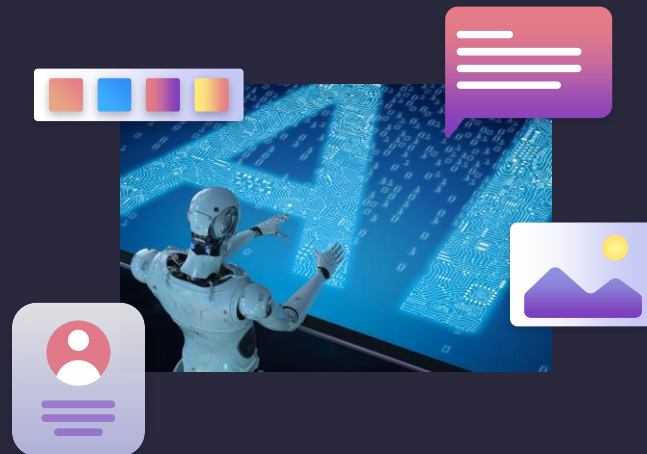


/TEMA 5 – RNAs III

Evaluación de modelos de IA



Sobre los Resultados de Aprendizaje de esta unidad....

3. Evalúa las mejoras en los negocios integrando convergencia tecnológica.

Criterios de evaluación:

- a) Se han identificado las ventajas que ofrece unificar procesos, servicios, herramientas, métodos y sectores.
- b) Se han identificado sistemas que facilitan la conexión tecnológica.
- c) Se han evaluado las características de dichos sistemas.
- d) Se ha evaluado como la convergencia tecnológica aporta seguridad en los negocios.
- e) Se ha evaluado la mejora en la capacidad de toma de decisiones estratégicas en un negocio conectado.

4. Evalúa modelos de automatización industrial y de negocio relacionándolos con los resultados esperados por las empresas.

Criterios de evaluación:

- a) Se han identificado las nuevas estrategias corporativas y modelos de negocio en las empresas.
- b) Se ha definido la relación entre empresas y clientes y su efecto en la forma en que las empresas organizan y gestionan sus activos y recursos.
- c) Se han evaluado modelos de automatización para los nuevos requerimientos industriales y de negocio.
- d) Se ha evaluado la conveniencia de cada modelo para conseguir los resultados esperados por las empresas.

/CONTENIDOS



/00 /EVALUACIÓN DEL SOFTWARE

/01 /EVALUACIÓN DE RESULTADOS EN I.A.

/02 /EVALUACIÓN DE LOS MODELOS DE CLASIFICACIÓN

/03 /MÉTRICAS PARA MODELOS DE CLASIFICACIÓN

/04 /EVALUACIÓN DE MODELOS DE NEGOCIO



/00 /EVALUACIÓN DEL SOFTWARE

CONCEPTOS BÁSICOS

MEDIDA

Una medida proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto.

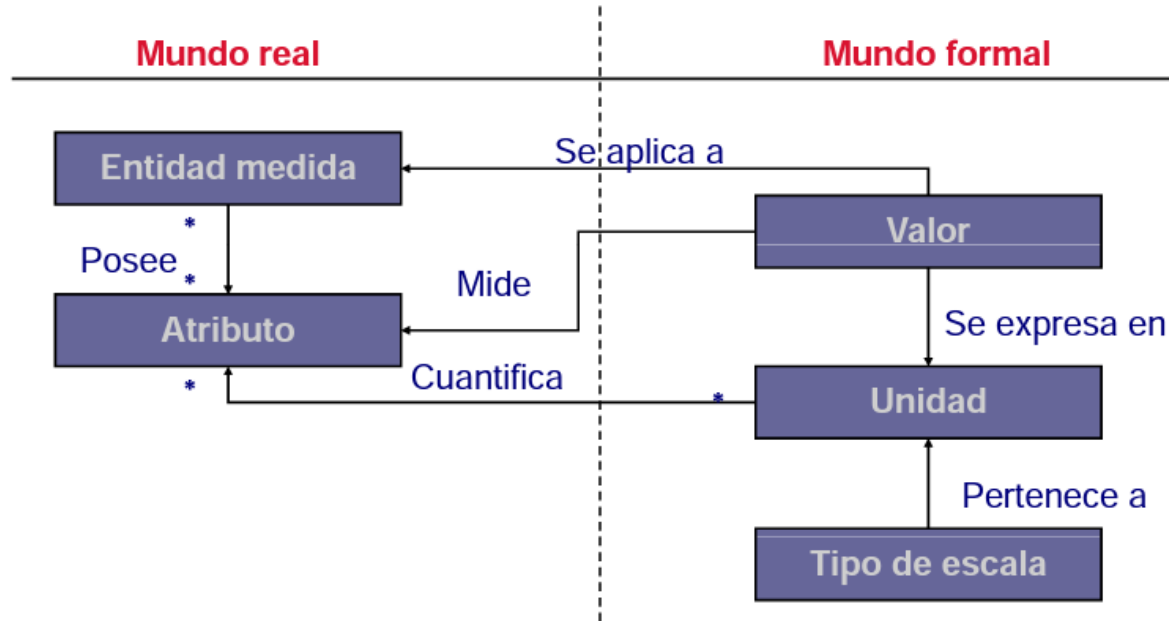
MÉTRICA

Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (IEEE, 1993).

INDICADOR

Métrica o combinación de métricas que proporcionan una visión profunda del proceso de software, del proyecto de software o del producto en sí (Ragland, 1995).

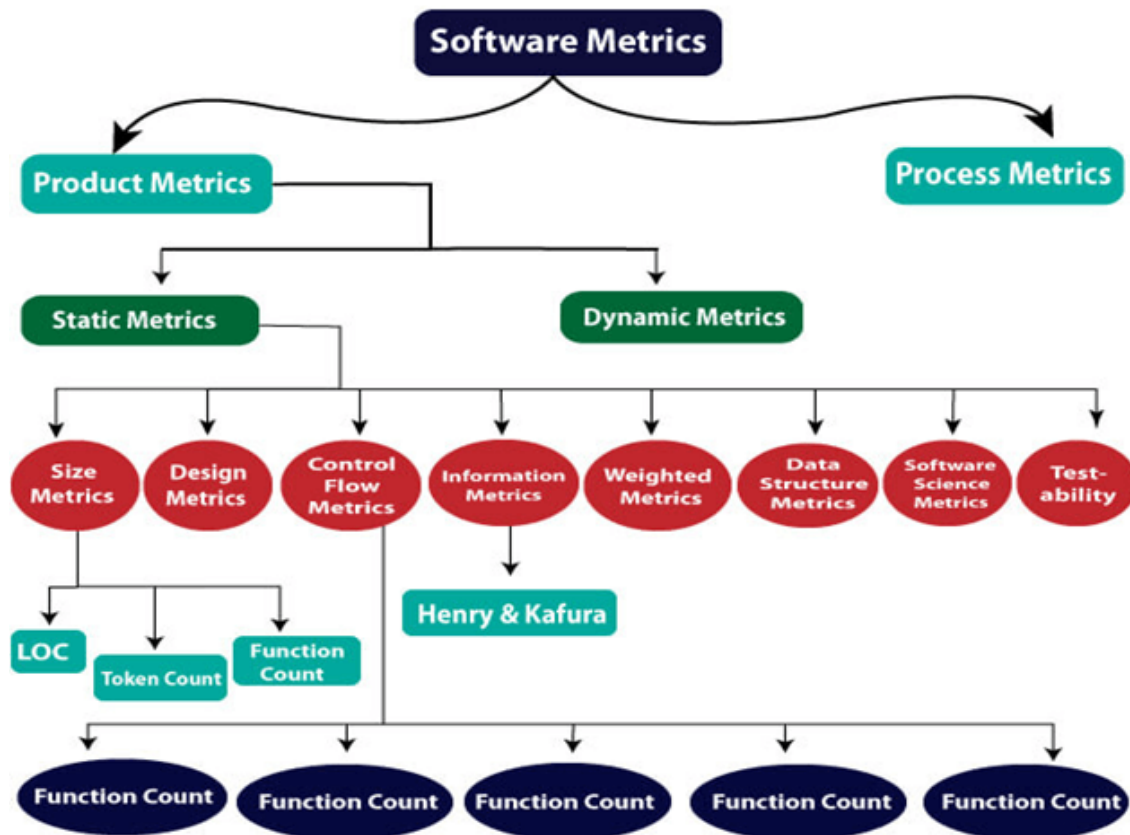
Elementos implicados en la medición [Modelo estructural de Kitchenham]



Las métricas del software abarcan muchas actividades:

- ☐ Estimación de coste y esfuerzo
- ☐ Modelos y medidas de productividad
- ☐ Modelos y medidas de calidad
- ☐ Modelos de fiabilidad
- ☐ Evaluación del rendimiento
- ☐ Métricas estructurales y de complejidad
- ☐ Valoración de capacidad de madurez
- ☐ Gestión mediante métricas
- ☐ Evaluación de métodos y herramientas







EVALUACIÓN DEL SOFTWARE

VS

EVALUACIÓN DE MODELOS DE I.A.

VS

EVALUACIÓN DE MODELOS NEGOCIOS CON I.A



/01 /EVALUACIÓN DE RESULTADOS EN I.A.

La evaluación de un modelo de aprendizaje en Inteligencia Artificial forma parte de su ciclo de vida: Podemos cuantificar la **calidad** del modelo y su **capacidad de predicción**

En la mayoría de las ocasiones existe una realimentación entre la fase de evaluación y la de diseño, tal que una evaluación con resultados insatisfactorios da pie a modificaciones sobre el diseño para mejorar la calidad.

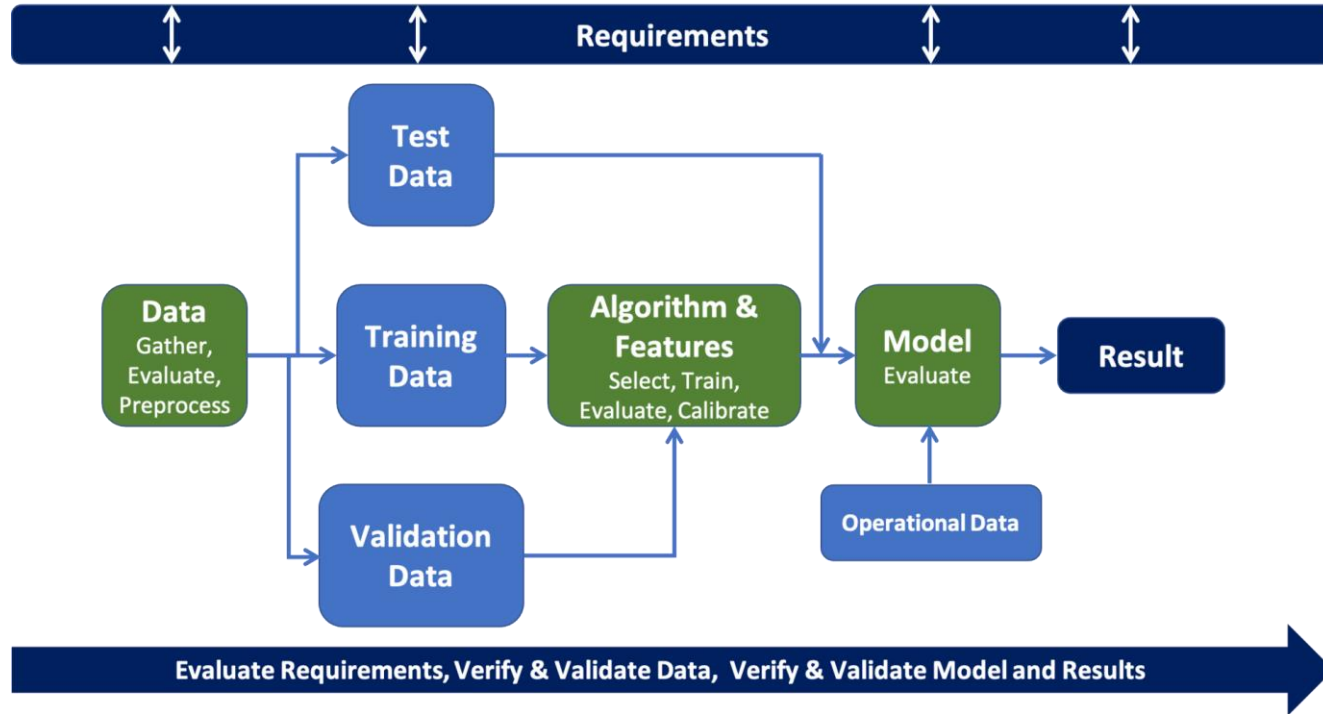
En una encuesta a miles de ingenieros, Ishikawa y Yoshioka (2019) identifican atributos del aprendizaje automático que dificultan la ingeniería del mismo. Según los ingenieros encuestados, los principales atributos son:

- ❑ Falta de un oráculo: es difícil o imposible definir claramente los criterios correctos para las salidas del sistema o las salidas correctas para cada entrada individual.
- ❑ Imperfección: es intrínsecamente imposible que un sistema de IA sea 100% preciso.
- ❑ Comportamiento incierto para datos no probados: existe una gran incertidumbre sobre cómo se comportará el sistema en respuesta a datos de entrada no probados, como lo demuestran los cambios radicales en el comportamiento dados cambios leves en la entrada (por ejemplo, ejemplos contradictorios).
- ❑ Alta dependencia del comportamiento de los datos de entrenamiento: el comportamiento del sistema depende en gran medida de los datos de entrenamiento.

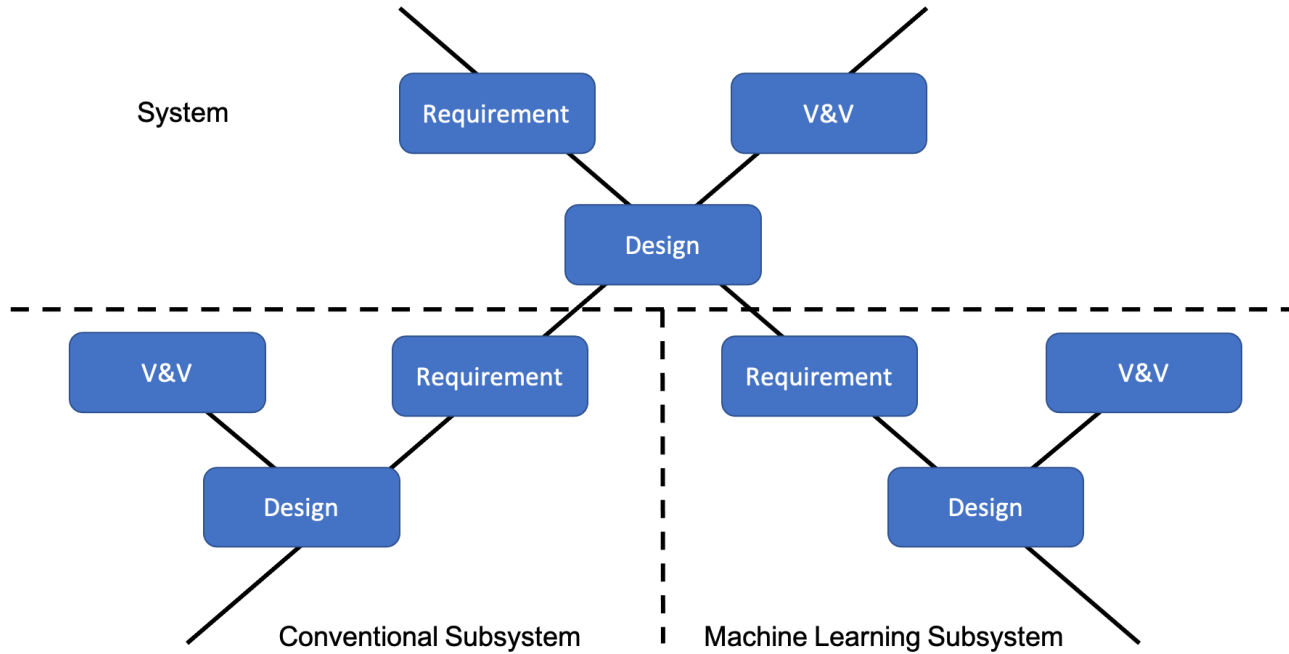
Estos atributos son característicos de la propia IA y se pueden generalizar de la siguiente manera:

- ❑ Erosión del determinismo (determinismo=nada sucede al azar)
- ❑ Imprevisibilidad e inexplicabilidad de los resultados individuales
- ❑ Comportamiento emergente imprevisto y consecuencias no deseadas de la toma de decisiones complejas de los algoritmos.
- ❑ Dificultad para mantener la consistencia y debilidad frente a ligeros cambios en las entradas

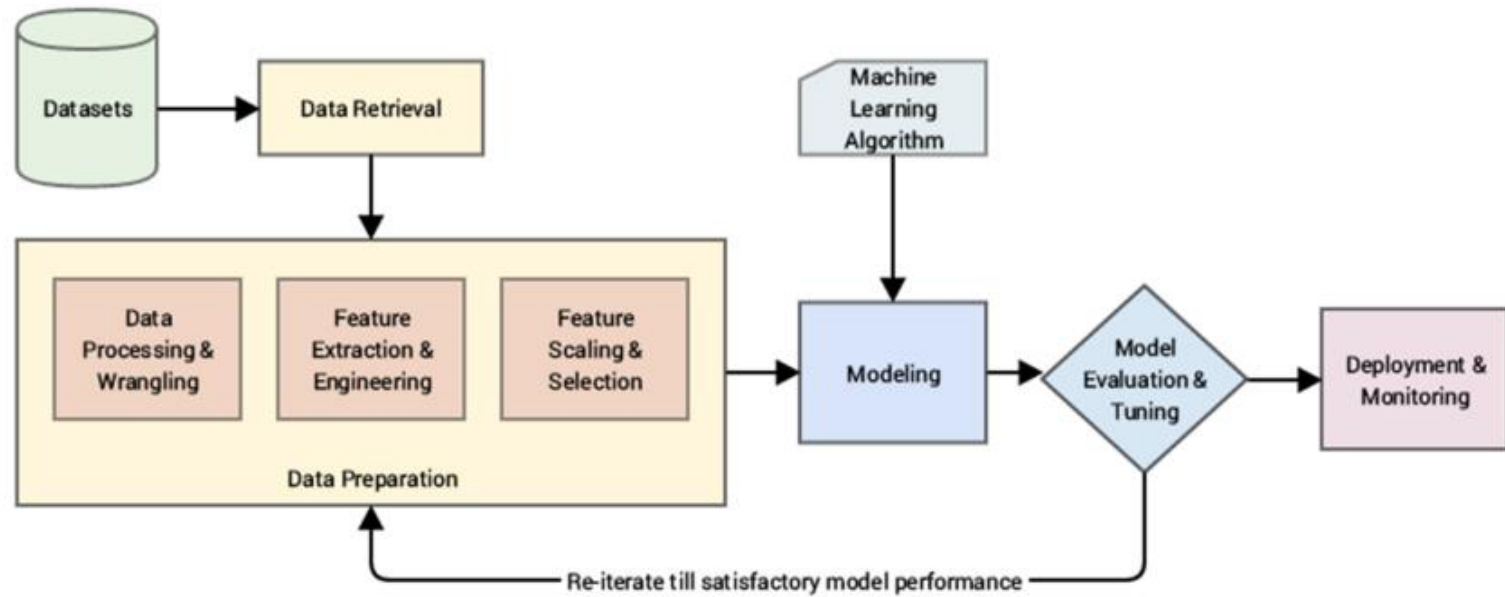
Ciclo de vida genérico de una aplicación con I.A. (Swebok)



V&V = Verification & Validation



CRISP-DM model



/02 /EVALUACIÓN DE LOS MODELOS DE CLASIFICACIÓN

Debido a la propia naturaleza de la I.A. siempre hay que evaluar la calidad y rendimiento del modelo aprendido, cuantificando mediante **varias métricas**.

La más simple: proporción de aciertos en la clasificación dada (**accuracy**)

La evaluación final de un modelo nunca debe hacerse sobre los datos que sirven para aprender el modelo. Ni siquiera sobre los datos que sirven para ajustar el modelo.

La manera mas básica de hacerlo:

Dividir el conjunto de datos disponibles en entrenamiento y prueba

Aprender con el conjunto de entrenamiento

Evaluar con el de prueba

A veces no es posible (pocos datos)

Metodologías para la evaluación:

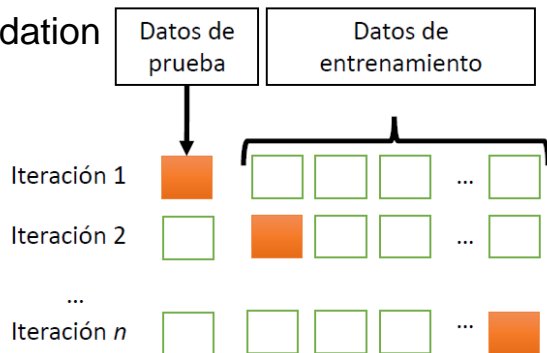
¿Cómo diseñamos el experimento de evaluación del modelo?

Tres métodos posibles:

HoldOut

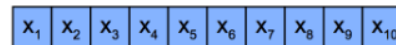


Cross-validation

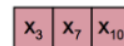
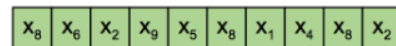


Bootstrap

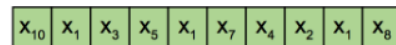
Original Dataset



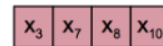
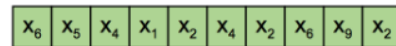
Bootstrap 1



Bootstrap 2



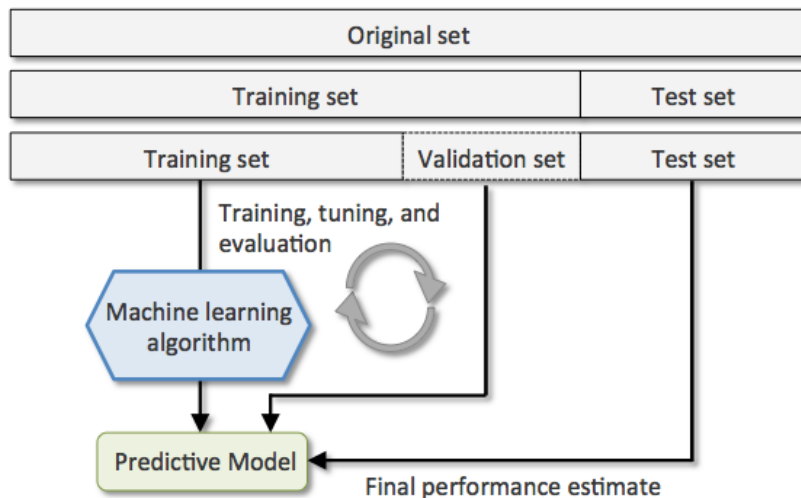
Bootstrap 3



Training Sets

Test Sets

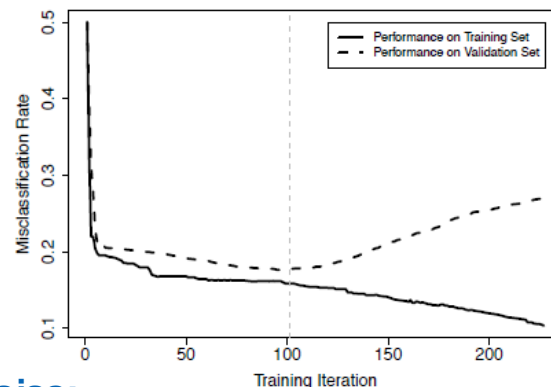
Método Holdout



Ventajas:

Permite prevenir sobreajuste

- se puede estimar dónde comienza a suceder el sobreajuste



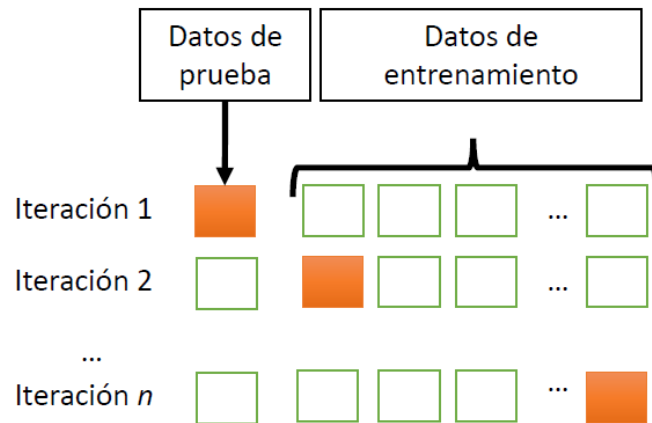
desventajas:

- Clases no balanceadas (900 + / 100-)
- A veces no podemos permitirnos ese "lujo"
- Puede que en el conjunto de entrenamiento queden los datos fáciles

La validación cruzada (cross validation):

Intenta solucionar los problemas del método HoldOut

- 1: Divide el conjunto de entrenamiento entre entrenamiento y validación [80-20%, 70-30%, 90-10%...]
2. Mezcla cada batch de entrenamiento y sepáralo en iteraciones.
3. Por cada iteración, entrena un bloque y prueba el modelo con los datos de test, obteniendo métricas.
4. En cada iteración se puede elegir al azar los datos que forman parte del entrenamiento y de la prueba



La validación cruzada en K partes (k-fold cross validation):

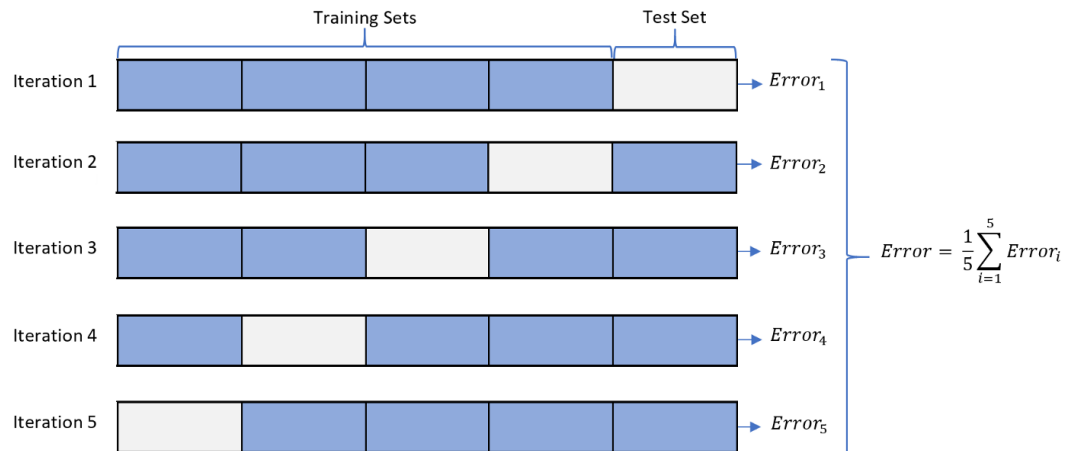
Variante del anterior

1: Se divide el conjunto de entrenamiento en k partes de igual tamaño, preferentemente estratificadas (valores usuales, k=10, k=5).

2. Se hacen k aprendizajes con sus correspondientes evaluaciones (con una métrica dada).

3. En cada aprendizaje, se usa como test una de las partes y como entrenamiento las k-1 restantes.

4. Se devuelve la media de las evaluaciones realizadas



Ventajas de la validación cruzada en K partes:

- ☐ Si no tenemos muchos datos, nos proporciona una buena manera de realizar el proceso de validación.
- ☐ Cada dato aparece exactamente una vez en un conjunto de test: ningún ejemplo se “escapa” del entrenamiento ni de la evaluación.
- ☐ Podemos medir la varianza de las evaluaciones entre distintos conjuntos de prueba.

Desventajas:

- ☐ Tiempo de computación

La validación cruzada (cross validation):



La validación cruzada no es un método para entrenar un modelo



Es una manera de evaluar cómo de bueno será (en términos de generalización) un algoritmo de aprendizaje sobre un conjunto de entrenamiento dado



Se suele usar, por ejemplo, para el ajuste de parámetros. Finalmente, se suele entrenar un modelo sobre todo el conjunto de entrenamiento, y se evalúa ese modelo sobre un conjunto de prueba independiente.

Bootstrap

- ❑ En cada iteración se obtiene una muestra del mismo tamaño que el conjunto de datos total, pero cada elemento se extrae aleatoriamente con reemplazo.
- ❑ Es decir, la muestra puede tener repeticiones y algunos datos pueden no entrar
- ❑ Se entrena con el conjunto extraído, y se evalúa sobre el conjunto original
- ❑ Se devuelve la media de todas las iteraciones
- ❑ Estimación optimista (gran solape entre conjuntos de entrenamiento y prueba)



/03 / MÉTRICAS PARA MODELOS DE CLASIFICACIÓN





En este apartado vamos a estudiar cómo podemos utilizar las métricas para los diferentes modelos de clasificación:

Ten en cuenta que cada métrica está adaptada a diversos modelos de aprendizaje

		predicted condition		
total population		prediction positive	prediction negative	Sensitivity
true condition	condition positive	True Positive (TP)	False Negative (FN) (Type II error)	Recall = $\frac{\sum TP}{\sum \text{condition positive}}$
	condition negative	False Positive (FP) (Type I error)	True Negative (TN)	Specificity = $\frac{\sum TN}{\sum \text{condition negative}}$
Accuracy = $\frac{\sum TP + \sum TN}{\sum \text{total population}}$		Precision = $\frac{\sum TP}{\sum \text{prediction positive}}$		F1 Score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$

LA MATRIZ DE CONFUSIÓN

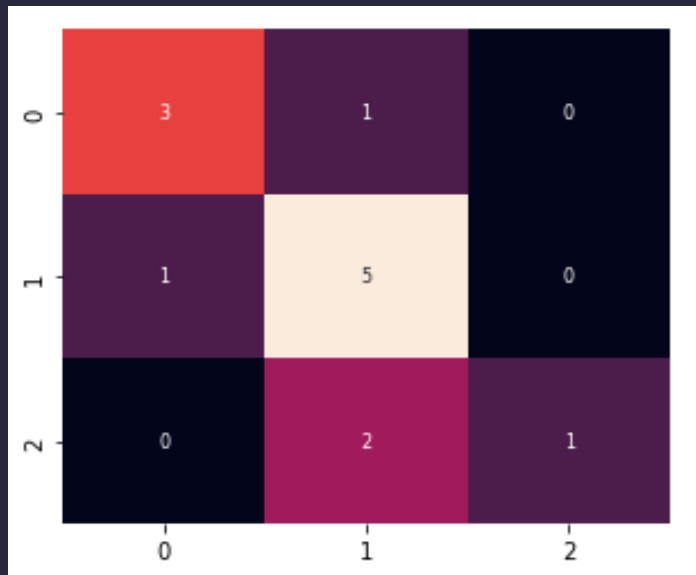
- ❑ *no puede ser considerada una métrica como tal*
- ❑ *fundamental para evaluar y optimizar los modelos de clasificación*
- ❑ *ayuda a profundizar en el tipo de error que el modelo está cometiendo*
- ❑ *ayuda a comprender otras métricas que se emplean*

		PREDICTED VALUES	
		Positive (CAT)	Negative (DOG)
ACTUAL VALUES	Positive (CAT)	 TRUE POSITIVE 6 YOU ARE A CAT	 FALSE NEGATIVE 1 TYPE II ERROR YOU ARE A DOG
	Negative (DOG)	 FALSE POSITIVE 2 TYPE I ERROR YOU ARE A CAT	 TRUE NEGATIVE 11 YOU ARE NOT A CAT

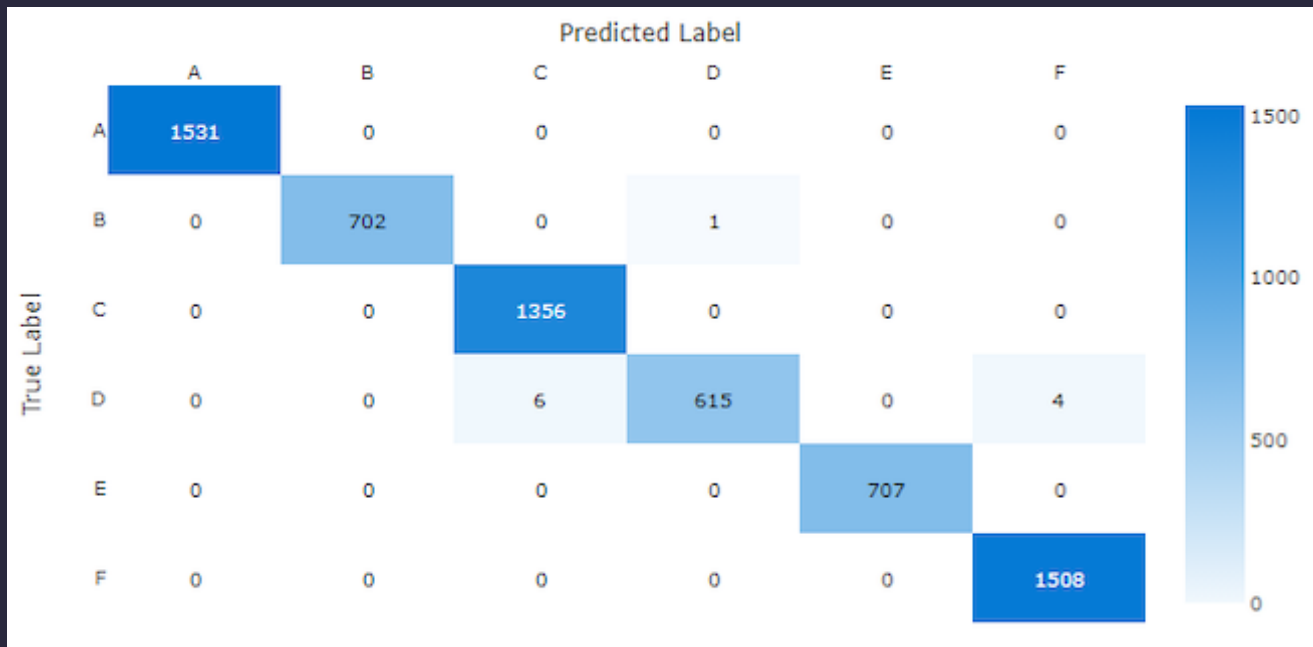
Imprimir MATRIZ DE CONFUSIÓN CON SKLearn y Seaborn

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sn
y_real = [1, 1, 2, 1, 1, 0, 2, 0, 0, 0, 2, 1, 1]
y_pred = [1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 2, 1, 1]
cm = confusion_matrix(y_real, y_pred)
```

```
#Creación de Figura
sn.heatmap(cm, annot=True, fmt='d') # font size
plt.show()
```



Matriz de confusión para un buen modelo



Matriz de confusión para un mal modelo

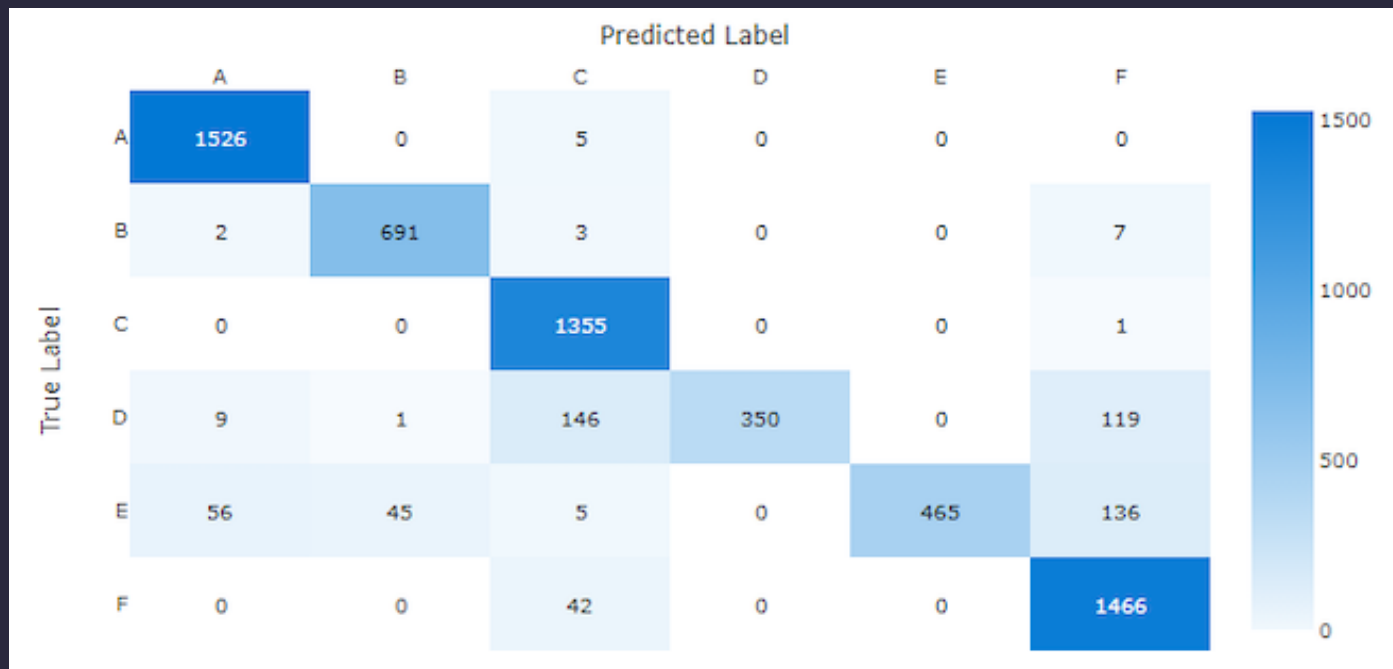


Figura más completa: Métricas para modelos de I.A. (Wikipedia)

		Predicted condition			
Total population $= P + N$		Positive (PP)	Negative (PN)	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$		False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR-) $= \frac{FNR}{TNR}$
Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$		Negative predictive value (NPV) $= \frac{TN}{PN}$ $= 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$		Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \frac{\sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}}{1}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Dependiendo del tipo de modelo se emplean unos métodos de evaluación u otros. Por ejemplo, para clasificación en scikit-learn, se pueden usar las siguientes métricas:

Scoring	Function	Comment
Classification		
'accuracy'	<code>metrics.accuracy_score</code>	
'balanced_accuracy'	<code>metrics.balanced_accuracy_score</code>	
'top_k_accuracy'	<code>metrics.top_k_accuracy_score</code>	
'average_precision'	<code>metrics.average_precision_score</code>	
'neg_brier_score'	<code>metrics.brier_score_loss</code>	
'f1'	<code>metrics.f1_score</code>	for binary targets
'f1_micro'	<code>metrics.f1_score</code>	micro-averaged
'f1_macro'	<code>metrics.f1_score</code>	macro-averaged
'f1_weighted'	<code>metrics.f1_score</code>	weighted average
'f1_samples'	<code>metrics.f1_score</code>	by multilabel sample
'neg_log_loss'	<code>metrics.log_loss</code>	requires <code>predict_proba</code> support
'precision' etc.	<code>metrics.precision_score</code>	suffixes apply as with 'f1'
'recall' etc.	<code>metrics.recall_score</code>	suffixes apply as with 'f1'
'jaccard' etc.	<code>metrics.jaccard_score</code>	suffixes apply as with 'f1'
'roc_auc'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovr'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovo'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovr_weighted'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovo_weighted'	<code>metrics.roc_auc_score</code>	

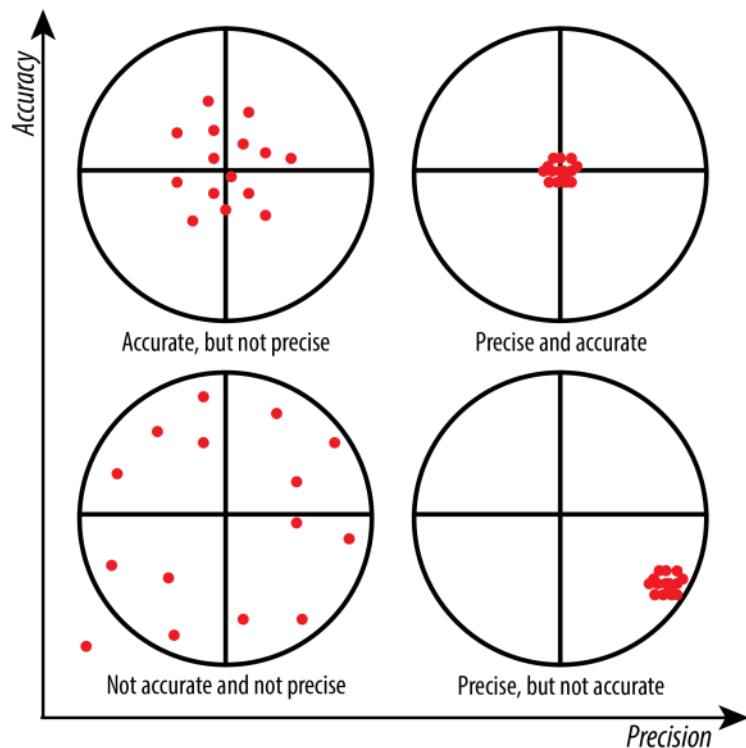
https://scikit-learn.org/stable/modules/model_evaluation.html

Diferencia?



Para clustering y regresión, se utilizan algunas más.

Accuracy vs precision



En castellano, tendemos a traducir accuracy por “precisión”. En inglés, no es lo mismo accuracy que precision

English – detected	↔	Spanish
accuracy	×	precisión
'akyeresē		

En realidad, deberíamos traducir:

Accuracy = exactitud

Precision = precisión

Epoch 23/25

79/79 [=====] - 47s 593ms/step - loss: 0.0639 - accuracy: 0.9782 - val_loss: 0.0475 - val_accuracy: 0.9704

Epoch 24/25

79/79 [=====] - 47s 588ms/step - loss: 0.0637 - accuracy: 0.9817 - val_loss: 0.0196 - val_accuracy: 0.9946

Epoch 25/25

79/79 [=====] - 47s 587ms/step - loss: 0.0439 - accuracy: 0.9829 - val_loss: 0.0105 - val_accuracy: 1.0000

/train/rock/rock01-000.png



/train/rock/rock01-001.png



/train/paper/paper01-000.png



/train/paper/paper01-001.png



/train/scissors/scissors01-000.png



/train/scissors/scissors01-001.png

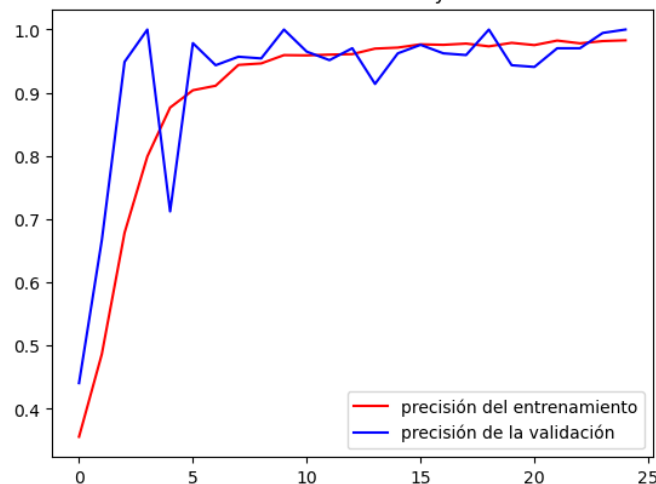


La tasa de aciertos “accuracy” no siempre es una buena métrica, **podemos obtener resultados “sesgados”** por el conjunto de entrenamiento:

¿Qué ocurre si el 90% de las clases es A y el 10% B?
Es posible que prediga que todas las clases son A,
90% de accuracy



Precisión de entrenamiento y validación



Accuracy vs Precision:

1000 personas: Queremos un modelo que nos diga si una persona está enferma (clase positiva)
 El 90% de la gente está sana
 Modelo: Predice que todos están sanos

		Predichas	
		P	N
Realidad	P	0	900
	N	0	100

Suma TP = 0 enfermos

Suma TN = 900 sanos

Accuracy = $(0+900)/1000=0.9 \rightarrow 90\%$

Precisión de enfermos = $0/1000 = 0$ (la precisión más baja)



Las etiquetas negativas no se tienen en cuenta

Accuracy =	Precision =
$\frac{\sum TP + \sum TN}{\sum \text{total population}}$	$\frac{\sum TP}{\sum \text{prediction positive}}$



Recall

Imagina que los enfermos que tenemos que detectar es que tienen un virus muy contagioso

Si levantamos una falsa alarma (decimos que un sano tiene el virus), es grave;

Pero es más grave diagnosticarlo como sano y dejarlo libre y permitir que el virus se extienda.

Recall nos ayuda a detectar estos casos:
 Nos ayuda cuando los casos positivos no están siendo detectados

Ej: de 24 personas

En realidad 12 infectados, 12 no infectados

El sistema diagnostica según esta matriz:

		Predichas	
		P	N
Realidad	P	4	8
	N	0	12

$$\text{Recall} = \frac{\sum TP}{\sum \text{condition positive}}$$

Sensitivity (indicated by a red arrow pointing to the formula)

$$\text{Accuracy} = (4+12)/24 = 66\%$$

$$\text{Precisión} = 4/4 = 100\%$$

$$\text{Recall} = 4/12 = 33\%$$

Recall + Precision = F1 Score

Observa el siguiente ejemplo de 18 muestras

		Predichas	
		P	N
Realidad	P	1	8
	N	1	8

Precisión “trucada”

$$\text{Precisión} = 1/1 = 100\%$$

$$\text{Recall} = 1/9 = 11\%$$

		Predichas	
		P	N
Realidad	P	1	1
	N	8	8

Recall “trucado”

$$\text{Precisión} = 1/8 = 12,5\%$$

$$\text{Recall} = 1/2 = 50\%$$

Recall y precisión se complementan: **se pueden combinar** con el **F1 Score** a través de su media armónica

F1 SCORE = “¿Son buenas y completas las predicciones?”

$$\begin{aligned}
 F_1 &= \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\
 &= \frac{2 \times \text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}
 \end{aligned}$$

Accuracy: Bueno para datos balanceados

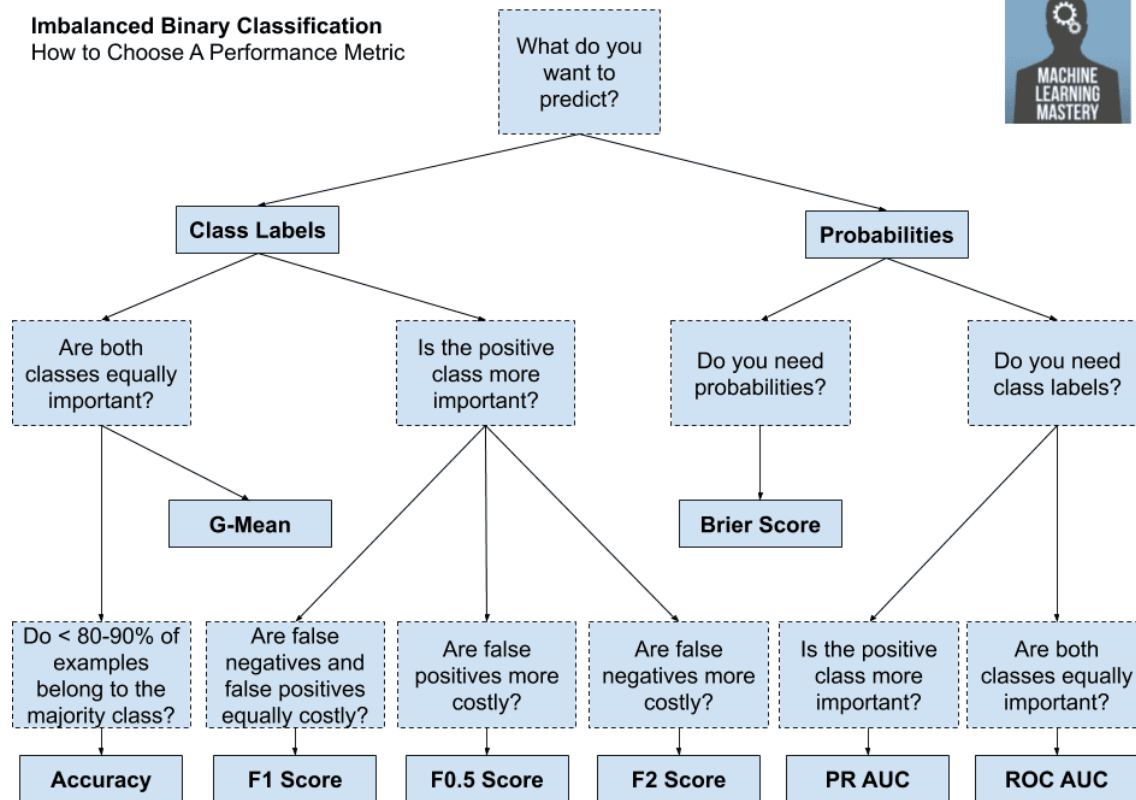
Precision: Bueno cuando el falso positivo es muy importante

Recall: Bueno cuando el falso negativo es muy importante

F1 Score: Bueno cuando tanto los falsos negativos como los falsos positivos son importantes

Otras métricas

Imbalanced Binary Classification How to Choose A Performance Metric



/04 /EVALUACIÓN DE MODELOS DE NEGOCIO

La inteligencia artificial ha cambiado la forma en la que las empresas implementan sus modelos de negocio

- ❑ Aún no se entiende muy bien cómo esta tecnología emergente influencia el crecimiento de los diferentes negocios
- ❑ Algunas empresas son vulnerables a la competencia que incorpora IA en su modelo de negocio
- ❑ Hay estudios que explican cómo la IA transforma negocios
- ❑ El “Círculo virtuoso” de la IA:



Emerging Technology and Business Model Innovation: The Case of Artificial Intelligence



emerging Technology and business model innovation

Industry Sectors

Technology/Media, Manufacturing, Consumer Products, Financial Services,
Health Care, Industrial, Energy, Public Sectors

Preconditions to success

Transform the core business

Scale new business

Grow the core business

AI-Based Business Model Innovation

Execute pilot projects

Build an in-house AI team

Provide broad AI training

Develop an AI strategy

Internal & external Comm.

Enablers

AI Technologies

Organizational Culture

Symbolic AI

Neural AI

Learning and innovation

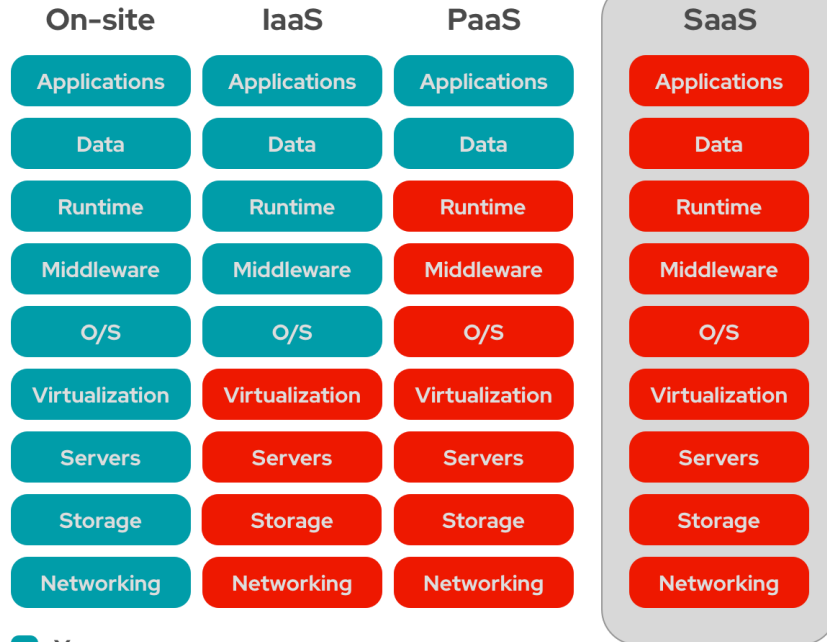
- Knowledge representation
- Reasoning

- Supervised learning
- Unsupervised learning
- Reinforcement learning

- Model innovation
- Willingness to learn and innovate

https://www.researchgate.net/publication/334620177_Emerging_Technology_and_Business_Model_Innovation_The_Case_of_Artificial_Intelligence

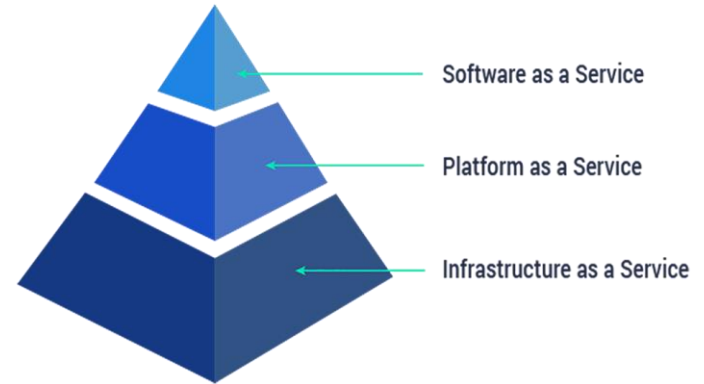
Modelos de negocio para I.A: SaaS vs PaaS vs IaaS

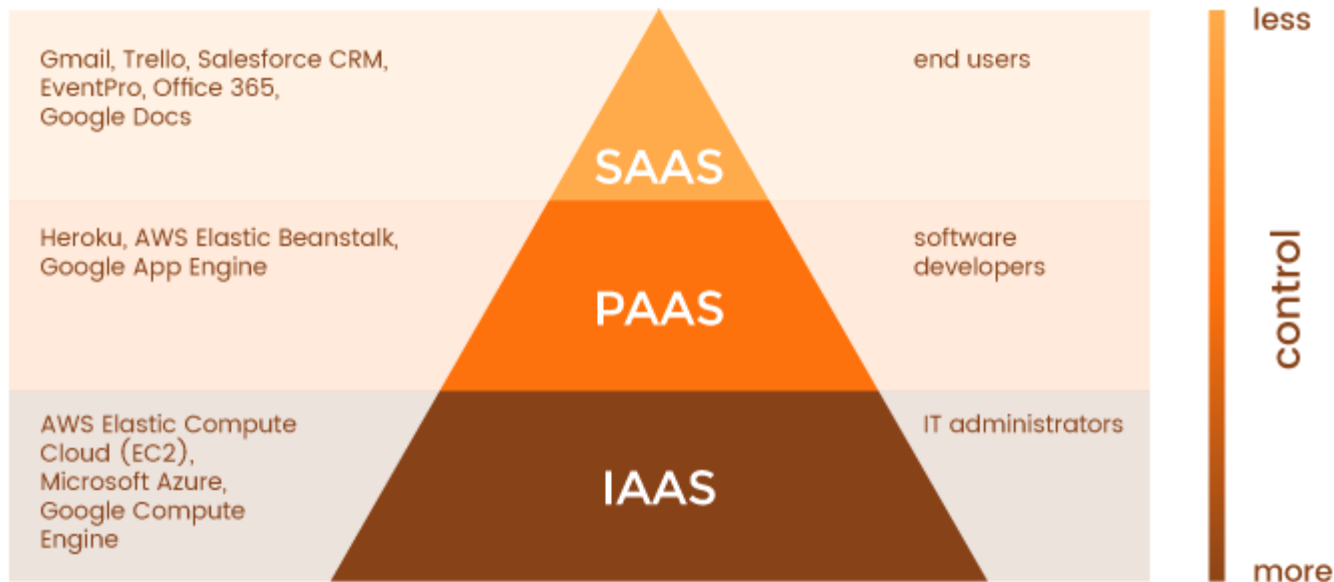


■ You manage

■ Service provider manages

UT7 - Plataformas de inteligencia artificial





AIAAS = Artificial Intelligence as a service



PLATFORM & INFRASTRUCTURE



Google Cloud Platform



PaaS - Pay Per Use



MODELS & ALGORITHMS

CONVERSAIONAL AGENTS



VISION



CORE ALGORITHMS



NLP & SEMANTICS

SPEECH



IaaS - Pay Per Use - MLOps



ENTERPRISE SOLUTIONS

CUSTOMER MANAGEMENT



HR & TALENT



MARKETING & SALES



RPA, OTHERS



INTELLIGENCE & ANALYTICS



CYBERSECURITY

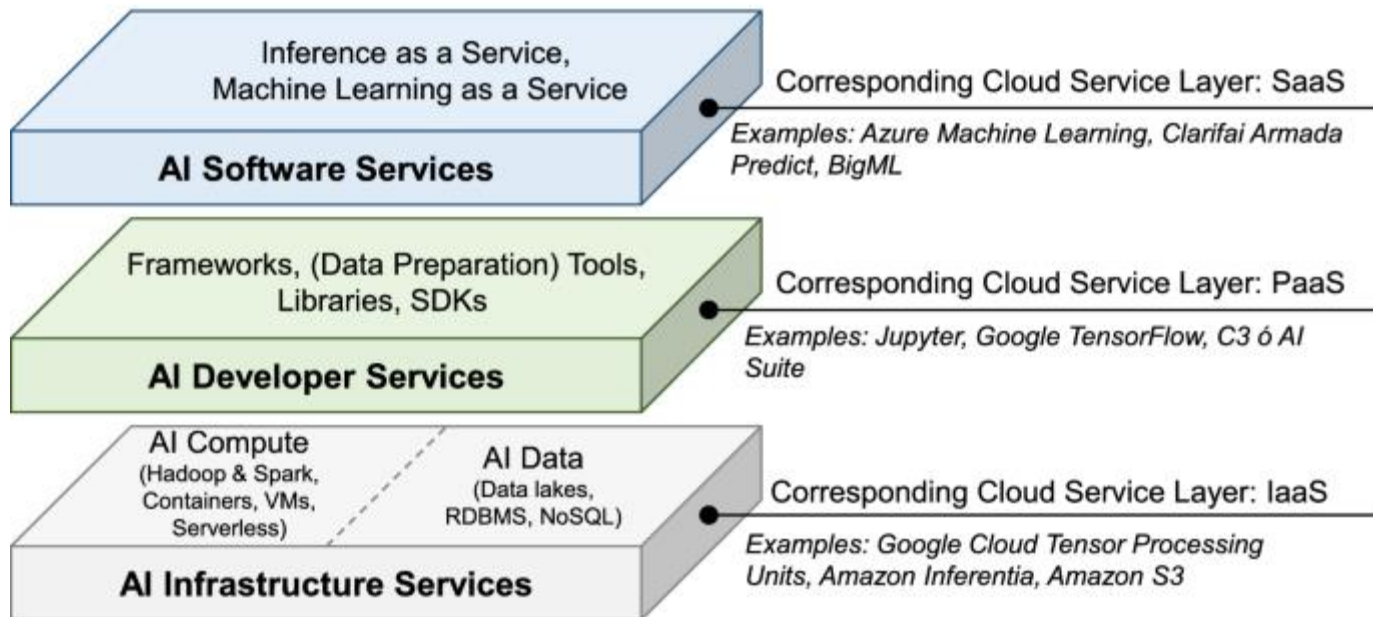


TOOLS



AaaS - SaaS - MLOps

AlaaS Stack



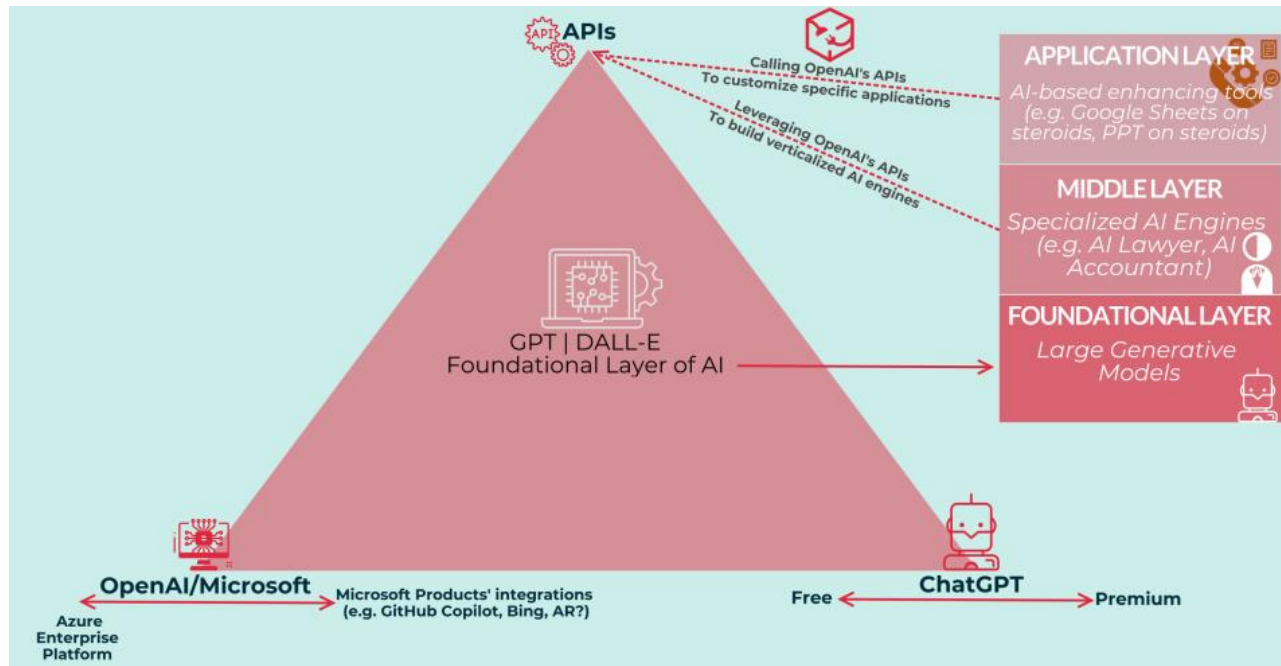
EJEMPLO DE MODELO DE NEGOCIO: Rehabilitación remota (pdf 5. evaluación de modelos de negocio para IA)



Evaluación de modelos de negocio para IA.pdf

EJEMPLO DE MODELO DE NEGOCIO:

Modelo de negocio de Open AI: Permite el acceso a su API a negocios para desarrollar sus aplicaciones por encima de su capa fundacional





<https://gpt3demo.com/> CASOS DE USO I.A.

New

Recently added GPT-3 apps



Generative Media
BuzzFeed



Project Management
checklist.gg



LegalTech
DoNotPay



Voice Cloning
ElevenLabs



Video Editing
Flawless AI



Travel
Getaiway



Essay Writing
GPTZero



Search Engines
Komo AI



Search Engines
Neeva



Music
PlaylistAI



Developer Tools
re:tune



Text-to-Music
Riffusion

See all →

Popular

See all →



Thought experiment generation
10 Thought experiments



Humor
500+ Openers for Tinder wri...



Deepfakes
AI Eminem



Search Engines
Ask me anything



AI Writing Assistants
Compose.ai



Essay Writing
Essay Writing by EduRef



Spreadsheets
Excel (OpenAI Tabulate)

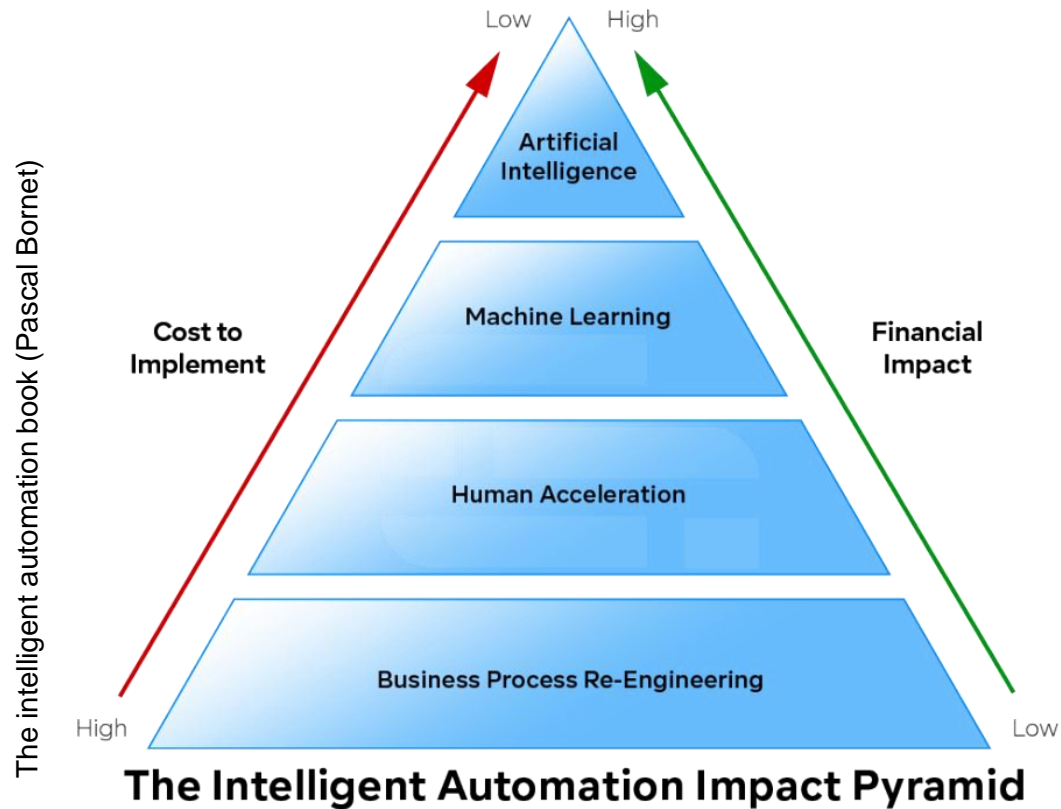


Learning
Learn From Anyone



Image Generation
DALL-E by OpenAI





The roadmap to a successful Intelligent Automation transformation

The intelligent automation book (Pascal Bornet)

