

09481: Inteligencia Artificial

Profesor del curso: Breyner Posso, Ing. M.Sc.
e-mail: breyner.posso1@u.icesi.edu.co

Programa de Ingeniería de Sistemas.
Departamento TIC.
Facultad de Ingeniería.
Universidad Icesi.
Cali, Colombia.

Agenda

1. Introducción.
2. Errores de sesgo (sub aprendizaje) y varianza (sobre aprendizaje).
3. Impacto de la complejidad del modelo de predicción en los errores de sesgo y varianza, así como en la capacidad de generalización.
4. Protocolos de evaluación (entrenamiento/validación/prueba, validación cruzada, y *bootstrapping*).

1. Introducción

DATOS:

Materia prima.

MODELO:

Implementación del
modelo de analítica.
Ajuste del modelo.

EVALUACION:

Viabilidad del negocio.
Validación criterios de
éxito.

DESPLIEGUE:

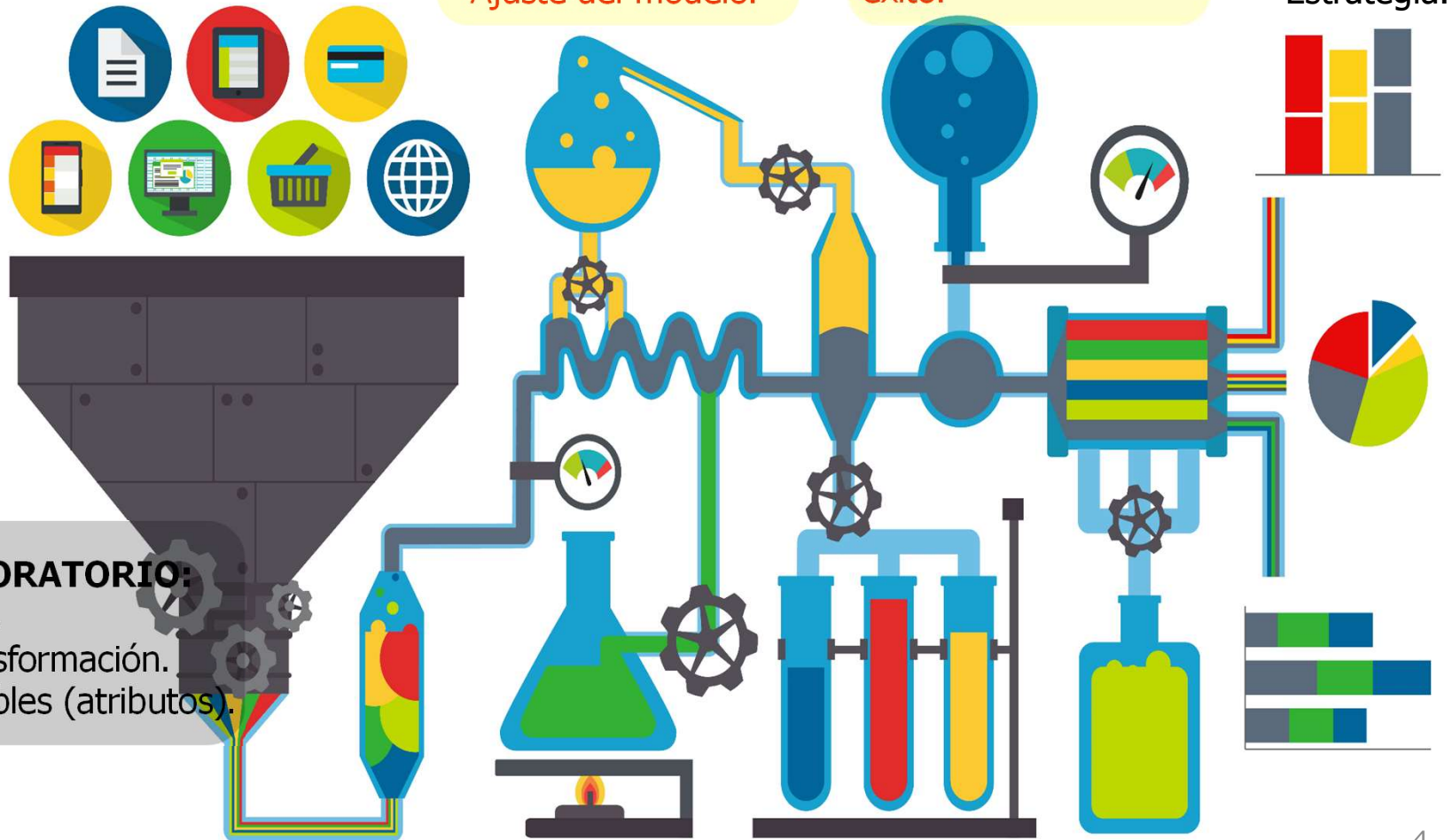
Resultados.
Conocimiento.
Estrategia.

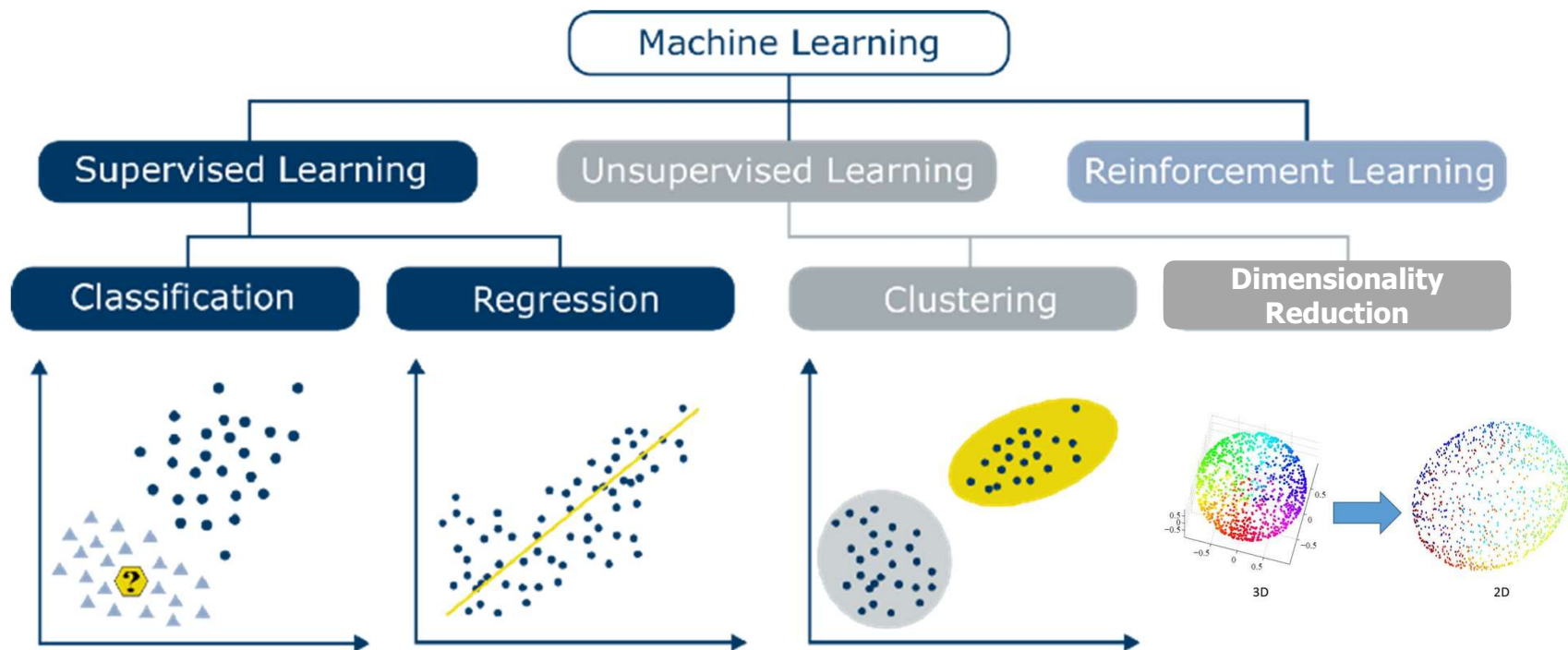
PREGUNTAS:

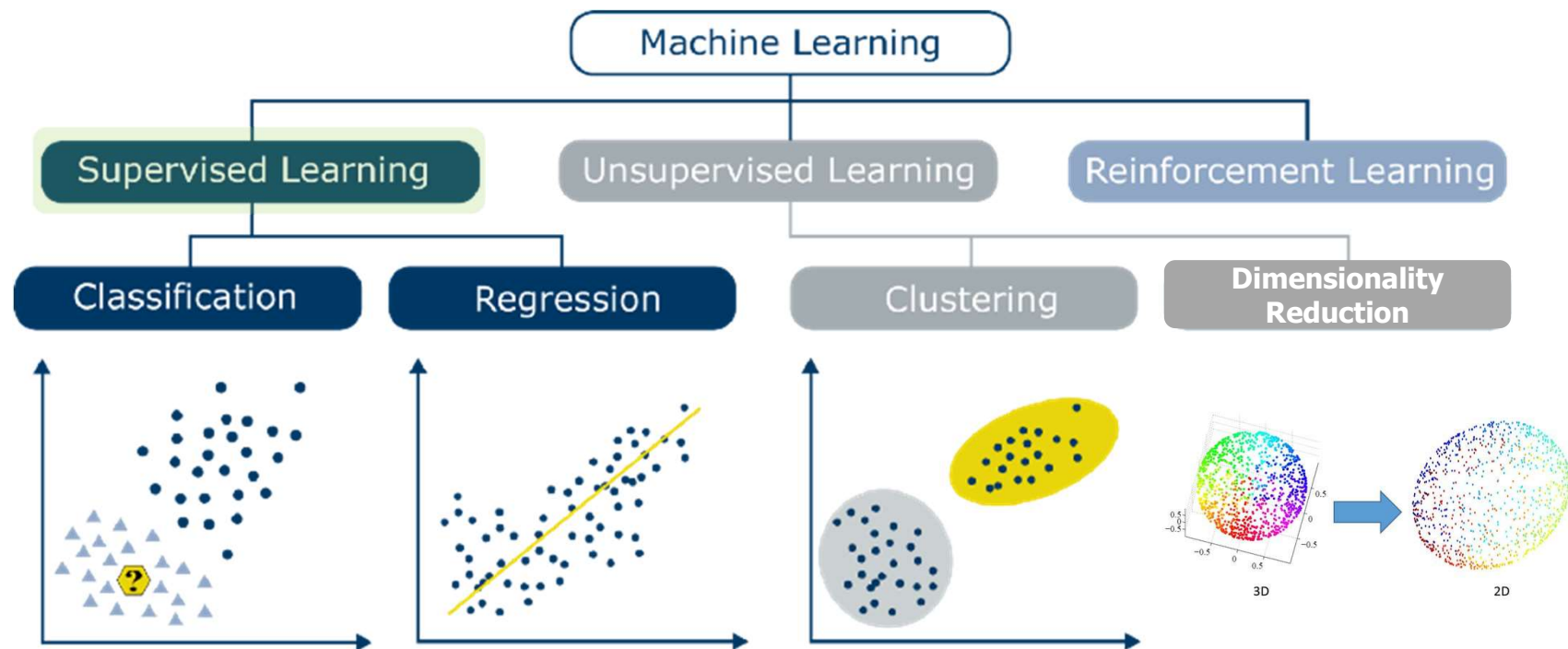
¿Cómo?
¿Cuáles?
¿Cuándo?
¿Por qué? *

ANALISIS EXPLORATORIO:

Limpieza de datos.
Preparación / transformación.
Selección de variables (atributos).



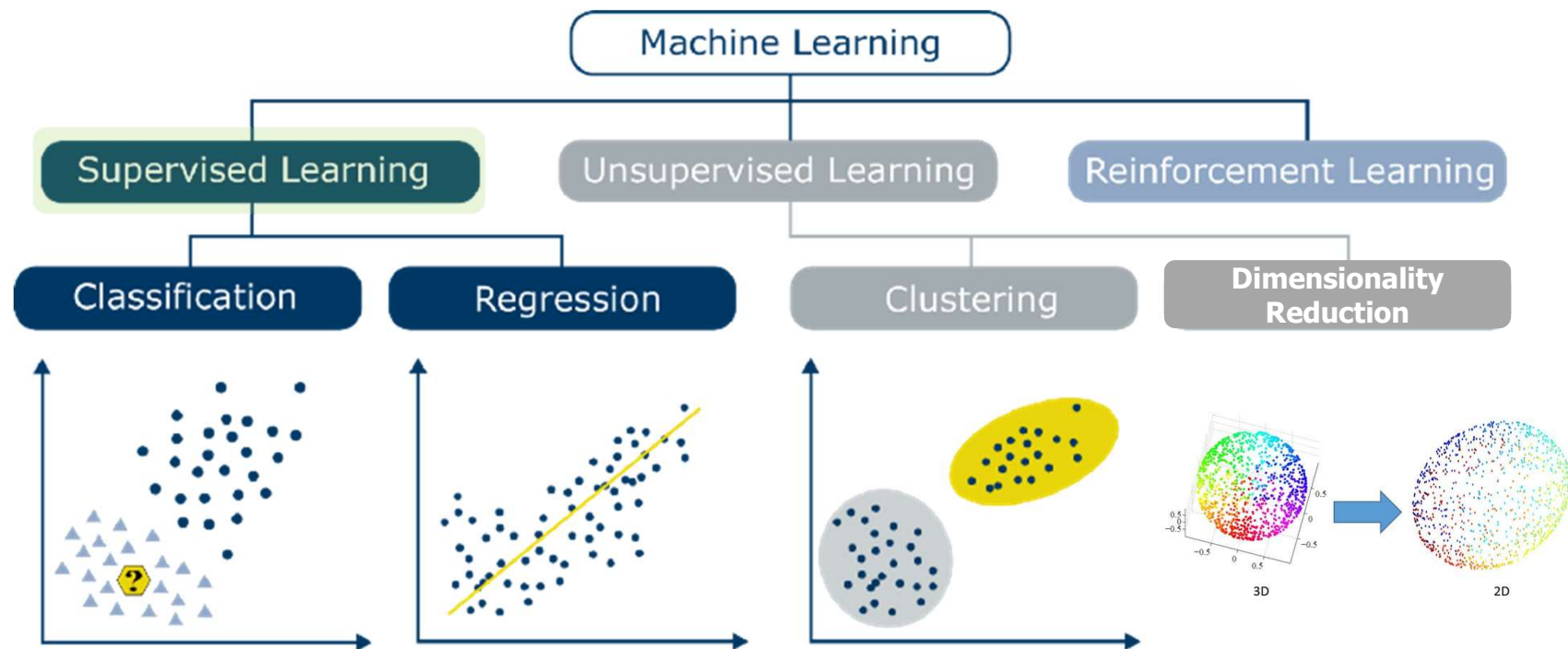




Métodos

KNN

Regresión Lineal Simple
Regresión Lineal Múltiple
Regresión Polinomial



Métodos

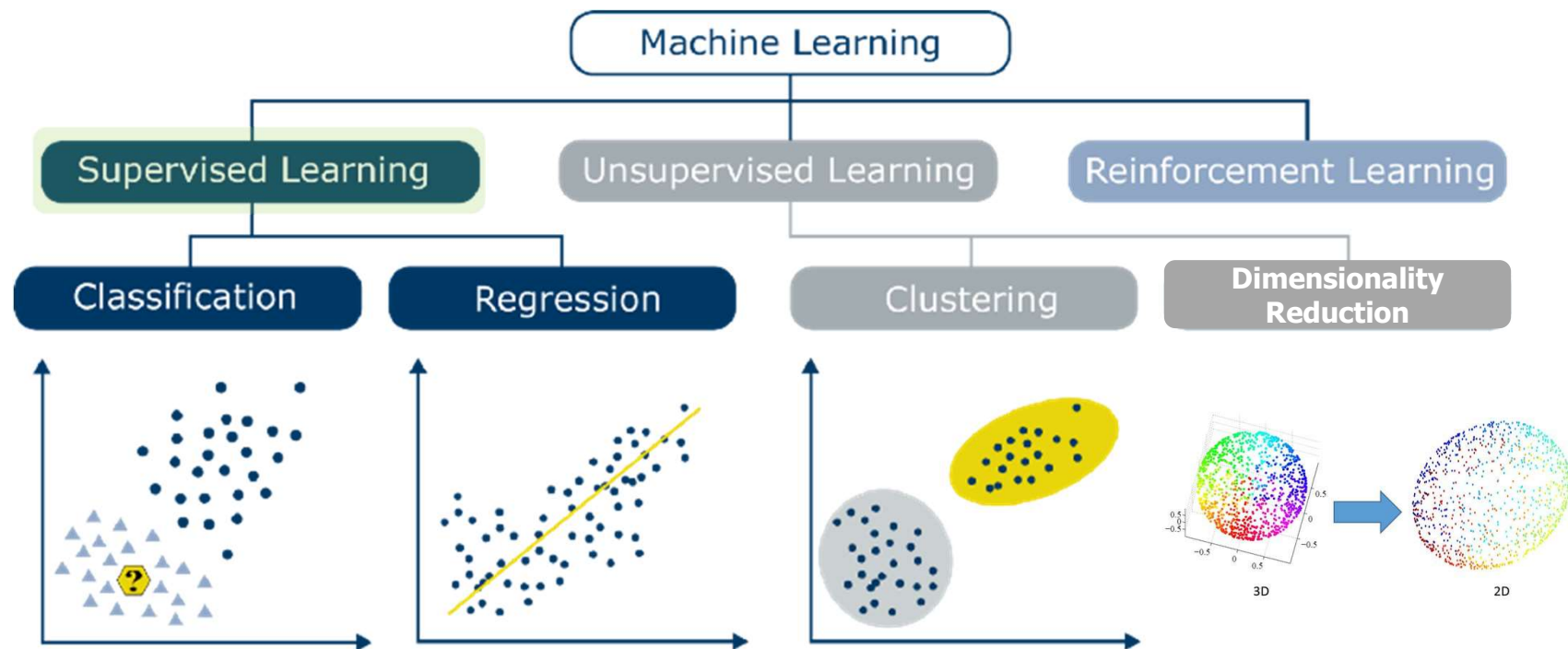
KNN

Regresión Lineal Simple
Regresión Lineal Múltiple
Regresión Polinomial

Evaluación

Accuracy, Precision,
Recall, F1 score,
ROC, etc.

MSE, RMSE, R^2 , etc.



Métodos

KNN

Regresión Lineal Simple
Regresión Lineal Múltiple
Regresión Polinomial

Evaluación

Accuracy, Precision,
Recall, F1 score,
ROC, etc.

MSE, RMSE, R^2 , etc.

Aprendizaje supervisado

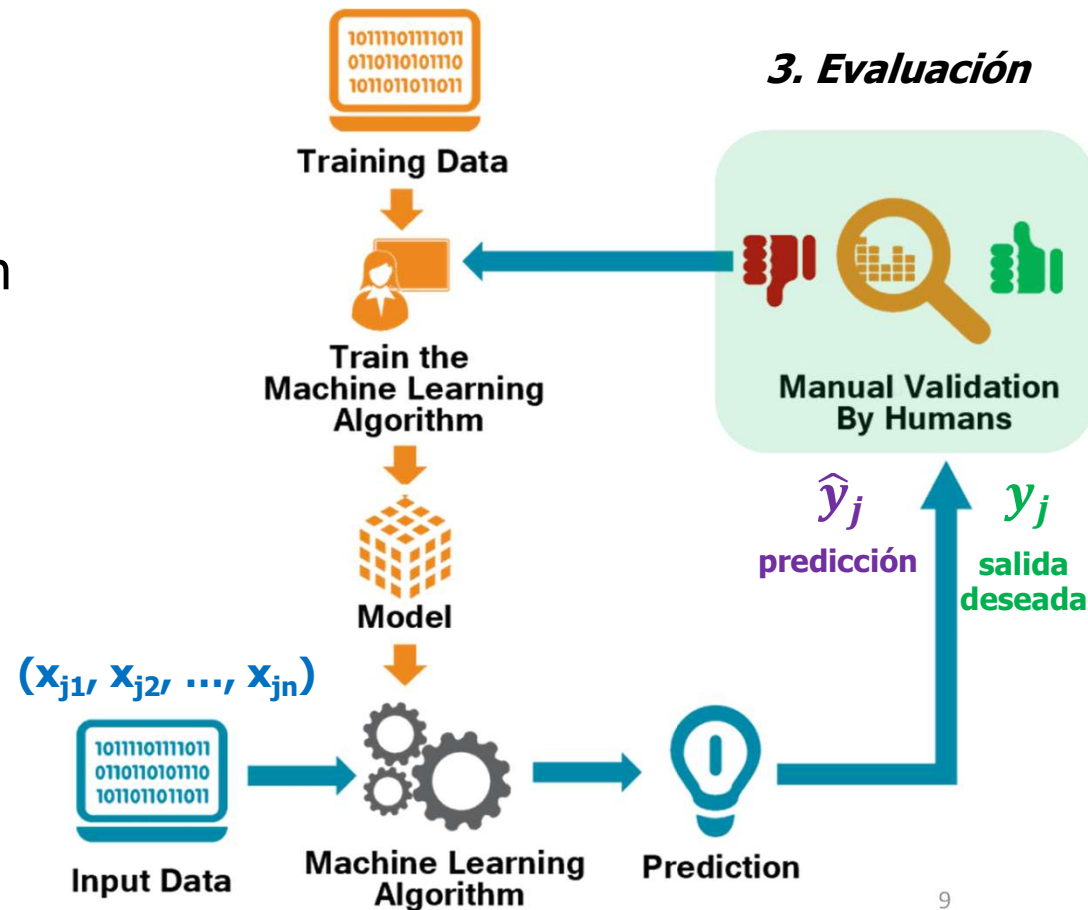
Fase 3: Evaluación

Al tener datos de entrenamiento **etiquetados** con una clase o valor, se puede comparar la salida del modelo con la salida deseada y así obtener un error.

$(x_1, x_2, \dots, x_n, y)$

Predictores, variables explicativas, o variables independientes.

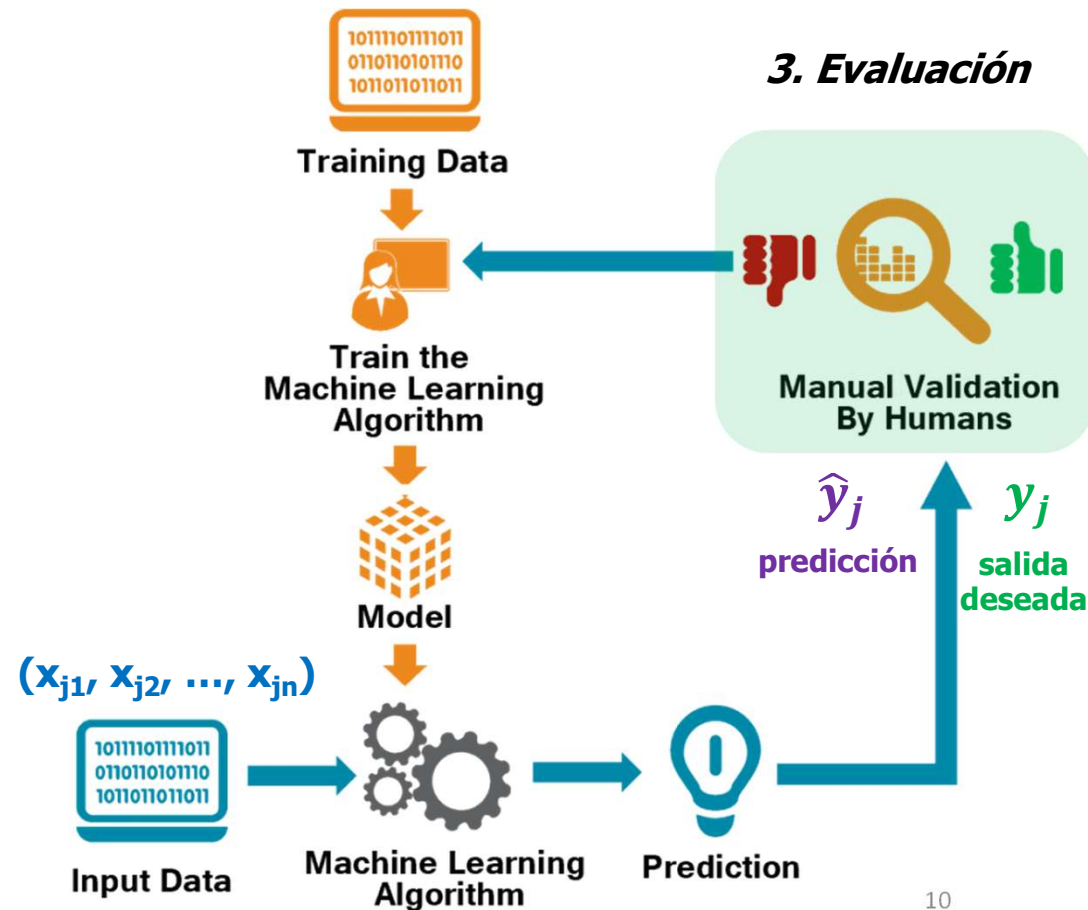
Variable dependiente, objetivo, o Salida deseada.



Aprendizaje supervisado

Fase 3: Evaluación

- La evaluación se realiza primero sobre los datos de entrenamiento y luego sobre los datos de prueba (y validación si es el caso).
- La evaluación sobre los datos de entrenamiento sirve para identificar si el modelo pudo “aprender” la relación entre los predictores y la variable objetivo.
- La evaluación sobre los datos de validación y prueba (datos que no se usaron para entrenamiento) se hace para determinar si el modelo generaliza bien.

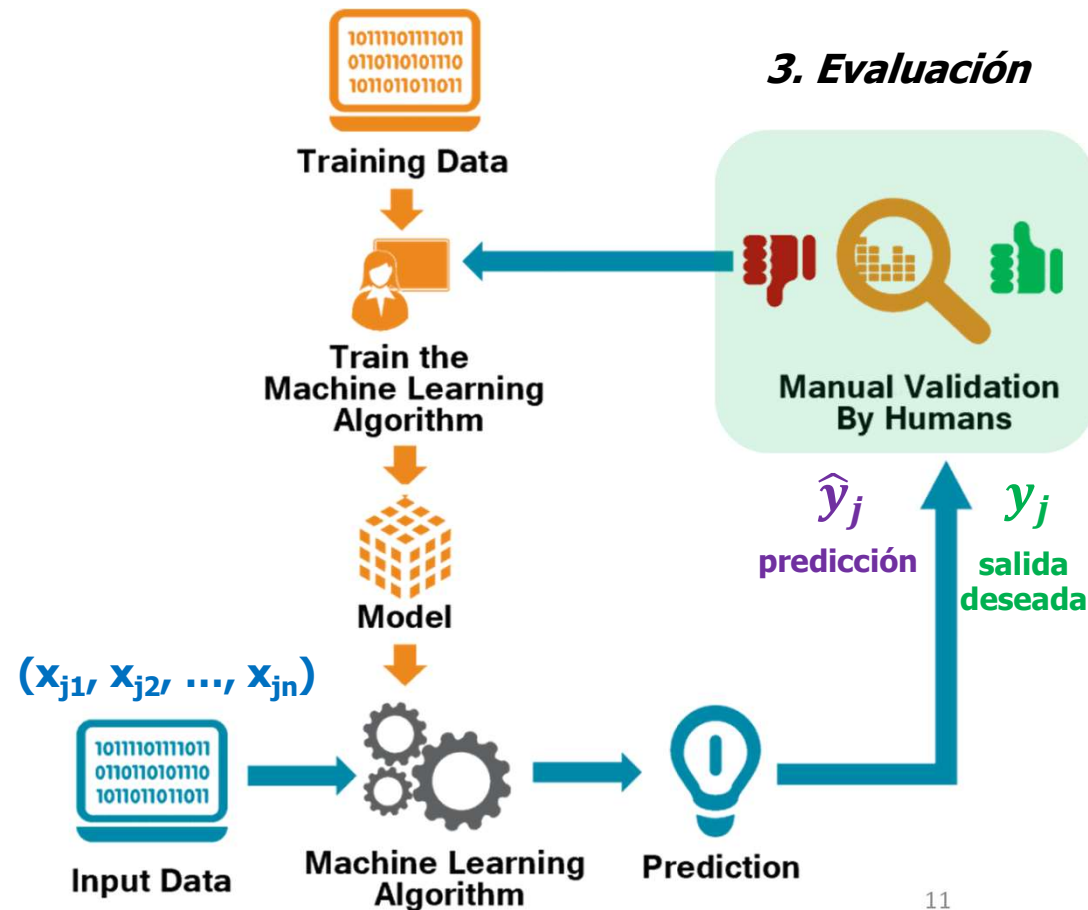


Aprendizaje supervisado

Fase 3: Evaluación

Al evaluar un modelo de aprendizaje supervisado puede presentarse una de las siguientes situaciones:

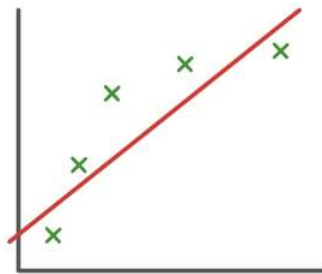
- Sub aprendizaje (underfitting)
- Ajuste óptimo.
- Sobre aprendizaje (overfitting)



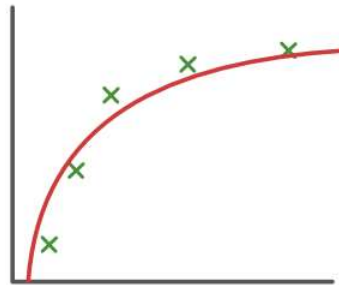
Aprendizaje supervisado

Regresión

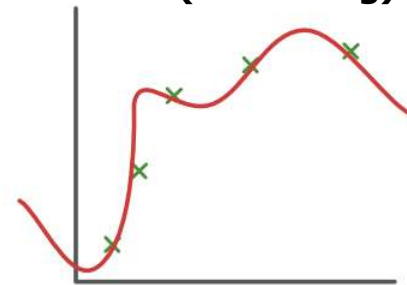
Sub aprendizaje
(underfitting)



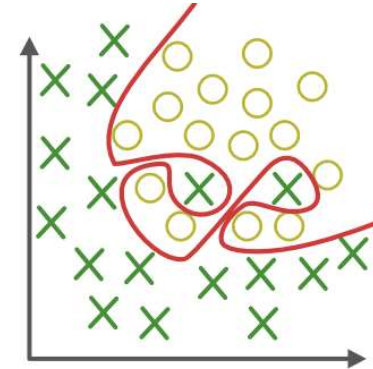
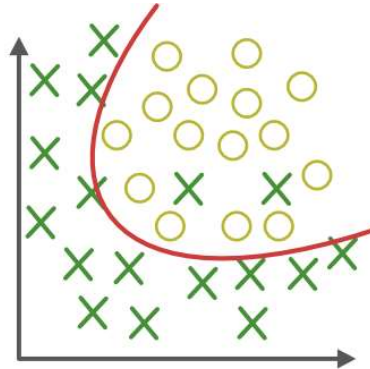
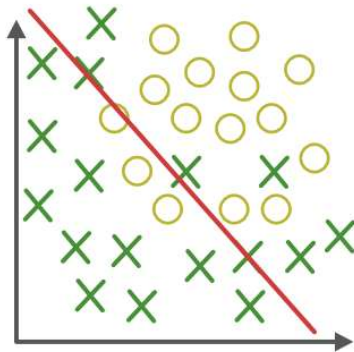
Ajuste óptimo



Sobre aprendizaje
(overfitting)



Clasificación



Evaluación en datos
de entrenamiento

Pobre.

Buena.

Muy buena.

Evaluación en datos de
validación y prueba

Pobre

Buena.

Pobre.

Aprendizaje supervisado

Surgen las siguientes preguntas:

¿De qué depende que un modelo tenga sub aprendizaje (underfitting), ajuste óptimo o sobre aprendizaje (overfitting)?

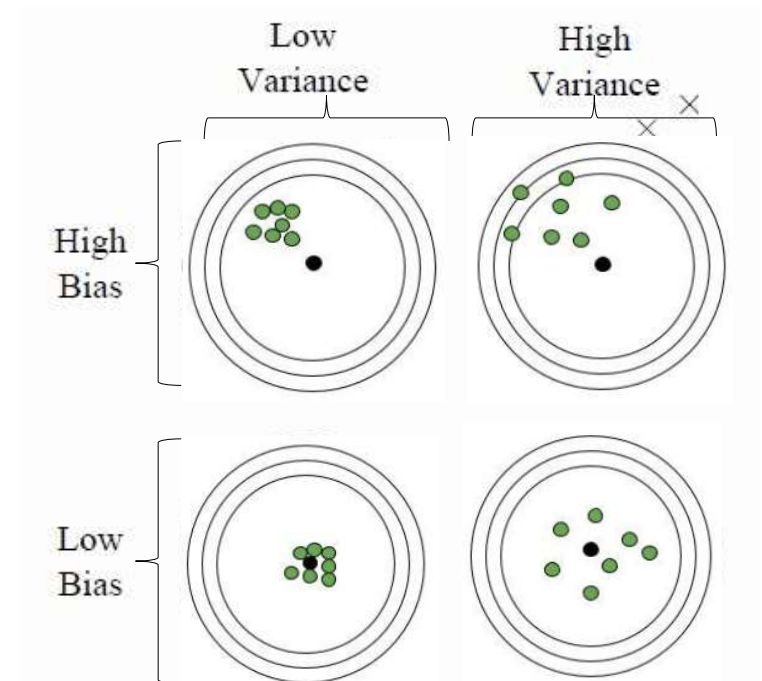
¿Cómo podemos contrarrestar el sub aprendizaje (underfitting) y el sobre aprendizaje (overfitting)

2. Errores de sesgo (sub aprendizaje) y varianza (sobre aprendizaje).

Sesgo y varianza: intuición

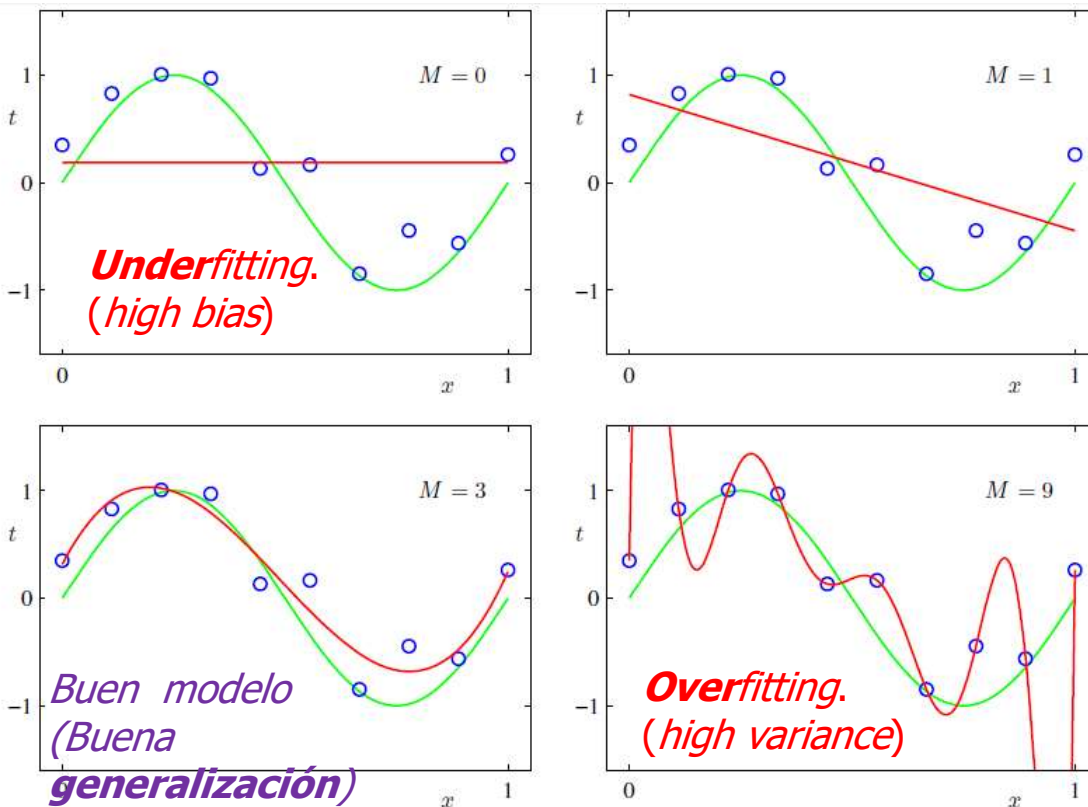


- **Sesgo (*bias*)**: ¿Qué tan lejos están del valor deseado las predicciones que hace el modelo a partir de las observaciones de entrada?
- **Varianza (*variance*)**: ¿Qué tanto varían las predicciones del modelo ante pequeñas variaciones en las observaciones de entrada?



Fuente: Domingo, 2012.

Sesgo y varianza: intuición



Modelo de regresión polinomial de orden M :

$$\hat{y} = h(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

x : Atributo de entrada.

\hat{y} : Salida predicha.

w_j : Coeficientes del modelo.

○: Representación de las observaciones reales (i.e.: ruidosas).

—: Salida ideal (no se conoce).

—: Predicción del modelo.

Dependiendo de la **complejidad** del modelo de predicción (que para este caso depende del orden del polinomio M) debemos encontrar un balance entre:

Sobre aprendizaje (sobre ajuste o *overfitting*).

y

Sub aprendizaje (sub ajuste o *underfitting*).

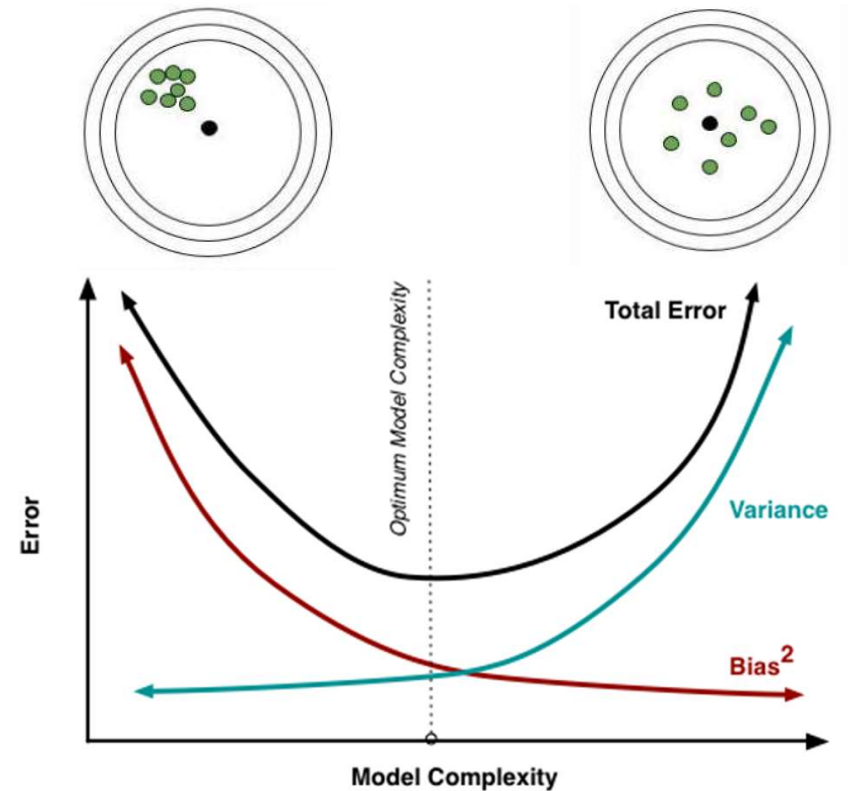
Fuente: Bishop, C. M. (2006).

Sesgo y varianza

- El sesgo y la varianza son fuentes de error.

$$\text{Error}(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible_Error}$$

- Debemos encontrar un **balance** (*trade-off*) entre ambos tipos de errores a partir de la complejidad del modelo para lograr una buena **generalización** durante las pruebas.
- Los parámetros de los modelos controlan la complejidad.



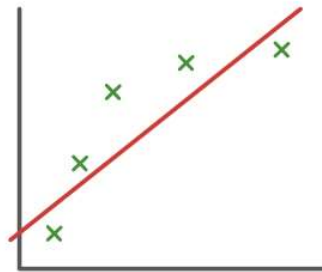
Fuente: <http://scott.fortmann-roe.com/docs/BiasVariance.html>.

Sesgo y varianza

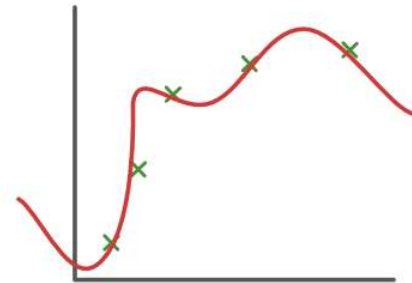
¿Cómo es el sesgo y la varianza en estos modelos?

Regresión

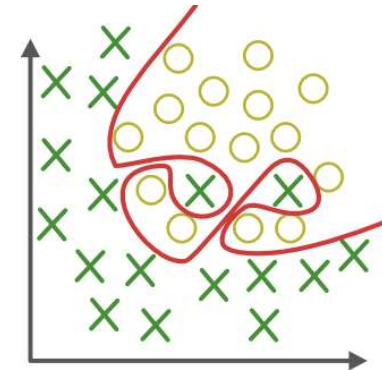
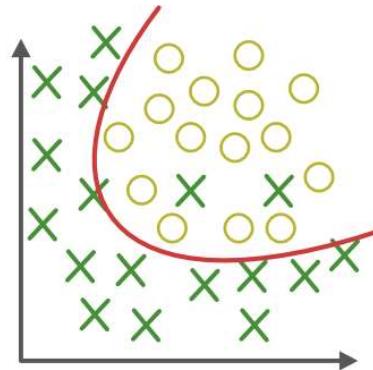
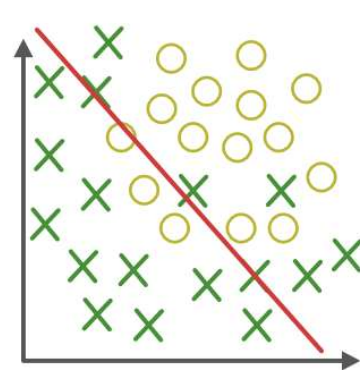
**High Bias.
Low Variance.**



**Low Bias.
High Variance.**



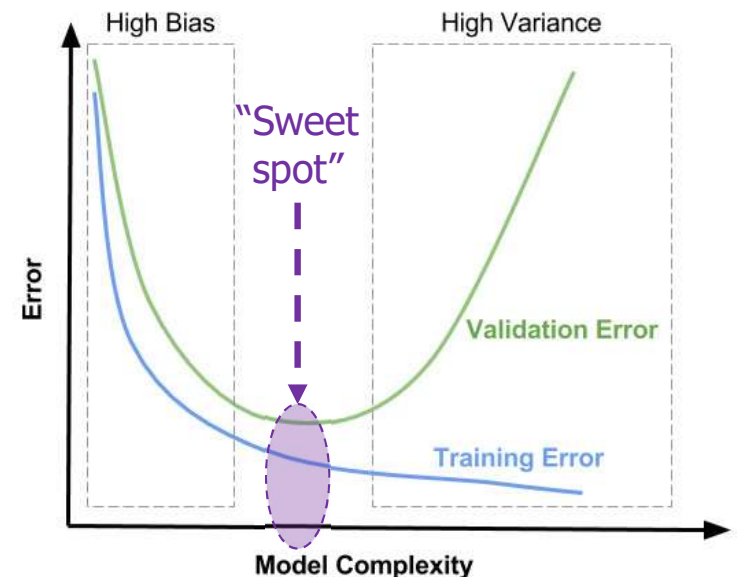
Clasificación



3. Impacto de la complejidad del modelo de predicción en los errores de sesgo y varianza, así como en la capacidad de generalización.

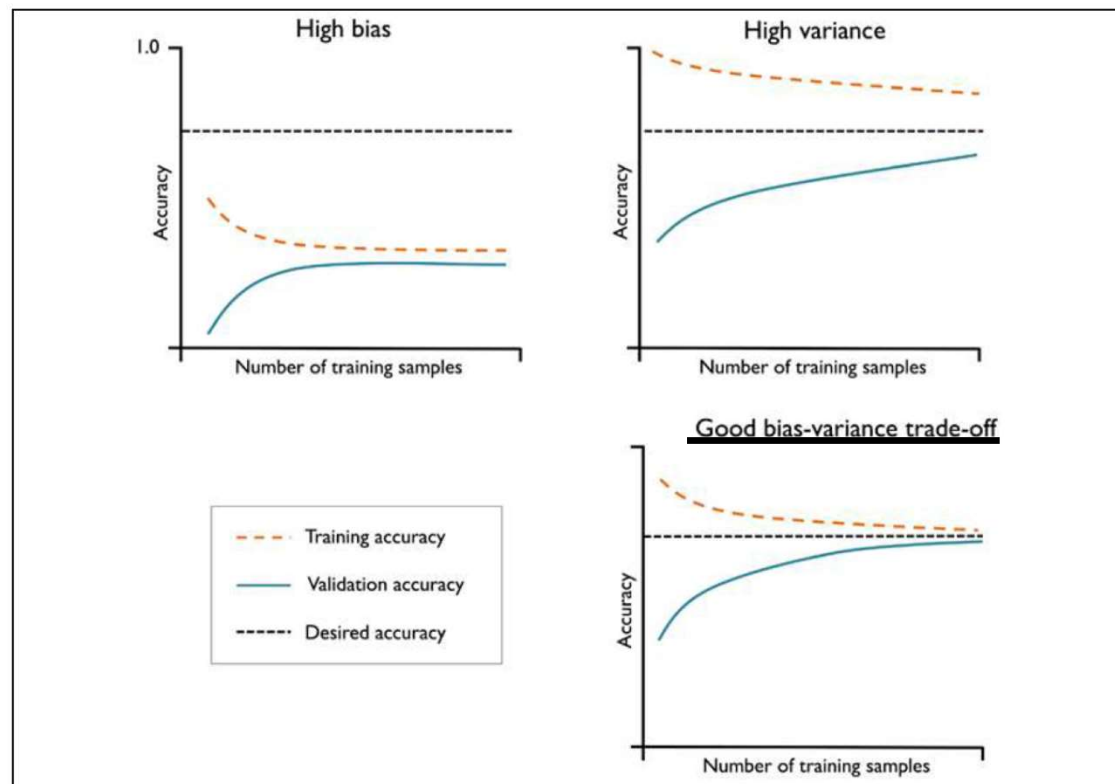
Complejidad del modelo y generalización [1/2]

- Los modelos tienden a ajustarse al conjunto de datos con que se entrenen. Por ende, el **error de entrenamiento (*training error*)** **no** es un buen predictor del desempeño futuro del modelo ante nuevos datos de entrada.
- Queremos encontrar una complejidad adecuada para el modelo que nos permita obtener un buen desempeño en otros conjuntos de datos. Para ellos podemos revisar adicionalmente el **error de validación (*validation error*)**.



Fuente: <https://www.linkedin.com/pulse/bias-variance-tradeoff-machine-learning-satya-mallick/>

Complejidad del modelo y generalización [2/2]



Notas:

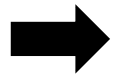
- En casos de sesgo alto, obtener más ejemplos no va a ayudar.
- Por el contrario, en casos de alta varianza, obtener más ejemplos si puede ayudar.

Fuente: Python Machine Learning, Packt (2019).

4. Protocolos de evaluación (entrenamiento/validación/prueba, validación cruzada, y *bootstrapping*).

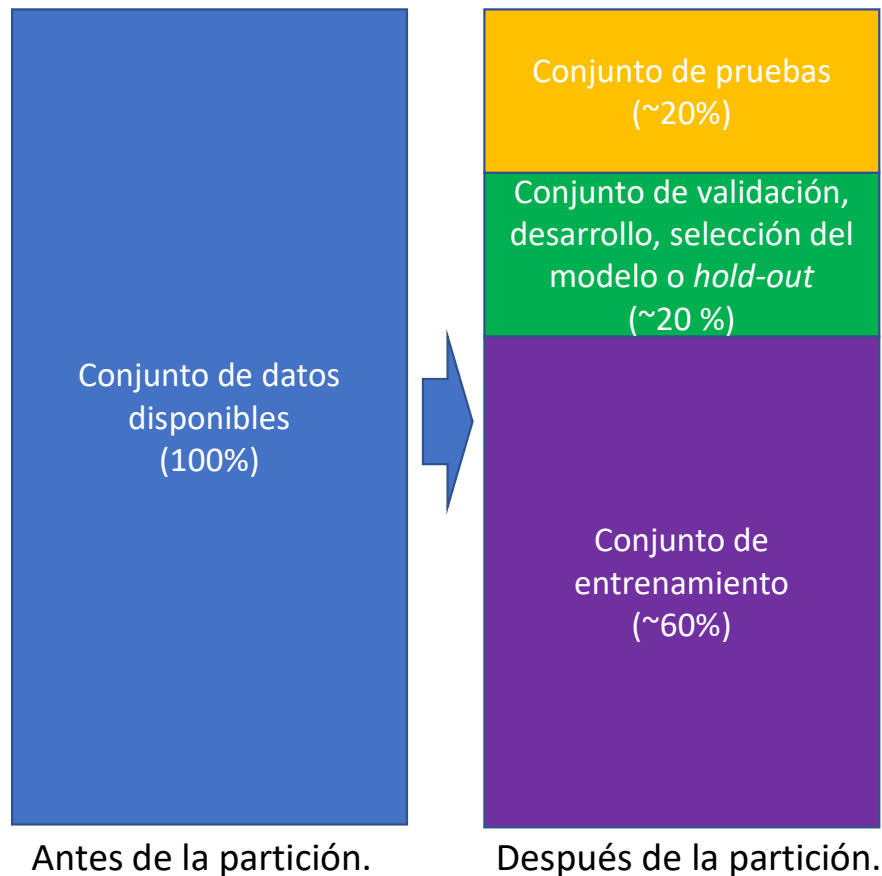
Protocolos de Evaluación

- Aplican para aprendizaje supervisado “en general” (tanto para clasificación como para regresión).
- Buscan:
 - ✓ Evitar el sesgo causado por una **subestimación del error (i.e.: menor que el real)** si se evalúa con el conjunto de entrenamiento.
 - ✓ Diferenciar entre el **error de entrenamiento y el error de validación**.
 - ✓ Establecer un compromiso entre sesgo y varianza, evitando el **sobre aprendizaje**, y buscando un modelo con buenas **capacidades predictivas**.
 - ✓ Evaluar cuál sería la capacidad de **generalización** del modelo ante datos nuevos.



Vamos a revisar algunos métodos de remuestreo: *train/val/test*, *cross-validation*, *bootstrapping*.

Protocolos de Evaluación: *train/val/test*



Conjunto de entrenamiento: es el conjunto con el que aprende el modelo.

Conjunto de validación: este conjunto sirve para seleccionar un buen modelo (i.e.: permite encontrar el balance entre sesgo y varianza), ajustar hiperparámetros, y comparar modelos de diferentes familias.

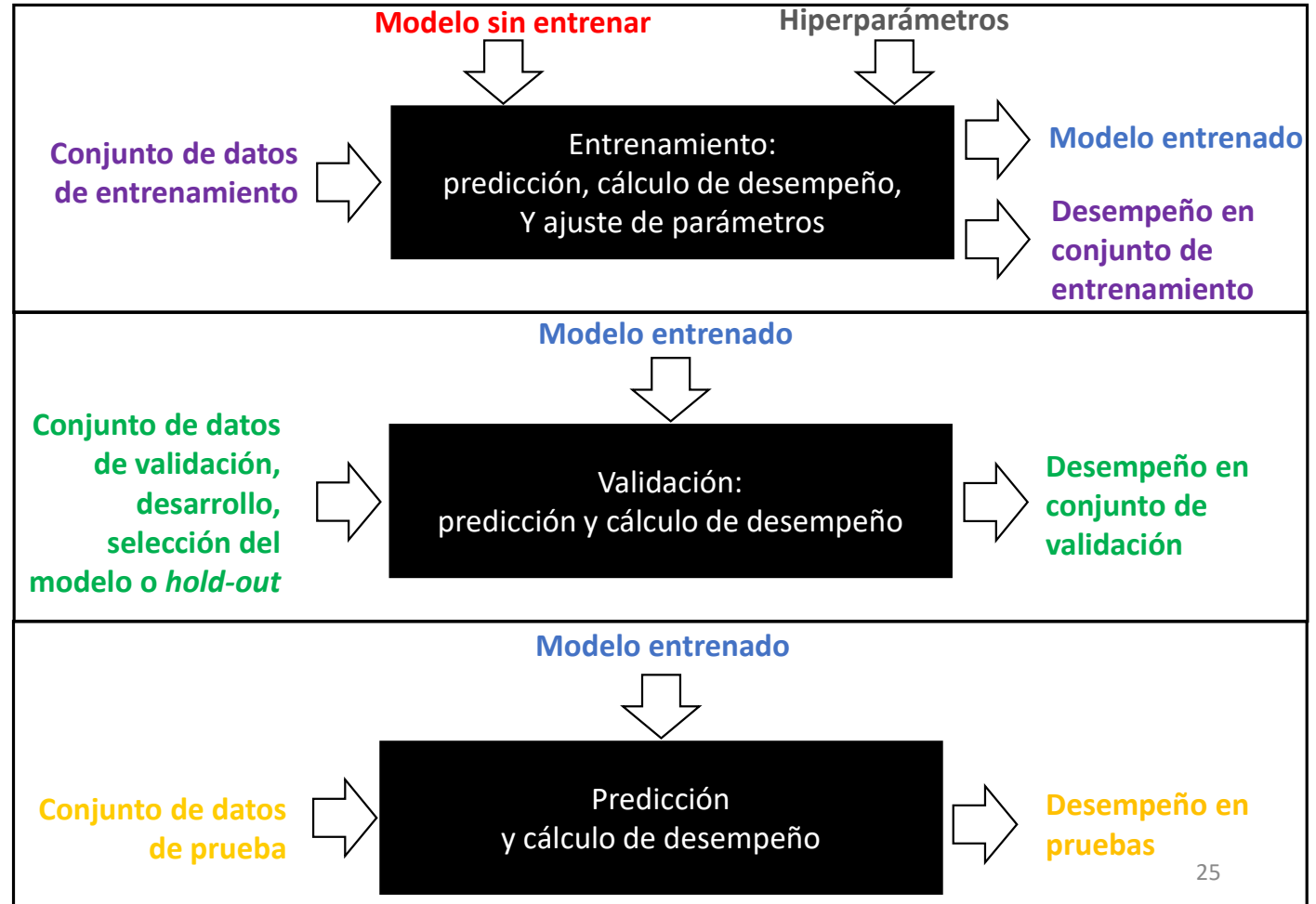
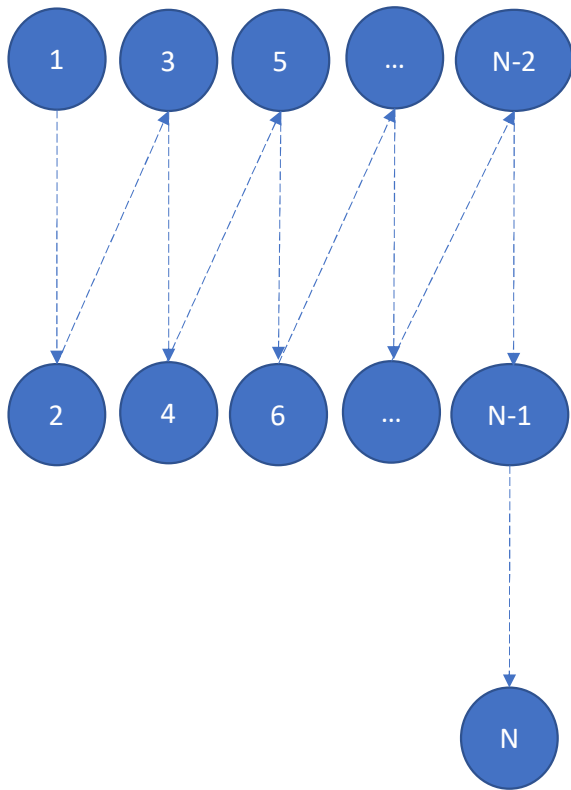
Conjunto de pruebas: sirve para reportar las métricas de desempeño del modelo ante datos nuevos.

Notas sobre las particiones:

- ✓ Aleatoriedad del particionamiento manteniendo la distribución de los datos (e.g.: porcentajes de cada categoría).
- ✓ Compromisos: entre más datos mejor el aprendizaje, y entre más datos mejor la evaluación.
- ✓ Los porcentajes de cada partición pueden variar dependiendo de la cantidad de datos que se tengan.
- ✓ Si se tienen muchos datos, quizás no sea necesario el conjunto de pruebas y sea suficiente el conjunto de validación.

Protocolos de Evaluación: *train/val/test*

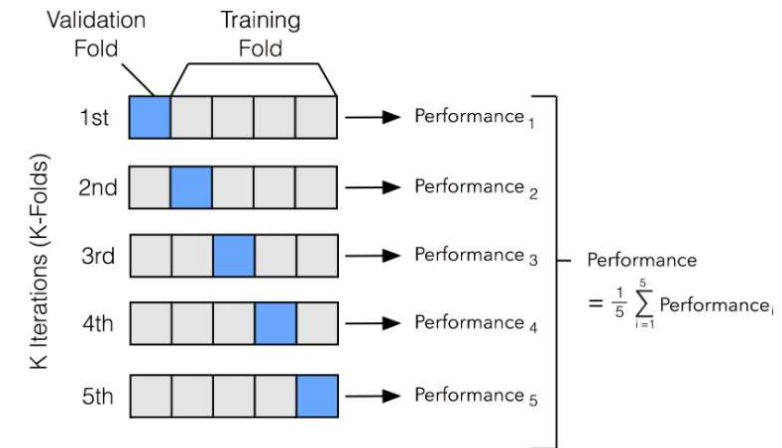
Pasos entre etapas (ilustrativo):



Protocolos de Evaluación: *K-fold Cross-Validation*

- **Validación cruzada de K iteraciones:**

- Se particiona el conjunto de datos de entrada en K conjuntos del mismo tamaño de forma tal que no se traslapen.
- Se seleccionan $K-1$ conjuntos y se usan para el entrenamiento del modelo. El conjunto restante se usa para validación, con él se extraen las métricas de desempeño asociadas a esa iteración.
- El proceso se repite K veces cambiando la selección de los conjuntos hasta que el conjunto de validación haya estado en cada una de las particiones.
- Al final se promedia el desempeño que se obtuvo en las K iteraciones.



Fuente: <https://sebastianraschka.com/blog/2016/model-evaluation-selection-part3.html>

Notas:

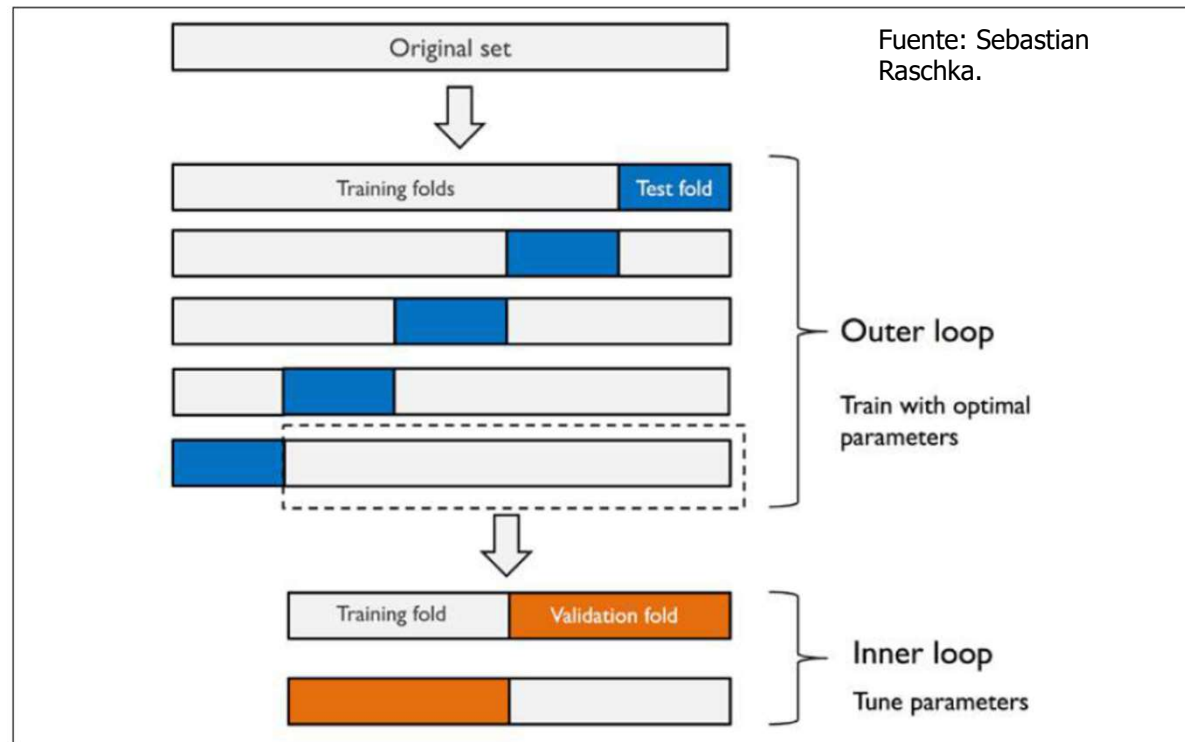
- Valores usuales para $K=\{5, 10\}$. Estos son valores empíricos que proporcionan una estimación del error de prueba que no sufre por errores demasiado grandes de sesgo ni de varianza.
- Cuando $K=m$ = número de observaciones, se obtiene LOOCV (*Leave-One-Out Cross-Validation*). En este caso cada partición tiene asociada una única observación. Puede ser razonable para conjuntos de datos medianos, pero para conjuntos de datos muy grandes es exigente computacionalmente dado que el modelo se debe entrenar m veces.

Protocolos de Evaluación: *Stratified K-fold CV*

- En la validación cruzada **estratificada**, las proporciones de las etiquetas de clase en problemas de clasificación se conservan en cada partición para garantizar que cada una de ellas sea representativa de las proporciones de clase del conjunto de datos de entrenamiento.
- En scikit-learn: StratifiedKFold.

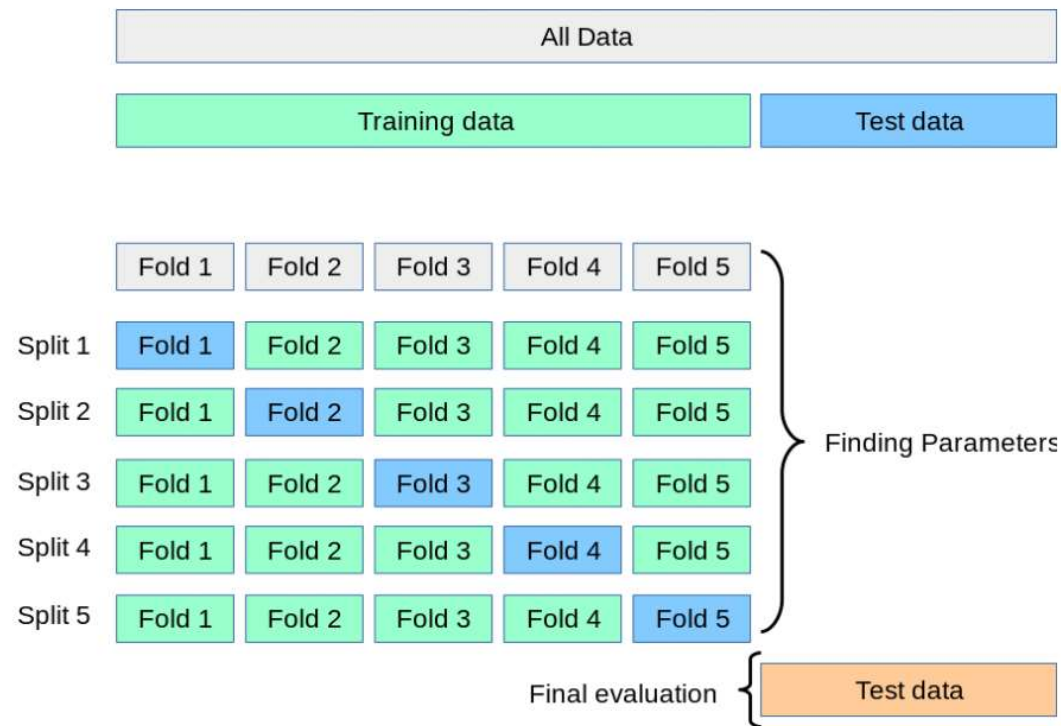
Protocolos de Evaluación: *Nested Cross-Validation*

- Útil cuando queremos seleccionar entre diferentes algoritmos de aprendizaje automático.
- En la validación cruzada anidada, tenemos un ciclo externo de validación cruzada de K_1 iteraciones para dividir los datos en particiones de entrenamiento y pruebas.
- También se usa un ciclo interno para seleccionar el modelo usando una validación cruzada de K_2 iteraciones en la partición de entrenamiento.
- Después de la selección del modelo, la partición de prueba se utiliza para evaluar el rendimiento del modelo.



Nota: este tipo particular de validación cruzada anidada también se conoce como validación cruzada 5x2.

Protocolos de Evaluación: *Combinando "train test split" con "cross-validation"*



Fuente: https://scikit-learn.org/stable/modules/cross_validation.html

Protocolos de Evaluación: *Bootstrapping*

▪ *Bootstrapping*:

- Se crean B conjuntos de datos a partir de los datos originales utilizando muestreo **con remplazo** (i.e.: cada observación puede aparecer más de una vez en cada uno de los nuevos conjuntos).
- Usualmente $B=\{100, 1000, 10000\}$.
- Cada conjunto tiene por lo general el mismo tamaño que el conjunto de datos original.



"pull yourself out, from what you have got".

Conjunto de datos original

$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(6)}$	$x^{(7)}$	$x^{(8)}$	$x^{(9)}$	$x^{(10)}$
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	------------

Bootstrap 1

$x^{(8)}$	$x^{(6)}$	$x^{(2)}$	$x^{(9)}$	$x^{(5)}$	$x^{(8)}$	$x^{(1)}$	$x^{(4)}$	$x^{(8)}$	$x^{(2)}$
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Bootstrap 2

$x^{(10)}$	$x^{(1)}$	$x^{(3)}$	$x^{(5)}$	$x^{(1)}$	$x^{(7)}$	$x^{(4)}$	$x^{(2)}$	$x^{(1)}$	$x^{(8)}$
------------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Bootstrap 3

$x^{(6)}$	$x^{(5)}$	$x^{(4)}$	$x^{(1)}$	$x^{(2)}$	$x^{(4)}$	$x^{(2)}$	$x^{(6)}$	$x^{(9)}$	$x^{(2)}$
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

.

.

.

Protocolos de Evaluación: *Bootstrapping*

- El *bootstrap* es una herramienta estadística flexible y poderosa que se puede utilizar para **cuantificar la incertidumbre** asociada con un estimador o método de aprendizaje automático dado.
- Por ejemplo, puede proporcionar **una estimación del error estándar de un coeficiente**, o **un intervalo de confianza para ese coeficiente**.
- ¿Puede el *bootstrap* estimar el error de predicción? No, **no es una buena forma de estimar el error de predicción**. Subestima el verdadero error de predicción de manera bastante dramática. Para este problema en particular, la validación cruzada es más fácil y funciona mejor.

Lecturas Complementarias

- ***Bias-Variance Tradeoff in Machine Learning***

<https://www.linkedin.com/pulse/bias-variance-tradeoff-machine-learning-satya-mallick/>

- ***Model evaluation, model selection, and algorithm selection in machine learning***

<https://sebastianraschka.com/blog/2016/model-evaluation-selection-part3.html>