



DigitalHouse >
Coding School

DATA SCIENCE

UNIDAD 3
MÓDULO 4

Support Vector Machines

Octubre 2017

1

Entender el funcionamiento del algoritmo Support Vector Machine a bajo nivel

2

Entender en qué casos se utiliza

3

Implementar el modelo y evaluar la performance

Support Vector Machine



Naive Bayes → **Modelo Generativo**: Describe el proceso generador de cada clase

SVM → **Modelo Discriminante**: Busca trazar una recta o curva divisoria en el hiperespacio.

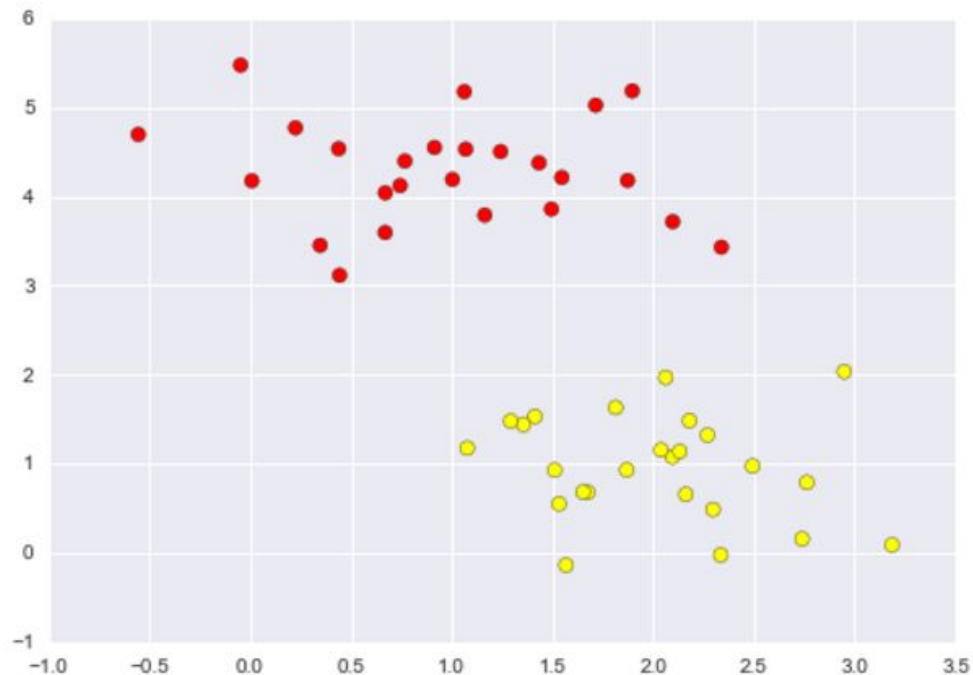
SVM es un algoritmo que sirve para resolver tanto problemas de clasificación como regresión.

La idea original surge en 1963 y era una aproximación muy simple a la clasificación de datos a través de separaciones lineales entre las clases.

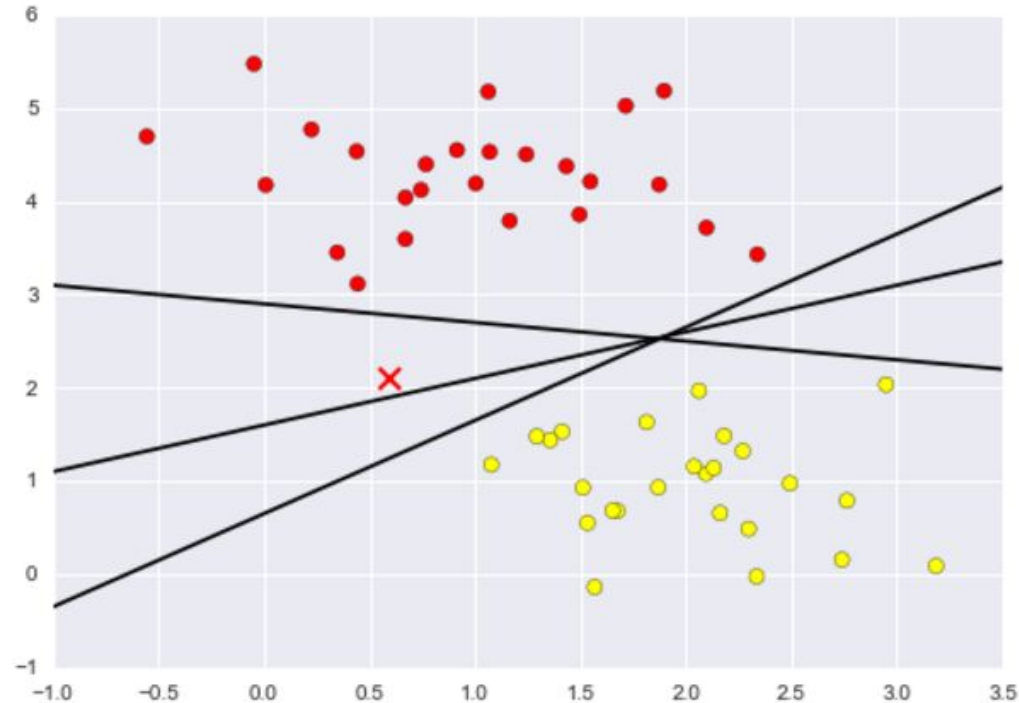
La popularidad de SVM, recién se produce en la década del 90 de la mano de dos innovaciones clave:

- El “kernel trick”
- Los “soft margins”

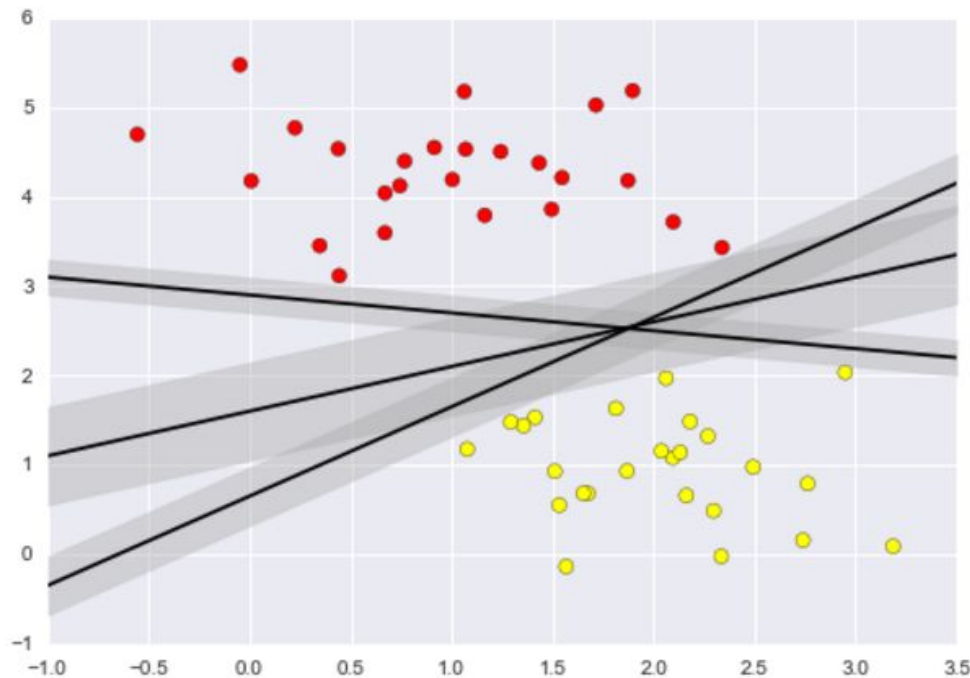
Supongamos que queremos clasificar el siguiente dataset ¿Cuál es la mejor forma de hacerlo?



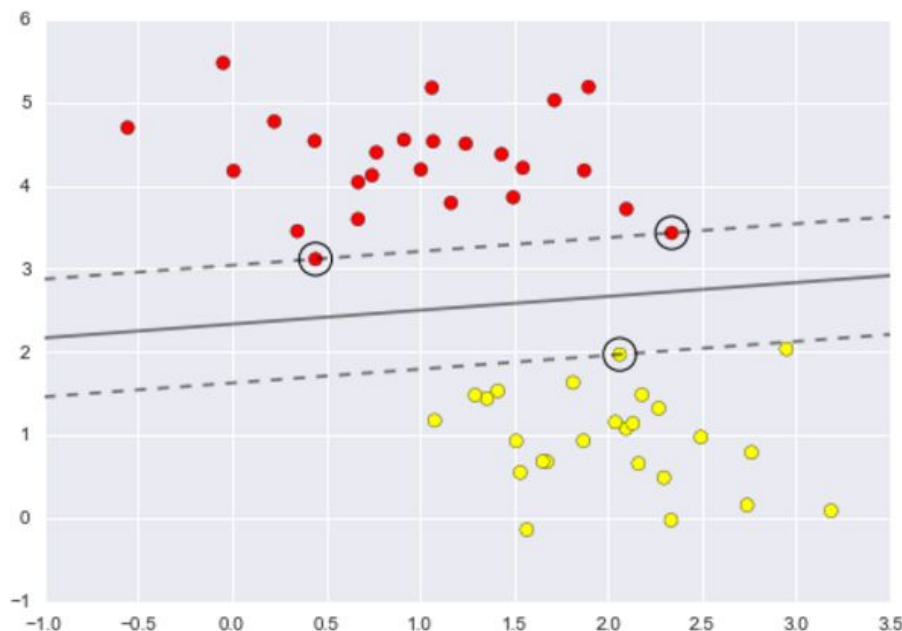
Existen muchas posibles líneas que podríamos trazar para dividir las dos clases ¿Cuál sería la mejor? Dependiendo la línea que se elija, una nueva observación puede quedar en una u otra clase.



Imaginen que trazamos un “margen” entre la línea discriminante y el punto más cercano de cada clase. Lo que hace el algoritmo SVM es encontrar la línea discriminante que **maximiza el ancho de este “margen”**.



Cuando ajustamos el modelo SVM a estos datos obtenemos la línea discriminante que se encuentra en el medio. Las líneas punteadas representan los límites del margen.

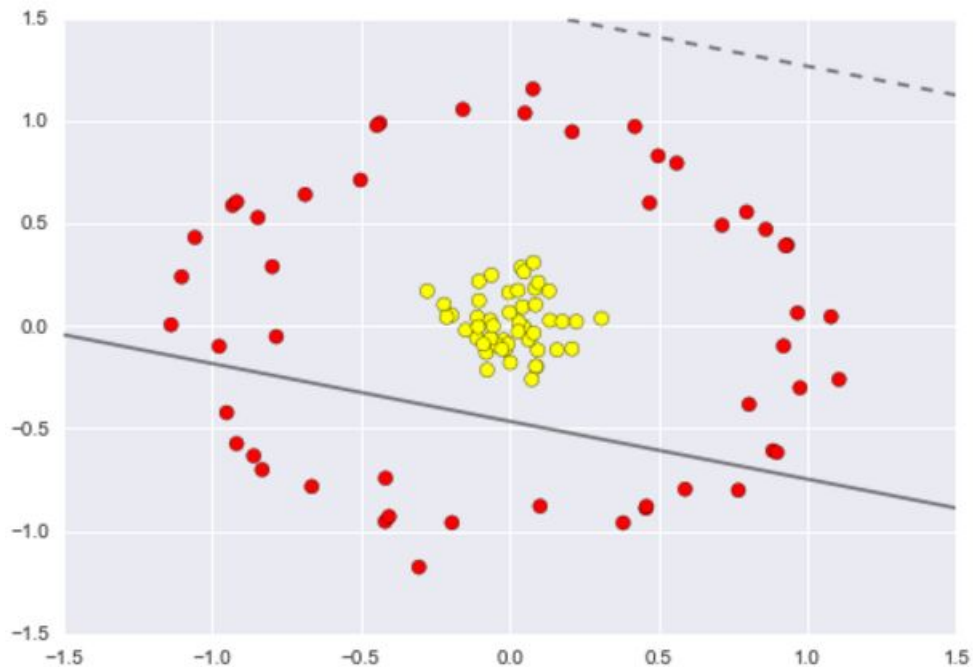


Es importante notar que solamente algunos de los puntos tocan los límites del margen. Estos puntos son los “**vectores de soporte**”. A la hora de hacer el ajuste del modelo, los únicos puntos que importan son estos. *El comportamiento de los que se encuentran lejos del “decision boundary” no tiene ninguna importancia.*

Cuando estudiamos regresión lineal, vimos la posibilidad de hacer transformaciones sobre los datos originales para ser capaces de modelar una relación no lineal entre los predictores y la respuesta con un modelo lineal.

Aquí vamos a hacer transformaciones sobre los datos para crear un espacio de más dimensiones donde el modelo SVM sí pueda discriminar.

Si tenemos un conjunto de datos como el siguiente, ningún modelo de clasificación lineal va a funcionar correctamente ¿Cómo podríamos aumentar la dimensión de los datos para que un clasificador lineal funcione?



Lo que podemos hacer es aplicar una “transformación de base radial”, donde generamos una nueva feature r , a partir de los valores del vector X formado por x_1 y x_2 .

```
r = np.exp(-(X ** 2).sum(1))
```

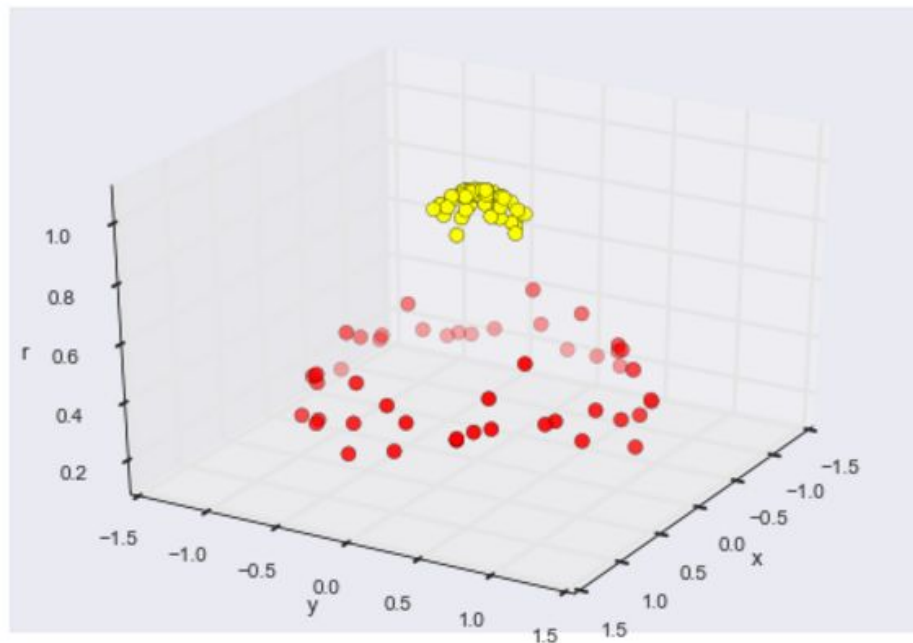
$$f(x) = \sum_{j=1}^m w_j h_j(x)$$

Esta es la fórmula general de una transformación radial.

$$h(x) = \exp\left(-\frac{(x-c)^2}{r^2}\right)$$

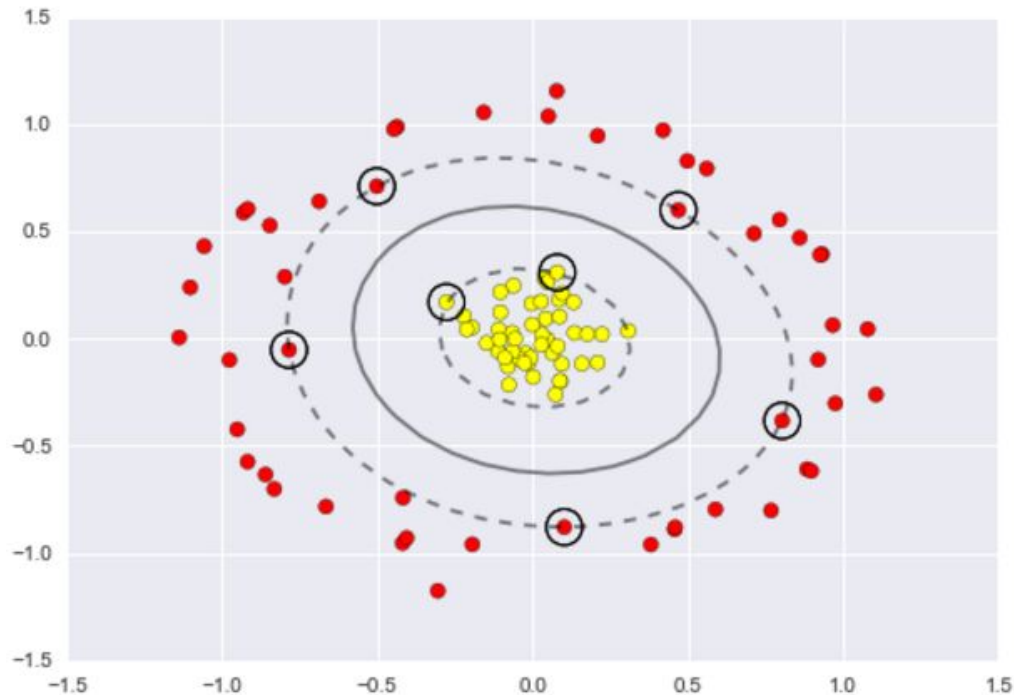
En nuestro caso los datos están centrados tienen radio 1, por lo tanto c (el centroide) vale 0 y r vale 1.

El resultado de agregar la dimensión "r" es el siguiente. Ahora sí podemos encontrar un plano que haga una buena discriminación lineal.

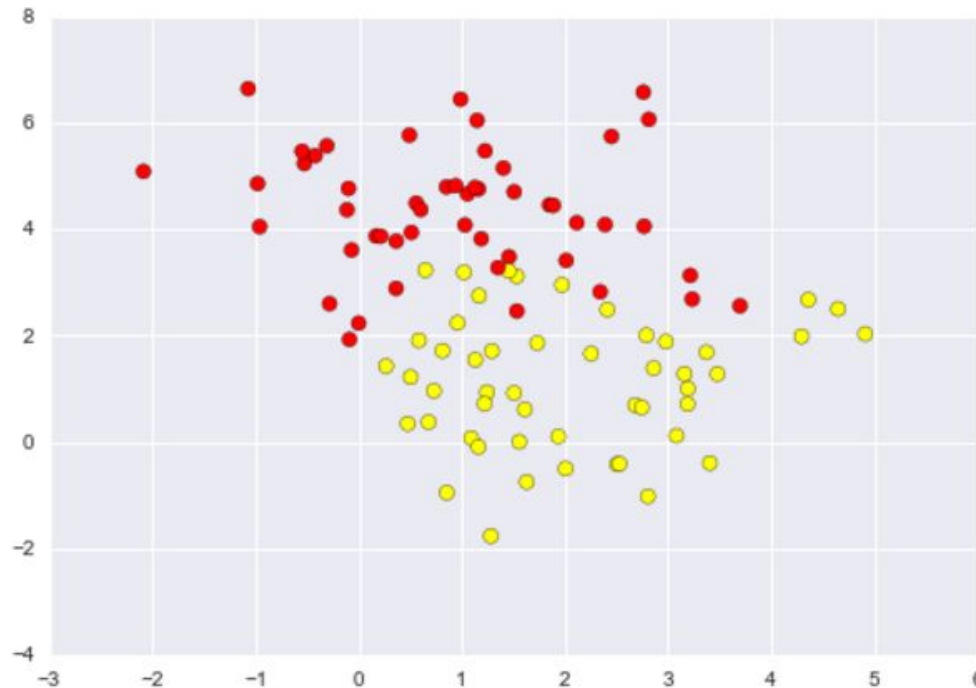


Sklearn tiene implementados distintos tipos de transformación kernel, no es necesario hacerlo manualmente.

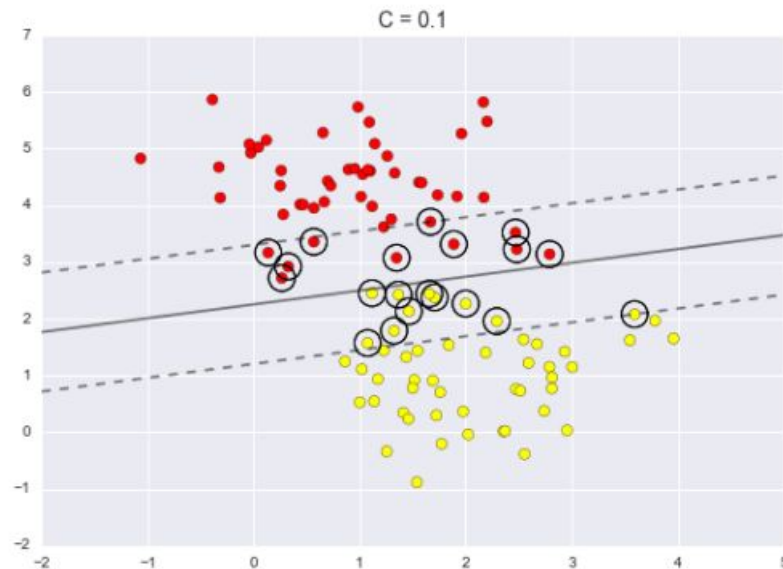
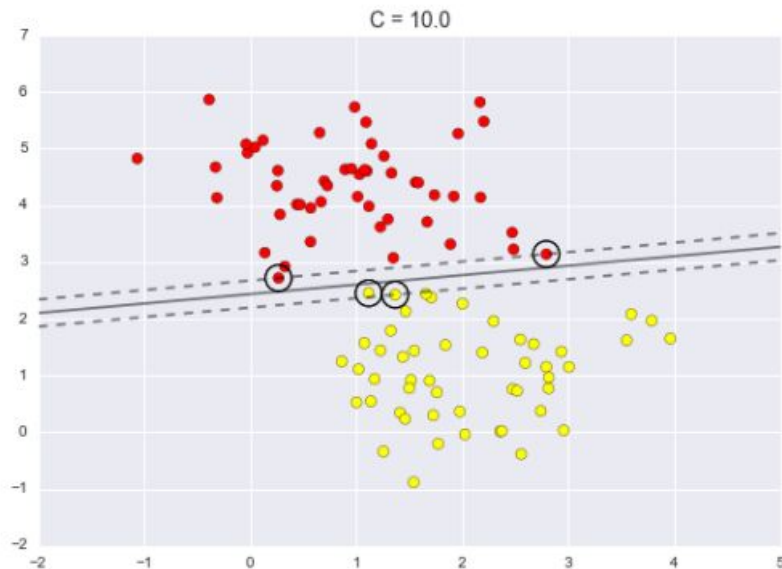
```
clf = SVC(kernel='rbf', C=1E6)  
clf.fit(X, y)
```



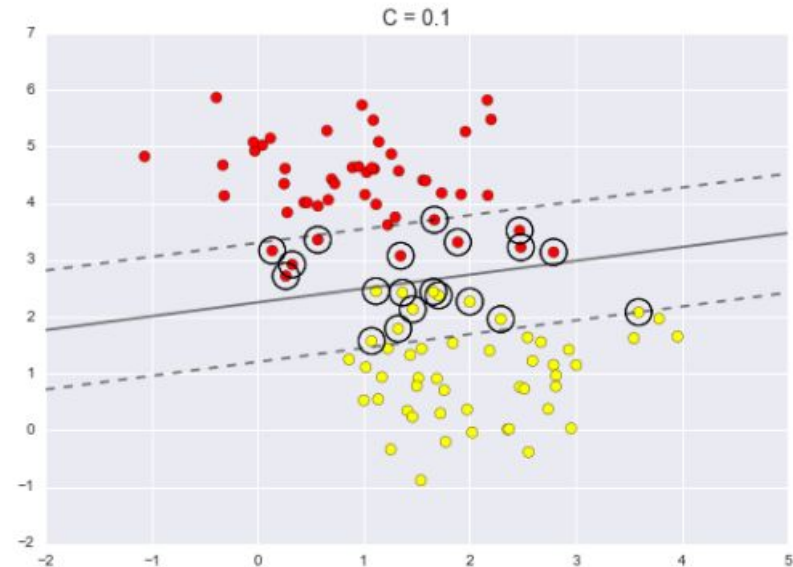
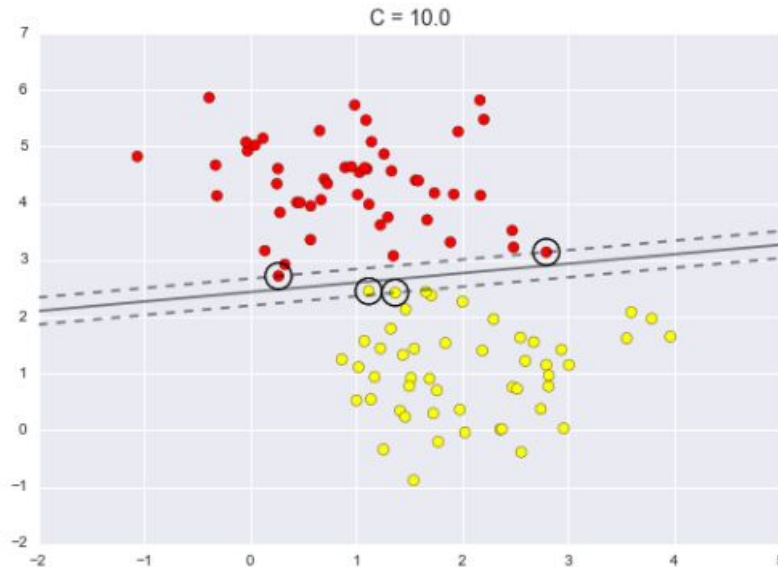
Todos los métodos que vimos hasta ahora funcionan en la medida que los datos se puedan discriminar linealmente de forma perfecta ¿Qué pasa si no se puede hacer una división perfecta?



Para resolver este problema, el algoritmo incorpora el parámetro C que permite que algunos puntos permanezcan dentro del margen para mejorar el ajuste.

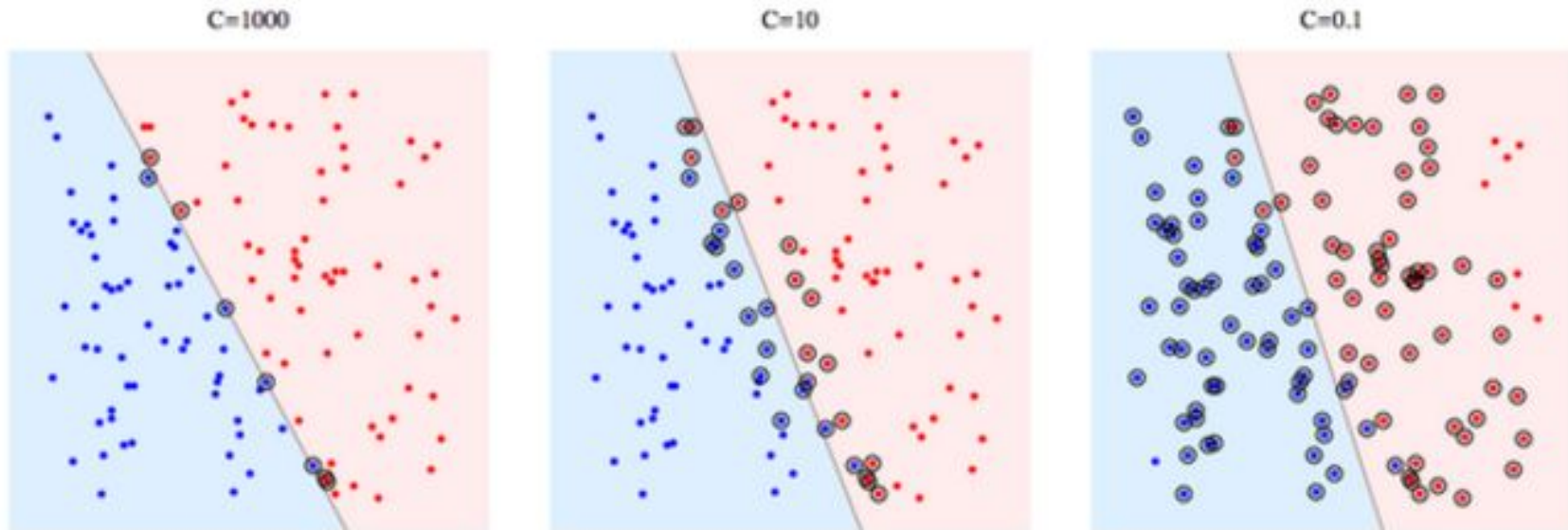


Para resolver este problema, el algoritmo incorpora el parámetro C que permite que algunos puntos permanezcan dentro del margen para mejorar el ajuste. El parámetro C le indica a SVM cuánto se penaliza una mala clasificación. Valores más altos de C tendrán márgenes más angostos.



El hiperparámetro C es uno los que regulariza el trade-off entre sesgo y varianza.

La forma de hallar el óptimo para nuestro conjunto de datos es con **cross-validation**.



Práctica Guiada SVM

Lab SVM

- SVM es un método de clasificación y regresión.
- EL hiperparámetro C es el que regulariza la complejidad del modelo

Ventajas de Support Vector Machine:

- Es ideal para espacios de muchas dimensiones (muchos vectores en la matriz de features)
- Sigue siendo aplicable cuando hay más features que datos
- Versátil, ya que soporta distintos kernels que pueden funcionar sobre el conjunto de datos.

Desventajas de Support Vector Machine:

- Es posible caer en el overfitting, si se elige mal la transformación kernel o los hiperparámetros.
- Son computacionalmente muy costosos
- No hace estimaciones probabilísticas directamente. Hacerlo es computacionalmente costoso.