

Diplomatura en Python orientado a científico de datos

APRENDIZAJE SUPERVISADO
MÁQUINAS DE SOPORTE VECTORIAL

Ing. Claudia Piaggio

Aprendizaje Supervisado

Máquinas de Soporte Vectorial



Aprendizaje Supervisado

Máquinas de Soporte Vectorial

Agenda

- ▶ Introducción a las SVM
- ▶ Conceptos Tempranos:
 - ▶ Hiperplanos
 - ▶ Transformaciones Lineales
 - ▶ Truco del Kernel
- ▶ SVM: Márgenes y Kernels
- ▶ Kernels Sklearn
- ▶ SVM: Clasificadores, Regresores y Deteccion de Outliers de Sklearn
- ▶ Ejemplo de Clasificación con el Dataset de Iris



Aprendizaje Supervisado

Máquinas de Soporte Vectorial

La Máquina de Soporte Vectorial (SVM) es un algoritmo que puede ser utilizado como:

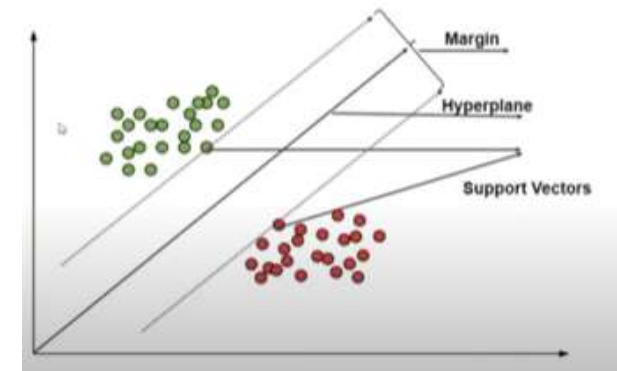
- ▶ Clasificador
- ▶ Regresor
- ▶ Detector de Outliers

Es un algoritmo que involucra el seteo de varios hiperparámetros

Aprendizaje Supervisado

Máquinas de Soporte Vectorial

- ▶ SVM es un clasificador discriminativo que está diseñado formalmente por un **hiperplano** de separación.
- ▶ Es una representación de **muestras** como **puntos** en el espacio, que se mapean para que los puntos de diferentes categorías estén separados por un espacio lo más amplio posible.
- ▶ Las muestras son los vectores (filas del dataset)



Máquinas de Soporte Vectorial

Conceptos Tempranos

¿ Que es un Hiperplano?

Un hiperplano es la generalización del concepto de Plano.

Recordemos un poco.

El **plano**, en geometría, es el ente ideal que sólo posee dos dimensiones, y contiene infinitos puntos y rectas;

Máquinas de Soporte Vectorial

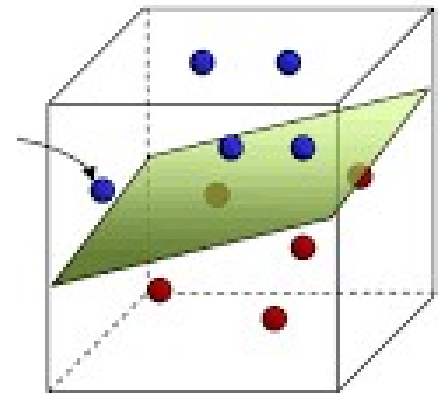
Conceptos Tempranos

En un espacio unidimensional (como una recta), un hiperplano es un punto: divide una línea en dos líneas.

En un espacio bidimensional (como el plano xy), un hiperplano es una recta: divide el plano en dos mitades.

En un espacio tridimensional, un hiperplano es un plano "común": divide el espacio en dos mitades.

En espacios de cuatro dimensiones y más, estos objetos divisores se llaman simplemente hiperplanos (su dimensión será $n-1$)

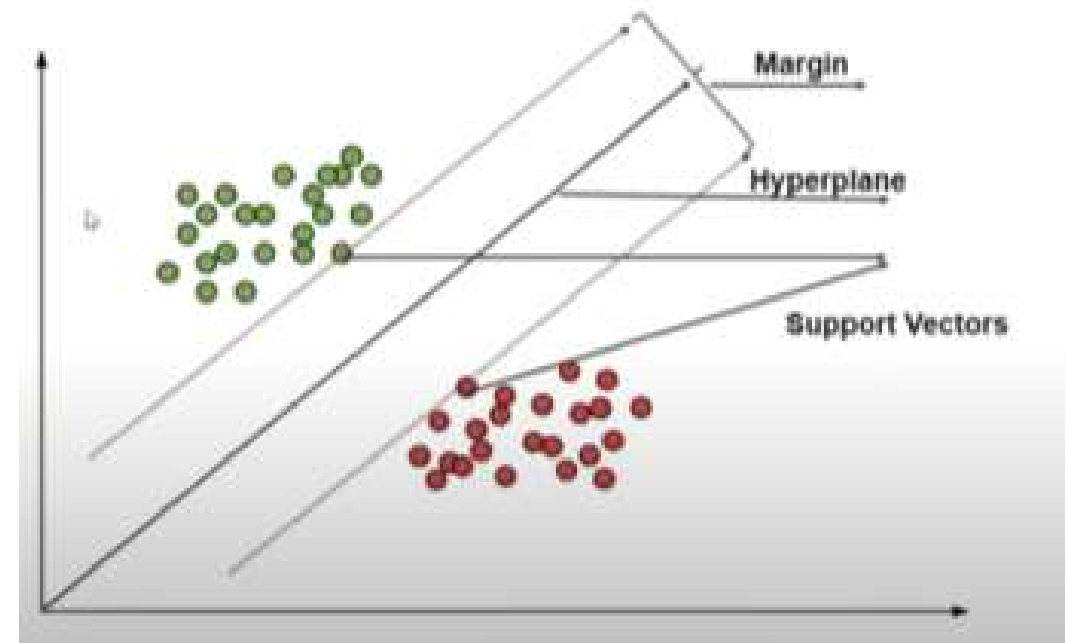


Máquinas de Soporte Vectorial

Conceptos Tempranos

¿Cómo funciona?

- ▶ Los puntos rojos y verdes pertenecen a clases diferentes.
- ▶ Dibujo un hiperplano que separa los puntos de las distintas clases lo mas posible.
- ▶ El hiperplano es equidistante de los "Vectores de Soporte"
- ▶ Vectores de Soporte: puntos mas cercanos al hiperplano.

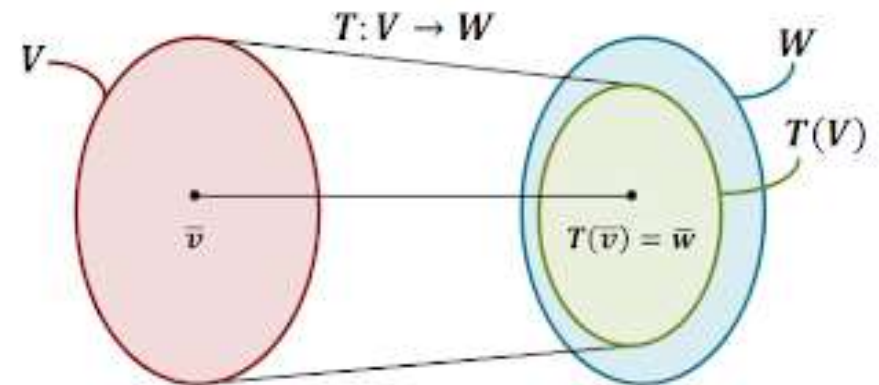


Máquinas de Soporte Vectorial

Conceptos tempranos

Transformación Lineal

- ▶ Son transformaciones entre espacios vectoriales (el dominio y la imagen o codominio de la función son vectores) que conservan la suma vectorial y la multiplicación por escalar.
- ▶ La Transformación $T: V \rightarrow W$ es lineal sí y sólo si:
 - ▶ $T(x+y) = T(x) + T(y)$
 - ▶ $T(kx) = k.T(x)$



Las transformaciones tienen tres partes esenciales: el dominio (V), el codominio (W), y la regla de asignación (T)

Máquinas de Soporte Vectorial

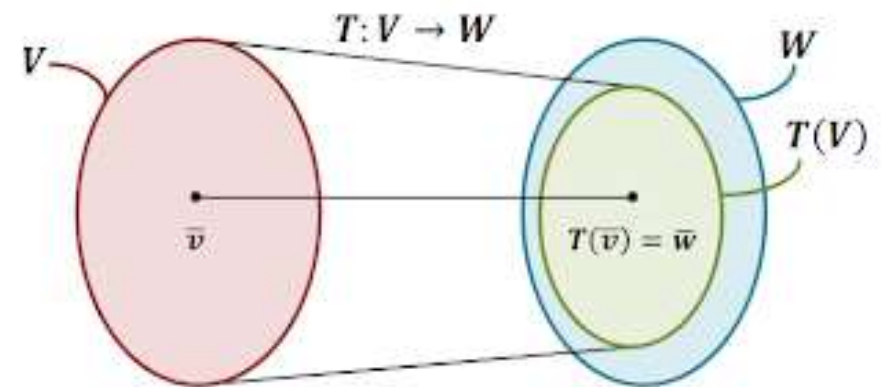
Conceptos tempranos

Transformación Lineal

En las TL existe un subconjunto especial llamado **núcleo**.

El **núcleo** es parte del dominio y es el conjunto de vectores de V cuya transformación bajo T tiene como único resultado al **vector nulo** del espacio W (codominio).

Dada $T: V \rightarrow W$, si existe un subconjunto $U \subset V$ tal que $T(U) = \{0_W\} \Rightarrow$ el subconjunto U es el núcleo de la transformación T .



Máquinas de Soporte Vectorial

No linealidad – Truco del Kernel

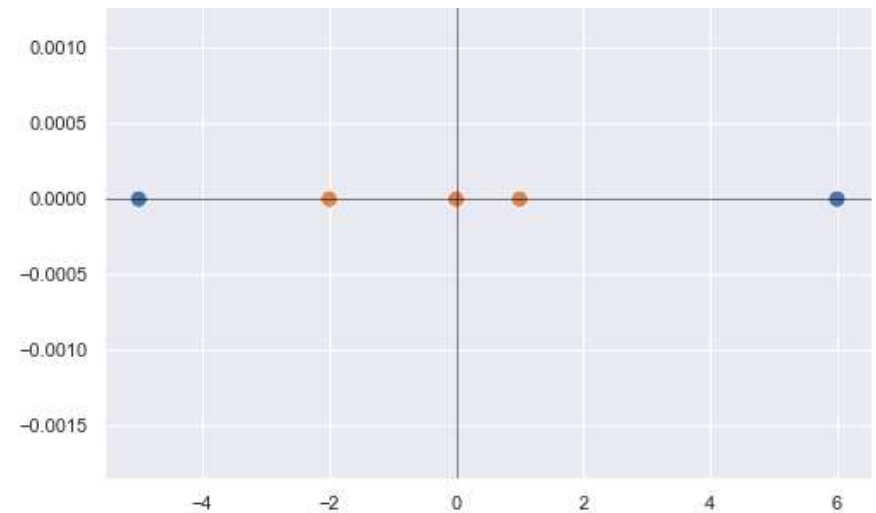
Truco del Kernel (Kernel Trick)

¿Qué pasa si mis puntos (vectores) a clasificar no son linealmente separables?

No voy a encontrar ningún hiperplano lineal para poder separarlos!

Por ej, en la figura no podemos separar los puntos azules de los naranjas con una recta

Acá viene el “truco del kernel”



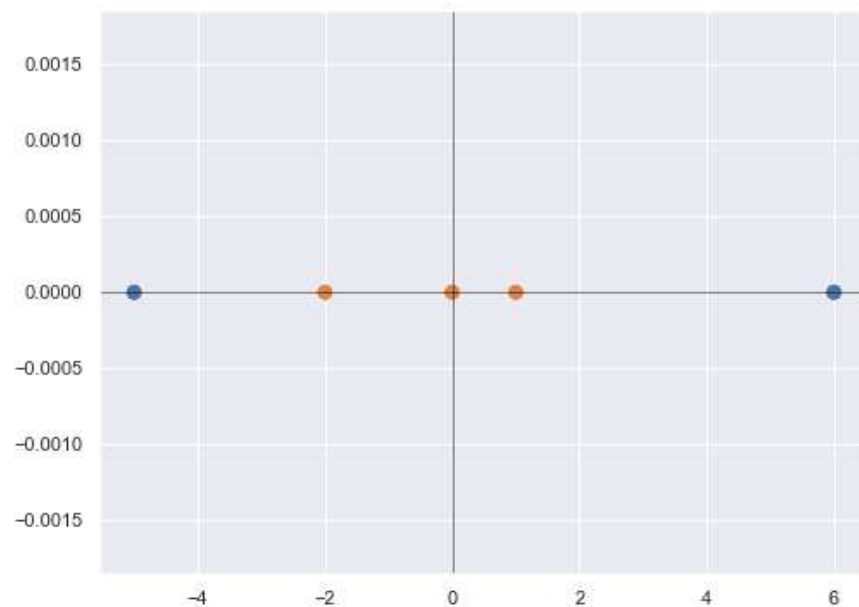
Máquinas de Soporte Vectorial

No linealidad –Truco del Kernel

- ▶ El truco del kernel consiste en transformar los puntos para llevarlos a un espacio de mayor dimensión tras aplicarles alguna transformación.
- ▶ En nuestro ejemplo supongamos que llevamos los puntos de \mathbb{R} a \mathbb{R}^2 , conservando las coordenadas X con su valor Original y asignándoles el cuadrado de ellas a las coordenadas Y .

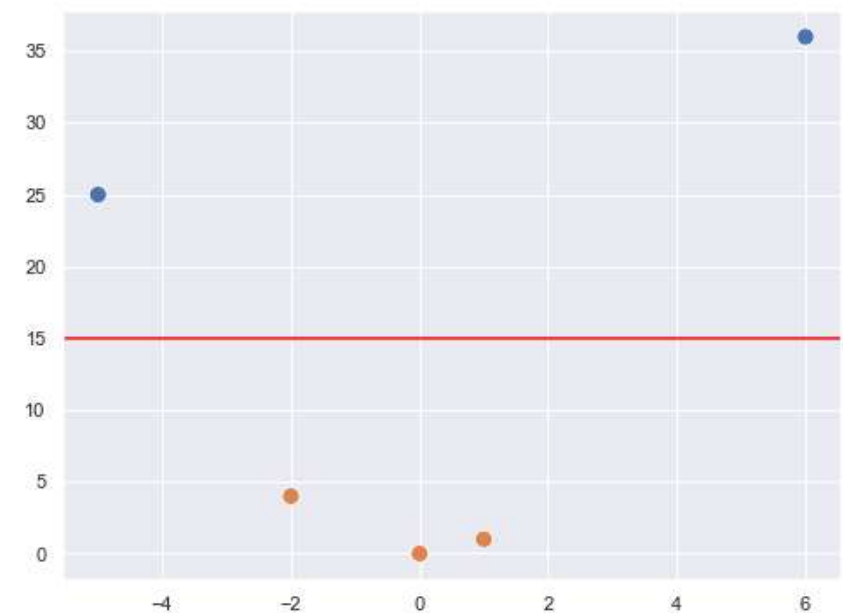
Máquinas de Soporte Vectorial

No linealidad –Truco del Kernel



Espacio Original: $x = [-5, -2, 0, 1, 6]$

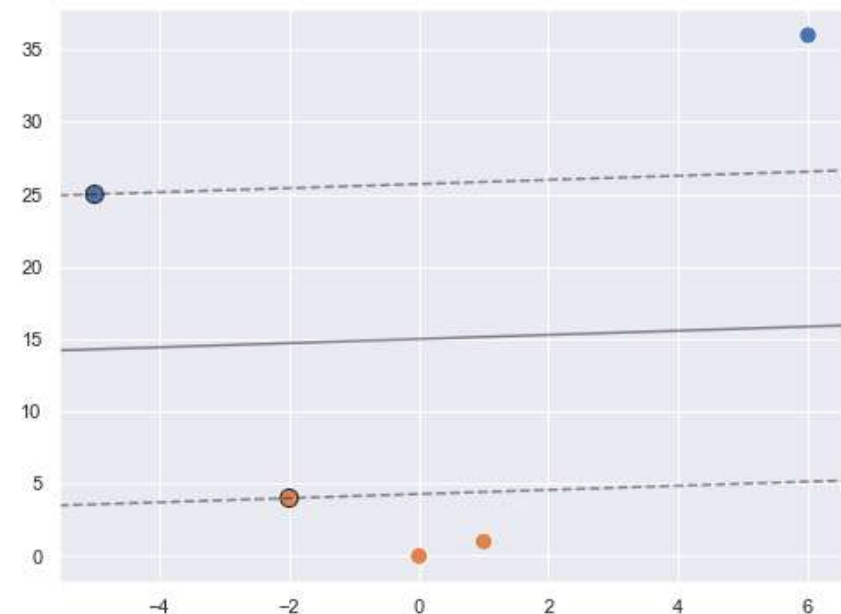
Espacio Transformado: $x = [-5, -2, 0, 1, 6]$ $y = [25, 4, 0, 1, 36]$



Máquinas de Soporte Vectorial

No linealidad –Truco del Kernel

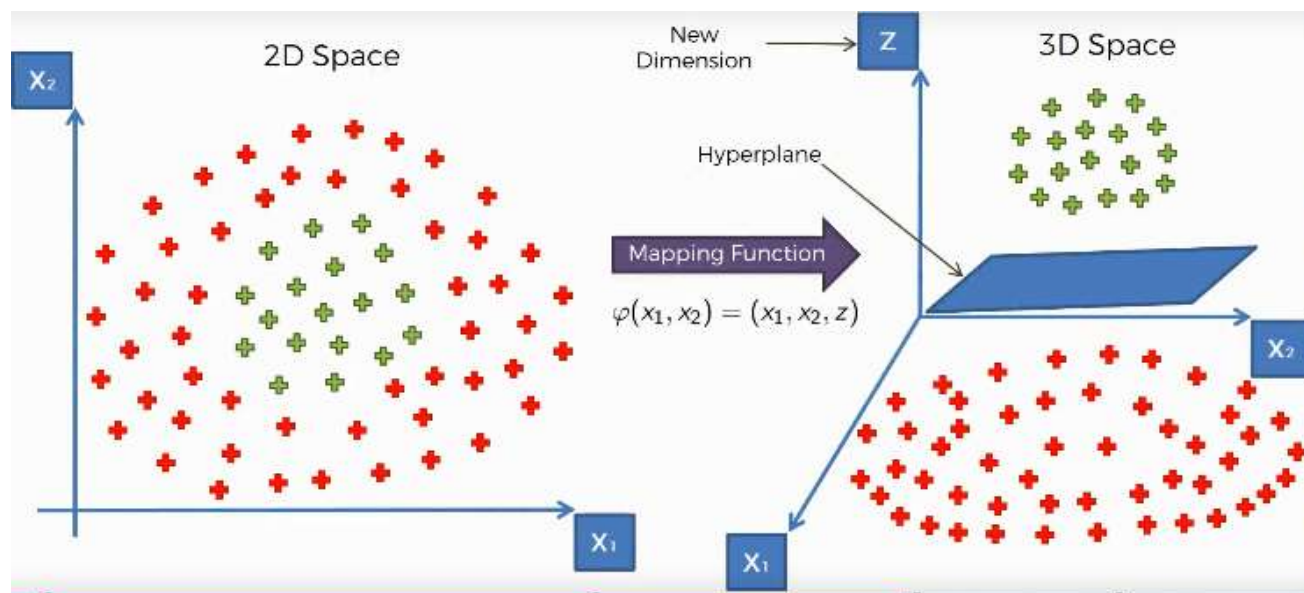
- ▶ En el nuevo espacio, sí es posible crear una recta que separe ambos grupos de puntos, por el ejemplo la recta roja que se muestra .
- ▶ SVM buscaría la recta que maximizase la distancia a ambos grupos de puntos en este nuevo espacio:



Máquinas de Soporte Vectorial

No linealidad –Truco del Kernel

- Veamos el efecto del truco del Kernel con otra distribución de puntos:



Máquinas de Soporte Vectorial

Kernels

El propósito del kernel es mapear los datos a un espacio de mayor dimensionalidad donde pueda existir una separación lineal más clara entre las clases.

En lugar de realizar la transformación de los datos en un espacio de mayor dimensionalidad, el kernel calcula el producto interno entre los vectores de características en ese espacio de mayor dimensión.

Diferentes tipos de kernels pueden producir diferentes resultados en términos de la capacidad del modelo para separar las clases.

Máquinas de Soporte Vectorial

Tipos de Kernels

- ▶ 1. Kernel lineal: Es el kernel más simple y se utiliza cuando se supone que los datos son linealmente separables en el espacio de características original.
- ▶ 2. Kernel polinómico: Transforma los datos en un espacio de mayor dimensión utilizando funciones polinómicas. Puede manejar datos que no son linealmente separables.
- ▶ 3. Kernel RBF (Función de Base Radial): También conocido como kernel Gaussiano, mapea los datos en un espacio de características infinitamente dimensional utilizando una distribución de probabilidad Gaussiana. Es útil para separar clases no linealmente separables.
- ▶ 4. Kernel sigmoidal: Este kernel se utiliza en situaciones donde los datos no son linealmente separables y se asemeja a la función de activación sigmoidal utilizada en redes neuronales.

Máquinas de Soporte Vectorial

Márgenes y Kernels

Márgenes Duros y Blandos

Si luego de aplicar el Kernel Trick, los puntos a clasificar son linealmente separables, podremos identificar el hiperplano de máximo margen que los separa maximizando las distancias a las clases.

En estas circunstancias no se estarán produciendo errores en la clasificación.

Sin embargo, si las clases no son linealmente separables, no será posible encontrar el hiperplano en cuestión.

Máquinas de Soporte Vectorial

Márgenes y Kernels

Márgenes Duros y Blandos

- ▶ Con el término "*margen duro*" hacemos referencia al escenario en el que no se permiten errores en el entrenamiento: si se encuentra un hiperplano de máximo margen es porque clasifica correctamente todas las muestras.
- ▶ Este enfoque, no funcionará si las clases no son linealmente separables, escenarios en los que puede resultar más práctico permitir ciertos errores en la clasificación a cambio de poder seguir encontrando el hiperplano de máximo margen. Este segundo enfoque es el que denominamos "*de margen blando*".

Máquinas de Soporte Vectorial

Márgenes y Kernels

Márgenes Duros y Blandos

- ▶ Este concepto -que determina la función de coste del algoritmo- está regulado con el parámetro **C**: un valor mayor de **C** implica un coste mayor derivado de las muestras mal clasificadas.
- ▶ Un valor de **C** menor implica que las muestras mal clasificadas suponen un coste menor, por lo que se permite un mayor número de errores. Se tiende a un escenario en el que se permite un mayor número de errores.

Máquinas de Soporte Vectorial

Kernels Scikit-learn

Scikit-learn ofrece 4 kernels básicos. En las fórmulas x_i y x_j representan los valores de la matriz de coeficientes del conjunto de entrenamiento, X^T la transpuesta de X , $\|X\|$ representa la norma L2 de X , y el punto (.) representa el producto de dos matrices.

► Lineal: $K(x_i, x_j) = x_i^T \cdot x_j$

► Polinómica $K(x_i, x_j) = (\gamma x_i^T \cdot x_j + r)^d$

y viene dada por el parámetro **gamma** de la función SVC, r por **coef0** y d por el parámetro **degree**.

► Radial: $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$

► Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T \cdot x_j + r)$

Máquinas de Soporte Vectorial

Clasificadores SKLearn

La implementación de SVM de Sklearn ofrece los siguientes **clasificadores**:

- ▶ `sklearn.svm.SVC`: SVC es la clase principal ofrecida por Scikit-learn. La función de coste viene determinada por el parámetro **C**. El parámetro **kernel** especifica el tipo de kernel a usar. Las implementaciones nativas son "*linear*", "*poly*", "*rbf*" y "*sigmoid*". También permite usar kernels customizados.
- ▶ `sklearn.svm.NuSVC`: En vez de emplear el parámetro C, utiliza el parámetro **nu** que controla el **número de vectores de soporte**.
- ▶ `sklearn.svm.LinearSVC`: Igual a SVC pero con el parámetro `kernel="linear"` (único kernel soportado). Según la documentación de Scikit-learn, esta implementación debería ofrecer mejor rendimiento que SVC cuando se trabaja con gran número de muestras.

Máquinas de Soporte Vectorial

Clasificadores SKLearn

- ▶ C es un parámetro de regularización que controla el equilibrio entre el ajuste a los datos de entrenamiento y la generalización del modelo a datos nuevos.
- ▶ C es inversamente proporcional a la regularización. Un valor más alto de C significa una regularización más débil, lo que permite que el modelo se ajuste más a los datos de entrenamiento incluso si eso implica una mayor complejidad del modelo. Por otro lado, un valor más bajo de C impone una mayor regularización, lo que fomenta un modelo más simple y generalizable.
- ▶ La elección adecuada del valor de C generalmente se realiza mediante técnicas de validación cruzada o búsqueda de hiperparámetros, donde se evalúa el rendimiento del modelo en diferentes valores de C y se selecciona aquel que brinde el mejor equilibrio entre ajuste y generalización.

Máquinas de Soporte Vectorial

Regresores: Support Vector Regression

Sklearn ofrece los siguientes **regresores** con SVM:

- ▶ SVR: Utiliza C para ajuste de la función de coste. El **kernel** a implementar puede ser “linear”, “poly”, “rbf”, “sigmoid”, “precomputed”.
- ▶ NuSVR: para la función de ajuste utiliza el parámetro “nu” que representa el número de vectores de soporte. (reemplaza al parámetro ϵ)
- ▶ LinearSVR. proporciona una implementación más rápida que SVR pero solo considera el kernel lineal.

Máquinas de Soporte Vectorial

Outliers: Support Vector Regression

- ▶ La clase **OneClassSVM** implementa una SVM que se utiliza en la detección de valores atípicos.

Máquinas de Soporte Vectorial

