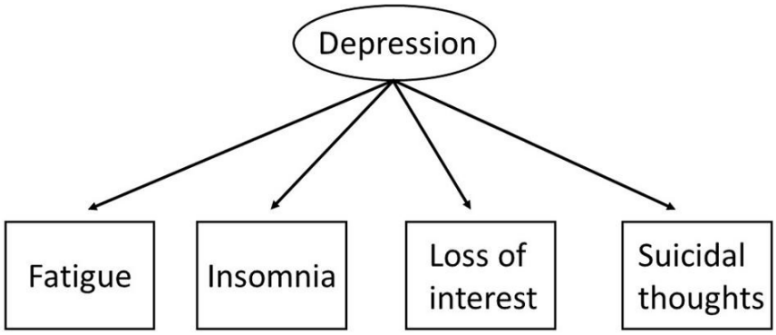
Principle Component Analysis (PCA)

an unsupervised learning technique used for **reducing the dimensionality of the data**. The method of representing the data with **approximately equivalent data in a reduced dimension**. PCA is considered to be part of data preprocessing.

We do PCA because **some features are redundant and/or noisy**!

- ex: data with 100 different variables from sensors
- only 2 sensors might have significant information
- 10 sensors have sometimes marginal data
- 88 sensors have insignificant information
- pick just a few of the first components that satisfy the problem

PCA involves transforming observed variables into a set of latent variables such that we lose minimal data when dropping into a lower dimension.

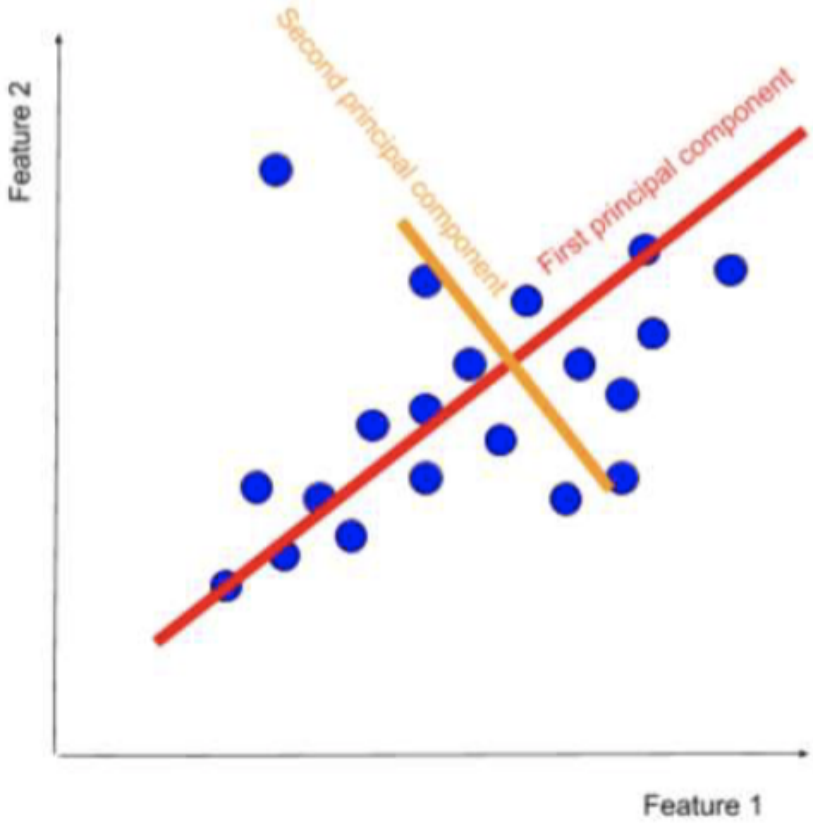


(source : [3])

Principle Components

the **direction of the data that explain a maximal amount of variance**

- the lines that capture the most amount data



(source:[6])

Applications of PCA

- visualizing multidimensional data
- compressing information
- simplification of complex business decisions

Advantages

- easy to compute
- speeds up other machine learning algorithms
- counteracting the issues of high-dimensional data by preventing overfitting

Disadvantages

- low interpretability of principle components
  - not easy to interpret the "meaning" of each component. This is because each data point can be plotted by some linear combination of the other attributes
- trade off between information loss and dimensionality reduction

- it doesn't directly consider the correlation between the principal components and the target variable in the original dataset

PCA also suffers from some llimitations

- sensitive to the scale of features (so we must standardize the data)
- not robust against outliers (biased against strong outliers)

Procedure

a dataset with N samples has D features. X is the data set represented as a matrix

$$X = \begin{bmatrix} X_1^T \\ X_2^T \\ X_3^T \\ \vdots \\ X_n^T \end{bmatrix} \in \mathbb{R}^{\mathbb{n} \times \mathbb{D}}$$

we want to find a new orthonormal basis in D that **best approximates X**

- A new orthonormal basis in the D-dimensional space

$$\begin{aligned} \mathbf{x}_1 &= z_{11}\mathbf{w}_1 + z_{21}\mathbf{w}_2 + \cdots z_{D1}\mathbf{w}_D \\ \mathbf{x}_2 &= z_{12}\mathbf{w}_1 + z_{22}\mathbf{w}_2 + \cdots z_{D2}\mathbf{w}_D \\ &\vdots \\ \mathbf{x}_n &= z_{1n}\mathbf{w}_1 + z_{2n}\mathbf{w}_2 + \cdots z_{Dn}\mathbf{w}_D \end{aligned}$$

- x can be approximated with d component which has large power

$$\begin{aligned} \mathbf{x}_i &\approx \mathbf{W}_d \mathbf{z}_i \\ \mathbf{W} &= [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d] \\ \mathbf{z}_i &= [z_{1i}, z_{2n}, \cdots z_{dn}]^T \end{aligned}$$

$$\mathcal{L}(W, \{\mathbf{z_i}\}_{i=1}^N) = \frac{1}{N}\sum^{N} ||\mathbf{x_n} - W\mathbf{z_n}||_2^2$$
$$= \frac{1}{N}\sum_{n=1}^{N} \mathbf{x_n}^T\mathbf{x_n} - 2\mathbf{x_n}^T W^T\mathbf{z_n} + \mathbf{z_n}^T\mathbf{z_n} \qquad \because W^TW = I$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z_i}}(W, \mathbf{z_i}) = \frac{\partial}{\partial \mathbf{z_i}}(-2\mathbf{x_i}^T W^T\mathbf{z_n} + \mathbf{z_i}^T\mathbf{z_i})$$
$$= -2\mathbf{x_n}^T W^T + 2\mathbf{z_i}^T$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z_i}}(W, \mathbf{z_i}) = 0$$
$$\implies \mathbf{x_i}^T W = \mathbf{z_i}^T$$
$$\implies \mathbf{z_i} = W^T\mathbf{x_i}$$

1. standardize the range of continuous initial variables
2. compute the covariance matrix of standardized variables
3. compute the eigenvectors and eigenvalues of the covariance matrix to identify the principall components
4. decide which principal components to keep and create a transformation matrix
5. recast the data along the new principal component axes

1. Standardizing

standardize the data to the range of continuous initial variables to "fairly" represent their contributions

$$Z = \frac{value - mean}{std.\,dev}$$

2. Compute Covariance Matrix

$$\mathrm{cov}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{y})}{n-1}$$

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \quad \bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \setminus$$

$$X = [X_1, X_2, \cdots, X_D]$$

$$K_X(i, j) = \mathrm{cov}(X_i, X_j)$$

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix} \leftarrow$$

Covariance matrix
3-dimensional data set

3. Find Eigenvectors

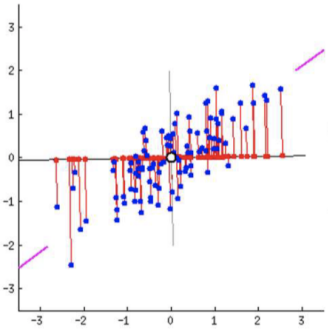
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components

$$K_X = Q \Lambda Q^T$$

One can use software tools

```
from numpy.linalg import eig
eigenvalues, eigenvectors = eig(cov_matrix)
```

Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components



$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

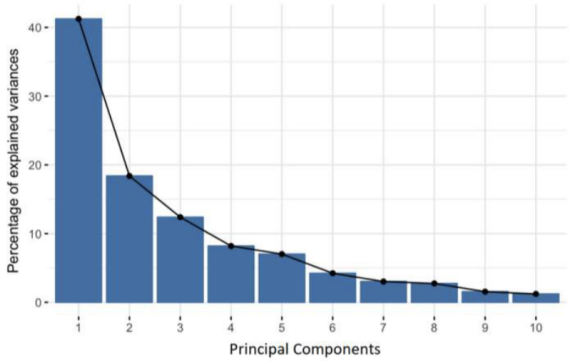
$$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

PC1 and PC2 carry 96% and 4% of the variance of the data.

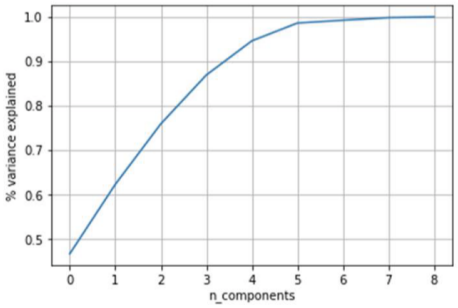
https://builtin.com/data-science/step-step-explanation-principal-component-analysis

4. Decide what to keep

4. Decide which principal components to keep and create transformation matrix



https://builtin.com/data-science/step-step-explanation-principal-component-analysis



https://towardsdatascience.com/how-to-remove-multicollinearity-in-dataset-using-pca-4b4561c28d0b#:~:text=Why%20is%20Multicollinearity%20a%20Potential,statistical%20significance%20of%20independent%20variables%20.

- typically we try to aim for principle components which cover **90%** of the data