

Machine Learning: Basi e Sue Applicazioni | L1

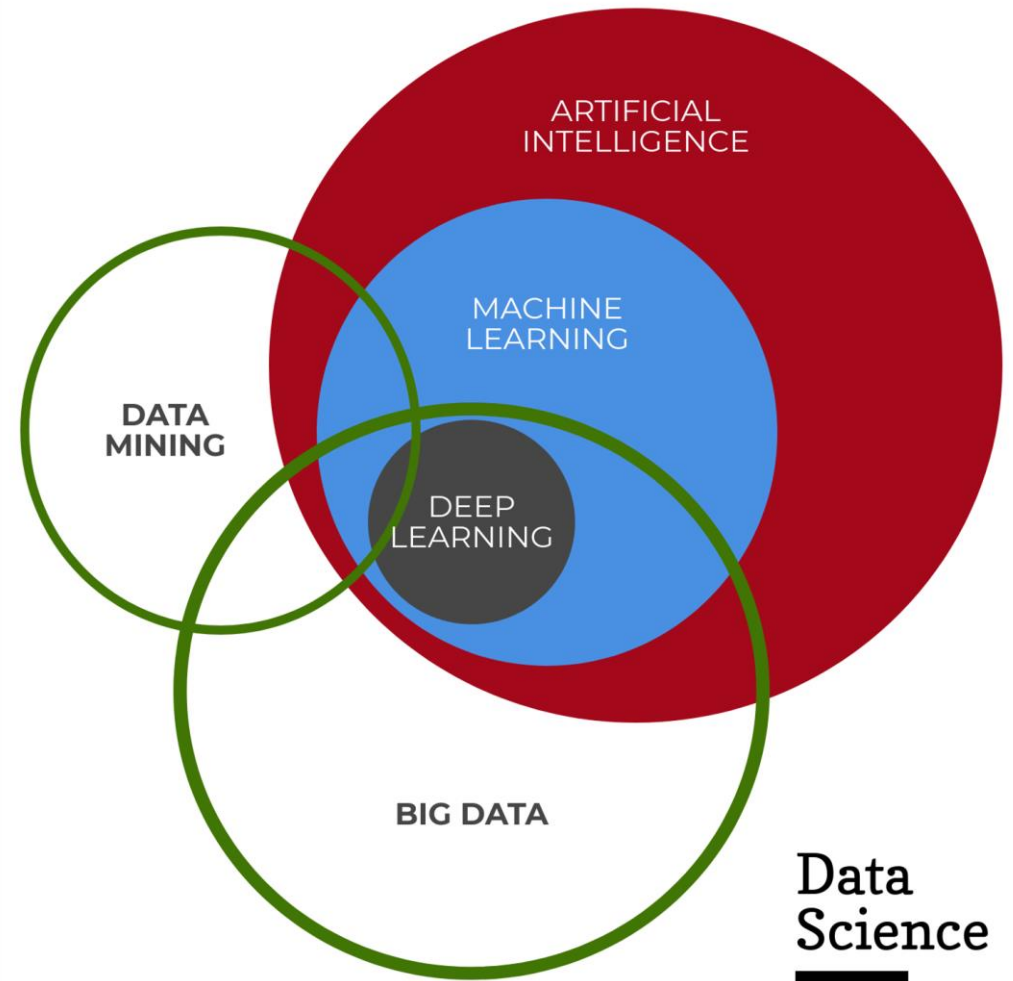
Christian Salvatore
Scuola Universitaria Superiore IUSS Pavia

2021-04-09

Machine learning

Machine learning is the subfield of **computer science** that gives **"computers the ability to learn without being explicitly programmed"**

Arthur Samuel, 1959



How is ML different from classical statistics?

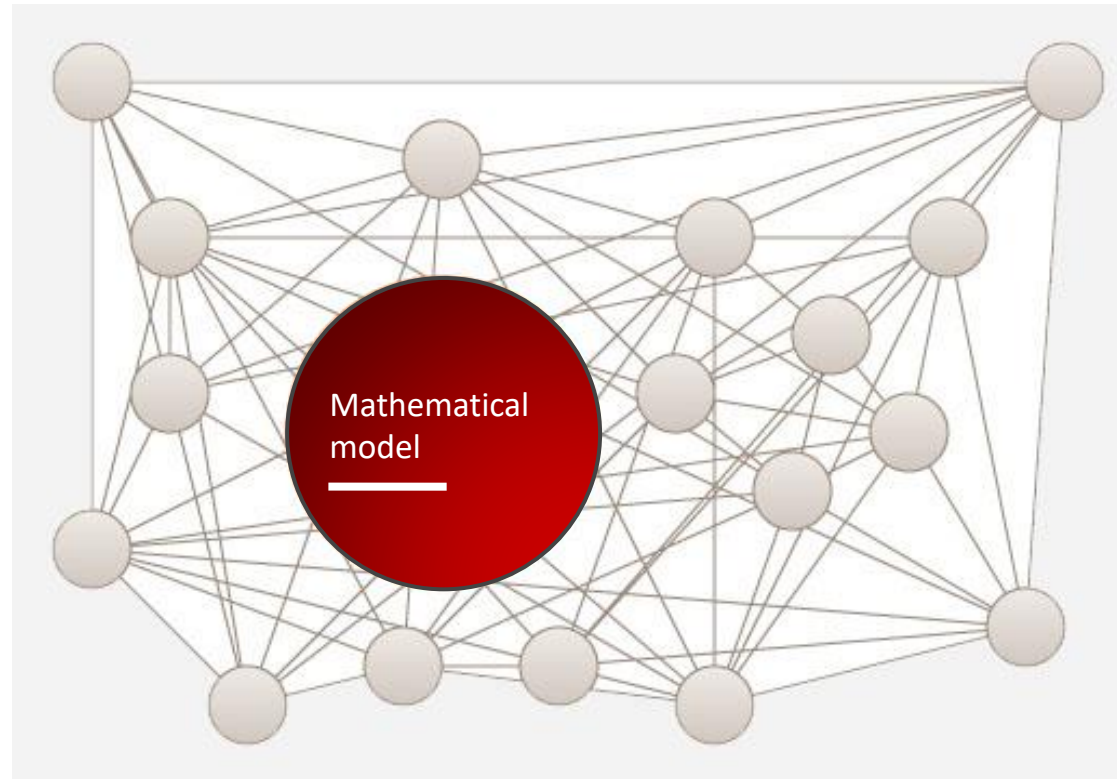
*It is focused on
classification rather than
inference*

Distribution-free approach

*High-dimensional
problems*

*Algorithmic and
computational aspects play a
central role*

Machine learning



Designing mathematical models able to

Machine learning



1. identify hidden patterns in data
2. handle and summarize a great quantity of data into a model
3. use that model to perform automatic classification

Machine learning

Training data \longrightarrow Target variable

From the trend
registered during
the last years

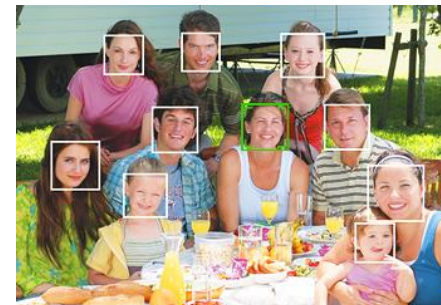
PREDICT

Future market performance

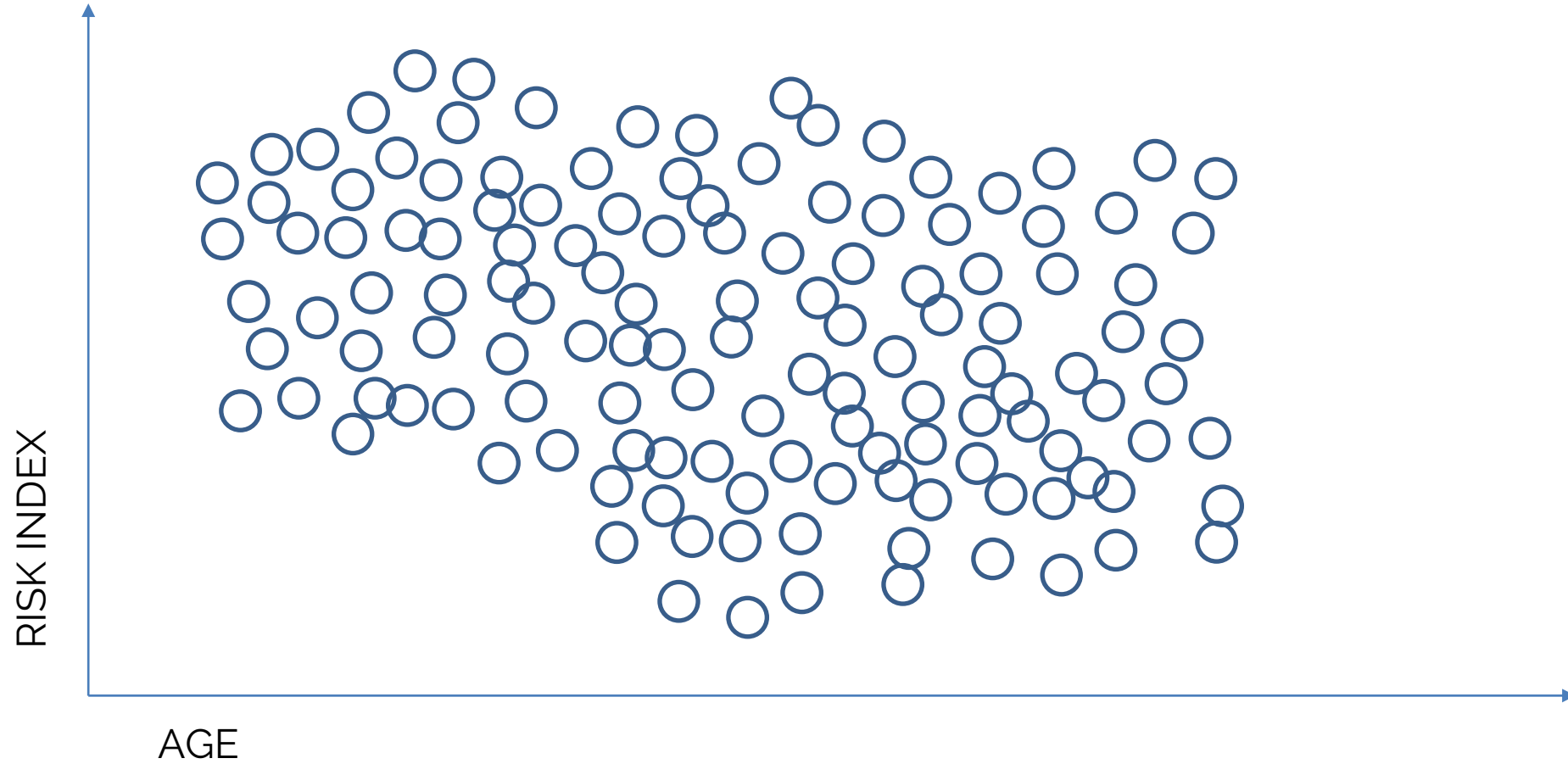
From a set
of images

DETECT

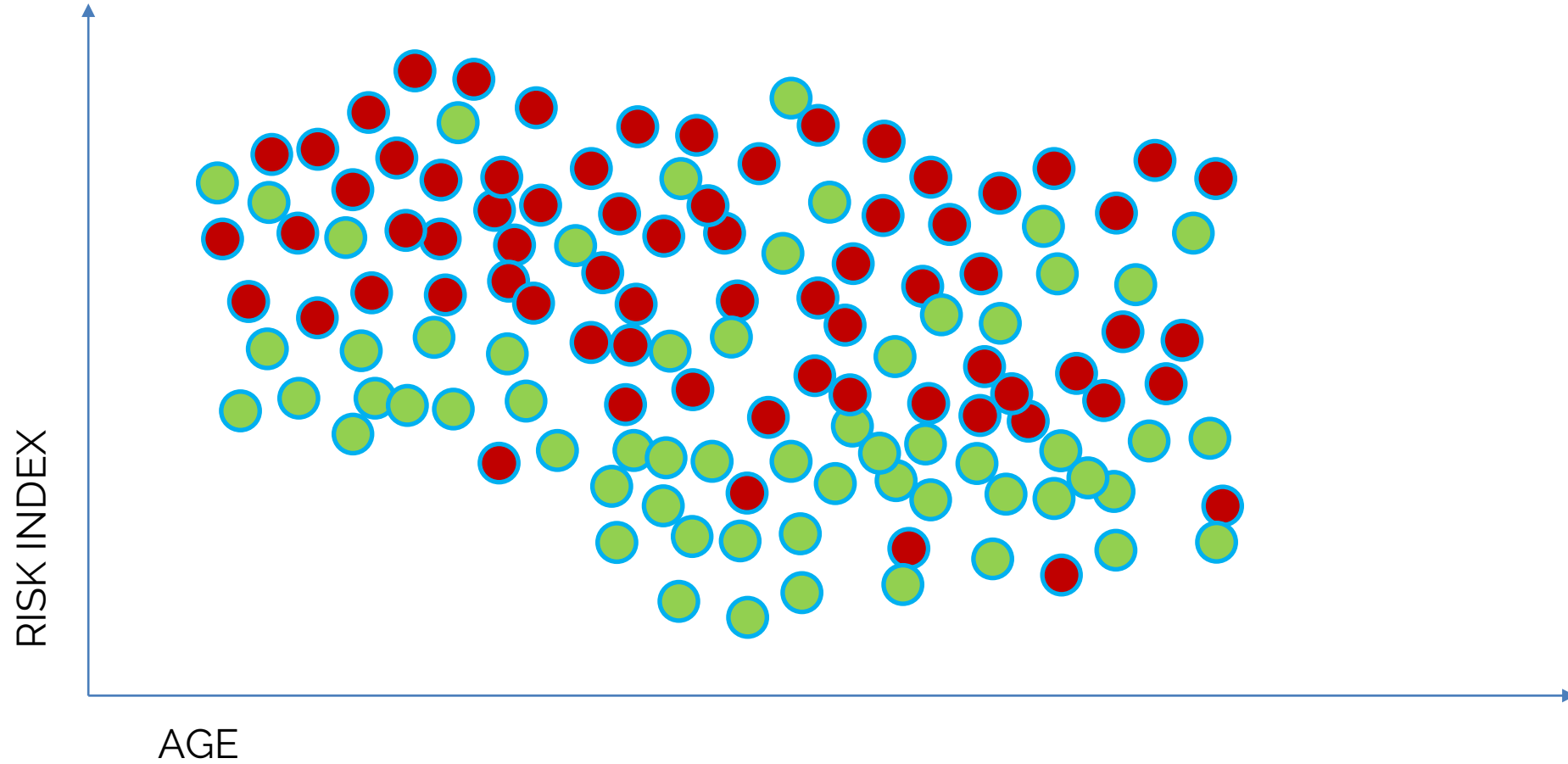
Faces (in new images)



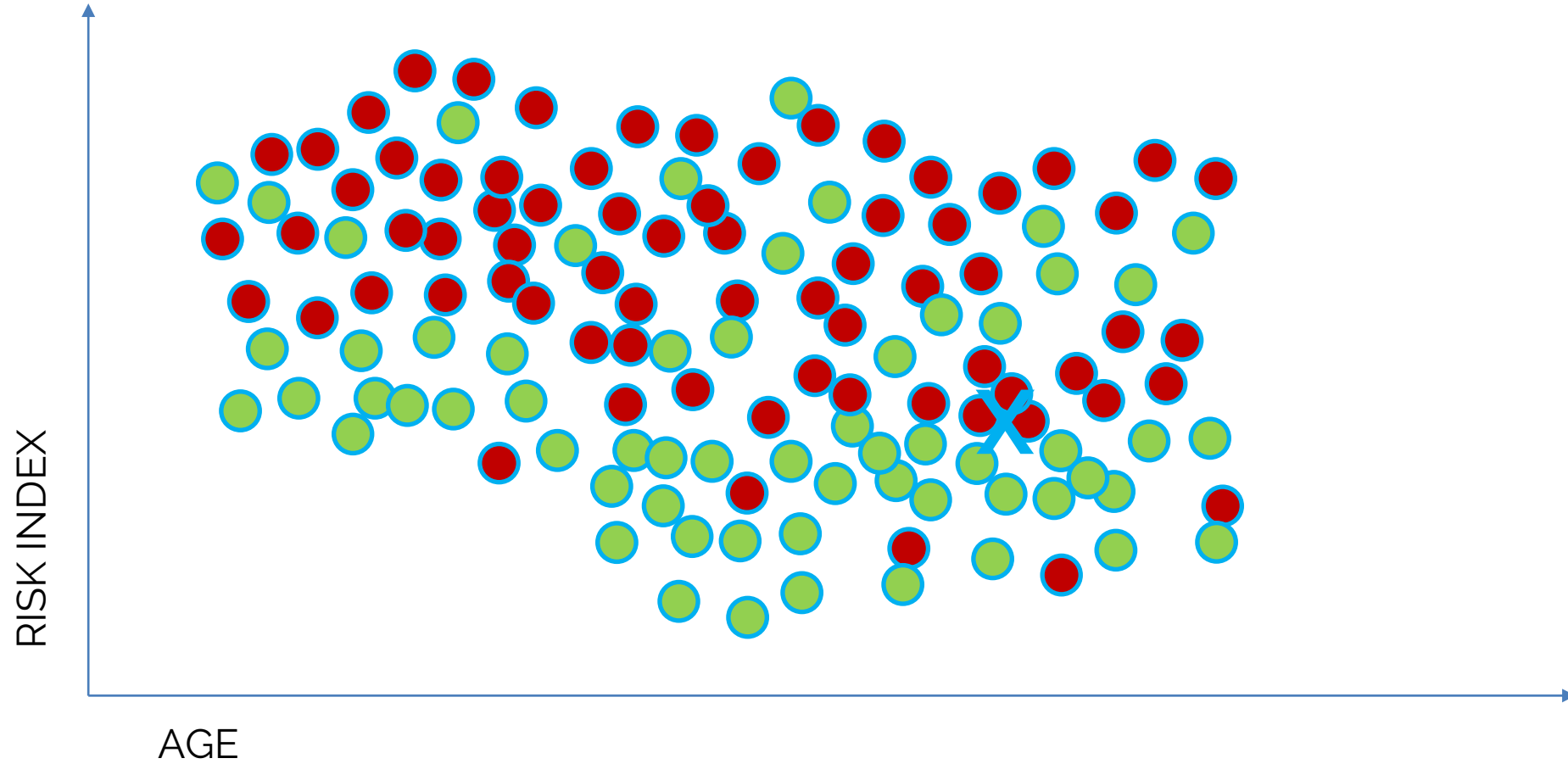
Classification



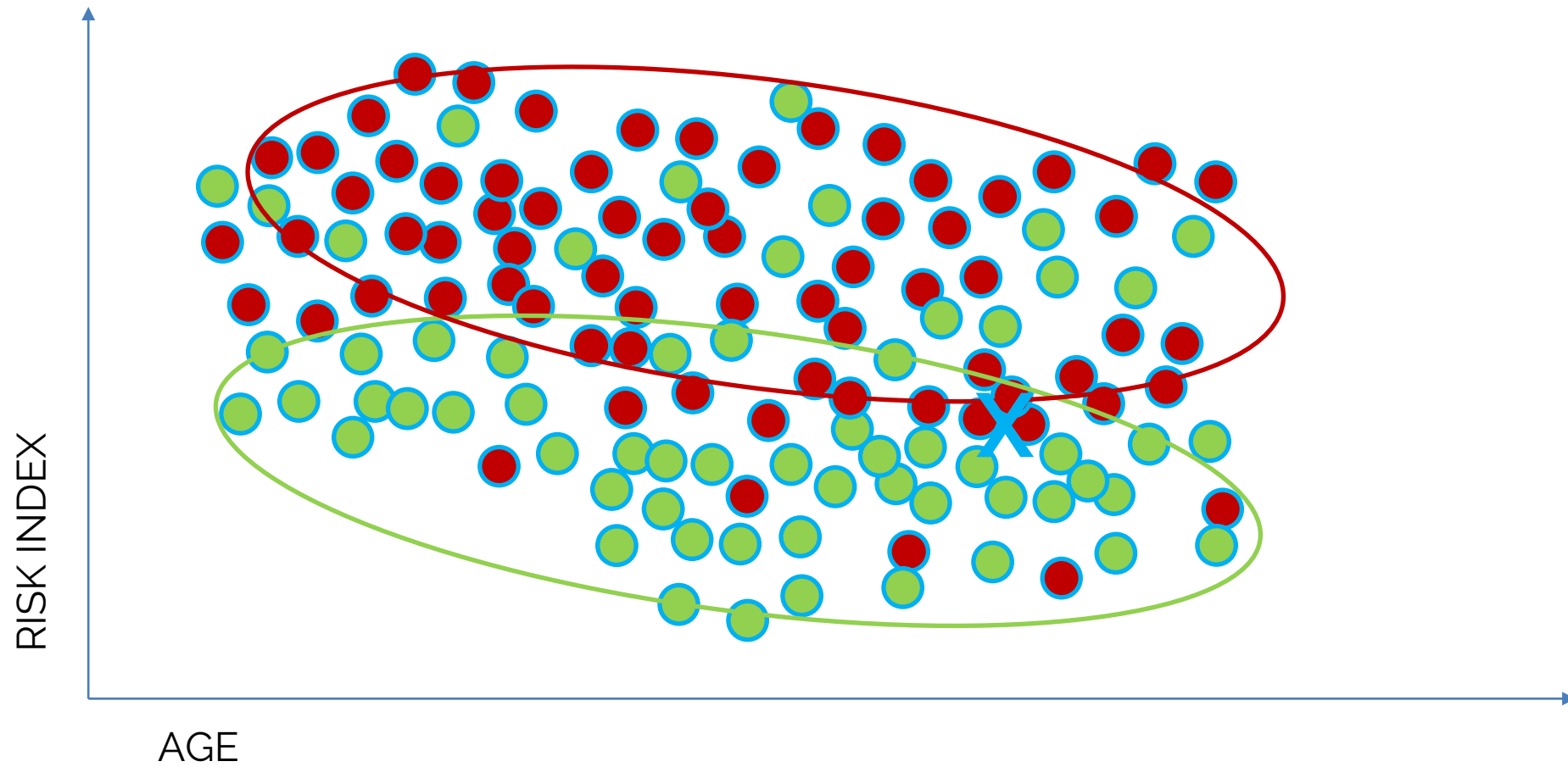
Classification



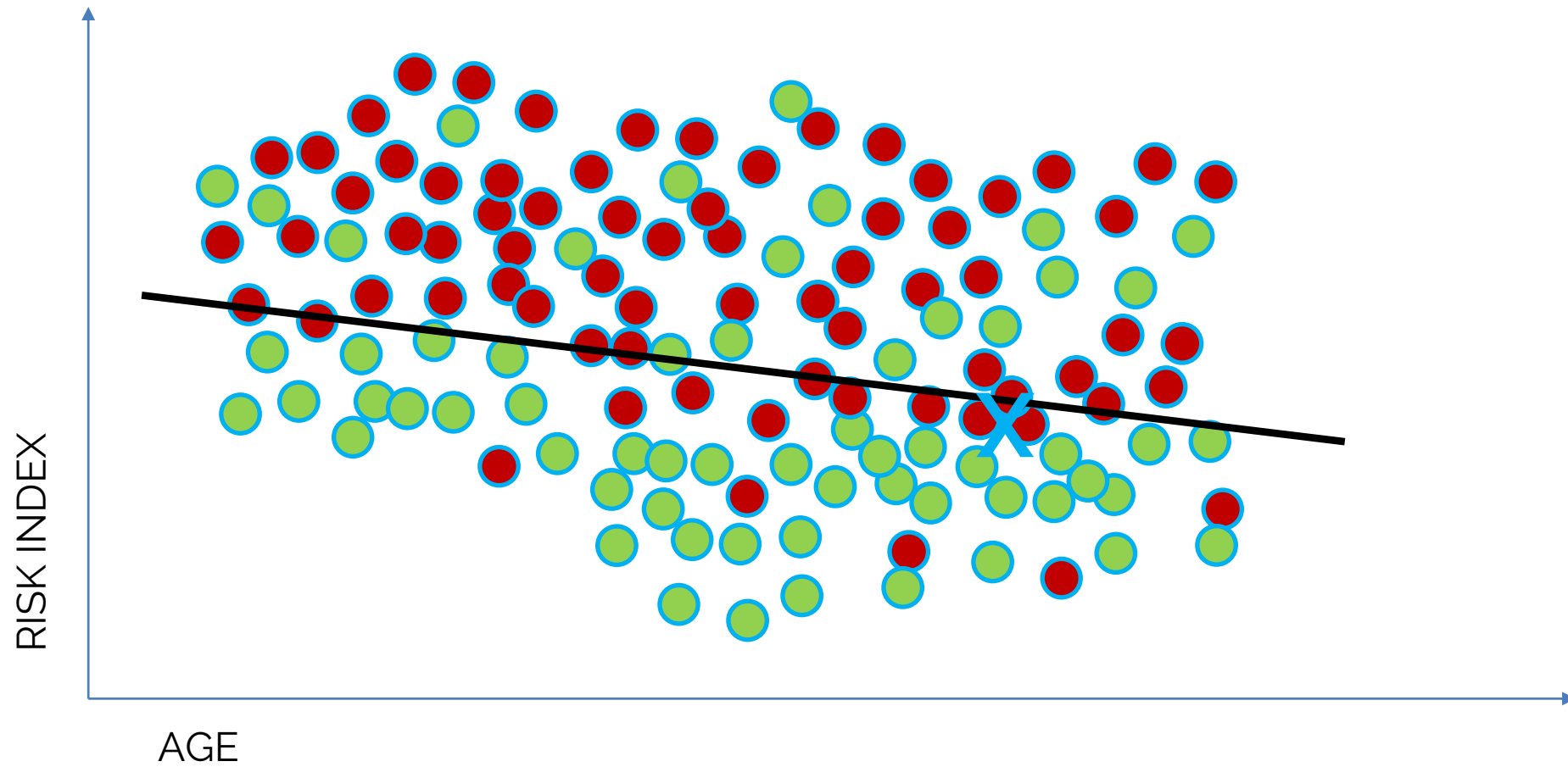
Classification



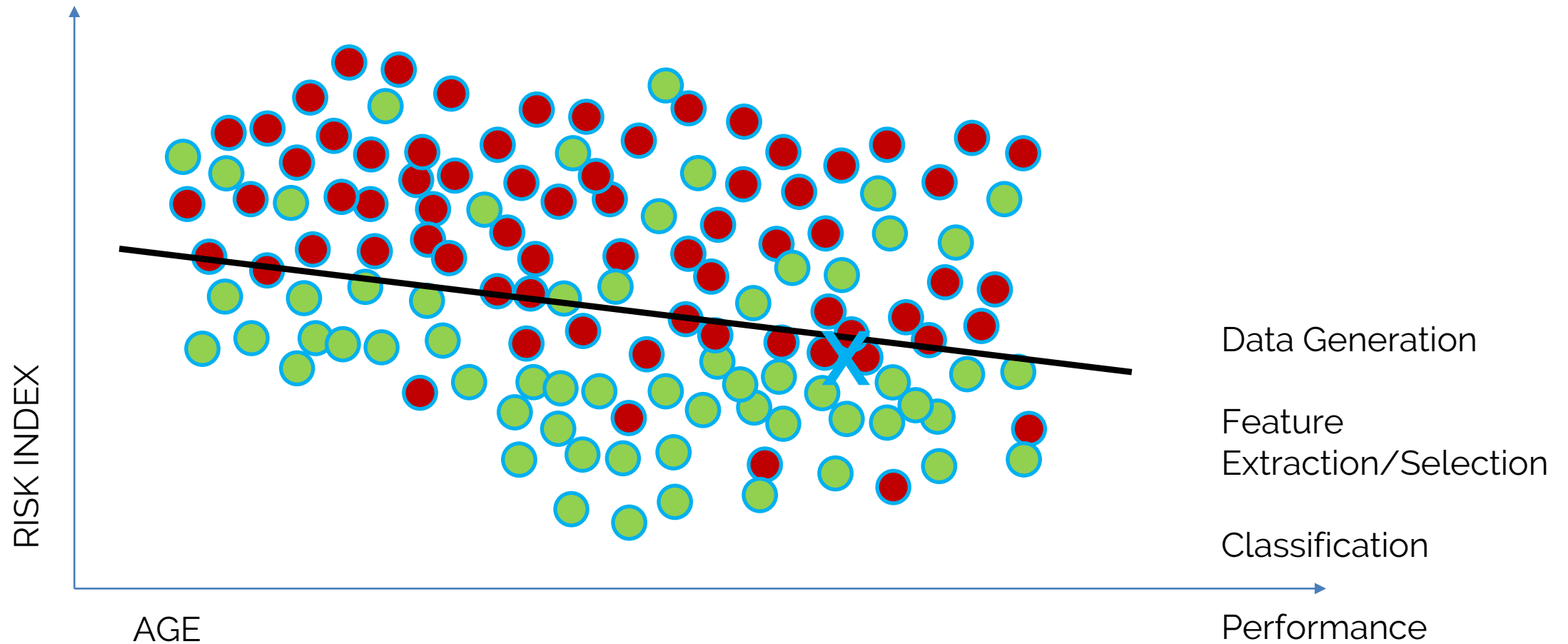
Classification



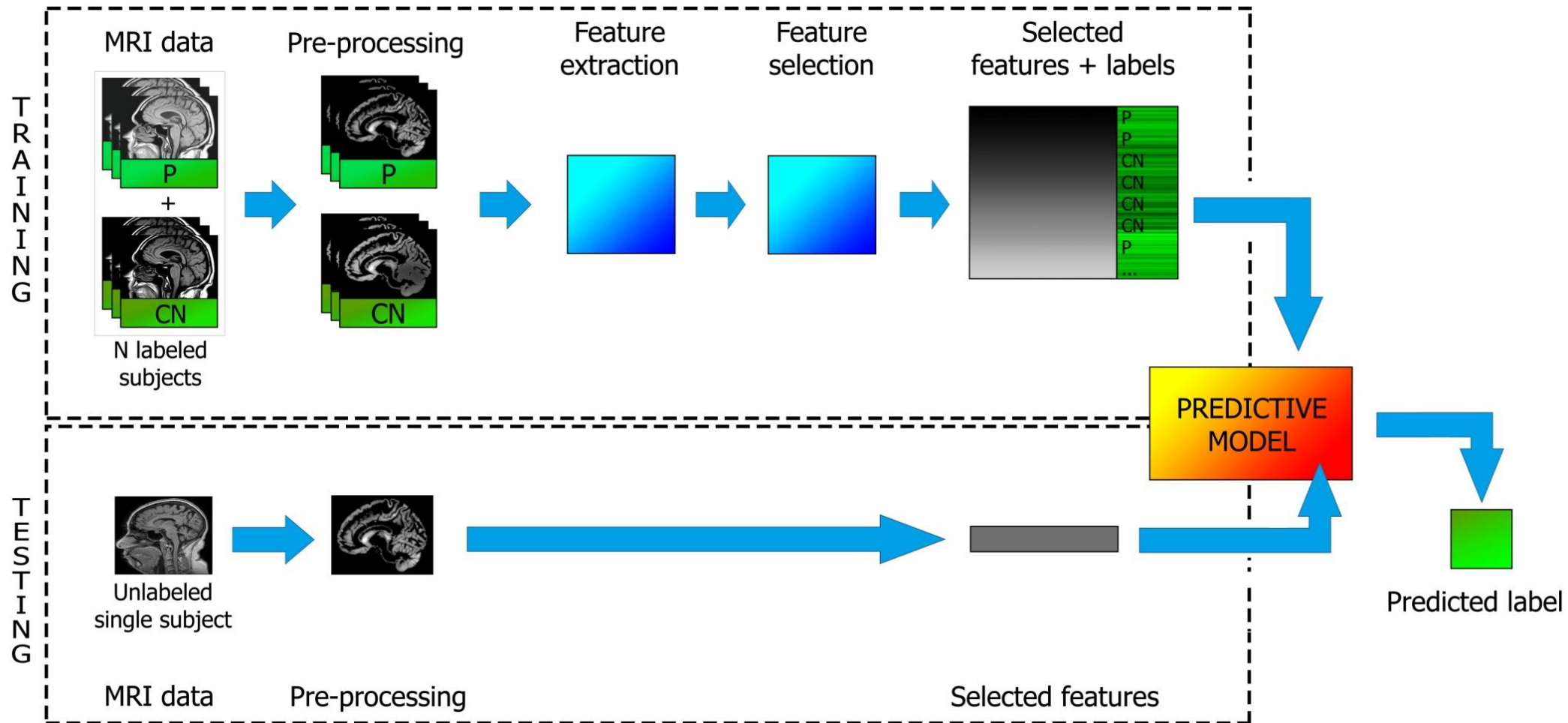
Classification



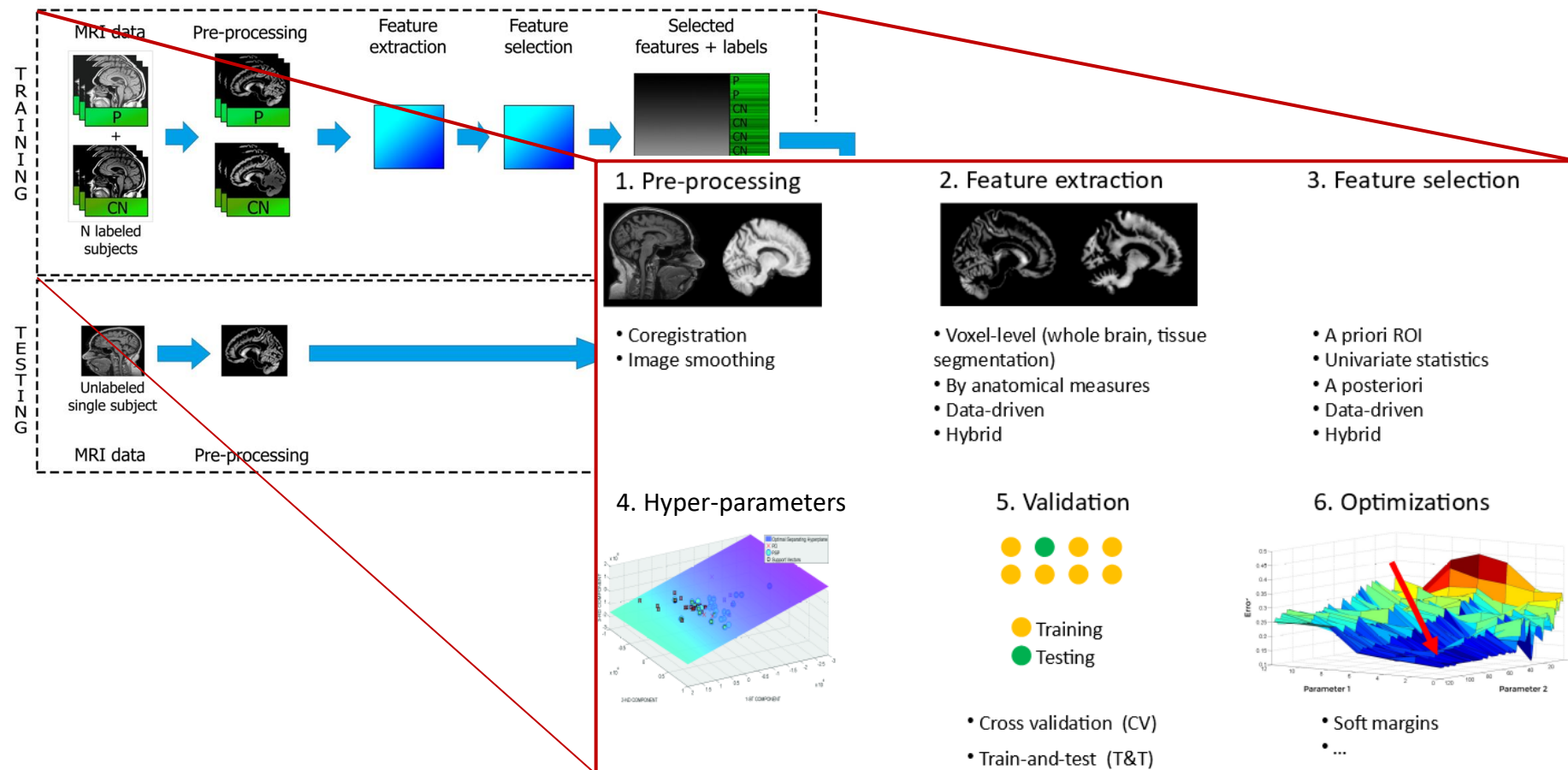
Classification



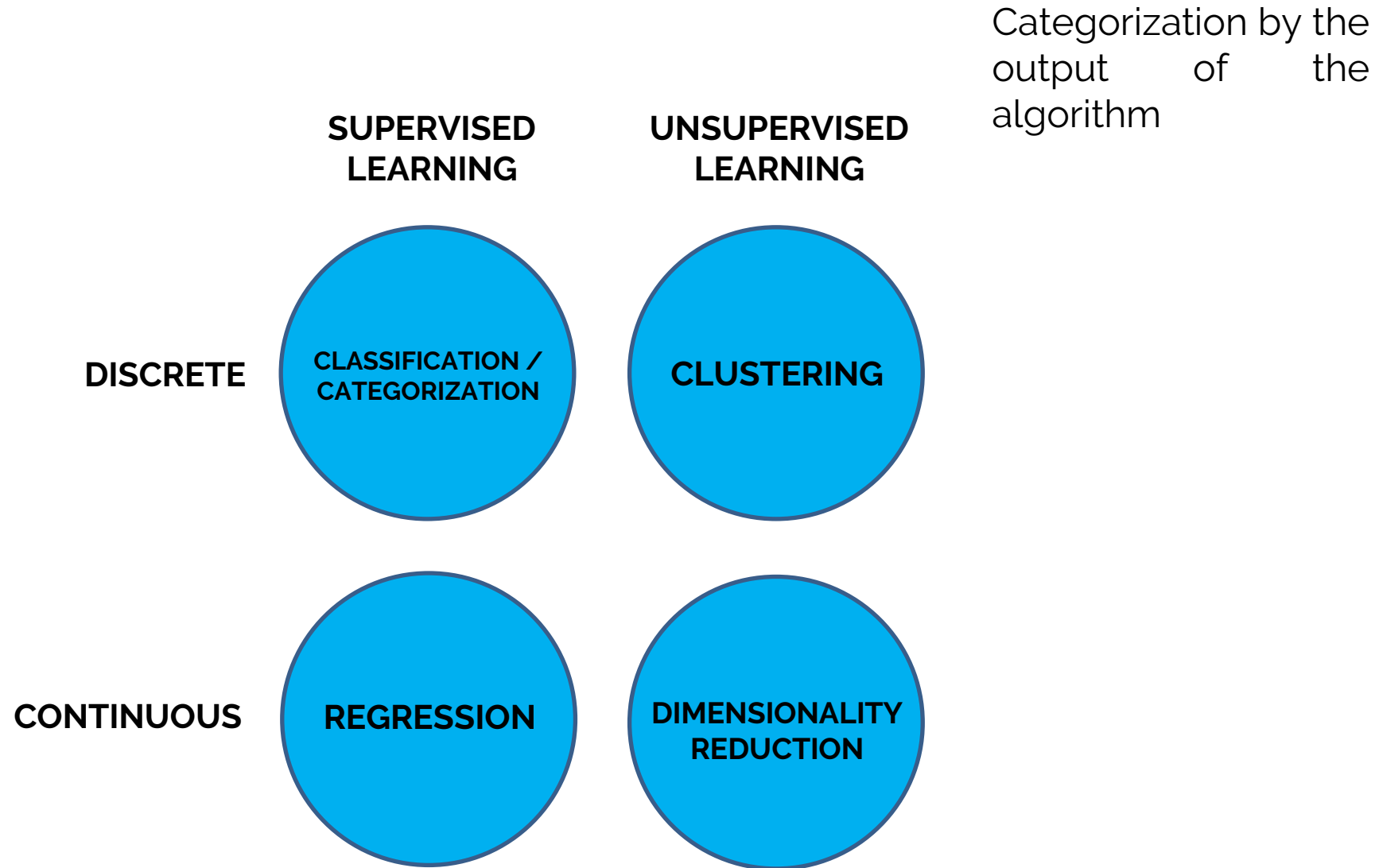
Machine Learning Applied to Medical Data



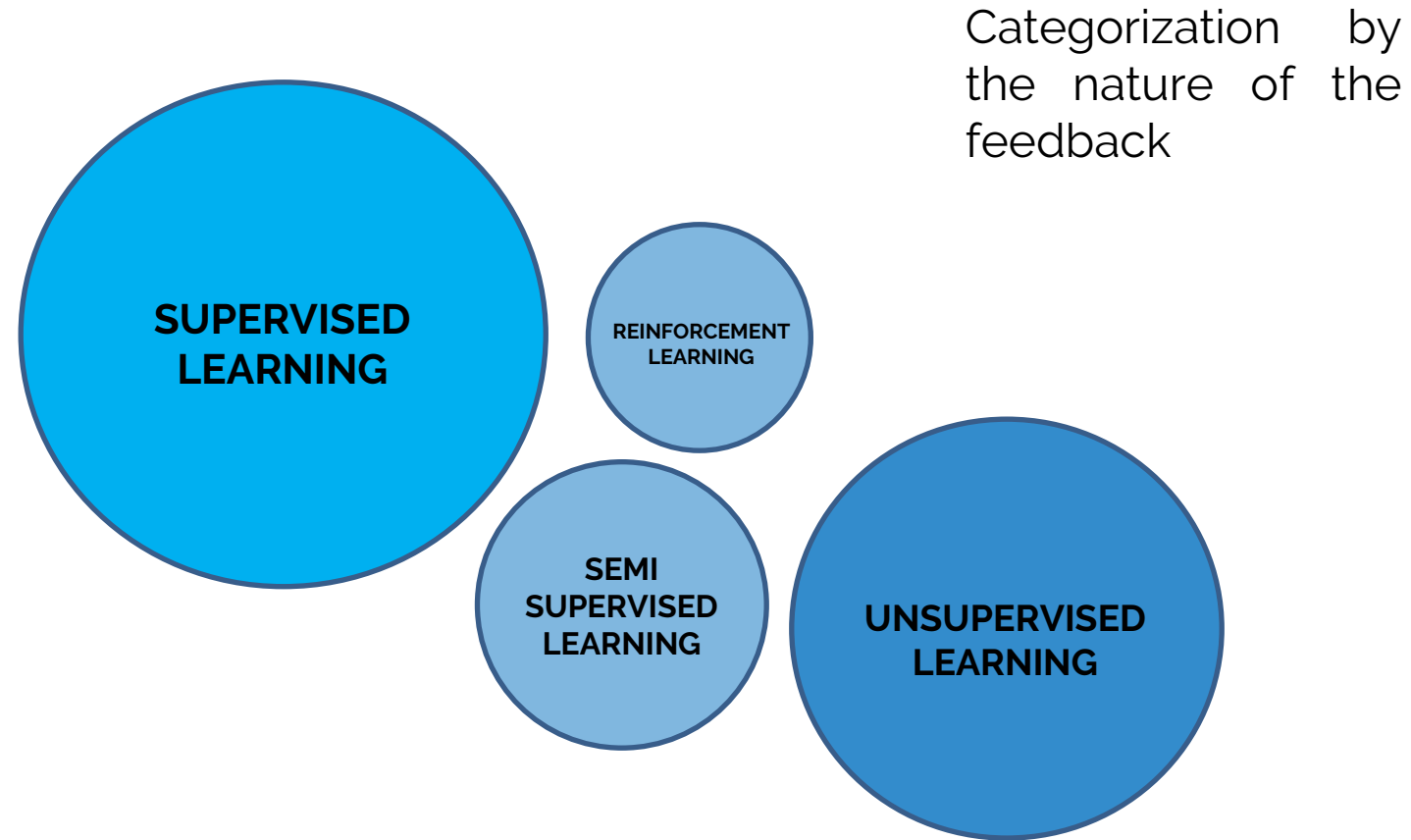
Machine Learning Applied to Medical Data



Machine learning



Machine learning



FEATURE EXTRACTION

Feature Extraction | Obiettivo

Eseguire un mapping dallo spazio iniziale a uno spazio di dimensione inferiore R^k , con $k < d$

Operare in spazi a dimensionalità inferiore

- rende più semplice addestrare algoritmi di machine learning (richiede meno dati per l'addestramento)
- scartando dati ridondanti (informazioni correlate) e rumorosi si migliorano anche le prestazioni e le si rendono più robuste

Obiettivo è scartare le informazioni non rilevanti o meno rilevanti per il problema di interesse

Ridurre la dimensionalità non significa ridurre alcune dimensioni e salvarne altre, ma combinare le dimensioni in modo opportuno.

Analisi delle Componenti Principali - Principal Component Analysis (PCA)
- trasformazione non-supervisionata che esegue un mapping lineare delle dimensioni con l'obiettivo di preservare al massimo l'informazione dei pattern (nota anche come Karhunen-Loeve –KL– transform)

(FEATURE SELECTION)

Analisi delle Discriminanti Lineari, Linear Discriminant Analysis (LDA)
- trasformazione supervisionata di mapping lineare

Feature Extraction | Principal Components Analysis

Dato un training set $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1 \dots n$, siano

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1 \dots n} \mathbf{x}_i$$

il vettore medio $\in \mathbb{R}^d$

$$\mathbf{\Sigma} = \frac{1}{n-1} \sum_{i=1 \dots n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^t$$

la matrice di covarianza $\in \mathbb{R}^{d \times d}$

allora per un dato k ($k < d$, $k < n$, $k > 0$), lo spazio k dimensionale $(S_{\mathbf{x}}, \Phi_k)$

è univocamente definito dal vettore medio e dalla matrice di proiezione $\Phi_k \in \mathbb{R}^{d \times k}$

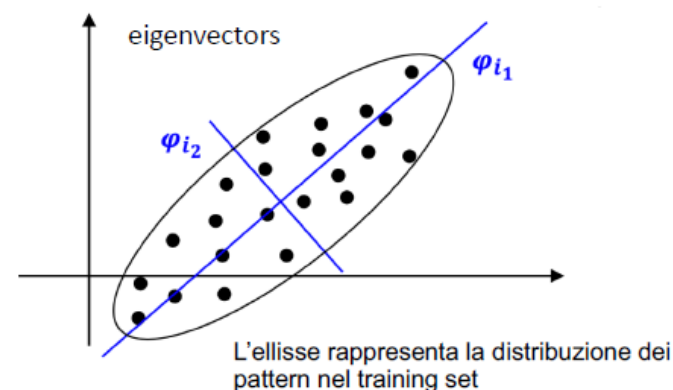
le cui colonne sono costituite dagli autovettori di $\mathbf{\Sigma}$ corrispondenti ai k più grandi autovalori

$$\Phi_k = [\boldsymbol{\varphi}_{i_1}, \boldsymbol{\varphi}_{i_2} \dots \boldsymbol{\varphi}_{i_k}] \text{ con } \lambda_{i_1} \geq \lambda_{i_2} \geq \dots \lambda_{i_k} \geq \dots \lambda_{i_d}$$

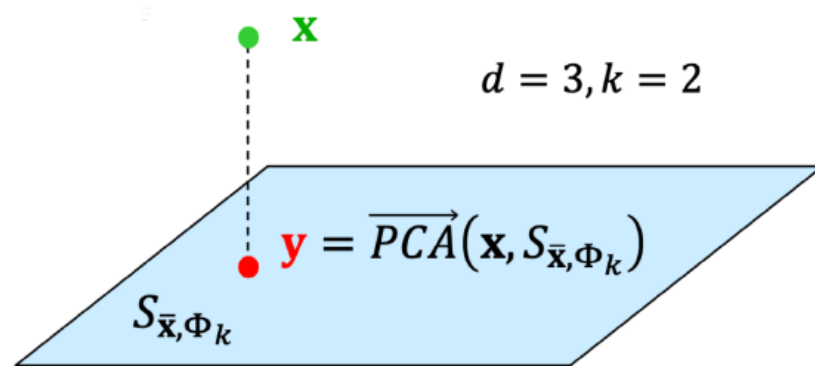
$\boldsymbol{\varphi}_{ir}$ autovettore di $\mathbf{\Sigma}$ corrispondente all'autovalore λ_{ir} $r=1 \dots d$

I primi k autovettori sono detti *componenti principali (PC)*

$\boldsymbol{\varphi}_{i_1}$ indica la direzione di maggior varianza nel training set



Feature Extraction | Principal Components Analysis



$$\mathbb{R}_d \rightarrow \mathbb{R}_k$$

Proiezione Una volta determinato lo spazio PCA, la proiezione di un pattern \mathbf{x} su tale spazio è semplicemente la proiezione geometrica del vettore \mathbf{x} sull'iperpiano che definisce lo spazio. In realtà la vera proiezione geometrica è un vettore che ha la stessa dimensionalità del vettore originale mentre in questo contesto indichiamo con proiezione il vettore (ridotto) nello spazio PCA. Matematicamente questa operazione è eseguita come prodotto della matrice di proiezione trasposta per il pattern \mathbf{x} al quale è preventivamente sottratta la media.

$$\overrightarrow{PCA}(\mathbf{x}, S_{\bar{\mathbf{x}}, \Phi_k}) = \Phi_k^t (\mathbf{x} - \bar{\mathbf{x}})$$

$$\mathbb{R}_k \rightarrow \mathbb{R}_d$$

Retroproiezione Dato un vettore \mathbf{y} nello spazio PCA, la sua retro-proiezione verso lo spazio originale si ottiene moltiplicando il vettore per la matrice di proiezione e sommando il vettore medio. Questa trasformazione non sposta spazialmente il vettore, che giace ancora sullo spazio PCA, ma opera un cambiamento di coordinate che ne permette la codifica in termini delle d componenti dello spazio originale.

$$\overleftarrow{PCA}(\mathbf{y}, S_{\bar{\mathbf{x}}, \Phi_k}) = \Phi_k \mathbf{y} + \bar{\mathbf{x}}$$

Feature Extraction | Principal Components Analysis

Se l'obiettivo è quello di scartare informazione inutile e dati correlati mantenendo gran parte del contenuto informativo si può scegliere k nel modo seguente:

Fissata una percentuale t del contenuto informativo che si vuole preservare (es. $t = 95\%$) si sceglie il minimo valore di k per cui la somma dei più grandi k autovalori e' maggiore o uguale a t rispetto alla somma di tutti gli autovalori

Considerando gli autovalori ordinati in ordine decrescente:

$$k = \arg \min_z \left\{ \frac{\sum_{i=1 \dots z} \lambda_i}{\sum_{i=1 \dots d} \lambda_i} \geq t \right\}$$

Poiche' che gli autovalori denotano la varianza lungo i diversi assi, il rapporto nella formula indica la varianza conservata rispetto alla varianza totale

La scelta di k è obbligata ad esempio per la visualizzazione 2D o 3D dei dati ($k=2, 3$)

Feature Extraction | Principal Components Analysis

Per d elevato (tipico nel caso di immagini, audio, ecc.) la matrice di covarianza può essere molto grande

Es per $d = 16384$, $\Sigma \in \mathbb{R}^{16384 \times 16384}$ oltre 268 milioni di valori !

E' piu' conveniente calcolare la matrice di proiezione attraverso la decomposizione ai valori singolari (Single Value Decomposition, SVD) della matrice rettangolare degli n pattern centralizzati $\mathbf{X} \in \mathbb{R}^{d \times n}$, $n \ll d$ senza passare per la matrice di covarianza

$$\mathbf{X} = \begin{bmatrix} \boxed{\mathbf{x}_1 - \bar{\mathbf{x}}} & \boxed{\mathbf{x}_2 - \bar{\mathbf{x}}} & \cdots & \boxed{\mathbf{x}_n - \bar{\mathbf{x}}} \end{bmatrix}$$

SVD per $n \ll d$: $\mathbf{X} = \mathbf{U}\mathbf{\Gamma}\mathbf{V}^t$, con $\mathbf{U} \in \mathbb{R}^{d \times n}$ ortonormale unitaria, $\mathbf{\Gamma} \in \mathbb{R}^{n \times n}$ diagonale, $\mathbf{V} \in \mathbb{R}^{n \times n}$ ortonormale

Gli elementi di $\mathbf{\Gamma}$ sono i valori singolari di \mathbf{X} , gli elementi di \mathbf{U} sono i vettori singolari sinistri di \mathbf{X} e gli elementi di \mathbf{V} sono i vettori singolari destri di \mathbf{X} . Si verifica che:

I vettori singolari di sinistra di \mathbf{X} sono gli autovettori di $\mathbf{X}\mathbf{X}^t$, i vettori singolari di destra di \mathbf{X} sono gli autovettori di $\mathbf{X}^t\mathbf{X}$, i vettori singolari non nulli di \mathbf{X} (che si trovano sulla diagonale principale di $\mathbf{\Gamma}$) sono le radici quadrate degli autovalori non nulli di $\mathbf{X}\mathbf{X}^t$ e $\mathbf{X}^t\mathbf{X}$

$$\Sigma = \frac{1}{n} \mathbf{X}\mathbf{X}^t = \frac{1}{n} \mathbf{U}\mathbf{\Gamma}\mathbf{V}^t\mathbf{V}\mathbf{\Gamma}\mathbf{U}^t = \frac{1}{n} \mathbf{U}\mathbf{\Gamma}^2\mathbf{U}^t$$

Gli autovettori e gli autovalori di Σ possono dunque essere ottenuti dalle colonne di \mathbf{U} (vettori singolari sinistri di \mathbf{X}) e corrispondenti elementi diagonali di $\mathbf{\Gamma}^2$ (valori singolari al quadrato di \mathbf{X}).

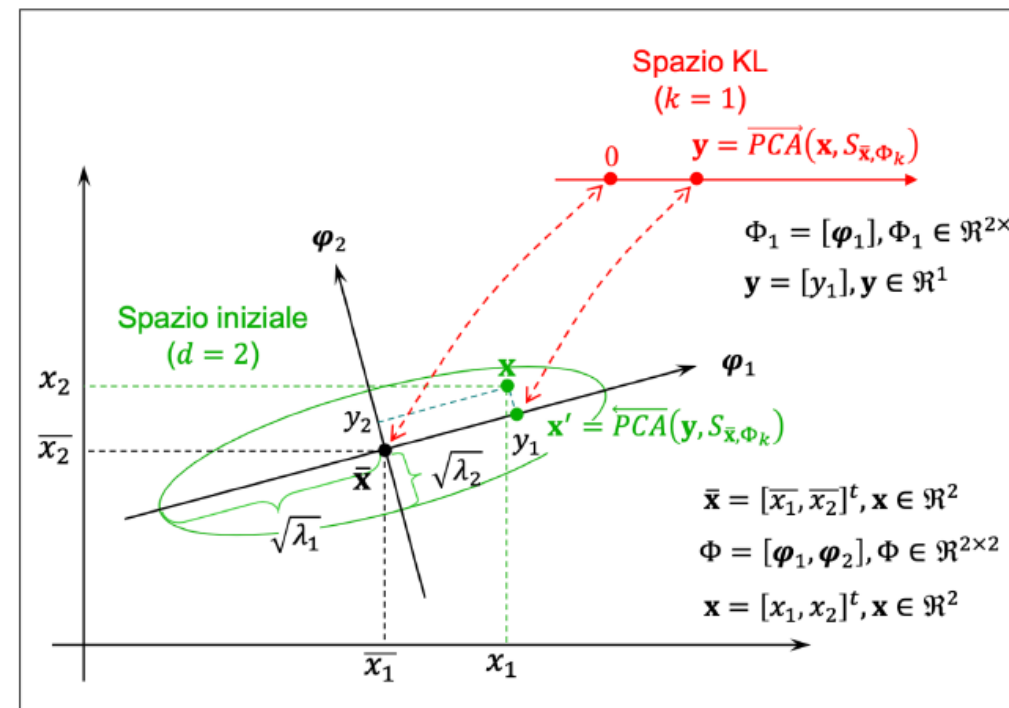
Feature Extraction | Principal Components Analysis

L'ellisse rappresenta la distribuzione dei pattern nel training set

ϕ_1 e ϕ_2 sono gli autovettori della matrice di covarianza

Gli autovalori λ_1 e λ_2 sono le varianze della distribuzione lungo gli assi φ_1 e φ_2 .

y_1 e y_2 sono le proiezioni di x sugli assi φ_1 e φ_2 .

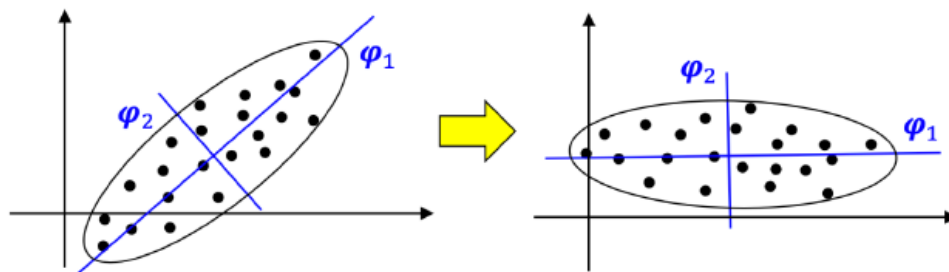


se λ_2 è piccolo, \mathbf{x} può essere approssimato con \mathbf{x}' (retroproiezione di \mathbf{y}) senza perdite significative di informazione

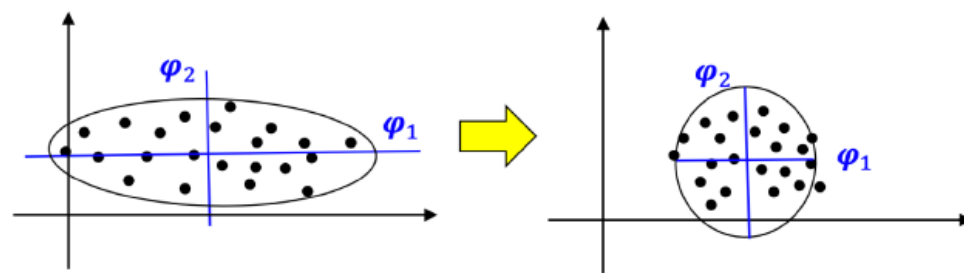
Feature Extraction | Principal Components Analysis

È una tecnica di pre-normalizzazione dei dati, che:

- Rimuove le correlazioni tra le dimensioni, ruotando la nuvola di punti per allineare gli assi di variazione principale dei dati (autovettori) agli assi cartesiani.



- Sfericizza l'ellissoide, uniformando le varianze (denotate dagli autovalori) a 1 lungo tutti gli assi



- Dopo aver proiettato i pattern sullo spazio PCA (definito dai primi k autovettori) è sufficiente dividere ogni dimensione per la radice quadrata dell'autovalore corrispondente (deviazione standard).

La matrice di covarianza dei dati normalizzati è l'identità

Feature Extraction | Principal Components Analysis

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

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

$$s = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)}}$$

Eigenvectors

Eigenvalues

$$s^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)} \quad var(X) = \frac{\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Step 5: Choosing components and forming a feature vector

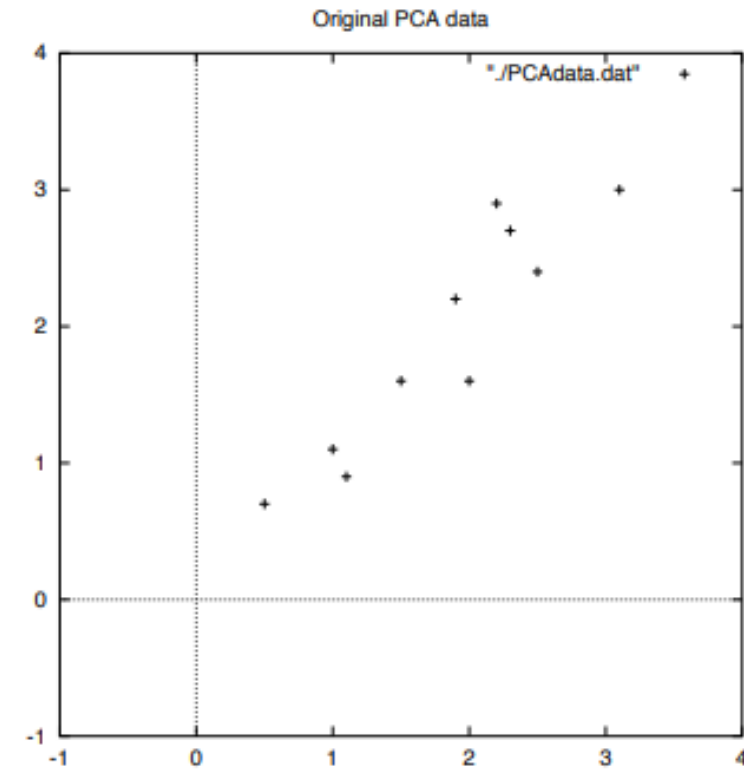
Step 5: Deriving the new data set

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9



A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

	x	y
Data =	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

	x	y
DataAdjust =	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
	1.29	1.09
	.49	.79
	.19	-.31
	-.81	-.81
	-.31	-.31
	-.71	-1.01

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

	<i>x</i>	<i>y</i>
	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
DataAdjust =	1.29	1.09
	.49	.79
	.19	-.31
	-.81	-.81
	-.31	-.31
	-.71	-1.01

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

	<i>x</i>	<i>y</i>
	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
DataAdjust =	1.29	1.09
	.49	.79
	.19	-.31
	-.81	-.81
	-.31	-.31
	-.71	-1.01

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

$$\text{eigenvalues} = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

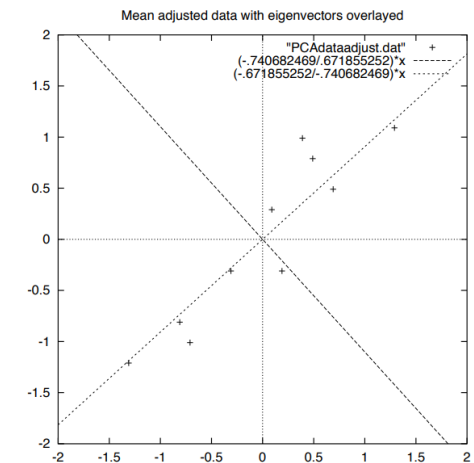


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

	<i>x</i>	<i>y</i>
	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
DataAdjust =	1.29	1.09
	.49	.79
	.19	-.31
	-.81	-.81
	-.31	-.31
	-.71	-1.01

$$\text{eigenvalues} = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

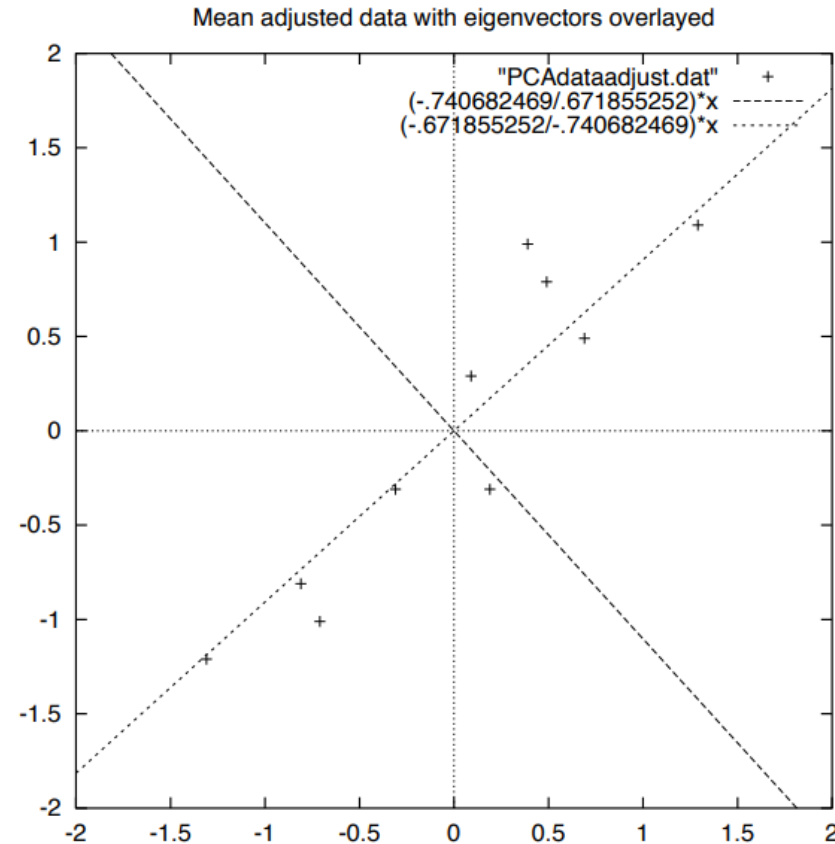


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Step 5: Choosing components and forming a feature vector

	<i>x</i>	<i>y</i>
	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
DataAdjust =	1.29	1.09
	.49	.79
	.19	-.31
	-.81	-.81
	-.31	-.31
	-.71	-1.01

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

$$FeatureVector = (eig_1 \ eig_2 \ eig_3 \ \ eig_n)$$

$$\begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

$$\begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Step 5: Choosing components and forming a feature vector

Step 5: Deriving the new data set

$$FinalData = RowFeatureVector \times RowDataAdjust,$$

where *RowFeatureVector* is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and *RowDataAdjust* is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Step 5: Choosing components and forming a feature vector

Step 5: Deriving the new data set

$$FinalData = RowFeatureVector \times RowDataAdjust,$$

where *RowFeatureVector* is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and *RowDataAdjust* is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

Transformed Data=

<i>x</i>	<i>y</i>
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

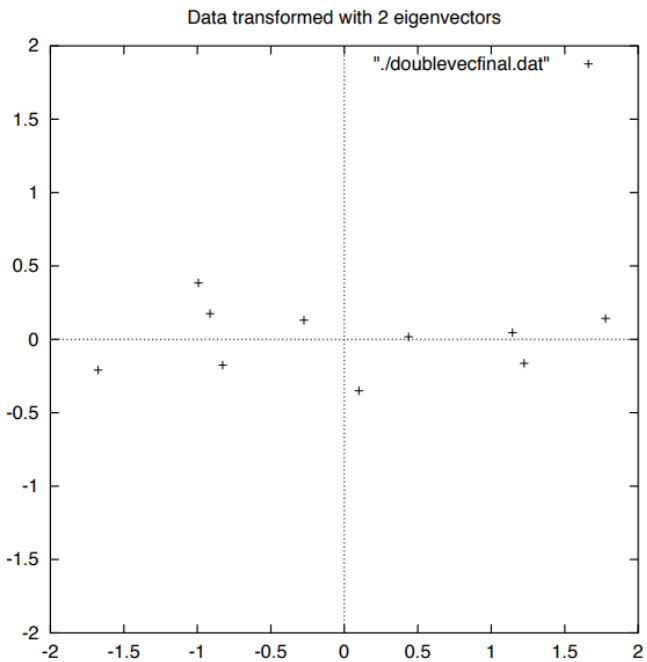


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Step 5: Choosing components and forming a feature vector

Step 5: Deriving the new data set

Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

$$FinalData = RowFeatureVector \times RowDataAdjust,$$

where *RowFeatureVector* is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and *RowDataAdjust* is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Step 5: Choosing components and forming a feature vector

Step 5: Deriving the new data set

Getting the old data back

$$FinalData = RowFeatureVector \times RowDataAdjust$$

$$RowDataAdjust = RowFeatureVector^{-1} \times FinalData$$

$$RowDataAdjust = RowFeatureVector^T \times FinalData$$

only true if the elements of the matrix are all unit eigenvectors

$$RowDataAdjust = RowFeatureVector^T \times FinalData$$

$$RowOriginalData = (RowFeatureVector^T \times FinalData) + OriginalMean$$

A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis

Step 1: Get some data

Step 2: Subtract the mean

Step 3: Calculate the covariance matrix

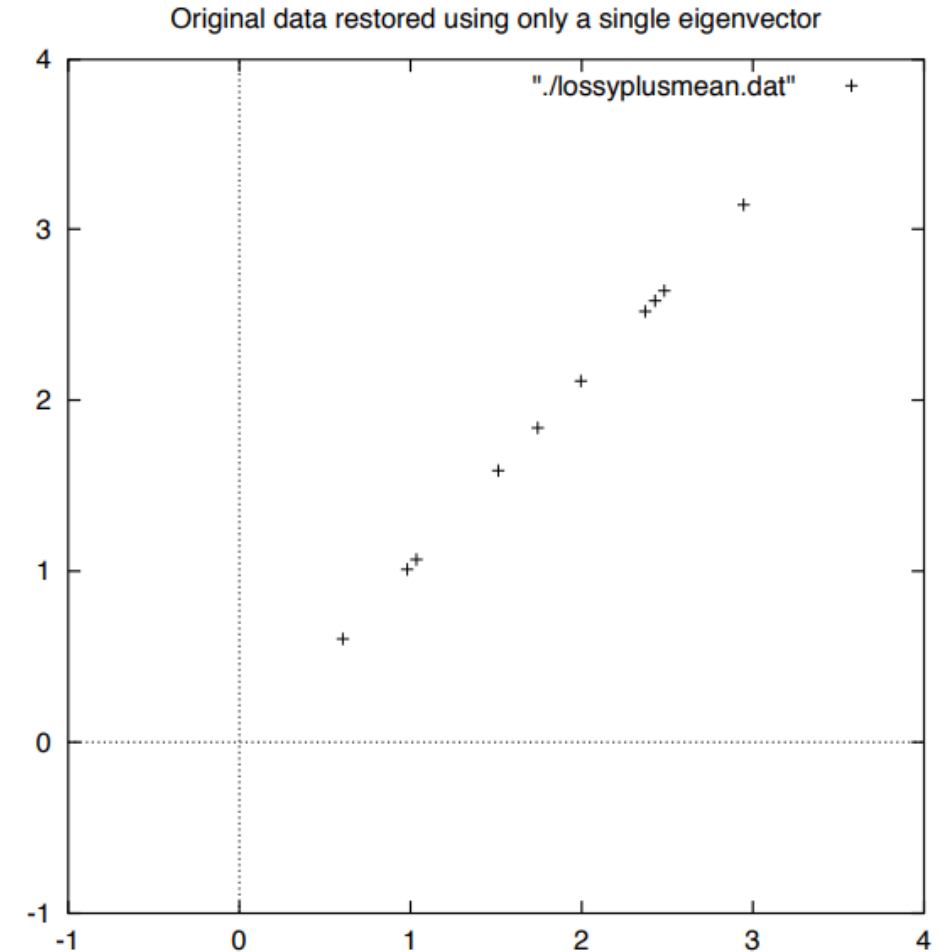
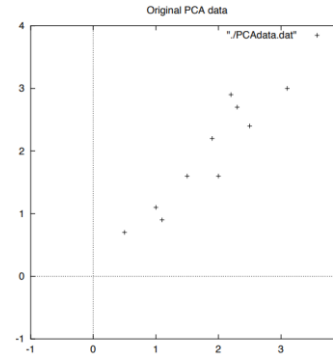
Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Step 5: Choosing components and forming a feature vector

Step 5: Deriving the new data set

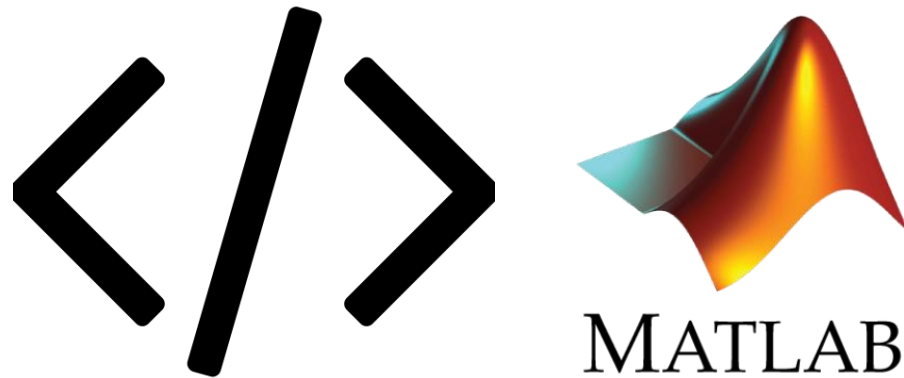
Getting the old data back

$$\text{RowOriginalData} = (\text{RowFeatureVector}^T \times \text{FinalData}) + \text{OriginalMean}$$



A tutorial on Principal Components Analysis by Lindsay I. Smith

Feature Extraction | Principal Components Analysis



Feature Extraction

- Accuracy improvements.
- Overfitting risk reduction.
- Speed up in training.
- Improved Data Visualization.
- Increase in explainability of our model.

Many other
techniques...

FEATURE SELECTION

Linear Discriminant Analysis

Riduzione di dimensionalità lineare e supervisionata il cui obiettivo è massimizzare la separazione tra le classi.

Per formulare il criterio di ottimizzazione di massima separazione tra le classi sono definite le seguenti matrici di scattering (sparpagliamento):

within-class

indica come i vettori sono scattered rispetto al centro delle classi (ciascuno rispetto alla propria classe).

between-class

indica come i centri delle classi sono scattered rispetto al centro generale della distribuzione (ovvero quanto le classi sono scattered).

Linear Discriminant Analysis

Dato un training set contenente n pattern $(\mathbf{x}_1, y_1) \dots (\mathbf{x}_n, y_n)$,

dove $\mathbf{x}_i \in \mathbb{R}^d$ sono i pattern multidimensionali e $y_i \in [1 \dots s]$ le etichette delle s classi.

Siano n_i e $\bar{\mathbf{x}}_i$ il numero di pattern e il vettore medio della classe i -esima.

Allora le matrici di scattering sono definite come:

within-class $\mathbf{S}_w = \sum_{i=1 \dots s} \mathbf{S}_i, \quad \mathbf{S}_i = \sum_{\mathbf{x}_j | y_j=i} (\mathbf{x}_j - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_i)^t$ matrice di covarianza senza
normalizzare per il numero di pattern

← pattern della classe i – esima

← pattern della classe i – esima

between-class $\mathbf{S}_b = \sum_{i=1 \dots s} n_i \cdot (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_0)(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_0)^t, \quad \bar{\mathbf{x}}_0 = \frac{1}{n} \sum_{i=1 \dots s} n_i \cdot \bar{\mathbf{x}}_i$

← media globale

Linear Discriminant Analysis

Il criterio per la soluzione ottimale e' intuitivo in quanto cerca di massimizzare lo scattering tra le classi (between class S_b) minimizzando al contempo quello all'interno di ogni classe (within class, S_w)

Il che equivale a massimizzare la quantità:

$$J_1 = \text{tr}(\mathbf{S}_w^{-1}\mathbf{S}_b) = \sum_{i=1\dots d} \lambda_i$$

ove tr è la traccia (somma degli autovalori) della matrice.

Si dimostra che per massimizzare J_1 lo spazio LDA e' definito dagli autovettori relativi ai primi k autovalori della matrice $\mathbf{S}_w^{-1}\mathbf{S}_b$ ($k < n$, $k < s$, $k < d$) (analogia con PCA)

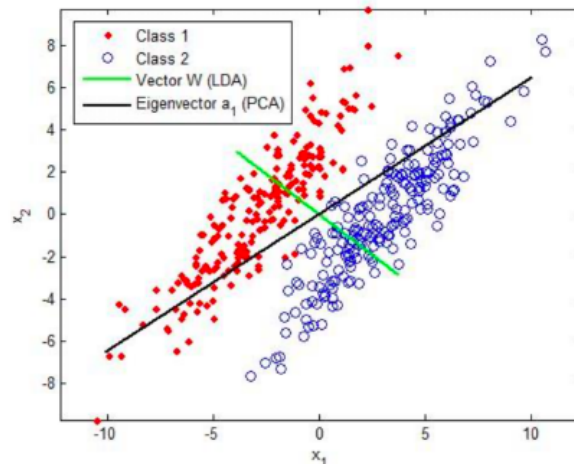


Valore massimo di $k = s - 1$

Linear Discriminant Analysis vs. Principal Components Analysis

Riduzione di dimensionalità da $d = 2$ a $k = 1$

Sono eseguiti mapping lineari $\mathbb{R}_2 \rightarrow \mathbb{R}_1$ ma la soluzione (retta) è profondamente diversa.



Il segmento nero che identifica la soluzione PCA è l'iperpiano sul quale proiettando i pattern (indipendentemente dalla loro classe) conserviamo al massimo l'informazione.

Il segmento verde che identifica la soluzione LDA è l'iperpiano sul quale proiettando i pattern siamo in grado di discriminare al meglio le due classi

Mentre PCA privilegia le dimensioni che rappresentano al meglio i pattern, LDA privilegia le dimensioni che discriminano al meglio i pattern del TS.