Initiation à l'apprentissage automatique en science des matériaux

3. Regression

J.-C. Crivello, LINK : jean-claude.crivello@cnrs.fr
C. Barreteau, ICMPE : celine.barreteau@cnrs.fr
S. Junier, ICMPE : sebastien.junier@cnrs.fr

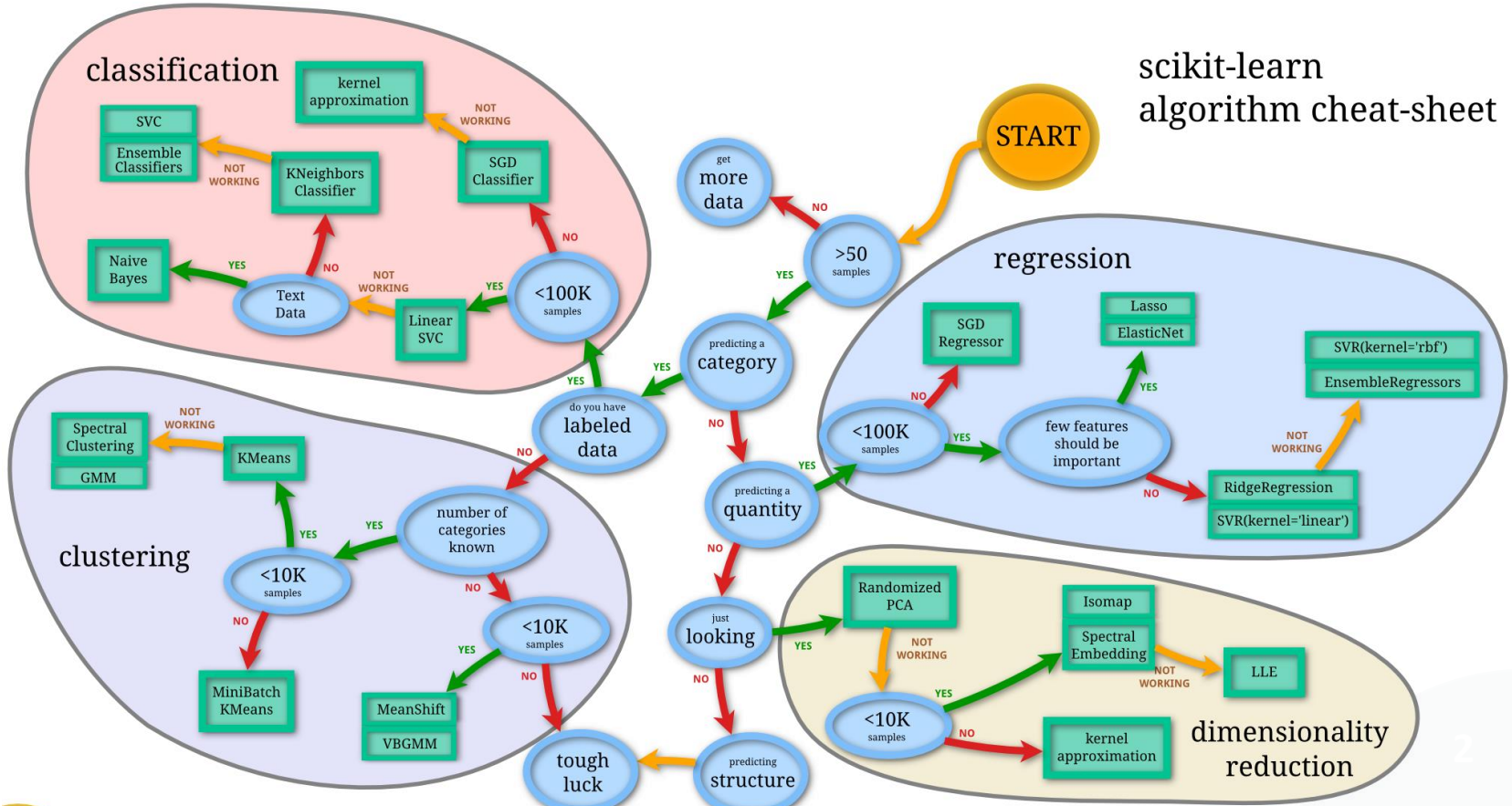
https://link.cnrs.fr/ML/

Several approaches of the data science

- Data-mining algorithms
- Optimization algorithms
- Machine learning algorithms (ML)
  - Reinforcement learning
  - Unsupervised learning (*clustering*)
  - Supervised learning

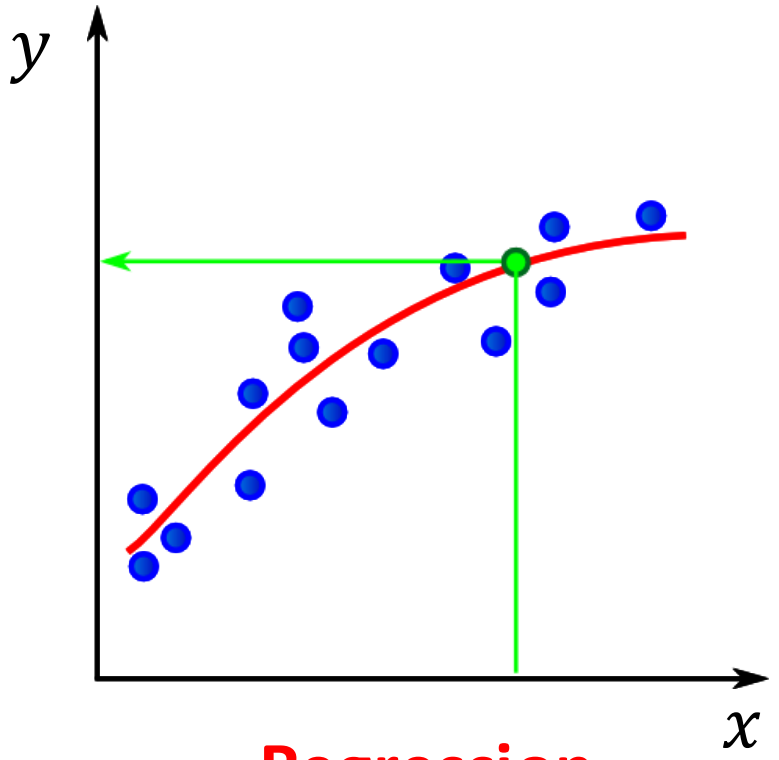
Choosing the right estimator?

https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

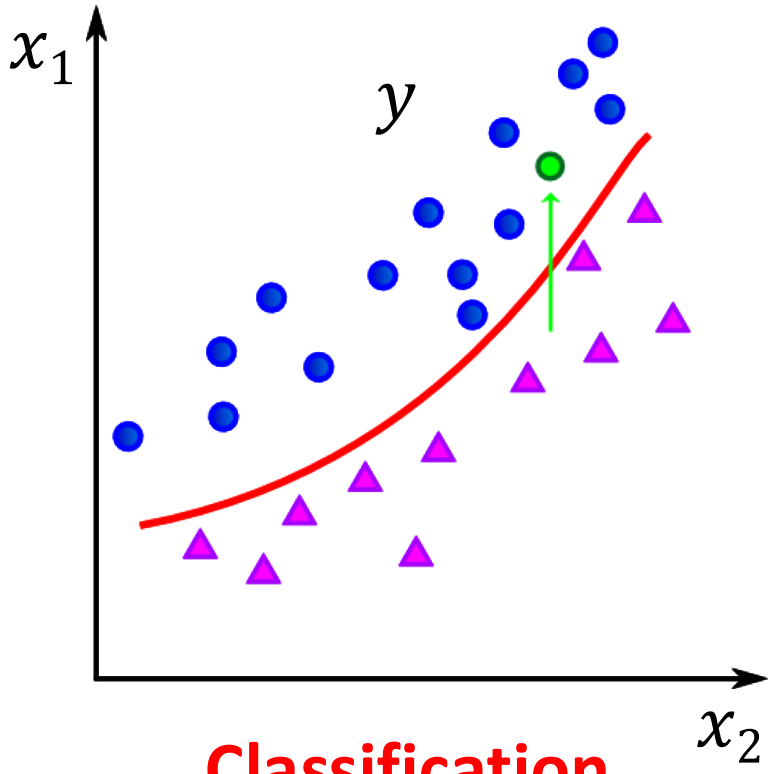


3. Supervised learning

The training dataset often consists of pairs of an input vector (or scalar) and the corresponding output vector (or scalar), the output of the function can be **regression** or **classification**.



Regression

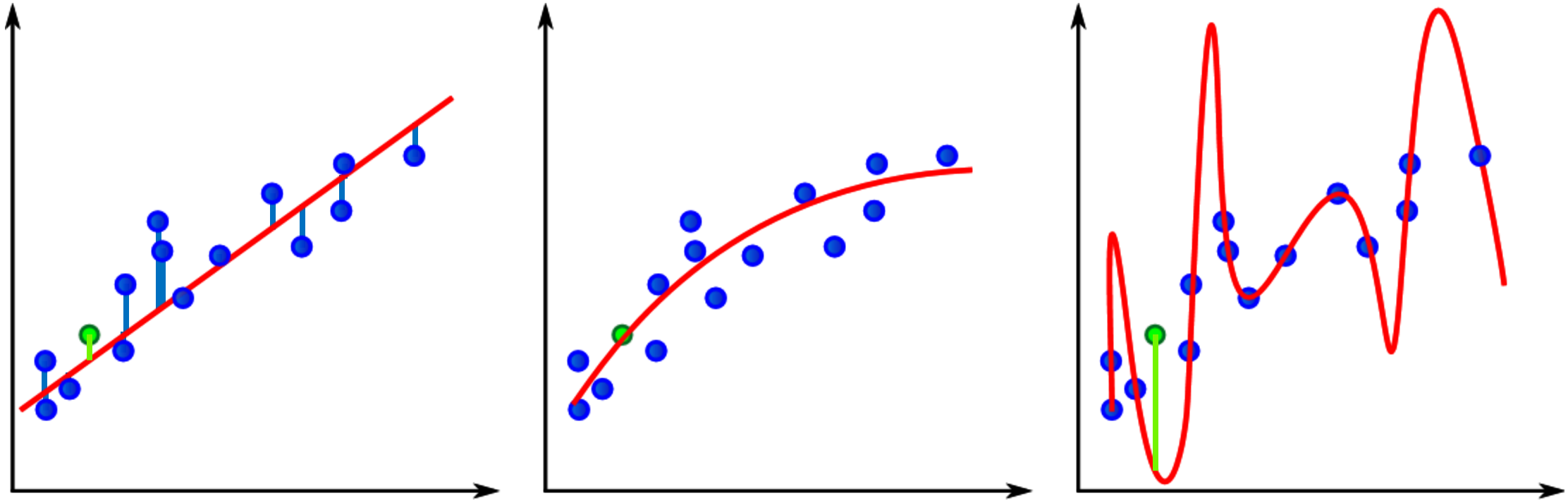


Classification

Biais and variance

Training set (Learning set)

Testing set



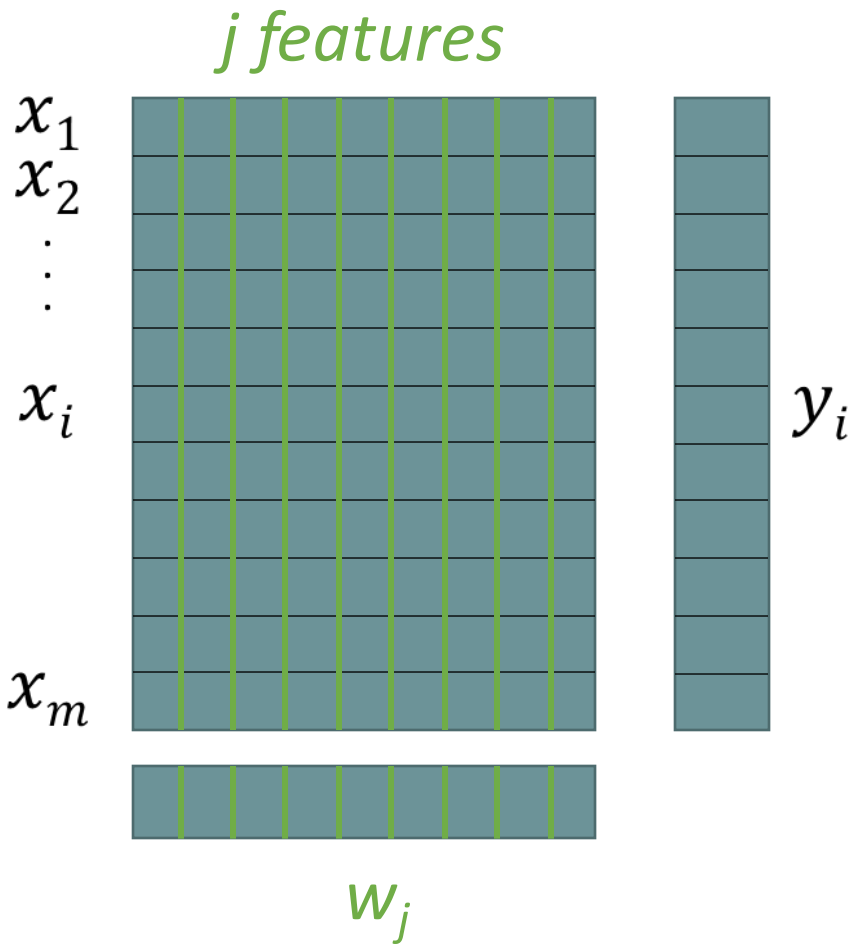
high biais

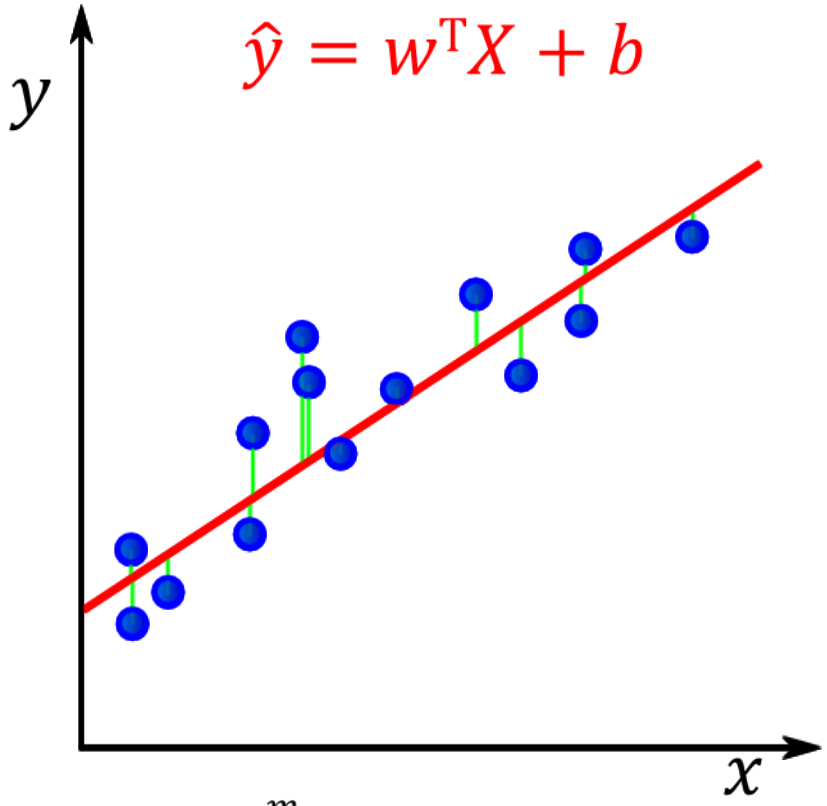
low variance

low biais

high variance

Optimization of the learning

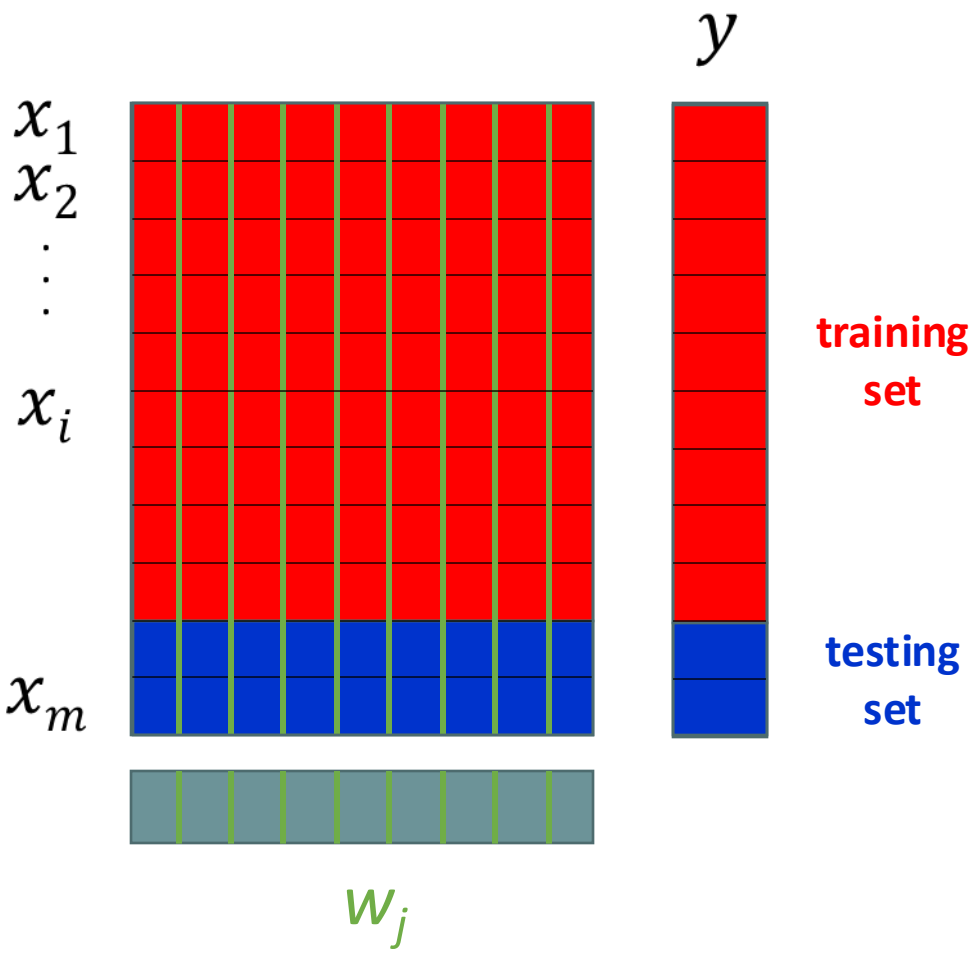


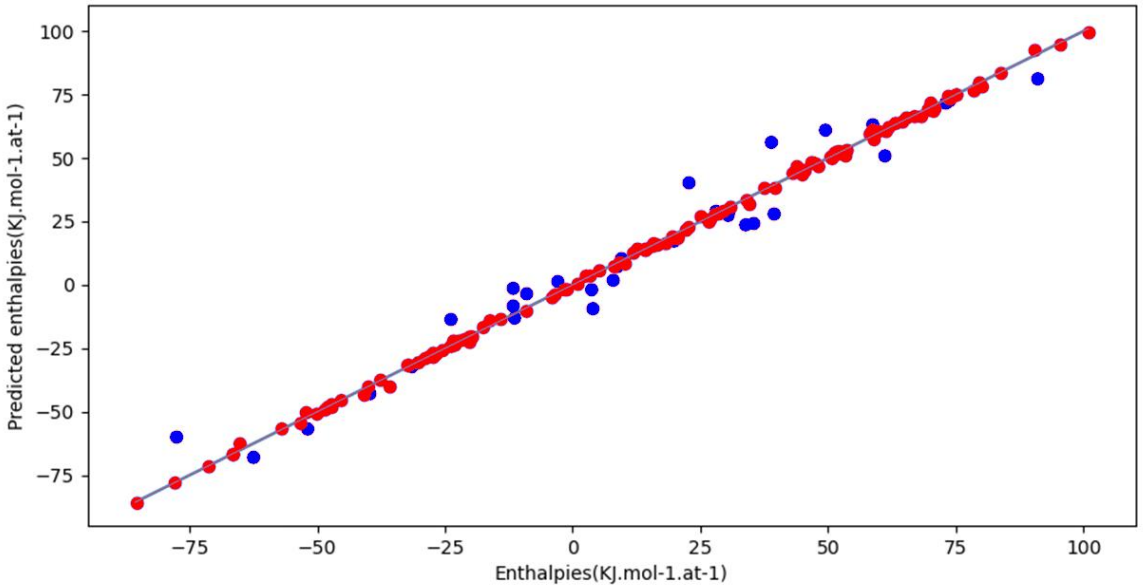


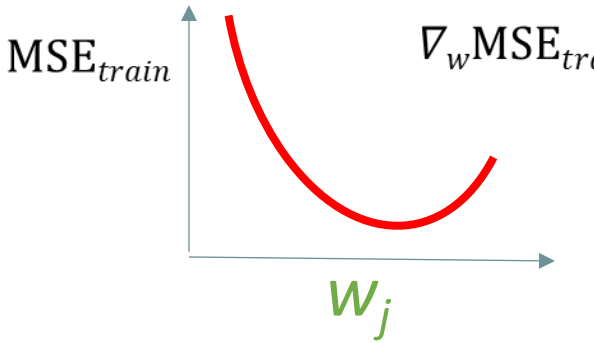
$$\mathrm{SSR} = \sum_{i}^{m} (\hat{y}_i - yi)^2$$

$$R^2 = 1 - \sum \frac{(\hat{y}_i - yi)^2}{(\bar{y}_i - yi)^2}$$

Estimation of the learning





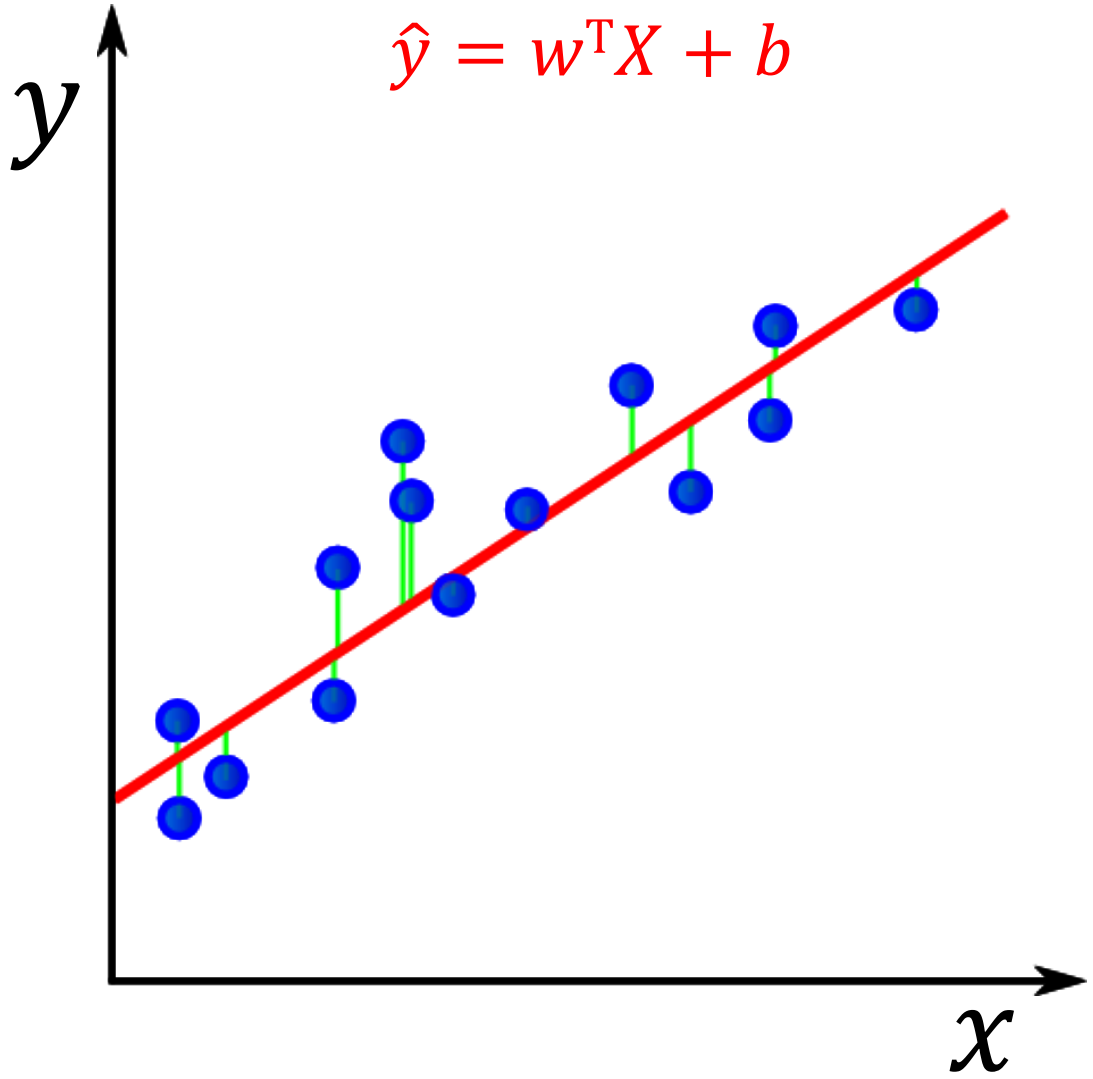


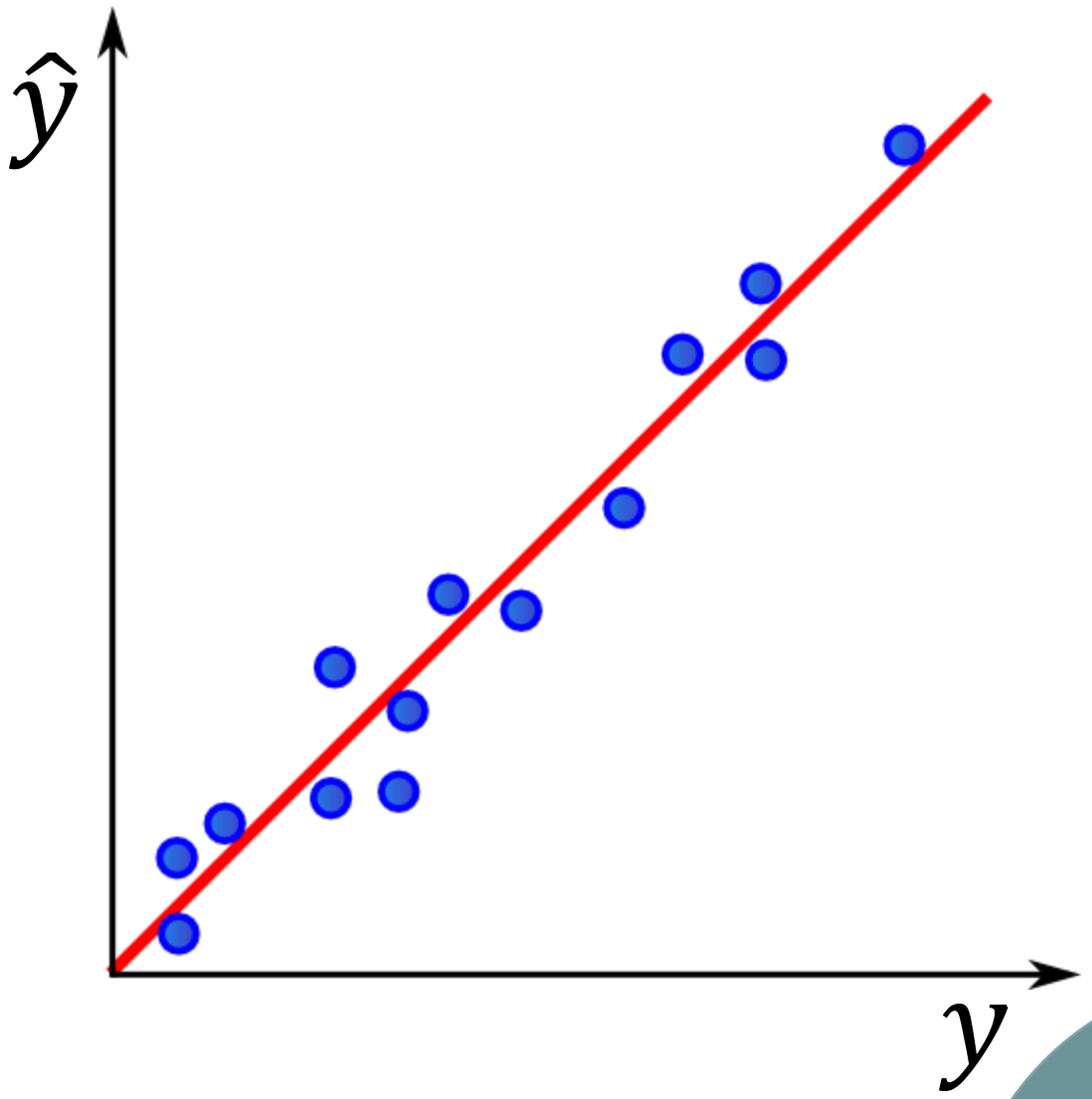
$$\mathrm{MSE}_{test} = \frac{1}{m}\sum_{i}^{m}(\widehat{y_{test}} - yte_{st})^2$$

$$\mathrm{RMSE}_{test} = \sqrt{\mathrm{MSE}_{test}}$$

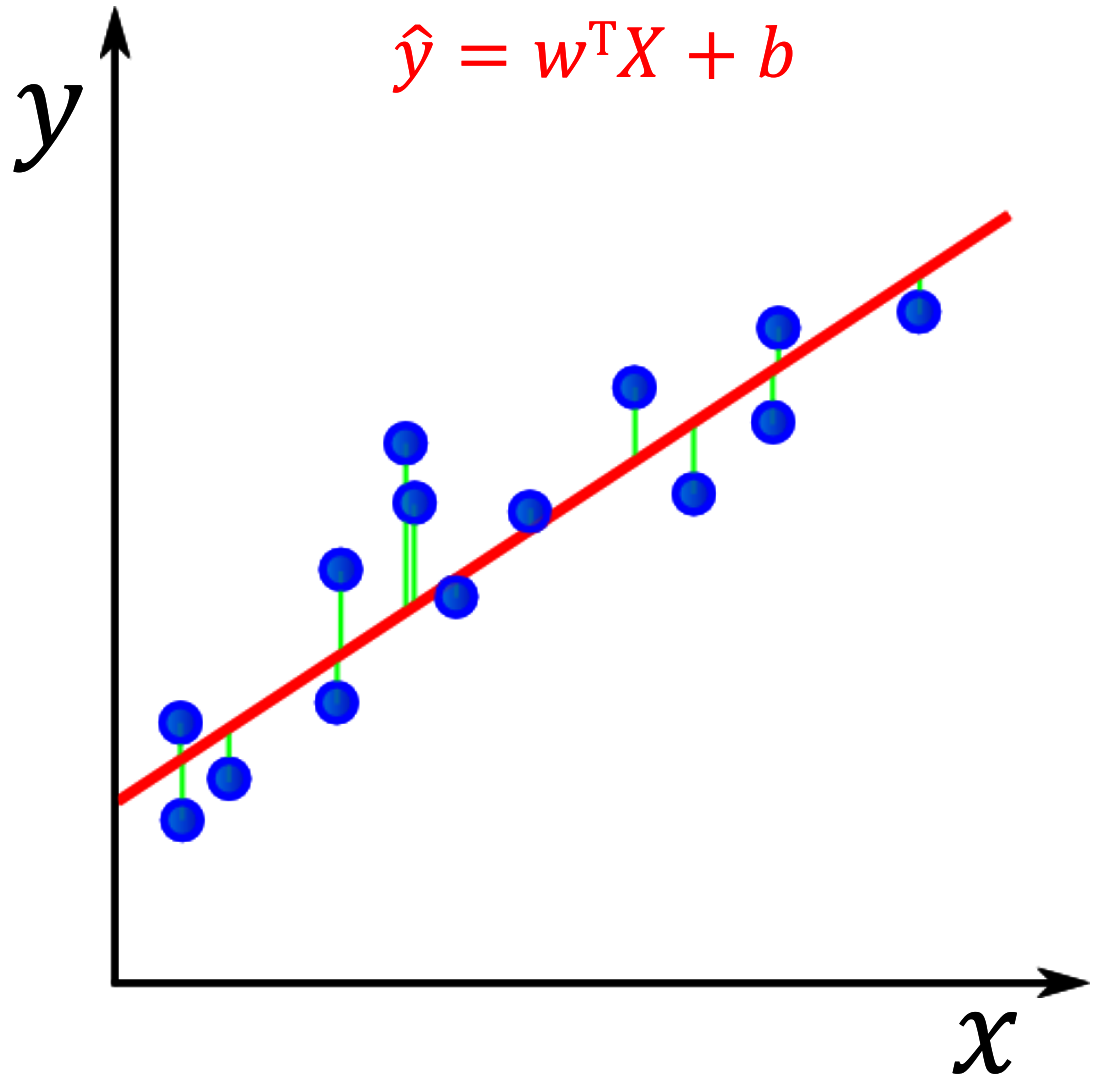
Linear regression

$$\hat{y} = w^{\mathrm{T}}X + b$$





Linear regression



Metrics:

$$R^2 = 1 - \sum \frac{(\hat{y}_i - yi)^2}{(\bar{y}_i - yi)^2}$$

$$\mathrm{MSE} = \frac{1}{m}\sum_{i}^{m}(\hat{y}_i - yi)^2$$

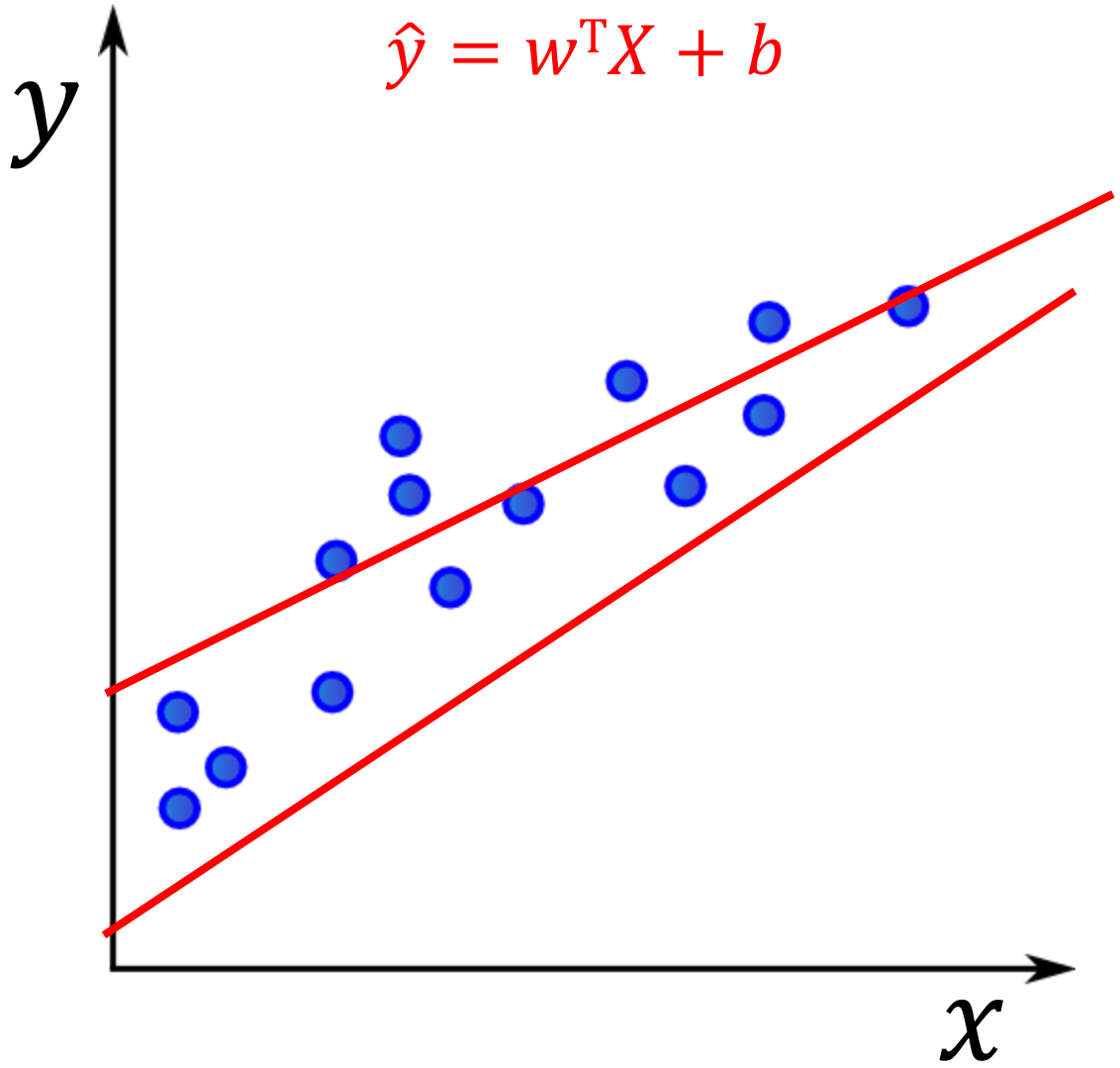
$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

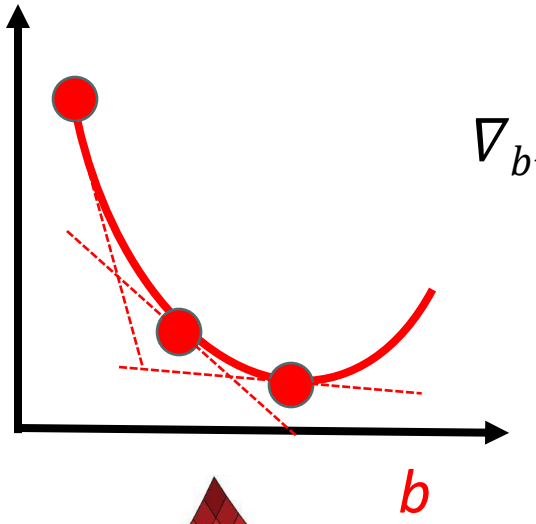
$$\nabla_w \mathrm{MSE}_{train} = 0$$

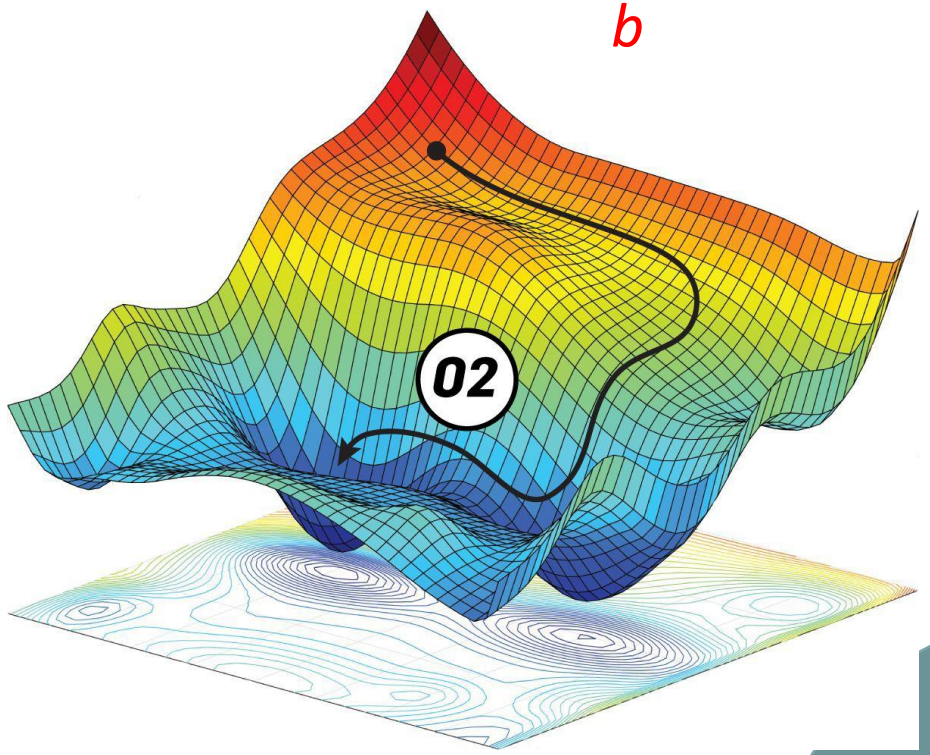
$$\mathrm{MSE} = \frac{1}{m}\sum_{i}^{m} |\hat{y}_i - yi|$$

Gradient Descent

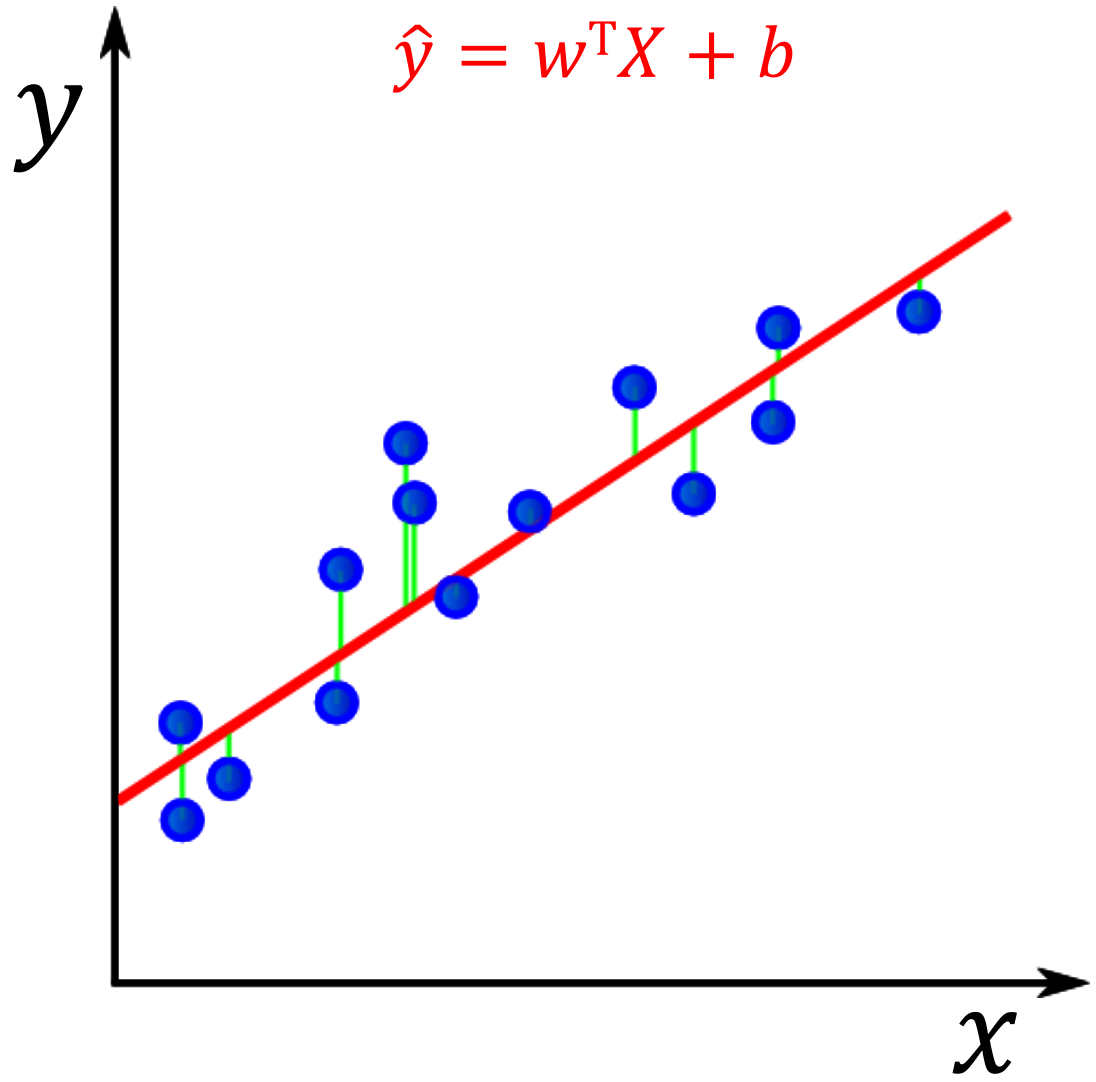
$$\text{Loss function} = \text{SSR} = \sum_{i}^{m} (\hat{y}_i - yi)^2$$







Linear regression



`sklearn.linear_model.LinearRegression`

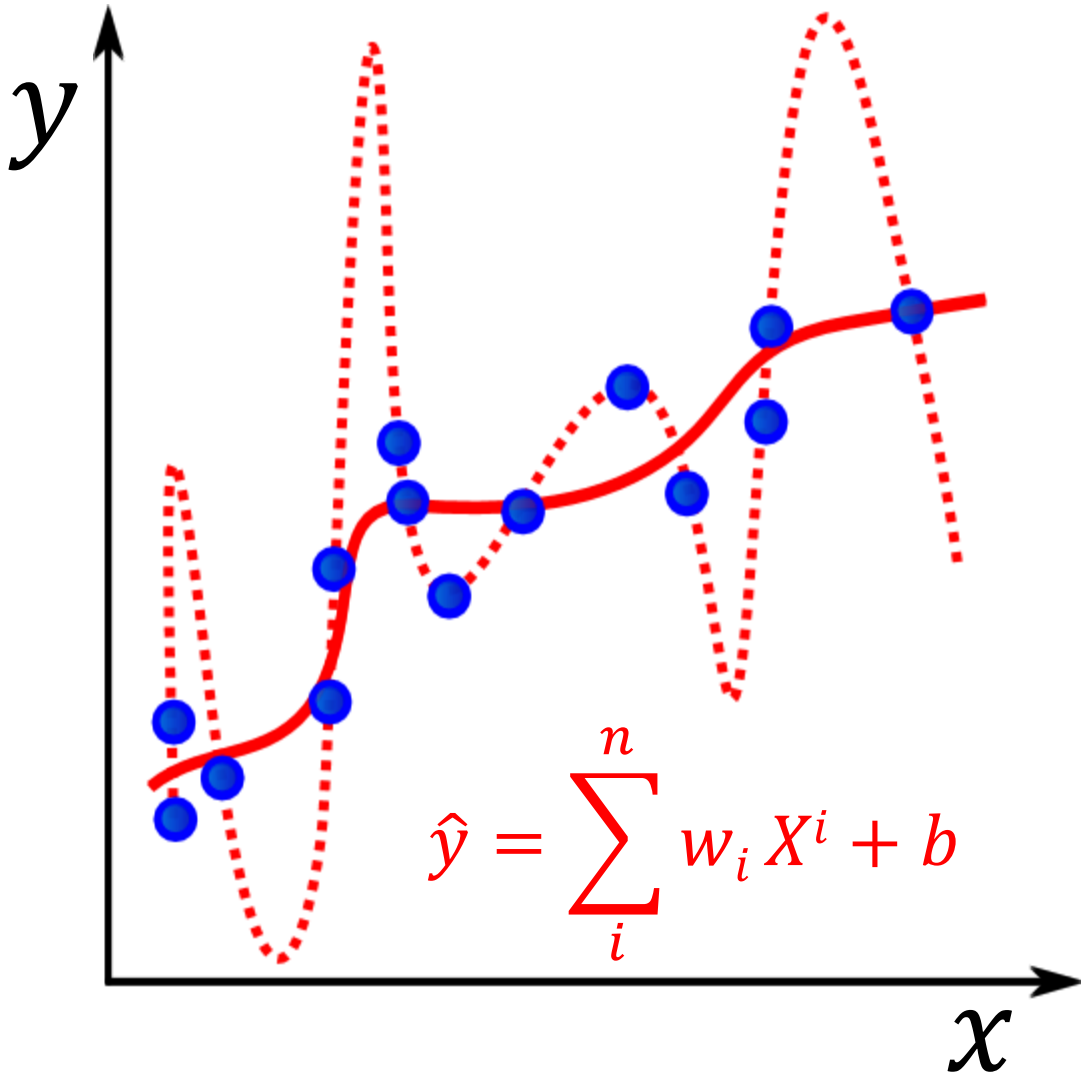
Strengths:

- Simple to perform (inversion of matrix)
- Simple to interpret
- Fast and efficient (if linearity)

Weaknesses:

- Linearity of the model?
- Sensitive to outliers
- Complexity in N^3

Polynomial regression



Strengths:

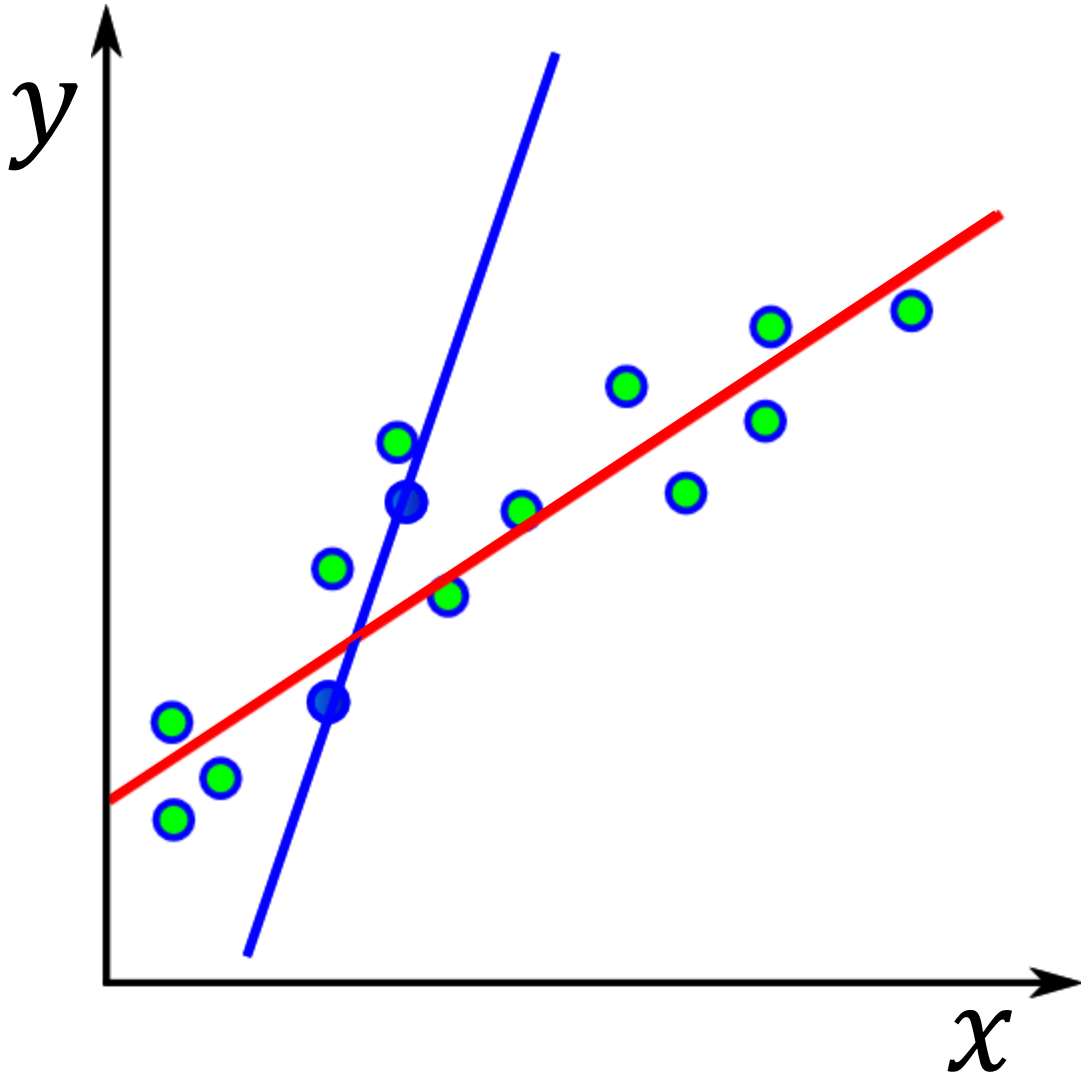
- Simple to perform (inversion of matrix)
- Simple to interpret

Weaknesses:

- Can overfit

Ridge regression

$$\hat{y} = w^{\mathrm{T}}X + b \; +?$$



$$L = \frac{1}{m}\sum_{i}^{m}(\hat{y} - w^{\mathrm{T}}X)^2 - \lambda.w^2$$

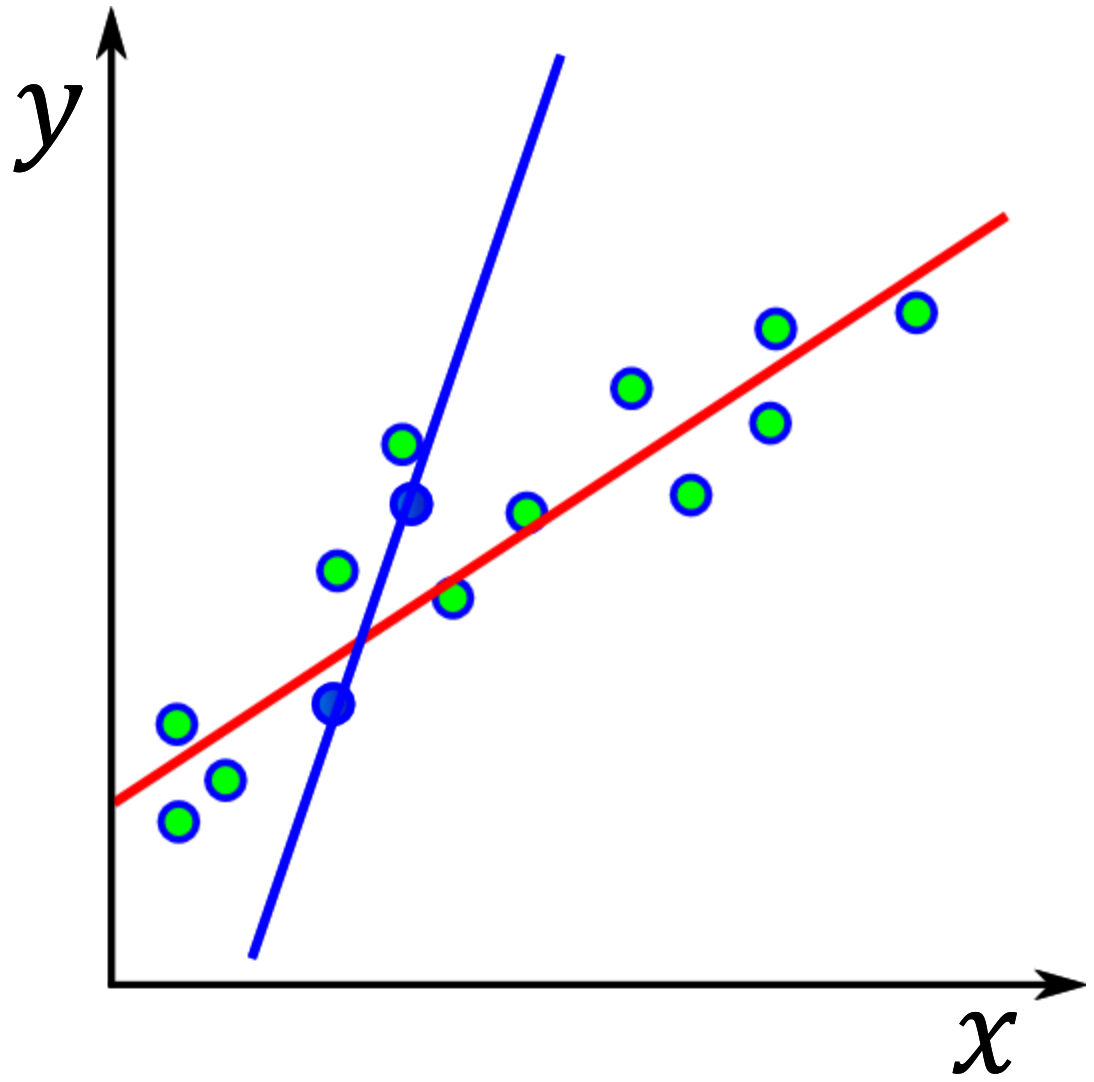
Ridge reduce variance by a penalty: λ (L2 regularization), decided by CV
It shrinks the weights and help to reduce multi-collinearity

Strengths:

- Simple to perform
- Works best when most of the variables are usefull

LASSO regression

$$\hat{y} = w^{\mathrm{T}}X + b \ +?$$



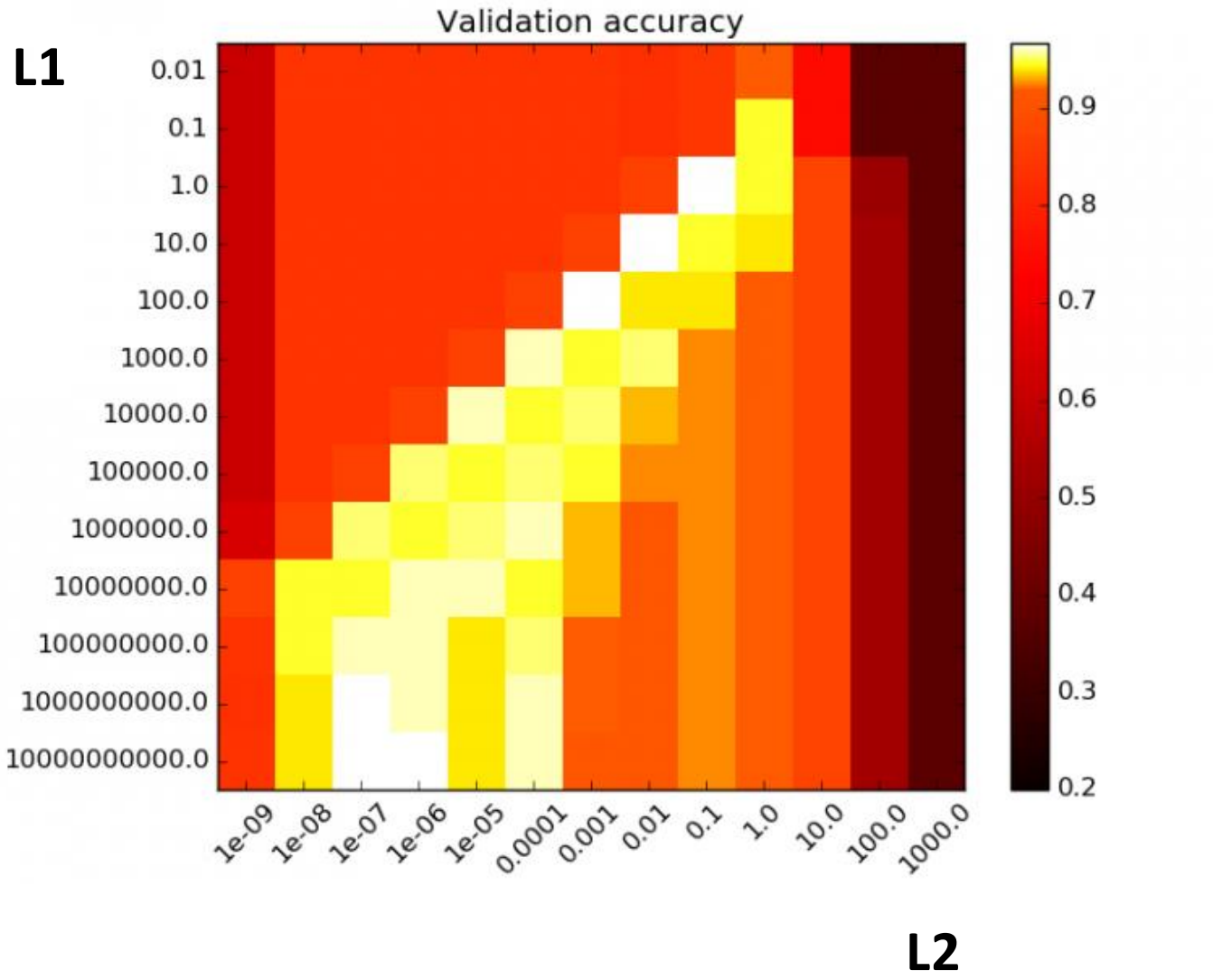
$$L = \frac{1}{m}\sum_{i}^{m}(\hat{y} - w^{\mathrm{T}}X)^2 - \lambda.w$$

Lasso use L1 regularization,
It helps in features selection

Strengths:

- Simple to perform
- Lasso works best when model contains a lot of useless variables

ElasticNet Regression



$$L = \frac{1}{m}\sum_i^m (\hat{y} - w^{\mathrm{T}}X)^2 - \lambda_1 . w - \lambda_2 . w^2$$

λ_1 and λ_2 parameters, optimized by <u>CV grid search</u>

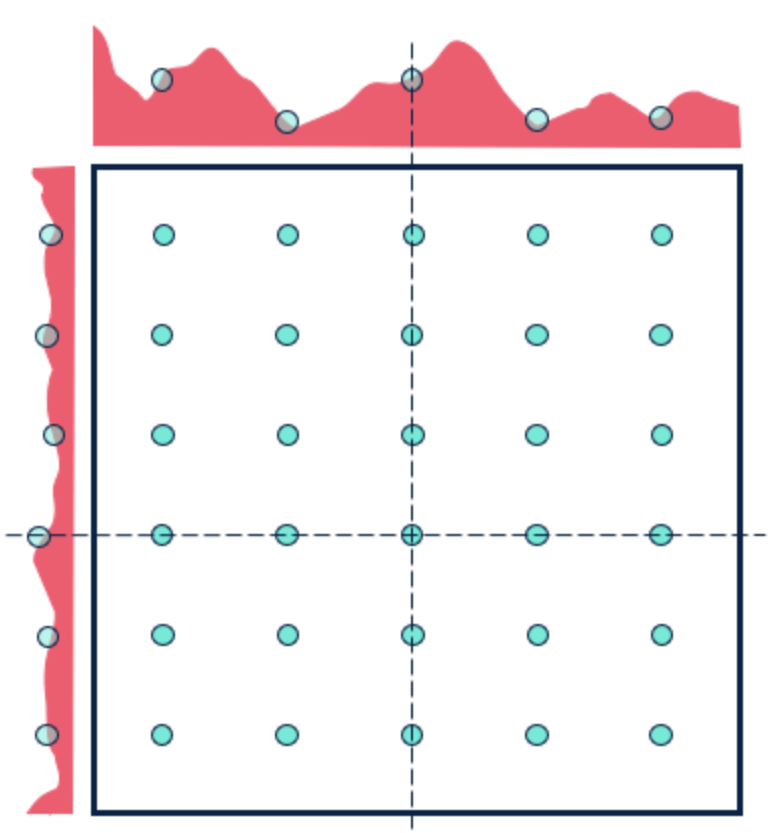
Strengths:

- Combines both Ridge and Lasso
- High quality regression if large dataset of x_i linked

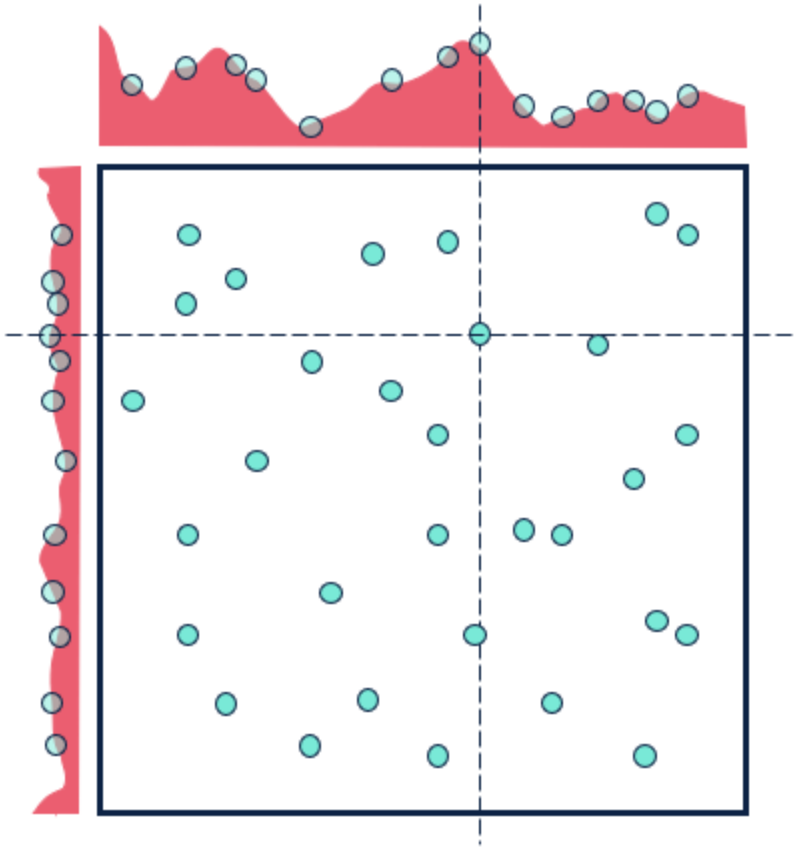
Weaknesses:

- Still a linear method

About Grid search



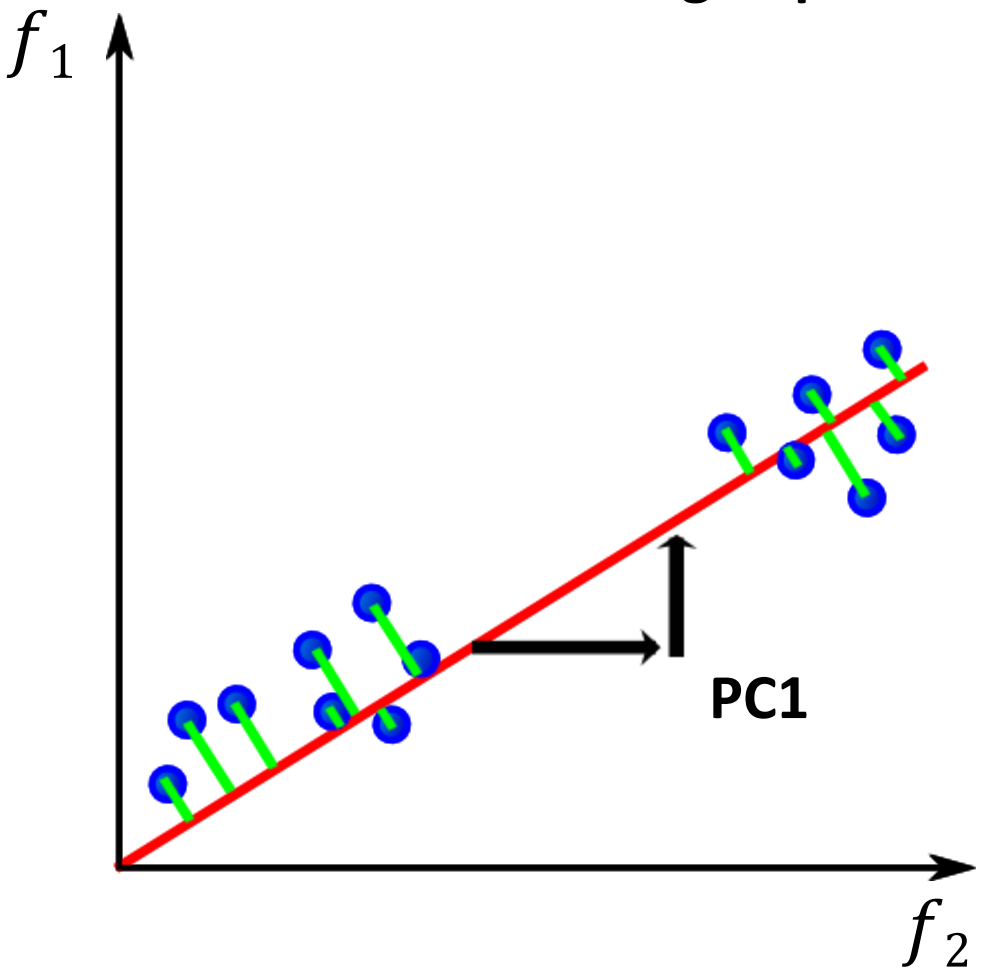
Grid Search

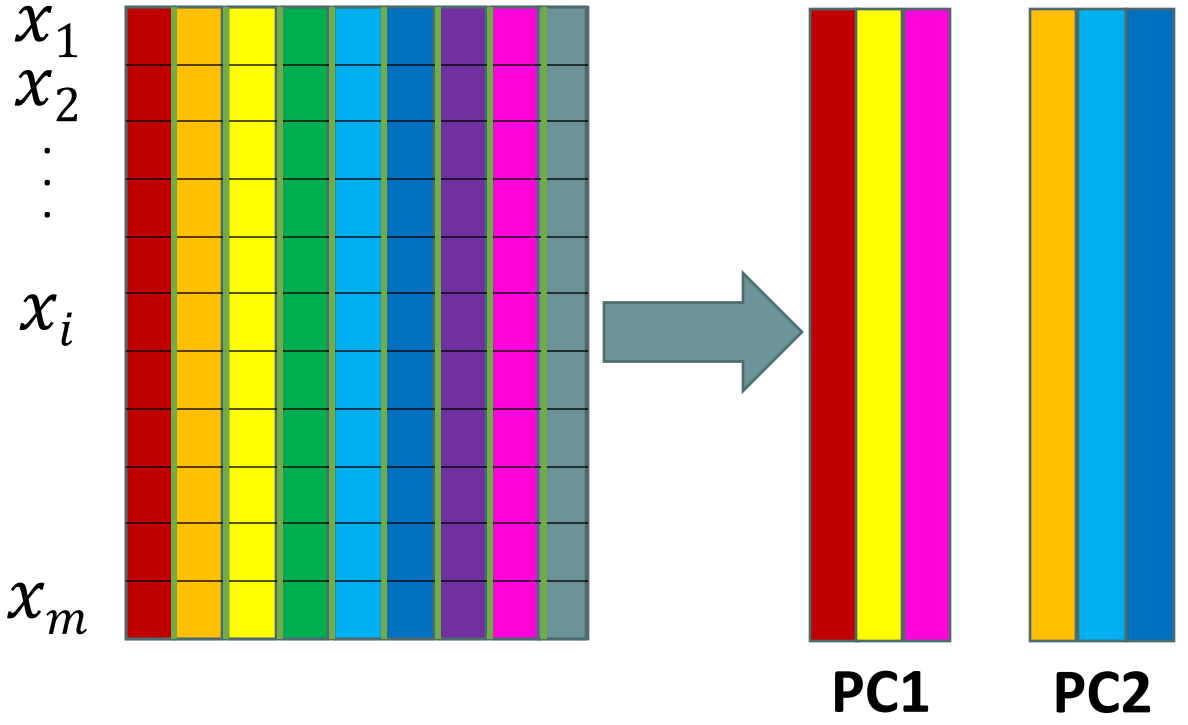


Random Search

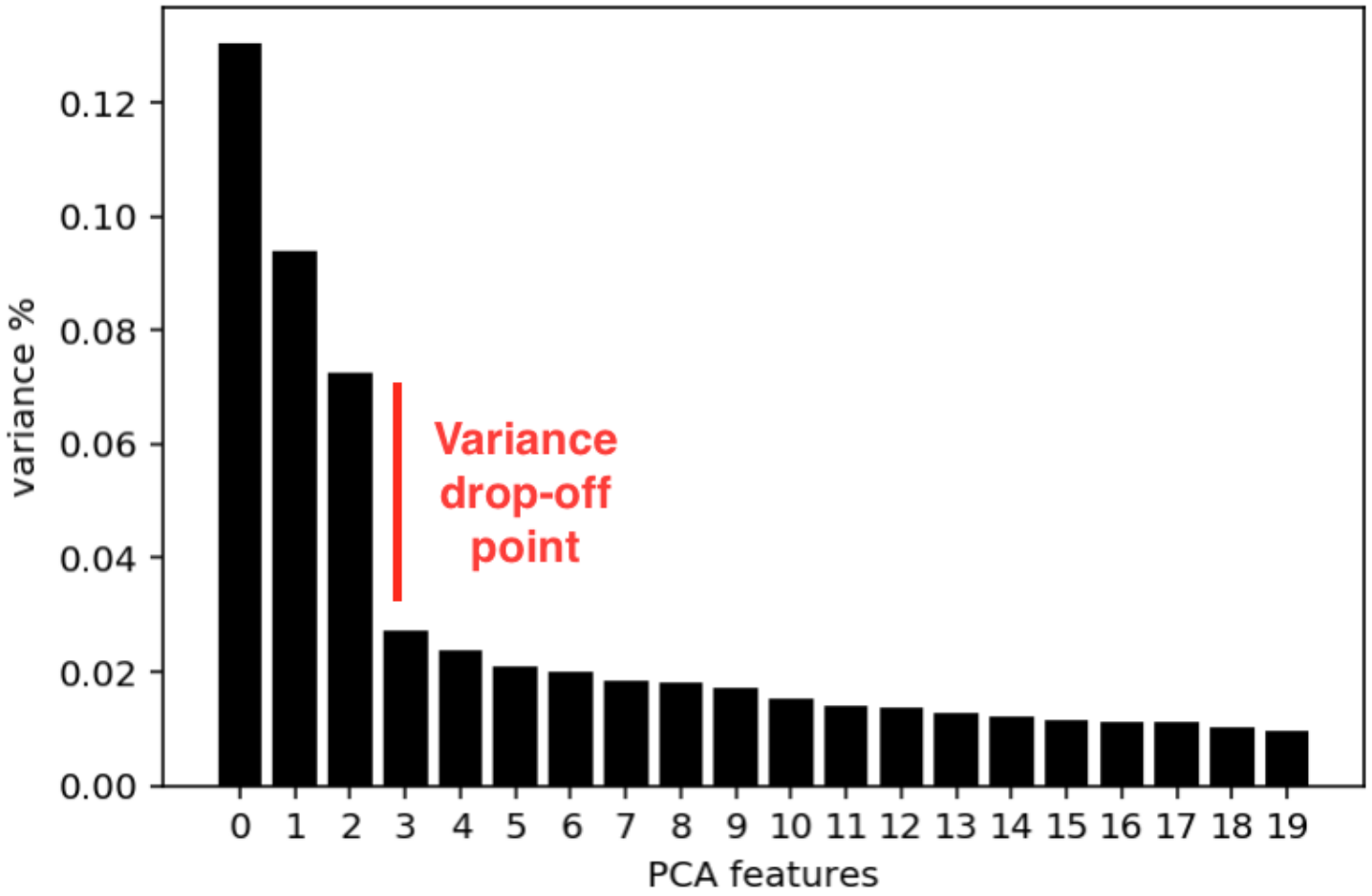
Principal Component analyse (PCA)

Strong dependance between variables: Multicollinearity?





Principal Component analyse PCA



Strengths:

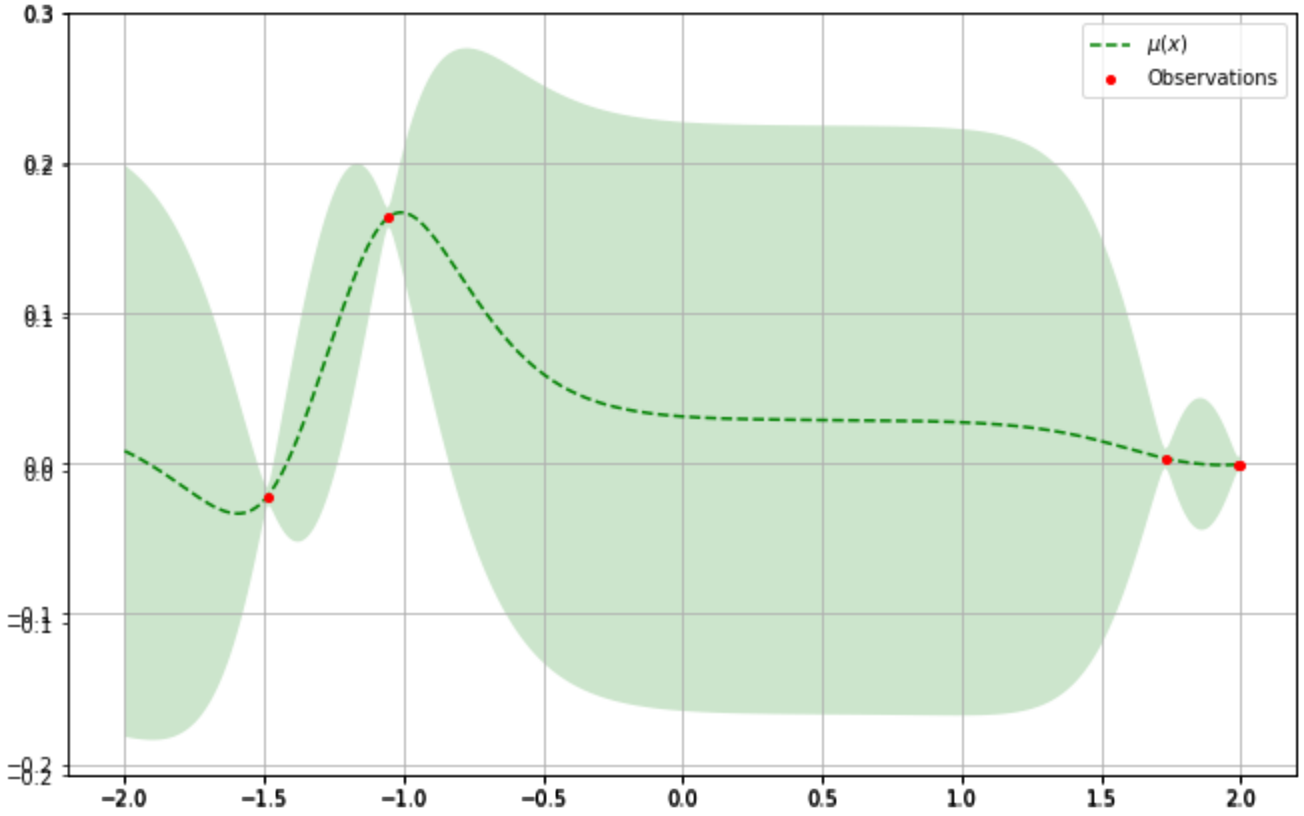
- Dimensionally reduction
- Removes correlated features
- Improves performance for large dataset
- Improves visualization

Weaknesses:

- Data standardization need
- Independent variables become less interpretable

Bayesian optimization

Target Value *y*



Parameters w_i, θ_i

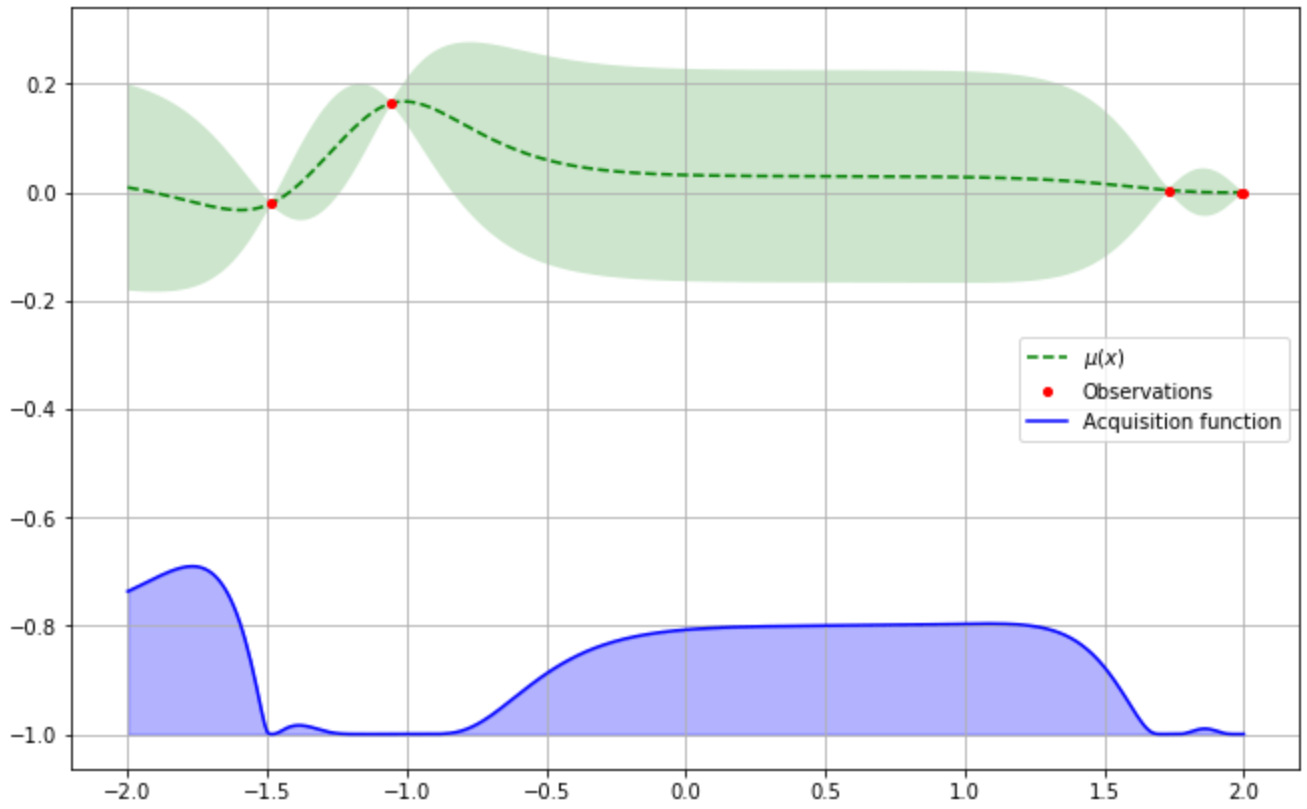
Few observables?

→ Use of a probabilistic model

e.g. Gaussian process to express the mean values μ and standard deviation s

Bayesian optimization

Target Value *y*



Parameters w_i, θ_i

Acquisition function?

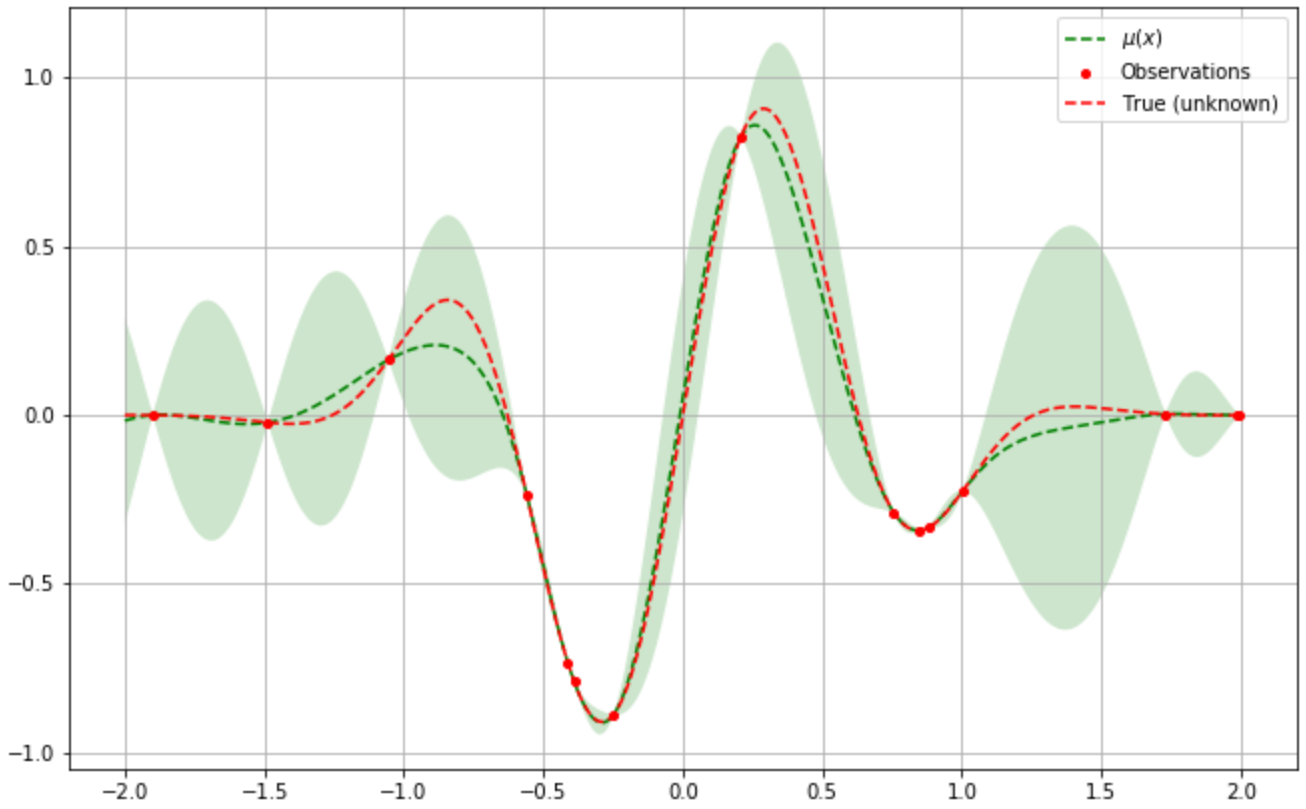
→Where is the strongest potential exploration part

Expected Improvement (EI)
Upper Confidence Bound (UCB)
Probability of Improvement (PI)...

$F(x_i) = \mu(x_i) + k * s(xi)$

Bayesian optimization → Active learming

Target Value *y*



Parameters w_i, θ_i

Iterative process with reoptimization

Strengths:

- Efficient with few data
- Suggest to explore new parameters
- Adaptability

Weaknesses:

- Need updated data
- Expensive for large dataset