DATA SCIENCE

# KNN

## Vecinos más cercanos

Comisión/Clase/Versión/Autor

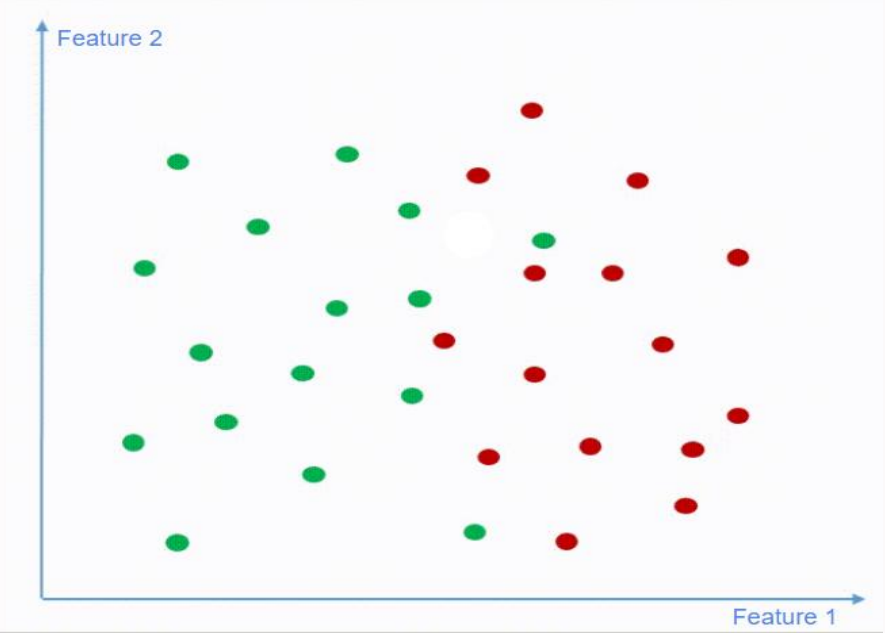
# KNN

Clasificación por criterio de k-vecinos más cercanos.

- Algoritmo de clasificación supervisado.
- Dada una serie de instancias ya etiquetadas, a qué grupo pertenece una nueva instancia.
- El concepto de distancia es la distancia euclidiana.
- Establecemos un radio alrededor de la nueva instancia y buscamos los k vecinos más cercanos.

# KNN

Datos de entrenamiento.
2 features. 31 instancias.

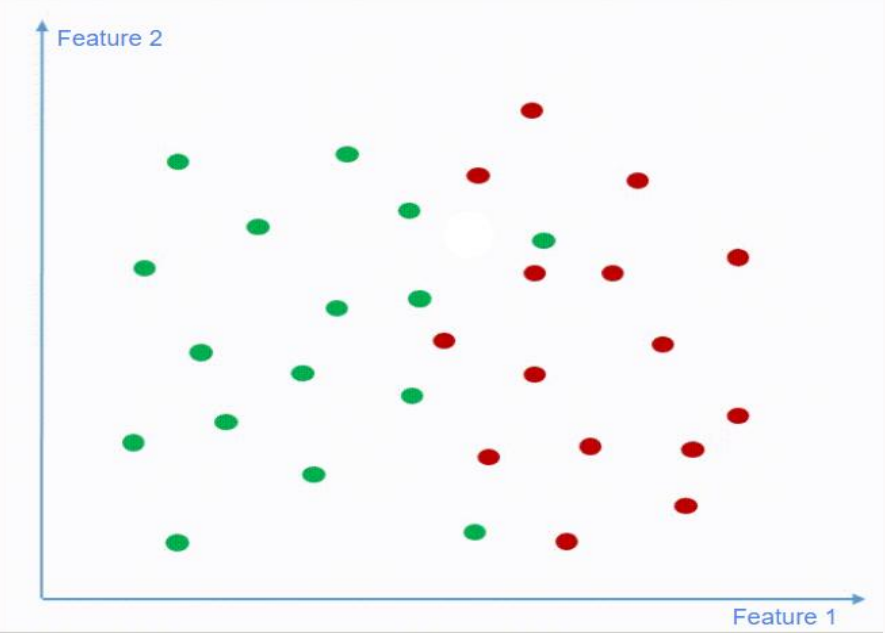


# KNN

Datos de entrenamiento.
2 features. 31 instancias.

Dada una instancia *imaginaria*:

¿Quién es mi vecina más cercana?

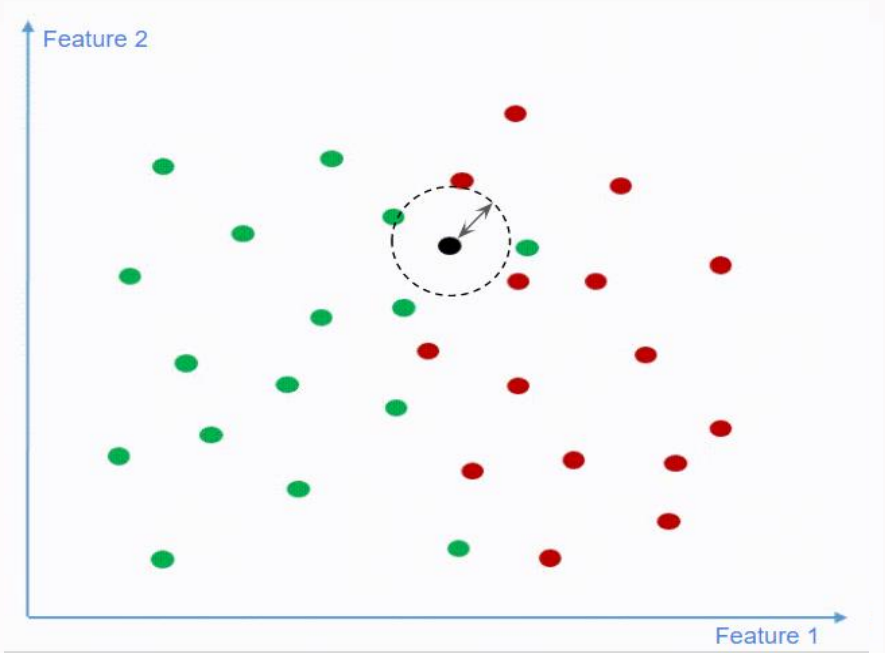


# KNN

Datos de entrenamiento.
2 features. 31 instancias.

Dada una instancia *imaginaria*:

¿Quién es mi vecina más cercana?

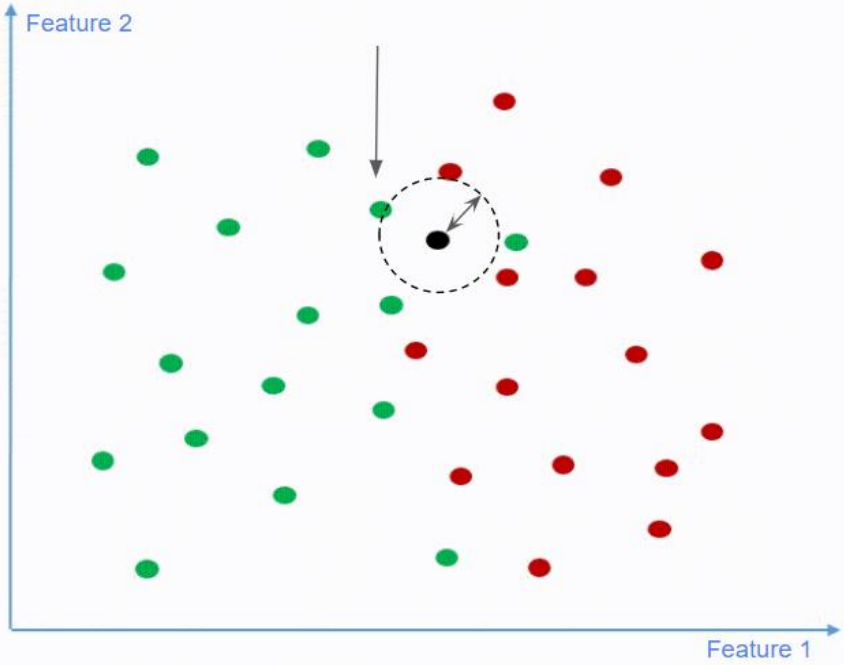


# KNN

Datos de entrenamiento.
2 features. 31 instancias.

Dada una instancia *imaginaria*:

¿Quién es mi vecina más cercana?
El punto verde



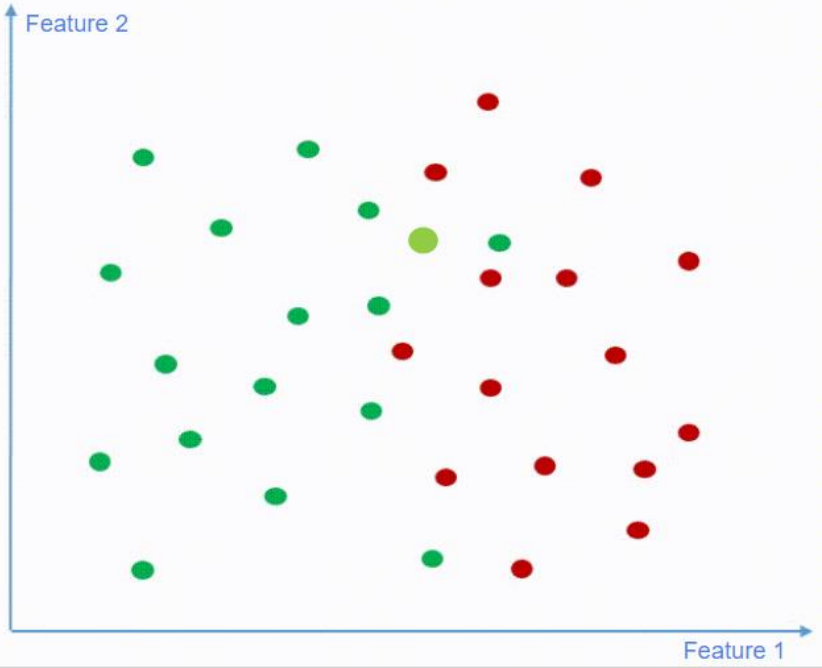
# KNN

Datos de entrenamiento.
2 features. 31 instancias.

Dada una instancia *imaginaria*:

¿Quién es mi vecina más cercana?
El punto verde.

**La nueva instancia es verde.**



# KNN

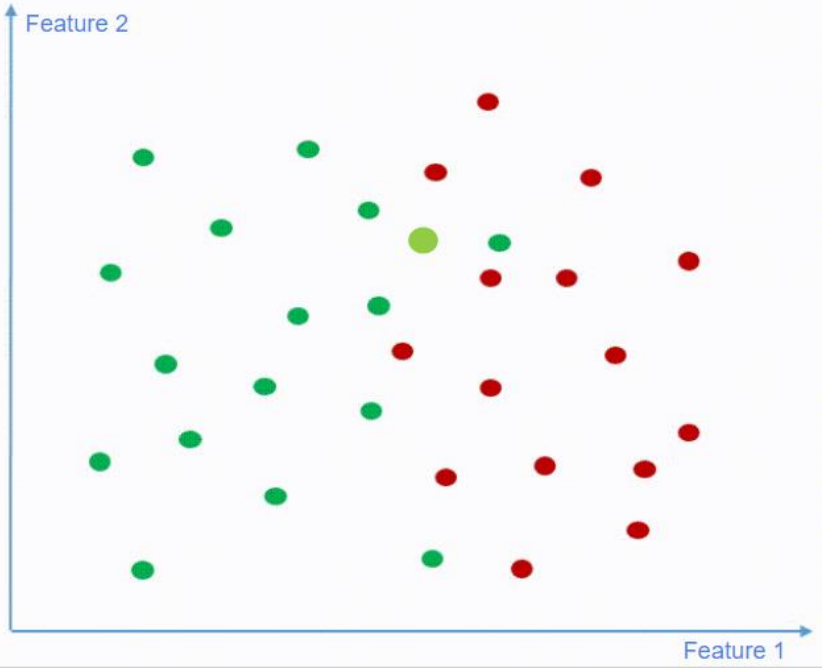
Datos de entrenamiento.
2 features. 31 instancias.

Dada una instancia *imaginaria*:

¿Quién es mi vecina más cercana?
El punto verde.

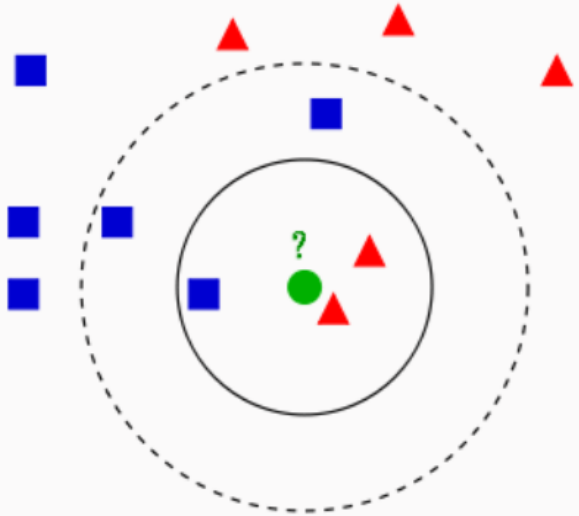
**La nueva instancia es verde.**

Repito para cada nuevo punto.



# Clasificador KNN

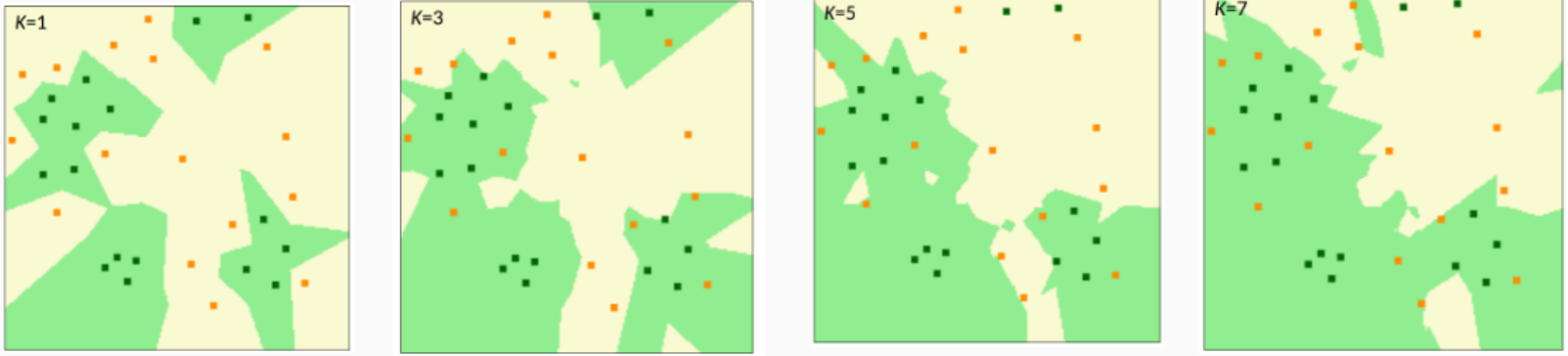
- Clasifica cada nuevo dato en el grupo que corresponda, según tenga **K** vecinos más cerca de un grupo o del otro.
- Calcula la ***distancia*** del elemento nuevo a cada uno de los existentes y ordena esas *distancias* para seleccionar a qué grupo al que pertenece.
- Selecciona la etiqueta (Y) qué más frecuente aparece en las K clases.




**HIPERPARÁMETROS**

- Número de vecinos relevantes K
- Distancia $d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

# **KNN.** Elegir K.



# Clasificador KNN

Dada una nueva instancia, devolver la clase más frecuente entre las K instancias más cercanas en D.
Hiperparámetros: K vecinos, distancia.

Ventajas:
- Simple,
- el entrenamientoes muy rápido.

Desventaja:
- la consulta es muy lenta
- el “modelo”ocupa mucho espacio en disco.

Para pensar:
La distancia se calcula con todos los atributos. ¿Qué pasa si algunos son irrelevantes?
¿Qué pasa si están en escalas muy distintas?

# Clasificador KNN

Dada una nueva instancia, devolver la clase más frecuente entre las K instancias más cercanas en D.
Hiperparámetros: K vecinos, distancia.

## 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.

# Clasificador KNN

Dada una nueva instancia, devolver la clase más frecuente entre las K instancias más cercanas en D.
Hiperparámetros: K vecinos, distancia.

## 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.

Algunas variantes:

- Distance-Weighted KNN
- Probabilistic KNN

# Métricas de evaluación para la clasificación

# Clasificación

¿Cómo evaluamos los resultados de la clasificación?

# Clasificación

Vamos a ver:

- Clasificación binaria
- Precisión/ Exhaustividad
- F-Score
- Matriz de confusión

# Clasificación Binaria

**Problema general:** Separar elementos en un conjunto de dos grupos bajo ciertas reglas.

Ejemplos:

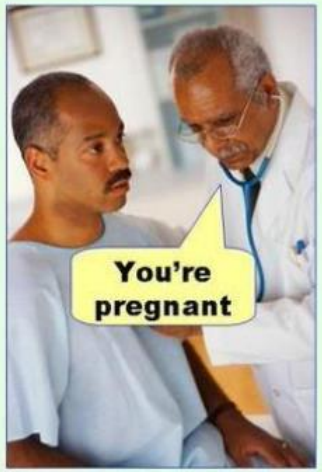
- Resultados de un test médico: enfermo/ no enfermo. Test de embarazo.
- Resultados de un examen académico: Aprobado/ no aprobado.
- Control de calidad.

# Clasificación Binaria

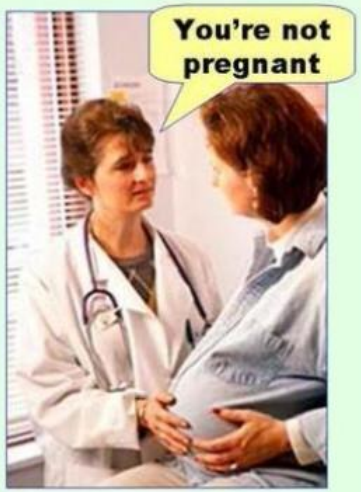
¿Qué puede pasar con **un** test? Ejemplo: Test de Embarazo

- **Verdadero Positivo (Acierto)**: Test positivo, paciente embarazada.
- **Falso positivo**: Test positivo, paciente no-embarazada.
- **Falso Negativo**: Test negativo, paciente embarazada.
- **Verdadero negativo**: Test negativo, paciente no-embarazada.

**Type I error**
(false positive)

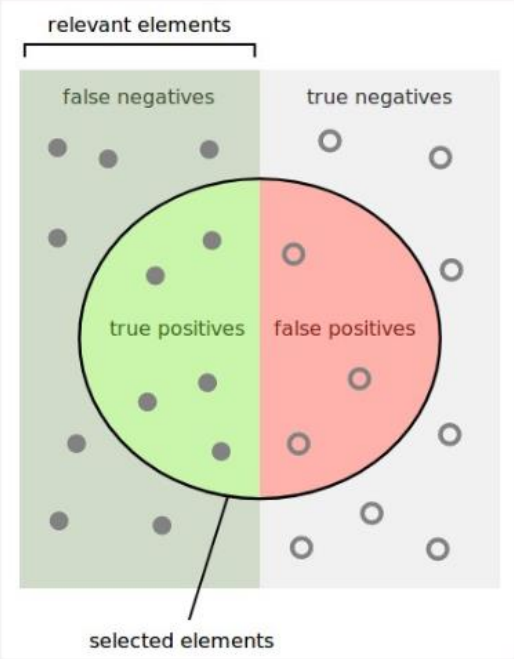



**Type II error**
(false negative)



# Clasificación Binaria

Si hacemos muchos tests




How many selected items are relevant?

$$\text{Precision} = \frac{\text{●}}{\text{◐}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{◔}}{\text{◑}}$$

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

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

# Clasificación Binaria

Para pensar: proponer un test **100% exhaustivo** y otro **100% preciso**.
¿Son útiles estos test?

# Clasificación Binaria

Para pensar: proponer un test **100% exhaustivo** y otro **100% preciso**. ¿Son útiles estos test?

**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**

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

# Clasificación Binaria

## Matriz de Confusión

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

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

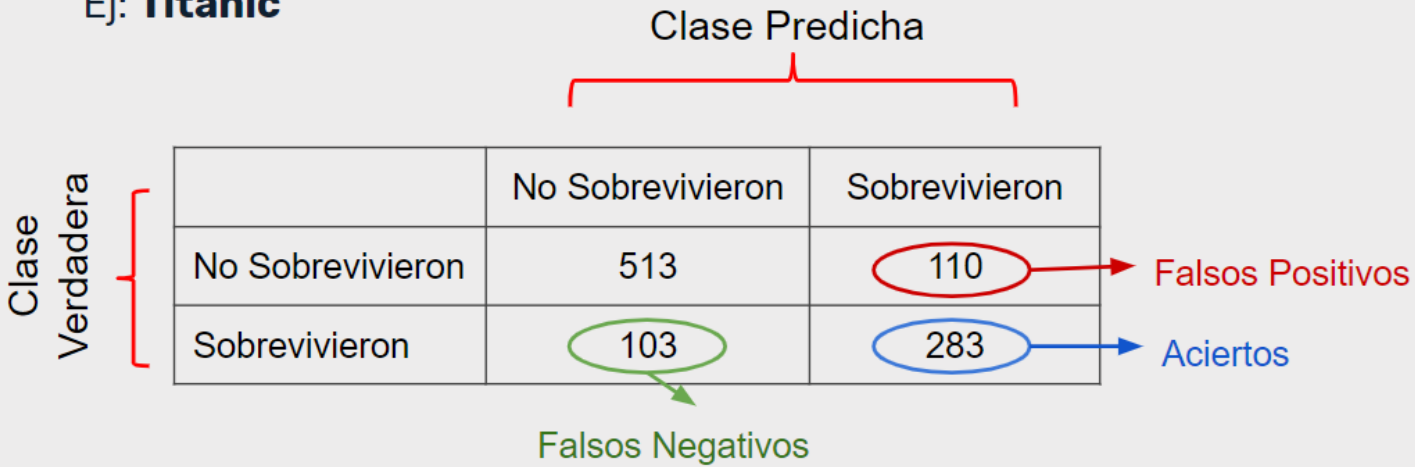
Matriz de Confusión
Ej: **Titanic**

Clase Predicha

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

# Clasificación Binaria

Matriz de Confusión
Ej: **Titanic**




Precisión: TP/(TP + FP) = 283/(283+110) = 0.72
Exhaustividad: TP/(TP + FN) = 283/(283+103) = 0.73

# Clasificación Binaria

Matriz de Confusión
Ej: **Titanic**

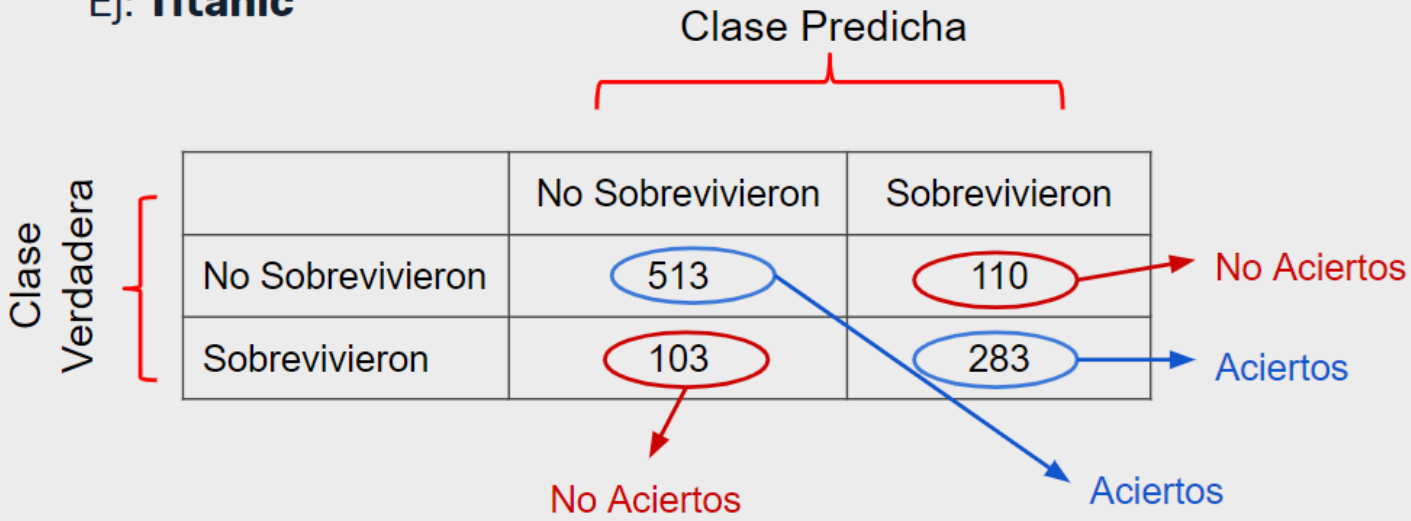
Clase Predicha

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

**¿Y si en lugar de la clase “Sobrevivieron” nos interesara “No Sobrevivieron? ¿Si en lugar de Clase queremos una métrica más general?**

# Clasificación Binaria

Matriz de Confusión
Ej: **Titanic**




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

# Clasificación Binaria

**F1 - Score**: métrica de clasificación binaria que tiene en cuenta tanto la precisión como la exhaustividad.

Puede ser una buena métrica para evaluar clases con distribuciones desbalanceadas, o cuando hay un gran número de negativos reales.

$$F_1 = \left( \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} \right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

# Clasificación Multiclase

¿Cómo se generalizan los conceptos?

**Precision y Exhaustividad**: por clase

**Exactitud**: Sigue valiendo