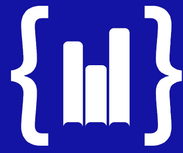
Programa Técnico Intensivo en Data Science

Support Vector Machines

Alberto Oteo García
Merck Kgaa
aog1395@gmail.com

IDAL

Intelligent
Data
Analysis
Laboratory



Datamecum

¿Qué es una SVM?

- El método de clasificación-regresión Máquinas de Vector Soporte (*Vector Support Machines, SVMs*) fue desarrollado en la década de los 90, dentro de campo de la ciencia computacional. Si bien originariamente se desarrolló como un método de clasificación binaria, su aplicación se ha extendido a problemas de clasificación múltiple y regresión. *SVMs* ha resultado ser uno de los mejores clasificadores para un amplio abanico de situaciones, por lo que se considera uno de los referentes dentro del ámbito de aprendizaje estadístico y *machine learning*.

- Las Máquinas de Vector Soporte se fundamentan en el *Maximal Margin Classifier*, que a su vez, se basa en el concepto de hiperplano. A lo largo de esta clase se introducen por orden cada uno de estos conceptos. Comprender los fundamentos de las *SVMs* requiere de conocimientos sólidos en álgebra lineal. No se profundizarán en todos los aspectos matemáticos, pero puede encontrarse una descripción detallada en el libro *Support Vector Machines Succinctly by Alexandre Kowalczyk.*

Programa Técnico Intensivo en Data Science

Conceptos básicos

Datamecum

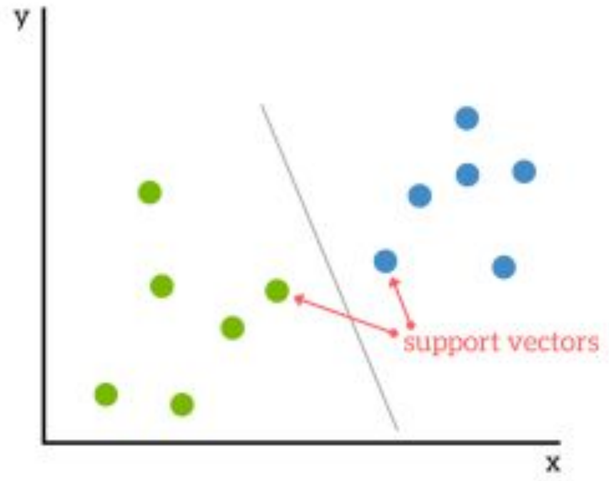
Hiperplano

Como ejemplo sencillo, para una tarea de clasificación con sólo dos características (como la imagen), se puede pensar en un hiperplano como una línea que separa y clasifica linealmente un conjunto de datos.

Intuitivamente, cuanto más lejos del hiperplano se encuentren nuestros puntos de datos, más seguros estaremos de que se han clasificado correctamente. Por lo tanto, queremos que nuestros puntos de datos estén lo más lejos posible del hiperplano, sin dejar de estar en el lado correcto del mismo.

Así, cuando se añaden nuevos datos de prueba, el lado del hiperplano en el que se sitúen decidirá la clase que les asignamos.

$$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b$$



Datamecum

Hiperplano

En un espacio *p*-dimensional, un hiperplano se define como un subespacio plano y afín de dimensiones $p-1$. El término afín significa que el subespacio no tiene por qué pasar por el origen. En un espacio de dos dimensiones, el hiperplano es un subespacio de 1 dimensión, es decir, una recta. En un espacio tridimensional, un hiperplano es un subespacio de dos dimensiones, un plano convencional. Para dimensiones $p>3$ no es intuitivo visualizar un hiperplano, pero el concepto de subespacio con $p-1$ dimensiones se mantiene. La definición matemática de un hiperplano es bastante simple. En el caso de dos dimensiones, el hiperplano se describe acorde a la ecuación de una recta:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$

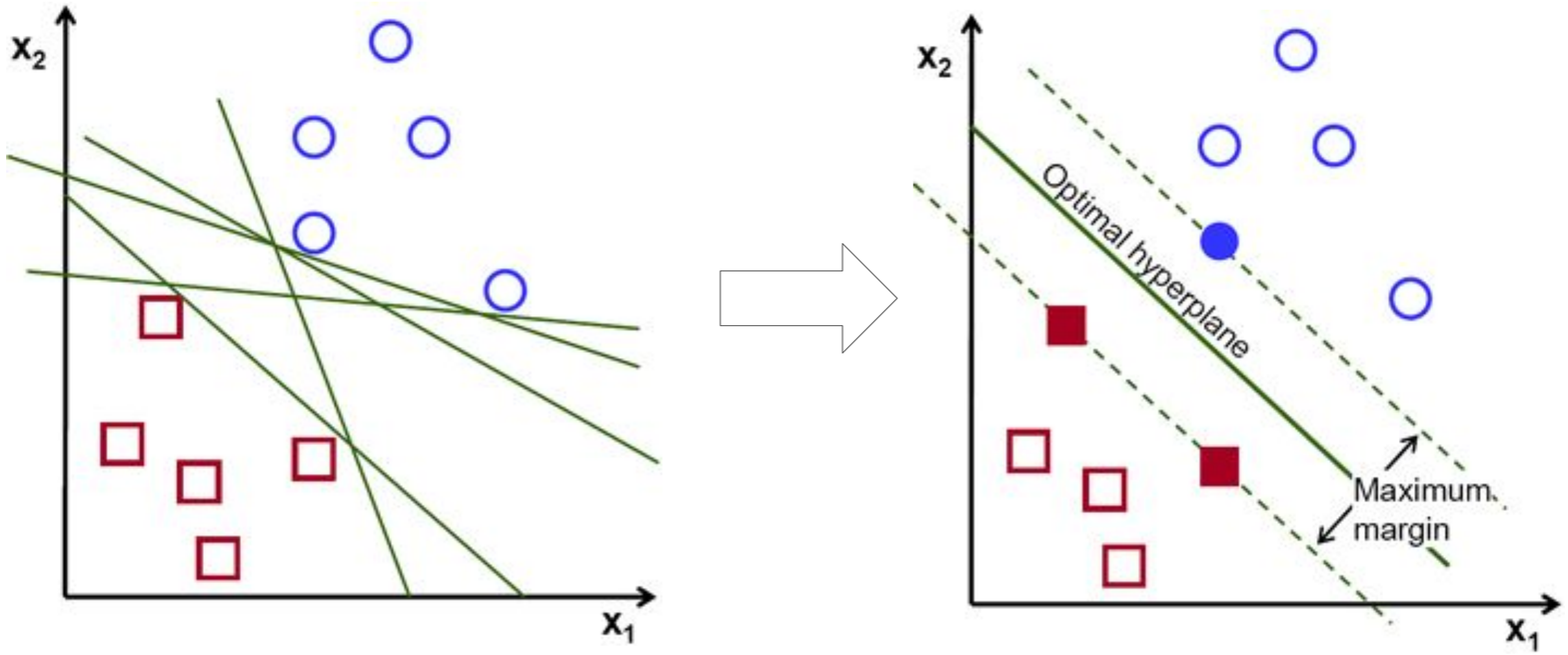
Dados los parámetros β_0, β_1 y β_2 todos los pares de valores $x=(x_1,x_2)$ para los que se cumple la igualdad son puntos del hiperplano.
Esta ecuación puede generalizarse para *p*-dimensiones:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p = 0$$

y de igual manera, todos los puntos definidos por el vector $(x=x_1,x_2,\ldots,x_p)$ que cumplen la ecuación pertenecen al hiperplano.

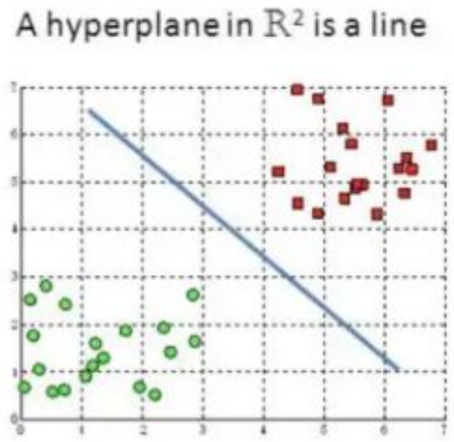
Vectores soporte

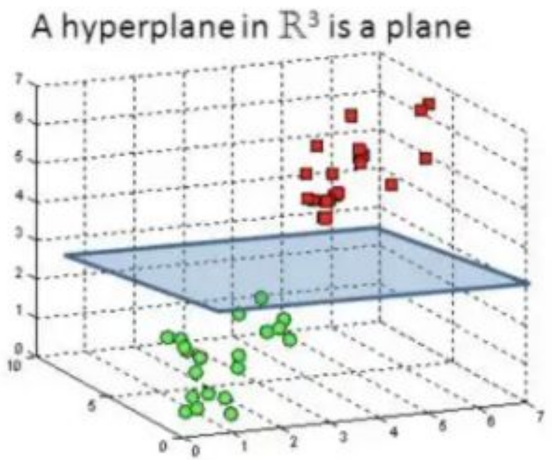
- Los vectores de apoyo son los puntos de datos que se encuentran más cerca de la superficie de decisión (o hiperplano) a la superficie de decisión (o hiperplano)
  - Son los puntos de datos más difíciles de clasificar.
  - Tienen relación directa con la ubicación óptima de la superficie de decisión



Vectores soporte

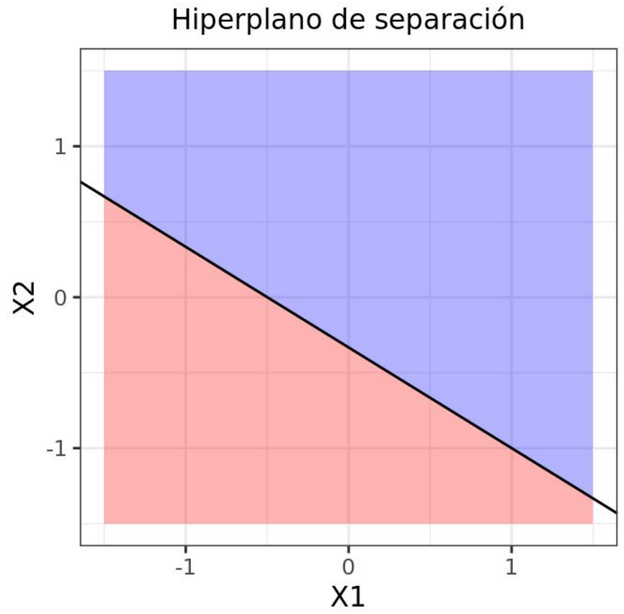
- Los vectores de apoyo son los puntos de datos que se encuentran más cerca de la superficie de decisión (o hiperplano) a la superficie de decisión (o hiperplano)
  - Son los puntos de datos más difíciles de clasificar.
  - Tienen relación directa con la ubicación óptima de la superficie de decisión





Clasificación utilizando un hiperplano

Cuando se dispone de n observaciones, cada una con p predictores y cuya variable respuesta tiene dos valores (identificados como +1 y −1), se pueden emplear hiperplanos para construir un clasificador que permita predecir a que grupo pertenece una observación en función de sus predictores. Este mismo problema puede abordarse también con otros métodos (regresión logística, LDA, árboles de clasificación…) cada uno con ventajas y desventajas.



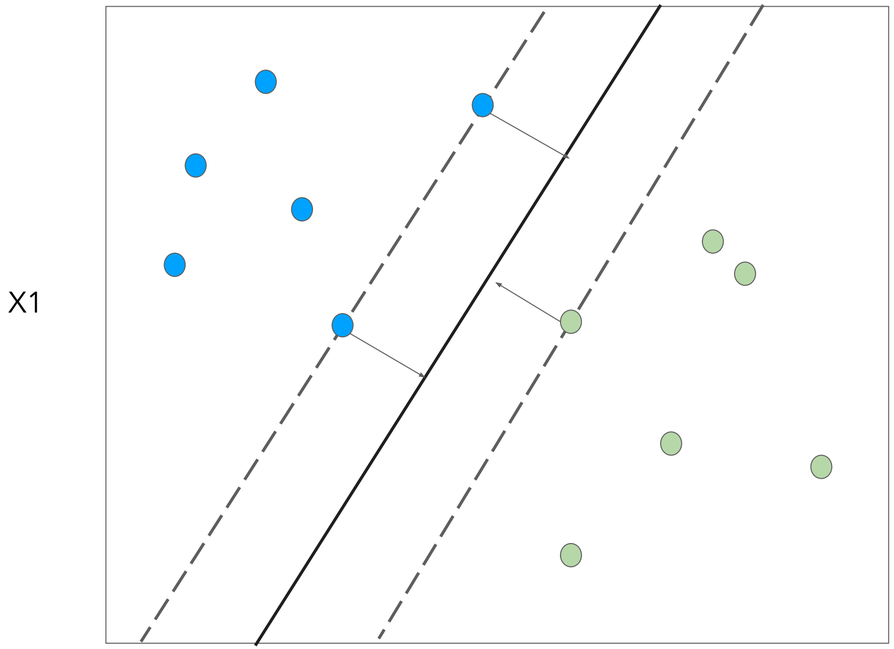
Maximo Margen Intuición

- En la regresión logística, tomamos la salida de la función lineal y aplastamos el valor dentro del intervalo [0,1] utilizando la función sigmoidea. Si el valor aplastado es superior a un valor umbral (0,5), le asignamos una etiqueta 1; de lo contrario, le asignamos una etiqueta 0.

- En SVM, tomamos la salida de la función lineal y si esa salida es mayor que 1, la identificamos con una clase y si la salida es -1, la identificamos con otra clase. Como los valores umbral se cambian a 1 y -1 en SVM, obtenemos este rango de valores de refuerzo([-1,1]) que actúa como margen.

Maximo Margen Intuición

La solución a este problema consiste en seleccionar como clasificador óptimo al que se conoce como *maximal margin hyperplane* o *hiperplano óptimo de separación*, que se corresponde con el hiperplano que se encuentra más alejado de todas las observaciones de entrenamiento. Para obtenerlo, se tiene que calcular la distancia perpendicular de cada observación a un determinado hiperplano. La menor de estas distancias (conocida como margen) determina como de alejado está el hiperplano de las observaciones de entrenamiento. El *maximal margin hyperplane* se define como el hiperplano que consigue un mayor margen, es decir, que la distancia mínima entre el hiperplano y las observaciones es lo más grande posible. Aunque esta idea suena razonable, no es posible aplicarla, ya que habría infinitos hiperplanos contra los que medir las distancias. En su lugar, se recurre a métodos de optimización. Para encontrar una descripción más detallada de la solución por optimización consultar (*Support Vector Machines Succinctly by Alexandre Kowalczyk*).

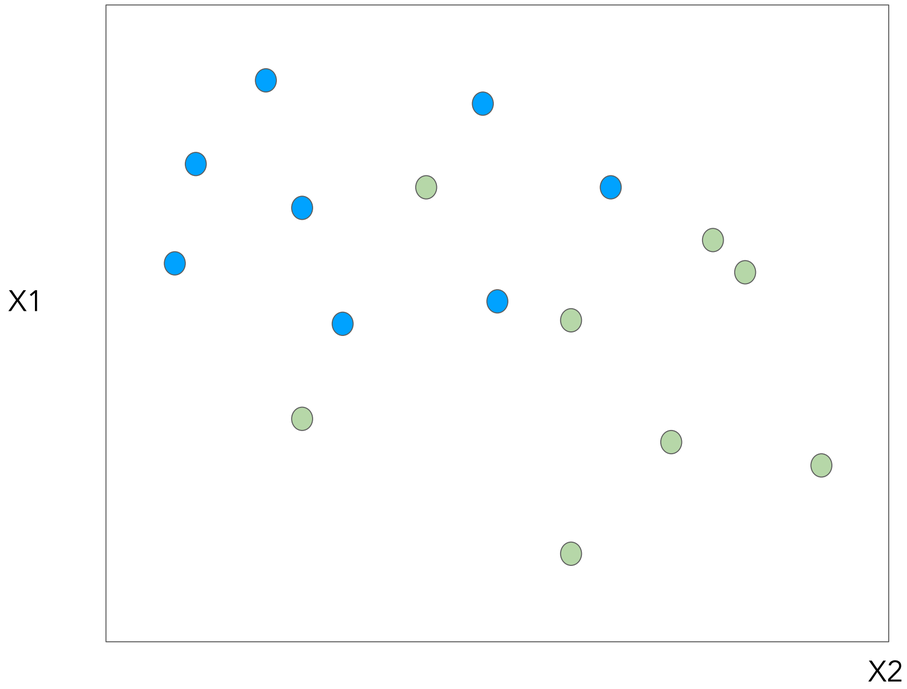


La imagen muestra el *maximal margin hyperplane* para un conjunto de datos de entrenamiento. Las tres observaciones equidistantes respecto al *maximal margin hyperplane* se encuentran a lo largo de las líneas discontinuas que indican la anchura del margen. A estas observaciones se les conoce como vectores soporte, ya que son vectores en un espacio *p*-dimensional y soportan (definen) el *maximal margin hyperplane*. Cualquier modificación en estas observaciones (vectores soporte) conlleva cambios en el *maximal margin hyperplane*. Sin embargo, modificaciones en observaciones que no son vector soporte no tienen impacto alguno en el hiperplano.

Datamecum

Casos cuasi separables linealmente

El *maximal margin hyperplane* descrito en el apartado anterior es una forma muy simple y natural de clasificación siempre y cuando exista un hiperplano de separación. En la gran mayoría de casos reales, los datos no se pueden separar linealmente de forma perfecta, por lo que no existe un hiperplano de separación y no puede obtenerse un *maximal margin hyperplane*.



Para solucionar estas situaciones, se puede extender el concepto de *maximal margin hyperplane* para obtener un hiperplano que casi separe las clases, pero permitiendo que cometa unos pocos errores. A este tipo de hiperplano se le conoce como *Support Vector Classifier* o *Soft Margin*.

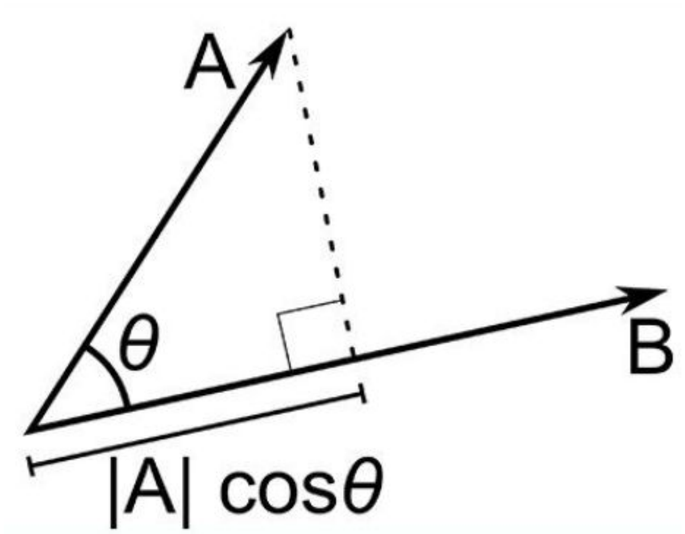
Programa Técnico Intensivo en Data Science

Intuición matemática

Datamecum

Dot product

- Todos sabemos que un vector es una cantidad que tiene magnitud y dirección y, al igual que los números, podemos utilizar operaciones matemáticas como la suma y la multiplicación. La diferencia producto escalar y producto vectorial estriba únicamente en que el producto escalar se utiliza para obtener un valor escalar como resultante, mientras que el producto cruzado se utiliza para obtener de nuevo un vector.



$$A \cdot B = |A| \cos\theta * |B|$$

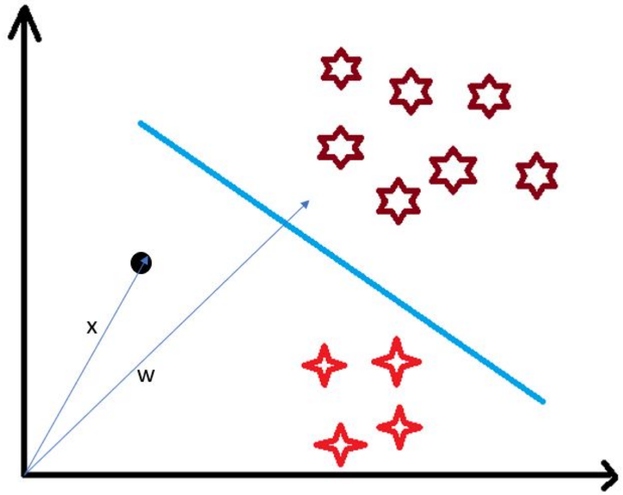
$$A.B = |A| \cos\theta * \text{unit vector of } B$$

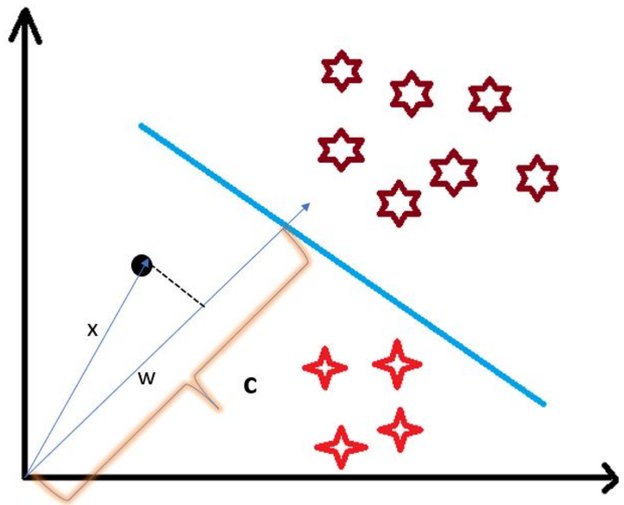
Datamecum

Uso del dot product en las SVM

- Consideremos un punto aleatorio X y queremos saber si se encuentra en el lado derecho del plano o en el lado izquierdo del plano (positivo o negativo).

- Para encontrarlo primero suponemos que este punto es un vector (X) y luego hacemos un vector (w) que es perpendicular al hiperplano. Digamos que la distancia del vector w desde el origen a la frontera de decisión es 'c'. Ahora tomamos la proyección del vector X sobre w.





- Ya sabemos que la proyección de cualquier vector sobre otro vector se llama producto punto. Por lo tanto, tomamos el producto punto de x y w vectores. Si el producto punto es mayor que 'c' entonces podemos decir que el punto se encuentra en el lado derecho. Si el producto punto es menor que 'c' entonces el punto se encuentra en el lado izquierdo y si el producto punto es igual a 'c' entonces el punto se encuentra en el límite de decisión.

Uso del dot product en las SVM

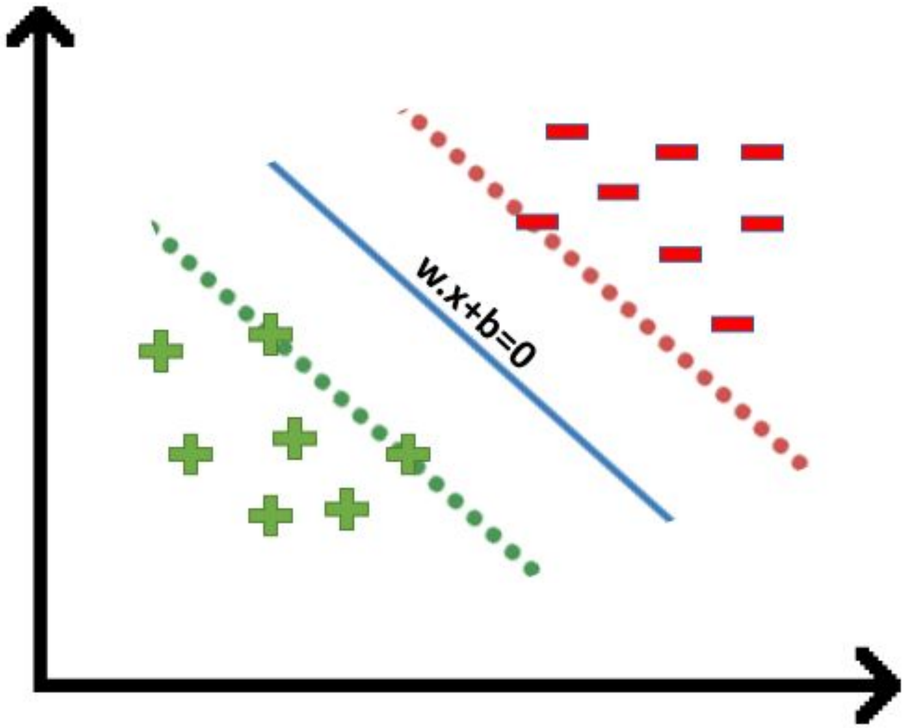
$$\overrightarrow{X}.\overrightarrow{w} = c \quad (the\ point\ lies\ on\ the\ decision\ boundary)$$

$$\overrightarrow{X}.\overrightarrow{w} > c\ (positive\ samples)$$

$$\overrightarrow{X}.\overrightarrow{w} < c\ (negative\ samples)$$

Margen en las SVM

Para clasificar un punto como negativo o positivo necesitamos definir una regla de decisión. Podemos definir la regla de decisión como



$$\vec{X}.\vec{w} - \mathrm{c} \geq 0$$

putting $-\mathrm{c}$ as b, we get

$$\vec{X}.\vec{w} + \mathrm{b} \geq 0$$

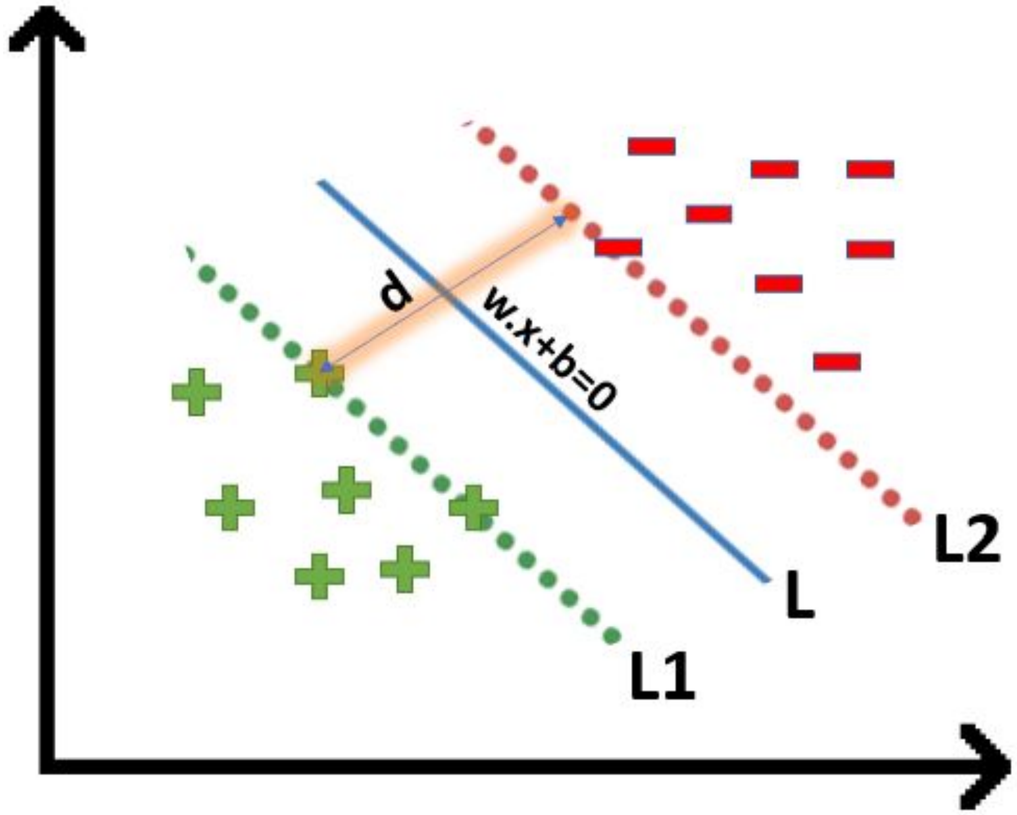
hence

$$\mathrm{y} = \begin{cases} +1 & \text{if } \vec{X}.\vec{w} + \mathrm{b} \geq 0 \\ -1 & \text{if } \vec{X}.\vec{w} + \mathrm{b} < 0 \end{cases}$$

Datamecum

Margen en las SVM

Si el valor de w.x+b>0 entonces podemos decir que es un punto positivo de lo contrario es un punto negativo. Ahora necesitamos (w,b) tal que el margen tenga una distancia máxima. Digamos que esta distancia es 'd'. Para calcular 'd' necesitamos la ecuación de L1 y L2. Para ello, supondremos que la ecuación de L1 es w.x+b=1 y la de L2 es w.x+b=-1.



Optimización y restricciones

Para obtener nuestra función de optimización, hay que tener en cuenta algunas restricciones. Esa restricción es que "Vamos a calcular la distancia (d) de tal manera que ningún punto positivo o negativo puede cruzar la línea de margen". Escribamos estas restricciones matemáticamente:

$$For\ all\ the\ Red\ points\ \vec{w}.\vec{X}+b \leq -1$$

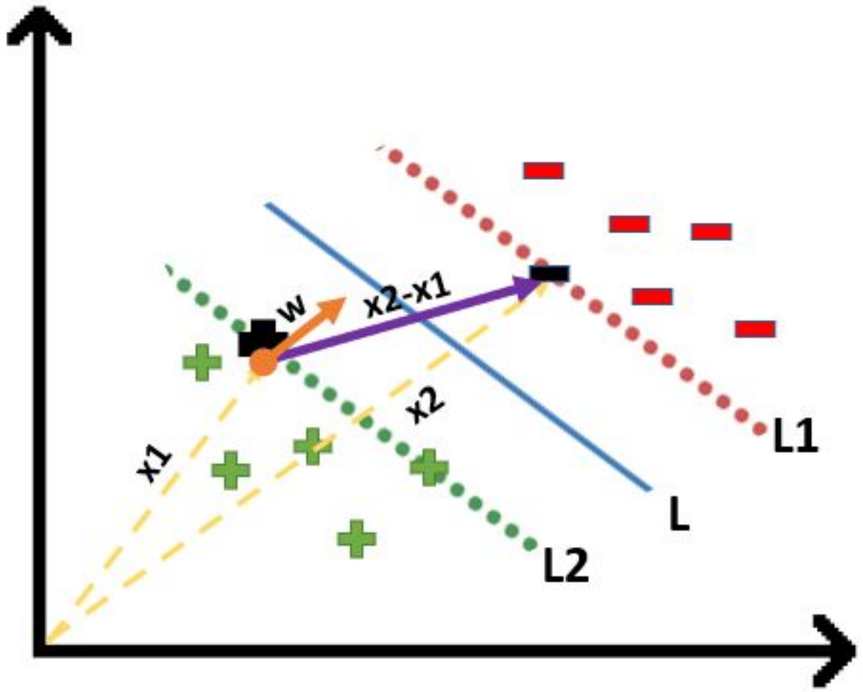
$$For\ all\ the\ Green\ points\ \vec{w}.\vec{X}+b \geq 1$$

En lugar de avanzar 2 restricciones, ahora intentaremos simplificar estas dos restricciones en 1. Suponemos que las clases negativas tienen y=-1 y las positivas y=1. Podemos decir que para que cada punto esté correctamente clasificado esta condición debe cumplirse siempre:

$$y_i(\vec{w}.\vec{X}+b) \geq 1$$

Margen en las SVM

Tomaremos 2 vectores soporte, 1º de la clase negativa y el 2º de la clase positiva. La distancia entre estos dos vectores x1 y x2 será (x2-x1) vector. Lo que necesitamos es, la distancia más corta entre estos dos puntos que se puede encontrar utilizando el producto escalar. Tomamos un vector 'w' perpendicular al hiperplano y a continuación hallamos la proyección del vector (x2-x1) sobre 'w'.



$$\Rightarrow (x2-x1).\frac{\vec{w}}{\|w\|}$$

$$\Rightarrow \frac{x2.\vec{w}-x1.\vec{w}}{\|w\|}$$

for positive point $y=1$

$$\Rightarrow 1\times(\vec{w}.x1+b) = 1$$

$$\Rightarrow \vec{w}.x1 = 1-b$$

Similarly for negative point $y=-1$

$$\Rightarrow -1\times\left(\vec{w}.x2+b\right) = 1$$

$$\Rightarrow \vec{w}.x2 = -b-1$$

FUNCIÓN DE OPTIMIZACIÓN

$$\text{argmax}(w^*,b^*)\ \frac{2}{\|w\|}\ \text{such that}\ y_i\left(\vec{w}.\vec{X}+b\right)\geq 1$$

Datamecum

Programa Técnico Intensivo en Data Science

Soft margin svm

Datamecum

Soft margin svm

- En aplicaciones de la vida real no encontramos ningún conjunto de datos que sea linealmente separable, lo que encontraremos es un conjunto de datos casi linealmente separable o un conjunto de datos no linealmente separable. En este caso, no podemos utilizar el truco que hemos probado antes, porque dice que sólo funcionará cuando el conjunto de datos sea perfectamente separable linealmente.

- Para hacer frente a este problema lo que hacemos es modificar la ecuación de tal manera que permita pocas clasificaciones erróneas, es decir, que permita clasificar erróneamente pocos puntos.

- Sabemos que max[f(x)] también se puede escribir como min[1/f(x)], es práctica común minimizar una función de coste para problemas de optimización; por lo tanto, podemos invertir la función.

$$\text{argmin}(\text{w}^*, \text{b}^*) \ \frac{\|\text{w}\|}{2} \text{ such that } \text{y}_\text{i}\left(\overrightarrow{w}.\overrightarrow{X} + b\right) \geq 1$$

Soft margin svm

- En aplicaciones de la vida real no encontramos ningún conjunto de datos que sea linealmente separable, lo que encontraremos es un conjunto de datos casi linealmente separable o un conjunto de datos no linealmente separable. En este caso, no podemos utilizar el truco que hemos probado antes, porque dice que sólo funcionará cuando el conjunto de datos sea perfectamente separable linealmente.

- Para hacer frente a este problema lo que hacemos es modificar la ecuación de tal manera que permita pocas clasificaciones erróneas, es decir, que permita clasificar erróneamente pocos puntos.

- Sabemos que max[f(x)] también se puede escribir como min[1/f(x)], es práctica común minimizar una función de coste para problemas de optimización; por lo tanto, podemos invertir la función.

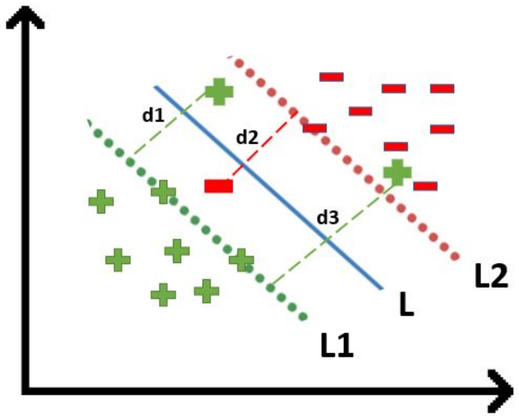
$$\text{argmin}(\text{w}^*, \text{b}^*) \ \frac{\|\text{w}\|}{2} \text{ such that } \text{y}_\text{i}\left(\vec{w}.\vec{X} + b\right) \geq 1$$

Para hacer una ecuación de margen suave añadimos 2 términos más a esta ecuación que es zeta y lo multiplicamos por un hiperparámetro 'c'

$$\text{argmin}\left(\text{w}^*, \text{b}^*\right) \ \frac{\|\text{w}\|}{2} \ + c\sum_{i=1}^{n}\zeta_i$$

Soft margin svm

Para todos los puntos correctamente clasificados nuestra zeta será igual a 0 y para todos los puntos incorrectamente clasificados la zeta es simplemente la distancia de ese punto en particular desde su hiperplano correcto, es decir, si vemos los puntos verdes mal clasificados el valor de zeta será la distancia de estos puntos desde el hiperplano L1 y para los puntos rojos mal clasificados zeta será la distancia de ese punto desde el hiperplano L2.



Así que ahora podemos decir que nuestros que son SVM Error = Error de margen + Error de clasificación. Cuanto mayor sea el margen, mayor será el error de margen, y viceversa. Digamos que usted toma un alto valor de 'c' =1000, esto significaría que no quiere centrarse en el error de margen y sólo quiere un modelo que no clasifica mal cualquier punto de datos.

Soft margin svm

Si alguien te pregunta cuál es mejor modelo, ¿el que tiene el margen máximo y tiene 2 puntos mal clasificados o el que tiene el margen muy reducido y todos los puntos están correctamente clasificados?

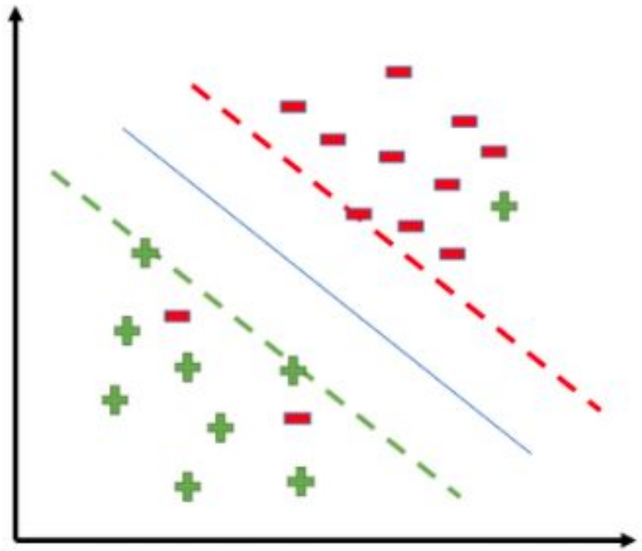


Figure 1

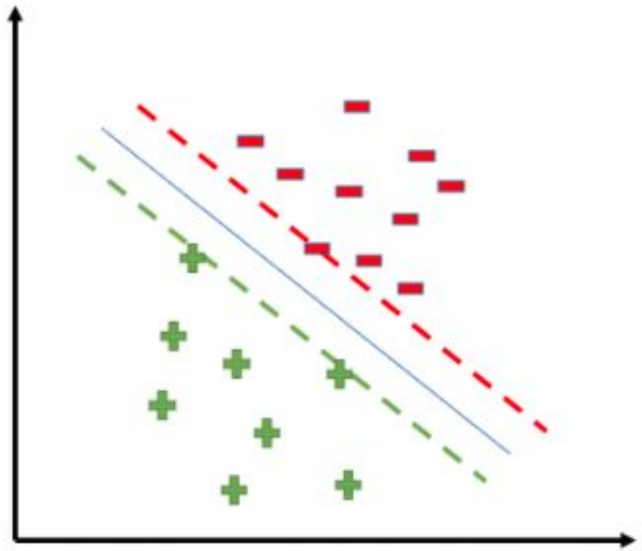


Figure 2

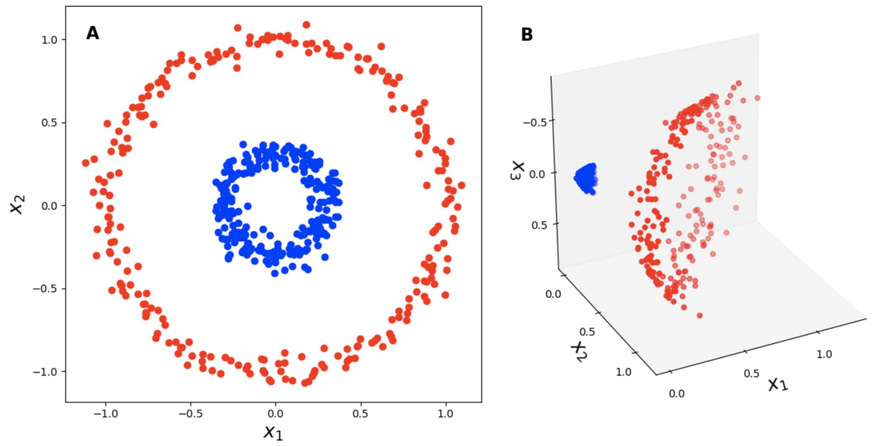
Programa Técnico Intensivo en Data Science

Kernel en las SVM

Datamecum

Kernels

La característica más interesante de SVM es que incluso puede trabajar con un conjunto de datos no lineal y para ello, utilizamos "Kernel Trick" que facilita la clasificación de los puntos. Supongamos que tenemos un conjunto de datos como este:



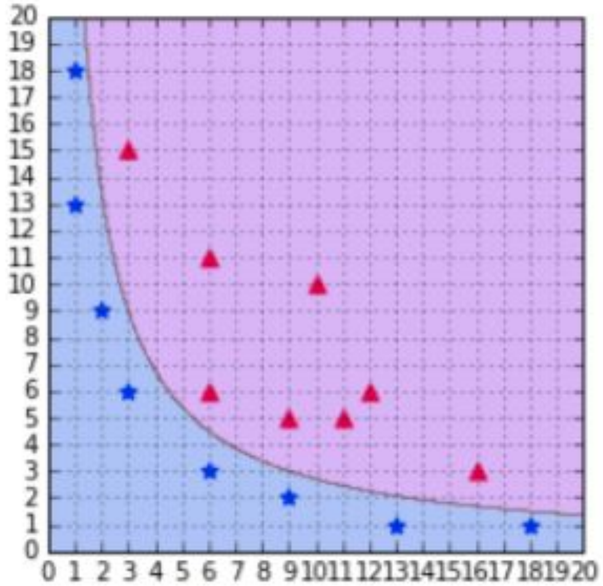
Aquí vemos que no podemos dibujar una sola línea o digamos hiperplano que pueda clasificar los puntos correctamente. Así que lo que hacemos es intentar convertir este espacio de dimensión inferior en un espacio de dimensión superior utilizando algunas funciones cuadráticas que nos permitirán encontrar un límite de decisión que divida claramente los puntos de datos. Estas funciones que nos ayudan a hacer esto se llaman Kernels y el kernel a utilizar viene determinado puramente por el ajuste de los hiperparámetros.

Funciones kernel: Polinomial kernel

A continuación se muestra la fórmula del núcleo polinómico:

$$f(X1, X2) = (X1^T . X2 + 1)^d$$

Aquí d es el grado del polinomio, que tenemos que especificar manualmente.



Funciones kernel: Sigmoid kernel

Podemos usarlo como sustituto de las redes neuronales. La ecuación es:

$$f(x1, x2) = \tanh(\alpha x^T y + x)$$

Es sólo tomar su entrada, mapearlos a un valor de 0 y 1 para que puedan ser separados por una simple línea recta.

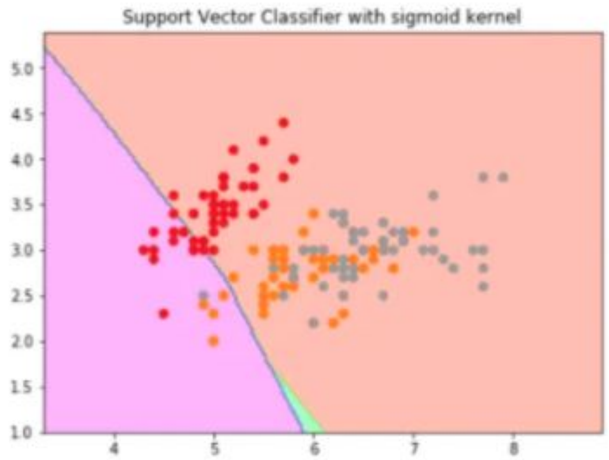


Image Source: https://dataaspirant.com/svm-kernels/#t-1608054630725

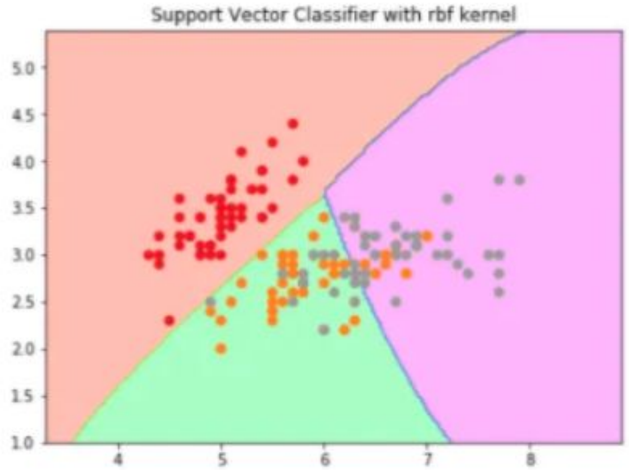
Funciones kernel: Sigmoid kernel

Lo que realmente hace es crear combinaciones no lineales de nuestras características para elevar sus muestras a un espacio de características de mayor dimensión donde podemos utilizar un límite de decisión lineal para separar sus clases Es el kernel más utilizado en las clasificaciones SVM, la siguiente fórmula lo explica matemáticamente:

$$f(x1, x2) = e^{\frac{-||(x1 - x2)||^2}{2\sigma^2}}$$

donde,

1. 'σ' es la varianza y nuestro hiperparámetro
2. $||X_1 - X_2||$ es la distancia euclídea entre dos puntos X_1 y X_2.



Datamecum

¿Cómo elegir el kernel correcto?

- Es necesario elegir una buena función kernel porque de ella depende el rendimiento del modelo.
- La elección de un kernel depende totalmente del tipo de conjunto de datos con el que se esté trabajando. Si es linealmente separable, entonces debe optar por la función kernel lineal, ya que es muy fácil de usar y la complejidad es mucho menor en comparación con otras funciones kernel.
- Por lo general, utilizamos SVM con RBF y la función de kernel lineal porque otros kernels como el kernel polinomial rara vez se utilizan debido a la escasa eficiencia. Pero ¿qué pasa si lineal y RBF ambos dan resultados aproximadamente similares? ¿Qué kernel elegimos ahora?
- El SVM lineal es un modelo paramétrico. Un modelo paramétrico es un concepto utilizado para describir un modelo en el que todos sus datos están representados dentro de sus parámetros. En resumen, la única información necesaria para predecir el futuro a partir del valor actual son los parámetros.
- La complejidad del kernel RBF crece a medida que aumenta el tamaño de los datos de entrenamiento. Además de que es más caro preparar el kernel RBF, también tenemos que mantener la matriz del kernel alrededor, y la proyección en este espacio "infinito" de mayor dimensión donde los datos se vuelven linealmente separables también es más cara durante la predicción.

Programa Técnico Intensivo en Data Science

SVM para más de dos clases

Datamecum

SVM para más de dos clases

El concepto de hiperplano de separación en el que se basan los *SVMs* no se generaliza de forma natural para más de dos clases. Se han desarrollado numerosas estrategias con el fin de aplicar este método de clasificación a situaciones con $k>2$-clases, de entre ellos, los más empleados son: *one-versus-one*, *one-versus-all* y *DAGSVM*.

One-vs-One

- Supóngase un escenario en el que hay $K > 2$ clases y que se quiere aplicar el método de clasificación basado en *SVMs*.

- La estrategia de *one-versus-one* consiste en generar un total de *K(K-1)/2 SVMs*, comparando todos los posibles pares de clases. Para generar una predicción se emplean cada uno de los *K(K-1)/2* clasificadores, registrando el número de veces que la observación es asignada a cada una de las clases.

- Finalmente, se considera que la observación pertenece a la clase a la que ha sido asignada con más frecuencia.

- La principal desventaja de esta estrategia es que el número de modelos necesarios se dispara a medida que aumenta el número de clases, por lo que no es aplicable en todos los escenarios.

One-vs-All

- Esta estrategia consiste en ajustar *K SVMs* distintos, cada uno comparando una de las *K* clases frente a las restantes *K-1* clases. Como resultado, se obtiene un hiperplano de clasificación para cada clase.

- Para obtener una predicción, se emplean cada uno de los *K* clasificadores y se asigna la observación a la clase para la que la predicción resulte positiva.

- Esta aproximación, aunque sencilla, puede causar inconsistencias, ya que puede ocurrir que más de un clasificador resulte positivo, asignando así una misma observación a diferentes clases.

- Otro inconveniente adicional es que cada clasificador se entrena de forma no balanceada. Por ejemplo, si el set de datos contiene 100 clases con 10 observaciones por clase, cada clasificador se ajusta con 10 observaciones positivas y 990 negativas.