I302 - Aprendizaje Automático y Aprendizaje Profundo

Trabajo Práctico 4: Estudio de metodos de clustering y reduccion dimensional

Ignacio Elian Gremes

31 de mayo de 2025

Ingeniería en Inteligencia Artificial

1. Algoritmos de clustering en aprendizaje no supervisado

Resumen

En este trabajo se utilizaron los algortimos de K-Means, GMM (Gaussian Mixture Models) y DBSCAN para encontrar cual performa mejor en ciertos manifolds, cuales son sus ventajas y desventajas. Para llevarlo a cabo se probo estos tres algoritmos en el mismo dataset, pero sue fueron variando los hiperparametros de estos. Se utilizo un K = 13, para los 3 algoritmos, y se obtuvo que tanto K-Means, GMM, inicializado con K-Means, dependen fuertemente de de la inicializacion de los centroides o las medias de los clusters, en caso de que estos sean seleccionados de forma incorrecta, el rendimiento de estos dos algoritmos sera pesimo. Por otro lado DBSCAN, no depende de una buena inicializacion, por lo tanto siempre se obtienen los mismos clusters en diferentes pruebas del algoritmo, pero si es muy dependiente de un buena eleccion de ϵ y **min pts** ya que si estos no son correctos, se puede terminar considerando a dos clusters juntos, como uno solo cuando en realidad son dos distribuciones separadas.

1.1. Introducción

El problema tratado en este trabajo es clusterizar un dataset, con modelos no supervisados. La entrada de este trabajo es un dataset con muestras que representan un lugar en un espacio 2D. Y la salida son los parametros del algoritmo seleccionado y los labels que indican a que cluster fue asignada cada muestra. Para obtener esto se utilizaron tres algoritmos distintos, K-Means, Gaussian Mixture Models (GMM) y DBSCAN.

1.2. Métodos

1.3. K-means

K-means es un algoritmo de particionado que divide un conjunto de datos $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ en K clústeres $C_1, C_2, \ldots, C_K$ minimizando la suma de las distancias cuadradas entre los puntos y los centroides correspondientes:

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

donde $\boldsymbol{\mu}_k$ es el centroide del clúster C_k.

1.4. Gaussian Mixture Models (GMM)

GMM asume que los datos provienen de una combinación de K distribuciones gaussianas multivariadas. La función de densidad conjunta está dada por:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

donde π_k es el peso del componente k, $\boldsymbol{\mu}_k$ su media y $\boldsymbol{\Sigma}_k$ su matriz de covarianza. Los parámetros se estiman mediante el algoritmo de *Expectation-Maximization* (EM).

1.5. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifica regiones densas en el espacio de características. Dos puntos $\mathbf{x}_i$ y $\mathbf{x}_j$ se consideran vecinos si:

$$\|\mathbf{x}_i - \mathbf{x}_j\| \leq \varepsilon$$

Un punto se considera *núcleo* si tiene al menos `MinPts` vecinos dentro de una distancia ε. Los clústeres se forman a partir de conexiones transitivas entre puntos núcleo, mientras que los puntos no conectados a ninguna región densa se etiquetan como *ruido*.

1.6. Métricas de Evaluación

1.6.1. Ganancias Decrecientes

Para evaluar la utilidad de aumentar el número de clústeres, se usó la métrica de *ganancias decrecientes*, que cuantifica la mejora relativa entre configuraciones consecutivas del modelo. Dada una métrica $M(k)$ (como el log-likelihood o la varianza explicada), se define:

$$\Delta M(k) = M(k) - M(k-1)$$

El cociente entre ganancias sucesivas permite identificar puntos de saturación:

$$r(k) = \frac{\Delta M(k)}{\Delta M(k-1)}$$

Valores decrecientes de $r(k)$ indican que el beneficio de aumentar k se está reduciendo.

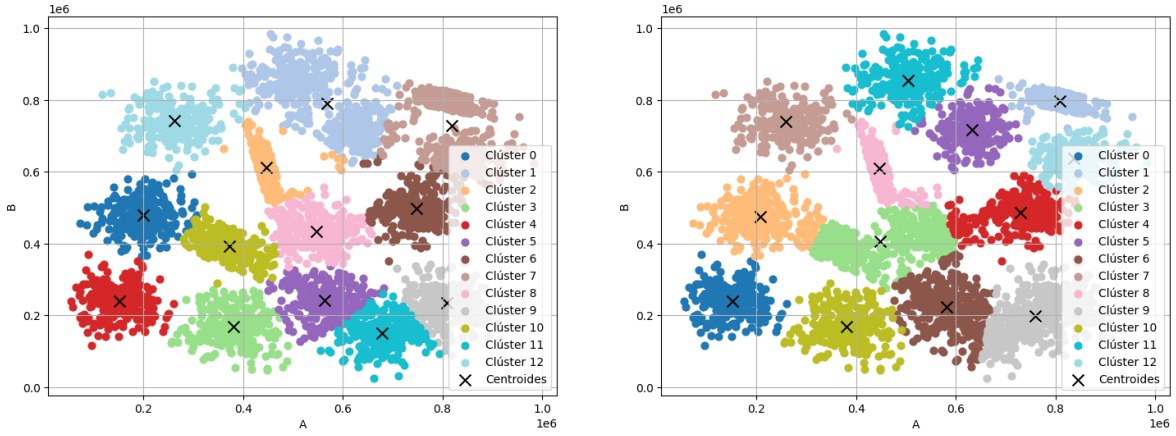
1.6.2. Log-Likelihood

En el caso de GMM, se evalúa la verosimilitud de los datos bajo el modelo entrenado. El log-likelihood total está dado por:

$$\log \mathcal{L}(\Theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

donde $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ representa los parámetros del modelo. Un mayor log-likelihood indica mejor ajuste, aunque se controla el sobreajuste mediante evaluación sobre un conjunto de validación.

1.7. Resultados



(a) Clusterizacion del algoritmo K-means con K = 13 e inicializacion random de los centroides, primera inicializacion

(b) Clusterizacion del algoritmo K-means con K = 13 e inicializacion random de los centroides, segunda inicializacion

Figura 1

Como se puede ver en la Figura 1, tenemos el resultado de haber utilizado, K-means con K = 13. Luego explico porque se utilizaron estos hiperparametros. Pero analicemos los clusters obtenidos. Si miramos lo obtenido en la Figura 1a, vemos que en los clusters superiores falló, unifico dos clusters que deberian ser distintos, esto es debido que este algoritmo es muy dependiente de la inicializacion de los centroides. Esto se puede comprobar en la Figura 1b, donde se ve que los clsuters superiores fueron separados correctamente, pero ahora fallan los del medio. Y lo unico que cambio fue la inicializacion de los centroides.

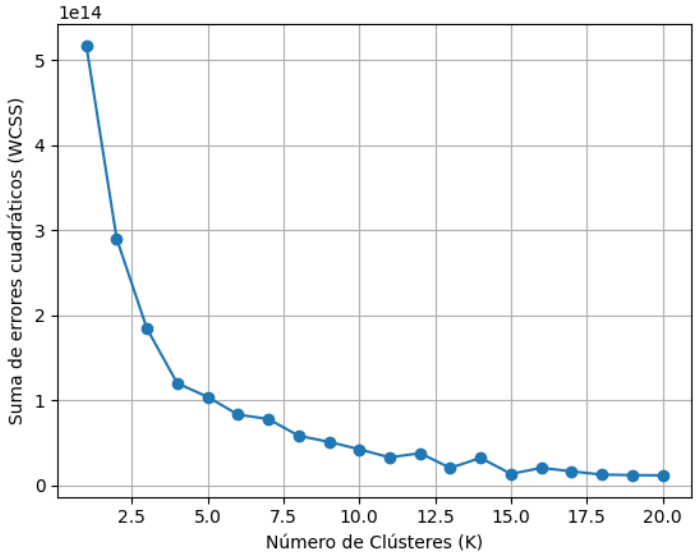
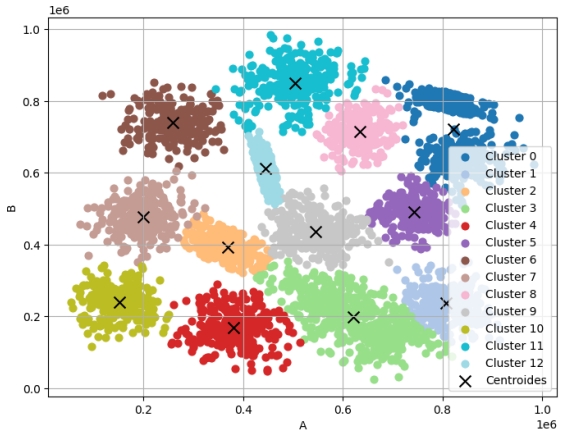
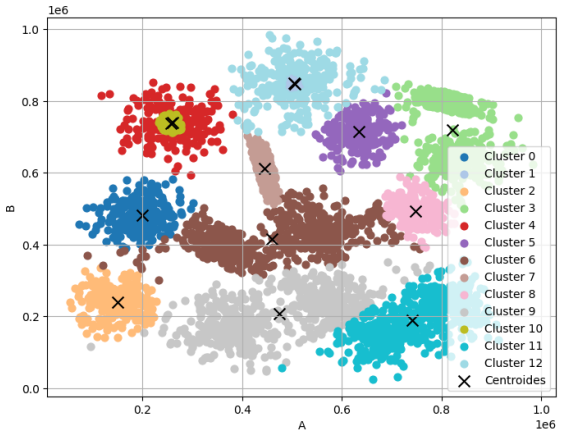


Figura 2: Ganancias decrecientes del algoritmo K-means.

Observando la Figura 7, con este metodo de ganancias decrecientes, decidimos implementar el algoritmo de K-Means con K = 13. Esto debido a que es un punto donde la suma de los errores se plancha y ya deja de haber une mejoria notoria al aumentar el valor de K. Cabe destacar que a partir de que se eligio K = 13 para K-Means, utilizaremos este K tambien para GMM, por una cuestion de estudiarlos a ambos en circunstancias iguales, pero de igual manera veremos que K = 13 tambien es un buen valor para GMM.



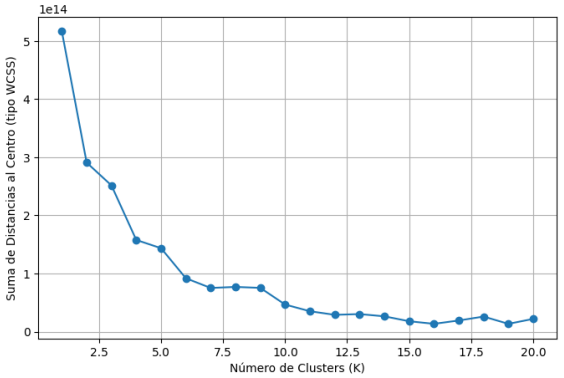
(a) Clusterizacion del algoritmo GMM con K = 13 e inicializacion con K-Means, primera inicializacion



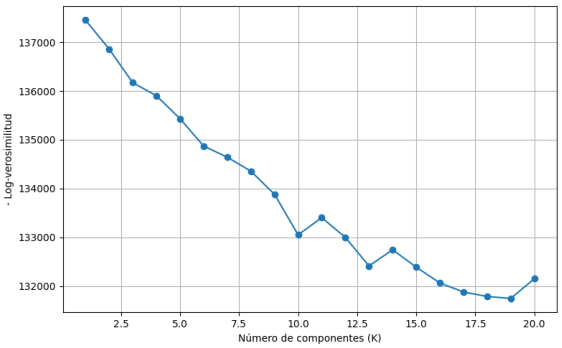
(b) Clusterizacion del algoritmo GMM con K = 13 e inicializacion con K-Means, segunda inicializacion

Figura 3

En GMM tenemos el mismo problema que con K-means, la inicializacion de los parametros es sumamente importante. Podemos ver en la Figura 3a que el algoritmo tambien falla en el cluster 0, unificando dos clusters que son en realidad distintos. Y si miramos la Figura 3b, la inicializacion fue pesima, al punto de que tenemos clusters dentro de cluster como se ve en el caso del cluster 4.



(a) Ganancias decrecientes del algoritmo GMM.



(b) Negative LogLikelihood del algoritmo GMM

Figura 4

Como se ve en la Figura 4a K = 13, es un valor que esta en el code de la curva, ero

para elegir cual de todos estos valores utilizar, miramos la Figura 4b y observamos que en 13, la negative loglikelihood es baja tambien y a pesar de que en valores mas cercanos a 17, es mas baja aun, nos parecio que la mejoria no era significativa. Ademas de que por cuestiones experimentales se prefirio utilizar K = 13 para mantener condiciones iguales entre los algoritmos.

Por lo tanto, ambos GMM y K-means obtienen resultados muy similares en este experimento, siendo ambos muy dependientes de una buena inicializacion de parametros, pero GMM tiene casos de inicializaciones donde funciona mucho peor que K-Means.

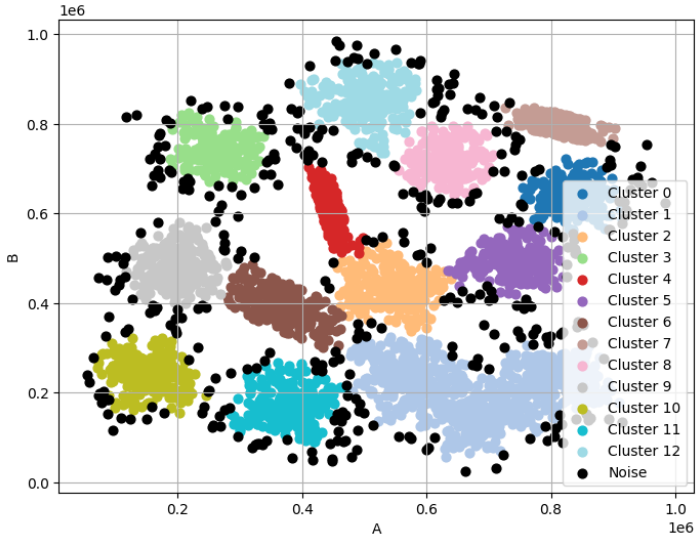


Figura 5: Clusterizacion del algoritmo DBSCAN con $\epsilon = 24000$ y **min pts** $= 11$

Luego de un barrido de hiperparametros, se llego a esta combinacion que dio los mejores resultados. Se puede ver que la clusterizacion es mejor que la de K-Means y la de GMM, ademas de que no hay tanta aleatoriedad como en los algoritmos anteriores, porque no hay que inicializar parametros. La unica desventaja con DBSCAN es que en el cluster 1, no es capaz de diferenciar que son 3 clusters distinos, ya que al estar tan cerca, es imposible que el algoritmo no salte de uno al otro, por mas que se aumente el **min pts**.

2. Reduccion de dimensionalidad

Resumen

En este experimento se busco estudiar la capacidad de reduccion de dimensionalidad de dos metodos, Principal Component Analysis (PCA) y Variatonal Auto Encoders (VAE). Para llevarlo a cabo se utilizo el dataset MNIST, se descompusieron los valores singulares de este dataset, y se estudio la cantidad de componentes principales para lograr una buena reconstruccion con PCA. Por otro lado en VAE, se fueron modifcando los hiperparametros de la red, y la estructura de esta para minimizar el error de reconstruccion.

2.1. Introducción

El objetivo de este trabajo fue obtener la mejor recontruccion de una imagen, a partir de aplicarle una reduccion dimensional con PCA y VAE. La entrada del algoritmo es una imagen, de los numeros del 0 al 9, en blanco y negro. Luego a partir de la reduccion dimensional, se obtiene una salida de la imagen reconstruida, comprimida.

2.2. Métodos

2.2.1. Análisis de Componentes Principales (PCA)

PCA es una técnica lineal que proyecta los datos a un subespacio de menor dimensión que maximiza la varianza.

En la práctica, PCA se implementa eficientemente utilizando la descomposición en valores singulares (SVD):

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

Donde:

- $\mathbf{U} \in R^{n\times n}$: vectores singulares izquierdos,
- $\mathbf{\Sigma} \in R^{n\times d}$: matriz diagonal con valores singulares,
- $\mathbf{V} \in R^{d\times d}$: vectores singulares derechos (autovectores de $\mathbf{X}^\top\mathbf{X}$).

La proyección a k dimensiones se obtiene tomando las primeras k columnas de $\mathbf{V}$:

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_k$$

Donde $\mathbf{V}_k \in R^{d\times k}$ contiene los k autovectores principales asociados a los mayores valores singulares.

2.2.2. Autoencoder Variacional (VAE)

Un *Autoencoder Variacional* es un modelo generativo probabilístico que aprende una representación latente continua y estructurada de los datos. Se basa en la idea de modelar la distribución de los datos $\mathbf{x} \in R^d$ mediante una variable latente $\mathbf{z} \in R^k$ con una distribución previa $p(\mathbf{z})$, típicamente $\mathcal{N}(0, \mathbf{I})$.

La probabilidad marginal se modela como:

$$p(\mathbf{x}) = \int p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})\, d\mathbf{z}$$

Dado que esta integral es intratable, se introduce un modelo aproximado para la posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$, parametrizado por una red neuronal (el encoder). El VAE se entrena maximizando una cota inferior (ELBO) del log-likelihood:

$$\log p(\mathbf{x}) \geq E_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\right)$$

Donde:

- $p_\theta(\mathbf{x} \mid \mathbf{z})$ es el decodificador, que reconstruye $\mathbf{x}$ desde $\mathbf{z}$.
- $\mathrm{KL}(\cdot\|\cdot)$ es la divergencia de Kullback-Leibler.

En la práctica, se asume que $q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x})))$ y se utiliza la técnica de *reparametrización* para hacer la función objetivo diferenciable:

$$\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$$

El VAE se entrena mediante descenso de gradiente estocástico para minimizar la pérdida total:

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}_{\text{Reconstrucción}} + \beta \cdot \underbrace{\mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\right)}_{\text{Regularización}}$$

donde $\hat{\mathbf{x}} = p_\theta(\mathbf{x} \mid \mathbf{z})$ y β controla el peso del término de regularización (en el VAE estándar, $\beta = 1$).

2.3. Métricas de comparación de imágenes

2.3.1. 1. Error cuadrático medio (MSE)

El **Mean Squared Error (MSE)** mide la diferencia promedio al cuadrado entre los píxeles correspondientes de dos imágenes, una original I y una reconstruida $\hat{I}$. Se define como:

$$\mathrm{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \Big(I(i,j) - \hat{I}(i,j)\Big)^2 \tag{1}$$

donde $M \times N$ es el tamaño de la imagen. Un valor de MSE más bajo indica una mejor reconstrucción.

2.3.2. 2. Relación señal a ruido de pico (PSNR)

La **Peak Signal-to-Noise Ratio (PSNR)** mide la calidad de la imagen reconstruida respecto a la original, expresada en decibelios (dB). Está relacionada con el MSE de la siguiente manera:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{L^2}{\mathrm{MSE}}\right) \tag{2}$$

donde L es el valor máximo posible de un píxel (por ejemplo, 255 para imágenes de 8 bits). Un valor de PSNR más alto indica una mejor calidad.

2.3.3. 3. Índice de similitud estructural (SSIM)

El **Structural Similarity Index (SSIM)** compara la estructura local entre dos imágenes. Considera luminancia, contraste y estructura. Su forma general es:

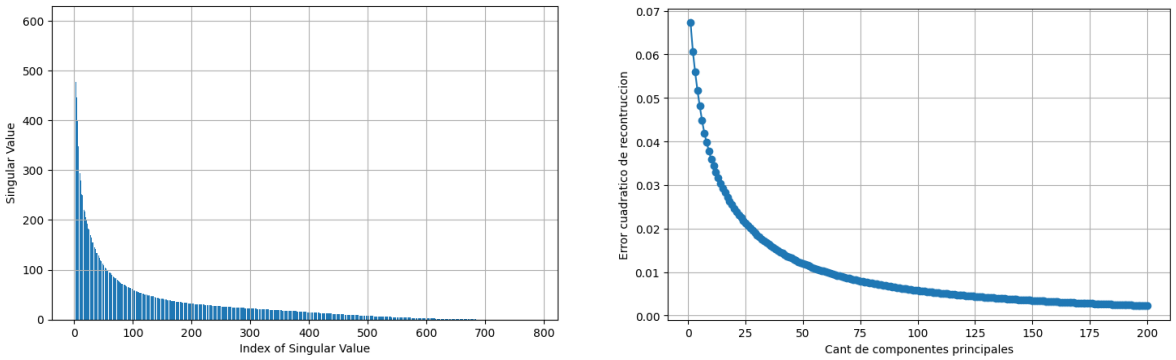
$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3}$$

donde:

- μ_x, μ_y son los promedios locales de las imágenes x y y,
- σ_x^2, σ_y^2 son las varianzas locales,
- σ_{xy} es la covarianza local,
- C_1, C_2 son constantes para estabilizar la división.

El SSIM varía entre 0 y 1, donde 1 indica similitud perfecta.

2.4. Resultados



(a) Valores de los valores singulares de SVD

(b) Error de reconstruccion segun cantidad de componentes principales usadas.

Figura 6

Como se ve en la Figura 6b el error cuadratico medio se plancha alrededor de la componente 200, y si encima miramos la Figura 6a vemos que para el valor singular 200, ya hemos agarrado a la mayoría de lo valores singulares que tienen la mayor cantidad de información. Por lo tanto se decidio utilizar 200 componentes principales para la reduccion dimensional.

Para el VAE se utilizo una capa de entrada y de salida de 784 neurona, una capa oculta de 400 neurona con activacion lineal, en el encoder, y otra en el decoder, y una capa latente con 30 neuronas con activación lineal. Se utilizo optimizador ADAM, con lr = 1e-3 con minibatch de tamaño 128. Y se entreno a la red durante 10 epocas.

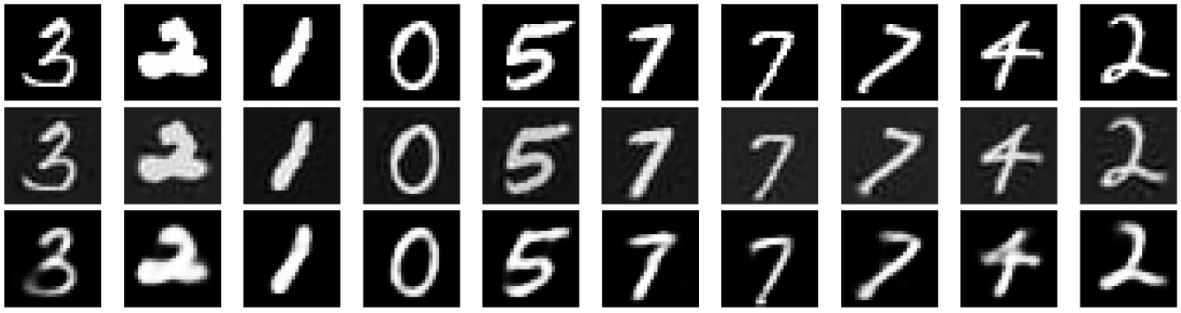


Figura 7: Comparacion entre imagen original y reconstruccion con PCA y VAE. Siendo la primera fila la imagen original, la segunda fila la imagen reconstruida con PCA y la tercera fila la imagen reconstruida con VAE.

Se puede ver que ambos métodos logran una buena reconstrucción, pero ambos tienen algunos problemas. En PCA se ve que reconstruye bien la forma de los numeros, pero no recuper bien el color original, a pesar de que estas imágenes están en blanco y negro, se ve que el negro pierde color y el blanco también, ambos tienden a un color mas grisacio. En cambio en VAE, el color se mantiene idéntico, pero pierde un poco la forma del numero, especialmente en los que tienen bastante curva como el 2 y el 3. Por lo tanto este análisis me lleva a pensar que para un caso de imágenes mas complejas, si lo importante es mantener la forma de la imagen, conviene usar PCA, pero si lo importante es mantener el color de la imagen entonces VAE es mejor. Esto también lo podemos corroborar con las siguiente métricas.

Cuadro 1: Métricas por imagen (Top 10)

Img	MSE_PCA	MSE_VAE	PSNR_PCA	PSNR_VAE	SSIM_PCA	SSIM_VAE
0	0.10129	0.01670	9.94	17.77	0.41925	0.87546
1	0.18828	0.01078	7.25	19.67	0.30474	0.96451
2	0.15108	0.00357	8.21	24.47	0.02230	0.97999
3	0.15131	0.00783	8.20	21.06	0.08992	0.95866
4	0.11790	0.01641	9.28	17.85	0.42730	0.92374
5	0.09464	0.00863	10.24	20.64	0.50688	0.94411
6	0.10922	0.00922	9.62	20.35	0.21088	0.91988
7	0.12376	0.00821	9.07	20.85	0.35365	0.94202
8	0.09548	0.01282	10.20	18.92	0.22029	0.91969
9	0.14059	0.01754	8.52	17.56	0.14077	0.89519

MSE indica la distancia entre los pixeles originales con los pixeles reconstruido y vemos que en VAE, este es menor, comporbando que VAE reconstruye mejor el valor original de los pixeles. Pero si miramos PSNR y SSIM que indican la calidad de la imagen y su similitud estructural perceptual, respectivamente, vemos que en estas VAE tambien rinde mejor, ya que en PSNR valores mas altos son mejores y en SSIM los valores mas cercanos a 1 son mejores. Por lo tanto, estudiando las metricas, llegamos a la conclusion de que la recontruccion de VAE es mejor que la de PCA, a diferencia de la suposición que hicimos antes de que para algunos casos PCA podria ser mejor.