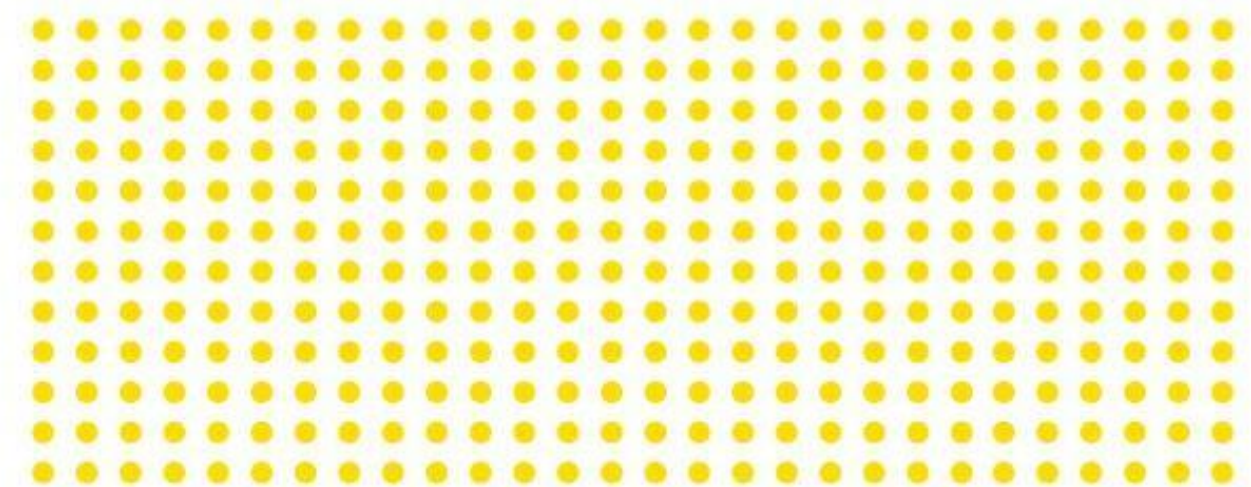


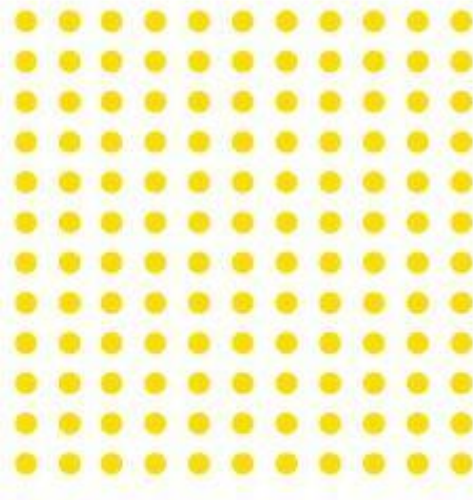


Universidad de
los Andes

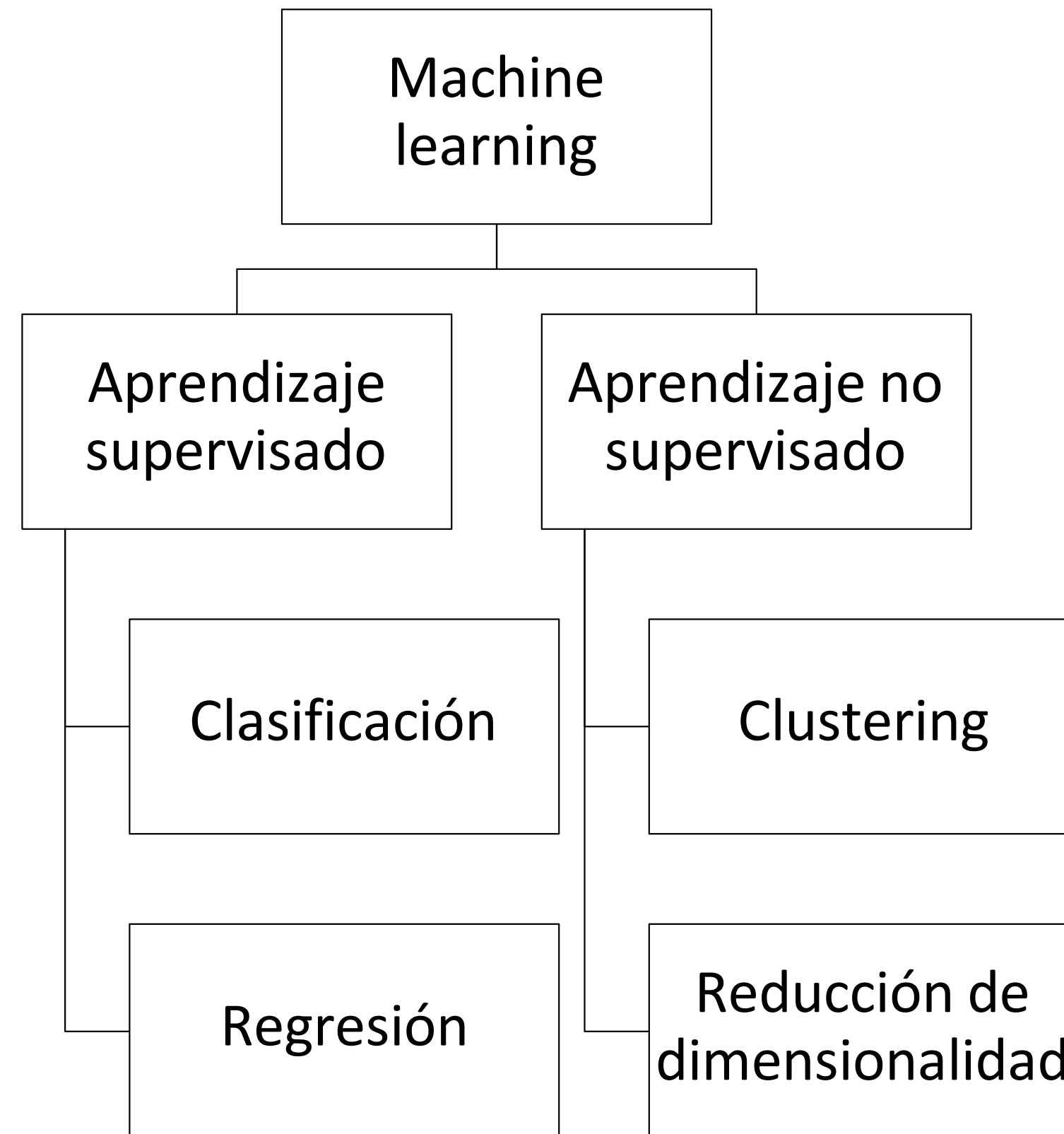
Educación
Continua
Vicerrectoría Académica



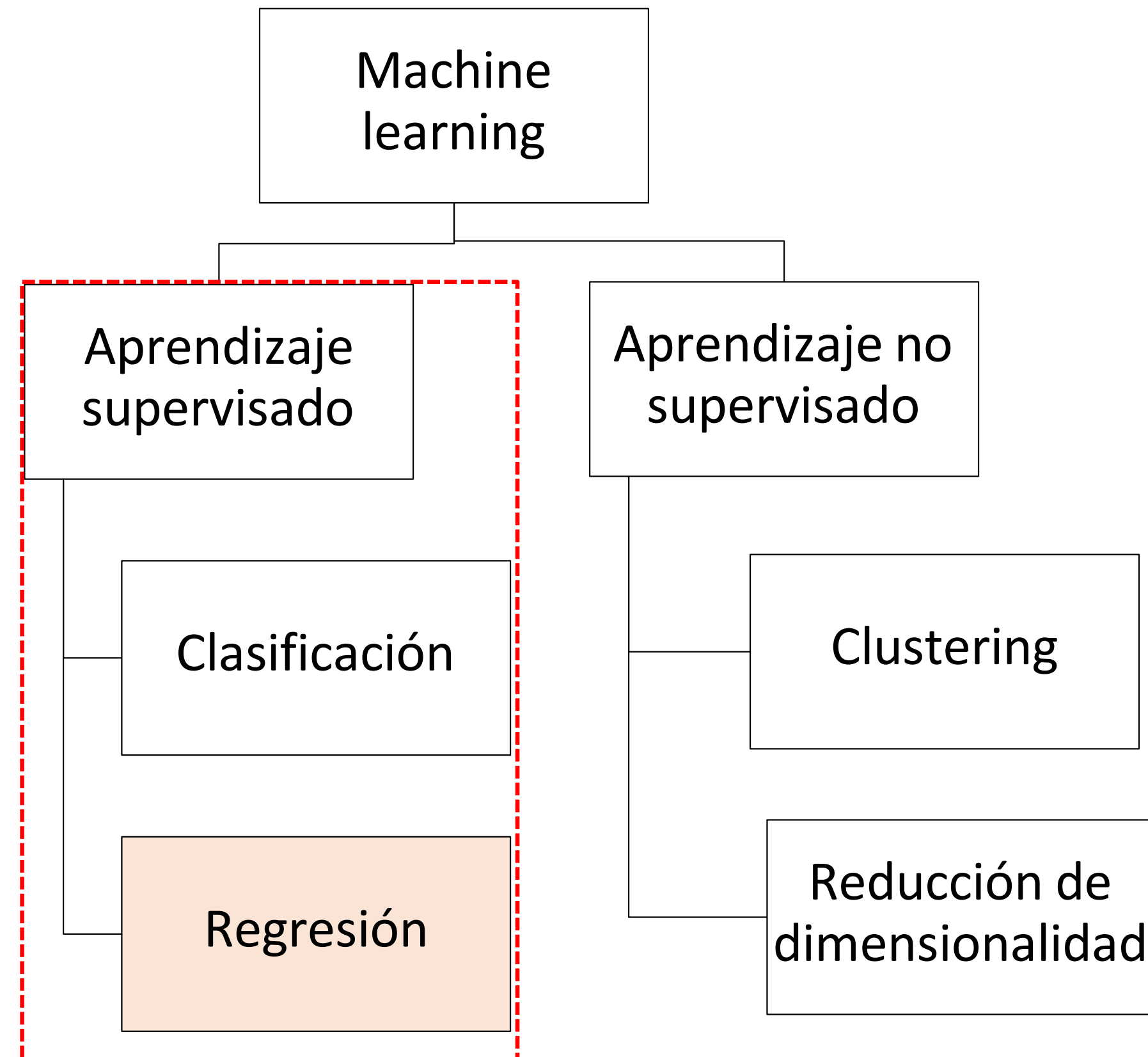
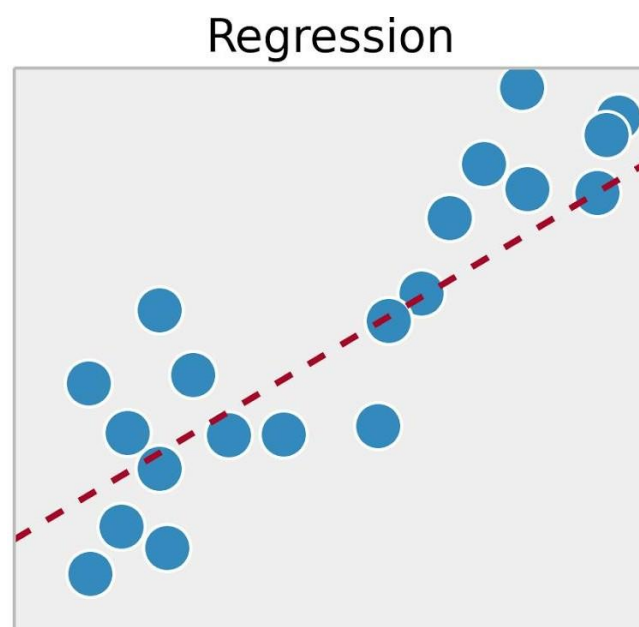
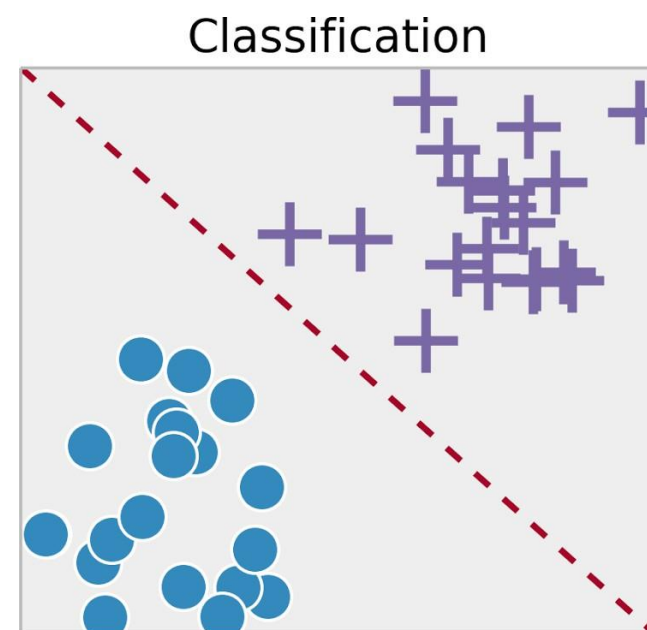
Contenido

- 
- Clasificación
 - K-NN (K-vecinos más cercanos)
 - Intro a fine-tuning
 - Support Vector Machines
 - Métricas de evaluación en clasificación
 - GridSearchCV y RandomSearchCV

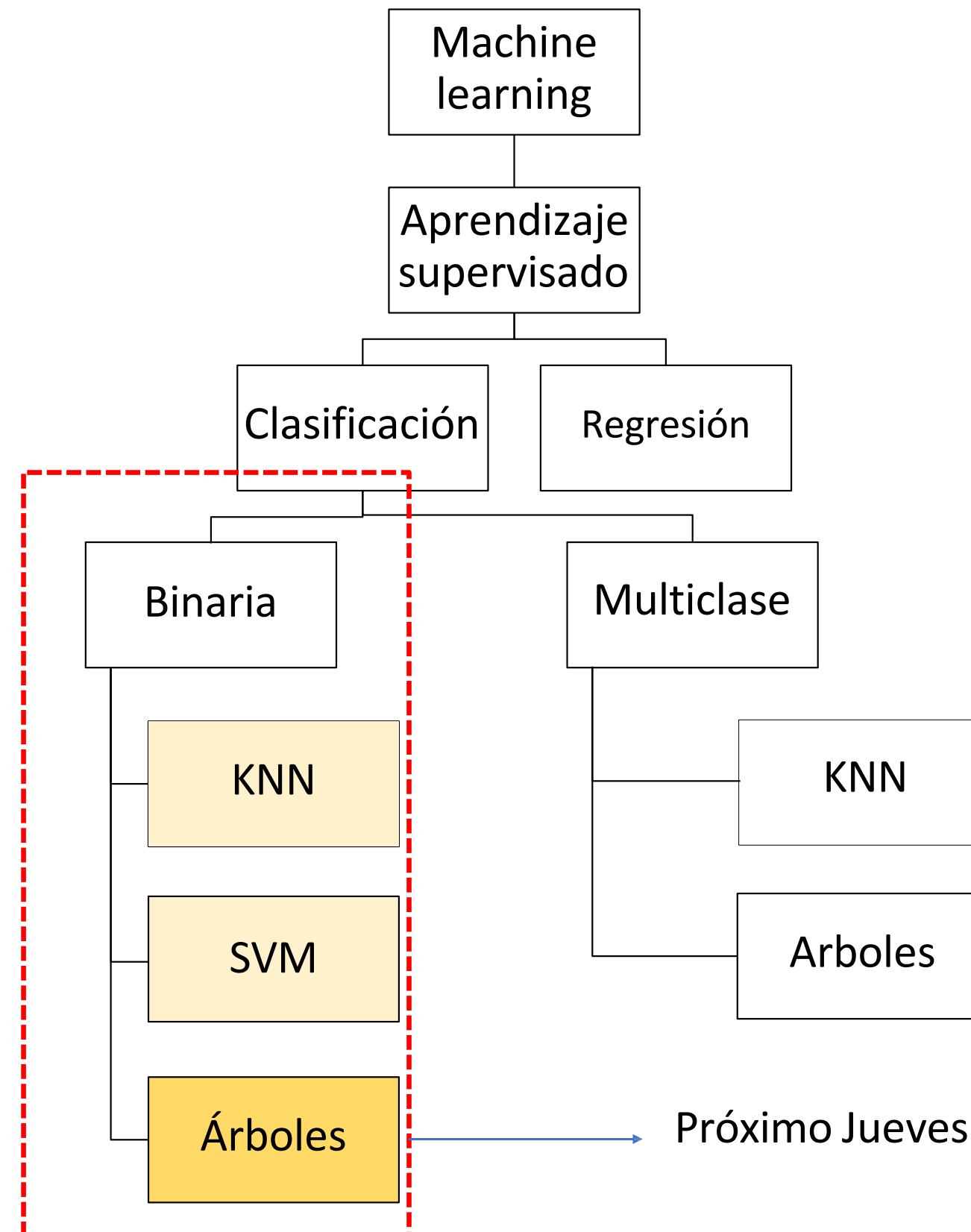
Machine Learning y clasificación

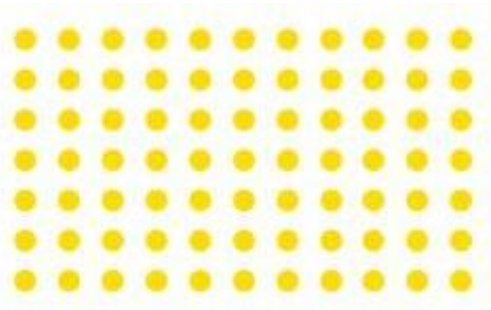


Machine Learning y clasificación



Machine Learning y clasificación

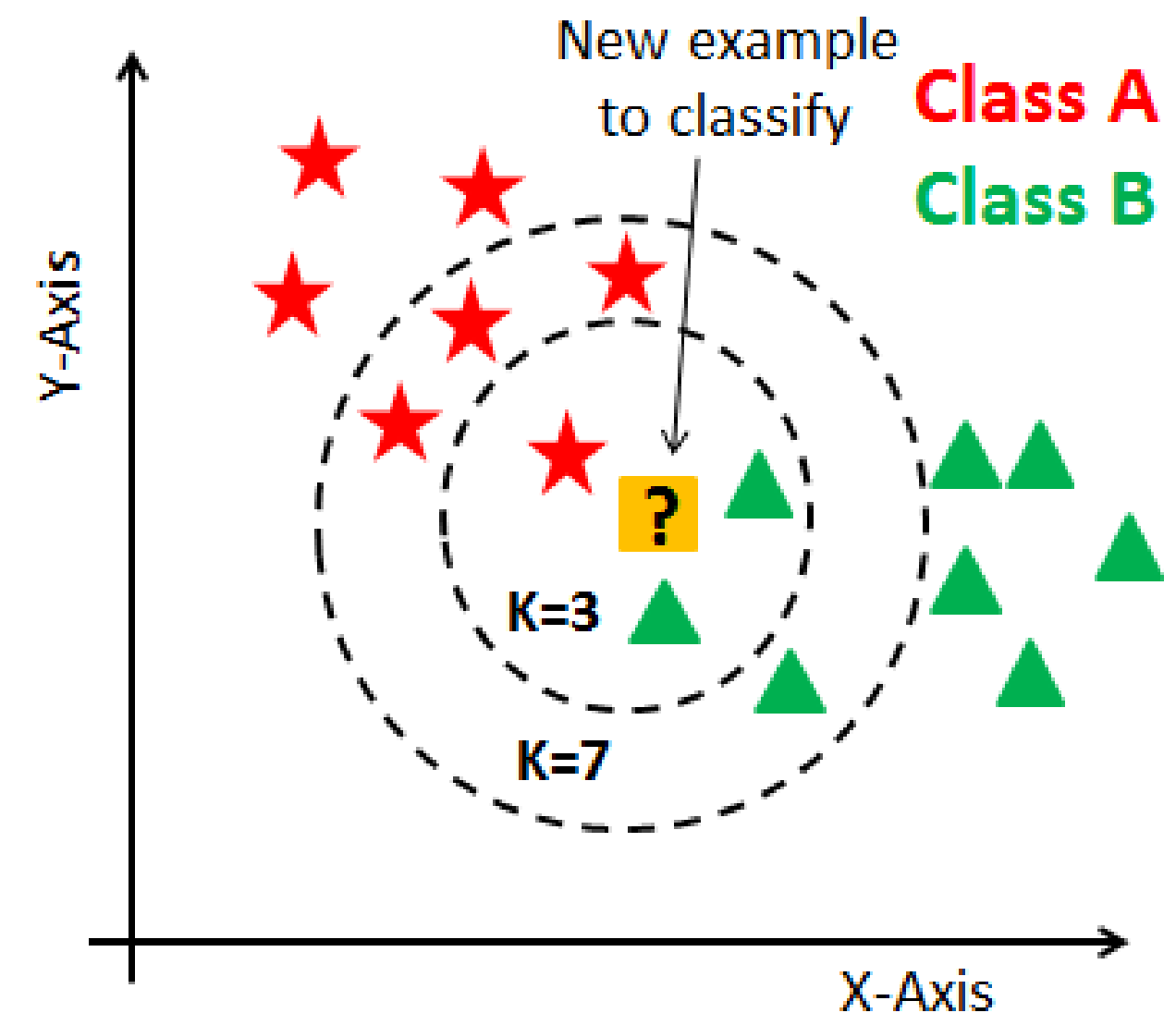




KNN- K Vecinos más Cercanos

K-NN: Vecinos más cercanos (Dime con quien andas...)

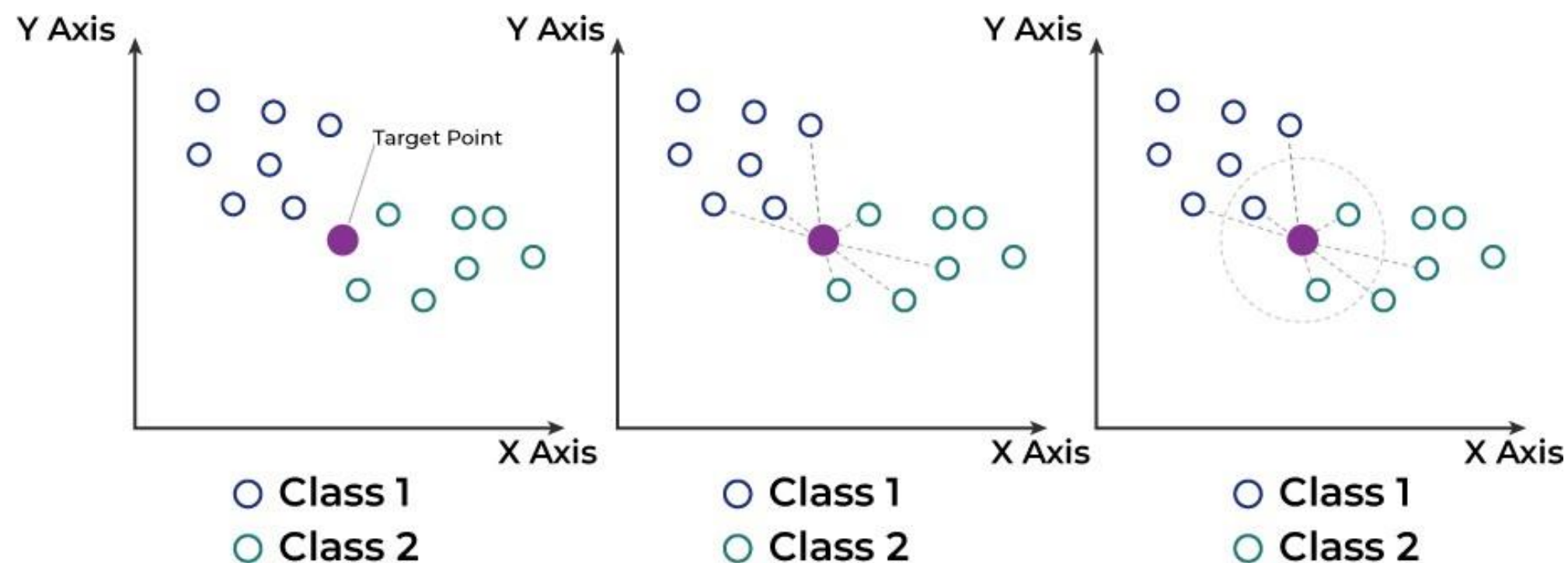
- Algoritmo de clasificación más sencillo (51')
- Supone que las variables de explicación son numéricas y hay una variable de clasificación
- Como ejemplo (BI), queremos predecir si alguien compra nuestro producto teniendo en cuenta variables socio-demográficas.
 - En el ejemplo más sencillo supongamos que depende de la edad y del ingreso del individuo
- El algoritmo predecirá teniendo en cuenta la etiqueta (si compró o no) de sus vecinos



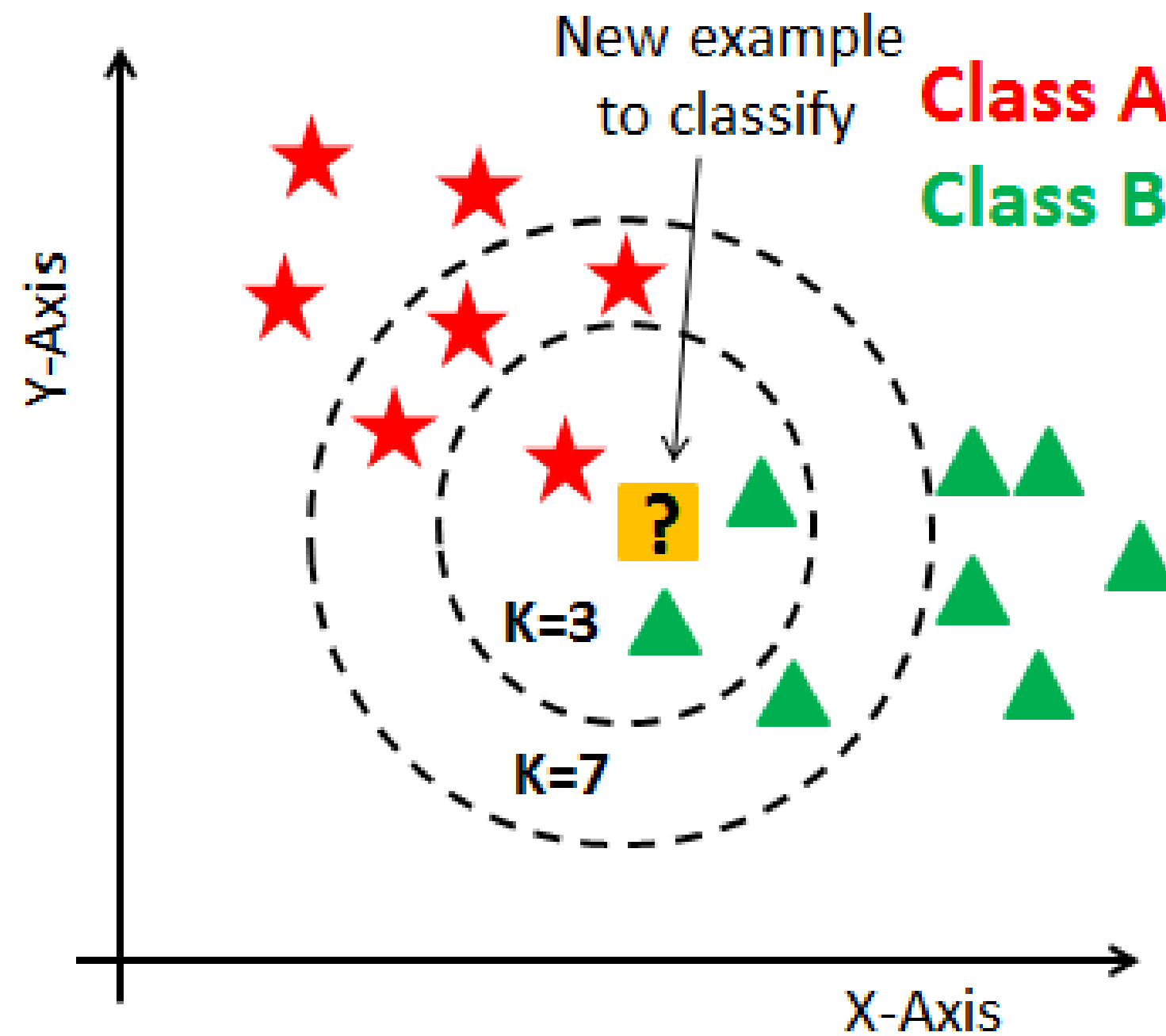
K-NN: Vecinos más cercanos

Algoritmo:

1. Fijando K (número de vecinos) encuentre la distancia del punto a predecir contra todos los puntos en la nube de datos.
2. Encuentre quiénes son los K-vecinos más cercanos por distancia.
3. La etiqueta predicha será aquella que tenga la mayoría de los vecinos.

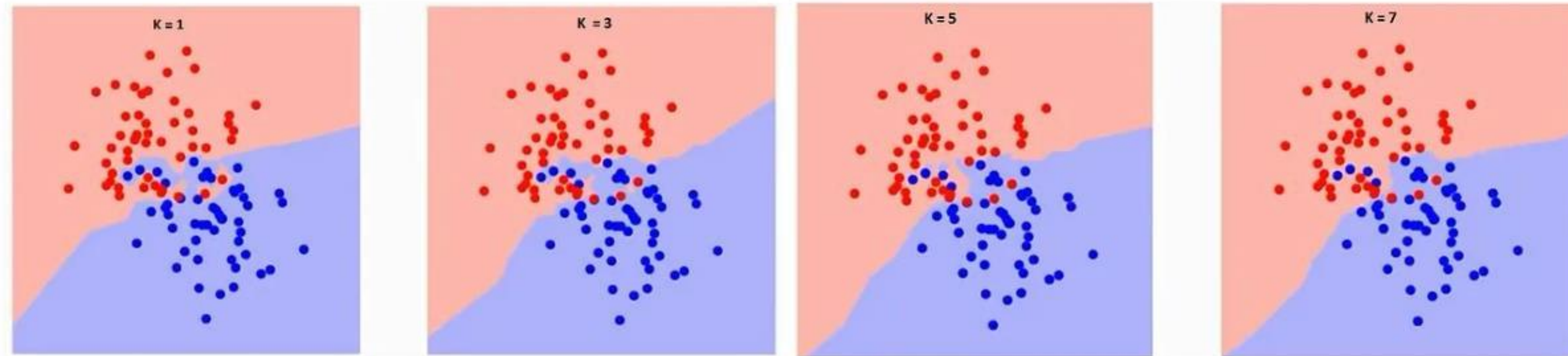


K-NN: Vecinos más cercanos



K-NN: Fine-tuning -> ¿Cómo escogemos K?

K es un *hiperparámetro* (Parámetro definido fuera del algoritmo).



- A medida que K aumenta, La region que "parte" el espacio se Vuelve más suave. ($k \mapsto \infty$ todos serán rojos o azules). Recuerden el trade-off de varianza y sesgo.
- Escoger K dependerá de un problema *ad-hoc* (Lo veremos después del break)
 - Una primera aproximación "ingenua" podría ser aquel K que minimiza la proporción de etiquetas incorrectas en test.



K-NN – Discusión



Ventajas

1. Fácil de implementar
2. Es un algoritmo de clasificación no – paramétrica
 1. No le exigimos nada a los datos
3. Pocos hiperparámetros (Hay que optimizar K y la distancia)

Desventajas

1. No es escalable -> Es exhaustivo en consumo de memoria y tiempo
2. **Maldición de la dimensionalidad** -> Ante dimensiones de covariables muy altas la efectividad del algoritmo cae
3. Hay tendencia al *overfitting*

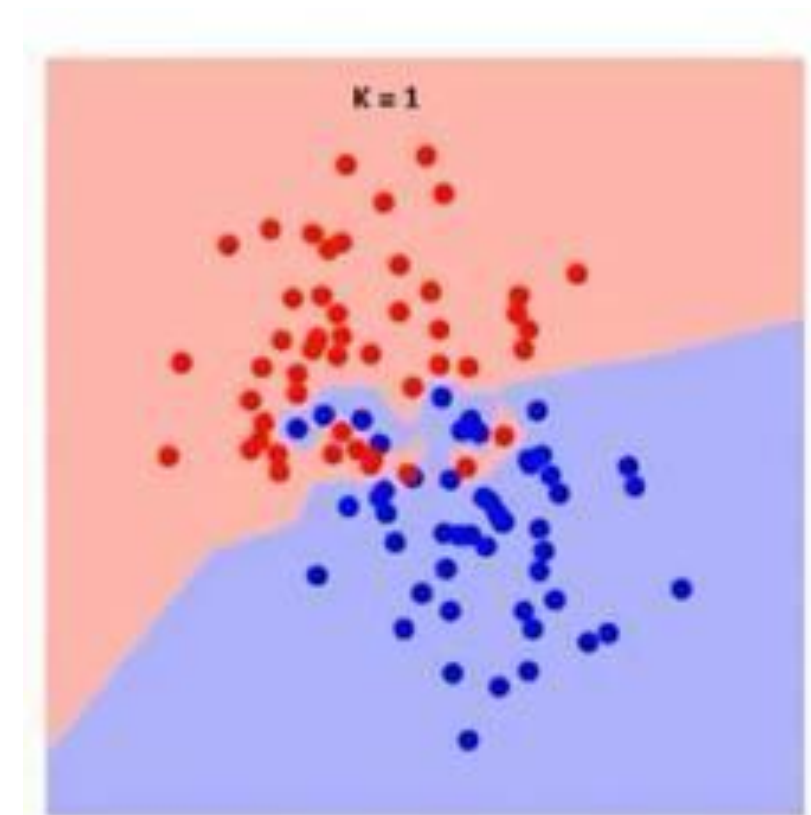
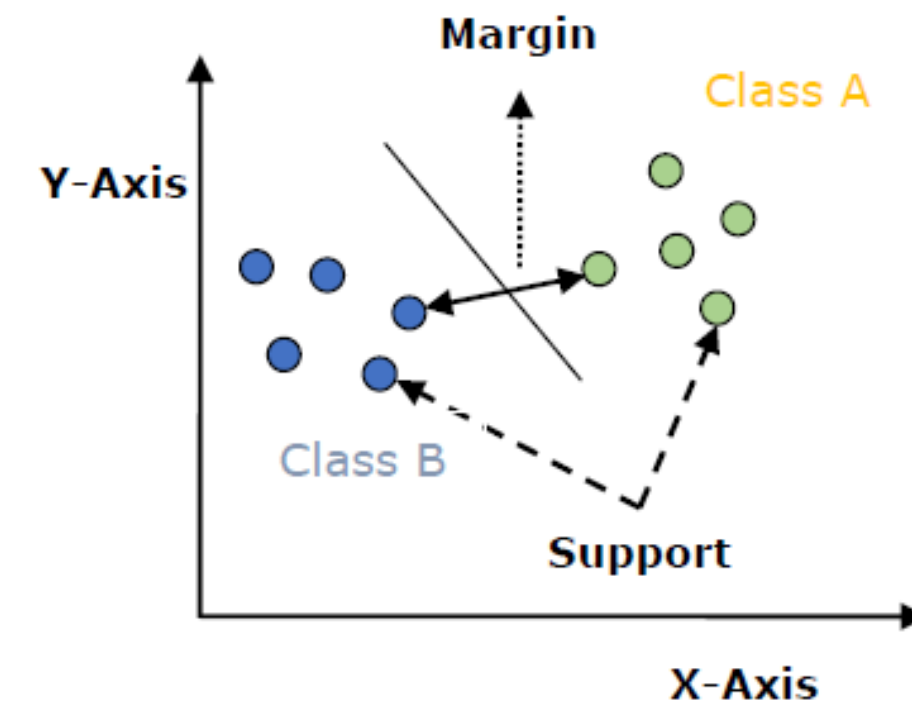


(SVM) Máquinas de Soporte Vectorial

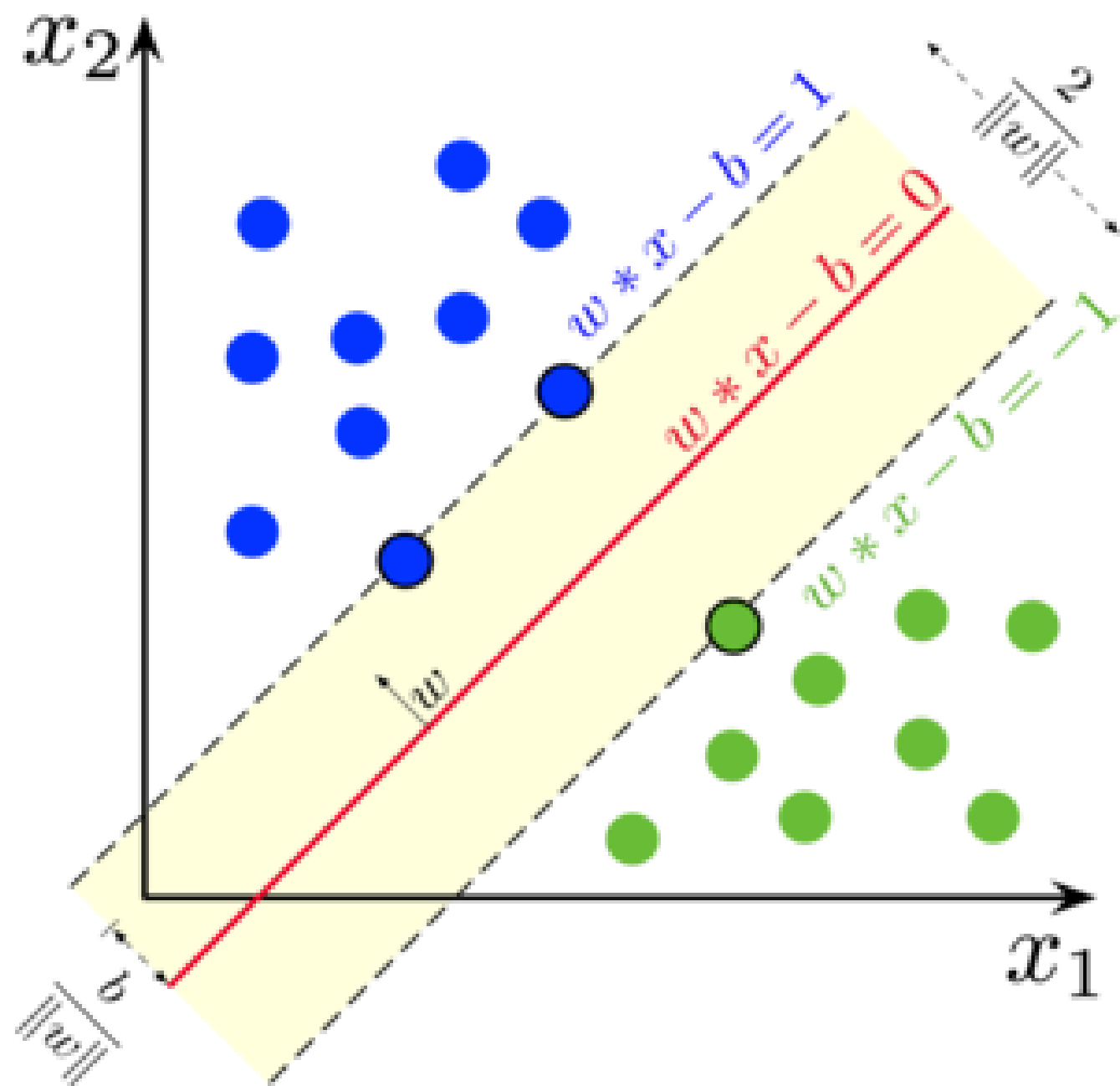
SVM – Máquinas de Soporte Vectorial

Otro de los primeros Algoritmos de clasificación
(74'/82')

- Acá queremos encontrar una híper-superficie separadora que me permita distinguir entre las etiquetas.
- El algoritmo se basa en el "soporte vectorial" (Aquella puntos en la frontera) para definir esta superficie separadora.



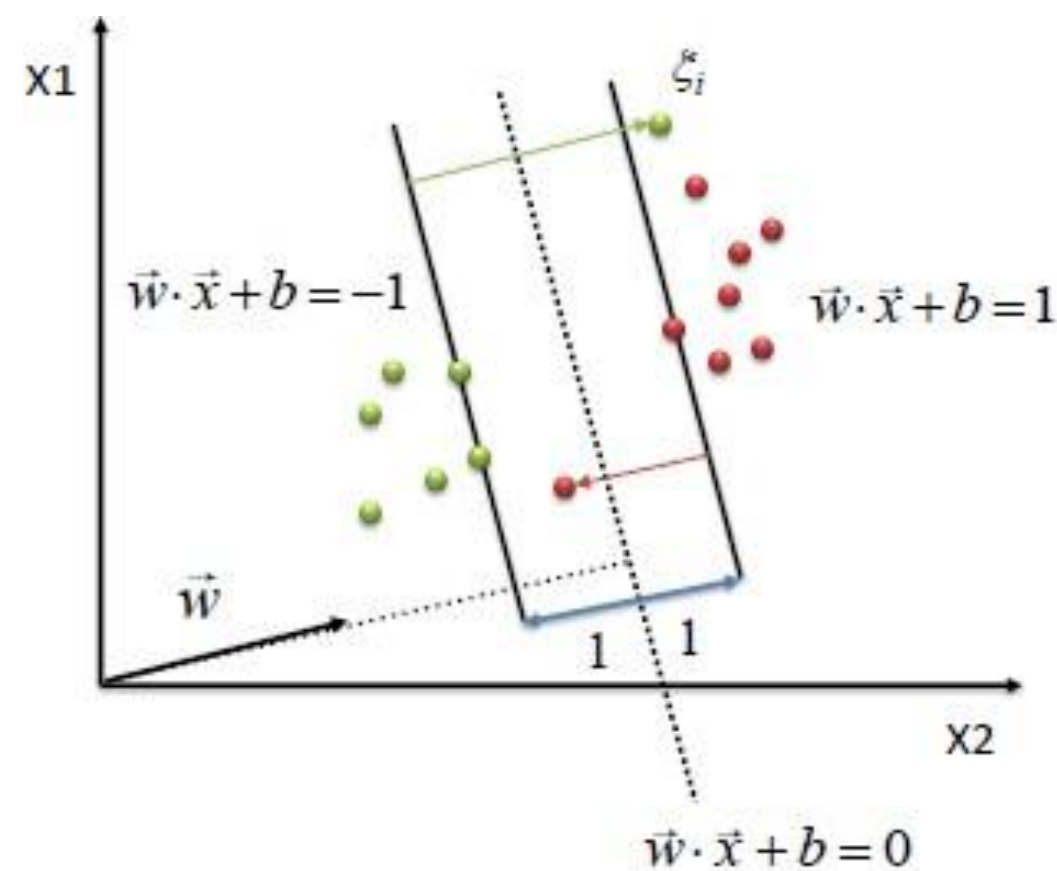
SVM – Intuición caso separable



Supongamos que los datos pueden ser linealmente separables

- El algoritmo funciona así:
 - Encuentra 2 hiper planos paralelos que separan las dos clases, tales que la distancia entre ellos es máxima.
 - Escoge el hiperplano de la mitad como aquel que maximiza el "margen".

SVM – ¿Y si los datos no son separables?



Constraint becomes :

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall x_i$$
$$\xi_i \geq 0$$

Objective function
penalizes for misclassified
instances and those within
the margin

$$\min \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

C trades-off margin width
and misclassifications

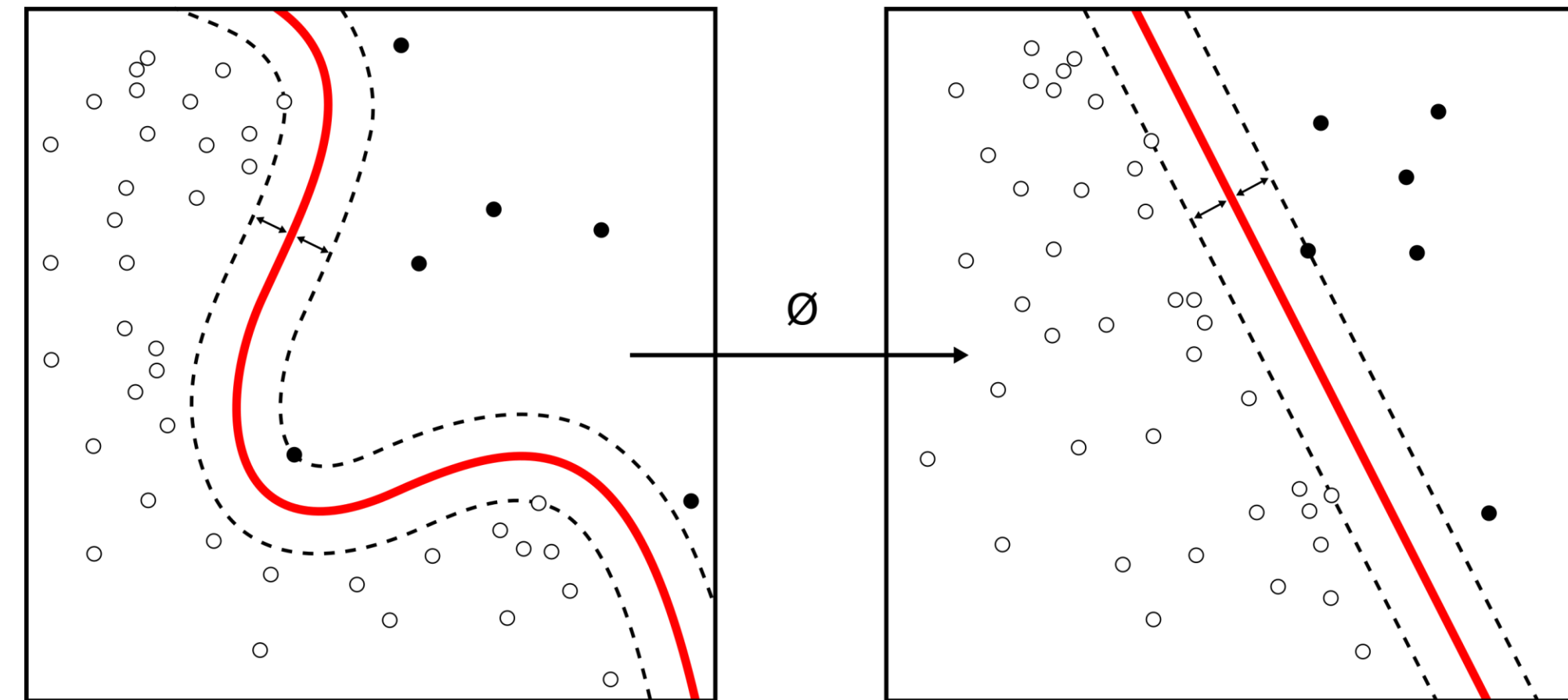
Supongamos que algunos datos están cruzados.

En este caso permitimos que algunos datos se "salgan" del margen (los medimos con gamma) pero los penalizamos con C

SVM – Y En un caso más extremo?

Si la region de separación no es lineal en absoluto, usamos las funciones de kernel:

- Estas hacen una transformación del espacio para que los datos se vuelvan "tan" separables como sea possible.
 - Polinomial
 - Gaussiana
 - Sigmoidal
 - Arcotangente inversa





SVM – Discusión

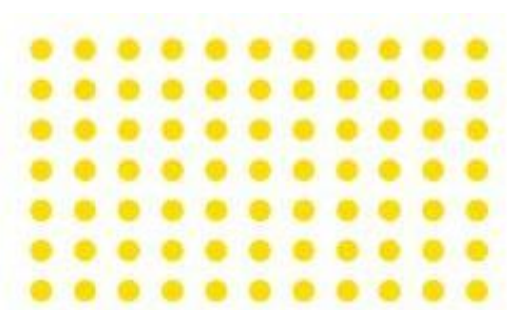


Ventajas

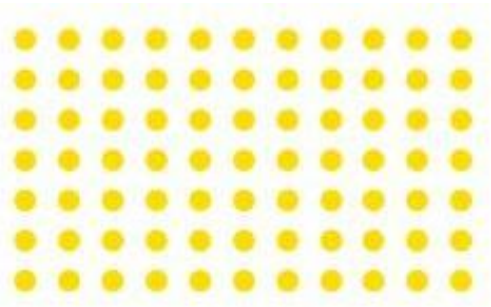
1. Efectivo en espacios de altas dimensiones
2. Usa pocos datos (el soporte vectorial) para definir la hiper-superficie separadora

Desventajas

1. Es *mu*y sensible a la escogencia del kernel
2. Es sensible a la escala de los datos
3. La dependencia del soporte vectorial puede jugar en contra si estos están mal codificados



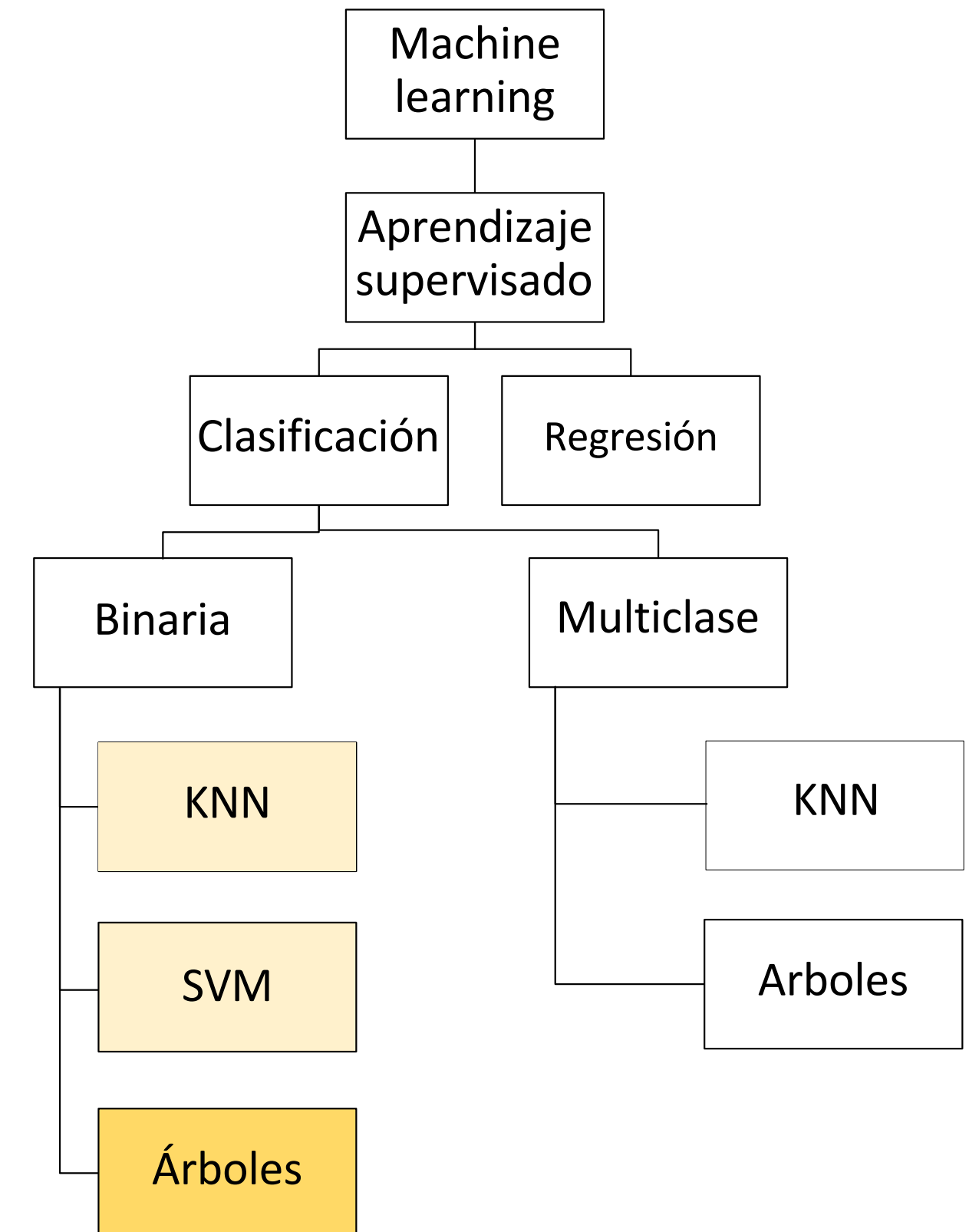
Break!



Métricas de evaluación en clasificación

Métricas de evaluación

- Para cualquier algoritmo que implementemos, queremos ver qué tan "bueno" es.
- Esta noción de bueno es relativa a los requerimientos de los algoritmos y la calidad de los datos.
 - Balanceo de categorías.
 - Business sense.
- Nuestra definición ingenua nos puede ayudar!



Métricas de evaluación – Matriz de confusión

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Error tipo I
(falso positivo)



Error tipo II
(falso negativo)



Métricas de evaluación – Matriz de confusión

		True Class		
		Positive	Negative	
Predicted Class	Positive	TP 147	FP 18	Se murió en el titanic $\hat{y} = 1$
	Negative	FN 30	TN 73	No se murió en el titanic $\hat{y} = 0$
		Se murió en el titanic ($y = 1$)	No se murió en el titanic ($y = 0$)	

Accuracy

Número de predicciones correctas dividido por el número total de predicciones. Ojo, cuando mi marca está desbalanceada, esta métrica puede ser engañosa!

		True Class		
		Positive	Negative	
Predicted Class	Positive	TP 147	FP 18	Se murió en el Titanic ($y = 1$)
	Negative	FN 30	TN 73	No se murió en el Titanic ($y = 0$)
		Se murió en el Titanic ($y = 1$)	No se murió en el Titanic ($y = 0$)	

$$Accuracy = \frac{TP + TN}{(TP + TN) + (FN + FP)}$$

$$Accuracy = \frac{147 + 73}{147 + 73 + 30 + 18} = \frac{220}{268} = 0.82$$

Accuracy puede ser misleading (Heterocromía)

		True Class	
		Positive	Negative
Predicted Class	Positive	TP 0	FP 0
	Negative	FN 2	TN 998

$$Accuracy = \frac{TP + TN}{(TP + TN) + (FN + FP)}$$

$$Accuracy = \frac{998}{1000} = 0.998$$

El modelo tiene un accuracy del 99% pero es **muy malo** identificando personas con heterocromía

Precision

De todo lo que el modelo predijo como positivo ¿A cuánto le pegue? En otras palabras, es la habilidad que tiene el modelo de no clasificar como positivo un evento negativo

		True Class		
		Positive	Negative	
Predicted Class	Positive	TP 147	FP 18	Se murió en el Titanic ($y = 1$)
	Negative	FN 30	TN 73	No se murió en el Titanic ($y = 0$)
		Se murió en el Titanic ($y = 1$)	No se murió en el Titanic ($y = 0$)	

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{147}{147 + 18} = \frac{147}{165} = 0.89$$

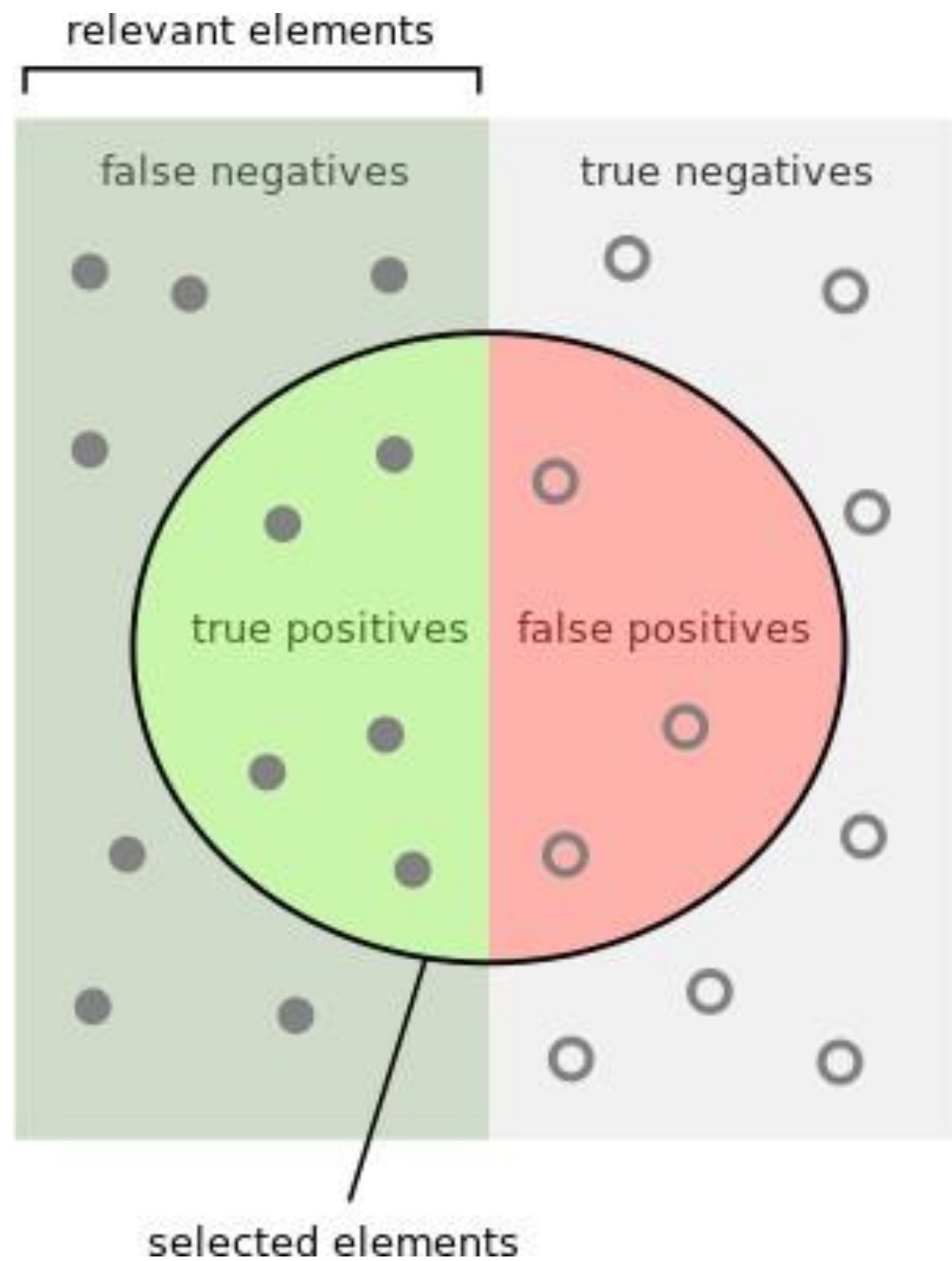
Recall (Sensitivity)

De todos los elementos que son positivos ¿Cuántos predije bien? – Recuerde el ejemplo de la heterocromía 😊

		True Class		
		Positive	Negative	
Predicted Class	Positive	TP 147	FP 18	Se murió en el Titanic ($y = 1$)
	Negative	FN 30	TN 73	No se murió en el Titanic ($y = 0$)
		Se murió en el Titanic ($y = 1$)	No se murió en el Titanic ($y = 0$)	

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{147}{147 + 30} = \frac{147}{177} = 0.83$$



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Trade-off entre Precision y Recall

Piensen en la historia del Pastorcito mentiroso.

- Cuando el Pueblo va a donde el Pastorcito y no hay ningún lobo ($y = 0$) cometen un **falso positivo**.
- Cuando el Pueblo no acude a la ayuda del Pastorcito cuando llega el lobo ($y = 1$) cometen un **falso negativo**.



Trade-off entre Precision y Recall

Piensen en la historia del Pastorcito mentiroso.

- Si el pueblo quisiera maximizar su Precision debe hacer 0 sus **Falsos Positivos**. Es decir que no debe acudir nunca a la llamada del Pastorcito. Pero, si hace eso, van a aumentar sus **Falsos Negativos**. Para subir la Precision tuvo que bajar el Recall.






F1 Score



Dependiendo de la aplicación, a veces es más importante tener un alto Precision que un Recall o viceversa. Cuando ambos son importantes, utilizamos la media armónica de ambas.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$F1\ score = 2 * \frac{0.89 * 0.83}{0.89 + 0.83} = \frac{1.48}{1.72} = 0.86$$



F_{β} Score



El F1 implica que el Recall es igual de Importante que el Precision. Pero esto se puede ajustar con el parámetro β . Entre más grande sea, más importante será el Recall en relación a Precision. Por ejemplo, con F2 Recall es dos veces más importante que Precision

$$F_{\beta} \text{ score} = (1 + \beta^2) * \frac{\text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}}$$

Specificity

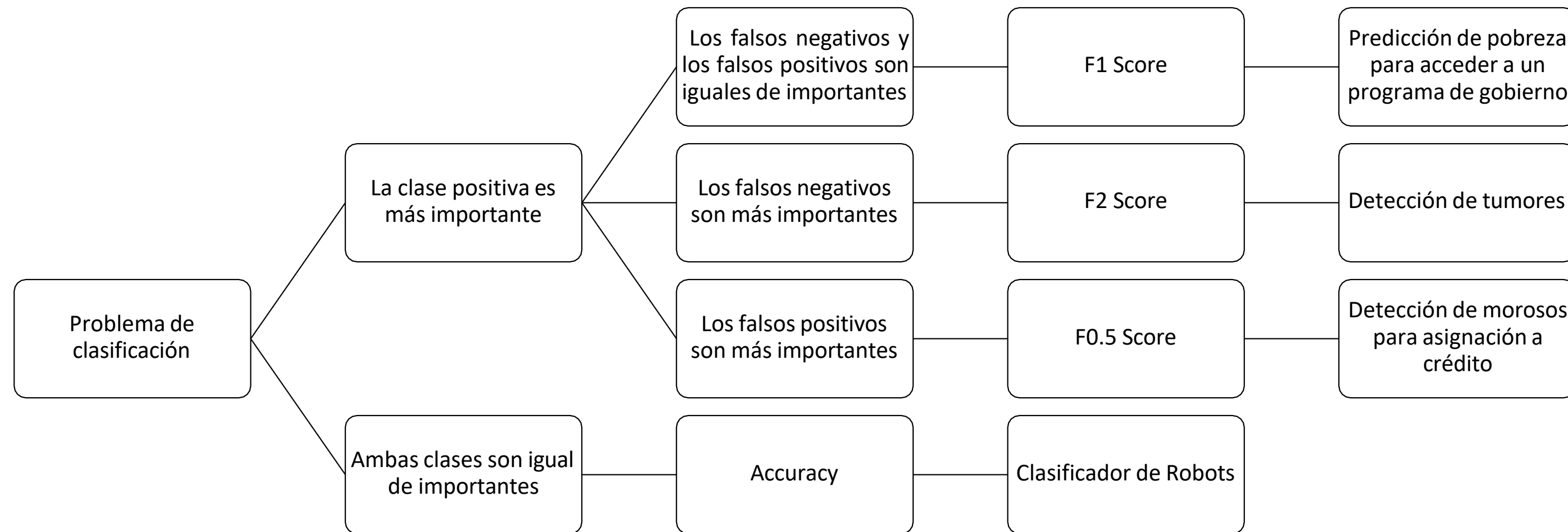
De todos los elementos que son negativos ¿Cuántos predije bien?

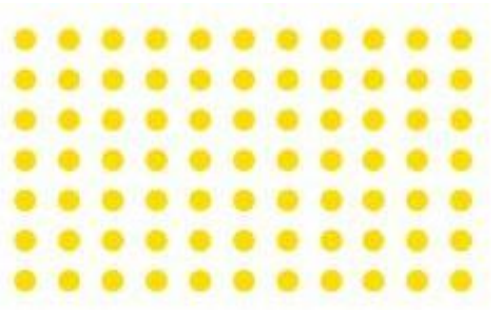
		True Class		
		Positive	Negative	
Predicted Class	Positive	TP 147	FP 18	Se murió en el Titanic ($y = 1$)
	Negative	FN 30	TN 73	No se murió en el Titanic ($y = 0$)
		Se murió en el Titanic ($y = 1$)	No se murió en el Titanic ($y = 0$)	

$$Specificity = \frac{TN}{TN + FP}$$

$$Specificity = \frac{73}{73 + 18} = \frac{73}{91} = 0.80$$

¿Cómo escoger la métrica?





Framework para clasificación



Uniendo todo lo aprendido (hasta ahora)



Ante un problema de clasificación, ustedes pueden (en principio) seguir la siguiente receta:

1. Identificar cuál(es) son las métricas relevantes en su contexto.
2. Con esta métrica en mente, pueden poner a "competir" a los distintos algoritmos que ya estudiamos para ver cuál es el mejor.
 1. Hagan un fit-predict de cada modelo como primer vistazo
 2. Hagan un fine-tuning del algoritmo (teniendo en cuenta temas de overfitting, etc) -> GridSearchCV, RandomizedCV
3. Comparen!

¡Gracias!

Aprendiendo juntos a lo largo de la Vida

educacioncontinua.uniandes.edu.co

Síguenos en **EdcoUniandes**

