Practica 3: Análisis de Componentes Principales

Machine Learning

Castro Elvira D.
2022710168
dcastroe2100@alumno.ipn.mx

Nava Méndez E. U.
2021710144
enavam2001@alumno.ipn.mx

UPIIT: Unidad Profesional Interdisciplinaria en Ingeniería Campus Tlaxcala Instituto Politécnico Nacional, Tlaxcala, Tlaxcala, México 9000

Ingeniera en Inteligencia Artificial

12 de octubre 2023

***Resumen*— El análisis de componentes principales (Principal Component Analysis PCA) es un método de reducción de dimensionalidad que permite simplificar la complejidad de espacios con múltiples dimensiones a la vez que conserva su información. El método de PCA permite por lo tanto "condensar" la información aportada por múltiples variables en solo unas pocas componentes. Aun así, no hay que olvidar que sigue siendo necesario disponer del valor de las variables originales para calcular las componentes.**

Palabras clave* — *PCA, Iris, Varianza, Covarianza, Componentes, Estandarización

I. INTRODUCCION

En las aplicaciones de Machine Learning es común trabajar con datasets de todo tipo y de diferentes tamaños. Aunque las computadoras hoy en día pueden procesar gran cantidad de datos, siempre se busca optimizar los procesos.

Un problema frecuente en el análisis de datos es que a veces los datasets cuentan con demasiados atributos para cada elemento. Es posible que existan atributos que no aporten mucho en el análisis o que sus valores no representen una diferencia significativa entre ellos.

Esto lo podemos considerar como un problema de dimensionalidad y para evitarlo es que se recurren a técnicas como el *PCA (Análisis de Componentes Principales)*.

A. MARCO TEORICO

a. Análisis de componentes principales (PCA)

PCA es un método matemático que pertenece a los métodos de aprendizaje no supervisado (no tenemos una variable target con etiquetas) y una de sus funciones es extraer información útil a partir de las variables originales. Esto nos ayuda a simplificar análisis posteriores como la representación de los datos en dos o tres dimensiones.

b. ¿Cómo funciona el PCA?

1. ***Estandarización de los datos:*** primero, se estandariza o normaliza los datos si es necesario. Esto implica restar la media de cada característica y dividir por su desviación estándar. La estandarización es opcional pero recomendada, especialmente si las características tienen diferentes escalas.
2. ***Matriz de covarianza:*** es una medida de la relación entre las diferentes características del conjunto de datos. La matriz de covarianza nos muestra cómo las características se relacionan entre sí y si hay alguna dependencia lineal.
3. ***Valores y vectores propios:*** Usamos la matriz de covarianza para calcular los valores propios (eigenvalues) y los vectores propios (eigenvectors) de la matriz. Estos valores y vectores propios capturan la variabilidad y la dirección de máxima variación en los datos.
4. ***Ordenamiento de los valores propios:*** Ordenamos los valores propios en orden descendente. Esto nos indica cuánta varianza es explicada por cada componente principal. Los valores propios más grandes corresponden a las direcciones de máxima variación en los datos.
5. ***Selección de componentes principales:*** Elegimos un número de componentes principales que deseamos retener. Esto puede ser igual o menor que el número de características originales. En la mayoría de los casos, se eligen las primeras componentes principales que explican la mayor parte de la varianza acumulada
6. ***Construcción de las nuevas características:*** tomamos los vectores propios correspondientes a las componentes principales seleccionadas y los utilizamos para proyectar los datos originales en un nuevo espacio de características.

c. Varianza

La varianza representa la cantidad de dispersión o variabilidad en los datos originales que es explicada por cada componente principal, es decir, indica cuánta información se conserva al proyectar los datos en una dimensión reducida. Los componentes principales se ordenan de manera que el primero capture la mayor cantidad de varianza y, a medida que avanzamos en los componentes principales, la varianza explicada disminuye.

d. Eigenvectors

Son una multiplicación entre una matriz y un vector. El vector resultante de la multiplicación es un múltiplo entero del vector original. Los *eigenvectors* de una matriz son todos aquellos vectores que, al multiplicarlos por dicha matriz, resultan en el mismo vector o en un múltiplo entero del mismo. Los *eigenvectors* tienen una serie de propiedades matemáticas específicas:

- Los *eigenvectors* solo existen para matrices cuadradas y no para todas. En el caso de que una matriz *n* x *n* tenga *eigenvectors*, el número de ellos es *n*.
- Si se escala un *eigenvector* antes de multiplicarlo por la matriz, se obtiene un múltiplo de este *eigenvector*. Esto se debe a que, si se escala un vector multiplicándolo por cierta cantidad, lo único que se consigue es cambiar su longitud, pero la dirección es la misma.
- Todos los *eigenvectors* de una matriz son perpendiculares (ortogonales) entre ellos, independientemente de las dimensiones que tengan.

e. Eigenvalues

Cuando se multiplica una matriz por alguno de sus eigenvectors se obtiene un múltiplo del vector original, es decir, el resultado es ese mismo vector multiplicado por un número. Al valor por el que se multiplica el eigenvector resultante se le conoce como eigenvalue. A todo eigenvector le corresponde un eigenvalue y viceversa.

En el método PCA, cada una de las componentes se corresponde con un eigenvector, y el orden de componente se establece por orden decreciente de eigenvalue. Así pues, la primera componente es el eigenvector con el eigenvalue más alto.

f. Loadings

Las $\alpha 1j,\ldots,\alpha pj$ son las cargas o **loadings** del *m*-ésimo componente principal y cada una de ellas marca el peso que tiene cada una de las *p* variables en el componente y nos ayuda a recoger qué tipo de información recoge cada componente principal.

g. Dataset iris

El conjunto de datos Iris fue introducido por el estadístico británico Ronald A. Fisher en 1936.

- Número de instancias: 150 (50 instancias por cada una de las 3 clases)
- Número de características: 4
- Clases: Setosa, Versicolor y Virginica (3 clases)
- Atributos: Longitud y ancho del sépalo, longitud y ancho del pétalo (en centímetros).

II. DESARROLLO

1. Carga de datos y preprocesamiento de los datos:

Para cargar los datos se usó el paquete de datasets de sklearn que incluye algunos conjuntos de datos de prueba, como es el caso del dataset Iris que es utilizado comúnmente para pruebas de *Mechine Learning* debido a su tamaño pues solamente incluye 150 instancias.

```
from sklearn.datasets import load_iris
iris = load_iris()
```

Análisis de datos

a) Gráfico de cajas

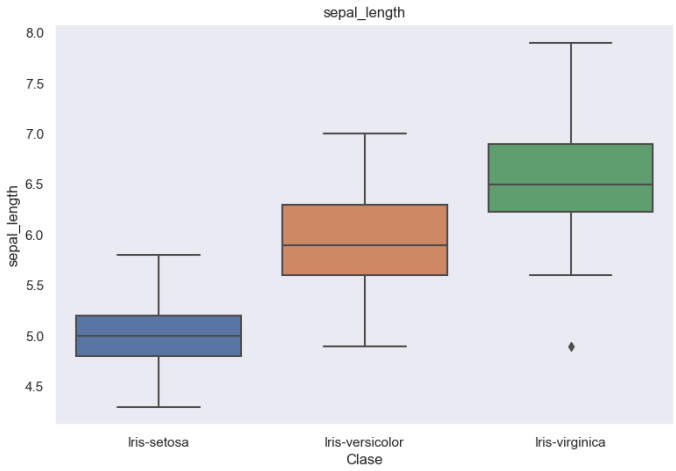

Gráfico 1. Diagrama de cajas. Sepal_length

Analizamos primero los datos originales del dataset con ayuda de los gráficos de caja para cada una de las características que hay.

En el caso de la *longitud del sépalo*, se observa que la clase *setosa* es más pequeña comparada a las demás, que tienen un rango más amplio. Cabe destacar que solo existe un valor atípico negativo en la clase *virginica*, que no afecta de manera considerable.

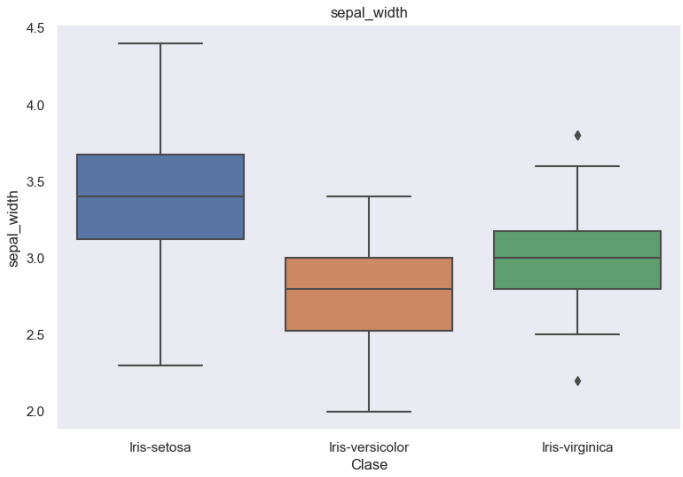

Gráfico 2. Diagrama de cajas. Sepal_width

Para el atributo *ancho de sépalo* el rango de valores se encuentra entre 2 y 4.5. En cada clase, los datos parecen tener una mayor distribución por todo el rango a comparación de la primera grafica. Se pueden observar también valores atípicos en la clase *virginica* pero no afectan la mucho la simetría de su gráfico.

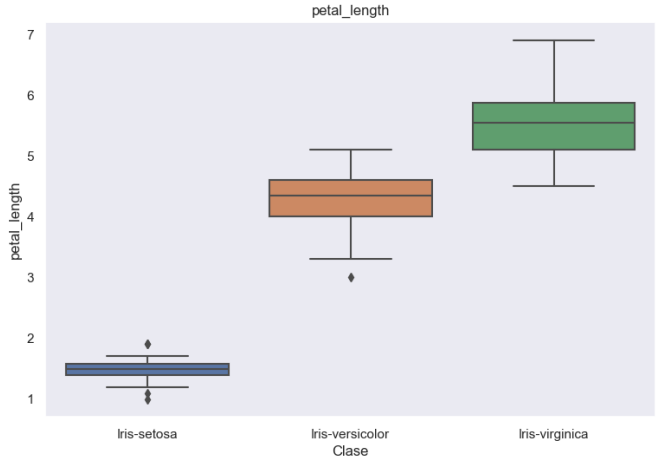

Gráfico 3. Diagrama de cajas. Petal_length

Entrando en el análisis de *petal_length*, se observa una tendencia de los datos a concentrarse en un rango pequeño, es decir a diferencia del *ancho del sépalo*, los rangos de valores contienen mayor cantidad de datos. Además, los bigotes de las cajas no son tan amplios como en el anterior, en este caso tenemos un rango aproximado de 1 a 7, sin embargo, existe una gran diferencia entre las clases *setosa* y *virginica*, siendo la más dominante esta segunda.

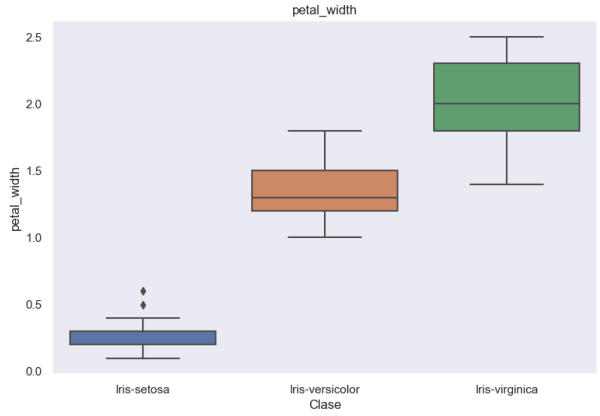

Gráfico 4. Diagrama de cajas. petal_width

Por último, tenemos la *anchura del pétalo*, que al igual que en la gráfica anterior, existe una clara concentración de datos entre clases, además de repetirse la diferencia entre *setosa* y *virginica*. En todos los casos la clase *versicolor* se encuentra en la mitad aproximadamente, en este caso los bigotes en general se aprecian más cortos a diferencia del sépalo

b) Histograma de densidad:

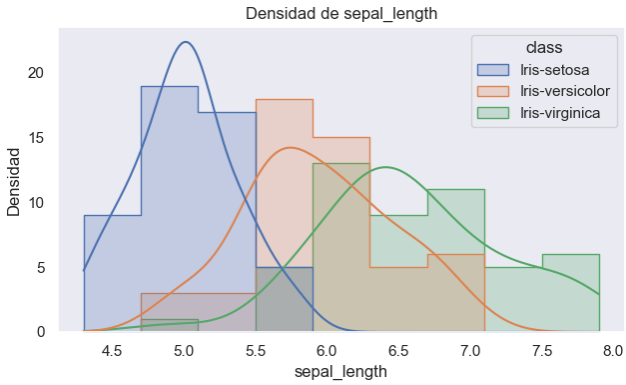

Gráfico 5. Histograma de densidad. Sepal_length

Comenzando del mismo modo con la longitud de sépalo, podemos ver que se tiene una densidad marcada en *setosa*, a pesar de tener los valores más chicos, se tiene una concentración de estos entre 4.5 y 5.5, pero la longitud de este es más limitada, a diferencia de *virginica* que tienen valores más altos, pero se encuentran más dispersos, de forma similar pasa con *versicolor*.

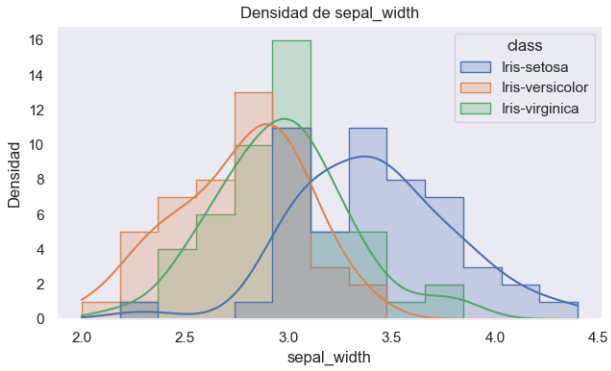

Gráfico 6. Histograma de densidad. Sepal_width

En el análisis de la anchura del sépalo, observamos que en todas las clases abarca una gran cantidad de datos a lo largo del rango, sin embargo, tenemos valores altos en *virginica* a diferencia de las demás que son más similares

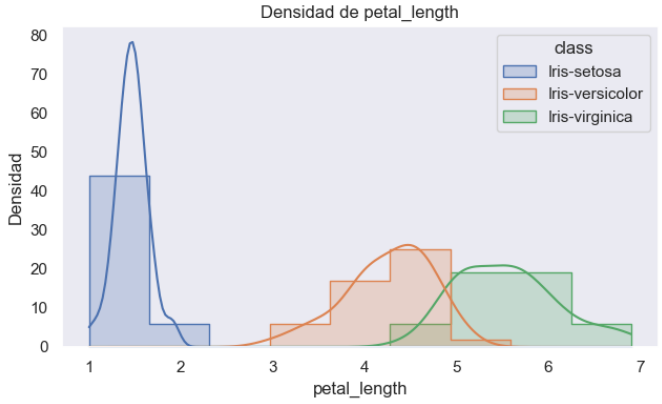

Gráfico 7. Histograma de densidad. petal_length

En la longitud del pétalo, afirmamos lo que se observó con las gráficas de cajas, donde observamos una dispersión de datos mayor, es decir las clases están muy separadas entre sí, podemos ver como *setosa* se separa de *versicolor* y *virginica*, las cuales son las que más datos relacionados tienen, en este caso vemos una mayor concentración de datos en un punto especifico y no se observa que abarquen mucho

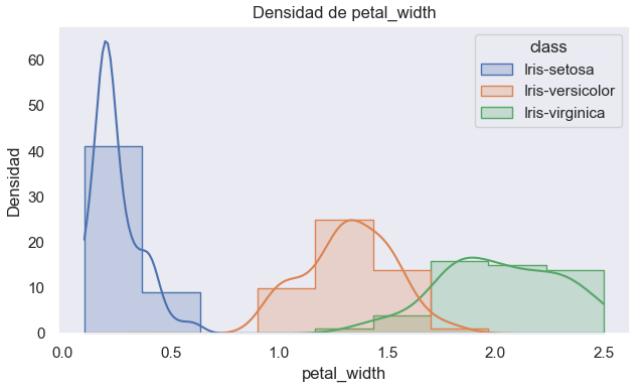

Gráfico 8. Histograma de densidad. petal_width

Para la anchura del pétalo, del mimo modo que el anterior, tenemos mayores concentraciones de datos en puntos específicos y no tanta dispersión de estos, se observa nuevamente una separación entre *setosa* y *versicolor* con *virginica*, sin embargo, *setosa* tiene una gran concentración de datos, a diferencia del otro grupo que están más dispersos a lo largo del rango

c) Estandarización o normalización de los datos:

En el análisis de componentes principales, la normalización puede ser útil si las características en los datos tienen diferentes escalas, ya que PCA es sensible a la varianza en las características. Si una característica tiene una escala mucho mayor que otra, puede dominar la variabilidad total en esa dirección, lo que puede afectar los resultados de PCA. Sin embargo, con el grafico de cajas y los histogramas de densidad, es donde podemos ver como se comportan los datos, en este caso, observamos que las características tienen escalas similares en el sentido de que están en el mismo orden de magnitud, por lo que una normalización no es necesaria.

Sin embargo, en el PCA si se recomienda estandarizar los datos, porque es el proceso con el cual transformaremos los datos para que tengan una media (promedio) de 0 y una desviación estándar de 1. Esto se logra restando la media de los datos y dividiendo por la desviación estándar. En el caso de iris aplicamos una estandarización de los datos para obtener mejores resultados y poder "centrar" los mismos.

2. Aplicación de PCA:

Para la aplicación de PCA usamos la librería de "sklearn.decomposition" importando "PCA", con el objetivo de lograr visualizar los resultados, sin embargo, para obtener control del resultado en cada paso del algoritmo, se realizó otro hecho "a mano", donde el procedimiento a seguir fue:

2.1. Estandarización de los datos: como se explica en el apartado anterior (preprocesamiento de los datos) se opto por hacer una estandarización de los datos, por recomendación del algoritmo, esto nos permitió "centrar" los datos y poder obtener mejores resultados

2.2. Calcular la matriz de covarianza: La matriz de covarianza es una matriz cuadrada que muestra las covarianzas entre cada par de características, en otras palabras, la covarianza es la varianza de 2 características, es decir, cómo varían 2 características entre sí. Es una información muy útil cuando se necesita extraer nuevos patrones o características a partir de características existentes. Por lo tanto, como segundo paso tenemos que calcular la matriz de covarianza de nuestro conjunto de datos.

2.3. Cálculo de los valores y vectores propios

Los *eigenvectores* y *eigenvalores* corresponden a números y vectores asociados a matrices cuadradas. Dada una matriz A de $n \times n$, su eigenvector $\vec{v}$ es una matriz $n \times 1$ tal que:

$$A \cdot \vec{v} = \lambda \cdot \vec{v}$$

donde λ es el eigenvalor, un valor escalar real asociado con el eigenvector.

Siempre que sean compatibles en tamaño, podemos multiplicar dos matrices entre sí. Los *eigenvectores* son un caso especial de esta operación entre una matriz y un vector, siendo los *eigenvectores* de una matriz todos aquellos que, al ser multiplicados por esta matriz, resulten en el mismo vector o en un múltiplo entero de él. Considerando el siguiente ejemplo:

$$\times \begin{pmatrix} 6 \\ 4 \end{pmatrix} = \begin{pmatrix} 24 \\ 16 \end{pmatrix} = 4 \times \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

el vector resultante $\begin{pmatrix} 24 \\ 16 \end{pmatrix}$ es múltiplo entero del vector original: $4 \times \begin{pmatrix} 6 \\ 4 \end{pmatrix}$, lo que es igual a decir que el vector resultante es 4 veces el vector original. Es, por tanto, un eigenvector de la matriz $\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$.

Algunas propiedades de los *eigenvectores* son:

- Solo las matrices cuadradas tienen *eigenvectores*, pero no todas las matrices cuadradas los tienen. Dada una matriz con *eigenvectores*, el número existente de ellos es n.
- Un eigenvector se multiplica por cierto valor antes de multiplicarlo por una matriz, el eigenvector continuará manteniendo su propiedad, ya que solo se cambia su longitud, no su dirección.
- Independientemente del número de dimensiones, todos los *eigenvectores* de una matriz son perpendiculares. Esto significa que podemos expresar los datos respecto a estos *eigenvectores*.
- Es frecuente escalar los *eigenvectores* para que tengan una longitud de 1, de manera que todos tengan la misma longitud.

Cada uno de los componentes principales generados por PCA se corresponde a un eigenvector (dirección).

Por otro lado, los eigenvalores o valores propios son los valores con los que se multiplica el eigenvector y que dan lugar al vector original. En el ejemplo anterior, el eigenvalor asociado al eigenvector se corresponde con el valor 4. Los eigenvalores miden la cantidad de variabilidad retenida por cada componente principal (siendo mayores para la primera componente principal que para el resto), por lo que pueden usarse para determinar el número de componentes principales a retener.

Un eigenvalor > 1 indica que la componente principal explica más varianza de lo que lo hace una de las variables originales, estando los datos estandarizados.

2.4. Elección de componentes principales

A partir del primer resultado, tenemos un valor propio para cada dimensión en los datos y un vector propio correspondiente en los resultados, como se ha indicado anteriormente. Lo que tenemos que hacer es ordenar los valores propios de mayor a menor. Y luego, elegimos algunos de los *eigenvectores* con mayor valor para construir nuestras nuevas características.

Como hemos discutido anteriormente, los valores propios representan el impacto o el poder de un vector, por lo que debemos elegir los vectores cuyo valor propio es mayor en valor. Por ejemplo, si queremos reducir la dimensionalidad de los datos del iris a 2, vamos a elegir los primeros vectores propios porque sus valores propios son los más altos 2 en los resultados. Los *eigenvectores* seleccionados de mayor valor serán nuestros componentes principales para construir un nuevo conjunto de datos reducido y destacado. Y, vamos a llamar a esta matriz como nuevo vector de características.

2.5. Construir un nuevo conjunto de datos reducido

Ahora estamos preparados para construir nuevos datos reducidos a partir de los componentes principales seleccionados en el paso anterior. Para construir el nuevo conjunto de datos, tenemos que multiplicar el transpuesto de la matriz original (R) a la izquierda del transpuesto del nuevo vector de características (componentes principales seleccionados).

La razón por la que multiplicamos la transposición del conjunto de datos original y los componentes principales es para obtener nuevos datos en función de los vectores propios que elijamos. Al realizar este proceso, perdemos información, pero dado que hemos seleccionado los 2 principales vectores propios, las nuevas características que hemos construido a partir de los componentes seleccionados deberían ser suficientes para seguir avanzando. En caso contrario podemos considerar aumentar el número de componentes para construir un nuevo conjunto de datos, para ello podemos hacer un análisis de la varianza explicada por cada clase.

3. Visualización de Datos

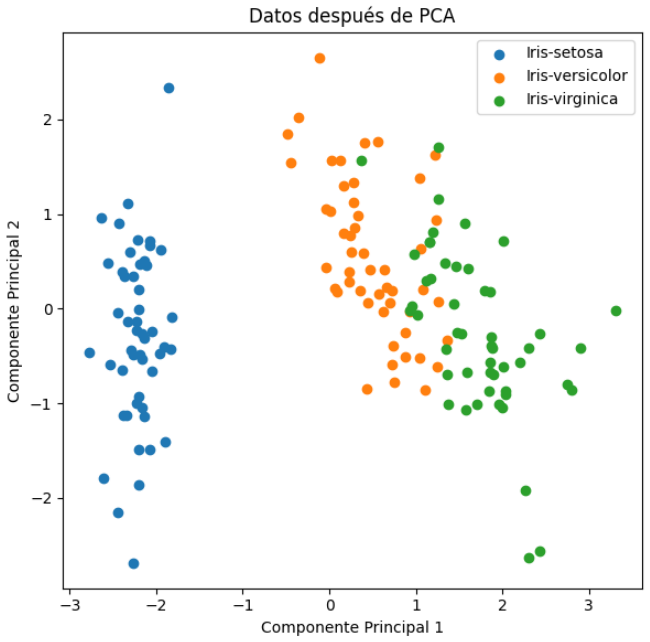

Gráfico 9. Datos tras aplicar PCA

III. RESULTADOS

Cuando ingresamos y graficamos los datos obtenemos:

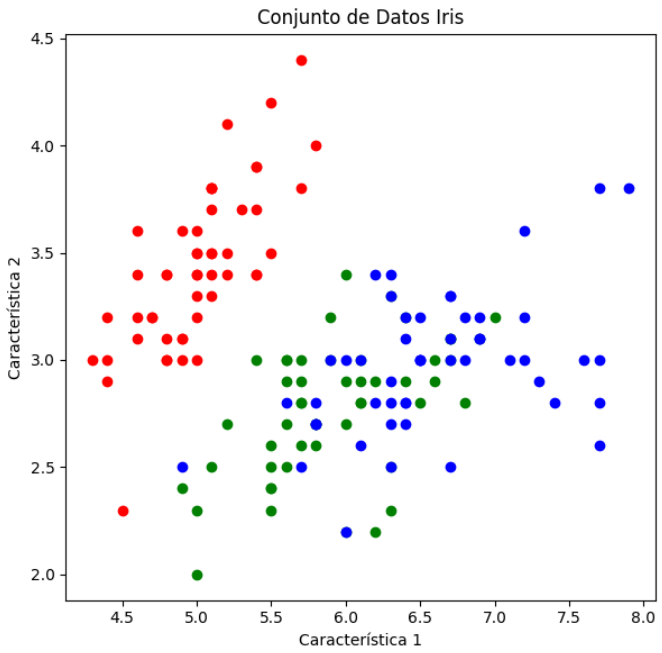

Gráfico 10.

Para el mejor funcionamiento del algoritmo se requiere una estandarización de los datos para centrarlos a pesar de no tener datos atípicos extremos.

	SEPAL LENGHT (CM)	SEPPAL WIDTH(CM)	PETAL LENGHT (CM)	PETAL WIDTH (CM)
0	-0.900681	1.019004	-1.340227	-1.315444
1	-1.143017	-0.131979	-1.340227	-1.315444
2	-1.385353	0.328414	-1.397064	-1.315444
3	-1.506521	0.098217	-1.283389	-1.315444
4	-1.021849	1.249201	-1.340227	-1.315444

Tabla 1. Datos estandarizados

Con los resultados obtenidos, tenemos datos con una media de 0 y una desviación estándar de 1.

El siguiente paso es el cálculo de la covarianza, donde se obtiene:

	SEPAL LENTH (CM)	SEPAL WIDTH (CM)	PETAL LENGTH (CM)	PETAL WIDTH (CM)
SEPAL LENTH (CM)	1.006711	-0.118359	0.877604	0.823431
SEPAL WIDTH (CM)	-0.118359	1.006711	-0.431316	-0.368583
PETAL LENGTH (CM)	0.877604	-0.431316	1.006711	0.969328
PETAL WIDTH (CM)	0.823431	-0.368583	0.969328	1.006711

Tabla 2. Cálculo de covarianza

Recordemos que la covarianza mide cómo dos variables varían juntas. Un valor positivo indica que cuando una variable tiende a aumentar, la otra también tiende a aumentar, y un valor negativo indica que cuando una variable aumenta, la otra tiende a disminuir.

Tenemos las relaciones entre pares de variables, donde si la covarianza entre dos variables es alta en valor absoluto, eso sugiere una relación fuerte entre esas variables. Si es cercana a cero, indica una relación débil. Por ejemplo, en la matriz siguiente, las características *longitud_pétalo* y *anchura_pétalo* de una flor del conjunto de datos tienen una covarianza positiva (0.9669328), lo que significa que estas dos características aumentan o disminuyen juntas.

Con los datos obtenidos de la covarianza, obtendremos los valores y vectores propios:

Valores propios (eigen values)			
2.93808505	0.9201649	0.14774182	0.02085386

Tabla 3. Eigen values.

Vectores propios (eigen vectores)			
0.52106591	-0.37741762	-0.71956635	0.26128628
-0.26934744	-0.92329566	0.24438178	-0.12350962
0.5804131	-0.02449161	0.14212637	-0.80144925
0.56485654	-0.06694199	0.63427274	0.52359713

Tabla 4. Eigen vectors.

Una vez obtenidos los valores y vectores propios, procedemos a escoger el numero de componentes, que podemos decir en pocas palabras a que dimensión queremos transformar los datos, pero en este proceso perderemos información, por lo que podemos analizar este valor calculando la varianza explicada, con ello podremos obtener la varianza acumulada la cual nos indica

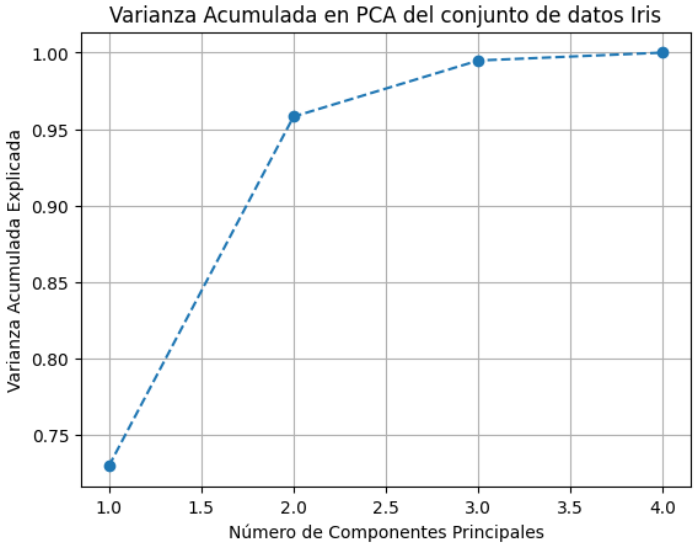

Gráfico 11. Método del codo.

Componente Principal 1: 0.7296
Componente Principal 2: 0.9581
Componente Principal 3: 0.9948
Componente Principal 4: 1.0000

Varianza Explicada por cada Componente Principal:

Componente Principal 1: 0.72962445 (alrededor del 72.96%)
Componente Principal 2: 0.22850762 (alrededor del 22.85%)
Componente Principal 3: 0.03668922 (alrededor del 3.67%)
Componente Principal 4: 0.00517871 (alrededor del 0.52%)

Esto nos indica, cuando escogemos 1 componente principal, tendremos una concentración de datos del 72%, por lo que tendremos una perdida de los datos en un 28%, que no es recomendable. A diferencia de usar 2 componentes principales, que tendremos una concentración de datos del 95%, teniendo una perdida del 5% que es lo más recomendable, y si nos fijamos en la gráfica 11 es el limite del valor que podemos usar para no tener una perdida muy considerable.
En el caso de elegir 3 componentes principales, obtendremos una concentración de datos del 99% pero solamente estaríamos reduciendo la dimensionalidad en 1, y el objetivo principal es reducir la dimensionalidad lo mayormente posible, pero sin perder una gran cantidad de datos, por lo que no es la mejor opción. Escoger 4 componentes principales, seria no reducir la dimensionalidad por lo que no tendría sentida aplicar el algoritmo.

Una vez analizado los diferentes componentes principales, deducimos que 2 es la mejor opción, por lo que buscaremos reducir la dimensionalidad en 2, es decir, vamos a elegir los primeros *eigenvectores* porque sus valores propios son los más altos en los resultados. Los *eigenvectores* seleccionados de mayor valor serán nuestros componentes principales para construir un nuevo conjunto de datos reducido y destacado. Con ello podemos construir nuestros nuevos datos redimensionados:

Resultado de PCA:
[[-2.26470281 -0.4800266]
[-2.08096115 0.67413356]
[-2.36422905 0.34190802]
[-2.29938422 0.59739451]
[-2.38984217 -0.64683538]]

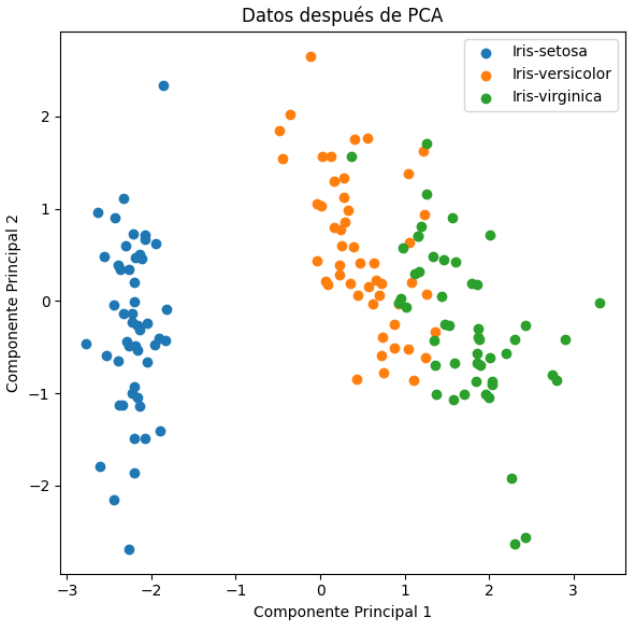

Gráfico 12. Resultados PCA

Si reducimos a 3 componentes principales obtendremos:

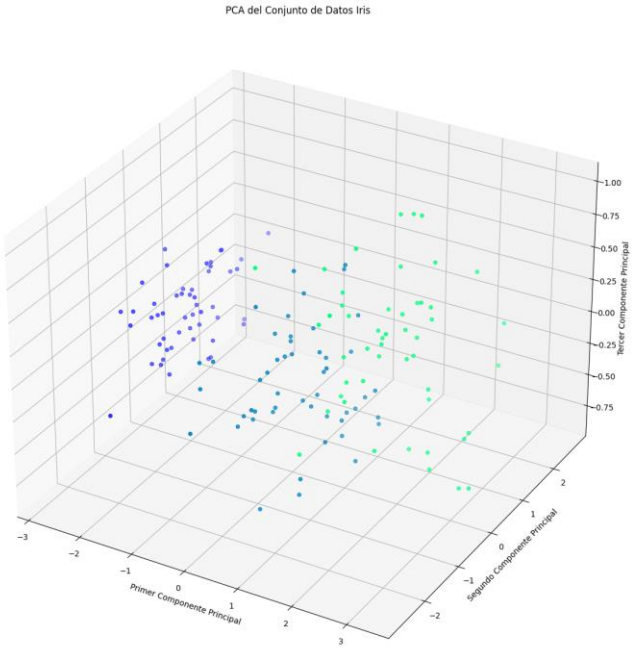

Gráfico 13. Representación de una reducción a 3 componentes.

Y a 1 componente:

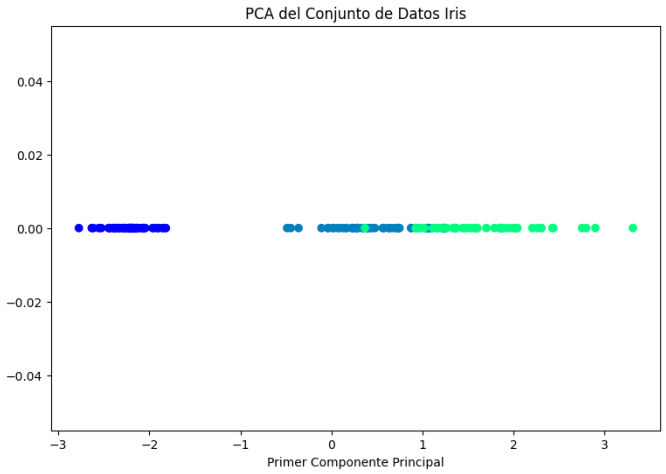

Gráfico 14. Representación de una reducción a 1 componente.

IV. CONCLUSIONES

Aunque la información que proporciona el dataset seleccionado no es muy grande, se hace notar una clara diferencia entre los gráficos originales y los generados después de aplicar el algoritmo.

En la gráfica 12 por ejemplo, se observa una clara separación entre los datos pertenecientes a cada clase de iris. Ya desde las graficas de cajas presentados al inicio se podía deducir una diferencia significativa de estos con respecto a las demás clases, sin embargo, no era el caso para *vericolor* y *viginica* pues los datos de estas últimas dos parecen estar mezcladas como se ve en el grafico 10 que si lo comparamos con la grafica 12, tras aplicar PCA estas clases se agrupan mejor.

Esta técnica tiene el costo de perder información en el proceso, y es importante tener cuidado al plantear las dimensiones a las que se van a reducir los datos. Por eso se recurre al método del codo que para el caso de esta practica nos indico reducir las dimensiones a solo 2.

Le método es aplicable para datasets de múltiples dimensiones y todos se pueden reducir. es posible incluso reducir los valores hasta una dimensión, sin embargo, se pierde mucha información y es difícil describir un comportamiento correcto sobre los datos como se puede observar en la gráfica 14. Caso contrario, si se reducen muy poco las dimensiones, existirá mucha información, pero aun así el comportamiento de los datos tampoco es claro, como por ejemplo en la grafica 13 los datos se redujeron a 3 componentes, pero la información aún se ve mezclada entre sí.

V. BIBLIOGRAFIA

[1] Anónimo (18/11/2021) Análisis de Componentes Principales: Implementación en Python https://blog.damavis.com/analisis-de-componentes-principales-implementacion-en-python/

[2] Hall B. (20/06/2020) An Intuitive Approach to PCA https://medium.com/swlh/an-intuitive-approach-to-pca-fc4d05c14c19
https://www.themachinelearners.com/k-means/

[3] Yasar K. (25/07/2018) PCA: Principal Component Analysis https://medium.com/@kyasar.mail/pca-principal-component-analysis-729068e28ec8

[4] Gil C. (06/2018) análisis de componentes principales (pca) https://rpubs.com/Cristina_Gil/PCA

[5] Código Máquina. *Análisis de Componentes Principales (PCA) para Reducir la Dimensionalidad de Datos usando Python*. (4 de julio de 2022). Accedido el 8 de octubre de 2023. [Video en línea]. Disponible: https://www.youtube.com/watch?v=x-7BHjMA15M