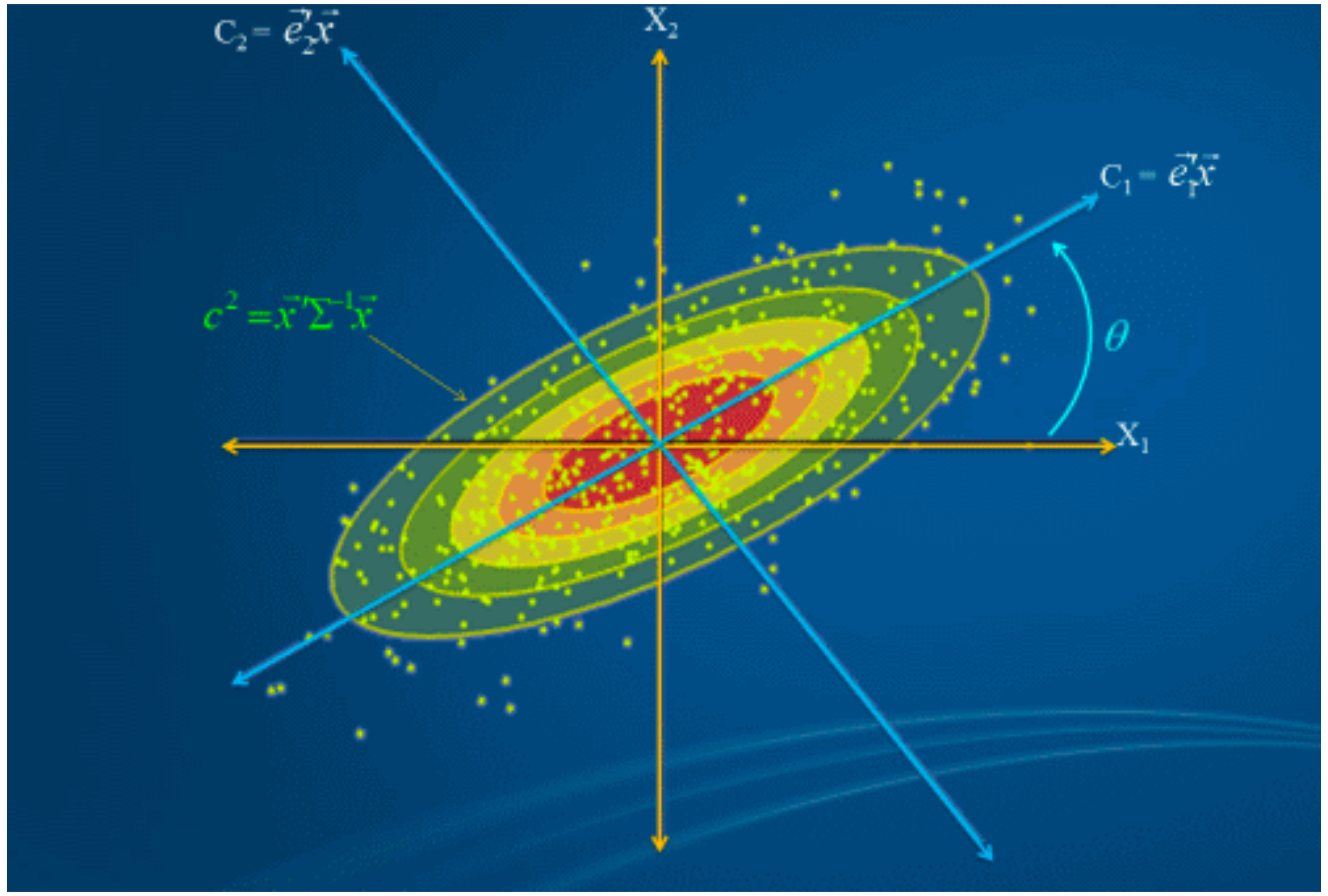
Machine Learning with Python
Unsupervised Learning - PCA

By Dr. Maryam Rahbaralam


$C_2 = \vec{e}_2^T\vec{x}$
X_2
$C_1 = \vec{e}_1^T\vec{x}$
$c^2 = \vec{x}^T\Sigma^{-1}\vec{x}$
θ
X_1

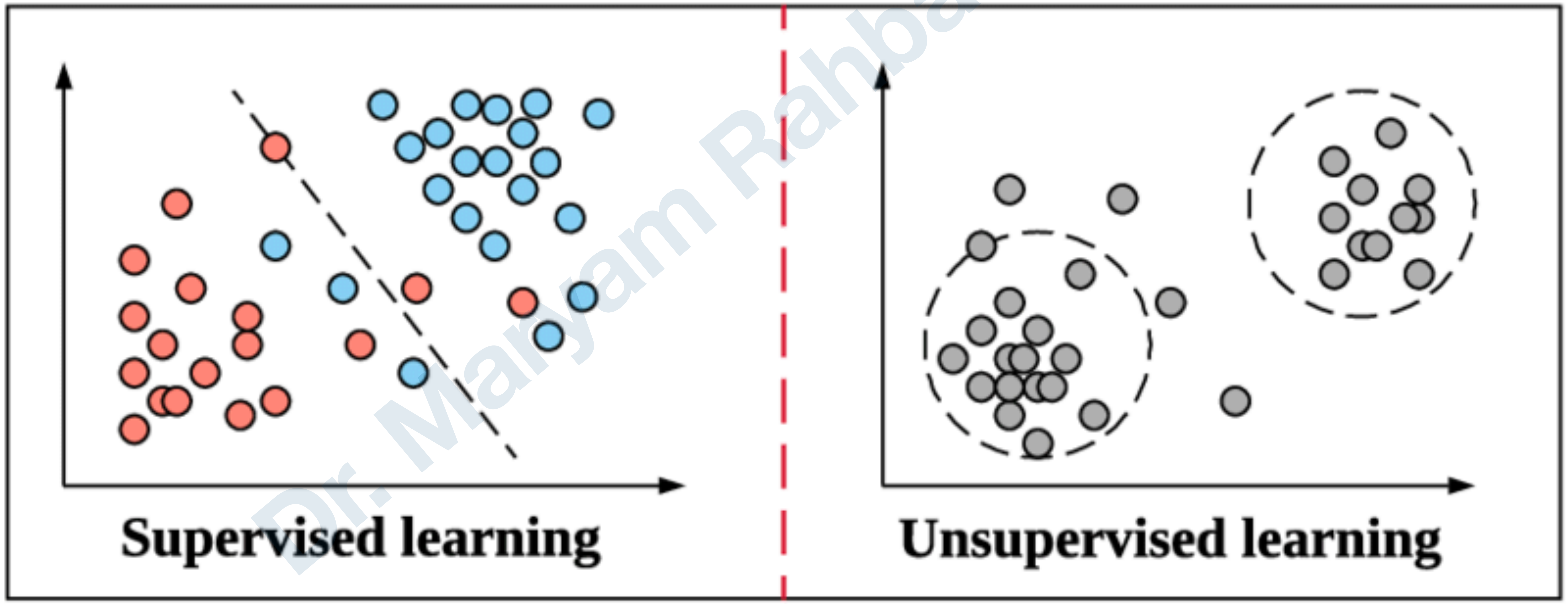
یادگیری بدون نظارت

PCA

دکتر مریم رهبر عالم

machinelearningir

Supervised learning
Unsupervised learning



Unsupervised learning

PCA Principal Component Analysis (PCA)

- GOAL: to preserve the essential parts that have more variation of the data and remove the non-essential parts with fewer variation.... but without a specific prediction task in mind
  - Dimensionality reduction
  - Visualization
  - Data Compression

features =dimensionality

WHY

Dimensionality reduction?

- Space required to store the data is reduced as the number of dimensions comes down
- Less dimensions lead to **less computation/training time**
- Some algorithms do not perform well when we have a large dimensions. So reducing these dimensions needs to happen for the algorithm to be useful
- It takes care of **multicollinearity by removing redundant features.** For example, you have two variables which are highly correlated. Hence, there is no point in storing both as just one of them does what you require

WHEN

Dimensionality reduction?

Dimensionality reduction is a **data preparation technique** performed on data prior to modeling.

It might be performed **after** data cleaning and data scaling and **before** training a predictive model.

PCA

Visualization

It is difficult to visualize the data with so many features i.e high dimensional data so we can use PCA to find the two principal components hence visualize the data in two-dimensional space with a single scatter plot and observe patterns more clearly

What is variance?

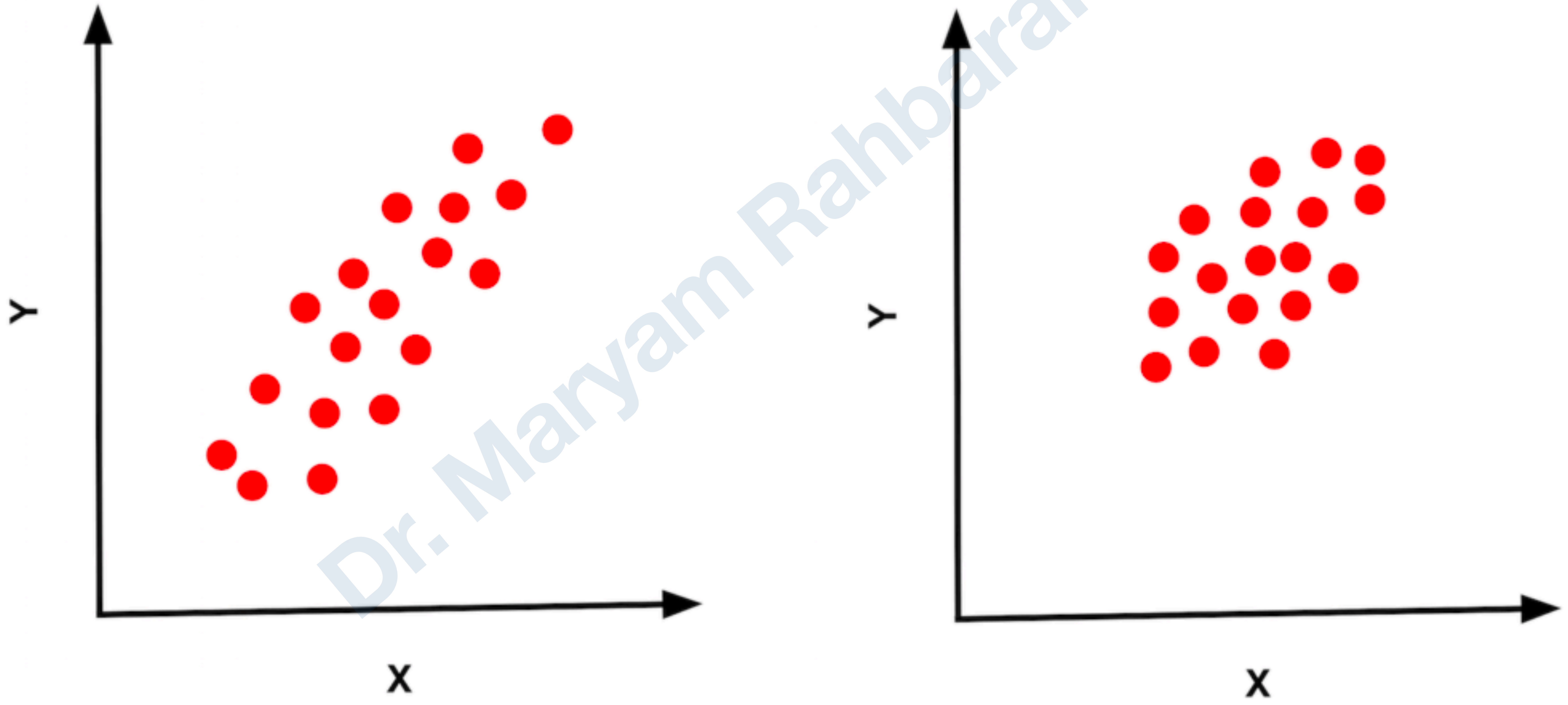
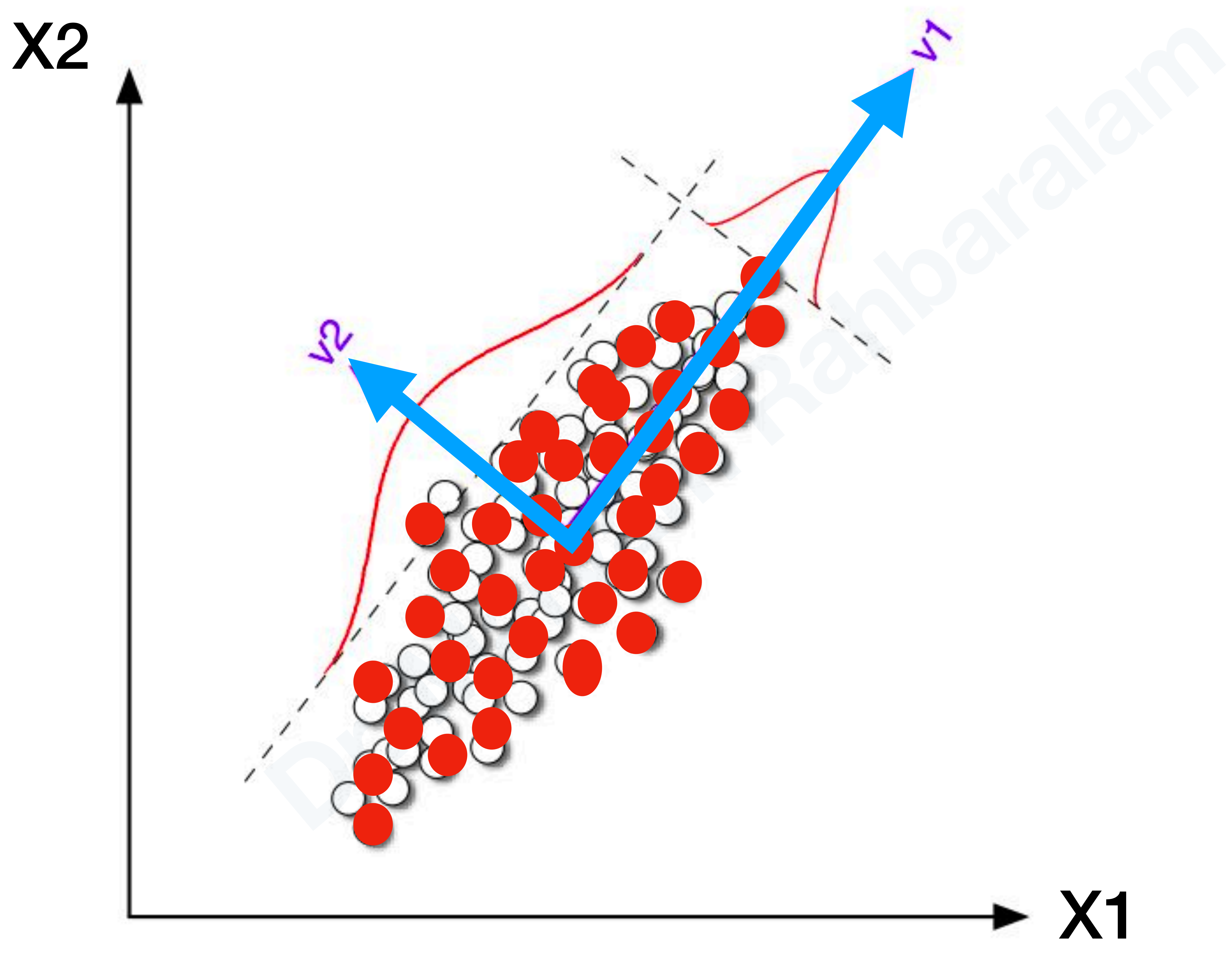


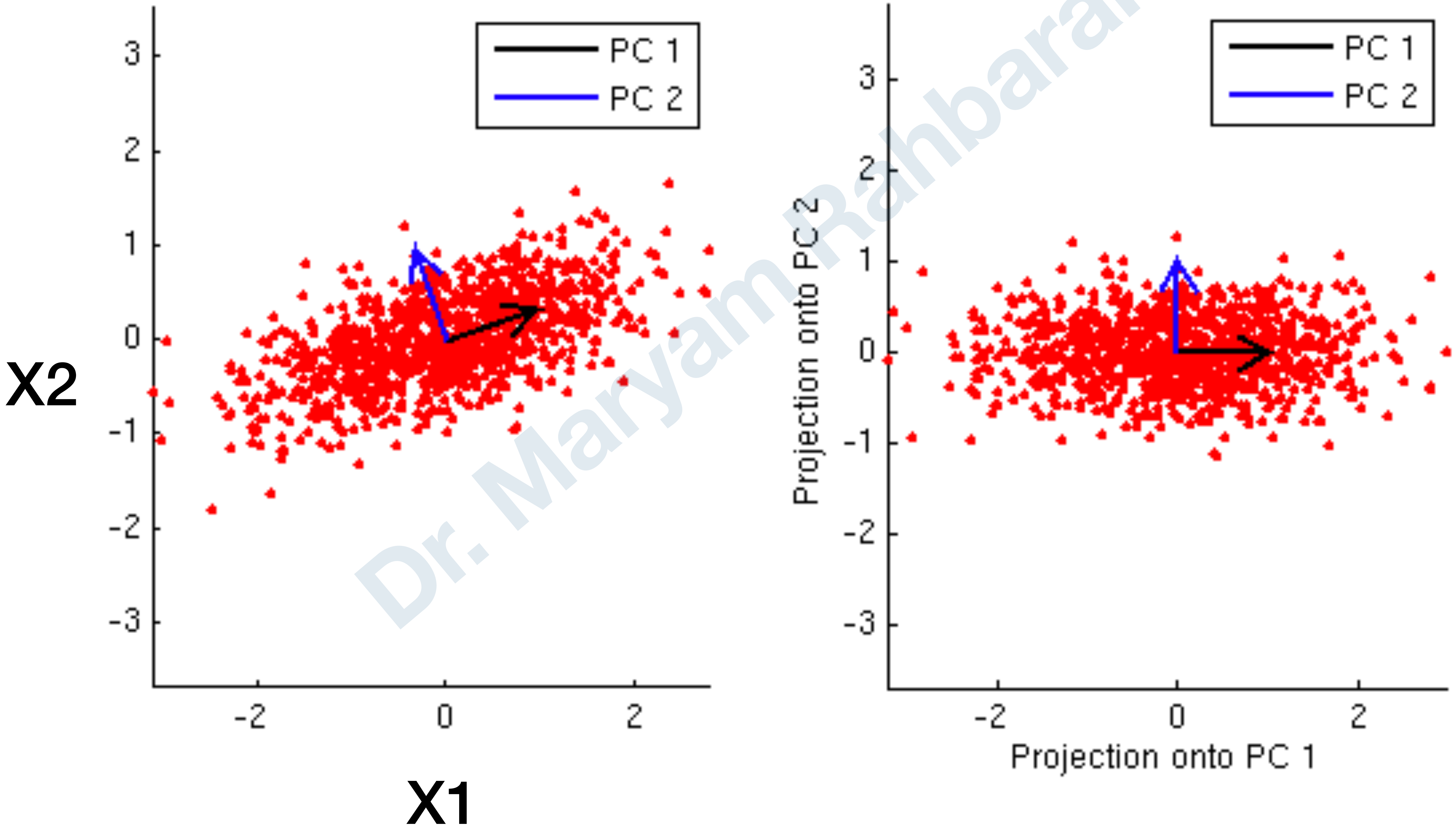
Figure 1. (a) Left : High Variance Data (b) Right : Low variance data

PCA



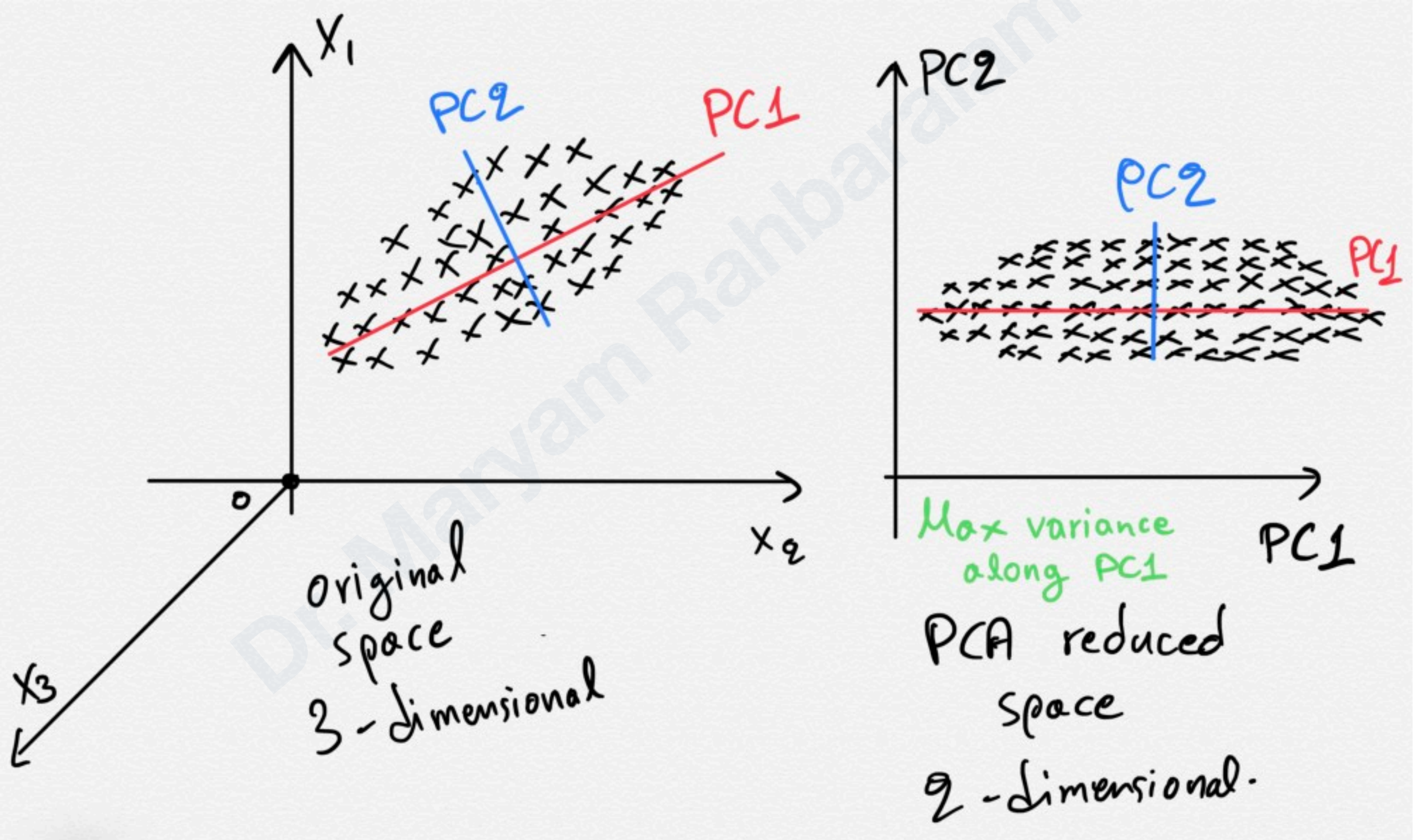
تحلیل مؤلفه‌های اصلی در تعریف ریاضی یک تبدیل خطی متعامد است که داده را به دستگاه مختصات جدید می‌برد به‌طوری‌که بزرگترین واریانس داده بر روی اولین محور مختصات، دومین بزرگترین واریانس بر روی دومین محور مختصات قرار می‌گیرد و همین‌طور برای بقیه. تحلیل مؤلفه‌های اصلی می‌تواند برای کاهش ابعاد داده مورد استفاده قرار بگیرد، به این ترتیب مؤلفه‌هایی از مجموعه داده را که بیشترین تأثیر در واریانس را دارند حفظ می‌کند

PCA



PC 1
PC 2
3
2
1
0
-1
-2
-3
-2
0
2
X2
X1
PC 1
PC 2
3
2
1
0
-1
-2
-3
-2
0
2
Projection onto PC 2
Projection onto PC 1

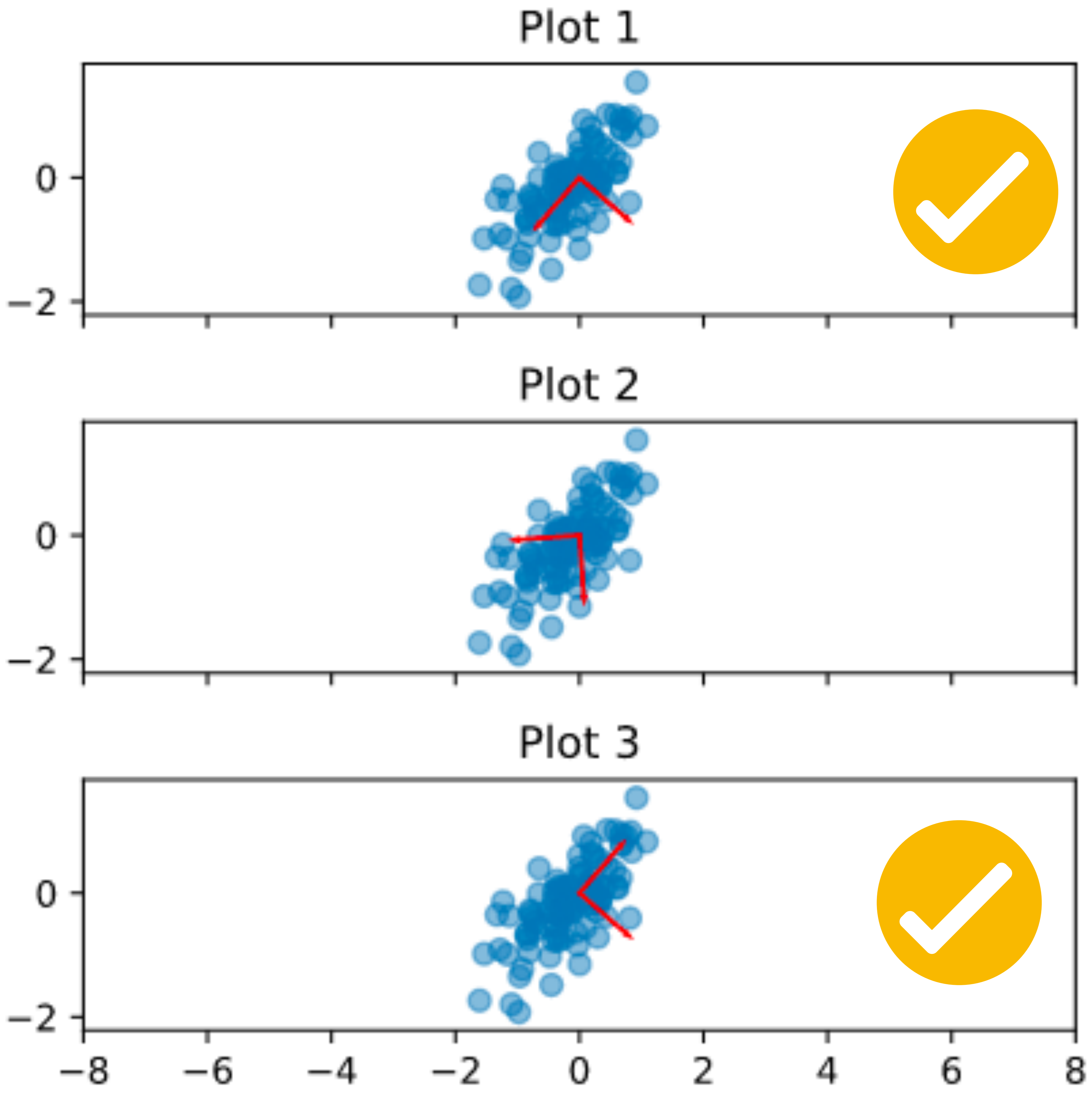
PCA


X_1
PC2
PC1
o
X_2
X_3
original space
3-dimensional
PC2
PC2
PC1
PC1
Max variance along PC1
PC1
PCA reduced space
2-dimensional.

PCA

STEP BY STEP EXPLANATION OF PCA

Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data.



PCA
STEP BY STEP EXPLANATION OF PCA

تحلیل مؤلفه‌های اصلی این تحلیل شامل تجزیه مقدارهای ویژهٔ ماتریس کواریانس می‌باشد

https://www.learnopencv.com/principal-component-analysis/

PCA

STEP BY STEP EXPLANATION OF PCA

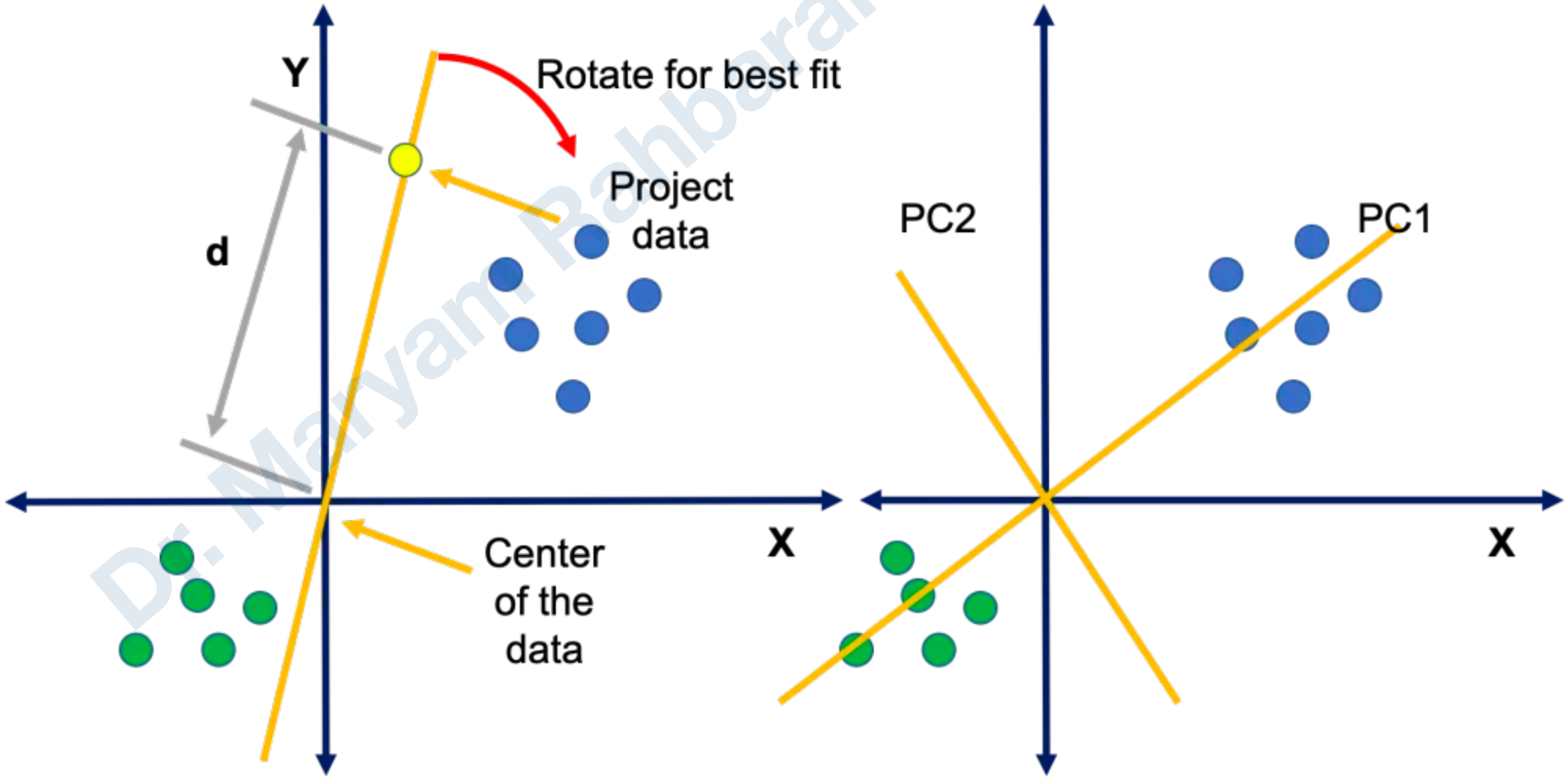
STEP 1: STANDARDIZATION

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

Standardization removes the mean and scale the data with standard deviation

$$z = \frac{value - mean}{standard\ deviation}$$

PCA


Y
Rotate for best fit
d
Project data
Center of the data
X
PC2
PC1
X

PCA

STEP 2: COVARIANCE MATRIX COMPUTATION

تحلیل مؤلفه‌های اصلی این تحلیل شامل تجزیه مقدارهای ویژهٔ ماتریس کواریانس می‌باشد

https://www.learnopencv.com/principal-component-analysis/

PCA

STEP BY STEP EXPLANATION OF PCA

STEP 2: COVARIANCE MATRIX COMPUTATION

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

$$\mathrm{A} = XX^T$$

- Need to find “first” k **eigenvectors** of *A*

PCA

STEP BY STEP EXPLANATION OF PCA

STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS

Since the covariance matrix is square, we can calculate the eigenvectors and eigenvalues for this matrix. These are rather important, as they tell us useful information about our data.

So, by this process of taking the **eigenvectors** of the covariance matrix, we have been able **to extract lines that characterise the data.**

$$\Sigma = \begin{bmatrix} \sigma_{00}^2 & \sigma_{01}^2 & \sigma_{02}^2 \\ \sigma_{10}^2 & \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{20}^2 & \sigma_{21}^2 & \sigma_{22}^2 \end{bmatrix}$$

PCA

STEP BY STEP EXPLANATION OF PCA

STEP 3: Singular Value Decomposition (SVD)

$$A = USV^T$$

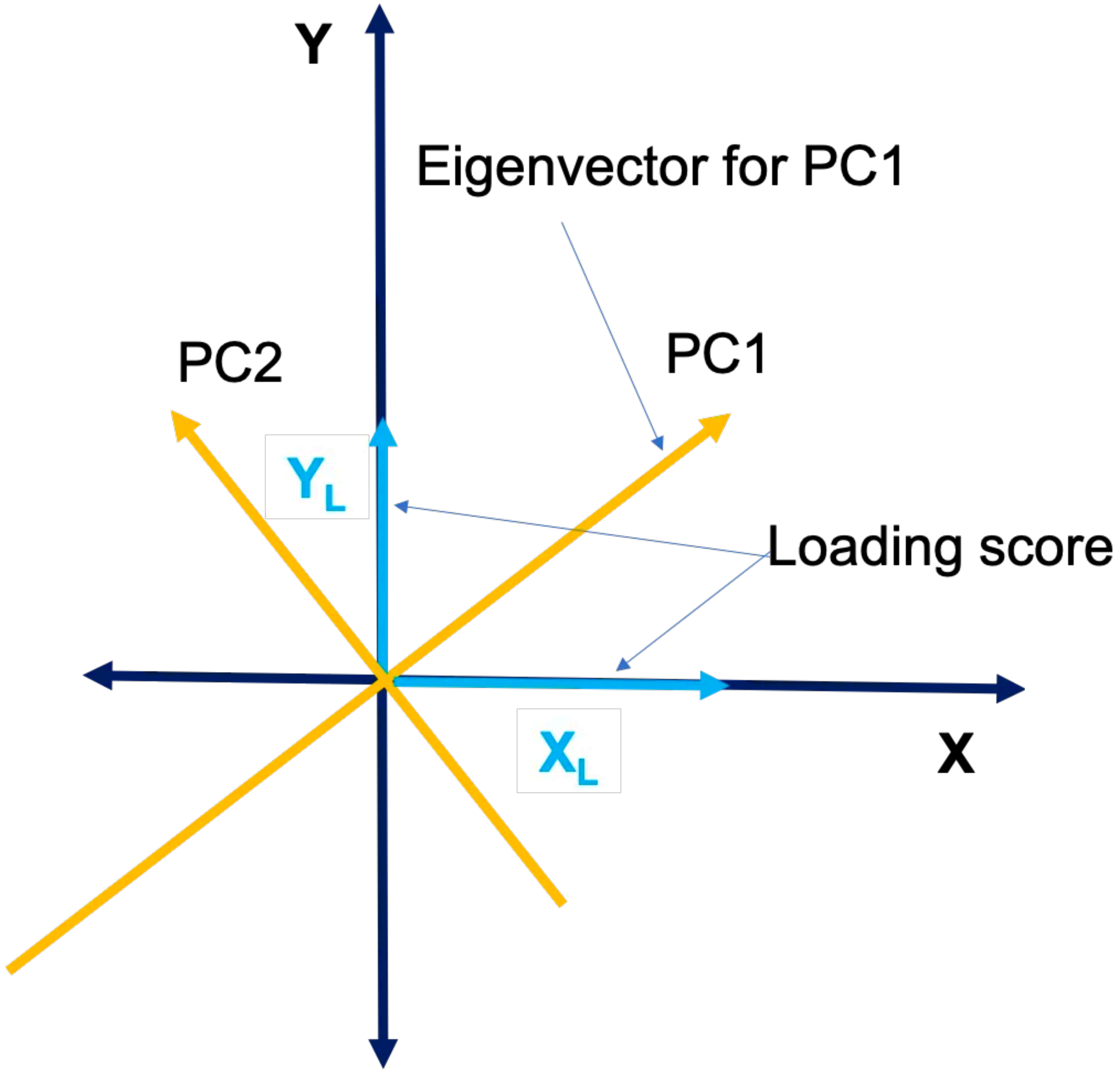
- Calculating the SVD consists of finding the eigenvalues and eigenvectors of XX^T and X^TX.
- The eigenvectors of XX^T make up the columns of U.
- The singular values are the diagonal entries of the S matrix and are arranged in descending order.

PCA

STEP BY STEP EXPLANATION OF PCA

STEP 4: : Choosing components and forming a feature vector

- In general, once eigenvectors are found from the covariance matrix, **the next step is to order them by eigenvalue, highest to lowest.**
- This gives you the components in order of significance.
- Now, if you like, you can decide to ignore the components of lesser significance.
- You do lose some information, but if the eigenvalues are small, you don't lose much.
- If you leave out some components, the final data set will have less dimensions than the original.



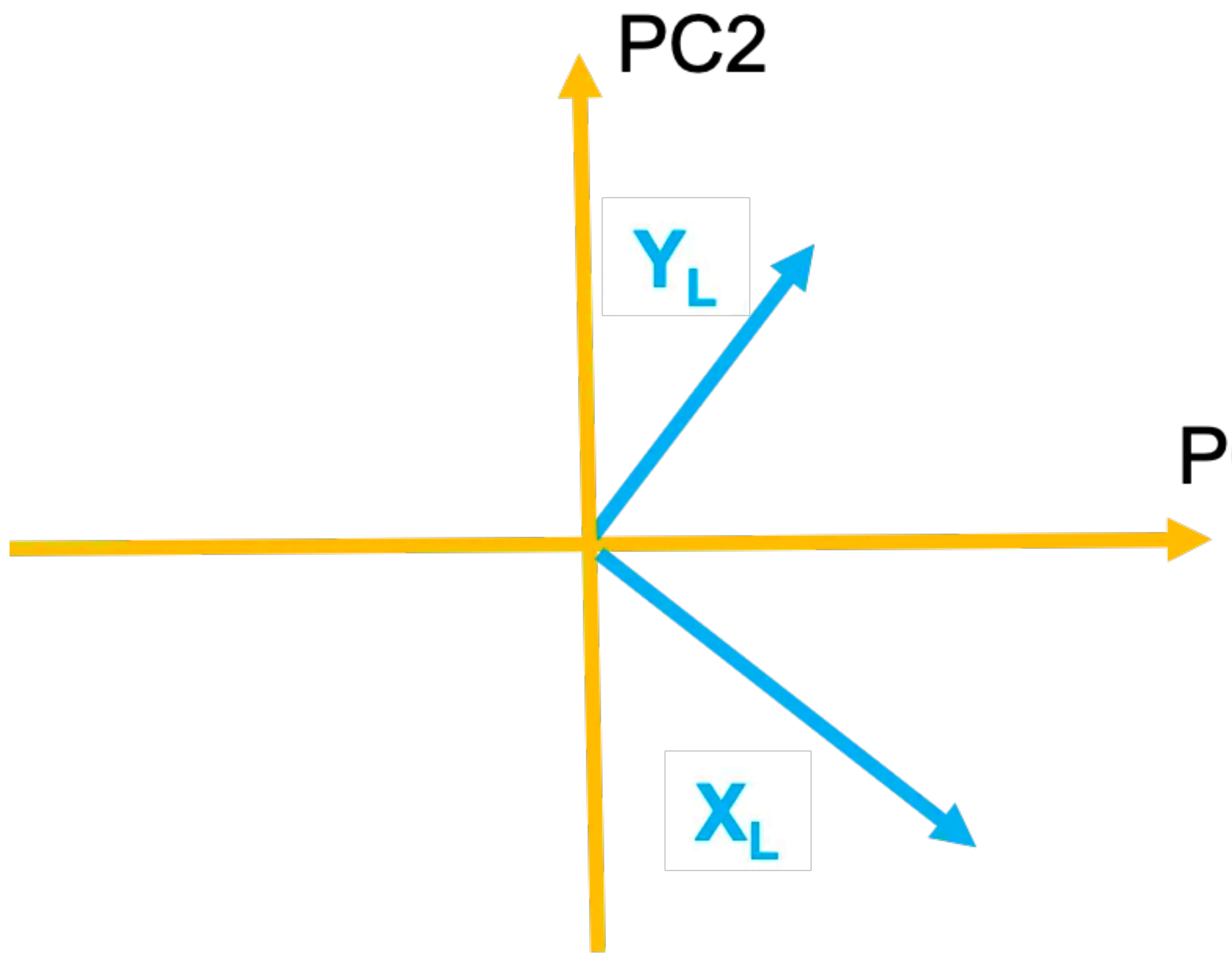
PCA

STEP BY STEP EXPLANATION OF PCA

STEP 5: Deriving the new data set

This the final step in PCA, and is also the easiest. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed.

Factor map



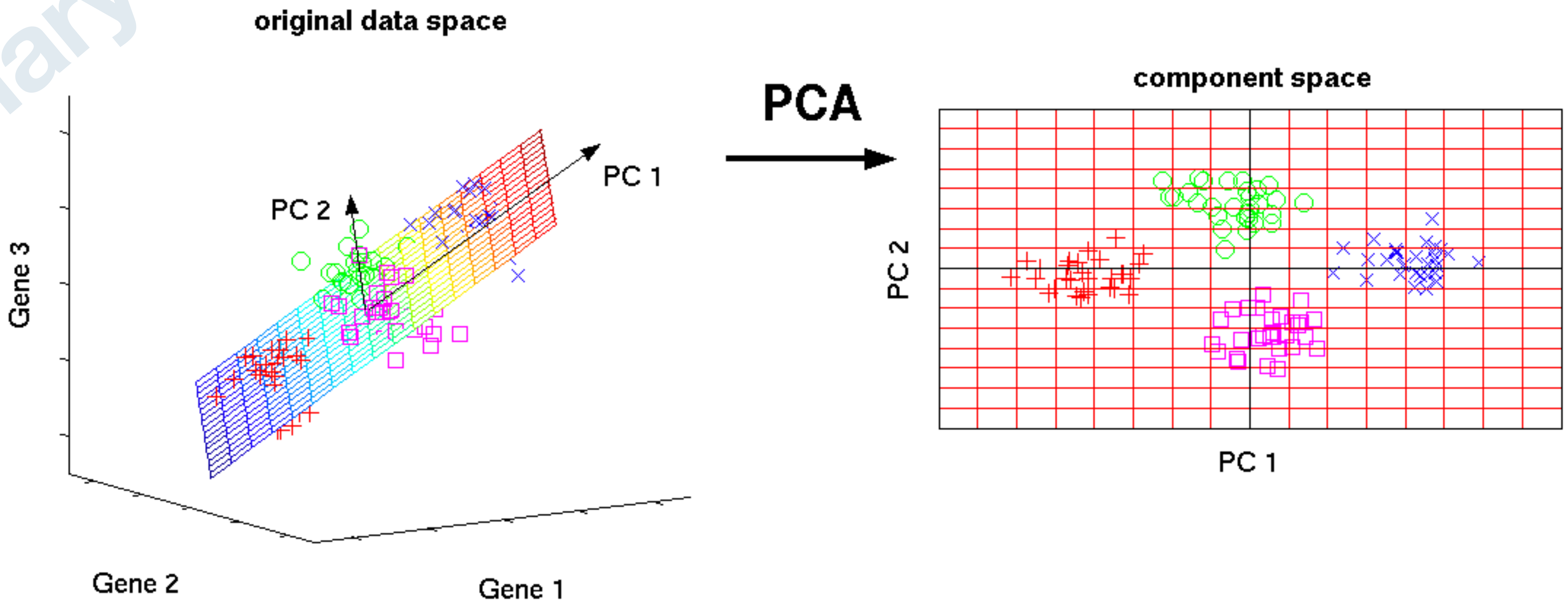
http://snowflect.com/principal-component-analysis/

PCA

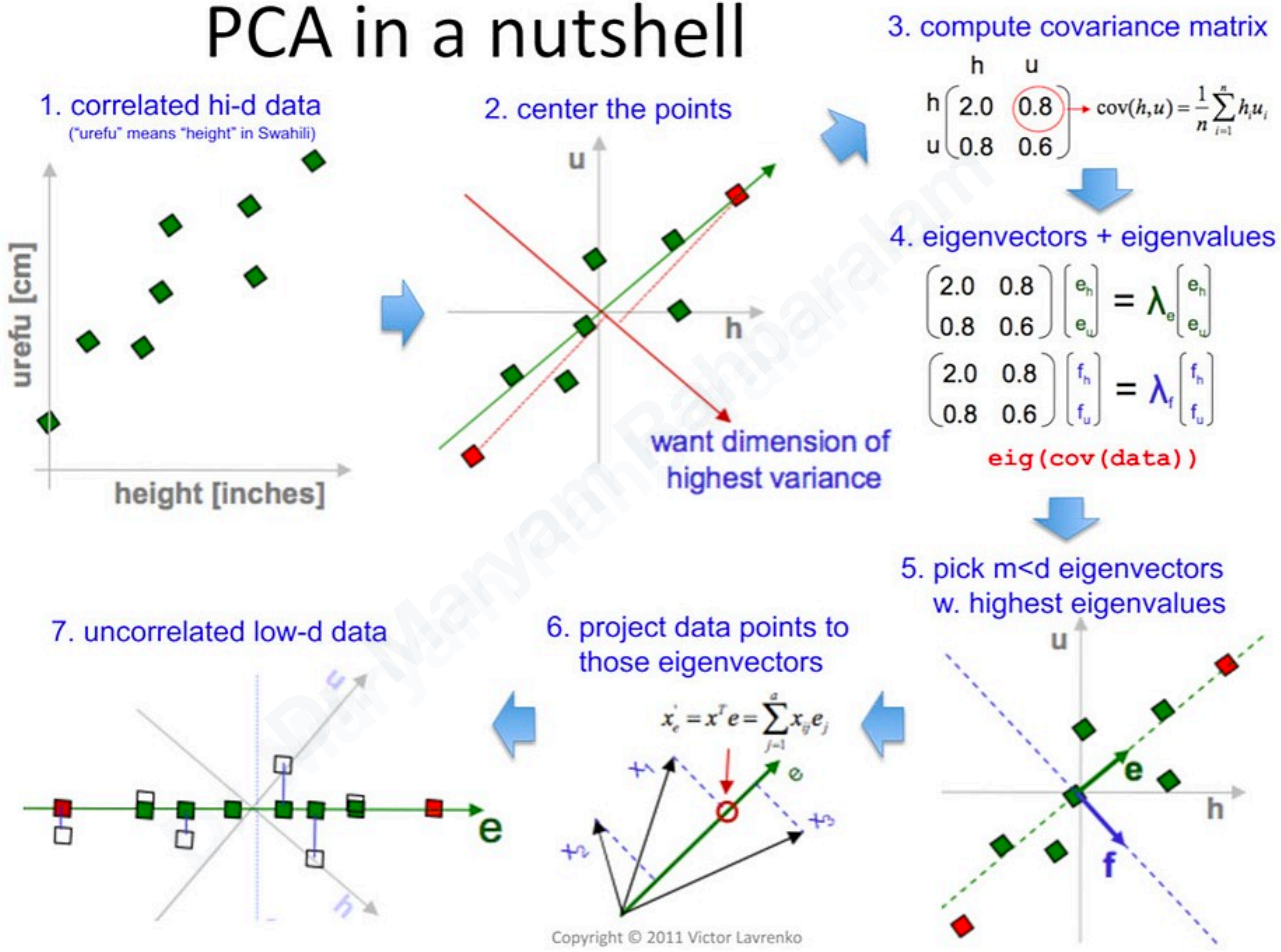
STEP BY STEP EXPLANATION OF PCA

STEP 1: So what have we done here?

Basically we have transformed our data so that is expressed in terms of the patterns between them, where the patterns are the lines that most closely describe the relationships between the data.



PCA in a nutshell



1. correlated hi-d data

("urefu" means "height" in Swahili)

2. center the points

want dimension of highest variance

3. compute covariance matrix

$$\begin{array}{cc} & \begin{matrix} h & u \end{matrix} \\ \begin{matrix} h \\ u \end{matrix} & \begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \end{array} \rightarrow \operatorname{cov}(h,u) = \frac{1}{n}\sum_{i=1}^{n} h_i u_i$$

4. eigenvectors + eigenvalues

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$$

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$$

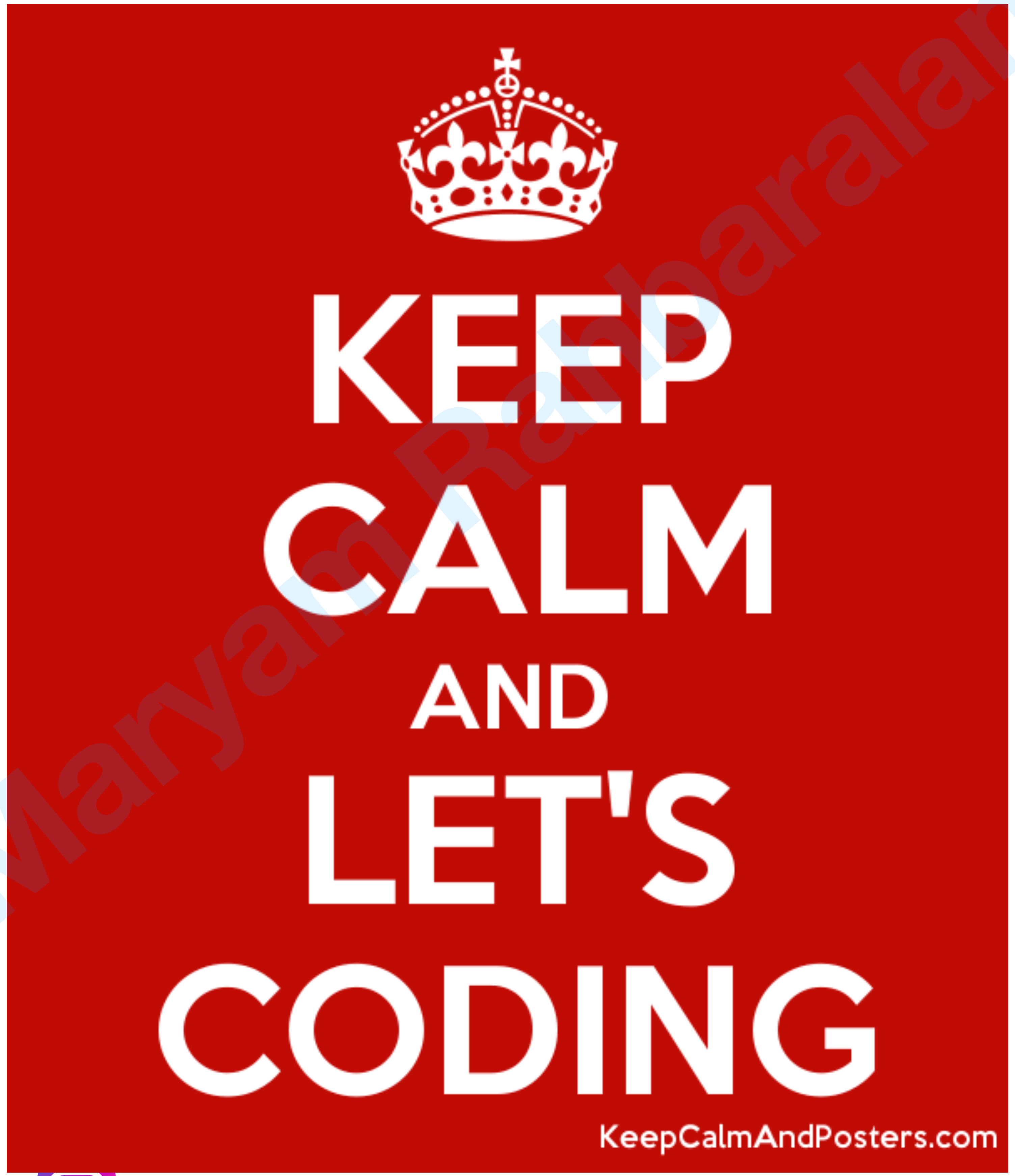
`eig(cov(data))`

5. pick m<d eigenvectors w. highest eigenvalues

6. project data points to those eigenvectors

$$x_e' = x^T e = \sum_{j=1}^{d} x_{ij} e_j$$

7. uncorrelated low-d data



@dr.maryrahbar

StandardScaler

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(df)
scaled_data = scaler.transform(df)
```

PCA in Python

Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the **SVD**.

- from **sklearn.decomposition** import **PCA**
- To create an instance of the PCA instance, use **PCA() = model**
  - **n_components:** Number of components to keep. if n_components is not set all components are kept:
- Apply the **fit_transform()** of model to pca_features
- To extract **the first principal component** of model, use **model.components_[0,:]**