ACÁMICA

¡Bienvenidos/as a Data Science!

Agenda

¿Cómo anduvieron?

Repaso: Machine Learning y Árboles de Decisión

Explicación: KNN

Hands-On

Break

Explicación: Métricas de Evaluación para Clasificación

Cierre

¿Cómo anduvieron?

Proyecto 2: Modelado

Análisis Exploratorio de Datos (EDA)

fase	ADQUISICIÓN Y EXPLORACIÓN		MODELADO				DEPLOY
entrega	Exploración de datos	Feature Engineering	Machine Learning: Clasificación y Regresión	Optimización de parámetros	Procesam. del lenguaje natural	Sistema de recomendación	Publicación de modelos
tiempo	SEM 1	SEM 5	SEM 8	SEM 12	SEM 14	SEM 18	SEM 22
	SEM 2	SEM 6	SEM 9	SEM 13	SEM 15	SEM 19	SEM 23
	SEM 3	SEM 7	SEM 10		SEM 16	SEM 20	SEM 24
	SEM 4		SEM 11		SEM 17	SEM 21	

Proyecto EDA: Hoja de ruta

Entrega 1

SEM 1 - 7

EDA:

- Python
- Numpy
- Pandas
- Visualización de datos: Matplotlib y Seaborn
- Estadística
- Transformación de Datos
- Outliers

SEM 8

- Intro a Machine Learning
- Aprendizaje Supervisado: Clasificación
- Árboles de Decisión
- Overfitting y Underfitting, Train/Test Split

Usted Está Aquí

SEM 9

- k-Vecinos más cercanos
- Métricas de Evaluación para Clasificación
- Repaso

Entrega 3

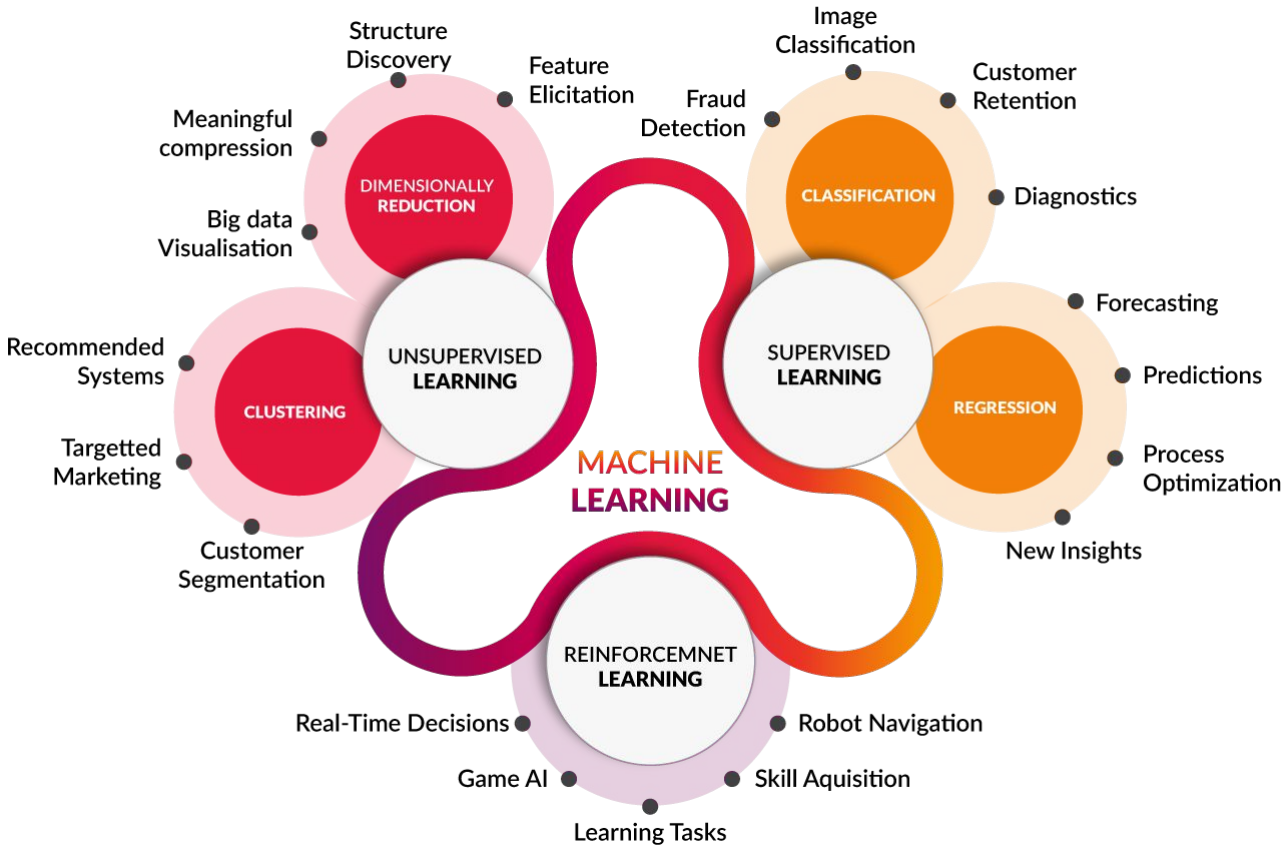
SEM 10

- Aprendizaje Supervisado: Regresión
- Métrica de Evaluación para Regresión

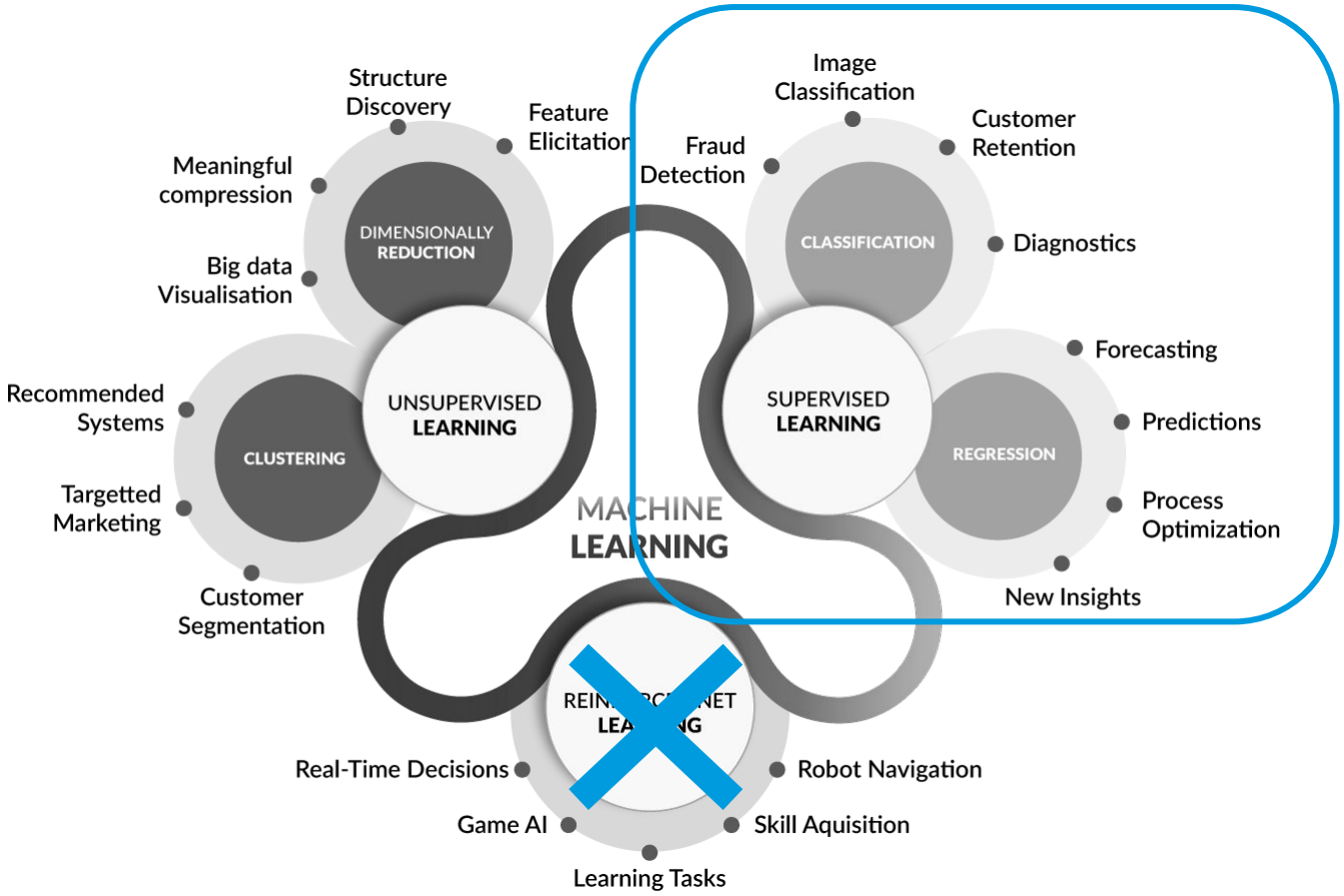
SEM 11

Repaso: Machine Learning y Árboles de Decisión

Mapa


MACHINE LEARNING
UNSUPERVISED LEARNING
SUPERVISED LEARNING
REINFORCEMNET LEARNING
DIMENSIONALLY REDUCTION
CLUSTERING
CLASSIFICATION
REGRESSION
Structure Discovery
Feature Elicitation
Meaningful compression
Big data Visualisation
Image Classification
Customer Retention
Fraud Detection
Diagnostics
Recommended Systems
Targetted Marketing
Customer Segmentation
Forecasting
Predictions
Process Optimization
New Insights
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks

Mapa


Structure Discovery
Feature Elicitation
Meaningful compression
Big data Visualisation
DIMENSIONALLY REDUCTION
Recommended Systems
Targetted Marketing
Customer Segmentation
CLUSTERING
UNSUPERVISED LEARNING
Image Classification
Customer Retention
Fraud Detection
Diagnostics
CLASSIFICATION
SUPERVISED LEARNING
Forecasting
Predictions
Process Optimization
New Insights
REGRESSION
MACHINE LEARNING
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks

Machine Learning

↓

Aprendizaje Supervisado

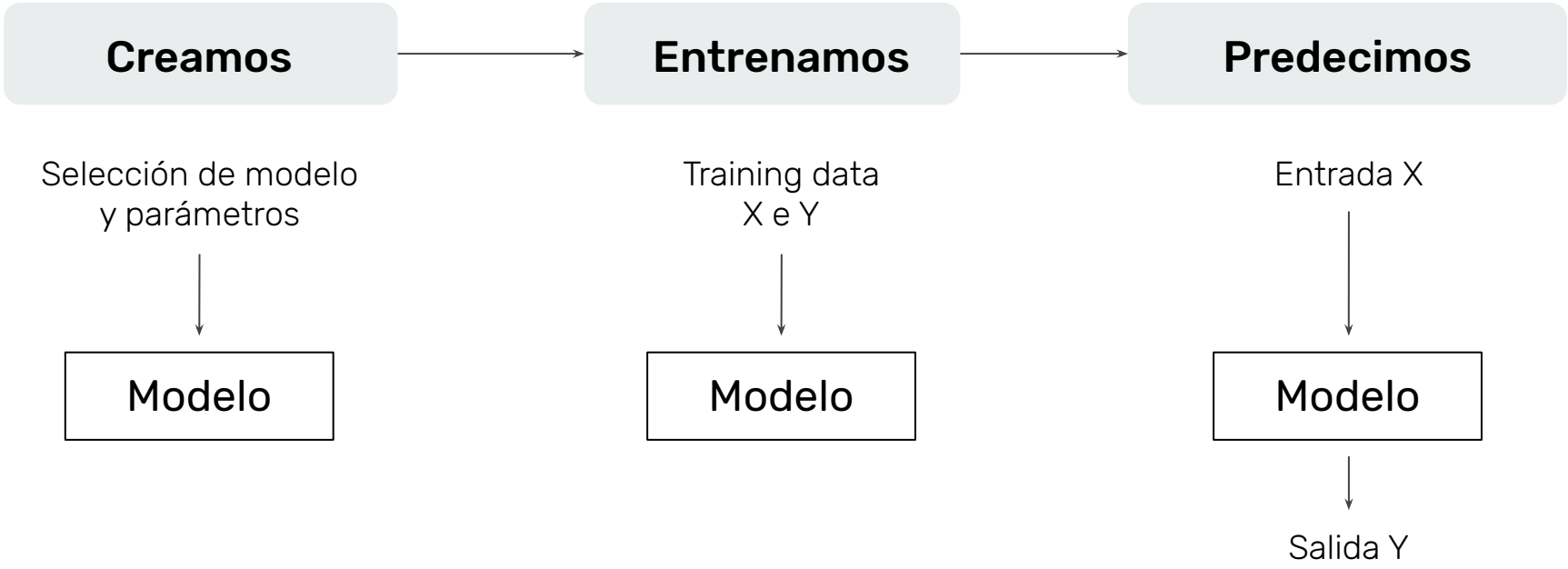
↓

Clasificación

Modelos

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

Flujo de trabajo Scikit Learn

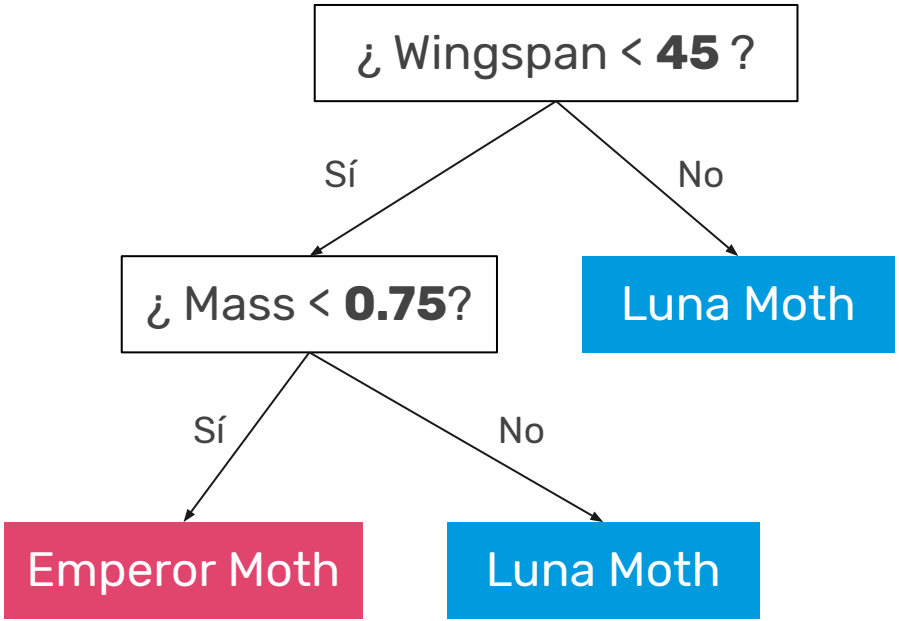



```
from sklearn import tree
clf = tree.DecisionTreeClassifier()
```

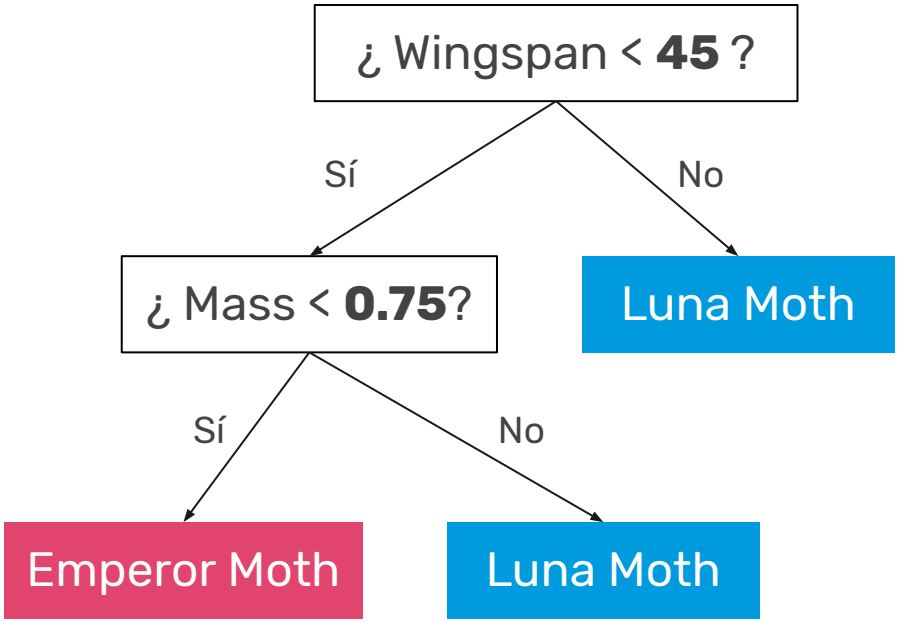
```
clf = clf.fit(X,Y)
```

```
Y_pred = clf.predict(X)
```

Primer modelo: Árboles de Decisión.



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



¿Cómo decide qué preguntar?

1. Impureza Gini
2. 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

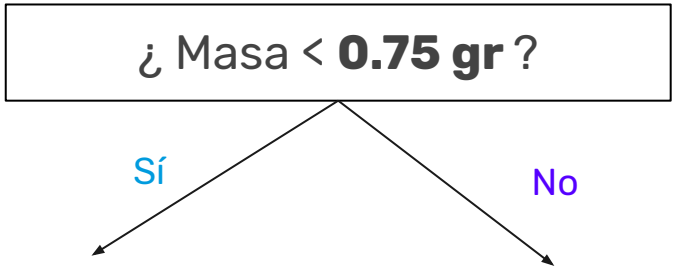
1. Calculamos la **Impureza Gini inicial** de la muestra.

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

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

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



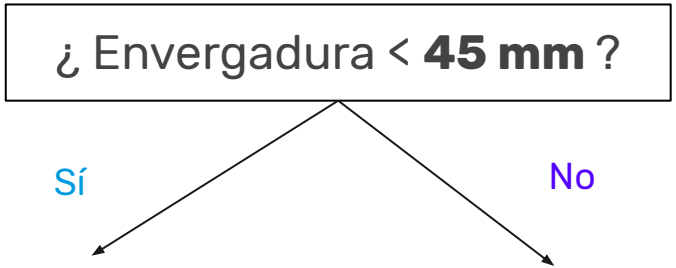
3 Luna Moth
4 Emperor Moth

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

3 Luna Moth
0 Emperor Moth

Gini para esta **hoja:**
$= 1 - (3/3)^2$
$= 0$

Gini de Masa = (7/10) 0.4898 + (3/10) 0 = 0.343



1 Luna Moth
4 Emperor Moth

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

5 Luna Moth
0 Emperor Moth

Gini para esta **hoja:**
$= 1 - (5/5)^2$
$= 0$

Gini de Env = (5/10) 0.32 + (5/10) 0 = 0.16

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!

Hoy veremos un nuevo modelo...

K-vecinos más cercanos

K-vecinos más cercanos (KNN)

Machine Learning

↓

Aprendizaje Supervisado

↓

Clasificación

Modelos

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

KNN - K Nearest Neighbors

IDEA: Dada una nueva instancia de la cual no conocemos la etiqueta objetivo, vamos a asumir que su etiqueta será igual a la de las instancias “vecinas” en el training set.

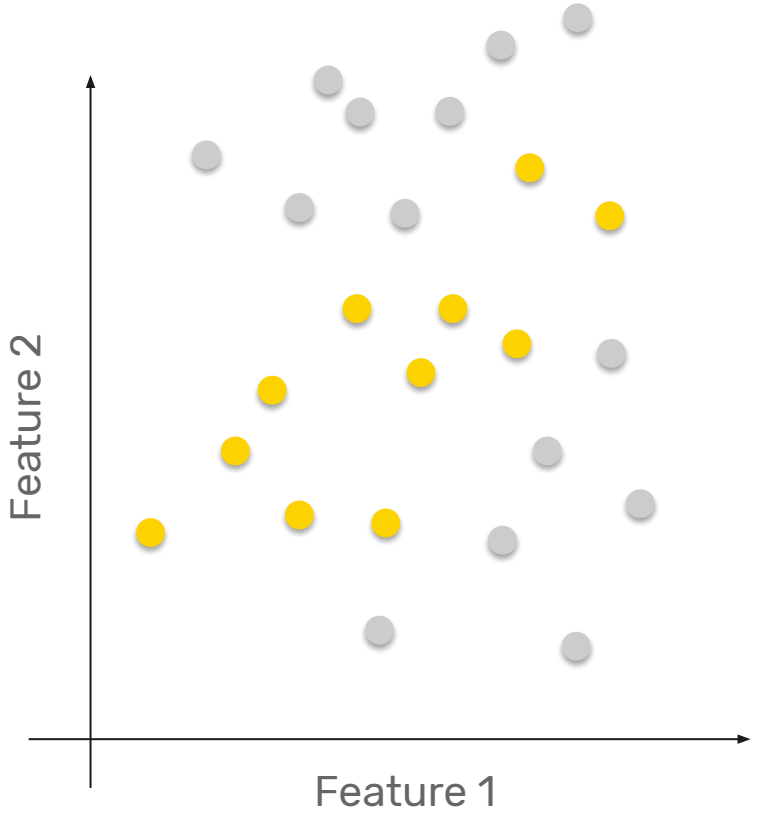
KNN - K Nearest Neighbors

IDEA: Dada una nueva instancia de la cual no conocemos la etiqueta objetivo, vamos a asumir que su etiqueta será igual a la de las instancias “vecinas” en el training set.

O dicho de otra forma...

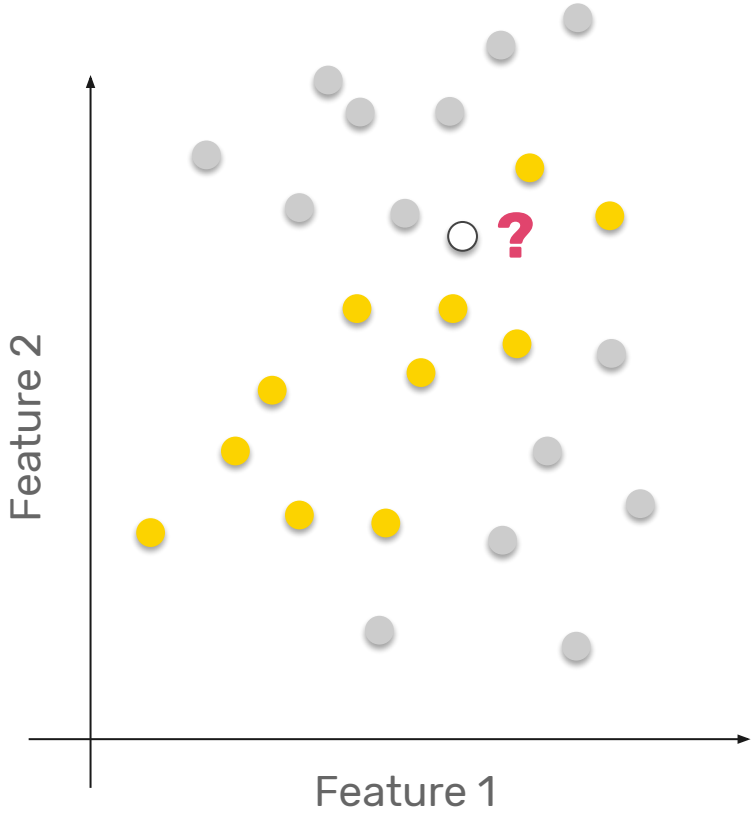
KNN - Ejemplo

Supongamos que tenemos un Dataset con 2 Features, en el cual cada instancia puede pertenecer a una de dos clases: “Gris” o “Amarillo”.



KNN - Ejemplo

Dada una nueva instancia, de la cual no conocemos su clase, vamos a recurrir a sus vecinos para clasificarla.



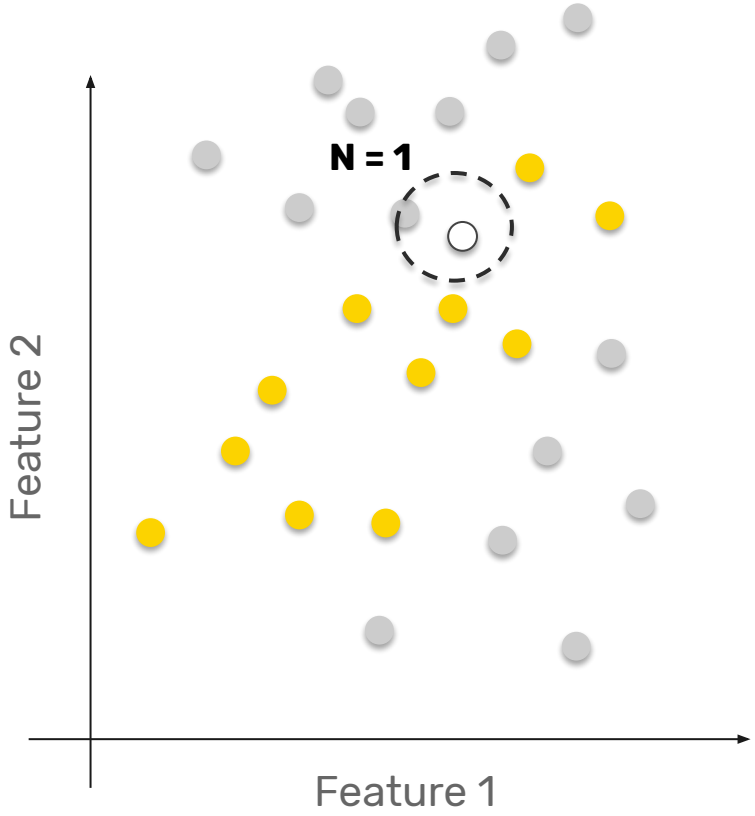
KNN - Ejemplo

¿Por K en nombre del algoritmo?

K es el número de vecinos que miramos para saber la clase de nuestra nueva instancia.

Si tomamos **K=1**, solo miraremos al vecino más cercano.

GRIS

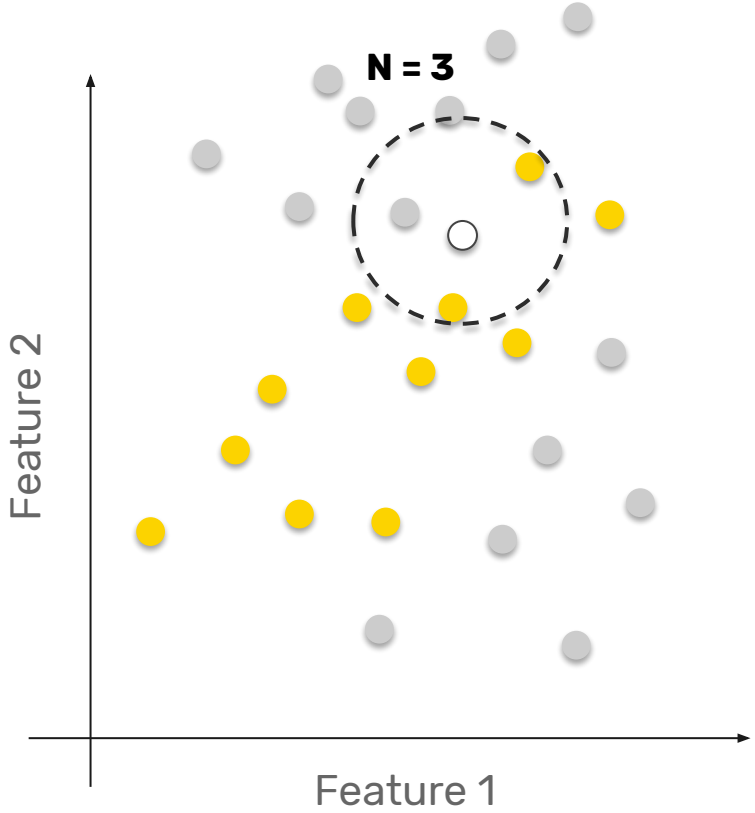


KNN - Ejemplo

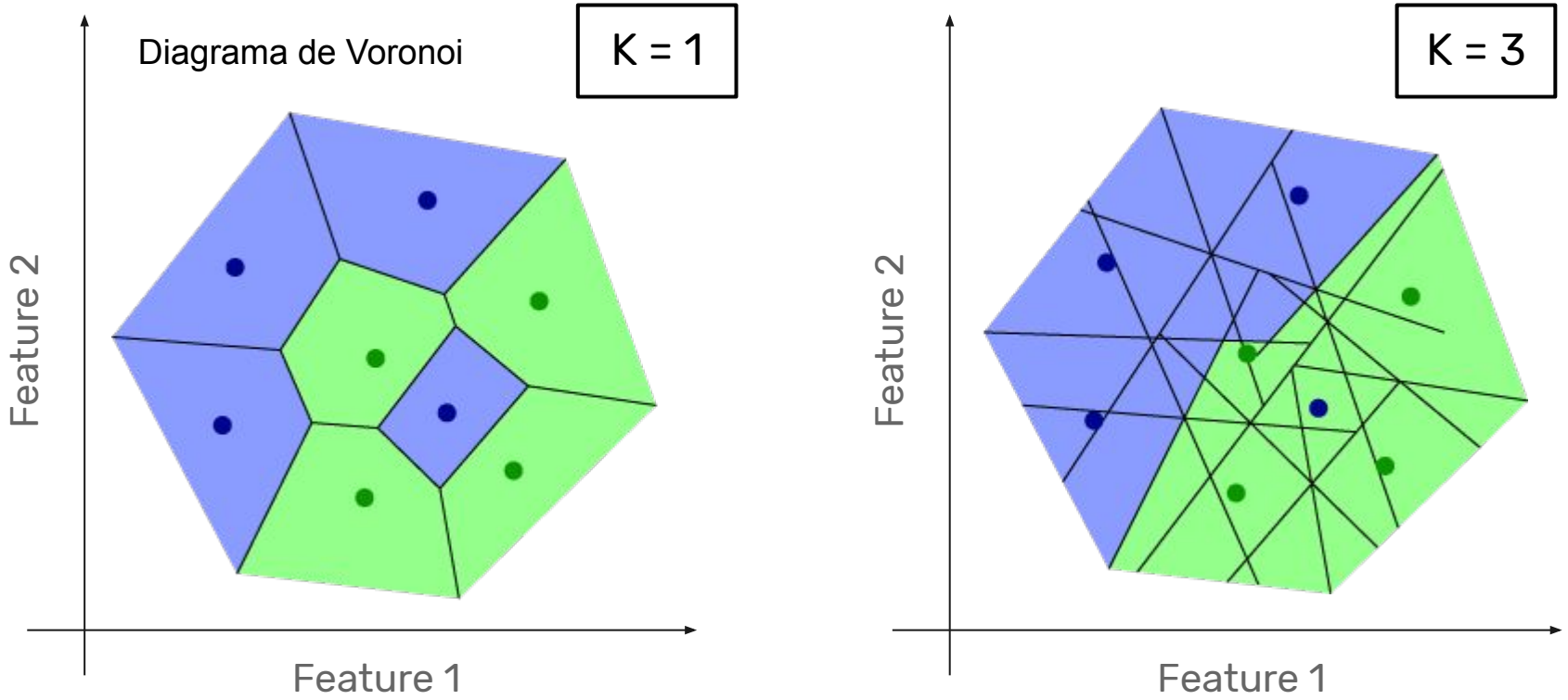
Si elegimos k > 1, se vota según el número de vecinos.

Por ejemplo, con k = 3 tenemos dos vecinos Amarillos y uno Gris.

AMARILLO



Fronteras de decisión según K



https://shapeofdata.wordpress.com/2013/05/07/k-nearest-neighbors/

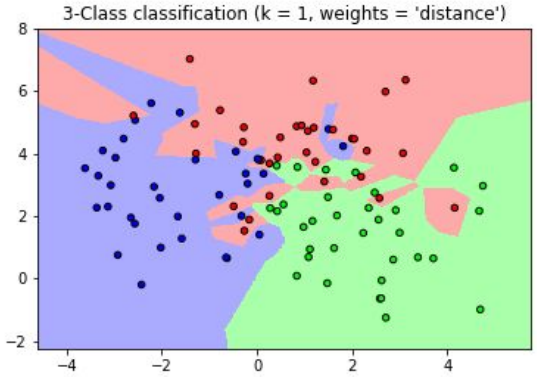
K es lo que llamamos un **Hiperparámetro** del modelo.

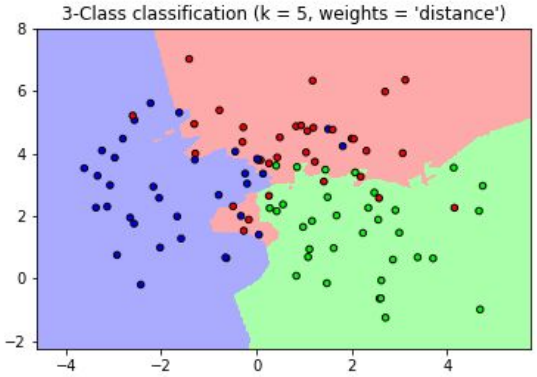
K es lo que llamamos un **Hiperparámetro** del modelo.

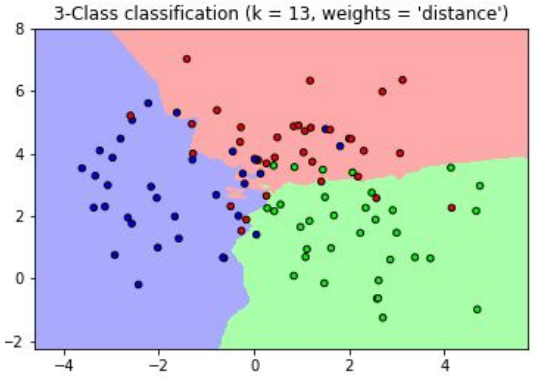
No hay una receta para elegir K de antemano, depende del problema. Muchas veces la solución es probar varios y ver cual modelo se desempeña mejor.

K es lo que llamamos un **Hiperparámetro** del modelo.

No hay una receta para elegir K de antemano, depende del problema. Muchas veces la solución es probar varios y ver cual modelo se desempeña mejor.







Otros hiperparámetros del modelo son el **Peso** y la **Métrica.**

sklearn.neighbors.KNeighborsClassifier

class sklearn.neighbors. **KNeighborsClassifier** (*n_neighbors=5*, *weights='uniform'*, *algorithm='auto'*, *leaf_size=30*, *p=2*, *metric='minkowski'*, *metric_params=None*, *n_jobs=None*, ***kwargs*) [source]

Classifier implementing the k-nearest neighbors vote.

Read more in the User Guide.

Parameters:

n_neighbors : *int, optional (default = 5)*

Number of neighbors to use by default for `kneighbors` queries.

weights : *str or callable, optional (default = 'uniform')*

weight function used in prediction. Possible values:

- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

KNN en Scikit-learn

```
from sklearn.neighbors import KNeighborsClassifier

modelo = KNeighborsClassifier(n_neighbors=10, weights='uniform’)

modelo.fit(X_train,y_train)
```

Ventajas y Desventajas de KNN

- Simple de interpretar
- Entrenamiento rápido.

- Lento para clasificar (predecir)
- Ocupa mucho espacio en el disco (tiene que guardar todo el set de entrenamiento)

Un último comentario sobre KNN...

¡ATENCIÓN!

Es muy importante aplicar reescalar los valores de los features (llevar a escala de z score) antes de usar este modelo.

¿Se imaginan por qué esto es así?

Para pensar...

¿cómo controlar el overfitting y el underfitting en KNN?

Hands-on training

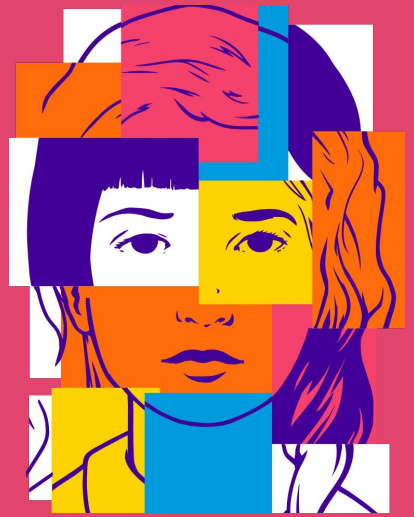
Hands-on training

DS_Clase_17_KNN.ipynb

¡BREAK!

Ph. Credit: Drew Coffmann

Métricas de evaluación para Clasificación



¿Cómo evaluamos los resultados de una clasificación?

¿Cómo evaluamos los resultados de una **clasificación**?

Clasificación Binaria

Precisión/Exhaustividad

F-Score

Matriz de Confusión

Clasificación Binaria

Problema general: separar los elementos de un conjunto en **dos grupos** bajo cierta/s regla/s de clasificación.

Clasificación Binaria

Problema general: separar los elementos de un conjunto en **dos grupos** bajo cierta/s regla/s de clasificación.

Ejemplos:

Examen médico

Enfermo / No enfermo

Test de embarazo

Educación

Aprobado / No aprobado

Bromatología

Apto / No apto para consumo

Control calidad

Seguro / No seguro

Clasificación Binaria

En general, nos interesa un grupo en particular.

Clasificación Binaria

¿Qué puede pasar con **un** test?
(ejemplo: test de embarazo)

Clasificación Binaria

¿Qué puede pasar con **un** test?
(ejemplo: test de embarazo)

Verdadero positivo (acierto)
test positivo, paciente embarazada

Verdadero negativo (acierto)
test negativo, paciente no embarazada

Clasificación Binaria

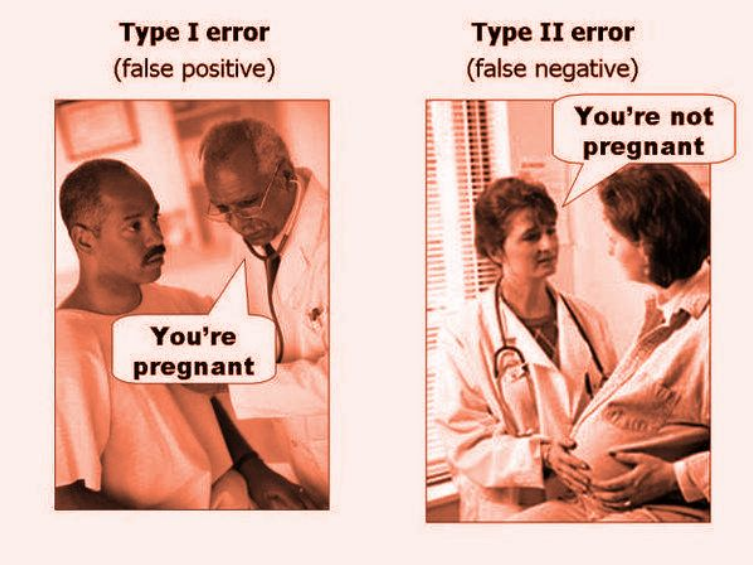
¿Qué puede pasar con **un** test? (ejemplo: test de embarazo)

Verdadero positivo (acierto)
test positivo, paciente embarazada

Verdadero negativo (acierto)
test negativo, paciente no embarazada

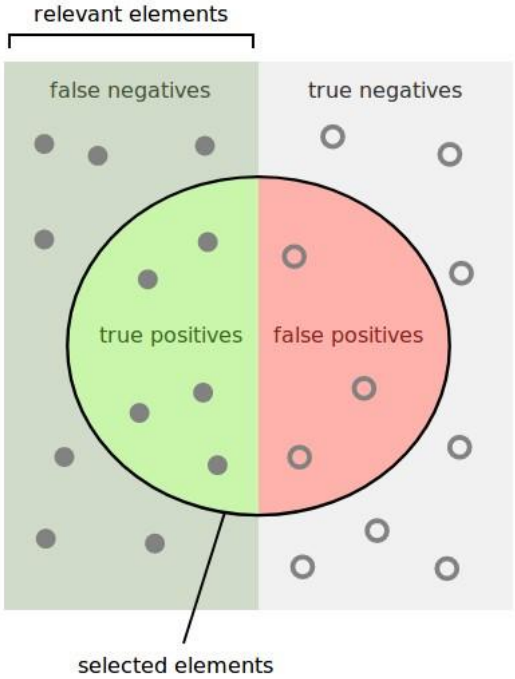
Falso positivo (error tipo 1)
test positivo, paciente no embarazada

Falso negativo (error tipo 2)
test negativo, paciente embarazada



Clasificación Binaria

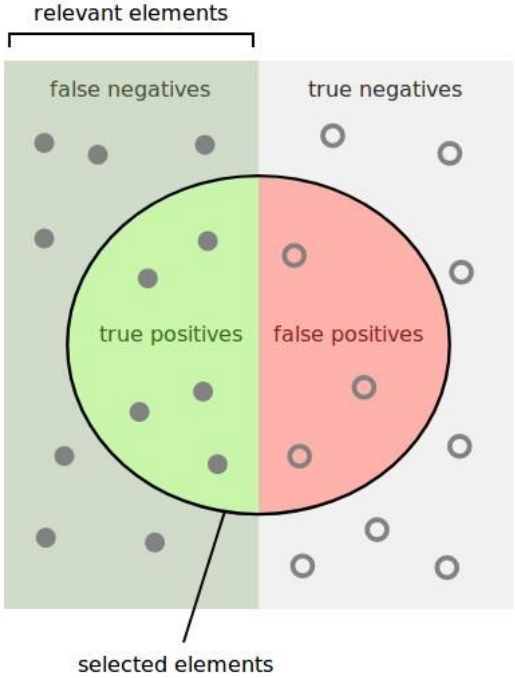
Cuando hacemos muchos tests...



Fuente: Wikipedia

Clasificación Binaria

Cuando hacemos muchos tests...



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

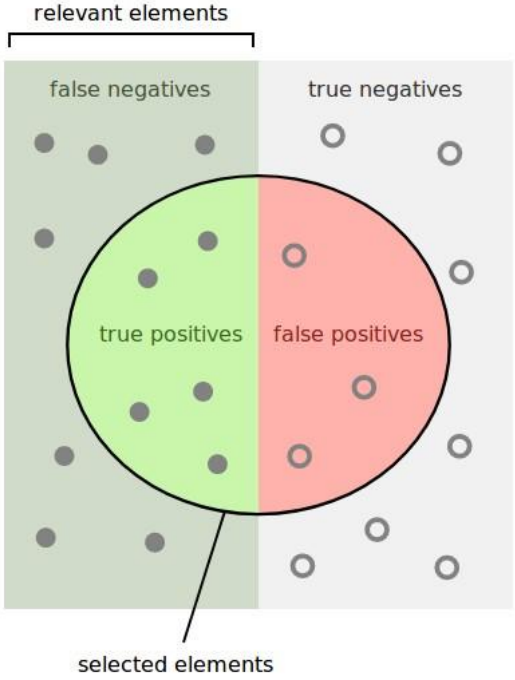
How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Fuente: Wikipedia

Clasificación Binaria

Cuando hacemos muchos tests...



Fuente: Wikipedia

How many selected items are relevant?

$$\text{Precision} = \frac{\text{Aciertos}}{\text{Aciertos + Falsos Positivos}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{Aciertos}}{\text{Aciertos + Falsos Negativos}}$$

$$Precisión = \frac{Aciertos}{Aciertos + Falsos\ Positivos}$$

$$Exhaustividad = \frac{Aciertos}{Aciertos + Falsos\ Negativos}$$

Para pensar...

proponer un test
100% exhaustivo y
otro 100% preciso.
¿Son útiles estos tests?

Para pensar...

...proponer un test 100% exhaustivo y otro 100% preciso. ¿Son útiles estos tests?

Problema: ninguna por sí sola alcanza para evaluar el desempeño del test, ya que precisión y exhaustividad compiten entre sí.

Objetivo: encontrar un compromiso entre ambas métricas.

Para pensar...

...proponer un test 100% exhaustivo y otro 100% preciso. ¿Son útiles estos tests?

Problema: ninguna por sí sola alcanza para evaluar el desempeño del test, ya que precisión y exhaustividad compiten entre sí.

Objetivo: encontrar un compromiso entre ambas métricas.

F-SCORE $\longrightarrow$ $F = 2 \times \frac{precisión \times exhaustividad}{precisión + exhaustividad}$

Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic

Clase Predicha

Clase Verdadera	Clase 1	Clase 2
Clase 1	Elementos de la clase 1 correctamente identificados	Elementos de la clase 1 identificados como clase 2
Clase 2	Elementos de la clase 2 identificados como clase 1	Elementos de la clase 2 correctamente identificados

Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic

Clase Predicha

Clase Verdadera

	Clase 1	Clase 2
Clase 1	Elementos de la clase 1 correctamente identificados	Elementos de la clase 1 identificados como clase 2
Clase 2	Elementos de la clase 2 identificados como clase 1	Elementos de la clase 2 correctamente identificados

¡Tiene toda la información que necesitamos!

Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic

Clase Predicha

Clase Verdadera	No Sobrevivieron	Sobrevivieron
No Sobrevivieron	513	110
Sobrevivieron	103	283

Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic

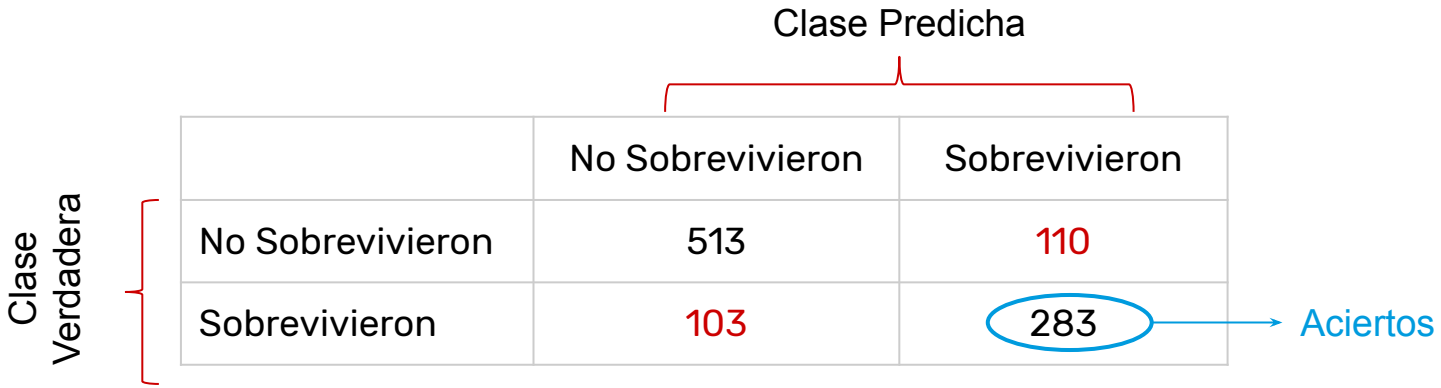
Clase Predicha

Clase Verdadera	No Sobrevivieron	Sobrevivieron
No Sobrevivieron	513	110
Sobrevivieron	103	283

Ejercicio: para la clase “sobrevivieron”, indicar Aciertos, Falsos Positivos, Falsos Negativos. Calcular Precisión y Exhaustividad.

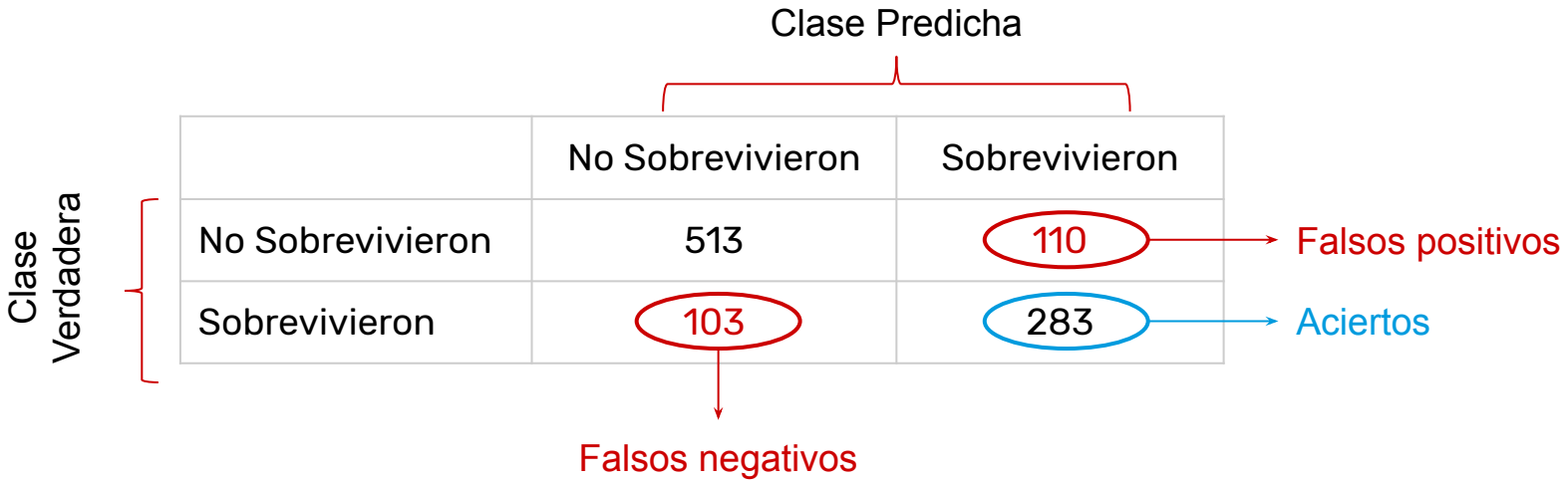
Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic



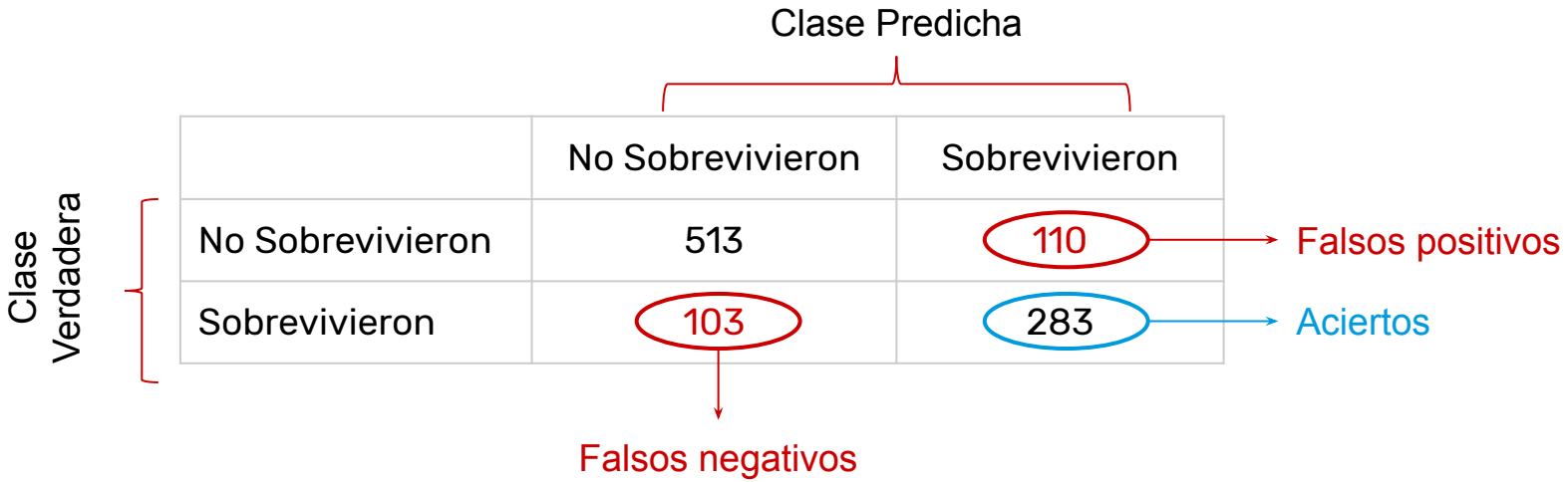
Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic



Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic

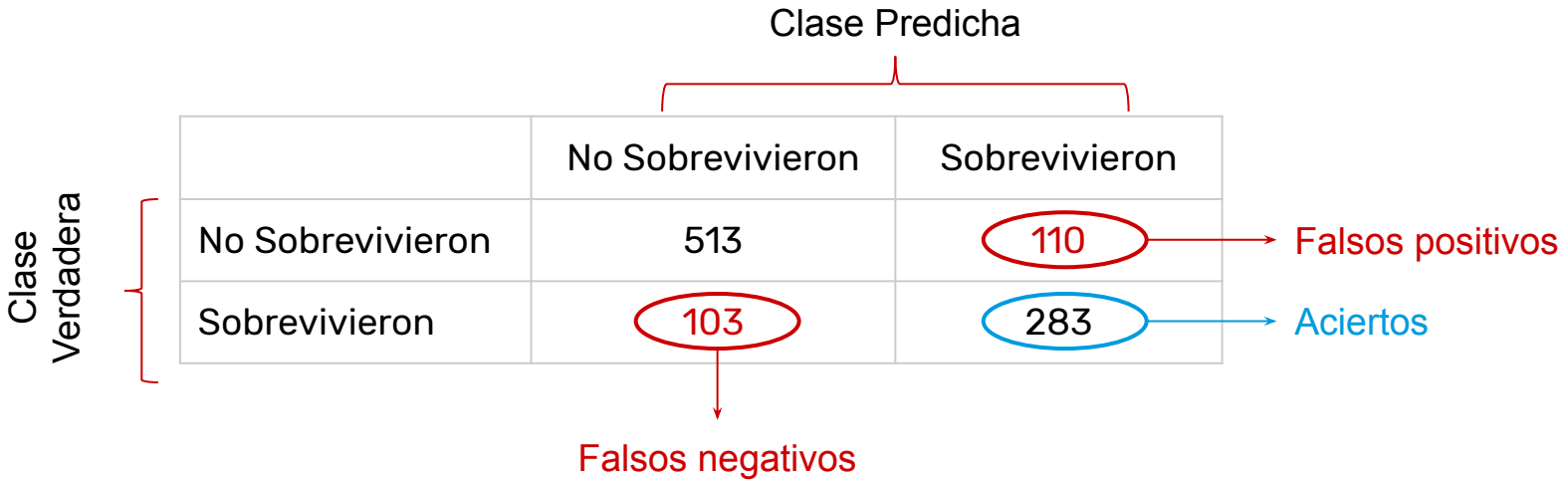


Precisión: TP/(TP + FP) = 283/(283+110) = 0.72

Exhaustividad: TP/(TP + FN) = 283/(283+103) = 0.73

Clasificación Binaria • Matriz de confusión

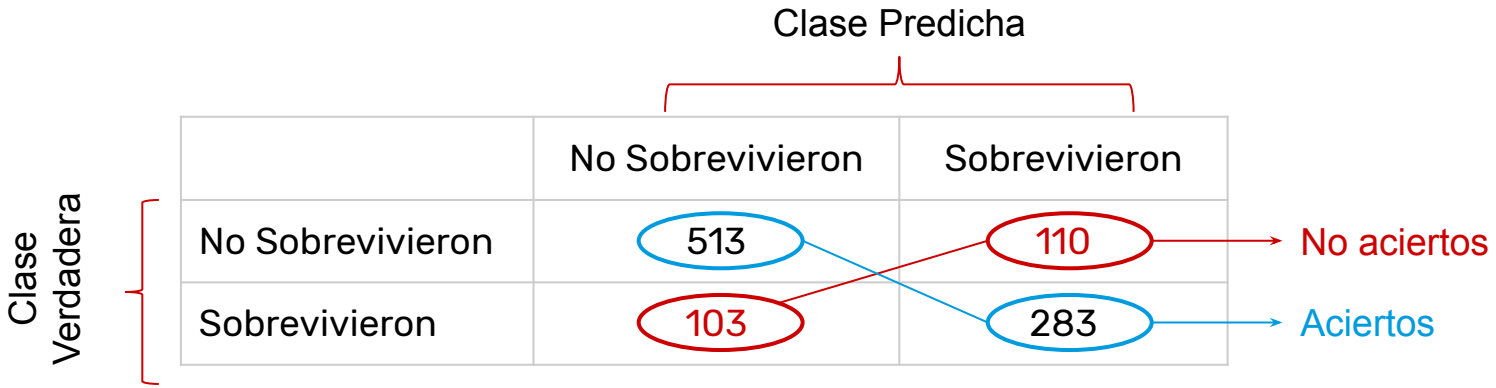
Ejemplo: Titanic



¿Y si en lugar de la clase “Sobrevivieron” nos interesaba “No Sobrevivieron”?

Clasificación Binaria • Matriz de confusión

Ejemplo: Titanic



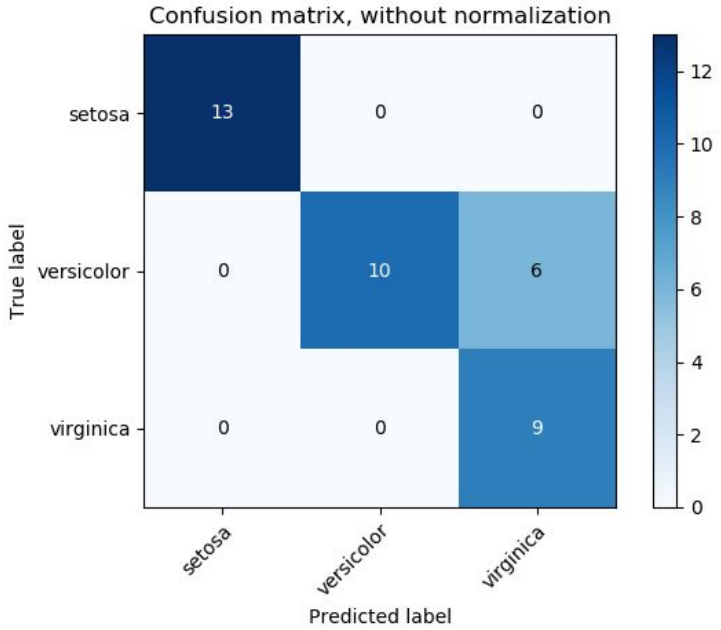
Exactitud = Aciertos/Total = (513 + 283)/(513 + 283 + 110 + 103) = 0.789

Clasificación Multiclase

¿Cómo se generalizan los conceptos?

Precisión y Exhaustividad: por clase

Exactitud: Sigue valiendo

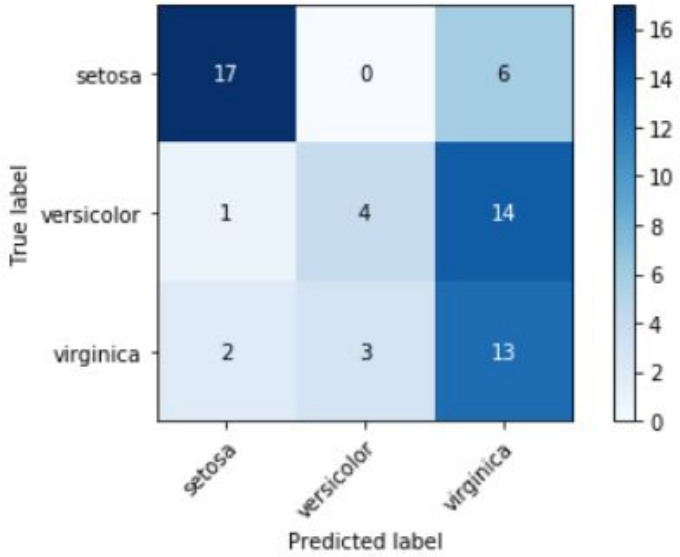


Clasificación Multiclase

Ejercicio

Dada la siguiente Matriz de Confusión:

- Elegir una clase e indicar Aciertos, Falsos Positivos, Falsos Negativos para esa clase.
- Calcular Precisión y Exhaustividad para la clase elegida.
- Calcular F-Score para la clase elegida
- Calcular la exactitud del modelo.



Clasificación Multiclase • Comentarios

Elegir la métrica correcta para nuestro problema es parte del trabajo que un Data Scientist tiene que hacer. A veces querremos favorecer precisión, otras veces exhaustividad.
¿En qué circunstancias les parece que preferimos una sobre otra?

Más adelante veremos:

- Curva ROC y área bajo la curva
- Funciones de costo

En scikit-learn:

https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics

Para la próxima

1. Completar notebooks atrasados

2. Pensar cómo aplicar las herramientas que vimos hasta ahora en el dataset que eligieron.

ACÁMICA