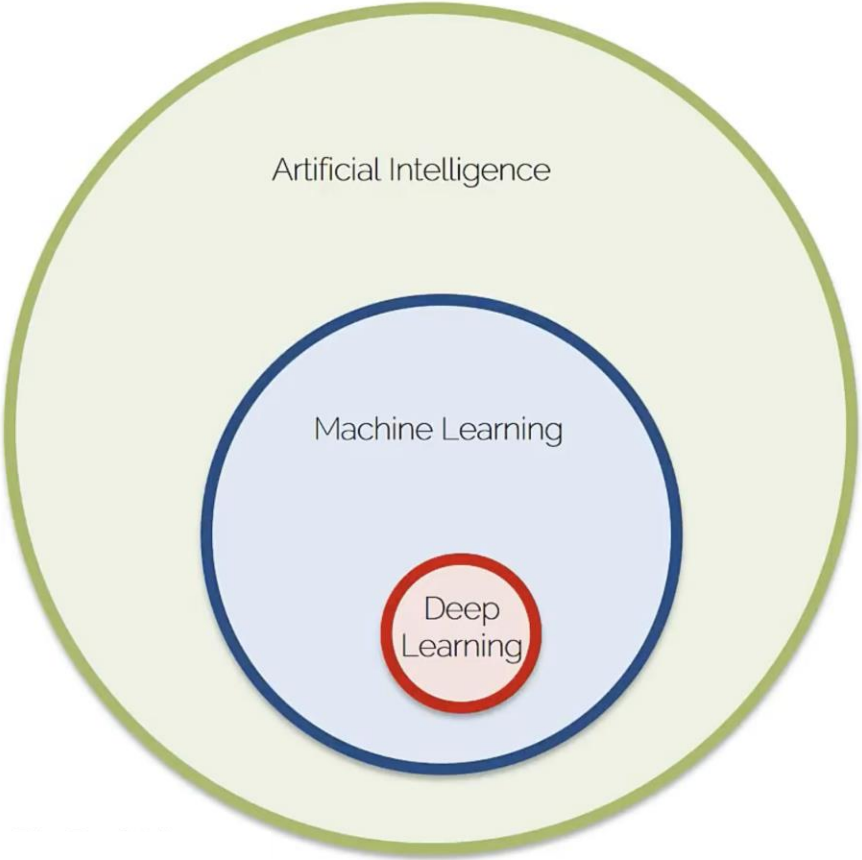
Fundamentos del ML

AI vs ML vs DL



Una tarea sencilla: Image Classification

Image Classification

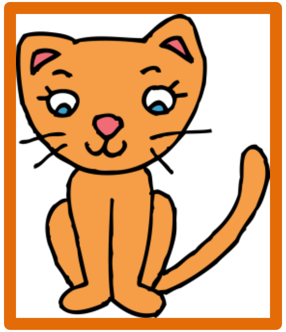


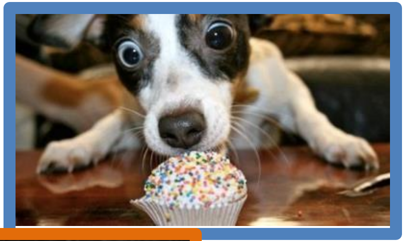







Image Classification





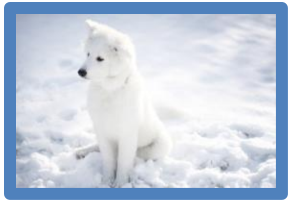
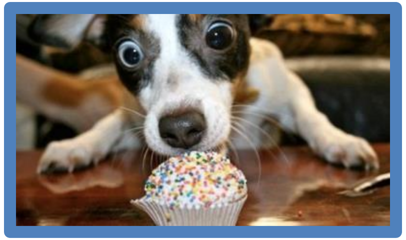


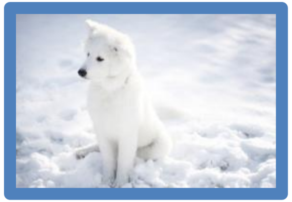
Image Classification

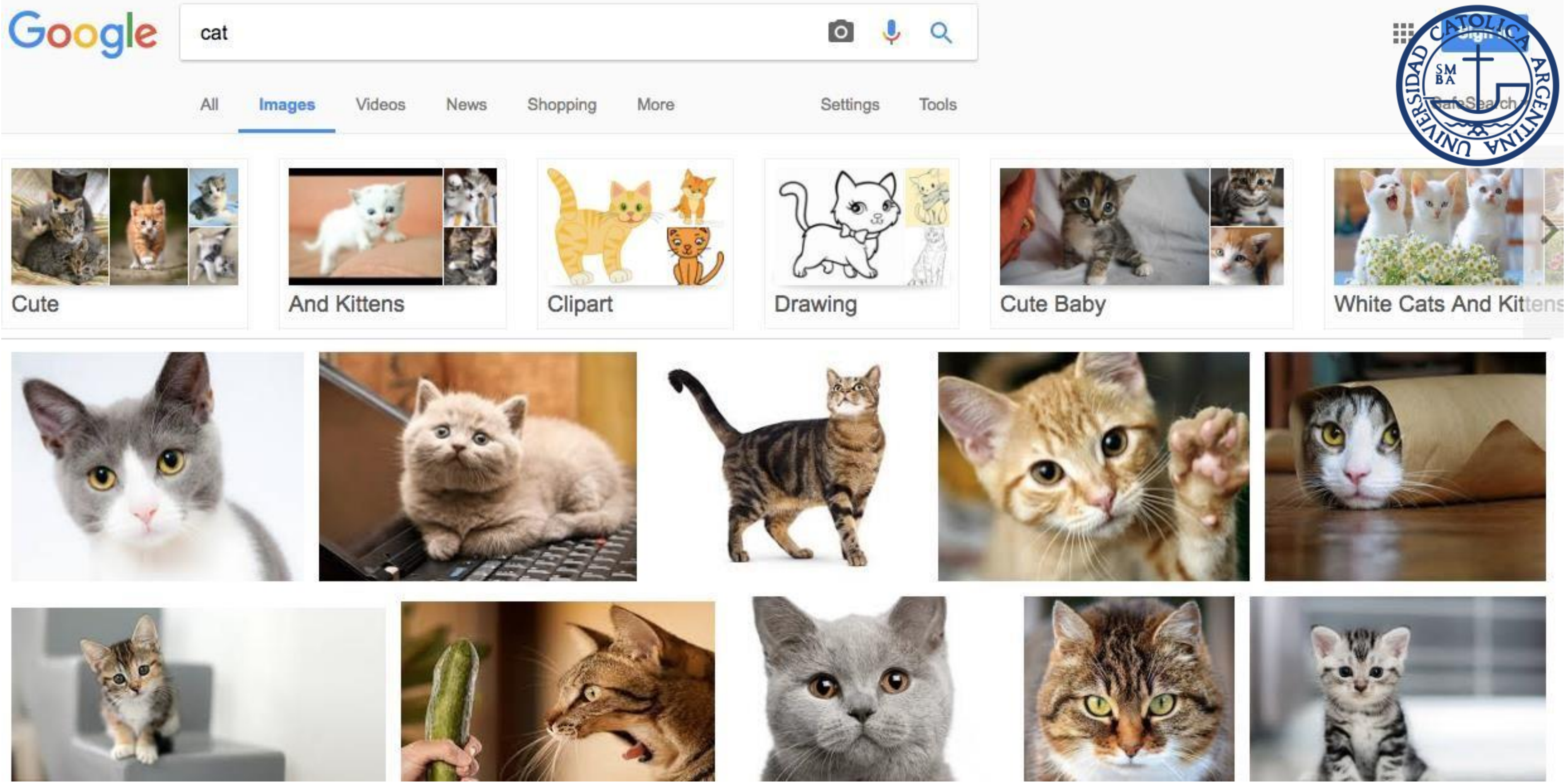


Obstrucciones

Image Classification

Desorden de fondo

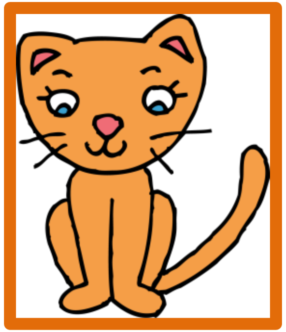


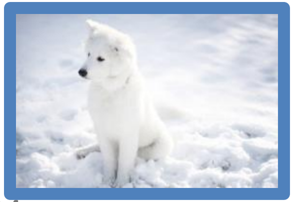


Pose **Iluminación** **Aspecto**

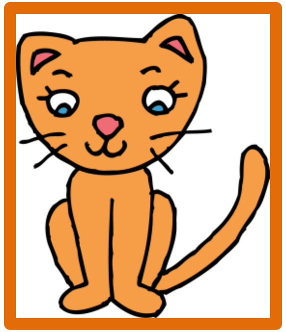
Un Clasificador Simple

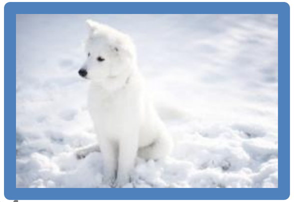
Nearest Neighbor





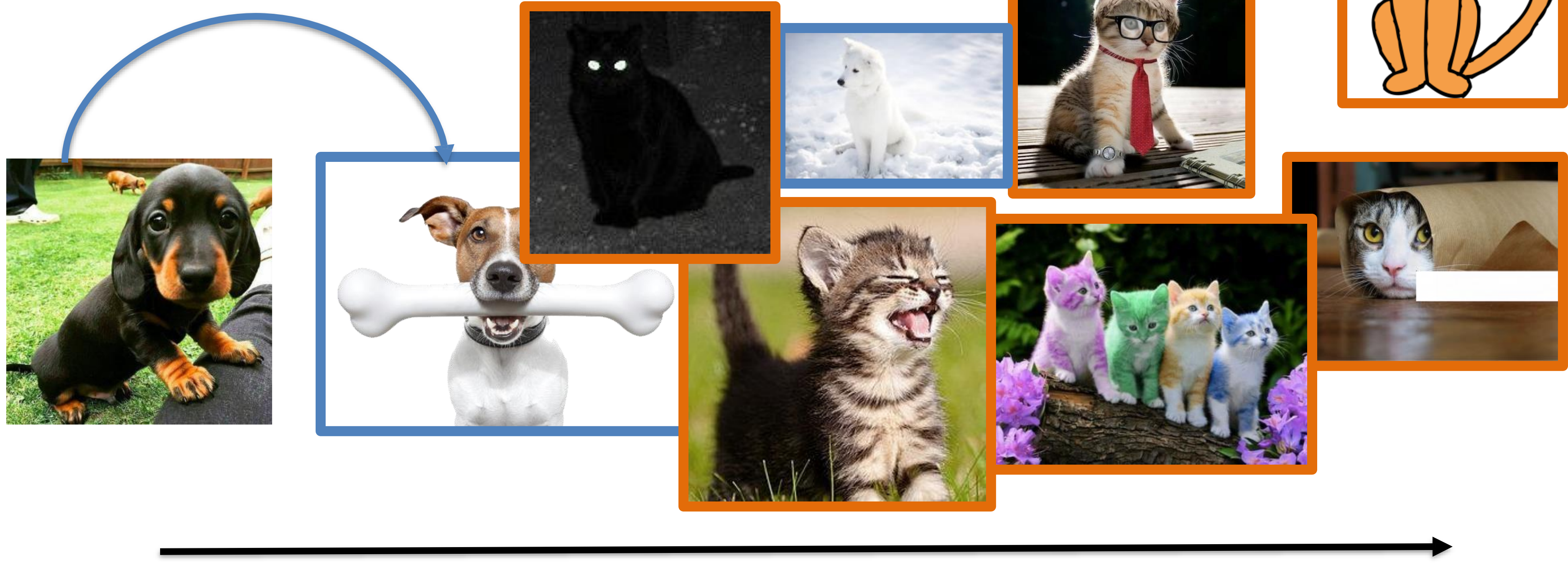
Nearest Neighbor





Nearest Neighbor

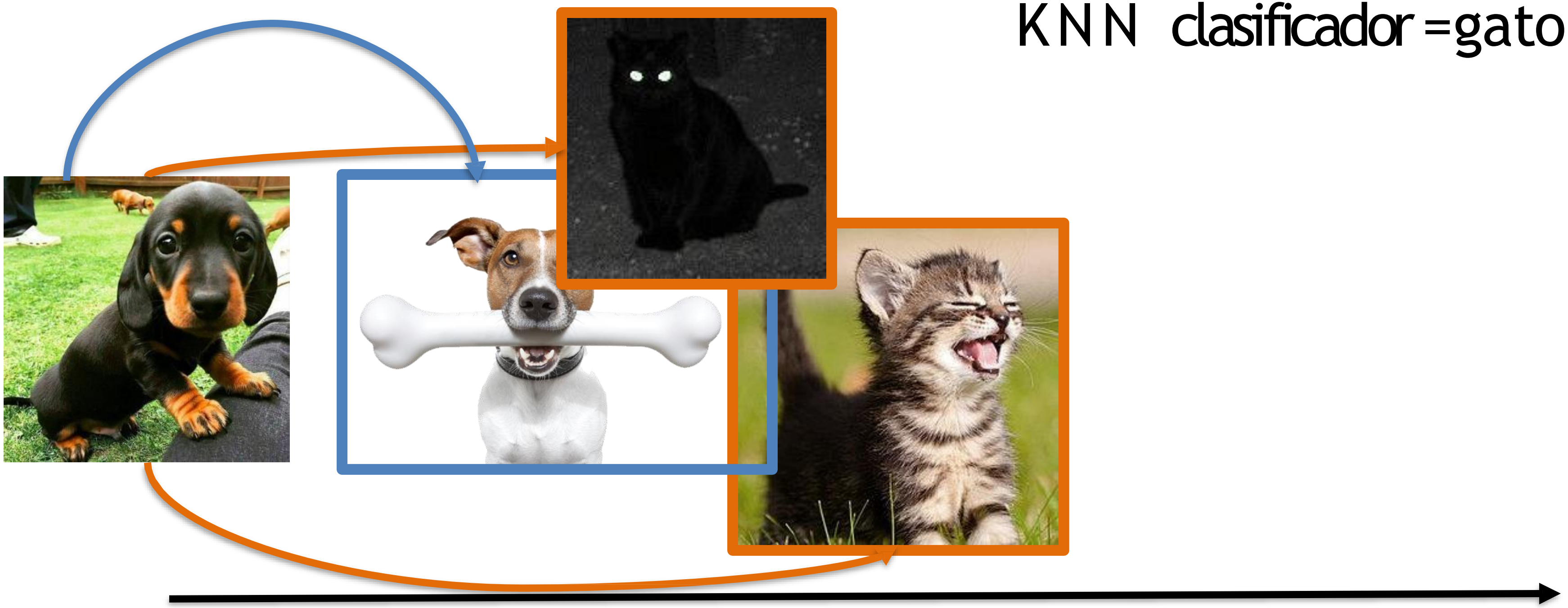
NN clasificador=perro



distancia

Nearest Neighbor

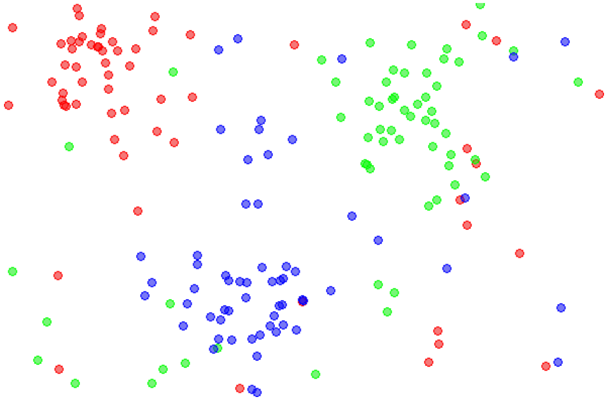
KNN clasificador = gato



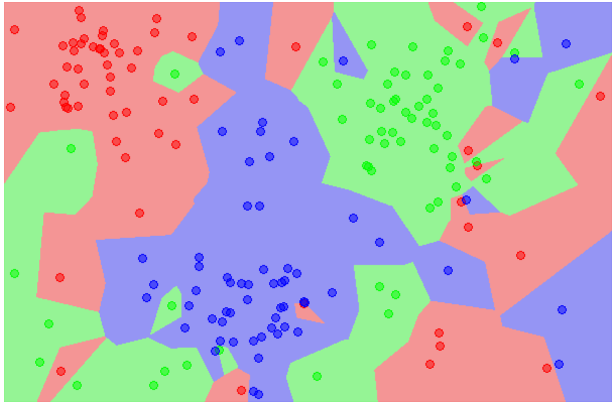
distancia

Nearest Neighbor

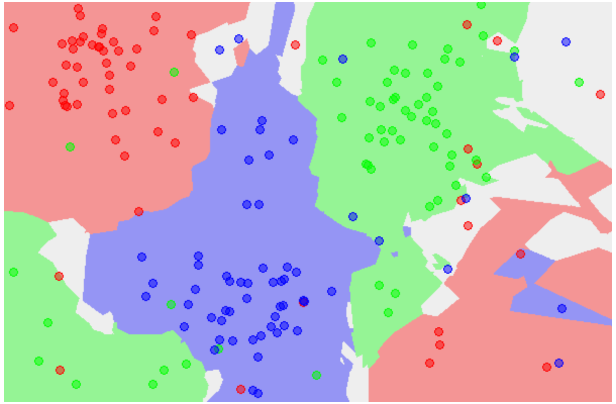
Los datos



Clasificador NN



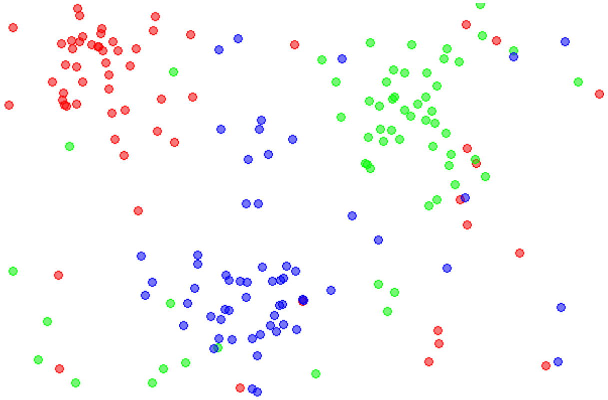
Clasificador 5NN



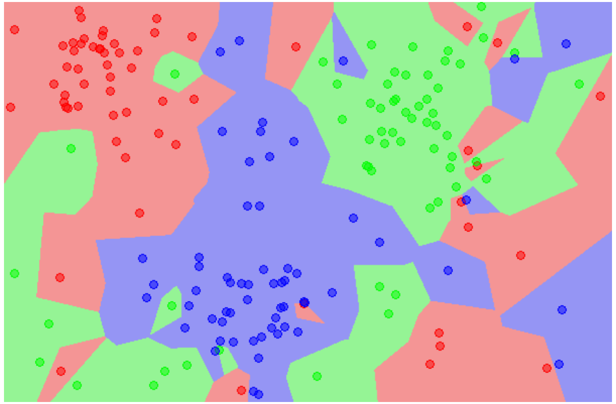
Fuente: [https://commons.wikimedia.org/wiki/File:Data3cla s ses.png](https://commons.wikimedia.org/wiki/File:Data3classes.png)

Nearest Neighbor

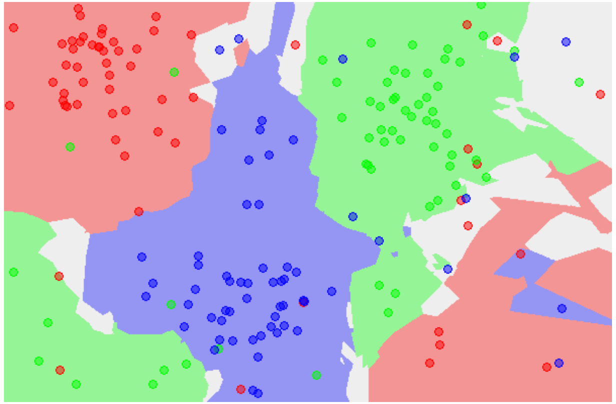
Los datos



Clasificador NN



Clasificador 5NN



¿Cómo funciona el clasificador NN en los datos de entrenamiento?

¿Qué clasificador tiene más probabilidades de obtener mejores resultados en la test data?

Nearest Neighbor

- Hiperparámetros
  - Distancia L1 : $|x - c|$
  - Distancia L2 : $||x - c|||_2$
  - Número de vecinos: k

- Los hiperparámetros dependen del problema.

- Cómo elegimos estos hiperparámetros?

Machine Learning for Classification

Machine Learning

- $M_{\theta}(I) = \{\text{DOG, CAT}\}$

Modelo

Parámetros del modelo

Imagen

Clase
Etiqueta

CAT

DOG

DOG

CAT

CAT

DOG

Machine Learning

- $M_\theta(I) = \{\mathrm{DOG}, \mathrm{CAT}\}$

Modelo

Parámetros del modelo

Imagen

Clase
Etiqueta

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_i \mathrm{D}(M_\theta(I_i) - Y_i)$$

"Distance" function

$\{\mathrm{DOG}, \mathrm{CAT}\}$

CAT

DOG

DOG

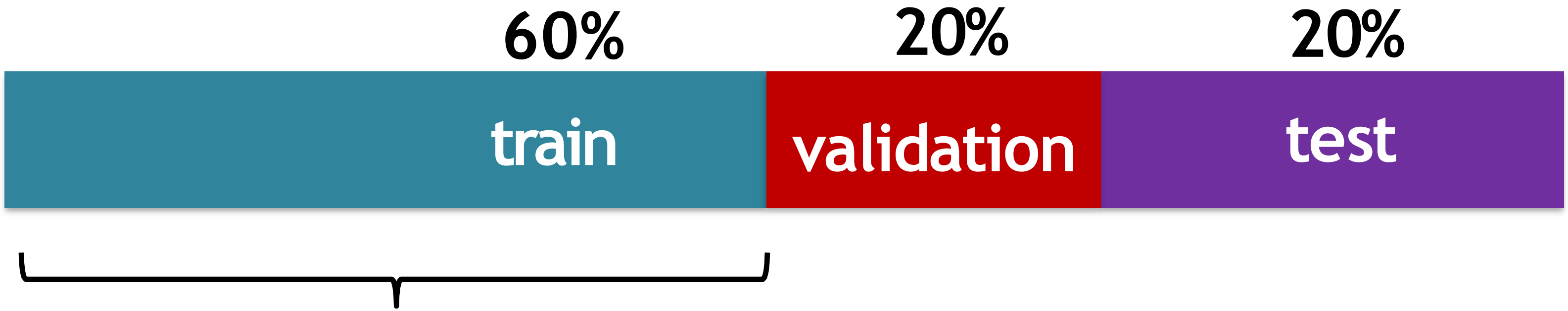
CAT

CAT

DOG

Receta básica para el ML

- Divida los datos

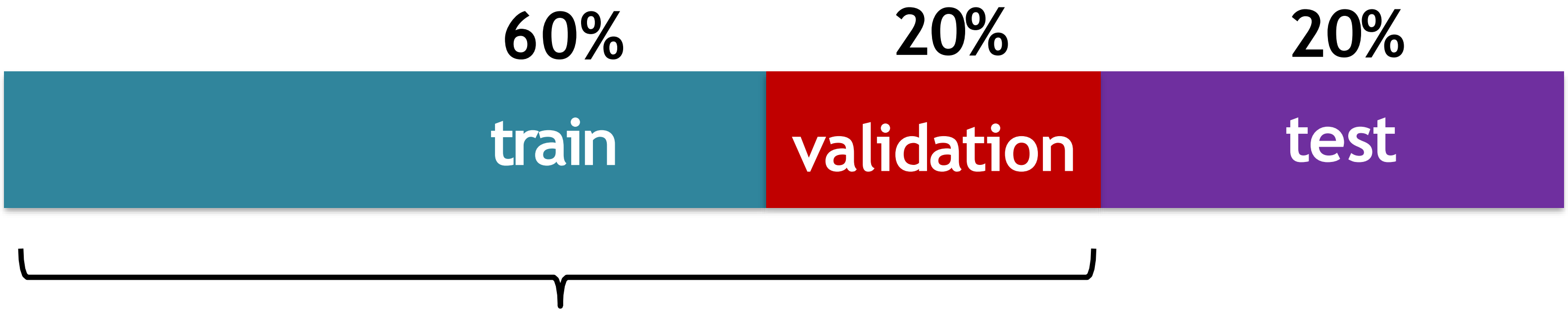


Encuentre los parámetros del modelo θ

También son posibles otras divisiones (por ejemplo, 80%/10%/10%)

Receta básica para el ML

- Divida los datos



Encuentre sus hiperparámetros

También son posibles otras divisiones (por ejemplo, 80%/10%/10%)

Receta básica para el ML

- Divida los datos

60%20% 20%

tros

Test set solo se utiliza una vez

Machine Learning

Aprendizaje no supervisado

Aprendizaje supervisado

- Labels o clases objetivo

Machine Learning

Aprendizaje no supervisado

Aprendizaje supervisado

DOG

DOG

CAT

CAT

DOG

Machine Learning

Aprendizaje no supervisado

- Sin labels ni clase objetivo
- Encontrar propiedades de la estructura de los datos
- Agrupación (k-means, PCA, etc.)

Aprendizaje supervisado

DOG

DOG

CAT

CAT

PER
RO

Machine Learning

Aprendizaje no supervisado

Aprendizaje supervisado

DOG

PER
RO

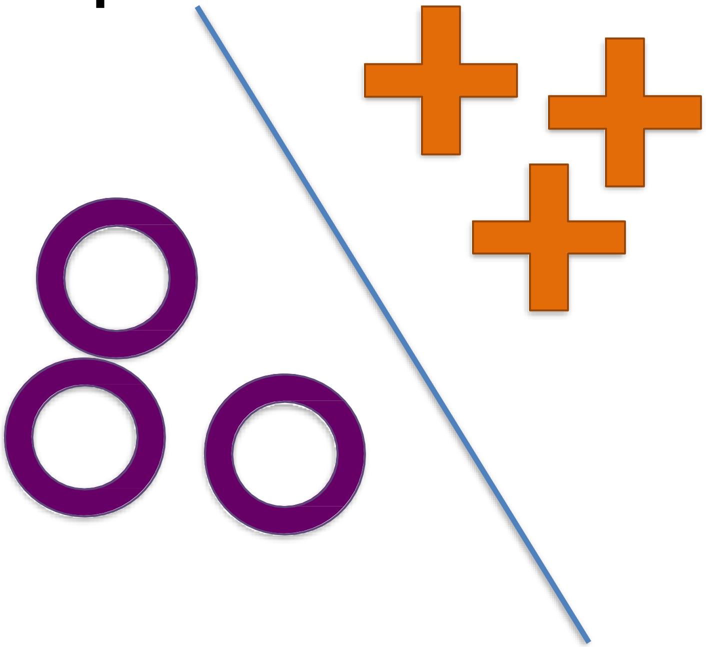
CAT

CAT

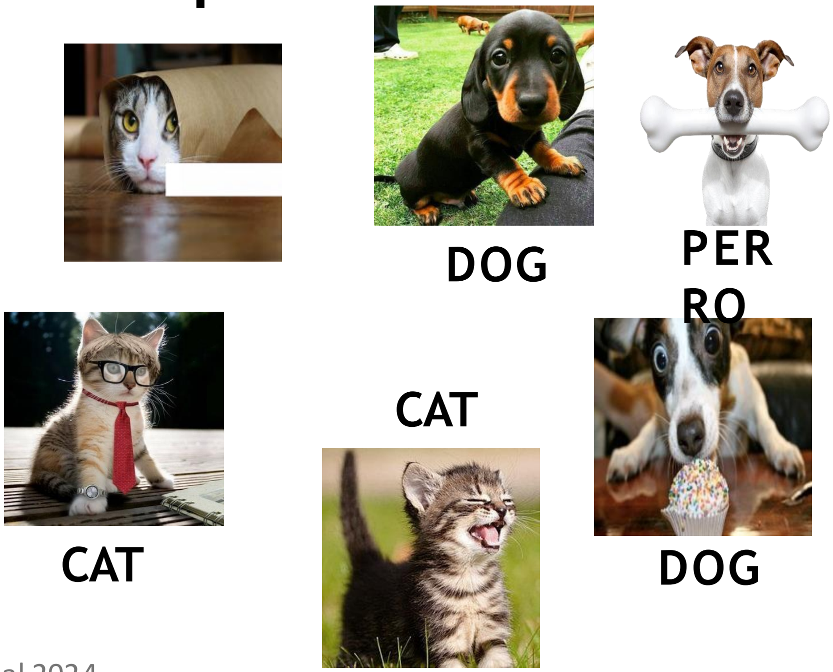
DOG

Machine Learning

Aprendizaje no supervisado



Aprendizaje supervisado



Machine Learning

Aprendizaje no supervisado

Aprendizaje supervisado

Machine Learning

Aprendizaje no supervisado

Aprendizaje supervisado

Reinforcement Learning

Agentes → interacción → Entorno

Machine Learning

Aprendizaje no supervisado

Aprendizaje supervisado

Reinforcement Learning

Agentes ← recompensa ← Entorno

Machine Learning

Aprendizaje no supervisado

Aprendizaje supervisado

Aprendizaje por refuerzo

Agentes

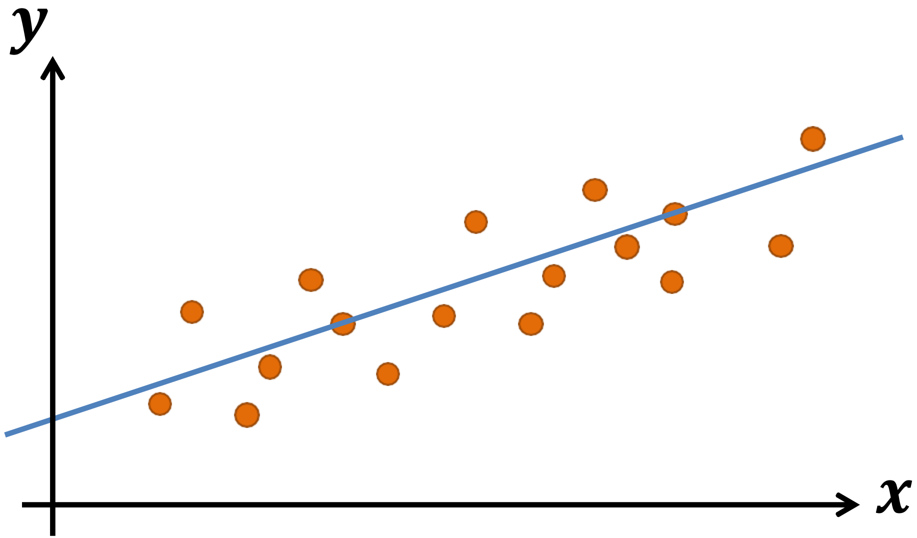
recompensa

Entorno

Regresión lineal

Regresión lineal

- Aprendizaje supervisado
- Encontrar un modelo lineal que explique un output y dados los input x



Predicción lineal

- **Un modelo lineal se expresa de la siguiente forma:**

input dimension

$$\hat{y}_i = \sum_{j=1}^{d} x_{ij}\theta_j$$

weights (i.e., model parameters)

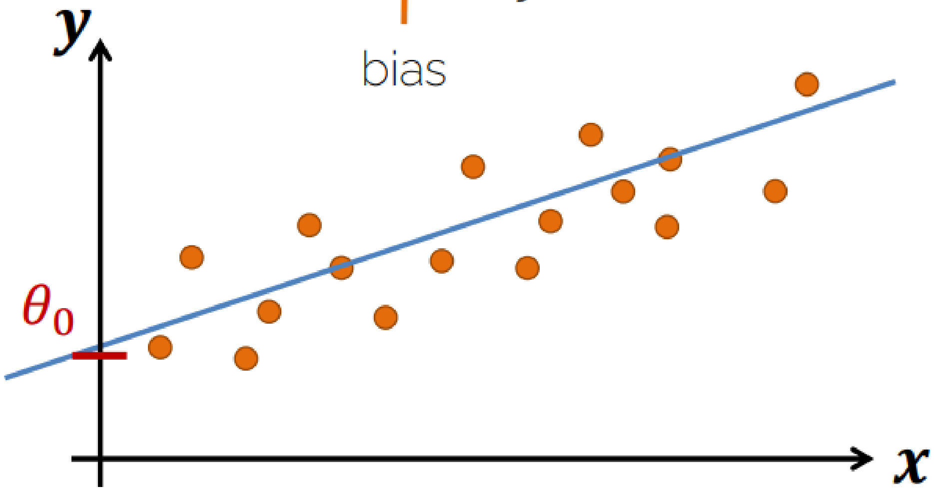
Input data, features

Predicción lineal

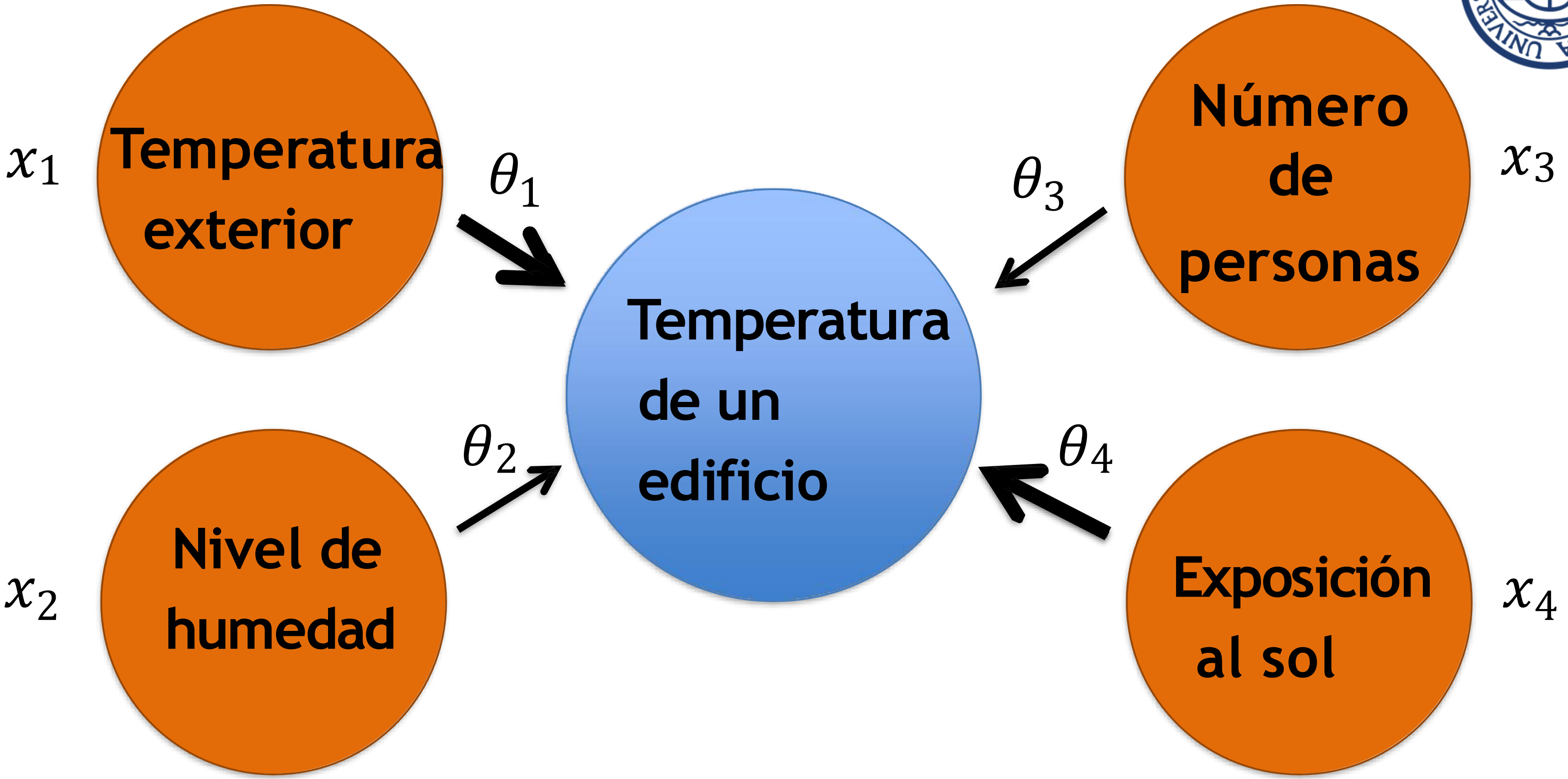
- **Un modelo lineal se expresa de la forma**

$$\hat{y}_i = \boxed{\theta_0} + \sum_{j=1}^{d} x_{ij}\theta_j = \theta_0 + x_{i1}\theta_1 + x_{i2}\theta_2 + \cdots + x_{id}\theta_d$$

bias



Predicción lineal



Predicción lineal

$$\begin{bmatrix} \boxed{\hat{y}_1} \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \theta_0 + \begin{bmatrix} \boxed{x_{11} & \cdots & x_{1d}} \\ x_{21} & \cdots & x_{2d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \quad \Rightarrow \quad \hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta}$$

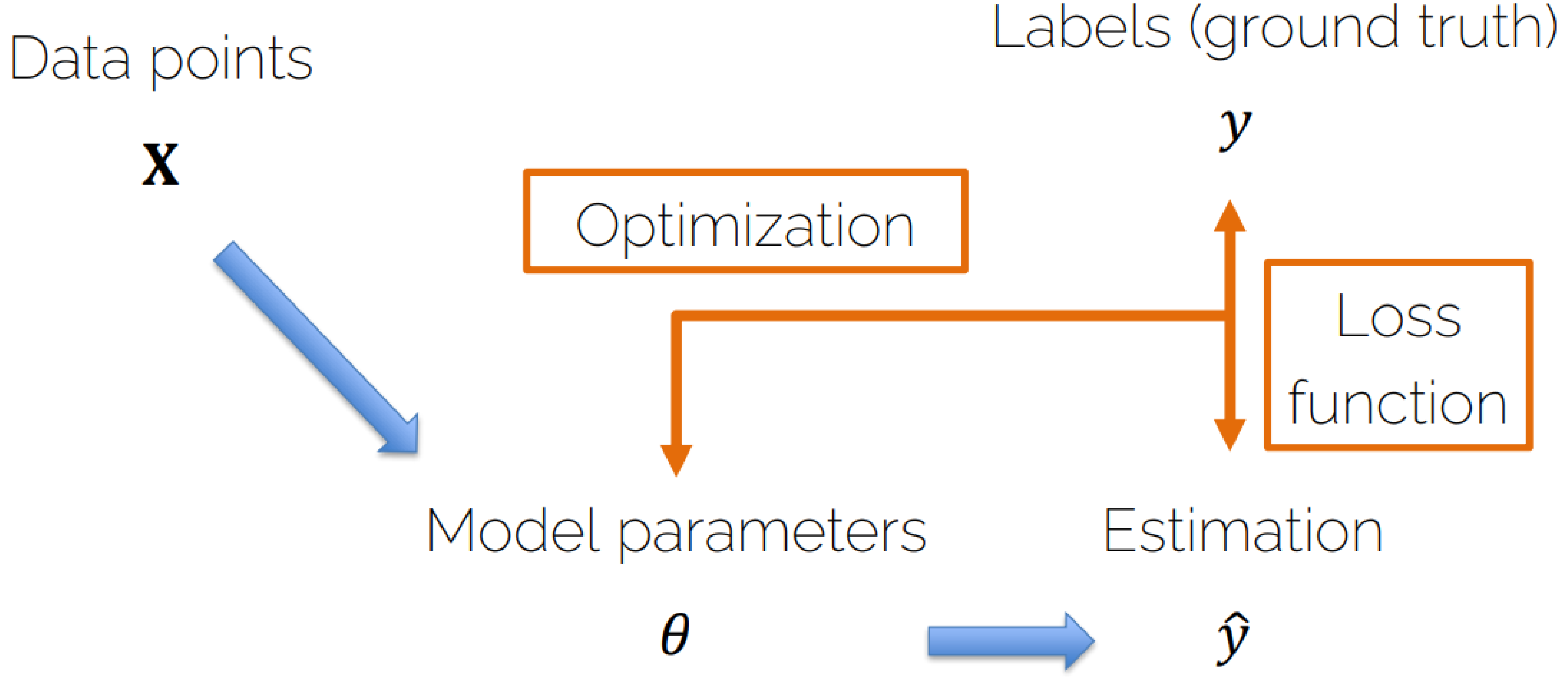
Predicción lineal

Temperatura del edificio

MODELO

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{array}{c} \begin{array}{ccccc} \text{Bias} & \text{Outside temperature} & \text{Humidity} & \text{Number people} & \text{Sun exposure (\%)} \end{array} \\ \begin{bmatrix} 1 & 25 & 50 & 2 & 50 \\ 1 & -10 & 50 & 0 & 10 \end{bmatrix} \end{array} \cdot \begin{bmatrix} 0.2 \\ 0.64 \\ \boxed{0} \\ \boxed{1} \\ \boxed{0.14} \end{bmatrix}$$

¿Cómo obtener el modelo?

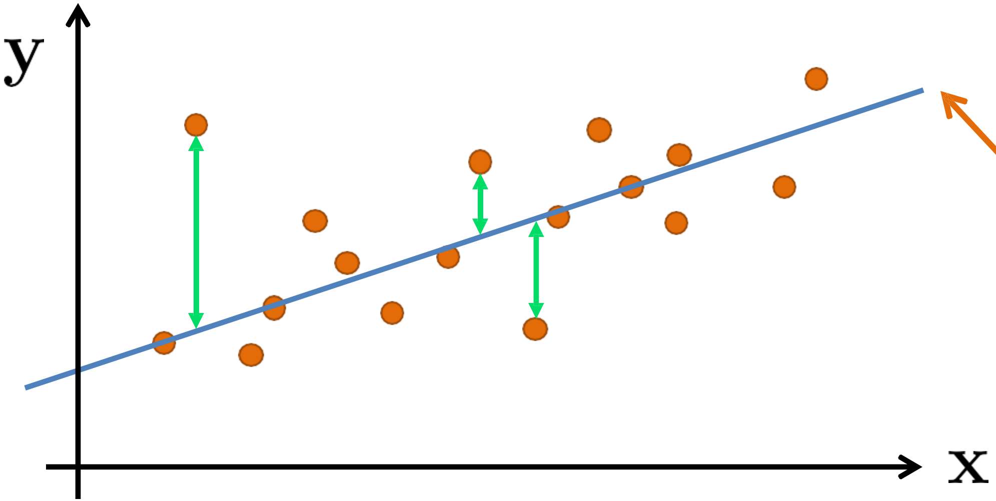


¿Cómo obtener el modelo?

- **Función de pérdida:** mide lo buena que es mi estimación (lo bueno que es mi modelo) e indica al método de optimización cómo mejorarlo.

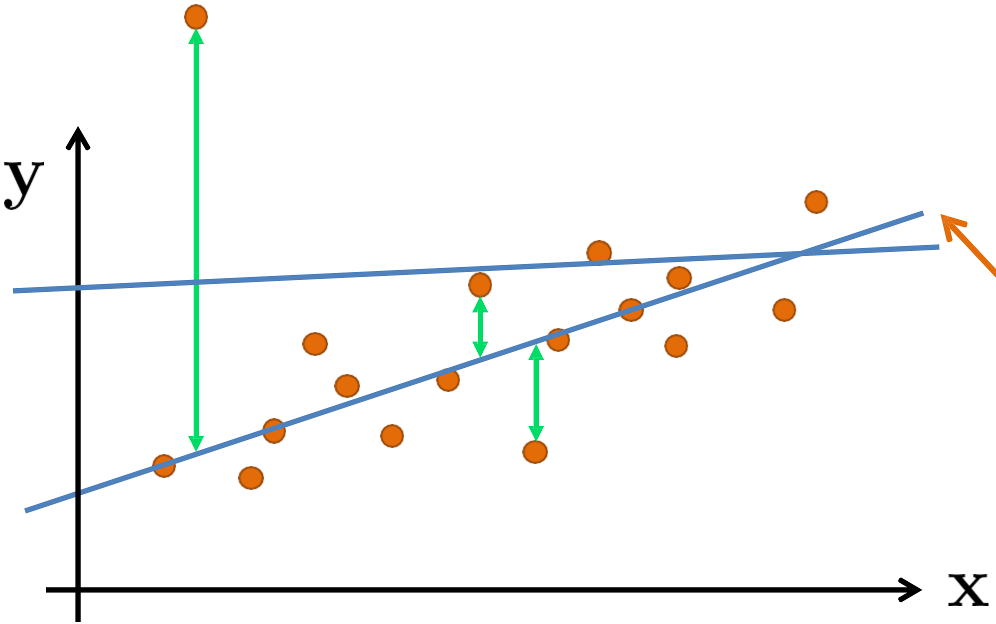
- **Optimización**: modifica el modelo para mejorar la función de pérdida (mejorar mi estimación).

Regresión lineal: Loss Function



Predicción:
Temperatura
del edificio

Regresión lineal: Loss Function



Predicción: Temperatura del edificio

Minimización

Función objetivo Energía
Función de coste

Optimización: Cuadrados Mínimos

- Cuadrados mínimos lineales: enfoque para ajustar un modelo lineal a los datos

$$\min_{\theta} \; J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

- Problema convexo, existe una solución de forma cerrada que es única. Es decir, hay un óptimo único

Optimización

$$\frac{\partial J(\theta)}{\partial \theta} = 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - 2\mathbf{X}^T\mathbf{y} = 0$$

$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Hemos encontrado una solución analítica a un problema convexo

Inputs: Temperatura exterior, número de personas,.

True Output: Temperatura del edificio

¿Es ésta la mejor estimación?

- **Estimación por cuadrados mínimos**

$$\min_{\theta} \; J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

Maximum Likelihood

Estimación de Maximum Likelihood (MLE)

$p_{data}(\mathbf{y}|\mathbf{X})$ Verdadera distribución subyacente

$\uparrow$

$p_{model}\ (\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ Familia paramétrica de distribuciones

Controlada por parámetro(s)

Estimación de Maximum Likelihood

- Método de estimación de los parámetros de un modelo estadístico dadas las observaciones,

$$p_{model}\ (\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$$

Observaciones de $pdata(\mathbf{y}|\mathbf{X})$

Estimación de Maximum Likelihood

- Método de estimación de los parámetros de un modelo estadístico dadas las observaciones, encontrando los valores de los parámetros que MAXIMIZAN LA PROBABILIDAD de realizar las observaciones.

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \; p_{model}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$$

Estimación de Maximum Likelihood

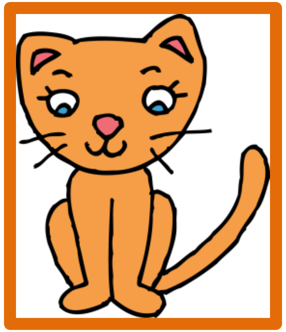
$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

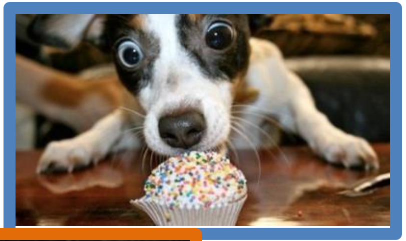
Regresión lineal

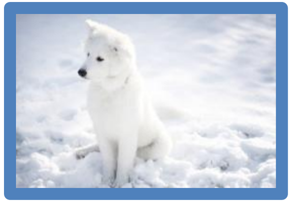
- La estimación de máxima verosimilitud (MLE) corresponde a la estimación por cuadrados mínimos (dados los supuestos)

- Introducción a los conceptos de función de pérdida y optimización para obtener el mejor modelo de regresión

Image Classification







Regression vs Clasificación

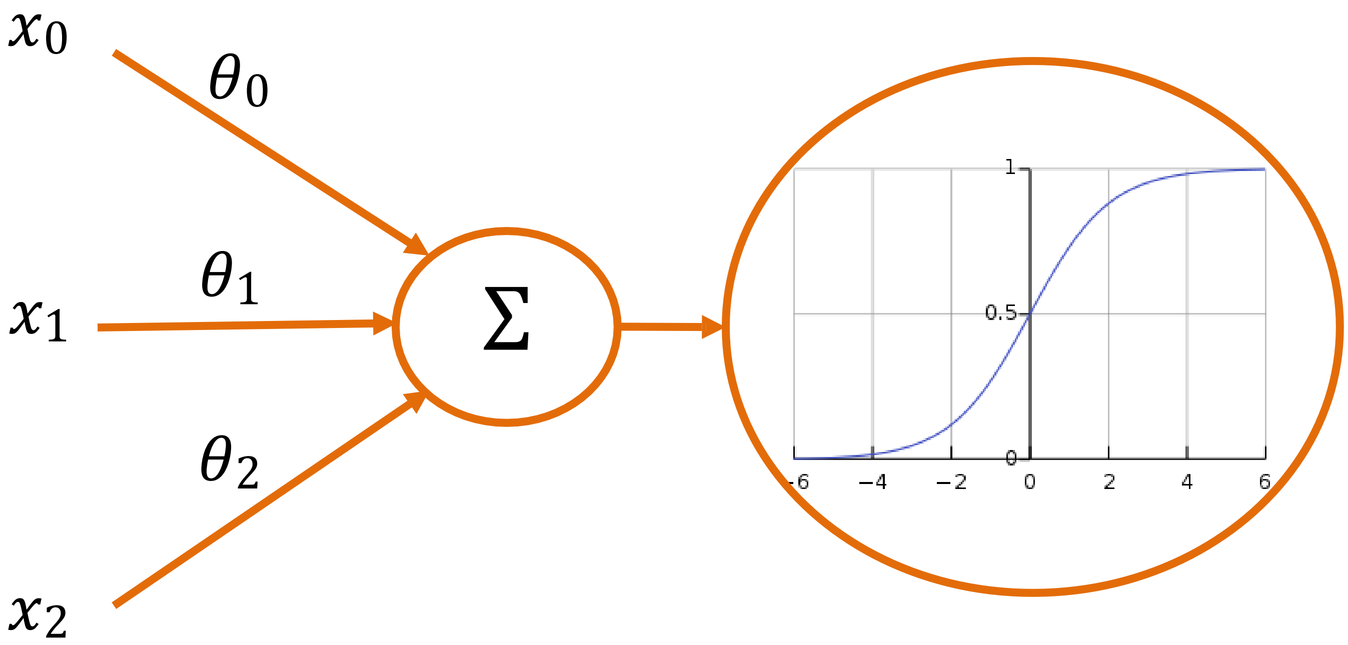
- Regresión: predecir un valor de salida continuo (por ejemplo, la temperatura de una habitación)

- Clasificación: predice un valor discreto
  - Clasificación binaria: la salida es 0 ó 1
  - Clasificación multiclase: conjunto de N clases

Regresión logística

Clasificador CAT

Sigmoide para predicciones binarias



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

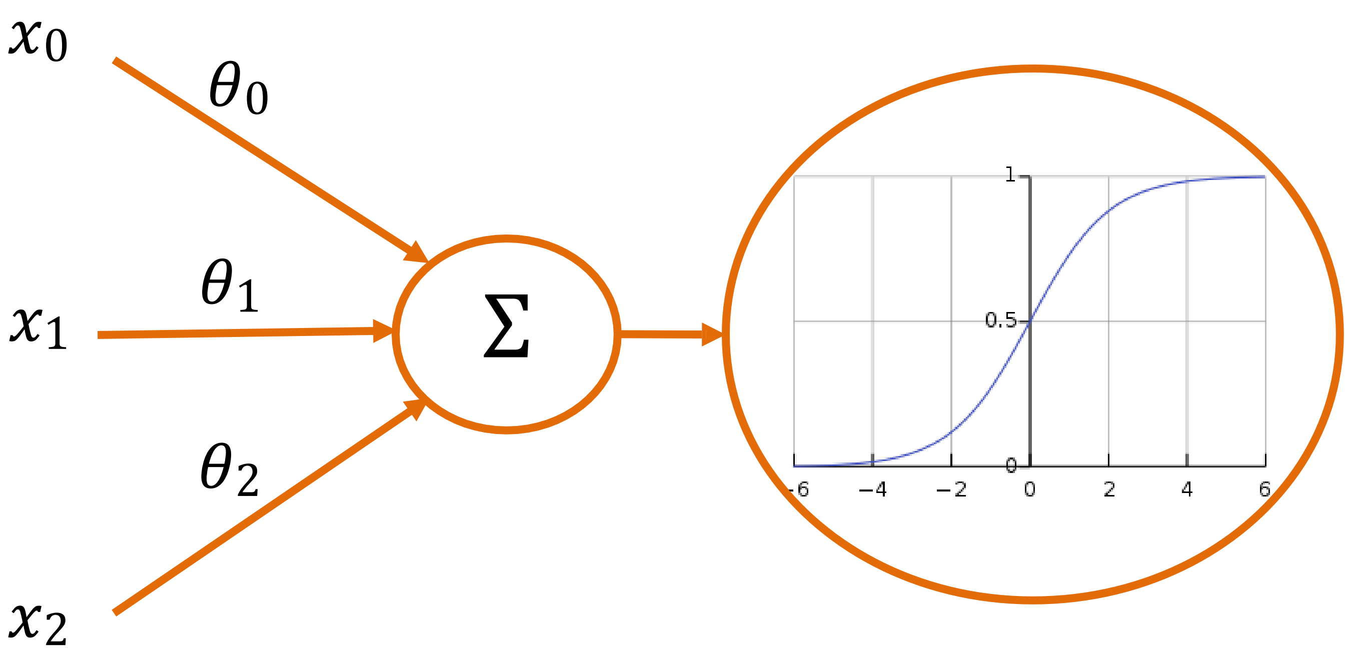
1

Puede interpretarse como una probabilidad

0

$$p(y = 1 | \mathbf{x}, \boldsymbol{\theta})$$

Red neuronal de 1 capa



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

1

Puede interpretarse como una probabilidad

0

$$p(y = 1 | \mathbf{x}, \boldsymbol{\theta})$$

Regresión logística

- Probabilidad de una salida binaria

$$\hat{\mathbf{y}} = p(\mathbf{y} = 1|\mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(y_i = 1|\mathbf{x}_i, \boldsymbol{\theta})$$

The prediction of our sigmoid

$$\hat{y}_i = \sigma(\mathbf{x}_i \boldsymbol{\theta})$$

Regresión logística: Loss Function

- Probability of a binary output

$$p(\mathrm{y}|\mathbf{X},\boldsymbol{\theta}) = \hat{\mathbf{y}} = \prod_{i=1}^{n} \hat{y}_i^{y_i}(1-\hat{y}_i)^{(1-y_i)}$$

- Maximum Likelihood Estimate

$$\boldsymbol{\theta}_{\boldsymbol{ML}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathrm{y}|\mathbf{X},\boldsymbol{\theta})$$

Regresión logística: Loss Function

$$p(\mathrm{y}|\mathbf{X},\boldsymbol{\theta}) = \hat{\mathbf{y}} = \boxed{\prod_{i=1}^{n}} \hat{y}_i^{y_i}(1-\hat{y}_i)^{(1-y_i)}$$

$$\boxed{\sum_{i=1}^{n}\log}\left(\hat{y}_i^{y_i}(1-\hat{y}_i)^{(1-y_i)}\right)$$

$$\sum_{i=1}^{n} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$$

Regresión logística: Loss Function

$$\mathcal{L}(\hat{y}_i, y_i) = -[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

Regresión logística: Loss Function

$$\mathcal{L}(\hat{y}_i, y_i) = -[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

Denominada pérdida de *entropía cruzada binaria* (BCE)

- Relacionada con la multi-class los que lo vamos a ver (también llamada *softmax loss*)

Regresión logística: Optimización

- Loss function

$$\mathcal{L}(\hat{y}_i, y_i) = -[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

- Cost function

$$C(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\hat{y}_i, y_i)$$

Minimization

$$\hat{y}_i = \sigma(\mathbf{x}_i \boldsymbol{\theta})$$

$$= -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)$$

Regresión logística: Optimización

- No hay solución de forma cerrada

- Utilizar un método iterativo → gradient descent

gradient descent - más adelante

Por qué está bueno el Machine Learning

- Podemos aprender de la experiencia
  -> Inteligencia, cierta capacidad para inferir el futuro

- Incluso los modelos lineales suelen ser bastante buenos para fenómenos complejos: por ejemplo, el tiempo atmosférico.

Parte Práctica

- Ejercicio de esta semana: Math Recap II

- Próxima clase:
  - Introducción a Redes Neuronales y Grafos computacionales

Referencias para lecturas complementarias

Cross Validation:

- https://medium.com/@zstern/k-fold-cross-validation-explained-5aeba90ebb3
- https://towardsdatascience.com/train-test-split-and-validación-cruzada-en-python-80b61beca4b6

Libro de General Machine Learning:

- Reconocimiento de patrones y ML. C. Bishop.

Hasta la semana que viene ☺