

# Aprendizaje Automático

# Aprendizaje automático

- Aprendizaje de máquina -> Aprendizaje automático



coffee  
machine

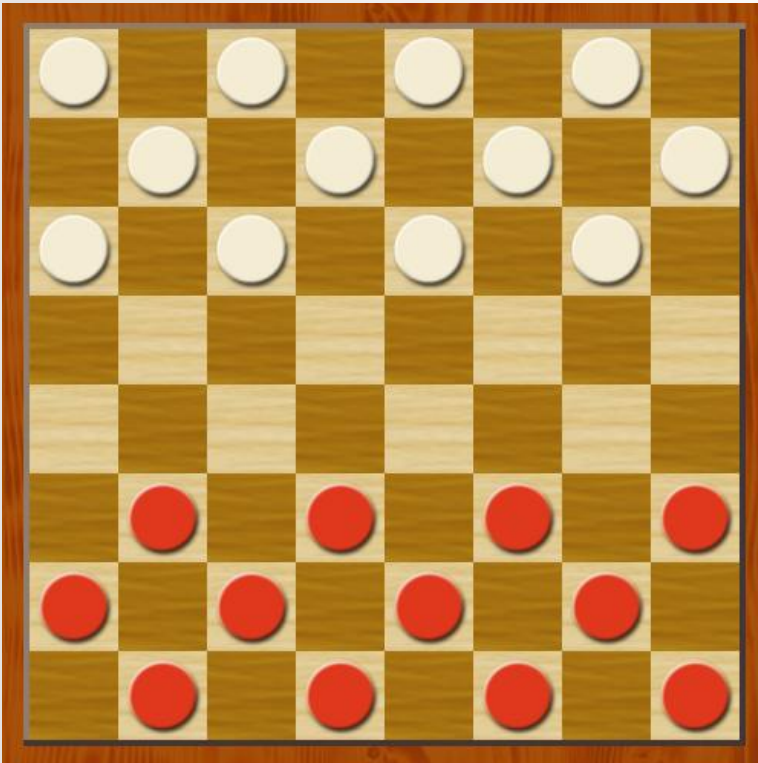


washing machine



answering  
machine

# Conceptos y fundamentos



Juego de damas

## ► Aprendizaje automático

- Término acuñado en 1959 por Arthur Samuel en el contexto de resolver el juego de las damas mediante una máquina
- Contraparte de la programación convencional (algoritmos deterministas y no deterministas)
- El programa o la máquina pueden deducir información por sí mismas y luego aplicarla en otros momentos.
- Un programa puede aprender para producir un comportamiento para el cual no fue programado y del cual el programador puede no estar “consciente”



Pros y contras

# Pros

- Inteligencias artificiales que aprenden por sí solas



- Hacer nuevas jugadas
- Adelantarse al oponente
- ...

# Contras

- Comportamientos inesperados



- Chatbot Tay
- Microsoft (2016)
- Conversaciones informales y entretenidas con jóvenes de 18 a 24 años
- En menos de 24 horas se volvió grosera, racista y xenófoba

# Aprendizaje Automático (AA)

- El AA es una subárea de la IA que se ocupa de la inducción de un modelo mediante un proceso de aprendizaje.
- Un área particular del AA, denominada Aprendizaje Inductivo, consiste en técnicas que inducen a estos modelos utilizando un conjunto de instancias o ejemplos previamente conocidos, denominados instancias de entrenamiento. Una vez inducido el modelo, puede aplicarse a datos nuevos desconocidos para el modelo

# Tareas del AA

- Tareas predictivas (aprendizaje supervisado)
  - Clasificación
  - Regresión
  - Optimización
- Tareas descriptivas (aprendizaje no supervisado)
  - Agrupación
  - Asociación

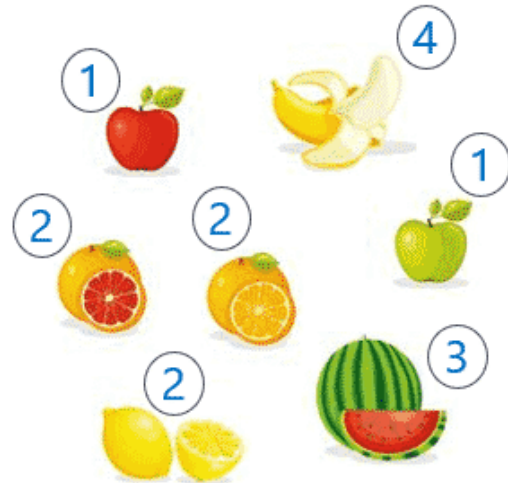
# Tipos de aprendizaje

- En el aprendizaje automático se distinguen dos tipos de aprendizaje
  - Aprendizaje supervisado
  - Aprendizaje no supervisado

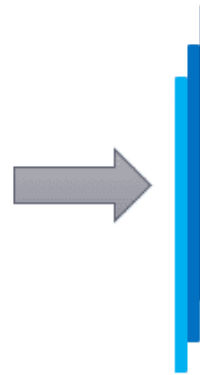


# Aprendizaje supervisado

## CONJUNTO DE ENTRENAMIENTO



Vectores de características



Algoritmo de aprendizaje automático



## CONJUNTO DE TEST



Vector de características



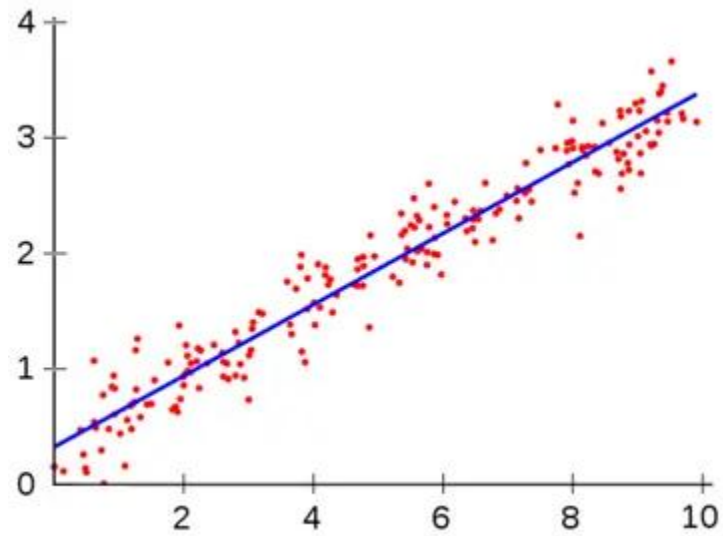
Modelo predictivo



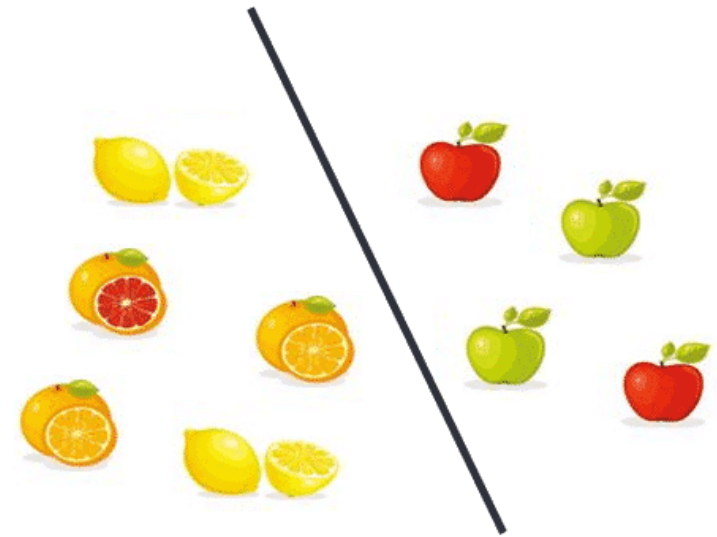
## RESULTADO



# Aprendizaje supervisado



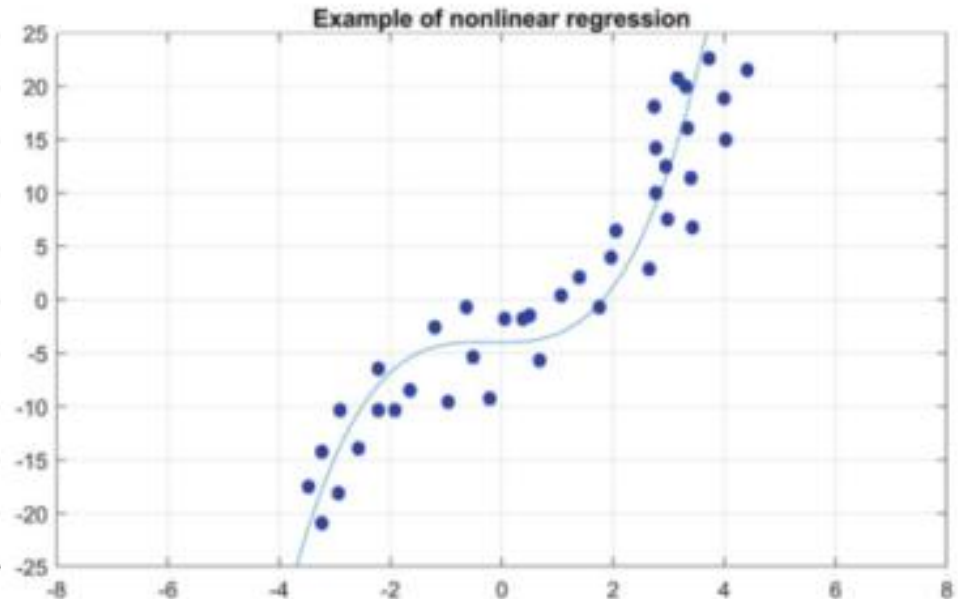
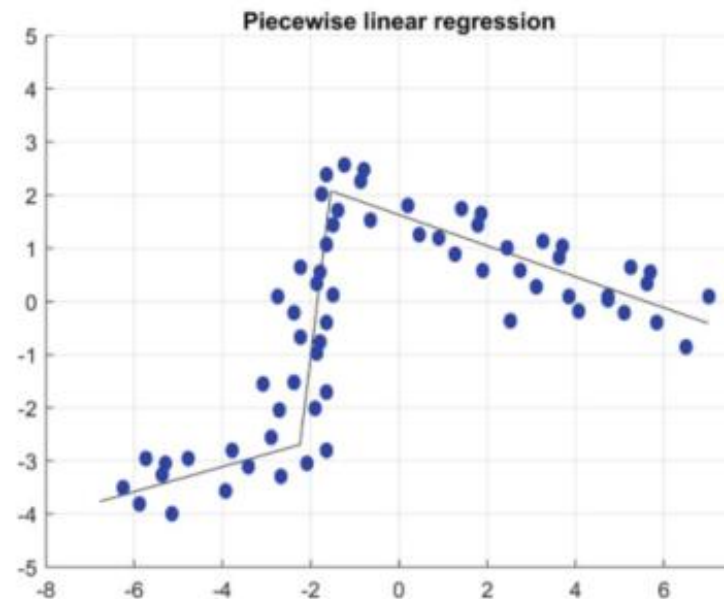
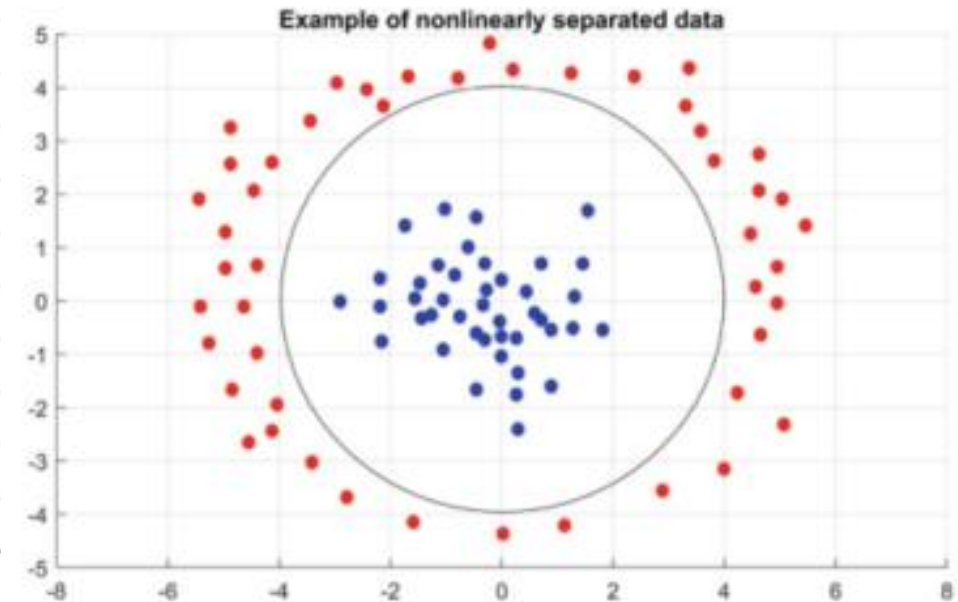
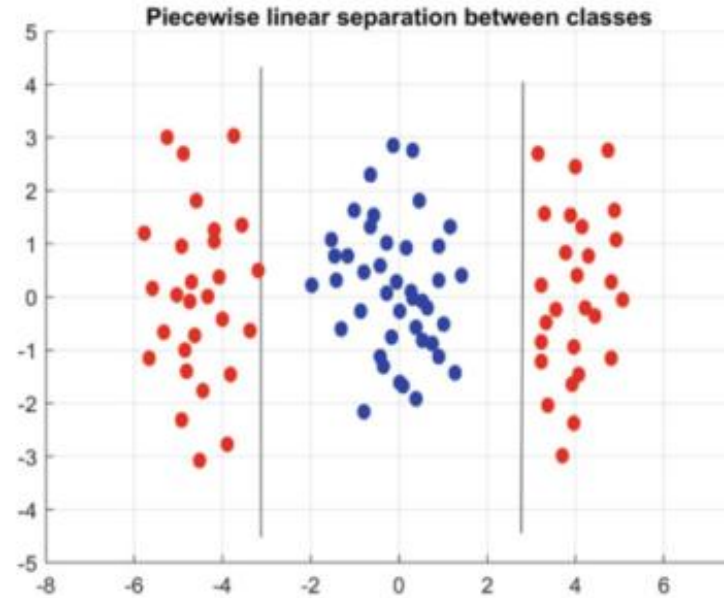
Regresión



Clasificación

## Linealidad vs No Linealidad

- En tareas de predictivas del AA como clasificación y regresión podemos encontrar linealidad y no linealidad
  - Datos no lineales con modelos lineales aplicados por partes para clasificar o predecir
  - Datos no lineales que requieren de modelos no lineales para clasificar o predecir



# Evaluación del aprendizaje supervisado

- Existen varias métricas para evaluar las predicciones realizadas por el modelo generado
  - Matriz de confusión (Confusion matrix)
  - Exactitud (Accuracy)
  - Precisión (Precision)
  - Exhaustividad o sensibilidad (Recall)
  - F1 o medida F (F-measure)
  - ...

# Aprendizaje no supervisado

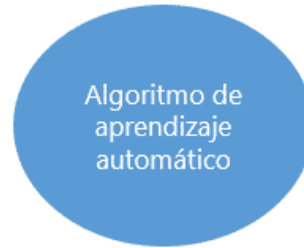
## CONJUNTO DE ENTRENAMIENTO



Vectores de características



Algoritmo de aprendizaje automático



## CONJUNTO DE TEST



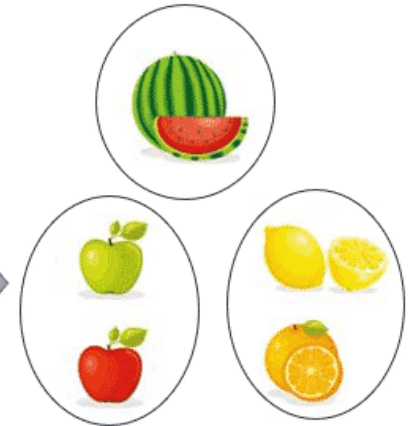
Vector de características



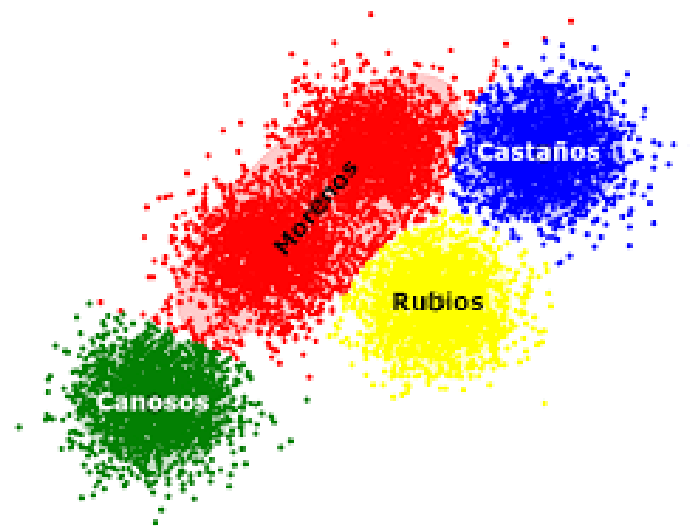
Modelo predictivo



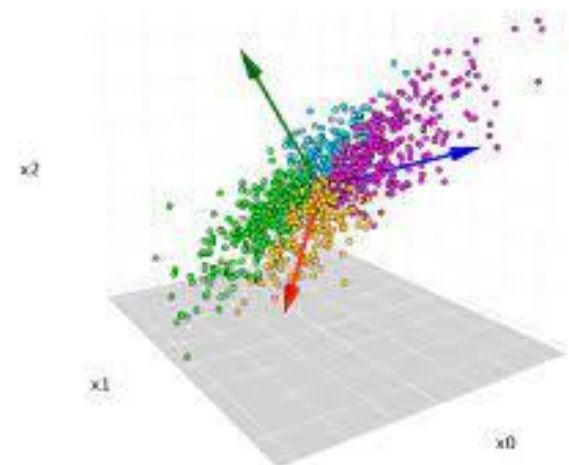
## RESULTADO



# Técnicas de aprendizaje no supervisado



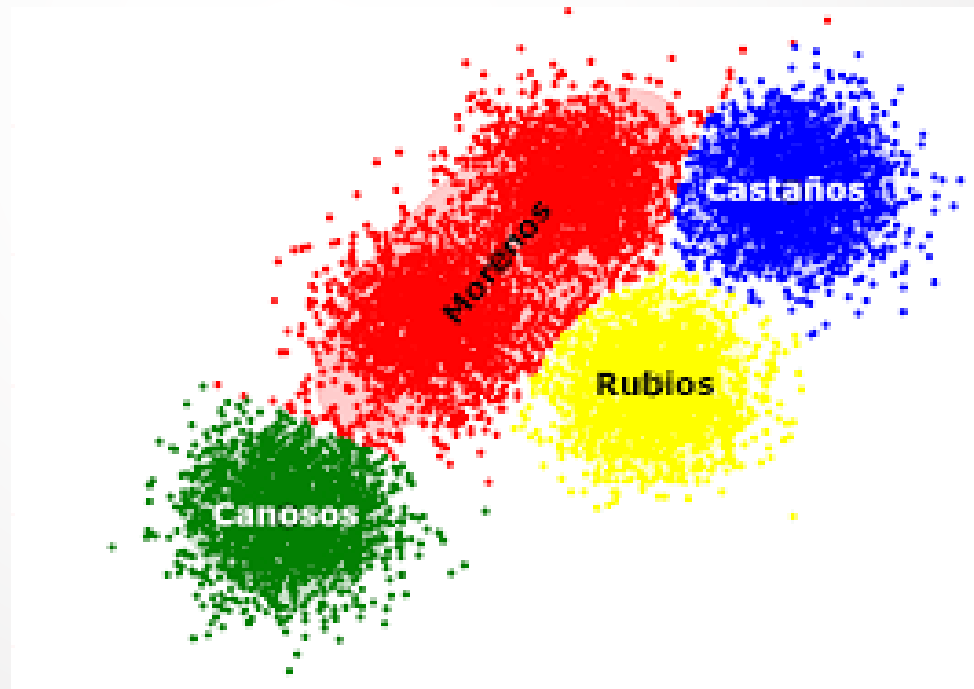
Basadas en agrupamiento (Clustering)



Análisis de componentes principales (PCA)

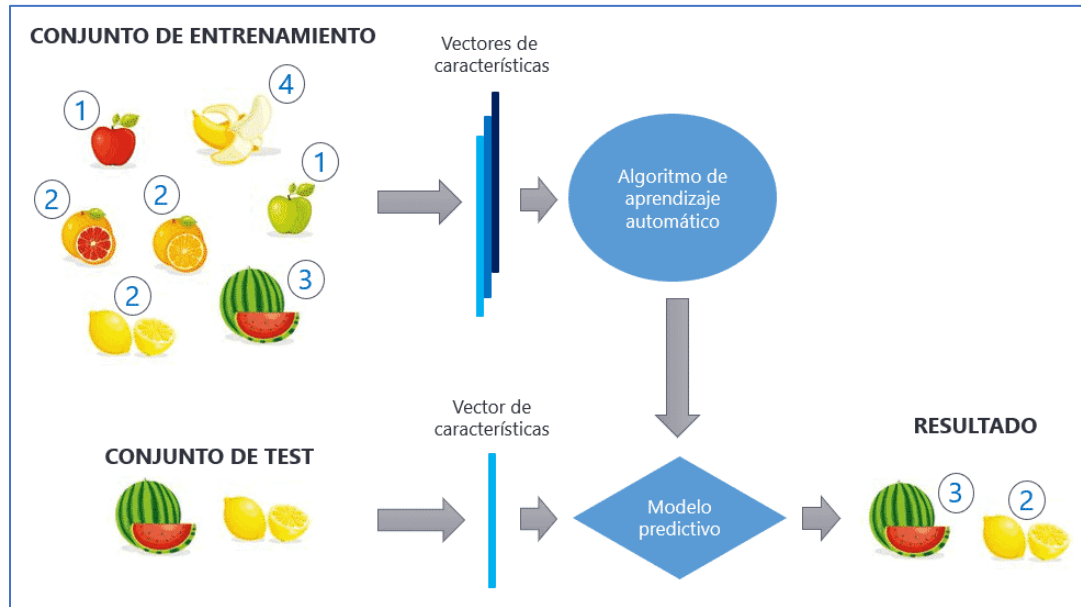
# Evaluación del aprendizaje no supervisado

- En la evaluación de este tipo de aprendizaje se siguen dos enfoques
  - Validación interna
  - Validación externa

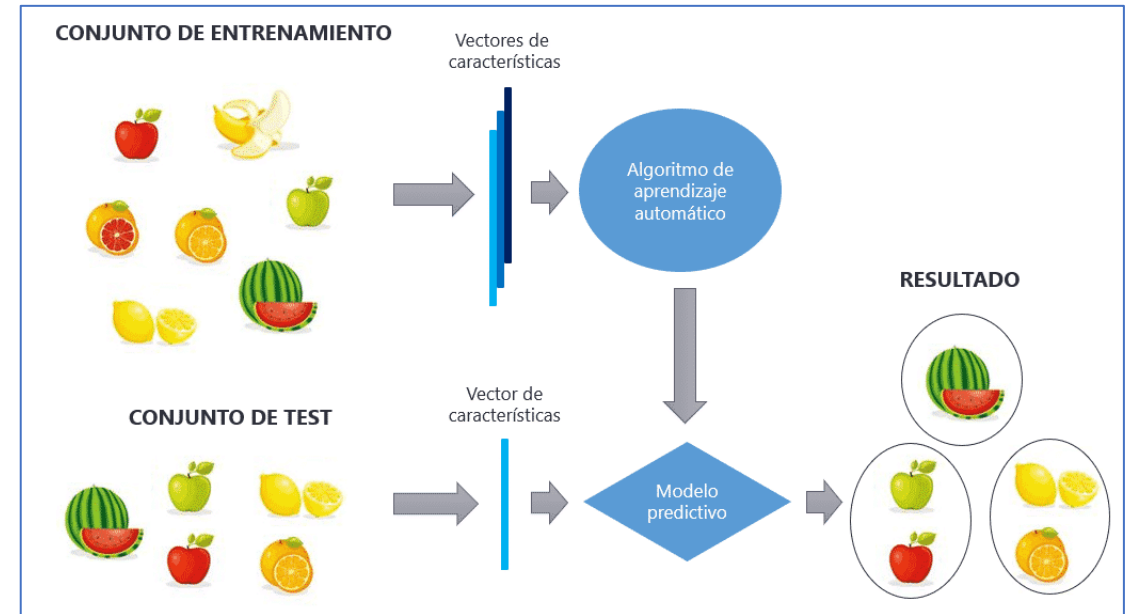


# Tipos de aprendizaje

## Aprendizaje supervisado



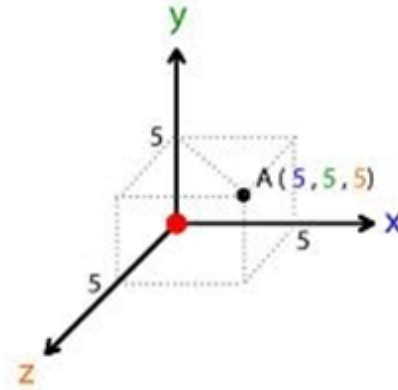
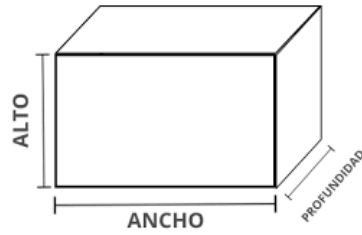
## Aprendizaje no supervisado



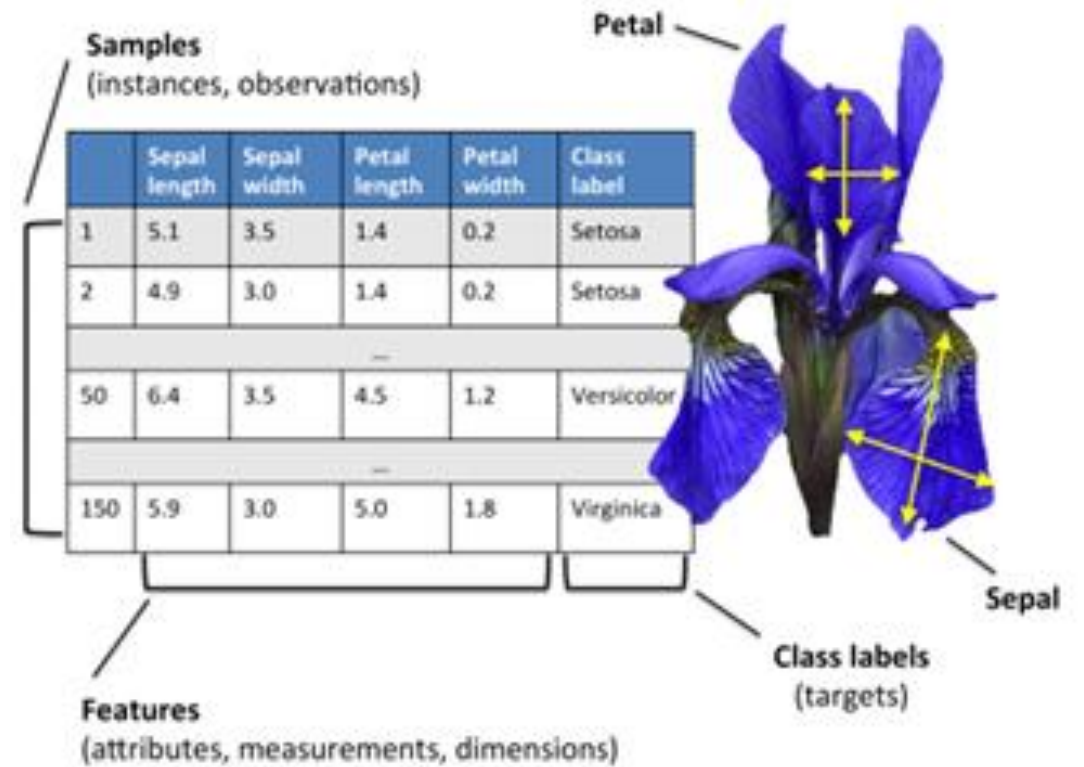


# El problema de la dimensionalidad

- Física



- AA



# Teorema de No Free Lunch

- David H. Wolpert y William G. Macready (1997)
- “Por cada par de algoritmos de búsqueda, hay tantos problemas en el que el primer algoritmo es mejor que el segundo como problemas en el que el segundo algoritmo es mejor que el primero”
- Si un algoritmo es mejor que otro en una clase de problemas, lo contrario se va a cumplir en otra clase de problemas
- El mostrar superioridad de un algoritmo sobre otro(s) no quiere decir que sea mejor en todos los casos

equals

$$\sum_{a \subset d, a' \subset d'} \sum_{a \supseteq d, a' \supseteq d'} P(c_{>m} | f, d, d', a \supseteq d, a' \supseteq d', \text{proc}).$$

By definition, we are implicitly restricting the sum to those  $a$  and  $a'$  so that our summand is defined. This means that we actually only allow one value for each component in  $a \subset d$  (namely, the value that gives the next  $x$  element in  $d$ ) and similarly for  $a' \subset d'$ . Therefore the sum reduces to

$$\sum_{a \supseteq d, a' \supseteq d'} P(c_{>m} | f, d, d', a \supseteq d, a' \supseteq d', \text{proc}).$$

Note that no component of  $a \supseteq d$  lies in  $d \setminus d'$ . The same is true of  $a' \supseteq d'$ . So the sum over  $a \supseteq d$  is over the same components of  $a$  as the sum over  $a' \supseteq d'$  is of  $a'$ . Now for fixed  $d$  and  $d'$ ,  $\text{proc}$ 's choice of  $a$  or  $a'$  is fixed. Accordingly, without loss of generality, the sum can be rewritten as

$$\sum_{a \supseteq d} P(c_{>m} | f, d, d', a \supseteq d)$$

with the implicit assumption that  $c_{>m}$  is set by  $a \supseteq d$ . This sum is independent of  $\text{proc}$ .

## APPENDIX H PROOF OF THEOREM 11

Let  $\text{proc}$  refer to a choosing procedure. We are interested in

$$\begin{aligned} & \sum_{a, a'} P(c_{>m} | f, m, k, a, a', \text{proc}) \\ &= \sum_{a, a', d, d'} P(c_{>m} | f, d, d', k, a, a', \text{proc}) \\ & \quad \times P(d, d' | f, k, m, a, a', \text{proc}). \end{aligned}$$

The sum over  $d$  and  $d'$  can be moved outside the sum over  $a$  and  $a'$ . Consider any term in that sum (i.e., any particular pair of values of  $d$  and  $d'$ ). For that term,  $P(d, d' | f, k, m, a, a', \text{proc})$  is just one for those  $a$  and  $a'$  that result in  $d$  and  $d'$ , respectively, when run on  $f$ , and zero otherwise. (Recall the assumption that  $a$  and  $a'$  are deterministic.) This means that the  $P(d, d' | f, k, m, a, a', \text{proc})$  factor simply restricts our sum over  $a$  and  $a'$  to the  $a$  and  $a'$  considered in our theorem.

- [6] D. H. Wolpert and T. G. Macready, "No free lunch theorems for search," Santa Fe Institute, Santa Fe, NM, Tech. Rep. SFI-TR-95-010, 1995.
- [7] F. Glover, "Tabu search I," *ORSA J. Comput.*, vol. 1, pp. 190–206, 1989.
- [8] —, "Tabu search II," *ORSA J. Comput.*, vol. 2, pp. 4–32, 1990.
- [9] E. L. Lawler and D. E. Wood, "Branch and bound methods: A survey," *Oper. Res.*, vol. 14, pp. 699–719, 1966.
- [10] R. Kinderman and J. L. Snell, *Markov Random Fields and Their Applications*. Providence, RI: Amer. Math. Soc., 1980.
- [11] D. H. Wolpert, "The lack of a prior distinctions between learning algorithms," *Neural Computation*, vol. 8, pp. 1341–1390, 1996.
- [12] —, "On bias plus variance," *Neural Computation*, vol. 9, pp. 1271–1248, 1996.
- [13] D. Griffiths, "Introduction to random fields," in *Denumerable Markov Chains*, J. G. Kemeny, J. L. Snell, and A. W. Knapp, Eds. New York: Springer-Verlag, 1976.
- [14] C. E. M. Strauss, D. H. Wolpert, and D. R. Wolf, "Alpha, evidence, and the entropic prior," in *Maximum Entropy and Bayesian Methods*. Reading, MA: Addison-Wesley, 1992, pp. 113–120.
- [15] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

## ACKNOWLEDGMENT

The authors would like to thank R. Das, D. Fogel, T. Grossman, P. Helman, B. Levitan, U.-M. O'Reilly, and the reviewers for helpful comments and suggestions.



**David H. Wolpert** received degrees in physics from the University of California, Santa Barbara, and Princeton University, Princeton, NJ. He was formerly Director of Research at TXN Inc and a Postdoctoral Fellow at the Santa Fe Institute. He now heads up a data mining group at IBM Almaden Research Center, San Jose, CA. Most of his work centers around supervised learning, Bayesian analysis, and the thermodynamics of computation.



**William G. Macready** received the Ph.D. degree in physics at the University of Toronto, Ont., Canada. His doctoral work was on high-temperature superconductivity. He recently completed a postdoctoral fellowship at the Santa Fe Institute and is now at IBM's Almaden Research Center, San Jose, CA. His recent work focuses on probabilistic approaches to machine learning and optimization, critical phenomena in combinatorial optimization, and the design of efficient optimization algorithms.

# Teorema del “No free Lunch”

- Considerando lo que menciona el Teorema de No Free Lunch ¿si en AA existen múltiples algoritmos para resolver tareas de descriptivas y predictivas elegir cualquiera de ellos da lo mismo, si en promedio todos obtendrán el mismo resultado?

# Teorema del “No free Lunch”

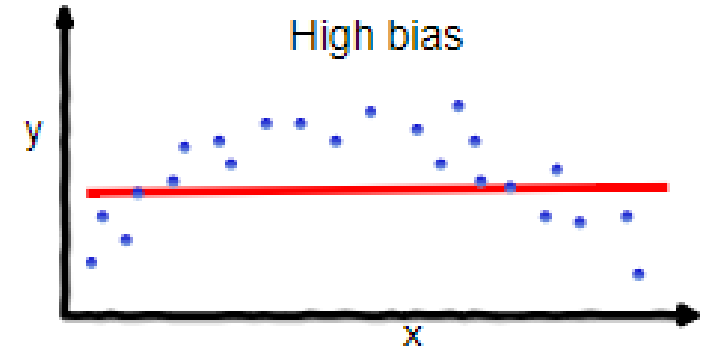
- Dependerá del problema que queramos abordar la selección del algoritmo. No es lo mismo diagnosticar una enfermedad o el clima del día de mañana

# Errores de predicción de un modelo de AA

- Error de sesgo (bias)
- Error de varianza (variance)
- Error irreducible. No se puede reducir, también se le conoce como ruido generalmente proviene de variables desconocidas, características incompletas o un problema mal enmarcado
- Los errores de sesgo y varianza se pueden reducir ya que se derivan de la elección del algoritmo

# Sesgo

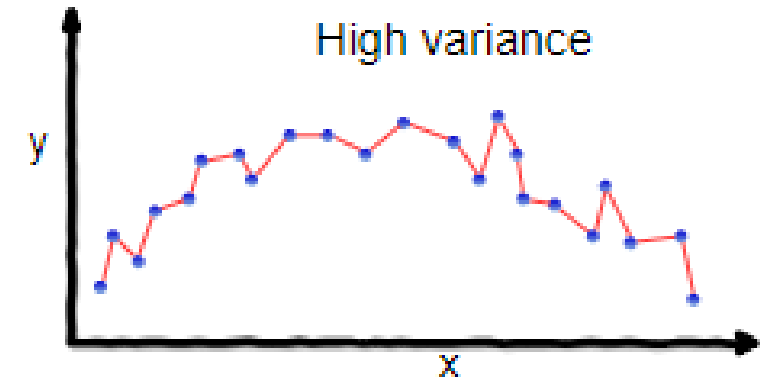
- Es la diferencia entre la predicción esperada de nuestro modelo y los valores verdaderos
- Un alto sesgo representa un problema de subajuste (**underfitting**)
- Algoritmos con bajo sesgo
  - Árboles de decisión
  - K-vecinos más cercanos (KNN)
  - Maquinas de soporte vectorial (SVM)
- Algoritmos con alto sesgo
  - Regresión lineal
  - Análisis discriminante lineal
  - Regresión logística



**underfitting**

# Varianza

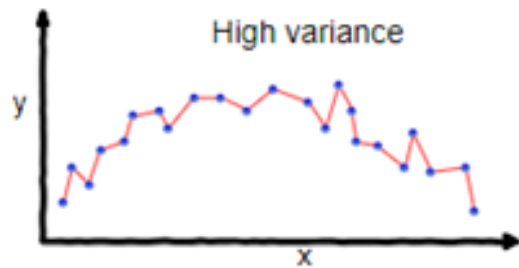
- Se refiere a cuanto varía la predicción según los datos que utilizemos para el entrenamiento
- Un modelo con varianza baja indica que cambiar los datos de entrenamiento produce cambios pequeños en la estimación
- Al contrario, un modelo con varianza alta quiere decir que pequeños cambios en el dataset conlleva a grandes cambios en la salida(**overfitting**)



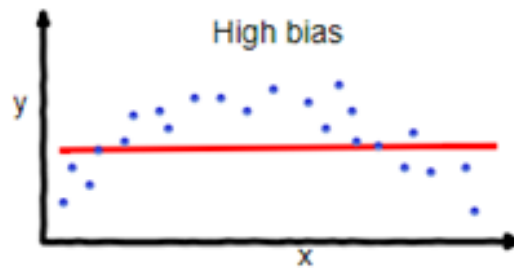
**overfitting**

# Sesgo y varianza

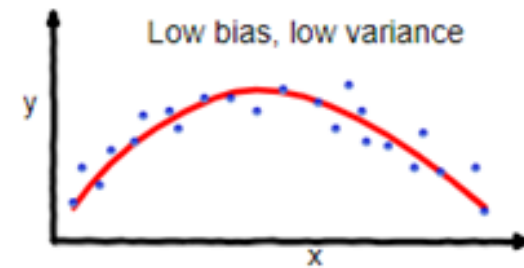
- Un modelo robusto tendrá poco sesgo y poca varianza
  - Disminuir la varianza implica aumentar el sesgo
  - Disminuir el sesgo hace que la varianza aumente
- Un modelo predictivo bueno será el que tenga un buen balance entre sesgo y varianza de manera que se minimice el error total



overfitting



underfitting



Good balance



# Modos de Aprendizaje Automático

- Estático o en lote
- Dinámico

# Aprendizaje estático o en lote

- Es como si tomáramos una foto de los datos
- Las propiedades de los datos permanecen constantes a lo largo del tiempo
- Un ejemplo de este aprendizaje es la clasificación de imágenes de diferentes animales



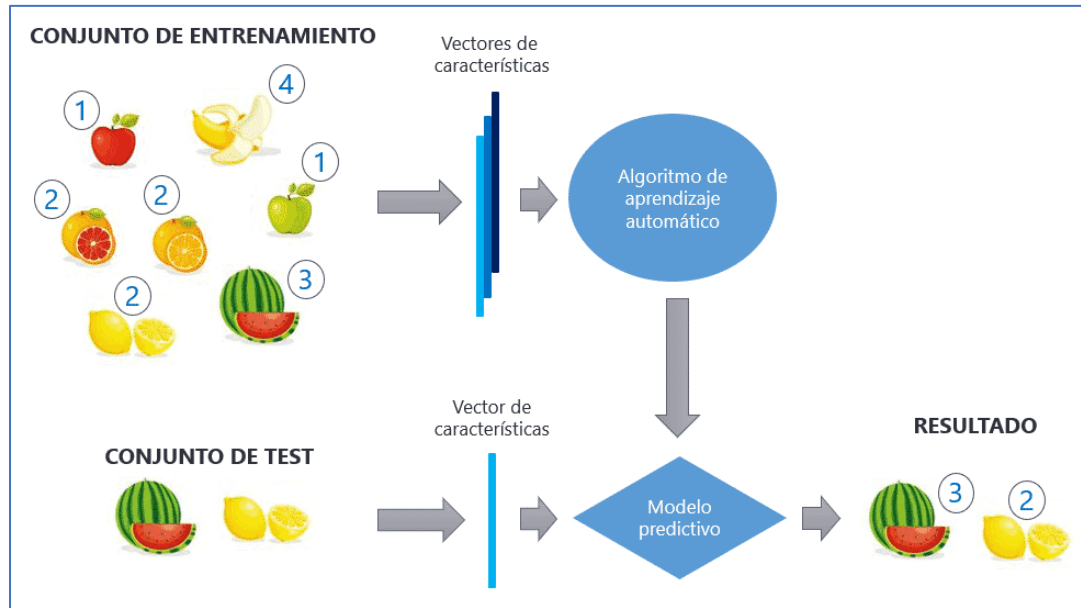
# Aprendizaje dinámico o en línea

- Aprendizaje basado en series temporales
- Los datos son sensibles al tiempo y cambian constantemente
- El modelo debe entrenarse continuamente (o después de cada ventana de tiempo razonable) para que siga siendo eficaz
- Un ejemplo típico de este tipo de problemas es la previsión meteorológica o las predicciones bursátiles

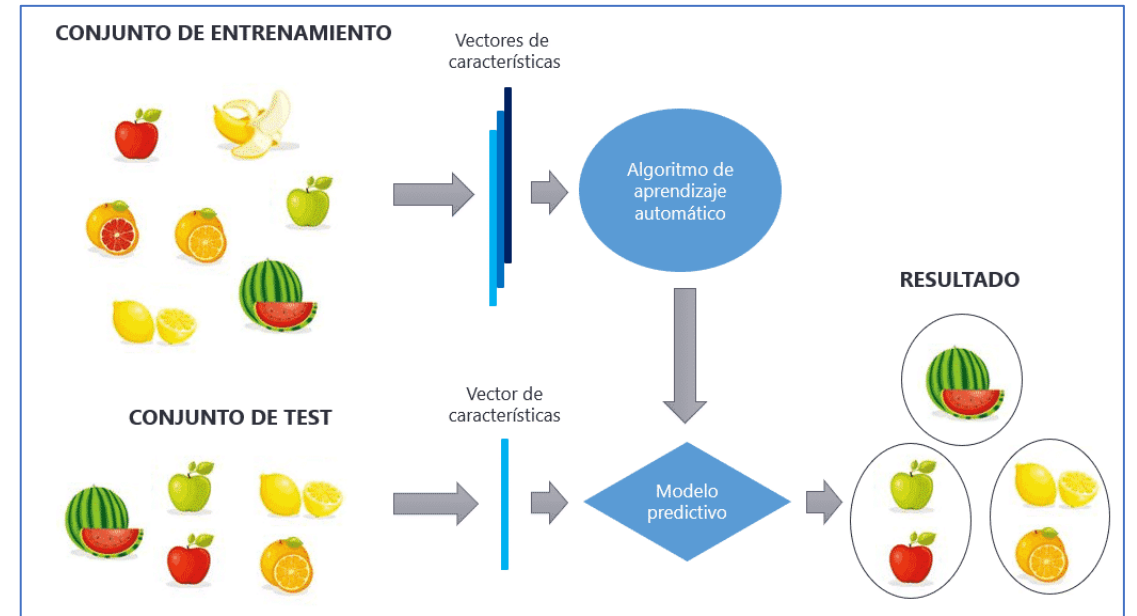


# Tipos de aprendizaje

## Aprendizaje supervisado



## Aprendizaje no supervisado



# Conjunto de datos (datasets)

- Conjunto de datos de análisis y predicción de ataques cardíacos
  - <https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset>
- Imágenes de cultivos de trigo, arroz , caña de azúcar, yute, maíz.
  - <https://www.kaggle.com/datasets/aman2000jaiswal/agriculture-crop-images>
- Conjunto de datos de noticias falsas y verdaderas
  - <https://www.kaggle.com/datasets/clmentbisailon/fake-and-real-news-dataset>

# Entrenar el modelo

Conjunto de entrenamiento



Conjunto de prueba



# Conjuntos de entrenamiento y prueba

- Instancias
  - Suficientes
  - Diversas

Generalización

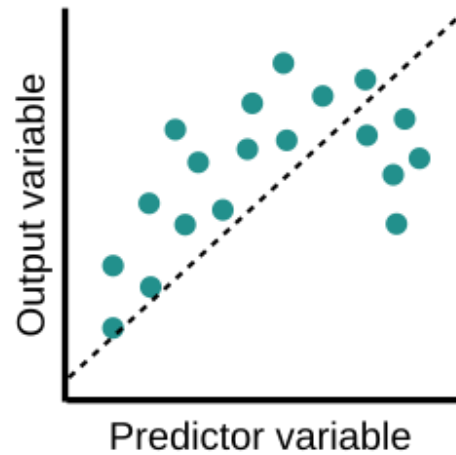
Asegurarnos de que el modelo tiene un buen rendimiento

# Conjunto de entrenamiento

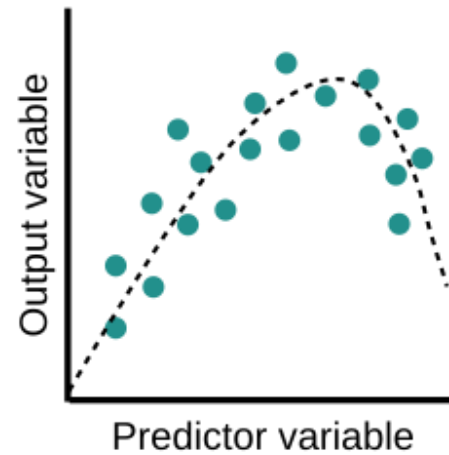
Pocas instancias  
(subajuste)



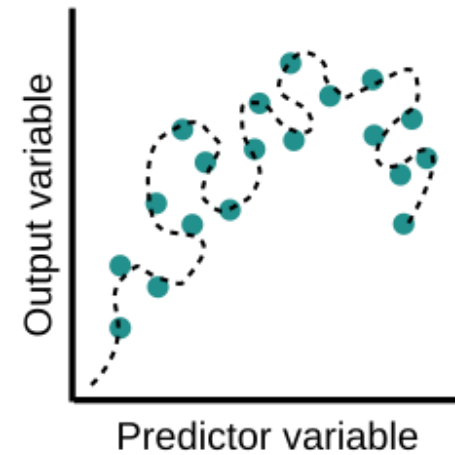
Underfit



Optimal



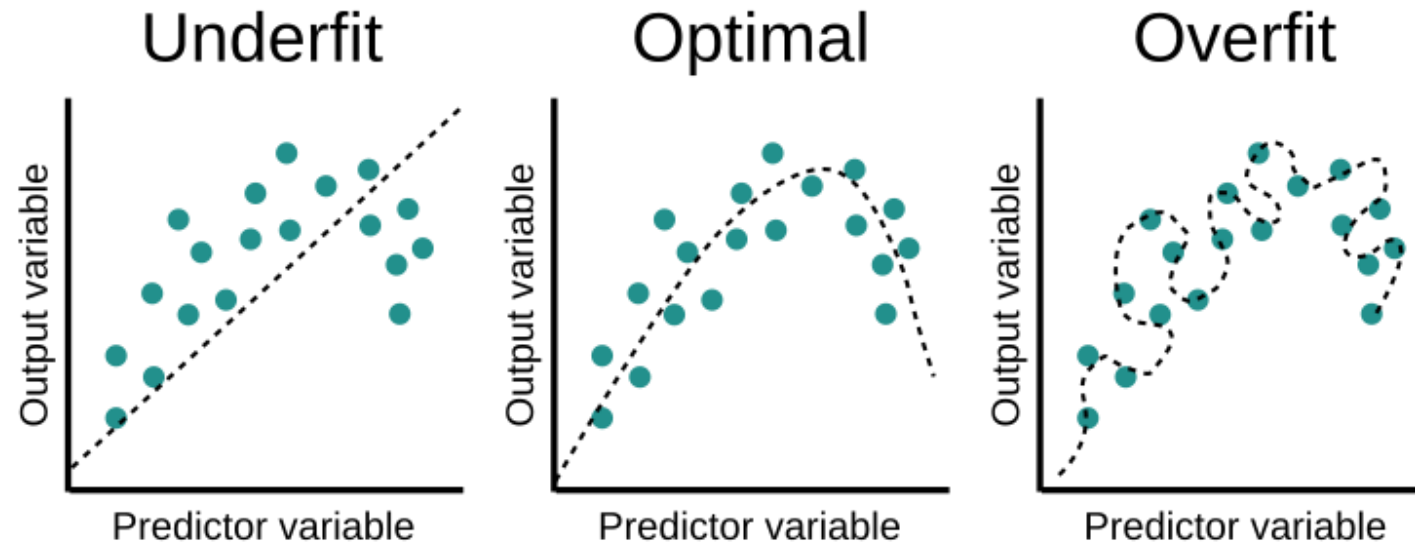
Overfit





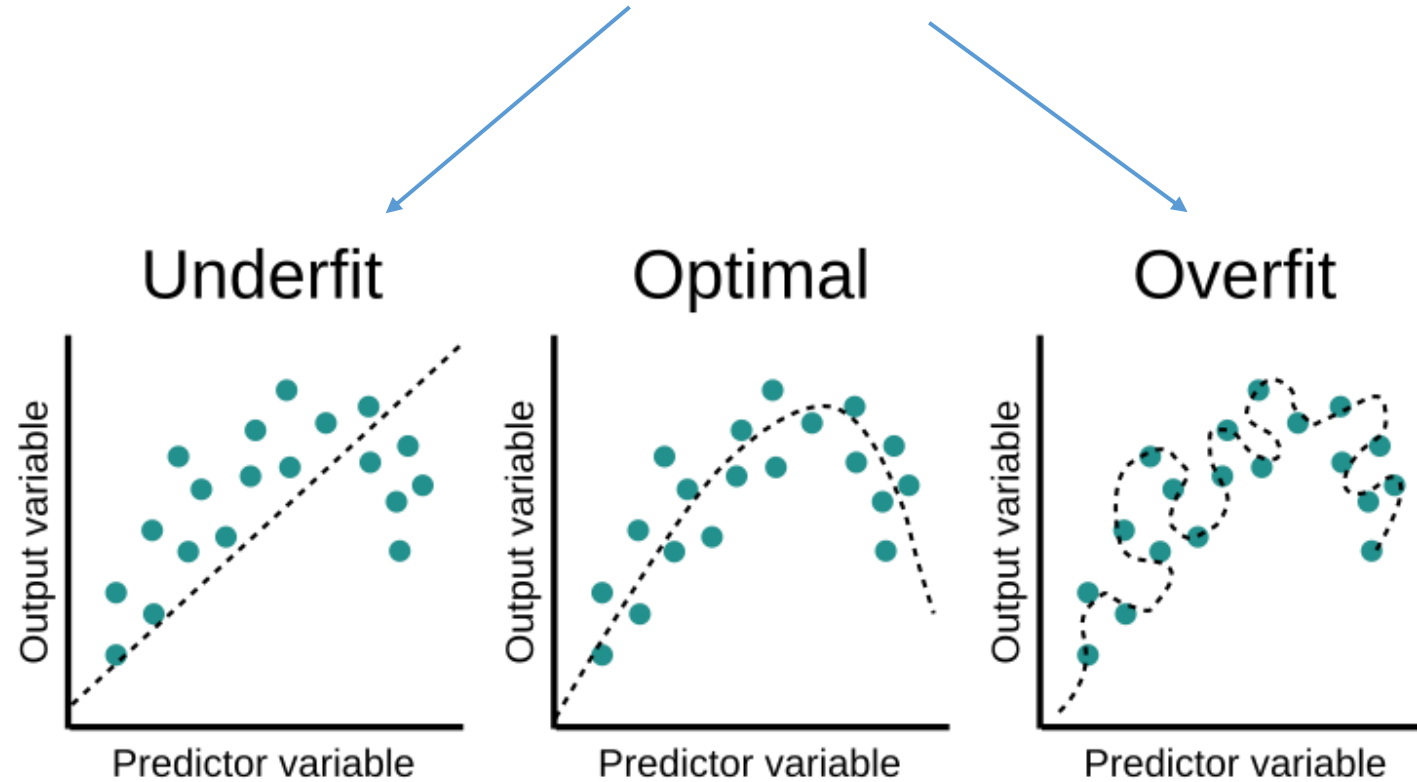
# Conjunto de entrenamiento

Demasiadas instancias  
(sobreajuste)



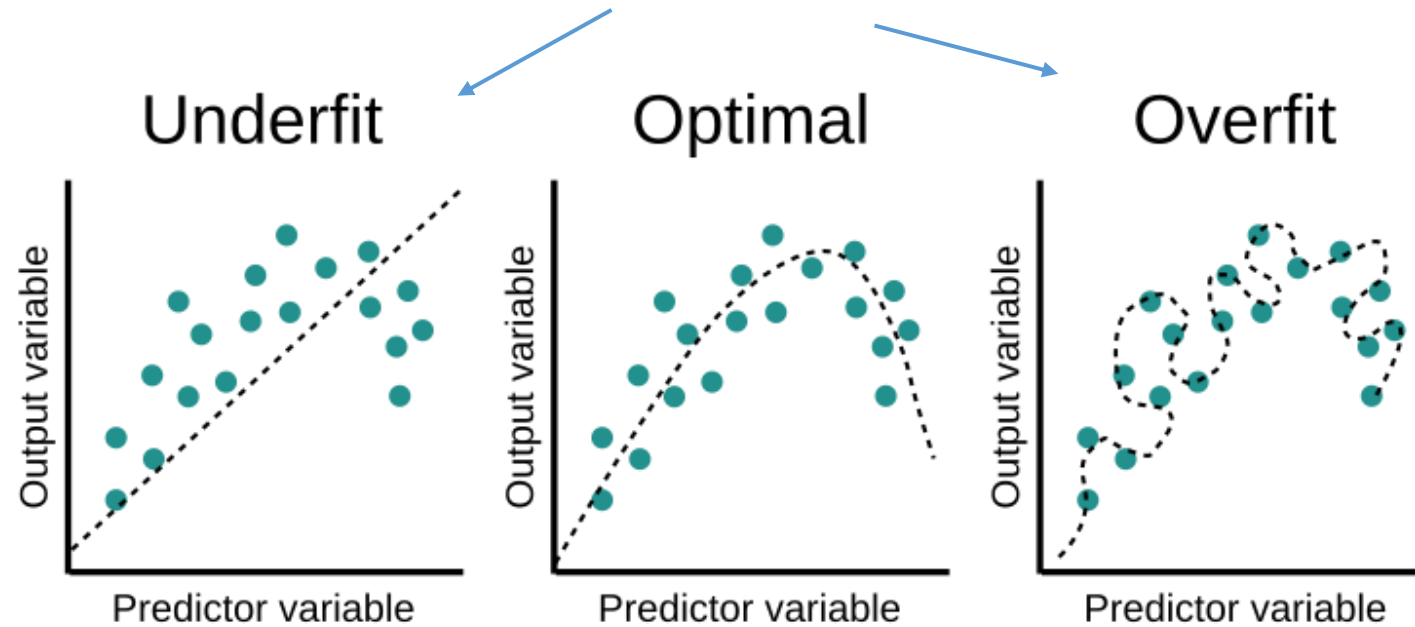
# Conjunto de entrenamiento

Poca diversidad



# Conjunto de entrenamiento

Dataset desbalanceado



- ¿Cuántas instancias son suficientes?
- ¿Cómo verifico que un dataset es diverso?
- ¿Cómo sé que un dataset esta balanceado o desbalanceado?
- Dado un dataset ¿cuántas instancias de entrenamiento y cuántas de prueba?

# Evitar underfit y overfit

- Revolver las instancias del dataset (shufle)



- Instancias suficientes y diversas
- Utilizar algún método para validar el modelo

# Métodos de validación

- Validación cruzada de k pliegues (*k-fold* cross-validation)
- Dejar uno fuera (Leave-one-out-cross-validation, LOOVC)
- Bootstrap sampling

# Validación cruzada

# Validación cruzada de $k$ pliegues


- Esta técnica separa el conjunto de entrenamiento en  $k$  pliegues distintos
- Se debe escoger el número de pliegues de antemano y en cada pliegue habrá un conjunto distinto de datos de entrenamiento y de prueba
- Este proceso es iterativo y termina cuando se hayan recorrido todos los pliegues establecidos
- Aunque no existe un consenso los valores recomendados para  $k$  son 10, 5 y 3



## Dataset de 20 pacientes

- Renglón 1: id paciente
- Renglón 2: síntoma
- Renglón 3: diagnostico
  - 0 no tiene la enfermedad
  - 1 si tiene la enfermedad

Entrenamiento 60%




13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1



Pliegues/ Conjuntos de validación

Prueba 40%



16	12	8	5	17	10	14	19
I	T	L	Y		R	I	K
0	0	0	1	0	1	1	0

# Validación cruzada $k = 2$

$k = 1$

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1

$k = 2$

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1

	Conjunto de prueba $= 1/2 \times 12$
	Conjunto de entrenamiento $= 1/2 \times 12$

# Validación cruzada $k = 3$

$k = 1$

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1

$k = 2$

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1

$k = 3$

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1

	Conjunto de prueba = $1/3 \times 12$
	Conjunto de entrenamiento = $2/3 \times 12$

# Validación cruzada $k = 5$

$k$	13	2	7	1	20	18	3	15	9	4	11	6
1	I	T	L	Y	P	R	I	K	Y	R	I	J
2	I	T	L	Y	P	R	I	K	Y	R	I	J
3	I	T	L	Y	P	R	I	K	Y	R	I	J
4	I	T	L	Y	P	R	I	K	Y	R	I	J
5	I	T	L	Y	P	R	I	K	Y	R	I	J

$k$	13	2	7	1	20	18	3	15	9	4	11	6
1	I	T	L	Y	P	R	I	K	Y	R	I	J
2	I	T	L	Y	P	R	I	K	Y	R	I	J
3	I	T	L	Y	P	R	I	K	Y	R	I	J
4	I	T	L	Y	P	R	I	K	Y	R	I	J
5	I	T	L	Y	P	R	I	K	Y	R	I	J



	Conjunto de prueba = $\lfloor 1/5 \rfloor \times 12 = \lfloor 2.4 \rfloor = 2$
	Conjunto de entrenamiento = $\lfloor 4/5 \rfloor \times 12 = \lfloor 9.6 \rfloor = 10$

	Conjunto de prueba = $\lceil 1/5 \rceil \times 12 = \lceil 2.4 \rceil = 3$
	Conjunto de entrenamiento = $\lfloor 4/5 \rfloor \times 12 = \lfloor 9.6 \rfloor = 9$

# Dejar uno fuera Leave-one-out-cross-validation (LOOVC)

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J
I	T	L	Y	P	R	I	K	Y	R	I	J

Dejar uno fuera = Validación cruzada  $k = n$


$k$	13	2	7	1	20	18	3	15	9	4	11	6
1	I	T	L	Y	P	R	I	K	Y	R	I	J
2	I	T	L	Y	P	R	I	K	Y	R	I	J
3	I	T	L	Y	P	R	I	K	Y	R	I	J
4	I	T	L	Y	P	R	I	K	Y	R	I	J
5	I	T	L	Y	P	R	I	K	Y	R	I	J
6	I	T	L	Y	P	R	I	K	Y	R	I	J
7	I	T	L	Y	P	R	I	K	Y	R	I	J
8	I	T	L	Y	P	R	I	K	Y	R	I	J
9	I	T	L	Y	P	R	I	K	Y	R	I	J
10	I	T	L	Y	P	R	I	K	Y	R	I	J
11	I	T	L	Y	P	R	I	K	Y	R	I	J
12	I	T	L	Y	P	R	I	K	Y	R	I	J

# Bootstrap sampling

## Dataset de 20 pacientes

- Renglón 1: id paciente
- Renglón 2: síntoma
- Renglón 3: diagnostico
  - 0 no tiene la enfermedad
  - 1 si tiene la enfermedad

Entrenamiento 60%




13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1



Conjuntos de validación

Prueba 40%



16	12	8	5	17	10	14	19
I	T	L	Y		R	I	K
0	0	0	1	0	1	1	0



# Bootstrap sampling

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1

3 conjuntos de entrenamiento  $t = 12$

13	13	13	1	15	2	20	9	2	2	4	1
I	I	I	Y	K	T		Y	T	T	R	Y
0	0	0	0	1	1	1	0	1	1	0	0

13	2	7	1	20	18	3	15	9	4	11	6
I	T	L	Y		R	I	K	Y	R	I	J
0	1	1	0	1	1	0	1	0	0	0	1

18	7	1	6	11	2	4	15	9	1	15	1
R	L	Y	J	I	T	R	K	Y	Y	K	Y
1	1	0	1	0	1	0	1	0	0	1	0

Conjuntos de prueba

7	18	3	11	6
L	R	I	I	J
1	1	0	0	1

Conjunto vacío

13	20	3
I		I
0	1	0