

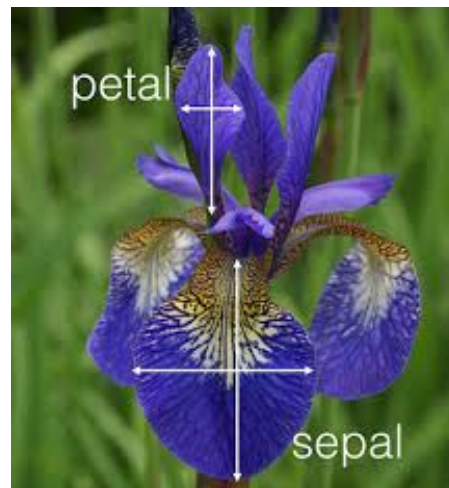
# Aprendizaje Automático

Clase 02 - ¿Cómo aprende un programa?

# Regresión lineal

Para entender cómo aprende un programa, vamos a partir con un ejemplo sencillo: hacer una regresión lineal

Vamos a partir por el *hello world* en el área de *Machine Learning*: el *dataset* de las flores Iris



# Regresión lineal

Este dataset se ve como una tabla con las siguientes columnas:

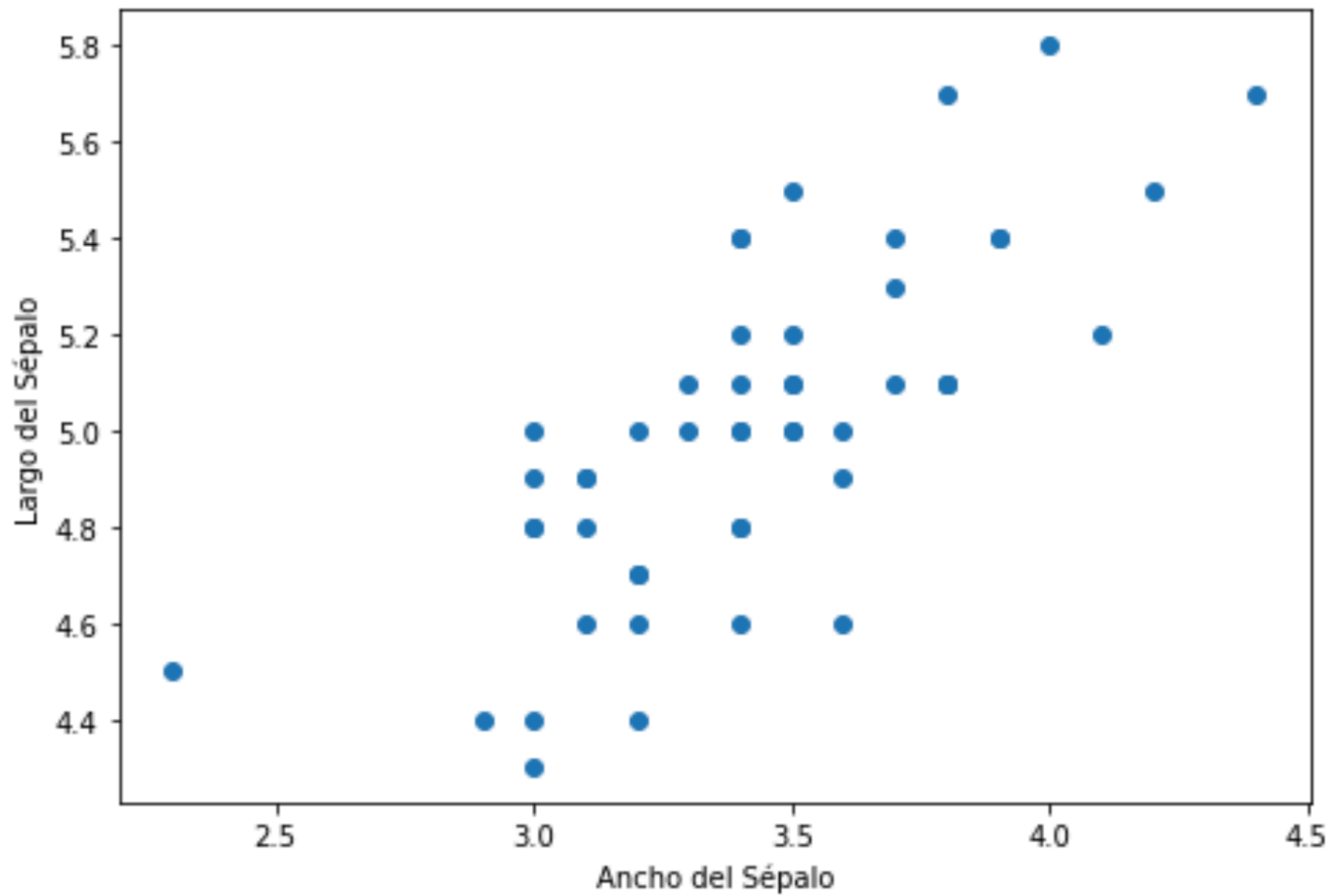
Largo del sépalo	Ancho del sépalo	Largo del pétalo	Ancho del pétalo	Tipo de flor
5.1	3.5	1.4	0.2	Iris Setosa
4.9	3.0	1.4	0.2	Iris Setosa
4.7	3.2	1.3	0.2	Iris Setosa
...	...	...	...	...

# Regresión lineal

En este ejemplo vamos a predecir los valores del **largo del pétalo** en función del ancho del pétalo para las flores de tipo **Setosa**

Un buen punto de partida es graficar en un plano las flores Iris Setosa, dejando en el eje  $x$  el ancho del pétalo y en el eje  $y$  el largo

# Regresión lineal

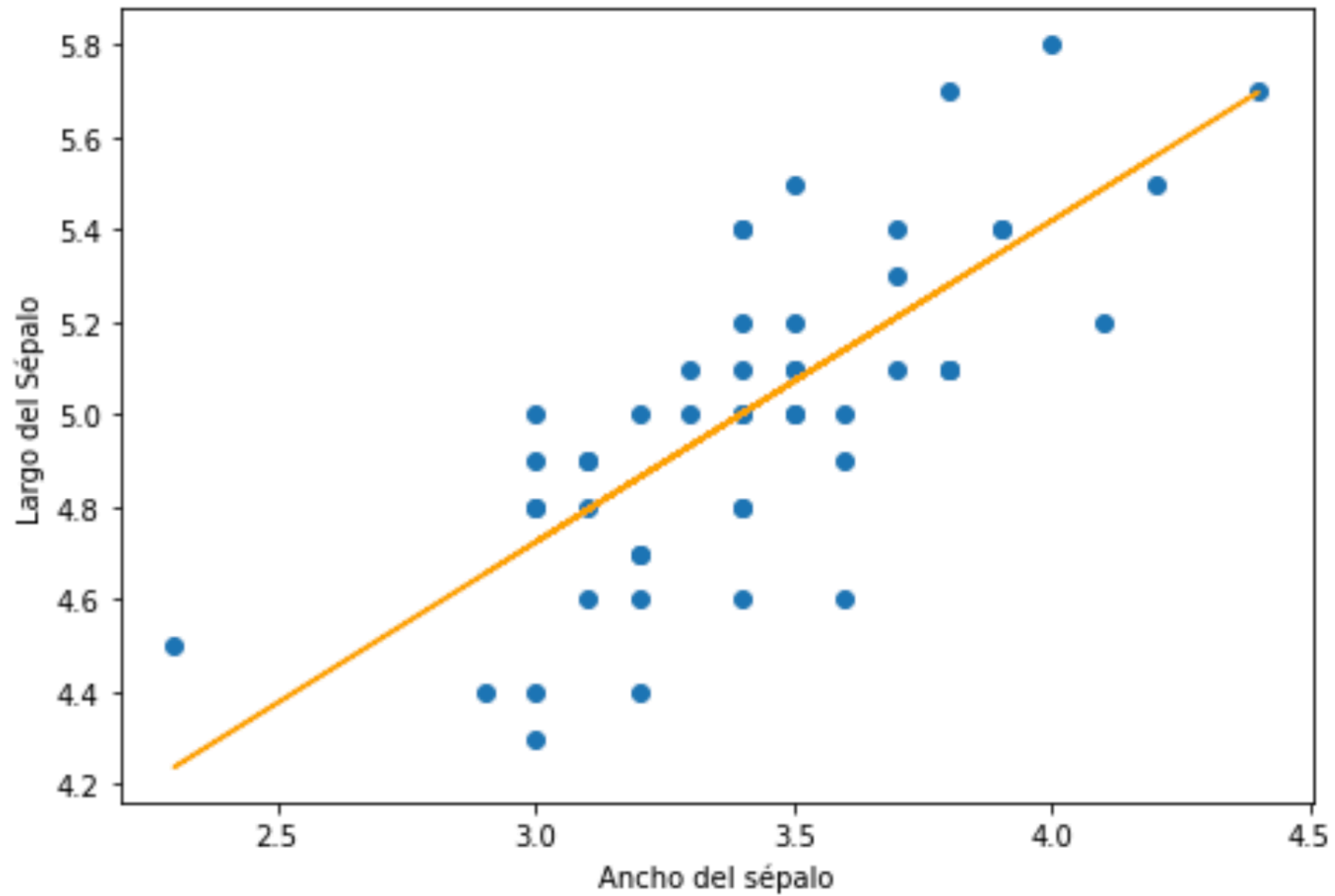


# Regresión lineal

Ahora supongamos que no conocemos el **largo del pétalo** para una flor donde el ancho del pétalo es 5

¿Cómo podemos predecir su largo?

# Regresión lineal



# Regresión lineal

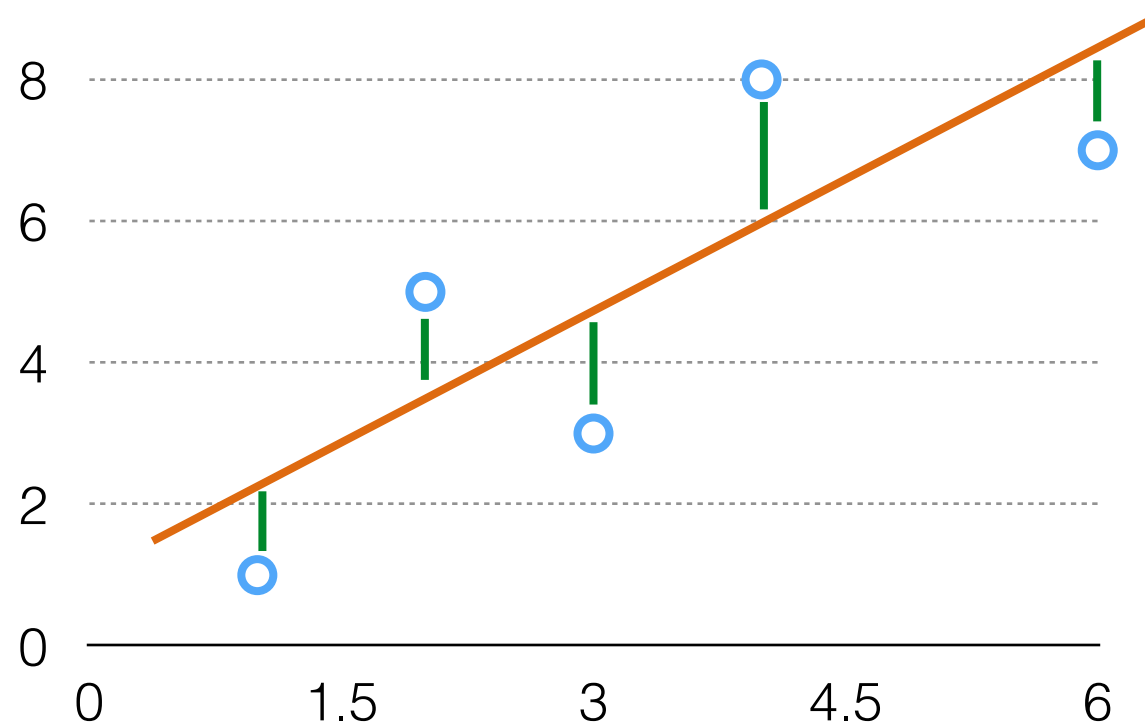
Podemos ajustar una regresión lineal, y ver el valor estimado para cuando el ancho del pétalo vale 5

Lo que estamos haciendo es descubrir los valores de  $\hat{m}$  y  $\hat{n}$  de forma tal que la recta  $\hat{y} = \hat{m}x + \hat{n}$  pasa lo más cerca posible de todos los puntos del *dataset*



# Regresión lineal

Para lograr esto, buscamos la recta que minimiza los errores



# Regresión lineal

Matemáticamente, queremos resolver el siguiente problema de optimización:

$$\min \sum_{i=1}^m |y_i - \hat{y}_i| = \min \sum_{i=1}^m |y_i - (\hat{m}x_i + \hat{n})|$$

Donde nuestro dataset tiene  $m$  filas (no confundir con  $\hat{m}$  que es la pendiente de la recta que queremos aprender)

Además  $y_i$  es el valor observado para la fila  $i$ , mientras que  $\hat{y}_i$  es el valor que vamos a predecir para esa fila

# Regresión lineal

¿Cómo resolvemos este problema de optimización?

# Regresión lineal

Para resolver el problema tenemos muchas opciones, por ejemplo, derivar en función de  $\hat{m}$  y  $\hat{n}$ , igualar a 0 y resolver el sistema de ecuaciones; con eso obtenemos la fórmula:

$$\hat{m} = \frac{\sum_{i=1}^n (x_i y_i - \bar{y} x_i)}{\sum_{i=1}^n (x_i^2 - \bar{x} x_i)} \qquad \hat{n} = \bar{y} - \hat{m} \bar{x}$$

Donde  $\bar{x}$  y  $\bar{y}$  es el promedio de todos los  $x$  e  $y$  respectivamente

# Regresión lineal

En resumen, lo que hicimos fue **aprender** los valores de  $\hat{m}$  y  $\hat{n}$  en función de nuestros datos

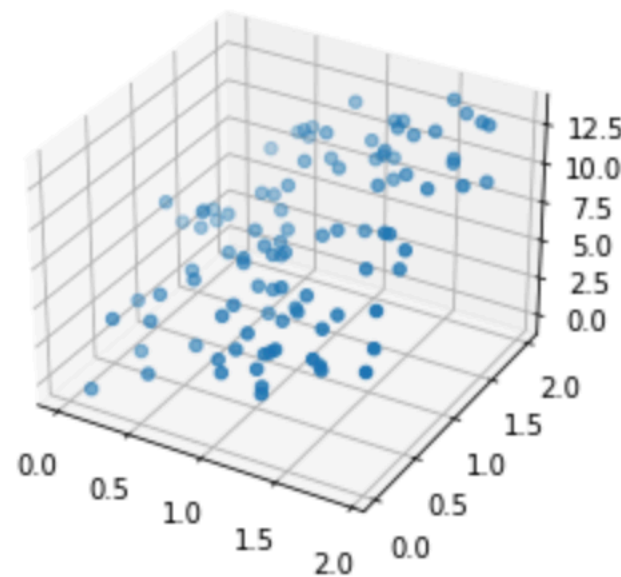
Esta idea es fundamental en el área de *Machine Learning*

# Regresión lineal

¿Qué pasa si queremos predecir en función de más variables? Por ejemplo en este caso podría ser los valores del sépalos junto con el ancho del pétalo

# Regresión lineal

El modelo de regresión lineal se puede extender para más variables; por ejemplo, si queremos obtener un valor  $z$  en función de valores  $x$  e  $y$ , podemos generar un plano que pasa en 3 dimensiones



# Regresión polinomial

Además, podemos generar modelos para ajustar polinomios en vez de rectas

Si bien el modelo se ajustará más a los datos, existe el riesgo de generar modelos sobre-ajustados (*overfitted*)



# Clasificación

Ahora, ¿qué pasa si dado los datos de una flor queremos predecir su tipo?

En este *dataset* los tipos son Iris Setosa, Versicolor y Virginica

**Primera idea.** Usar un modelo de regresión y asignar para cierto rango de valores un tipo

# Clasificación

El problema es que para clasificar, nos gustaría tener un valor acotado, por ejemplo, una probabilidad de que una instancia pertenezca a una clase

**Posible solución.** El resultado de la regresión puede pasar por la función  $\sigma : \mathbb{R} \rightarrow [0,1]$

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

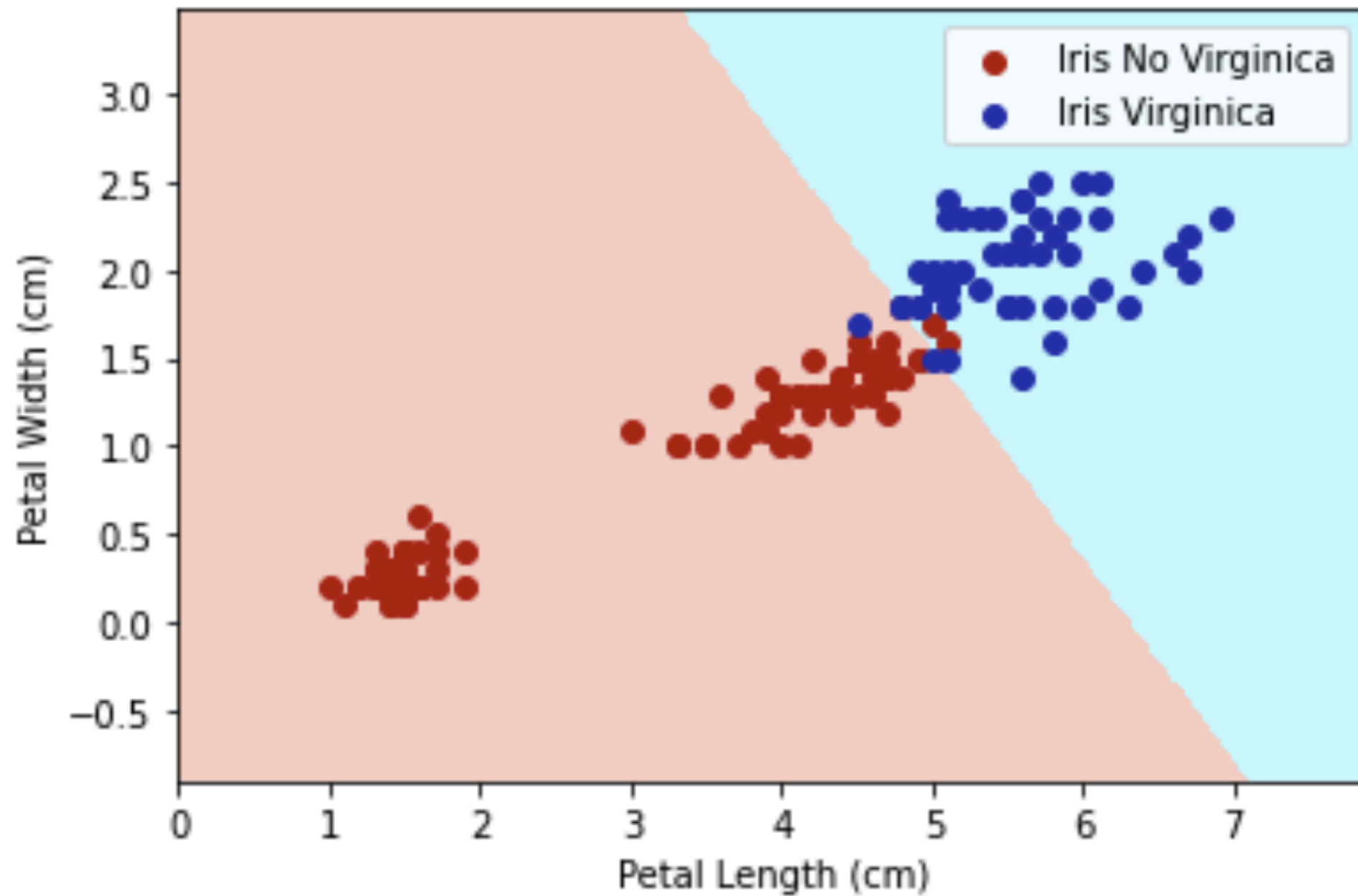
# Regresión logística

Modelo utilizado para clasificación binaria, en que una instancia pasa por una regresión lineal, y ese resultado por una sigmoide

Es un modelo de clasificación lineal, en el que aprendemos los coeficientes de la regresión (ejemplo en clases)

Para hacer esto, resolvemos un problema de optimización utilizando *Gradient Descent*

# Regresión logística



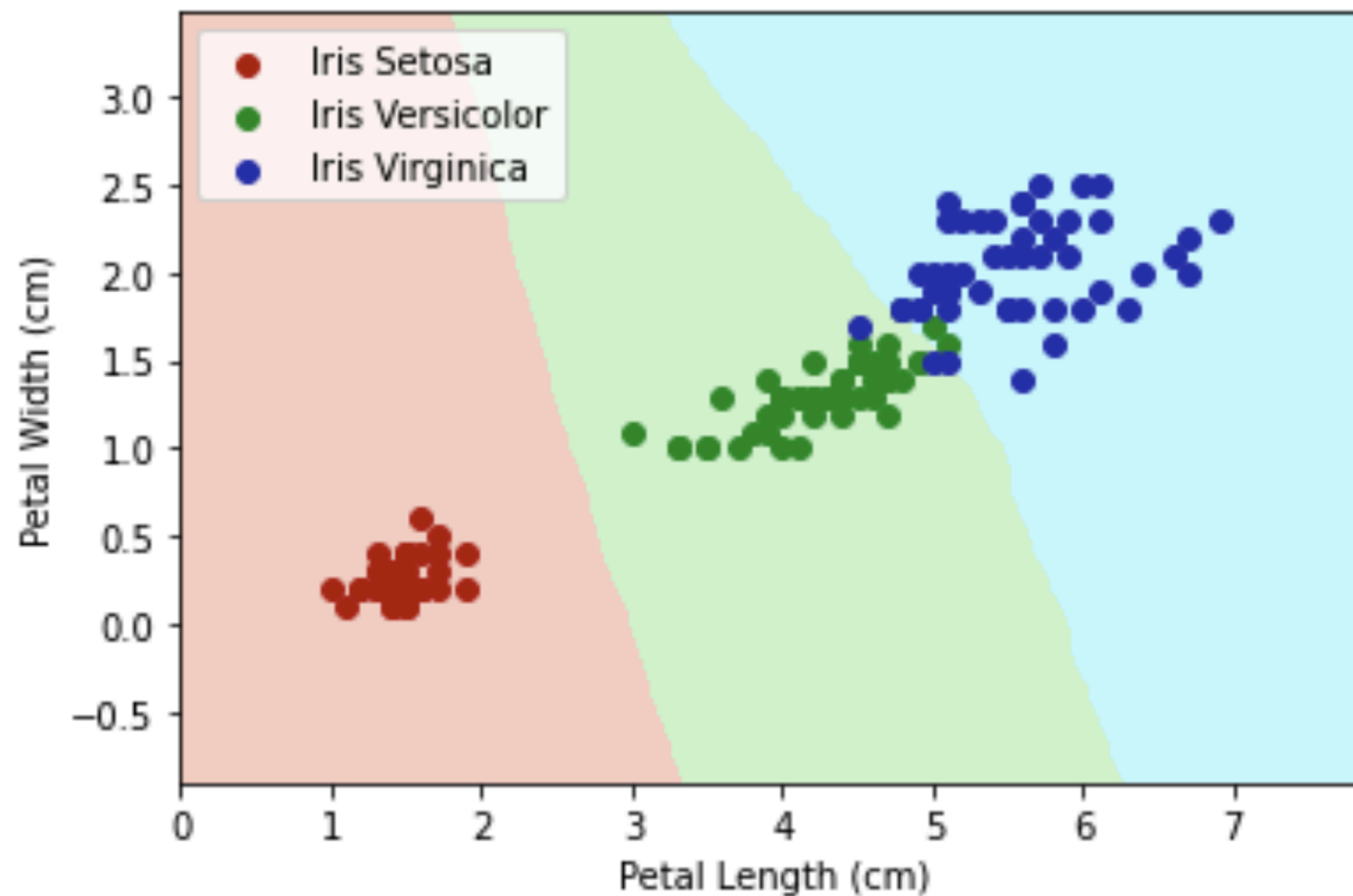
# Clasificación

En general, los distintos modelos de *Machine Learning* van a generar distintas **fronteras de decisión** (i.e. formas de separar el espacio)

Esto se logra ya que cada modelo resuelve un problema de optimización distinto

# KNN

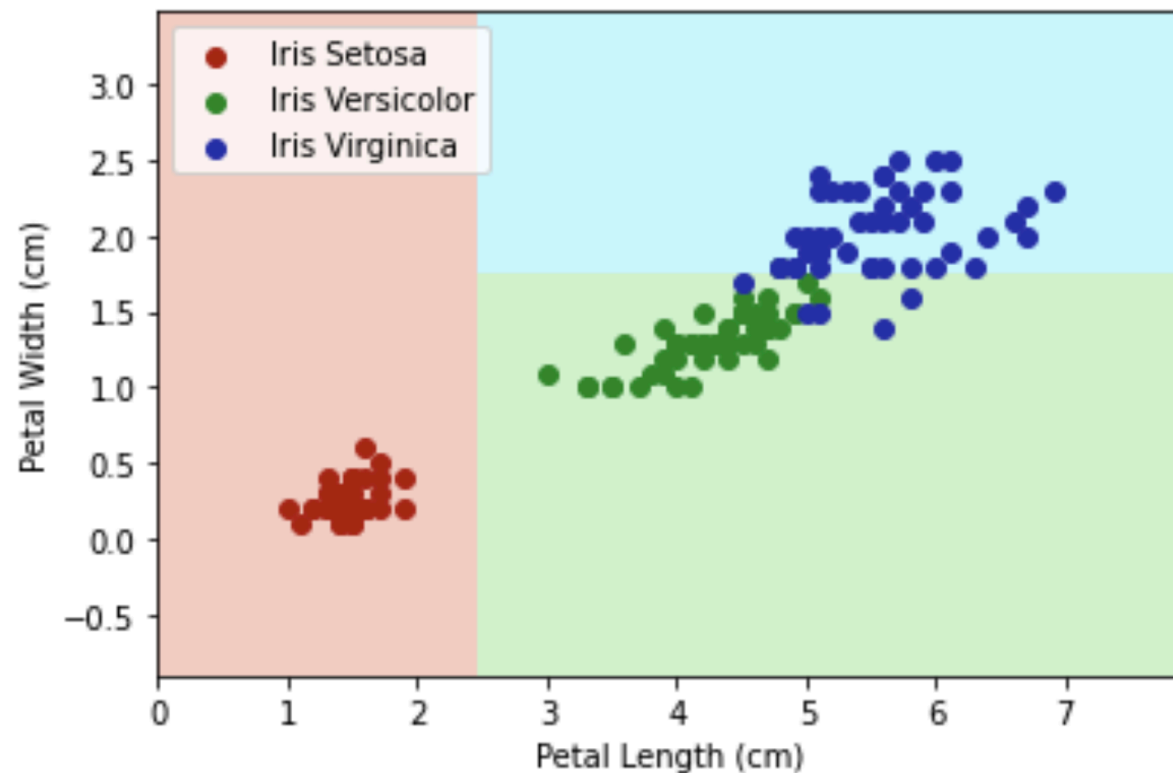
KNN es un modelo basado en distancias



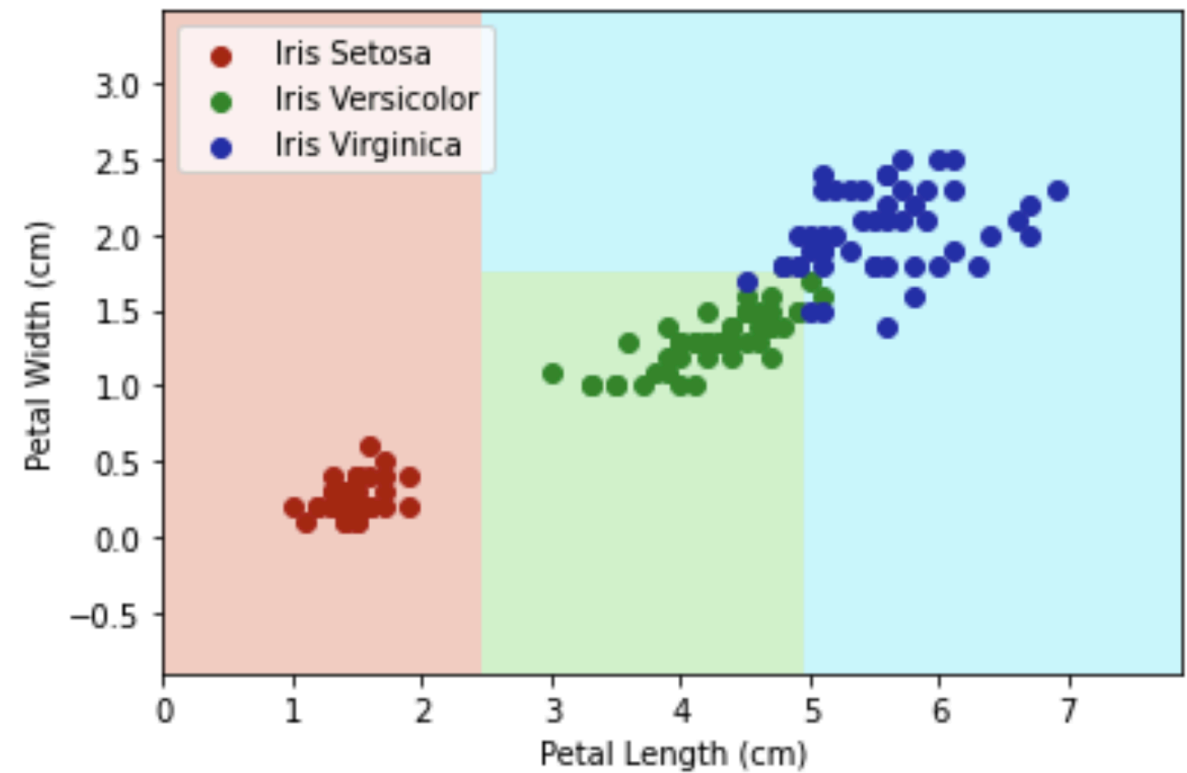
# Decision Trees

Un árbol de decisión clasifica una instancia de acuerdo a distintas reglas

Árbol de prof. 2



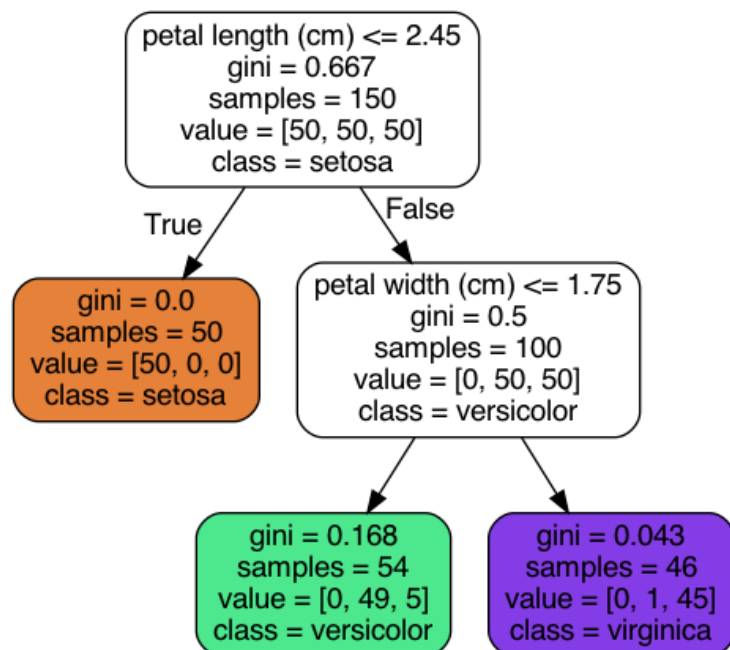
Árbol de prof. 3



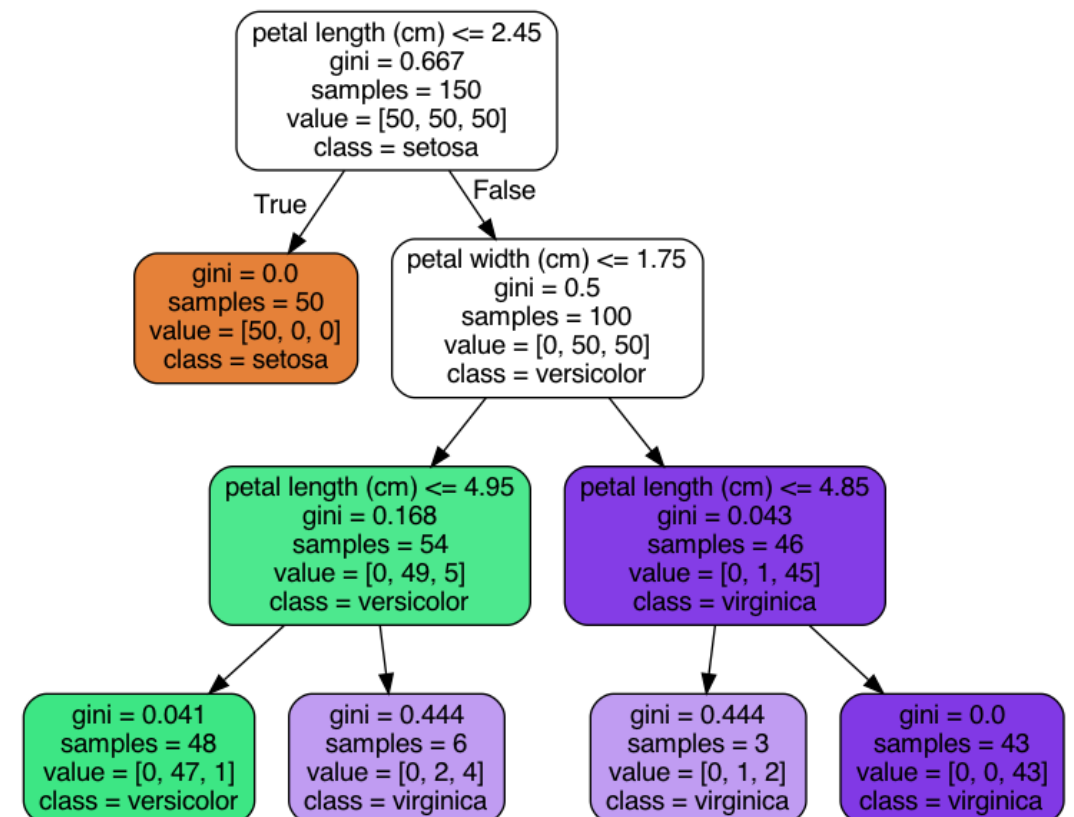
# Decision Trees

Un árbol de decisión clasifica una instancia de acuerdo a distintas reglas

Árbol de prof. 2



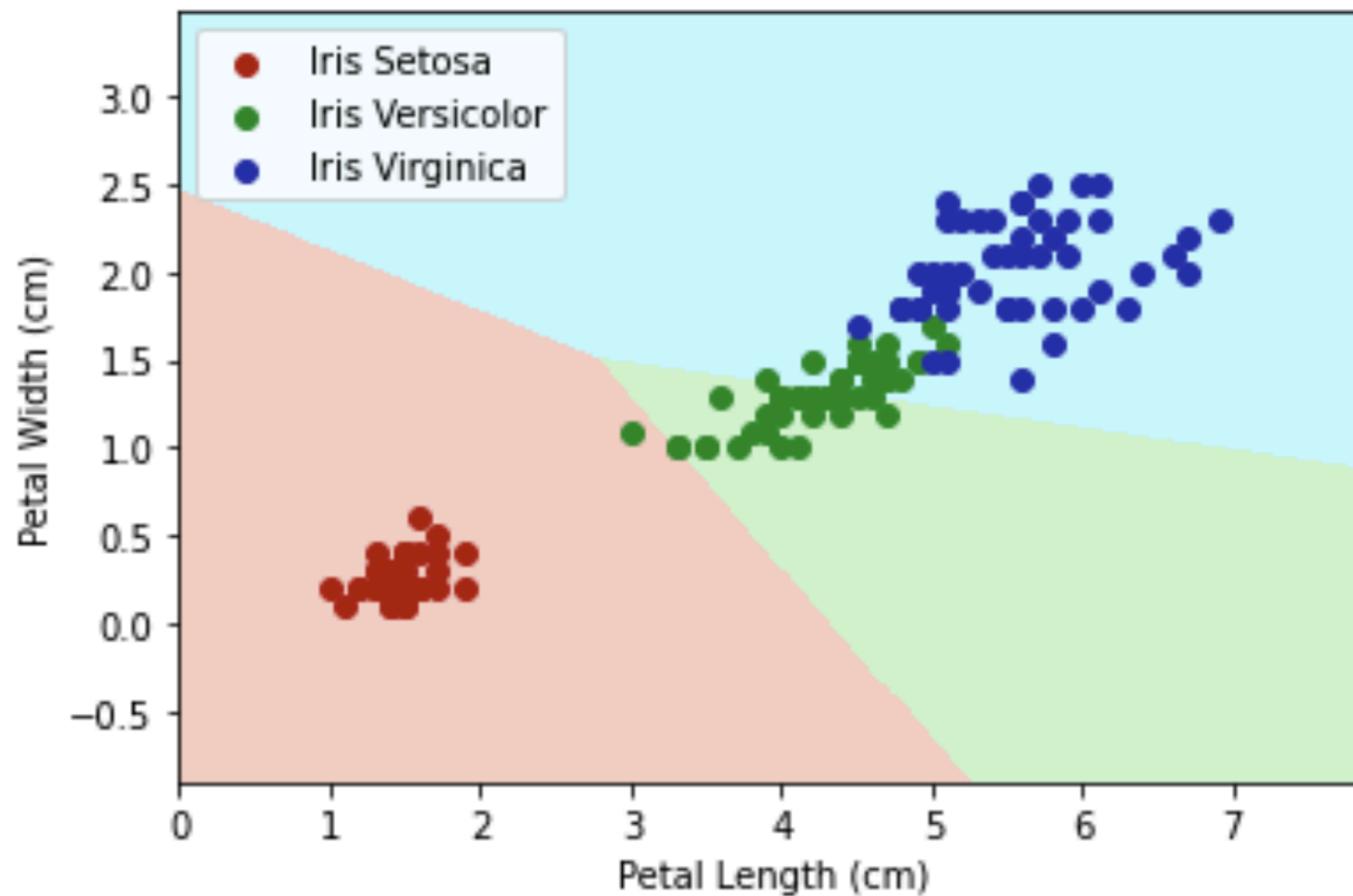
Árbol de prof. 3





# SVM

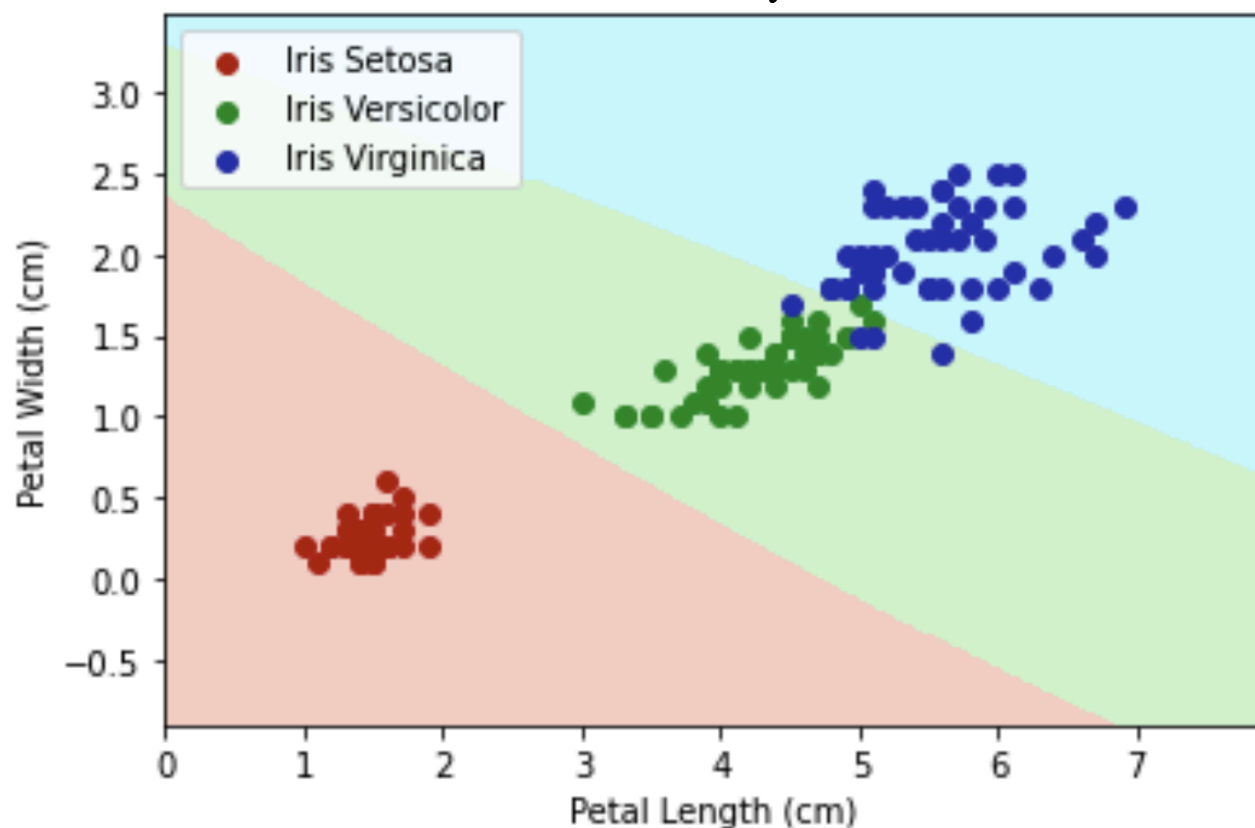
SVM busca la recta de mayor margen entre instancias



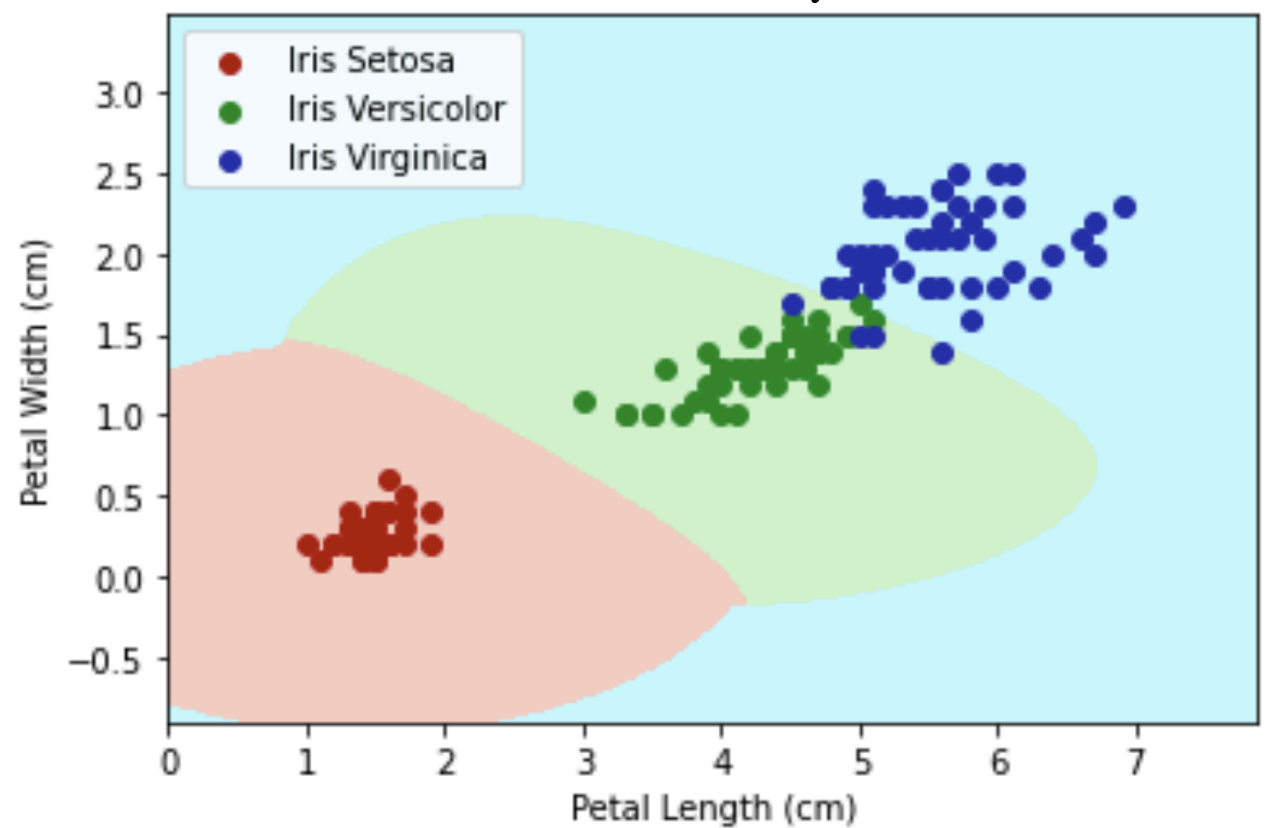
# SVM

Pero SVM también puede hacer clasificación no lineal gracias al *truco del kernel*

Kernel RBF  $\gamma = 0.1$



Kernel RBF  $\gamma = 1$



¿Y las redes neuronales?

# Aprendizaje Automático

Clase 02 - ¿Cómo aprende un programa?