Introduction to Machine Learning & SVM

Antonio Fonseca

Agenda

1) Logistics
- 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

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

Class 2: Optimizers and the Perceptron (pt. 1)

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

Our roadmap

Class 3: Perceptron (pt. 2) and Neural Networks (pt. 1)

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

Class 4: Neural Networks (pt. 2)

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

Our roadmap

Class 5: Convolutional Neural Networks

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

Class 6: BYOP (Bring Your Own Paper)

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

What is machine learning?

https://tinyurl.com/GeoComp2023

What is machine learning?

Machine learning is the process of identifying patterns in data.

Two kinds of machine learning

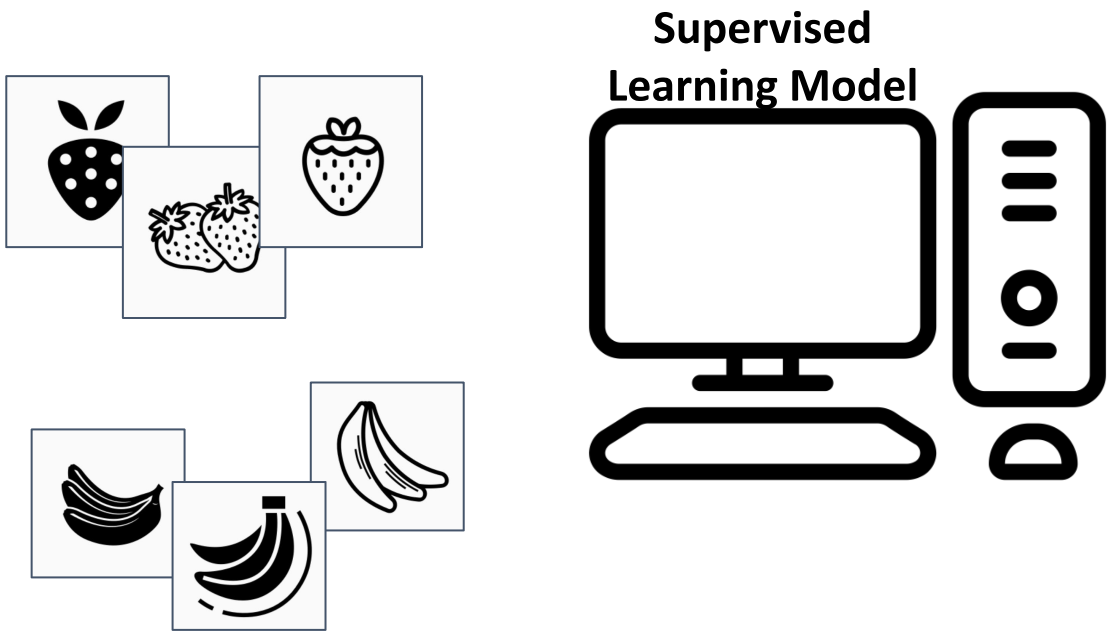
Supervised learning

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

Two kinds of machine learning

Supervised learning

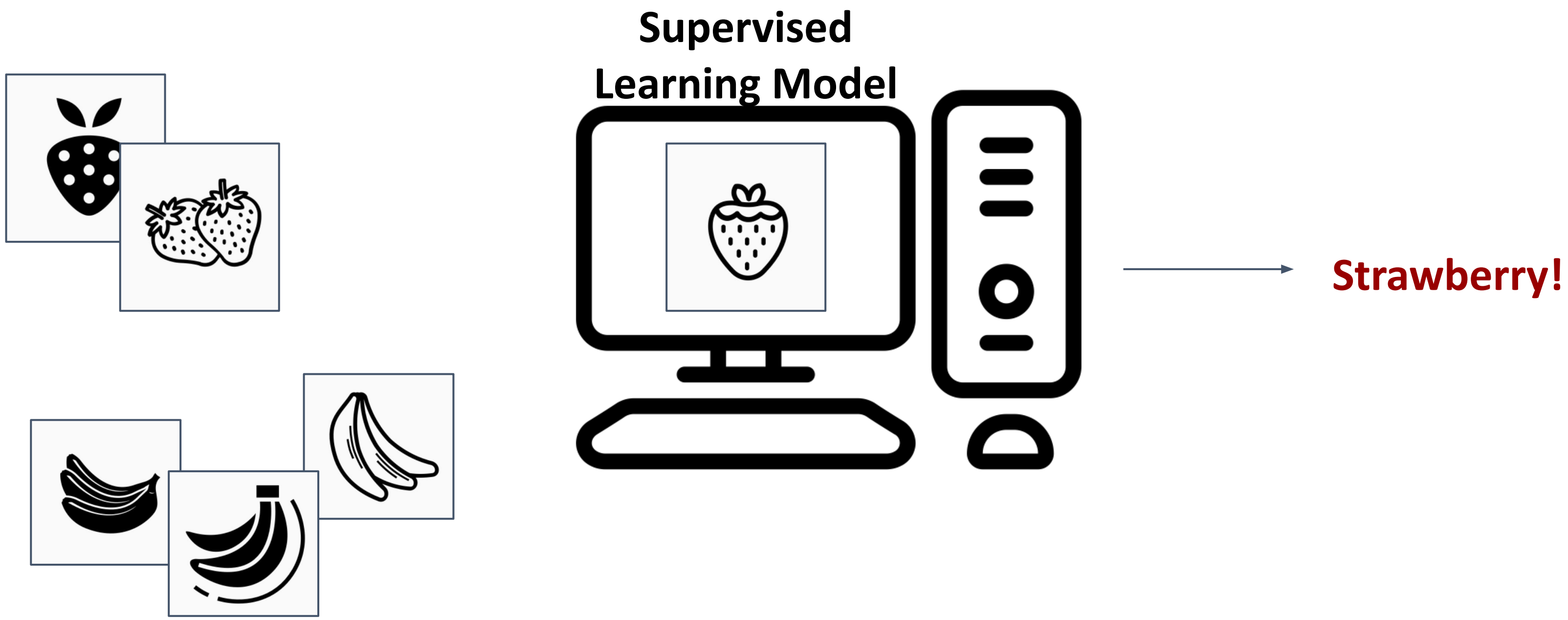
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

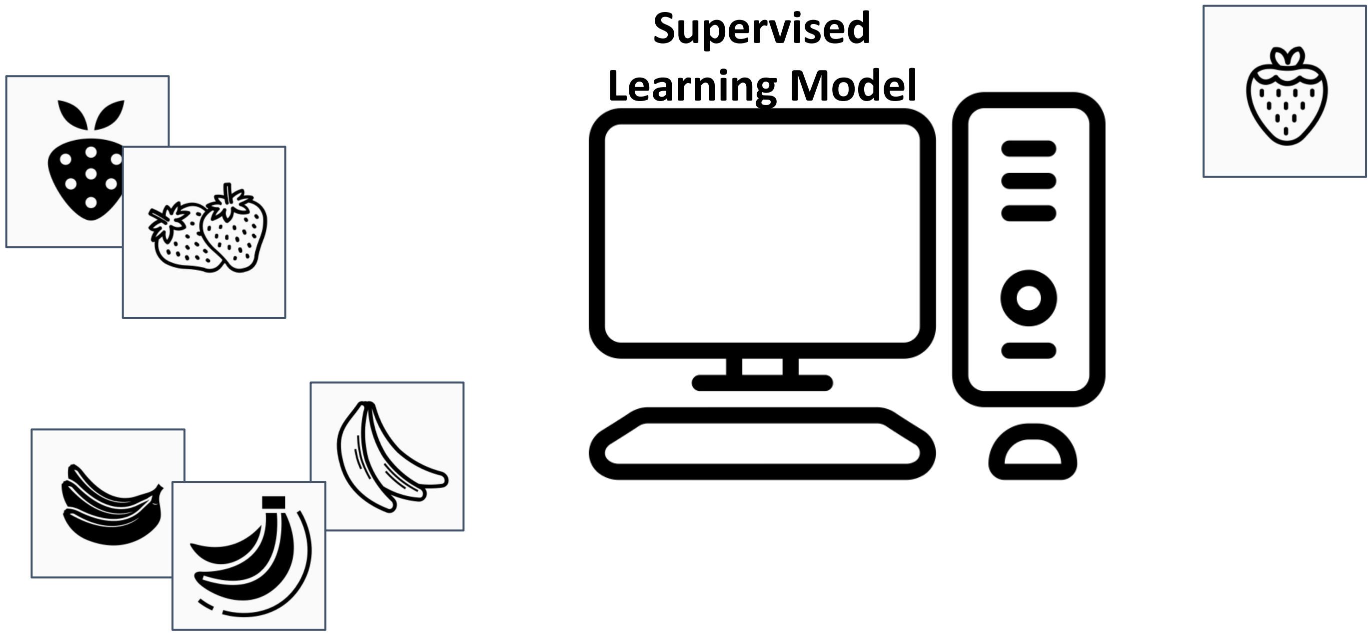
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

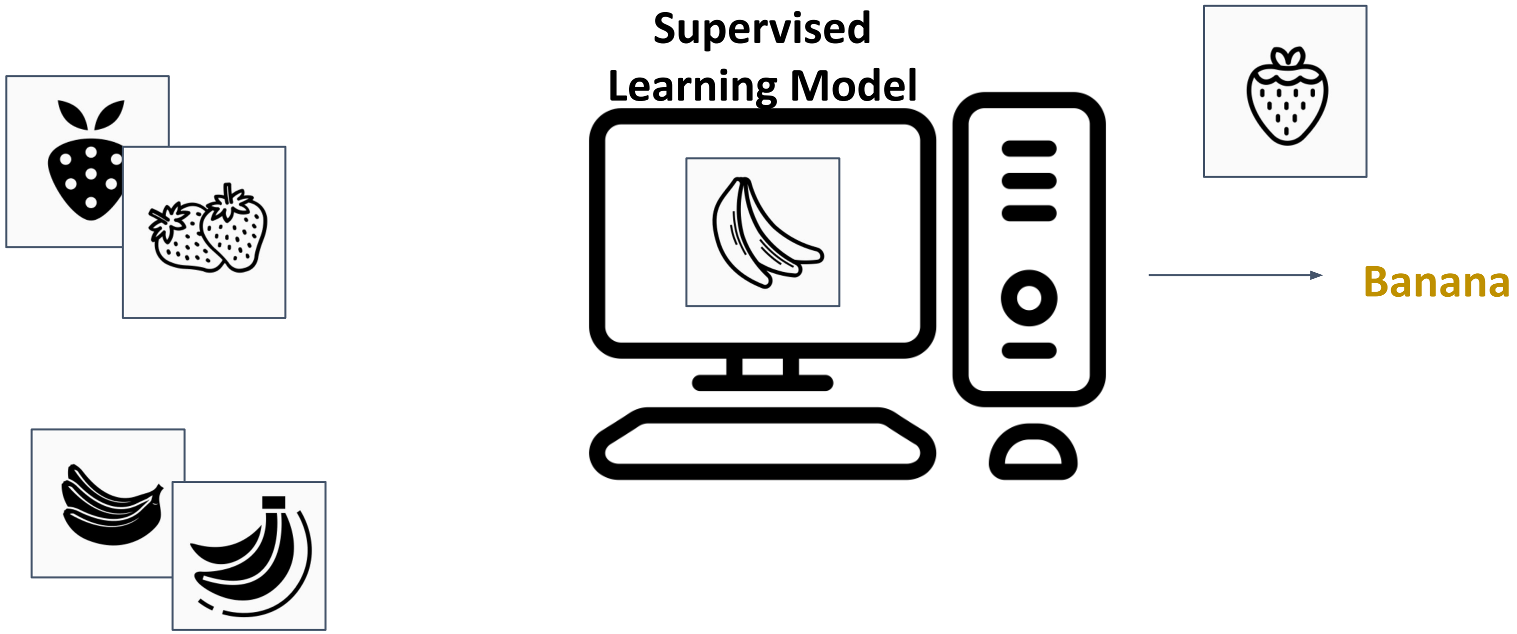
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

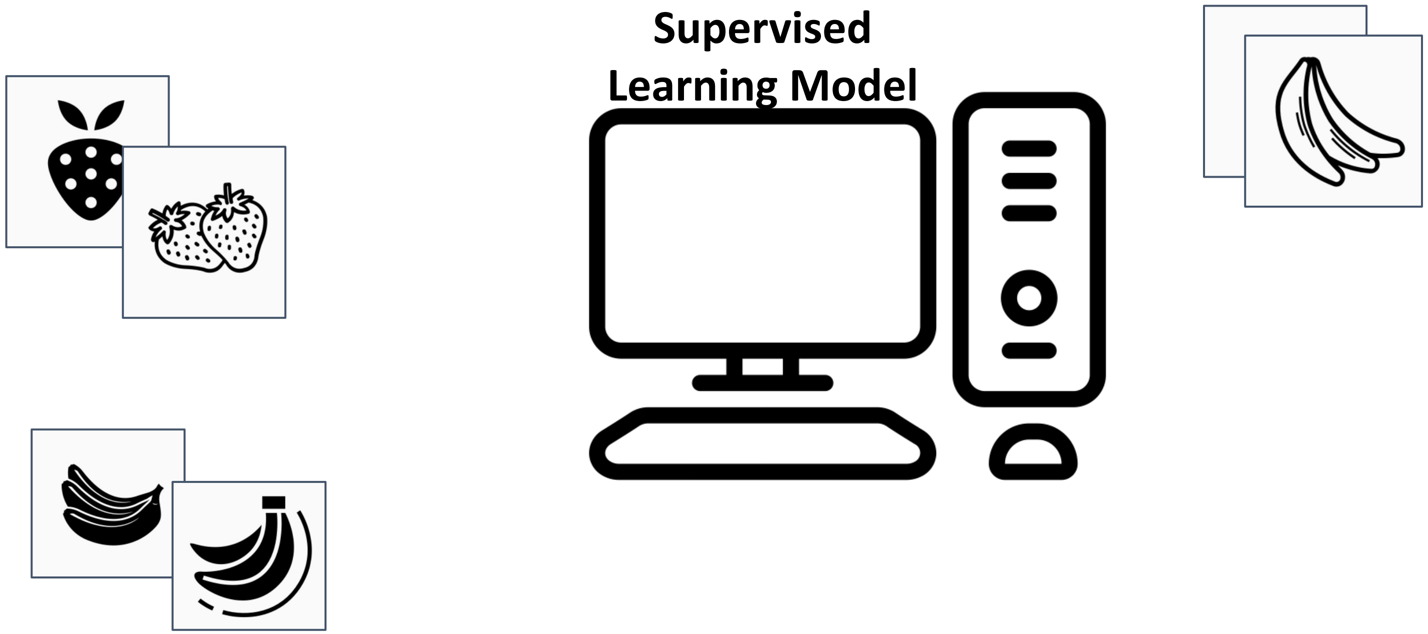
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

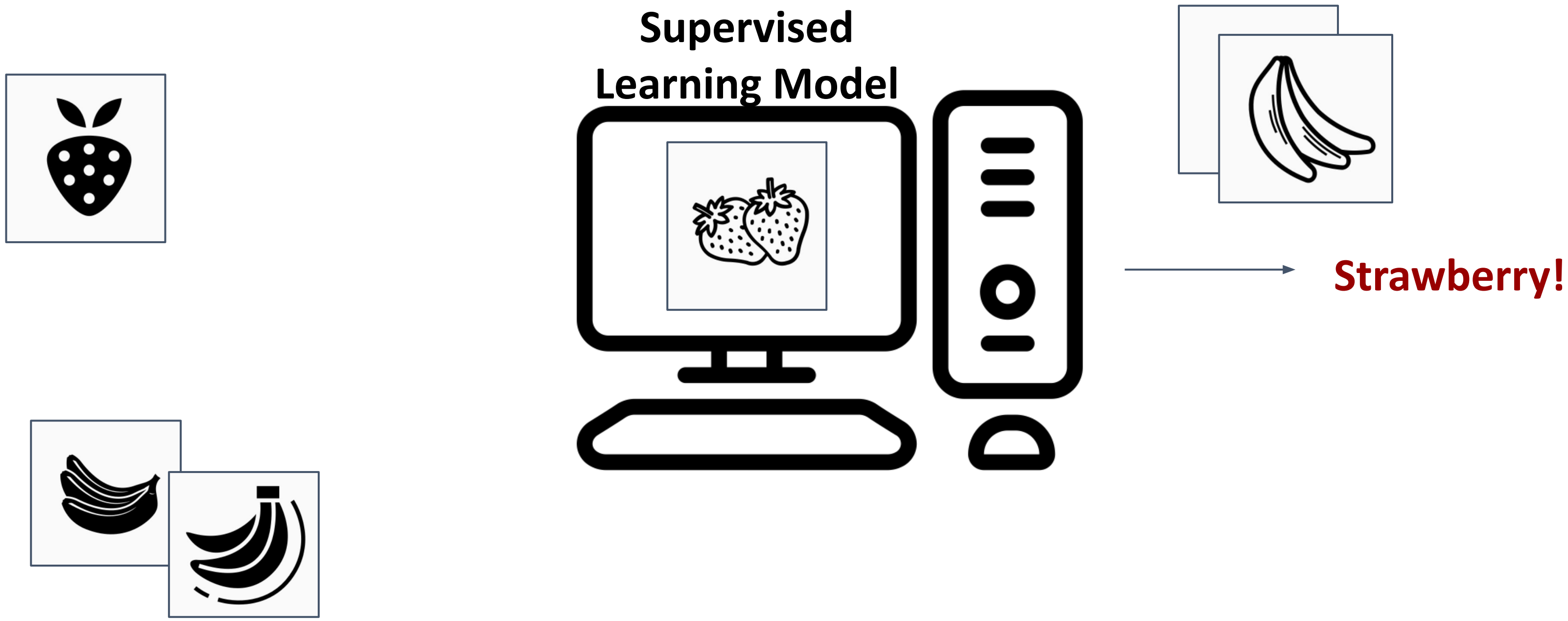
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

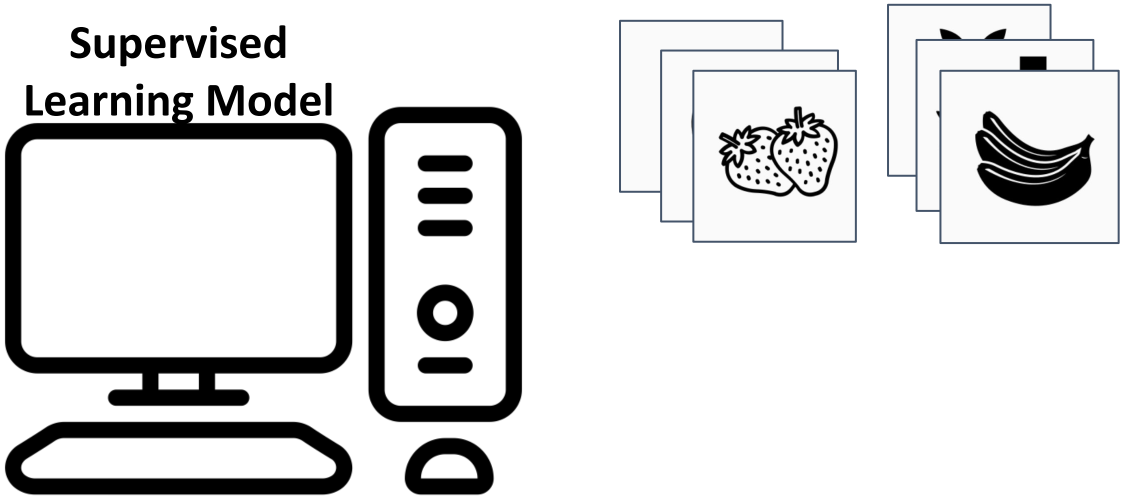
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

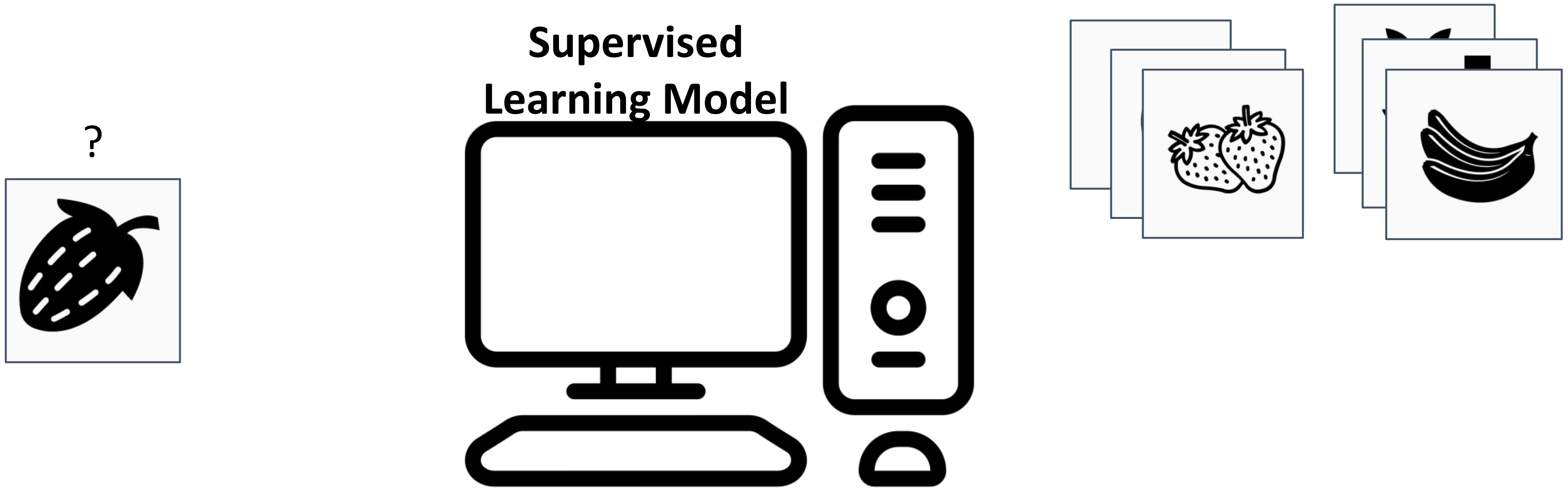
- Have a bunch of labelled data, want to label new data



Two kinds of machine learning

Supervised learning

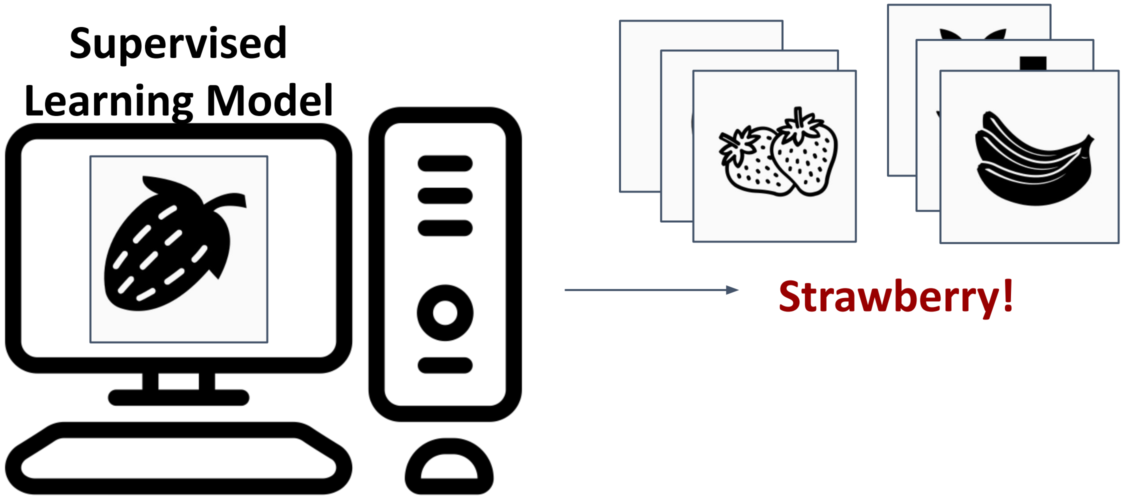
- Have a bunch of labelled data,
  want to label new data



Two kinds of machine learning

Supervised learning

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



Two kinds of machine learning

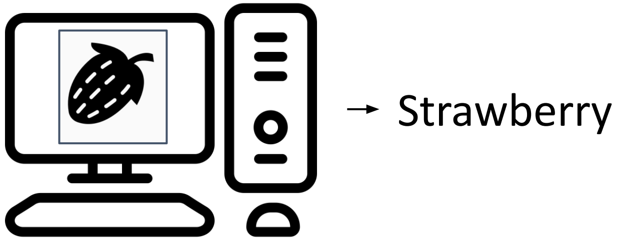
Supervised learning

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

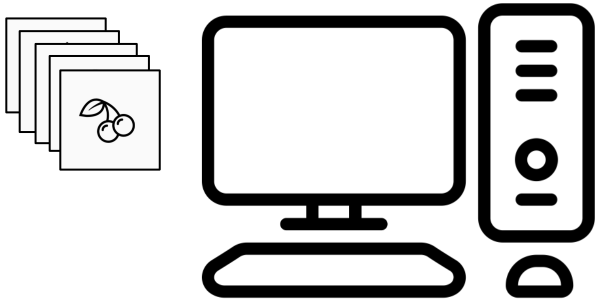
Unsupervised learning

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

Supervised Learning Model



Unsupervised Learning Model



Two kinds of machine learning

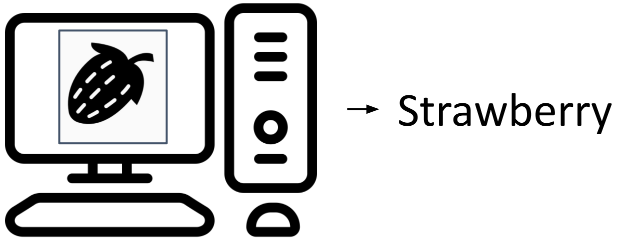
Supervised learning

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

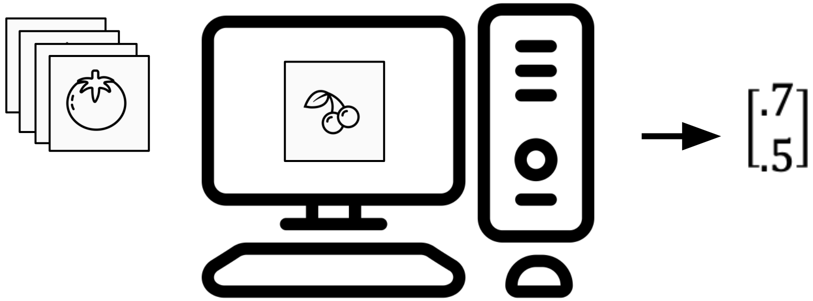
Unsupervised learning

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

Supervised Learning Model



Unsupervised Learning Model



Two kinds of machine learning

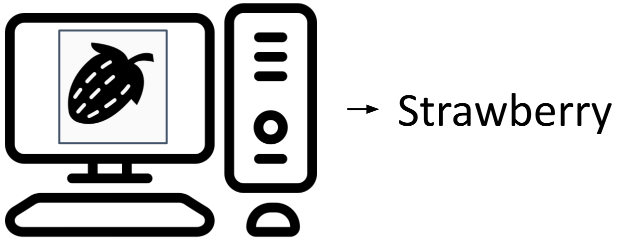
Supervised learning

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

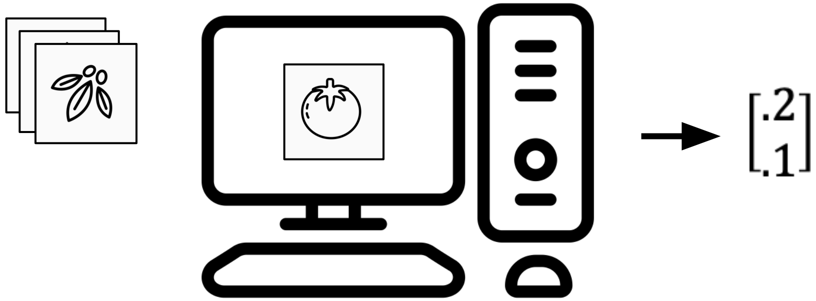
Unsupervised learning

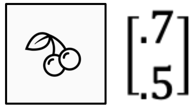
- Have a bunch of unlabeled data, want to organize it

Supervised Learning Model



Unsupervised Learning Model





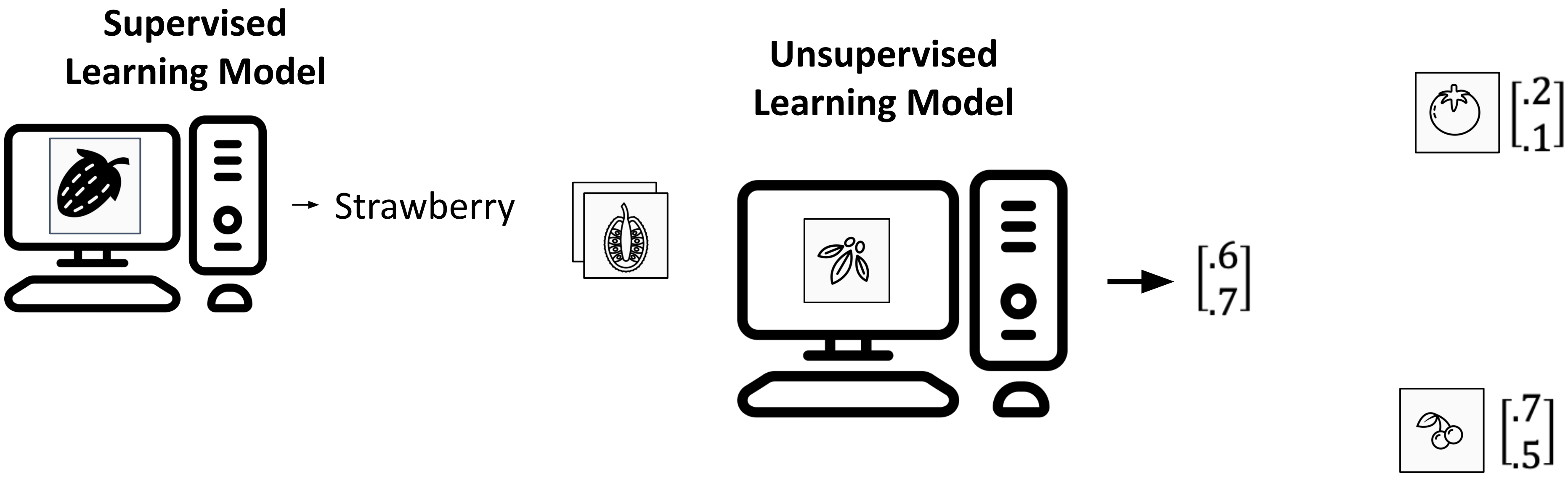
Two kinds of machine learning

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

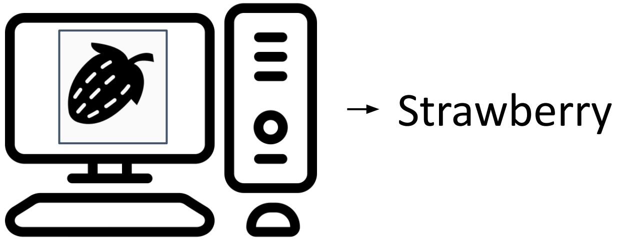
Supervised learning

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

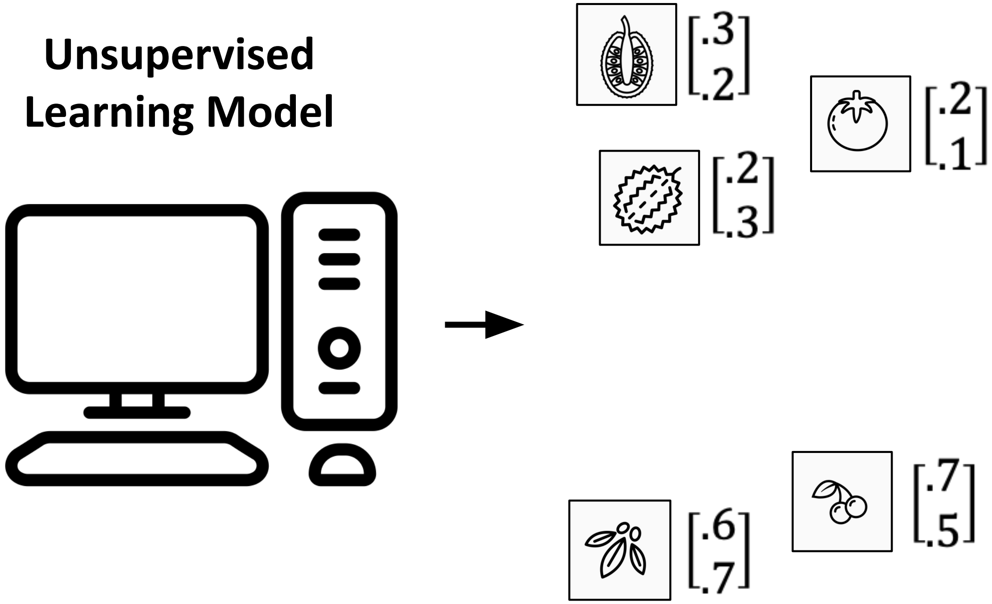
Unsupervised learning

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

Supervised Learning Model



Unsupervised Learning Model



Two kinds of machine learning

Supervised learning

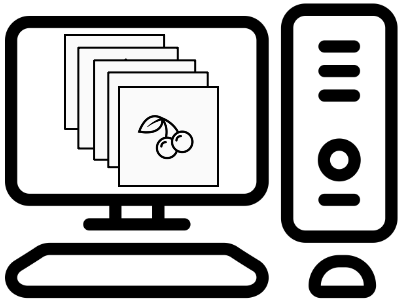
- Have a bunch of labelled data, want to label new data
- Learn a function $f(X) \rightarrow Y$ where all values of Y are known for some samples of X

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)

Supervised Learning Model

Unsupervised Learning Model

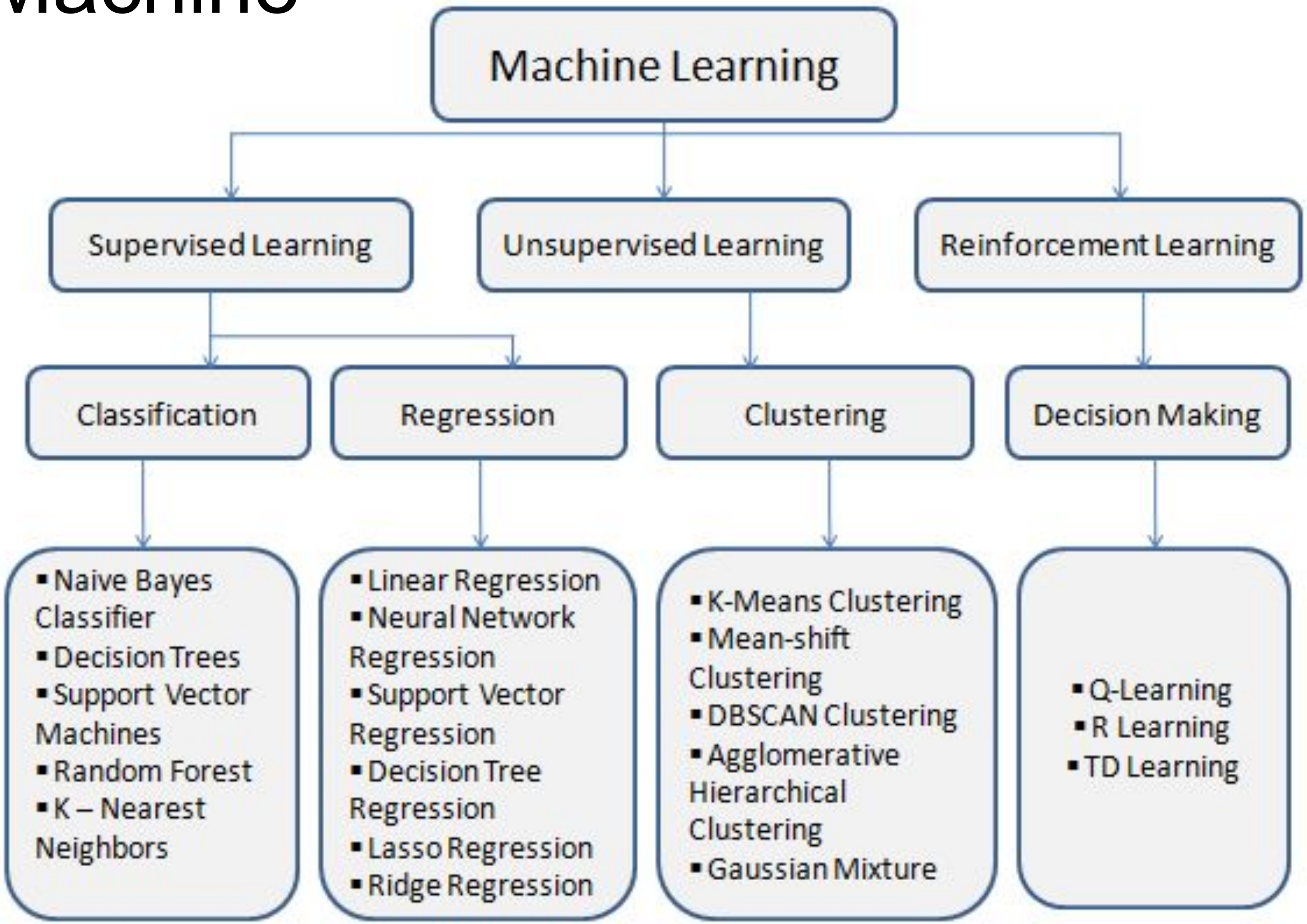


Learning algorithms

"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)	**Performance (P)**	**Experience (E)**
Transcription		
Machine Translation	Accuracy rate	Supervised Learning
Classification		
Anomaly detection		Unsupervised Learning
Synthesis and sampling		
⋮	Adjusted R^2	
Regression	RMSE/MSE/MAE	Reinforcement Learning

Types of Machine Learning



Putting these frameworks in perspective

- **"Pure" Reinforcement Learning** (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning** (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning** (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

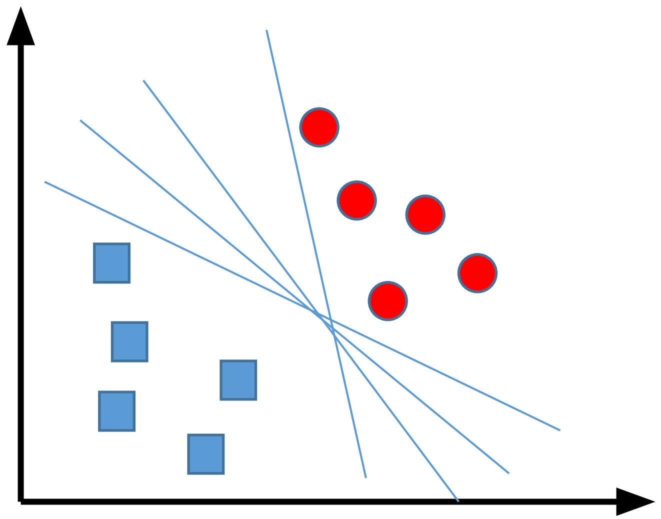
Original LeCun cake analogy slide presented at NIPS 2016

Time for a little quiz!

https://tinyurl.com/GeoComp2023

Decision Boundaries

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

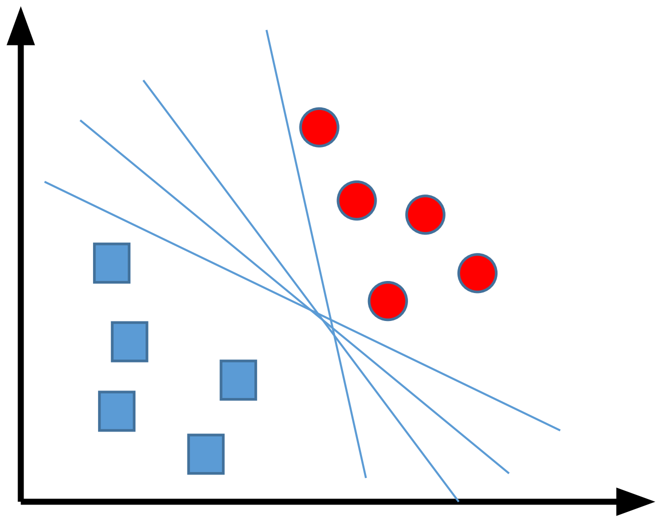


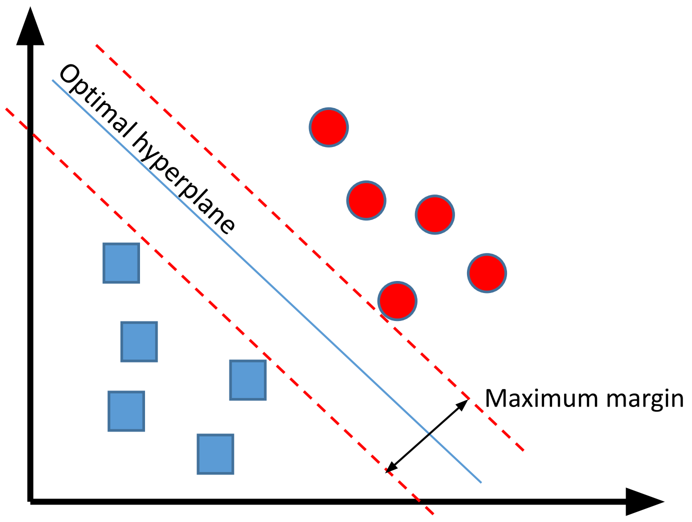
What is the correct decision boundary for this problem?

https://tinyurl.com/GeoComp2023

Support Vector Machine

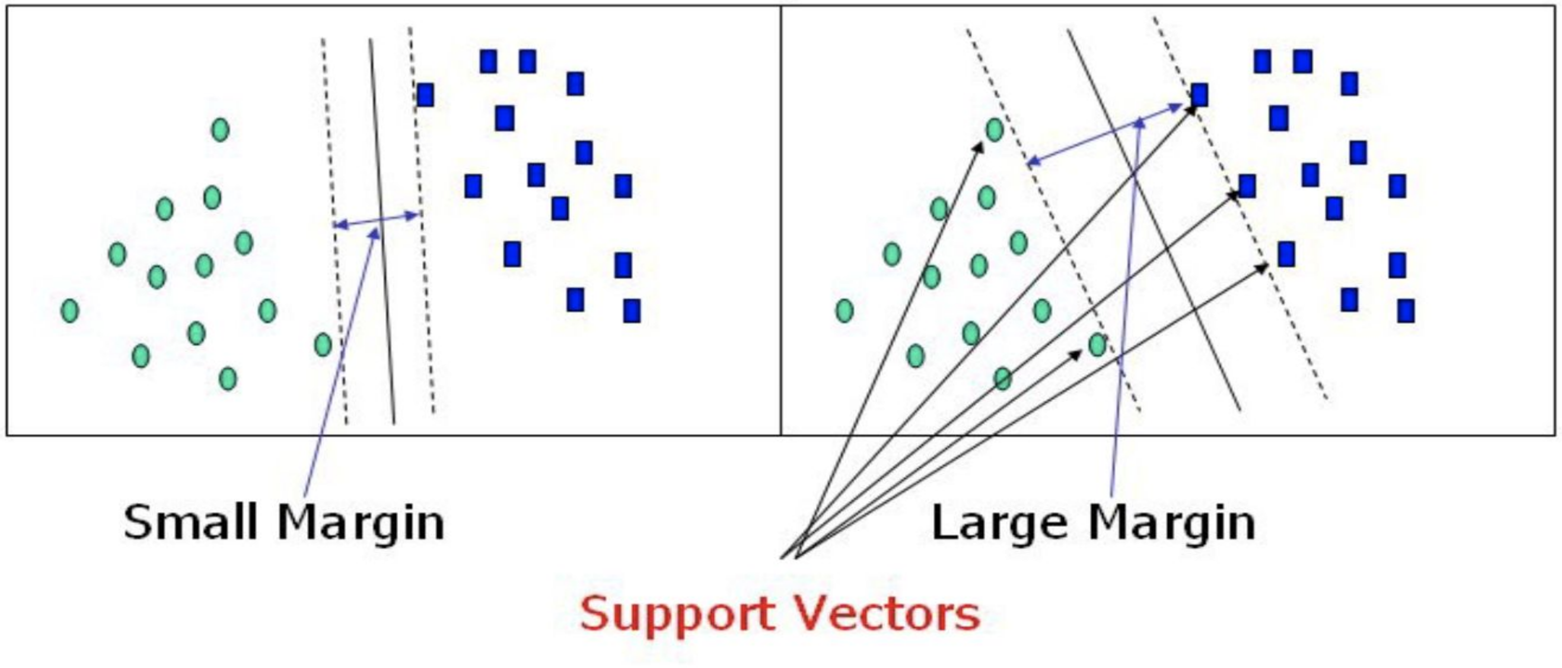
Find the optimal hyperplane in an N-dimensional space that distinctly classifies the data points.





Support Vector Machine

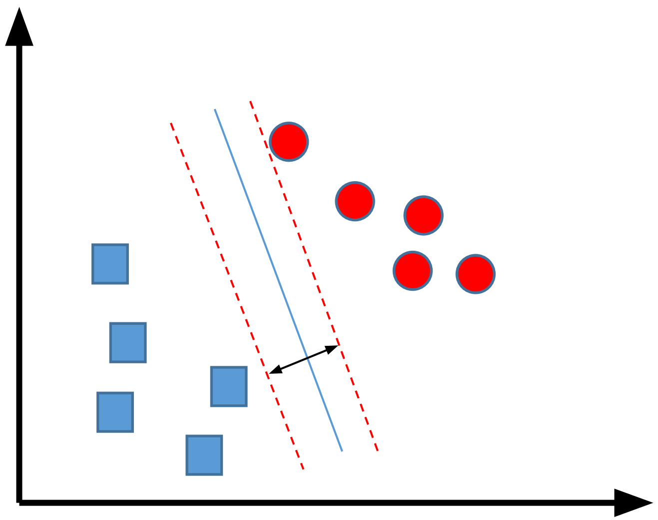
Maximize the margin of the classifier



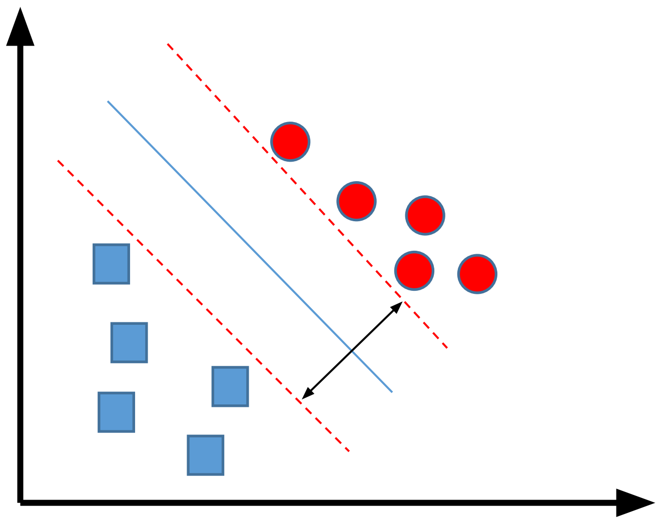
Support Vectors

Support Vector Machine

Maximize the margin of the classifier



Small margin



Large margin

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

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

Gradients

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

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

Updating the weights:

No misclassification

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

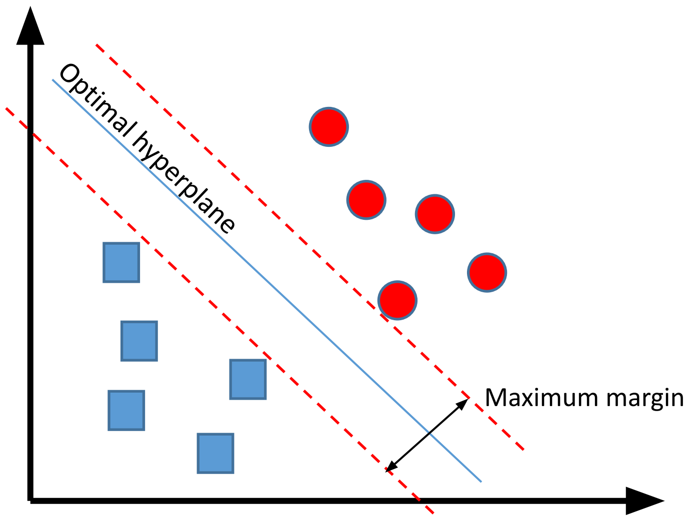
Misclassification

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

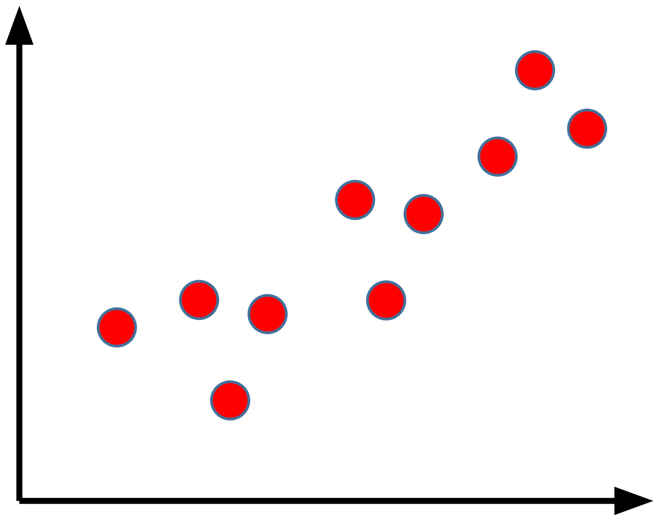
Support Vector Machine for Regression

How do I turn the SVM into a SVR?

SVM



SVR(?)



SVR Optimization

Loss function for the SVR

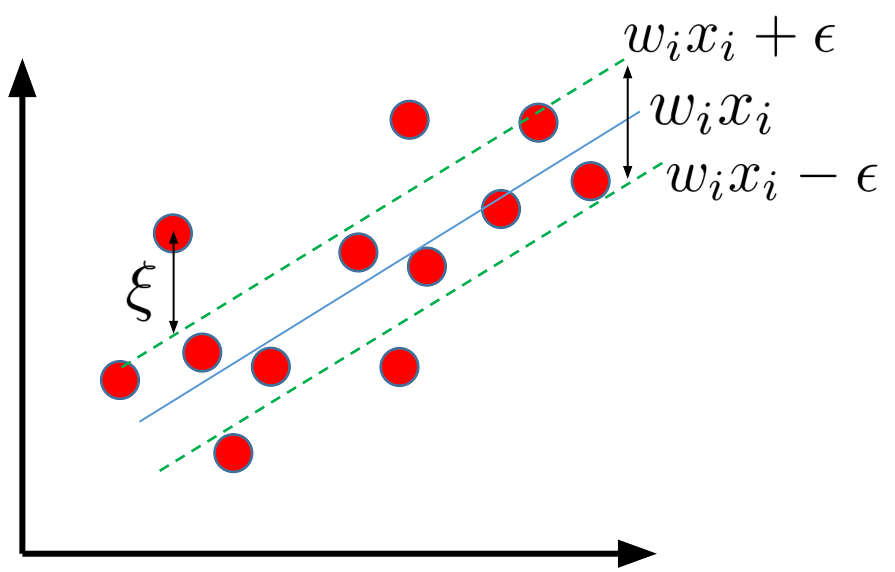
$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} |\xi_i|$$

Constraints

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

ϵ → Margin of error

$|\xi_i|$ → Deviation from the margin



Support Vector Machine for Regression

- The best fit line is the hyperplane that has the maximum number of points.
- Limitations
  - The fit time complexity of SVR is more than quadratic with the number of samples
  - SVR scales poorly with number of samples (e.g., >10k samples). For large datasets, **Linear SVR** or **SGD Regressor**
  - Underperforms in cases where the number of features for each data point exceeds the number of training data samples
  - Underperforms when the data set has more noise, i.e. target classes are overlapping.