Machine Learning and Material Science
3. Support Vector Machine

Steffen Brinckmann, Claas Hüter
IEK-2, Forschungszentrum Jülich GmbH

Support Vector Machine

For classification problems

Find linearly separable hyperplane to separate classes

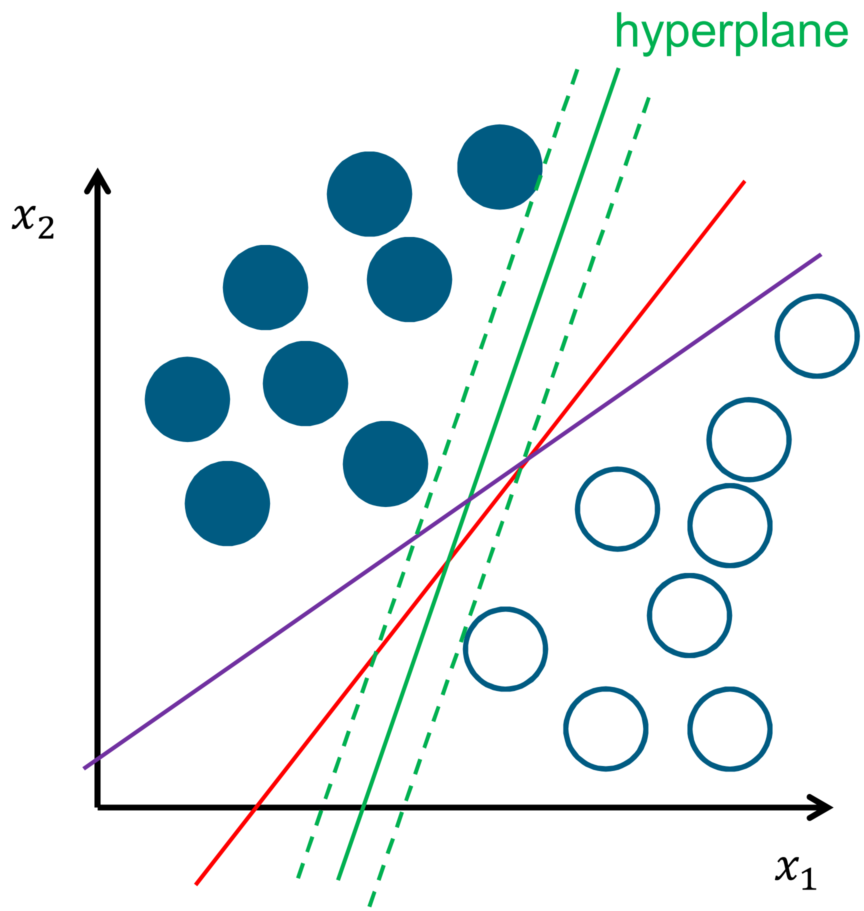
If data cannot be separated linearly

uses nonlinear mapping to transform data into higher dimension

search for separating hyperplane

Algorithm finds hyperplane using support vectors and margins

Support Vector Machine
Define hyperplane to discriminate / classify



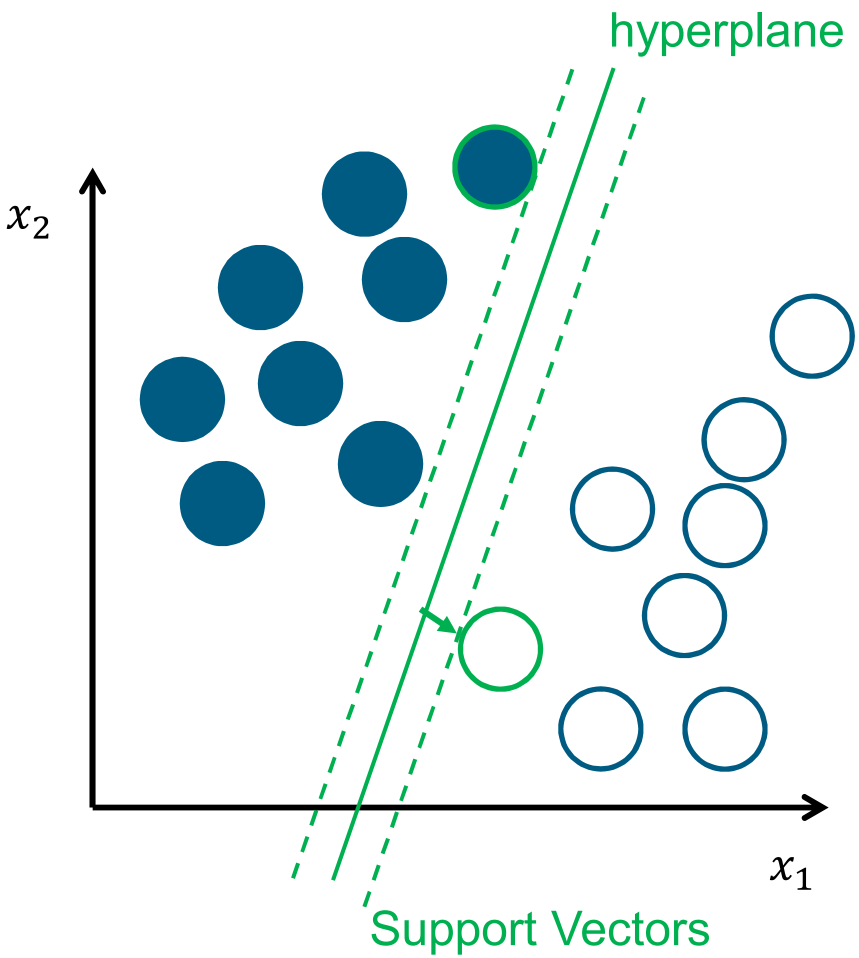
Example:
Datapoints with $X=(x_1,x_2)$ and
class (-1 or 1)

Finding best way to separate data

Best hyperplane is the one with the longest support vector or maximum margin

Support vector:
Shortest vectors between hyperplane and nearest data points of each class

Support Vector Machine
Define hyperplane to discriminate / classify



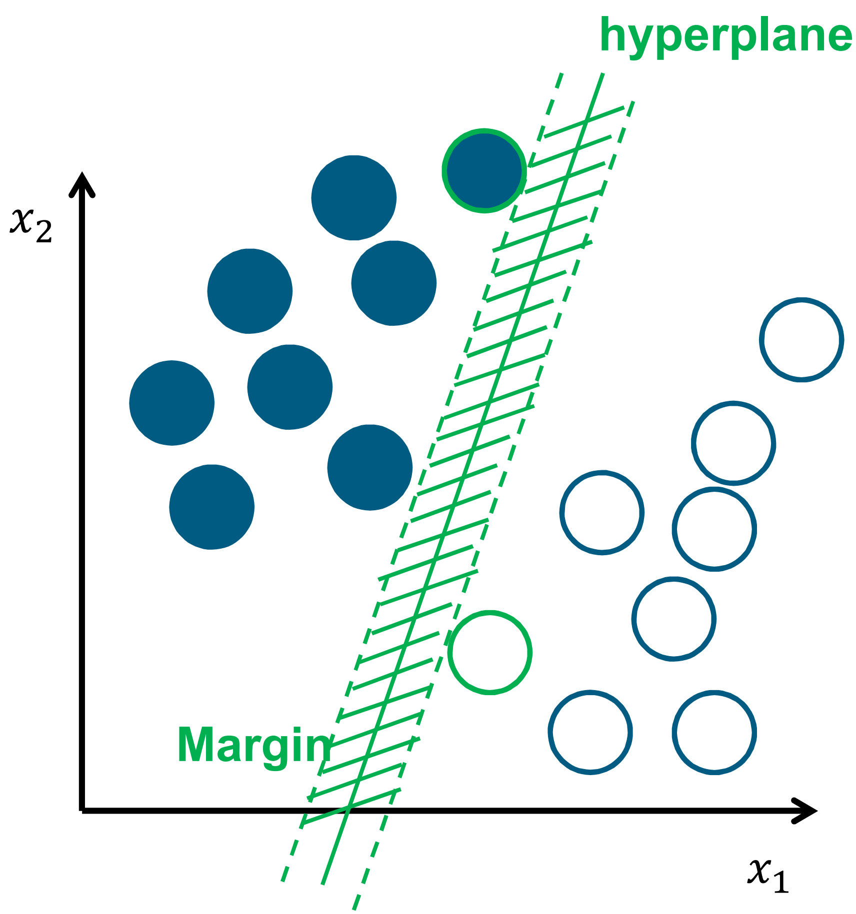
Example:
Datapoints with $X=(x_1,x_2)$ and Y the class (-1 or 1)

Finding best way to separate data

Best hyperplane is the one with the longest support vector or maximum margin

Support vector:
Shortest vectors between hyperplane and nearest data points of each class

Support Vector Machine
Define hyperplane to discriminate / classify



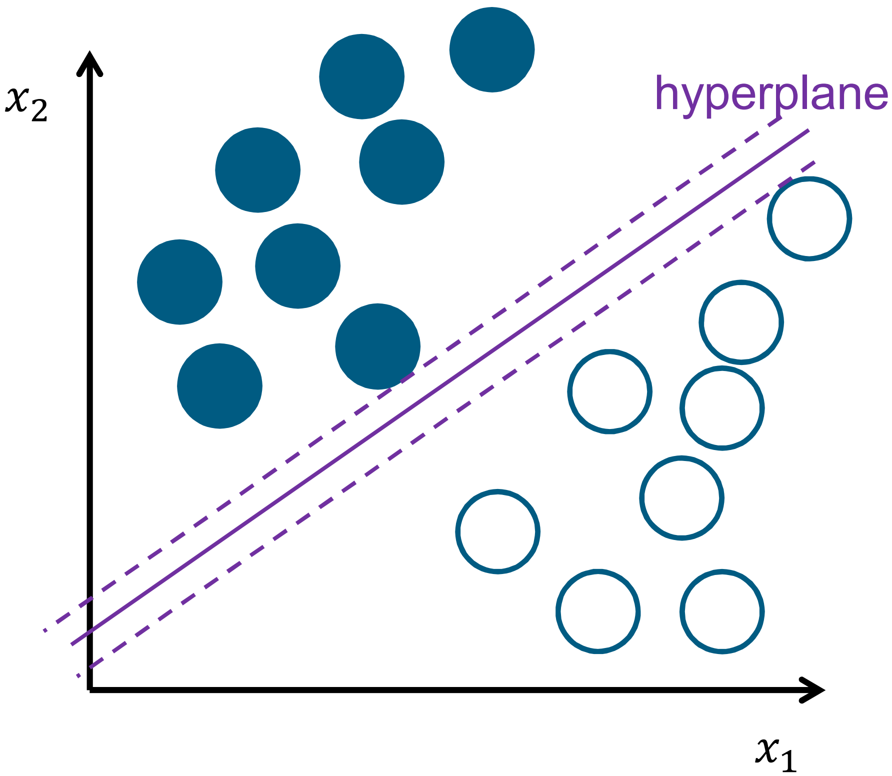
Example:
Datapoints with X=(x_1,x_2) and Y the class (-1 or 1)

Finding best way to separate data

Best hyperplane is the one with the longest support vector or maximum margin

Support vector:
Shortest vectors between hyperplane and nearest data points of each class

Support Vector Machine
Define hyperplane to discriminate / classify



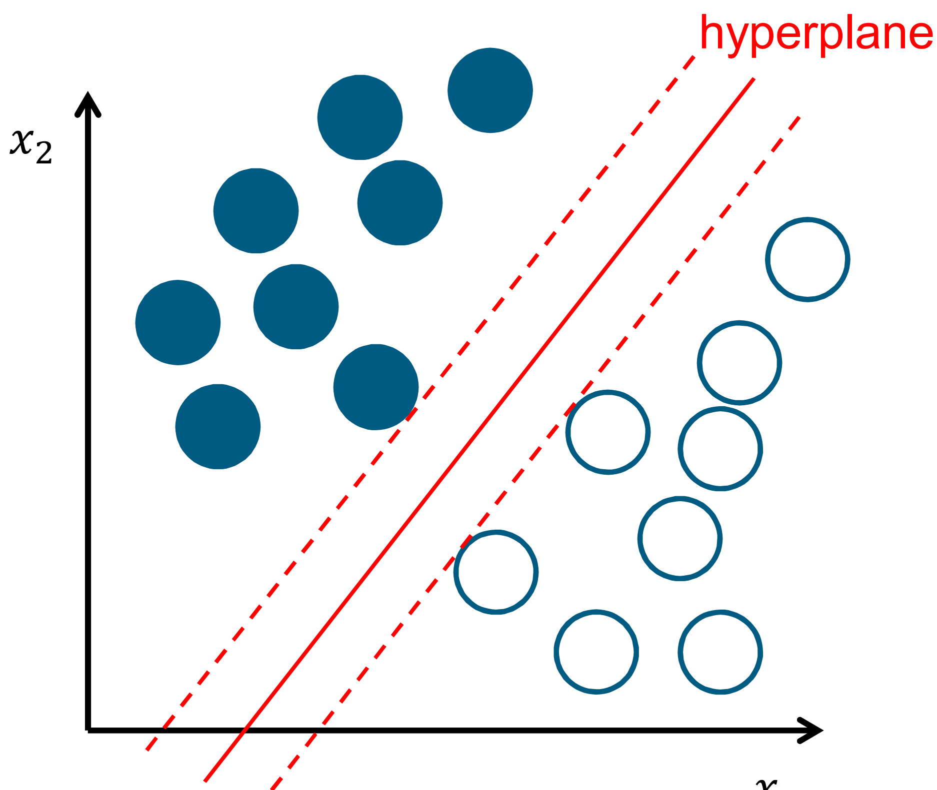
Example:
Datapoints with $X=(x_1,x_2)$ and Y the class (-1 or 1)

Finding best way to separate data

Best hyperplane is the one with the longest support vector or maximum margin

Support vector:
Shortest vectors between hyperplane and nearest data points of each class

Support Vector Machine
Define hyperplane to discriminate / classify



Example:
Datapoints with $X=(x_1,x_2)$ and Y the class (-1 or 1)

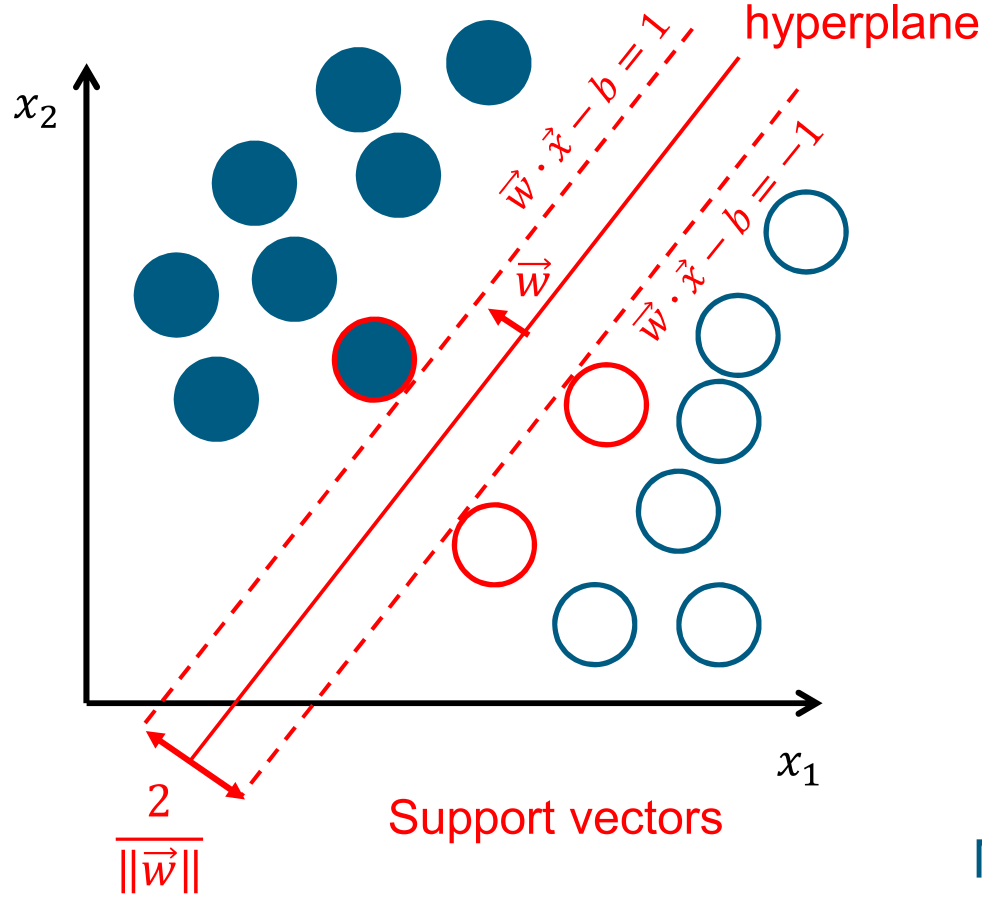
Finding best way to separate data

Best hyperplane is the one with the longest support vector or maximum margin

Support vector:
Shortest vectors between hyperplane and nearest data points of each class

Support Vector Machine

JÜLICH FORSCHUNGSZENTRUM



Hyperplane: $\vec{w} \cdot \vec{x} - b = 0$

$\vec{w}$: normal vector (not unit length)

maximizing margin

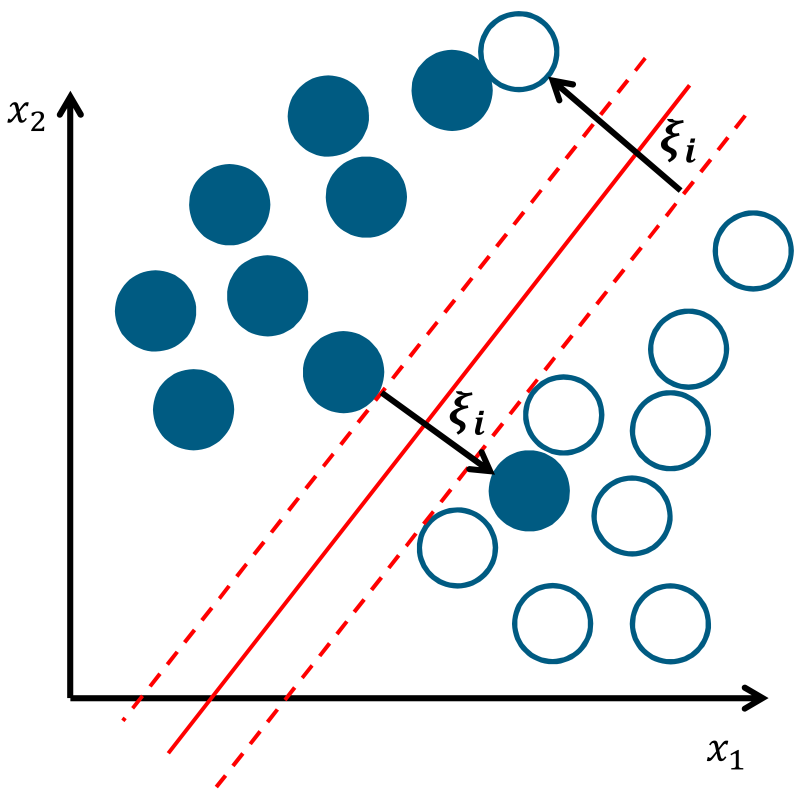
→ minimizing $||\vec{w}||$

Prevent data inside margin

$$\begin{cases} \vec{w} \cdot \vec{x}_i - b \geq 1, & if\ y_i = 1 \\ \vec{w} \cdot \vec{x}_i - b \leq -1, & if\ y_i = -1 \end{cases}$$

Minimizing $||\vec{w}||$ with $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$

Allow for misclassification / errors in data



ξ_i: slack variables allow for some misclassification

Data is noisy and maximal margin is bad solution

Hyperplane which **almost** separates => **soft margin**.

Small number of samples cross margin => violate margin

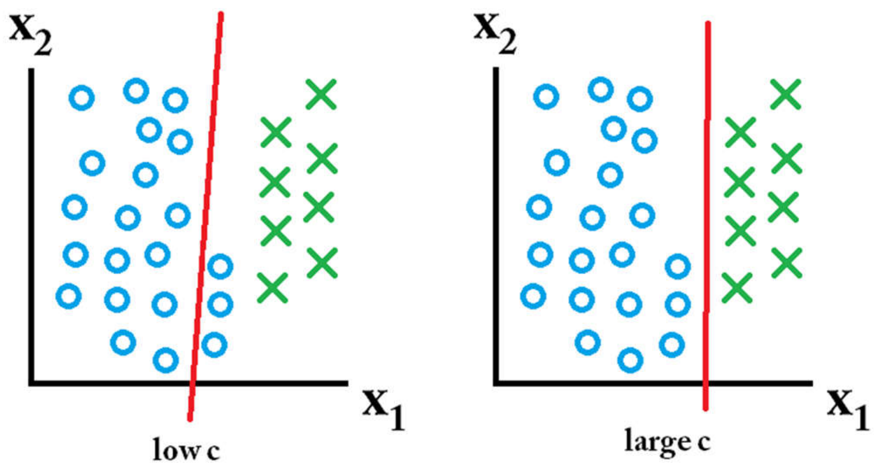
Minimizing $[||\vec{w}|| + C \sum_i \xi_i]$
subject to $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1 - \xi_i$

$\boldsymbol{C}$ is regularization parameter to avoid misclassification

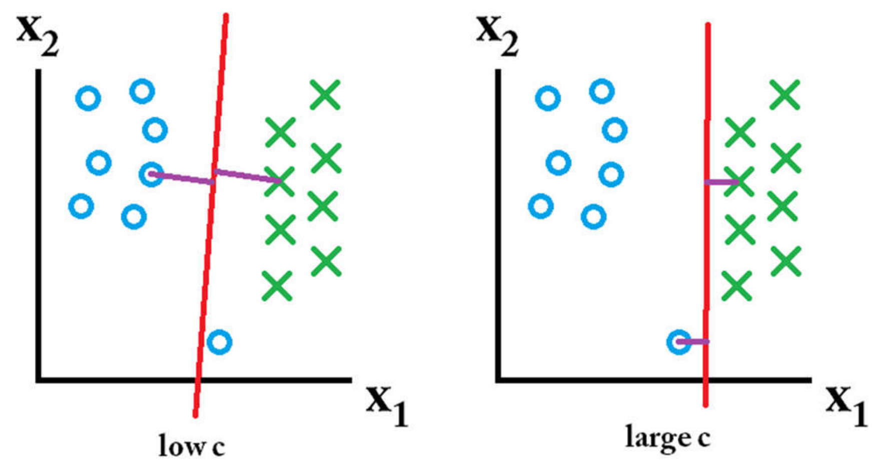
(smaller $\boldsymbol{C}$ = softer, less penalty for error)

User has to be smart

Support Vector Machine



Large C is better!



Low C is better!

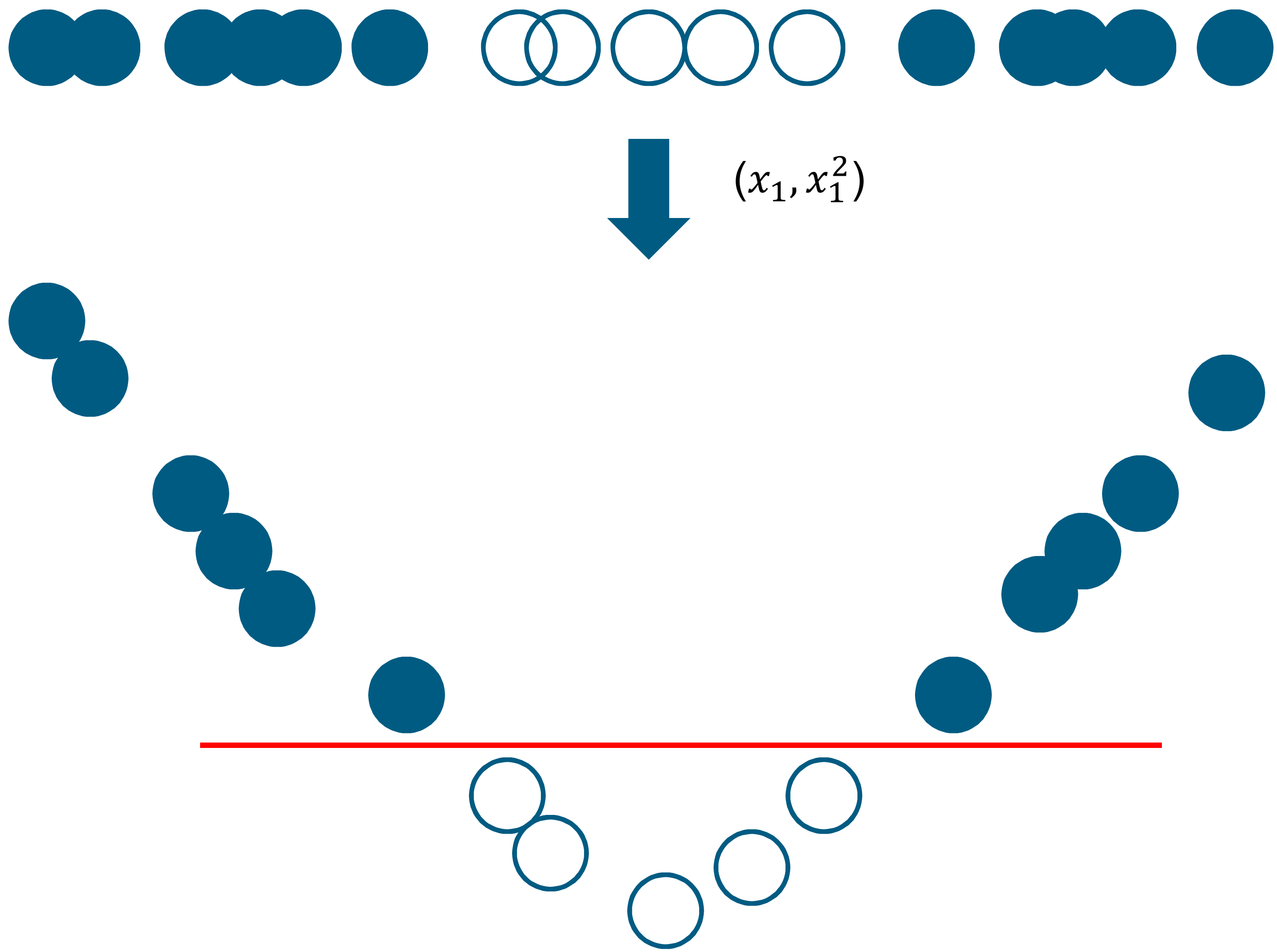
Pictures source: https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel

Support Vector Machine

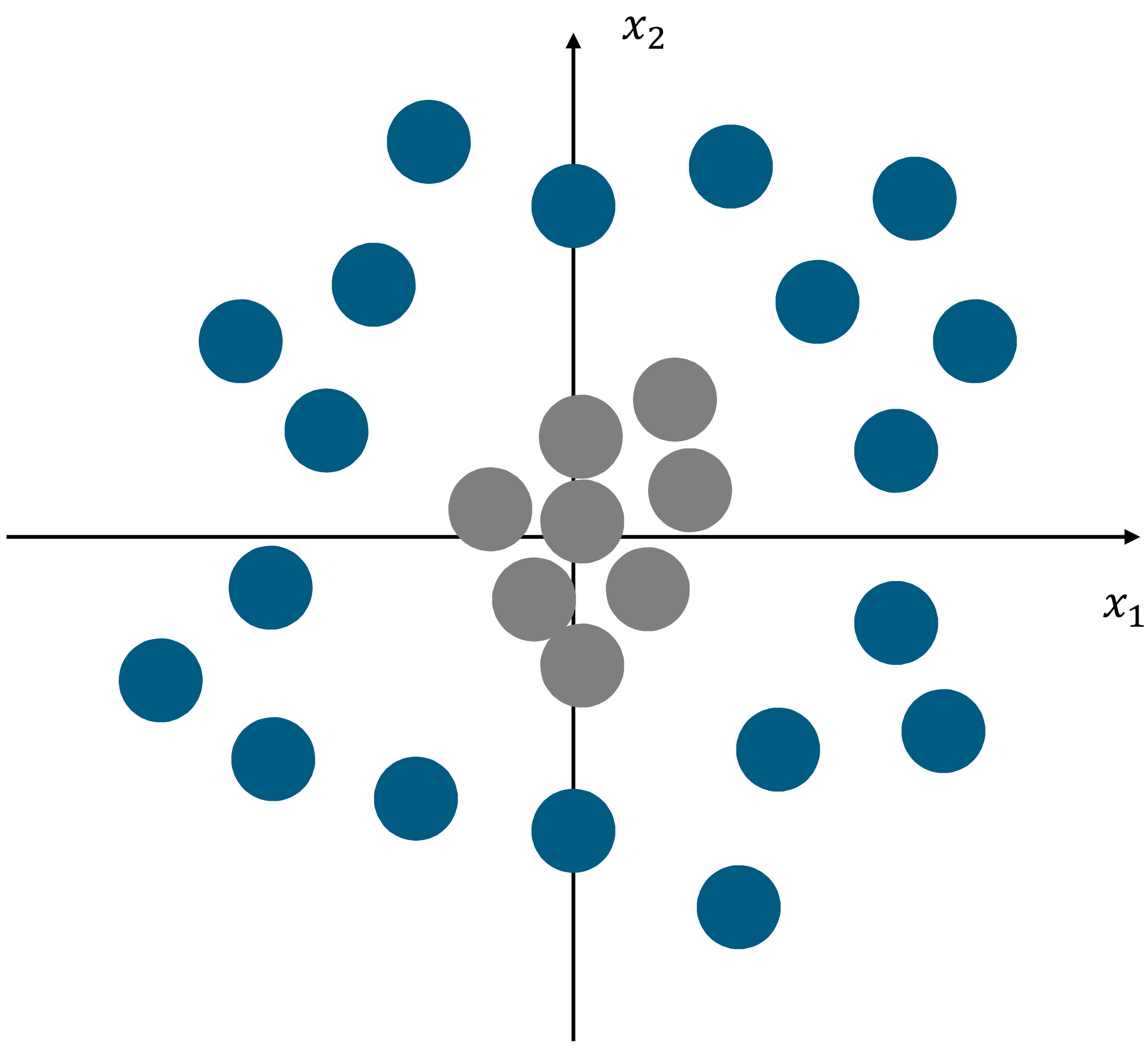
If data is linearly separable
SVM finds best separator

How about not linearly
separable data?

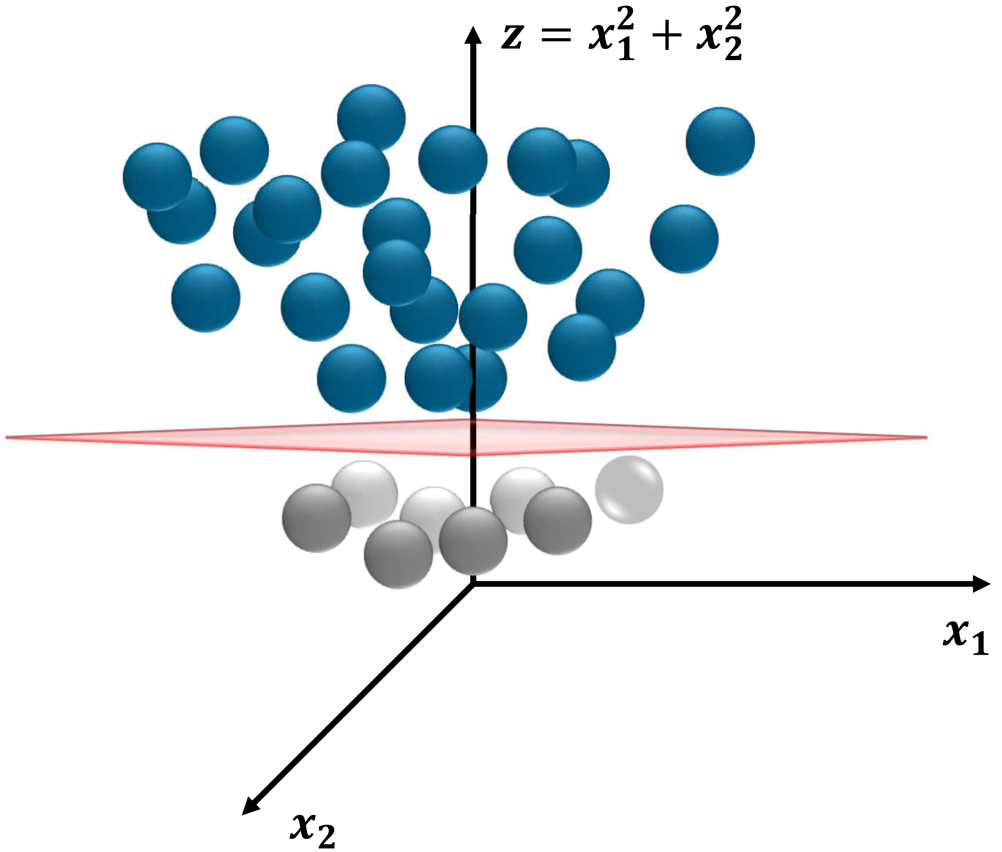
Classification problem: not linearly separable



Classification problem: not linearly separable



Classification problem: not linearly separable



Low dimension
Not linearly separable

Kernel trick →

High dimensional space
Linearly separable

Support Vector Machine

If there are **N** (N>2) classes:

'one-against-one' approach: N(N-1)/2 classifiers will be constructed and each one is trained on data from two classes. To predict, each classification gives one vote to the winning class and the point is labeled with the class having most votes.

'one-against-all' approach: k SVM models will be constructed, where k is the number of classes. The m^{th} SVM is trained with all of the examples, in the m^{th} class with positive label, and all other examples with negative labels. To predict, choose the class which has the largest value of $(\vec{w}^T\vec{x} + b)$.

Support Vector Machines

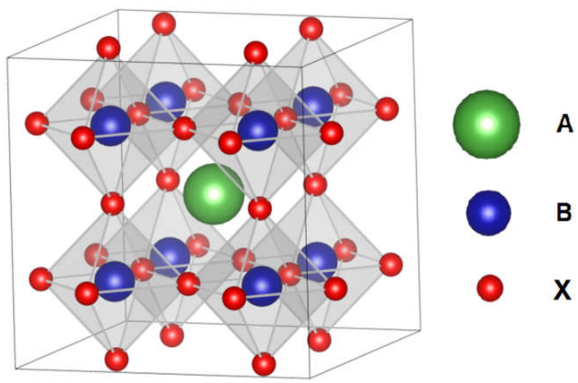
are based on

- Dot product of separating hyperplanes
- Hard margin and soft margin classification
- Kernel trick to higher dimension

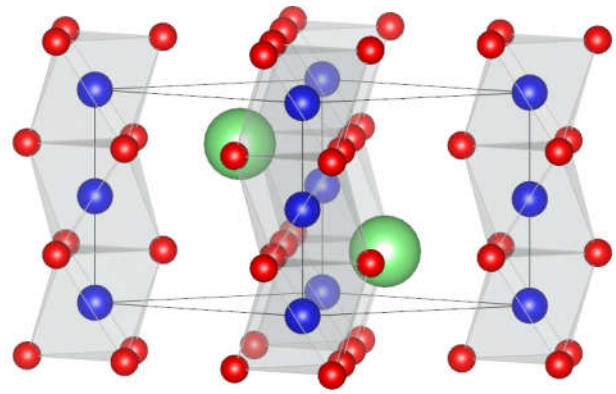
Pros & Cons:

- Kernel Trick is efficient in separating classes if problem is not too complex
- Kernel Trick possibly increases error
- require larger data sets for complex kernels

Example: Perovskite or $BaNiO_3$-type structure?



ideal cubic perovskite with corner-shared BX_6 octahedral



$BaNiO_3$ structure with face-shared BX_6 octahedral

Classification:

- $ABCl_3$ with ideal cubic perovskite structure
- $BaNiO_3$ structure

[11]: W. C. Lu et al., 1:151-159, 2003.

Example: Perovskite or $BaNiO_3$-type structure?

Feature $\vec{x}$: $(R_A, R_B, X_A, X_B \text{ and } N_d)$

R_A and R_B: ionic radii of A and B in $ABCl_3$
X_A and X_B: electroegativity of A and B
N_d: the number of d electrons in the unfilled shell of d electrons of B ions

Two classes: perovskite structure and $BaNiO_3$-type structure

Data set: 23 samples
Training data: 22 samples
Testing data: 1 sample

The criterion for the formation of $BaNiO_3$- type structure

$$4.52R_B - 1.83R_A + 2.23X_A - 0.142X_B - 4.10N_d + 0.589 < 0$$

W. C. Lu et al., 1:151-159, 2003.

How to communicate classification?

Example: check ultrasonic reflection
to determine ‘damaged’ or ‘not damaged’

	Actual state		
Predicted state		damaged	not damaged
	damaged	True Positive TP	False Positive FP
	not damaged	False Negative FN	True Negative TN

4 individual values: which to optimize

What is a good classifier value?

Sensitivity/Recall/Hit Rate/True Positive Rate

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

Specificity/Selectivity/True Negative Rate

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP}$$

Precision/Positive Predictive Value

$$PPV = \frac{TP}{TP + FP}$$

Fall-Out/False Positive Rate

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$$

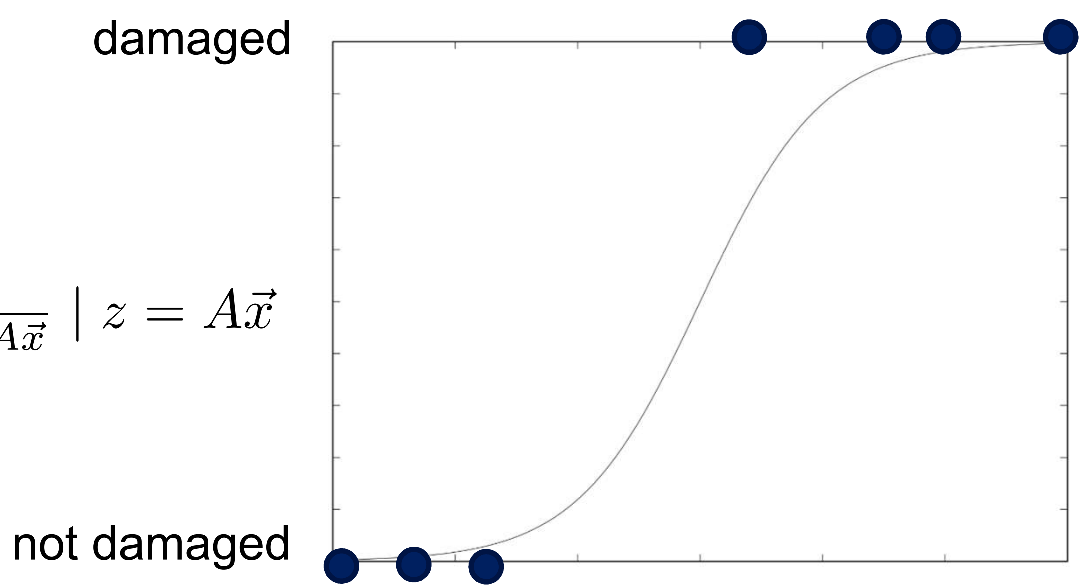
	damaged	not damaged
damaged	True Positive TP	False Positive FP
not damaged	False Negative FN	True Negative TN

Logistic Regression classification

Given set of inputs X, want to assign them to either class 0 or class 1

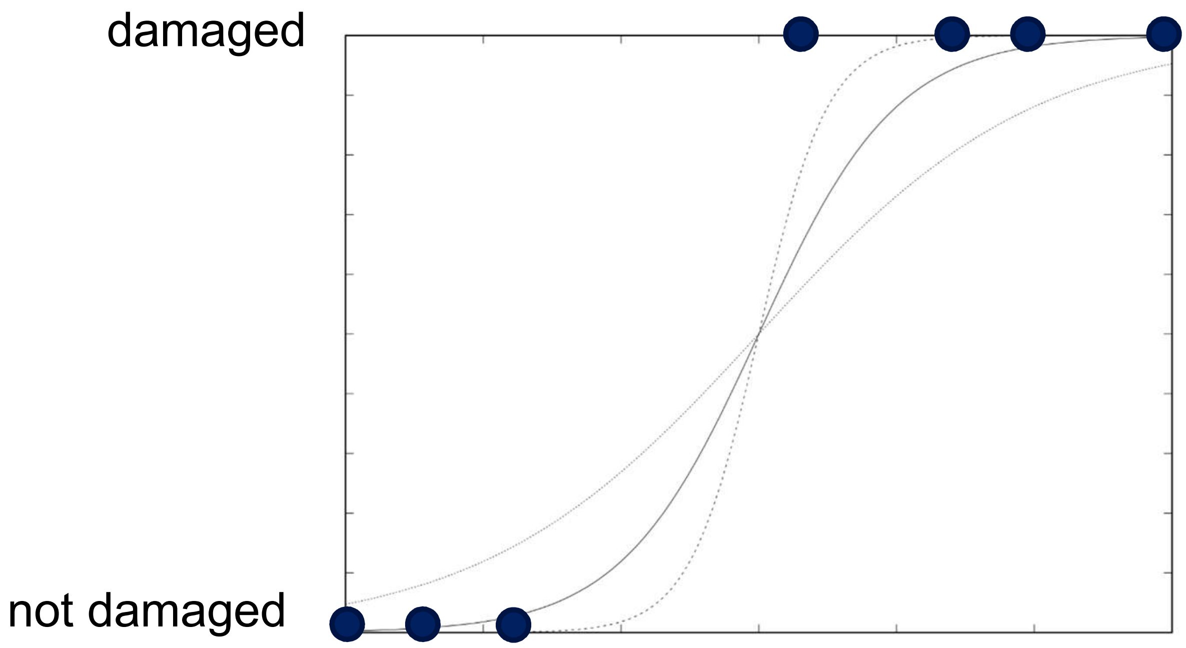
Logistic Regression simulates probability that each data point belongs either to class 0 or 1

$$sig(z) = \frac{1}{1 + e^{-A\vec{x}}} \mid z = A\vec{x}$$



Logistic Regression: Maximum Likelihood

- Calculate likelihood for each datapoint to be described by the sigmoid, take product of likelihoods for all dps
- Optimise weights such that maximum value for the product of likelihoods of all dps is reached



What makes a good (binary) classifier?

Visualisation depends on the symmetry of your dataset:

Symmetric distribution of positives and negatives:

Receiver Operating Characteristic (ROC)

-Visualizes Trade-Off between TPR and FPR

-Area under ROC (AUROC) is scalar measure for comparison of binary classifiers

Asymmetric distribution of positives and negatives:

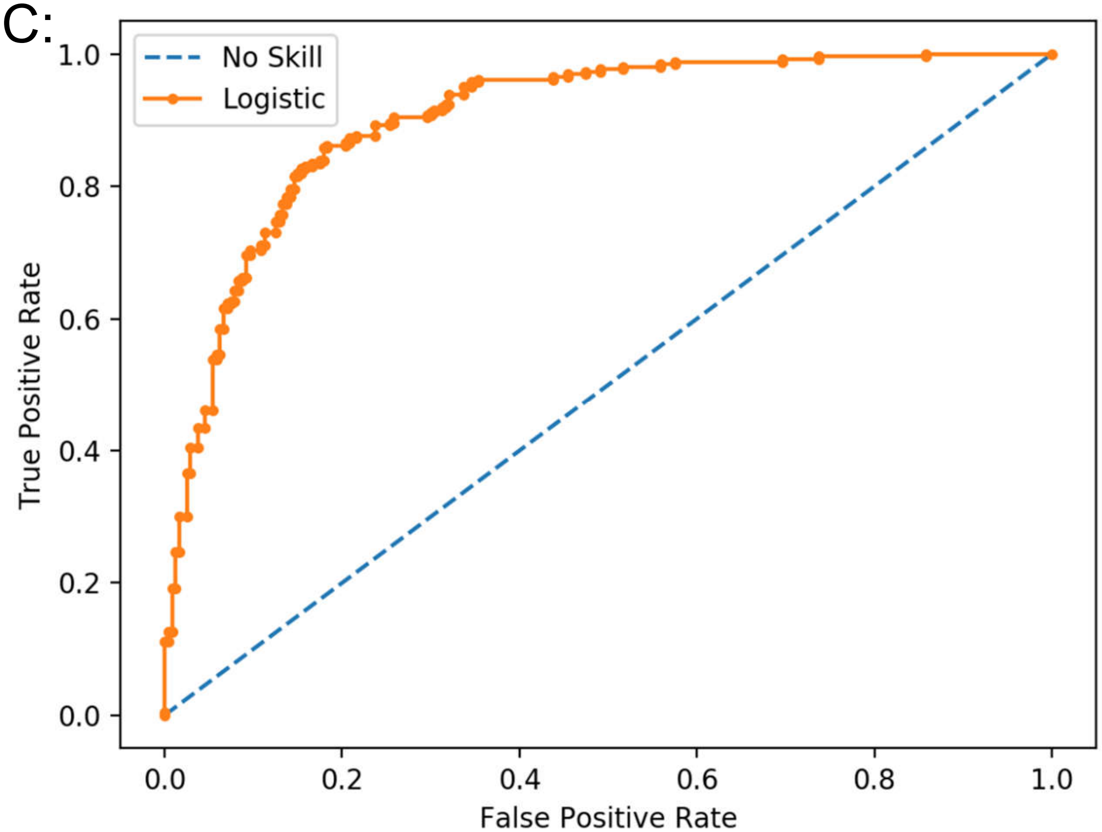
Precision-Recall Curve (PR)

-Visualizes Trade-off between high Precision and high Recall

-better suited than ROC for asymmetric data since true negatives (TN), the majority population, are ignored

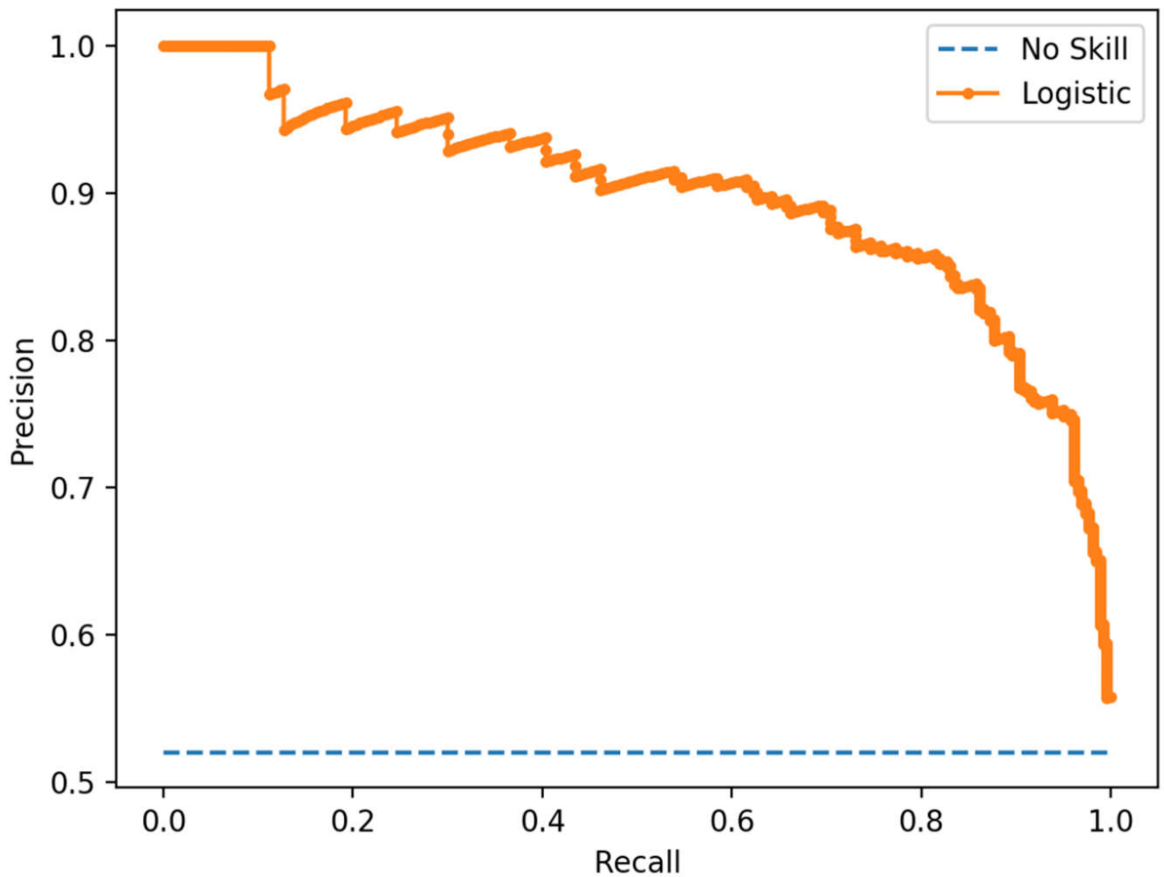
What makes a good (binary) classifier?

(AU)ROC:



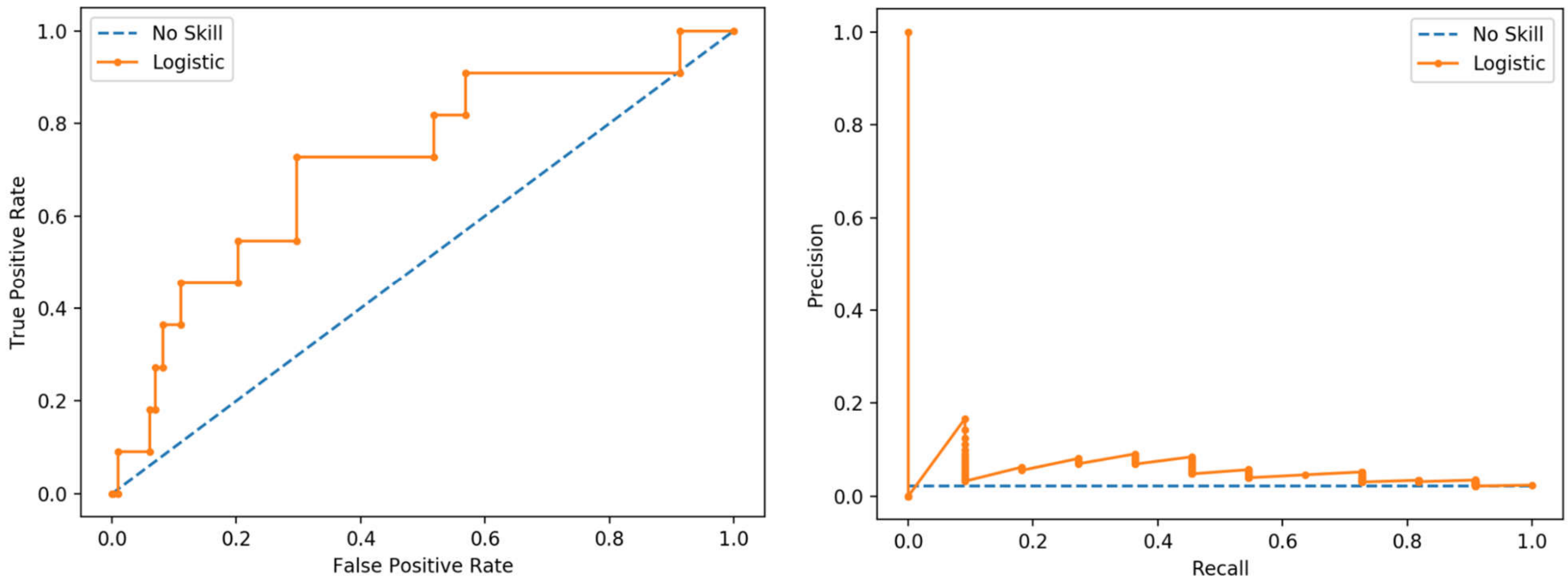
What makes a good (binary) classifier?

PR-Curve



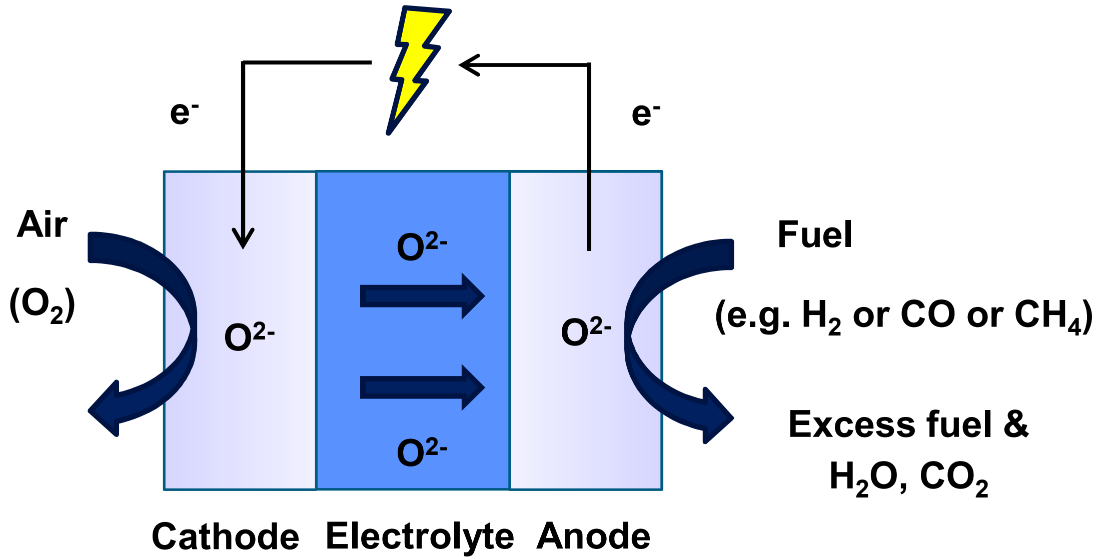
What makes a good (binary) classifier?

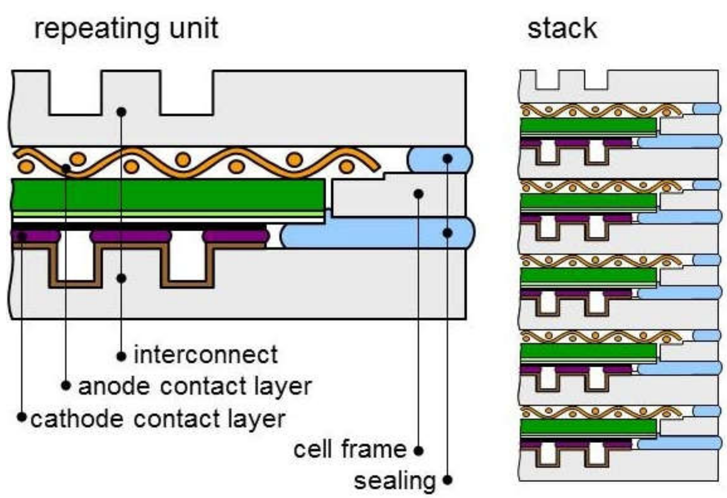
ROC vs PR for samples with 90% negatives:



High fraction of correctly classified negatives can
give optimistic impression of skill of binary classifier

Our exercise: Solid oxide fuel cell (SOFC) and their stacks





Clean "green" energy:
- produce energy from hydrogen or methanol

Use stacks of SOFC
- increase voltage
- use steel to connect

Our exercise: Solid oxide fuel cell (SOFC)

During operation:

- Cr_2O_3 containing oxide scale forms on steel

  evaporation of gaseous Cr (CrO_3 or $CrO_2(OH)_2$)

- Sr is reactive element in the cathode

  Sr→SrO which segregates out of cathode

- Reaction of SrO & Cr species

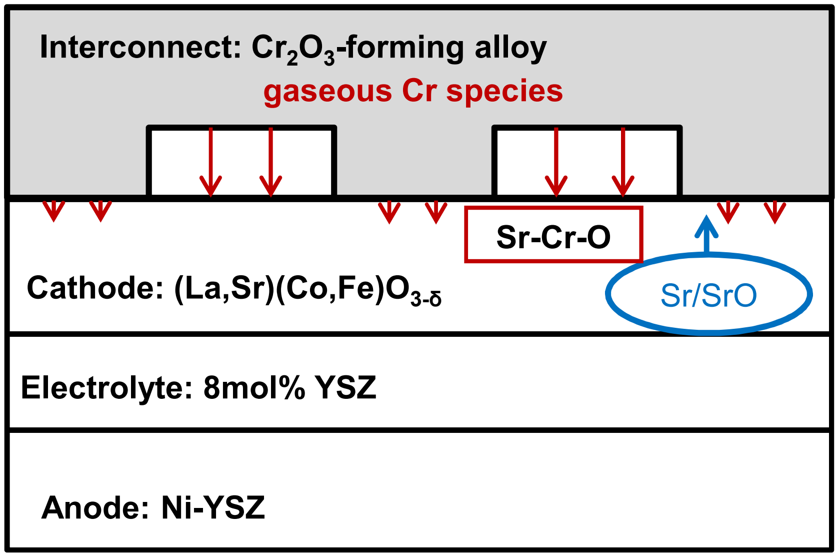
  secondary phases that block oxygen reduction / functionality

$$SrO + CrO_3 \leftrightharpoons SrCrO_4$$

$$SrO + CrO_3 \leftrightharpoons SrCrO_3 + \frac{1}{2}O_2$$

$$SrO + \frac{2}{3}CrO_3 \leftrightharpoons \frac{1}{3}Sr_3Cr_2O_8 + \frac{1}{6}O_2$$

$$SrO + \frac{1}{2}CrO_3 \leftrightharpoons \frac{1}{2}Sr_2CrO_4 + \frac{1}{4}O_2$$



Our exercise: Solid oxide fuel cell (SOFC)

CSV file with data

Response/Target: chemical component

last column

Parameters first columns:

- Partial pressure of CrO_3 ($pCrO_3$)
- local oxygen partial pressure (pO_2)