Proyecto 2 - Reducción de la dimensionalidad

[*]Link del código: https://github.com/m41k1204/ml_classification

1ro Michael Hinojosa
Facultad de Computación
Ciencias de la Computación
michael.hinojosa@utec.edu.pe
Contribución: 100%

2do Jean Piero Huaylla
Facultad de Computación
Ciencias de la Computación
jean.huaylla@utec.edu.pe
Contribución: 100%

3ro Hector Espinoza
Facultad de la Computación
Ciencias de la Computación
hector.espinoza@utec.edu.pe
Contribución: 100%

I. Introducción

En el siguiente documento se presentará el desarrollo y los hallazgos del Proyecto 2 de CS-3061 Machine Learning. El objetivo del proyecto es analizar y comparar diversos métodos de reducción de dimensionalidad aplicados a la clasificación de imágenes de moda. Se implementarán técnicas lineales y no lineales: Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF), t-Distributed Stochastic Neighbor Embedding (t-SNE) y Uniform Manifold Approximation and Projection (UMAP), así como la investigación de Spectral Embedding e Isomap. Antes de evaluar el rendimiento de cada técnica, se realizará un preprocesamiento del dataset Fashion-MNIST, normalizando los valores de píxeles, agregando el bias y analizando los datos. Posteriormente, se aplicarán las técnicas de reducción de dimensionalidad mencionadas, generando visualizaciones bidimensionales que permitan comprender la estructura y separabilidad de las clases. Asimismo, se entrenarán múltiples clasificadores sobre los datos reducidos: Máquinas de Soporte Vectorial (SVM), Regresión Logística, Random Forest y k-Vecinos Más Cercanos (KNN). Con el objetivo de encontrar la mejor combinación de técnica de reducción y clasificador, se compararán los algoritmos mediante métricas de evaluación aprendidas en el curso: Accuracy, Precision, Recall y F1-Score. Por último, se realizará un análisis crítico de los resultados para determinar cuál combinación ofrece el mejor balance entre rendimiento y eficiencia computacional.

II. Objetivos

A. Objetivo principal

El objetivo principal del trabajo es encontrar el mejor par de técnica de reducción de dimensionalidad y modelo clasificador.

B. Objetivos Específicos

- Implementar todas las técnicas de reducción de dimensionalidad aprendidas en el curso.
- Implementar las técnicas de reducción de dimensionalidad investigadas: Spectral Embedding y Isomap.
- Evaluar 4 modelos clasificadores: KNN, Regresión Logística, Random Forest y SVM con la ayuda de las técnicas de reducción de dimensionalidad.
- Utilizar métricas objetivas como: Accuracy, Precision, Recall y F1-Score.

III. Metodología

A. Fases

- La **Fase 1** constó de la carga de datos y su posterior normalización. También se le agregó el bias a los datos de entrenamiento y testeo. Finalmente se realizó un *Exploratory Data Analysis (EDA)* para poder comprender mejor los datos con los que se trabajará..
- En la **Fase 2** se implementaron las 6 técnicas de reducción de la dimensionalidad. Para cada técnica se brindará una explicación detallada de la teoría detrás del modelo.
- Durante la **Fase 3** se implementaron los 4 modelos clasificadores pedidos para validar y comparar las diferentes técnicas de reducción de la dimensionalidad.
- Finalmente en la **Fase 4** se analizó los resultados y se comparó, de la mano de métricas de evaluación, las técnicas utilizadas.

B. EDA

Los 10 tipos de ropa que hay en el Fashion-MNIST son:

1) T-shirt/top (Camiseta)
2) Trouser (Pantalón)
3) Pullover (Sudadera)
4) Dress (Vestido)
5) Coat (Abrigo)
6) Sandal (Sandalia)
7) Shirt (Camisa)
8) Sneaker (Zapatilla deportiva)
9) Bag (Bolsa)
10) Ankle boot (Botín)

En el presente informe se referirá a las clases por su nombre en inglés y en letra cursiva, por ejemplo: *Ankle boot*.
Las 10 clases pueden ser apreciadas en la siguiente imagen:

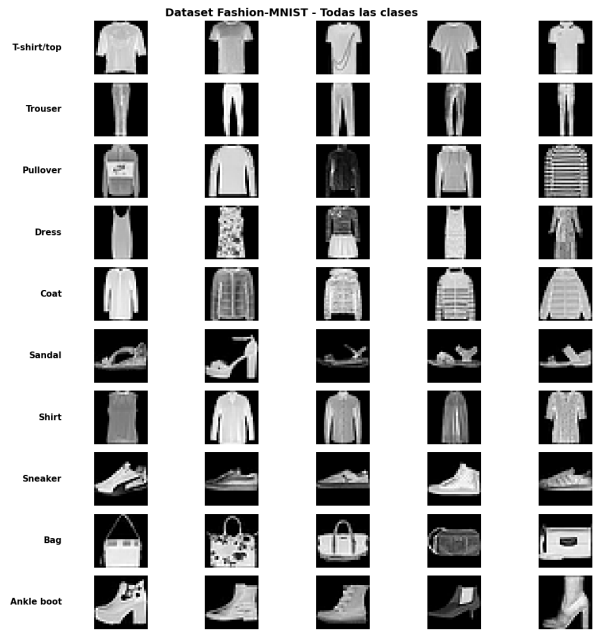

Fig. 1. Ejemplos de las 10 clases del dataset Fashion-MNIST

A simple vista se pueden identificar clases que no tienen muchas similitudes con otras, como por ejemplo *Bag* y *Trouser*. Estas son las más fáciles de clasificar, puesto que no hay mucho espacio para que el clasificador se confunda. Luego hay clases que si bien tienen similitudes entre si, aun así resulta fácil identificar diferencias. Estas son por ejemplo *Sneaker* y *Ankle boot*.

Finalmente, existen clases cuya identificación resulta más compleja debido a sus similitudes visuales. En particular, ciertas imágenes presentan características que las hacen ambiguas y podrían clasificarse legítimamente en múltiples categorías. Esto es especialmente evidente en los pares: *Shirt* con *T-Shirt*, *Shirt* con *Coat* y *Shirt* con *Pullover*.

Por un lado, las prendas *Shirt* presentan variabilidad en la longitud de mangas, pudiéndose encontrar ejemplares sin mangas o con mangas cortas, características que coinciden con la definición estándar de *T-Shirt*. Por otro lado, tanto *Shirt* como *Pullover* y *Coat* comparten la característica de poseer mangas largas, lo que dificulta su diferenciación basándose únicamente en este atributo. Consecuentemente, las clases *Pullover* y *Coat* también exhiben un alto grado de similitud, ya que ambas corresponden a prendas de abrigo con mangas largas, cuyas distinciones visuales son sutiles.

Este análisis visual preliminar proporciona un punto de referencia importante. Posteriormente, cuando se apliquen las técnicas de reducción de dimensionalidad, se podrá verificar si estas distinciones y confusiones entre clases se preservan o se transforman en las representaciones de baja dimensión, lo que dará información valiosa sobre la calidad y efectividad de cada técnica.

Finalizando la **Fase 1** del proyecto, se continuará con la **Fase 2**.

IV. Reducción de la Dimensionalidad

A. Principal Component Analysis (PCA)

PCA busca una proyección lineal ortonormal que capture la máxima varianza de los datos [1]. Sobre la matriz de observaciones centrada por columnas se calcula la descomposición espectral (o la *SVD*) y se seleccionan las primeras k direcciones principales; la representación reducida se obtiene proyectando las muestras sobre ese subespacio. En la implementación práctica, la fórmula que se utiliza directamente para obtener las representaciones reducidas es

$$\mathbf{Z} \;=\; \mathbf{X}\,\mathbf{W}, \tag{1}$$

donde $\mathbf{W} \in \mathbb{R}^{d\times k}$ contiene las k direcciones principales (columnas de $\mathbf{V}_k$ en la *SVD*). Se recomienda, cuando $k \ll d$, usar *randomized SVD* para reducir el coste computacional [2].

B. Non-negative Matrix Factorization (NMF)

NMF aproxima una matriz de datos no negativa por el producto de dos factores no negativos; el objetivo de referencia en la mayoría de implementaciones es la minimización del error de reconstrucción en norma de *Frobenius* [3]. La formulación usada por el algoritmo es

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \; \frac{1}{2}\big\|\mathbf{X}-\mathbf{W}\mathbf{H}\big\|_F^2. \tag{2}$$

Una implementación práctica y ampliamente utilizada son las actualizaciones multiplicativas (Lee & Seung), que por iteración se aplican en su forma final como

$$\begin{aligned} \mathbf{H} &\leftarrow \mathbf{H} \odot \frac{\mathbf{W}^\top \mathbf{X}}{\mathbf{W}^\top \mathbf{W}\mathbf{H} + \varepsilon}, \\ \mathbf{W} &\leftarrow \mathbf{W} \odot \frac{\mathbf{X}\mathbf{H}^\top}{\mathbf{W}\mathbf{H}\mathbf{H}^\top + \varepsilon}, \end{aligned} \tag{3}$$

donde $\odot$ es producto elemento a elemento y $\varepsilon > 0$ evita divisiones por cero.

C. t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE construye afinidades locales en el espacio original ajustando localmente la escala para una *perplexity* objetivo, y define afinidades en el espacio embebido mediante un núcleo de Student de una cola pesada. El algoritmo minimiza la divergencia de *Kullback–Leibler* (*KL*) entre ambas distribuciones por descenso de gradiente [4]. Las expresiones finales empleadas en la optimización son

$$\begin{aligned} C = \mathrm{KL}(P\|Q) \;&=\; \sum_{i\neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \\ q_{ij} &\propto \big(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\big)^{-1}, \end{aligned} \tag{4}$$

y el gradiente utilizado en cada paso para actualizar las bajas dimensiones $\{\mathbf{y}_i\}$ se evalúa como

$$\frac{\partial C}{\partial \mathbf{y}_i} \;=\; 4 \sum_j (p_{ij} - q_{ij}) \, \frac{\mathbf{y}_i - \mathbf{y}_j}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2}. \tag{5}$$

D. Uniform Manifold Approximation and Projection (UMAP)

UMAP aproxima la estructura local del *manifold* mediante un grafo difuso de afinidades y optimiza una proyección de baja dimensión que preserve esa estructura [5]. En la implementación práctica se usan las afinidades dirigidas normalizadas μ_{ij}, su simetrización difusa s_{ij}, una función de similitud en el espacio embebido $\phi(\cdot,\cdot)$ con cola pesada y una pérdida tipo entropía cruzada:

$$\begin{aligned} \mu_{ij} &= \exp\!\left(-\frac{d(\mathbf{x}_i,\mathbf{x}_j)-\rho_i}{\sigma_i}\right) \quad (j \in \mathrm{NN}_k(i)), \\ s_{ij} &= \mu_{ij} + \mu_{ji} - \mu_{ij}\mu_{ji}, \end{aligned} \tag{6}$$

$$\phi(\mathbf{y}_i,\mathbf{y}_j) = \frac{1}{1+a\|\mathbf{y}_i-\mathbf{y}_j\|^{2b}}, \tag{7}$$

$$\mathcal{L} = \sum_{i<j}\Big(s_{ij}\log\frac{s_{ij}}{\phi_{ij}} + (1-s_{ij})\log\frac{1-s_{ij}}{1-\phi_{ij}}\Big). \tag{8}$$

En la práctica, *UMAP* utiliza índices aproximados para k-NN (por ejemplo, *NN-descent*) y *negative sampling* para el término de repulsión; los hiperparámetros clave son *n_neighbors* (escala local) y *min_dist* (compacidad en el espacio embebido).

E. Spectral Embedding

Spectral Embedding es una técnica no lineal de reducción de dimensionalidad basada en la teoría espectral de grafos [6]. Representa los datos como un grafo donde los nodos son puntos de datos y las aristas representan similitudes entre ellos.

1) Matriz de Afinidad y Laplaciano: Se construye una matriz de afinidad $W \in \mathbb{R}^{n\times n}$ donde W_{ij} representa la similitud entre puntos. La matriz de grado diagonal D se define como $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

La matriz Laplaciana sin normalizar se define como [6]:

$$L = D - W$$

La matriz Laplaciana captura la estructura topológica del grafo. Intuitivamente, minimizar la función objetivo basada en el Laplaciano preserva las similitudes locales: puntos cercanos en el grafo deben permanecer cercanos en la proyección de baja dimensión.

2) Eigenvalores y Eigenvectores: Se resuelve el problema de eigenvalores [6]:

$$Lv = \lambda v$$

Los eigenvalores se ordenan: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Los eigenvectores asociados a los eigenvalores más pequeños contienen la información más valiosa sobre la estructura del grafo. Se seleccionan los k eigenvectores correspondientes a los k eigenvalores más pequeños y estos k eigenvectores forman las columnas de la matriz de embedding, donde cada punto x_i se proyecta a un vector k-dimensional y_i [6].

F. Isomap

Isomap (Isometric Mapping) es una técnica de reducción de dimensionalidad no lineal que preserva la geometría global de los datos [7, p. 2319]. A diferencia de métodos lineales como PCA y MDS clásico, Isomap busca preservar las distancias geodésicas en lugar de las distancias euclidianas.

1) Distancia Geodésica vs. Euclidiana: El concepto fundamental de Isomap es la distinción entre dos tipos de distancias. Para datos en variedades no lineales, puntos que están alejados sobre la variedad pueden aparecer deceptivamente cercanos en el espacio euclidiano de entrada. Como se ilustra en el ejemplo del Swiss roll, *"puntos lejanos sobre la variedad subyacente, medidos por su distancia geodésica o camino más corto, pueden aparecer engañosamente cercanos en el espacio de entrada de alta dimensión, medidos por su distancia euclidiana en línea recta"* [7, p. 2320]. Solo las distancias geodésicas reflejan la verdadera geometría de baja dimensión de la variedad [7, p. 2320].

2) Algoritmo de Tres Pasos: El algoritmo completo de Isomap consta de tres pasos [7, p. 2320]: en el primer paso se determina cuáles puntos son vecinos en la variedad, conectando cada punto a sus K vecinos más cercanos con pesos iguales a la distancia euclidiana. En el segundo paso, Isomap estima las distancias geodésicas entre todos los pares de puntos calculando los caminos más cortos en el grafo [7, p. 2320]. En el paso final, se aplica *Multidimensional Scaling (MDS)* clásico sobre la matriz de distancias geodésicas para obtener el embedding de baja dimensión que mejor preserva la geometría intrínseca de la variedad [7, p. 2320].

G. Implementación

La implementación de *PCA* y *NMF* se hizo acorde a la teoria explicada en el proyecto, mientras que las 4 siguientes técnicas se desarrollaron de la mano de la libreriía *Scikit Learn*.
Se continuará con la **Fase 3** del proyecto.

V. Evaluación

Una vez reducida la dimensionalidad de los datos, es necesario entrenar modelos que clasifiquen las imágenes en las 10 categorías. A continuación se describen brevemente los 4 modelos utilizados:

1) k-Nearest Neighbors (KNN): KNN es un algoritmo de aprendizaje basado en instancias que clasifica un punto según la mayoría de votos de sus k vecinos más cercanos en el espacio de características. Es simple, no paramétrico, y funciona bien cuando los datos tienen una estructura local clara. Su principal desventaja es que puede ser computacionalmente costoso en datasets grandes.

2) Regresión Logística: La regresión logística es un modelo de clasificación que utiliza una función sigmoide para estimar probabilidades de pertenencia a cada clase. Es rápido, interpretable, y funciona bien cuando las clases son aproximadamente linealmente separables. Para problemas multiclase, se extiende mediante el enfoque Softmax.

3) Random Forest: Random Forest es un algoritmo de ensamble que entrena múltiples árboles de decisión sobre subconjuntos aleatorios de los datos y características, combinando sus predicciones mediante votación. Es robusto, maneja bien características de diferentes escalas, y es menos propenso al sobreajuste que un árbol individual. Proporciona además una medida de importancia de características.

4) Máquinas de Soporte Vectorial (SVM): SVM busca encontrar el hiperplano que maximiza el margen entre clases, siendo especialmente efectivo en espacios de alta dimensión. Para problemas multiclase utiliza estrategias como One-vs-All o One-vs-One. Es potente y flexible gracias a los kernels, aunque requiere más ajuste de hiperparámetros que otros modelos.

VI. Resultados y Análisis

Finalmente se procederá con la **Fase 4:** un análisis comparativo de las técnias y modelos.

A. Análisis de PCA

El análisis de varianza explicada revela que el dataset posee una dimensionalidad intrínseca significativamente menor que sus 784 características originales.
La siguiente tabla presenta los puntos críticos donde se alcanza determinados niveles de varianza acumulada:

TABLE I
Componentes requeridos para alcanzar umbrales de varianza acumulada en PCA.

Umbral	Varianza	Componentes	% Dimensión	Reducción
80%	0.800	23	2.93%	97.07%
90%	0.900	87	11.10%	88.90%
95%	**0.950**	**299**	**38.14%**	**61.86%**
99%	0.990	524	66.84%	33.16%

que puede ser apreciado en la siguiente imagen, que detalla el crecimiento exponencial amortiguado de la varianza.

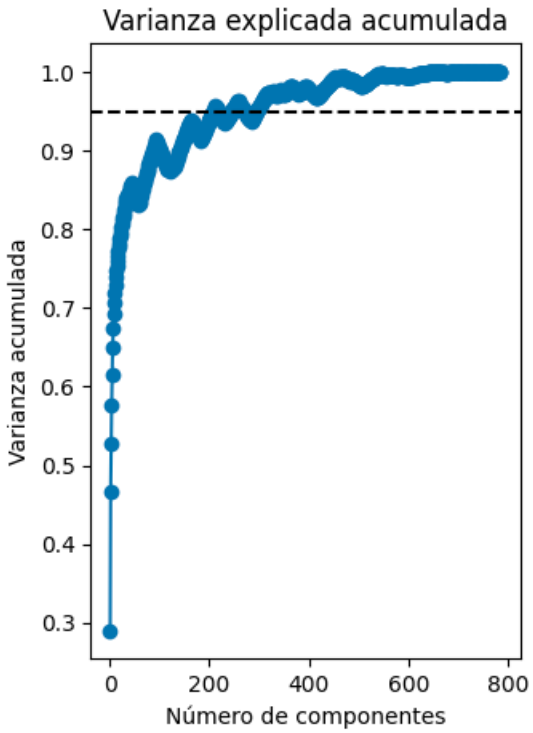

Fig. 2. Varianza explicada acumulada por componentes principales

El hallazgo más significativo es que el 95% de la varianza se concentra en apenas 299 componentes, lo que equivale al 38.14% de las dimensiones originales. Las 485 dimensiones restantes aportan únicamente el 5% de la varianza total, demostrando que los datos presentan una estructura de baja dimensionalidad.
Al utilizar $k = 299$ componentes, se logra una compresión de 2.62:1 (de 784 a 299 dimensiones) perdiendo solo el 5% de la información. Incrementar a $k = 500$ componentes agrega apenas un 4% adicional de varianza, requiriendo triplicar el número de dimensiones.

B. Visualizacion de Datos en 2D

Al visualizar los datos en 2D, luego de aplicar técnicas como *Isomap* o *Spectral Embedding* recibimos los siguientes resultados:

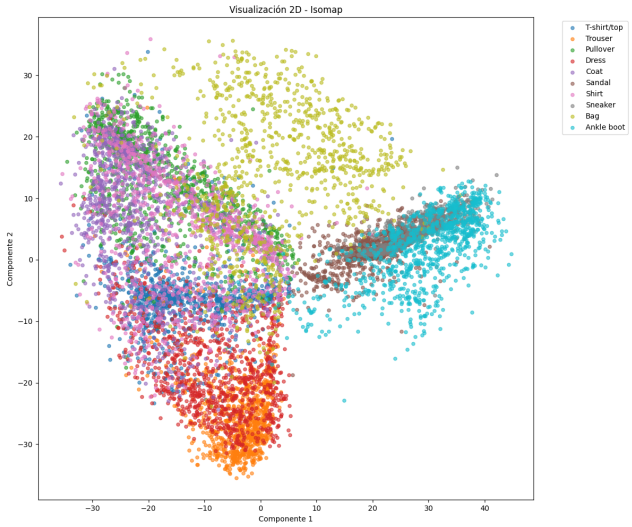

Fig. 3. Visualización 2D con Isomap

Fig. 4. Visualización 2D con Spectral Embedding

Si bien las reducciones estan en 2D y no en 50, 100 o 200 dimensiones, en las que las clases estarían más separadas, aún se pueden observar claras similitudes y diferencias entre las clases. Y aunque la posición de los puntos tanto en *Isomap* y *Spectral Embedding* sean completamente distintas, se pueden observar los mismos patrones. Por ejemplo *Bag*, la clase de color verde claro, o *Ankle boot*, en color celeste, estan separados del resto de clases y se pueden diferenciar bien claro. Sin embargo clases como *Shirt*, *T-Shirt*, *Pullover* y *Coat* tienen sus puntos en el mismo rango, estando muy acopladas. Asimismo, resulta interesante que las clases *Ankle boot*, *Sandal* y *Sneakers* estén cerca entre si, siendo las 3 clases de calzado.
La visualización de *UMAP* es aun más clara a la hora de mostrar las diferencias y similitudes que observamos:

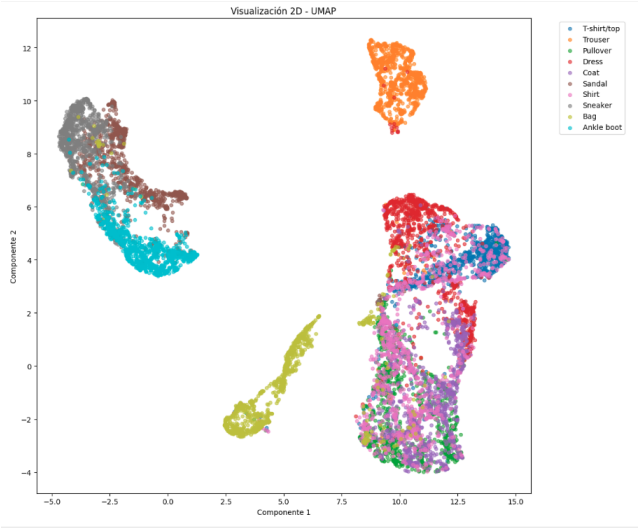

Fig. 5. Visualización 2D con UMAP

Se puede apreciar que el grupo, antes destacado de *Sandal*, *Ankle boot* y *Sneaker* estan juntos, pero aun separados en el costado izquierdo. *Bag* y *Trouser* estan, mientras que *Shirt*, *T-Shirt*, *Coat* y *Pullover* vuelven a estar sumamente acoplados entre si.
Estas observaciones responden a los resultados que se obtuvo en las técnicas de reducción.Sin embargo, en las 5 técnias implementadas, el modelo con el mejor f1-score, tenía siempre las peores métricas en la clase *Shirt*.

C. Análisis detallado de los resultados

El mejor resultado que obtuvimos: PCA + SVM tuvo la siguiente matriz de confusión:

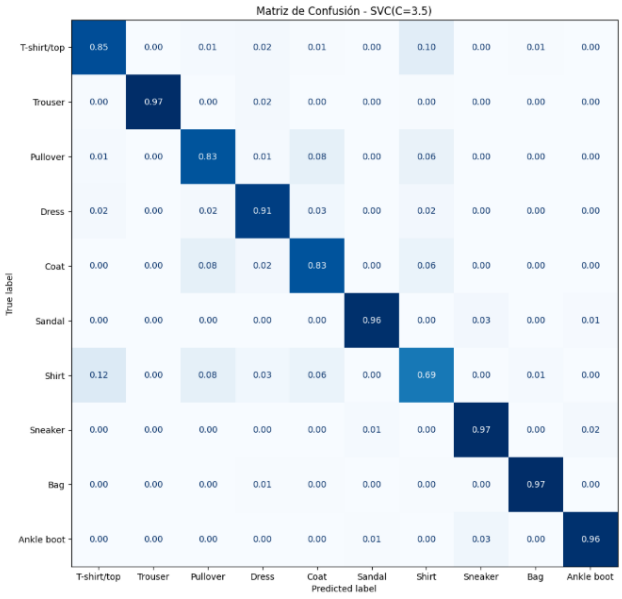

Fig. 6. Matriz de confusión para PCA + SVM

Se puede observar que *Shirt* tiene una notoria diferencia en precisión en comparación con el resto de clases. Para cuantificar esta diferencia, se presenta un análisis estadístico de las clases que mayor confusión generan con *Shirt*: *T-Shirt*, *Pullover* y *Coat*.

Para poder evidenciar mejor la disparidad en el rendimiento de *Shirt*, se calculan las medidas de tendencia central y dispersión:

- Precisión promedio: $\bar{P} = 0.808$
- Desviación estándar (Precisión): $\sigma_P = 0.0454$
- Precisión de *Shirt*: $P_{\text{Shirt}} = 0.742$ (diferencia: -0.0655, equivalente a $-\mathbf{1.44}\sigma$)
- Recall promedio: $\bar{R} = 0.798$
- Desviación estándar (Recall): $\sigma_R = 0.0652$
- Recall de *Shirt*: $R_{\text{Shirt}} = 0.690$ (diferencia: -0.1075, equivalente a $-\mathbf{1.65}\sigma$)

Los resultados evidencian que *Shirt* se desvía significativamente del desempeño de las clases más similares. El valor de precisión de *Shirt* se ubica a 1.44 desviaciones estándar por debajo de la media, mientras que el recall se encuentra a 1.65 desviaciones estándar por debajo. Esto confirma que la clase *Shirt* presenta mayores dificultades en su clasificación, particularmente en la diferenciación con respecto a *T-Shirt*, *Pullover* y *Coat*, tal como se evidencia en la matriz de confusión. Este análisis ya se podía vaticinar desde el análisis visual preliminar que se hizo en la **Fase 1**, sin embargo con la data de los experimentos se ha podido concluir que *Shirt* causa problemas para los clasificadores.

Sin embargo este resultado no es único de la combinación de *PCA + SVM*. Se pudo identificar las mismas tendencias en todas las mejores combinaciones de tecnica + clasificador. En la siguiente tabla se pueden apreciar los resultados de *Shirt* en los mejores resultados de cada técnina con clasificador.

TABLE II
RESULTADOS DE *shirt* EN LOS MEJORES RESULTADOS DE CADA TÉCNICA CON CLASIFICADOR.

Técnica	Modelo	Accuracy	Precision	Recall	F1-Score
PCA	SVM	0.69	0.74	0.69	0.72
NMF*	SVM	0.57	0.62	0.57	0.60
t-SNE	–	–	–	–	–
UMAP	RF	0.51	0.56	0.51	0.53
Spectral E.	RF	0.53	0.62	0.53	0.57
Isomap	RF	0.48	0.39	0.48	0.43

En la siguiente tabla se pueden apreciar los mejores resultados de cada técnica utilizada con el clasificador que consiguió el mejor f1-score.

* *Nota:* SVM y Random Forest obtuvieron los mismos resultados en NMF, por lo que se escogió solamente 1.

Por otro lado la matriz de confusión de UMAP muestra lo que se pudo observar en la visualización 2D: *Sneaker* y *Ankle boot* tienen muy buenos resultados: F1=0.94 y se equivocan entre sí. Asimismo, las clases *Trousers* y *Bag* también cuentan con la misma métrica, con la diferencia que no hay una clara clase con la que se confundan.

TABLE III
MEJORES RESULTADOS DE CLASIFICACIÓN POR TÉCNICA Y MODELO.

Técnica	Modelo	Accuracy	Precision	Recall	F1-Score
PCA	SVM	0.89	0.90	0.89	0.89
NMF*	SVM	0.81	0.82	0.81	0.81
t-SNE	–	–	–	–	–
UMAP	RF	0.82	0.825	0.82	0.82
Spectral E.	RF	0.84	0.84	0.84	0.84
Isomap	RF	0.81	0.82	0.81	0.81

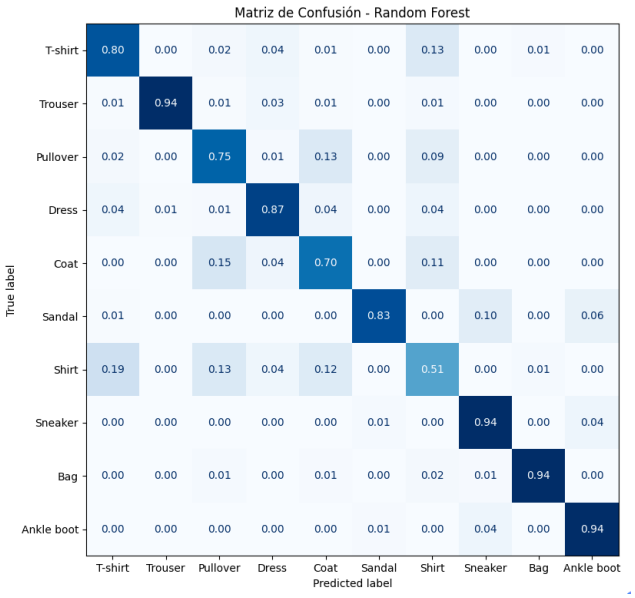

Fig. 7. Matriz de confusión para UMAP + RF

1) Analisis PCA: Primera Mejor Opción: PCA+SVM alcanza el mejor rendimiento global (F1=0.89), superando a todas las técnicas no lineales por márgenes significativos: 5 puntos sobre *Spectral Embedding*, 7 puntos sobre *UMAP*, y 8 puntos sobre *NMF* e *Isomap*. Esta superioridad se explica por:

- Estructura predominantemente lineal: El análisis de varianza revela que 215 componentes lineales (27.4% de dimensiones) capturan 95% de la varianza, indicando que *Fashion-MNIST* no requiere transformaciones no lineales complejas.
- Preservación óptima de información global: PCA maximiza varianza explicada, que para este dataset correlaciona fuertemente con separabilidad de clases (excepto *Shirt*, cuyas características residen en componentes de baja varianza).
- Determinismo y estabilidad: A diferencia de técnicas estocásticas *(UMAP, t-SNE)*, PCA produce resultados reproducibles sin depender de inicializaciones aleatorias o hiperparámetros sensibles.

- Eficiencia computacional: Tiempo de entrenamiento y transformación órdenes de magnitud menor que técnicas de manifold learning.

2) Spectral Embedding: Segunda Mejor Opción: Spectral Embedding alcanza F1=0.84, posicionándose como la segunda mejor técnica. Este desempeño se debe a:

- Balance local-global: La descomposición espectral del Laplaciano captura tanto estructura de vecindad (local) como topología global del grafo.
- Determinismo: Produce embeddings consistentes (dado un grafo fijo), garantizando reproducibilidad.
- Fundamento matemático sólido: Minimiza una función de energía bien definida que penaliza desconexiones en el grafo.

Sin embargo, su rendimiento inferior a PCA (5 puntos porcentuales) indica que la construcción explícita de grafos de vecindad no ofrece ventajas sobre proyecciones lineales para este dataset específico.
Adicionalmente, Spectral Embedding, a diferencia de las demás técnicas implementadas, no tenía un transformer en su implementacion de la librería *sklearn* por lo que se tuvo que programar uno, de la mano de otras clases de la librería *sklearn*.

3) Técnicas Lineales vs No Lineales: Los resultados muestran una clara ventaja de las técnicas lineales *(PCA)* sobre las demás en términos de rendimiento global. PCA+SVM alcanza un *F1-Score* de 0.89, superando incluso a técnicas no lineales. Este hallazgo es contraintuitivo, considerando que las técnicas no lineales están diseñadas para capturar estructuras más complejas.

La explicación radica en varios factores:

- Dimensionalidad efectiva: Fashion-MNIST, a pesar de tener 784 dimensiones originales, presenta una estructura predominantemente lineal, como lo demuestra el análisis de varianza explicada de PCA.
- Preservación de información global: PCA preserva la máxima varianza global, lo que resulta beneficioso para la mayoría de las clases. Las técnicas no lineales, al enfocarse en preservar relaciones locales, pueden perder información discriminativa a escala global.
- Estabilidad numérica: PCA es determinística y no depende de inicializaciones aleatorias ni hiperparámetros complejos, a diferencia de t-SNE y UMAP.

4) Mejores modelos: Los mejores modelos consistentemente fueron Random Forest y SVM, mientras que la Regresión Logística y KNN no fueron en ninguna técnica los mejores.

VII. Conclusiones y Recomendaciones

A. Conclusiones

- PCA+SVM es la combinación óptima: Alcanza el mejor *F1-Score* (0.89) con la mayor eficiencia computacional. La simplicidad de PCA no implica inferioridad; de hecho, es la técnica más efectiva para este dataset.
- La estructura de *Fashion-MNIST* es predominantemente lineal: El éxito de PCA sobre técnicas no lineales sofisticadas indica que la dimensionalidad intrínseca del dataset se captura mejor mediante proyecciones lineales que preservan máxima varianza.
- La clase *Shirt* es sistemáticamente problemática: Con desviaciones de 1.44σ y 1.65σ en precisión y recall respectivamente, *Shirt* presenta confusión inherente con *T-Shirt*, *Pullover* y *Coat*. Este problema persiste en todas las técnicas, sugiriendo ambigüedad en el dataset mismo.
- Maximizar varianza no garantiza maximizar separabilidad: Las características discriminativas de *Shirt* residen en componentes de baja varianza (componentes 51+), lo que explica su bajo rendimiento incluso con 95% de varianza retenida.
- Clases como *Bag* o *Trousers* presentan los mejores resultados, siendo ello correlatorio con lo observado en las reducciones de dimensionalidad en 2D, puesto que siempre eran las clases con mayor alejamiento del resto y menor acoplamiento con otras clases .
- NMF presenta el peor rendimiento: Con F1=0.81, la restricción de no-negatividad limita severamente su capacidad discriminativa, aunque ofrece ventajas en interpretabilidad.

B. Recomendaciones

- Para exploración y visualización: UMAP proporciona embeddings 2D/3D de alta calidad que preservan estructura global y local.
- Para datasets con estructura no lineal compleja: Considerar UMAP o Spectral Embedding, pero validar que justifiquen el costo computacional adicional mediante validación cruzada.
- Para mejorar clasificación de *Shirt*: Se recomienda:
  1) Aumentar dimensionalidad reducida (k=215-300) para capturar componentes de baja varianza.
  2) Considerar técnicas de *data augmentation* específicas para clases confusas.
  3) Explorar métodos de aprendizaje de métricas *metric learning* que enfaticen separación entre clases similares.

C. Limitaciones y Trabajo Futuro

1) Limitaciones:

- Al tratarse de imagenes de 28 x 28 píxeles, se pierde mucha información de detalles. Para clases que no tienen mucha similitud con el resto de clases como por ejemplo *Bag* o *Trousers* no hay problema. Sin embargo, cuando se trata con clases con mayor similitud como *Shirt* y *T-Shirt*, más informacipodría haber sido útil.
- En técnicas con coste computacional alto se tuvo que aplicar sampling de la data, ya que consumía muchos recursos y usualmente los programas fallaban si se usaban muchos datos de entrenamiento.
- En técnicas como Spectral Embedding, la redución de dimensionalidad con todo el dataset tomaba aprox. 40

minutos, sin contar el posterior entrenamiento de los clasificadores. Esto dificultó el desarrollo del proyecto, puesto que cuando se queriía alterar hiperparametros como por ejemplo el número de componentes, se tuvo que hacer una reducida cantidad de veces, puesto que el tiempo que demoraba en correr el programa era muy elevado.

2) Trabajo Futuro:

- Utilizar equipos con mayor capacidad de recursos para poder entrenar modelos con más datos
- Explorar técnicas híbridas que combinen fortalezas de métodos lineales y no lineales.
- Investigar autoencoders variacionales (VAE) y técnicas de deep learning para reducción de dimensionalidad.
- Aplicar técnicas de explicabilidad (e.g., SHAP, LIME) para entender qué características captura cada método.
- Evaluar robustez ante ruido y datos adversariales.

REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis* (Springer Series in Statistics), 2nd. New York, NY: Springer, 2002, Comprehensive treatment of PCA and its extensions. DOI: 10.1007/b98835.

[2] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011, Introduces randomized SVD algorithms for efficient PCA computation. DOI: 10.1137/090771806.

[3] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999, Original paper introducing multiplicative updates for NMF. DOI: 10.1038/44565.

[4] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008, Seminal paper introducing t-SNE for nonlinear dimensionality reduction.

[5] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018, Preprint describing the UMAP algorithm and its theoretical foundations.

[6] N. Ben-Ari, A. Yacobi, and U. Shaham, "Generalizable spectral embedding with an application to umap," *Transactions on Machine Learning Research*, 2025, Published in 07/2025.

[7] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.