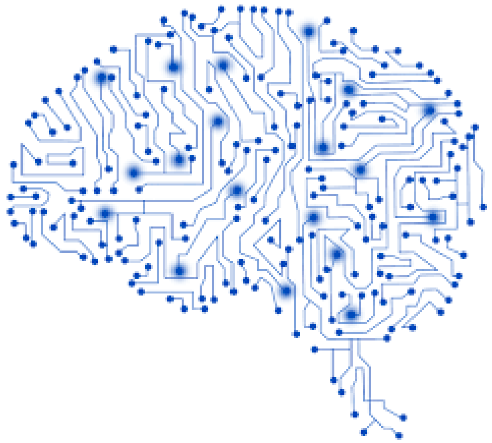
University of Minho
School of Engineering

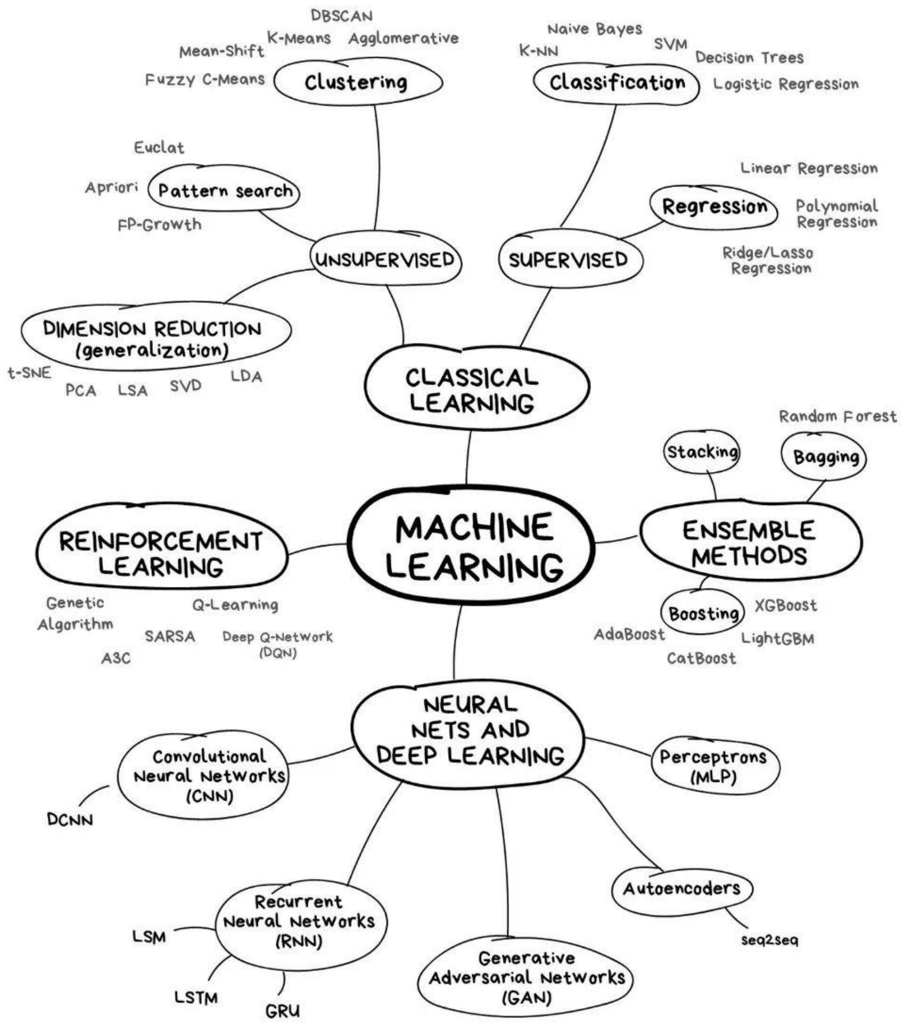


Dados e Aprendizagem Automática

Support Vector Machine

DAA @ MEI/1º ano – 1º Semestre

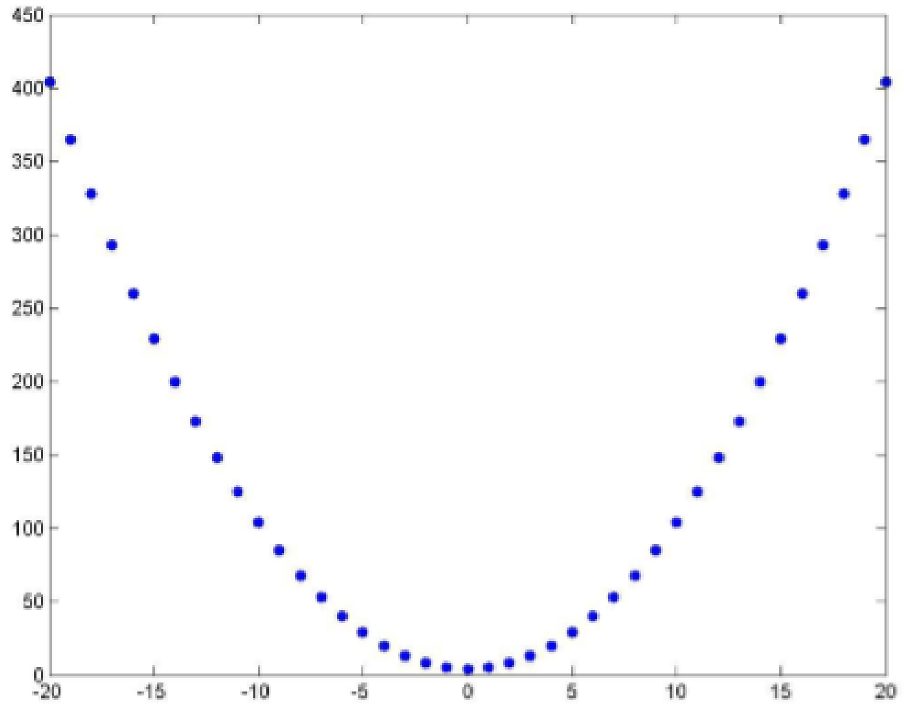
Context



Source: The map of the machine learning world
Vasily Zubarev (vas3k.com)

Problem

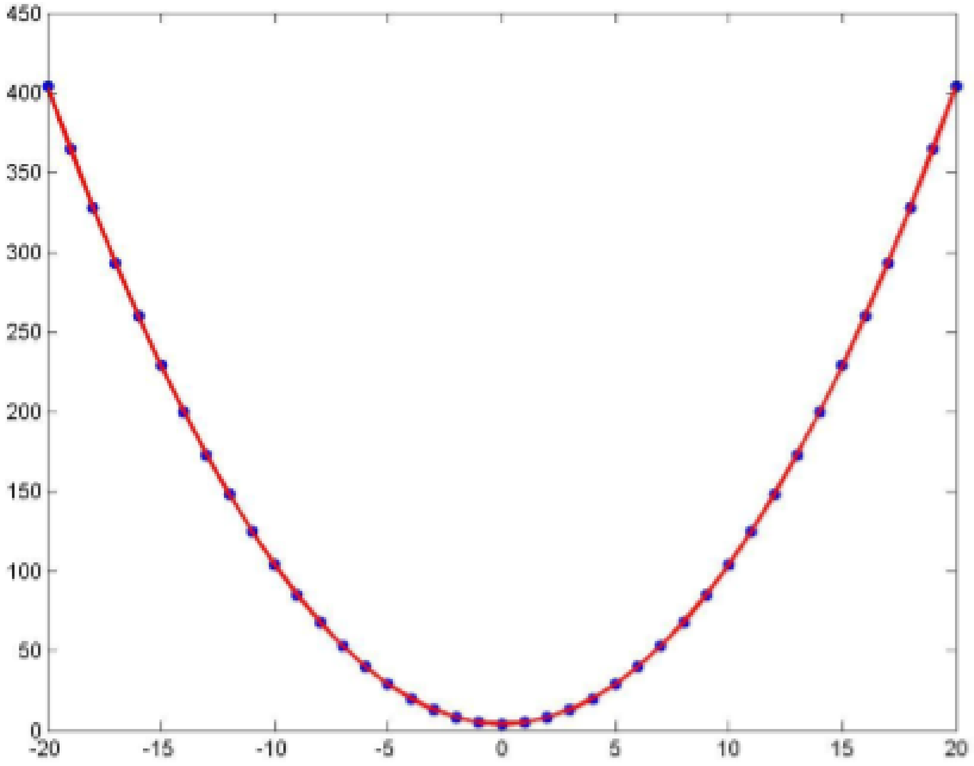
- Predict y given x



- Try to fit a function to describe this

Problem

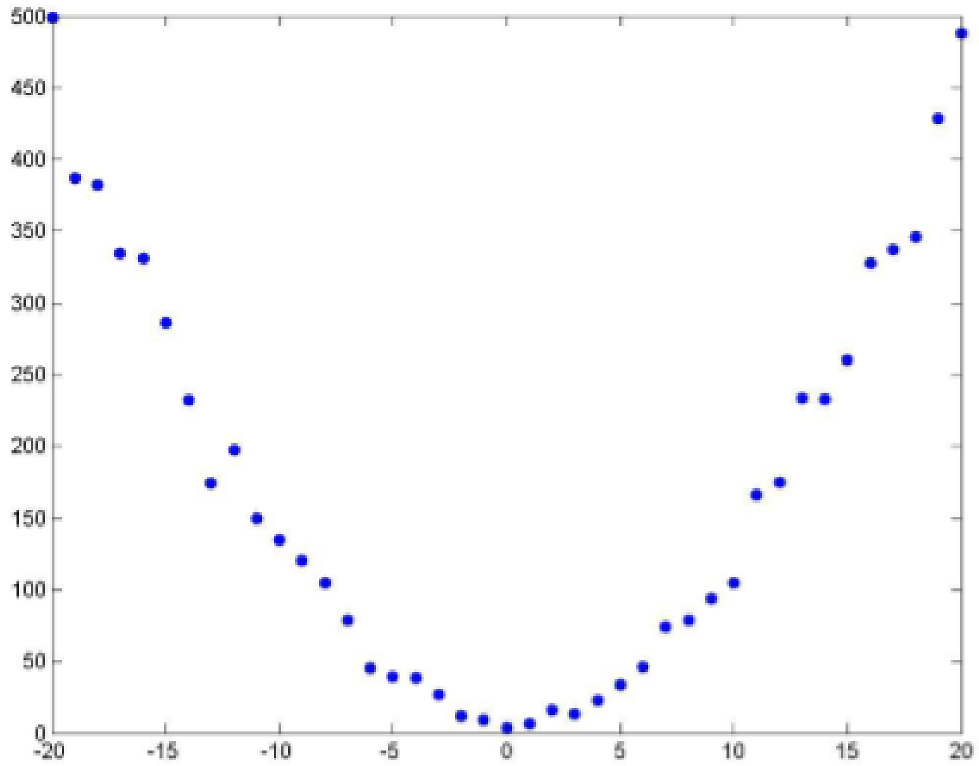
- Easy!



- For a new point we will be correct

Problem

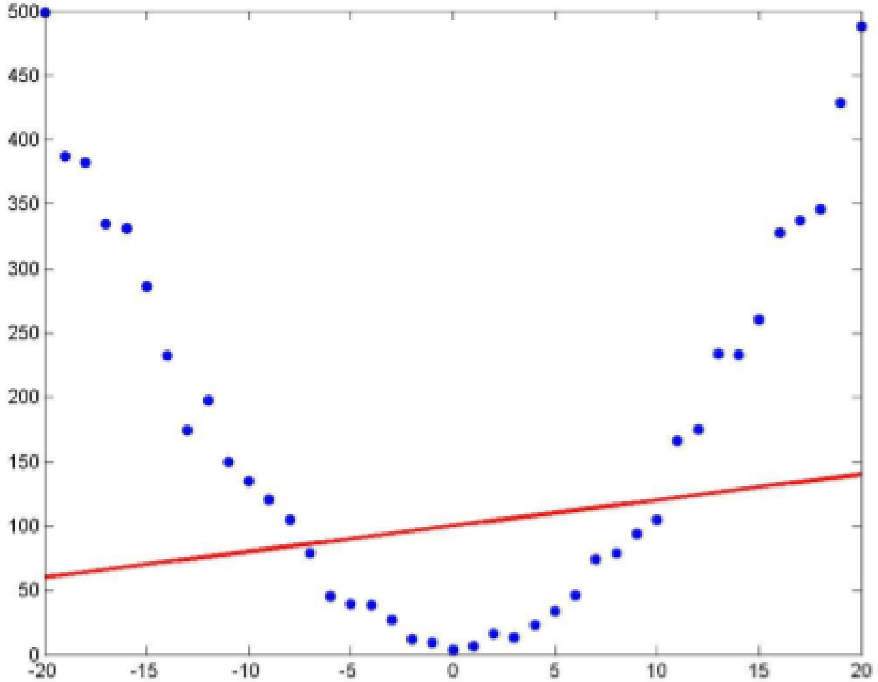
- What if we add some noise?



- In real environment (data) we cannot “see” the function

Problem

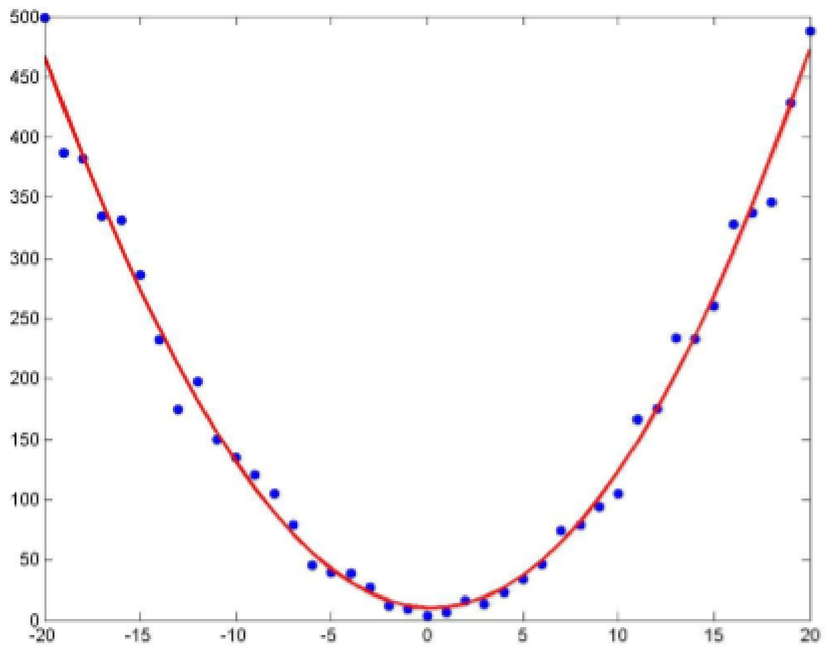
- We could assume the relationship to be linear



- How wrong are we?
- How do we know which parameters are the best?

Problem

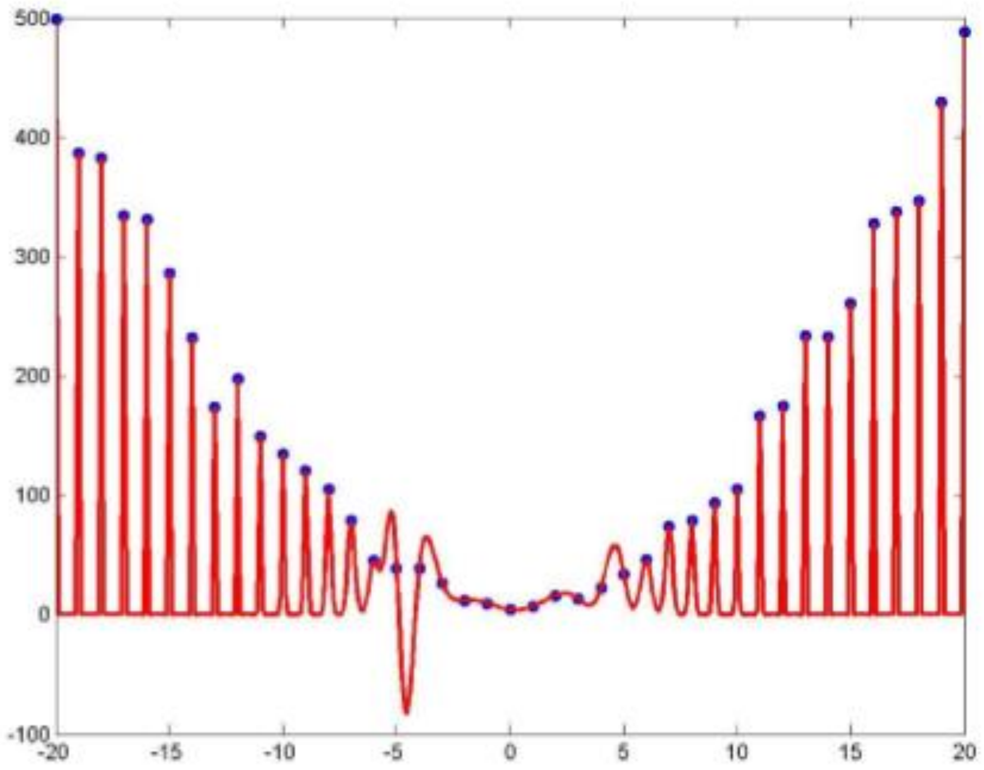
- Linear is still bad
- Increase parameters, and we can consider a quadratic function



- This is better
- But still has some loss
- Increase parameters!

Problem

- Zero error on training set!
- What about a test example?
- Too specific = overfitting
- Empirical risk minimization: overfits if you follow it blindly



Problem

- How to avoid it?
- Need to balance training error and capacity of a function:
  - Too complex a function will overfit;
  - Not enough complexity will not generalize well either.

Support Vector Machine

Support Vector Machine

Support Vector Machine is a supervised Machine Learning algorithm that can be used for both classification (mostly) or regression problems.

The main idea is to plot each data item as a point in an n-dimensional space (n is the number of features), performing classification by finding the hyper-plane that differentiates the classes.

Support Vector Machine

- Works well for classifying higher-dimensional data (datasets with lots of features)
- Finds higher-dimensional support vectors which “divides” the data
- Applies kernels to represent data in higher-dimensional spaces to find hyperplanes that might not be apparent in lower dimensions

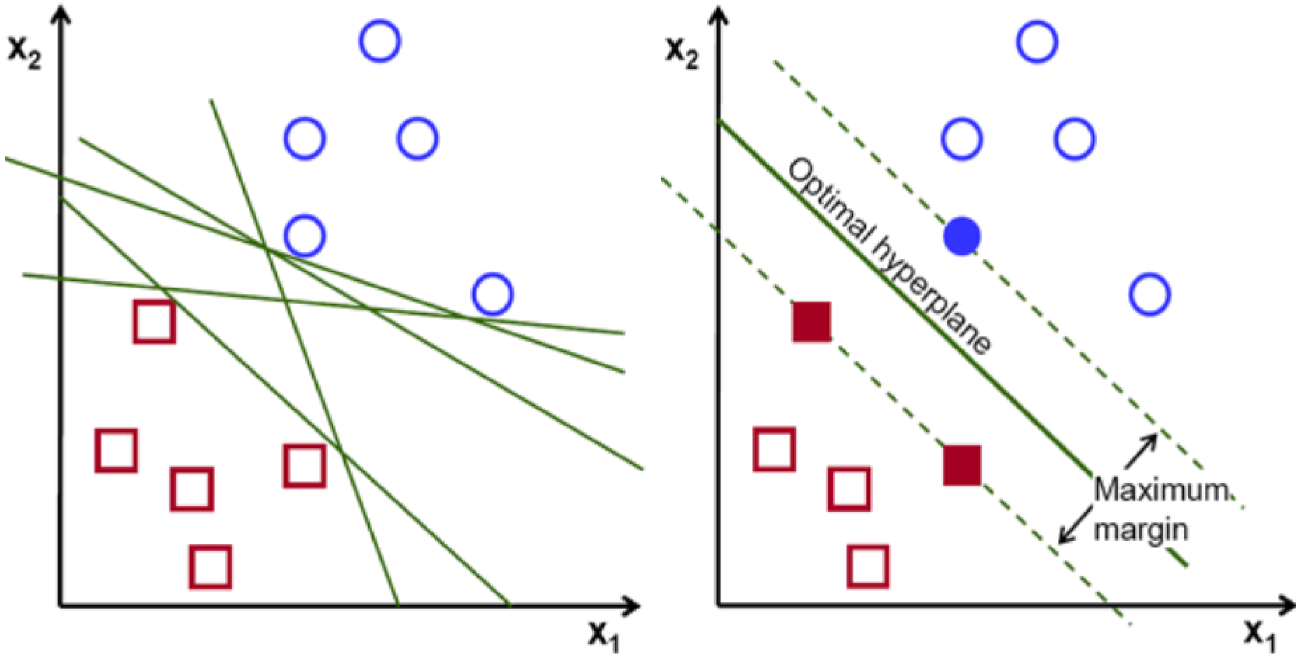
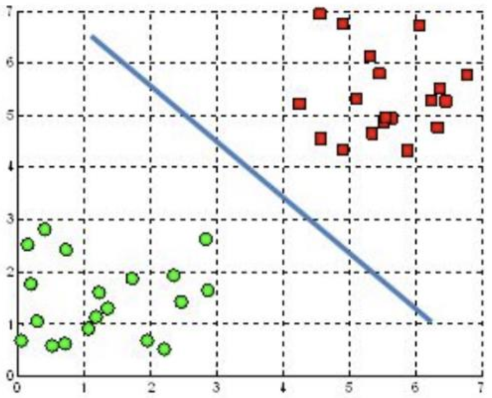


Image source: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

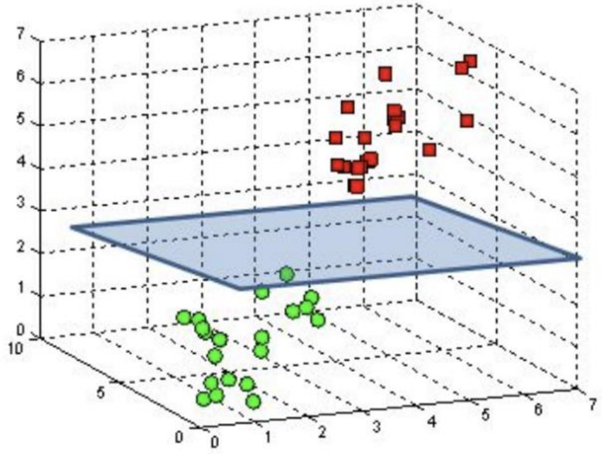
Support Vector Machine

- Hyperplanes are decision boundaries that help classify the data points;
- Data points falling on either side of the hyperplane can be attributed to different classes;
- The dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane;
- Difficult to imagine when the number of features exceeds 3.

A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane

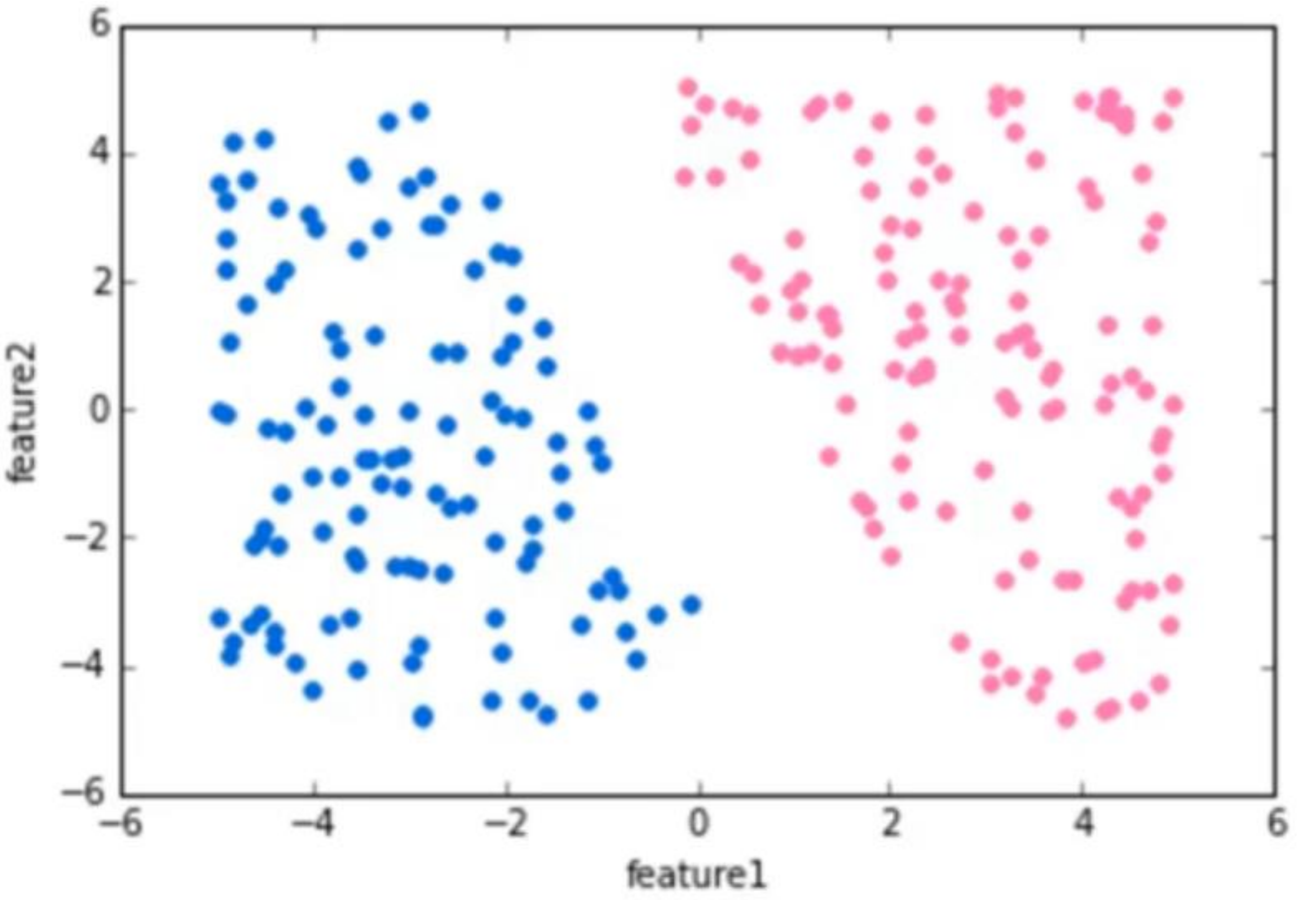


Source: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Support Vector Machine

- How it works?

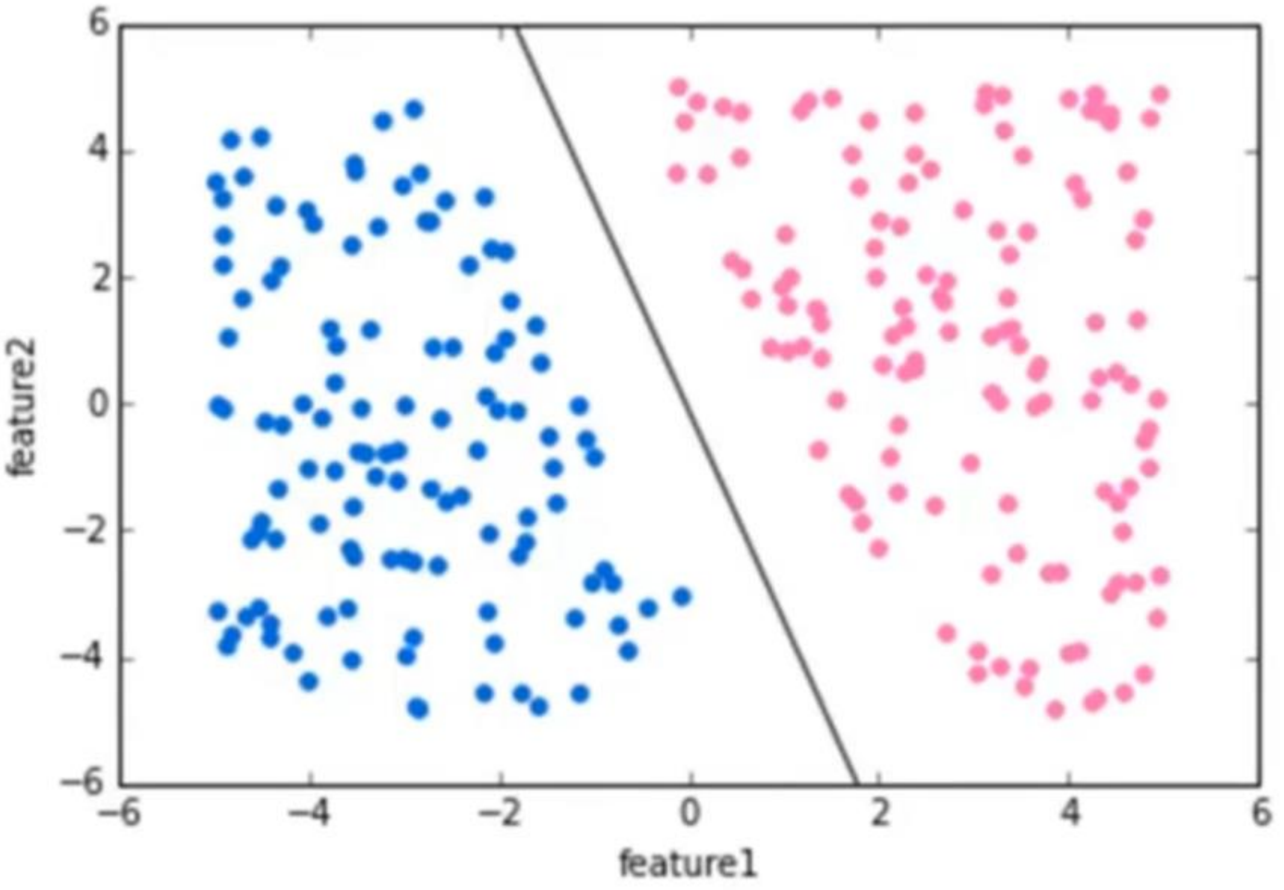
1) Image a labeled training data



Support Vector Machine

- How it works?

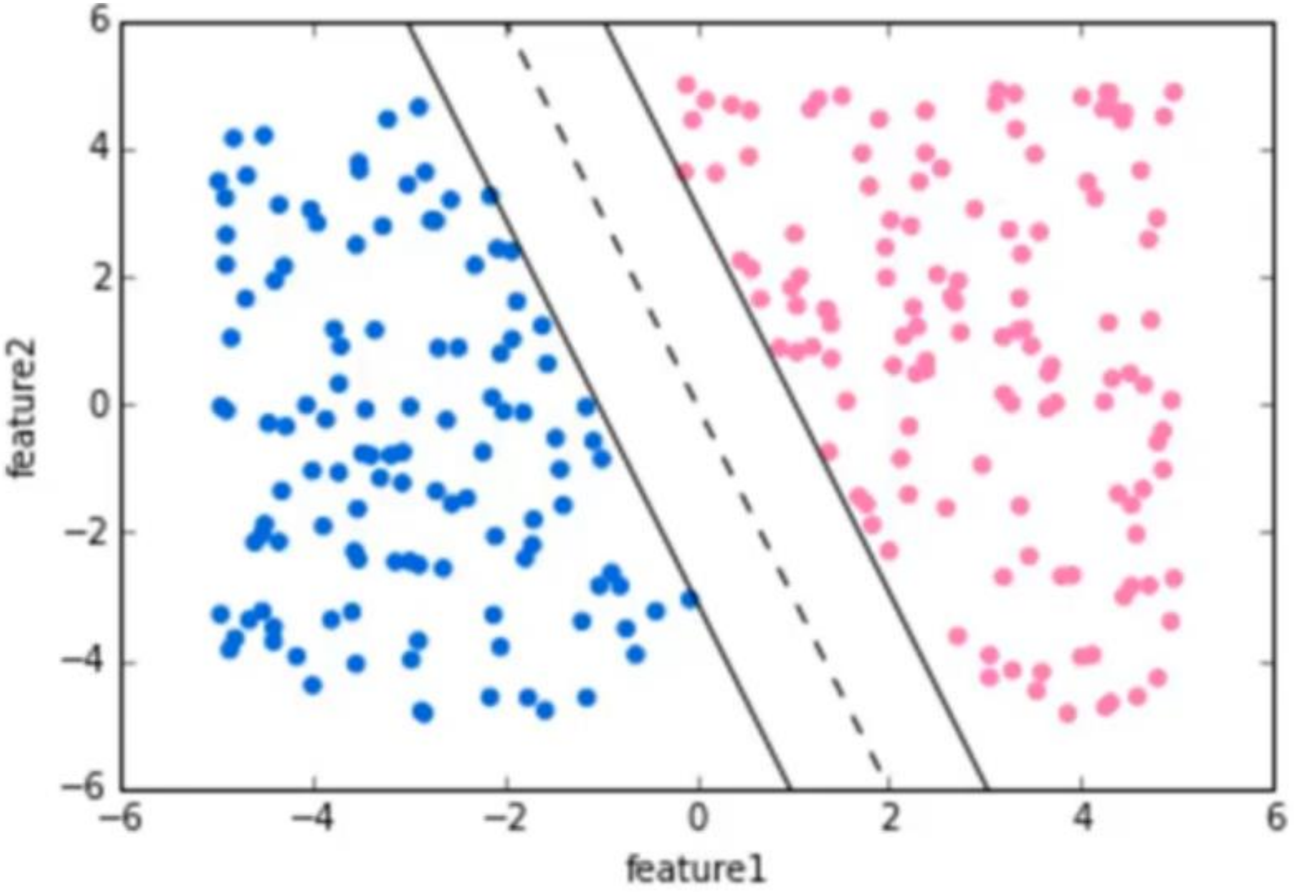
1) Image a labeled training data
2) Draw a separating "hyperplane" between the classes



Support Vector Machine

- How it works?

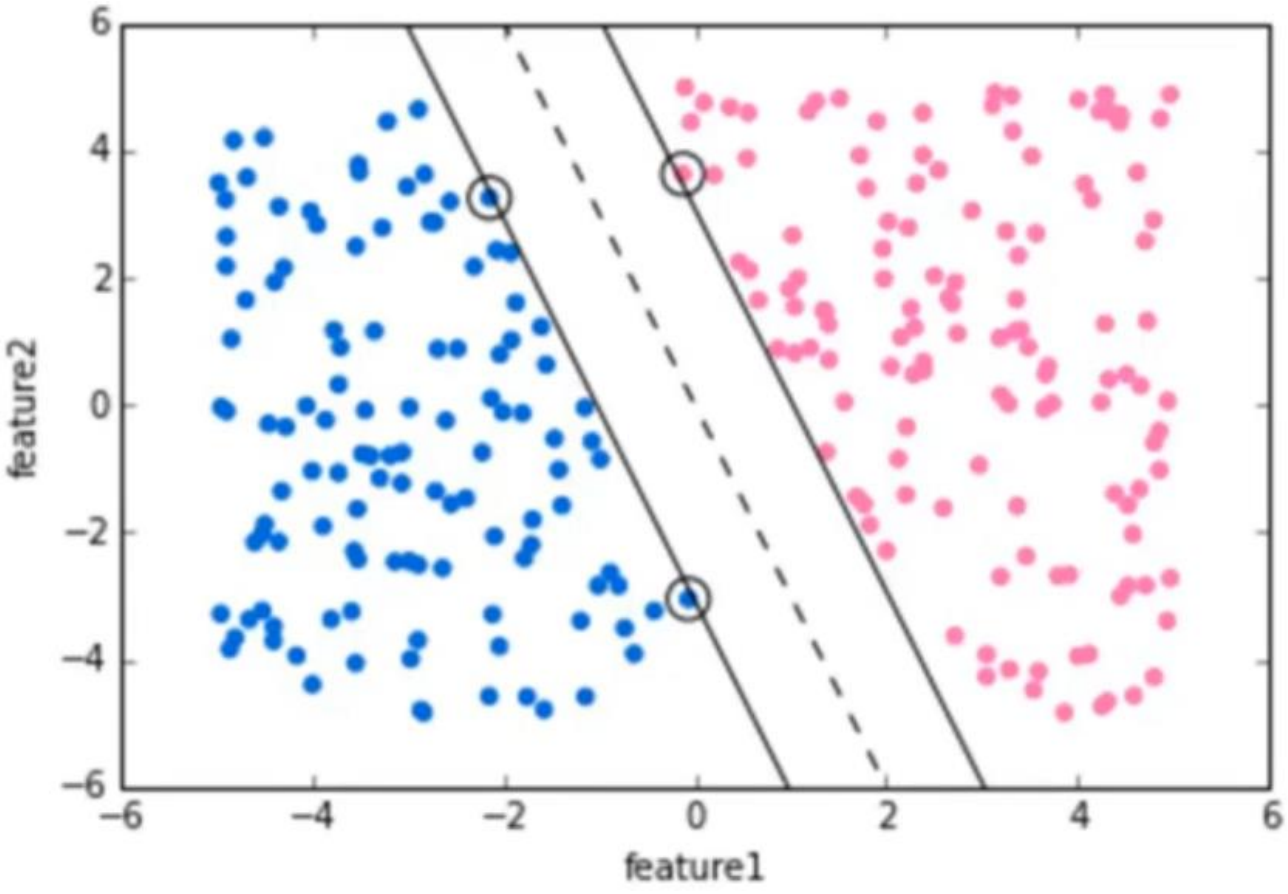
1) Image a labeled training data
2) Draw a separating “hyperplane” between the classes – many options that separate perfectly…
3) Choose a hyperplane that maximizes the margin between classes



Support Vector Machine

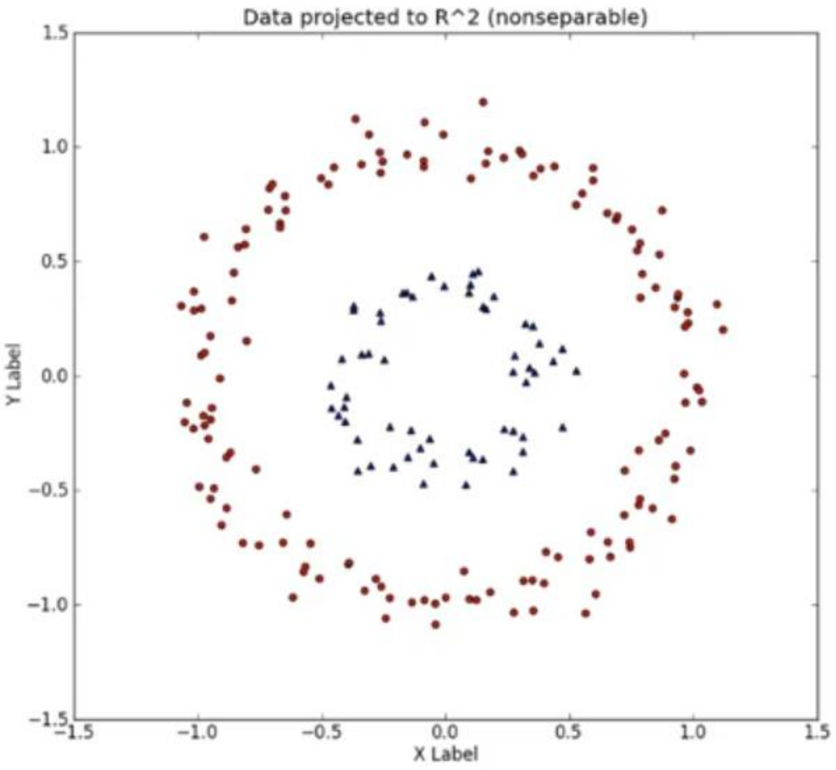
- How it works?

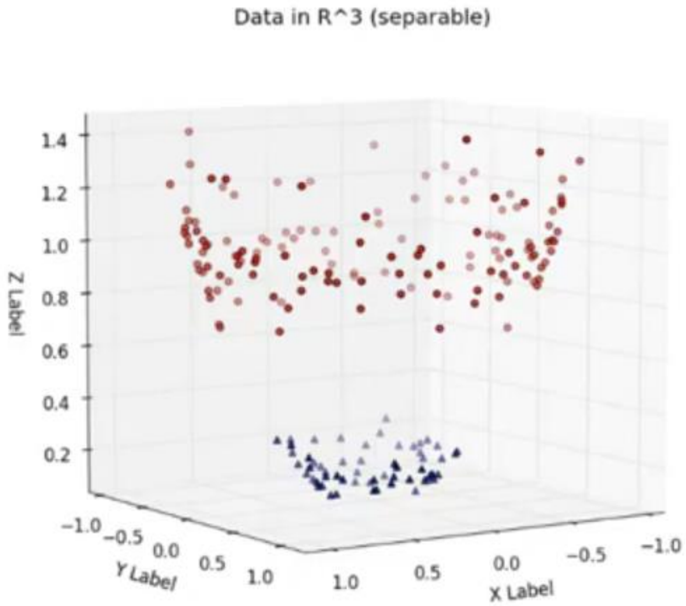
1) Image a labeled training data
2) Draw a separating "hyperplane" between the classes – many options that separate perfectly…
3) Choose a hyperplane that maximizes the margin between classes – vector points that the margin lines touch are known as Support Vectors



Support Vector Machine

- The idea can be expanded to non-linearly separable data through the "kernel trick"

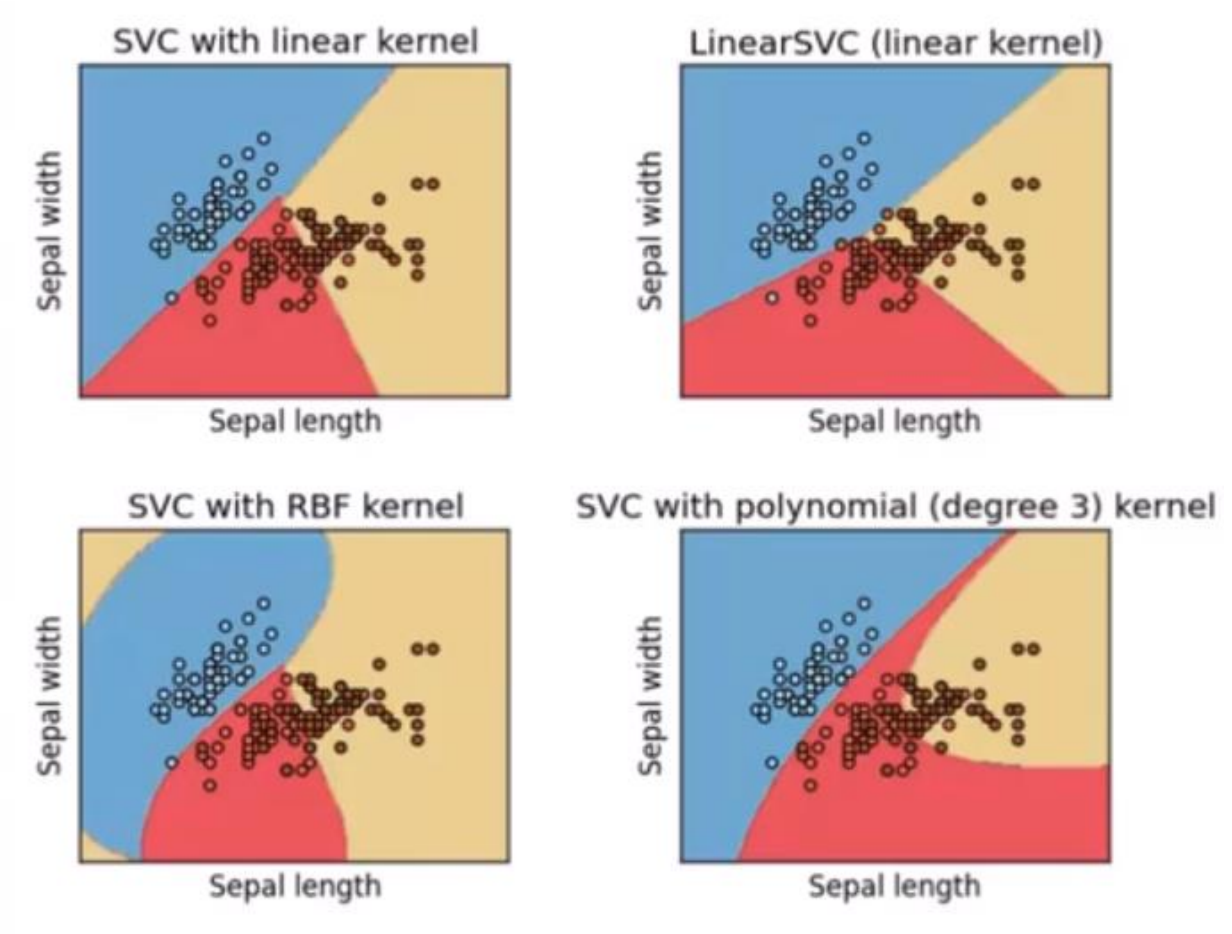




Support Vector Machine

Support Vector Classification

- Use of SVC to classify data using SVM
- Apply different "kernels" with SVC
- Different kernels provide different results for a given dataset



Loss Functions

- We are looking to maximize the margin between the data points and the defined hyperplane. For that, SVMs use the hinge loss:

$$\ell(y) = \max(0, 1 - t \cdot y)$$

Return 0 if the predicted value and the real value have the same sign.

Otherwise: calculate loss!

Loss Functions

- We then take partial derivatives with respect to the weights to update the weights:

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

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

Source: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Loss Functions

- If the model predicts correctly, the new weight is updated based on its previous value:

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

- Otherwise, we must include the loss value:

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

Source: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Support Vector Machines

- Kernel functions

Linear:
$K(x, x') = x \cdot x'$

Gaussian Radial Basis Function (RBF):
$K(x, x') = \exp(-\gamma \|x - x'\|^2)$,

$$\text{where } \gamma = \frac{1}{2\sigma^2}$$

Polynomial:
$K(x, x') = (x^T x' + c)^d$

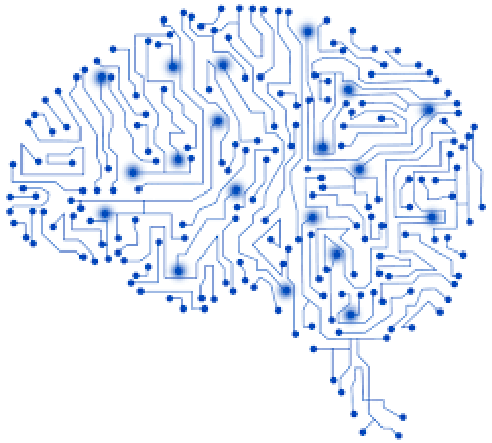
Sigmoidal:
$K(x, x') = \tanh(k x^T x' - \delta)$

References

- Pisner, Derek A., and David M. Schnyer. "Support vector machine." Machine Learning. Academic Press, 2020.
- Cervantes, Jair, et al. "A comprehensive survey on support vector machine classification: Applications, challenges and trends." Neurocomputing, 2020.

University of Minho
School of Engineering

ISLab



Dados e Aprendizagem Automática

Support Vector Machine

DAA @ MEI/1º ano – 1º Semestre