FAMAF Facultad de Matemática, Astronomía, Física y Computación
UNC Universidad Nacional de Córdoba

Aprendizaje Supervisado 2024

Docentes:

- **Dra. Ing. Karim Nemer Pelliza**
- **Dr. Ing. Diego González Dondo**

Contenido de la Primera clase

- Introducción al ML
- Etapas en la aplicación del ML
- Repaso:
  - Regresión Lineal y Polinomial,
  - Regresión Logística
  - Perceptrón.
- Support Vector Machines.
- SVC/SVR.
  - Datos no linealmente separables.
  - Función de costo.
- Redes neuronales multicapa
- Naive Bayes
- Repaso:
  - Árboles de decisión
- Ensemble learning.
  - Random Forest,
  - Bagging, Boosting y Voting.
- Sistemas de recomendación.
  - Filtrado colaborativo.
- Prácticas

FAMAF Facultad de Matemática, Astronomía, Física y Computación
UNC Universidad Nacional de Córdoba

Primera Clase

Introducción al Machine Learning

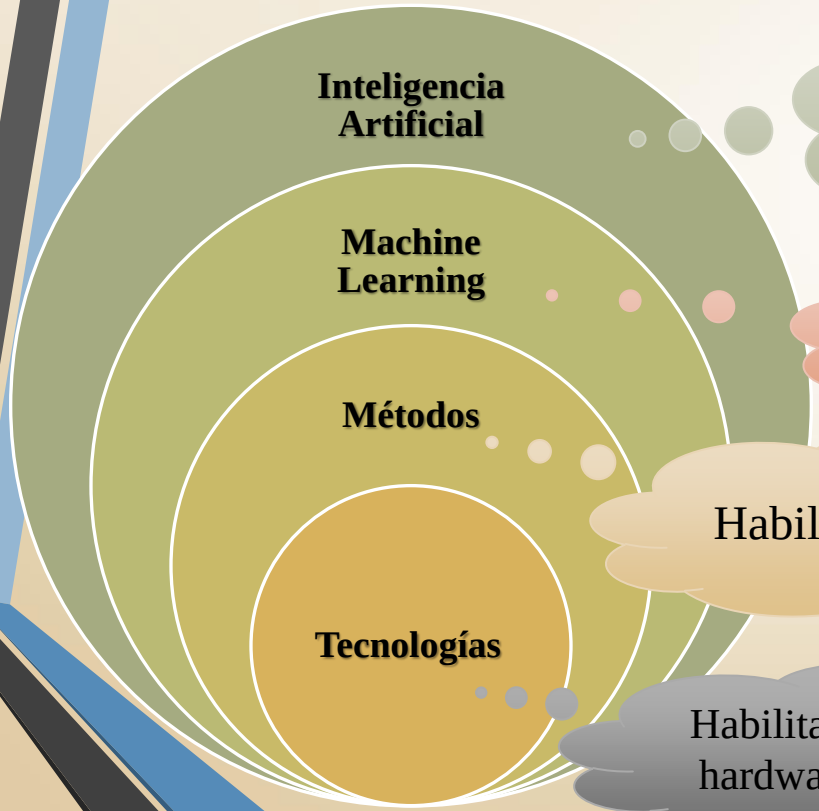
El aprendizaje automático es el proceso que le da a las computadoras la habilidad de aprender sin ser explícitamente programadas.

A.L Samuel

Se dice que un programa de computación aprende de la experiencia E con respecto a una tarea T y alguna medida de rendimiento P, si es que el rendimiento en T, medido por P, mejora con la experiencia

E.T.M Mitchell

¿Cómo se relacionan la IA y el ML?



Capacidad de censar, razonar, relacionar y aprender

- Visión por computadora
- Procesamiento del lenguaje natural
- Reconocimiento de voz
- Robótica y movimiento
- Planificación y optimización
- Simulación de conocimiento

Habilidad de aprender

- Aprendizaje supervisado,
- Aprendizaje no supervisado
- A. por refuerzo

Habilidad de razonar

- Regresión
- Árboles de decisión
- ANN
- SVM
- Ensemble Learning

Habilitación física de hardware y software

Principales usos del aprendizaje supervisado

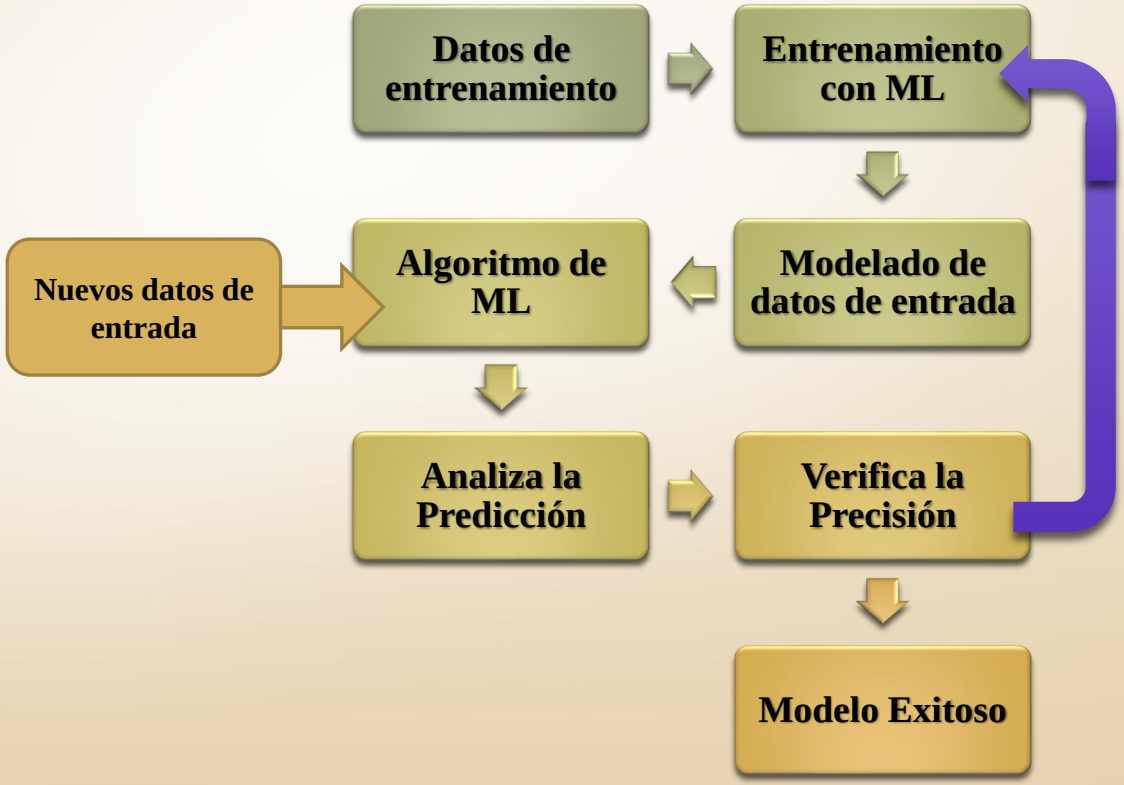
FAMAF Facultad de Matemática, Astronomía, Física y Computación
UNC Universidad Nacional de Córdoba

Etapas en la aplicación del Aprendizaje Supervisado

Descripción del problema: Aprendizaje supervisado

Datos: Se dispone de un conjunto de registros (o ejemplos, o instancias) descritos por *n* atributos: A_1, A_2, ... , A_n y cada instancia está anotada con una etiqueta, pudiendo ser una clase o un valor numérico.

Objetivo: Aprender un modelo (o función) a partir de los datos, buscando predecir sus etiquetas a partir de los atributos. Este modelo puede ser utilizado para predecir las etiquetas de nuevos registros sin anotar.



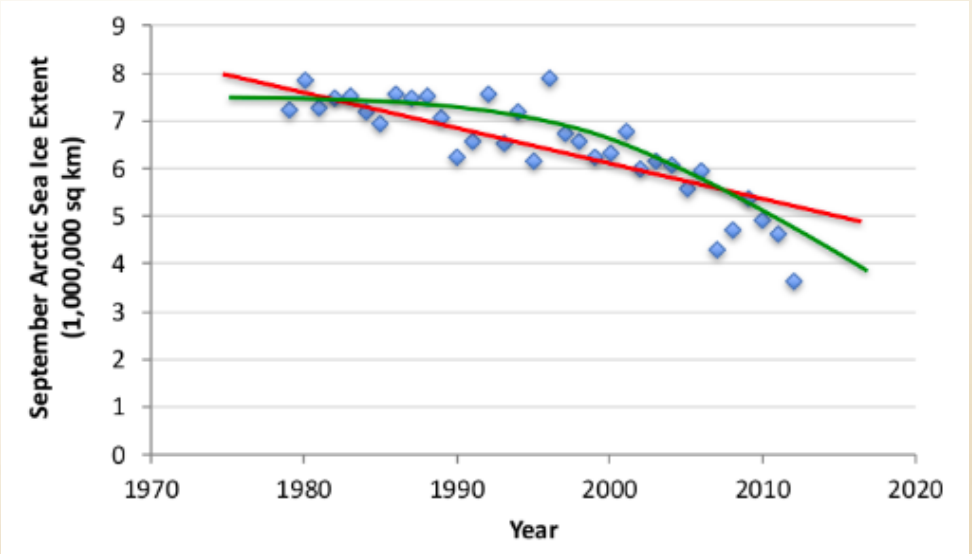
Aprendizaje Supervisado

Regresión

Dados (x_1,y_1), (x_2, y_2), …, (x_n, y_n)

Aprender una *f(x)* que permita predecir y a partir de *x*

Si $y \in \mathbb{R}^n$: Es un problema de **regresión**.

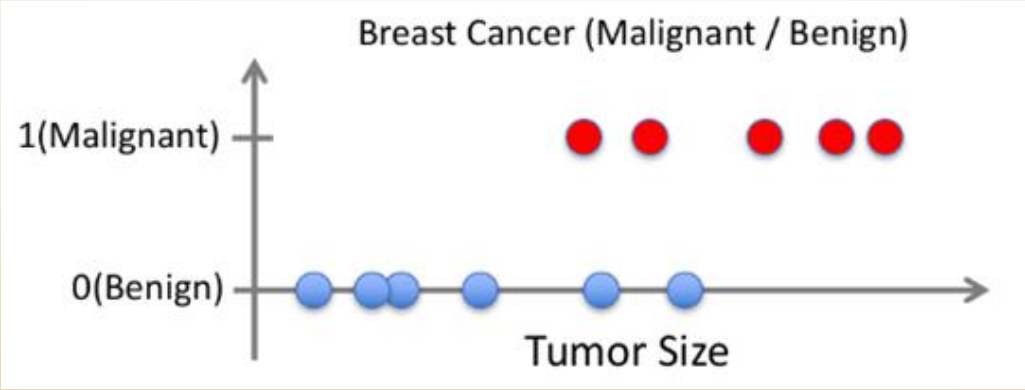


Clasificación

Dados (x_1,y_1), (x_2, y_2), …, (x_n, y_n)

Aprender una $f(x)$ que permita predecir y a partir de x

Si y es categórica: Es un problema de **clasificación**.

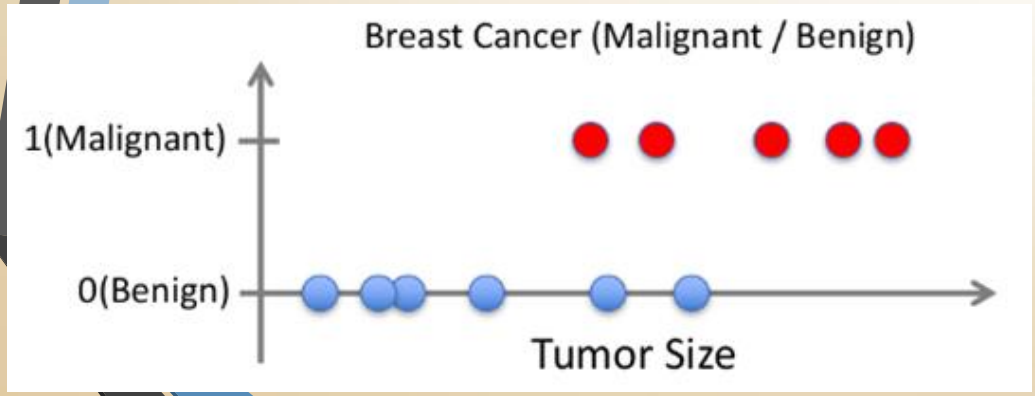


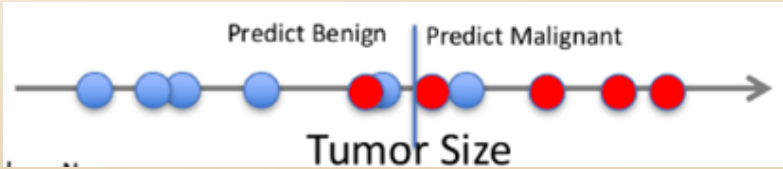
Clasificación

Dados (x_1,y_1), (x_2, y_2), …, (x_n, y_n)

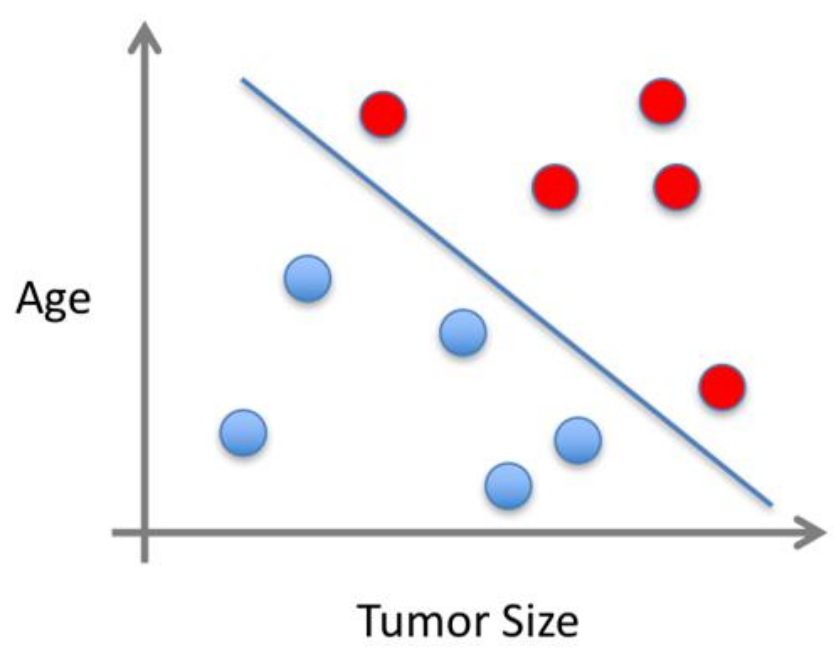
Aprender una *f(x)* que permita predecir y a partir de *x*

Si *y* es categórica: Es un problema de **clasificación**.





Aprendizaje Supervisado



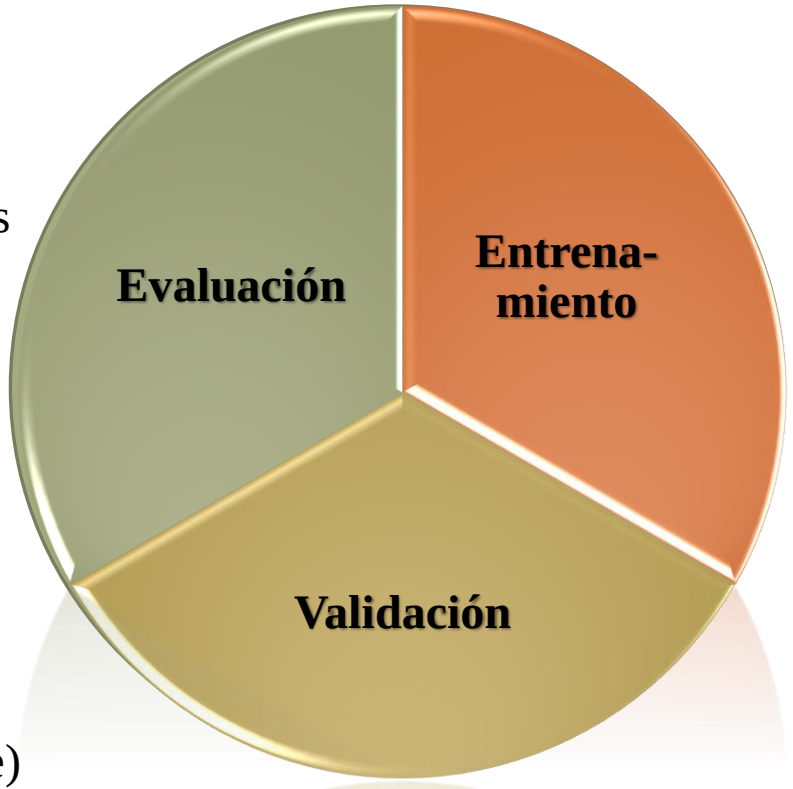
- La variable ***x*** puede ser multidimensional.
- Cada dimensión corresponde a un atributo:
  - Edad del paciente
  - Tamaño del tumor
  - Uniformidad en la forma de la célula
  - Etcétera
- La regresión busca “acercar” los datos a una función (lineal, polinomial, etc.)
- La clasificación busca separar los datos mediante ciertos “bordes”.

Elección de *hiperparámetros*

Se tiene la base de datos con todos los datos.

Dividir el conjunto total de ejemplos en tres subconjuntos

- **Entrenamiento**: aprendizaje de variables del modelo
- **Validación**: ajuste/elección de hiperparámetros
- **Evaluación**: estimación final del desempeño del modelo entrenado (y con hiperparámetros elegidos adecuadamente)



FAMAF Facultad de Matemática, Astronomía, Física y Computación
UNC Universidad Nacional de Córdoba

Regresión Lineal y Polinomial

Regresión Lineal

Busca ajustar los datos de entrenamiento mediante una función que sea un hiperplano.

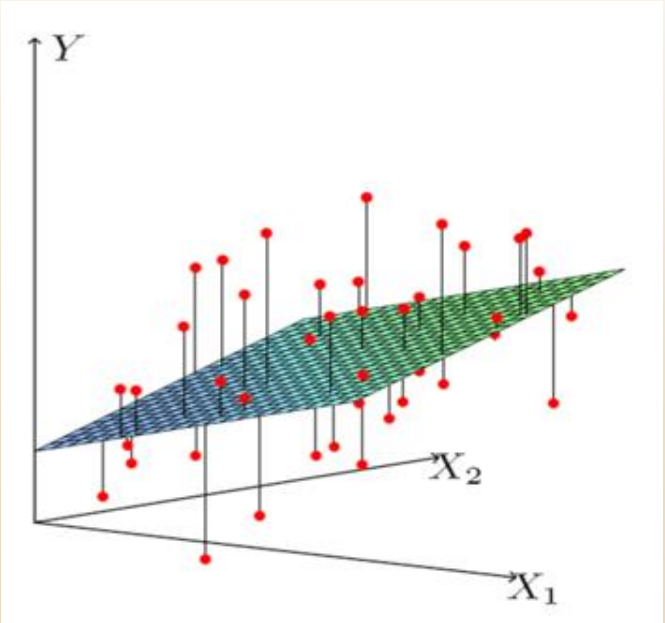
$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ \ldots + \theta_d x_d = \sum_{j=0}^{d} \theta_j x_j$$

Los valores θ son los pesos de los atributos o *features*.

Se entrena minimizando la suma del error cuadrático.

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta \left(x^i \right) - y^i \right)^2$$

Se resuelve mediante $\min_\theta J(\theta)$

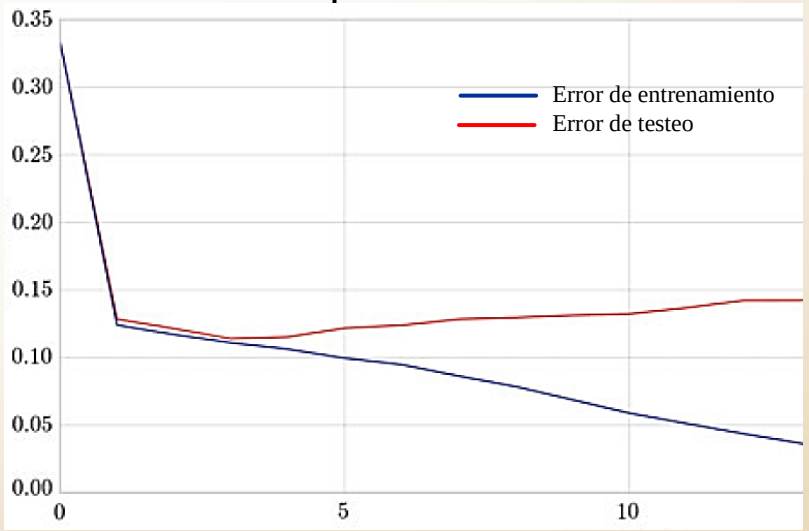


Regresión Polinomial

Busca ajustar los datos de entrenamiento mediante una función polinomial:

$$y(x, W) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$$

$$= \sum_{j=0}^{M} w_j \, x^j$$



Mientras más alto el grado del polinomio, más se ajusta a los datos (pero se vuelve más complejo y tiende a sobreajustar).

Demo Time
(demo_1_linear_regression)

FAMAF Facultad de Matemática, Astronomía, Física y Computación
UNC Universidad Nacional de Córdoba

Regresión Logística

Regresión Logística

Usa un enfoque probabilístico.

$h_\theta(x)$ debería devolver $p(y = 1|x; \theta)$

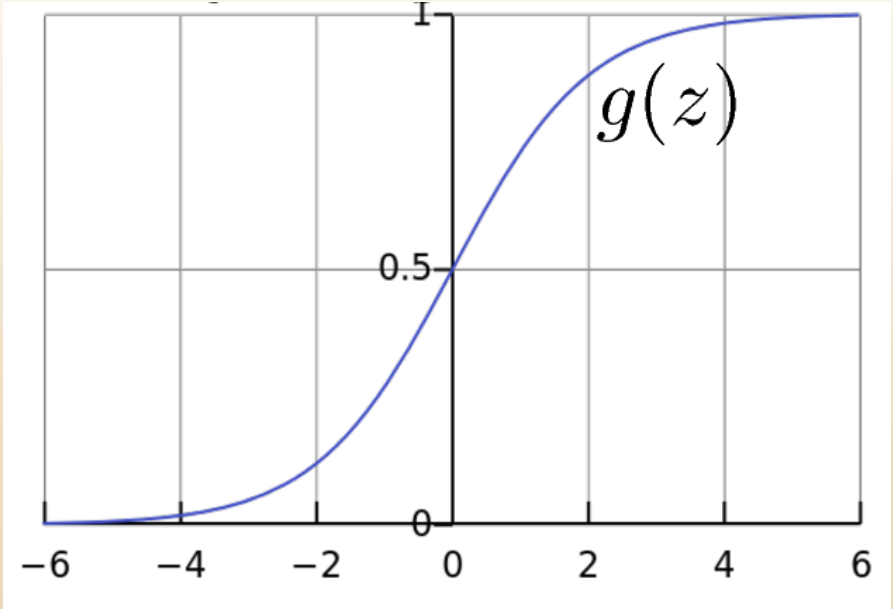
Como $0 \leq h_\theta(x) \leq 1$

Modelo de regresión logística

$$h_\theta(x) = g(\theta^T x)$$

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

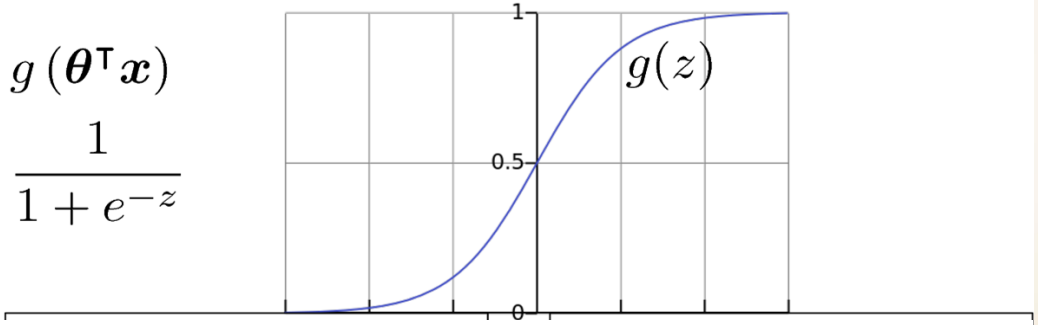
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^t x}}$$



Regresión Logística

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}\right)$$

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

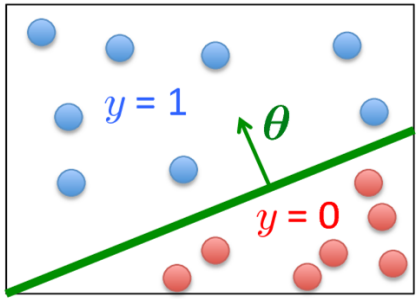


$\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}$ debería tener *valores* ***negativos*** *grandes para instancias negativas* y *valores* ***positivos*** *grandes para instancias positivas*.

Definir un umbral y...

Predecir : $y = 1$ si $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5$

Predecir : $y = 0$ si $h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5$



Based on slide by Andrew Ng

Regresión Logística: Función de costo

$$J(\theta) = -\sum_{i=1}^{n} \left[y^i \log h_\theta(x^i) + (1 - y^i) \log\left(1 - h_\theta(x^i)\right) \right]$$

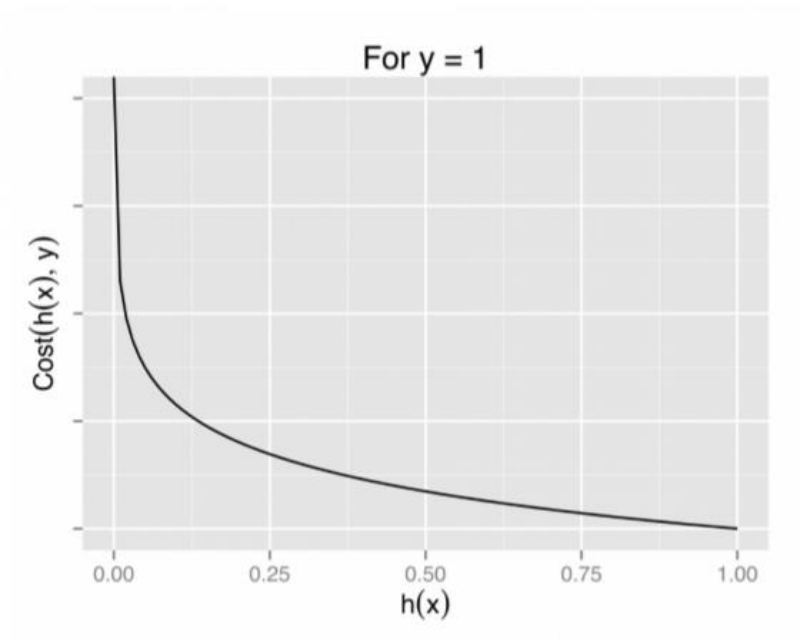
El costo de una sola instancia de los datos se define como:

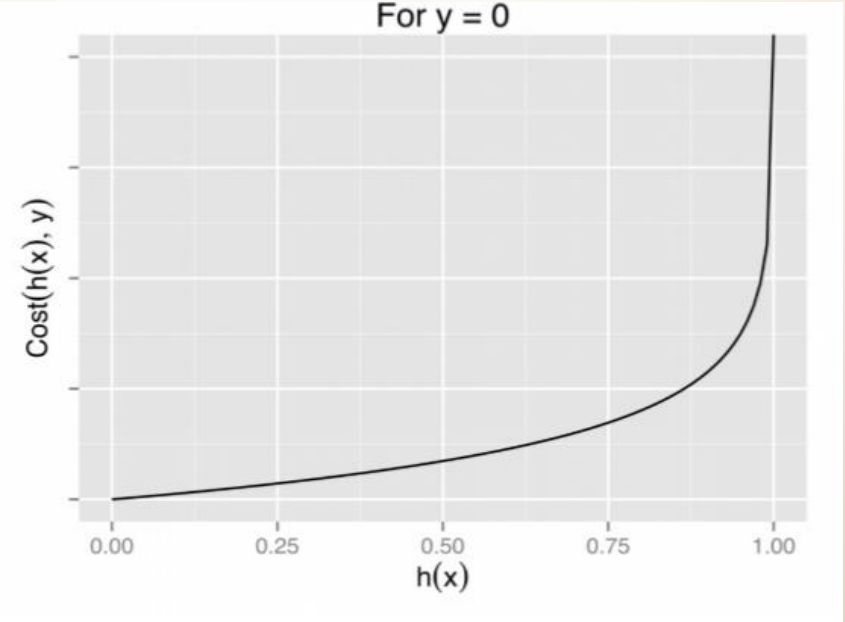
$$cost(h_\theta(x), \mathrm{y}) = \begin{cases} -\log(h_\theta(x)) & si\ y = 1 \\ -\log(1 - h_\theta(x)) & si\ y = 0 \end{cases}$$

Reescribimos la función de costo como:

$$J(\theta) = -\sum_{i=1}^{n} [cost(h_\theta(x^i), \mathrm{y}^i)]$$

Comportamiento de la función de costo





Demo Time
(demo_2_logistic_regression)

Algoritmo del perceptrón

El algoritmo del "perceptrón"

- Propuesto por Frank Rosenblatt en 1958
- El objetivo es encontrar un hiperplano de separación, esto es, sólo encuentra la solución si los datos son linealmente separables
- Es un algoritmo online (procesa un ejemplo a la vez)
- Es la red neuronal más simple
- Tiene una capa de entrada y un único nodo de salida.

El algoritmo del "perceptrón"

Entrada

- Conjunto de entrenamiento
- Tasa de aprendizaje

Algoritmo

- Inicializar los pesos
- $\omega^0 \in \mathbb{R}$

Proceso

- Para cada ejemplo (x_i, y_i) predecir $y_i^t = signo(\omega^t x_i + \omega_0)$

Comparo salidas real y obtenida

- Si $y_i^t \neq y_i \therefore \omega^{t+1} \leftarrow \omega^t + r(y_i x_i)$

Sólo se actualiza cuando se comete un error

Si se han actualizado algunos pesos se vuelve a "Proceso", si no, se finaliza

Link

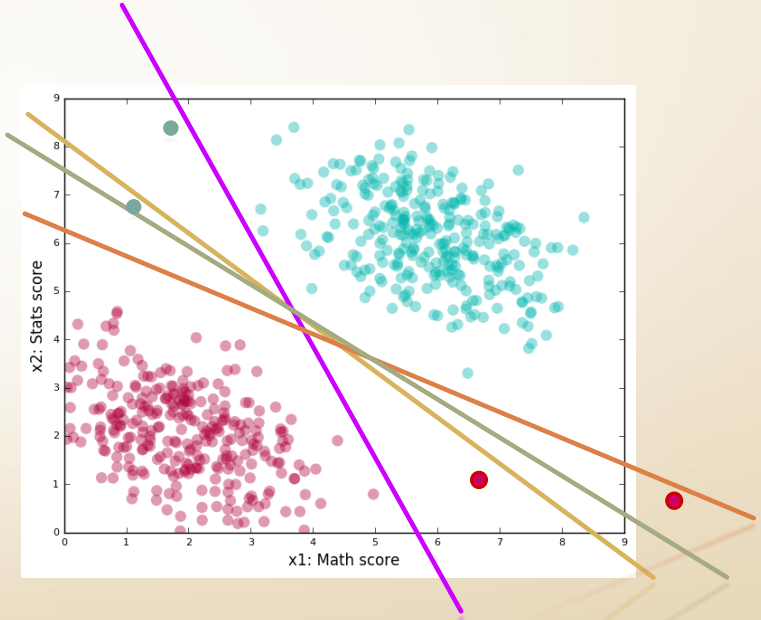
Demo Time
(demo_3_perceptron)

FAMAF Facultad de Matemática, Astronomía, Física y Computación
UNC Universidad Nacional de Córdoba

Support Vector Machines

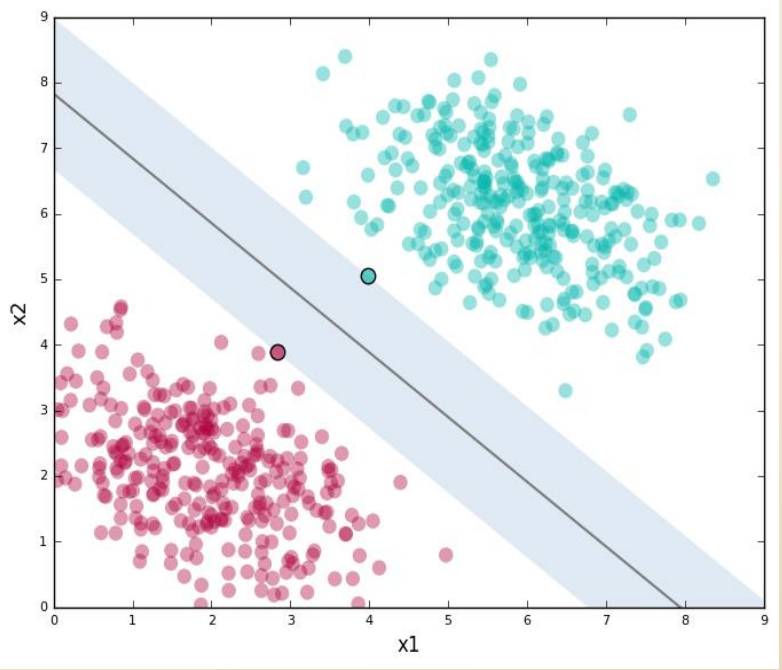
SVM: Fronteras de decisión en clasificación

- 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?
- Cualquiera de las líneas separa los datos correctamente.
- Buscamos una línea que capture el patrón general entre los datos.
- La línea fucsia tiene menos margen entre ella y ambos clústeres de datos.
- La línea azul se encuentra bien a la mitad de ambos clústeres.



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).



Link

SVM: Función de costo y a optimizar

- 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:

$$\boldsymbol{c\big(x, y, f(x)\big) = max(0, 1 - y * f(x))}$$

- 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).

- La función que buscamos minimizar es la siguiente:

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

- Dónde $\lambda \, \|\omega\|^2$ es el parámetro de regularización.

SVM: Gradientes

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

$$\frac{\delta}{\delta \omega_k} = \lambda \|\omega\|^2 = 2\lambda\omega_k$$

$$\frac{\delta}{\delta \omega_k} \max(0; 1 - y_i \langle x_i, \omega \rangle = \begin{cases} 0 & si\ y_i \langle x_i, \omega \rangle \geq 1 \\ -y_i\, x_{ik} & c.c. \end{cases}$$

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

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

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

$$\omega = \omega - \alpha(y_i x_i - 2\lambda\omega)$$

SVM con outliers

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).

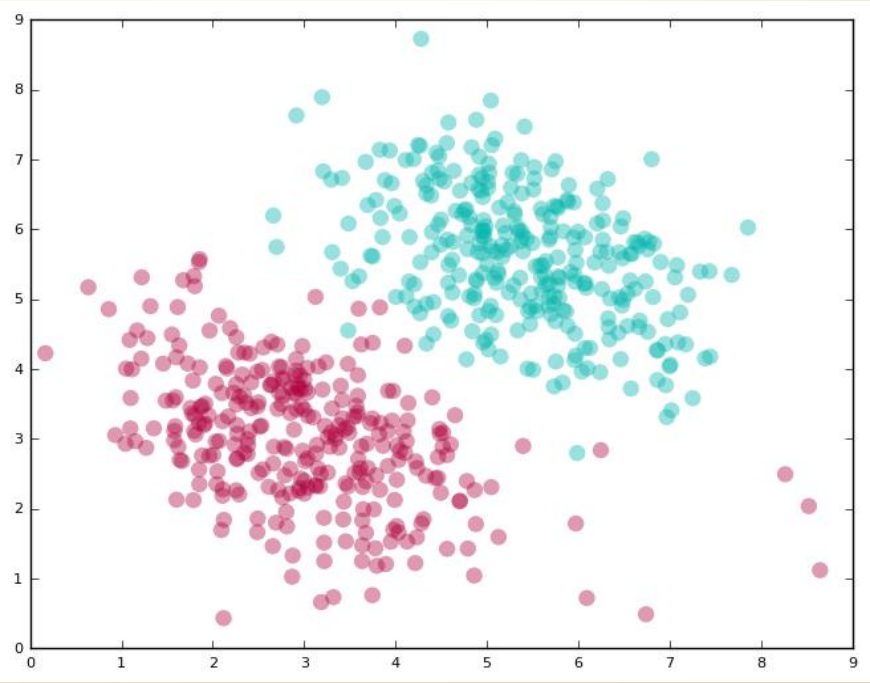


Imagen de https://blog.statsbot.co/

SVM: Parámetro C

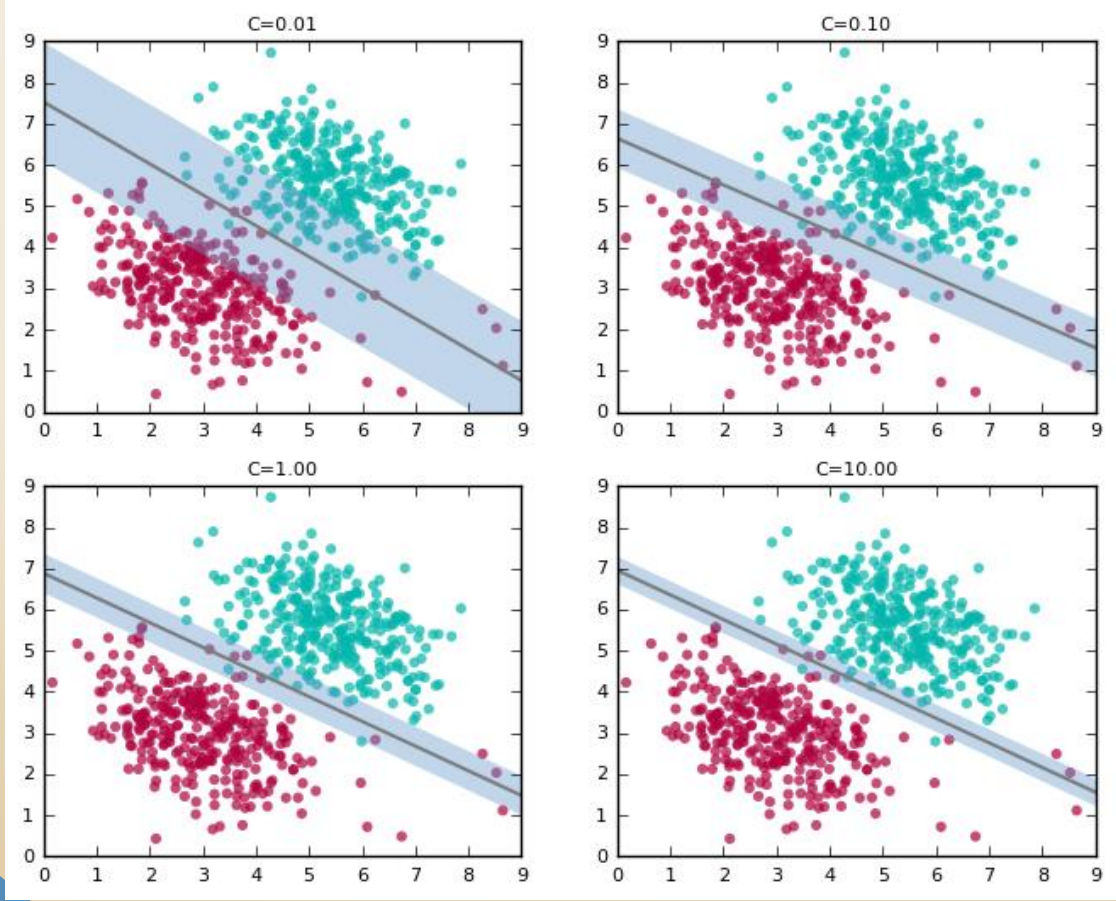


Imagen de https://blog.statsbot.co/

Demo Time (demo_4_svm)

Preguntas y consultas