

# END-TO-END MACHINE LEARNING

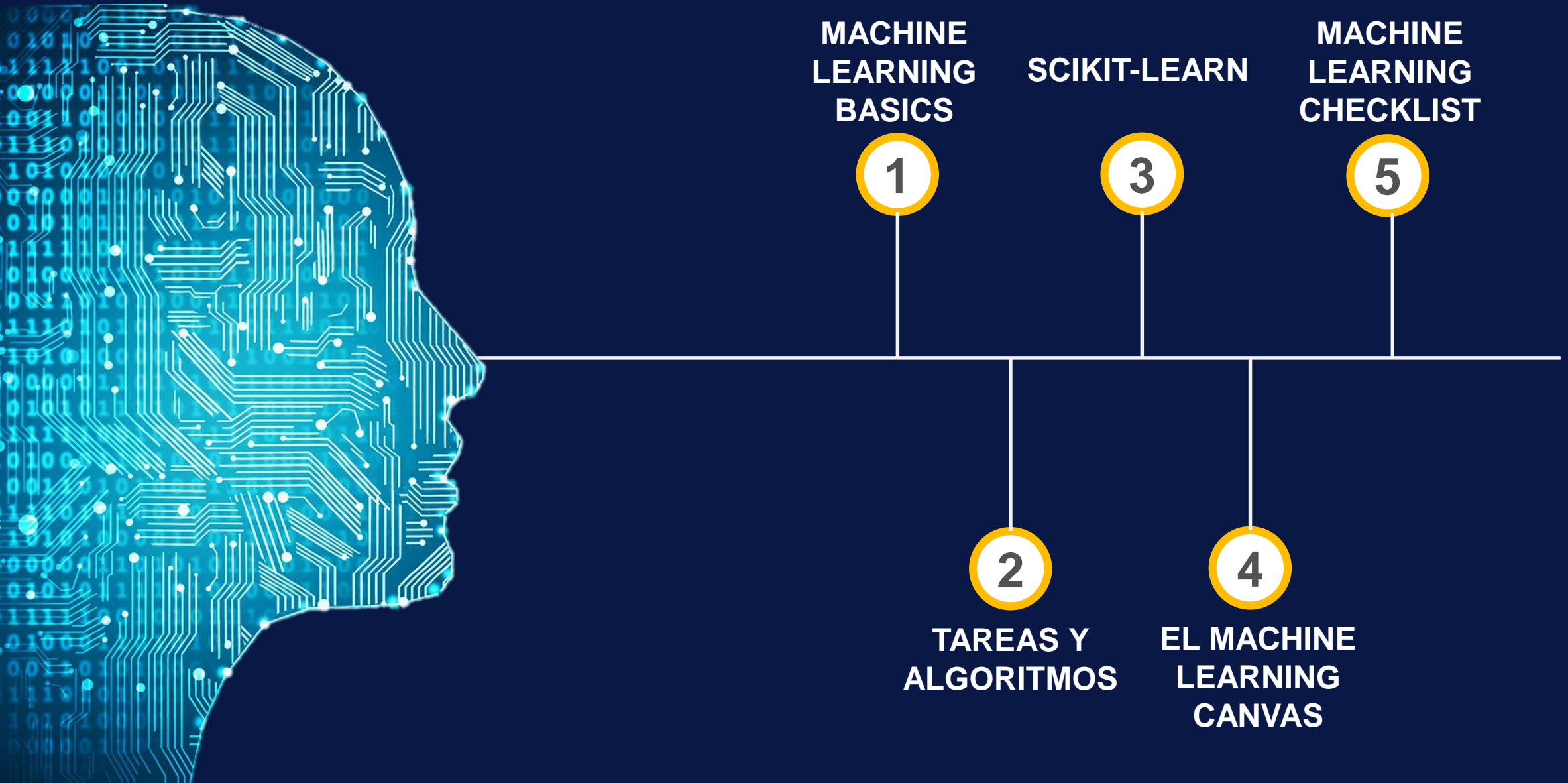
Guillem Sitges i Puy

---

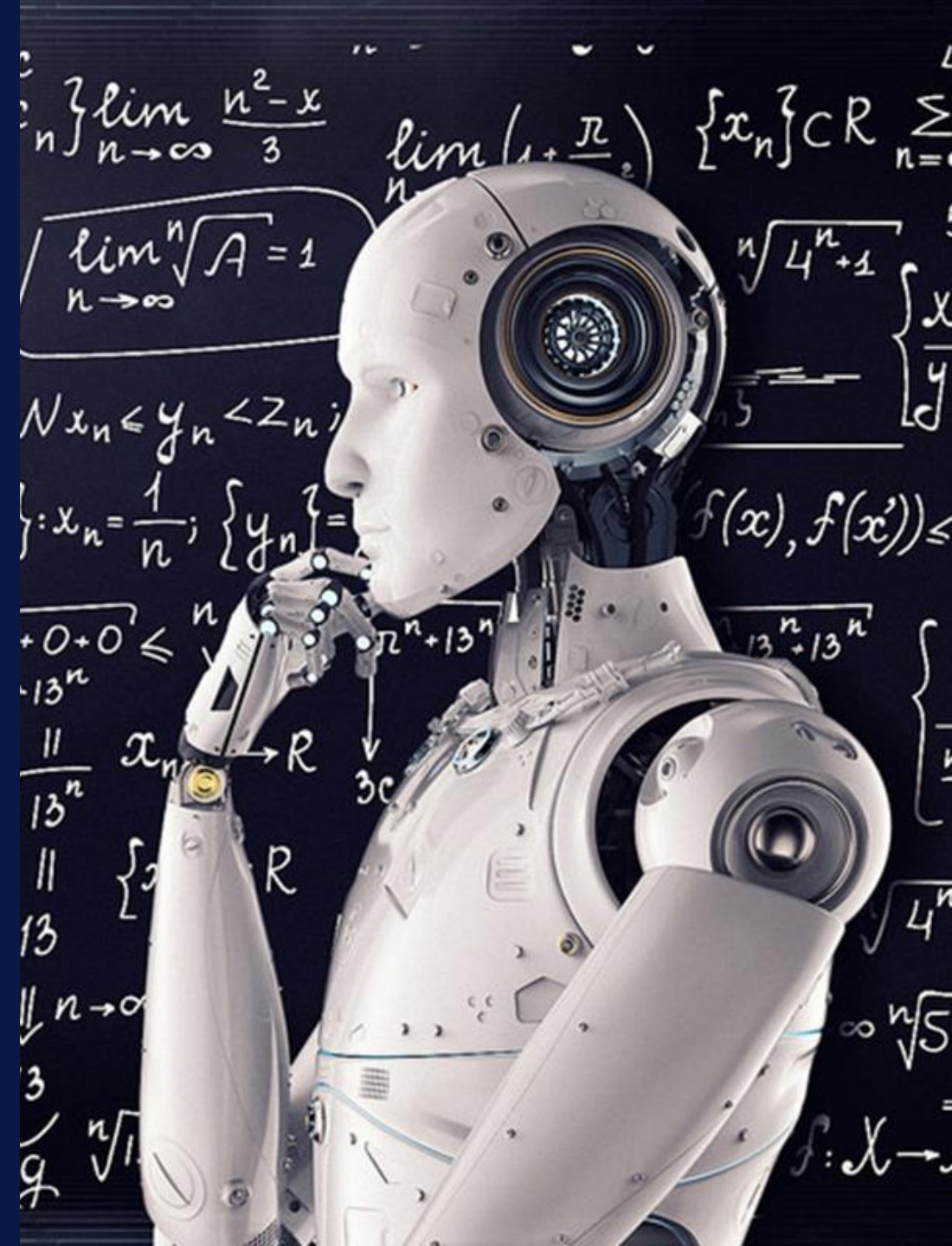
## Sesión 18

*Introducción al aprendizaje supervisado  
ML Checklist*

# ÍNDICE END-TO-END MACHINE LEARNING



# 01. Machine Learning Basics





# 1 ML BASICS: DEFINICIÓN

Nos referimos al **Machine Learning**, Aprendizaje Automático o Reconocimiento de Patrones como el

- **subámbito de la Inteligencia Artificial**
- dedicado a proporcionar a las máquinas la capacidad de **aprender a realizar una tarea en base a la experiencia**,
- sin que se **proporcionen instrucciones explícitas para la realización de la misma (sin programación apriorística)**.

Se dice que una máquina aprende a realizar una tarea (T) en base a la experiencia (E) cuando la medida del rendimiento de la máquina en la realización de la tarea (P) aumenta con la experiencia.

En general, para que haya aprendizaje **se espera capacidad de generalización**, es decir, que la máquina sea capaz de obtener una métrica de rendimiento P similar tanto en experiencias observadas como en nuevas experiencias (de lo contrario, hablamos de memorización).

# 1 ML BASICS : TERMINOLOGÍA

1

## Modelos

- ✓ El proceso de aplicación de Machine Learning tiene como **resultado la generación de un modelo** para resolver una tarea dada.
- ✓ Los modelos son formulas que nos permiten generar una **cantidad de interés (resultado del modelo) a partir de una serie de atributos conocidos**. Dichas fórmulas pueden ser matemáticas, lógicas o una combinación de ambas.

2

## Modelos predictivos vs Modelos descriptivos

- ✓ La **mayoría de modelos generados en Data Science son modelos Predictivos**. En terminología general, predecir se refiere a provisionar o estimar un valor futuro. En Data Science, sin embargo, cuando hablamos de predicción nos referimos a estimar cualquier cantidad desconocida (en el pasado, presente o futuro).
- ✓ Los modelos Predictivos contrastan con los modelos Descriptivos, cuyo propósito no es la estimación de un valor sino la mejor comprensión de un fenómeno o proceso, y también son aplicaciones del Data Science. Como veremos, las mismas técnicas servirán para realizar modelos predictivos y modelos descriptivos.

3

## Inducción de modelos a partir de datos

- ✓ Al **proceso de generar un modelo a partir de los datos se le conoce como inducción o aprendizaje del modelo**. El mecanismo que aprende el modelo a partir del dato se llama **algoritmo** de inducción o aprendizaje (los modelos se aprenden de los datos).
- ✓ Nos referiremos a este proceso de aprendizaje del modelo a partir de los datos mediante un algoritmo como **entrenamiento del modelo**.
- ✓ Los datos etiquetados (que incluyen la clase) que se utilizan para para inducir el modelo se llaman datos de entrenamiento.

# 1 ML BASICS : TERMINOLOGÍA

## Datasets

- ✓ Los datos de entrenamiento (que utilizamos para entrenar el modelo) se recogen en un **Dataset**.
- ✓ Los Datasets contienen instancias u observaciones de una población, y un conjunto de ejemplos representa una muestra de esta población.
- ✓ Cada instancia del Dataset es un vector ordenado y de longitud fija de atributos.
- ✓ En la mayoría de tareas de Machine Learning, **las instancias contendrán un valor de Clase o Target, que es el valor que buscamos predecir**, de manera que aprenderemos un modelo a partir de datos etiquetados para predecir el target de futuras observaciones no etiquetadas.

Predictores o atributos			Target o Clase
Cabeza	Cuerpo	Color	Churn
Cuadrada	Redondo	Blanco	No
Redonda	Redondo	Negro	Yes
Cuadrada	Cuadrado	Blanco	Yes
Cuadrada	Cuadrado	Blanco	Yes
Cuadrada	Cuadrado	Blanco	Yes
Redonda	Cuadrado	Negro	No

Instancias, observaciones, ejemplos o registros

No

Yes

Yes

Yes

Yes

No

Yes

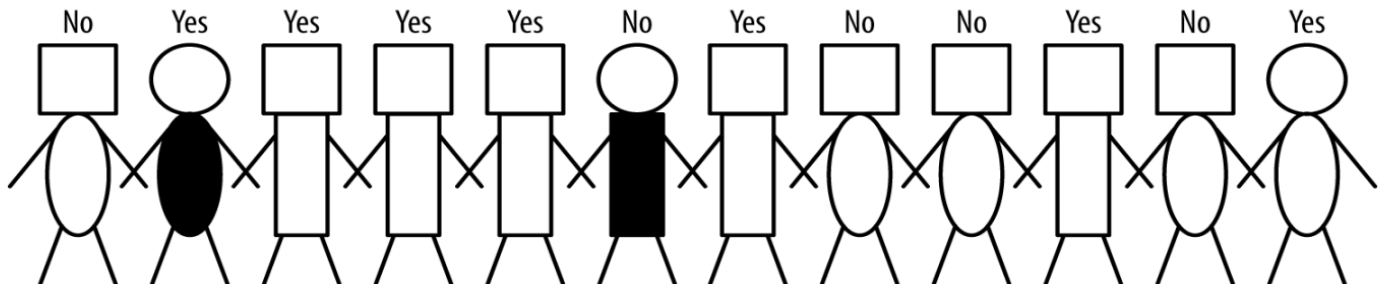
No

No

Yes

No

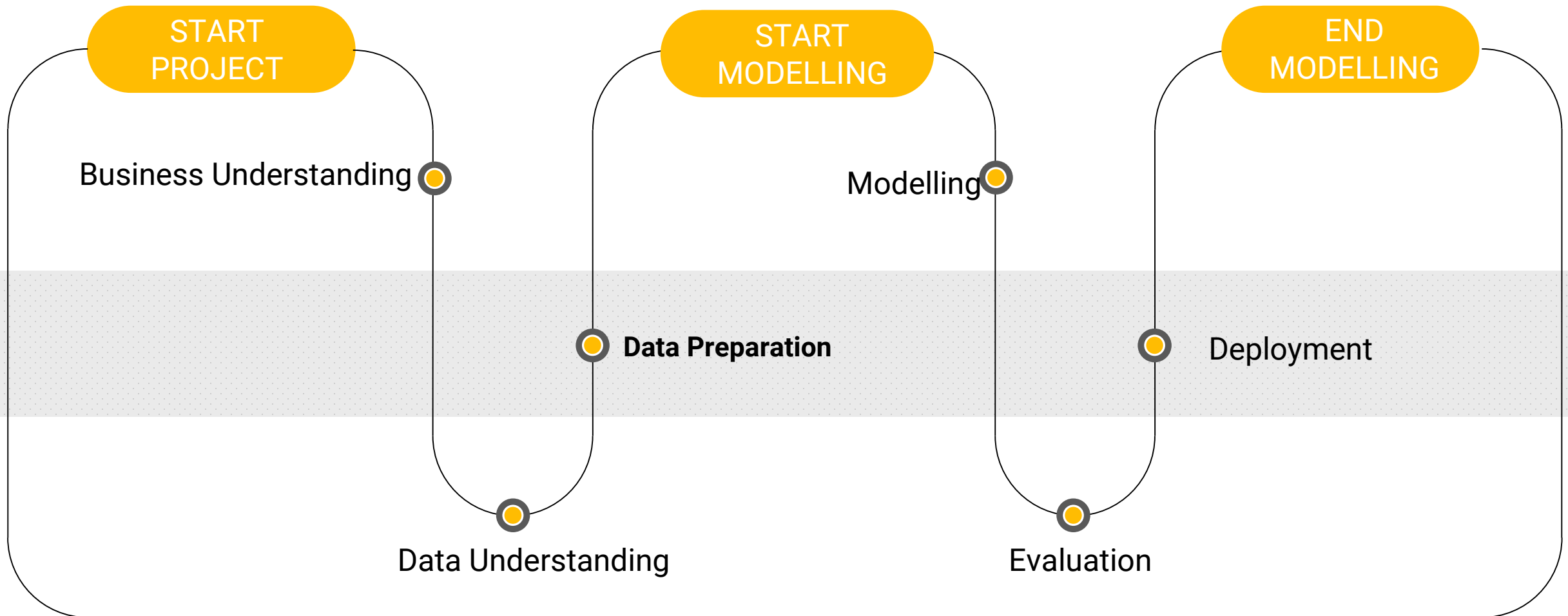
Yes



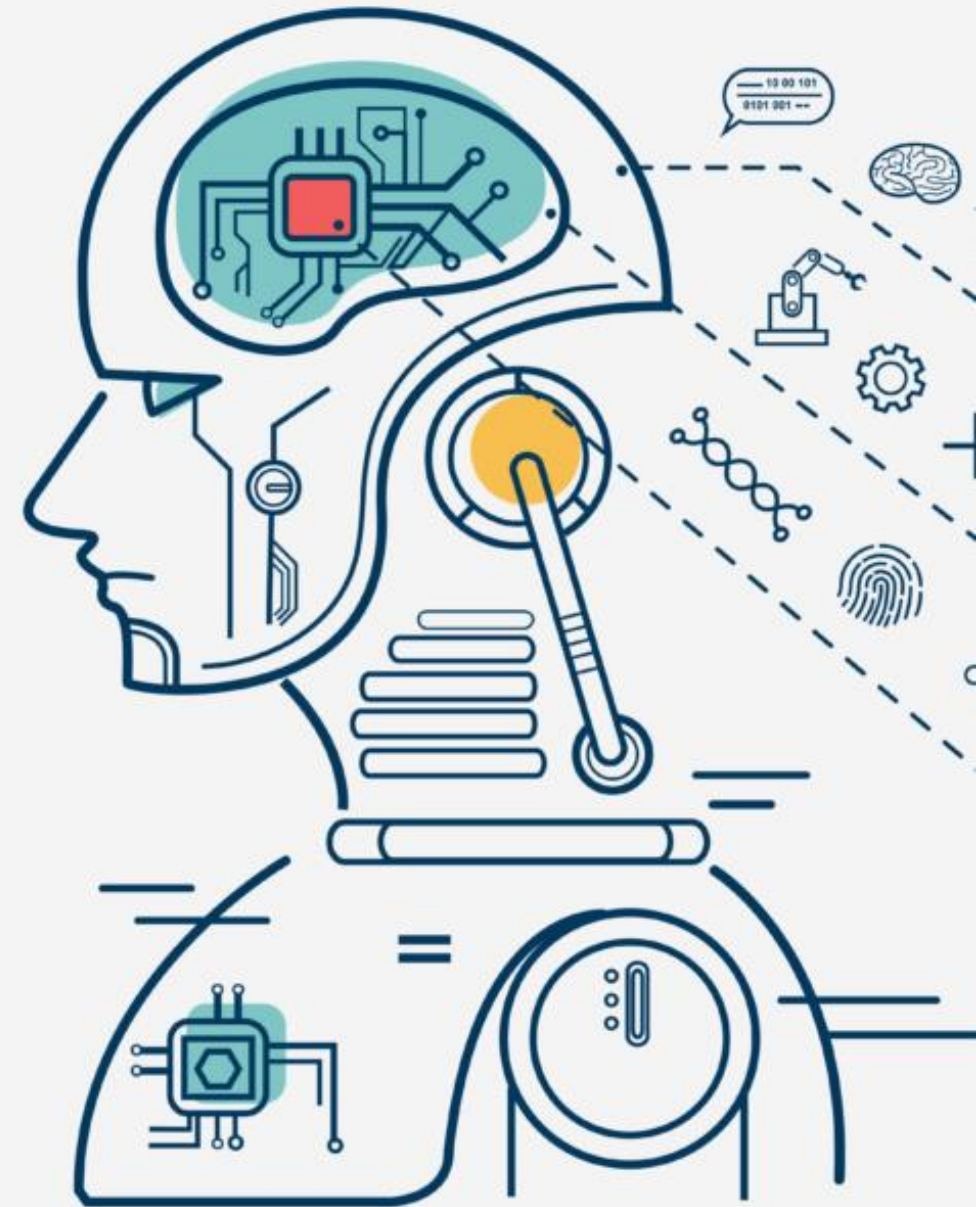
(8) La inducción se refiere a la generalización, a partir de casos concretos, de una regla general (en oposición a la deducción).

# 1 ML BASICS: PIPELINES EN MACHINE LEARNING

Un aspecto clave del desarrollo de modelos de Machine Learning es el feedback continuo y la alimentación del modelo con conocimiento del negocio, que permita generar información rica para poder desarrollar sobre ésta un buen modelo (*garbage in – garbage out*).



## 02. Algoritmos y tareas

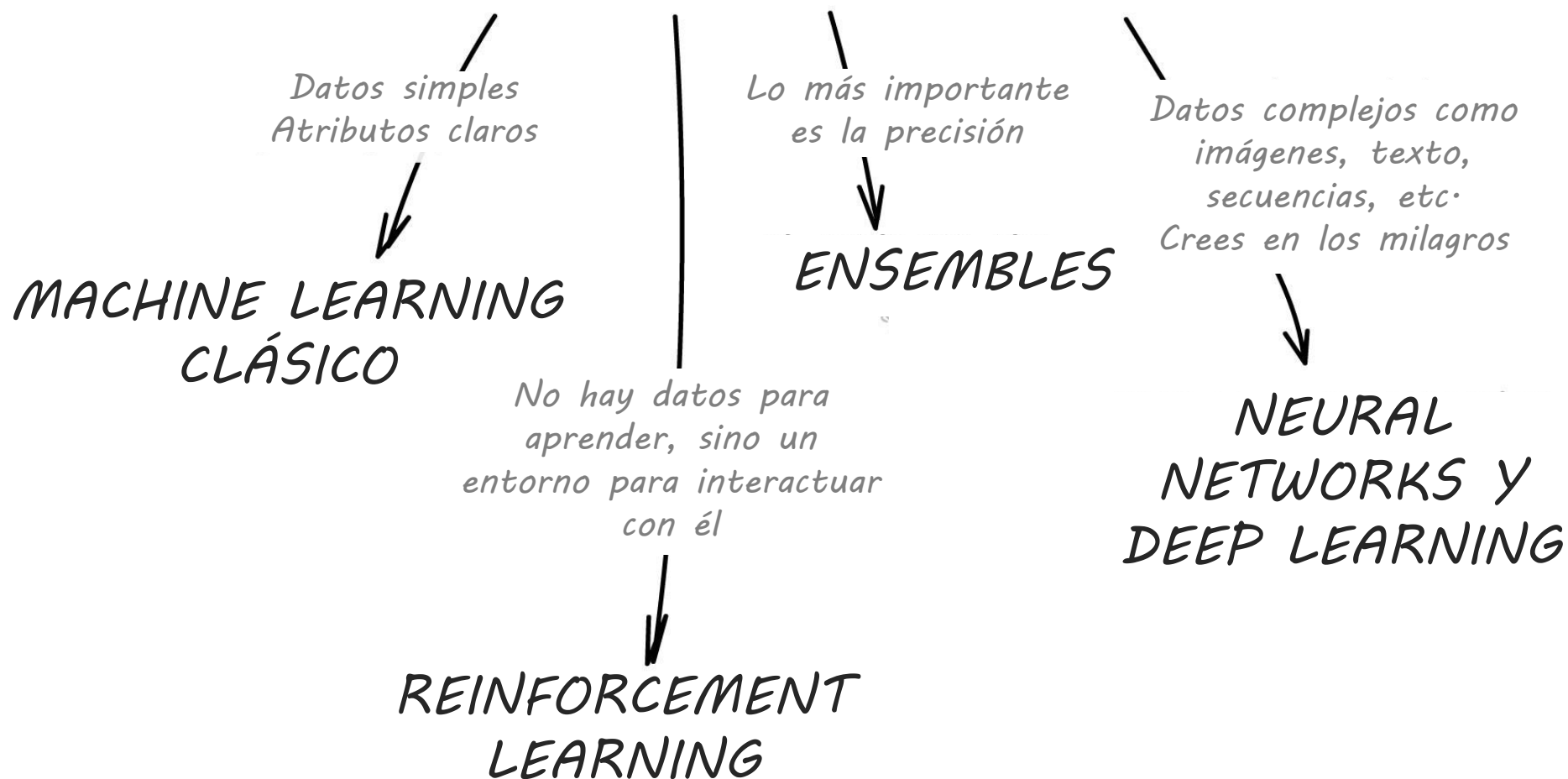




## 2 **ALGORITMOS Y TAREAS:** TIPOS DE ALGORITMOS EN ML

---

## LOS PRINCIPALES TIPOS DE ALGORITMOS EN ML

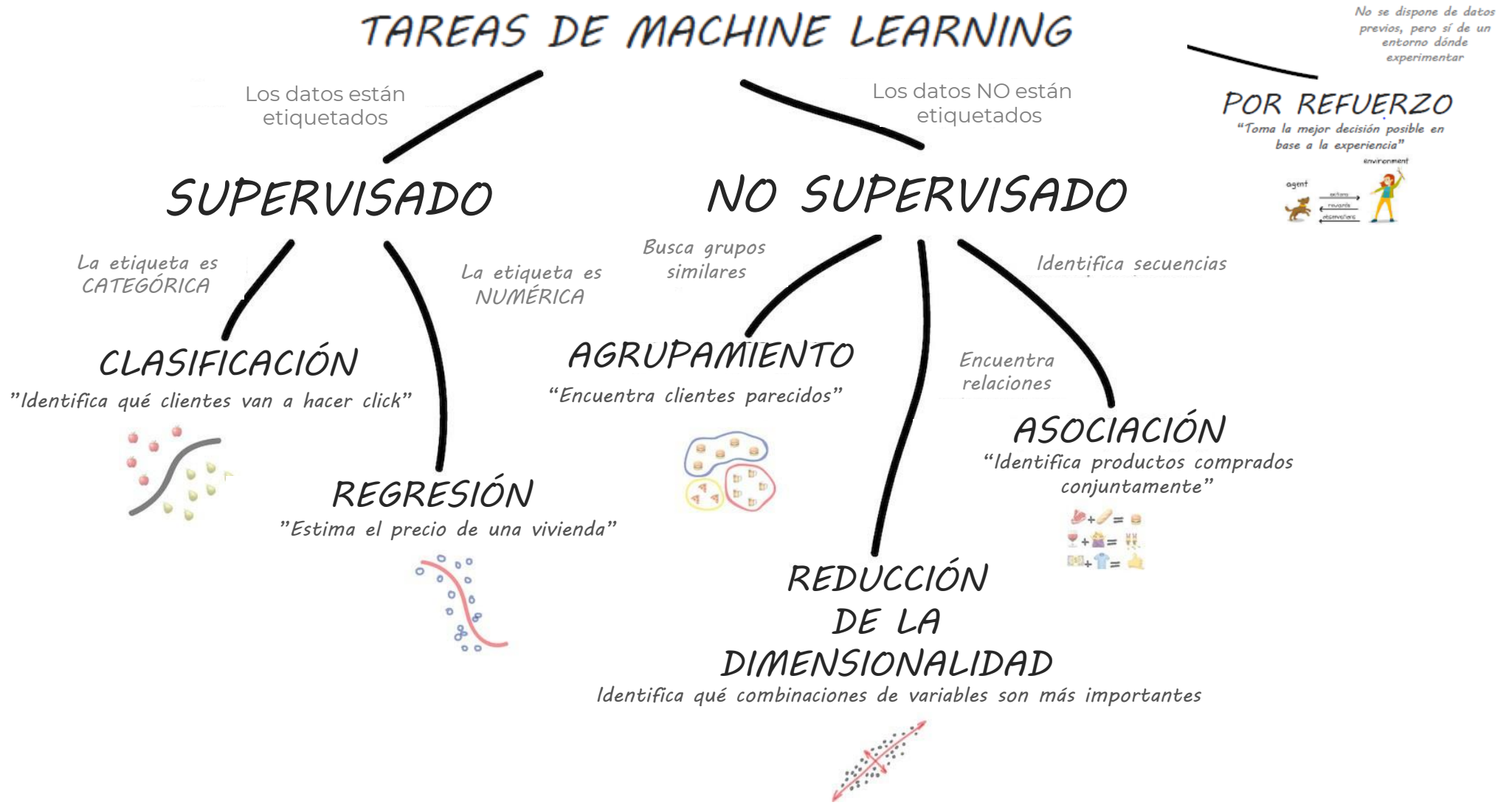


2

## ALGORITMOS Y TAREAS: TIPOS DE TAREAS EN ML

2

# ALGORITMOS Y TAREAS: TIPOS DE TAREAS EN ML



2

ALGORITMOS Y TAREAS: FAMILIAS DE ALGORITMOS EN ML

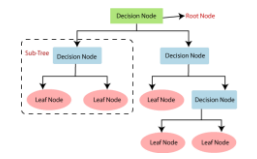
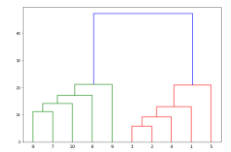
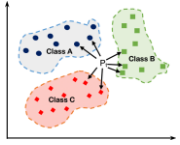
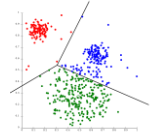
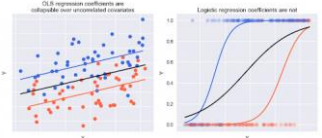
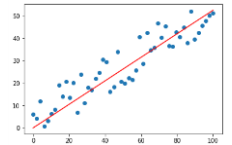
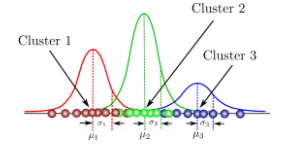
-  Simbolistas

MÉTODOS DE REGLAS
-  Analogistas

MÉTODOS DE VECINDAD
-  Formales

MÉTODOS GEOMÉTRICOS
-  Bayesianos

MÉTODOS DE PROBABILIDAD

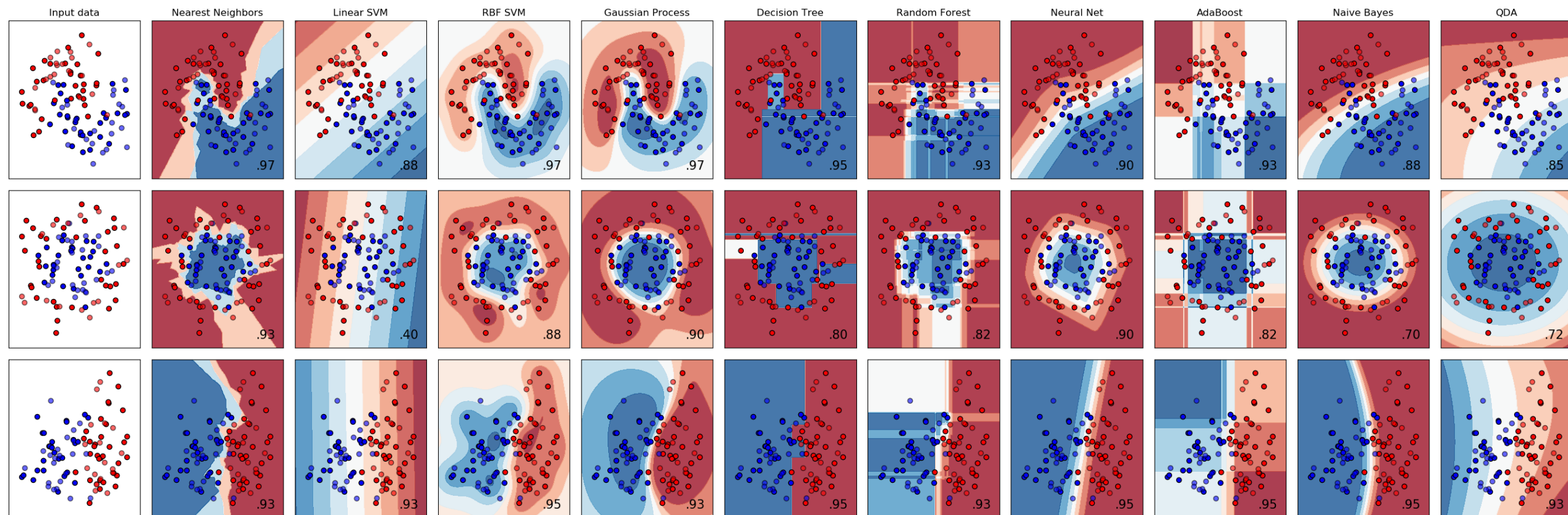
CLASIFICACIÓN	REGRESIÓN	AGRUPAMIENTO
<div><p>CLASSIFICATION AND REGRESSION TREES</p></div>		<div><p>HIIERARCHICAL CLUSTERING</p></div>
<div><p>K-NEAREST NEIGHBORS</p></div>		<div><p>K-MEANS / K-MODES / DBSCAN</p></div>
<div><p>LOGISTIC REGRESSION / SVM</p></div>	<div><p>LINEAR REGRESSION</p></div>	
<div><div><div>Likelihood</div><div>Class Prior Probability</div><div><math display="block">P(c x) = \frac{P(x c)P(c)}{P(x)}</math></div><div>Posterior Probability</div><div>Predictor Prior Probability</div></div><p>NAÏVE BAYES</p></div>		<div><p>GAUSSIAN MIXTURES</p></div>



# 2

## ALGORITMOS Y TAREAS: FAMILIAS DE ALGORITMOS EN ML

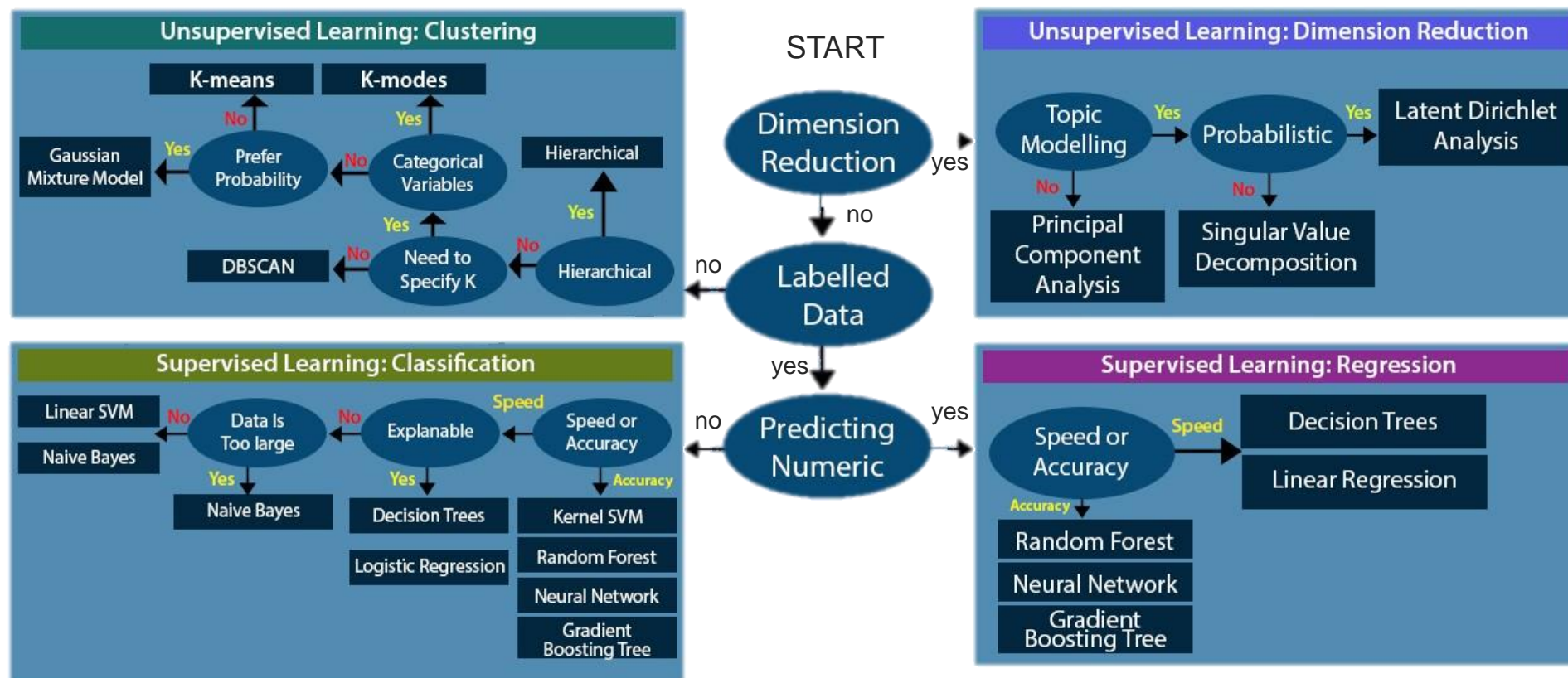
En Machine Learning existe un teorema llamado “**No Free Lunch**”, que dice que no hay un algoritmo universal que se ajuste mejor que el resto a cualquier tipo de problema, y que el rendimiento del mismo dependerá, sobre todo, de las características del conjunto de datos (dataset) y de nuestras necesidades concretas.



# 2

## ALGORITMOS Y TAREAS: SELECCIÓN DE ALGORITMOS EN ML

Aunque existen principios y propiedades de cada tipo de algoritmo que permitirán decidir cuál es el mejor algoritmo, es habitual recorrer a la aplicación de distintos algoritmos y comparación de resultados para seleccionar el modelo definitivo.

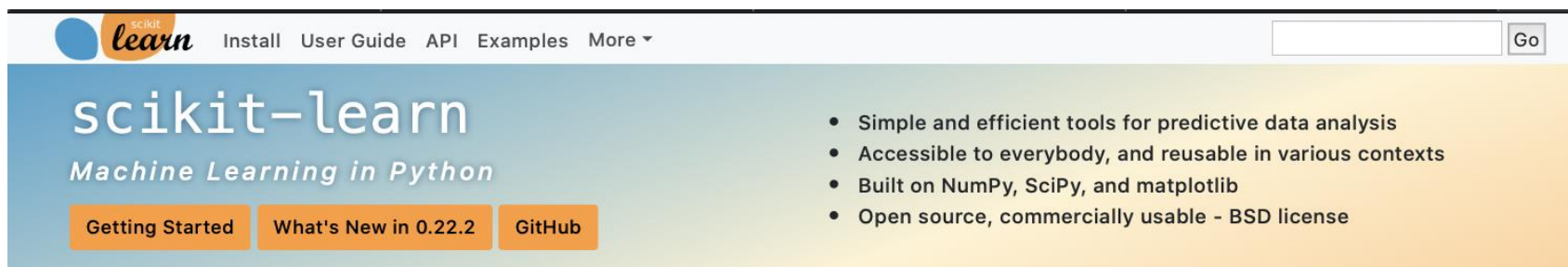


## 03. SCIKIT - LEARN



# 3 SCIKIT - LEARN: BASICS

Scikit-learn es una librería de Python desarrollada inicialmente por David Cournapeau como proyecto de verano dentro de Google en 2007. Más tarde, Matthieu Brucher se unió al proyecto y empezó a utilizarlo como parte de su tesis. En 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort y Vincent Michel de INRIA tomaron el liderazgo del proyecto y el 1 de febrero de 2010 hicieron el primer lanzamiento público. Desde entonces, han aparecido varios lanzamientos con un ciclo de aproximadamente 3 meses. Desde entonces, la comunidad internacional ha estado liderando el desarrollo. El sponsorship actual de scikit-learn incluye a INRIA, Google y la Python Software Foundation.



Scikit-learn proporciona una amplia gama de algoritmos de aprendizaje supervisado y no supervisado a través de una interfaz consistente en Python (programación orientada a objetos). Scikit-learn se licencia bajo una licencia BSD simplificada permisiva y se distribuye bajo muchas distribuciones de Linux, fomentando el uso académico y comercial. La biblioteca está construida sobre diversas librerías de Python, entre las que se incluyen Numpy, SciPy, Matplotlib, Sympy o pandas. Las extensiones o módulos para SciPy se denominan convencionalmente SciKits. Como tal, el módulo completo basado en algoritmos de aprendizaje se llama scikit-learn. La visión de la biblioteca es que sea suficientemente robusta para su uso en sistemas de producción. Esto significa que scikit-learn está construida con un enfoque en la facilidad de uso, la calidad del código, la colaboración, la documentación y el rendimiento. Aunque la interfaz es Python, la mayoría de funciones de fondo de scikit-learn son C para mejorar el rendimiento.

# 3 SCIKIT - LEARN: FEATURES

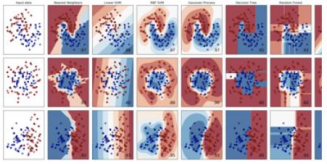
1

### Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, and more...



Examples

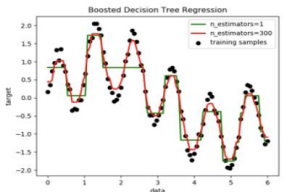
2

### Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, nearest neighbors, random forest, and more...



Examples

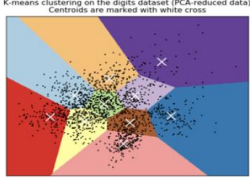
3

### Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, and more...



Examples

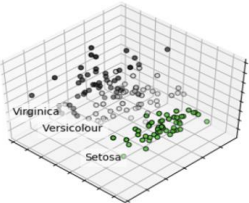
2

### Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** k-Means, feature selection, non-negative matrix factorization, and more...



Examples

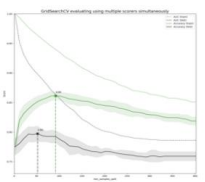
3

### Model selection

Comparing, validating and choosing parameters and models.

**Applications:** Improved accuracy via parameter tuning

**Algorithms:** grid search, cross validation, metrics, and more...



Examples

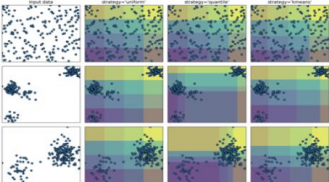
3

### Preprocessing

Feature extraction and normalization.

**Applications:** Transforming input data such as text for use with machine learning algorithms.

**Algorithms:** preprocessing, feature extraction, and more...



Examples

- 1 Scikits de Aprendizaje
- 2 Scikits de Validación (estrategia y métricas)
- 3 Scikits de Preprocesamiento



# 04. EL MACHINE LEARNING CANVAS



4

ML CANVAS: BASICS

Proyecto:

Fecha:

Versión:

<p>NUEVOS DATOS Y REENTRENAMIENTO</p> <p>¿Cómo reentrenaremos el modelo? ¿Cómo nos llegarán los nuevos datos? ¿Necesitaremos tratarlos on-line? ¿Y cruzarlos? ¿Cada cuanto reentrenaremos el modelo? ¿Qué SLA de entrenamiento tenemos?</p>	<p>PREDICCIÓN (ON / OF)</p> <p>¿Cómo realizaremos las predicciones? ¿Realizaremos una predicción en batch u on-line? ¿Cómo serviremos los resultados de la predicción? ¿Cuál es el SLA para realizar la predicción?</p>	<p>PROPUESTA DE VALOR</p> <p>Definición informal del problema (en términos de negocio) ¿Cuál es el problema? ¿Cuáles so las motivaciones para resolverlo? ¿A qué objetivos sirve el modelo?</p>	<p>ORÍGENES DE DATOS</p> <p>¿De qué datos disponemos? ¿Qué orígenes de datos (internos y externos) disponemos? ¿Qué formatos tienen estos datos? ¿Cómo los integraremos? <u>Comprobación de obligaciones legales y autorizaciones</u></p>	<p>TAREA DE ML</p> <p>¿A qué tipo de tarea corresponde el problema? Clasificación Supervisada / No Supervisada / Por refuerzo ¿Es un problema a resolver con Deep Learning?</p>
<p>EVALUACIÓN EN SERVICIO Y ALM.</p> <p>¿Cómo mediremos el rendimiento del modelo en servicio? ¿Qué métrica utilizaremos para analizar el rendimiento del modelo una vez esté en servicio? ¿Qué métodos utilizaremos para obtener estas métricas después del deployment? ¿La degradación del modelo dará lugar a alguna política de reentrenamiento?</p>	<p>MÉTRICA DE EVALUACIÓN (EN DESARROLLO)</p> <p>¿Con qué métrica mediremos el rendimiento del modelo en desarrollo? ¿Sobre qué base realizaremos la medición de la métrica? ¿Hay un valor mínimo esperado?</p>	<p>DEFINICIÓN DEL PROBLEMA</p>	<p>ATRIBUTOS</p> <p>¿Con qué tipo de atributos alimentaremos el modelo? Listado de variables disponibles y accesibles; Listado de atributos no disponibles que se pueden construir, y estimación del coste; Listado de variables deseables que no se pueden construir (para futuro)</p>	<p>DEFINICIÓN DEL PERÍMETRO Y TARGET (SOLO EN CS)</p> <p>¿Con qué población entrenaremos el modelo? ¿Cuál es la población disponible? ¿Hay alguna característica específica que la defina? En caso de que sea un problema de Clasificación Supervisada, ¿Cómo definiremos el target? ¿Toda la población es elegible para el target? ¿Target y resto de la población cumplen las mismas condiciones de perímetro?</p>
<p>VALIDACIÓN Y DEPLOYMENT</p>		<p>USO DEL MODELO, TOMA DE DECISIONES Y EXPLICABILIDAD</p> <p>¿Cómo se va a utilizar el modelo? ¿El modelo servirá para que otros tomen decisiones asistidas? ¿Será prescriptivo? ¿Es necesario que la persona que toma las decisiones comprenda el funcionamiento? ¿Cómo le entregaremos los resultados?</p>		

4

# ML CANVAS: BASICS

Proyecto:

Fecha:

Versión:

<b>NUEVOS DATOS Y REENTRENAMIENTO</b>  ¿Cómo reentrenaremos el modelo? ¿Cómo nos llegarán los nuevos datos? ¿Necesitaremos tratarlos on-line? ¿Y cruzarlos? ¿Cada cuanto reentrenaremos el modelo? ¿Qué SLA de entrenamiento tenemos?	<b>PREDICCIÓN (ON / OF)</b>  ¿Cómo realizaremos las predicciones? ¿Realizaremos una predicción en batch u on-line? ¿Cómo serviremos los resultados de la predicción? ¿Cuál es el SLA para realizar la predicción?	<b>PROPUESTA DE VALOR</b>  Definición informal del problema (en términos de negocio) ¿Cuál es el problema? ¿Cuáles so las motivaciones para resolverlo? ¿A qué objetivos sirve el modelo?	<b>ORÍGENES DE DATOS</b>  ¿De qué datos disponemos? ¿Qué orígenes de datos (internos y externos) disponemos? ¿Qué formatos tienen estos datos? ¿Cómo los integraremos? <u>Comprobación de obligaciones legales y autorizaciones</u>	<b>TAREA DE ML</b>  ¿A qué tipo de tarea corresponde el problema? Clasificación Supervisada / No Supervisada / Por refuerzo ¿Es un problema a resolver con Deep Learning?
<b>EVALUACIÓN EN SERVICIO Y ALM</b>  ¿Cómo mediremos el rendimiento del modelo en servicio? ¿Qué métrica utilizaremos para analizar el rendimiento del modelo una vez esté en servicio? ¿Qué métodos utilizaremos para obtener estas métricas después del deployment? ¿La degradación del modelo dará lugar a alguna política de reentrenamiento?	<b>MÉTRICA DE EVALUACIÓN (EN DESARROLLO)</b>  ¿Con qué métrica mediremos el rendimiento del modelo en desarrollo? ¿Sobre qué base realizaremos la medición de la métrica? ¿Hay un valor mínimo esperado?		<b>ATRIBUTOS</b>  ¿Con qué tipo de atributos alimentaremos el modelo? Listado de variables disponibles y accesibles; Listado de variables no disponibles que se pueden construir, y estimación del coste; Listado de variables deseables que no se pueden construir (para futuro)	<b>DEFINICIÓN DEL PERÍMETRO Y TARGET (SÓLO EN CS)</b>  ¿Con qué población entrenaremos el modelo? ¿Cuál es la población disponible? ¿Hay alguna característica específica que la defina? En caso de que sea un problema de Clasificación Supervisada, ¿Cómo definiremos el target? ¿Toda la población es elegible para el target? ¿Target y resto de la población cumplen las mismas condiciones de perímetro?
<b>USO DEL MODELO, TOMA DE DECISIONES Y EXPLICABILIDAD</b>  ¿Cómo se va a utilizar el modelo? ¿El modelo servirá para que otros tomen decisiones asistidas? ¿Será prescriptivo? ¿Es necesario que la persona que toma las decisiones comprenda el funcionamiento? ¿Cómo le entregaremos los resultados?				

4

# ML CANVAS: BASICS

Proyecto:

Fecha:

Versión:

NUEVOS DATOS Y REENTRENAMIENTO	PREDICCIÓN (ON / OF)	PROPUESTA DE VALOR	ORÍGENES DE DATOS	TAREA DE ML
EVALUACIÓN EN SERVICIO Y ALM	MÉTRICA DE EVALUACIÓN (EN DESARROLLO)		ATRIBUTOS	DEFINICIÓN DEL PERÍMETRO Y TARGET (SÓLO EN CS)
	USO DEL MODELO, TOMA DE DECISIONES Y EXPLICABILIDAD			

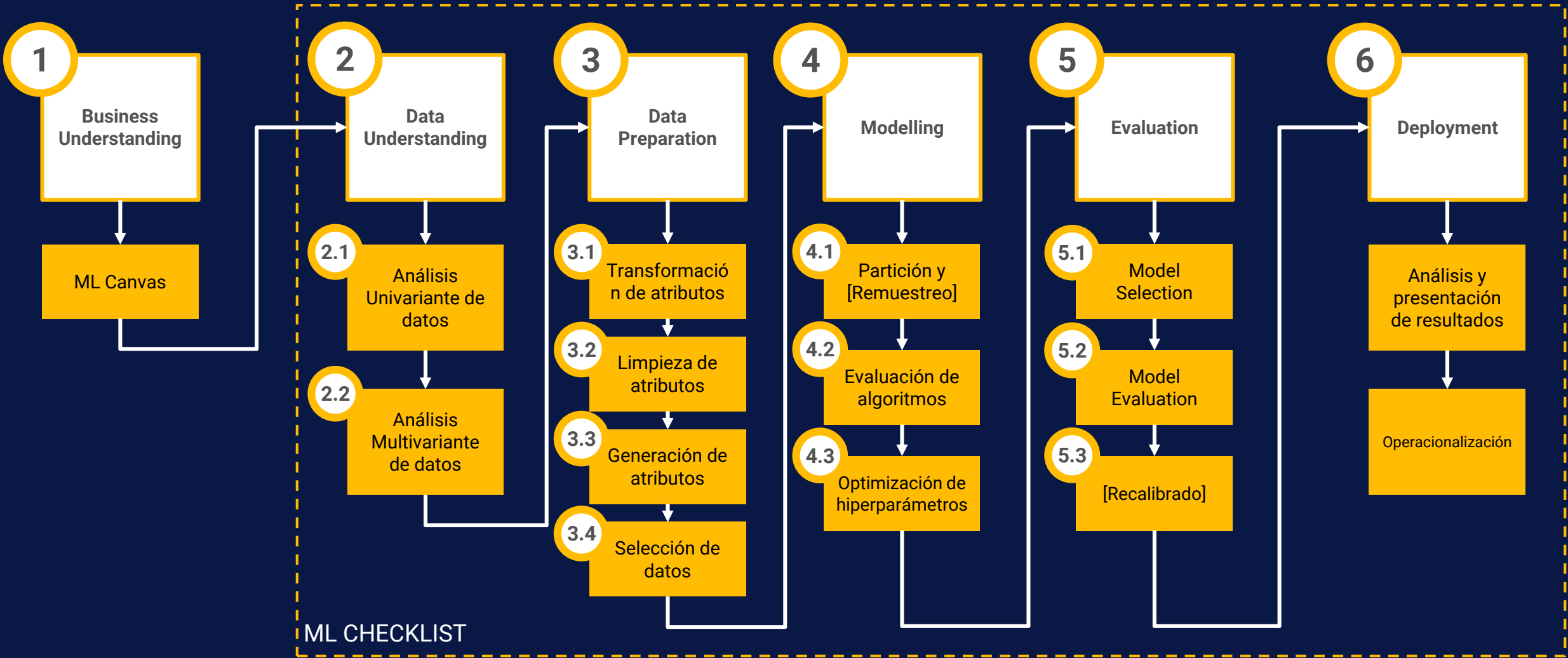
# 05. LA MACHINE LEARNING CHECKLIST





5

# ML CHECKLIST: PIPELINES



5

# ML CHECKLIST : DATA UNDERSTANDING

2.1

## Análisis univariante de los datos

1. Tamaño del Dataset (tamaño en memoria, número de registros y atributos, etc.)
2. Visualización directa de los datos (head)
3. Tipo de atributos disponibles (numéricos, categóricos)
4. Estadísticos descriptivos (valores medios, dispersión, percentiles, etc.)
5. Número de valores nulos
6. Distribución / rango de valores del target (sólo en clasificación supervisada)
7. Identificación de outliers
8. Identificación de datos erróneos
9. Correlación de variables con el target
10. Correlación de las variables con la clase (unidimensional)
11. Visualización gráfica de las distribuciones
  - ✓ Numéricas: histogramas, box-plots, violin-plots, vista por deciles, etc.
  - ✓ Categóricas: bar-charts, conteo directo, etc.

2.2

## Análisis multi-variante de los datos

1. Distribución de variables 2 a 2 (scatter-plots)
2. Correlación de las variables 2 a 2 (correlación lineal)
3. Cross-tabs
4. Correlación de combinaciones de variables con la clase

# 5 ML CHECKLIST : DATA PREPARATION

## 3.1

### Limpieza de atributos

1. Ajuste de tipos
2. Imputación de valores nulos (cero, media, moda, valor fijo, etc.) o eliminación de registros que los contengan
3. Corrección de valores atípicos o eliminación de registros que los contengan
4. Eliminación de atributos de baja varianza o con elevada correlación con otros

## 3.2

### Transformación de atributos

#### ✓ Variables categóricas

1. One-Hot Encoding
2. Label Encoding / Tipado
3. Agrupación de valores

#### ✓ Variables numéricas

1. Escalado (min-max, estandarización, etc.)
2. Categorización de variables numéricas
3. Redondeo

#### ✓ Variables de fecha

1. Conversión a tiempo (num)
2. Agregación
3. Diferencias

#### ✓ Variables textuales

1. Numerización de variables
2. Extracción de patrones
3. Palabras clave

## 3.3

### Generación de atributos

1. Generación de atributos por combinación de variables
2. “Añade una variable aleatoria al modelo”

## 3.4

### Selección de datos

1. Muestreo de registros (guiado, aleatorio o estratificado)
2. Selección de atributos / Reducción de dimensionalidad

5

# ML CHECKLIST : MODELLING/EVALUATION

4.1

## Partición y muestreo del dataset

- ✓ Definición de estrategia de validación (Random Holdout, k-fold, Bootstrap, etc.)
- ✓ Definición de la política de partición (aleatorio o guiado)
- ✓ Definición de estrategias de remuestreo (mantener siempre la validación intacta)
- ✓ Definición de la métrica de evaluación (sólo una métrica)

4.2

## Evaluación de algoritmos

1. Definición del modelo de base (Keep-It-Simple)
2. Short-list de algoritmos de distintos tipos

4.3

## Optimización de modelos

1. Optimización de hiperparámetros
2. Algoritmos de ensemble

5.1

## Model Selection & Evaluation

1. Selección del modelo con mejor métrica en test
2. Comprobación de rendimiento en validación (capacidad de generalización)

5.2

## Recalibrado

1. En caso de que sea necesario, recalibrar los scorings para obtener la probabilidad real

# 5

## ML CHECKLIST : DEPLOYMENT

### 6.1

#### Análisis y presentación de resultados

1. Presentación de la visión global del problema y contextualización
2. Representación de variables más significativas
3. Análisis de métricas o curvas de rendimiento
4. Análisis de tipos de errores más habituales
5. Representación directa de reglas del modelo o meta-modelos tipo rule-fit o Shapley Values
6. Análisis unidimensional de variables y/o reglas

### 6.2

#### Operacionalización del modelo

1. Documentación de código
2. Adaptación del código para entorno productivo
3. Preparación de los sistemas para la operacionalización (paquetería, etc.)
4. Preparación del sistema de monitorización y alertas de caída de rendimiento



# ¡MUCHAS GRACIAS!

Guillem Sitges i Puy

---

Sesión 18