




Aprendizaje Supervisado

Matías Marenchino - Cristian Cardellino





Segunda Clase



Temario de la Clase

- ¿Qué es aprendizaje supervisado?
- Repaso general de introducción al aprendizaje automático.
 - Regresión Lineal y Polinomial, Regresión Logística, Perceptrón.
- Árboles de decisión
- Naive Bayes
- **Support Vector Machines**
- Ensemble learning.
 - Random Forest, Bagging, Boosting.
- Redes neuronales.
 - Perceptrón multicapa.
- Sistemas de recomendación.
 - Filtrado colaborativo, máquinas de factorización.
- Prácticas de reproducibilidad

Resumen de la clase anterior

Repaso de lo visto

Aprendizaje supervisado busca aprender un modelo matemático a partir de datos anotados con ciertas etiquetas.

Si la etiqueta es un número real, entonces estamos ante un problema de regresión.

Si la etiqueta está en una categoría, entonces es un problema de clasificación.

Repaso de lo visto: Regresión

Busca modelar una función $f(x) \rightarrow y$ que devuelva un valor real.

Se utilizan en tareas como predicción de precios, valuación de activos (e.g. en bolsa), predicción de temperatura, etc.

Los algoritmos que vimos son: regresión lineal, regresión polinomial y árboles de decisión.

Repaso de lo visto: Clasificación

Busca modelar una función $f(x) \rightarrow y$ que devuelva un valor que pueda ser utilizado para clasificar algo entre varias opciones (e.g. una probabilidad).

Se utilizan en tareas como identificación de correo basura, clasificación de imágenes, análisis de sentimiento en oraciones.

Los algoritmos que vimos son: regresión logística, árboles de decisión, naive bayes y el algoritmo del perceptrón.

Support Vector Machines

Fronteras de decisión en clasificación

Un clasificador busca **separar los datos** de una y otra clase de la mejor manera.

Esta separación se da mediante una **frontera de decisión**.

¿Qué determina que tan “buena” es una frontera de decisión?

¿Qué es una “buena” frontera de decisión?

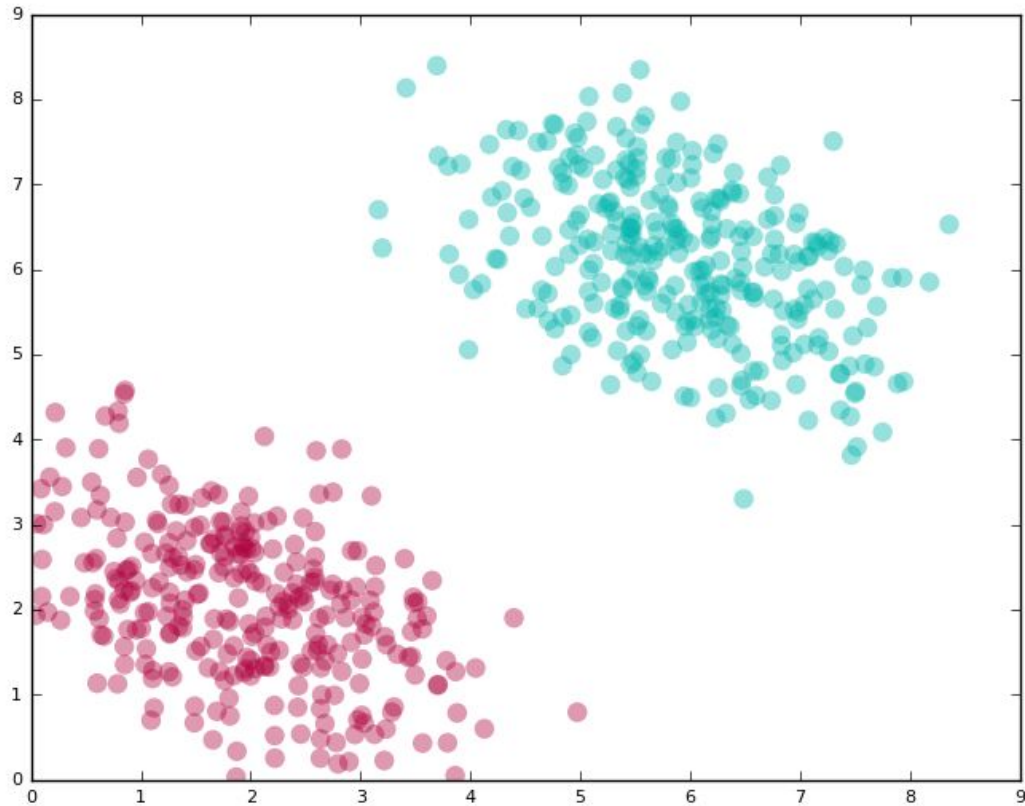
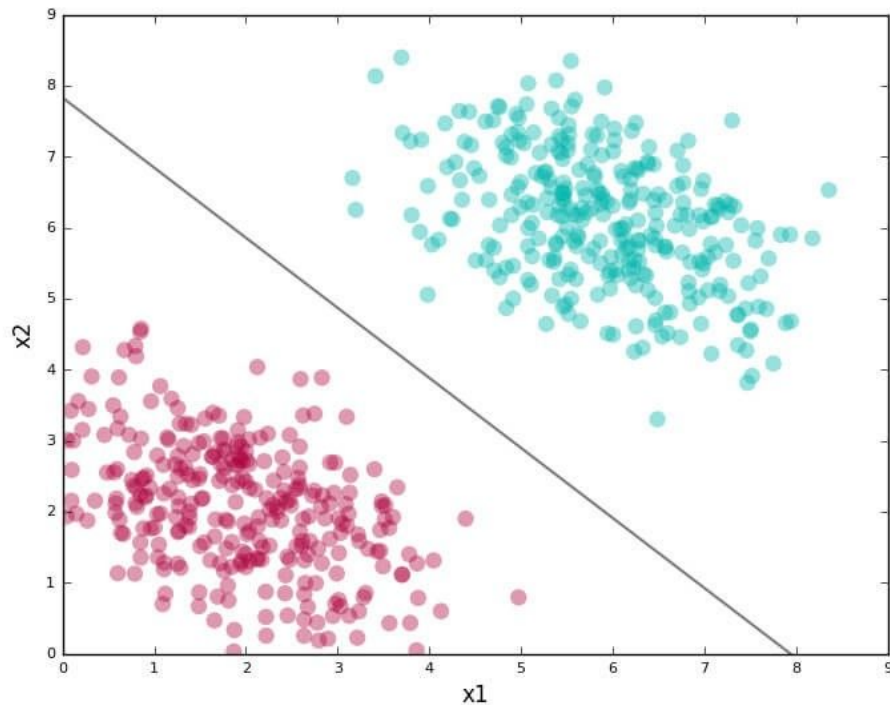
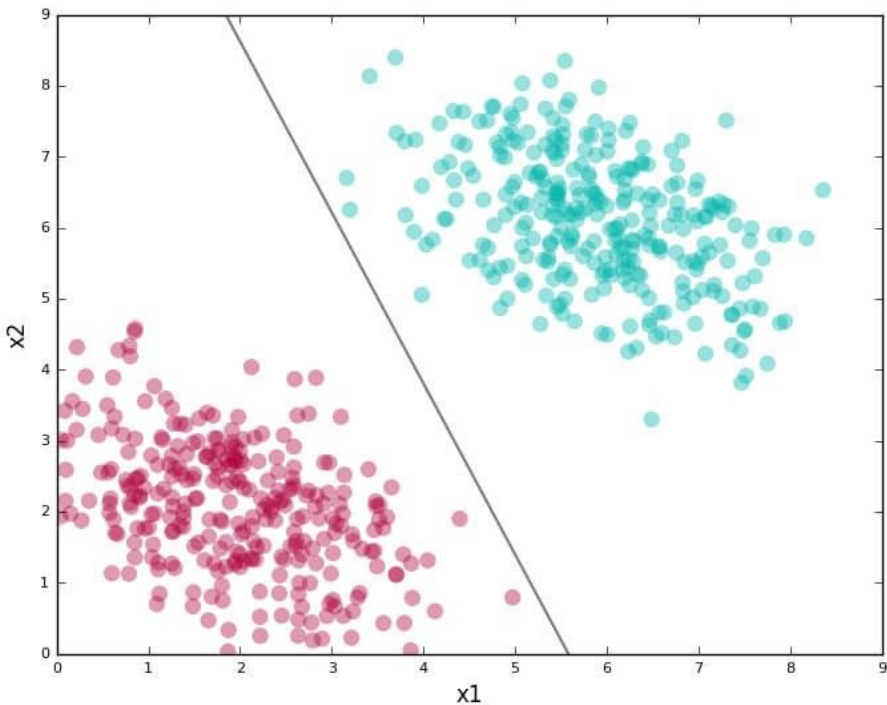


Image from <https://blog.statsbot.co/>

¿Qué es una “buena” frontera de decisión?



Margen de la frontera

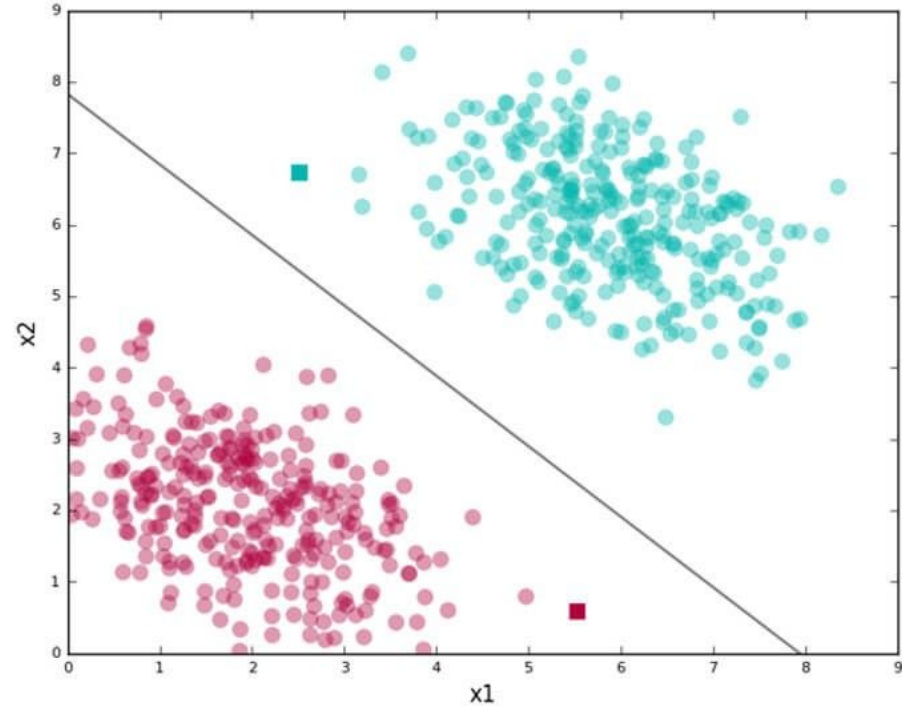
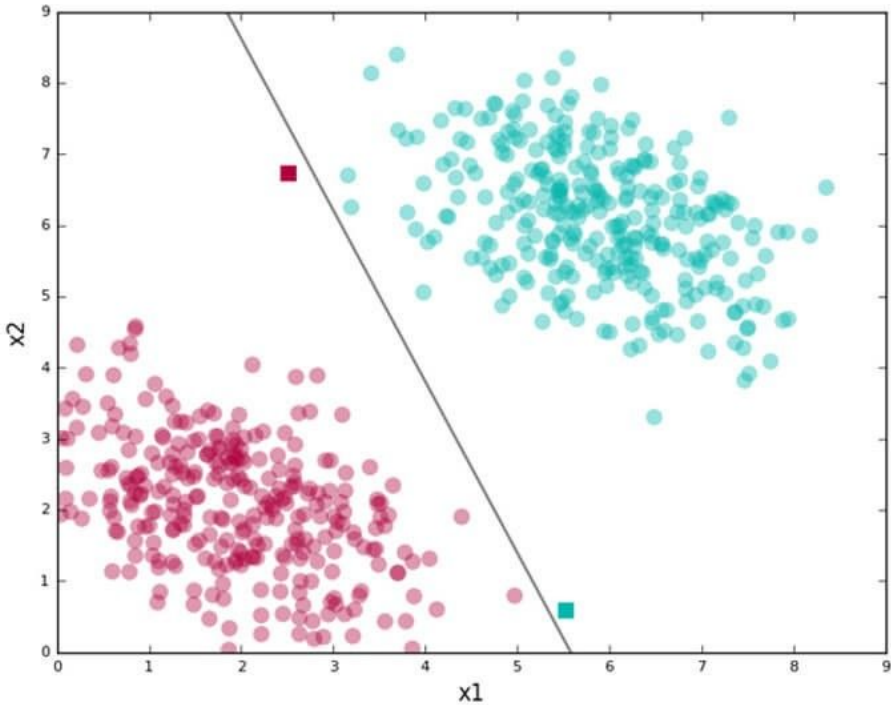
En el gráfico anterior, cualquiera de las líneas separan los datos correctamente.

Buscamos una línea que capture el patrón general entre los datos.

En el gráfico de la izquierda, la línea de separación está algo sesgada. Tiene menos margen entre ella y ambos clústeres de datos.

La línea en el gráfico de la derecha, en cambio, se encuentra bien a la mitad de ambos clústeres.

¿Qué pasa al clasificar nuevos datos?



Support Vector Machines

Es un algoritmo que busca separar los datos mediante la mejor frontera de decisión. Esta frontera de decisión es conocida como **hiperplano**.

En este caso, “mejor” se refiere a aquella que esté lo más separada posible de los puntos más cercanos a ella. Estos puntos son conocidos como **vectores de soporte**, y el espacio entre ellos y el hiperplano se conoce como **margen**.

En términos más técnicos, un algoritmo de SVM encuentra el hiperplano que devuelva el mayor margen entre sí mismo y los vectores de soporte.

Este tipo de clasificador a veces es conocido como “clasificador por márgenes” (margin classifier).

Support Vector Machines

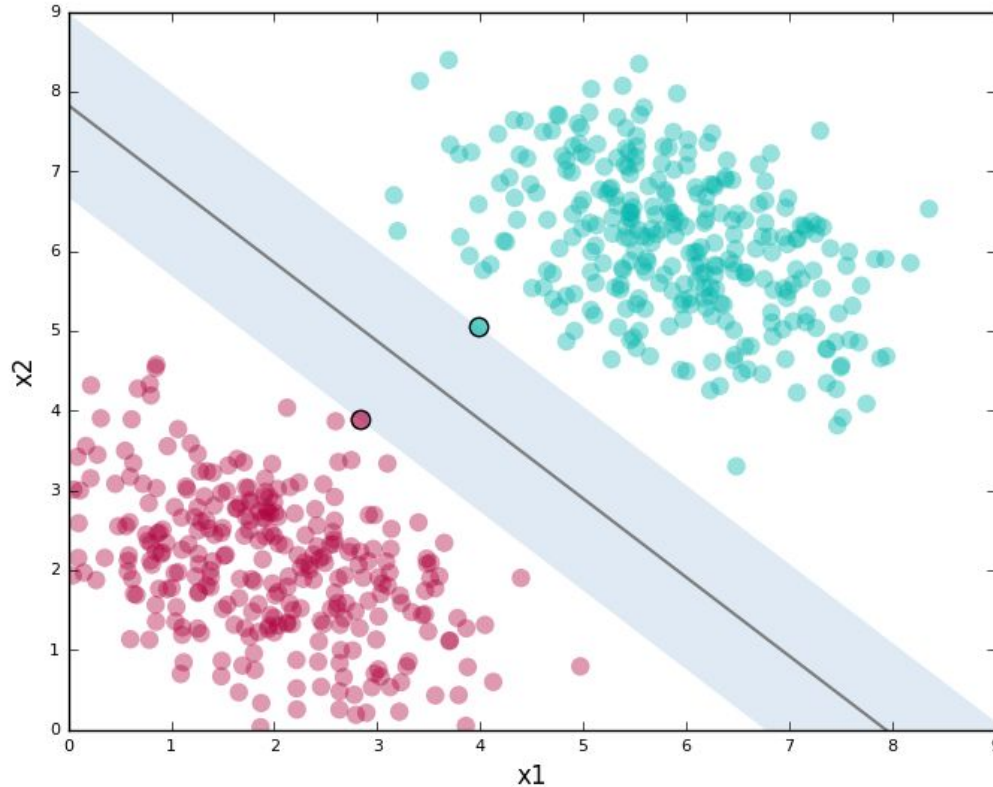


Image from <https://blog.statsbot.co/>

SVM: Función de Costo y Entrenamiento

SVM: Función de costo

Los SVM utilizan una función de costo conocida como **hinge loss**.

A diferencia de regresión logística, los datos se anotan con $\{-1, 1\}$ de acuerdo al valor de la etiqueta.

La función de costo de Hinge se define como:

$$c(x, y, f(x)) = \max(0, 1 - y * f(x))$$

Donde el costo es 0 si el valor real y el predicho tienen el mismo signo y están dentro del margen de error (por lo general 1).

SVM: Función a optimizar

La función que buscamos minimizar es la siguiente:

$$\min_w \sum_{i=1}^n \max(0, 1 - y_i \langle x_i, w \rangle) + \lambda ||w||^2$$

Dónde $\lambda/||w||^2$ es el parámetro de regularización.

SVM: Gradientes

Tenemos dos factores en la función de costo que hay que derivar:

$$\frac{\delta}{\delta w_k} \lambda ||w||^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \max(0, 1 - y_i \langle x_i, w \rangle) = \begin{cases} 0 & \text{si } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik} & \text{c.c.} \end{cases}$$

SVM: Actualización de los pesos

Al actualizar los pesos, de acuerdo al signo de la predicción, tendremos para el caso donde el signo sea el mismo:

$$w = w - \alpha(2\lambda w)$$

Mientras que cuando el signo entre la predicción y el valor real es diferente:

$$w = w + \alpha(y_i x_i - 2\lambda w)$$

SVM con outliers

SVM: Outliers

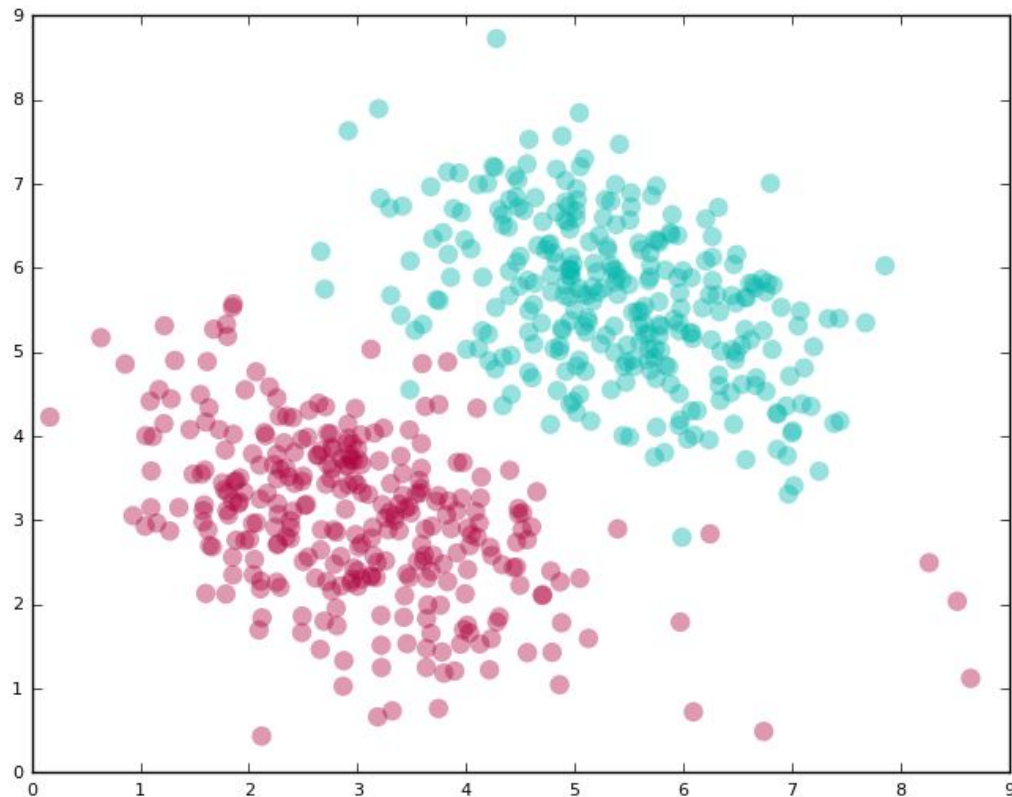


Image from <https://blog.statsbot.co/>

SVM: Outliers

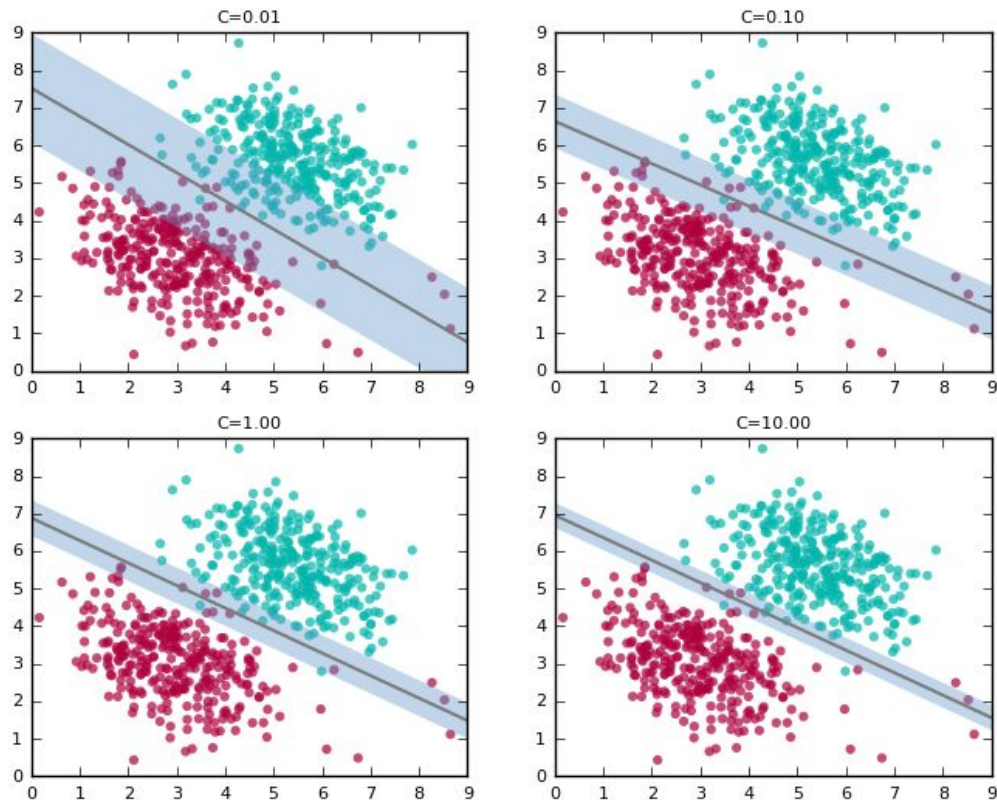
La mayoría de los casos, los datos no son linealmente separables.

En algunos casos, existen outliers.

Hay un parámetro que define qué tan tolerante puede ser SVM sobre la clasificación incorrecta de datos.

El “parámetro C ”, define un tradeoff entre clasificar mejor los datos de entrenamiento y tener una mejor “separación” (un margen más amplio).

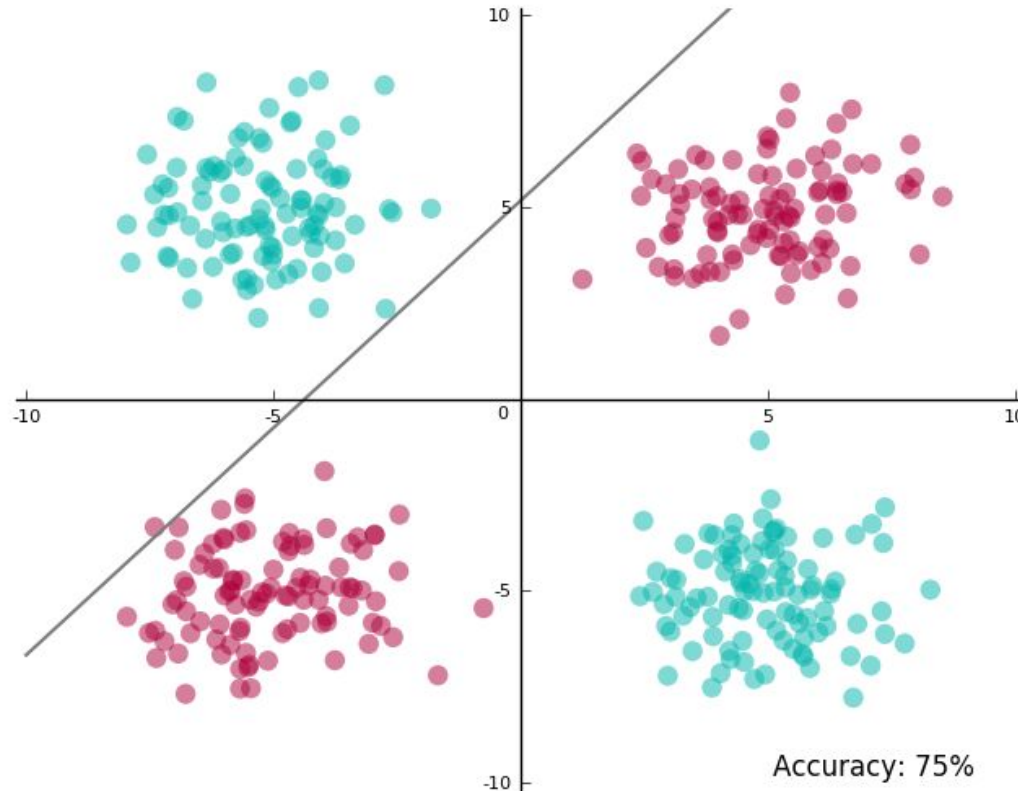
SVM: Parámetro C



Demo Time (demo5)

SVM con datos no linealmente separables

Datos no linealmente separables



¿Qué hacer con datos no linealmente separables?

SVM es una técnica para separar los datos mediante un hiperplano.

Si los datos no son linealmente separables, dicho hiperplano no existe.

Solución: Proyectar los datos a una dimensión donde sí sean linealmente separables.

En el ejemplo anterior, tomamos el conjunto de datos en dos dimensiones, y lo proyectamos a tres dimensiones con la siguiente ecuación:

$$\begin{aligned}X_1 &= x_1^2 \\X_2 &= x_2^2 \\X_3 &= \sqrt{2}x_1x_2\end{aligned}$$

¿Cómo se ve el plano proyectado?



Image from <https://blog.statsbot.co/>

¿Cómo se ve el plano proyectado?

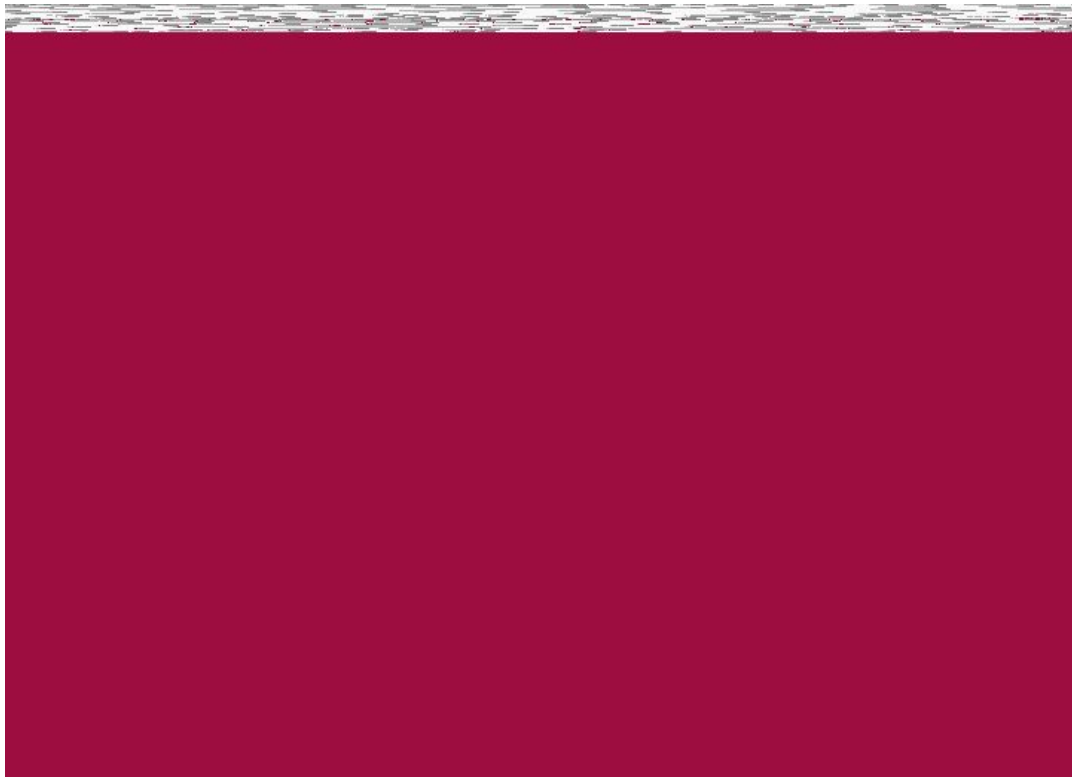
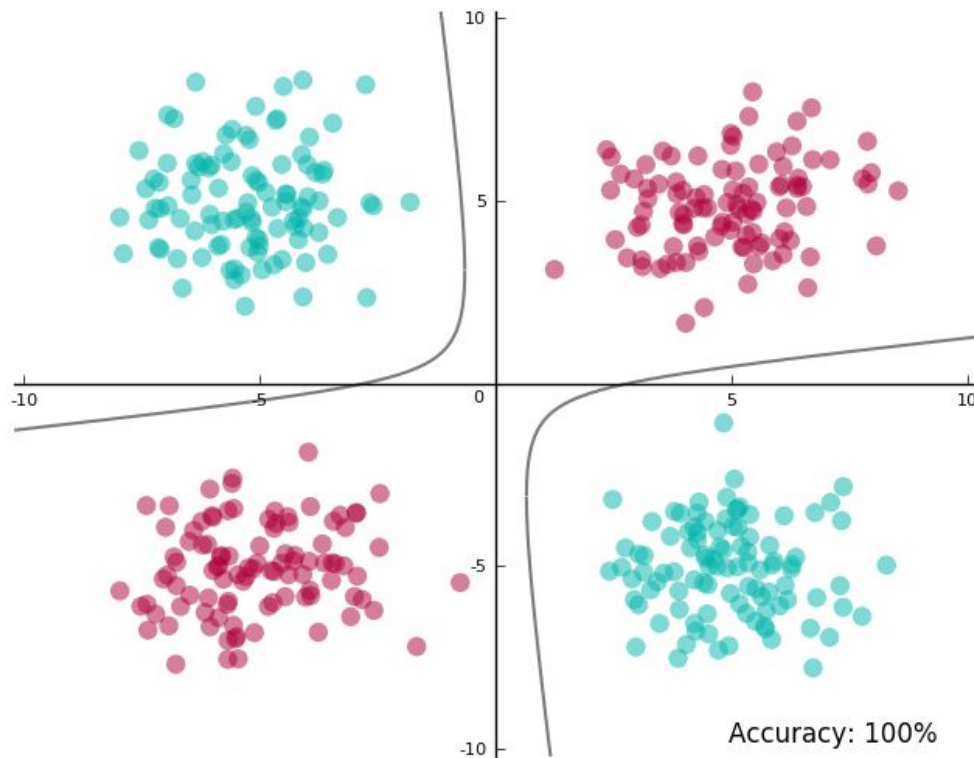


Image from <https://blog.statsbot.co/>

¿Y en 2 dimensiones?



SVM: Kernels

La manera en que el algoritmo de SVM realiza la proyección es mediante el uso de kernels.

Las funciones de kernel toma dos puntos del espacio original, y devuelve el producto punto en el espacio proyectado.

Este producto punto es lo que la función de SVM necesita para calcular el costo.

En el ejemplo anterior, el kernel es: $K(x_i, x_j) = \langle x_i, x_j \rangle^2$

¿Cómo elegir el kernel?

Este no es un problema trivial. Requiere mucho conocimiento matemático encontrar la proyección correcta.

En general, los frameworks más utilizados para hacer SVM tienen algunos kernels bastante comunes:

- Polinomial: $K(x, z) = (\langle x, z \rangle + c)^d$
- Radial Basis Functions (RBF): $K(x, z) = \exp(-(x - z)^2 / 2\sigma^2)$
- Sigmoid: $K(x, z) = \tanh(c \langle x, z \rangle + h)$

Support Vector Regression

Se basa en la idea de SVMs de buscar los vectores de soporte, pero en este caso el valor de y_i es un número real.

Utiliza necesariamente “márgenes blandos”, requiere un parámetro adicional ε para calcular la función de costo.

En general la regresión lineal es más popular, pero con el uso de kernels, se pueden lograr regresiones no lineales muy interesantes.

Demo Time (demo6)

Kaggle Time!



Fin de la segunda clase

