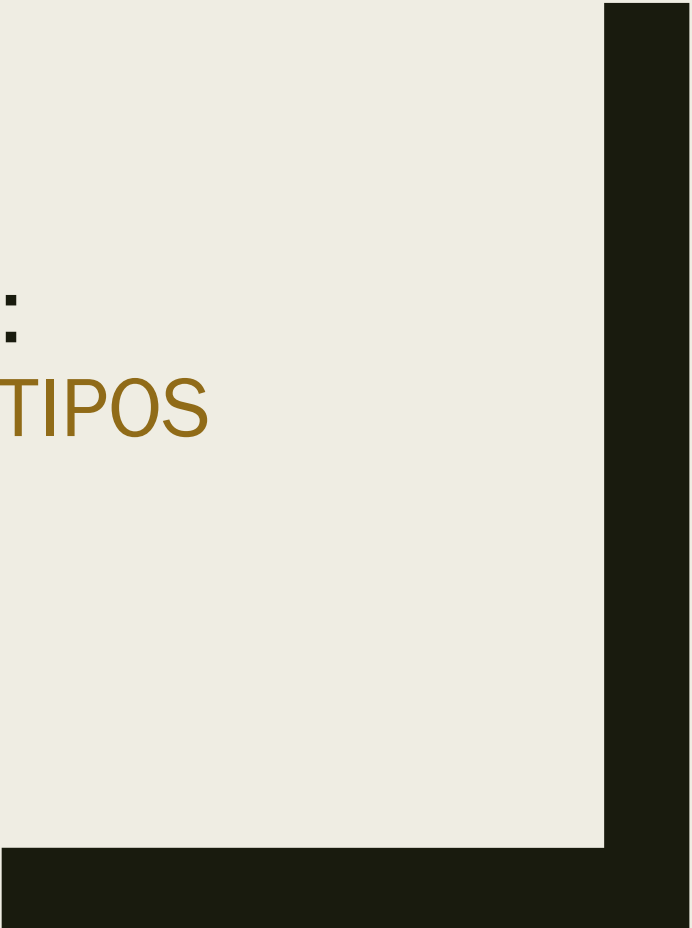


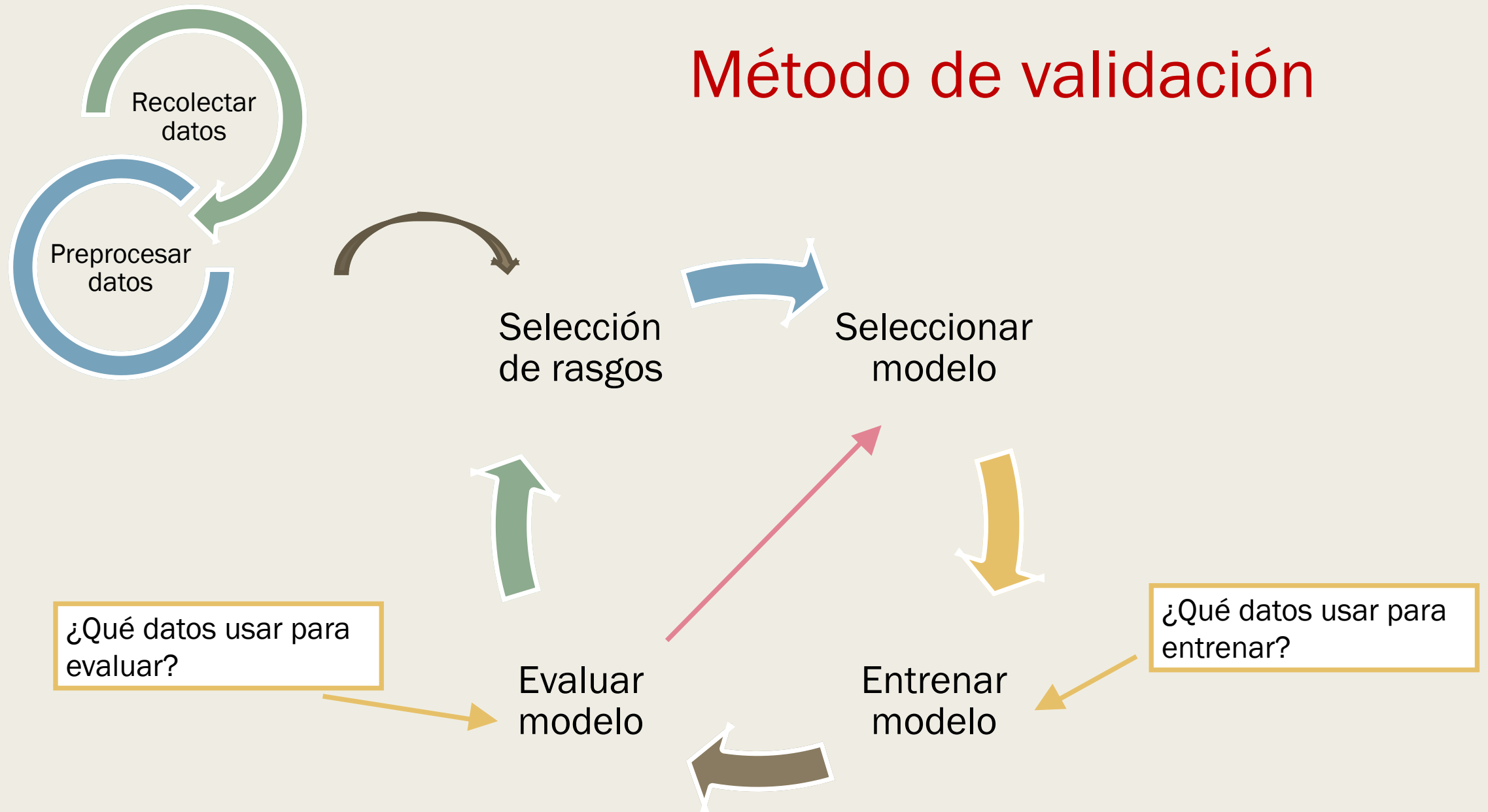


ANALÍTICA AVANZADA DE DATOS: VALIDACIÓN

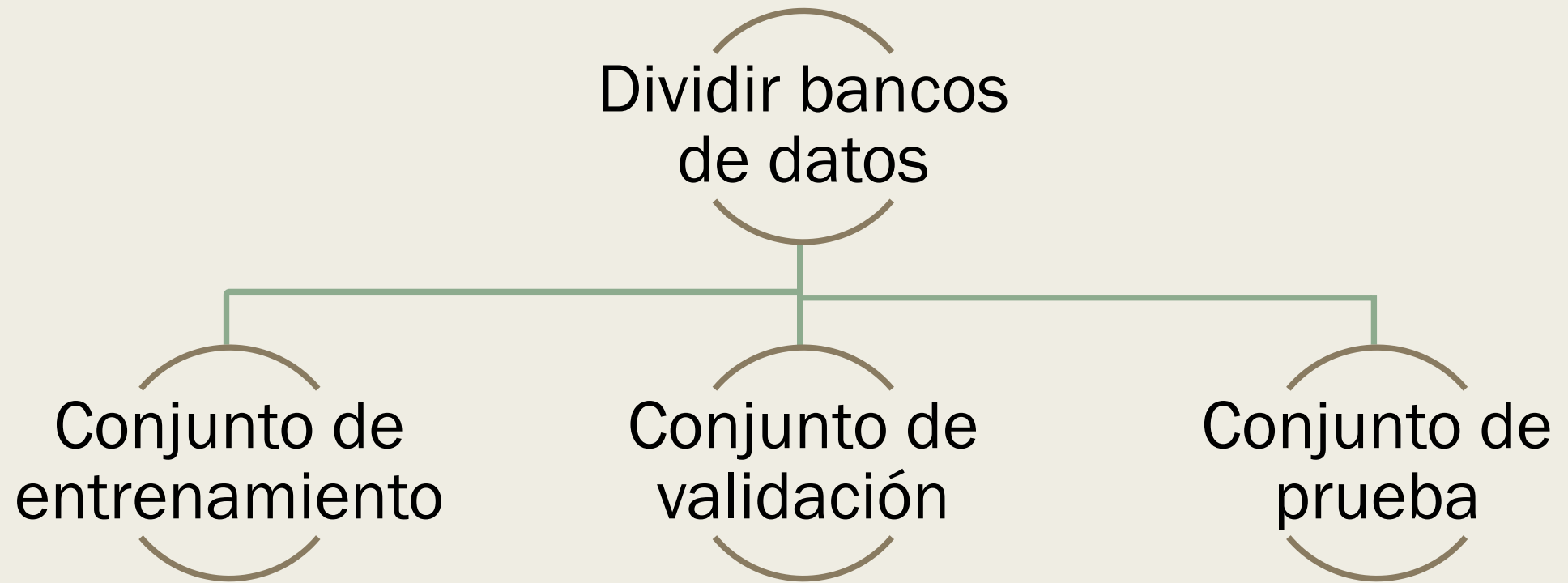
A. Alejandra Sánchez Manilla
asanchezm.q@gmail.com



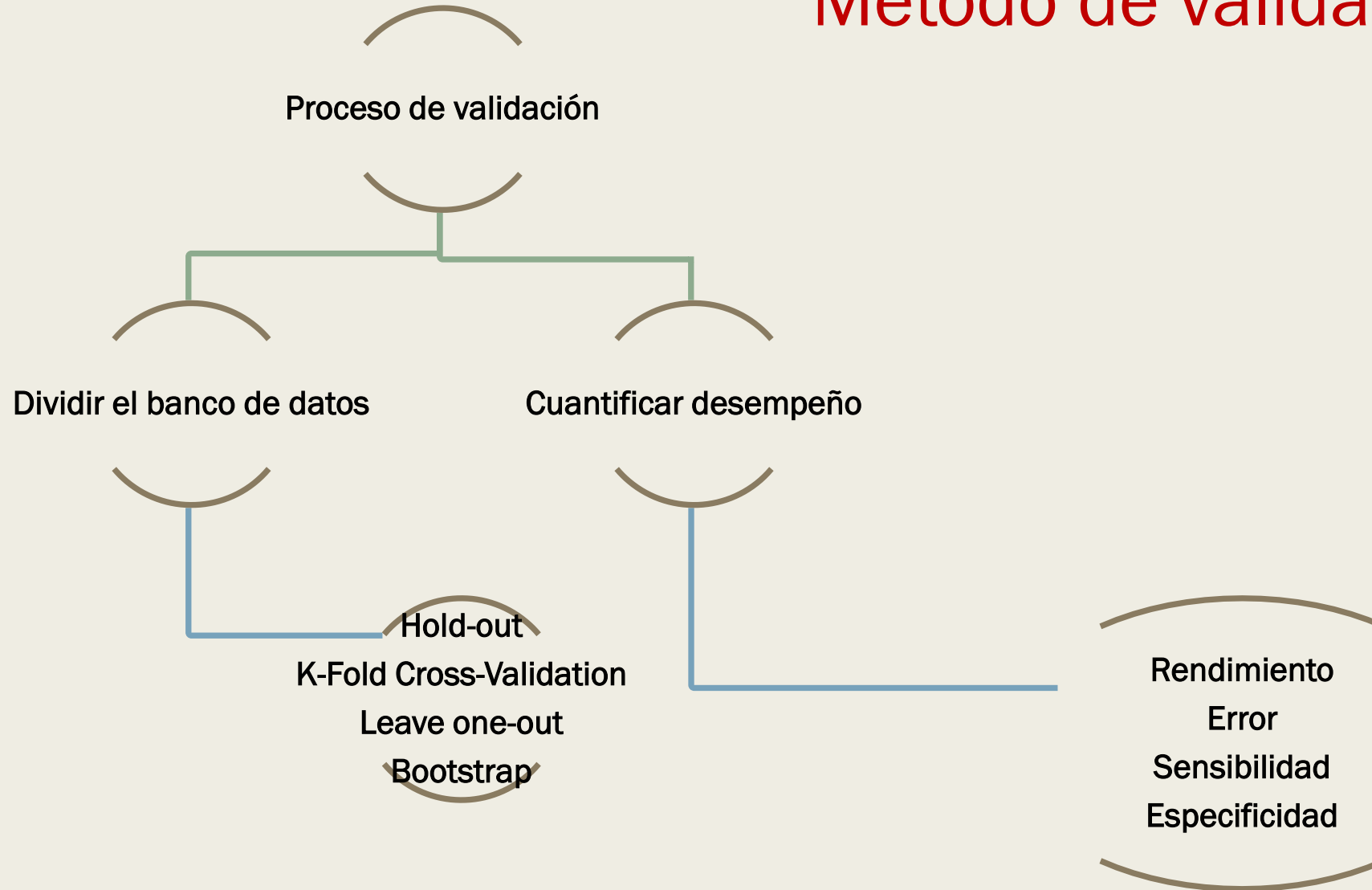
Método de validación



Método de validación



Método de validación



Método de validación

En la actualidad se identifican varios métodos de validación que la comunidad científica prefiere utilizar en sus publicaciones JCR:

- En primer lugar, el método de validación K-fold cross-validation, siendo el valor de k más usado 10.
- En segundo lugar, el método de validación Leave-one-out cross-validation (LOOCV).
- 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.

Método de validación

Factor de Olvido (Resubstitution Error)

- No es propiamente un método de validación.
- Es el error en el conjunto de entrenamiento.
- Generalmente **no es cero** en los algoritmos de aprendizaje automático, pero esperamos que sea bajo.



Saltillo es la capital de Coahuila

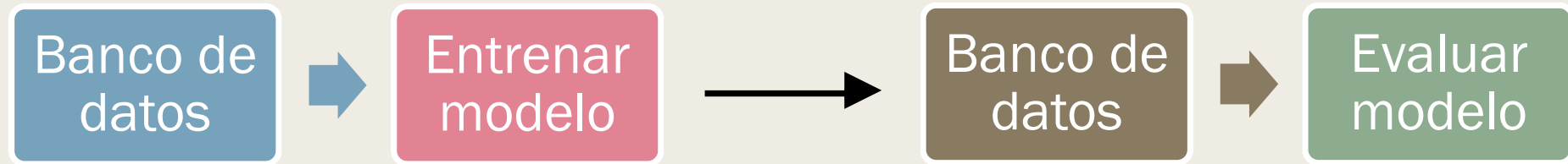
¿Cuál es la capital de Coahuila?



Mexicalli

Factor de Olvido (Resubstitution Error)

Datos de entrenamiento = Datos de prueba



- Se utiliza para ajuste de parámetros
- El error en los datos de entrenamiento NO es un buen indicador de rendimiento sobre datos futuros ya que no mide ningún dato aún no visto

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

Método de validación

Factor de Olvido (Resubstitution Error)

Datos de entrenamiento = Datos de prueba

- Se utiliza para ajuste de parámetros.
- El error en los datos de entrenamiento NO es un buen indicador de rendimiento sobre datos futuros ya que no mide ningún dato aún no visto.

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

Método de validación

Hold - out

1. Dividir el banco de datos en los subconjuntos de entrenamiento (E) y prueba(P), de la siguiente forma:

$$|E| = r * N$$

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

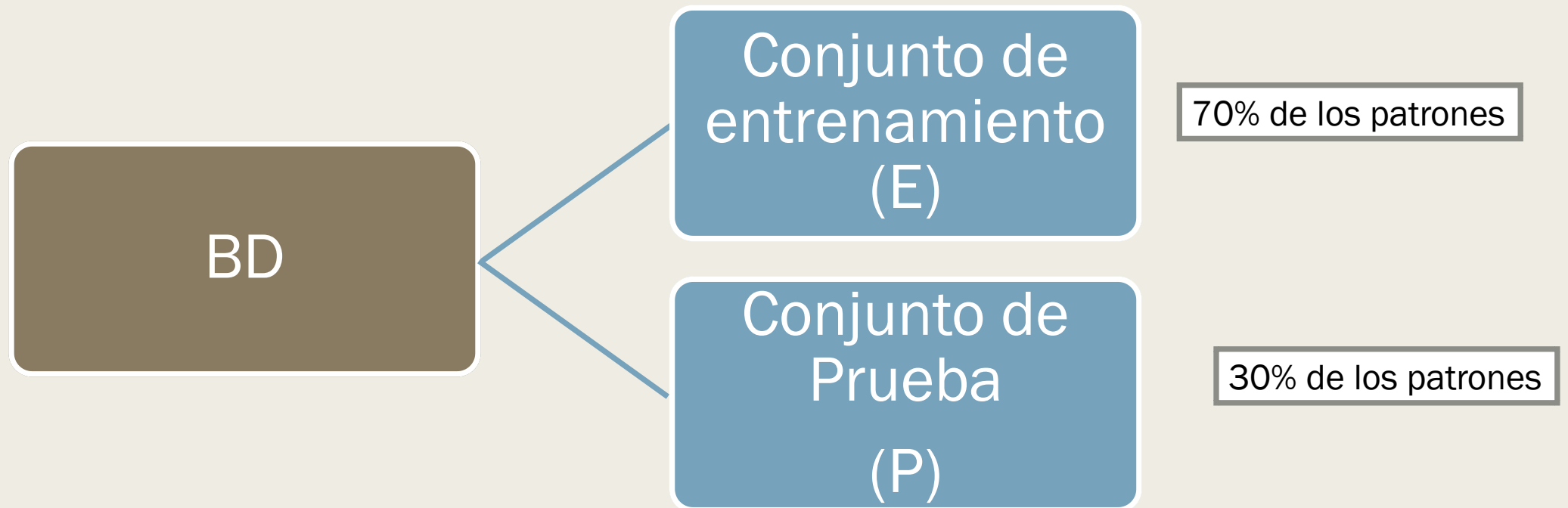
donde:

N es el # de patrones del banco de datos

r porcentaje de entrenamiento: 0.7

2. Seleccionar aleatoriamente $|E|$ patrones del banco de datos para crear el conjunto de entrenamiento
3. Seleccionar aleatoriamente $|P|$ patrones del banco de datos para crear el conjunto de prueba
4. Entrenar el algoritmo con E y probar con P

Hold-out



Hold-out

1	Iris Setosa	}	Patrones del 1 al 50
⋮	⋮		
50	Iris Setosa		
51	Iris Versicolor	}	Patrones del 51 al 100
⋮	⋮		
100	Iris Versicolor		
101	Iris Virginica	}	Patrones del 101 al 150
⋮	⋮		
150	Iris Virginica		

Para entrenamiento, el 70 %de los patrones correspondería a:

$$|E| = 150 * 0.7 = 105 \text{ patrones}$$

$$E = \{1,2,3 \dots, 5\}$$

$$P = \{106,107, \dots, 150\}$$

En el conjunto de entrenamiento no existirían patrones representantes de la iris virginica, mientras que en el prueba no habría patrones de la setosa y versicolor

Método de validación

Hold-out Estratificado

1. Dividir el banco de datos en los subconjuntos de entrenamiento (E) y prueba (P), de la siguiente forma, por cada clase i :

$$|E| = \sum_{i \in C} r * |C_i|$$

$$|P| = N - |E|$$

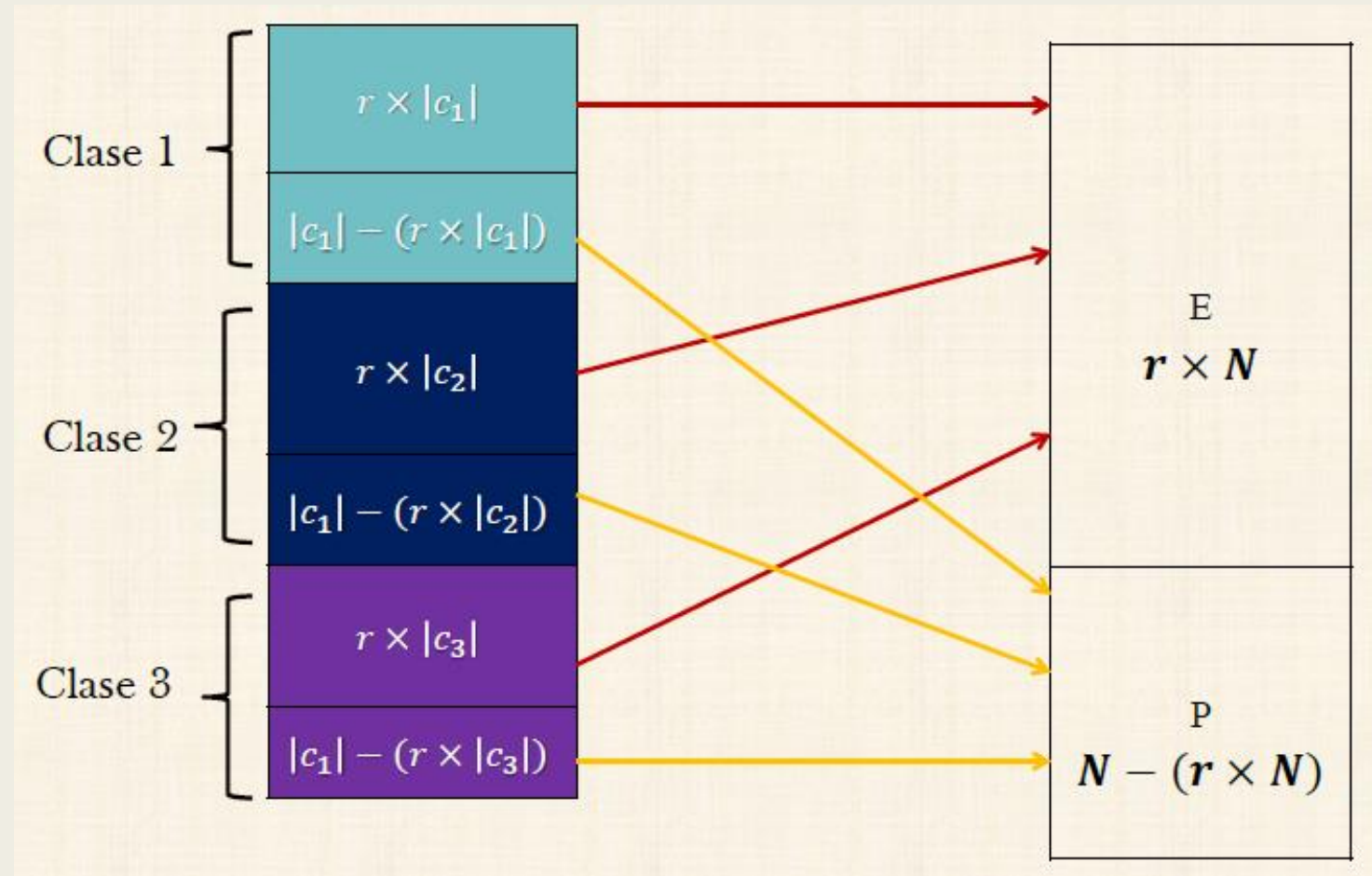
donde:

$|C_i|$ es el # de patrones de la clase i

r es el porcentaje de entrenamiento: 0.7

2. Seleccionar aleatoriamente $|E|$ patrones del banco de datos para crear el conjunto de entrenamiento
3. Seleccionar aleatoriamente $|P|$ patrones del banco de datos para crear el conjunto de prueba
4. Entrenar el algoritmo con E y probar con P

Hold-out Estratificado



Hold-out Estratificado

Ejemplo con la irisPlant y $r = 0.7$

$$r \times |c_1| = 0.7 \times 50 = 35 \text{ patrones}$$

$$|c_1| - (r \times |c_1|) = 50 - 35 = 15$$

$c_1 = \text{Iris Setosa}$ 50 patrones
$c_2 = \text{Iris Virginica}$ 50 patrones
$c_3 = \text{Iris Versicolor}$ 50 patrones

Es decir que **35** patrones seleccionados al azar de la clase Iris Setosa deben usarse para el conjunto de entrenamiento y **15** patrones de la misma clase se deben usar para el de prueba

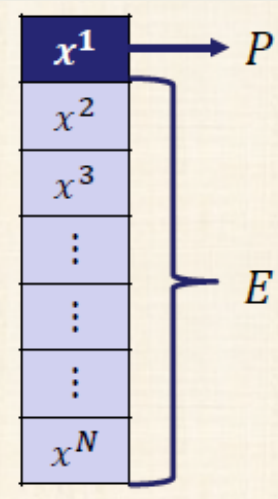
Se procede igual con las otras 2 clases y se juntan en un solo conjunto todos los patrones de las 3 clases, es decir:

- 105 patrones para el conjunto de entrenamiento
- 45 patrones para el conjunto de prueba

Método de validación

Leave-one-out

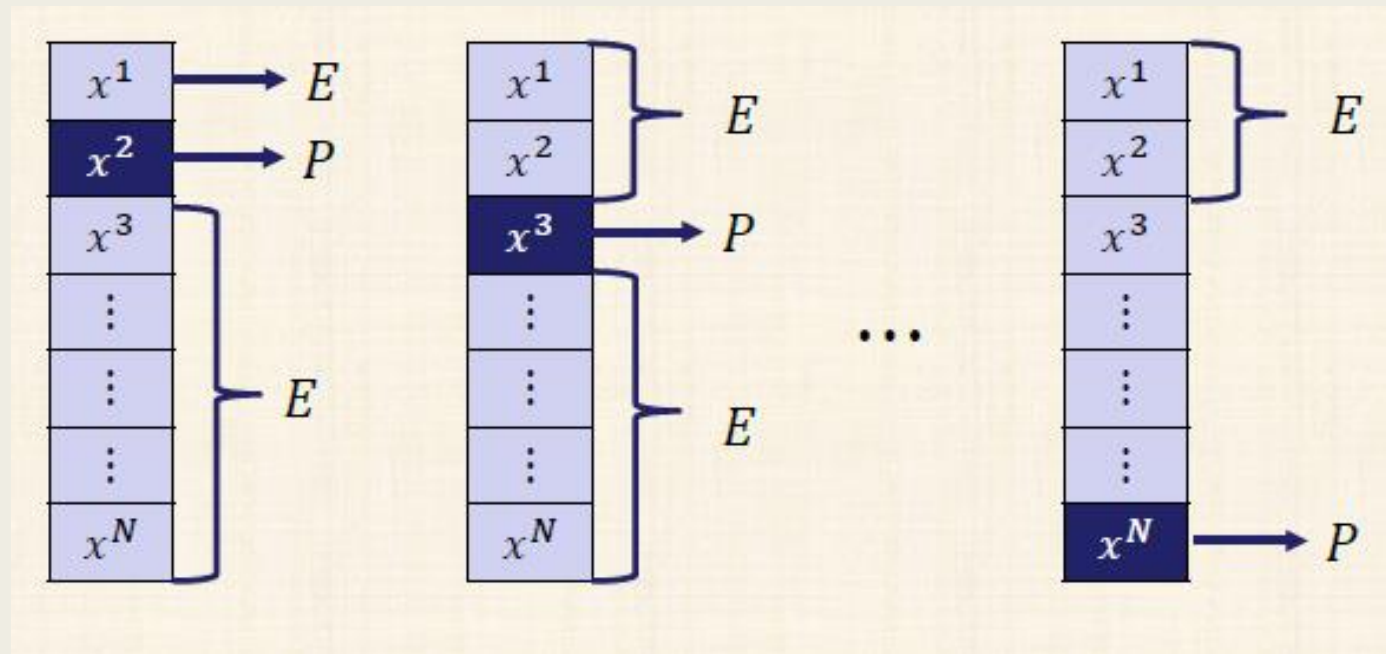
1. Seleccionar un patrón del conjunto de datos x^1 para formar el conjunto de prueba, es decir $P = x^1$. El resto de patrones, es decir $N - 1$, será el conjunto de entrenamiento



2. Entrenar el algoritmo con E y probar el algoritmo con P

Leave-one-out

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



Método de validación

K-fold Cross-validation estratificado

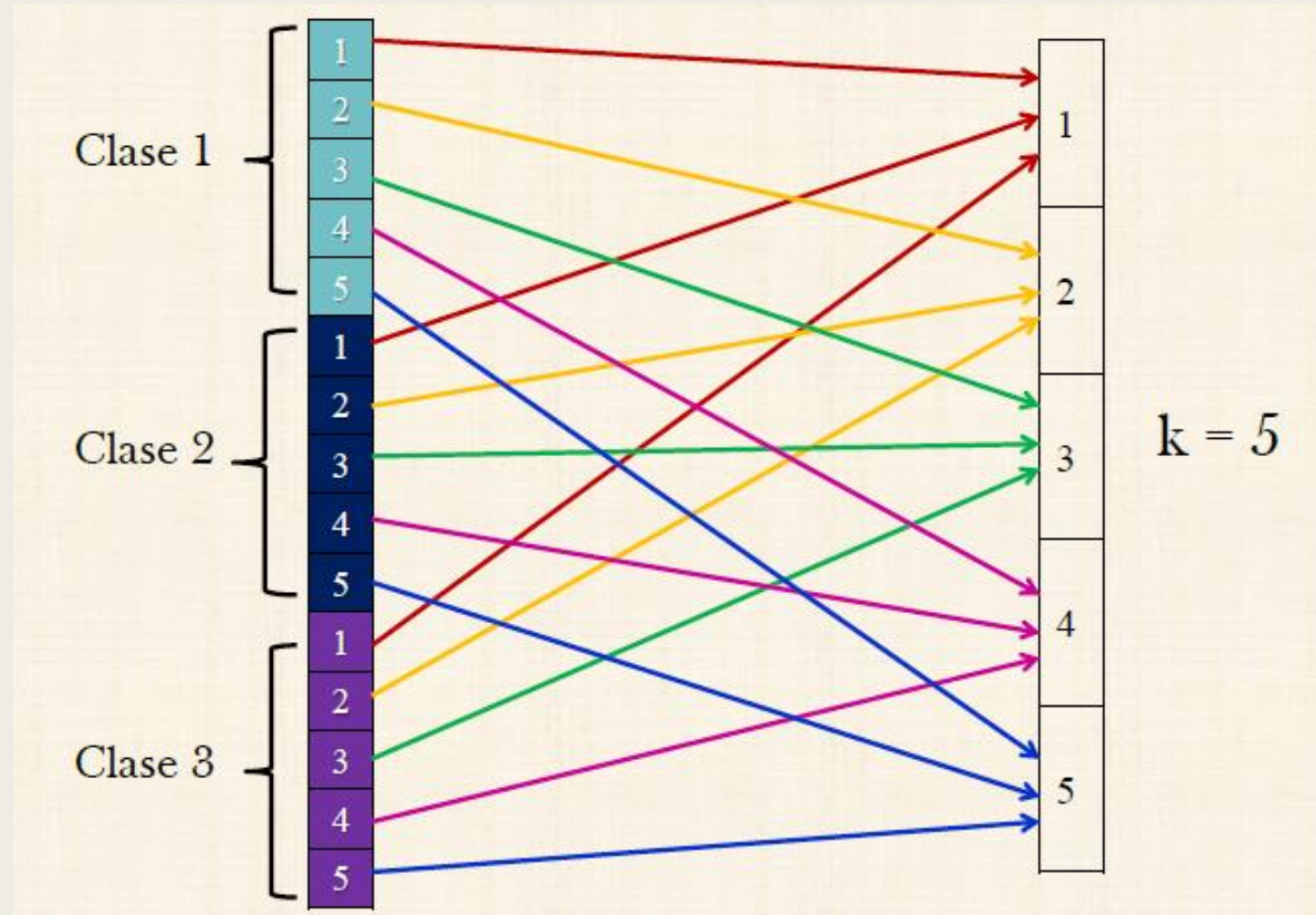
1. Elegir un valor adecuado para k (generalmente 10)
2. Separar el conjunto de datos por clase y cada clase debe dividirse en k partes
3. Formar los conjuntos de entrenamiento (E) y prueba (P) de la siguiente forma:

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

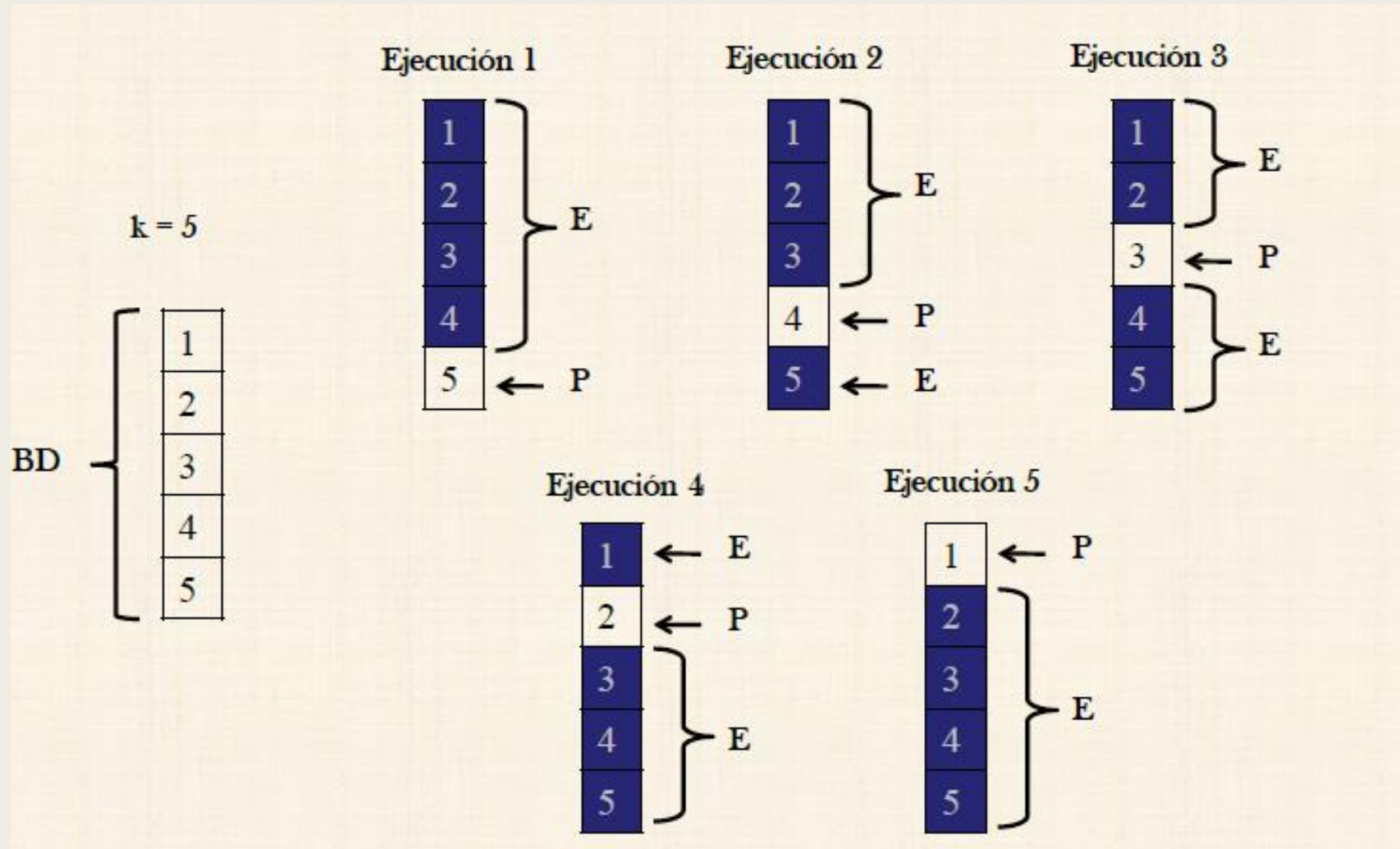
$$P = 1k$$

4. Entrenar el algoritmo con E y probar el algoritmo con P
5. El proceso debe repetirse k veces variando que participaciones se usan en E y P

K-fold Cross-validation estratificado

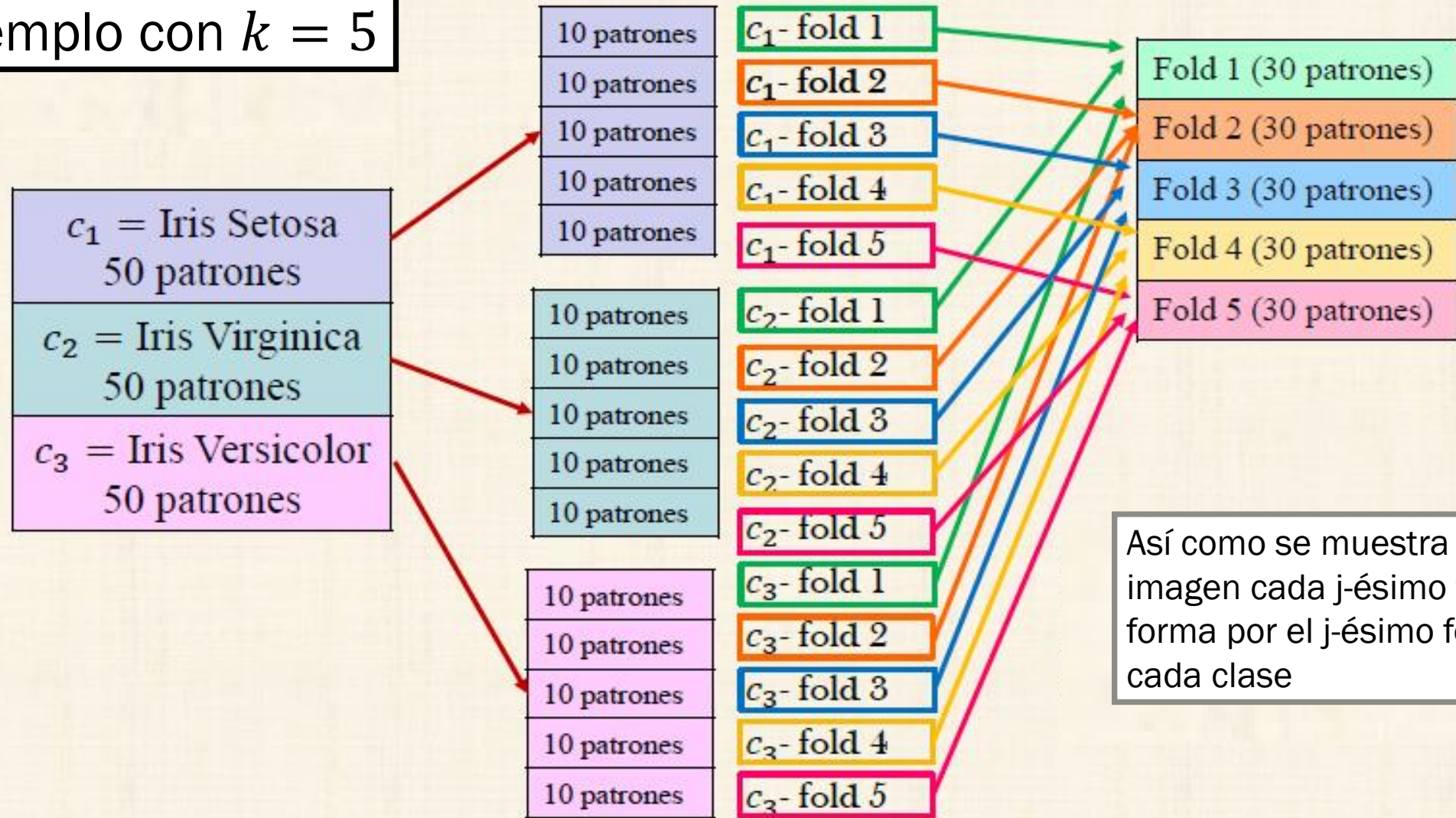


K-fold Cross-validation estratificado



K-fold Cross-validation estratificado

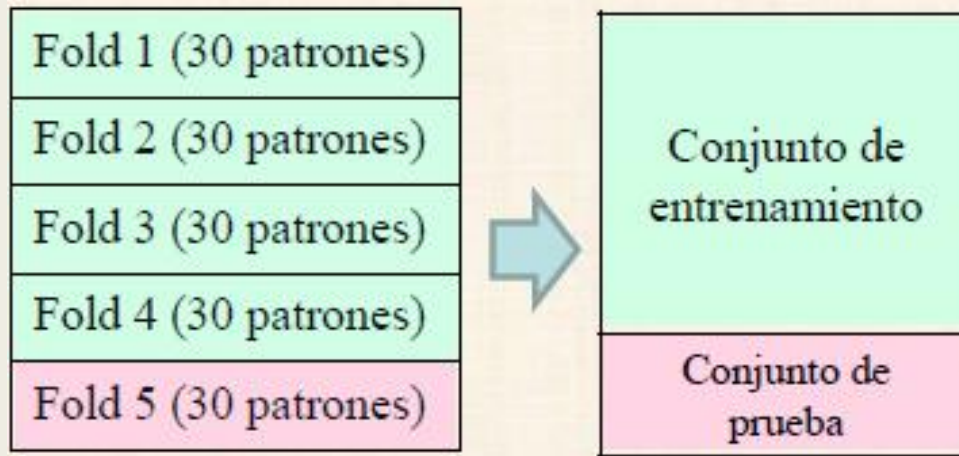
Ejemplo con $k = 5$



K-fold Cross-validation estratificado

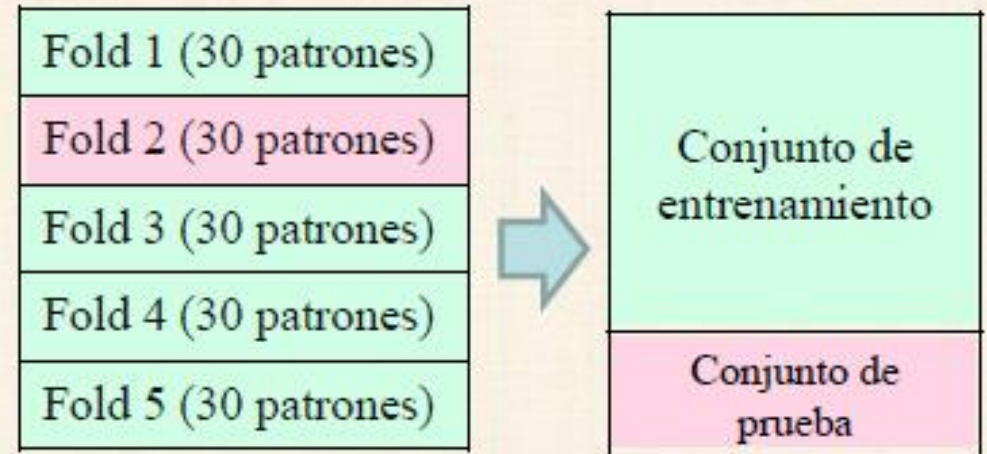
Ejemplo con $k = 5$

Ejecución 1 del algoritmo



- Los *folds* 1,2,3,4 forman el conjunto de entrenamiento
- El *fold* 5 forma el conjunto de prueba

Ejecución 2 del algoritmo

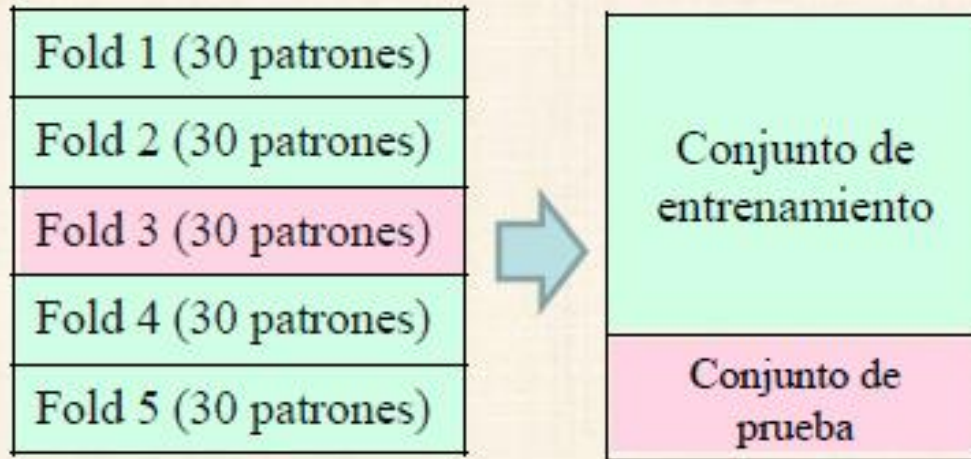


- Los *folds* 1,3,4,5 forman el conjunto de entrenamiento
- El *fold* 2 forma el conjunto de prueba

K-fold Cross-validation estratificado

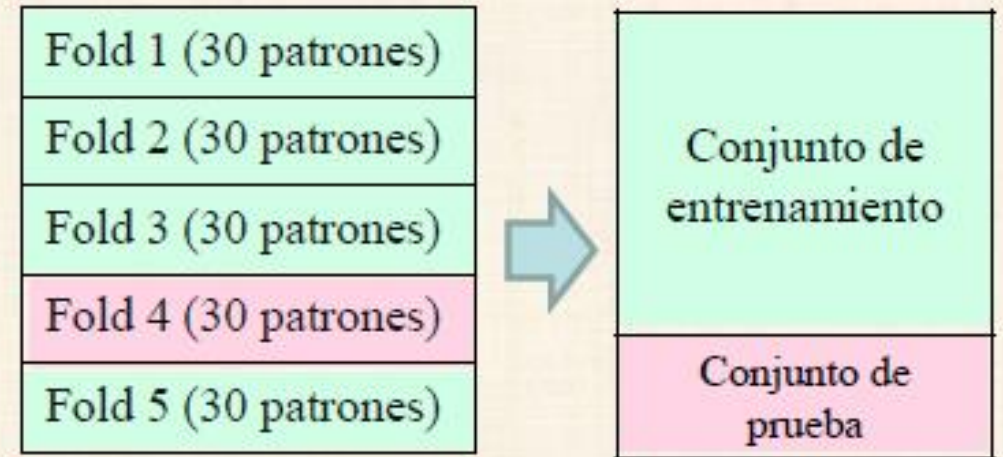
Ejemplo con $k = 5$

Ejecución 3 del algoritmo



- Los *folds* 1,2,4,5 forman el conjunto de entrenamiento
- El *fold* 3 forma el conjunto de prueba

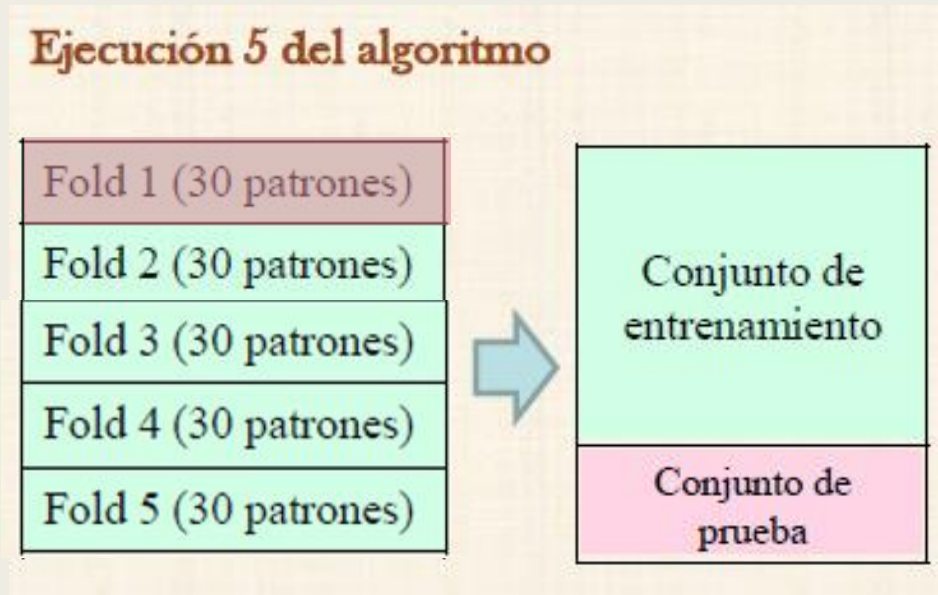
Ejecución 4 del algoritmo



- Los *folds* 1,2,3,5 forman el conjunto de entrenamiento
- El *fold* 4 forma el conjunto de prueba

K-fold Cross-validation estratificado

Ejemplo con $k = 5$



- Los *folds* 2,3,4,5 forman el conjunto de entrenamiento
- El *fold* 1 forma el conjunto de prueba

Método de validación

Matriz de confusión

Clase real	Clase predicha	
	P	N
	P	N
P	TP	FN
N	FP	TN

Se le pasan estos datos a un algoritmo de aprendizaje automático y se obtienen los siguientes resultados:

Ejemplo:

Pacientes	Cantidad
Sanos	80
Enfermos	35

Enfermos → Clase positiva

Sanos → Clase negativa

	P	N
P	28	7
N	5	75

Cuantificar el rendimiento - Clasificación

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 esta enfermo; y como **negativo** el caso de algún individuo sano, que no padece la enfermedad.

		<u>True class</u>	
		p	n
<u>Hypothesized class</u>	Y	True Positives	False Positives
	N	False Negatives	True Negatives

Cuidado con las representaciones de la matriz ya que en ocasiones cambia el orden

Cuantificar el rendimiento - Clasificación

Sensibilidad
(Recall, TPR)

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

Especificidad
(TNR)

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

Exactitud
(Accuracy)

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Precisión

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

F1-score

$$2 \times \frac{\text{Sensibilidad} \times \text{Precisión}}{\text{Sensibilidad} + \text{Precisión}}$$

AUC
(Área bajo la curva)

$$\frac{\text{Sensibilidad} \times \text{especificidad}}{2}$$

Referencias

- 1. Russell, S. J. & Norvig, P. (2010). Artificial intelligence a modern approach. 3ra edición. Pearson Education, Inc.
- 2. Tom, T. (2019). Artificial Intelligence Basics: A Non-Technical Introduction. Monrovia, CA, USA: Appres.
- 3. Ertel, W. (2018). Introduction to artificial intelligence. 2da edición. Springer.
- 4. Taulli, T. (2019). Artificial Intelligence basics: A non-technical introduction. Apress.
- 5. Géron, A. (2017). Hands-on machine learning with scikit-learn and tensorflow: Concepts, Tools, and Techniques to build intelligent systems.
- 6. 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.