Machine Learning III Clasificadores

Claudia Chávez O.

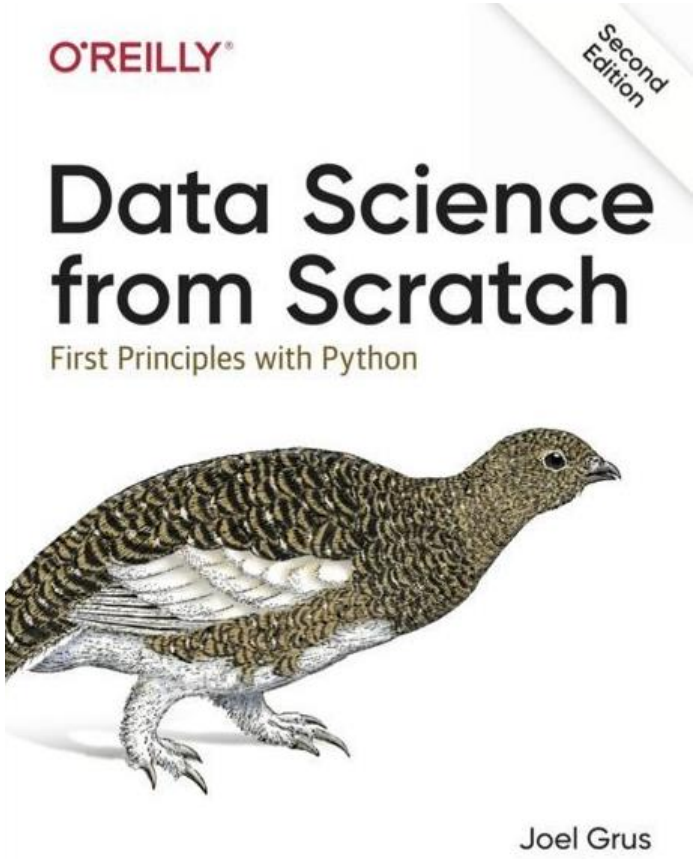
- Claudia Chávez O.
  - Ingeniero Estadístico USACH
  - Magíster Bioestadística Universidad de Chile
  - Diploma Big Data para políticas publicas Universidad Adolfo Ibañez
  - Diploma Inteligencia Artificial Universidad Adolfo Ibañez
  - Actualmente terminando Magíster en Inteligencia Artificial Universidad Adolfo Ibañez
  - Investigadora Programa Trabajo Empleo Equidad y Salud (TEES), Facultad Latinoamericana de Ciencias Sociales
  - Fundadora Grupo Medición y Evaluación (gMEv)

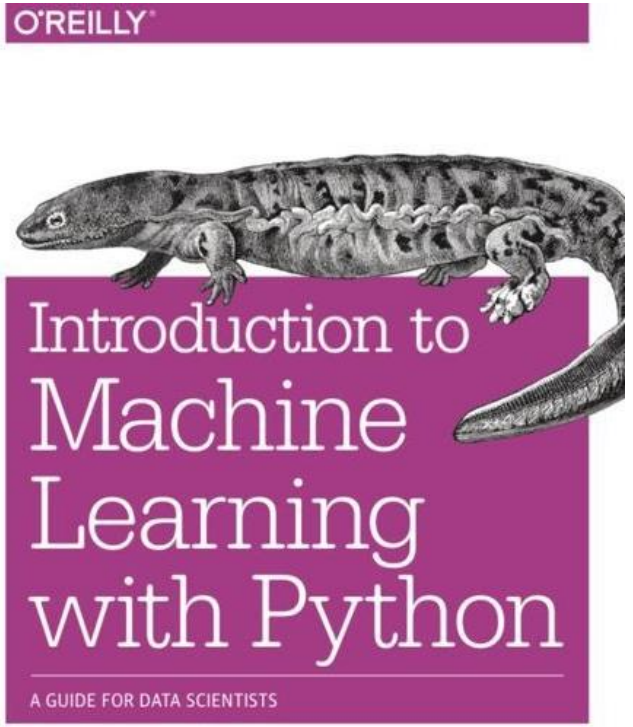
Presentación

- Para dudas claudia.chavez@usach.cl- cchavezo@gmail.com
- Tarea practica
- Fechas y contenidos:

30-Jun	Panorama general en modelos de clasificación y Machine Learning
05-Jul	Clasificación lineal: Logistic Regression
07-Jul	Modelos probabilísticos: Naïve Bayes
02-Ago	Árboles de clasificación
04-Ago	Support Vector Machines

Cronograma clases 19:00-20:20 (descanso 20 minutos) 20:40-22:00





Libros recomendados

Grus, J. (2019). *Data science from scratch: first principles with python*. O'Reilly Media.

Cutler, J., & Dickenson, M. (2020). Introduction to Machine Learning with Python. In *Computational Frameworks for Political and Social Research with Python* (pp. 129-142). Springer, Cham.

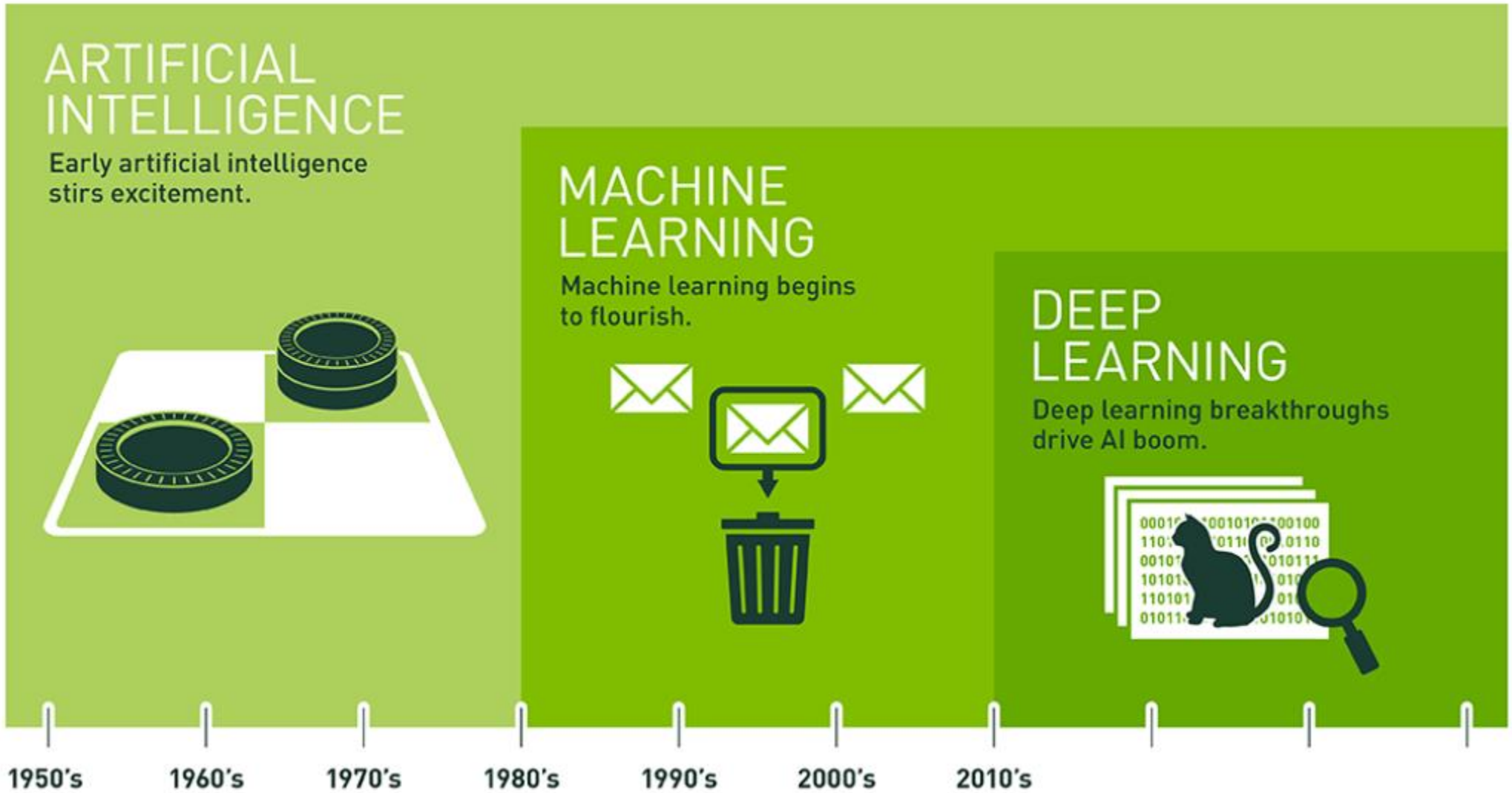
Fechas de entrega Tarea

- Tarea practica cuyo objetivo es que:
  - Realicen análisis exploratorio de datos
  - Apliquen tres algoritmos vistos en clases
  - Evaluar con indicadores de rendimiento

- Jueves 7 de julio les entregaré el problema a resolver

- Entrega: 9 Agosto a las 23:59 hrs.

Panorama general en modelos de clasificación y Machine Learning



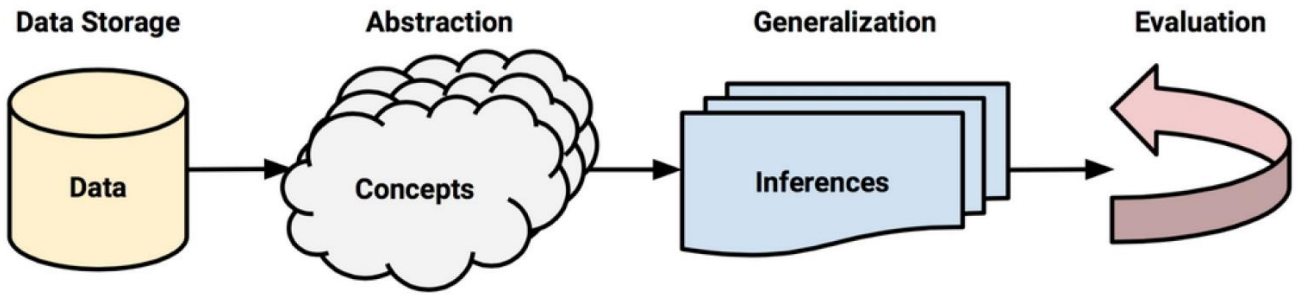
https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Machine Learning

- Inteligencia Artificial es la que simula el comportamiento y razonamiento de los humanos.
  - Ejemplo: Siri y Alexa
- Machine Learning es una **forma analítica de resolver problemas mediante la identificación, la clasificación o la predicción**. Los algoritmos aprenden de los datos y luego usan el conocimiento para sacar conclusiones de nuevos datos.
  - Ejemplo: Predicción preemergencia ambiental según datos monitoreo calidad del aire.
- Deep Learning es similar a Machine Learning, pero **usa algoritmos distintos**. ML trabaja mediante **algoritmos de regresión** o con **árboles de decisión**, Deep learning usa **redes neuronales** que funcionan de forma muy parecida a las conexiones neuronales biológicas de nuestro cerebro.
  - Ejemplo: Chatbots

Limitaciones de Machine Learning

- No es un sustituto para un cerebro humano
- No tiene flexibilidad
- No tiene sentido común
- Hay que identificar qué aprendió un algoritmo antes de aplicarlo
- Consideraciones éticas



Pasos del aprendizaje

- Cuatro pasos fundamentales
  - Almacenamiento de datos
  - Abstracción
  - Generalización
  - Evaluación

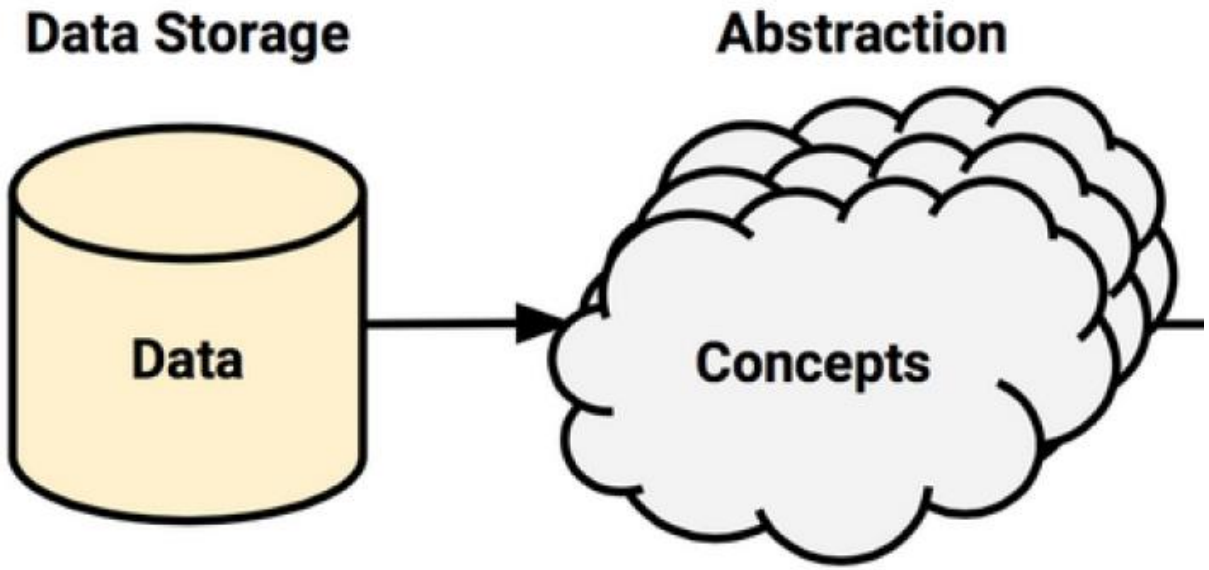
Almacenamiento datos-Abstracción

Data Storage

- ¿Qué guardar?
- ¿Cuánto guardar?

Abstraction

- Asignación sentido a los datos
- Elección de un modelo
- Entrenamiento de un modelo



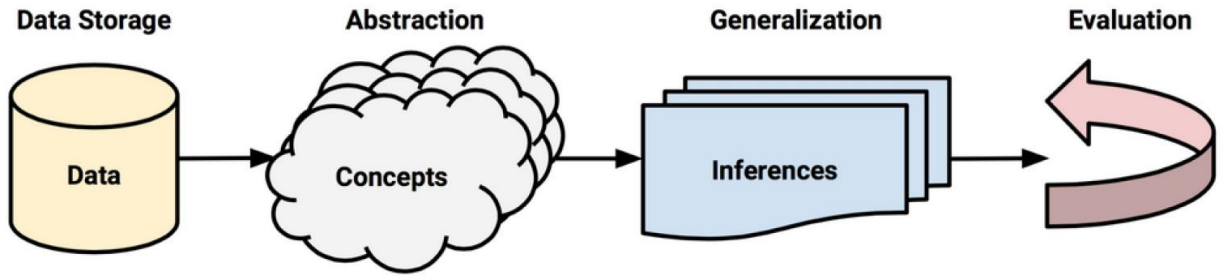
Generalización y evaluación

Generalización

- Convertir conocimiento para ser utilizado en el futuro
- Aplicar modelo para hacer acciones futuras
- Sesgo o Bias

Evaluación

- Nunca será perfecto
- Calidad del modelo
- Evitar sobreajuste (overfitting)



Tipos de aprendizaje

- **Aprendizaje supervisado: Aprendizaje basado en un conjunto de datos de entrenamiento para predecir/clasificar una variable objetivo.**
- Aprendizaje no supervisado: Aprendizaje donde un modelo se ajusta a las observaciones sin variable objetivo. Utilizado para agrupar, asociar o detectar anomalías.
- Aprendizaje reforzado: Aprendizaje donde un agente decide curso de acciones basado en maximización de recompensa

Algoritmos de clasificación supervisada

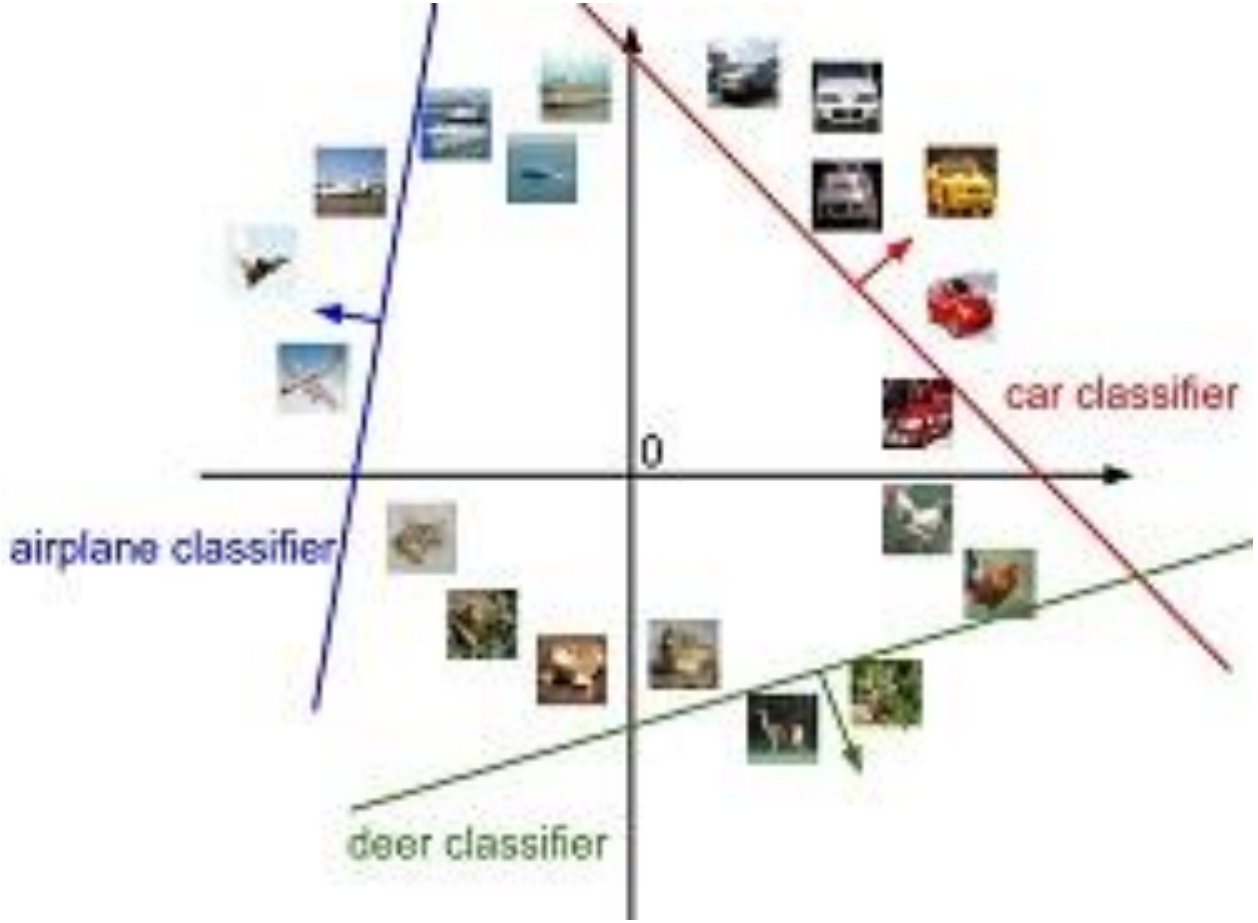
- Tarea más frecuente de los sistemas inteligentes
  - Fáciles de usar y de interpretar
- Fundamento estadístico y algebra lineal
- Los modelos están creados con técnicas estadísticas
- Cuando tengo una clase
- Cuando tengo multiclase

¿Qué es una tarea de clasificación?

- Manera clásica de entrenar modelos, donde se indica si un registro pertenece a una clase o a otra.
- Por otro lado, se pueden implementar estos modelos prediciendo las probabilidades de pertenencia a una clase.
  - Mayor flexibilidad ya que dichas probabilidades pueden ser interpretadas utilizando diferentes umbrales.
  - **Trade off** entre los errores de los modelos **(Equilibrio entre los falsos positivos vs los falsos negativos)**.

¿Qué es un clasificador?

- Algoritmo que recibe como entrada cierta información, define si un objeto es de cierta categoría o clase.
  - Número acotado de clases
- Puede ser un algoritmo que proporciona un valor numérico que indica la confianza (probabilidad) que tenemos de que la entrada se corresponda con una cierta clasificación.



Fases clasificador

- Field Cady nos dice que, en realidad, un clasificador, uno basado en Machine Learning, funciona en dos fases:

En una primera fase es entrenado, es decir, recibe una gran cantidad de datos de ejemplo y su clasificación correcta, de forma que se pueden ajustar los parámetros del algoritmo para un funcionamiento óptimo.

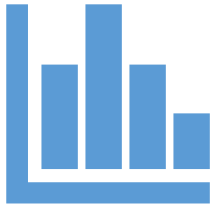
Cuando ya está entrenado es cuando ya funciona como describíamos más arriba: recibe unos datos de entrada y da la clasificación correspondiente como salida.

Cady, F. (2017). *The data science handbook*. John Wiley & Sons.

Ejemplos de clasificadores

- Clasificar si el correo que llega es Spam o No es Spam
- Dados unos resultados clínicos de un tumor clasificar en “Benigno” o “Maligno”.
- El texto de un artículo a analizar es: Entretenimiento, Deportes, Política ó Ciencia
- A partir de historial bancario conceder un crédito o no

Distintos modelos de Machine Learning clasificación



Desde la estadística

Regresión logística

Regresión Probit

KNN (K vecinos próximos)

Naive Bayes

Desde la IA

Redes Neuronales

Árboles de Decisión

Suport Vector Machine

Modelos de clasificación que veremos

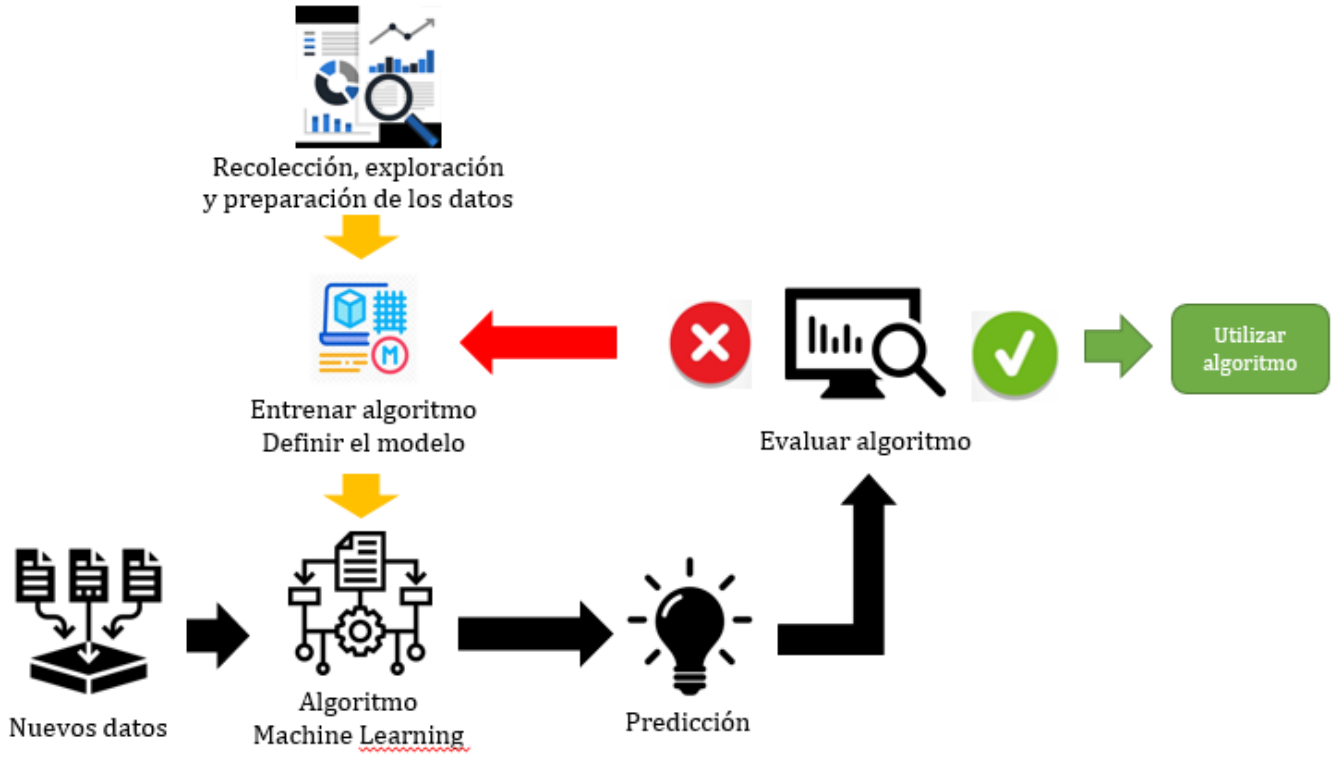
- **Árbol de decisión (‘*Decision Tree*’)**: simplemente una estructura donde en cada nodo se hace una pregunta y, en función de la respuesta, se sigue hacia un nodo u otro. La respuesta final es la categoría que andamos buscando.
- **Máquinas de Vectores de Soporte (‘*Support Vector Machines*’)**: Se trata de un tipo de clasificadores que asumen que se puede establecer fronteras lineales en el espacio total de valores posibles de forma que, dependiendo de dónde se encuentre un conjunto de datos concreto, estará a alguno de los lados de la frontera, lo que lleva a su clasificación según el lado en que se encuentre. Asumen una separabilidad lineal (las fronteras vienen dadas por ecuaciones lineales).

- **Regresión Logística (‘*Logistic regression*’)**: Es como una versión no binaria de las Support Vector Machines. También suponen algún tipo de frontera pero lo que hacen es asignar probabilidades de una clasificación u otra según lo cerca o lejos que se encuentren de la frontera.
- **Bayes ingénuo (‘*Naive Bayes*’)**: una aplicación simplificada de la estadística bayesiana en que, se parte de unos valores iniciales de confianza en una clasificación y, durante el entrenamiento se van ajustando

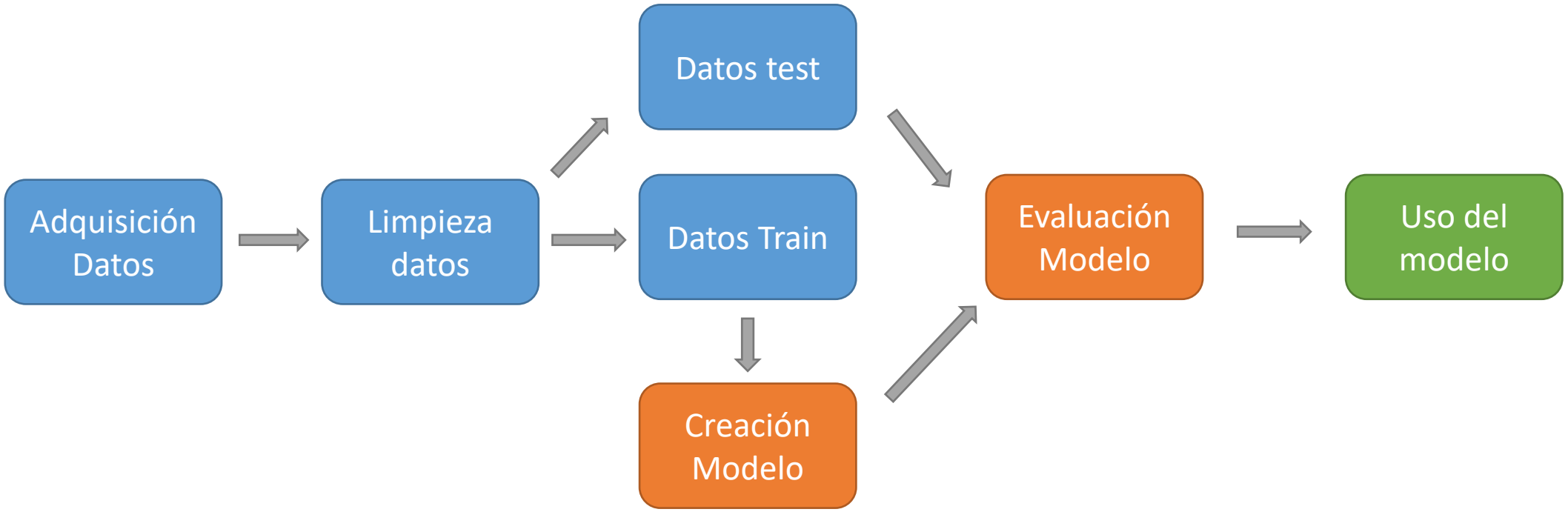
Cady, F. (2017). *The data science handbook*. John Wiley & Sons.

Pasos en ML

- Elaboración propia en base a COGITO, 2018; Lantz, 2019

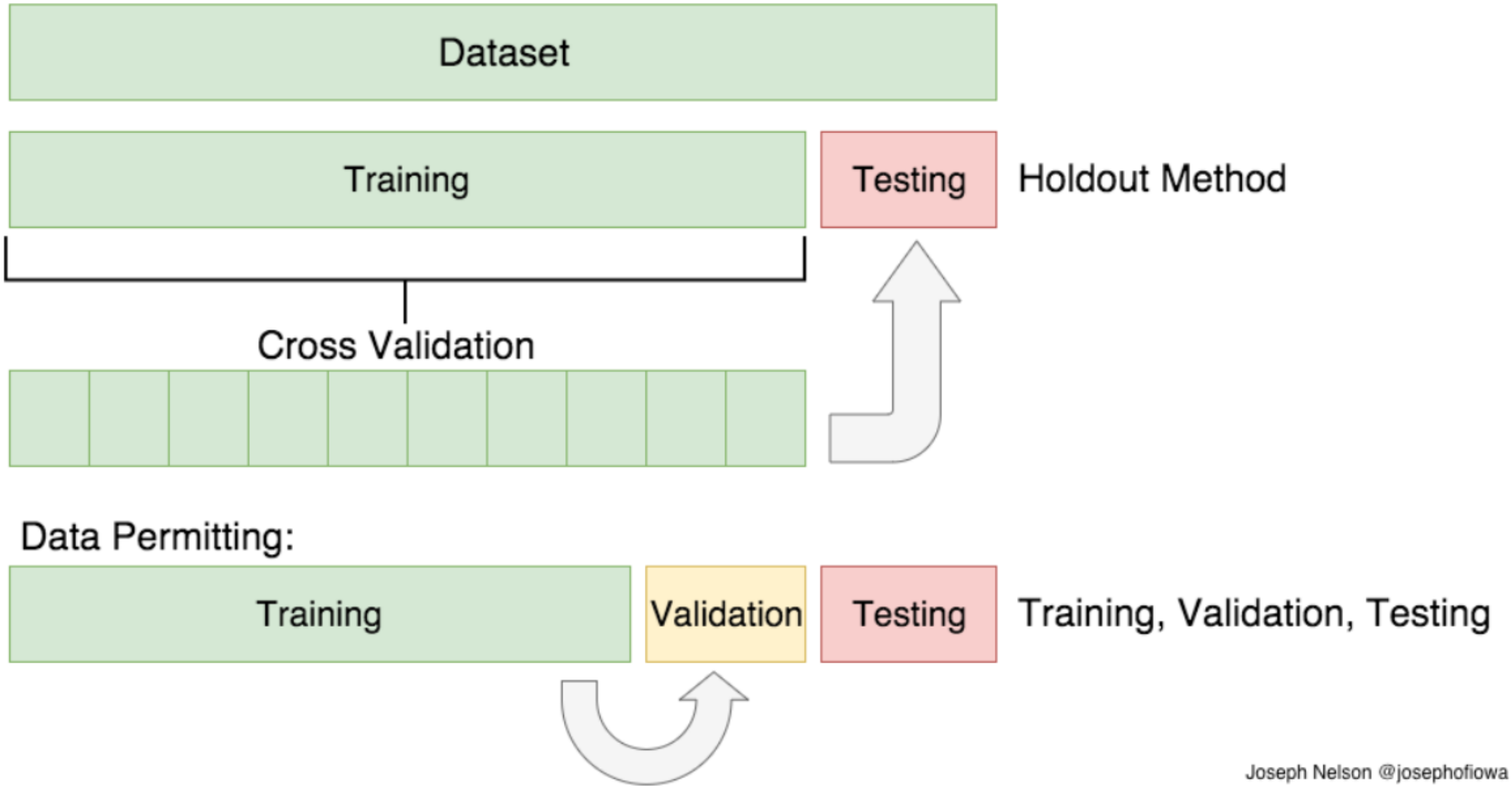


Pasos en ML


Adquisición Datos
Limpieza datos
Datos test
Datos Train
Creación Modelo
Evaluación Modelo
Uso del modelo

Splitting Data

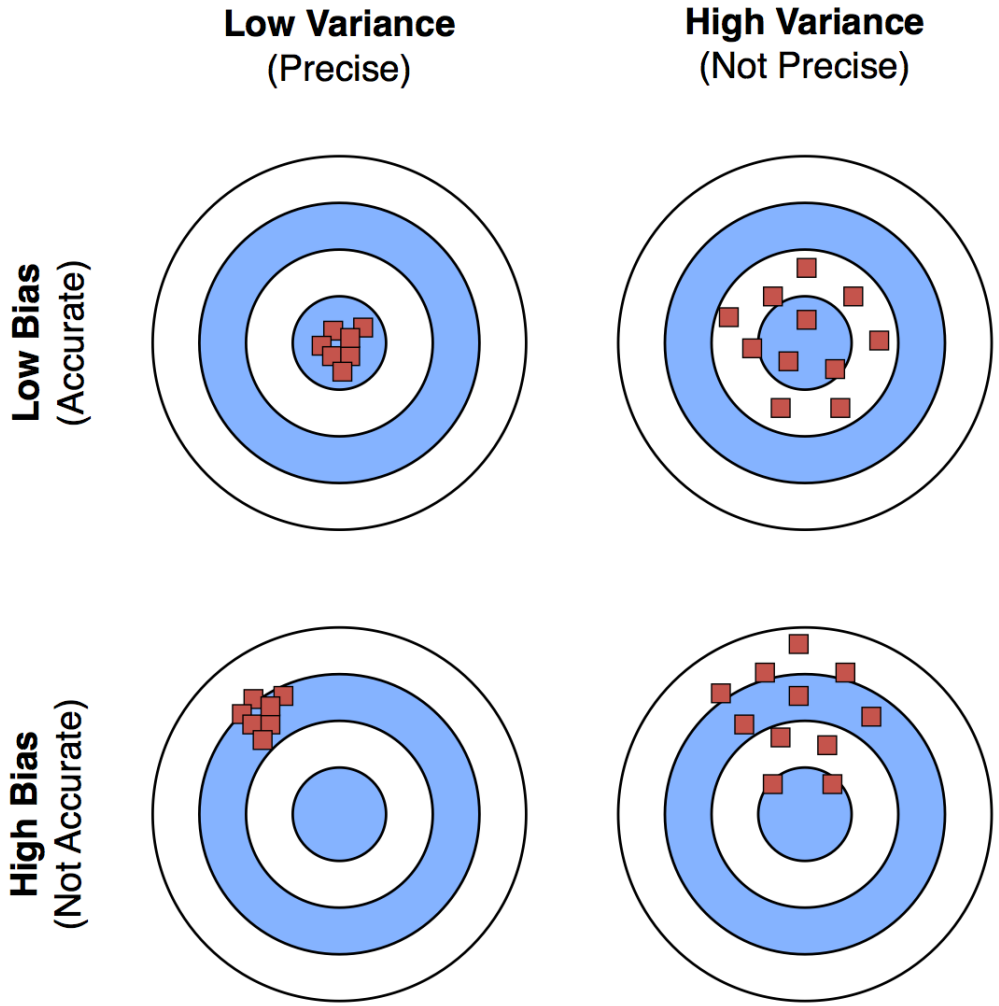
- Paso previo a aplicar un método de clasificación
- Datos de entrenamiento y test
  - Entrenamiento es utilizado para estimar los parámetros del modelo
  - Test se emplea para comprobar el comportamiento del modelo estimado.
- Procedimientos de generación: **muestreo aleatorio simple o muestreo estratificado.**
- Lo ideal es entrenar el modelo con un conjunto de datos independiente de los datos con los que realizamos el test.
- 70% y 30%, 20% y 80%, 25% y 75%, etc.



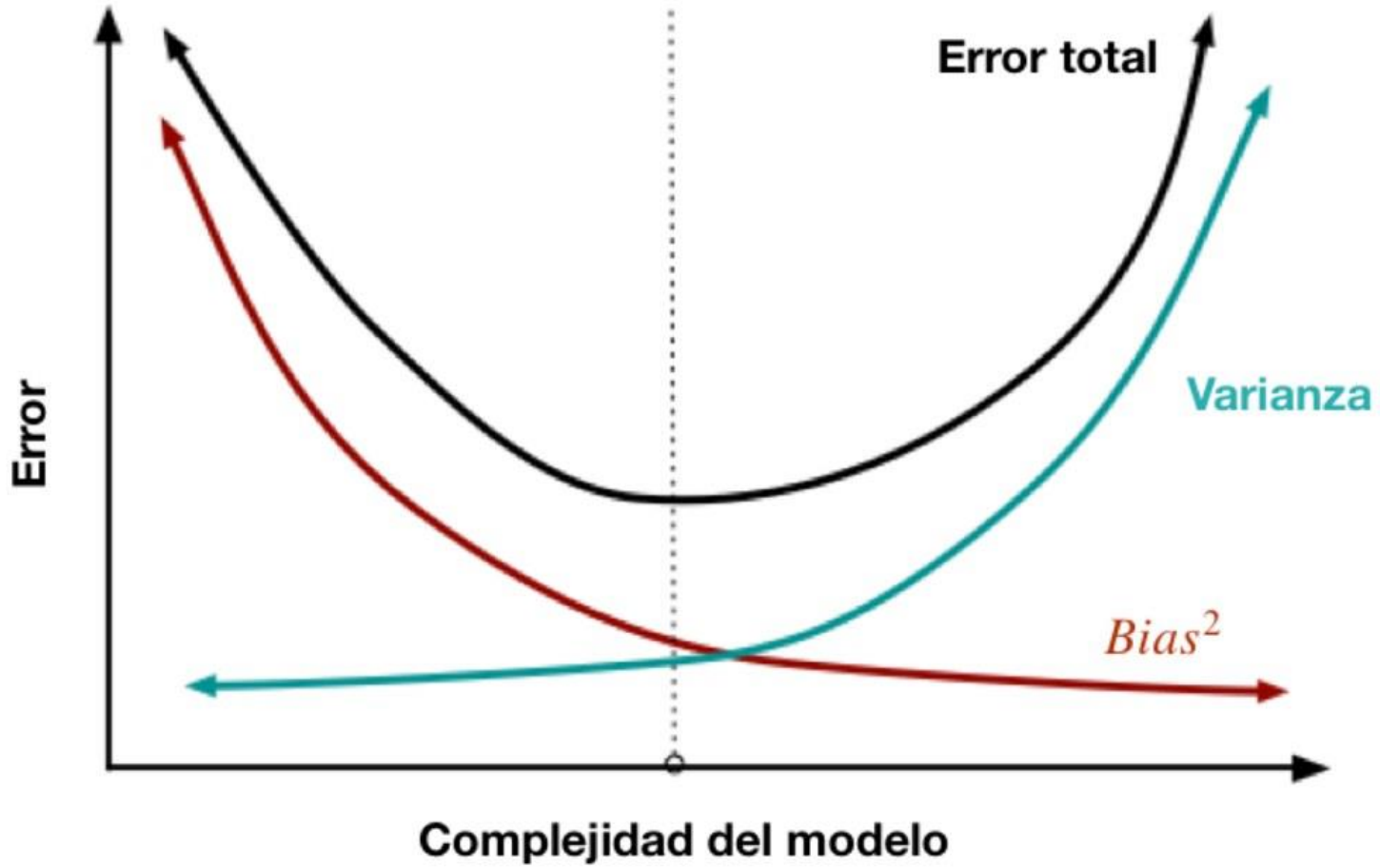
Joseph Nelson @josephofiowa

Errores en la clasificación

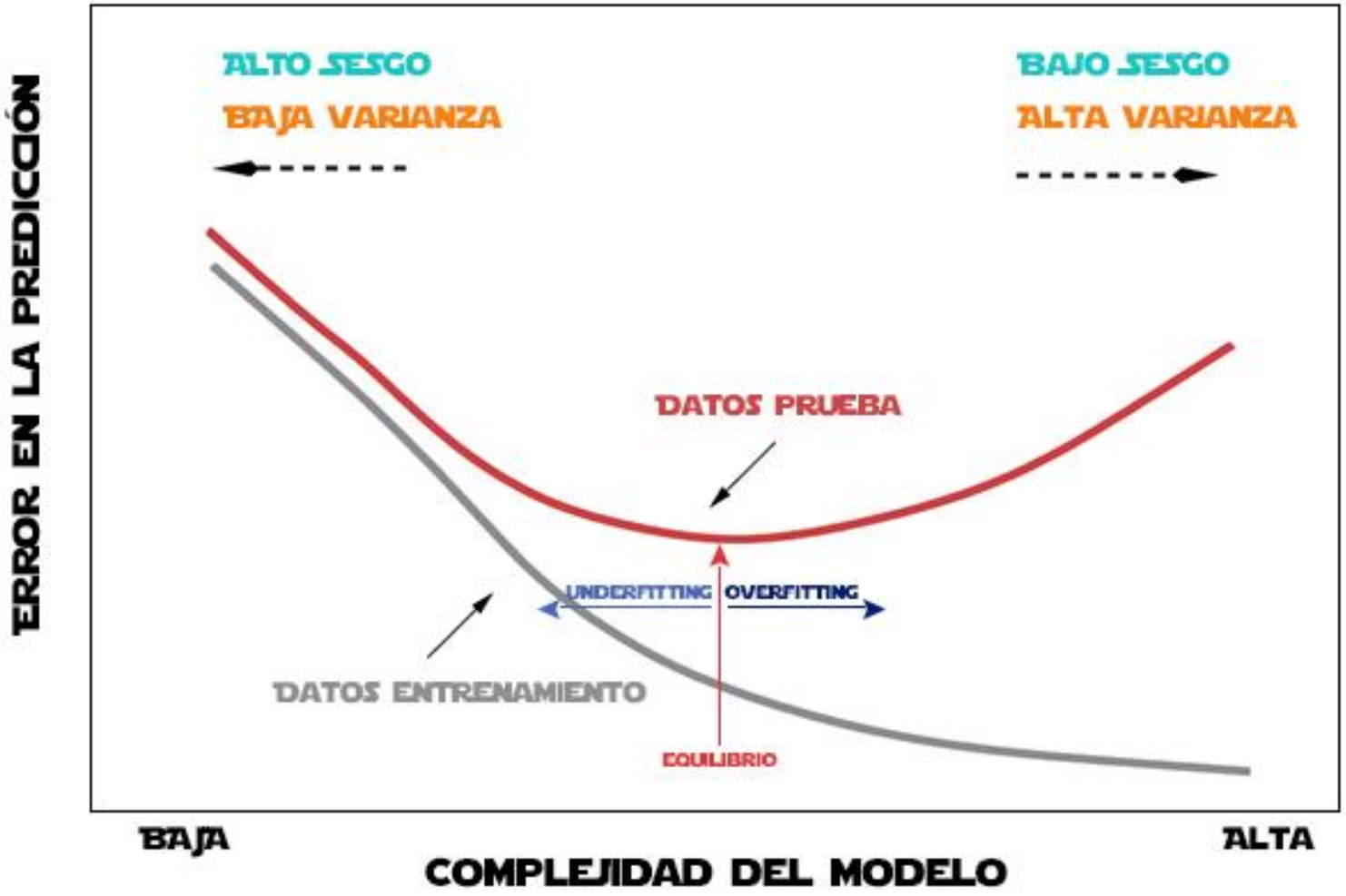
- Buen o mal ajuste del modelo
- Accuracy=% de predicciones correctas
- Tasa de error, inverso a accuracy
- No siempre funciona bien



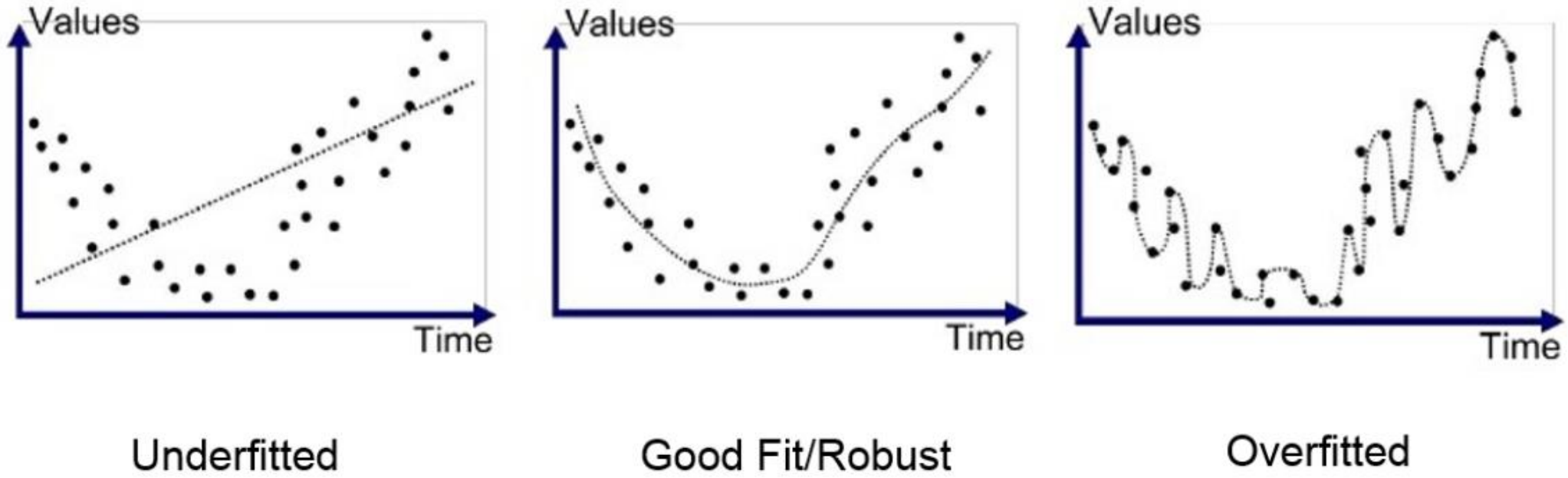
Sesgo y varianza



Sesgo y varianza

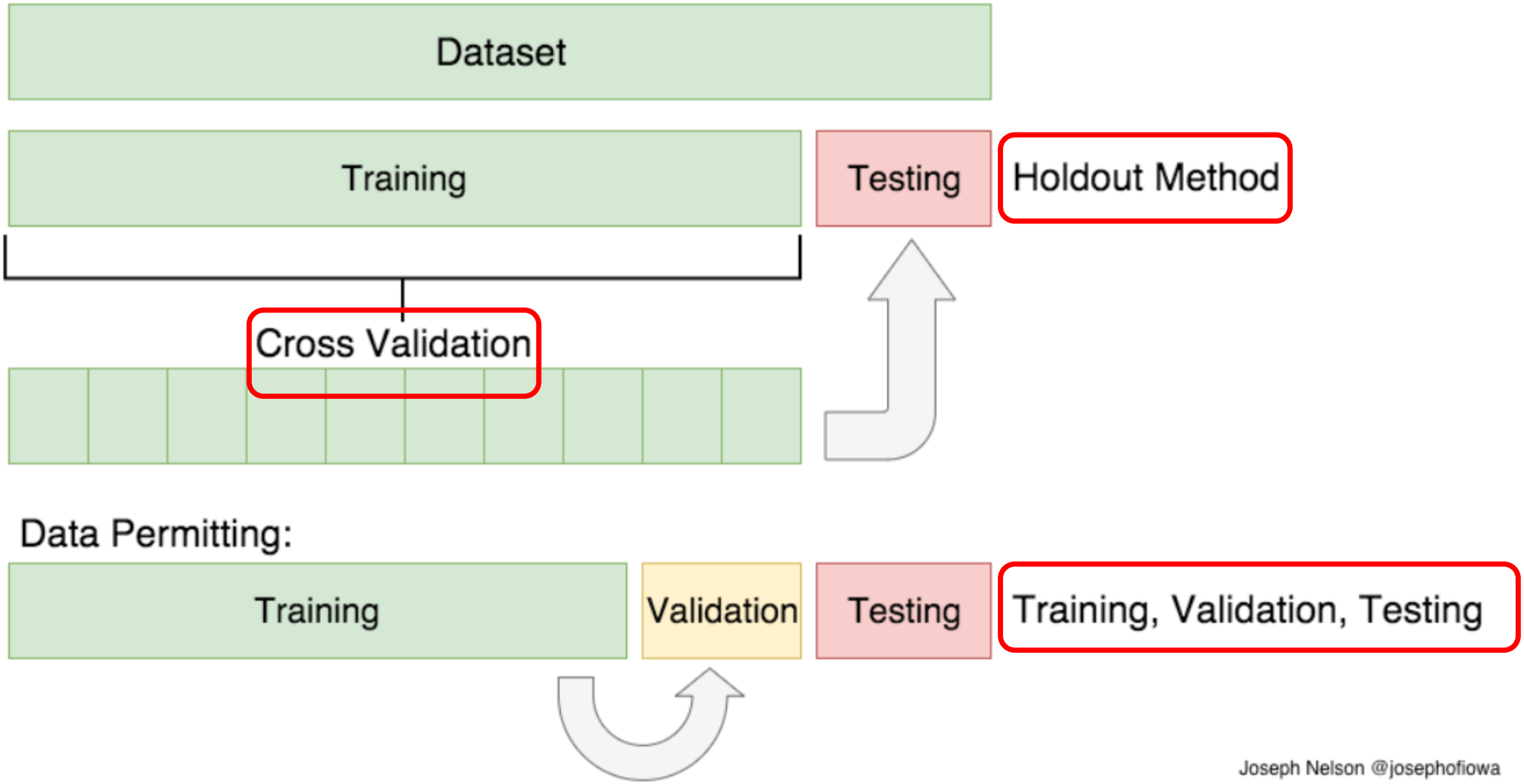


Sobre ajuste u overffitting



Cuando el algoritmo de aprendizaje se ajusta tanto a los datos de entrada que pierde su capacidad de generalizar

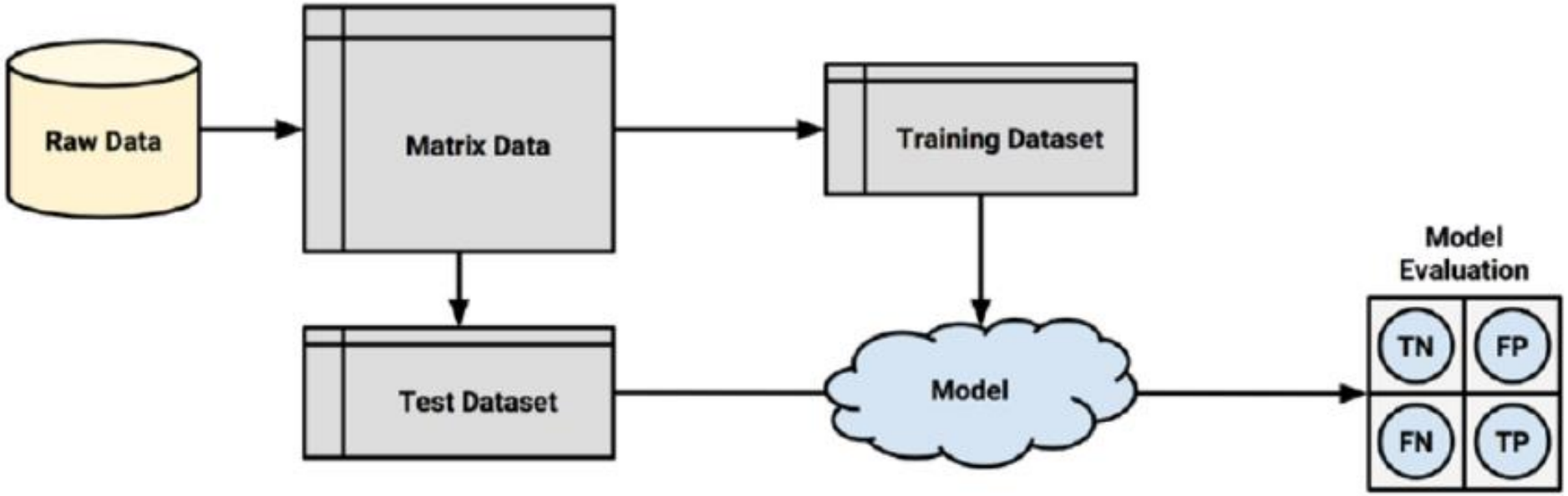
https://bookdown.org/content/2031/arboles-de-decision-parte-i.html


Dataset
Training
Testing
Holdout Method
Cross Validation
Data Permitting:
Training
Validation
Testing
Training, Validation, Testing
Joseph Nelson @josephofiowa

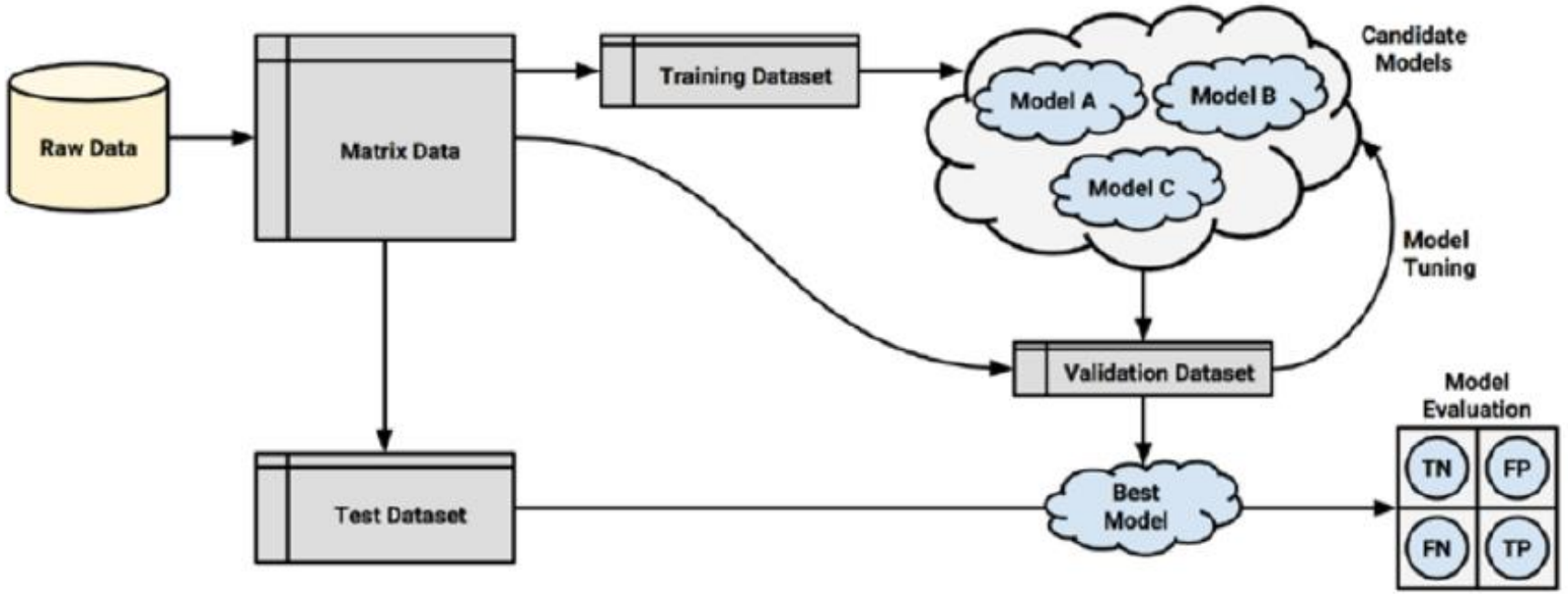
```
from sklearn import datasets
from sklearn.model_selection import train_test_split
#
# Load the Boston Dataset
#
bhp = datasets.load_boston()
#
# Create Training and Test Split
#
X_train, X_test, y_train, y_test = train_test_split(bhp.data, bhp.target, random_state=42,
test_size=0.3)
```

Método Houldout

Holdout simple

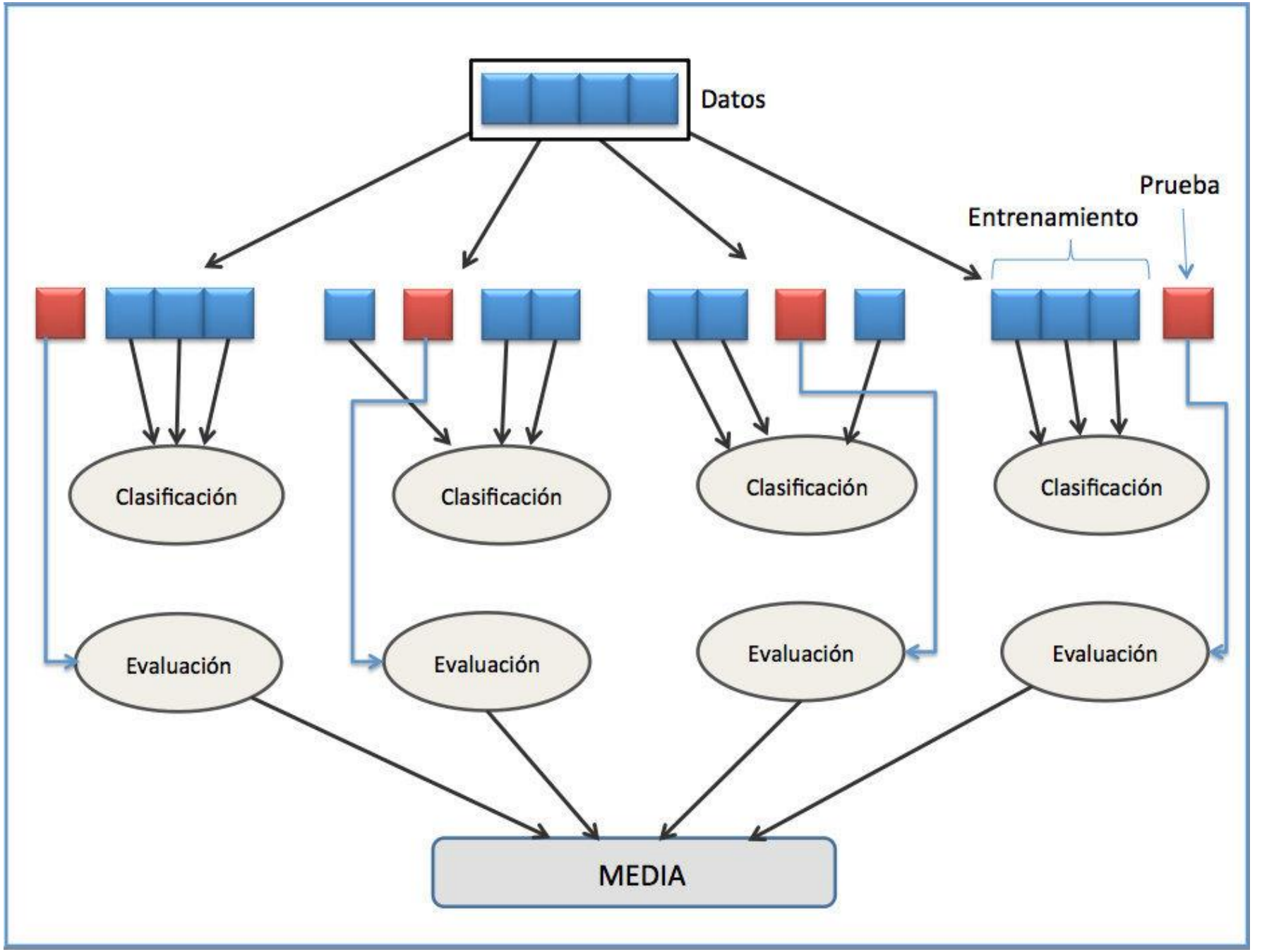


Houldout con validación



Validación cruzada o Cross-validation

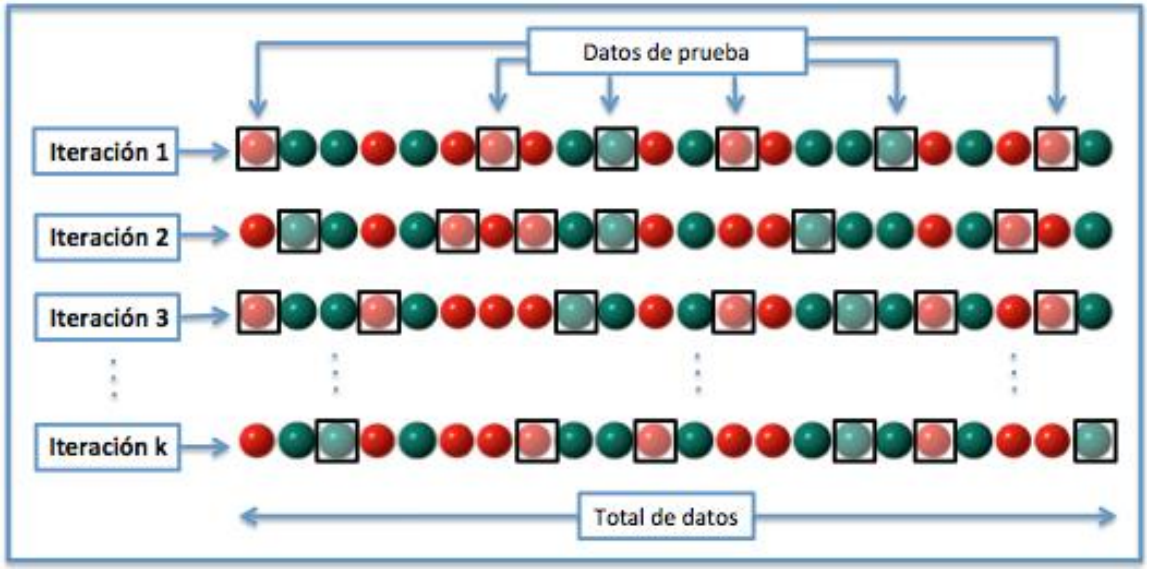
- Houldout con repetición (k-fold cross validation)
- La validación cruzada o cross-validation es una técnica utilizada para evaluar los resultados de un análisis estadístico cuando el conjunto de datos se ha segmentado en una muestra de entrenamiento y otra de prueba, la validación cruzada comprueba si los resultados del análisis son independientes de la partición.
- Aunque la validación cruzada es una técnica diseñada para modelos de regresión y predicción, su uso se ha extendido a muchos otros ejercicios de machine learning.



https://commons.wikimedia.org/w/index.php?curid=17617674

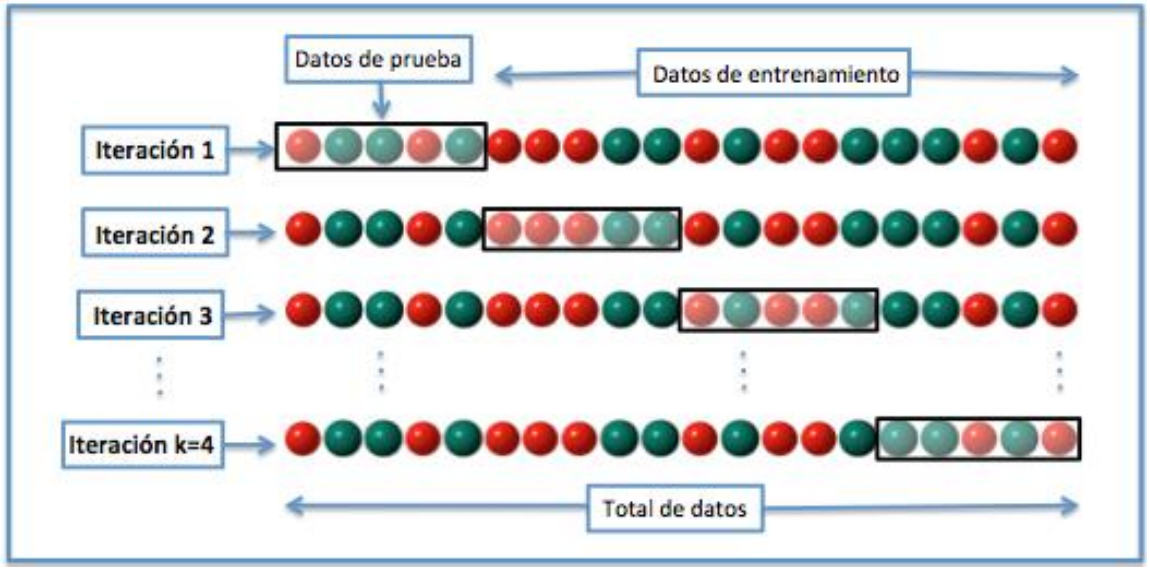
Validación cruzada aleatoria

- Una aplicación alternativa consiste en repetir el proceso anterior, seleccionando aleatoriamente distintos conjuntos de datos de entrenamiento, y calcular los estadísticos de validación a partir de la media de los valores en cada una de las repeticiones.
- Los inconvenientes es que hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar.



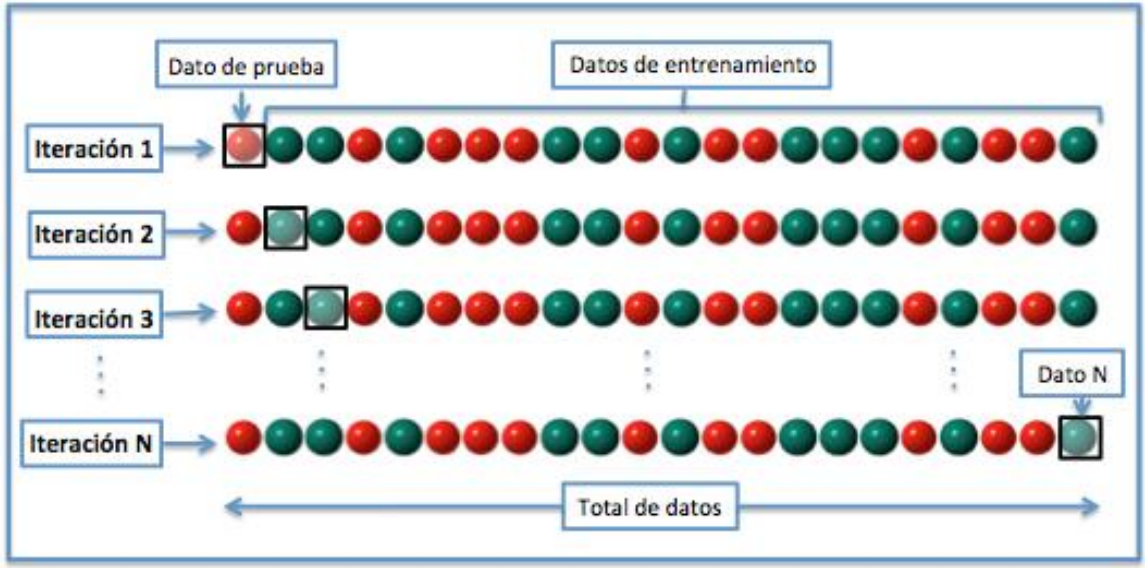
Validación cruzada de k iteraciones

- Mayor utilidad cuando el conjunto de datos es pequeño es la *Validación cruzada de K iteraciones o K-fold cross-validation*.
- El total de los datos se dividen en k subconjuntos, de manera que aplicamos el método hold-out k veces, utilizando cada vez un subconjunto distinto para validar el modelo entrenado con los otros k-1 subconjuntos.
- El error medio obtenido de los k análisis realizados nos proporciona el error cometido por el método. En la práctica, la elección del número de iteraciones depende de la medida del conjunto de datos.
- Lo más común es utilizar la validación cruzada de 10 iteraciones (10-fold cross-validation).



Validacion cruzada leave one out (*LOOCV)*

- Se separan los datos de forma que en cada iteración tengamos una solo dato para el test, constituyendo el resto la muestra de entrenamiento, el error obtenido sería el promedio de los errores cometidos en cada iteración.
- En este método la estimación del error es más estable, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como sea el tamaño de nuestro conjunto de datos ya que cada dato se evalúa en la fase de test.



¿Cuánto me equivoco?
… en clasificación binaria (una clase)

- Ejemplo: clasificación binaria, esto es **0 y 1.** Detección de infección por COVID-19
  - Persona que tiene covid19 y el modelo lo clasificó como covid19 (+) . Esto sería un **verdadero positivo** o VP .
  - Persona que no tiene covid19 y el modelo lo clasifico como covid19 (-) . Este seria un **verdadero negativo** o sea un VN.
  - Persona que tiene covid19 y el modelo lo clasificó como covid19 (-) . Éste seria un error tipo II o **falso negativo** o FN.
  - Persona que no tiene covid19 y el modelo lo clasificó como covid19 (+) . Este es un error tipo I, o **falso positivo** o FP.

Ejemplo COVID-19

- Ejemplo: Modelo de detección de infección por COVID-19, deberíamos estar más preocupados de tener pocos falsos negativos que de tener pocos falsos positivos.
  - Un falso negativo significaría que el modelo ha identificado que no hay infección cuando sé que la habrá con sus efectos derivados.
  - Un falso positivo solo provocaría que se mandara a cuarentena a una persona sin infección.
- **Entender el impacto de los falsos negativos y de los falsos positivos es clave para todo modelo de clasificación**.

		Actual	
		Positive (1)	**Negative (0)**
Predict	**Positive (1)**	TP (True Positive)	FP (False Positive)
	Negative (0)	FN (False Negative)	TN (True Negative)

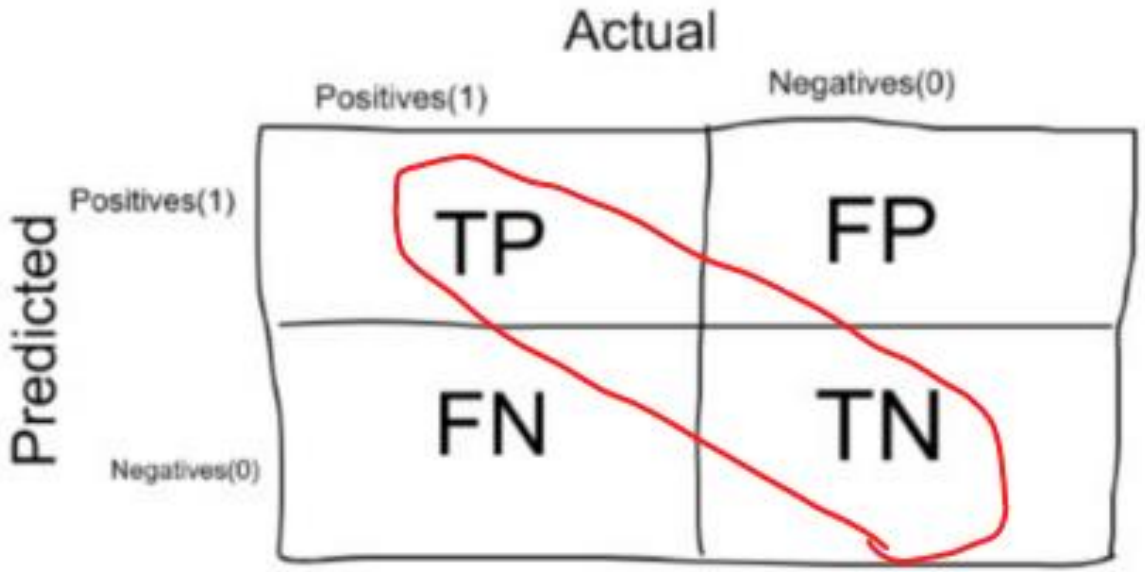
Matriz de confusión

- **Falso Positivo**. Predecir un evento cuando no hubo evento
- **Falso Negativo**. Predecir que no hubo un evento cuando sí que hubo evento

- Al calibrar el umbral de nuestro modelo de predicción, el balanceo entre estos dos errores es lo que nos otorgará el nivel de umbral óptimo.

Cáncer: 100 datos, 5 personas tienen cáncer

Predicción:
nadie tiene
cáncer

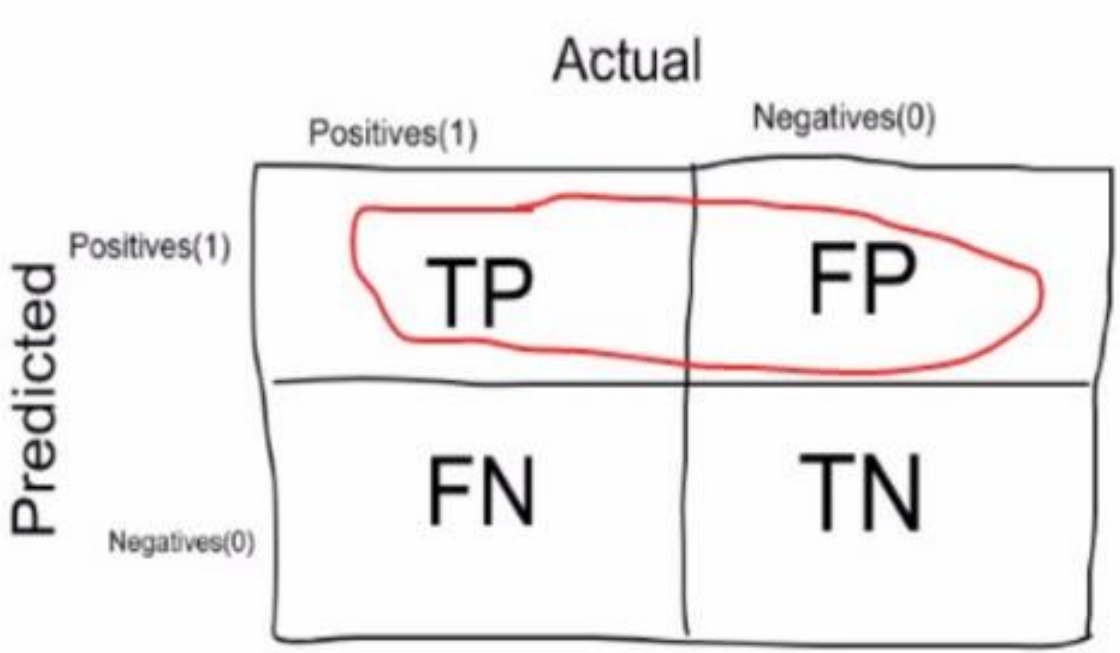


Acurracy=(0+95)/100 = 95%

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Cáncer: 100 datos, 5 personas tienen cáncer

Predicción: 100 personas con cáncer

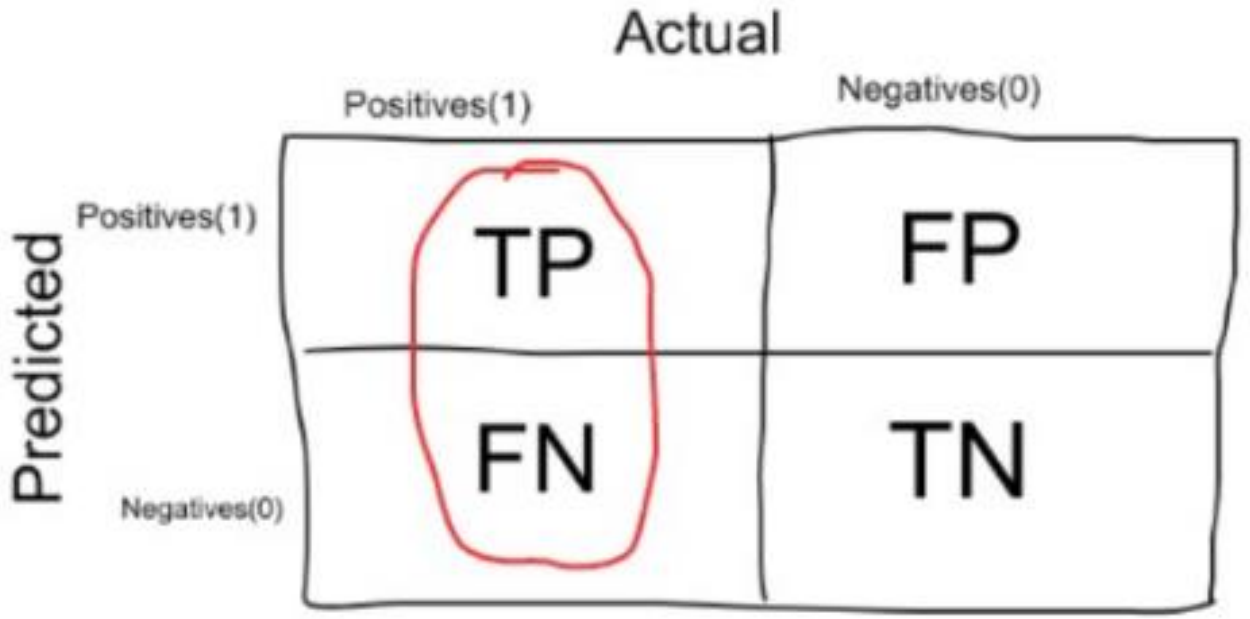


Precision=5/100=5%

$$\text{Precision} = \frac{TP}{TP + FP}$$

Cáncer: 100 datos, 5 personas tienen cáncer

Predicción: 100 personas con cancer



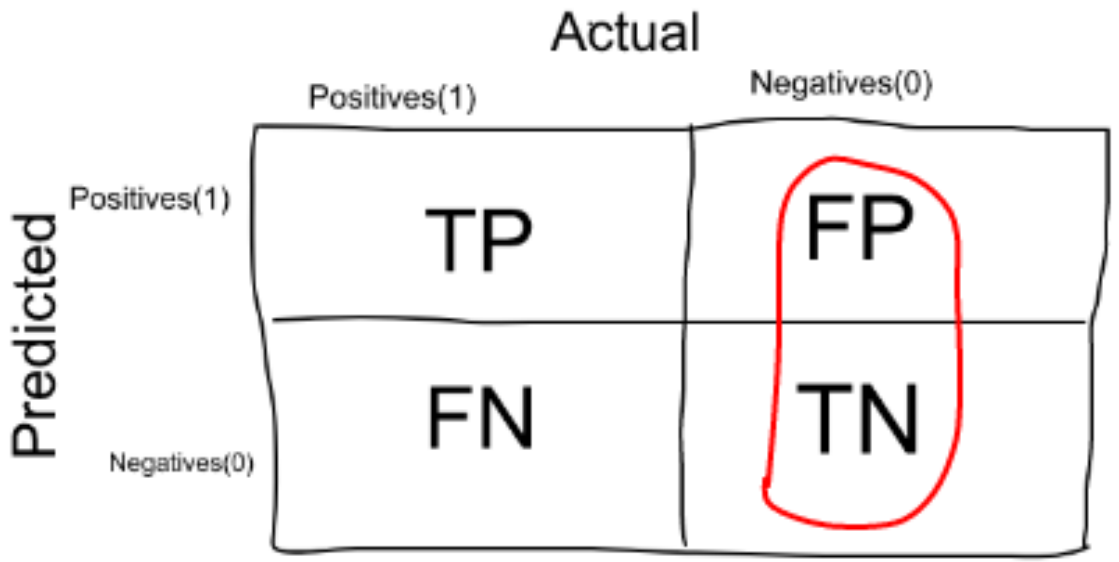
$$\text{Recall} = \frac{TP}{TP + FN}$$

Sensibilidad=Recall =5/5=100%

Sensibilidad=Recall

Cáncer: 100 datos, 5 personas tienen cáncer

Predicción: 100 personas con cáncer



$$\text{Specificity} = \frac{TN}{TN + FP}$$

Especificidad= 0/95=0%

En resumen…

Más representativo en clases balanceadas y no balanceadas

- **Precisión:** calidad de la predicción. ¿Qué porcentaje de los que hemos dicho que de la clase positiva lo son?
- **Recall:** cantidad. ¿Qué porcentaje de la clase positiva se ha estimado bien?
- **F1:** Combina precisión y recall en un solo indicador.
- **Matriz de confusión:** indica errores cometidos en la estimación.

No funciona muy bien clases desbalanceadas

- **Accuracy:** Porcentaje que el modelo a estimado correctamente.

Alternativa: Curva ROC (Receiver Operating Characteristic)

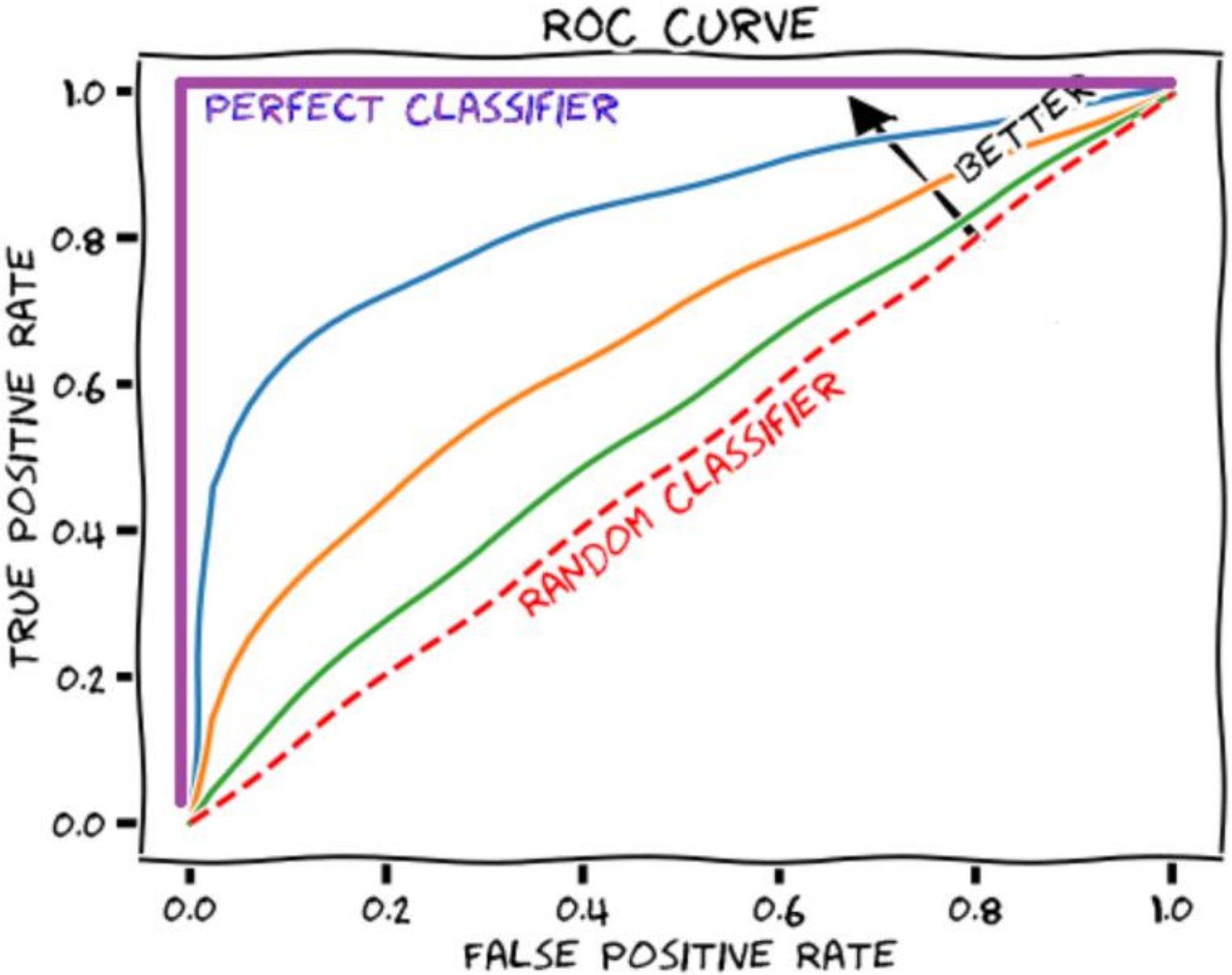
- El uso de estas curvas permitirá interpretar correctamente las predicciones de probabilidad de modelos de clasificación binaria.
- Muestra la distribución de las fracciones de verdaderos positivos y de falsos positivos
- La fracción de verdaderos positivos se conoce como sensibilidad, sería la probabilidad de clasificar correctamente a un individuo cuyo estado real sea definido como positivo.
- La especificidad es la probabilidad de clasificar correctamente a un individuo cuyo estado real sea clasificado como negativo. Esto es igual a restar uno de la fracción de falsos positivos.

Curva ROC

- La curva ROC es conocida como la representación de sensibilidad frente a (1-especificidad).
- Cada resultado de predicción representa un punto en el espacio ROC.
- El mejor método posible de predicción se situaría en un punto en la esquina superior izquierda, o coordenada (0,1) del espacio ROC, representando un 100% de sensibilidad (ningún falso negativo) y un 100% también de especificidad (ningún falso positivo).
- Una clasificación totalmente aleatoria daría un punto a lo largo de la línea diagonal, que se llama también línea de no-discriminación. En definitiva, se considera un modelo inútil, cuando la curva ROC recorre la diagonal positiva del gráfico.
- En tanto que en un test perfecto, la curva ROC recorre los bordes izquierdo y superior del gráfico. La curva ROC permite comparar modelos a través del área bajo su curva.

Área bajo la curva

Sobresaliente: 0.9 - 1.0
Excelente: 0.8 - 0.9
Aceptable: 0.7 - 0.8
Pobre: 0.6 - 0.7
No clasifica: 0.5 - 0.6



https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

Otra aproximación

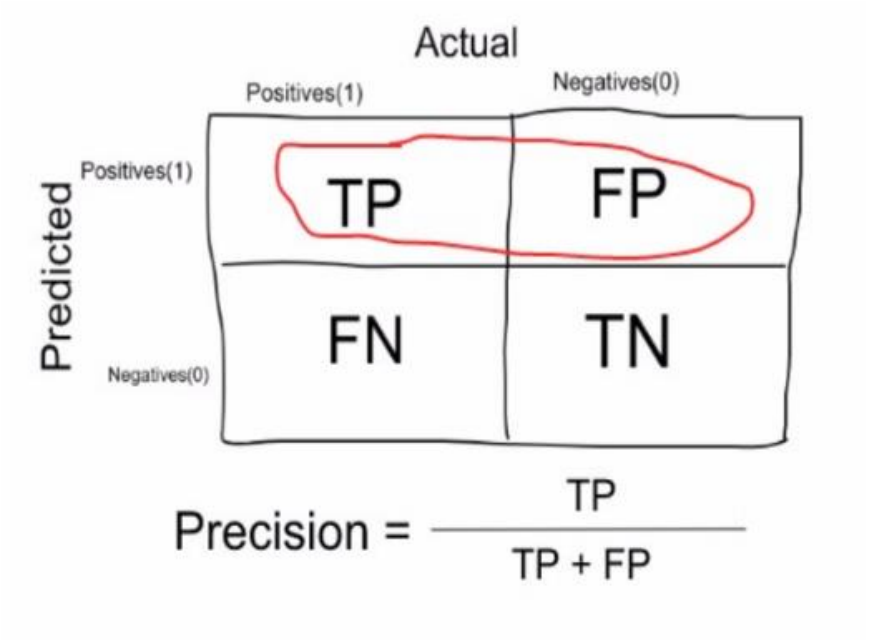
- Una aproximación distinta es predecir las probabilidades de cada clase. La clase final se determinará utilizando esta probabilidad y un umbral de decisión.
- Es decir, podríamos obtener la clase utilizando un umbral de 0.5
  - Toda predicción con una probabilidad inferior al umbral (<0.5) se considerará que pertenece a la clase 0
  - Toda predicción con un valor igual o superior al umbral (0.5) se considerará que pertenece a la clase 1.
- Este umbral puede ser ajustado para modificar el comportamiento de nuestro modelo para un problema específico.
  - La modificación del umbral serviría si se quisiera inclinar el modelo hacia una de las dos clase.
  - Esta orientación del modelo hacia una clase serviría para tratar de minimizar los fallos de predicción de una determinada clase.

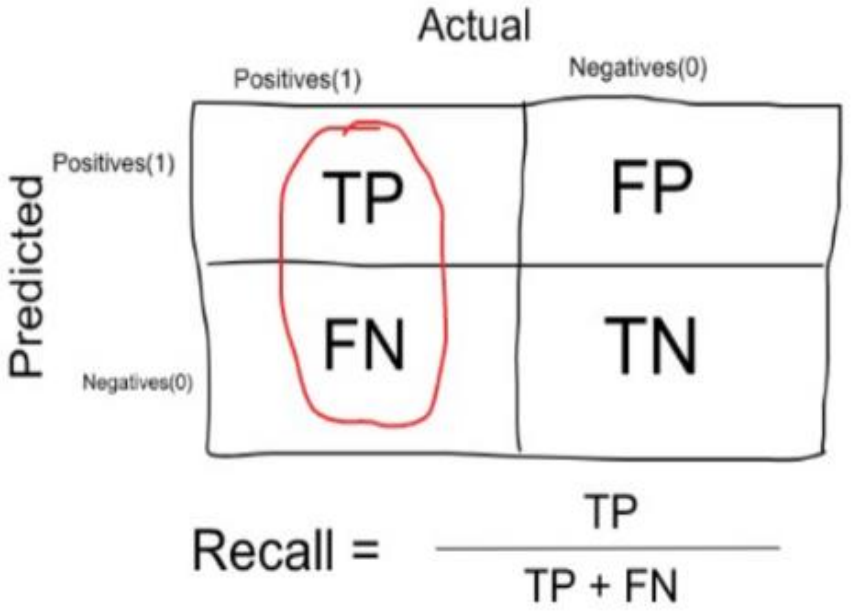
Utilidad de la curva ROC

- Permite comparar diferentes modelos para identificar cual otorga mejor rendimiento como clasificador.
- El **área debajo de la curva (AUC)** puede ser utilizado como resumen de la calidad del modelo.
- En resumen:
  - Valores pequeños en el eje X indican pocos falsos positivos y muchos verdaderos negativos
  - Valores grandes en el eje Y indican elevados verdaderos positivos y pocos falsos negativos
  - El valor **AUC se utiliza como resumen del rendimiento del modelo**. **Cuanto más esté hacia la izquierda la curva**, más área habrá contenida bajo ella y por ende, mejor será el clasificador. El clasificador aleatorio tendría una AUC de 0.5 mientras que el clasificador perfecto tendría un AUC de 1.

Curvas de precisión-sensibilidad (Precision-Recall)

- Centrada en precisión y sensibilidad
- La **precisión** se calcula como el **número de verdaderos positivos entre la suma de verdaderos positivos y de falsos positivos**. Describe cómo de bueno es el modelo a la hora de predecir las salidas de la clase positiva. Otra forma de llamar a la precisión también se le llama poder predictivo positivo.
- Vuelve a aparecer la **sensibilidad (*recall)***, que ya ha sido comentada en apartados anteriores. **Verdaderos positivos divididos entre la suma de verdaderos positivos y de falsos positivos**.
- Como resumen, la curva de precisión-sensibilidad enfrenta la precisión (eje y) con la sensibilidad (eje x) para diferentes umbrales.
- En la imagen posterior se muestra dicha curva para una serie de modelos, cada uno con un poder predictivo distinto. El modelo (morado) en la esquina representaría a un clasificador perfecto. Las curvas más alejadas de esa esquina representarían a modelos peores. Un modelo aleatorio sin entrenar estaría representado como una línea horizontal a media altura (0.5).

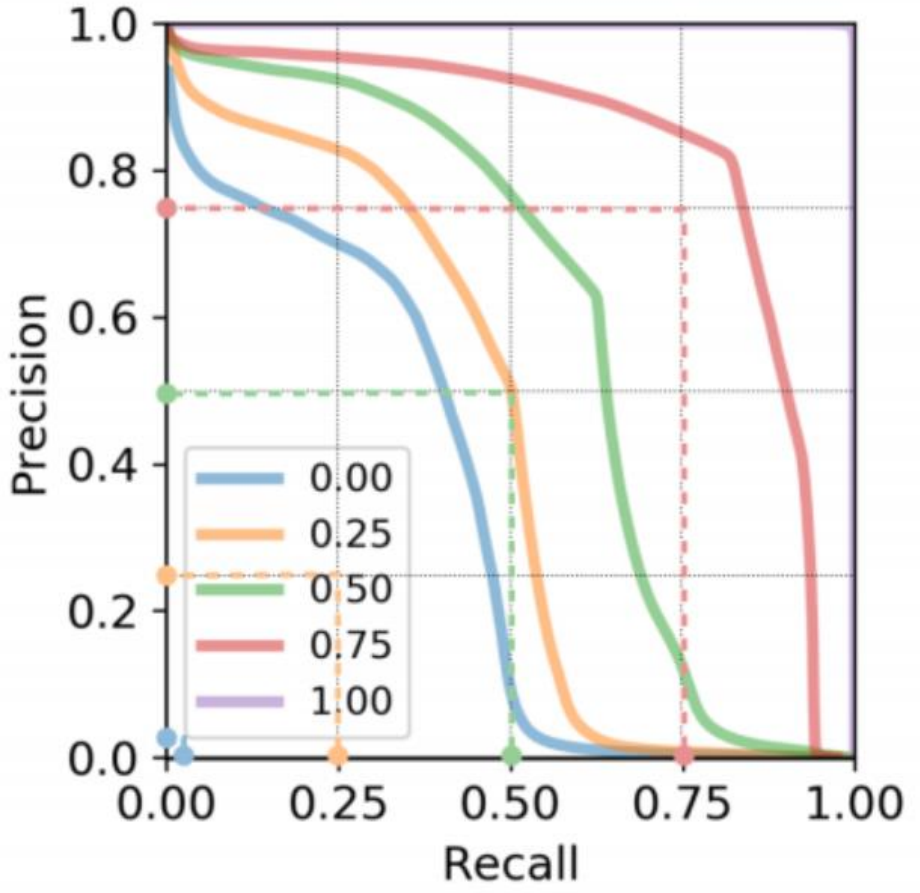




Sensibilidad=Recall

Precision-recall

- https://arxiv.org/pdf/1905.05441.pdf



Utilidad Precisión-Sensibilidad

- Útil en aquellos **casos en los que tenemos clases desbalanceadas**.
  - Suele ser bastante común que haya muchos registros negativos (clase 0) y muy pocos positivos (clase 1).
- COVID-19. El número de registros negativos (clase 0) es mucho más alto que el de registros positivos (clase 1).
  - Este desbalance desemboca en que nuestro interés principal se enfoque en la capacidad del modelo para predecir la clase minoritaria (clase 1).
  - No interesa el rendimiento de la clase 0, por lo que no se le presta atención a los verdaderos negativos.
- La clave del uso de la curva de precisión-sensibilidad es que no tiene en cuenta los falsos negativos. La curva de precisión-sensibilidad solo se preocupa de la clase positiva, es decir, de la clase minoritaria.

Utilidad Precisión-Sensibilidad

- Valor F (F-Score): Calcula la media armónica de la precisión y la sensibilidad. (Se utiliza la media armónica porque ambos valores son tasas)

$$F - measure = 2\frac{Precision * Recall}{Precision + Recall}$$

  - En términos de elección del modelo, el valor F resume la habilidad del modelo para un valor específico del umbral.
- Área bajo la curva: **AUC**. Al igual que en la curva ROC, el valor del área bajo la curva nos permite también determinar el rendimiento del modelo
  - Esto convierte a la curva precisión-sensibilidad en la manera óptima de medir el rendimiento de modelos de clasificación binaria cuyas clases **no estén balanceadas**

¿Cuándo utilizar la curva ROC y cuándo utilizar la curva de precisión-sensibilidad?

- Como norma general, el criterio a seguir a la hora de elegir es el nivel de desbalance entre las clases que presenta el modelo
  - Las curvas ROC se deberían utilizar cuando más o menos existen las mismas observaciones para ambas clases.
  - Las curvas de precisión sensibilidad se deberían utilizar cuando existe un notable desbalance entre el número de observaciones de cada clase

- La razón reside en que las curvas ROC nos muestran una versión «optimista» de aquellos modelos con un desbalance de clases considerado. Esta versión optimista se debe a que la curva ROC utiliza la tasa de falsos positivos mientras que la curva de precisión-sensibilidad omite esta tasa.

Ejemplo Notebook Python

- USACH_clase1_2022.ipynb
  - Un ejemplo de *clasificación*
  - Clasificación *binaria*
  - Medidas de desempeño
  - Curvas ROC
  - Curvas Precisión-Sensibilidad

¡Manos a la obra!