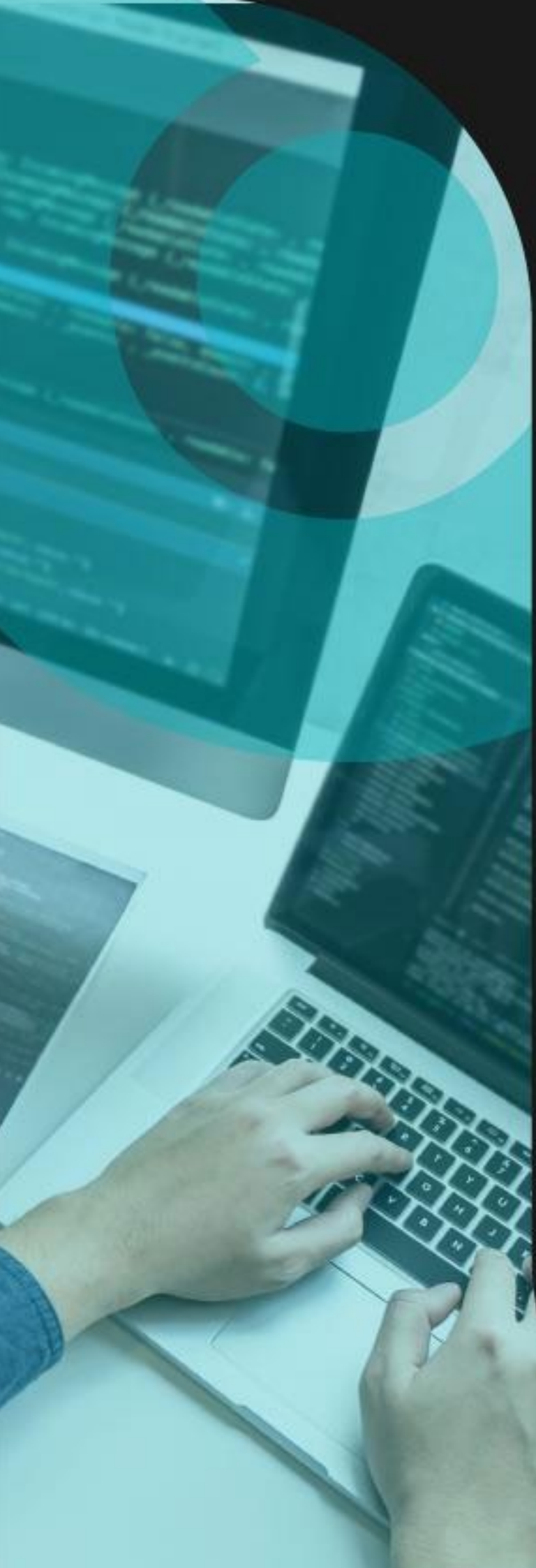


AWAKELAB

BASECAMP

Ciencia de Datos

Módulo: Aprendizaje de Máquina No Supervisado

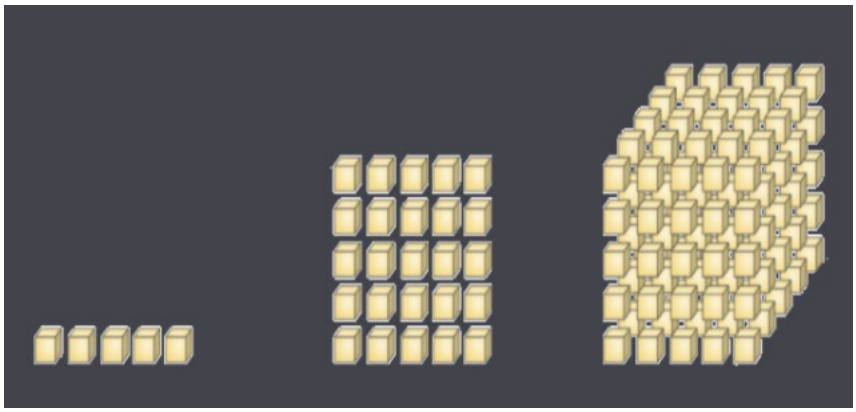
Aprendizaje Esperado

---

4. Elabora un modelo predictivo utilizando técnicas de reducción dimensional para resolver un problema de aprendizaje de máquina.

Reducción de Dimensionalidad

En los últimos años, la mayoría de empresas y organizaciones han almacenado grandes cantidades de datos de forma sistemática, sin ningún uso potencial claro. Es por esto que uno de los desafíos en la Ciencia de Datos es determinar cuáles y cuántos son los features a utilizar en nuestro algoritmo (ya sea una regresión logística o un árbol de decisión, clustering, etcétera).



El problema de tener una gran cantidad de variables es:

- El tiempo de procesamiento involucrado puede ser muy alto
- Probable multicolinealidad de los features, lo que puede producir gran inestabilidad e ineficiencia en determinados algoritmos (regresiones, redes neuronales, etcétera).

- Es muy posible que no todas aportan realmente información al modelo (variables ruido).
- Empeora el rendimiento del modelo, si el número de observaciones es pequeño, el incremento en la variabilidad de las predicciones puede no compensar la disminución del sesgo de predicción (conocido como la maldición de la dimensionalidad).

Así, la reducción de dimensionalidad trata de que al tener variables, se puede reducir la dimensión a z (con $z < p$) variables que puedan entregar (idealmente) tanta información como el total de variables.

¿Qué es la reducción de dimensionalidad?

La reducción de dimensionalidad se refiere a técnicas para reducir el número de variables de entrada en los datos de entrenamiento.

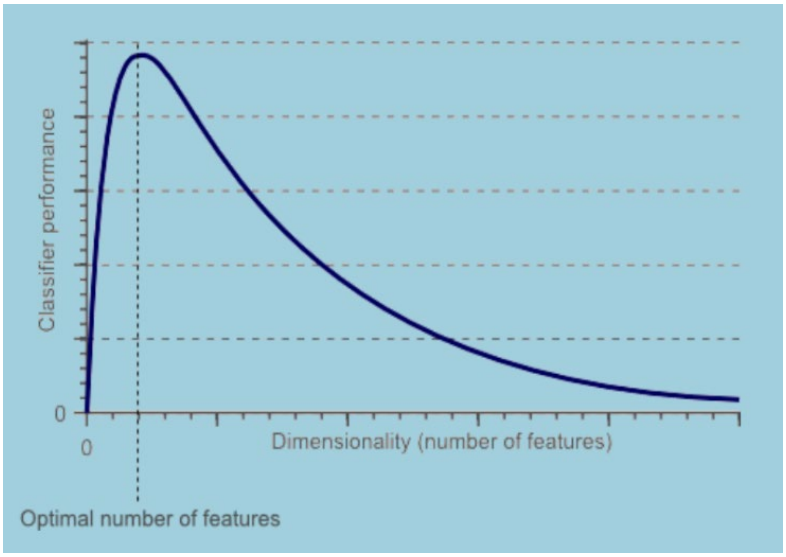
"Cuando se trata de datos de alta dimensión, a menudo es útil reducir la dimensionalidad proyectando los datos a un subespacio dimensional más bajo que captura la "esencia" de los datos. Esto se llama reducción de dimensionalidad" (Página 11, Machine Learning: A Probabilistic Perspective, 2012).

La alta dimensionalidad puede significar cientos, miles o incluso millones de variables de entrada.

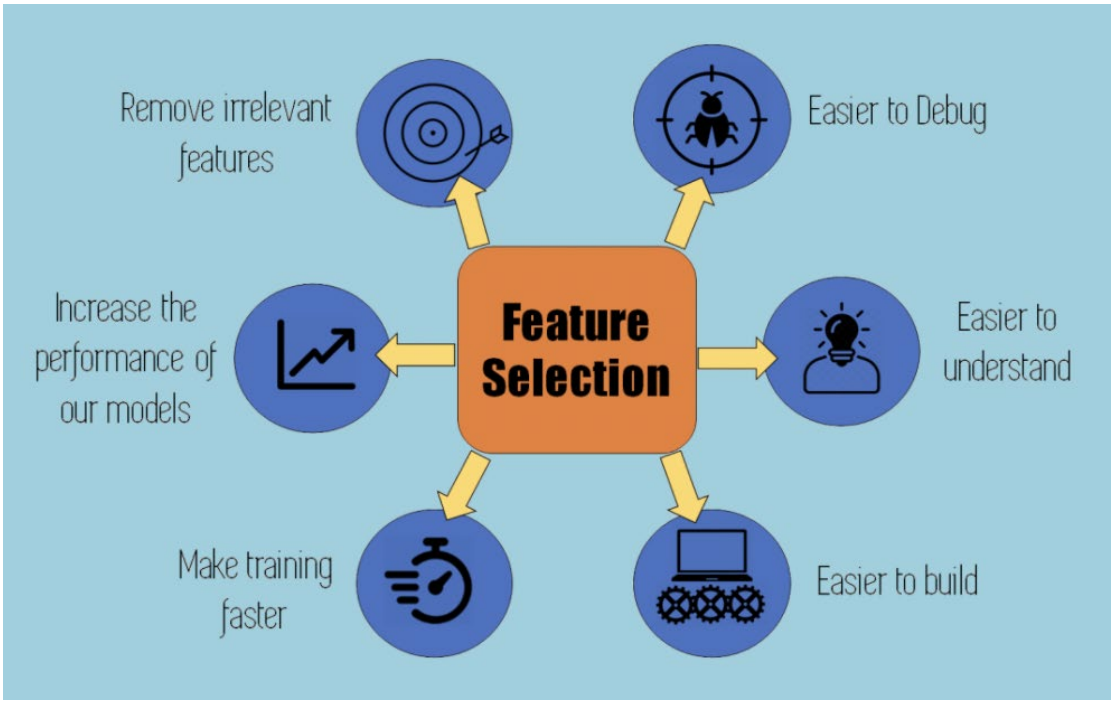
Menos dimensiones de entrada a menudo significan menos parámetros o una estructura más simple en el modelo de aprendizaje automático, lo que se conoce como grados de libertad. Es probable que un modelo con demasiados grados de libertad se ajuste demasiado al conjunto de datos de entrenamiento y, por lo tanto, es posible que no funcione bien con los datos nuevos.

Es deseable tener modelos simples que generalicen bien y, a su vez, datos de entrada con pocas variables de entrada. Esto es particularmente cierto para los modelos lineales donde el número de entradas y los grados de libertad del modelo suelen estar estrechamente relacionados.

A medida que aumenta el número de características o dimensiones, la cantidad de datos que necesitamos para generalizar con precisión aumenta exponencialmente.



Selección de Features para reducir dimensionalidad



Existen dos tipos de métodos para reducir dimensionalidad basado en búsqueda o selección de características:

1. Eliminación de características

Se eliminan algunas variables completamente si son redundantes con alguna otra variable o si no están proporcionando ninguna información nueva sobre el conjunto de datos. Por ejemplo, los métodos backward que eliminan variables si no aportan al modelo, o también, utilizar Random Forest para calcular la importancia de los features y eliminar aquéllos que no poseen mayor importancia.

2. Extracción de variables

Se crean nuevas variables a partir de las antiguas, de modo tal que representan la misma cantidad de información pero con ventajas matemáticas y computacionales. Por ejemplo, Análisis de Componentes Principales.

Métodos de Reducción de Dimensionalidad Lineal

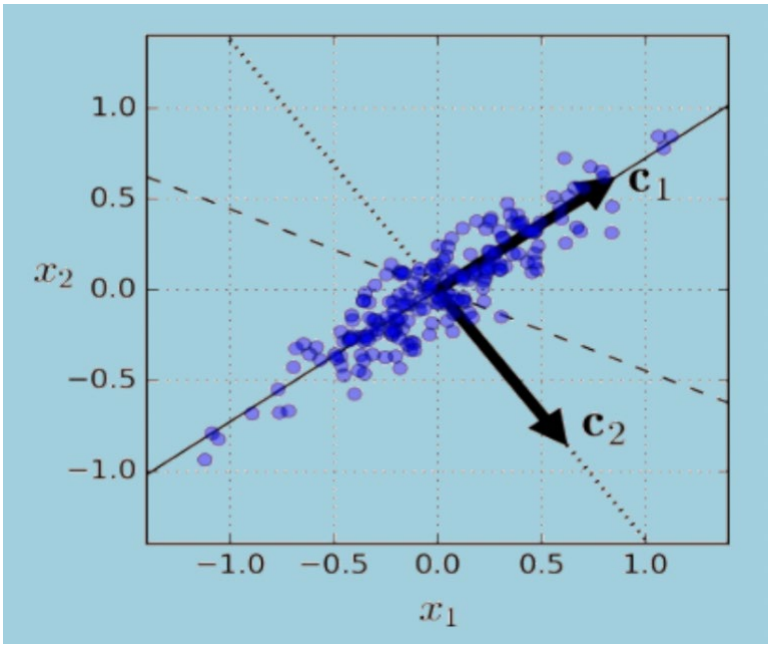
Los métodos de reducción de dimensionalidad más comunes y conocidos son los que aplican transformaciones lineales, como:

- **PCA (Análisis de componentes principales):** popularmente utilizado para la reducción de la dimensionalidad en datos continuos, PCA rota y proyecta datos en la dirección de la varianza creciente. Las características con la varianza máxima son los componentes principales.
- **Análisis factorial:** una técnica que se utiliza para reducir un gran número de variables en un menor número de factores. Los valores de los datos observados se expresan como funciones de un número de causas posibles para encontrar cuáles son las más importantes. Se supone que las observaciones son causadas por una transformación lineal de factores latentes dimensionales más bajos y ruido gaussiano agregado.
- **LDA (Análisis Discriminante Lineal):** proyecta datos de manera que se maximiza la separabilidad de clases. Los ejemplos de la misma clase se juntan muy de cerca por la proyección. Los ejemplos de diferentes clases se colocan muy separados por la proyección.

Análisis de Componentes Principales (PCA)

Es una técnica de Machine Learning para la reducción de la dimensionalidad del conjunto de datos, manteniendo la máxima cantidad de información posible y eliminando las redundancias. Determina la relación entre variables a través de vectores (componentes principales) incorrelacionados, que son combinaciones lineales de las variables originales, es decir, identifica el hiperplano que se encuentra más cerca de los datos y luego proyecta los datos en él.

El objetivo es explicar la mayor parte de la variabilidad de los datos con un número menor de variables que el conjunto de datos original.



¿En qué consiste PCA?

El primer componente principal de un conjunto de datos $X_1, X_2, \ldots, X_p$ Es la combinación lineal de las variables originales que representa la mayor variabilidad.

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

Donde ϕ_1 es el vector de pesos normalizados de este componente, esto es: $\sum_{j=1}^{p} \quad \phi_{j1}^2 = 1$.

Los demás componentes se describen por:

$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \cdots + \phi_{p2}X_p$$

$$\vdots$$

$$Z_p = \phi_{1p}X_1 + \phi_{2p}X_2 + \cdots + \phi_{pp}X_p$$

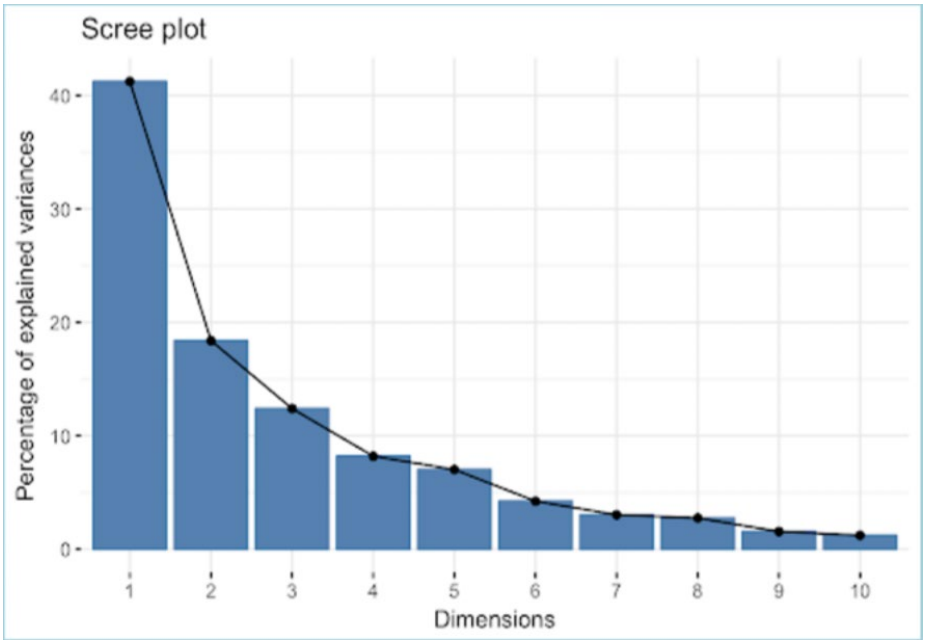
Los componentes son independientes entre sí.

Varianza Explicada en cada PCA

Si tenemos un set de datos con p features, podemos generar p componentes principales. Sin embargo, podemos utilizar k (con $k < p$) componentes principales basadas en el porcentaje de variabilidad explicada (PVE):

$$PVE_m = \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} \phi_{jm} x_{ij}^2}{\sum_{i=1}^{n} \sum_{j=1}^{p} x_{ij}^2}$$

Para que el cálculo de la variabilidad explicada no esté sesgado por unidades de medida de las variables, se debe tener aproximadamente la misma escala y varianza en todas las variables previo a obtener las componentes principales.



¿Cuándo debería usar PCA?

- ¿Desea reducir la cantidad de variables, pero no puede identificar las variables para eliminarlas por completo?
- ¿Quiere asegurarse de que sus variables sean independientes entre sí?
- ¿Se siente cómodo haciendo que sus variables independientes sean menos interpretables?

Si respondió "sí" a las tres preguntas, entonces PCA es un buen método para usar. Si respondió "no" a la pregunta 3, no debe usar PCA.

Algunos casos de uso más particulares para PCA incluyen:

- Cuando las características latentes están impulsando los patrones en los datos.
- Para reducción de dimensionalidad.
- Para visualizar datos de alta dimensión.
- Para reducir el ruido.
- Como un paso de preprocesamiento para mejorar el rendimiento de otros algoritmos.

Deficiencias de PCA

Si el número de variables es grande, se vuelve difícil interpretar los componentes principales. PCA es más adecuado cuando las variables tienen una relación lineal entre ellas. Además, PCA es susceptible a grandes valores atípicos.

Conclusión

PCA es un método antiguo y ha sido bien investigado. Hay muchas extensiones de PCA básico que abordan sus deficiencias, como PCA robusto, PCA de kernel, PCA incremental.

Implementación en Py

```
import numpy as np

from sklearn.decomposition import PCA

X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1],
[2, 1], [3, 2]])

pca = PCA(n_components=2)

pca.fit(X)

PCA(n_components=2)

print(pca.explained_variance_ratio_)

print(pca.singular_values_)

#[0.99244289 0.00755711]

#[6.30061232 0.54980396]


pca = PCA(n_components=2, svd_solver='full')

pca.fit(X)

PCA(n_components=2, svd_solver='full')

print(pca.explained_variance_ratio_)

print(pca.singular_values_)

#[0.99244289 0.00755711]

#[6.30061232 0.54980396]
```

```
pca = PCA(n_components=1, svd_solver='arpack')

pca.fit(X)

PCA(n_components=1, svd_solver='arpack')

print(pca.explained_variance_ratio_)

print(pca.singular_values_)

#[0.99244289]

#[6.30061232]
```

Referencias

[1] Técnicas de reducción de dimesionalidad
https://topbigdata.es/6-algoritmos-de-reduccion-de-la-dimensionalidad-con-python/#:~:text=La%20reducci%C3%B3n%20de%20la%20dimensionalidad%20es%20una%20t%C3%A9cnica%20de%20preparaci%C3%B3n,de%20entrenar%20un%20modelo%20predictivo.

[2] Reducción de dimensionalidad
https://pharos.sh/reduccion-dimensional-en-python-con-scikit-learn/

[3] PCA
https://www.cienciadedatos.net/documentos/35_principal_component_analysis

[4] LDA
https://economipedia.com/definiciones/analisis-discriminante.html

Material Complementario

[1] Reducción de dimensionalidad
https://www.youtube.com/watch?v=jPmV3j1dAv4
[2] Análisis de Discriminante – full video
https://www.youtube.com/watch?v=bdU4XBYkvmA
[3] Componentes principales – full video
https://www.youtube.com/watch?v=oAsZfT-paG0