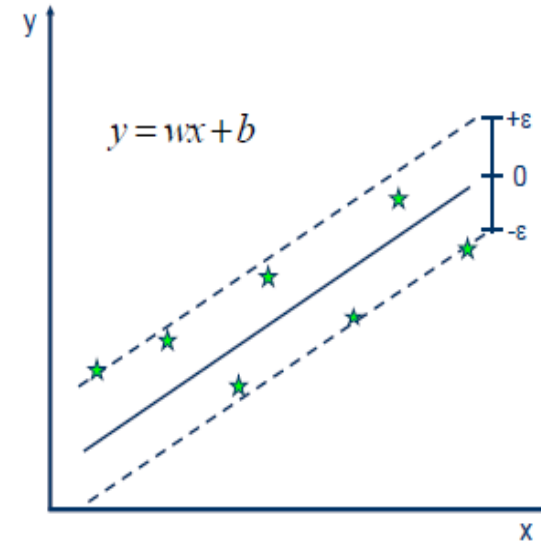
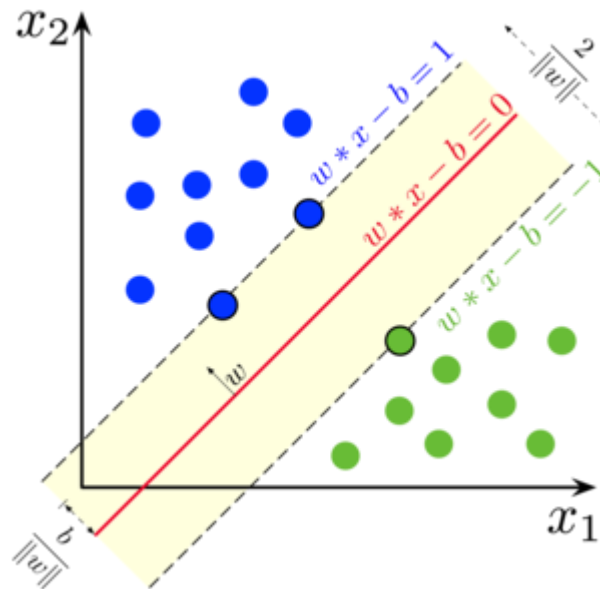


# Class 14 –15 Support Vector Machines

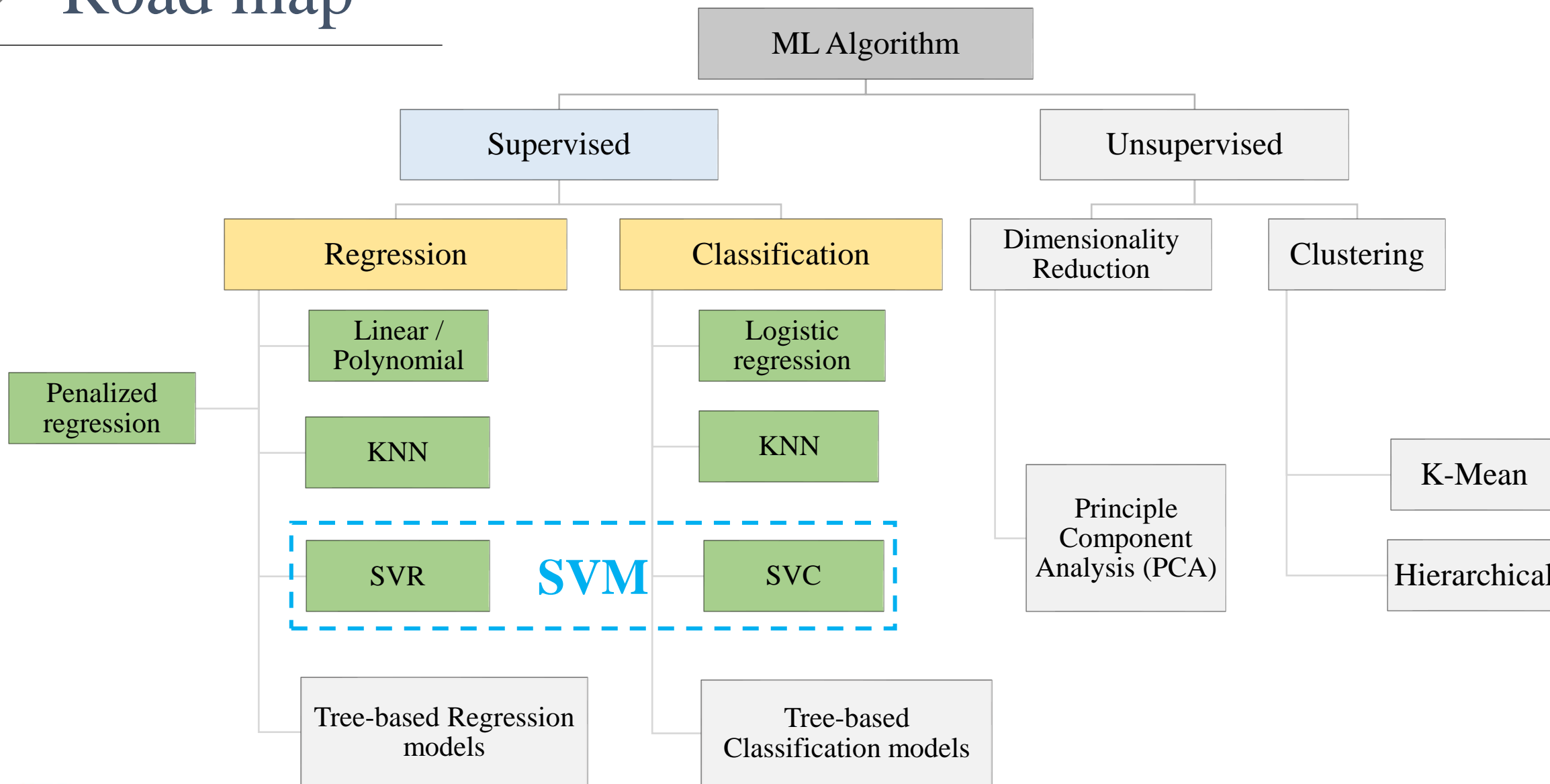
## SVM

Prof. Pedram Jahangiry





# Road map





# Topics

---

## **Part I**

1. SVM Geometry
2. SVM Motivation

## **Part II**

1. Maximum Margin Classifier (MMC)
2. Support Vector Classifiers (SVC)
3. Support Vector Machines (SVM)

## **Part III**

1. Support Vector Regressors (SVR)

## **Part IV**

1. Tuning Hyperparameters
2. SVM pros and cons
3. SVM applications in Finance

# Part I

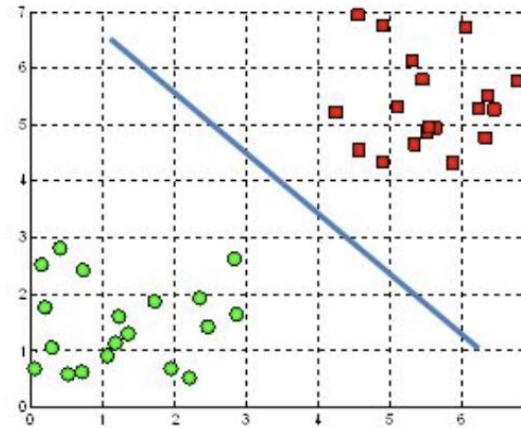
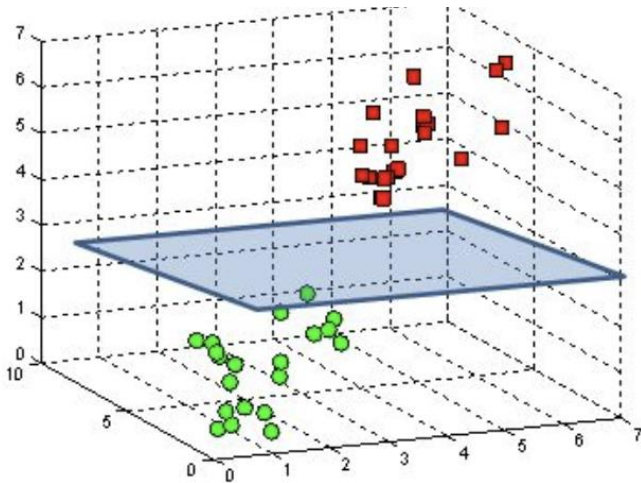
## Geometry

### SVM Motivation



# SVM Geometry

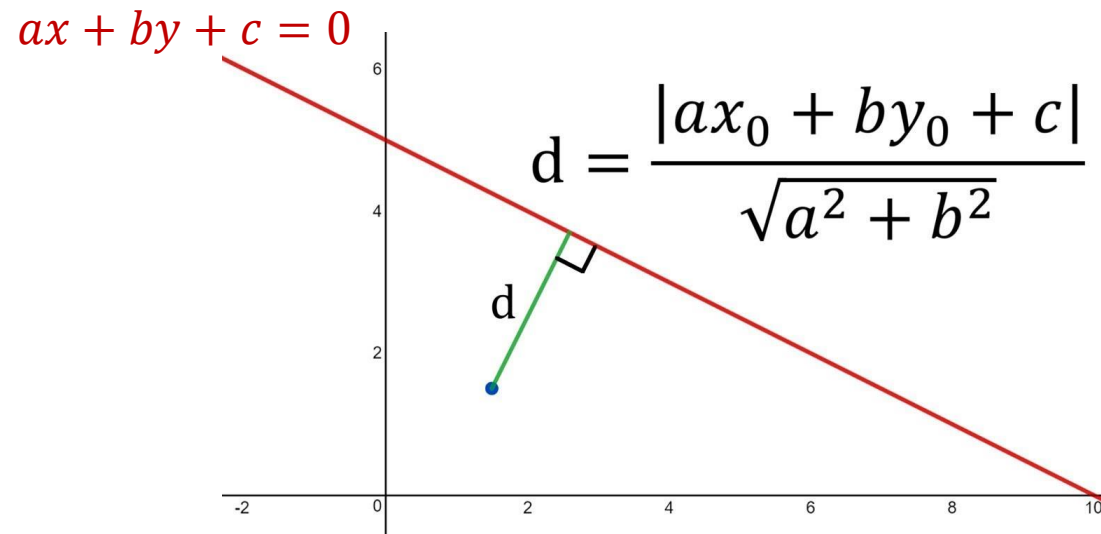
- In geometry, a **hyperplane** is a subspace whose dimension is **one less** than that of its **ambient space**. A hyperplane separates the space into two spaces.
- If a space is 3-dimensional then its hyperplanes are the 2-dimensional **planes**,
- If the space is 2-dimensional, its hyperplanes are the 1-dimensional **lines**.
- If the space is 1-dimensional, its hyperplanes are **single points**.

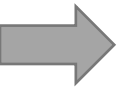




# Geometry

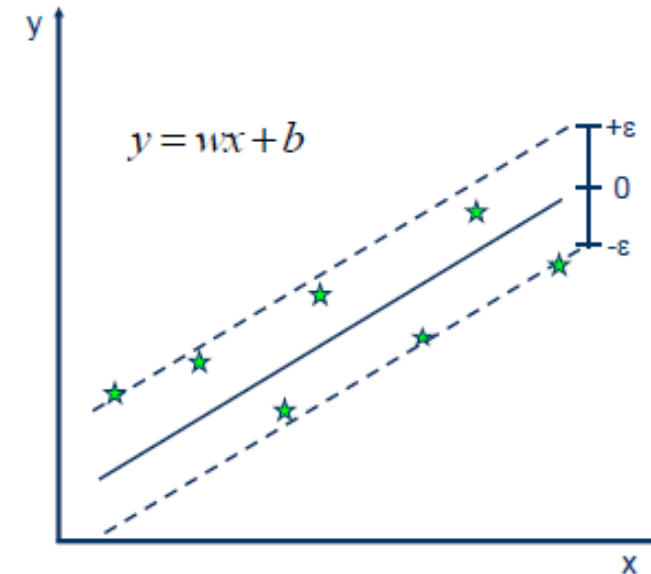
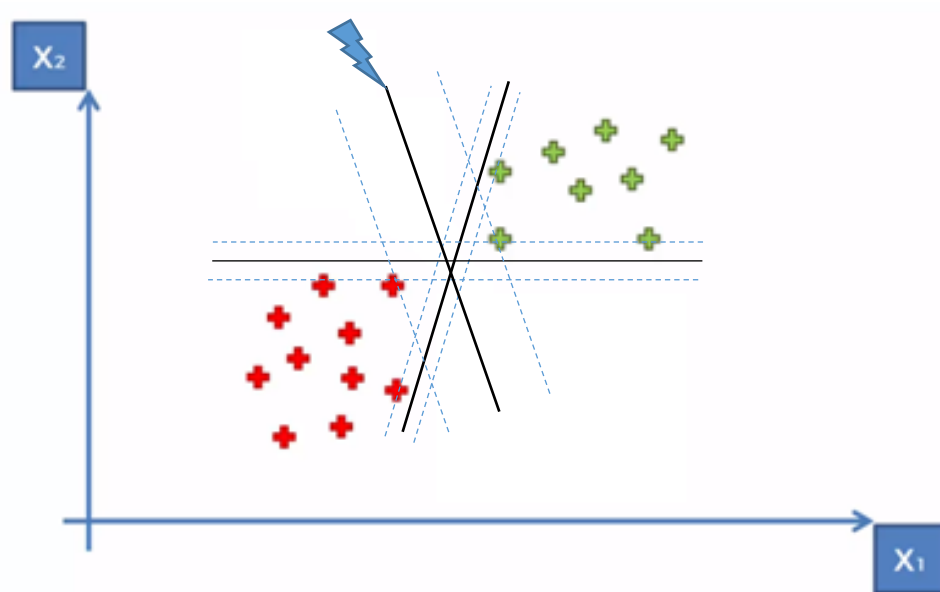
- The **perpendicular distance** between two objects is the distance from one to the other, measured along a line that is perpendicular to one or both.
- The **distance** between a point  $(x_0, y_0)$  and a line parameterized by  $ax + by + c = 0$  is equal to:





# SVM Motivation

**Support vector machine (SVM)** is one of the most popular algorithms in machine learning. It is a powerful supervised algorithm used for **classification** and **regression**.



# Part II

Maximum Margin Classifier (MMC)

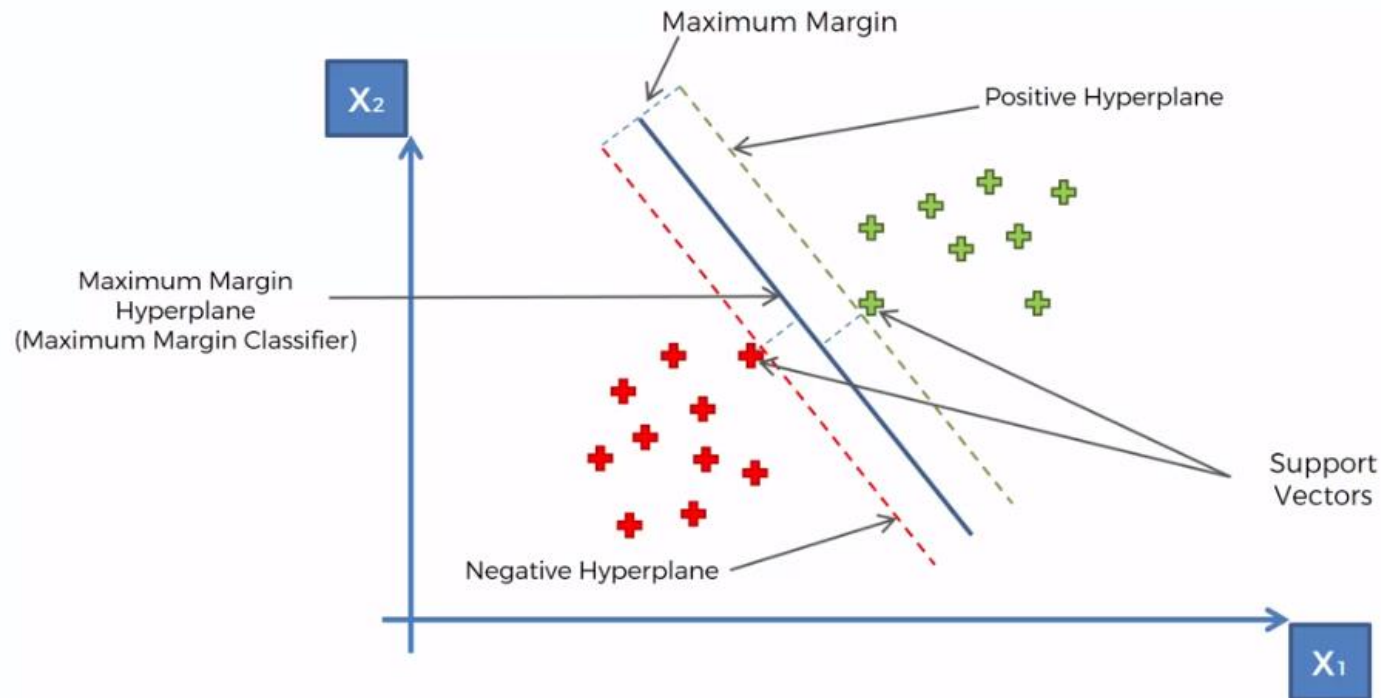
Support Vector Classifiers (SVC)

Support Vector Machines (SVM)



## Maximum Margin Classifier (MMC) – Hard Margin

MMC is the hyperplane that among all separating hyperplanes, find the one that makes the biggest gap (margin) between two classes.

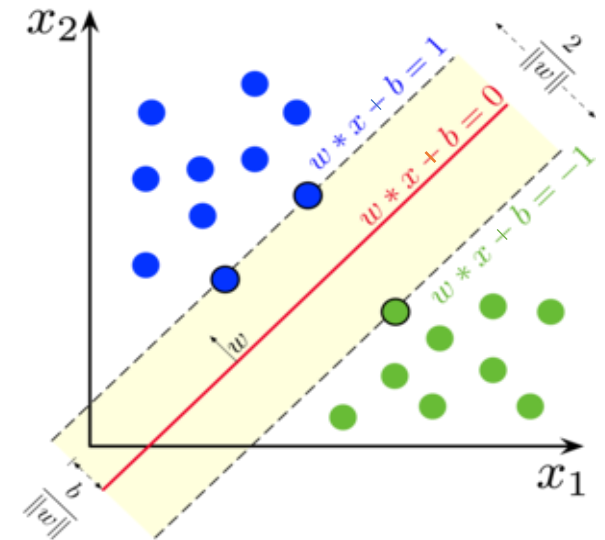


# → MMC optimization problem

- The core idea of **hard margin** is to maximize the margin, under the constraint that the classifier does **not** make **any** mistake.
- SVMs try to pick the most **robust** model (by finding the  $w^*$  and  $b^*$ ) among all those that yield a correct classification. If we numerically define blue circles as +1 and green circles as -1, any **good** linear model is expected to satisfy:

$$\begin{aligned} \text{Min}_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i \left( \sum_{k=1}^K w_k x_{i,k} + b \right) \geq 1 \end{aligned}$$

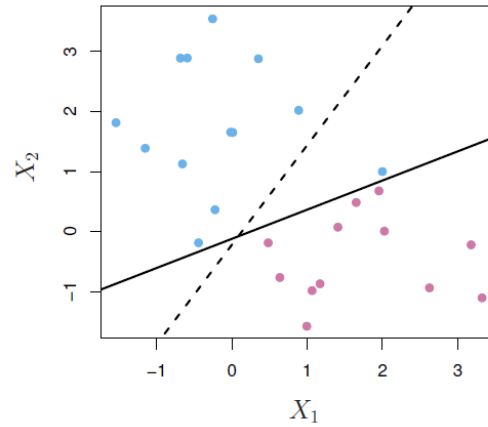
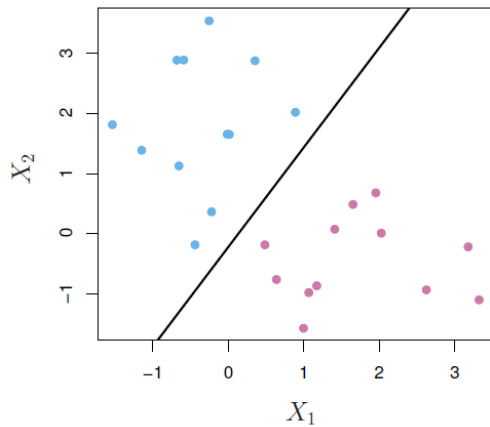
$$\begin{cases} \sum_{k=1}^K w_k x_{i,k} + b \geq +1 & \text{when } y_i = +1 \\ \sum_{k=1}^K w_k x_{i,k} + b \leq -1 & \text{when } y_i = -1 \end{cases}$$



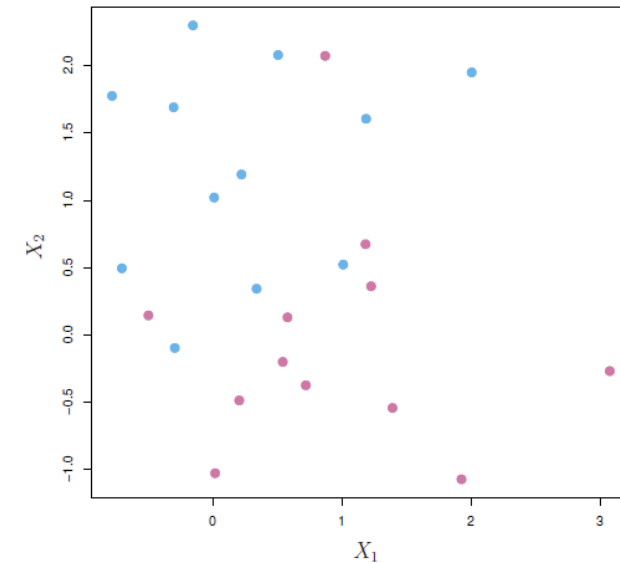
# Support Vector Classifier (SVC) – Soft Margin

- The MMC optimization problem becomes infeasible whenever the condition cannot be satisfied, that is, when **a simple line cannot perfectly separate the labels**, no matter the choice of coefficients.
- This happens when:

1- The data is **noisy**  
MMC is very sensitive to outliers

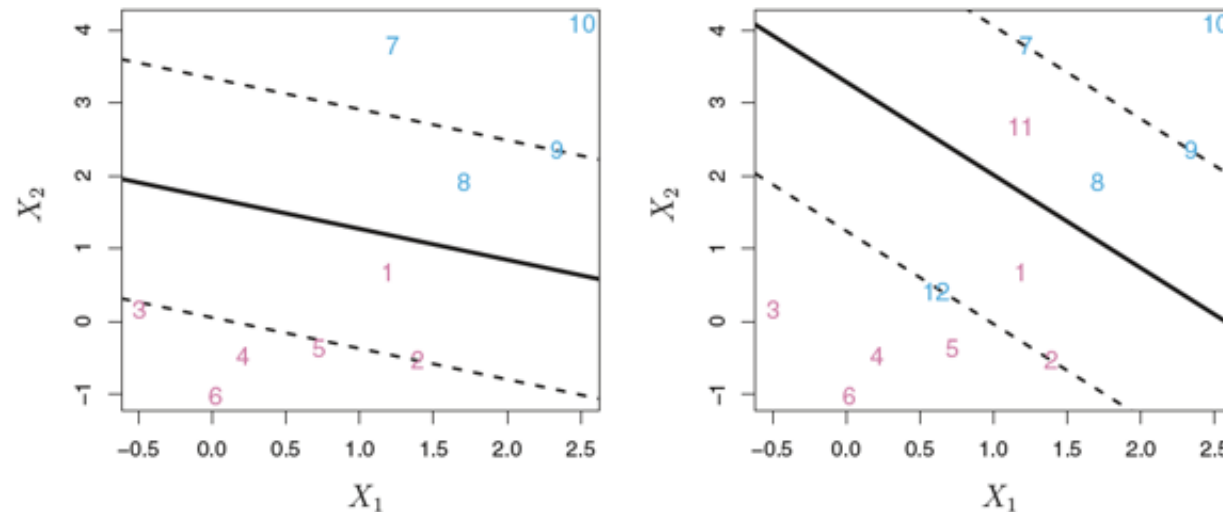


2- The data is **non-separable (overlap)**



# Support Vector Classifier (SVC) – Soft Margin

- **Solution:** we can extend the concept of a separating hyperplane in order to develop a hyperplane that **almost** separates the classes, using a so-called **soft margin**.
- The generalization of the maximal margin classifier using soft margin is known as the **support vector classifier (SVC)**.
- It could be worthwhile to **misclassify a few training observations** in order to do a **better job in classifying the remaining observations**.



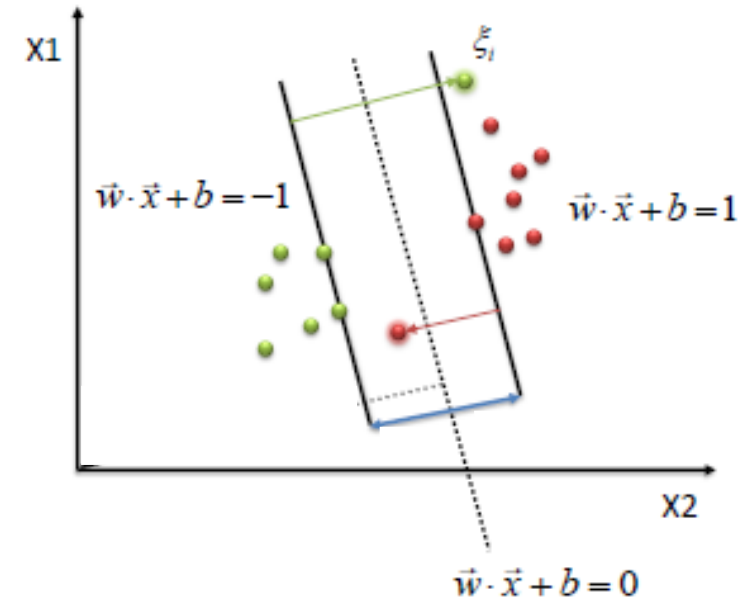
# SVC optimization problem

- Soft margin classification adds a **penalty (C)** to the objective function for observations in the training set that are misclassified. In essence, the SVM algorithm will choose a decision boundary that optimizes the trade-off between a wider margin and a lower total error penalty.
- **Slack variable  $\xi_i$**  allow some observations to fall on the wrong side of the margin, but will penalized them by parameter **C: Cost of misclassification**

$$\text{Min}_{w,b} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^I \xi_i$$

$$\text{s.t. } y_i \left( \sum_{k=1}^K w_k x_{i,k} + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall_i$$

$$\begin{cases} \sum_{k=1}^K w_k x_{i,k} + b \geq +1 - \xi_i & \text{when } y_i = +1 \\ \sum_{k=1}^K w_k x_{i,k} + b \leq -1 + \xi_i & \text{when } y_i = -1 \end{cases}$$



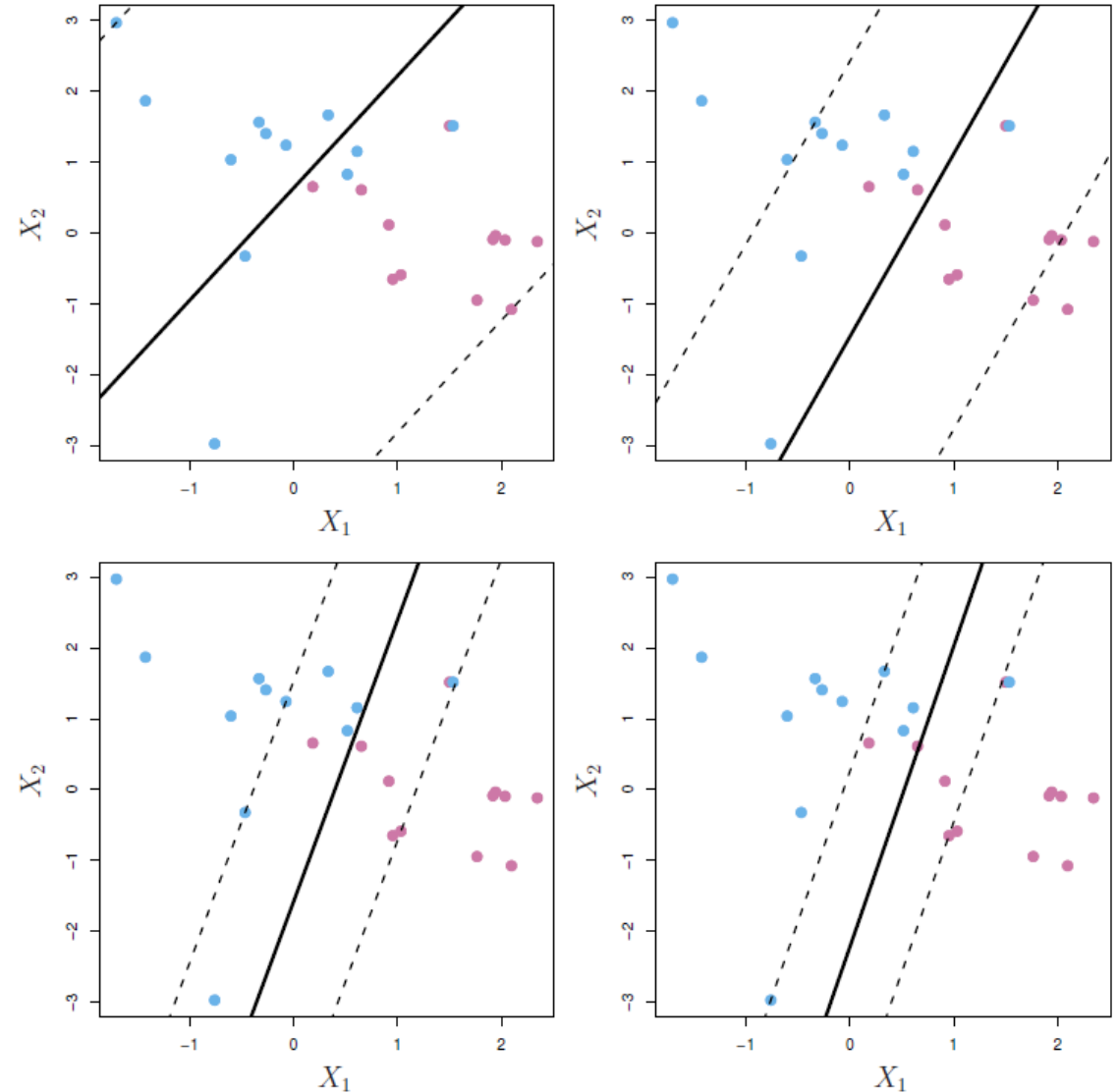
# Regularization parameter

$C$ : Cost of misclassification



Small  $C$  :  
wide margin : high bias : low variance

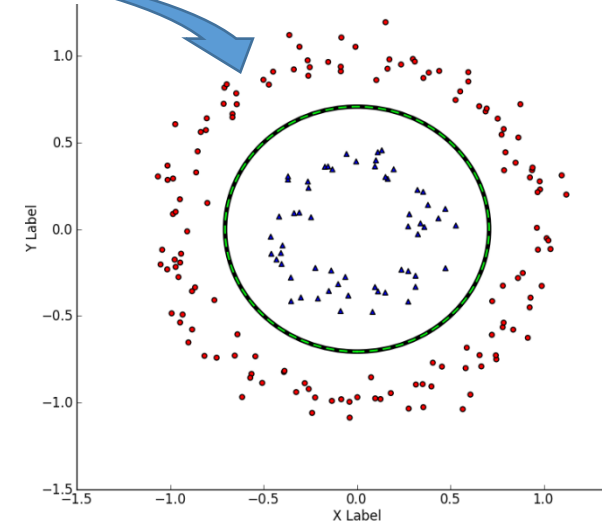
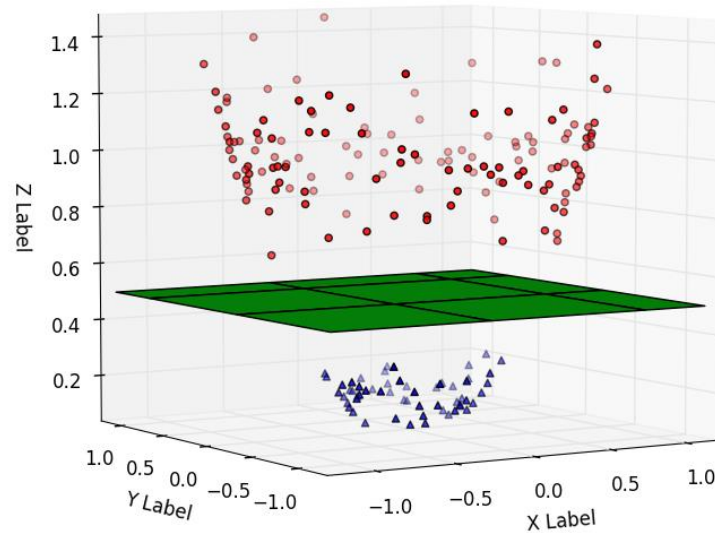
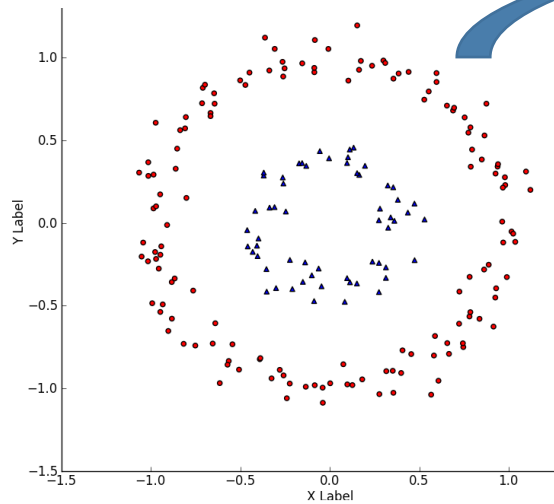
Large  $C$  :  
narrow margin : low bias : High variance





# Kernel Trick!

- **Non-linearly separable** data: sometime a linear boundary simply **won't work**, no matter what value of  $C$ .
- We need a non-linear decision boundary!
- Mapping to higher dimensional space, finding the hyper plane and projecting it back to low dimensional space can be **computationally expensive**.
- Solution: **Kernel Trick!**



# Support Vector Machines (SVM)

- SVM generalizes the SVC to a nonlinear model, via the **kernel  $\phi$**  which is applied to the input points  $x_{i,k}$ .
- The Kernel  **$\phi(x_{i,k})$**  is a function that quantifies the **similarities** between observations by summarizes the relationship between every single pairs in the training set.

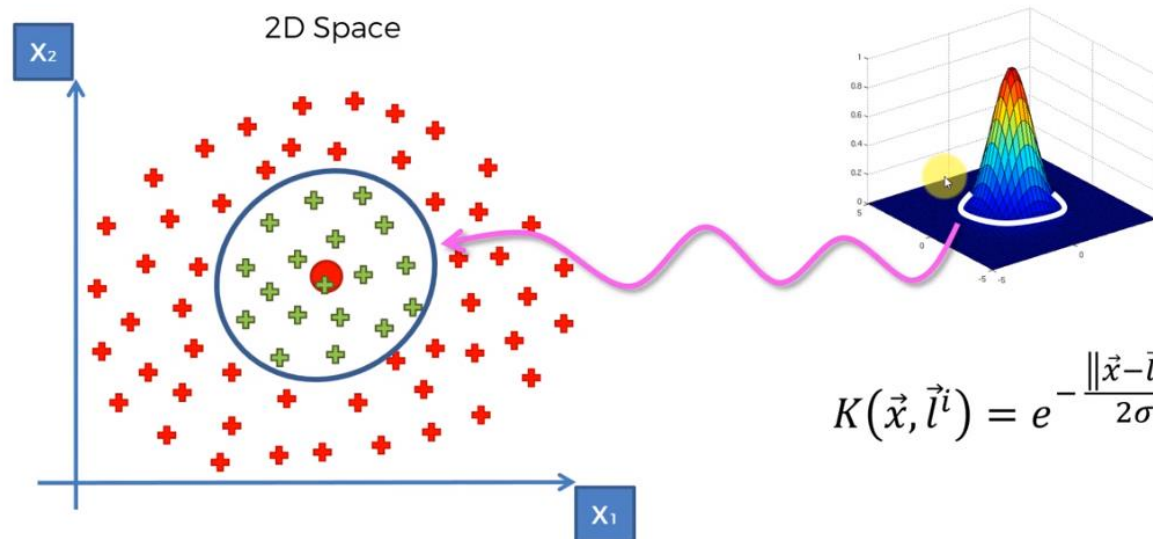
SVC + Non-linear Kernel = SVM

$$\begin{aligned} & \text{Min}_{w,b} \quad \frac{1}{2} ||w||^2 + C \sum_{i=1}^I \xi_i \\ & \text{s. t.} \quad y_i \left( \sum_{k=1}^K w_k \phi(x_{i,k}) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall_i \end{aligned}$$



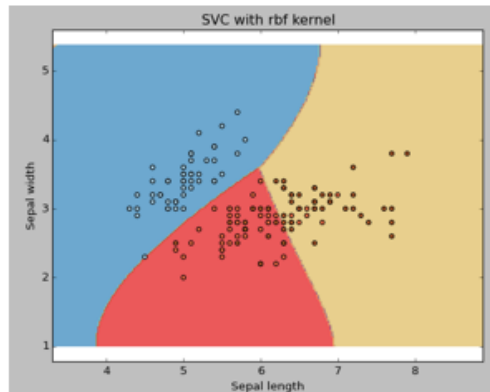


# The Gaussian RBF Kernel (Radial Basis Function)

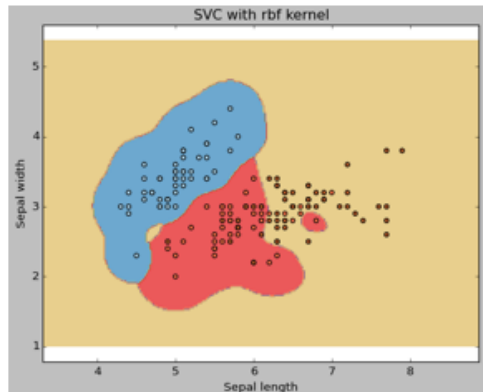


$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x} - \vec{l}^i\|^2}{2\sigma^2}} \quad , \quad \gamma = \frac{1}{2\sigma^2}$$

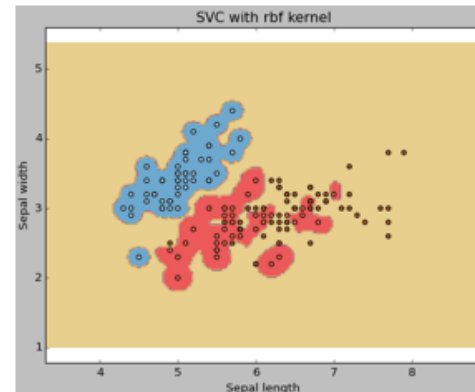
gamma = 0



gamma = 10

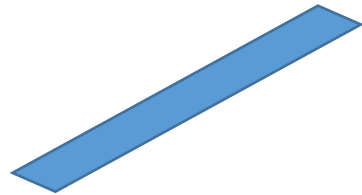


gamma = 100

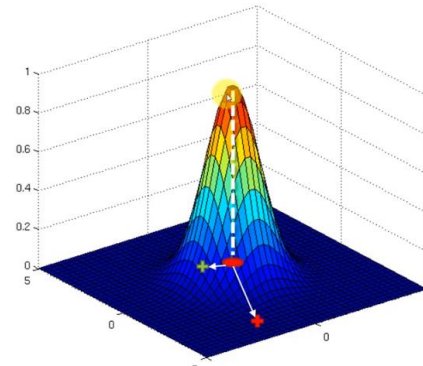


# ➔ Most common types of Kernel

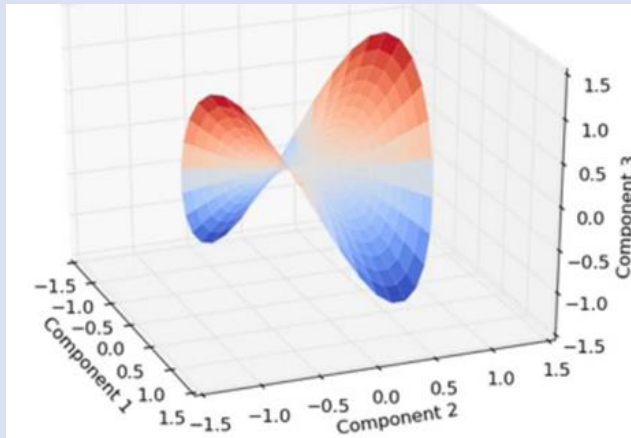
Linear Kernel



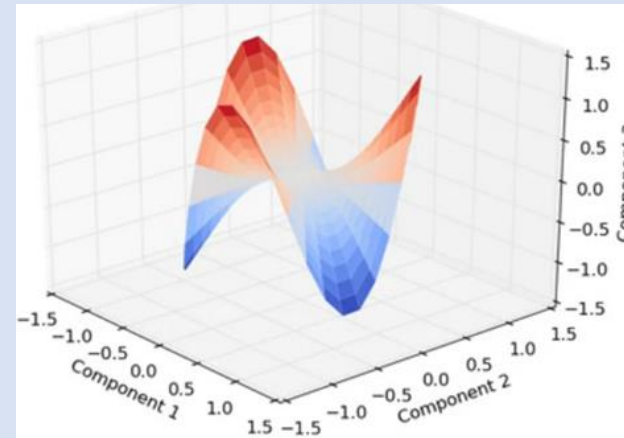
The Gaussian RBF Kernel



Polynomial Kernel

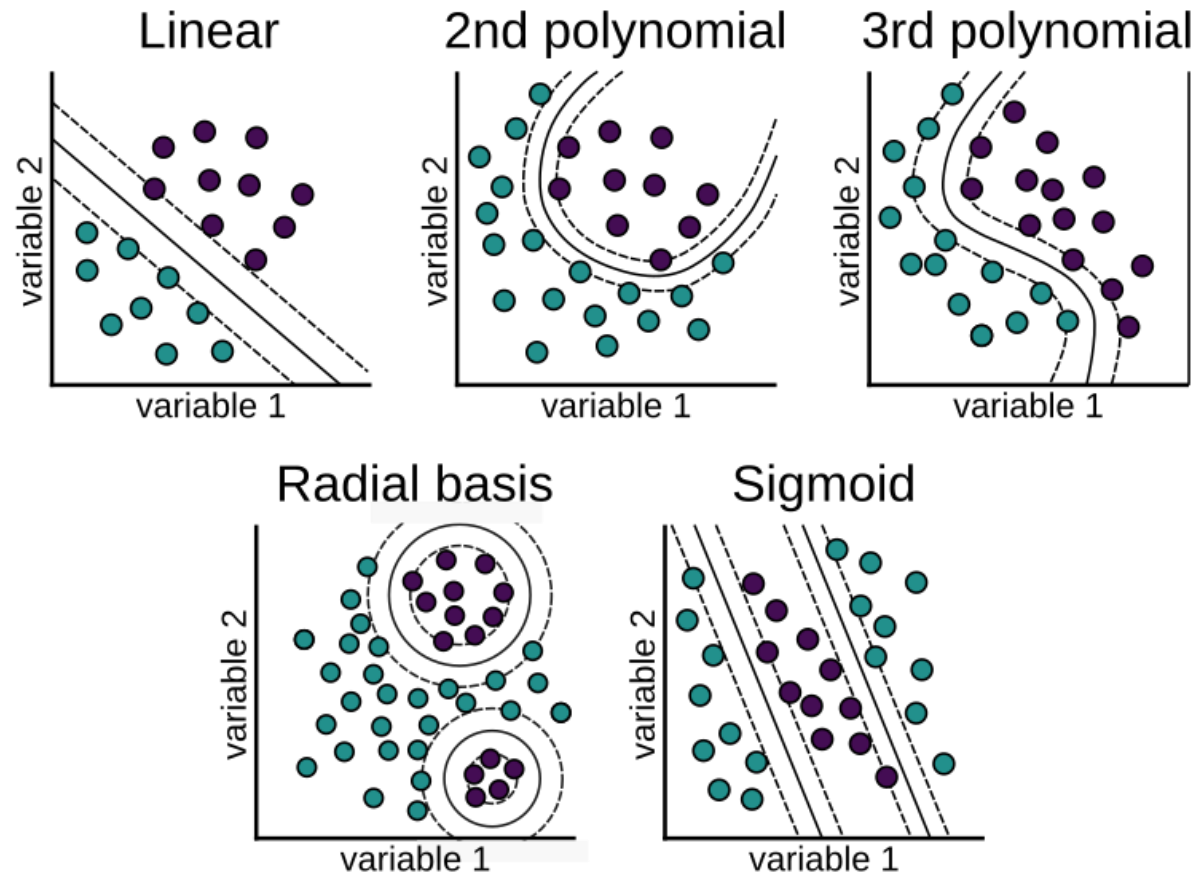


Sigmoid Kernel





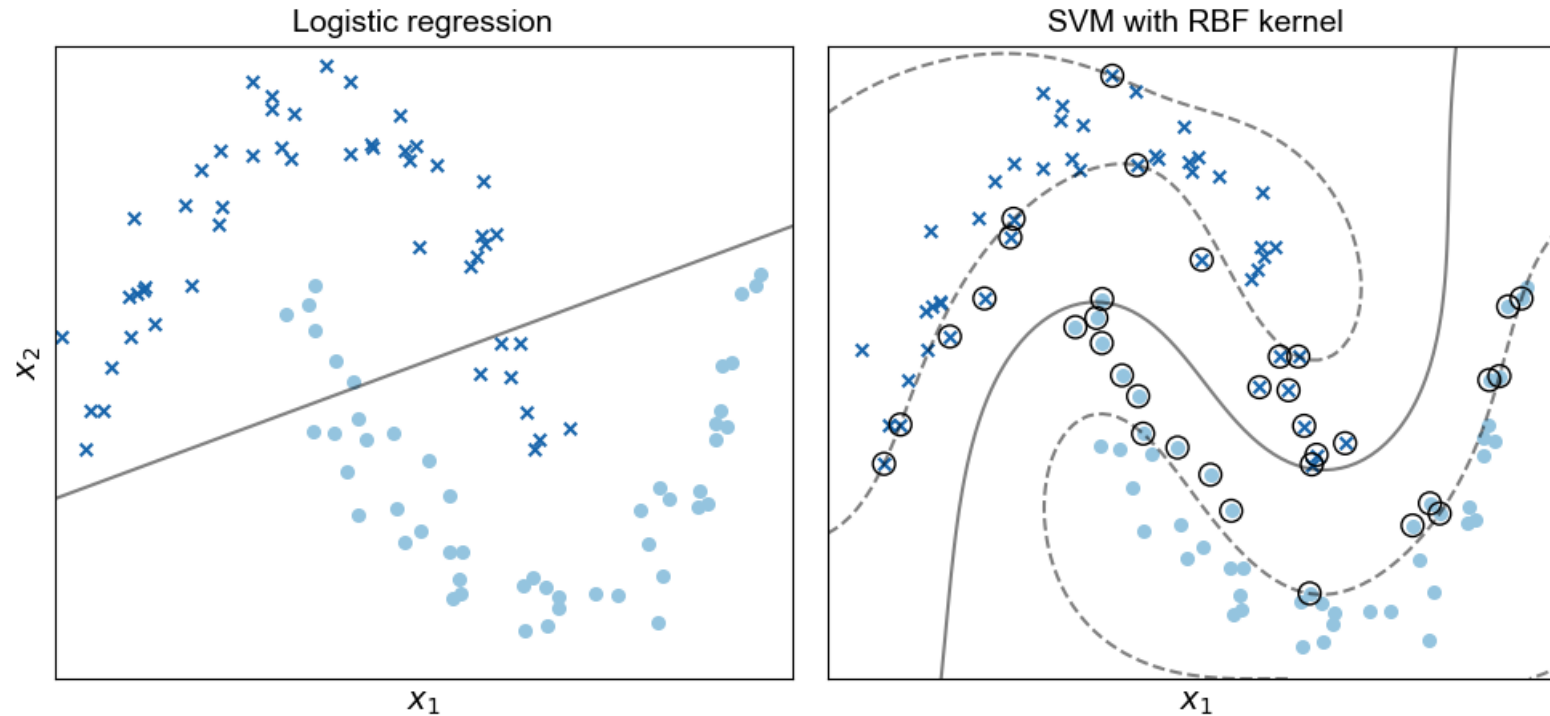
# Decision boundaries with different Kernels



Source: [Machine learning with R, tidyverse, and mlr](#)



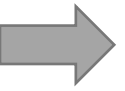
# Comparing Logistic Regression and SVM



Source: <http://gregorygundersen.com/blog/2019/12/23/random-fourier-features/>

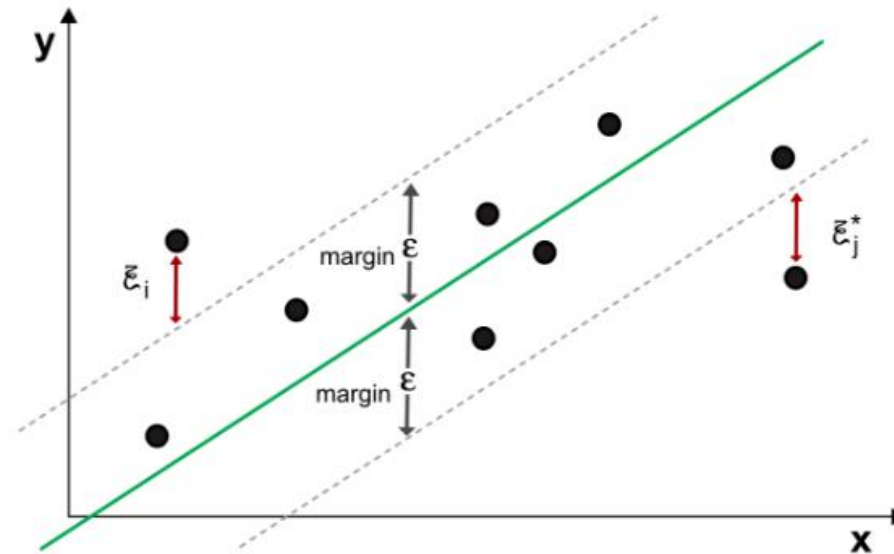
# Part III

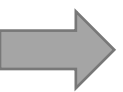
## Support Vector Regressors (SVR)



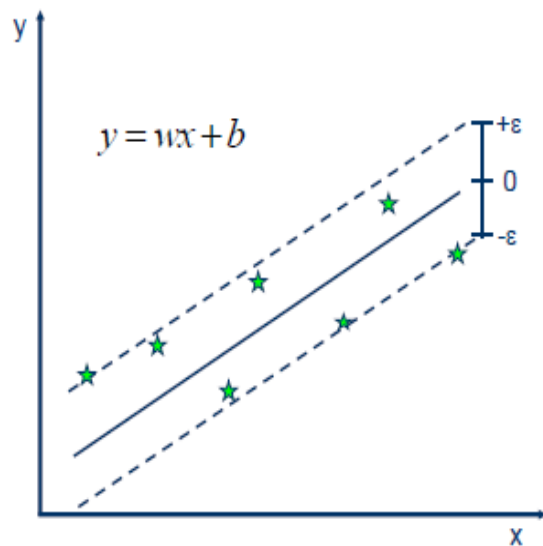
# SVM for regression (Support Vector Regressors)

- The idea of SVM classification can be transposed to regression problems.
- However, the **role of the margin** is different. Our objective, is to basically find the hyperplane that holds maximum training observations within the margin  $\epsilon$  (tolerance level).

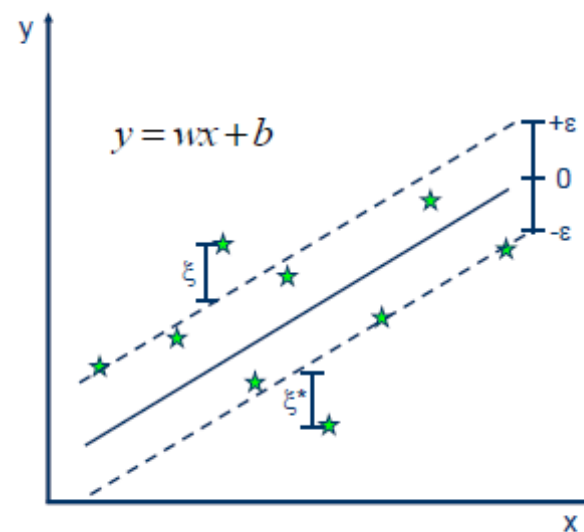




# SVR optimization



- Minimize:
$$\min \frac{1}{2} \|w\|^2$$
- Constraints:
$$y_i - wx_i - b \leq \varepsilon$$
$$wx_i + b - y_i \leq \varepsilon$$

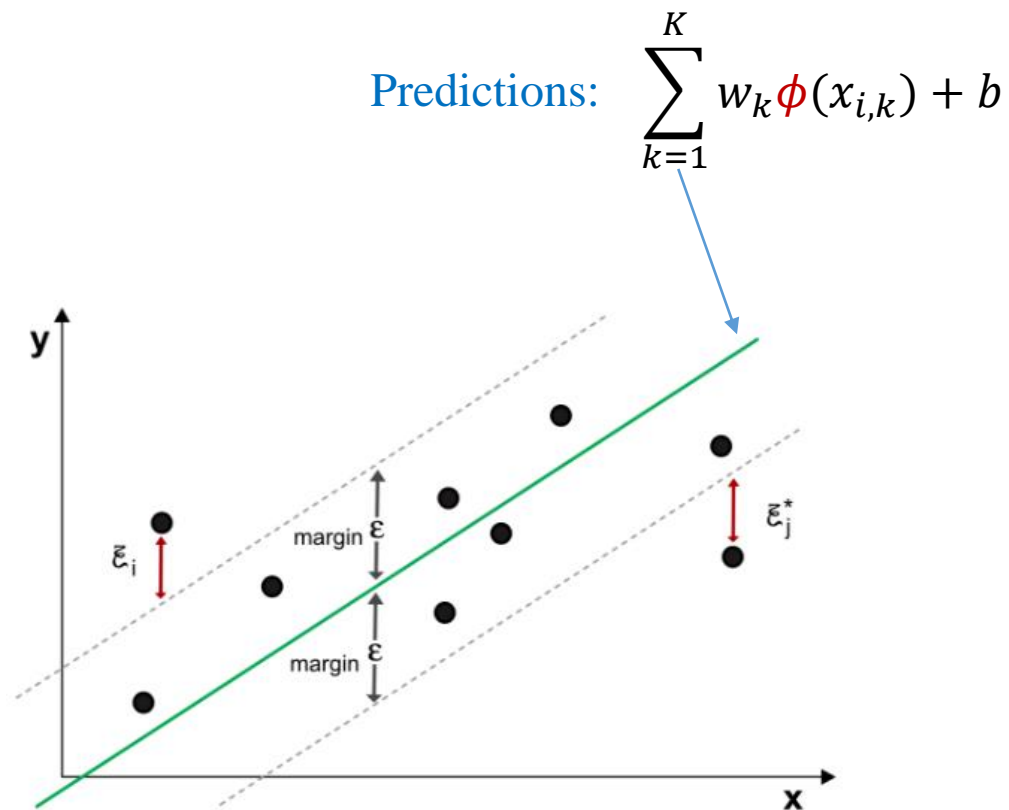


- Minimize:
$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$
- Constraints:
$$y_i - wx_i - b \leq \varepsilon + \xi_i$$
$$wx_i + b - y_i \leq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

Source: [https://www.saedsayad.com/support\\_vector\\_machine\\_reg.htm](https://www.saedsayad.com/support_vector_machine_reg.htm)

# Kernel SVR optimization

$$\begin{aligned} \text{Min}_{w,b} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^I (\xi_i + \xi_i^*) \\ \left( \sum_{k=1}^K w_k \phi(x_{i,k}) + b \right) - y_i & \leq \epsilon + \xi_i^* \\ y_i - \left( \sum_{k=1}^K w_k \phi(x_{i,k}) + b \right) & \leq \epsilon + \xi_i \\ \xi_i, \xi_i^* & \geq 0 \quad \forall_i \end{aligned}$$

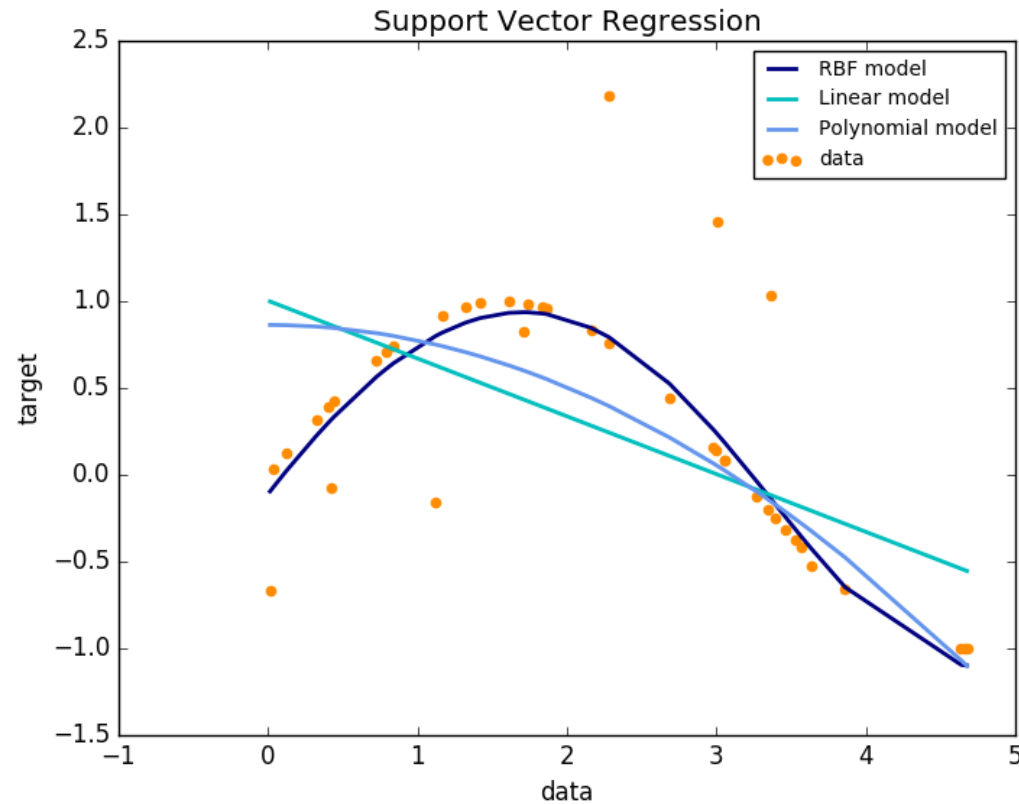


- Goal: minimize the sum of squared **weights** subject to the **error being small** enough
- This is somewhat the opposite of the penalized linear regressions which seek to minimize the error, subject to the weights being small enough





# SVR using Linear and Non-linear Kernels



Source: Scikit learn documentation

# Part IV

## Tuning hyperparameters

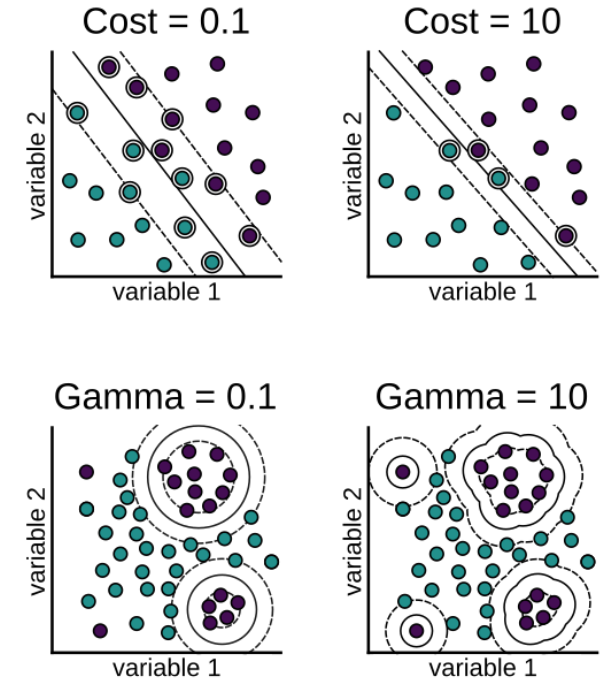
## SVM pros and cons

## SVM applications in Finance

# → Tuning hyperparameters

SVM hyperparameters:

- 1) **C**, Cost of misclassification: controls bias variance trade off
- 2) **Kernel**
- 3) **Gamma**, controls how far the influence of a single training set reaches



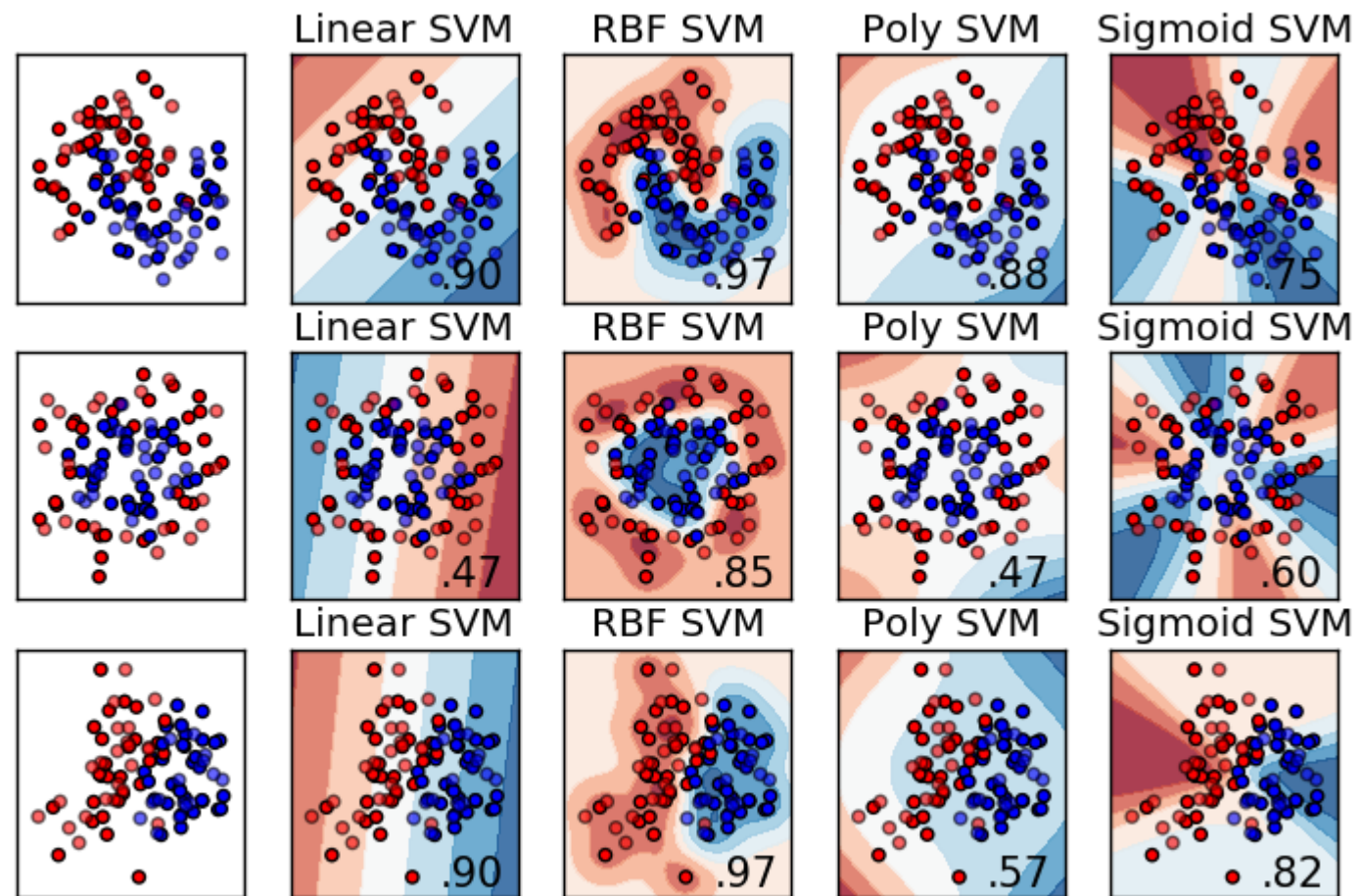
Source: [Machine learning with R, tidyverse, and mlr](#)

Grid search cross validation is used to tune the hyper parameters.

Kernel	C	Gamma	CV
Linear, rbf, poly, ...	0.1, 1, 10, 100, ...	0.001, 0.01, 0.1, 1, ...	5,10,...

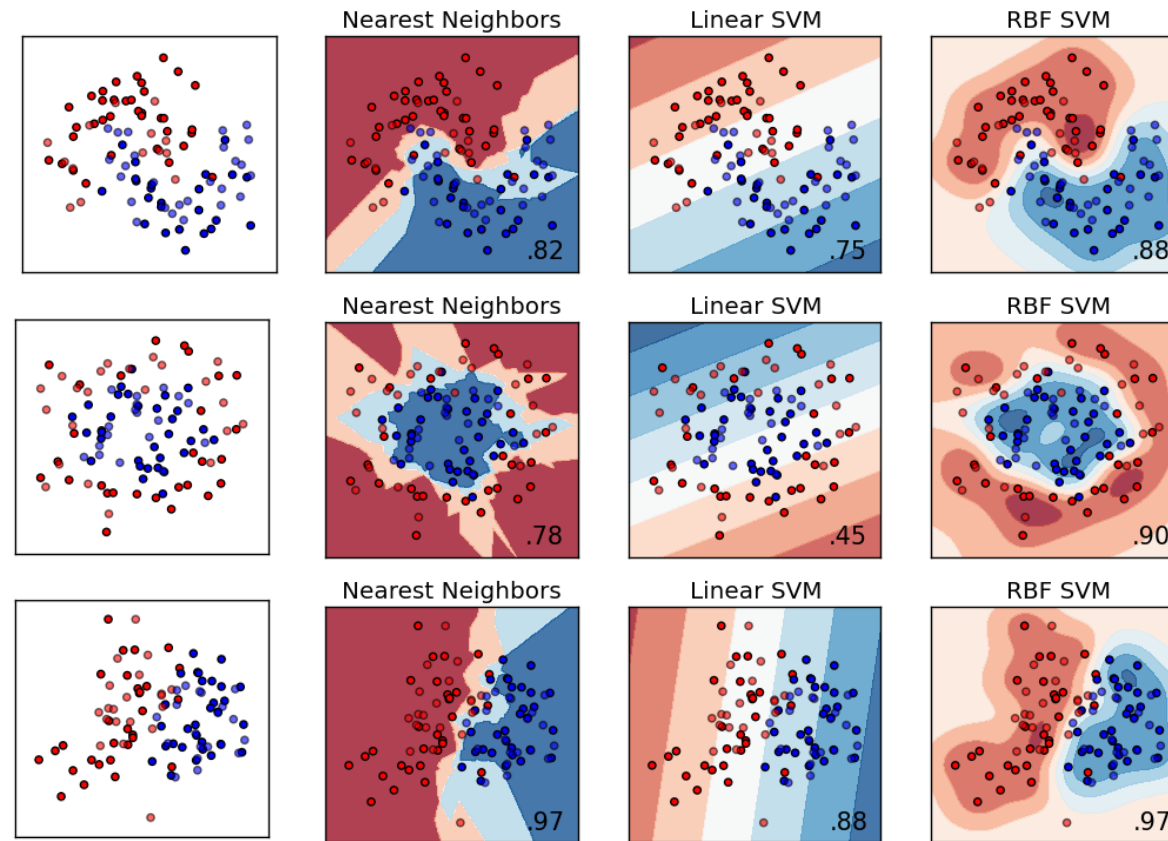


# Decision boundaries with different Kernels



Source <https://www.kaggle.com/residentmario/kernels-and-support-vector-machine-regularization>

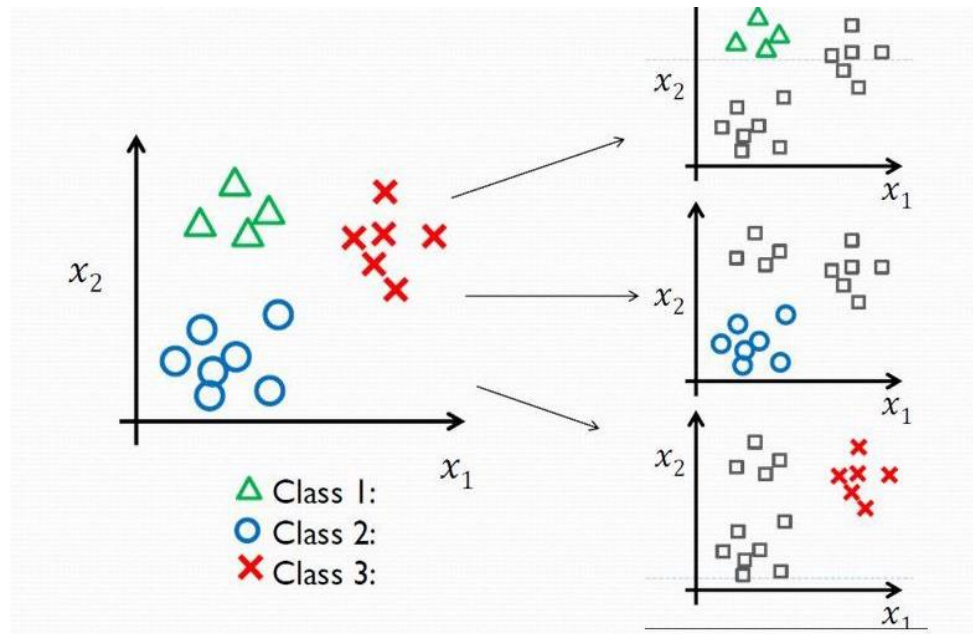
# → Comparing classifiers (so far)



Source: [https://scikit-learn.org/stable/auto\\_examples/classification/plot\\_classifier\\_comparison.html](https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html)

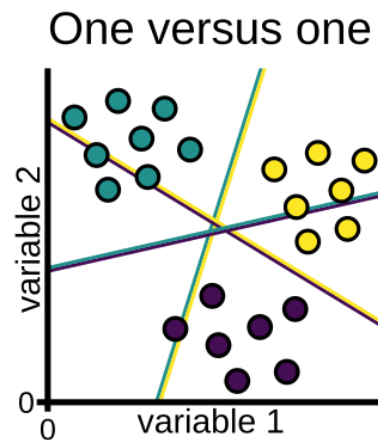
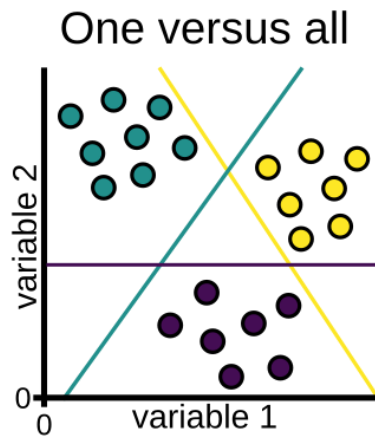
# ➔ K-Multiple class SVM

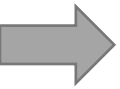
- One-VS-All (OVA)
  1. Fit K different 2-class SVM classifiers  $\hat{f}_k(x)$ , each class versus the rest
  2. Classify  $x_{te}$  to the class for which  $\hat{f}_k(x_{te})$  is largest.



# ➔ K-Multiple class SVM

- One-VS-One (OVO)
- 1. Fit all  $\binom{K}{2}$  pairwise classifiers  $\widehat{f}_{kl}(x)$ , each class versus the rest
- 2. Classify  $x_{te}$  to the class that wins the most pairwise competitions.





# SVM's Pros and Cons

## Pros:

- SVM can be memory efficient! uses only a subset of the training data (support vectors)
- Can handle non-linear data sets
- Can handle high dimensional spaces (even when  $D > N$ )
- Used both for classification and regression
- SVM are not very sensitive to overfitting (soft margin; regularization)
- Can have high accuracy (even compared to NN)

## Cons:

- No probability outcome!
- Long training time when we have large data sets.
- Limited interpretability (specially for Kernel SVM)
- Does not perform well with noisy data
- Suited for small to medium size data





# → SVM's Applications in finance

- Corporate financial statements and bankruptcy (high dimensional)
- Identifying stressed companies to short sell (using many fundamental and technical features)
- Sentiment analysis (classify text from documents e.g., news articles, company announcements, and company annual reports into useful categories for investors)
- Money laundering analysis and spam detection
- Loan management



# Appendix



# MMC solution

---

$$\operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i \left( \sum_{k=1}^K w_k x_{i,k} + b \right) \geq 1$$

$$L(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^I \lambda_i \left( y_i \left( \sum_{k=1}^K w_k x_{i,k} + b \right) - 1 \right)$$

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\lambda}) = \mathbf{0}, \quad \frac{\partial L}{\partial b} L(\mathbf{w}, b, \boldsymbol{\lambda}) = 0,$$

$$\mathbf{w}^* = \sum_{i=1}^I \lambda_i u_i \mathbf{x}_i.$$

# Students' questions

---

- 1) What does non-separable data mean?
- 2) Does the Kernel function act the same way as standardization conceptually, whereby the data is scaled up, but the distribution of data does not change?
- 3) When do we use MMC vs SVM?
- 4) Even after using the kernel trick, does SVM perform faster than KNN on average?
- 5) When using grid search to optimize hyper-parameters in your SVM, are there ways to be more efficient and apply theory similar to gradient descent?