

UNIVERSIDAD
NACIONAL
DE COLOMBIA



CLASIFICACIÓN Y RECONOCIMIENTO DE PATRONES

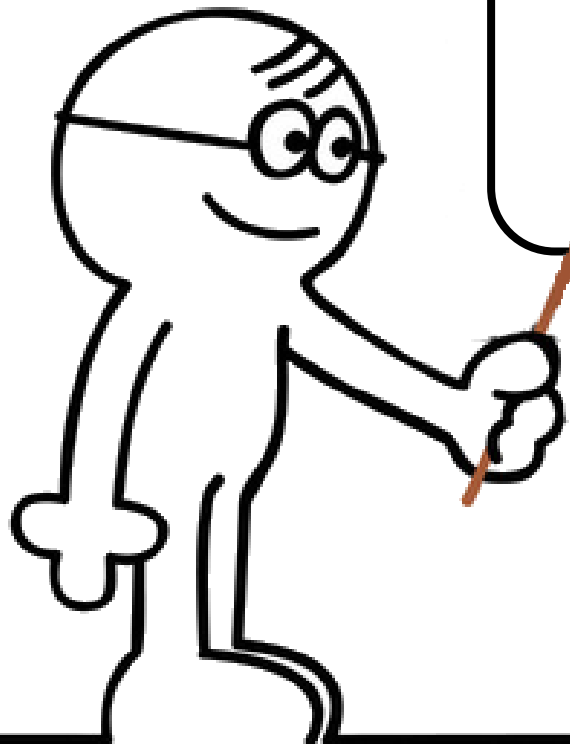
Carlos Mera
John Willian Branch

Departamento de Ciencias de la Computación y de la Decisión
Investigador del Grupo de I+D en Inteligencia Artificial – GIDIA

camerab@unal.edu.co

Contenido

1. Motivación
2. Métodos de Aprendizaje de Máquina:
 - a. Métodos Supervisados
 - b. Métodos No Supervisado
3. Regresión Lineal
4. Regresión Logística
5. KNN
6. Naïve Bayes
7. LDA y QDA
8. Máquinas de Vectores de Soporte



¿QUÉ ES LA REGRESIÓN LOGÍSTICA?

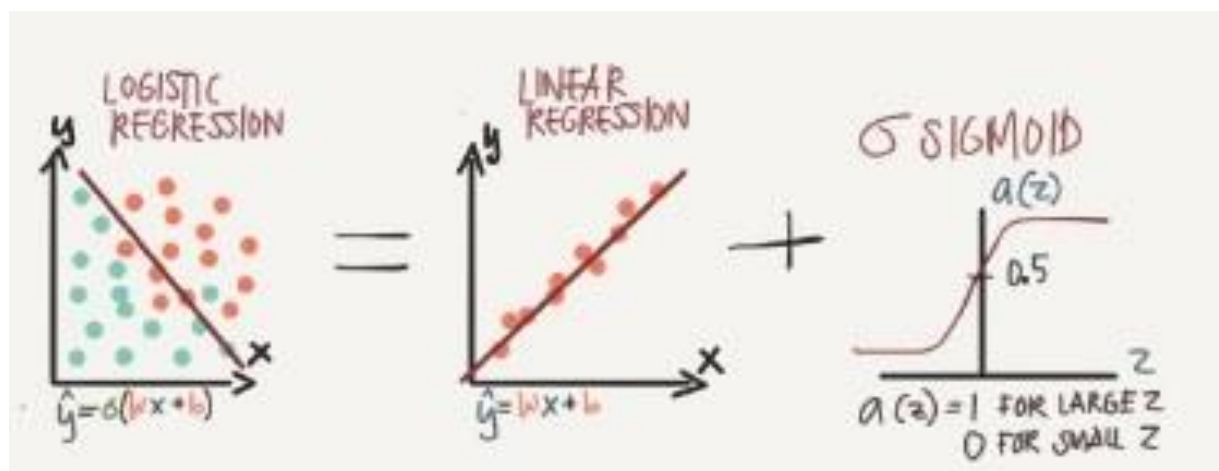
Regresión Logística

DEFINICIÓN:

La regresión logística es un modelo de clasificación que se utiliza para **predecir la probabilidad** $P(y = 1)$ **de una variable dependiente categórica** en función de x . Así, la variable y es una variable binaria codificada como 1 (positivo, éxito, etc.) o 0 (negativo, falla, etc.).

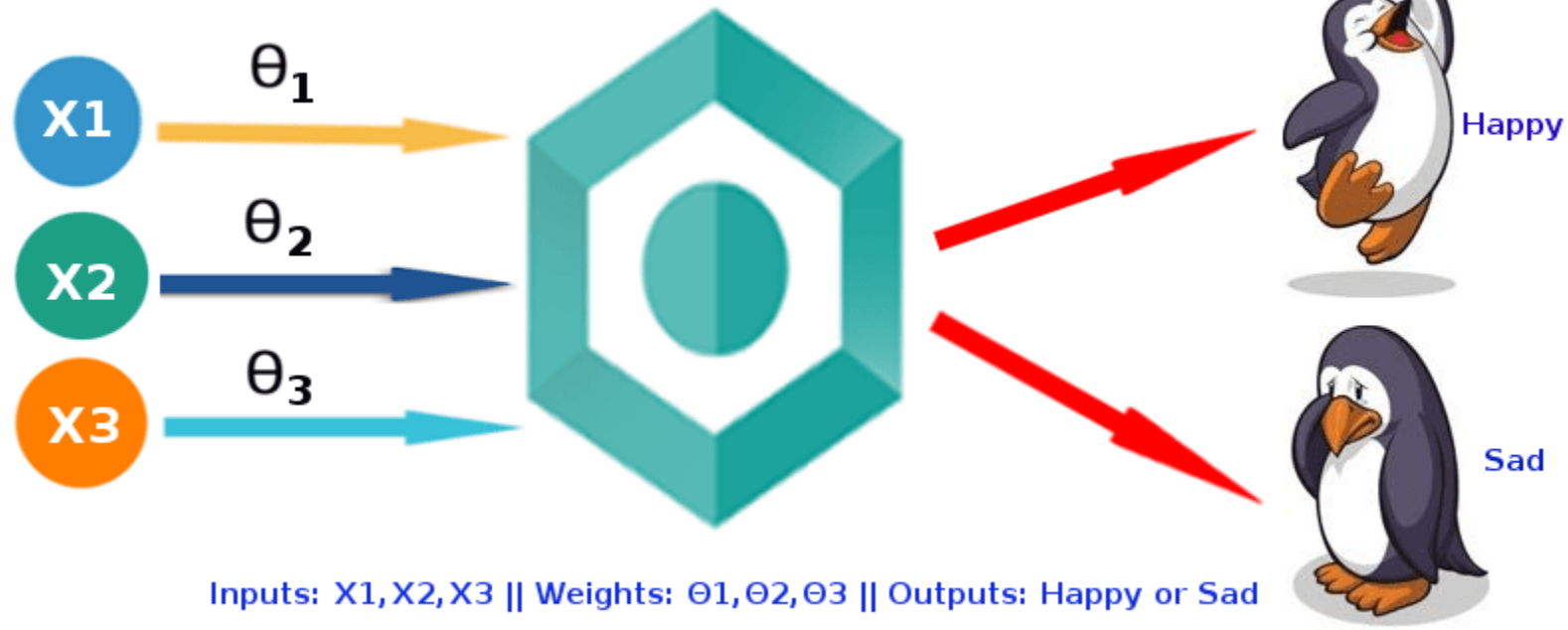
Algunos ejemplos de aplicación:

- E-mail: spam/no spam
- Transacciones en línea: fraude/no fraude
- Tumores: maligno/no maligno



Regresión Logística

Logistic Regression Model



@dataaspirant.com

Regresión Logística

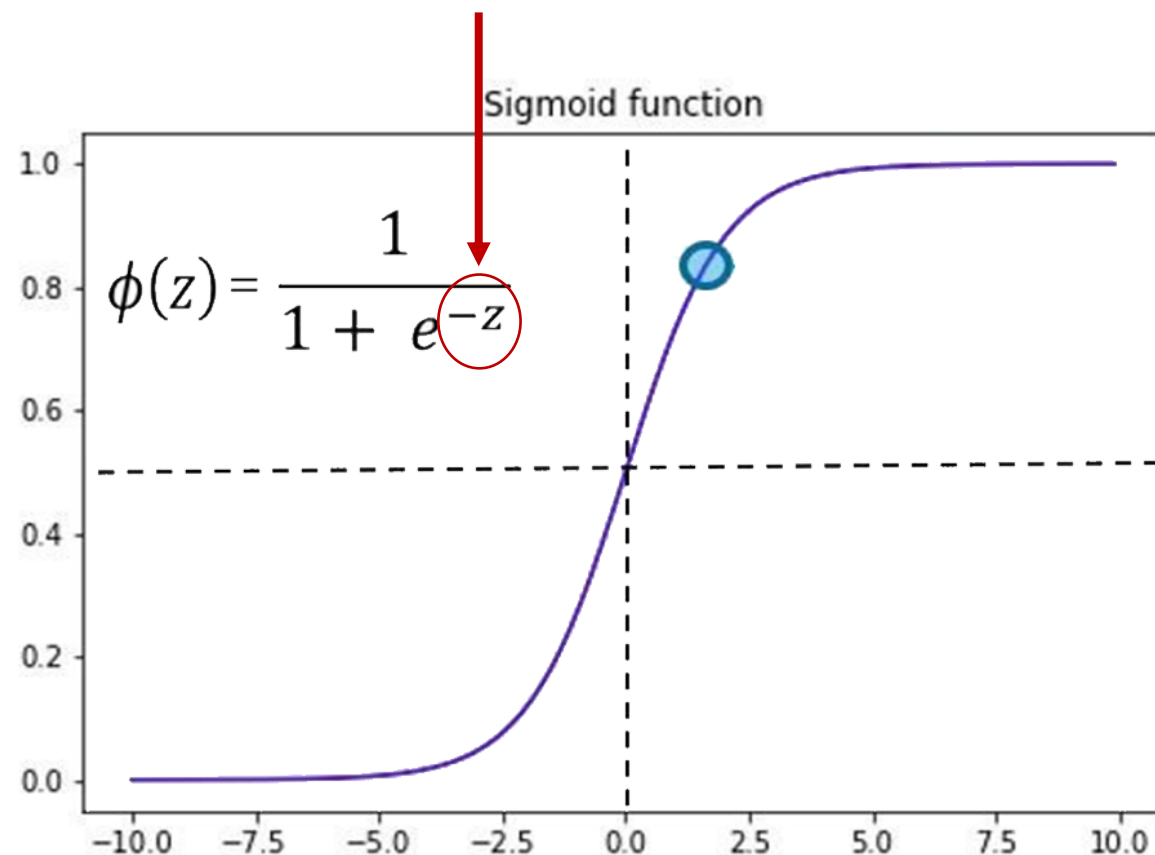
▪ Funcionamiento:

La probabilidad de que **cierta muestra** pertenezca a una clase particular es la inversa de la función *logit*, es decir la función logística o sigmoide, la cual está definida como:

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

donde z es la combinación lineal entre los pesos (w_i) y las características (x^i), es decir, $z = \mathbf{W}^T \mathbf{x}$.

Esta es la salida del modelo de regresión lineal



Regresión Logística

▪ Funcionamiento:

La salida de la función sigmoide puede ser interpretada como la **probabilidad** de que una observación particular pertenezca a la Clase 1, dadas sus características \mathbf{x} , ponderadas por los pesos \mathbf{W} .

$$\phi(z) = P(y = 1|\mathbf{x};\mathbf{W})$$

Con base en esto, a través de la función *logit* obtenemos el modelo lineal:

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m$$

▪ Ejemplo: Diagnóstico de cáncer

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$

Para un tamaño de tumor definido el modelo tiene como salida:


$$\phi(z) = 0,7$$

Esto nos indica que el paciente tiene 70% de probabilidad de que el tumor sea maligno por su tamaño

Regresión Logística

- **Funcionamiento:**

Para estimar los coeficientes \mathbf{W} se debe minimizar una función de costo. En nuestro caso la función de costo, interpretada como la suma de los errores, corresponde al logaritmo de la verosimilitud que es:

$$J(\mathbf{W}) = \sum_{i=1}^n -y_i \log(\phi(z_i)) - (1 - y_i) \log(1 - \phi(z_i))$$


Es la predicción hecha por el modelo

$$P(y = 1|\mathbf{x};\mathbf{W})$$

Este error se puede ver como la diferencia que hay entre la clase estimada y la etiqueta real de cada observación.

Regresión Logística

VENTAJAS:

- Es un modelo de clasificación eficiente y simple.
- No es necesario disponer de grandes recursos computacionales.
- Los resultados son altamente interpretables.



DESVENTAJAS:

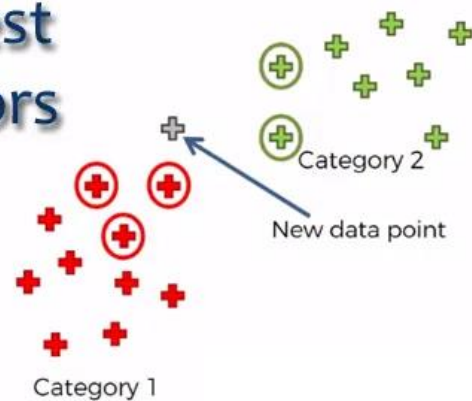
- Imposibilidad de resolver directamente problemas no lineales.
- La variable objetivo esta ha de ser linealmente separable.
- La regresión logística no es uno de los algoritmos más potentes que existen.



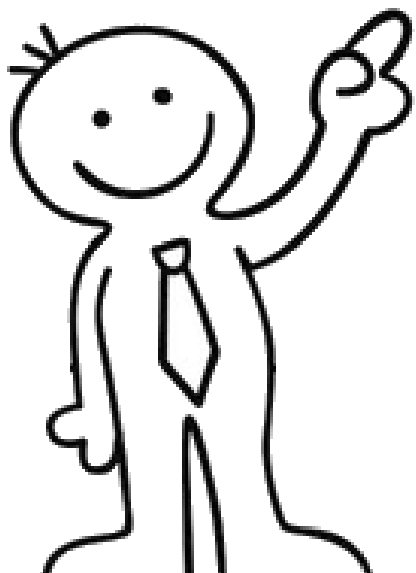


VAMOS A CODIFICAR!

K-Nearest Neighbors






CLASIFICADOR DE LOS K-VECINOS MÁS
CERCANOS



Clasificador kNN

LA BASE DEL CLASIFICADOR

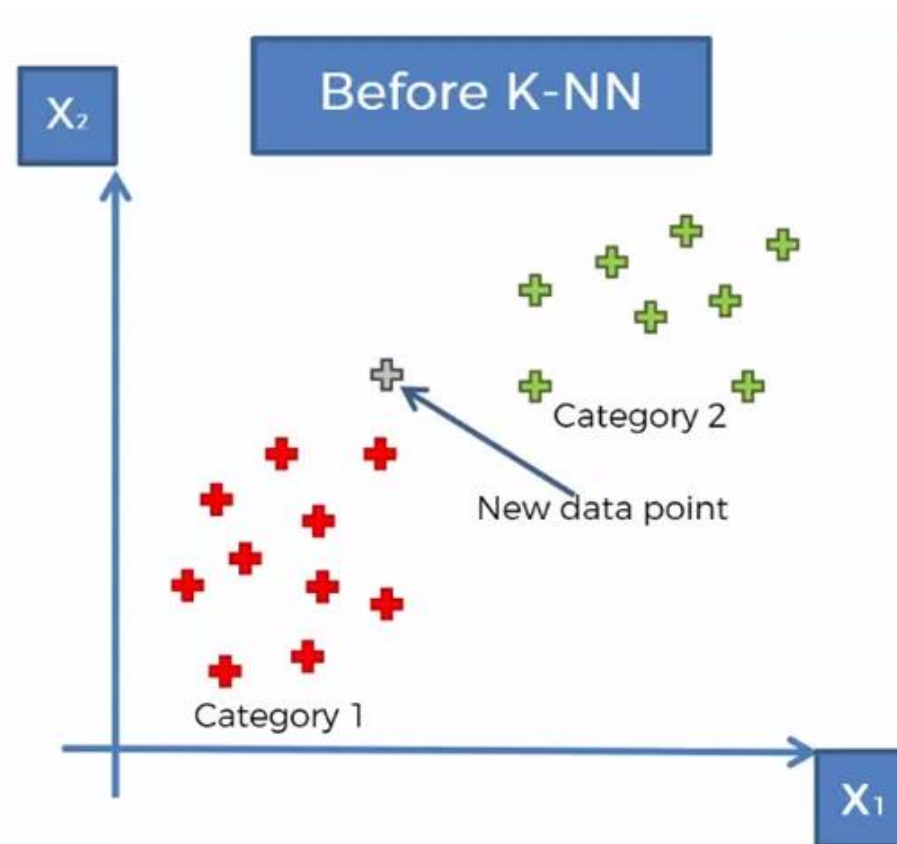
-  Este es un clasificador basado en las instancias en que no se aprende o se crea un modelo generalizado de los datos, sino que se crean modelos locales a demanda:
-  El clasificador simplemente almacena los datos de entrenamiento y el proceso de clasificación se realiza a partir de un voto mayoritario simple entre los k -vecinos del vecindario (V) de un punto P . Es decir, se cuentan el número n de vecinos de P que pertenecen a cada clase y se escoge la clase con más votos.
-  Esta regla basa su operación en el supuesto de que la clase a la que pertenece un punto P es la que tienen la mayor probabilidad de aparición en el vecindario de P .

Clasificador kNN



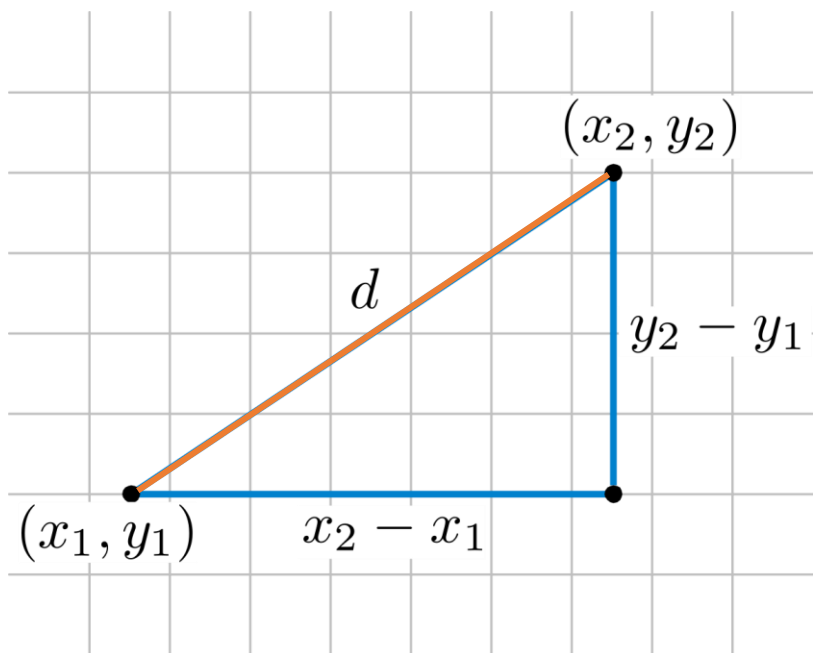
ALGORITMO DE LOS K VECINOS MÁS CERCANOS

1. Se toma el punto P a evaluar (+) y se calcula la distancia de este a cada uno de los puntos en el conjunto de datos de entrenamiento.
2. Se seleccionan los k puntos más cercanos, es decir aquellos con menor distancia (según la función de distancia usada)
3. Se realiza una votación y se asigna a P la etiqueta de clase que predomine entre los k-vecinos



Clasificador kNN

ALGORITMO DE LOS K VECINOS MÁS CERCANOS – MÉTRICAS DE DISTANCIA



DISTANCIA EUCLIDIANA: distancia de la línea recta que conecta a los dos puntos

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

DISTANCIA MANHATTAN: distancia por bloques entre los puntos

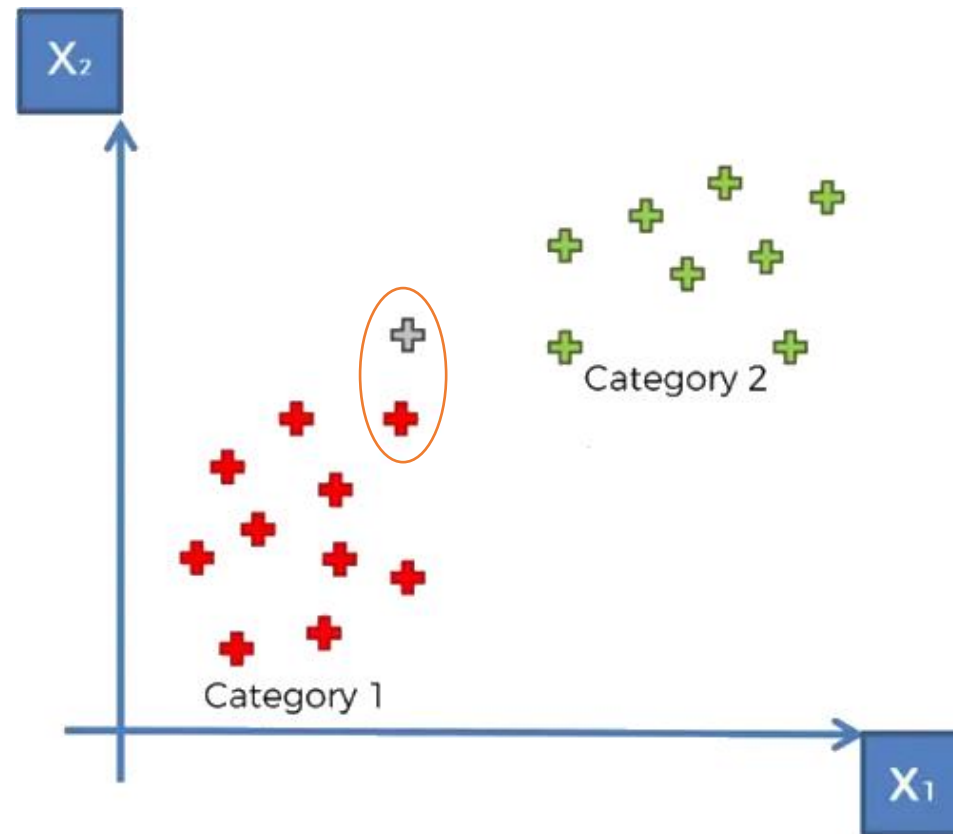
$$d = |x_2 - x_1| + |y_2 - y_1|$$

DISTANCIA MINKOWSKI: Es una generalización de las distancias

$$d = \left(\sum_{i=1}^n |x_2^i - x_1^i|^p \right)^{\frac{1}{p}}$$

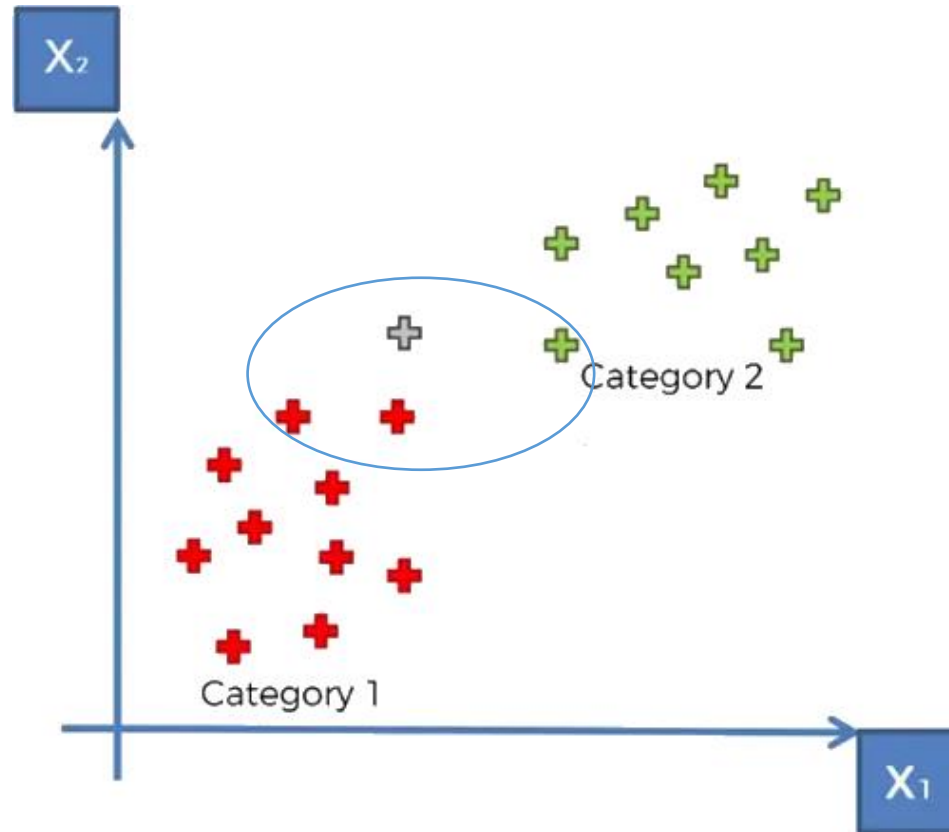
Clasificador kNN

USANDO 1 VECINO



Clasificador kNN

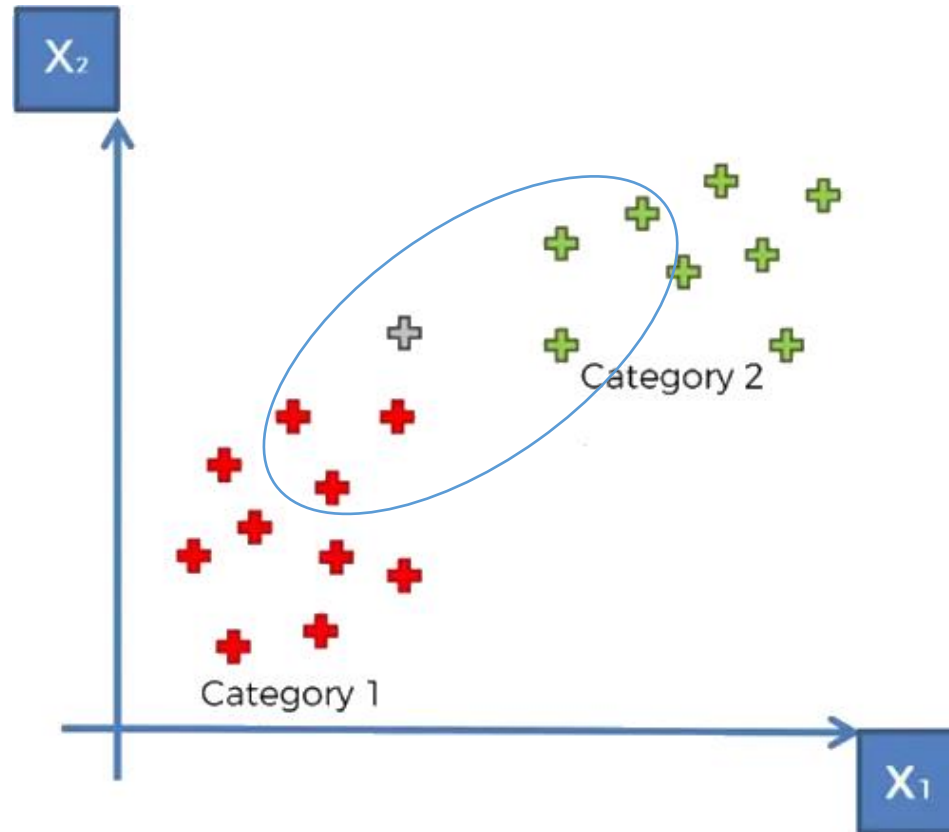
USANDO 3 VECINOS



Clasificador kNN



USANDO 6 VECINOS



Clasificador kNN

VENTAJAS:

- Es clasificador bastante simple
- Puede lidiar con fronteras de decisión complejas y de formas arbitrarias
- Se ha demostrado que la precisión de este clasificador puede ser bastante fuerte y en muchos casos tan precisa como otros métodos elaborados
- En un clasificador **no paramétrico** por lo que no hace suposiciones explícitas sobre la distribución subyacente de los datos
- Suele ser insensible a los valores atípicos



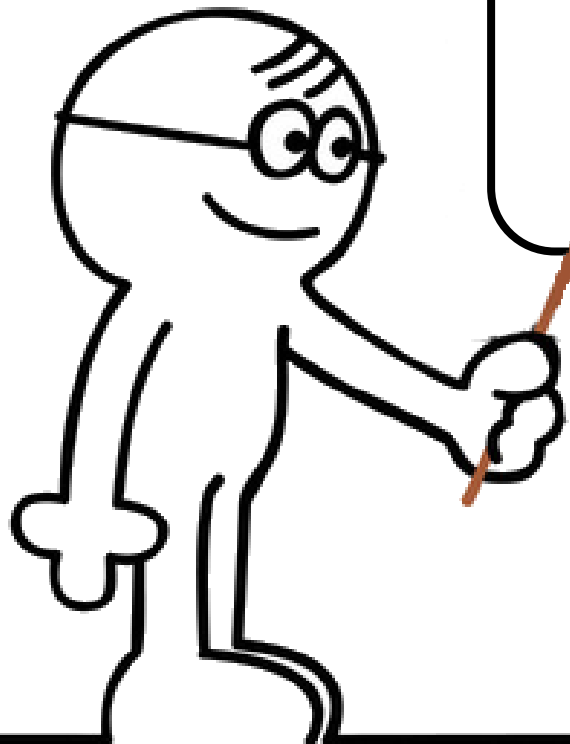
DESVENTAJAS:

- Tiene requisito de memoria alta porque almacena todos los datos de entrenamiento
- Es computacionalmente costoso (lento) porque se deben calcular las distancias a todos los puntos del conjunto de entrenamiento con cada uso
- El “modelo” que genera el clasificador no es interpretable





VAMOS A CODIFICAR!



¿EN QUÉ CONSISTE UN CLASIFICADOR
INGENUO DE BAYES?

Teorema de Bayes

🏆 LA REGLA DE BAYES

El **Teorema de Bayes** expresa la probabilidad *a posteriori* de un evento aleatorio A (que es una clase c_i) dado un evento B (que es el vector de características, x) en términos de la distribución de probabilidad condicional y la probabilidad marginal.

GAUSSIAN
NAIVE BAYES
CLASSIFIER

"Gaussian" because this is a normal distribution →

This is our prior belief →

$$P(\text{class} | \text{data}) = \frac{P(\text{data} | \text{class}) \times P(\text{class})}{P(\text{data})}$$

We don't calculate this in naive bayes classifiers →

ChrisAlbon

Teorema de Bayes

LA REGLA DE BAYES

El **Teorema de Bayes** expresa la probabilidad *a posteriori* de un evento aleatorio A (que es una clase c_i) dado un evento B (que es el vector de características, x) en términos de la distribución de probabilidad condicional y la probabilidad marginal.

$$P(c_i|x) = \frac{p(x|c_i)P(c_i)}{p(x)}$$

$P(c_i)$ = Es la probabilidad de que en la población haya un objeto de clase c_i .

$p(x|c_i)$ = Es la probabilidad de que en la clase c_i haya un objeto con un vector de características x .

$P(c_i|x)$ = Es la probabilidad de que el objeto con un vector de características x pertenezca a la clase c_i .

$p(x)$ = Es la probabilidad de que haya un objeto con el vector de características x , independiente de la clase a la que pertenezca.

Teorema de Bayes

$$P(c_i|x) = \frac{p(x|c_i)P(c_i)}{p(x)}$$

🏆 LA REGLA DE BAYES

🎯 Para un problema de 2 clases (A y B):

$$\begin{aligned} p(A|x) &> p(B|x) && \rightarrow A \text{ else } B \\ \frac{p(x|A) p(A)}{p(x)} &> \frac{p(x|B) p(B)}{p(x)} && \rightarrow A \text{ else } B \\ p(x|A) p(A) &> p(x|B) p(B) && \rightarrow A \text{ else } B \end{aligned}$$

🎯 Así un clasificador g se define como: $g(x) = \begin{cases} 1 & \text{si } P(c_1|x) > P(c_2|x) \\ 2 & \text{en otro caso} \end{cases}$

🎯 Para múltiples clases el clasificador se define como $g(x) = \arg \max_{c_i} (P(x|c_i)P(c_i))$

Naïve Bayes

■ Características:

- Es un clasificador lineal, simple y eficiente.
- Su modelo probabilístico se basa en el Teorema de Bayes
- El adjetivo de “ingenuo” viene de la suposición de que las características son mutuamente independientes (i.i.d.).
- En la práctica, la suposición de independencia se viola frecuentemente, pero este clasificador tiene un buen desempeño aún bajo esta suposición, especialmente para tamaños pequeños de muestra.

Por ejemplo: una fruta puede ser considerada como una manzana si es roja, redonda y de alrededor de 7 cm de diámetro.

Este clasificador considera que cada característica contribuye de manera independiente a la probabilidad de que esta fruta sea una manzana, independientemente de la presencia o ausencia de las otras características.

Naïve Bayes

■ Definiciones:

- Para el clasificador de Bayes ingenuo, la independencia significa que la probabilidad de una observación no afecta la probabilidad de otra observación. Esto lleva a la probabilidad de clase condicional puede calcularse como:

$$P(\mathbf{x}|\mathbf{c}_j) = \prod_{k=1}^d P(x_k|\mathbf{c}_j)$$

Donde $P(\mathbf{x}|\mathbf{c}_j)$ significa ¿Qué tan probable es observar este patrón particular \mathbf{x} dado que pertenece a la clase \mathbf{c}_j ?

■ Definiciones:

- Las verosimilitudes individuales para cada característica pueden estimarse vía estimación de máxima verosimilitud, lo que es una simple frecuencia en el caso de datos categóricos:

$$\hat{P}(x_i|\mathbf{c}_j) = \frac{N_{x_i,\mathbf{c}_j}}{N_{\mathbf{c}_j}}$$

donde N_{x_i,\mathbf{c}_j} es el número de veces que la característica x_i aparece en las observaciones de clase \mathbf{c}_j , y $N_{\mathbf{c}_j}$ es el conteo total de todas las características en la clase \mathbf{c}_j

Naïve Bayes

■ Definiciones:

- La probabilidad *a priori* también es llamada *prior* de clase y describe la probabilidad general de encontrar una clase particular. Si los *prior*es siguen una distribución uniforme, las probabilidades posteriores estarán determinadas por completo por la probabilidad de clase condicional y por la evidencia.

■ Definiciones:

- Eventualmente, el conocimiento a priori puede ser determinado a través del conjunto de entrenamiento, si se asume que los datos de entrenamiento son i.i.d y que son una muestra representativa de toda la población.

$$\hat{P}(c_j) = \frac{N_{c_j}}{N}$$

donde N_{c_j} es el número de muestras de la clase c_j y N es el número total de muestras.

Naïve Bayes

■ Definiciones:

- La evidencia $P(x)$ puede entenderse como la probabilidad de encontrar un patrón particular x independientemente de la etiqueta de clase. Usualmente se puede eliminar de la regla de decisión, debido a que suele ser común a todos los términos.

■ Funcionamiento del Clasificador:

- **Paso 1:** Calcule la probabilidad a priori para cada una de las clases
- **Paso 2:** Estime la probabilidad $\hat{P}(x_i|c_j)$ para cada característica dada cada clase usando un estimador de máxima verosimilitud
- **Paso 3:** Use la fórmula de Bayes y calcule la probabilidad posterior
- **Paso 4:** Determine cuál de las clases tiene la probabilidad más alta, y asigne esa clase a un dato particular x

Naïve Bayes

▪ Ejemplo:

- Necesitamos determinar la probabilidad si un jugador jugará un partido con base en las condiciones climáticas. Para ello tenemos los datos en la tabla.
- Lo primero que debemos hacer es crear tres tablas: la tabla de frecuencias por condición climática y las tablas de verosimilitud.

Wheter	Play
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Rainy	Yes
Overcast	Yes
Sunny	Yes
Sunny	No
Rainy	No
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	Yes

Naïve Bayes

Whether	Play
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Rainy	Yes
Overcast	Yes
Sunny	Yes
Sunny	No
Rainy	No
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	Yes



Frequency Table

Whether	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9



Likelihood Table 1

Whether	No	Yes		
Overcast		4	=4/14	0.29
Sunny	2	3	=5/14	0.36
Rainy	3	2	=5/14	0.36
Total	5	9		
	=5/14	=9/14		
	0.36	0.64		

$P(x)$

$\hat{P}(c_j)$

Likelihood Table 2

Whether	No	Yes	Posterior Probability for No	Posterior Probability for Yes
Overcast		4	0/5=0	4/9=0.44
Sunny	2	3	2/5=0.4	3/9=0.33
Rainy	3	2	3/5=0.6	2/9=0.22
Total	5	9		

$\hat{P}(x_i|c_j)$

Naïve Bayes

- Ejemplo:

- Ahora suponga que desea calcular la probabilidad de jugar cuando el clima está nublado, entonces se utiliza el teorema de Bayes con base en las probabilidades calculadas.

$$P(Yes|Overcast) = \frac{P(Overcast|Yes)P(Yes)}{P(Overcast)}$$

De las tablas tenemos:

$$P(Overcast) = 4/14 = 0.29$$

$$P(Yes) = 9/14 = 0.64$$

$$P(Overcast|Yes) = 4/9 = 0.44$$

$$P(Yes | Overcast) = 0.44 * 0.64 / 0.29 = 0.98$$

Naïve Bayes

VENTAJAS:

- Es fácil y rápido hacer una predicción.
- Cuando se mantiene la suposición de independencia, el clasificador de Bayes funciona mejor en comparación con otros modelos como la Regresión Logística.
- Funciona bien en el caso de variables de entrada categóricas, comparado con variables numéricas.



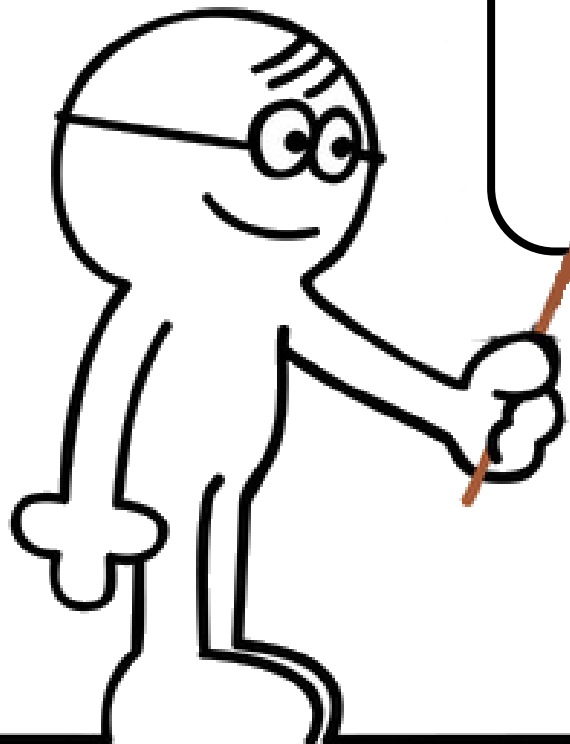
DESVENTAJAS:

- Si la variable categórica tiene una categoría en el conjunto de datos de prueba, que no se observó en el conjunto de datos de entrenamiento, el modelo asignará una probabilidad de 0 y no podrá hacer una predicción.
- En la vida real, es casi imposible que obtengamos un conjunto de predictores (o características) que sean completamente independientes, por lo que este clasificador es muy limitado en ese sentido.





VAMOS A CODIFICAR!



¿EN QUÉ CONSISTE LDA Y QDA?

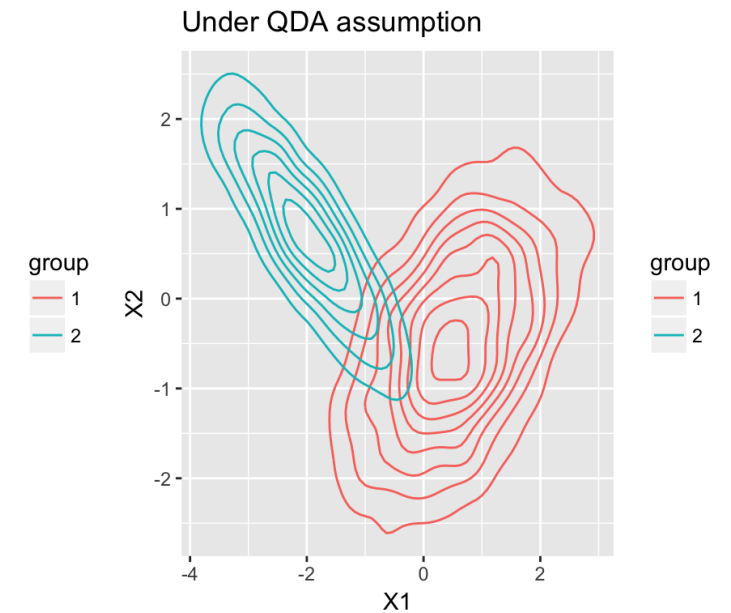
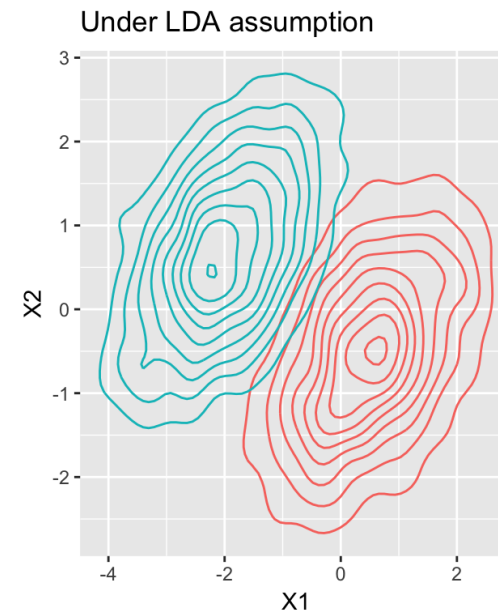
LDA & QDA

DESCRIPCIÓN:

El **Análisis Discriminante Lineal** o *Linear Discriminant Analysis (LDA)*, al igual que el **Análisis Discriminante Cuadrático** o *Quadratic Discriminant Analysis (QDA)* son métodos de clasificación que asumen distribuciones gaussianas sobre cada una las características.

La base de ambos clasificadores es el Teorema de Bayes

$$P(\text{class} | \text{data}) = \frac{P(\text{data} | \text{class}) \times P(\text{class})}{P(\text{data})}$$



LDA & QDA

🌀 **LA BASE:** la función discriminante general toma la forma:

$$g(\mathbf{x}) = \arg \max_{c_i} (P(\mathbf{x}|c_i)P(c_i))$$

Es decir que para cada clase su función discriminante es:

$$g_i(\mathbf{x}) = P(\mathbf{x}|c_i)P(c_i)$$

Por simplicidad, se puede usar una función **monótona creciente** para transformar las salidas manteniendo el orden de la comparación entre las funciones discriminantes, por tanto,

$$\begin{aligned} h_i(\mathbf{x}) &= \ln(g_i(\mathbf{x})) \\ &= \ln(P(\mathbf{x}|c_i)) + \ln(P(c_i)) \end{aligned}$$

Se asume que la probabilidad $P(\mathbf{x}|c_i)$ se comporta como una distribución normal multivariada, para cada clase:

$$P(\mathbf{x}|c_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

Así,

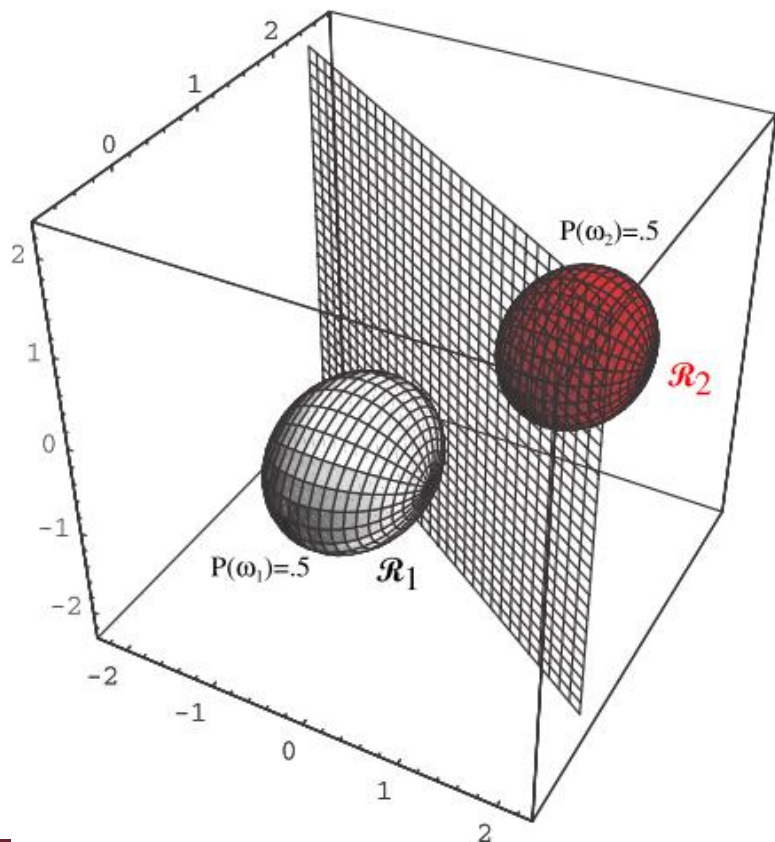
$$P(\mathbf{x}|c_i) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\right)$$

Ahora, si reemplazamos $p(\mathbf{x}|c_i)$ en la función discriminante $h_i(\mathbf{x})$ tenemos que:

$$h_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) - \frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln(|\boldsymbol{\Sigma}_i|) + \ln(P(c_i))$$

LDA & QDA

- 🌀 **CASO 1:** $\Sigma_i = \sigma^2 I$: Se asume que las características son estadísticamente independientes y todas tienen la misma varianza



Partiendo de la ecuación:

$$h_i(x) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln(|\boldsymbol{\Sigma}_i|) + \ln(P(c_i))$$

Con el supuesto anterior tenemos que:

$$|\boldsymbol{\Sigma}_i| = \sigma^{2d}$$

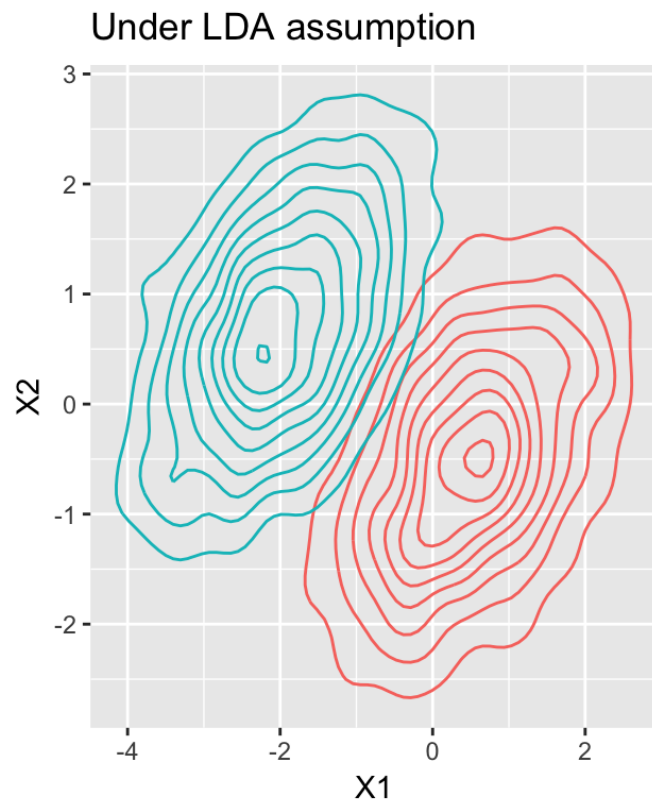
$$\boldsymbol{\Sigma}_i^{-1} = \left(\frac{1}{\sigma^2}\right) \mathbf{I}$$

Como $|\boldsymbol{\Sigma}_i|$ no depende de i (la clase), al igual que el término $\frac{d}{2} \ln(2\pi)$, entonces se toman como constantes aditivas que se pueden ignorar. Así la función discriminante de cada clase es:

$$h_i(x) = \frac{-\|\mathbf{x} - \boldsymbol{\mu}_i\|^2}{2\sigma^2} + \ln(P(c_i))$$

LDA & QDA

CASO 2: $\Sigma_i = \Sigma$: Se asume que las matrices de covarianzas son iguales para todas las clases, pero es arbitraria



Partiendo de la ecuación:


$$h_i(x) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln(|\boldsymbol{\Sigma}_i|) + \ln(P(c_i))$$

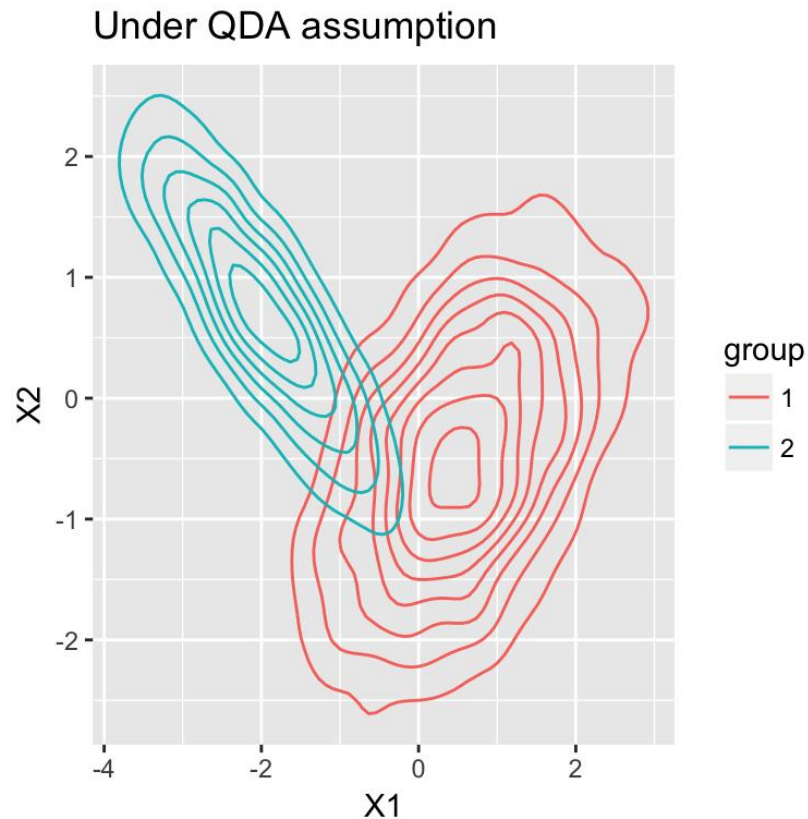
Con el supuesto anterior tenemos que $|\boldsymbol{\Sigma}_i|$ no depende de i (la clase), al igual que el término $\frac{d}{2} \ln(2\pi)$, entonces se toman como constantes aditivas que se pueden ignorar. Así la función discriminante de cada clase es:

$$h_i(x) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) + \ln(P(c_i))$$

Esta función aún sigue siendo lineal y está basada en el cuadrado de la distancia Mahalanobis.

LDA & QDA

 **CASO 3: $\Sigma_i = \text{arbitraria}$:** Se asume que las matrices de covarianzas son arbitrarias para todas las clases



Partiendo de la ecuación:

$$h_i(x) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln(|\boldsymbol{\Sigma}_i|) + \ln(P(c_i))$$

En este caso el único término que puede ser ignorado es $\frac{d}{2} \ln(2\pi)$.
Así la función discriminante de cada clase es:

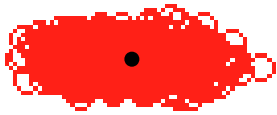
$$h_i(x) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{1}{2} \ln(|\boldsymbol{\Sigma}_i|) + \ln(P(c_i))$$

Esta función ya es una función cuadrática y representa al clasificador QDA.

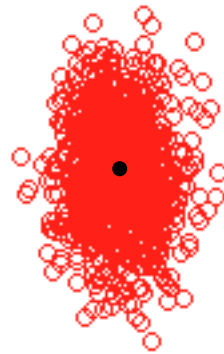
LDA & QDA

🌀 **CASO 3: $\Sigma_i = \textit{arbitraria}$:** Se asume que las matrices de covarianzas son arbitrarias para todas las clases.

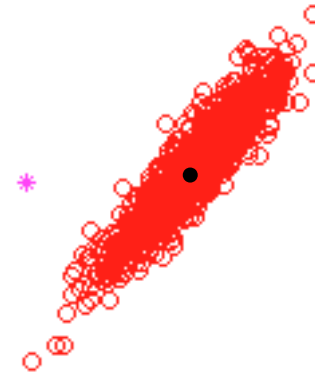
Ejemplos de Σ y μ en 2D:



6.0057	-0.1020
-0.1020	1.0632



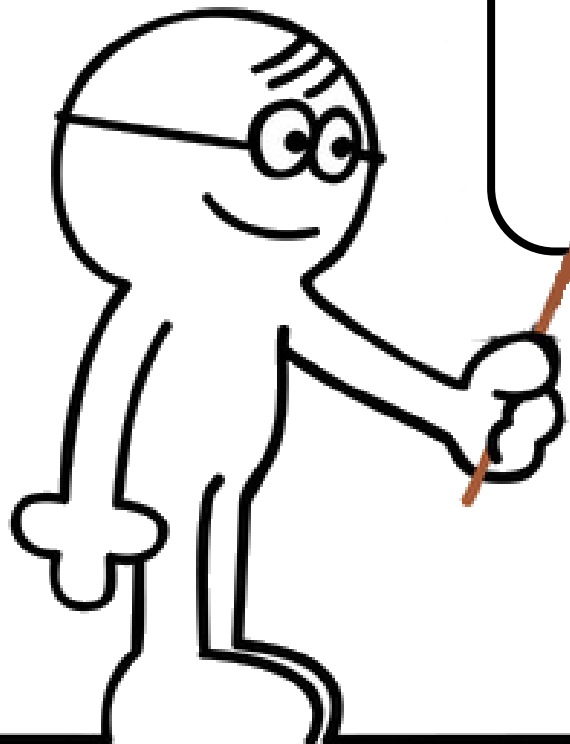
7.7947	-0.7772
-0.7772	43.1459



15.1951	21.8267
21.8267	37.6734



VAMOS A CODIFICAR!



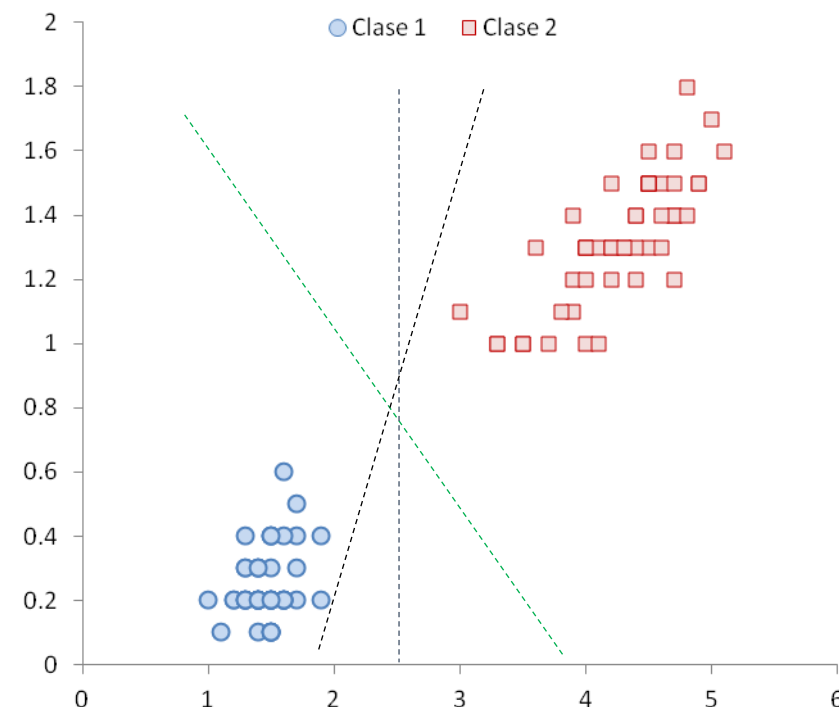
¿EN QUÉ CONSISTE LAS SVM?

SVM

INTRODUCCIÓN:

- Las **Máquinas de Vectores de Soporte (SVM)** buscan el hiperplano que separe de forma **óptima** los puntos de una clase, respecto de la de otra.
- En un problema de dos clases, linealmente separables, pueden existir muchas fronteras de decisión (o hiperplanos) que pueden separar las clases.
- Sin embargo, ¿Son todas esas fronteras igual de buenas?

Lea mas en: <http://svms.org/tutorials/>



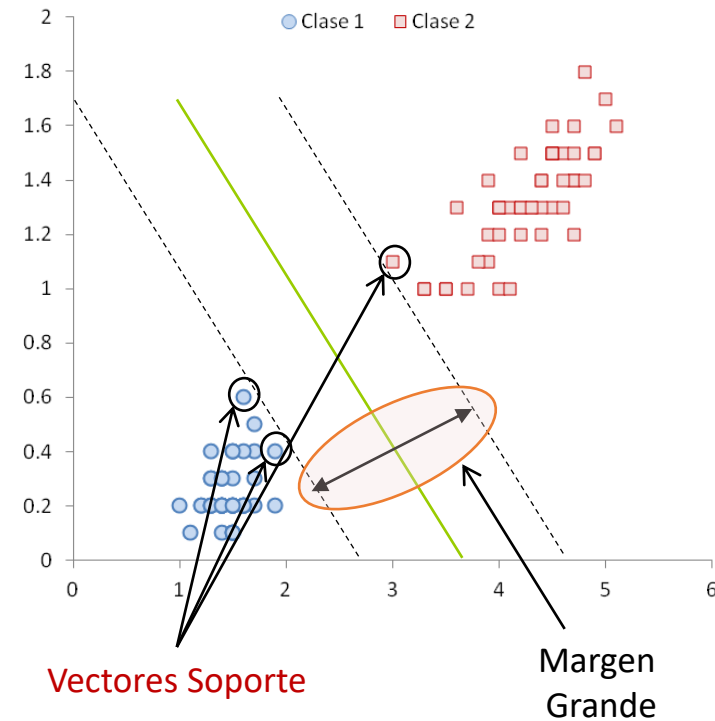
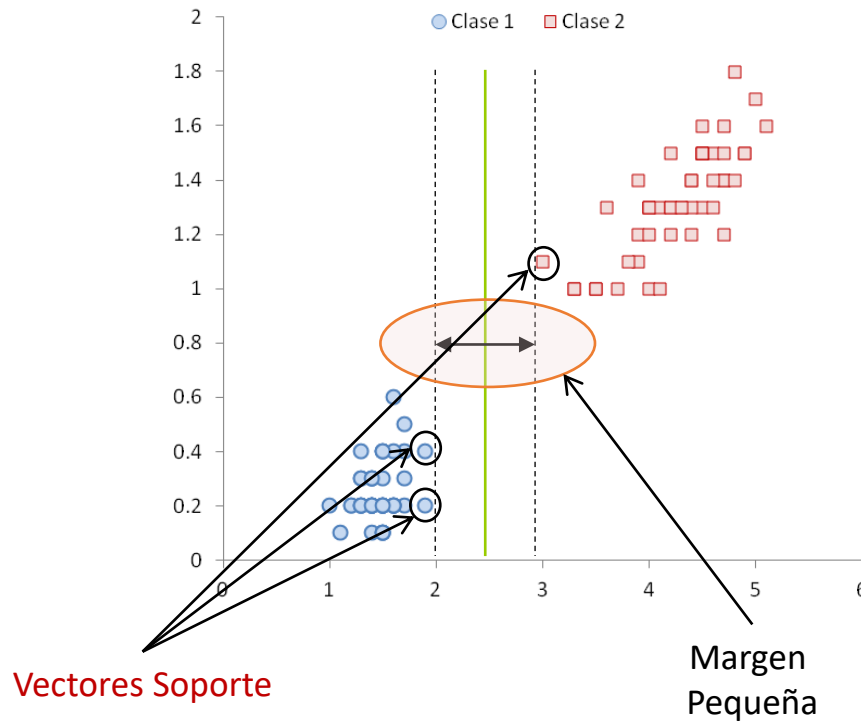
¿CÓMO SELECCIONAR EL MEJOR
HIPERPLANO DE SEPARACIÓN?



SVM

FUNCIONAMIENTO:

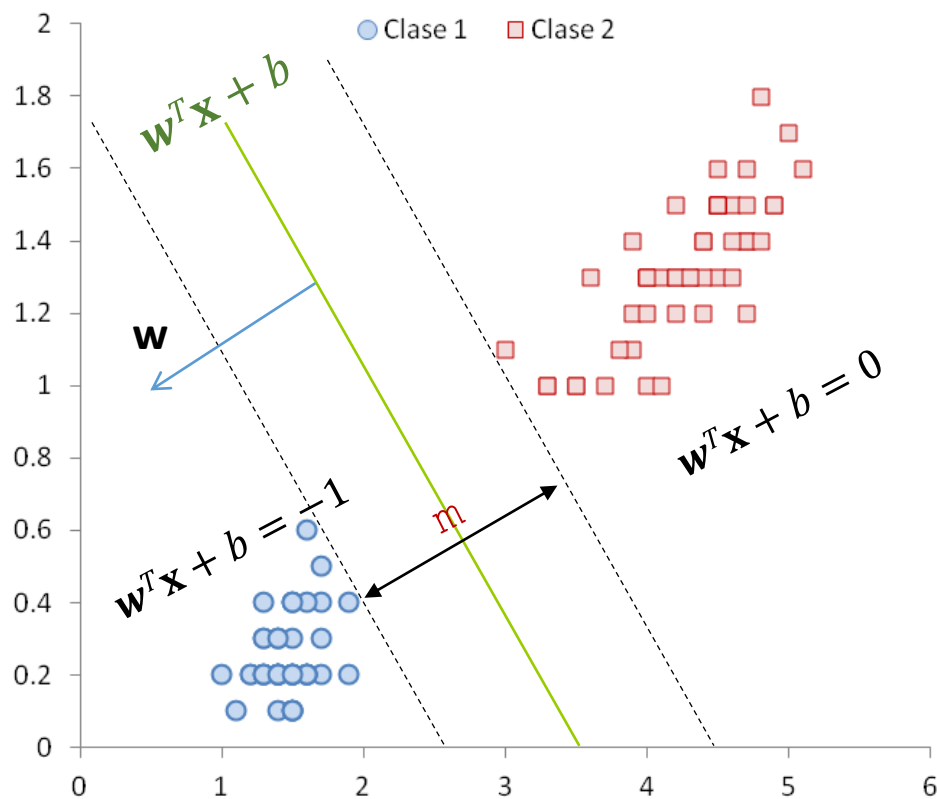
La SVM busca el hiperplano que **maximiza** la distancia (o margen) con los puntos que estén más cerca de él, razón por la cual también se les conoce a las SVM como clasificadores de margen máximo.



SVM

EN NOTACIÓN FORMAL:

Como el hiperplano separa las muestras positivas (+1) de las negativas (-1), los puntos que están en el hiperplano deben satisfacer la ecuación: $\mathbf{w}^T \mathbf{x} + b = 0$.



TAL QUE ...

El vector \mathbf{W} es la normal al hiperplano

$\|\mathbf{W}\|$ es la norma del vector

$\frac{b}{\|\mathbf{W}\|}$ es la distancia perpendicular del hiperplano al origen.

$m = \frac{2}{\|\mathbf{W}\|}$ es la margen o distancia entre los hiperplanos positivo y negativo.

SVM

EN NOTACIÓN FORMAL:

- El problema en las SVM es entonces maximizar $\frac{2}{\|\mathbf{W}\|}$
- O lo que es lo mismo, minimizar $\left(\frac{1}{2} \|\mathbf{W}\|^2\right)$ sujeto a que $y_i(\mathbf{W}^T \mathbf{x}_i - b) \geq 1 \quad \forall i = 1 \dots n$
- Para resolver este problema se usan Multiplicadores de Lagrange, de tal forma que se debe construir una función Lagrangiana tal que:

$$\tilde{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

Sujeta a que: $\alpha_i \geq 0$ y $\sum_{i=1}^n \alpha_i y_i = 0$

- Finalmente, el vector \mathbf{W} se puede calcular gracias a los términos α como $\mathbf{W} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$

Con $b = \frac{1}{n} \sum_{i=1}^n (\mathbf{W}^T \mathbf{x}_i - y_i)$

Nota: Las bibliotecas de funciones hacen todo esto por nosotros :)

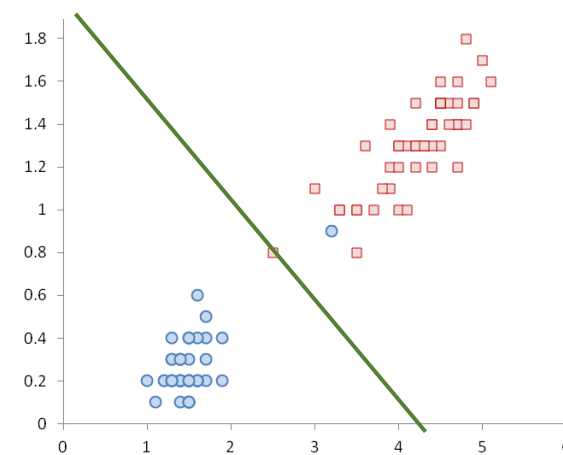
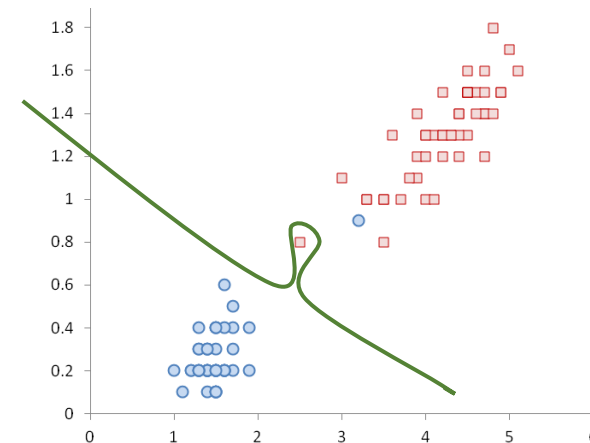
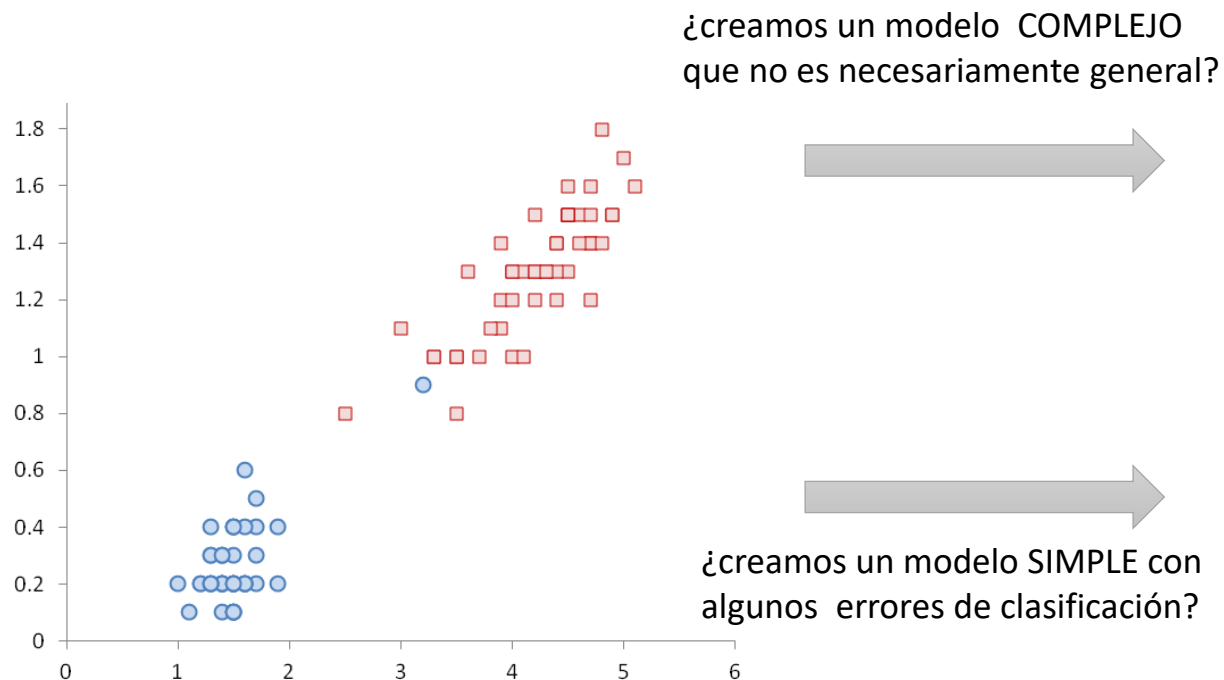


PERO, ¿QUÉ PASA CON
UN CASO COMO EL QUE SIGUE?

SVM

SOFT-MARGIN:

¿Qué hacer cuando sólo hay unos pocos datos que no permiten separar linealmente el conjunto de datos?



SVM

SOFT-MARGIN:

- Con el fin de permitir cierta flexibilidad, las SVM manejan un parámetro **C** que controla la compensación entre errores de entrenamiento y los márgenes rígidos:

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

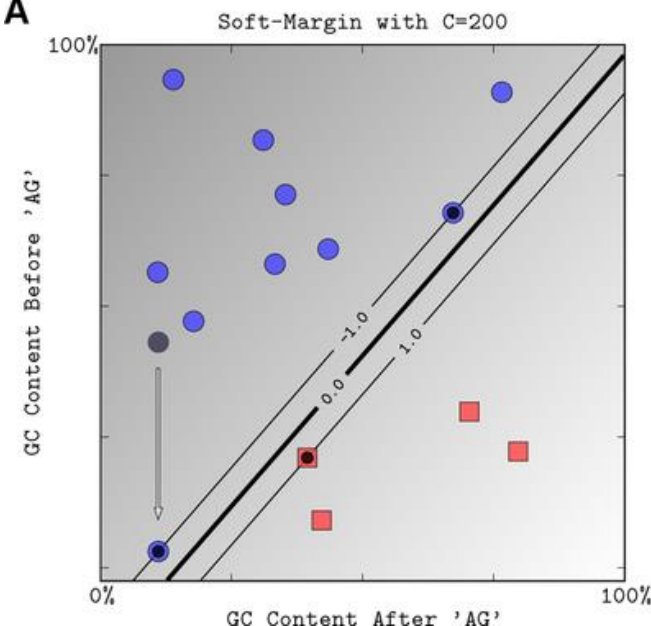
- Nótese que si $\xi_i = 0$, no hay error, así se busca minimizar:

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

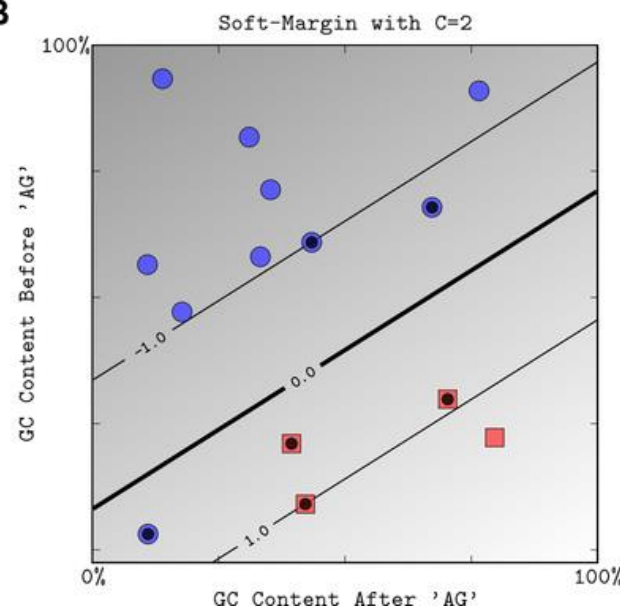
- La constante **C** determina la holgura del margen blando. La elección de este valor y del tipo de **función kernel** influyen en el desempeño de las SVM.

Con C grande

A



B



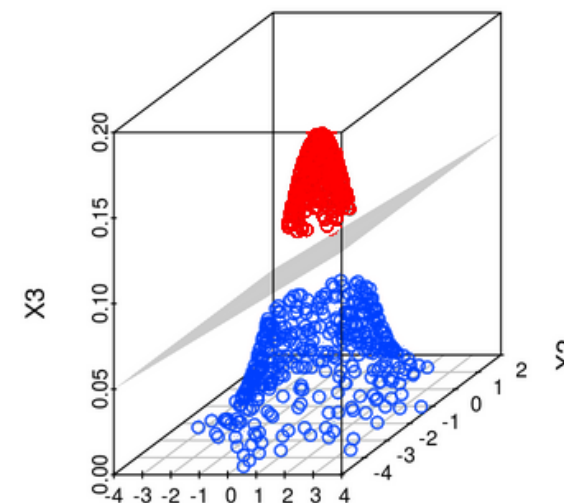
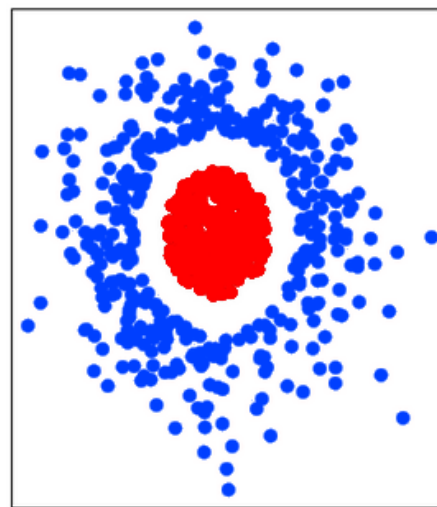
Con C apropiado

¿Y SI LOS DATOS NO SE PUEDEN SEPARAR
LINEALMENTE?



MODELOS NO LINEALES:

Cuando los datos no se pueden separar linealmente se hace un cambio de espacio mediante una función de transformación que aumenta la dimensionalidad de los vectores de entrada a un espacio al que se puedan separar linealmente por un hiperplano. A tal función se llama **Kernel**.



Al introducir un kernel, los parámetros α del vector \mathbf{W} se hayan así:

$$\tilde{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

MODELOS NO LIENALES:

Existen diferentes tipo de funciones *Kernel*:

-  **Kernel Polinomial** de grado d (muy usado en reconocimiento de imágenes)

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^d$$

-  **Kernel de Base Radial** de ancho s

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / (2\sigma^2))$$

Su espacio de llegada es de dimensión infinita.

-  **Kernel Sinusoidal** con parámetros k y q

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x}^T \mathbf{y} + \theta)$$

VENTAJAS:

- El modelo final puede ser escrito como una combinación de un número muy pequeño de vectores de entrada, llamados vectores de soporte, lo cual se presenta ya que el Lagrangiano de la mayoría de los vectores es igual a cero.
- Tiene una gran capacidad de generalización, incluso cuando el conjunto de entrenamiento es pequeño, esto gracias a la función kernel.
- Existen pocos parámetros a ajustar; el modelo solo depende de los datos con mayor información.
- La estimación de los parámetros se realiza a través de la optimización de una función de costo convexa, lo cual evita la existencia de un mínimo local



DESVENTAJAS:

- Cuando los parámetros son mal seleccionados se puede presentar un problema de sobreentrenamiento, el cual ocurre cuando se han aprendido muy bien los datos de entrenamiento pero no se pueden clasificar bien ejemplos nunca antes vistos.
- En gran medida, la solución al problema, así como su generalización depende del kernel que se use y de los parámetros del mismo.





VAMOS A CODIFICAR!