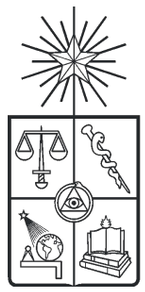


CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Clasificación

Framework y Evaluación

Felipe Bravo
(Basado en una versión previa de Bárbara Poblete)

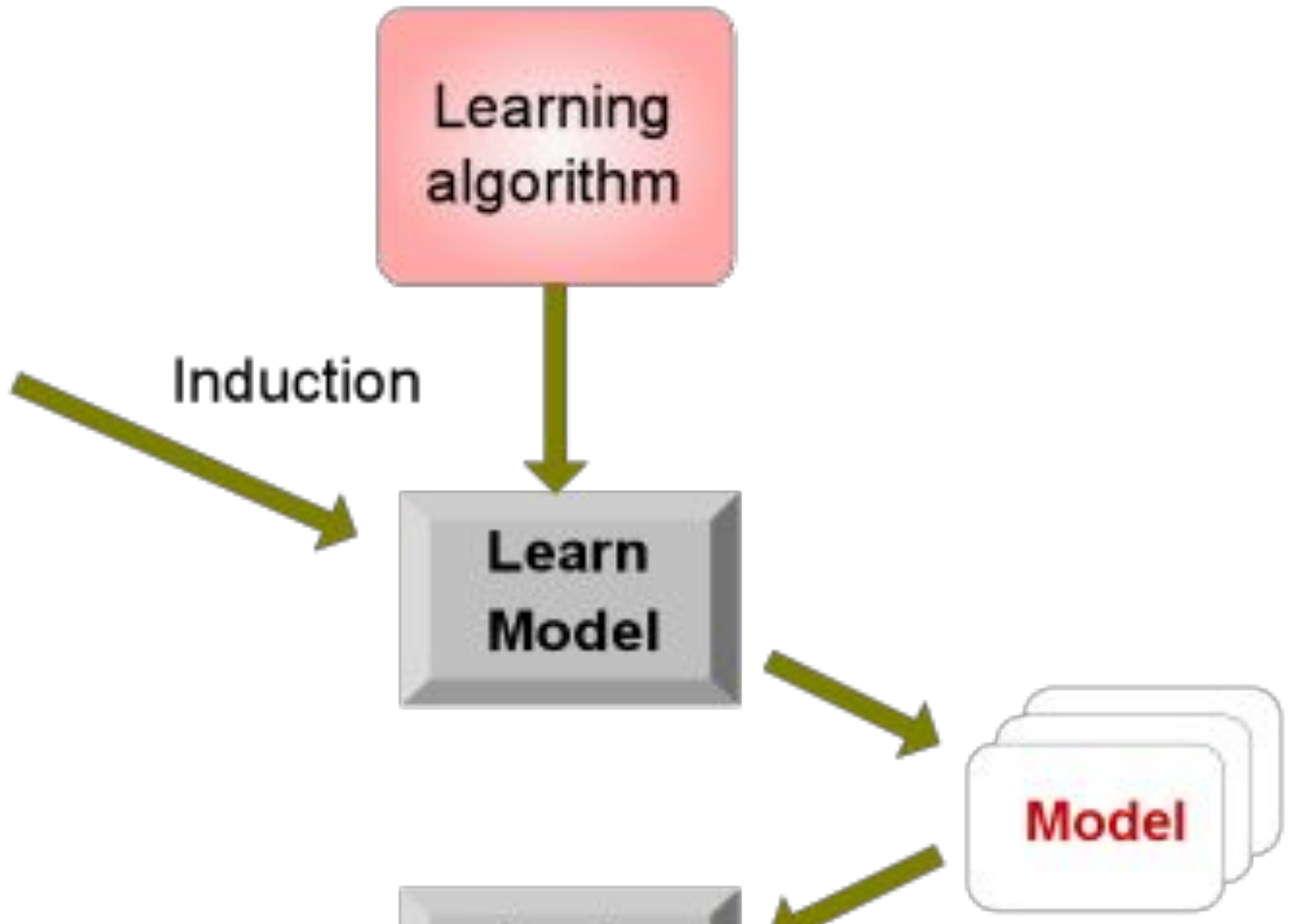
Proceso de Clasificación

1. Datos de entrenamiento etiquetados
2. Entrenar un algoritmo de clasificación.
3. Evaluar en un dataset de validación.
4. Poner el modelo de clasificación en producción.

Proceso de Clasificación

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Tarea de mapear set *x* a una clase *y*

En **machine learning**, a la clasificación se le considera como un enfoque de **aprendizaje supervisado**, pues requiere datos etiquetados.

Machine learning vs Data Mining

- Cuando hacemos clasificación en Machine Learning queremos automatizar una tarea (e.g., reconocer rostros en imágenes).
- Cuando hacemos clasificación en Data Mining queremos encontrar un patrón en los datos (i.e., queremos entender cómo se relaciona x con y por medio de un modelo predictivo).

¿Cómo saber si un modelo es bueno o no?

- Lo más importante es la **capacidad predictiva** del modelo.
- Pero hacer predicciones correctas sobre los datos de entrenamiento no es suficiente para determinar la capacidad predictiva.
- El modelo construido debe **generalizar**, es decir, debe ser capaz de realizar predicciones correctas en datos distintos a los datos de entrenamiento.
- Otros factores importantes: interpretabilidad, eficiencia.

¿Cómo saber si un modelo es bueno?

1. Resumimos la capacidad predictiva de un modelo mediante **métricas de desempeño** (performance metrics).
2. Las métricas se calculan **contrastando** los valores predichos versus los valores reales de la variable objetivo.
3. Este se hace con datos no usados durante entrenamiento.
4. Diseñamos experimentos en que comparamos las métricas de desempeño para varios modelos distintos y nos quedamos con el mejor.

Performance Metrics
(métricas de desempeño)

- Basadas en contar datos **correcta** e **incorrectamente** clasificados

- Accuracy (Exactitud)

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}.$$

- Error rate (Tasa de error)

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}}$$

Matriz de Confusión

		Clase predicha	
Clase real		clase = +	clase = -
	clase = +	a	b
	clase = -	c	d

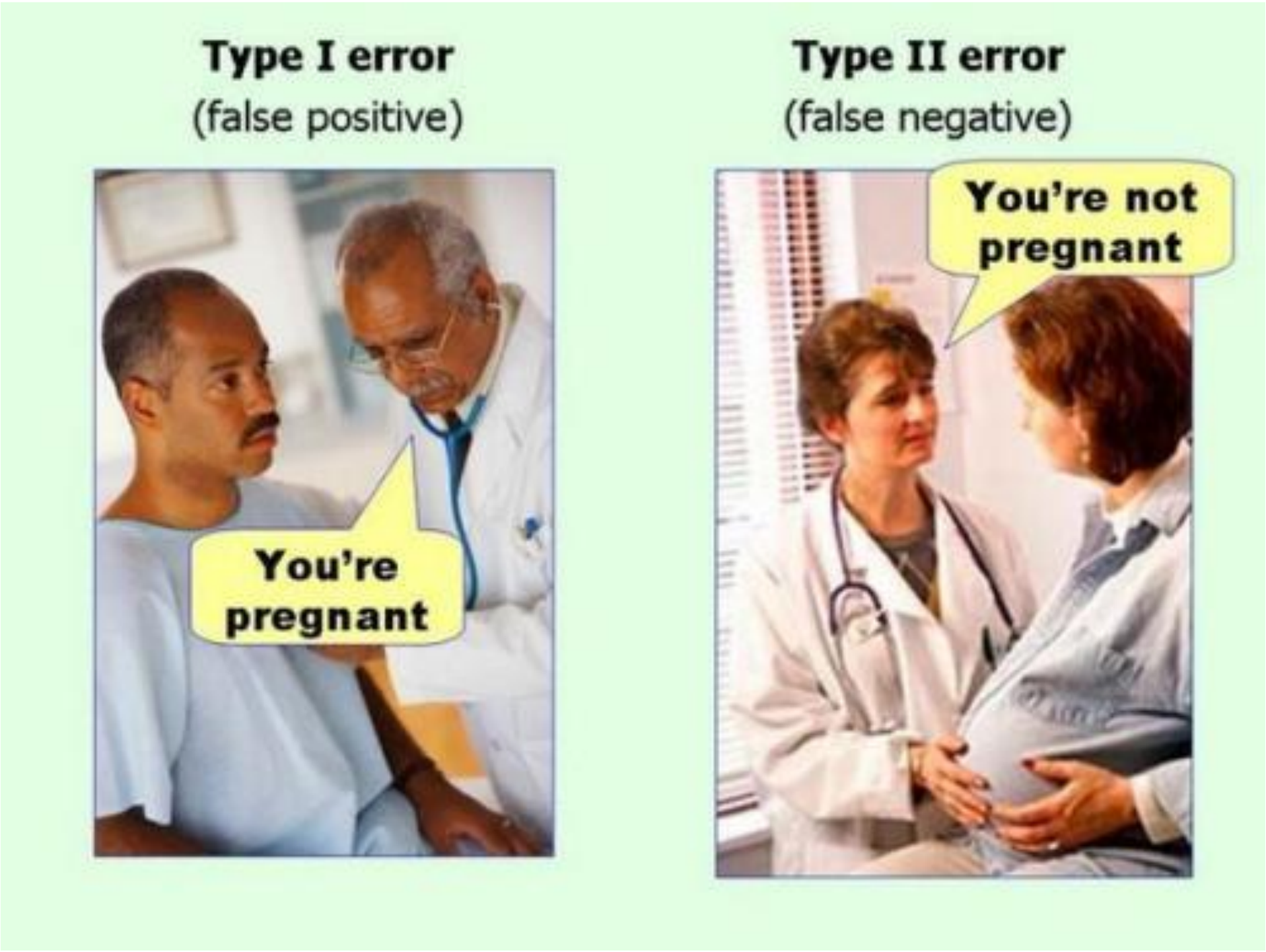
Accuracy (Exactitud)

		Clase predicha	
		clase = +	clase = -
Clase real	clase = +	a **(TP)**	b **(FN)**
	clase = -	c **(FP)**	d **(TN)**

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Error Rate = 1 - Accuracy

Falsos Positivos y Falsos Negativos



Limitaciones del Accuracy

- Consideren un problema de 2-clases
- Num. de ejemplos de la Clase 0 = 9990
- Num. de ejemplos de la Clase 1 = 10
- Accuracy de un clasificador que clasifica todo como Clase 0 = 9990/10000

Accuracy no es una buena métrica cuando tenemos clases desbalancedas.

Precision y Recall

En un problema de clasificación binaria tenemos que escoger cual es la clase positiva.
Podemos pensar que clasificar algo como “positivo” es equivalente a “seleccionarlo”.

- **Precision**: % de los casos “seleccionados” que son correctos = TP/(TP + FP)
- **Recall**: % of de los casos “positivos” que son “seleccionados” = TP/(TP+FN)
- Existe un trade-off entre **Precision** y **Recall**.

F-measure

- La F-measure combina Precision y Recall mediante un promedio armónico ponderado:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

- La media armónica es conservadora y tiende a estar más cerca del mínimo.
- Generalmente usamos la F1 measure
  - i.e., con $\beta = 1$ (o $\alpha = ½$): $F = 2PR/(P+R)$

F1 ignora TN.

Ejercicio: reincidencia de cáncer

Considere 286 mujeres: 201 no tienen reincidencia de cáncer después de 5 años y 85 sí tienen. Compare los modelos:

- M1: "todas reinciden" vs.
- M2: "ninguna reincide"

- Hacer matrices de confusión, calcular accuracy, precision, recall y F1.

Ejercicio: reincidencia de cáncer

Considere 286 mujeres: 201 no tienen reincidencia de cáncer después de 5 años y 85 sí tienen. Compare los modelos:

M1: Todas reinciden

M1	+	-
+	85	0
-	201	0

Accuracy: 85/286= 0.3
Precision: 85/286= 0.3
Recall: 1
F1: 2*0.3/(0.3+1)=0.46

M2: Ninguna reincide

M2	+	-
+	0	85
-	0	201

Accuracy: 201/286 = 0.7
Precision: 0/0 = undef
Recall: 0/85=0
F1: undef

Matriz de Costo

A veces yo se cuales errores son más costosos y cuales aciertos son más valiosos.

		Clase predicha	
	C(i|j)	clase = +	clase = -
Clase real	clase = +	C(+|+)	C(-|+)
	clase = -	C(+|-)	C(-|-)

C(i|j): Costo de clasificar un objeto como clase j dado que es clase i

Calculando el costo de la clasificación

A mayor costo peor el modelo.

Matrix Costo	Clase predicha		
Clase real	C(i|j)	+	-
	+	-1	100
	-	1	0

Modelo M1	Clase predicha		
Clase real		+	-
	+	150	40
	-	60	250

Accuracy(M1) = 0.8
C(M1) = -1*150+100*40+1*60+0*250 = 3910

Modelo M2	Clase predicha		
Clase real		+	-
	+	250	45
	-	5	200

Accuracy(M2) = 0.9
C(M2) = -1*250+100*45+1*5+0*200 = 4255

Clasificación Multi-clase

Cuando tenemos k etiquetas, la matriz de confusión es una matriz de k X k.

Docs in test set	Assigned UK	Assigned poultry	Assigned wheat	Assigned coffee	Assigned interest	Assigned trade
True UK	95	1	13	0	1	0
True poultry	0	1	0	0	0	0
True wheat	10	90	0	1	0	0
True coffee	0	0	0	34	3	7
True interest	-	1	2	13	26	5
True trade	0	0	2	14	5	10

Métricas de desempeño por clase

Recall: Fracción de ejemplos de la clase *i* correctamente clasificado.

$$\frac{c_{ii}}{\sum_j c_{ij}}$$

Precision: Fracción de ejemplos asignados a la clase *i* que realmente son de la clase i.

$$\frac{c_{ii}}{\sum_j c_{ji}}$$

Accuracy: (1 - error rate)

Fracción total de ejemplos correctamente clasificados.

$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

Micro- vs. Macro-Averaging

- Si tenemos más de una clase, ¿cómo combinamos múltiples métricas de desempeño en un solo valor?
- **Macroaveraging**: computar métrica para cada clase y luego promediar.
- **Microaveraging**: crear matriz de confusión binaria para cada clase, combinar las matrices y luego evaluar.

Micro- vs. Macro-Averaging: Ejemplo clasificación de Spam

		gold labels urgent	normal	spam	
system output	urgent	8	10	1	$\textbf{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\textbf{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\textbf{precision}_s = \frac{200}{3+30+200}$
		$\textbf{recall}_u = \frac{8}{8+5+3}$	$\textbf{recall}_n = \frac{60}{10+60+30}$	$\textbf{recall}_s = \frac{200}{1+50+200}$	

Figure 4.5 Confusion matrix for a three-class categorization task, showing for each pair of classes (c_1, c_2), how many documents from c_1 were (in)correctly assigned to c_2

- Fuente: https://web.stanford.edu/~jurafsky/slp3/4.pdf

Micro- vs. Macro-Averaging: Ejemplo clasificación de Spam

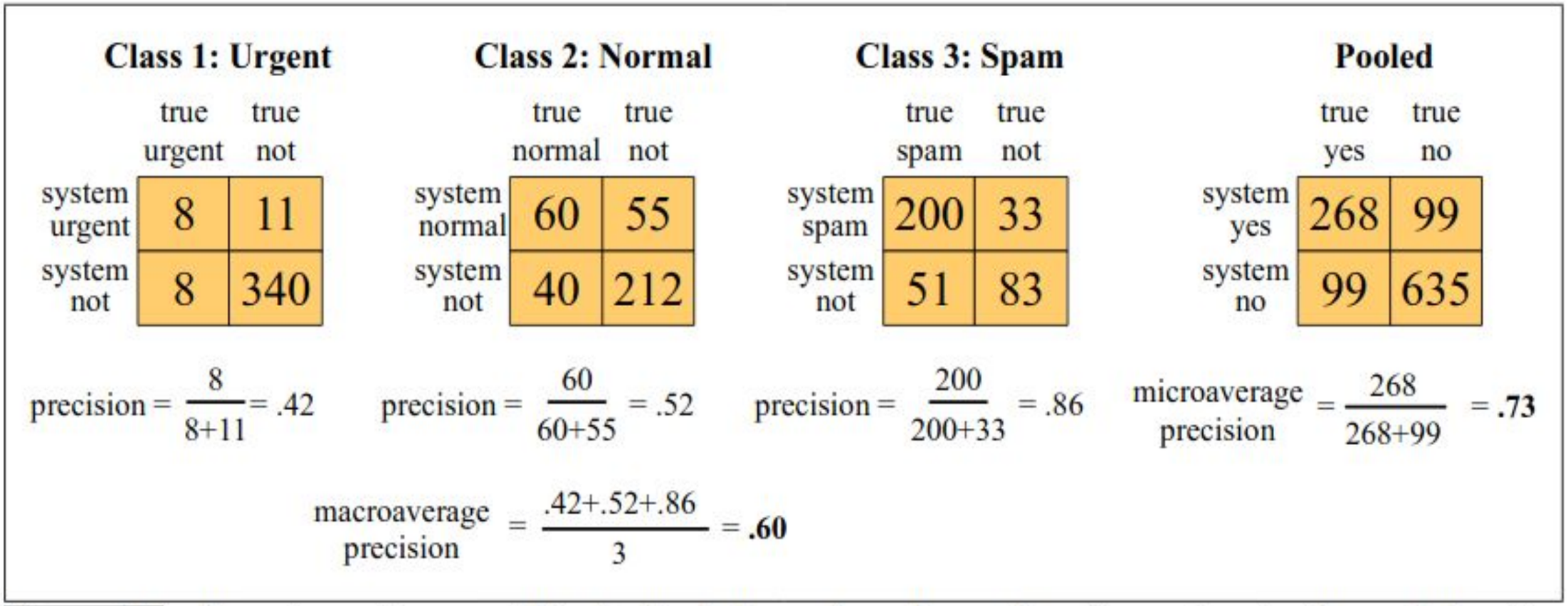


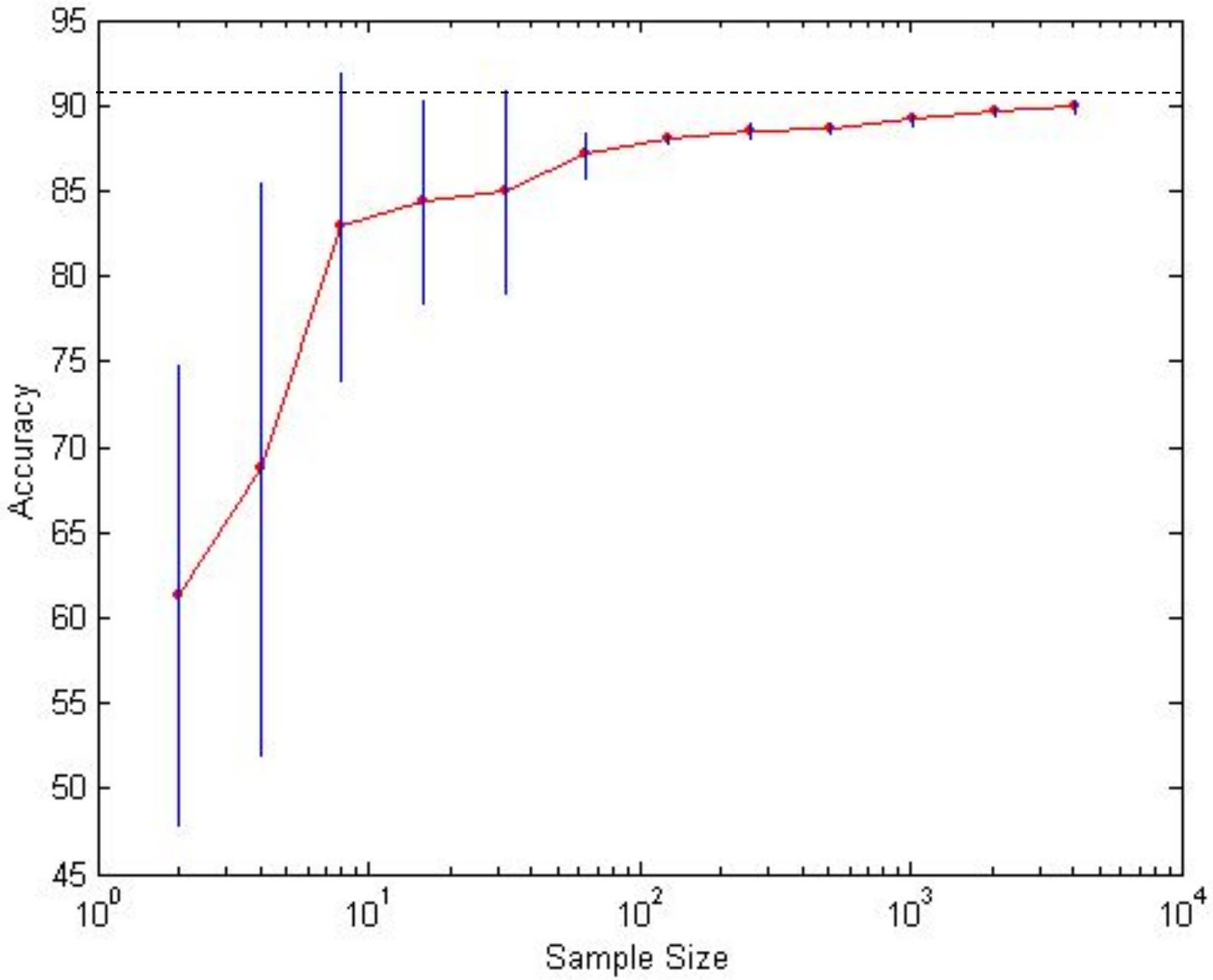
Figure 4.6 Separate contingency tables for the 3 classes from the previous figure, showing the pooled contingency table and the microaveraged and macroaveraged precision.

- Los micro-promedios son dominados por las clases más frecuentes.
- Los macro-promedios pueden sobre-representar a clases minoritarias.

Evaluación del desempeño del modelo

- El desempeño de un modelo puede depender de factores diferentes al algoritmo de aprendizaje
  - Distribución de las clases
  - Costo de clasificaciones erróneas
  - Tamaño de los datos de entrenamiento y test

Curva de aprendizaje



Métodos para evaluar el desempeño de un modelo

La idea es estimar la capacidad de generalización de modelo, evaluándolo en datos distintos a los de entrenamiento.

1) Holdout
2) Random subsampling (subsampleo aleatorio)
3) Cross validation (validación cruzada)

Holdout

- Particionamos los datos etiquetados en una partición de training y otra de testing.
  - Usualmente usamos 2/3 para entrenamiento y 1/3 para evaluación.
- Limitaciones:
  - La evaluación puede variar mucho según las particiones escogidas.
  - Training muy pequeño => modelo sesgado.
  - Testing muy pequeño => accuracy poco confiable.

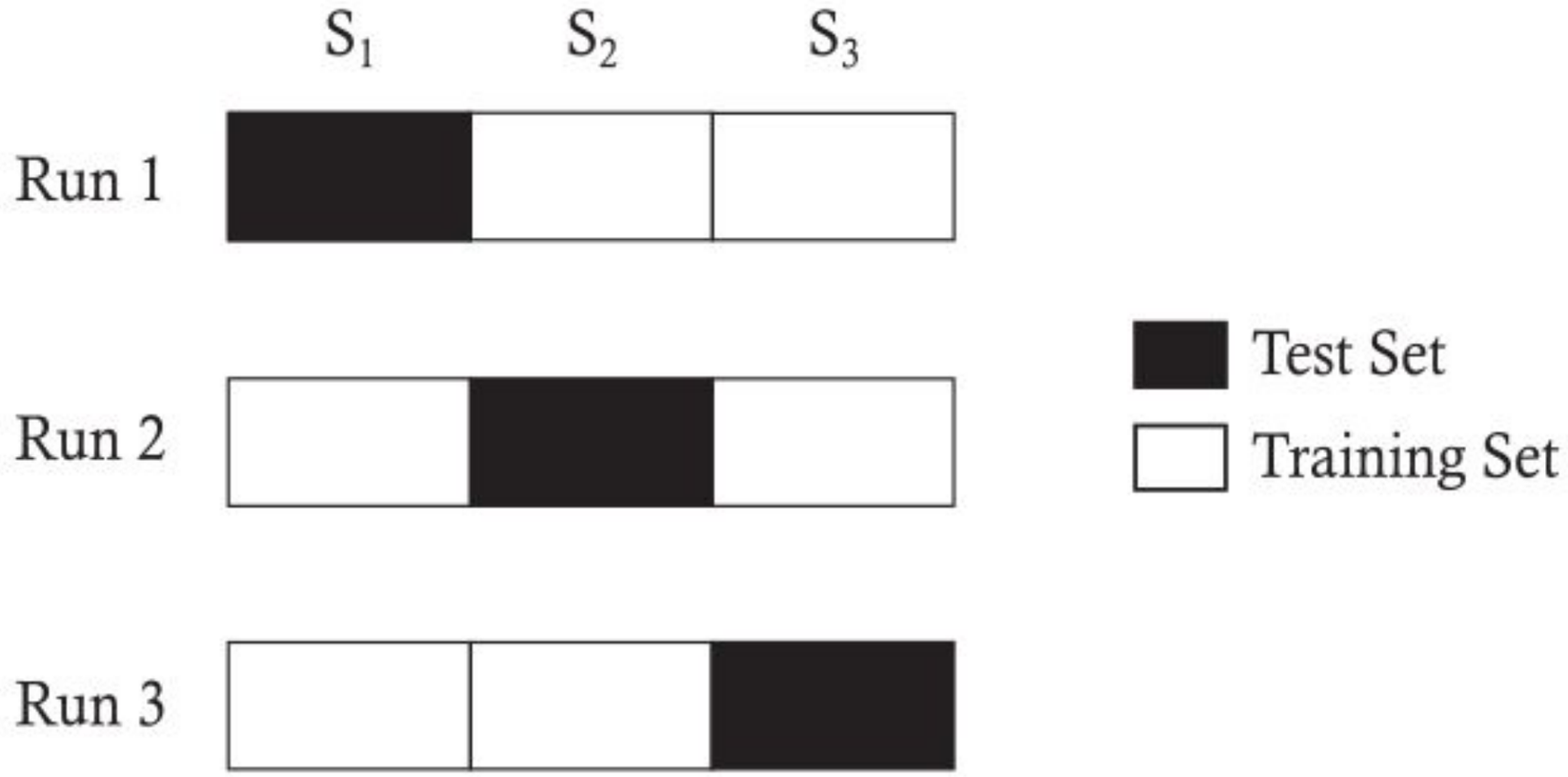
Random Subsampling

- Se repite el método holdout varias veces sobre varias particiones de training y testing.
- Permite obtener una distribución de los errores o medidas de desempeño.
- Limitaciones:
  - Puede que algunos datos nunca se usen para entrenar.
  - Puede que algunos datos nunca se usen para evaluar.

Validación cruzada (cross-validation)

- Se particiona el dataset en en k conjuntos disjuntos o folds (manteniendo distribución de las clases en cada fold).
- Para cada partición i:
  - Juntar todas las k-1 particiones restantes y entrenar el modelo sobre esos datos.
  - Evaluar el modelo en la partición i.
- El error total se calcula sumando los errores hechos en cada fold de testing.
- Estamos entrenando el modelo k veces.
- Variante: leave-one-out (k=n)

Ejemplo: 3-fold cross-validation

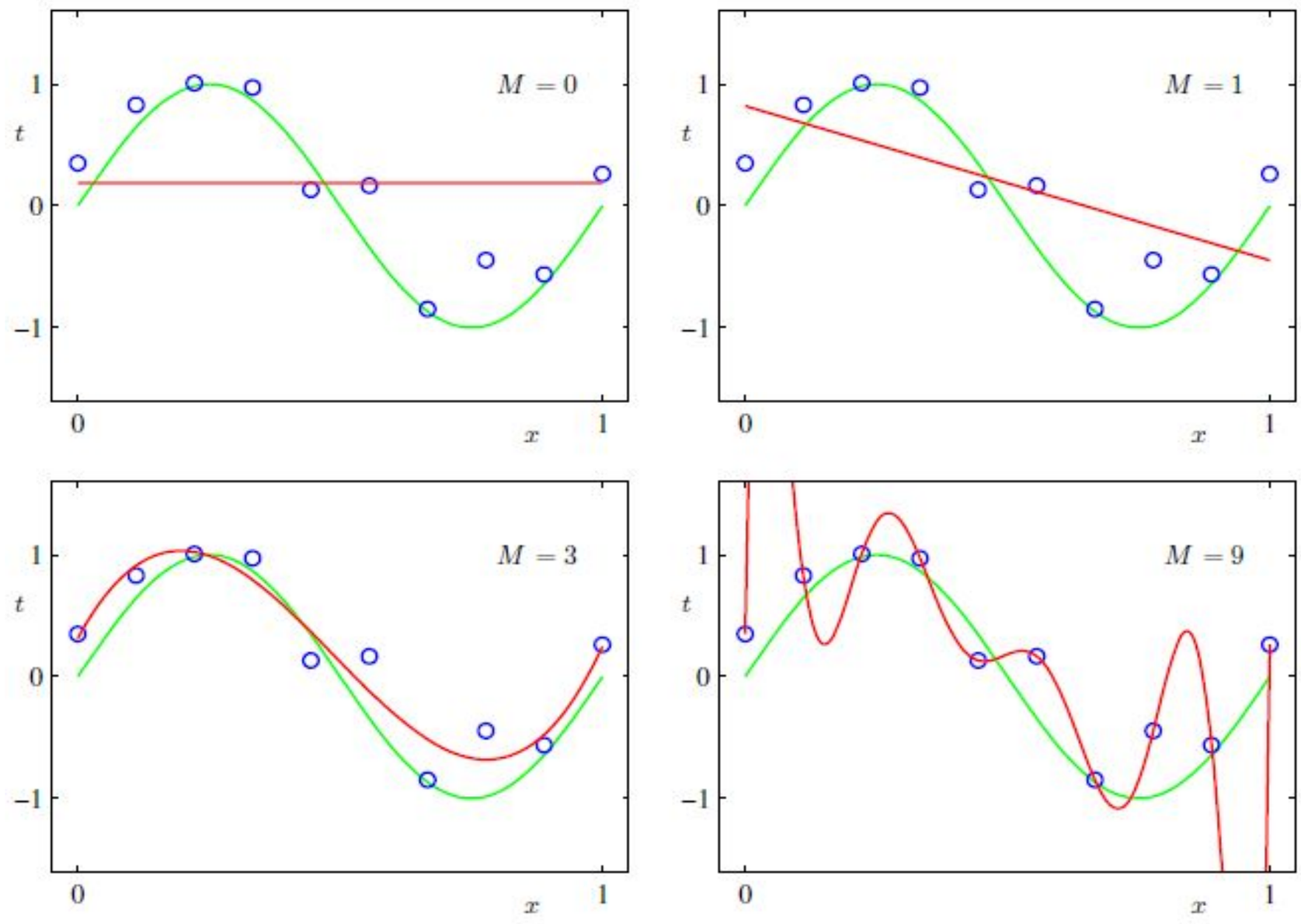


Problemas prácticos en la clasificación

- Errores de entrenamiento (malos resultados sobre los datos de entrenamiento): esto ocurre cuando el clasificador no tiene capacidad de aprender el patrón.

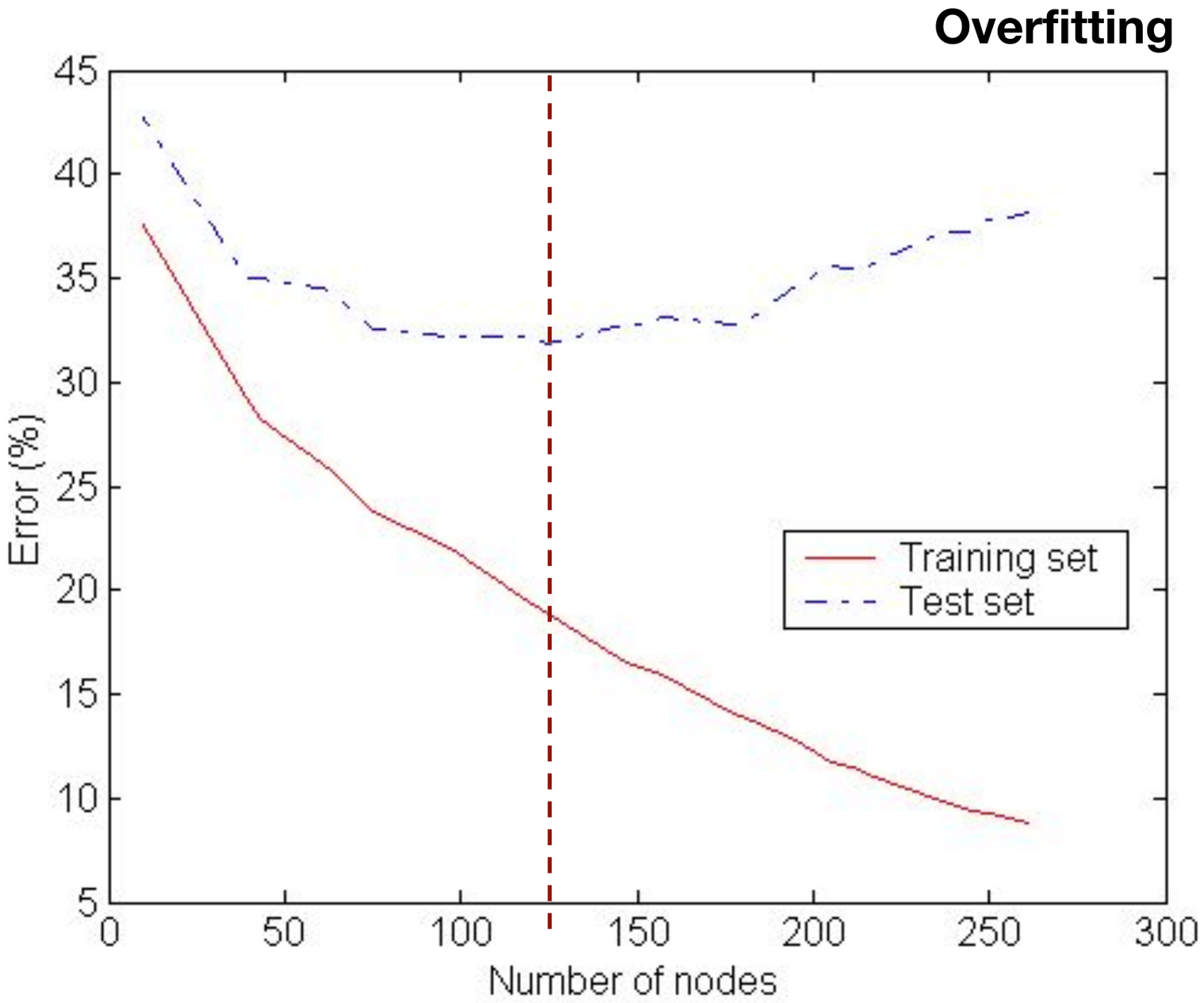
- Errores de generalización (malos resultados sobre datos nuevos): esto ocurre cuando el modelo se hace demasiado específico a los datos de entrenamiento.

Overfitting y Underfitting usando polinomios para un problema de regresión

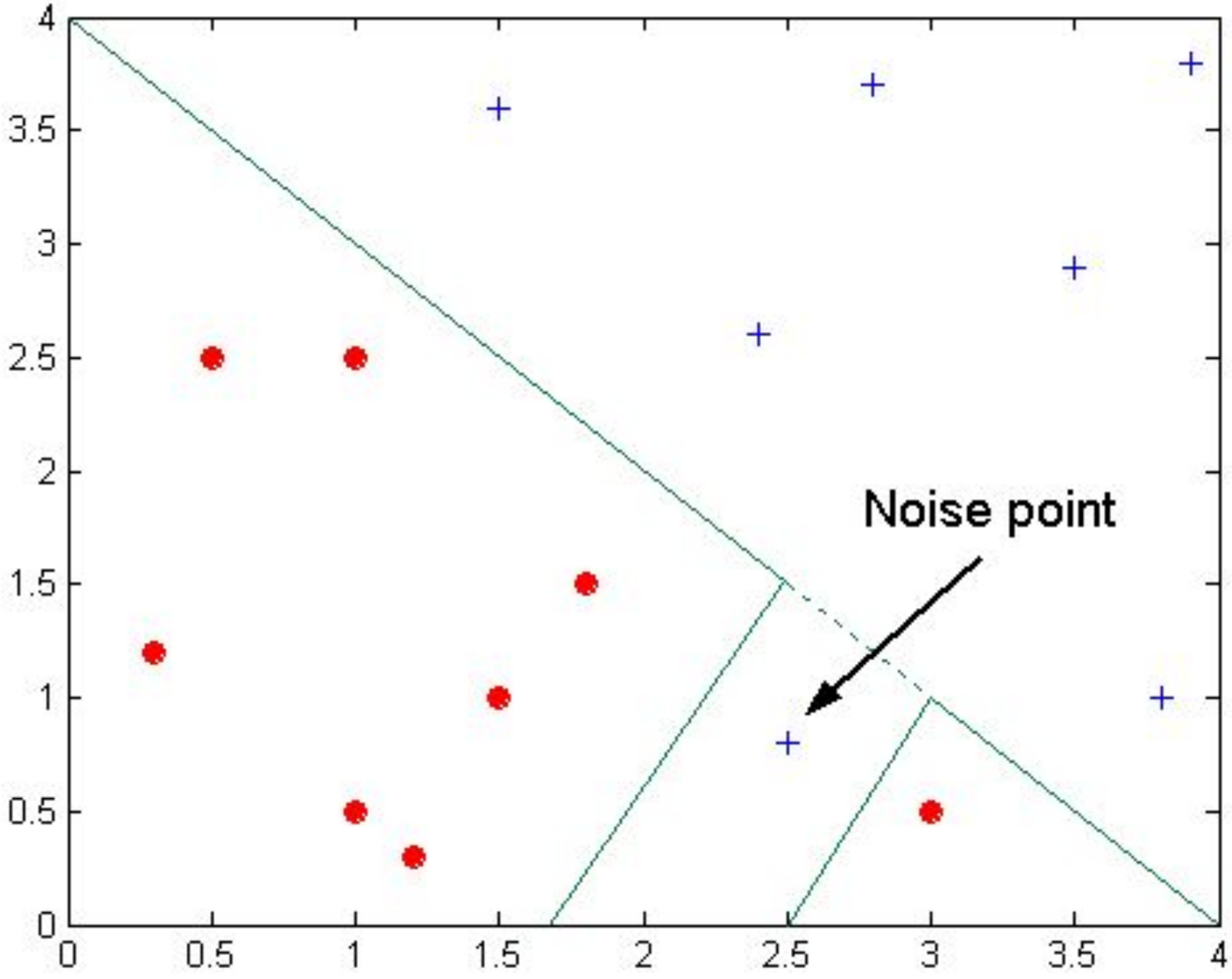


M es el orden del polinomio.

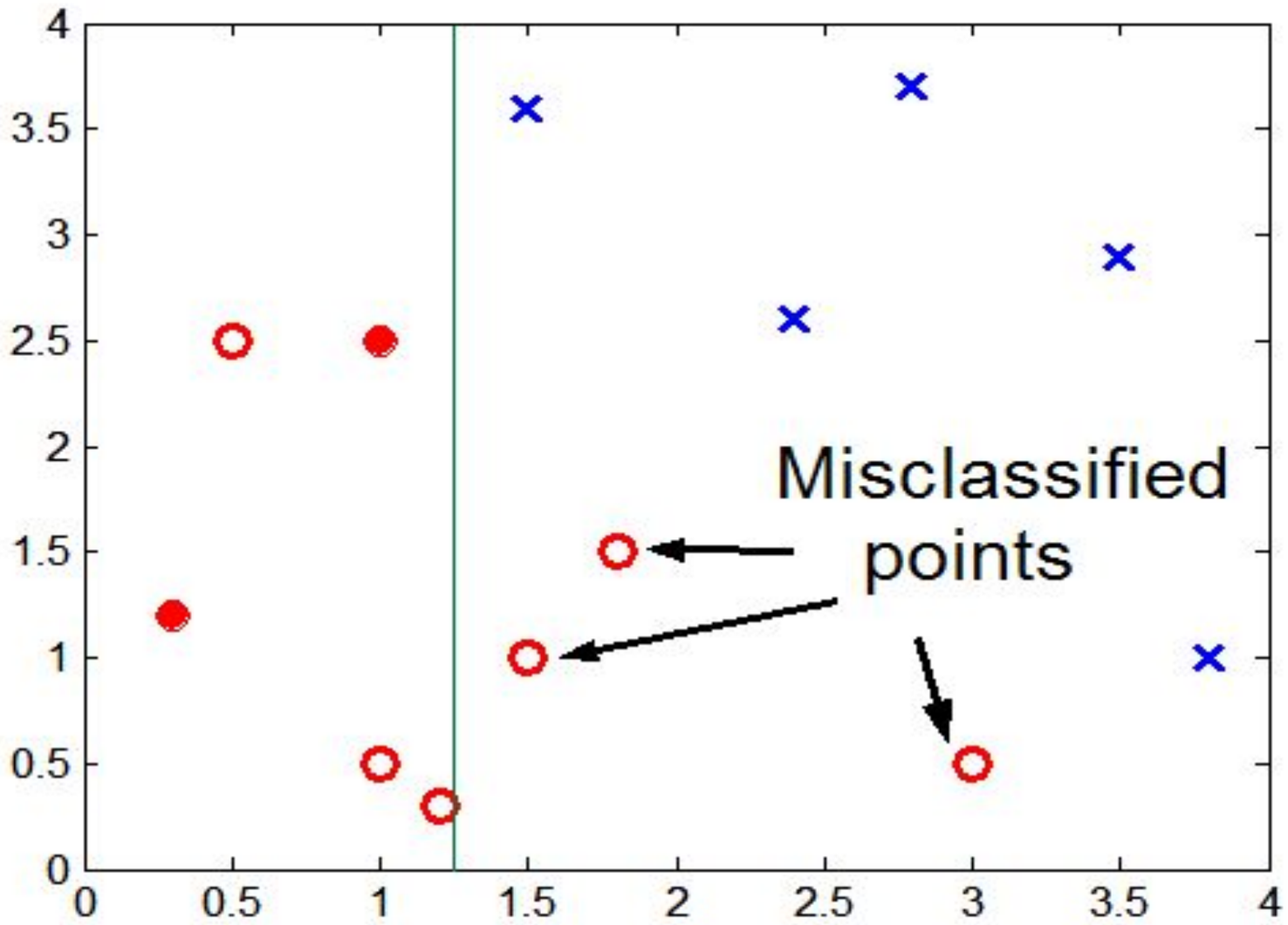
Underfitting y Overfitting



Overfitting por ruido



Overfitting por ejemplos insuficientes



Notas sobre el Overfitting

- El overfitting es un reflejo de un modelo más complejo que lo necesario.
- El error de entrenamiento no es un indicador confiable de cómo se desempeñaría el modelo sobre datos nuevos.

Curva ROC
Receiver Operating Characteristic Curve

- De manera similar que el trade-off entre Precision y Recall también existe un tradeoff entre la tasa de verdaderos positivos y la tasa de falsos positivos.

TP Rate: TP / (TP +FN)

FP Rate: FP / (FP + TN)

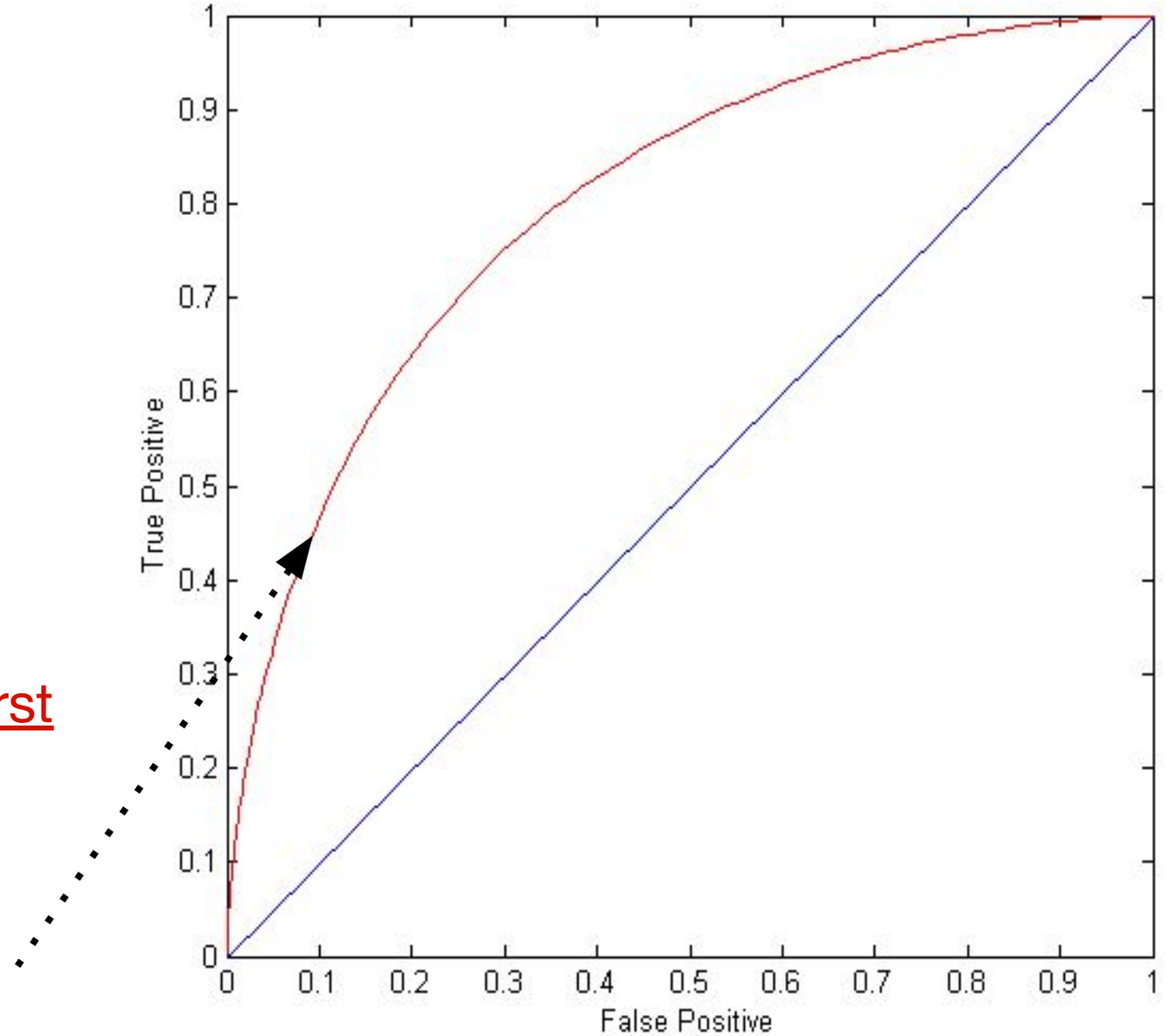
- La curva ROC se construye graficando TP Rate vs FP Rate para varios umbrales de clasificación de un clasificador probabilístico (ej: regresión logística, naive Bayes).

Curva ROC
Receiver Operating Characteristic Curve

- Entre mayor sea el área bajo la curva mejor es el modelo.

- El área bajo la curva ROC se conoce como AUC y es una métrica ampliamente usada.

- Un tutorial recomendado: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5



Ejemplo: Curva ROC sobre predicciones probabilísticas

		predicted				
actual	$Pr(class = yes)$	$\lambda > 0.9$	$\lambda > 0.7$	$\lambda > 0.3$	$\lambda > 0.2$	$\lambda > 0.0$
yes	0.9	no	yes	yes	yes	yes
no	0.7	no	no	yes	yes	yes
yes	0.3	no	no	no	yes	yes
no	0.2	no	no	no	no	yes
TN		2	2	1	1	0
FN		2	1	1	0	0
FP		0	0	1	1	2
TP		0	1	1	2	2
TP Rate		0.0	0.5	0.5	1.0	1.0
FP Rate		0.0	0.0	0.5	0.5	1.0
Precision		NaN	1.0	0.5	0.66	0.5
Recall		0	0.5	0.5	1.0	1.0

Table 5: Performance metrics for different threshold values.

Ejemplo: Curva ROC sobre predicciones probabilísticas

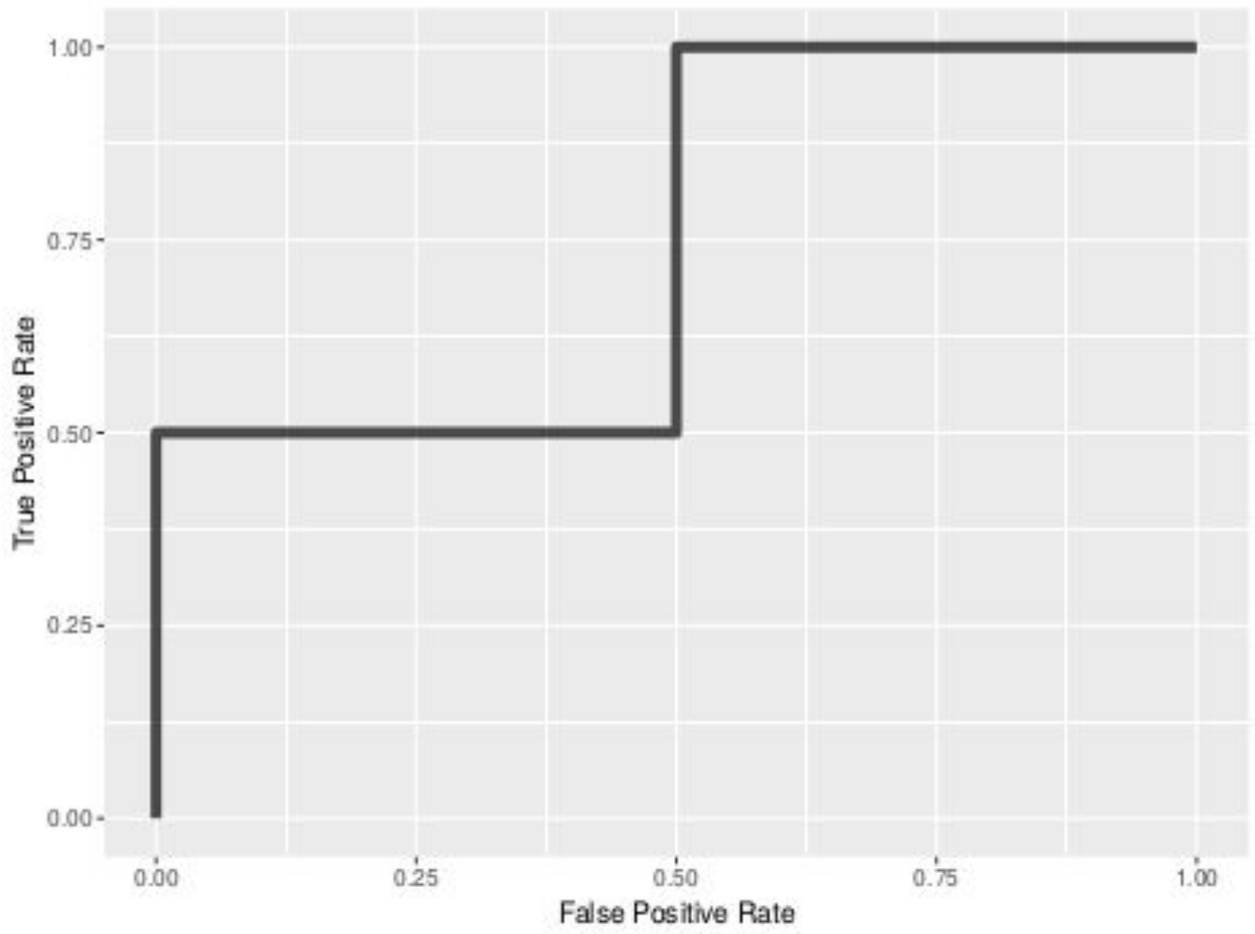


Figure 1: ROC curve.

dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl

/ DCCUCHILE