Estadística III para Ingenieros de Sistemas

Jose Daniel Ramirez Soto 2023
jdr2162@columbia.edu

Agenda

- anuncios varios
- modelos de analitica (machine learning-ML) No supervisado
  - PCA
  - K-means, Cluster jerárquico o Dendograma
- Practica uso de clusters y PCA en python

Anuncios

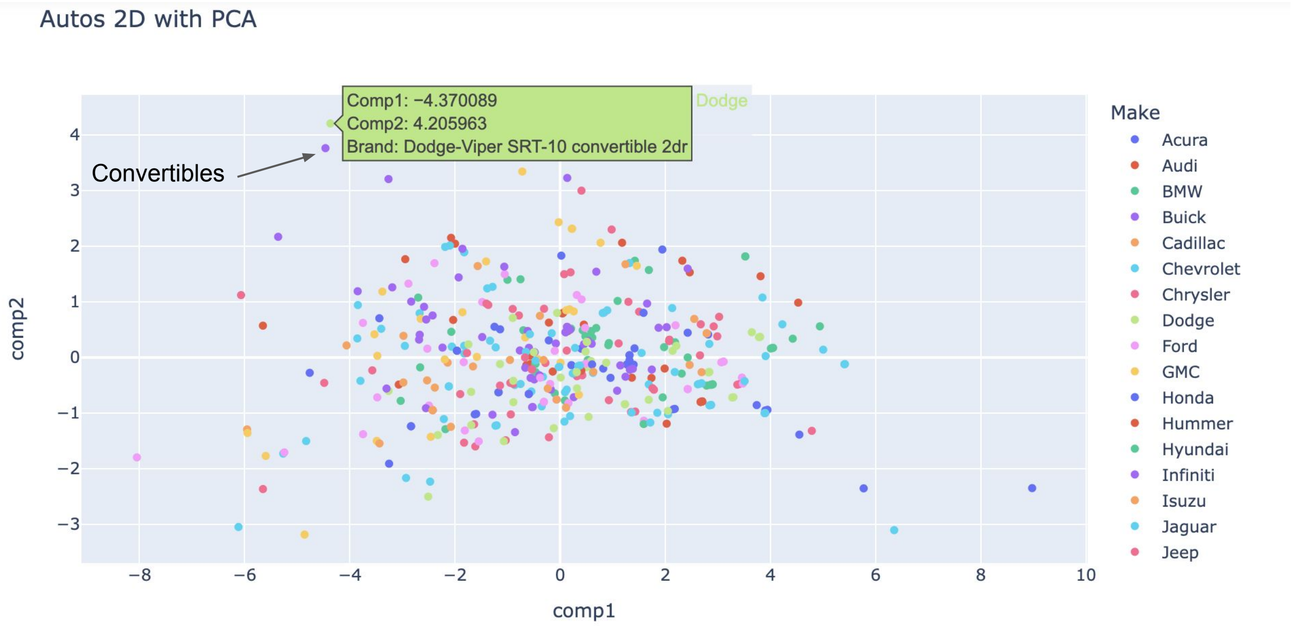
- **Resumen del curso está en el github**
- **La primera tarea se publicará en el github de la clase el día de mañana (24-02-2023 8:00 am). Tiene parte teórica y práctica. Fecha de entrega es el 02-03-2023 11:59 pm. (-1 punto por cada día adicional).**
- **Office hours, Sábados 10:30-11:30.**
- **(Opcional) XVIII JORNADA DE EXPOSICIÓN DE ARTÍCULOS CIENTÍFICOS, seleccionar un artículo de machine learning para presentarlo, tiempo de selección 3 de marzo.**

No supervisado, PCA Principal Components Analysis

PCA, es una técnica que nos permite crear combinaciones de las variables **más representativas** de una base de datos y resumir los datos un numero **menor** de dimensiones. Se utiliza para:

- **Análisis**: Encontrar las variables más importantes.
- **visualizar**: pintar datos en 2 o 3D aunque la base de datos original contenga muchas variables

Ejemplo: utilizando datos continuos del dataset de carros podemos visualizar los que son más similares.



Utilizar 2 variables para resumir 8 variables continuas de los carros

* A Course of Machine Learning http://ciml.info/

No supervisado, PCA Principal Components Analysis

Proceso de PCA: Objetivo de reducir la dimensionalidad conservando la mayoría de información.

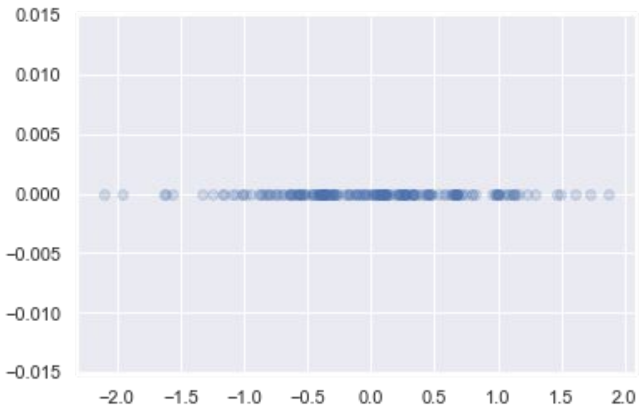
- **Standardize los datos:** Quitar la media y dividir por la desviación estándar.
- **Calcular matriz de covarianza :** Miden como 2 variables varían de forma conjunta

$$Cov(x,y) = \frac{\Sigma\,(x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

- **Calcular eigenvalues :** Es la magnitud de la varianza capturada en cada componente.
- **Calcular eigenvector :** Son los nuevos ejes en los que los datos serán transformados.
- **Seleccionar K:** Cuántas dimensiones tendrá los datos transformados
- **Transformar datos** : Representación de los datos X en una menor dimensión.

X, datos originales

X*, datos transformados



* A Course of Machine Learning http://ciml.info/

No supervisado, PCA Principal Components Analysis

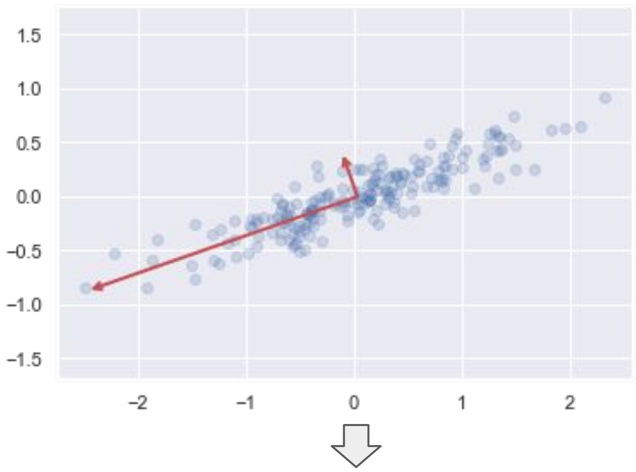
Proceso de PCA: Objetivo de reducir la dimensionalidad conservando la mayoría de información.

- **Standardize los datos:** Quitar la media y dividir por la desviación estándar.
- **Calcular matriz de covarianza :** Miden como 2 variables varían de forma conjunta

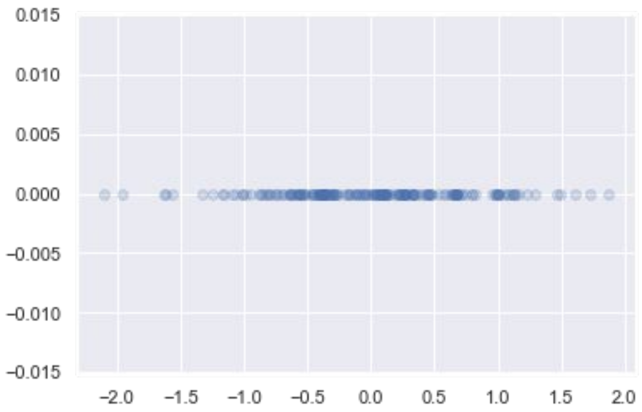
$$Cov(x,y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

- **Calcular eigenvalues :** Es la magnitud de la varianza capturada en cada componente.
- **Calcular eigenvector :** Son los nuevos ejes en los que los datos serán transformados.
- **Seleccionar K:** Cuántas dimensiones tendrá los datos transformados
- **Transformar datos** : Representación de los datos X en una menor dimensión.

X, datos originales



X*, datos transformados



* A Course of Machine Learning http://ciml.info/
*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

No supervisado, PCA Principal Components Analysis

$$\mathbf{X} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 2 & 2 \\ -2 & -2 \end{pmatrix}$$

* A Course of Machine Learning http://ciml.info/
*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

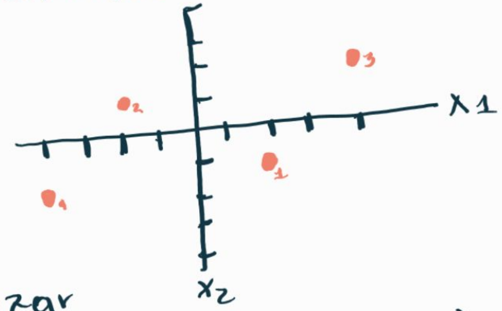
No supervisado, PCA Principal Components Analysis(estandarizar)

Universidad Católica de Oriente

PCA

Reducir la dimension 2D a 1D manteniendo la mayor varianza.

$$
X = \begin{array}{c c} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 2 & -1 \\ -2 & 1 \\ 4 & 2 \\ -4 & -2 \end{bmatrix} \\ & \begin{array}{cc} 1 & 2 \end{array} \end{array}
$$



1) centrar y estandarizar

$$
u_1 = \frac{(2-2+4-4)}{4} = 0 \qquad u_2 = \frac{(-1+1+2-2)}{4} = 0
$$

$$
\sigma^2_{x_1} = \frac{\sum_{i=1}^{n}(x_{1_i} - u_1)^2}{n} \qquad \sigma^2_{x_2} = \frac{\sum_{i=1}^{n}(x_{2_i} - u_2)^2}{n}
$$

$$
\sigma^2_{x_1} = \frac{4+4+16+16}{4} = 10 \qquad \sigma^2_{x_2} = \frac{1+1+4+4}{4} = \frac{5}{2}
$$

$$
X_N = \begin{bmatrix} 0.63 & -0.63 \\ -0.63 & 0.63 \\ 1.26 & 1.26 \\ -1.26 & -1.26 \end{bmatrix} \qquad \begin{array}{l} u_1 = u_2 = 0 \\ \sigma_1 = \sigma_2 = 1 \end{array}
$$

* A Course of Machine Learning http://ciml.info/

*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

No supervisado, PCA Principal Components Analysis(covarianza)

2)

Calcular Matriz de covarianza

$$\Sigma = \begin{bmatrix} \sigma_{x_1}^2 & cov(x_1, x_2) \\ cov(x_2, x_1) & \sigma_{x_2}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

$$cov(x_1, x_2) = \frac{\sum_{i=1}^{n} (x_1 - \mu_1)(x_2 - \mu_2)}{N} = \frac{0.63 * -0.63 + (-0.63 * 0.63) + 1.26 * 1.26 + (-1.26 * -1.26)}{4} = \frac{2.4}{4} = 0.6$$

$$cov(x_2, x_1) = \frac{\sum_{i=1}^{n} (x_2 - \mu_2)(x_1 - \mu_1)}{N} = 0.6$$

$$\sigma_{x_1}^2 = \frac{\sum_{i=1}^{N} (x_1 - \mu_1)^2}{4} = \frac{0.4 + 0.4 + 1.6 + 1.6}{4} = 1$$

$$\sigma_{x_2}^2 = \frac{\sum_{i=1}^{N} (x_2 - \mu_2)^2}{4} = \frac{0.4 + 0.4 + 1.6 + 1.6}{4} = 1$$

* A Course of Machine Learning http://ciml.info/

*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

No supervisado, PCA Principal Components Analysis(eigenvalues)

UCO Universidad Católica de Oriente

3) Hallar eigenvalues que corresponden a los

$$\det(\Sigma - \lambda I) = 0 \quad , \sum_{i=1}^{D} W_i^T \Sigma W_i - \sum_{i=1}^{K} W_i^T \Sigma W_i$$

$$\det\left(\begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = \det \begin{pmatrix} 1-\lambda & 0.6 \\ 0.6 & 1-\lambda \end{pmatrix}$$

$$= (1-\lambda)^2 - 0.6^2$$

$$= 1 - 2\lambda + \lambda^2 - 0.36 = 0.64 - 2\lambda + \lambda^2$$

$$\lambda = \frac{-b \pm \sqrt{b2 - 4ac}}{2a} = \lambda = \frac{+2 \pm \sqrt{4 - 4x0.64}}{2}$$

$$\lambda = \frac{2 \pm \sqrt{4 - 2.56}}{2} = \frac{2 \pm 1.2}{2} \Rightarrow \boxed{\begin{matrix} \lambda_1 = 1.6 \\ \lambda_2 = 0.4 \end{matrix}}$$

calcolate eigen vector con $\lambda_1 = 1.6$ que es el valor mayor

* A Course of Machine Learning http://ciml.info/

*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

No supervisado, PCA Principal Components Analysis(eigenvector)

calculate eigen vector con $\lambda_1 = 1.6$ que es el valor mayor

$$\begin{bmatrix}1 & 0.6\\ 0.6 & 1\end{bmatrix}\begin{bmatrix}X_1\\ X_2\end{bmatrix} = \lambda\begin{bmatrix}X_1\\ X_2\end{bmatrix}$$

① $X_1 + 0.6X_2 = 1.6X_1 \Rightarrow -0.6X_1 = -0.6X_2$
$X_1 = X_2$

② $0.6X_1 + X_2 = 1.6X_2 \Rightarrow 0.6X_1 - 0.6X_2 = 0 \Rightarrow X_1 = X_2$

Si resolvemos el sistema de ecuaciones nos da $0 = 0$ que quiere decir infinitas soluciones. Por lo tanto cualquier vector $x_1 = x_2$ cumple esta condición $V_1 = \begin{bmatrix}1\\ 1\end{bmatrix}$

Un test simple es validar que $\Sigma V_1 = \lambda_1 V_1$

$$\begin{bmatrix}1 & 0.6\\ 0.6 & 1\end{bmatrix}\begin{bmatrix}1\\ 1\end{bmatrix} = 1.6\begin{bmatrix}1\\ 1\end{bmatrix} \Rightarrow \begin{bmatrix}1.6\\ 1.6\end{bmatrix} = 1.6\begin{bmatrix}1\\ 1\end{bmatrix}$$

Retomando la teoria de PCA, la norma de U es 1

$$\max_{u} \sum_{n} (x_n \cdot u)^2 \quad \text{subj. to} \quad ||u||^2 = 1$$

Por lo tanto el eigenvector V_1 debe tener una norma de 1. Como $x_1 = x_2$ entonces

$||V_1|| = \sqrt{x_1^2 + x_2^2} = 1 \Rightarrow 1 = \sqrt{2X_1} \Rightarrow X_1 = \sqrt{0.5} = 0.707$

Calcularemos el eigen vector 2 para $\lambda_2 = 0.4$

Recuerda siempre llevar el vector a ser unitario, ||v||=1, V1= [0.707 0.707]

* A Course of Machine Learning http://ciml.info/
*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

No supervisado, PCA Principal Components Analysis(eigenvector)

Calcularemos el eigen vector z para $\lambda_2 = 0.4$

$$\begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0.4 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

① $x_1 + 0.6x_2 = 0.4\,x_1 \Rightarrow 0.6x_1 + 0.6x_2 = 0 \Rightarrow x_1 = -x_2$

② $0.6x_1 + x_2 = 0.4\,x_2 \Rightarrow 0.6x_1 + 0.6x_2 = 0 \Rightarrow x_1 = -x_2$

$0 = 0$ es infinitas soluciones, cualquier vector $x_1 = -x_2$ cumple $v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

la norma de $\|v_2\| = 1 \Rightarrow 1 = \sqrt{x_1^2 + x_2^2} \Rightarrow 1^2 = x_1^2 + (-x_1)^2$

$X_1 = 0.707 \quad X_2 = -0.707$

$\lambda_1 = 1.6$, representa $\lambda_1 / \sum_{i=1}^{k} \lambda_i$, $\frac{1.6}{2} = 80\%$ de la varianza

* A Course of Machine Learning http://ciml.info/
*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

No supervisado, PCA Principal Components Analysis(proyectar)

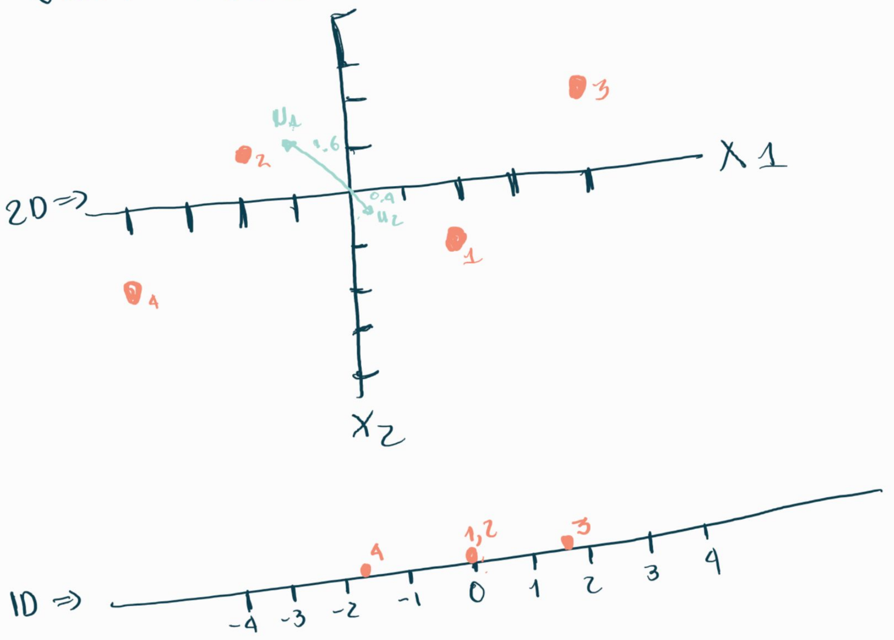
4) Seleccionar e numero de dimensiones, 1D

$$\tilde{X} = X v_1 v_1^T + X v_2 v_2^T \quad \text{or} \quad XVV^T \quad \text{or} \quad ZV^T$$

Utilizando los eigen vectors podemos transformar de 2D ⇒ 1D

$$Z = \begin{bmatrix} 0.63 & -0.63 \\ -0.63 & 0.63 \\ 1.26 & 1.26 \\ -1.26 & -1.26 \end{bmatrix} \cdot \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1.781 \\ -1.781 \end{bmatrix}$$ Proyección 1D

varianza.



El resultado es la reducción de 2 dimensiones a 1 dimensión. Después podemos devolvernos y notar los errores en la reconstrucción

transformar datos de 1D a 2D, ZV^T

$$\tilde{X} = \begin{bmatrix} 0 \\ 0 \\ 1.781 \\ -1.781 \end{bmatrix} \cdot \begin{bmatrix} 0.707 & 0.707 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1.259 & 1.259 \\ -1.259 & -1.259 \end{bmatrix}$$

Como podemos ver $\hat{X}$ es muy similar a los datos X

$$\hat{X} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1.259 & 1.259 \\ -1.259 & -1.259 \end{bmatrix} \qquad X = \begin{bmatrix} 0.63 & -0.63 \\ -0.63 & 0.63 \\ 1.26 & 1.26 \\ -1.26 & -1.26 \end{bmatrix}$$

* A Course of Machine Learning http://ciml.info/
*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

No supervisado, PCA Principal Components Analysis(Python)

Universidad Católica de Oriente

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import pandas as pd
import numpy as np
X = pd.DataFrame([[2,-1],[-2,1],[4,2],[-4,-2]] , columns=["x1","x2"])
print("Estandarizar")
scaler = StandardScaler()
scaler.fit(X)
X_sc = scaler.transform(X)
print("Eigenvalues: \n",X_sc)
print("Entrenar PCA")
pca = PCA(n_components=2)
pca.fit(X_sc)
print("Eigenvalues: \n",pca.explained_variance_)
print("% Varianza explicada: \n",pca.explained_variance_ratio_)
print("% Varianza explicada: \n",pca.explained_variance_ratio_)
print("Proyectar los datos: \n",pca.transform(X_sc))
```

```
Estandarizar
Eigenvalues:
 [[ 0.63245553 -0.63245553]
 [-0.63245553  0.63245553]
 [ 1.26491106  1.26491106]
 [-1.26491106 -1.26491106]]
Entrenar PCA
Eigenvalues:
 [2.13333333 0.53333333]
% Varianza explicada:
 [0.8 0.2]
% Varianza explicada:
 [0.8 0.2]
Proyectar los datos:
 [[ 1.67871766e-17  8.94427191e-01]
 [-1.67871766e-17 -8.94427191e-01]
 [ 1.78885438e+00  3.35743533e-17]
 [-1.78885438e+00 -3.35743533e-17]]
```

```
pca = PCA(n_components=1)
pca.fit(X_sc)
print("Eigenvalues: \n",pca.explained_variance_)
print("% Varianza explicada: \n",pca.explained_variance_ratio_)
print("% Varianza explicada: \n",pca.explained_variance_ratio_)
print("Proyectar los datos: \n",pca.transform(X_sc))
```

```
Estandarizar
Eigenvalues:
 [[ 0.63245553 -0.63245553]
 [-0.63245553  0.63245553]
 [ 1.26491106  1.26491106]
 [-1.26491106 -1.26491106]]
Entrenar PCA
Eigenvalues:
 [2.13333333]
% Varianza explicada:
 [0.8]
% Varianza explicada:
 [0.8]
Proyectar los datos:
 [[ 0.        ]
 [ 0.        ]
 [ 1.78885438]
 [-1.78885438]]
```

Utilizar python para calcular la proyección (mismo resultado), algunos eigenvalores pueden diferir por la fórmula de covarianza utilizada (población vs muestra)

* A Course of Machine Learning http://ciml.info/
*https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613

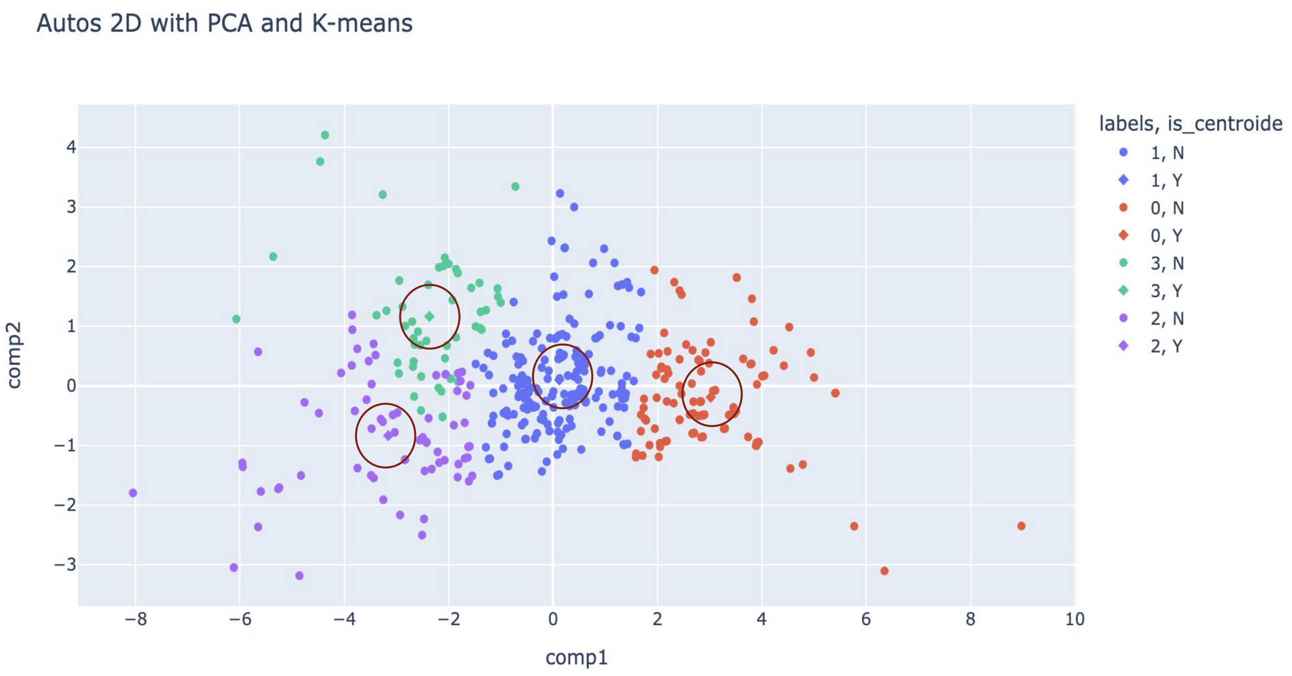
No Supervisado, K-means (Clusters)

Cuando no se tiene etiquetas, el clustering puede ayudarnos a entender cómo crear agrupaciones de los datos basados en su similaridad.

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Parámetro K es el numero grupos que quiero obtener del cluster

Ejemplo: utilizando datos del dataset de automóviles



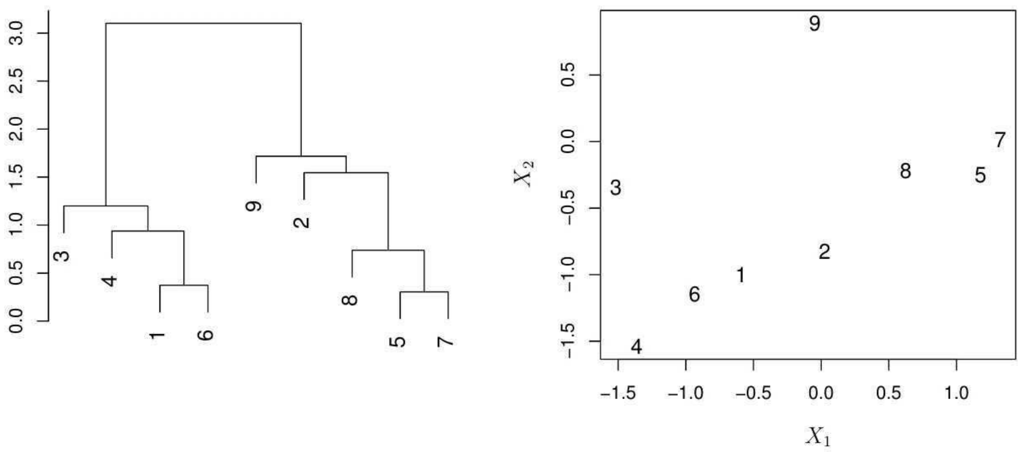
* A Course of Machine Learning http://ciml.info/dl/v0_99/ciml-v0_99-ch15.pdf

No supervisado, Dendograma o cluster jerárquico

Proceso de :

- **Inicia:** Cada observación es un cluster de 1 registro.
- **Medir la distancia entre los clusters:** Medir las distancia entre los clusters:
  - Complete: Max intercluster diferencias, por pares
  - Single: Minimal intercluster diferencias, por pares
  - Average: Mean intercluster diferencias,
  - Centroid: Distancia entre los centroides
- **Combinar**: combinar los que tengan menor diferencias
- **Repetir** paso 2 y 3 hasta tener 1 solo grupo

Ejemplo: utilizando ejemplo con 9 registros



* A Course of Machine Learning http://ciml.info/

No supervisado, Dendograma o cluster jerárquico

Proceso de :

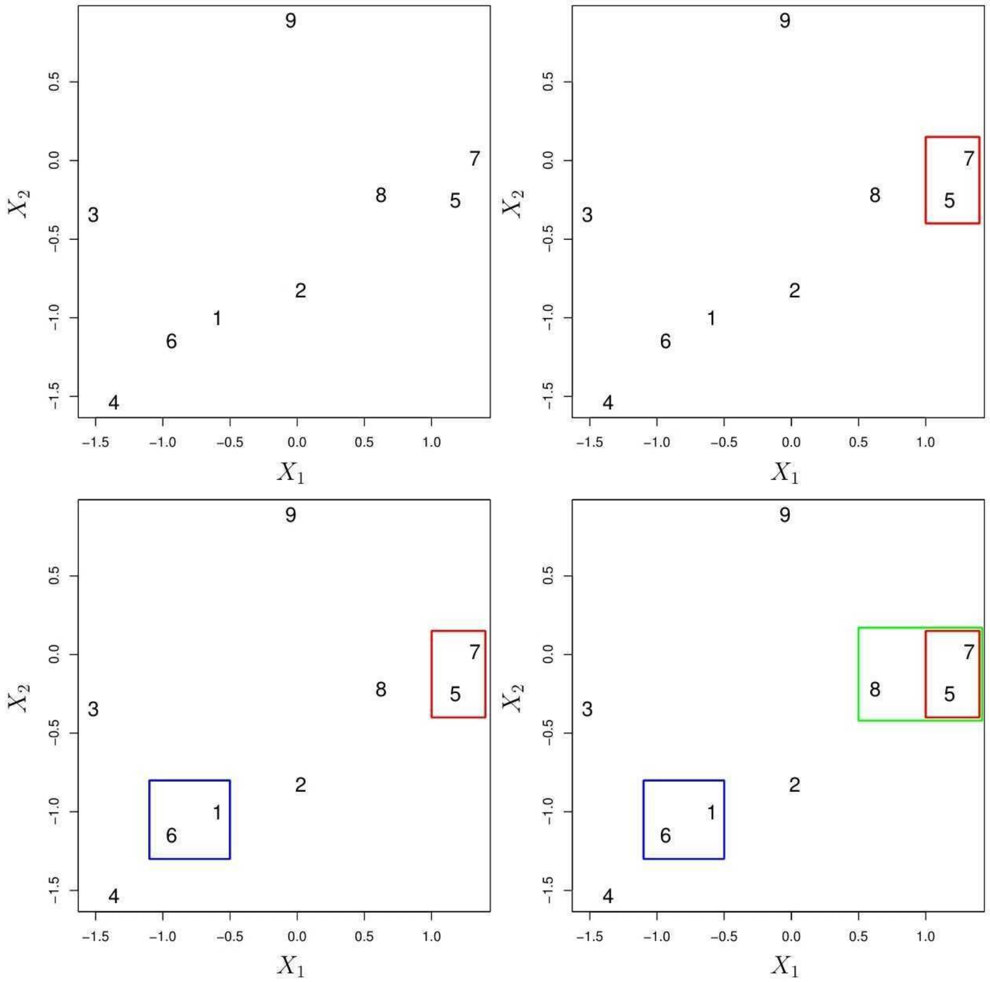
- **Inicia:** Cada observación es un cluster de 1 registro.
- **Medir la distancia entre los clusters:**

  {7} con {5,8,2,9,1,6,3,4}

  {6} con {1,4,2,3,8,5,9} …..
- **Combinar**: combinar los que tengan menor diferencias

  {7,5} 5,8,2,9,1,6,3,4
- **Repetir** paso 2 y 3 hasta tener 1 solo grupo

Ejemplo: utilizando ejemplo con 9 registros



* A Course of Machine Learning http://ciml.info/