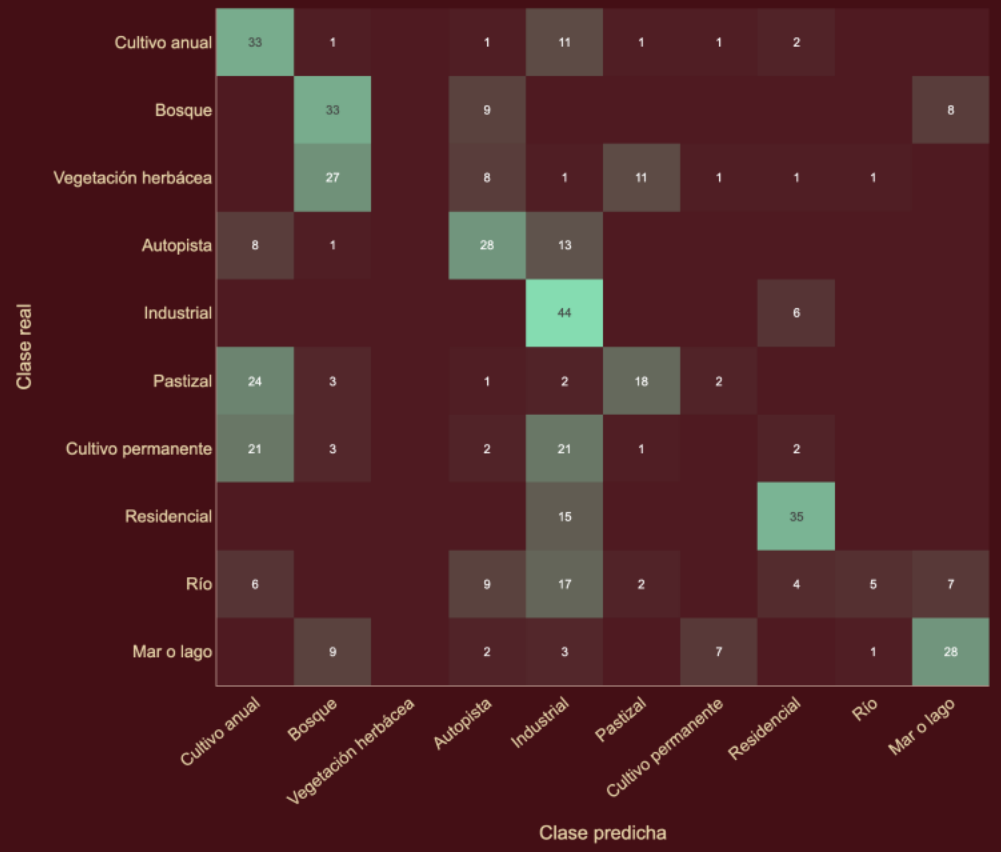
Recall

De las que **son** de la clase, ¿cuántas reconoció? Se lee por renglón.

Precisión

De las que **dijo** que son de la clase, ¿cuántas lo son? Por columna.

Con clases balanceadas la exactitud es un buen resumen, pero dos modelos con la misma exactitud pueden equivocarse de formas muy distintas.



RESULTADO · ZERO-SHOT, 44.8%

¿En qué se equivoca?

- **Industrial** recibe 127 predicciones, con solo 50 imágenes reales.
- **Vegetación herbácea** y **Cultivo permanente**: 0% de recall. Nunca las predice.
- **Río**: solo 10%, por la misma razón que en el buscador.

Sus nombres son términos técnicos que casi no aparecen en los textos de internet.

Renglón: clase real · columna: clase predicha

EJERCICIO 2 · 5 MINUTOS

Supera el 44.8% cambiando solo las frases

¿Cómo describirías una imagen de vegetación herbácea a alguien que no la ve?

```
zero_shot("tu plantilla con {}", X_test)
```

Solución: shrubland y an orchard or vineyard suben de 44.8% a **50.4%.**

CUIDADO · FUGA DE INFORMACIÓN

Elegimos esas frases **viendo el conjunto de prueba**: el 50.4% es optimista. En un proyecto real se ajustan con un conjunto de **validación** y la prueba se usa una sola vez, al final.

06

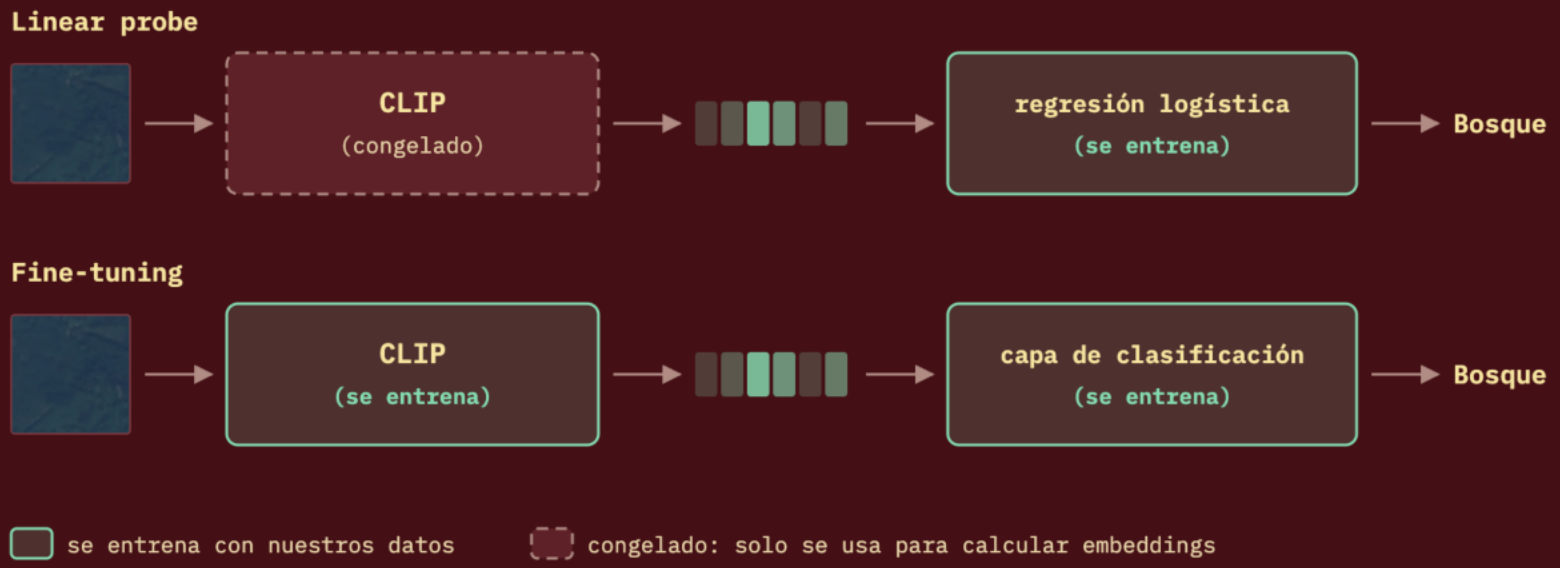
Clasificación con pocos datos

Un modelo general no es experto en imágenes satelitales. ¿Cuánto ayudan unos cuantos ejemplos de nuestros propios datos?

DEFINICIÓN · LINEAR PROBE

Un modelo sencillo encima

Cada imagen es ahora **un renglón con 512 columnas numéricas**: un problema tabular como cualquier otro.



Linear probe: rápido, barato y funciona con pocos datos

A pesar de su nombre, clasifica

1. Para cada clase, un **puntaje**: una suma ponderada de las 512 columnas.
2. **Softmax** convierte los 10 puntajes en probabilidades que suman 1.
3. Gana la clase con la mayor probabilidad.
4. Entrenar es encontrar los **pesos** que mejor separan las clases.

```
clf = LogisticRegression(max_iter=3000)
clf.fit(X_train, y_train)
clf.score(X_test, y_test)
```

Se entrena en un par de segundos, sin GPU. El parámetro C controla la regularización.

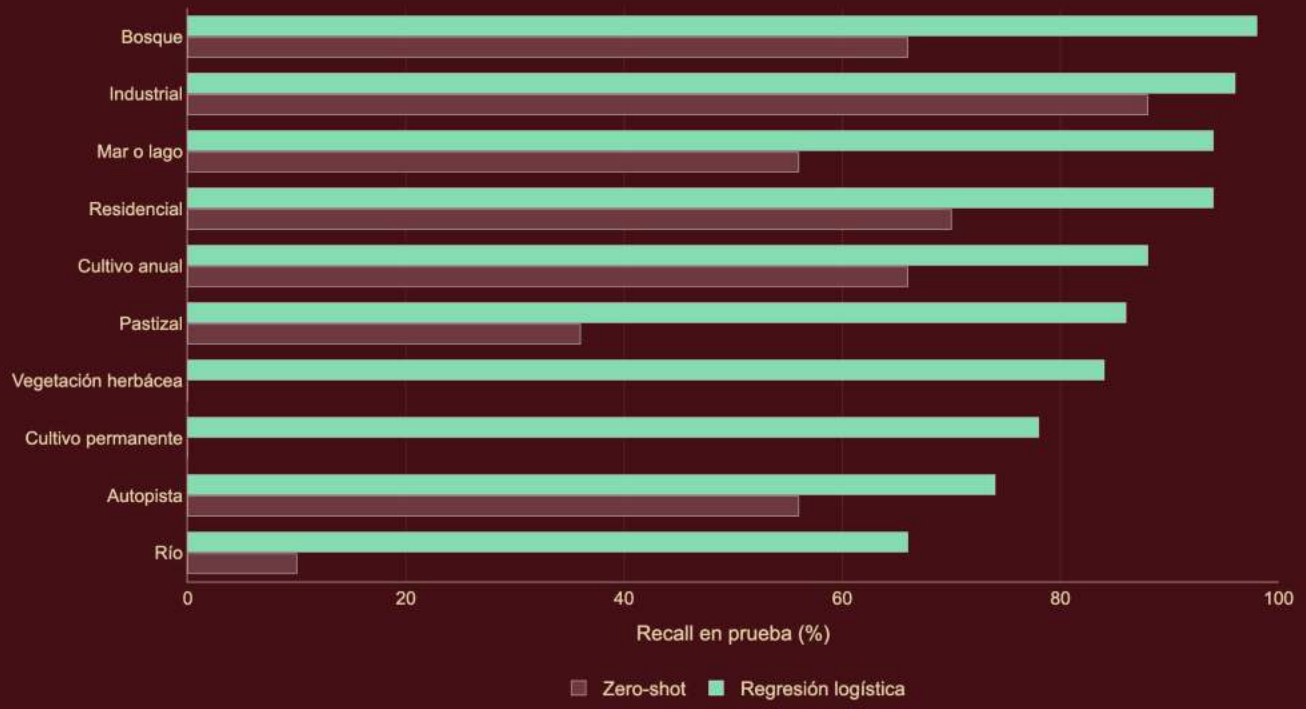
REGRESIÓN LOGÍSTICA SOBRE EMBEDDINGS · 150 IMÁGENES POR CLASE

44.8% → 85.8%

La exactitud casi se duplica, con un modelo que se entrena en segundos y sin GPU.

RESULTADO · RECALL POR CLASE

Ganan más las que CLIP no veía



0 → 84%

Vegetación herbácea

0 → 78%

Cultivo permanente

Río (66%) y Autopista (74%) siguen siendo las difíciles.

Gris: zero-shot · menta: regresión logística · 500 imágenes de prueba

DEFINICIÓN

Sobreajuste y subajuste

Sobreajuste

El modelo memoriza los datos de entrenamiento y falla con datos nuevos. Entrenamiento **mucho mayor** que prueba.

Subajuste

El modelo es demasiado simple para los patrones. Las dos exactitudes son parecidas, pero **bajas**.

86.7%
entrenamiento

85.8%
prueba

Casi iguales y altas: el modelo **no está memorizando**.

DEFINICIÓN · FEW-SHOT

Una tarea *N-way K-shot*

N clases, **K** ejemplos etiquetados de cada una (el **conjunto de soporte**) y una imagen nueva que clasificar (la **consulta**). Aquí: 10-way 1-shot.


Cultivo anual


Bosque


Veg. herbácea


Autopista


Industrial


Pastizal


Cultivo perm.


Residencial


Río


Mar o lago



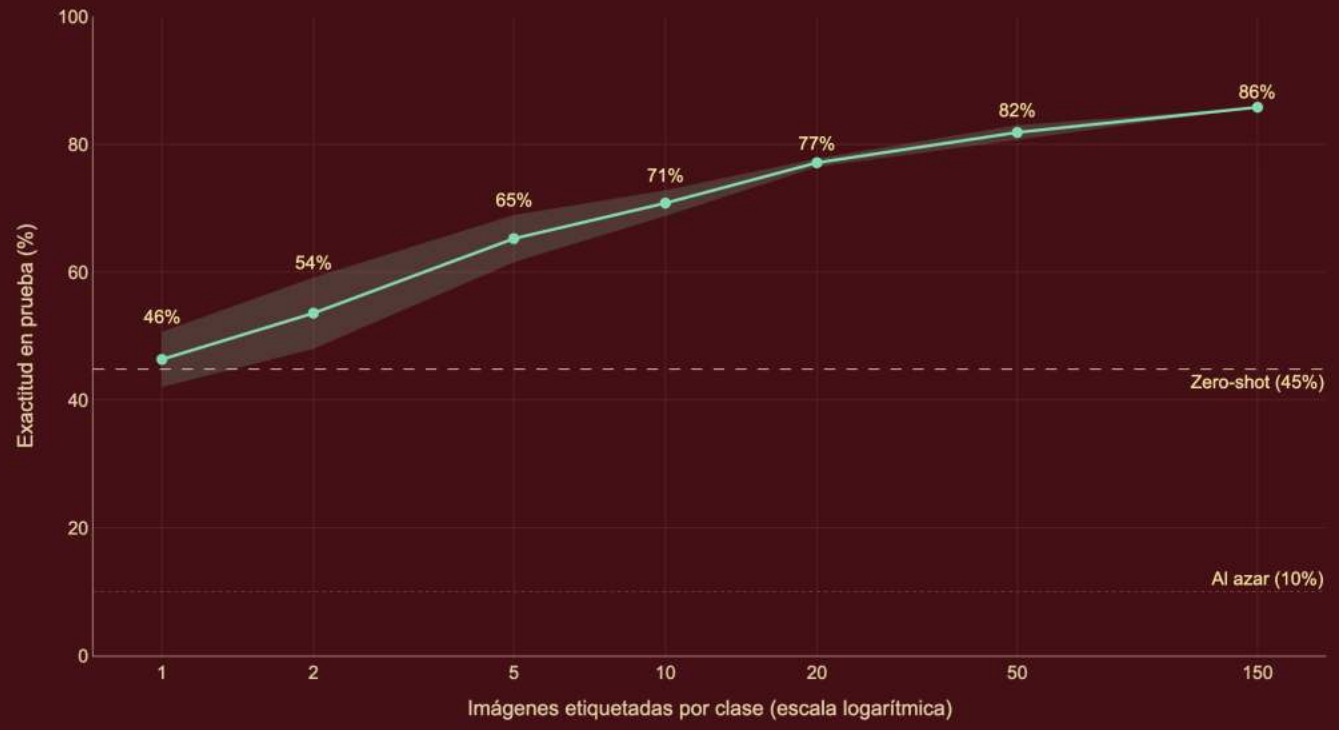
Consulta: ¿a cuál de las 10 clases pertenece?

Respuesta: **Bosque**, aunque nunca habíamos visto esta imagen.

Nuestra curva de aprendizaje es una tarea 10-way K-shot con K = 1, 2, 5, 10, 20, 50 y 150

DEFINICIÓN · CURVA DE APRENDIZAJE

¿Cuántos ejemplos necesitamos?



1 por clase ya iguala al zero-shot: 46.3% contra 44.8%.

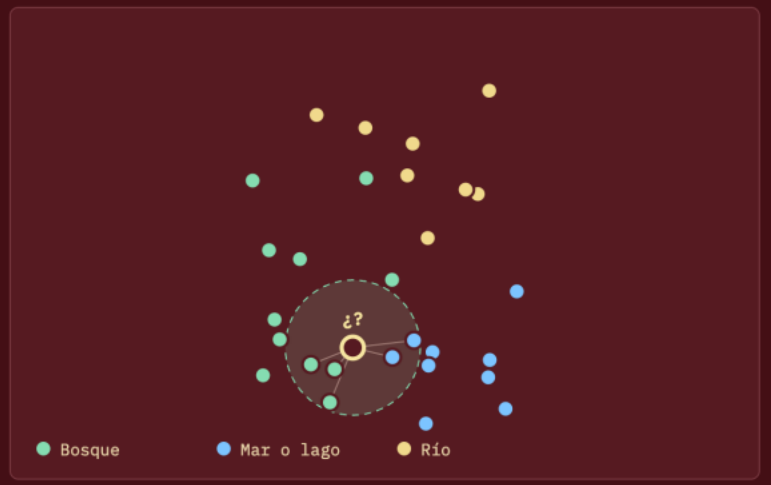
5 por clase: 65.2% · **20**: 77.1%.

Rendimientos decrecientes: de 20 a 150 solo suma 8.7 puntos.

Promedio de 10 repeticiones · banda: ± una desviación estándar

EJERCICIO 3 · K VECINOS MÁS CERCANOS

El buscador también es un clasificador



Los k = 5 vecinos más cercanos votan:

Bosque: 3

Mar o lago: 2

Predicción: Bosque

Es el buscador de la parte I más una votación.

k-NN = el buscador del bloque 4 + una votación.

Modelo	Exactitud
k-NN, k = 1	82.6%
k-NN, k = 5	85.0%
k-NN, k = 15	83.0%
Regresión logística	85.8%

```
KNeighborsClassifier(n_neighbors=k, metric="cosine")
```

LO QUE DICE LA INVESTIGACIÓN

Un buen embedding más un clasificador sencillo ya es un rival muy difícil de vencer.

Eso es justo lo que hicimos: CLIP aporta el embedding y la regresión logística, el clasificador.

Tian et al. (2020), Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need? · Chen et al. (2019), A Closer Look at Few-shot Classification

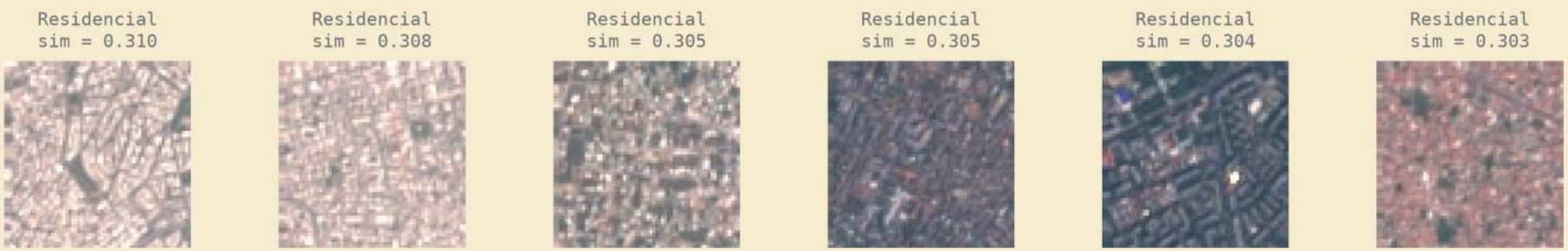
07

Búsqueda con texto y cierre

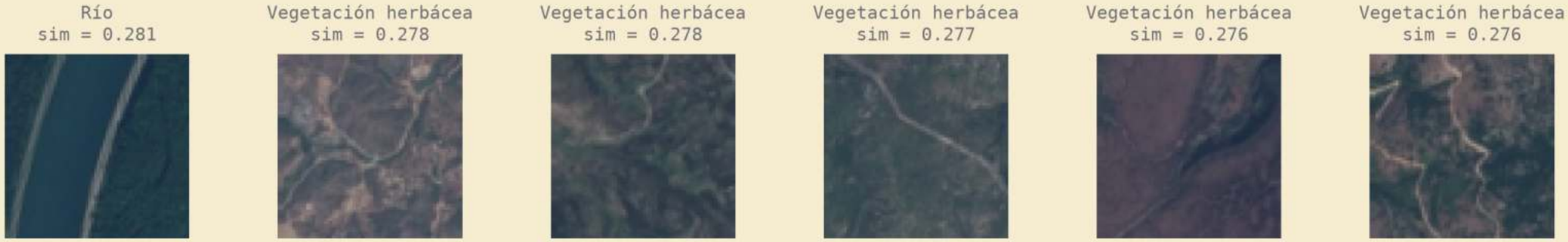
Buscar imágenes escribiendo una frase, y la respuesta a la pregunta guía.

Buscar imágenes escribiendo una frase

"a dense city with many buildings"



"a river crossing a forest"



Búsqueda semántica: CLIP es multimodal, texto e imagen en el mismo espacio

RESUMEN DE RESULTADOS

Del azar a 85.8%, sin entrenar redes

Método	Datos etiquetados	Exactitud
Al azar	0	10.0%
Zero-shot con el nombre de la clase	0	36.6%
Zero-shot: “a centered satellite photo of …”	0	44.8%
Regresión logística sobre embeddings	5 por clase	65.2%
Regresión logística sobre embeddings	20 por clase	77.1%
Regresión logística sobre embeddings	150 por clase	85.8%

Y el buscador de imágenes similares: **78.8%** de precisión en 5.

500 imágenes de prueba · muestra de 2,000 imágenes de EuroSAT

VOLVIENDO A LA PREGUNTA GUÍA

¿Basta un modelo general?

Para explorar y buscar

Sí

Sin una sola etiqueta: mapa del dataset, buscador con 78.8% y búsqueda con frases.

Para clasificar con nuestras categorías

No del todo

El zero-shot se queda en 44.8%: no conoce nuestro vocabulario ni nuestras imágenes.

La combinación

Lo mejor

El modelo aporta la representación; nuestros datos, el dominio. 20 por clase: 77.1%.

LA IDEA PARA LLEVARSE

Los embeddings son ingeniería de características automática.

Con la tabla de 2,000 × 512, todo lo demás (PCA, entrenamiento y prueba, regresión logística, k-NN, matriz de confusión) es lo mismo que con cualquier dataset tabular.

LIMITACIONES

Lo que hay que tener en cuenta

- **Imágenes pequeñas:** CLIP espera 224 × 224; las de 64 × 64 se amplían y pierden detalle.
- **Fotos de internet:** CLIP vio pocas imágenes satelitales. Hay modelos entrenados con percepción remota, como RemoteCLIP.
- **Solo 3 de 13 bandas:** el infrarrojo distingue muy bien vegetación, agua y zonas construidas.
- **Datos de Europa:** habría que verificar el desempeño con paisajes, cultivos y ciudades de México.
- **Una muestra:** 2,000 imágenes y 500 de prueba; los porcentajes pueden variar algunos puntos.

Liu et al. (2024), RemoteCLIP · arXiv:2306.11029

PARA SEGUIR EXPLORANDO

Para después del taller

1. **Más datos:** aumenta las imágenes por clase y repite la curva de aprendizaje.
2. **Ajusta C** de la regresión logística con validación cruzada.
3. **Tus propias frases:** ¿qué devuelve *"a golf course"* o *"an airport"*? ¿Por qué?
4. **Escala el buscador** con una base vectorial como FAISS.

REPOSITORIO

github.com/RodolfoFerro/buscador-imagenes-ia

BLOG · DOS PARTES

rodolfoferro.xyz/buscador-imagenes-ia

DATOSTADA 2026

¡Gracias!

¿Preguntas?



Rodolfo Ferro

rodolfoferro.xyz · @rodo_ferro · github.com/RodolfoFerro

Notebook: rodolfoferro.xyz/buscador-imagenes-notebook