CSCAR CONSULTING FOR STATISTICS, COMPUTING & ANALYTICS RESEARCH
UNIVERSITY OF MICHIGAN

Support Vector Machines (SVMs) in Python using Scikit-learn

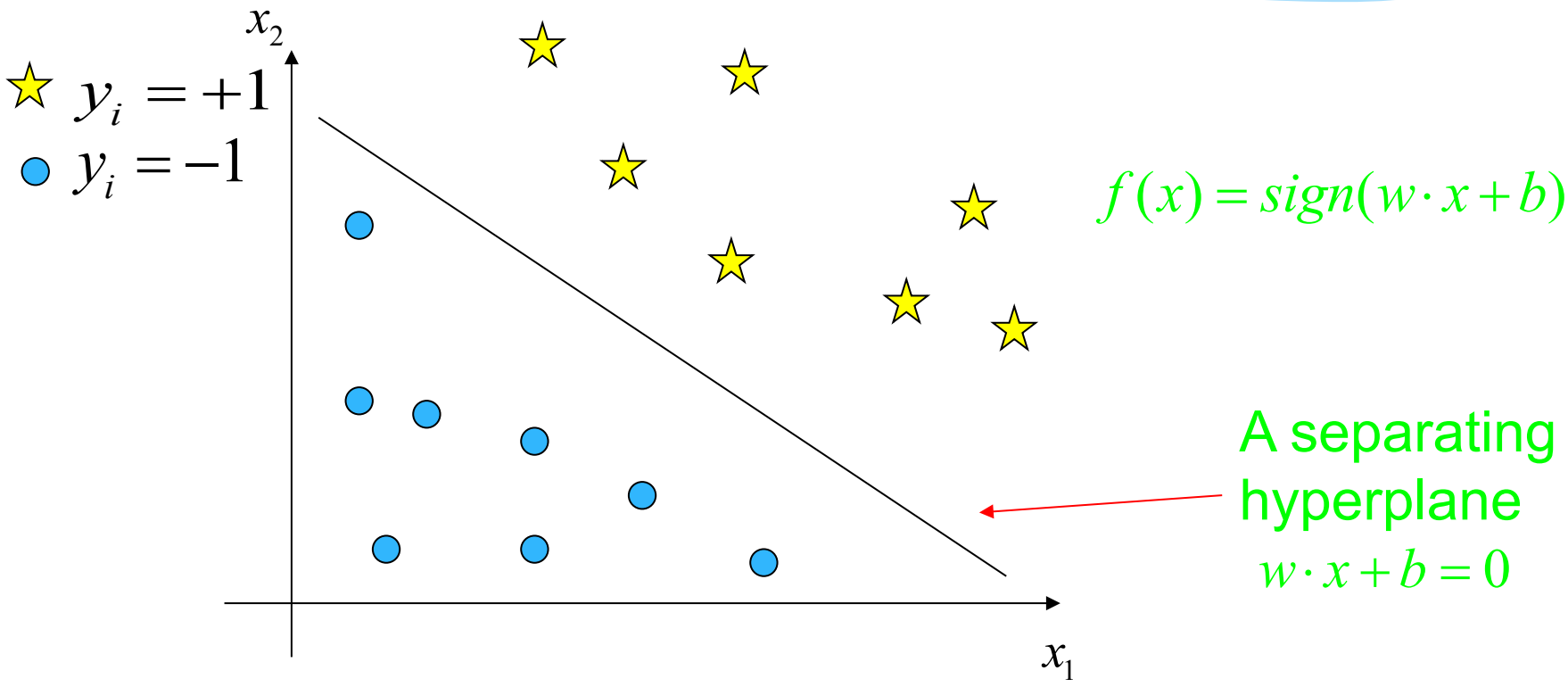
Data Science Skills Series

Márcio Mourão

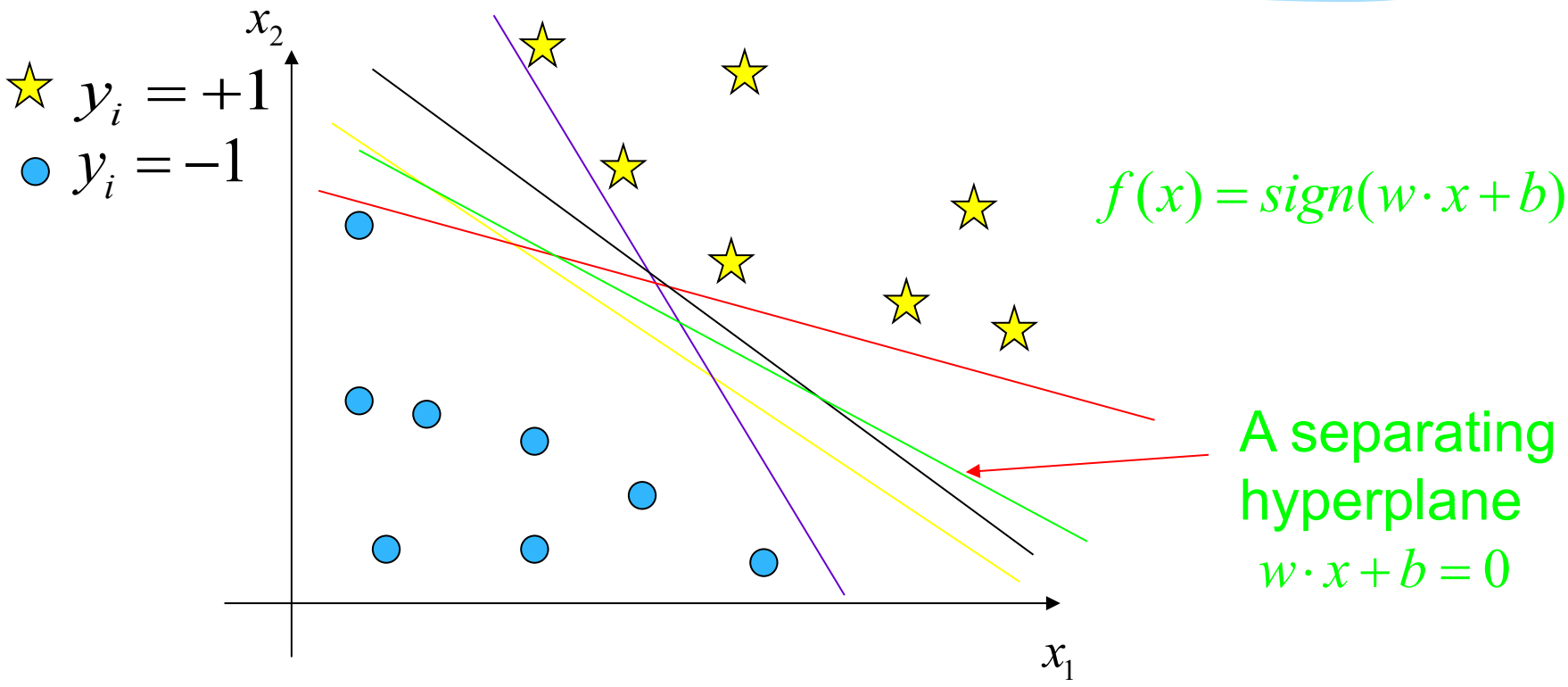
References

* http://scikit-learn.org/
* http://scikit-learn.org/stable/modules/svm.html/
* https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/
* http://www.svm-tutorial.com/2014/11/svm-understanding-math-part-1/
* *http://cs.haifa.ac.il/hagit/courses/seminars/visionTopics/Presentations/SVM_Lecture.ppt/*

The goal of SVM is to find an hyperplane that separates two classes



There are many hyperplanes separating the two classes



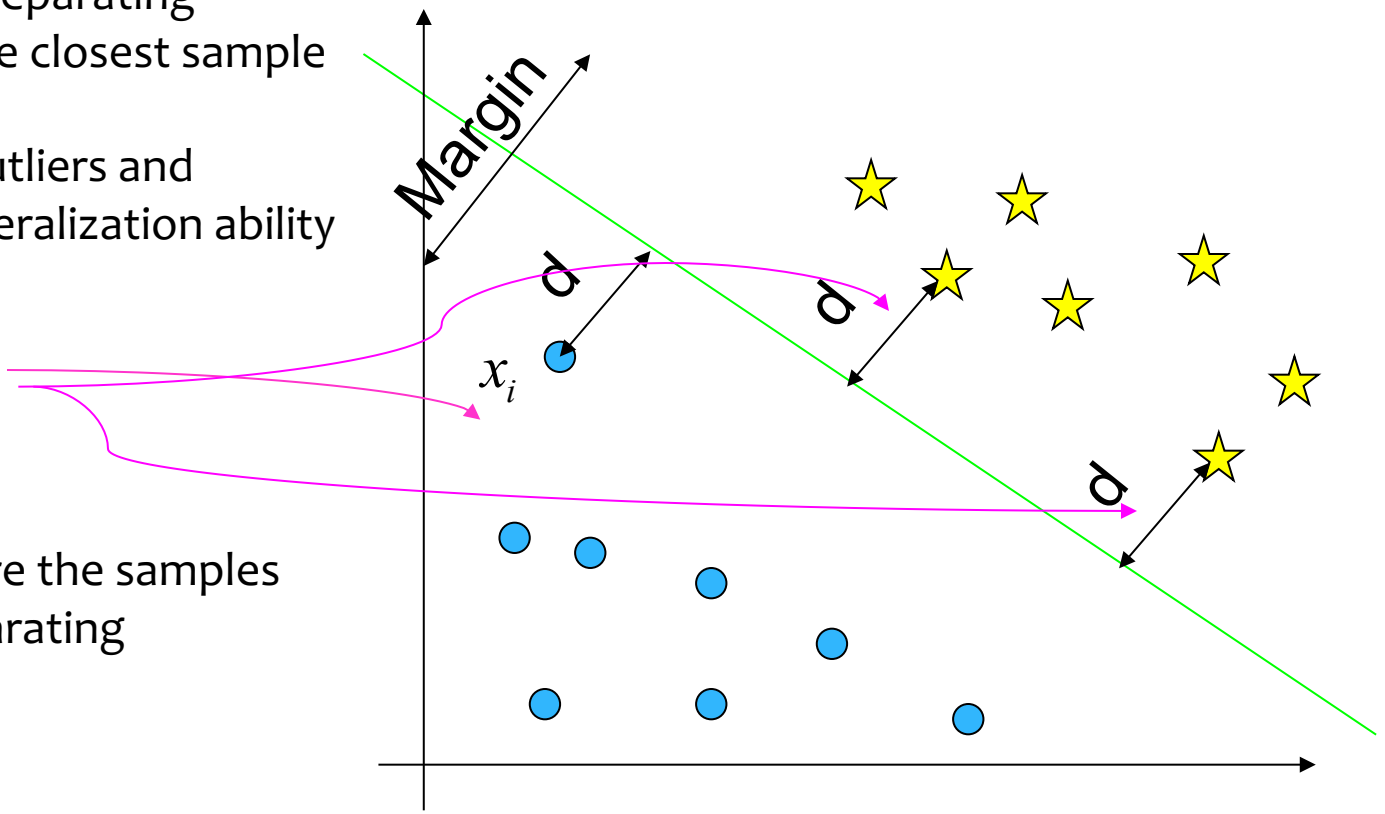
SVM's goal is to maximize the Margin

The Margin is twice the distance "d" between the separating hyperplane and the closest sample

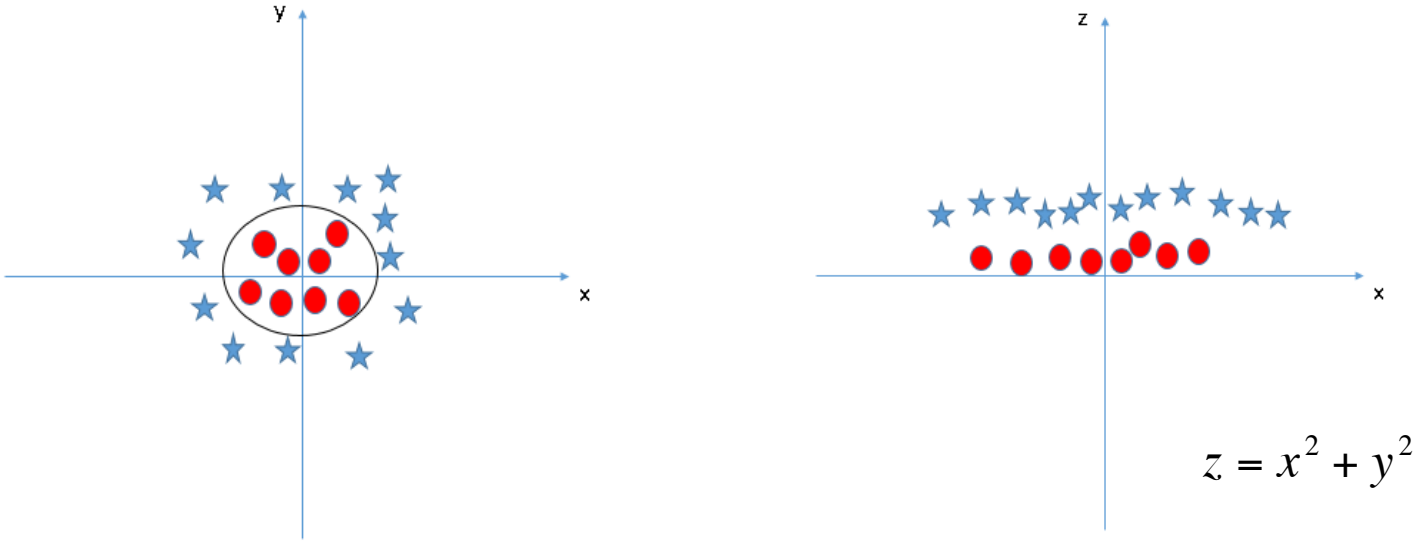
This is robust to outliers and hence, strong generalization ability

Support vectors

Support vectors are the samples closest to the separating hyperplane



What if the points are not linearly separable?

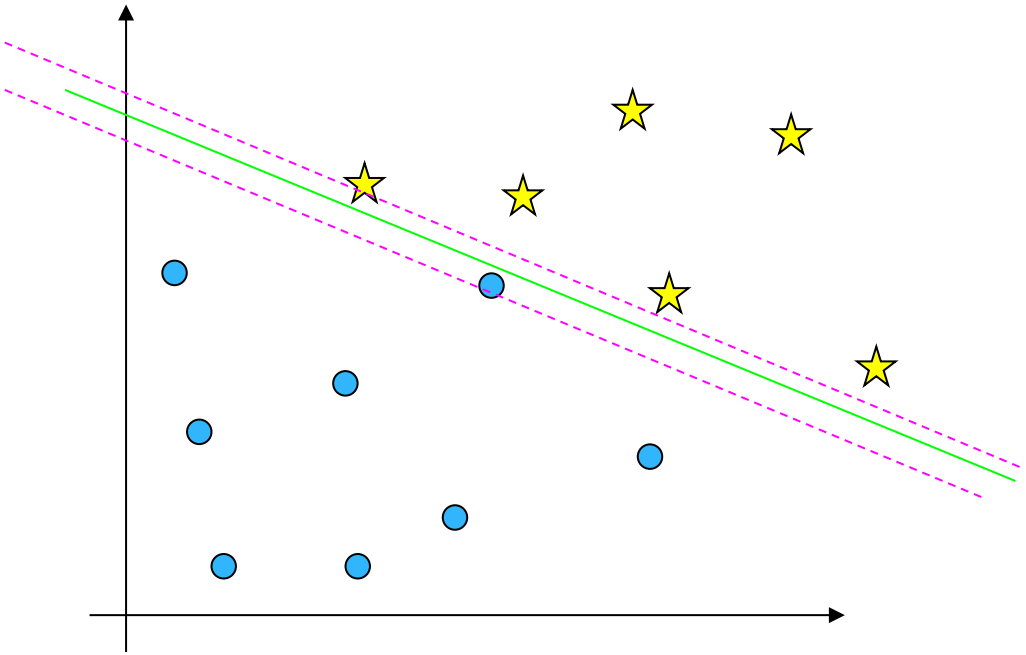


Key idea: map the points to a space of sufficiently high dimension so that they will be separable by a hyperplane: SVM has Kernel functions which take low dimensional input space and transforms it to a higher dimensional space

Higher dimensional spaces introduce additional problems

A strong kernel ,which lifts the data to a great number of dimensions, sometimes may lead us the severe problem of overfitting:

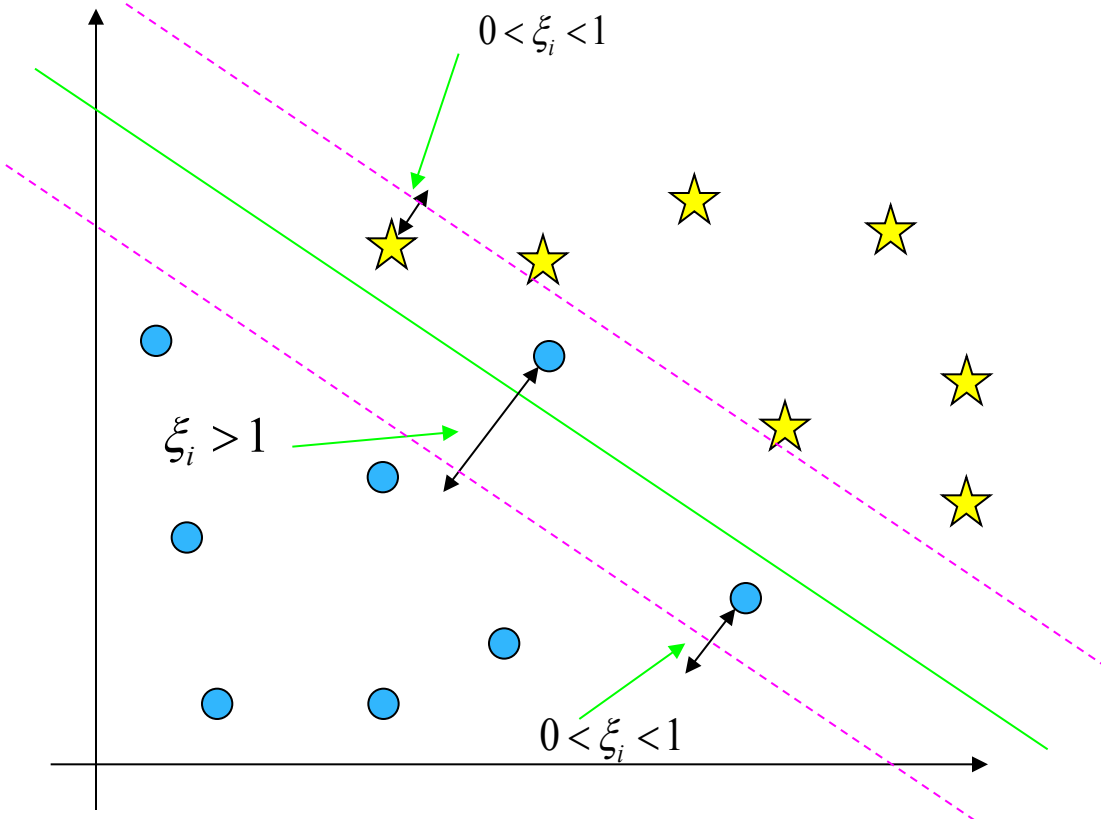
* Low margin → poor classification performance.
* Large number of support vectors→ Slows down the computation.



Overfitting can be partially solved by introducing soft margins

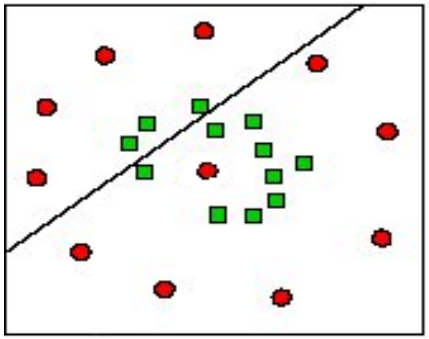
Soft margins allows for some error in classification

Introduction of parameter C for controlling the error term: this controls the trade off between a soft boundary and classifying the training points correctly



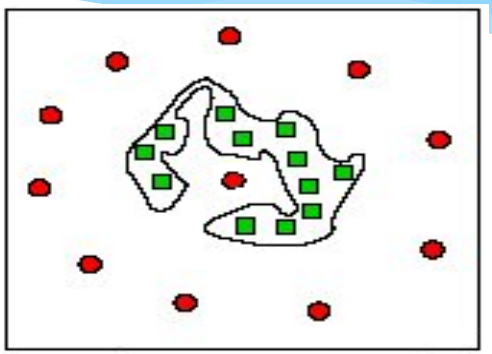
You need to consider the tradeoff between underfitting and overfitting

Under-Fitting

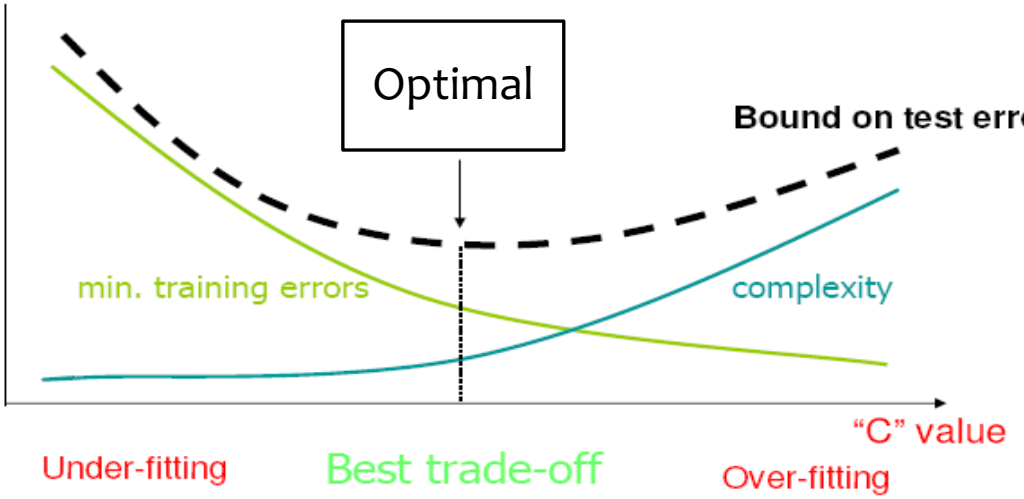


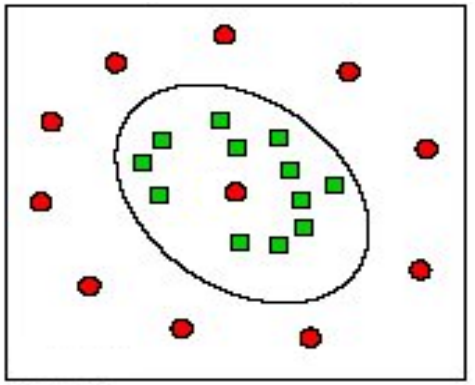
Too much simple!

Over-Fitting



Too much complicated!





Trade-Off

Pros and Cons with SVM

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

* **Cons:**
  * It doesn't perform well, when we have 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 using Scikit-Learn in Python

```
#Import Library
from sklearn import svm

# Create SVM classification object
model = svm.svc(kernel = 'rbf',  gamma = 'auto', C = 1)

# Train the model using the training sets
model.fit(X, y)

#Predict Output
predicted= model.predict(x_test)
```