Masini cu suport vectorial
Support vector machines (SVM)

Materiale de citit

- **Capitolul 9** din cartea: **An Introduction to Statistical Learning with Applications in R**. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. Springer-Verlag, 2013. (carte disponibilă gratuit online aici: http://www-bcf.usc.edu/~gareth/ISL/)

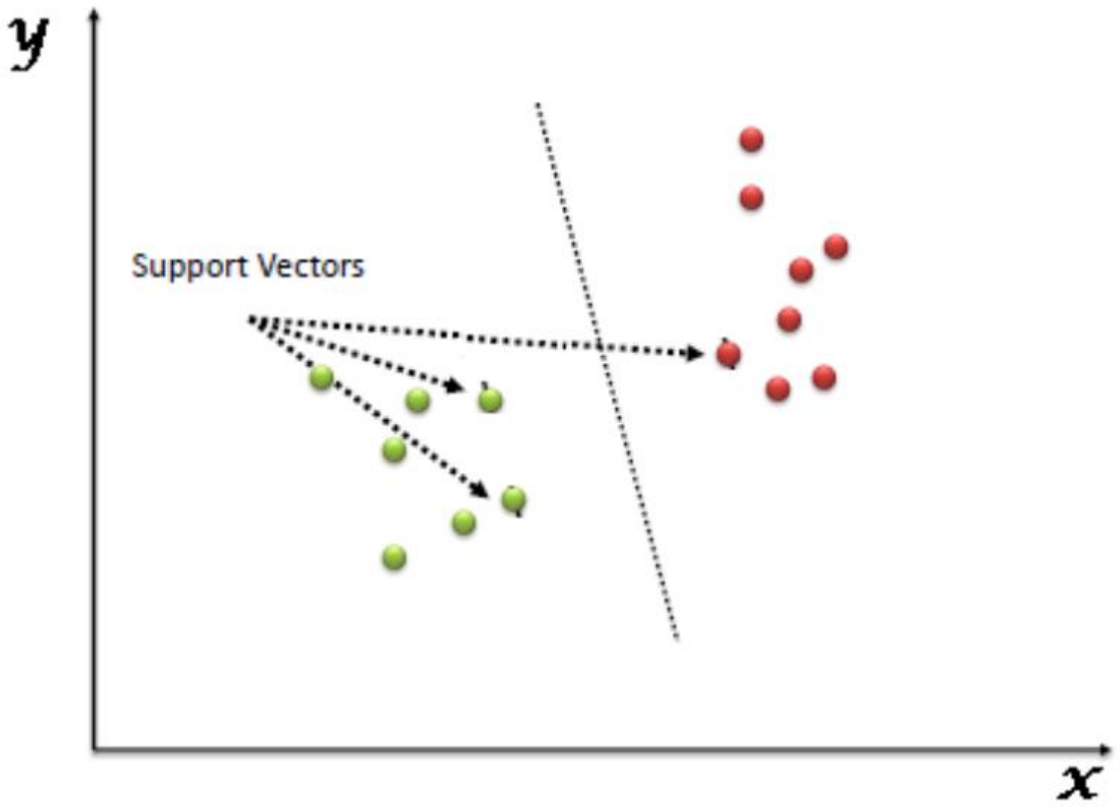
Prezentare

- Unelte foarte puternice in:
  - Clasificare
    - Recunoasterea scrisului
    - Recunoasterea obiectelor
    - Recunoasterea vorbirii
    - Categorizare de texte
    - In principiu binara – metode de extinder multe clase
  - Regresie

- **Faza de antrenament:** invatare *(exemplu → eticheta)*
- **Faza de test**: predictie asupra *(exemplu nou → ?)*

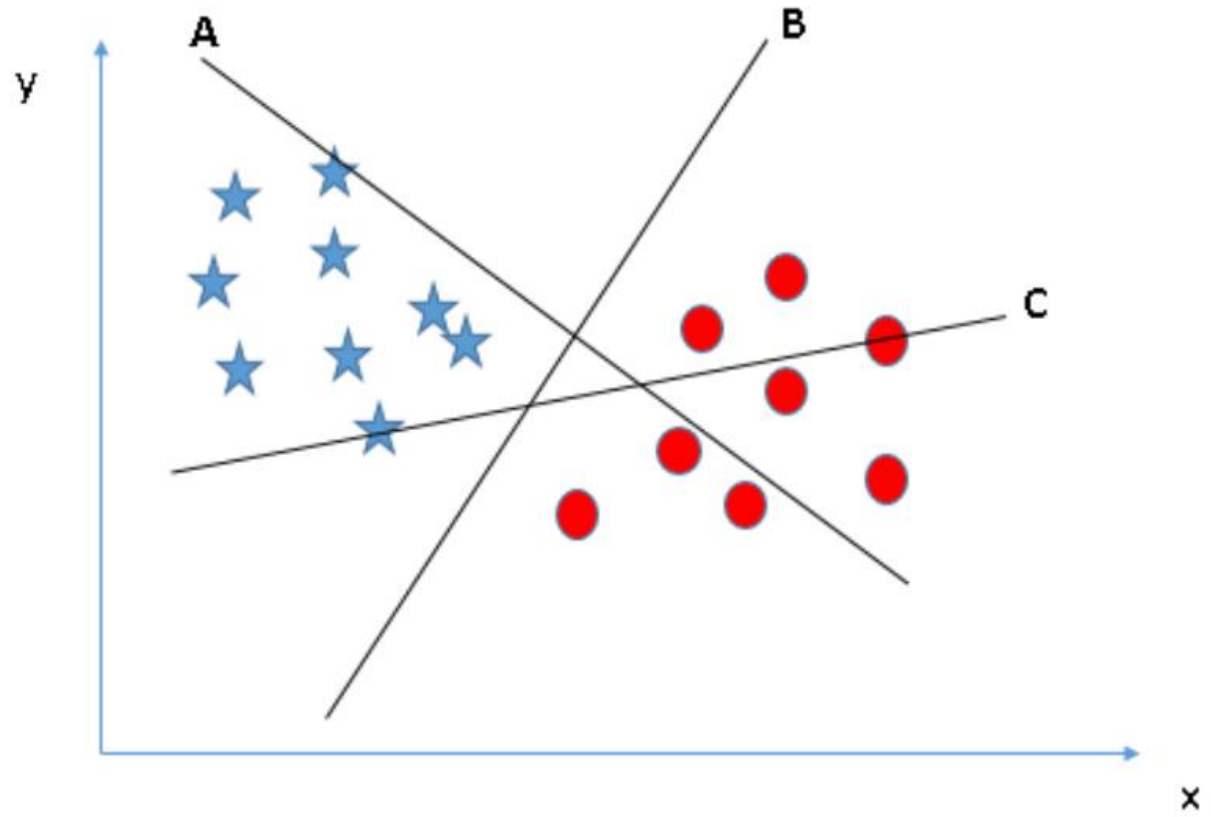
- Un tip de masina instruibila supervizata

Ce sunt SVM?

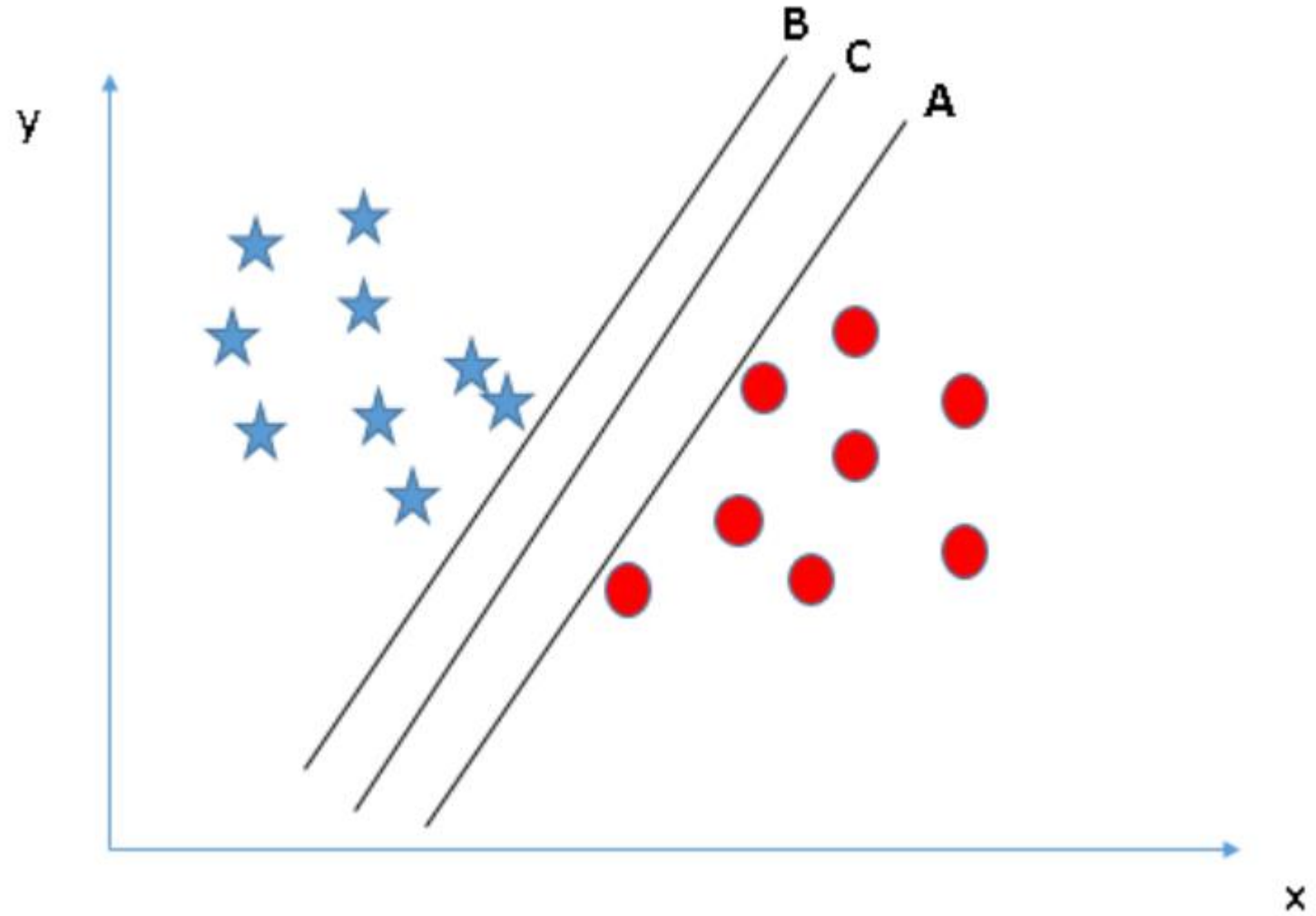


- Se reprezinta fiecare punct intr-un spatiu n-dimensional (n fiind numarul de atribute), si valoarea unui atribut reprezentand o coordonata
- Se cauta hiperplanul care separa cel mai bine punctele
- Vectorii support sunt coordonatele

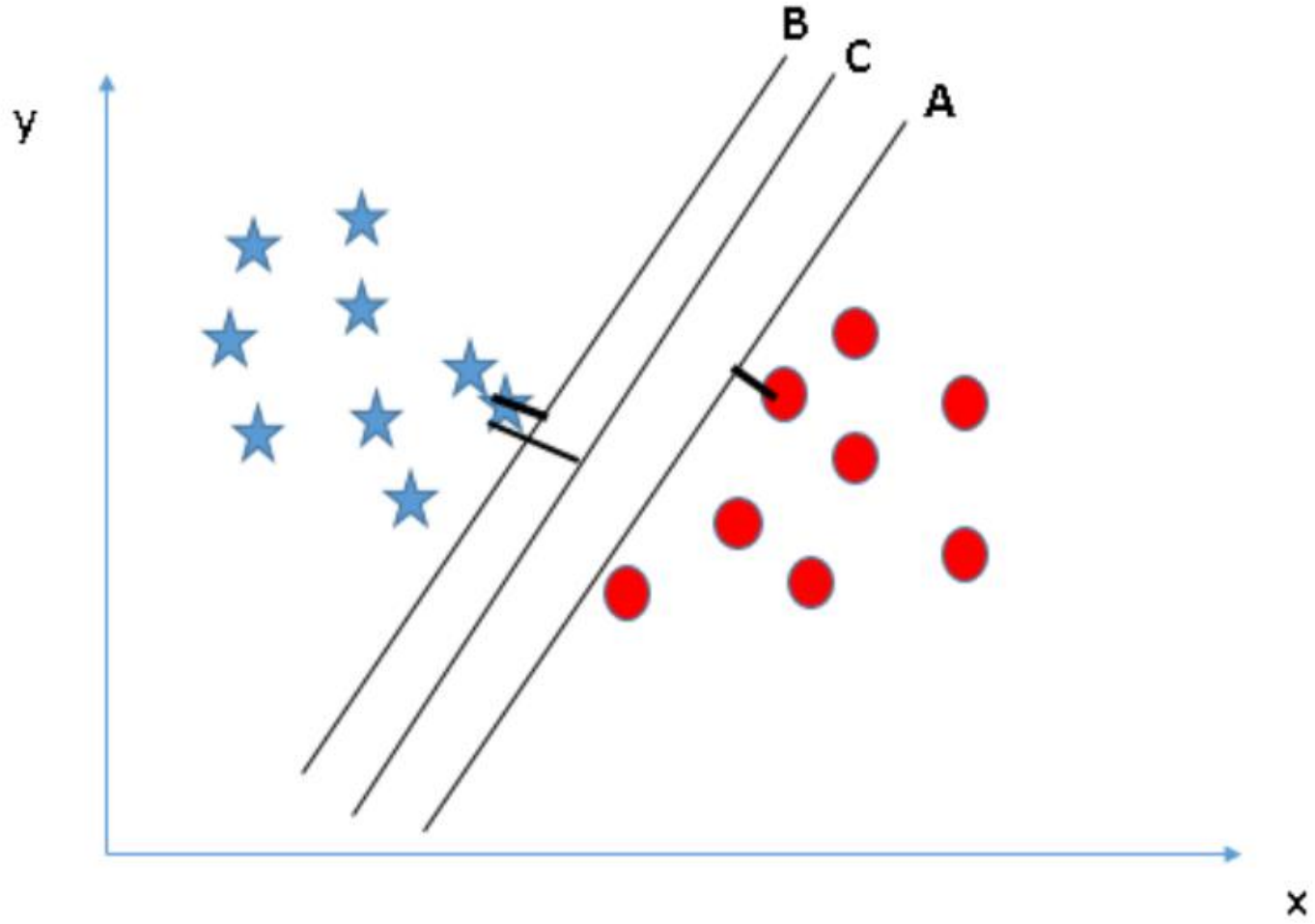
Care este hiperplanul cel mai bun?



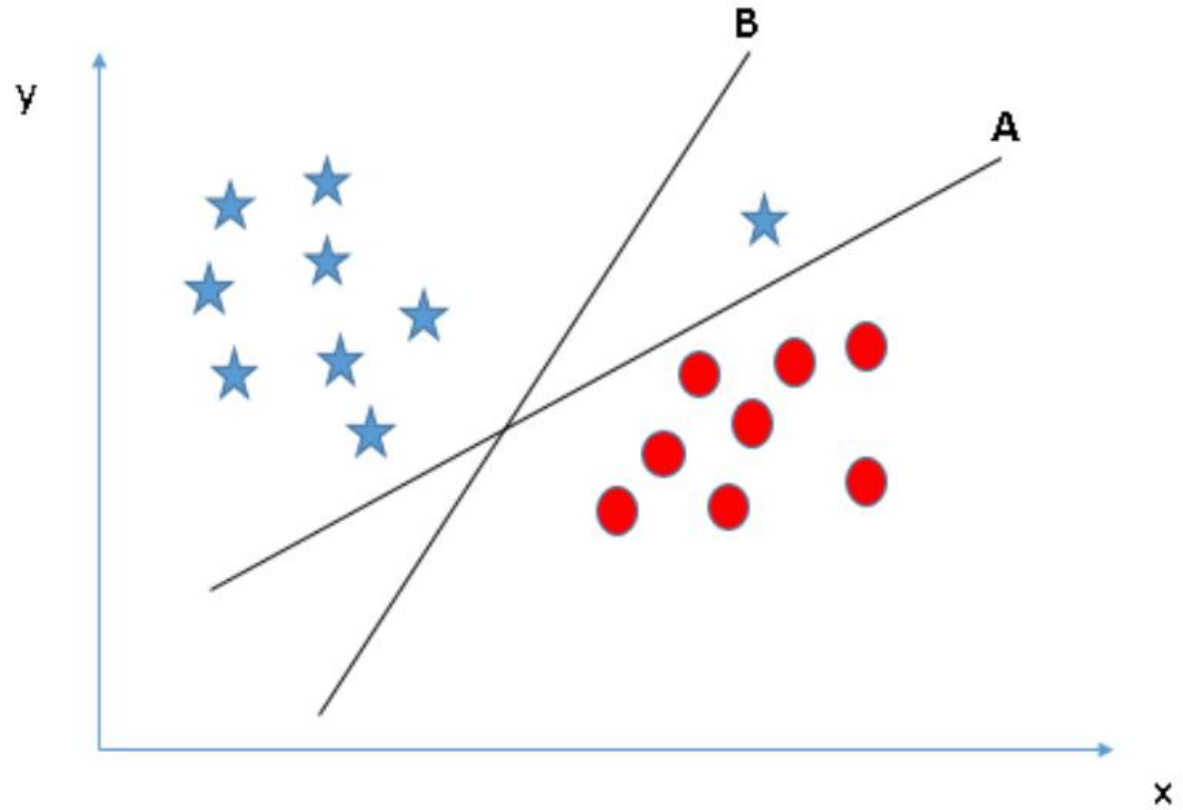
Care este hiperplanul cel mai bun?



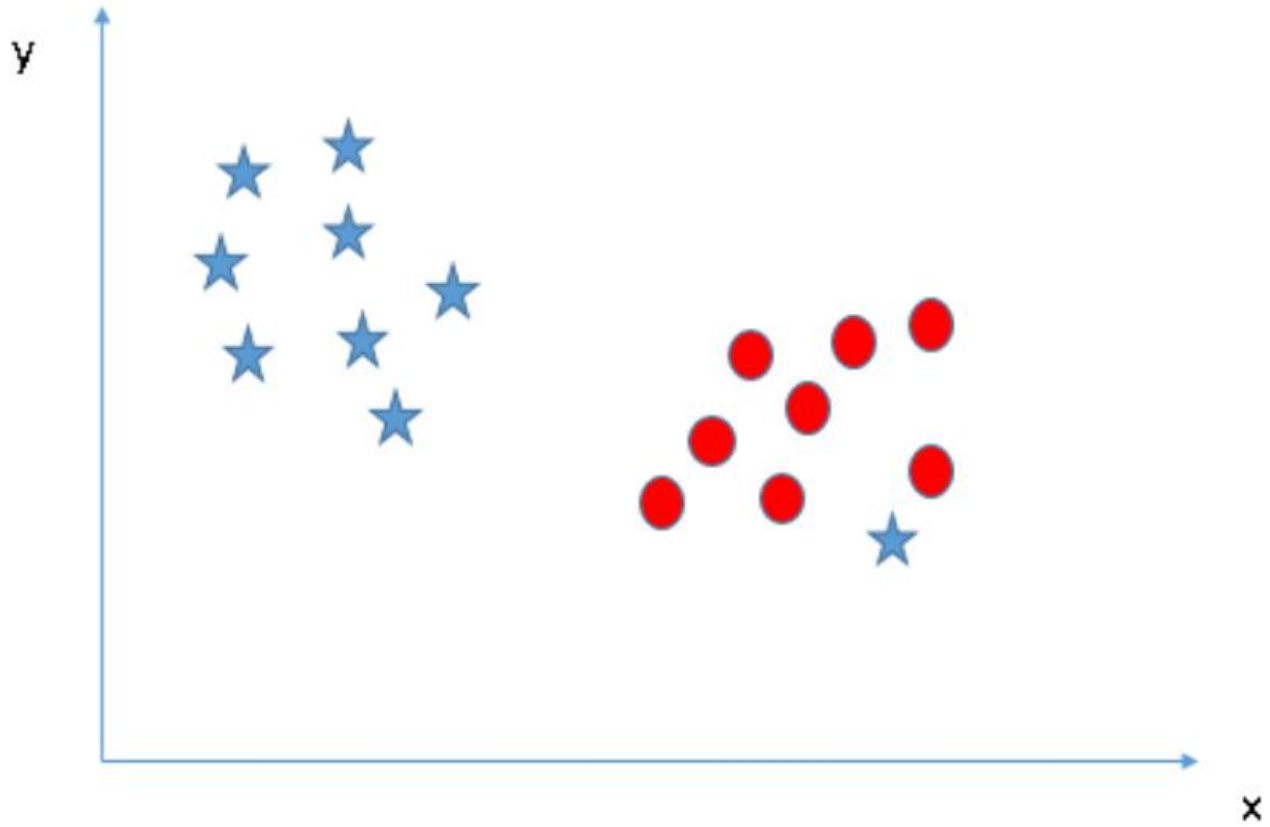
Care este hiperplanul cel mai bun?



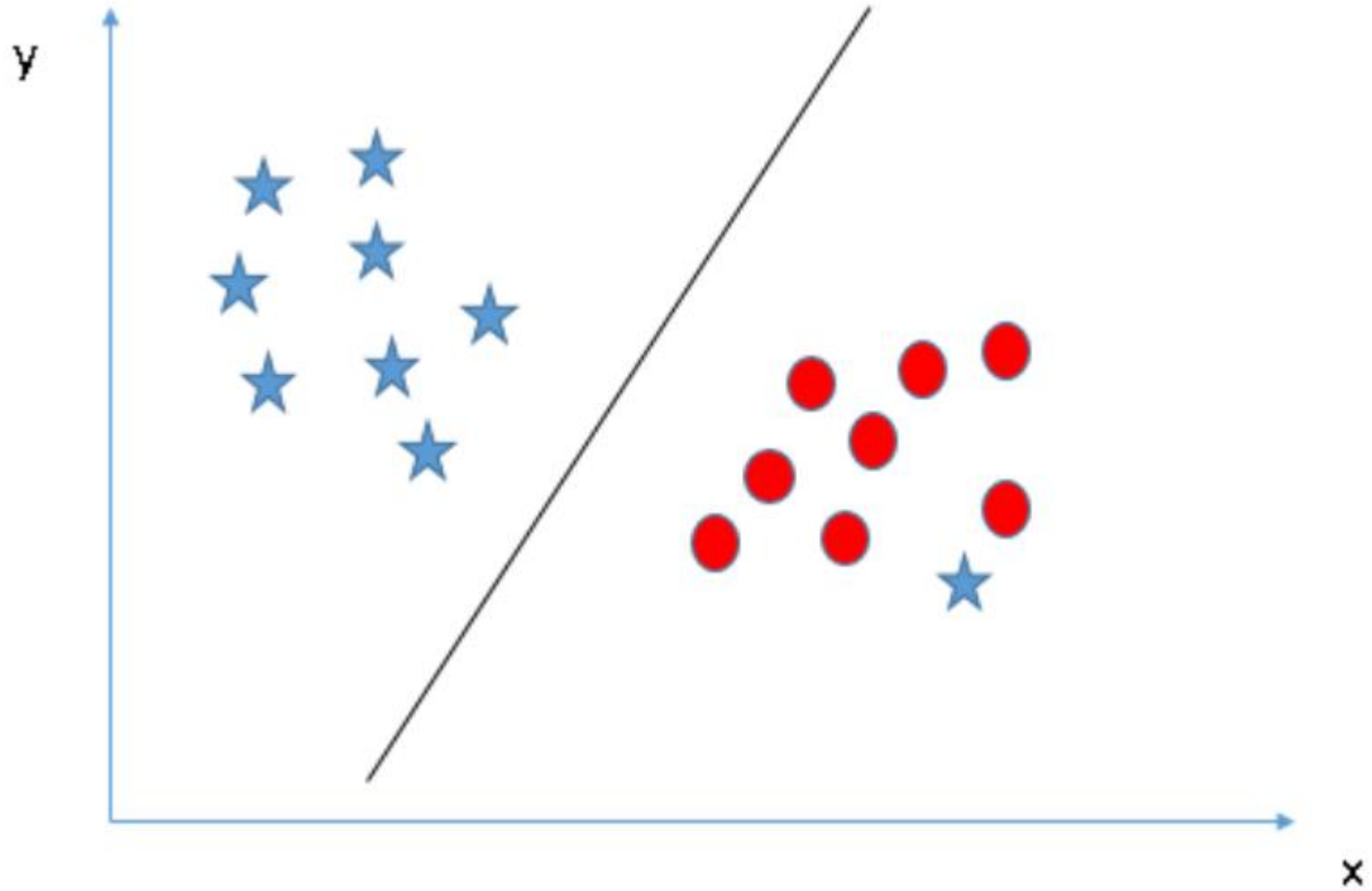
Care este hiperplanul cel mai bun?



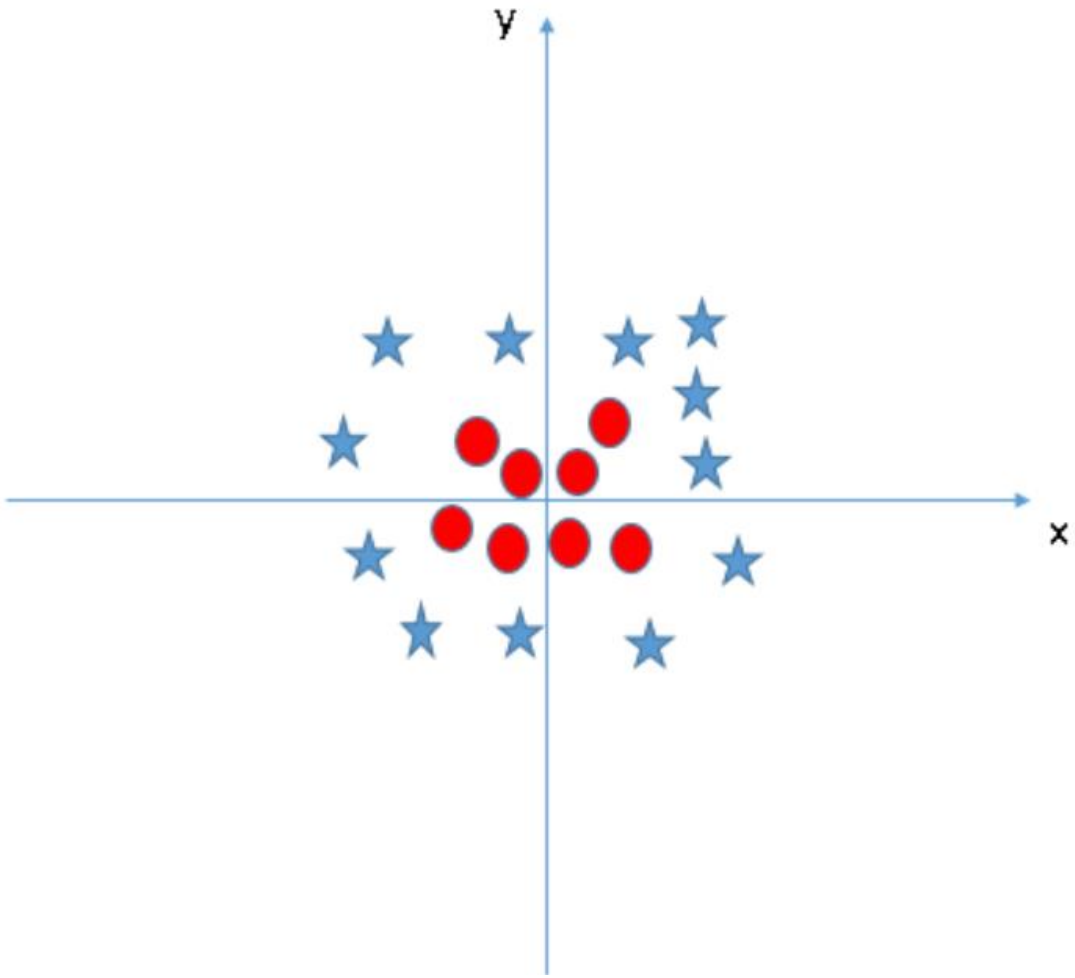
Care este hiperplanul cel mai bun?



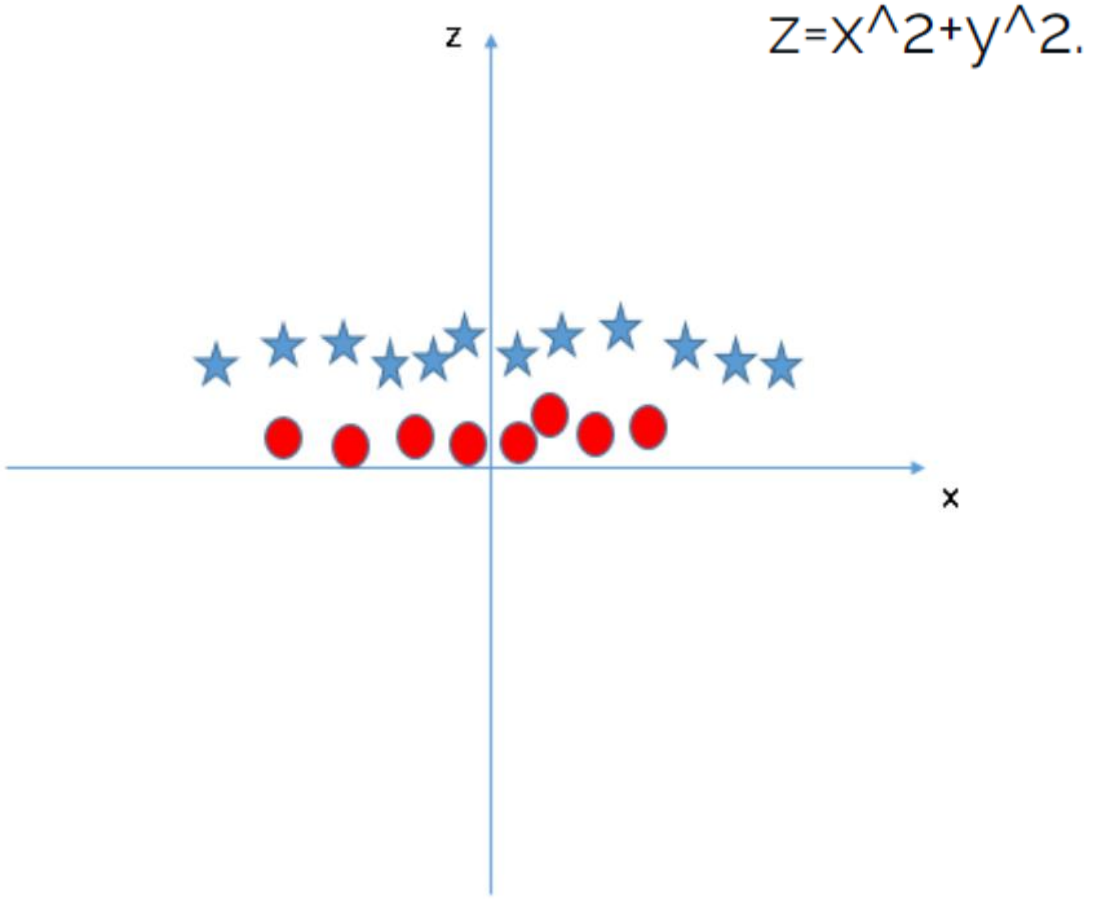
Care este hiperplanul cel mai bun?



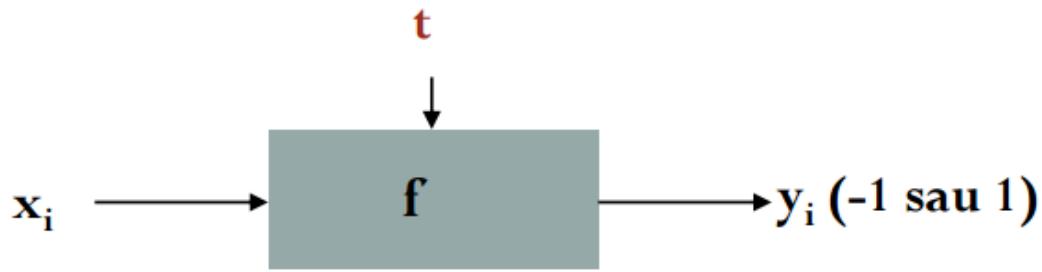
Care este hiperplanul cel mai bun?



Care este hiperplanul cel mai bun?



Clasificare binara



- O multime de date de antrenament

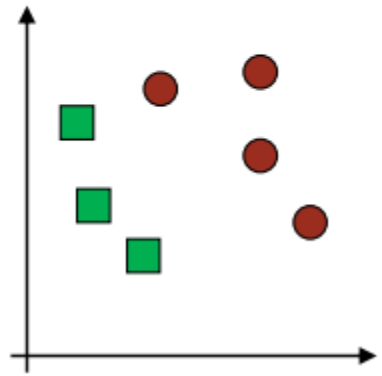
$$\{(x_i, y_i)\}_{i=1,2,\ldots m} \qquad x_i \in R^n \qquad y_i \in \{-1,+1\}$$

- O familie de functii

$$\{f_t \mid t \in T\} \qquad f_t : \Re^n \to \{-1,+1\}$$

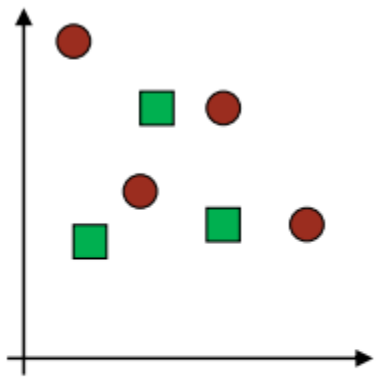
- Gaseste parametrii t pentru a invata corespondenta optima intre fiecare x si y

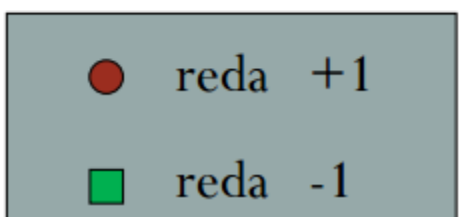
Invatarea in cadrul SVM



Liniar separabile

Masini cu suport vectorial liniare



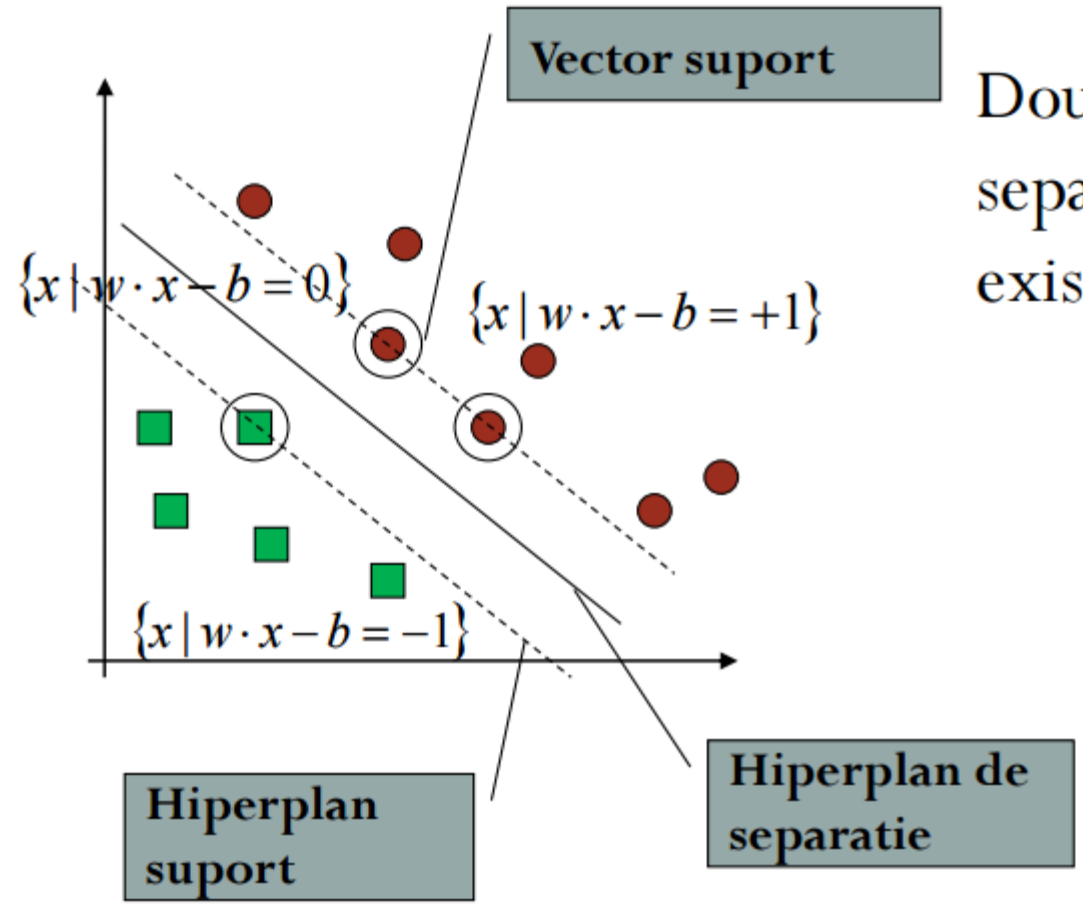


Liniar neseparabile

Masini cu suport vectorial liniare

Masini cu suport vectorial neliniare

SVM liniare pentru date separabile



Doua submultimi sunt liniar separabile daca si numai daca exista $w \in R^n$ si $b \in R$ a.i.:

$$\begin{cases} w \cdot x_i - b \geq +1, y_i = +1 \\ w \cdot x_i - b \leq -1, y_i = -1 \end{cases}$$

$$i = 1,2,\ldots,m$$

SVM liniare pentru date separabile

- O familie de functii:

$$\left\{f_{w,b} \mid w \in \Re^n, b \in \Re\right\} \qquad f_{w,b} : \Re^n \to \{-1, 1\} \qquad f_{w,b}(x) = \operatorname{sgn}(w \cdot x - b)$$

- O multime de date de antrenament:

$$\{(x_i, y_i)\}_{i=1,2,\ldots,m} \qquad x_i \in R^n \qquad y_i \in \{-1,+1\}$$

- Invata corespondenta optima

Principiul minimizarii riscului

- Pentru o sarcina de invatare, cu o anumita cantitate finita de date de antrenament, o masina este un bun generalizator daca:
  - Are o acuratete buna pe multimea de antrenament data
  - Are capacitatea de a invata si alte multimi de antrenament fara eroare

Hiperplanul optim la SVM liniare pentru date separabile

$w = ?$ si $b = ?$ ***a.i.***

1. **Acuratete** $y_i(w \cdot x_i - b) - 1 \geq 0 \quad i = 1,2,\ldots,m$

2. **Marginea de separatie maxima**

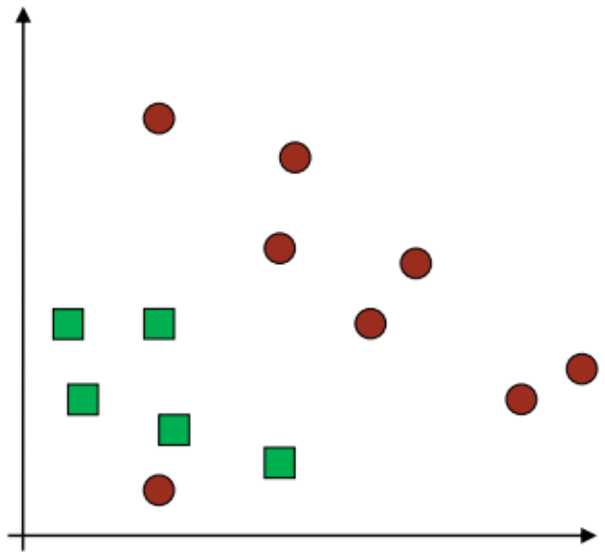
Latimea cu care granita dintre clase poate fi marita pana la lovi un exemplu

➡ **minimizeaza** $\frac{\|w\|^2}{2}$ **De ce?**

Distanta de la fiecare cel mai apropiat punct de hiperplanul de separatie din cele doua parti ale sale este :

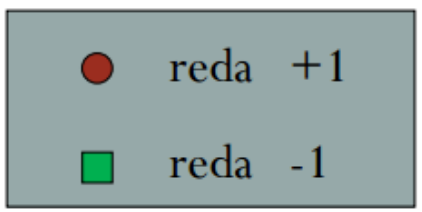
$$\frac{|w \cdot x_i - b|}{\|w\|} = \frac{1}{\|w\|} \quad \Rightarrow \frac{2}{\|w\|} \rightarrow \text{maxim}$$

SVM liniare pentru date neseparabile

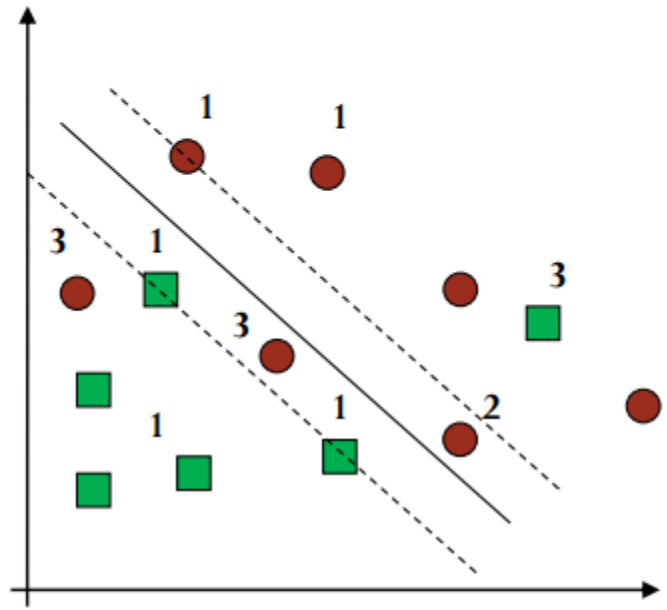


Cum separam aceste noi date?

Putem relaxa conditia de separare.



SVM liniare pentru date neseparabile



Fiecare vector de antrenament are o deviatie de la hiperplanul sau suport de

$$\pm \frac{\xi_i}{\|w\|} \qquad \xi_i \geq 0$$

$\xi_i = 0$	**1 Corect**
$\xi_i < 1$	**2 Corect (in margine)**
$\xi_i > 1$	**3 Eroare**

reda +1

reda -1

Hiperplanul optim la SVM liniare pentru date neseparabile

- Se relaxeaza constrangerile :

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i$$

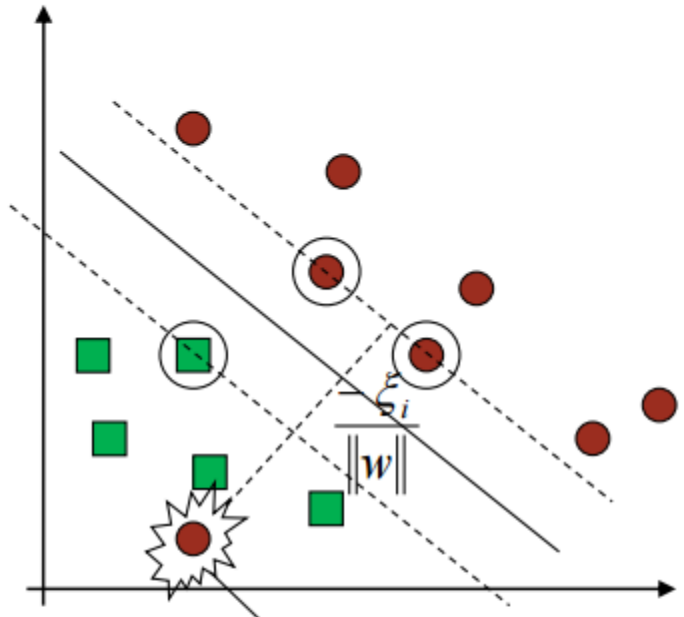
$$\xi_i \geq 0 \quad i = 1,2,\ldots,m$$

- Suma clasificarilor gresite trebuie minimizata impreuna cu maximinarea marginii de separatie:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$C > 0$$

SVM liniare pentru date neseparabile

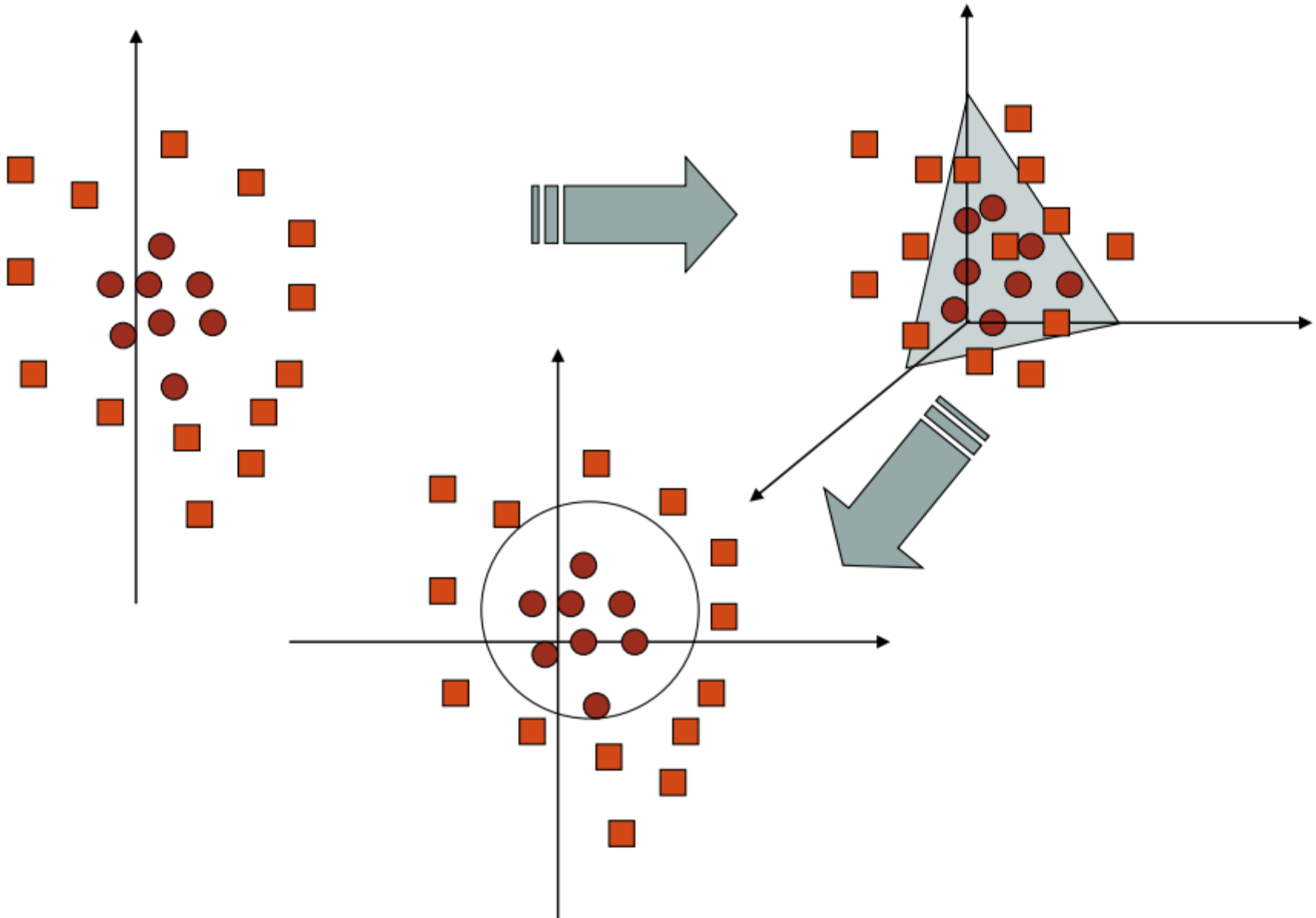


Punct incorect clasificat

reda +1

reda -1

SVM neliniare



Transformarea neliniara

- Kernel – functie care exprima un produs scalar intr-un spatiu al trasaturilor.

- Convergenta masinilor cu suport vectorial catre o solutie impune ca o astfel de functie sa fie pozitiv (semi-)definita.
- Un astfel de kernel este unul care satisface teorema lui Mercer.
- Face imposibila detectarea de suprafete eficiente dar care nu o respecta.

Kernel

- Kernel – functie care exprima un produs scalar intr-un spatiu al trasaturilor.
- Exista kernele clasice pentru care s-a demonstrat respectarea conditiei lui Mercer:

  - Kernelul polinomial: $K(x, y) = (x \cdot y)^p$
  - Kernelul radial: $K(x, y) = e^{\frac{\|x-y\|^2}{\sigma}}$

Rezolvarea problemei de optimizare

- Se bazeaza pe notiuni de convexitate.
- Se construieste Lagrangianul si se aplica conditiile Karush-Kuhn-Tucker-Lagrange (KKTL).
- Se construieste duala problemei de optimizare.
- Se egaleaza gradientul noii functii obiectiv cu 0 si se rezolva sistemul rezultat.
- ***w*** si ***b*** rezulta din conditiile KKTL, daca se pot calcula;
  - daca nu, masina cu suport vectorial va determina direct clasa pentru vectorii de test.
- Semnul lui $f_{w,b}$ da clasa finala – pozitiva sau negativa.

SVM pentru mai multe clase

- In utilizarea standard, SVM este o masina instruibila pentru probleme binare (cu doua clase).

- In extinderea la mai multe clase (*k*), exista doua tehnici clasice:

  - One-against-all (Unul impotriva tuturor)

  - One-against-one (Unu la unu)

One-against-all (1aa)

- Se construiesc k clasificatori SVM.
  - Pentru al i-lea SVM, clasa i este pozitiva si celelalte impreuna reprezinta clasa negativa.

- Fiecare al i-lea SVM determina hiperplanul optim de coeficienti w^i si b^i.

- Eticheta pentru un exemplu nou este data de clasa care are valoarea maxima pentru functia invatata f_{w^i,b^i}.

One-against-one (1a1)

- Se construiesc k(k-1)/2 clasificatoare SVM pentru fiecare doua clase pe rand.
  - Clasa i este clasa pozitiva, iar j cea negativa

- Se determina hiperplanul de decizie intre fiecare doua clase i si j cu coeficientii w^{ij} si b^{ij}.

- Eticheta pentru un exemplu nou se determina prin vot.
  - Pentru fiecare SVM se calculeaza clasa exemplului, care primeste un vot.
  - Castiga clasa cu cele mai multe voturi obtinute.

SVM pentru regresie

- In acest caz, datele sunt constranse sa se afle pe un hiperplan care
  - Permite o anumita eroare ϵ fata de iesirile reale
  - Are abilitate mare de generalizare

- Pentru cazul liniar:

$$\begin{cases} y_i - w \cdot x_i + b \le \varepsilon \\ w \cdot x_i - b - y_i \le \varepsilon \end{cases} \qquad i = 1,2,\ldots,m$$

  - si minimizeaza $\|w\|^2$

Problema de optimizare - regresie

- Pentru cazul general al datelor neseparabile:

$$\begin{cases} y_i - w \cdot x_i + b \le \varepsilon + \xi_i \\ w \cdot x_i - b - y_i \le \varepsilon + \xi_i^* \end{cases} \qquad \xi_i, \xi_i^* \ge 0$$

  - si minimizeaza $\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \qquad C > 0$

- Aplicarea de kernele este similara cu cea in cazul clasificarii.
- Functia $f_{w,b}$ rezultata da iesirea prognozata pentru un nou exemplu.

```
#Import Library
from sklearn import svm
#Assumed you have, X (predictor) and Y (target) for training data set and x_test(pr
edictor) of test_dataset
# Create SVM classification object
model = svm.svc(kernel='linear', c=1, gamma=1)
# there is various option associated with it, like changing kernel, gamma and C val
ue. Will discuss more # about it in next section.Train the model using the training
 sets and check score
model.fit(X, y)
model.score(X, y)
#Predict Output
predicted= model.predict(x_test)
```

Parametrii optimi

```
sklearn.svm.SVC(C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True
, probability=False,tol=0.001, cache_size=200, class_weight=None, verbose=False, ma
x_iter=-1, random_state=None)
```

Kernelul

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
```

```
# import some data to play with
iris = datasets.load_iris()
X = iris.data[:, :2] # we only take the first two features. We could
 # avoid this ugly slicing by using a two-dim dataset
y = iris.target
```

Kernelul

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
```

```
# import some data to play with
iris = datasets.load_iris()
X = iris.data[:, :2] # we only take the first two features. We could
 # avoid this ugly slicing by using a two-dim dataset
y = iris.target
```

Kernelul

```
# we create an instance of SVM and fit out data. We do not scale our
# data since we want to plot the support vectors
C = 1.0 # SVM regularization parameter
svc = svm.SVC(kernel='linear', C=1,gamma=0).fit(X, y)
```

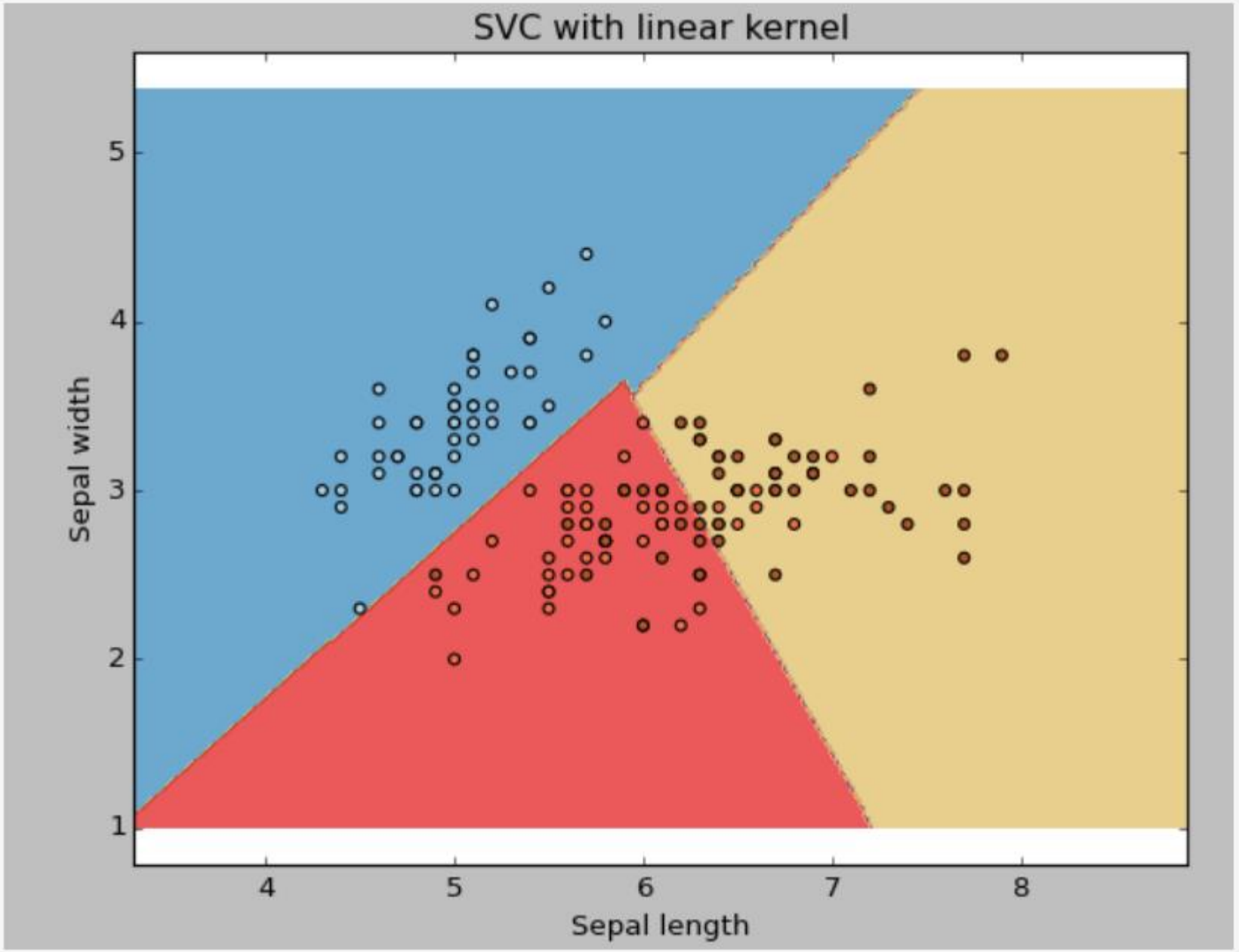
```
# create a mesh to plot in
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
h = (x_max / x_min)/100
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
 np.arange(y_min, y_max, h))
```

Kernelul

```
plt.subplot(1, 1, 1)
Z = svc.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=0.8)
```

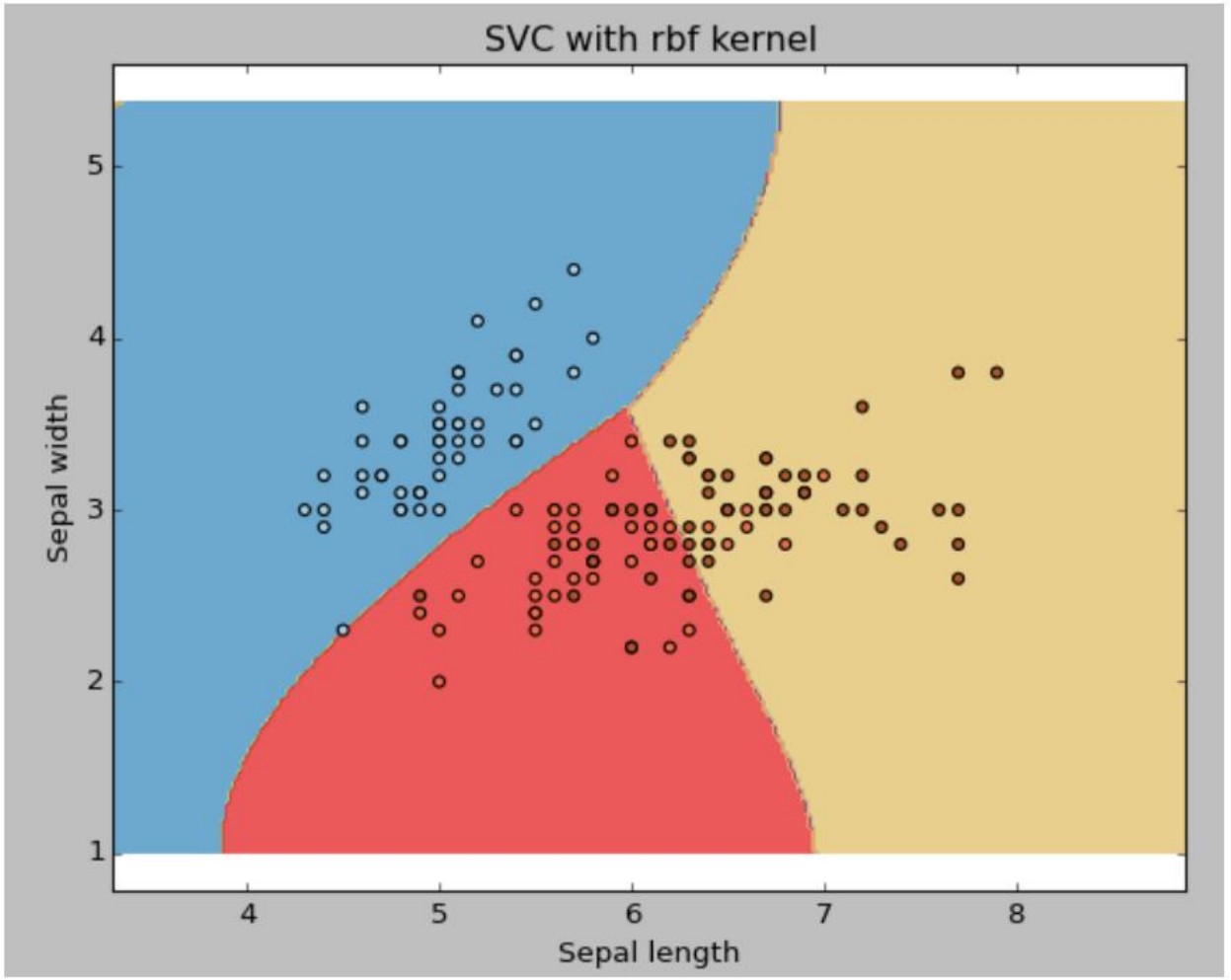
```
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.xlim(xx.min(), xx.max())
plt.title('SVC with linear kernel')
plt.show()
```

Kernelul


SVC with linear kernel
Sepal width
Sepal length
1
2
3
4
5
4
5
6
7
8

Kernelul

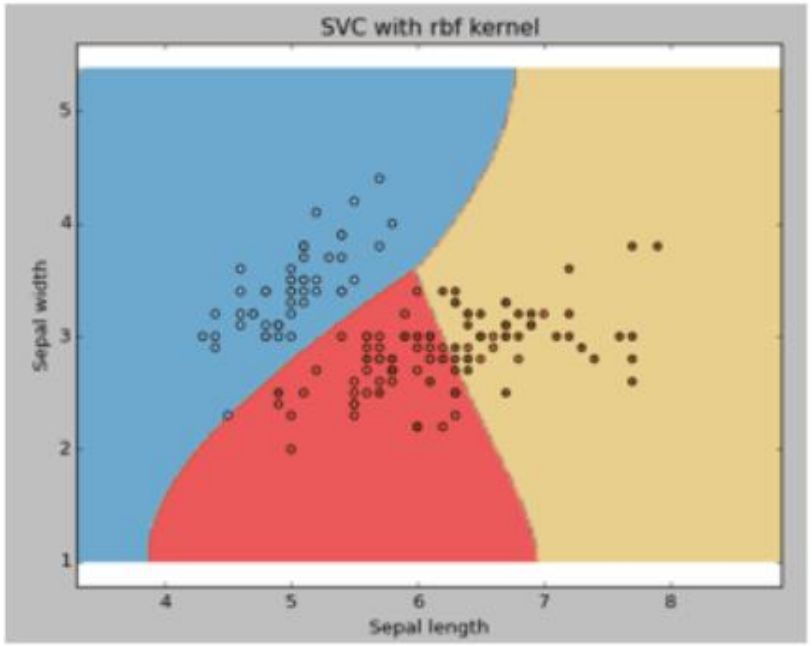
```
svc = svm.SVC(kernel='rbf', C=1,gamma=0).fit(X, y)
```



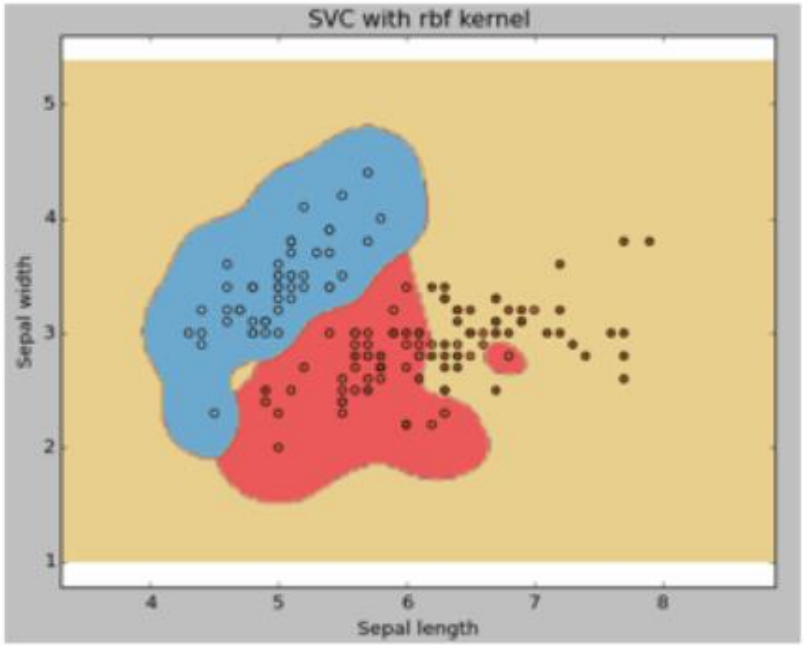
Kernelul

```
svc = svm.SVC(kernel='rbf', C=1,gamma=0).fit(X, y)
```

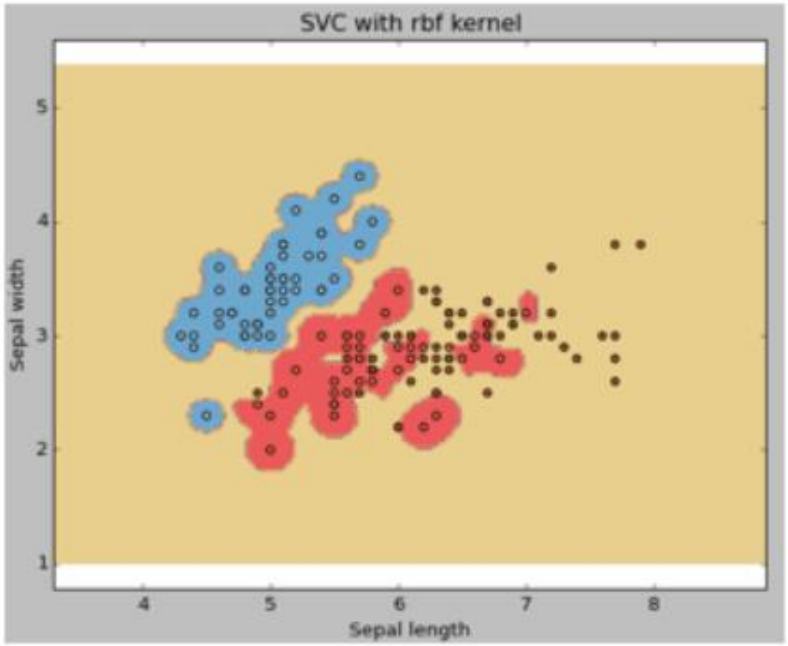
gamma =0



gamma =10

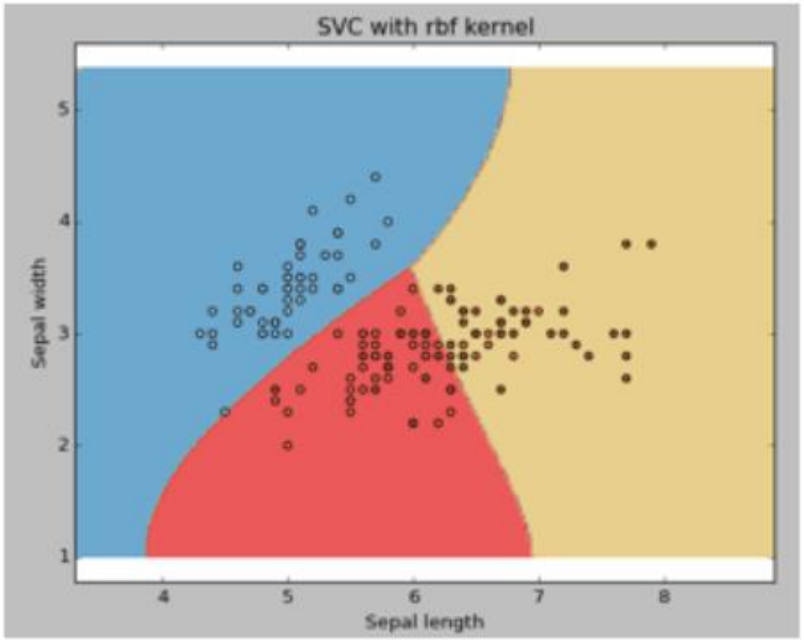


gamma =100

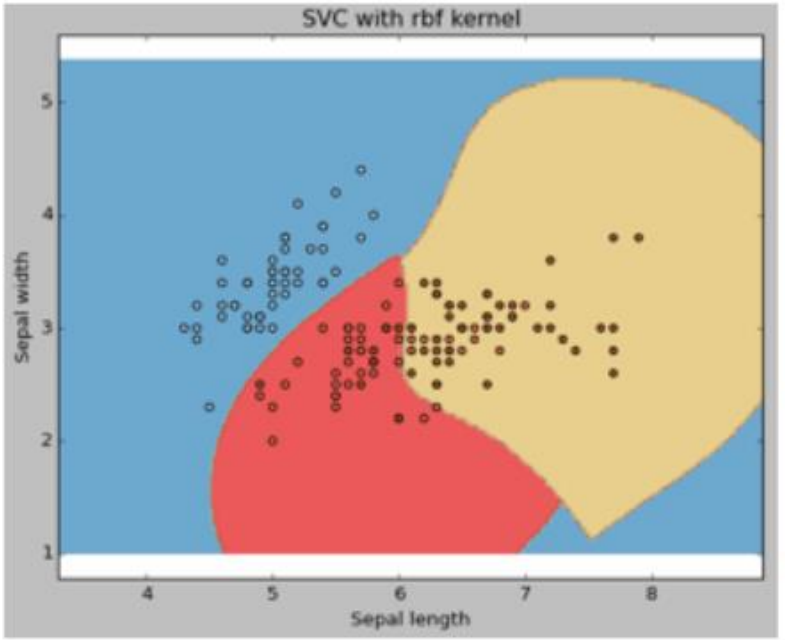


Kernelul

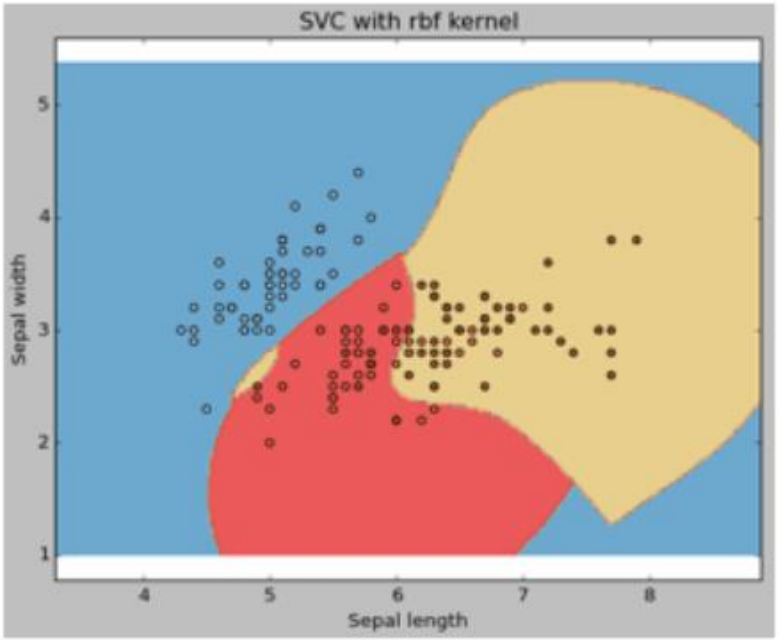
c =1



C =100



c =1000



Bibliografie

- Informaţiile prezentate au fost colectate din diferite surse de pe internet, precum și din slide-urile conf.univ.dr. Ruxandra Stoean Univ. din Craiova, disponibile pe internet.
- https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
- https://medium.com/deep-math-machine-learning-ai/chapter-3-1-svm-from-scratch-in-python-86f93f853dc