Principal Component Analysis

November 3, 2020

0.0.1 EJEMPLO 1

[4]:

```
# Análisis de Componentes Principales

from numpy import array, mean, cov
from numpy.linalg import eig
# Definiendo la matriz
A = array([
    [1, 2],
    [3, 4],
    [5, 6]
])
print(A)
# Media de las columnas
M = mean(A.T, axis = 1)
# Centrando la matriz
C = A - M
# Calculando la matriz de covarianzas
V = cov(C.T)
# Factorizando la matriz de covarianzas
valores, vectores = eig(V)
print(valores)
print(vectores)
# Proyectando los datos
P = vectores.T.dot(C.T)
print(P.T)
```

```
[[1 2]
 [3 4]
 [5 6]]
[8. 0.]
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
[[-2.82842712  0.        ]
 [ 0.          0.        ]
 [ 2.82842712  0.        ]]
```

0.0.2 EJEMPLO 2

[6]:

```python
# PCA con scikit-learn

from numpy import array
from sklearn.decomposition import PCA
# Definiendo la matriz
A = array([
    [1, 2],
    [3, 4],
    [5, 6]
])
print(A)
# Creando la transformación
pca = PCA(2)
# Aplicando la transformación
pca.fit(A)
# Obteniendo los valore sy vectores
print(pca.components_)
print(pca.explained_variance_)
# Transformando los datos
B = pca.transform(A)
print(B)
```

```
[[1 2]
 [3 4]
 [5 6]]
[[ 0.70710678  0.70710678]
 [ 0.70710678 -0.70710678]]
[8.00000000e+00 2.25080839e-33]
[[-2.82842712e+00  2.22044605e-16]
 [ 0.00000000e+00  0.00000000e+00]
 [ 2.82842712e+00 -2.22044605e-16]]
```