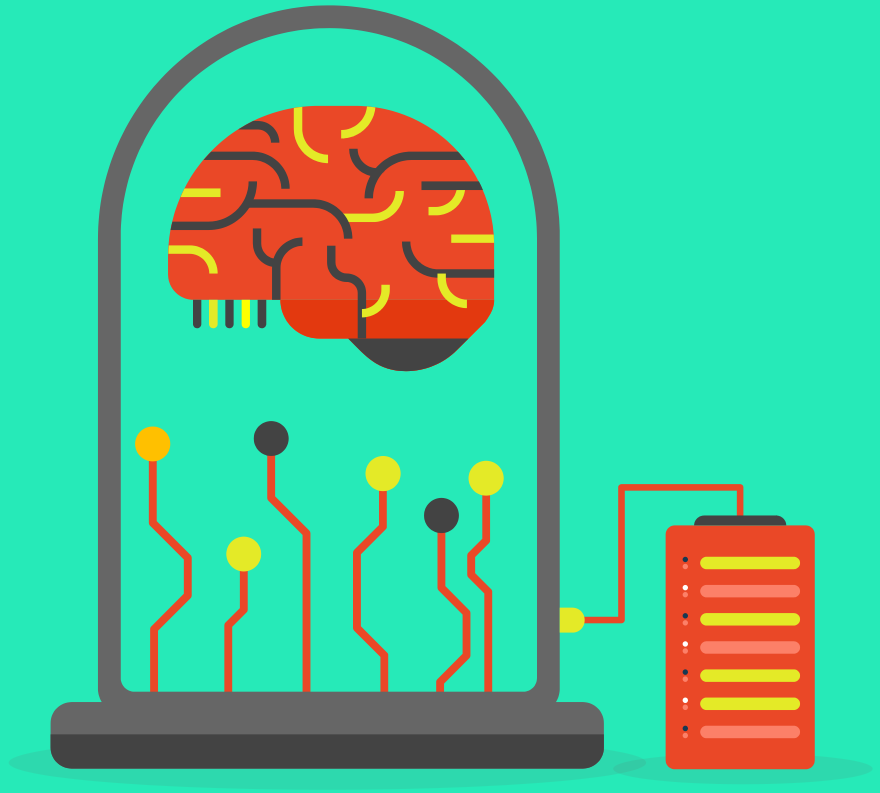
19. Selección de Modelos



Selección de Modelos en Machine Learning

Para este momento, ya hemos estudiado e implementado una serie importante de algoritmos de Machine Learning para los casos de regresión, clasificación y agrupamiento. Específicamente para los casos de regresión y clasificación, vimos una gran variedad de algoritmos que, si bien tienen cada uno sus ventajas y desventajas, cualquiera de ellos puede resultar un buen candidato a la hora de encontrar soluciones a problemas de Machine Learning.

Sin embargo, ahora que tenemos estas alternativas, podemos plantearnos lo siguiente: Dado un conjunto de datos de estudio, ¿cómo seleccionar, entre todos los modelos disponibles, el más adecuado? ¿Qué consideraciones o criterios podemos tomar para decidir cuál es el mejor modelo?

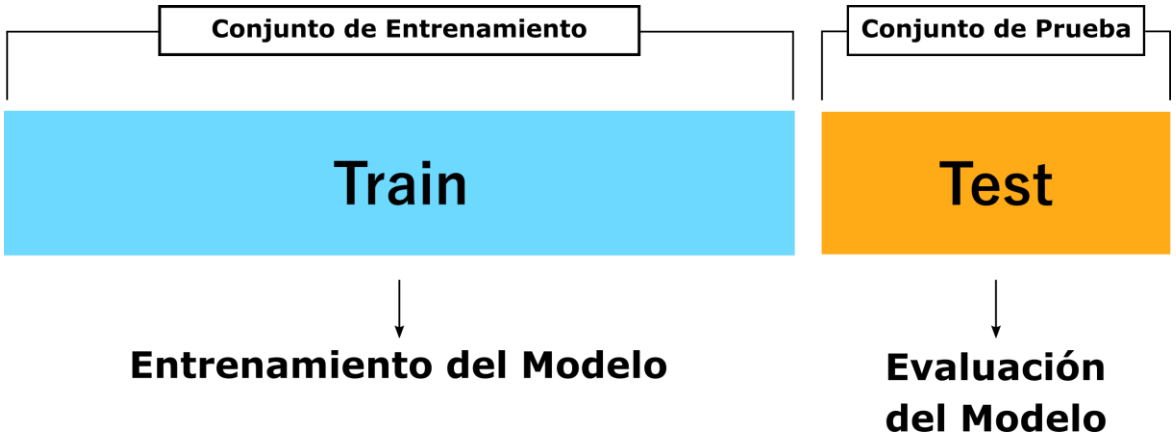
Selección de Modelos en Machine Learning

A este aspecto del Machine Learning se le conoce como “Selección de Modelos”, y las dos principales técnicas o metodologías para ello son la **Validación Cruzada K-Fold** (K-Fold Cross Validation) y la **Afinación de Hiperparámetros** (Hyperparameters Tuning).

Al aplicar estas técnicas en conjunto, contaremos entonces con una serie de herramientas fundamentales y, hoy en día, de práctica común, que garantizarán en gran medida la construcción y validación del mejor modelo posible para resolver o enfrentar nuestro problema de Machine Learning (regresión o clasificación).

Selección de Modelos en Machine Learning

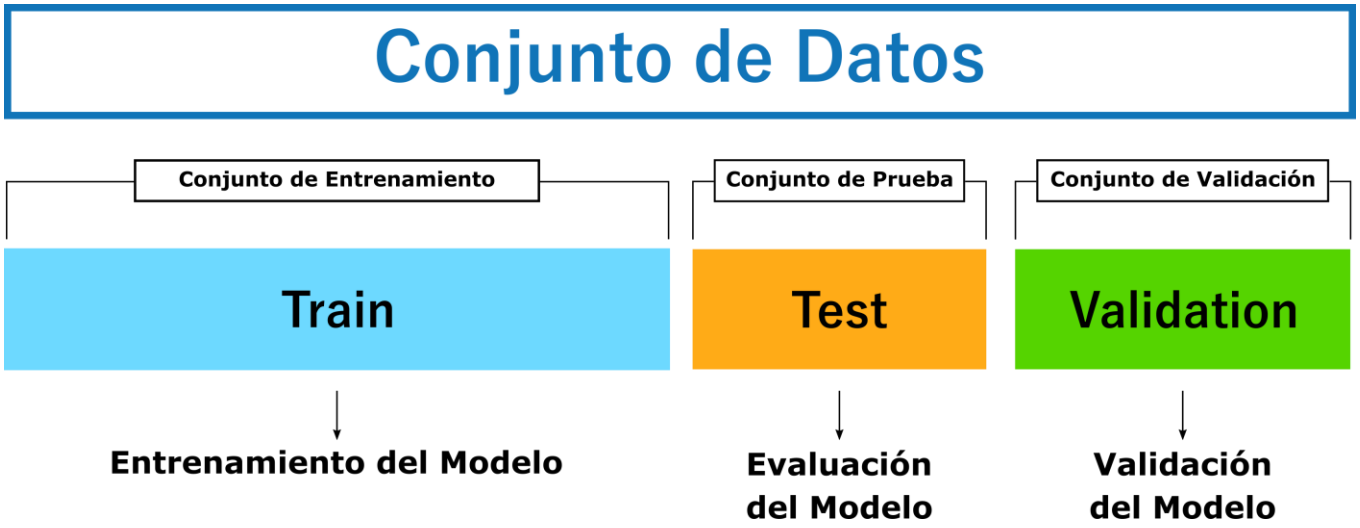
Como estudiamos anteriormente, durante todo proceso de entrenamiento y evaluación de un modelo de Machine Learning, resulta imprescindible dividir los datos de entrada del modelo en un conjunto de Entrenamiento y en un conjunto de Prueba:



Tal y como vimos en las implementaciones prácticas, esto nos permite conocer de manera cuantitativa el desempeño de cada modelo y su capacidad de generalización.

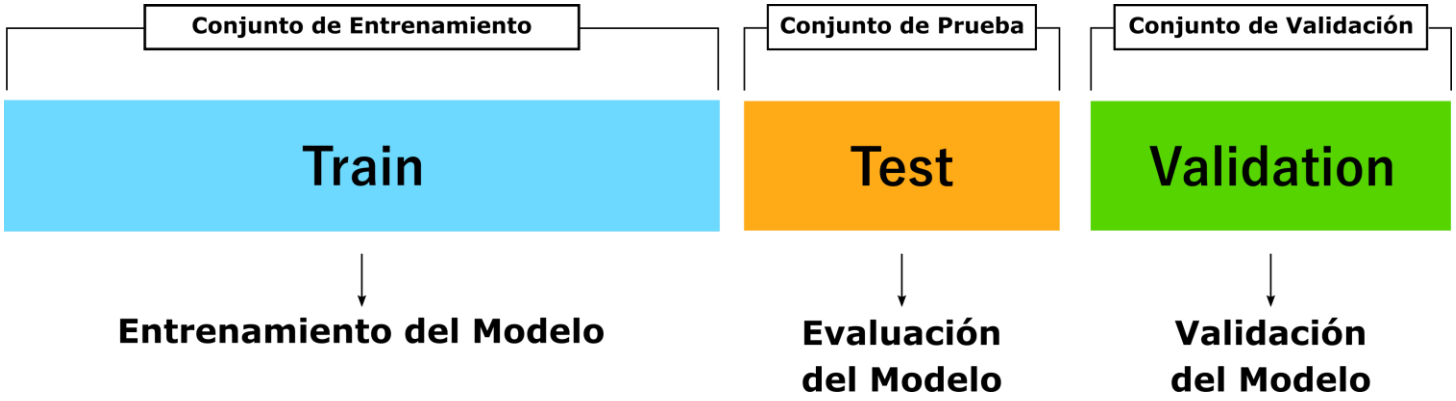
Selección de Modelos en Machine Learning

Ahora bien, supongamos que disponemos de un conjunto de datos específico, y además contamos con varios modelos de Machine Learning distintos que se pueden aplicar a nuestro problema. A fin de seleccionar de la manera más adecuada posible el mejor modelo, el método recomendado será entonces dividir los datos de entrada en tres grupos:



Selección de Modelos en Machine Learning

En este caso tendremos, tal y como ya lo vimos, un porcentaje del total para entrenar cada modelo y afinar sus hiperparámetros, y otro grupo más pequeño para evaluar el desempeño individual de cada uno. Luego, tendremos un tercer conjunto de datos, llamado de “Validación”, que no usaremos sino hasta el final, una vez que todos los modelos estén entrenados.



El conjunto de validación sólo se usará para elegir, según algún criterio (por ejemplo, el F1-score) el mejor modelo definitivo.

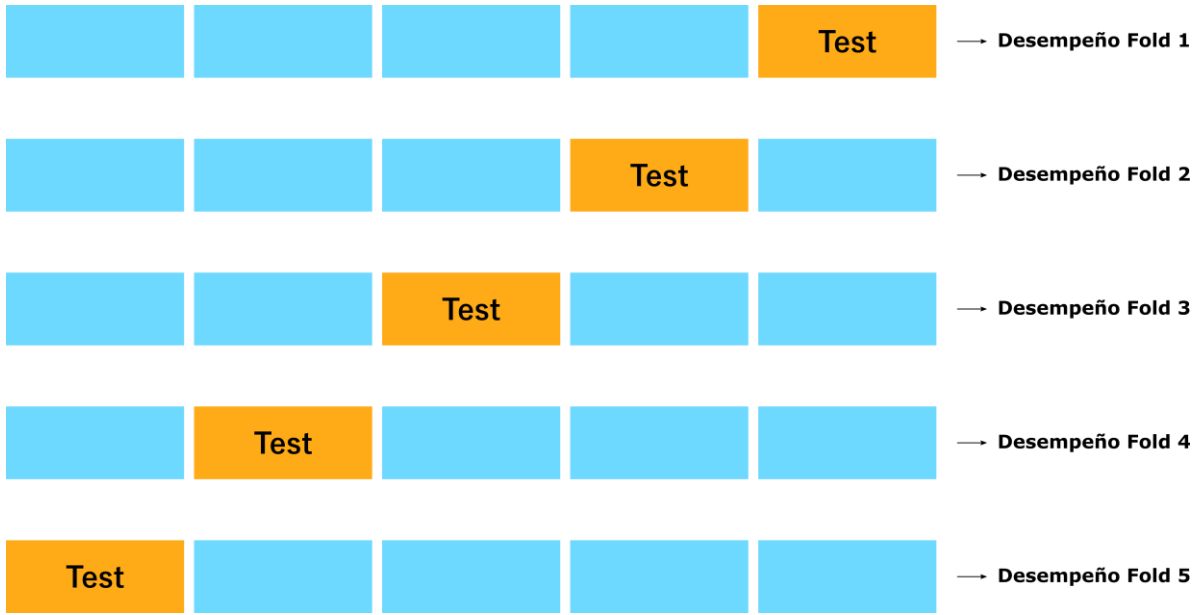
Validación Cruzada K-Fold

Ahora bien, antes de realizar esta validación final, es necesario asegurarse de que cada modelo está construido y afinado de tal manera que ofrece la mejor solución posible a los datos de entrada. El primer paso para ello, es aplicar lo que se conoce como **Validación Cruzada K-Fold**.

Esta validación es similar a la que vimos en la parte práctica, solo que ahora, en vez de entrenar con un porcentaje de los datos y evaluar el modelo con el restante, los datos se van a dividir en **K** grupos (generalmente 5 o 10): uno de esos grupos se usará para probar el modelo (test) mientras que el resto se usa como conjunto de entrenamiento (train). Luego, se repite el experimento pero ahora cambiando el bloque de datos test, y el resto se utiliza como nuevo conjunto train. Y así sucesivamente hasta haber utilizado todos los **K** grupos de datos.

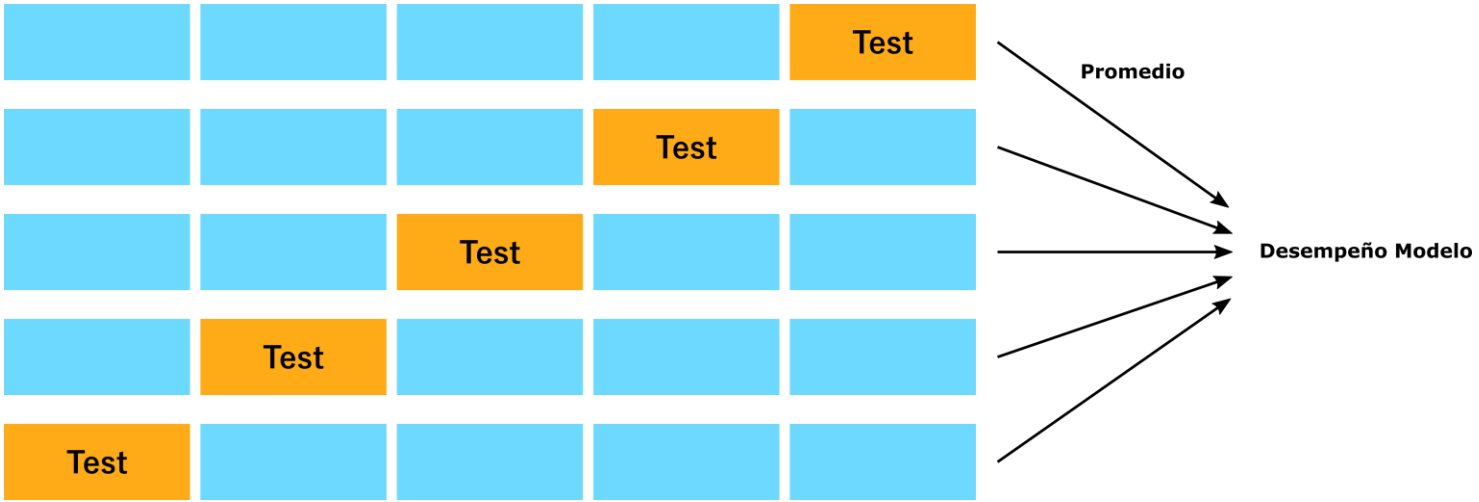
Validación Cruzada K-Fold

En cada experimento, se deberá llevar registro del desempeño (métrica de error seleccionada). El desempeño final para dicho modelo, será entonces el promedio de cada uno de los obtenidos en los experimentos separados.



Validación Cruzada K-Fold

La razón por la cual se hace esto, es que al dividir en cada subgrupo los datos de entrenamiento, y evaluar el desempeño con un bloque distinto, estamos garantizando la variabilidad estadística de los datos tanto de entrenamiento como de prueba. Es decir, que validamos el desempeño promedio del modelo en condiciones distintas de entrenamiento dentro del espacio en donde viven los datos originales.



Afinación de Hiperparámetros

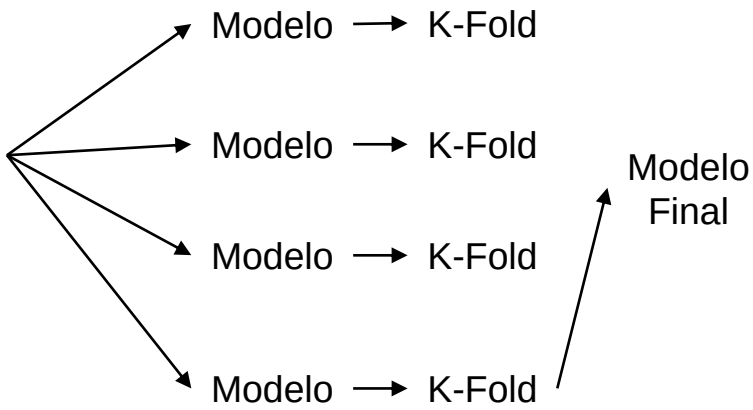
Sin embargo, todavía queda un aspecto más a ajustar en nuestros modelos durante la etapa de entrenamiento, y estos son los “Hiperparámetros”. Como vimos en las implementaciones prácticas, los modelos de Machine Learning suelen tener parámetros que son propios de cada modelo, que son constantes durante el entrenamiento y que se preseleccionan antes de iniciarlo (por ejemplo, la profundidad de los árboles de decisión, la cantidad de nodos, etc).

En la parte práctica del curso, también pudimos ver de primera mano el cómo la variación de estos parámetros puede impactar el resultado de cada modelo. Ya que este proceso de afinación puede resultar tedioso y muy propenso al “ensayo y error”, una manera ampliamente usada para el ajuste de hiperparámetros es el llamado **Grid Search**, o **Búsqueda en Malla**.

Grid Search

El Grid Search consiste en la siguiente idea: dados todos los hiperparámetros posibles de un modelo de Machine Learning, se establece una “Malla” de valores para cada uno dentro de un rango adecuado. Entonces, para todas y cada una de las combinaciones existentes, se va a construir un modelo y se va a aplicar la Validación K-Fold. Aquella combinación de hiperparámetros que maximice el desempeño de la Validación, será entonces la que escogeremos para nuestro modelo final.

	Valores				
Parámetro 1	1	2	3	4	5
Parámetro 2	0.2	0.4	0.6	0.8	1
Parámetro 3	10	20	30	40	50
Parámetro 4	0.001	0.005	0.008	0.015	0.025



Selección del Modelo Ganador

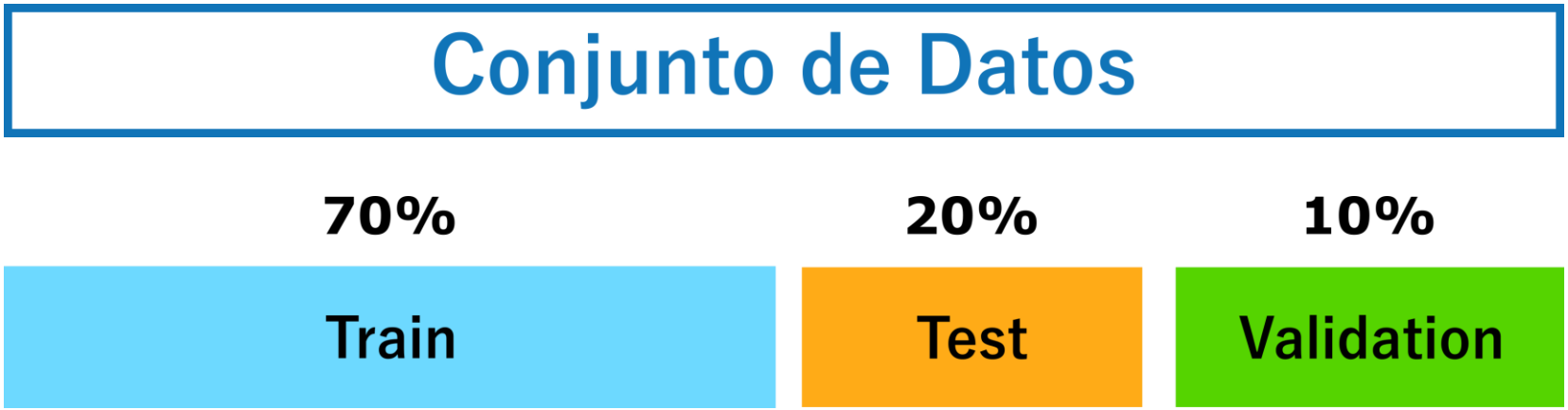
Una vez cada uno de nuestros modelos ha pasado por el proceso de afinación de hiperparámetros y K-Fold, emplearemos entonces el conjunto de validación para seleccionar el Modelo Ganador. En este paso es necesario seleccionar algún criterio para medir el desempeño de los modelos (por ejemplo, la precisión, el recall, etc) aunque generalmente se trabaja con el F1-score.
Aquel modelo final que presente el F1-score más alto, será entonces el modelo final que emplearemos para nuestro problema o bien para predecir regresiones o clasificaciones para nuevos conjuntos de datos desconocidos.

Veamos un ejemplo final a modo ilustrativo: supongamos que tenemos un conjunto de datos con el cual pretendernos clasificar entre dos clases (problema de clasificación binaria). Para ello, vamos a poner a prueba un modelo de Regresión Logística, un modelo de Vectores de Soporte, y un Bosque Aleatorio. Nuestro “flujo de trabajo” debería ser, entonces:

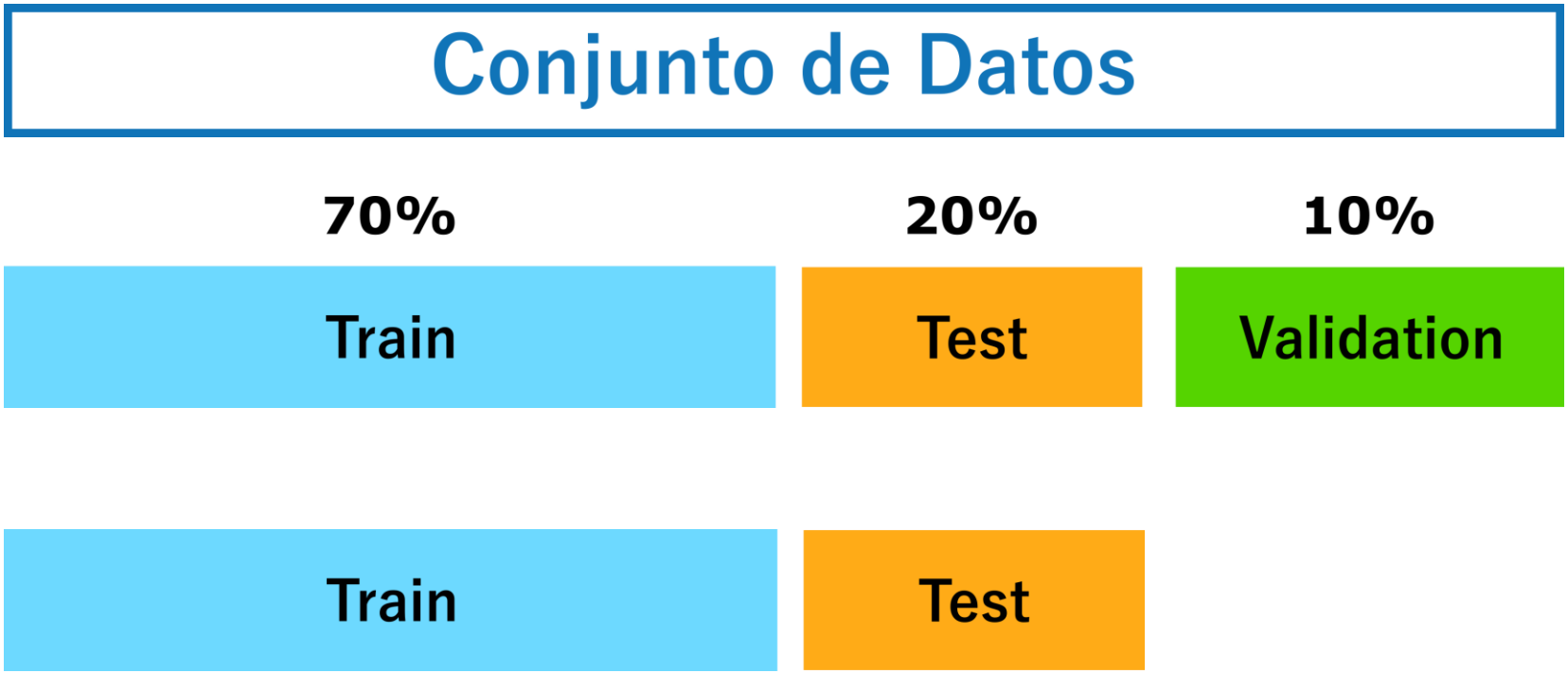
Selección del Modelo Ganador

Conjunto de Datos

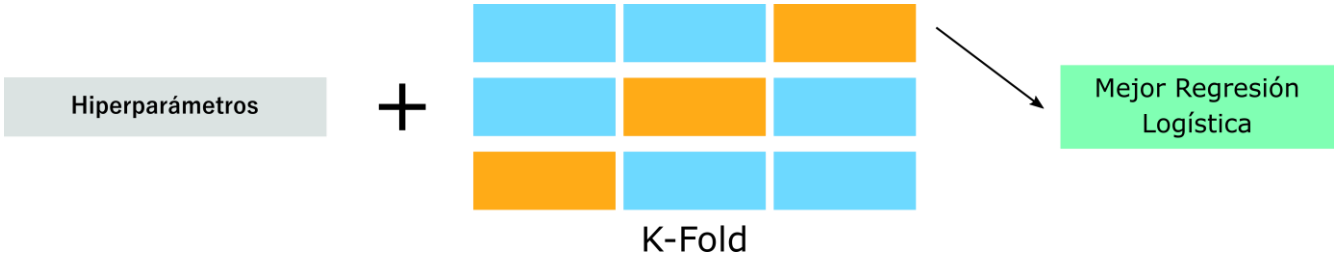
Selección del Modelo Ganador



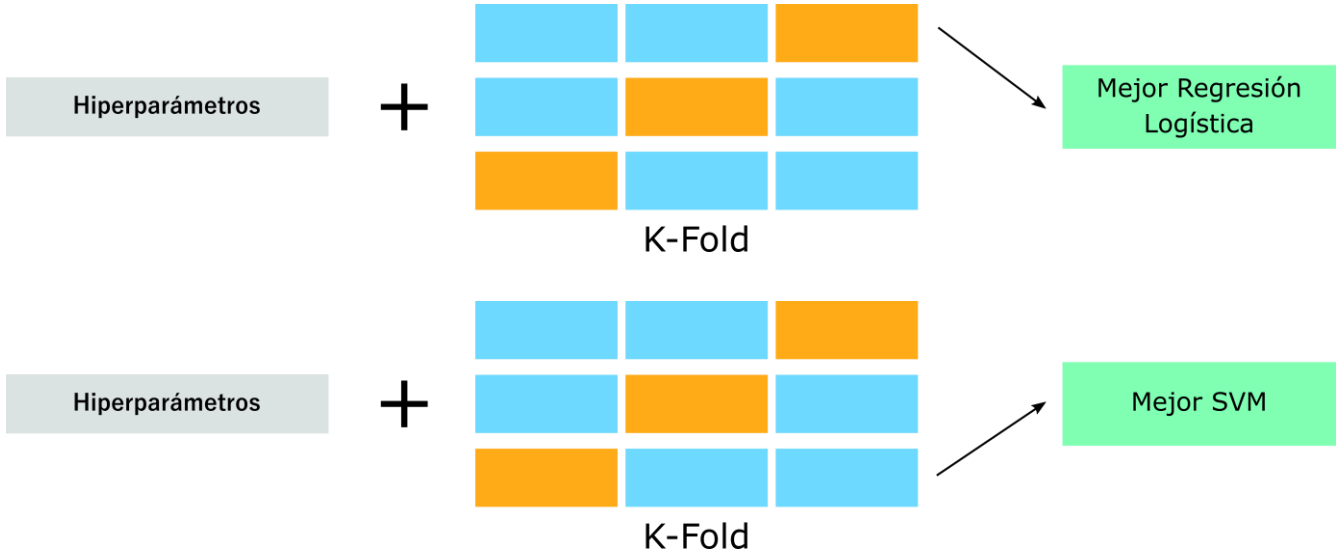
Selección del Modelo Ganador



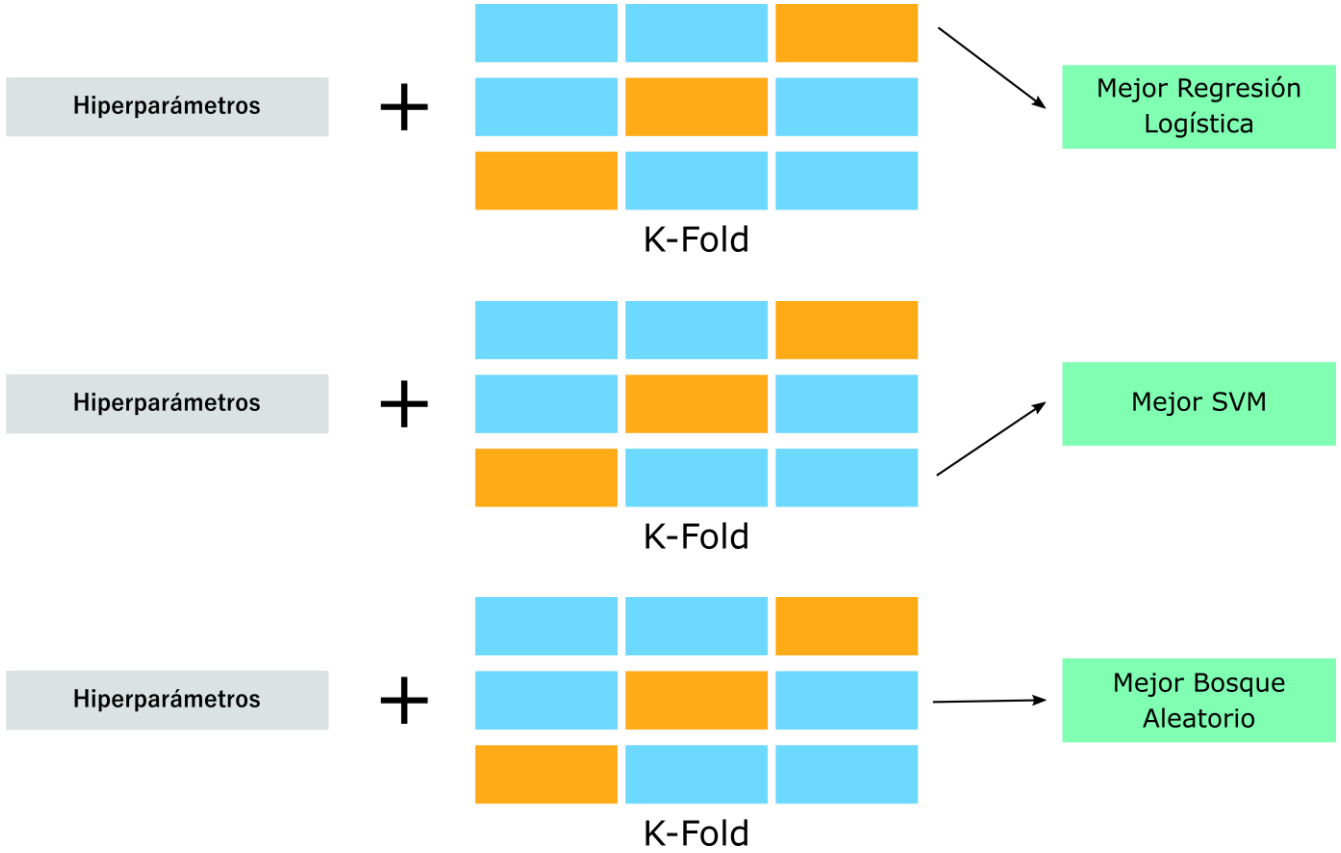
Selección del Modelo Ganador



Selección del Modelo Ganador

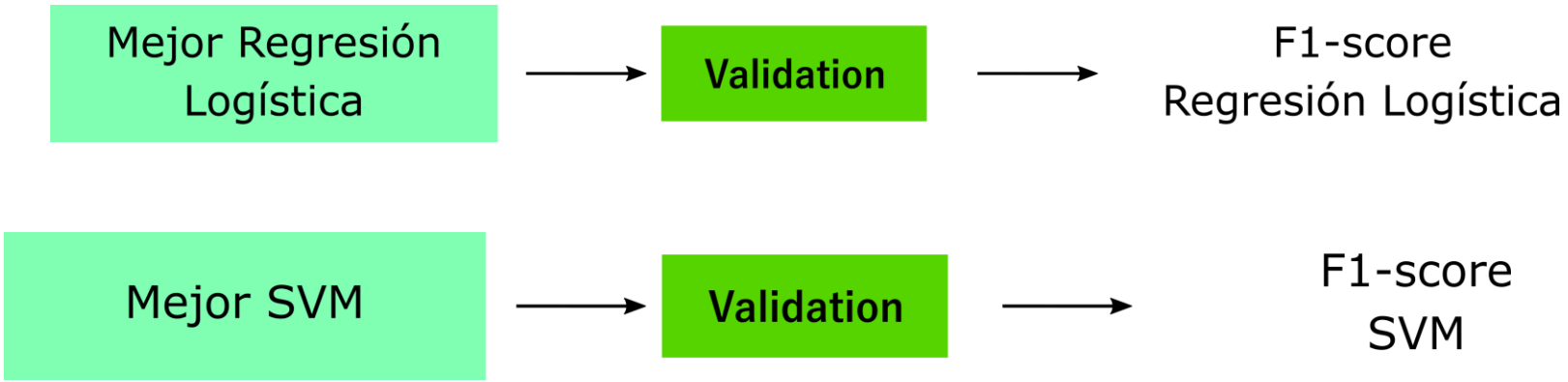


Selección del Modelo Ganador

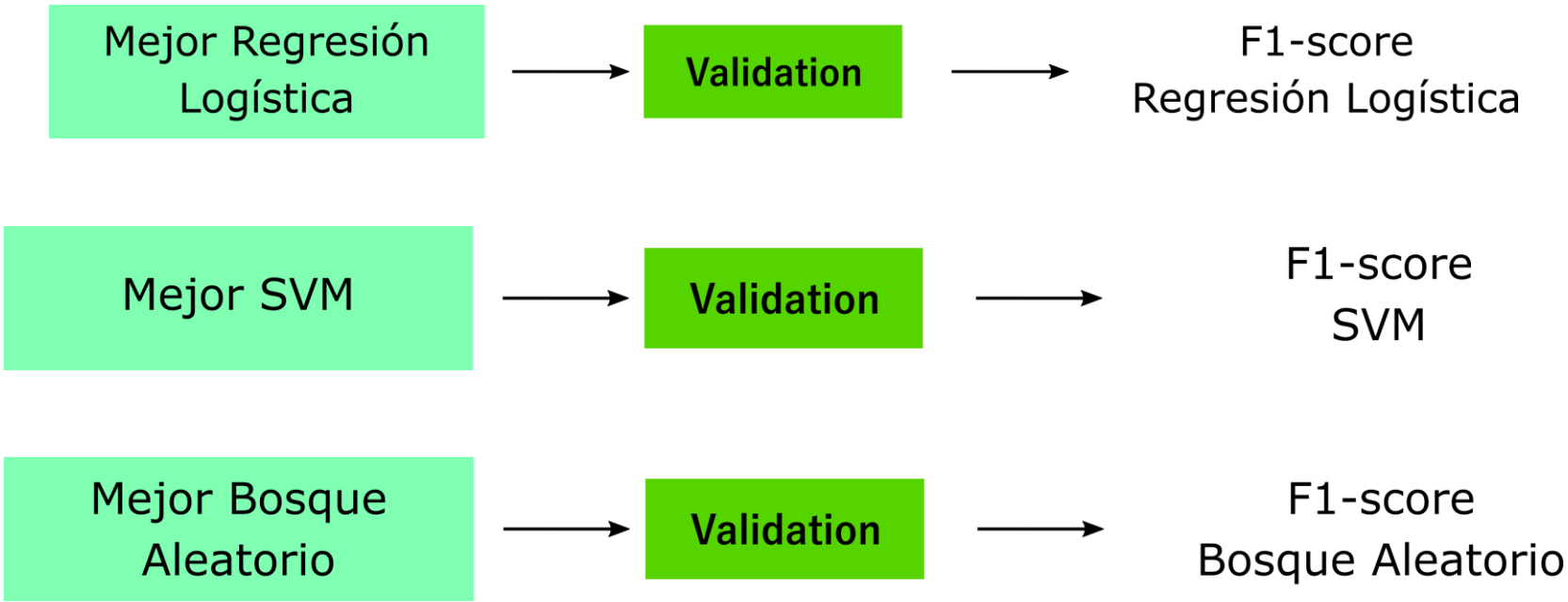


Selección del Modelo Ganador

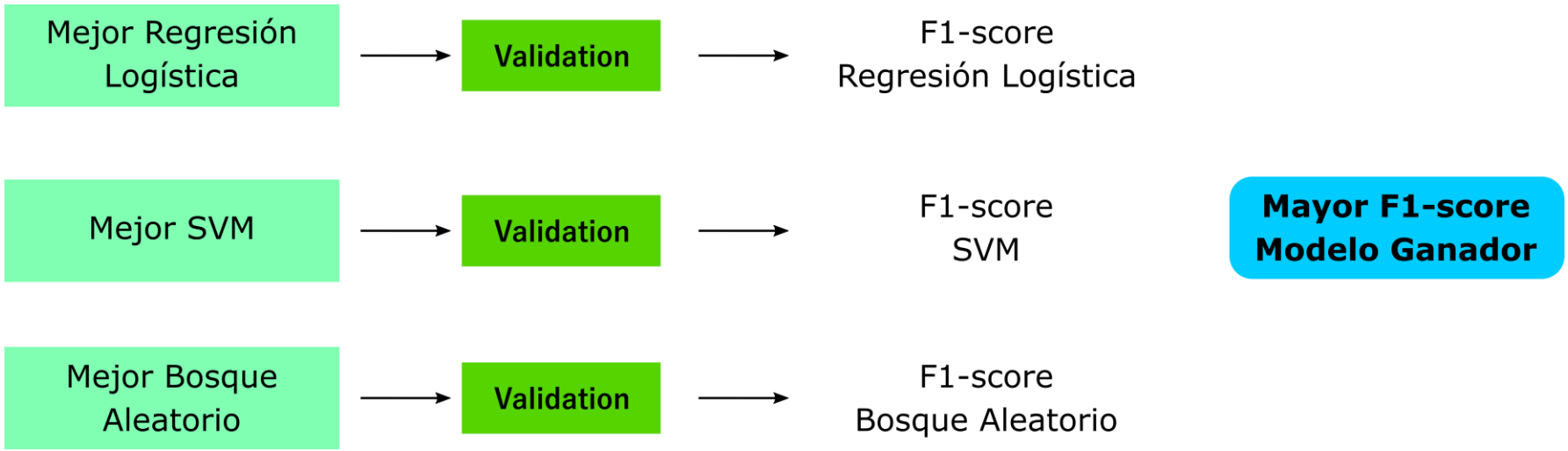
Selección del Modelo Ganador



Selección del Modelo Ganador



Selección del Modelo Ganador



Décimo Noveno Notebook Práctico

Veamos el cómo se implementa la Validación Cruzada K-Fold así como el Grid Search para el caso de un problema de clasificación en Machine Learning.

Décimo Noveno Notebook Práctico