# Introducción al Machine Learning para políticas públicas en Python Parte 1

*Reinel Tabares Soto*
*Harold Brayan Arteaga Arteaga*

UTraining Entrenamiento en Innovación y Tecnología
UNIVERSIDAD DE ANTIOQUIA
Universidad de Caldas
ubicua Technology

# Agenda

## Contenido del módulo 3

- Análisis exploratorio de datos - EDA.
- K-Nearest Neighbors (modelo base para sección práctica).
- Métricas de medición de rendimiento.
- Transformación de características.
- Balance de clases.
- Reducción de dimensionalidad.

# Análisis exploratorio de datos - EDA.

# Algunas recomendaciones para realizar un EDA

**Análisis exploratorio de datos - EDA.**

- Verificar la cantidad de clases
- Verificar datos faltantes y outliers
- Verificar el tipo de datos que componen el dataset
- Analizar histogramas de las features
- Analizar correlaciones entre las features con mapas de calor
- Analizar los gráficos entre features con pairplot
- Analizar los gráficos como boxplot, catplot (ejemplo: bar y violin), entre los labels y features seleccionadas

# Conjuntos de entrenamiento y prueba

**Análisis exploratorio de datos - EDA.**

- Conocidos en inglés como conjuntos (sets) de train y test
- No podemos entrenar y probar el modelo con el mismo conjunto de datos
- Se suele separar un conjunto aleatorio de datos para probar (evaluar) los resultados del modelo
- El resto de los datos son usados para entrenar
- Una separación común es 30% para el conjunto de prueba y 70% para el conjunto de entrenamiento

# Dataset sobre el diagnóstico de cáncer

**Análisis exploratorio de datos - EDA.**

- **Descripción dataset :**
  https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

- **Descargar dataset:**
  https://github.com/stedy/Machine-Learning-with-R-datasets/blob/master/wisc_bc_data.csv

Carpeta con la Data requerida por los Notebooks link

# A programar

Link

**Análisis exploratorio de datos - EDA.**

- **Librerias**
- **Lectura del dataset**
- **Eliminar columnas innecesarias del dataset**
- **Análisis exploratorio de datos (Exploratory data analysis - EDA)**
- **División en datos de entrenamiento y testing**
- K-Nearest Neighbors
- Métricas
- Métricas bonitas
- KNN con preprocesamiento de las features
- Balance de clases
- Reducción de dimensionalidad

# K-Nearest Neighbors (modelo base para sección práctica).

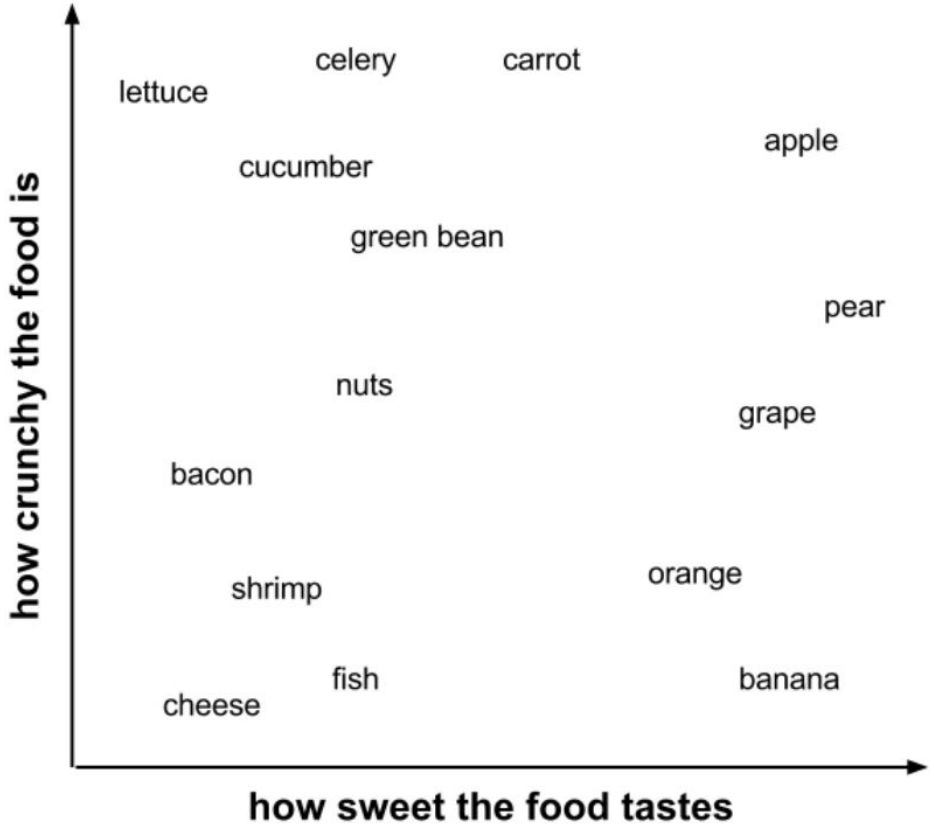
# Clasificando comidas

**K-Nearest Neighbors (modelo base para sección práctica).**

| ingredient | sweetness | crunchiness | food type |
|---|---|---|---|
| apple | 10 | 9 | fruit |
| bacon | 1 | 4 | protein |
| banana | 10 | 1 | fruit |
| carrot | 7 | 10 | vegetable |
| celery | 3 | 10 | vegetable |
| cheese | 1 | 1 | protein |

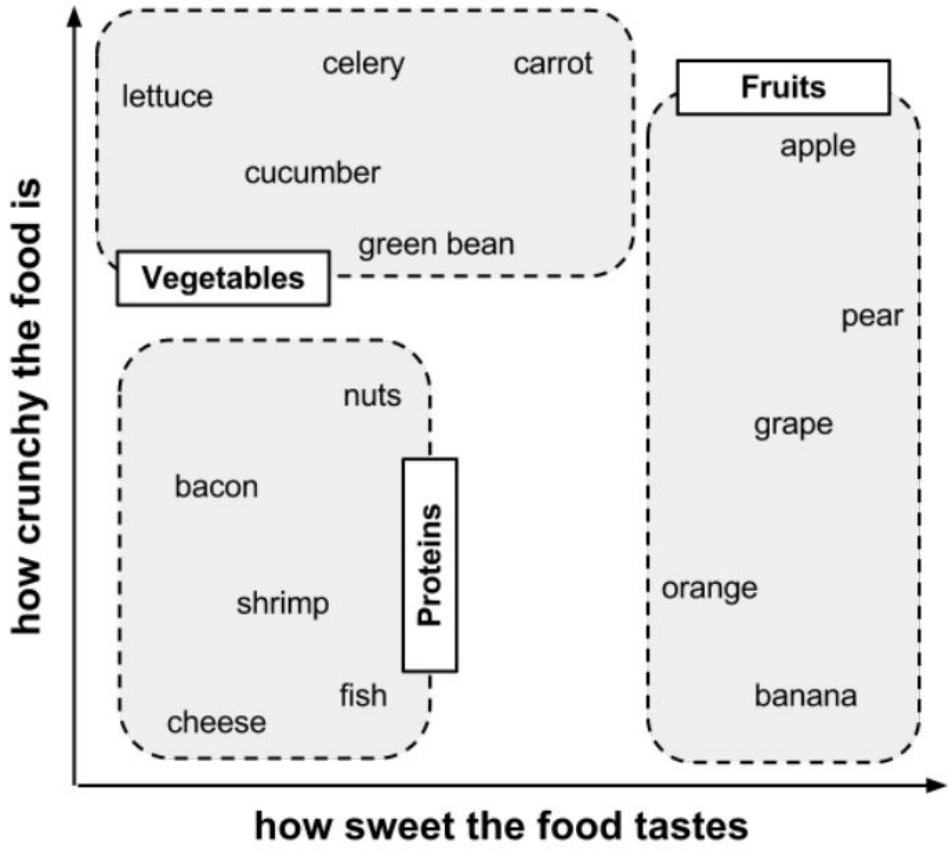
# Clasificando comidas

**K-Nearest Neighbors (modelo base para sección práctica).**



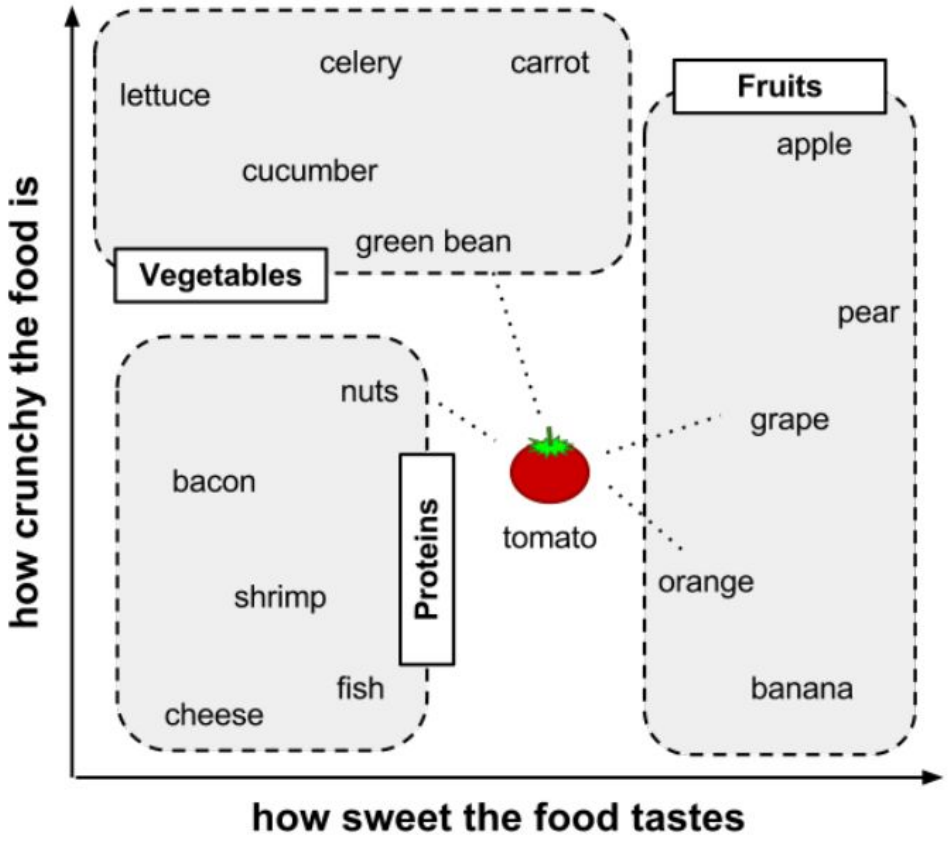
# Clasificando comidas

**K-Nearest Neighbors (modelo base para sección práctica).**



# Clasificando comidas

**K-Nearest Neighbors (modelo base para sección práctica).**



# Distancias

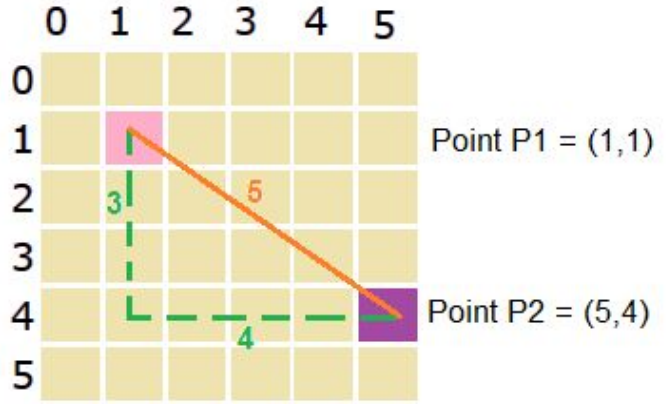
**K-Nearest Neighbors (modelo base para sección práctica).**

- Distancia Euclidiana

$$dist(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

- Distancia Manhattan

$$dist(x,y) = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|$$




Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Manhattan distance = $|5-1| + |4-1| = 7$

# Clasificando comidas

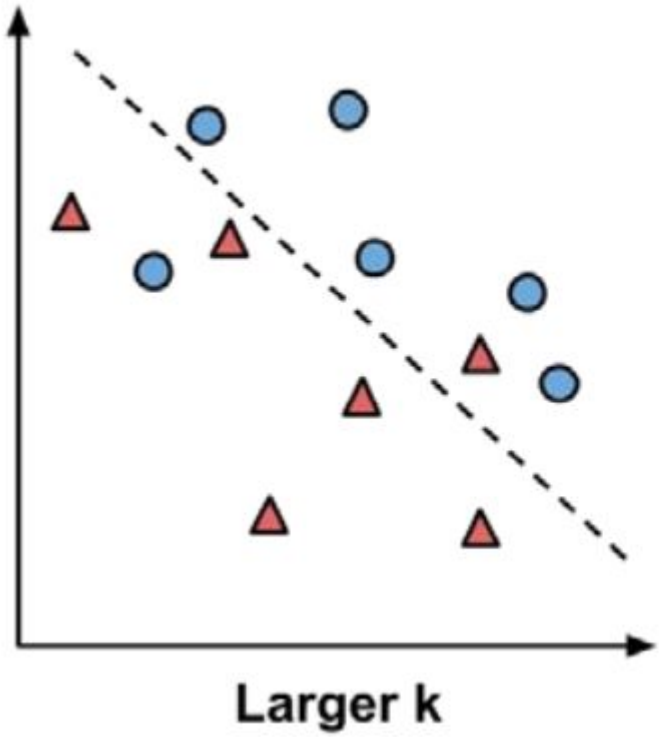
**K-Nearest Neighbors (modelo base para sección práctica).**

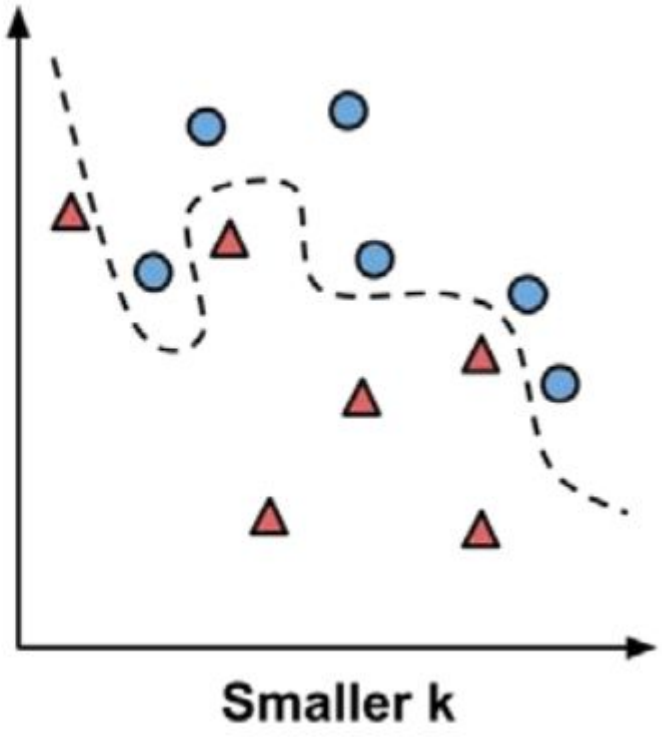
- Tomate (sweetness = 6, crunchiness = 4)

| ingredient | sweetness | crunchiness | food type | distance to the tomato |
|---|---|---|---|---|
| grape | 8 | 5 | fruit | *sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2* |
| green bean | 3 | 7 | vegetable | *sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2* |
| nuts | 3 | 6 | protein | *sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6* |
| orange | 7 | 3 | fruit | *sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4* |

# Eligiendo un K

**K-Nearest Neighbors (modelo base para sección práctica).**



# K-Nearest Neighbors

**K-Nearest Neighbors (modelo base para sección práctica).**

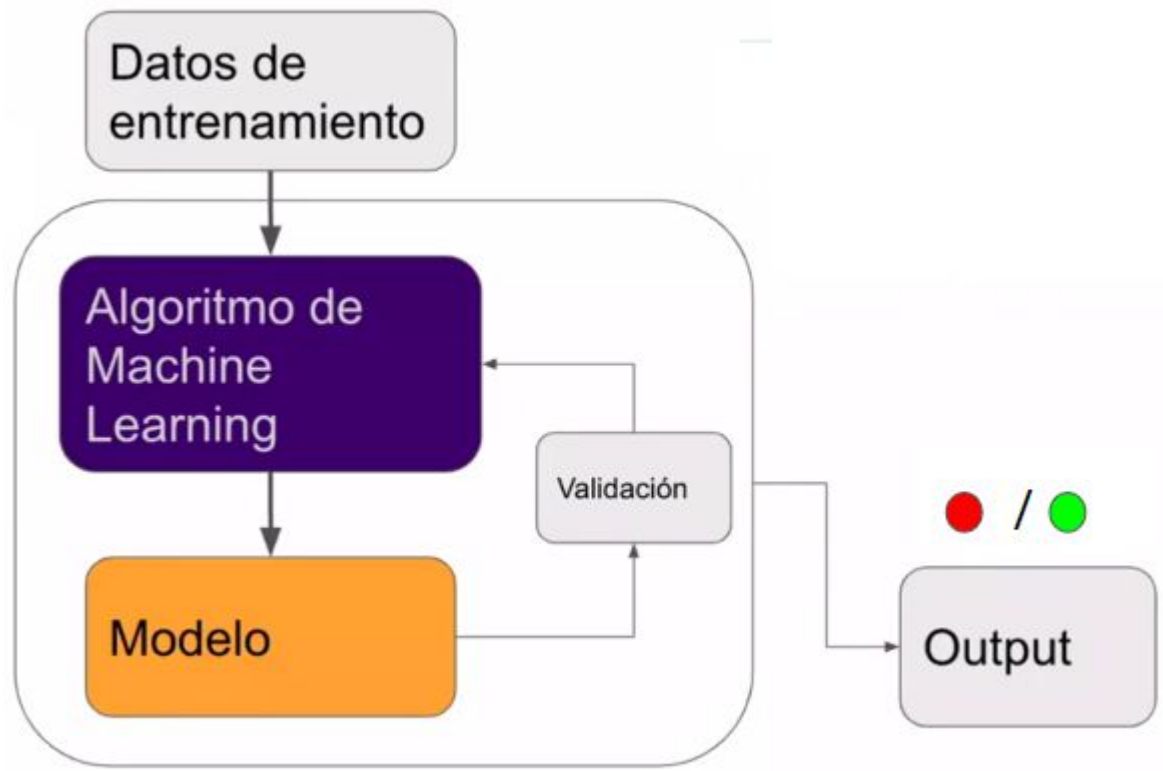
- Ventajas
  - Simple y efectivo
  - No se asume nada sobre los datos
  - No hay “entrenamiento”

- Desventajas
  - Sensible al valor del K
  - Etapa de clasificación lenta
  - Requiere mucha memoria

# Métricas de medición de rendimiento.

# Machine Learning

## Métricas de medición de rendimiento.




# Plomo en Chicago

**Métricas de medición de rendimiento.**

- Dados:
  - Información sobre una construcción
    - barrio
    - año de construcción
- Queremos saber:
  - ¿Tiene pintura con plomo esta casa?

# Historial de inspecciones (datos de entrenamiento)

**Métricas de medición de rendimiento.**

| Fecha inspección | Año de construcción | Ubicación | ¿Tiene plomo? |
|---|---|---|---|
| 2017-06-15 | 1947 | Englewood | Sí |
| 2013-11-23 | 2012 | Chinatown | No |
| 2016-03-24 | 1956 | Downtown | No |
| 2012-01-01 | 1974 | Chinatown | Sí |
| ... | ... | ... | ... |

# “Si año de construcción < 2000, entonces tiene plomo”

**Métricas de medición de rendimiento.**

| Fecha inspección | Año de construcción | Ubicación | ¿Tiene plomo? |
|---|---|---|---|
| 2017-06-15 | 1947 | Englewood | Sí |
| 2013-11-23 | 2012 | Chinatown | No |
| 2016-03-24 | 1956 | Downtown | No |
| 2012-01-01 | 1974 | Chinatown | Sí |
| ... | ... | ... | ... |

# “Si año de construcción < 2000, entonces tiene plomo”

**Métricas de medición de rendimiento.**

| Fecha inspección | Año de construcción | Ubicación | ¿Tiene plomo? |
|---|---|---|---|
| 2017-06-15 | 1947 | Englewood | Sí |
| 2013-11-23 | 2012 | Chinatown | No |
| 2016-03-24 | 1956 | Downtown | No |
| 2012-01-01 | 1974 | Chinatown | Sí |
| ... | ... | ... | ... |

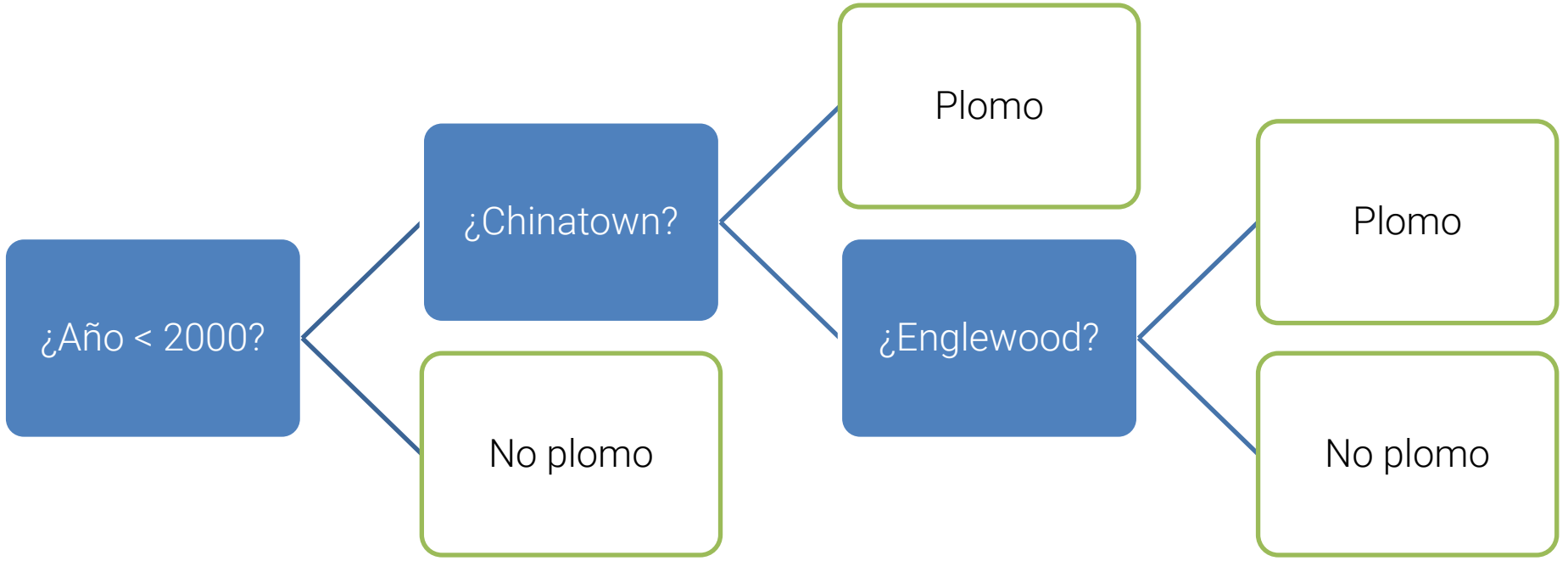
# “Si año de construcción < 2000 y está en Chinatown o en Englewood, entonces tiene plomo”

**Métricas de medición de rendimiento.**

| Fecha inspección | Año de construcción | Ubicación | ¿Tiene plomo? |
|---|---|---|---|
| 2017-06-15 | 1947 | Englewood | Sí |
| 2013-11-23 | 2012 | Chinatown | No |
| 2016-03-24 | 1956 | Downtown | No |
| 2012-01-01 | 1974 | Chinatown | Sí |
| … | … | … | … |

# Árbol de decisión

**Métricas de medición de rendimiento.**



# Predicción

**Métricas de medición de rendimiento.**

| Año de construcción | Ubicación | ¿Tiene plomo? |
|---|---|---|
| 1947 | Chinatown | ? |
| 2005 | Downtown | ? |
| 1960 | Englewood | ? |
| 1999 | Chinatown | ? |

# Predicción

**Métricas de medición de rendimiento.**

| Año de construcción | Ubicación | ¿Tiene plomo? |
|---|---|---|
| 1947 | Chinatown | Si |
| 2005 | Downtown | No |
| 1960 | Englewood | Si |
| 1999 | Chinatown | Si |

# Predicción

**Métricas de medición de rendimiento.**

| Año de construcción | Ubicación | ¿Tiene plomo? (predicción) | ¿Tiene plomo? (real) |
|---|---|---|---|
| 1947 | Chinatown | Si | Si |
| 2005 | Downtown | No | No |
| 1960 | Englewood | Si | Si |
| 1999 | Chinatown | Si | No |

# Evaluación

**Métricas de medición de rendimiento.**

| Predicción<br><br>Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| Tiene plomo | Verdadero Positivo (VP o TP) | Falso Negativo (FN) |
| No tiene plomo | Falso Positivo (FP) | Verdadero Negativo (VN o TN) |

# Evaluación

**Métricas de medición de rendimiento.**

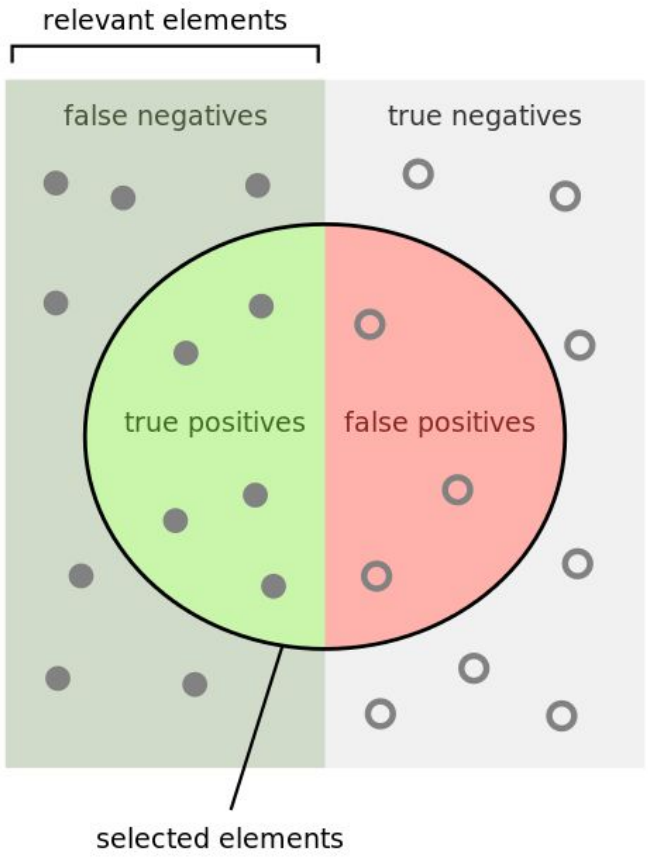
| Año de construcción | Ubicación | ¿Tiene plomo? (predicción) | ¿Tiene plomo? (real) |
|---|---|---|---|
| 1947 | Chinatown | Si | Si |
| 2005 | Downtown | No | No |
| 1960 | Englewood | Si | Si |
| 1999 | Chinatown | Si | No |

| Predicción<br>Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| Tiene plomo | 2 | 0 |
| No tiene plomo | 1 | 1 |

# Evaluación

## Métricas de medición de rendimiento.

| Predicción / Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| Tiene plomo | Verdadero Positivo (VP o TP) | Falso Negativo (FN) |
| No tiene plomo | Falso Positivo (FP) | Verdadero Negativo (VN o TN) |




How many selected items are relevant?

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

How many relevant items are selected?

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

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

**Accuracy:** Es cuantas casas con plomo y sin plomo predijo correctamente del total

**Precision:** Es cuantas casas que yo dije que tenían plomo que realmente tienen plomo con respecto las que yo dije que tenían plomo

**Recall (True Positive Rate):** Es cuantas casas con plomo acerté de todas las casas con plomo que había.

# Evaluación

**Métricas de medición de rendimiento.**

| Predicción<br><br>Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| Tiene plomo | 2 | 0 |
| No tiene plomo | 1 | 1 |

Precision = 2/3 = 66.6%
Recall = 2/2 = 100%
Accuracy = 3/4 = 75%

# Evaluación

**Métricas de medición de rendimiento.**

| Predicción / Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| Tiene plomo | 500 (TP) | 1000 (FN) |
| No tiene plomo | 500 (FP) | 10000 (TN) |

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

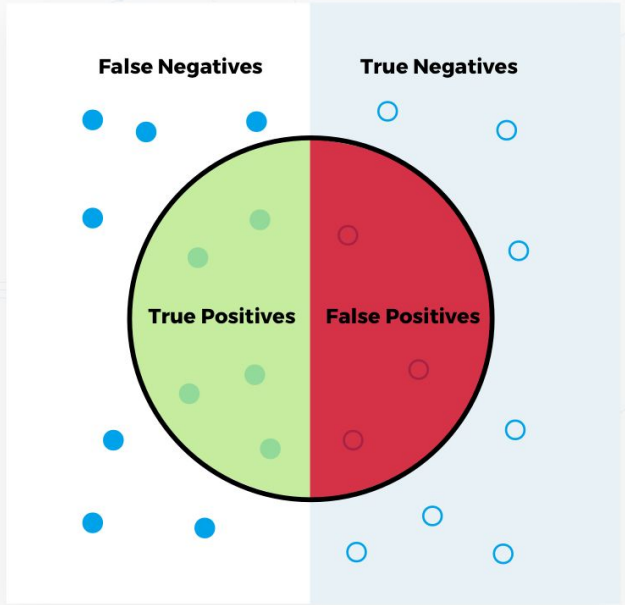
| Predicción / Realidad | Tiene Covid | No tiene Covid |
|---|---|---|
| Tiene Covid | 2000 (TP) | 10 (FN) |
| No tiene Covid | 500 (FP) | 2000 (TN) |

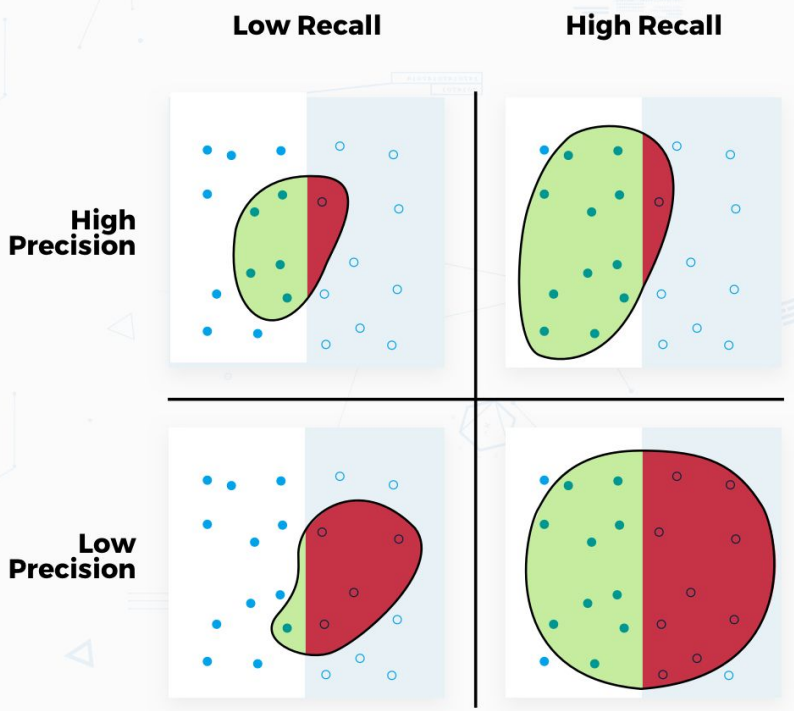
Precision = 500/1000 = 50%
Recall = 500/1500 = 33.3%
Accuracy = 10500/12000 = 87.5%

Precision = 2000/2500 = 80%
Recall = 2000/2010 = 99.5%
Accuracy = 4000/4510 = 88.7%

# Evaluación

## Métricas de medición de rendimiento.




**Accuracy:** Es cuantas casas con plomo y sin plomo predijo correctamente del total

**Precision:** Es cuantas casas que yo dije que tenían plomo que realmente tienen plomo con respecto las que yo dije que tenían plomo

**Recall:** Es cuantas casas con plomo acerté de todas las casas con plomo que había.

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

# Desbalance vs sobreentrenamiento

## Métricas de medición de rendimiento.

| ¿Tiene plomo? (predicción) | ¿Tiene plomo? (real) |
|---|---|
| NO | NO |
| NO | NO |
| NO | NO |
| NO | NO |
| NO | NO |
| NO | NO |
| SI | NO |
| NO | NO |
| NO | NO |
| NO | SI |

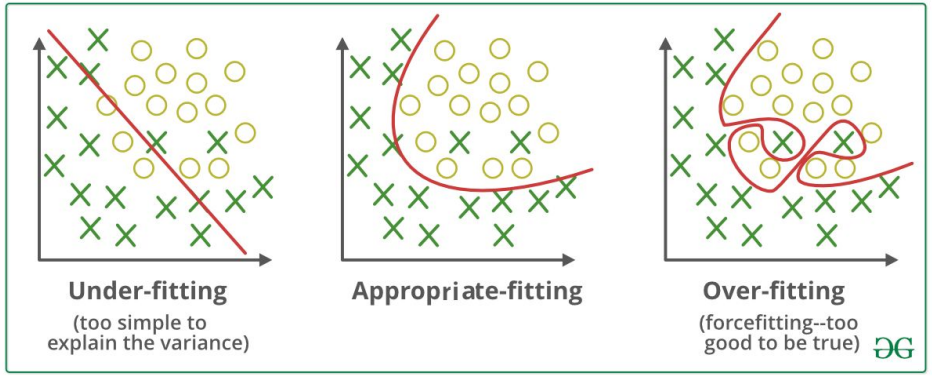
| Predicción / Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| Tiene plomo | 0 (TP) | 1 (FN) |
| No tiene plomo | 1 (FP) | 8 (TN) |

Precision = 0/1 = 0%
Recall = 0/1 = 0%
Accuracy = 8/10 = 80%

$$\text{Precision} = \frac{tp}{tp + fp}$$

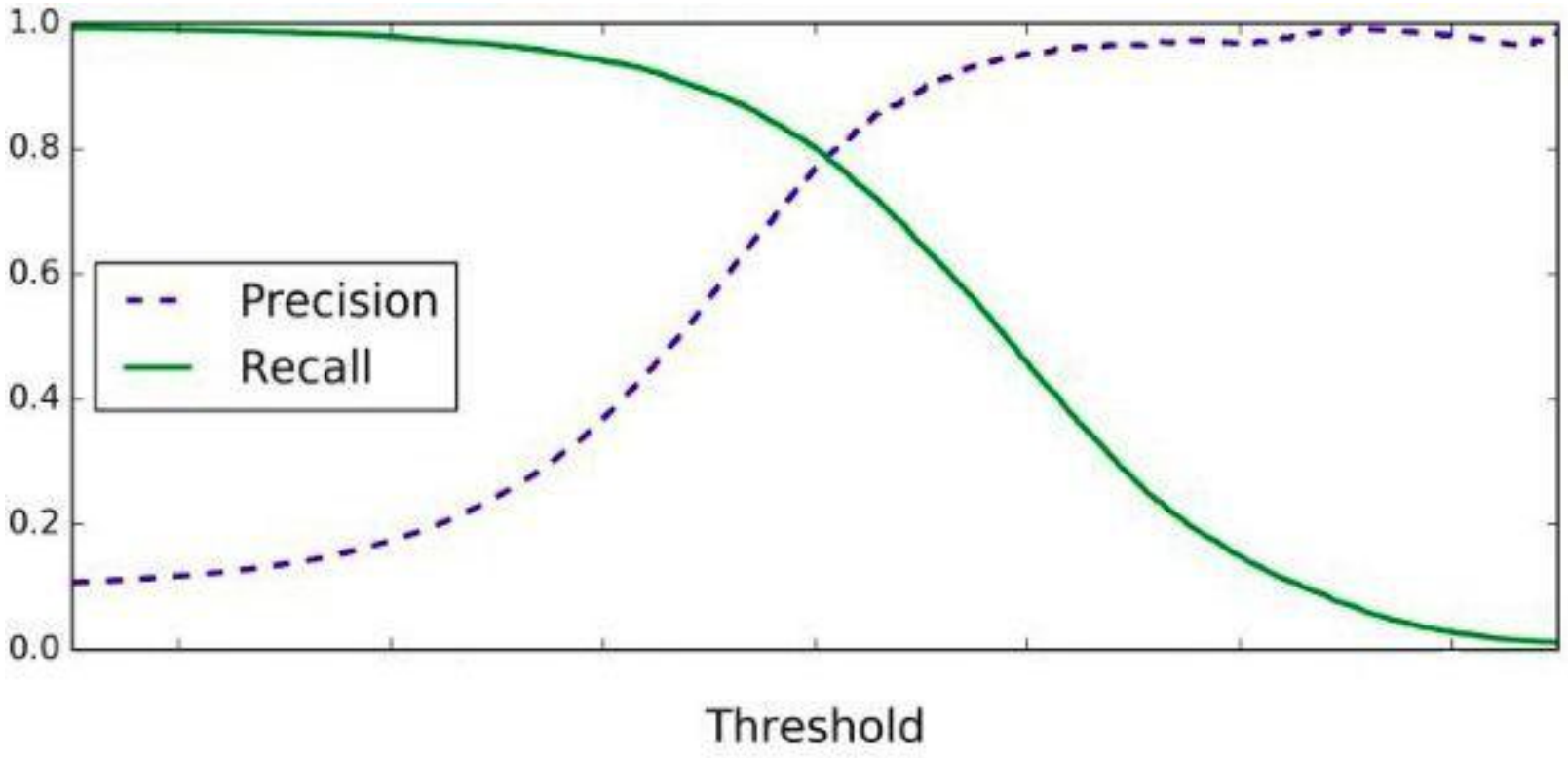
$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$



# Precision y Recall

**Métricas de medición de rendimiento.**



# F-measure

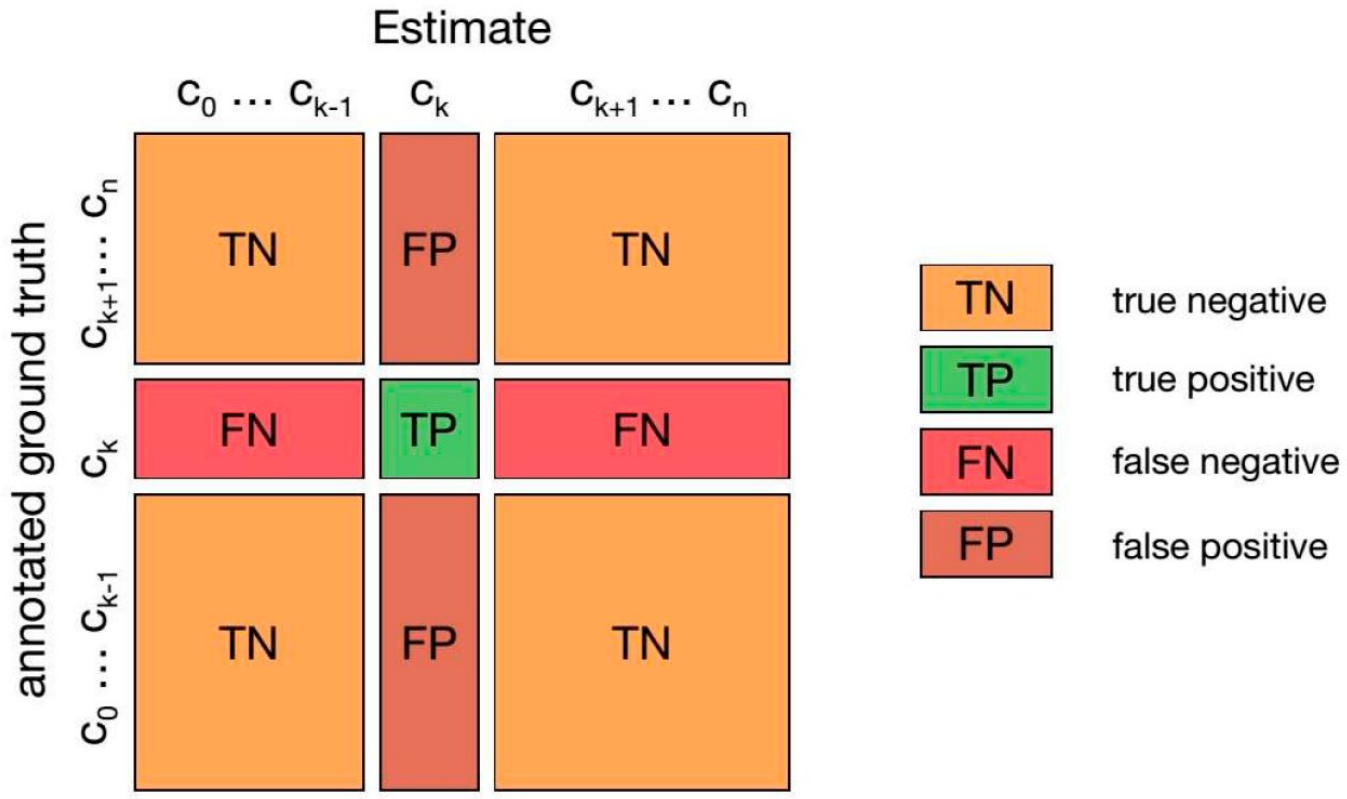
**Métricas de medición de rendimiento.**

- También conocido como F-score
- Media armónica de precision y recall
- Precision y recall son igual de importantes

$$F - measure = 2\frac{Precision * Recall}{Precision + Recall}$$

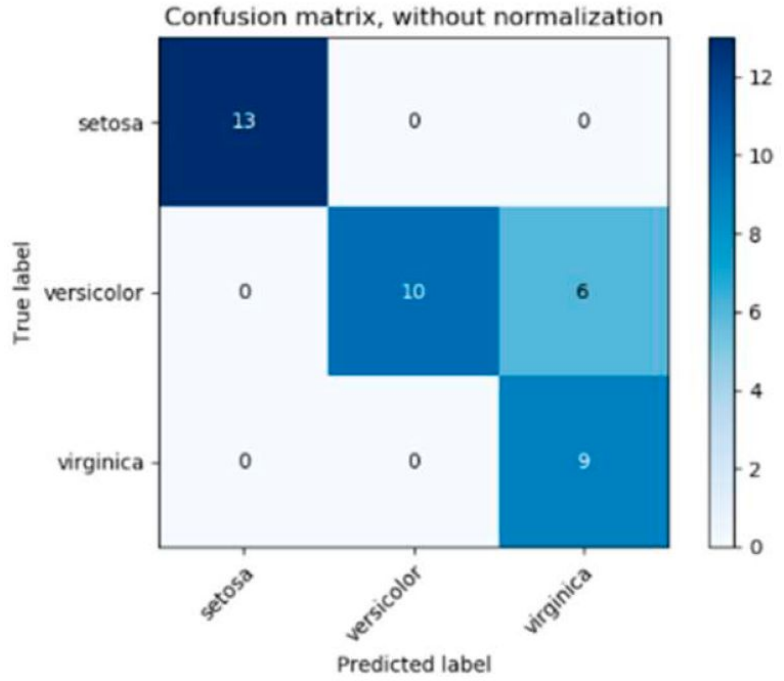
# Multiclase

## Métricas de medición de rendimiento.









# Multiclase

**Métricas de medición de rendimiento.**

| True Class | Predicted Class | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | 8 | 2 | 0 |
| 2 | 1 | 9 | 0 |
| 3 | 1 | 2 | 7 |

| True Class | Predicted Class | |
|---|---|---|
| | 1 – Yes | 2 & 3 – No |
| 1 – Yes | TP - 8 | FN -2 |
| 2 & 3 – No | FP - 2 | TN - 18 |

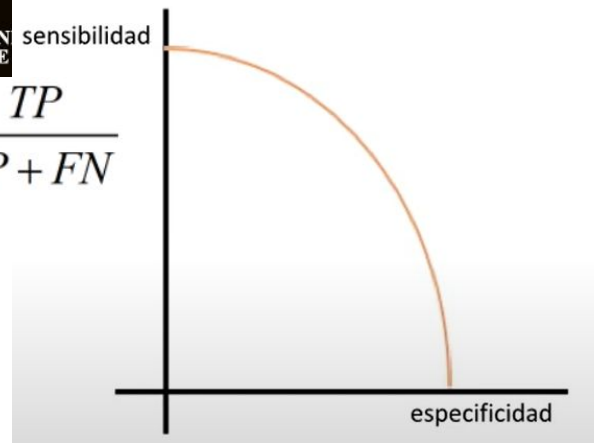
| True Class | Predicted Class | |
|---|---|---|
| | 2 – Yes | 1 & 3 – No |
| 2 – Yes | TP - 9 | FN - 1 |
| 1 & 3 – No | FP - 4 | TN - 16 |

| True Class | Predicted Class | |
|---|---|---|
| | 3 – Yes | 1 & 2 – No |
| 3 – Yes | TP - 7 | FN - 3 |
| 1 & 2 – No | FP - 0 | TN- 20 |

# Curvas ROC: Sensibilidad, especificidad

## Métricas de medición de rendimiento.

| Predicción / Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| **Tiene plomo** | Verdadero Positivo (TP) | Falso Negativo (FN) |
| **No tiene plomo** | Falso Positivo (FP) | Verdadero Negativo (TN) |

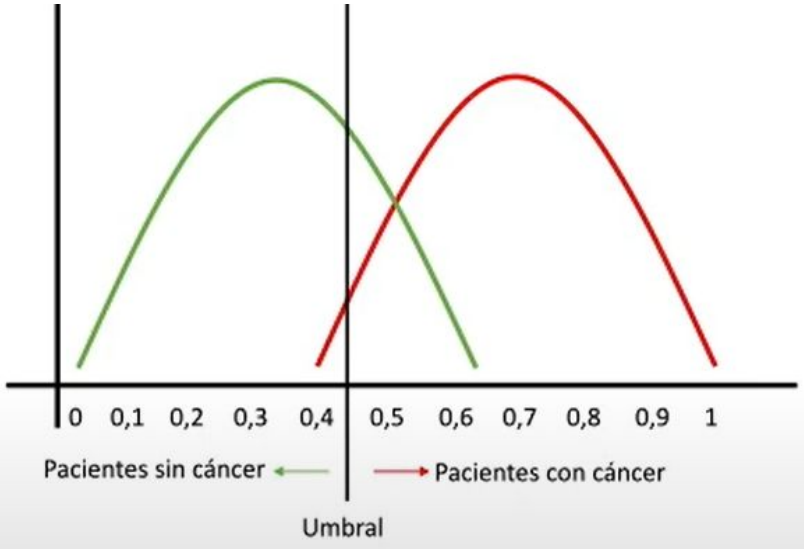



$$sensibilidad = \frac{TP}{TP + FN}$$

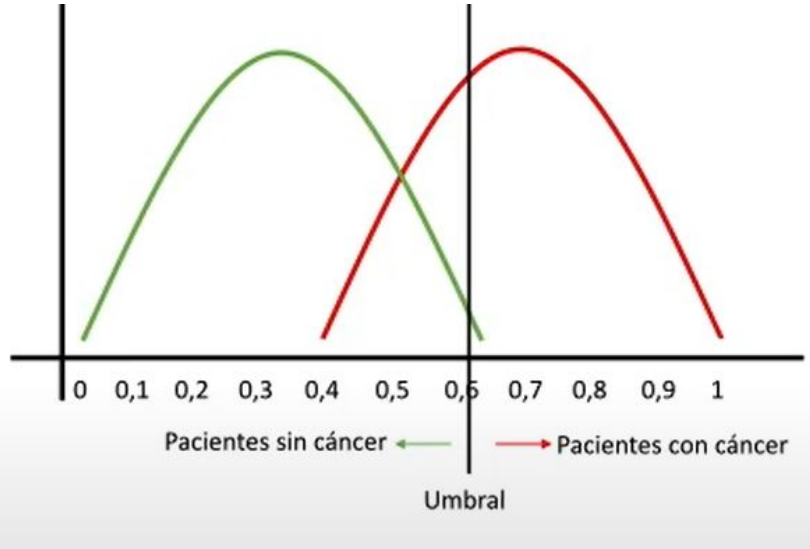
$$especificidad = \frac{TN}{TN + FP}$$

**Sensibilidad:** mide qué tan buena es una prueba o modelo para detectar correctamente los casos positivos. Es la proporción de casos positivos que la prueba o modelo identifica correctamente como positivos.

**Especificidad:** mide qué tan buena es una prueba o modelo para detectar correctamente los casos negativos. Es la proporción de casos negativos que la prueba o modelo identifica correctamente como negativos.




Sensibilidad Aumenta, Especificidad Disminuye

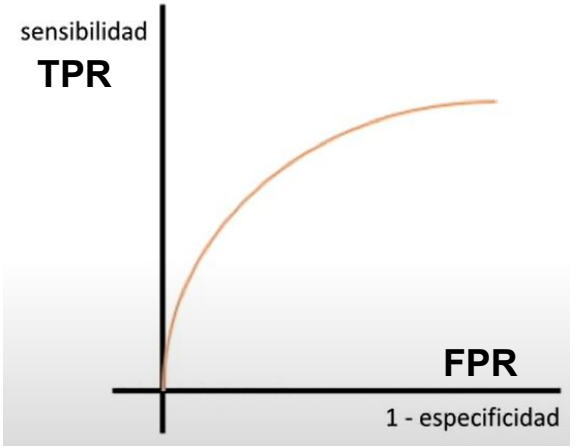



Sensibilidad Disminuye, Especificidad Aumenta

https://www.youtube.com/watch?v=AcbbkCL0dlo

# Curvas ROC: TPR: True Positive Rate y FPR: False Positive Rate

## Métricas de medición de rendimiento.




| Predicción / Realidad | Tiene plomo | No tiene plomo |
|---|---|---|
| Tiene plomo | Verdadero Positivo (TP) | Falso Negativo (FN) |
| No tiene plomo | Falso Positivo (FP) | Verdadero Negativo (TN) |

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

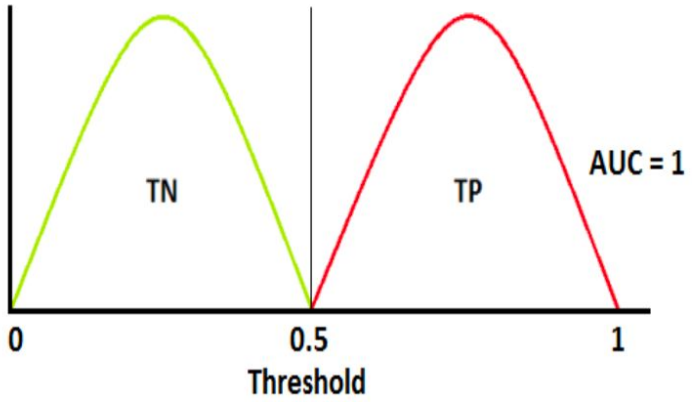
**TPR** o **Recall** o **Sensibilidad**: (**tasa de verdaderos positivos**) mide qué tan bueno es un modelo para detectar correctamente los casos positivos. En otras palabras, es la proporción de casos positivos que el modelo clasifica correctamente como positivos.

**FPR** o **1 - Especificidad:** (**tasa de falsos positivos**) mide qué tan propenso es el modelo a clasificar incorrectamente los casos negativos como positivos. En otras palabras, es la proporción de casos negativos que el modelo clasifica erróneamente como positivos.

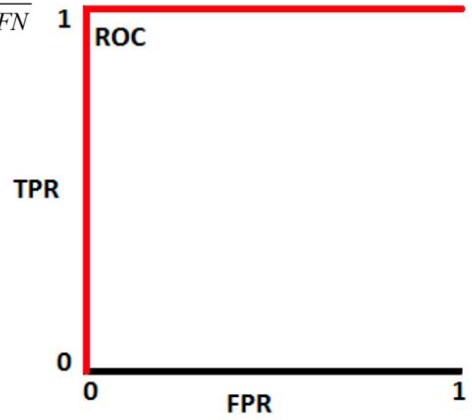
https://www.youtube.com/watch?v=TmhzUdPpVPQ

# Curvas ROC: Algunos ejemplos

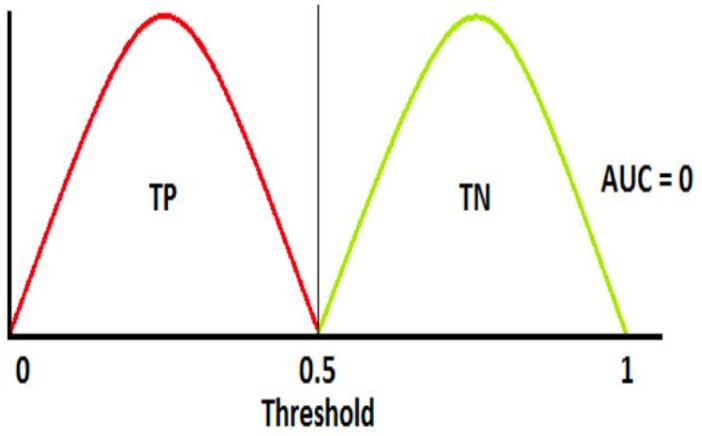
## Métricas de medición de rendimiento.



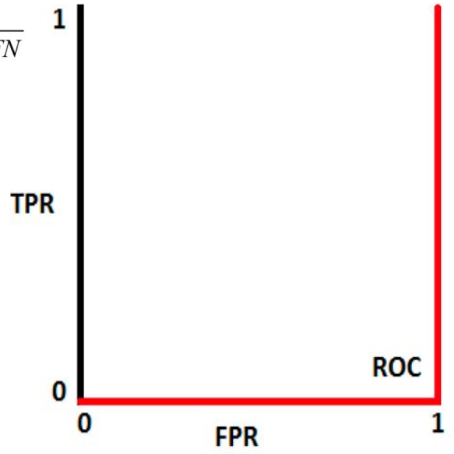

$$TPR = \frac{TP}{TP + FN}$$




$$FPR = \frac{FP}{FP + TN}$$



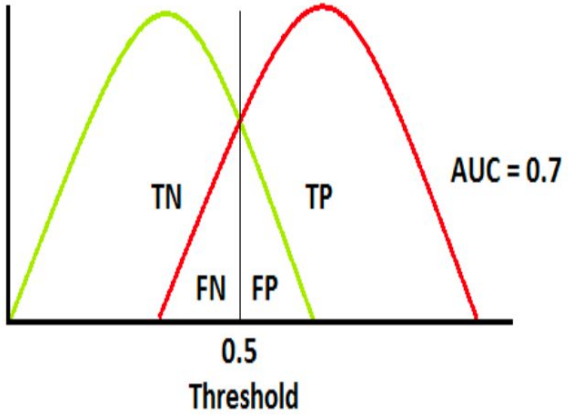

$$TPR = \frac{TP}{TP + FN}$$

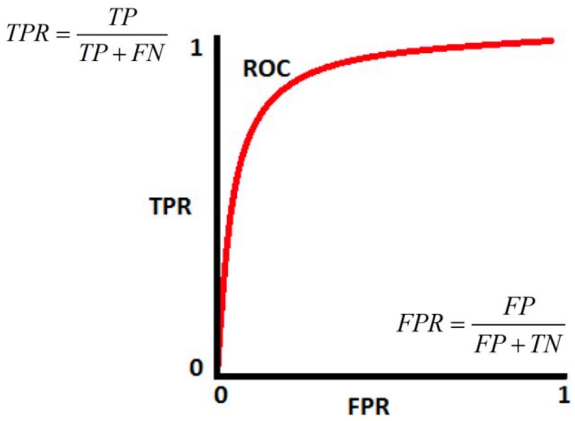


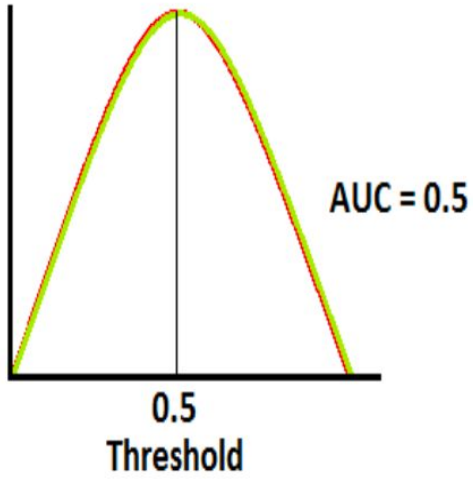

$$FPR = \frac{FP}{FP + TN}$$

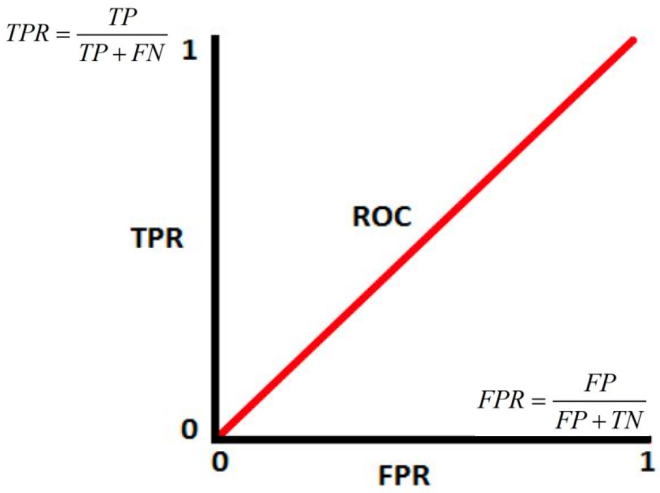
# Curvas ROC: Algunos ejemplos

## Métricas de medición de rendimiento.



# Evaluación

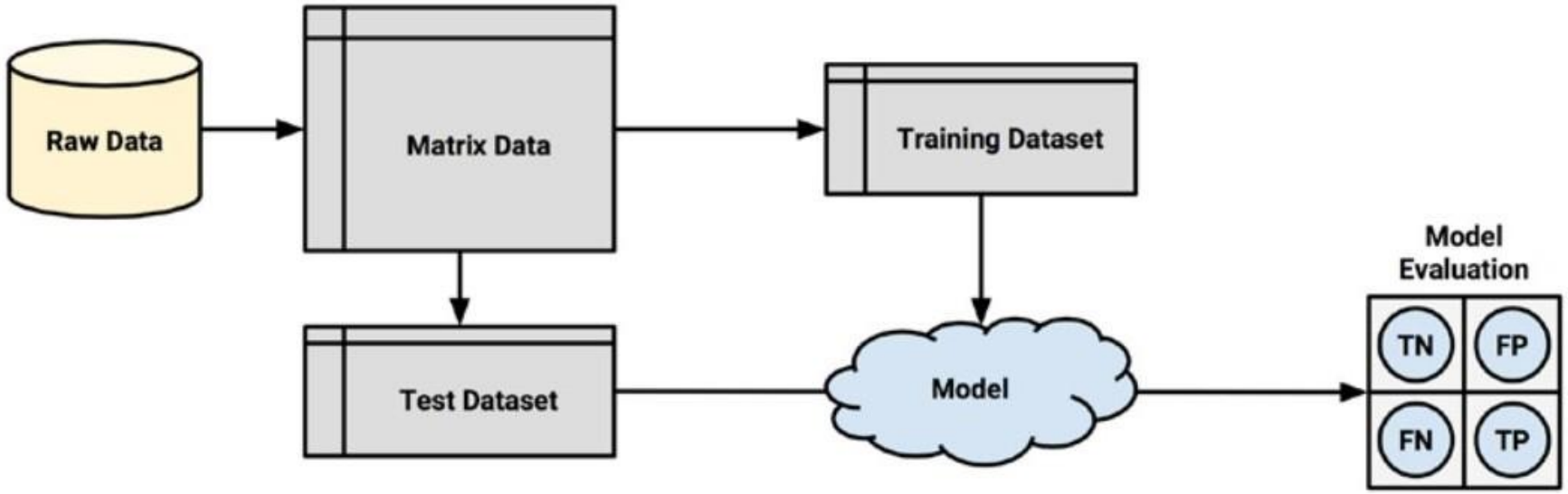
**Métricas de medición de rendimiento.**

- Generar un modelo con todos los datos disponibles y luego evaluarlos sobre los mismos, puede resultar en modelos sobre ajustados. Es decir, esos modelos no generalizan bien para datos nuevos.
- Para mitigar este riesgo, se genera un modelo en datos de entrenamiento y luego evaluamos en datos de prueba (que no hayan sido vistos durante el proceso de entrenamiento).

# Evaluación

**Métricas de medición de rendimiento.**

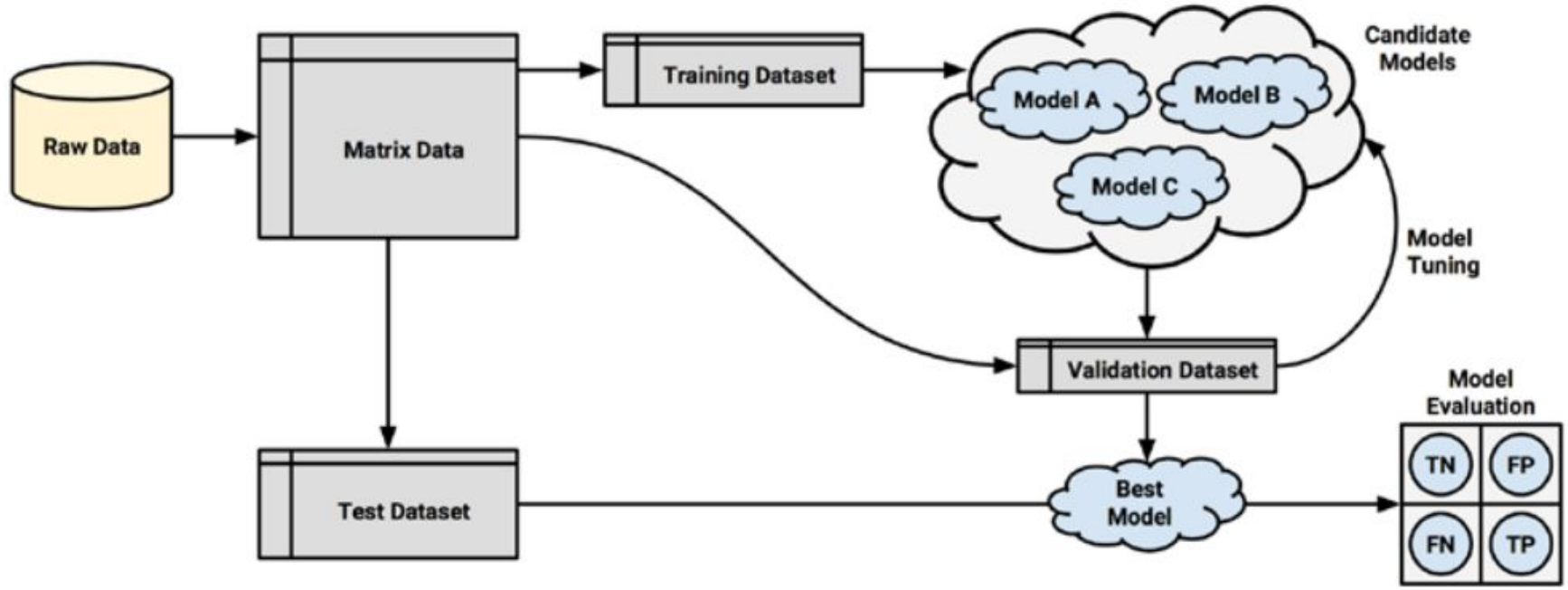
- Holdout simple: particionar en datos de entrenamiento y  prueba.



# Evaluación

**Métricas de medición de rendimiento.**

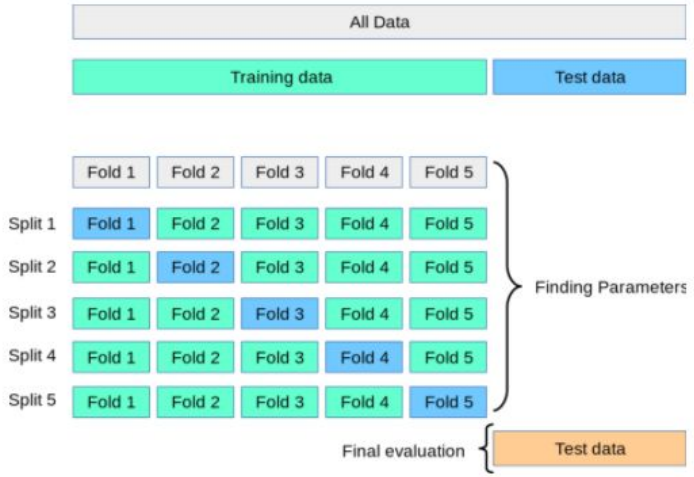
- Holdout con validación: particionar en datos de entrenamiento, validación y prueba



# Evaluación

**Métricas de medición de rendimiento.**

- Holdout con repetición (K-fold cross-validation).
  - Elegimos algún K y generamos K particiones (folds) de los datos.
  - Entrenamos y calculamos alguna métrica con cada fold.
  - Iteramos con todas las particiones y promediamos.
- Es problemático si el entrenamiento es lento.
- Evaluación estándar para modelos de Machine Learning.



# Transformación de características.

# Normalización de datos

**Transformación de características.**

Normalización: ajustar los valores medidos en diferentes escalas respecto a una escala común

- Normalización min-max $$z_i = \frac{x_i - min(x)}{max(x) - min(x)}$$
- Normalización z-score $$z_i = \frac{x_i - \overline{x}}{\sigma}$$

Carpeta con la Data requerida por los Notebooks link

# A programar

Link

**Transformación de características.**

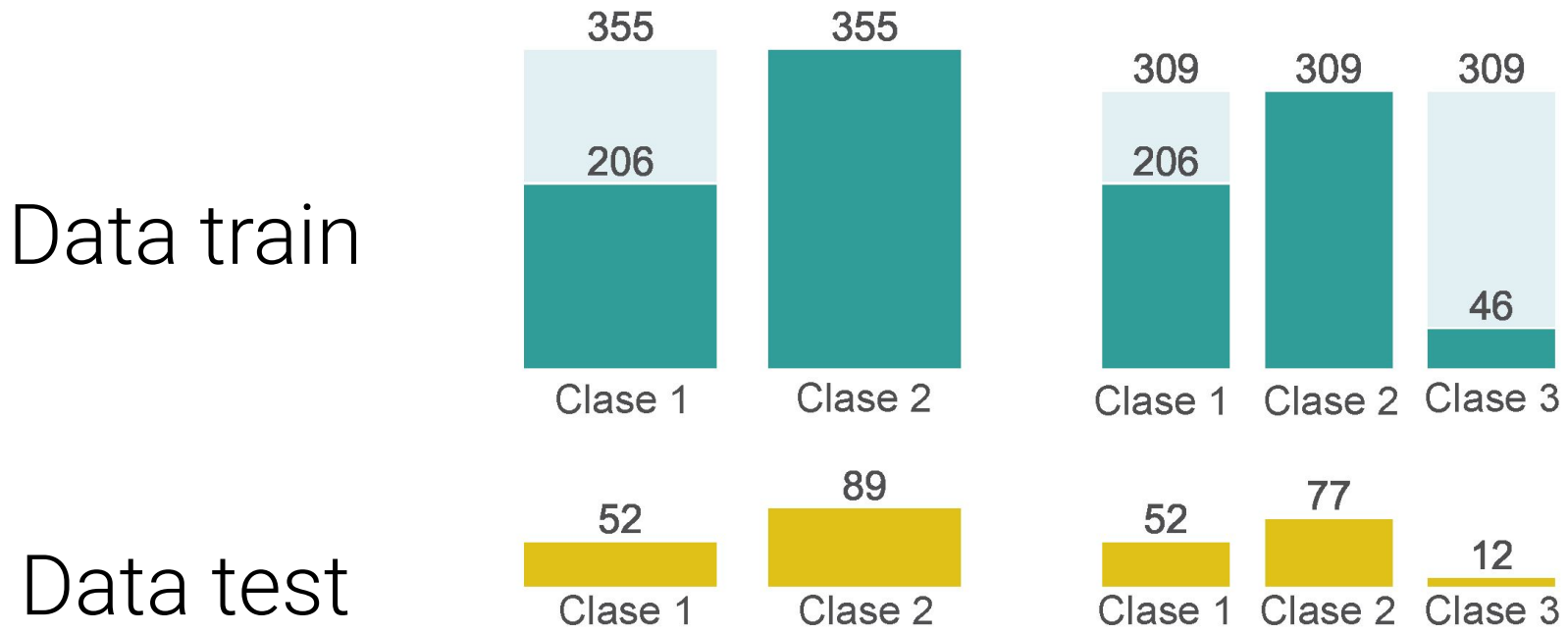
- Librerias
- Lectura del dataset
- Eliminar columnas innecesarias del dataset
- Análisis exploratorio de datos (Exploratory data analysis - EDA)
- División en datos de entrenamiento y testing
- **K-Nearest Neighbors**
- **Métricas**
- **Métricas bonitas**
- **KNN con preprocesamiento de las features**
- Balance de clases
- Reducción de dimensionalidad

# Balance de clases.

# ¿Cuándo balancear?

**Balance de clases.**

Se deben explorar los datos, si estos se encuentran desbalanceados, se puede realizar el balance de clases.



# Técnicas para balancear clases

**Balance de clases.**

- **A la clase mayor**
  - Random
  - SMOTE
    - BorderlineSMOTE
    - KMeansSMOTE
    - SVMSMOTE
    - SMOTEN
  - ADASYN

# Técnicas para balancear clases

**Balance de clases.**

- **A la clase menor**
  - Random
  - ClusterCentroids
  - NearMiss

# Reducción de dimensionalidad.

# Análisis de Componentes Principales (PCA)

**Reducción de dimensionalidad.**

- Técnica utilizada para describir un conjunto de datos en términos de **nuevas variables** («componentes») **no correlacionadas**. Los componentes se ordenan por la cantidad de varianza original que describen, por lo que la técnica es útil para reducir la dimensionalidad de un conjunto de datos.
- Técnicamente, el PCA busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. Esta convierte un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables sin correlación lineal llamadas **componentes principales**.
- Se puede seleccionar las primeras componentes para representar los datos.

Carpeta con la Data requerida por los Notebooks link

# A programar

Link

**Transformación de características.**

- Librerias
- Lectura del dataset
- Eliminar columnas innecesarias del dataset
- Análisis exploratorio de datos (Exploratory data analysis - EDA)
- División en datos de entrenamiento y testing
- K-Nearest Neighbors
- Métricas
- Métricas bonitas
- KNN con preprocesamiento de las features
- **Balance de clases**
- **Reducción de dimensionalidad**