UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FISICO MATEMATICAS

LICENCIADO DE CIENCIAS COMPUTACIONALES

7MO SEMESTRE

INTRODUCCIÓN AL APRENDIZAJE AUTOMÁTICO

EVIDENCIA #4 y #5

Profesor: Ángel Adrián Domínguez Lozano

Jared Evander Fuentes Sánchez	1868002
Humberto Gerardo Peña Páez	1862464
Marco Antonio Vázquez Rivera	1678576
Roberto Carlos Rangel Montenegro	1751007

San Nicolás de los Garza, Nuevo León el 05 de Noviembre del 2021

ÍNDICE

Aprendizaje no Supervisado **1**
Reducción de Dimensionalidad 1
Técnicas de Reducción de Dimensionalidad 2
Eliminación de Características 2
Reducción de Dimensionalidad 3
Lineales 3
No Lineales 4
Clusterización o Agrupamiento 5
Métodos de Clusterización 5
Tipos de Clusterización 5
Algoritmos de Agrupación 8
Similaridad 9
Disimilaridad 10

Algoritmos **10**
PCA 10
K-Means 12
Clustering Jerárquico 13
Clustering Espectral 14
TSNE 15

Plots usados en Collab **17**
PCA 17
K-Means 18
Clustering Jerárquico 19
TSNE 20

REFERENCIAS **22**

Aprendizaje no Supervisado

Los algoritmos basados en este tipo de aprendizaje no utilizan datos etiquetados previamente, en vez, tienen que agrupar datos con la ayuda de patrones y similitudes, o sea, están infiriendo cual es la estructura de los datos para realizar futuras predicciones.

El aprendizaje no supervisado está más previsto para aplicaciones del mundo real, como buscadores, identificación de células cancerosas, el área de biología y estudio de un mercado objetivo.

Debido a que los algoritmos no tienen un punto de referencia para los valores de salida, se puede caer en resultados imprecisos y poco confiables, por lo que el aprendizaje no supervisado se divide en algoritmos que usan la reducción de dimensionalidad y los que utilizan el agrupamiento para sesgar los datos y obtener mejores resultados más fiables.

Reducción de Dimensionalidad

Muchas veces, nos topamos que nuestros datos contienen características con ruido que terminan perjudicando a nuestros modelos a caer en el sobreajuste y con un lento procesamiento para clasificar datos. También podemos toparnos que nuestro conjunto de datos tienen bastantes características que incluso pueden superar a la cantidad de muestras. A este último se le conoce como la maldición de la dimensionalidad.

En respuesta a este problema, se tiene un método famoso para muchos algoritmos de aprendizaje no supervisado, la reducción de dimensionalidad, cuyo objetivo es reducir la cantidad de características (o dimensiones) a nuestro conjunto de datos, de manera que obtenemos un resultado similar a utilizar todo el conjunto para entrenar nuestro modelo.

Técnicas de Reducción de Dimensionalidad

Hay dos principales técnicas para lograr una eficaz reducción de dimensiones

Eliminación de Características

En esta técnica buscamos eliminar las características que se consideran irrelevantes, dejando un subconjunto de datos que nos entregará un modelo con un mayor accuracy, es una técnica sencilla de implementar, la desventaja es que al quitar variables, estamos dejando ir diferentes evaluaciones.

Algunos algoritmos basados en esta técnica son:

- **Ratio de valores perdidos**: se eliminan ciertas características del conjunto de datos donde hay un cierto porcentaje de datos faltantes, tomando de referencia un umbral determinado.
- **Filtro de baja varianza**: Parecido al algoritmo anterior, la diferencia es que solo se eliminan las columnas con una desviación estándar menor al umbral, la varianza está fuertemente relacionado al rango de cada columna, por lo que se recomienda normalizarlas previamente a la aplicación de este algoritmo.
- **Filtro de alta correlación**: En este caso, calculamos el coeficiente de correlación de Pearson de las columnas numéricas, y el valor de chi cuadrado de las columnas nominales en nuestro conjunto de datos, entonces buscamos quedarnos con solo una columna entre varias que comparten valores muy parecidos.
- **Eliminación de características hacia atrás**: en este algoritmo, se realizan ‘k’ iteraciones, cada una quitando una columna al conjunto de datos y calculando la tasa de error, cuando se terminan las iteraciones, se escoge el nuevo modelo con la máxima tasa de error permisible.
- **Construcción de característica secuencial hacia adelante**: a diferencia del anterior algoritmo, en vez de quitar columnas, en este algoritmo, empiezas

con una columna y en cada iteración se agrega una más, en esta ocasión se busca el mayor aumento de rendimiento.

Reducción de Dimensionalidad

En vez de quitar dimensiones, en esta técnica buscamos transformar nuestro conjunto de datos, para formar otro con menos dimensiones, pero no por ello, con menos información, esta transformación nos da la misma información son que una cantidad distinta de dimensiones.

En esta técnica, hay dos enfoques de los que se basan los distintos algoritmos.

➔ **Lineales**

Son los métodos más comunes y fáciles de entender.

Algunos algoritmos basados en esta técnica son:

- **Análisis Factorial**: reducimos la cantidad de variables del conjunto de datos a una menor cantidad de factores, estos factores son decididos por medio de transformaciones lineales con el añadido del ruido gaussiano.
- **Análisis de Componentes Principales (PCA)**: se toma las ‘n’ dimensiones numéricas pertenecientes al conjunto de datos y los transforma en un conjunto con ‘m’ componentes principales. Las restricciones son: (1) el primer componente principal posee la mayor varianza posible, (2) los componentes principales subsiguientes son ortogonales con respecto a los componentes precedentes y (3) $m<n$.
- **Análisis Discriminante Lineal (LDA)**: después de realizar una proyección del conjunto de datos, se trata de separar lo más posible de todos los tipos de clases presentes en nuestros datos.

➔ **No Lineales**

Estos algoritmos, se basan en la idea de transformar una estructura de datos no lineales de alta dimensión en concentrar toda la información en pocos colectores de menor dimensión.

Algunos algoritmos basados en esta técnica son:

- **Escala Multidimensional (MDS)**: Analizamos la similitud y disimilitud entre las columnas de nuestros datos como distancias en un espacio geométrico, usualmente estas distancias son euclidianas.
- **Mapeo de características Isométricas (ISOMAP)**: se realiza una proyección de nuestro conjunto de datos a una dimensión menor, en base a la distancia geodésica.
- **Incrustación Localmente Lineal (LLE)**: en un espacio geométrico, buscamos conjuntos de vecinos cercanos para cada punto de nuestros datos, y usando eigenvectores para encontrar la incrustación de puntos para realizar un nuevo conjunto de datos con una menor dimensión que el original.
- **Mapas Hessien (HLLE)**: Utiliza el método de vecinos próximos de LLE, pero con el operador Hessien para un resultado más normalizado.
- **Incrustación de Vecinos Estocásticos distribuidos en t (t-SNE)**: dado nuestro conjunto de datos de alta dimensión colocado en un hiperplano geométrico todos los puntos de nuestros datos, se calcula la probabilidad de relación entre ciertos pares de puntos. Al final se crea una incrustación para un nuevo conjunto de datos con menor dimensión.

Clusterización o Agrupamiento

El agrupamiento (clusterización o clustering) es otro método para manejar conjuntos de datos no etiquetados que busca agrupar datos en distintos grupos usando medidas de similaridad para encontrar patrones.

Algunas aplicaciones de este método de aprendizaje no supervisado son:

- ☐ Análisis de datos estadísticos
- ☐ Análisis de Redes sociales
- ☐ Detección de anomalías (ejemplo: el cáncer)
- ☐ Segmentación de imágenes
- ☐ Sistema de recomendación de películas

Métodos de Clusterización

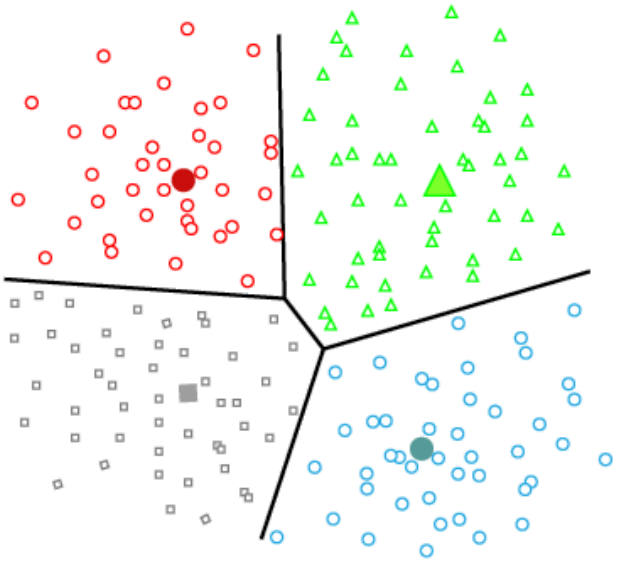
La clusterización se divide en dos subgrupos principales:

- **Hard Clustering**: cada punto de los datos pertenece a uno y solo un grupo
- **Soft Clustering**: cada punto tiene relacionado una probabilidad de pertenecer a un grupo o a otro.

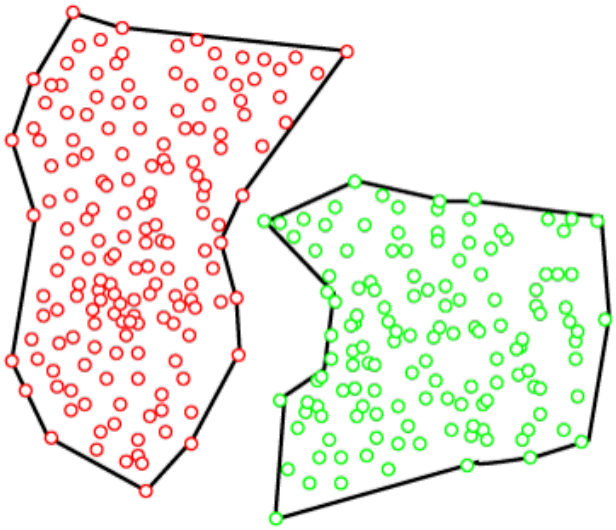
Tipos de Clusterización

Tomando de base los dos subgrupos, 5 tipos de clusterización se forman para realizar todos los algoritmos que toman este método.

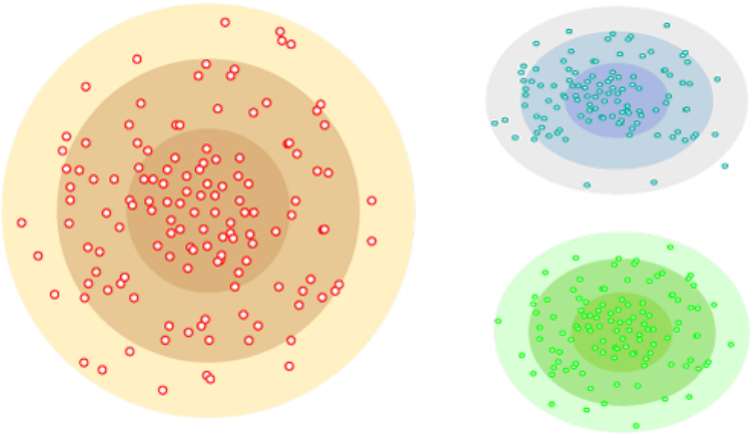
- **Clustering Particional**: también llamado método basado en centroides, dividimos nuestros datos en K grupos no jerárquicos, donde el centro de cada grupo se crea de tal manera que la distancia de todos los puntos pertenecientes a dicho grupo sea la mínima en comparación de otros grupos.



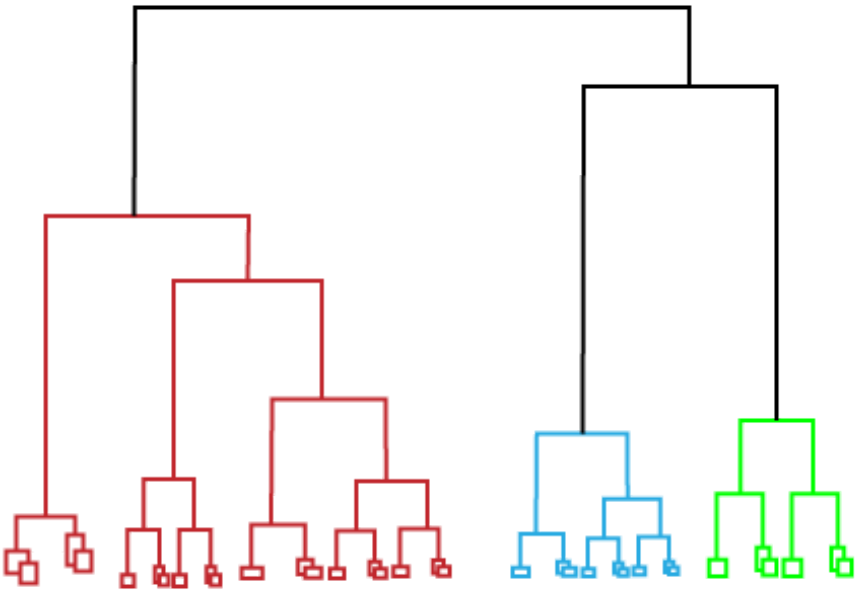
- **Clustering de Densidad**: Se forman áreas de densidad que se conectan en grupos. Los algoritmos basados en esta técnica tienen problemas para agrupar datos con densidades variables o altas dimensiones.



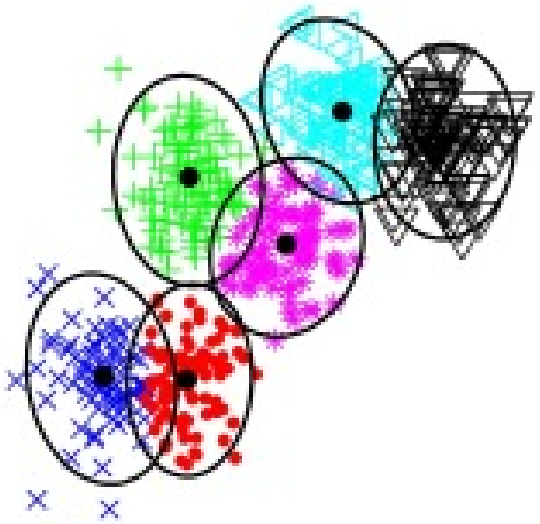
- **Clustering basado en Distribución de Modelos**: Los grupos son formados a partir de probabilidades, donde todos los datos tienen una cierta probabilidad de pertenecer a un grupo o a otro, generalmente en distribuciones gaussianas.



- **Clustering Jerárquica**: Tomando como inicio nuestro conjunto de datos de entrada, los grupos se van formando en varias iteraciones, dividiendo el conjunto, estas particiones forman un árbol o un dendrograma.



- **Clustering Espectral**: aquí los puntos están difusamente agrupados, ya que por lo general, muchos de estos pertenecen a dos o más grupos al mismo tiempo, nuestros grupos son calculados con los coeficientes de pertenencia del conjunto de datos.



Algoritmos de Agrupación

- **K-Means**: toma el conjunto de datos y lo divide en distintos grupos (especificado por el usuario) cada uno con varianzas equivalentes.
- **Cambio Medio**: busca áreas de densidad uniformes para clasificar los datos en grupos.
- **DBSCAN**: (Agrupación Espacial Basada en Densidad de Aplicaciones con Ruido en inglés) es una mejora del algoritmo de cambio medio, donde se separan las áreas de alta densidad con las de menor densidad, haciendo un mejor agrupamiento de datos.
- **Agrupación en Clústeres de Maximización de Expectativas mediante GMM**: una alternativa a K-means, donde todos los puntos son distribuciones gaussianas.
- **Aglomeración Jerárquico**: este algoritmo asume que todo el conjunto de datos es un solo grupo y conforme se realizan las iteraciones, este grupo se divide con base en distintas características. Usualmente se representa como un árbol de arriba a abajo.

Similaridad

La similaridad es encontrar semejanzas entre conjuntos de datos utilizando algoritmos que realizan cálculos para encontrar el nivel de semejanza entre los conjuntos.

Existen varios algoritmos para encontrar o calcular la similaridad, entre ellos están el algoritmo de similaridad de Jaccard, de coseno y de Pearson.

El **algoritmo de Jaccard** aplica análisis de grafos utilizando el coeficiente de Jaccard para medir la similitud entre los conjuntos de datos. El algoritmo funciona bajo la siguiente fórmula

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

en donde A y B son los conjuntos de datos y la operación resulta en el coeficiente de Jaccard que indica la similaridad entre ambos conjuntos.

El **algoritmo de similaridad de coseno** realiza el cálculo entre dos vectores diferentes de cero en donde se calcula el coseno del ángulo entre ambos para obtener el coeficiente de similaridad.

El **algoritmo de similaridad de Pearson** aplica la correlación de Pearson, donde se evalúan dos variables y se obtienen valores entre -1 y +1 para obtener correlaciones y determinar la similaridad entre los conjuntos, donde valores cercanos a -1 representan diferencia y valores cercanos a +1 representan similitud, siendo -1 totalmente diferentes y +1 totalmente similares.

Disimilaridad

Acerca de la disimilaridad, encontramos que es la distancia entre dos elementos y se define como el cálculo de desemejanza entre ellos. Dicho cálculo se denota con la siguiente fórmula

$$d(i,j)^2 = \sum_{h=1}^{p} (x_{ih} - x_{jh})^2$$

donde sí el resultado es mayor, mayor será la diferencia entre ambos.

Es el concepto inverso a la similaridad, ya que esta medida se usa para encontrar la diferencia entre en vez de la similitud entre los conjuntos.

Esto es útil en los métodos de agrupación, ya que se requiere utilizar estas medidas de distancias o disimilitud para definir las agrupaciones según el grado de disimilitud que exista.

Algoritmos

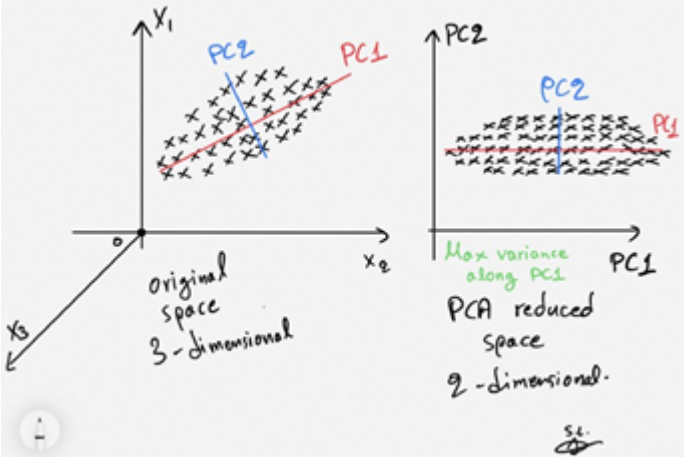
PCA

En cada problema casi siempre hay un montón de features que potencialmente podrían ayudarnos a hacer predicciones. Sin embargo, no siempre es práctico utilizarlas todas.

Incluso teniendo alta experiencia en la materia en cuestión sería difícil escoger las features que son más significativas. Ahí es donde entra el Algoritmo de Componentes Principales (o PCA en inglés).

El PCA me puede ayudar a reducir la cantidad de variables a una cantidad más pequeña pero siempre seleccionando las más significativas.

PCA nos da un set ideal de features que están ordenados por su varianza, que no están correlacionados entre sí y que son menos que en el dataset original.



El algoritmo de manera general funciona de la siguiente manera:

1. El algoritmo se pregunta cuál es la feature que tiene la más fuerte tendencia subyacente (es decir, tendencia que es real, pero podría no ser obvia). O, dicho de otra manera, la feature con más alta varianza, pues entre más alta sea la varianza más información útil puede salir de ella.
2. Luego se pregunta cuál es la feature que tiene la segunda tendencia subyacente más fuerte y que además sea independiente de la primera.
3. Después se pregunta cuál es la feature que tiene la tercera tendencia subyacente más fuerte y que además sea independiente de la primera y segunda.
4. Y así sucesivamente.

Para saber qué feature escoger el PCA utiliza:

1. La matriz de covarianza para medir la asociación entre cada variable y calcular los siguientes datos
2. Los eigenvectores para conocer las direcciones en que están dispersos los datos
3. Los eigenvalores de para saber la importancia de esas direcciones

K-Means

Cuando se quiere agrupar datos que son similares con el propósito de descubrir patrones la herramienta más sencilla y comúnmente utilizada es el algoritmo K-means.

La K es el número de clusters que el algoritmo va a buscar.

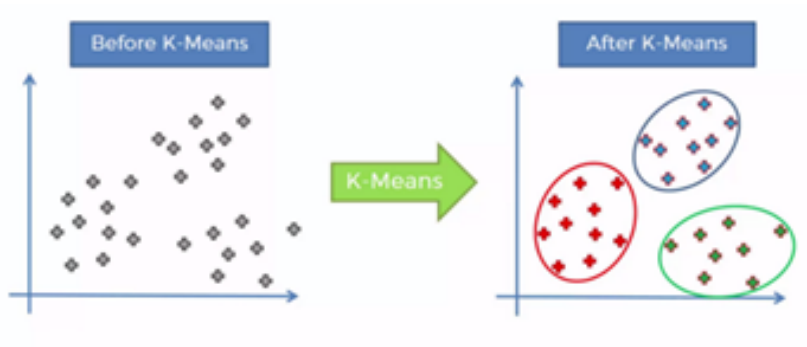
Un cluster es una serie de datos que están unidos por el hecho de tener alguna similaridad.

Para conseguir esos clusters se utilizarán K centroides (generalmente colocados al azar). Un centroide representa el centro de un cluster.

Para cada dato se define cual centroide está más cerca y el dato pasará a pertenecer a tal cluster.

Luego los centroides se actualizan para incluir a sus nuevos integrantes.

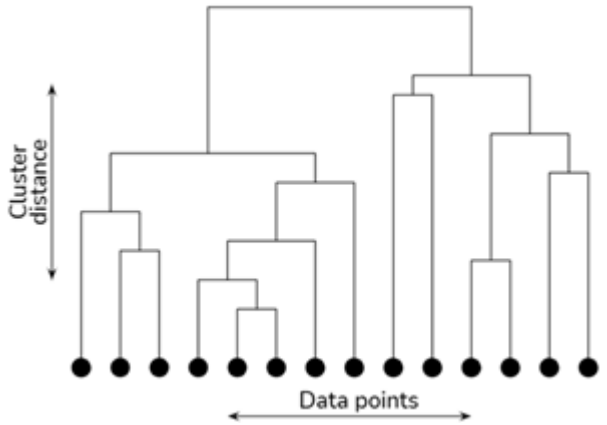
El algoritmo termina cuando se llegó al límite de iteraciones o cuando los centroides se estabilizan, o, dicho de otra manera, no se actualizan porque la clusterización ya se ha completado.



En el ejemplo de la imagen K=3.

Clustering Jerárquico

El clustering jerárquico se basa en construir un dendrograma para poder analizarlo. Un dendrograma es una especie de árbol específicamente creado para poder visualizar cómo se relacionan los datos.



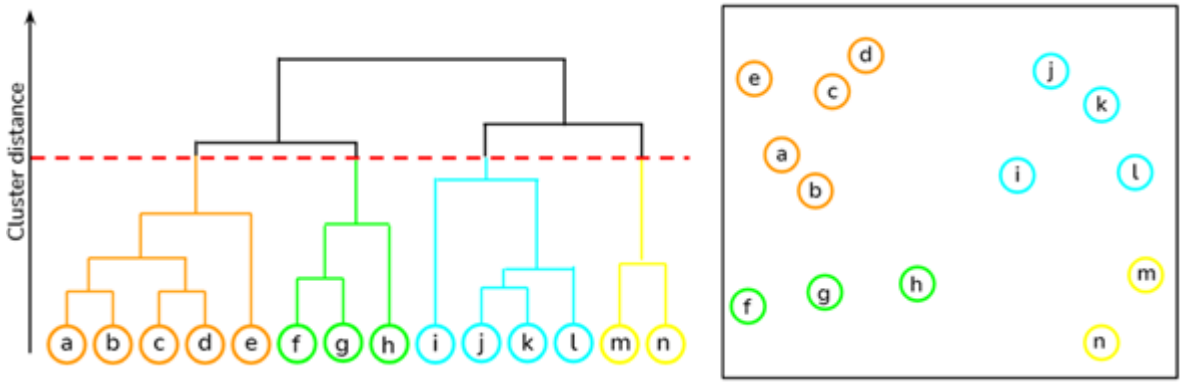
Entre más bajas por el árbol más se divide en clusters hasta llegar al fondo del árbol donde están todos los datos.

La longitud vertical de las ramas representa la longitud entre cada cluster, entre más larga sea la línea más distancia hay entre clusters.

La longitud entre clusters puede medirse con diferentes métricas.

En K-means tu eliges el número de clusters a encontrar, pero en el clustering jerárquico no. En su lugar el analista de datos decide donde se le hace un corte horizontal (no necesariamente una línea recta) al dendrograma y todos los hijos debajo del corte representan los clusters.

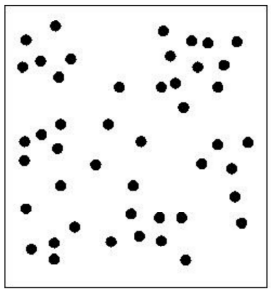
Los clusters resultantes no tienen mucho sentido por sí mismos, hubiera que marcar cada dato para visualizar las relaciones.

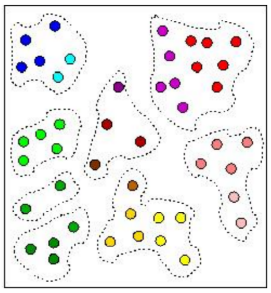


Hay dos maneras de construir el dendrograma:

1. Clustering aglomerativo: Inicialmente cada dato es un cluster, se van uniendo entre sí para formar clusters más grandes. Se termina cuando haya un solo cluster que abarque todos los datos.
2. Clustering divisivo: Inicialmente un cluster abarca todos los datos, se van dividiendo para formar clusters más pequeños. Se termina cuando se separan todos los datos o cuando encuentras el mínimo número de datos que deseas que haya en un cluster.

Clustering Espectral





Es un método de agrupación basado en la teoría de grafos, donde el enfoque se utiliza para identificar vecindades de nodos en un grafo basado en el número de aristas que se conectan. Es importante mencionar que el método es flexible y también se no da la oportunidad de agrupar datos no gráficos. En general, el gráfico no dirigido ponderado se divide en dos o más subgrafos óptimos de modo que los subgrafos sean lo más similares posible, y la distancia entre los subgrafos es lo más posible para lograr propósitos comunes de agrupamiento.

Este método de agrupación en sí misma también proporciona varios métodos de corte diferentes (corte mínimo, corte de relación, corte normalizado) y cada método

corresponde a un objetivo de optimización. Solo se tratará el más común, práctico y económico: Corte normalizado.

Los pasos para trabajar este algoritmo son:

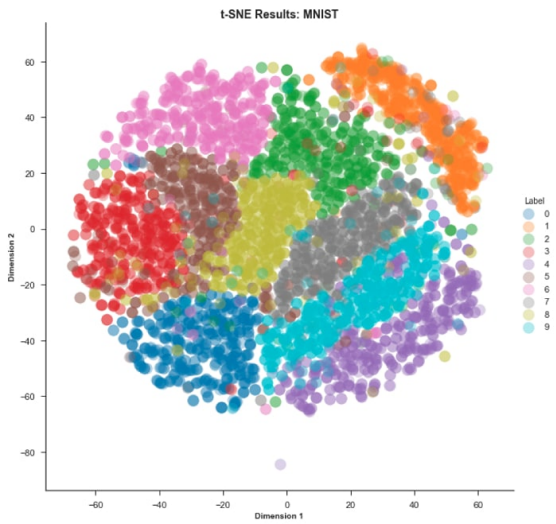
- Construir una matriz de similitud W. Hay que establecer el peso del borde de los puntos conectados en 1 y el resto en 0.
- Construir una matriz de grados D. Con grados se refiere al peso total entre los puntos asociados con el mismo. Esta matriz es una diagonal.
- Construir una matriz laplaciana L. La ecuación para obtener la matriz laplaciana es la resta de la matriz de grados y la matriz de similitud. L = D-W.
- Normalizar la matriz L. La ecuación a seguir para normalizar la matriz L es la siguiente:

$$L = D^{-\frac{1}{2}} \cdot L \cdot D^{-\frac{1}{2}}$$

La cual se transforma para obtenerla de la siguiente manera:

$$E = D^{-\frac{1}{2}} \cdot W \cdot D^{-\frac{1}{2}}$$

- Calcular los autovalores y autovectores de la matriz L. La matriz anterior se construye para reflejar la relación entre los datos de la matriz. Entonces los valores y vectores propios se calculan para reducir la dimensión de N a k.



TSNE

El algoritmo t-SNE por su siglas en inglés T-distributed Stochastic Neighbor Embedding, es diseñado para trabajar y visualizar con conjuntos de una cantidad alta de datos. Trabaja en la forma de intentar reproducir la distribución que existía en el espacio original en el espacio final.

Este se ejecuta en dos pasos. En primer lugar construye una distribución de probabilidad sobre parejas de muestras en el espacio original, de forma tal que las

muestras semejantes reciben alta probabilidad de ser escogidas, mientras que las muy diferentes reciben baja probabilidad de ser escogidas. El concepto de “semejanza” se basa en la distancia entre puntos y densidad en las proximidades de un punto. En segundo lugar, lleva los puntos del espacio de altas dimensiones al espacio de más bajas dimensiones de forma aleatoria, se define una distribución de probabilidad semejante a la vista en el espacio destino, minimizando la divergencia de Kullback-Leibler(que mide la similitud o en su caso la diferencia entre dos funciones de probabilidad) entre las dos distribuciones con respecto a las posiciones de los puntos en el mapa.

Scikit-Learn implementa este algoritmo y la clase incluye varios parámetros que definen el comportamiento del algoritmo:

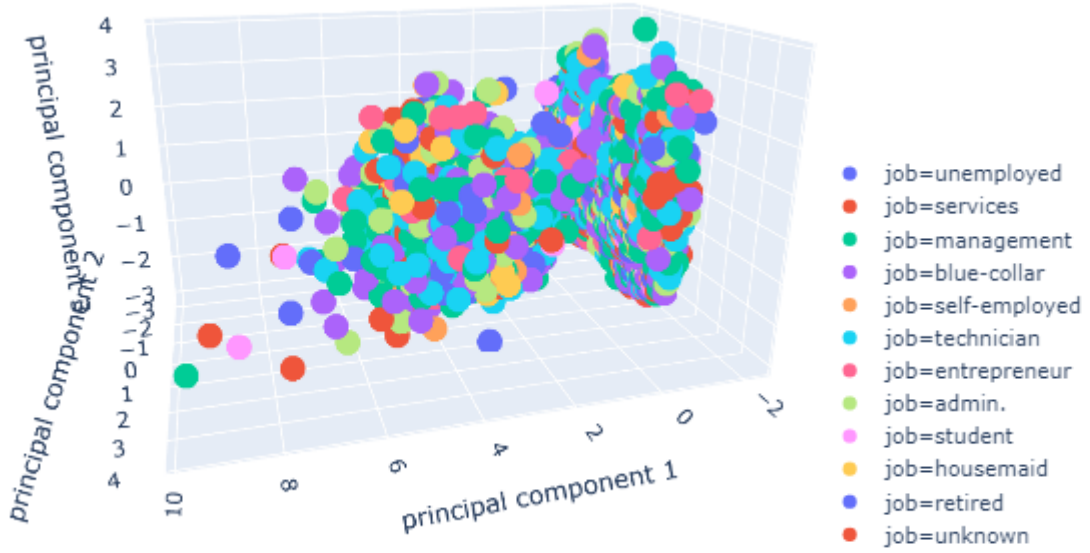
- **n_components**(default:2): Dimensiones del conjunto transformado.
- **perplexity**(default:30): Se recomienda un valor entre 5 y 50 cuanto mayor sea el dataset, aunque el algoritmo no es muy sensible a este parámetro.
- **early_exaggeration**(default:12): Este parámetro controla la distancia entre bloques semejantes en el espacio final. La elección de este valor no es crític.
- **learning_rate**(default:200): Habitualmente en el rango (10-1000). Si es muy elevado, los datos transformados estarán formados por una bola de puntos equidistantes unos de otros. Si es muy bajo, los puntos se mostrarán comprimidos en una densa nube con algunos outliers.
- **n_iter**(default:1000): Número máximo de iteraciones para la optimización. Debería ser, por lo menos 250.
- **metric**: Métrica a utilizar para la medición de distancias.
- **method**: Algoritmo a usar para el cálculo del gradiente.

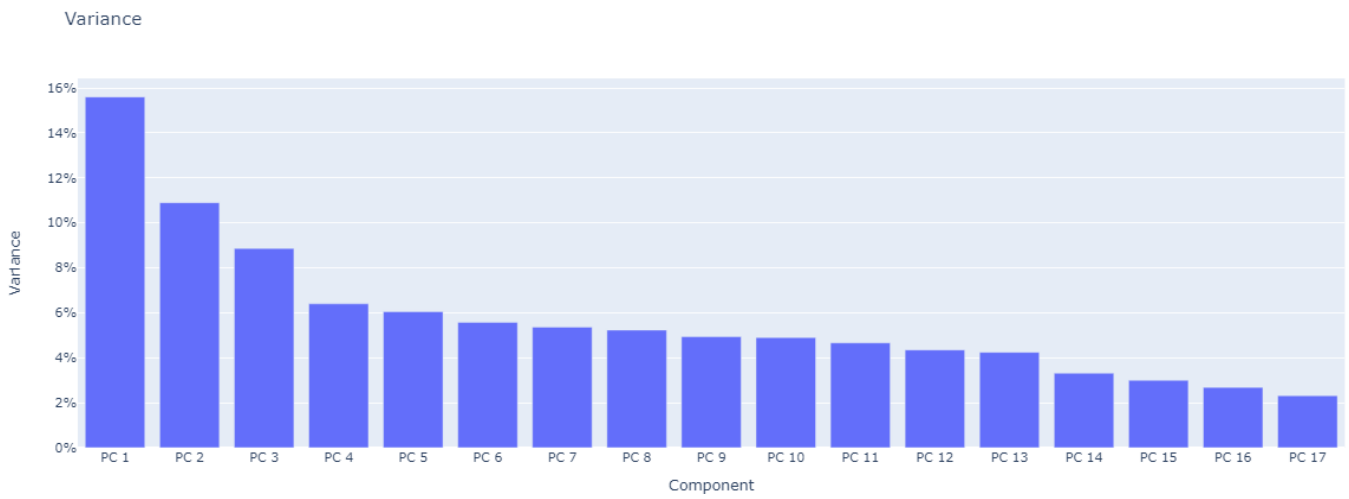
Plots usados en Collab

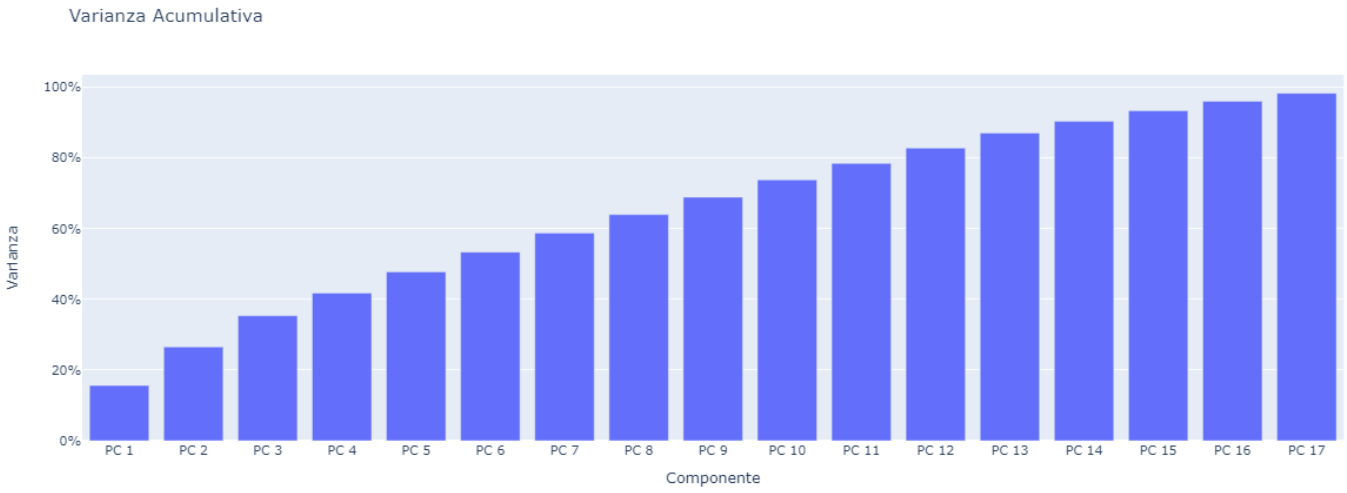
Link para el notebook en google collab

https://colab.research.google.com/drive/1HzQiuX5pE5PpcAMJXARdVxHFn8d--D4O?usp=sharing

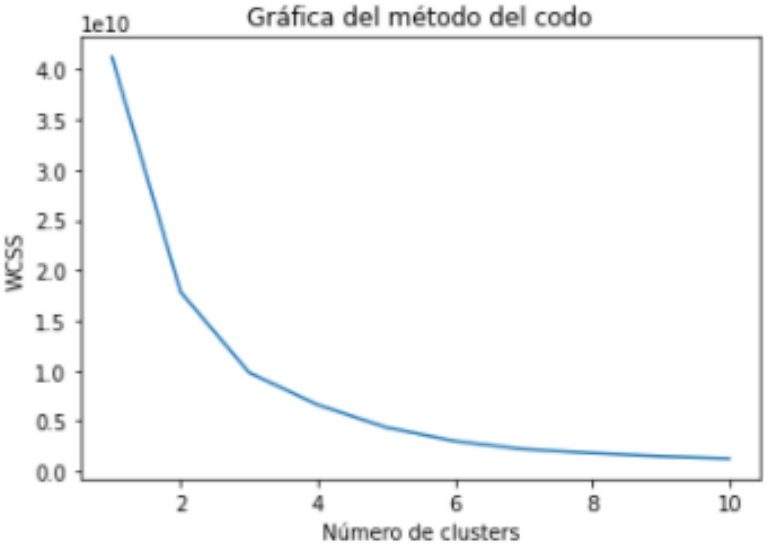
- PCA

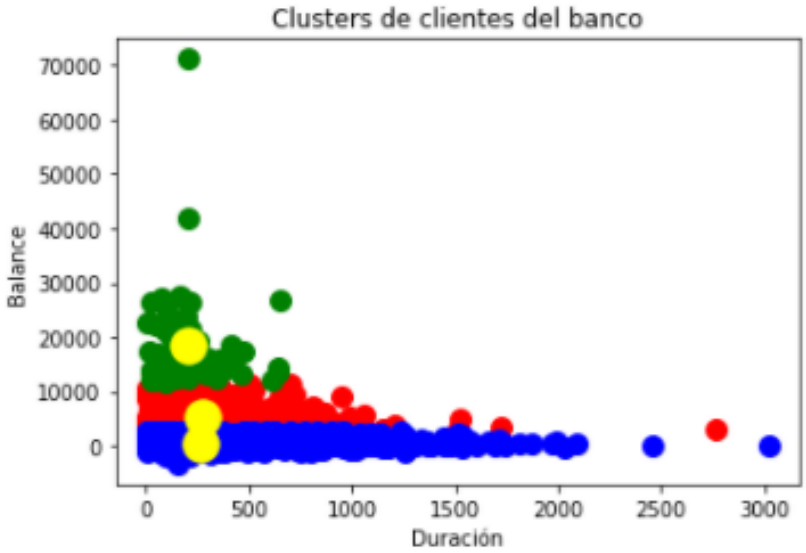




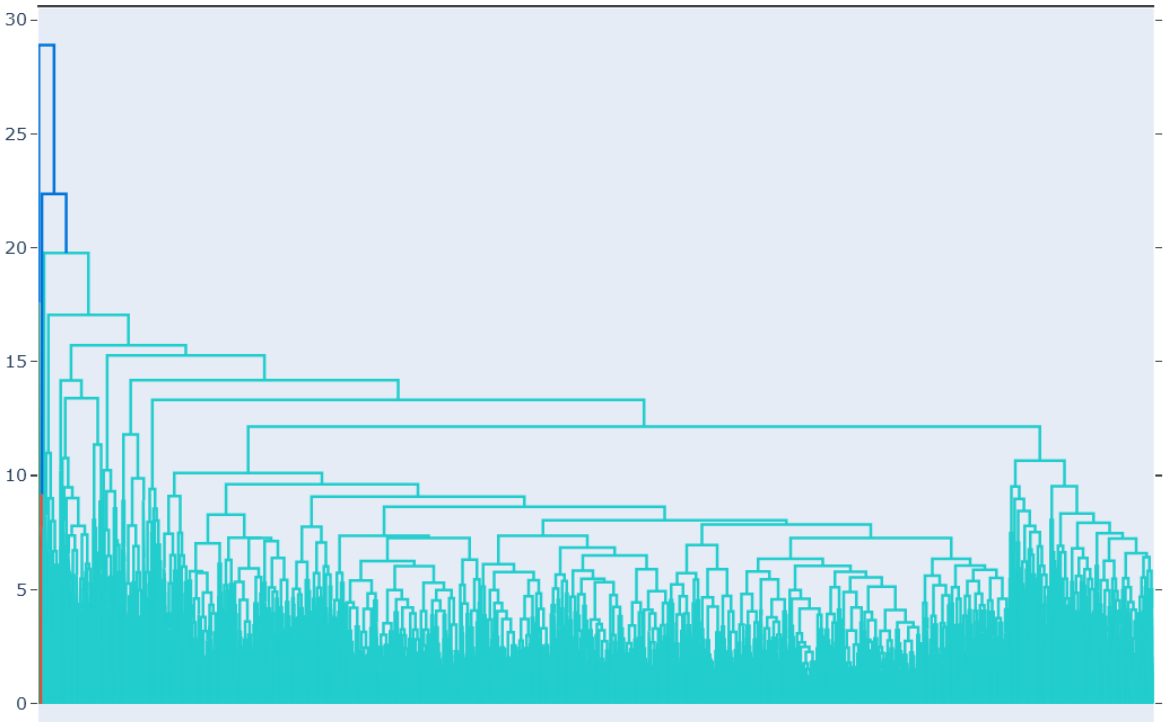

Varianza Acumulativa
100%
80%
60%
40%
20%
0%
Varianza
PC 1
PC 2
PC 3
PC 4
PC 5
PC 6
PC 7
PC 8
PC 9
PC 10
PC 11
PC 12
PC 13
PC 14
PC 15
PC 16
PC 17
Componente

- K-Means


Gráfica del método del codo
1e10
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
WCSS
2
4
6
8
10
Número de clusters

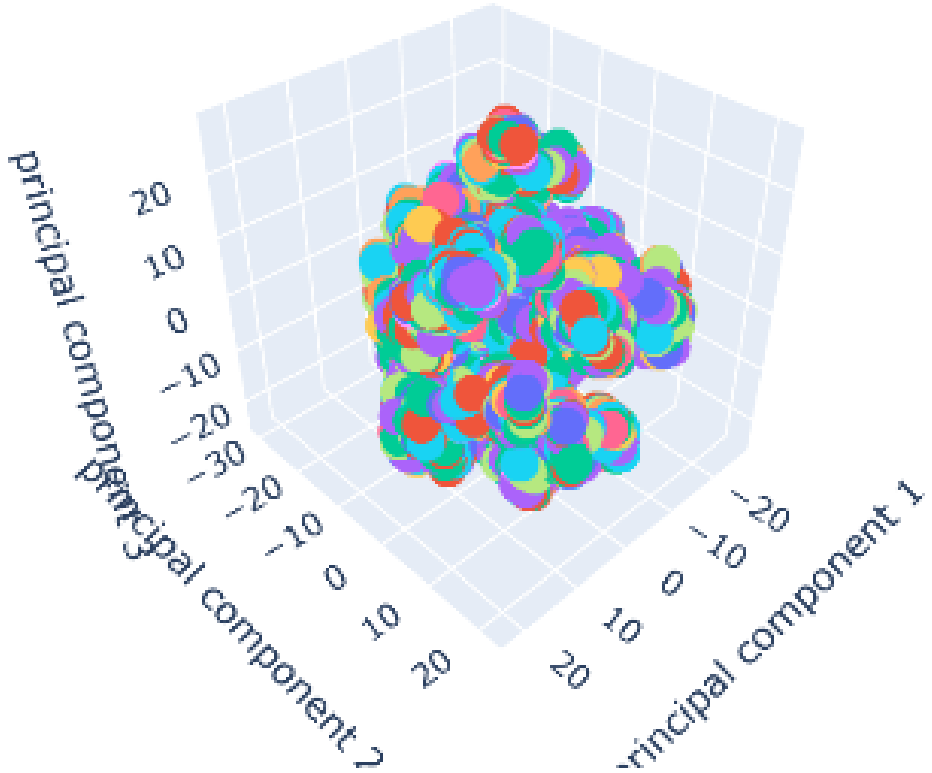


- Clustering Jerárquico

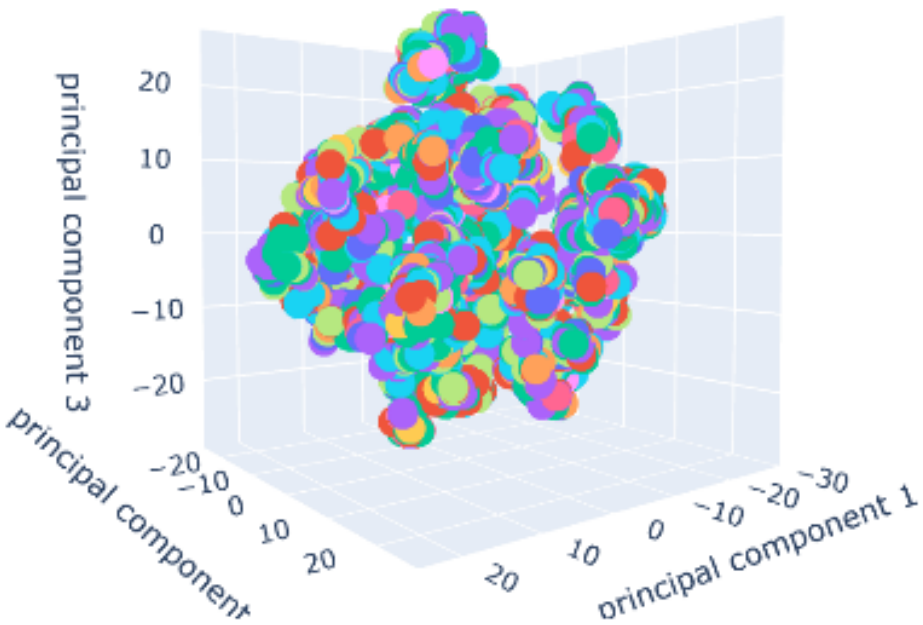


- TSNE

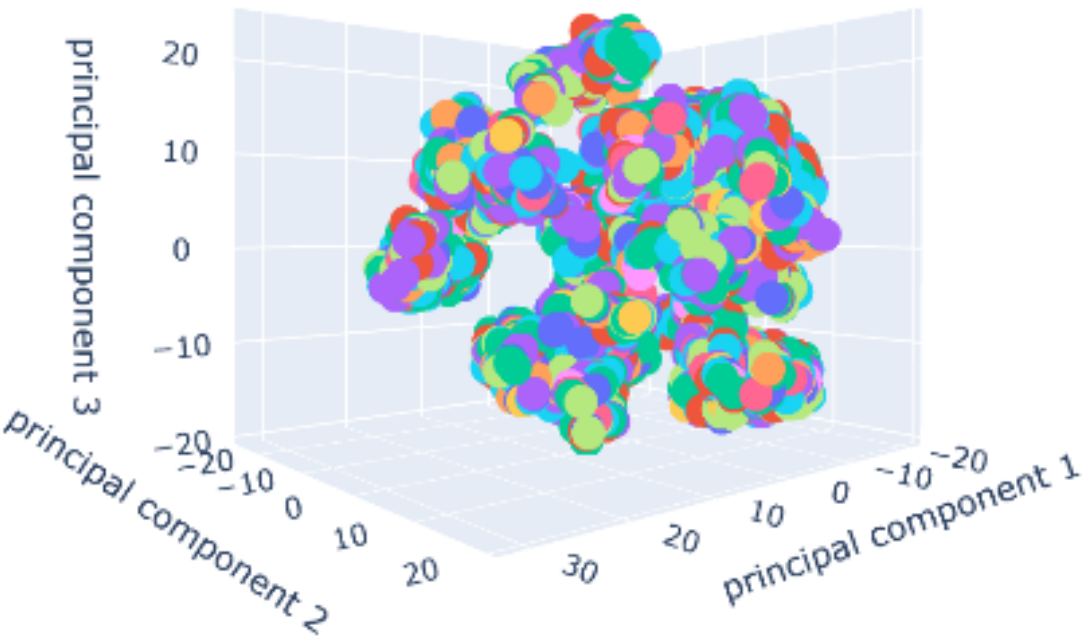
t-SNE con perplejidad 40



t-SNE con perplejidad 25



t-SNE con perplejidad 50



REFERENCIAS

[1] J. Oliver, R. Baxter & C. Wallace (2013). *Unsupervised Learning Using MML*. psu. Retrieved October 24, 2021, from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.5587&rep=rep1&type=pdf

[2] J. Dy & C. Brodley (2004). *Feature Selection for Unsupervised Learning*. Journal of Machine Learning Research. Retrieved October 24, 2021, from https://www.jmlr.org/papers/volume5/dy04a/dy04a.pdf

[3] Harmouch, M. (2021, April 2). *17 types of similarity and dissimilarity measures used in data science.* Medium. Retrieved November 4, 2021, from https://towardsdatascience.com/17-types-of-similarity-and-dissimilarity-measures-used-in-data-science-3eb914d2681

[4] Education, I. C. (2021, June 29). *Unsupervised Learning*. IBM. Retrieved November 4, 2021, from https://www.ibm.com/cloud/learn/unsupervised-learning

[5] Data, S. B. (2019, October 27). *Algoritmo t-SNE con Python, una breve introducción*. sitiobigdata.com. Retrieved November 4, 2021, from https://sitiobigdata.com/2018/08/27/algoritmo-t-sne-con-python/

[6] O. (2019, 29 octubre). *Algoritmos de Similiridad*. GraphEverywhere. https://www.grapheverywhere.com/algoritmos-de-similaridad/

[7] Nagpal, A. (2017, November 22). *Principal component analysis- Intro*. Medium. Retrieved November 5, 2021, from https://towardsdatascience.com/principal-component-analysis-intro-61f236064b38

[8] Yiu, T. (2021, September 29). *Understanding PCA*. Medium. Retrieved November 5, 2021, from https://towardsdatascience.com/understanding-pca-fae3e243731d

[9] Garbade, D. M. J. (2018, September 12). *Understanding K-means clustering in machine learning*. Medium. Retrieved November 5, 2021, from https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

[10] Pai, P. (2021, May 10). *Hierarchical clustering explained*. Medium. Retrieved November 5, 2021, from https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8