

“... los animales se dividen en

- (a) pertenecientes al Emperador,
- (b) embalsamados,
- (c) amaestrados,
- (d) lechones,
- (e) sirenas,
- (f) fabulosos,
- (g) perros sueltos,
- (h) incluidos en esta clasificación,
- (i) que se agitan como locos,
- (j) innumerables,
- (k) dibujados con un pincel finísimo de pelo de camello,
- (l) etcétera,
- (m) que acaban de romper el jarrón,
- (n) que de lejos parecen moscas.”

El Idioma Analítico de John Wilkins  
Jorge Luis Borges



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Laboratorio de datos Clasificación, KNN y métricas

Primer cuatrimestre 2025

# Ejemplo

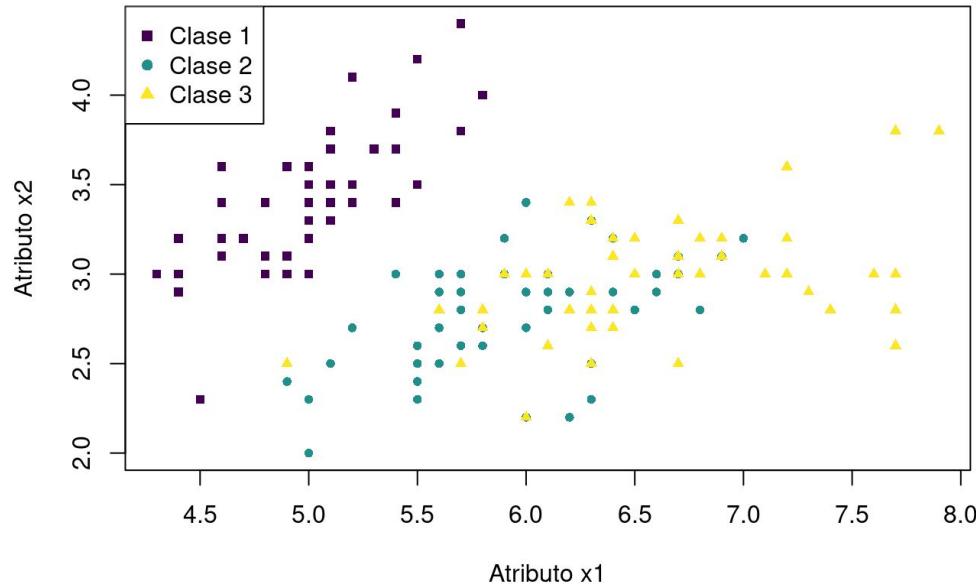
Tenemos un conjunto de datos con variables  $x_1, x_2, y$ .

- Variables explicativas continuas  $x_1, x_2$
- Variable a explicar y categórica

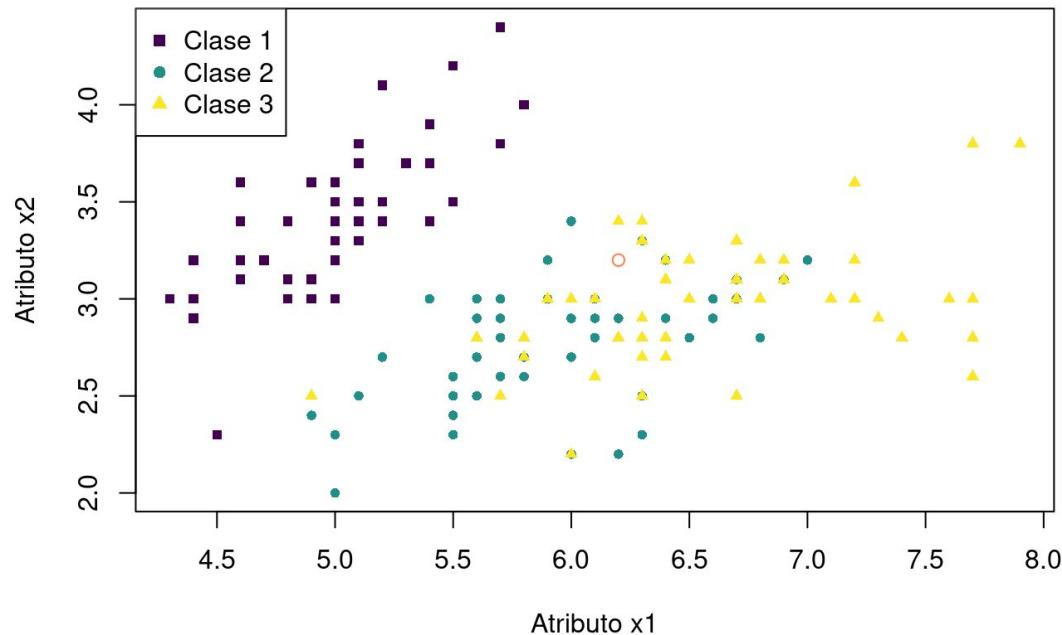
Graficamos  $x_1, x_2$  en los ejes.

La variable a explicar toma 3 valores: Clase 1, Clase 2, Clase 3 y se representa con símbolos/colores.

Scatterplot por clases



### Scatterplot por clases



¿Qué clase le asignamos a la nueva observación?

# Clasificación

- A partir de los atributos (variables explicativas) queremos poder determinar la etiqueta - la variable categórica Y.
- Aprendizaje supervisado: contamos con un conjunto de entrenamiento en el cual conocemos las etiquetas - valores de la variable Y.
- Evaluación del modelo: medida relacionada con la cantidad de elementos bien o mal clasificados.

# Clasificación

## Tipos de clasificación

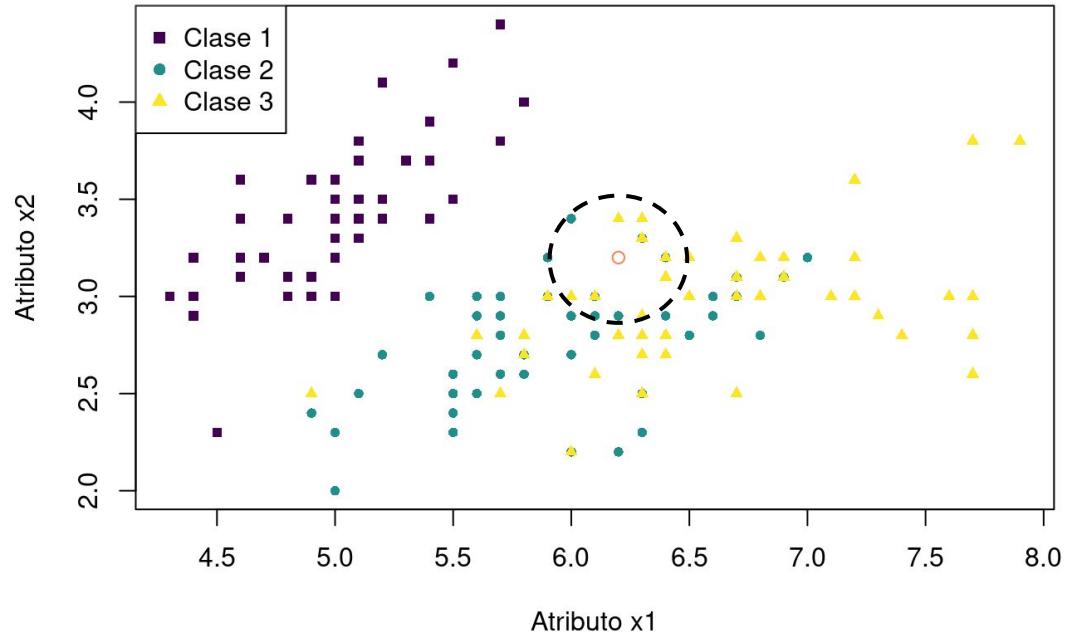
- Binaria - Dos clases, True/False, Positivo/Negativo
- Multiclasificación - Más de dos clases

# Clasificación

A ojo: ¿QUÉ CLASE SERÁ?

Una idea puede ser:  
mirar alrededor.

Scatterplot por clases



# K Nearest Neighbors (KNN)

Para determinar la clase de una nueva observación:

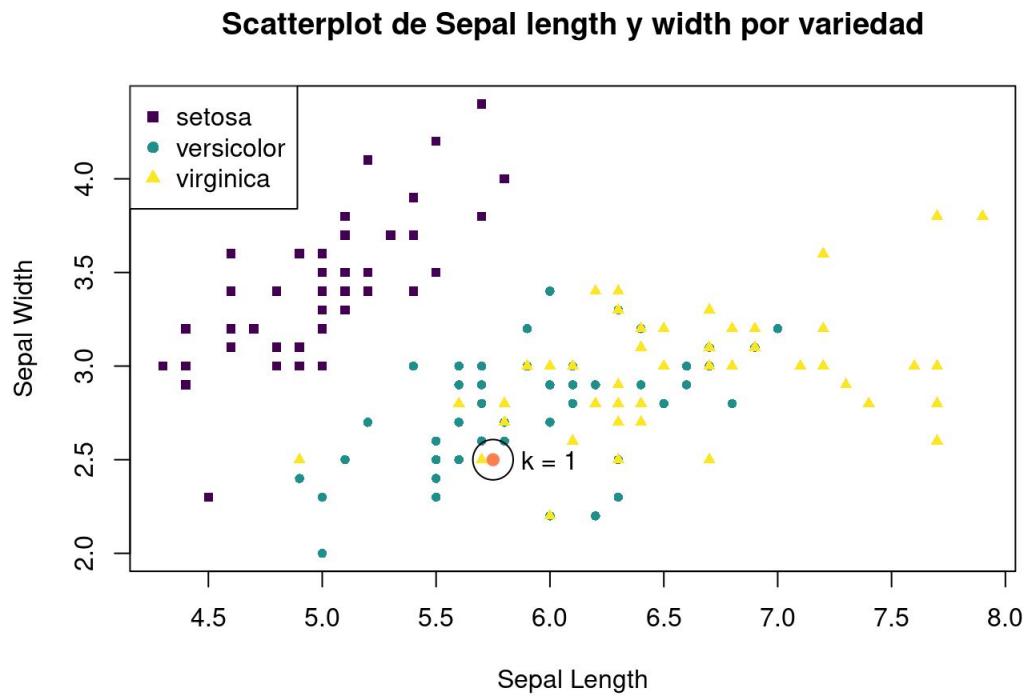
1. Buscar los puntos más cercanos, dentro del conjunto de entrenamiento
2. Ver qué clases tienen
3. Elegir la mayoritaria

¿Cuántos puntos consideramos? Depende del valor de k.

Si  $k = 1$ :

Buscamos el vecino más cercano, entre los que ya tenemos etiquetados.

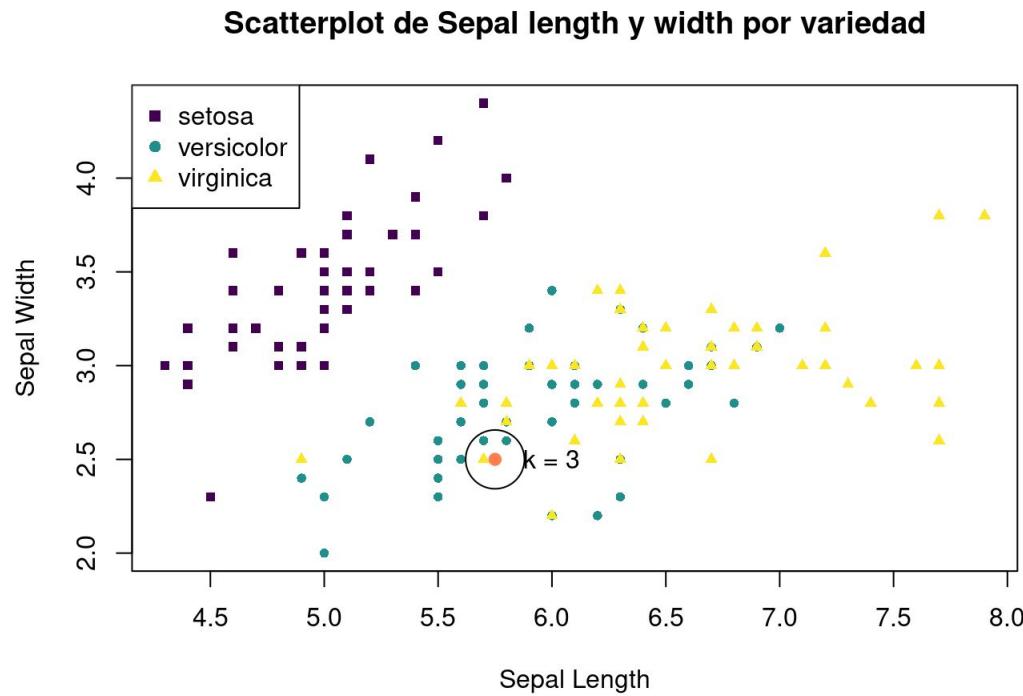
Nos copiamos esa etiqueta.



Si  $k = 3$ :

Buscamos los 3 más cercanos, entre los que ya tenemos etiquetados.

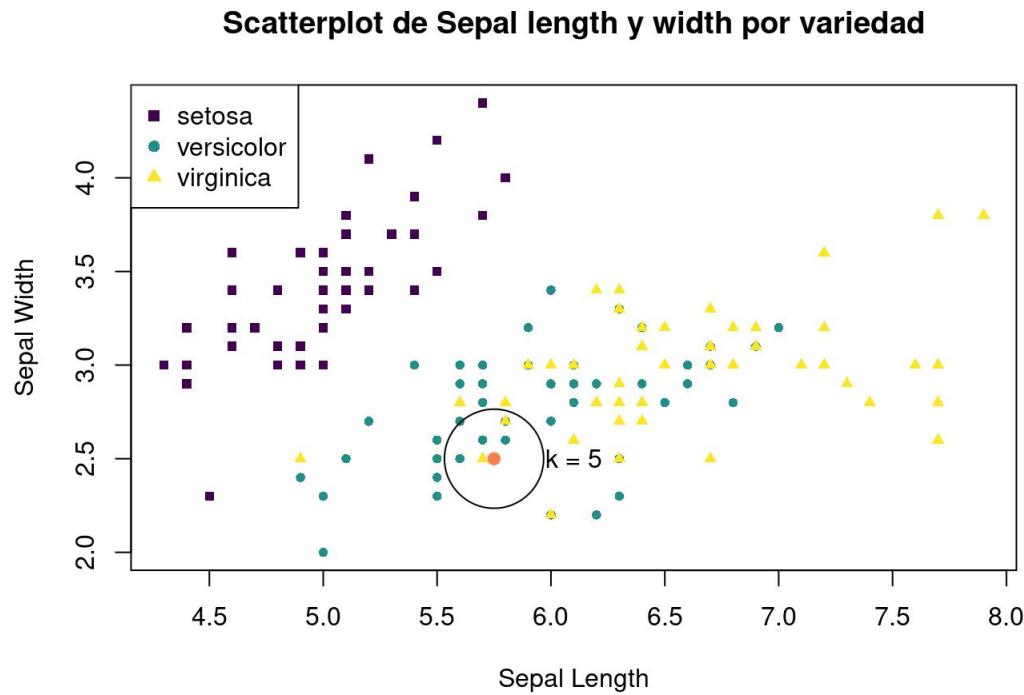
Tomamos la clase mayoritaria.



Si  $k = 5$ .

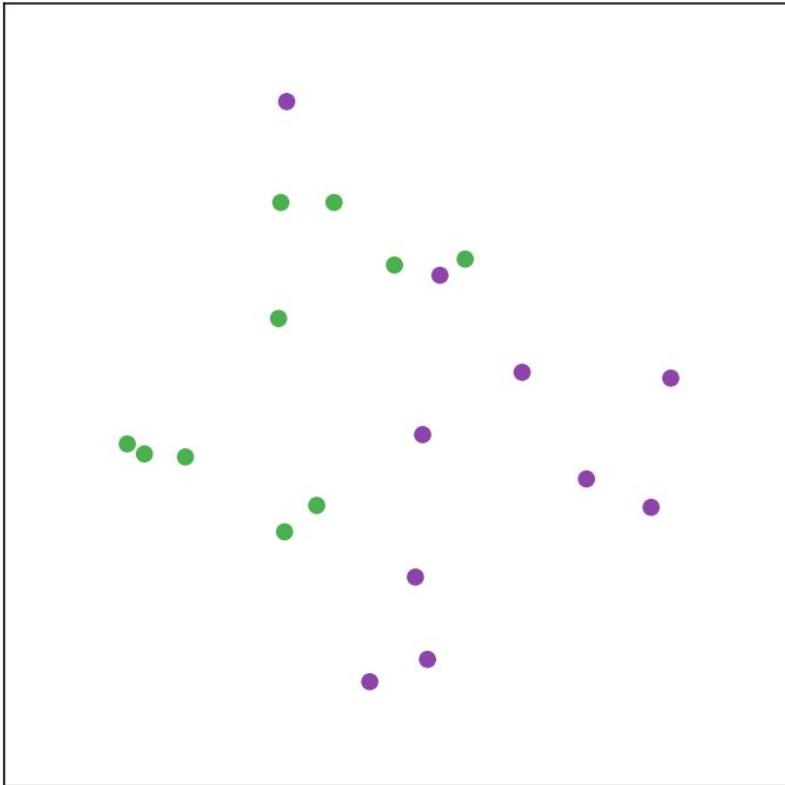
Buscamos los 5 más cercanos, entre los que ya tenemos etiquetados.

Tomamos la clase mayoritaria.

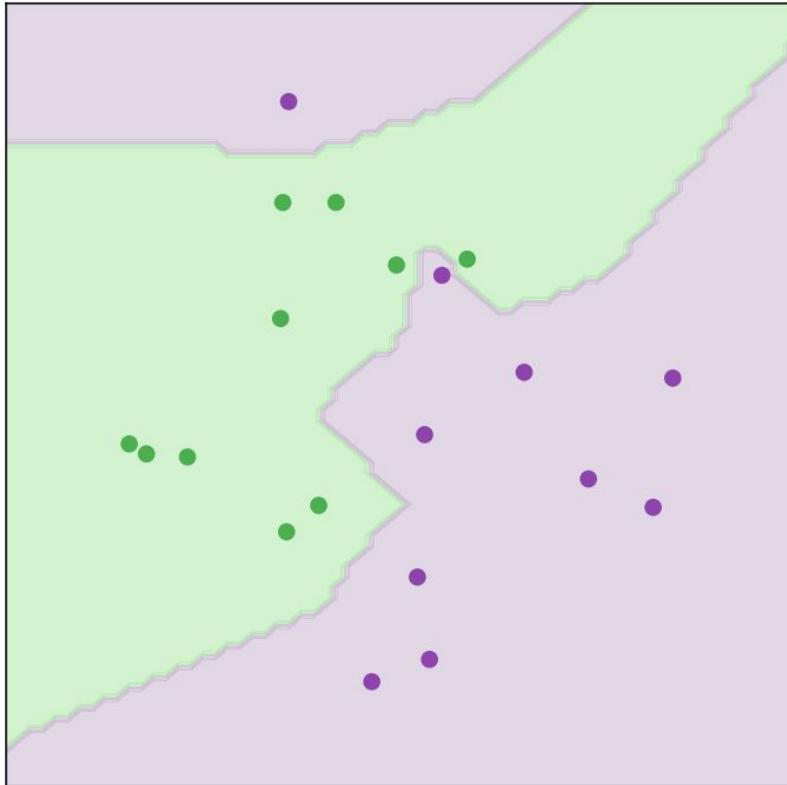


# Clasificación con K Nearest Neighbors (KNN)

- Depende de k
- Depende de los atributos elegidos
- Depende de la distancia elegida para determinar cercanía

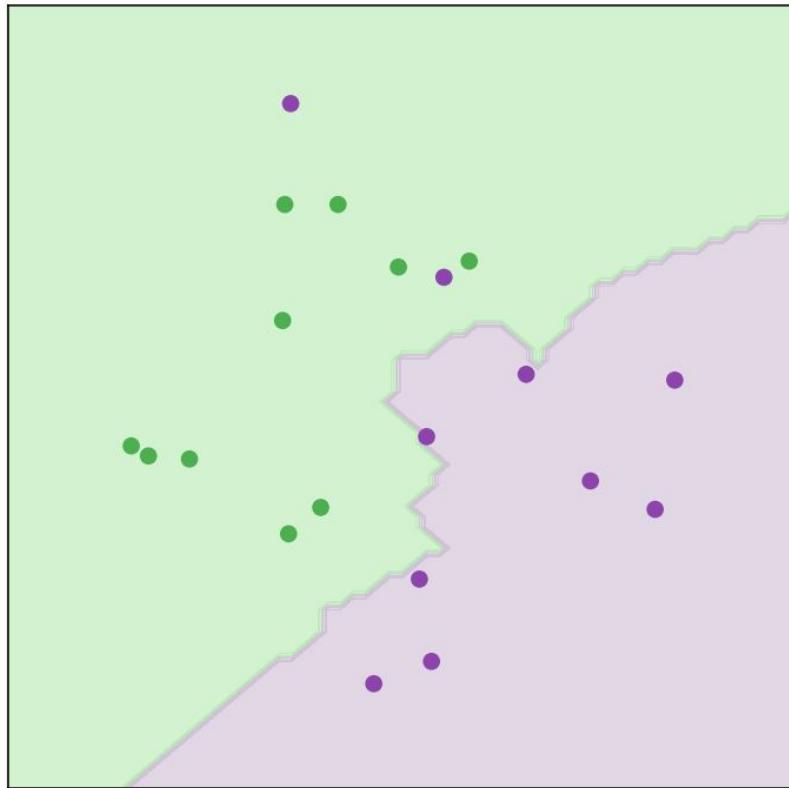


Conjunto de datos  
*etiquetados*.  
Dada una *nueva*  
observación, la misma será  
clasificada según la  
proporción de clases en su  
entorno.

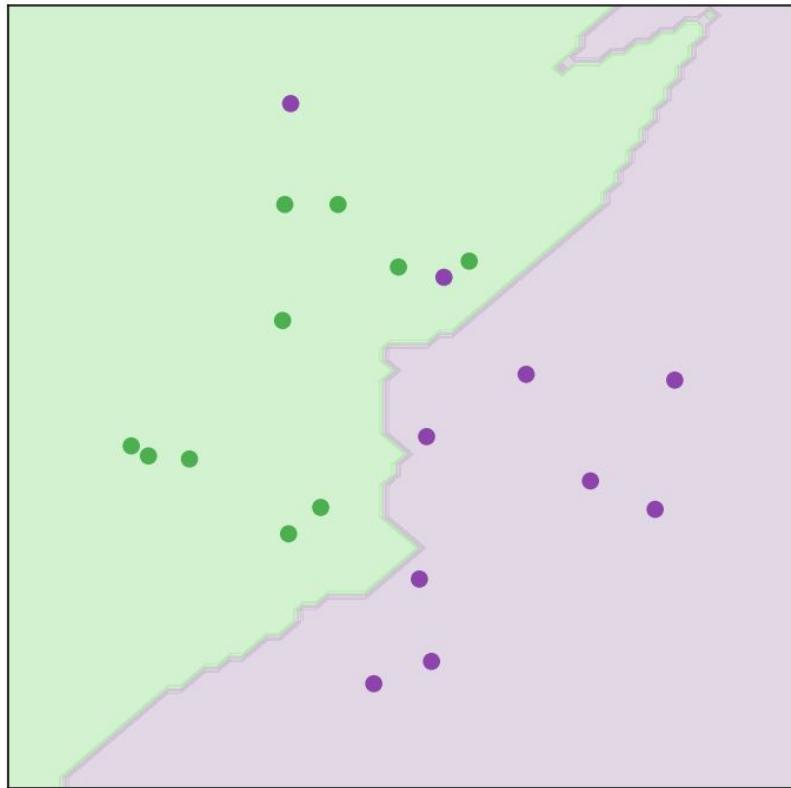


$$K = 1$$

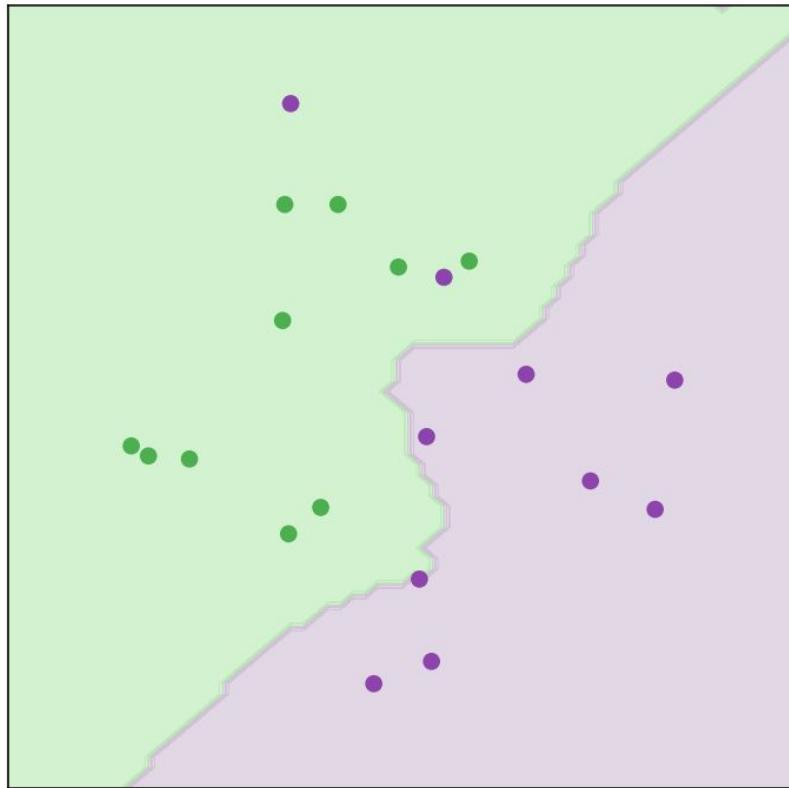
Dada una nueva observación, la misma será clasificada según si cae en la zona verde o en la zona lila.



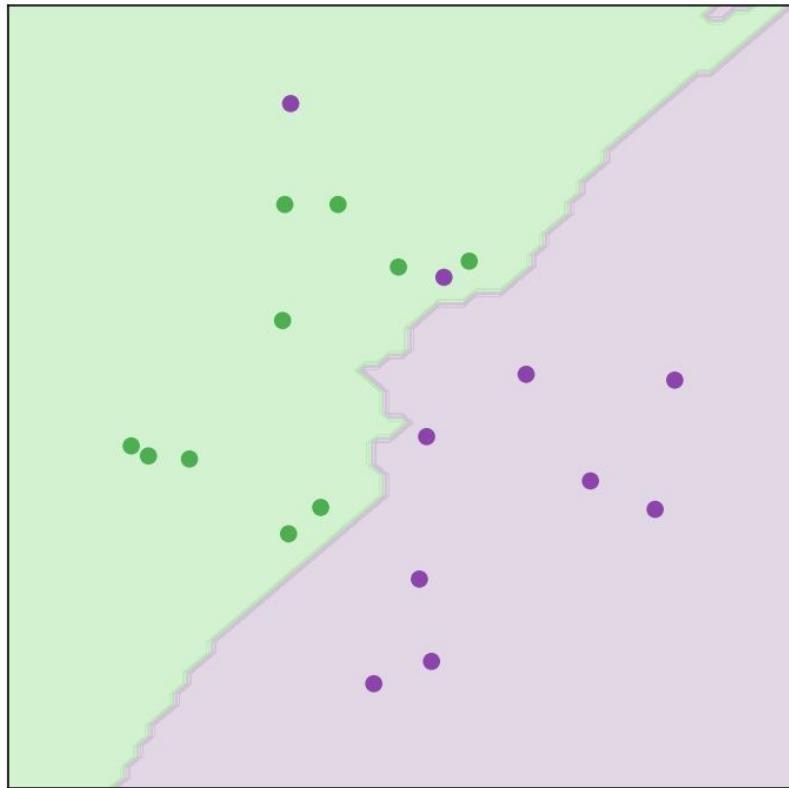
$$K = 2$$



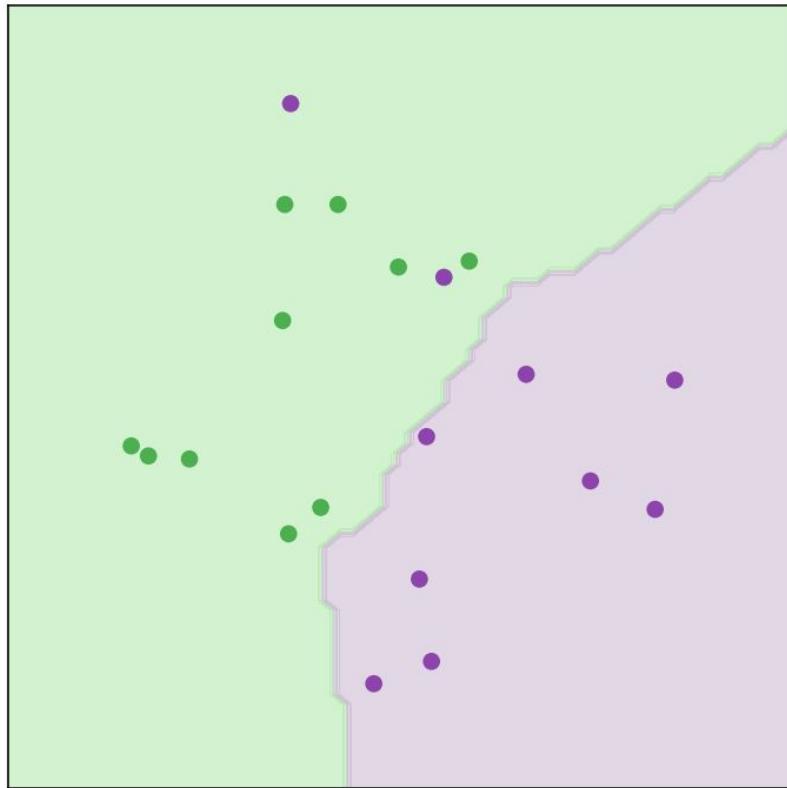
$$K = 3$$



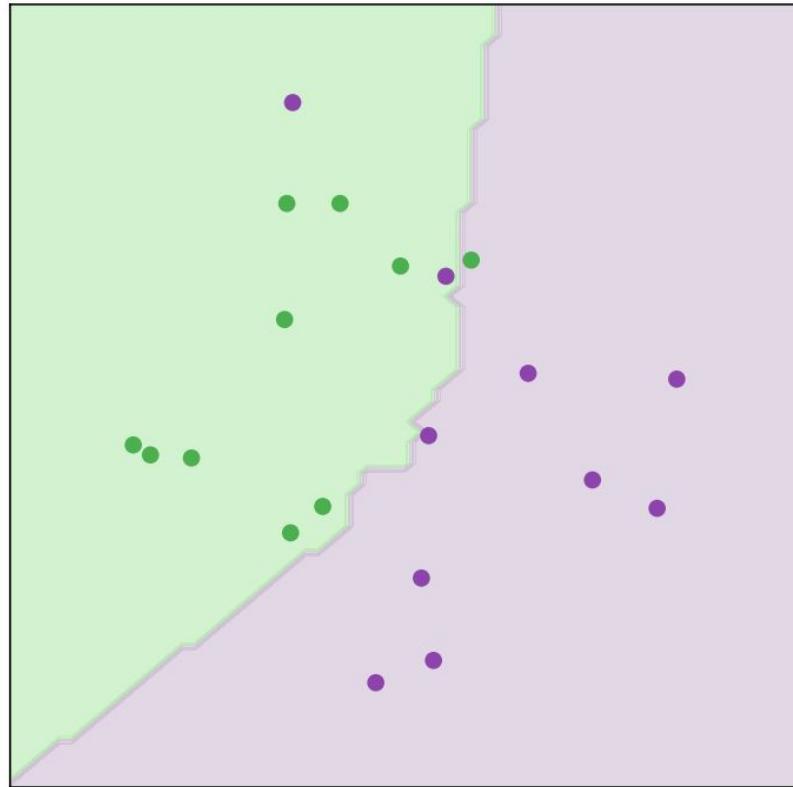
$$K = 4$$



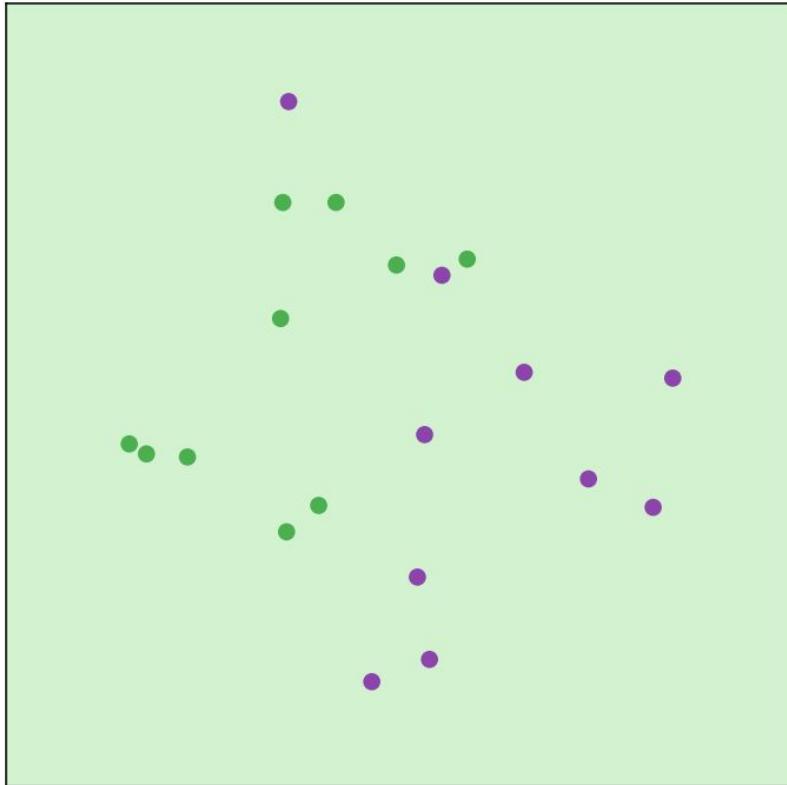
$K = 5$



$K = 6$



$K = 13$



$K = 20$

# KNN

¿Cómo cambia el clasificador a medida que aumenta el valor de k?

# KNN con sklearn

```
# importar paquete de sklearn
from sklearn.neighbors import KNeighborsClassifier

# construir el modelo
clasificador = KNeighborsClassifier(n_neighbors = K)

# entrenar el modelo con datos
# X son los atributos, y son las etiquetas
clasificador.fit(X, y)

# utilizar el modelo para predecir nuevo(s) caso(s)
prediccion = clasificador.predict(Xnuevo)
```

# Ejemplo - clasificación binaria

Entrenaremos un clasificador binario para predecir si un paciente tiene o no diabetes.

**Objetivo del modelo:** Predecir si un paciente tiene o no diabetes basándose en determinadas mediciones incluidas en el conjunto de datos.

Se trata de un recorte de una base de datos más amplia. En particular, todos los pacientes son mujeres de al menos 21 años y de ascendencia india pima.

# Consigna

- Trabajar con los datos de diabetes.
- Construir un clasificador basado en ['Glucose', 'BMI'], probar con distintos valores de k, determinar el diagnóstico para una paciente con valores de glucosa = 130 y BMI = 32.
- Construir un clasificador basado en otro conjunto de atributos y comparar resultados.

## Ejemplo - datos sintéticos

Trabajamos con datos sintéticos para ver cómo cambia la frontera de decisión.

# Consigna

- Utilizando `from sklearn.datasets import make_moons` construir un dataset "moons" con 200 datos y ruido = 0.2.
- Construir y entrenar un clasificador con k = 5 y graficar la frontera de decisión.
- Comparar con otros valores posibles para k.

¿Tenemos un clasificador bueno?

# Medidas para evaluar clasificadores

## Matriz de confusión

Para cada clase i, nos fijamos cuántas observaciones de la clase fueron clasificadas en cada clase j.

Esto nos da una matriz cuadrada, con una fila y columna por cada clase.

	0	1	2
0	50	0	0
1	0	29	21
2	0	0	50

{0: 'setosa', 1: 'versicolor', 2: 'virginica'}

- De 50 casos de clase 0, todos clasificados como 0.
- De los 50 casos de clase 1, 29 clasificados como clase 1 y 21 como clase 2.
- De los 50 casos de clase 2, todos clasificados como clase 2.

# Medidas para evaluar clasificadores

## Matriz de confusión

$M_{ij} = \#$  cuántas observaciones  $i$  fueron clasificadas como  $j$

En la diagonal se contabilizan los aciertos, fuera de la diagonal figuran los errores.

	0	1	2
0	50	0	0
1	0	29	21
2	0	0	50

{0: 'setosa', 1: 'versicolor', 2: 'virginica'}

- De 50 casos de clase 0, todos clasificados como 0.
- De los 50 casos de clase 1, 29 clasificados como clase 1 y 21 como clase 2.
- De los 50 casos de clase 2, todos clasificados como clase 2.

# Medidas para evaluar clasificadores

## Matriz de confusión

A veces se representan valores relativos, en vez de absolutos.

$$M_{ij} = \# (\text{observaciones } i \text{ clasificadas como } j) / N$$

	0	1	2
0	50	0	0
1	0	29	21
2	0	0	50

	0	1	2
0	0.333	0	0
1	0	0.193	0.14
2	0	0	0.333

# Medidas para evaluar clasificadores

## Exactitud

La exactitud o *accuracy* que es una medida numérica que cuenta la proporción de observaciones *bien* clasificadas.

En la matriz de confusión, sumamos los elementos de la diagonal.

$$\text{Acc} = \sum_i M_{ii} \text{ (suma de la diagonal)}$$

	0	1	2
0	50	0	0
1	0	29	21
2	0	0	50


$$50+29+50 = 129$$
$$129/150 = 0.86$$

# Métricas en sklearn

```
# importo las métricas
from sklearn.metrics import accuracy_score, confusion_matrix

# calculo mis predicciones
y_pred = clasificador.predict(x2)

# computo la matriz de confusión comparando y con y_pred
confusion_matrix(y,y_pred)

# computo la exactitud comparando y con y_pred
accuracy_score(y,y_pred)
```

# Ejemplos

En los ejemplos vistos, comparar la exactitud obtenida con distintos valores de k.

# Evaluación de modelos ↔ selección de modelos

Necesitamos poder evaluar los modelos de una forma efectiva para:

- Comparar configuraciones de algoritmos
- Estimar la performance que tendrá el modelo “en la realidad”

Evaluar bien significa entender cómo será el uso, cuál es el objetivo del modelo, qué métrica refleja bien lo que queremos medir.

# Ejemplo

Se trata de detectar una enfermedad. Se estima que la proporción de población enferma es del 6%.

Desarrollaron un test que tiene 94% de exactitud. ¿Es bueno?

# Ejemplo

1% de las mujeres tienen cáncer de mama. Desarrollaron un test que tiene esta performance.

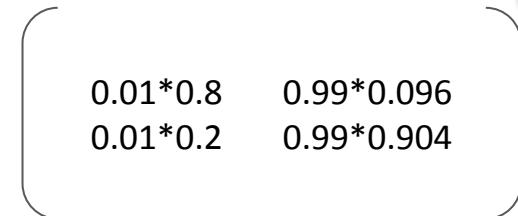
	Cancer (1%)	No Cancer (99%)
Test Pos	80%	9.6%
Test Neg	20%	90.4%

Es decir:

- 1% de los casos es positivo
- De los casos positivos, el 80% testea positivo.
- De los negativos, 9.6% testea positivo.

¿Cómo es la matriz de confusión?

$$\begin{array}{cc} 0.01 \cdot 0.8 & 0.99 \cdot 0.096 \\ 0.01 \cdot 0.2 & 0.99 \cdot 0.904 \end{array}$$



¿SUMA 100%?

# Ejemplo

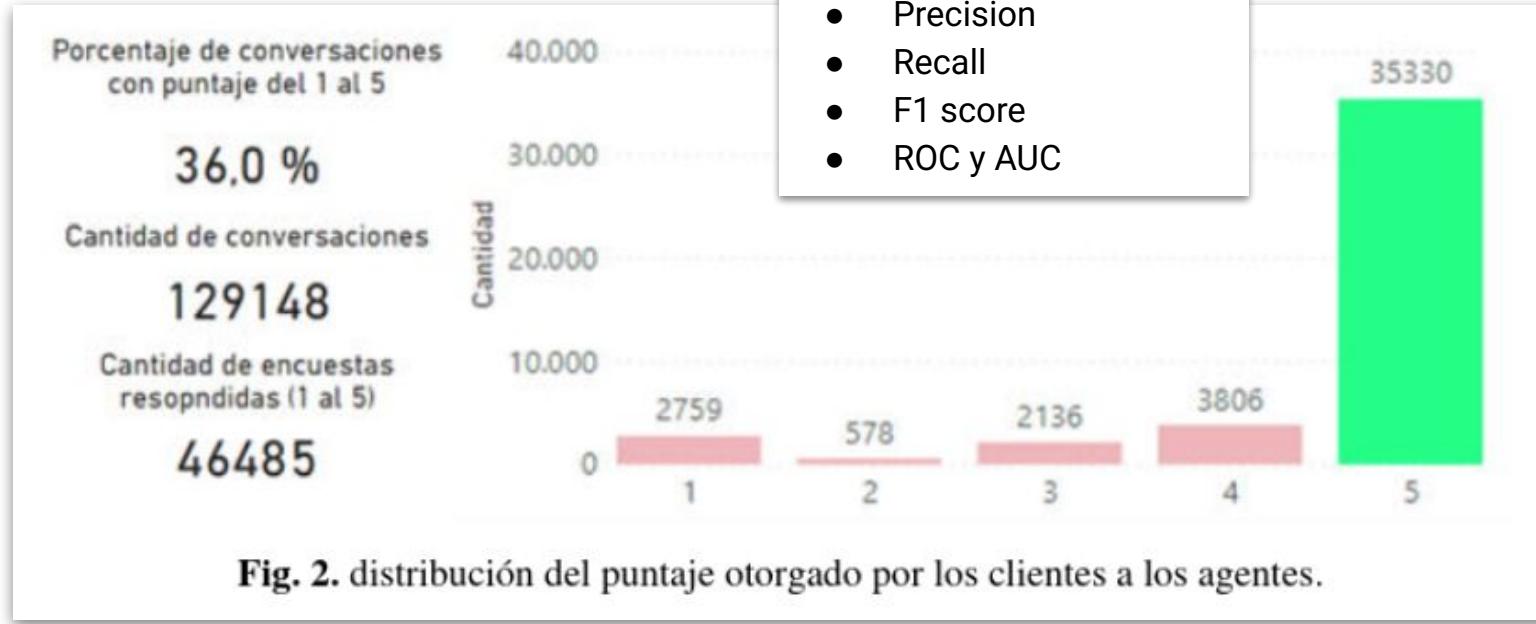
- ¿Cuánto es el accuracy?
- ¿Si el test da positivo, significa que el paciente tiene la enfermedad?

$$\begin{bmatrix} 0.01*0.8 & 0.99*0.096 \\ 0.01*0.2 & 0.99*0.904 \end{bmatrix}$$

# Ejemplos

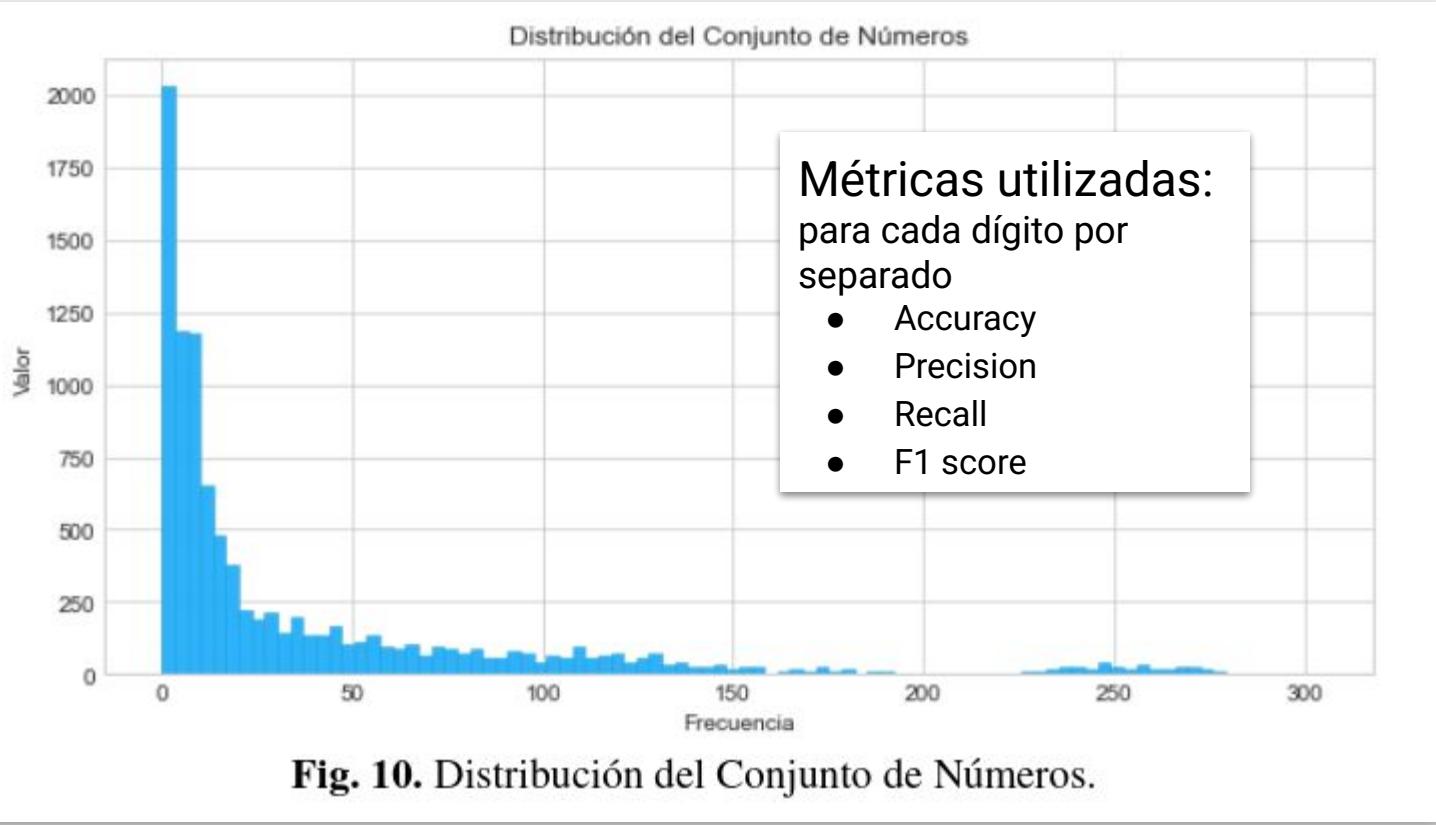
## Métricas utilizadas:

- Accuracy
- Precision
- Recall
- F1 score
- ROC y AUC



*Predicting user satisfaction from customer service chats*

<https://publicaciones.sadio.org.ar/index.php/EJS/article/view/839/677>



*Reuse of a Deep Learning model for handwritten digit recognition*

<https://publicaciones.sadio.org.ar/index.php/EJS/article/view/841/679>

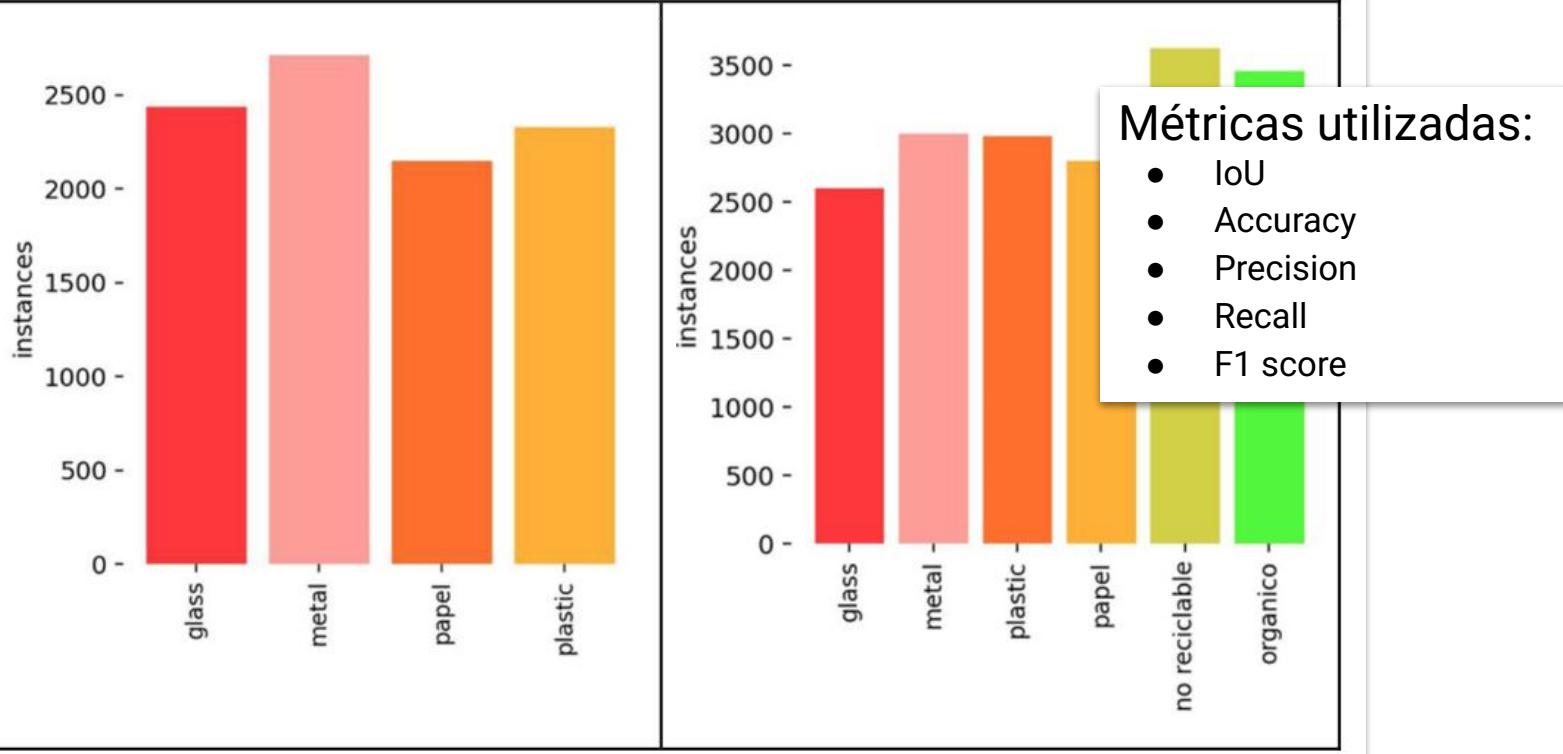


Fig. 4. Comparación de los dos enfoques utilizados

Técnicas de Deep Learning aplicadas a un sistema de clasificación de objetos para un recolector de residuos inteligente - <https://publicaciones.sadio.org.ar/index.php/EJS/article/view/844/681>

# Medidas de performance

## Matriz de confusión - caso binario

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

ERA ASÍ O TRASPUESTA?

TP: true positives

FP: false positives

TN: true negatives

FN: false negatives

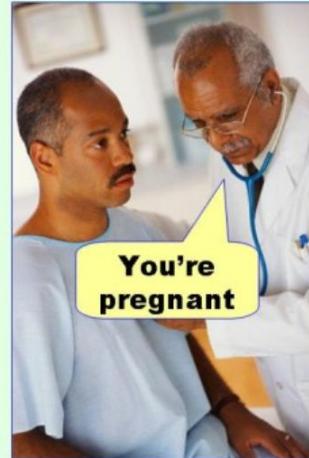
No dice nada sobre los tipos de aciertos y de errores que tiene el modelo.

Ej: autenticación en aplicación por voz.

- FP: autentica a un impostor
- FN: no autentica a un usuario válido

# Tipos de error

**Type I error**  
(false positive)



**Type II error**  
(false negative)



Tomado de Towards Data Science

# Medidas de performance

$$\text{Precisión} = \frac{TP}{TP + FP}$$

de las instancias clasificadas como positivas, cuántas lo son  
(cuán útiles son los resultados de búsqueda)

$$\text{Recall} = \frac{TP}{TP + FN}$$

(exhaustividad)

de las instancias positivas, cuántas fueron clasificadas como positivas  
(cuán completos son los resultados)

# Medidas de performance

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{Precisión} = \frac{TP}{TP + FP}$$

¿Cuál medida de performance debería priorizar cada uno de estos sistemas?

- enfermedad contagiosa
- test de embarazo
- filtro spam

**Media armónica:**

$$F\text{-measure} = 2 \cdot \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}}$$

También llamada **F<sub>1</sub> score**.

Fórmula general:

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{Precisión} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precisión}) + \text{Recall}}$$

F<sub>2</sub> da más peso a Recall

F<sub>0.5</sub> da más peso a Precisión

# Medidas de performance

$$\text{Recall} = \frac{TP}{TP + FN} = \textcolor{blue}{\text{Sensitivity o bien True Positive Rate}}$$

$$\frac{TN}{TN + FP} = \textcolor{blue}{\text{Specificity o bien True Negative Rate}}$$

**Sensitivity/TPR:** Porcentaje de pacientes **enfermos** correctamente diagnosticados.  
Proporción de usuarios válidos autenticados

**Specificity:** Porcentaje de pacientes **sanos** correctamente diagnosticados.

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

Ej. FPR: Proporción de impostores que aceptamos erróneamente.

$$\text{Precisión} = \text{PPV} = \frac{TP}{TP + FP}$$

¿Qué hacemos con un resultado de un estudio médico que nos da mal, pero que tiene bajo PPV?

# Ejercicio

En los ejemplos vistos, comparar los clasificadores con las distintas métricas.

- Accuracy (Exactitud)
- Precision
- Recall
- F1

# Casos multiclasificación

	Manzana (predicho)	Naranja (predicho)	Oliva (predicho)	Pera (predicho)
Manzana (real)	MM	MN	MO	MP
Naranja (real)	NM	NN	NO	NP
Oliva (real)	OM	ON	OO	OP
Pera (real)	PM	PN	PO	PP

Las medidas **precisión, recall, etc.** sólo pueden formularse en forma binaria: **cada clase contra el resto**

$$\text{Precisión(Manzana)} = \frac{MM}{MM + NM + OM + PM}$$

$$\text{Recall(Manzana)} = \frac{MM}{MM + MN + MO + MP}$$

# Tarea

Resolver la guía de ejercicios de clasificación con KNN.

# Bibliografía

## Libros:

- Introduction to Machine Learning with Python, Müller & Guido
- Machine Learning - Mitchell

