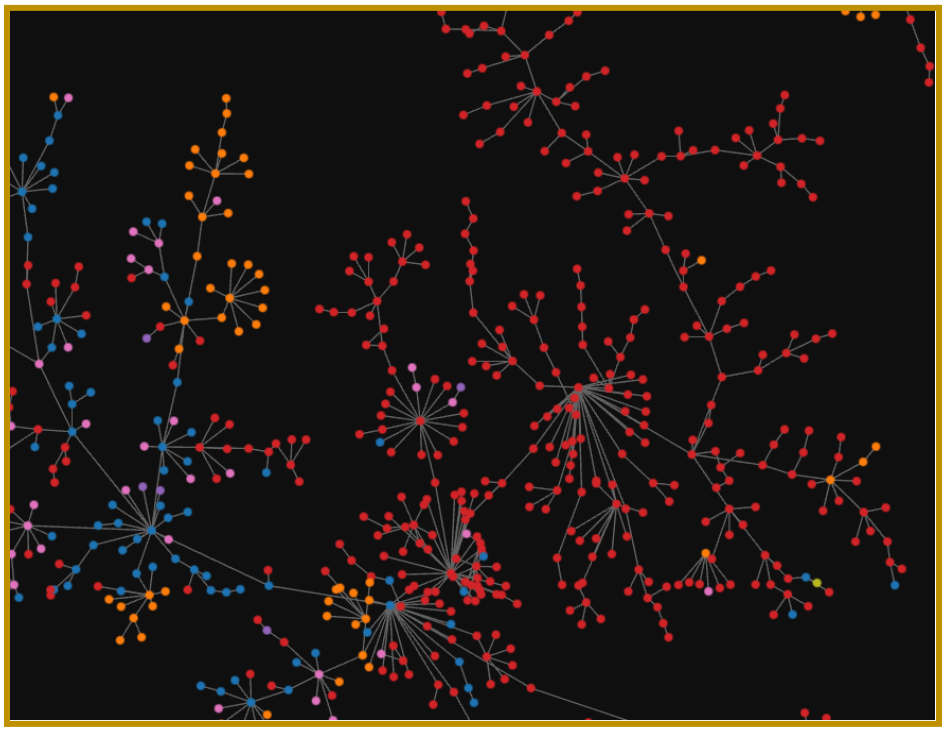
Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías

Análisis de Clustering con TMAP en base de datos



Asignatura: Análisis de Algoritmos
Sección: D06 **Calendario:** 2025 B
Profesor: Lopez Arce Delgado Jorge Ernesto

Integrantes del Equipo
Nombre: Gutierrez Vazquez Axel Código:220575328
Nombre: Quintero Arreola Laura Vanessa Código: 220577347

Fecha de entrega: 11 de octubre de 2025

Índice

Introducción

En el análisis de grandes volúmenes de datos, especialmente aquellos con alta dimensionalidad, resulta fundamental aplicar técnicas que permitan visualizar y comprender la estructura subyacente de la información. En este contexto, el algoritmo TMAP (Tree Map Approximation) se presenta como una herramienta eficiente para representar gráficamente conjuntos de datos complejos mediante una estructura jerárquica similar a un árbol, facilitando la detección de patrones, grupos y relaciones entre las muestras.

En esta práctica se utiliza el conjunto de datos Fashion-MNIST, el cual contiene imágenes en escala de grises de diferentes artículos de ropa y accesorios, como camisetas, zapatos, bolsas, entre otros. El objetivo principal es aplicar TMAP para realizar una reducción de dimensionalidad y visualizar cómo se agrupan los diferentes tipos de prendas según sus características visuales.

Además, se hace uso de librerías como Pandas, NumPy, Matplotlib y Scikit-learn para la carga, preprocesamiento y normalización de los datos, así como para complementar el análisis visual y la generación de reportes. Mediante esta actividad, se adquiere experiencia práctica en el uso de herramientas de machine learning y análisis de datos, explorando cómo la visualización basada en grafos puede ofrecer una perspectiva más intuitiva sobre la organización interna de un dataset complejo.

Objetivo

- Aprender a utilizar librerías de Python como TMAP, Pandas y Matplotlib para la reducción de dimensionalidad y visualización de grandes conjuntos de datos.
- Aplicar la técnica TMAP para analizar la estructura interna de un conjunto de datos de alta dimensión (Fashion-MNIST) y representar sus relaciones de manera visual.
- Identificar y explorar clusters y subclusters dentro del conjunto de datos, comprendiendo cómo se agrupan las instancias con características similares.
- Desarrollar habilidades en la manipulación de datos con Pandas, separando variables de características (X) y etiquetas (y) para el análisis.
- Implementar visualizaciones interactivas que permitan interpretar de forma intuitiva los resultados del proceso de reducción de dimensionalidad.
- Fomentar el análisis crítico de los resultados obtenidos, comparando la agrupación generada con las categorías reales del dataset.

Investigación Previa

Definición de Clustering

Consiste en agrupar un conjunto de datos en grupos o clusters que compartan características similares entre sí, pero que sean diferentes de los grupos adyacentes. Mediante este método se pueden descubrir patrones o estructuras inherentes en los datos, así como identificar relaciones entre variables. Este proceso es conocido como análisis de cluster o clustering, busca clasificar los datos en función de su semejanza o distancia dentro del espacio de características, permitiendo así una organización lógica de la información.

Esta técnica resulta útil para reducir la complejidad de un conjunto de datos, ya que permite representar la información de forma más ordenada y comprensible.

Definición de Base de datos

Una base de datos es un archivo digital organizado donde se guarda y protege una gran cantidad de información, se podría ver como una biblioteca bien estructurada para datos, que permite no solo almacenarlos, sino también administrarlos y mantenerlos seguros. Su principal función es permitir que las empresas y organizaciones manejen y guarden su información, esta información puede ser de cualquier tipo, ya sea desde datos de sus empleados hasta registros de que se ha hecho en la empresa/organización. Tomando esto en cuenta las bases de datos son fundamentales para que los datos sean accesibles y coherentes para quienes los necesitan, ya sean personas o aplicaciones.

Reducción de dimensionalidad

La reducción de dimensionalidad es un método que se emplea para poder representar un conjunto de datos con alta dimensión y poderlos representar con una menor dimensión, al hablar de dimensiones nos referimos a las características que conforman a un dato, entonces para poder representar un conjunto de datos complejos podemos emplear la reducción de dimensionalidad para que analicemos qué características de un dato son redundantes o menos importantes pero que el valor principal de ese dato no se vea afectado y asimismo pueda reducir su dimensión.

La importancia y la utilidad de la reducción de dimensionalidad es clave para el aprendizaje automático, ya que tener demasiadas dimensiones, aunque parezca bueno, en realidad puede llegar a causar varios problemas, principalmente, consume mucho tiempo

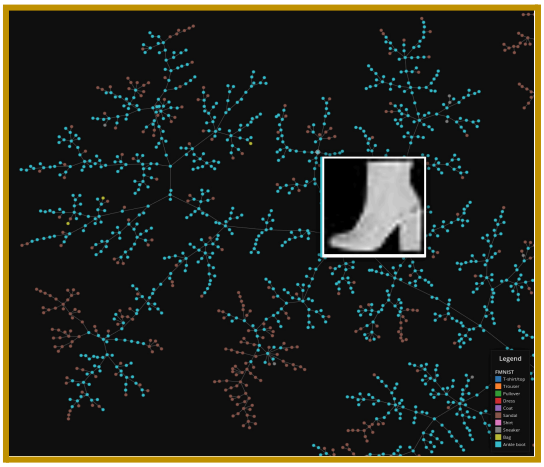
y espacio de almacenamiento, pero lo más importante es que reduce la precisión de los modelos predictivos, a este problema se le conoce como “maldición de la dimensionalidad”, esto sucede porque a medida que añadimos más y más variables, los datos se vuelven más dispersos y por ende hace que sea más difícil establecer una conexión entre todos los datos.

TMAP

Es una librería que se utiliza principalmente para diseñar conjuntos de datos muy grandes como árboles, un ejemplo muy simple de su uso es el diseño de un gráfico. Esta herramienta permite manejar información compleja mediante estructuras como árboles, mapas o redes jerárquicas, facilitando la identificación de relaciones y conexiones entre los datos

Para hacer uso de TMAP esta necesita ser importada como una librería de la siguiente manera: import tmap.

Una vez importada, la librería permite construir representaciones visuales basadas en mapas topológicos, lo que la hace especialmente útil en campos como la bioinformática, la minería de datos, el aprendizaje automático o el análisis de similitud química. En estos contextos, TMAP ayuda a identificar agrupamientos naturales (clusters) y visualizar relaciones entre miles o millones de elementos que, de otro modo, serían imposibles de analizar visualmente.



Al representar gráficos de gran tamaño, es necesario descartar o simplificar algunas aristas con el fin de obtener un diseño más claro y fácil de interpretar. De lo contrario, el gráfico resultante podría verse saturado por un exceso de conexiones e intersecciones, dificultando la identificación de las relaciones principales entre los elementos. Esta reducción o filtrado de aristas permite resaltar las estructuras más relevantes del conjunto de datos y facilita la comprensión visual de la información representada.

El uso de TMAP no solo mejora la legibilidad de los gráficos grandes, sino que también permite identificar patrones y relaciones que serían difíciles de detectar de otra manera. Al organizar los datos de manera jerárquica y optimizar la representación de los nodos y sus conexiones, los usuarios pueden analizar conjuntos de datos masivos de forma eficiente, detectando clusters, tendencias y relaciones complejas que aportan valor a la investigación, la visualización de información o el análisis científico.

UMAP

Es una librería que se utiliza principalmente para proyectar conjuntos de datos muy grandes y complejos en un espacio de baja dimensión (como un mapa 2D), facilitando su visualización. Esta herramienta permite manejar información compleja no mediante árboles, sino creando un "mapa" donde la proximidad entre los puntos refleja su similitud real, facilitando la identificación de relaciones y conexiones.

Para hacer uso de UMAP, esta necesita ser importada como una librería de la siguiente manera: `from umap import UMAP`. Una vez importada, la librería permite construir representaciones visuales que preservan la estructura topológica de los datos originales. Esto la hace especialmente útil en campos como la bioinformática, el análisis de datos, el aprendizaje automático o el análisis de imágenes. En estos contextos, UMAP ayuda a identificar agrupamientos naturales (clusters) y visualizar la "forma" de los datos que, de otro modo, serían imposibles de analizar.

Al representar datos de gran tamaño, es necesario encontrar una proyección que muestre las relaciones más importantes de forma clara. UMAP logra esto al construir un mapa donde los puntos similares se atraen y los disímiles se repelen. El resultado es un diseño más limpio donde se pueden interpretar las relaciones principales entre los elementos sin la saturación de un gráfico tradicional. Esta proyección optimizada permite resaltar las estructuras más relevantes del conjunto de datos.

El uso de UMAP no solo mejora la legibilidad de los gráficos grandes, sino que también permite identificar patrones y la forma general de los datos que serían difíciles de detectar de otra manera. Al organizar los puntos basándose en su estructura topológica, los usuarios pueden analizar conjuntos de datos masivos de forma eficiente, detectando clusters, tendencias y relaciones complejas que aportan valor a la investigación o al análisis científico.

Requisitos

Software:

- Python 3.7 + (Anaconda o Miniconda)

Librerías de Python:

- `pandas`
- `scikit-learn`
- `umap-learn`
- `scipy`
- `plotly`
- `dash`
- `Pillow`

Archivo de Datos:

- `fashion-mnist_train.csv`

Desarrollo

No solo logró realizar un cluster con la librería tmap, debido a que esta era demasiado antigua y no era compatible con ninguna version del resto del entorno, se hicieron intentos por hacer que esta funcionara de varias maneras, probando crear un entorno preferencial a lo que solicitaba tmap para funcionar correctamente y aun asi no funcionaba como en la siguiente imagen se puede observar:

```
(tmap-env) C:\Users\axelg\Downloads\test>python check.py
Traceback (most recent call last):
  File "check.py", line 1, in <module>
    import tmap
ModuleNotFoundError: No module named 'tmap'
```

la imagen muestra cómo a pesar de ya tener el tmap instalado y todo lo necesario para su ejecución, este seguía sin ser detectado, finalmente tuvimos que pensar en utilizar alternativas temporales a tmap ejecución del código.

El desarrollo del código inicia con la preparación de los datos. Se utiliza la librería pandas para leer el archivo fashion-mnist_train.csv y extraer una muestra aleatoria de un tamaño predefinido. A partir de esta muestra, se filtran los datos para aislar únicamente las filas correspondientes a la categoría de "vestido". Estos datos de píxeles, que representan las características de cada imagen, son normalizados mediante Standard Scaler de scikit-learn. Este paso es fundamental para que todas las características (pixeles) tengan la misma importancia en los cálculos posteriores. En paralelo, cada fila de datos se procesa con la librería Pillow para convertirla en una imagen, la cual es codificada a un string en formato Base64 para su posterior uso en la interfaz interactiva.

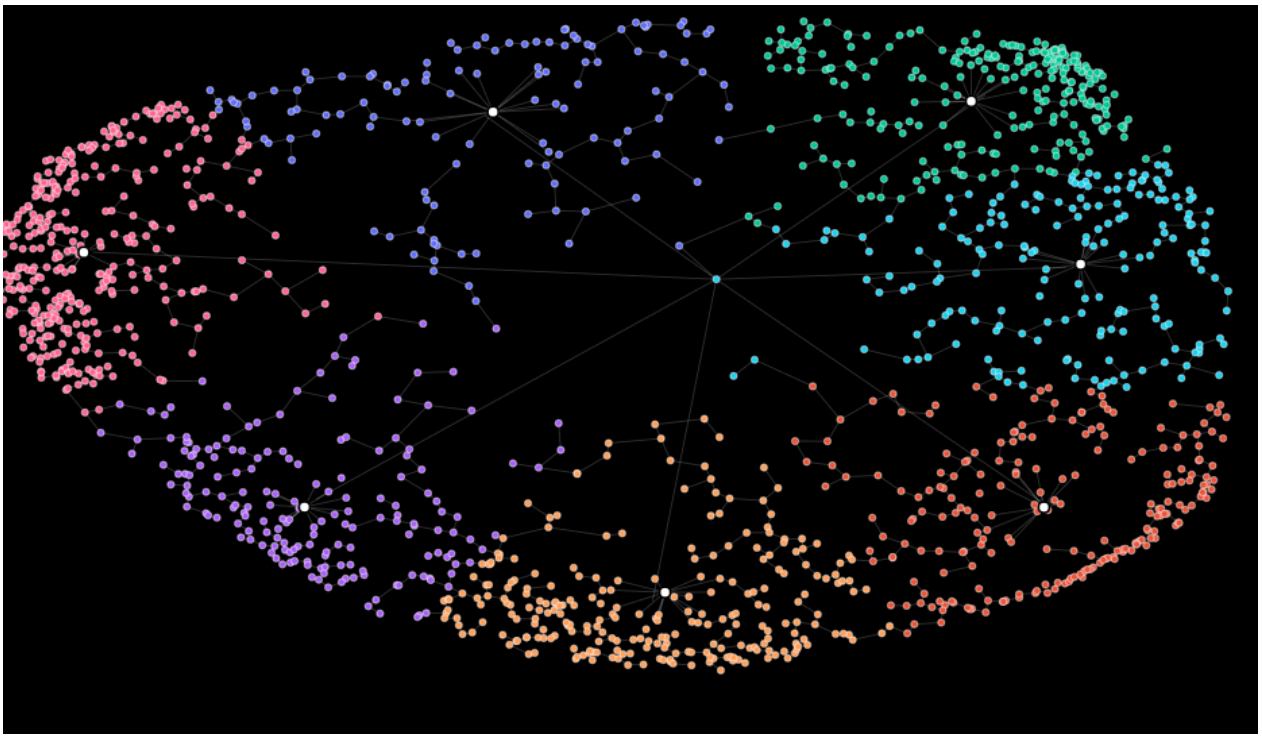
Una vez preparados los datos, se procede a la reducción de dimensionalidad para poder visualizar los puntos en un plano 2D. Se emplea el algoritmo UMAP debido a su capacidad para preservar la estructura topológica de los datos. Se configuran sus hiperparametros, como n_neighbors y repulsion_strength, para forzar una disposición global cohesiva y densa. Sin embargo, la salida natural de UMAP no garantiza una forma circular, por lo que se aplica un paso de post-procesamiento. Esta proyección polar centra la nube de puntos en el origen, convierte sus coordenadas cartesianas a polares, normaliza los radios para ajustarlos a un disco, y finalmente los convierte de nuevo a coordenadas cartesianas, moldeando la disposición en la forma circular deseada.

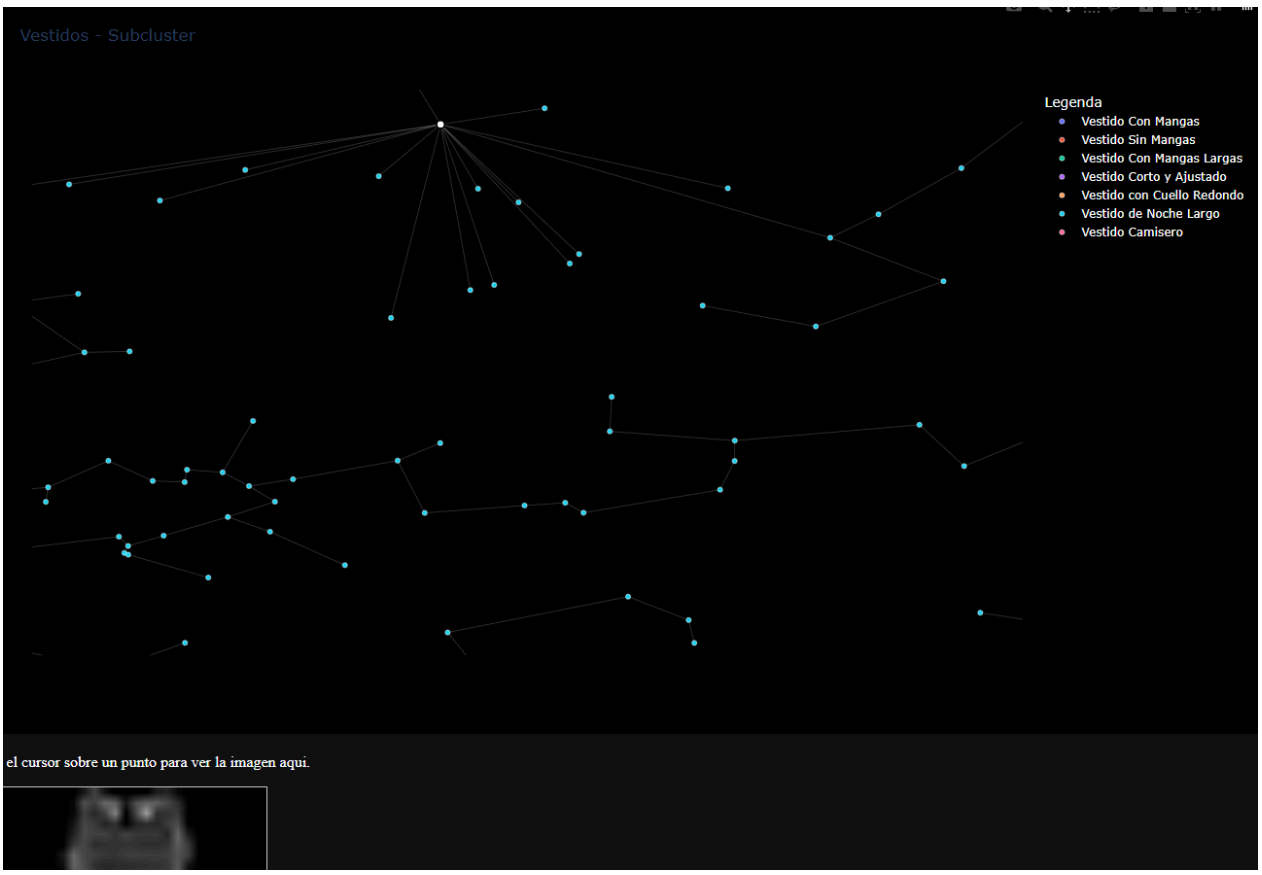
Con los puntos ya posicionados en el plano, se define la estructura de conexión y la segmentación visual. Primero, se aplica el algoritmo KMeans sobre las coordenadas 2D finales para agrupar los puntos en un número definido de subclusters, lo que permite asignar un color distintivo a cada grupo. Después, se construye una estructura de conexión híbrida. Se calcula un Árbol de Expansión Mínima (MST) sobre todos los puntos para crear una red base que conecta todo. Sobre este árbol se realiza una "cirugía": se

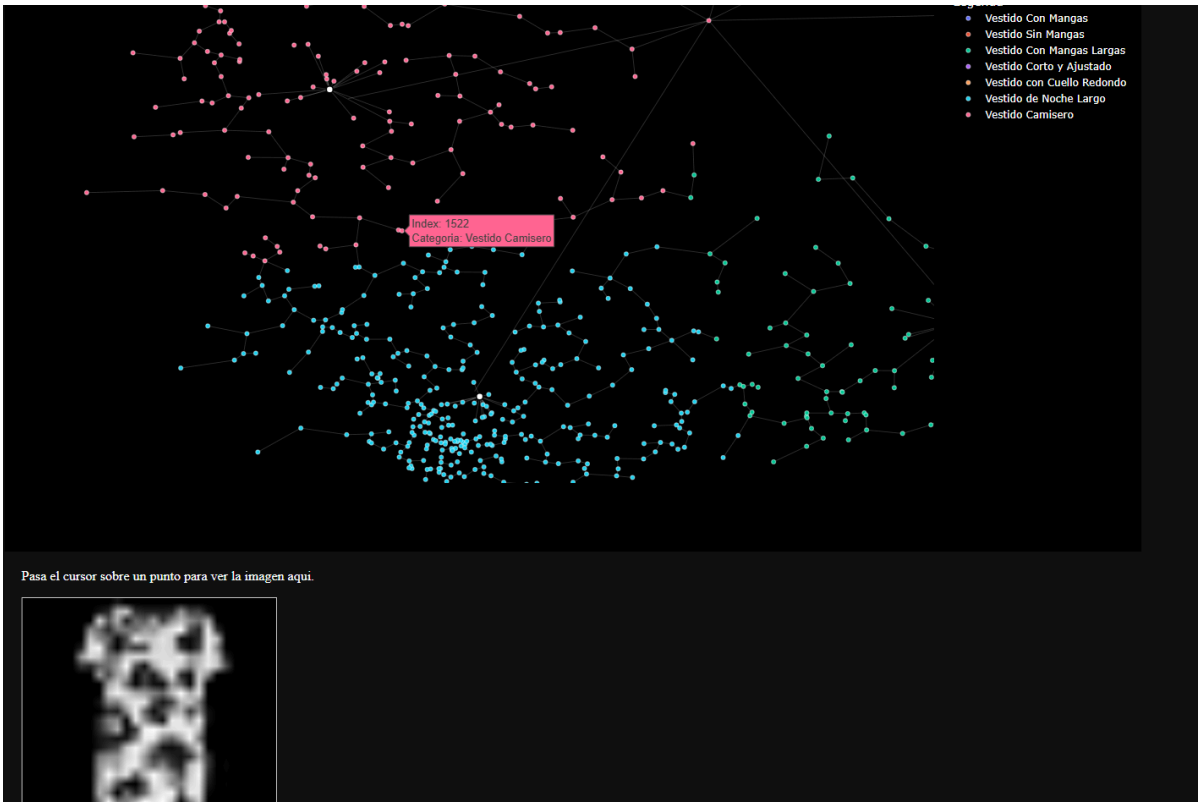
identifican "nodos centrales" en cada cluster y sus vecinos más cercanos ("nodos radiales"). Se eliminan las conexiones del MST que involucran a los nodos radiales y se reemplazan con nuevas líneas que los unen directamente a su nodo central, creando un efecto de "sol". Finalmente, se añaden "puentes" para reconectar cada sol a la estructura principal del árbol.

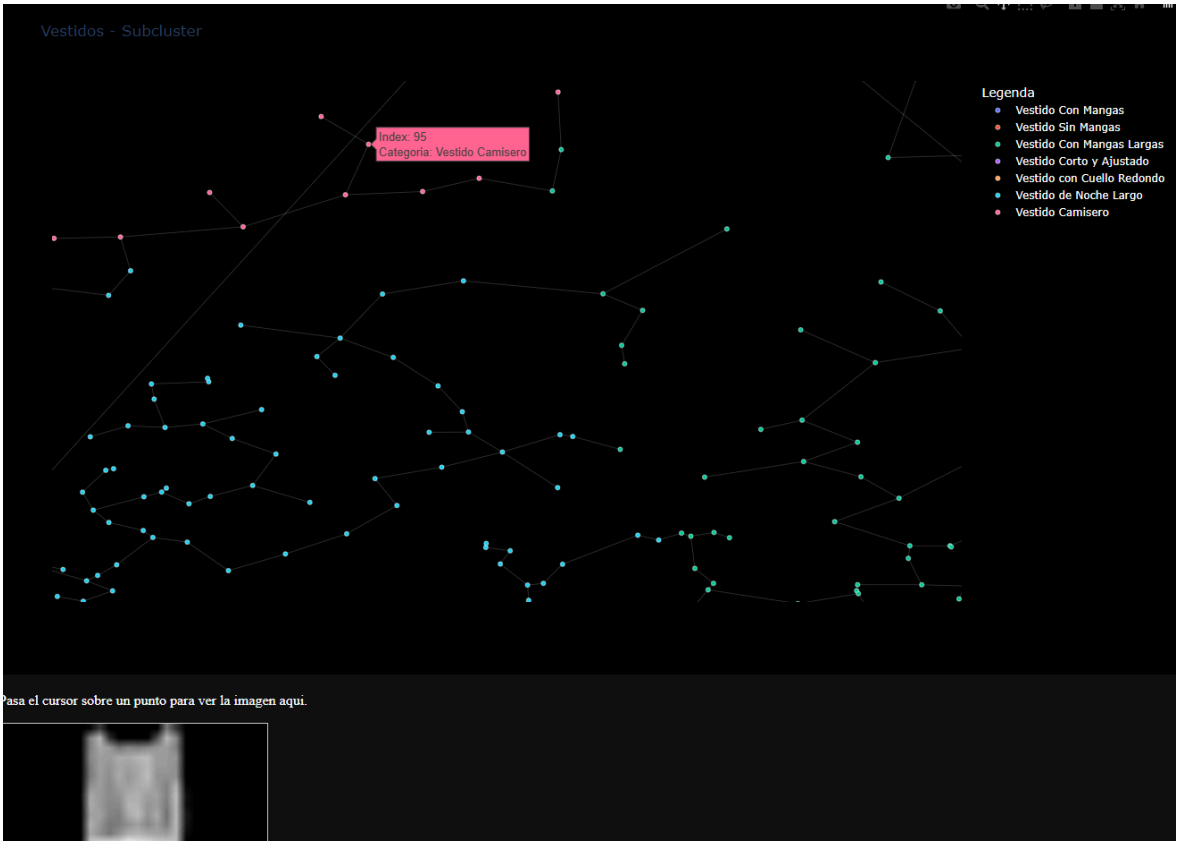
La fase final consiste en la construcción de la visualización interactiva utilizando Plotly y Dash. La información de las conexiones se dibuja como un único trazado de líneas. Luego, se añade un trazado de puntos por cada cluster, asignando el color correspondiente y almacenando el string de la imagen en Base64 en el atributo custom data de cada punto. Toda esta figura de Plotly se integra en una aplicación web Dash. La interactividad se logra mediante un callback que monitorea los eventos de hover sobre el gráfico. Cuando el cursor se posa sobre un punto, el callback extrae la información de customdata y la utiliza para actualizar un componente de imagen en un panel lateral, mostrando el vestido correspondiente al punto seleccionado.

Evidencia




Vestidos - Subcluster
Legenda
Vestido Con Mangas
Vestido Sin Mangas
Vestido Con Mangas Largas
Vestido Corto y Ajustado
Vestido con Cuello Redondo
Vestido de Noche Largo
Vestido Camisero
el cursor sobre un punto para ver la imagen aqui.


Vestido Con Mangas
Vestido Sin Mangas
Vestido Con Mangas Largas
Vestido Corto y Ajustado
Vestido con Cuello Redondo
Vestido de Noche Largo
Vestido Camisero
Index: 1522
Categoria: Vestido Camisero
Pasa el cursor sobre un punto para ver la imagen aqui.


Vestidos - Subcluster
Legenda
Vestido Con Mangas
Vestido Sin Mangas
Vestido Con Mangas Largas
Vestido Corto y Ajustado
Vestido con Cuello Redondo
Vestido de Noche Largo
Vestido Camisero
Index: 95
Categoria: Vestido Camisero
Pasa el cursor sobre un punto para ver la imagen aqui.

Conclusiones

Gutierrez Vazquez Axel
Durante el desarrollo de esta actividad se comprendió cómo formar clústeres usando distintas librerías. Aunque hubo problemas iniciales con Tmap, se logró el objetivo al implementar una solución personalizada con UMAP para la reducción de dimensionalidad, ajustando sus parámetros para obtener una forma cohesiva y circular. Sobre esta, se usó KMeans para asignar colores y se construyó un árbol. Finalmente, se integró todo en una aplicación web interactiva con Dash y Plotly, demostrando la capacidad de combinar diferentes herramientas para el análisis y visualización de datos complejos.

Quintero Arreola Laura Vanessa
Aunque en un principio no pudimos desarrollar la actividad con la librería en específico que se solicitó pudimos resolver el problema empleando otra librería, y con esta logramos concluir exitosamente la actividad aunque nos costó un poco de trabajo comprender como hacer nuestra reducción de dimensionalidad pues prácticamente estábamos trabajando con imágenes y era más complejo dividir una sección en esta cosa los vestidos en subsecciones más específicas a que solo dividir por secciones de ropa.

Referencias

Awan, A. A. (21 de enero de 2025). Comprender la reducción de la dimensionalidad. Datacamp.com; DataCamp. https://www.datacamp.com/es/tutorial/understanding-dimensionality-reduction

Introducción: clústeres. (30 junio de 2025). IBM. Recuperado 5 de octubre de 2025, de https://www.ibm.com/docs/es/was-zos/9.0.5?topic=servers-introduction-clusters

Ph.D, J. M., & Kavlakoglu, E. (5 de enero de 2024). Reducción de la dimensionalidad. Ibm.com. https://www.ibm.com/mx-es/think/topics/dimensionality-reduction

tmap. Documentación oficial, tmap.gdb.tools. Recuperado 5 de octubre de 2025, de https://tmap.gdb.tools/#support

Kosinski, M. (30 de septiembre de 2024). Database. Ibm.com. https://www.ibm.com/mx-es/think/topics/database

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426. https://arxiv.org/abs/1802.03426