



PUCP

Curso: Python aplicado a Data Science

Sesión 5: Reducción de Dimensionalidad



Contenido del curso

Sesión 1: Introducción a Ciencia de Datos y Python

Sesión 2: Preprocesamiento de datos con **Numpy y Pandas**

Sesión 3: **Visualización** de datos con Seaborn, Matplotlib y Plotly

Sesión 4: Análisis de Asociaciones: Market Basket Analysis

Sesión 5: Reducción de Dimensionalidad (PCA)

Sesión 6: Análisis de Agrupamientos: **K-Means, DBSCAN**

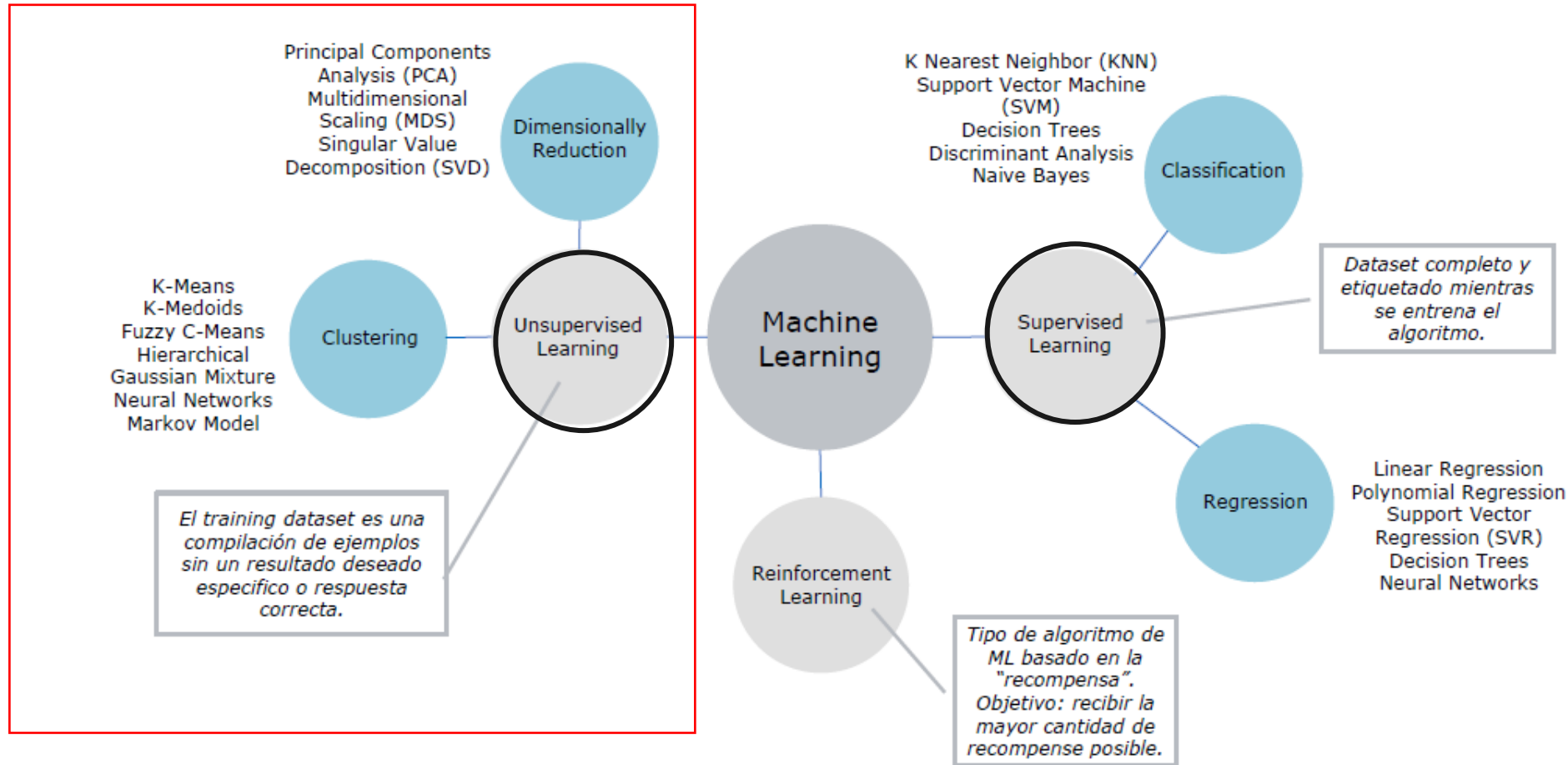
Sesión 7: Analítica predictiva: Modelos de **regresión**

Sesión 8: Analítica predictiva: Modelos de **clasificación**

Agenda

- Tareas del aprendizaje No Supervisado
- Reducción de Dimensionalidad
- Análisis de Componentes Principales
- Conclusiones PCA

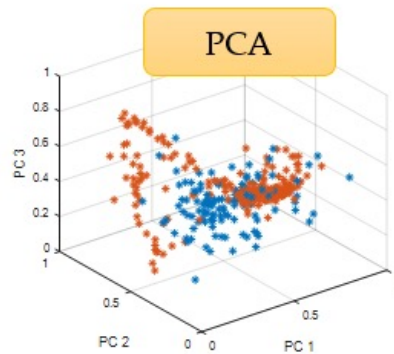
¿Qué hemos visto hasta ahora?



Principales tareas del Aprendizaje No Supervisado

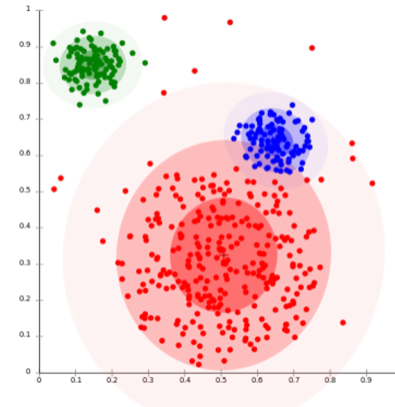


Análisis de Asociaciones



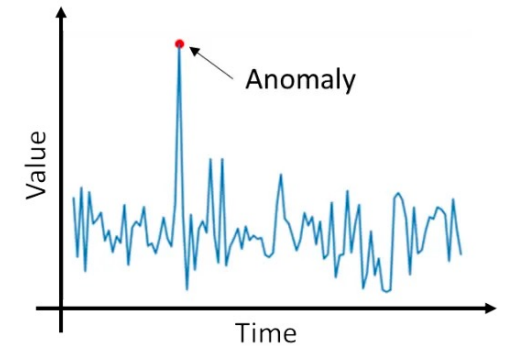
Reducción Dim.

Ej. Compresión de datos.
Visualización y entendimiento



Clustering

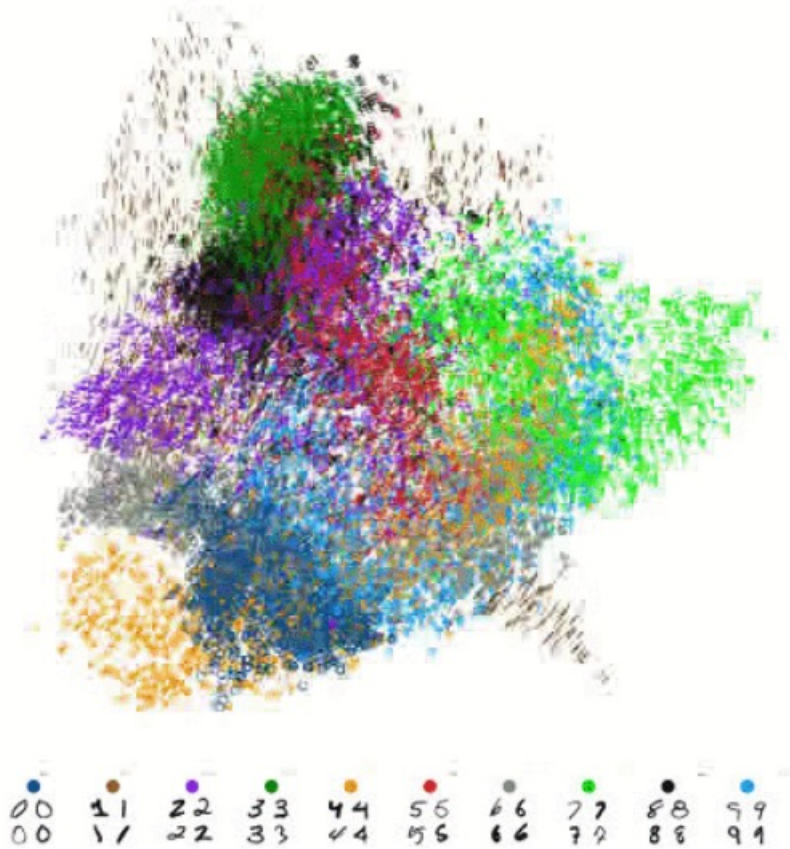
El objetivo es agrupar instancias en clusters. Clientes con preferencias similares, agrupación de textos.



Detección de Anomalías

Ej. productos defectuosos,
patrones de uso de tarjetas.

¿Por qué reducir dimensiones?

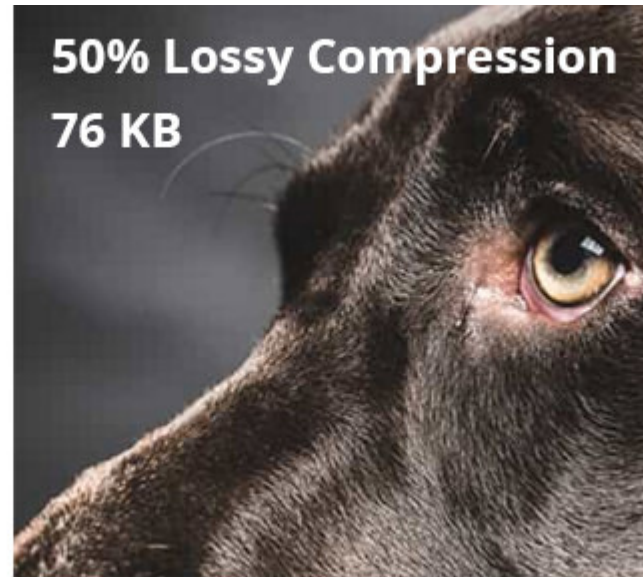


Muchos problemas de aprendizaje automático manejan miles de características.

Esto no sólo **ralentiza el entrenamiento de los modelos**, también puede complicar mucho la tarea de **encontrar una buena solución**. Además **dificulta la visualización** de la información.

Afortunadamente, en problemas reales, suele ser posible **reducir el número de características sin perder mucha información**.

¿Por qué reducir dimensiones?



Hasta cierto punto el perder dimensiones puede ser tolerable
Trade off entre información vs precisión

Beneficios de reducir dimensiones

Permite reducir el **overfitting**. En ese sentido, mejora el desempeño de modelos de clasificación/regresión y aprendizaje no supervisado.

Permite realizar una mejor **visualización** de datos.

Puede resultar en una **reducción de ruido y redundancia**, dependiendo de la naturaleza de los datos.

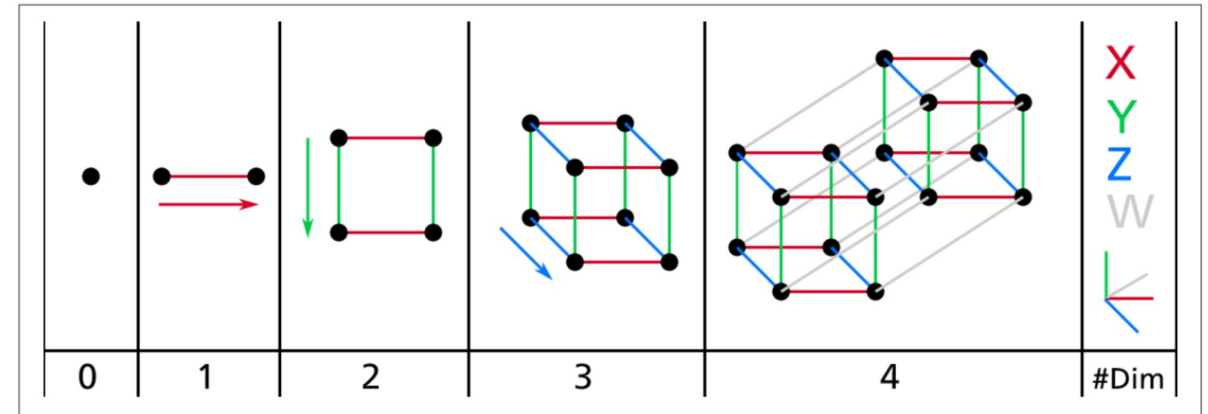


Figure 8-1. Point, segment, square, cube, and tesseract (0D to 4D hypercubes)²

Proyección de datos en espacios de menor dimensión

Las instancias de entrenamiento **no están distribuidas uniformemente**. Algunas características mantienen **valores constantes** y otras están **altamente correlacionadas**. Como resultado, las instancias de entrenamiento **se ubican cerca a una superficie de menor dimensión**.

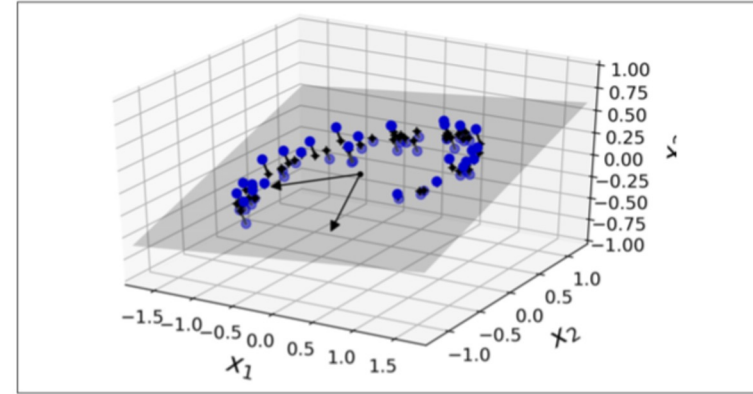


Figure 8-2. A 3D dataset lying close to a 2D subspace

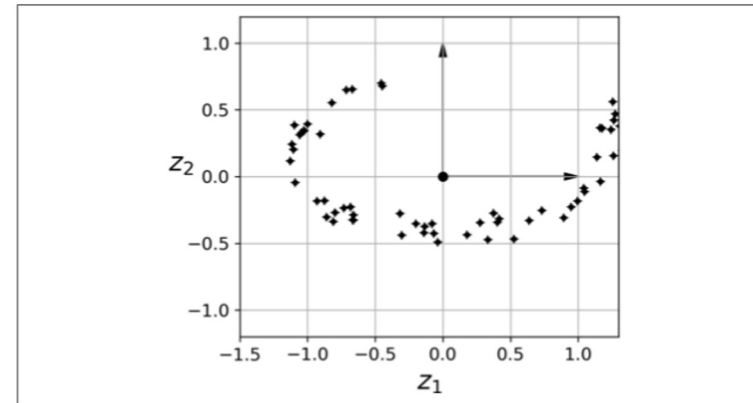
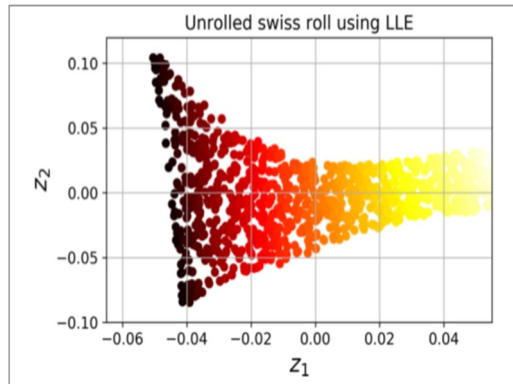


Figure 8-3. The new 2D dataset after projection

Algunos algoritmos para reducir dimensiones

Locally Linear Embedding (LLE)

Calcula la distancia lineal entre instancias vecinas y **construye la superficie de menor dimensión** que mejor preserve la distancia de una instancia a sus k vecinas más cercanas.



Multi-Dimensional Scaling (MDS)

Reduce dimensionalidad **preservando distancias entre las instancias**. Busca la superficie que mejor preserve la distancia entre las instancias de manera general.

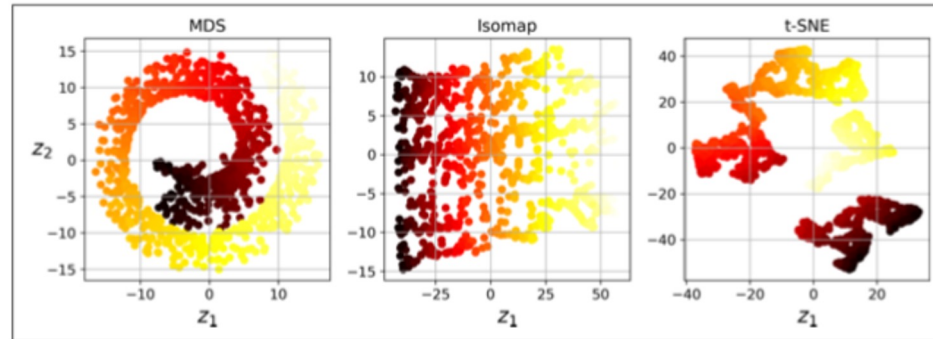


Figure 8-13. Reducing the Swiss roll to 2D using various techniques

Isomap

Conecta las instancias con sus **vecinos más cercanos en un grafo**. Preserva la distancia geodésica, es decir, el camino más corto entre nodos del grafo.

T-Distributed Stochastic Neighbor Embedding

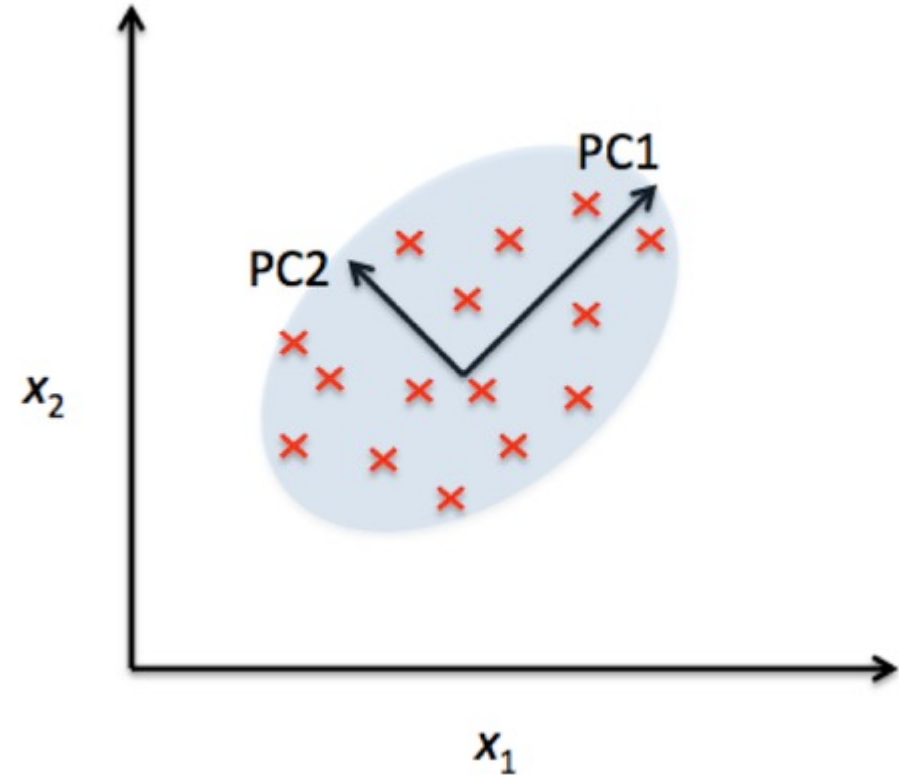
Reduce la dimensionalidad tratando de mantener instancias **similares cercanas y instancias diferentes alejadas**. Se usa más para visualización de datos.

Análisis de Componentes Principales (PCA)

Algoritmo muy popular para **reducción de dimensionalidad**.

Busca proyectar los datos hacia el hiperplano más cercano **preservando la varianza de los datos**.

Puede trabajar sobre **cualquier conjunto de datos**, no requiere asunciones sobre los datos o su distribución.



PCA – Conservar la mayor varianza

Se busca el hiperplano que preserve mejor la varianza de los datos.

En el ejemplo:

¿Que línea permitirá mantener mayor información sobre los datos?

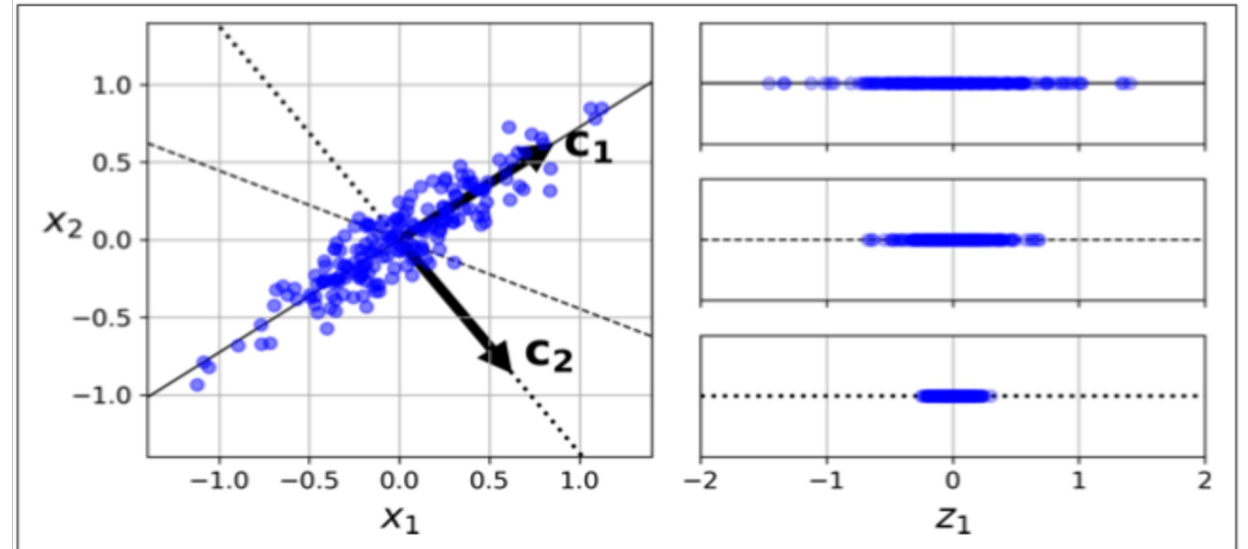
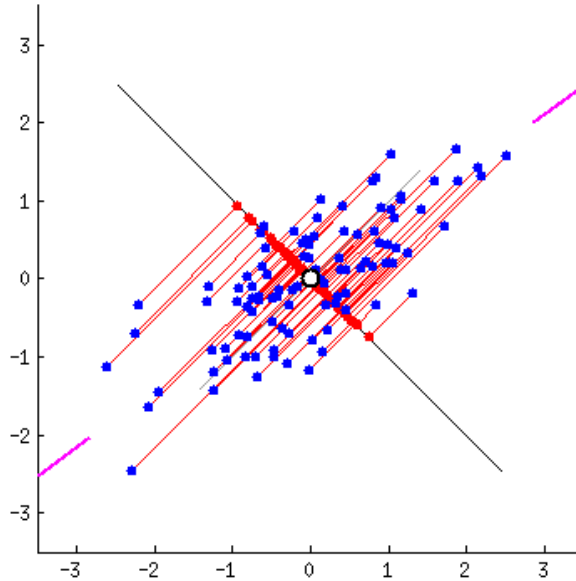
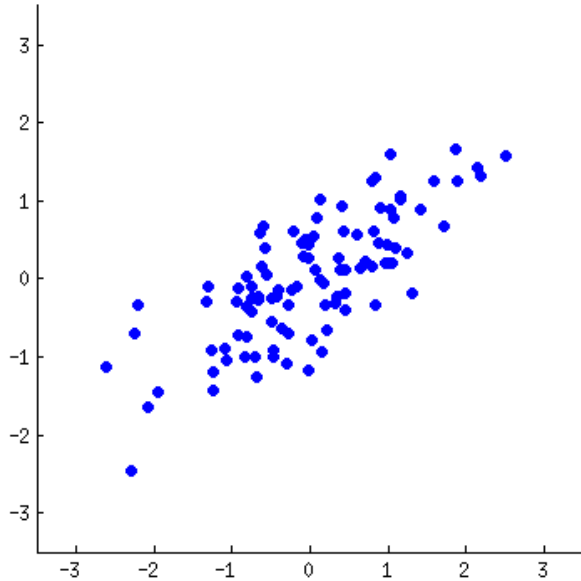


Figure 8-7. Selecting the subspace onto which to project

PCA – ¿Cómo encontramos los componentes? - Idea



1. Se busca una **recta con el mejor ajuste** a los datos.
2. El **primer componente principal** es el vector unitario en la dirección de esa recta.
3. El **segundo componente principal** se obtiene calculando el vector unitario de la recta perpendicular al PC1.
4. Para mayores dimensiones el **i-ésimo PC** se **calcula ajustando la mejor recta perpendicular a los anteriores componentes principales** calculados.

Varianza

La varianza nos ayuda a entender **qué tan dispersa está nuestra variable aleatoria** respecto a la media. Por ejemplo, los ingresos de las personas pueden tener una varianza alta si algunas personas tienen niveles de ingresos elevados.

La fórmula de la varianza para una muestra es la siguiente:

$$\sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

donde n es el número de muestras (por ejemplo, el número de personas) y X barra es la media de la variable aleatoria X(media de los ingresos).

Además, la matriz de covarianza siempre será una matriz cuadrada y su dimensión será equivalente al número de variables

Covarianza

Mide cuánto varían juntas **dos variables aleatorias**.

Más precisamente, la covarianza se refiere a la medida de cómo dos variables aleatorias en un conjunto de datos cambiarán juntas.

Una **covarianza positiva significa que las dos variables están positivamente relacionadas** y se mueven en la misma dirección. Una covarianza negativa significa que las variables están inversamente relacionadas, o que se mueven en direcciones opuestas.

La fórmula para la covarianza es la siguiente:

$$\sigma(x, y) = \frac{1}{n - 1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Matriz de Covarianza

Siguiendo las ecuaciones anteriores, la matriz de covarianza para las dos dimensiones se da por:

$$C = \begin{pmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{pmatrix}$$

En esta matriz, las varianzas aparecen a lo largo de la diagonal, y las covarianzas aparecen en los elementos fuera de la diagonal.

Para el caso general de una matriz de n dimensiones:

$$\Sigma = \begin{pmatrix} Var(X_1) & Cov(X_1, X_2) & \dots & Cov(X_1, X_n) \\ Cov(X_2, X_1) & Var(X_2) & \dots & Cov(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_n, X_1) & Cov(X_n, X_2) & \dots & Var(X_n) \end{pmatrix}$$

Además, la matriz de covarianza siempre será una matriz cuadrada y su dimensión será equivalente al número de variables

Matriz de Covarianza



covarianza
negativa



covarianza cero
(o muy pequeña)



covarianza
positiva

Matriz de Covarianza



$$\Sigma = \begin{pmatrix} \text{Var}(X) & \text{Cov}(X, Y) \\ \text{Cov}(X, Y) & \text{Var}(Y) \end{pmatrix}$$

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

Eigen values & Eigen vectors

Sea A una matriz, necesitamos encontrar los vectores \mathbf{x} y los valores λ , tal que:

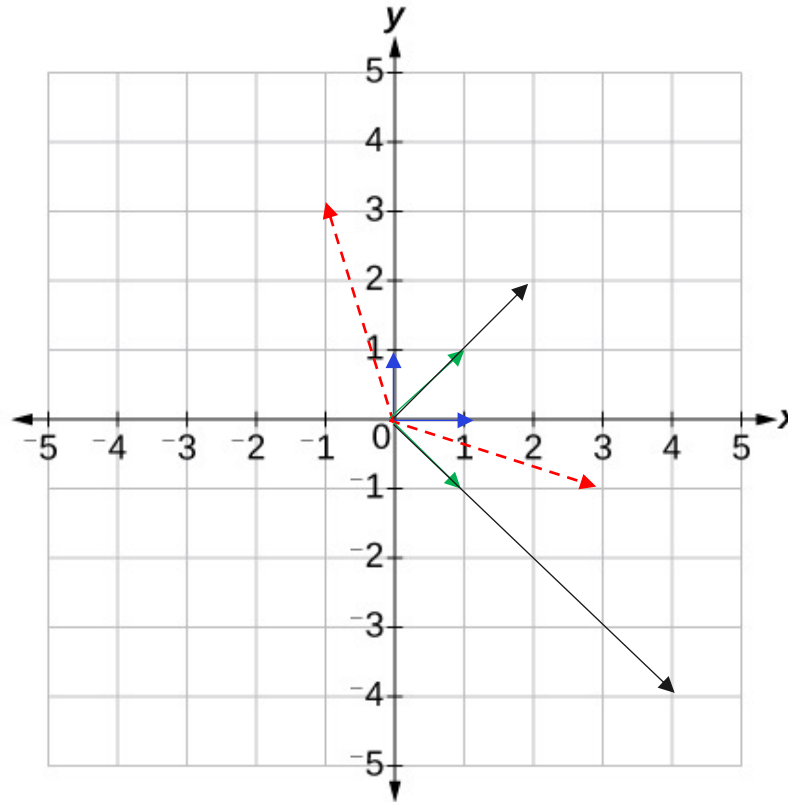
$$A\mathbf{x} = \lambda\mathbf{x}$$

↙↘

Eigenvector Eigenvalue

Por ejemplo, para A:

$$A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$$



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \longrightarrow A\mathbf{x} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \longrightarrow A\mathbf{x} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \longrightarrow A\mathbf{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} = 2\mathbf{x}$$

\mathbf{x} es un eigenvector con eigenvalue $\lambda = 2$

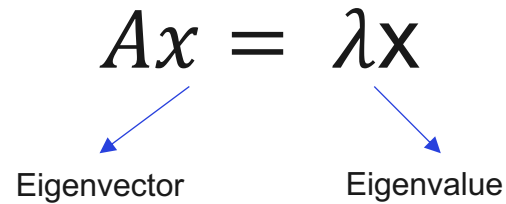
$$\mathbf{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \longrightarrow A\mathbf{x} = \begin{bmatrix} 4 \\ -4 \end{bmatrix} = 4\mathbf{x}$$

\mathbf{x} es un eigenvector con eigenvalue $\lambda = 4$

Eigen values & Eigen vectors

Se A una matriz, necesitamos encontrar los vectores x y los valores λ , tal que:

$$Ax = \lambda x$$



$$Ax = \lambda Ix$$
$$(A - \lambda I) x = 0$$

Si $x \neq 0$:

$$\det(A - \lambda I) = 0$$

Paso 1: Encontrar λ tal que:

$$\det(A - \lambda I) = 0$$

$$A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$$

$$A - \lambda I = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 3 - \lambda & -1 \\ -1 & 3 - \lambda \end{bmatrix}$$

$$\det(A - \lambda I) = (3 - \lambda)(3 - \lambda) - 1 = \lambda^2 - 6\lambda + 8 = 0$$
$$= (\lambda - 4)(\lambda - 2) = 0$$

$$\lambda = 4, \lambda = 2$$

Eigen values & Eigen vectors

Paso 2: Encontrar x para cada λ .

Cuando $\lambda=2$:

$$(A - 2I)x = 0 \rightarrow \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Cuando $\lambda=4$:

$$(A - 4I)x = 0 \rightarrow \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$
$$x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Como conclusión, tenemos que la siguiente matriz A:

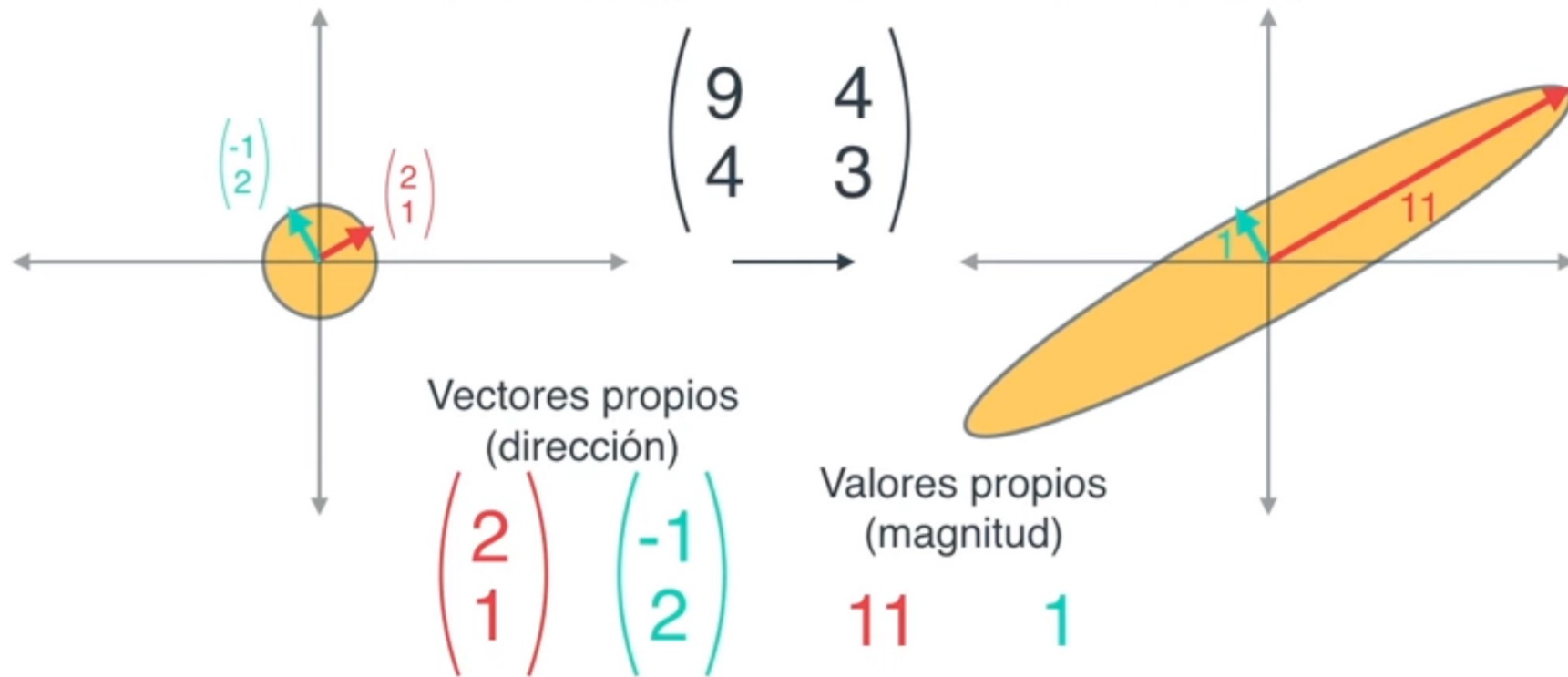
$$A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$$

Tiene los siguientes eigenvalues que corresponden a los siguientes eigenvectors:

$$\lambda = 4, \quad x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\lambda = 2, \quad x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Matriz de Covarianza



Principal Component Analysis

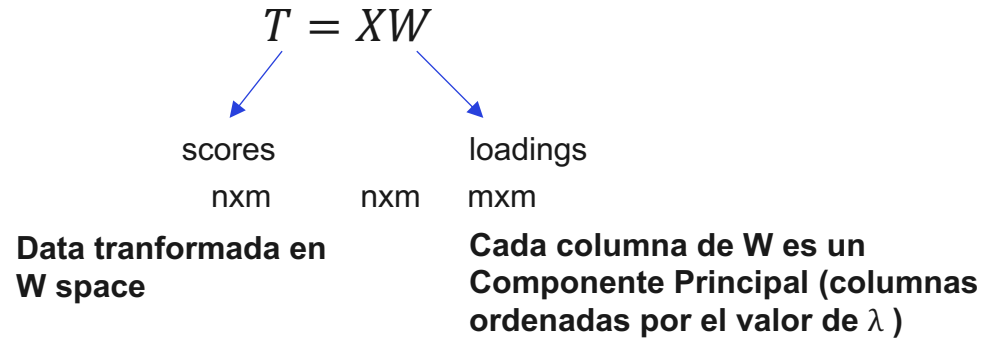
Tenemos una matriz X , de la forma:

$$X = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix} \begin{matrix} \text{n muestras} \\ \\ \end{matrix}$$

nxm matriz m columnas

PCA es el **eigendecomposition** de la matriz de covarianzas de X : (XX^T)

$W \rightarrow \text{eigenvector}$
 $\lambda \rightarrow \text{eigenvalue}$



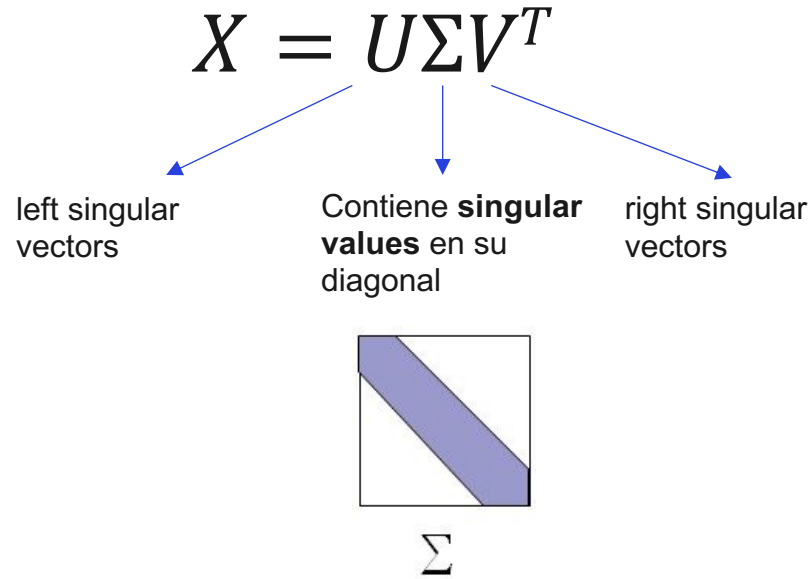
Podemos elegir las primeras r columnas de W .
Por ejemplo $r=2$ (2 primeros Principal Components)

$$T_r = XWr$$

mx2 mxr

Calcular (XX^T) puede ser computacionalmente costoso.
Veamos otra forma de obtener W y λ

Singular Value Decomposition



$$T = XW$$

scores
nxm

loadings
nxm mxm

V es idéntico a W (matriz de loadings visto previamente)

U y V son vectores unitarios:

$$UU^T = I$$

$$VV^T = I$$

Multiplicamos a la derecha por el vector V:

$$XV = U\Sigma V^T V$$

$$T = U\Sigma$$

scores

Varianza total explicada

Sigma es una matriz triangular ordenada, en donde cada sigma corresponde a los eigenvalues (varianza explicada)

$$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_2 \end{bmatrix}$$

$$\text{explained variance of } PC_k = \left(\frac{\text{eigenvalue of } PC_k}{\sum_{i=1}^p \text{eigenvalue of } PC_i} \right)$$

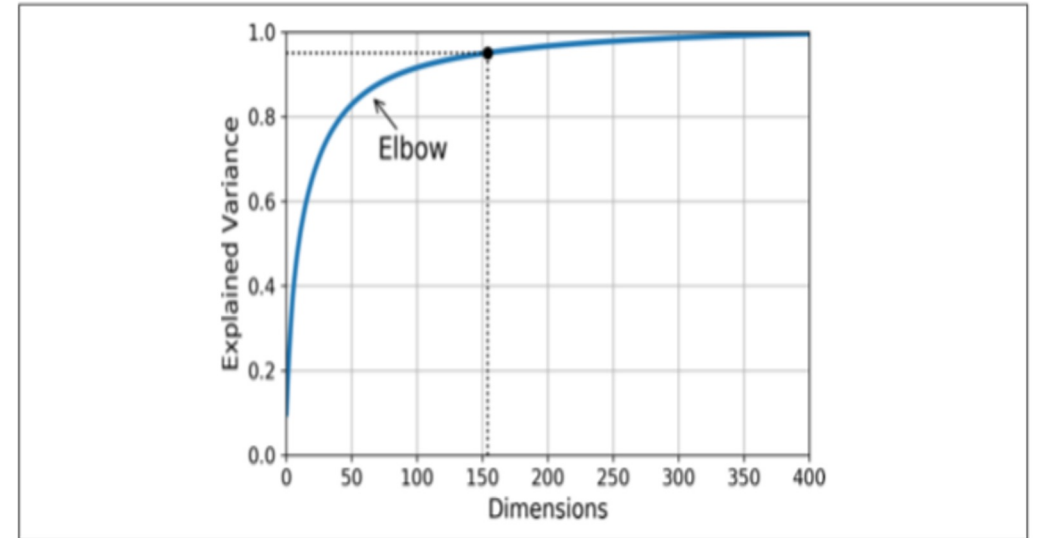


Figure 8-8. Explained variance as a function of the number of dimensions

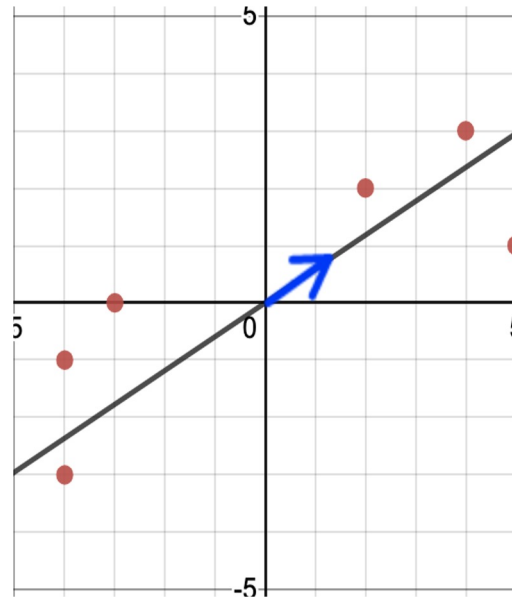
Podemos usar la variable **explained_variance_ratio_** para **calcular la varianza acumulada por componente** Y luego seleccionar la cantidad de dimensiones en base a un porcentaje de información.

PCA1 – Primer componente principal

El **primer componente principal** es el vector unitario paralelo a esta recta.

La pendiente de esta recta se puede interpretar como la contribución de cada característica respecto a la varianza explicada por las proyecciones de las observaciones.

Se le denomina **combinación lineal** de los ejes originales.



Componente Principal (CP): También se le llama vector singular o eigenvector:

$$\text{PCA1} = 0.97 X_1 + 0.242 X_2$$

Cada componente del vector CP se denomina *loading score* de la característica correspondiente.

Eigenvalue de CP: Suma de cuadrados de las proyecciones sobre CP (varianza).

$$\text{SS}(\text{distances for PC1}) = \text{Eigenvalue for PC1}$$

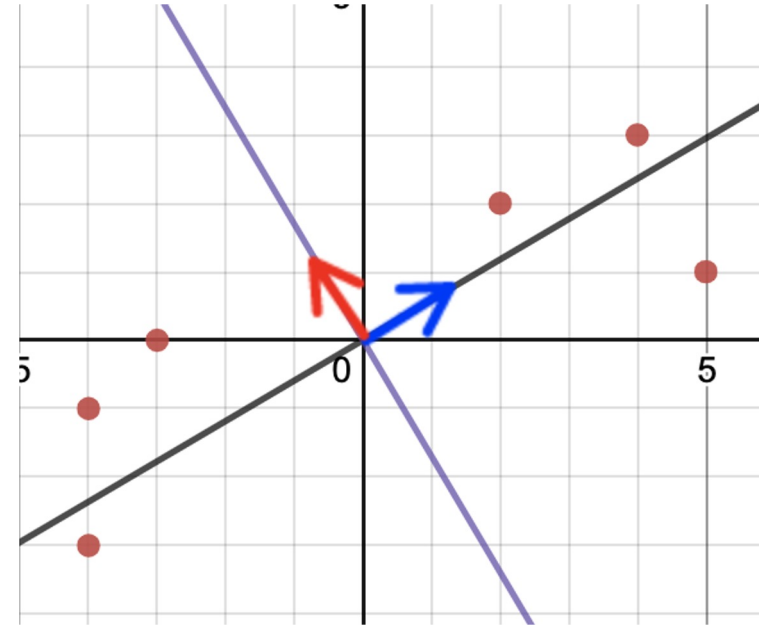
Valor singular de CP: $\sqrt{\text{Eigenvalue}}$

$$\sqrt{\text{Eigenvalue for PC1}} = \text{Singular Value for PC1}$$

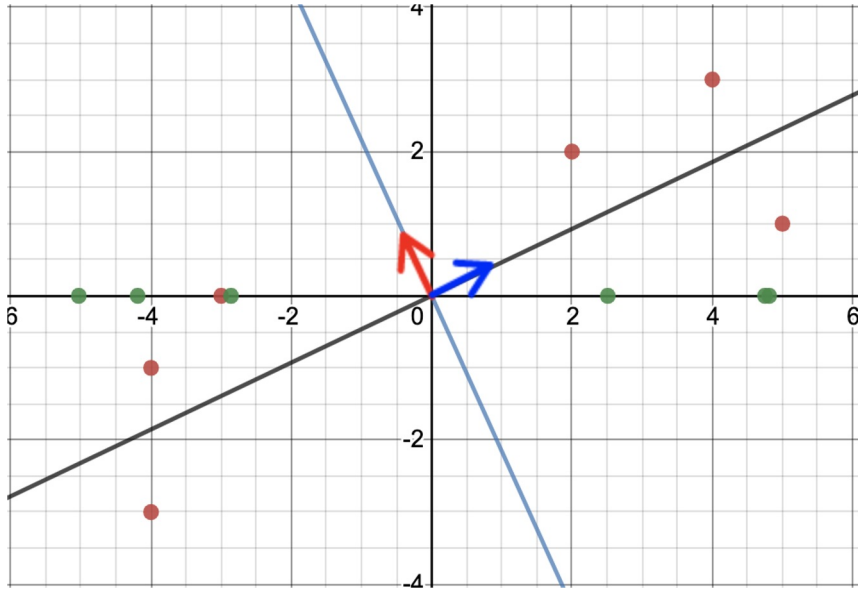
PCA2 – Segundo componente principal

El **segundo componente principal** se obtiene calculando el vector unitario de la recta perpendicular al primer componente principal.

Para mayores dimensiones el i -ésimo componente principal se calcula ajustando la mejor recta perpendicular a los anteriores componentes principales calculados.



Proporción de la varianza explicada



```
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components = 2)  
X2D = pca.fit_transform(X)
```

Una vez que se tiene los **componentes principales** para cada dimensión, se puede proyectar las observaciones a una menor dimensión. Basta con usar los N primeros componentes donde N es menor a la dimensión original de las observaciones.

Se puede calcular la varianza respecto al origen para cada componente **dividiendo la suma de los cuadrados de las proyecciones sobre el PC (eigenvalue) entre el número de observaciones.**

$$\frac{SS(\text{distances for PC1})}{n - 1} = \text{Variation for PC1}$$

$$\frac{SS(\text{distances for PC2})}{n - 1} = \text{Variation for PC2}$$

PC1: 94%

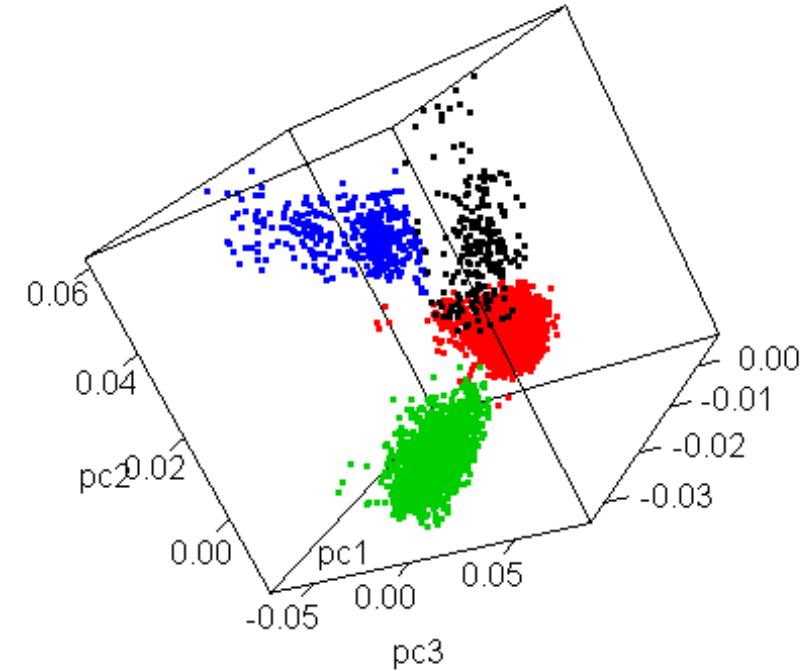
PC2: 6%

Conclusiones - PCA

Es útil cuando existe **multicolinealidad** entre las características/variables.

PCA se puede utilizar cuando las dimensiones de las características de entrada son altas y se quiere **reducir el tiempo de procesamiento** (por ejemplo, muchas variables).

PCA también se puede utilizar para la eliminación de ruido y la **compresión de datos**.



Referencias

Hands on Machine Learning with Scikit-Learn, Keras & Tensor Flow – O'Reilly

Capítulo 8 y 9 Páginas 215-274 (Dimensionality Reduction and Unsupervised Learning Techniques)

Unsupervised Learning, Recommenders, Reinforcement Learning

Unsupervised Learning

Deeplearning.ai @ coursera

<https://www.coursera.org/specializations/machine-learning-introduction>

Curso de Fundamentos de Aprendizaje de Máquina

Diplomado de Desarrollo de Aplicaciones de Inteligencia Artificial, PUCP.