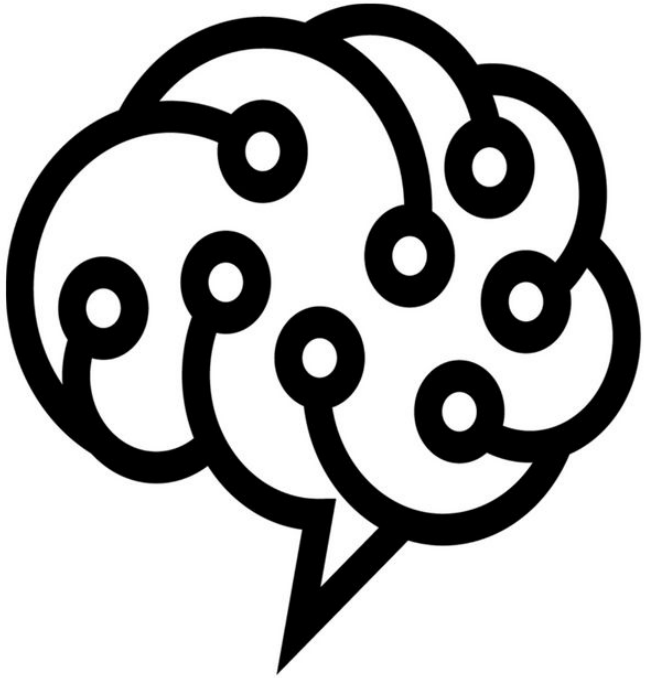


school of ai

Support Vector Machines

https://github.com/wilcobonestroo/schoolofai

Support Vector Machines

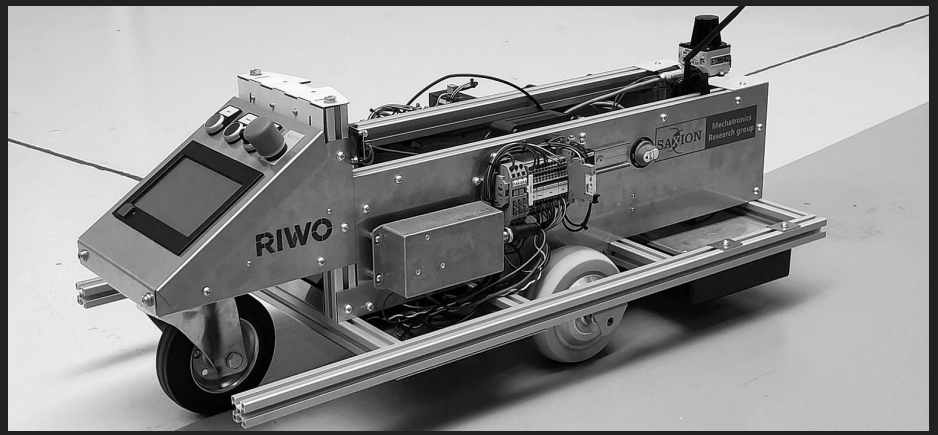
Theory and Code

Wilco Bonestroo

Researcher at Mechatronics research group at Saxion

Software and AI for robots

Autonomous mobile robots

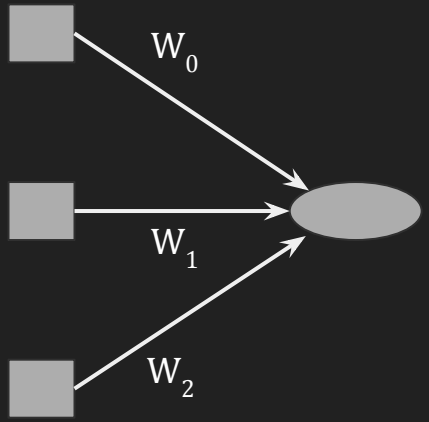


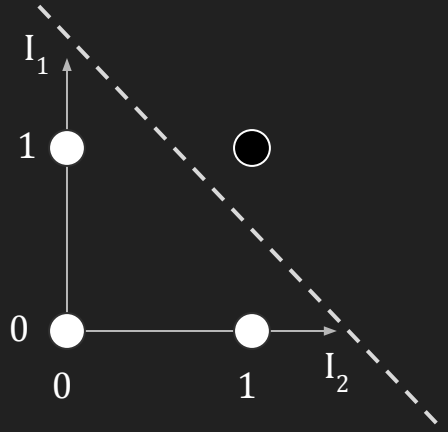
Research group Mechatronics

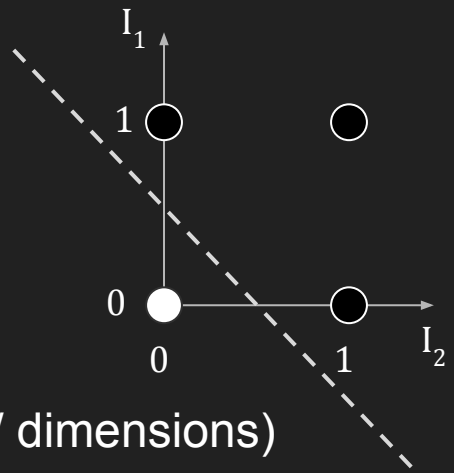
- Unmanned Robotics Systems
- Smart Industrial Systems

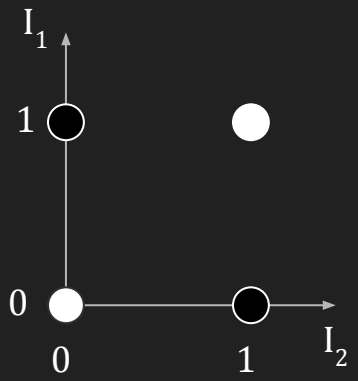
saxion.nl/mechatronics

Look back: Perceptrons



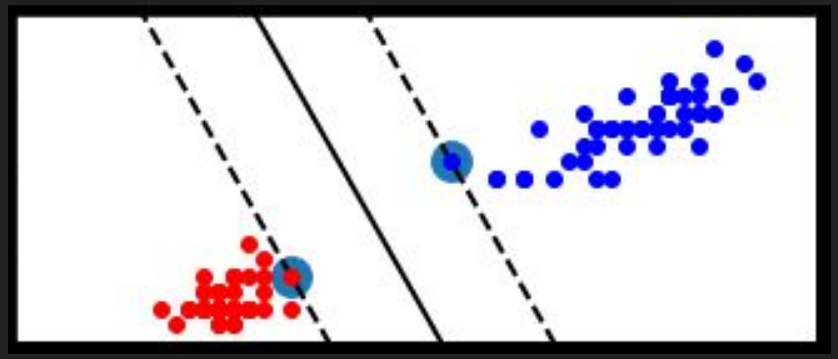






Linear Separable (two inputs / dimensions)

Super Vector Machines



Machine Learning Algorithm

Basically it is a **linear binary classifier**

But it can also be used for **multiple classes** and for **regression** and **outlier detection**

Is tries to get “the best” separation and uses “support vectors”

Basic idea

Samples

With labels or class, so it is supervised learning

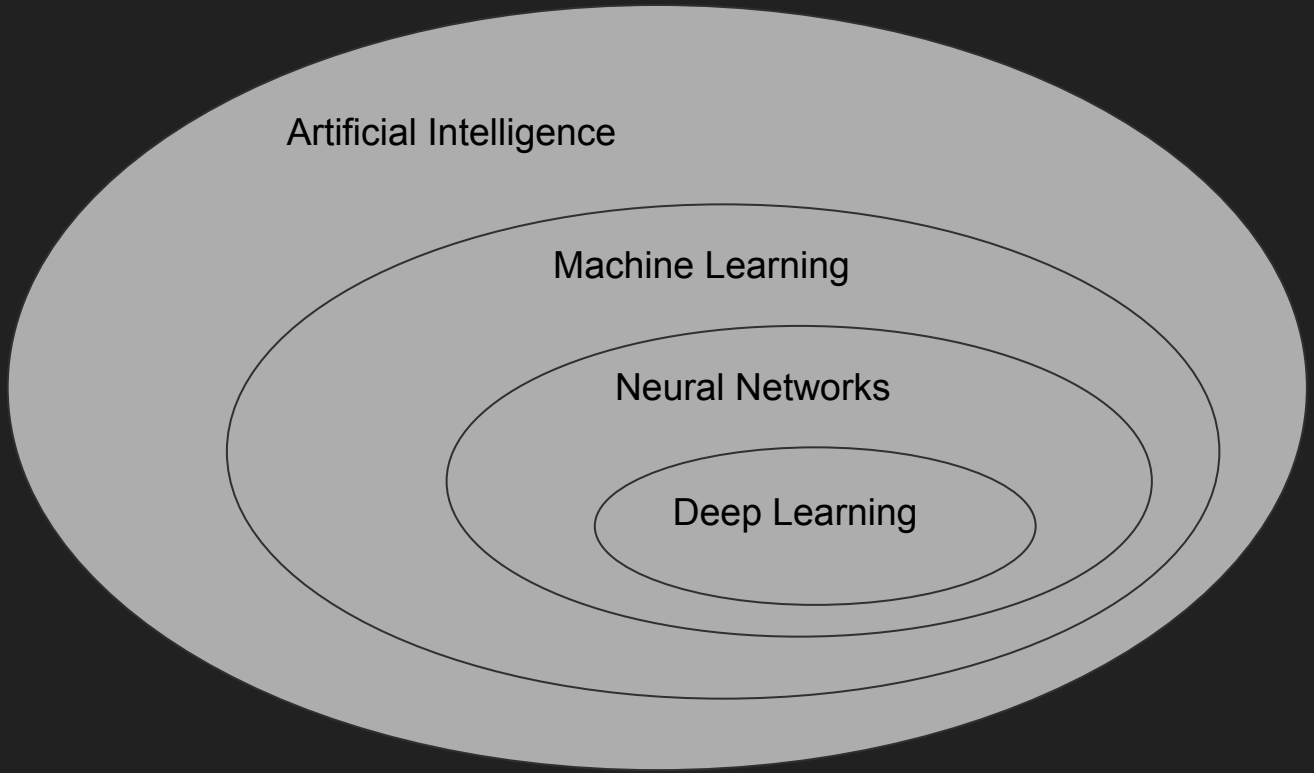
Feature 1	Feature 2	Feature 3	Class
234	456	275	0
26	65	343	1
54	446	2	0
...	...	...	...

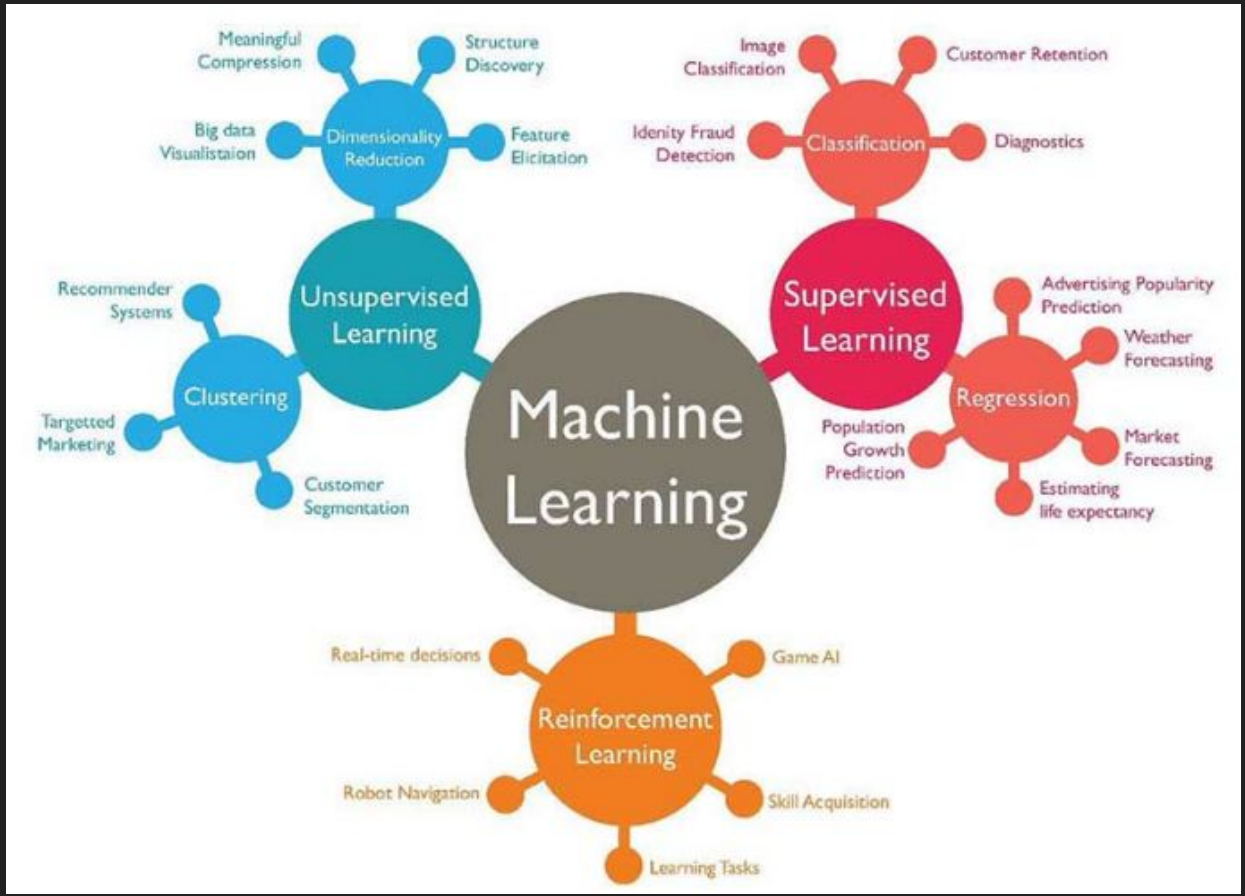
X y

```
classifier.fit(X,y)   #Training

classifier.predict([32,624,6])
```

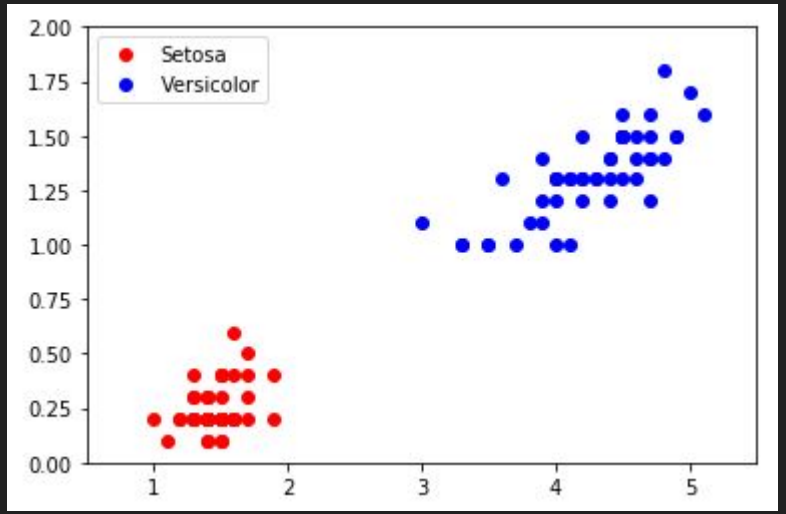
The bigger picture

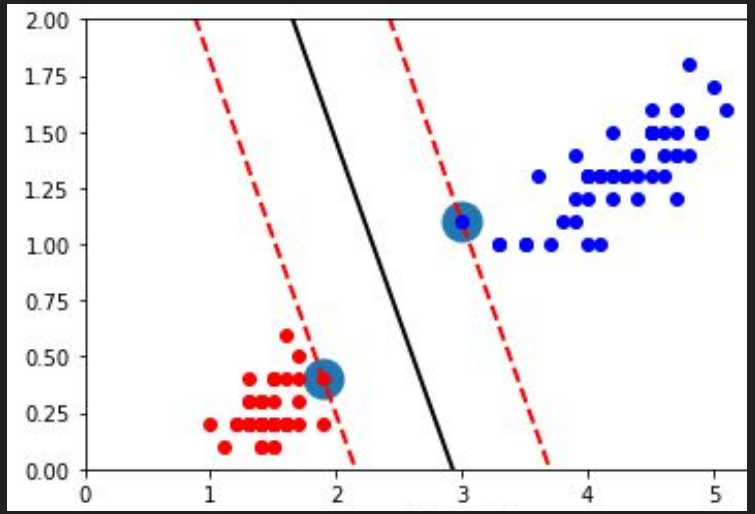




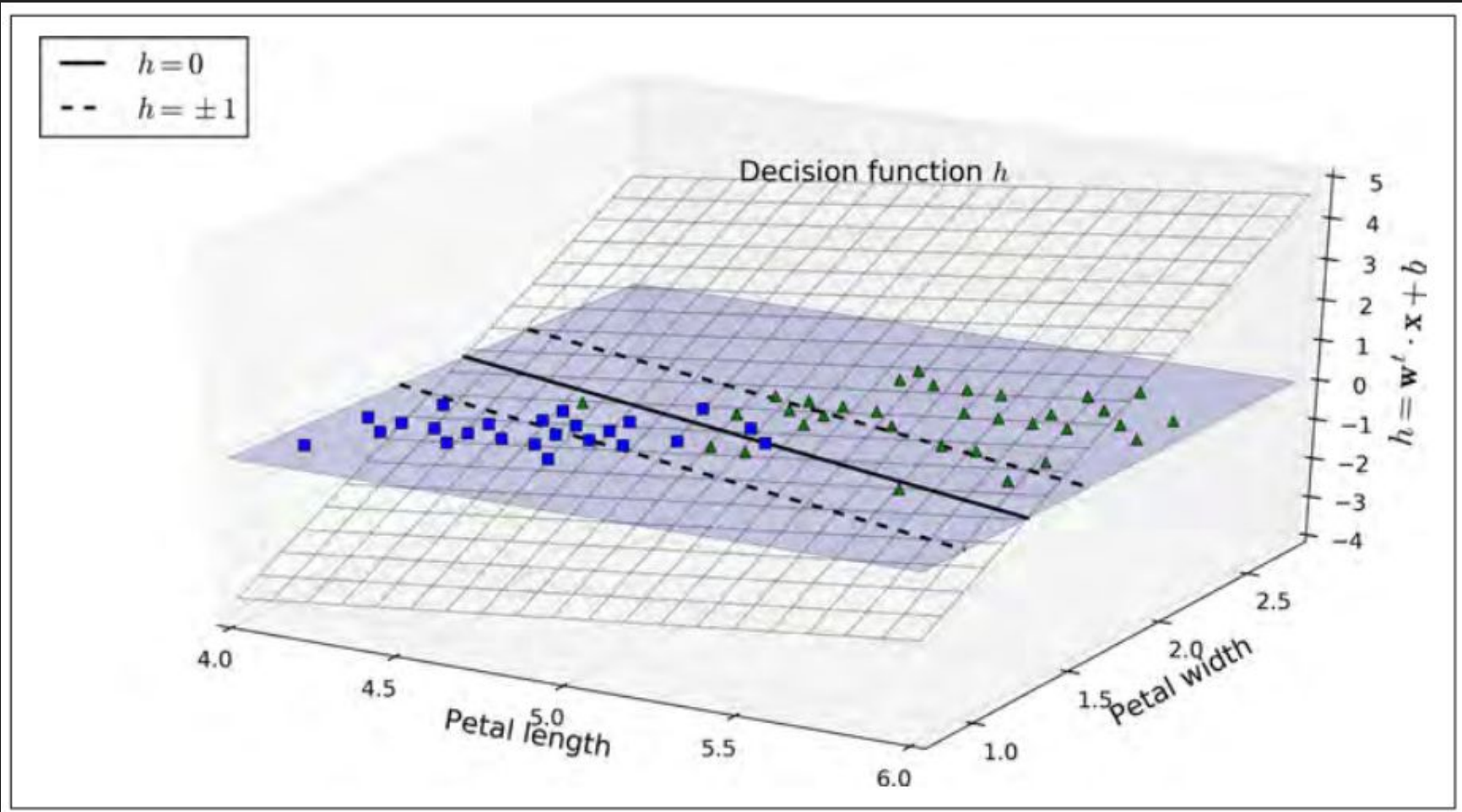
Source: https://datavibes.nl/wat-is-machine-learning/types-of-machine-learning-expertise-datavibes/

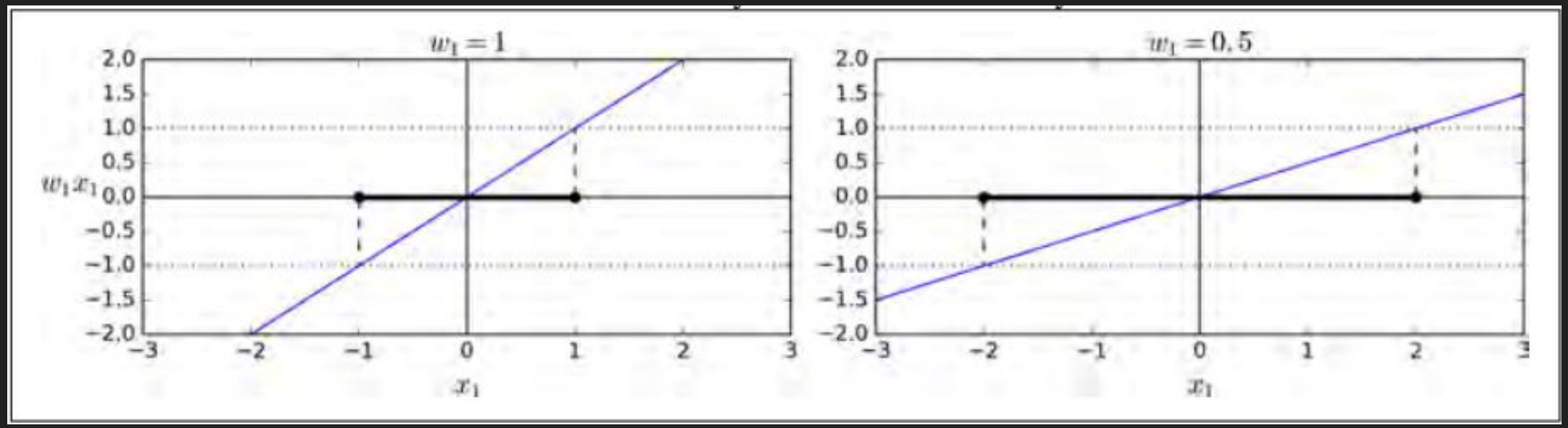
The “best” separation





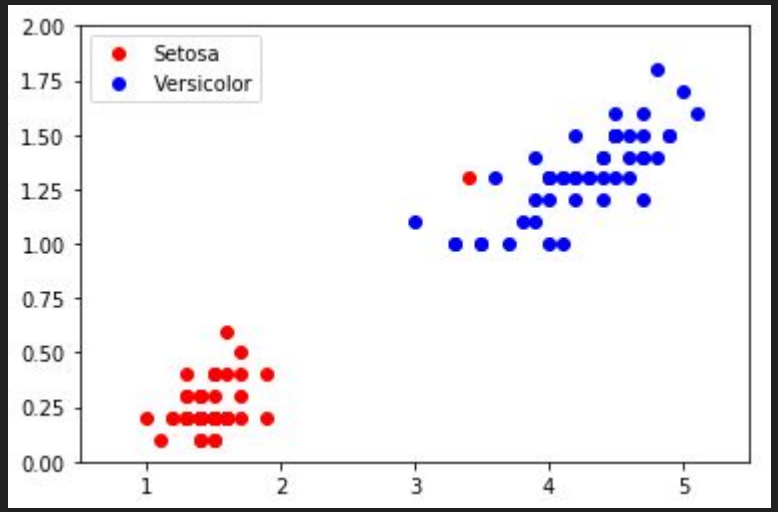
The “widest road” between the groups

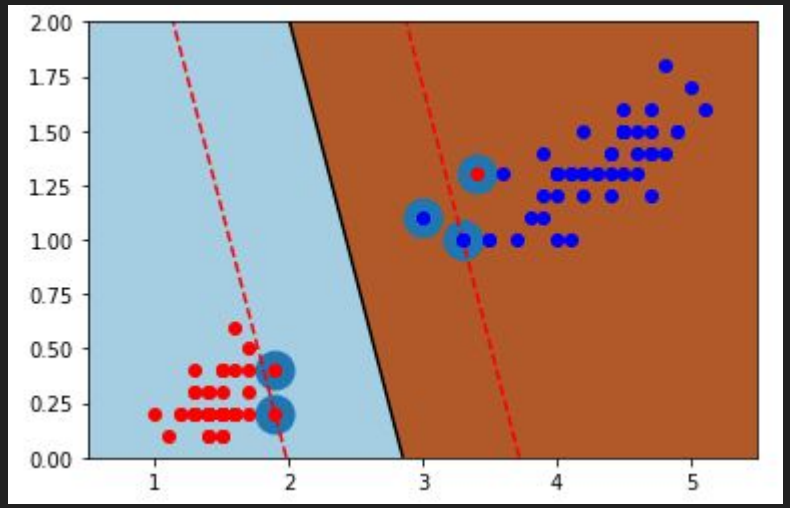

$h = 0$
$h = \pm 1$
Decision function h
$h = \mathbf{w}^t \cdot \mathbf{x} + b$
5
4
3
2
1
0
−1
−2
−3
−4
4.0
4.5
5.0
5.5
6.0
Petal length
1.0
1.5
2.0
2.5
Petal width



Minimizing the weight results in a wider margin

What about this?





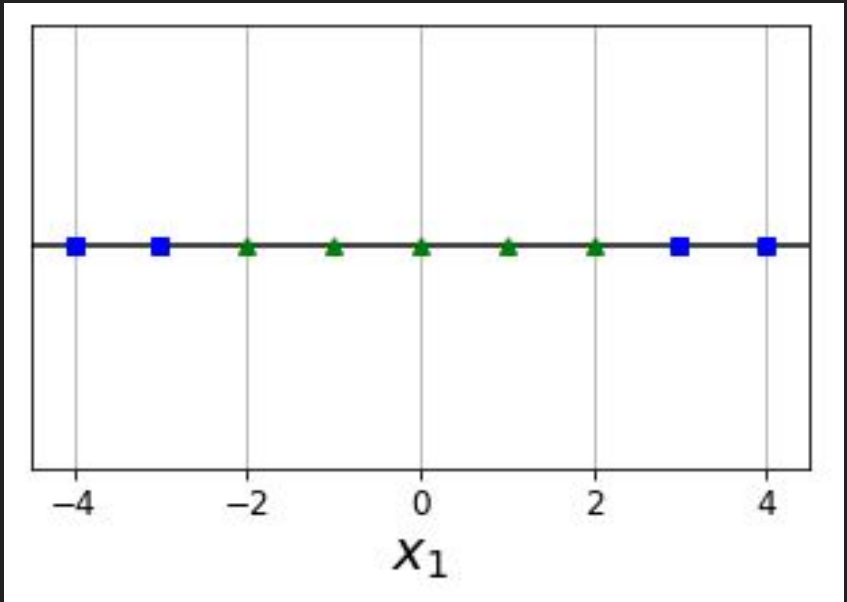
We can relax the constraints of the SVM. This is called **Soft Margin**.

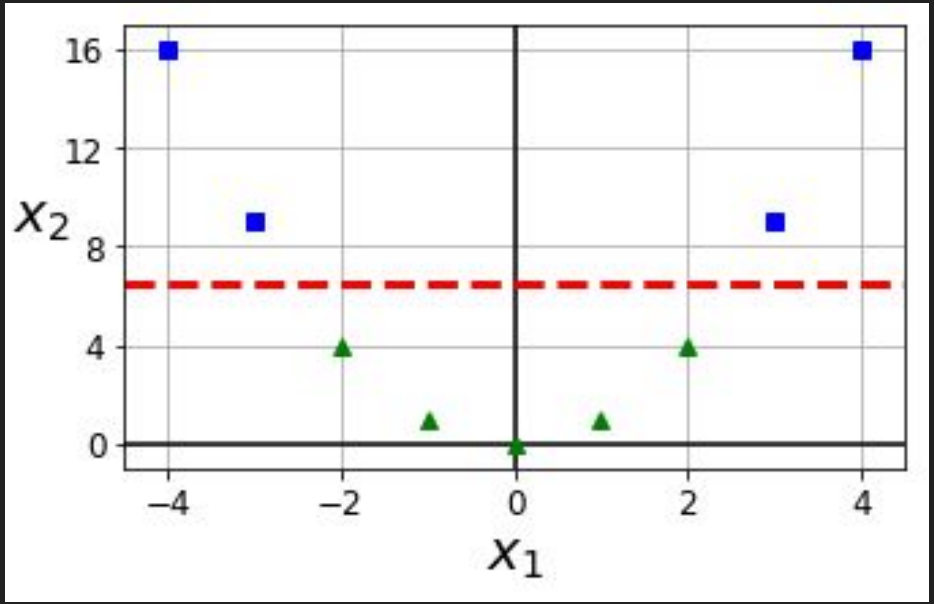
Soft margin SVM

$$\underset{\mathbf{w}, b, \zeta}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^T \cdot \mathbf{w} + C \sum_{i=1}^{m} \zeta^{(i)}$$

Weights + C Mistakes

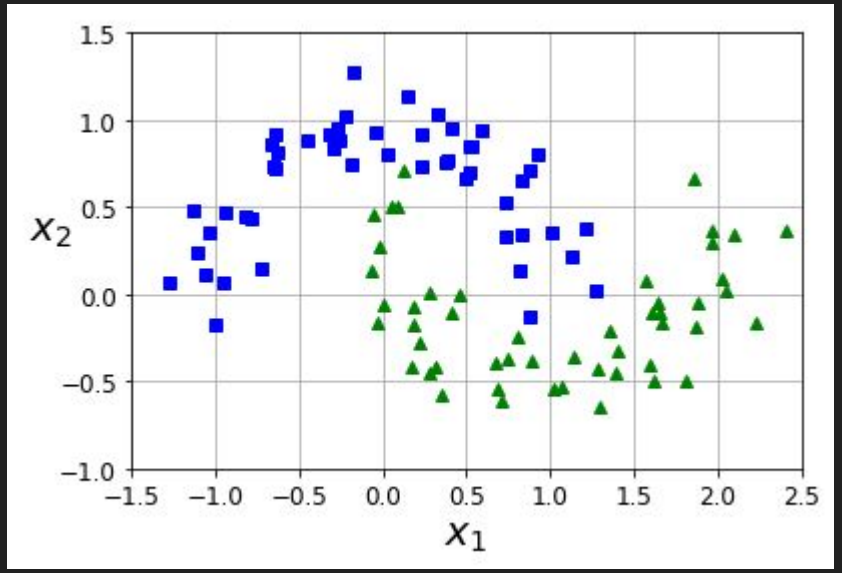
What if data is not separable?

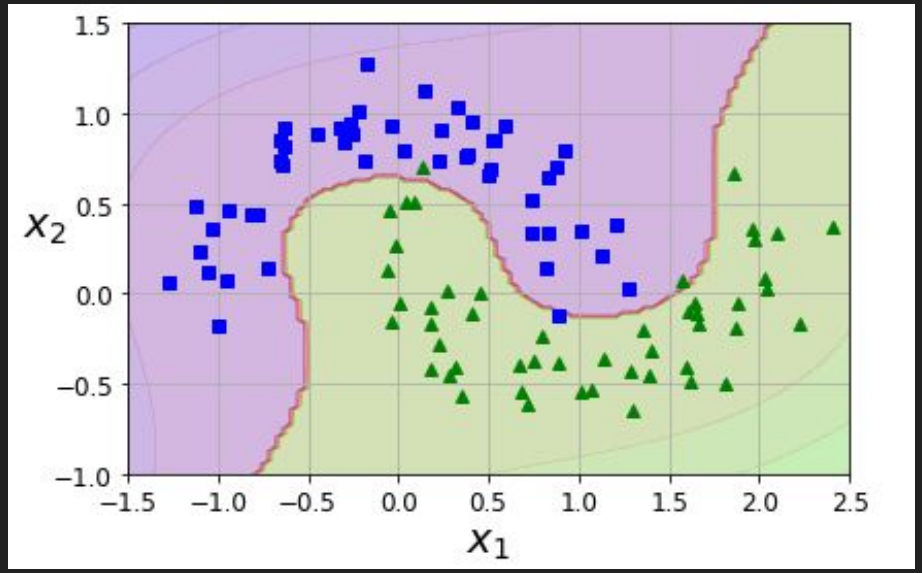




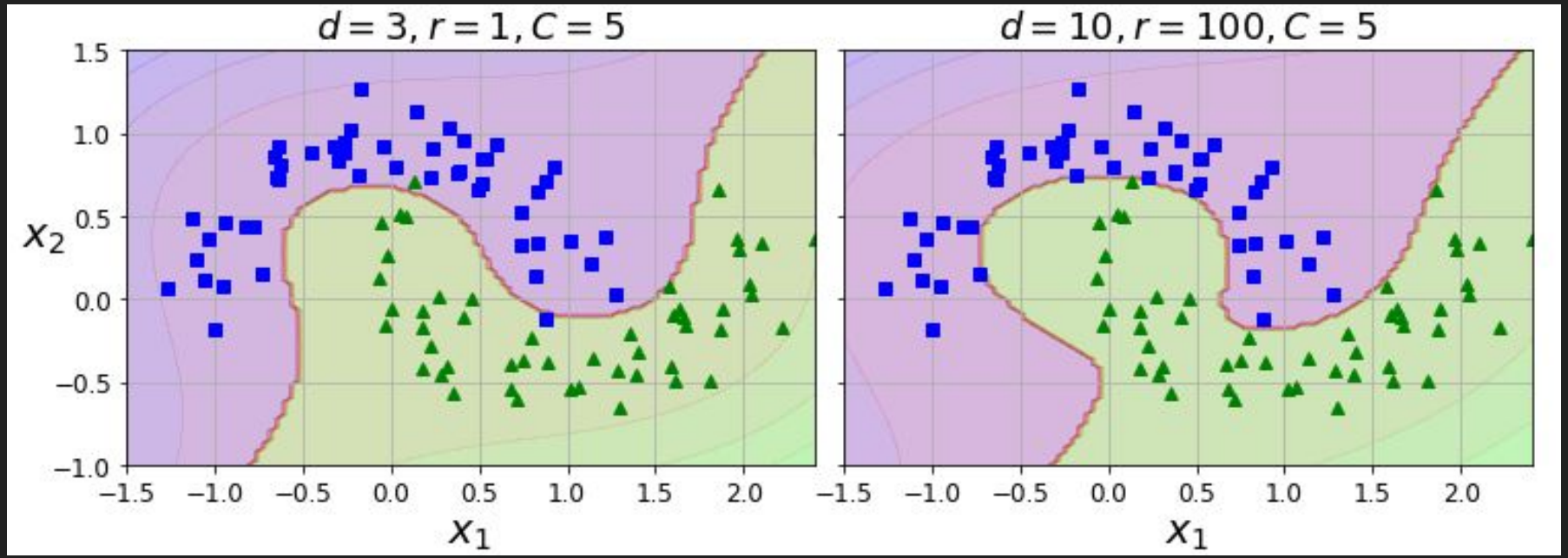
Add features

Add polynomial features



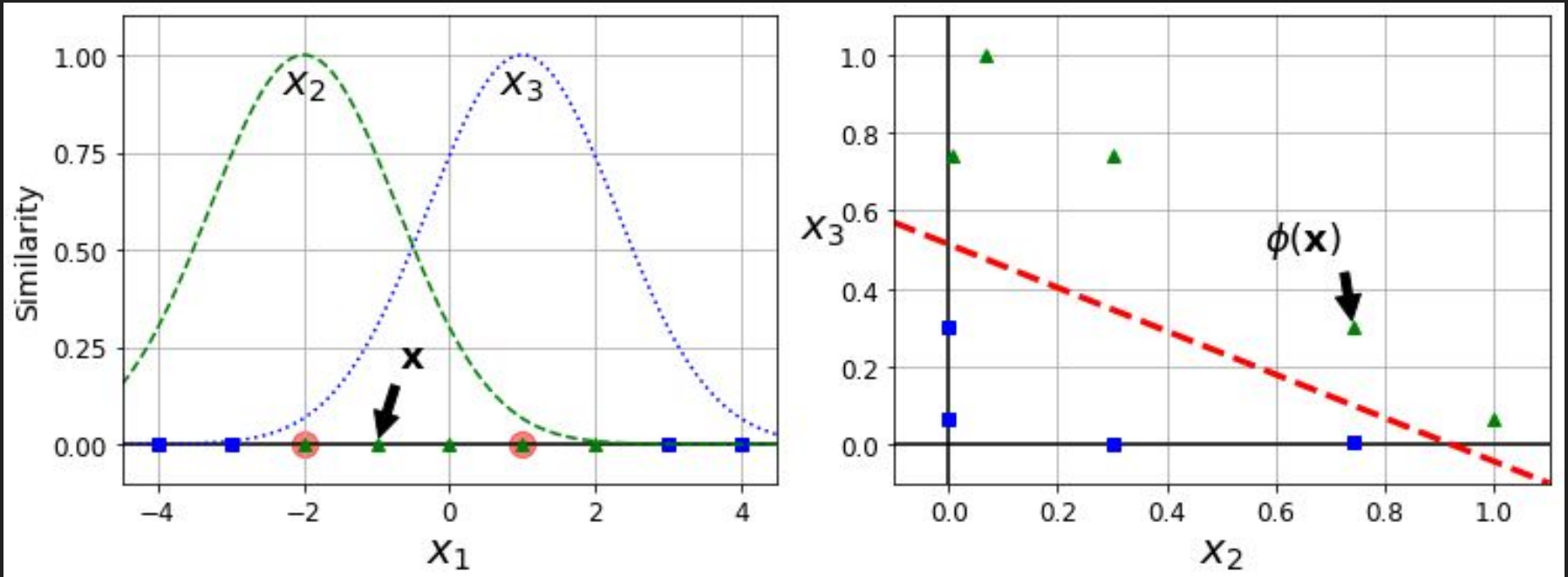


Kernel Trick polynomial

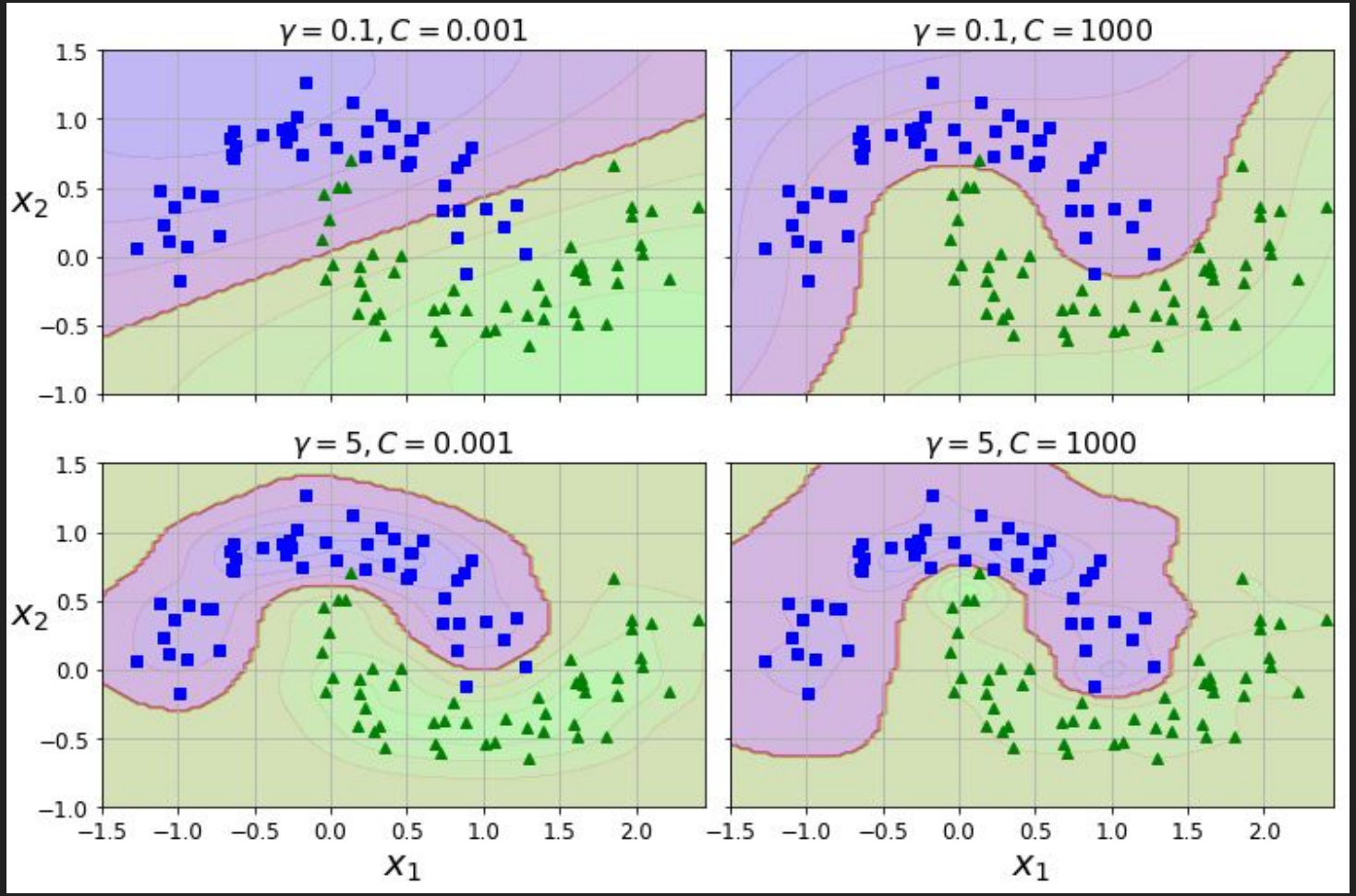


Add similarity features

Radial Basis Function (RBF)



Kernel with similarity



Conclusion

It is pretty powerful for a “linear binary classifier”

From literature:

It is not super efficient. Works for complex, but not too large datasets.

Try it yourself

Go to: https://github.com/wilcobonestroo/schoolofai

Go to Support Vector Machines folder.

Click on .ipynb file

Click “run in Google Colab”

Or download the .ipynb file and run it on your own Jupyter Notebook

References

This material is based on the book Hands-On Machine Learning with Scikit-Learn and TensorFlow by Aurélien Géron (O'Reilly) and on the accompanying Github repository.

Scikit Learn website:
https://scikit-learn.or