

ACÀMICA

---

# ¡Bienvenidos/as a Data Science!



# Agenda

---

¿Cómo anduvieron?

Repaso: Machine Learning

Explicación: Árboles de Decisión

Hands-On

Break

Explicación: Evaluación de Modelos

Hands-On

Cierre



# ¿Cómo anduvieron?



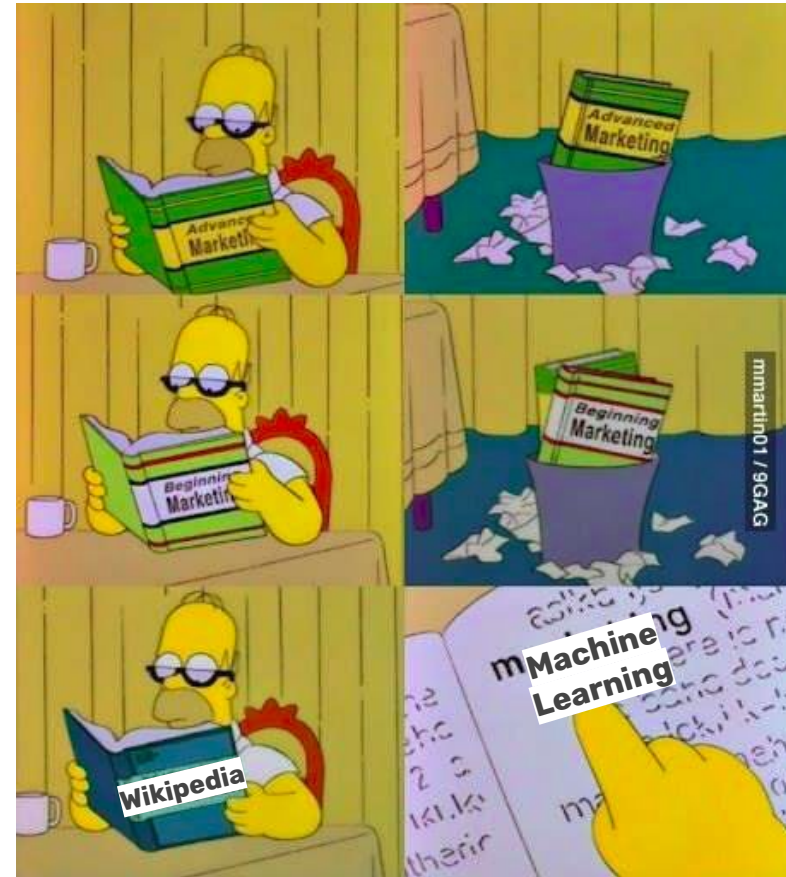
# Repaso: Machine Learning



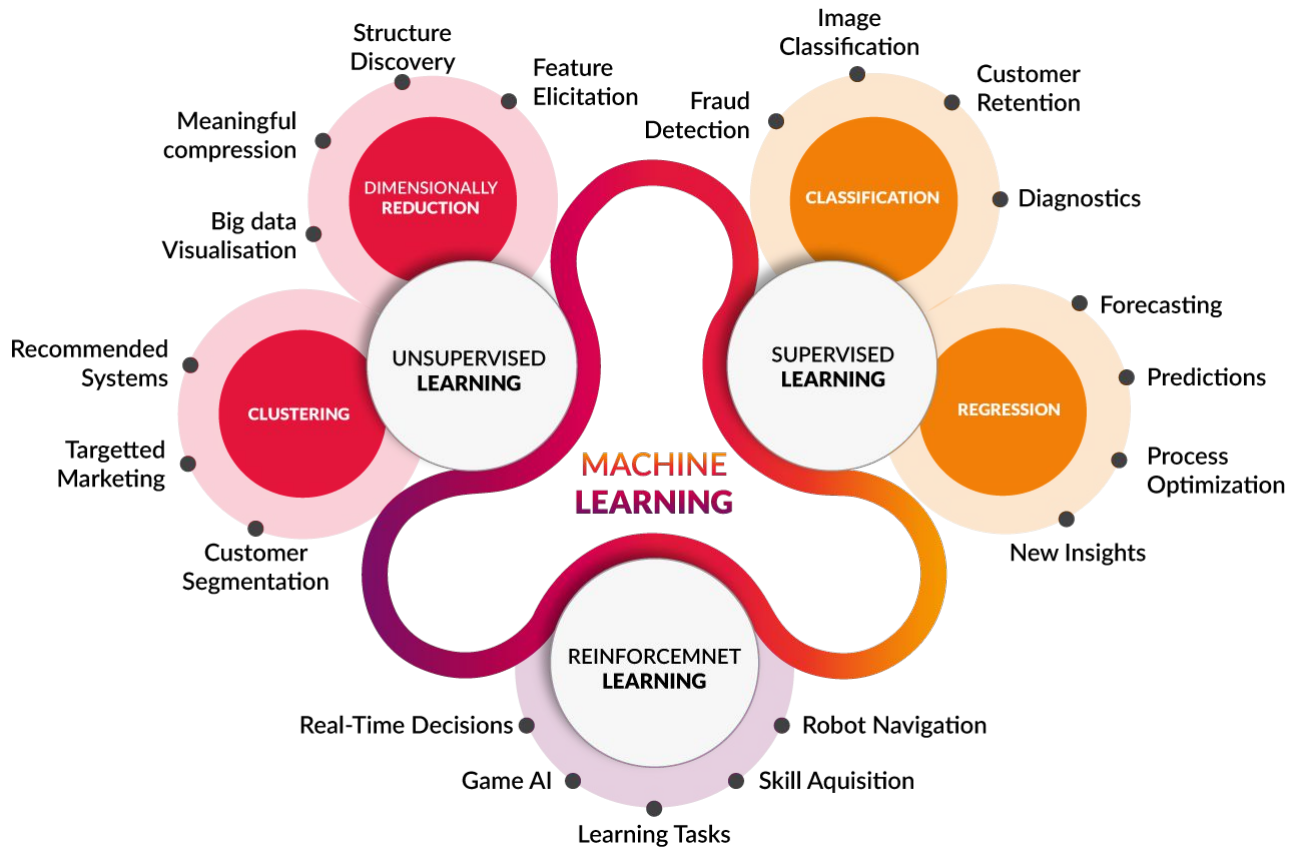
Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.

Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.<sup>[1][2]:2</sup>

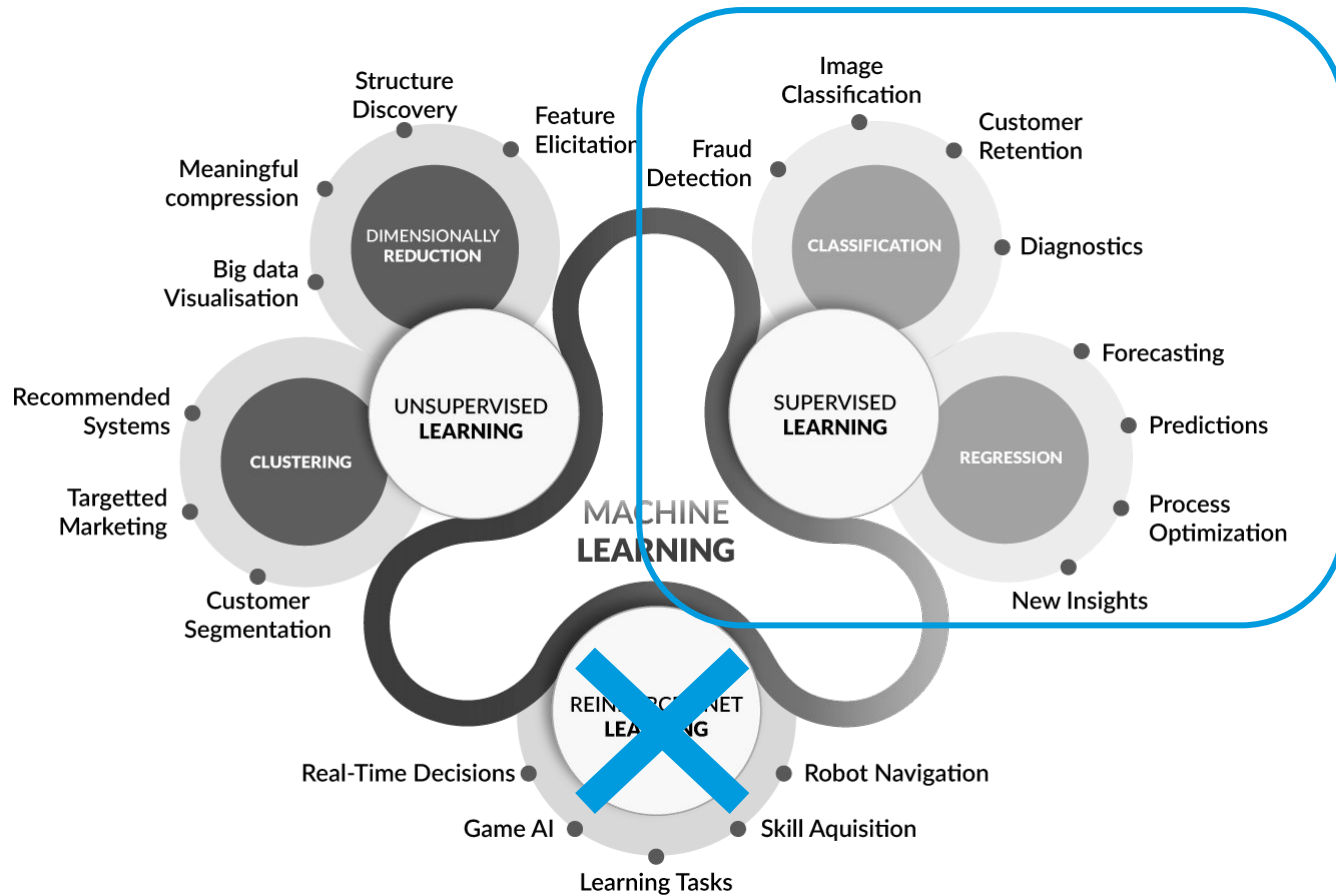
Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.



# Mapa



# Mapa





# Aprendizaje Supervisado

$$f(X) = Y$$

tenemos datos X

tenemos datos Y

# Aprendizaje Supervisado

$$f(X) = Y$$

¿Qué buscamos  
con "f"?

tenemos datos Y

tenemos datos X

# Aprendizaje Supervisado

$$f(X) = Y$$

Un modelo **f** que  
permita determinar  
la salida a partir de la  
entrada

tenemos datos X

tenemos datos Y

# Aprendizaje Supervisado

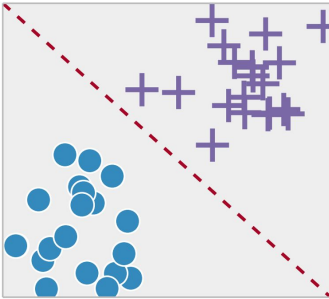
$$f(X) = Y$$



Con este modelo podremos predecir **Y**, para nuevos datos **X** de los cuales no conocamos la salida.

# Aprendizaje Supervisado

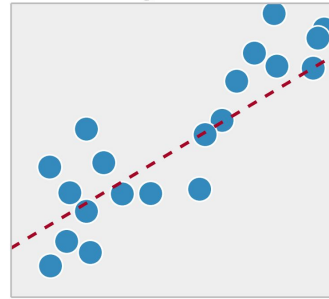
## Clasificación



La variable de salida es una categoría:

- Enfermo / Sano
- Gato / Perro / Pájaro
- **Spam / no Spam**

## Regresión



La variable de salida es un valor:

- Precio
- Cantidad

Machine Learning



Aprendizaje Supervisado



## Clasificación



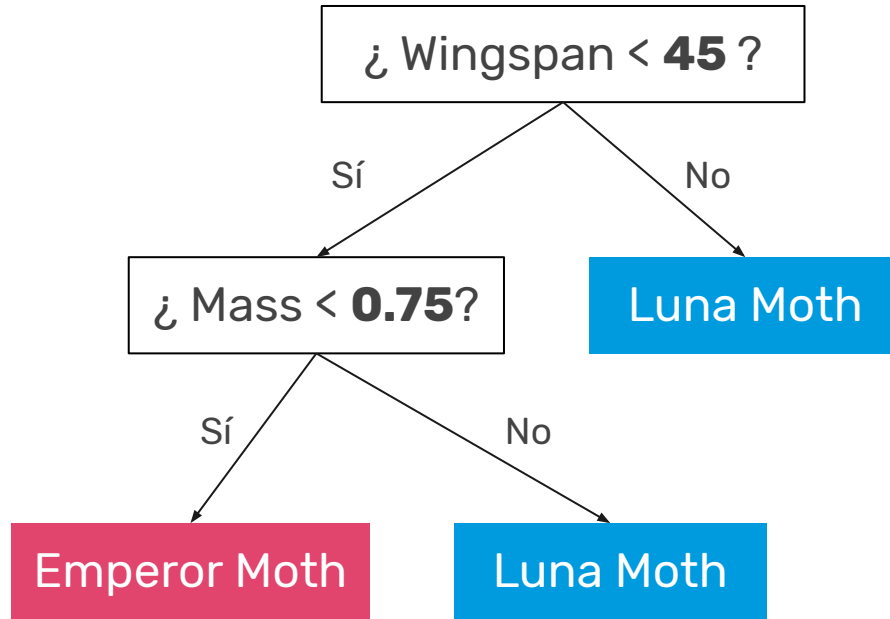
## Modelos

- **Árbol de Decisión**
- Support Vector Machines
- k-nearest neighbors
- Random Forest
- Perceptrón
- etc...

# Repaso: Árboles de Decisión

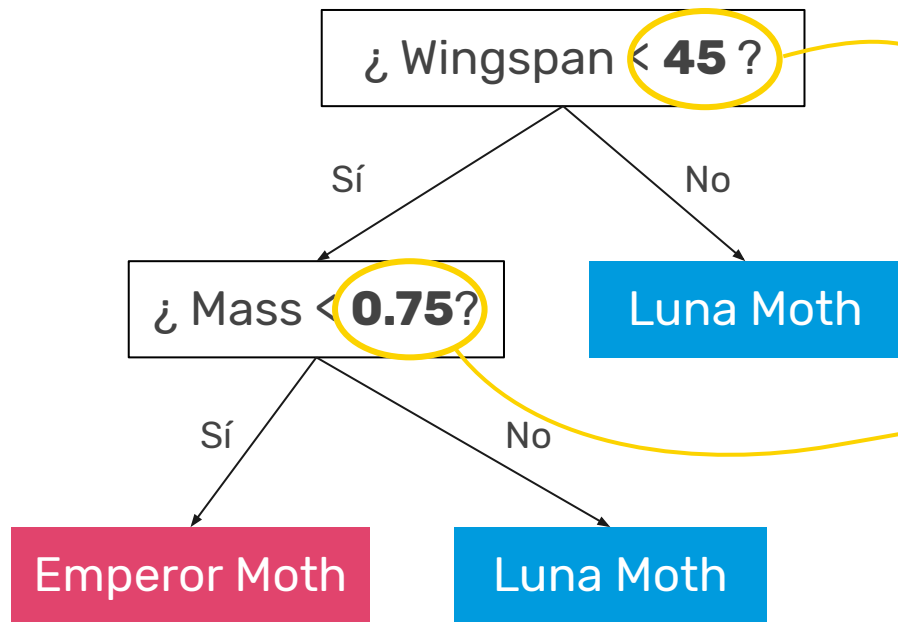


# Árboles de Decisión



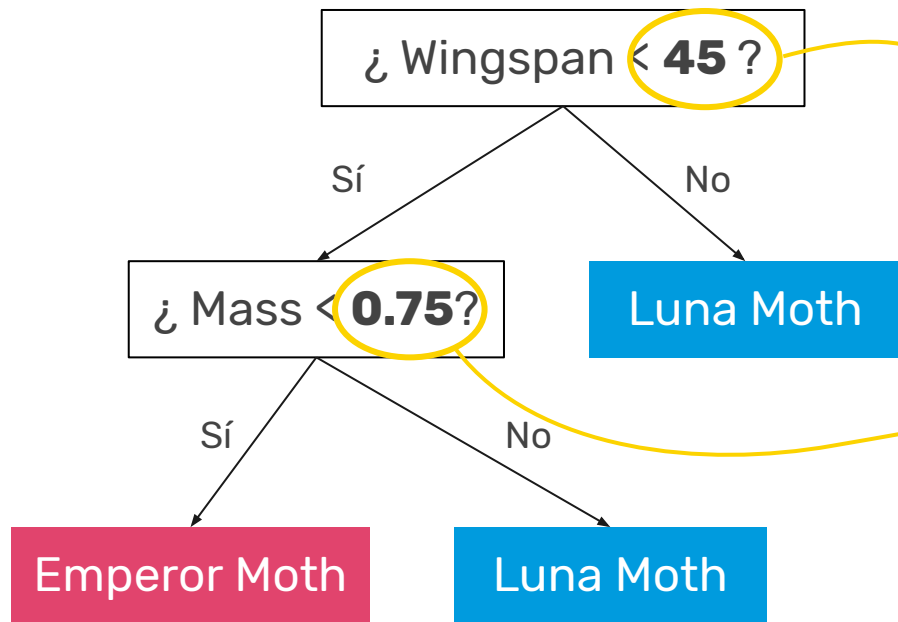


# ¿Por qué decimos que es Machine Learning?



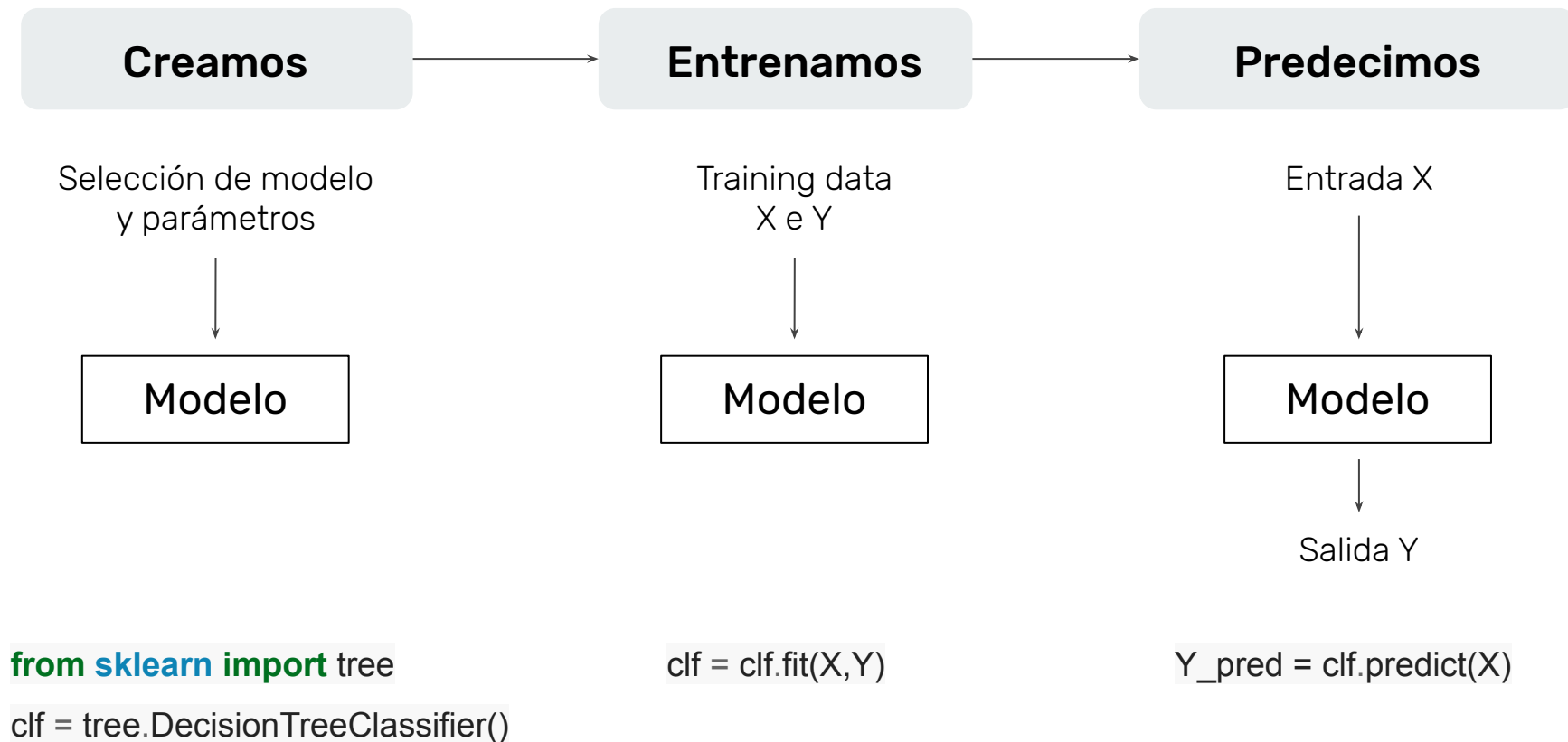
Es **Machine Learning** porque estos valores se eligen automáticamente al entrenar el modelo, a partir de los datos **X** e **Y**.

# ¿Por qué decimos que es Machine Learning?



Hoy vamos a ver **cómo** es que se eligen estos valores a partir de un proceso matemático.

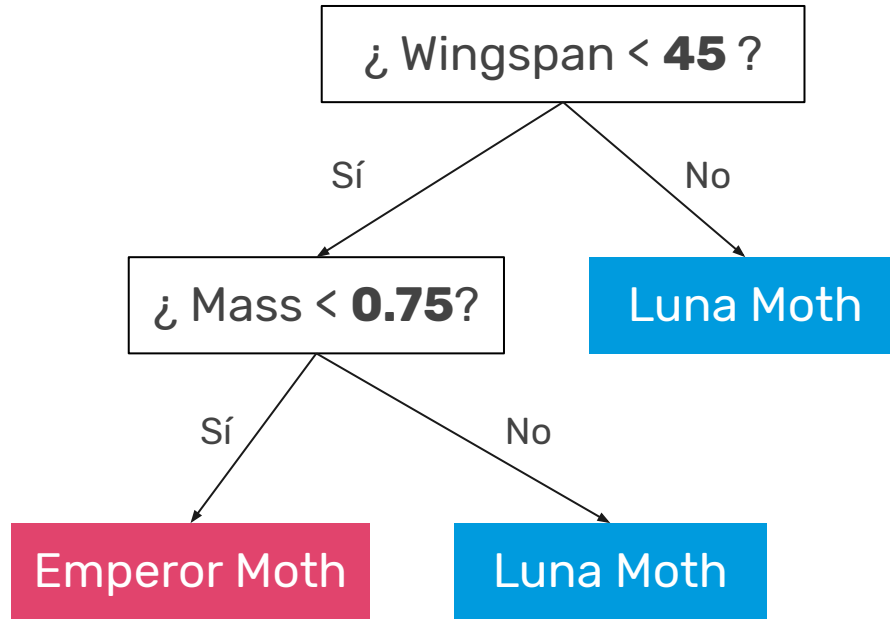
# Flujo de trabajo **Scikit Learn**



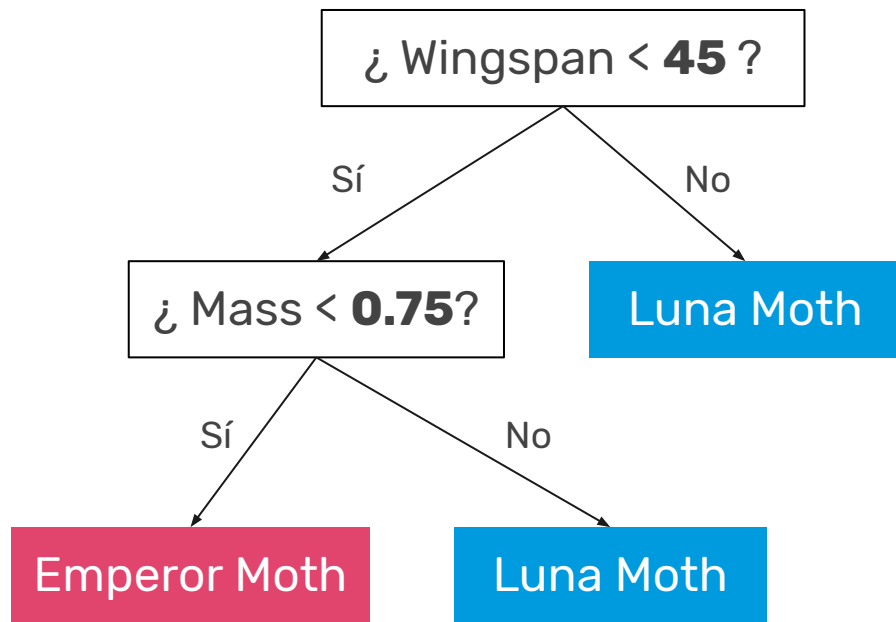
# Árboles de Decisión



Un árbol de decisión “hace preguntas” y va clasificando de acuerdo a las respuestas.

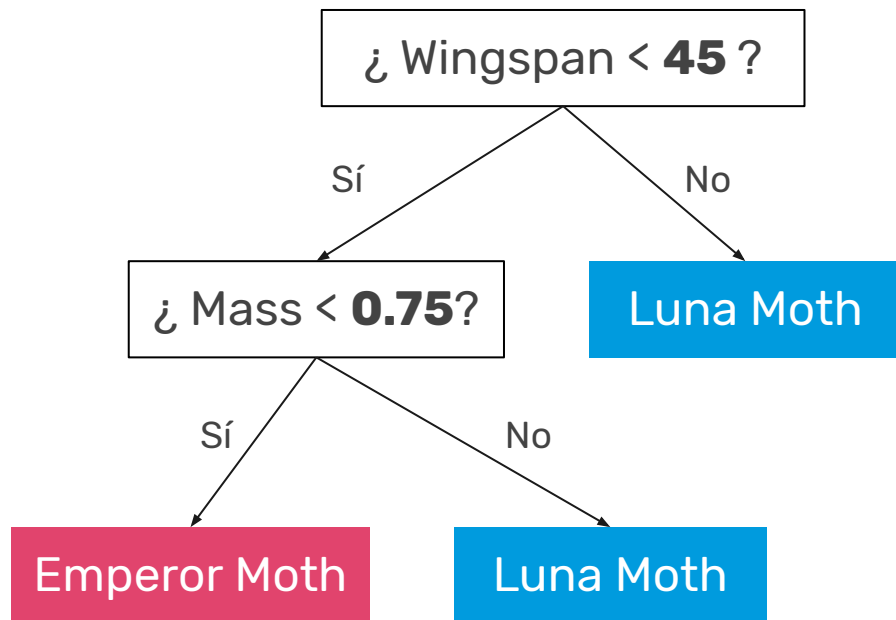


Un árbol de decisión “hace preguntas” y va clasificando de acuerdo a las respuestas.



¿Cómo decide qué preguntar?

Un árbol de decisión “hace preguntas” y va clasificando de acuerdo a las respuestas.

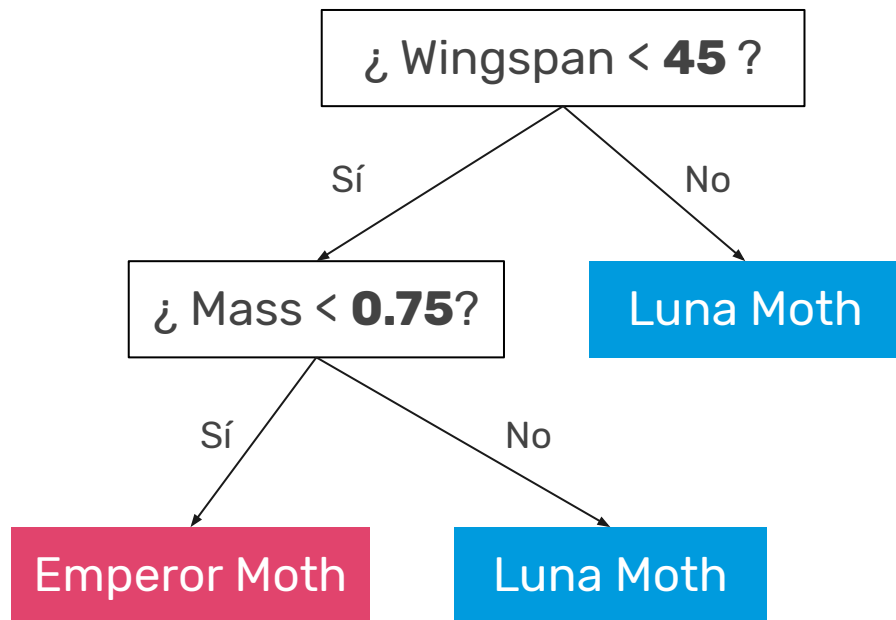


## ¿Cómo decide qué preguntar?

1. Impureza Gini
1. Entropía/Ganancia de información

Son cálculos que se hacen sobre los datos que ayudan a descubrir cuán bueno es un feature para separar las instancias por sus etiquetas.

Un árbol de decisión “hace preguntas” y va clasificando de acuerdo a las respuestas.



## ¿Cómo decide qué preguntar?

### 1. Impureza Gini

1. Entropía/Ganancia de información

Son cálculos que se hacen sobre los datos que ayudan a descubrir cuán bueno es un feature para separar las instancias por sus etiquetas.



# Impureza Gini

Supongamos que tenemos este dataset para el ejemplo de las polillas del video (muy simplificado).

**¿Cuál será un mejor atributo para “preguntar”?**

Pero... ¿qué es un mejor atributo?

Intuitivamente, un mejor atributo será el que separe **“mejor”** las clases.

que las muestras obtenidas sean lo más “puras” posibles. Es decir, tengan instancias de una sola de las clases.

Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth

# Impureza Gini

A simple vista, es muy difícil determinar cuál atributo es mejor para separar clases, y eso que sólo tenemos diez instancias, dos atributos y solamente dos valores por atributo.

Para hacerlo eficientemente, necesitamos algún estadístico que cuantifique la pureza de las muestras.

Para eso existe la **Impureza Gini**.

Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth

# Impureza Gini: ¿Cómo funciona?

Probemos construyendo una *pregunta* por cada feature y veamos cual deja mejor separadas las instancias.

¿ Masa < **0.75 gr** ?

Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth

# Impureza Gini: ¿Cómo funciona?

Probemos construyendo una *pregunta* por cada feature y veamos cual deja mejor separadas las instancias.

¿ Masa < **0.75 gr** ?

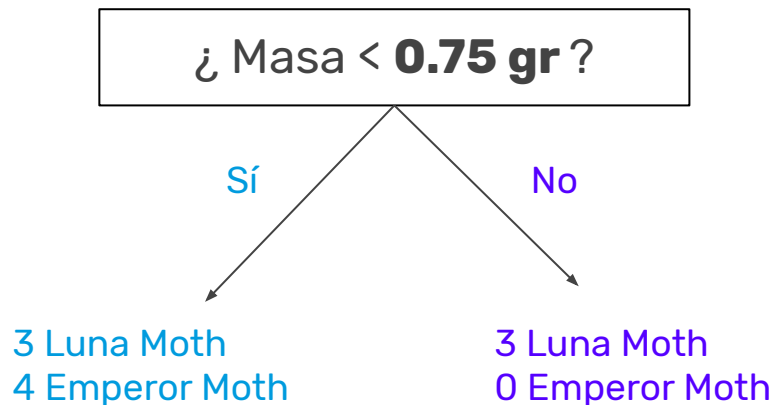
Sí

3 Luna Moth  
4 Emperor Moth

Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
<b>Menor a 0.75 gr</b>	Mayor a 45 mm	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
<b>Menor a 0.75 gr</b>	Mayor a 45 mm	Luna Moth
<b>Menor a 0.75 gr</b>	Menor a 45 mm	Emperor Moth
<b>Menor a 0.75 gr</b>	Mayor a 45 mm	Luna Moth
<b>Menor a 0.75 gr</b>	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
<b>Menor a 0.75 gr</b>	Menor a 45 mm	Emperor Moth
<b>Menor a 0.75 gr</b>	Menor a 45 mm	Emperor Moth

# Impureza Gini: ¿Cómo funciona?

Probemos construyendo una *pregunta* por cada feature y veamos cual deja mejor separadas las instancias.



Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth

# Impureza Gini: ¿Cómo funciona?

Probemos construyendo una *pregunta* por cada feature y veamos cual deja mejor separadas las instancias.

¿ Envergadura < **45 mm** ?

Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth

# Impureza Gini: ¿Cómo funciona?

Probemos construyendo una *pregunta* por cada feature y veamos cual deja mejor separadas las instancias.

¿ Envergadura < **45 mm** ?

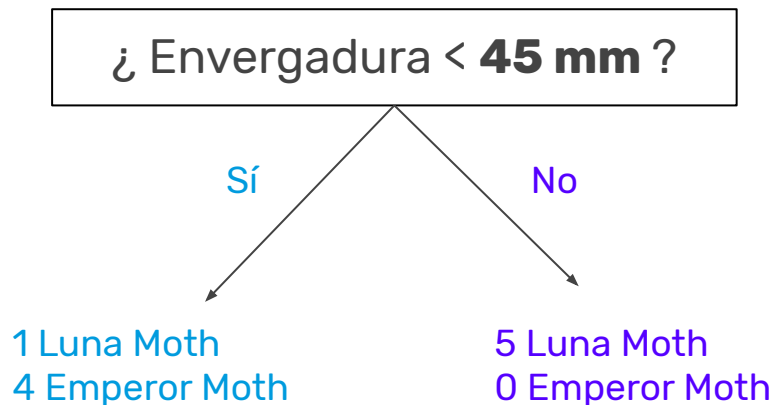
Sí

1 Luna Moth  
4 Emperor Moth

Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	Mayor a 45 mm	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth

# Impureza Gini: ¿Cómo funciona?

Probemos construyendo una *pregunta* por cada feature y veamos cual deja mejor separadas las instancias.

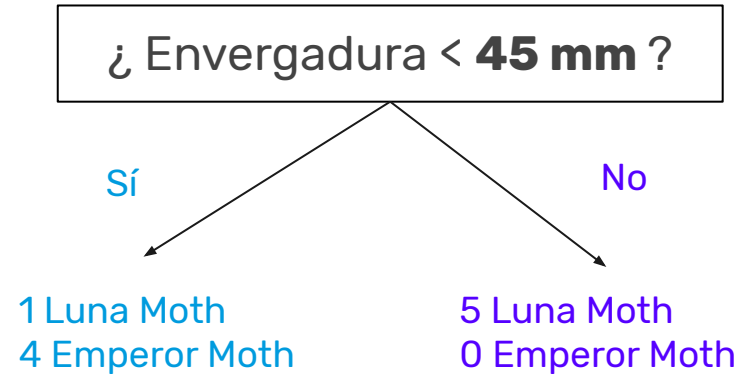
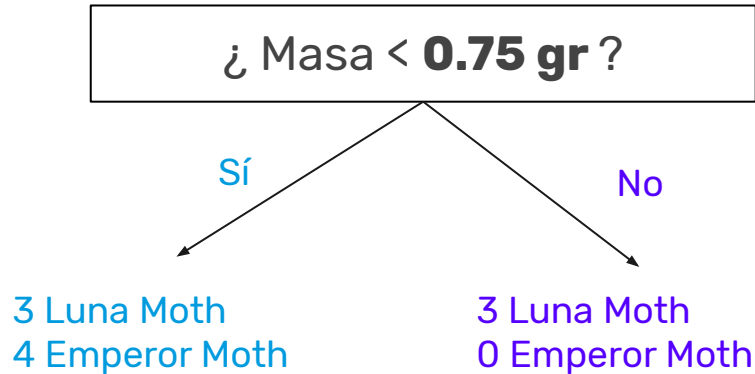


Masa	Envergadura	Tipo de polilla
Mayor a 0.75 gr	<b>Mayor a 45 mm</b>	Luna Moth
Menor a 0.75 gr	<b>Mayor a 45 mm</b>	Luna Moth
Mayor a 0.75 gr	Menor a 45 mm	Luna Moth
Menor a 0.75 gr	<b>Mayor a 45 mm</b>	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	<b>Mayor a 45 mm</b>	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Mayor a 0.75 gr	<b>Mayor a 45 mm</b>	Luna Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth
Menor a 0.75 gr	Menor a 45 mm	Emperor Moth



# Impureza Gini

¿Cuál de las dos preguntas *separó* mejor las clases?





# Tratemos de cuantificarlo...

## Impureza Gini



1. Calculamos la **Impureza Gini inicial** de la muestra.
2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.
3. Elegimos el atributo con **mayor reducción de impureza** (Ganancia Gini).
4. Si consideramos que las instancias ya están clasificadas suficientemente bien, FIN. Si no, seguimos construyendo el árbol de forma iterativa, tomando como muestra inicial la muestra de cada hoja y realizando los pasos 1 - 4.

## Impureza Gini

1. Calculamos la **Impureza Gini inicial** de la muestra.

$$Gini_{inicial} = 1 - (proporción\ de\ Luna\ Moth)^2 - (proporción\ de\ Emperor\ Moth)^2$$



## Impureza Gini

1. Calculamos la **Impureza Gini inicial** de la muestra.

$$Gini_{inicial} = 1 - (\text{proporción de Luna Moth})^2 - (\text{proporción de Emperor Moth})^2$$

**Como son diez instancias, (6 Luna Moth y 4 Emperor Moth), entonces:**

$$Gini_{inicial} = 1 - (6/10)^2 - (4/10)^2 = 0.48$$



## Impureza Gini



1. Calculamos la **Impureza Gini inicial** de la muestra.

$$Gini_{inicial} = 1 - (\text{proporción de Luna Moth})^2 - (\text{proporción de Emperor Moth})^2$$

**Como son diez instancias, (6 Luna Moth y 4 Emperor Moth), entonces:**

$$Gini_{inicial} = 1 - (6/10)^2 - (4/10)^2 = 0.48$$

**\*Si la muestra tiene solamente miembros de una clase, entonces**

$$Gini = 1 - (\text{proporción única clase})^2 = 0$$

**\*y si tiene mitad y mitad:**

$$Gini = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

## Impureza Gini



2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.

¿ Masa < **0.75 gr** ?

Sí

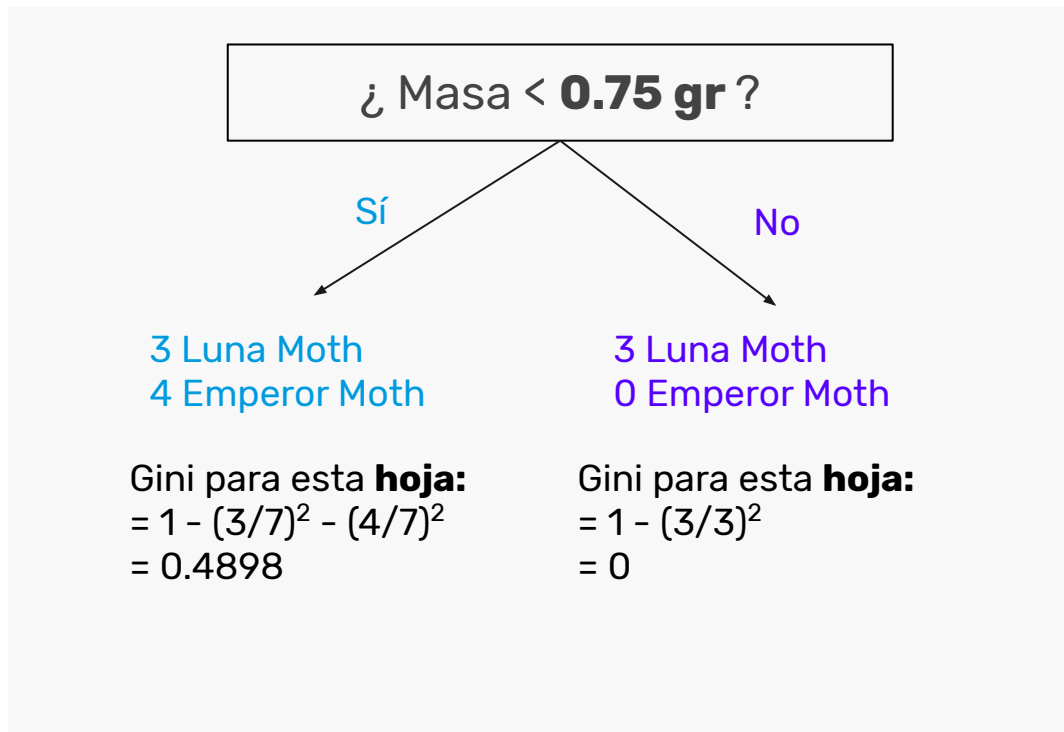
3 Luna Moth  
4 Emperor Moth

Gini para esta **hoja**:  
 $= 1 - (3/7)^2 - (4/7)^2$   
 $= 0.4898$

## Impureza Gini



2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.

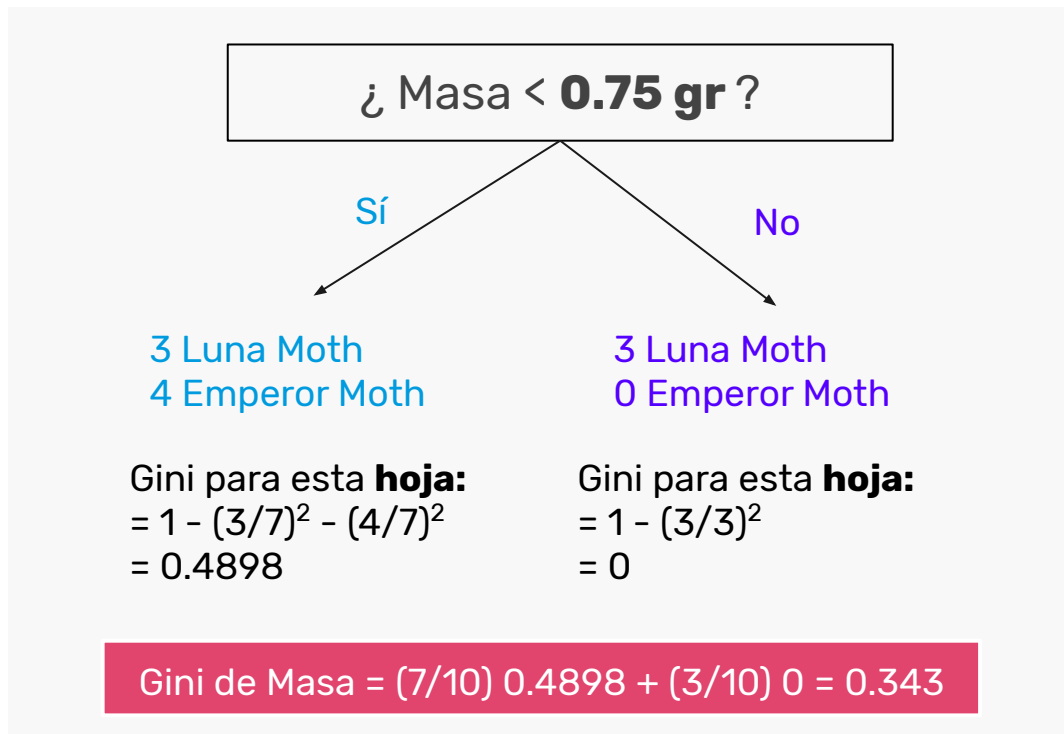




## Impureza Gini



2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.



## Impureza Gini



2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.

¿ Envergadura < **45 mm** ?

Sí

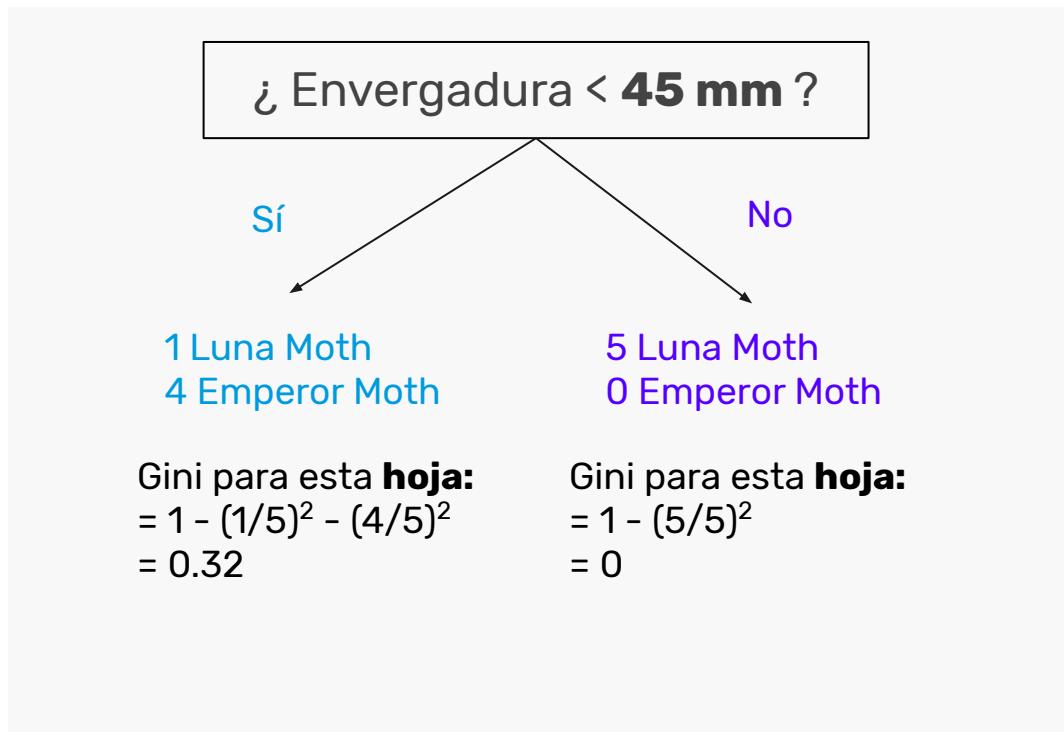
1 Luna Moth  
4 Emperor Moth

Gini para esta **hoja**:  
 $= 1 - (1/5)^2 - (4/5)^2$   
 $= 0.32$

## Impureza Gini



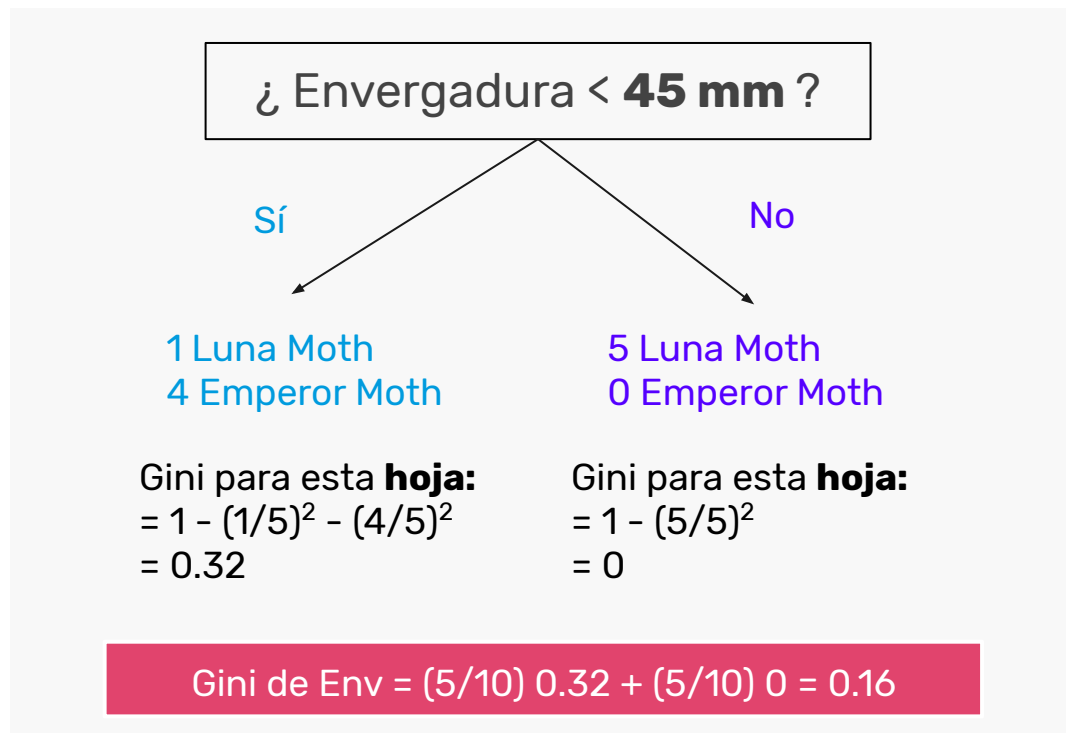
2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.



## Impureza Gini



2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.



## Impureza Gini

3. Elegimos el atributo con **mayor reducción de impureza** (Ganancia Gini).

- Masa:  $0.48 - 0.343 = 0.137$

- Envergadura:  $0.48 - 0.16 = 0.32$





4. Si consideramos que las instancias ya están clasificadas suficientemente bien, **FIN.**

**Si no,** seguimos construyendo el árbol de forma iterativa, tomando como muestra inicial la muestra de cada hoja y realizando los pasos 1 - 4.

## Impureza Gini



### En resumen...

1. Calculamos la **Impureza Gini inicial** de la muestra.
2. Calculamos la **Impureza Gini** luego de hacer cada pregunta. Para ellos, hacemos un **promedio ponderado** de las impurezas resultantes en cada **hoja**, por pregunta.
3. Elegimos el atributo con **mayor reducción de impureza** (Ganancia Gini).
4. Si consideramos que las instancias ya están clasificadas suficientemente bien, FIN. Si no, seguimos construyendo el árbol de forma iterativa, tomando como muestra inicial la muestra de cada hoja y realizando los pasos 1 - 4.

# Árboles: Algunos comentarios

---

1. **Entropía/ganancia de información** es otro criterio que podemos utilizar para medir el grado de impureza de una muestra y elegir el atributo que más la reduce. Conceptualmente es MUY parecido.



# Árboles: Algunos comentarios

---

1. **Entropía/ganancia de información** es otro criterio que podemos utilizar para medir el grado de impureza de una muestra y elegir el atributo que más la reduce. Conceptualmente es MUY parecido.
2. Existen otras métricas que se podrían utilizar, que tienen ventajas en algunas situaciones específicas (por ejemplo, **Gain Ratio**, que corrige la preferencia de ganancia de información por atributos con demasiados valores) .

# Árboles: Algunos comentarios

---

1. **Entropía/ganancia de información** es otro criterio que podemos utilizar para medir el grado de impureza de una muestra y elegir el atributo que más la reduce. Conceptualmente es MUY parecido.
2. Existen otras métricas que se podrían utilizar, que tienen ventajas en algunas situaciones específicas (por ejemplo, **Gain Ratio**, que corrige la preferencia de ganancia de información por atributos con demasiados valores) .
3. Nosotros aquí mostramos un ejemplo de **Clasificación Binaria** (dos clases). Los árboles generalizan muy bien a problemas multiclase y de regresión.

# Árboles: Algunos comentarios

---

1. **Entropía/ganancia de información** es otro criterio que podemos utilizar para medir el grado de impureza de una muestra y elegir el atributo que más la reduce. Conceptualmente es MUY parecido.
2. Existen otras métricas que se podrían utilizar, que tienen ventajas en algunas situaciones específicas (por ejemplo, **Gain Ratio**, que corrige la preferencia de ganancia de información por atributos con demasiados valores) .
3. Nosotros aquí mostramos un ejemplo de **Clasificación Binaria** (dos clases). Los árboles generalizan muy bien a problemas multiclase y de regresión.
4. Hay mucha jerga en árboles: hojas, raíz, nodo, poda (pruning), Gini, información, profundidad, etc. Es fácil marearse. [Este artículo](#) - que compartimos la clase anterior -, la documentación de Scikit-Learn - que podrán encontrar los links dentro de pocas diapositivas - y, sobretodo, la práctica, les servirán para ir incorporándolos.

# Árboles: Ventajas y desventajas

---



- Simple de entender, interpretar y visualizar. Esto es una gran ventaja, también, al momento de comunicar nuestro trabajo.
- Entrenamiento rápido.
- Modelo base para modelos más complejos (Random Forest, xgboost, etc.).
- ¡Muchas implementaciones y variantes!



- Poder de generalización relativamente bajo en muchas circunstancias.
- Desempeño inferior a modelos más modernos.
- ¡Muchas implementaciones y variantes!

## En Scikit-Learn

El módulo que contiene la implementación de árboles de decisión en Scikit-Learn es *tree*.

Como siempre, la [documentación](#) es muy buena.

Sus principales clases son:

- [DecisionTreeClassifier](#)
- [DecisionTreeRegressor](#) (esta la usaremos más adelante cuando veamos regresión).

**Recomendamos mirar sus atributos, métodos y ejemplos.**



# Hands-on training



# DS\_Clase\_16\_Arboles.ipynb

Parte 1 y 2



A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup is placed on a matching white saucer. In the background, a white napkin and a silver fork are visible, though they are out of focus. The overall lighting is soft and warm, creating a cozy atmosphere.

**¡BREAK!**

---





# Evaluación de Modelos



# ¿Cómo podemos evaluar si el modelo está *aprendiendo* o no de nuestros datos?<sup>1</sup>

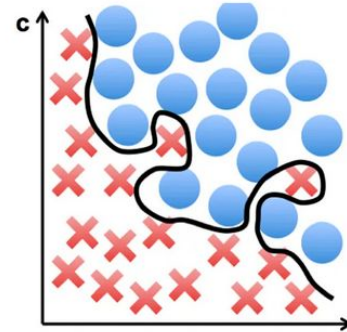
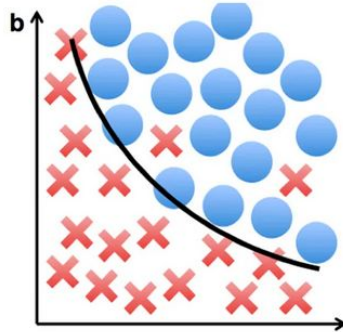
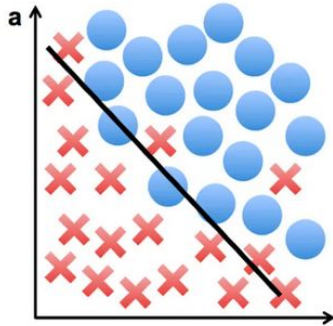
Una forma práctica de evaluar si nuestro modelo aprendió o no de nuestro datos es **observar su desempeño frente a nuevas instancias**.

Pero, ¿por qué necesitamos nuevas instancias y no usamos, simplemente, las instancias que usamos para entrenar?



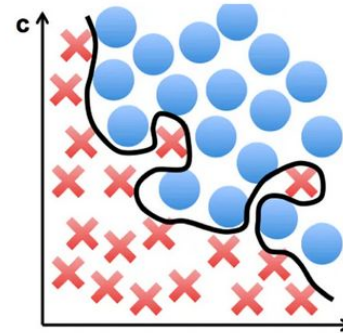
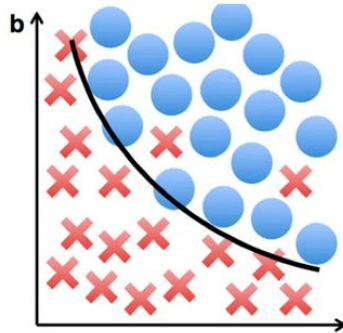
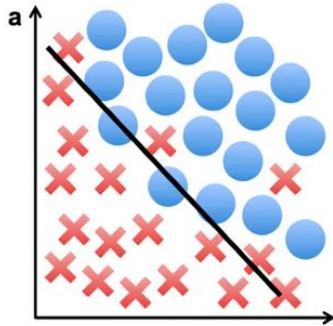
<sup>1</sup>Podríamos preguntarnos también qué es aprender, pero por ahora vamos a ignorar esa pregunta.

# Consideremos estas 3 situaciones...



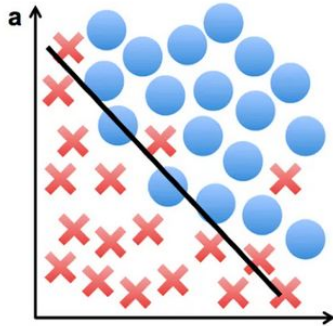
**¿Qué modelo les parece mejor?**

# Consideremos estas 3 situaciones...

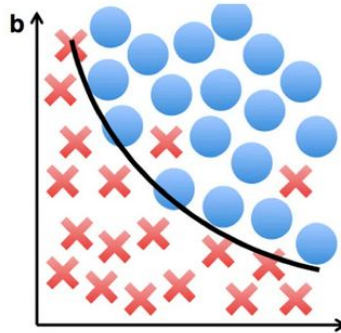


El **modelo a** es muy simple y no reproduce correctamente la frontera entre las clases. Llamaremos **underfitting** a esta situación.

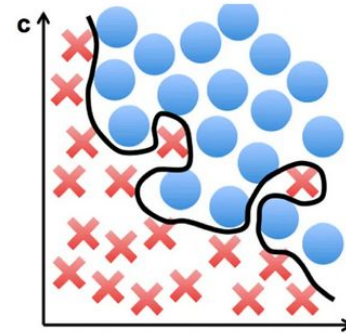
# Consideremos estas 3 situaciones...



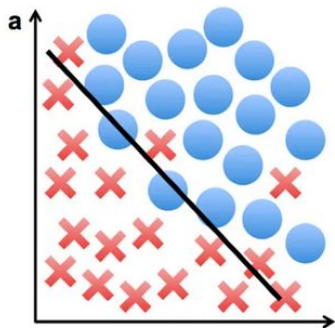
El **modelo a** es muy simple y no reproduce correctamente la frontera entre las clases. Llamaremos **underfitting** a esta situación.



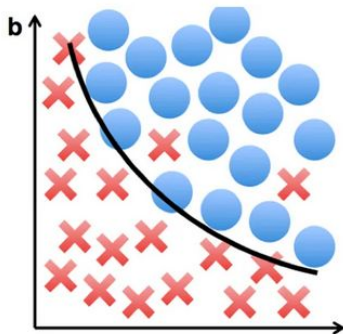
El **modelo b** tiene la complejidad suficiente para encontrar una frontera que parece apropiada para estos datos.



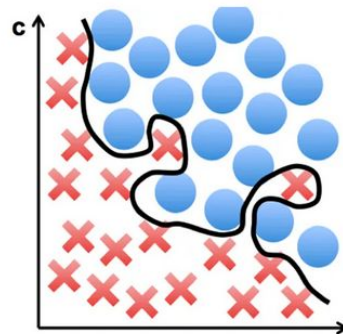
# Consideremos estas 3 situaciones...



El **modelo a** es muy simple y no reproduce correctamente la frontera entre las clases. Llamaremos **underfitting** a esta situación.



El **modelo b** tiene la complejidad suficiente para encontrar una frontera que parece apropiada para estos datos.



El **modelo c** parece muy flexible y se adaptó demasiado a los datos con los que fue entrenado. Llamaremos **overfitting** a esta situación.

WARNING

Entonces, podría ocurrir que el modelo se aprenda “de memoria” los datos con los que fue entrenado, por lo que **es importante evaluarlo con datos que nunca vio**.

Interviewer: What's your biggest strength?

Me: I'm an expert in machine learning.

Interviewer: What's  $6 + 10$ ?

Me: Zero.

Interviewer: Nowhere near, it's 16.

Me: It's 16.

Interviewer: Ok... What's  $10 + 20$ ?

Me: It's 16.

Muchos modelos que utilizaremos son muy flexibles y, de esas dos situaciones, **en general tendremos que preocuparnos más por el Sobreajuste (Overfitting).\***

\*Más adelante, también veremos técnicas más complejas para evaluar correctamente nuestros modelos y prevenir el Overfitting y el Underfitting



# ¿Cómo entreno un modelo?

En nuestro flujo de trabajo, tendremos que emular una situación donde el modelo es entrenado con ciertos datos y luego es evaluado con datos nuevos.

¡Hacerlo es muy sencillo!

Antes de entrenar un modelo...

- 1. Separo una porción de los datos**
- 2. Evalúo el desempeño del modelo sobre esos datos.**

# ¿Cómo entreno un modelo?

En nuestro flujo de trabajo, tendremos que emular una situación donde el modelo es entrenado con ciertos datos y luego es evaluado con datos nuevos.

¡Hacerlo es muy sencillo!

Antes de entrenar un modelo...

- 1. Separo una porción de los datos**
- 2. Evalúo el desempeño del modelo sobre esos datos.**

---

En general, los datos se separan al azar para evitar cualquier orden o estructura subyacente en los datos<sup>1</sup>.

<sup>1</sup> En algunos problemas, por ejemplo cuando queremos entrenar un modelo que haga predicciones a futuro, esto no es válido.

## En Scikit-Learn

¡Esta función es tan importante que viene en todos los entornos de desarrollo de Machine Learning!

En Scikit-Learn, la función se llama [train\\_test\\_split](#).

```
sklearn.model_selection.train_test_split
```

```
sklearn.model_selection. train_test_split (*arrays, **options)
```

[\[source\]](#)



**Para pensar, ¿cómo controlar el overfitting y el underfitting en los árboles de decisión?**



## Para pensar, ¿cómo controlar el overfitting y el underfitting en los árboles de decisión?

El principal parámetros que controla si “overfiteamos” o “underfiteamos” es la profundidad del árbol. Para evitar el overfitting, existen algunos métodos:

- **Criterio de parada:** no construir más allá de cierta profundidad. Ésta es una de las reglas más usadas.
- **Podar:** construir el árbol entero. Podar las ramas cuando ello mejore la performance sobre datos separados
- Y más...

# Hands-on training



# DS\_Clase\_16\_Arboles.ipynb

Parte 3 y 4



# Para la próxima

---

1. Ver los videos de la plataforma “Machine Learning: KNN” (**¡nos salteamos algunos videos!**)
2. Completar el notebook de hoy.



ACÀMICA