UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS
ALMACENES Y MINERÍA DE DATOS

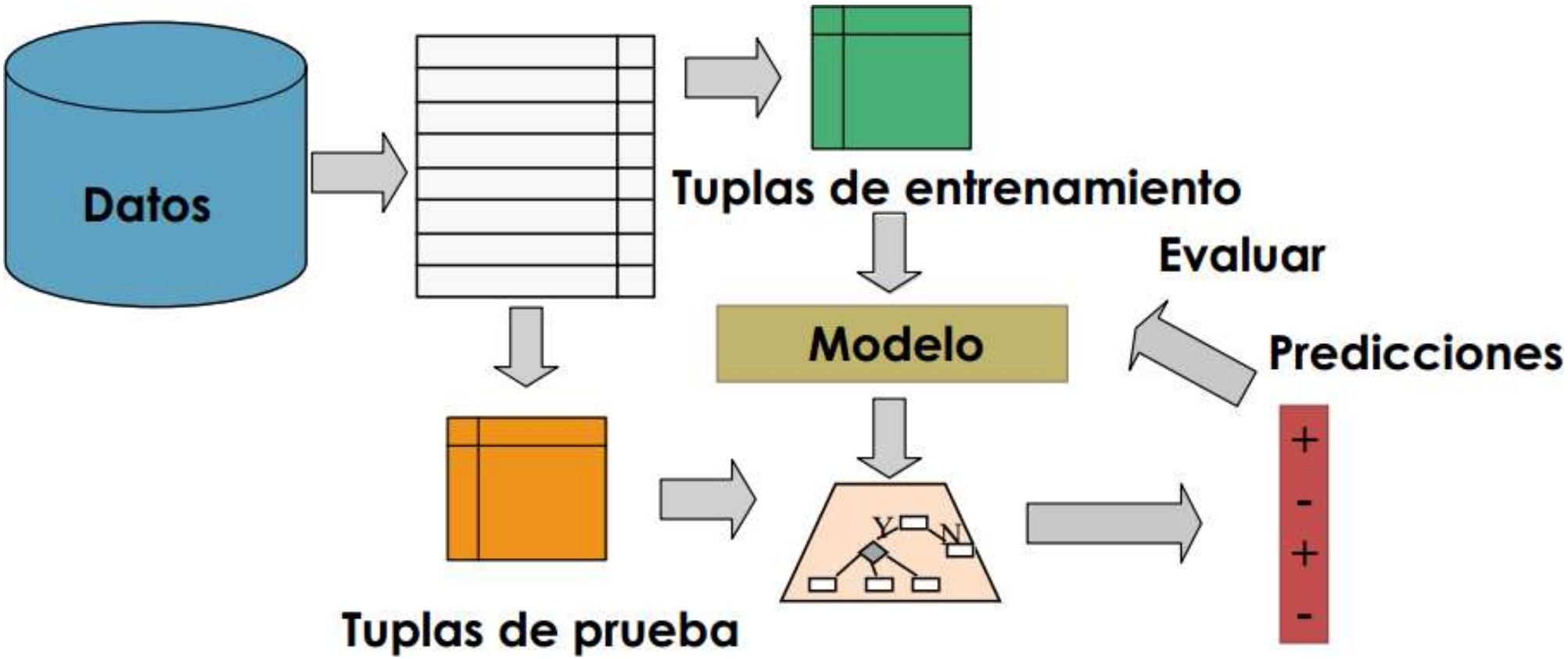
Clasificación: Evaluación

Gerardo Avilés Rosas
gar@ciencias.unam.mx

Exactitud del clasificador

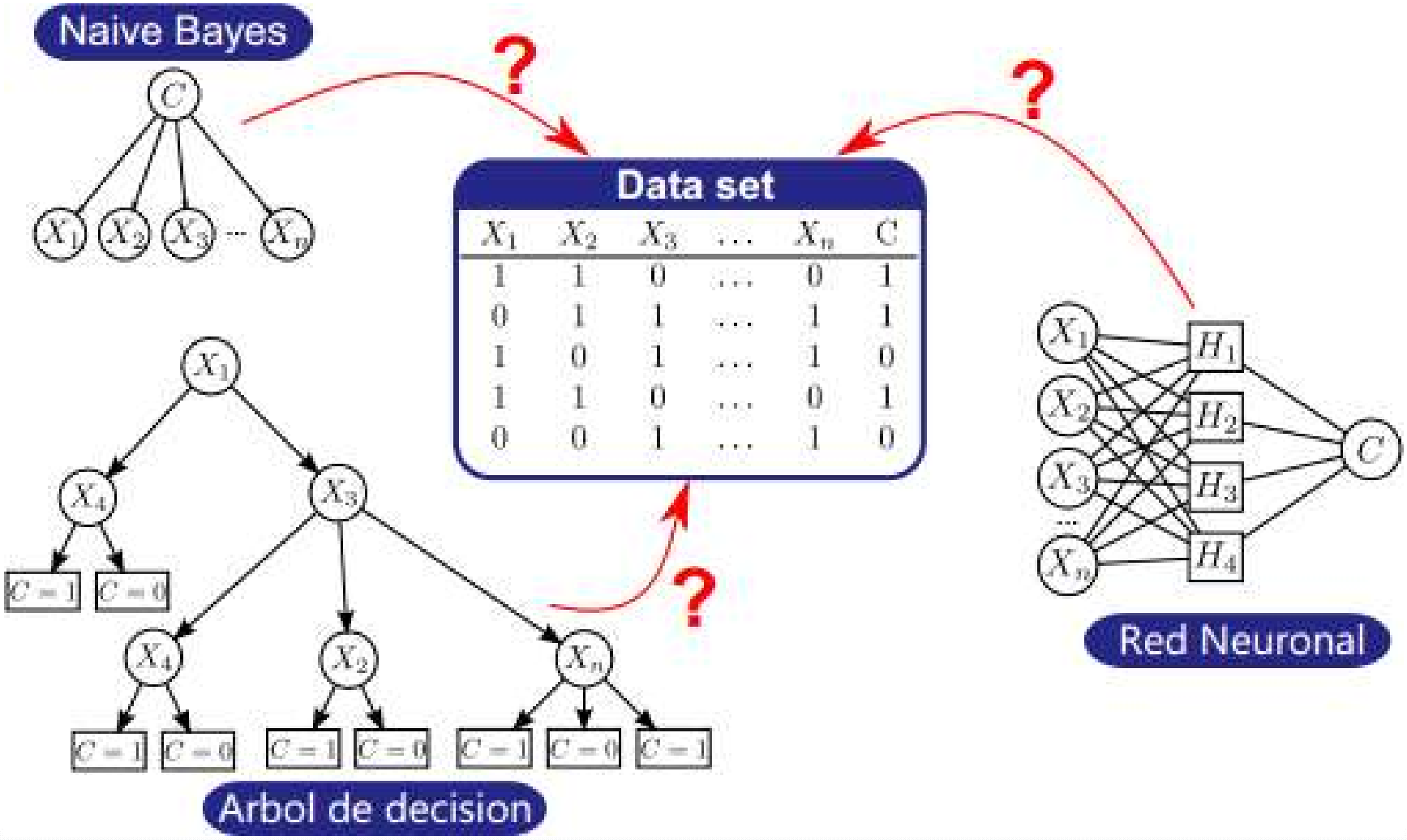
- *No hay razones independientes del contexto y de la aplicación que justifiquen la superioridad de un tipo de clasificador sobre otro.*
- *Si un algoritmo parece superior a otro en determinadas circunstancias, es consecuencia de su ajuste particular al problema de clasificación, no a su superioridad general como algoritmo.*
- *Los aspectos más importantes son: la información a priori y la cantidad de patrones para el entrenamiento.*

…¿Cómo trabaja la clasificación?



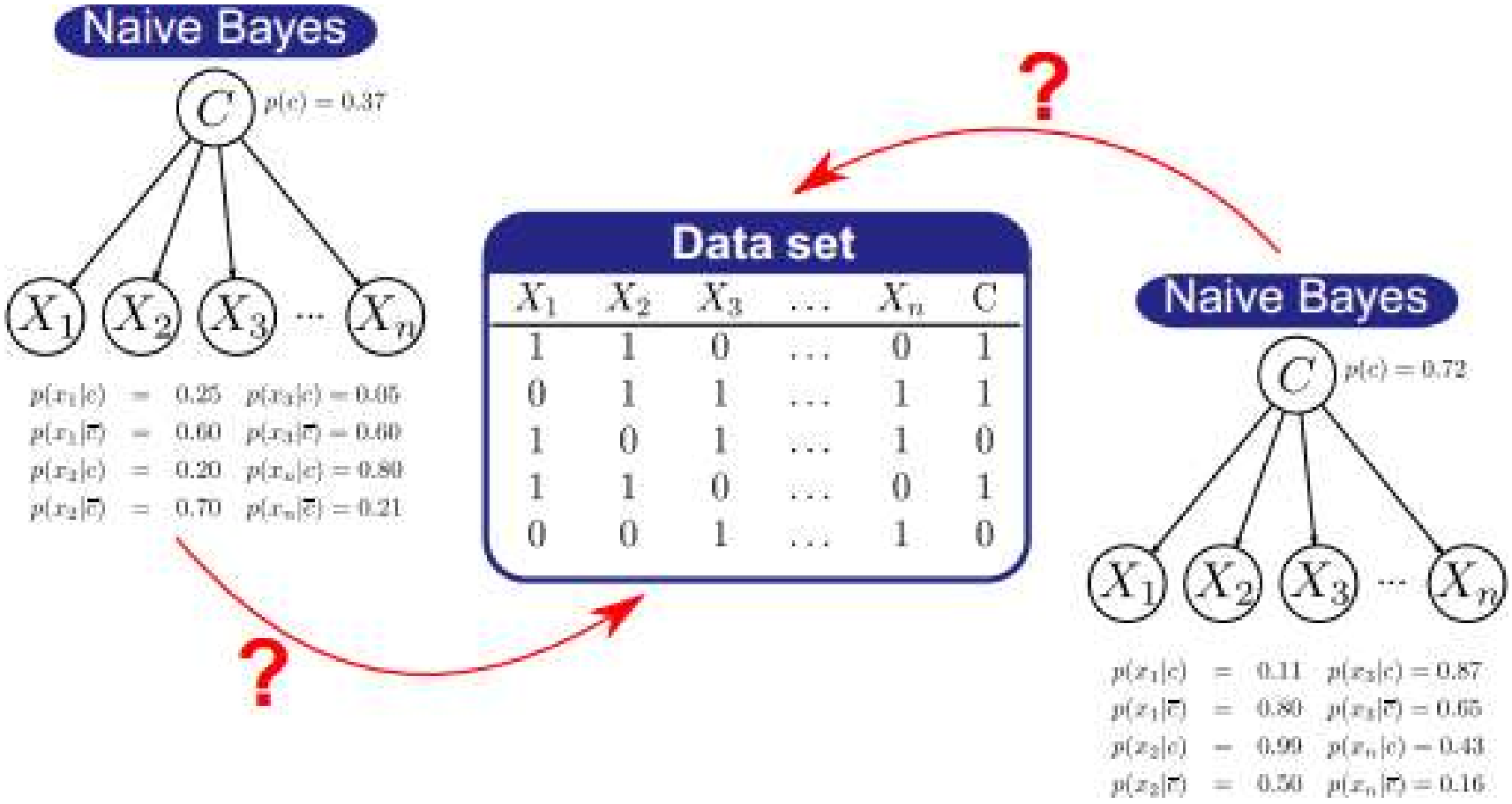
…Elegir un clasificador

¿Cuál es el mejor paradigma para un problema de clasificación?



…Elegir un clasificador

¿Cuál es el mejor configuración para un problema de clasificación?



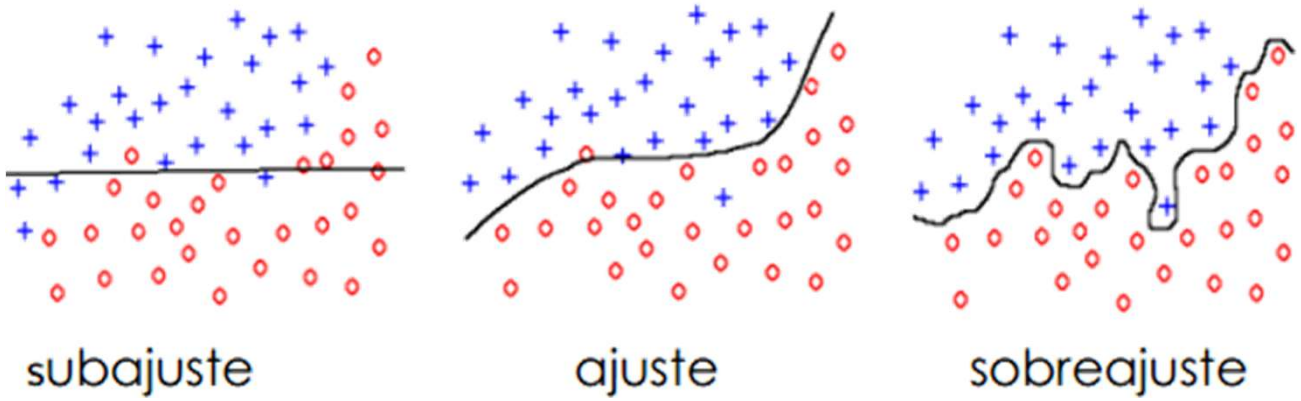
Credibilidad

¿Qué tanto deberíamos creer en un modelo que aprendió?

¿Qué tan predictivo es el modelo que aprendió?

- Estimar la exactitud del clasificador sobre las tuplas de entrenamiento no es un **buen indicador** del desempeño para futuros datos:
  - Hacer que un clasificador aprenda sobre el mismo conjunto de entrenamiento provoca que cualquier estimación basada en estos datos resulte en **estimaciones bastante optimistas** o engañosas.
  - El problema es que los nuevos datos probablemente **no sean exactamente** los mismos que las tuplas sobre las que se entrenó.
- Por otro lado, el **aprendizaje excesivo** a partir de los datos de entrenamiento suele conducir a resultados deficientes en la clasificación de nuevos datos.
- Debemos lograr que el clasificador **tenga la habilidad de generalizar**.

…Credibilidad



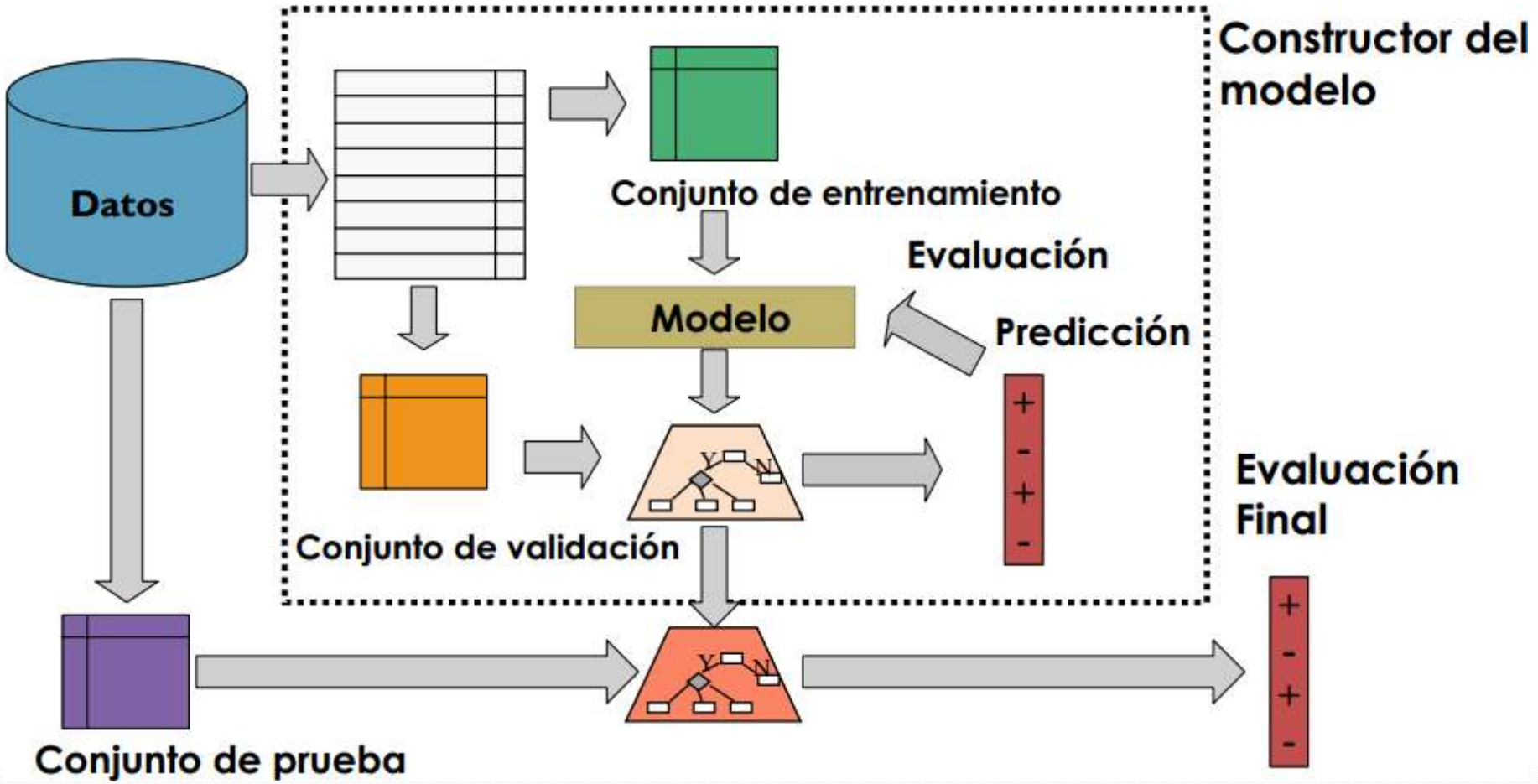
- La **exactitud de un clasificador** en un conjunto de prueba dado, es el **porcentaje de tuplas** que se clasifican **correctamente** mediante el modelo aprendido.
- Para evaluar el desempeño, debemos considerar entre otros:
  - Datos de prueba
  - Medidas de efectividad
  - Decidir cómo medir cuando los datos son limitados o están sesgados
  - Considerar o no el costo de los errores
  - Evaluar un modelo vs. comparar varios modelos
  - Tiempo que tarda en realizar una clasificación
  - Costo de construcción del modelo
  - Interpretabilidad, etc.

Tasa de error de clasificación

- Se trata de una **medida natural** para problemas de clasificación, **mide la porción de errores cometidos** en todo el conjunto de instancias.
- Datos de entrenamiento vs. datos de prueba
  - **Entre más datos de entrenamiento** se tengan, el modelo podrán realizar **mejores generalizaciones**.
  - **Entre más datos de prueba** se tengan, el modelo dará una **mejor estimación** de la **probabilidad de error de clasificación**.
- La tasa de error se puede estimar con base en:
  - Conjunto de tuplas de entrenamiento → **Resustitución**
  - Conjunto de tuplas de prueba → **Hold out**
- **Recomendación:**
  - **Nunca evaluar el rendimiento de un modelo sobre los datos de entrenamiento**. La conclusión sería optimista y parcial.
  - Los **datos de prueba no se deben utilizar** de ninguna manera para **crear el clasificador** ni tampoco para **ajustar parámetros** del modelo.

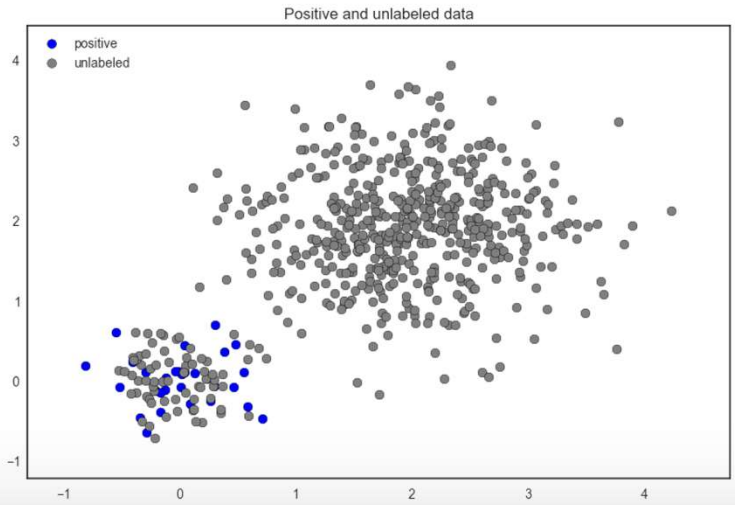
Ajuste de parámetros

- El procedimiento adecuado es utilizar tres conjuntos: ***datos de entrenamiento, datos de validación y datos de prueba.***
- Los datos de validación se usan para **optimizar los parámetros.**



Datos no balanceados

- En ocasiones, las clases tienen una frecuencia muy desigual:
  - Diagnóstico médico: **95% saludable**, **5% enfermedad**
  - Comercio electrónico: **99% no compra**, **1% compra**
  - Seguridad: **99.99%** de los ciudadanos **no son terroristas**
- De acuerdo al primer escenario, el modelo clasifica correctamente el **95% de las ocasiones**, lo cual sería muy bueno, **pero no siempre es útil**.
- Una situación similar se presenta para clasificadores que trabajan con varias clases.
- **¿Cómo debemos entrenar un clasificador y evaluarlo para problemas con datos no balanceados?**



…Manejando datos no balanceados

Problemas con dos clases:

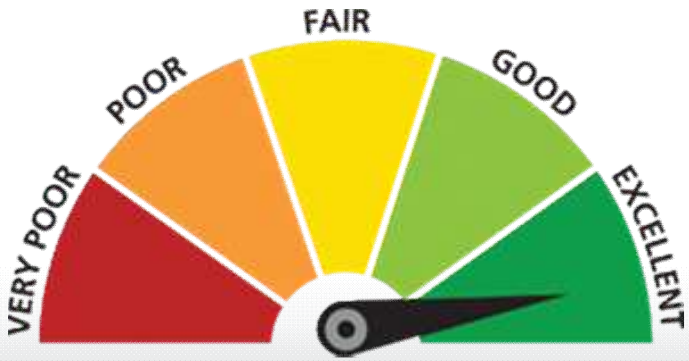
- Construir un **conjunto de entrenamiento equilibrado** y utilizarlo para entrenar al clasificador:
  - Seleccionar al azar el número deseado de instancias de clases minoritarias.
  - Añadir el mismo número de instancias de clase mayoritarias, seleccionadas aleatoriamente.
- Construir un **conjunto de pruebas equilibrado** (diferente del conjunto de entrenamiento) y probar el clasificador que lo utiliza.

Problemas con múltiples clases:

- Generalizar el "equilibrio" a varias clases.
- Asegurar que cada clase esté representada con proporciones aproximadamente iguales en los conjuntos de datos de entrenamiento y prueba.

¿Cómo comparar modelos de clasificación?

- Se necesita algún medio que permita **medir el rendimiento** de un modelo de clasificación**:**
  - **Puntuación** (score): Función que proporciona una medida de **calidad** para un clasificador al resolver un problema de clasificación.
- Nos interesa la **mejor calidad**, ¿qué significa **mejor calidad**?
  - En qué estamos interesados, qué queremos optimizar, características del problema, características del dataset, etc.
- Existen **diferentes funciones** de puntuación:
  - Basadas en la **matriz de confusión**: exactitud, error de clasificación, recall, especificidad, precisión, F-score, índica Kappa, sensibilidad.
  - Basados en la **curva ROC**(***Receiver Operating Characteristics***): área bajo la curva, curva Lift.



Matriz de confusión

- Es una herramienta útil para analizar **qué tan bien** un clasificador puede **reconocer tuplas de diferentes clases**.
  - **Si se tienen m clases, la matriz se construye a partir de una tabla de mxm.**
- Una entrada, $\mathbf{cm_{i,j}}$ en cada una de las **m filas** indican **el número de tuplas** de la **clase i** que fueron etiquetadas por el clasificador como **clase j**. Para un problema de **dos clases**:

		Predicción		
		c^+	c^-	Total
Actual	c^+	TP	FP	N^+
	c^-	FN	TN	N^-
	Total	$\hat{N}^+$	$\hat{N}^-$	N

R. Kohavi, F. Provost: Glossary of terms, Machine Learning, Vol. 30, No. 2/3, 1998, pp. 271-274.

…Matriz de confusión

- Para un problema de múltiples clases:

		Predicción					
		c_1	c_2	c_3	...	c_n	Total
Actual	c_1	TP_1	FP_{12}	FP_{13}	...	FP_{1n}	N_1
	c_2	FN_{21}	TP_2	FP_{23}	...	FP_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	...	FP_{3n}	N_3
	...	...	...	...	...	...	...
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	...	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_3$	...	$\hat{N}_n$	N

- Para que un clasificador tenga **buena exactitud**, idealmente **la mayor parte de las tuplas** debería están a lo largo de la **diagonal principal** de la matriz de confusión y el resto de las entradas estar **próximos a cero**.
- La tabla puede tener filas o columnas adicionales para proporcionar los totales o los tipos de reconocimiento por clase

Medidas de desempeño

- Dadas dos clases, podemos hablar en términos de **tuplas positivas** vs. **tuplas negativas**:
  - Los **verdaderos positivos** se refieren a las tuplas positivas que fueron etiquetados correctamente por el clasificador, mientras que los **verdaderos negativos** son las tuplas negativas que fueron etiquetados correctamente por el clasificador.
  - Los **falsos positivos** son las tuplas negativas que fueron etiquetados incorrectamente. Del mismo modo, los **falsos negativos** son las tuplas positivas que fueron etiquetados incorrectamente.

		Clase predicha	
		(+) C_1	(−) C_2
Clase actual	(+) C_1	Verdaderos positivos (**TP**)	Falsos negativos (**FN**)
	(−) C_2	Falsos positivos (**FP**)	Verdaderos negativos (**TN**)

…Medidas de desempeño

$$\text{Exactitud, ACC} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{Error de clasificación} = \frac{FP + FN}{TP + FN + FP + TN}$$

$$\text{Sensibilidad}(\text{TP rate, recall}) = \frac{TP}{TP + FN}$$

$$\text{Especificidad}(\text{TN rate}) = \frac{TN}{TN + FP}$$

$$\text{Precisión } (\text{Valor predicho positivo, PPV}) = \frac{TP}{TP + FP}$$

$$\text{Valor predicho negativo } (\text{NPV}) = \frac{TN}{TN + FN}$$

$$\text{Falsas alarmas } (\text{FP rate}) = \frac{FP}{FP + TN} = 1 - especificidad$$

$$\text{Tasa de falsos descubrimientos}(FDR) = \frac{FP}{FP + TP} = 1 - PPV$$

$$\text{Tasa de falsos negativos}(\text{FN rate}) = \frac{FN}{FN + TP}$$

Clase predicha

Clase actual	(+) C_1	(−) C_2
(+) C_1	**TP**	**FN**
(−) C_2	**FP**	**TN**

…Medidas de desempeño

$$\text{Exactitud Balanceada}(BACC) = \left(\frac{TP}{TP+FN} + \frac{TN}{FP+TN} \right) \Big/ 2$$

$$\text{F-measure} = \left[\frac{1}{2} \left(\frac{1}{\text{Precision}} \right) + \left(\frac{1}{\text{Recall}} \right) \right]^{-1} = \frac{2\text{Recall} \bullet \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$kappa = \frac{TP + TN - E(TP + TN)}{TP + TN + FP + FN - E(TP + TN)}$$

Clase predicha

Clase actual		(+) C_1	(−) C_2
	(+) C_1	**TP**	**FN**
	(−) C_2	**FP**	**TN**

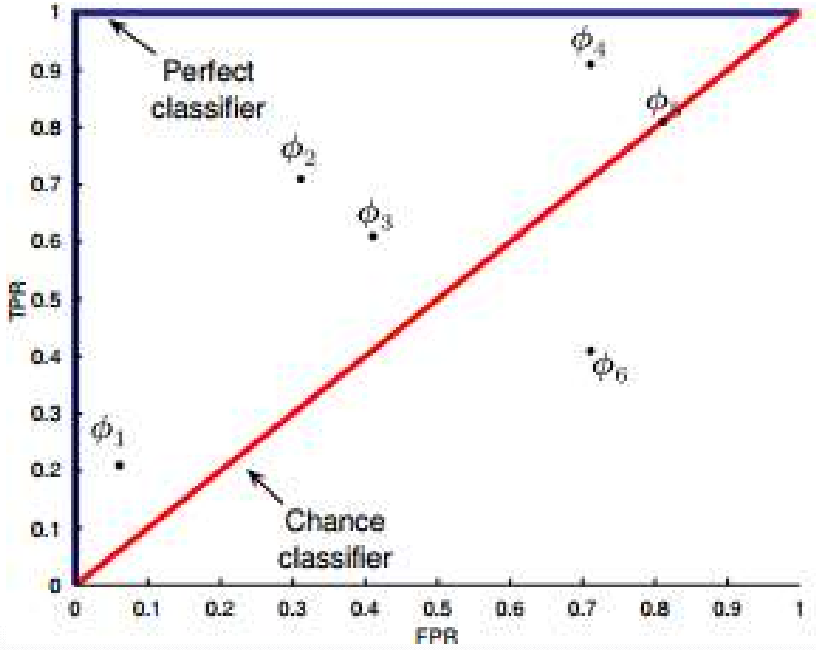
Aplicaciones de la precisión y recall

- **Filtrar SPAM**
  - Decidir si un correo electrónico es SPAM o no
  - **Precisión:** Proporción de SPAM real en la bandeja de SPAM.
  - **Recall:** Proporción de mensajes de SPAM totales identificados por el sistema

- **Análisis de sentimientos**
  - Clasificar opiniones sobre productos específicos dados por los usuarios en Blogs, webs, foros, etc.
  - **Precisión:** Proporción de opiniones clasificadas como positivas siendo realmente positivas.
  - **Recall:** Proporción de opiniones positivas identificadas como positivas.

Receiver Operating Characteristic

- Es una **representación gráfica** de la **sensibilidad** para un sistema de clasificación binario según se varía el umbral de discriminación.
- También representa la **tasa de verdaderos positivos (eje Y)** vs. **falsos positivos (eje X)**.
- Proporciona herramientas para seleccionar los modelos posiblemente óptimos y descartar modelos que nos son óptimos independientemente de el costo de la distribución de las dos clases sobre las que se decide.

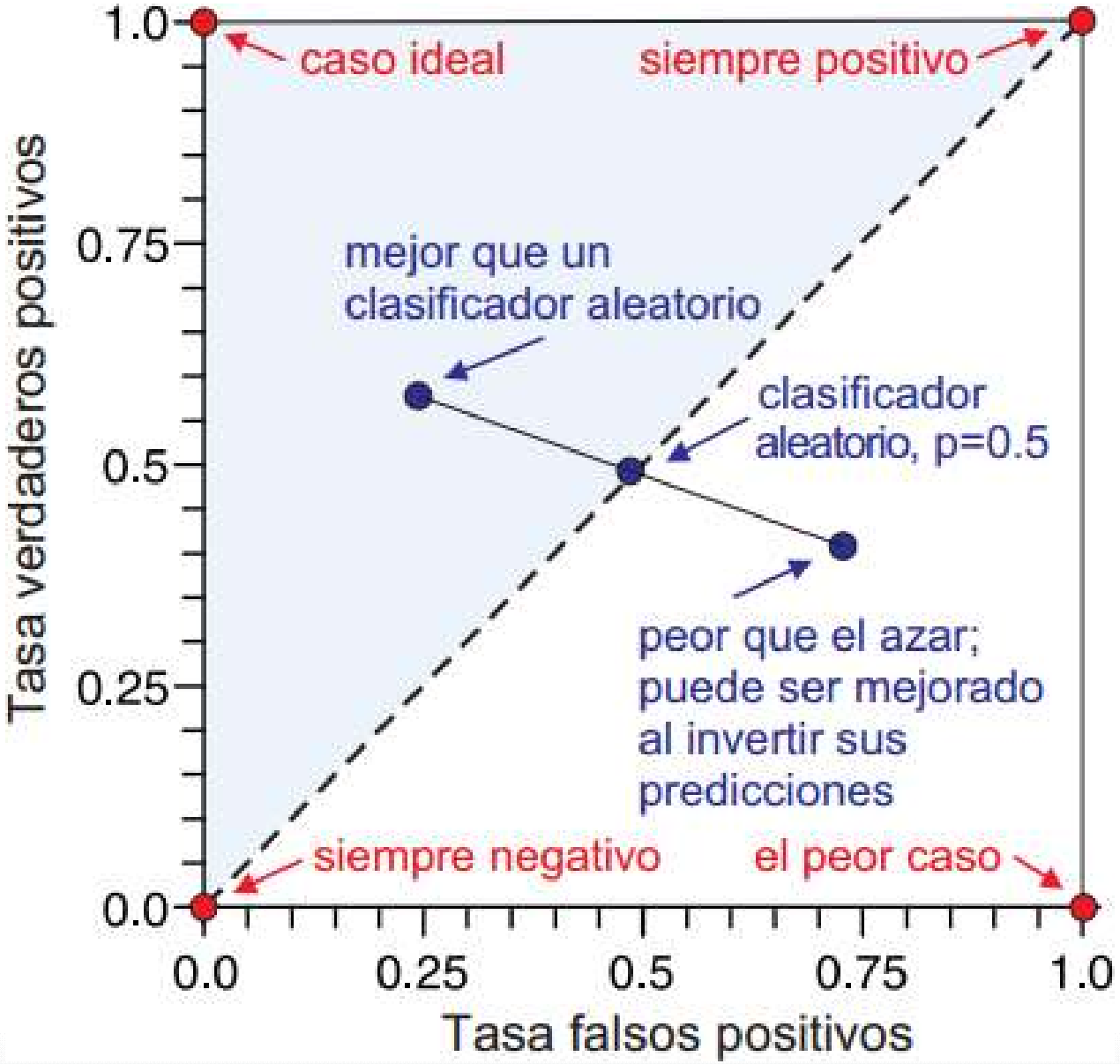


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

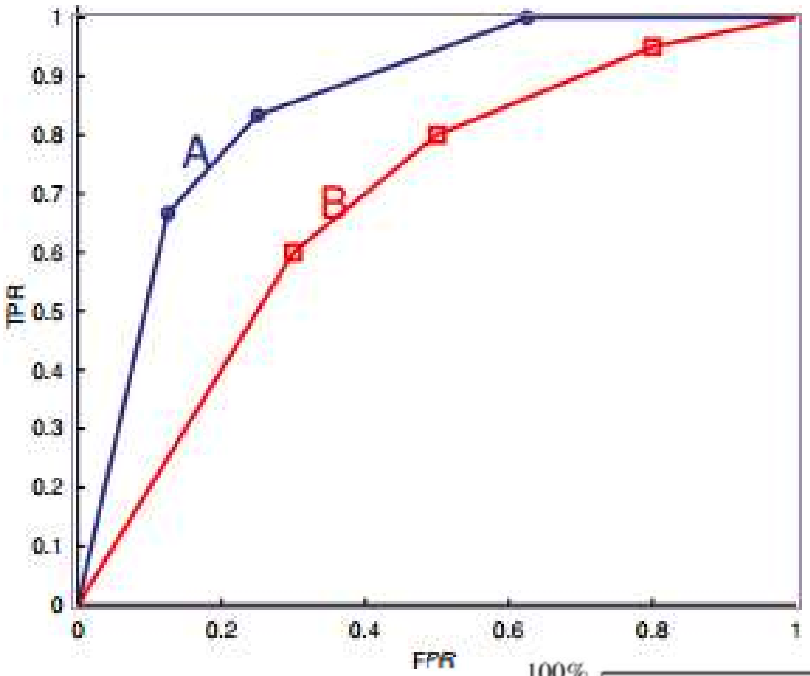
…Receiver Operating Characteristic

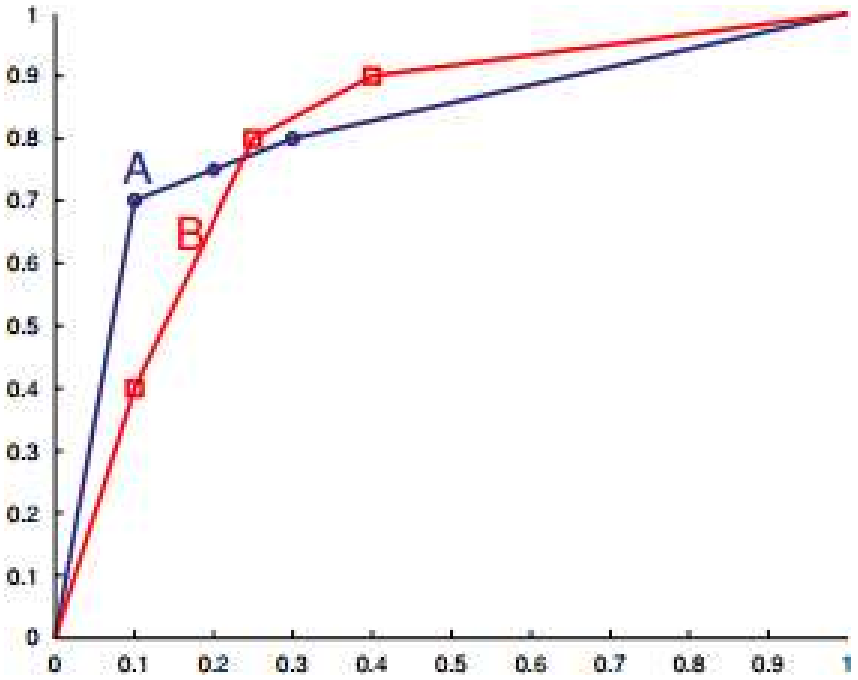
- El **mejor método posible** de predicción **se situaría** en un punto en la **esquina superior izquierda** del espacio ROC, representando un **100% de sensibilidad** (*ningún falso negativo*) y un **100% de especificidad** (*ningún falso positivo*). A este punto también se le llama una **clasificación perfecta**.
- Por el contrario, **una clasificación totalmente aleatoria** daría **un punto** a lo largo de la **línea diagonal** (*línea de no-discriminación*), desde el extremo inferior izquierdo hasta la esquina superior derecha.
- La **diagonal divide el espacio ROC**, los puntos **por encima** de la diagonal representan **buenos resultados de clasificación** (*mejor que el azar*), **puntos por debajo** de la línea representan **resultados pobres** (*peor que al azar*).
- Es importante notar que la salida de un predictor consistentemente pobre simplemente podría ser invertida para obtener un buen predictor.

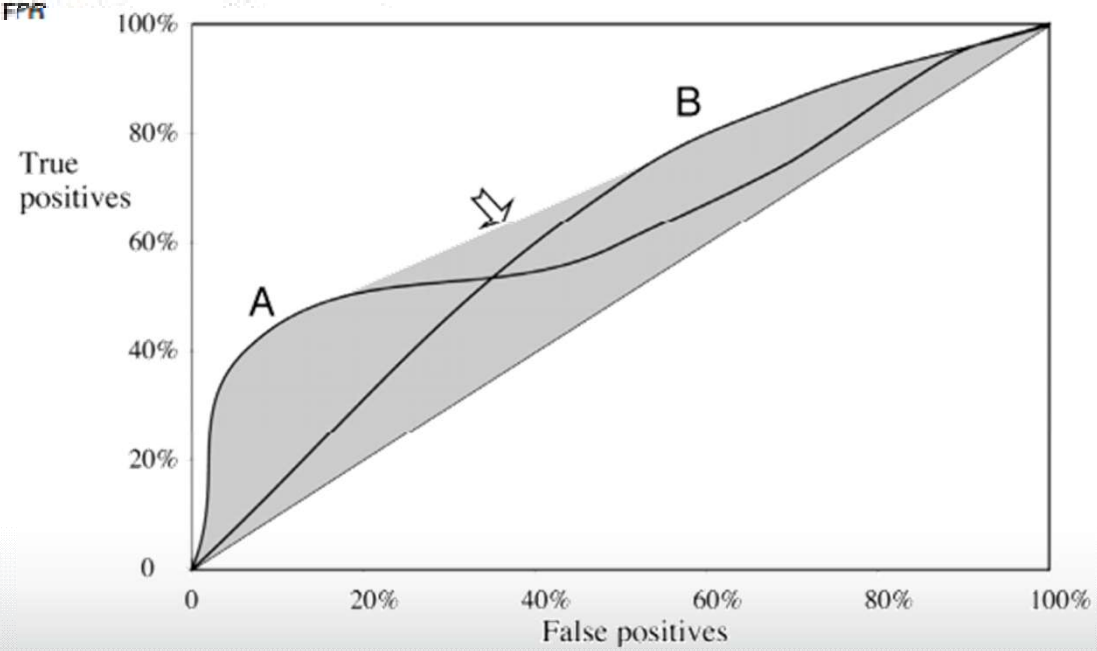
…Receiver Operating Characteristic



Dominancia vs. No dominancia



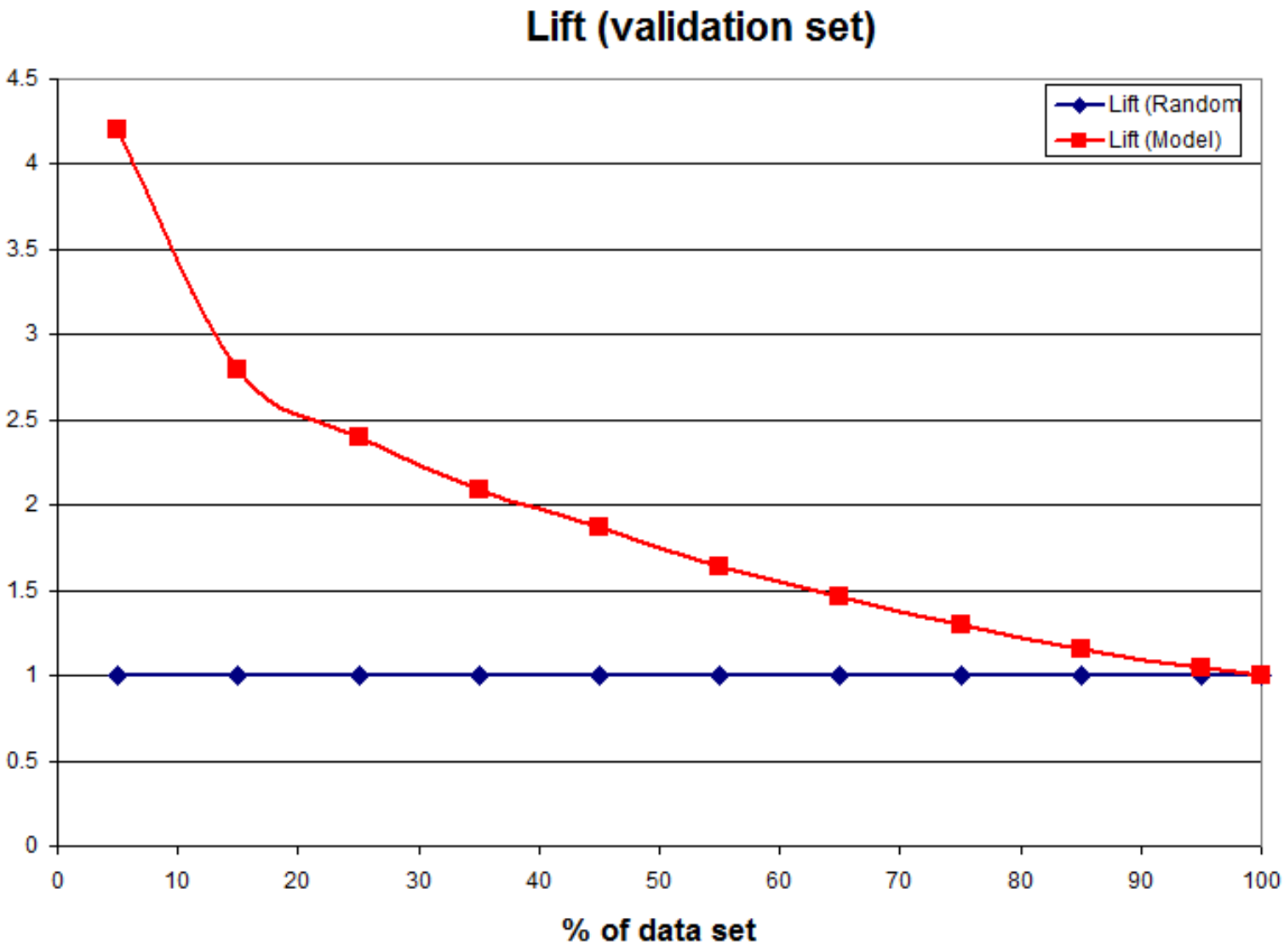




Gráficas Lift

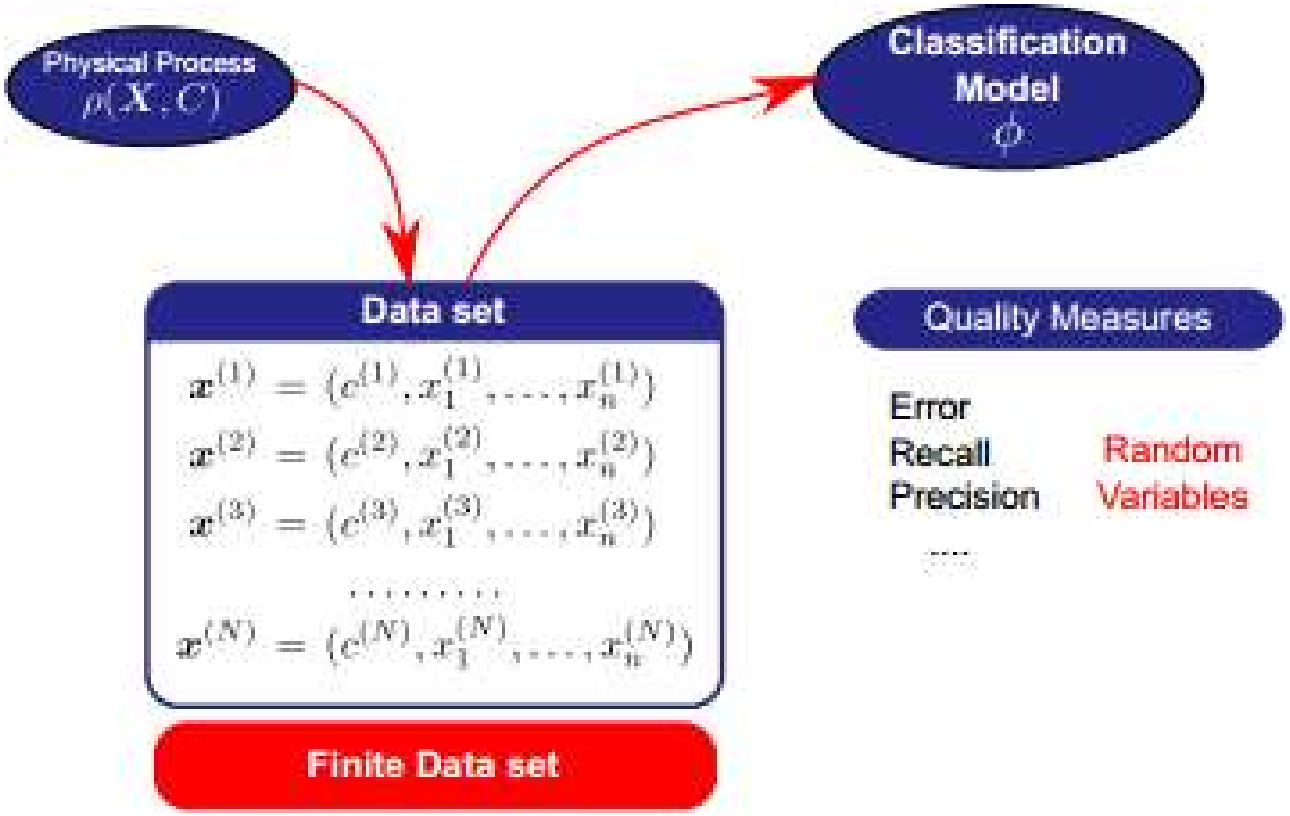
- **Idea:** dada una clase, cada éxito representa un beneficio
- **Motivación:** mercadotecnia
  - Modelo predice si un cliente responde a una oferta.
  - Cuántos folletos debemos enviar para que respondan X número de clientes
- **Solución:**
  - Graficar porcentaje de datos vs. número de éxitos (lift chart)
  - Ver qué porcentaje de datos necesito para tener X éxitos.
- **Construcción:**
  - Fijar una clase
  - Ordenar el conjunto de prueba de mayor a menor probabilidad de la clase
  - Graficar en el eje X el porcentaje del conjunto de prueba
  - Graficar en el eje Y el número de datos correctamente predichos en la clase

…Gráficas Lift



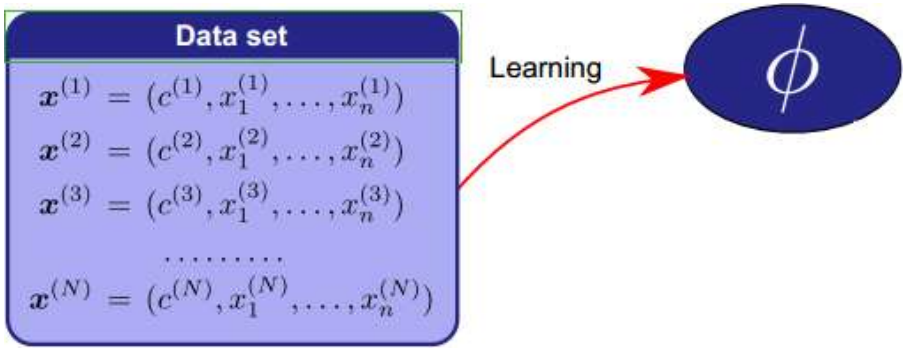
Estimación de modelos

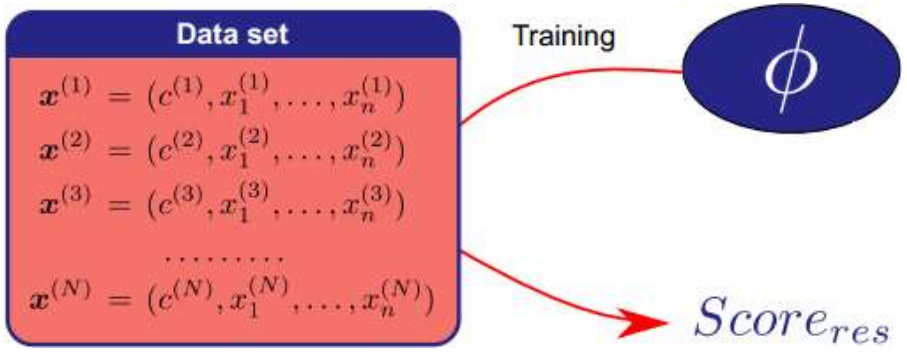
- Seleccionar una puntuación para medir la calidad.
- Calcular el valor real de la puntuación.
- Poca información disponible



Resustitución

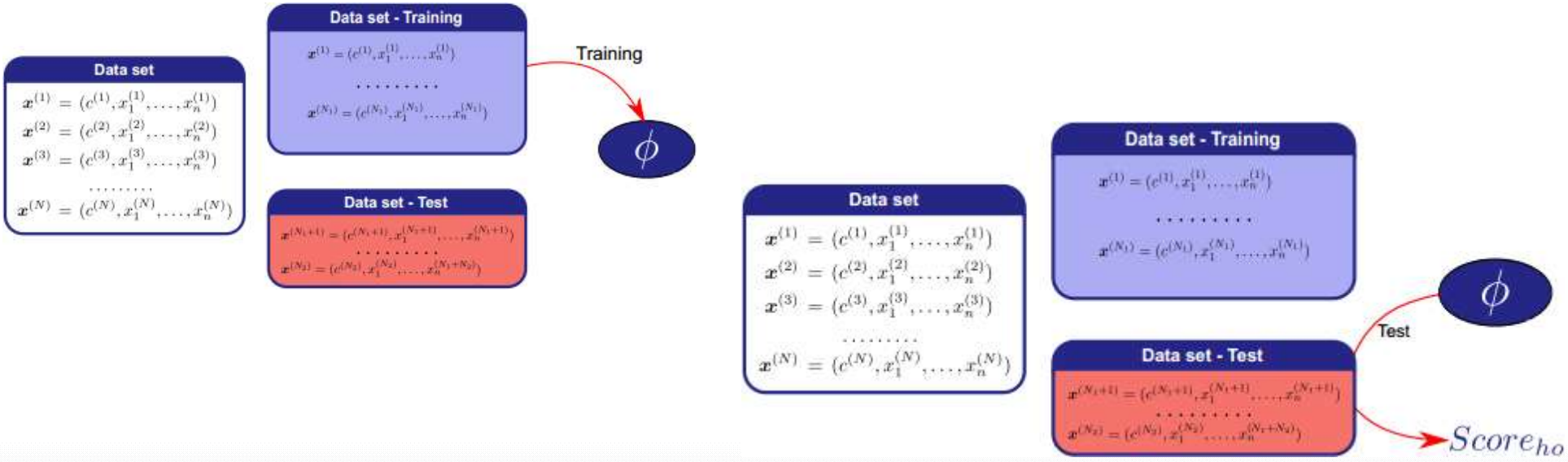
- Se trata del **más simple** de los métodos de estimación.
- Es un método que realiza la **estimación del error** directamente sobre los datos de entrenamiento.
- A este tipo de error se le conoce como **error de resustitución**.
- Problemas:
  - El modelo tiende a mostrar buen desempeño en datos de entrenamiento.
  - Mala predicción de error sobre los datos objetivo.
  - Realiza una estimación sesgada
  - Variación menor.
  - Demasiado optimista (problema de superposición).
  - Mal estimador del verdadero error de clasificación.





Hold – Out

- Típicamente se cuenta con un conjunto de datos.
- Se dividen los datos disponibles en dos conjuntos de datos disjuntos: entrenamiento y prueba:
  - Se requiere que los dos conjuntos sean representativos de los datos objetivo.
- Problema:
  - Para producir un buen clasificador necesitamos usar la mayor cantidad de datos para entrenamiento.
  - Para obtener una buena estimación del error objetivo necesitamos usar la mayor cantidad de datos para prueba

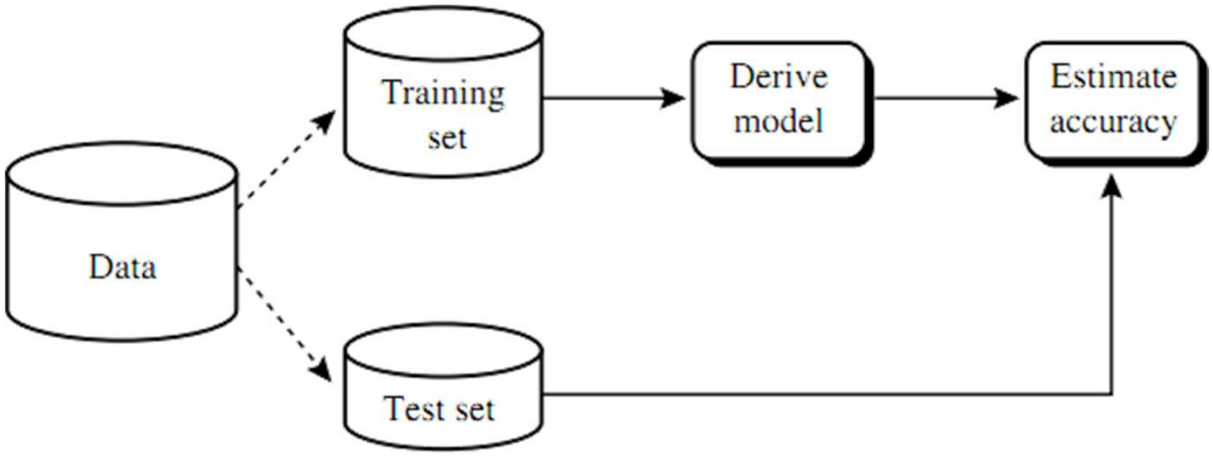


…Hold – Out

- En general, no se puede saber si los datos (entrenamiento o prueba) son representativos pero, sí podemos saber si están todas las clases "representadas":
  - Si una clase no está representada en los datos de entrenamiento, es de esperar que el modelo no se desempeñe bien en datos de esa clase.
  - Si la clase no está representada en los datos de entrenamiento, no mediremos el error asociado a datos que no se clasifican bien en la clase.
- **Retención:**
  - 2/3 de los datos se asignan para entrenamiento, el 1/3 restante para prueba.
  - Estimación pesimista porque sólo una parte de los datos se utiliza para derivar el modelo
- **Hold-out estratificado:**
  - Clases ocurren con la misma frecuencia en partición entrenamiento/prueba.
  - Salvaguarda básica para sesgo.
- **Holdout repetitivo:**
  - Repetir la prueba varias veces pero cambiando la partición entrenamiento/prueba.
  - Error estimado: promedio de errores de cada iteración

Submuestreo aleatorio simple

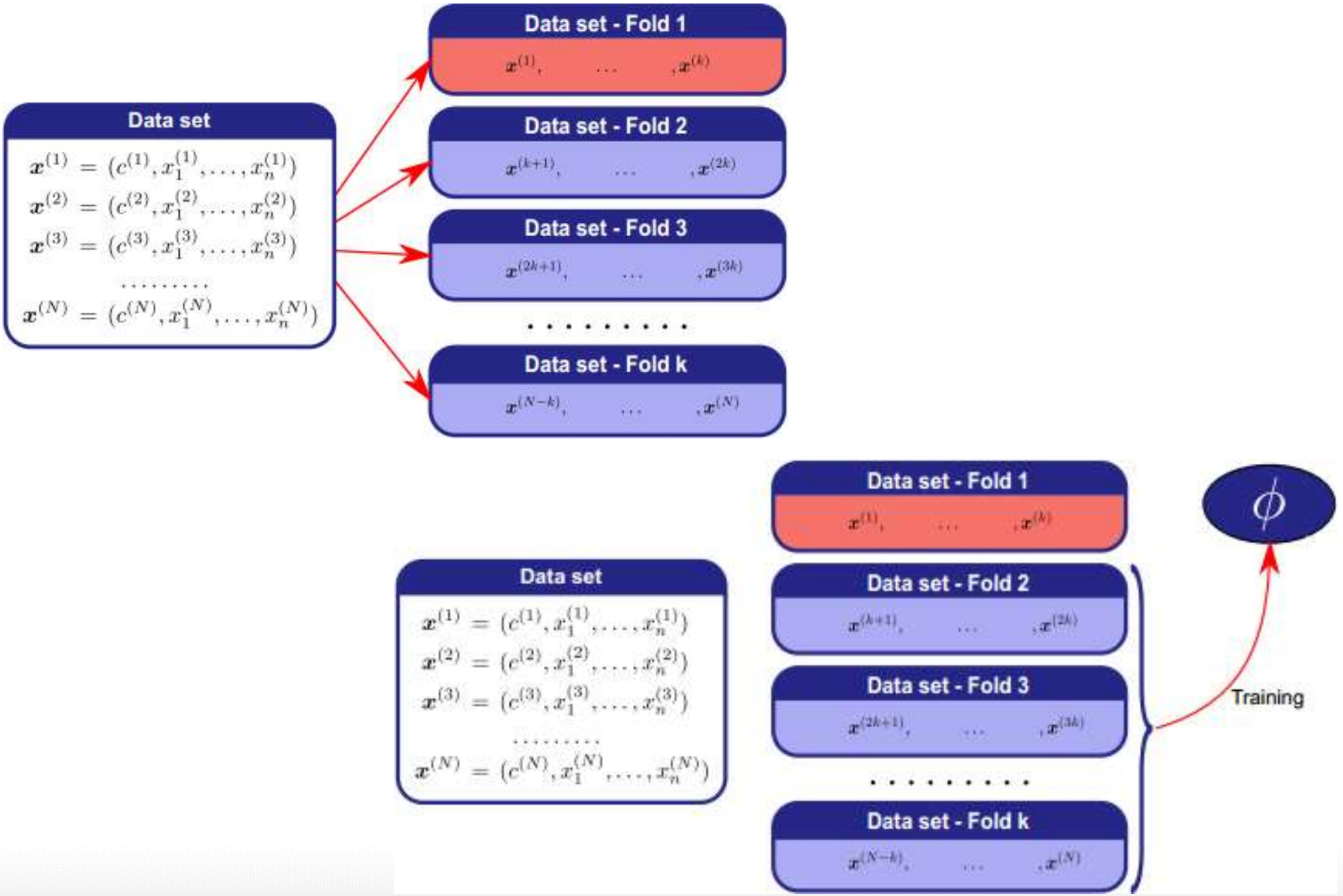
- **Submuestreo aleatorio** es una variación del **método de retención** en la que se repite el **método de retención k veces**.
  - En cada iteración, una cierta porción de los datos se selecciona de forma aleatoria para la etapa de entrenamiento.
  - La estimación global exactitud se toma como el promedio de las precisiones de los obtenidos a partir de cada iteración. (Para la predicción, podemos tomar el promedio de los índices de error de predicción).
  - También se conoce como **Hold-out de repetitivo**.
  - No es un método óptimo ya que los diferentes conjuntos de pruebas se pueden superponer.



Validación cruzada

- Es otra forma de **Hold-out repetitivo**.
- Se conoce como **validación cruzada de k fold**:
  - Los datos iniciales se **dividen aleatoriamente** en ***k subconjuntos mutuamente excluyentes*** o "pliegues", $\boldsymbol{D_1, D_2, \ldots, D_K}$, cada uno de **aproximadamente el mismo tamaño**.
  - De esta forma, el entrenamiento y las pruebas se realizan ***k veces***. En la ***iteración i***, la ***partición*** $\boldsymbol{D_i}$ se reserva como conjunto de prueba, y las particiones restantes se utilizan en conjunto para entrenar el modelo.
  - Es decir, en la primera iteración, subconjuntos $\boldsymbol{D_2, D_3, \ldots, D_k}$ sirven colectivamente como el conjunto de entrenamiento con el fin de obtener un primer modelo , que se prueba en $\mathbf{D_1}$; la segunda iteración es entrenado en subconjuntos $\boldsymbol{D_1, D_3, \ldots, D_k}$ y probado en $\mathbf{D_2}$, y así sucesivamente.
- A diferencia de los métodos de **retención** y **submuestreo**, cada muestra se utiliza el mismo número de veces para la entrenamiento y una vez para la prueba.
- Su gran desventaja es el **costo computacional**: se debe inducir el modelo **n veces**, de manera que no es factible hacerlo para conjuntos de datos grandes.

…Validación cruzada



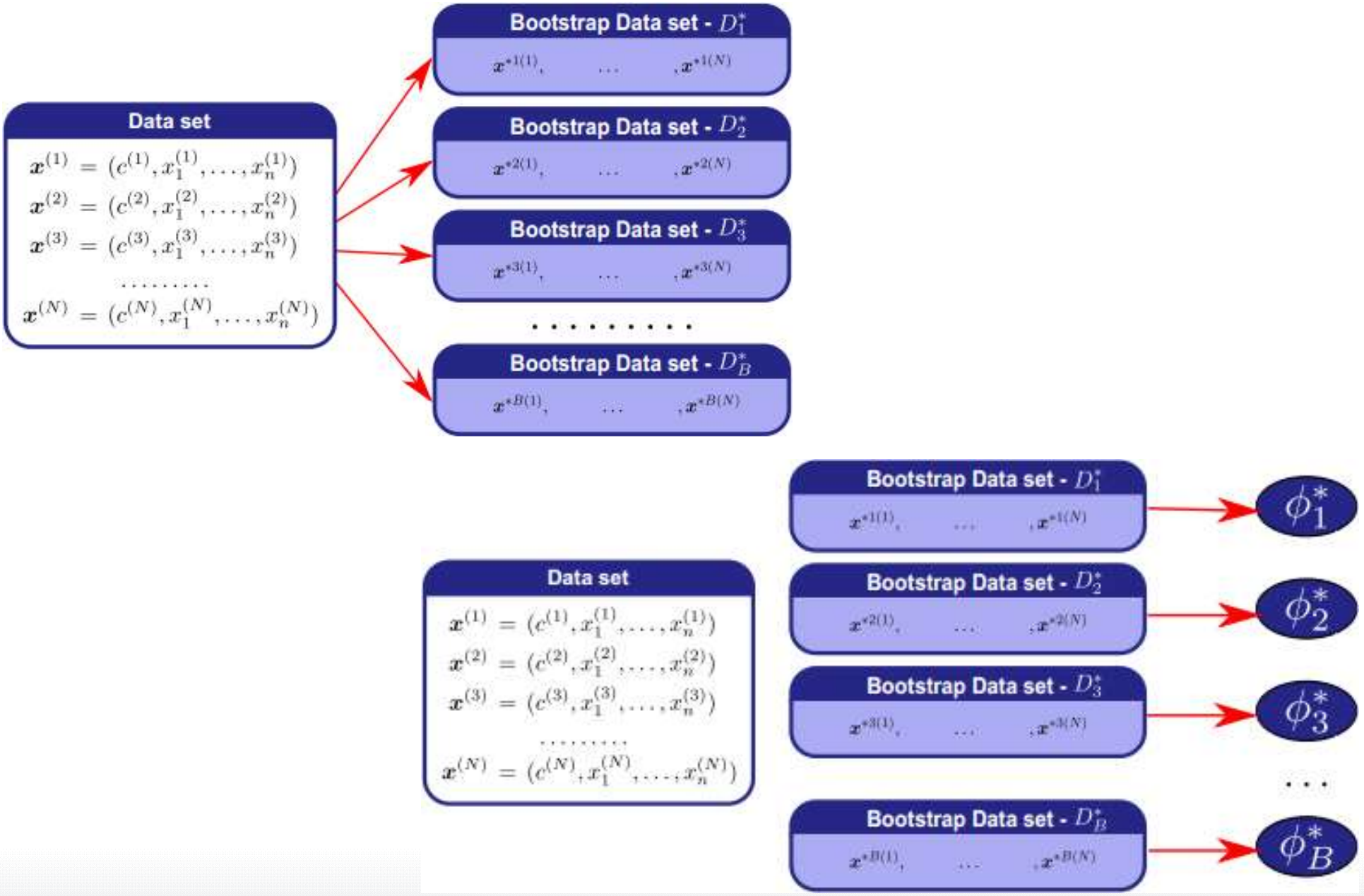
…Validación cruzada

- Para la clasificación, la estimación de la precisión es el número total de clasificaciones correctas de las iteraciones k, dividido por el número total de tuplas en los datos iniciales.
- Para la predicción , la estimación del error se puede calcular como la pérdida total de las iteraciones k, dividido por el número total de tuplas iniciales.
- En **validación cruzada estratificada**, los folds son estratificado de modo que la distribución de clases de las tuplas en cada pliegue es aproximadamente la misma que en los datos iniciales.
- En general, se recomienda una validación cruzada de **10 veces** para estimar la precisión, debido a su relativamente bajo sesgo y varianza.
- **Leave-one-out**, se trata de un caso particular:
  - k es el número de datos, sólo una muestra se "deja fuera" a la vez durante la prueba.
  - Utiliza la mayor cantidad de datos para entrenar. Útil cuando se tienen pocos datos.
  - Muy costoso computacionalmente. No se puede implementar estratificación
  - Determinístico: el experimento siempre da el mismo resultado.

Bootstrap

- Método que **realiza un muestreo** de las tuplas de entrenamiento **con reemplazo** (*i.e. cada vez que se selecciona una tupla, es igualmente probable que se seleccione otra vez y se vuelva a agregar al conjunto de entrenamiento*).
- Existen varios métodos, el más común se conoce como **bootstrap .632:**
  - Suponga que se tiene un conjunto de **D tuplas**. El conjunto de datos es **muestreado D veces**, con reemplazo, resultando en un conjunto de entrenamiento de D muestras.
  - Las tuplas que no formaron parte del conjunto de entrenamiento forman el conjunto de prueba.
  - Aproximadamente el **63.2%** de los datos originales forman el **conjunto de entrenamiento** y el restante, **36.8%** formará el **conjunto de prueba**.
- Cada tupla tiene la probabilidad de **1/d** de ser seleccionada y la probabilidad de que no lo sea es **1-1/d**:
  - Dado que se puede seleccionar d veces, la probabilidad de que una tupla no sea seleccionada durante todo el tiempo es de $\mathbf{(1-1/d)^d}$. Si d es lo suficientemente grande, la probabilidad se aproxima a $\mathbf{e^{-1} = 0.368}$.

...Bootstrap


Data set
$x^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$x^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$x^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
$x^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Bootstrap Data set - D_1^*
$x^{*1(1)}, \ldots, x^{*1(N)}$
Bootstrap Data set - D_2^*
$x^{*2(1)}, \ldots, x^{*2(N)}$
Bootstrap Data set - D_3^*
$x^{*3(1)}, \ldots, x^{*3(N)}$
Bootstrap Data set - D_B^*
$x^{*B(1)}, \ldots, x^{*B(N)}$
Data set
$x^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$x^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$x^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
$x^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Bootstrap Data set - D_1^*
$x^{*1(1)}, \ldots, x^{*1(N)}$
ϕ_1^*
Bootstrap Data set - D_2^*
$x^{*2(1)}, \ldots, x^{*2(N)}$
ϕ_2^*
Bootstrap Data set - D_3^*
$x^{*3(1)}, \ldots, x^{*3(N)}$
ϕ_3^*
Bootstrap Data set - D_B^*
$x^{*B(1)}, \ldots, x^{*B(N)}$
ϕ_B^*

…Bootstrap

- Se puede repetir el procedimiento de muestreo k veces, donde en cada iteración, se utiliza el conjunto de prueba actual para obtener una exactitud estimada del modelo obtenido de la muestra de arranque actual.
- La exactitud total del modelo se estima a partir de:

$$\mathrm{Acc}(M) = \sum_{i=1}^{k} \left(0.632 \times \mathrm{Acc}\left(M_i\right)_{test\_set} + 0.368 \times \mathrm{Acc}\left(M_i\right)_{train\_set} \right)$$

- Este modelo trabaja mejor con conjunto de datos pequeños

¿Cuál es el mejor método?

- Depende de muchos factores:
  - El tamaño del conjunto de datos
  - El paradigma de clasificación utilizado
  - La estabilidad del algoritmo de aprendizaje
  - Las características del problema de clasificación
  - El sesgo / varianza / costo computacional
- **En grandes datasets:**
  - Hold-out puede ser una buena opción: computacionalmente no es tan caro, tiene mayor sesgo pero depende del tamaño del conjunto de datos.
- **En datasets pequeños:**
  - Validación cruzada repetida
  - Bootstrap 0.632
  - Pueden no ser informativos.