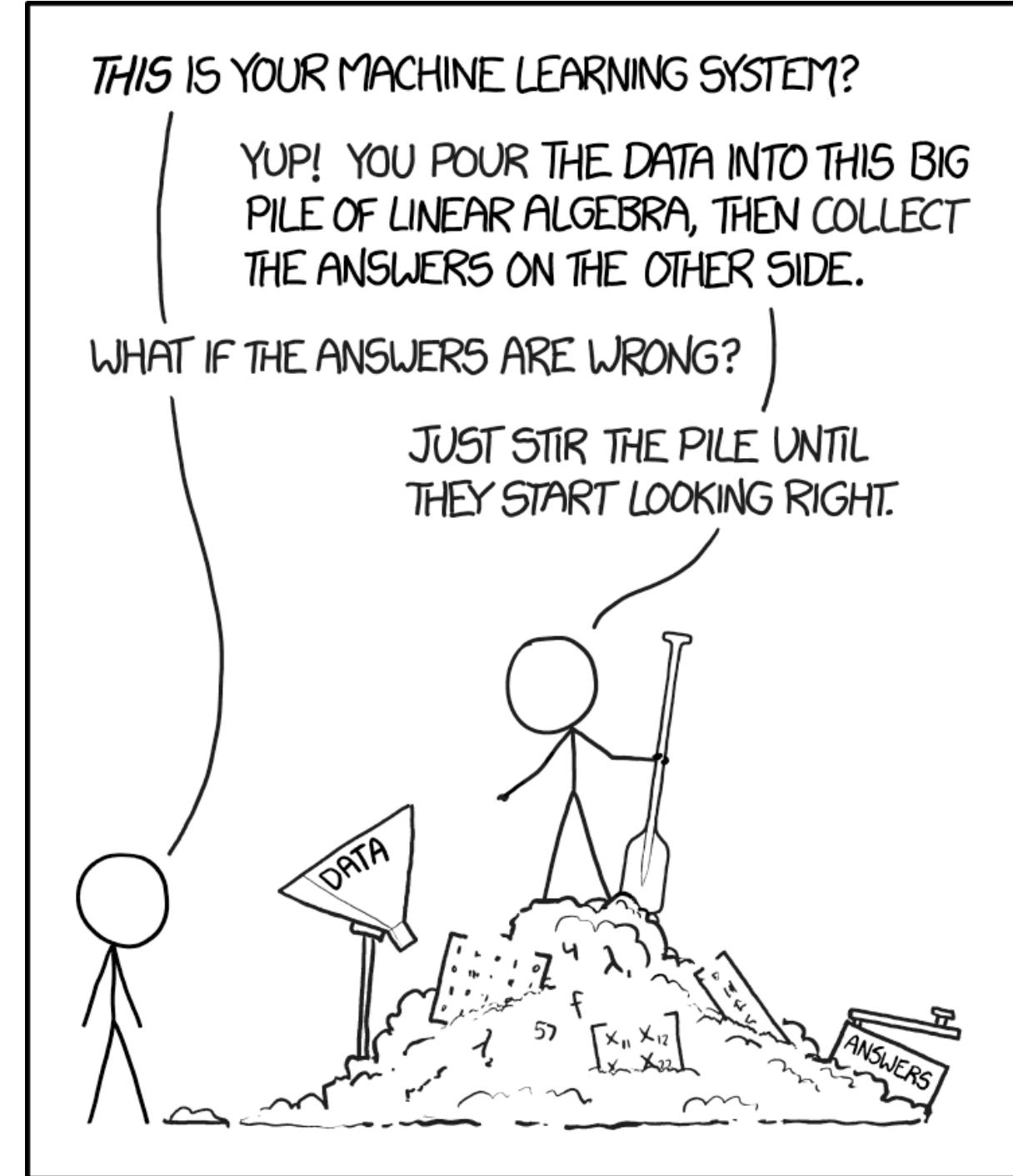


Machine Learning



From [xkcd](#)

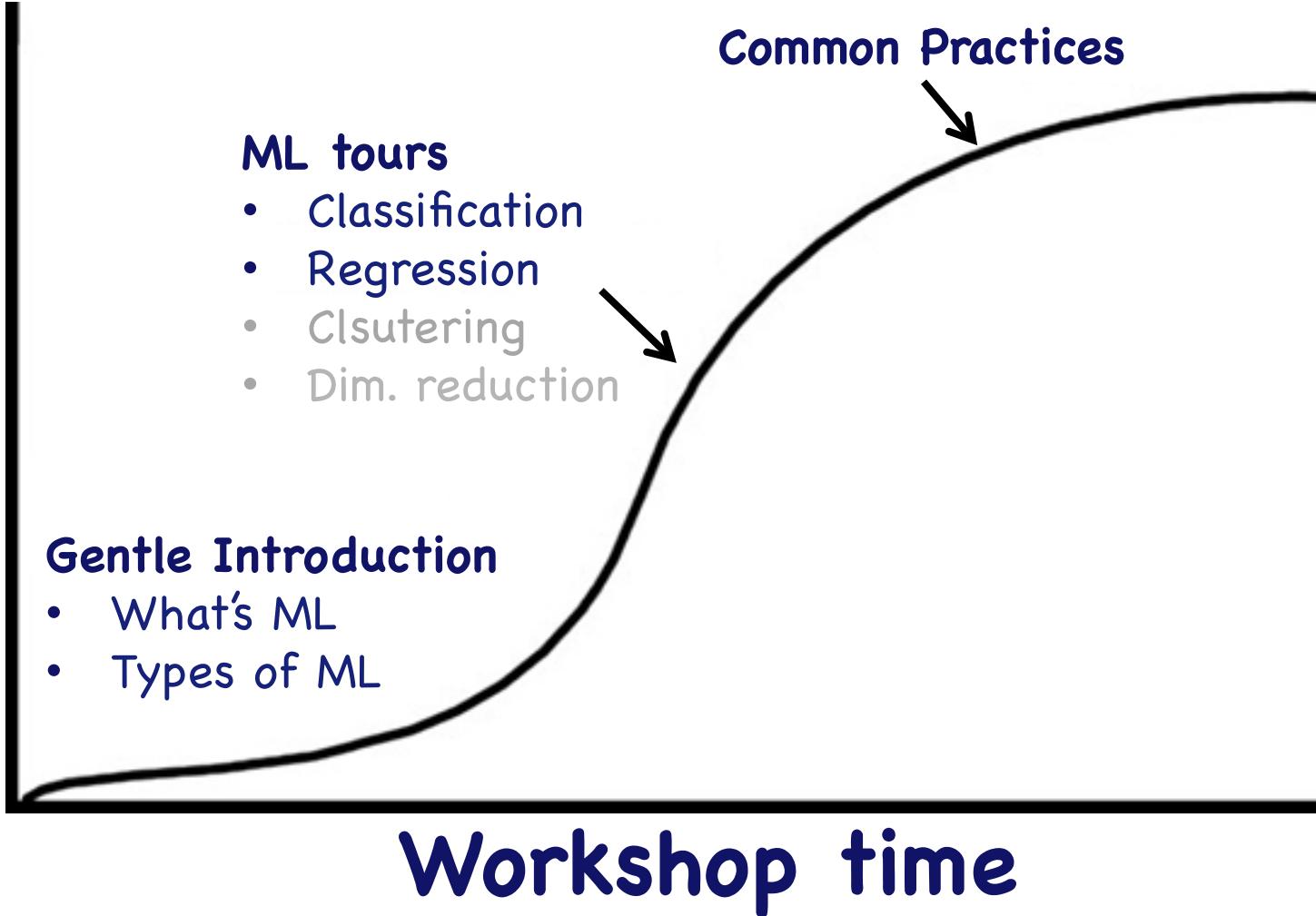


Machine learning - using scikit-learn

Qingkai Kong
2017-11-20

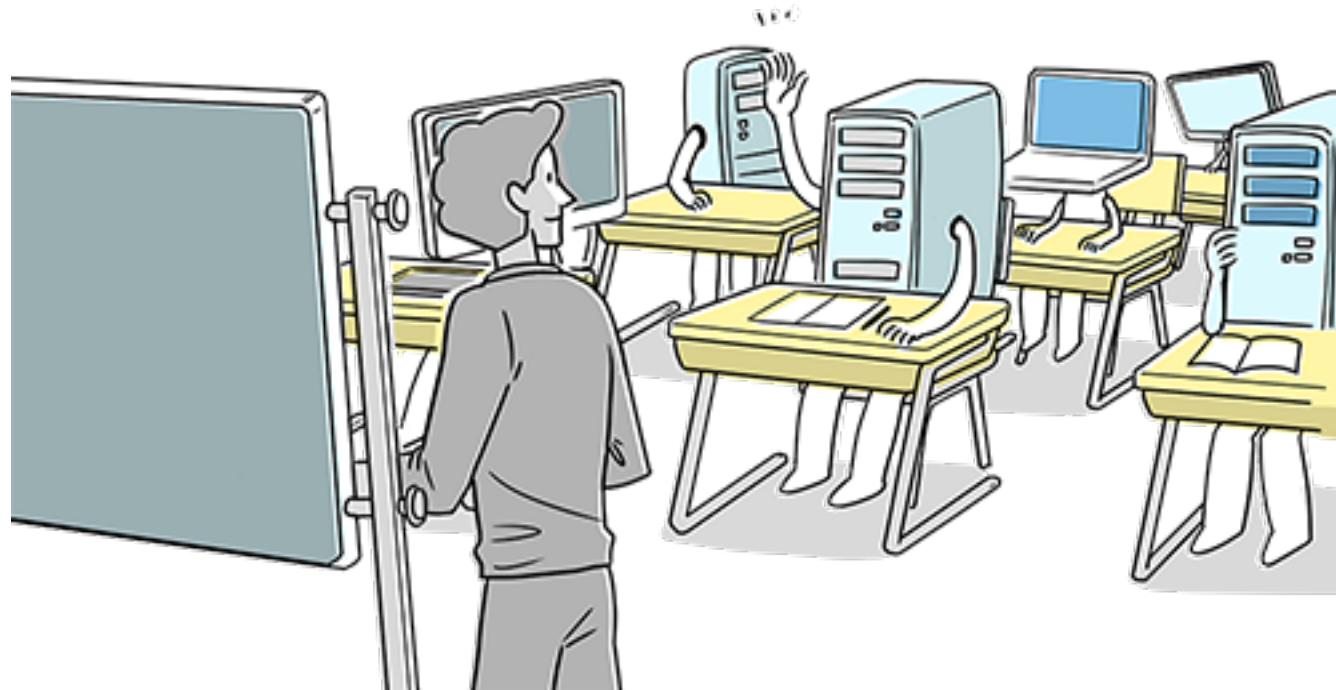
<http://seismo.berkeley.edu/qingkaikong/>

Learning curve



https://github.com/qingkaikong/20171120_ML_basics_EOW

What is machine learning?



https://github.com/qingkaikong/20171120_ML_basics_EOW

Data
examples

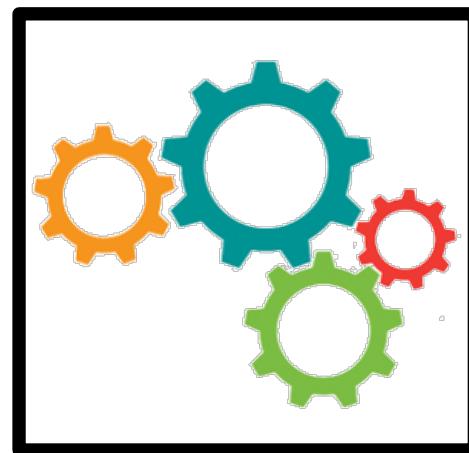
01100
10110
11110

Optimization
algorithm



Pipeline of training a machine learning model.

Tunable
Model



Trained
Model

MAKE
THINGS
HAPPEN!



amazon.com

Recommended for You

Amazon.com has new recommendations for you based on items you purchased or told us you own.

 Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop	 Google Apps Administrator Guide: A Private-Label Web Workspace	 Googlepedia: The Ultimate Google Resource (3rd Edition)
---	--	---

**Self-driving car
Voice recognition**

...

https://github.com/qingkaikong/20170628_ML_sklearn

Not always
working

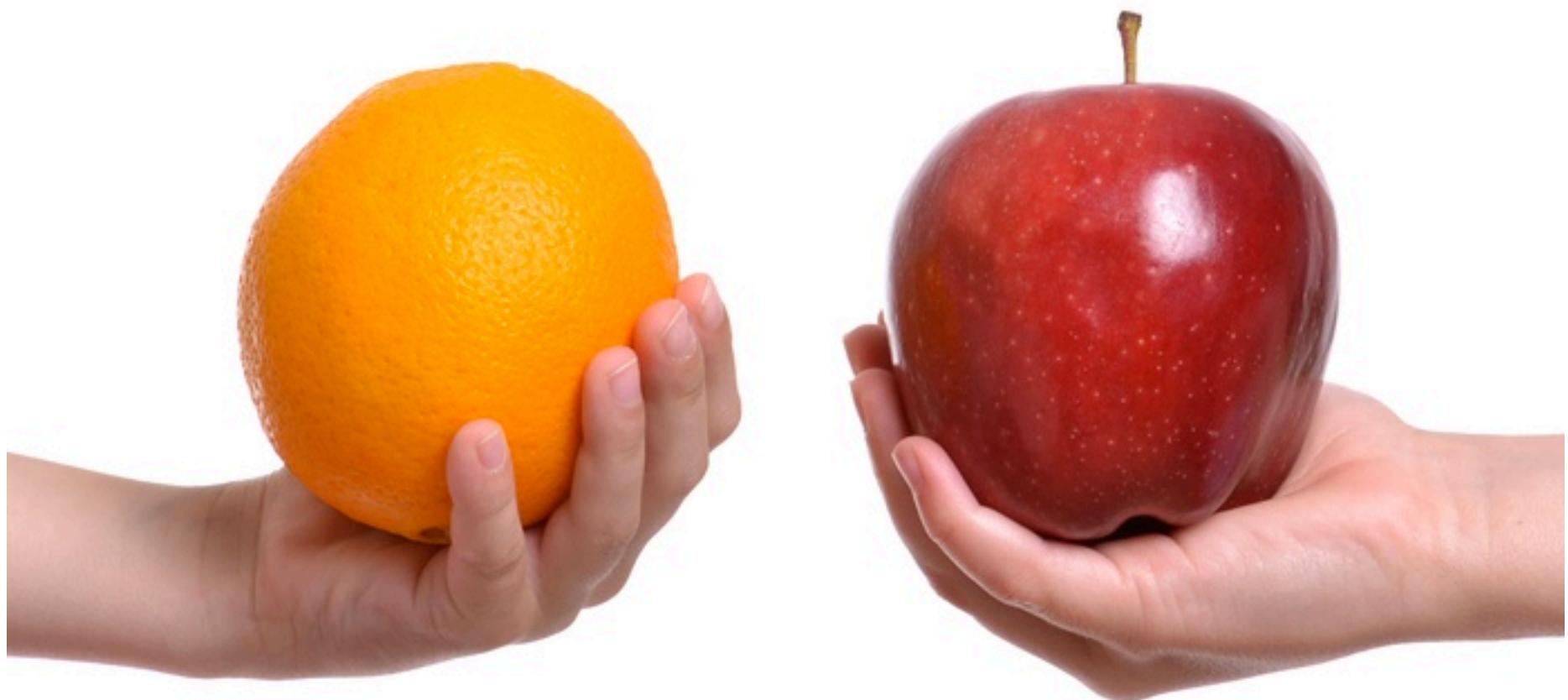


Common types of machine learning

Supervised
learning

Unsupervised
learning

Supervised learning





Unsupervised learning

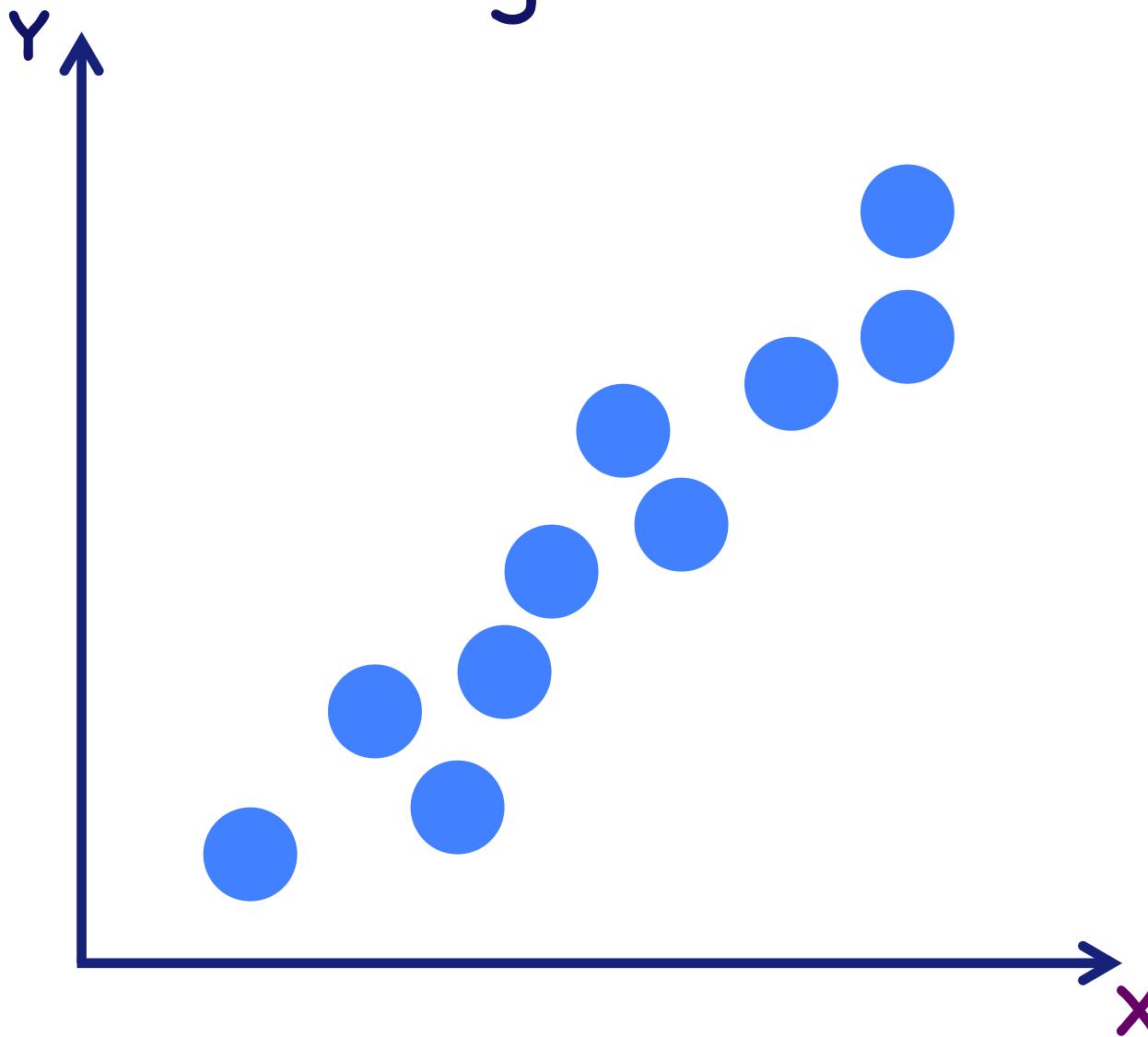


Supervised
learning

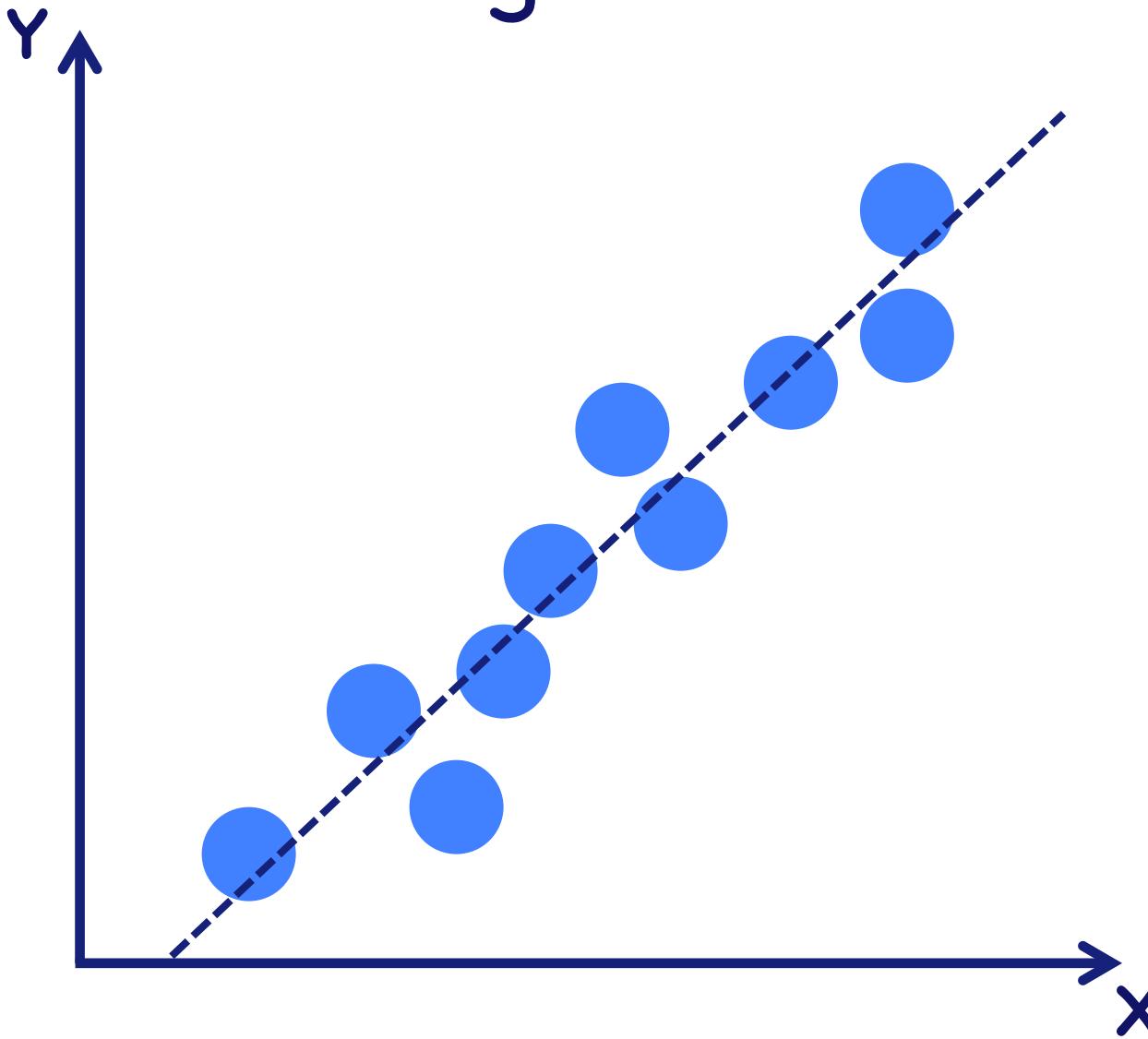
Regression

Classification

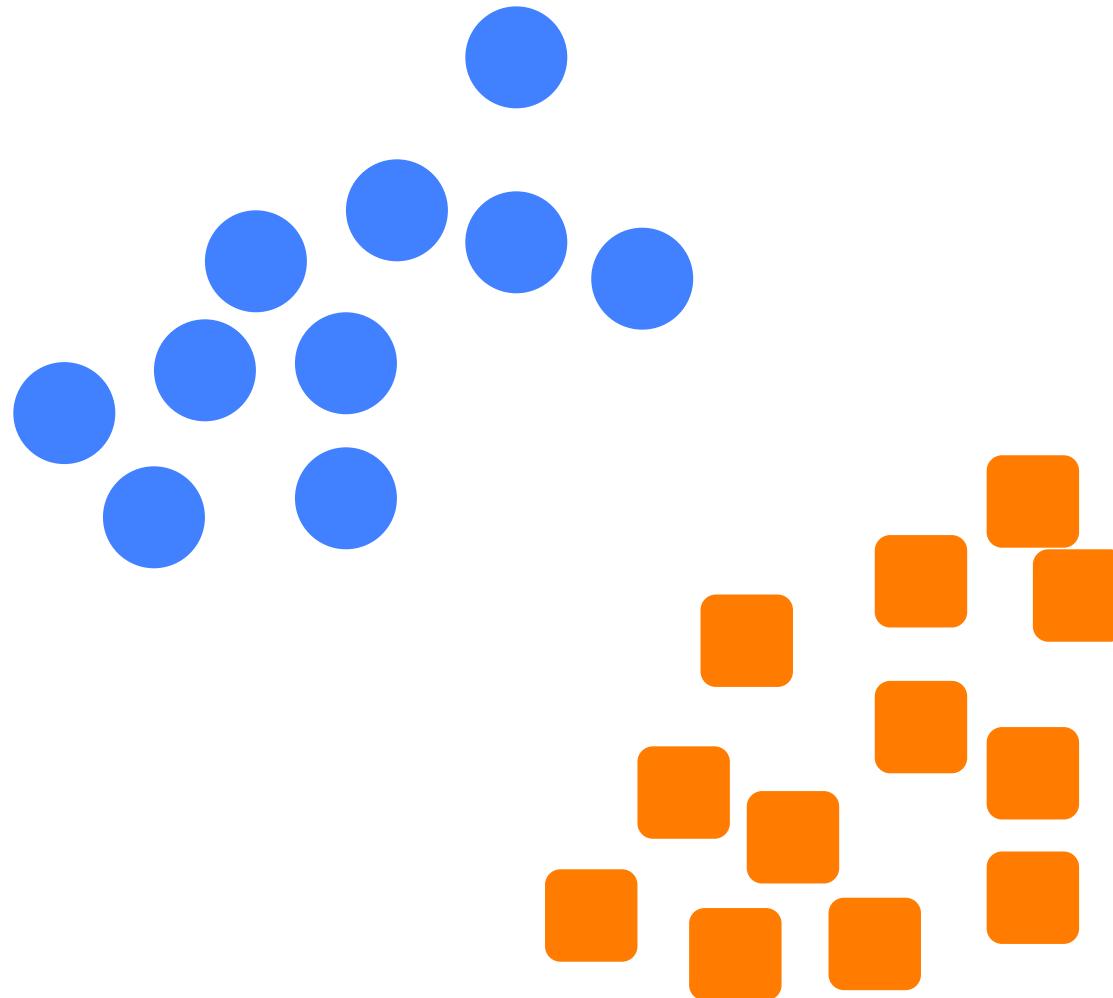
Regression



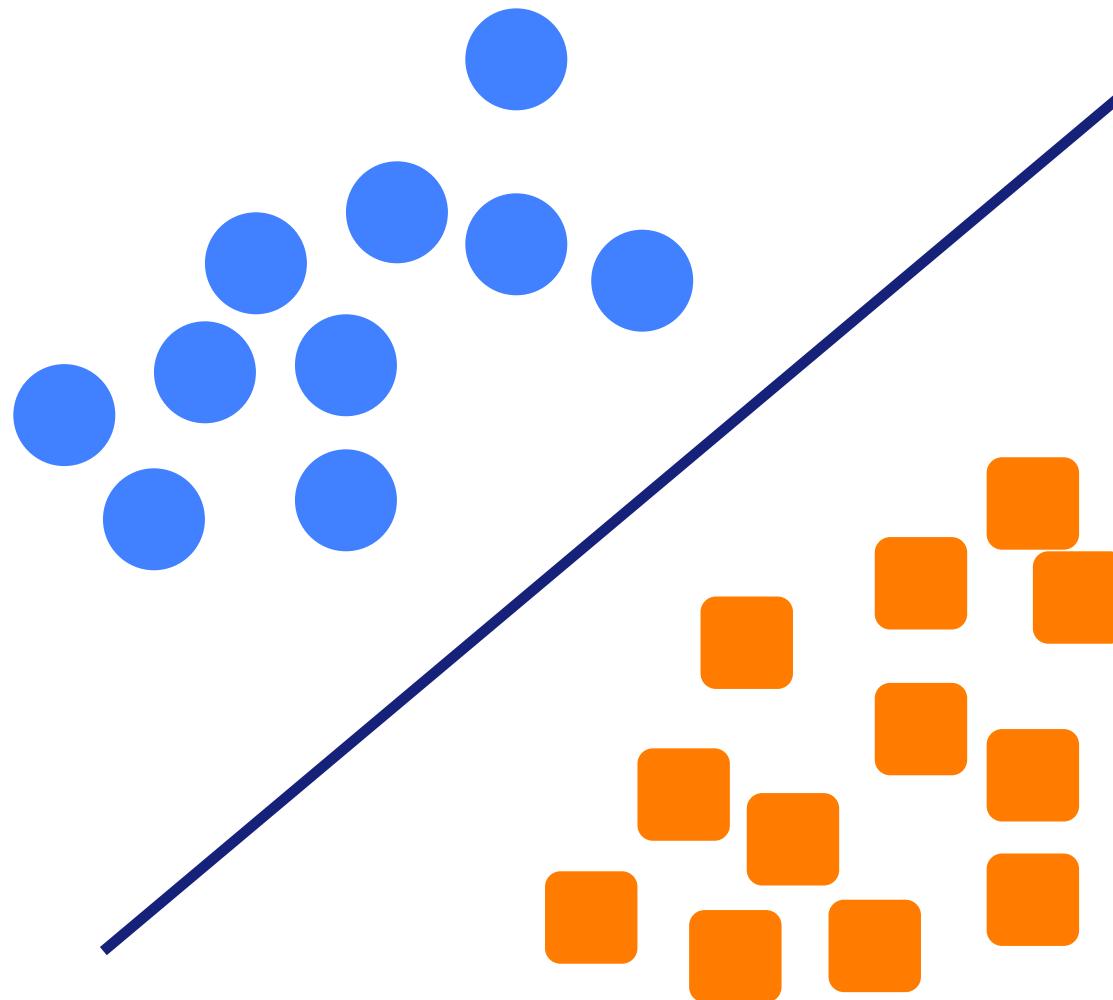
Regression



Classification



Classification





Unsupervised
learning

Clustering

Dimensionality
reduction

Data
examples

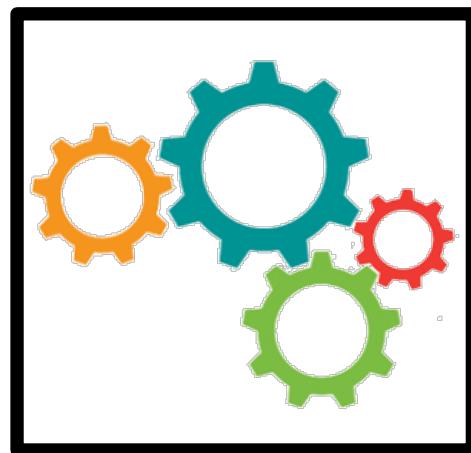
01100
10110
11110

Optimization
algorithm



Pipeline of training a machine learning model.

Tunable
Model



Trained
Model

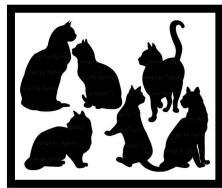
MAKE
THINGS
HAPPEN!

Representation of data

Raw data



Documents



Images



Numbers

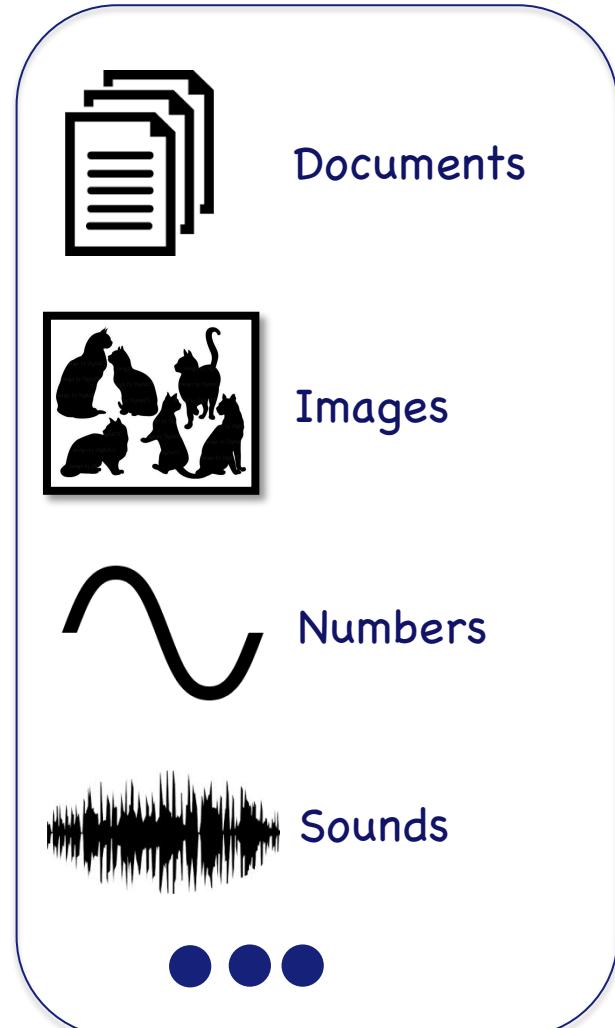


Sounds

• • •

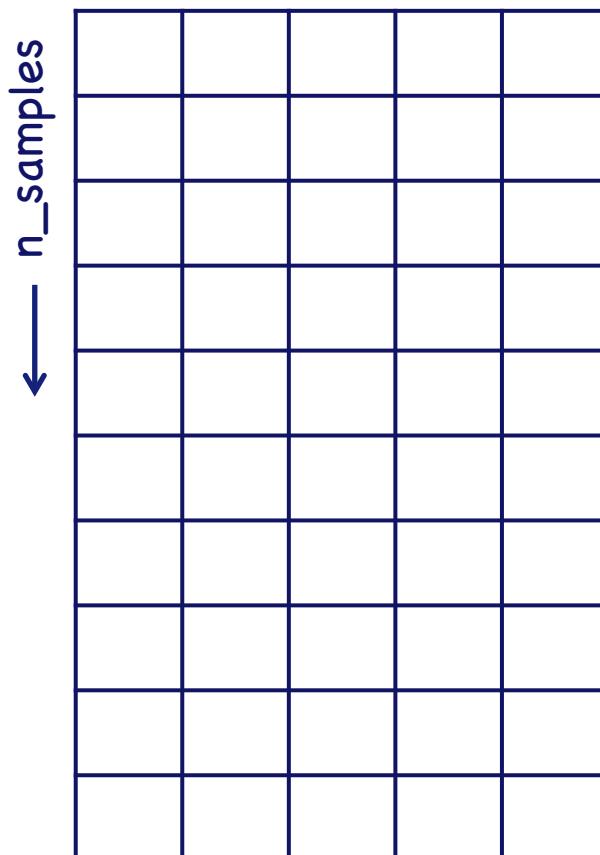
Representation of data

Raw data



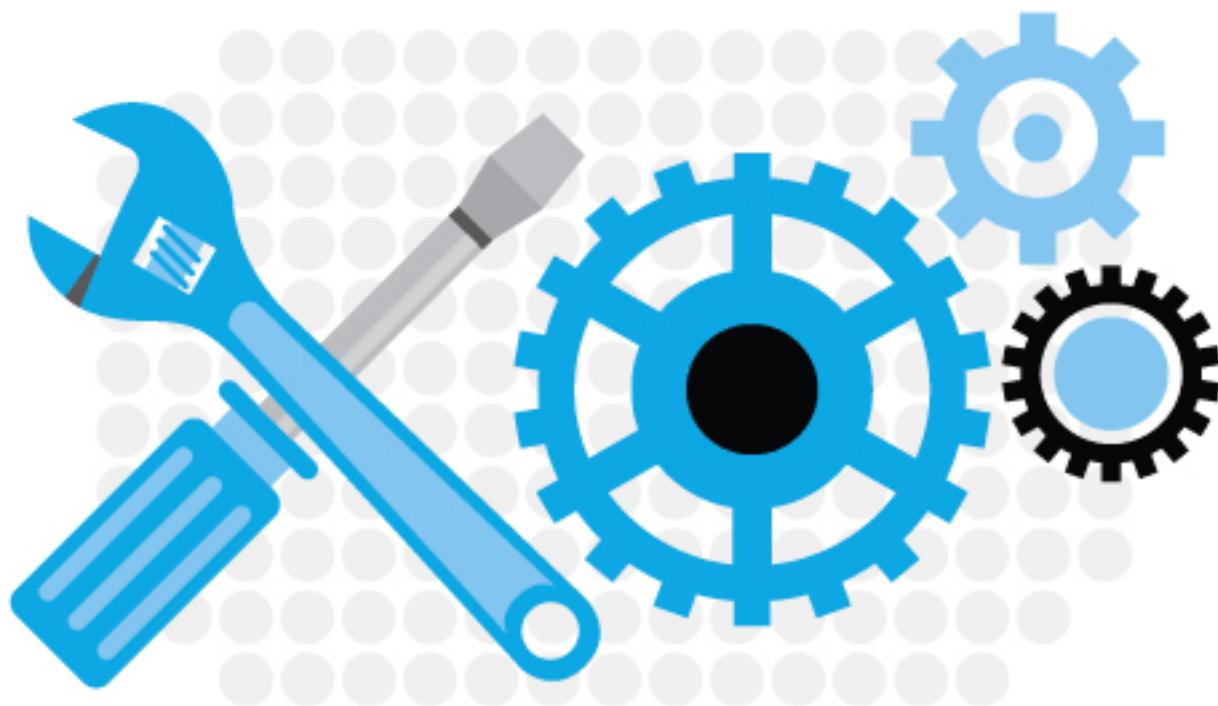
Feature matrix (X)

n_features →

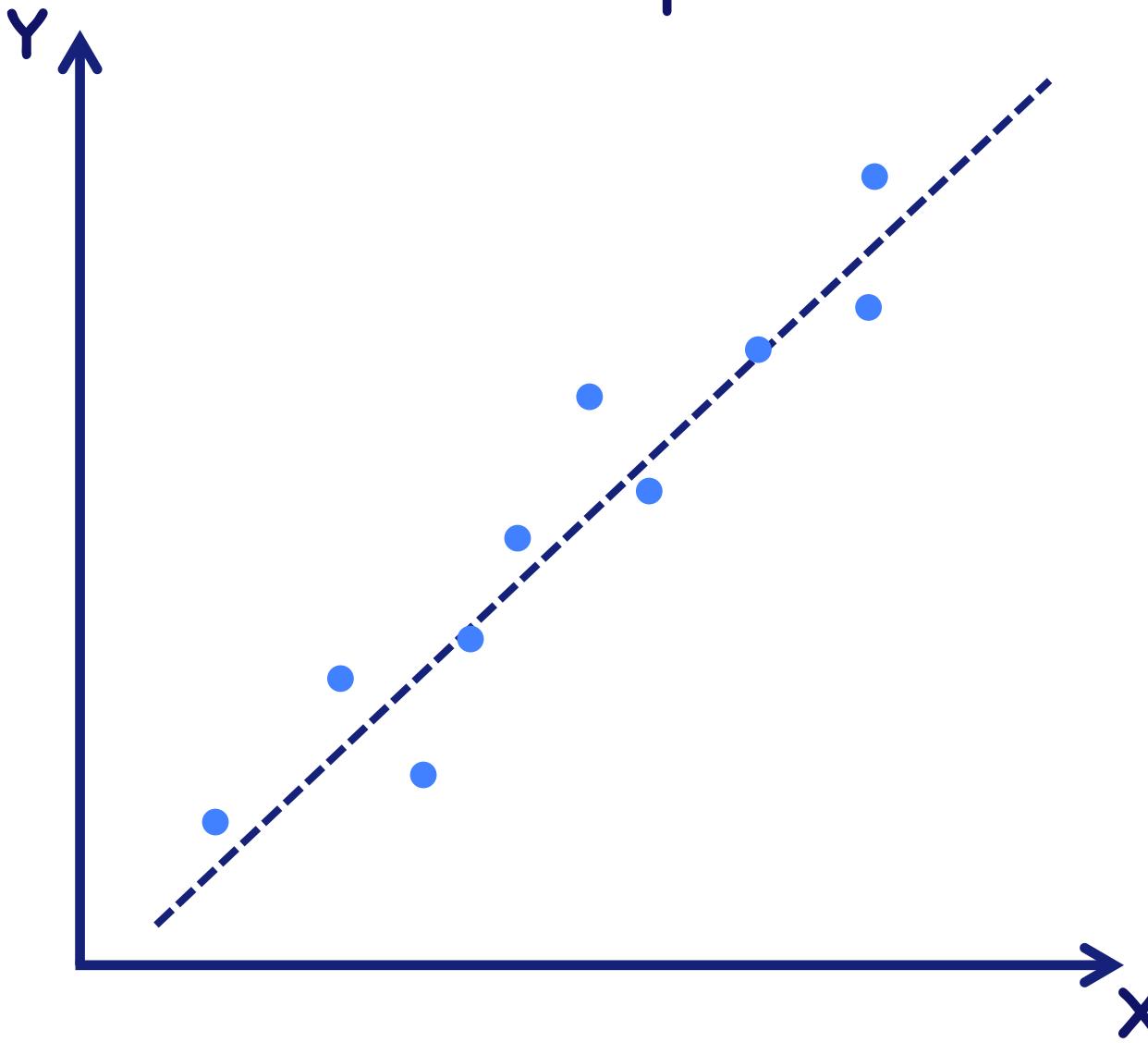


Target (y)

Optimization

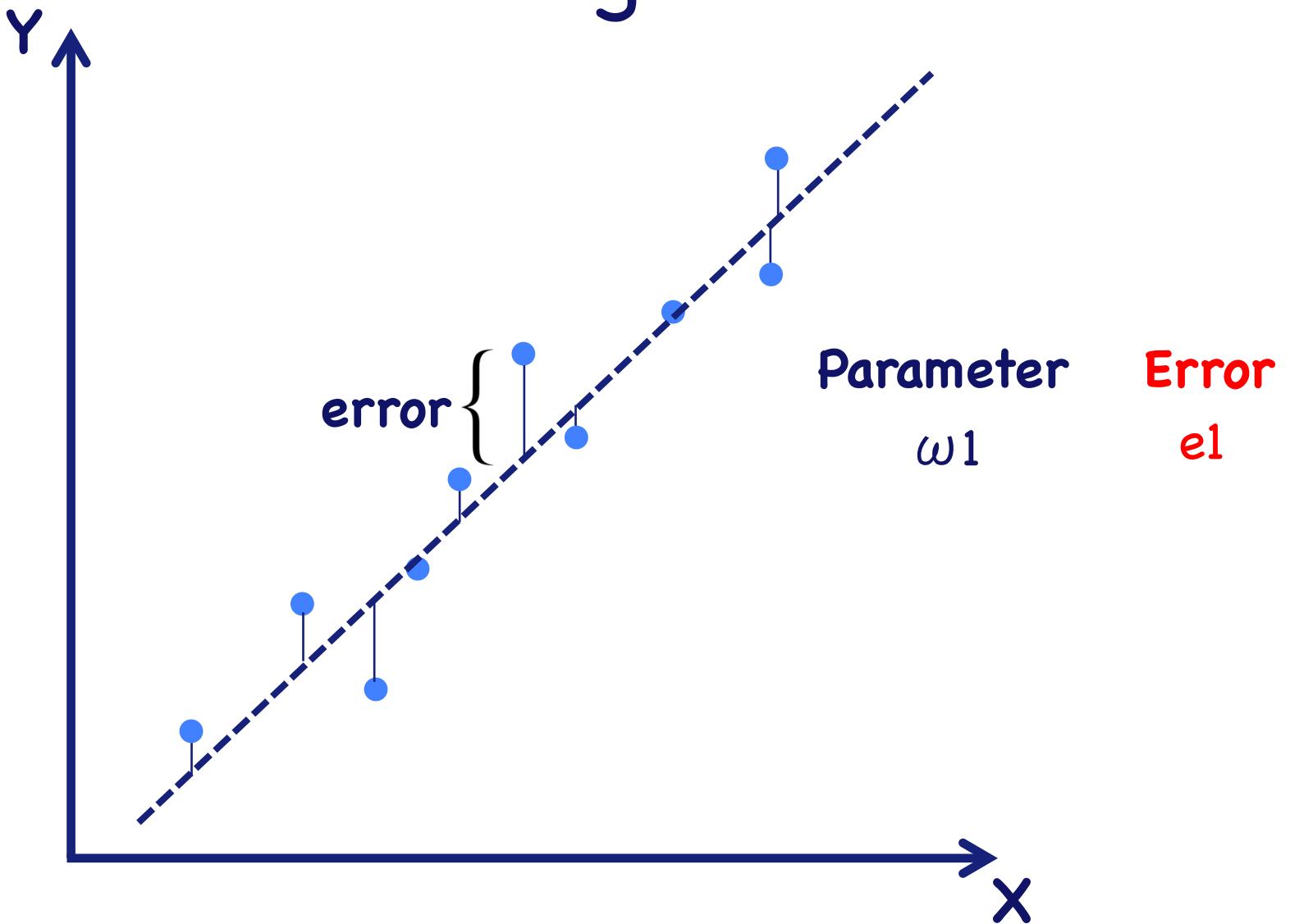


How we optimize?

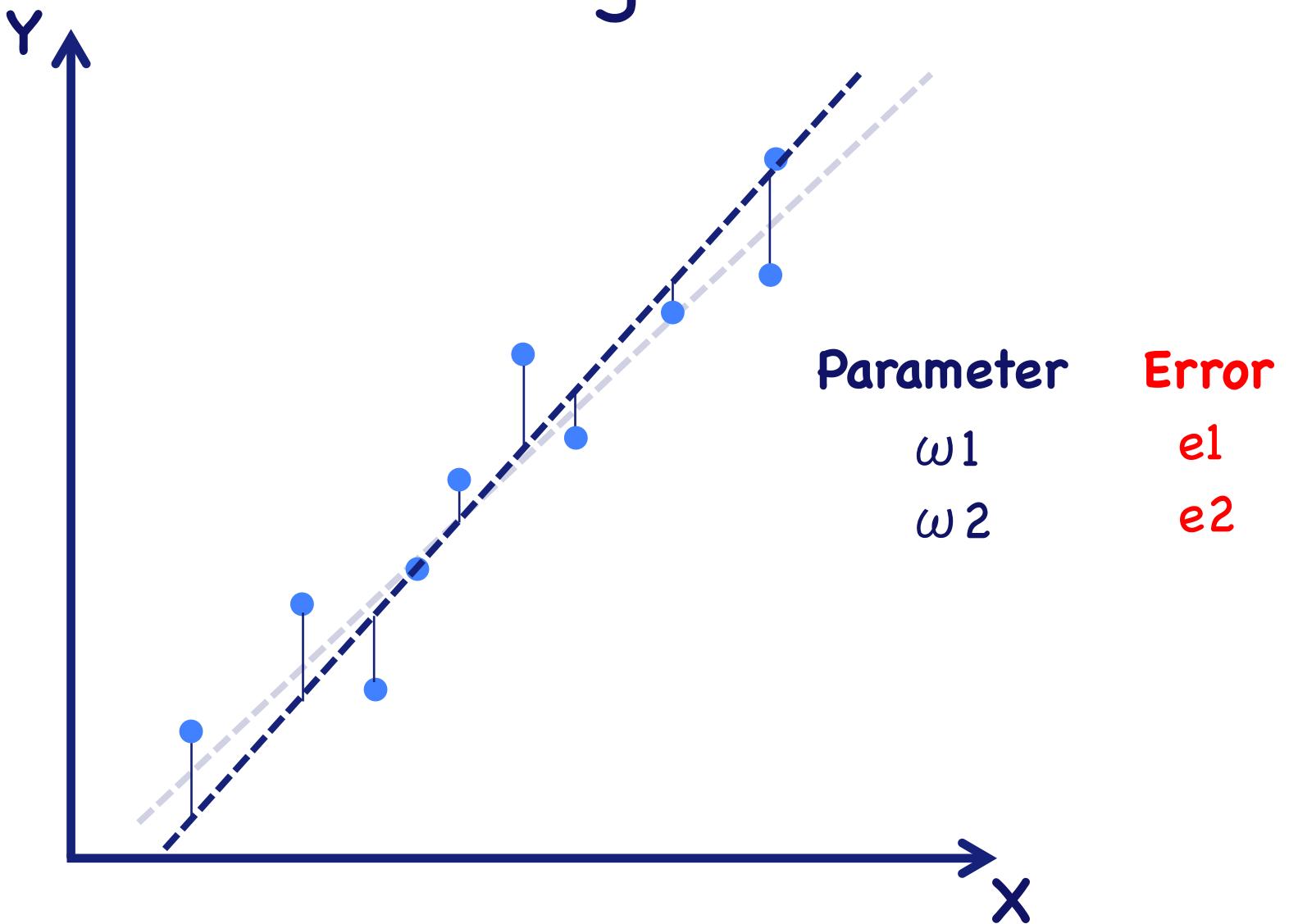


A simple example

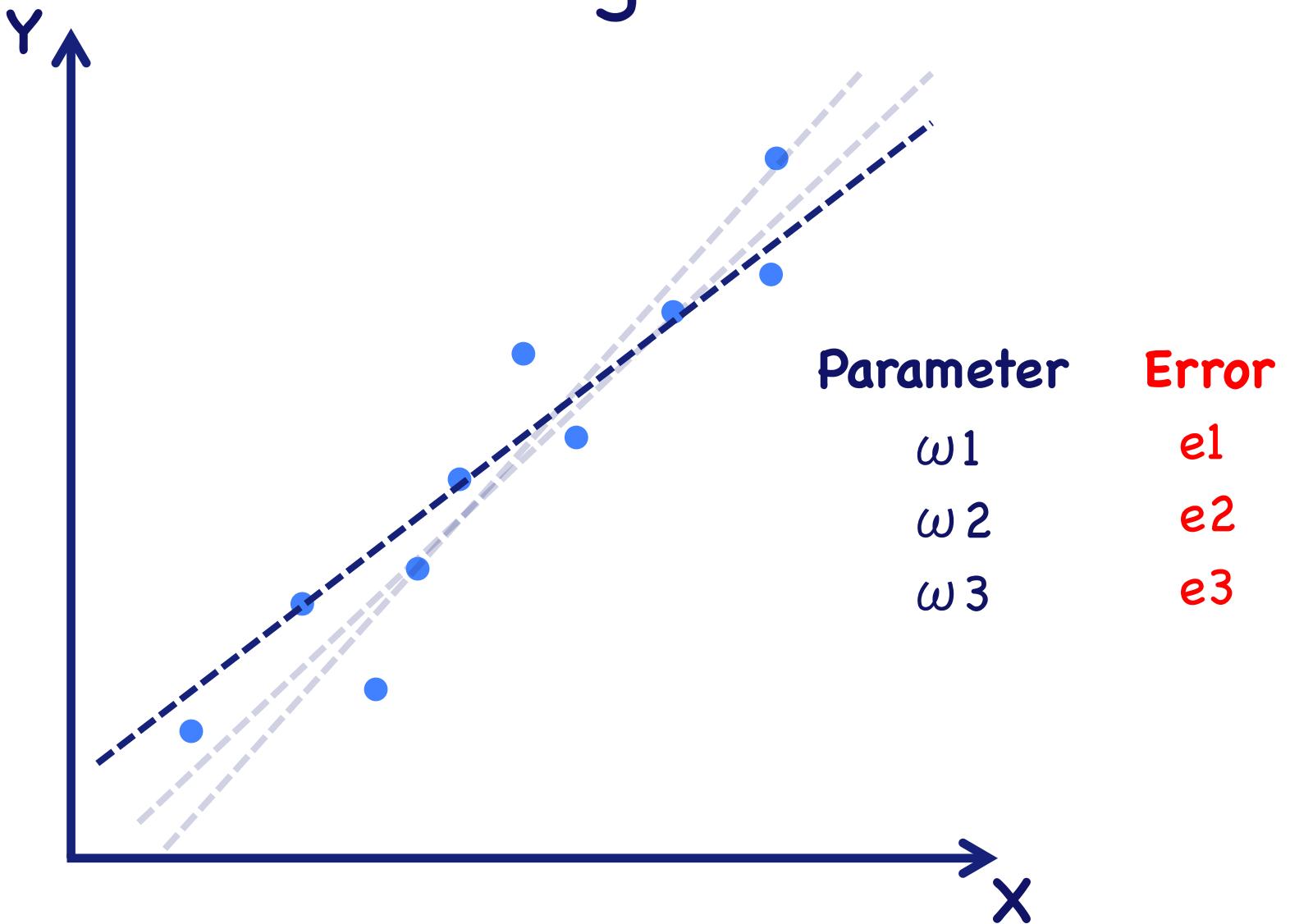
Measuring error



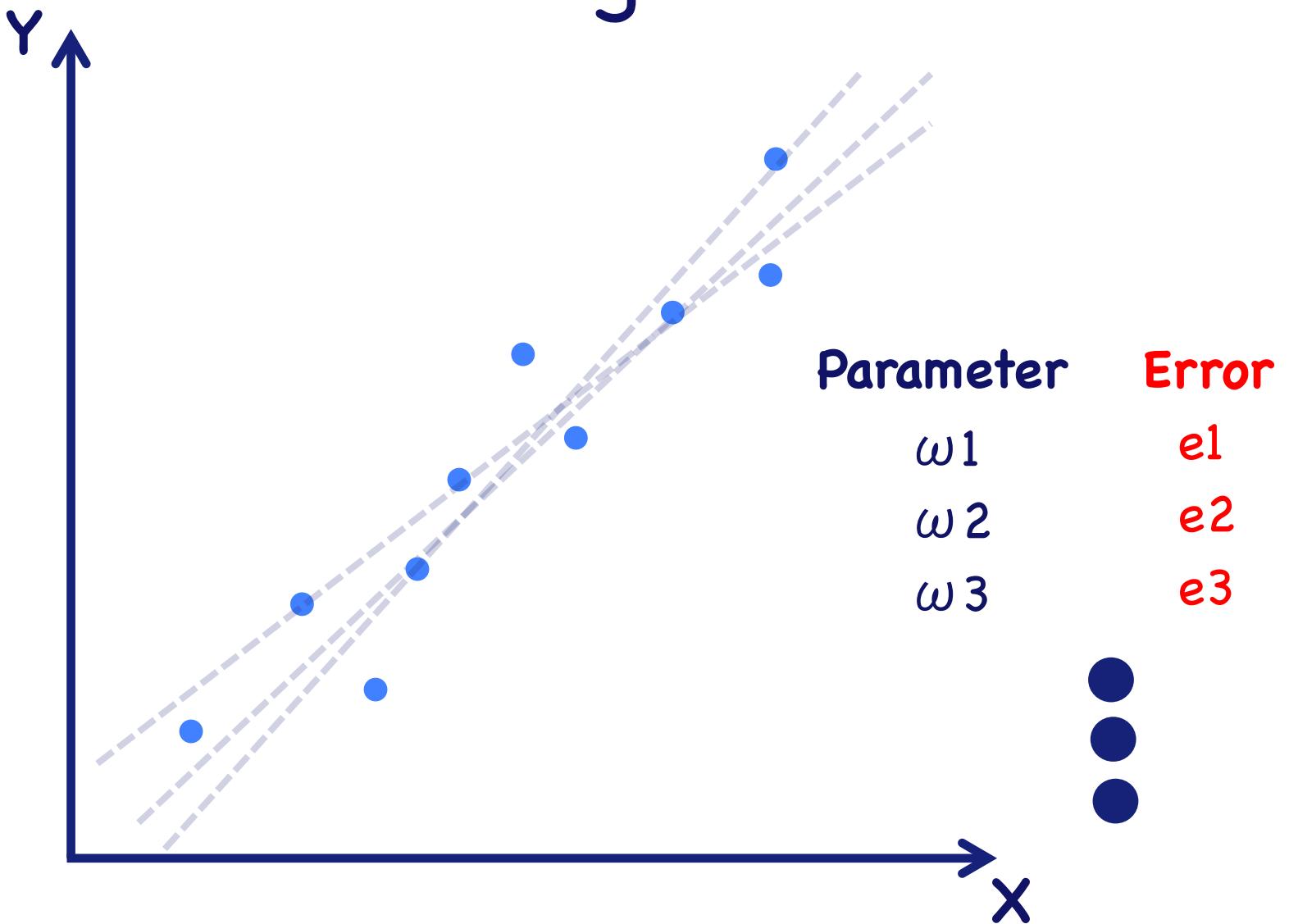
Measuring error



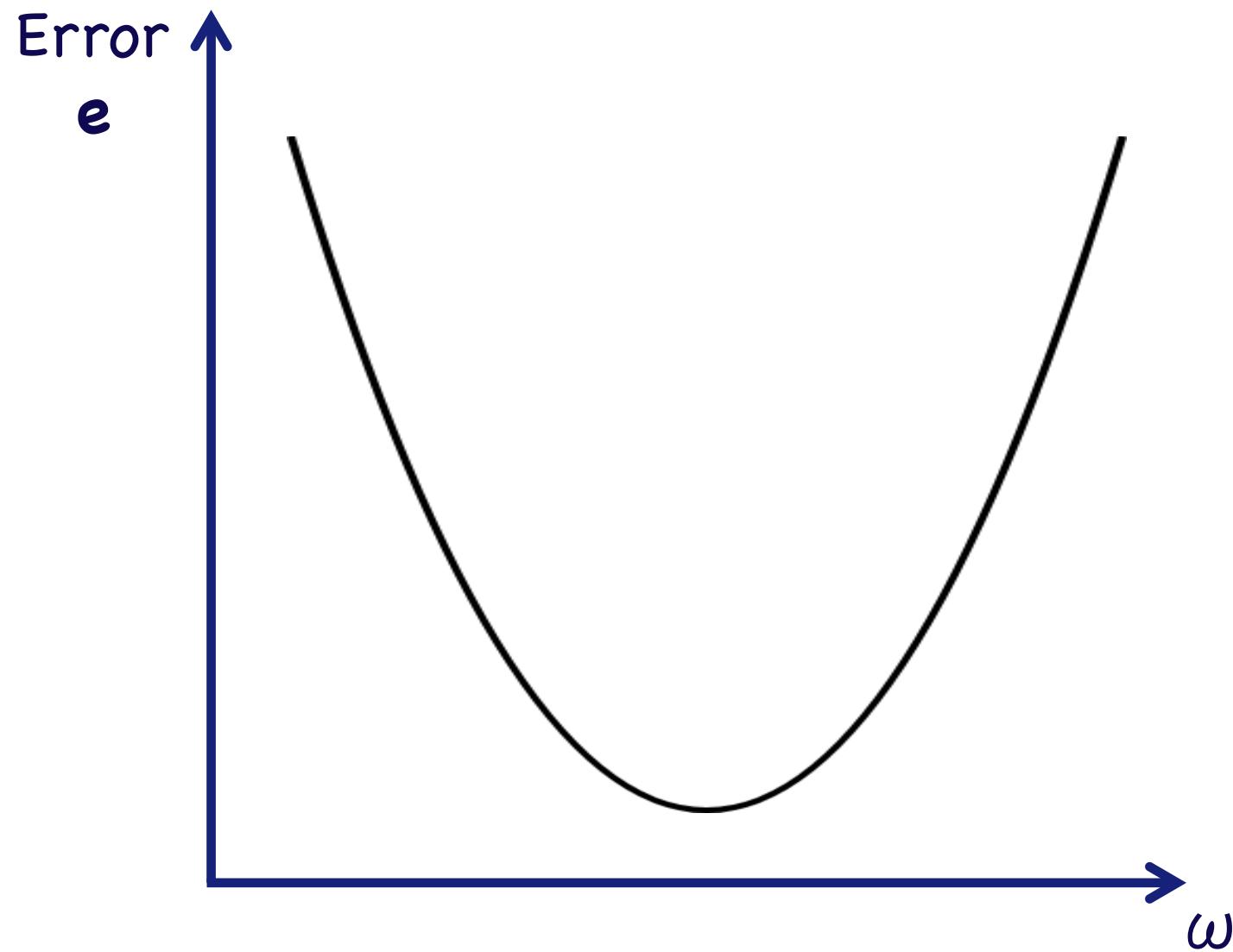
Measuring error



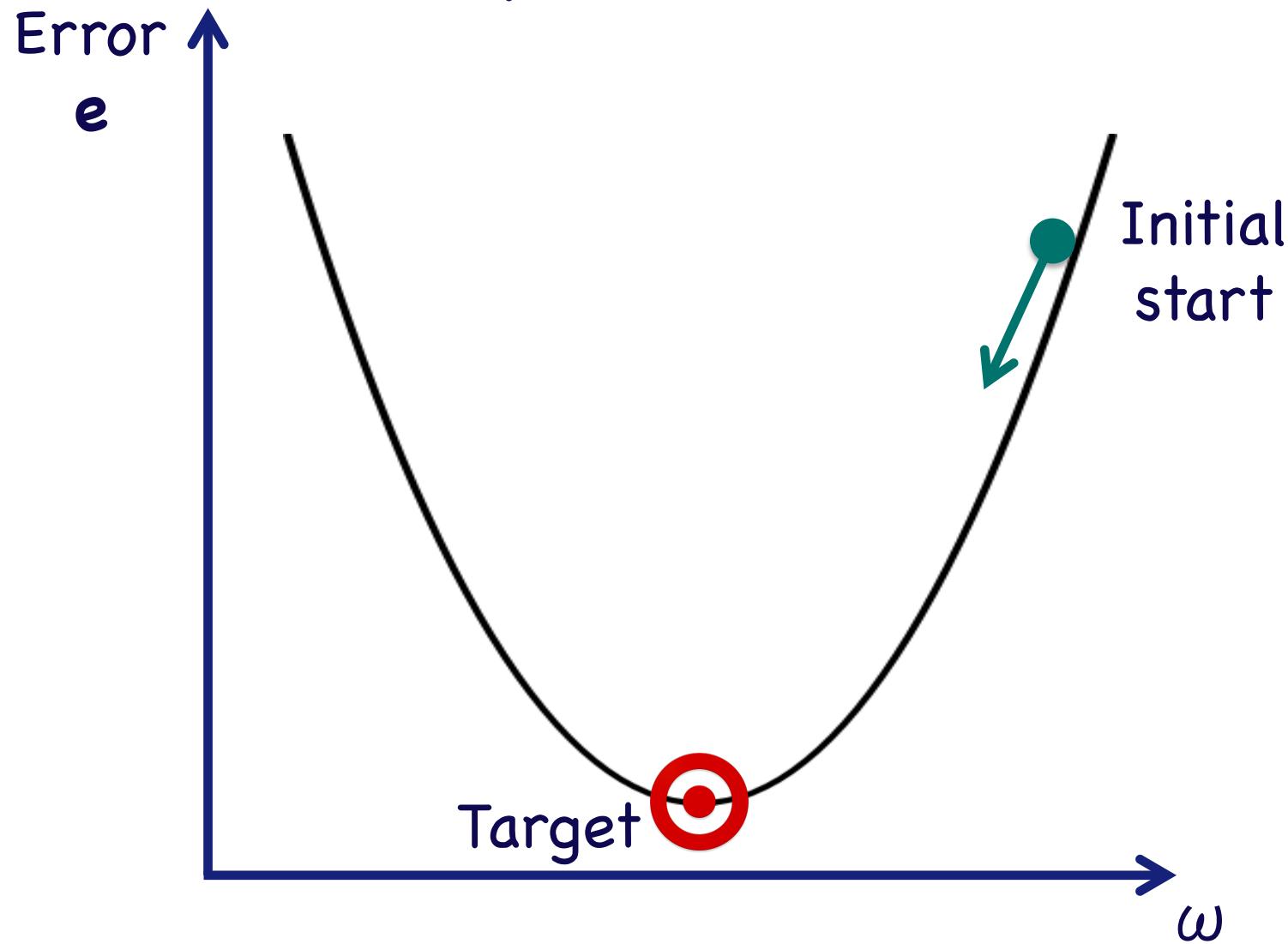
Measuring error



Cost function

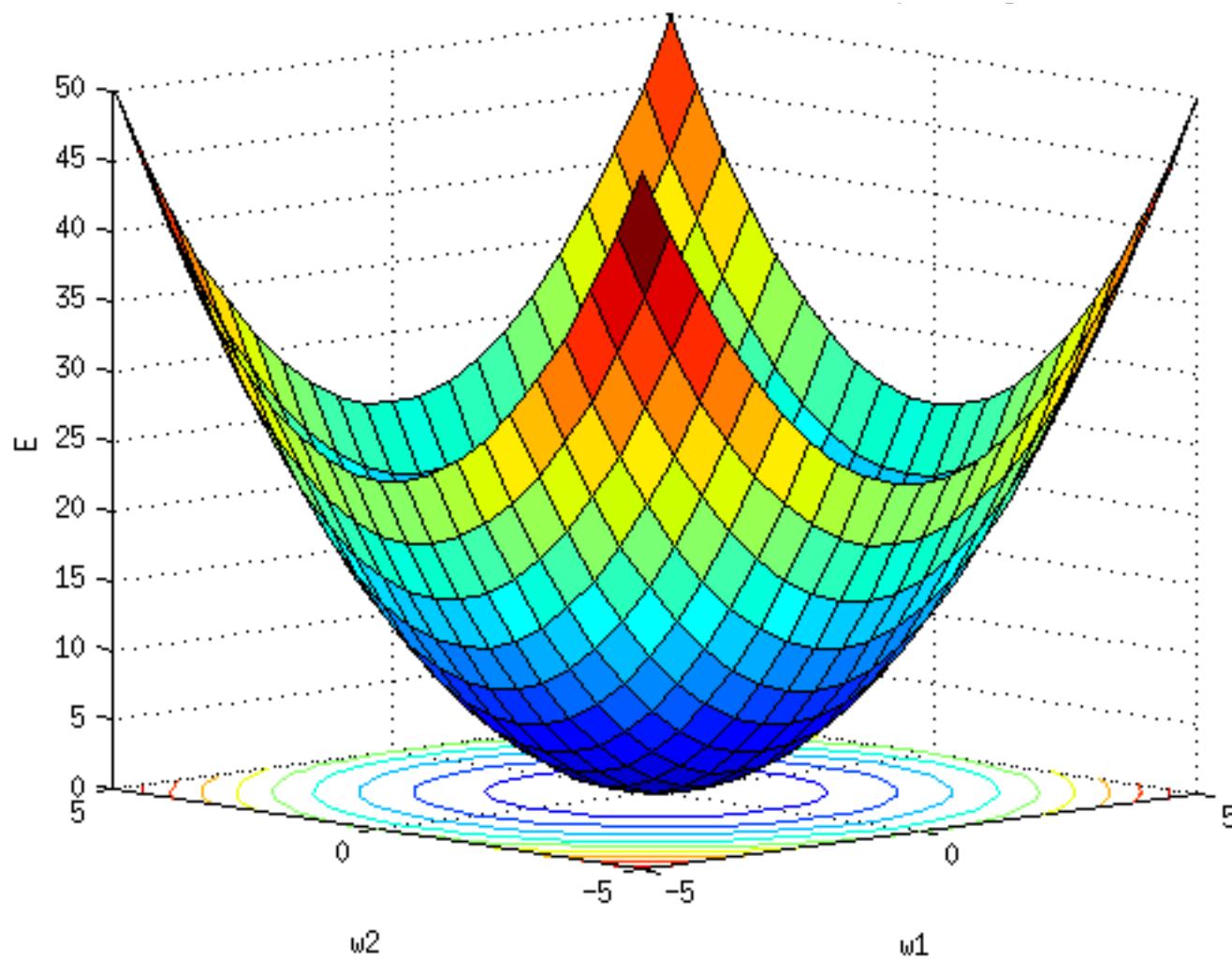


Optimization



Of course, there are many different types of cost functions and optimization algorithms, but the general idea is very similar.

Ideal Cost Function



Real-world Cost Function



Convolutional Neural Network Quadratic Discriminant Analysis

Generative Adversarial Networks Hierarchical clustering

Nearest Neighbor

Radial Basis Function Network

Support Vector Machine

Artificial Neural Network

K-means

Linear Regression

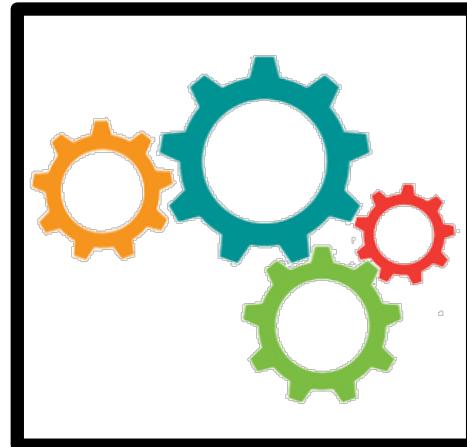
Naïve Bayes

Logistic Regression

AdaBoost Boosting

Self-Organizing Map

Decision Tree



Stepwise Regression

Ridge Regression

Random Forest LASSO

Recurrent Neural Network

Principal Component Analysis

Bayesian Network

Ordinary Least Squares Regression

Linear Discriminant Analysis

Gaussian Mixture Model

Convolutional Neural Network Quadratic Discriminant Analysis

Generative Adversarial Networks Hierarchical clustering

Nearest Neighbor

Radial Basis Function Network

Support Vector Machine

K-means

Naïve Bayes

AdaBoost Boosting

Decision Tree

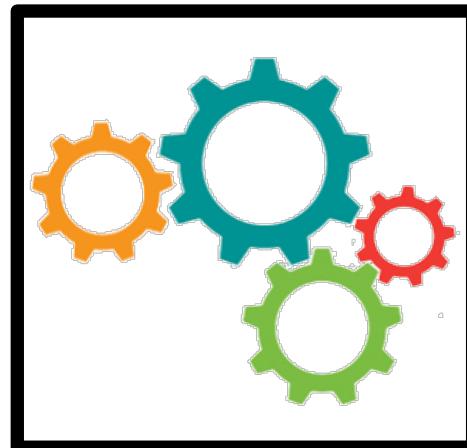
Stepwise Regression

Random Forest

LASSO

Bayesian Network

Linear Discriminant Analysis



Artificial Neural Network

Linear Regression

Logistic Regression

Self-Organizing Map

Ridge Regression

Recurrent Neural Network

Principal Component Analysis

Ordinary Least Squares Regression

Gaussian Mixture Model



<http://scikit-learn.org/stable/>

Machine Learning



what society thinks I do



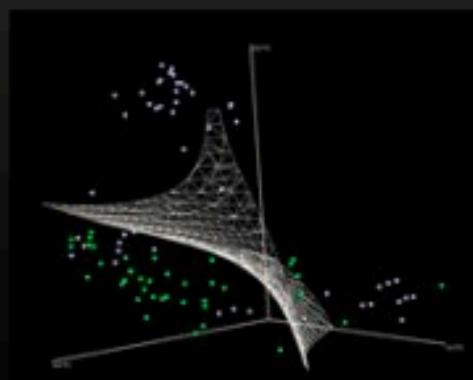
what my friends think I do



what my parents think I do

$$\begin{aligned}L_p &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^n \alpha_i \\ \alpha_i &\geq 0, \forall i \\ \mathbf{w} &= \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^n \alpha_i = 0 \\ \nabla g(\theta_t) &= \frac{1}{n} \sum_{i=1}^n \nabla \ell(x_i, y_i; \theta_t) + \nabla r(\theta_t). \\ \theta_{t+1} &= \theta_t - \eta_t \nabla \ell(x_{i(t)}, y_{i(t)}; \theta_t) - \eta_t \cdot \nabla r(\theta_t) \\ E_{i(t)}[\ell(x_{i(t)}, y_{i(t)}; \theta_t)] &= \frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i; \theta_t).\end{aligned}$$

what other programmers think I do



what I think I do

```
>>> from sklearn import svm
```

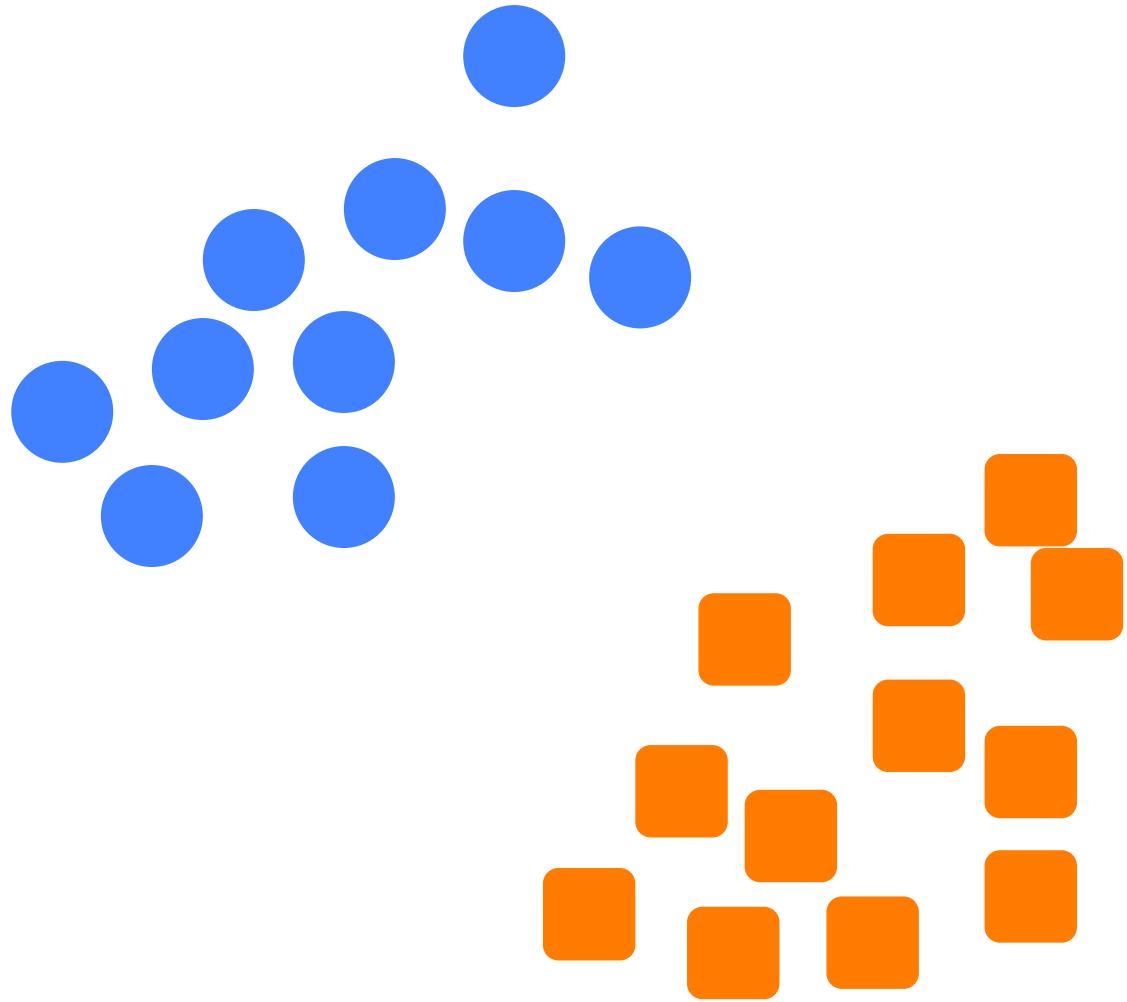
what I really do

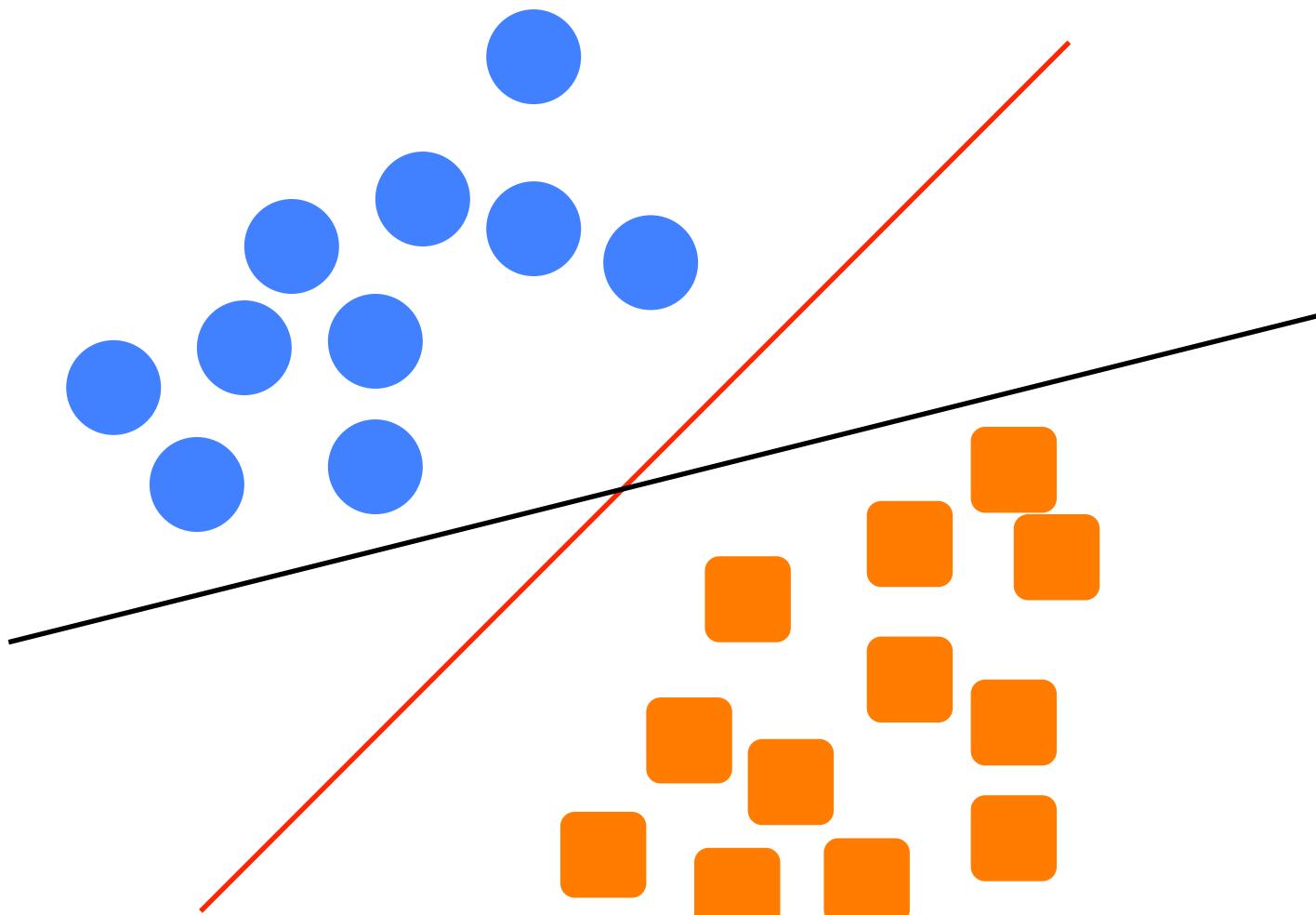
Go to notebook 01

Classification

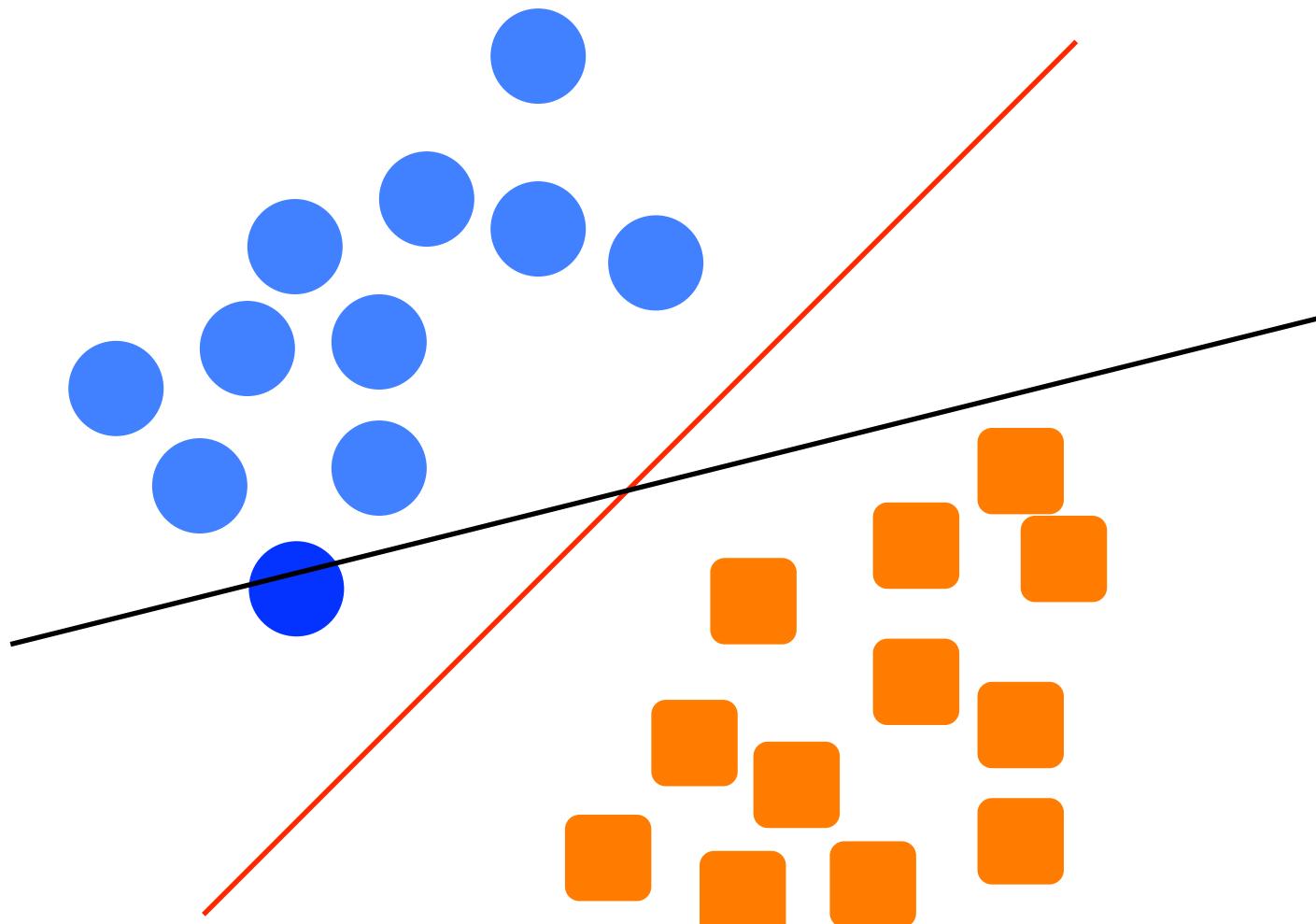
Support Vector Machine
Artificial Neural Network



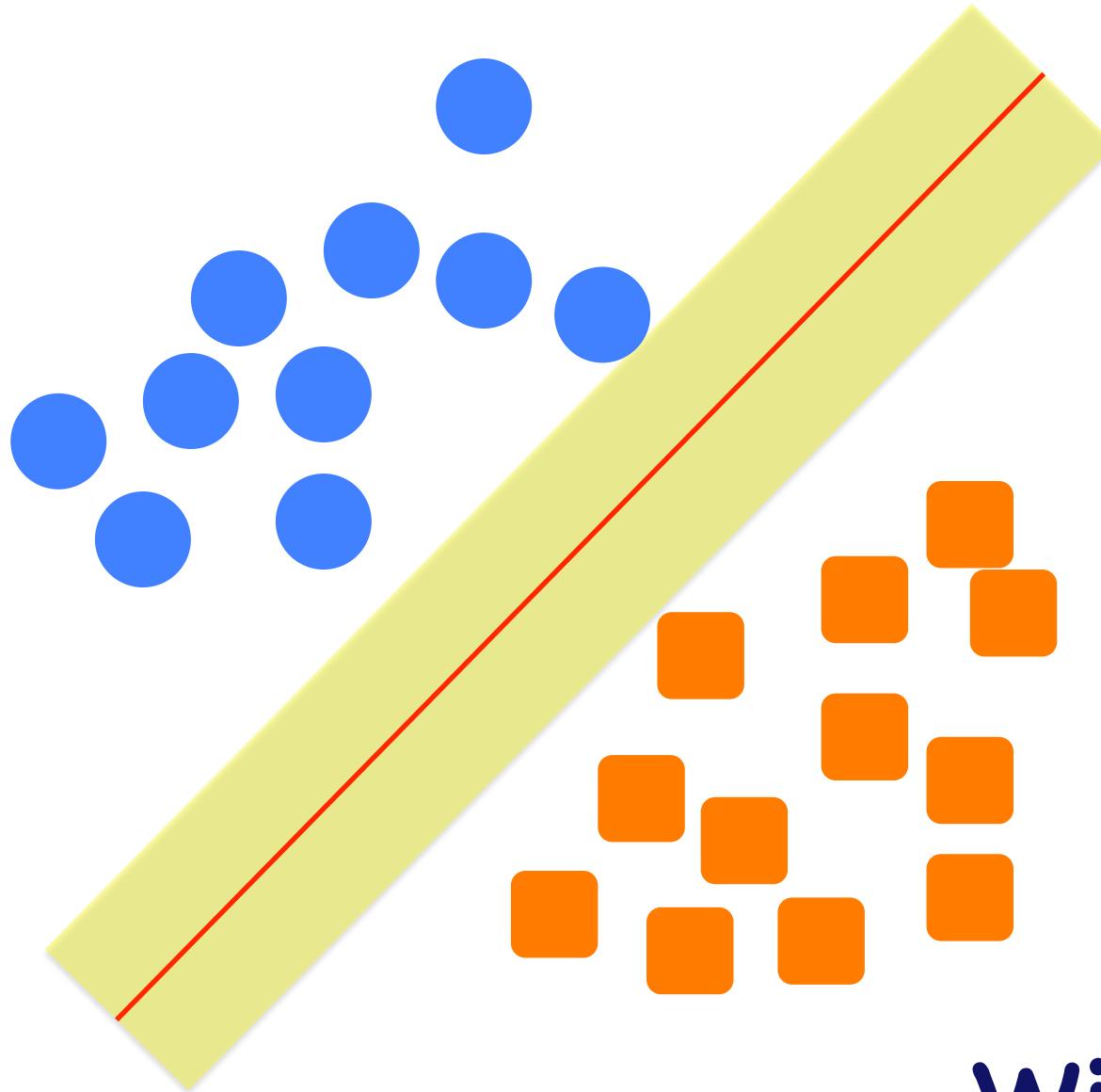




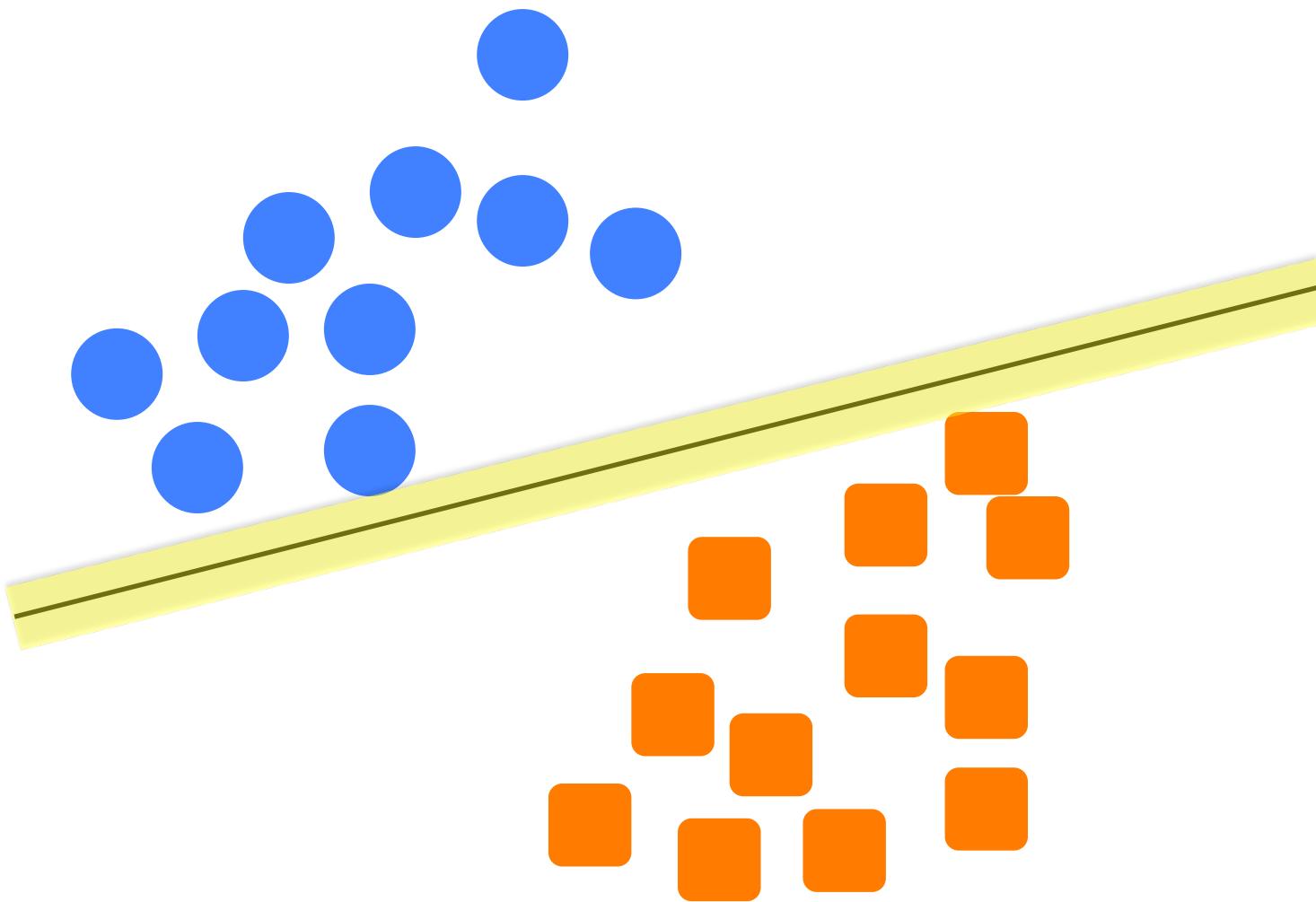
Which one is
better?



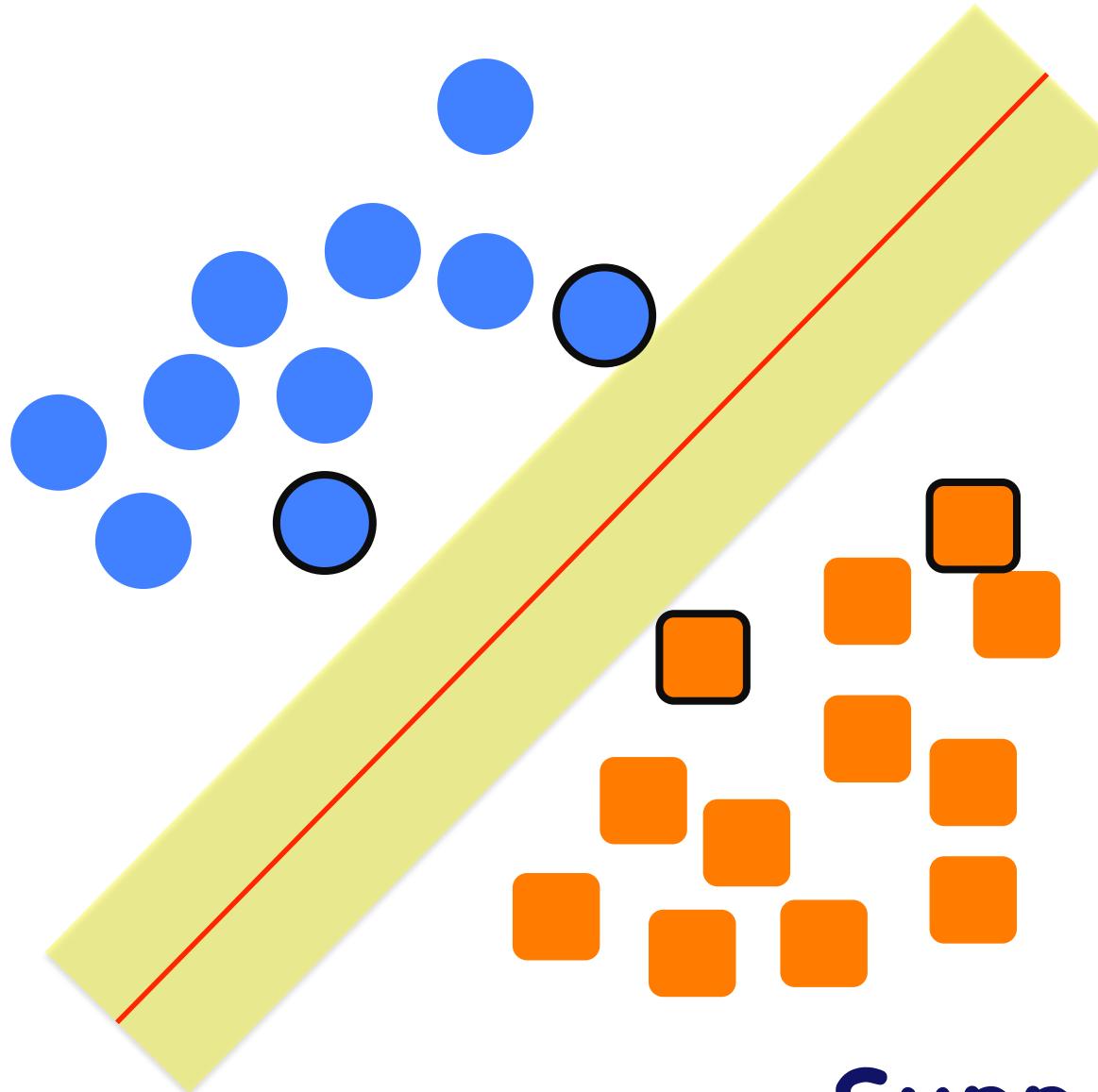
A new data point



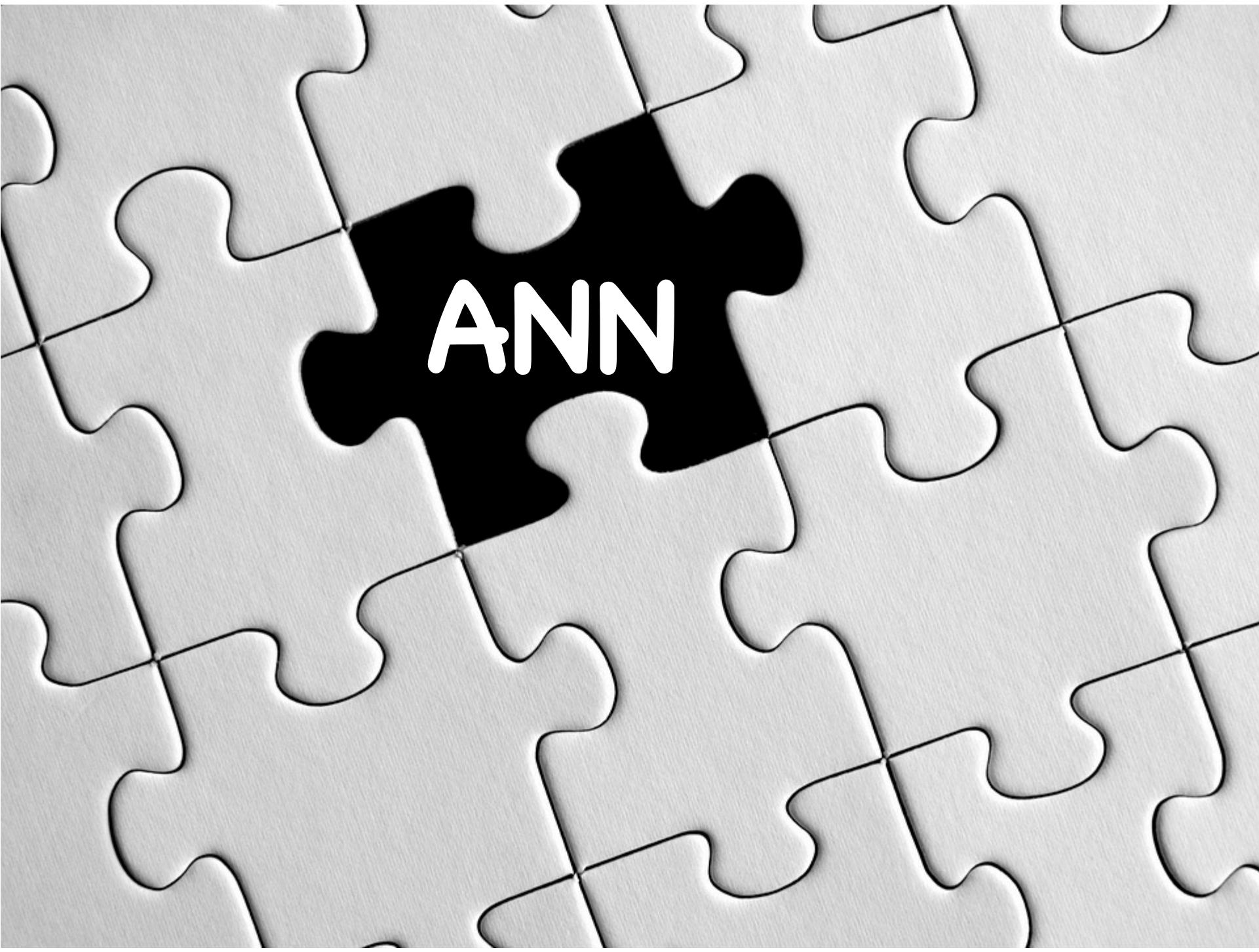
Wide margin



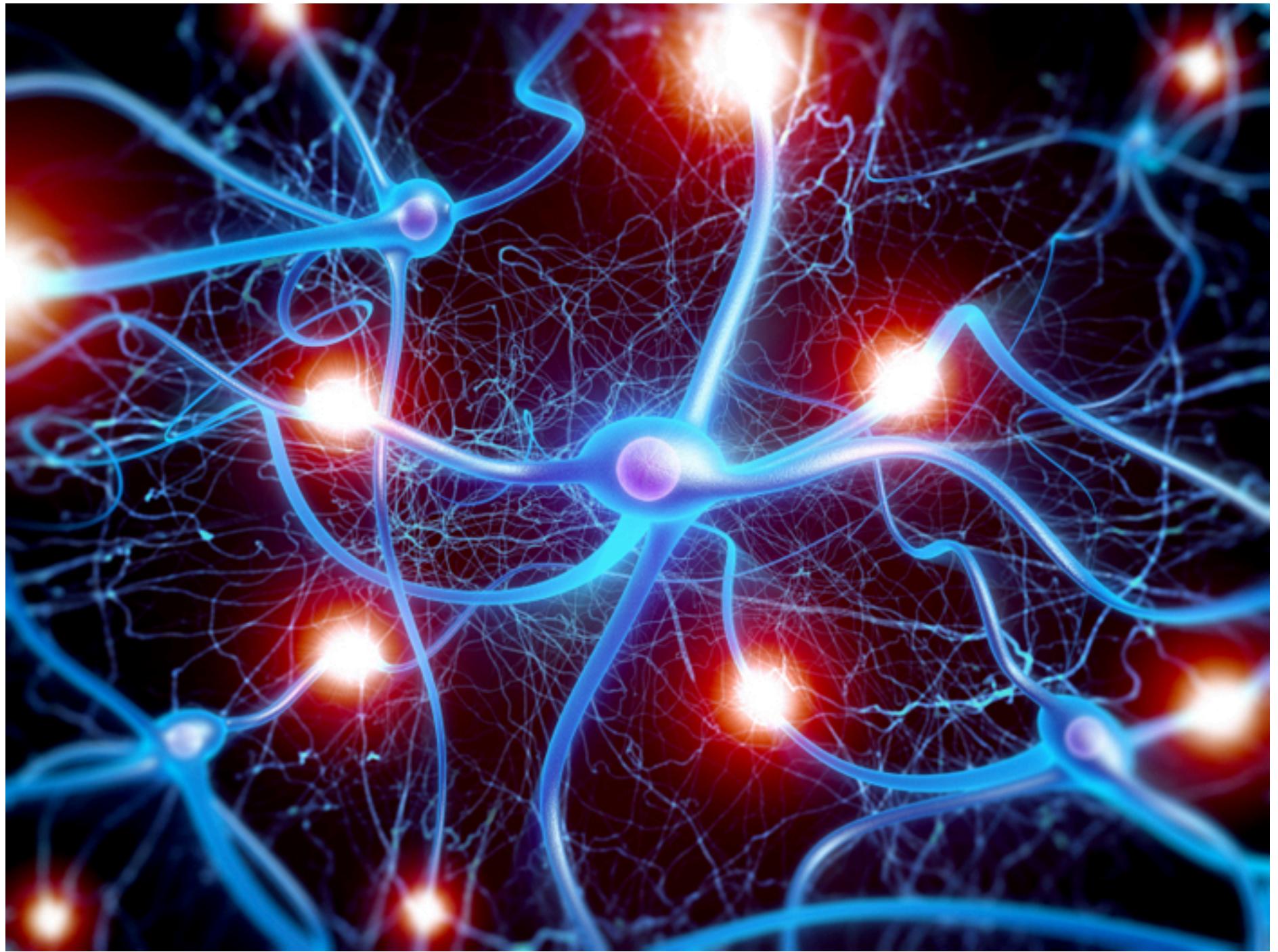
Narrow margin



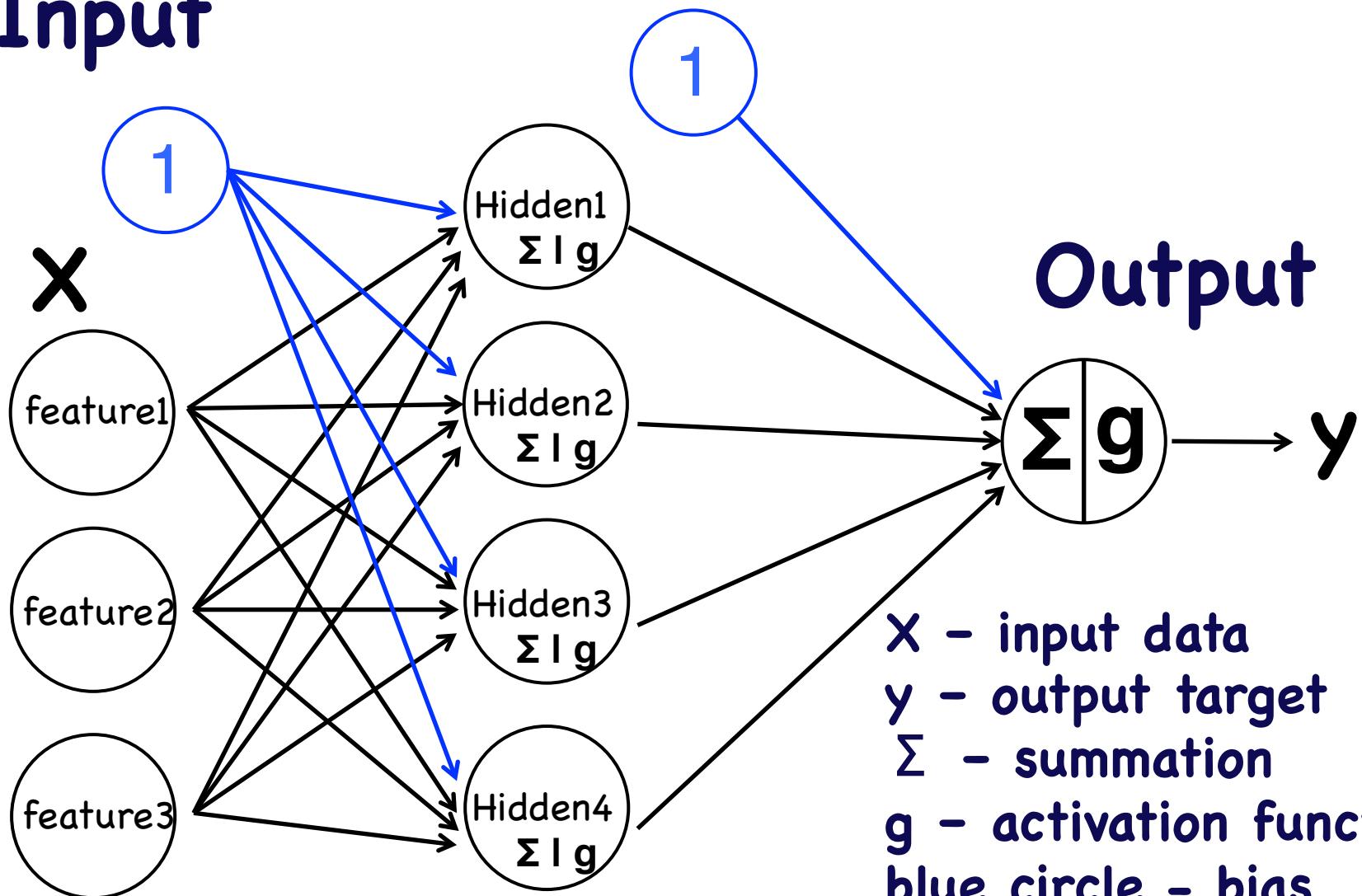
Support Vectors



ANN



Input



x - input data
 y - output target
 Σ - summation
 g - activation function
blue circle - bias

Output



YOU'RE IN MY SPOT

GRAPHICS GARAGE

Input



•
•
•



Intuitive Artificial Neural Network

$$w_1$$

$$w_2$$

$$w_n$$

$$F(\text{eye} \times w_1 + \text{nose} \times w_2 + \dots + \text{mouth} \times w_n)$$



Output



Input



•
•
•



Intuitive Artificial Neural Network

Output

$$F(\text{eye} \times w_1 + \text{nose} \times w_2 + \dots + \text{mouth} \times w_n)$$



error
feedback



Input



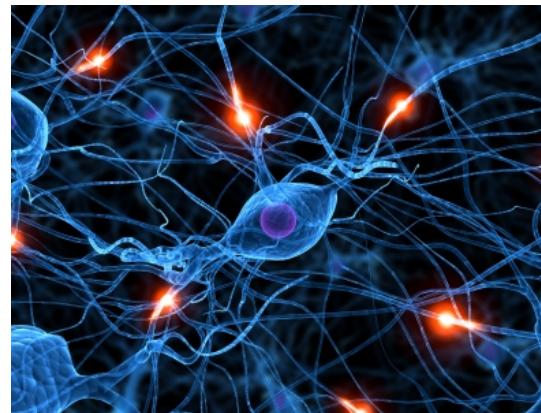
•
•
•



Intuitive Artificial Neural Network

Output

$$F(\text{eye} \times w_1 + \text{nose} \times w_2 + \dots + \text{mouth} \times w_n)$$



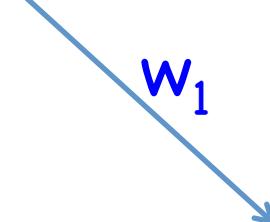
Input



•
•
•



Intuitive Artificial Neural Network

$$w_1$$


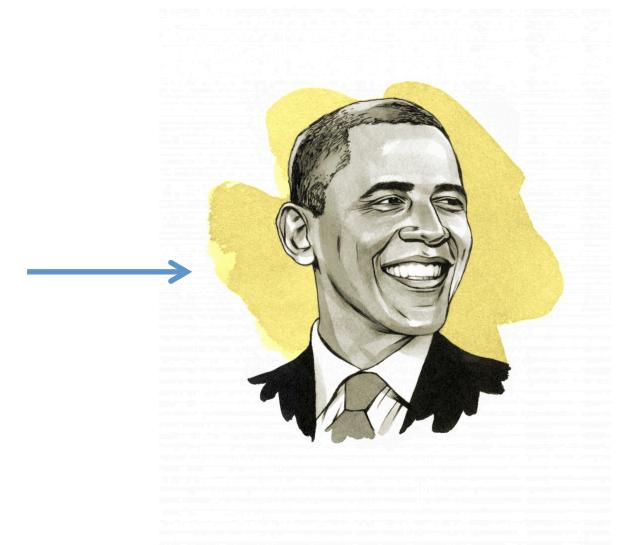
$$w_2$$


$$w_n$$


$$F(\text{eye} \times w_1 + \text{nose} \times w_2 + \dots + \text{mouth} \times w_n)$$



Output

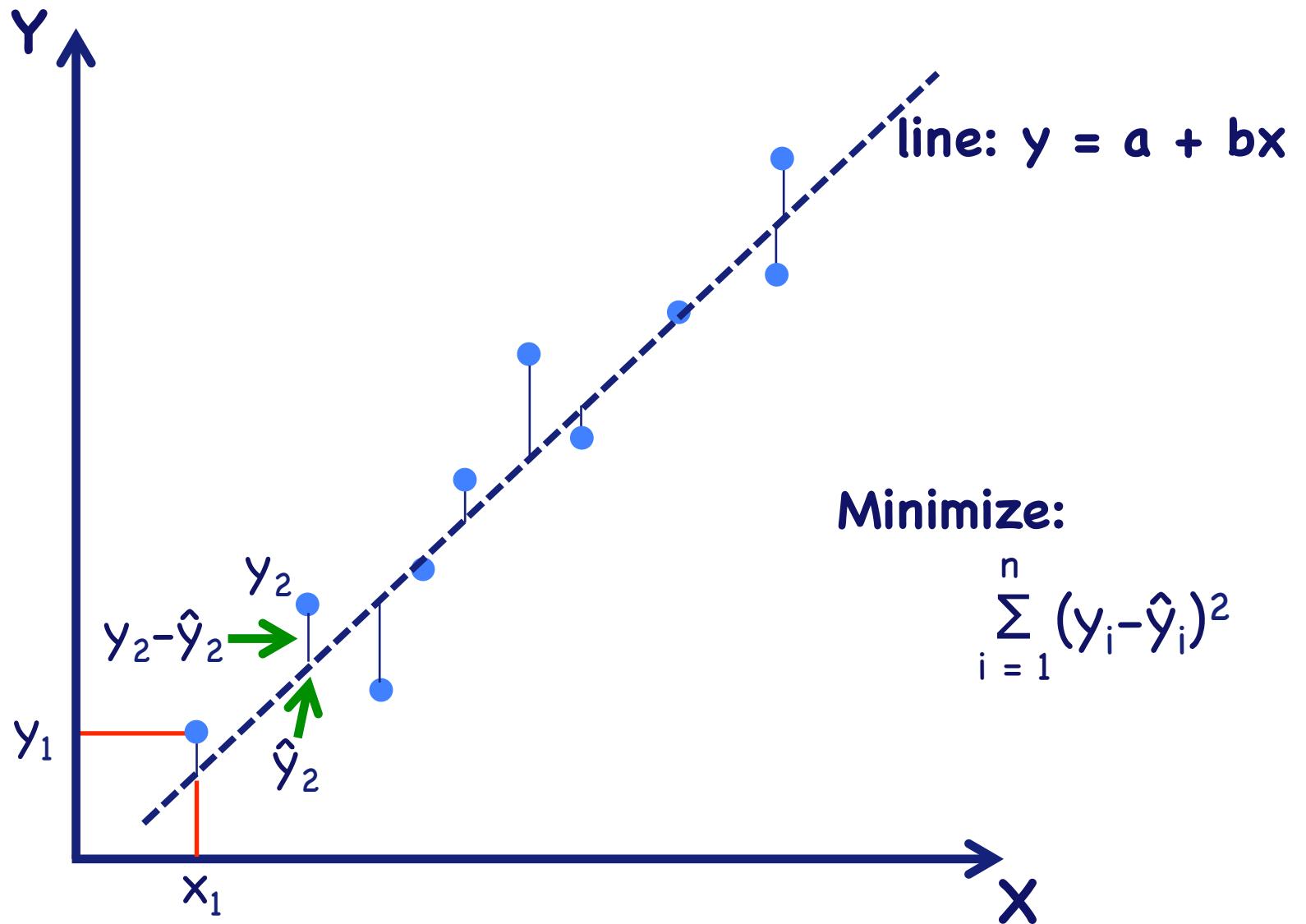


Go to notebook 02

Regression

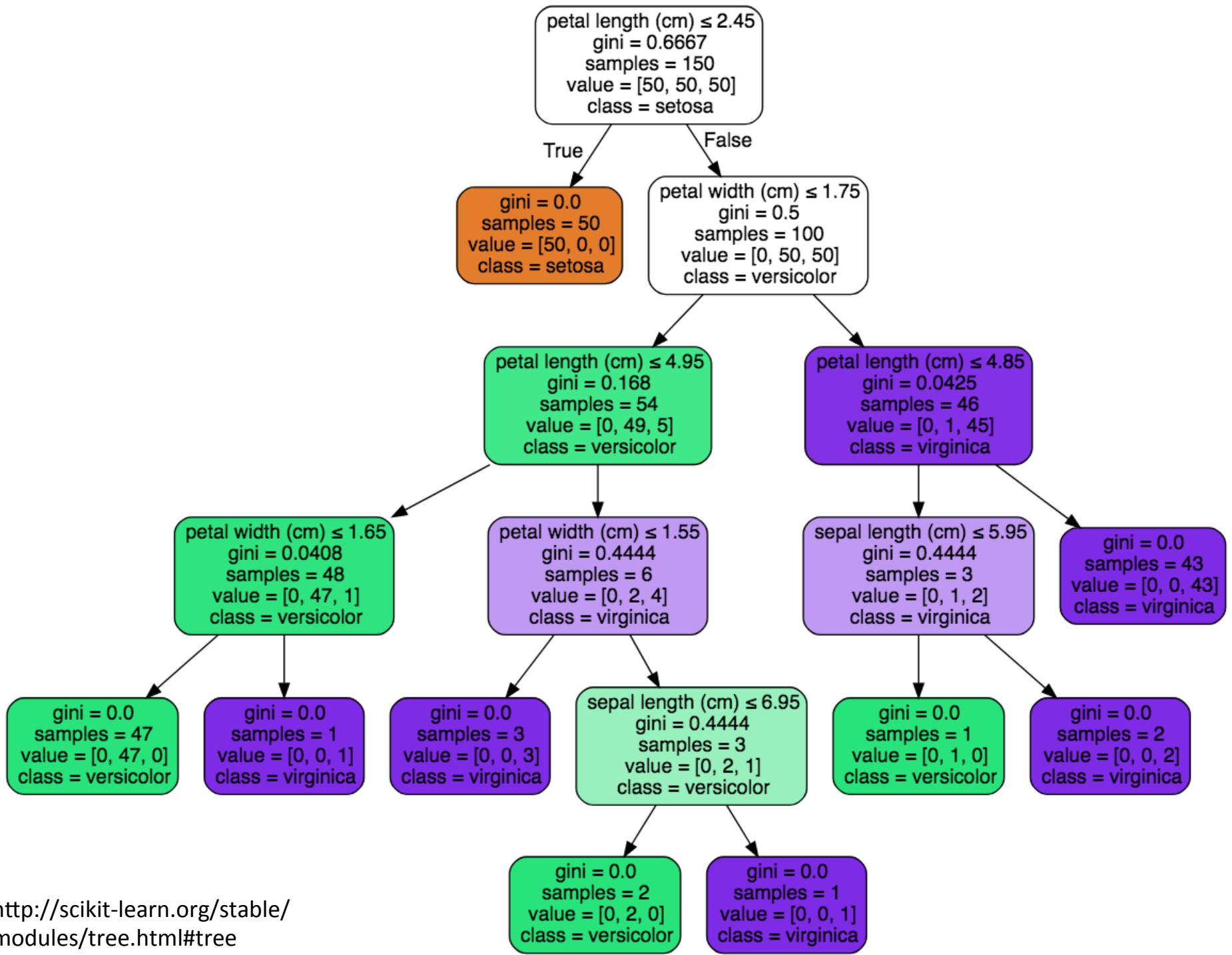
Supervised
learning

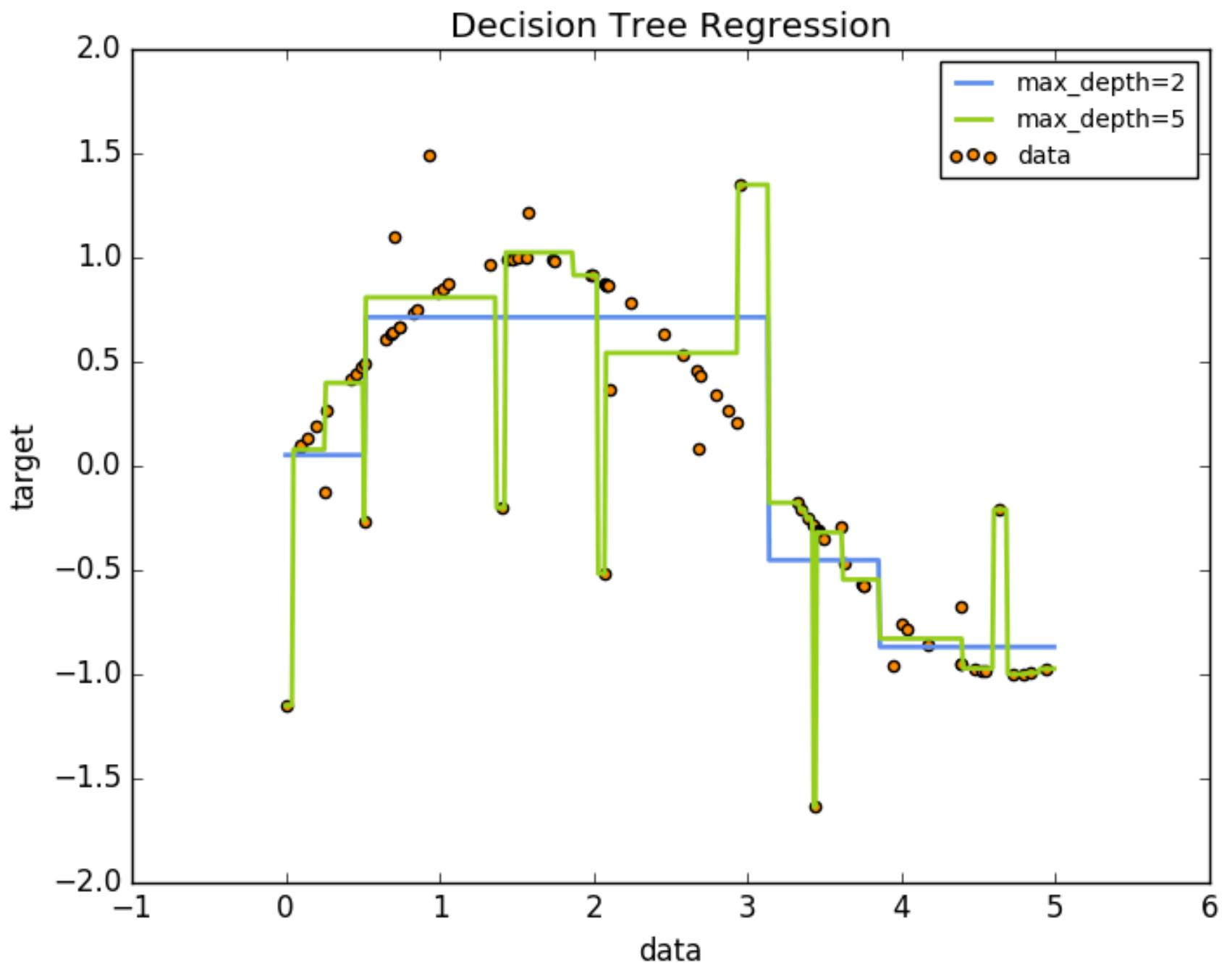
Simple linear regression



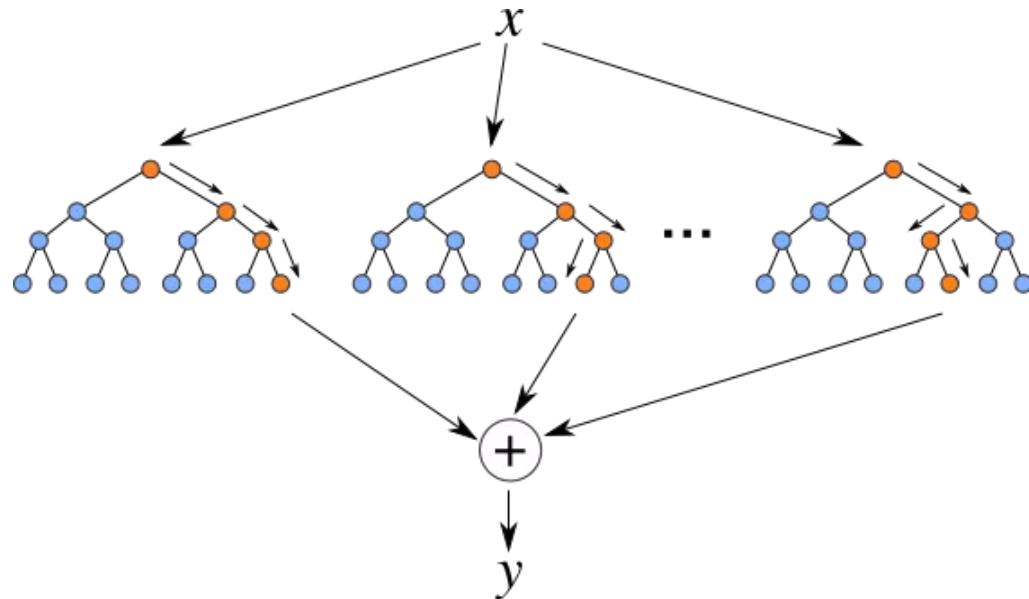
Random Forest



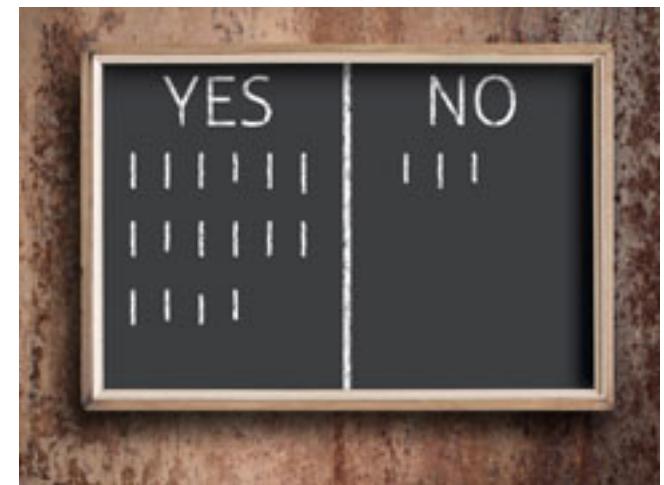




Random Forest



Majority vote



Go to notebook 03

More resources

