

# Inteligencia Artificial

Juan Pablo Restrepo Uribe

Ing. Biomedico - MSc. Automatización y Control Industrial

[jprestrepo@correo.iue.edu.co](mailto:jprestrepo@correo.iue.edu.co)

2023

Institución Universitaria de Envigado

## Tipos de Inteligencia Artificial

Supervisado	No Supervisado
Los ejemplos utilizados para el entrenamiento incluyen la solución esperada o las etiquetas	La información de entrenamiento se encuentra sin marcar. Y aprende sin la necesidad de un “profesor”

# Aplicaciones de Inteligencia Artificial

## Visión por Computadora

**Classification**



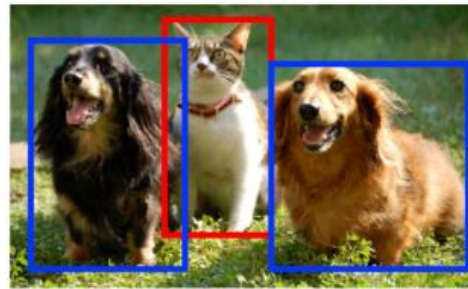
CAT

**Classification  
+ Localization**



CAT

**Object Detection**



CAT, DOG

**Instance Segmentation**



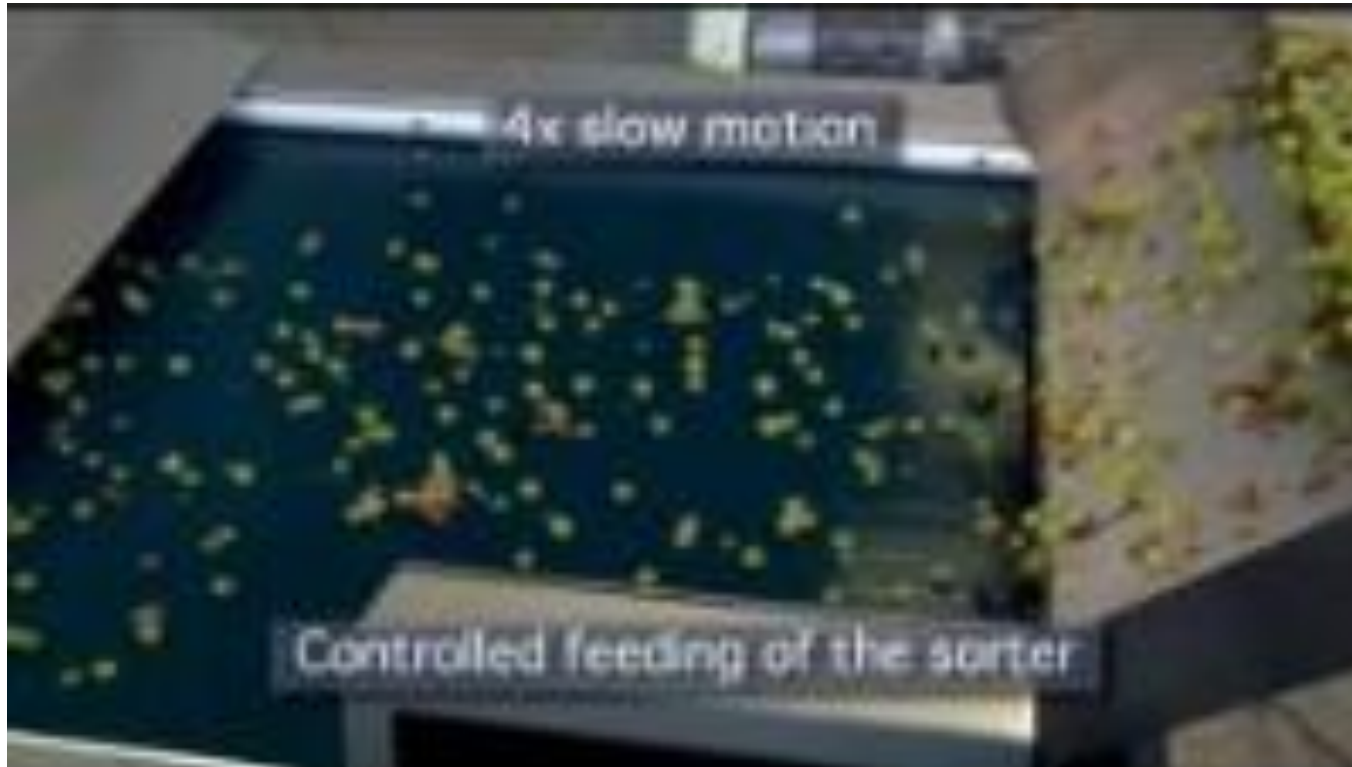
CAT, DOG

Single object

Multiple objects

# Aplicaciones de Inteligencia Artificial

## Visión por Computadora



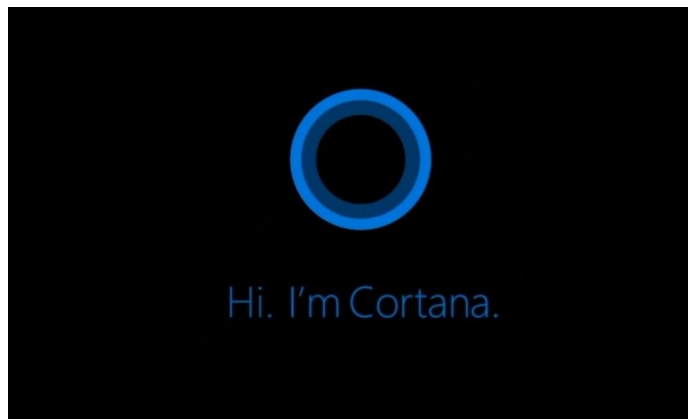
# Aplicaciones de Inteligencia Artificial

## Procesamiento del Lenguaje Natural



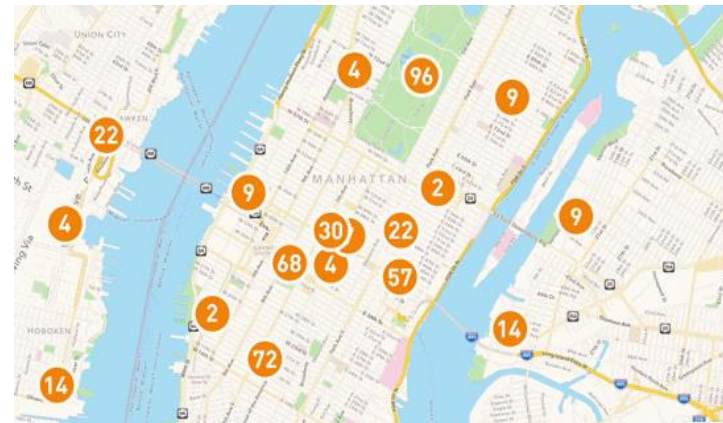
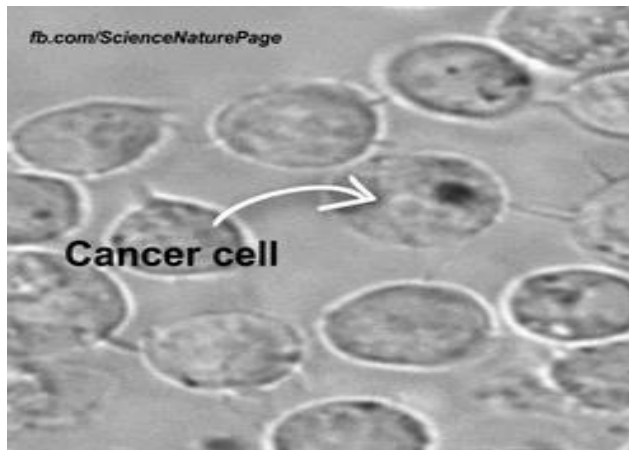
# Aplicaciones de Inteligencia Artificial

## Reconocimiento de voz



# Aplicaciones de Inteligencia Artificial

## Sistemas Expertos



**NETFLIX**



# Principales Retos en Inteligencia Artificial

## Datos de entrenamiento insuficientes

El proceso de aprendizaje en IA, involucra una gran cantidad de datos para funcionar de forma adecuada. Incluso para algunos problemas es necesario tener miles o millones de ejemplos para el conjunto de entrenamiento.



# Principales Retos en Inteligencia Artificial

## Datos de entrenamiento insuficientes

El proceso de aprendizaje en IA, involucra una gran cantidad de datos para funcionar de forma adecuada. Incluso para algunos problemas es necesario tener miles o millones de ejemplos para el conjunto de entrenamiento.

Un artículo\* publicado en 2001 por Michele Banko y Erick Brill (Investigadores de Microsoft) demostró que el comportamiento de diferentes modelos de aprendizaje de máquina era muy similar incluso para modelos menos complejos, cuando se proporcionaba la información suficiente para un procesamiento de lenguaje natural..

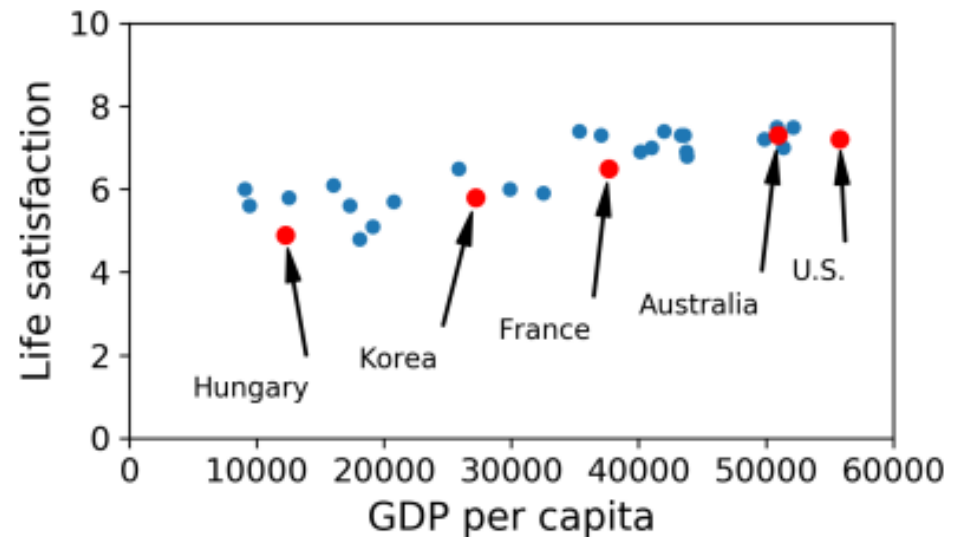
\*Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL '01). Association for Computational Linguistics, USA, 26–33. DOI:<https://doi.org/10.3115/1073012.1073017>

# Principales Retos en Inteligencia Artificial

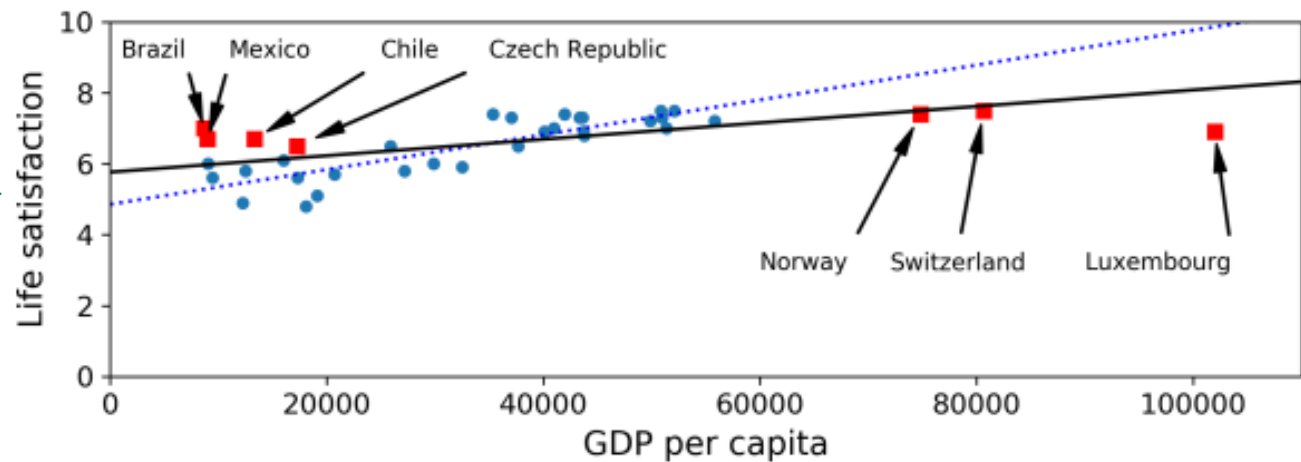
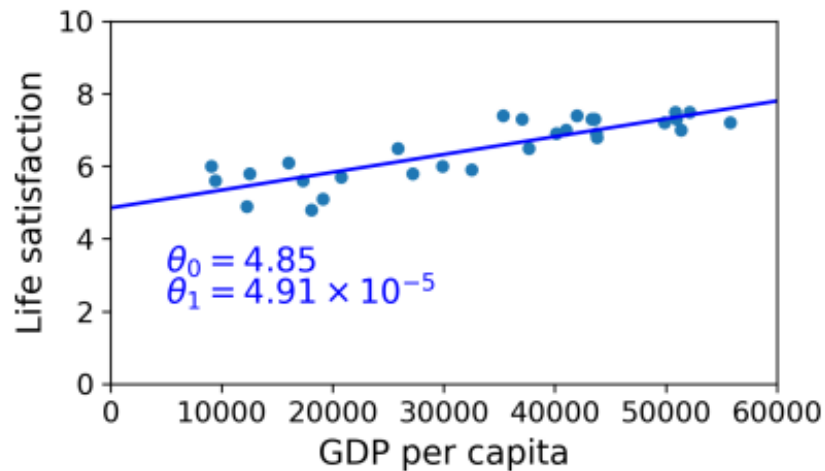
## Conjuntos de datos no representativos

Por ejemplo: supongamos que deseamos establecer el nivel de felicidad de los habitantes de un país basándose en los ingresos mensuales de los mismos.

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2



## Conjuntos de datos no representativos



# Principales Retos en Inteligencia Artificial

## Datos de baja calidad

Supongamos que se posee una cantidad de datos de entrenamiento llenos de ruido, errores de adquisición u otros problemas. El proceso de aprendizaje se dificultará debido a que las características relevantes no son de fácil acceso por lo que el sistema será menos preciso.

- En algunos casos la calidad de los datos no es lo suficiente buena para el proceso por lo que deben procesarse o eliminarse según sea el caso.
- En otras ocasiones puede que los datos tengan buena calidad (sin ruido), pero falta alguna información

# Principales Retos en Inteligencia Artificial

## Características Irrelevantes

“Si entra basura, sale basura” (GIGO). Si la calidad de lo que ingresa no es buena el resultado normalmente tampoco lo es.

Un sistema solo puede aprender de forma exitosa si se alimenta con características relevantes. El proceso de selección de características resulta entonces en uno de los mas críticos dentro del proceso de aprendizaje.

# Principales Retos en Inteligencia Artificial

## Características Irrelevantes

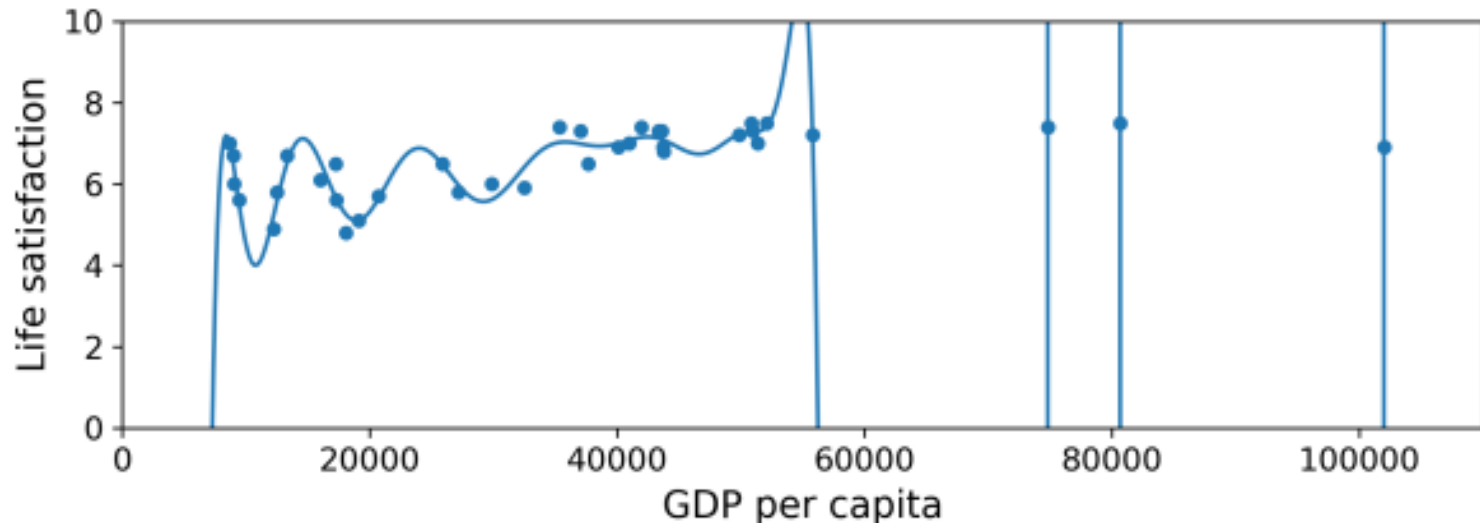
El proceso de escoger o adquirir un buen conjunto de características se conoce como ingeniería de características, e involucra:

- Selección de características
- Extracción de características
- Creación de nuevas características

# Principales Retos en Inteligencia Artificial

## Overfitting del conjunto de aprendizaje

Es similar al proceso de sobre generalizar las cosas, y se traduce en que el sistema tiene muy buen funcionamiento en el conjunto de aprendizaje, pero no con datos diferentes al de aprendizaje.



# Principales Retos en Inteligencia Artificial

## *Underfitting* del conjunto de aprendizaje

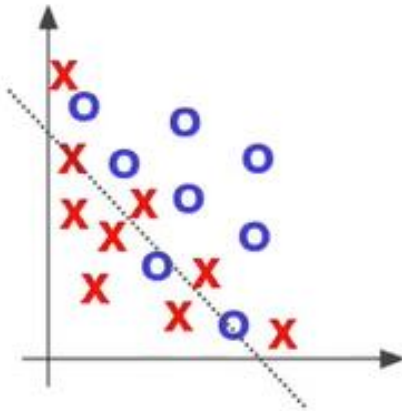
Es exactamente lo opuesto que el **overfitting**, ocurre cuando el modelo es muy simple para aprender la estructura verdadera de los datos. Las principales opciones para resolver este problema son:

- Seleccionar un modelo mas poderoso con mas parámetros
- “Alimentar” el sistemas con mejores características
- Reducir las restricciones del sistema,

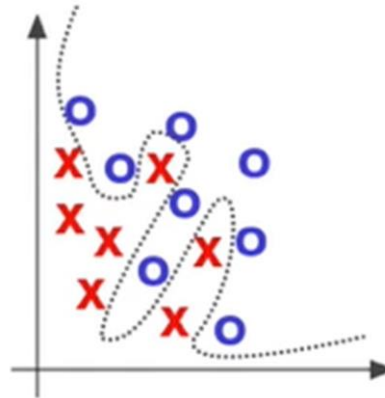


# Principales Retos en Inteligencia Artificial

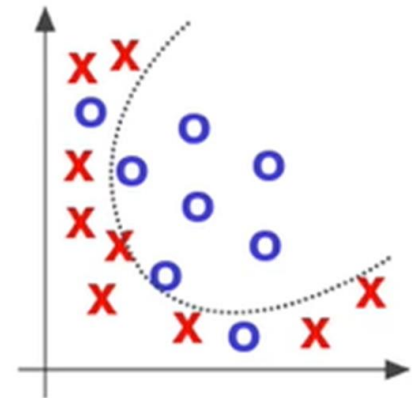
Underfitting



Overfitting



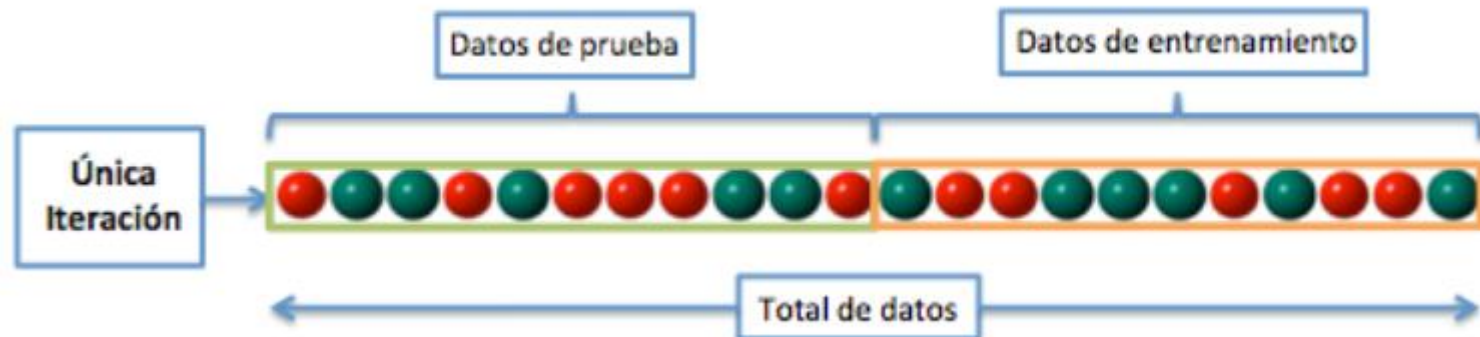
Modelo Apropriado.



# Principales Retos en Inteligencia Artificial

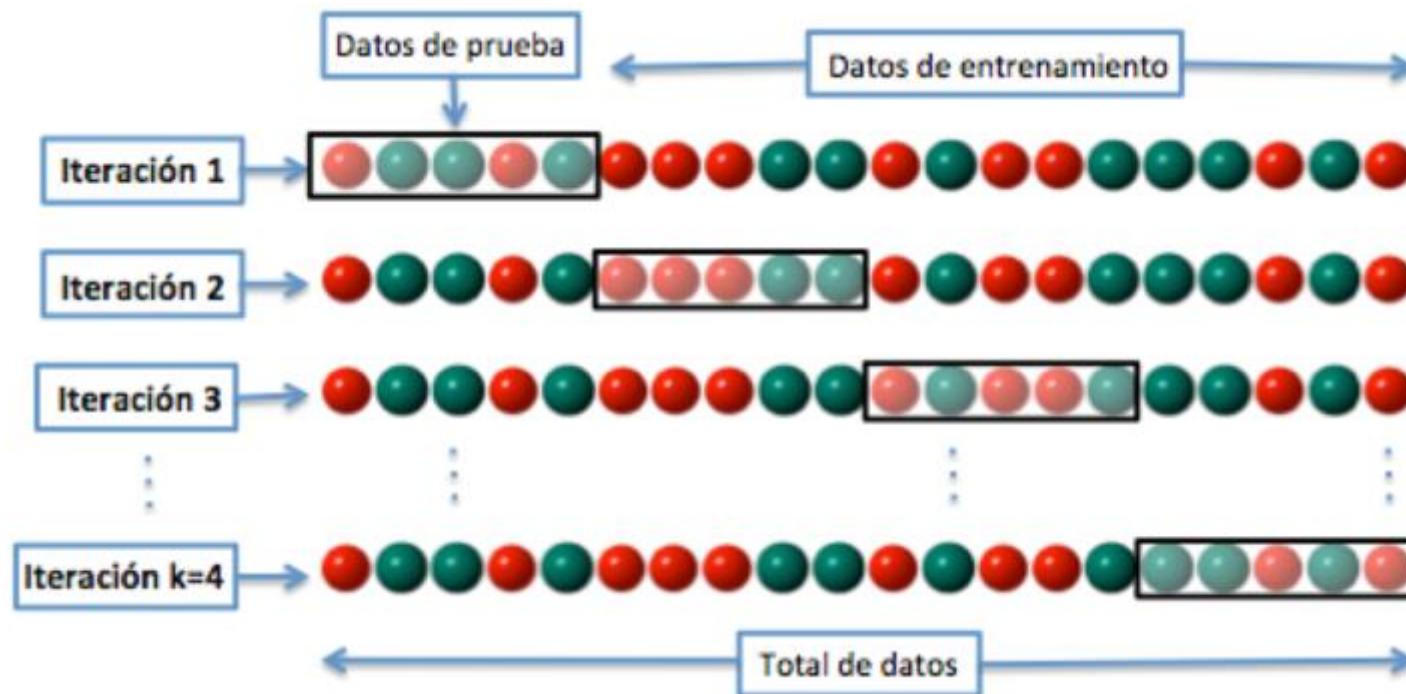
## *Evaluación y Validación*

La única forma de saber que también puede un modelo generalizar nuevos casos, es probando con nuevos casos. Una buena opción para resolver esto, es dividir la cantidad de datos disponibles en mínimo 2 conjuntos.



# Principales Retos en Inteligencia Artificial

## *Evaluación y Validación*



# Principales Retos en Inteligencia Artificial

## *Evaluación y Validación*

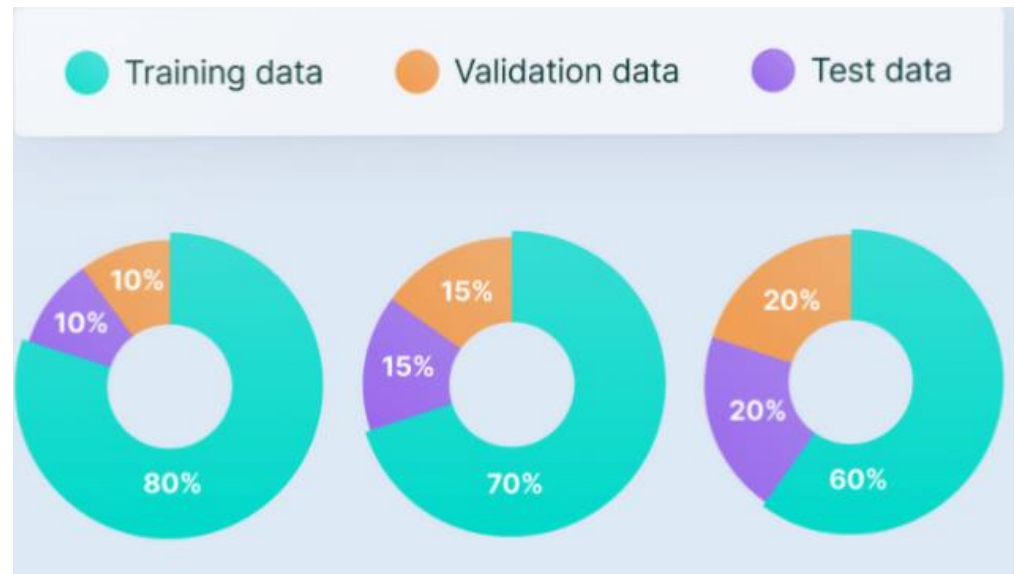
### Splitter Classes

<code>model_selection.GroupKFold([n_splits])</code>	K-fold iterator variant with non-overlapping groups.
<code>model_selection.GroupShuffleSplit(...)</code>	Shuffle-Group(s)-Out cross-validation iterator
<code>model_selection.KFold([n_splits, shuffle, ...])</code>	K-Folds cross-validator
<code>model_selection.LeaveOneGroupOut()</code>	Leave One Group Out cross-validator
<code>model_selection.LeavePGroupsOut(n_groups)</code>	Leave P Group(s) Out cross-validator
<code>model_selection.LeaveOneOut()</code>	Leave-One-Out cross-validator
<code>model_selection.LeavePOut(p)</code>	Leave-P-Out cross-validator
<code>model_selection.PredefinedSplit(test_fold)</code>	Predefined split cross-validator
<code>model_selection.RepeatedKFold(*[, n_splits, ...])</code>	Repeated K-Fold cross validator.
<code>model_selection.RepeatedStratifiedKFold(*[, ...])</code>	Repeated Stratified K-Fold cross validator.
<code>model_selection.ShuffleSplit([n_splits, ...])</code>	Random permutation cross-validator
<code>model_selection.StratifiedKFold([n_splits, ...])</code>	Stratified K-Folds cross-validator.
<code>model_selection.StratifiedShuffleSplit(...)</code>	Stratified ShuffleSplit cross-validator
<code>model_selection.StratifiedGroupKFold(...)</code>	Stratified K-Folds iterator variant with non-overlapping groups.
<code>model_selection.TimeSeriesSplit([n_splits, ...])</code>	Time Series cross-validator

# Principales Retos en Inteligencia Artificial

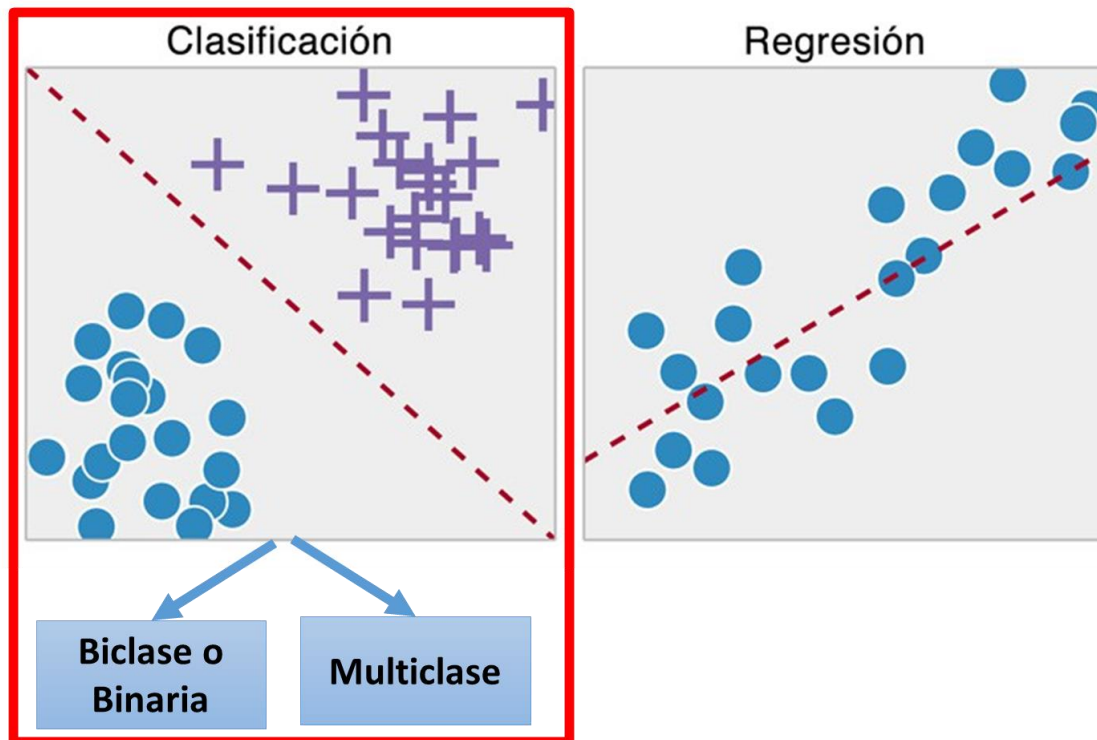
## Ajuste de hiperparametros y selección de modelos

- **Training set:** A set of examples used for learning, that is to fit the parameters of the classifier.
- **Validation set:** A set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network.
- **Test set:** A set of examples used only to assess the performance of a fully-specified classifier.



# Principales Retos en Inteligencia Artificial

## Identificación del problema



# Principales pasos para trabajar IA

## Matriz de características

MUESTRAS	CARACTERISTICAS					ETIQUETAS
	SEXO	HEMATOCRITO	HEMOGLOBINA	FERRITINA	Caracteristica n	
sujeto 1	1	40	14	10	4556	1
sujeto 2	2	36	12,3	230	456	-1
sujeto 3	2	45,9	15,8	150	564	-1
sujeto 4	1	56	14,8	220	654	1
sujeto 5	1	60	16,4	550	456	1
sujeto 6	1	23,2	13,8	15	54	1
sujeto 7	2	50,2	14,1	6	13	-1
.						.
.						.
.						.
sujeto m	1	43,8	12,5	30	645	1

# Principales pasos para trabajar IA

## Normalización

Debido a que las magnitudes entre las características pueden ser muy diferentes, es decir que pueden haber características con rangos entre 0 y 1000 y otras con rangos entre 0 y 10 entonces debemos aplicar una normalización a los datos para mejorar el rendimiento del algoritmo, la ecuación de normalización es la siguiente:

$$\frac{X - X_{min}}{X_{max} - X_{min}}$$



# Principales pasos para trabajar IA

## Evaluación del desempeño (clasificación)

		True class	
		Positive	Negative
Predicted class	Positive	True Positives TP	False Positives FP
	Negative	False Negatives FN	True Negatives TN

# Principales pasos para trabajar IA

## Evaluación del desempeño (clasificación)

$$\textit{exactitud} = \frac{VP + VN}{VP + FP + FN + VN}$$

# Principales pasos para trabajar IA

## Evaluación del desempeño (clasificación)

$$\textit{precision} = \frac{VP}{VP + FP}$$

## Principales pasos para trabajar IA

### Evaluación del desempeño (clasificación)

$$\textit{sensibilidad} = \frac{VP}{VP + FN}$$

## Principales pasos para trabajar IA

### Evaluación del desempeño (clasificación)

$$\textit{especificidad} = \frac{VN}{VN + FP}$$

# Principales pasos para trabajar IA

## Evaluación del desempeño (Regresión)

$$ME = \frac{1}{m} \sum_{t=T-m+1}^T e_t$$

- Es una medida de la desviación promedio de los valores pronosticados respecto a los valores reales.
- Muestra la dirección del error
- Un ME de cero no significa que los pronósticos sean perfectos o que no existan errores de pronóstico, si no que indica que los pronósticos no poseen ningún sesgo, y en consecuencia, están apuntando hacia el objetivo correcto.

# Principales pasos para trabajar IA

## Evaluación del desempeño (Regresión)

$$MSE = \frac{1}{m} \sum_{t=T-m+1}^T (e_t)^2$$

- Los errores positivos y negativos no se compensan entre sí, y en consecuencia, proporciona una idea general del error ocurrido durante el pronóstico.
- El MSE enfatiza el hecho de que el error de pronóstico total está, de hecho, muy afectado por grandes errores individuales, es decir, los errores grandes son mucho más caros que los errores pequeños.

# Principales pasos para trabajar IA

## Evaluación del desempeño (Regresión)

$$RMSE = \sqrt{\frac{1}{m} \sum_{t=T-m+1}^T (e_t)^2}$$



# Principales pasos para trabajar IA

## Evaluación del desempeño (Regresión)

$$MAPE = \frac{1}{m} \sum_{t=T-m+1}^T \left| \frac{e_t}{y_t} \right| \times 100$$

- Mide el porcentaje de error absoluto promedio ocurrido de los pronósticos.
- Es independiente de la escala de medición

## Un poco de código

```
from sklearn.preprocessing import MinMaxScaler  
data = [[-1, 2], [-0.5, 6], [0, 10], [1, 18]]  
scaler = MinMaxScaler()  
(scaler.fit(data))  
(scaler.data_max_)  
(scaler.transform(data))  
(scaler.transform([[2, 2]]))
```

## Un poco de código

```
from sklearn.metrics import confusion_matrix  
y_true = [2, 0, 2, 2, 0, 1]  
y_pred = [0, 0, 2, 2, 0, 2]  
confusion_matrix(y_true, y_pred)
```

## Un poco de código

```
import matplotlib.pyplot as plt
```

```
from sklearn.datasets import make_classification
```

```
from sklearn.metrics import plot_confusion_matrix
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.svm import SVC
```

```
X, y = make_classification(random_state=0)
```

```
X_train, X_test, y_train, y_test = train_test_split( X, y, random_state=0)
```

```
clf = SVC(random_state=0)
```

```
clf.fit(X_train, y_train)
```

```
plot_confusion_matrix(clf, X_test, y_test)
```

```
plt.show()
```