

# **Introduction to Machine Learning & SVM**

**Raffaele Scaringi**

# Agenda

## 1) Basics

- Structure of the classes
- Our roadmap

## 2) Intro to machine learning

- Defining learning
- Supervised vs Unsupervised learning
- The framework of learning algorithms

## 3) Example of Supervised learning

- Support Vector Machine (SVM)
- Optimization of SVM
- Extension of SVM to regression (SVR)

# Structure of the classes

- Recap of the previous class (aka, warm up) - 15 min
- Address questions from the previous class/assignment - 15 min
- New content - 30 min
- Coffee break - 10 min
- More content / Quiz - 30 min
- Hands-on tutorial - 30 min
- Questions - 20 min

# Our roadmap

Class 1: Intro to machine learning (ML) and SVM (Oct 22<sup>nd</sup>)

- Types of learning
- Hyperplanes and boundaries
- Support Vector Machine

Class 2: Optimizers and the Perceptron (pt. 1) (Oct 27<sup>th</sup>)

- Regression with and without ML
- Minimizing loss functions
- Optimizers
- Perceptron

# Our roadmap

## Class 3: Perceptron (pt. 2) and Neural Networks (pt. 1) (Oct 29<sup>th</sup>)

- Perceptron as a regressor
- Activation functions
- When Perceptrons will fail you
- Neural Networks

## Class 4: Neural Networks (pt. 2) (Nov 3<sup>rd</sup>)

- How to train your network
- Hyperparameter search
- Using Weights and Biases to inspect your models

# Our roadmap

## Class 5: Convolutional Neural Networks (Nov 5<sup>th</sup>)

- Neural networks for spatial data
- Kernels, padding, pooling
- Study case with satellite images

## Class 6: What you need (Nov 10<sup>th</sup>)

- ??

# Our roadmap

Class 7: BYOP (Bring Your Own Paper) (Nov 18<sup>th</sup>)

- Pick a paper related to your field that is using machine learning
- Challenge me!

# What is machine learning?



[https://pe.app/response\\_links/0bcc05e7-ef10-4317-b35e-065e69fe1a83/start](https://pe.app/response_links/0bcc05e7-ef10-4317-b35e-065e69fe1a83/start)

# What is machine learning?

Machine learning is the process of identifying patterns in data.

# Two kinds of machine learning

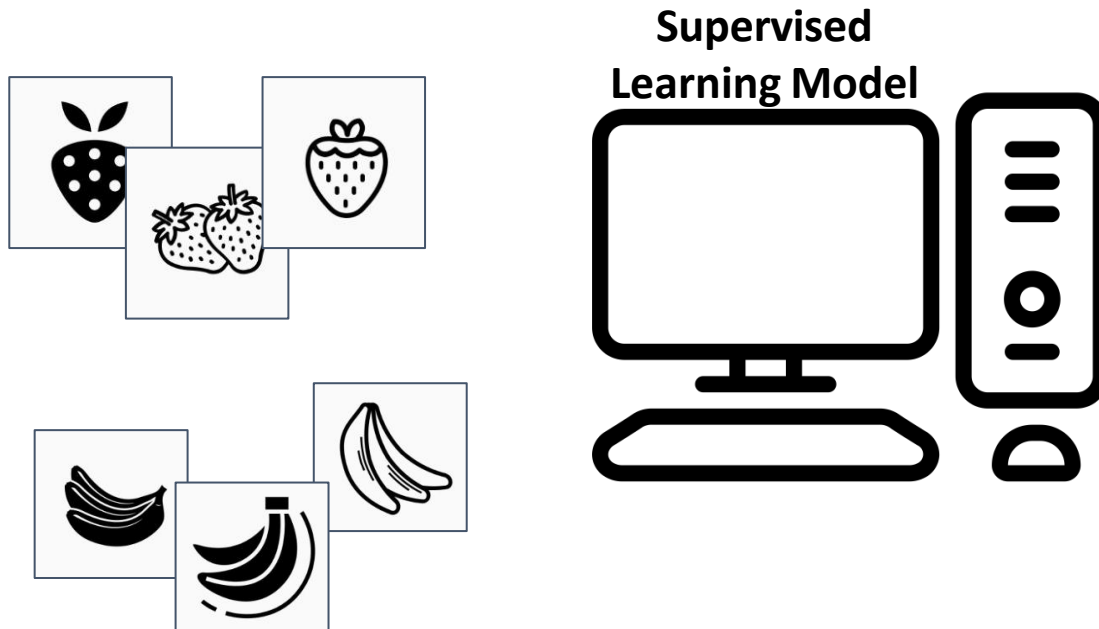
## Supervised learning

- Have a bunch of labelled data,  
want to label new data

# Two kinds of machine learning

## Supervised learning

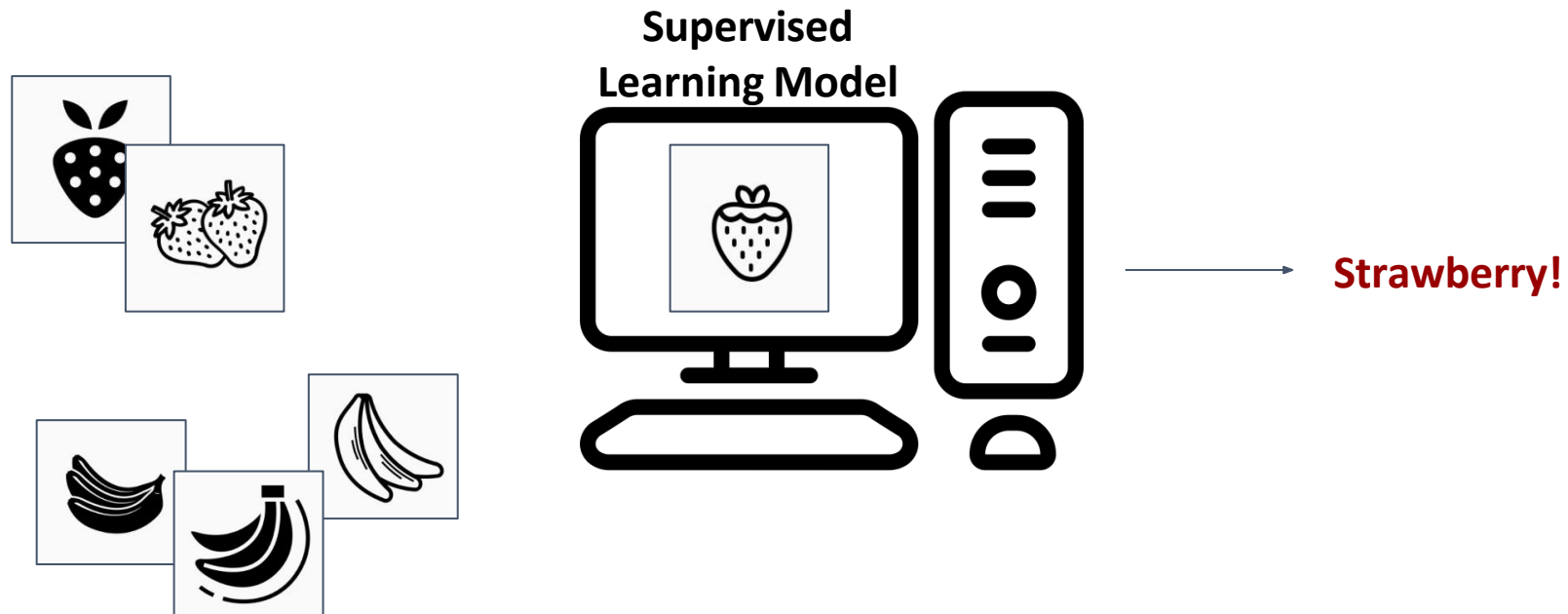
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

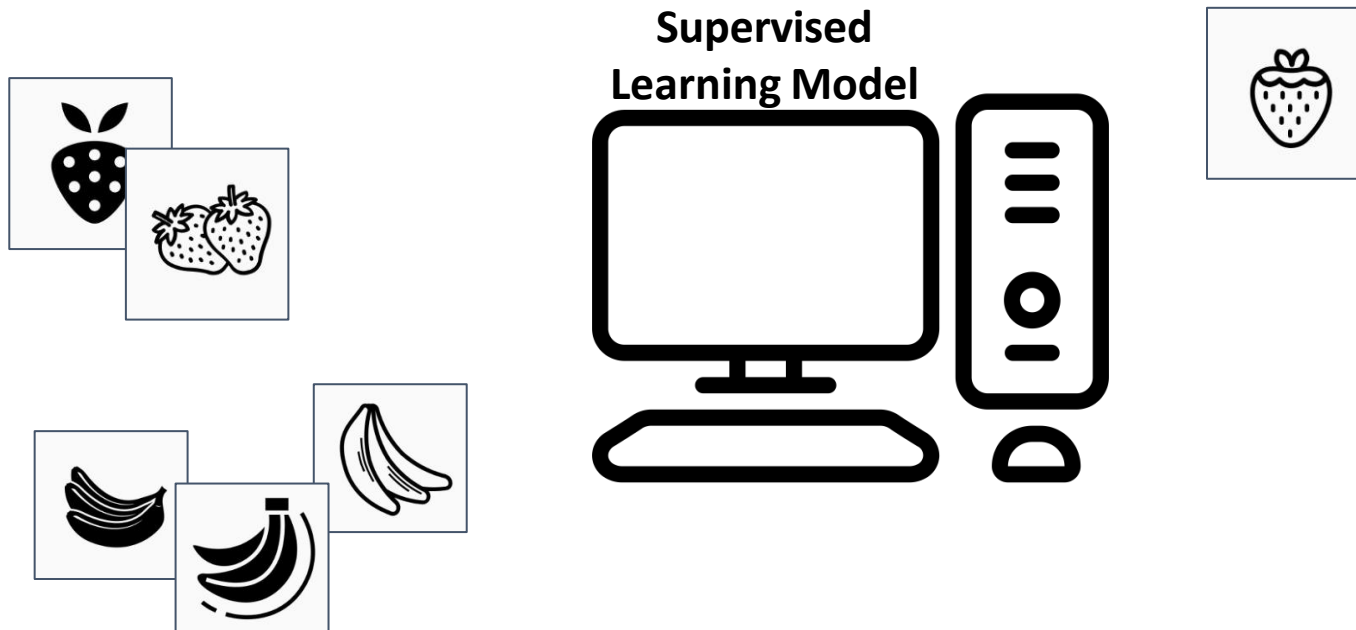
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

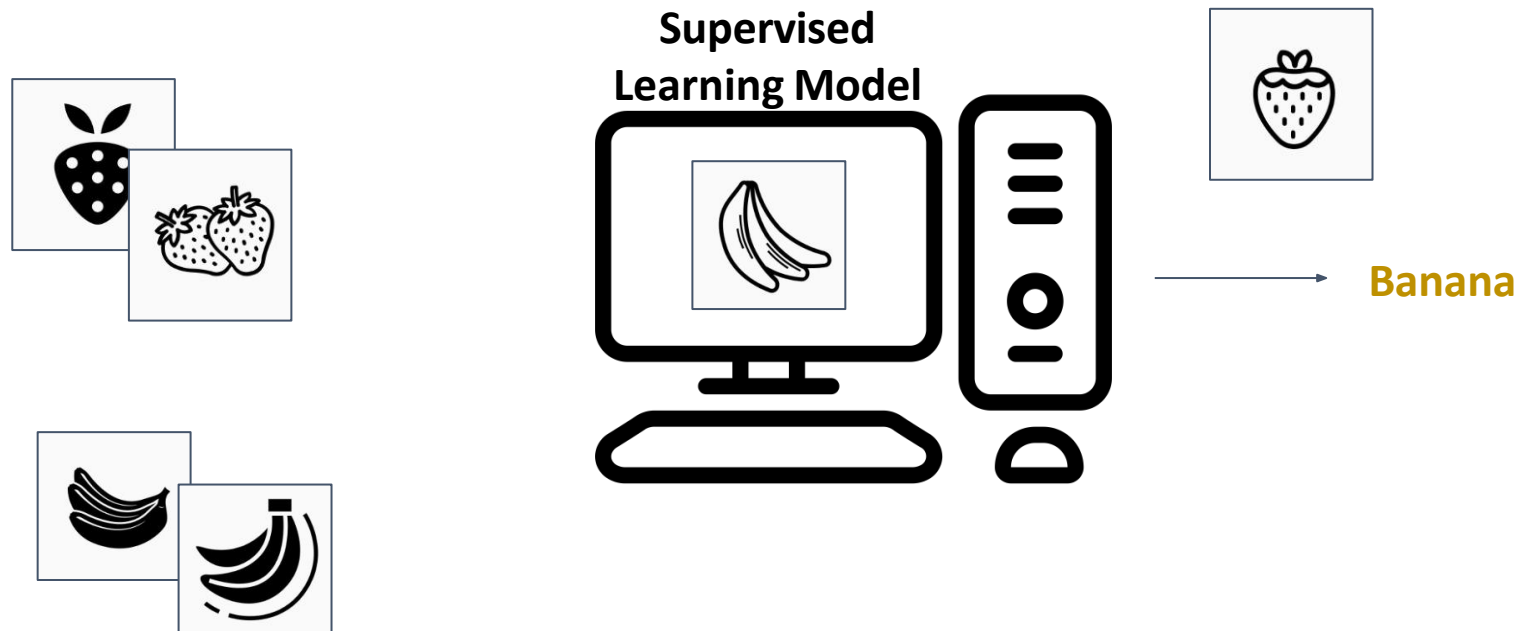
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

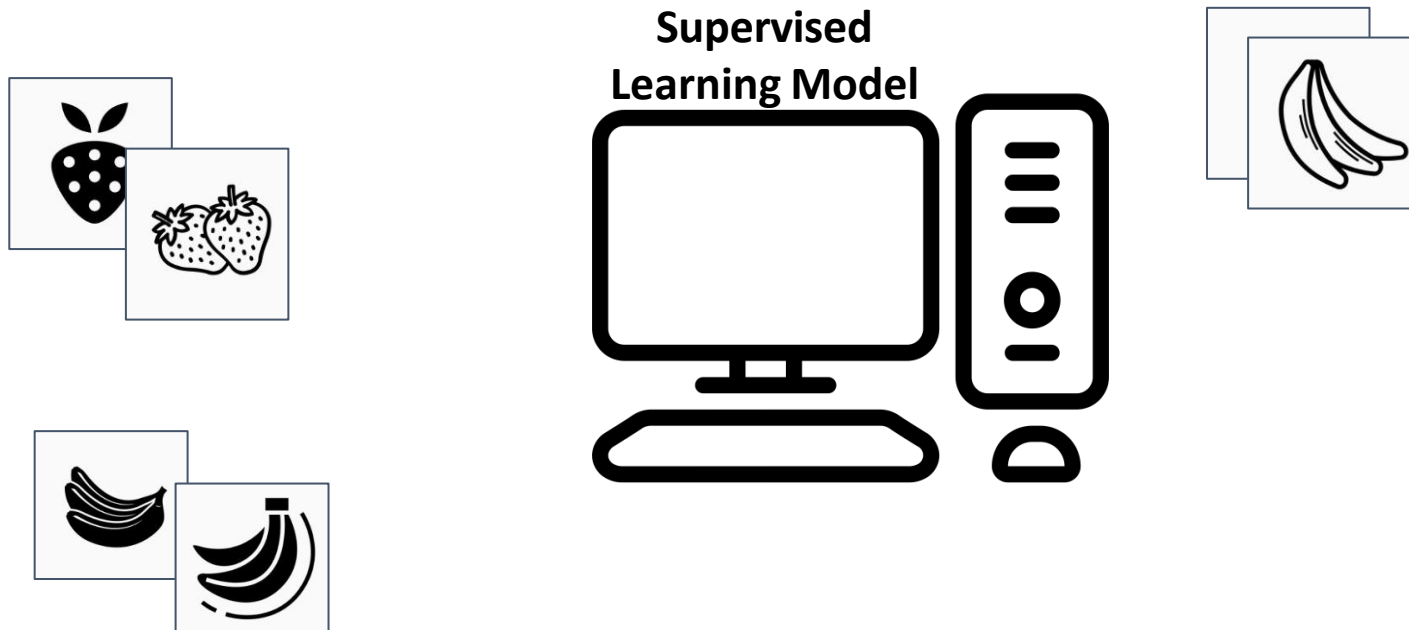
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

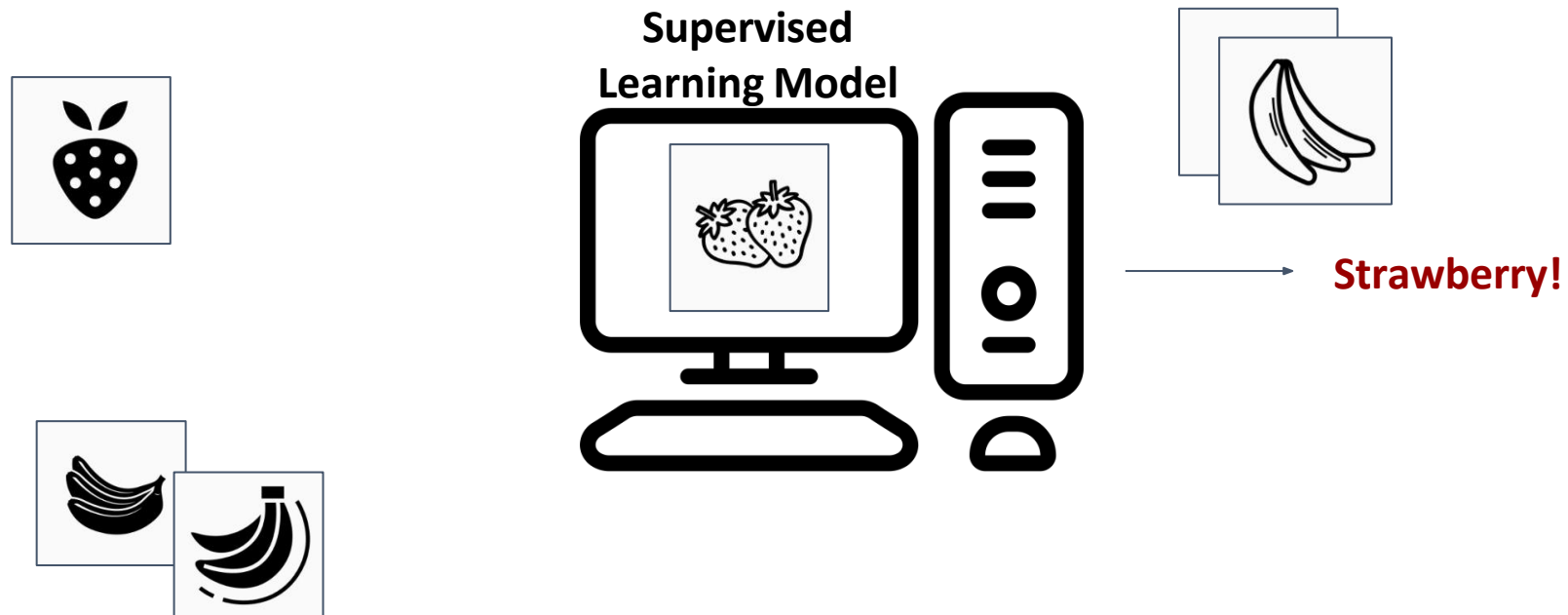
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

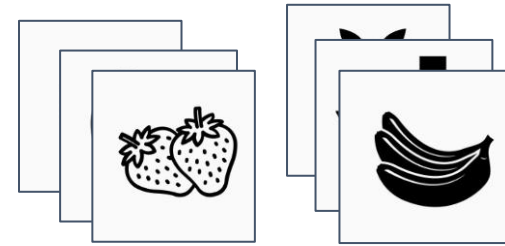
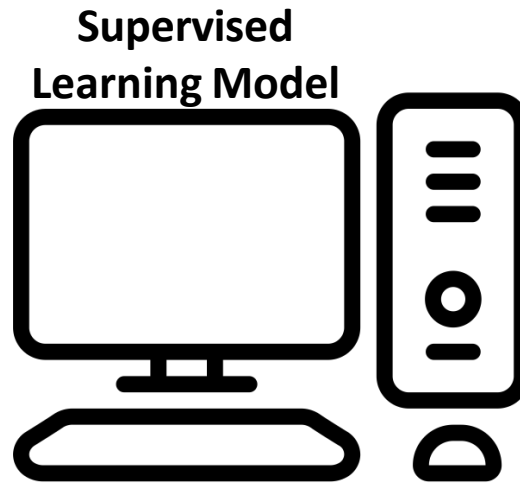
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

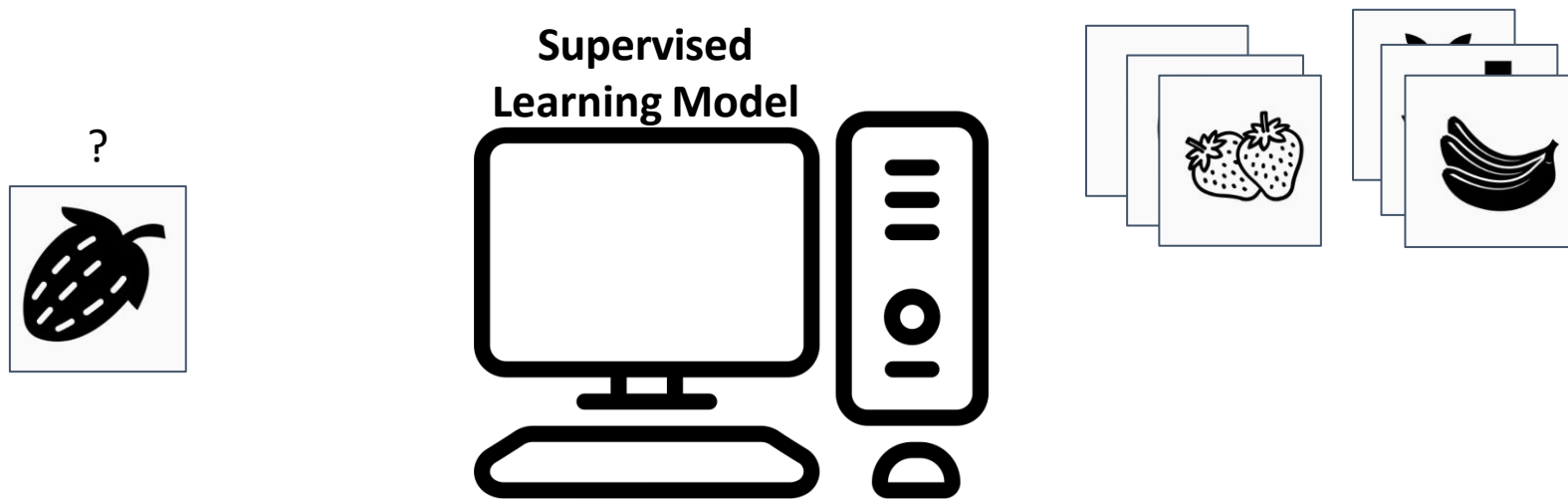
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

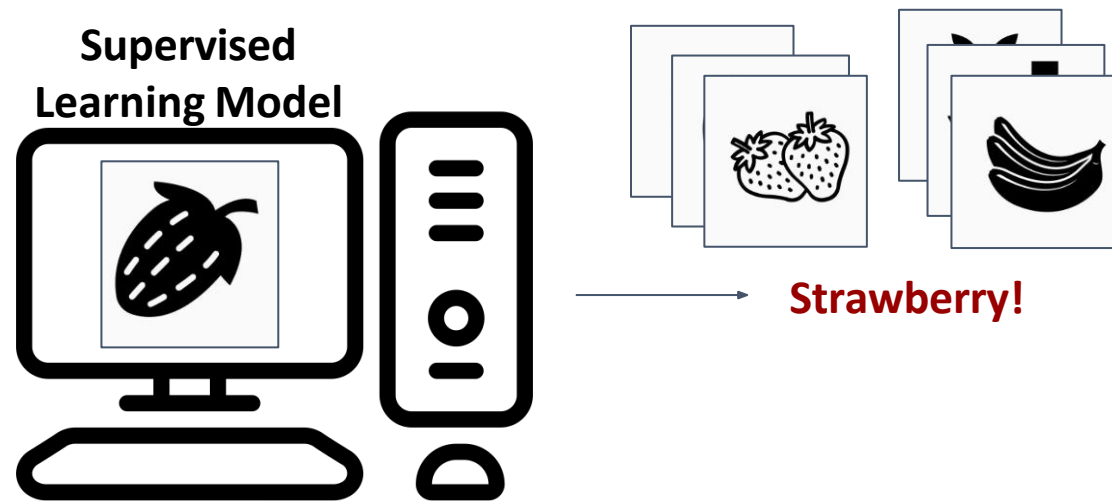
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

## Supervised learning

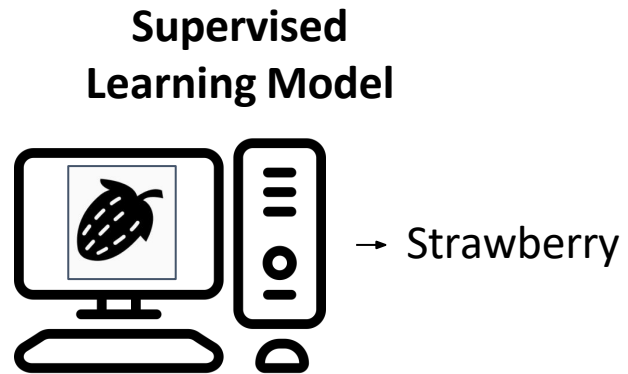
- Have a bunch of labelled data, want to label new data



# Two kinds of machine learning

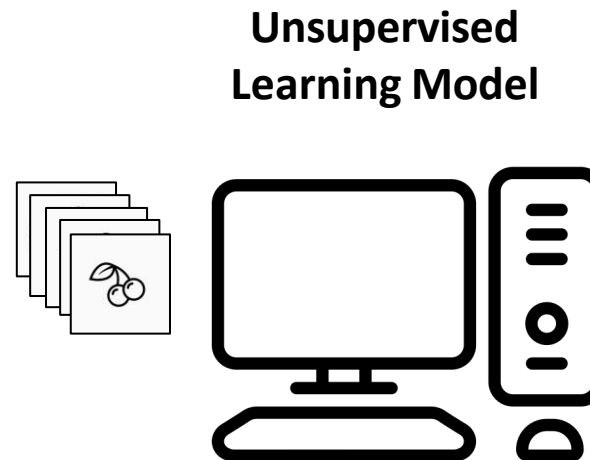
## Supervised learning

- Have a bunch of labelled data, want to label new data



## Unsupervised learning

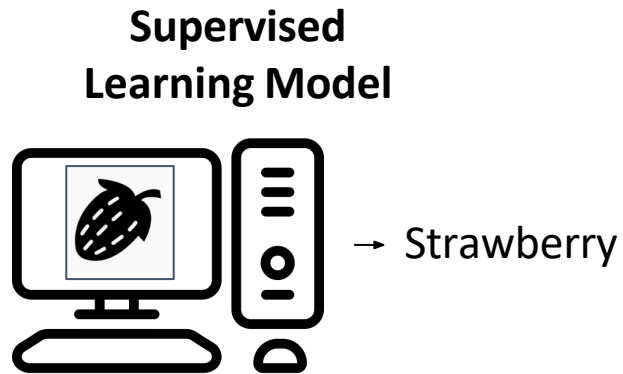
- Have a bunch of unlabeled data, want to organize it



# Two kinds of machine learning

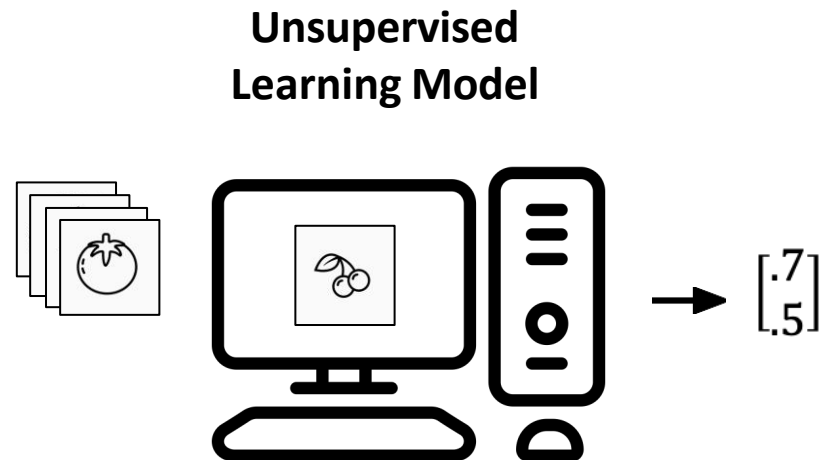
## Supervised learning

- Have a bunch of labelled data, want to label new data



## Unsupervised learning

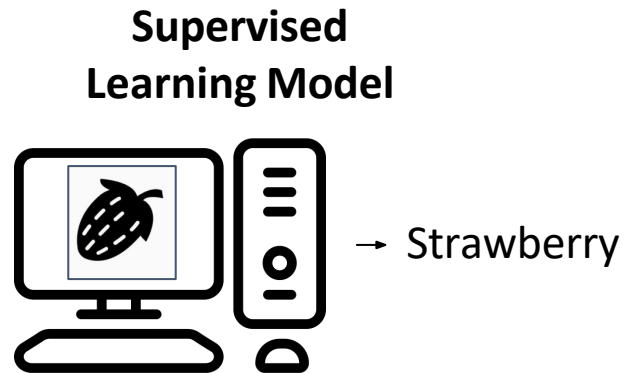
- Have a bunch of unlabeled data, want to organize it



# Two kinds of machine learning

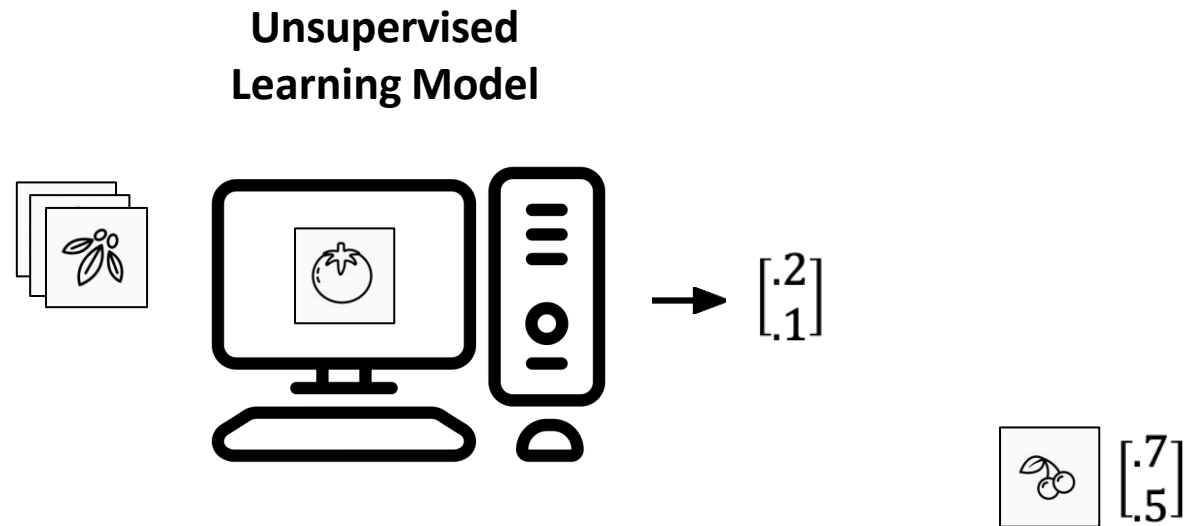
## Supervised learning

- Have a bunch of labelled data, want to label new data



## Unsupervised learning

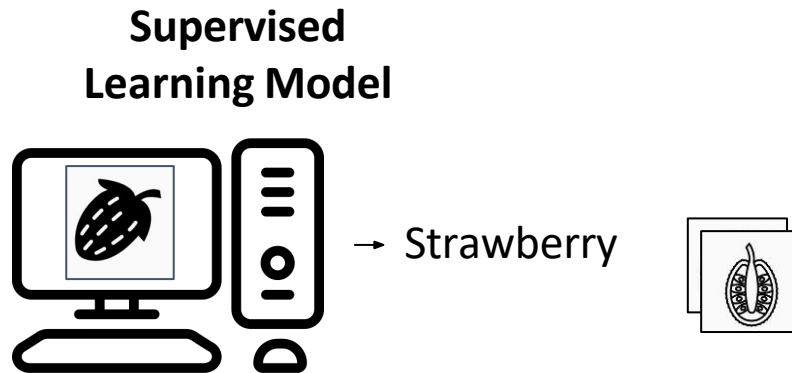
- Have a bunch of unlabeled data, want to organize it



# Two kinds of machine learning

## Supervised learning

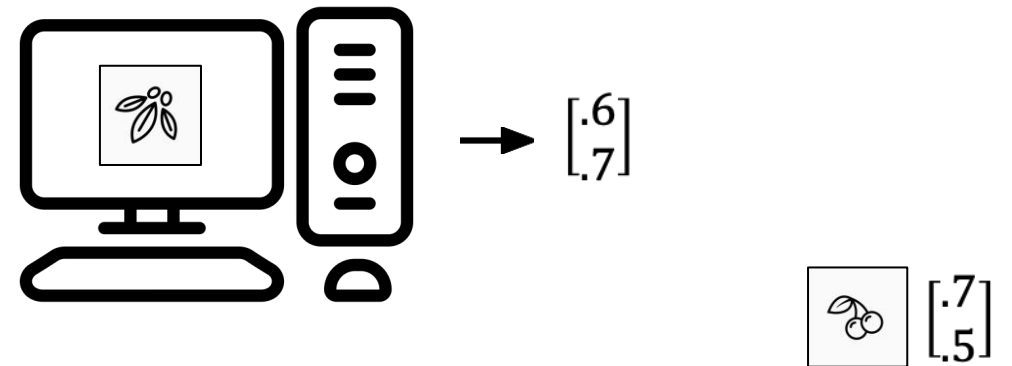
- Have a bunch of labelled data, want to label new data



## Unsupervised learning

- Have a bunch of unlabeled data, want to organize it

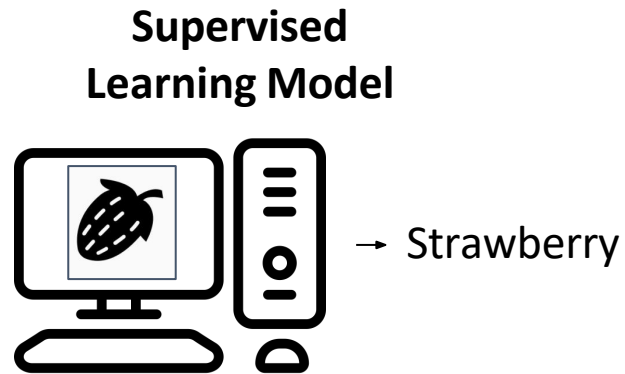
**Unsupervised Learning Model**



# Two kinds of machine learning

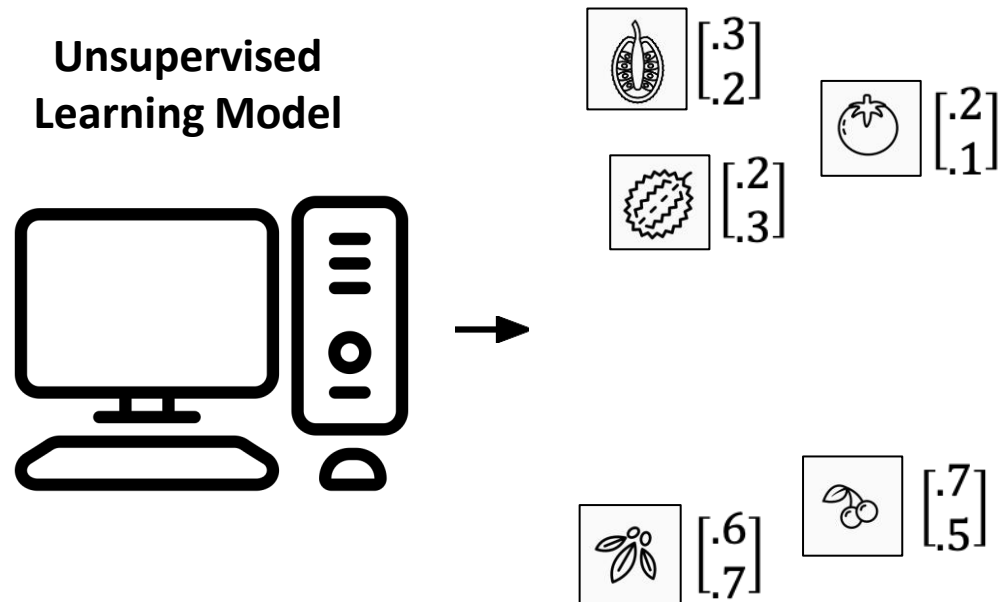
## Supervised learning

- Have a bunch of labelled data, want to label new data



## Unsupervised learning

- Have a bunch of unlabeled data, want to organize it



# Two kinds of machine learning

## Supervised learning

- Have a bunch of labelled data, want to label new data
- Learn a function  $f(X) \rightarrow Y$  where all values of  $Y$  are known for some samples of  $X$

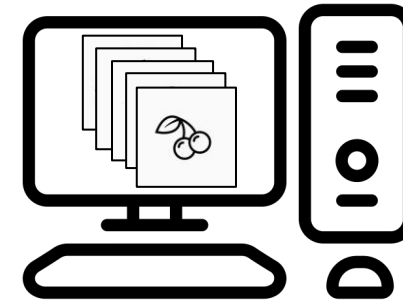
Supervised  
Learning Model



## Unsupervised learning

- Have a bunch of unlabeled data, want to organize it
- Learn an embedding  $f(X) \rightarrow Y, X \in \mathbb{R}^n, Y \in \mathbb{R}^m, n \gg m$
- Lower dimensional, easier to interpret (e.g. as clusters)

Unsupervised  
Learning Model



# Learning algorithms

“A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P** , if its performance at tasks in **T** , as measured by **P** , improves with experience **E**.”

## Tasks (T)

Transcription  
Machine Translation  
Classification  
Anomaly detection  
Synthesis and sampling  
⋮  
Regression

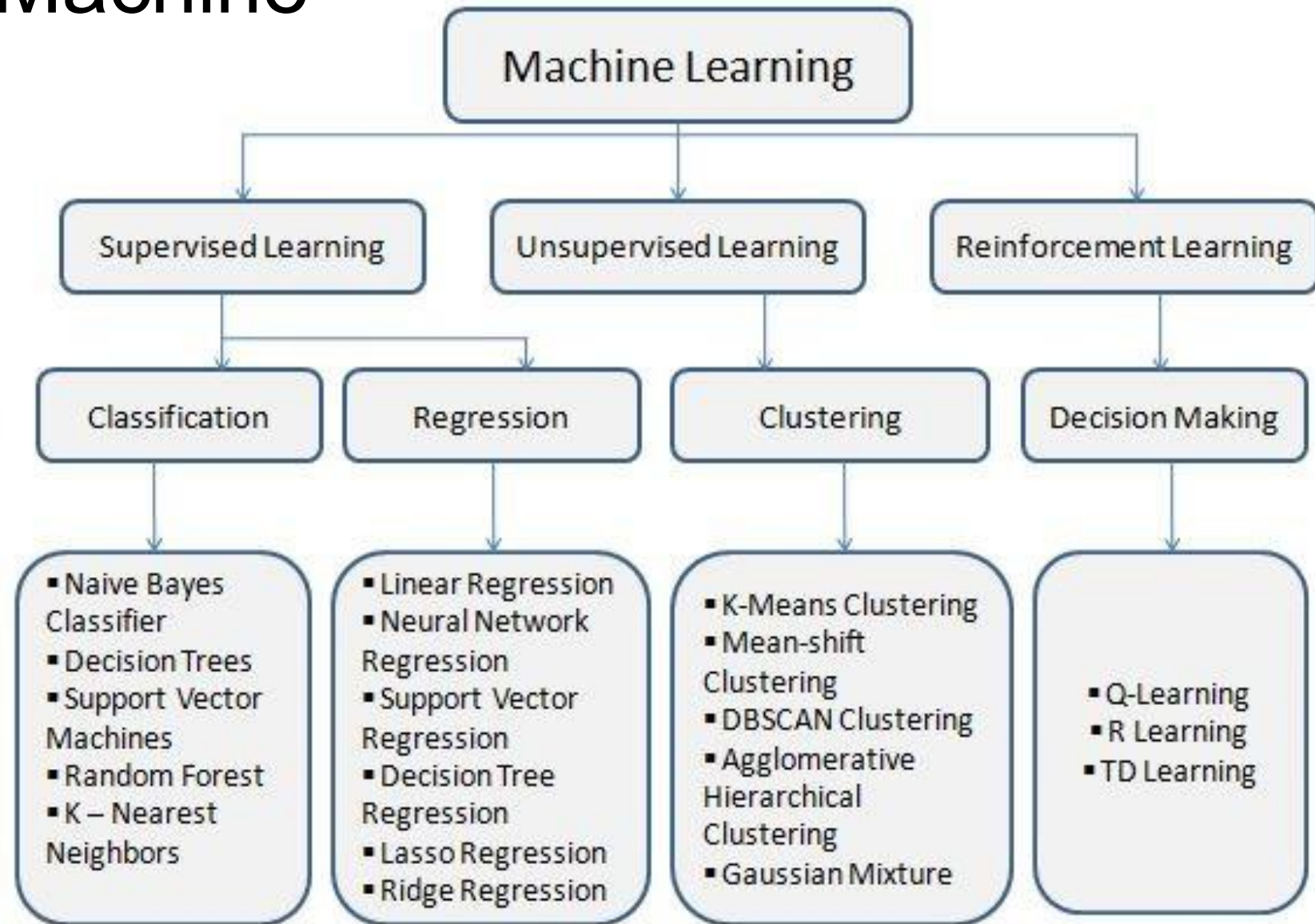
## Performance (P)

Accuracy rate  
  
Adjusted  $R^2$   
RMSE/MSE/MAE

## Experience (E)

Supervised Learning  
  
Unsupervised Learning  
  
Reinforcement Learning

# Types of Machine Learning



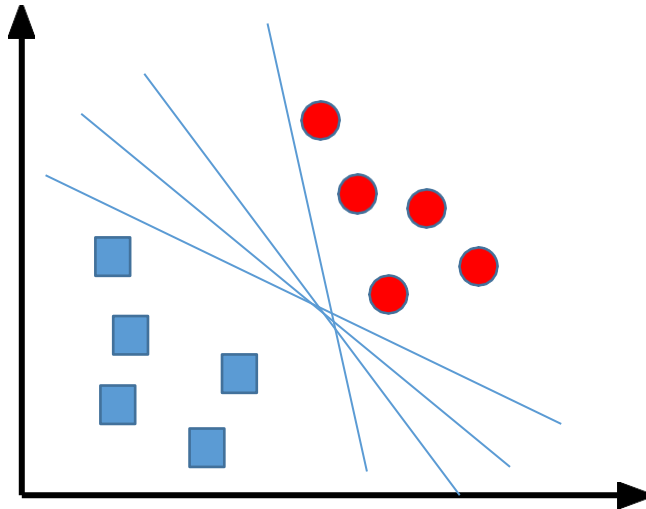
# Time for a little quiz!



[https://pe.app/response\\_links/46711d04-0b3a-48a2-8da0-34f067354e73/start](https://pe.app/response_links/46711d04-0b3a-48a2-8da0-34f067354e73/start)

# Decision Boundaries

Find a hyperplane in an N-dimensional space that distinctly classifies the data points.



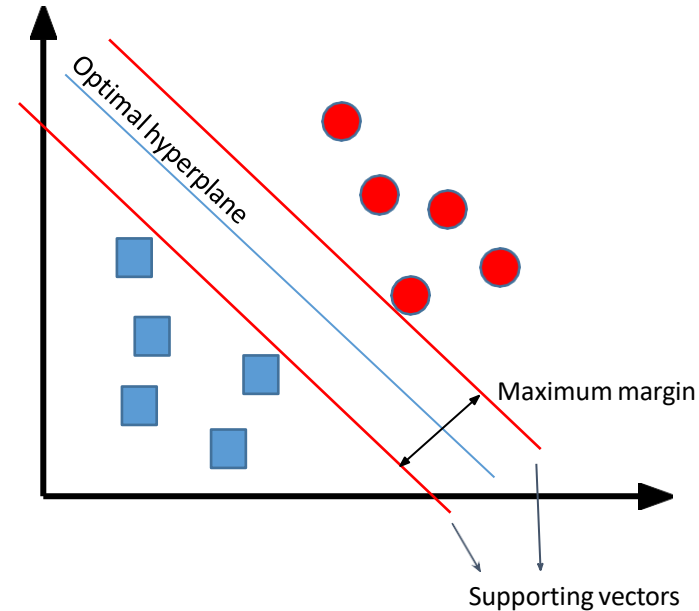
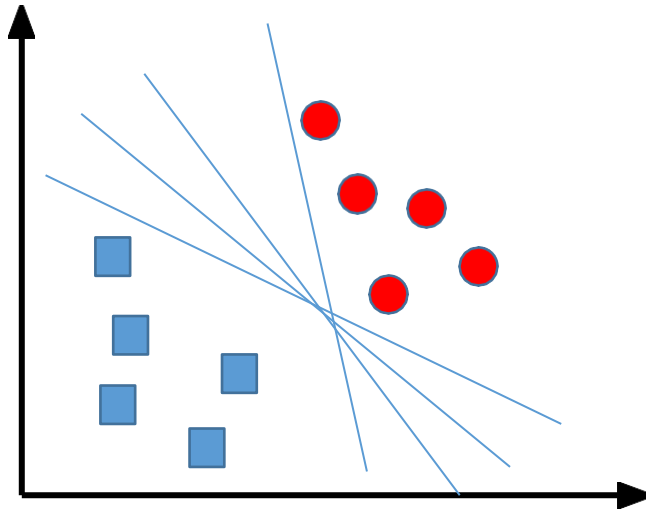
What is the correct decision boundary for this problem?



[https://pe.app/response\\_links/2b73958e-0c8c-493c-a4a6-716765fab25e/start](https://pe.app/response_links/2b73958e-0c8c-493c-a4a6-716765fab25e/start)

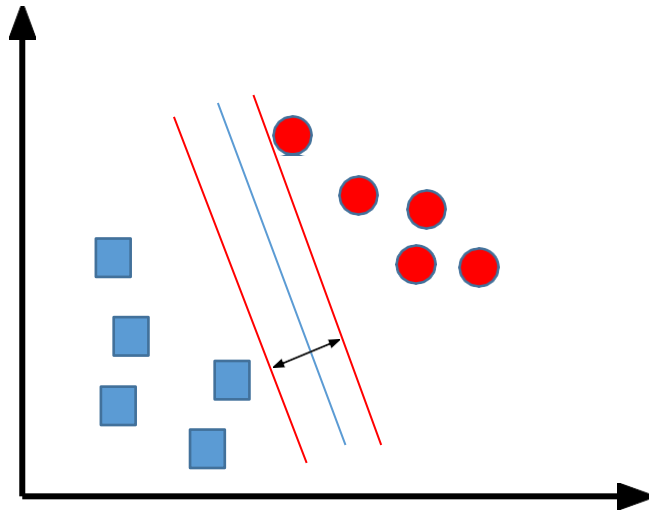
# Support Vector Machine

Find **the optimal** hyperplane in an N-dimensional space that distinctly classifies the data points.

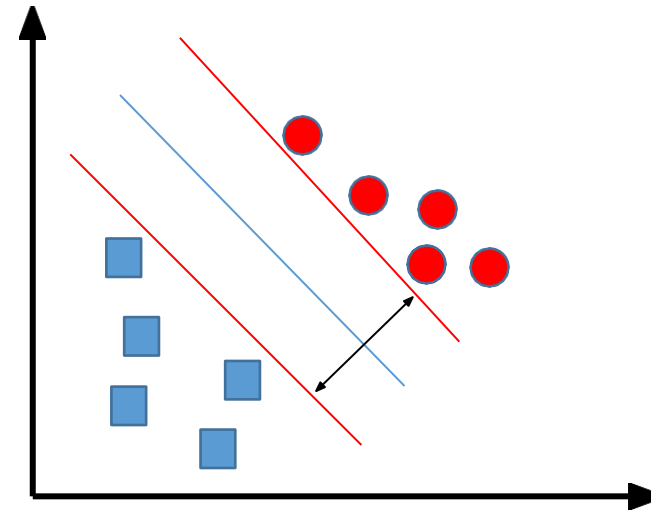


# Support Vector Machine

Maximize the margin of the classifier

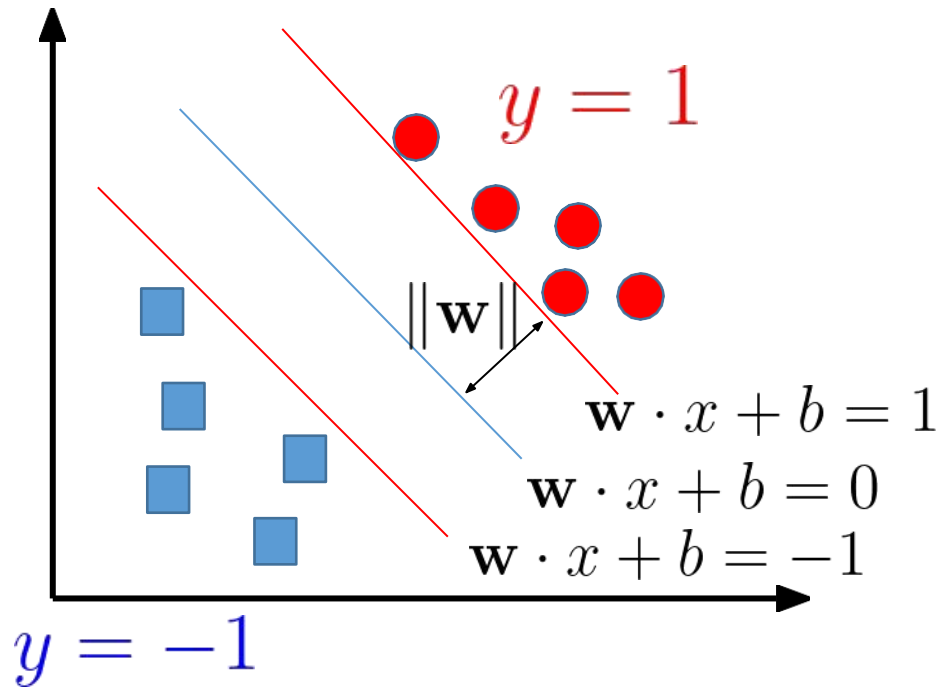


Small margin



Large margin

# Support Vector Machine



Hyperplane equation:  $f(x) = \mathbf{w} \cdot x + b$

Distance (D) from a point to the hyperplane

$$D = \frac{|\mathbf{w} \cdot x + b|}{\|\mathbf{w}\|}$$

Minimize the weights, increase distance

Classification task

$$\begin{cases} wx_i + b \geq +1 & \text{when } y_i = +1 \\ wx_i + b \leq -1 & \text{when } y_i = -1, \end{cases}$$

# SVM Optimization

Hinge loss function

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

Loss function for the SVM

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^n (1 - y_i \langle x_i, w \rangle)_+$$

Updating the weights:

No misclassification

$$w = w - \alpha \cdot (2\lambda w)$$

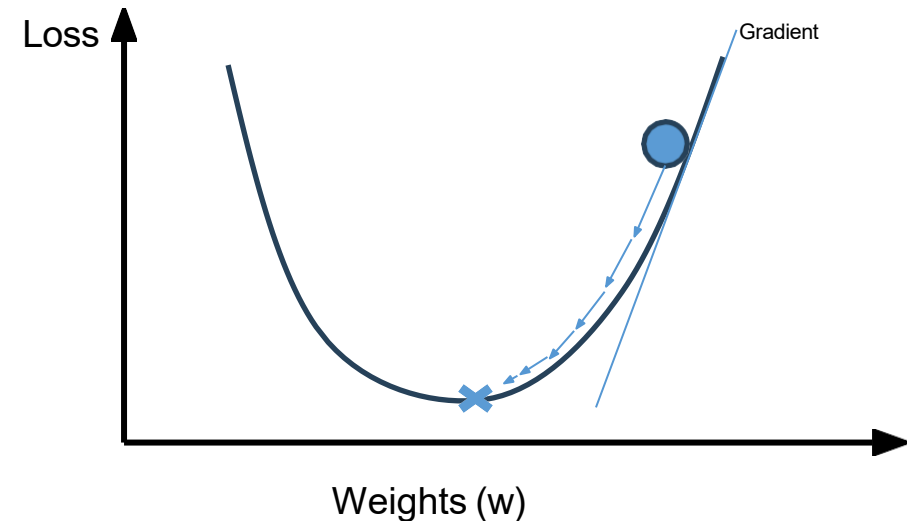
Misclassification

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

Gradients

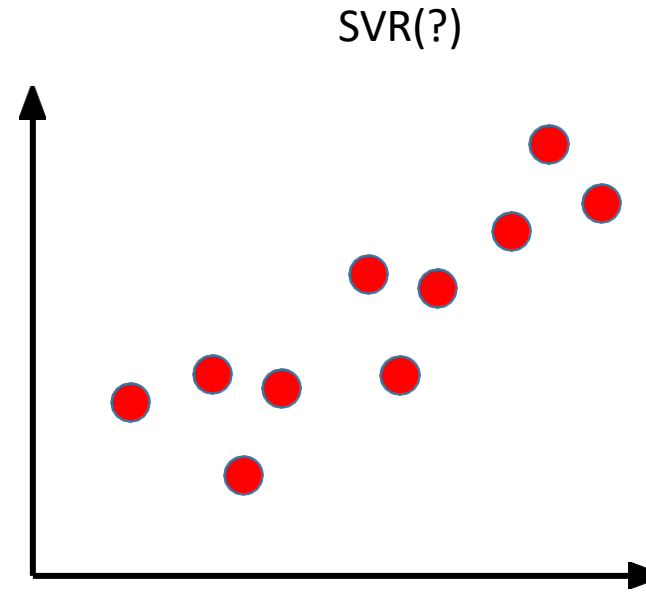
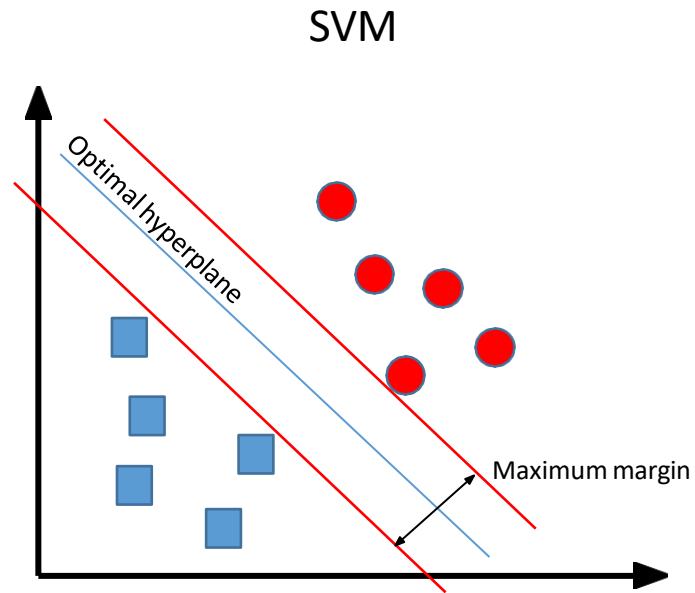
$$\frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$



# Support Vector Machine for Regression

How do I turn the SVM into a SVR?



# SVR Optimization

Loss

$$L(y, f(x, \mathbf{w})) = \begin{cases} 0, & |y - f(x, \mathbf{w})| \leq \epsilon \\ |y - f(x, \mathbf{w})| & \text{o.w.} \end{cases}$$

Constraints

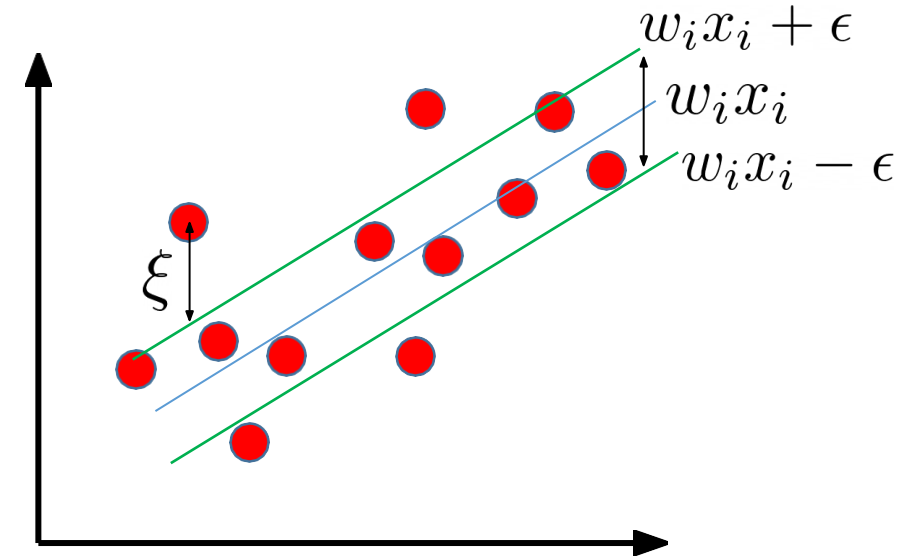
$$|y_i - w_i x_i| \leq \epsilon + |\xi_i|$$

Margin of error

Deviation from the margin (slack)

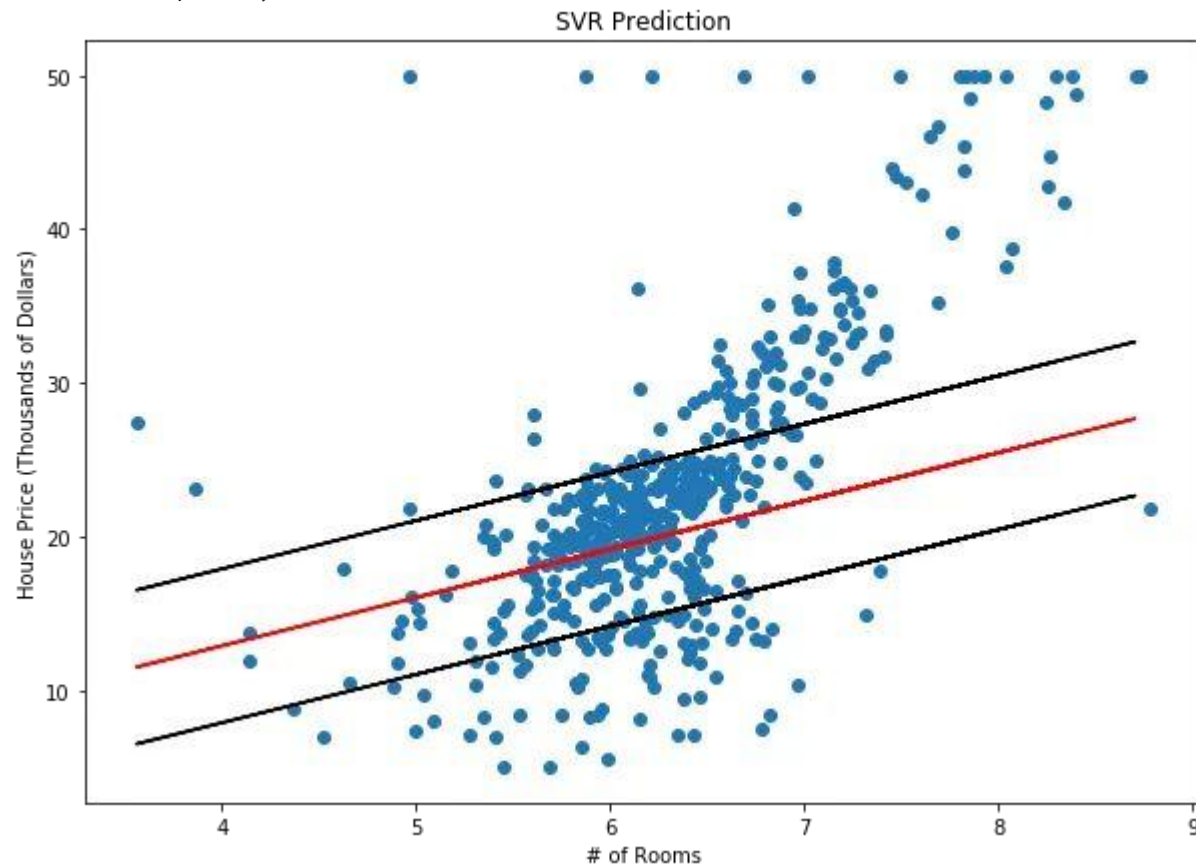
Loss function for the SVR

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n |\xi_i|$$



# Example: House price in Boston

SVR,  $\epsilon=5$ ,  $C=0$

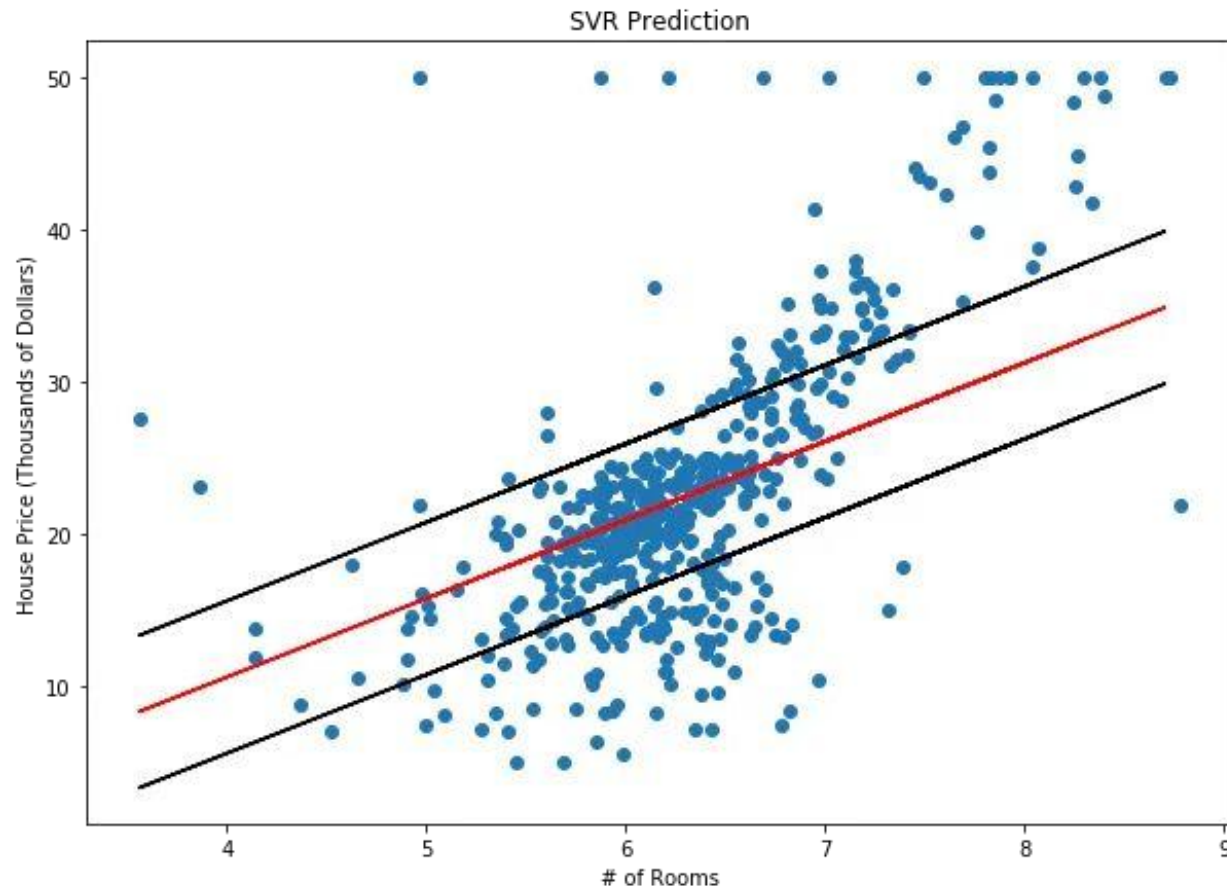


Conclusions:

- Some of the points still fall outside the margins.
- Consider the possibility of errors that are larger than  $\epsilon$ .
- Add some slack (ie,  $C$ )
- Notice that in [sklearn](#), the strength of the regularization is the inverse of  $C$

# Example: House price in Boston

SVR,  $\epsilon=5$ ,  $C=1.0$



Conclusions:

- As  $C$  increases, our tolerance for points outside of  $\epsilon$  also increases.
- As  $C$  approaches 0, the tolerance approaches 0 and the equation collapses into the simplified (although sometimes infeasible) one.

# Example: House price in Boston

- We can use grid search over  $C$  to find the ideal amount of slack (more points within margin).
- Since our original objective of this model was to maximize the prediction within our margin of error (\$5,000), we want to find the value of  $C$  that maximizes % within *Epsilon*. Thus,  $C=6.13$ .

