

Aprendizaje Automático

Segundo Cuatrimestre de 2016



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Aprendizaje Automático

Segundo Cuatrimestre de 2016

Primera mitad de la clase de hoy:

- Cuestiones administrativas: horarios, docentes, web, etc.
- Objetivos, correlatividades, evaluación, bibliografía.
- Repaso superficial del contenido de toda la materia.
- Taller: setup de cuentas, Linux, Python.

Segunda mitad: Primera clase teórica.

Aprendizaje Automático

Segundo Cuatrimestre de 2016

Docentes:	Agustín Gravano (profesor) Facundo Carrillo (ayudante de primera)
Horario y lugar:	Miércoles 13 a 17 horas. Aula 2 y laboratorio 4.
Lista de mail:	aa-alu@dc.uba.ar
Página web:	www.dc.uba.ar/aa
Puntajes:	3 puntos para Licenciatura y Doctorado (Computación)
Correlativas:	Métodos Numéricos, Teoría de Lenguajes

Aprendizaje Automático

Segundo Cuatrimestre de 2016

Objetivos:

El Aprendizaje Automático (o *Machine Learning*) es una rama de la Inteligencia Artificial que se dedica al estudio de los programas que aprenden a realizar una tarea en base a la experiencia.

Esta materia consiste en una introducción abarcativa de las principales técnicas y aplicaciones del área, con un balance entre teoría y ejercitación práctica.

Aprendizaje Automático

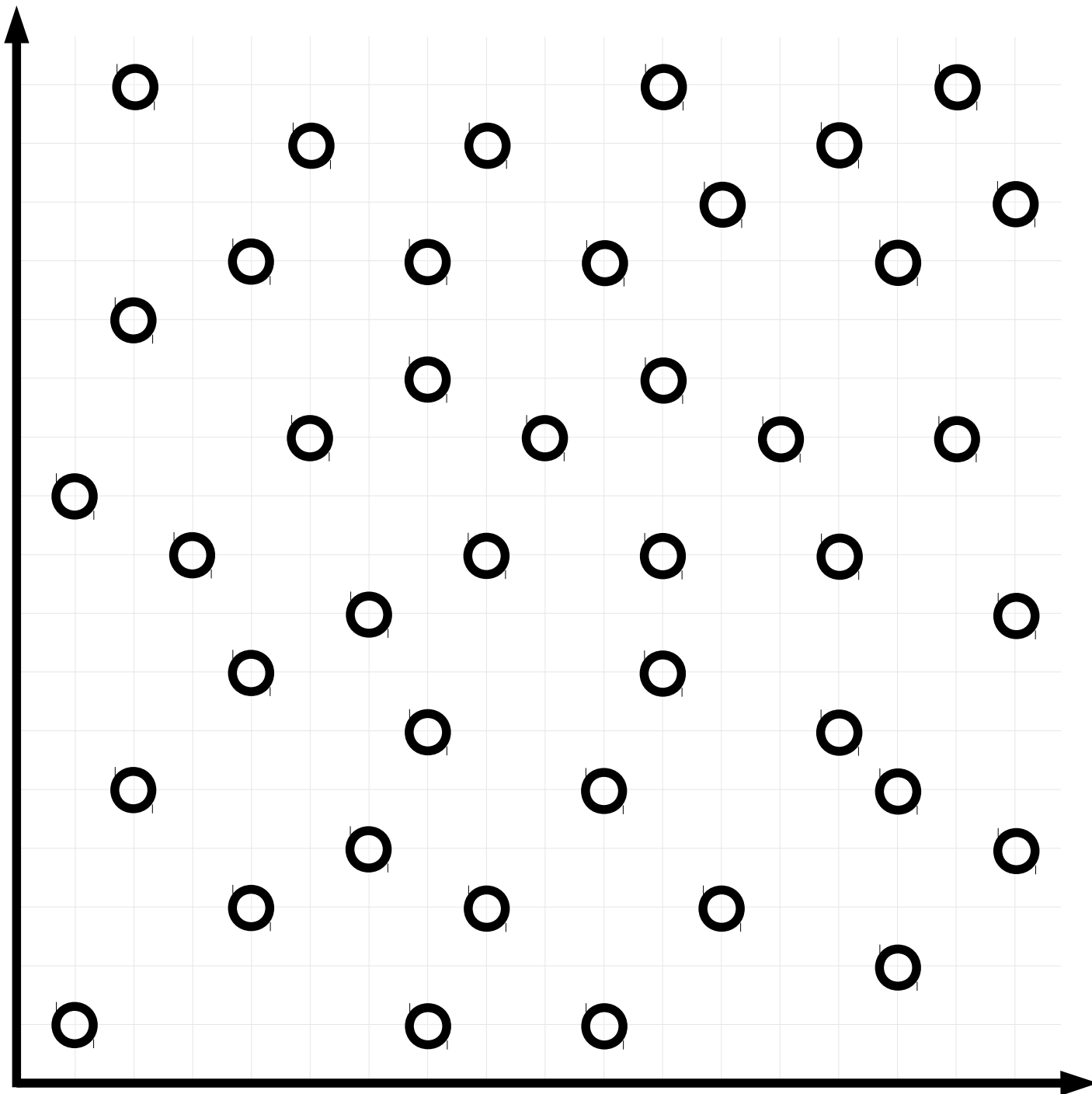
Segundo Cuatrimestre de 2016

Modo de evaluación:

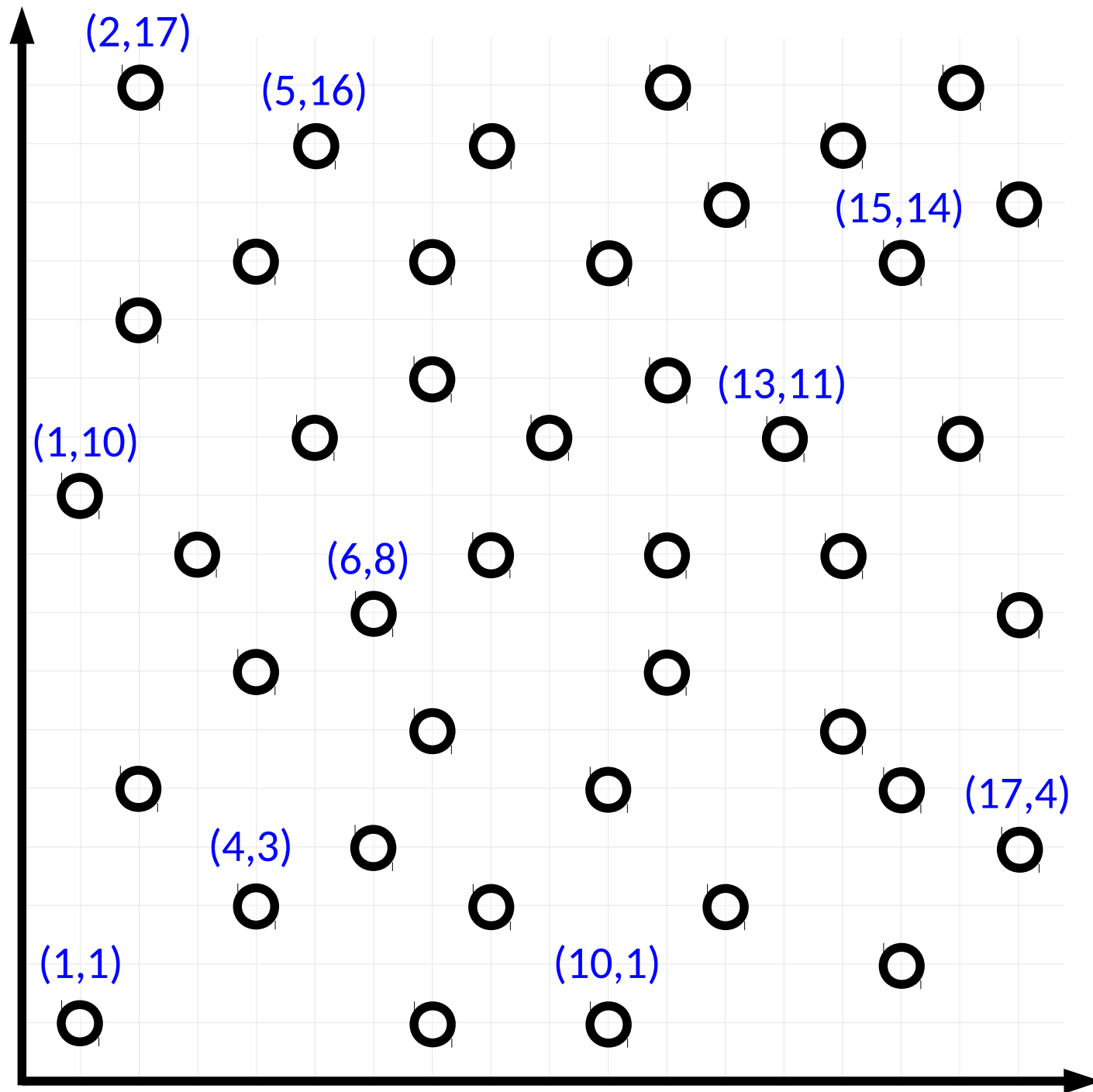
- Dos exámenes parciales.
- Dos trabajos prácticos grupales (tres integrantes).
- Promoción:
 - La materia se promociona obteniendo nota 8 o superior en todas las instancias.
 - Quienes obtengan al menos una nota inferior a 8 deben dar el final.
 - Para quienes recuperen un parcial o TP, la nota que cuenta es la del recuperatorio.
 - La promoción es opcional: pueden optar por dar final para levantar la nota.

Bibliografía:

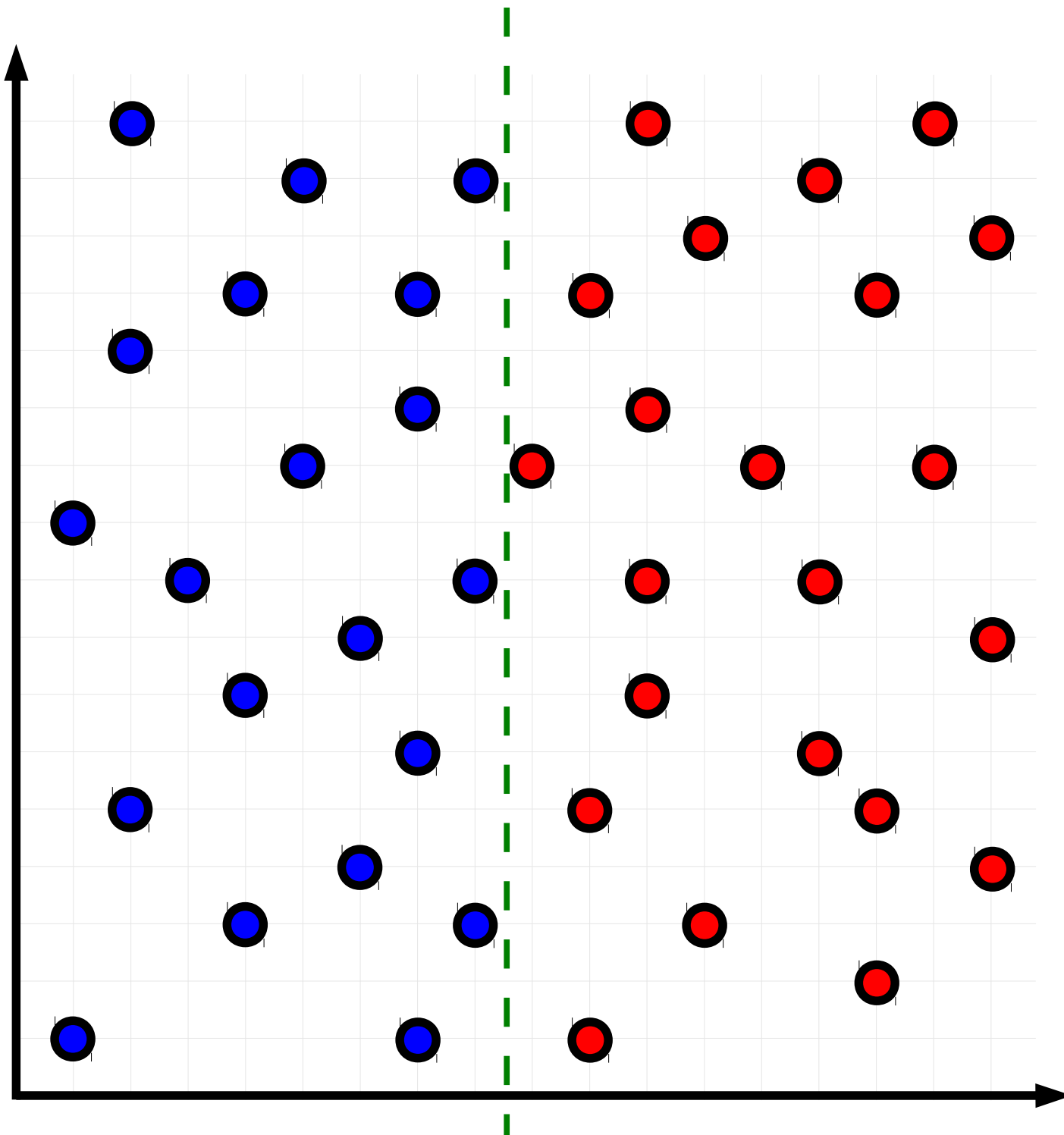
- Mitchell, “Machine Learning”, McGraw-Hill, 1997.
- Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006.
- Marsland, “Machine Learning. An Algorithmic Perspective”, 2nd ed, Chapman&Hall, 2015.
- Duda, Hart & Stork, “Pattern Classification”, 2nd ed, Wiley, 2001.



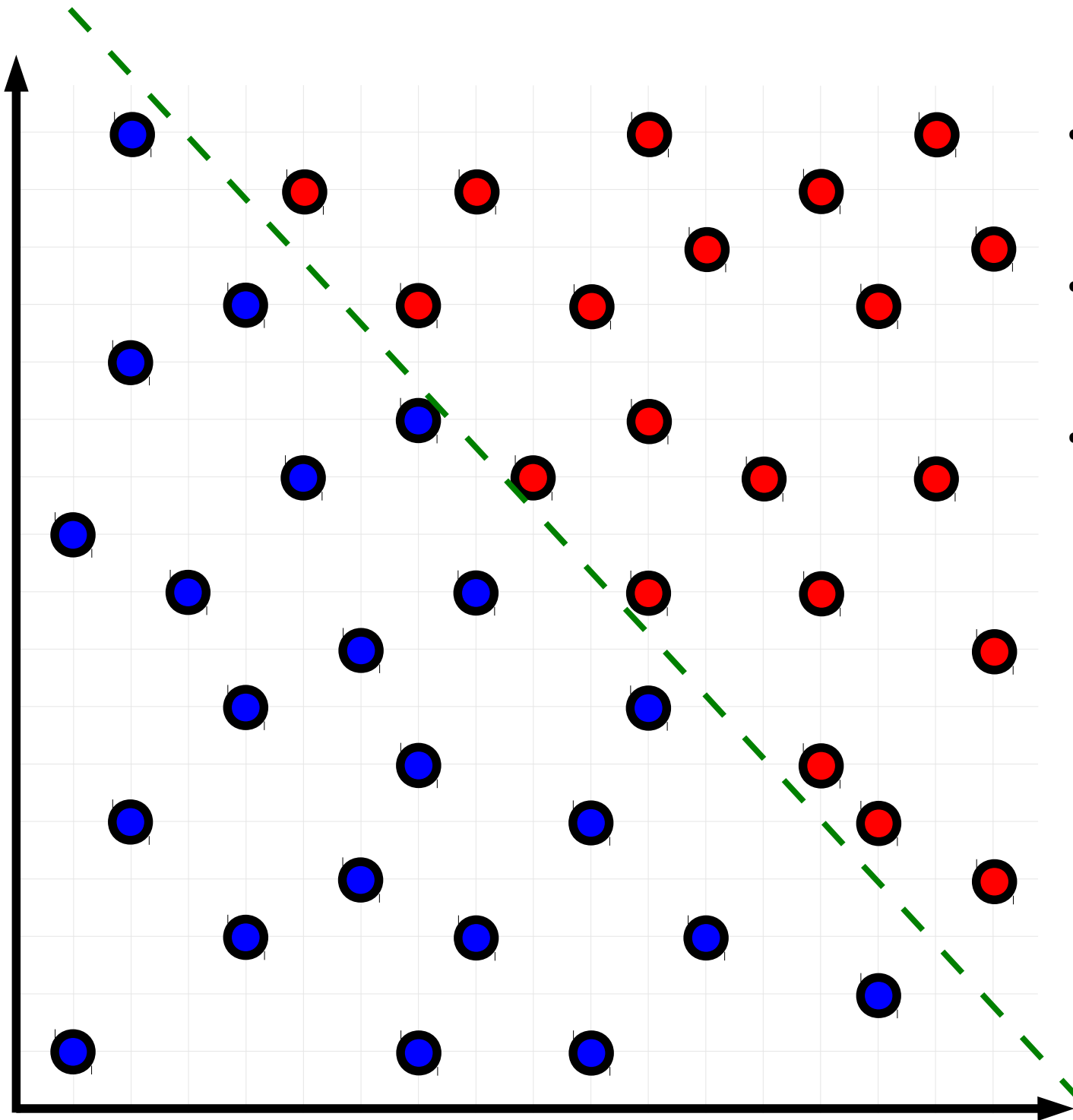
- Tenemos N puntos en el plano.



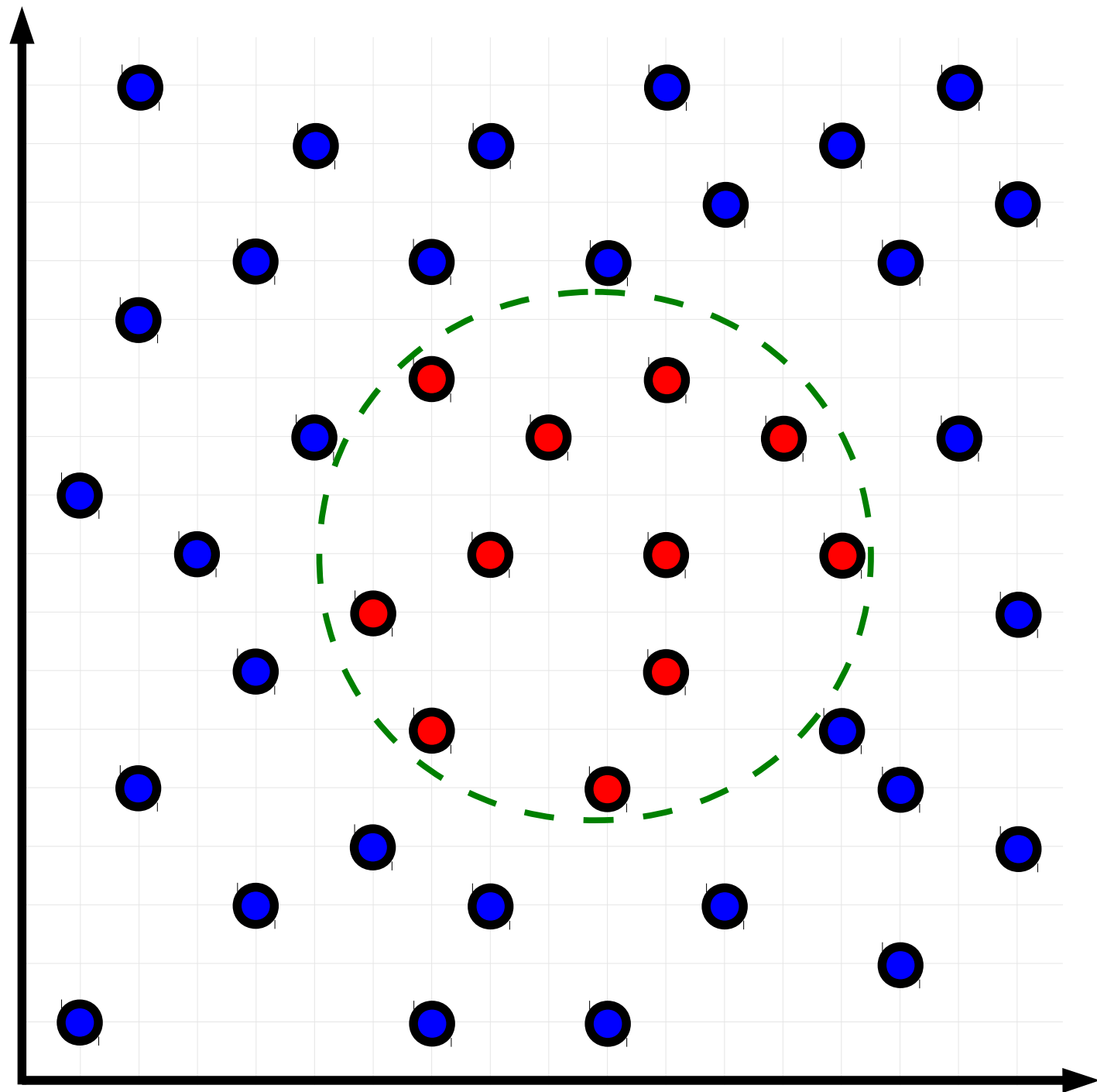
- Tenemos N puntos en el plano.
- Cada punto tiene dos coordenadas: (x, y) .

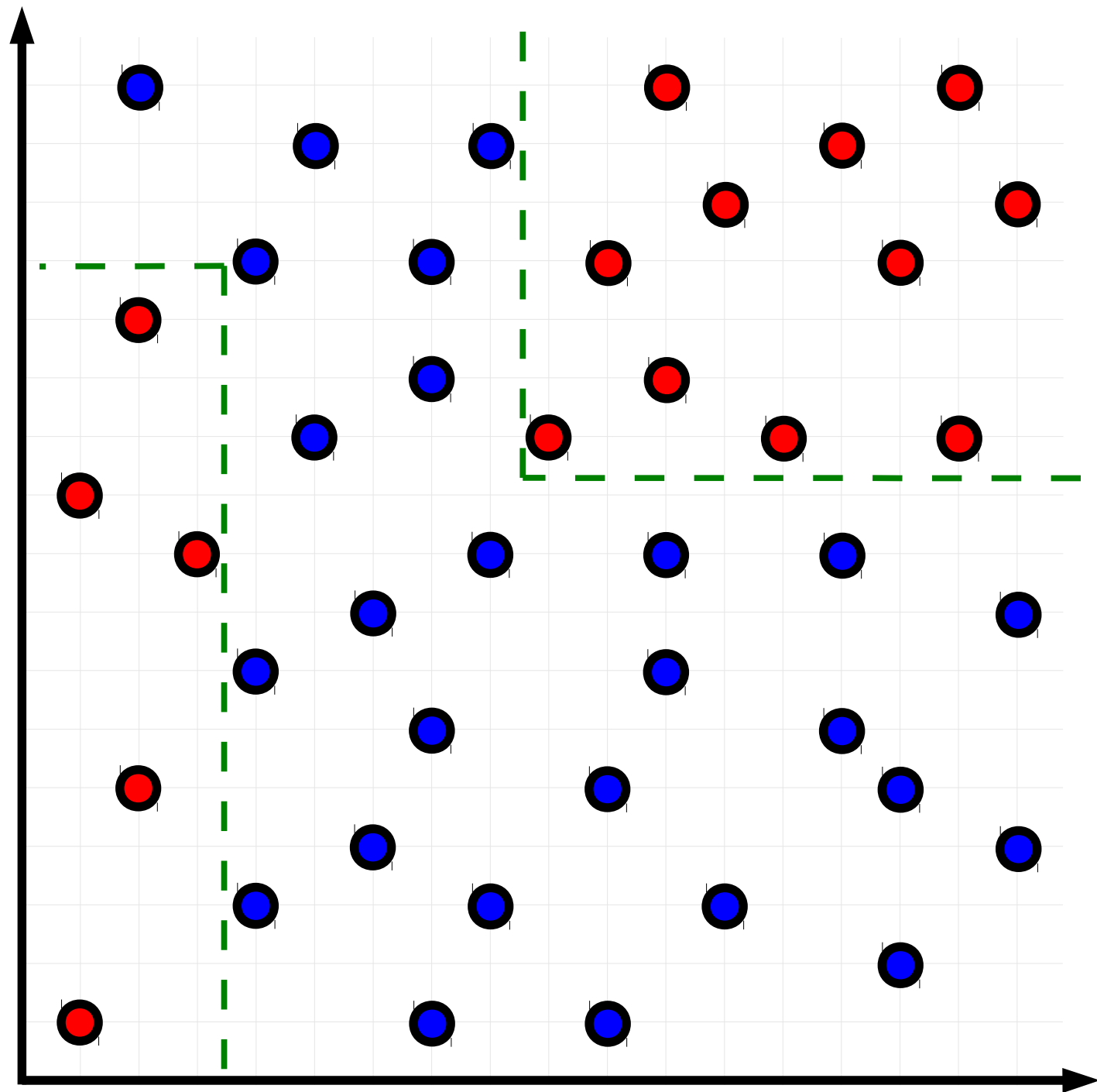


- Tenemos N puntos en el plano.
- Cada punto tiene dos coordenadas: (x, y) .
- Cada punto tiene un color: **azul** o **rojo**.
- Queremos hallar una forma de **predecir** el color de puntos **nuevos**.
- Función $f(x, y) \rightarrow \text{color}$
- $f(x, y) =$
rojo si $x > 8$
azul en caso contrario



- $f(x, y) =$
rojo si $y > m x + b$
azul en caso contrario
- m y b son parámetros del modelo que deben ajustarse a los datos.
- También es un parámetro el **color superior**.





Humanos vs. Máquinas

- Los humanos somos buenos encontrando (y programando) estas reglas en 2D.
- Pero, ¿qué pasa si los puntos tienen miles de coordenadas?
- Ejemplo: Detección de caras.



- Los humanos somos muy buenos detectando (y reconociendo) caras.
- Pero ¿podemos **programar estas funciones?**

Aprendizaje Automático

Un programa **aprende** una tarea, si su performance mejora con la experiencia.

Tenemos que definir:

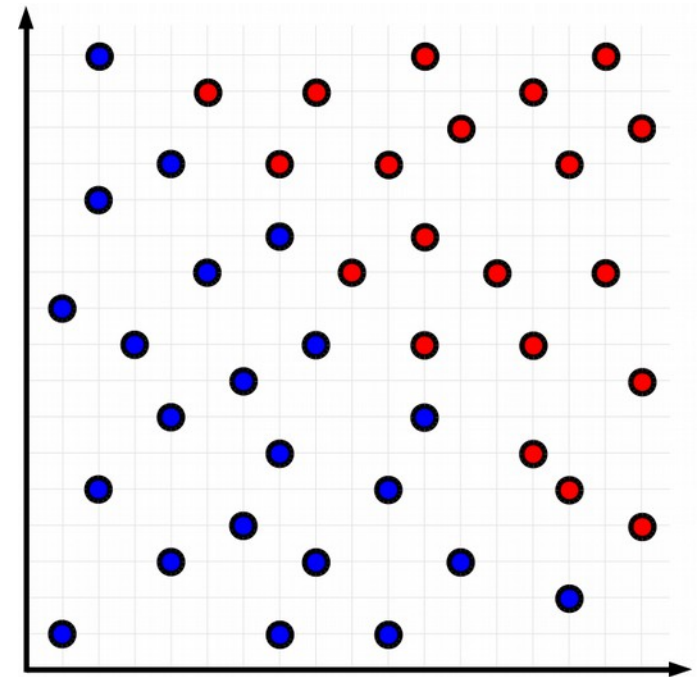
- **Tarea**
- **Medida de performance**
- **Experiencia**

Aprendizaje Automático

Un programa **aprende** una tarea, si su performance mejora con la experiencia.

Ejemplo:

- **Tarea:** Predecir el color de un punto.
- **Medida de performance:** % puntos coloreados correctamente.
- **Experiencia:** Base de datos de puntos con su color correspondiente.

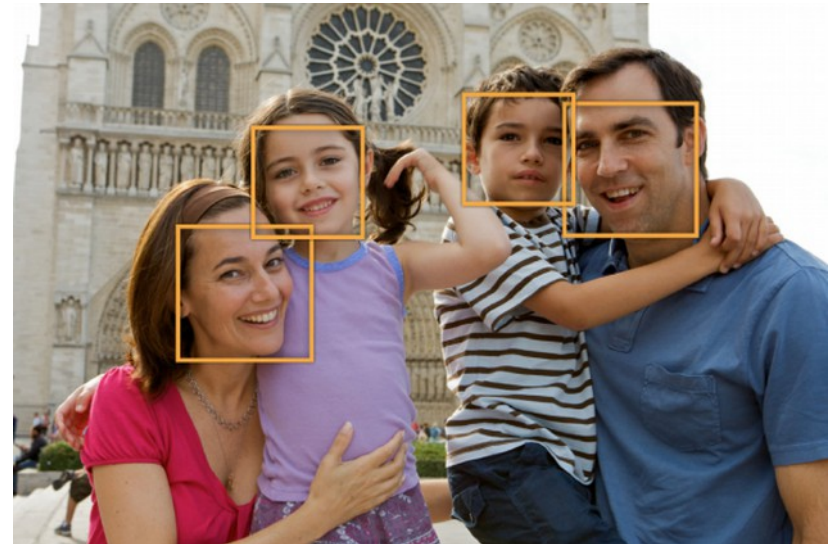


Aprendizaje Automático

Un programa **aprende** una tarea, si su performance mejora con la experiencia.

Ejemplo:

- **Tarea:** Detectar caras en una imagen.
- **Medida de performance:** % caras detectadas correctamente.
- **Experiencia:** Base de datos de imágenes con las caras marcadas.

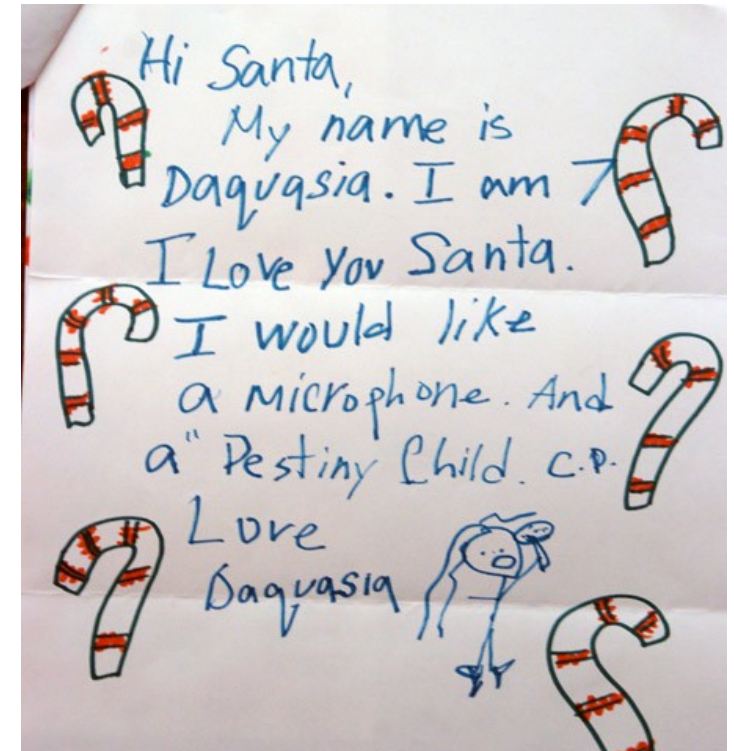


Aprendizaje Automático

Un programa **aprende** una tarea, si su performance mejora con la experiencia.

Ejemplo:

- **Tarea:** Reconocer textos manuscritos.
- **Medida de performance:** % palabras reconocidas correctamente.
- **Experiencia:** Base de datos de palabras manuscritas con sus transcripciones correctas.



Aprendizaje Automático

Un programa **aprende** una tarea, si su performance mejora con la experiencia.

Ejemplo:

- **Tarea:** Manejar un auto por la calle usando sensores visuales.
- **Medida de performance:** Distancia promedio recorrida hasta cometer un error.
- **Experiencia:** Secuencia de *mediciones* (ej: imágenes, acelerómetros) y *acciones* (acelerar, frenar, doblar, etc.) grabados mientras conduce un ser humano.



Aprendizaje Supervisado

- Los datos de entrenamiento están **anotados** con la respuesta correcta.
 - Típicamente: anotación manual, costosa!
 - Amazon Mechanical Turk
 - reCAPTCHA
 - etc.



Provide 3 tags for this image.

Instructions:

- You must provide 3 tags for this image.
- Each tag must be a single word
- No tag can be longer than 25 characters
- The tags must describe the image, the contents of the image, or some relevant context.

Image:



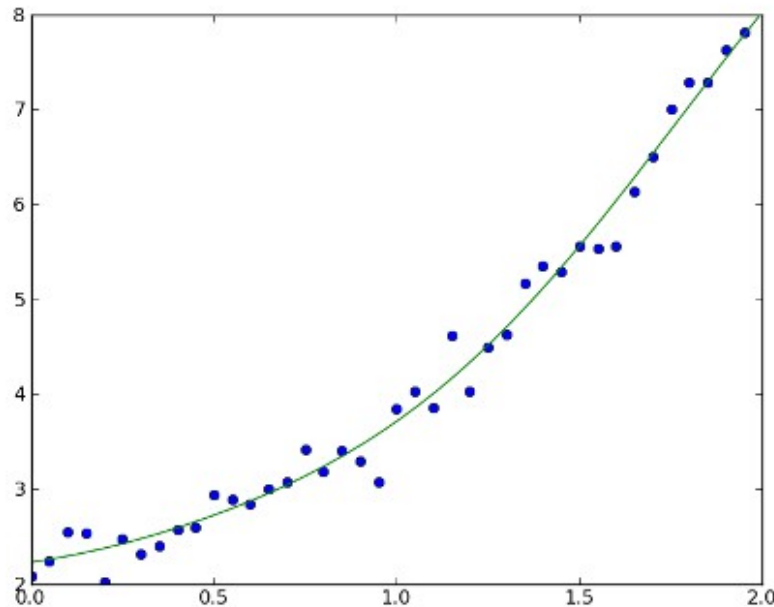
Tag 1:

Tag 2:

Tag 3:

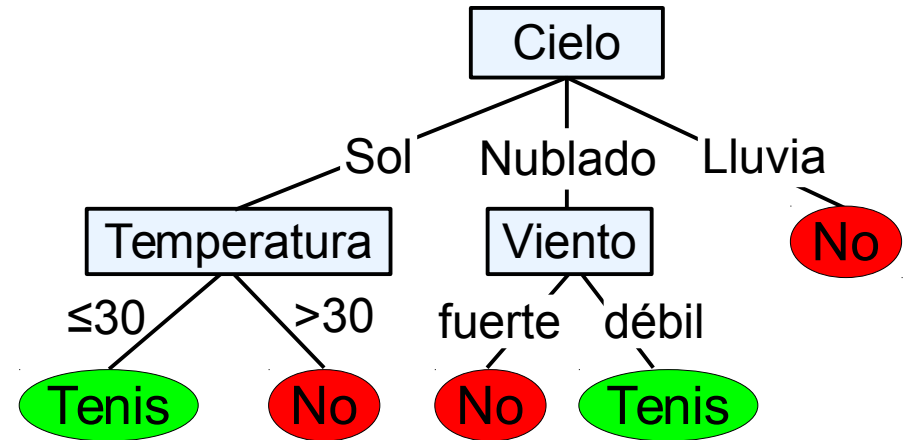
Aprendizaje Supervisado

- Tipos de aprendizaje supervisado:
 - **Clasificación**: Cada instancia pertenece a una **clase**.
 - SpamFilter: mensaje \rightarrow {spam, no-spam}
 - **Regresión**: Cada instancia tiene un **valor numérico**.
 - SpamFilter: mensaje \rightarrow probabilidad $[0,1]$ de ser spam.



Algoritmos de Aprendizaje Supervisado

- Árboles de decisión



- Reglas

IF (Cielo=Sol \wedge Temperatura>30) THEN Tenis=No

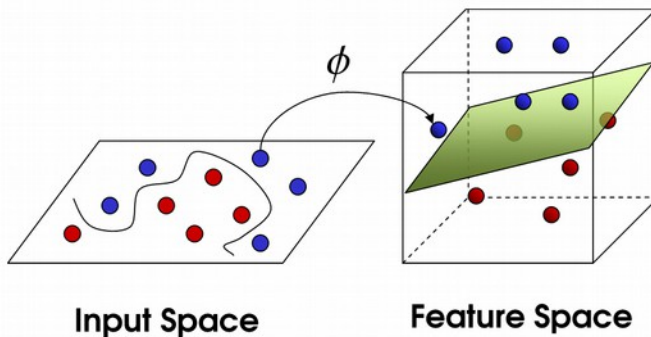
IF (Cielo=Nublado \wedge Viento=Débil) THEN Tenis=Sí

IF (Cielo=Lluvia) THEN Tenis=No

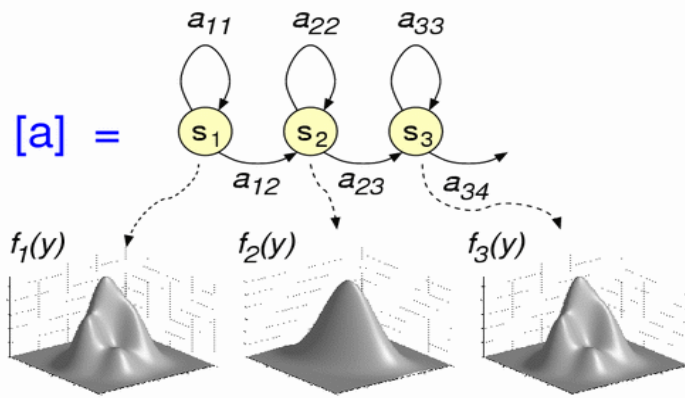
- Vecinos más cercanos (KNN)
- Naive Bayes
- ...

Algoritmos de Aprendizaje Supervisado

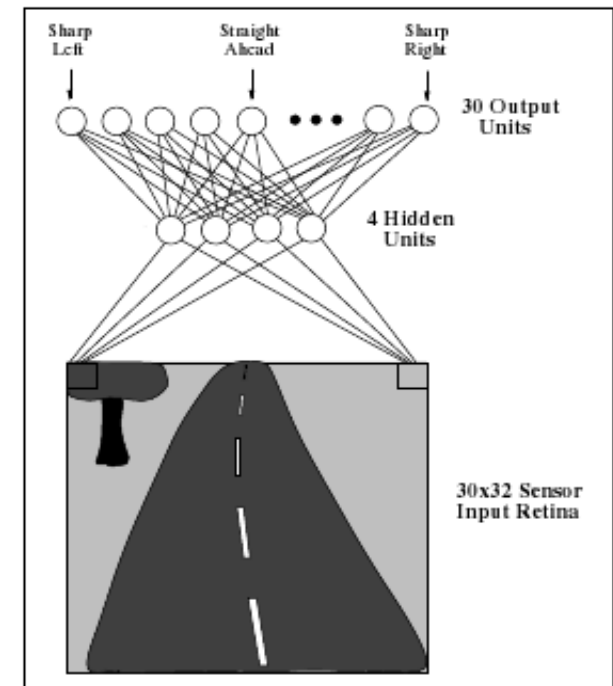
- Support Vector Machines



- Hidden Markov Models

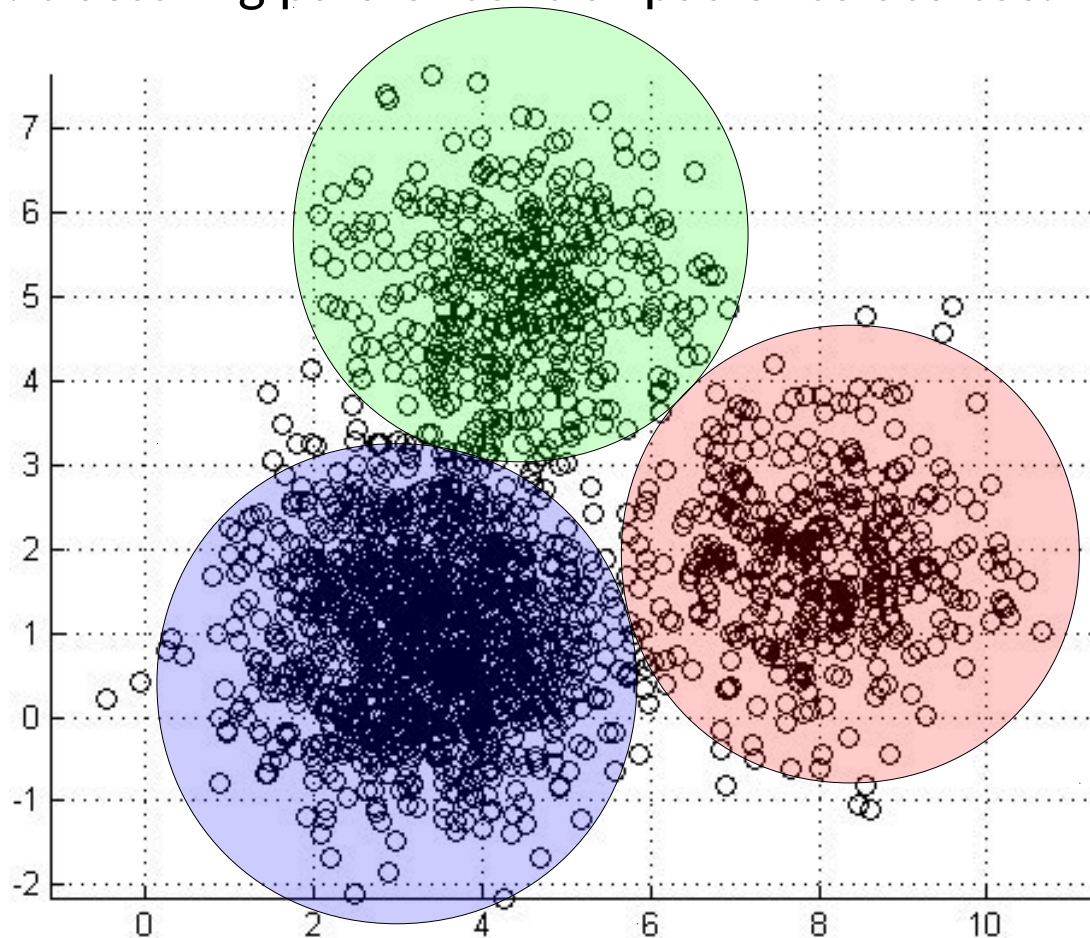


- Redes Neuronales



Aprendizaje No Supervisado

- Los datos de entrenamiento **no están anotados**.
 - Big Data.
 - Ejemplo: clustering para encontrar patrones ocultos.



Aprendizaje Automático

- Aprendizaje **supervisado**
 - Los datos de entrenamiento están anotados.
 - Clasificación y Regresión.
- Aprendizaje **no supervisado**
 - Los datos de entrenamiento no están anotados.
- Aprendizaje **por refuerzos**
 - Aprendizaje gradual, en base a premios y castigos.

Aprendizaje por Refuerzos

- Conjunto de **estados** que definen el medio.
- Conjunto de **acciones** que el agente puede realizar.
- Reglas de **transiciones** entre estados.
- Reglas que asignan una **recompensa** inmediata a cada transición.
- Reglas que determinan qué **observa** el agente.
- Ejemplos: control de robots, programa de ascensores, ruteo de paquetes, juego del Go.



Esquema General de Aprendizaje

- Definición de la tarea de aprendizaje.
 - Instancias, clases, medidas de performance.
- Recolección y preparación de **datos**.
 - Cantidad vs. calidad de datos. Ruido.
 - Extracción de atributos.
- Experimentación
 - Selección de los atributos más útiles.
 - Elección de algoritmos.
 - Entrenamiento y validación de modelos.
- Evaluación del modelo final sobre datos frescos.

Cronograma Tentativo

- 10/8 (Hoy). Aprendizaje de conceptos.
- 17/8. Árboles de decisión.
- 24/8. Evaluación de hipótesis.
- 31/8. Clasificadores: KNN, Naive Bayes, SVM, Random Forest.
- 7/9. Taller.
- 14/9. Reducción de dimensionalidad. Selección y transformación de atributos.
- Hidden Markov Models. Modelos generativos y discriminativos. Ensamble de modelos. Regresión lineal simple, múltiple, logística. Redes Neuronales. Aprendizaje no supervisado. Clustering. Algoritmo EM. Aprendizaje por refuerzos. Aplicaciones.