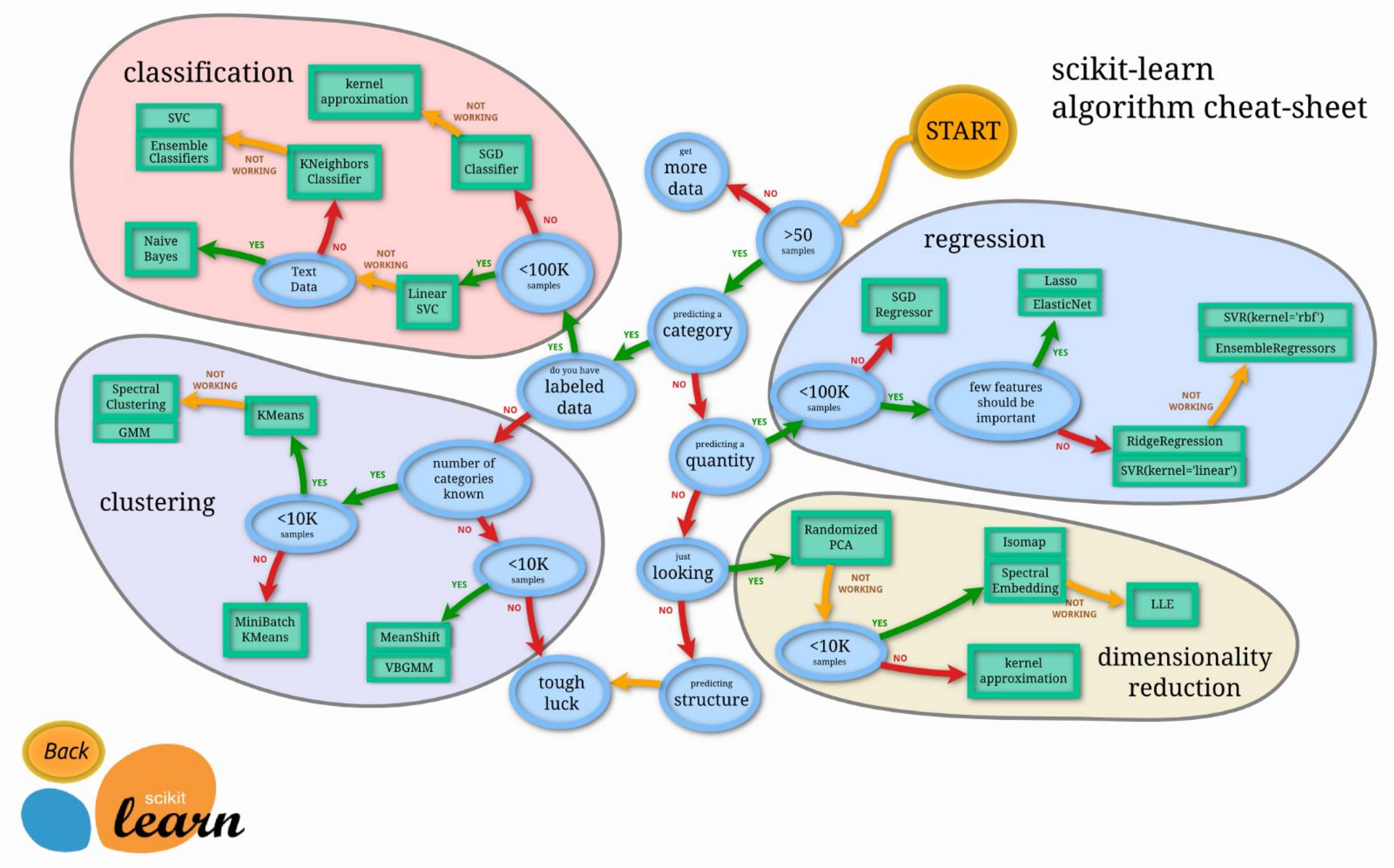
Unsupervised learning 1

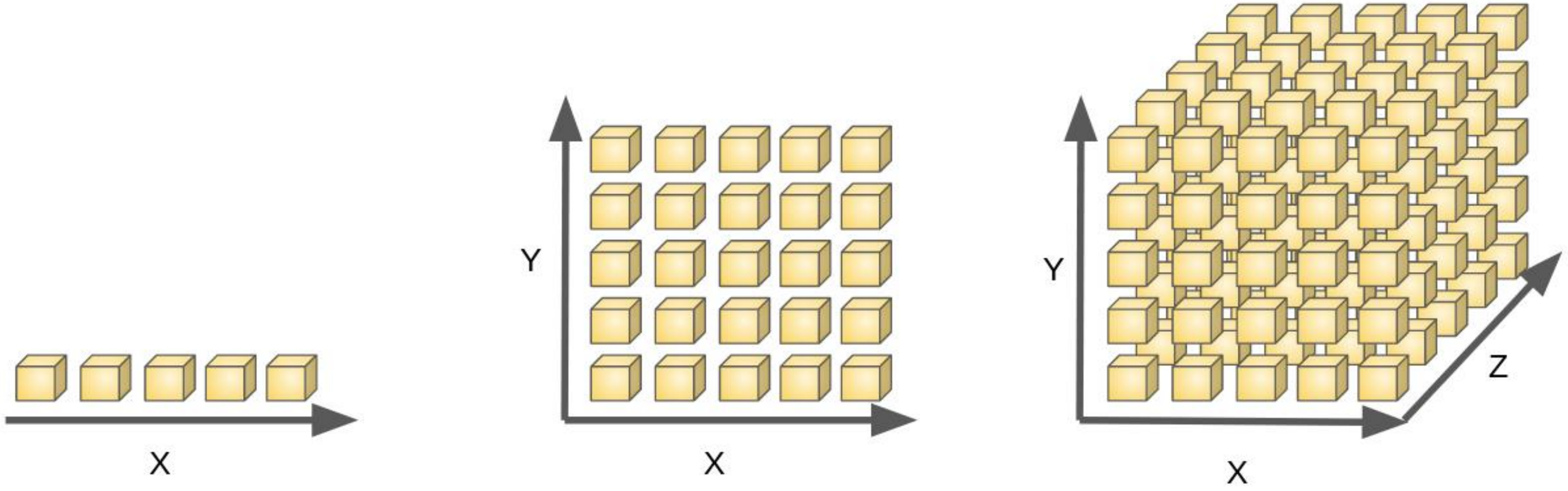
Dimensionality reduction og data visualisering

- PCA

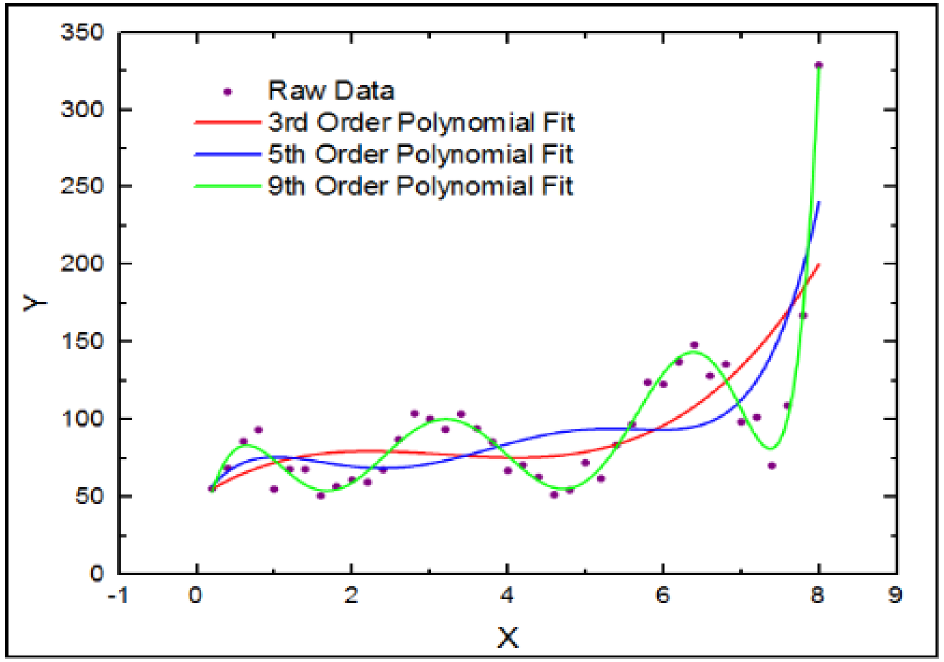
Hvor er vi ? Dimensionality reduction..



Curse of dimensionality



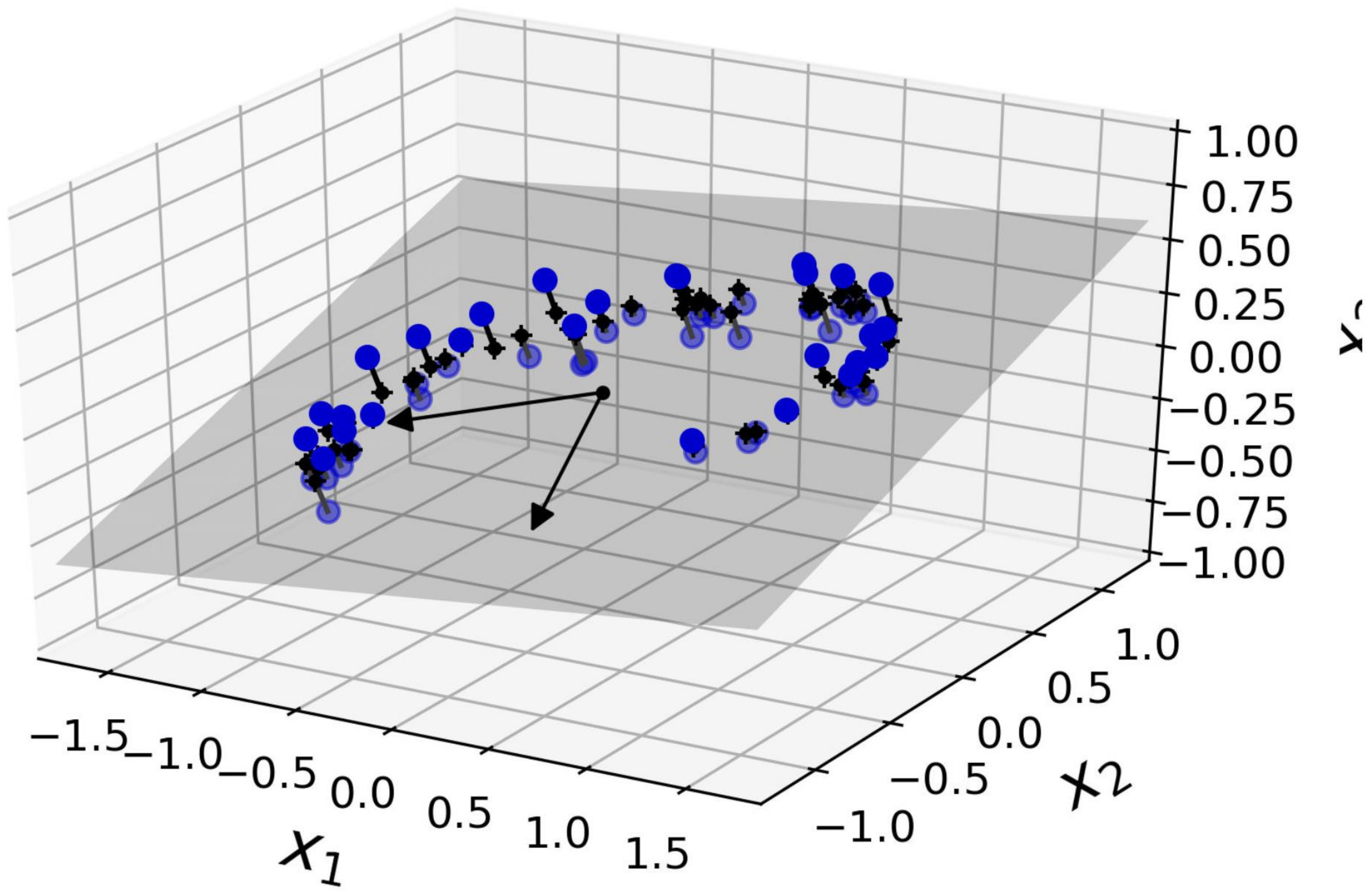
Curse of dimensionality – e.g. polynomial regression



Polynomial curve fitting, M = 3

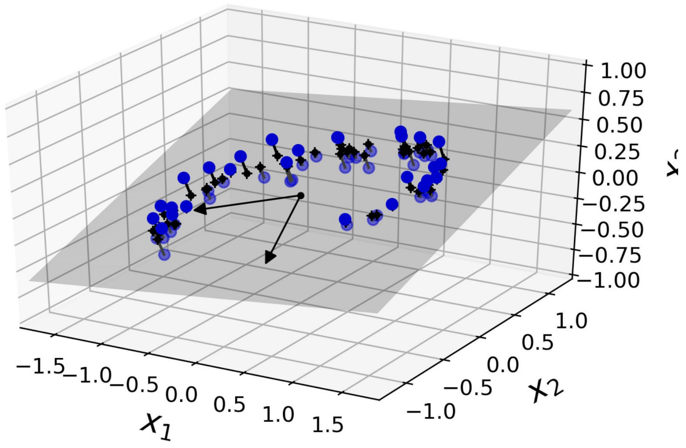
$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

Dimensions-reduktion

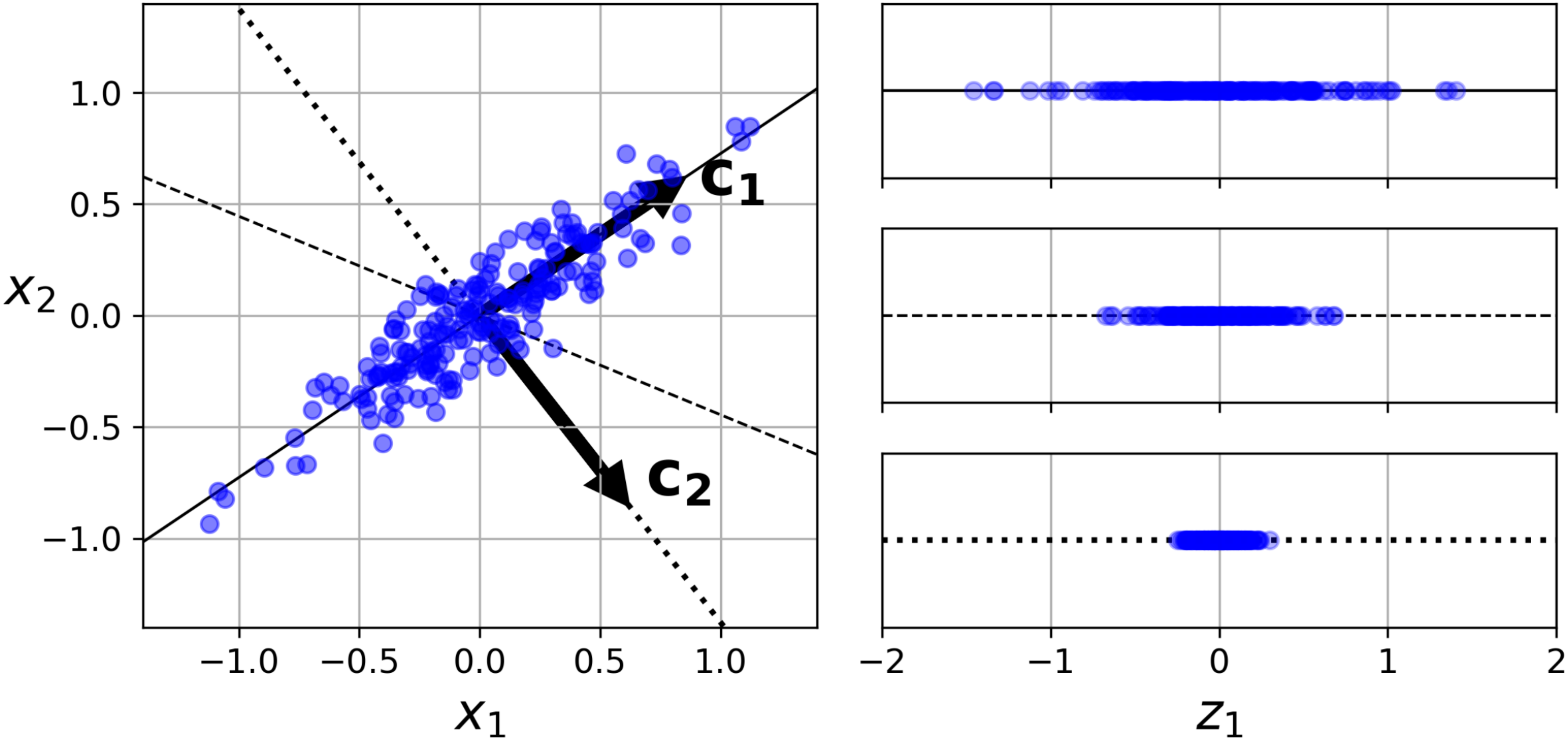


Principal Component Analysis (PCA) – formål

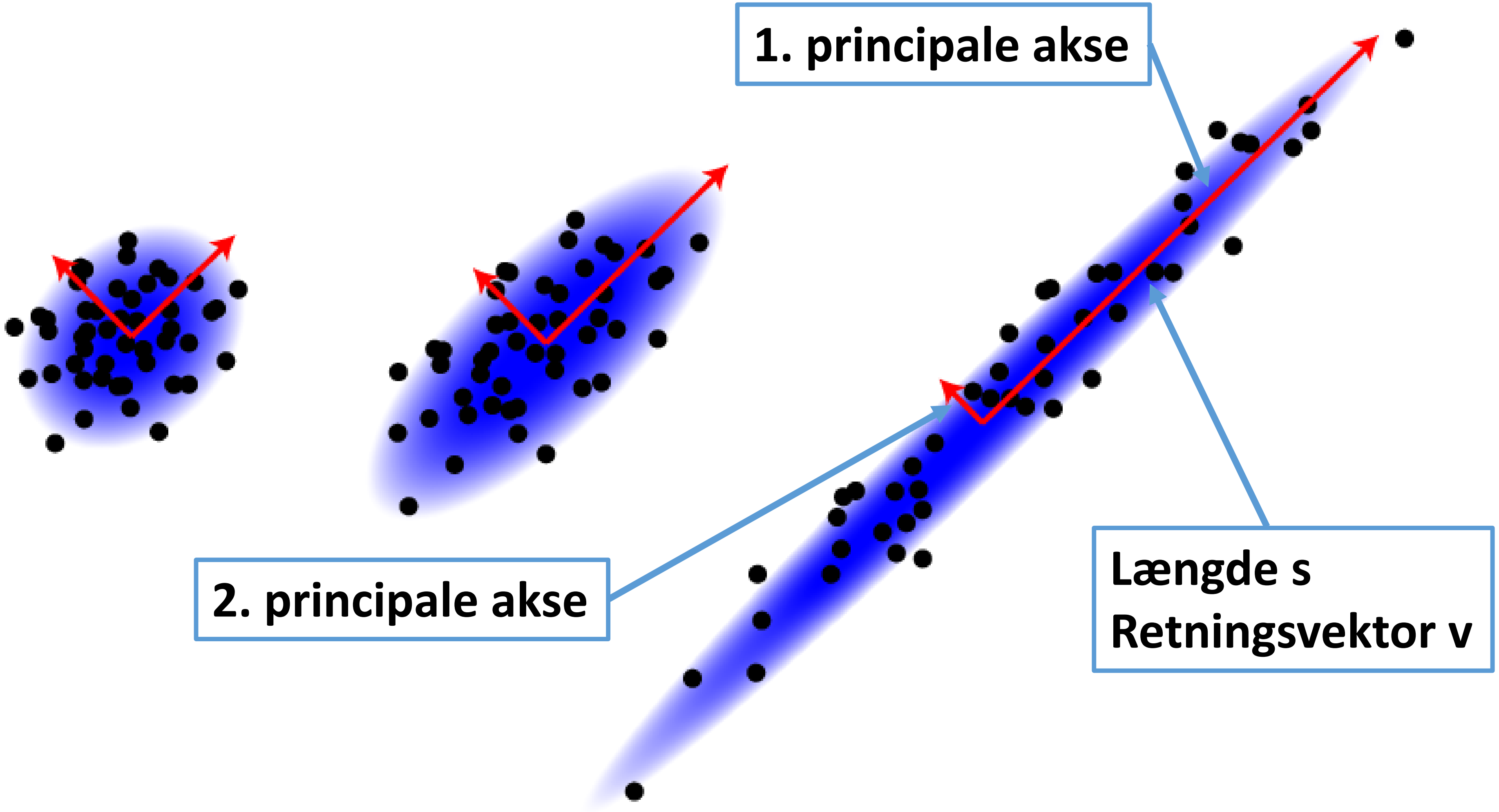
- Dimensionsreduktion
  - Visualisering
  - Præprocessering - undgå overfit og hurtigere træning
- Data analyse
- Kompression
- ..



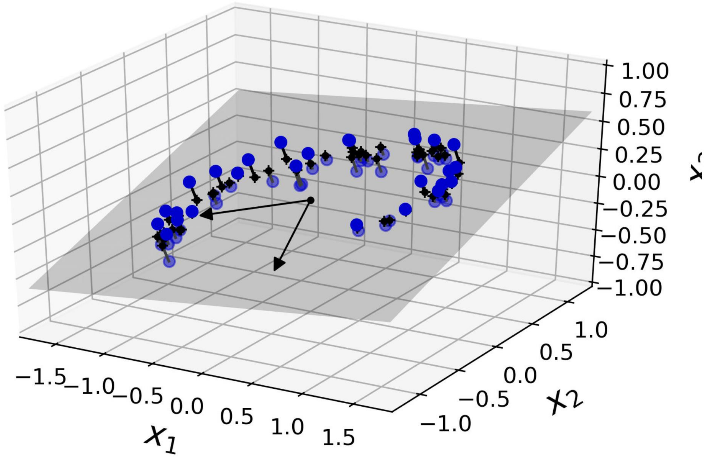
PCA Princip : Maksimer varians



PCA – geometrisk fortolkning



PCA - Projektion på principale akser



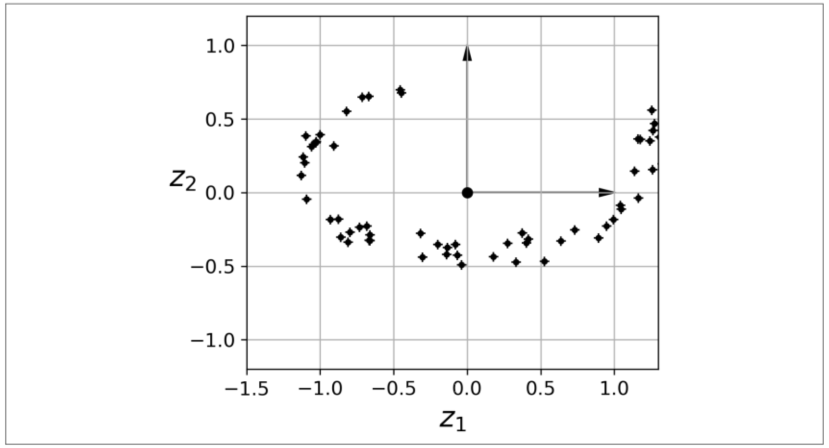


Figure 8-3. The new 2D dataset after projection

Equation 8-2. Projecting the training set down to d dimensions

$$\mathbf{X}_{d\text{-proj}} = \mathbf{X}\mathbf{W}_d$$

Valg af antal komponenter

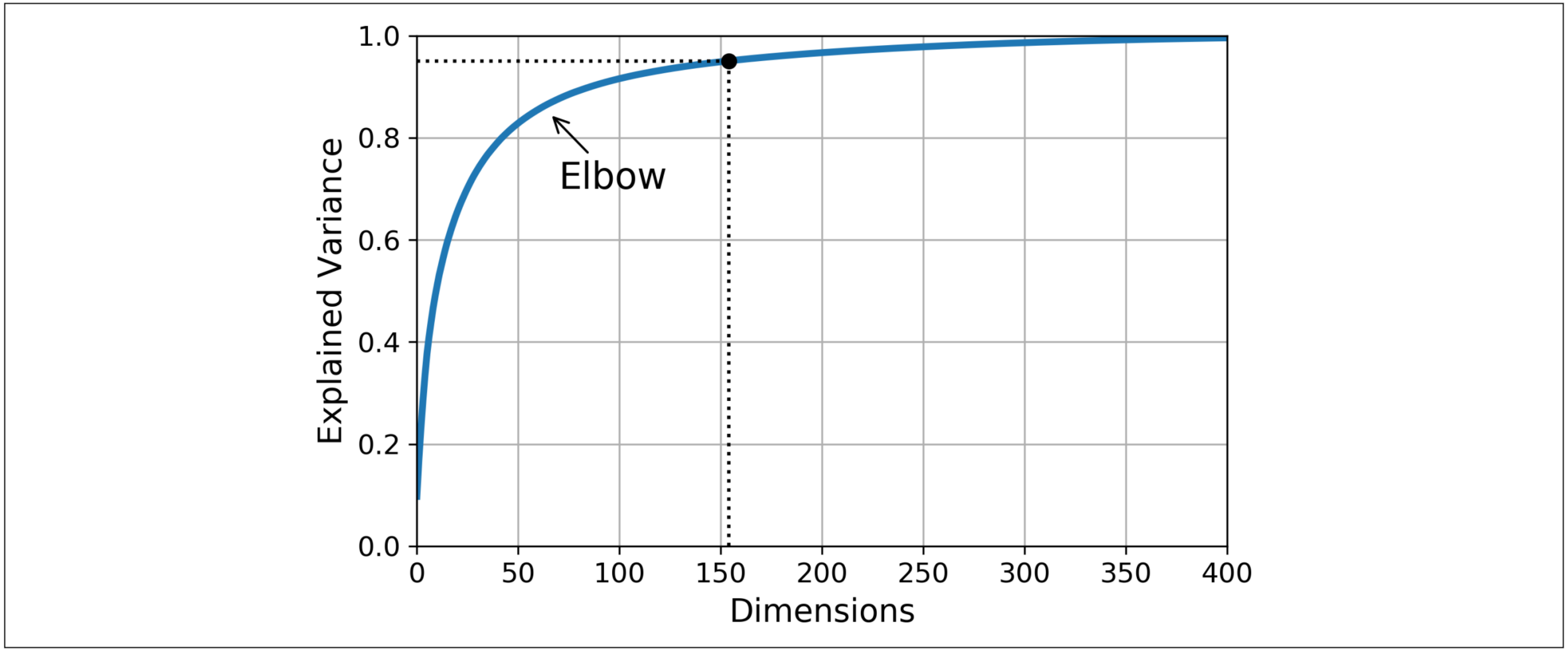


Figure 8-8. Explained variance as a function of the number of dimensions

PCA til kompression

```
pca = PCA(n_components = 154)
X_reduced = pca.fit_transform(X_train)
X_recovered = pca.inverse_transform(X_reduced)
```

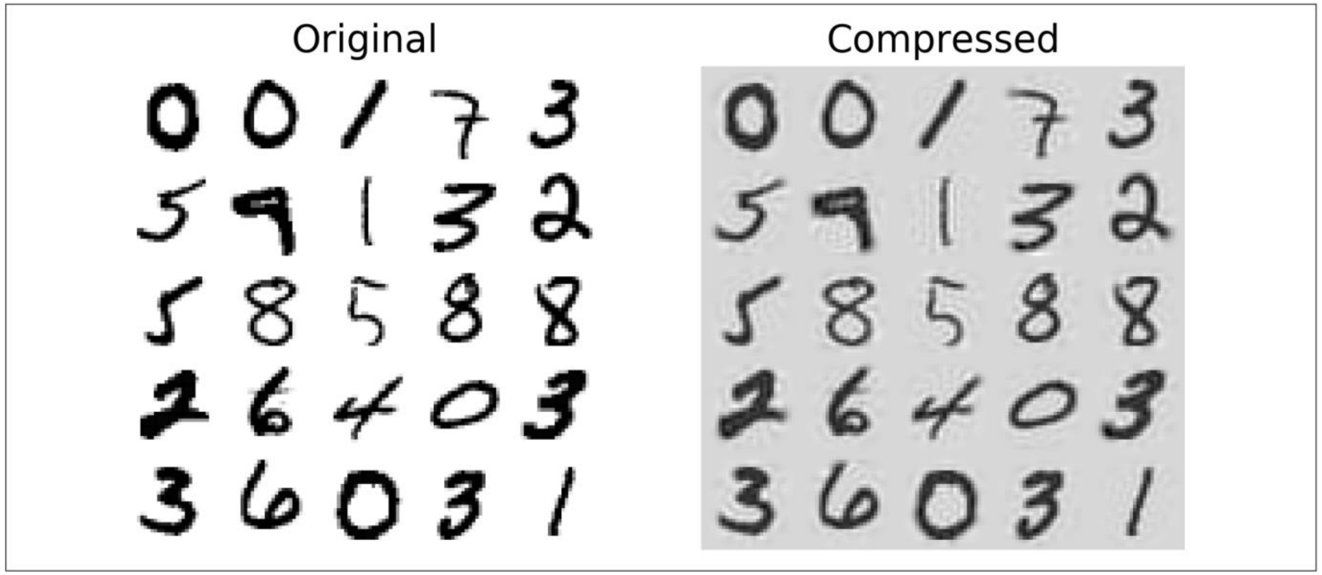


Figure 8-9. MNIST compression preserving 95% of the variance

PCA – Scikit implementation

class `sklearn.decomposition.` **PCA** (*n_components=None, copy=True, whiten=False, svd_solver='auto', tol=0.0, iterated_power='auto', random_state=None*) [source]

Principal component analysis (PCA)

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.

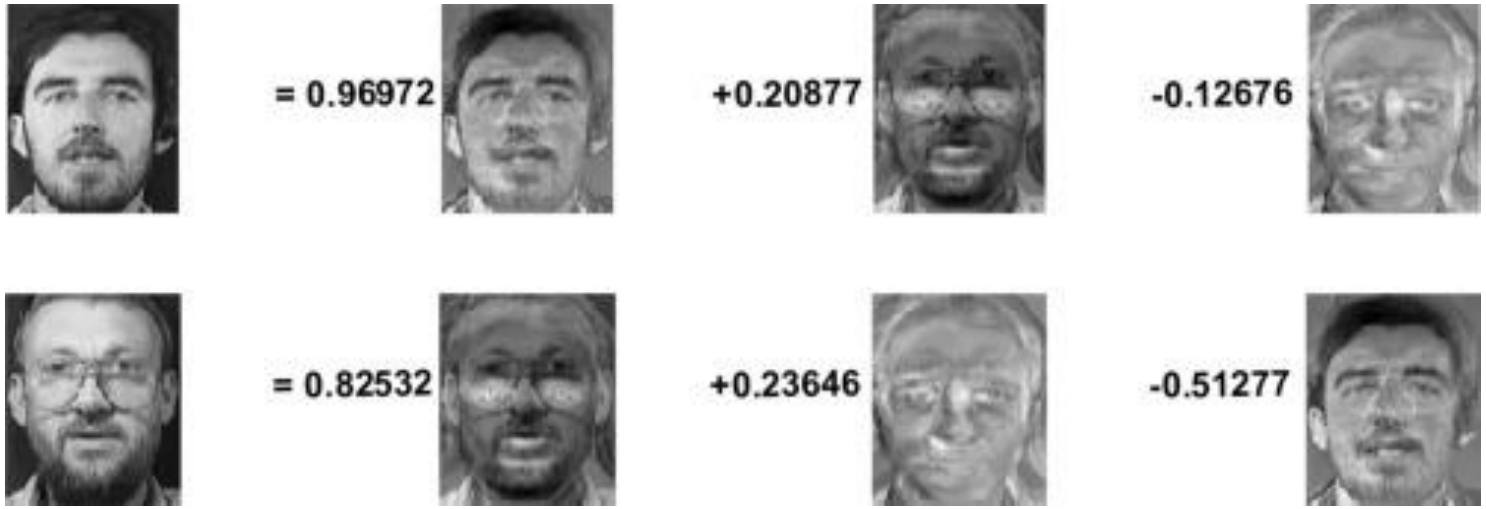
PCA – med SVD

$$\mathbf{V} = \begin{pmatrix} | & | & & | \\ \mathbf{c_1} & \mathbf{c_2} & \cdots & \mathbf{c_n} \\ | & | & & | \end{pmatrix}$$

```
X_centered = X - X.mean(axis=0)
U, s, Vt = np.linalg.svd(X_centered)
c1 = Vt.T[:, 0]
c2 = Vt.T[:, 1]
```

X = data matrix (N_samples x N_dims)
Vt = Eigenvectors (retningsvektorer med længde 1)
s = Singular values (angiver længde af vektorer)
U – benyttes ikke her

Eigenfaces – repræsentation af ansigter



Data i "non-linear manifold" (mangfoldighed) / embedding

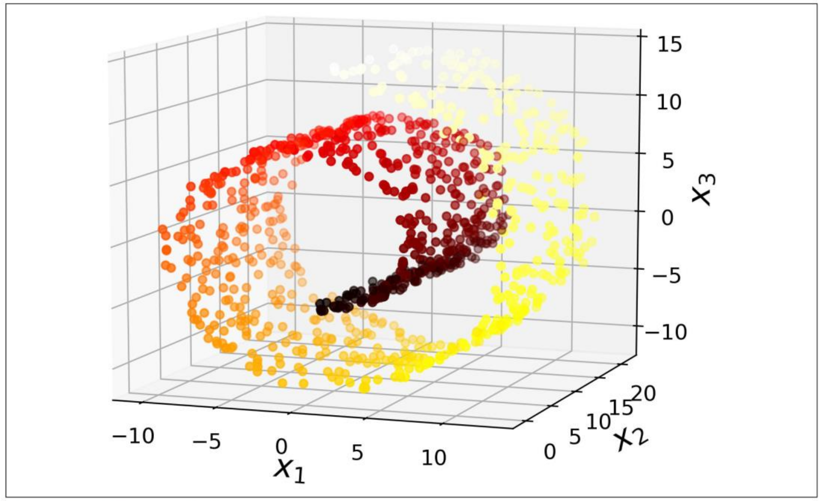


Figure 8-4. Swiss roll dataset

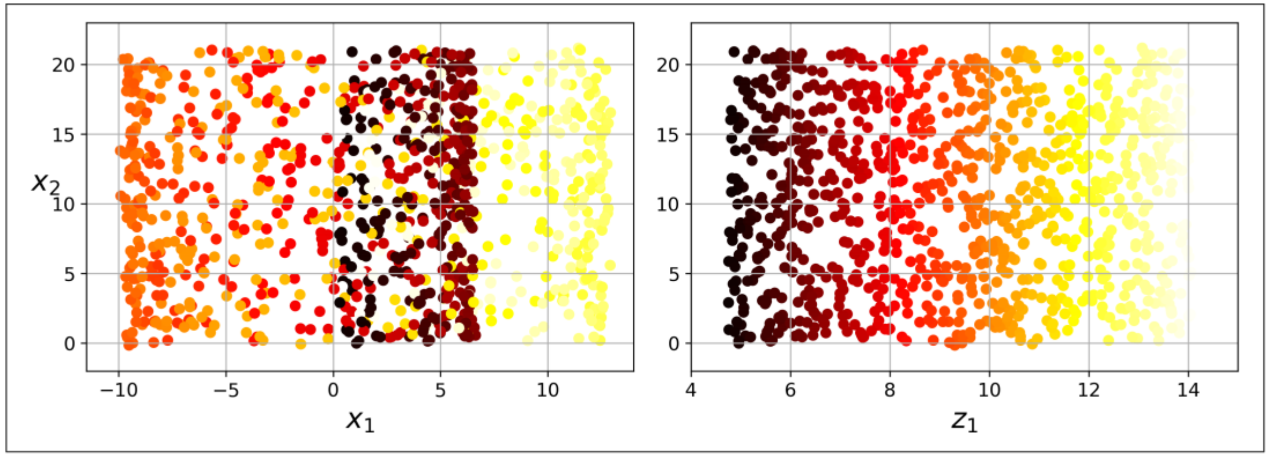


Figure 8-5. Squashing by projecting onto a plane (left) versus unrolling the Swiss roll (right)

Other methods

- Clustering metoder (i Unsupervised Learning 2)
  - Kmeans / GMM, hierarchical clustering
- Manifold learning
  - Isomap, LLE, t-SNE, ..
- Decomposition
  - ICA, NMF, ..
- Deep learning methods
  - Autoencoders
  - GAN