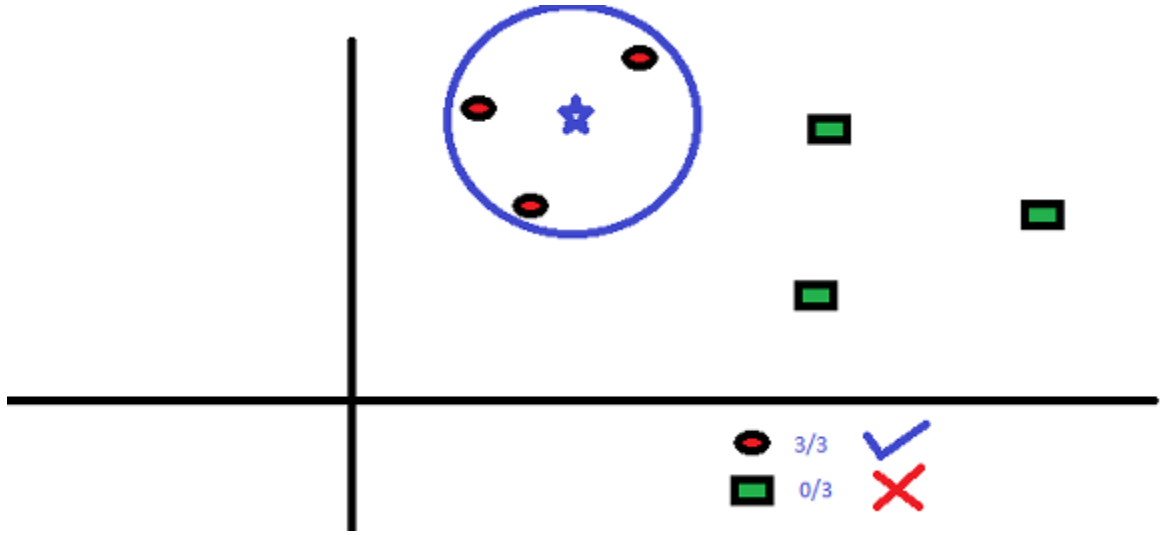
Exercise Session 3

Introduction to KNN & SVM

KNN

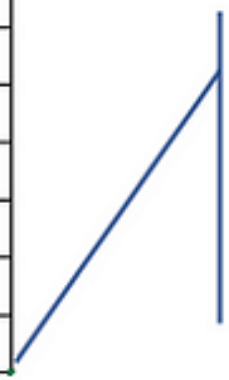
- Simple and intuitive
  1. Calculate the distance between the test data point and the training data points
  2. Sort the distance
  3. Get the nearest training data points
  4. Let the majority vote for the class



KNN

- Let’s take a small example

Customer	Age	Loan	Default
John	25	40000	N
Smith	35	60000	N
Alex	45	80000	N
Jade	20	20000	N
Kate	35	120000	N
Mark	52	18000	N
Anil	23	95000	Y
Pat	40	62000	Y
George	60	100000	Y
Jim	48	220000	Y
Jack	33	150000	Y
Andrew	48	142000	?



We need to predict Andrew default status by using Euclidean distance

KNN

- Let's take a small example

Customer	Age	Loan	Default	Euclidean distance
John	25	40000	N	1,02,000.00
Smith	35	60000	N	82,000.00
Alex	45	80000	N	62,000.00
Jade	20	20000	N	1,22,000.00
Kate	35	120000	N	22,000.00
Mark	52	18000	N	1,24,000.00
Anil	23	95000	Y	47,000.01
Pat	40	62000	Y	80,000.00
George	60	100000	Y	42,000.00
Jim	48	220000	Y	78,000.00
Jack	33	150000	Y	8,000.01
Andrew	**48**	**142000**	**?**	

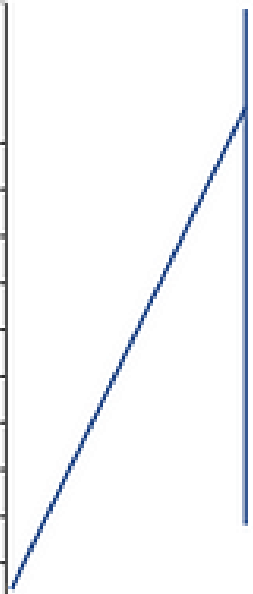
First Step calculate the Euclidean distance dist(d) = Sq.rt $(x_1-y_1)^2 + (x_2-y_2)^2$
= Sq.rt$(48-25)^2 + (142000 - 40000)^2$
dist (d_1) = 1,02,000.

We need to calcuate the distance for all the datapoints

KNN

- Let’s take a small example

Customer	Age	Loan	Default	Euclidean distance	Minimum Euclidean Distance
John	25	40000	N	1,02,000.00	
Smith	35	60000	N	82,000.00	
Alex	**45**	**80000**	**N**	**62,000.00**	**5**
Jade	20	20000	N	1,22,000.00	
Kate	**35**	**120000**	**N**	**22,000.00**	**2**
Mark	52	18000	N	1,24,000.00	
Anil	**23**	**95000**	**Y**	**47,000.01**	**4**
Pat	40	62000	Y	80,000.00	
George	**60**	**100000**	**Y**	**42,000.00**	**3**
Jim	48	220000	Y	78,000.00	
Jack	**33**	**150000**	**Y**	**8,000.01**	**1**
Andrew	48	142000	?		



Let assume K = 5

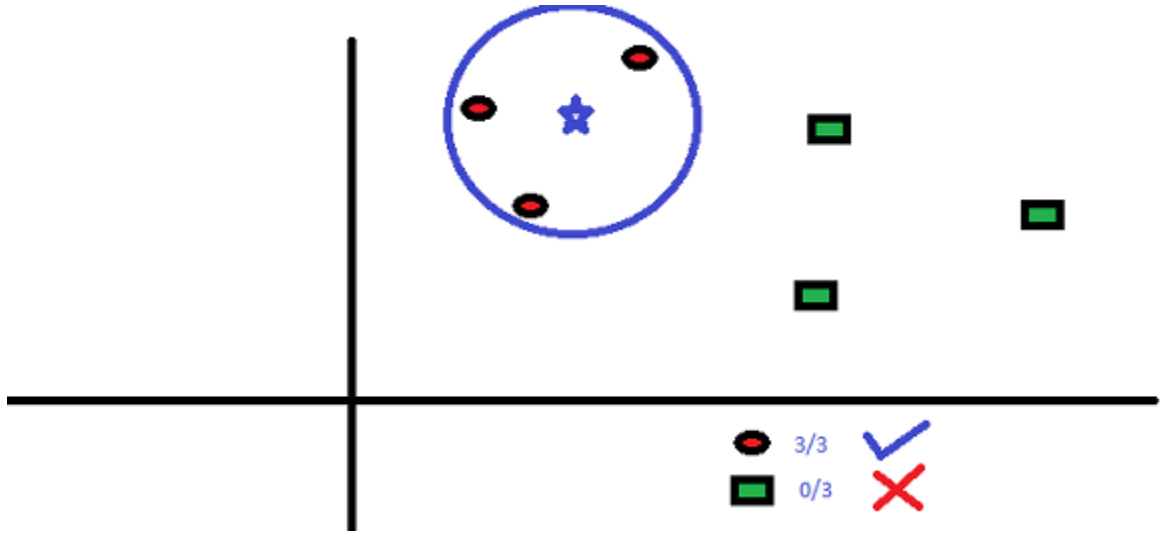
Find minimum euclidean distance and rank in order (ascending)

In this case, 5 minimum euclidean distance. With k=5, there are two Default = N and three Default = Y out of five closest neighbors.

We can say Andrew default stauts is 'Y' (Yes)

KNN

- What you may want to decide: How to choose k?
  - Not easy!
  - Smaller value: noisy result
  - Larger value: increased bias & slower computation
  - In practice, start with k = sqrt(N)



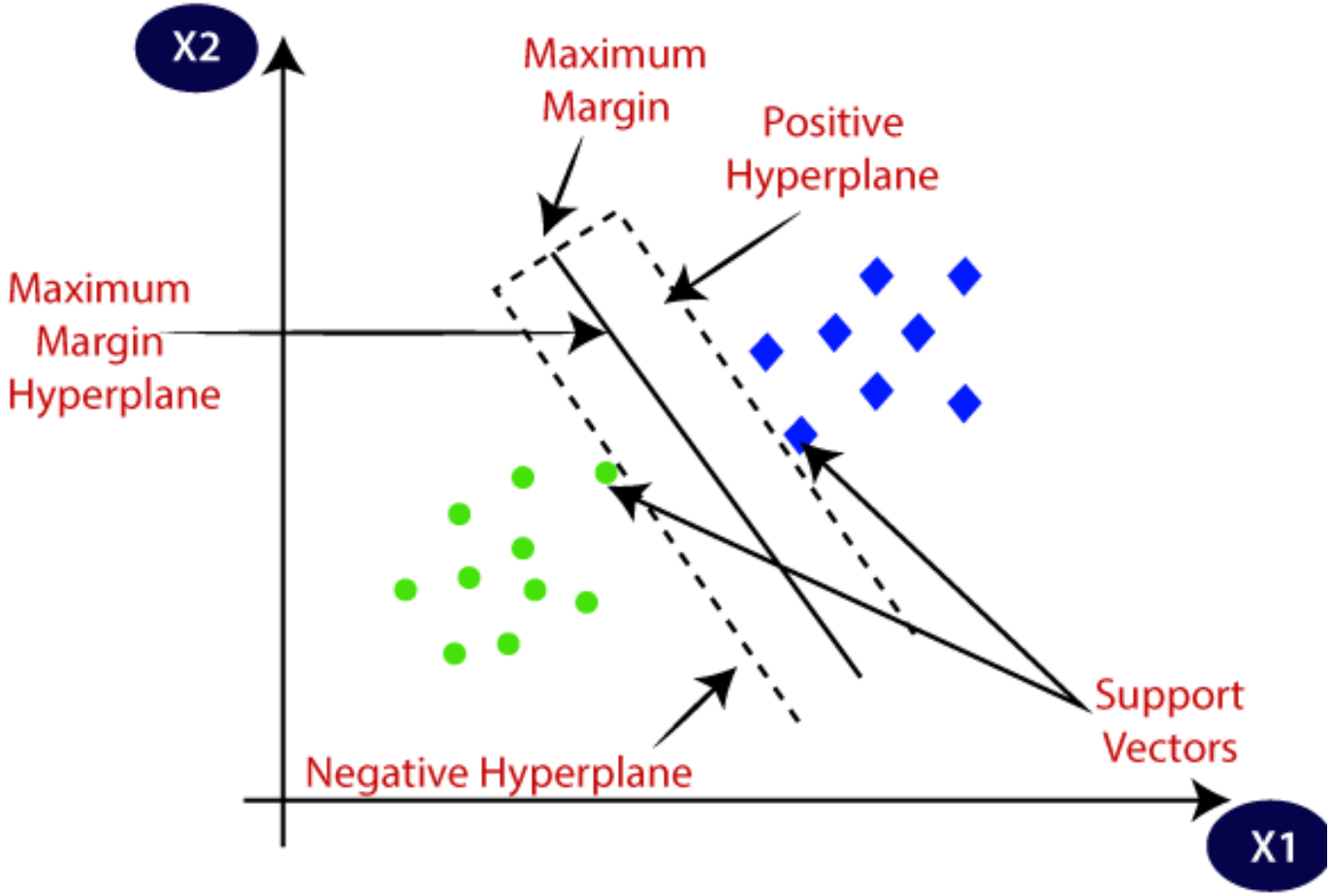
KNN

- Pros
  - Simple to implement
  - Flexible to feature/distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data

- Cons
  - Need to determine the value of parameter K (number of nearest neighbors)
  - Computation cost is quite high because we need to compute the distance of each query instance to all training samples.
  - Storage of data
  - Must know we have a meaningful distance function.

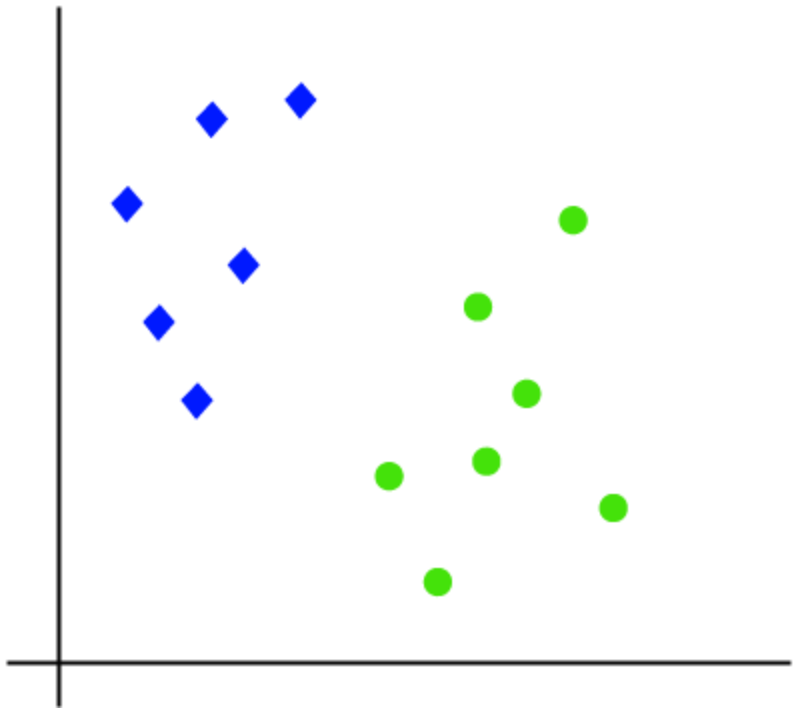
SVM

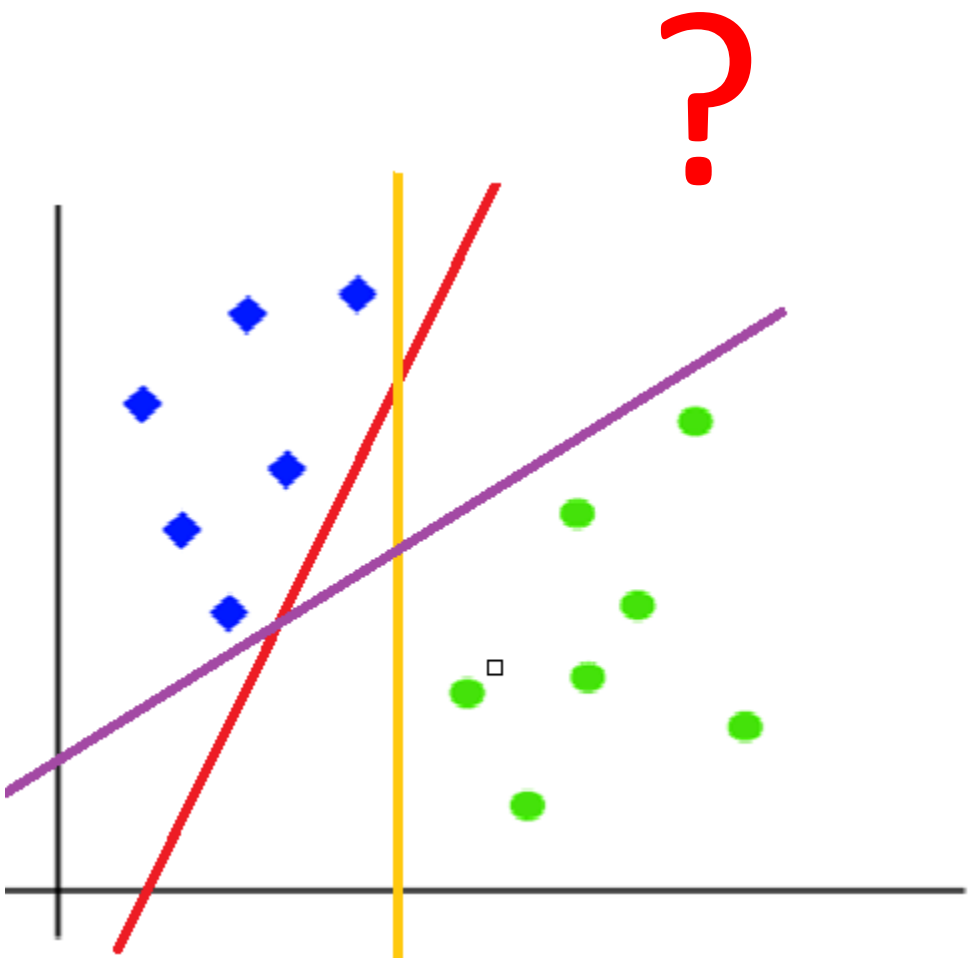
- Linear classification/regression
  - Support vectors
  - Margin



SVM

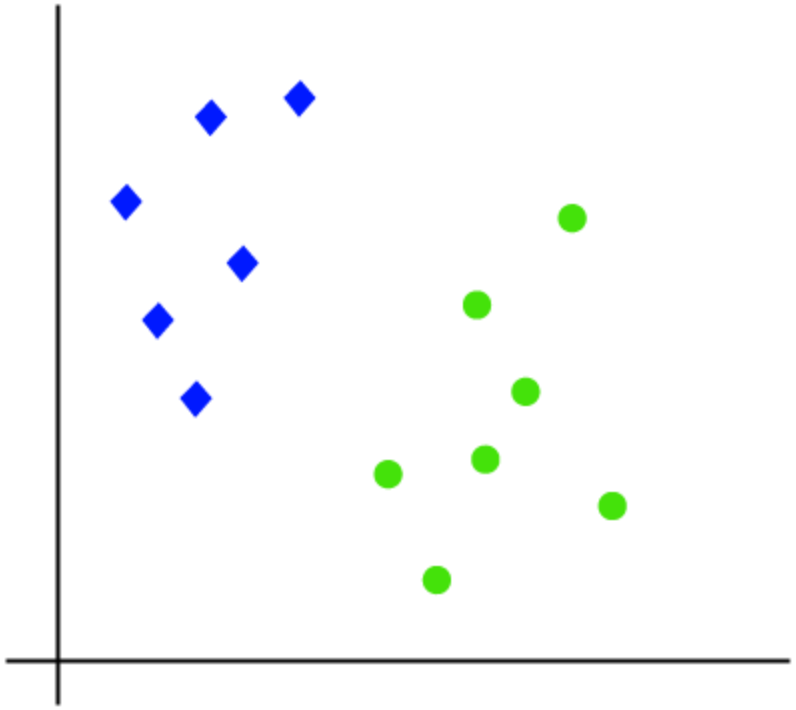
- Which one is the optimal hyperline/hyperplane?

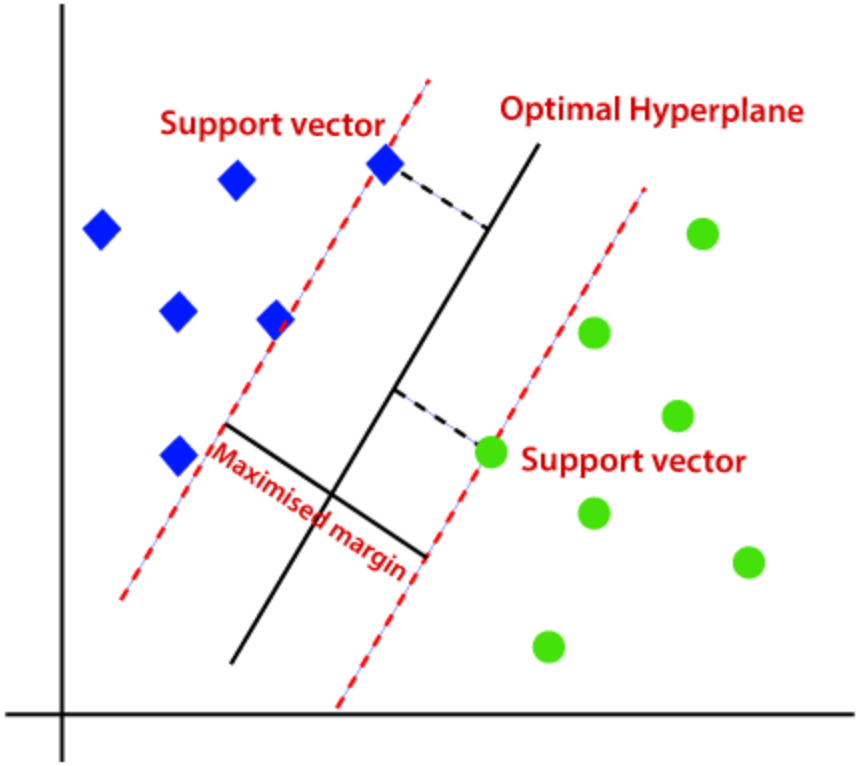




SVM

- Which one is the optimal hyperline/hyperplane?
  - Maximizing the distance of classes



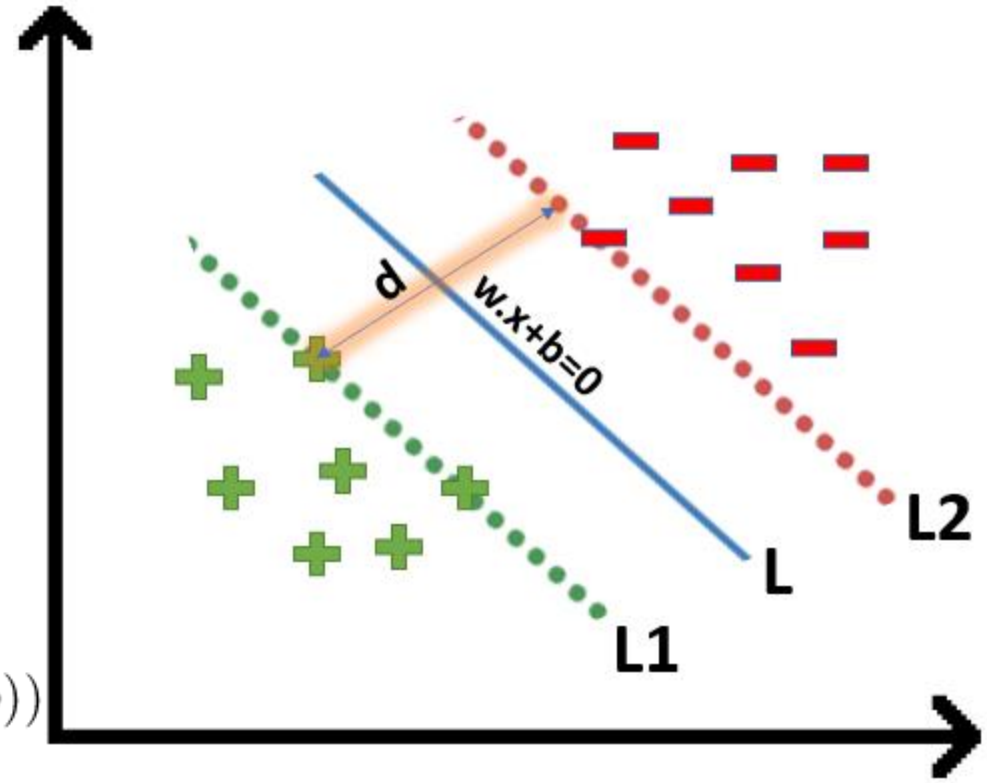


SVM

- How do we find the maximum hyperplane
  - L ： **wx + b = 0**
  - L1： **wx + b = 1**
  - L2： **wx + b = -1**

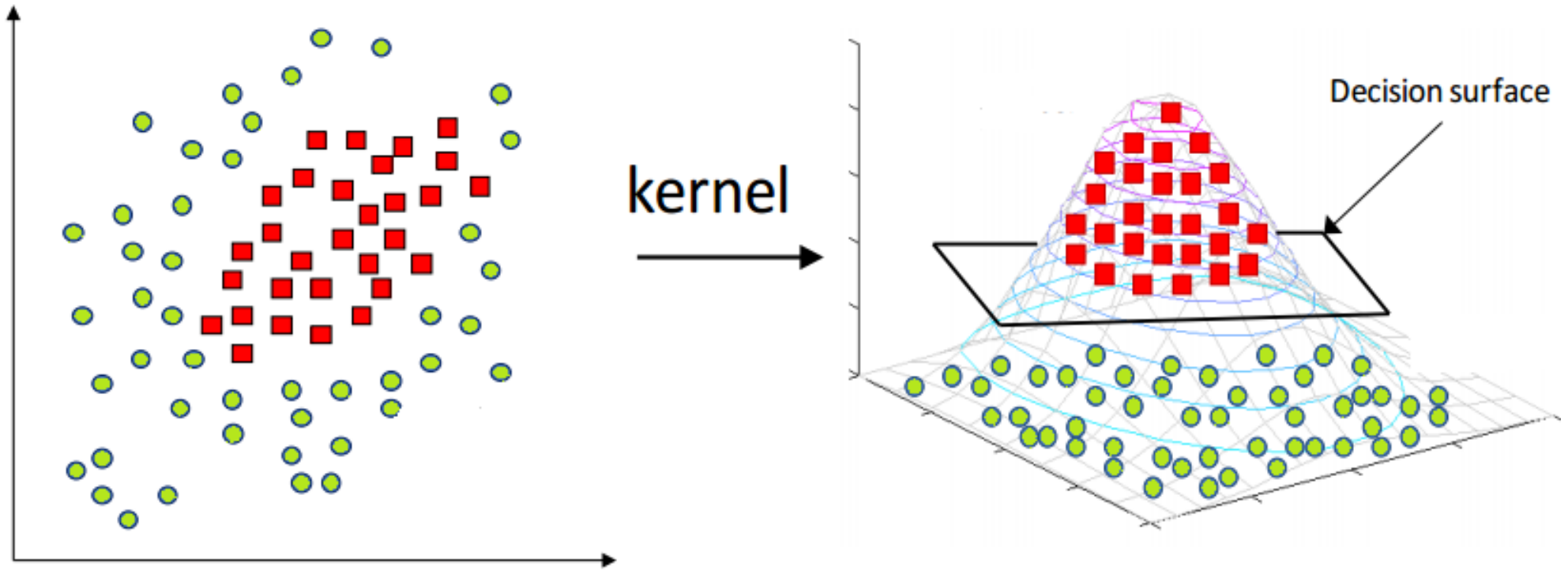
- **Goal**
  - **Correctly classify the data points**
  - **Maximize the margin**

$$\mathcal{J}(w,b) = \lambda\frac{1}{2}||w||^2 + \frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(w \cdot x_i + b))$$



SVM

- For non-linearly separable data:



SVM

- Pros
  - It works really well with a clear margin of separation
  - It is effective in high-dimensional spaces
  - It is effective in cases where the number of dimensions is greater than the number of samples
  - It uses a subset of the training set in the decision function (called support vectors), so it is also memory efficient

- Cons
  - It doesn’t perform well when we have a large data set because the required training time is higher
  - It also doesn’t perform very well when the data set has more noise, i.e., target classes are overlapping
  - SVM doesn’t directly provide probability estimates; these are calculated using an expensive five-fold cross-validation. It is included in the related SVC method of the Python scikit-learn library