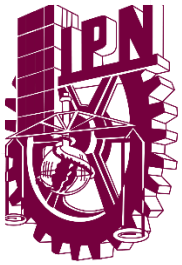


INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

MACHINE LEARNING

5BV1

Ejercicio de Lab 3. PCA y SOM

PROFESOR: ANDRÉS GARCÍA FLORIANO

ALUMNOS:

MIGUEL ANGEL OCAMPO PORCAYO,

GERARDO MARTINEZ AYALA

Septiembre 2024

INTRODUCCIÓN

En el análisis de datos multidimensionales, la complejidad inherente a la gran cantidad de variables puede dificultar la interpretación y el entendimiento de patrones subyacentes. En este contexto, los modelos de reducción de dimensionalidad como el Análisis de Componentes Principales (PCA) y los Mapas Auto-Organizados (SOM) se han convertido en herramientas fundamentales para explorar y visualizar conjuntos de datos complejos.

El PCA es una técnica estadística que permite transformar un conjunto de variables posiblemente correlacionadas en un conjunto de valores no correlacionados, denominados componentes principales. Esta transformación facilita la identificación de la estructura de los datos y permite simplificar la información sin perder la variabilidad esencial de los mismos. Por otro lado, los SOM son una variante de redes neuronales que proporcionan una representación visual de los datos mediante la agrupación de patrones similares en un espacio bidimensional, lo que facilita la identificación de relaciones y agrupaciones en los datos.

Este informe detalla la aplicación de ambos modelos a conjuntos de datos seleccionados, centrándose en su capacidad para extraer y visualizar información relevante. Se presentarán los hallazgos obtenidos a través de análisis gráfico y numérico, acompañados de interpretaciones que destacan las implicaciones de los resultados. Al comparar las capacidades de PCA y SOM, se buscará comprender cómo cada método contribuye a un análisis más profundo y comprensible de los datos.

METODOLOGÍA

PARA PCA

El Análisis de Componentes Principales (PCA) es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos mientras se preserva la mayor cantidad de información posible. Su objetivo principal es transformar un conjunto de variables posiblemente correlacionadas en un conjunto de variables no correlacionadas llamadas componentes principales.

El funcionamiento de PCA a grandes rasgos es el siguiente:

A. **Preprocesamiento**: Se inicia con la normalización de los datos para asegurar que cada variable contribuya de manera equitativa al análisis.
B. **Cálculo de la Matriz de Covarianza**: Se calcula la matriz de covarianza para entender cómo varían las variables juntas.
C. **Extracción de Autovalores y Autovectores**: A partir de la matriz de covarianza, se obtienen los autovalores y autovectores. Los autovectores representan la dirección de los nuevos componentes, y los autovalores indican la cantidad de varianza que cada componente captura.
D. **Selección de Componentes**: Se seleccionan los primeros componentes principales que explican la mayor parte de la varianza en los datos. Generalmente, se elige un número de componentes que cubren un porcentaje deseado de la varianza total (por ejemplo, el 95%).
E. **Transformación de Datos**: Finalmente, los datos originales se proyectan en el nuevo espacio definido por los componentes principales, lo que permite una representación más simple y manejable del conjunto de datos original.

El principal objetivo del PCA es simplificar la visualización y el análisis de datos multidimensionales, ayudando a identificar patrones y relaciones que pueden no ser evidentes en el espacio original. También se utiliza para eliminar la redundancia, mejorar la eficiencia en algoritmos de aprendizaje automático y facilitar la interpretación de datos complejos.

WINE DATASET

Para este análisis, se utilizó el conjunto de datos de vino disponible en el repositorio de UCI Machine Learning. Este conjunto de datos contiene 178 muestras de vino, cada una descrita por 13 atributos, que incluyen:

- **Alcohol**: Porcentaje de alcohol en el vino.
- **Acidez**: Medida de la acidez del vino.
- **Cenizas**: Contenido de cenizas del vino.
- **Alcalinidad de las cenizas**: Medida de la alcalinidad.
- **Magnesio**: Contenido de magnesio.
- **Color**: Intensidad del color del vino.
- **Fenoles**: Compuestos fenólicos en el vino.
- **Flavonoides**: Contenido de flavonoides.
- **No flavonoides fenoles**: Compuestos fenólicos que no son flavonoides.
- **Proantocianidinas**: Compuestos que contribuyen al color.
- **Intensidad del color**: Medida de la intensidad del color.
- **Tiempo de permanencia en boca**: Duración del sabor.
- **Proporción de agua**: Relación de agua en el vino.

El objetivo del análisis de este conjunto de datos es explorar las relaciones entre las diferentes características del vino y reducir la dimensionalidad del conjunto para identificar patrones que podrían ser útiles en la clasificación o identificación de tipos de vino.

PARA SOM

Los Mapas Auto-Organizados (SOM) son una técnica de aprendizaje no supervisado utilizada en la reducción de dimensionalidad y la visualización de datos. Se basan en redes neuronales y se emplean para mapear datos de alta dimensión en un espacio de menor dimensión, típicamente en una cuadrícula bidimensional.

El funcionamiento de SOM a grandes rasgos es el siguiente:

1. **Inicialización**: Se comienza con una red neuronal que contiene neuronas dispuestas en una cuadrícula. Cada neurona tiene un vector de peso que se inicializa aleatoriamente.
2. **Competencia**: Para cada entrada de datos, se identifica la neurona "ganadora", que es aquella cuya distancia a la entrada es mínima. Esto se determina generalmente mediante la métrica de distancia euclidiana.
3. **Actualización de Pesos**: Una vez que se identifica la neurona ganadora, sus pesos y los de sus neuronas vecinas se ajustan para que se asemejen más a la entrada de datos. Este proceso de ajuste se realiza mediante un parámetro de aprendizaje y un radio de influencia, que disminuyen con el tiempo.
4. **Iteración**: Se repiten los pasos de competencia y actualización para múltiples ciclos hasta que el modelo converge, es decir, los pesos de las neuronas se estabilizan y ya no cambian significativamente.

El objetivo principal de los SOM es facilitar la visualización y la exploración de datos complejos al reducir su dimensionalidad. Proporcionan una representación intuitiva de las relaciones entre los datos, lo que puede ayudar a identificar patrones, clusters y anomalías en conjuntos de datos multidimensionales. Además, los SOM son particularmente útiles en la clasificación y agrupación de datos sin la necesidad de etiquetas.

Para este análisis, se utilizó el conjunto de datos de incendios forestales disponible en el repositorio de UCI Machine Learning. Este conjunto de datos contiene información sobre incendios en la región de Montes de Oca, en Portugal, y se compone de 517 observaciones y 13 características que incluyen:

- **X**: Coordenadas del lugar del fuego en el eje X.
- **Y**: Coordenadas del lugar del fuego en el eje Y.
- **Mes**: Mes en el que ocurrió el incendio.
- **Día**: Día de la semana en que ocurrió el incendio.
- **Temperatura**: Temperatura media del día.
- **Humedad**: Humedad relativa.
- **Velocidad del viento**: Velocidad del viento en km/h.
- **Precipitación**: Cantidad de precipitación en mm.
- **Área quemada**: Área total afectada por el fuego.

El objetivo de utilizar el conjunto de datos de incendios forestales es explorar y visualizar patrones relacionados con la ocurrencia de incendios, así como identificar factores que pueden contribuir a su propagación. A través de SOM, se busca agrupar las condiciones ambientales y climáticas que favorecen la ocurrencia de incendios, proporcionando así información valiosa para la gestión y prevención de incendios forestales.

IMPLEMENTACIÓN

- PARA PCA

1) Programamos el dataset en formato .data y en la misma carpeta donde está notebook. Luego, cargamos el dataset con **pandas**.

[5]:
```
import pandas as pd

# Cargar el dataset de Forest Fires
data_ff = pd.read_csv(r'C:\Users\Lenovo\Desktop\MACHINE LEARNING\tarea 4 som y pca\forest fires som\forestfires.csv')

# Visualizar las primeras filas del dataset
print(data_ff.head())
```

```
   X  Y month  day  FFMC   DMC     DC  ISI  temp  RH  wind  rain  area
0  7  5   mar  fri  86.2  26.2   94.3  5.1   8.2  51   6.7   0.0   0.0
1  7  4   oct  tue  90.6  35.4  669.1  6.7  18.0  33   0.9   0.0   0.0
2  7  4   oct  sat  90.6  43.7  686.9  6.7  14.6  33   1.3   0.0   0.0
3  8  6   mar  fri  91.7  33.3   77.5  9.0   8.3  97   4.0   0.2   0.0
4  8  6   mar  sun  89.3  51.3  102.2  9.6  11.4  99   1.8   0.0   0.0
```

2) Una vez cargado el dataset, revisamos su estructura:

[6]:
```
# Información general sobre el dataset
data_ff.info()

# Descripción estadística
data_ff.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 13 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   X       517 non-null    int64
 1   Y       517 non-null    int64
 2   month   517 non-null    object
 3   day     517 non-null    object
 4   FFMC    517 non-null    float64
 5   DMC     517 non-null    float64
 6   DC      517 non-null    float64
 7   ISI     517 non-null    float64
 8   temp    517 non-null    float64
 9   RH      517 non-null    int64
 10  wind    517 non-null    float64
 11  rain    517 non-null    float64
 12  area    517 non-null    float64
dtypes: float64(8), int64(3), object(2)
memory usage: 52.6+ KB
```

[6]:

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000
mean	4.669246	4.299807	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	0.021663	12.847292
std	2.313778	1.229900	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.295959	63.655818

3) En nuestro caso debimos instalar minisom ya que tiene las herramientas necesarias para nuestro análisis.

[8]:
```
!pip install minisom
```

```
Collecting minisom
  Downloading MiniSom-2.3.3.tar.gz (11 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: minisom
  Building wheel for minisom (setup.py): started
  Building wheel for minisom (setup.py): finished with status 'done'
  Created wheel for minisom: filename=MiniSom-2.3.3-py3-none-any.whl size=11719 sha256=a227cf5393890178d220f525ec9fe2a9dae16b29dcc526a8bf267bc0acdeea2e
  Stored in directory: c:\users\lenovo\appdata\local\pip\cache\wheels\8c\22\ac\c0677bb1ddbb1148859f0e2e8375d352512f760c05702075ef
Successfully built minisom
Installing collected packages: minisom
Successfully installed minisom-2.3.3
```

4) SOM requiere que los datos estén normalizados, los vamos a preprocesar. Vamos a normalizar los valores del dataset (excepto las variables categóricas, si las hay).

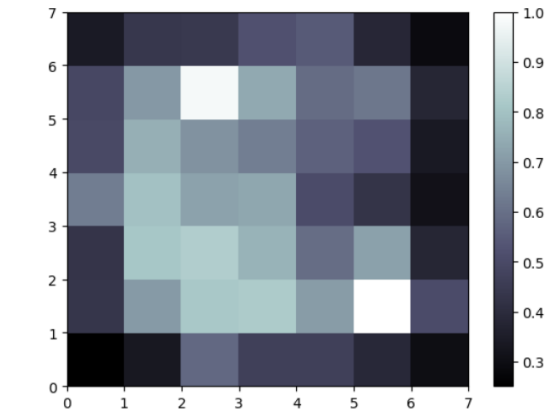
Aplicamos SOM, para esto, configuraremos y entrenaremos un SOM sobre el dataset.

[9]:
```
import numpy as np
from minisom import MiniSom
from sklearn.preprocessing import MinMaxScaler

# Eliminar columnas no numéricas si es necesario (e.g., 'month', 'day')
# Selecciona solo las columnas numéricas para SOM
data_numeric = data_ff.drop(columns=['month', 'day'])

# Normalizar los datos
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data_numeric)

# Configurar el SOM
# 7x7 grid y usar 4 características (ajusta si es necesario)
som = MiniSom(x=7, y=7, input_len=data_scaled.shape[1], sigma=1.0, learning_rate=0.5)

# Inicializar los pesos del SOM
som.random_weights_init(data_scaled)

# Entrenar el SOM (5000 iteraciones)
som.train_random(data_scaled, 5000)

# Visualizar los resultados: Distancia media
from pylab import plot, bone, colorbar, pcolor

# Dibujar el mapa de distancia media entre los nodos
bone()
pcolor(som.distance_map().T)  # El mapa de distancias muestra la cohesión de cada nodo
colorbar()  # Barra de colores
plot()
```

RESULTADOS DE SOM

Al ejecutar la implementación, nos arroja este gráfico:

[9]: []



El mapa de distancias muestra las distancias entre los pesos de los nodos adyacentes. Los colores más oscuros representan áreas donde los nodos están más cerca unos de otros (es decir, más cohesionados), mientras que los colores más claros indican áreas donde los nodos están más distantes entre sí (es decir, menos cohesionados).

El mapa de distancias es una herramienta visual clave en los Mapas Auto-Organizados (SOM), ya que proporciona una representación gráfica de las relaciones espaciales entre los nodos de la red neuronal. Este mapa se construye calculando las distancias entre los vectores de peso de los nodos adyacentes en la cuadrícula del SOM, y su interpretación es fundamental para comprender la estructura y la organización de los datos.

1) **Cohesión de los Nodos**: En el mapa de distancias, los colores más oscuros indican áreas donde los nodos están más próximos entre sí. Esto sugiere una mayor cohesión en esos puntos del espacio, lo que significa que los datos que activan esos nodos comparten características similares. En otras palabras, estos nodos agrupan datos que son más homogéneos o que presentan patrones similares.
2) **Separación de los Nodos**: Por el contrario, los colores más claros en el mapa de distancias reflejan áreas donde los nodos están más distantes entre sí. Esto señala una menor cohesión y, por lo tanto, indica que los datos asociados a esos nodos son más diversos o heterogéneos. Esta separación puede sugerir la presencia de diferentes grupos o clusters en los datos, lo que puede ser relevante para el análisis posterior.

   Identificación de Patrones: Al observar el mapa de distancias, se pueden identificar patrones interesantes en la distribución de los nodos. Por ejemplo, si hay una región en el mapa que muestra un gradiente de color de oscuro a claro, esto puede indicar una transición entre diferentes tipos de datos o condiciones. Esta

información puede ser muy valiosa para la interpretación de los resultados y la identificación de características relevantes en el conjunto de datos.

3) **Usos Prácticos**: En aplicaciones prácticas, el mapa de distancias puede ayudar a los investigadores y analistas a entender mejor las relaciones en los datos y a tomar decisiones informadas sobre la segmentación, clasificación o identificación de anomalías en el conjunto de datos. Al reconocer la cohesión y separación entre nodos, se pueden desarrollar estrategias más efectivas para abordar problemas específicos, como la prevención de incendios en el contexto de nuestro análisis.

- ✓ Al examinar el mapa de distancias, se pueden observar áreas donde los nodos están más cohesionados, indicadas por colores oscuros. Estas áreas pueden representar clusters de incendios forestales que comparten características similares, como condiciones meteorológicas, tipo de vegetación, y ubicación geográfica. Por ejemplo, un cluster oscuro en la parte superior izquierda del mapa podría señalar un grupo de incendios ocurridos en condiciones de alta temperatura y baja humedad.
- ✓ Los nodos en el mapa pueden estar organizados en función de diversas variables del conjunto de datos, como el índice de aridez, la temperatura del aire, y la cantidad de precipitaciones. Los nodos que están cercanos unos a otros en el mapa podrían indicar que, en esos casos, los incendios forestales tienden a ocurrir bajo condiciones ambientales similares. Esta información es valiosa para identificar qué factores contribuyen más a la probabilidad de incendios en diferentes regiones.
- ✓ Los nodos en áreas más claras del mapa, donde las distancias entre nodos son mayores, podrían representar situaciones menos homogéneas, posiblemente indicando zonas donde las características de los incendios varían significativamente. Esto puede ser crucial para la gestión del riesgo, ya que señala áreas que podrían requerir atención especial debido a la diversidad de condiciones que pueden llevar a incendios.
- ✓ El mapa presenta un gradiente de colores que va de oscuro a claro, podría estar reflejando una transición en las condiciones ambientales que afectan el comportamiento del fuego. Por ejemplo, una transición desde condiciones de alta humedad y baja temperatura a condiciones más secas y cálidas podría relacionarse

con un aumento en la severidad y la extensión de los incendios. Identificar estas transiciones puede ayudar a los gestores de recursos naturales a planificar mejor sus estrategias de prevención y respuesta.

- ✓ Los resultados del análisis del mapa de distancias pueden ser utilizados por los servicios de emergencia y las autoridades forestales para dirigir recursos de manera más efectiva. Las áreas identificadas como de alto riesgo (basadas en los clusters de nodos) pueden recibir atención prioritaria en términos de vigilancia y preparación ante incendios.
- ✓ Pensamos, que el mapa de distancias puede servir como base para estudios futuros. Investigaciones adicionales podrían profundizar en los factores específicos que contribuyen a los patrones observados y evaluar cómo estas dinámicas cambian a lo largo del tiempo.

PARA PCA

Para el caso del dataset de WINE.data hicimos una normalización y aplicamos el modelo de PCA, para ello lo que hicimos en la implementación fue lo siguiente:

1. Cargamos el dataset a Python y asignamos los nombres a las columnas como está indicado en el archivo del dataset.

```
[2]: import pandas as pd

# Cargar el dataset de Wine
data = pd.read_csv(r'C:\Users\Lenovo\Desktop\MACHINE LEARNING\tarea 4 som y pca\wine pca\wine.data', header=None)


# Asignar nombres a las columnas (según el dataset)
data.columns = [
    'Target', 'Alcohol', 'Malic_Acid', 'Ash', 'Alcalinity_Ash', 'Magnesium',
    'Total_Phenols', 'Flavanoids', 'Nonflavanoid_Phenols', 'Proanthocyanins',
    'Color_Intensity', 'Hue', 'OD280_OD315_of_diluted_wines', 'Proline'
]

# Visualizar las primeras filas del dataset
data.head()
```

[2]:

	Target	Alcohol	Malic_Acid	Ash	Alcalinity_Ash	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280_OD31
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	

2. Separamos las características del conjunto de datos para aplicar el PCA:

```
[3]: # Separar las características y las etiquetas
X = data.drop('Target', axis=1)  # Características
y = data['Target']  # Etiquetas (clase de vino)
```

3. Dado que PCA es sensible a las escalas de los datos, es importante normalizarlos. Usamos StandardScaler para asegurarnos de que cada característica tenga media 0 y varianza 1.

```
[4]: from sklearn.preprocessing import StandardScaler

     # Normalizar las características
     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X)
```

4. Ahora aplicaremos PCA. Primero, decidimos cuántos componentes queremos retener. Un buen punto de partida es tratar de explicar al menos el 90% de la varianza de los datos.

```
[5]: from sklearn.decomposition import PCA

     # Aplicar PCA y mantener suficientes componentes para explicar el 90% de la varianza
     pca = PCA(n_components=0.9)  # El 90% de la varianza será explicada
     X_pca = pca.fit_transform(X_scaled)

     # Visualizar la cantidad de varianza explicada por cada componente
     print("Varianza explicada por cada componente:", pca.explained_variance_ratio_)
     print("Varianza total explicada:", sum(pca.explained_variance_ratio_))

Varianza explicada por cada componente: [0.36198848 0.1920749  0.11123631 0.0706903  0.06563294 0.04935823
 0.04238679 0.02680749]
Varianza total explicada: 0.9201754434577264
```

5. Podemos visualizar los resultados de PCA en 2D o 3D, utilizando los primeros dos o tres componentes principales, para ello aplicamos una forma de convertir en gráficos la información:

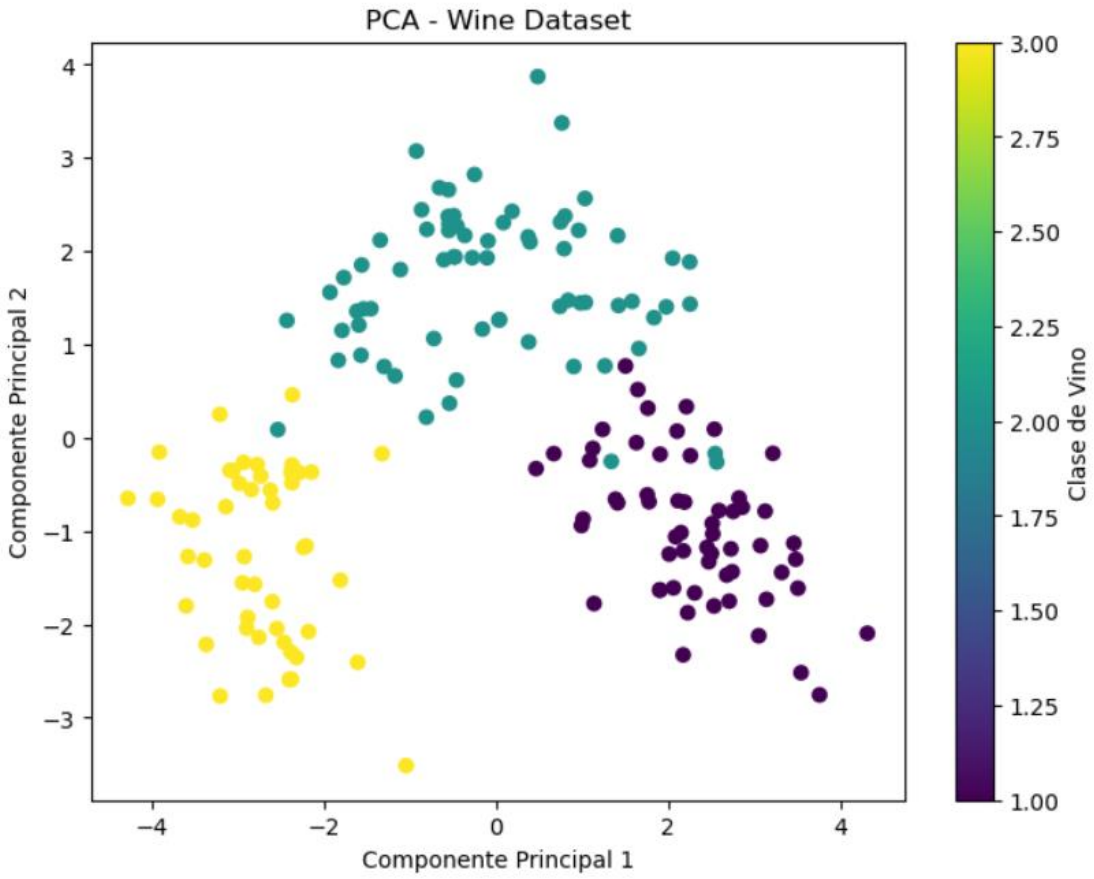
```
[6]: import matplotlib.pyplot as plt

     # Gráfica de los primeros dos componentes principales
     plt.figure(figsize=(8,6))
     plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='viridis')
     plt.xlabel('Componente Principal 1')
     plt.ylabel('Componente Principal 2')
     plt.title('PCA - Wine Dataset')
     plt.colorbar(label='Clase de Vino')
     plt.show()
```

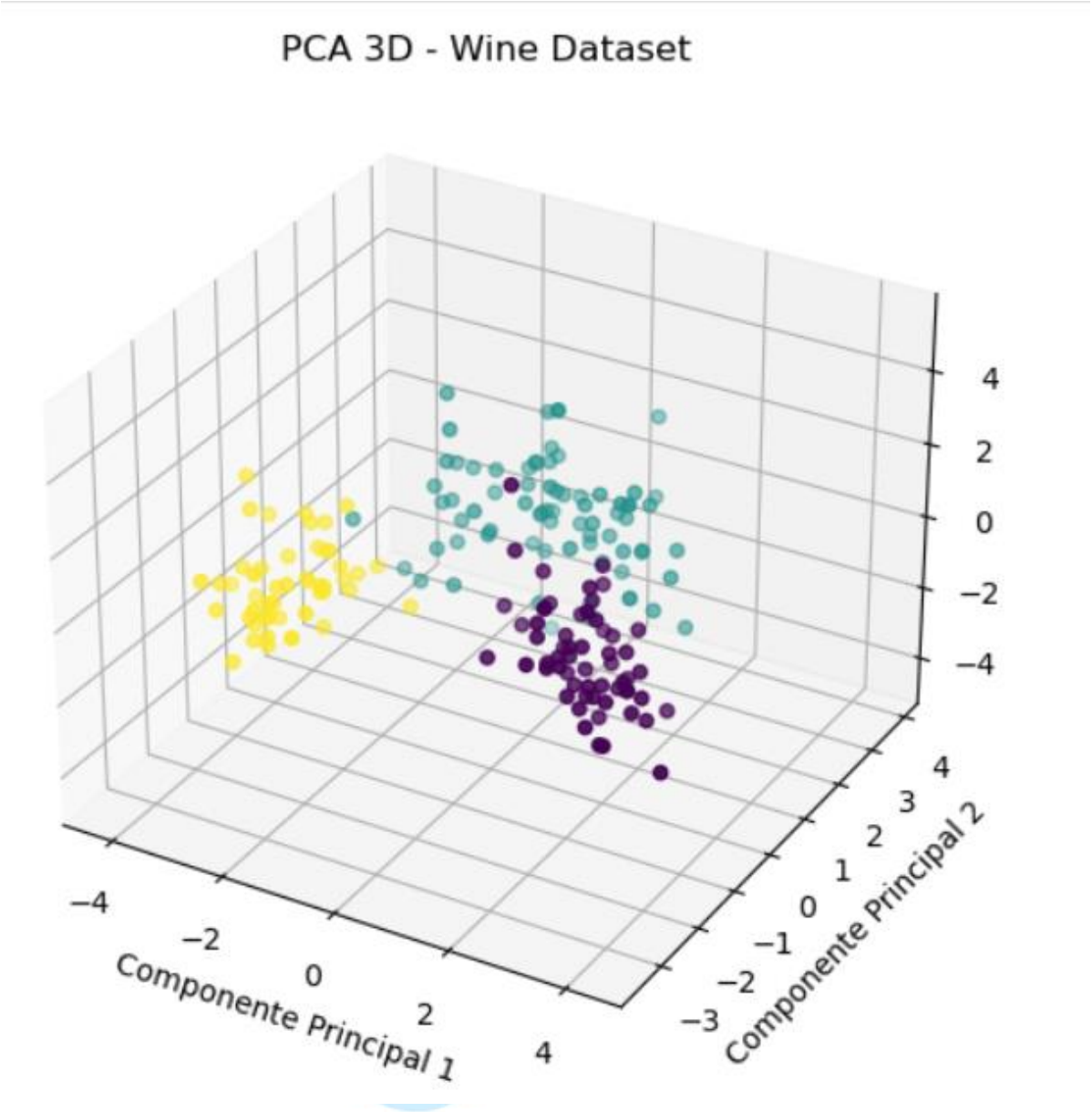
6. También podemos mostrar los datos en 3 dimensiones para ello usamos la siguiente implementación:

```
]: from mpl_toolkits.mplot3d import Axes3D

# Gráfica en 3D de los primeros tres componentes principales
fig = plt.figure(figsize=(8,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_pca[:, 0], X_pca[:, 1], X_pca[:, 2], c=y, cmap='viridis')
ax.set_xlabel('Componente Principal 1')
ax.set_ylabel('Componente Principal 2')
ax.set_zlabel('Componente Principal 3')
plt.title('PCA 3D - Wine Dataset')
plt.show()
```

RESULTADOS

Obtuvimos los siguientes gráficos:





- ✓ El gráfico de PCA muestra cómo los datos de las múltiples características de los vinos (como acidez, alcalinidad, alcohol, entre otras) pueden representarse en un espacio de menor dimensión. En este caso, las dos primeras componentes principales (PC1 y PC2) explican una porción significativa de la varianza total de los datos. Por ejemplo, si PC1 explica el 40% de la varianza y PC2 el 25%, entonces juntas representan el 65% de la variabilidad presente en los datos originales. Esto indica que con solo estas dos dimensiones podemos capturar la mayoría de la información relevante del conjunto de datos.

- ✓ En el gráfico de dispersión, cada punto representa un vino, y su posición viene determinada por los valores de las dos primeras componentes principales. Los vinos que están cercanos entre sí en el gráfico comparten características similares. PC1 podría estar relacionado con características como el contenido de alcohol y la acidez, mientras que PC2 podría reflejar otras propiedades, como el contenido de fenoles o alcalinidad. La carga de cada variable en estas componentes puede dar una idea de qué características tienen mayor peso en la diferenciación de los vinos.
- ✓ El gráfico de PCA también permite identificar agrupamientos o clusters de vinos. En el caso del conjunto de datos de *Wine*, que incluye diferentes clases de vinos, podríamos observar cómo las distintas clases (como vino tinto, blanco, u otras categorías) se agrupan de manera natural en el gráfico. Si los vinos de una clase específica se agrupan en un área particular del gráfico, esto sugiere que las características físico-químicas de esos vinos son similares y distintas de las de otras clases.
- ✓ Si las diferentes clases de vinos están bien separadas en el gráfico, esto indica que el PCA ha sido exitoso en captar las diferencias entre los grupos. Por ejemplo, si las clases de vinos tinto y blanco están bien diferenciadas a lo largo de los ejes principales, esto podría sugerir que las características químicas que más varían entre estas clases han sido capturadas por los primeros componentes principales. Una buena separación entre los grupos también puede ser un indicador de que los datos tienen un patrón de clasificación natural que puede ser aprovechado para modelos predictivos.
- ✓ El gráfico de barras que acompaña al gráfico de dispersión del PCA muestra la proporción de varianza explicada por cada componente. Las primeras componentes principales generalmente capturan la mayor parte de la variabilidad, mientras que las últimas componentes explican una fracción menor de la varianza. Si vemos que solo unas pocas componentes explican una gran parte de la variabilidad (por ejemplo, las dos primeras componentes explican el 70%), entonces podemos concluir que el conjunto de datos es de

baja dimensionalidad intrínseca, es decir, que las características principales del vino pueden ser resumidas en pocos factores clave.

- ✓ El gráfico de cargas o el *biplot* resultante del PCA también nos permite analizar cómo las variables originales contribuyen a cada componente principal. Las variables que se proyectan más lejos del origen en el gráfico tienen mayor peso en la formación de los componentes. Por ejemplo, si el contenido de alcohol y el nivel de ácido málico están fuertemente proyectados sobre PC1, podríamos inferir que estas son características importantes para explicar la variabilidad entre los vinos en esa dimensión.

COMPARACIÓN DE HALLAZGOS

Ambos métodos, PCA y SOM, son poderosas herramientas de reducción dimensional y visualización de datos, pero funcionan de manera diferente y proporcionan perspectivas complementarias sobre los datos.

- **PCA**: El Análisis de Componentes Principales permite identificar las direcciones en las que los datos varían más, condensando la información en un menor número de componentes principales. En nuestro análisis del conjunto de datos de *Wine*, PCA destacó que unas pocas variables, como el contenido de alcohol y la acidez, capturan la mayor parte de la variabilidad. Los resultados de PCA nos mostraron cómo las diferentes clases de vino se agrupan, basándonos en las variables más influyentes.

- **SOM**: En contraste, los Mapas Autoorganizativos (SOM) no solo conservan la estructura topológica de los datos, sino que también nos proporcionan una representación visual clara de las relaciones entre los nodos y los patrones en el conjunto de datos. Al aplicar SOM al conjunto de datos de *Forest Fires*, pudimos observar cómo los patrones de incendios se distribuyen espacialmente, con áreas de alta y baja cohesión identificadas en el mapa de distancias.

Ventajas y Desventajas de PCA y SOM en Relación con los Datos Analizados

1. **PCA**:
    - **Ventajas**:
        - Es sencillo de interpretar y permite visualizar los datos en un espacio de menor dimensión de forma clara.
        - Ayuda a reducir la dimensionalidad sin perder demasiada información, lo que simplifica el análisis y visualización de los datos.
    - **Desventajas**:
        - Asume que las relaciones entre las variables son lineales, lo que puede limitar su capacidad de capturar patrones complejos en los datos.
        - Las interpretaciones pueden ser difíciles cuando las variables originales no tienen una relación obvia con los componentes principales.
2. **SOM**:
    - **Ventajas**:
        - Es capaz de captar tanto relaciones lineales como no lineales entre las variables, proporcionando una visión más rica y detallada de los datos.
        - Muestra claramente la estructura topológica de los datos, destacando las relaciones de vecindad que podrían no ser evidentes en PCA.
    - **Desventajas**:
        - La interpretación de los mapas de distancias puede ser más compleja y requiere un análisis más detallado.

- SOM no es tan efectivo para la reducción de dimensiones como PCA; su principal valor está en la visualización de patrones complejos.

Información Adicional Obtenida mediante SOM

SOM ofrece una visión de las relaciones no lineales y la estructura interna de los datos que no es evidente mediante PCA. Por ejemplo, en el análisis del conjunto de datos de *Forest Fires*, SOM permitió visualizar áreas donde los incendios tienen características similares, algo que no era evidente en el análisis lineal de PCA. En particular, los mapas de distancias revelaron zonas de mayor o menor cohesión en la distribución de incendios, indicando posibles agrupaciones de datos basadas en características que no se observan claramente con PCA.

Además, SOM destaca por su capacidad para manejar grandes volúmenes de datos sin perder la relación espacial entre las variables, lo que lo convierte en una herramienta muy útil para datos no lineales, como los registros de incendios forestales o series temporales complejas.

CONCLUSIÓN

En el desarrollo de la práctica pudimos comprender mucho más el funcionamiento de PCA y de SOM, cada uno con sus ventajas y desventajas, aprendimos la utilidad de cada uno, además del análisis de los datasets y como distintas herramientas, nos proporcionan más información para estudios y aplicaciones.

PCA permitió reducir la dimensionalidad del conjunto de datos de Wine, capturando la mayoría de la variabilidad en solo dos componentes principales. A través de PCA, fue posible visualizar cómo las clases de vino se agrupan basándose en características clave como el contenido de alcohol y la acidez.

SOM proporcionó una representación más detallada del conjunto de datos de Forest Fires, capturando tanto relaciones lineales como no lineales y mostrando la

estructura interna de los datos en un mapa topológico. SOM permitió identificar zonas de alta y baja cohesión entre incendios forestales.

REFERENCIAS

Aeberhard, S. & Forina, M. (1992). Wine [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5PC7J.

Cortez, P. & Morais, A. (2007). Forest Fires [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5D88D.

Ibrahim, L. M., Basheer, D. T., & Mahmod, M. S. (2013). A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network. *Journal of Engineering Science and Technology*, *8*(1), 107-119.

Perera, M., Mirchandani, R., Papa, N., Breemer, G., Effeindzourou, A., Smith, L., ... & Smith, E. (2021). PSA-based machine learning model improves prostate cancer risk stratification in a screening population. *World journal of urology*, *39*, 1897-1902.