DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Laboratorio de datos

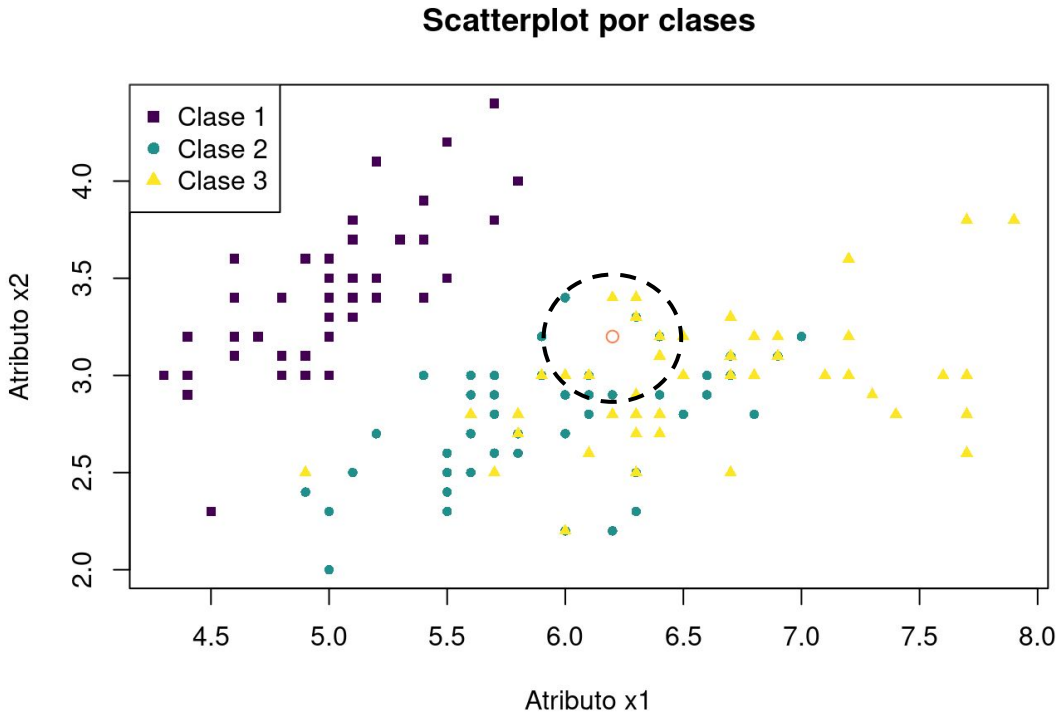
Clasificación 2

Verano 2025

Clasificación

A ojo: ¿QUÉ CLASE SERÁ?

Una idea puede ser: mirar alrededor.



K Nearest Neighbors (KNN)

Para determinar la clase de una nueva observación:

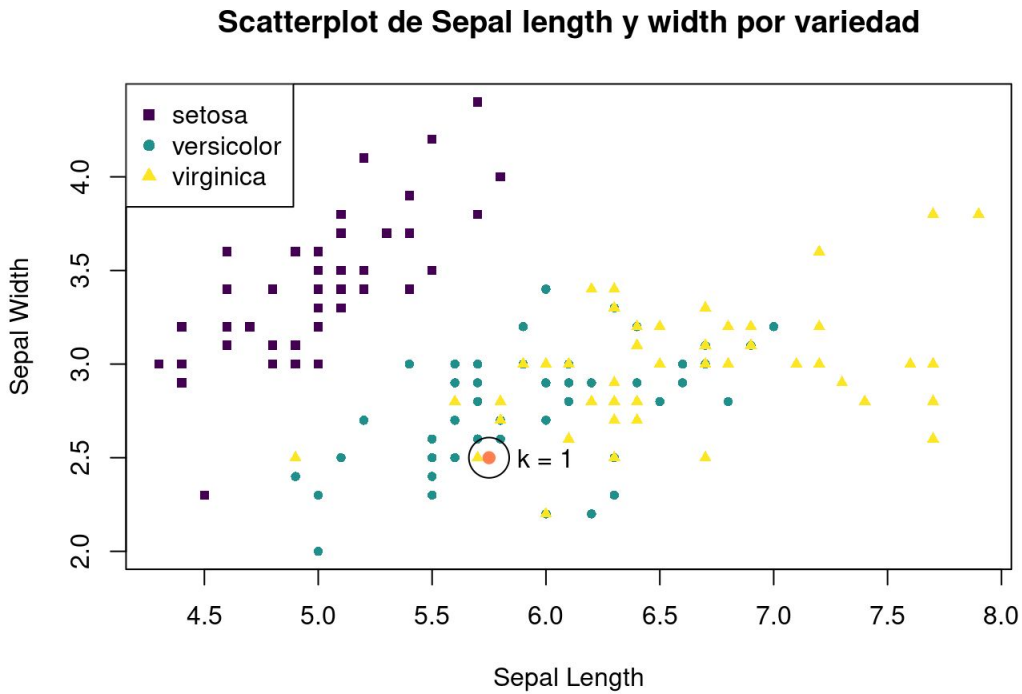
1. Buscar los puntos más cercanos, dentro del conjunto de entrenamiento
2. Ver qué clases tienen
3. Elegir la mayoritaria

¿Cuántos puntos consideramos? Depende del valor de k.

Si k = 1:

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

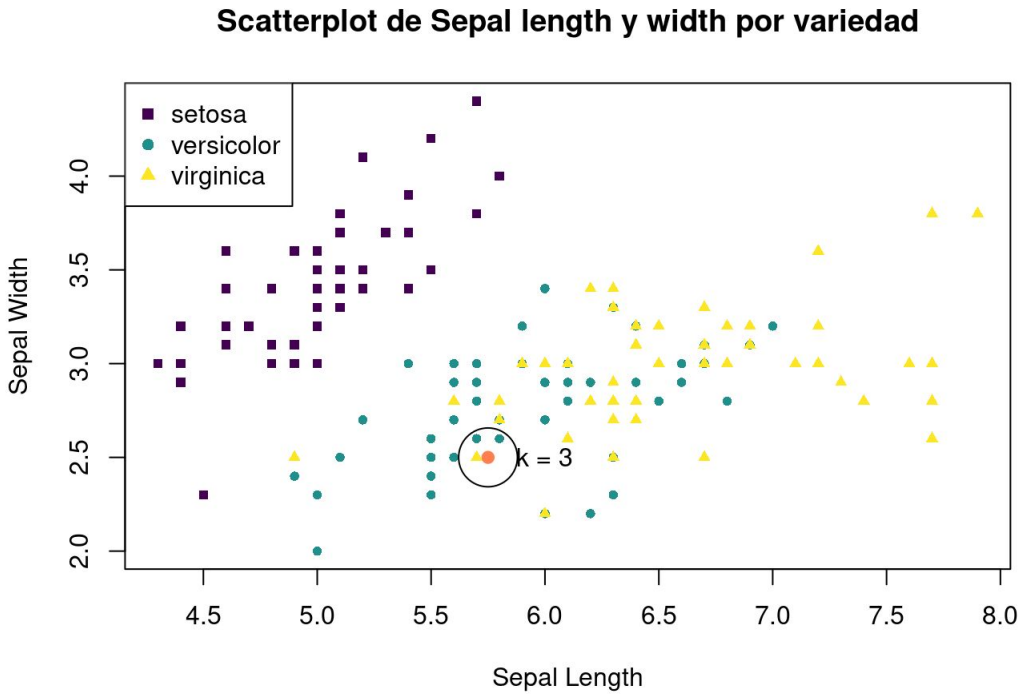
Nos copiamos esa etiqueta.



Si k = 3:

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

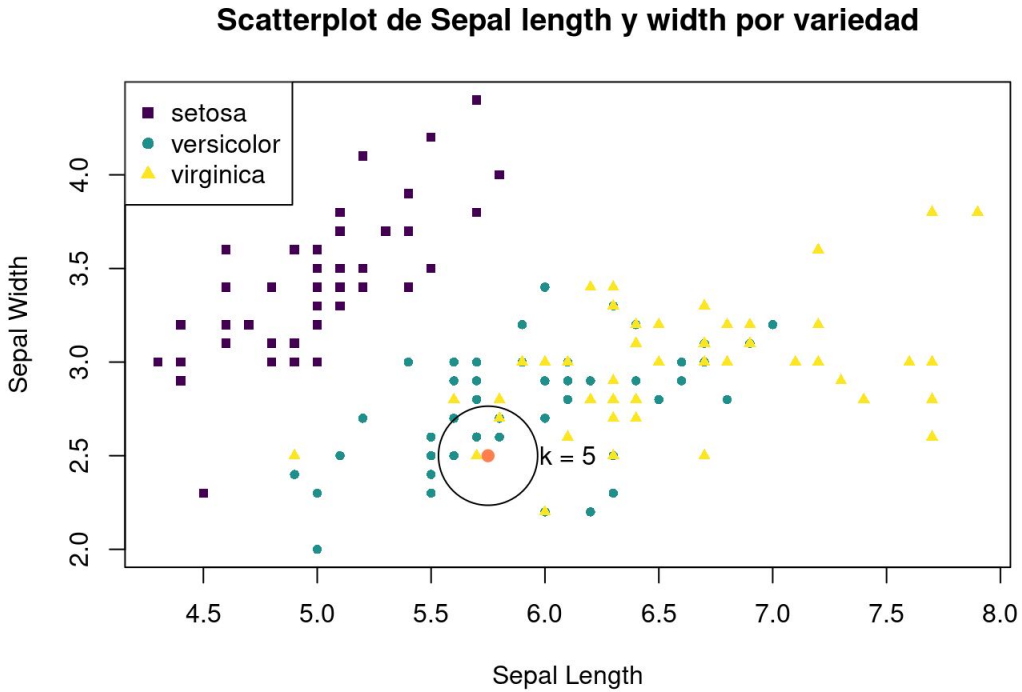
Tomamos la clase mayoritaria.



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

Ejemplos con Iris

Ejemplos, usando todo el dataset, con distintos valores de k.

Vamos a usar los 4 atributos (4 primeras columnas del dataframe) y la distancia euclídea.

Entrenamos y evaluamos el modelo.

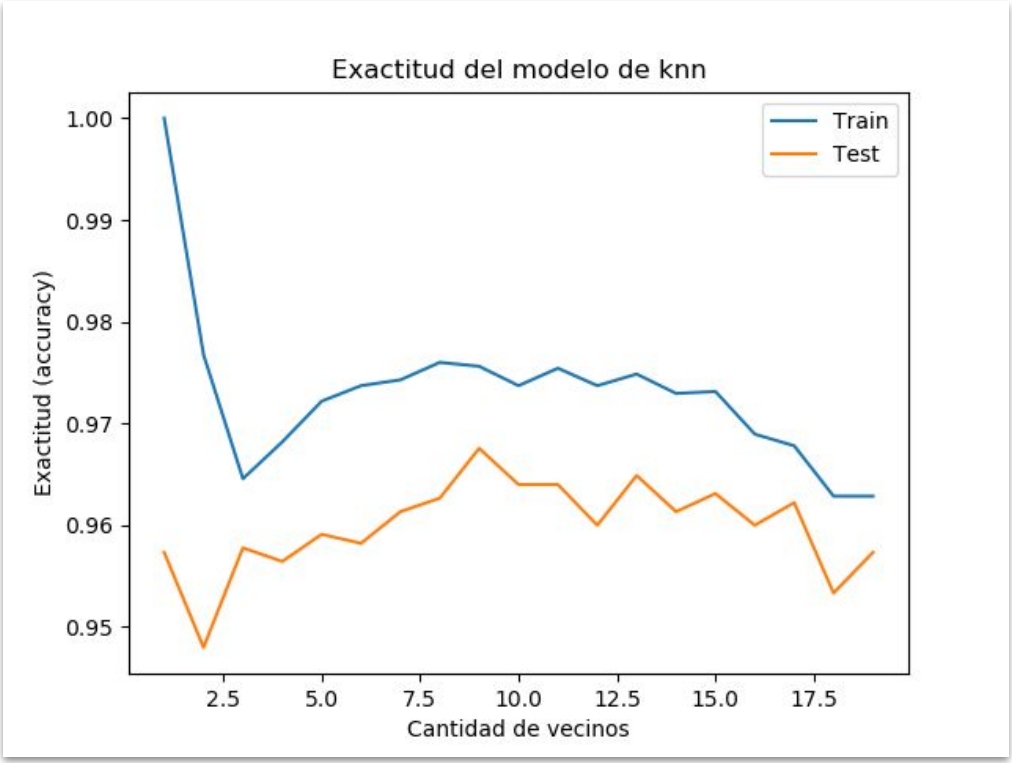
Performance de un modelo - ¿dónde?

Medir la performance sobre datos de entrenamiento no es una buena idea. Surge la necesidad de separar un % de datos, para validar los modelos: datos de validación (o test).

Ejemplos con Iris

¿Cuál es el mejor valor de k?

Evaluar los distintos valores de k y comparar.



Ejercicios

1. Pasamos al dataset de árboles. Cargar el csv.
2. Hacer un train-test split y hacer una clasificación con knn. Probar con distintos valores de k.
3. Cross-validation con k-folding: ajustar el modelo para cada valor de k dentro de un rango, y graficar la exactitud en función del k.
4. Reescalar los atributos para que tomen valores entre 0 y 1 y repetir. ¿Mejora la clasificación?

KNN involucra distancia
¿Cuál estamos usando?
¿Por qué puede impactar la escala?

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.

Evaluación de modelos - **selección** de modelos

- ¿Cómo sabemos cuán bueno es nuestro modelo?
- ¿Cuál de los posibles modelos es el mejor?

Primera idea:

- Accuracy (exactitud) sobre el conjunto de entrenamiento: porcentaje de datos de entrenamiento clasificados correctamente.
- Pero:
    - El modelo puede **memorizar** los datos de entrenamiento y tener **accuracy de 100%**. Medir **performance sobre los datos de entrenamiento** tiende a **sobreestimar los resultados**.

Selección de modelos

¿Por qué tendríamos distintos modelos para comparar?

- Distintos **atributos** (selección y transformación de atributos)
- Distintos **algoritmos** (umbral, árboles, KNN, SVM, …)
- Distintos **hiperparámetros** de cada algoritmo.

Ejemplo: hiperparámetros de los árboles de decisión

- Criterio de elección de atributos en cada nodo (Information Gain, Gini Gain...)
- Criterio de parada (ej: máxima profundidad)
- Estrategia de poda

Métricas en clasificación

Vimos: matriz de confusión y accuracy (exactitud).

M_{ii} = # cuántas observaciones

i fueron clasificadas como j

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


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

Más allá del accuracy: la elección de una métrica de evaluación debe basarse en el problema que se está abordando.

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

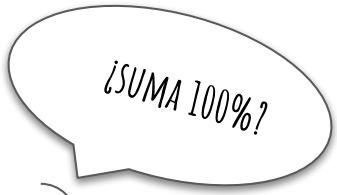
1% de las mujeres tienen cáncer de mama. Desarrollaron un test que tiene esta performance.

	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?



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

Ejemplo

- ¿Si el test da positivo, qué quiere decir?
- ¿Cuánto es el accuracy?

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

Ejemplos

Métricas utilizadas:

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

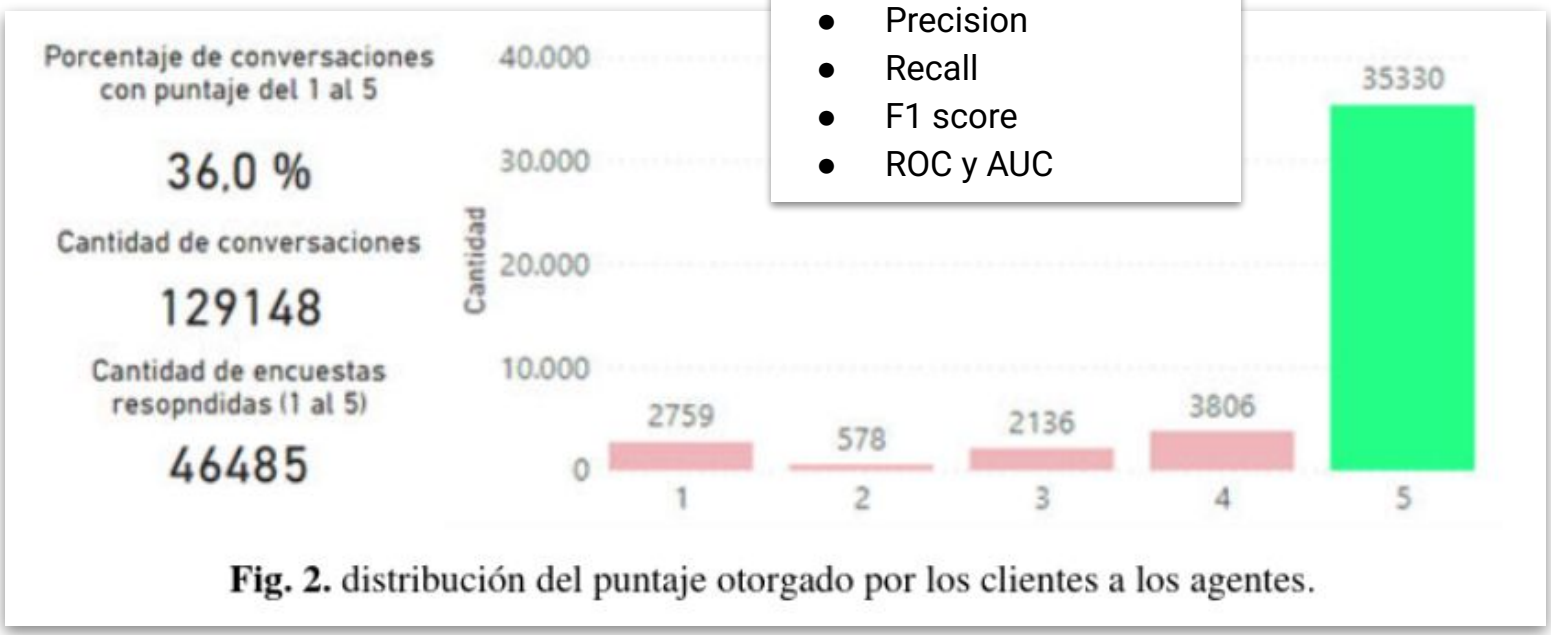


Fig. 2. distribución del puntaje otorgado por los clientes a los agentes.

Predicting user satisfaction from customer service chats
https://publicaciones.sadio.org.ar/index.php/EJS/article/view/839/677

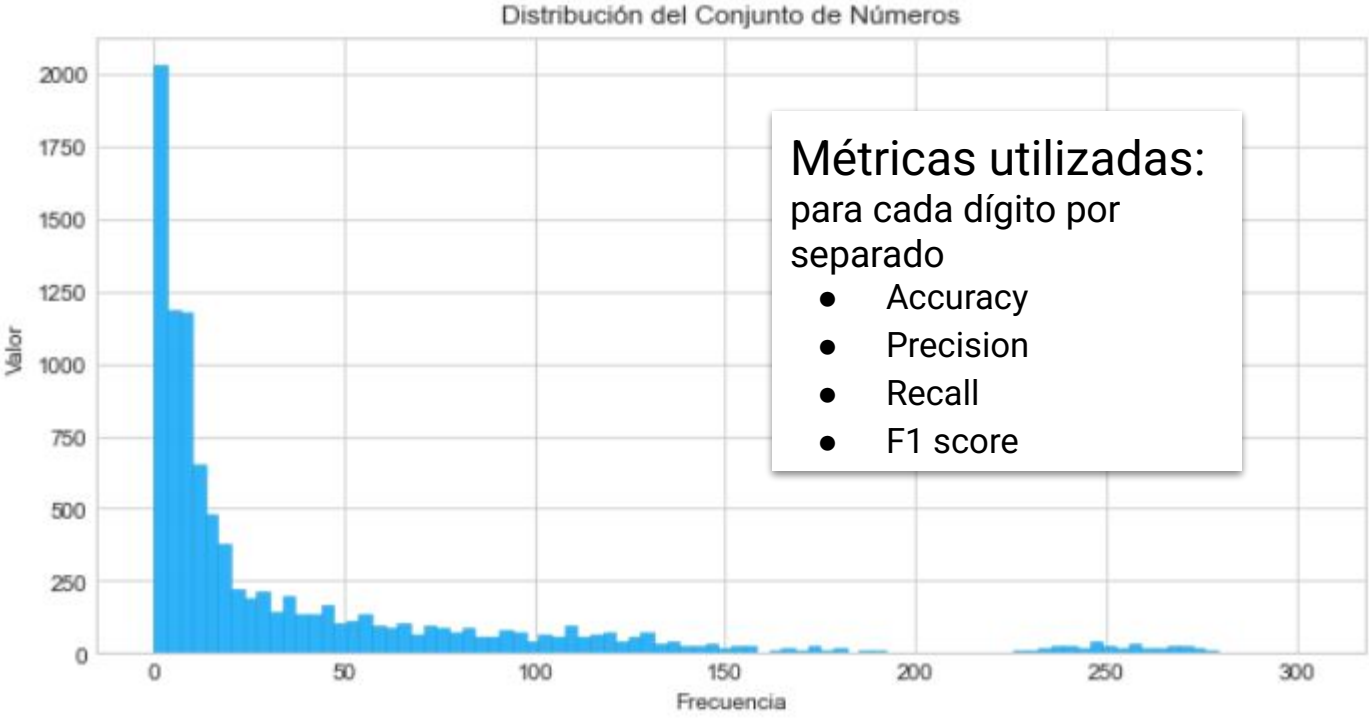
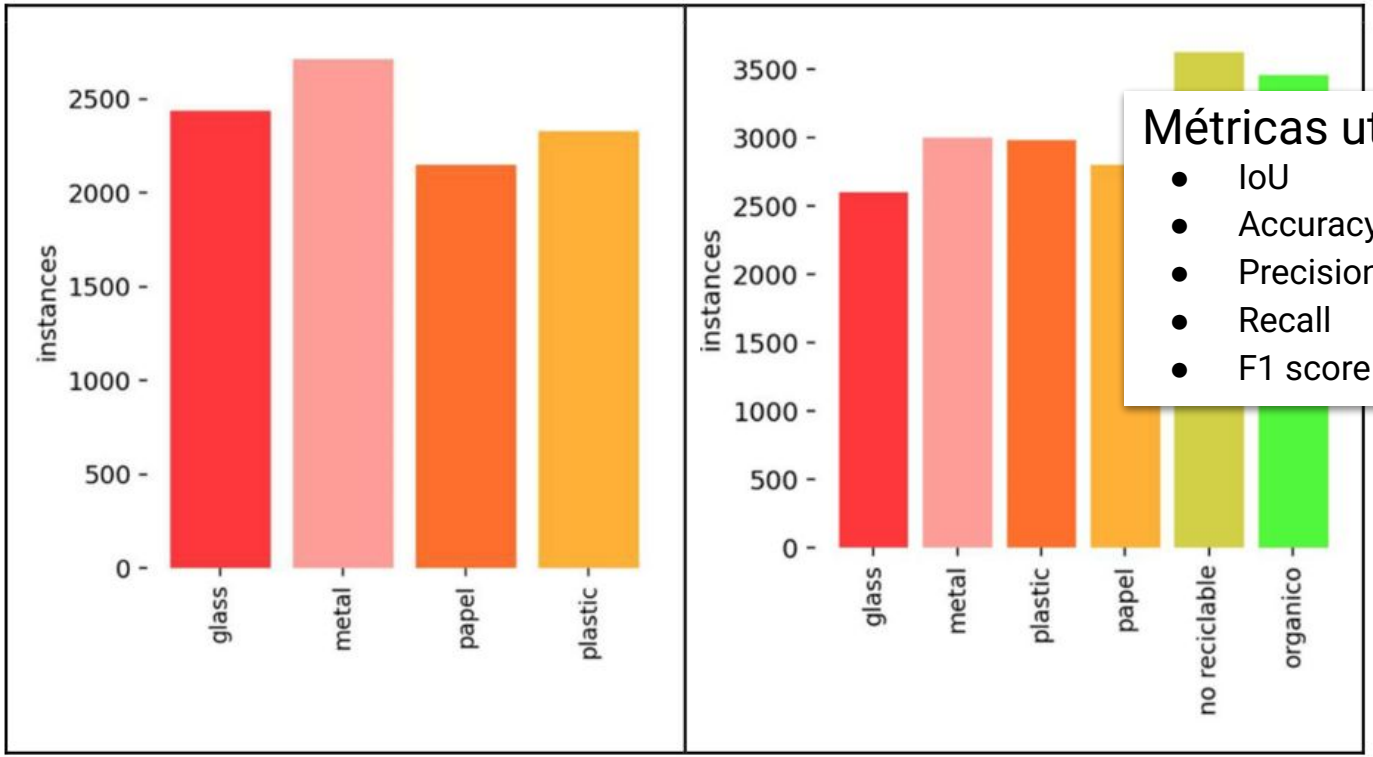


Fig. 10. Distribución del Conjunto de Números.

Reuse of a Deep Learning model for handwritten digit recognition
https://publicaciones.sadio.org.ar/index.php/EJS/article/view/841/679



Métricas utilizadas:

- IoU
- Accuracy
- Precision
- Recall
- F1 score

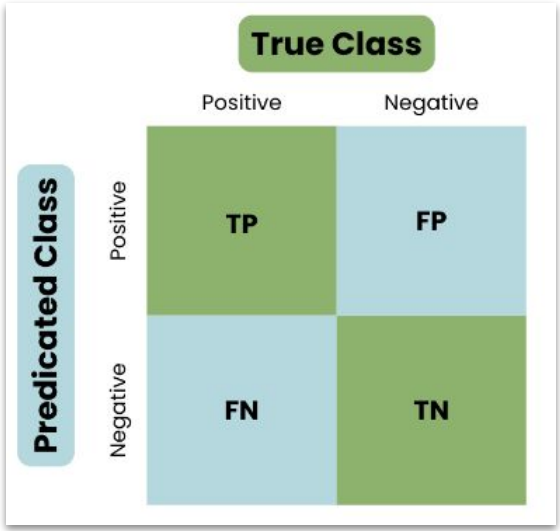
Fig. 4. Comparación de los dos enfoques utilizados

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

ERA ASÍ O TRASPUESTA?

Matriz de confusión - caso binario



TP: true positives

FP: false positives

TN: true negatives

FN: false negatives

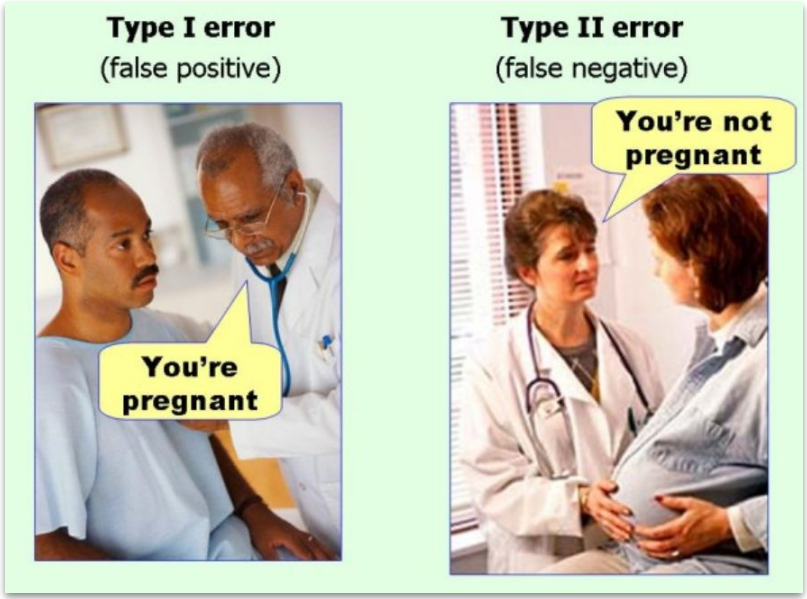
$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

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



Tomado de Towards Data Science

Medidas de performance

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

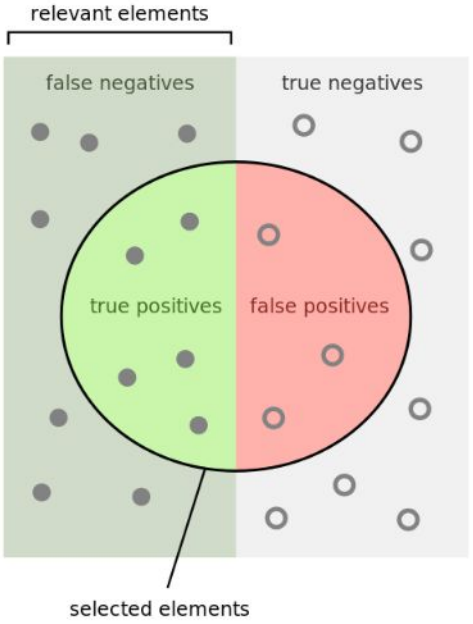
de las instancias clasificadas como positivas, cuántas lo son

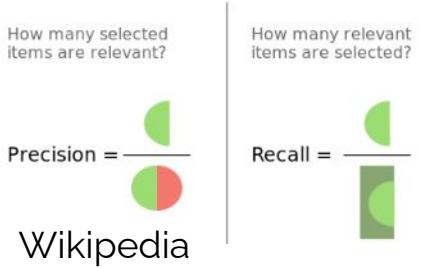
(cuán útiles son los resultados de búsqueda)

$$\text{Recall} = \frac{TP}{TP + FN}$$ (exhaustividad)

de las instancias positivas, cuántas fueron clasificadas como positivas

(cuán completos son los resultados)





Wikipedia

Medidas de performance

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

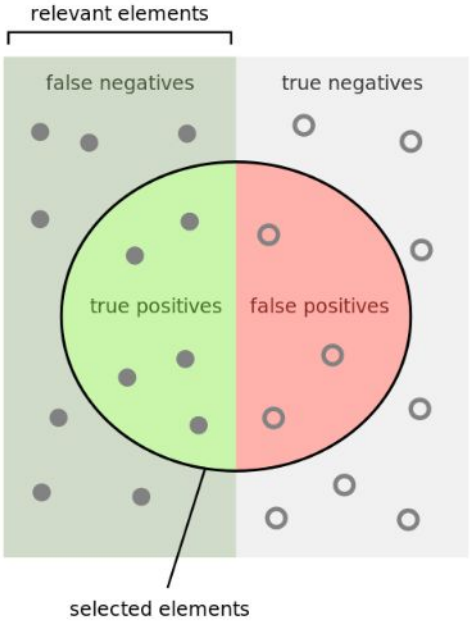
(cuán útiles son los resultados de búsqueda)

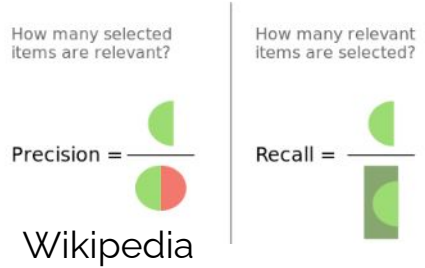
$$\text{Recall} = \frac{TP}{TP + FN}$$

(cuán completos son los resultados)

Se clasifican 4 como gatos (el primer y los últimos tres animales)

- TP: 3
- FP: 1
- P = 3/4, R = 3/3,





Wikipedia

Medidas de performance

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

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

- enfermedad contagiosa
- test de embarazo

Media armónica:

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

También llamada **F_1 score.**

Fórmula general:

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

F_2 da más peso a Recall
$F_{0.5}$ da más peso a Precisión

Medidas de performance

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

$$\frac{TN}{TN + FP} = \textit{Specificity o bien True Negative Rate}$$

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?

Medidas de performance

CURVA ROC (Receiver operating characteristic)

- **Gráfico TPR (Recall) vs. FPR**

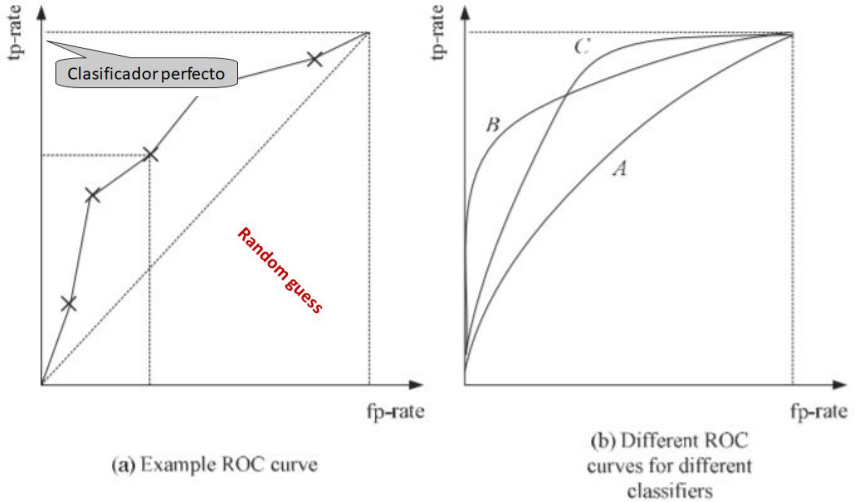
$$\text{Recall} = TPR = \frac{TP}{TP + FN}$$

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

Construcción: Variar el umbral de detección entre 0 y 100%. Para cada valor, calcular TPR y FPR (un punto en la curva).

Área bajo la curva (AUC)

- Un valor numérico. entre 0 y 1. Azar=0.5



Fuente: Introduction to ML, Alpaydin

Matriz de confusión n-aria - caso multiclase

	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**

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

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

Ejercicios

1. Medir la performance del modelo de clasificación generado para especies de árboles, de distintas maneras.
2. Para un problema genérico de clasificación binaria, definir una funcion matriz_confusion_binaria, que tome dos listas Y_test, Y_pred y devuelvas los valores (en orden) de TP, TN, FP, FN.

```
def matriz_confusion_binaria(Y_test, Y_pred):
    # Y_test e Y_pred deben ser listas de 0 y 1
    # completar
    return tp, tn, fp, fn
```

Ejercicios

3. Para un problema genérico de clasificación binaria, definir funciones para cada una de las siguientes métricas: accuracy, precision, recall, F1. Las funciones deben tomar como parámetros los TP, TN, FP, FN.

```
def accuracy_score(tp, tn, fp, fn):
    # completar
    return acc

def precision_score(tp, tn, fp, fn):
    # completar
    return prec
```

Ejercicios

4. Construir, usando sklearn, un árbol de decisión para el problema Titanic, y analizar su performance de distintas maneras.

Performance de un modelo - ¿dónde?

Medir la performance sobre datos de entrenamiento no es una buena idea. Surge la necesidad de separar un % de datos, para validar los modelos: datos de validación (o test).