



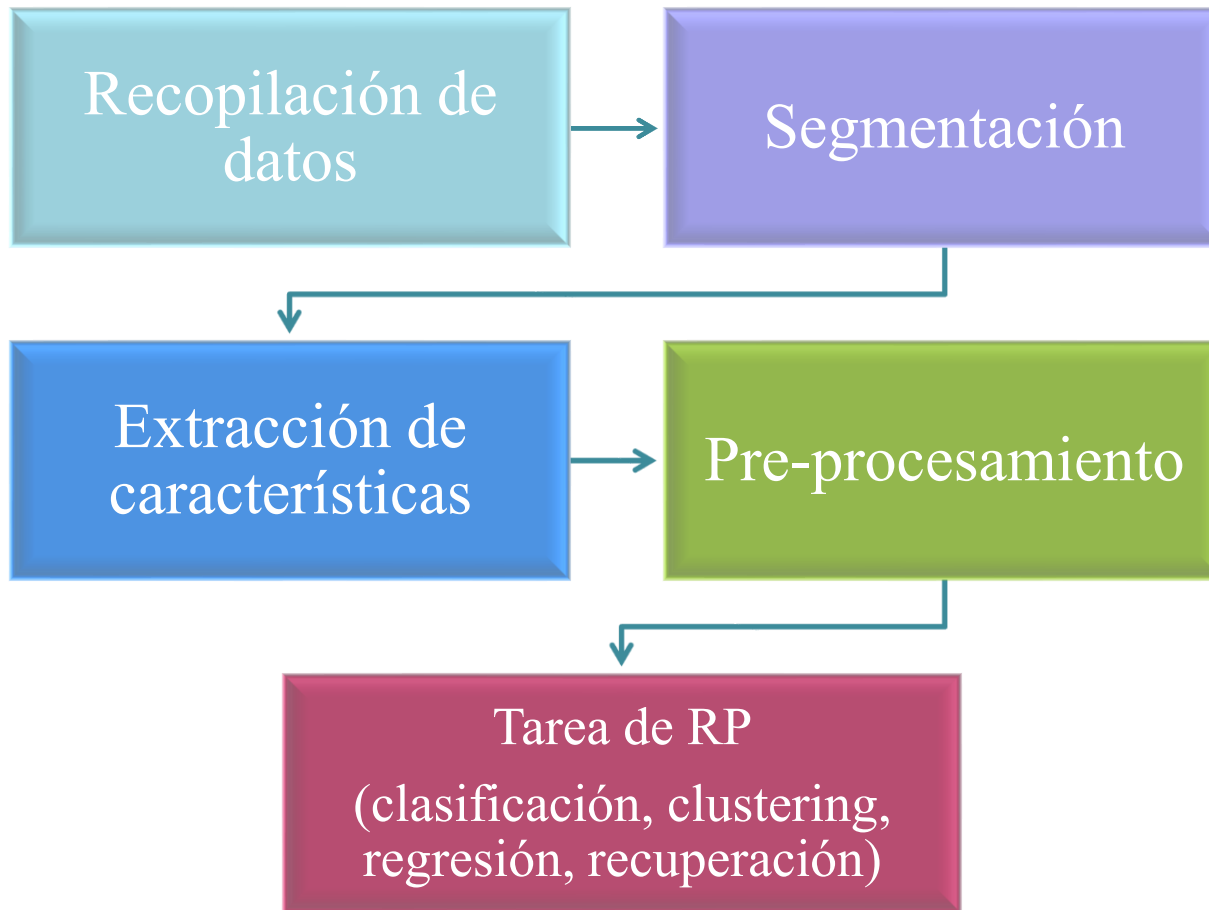
Centro de
INNOVACIÓN y DESARROLLO
Tecnológico en Cómputo



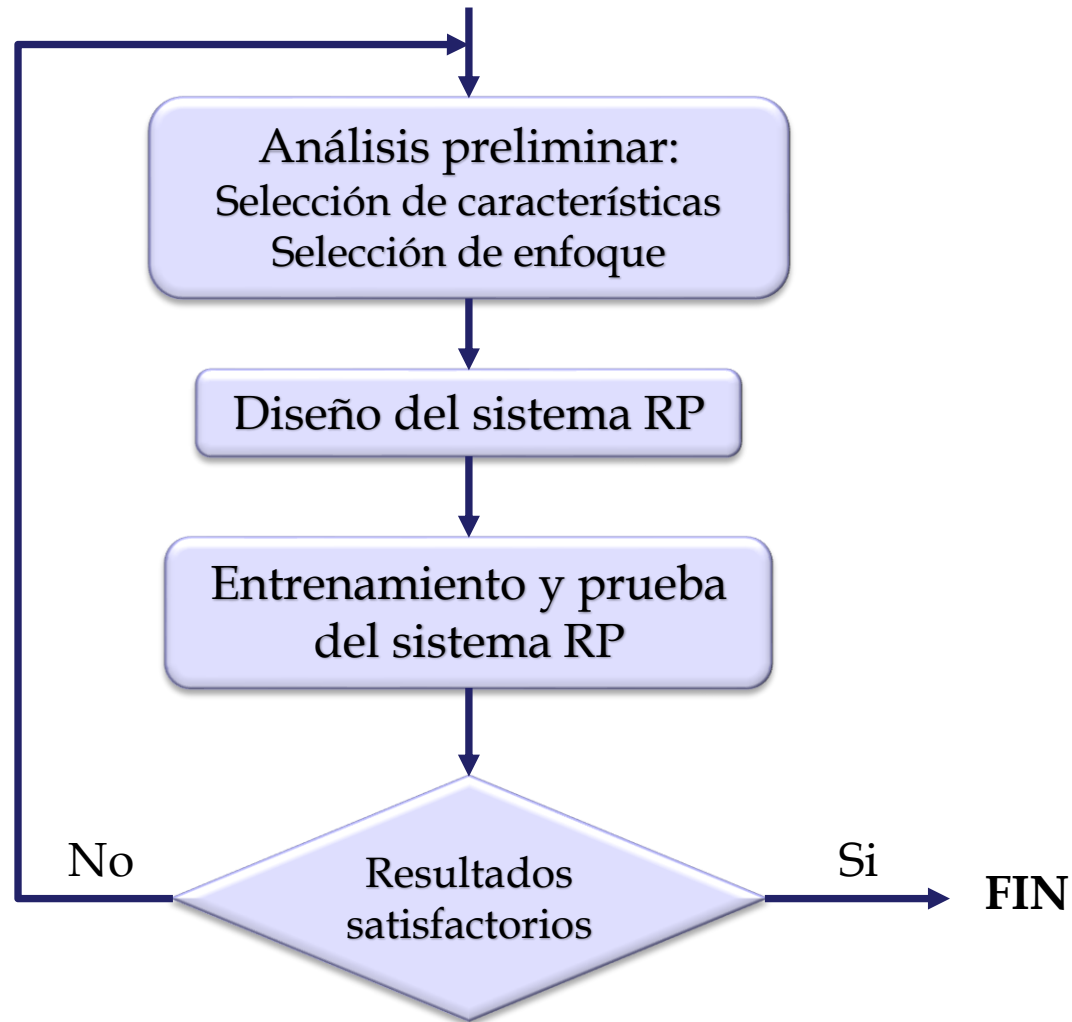
Reconocimiento de Patrones

Lección 3: Métodos de Validación

Etapas en un sistema de RP



Esquema General de un Sistema RP



Métodos de Validación

Asumiendo que se cuenta con un banco, se procede a la selección de un método de validación para el sistema RP.

Si contásemos con un número infinito de patrones de cada clase, tomaríamos un subconjunto previamente clasificado para entrenar al clasificador (aprendizaje supervisado) y probaríamos con los patrones que están sin clasificar.



Métodos de Validación

Una partición del conjunto X es una familia de B subconjuntos disjuntos no vacíos de X , cuya unión es X . Es decir:

$$B = \{X_i : i \in I\}$$

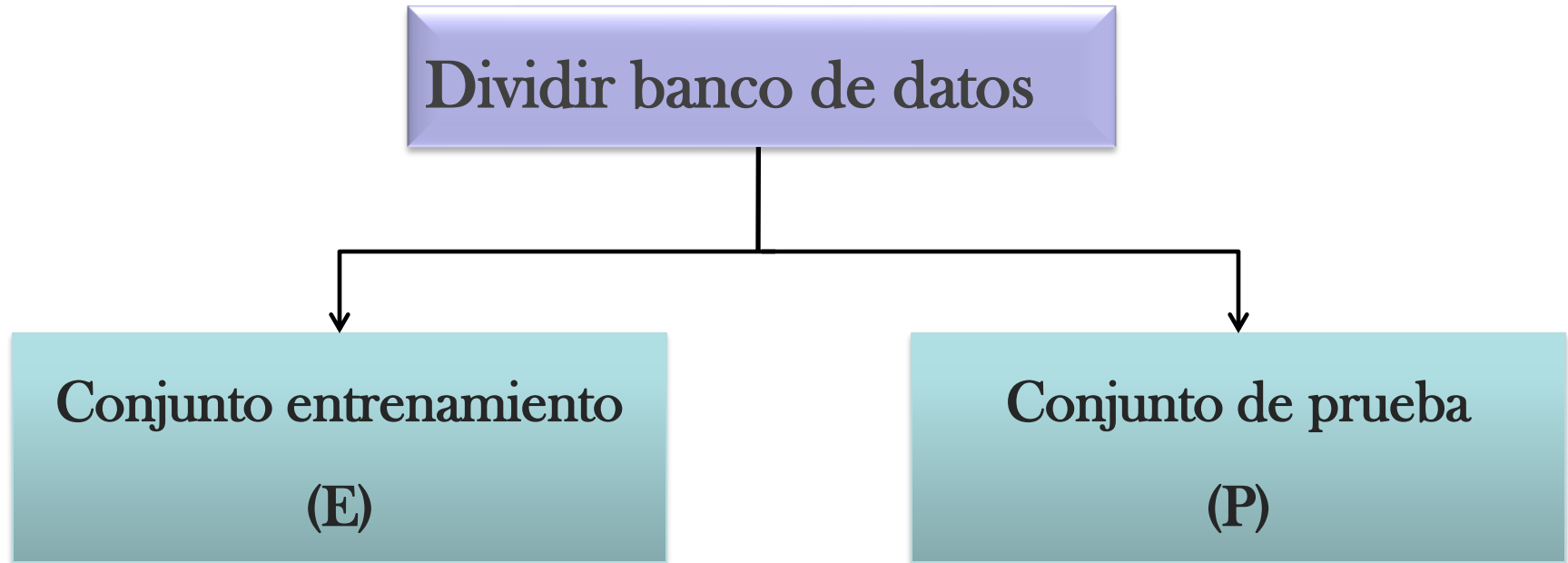
donde se cumple:

$$\forall i \in I, X_i \subseteq X \text{ y } X_i \neq \emptyset$$

$$\forall (X_i, X_j), i \neq j \text{ } X_i \cap X_j = \emptyset$$

$$\bigcup_{i \in I} X_i = X$$

Métodos de Validación



Los sub-conjuntos E y P forman una partición del conjunto original de datos.



Métodos de Validación

Proceso de Validación

Dividir el Banco de Datos

Factor de Olvido
Hold-Out
K-Fold Cross-Validation
Leave One-Out
Bootstrap

Cuantificar desempeño

Rendimiento
Error
Sensibilidad
Especificidad

Factor de Olvido

En ingles el término utilizado es *Rebsustitución Error*.

Una forma de validar un sistema RP es evaluando su capacidad para clasificar los patrones del conjunto de entrenamiento. Es decir que el conjunto de prueba es el mismo que el conjunto de entrenamiento ($E=P=X$).

“The error rate on the training data is called the resubstitution error because it is calculated by resubstituting the training instances into a classifier that was constructed from them. Although it is not a reliable predictor of the true error rate on new data, it is nevertheless often useful to know.” [5]



Métodos de Validación (Factor de Olvido)

- ➔ Realizar la fase de entrenamiento del sistema RP con todos los patrones del banco de datos (BD).
- ➔ Realizar la fase de prueba del sistema RP con todos los patrones del banco de datos.
- ➔ Calcular el Factor de Olvido (FO)

$$FO = \frac{\text{Errores}}{\# \text{ patrones del BD}}$$



Métodos de Validación

¿Es la tasa de error sobre el conjunto de entrenamiento una buena medida del desempeño de un clasificador?

La respuesta es NO. Dado que el clasificador ha sido creado o entrenado con esos datos, cualquier estimación realizada con ellos sería extremadamente optimista [5].

Para determinar el desempeño de un clasificador sobre datos nuevos es necesario medir la tasa de error sobre datos que no formaron parte en la creación del mismo [5].



Métodos de Validación (*Hold-out*)

1. Dividir el banco de datos en los conjunto E y P, de la siguiente forma, por cada clase i :

$$|E| = r * N$$

$$|P| = N - (r * N)$$

Valores típicos de r : 0.7, 0.8.

2. Ejecutar algoritmo de RP con los conjuntos E y P obtenidos.
3. Repetir el mayor número de veces posibles.



Métodos de Validación (*Hold-out* estratificado)

1. Separar el conjunto de datos por clase.
2. Dividir cada clase para crear los conjunto E y P, de la siguiente forma, por cada clase i :

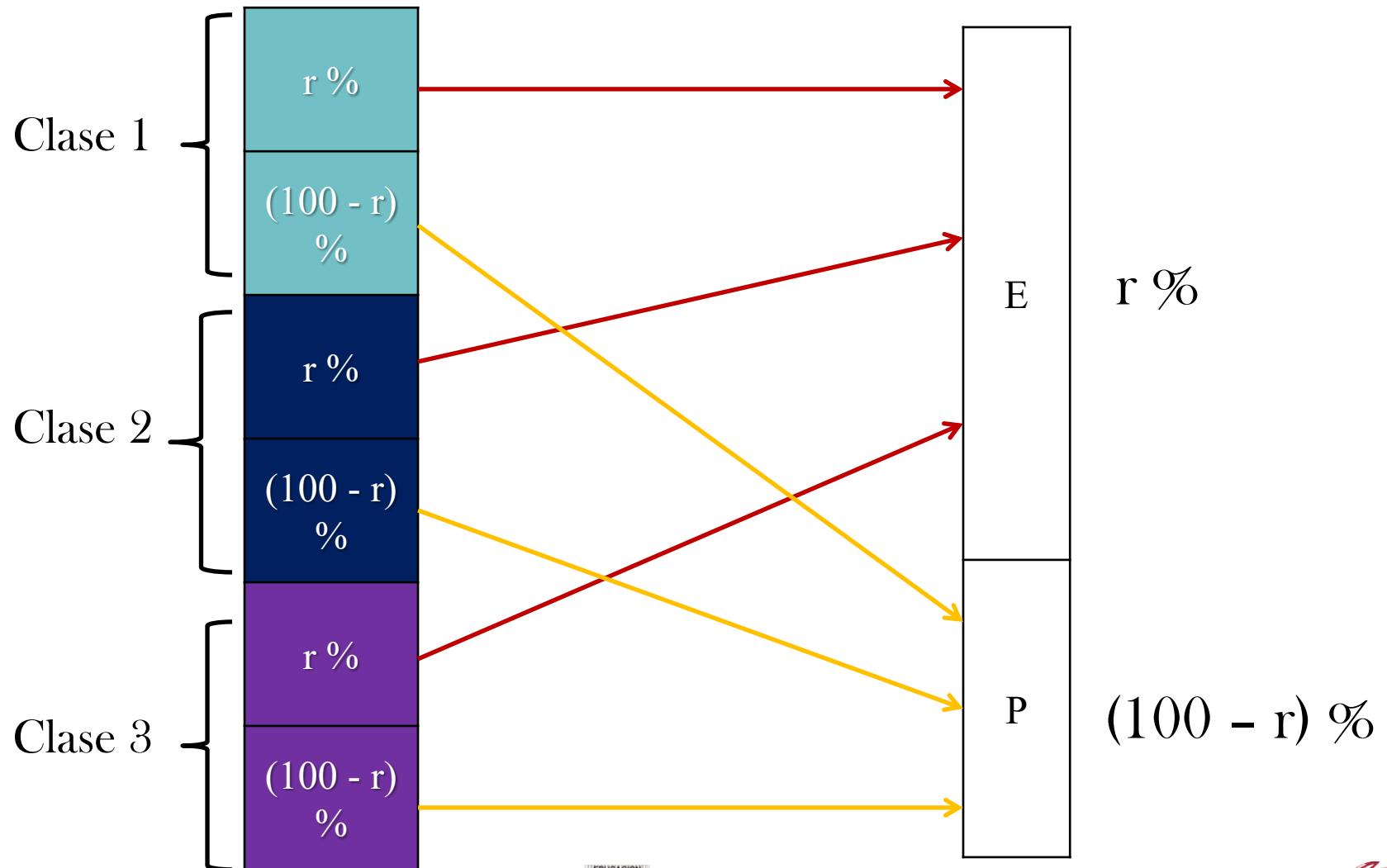
$$|E| = \sum_{i \in c} r * N(\omega_i)$$

$$|P| = \sum_{i \in c} N(\omega_i) - [r * N(\omega_i)] \quad \text{Valores típicos de } r: 0.7, 0.8.$$

3. Ejecutar algoritmo de RP con los conjuntos E y P obtenidos.
4. Repetir el mayor número de veces posibles (valor recomendado: 100).



Métodos de Validación (*Hold-out estratificado*)



Métodos de Validación (*Hold-out* estratificado)

El cálculo del desempeño con el método *Hold-out*:

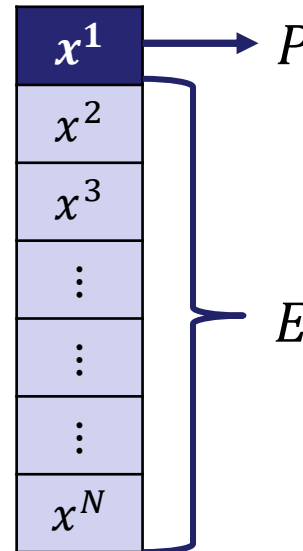
1) El desempeño se evalúa dividiendo el número de acierto obtenidos en el conjunto de prueba.

$$accuracy = \frac{\#aciertos}{|P|}$$



Métodos de Validación (*Leave one-out*)

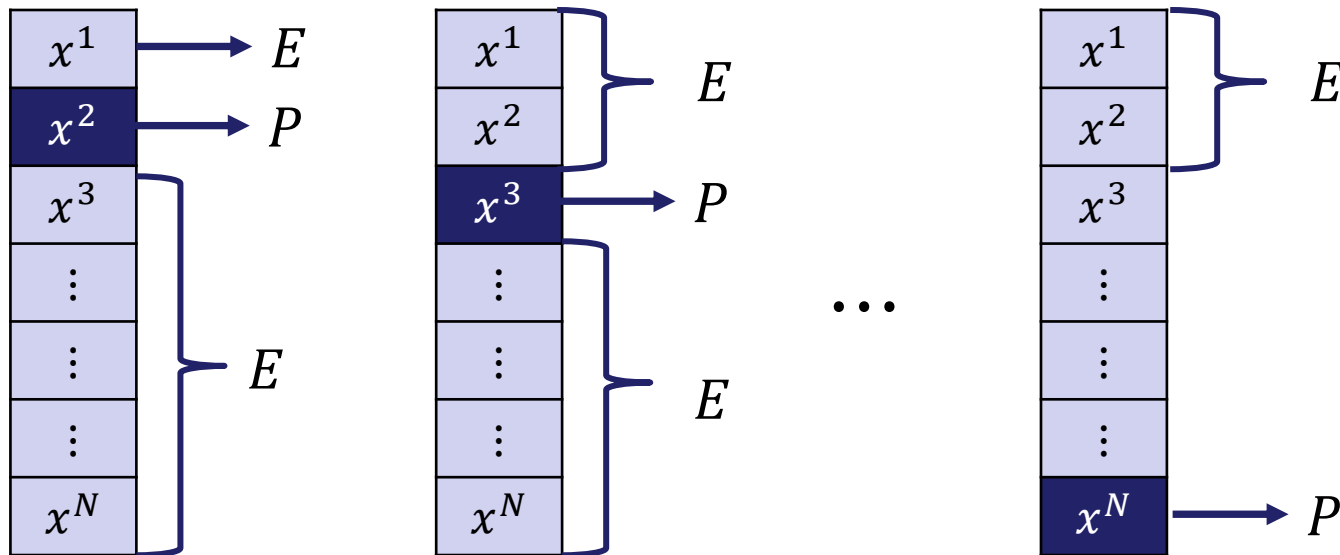
1. Seleccionar un patrón del conjunto de datos x^1 para formar el conjunto de prueba, es decir $P = x^1$. El resto de patrones, es decir $N - 1$, será el conjunto de entrenamiento.



2. Ejecutar el algoritmo RP con los conjuntos E y P obtenidos.

Métodos de Validación (*Leave one-out*)

3. Repetir el proceso N veces variando el patrón $x^i \forall i \in N$



4. Por cada iteración se almacena si el patrón fue clasificado correctamente o no; al final, el rendimiento estará determinado por el número total de patrones clasificados correctamente.

Métodos de Validación (*Leave one-out*)

1. Este es el primer método de validación cruzada de modelos.
2. Leave-one-out cross-validation (LOOCV) surgió como respuesta a las desventajas que exhibían los dos métodos en boga a finales de los sesenta, Hold-out y Resubstitution Error, los cuales no son métodos de validación cruzada.
3. Obsérvese que en el procedimiento no se requiere de algún cálculo aleatorio. Por ello, basta ejecutarlo sólo una vez.
4. Lo anterior significa que Leave-one-out cross-validation (LOOCV) es un método determinista.
5. Si el algoritmo del clasificador que se aplica con Leave-one-out cross-validation también es determinista, el resultado será más estable que al usar Hold-out.



Métodos de Validación (*Leave-one-out*)

El cálculo del desempeño con el método *Leave-one-out*:

1) Por cada ejecución del método solo vamos a obtener un resultado (un solo patrón), acierto o error. En cada ejecución ir acumulando los aciertos.

2) Dividir el número de aciertos entre la cardinalidad del BD:

$$accuracy = \frac{\#aciertos}{|N|}$$



Métodos de Validación (*K-fold cross-validation estratificado*)

En la actualidad se identifican varios métodos de validación que la comunidad científica prefiere utilizar en sus publicaciones JCR. los tres modelos de validación más populares entre los autores especializados en CIP y áreas afines son:

- 1) En primer lugar, el método de validación *K-fold cross-validation*, siendo el valor de k más usado 10.
- 2) En segundo lugar, el método de validación *Leave-one-out cross-validation* (LOOCV).
- 3) En tercer lugar, el método de validación *Hold-out*. En este caso no hay una configuración de porcentajes definida de manera contundente; no obstante, entre las más usadas están: 80-20, 70-30 y 75-25.

Métodos de Validación (*K-fold cross-validation estratificado*)

1. Separar el conjunto de datos por clase. Elegir valor de k , el cual puede ser uno los dos más populares (10, 5) o cualquier otro adecuado para el proyecto (*i.e.*, en un BD de 175 patrones, se eligió $k = 7$).
2. Dividir cada clase en k partes y formar los conjunto E y P de la siguiente forma:

$$E = (k - 1) \text{ partes}$$

$$P = 1k$$

3. Cada k partes debe representar de forma proporcional a cada clase.

$$|k| = \sum_{i \in c} N(\omega_i) / k$$

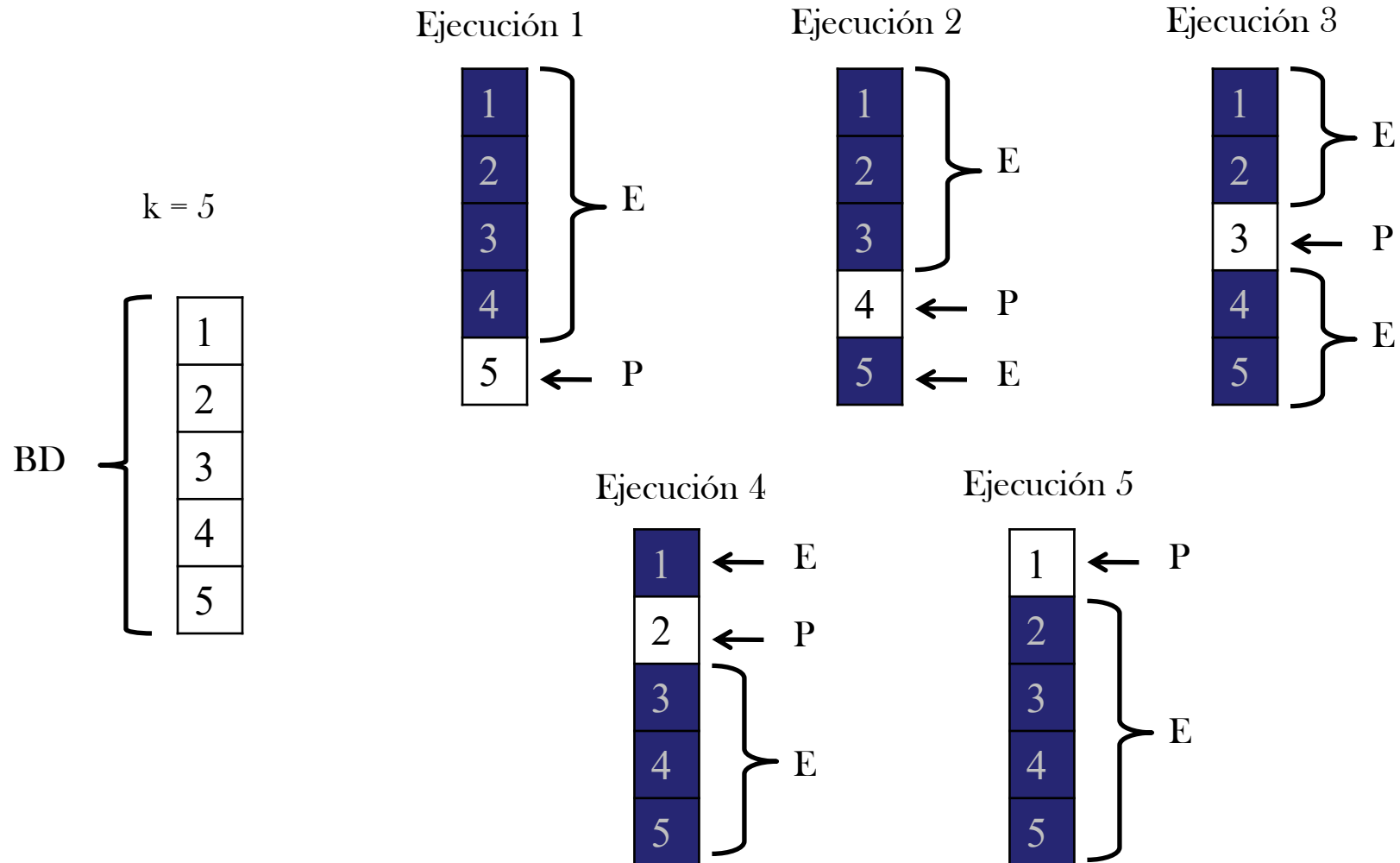


Métodos de Validación (*K-fold cross-validation estratificado*)

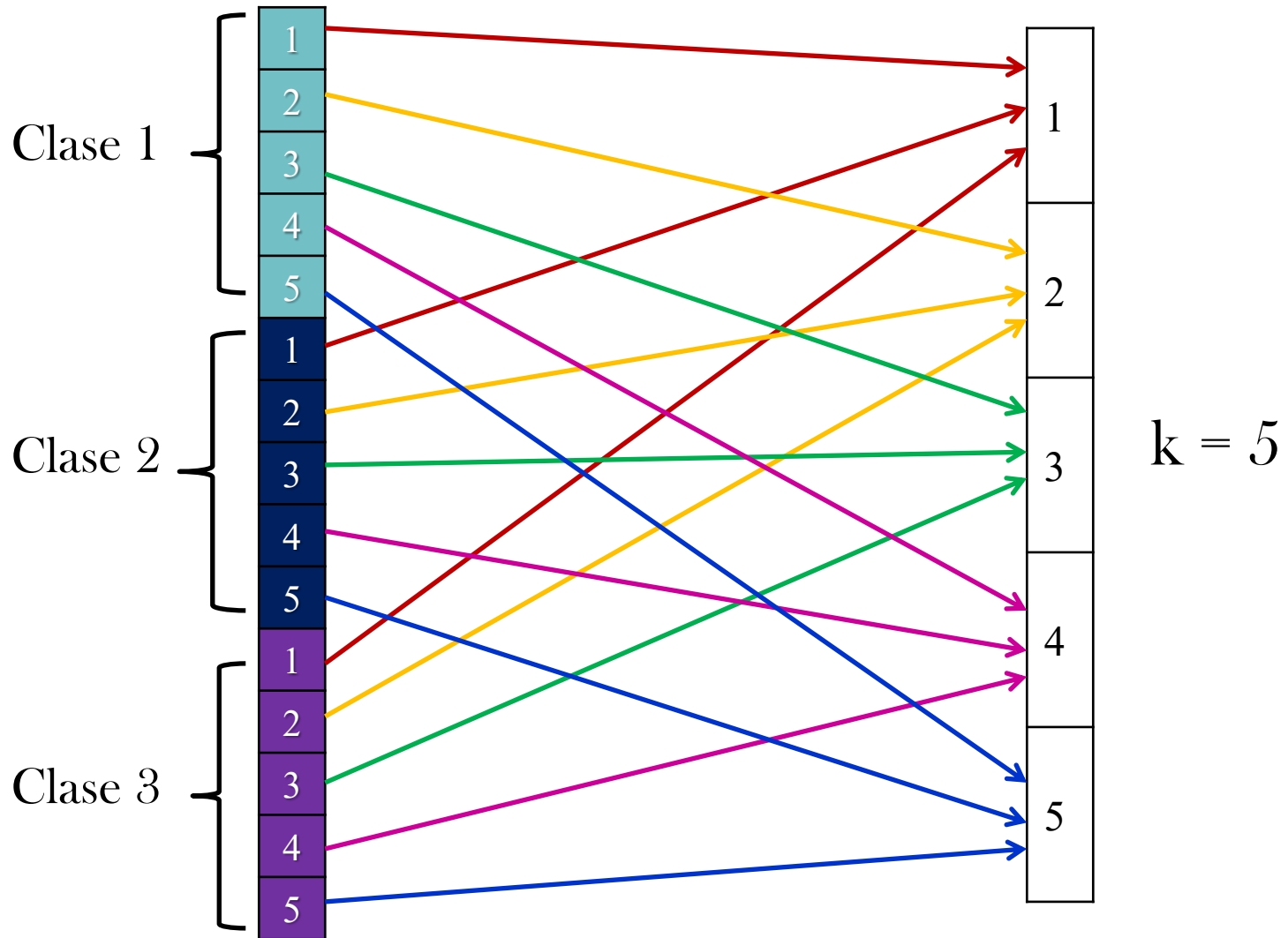
4. Ejecutar el algoritmo RP con los conjuntos E y P obtenidos.
5. Repetir el proceso N veces variando la k -ésima parte que forma el conjunto de prueba y por tanto variando el conjunto de entrenamiento.
6. Se recomienda repetir este proceso k^2 ; sin embargo, en ocasiones este proceso puede ser muy costoso
7. El método *Leave-one-out cross-validation* (LOOCV) es un caso particular del método *k-fold cross-validation*, cuando el número de *folds* es igual a la cardinalidad del BD.



Métodos de Validación (*K-fold cross-validation estratificado*)



Métodos de Validación (*K-fold cross-validation estratificado*)



Métodos de Validación (*K-fold cross-validation* estratificado)

El cálculo del desempeño con el método *K-fold cross-validation* :

1) Número de aciertos del i -ésimo *fold* entre la cardinalidad de dicho *fold*.

$$accuracy_i = \frac{\#aciertos}{|k_i|}$$

2) Promediar los valores de *accuracy* de todos los *folds*:

$$accuracy = \frac{1}{k} \left(\sum_{i=1}^k accuracy_i \right)$$

Métodos de Validación (*K-fold cross-validation estratificado*)

“ The average results of the k runs are taken as the results of the cross-validation. To reduce the influence of randomness introduced by data split, the k -fold cross-validation can be repeated t times, which is called **t -times k -fold cross-validation**. Usual configurations include *10-times 10-fold cross-validation*, and *5-times 2-fold cross-validation* suggested by Dietterich [1998]. Extremely, when k equals the number of instances in the original data set, there is only one instance in each validation set; this is called **leave-one-out (LOO)** validation. ”

Zhou, Z.H. (2012). Ensemble Methods, Chapman & Hall/CRC.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, 10(7): 1895–1923.



Métodos de Validación (*K-fold cross-validation estratificado*)

En los BD muy desbalanceados donde la cardinalidad de la clase minoritaria es muy pequeña, se dificulta aplicar el *k-fold cross-validation*.

Se recomienda usar el método de validación *5x2cv*. Este método consiste en ejecutar 5 veces una validación cruzada con $k = 2$, por cada división del conjunto de entrenamiento se realizan 2 ejecuciones, alternando que conjunto se usa para entrenamiento y prueba. Por lo que al final se tienen 10 resultados. Ver [9] para mayor detalle.



Métodos de Validación (*Bootstrap*)

Este método esta basado en un procedimiento estadístico de muestreo con reemplazo.

Hasta ahora en los métodos estudiados, cuando un patrón era tomado del banco de datos para ser parte del conjunto de entrenamiento o prueba, se hacia sin reemplazo. Es decir, el mismo patrón una vez seleccionado, no puede ser seleccionado otra vez.

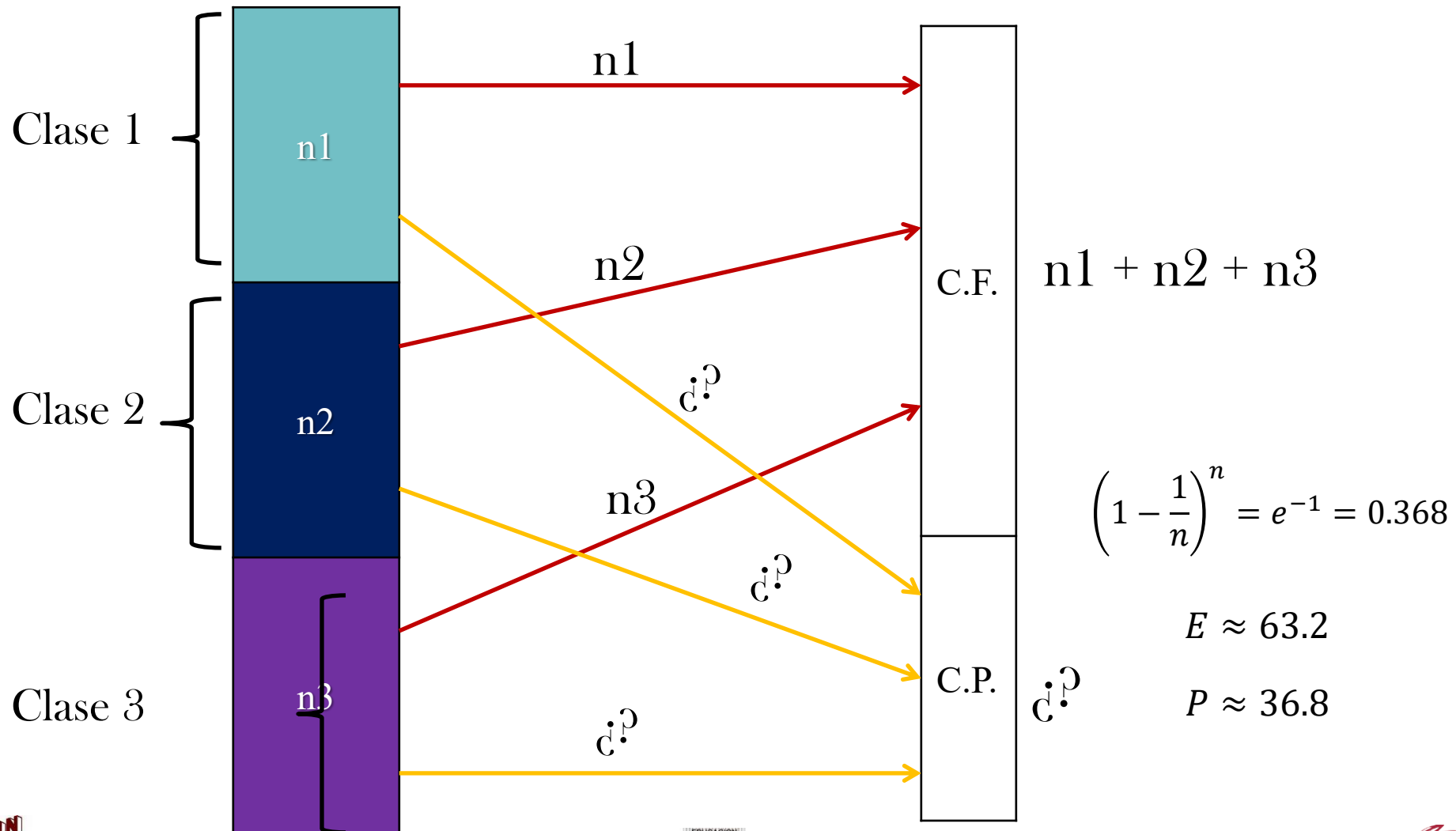
Ejemplo: cuando se seleccionan jugadores para ser partes de un equipo de futbol. La misma persona no puede ser seleccionada 2 veces.



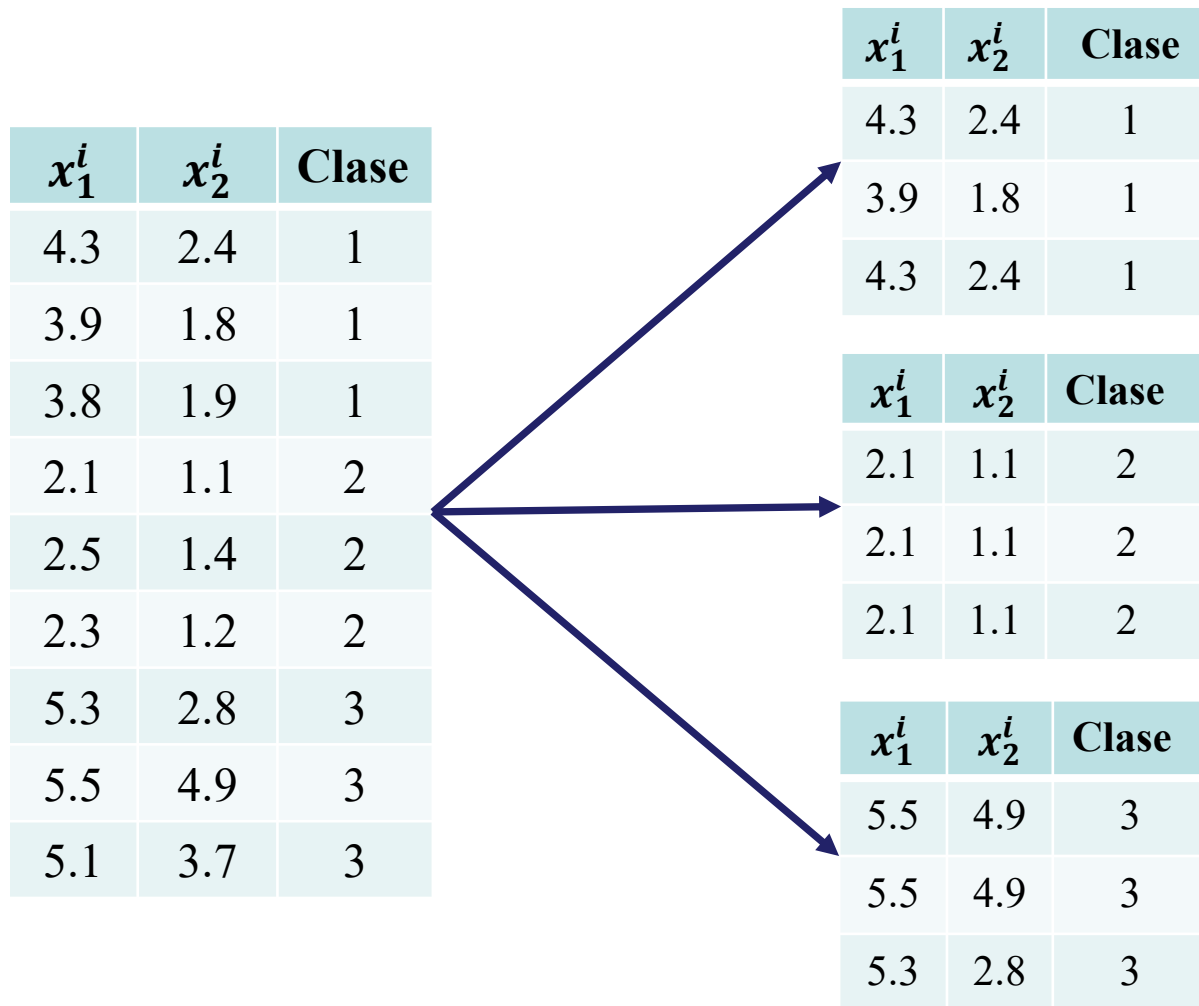
Métodos de Validación (*Bootstrap*)

1. Por cada clase i , escoger aleatoriamente un patrón que pasa a formar parte del conjunto E.
2. No se elimina el seleccionado del BD original.
3. Continuar eligiendo patrones aleatorios $N(\omega_i)$ veces, donde $N(\omega_i)$ el número de patrones en cada clase i .
4. Los que no fueron seleccionados después de $N(\omega_i)$ selecciones son usados para el conjunto P.
5. Repetir por cada clase.

Métodos de Validación (*Bootstrap*)



Métodos de Validación (*Bootstrap estratificado*)



Métodos de Validación (*Bootstrap*)

E

x_1^i	x_2^i	Clase
4.3	2.4	1
3.9	1.8	1
4.3	2.4	1
2.1	1.1	2
2.1	1.1	2
2.1	1.1	2
5.5	4.9	3
5.5	4.9	3
5.1	3.7	3

P

x_1^i	x_2^i	Clase
3.8	1.9	1
2.5	1.4	2
2.3	1.2	2
5.3	2.8	3

Obsérvese que E tiene el mismo tamaño que todo el BD original pero tiene patrones repetidos.

Métodos de Validación

Nota importante

Los métodos de validación no son comparables entre sí.

Por ejemplo: el resultado de aplicar *Hold-Out* no puede compararse con el resultado de aplicar *Leave One-Out*, *K-Fold Cross-Validation*, *Bootstrap*, entre otros.



Cuantificación del Rendimiento

Medidas de Rendimiento

Matriz de Confusión

Clase predicha

	P	N
P	TP	FN
N	FP	TN

Clase real

Nombre	Fórmula
Sensibilidad (Recall, TPR)	$\frac{TP}{TP + FN}$
Especificidad (TNR)	$\frac{TN}{TN + FP}$
Rendimiento (Accuracy)	$\frac{\text{Aciertos}}{\text{\#patrones en } P}$
Precisión	$\frac{TP}{TP + FP}$
AUC (Área bajo la curva)	$\frac{\text{sensibilidad} + \text{especificidad}}{2}$

Matriz de confusión

Los nombres de los elementos de la matriz de confusión tienen su origen en la terminología médica, que nombra como **positivo** el caso de algún paciente que sí padece cierta enfermedad, que está **enfermo**; y como **negativo** el caso de algún individuo **sano**, que no padece la enfermedad.

Sin embargo, hay una falta de estandarización en la terminología utilizada, causando cierta confusión.

Diferentes autores disponen de diferentes maneras los elementos de la matriz de confusión.

Matriz de confusión

En 1998, Kohavi y Provost editaron su *Glossary of Terms*, uno de los primeros intentos por estandarizar términos en áreas afines al cómputo inteligente. En palabras de los autores: “*To help readers understand common terms in machine learning, statistics, and data mining, we provide a glossary of common terms. The definitions are not designed to be completely general, but instead are aimed at the most common case.*” [10].

Confusion matrix A matrix showing the predicted and actual classifications. A confusion matrix is of size $\ell \times \ell$, where ℓ is the number of different label values. The following confusion matrix is for $\ell = 2$:

↓ actual \ predicted →	negative	positive
negative	a	b
positive	c	d



Matriz de confusión

WEKA hace algo aún más confuso; realiza 2 cálculos. En uno, toma la primer clase como positiva y la otra como negativa, y luego las invierte.

```
=== Stratified cross-validation ===
=== Summary ===
```

```
Correctly Classified Instances      185          64.6853 %
Incorrectly Classified Instances    101          35.3147 %
Kappa statistic                    0.1575
Mean absolute error                 0.3552
Root mean squared error            0.5423
Relative absolute error             84.8811 %
Root relative squared error        118.654 %
Total Number of Instances         286
```

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.746	0.588	0.750	0.746	0.748	0.158	0.623	0.790	no-recurrence-events
	0.412	0.254	0.407	0.412	0.409	0.158	0.623	0.410	recurrence-events
Weighted Avg.	0.647	0.489	0.648	0.647	0.647	0.158	0.623	0.677	

```
=== Confusion Matrix ===
```

```
  a   b  <-- classified as
150  51 |  a = no-recurrence-events
 50  35 |  b = recurrence-events
```

Matriz de confusión

Y hay autores que intercambian la posición de las clases reales con las predichas, en la matriz:

		<u>True class</u>	
		p	n
<u>Hypothesized class</u>	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Column totals:		P	N

Cuantificación del Rendimiento (BD desbalanceados)

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose **MultilayerPerceptron** -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds **10**

☐ Percentage split % 66

More options...

(Nom) Outcome

Start Stop

Result list (right-click for options)

04:56:07 - trees.RandomForest

05:01:30 - functions.MultilayerPerceptron

Classifier output

Time taken to build model: 11.28 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances **840** **98.1308 %**

Incorrectly Classified Instances 16 1.8692 %

Kappa statistic -0.0071

Mean absolute error 0.0256

Root mean squared error 0.13

Relative absolute error 88.6547 %

Root relative squared error 110.5473 %

Total Number of Instances 856

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.000	0.005	0.000	0.000	0.000	-0.008	0.666	0.057	Act
	0.995	1.000	0.986	0.995	0.991	-0.008	0.666	0.989	Ina
Weighted Avg.	0.981	0.986	0.972	0.981	0.977	-0.008	0.666	0.976	

=== Confusion Matrix ===

a b <-- classified as

0 12 | a = Active

4 840 | b = Inactive

Cuantificación del Rendimiento (BD desbalanceados)

El sesgo exhibido por *accuracy* en este ejemplo, la convierte en una medida de desempeño como no adecuada para BD desbalanceados.

¿Por qué ocurre esto? Porque *accuracy* mide el desempeño global sin considerar particularidades y los diferentes costos que implican las decisiones individuales.

Por ejemplo, el costo del error de clasificar un paciente enfermo como sano, es mayor que el costo del error que resulta al clasificar un individuo sano como enfermo. Otro ejemplo, el costo del error de clasificar un cliente fraudulento como confiable, es mayor que el costo del error que resulta al clasificar un cliente confiable como fraudulento.

Por ello, es preciso considerar medidas de desempeño alternas, las cuales se definen a partir de la matriz de confusión.



Tarea 2

- Localizar 3 artículos en revistas JCR donde se apliquen: *Hold-out*, *Leave-one-out* and *k-fold cross-validation*. Reportar una breve descripción del artículo (artículos publicados entre 2016, 2017 y 2018).
- Reportar si en la búsqueda se encontró otros métodos de validación que no hayan sido presentados en clase.



- [1] **Leondes, C.T. (2018).** *Image Processing and Pattern Recognition*. California: Academic Press.
- [2] **Duda, R.O., Hart, P.E. & Stork, D.G. (2001).** *Pattern Classification*. 2nd edition. Wiley-Interscience.
- [3] **Marques de Sá, J:P. (2001).** *Pattern Recognition: Concepts, Methods and Applications*. Berlin: Springer-Verlag.
- [4] **Kuncheva, L. (2014).** *Combining Pattern Classifiers: Methods and Algorithms*. 2nd edition. USA: Wiley.
- [5] **Witten, I.H., Frank, E. & Hall, M.A. (2011).** *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd edition. USA: Elsevier.
- [6] **Murty, N.M. & Devi, V.S. (2011).** *Pattern Recognition: An Algorithmic Approach*. Springer.
- [7] **Zaki, M.J. & Meira, W. (2014).** *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press.
- [8] **Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H. & Bing, G. (2017).** Learning from class-imbalanced data: Review of methods and applications. *Expert Systems With Applications*, 73, 220-239.
- [9] **Dietterich, T.G. (1998).** Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7), 1895 – 1923.
- [10] **Kohavi, R. & Provost, F. (1998).** Glossary of terms. *Machine Learning*, 30(2/3), 271 – 274.





¡ Gracias !

Thanks !

Obrigado

Xie xie ni

Domo arigatou

Спасибо

Merci

Grazie

Alfa Beta