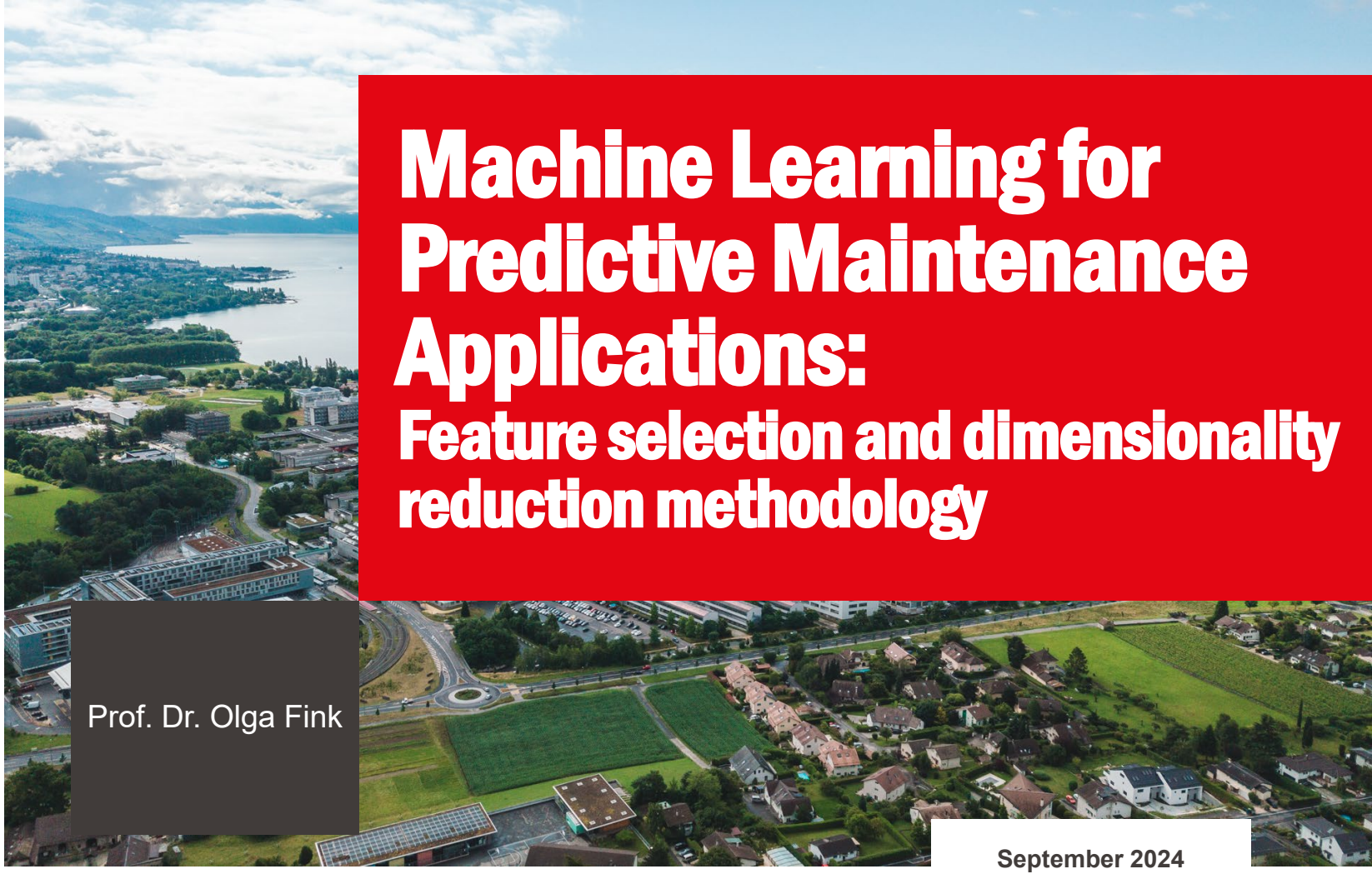


Machine Learning for Predictive Maintenance Applications:

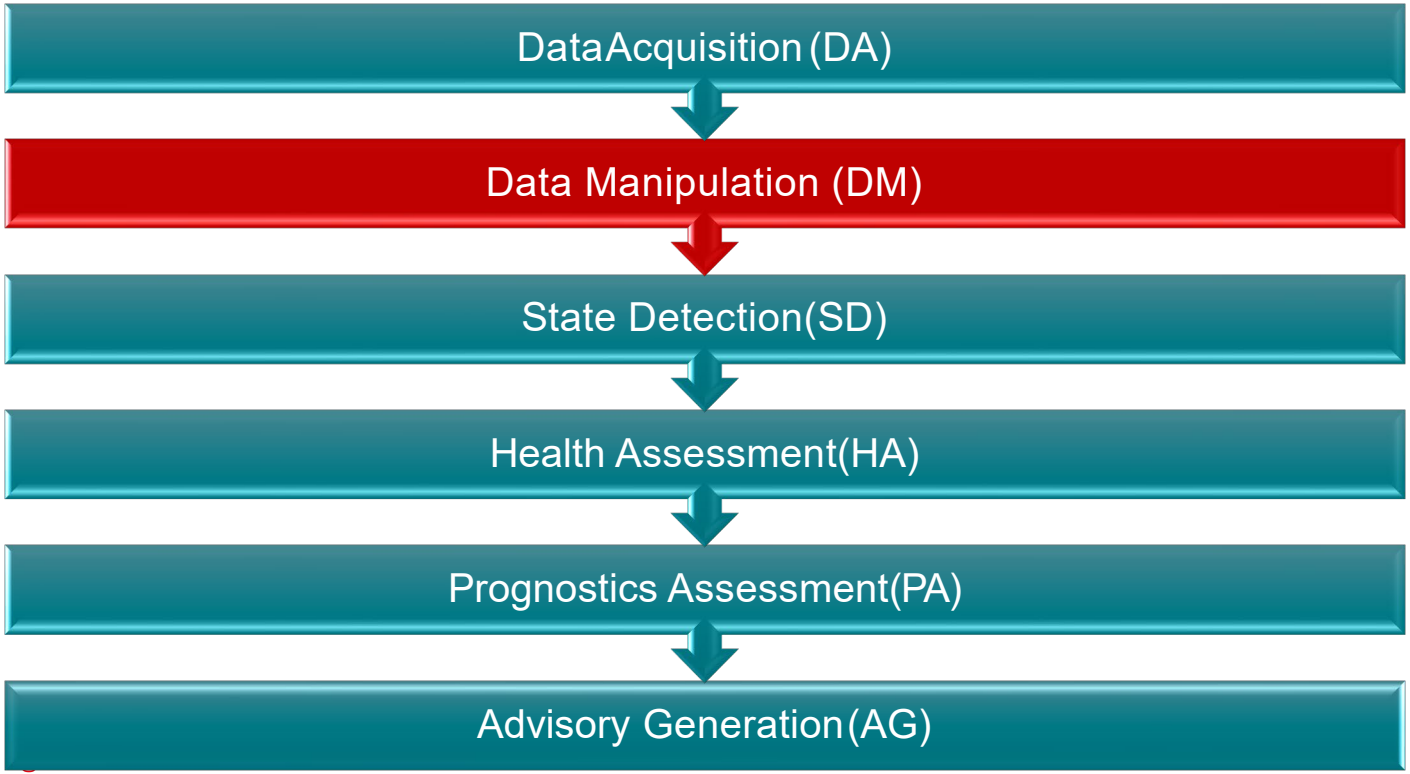
Feature selection and dimensionality reduction methodology

Prof. Dr. Olga Fink

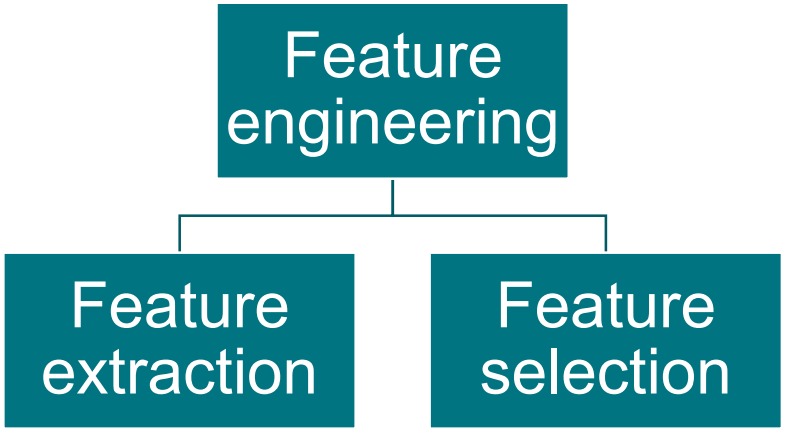
École polytechnique fédérale de Lausanne

September 2024

PHM Process


DataAcquisition (DA)
Data Manipulation (DM)
State Detection(SD)
Health Assessment(HA)
Prognostics Assessment(PA)
Advisory Generation (AG)

Feature engineering



Why feature subset selection?

- Features may be expensive to obtain
  - You evaluate a large number of features (sensors) in the test bed and select only a few for the final implementation
- You may want to extract meaningful rules from your classifier / regression algorithm
  - When you project, the measurement units of your features (length, weight, etc.) are lost
- Features may not be numeric
- …

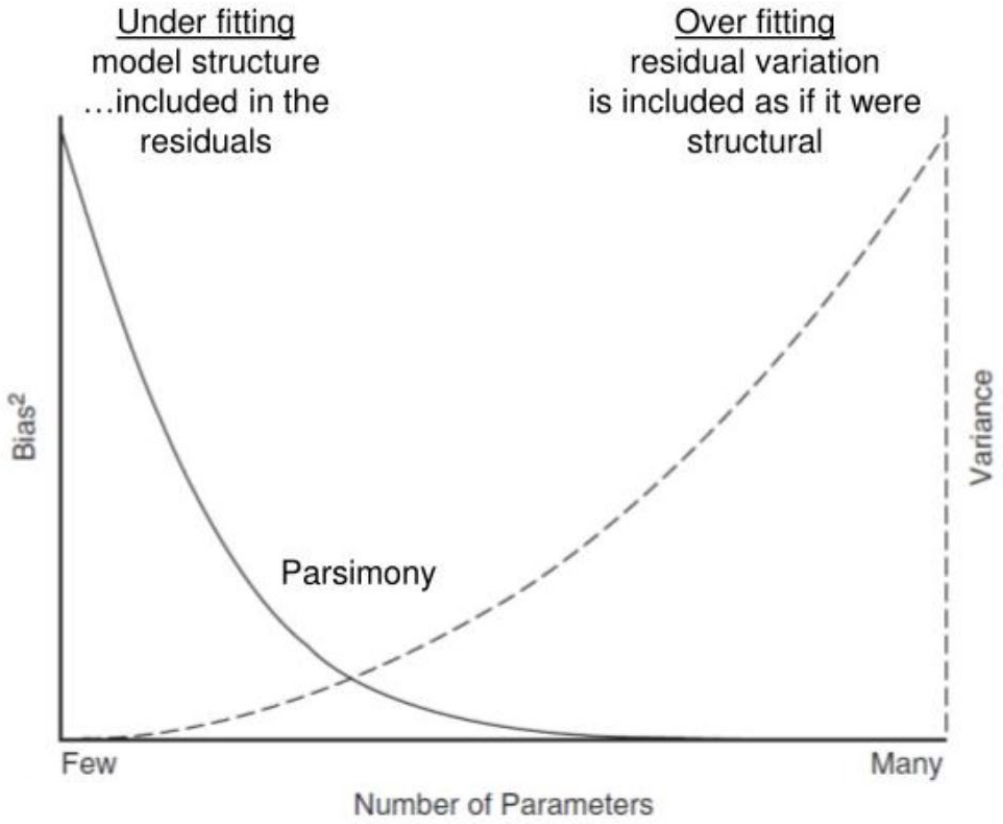
Feature selection

- Aims to choose a small subset of the relevant features from the original features by removing
  - irrelevant,
  - redundant,
  - or noisy features.
- Can usually lead to
  - better learning performance,
  - higher learning accuracy,
  - lower computational cost,
  - and better model interpretability.

Obejctives of feature selection

- Improved Model Performance
- Enhanced Interpretability
- Computational Efficiency
- Better Convergence
- Enhanced generalization by reducing overfitting

Underfitting vs overfitting



Important criteria to consider for feature selection

- Relevance + Redundancy (concerning the goal)
- Relevance of the feature is measured based on the characteristics of the data not by its value
- Redundant features are those that are weakly relevant but can be completely replaced with a set of other features such that the target distribution is not disturbed
- Redundancy is always inspected in multivariate cases (when examining feature subset)
- Relevance is established for individual features.
- Feature subsets can be classified as
  - noisy and irrelevant
  - redundant & weakly relevant
  - weakly relevant and non-redundant
  - strongly relevant
- The distortion of irrelevant and redundant features is not due to the presence of un-useful information

→ because the features did not have a statistical relationship with other features

Aim of feature selection

- **Maximize relevance and minimize redundancy!!!**

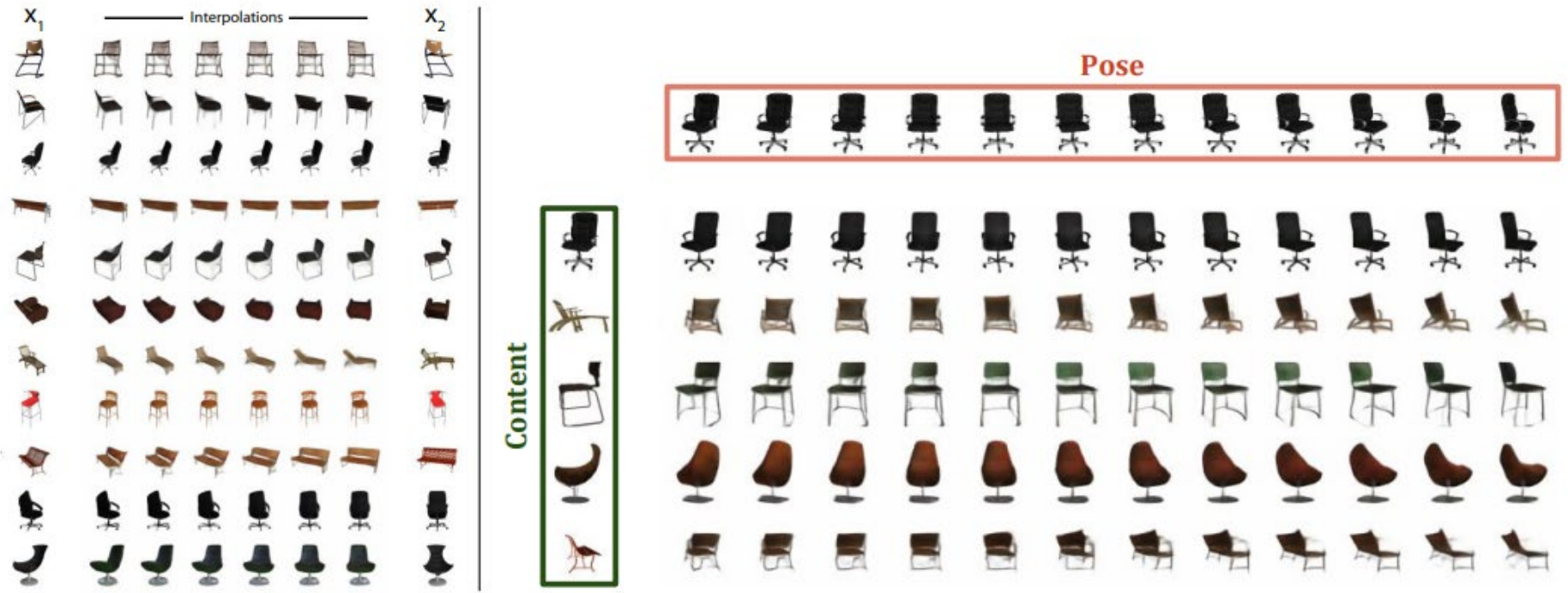
What Makes Some Feature Representations Better Than Others?

- Disentangling of causal factors
- Easy to model
- Works well with regularization strategies

Idea of disentanglement

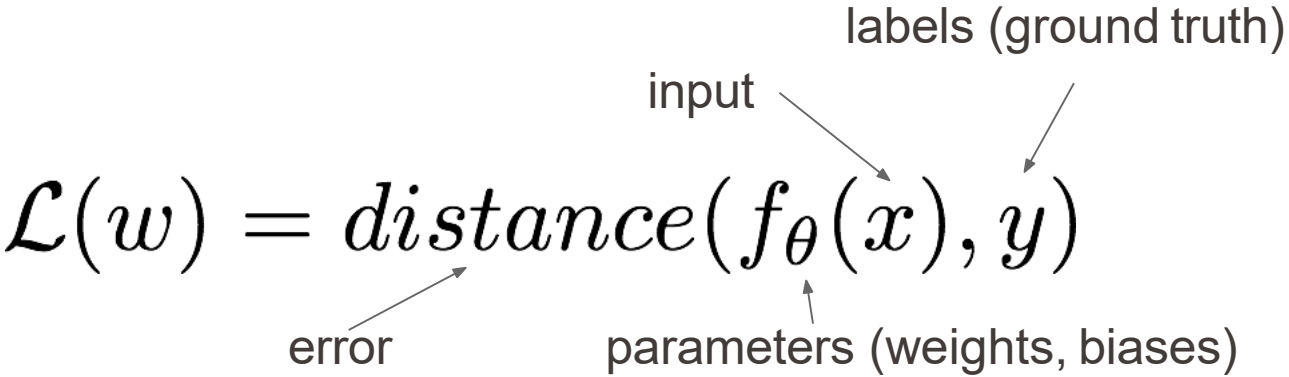
- The concept of disentanglement is based on the hypothesis that real-world data is generated by a few independent explanatory factors of variation
- Can be sued for controlled data generation:
  - learn a disentangled feature representation of the data
  - use these disentangled features representing independent factors of variation to generate data samples with desired characteristics in controlled ways

Disentangled features: generation



Denton, Emily L. "Unsupervised learning of disentangled representations from video." *Advances in neural information processing systems* 30 (2017).

Loss functions



Preliminaries

$\mathbf{x} = (x_1, \ldots, x_N) \in X$ - a vector of inputs

$y \in T$ - a target variable

$f_\theta(\mathbf{x})$ - a prediction model

$\mathcal{L}(y, f_\theta(\mathbf{x}))$ - the loss function for measuring errors.

Usual choices for regression:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2$$ squared error, L_2-norm

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}|y_i - f_\theta(x_i)|$$ absolute error, L_1-norm

... and classiication:

labels (one-hot)

$$\mathcal{L} = -\sum_{i=1}^{n}\mathbf{y}_i \log(S(f_\theta(\mathbf{x}_i)))$$ Cross-entropy loss

Softmax

Basic principles of dimensionality reduction

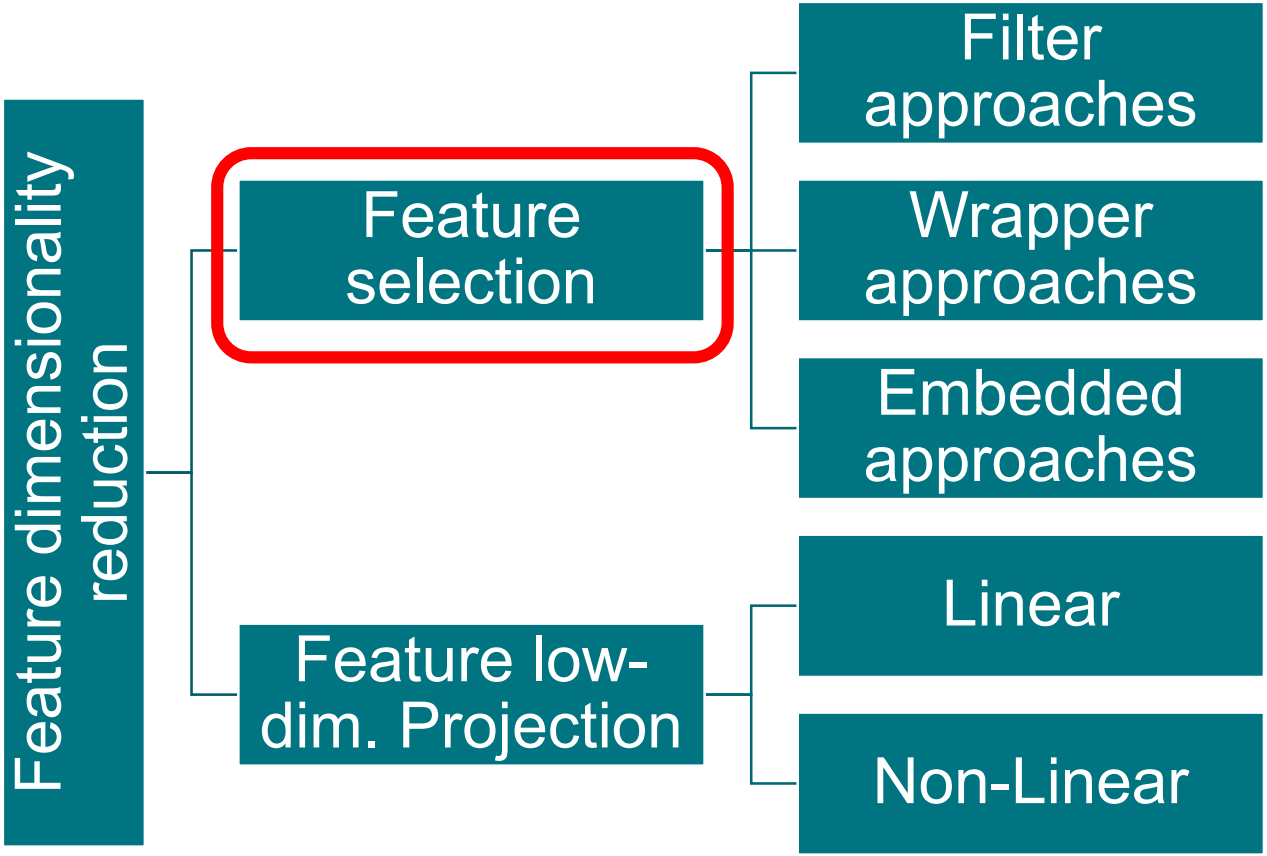
- Given a feature space $x_i \in \Re^N$ find a mapping $y = f_\theta(x): R^N \rightarrow R^M$

With M<N such that the transformed feature vector $y \in R^M$ preserves (most of) the information in structure in R^N

- An optimal mapping $y= f_\theta(x)$ is one that does not increase $P[error]$

- Two approaches:
  - Feature extraction: creating a subset of new features by combinations of the existing features
  - Feature selection: choosing a subset of all the features

Different approaches to dimensionality reduction



Feature subset selection

- Given a feature set $X = \{x_i | i = 1 \ldots N\}$, find a subset Y_M, with M<N, that maximizes an objecive function $J(Y)$, ideally P(correct)

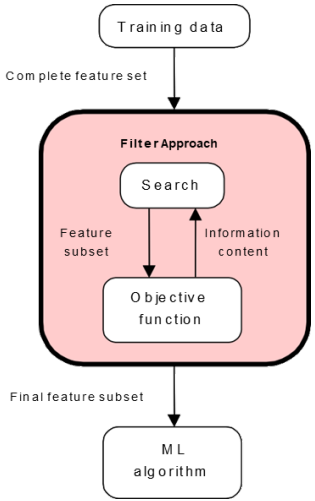
$$Y_M = \{x_{i1}, x_{i2}, \ldots, x_{iM}\} = \arg\max_{M, i_M} J\{x_i | \, i = 1..N\}$$

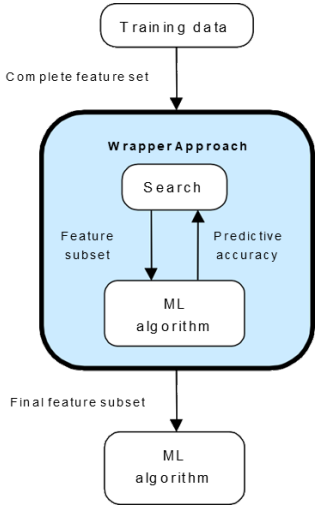
Search strategy and objective function

- Two inputs required:
  - A search strategy to select candidate subsets
  - An objective function to evaluate these candidates
- Objective Function
  - The objective function evaluates candidate subsets and returns a measure of their "goodness", a feedback signal used by the search strategy to select new candidates
- Search strategy
  - Exhaustive evaluation of feature subsets involves $\binom{N}{M}$ combinations for a fixed value of M, and 2^N combinations if M must be optimized as well

Objective functions

- **Filters**: evaluate subsets by their information content, e.g., interclass distance, statistical dependence or information-theoretic measures
- **Wrappers**: use a classifier to evaluate subsets by their predictive accuracy (on test data) by statistical resampling or cross-validation





Source: Gutierrez-Osuna, 2013

Filter types

- Distance or separability measures
  - Distance between classes: Euclidean, Mahalanobis, etc.

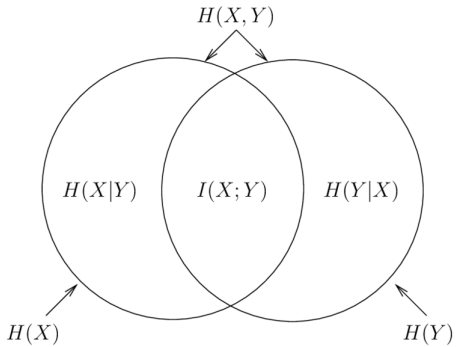
- Correlation and information-theoretic measures
  - are based on the rationale that good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other
  - Linear relation measures
    - Linear relationship between variables can be measured using the correlation coefficient

$$J(Y_M) = \frac{\sum_{i=1}^{M} \rho_{ic}}{\sum_{i=1}^{M} \sum_{j=i+1}^{M} \rho_{ij}}$$

    - Where ρ_{ic} is the correlation coefficient between feature i and the class label and ρ_{ij} is the correlation coefficient between features i and j

Filter types

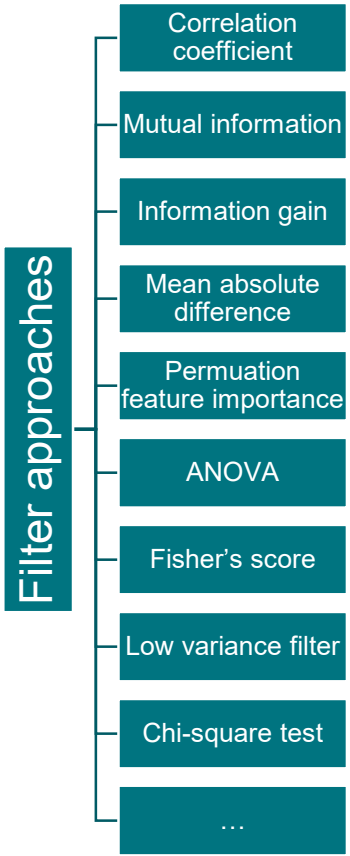
- Non-linear relation measures
  - Correlation is only capable of measuring linear dependence
  - A more powerful measure is the mutual information $I(Y_M; C)$



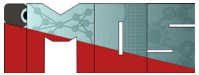
$$J(Y_M) = I(Y_M; C) = H(C) - H(C|Y_M) = \sum_{c=1}^{C} \int_{Y_M} p(Y_M, \omega_c) log \frac{p(Y_M, \omega_c)}{p(Y_M)P(\omega_c)} dx$$

  - The mutual information between the feature vector and the class label $I(Y_M; C)$ measures the amount by which the uncertainty in the class $H(C)$ is decreased by knowledge of the feature vector $H(C|Y_M)$, where $H(\cdot)$ is the entropy function
  - Note that mutual information requires the computation of the multivariate densities $p(Y_M)$ and $p(Y_M , \omega_c)$, which is ill-posed for high-dimensional spaces

Filter approaches



Basic ideas of the filter approaches (examples)



- Correlation Coefficient: Evaluates the linear relationship between numerical features and the target variable. High absolute correlation values indicate strong relationships.
- Chi-Square Test: Assesses the independence between categorical features and the target variable. Features with low p-values are considered significant.
- ANOVA (Analysis of Variance): Determines whether there are statistically significant differences between the means of numerical features across different groups.
- Mutual Information: Measures the mutual dependence between features and the target variable, capturing any kind of relationship (not just linear).
- Low variance filter: Eliminates features with low variance, assuming they have little information content.

Filtering procedure

- Order the features (individual feature ranking or nested subsets of features) based on either the correlation or the information theoretic measures
- Select M features
- Handling of redundant features

Minimum-redundancy-maximum-relevance (mRMR) feature selection

- Standard approach only uses the maximum-relevance selection: features with the strongest correlation to the classification variable
- However, preference for features that are mutually far away from each other while still having "high" correlation to the classification variable
- → Minimum Redundancy Maximum Relevance (mRMR) selection
- → found to be more powerful than the maximum relevance selection
- can use either mutual information, correlation, or distance/similarity scores to select features
- The aim is to penalize a feature's relevancy by its redundancy in the presence of the other selected features.

mRMR criterion

- The relevance of a feature set *S* for the class *c* is defined by the average value of all mutual information values between the individual feature f_i and the class *c* :

$$D(S,c) = \frac{1}{|S|} \sum_{f_i \in S} I(f_i; c)$$

- The redundancy of all features in the set *S* is the average value of all mutual information values between the feature f_i and the feature f_j:

$$R(S) = \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i; f_j)$$

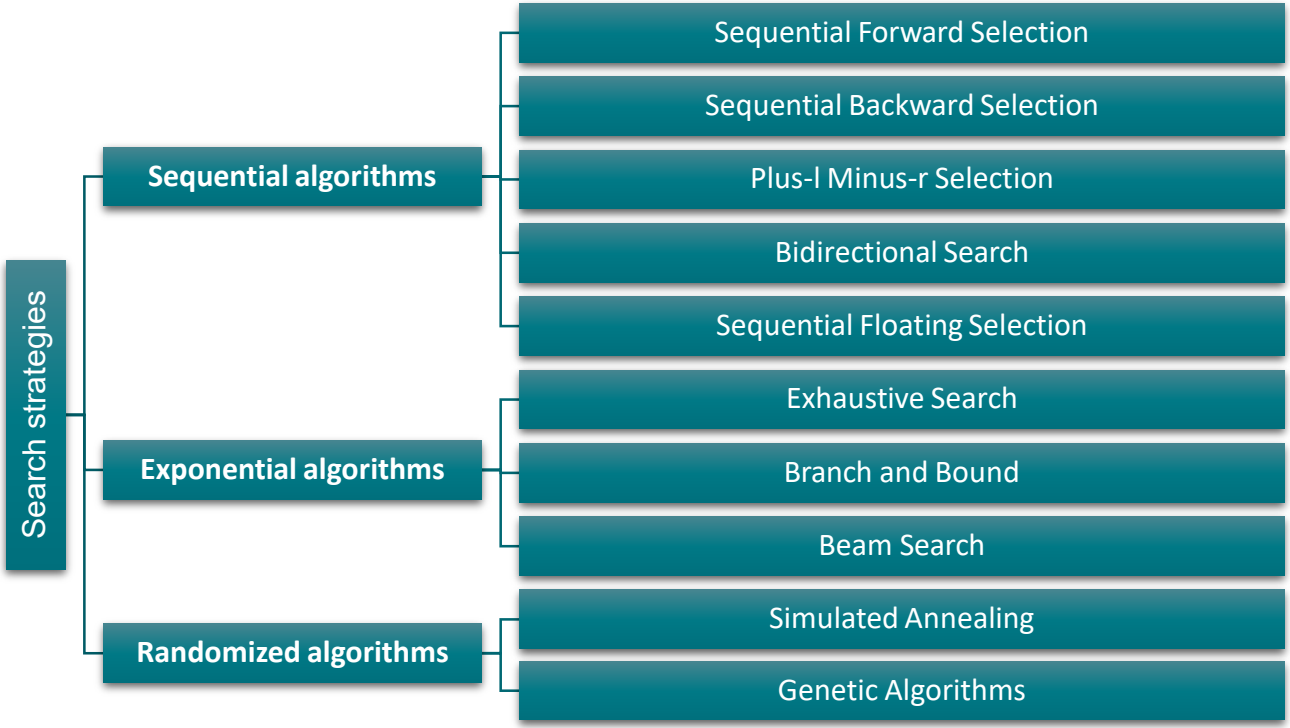
mRMR

- The mRMR criterion is a combination of the two measures:

$$\text{mRMR} = \max_S \left[\frac{1}{|S|} \sum_{f_i \in S} I(f_i; c) - \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i; f_j) \right]$$

Peng, Hanchuan, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." *IEEE Transactions on pattern analysis and machine intelligence* 27.8 (2005): 1226-1238.

Different search strategies

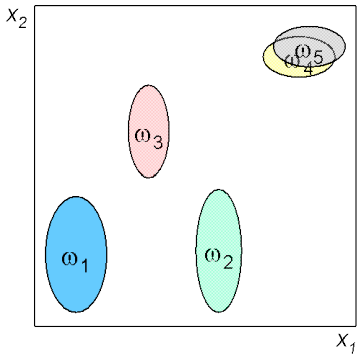


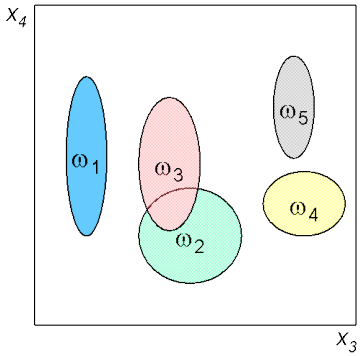
Naïve sequential feature selection

- Evaluating each individual feature separately and select the best M features

→does not account for feature dependence

→Example



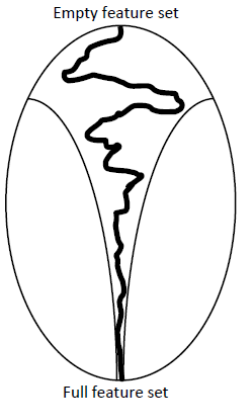


Source: Gutierrez-Osuna, 2013

Sequential forward selection (SFS)

- Starting from the empty set, sequentially add the feature x^+ that maximizes $J(Y_k + x^+)$ when combined with the features Y_k that have already been selected

1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \arg\max_{x \notin Y_k} J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+;\ k = k + 1$
4. Go to 2



- SFS performs best when the optimal subset is small
  - When the search is near the empty set, a large number of states can be pot[illegible], evaluated
  - Towards the full set, the region examined by SFS is narrower since most features have already been selected
  - The main disadvantage of SFS is that it is unable to remove features that become obsolete after the addition of other features

Source: Gutierrez-Osuna, 2013

Example

- **Run SFS to completion for the following objective function**

  where x_k are indicator variables, which indicate whether the k^{th} feature has been selected $x_k = 1$ or not $x_k = 0$

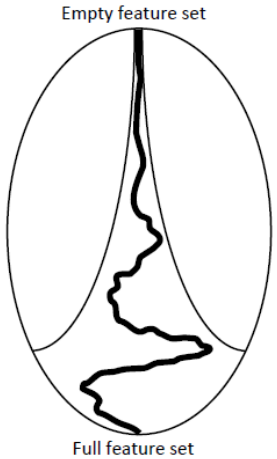
$$J(X) = -2x_1x_2 + 3x_1 + 5x_2 - 2x_1x_2x_3 + 7x_3 + 4x_4 - 2x_1x_2x_3x_4$$

Source: Gutierrez-Osuna, 2013

Sequential backward selection (SBS)

- Starting from the full set, sequentially remove the feature x^- that least reduces the value of the objective function $J(Y - x^-)$
  - Removing a feature may actually increase the objective function $J(Y_k - x^-) > J(Y_k)$; such functions are said to be non-monotonic

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \arg\max_{x \in Y_k} J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-;\ k = k + 1$
4. Go to 2



- SBS works best when the optimal feature subset is large, sir spends most of its time visiting large subsets
- The main limitation of SBS is its inability to reevaluate the us a feature after it has been discarded

Source: Gutierrez-Osuna, 2013

Plus-L minus-R selection (LRS)

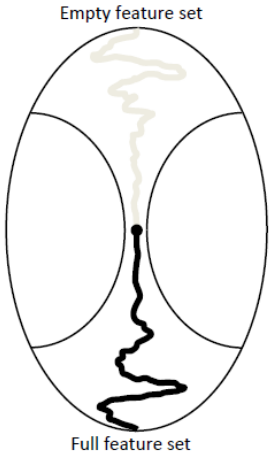
- A generalization of SFS and SBS
  - If L>R, LRS starts from the empty set and repeatedly adds L features and removes R features
  - If L<R, LRS starts from the full set and repeatedly removes R features followed by L additions
  - LRS attempts to compensate for the weaknesses of SFS and SBS with some backtracking capabilities
  - Its main limitation is the lack of a theory to help predict the optimal values of L and R

1. If L>R then $Y_0 = \{\emptyset\}$
   else $Y_0 = X$; go to step 3
2. Repeat L times
   $$x^+ = \arg\max_{x \notin Y_k} J(Y_k + x)$$
   $$Y_{k+1} = Y_k + x^+; \; k = k + 1$$
3. Repeat R times
   $$x^- = \arg\max_{x \in Y_k} J(Y_k - x)$$
   $$Y_{k+1} = Y_k - x^-; \; k = k + 1$$
4. Go to 2

Source: Gutierrez-Osuna, 2013

Bi-directional Search (BDS)

- BDS is a parallel implementation of SFS and SBS
  - SFS is performed from the empty set
  - SBS is performed from the full set
  - To guarantee that SFS and SBS converge to the same
    - Features already selected by SFS are not removed by
    - Features already removed by SBS are not selected by



1. Start SFS with $Y_F = \{\emptyset\}$
2. Start SBS with $Y_B = X$
3. Select the best feature

$$x^+ = \underset{\substack{x \notin Y_{F_k} \\ x \in F_{B_k}}}{\arg\max} J(Y_{F_k} + x)$$

$$Y_{F_{k+1}} = Y_{F_k} + x^+$$

4. Remove the worst feature

$$x^- = \underset{\substack{x \in Y_{B_k} \\ x \notin Y_{F_{k+1}}}}{\arg\max} J(Y_{B_k} - x)$$

$$Y_{B_{k+1}} = Y_{B_k} - x^-;\ k = k + 1$$

5. Go to 2

Source: Gutierrez-Osuna, 2013

Filters vs. wrappers

- Filters
  - Fast execution (+): Filters generally involve a non-iterative computation on the dataset, which can execute much faster than a classifier training session
  - Generality (+): Since filters evaluate the intrinsic properties of the data, rather than their interactions with a particular classifier, their results exhibit more generality: the solution will be “good” for a larger family of classifiers
  - Tendency to select large subsets (-): Since the filter objective functions are generally monotonic, the filter tends to select the full feature set as the optimal solution. This forces the user to select an arbitrary cutoff on the number of features to be selected

- Wrappers
  - Accuracy (+): wrappers generally achieve better recognition rates than filters since they are tuned to the specific interactions between the classifier and the dataset
  - Ability to generalize (+): wrappers have a mechanism to avoid overfitting, since they typically use cross-validation measures of predictive accuracy
  - Slow execution (-): since the wrapper must train a classifier for each feature subset (or several classifiers if cross-validation is used), the method can become unfeasible for computationally intensive methods
  - Lack of generality (-): the solution lacks generality since it is tied to the bias of the classifier used in the evaluation function. The “optimal” feature subset will be specific to the classifier under consideration

Source: Gutierrez-Osuna, 2013

Embedded Approaches

- Embedded methods perform feature selection and function estimation simultaneously – feature selection is embedded within the machine learning algorithm
- There are several approaches to embedded methods, one of which is the Lasso: (lest absolute shrinkage and selection operator)
- Lasso is a method for linear regression that solves:

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i - b)^2 + \lambda \|w\|_1$$

With $\|w\|_1 = \sum_{j=1}^{d} |w^{(j)}|$ as L_1 norm

And more generally for, $1 < p < \infty$, $\|w\|_p = (\sum_{j=1}^{d} |w^{(j)}|^p)^{\frac{1}{p}}$ for $p \geq 1$

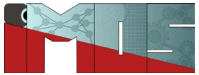
Embedded methods cont.

- Key observation about the L_1 –penalized least square solution is that $\hat{w}$ is sparse, meaning that the method automatically selects the relevant features

Feature Importance from Models

- Many machine learning models offer built-in mechanisms to assess feature importance, which can be used for feature selection.
- **Tree-Based Models**: Algorithms like **Random Forests**, **Gradient Boosting Machines** (e.g., XGBoost, LightGBM, CatBoost), and **Decision Trees** compute feature importance based on how much each feature decreases impurity or improves split quality.
  - **Gini Importance (Mean Decrease in Impurity)**: Measures the total reduction of impurity brought by each feature across all trees in the model.
  - **Permutation Importance**: Evaluates the decrease in model performance when a feature's values are randomly shuffled, breaking the relationship between the feature and the target.
- **Linear Models**: In models like linear or logistic regression, the absolute values of the coefficients indicate feature importance.
  - **Regularization Techniques**: Applying L1 regularization (**Lasso Regression**) can shrink less important feature coefficients to zero, effectively performing feature selection.
- **Usage in Feature Selection**:
  - **Rank Features**: Order features based on their importance scores from the model.
  - **Select Top Features**: Choose a subset of features with the highest importance scores.
  - **Iterative Refinement**: Retrain the model using the selected features and reassess feature importance.

Partial Dependence Plots (PDPs) and Individual Conditional Expectation (ICE) Plots



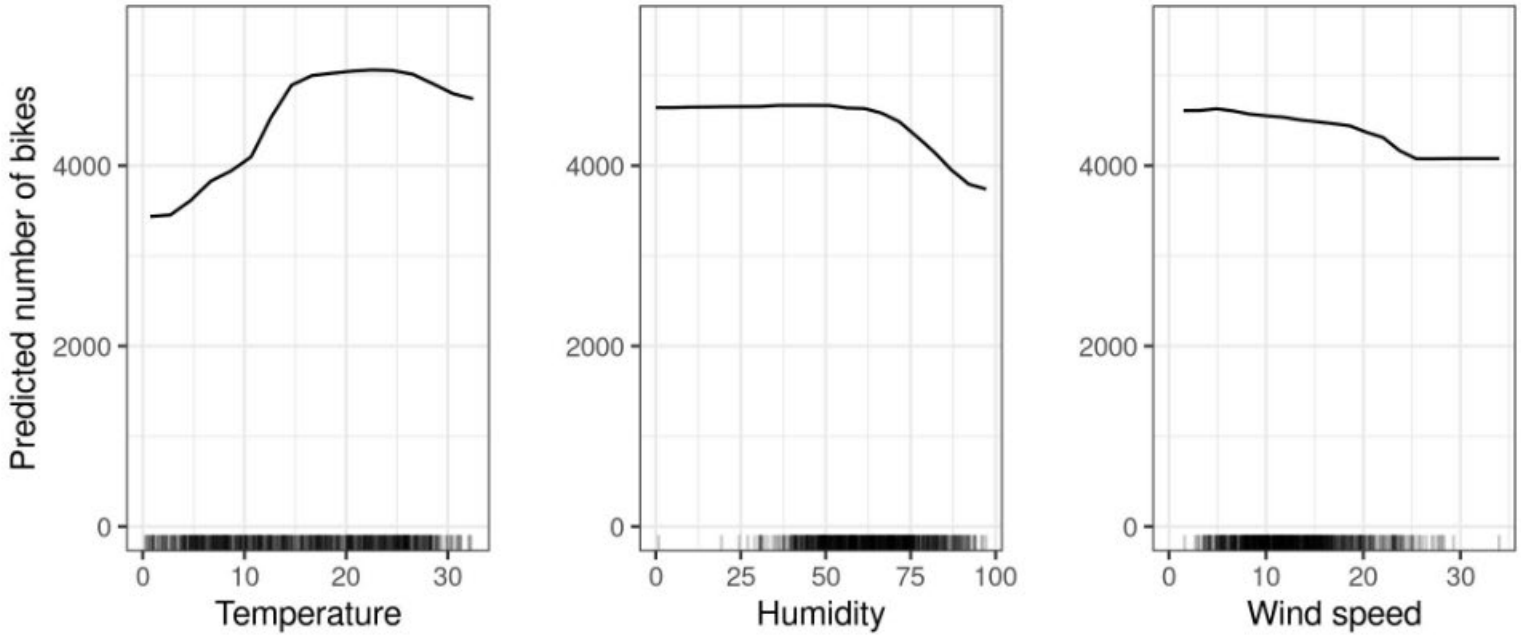
- These plots help visualize the relationship between features and the predicted outcome.
- **Usage in Feature Selection**:
  - **Analyze Feature Impact**: Use PDPs to assess the average effect of a feature on predictions.
  - **Detect Non-Linear Relationships**: Identify features with strong, consistent effects.
  - **Select Features**: Choose features that show significant influence in the plots.

Partial Dependence Plot (PDP)

$$\hat{f}_{x_S}(x_s) = \frac{1}{n}\sum_{i=1}^{n}\hat{f}\left(x_S, x_C^i\right)$$

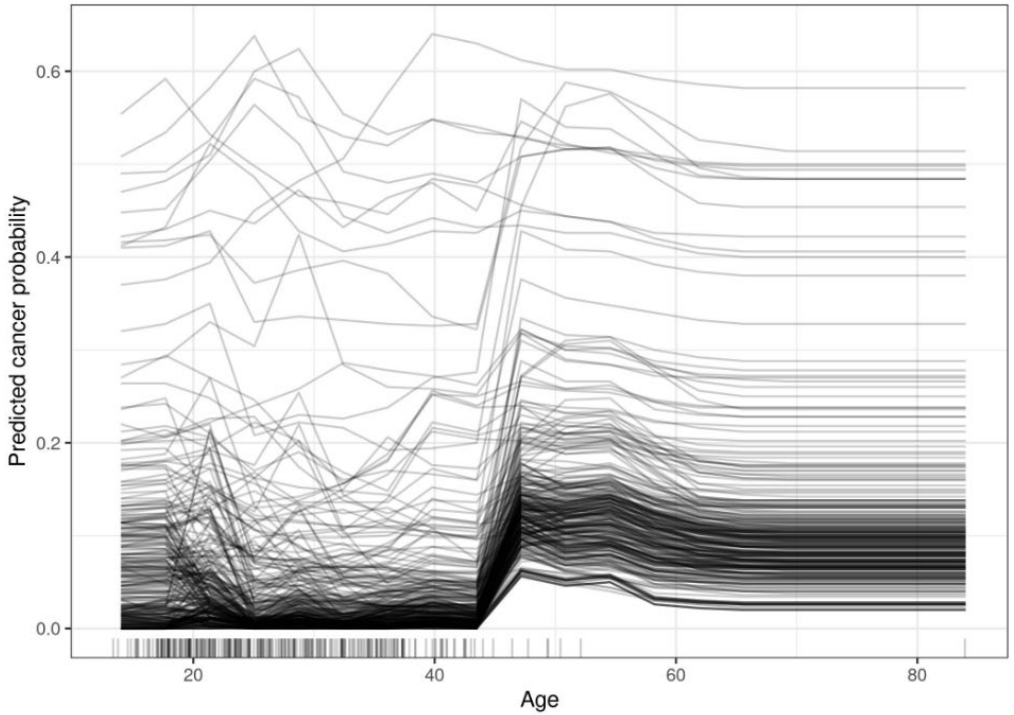
- Where $\hat{f}$ is the model prediction function, x_C^i are actual feature values (not in S) and n is the number points.
- The partial function tells us for given value(s) of features S what the average marginal effect on the prediction is.
- An assumption of the PDP is that the features in C are not correlated with the features in S.
- If this assumption is violated, the averages calculated for the partial dependence plot will include data points that are very unlikely or even impossible

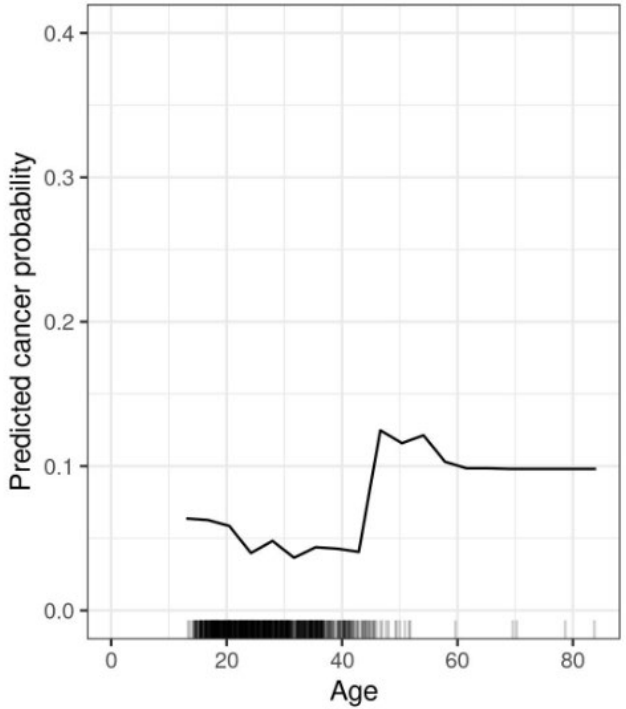
Example PDP plots



Source: interpretable-ml-book

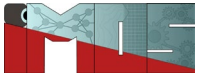
Example ICE (left)/PDP (right) plot





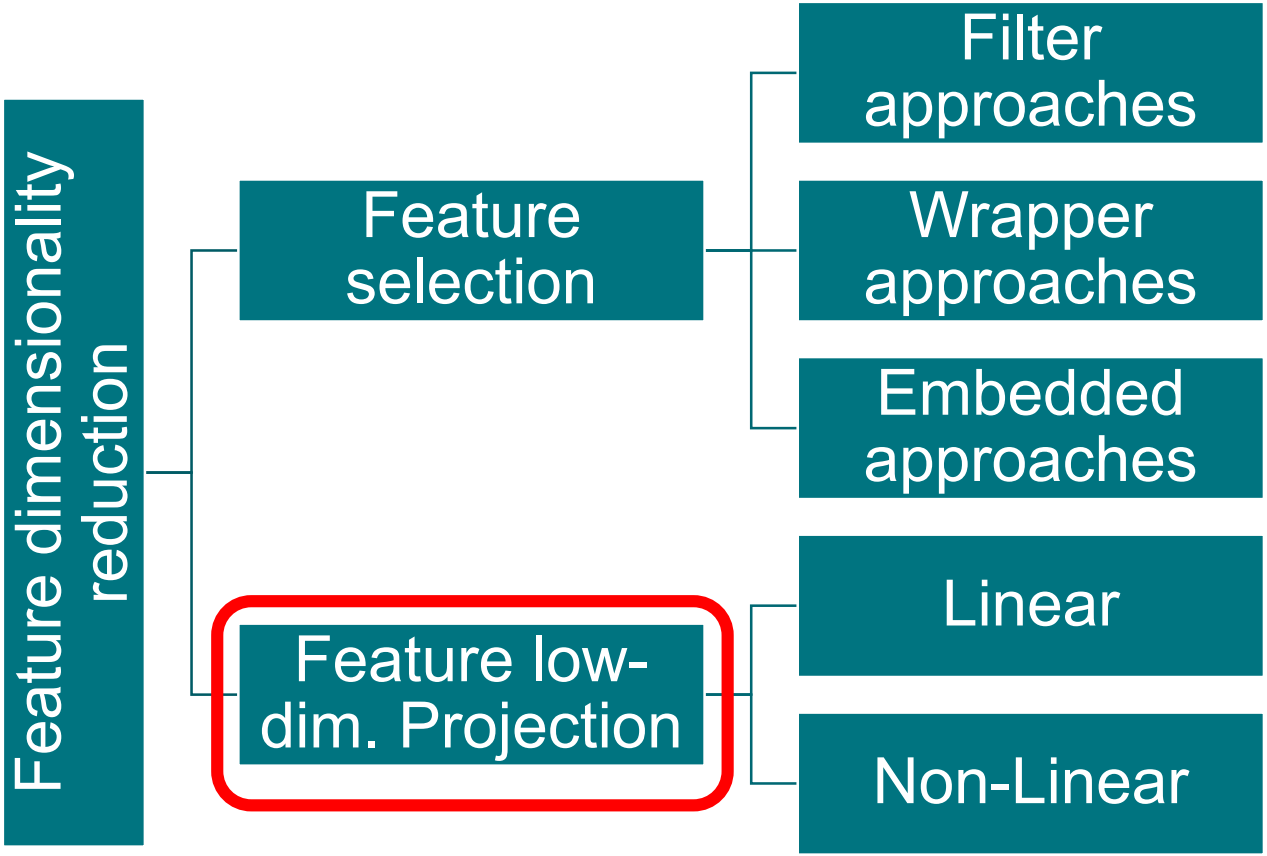
Source: interpretable-ml-book

Feature selection based on explainability methods



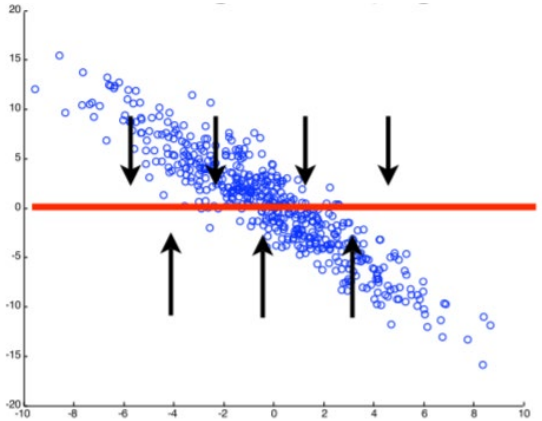
- **Train an Initial Model**: Use all available features to train a baseline model.
- **Compute Explainability Metrics**: Apply methods like SHAP values, permutation importance, or feature importance from models.
- **Rank Features**: Order features based on their importance scores.
- **Select Top Features**: Decide on a threshold (e.g., top 10 features) or use domain knowledge to select features.
- **Retrain the Model**: Build a new model using only the selected features.
- **Evaluate Performance**: Compare the new model's performance against the baseline using metrics like accuracy, precision, recall, or AUC.
- **Iterate if Necessary**: Adjust the feature set based on performance and explainability insights.

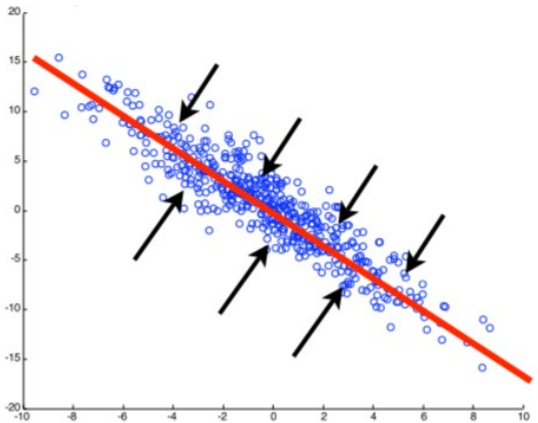
Different approaches to dimensionality reduction



Feature selection vs. Feature low-dim. Projection

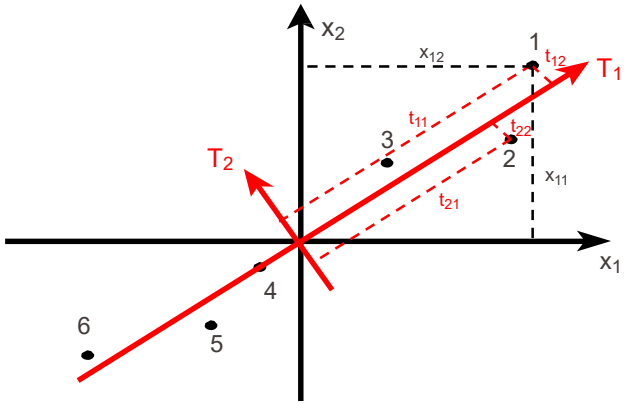
- Removing features → Equivalent to projecting data onto lower-dimensional linear subspace perpendicular to the feature removed





Principal component analysis

- PCA seeks preserve as much of the randomness (variance) in the high-dimensional space as possible
- Projection of the dataset onto a lower dimensional space
- In the direction of maximum variance



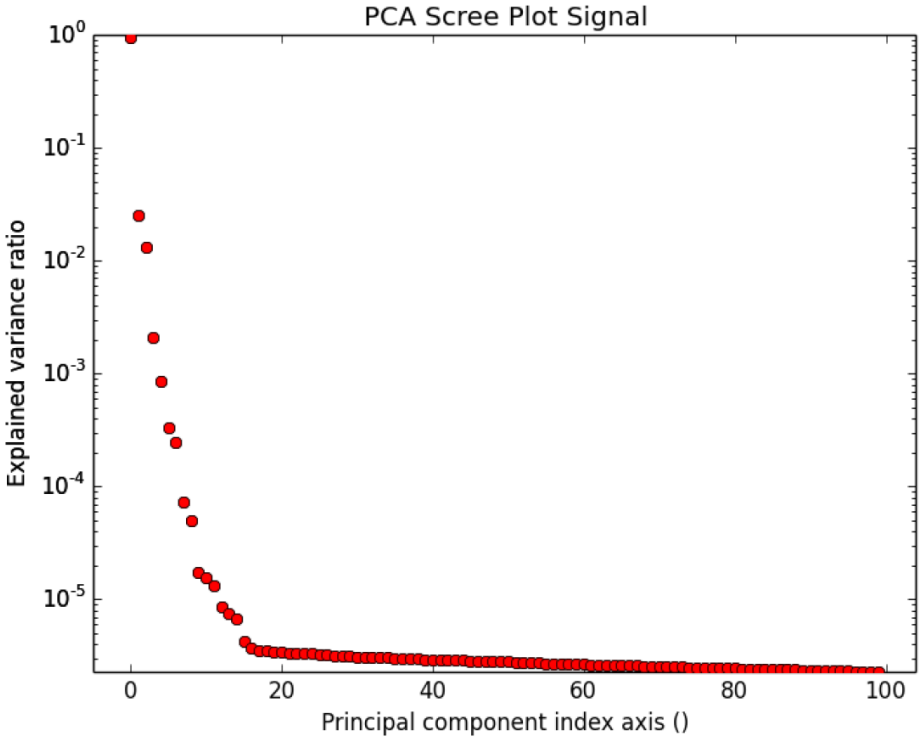
Key concepts of PCA

- Dimensionality Reduction: PCA reduces the number of variables (features) in a dataset while preserving as much variability (information) as possible.
- Principal Components: New uncorrelated variables that are linear combinations of the original variables. Each principal component captures a portion of the total variance in the data.
- Variance Maximization: The first principal component captures the maximum variance, the second captures the next highest variance orthogonal to the first, and so on.
- Orthogonality: Principal components are orthogonal (uncorrelated) to each other, which eliminates redundancy.

How PCA works

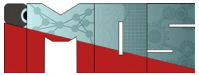
- Standardization:
  - Purpose: Ensures that each feature contributes equally to the analysis.
  - Method: Subtract the mean and divide by the standard deviation for each feature.
- Covariance Matrix Computation:
  - Purpose: Measures how variables change together.
  - Method: Calculate the covariance matrix of the standardized data.
- Eigenvalue and Eigenvector Calculation:
  - Eigenvalues: Indicate the amount of variance captured by each principal component.
  - Eigenvectors: Define the direction of the principal components.
- Selecting Principal Components:
  - Criteria: Choose components with the highest eigenvalues.
  - Methods:
    - Scree Plot: Visualize the eigenvalues to determine the "elbow point.“
    - Explained Variance Ratio: Select components that cumulatively explain a desired amount of total variance (e.g., 95%).
- Transforming the Data:
  - Projection: Multiply the original data by the selected eigenvectors to obtain the principal components.
  - Result: A reduced dataset with uncorrelated features.

Scree test for determining the number of PCs



Source: Hyperspy 2011

Summary of the most important assumptions and limitations of a PCA



- Linearity of the problem: data set is a linear combination of a certain base → Problem solution by means of linear algebra
- PCA uses the eigenvectors of the covariance matrix and finds only independent base vectors assuming a Gaussian probability distribution.
- Assumption that large variances reflect important dynamics. PCA essentially only performs a rotation of the coordinate system in the direction of maximum variance.
- Large variance principal components represent interesting dynamics; small variance components represent noise.
- Role of SNR (Signal to Noise Ratio)
- Main components are orthogonal → Simplification that makes PCA solvable by means of linear algebra.

Applications of PCA

- Data Visualization: Reducing dimensions to 2D or 3D for plotting.
- Noise Reduction: Eliminating less significant components to reduce noise.
- Feature Extraction: Creating new features for machine learning models.
- Data Compression: Reducing storage requirements while retaining essential information.
- Preprocessing Step: Simplifying data before applying other algorithms.

Importance of Q and T^2 Statistics in PCA

- In **Principal Component Analysis (PCA)**, the **Q-statistic** (also known as the **Squared Prediction Error (SPE)**) and **Hotelling's T^2 statistic** are essential metrics used for assessing the fit of observations within the PCA model. They play a crucial role in:
  - **Outlier Detection**
  - **Process Monitoring**
  - **Fault Detection**
  - **Quality Control**
- These statistics help identify observations that deviate significantly from the modeled behavior, enabling analysts to take corrective actions or investigate anomalies.

Calculation of the Q and T²-Statistics

$$T_i^2 = \sum_{j=1}^{r} \frac{t_{sij}^2}{\lambda_j} = \boldsymbol{t}_{\mathrm{si}} \Lambda^{-1} \boldsymbol{t}_{\mathrm{si}}^{\mathrm{T}} = \boldsymbol{x}_i \mathbf{P} \Lambda^{-1} \mathbf{P}^{\mathrm{T}} \boldsymbol{x}_i^{\mathrm{T}}$$

$\boldsymbol{t}_{si} = \boldsymbol{x}_i \mathbf{P}$ Projection of the sample x_i on the principal component

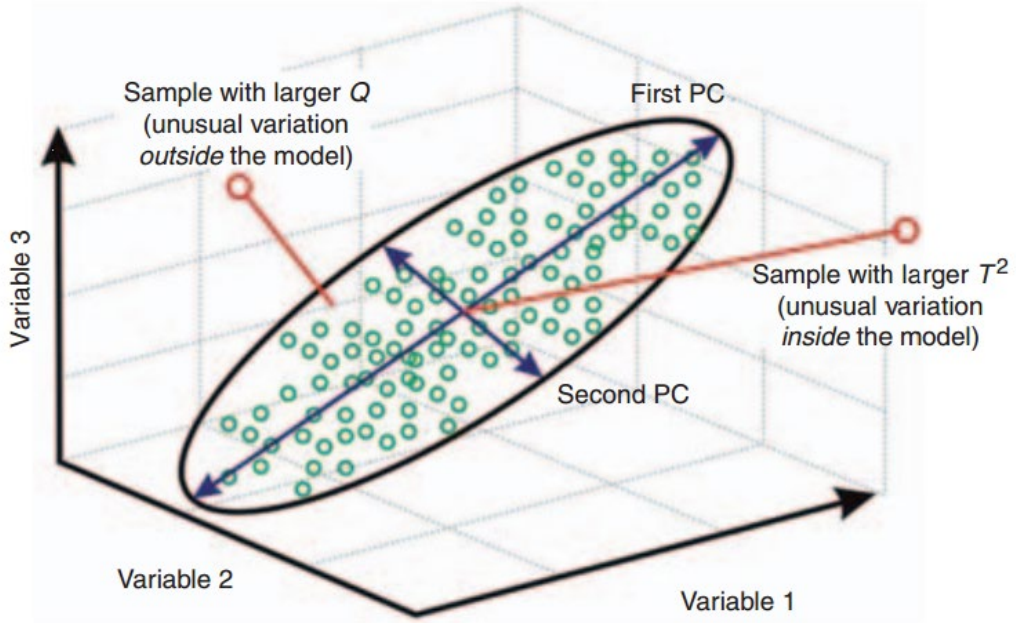
$\boldsymbol{x}_i$ Sample vector representing all the measurements at the point i

$$Q_i = \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^{\mathrm{T}} = \boldsymbol{x}_i \left(\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathrm{T}}\right) x_i^{\mathrm{T}}$$

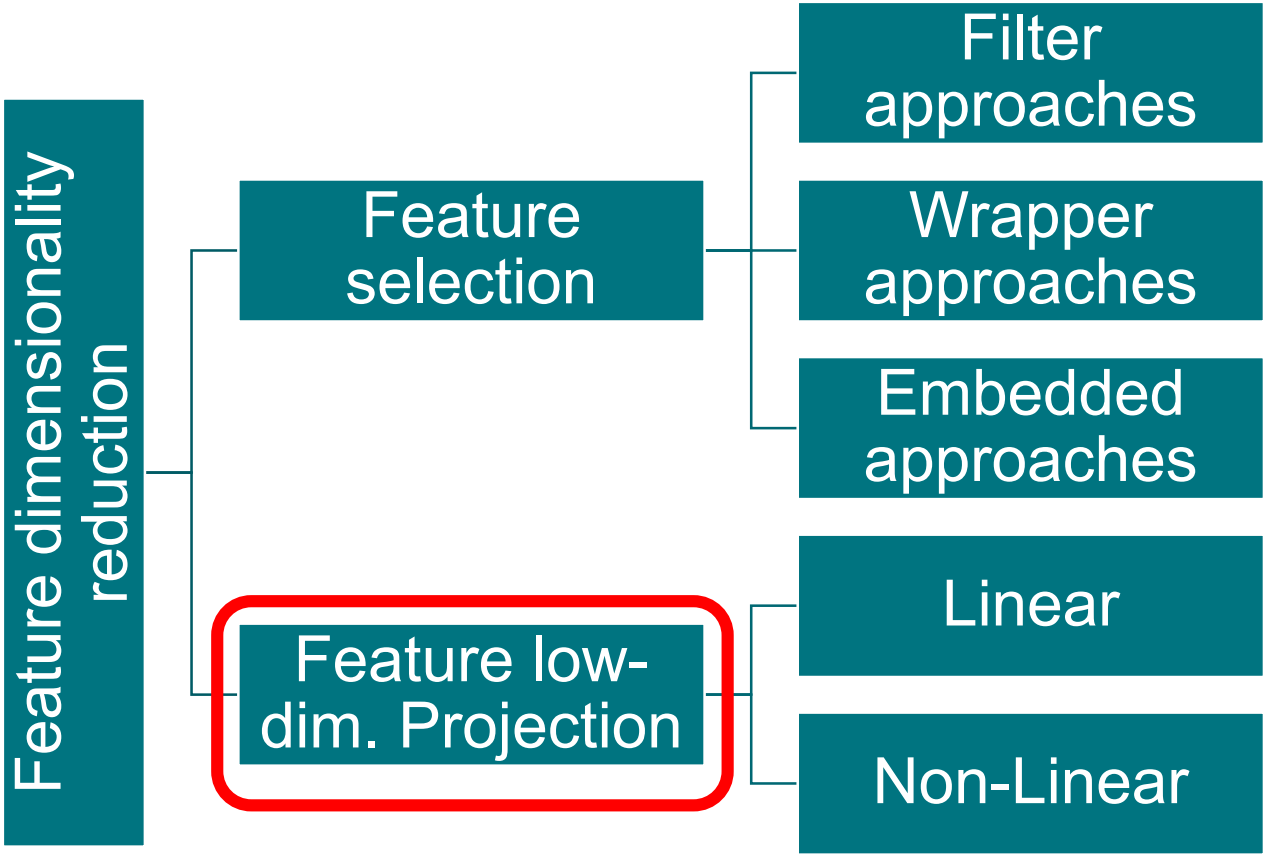
$\tilde{\boldsymbol{x}}_i$ Projection of the sample x_i on the residuals

- Hotelling's T² statistic measures the variation of an observation within the principal component (score) space defined by the PCA model.
- It quantifies how far an observation's scores are from the center (mean) of the model, considering the variability captured by the selected principal components.
- The Q-statistic measures the residual variation of an observation not explained by the PCA model.
- It quantifies the distance between the original observation and its reconstruction from the retained principal components.

Q and T^2-Statistic



Different approaches to dimensionality reduction



Other dimensionality reduction approaches

- Nonlinear PCA (NLPCA)
- Kernel PCA
- Exploratory Projection Pursuit (EPP)
  - EPP seeks an M-dimensional (M=2,3 typically) linear projection of the data that maximizes a measure of «interestingness»
  - Interestingness is measured as departure from multivariate normality
    - This measure is not the variance and is commonly scale-free. In most implementations, it is also affine invariant, so it does not depend on the correlations between features
- Kernel LDA
- T-distributed Stochastic Neighbor Embedding (t-SNE) (for visualization purposes)

Autoencoder (will be considered later)