

# AI vs Real Image

Alejandro Castro Rondon  
Nicolás Linares Rojas

# ENFOQUE



Clasificar imágenes entre reales y generadas por inteligencia artificial



Generación de imágenes por IA



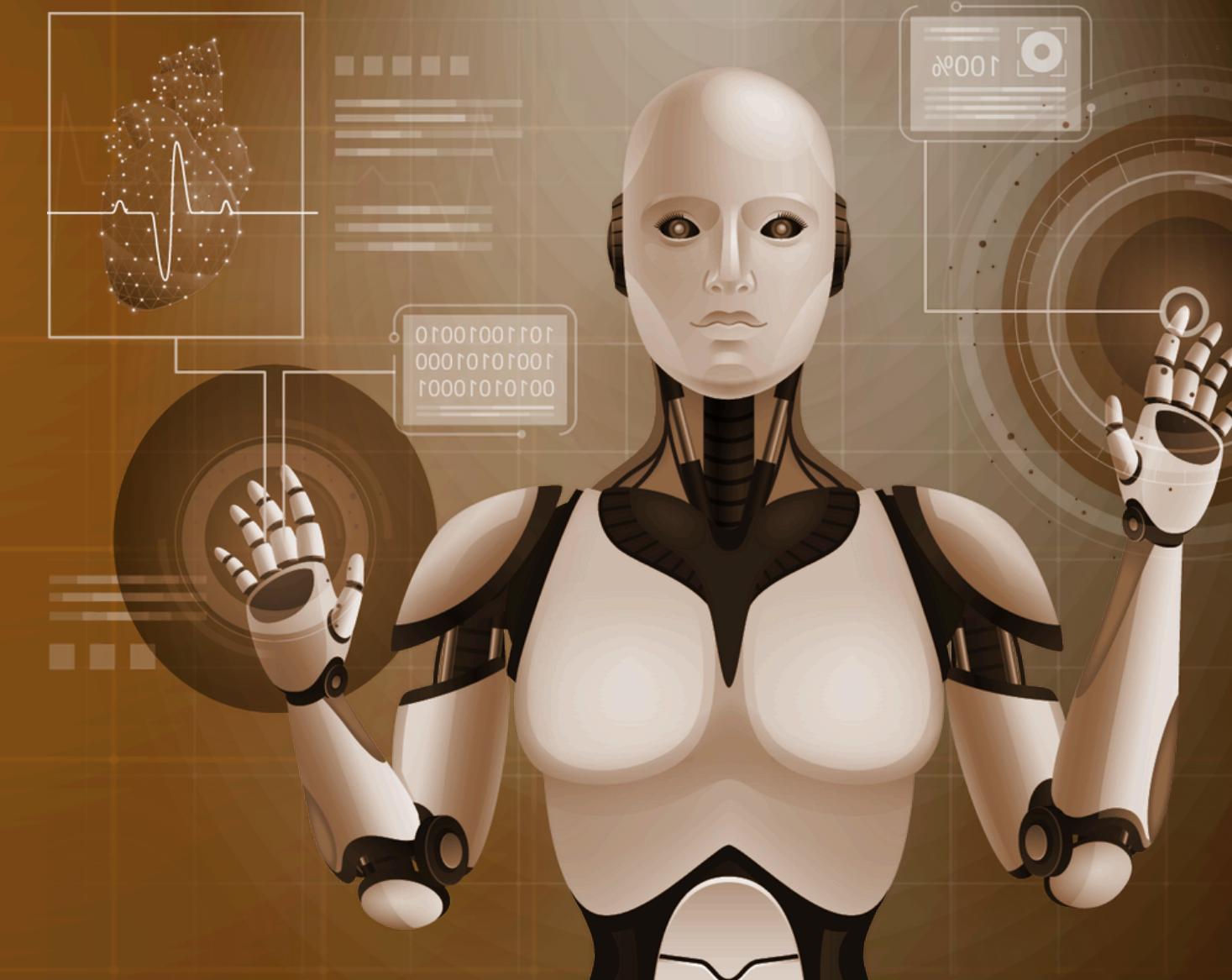
Redes Sociales



Ciberseguridad



Evidencia Legal

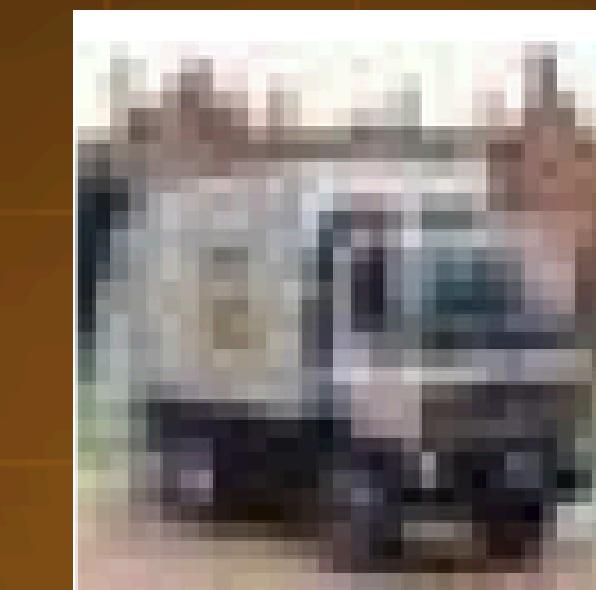


# The Data-Set (CIFAKE)



**REAL**

60,000 really taken  
images



**FAKE**

60,000 synthetically-  
generated images

Images were normalized (vals between 0-1)



20,000 were used for  
testing  
100,000 were used for...  
training

# Pre-Procesamiento

## Carga de Imágenes y Etiquetado

```
real_images, real_labels = load_image_data(os.path.join(TRAIN_PATH, 'REAL'), [0, 1])  
fake_images, fake_labels = load_image_data(os.path.join(TRAIN_PATH, 'FAKE'), [1, 0])
```



Se definen funciones para cargar imágenes de las carpetas REAL y FAKE, asignando etiquetas personalizadas:

# Normalización de Datos

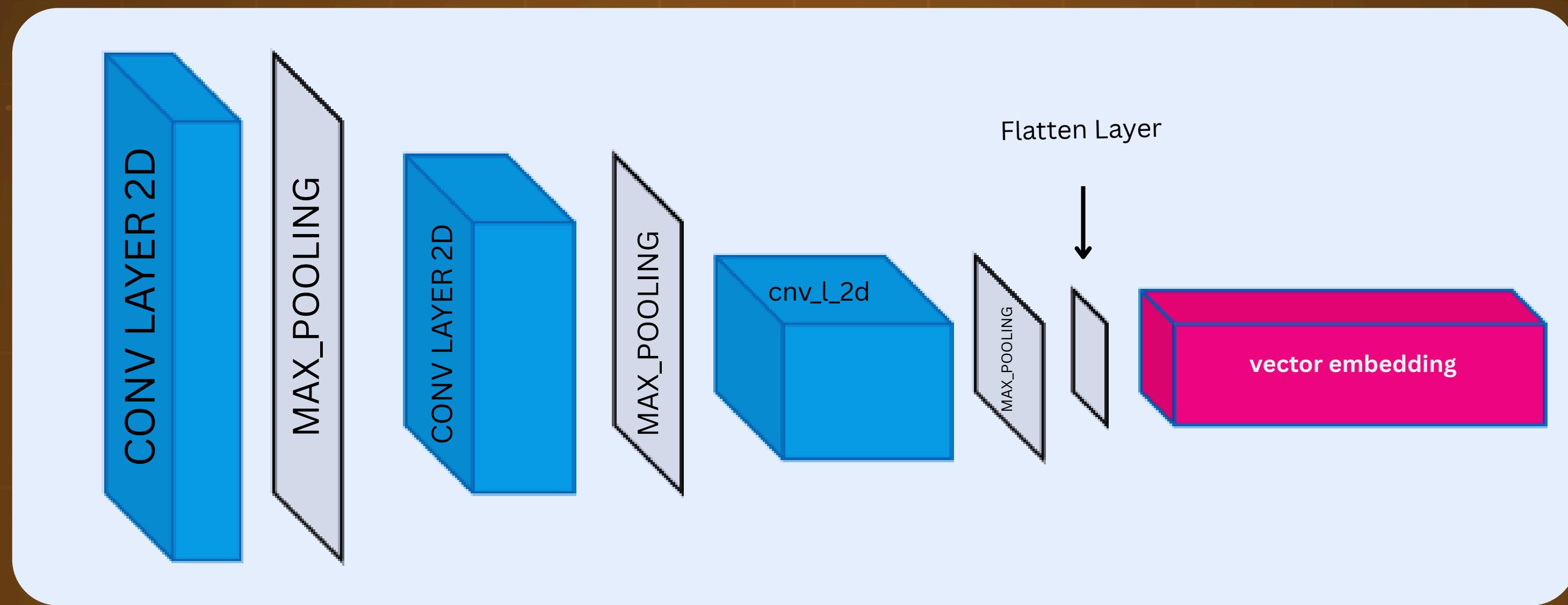
Las imágenes se normalizan dividiendo los valores de píxeles por 255, para que estén en el rango  $[0, 1]$ .



```
x_train = np.concatenate((real_images, fake_images)).astype('float32') / 255.0  
y_train = np.concatenate((real_labels, fake_labels))  
x_test = np.concatenate((test_real_images, test_fake_images)).astype('float32') / 255.0  
y_test = np.concatenate((test_real_labels, test_fake_labels))
```

Esto mejora la eficiencia del entrenamiento.

# Convolutional Pretraining

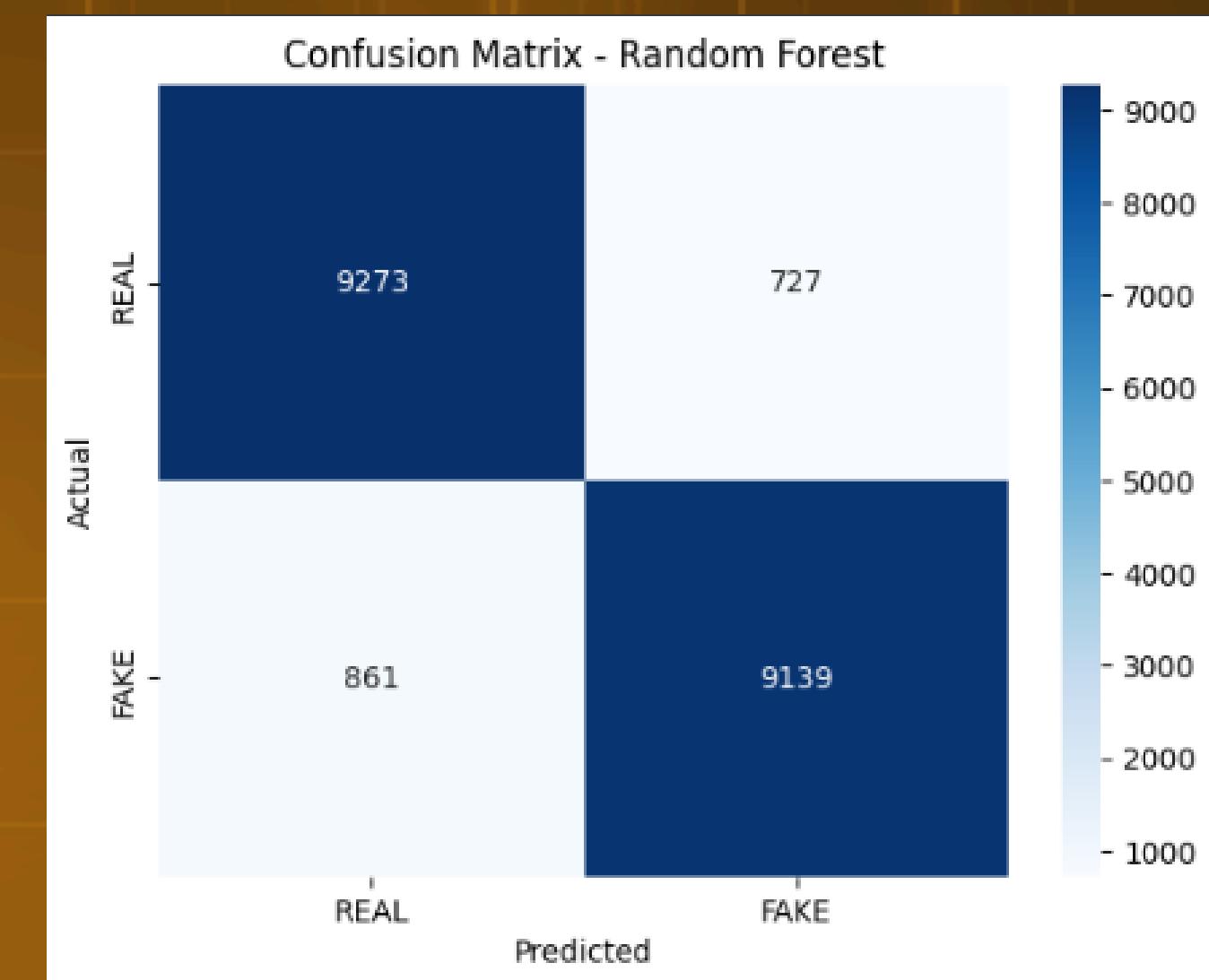


# Metodos Machine Learning

## Random Forest

**Accuracy: 0.9282**  
**Precision: 0.9222**  
**Recall: 0.9352**

- Vectores de Características
- Diferencias Detectables



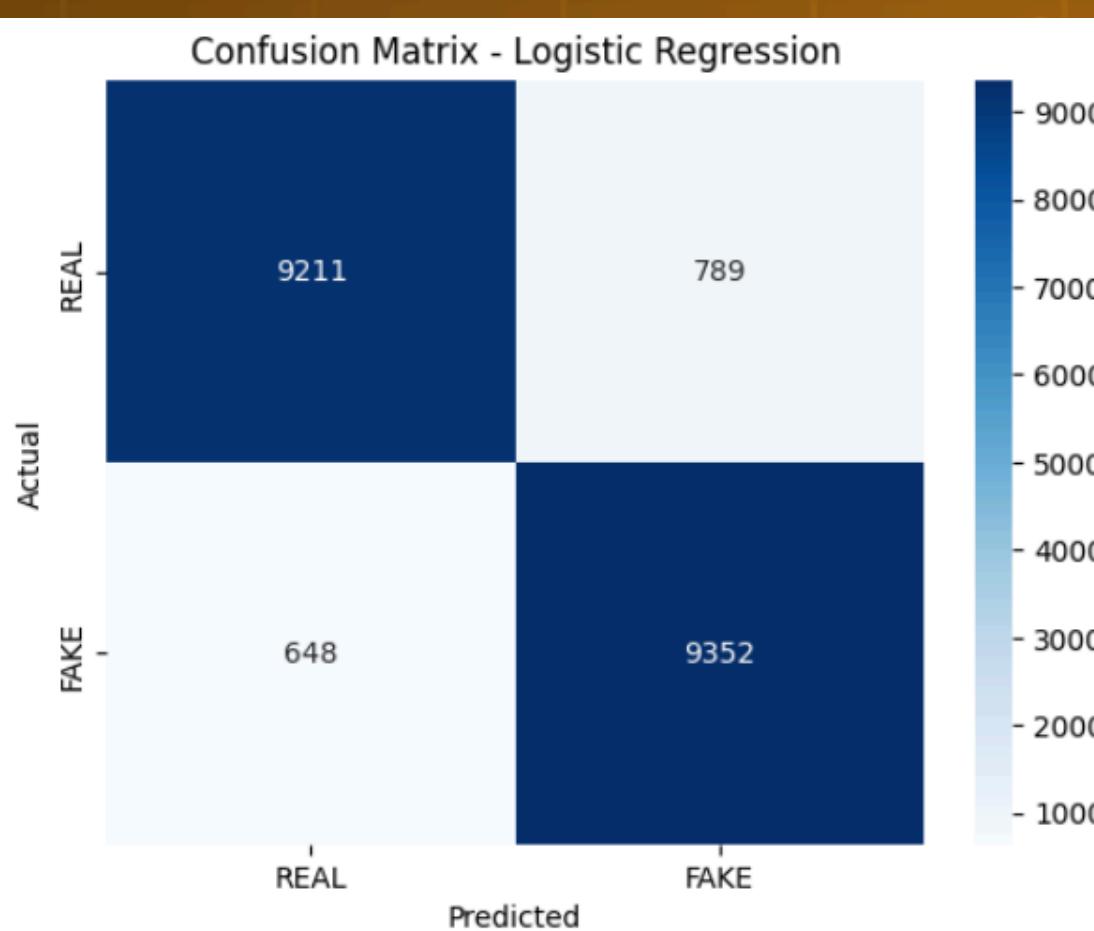
# Logistic Regression



Accuracy: 0.9282

Precision: 0.9222

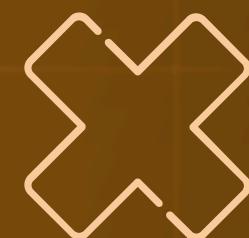
Recall: 0.9352



REAL / FAKE

Captura la relacion sin  
sobreajustarse

Datos Lineales



# XGBoost



Boosting

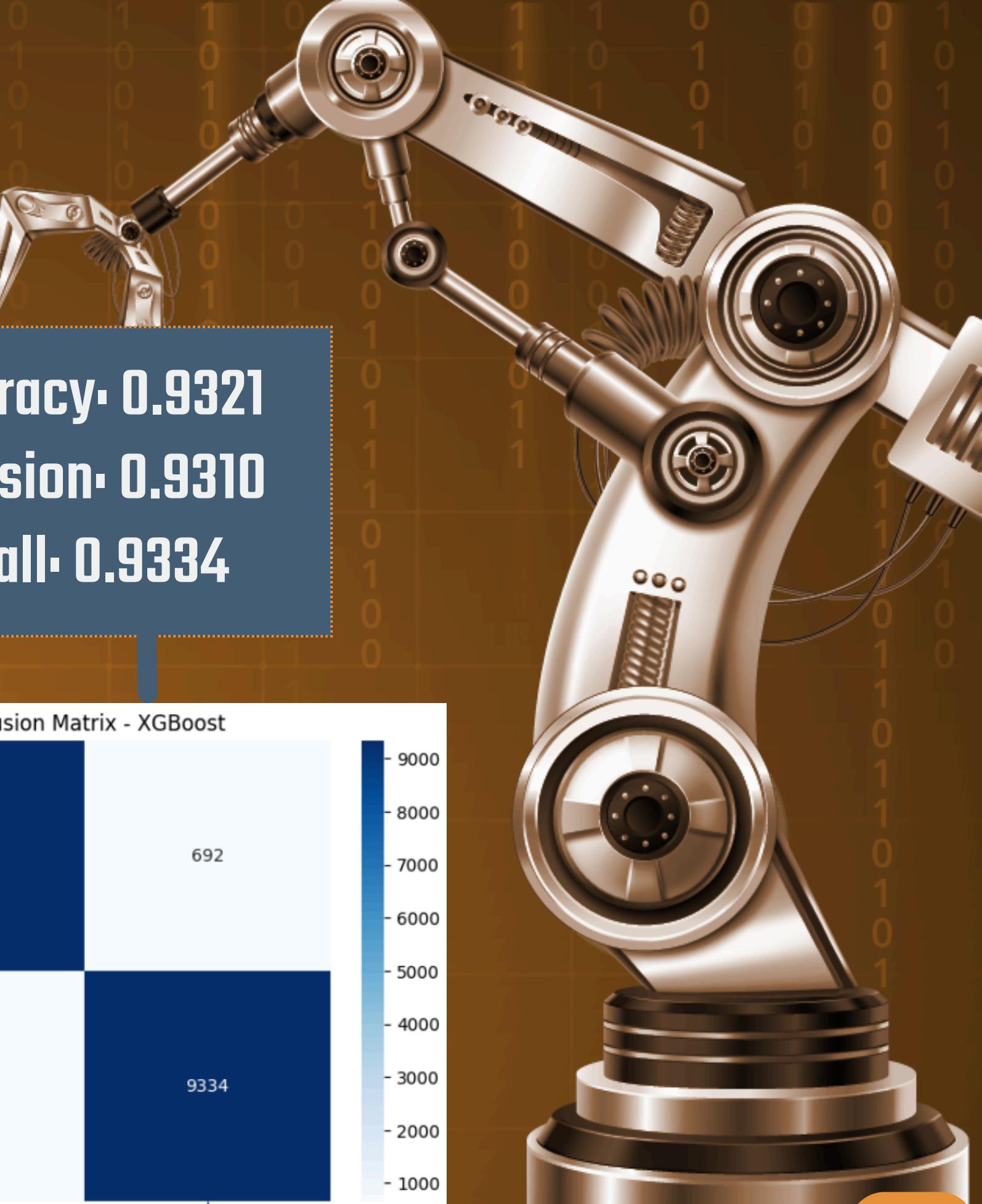
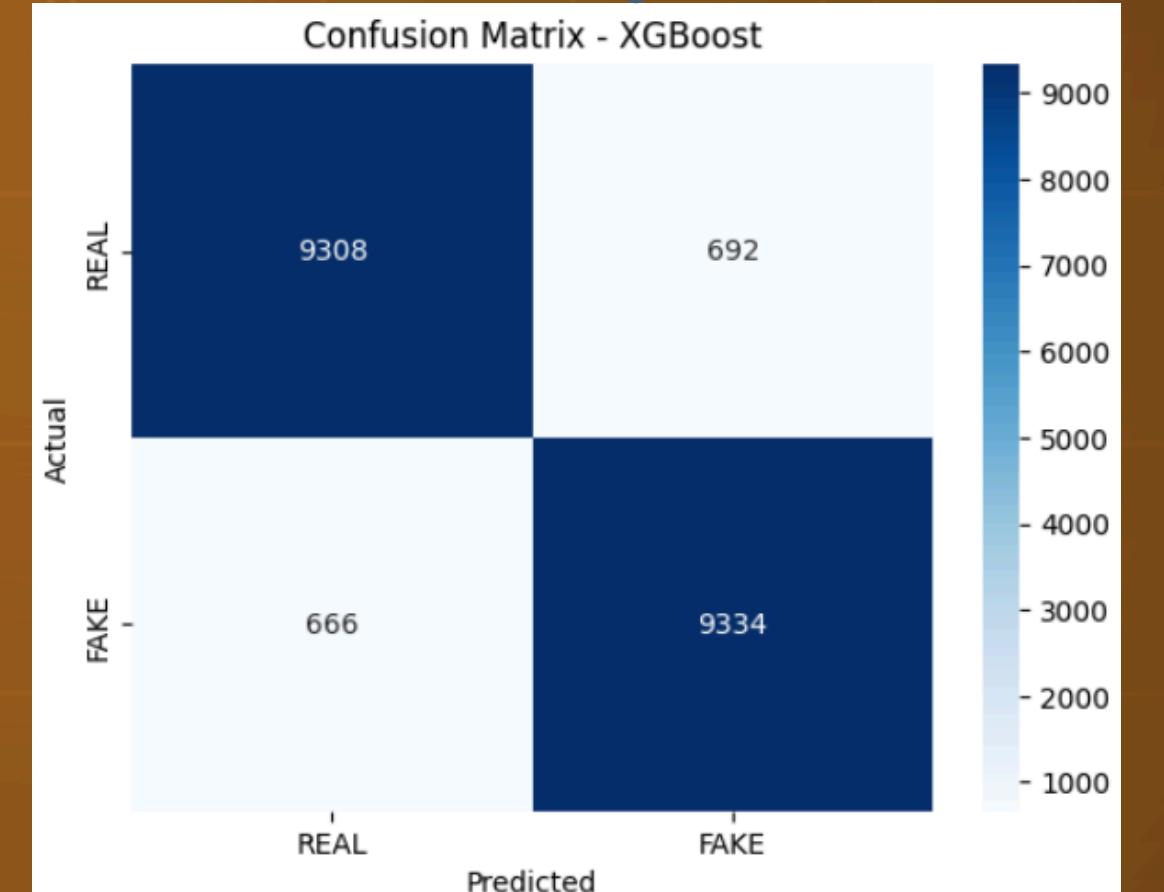


Extrae Features de las imagenes

Aprende Patrones Lineales

Características Densas → Embeddings

Accuracy - 0.9321  
Precision - 0.9310  
Recall - 0.9334

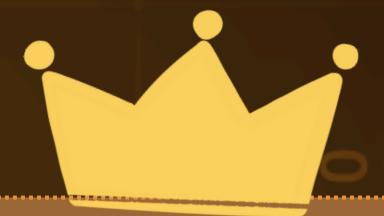




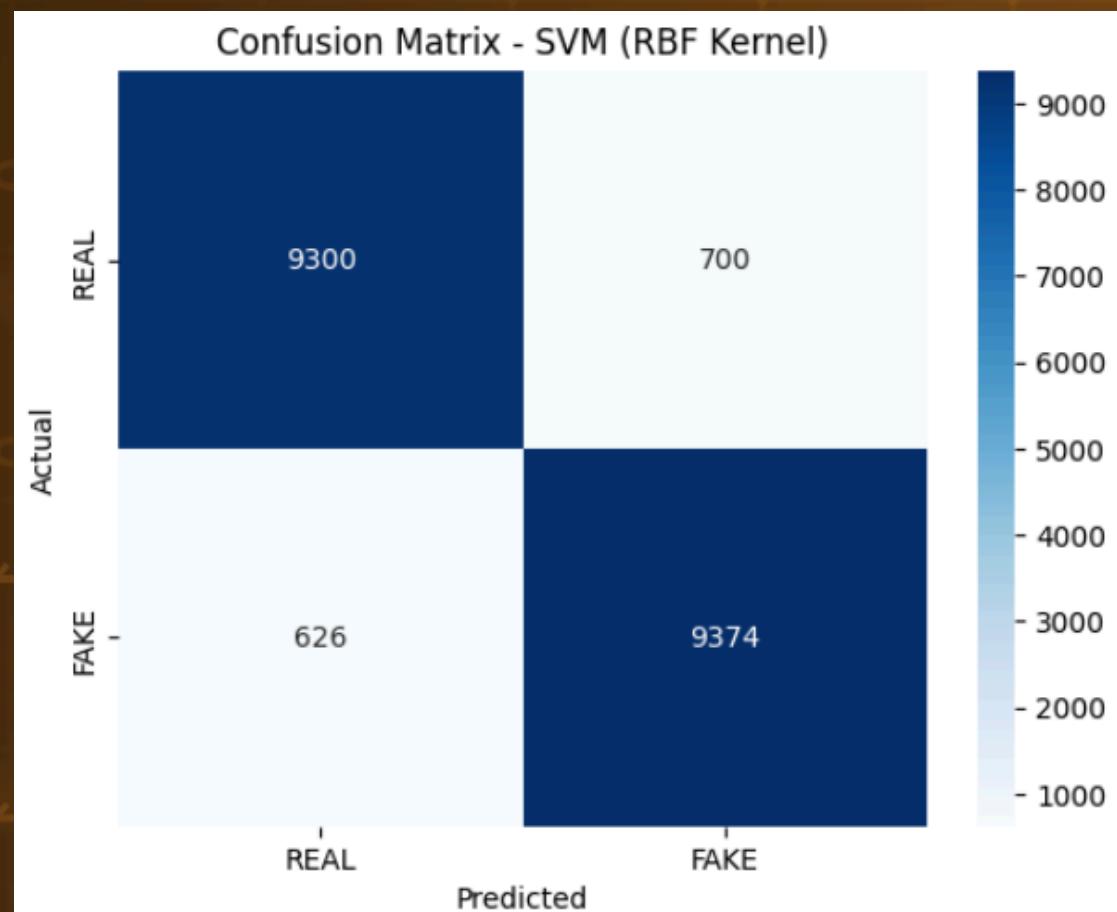
# SVW



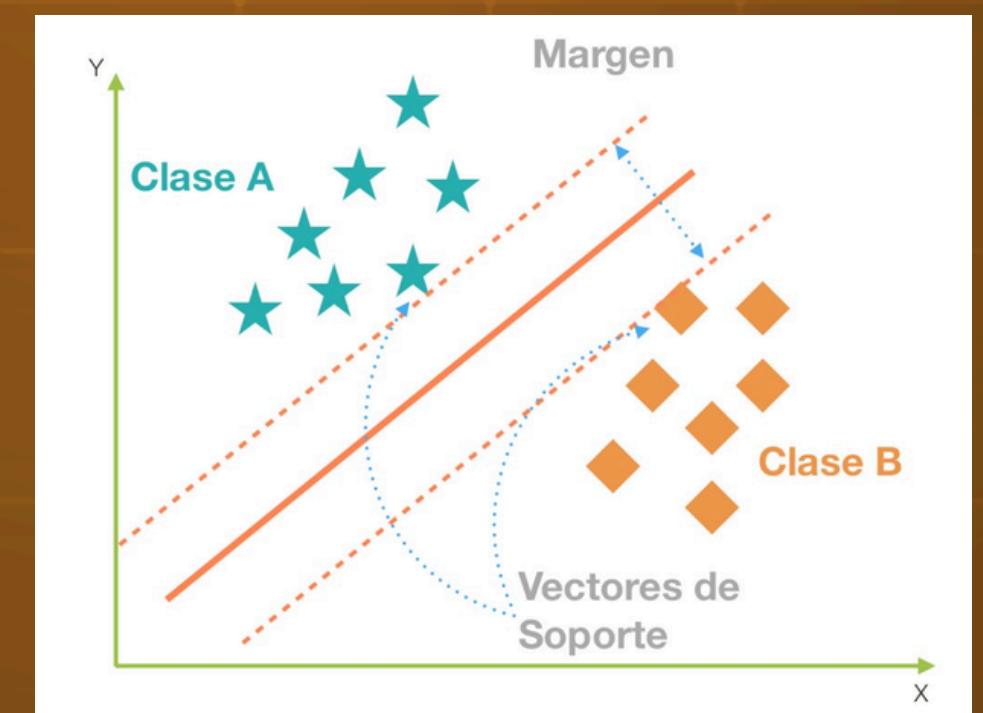
# RBF KERNEL



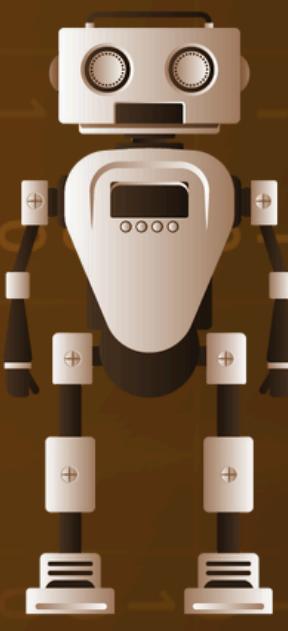
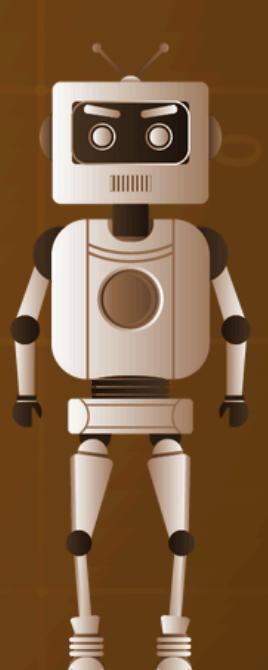
Accuracy: 0.9337  
Precision: 0.9305  
Recall: 0.9374



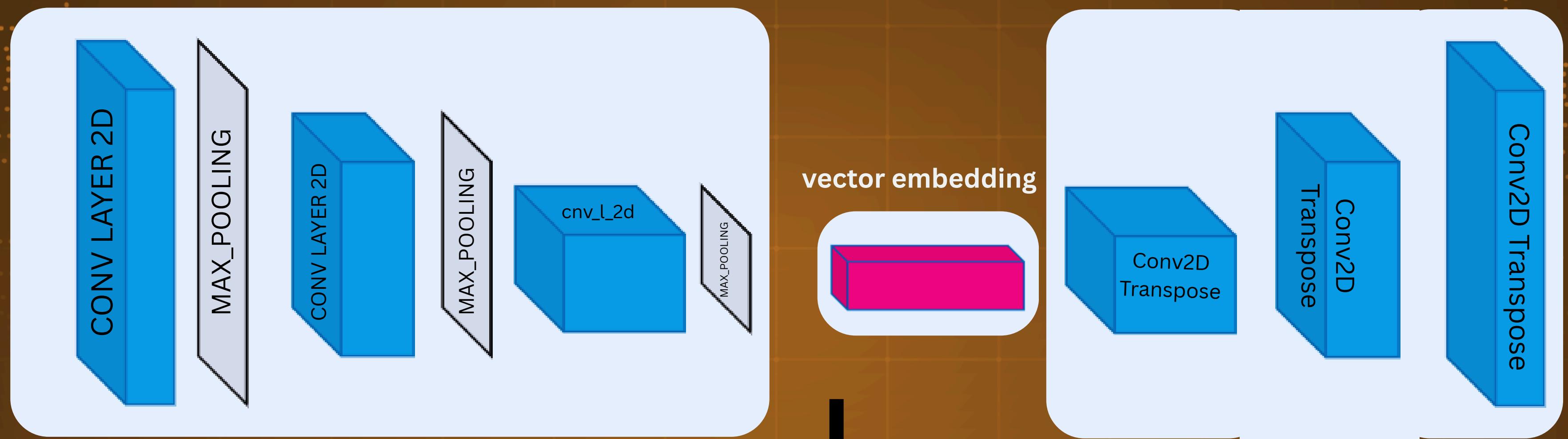
## Embeddings



Hiperplano



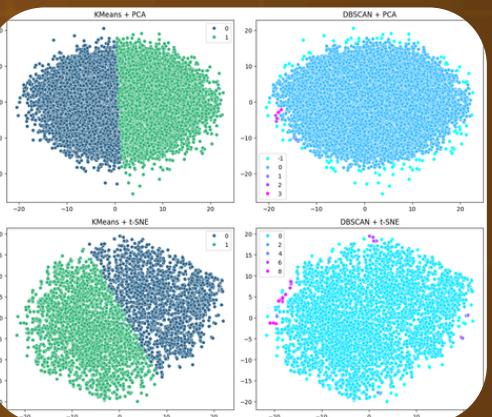
# Autoencoder architecture



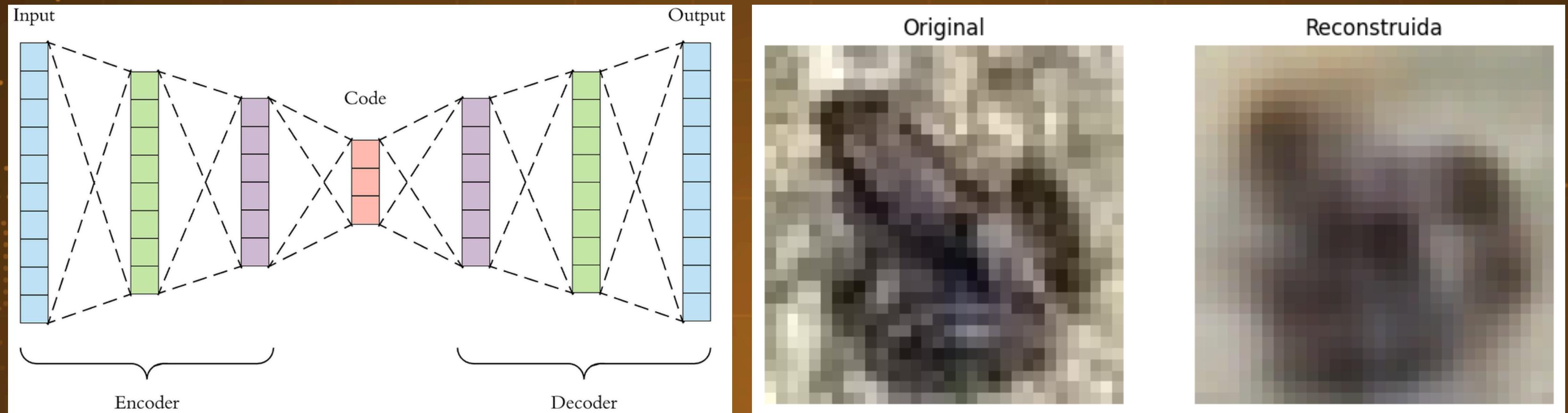
vector embedding



Clustering methods



# Autoencoder



```
→ Training model...
Epoch 1/5
2813/2813 727s 255ms/step - loss: 0.0286 - psnr_metric: 16.4029
Epoch 2/5
2813/2813 729s 251ms/step - loss: 0.0152 - psnr_metric: 18.8291
Epoch 3/5
2813/2813 704s 237ms/step - loss: 0.0135 - psnr_metric: 19.3530
Epoch 4/5
2813/2813 719s 250ms/step - loss: 0.0126 - psnr_metric: 19.6502
Epoch 5/5
2813/2813 744s 251ms/step - loss: 0.0121 - psnr_metric: 19.8452
```

# Metodos No Supervisados

Reducción Dimensionalidad

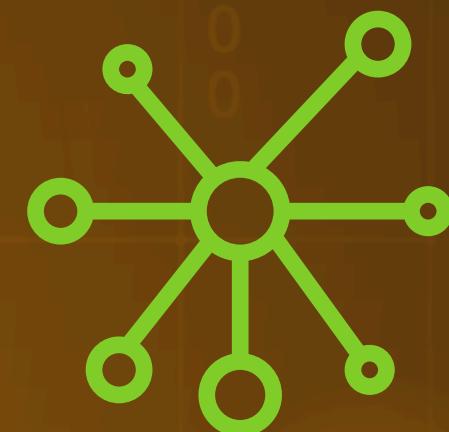
PCA



t-SNE

Clustering

KMeans



DBSCAN

Silhouette Score, Calinski-Harabasz y Davies-Bouldin.

## Silhouette Score



Cohesion vs Separacion

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

1 - Mejor  
Agrupamiento



## Calinski-Harabasz Score

Dispersión entre clústeres y la dispersión dentro de los clústeres.

$$CH = \frac{\text{Between-cluster dispersion}}{\text{Within-cluster dispersion}} \times \frac{n - k}{k - 1}$$



Mejor Separación y Agrupamiento

## Davies-Bouldin Score

La similitud promedio entre cada clúster y su clúster más similar.

### DB Score

Razón entre:

- Dispersión dentro del clúster i
- Distancia a su clúster más cercano j



## KMeans + PCA

Clusters encontrados: 2

Silhouette Score: 0.331

Calinski-Harabasz Score: 49418.195

Davies-Bouldin Score: 1.179

## DBSCAN + PCA

Clusters encontrados: 4

Silhouette Score: 0.370

Calinski-Harabasz Score: 24.856

Davies-Bouldin Score: 2.755

## KMeans + t-SNE

Clusters encontrados: 2

Silhouette Score: 0.401

Calinski-Harabasz Score: 4117.347

Davies-Bouldin Score: 0.995

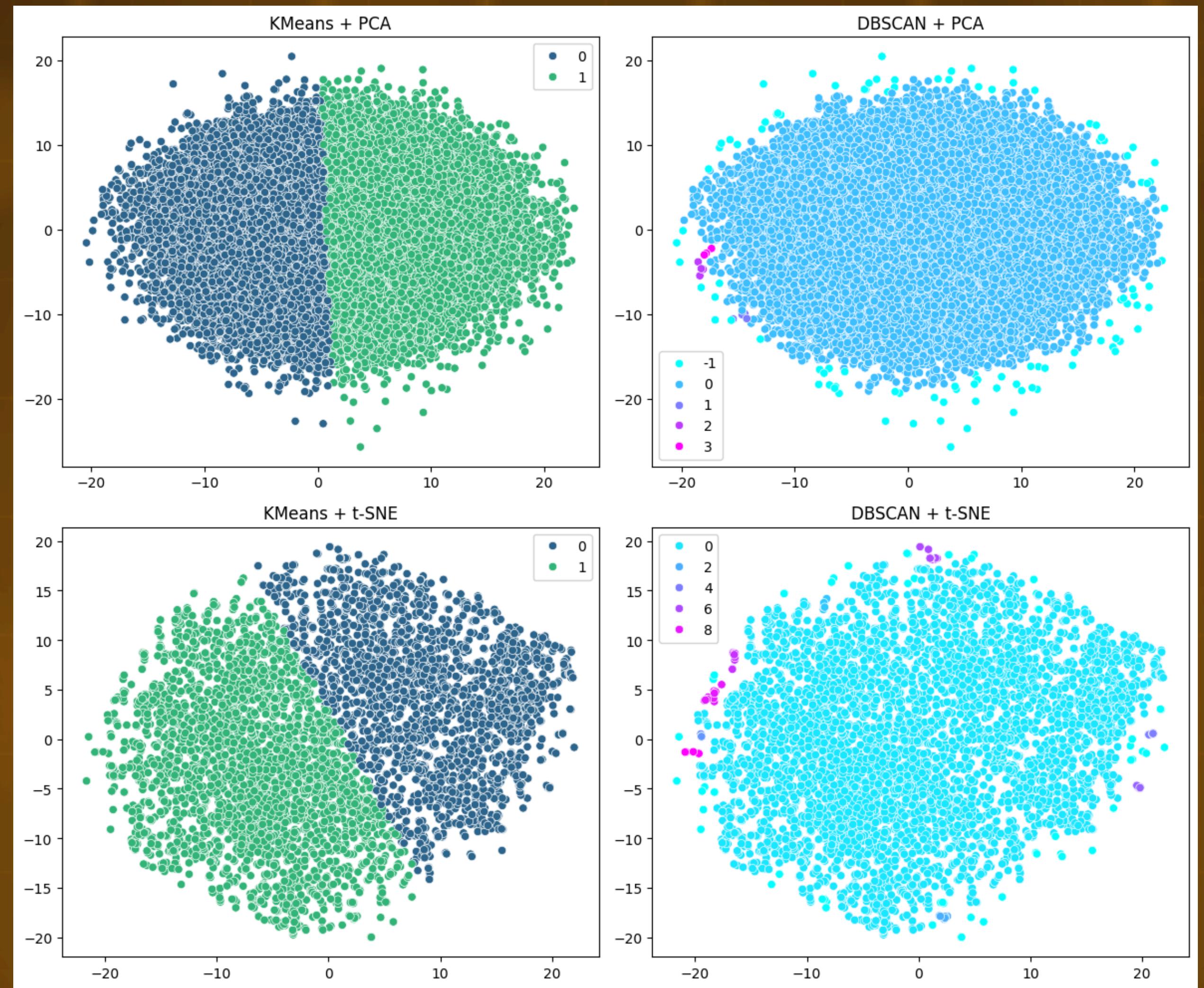
## DBSCAN + t-SNE

Clusters encontrados: 10

Silhouette Score: -0.365

Calinski-Harabasz Score: 15.556

Davies-Bouldin Score: 3.775



### KMeans + t-SNE

Clusters encontrados: 2

Silhouette Score: 0.401

Calinski-Harabasz Score: 4117.347

Davies-Bouldin Score: 0.995

### DBSCAN + t-SNE

Clusters encontrados: 10

Silhouette Score: -0.365

Calinski-Harabasz Score: 15.556

Davies-Bouldin Score: 3.775

Preserva relaciones locales entre puntos

### KMeans



Clústeres convexos y bien separados



### DBSCAN

Necesita densidades y distancias reales

No preserva distancias globales ni densidades reales

KMeans + PCA - Metricas Supervisadas

Accuracy: 0.558

Precision: 0.573

Recall: 0.463

DBSCAN + PCA - Metricas Supervisadas

Accuracy: 0.500

Precision: 0.500

Recall: 0.999

KMeans + t-SNE - Metricas Supervisadas

Accuracy: 0.547

Precision: 0.561

Recall: 0.494

DBSCAN + t-SNE - Metricas Supervisadas

Accuracy: 0.512

Precision: 0.510

Recall: 0.993

Identificar patrones útiles en los  
embebidos sin necesidad de etiquetas.



**GRACIAS!**

