

Aprendizaje automático

Evaluación de modelos de regresión

// Evaluación de modelos

- Dado un modelo de regresión, algo importante a determinar es si el modelo puede generar predicciones acertadas ante nuevas observaciones de las variables independientes.
- En otras palabras, queremos medir la **capacidad de generalización del modelo**.
- Para ello, necesitamos un nuevo conjunto de datos (**conjunto de prueba**), con el cual es posible estimar la exactitud del modelo, así como error de predicción.

No debemos usar los datos de entrenamiento para concluir si un modelo generaliza adecuadamente, ya que los resultados estarían sesgados positivamente.

Sería como preguntar en un examen las mismas preguntas vistas en clase sin cambio alguno.

¿Qué podemos hacer con un conjunto de prueba?

- Calcular alguna medida de error para regresión
 - Error cuadrático medio (MSE)
 - Error absoluto medio (MAE)
- Calcular el valor R^2 (sólo para regresión lineal)
- Analizar los residuos obtenidos con el conjunto de prueba

Medidas de error

- Mean squared error ○ *MSE*

$$MSE(D, \beta) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n (r_i)^2$$

- Mean absolute error ○ *MAE*

$$MAE(D, \beta) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| = \frac{1}{n} \sum_{i=1}^n |r_i|$$

Coeficiente de determinación

- El **coeficiente de determinación** (R^2) es una medida para evaluar si un modelo de regresión lineal se ajusta adecuadamente a un conjunto de datos. Un valor cercano a **1** indica que el modelo lineal es adecuado. Un valor de **0** indica lo contrario.
- Este valor se calcula como:

$$R^2 = 1 - \frac{RSS}{TSS}$$

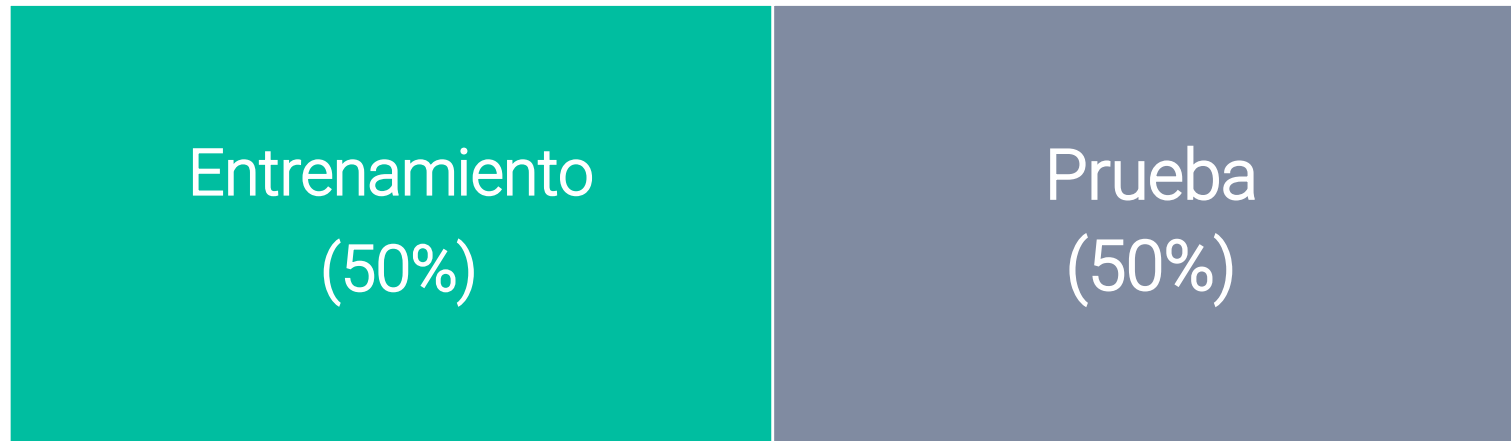
donde

$$RSS(W) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (\text{residual sum of squares o } \textcolor{red}{RSS})$$

$$TSS(W) = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (\text{total sum of squares o } \textcolor{red}{TSS})$$

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (\text{valor medio de } \textcolor{teal}{Y})$$

División del conjunto de datos para entrenamiento y prueba



¿Qué pasa si mi conjunto de datos es pequeño y no lo podemos dividir en dos?

Validación cruzada

- En la práctica, es difícil tener suficientes observaciones para entrenar adecuadamente el modelo y evaluarlo.
- **Validación cruzada** es el método más utilizado para entrenar y evaluar un modelo utilizando **todos los datos disponibles**.
- La idea de validación cruzada es muy simple:
 - **Divide** los datos en dos conjuntos, entrenamiento y prueba.
 - **Entrena** el modelo y **evalúalo** con esta partición.
 - **Repite** lo mismo varias veces.
 - Después de varias repeticiones, **calcula el promedio** de los resultados (promedio de las matrices de confusión calculadas en cada paso), así como la varianza.

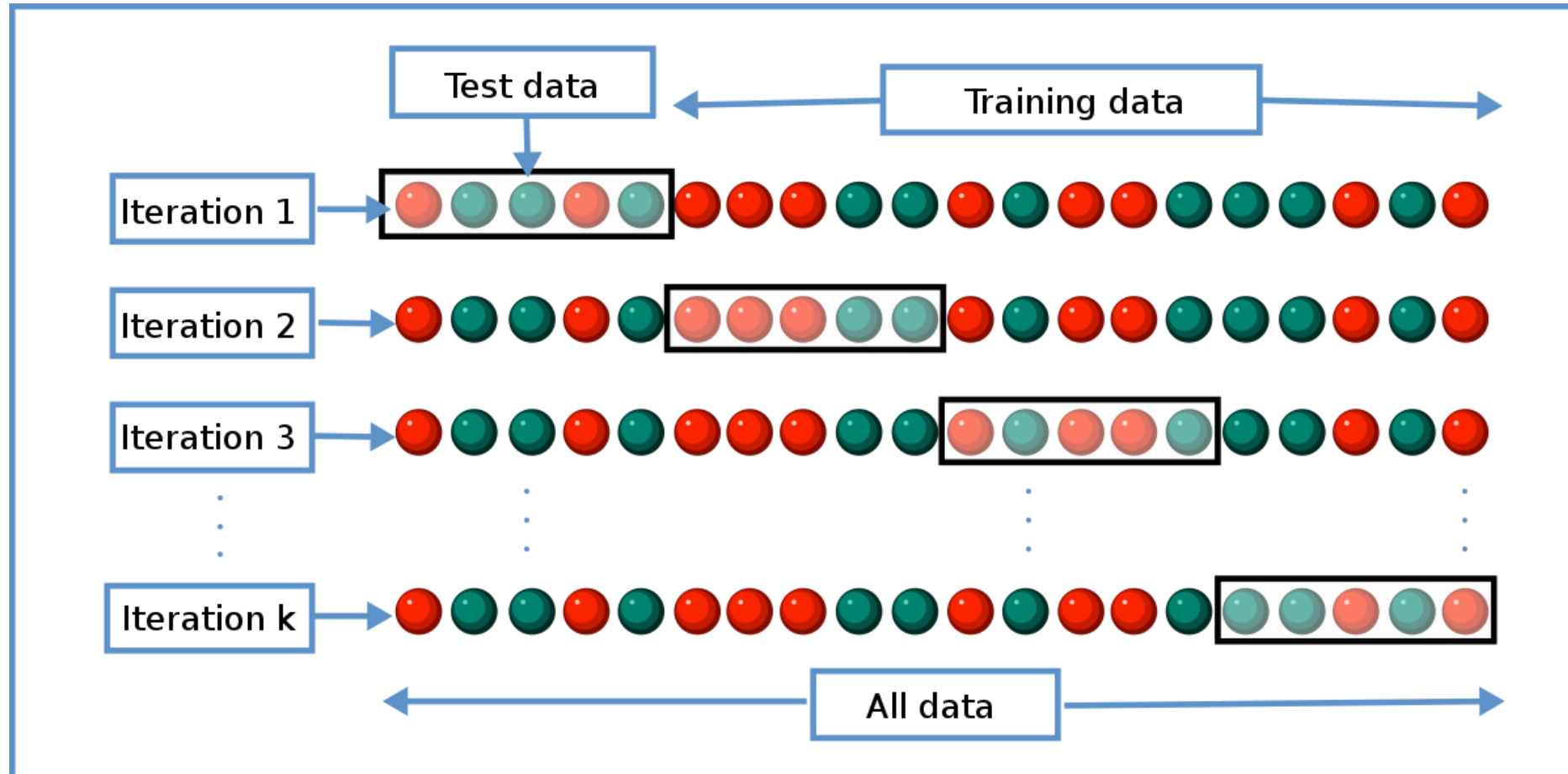
Validación cruzada

- Los métodos **exhaustivos** de validación cruzada prueban todas las formas posibles de dividir los datos disponibles en conjuntos de entrenamiento y prueba.
 - Leave-p-out cross validation, leave-one-out cross validation
- Los métodos **no exhaustivos** de validación cruzada no calculan todas las formas posibles de particionar los datos originales. Estos métodos son aproximaciones de los métodos exhaustivos.
 - K-fold cross validation, hold-out, Monte Carlo

Validación cruzada en k-pliegues

- Existen muchas formas de validación cruzada. La más común se conoce como **validación cruzada en k-pliegues** (**k-fold cross validation**):
 - Divide aleatoriamente el conjunto de datos en k particiones o pliegues.
 - Por cada partición i :
 - Entrena con el resto de las particiones que no sean la partición i .
 - Evalúa el modelo con la partición i .
 - Calcula el rendimiento promedio obtenido en las k evaluaciones.
 - Entrena el modelo con todos los datos.

Validación cruzada en k-pliegues

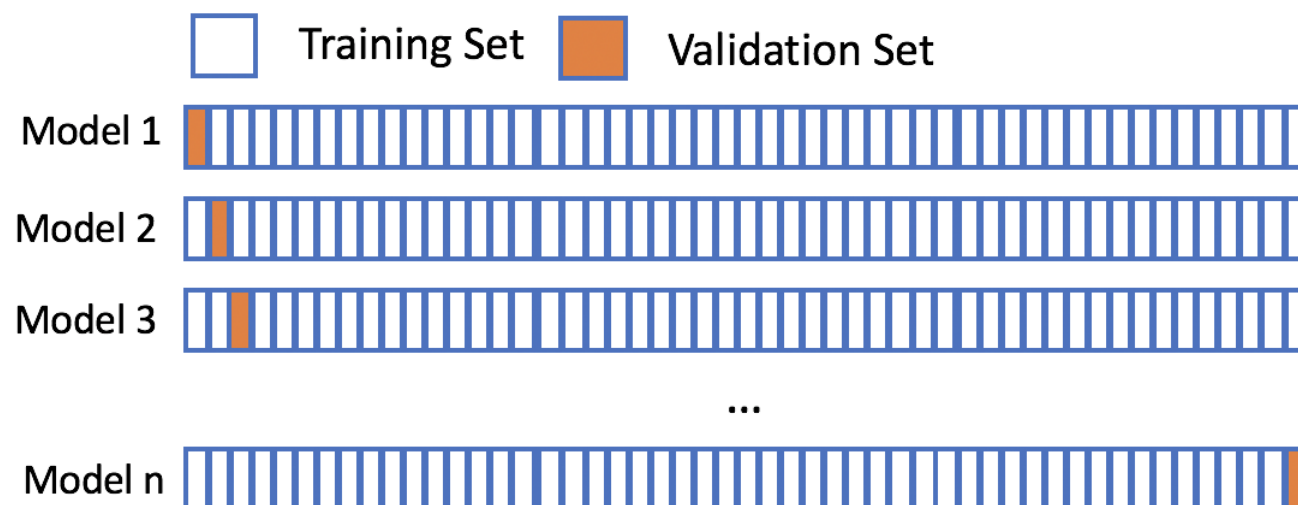


Leave-p-out cross validation

- Leave-p-out cross validation (LpOCV) involucra utilizar p observaciones como conjunto de prueba y el resto como conjunto de entrenamiento.
- Esto se repite para todas las formas posibles en las que se puede particionar el conjunto entre p observaciones para prueba y el resto para entrenamiento.
- LpOCV requiere entrenar y probar el modelo C_p^N veces, donde N es la cantidad de datos el conjunto de datos original.

Leave-one-out cross validation

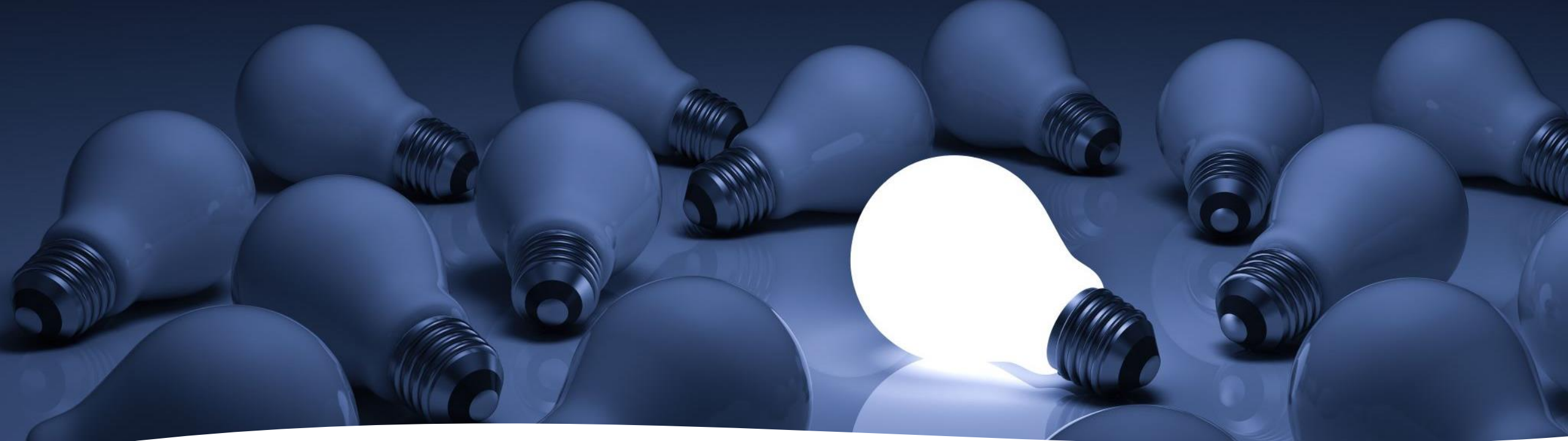
- Leave-one-out cross validation (LOOCV) es un caso especial de L_p OCV cuando $p = 1$. Requiere en total $C_1^N = N$ veces entrenar y evaluar el modelo.



[DataCamp – Cross validation](#)

Otros métodos de validación cruzada

- Método **Holdout**. Se dividen los datos sólo una vez de manera aleatoria (fijando un tamaño para los conjuntos) para entrenar y evaluar el modelo.
- **Método Monte Carlo**. Se dividen los datos aleatoriamente m veces. Para cada partición aleatoria, se entrena y evalúa el modelo.
- En el método Monte Carlo, conforme m se incrementa, el resultado se aproxima al resultado que se observaría con **LpOCV**.



¿Por qué validación cruzada funciona?

- Aun cuando en cada iteración en validación cruzada se esté evaluando un modelo distinto al que se obtendría al entrenar con todos los datos, se supone que dichos modelos **no difieren demasiado**.
- Lo que se intenta con validación cruzada es **evaluar la capacidad de una metodología** de generar un modelo que pueda generalizar adecuadamente.

Bibliografía

- James, G., Witten, D., Hastie, T. & Tibshirani, R. (2023). *An introduction to statistical learning: with applications in Python* (2da ed.). Springer.
 - Capítulo 5
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2da ed.). Springer.
 - Capítulo 7
- Wikipedia. *Cross validation*. [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)).