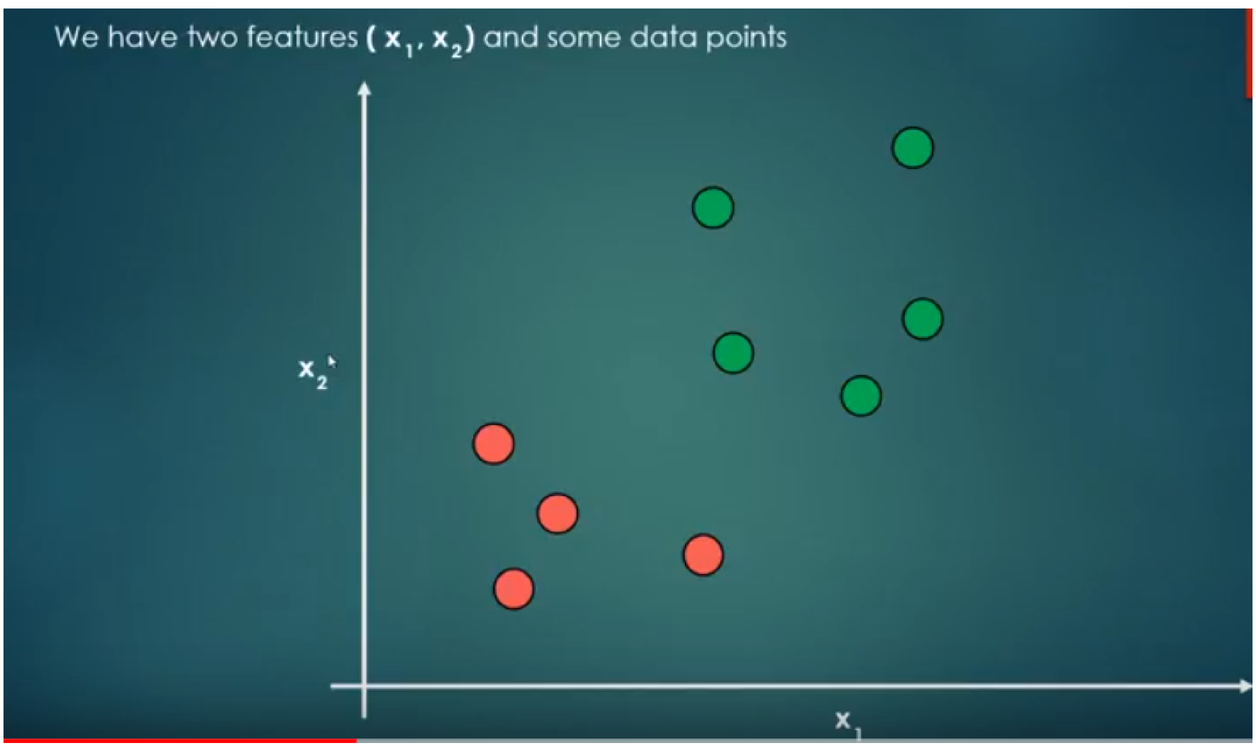
Support vector machine (SVM)

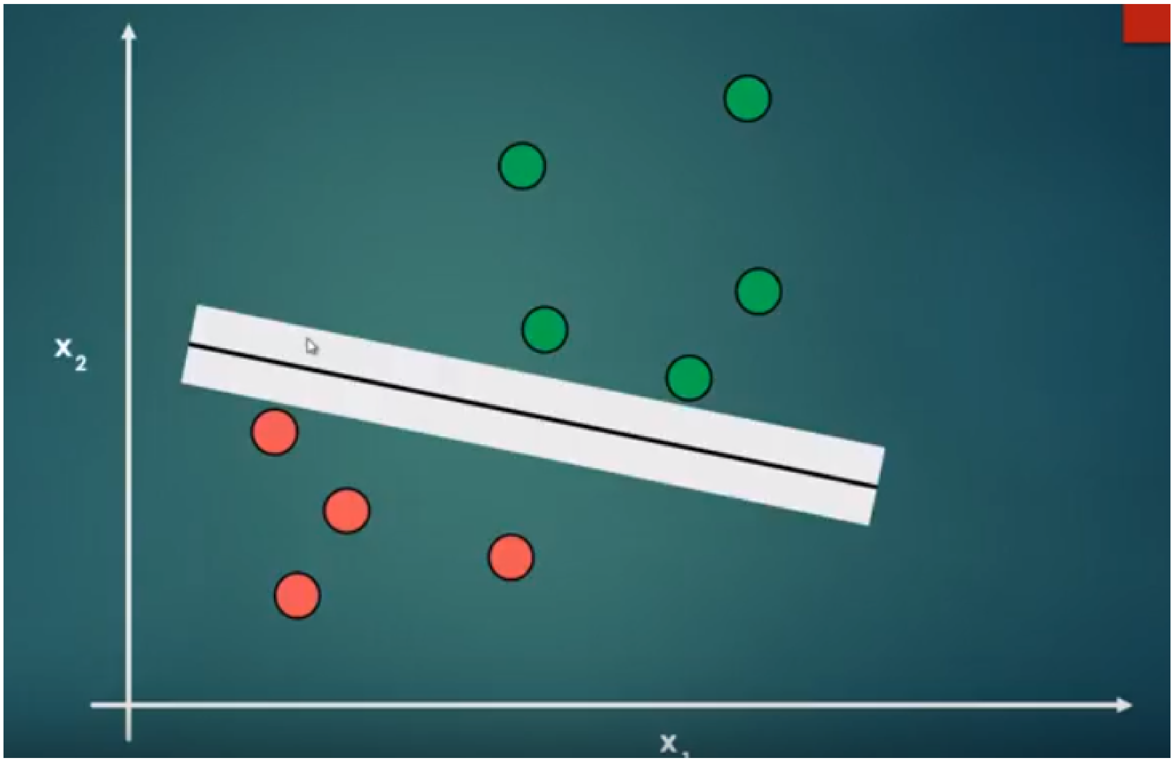
- Very popular and widely used supervised learning classification algorithm
- The great benefit: it can operates even in infinite dimension
- SVM finds a hyperplane or decision surface or (line) that leads to a homogeneous partition of data
- A good separation is achieved by the hyperplane that has largest distance to the nearest training data points of any class
- so we have to maximize the margin
- performs classification by finding the hyperplane that maximize the margin between two classes

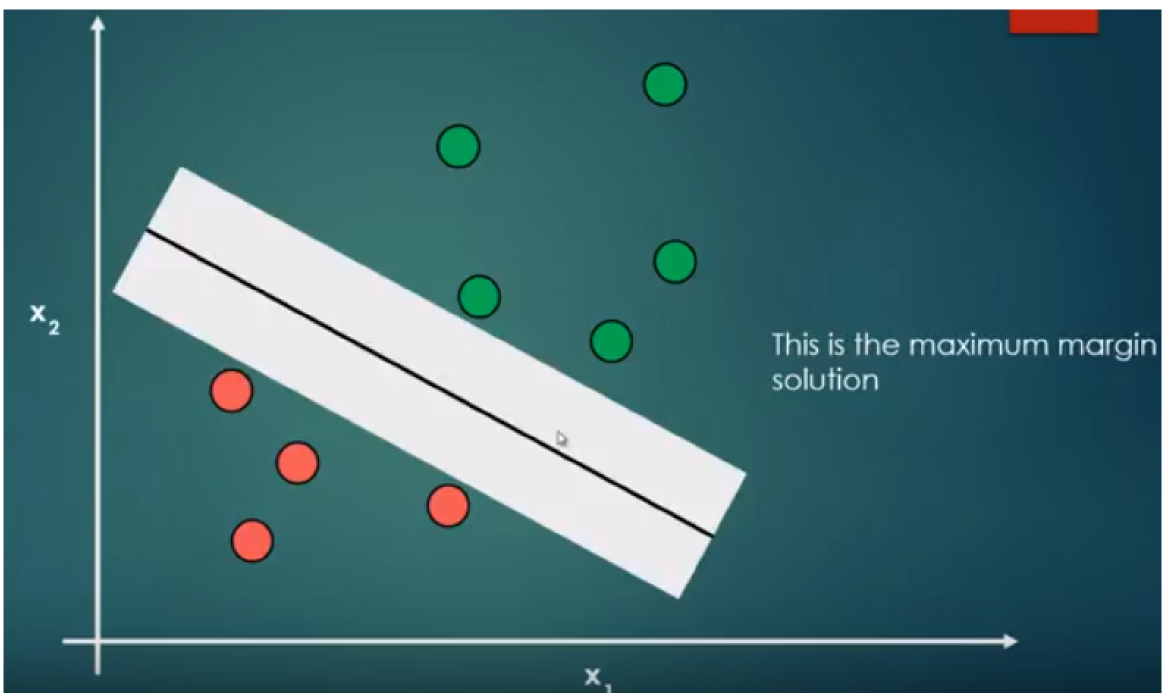


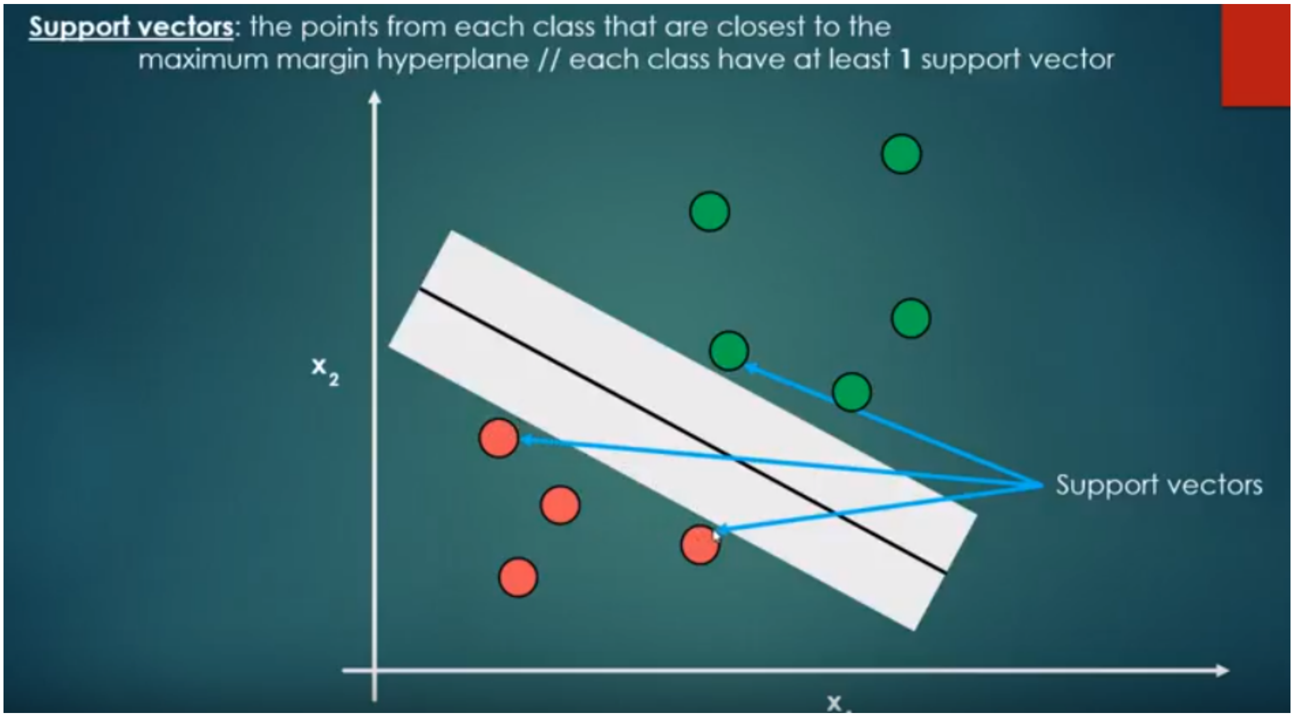
Goal is:

classifiy the data points by finding the straight line (or hyper-plane) that differentiate the two classes with **maximum margin**

Hyperplane might be this...

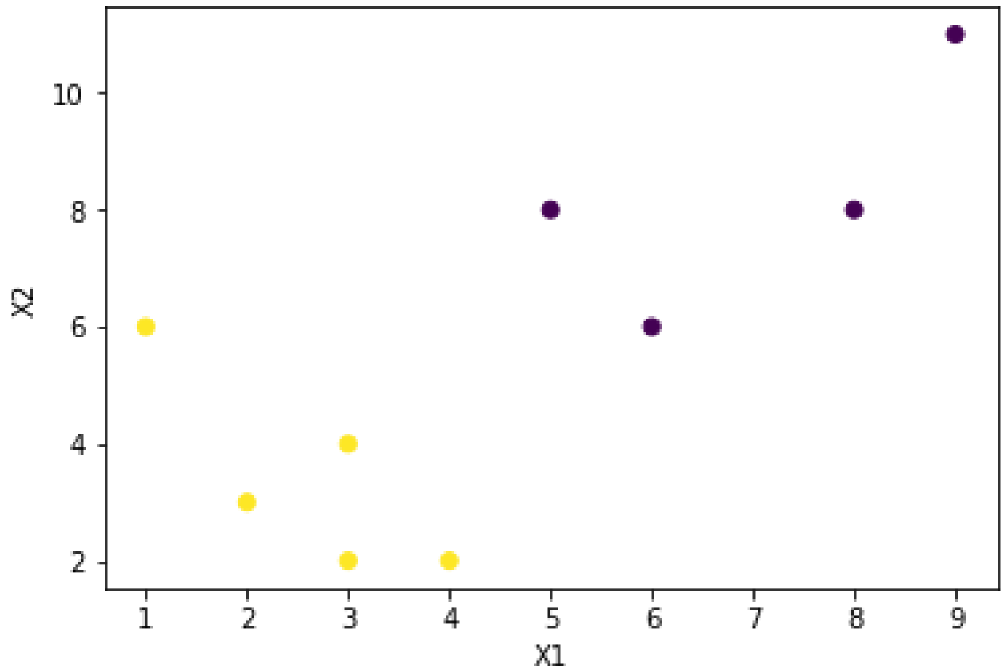






How to find hyperplane when problem is linearly separable

```
import  numpy as np

import matplotlib.pyplot as plt

x1=[1,2,5,4,3,8,3,9,6]

x2 =[6,3,8,2,4,8,2,11,6]

y=[1,1,0,1,1,0,1,0,0]

plt.scatter(x1,x2,c=y)

plt.xlabel('X1')

plt.ylabel('X2')

plt.show()
```



Here a line easily separate these data points so hyperplane is a line

clf = SVC(C=1.0,kernel='linear')

and line:

mX+b

or

wX+b here w or m is weight of feature x

if we have 2 features then line:

m1X1+m2X2+b

or

w1X1+w2X2+b

similarly for n features

m1X1+m2X2+………..mnXn+ b

or

w1X1+w2X2+……+wnXn +b

or

$$\vec{w} * \vec{x} + b$$

If we have 2 classes then

$$w * x + b \geq +1$$

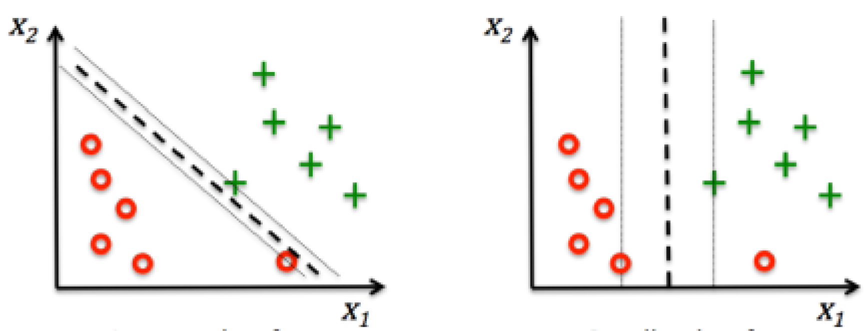
testing data points to class 1

And,

$$w * x + b \leq -1$$ more precisely < -1

testing data points to class 0

Role of C parameter:

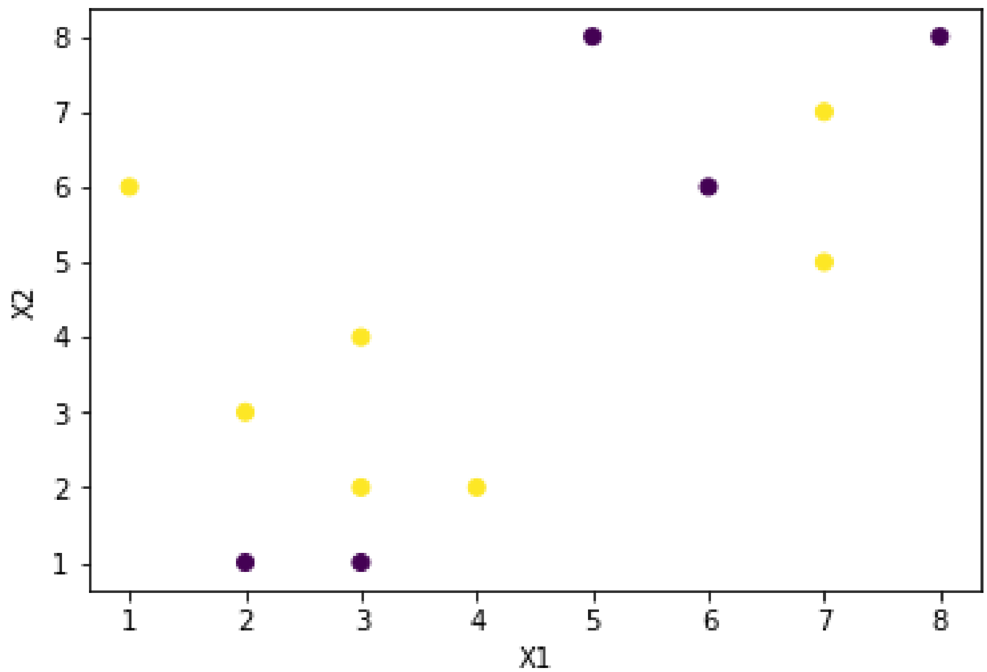


Large value for parameter *C*　　Small value for parameter *C*

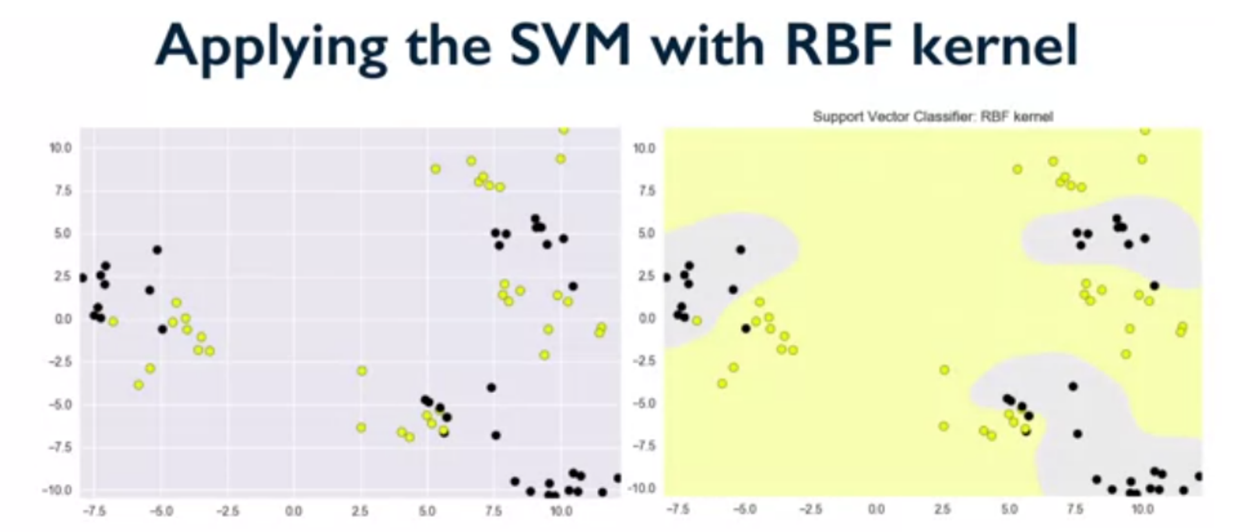
- Regularization: How much importance should you give individual data points as compared to better generalized model
- Regularization parameter c
  - Larger values of c = less regularization
    - Fit training data as well as possible, every data point important
  - Smaller values of c = more regularization
    - More tolerant to errors on individual data points

How to find hyperplane when problem is non-linearly separable

```
import  numpy as np

import matplotlib.pyplot as plt

x1=[1,2,5,4,3,8,3,2,6,3,7,7]

x2 =[6,3,8,2,4,8,2,1,6,1,7,5]

y=[1,1,0,1,1,0,1,0,0,0,1,1]

plt.scatter(x1,x2,c=y)

plt.xlabel('X1')

plt.ylabel('X2')

plt.show()
```



clf = SVC(kernel='rbf') // (Radial basis function)



In this case value of y is given by:

Exp(-gamma*sqrt(sqr(x1-x2)))+b

Now classification is done on the basis of

Exp(-gamma*sqrt(sqr(x1-x2)))+b   >=0   class 1

And

Exp(-gamma*sqrt(sqr(x1-x2)))+b   <0   class 0

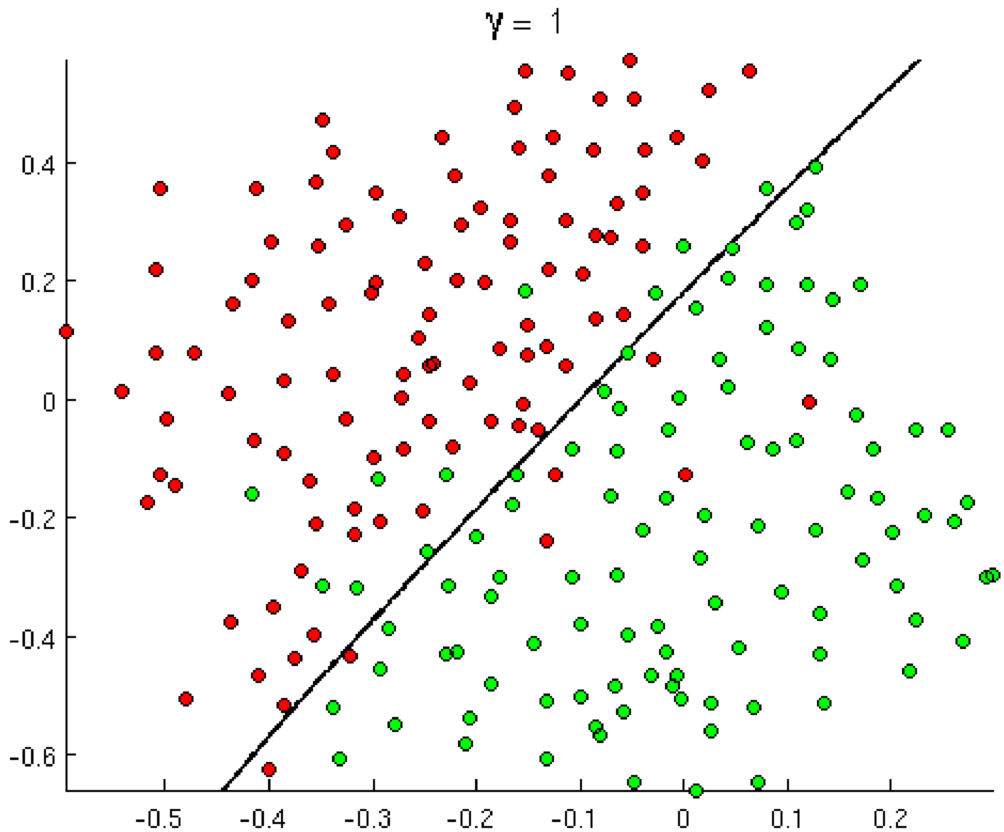
Role of gamma parameter in RBF:

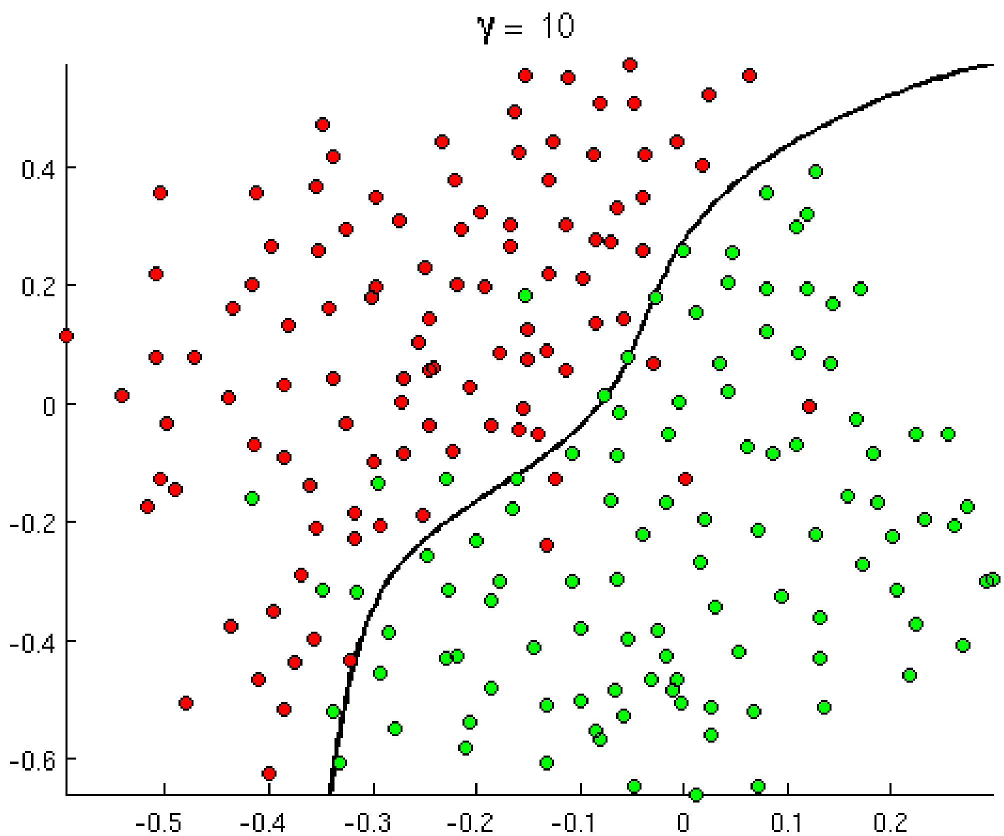
This shows that as gamma increases, the algorithm tries harder to avoid misclassifying training data, which leads to overfitting.

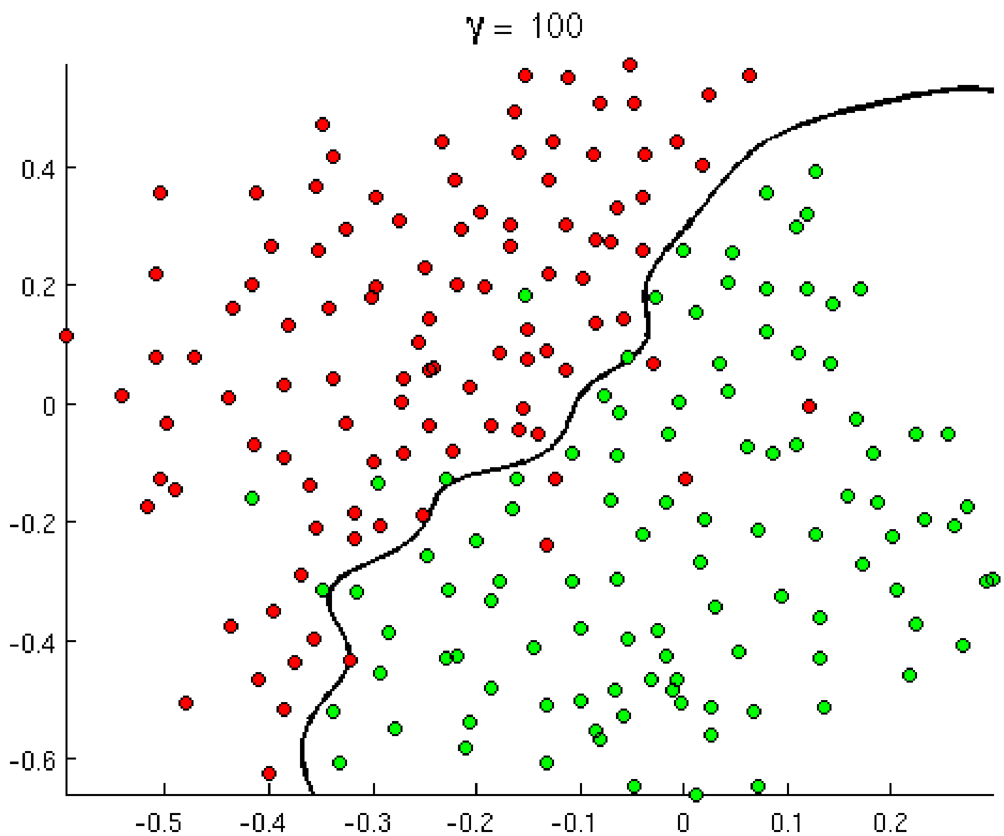
i.e.

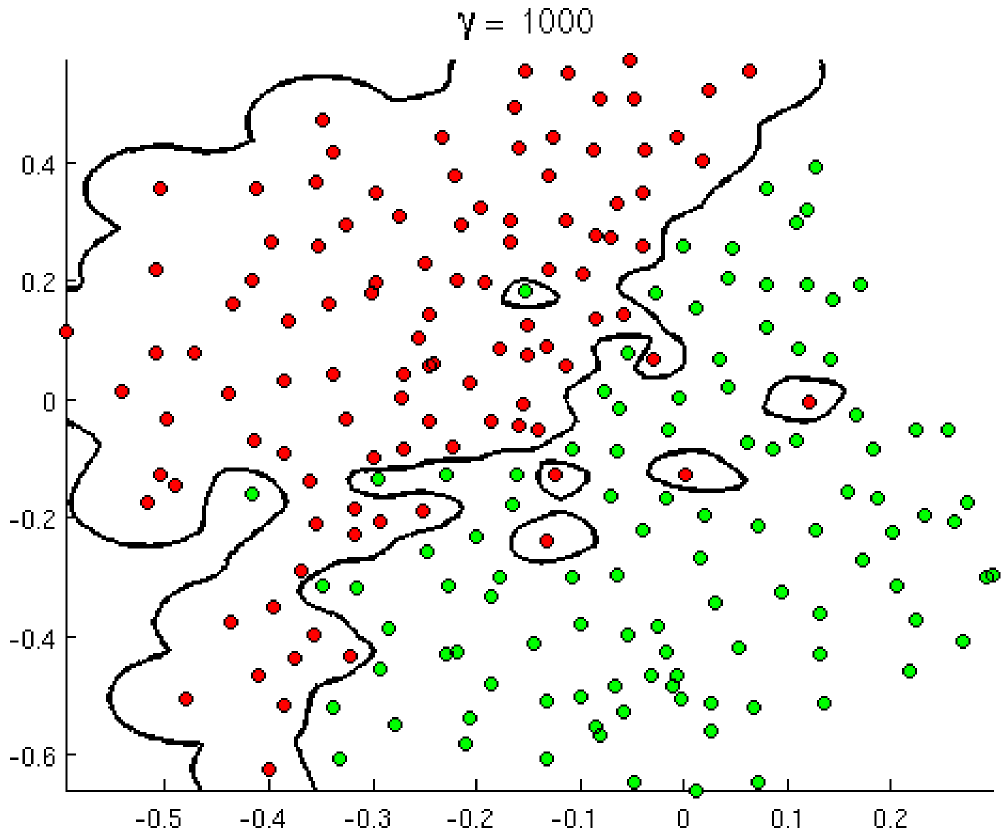
higher value of gamma ………..overfittting

lesser value of gamma………….underfitting


γ = 1
0.4
0.2
0
-0.2
-0.4
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2


γ = 10
0.4
0.2
0
-0.2
-0.4
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2


γ = 100
0.4
0.2
0
-0.2
-0.4
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2



γ = 1000
0.4
0.2
0
-0.2
-0.4
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2