DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Laboratorio de datos

Evaluación y selección de modelos

2do Cuatrimestre 2024

Laje, López-Rosenfeld, de Erausquin

(Agradecimientos: Manuela Cerdeiro - Paula Perez Bianchi - Viviana Cotik - Pablo Brusco)

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

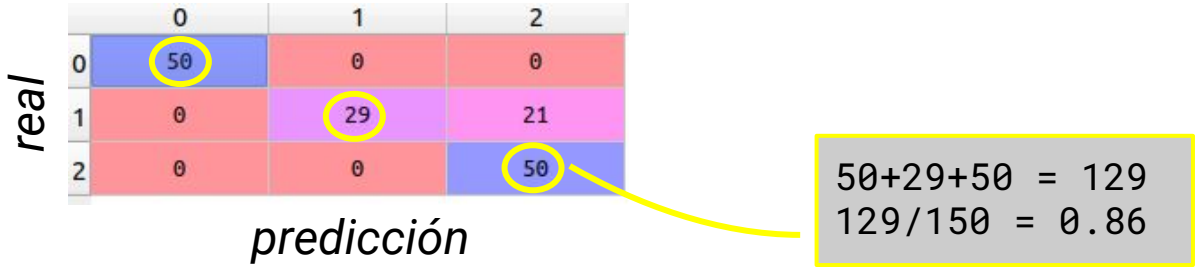
- Criterio de elección de atributos en cada nodo (Information Gain, Gini Gain...)
- Criterio de parada (ej: máxima profundidad)
- Estrategia de poda

Métricas en clasificación

Vimos: matriz de confusión y accuracy (exactitud).

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

Acc = $\sum_i M_{ii} / N$ (suma de los correctamente clasificados / total)



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

Ejemplos

Métricas utilizadas:

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

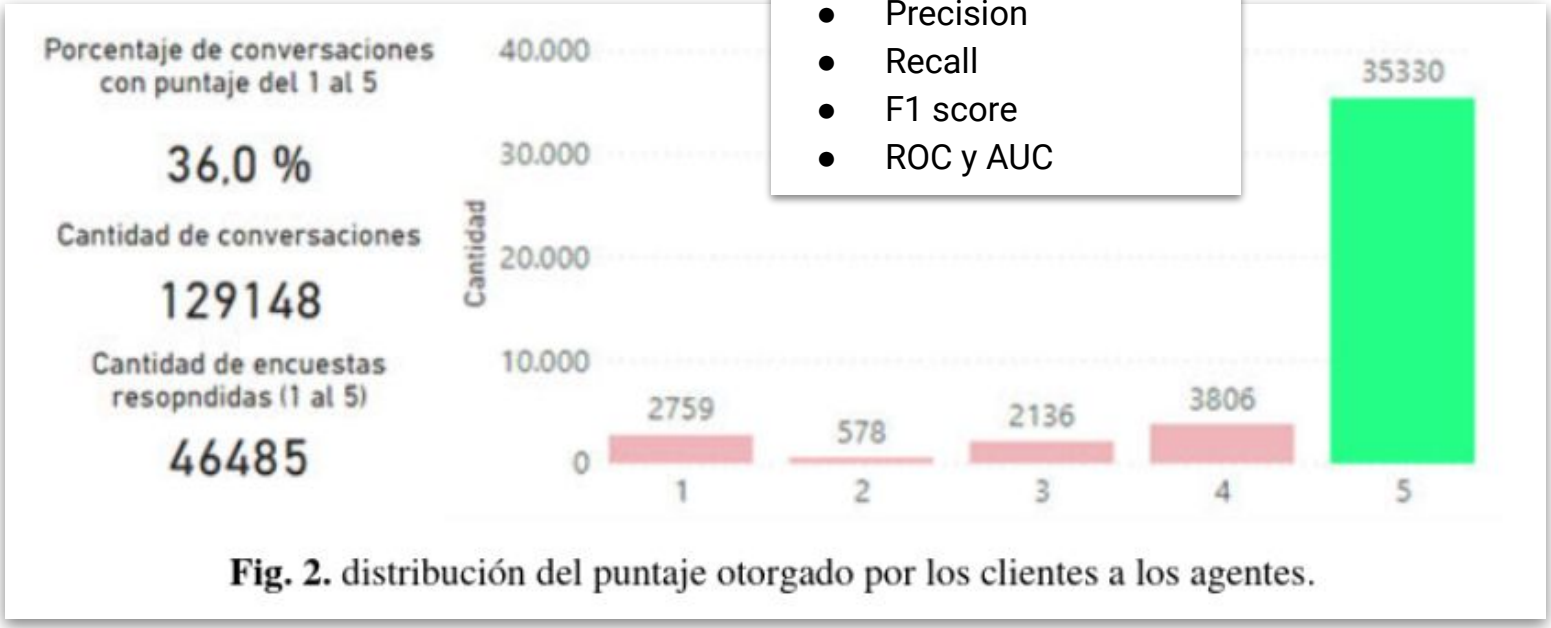


Fig. 2. distribución del puntaje otorgado por los clientes a los agentes.

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

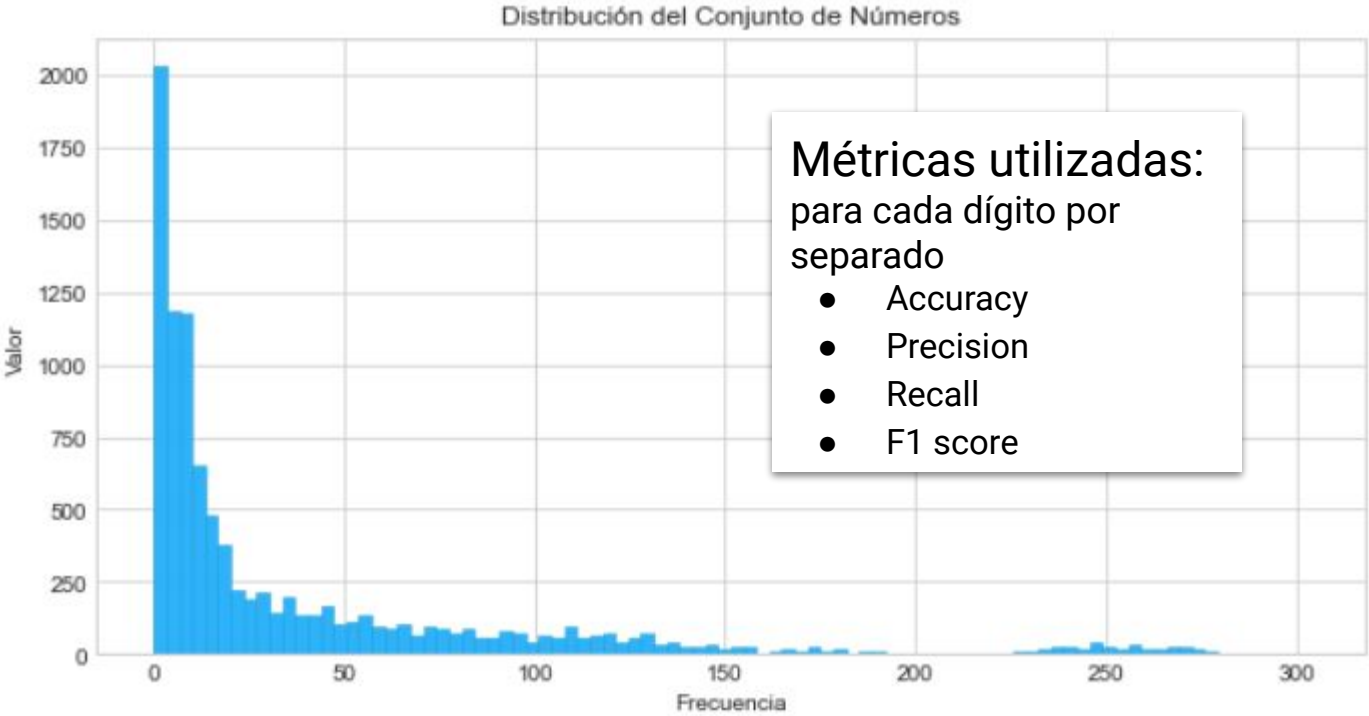
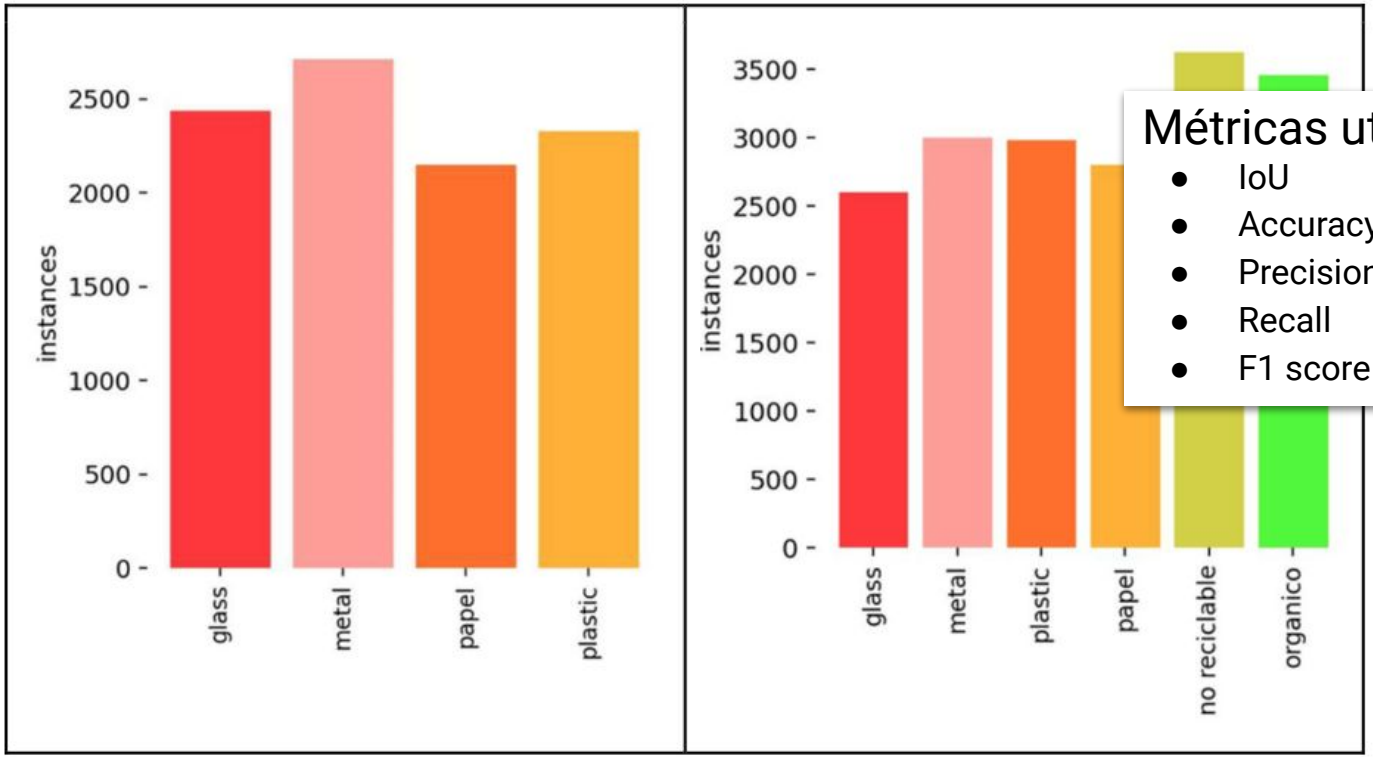


Fig. 10. Distribución del Conjunto de Números.

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

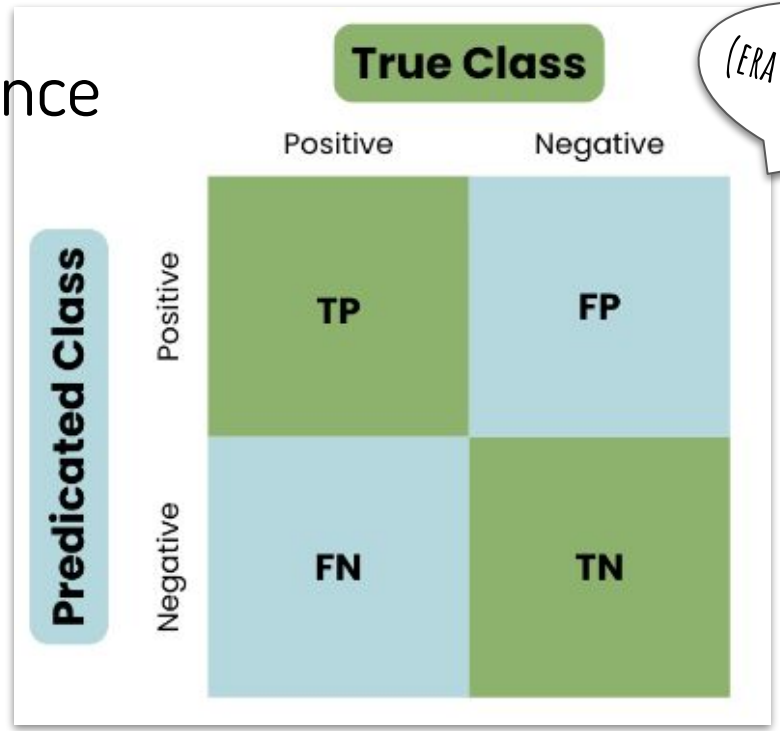
Ejemplo

Se trata de detectar una enfermedad. Se estima que la proporción de población enferma es del 1%.

Desarrollaron un test que tiene 90% de exactitud. ¿Es bueno?

Medidas de performance

Matriz de confusión - caso binario



(ERA ASÍ O TRASPUESTA?)

TP: true positives

FP: false positives

TN: true negatives

FN: false negatives

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

La exactitud está ok pero no dice nada sobre los tipos de aciertos y de errores que tiene el clasificador.

Ej: autenticación en aplicación por voz.

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

Ejemplo

1% de las mujeres tienen cáncer de mama. Desarrollaron un test que tiene la siguiente performance:

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

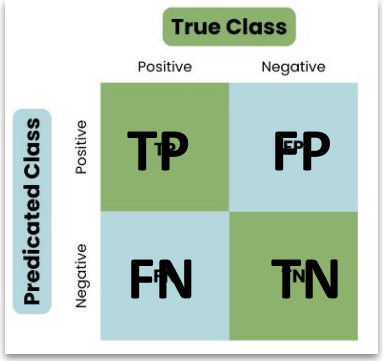
Es decir:

- 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



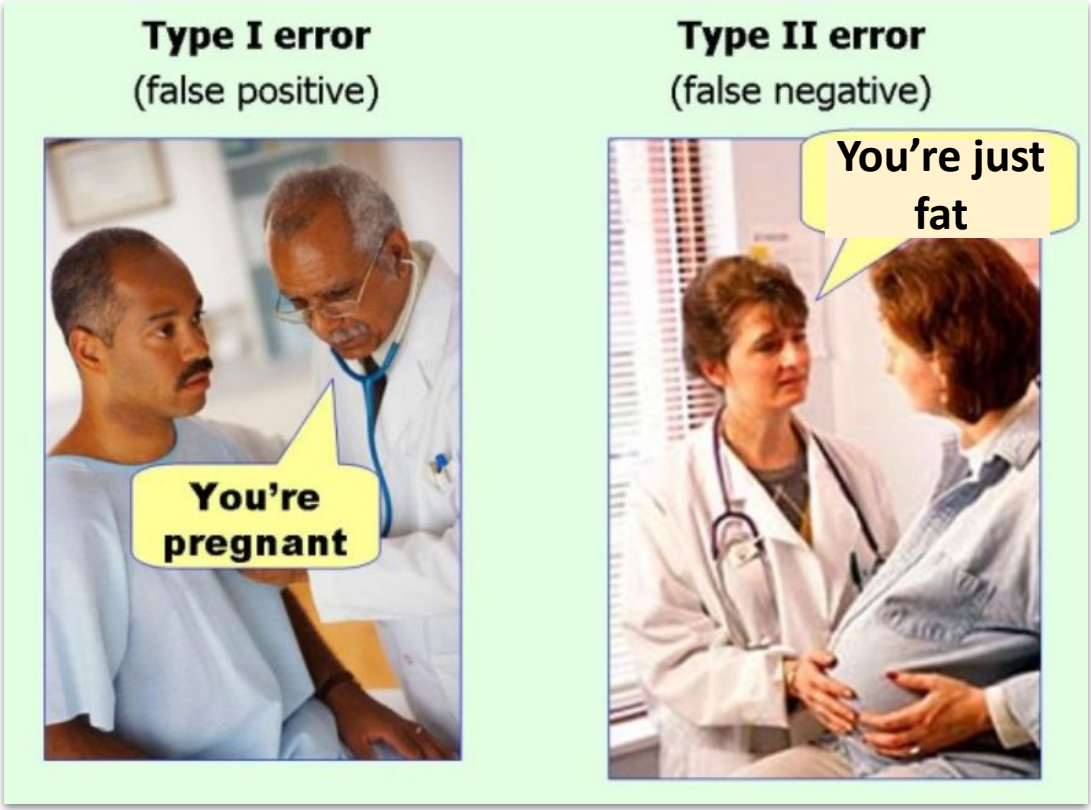
- ¿Cuánto es la exactitud? (porcentaje de clasificaciones correctas)
- ¿Si el test da positivo, qué quiere decir?

Resultado test	Casos reales (pos - neg)	
(pos)	0.008	0.09504
(neg)	0.002	0.89496

Exactitud = (TP+TN)/(TP+TN+FP+FN)
= (0.008+0.89496) / 1
= **0.90296**

Supongamos test positivo, cuál es la chance de que realmente sea positivo?
Cantidad de positivos reales en relación a todos los que dieron positivo
= TP / (TP + FP) = PRECISIÓN o Positive Predicted Value
= 0.008 / (0.008+0.09594) = **0.07764**

Tipos de error



(Towards Data Science)

Medidas de performance

$$\text{Precisión} = \frac{TP}{TP + FP}$$ (positive predictive value)

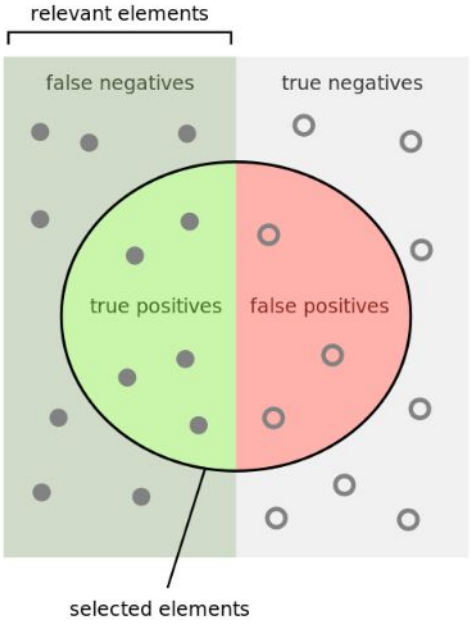
de las instancias clasificadas como positivas, cuántas lo son realmente

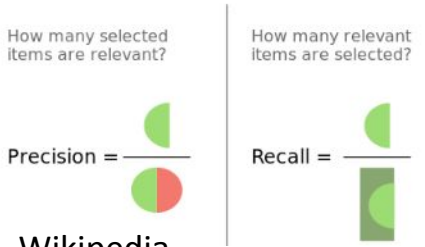
(“cuán útiles son los resultados”)

$$\text{Recall} = \frac{TP}{TP + FN}$$ (sensitivity, true positive rate, exhaustividad)

de las instancias realmente positivas, cuántas fueron clasificadas como positivas

(“cuán completos son los resultados”)





Wikipedia

Medidas de performance

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

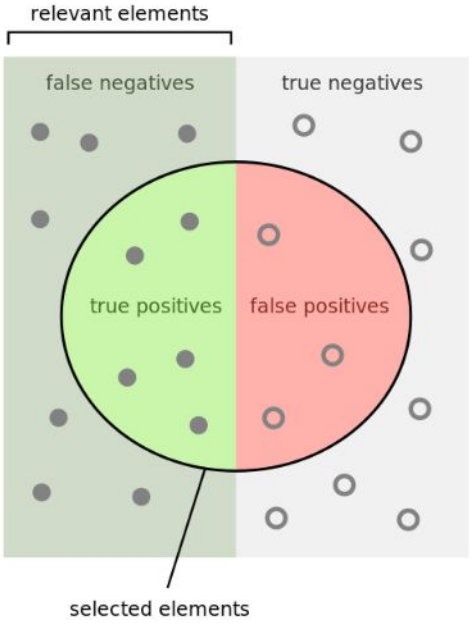
(“cuán útiles son los resultados”)

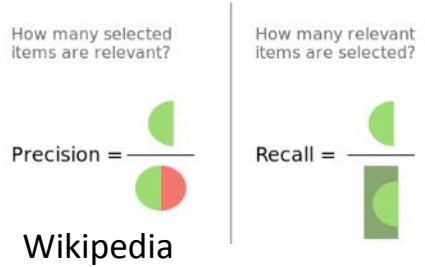
$$\text{Recall} = \frac{TP}{TP + FN}$$

(“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**

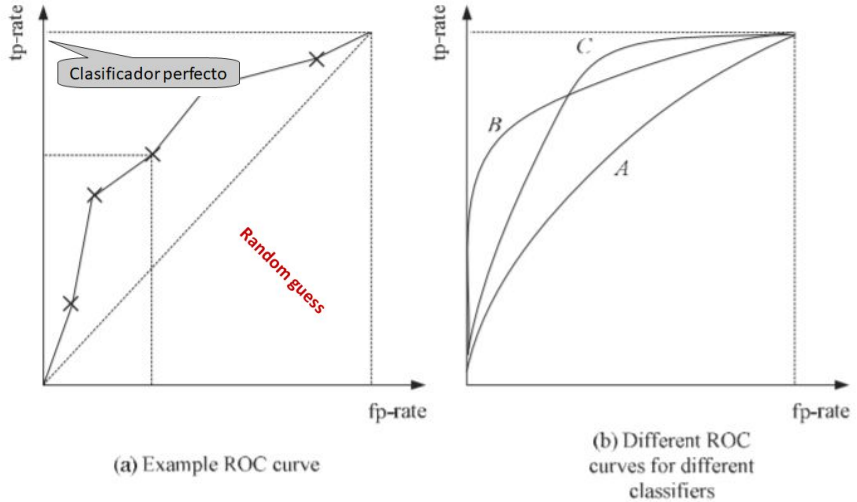
$$\text{Recall} = TPR = \frac{TP}{TP + FN}$$

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

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

Medidas de performance

CURVA ROC (Receiver operating characteristic)

- **Gráfico TPR (Recall) vs. FPR**

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

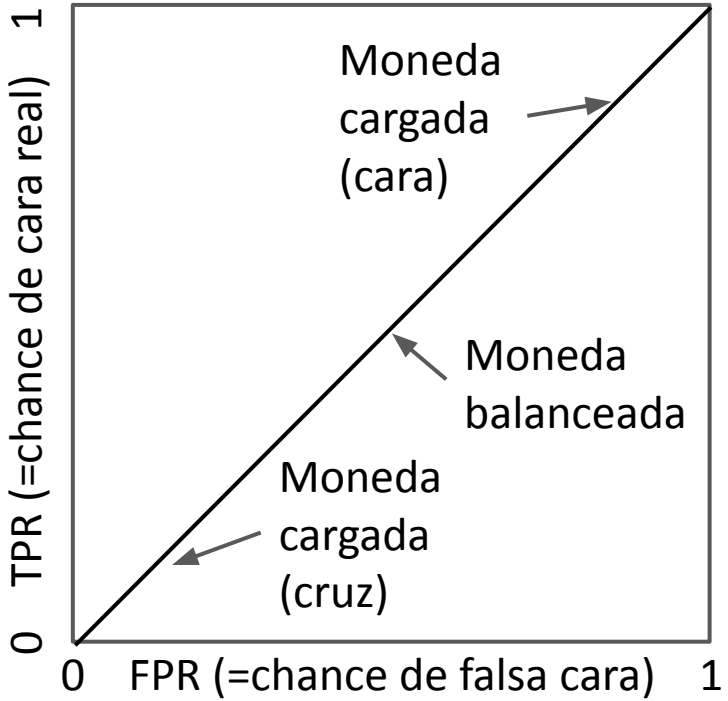
$$\text{FPR} = \frac{FP}{FP + TN}$$

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

Clasificador al azar: tirar una moneda muchas veces. Sale cara = positivo. Parámetro: cuán cargada está



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 resto**.

$$\text{Precisión}(Manzana) = \frac{MM}{MM + NM + OM + PM} \qquad \text{Recall}(Manzana) = \frac{MM}{MM + MN + MO + MP}$$

Guía de 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
```

Guía de 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
```

Guía de 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?

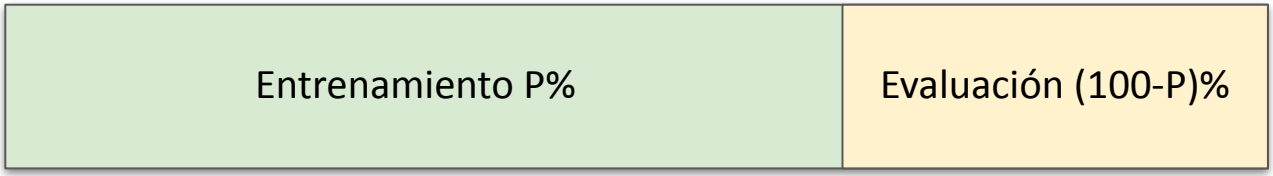
Medir la performance sobre datos de entrenamiento no es una buena idea. Surge la necesidad de separar un % de datos, para validar los modelos: datos de validación (o test).

Validación cruzada

Entrenamos nuestro modelo con **algunos** de nuestros datos, y vemos cómo funciona en los otros datos.

- al azar (elijo algunos, evalúo en otros)

¿Posibles problemas?



Validación cruzada

Entrenamos nuestro modelo con **algunos** de nuestros datos, y vemos cómo funciona en los otros datos.

- al azar (elijo algunos, evalúo en otros)
- al azar varias veces y promediar

¿Posibles problemas?

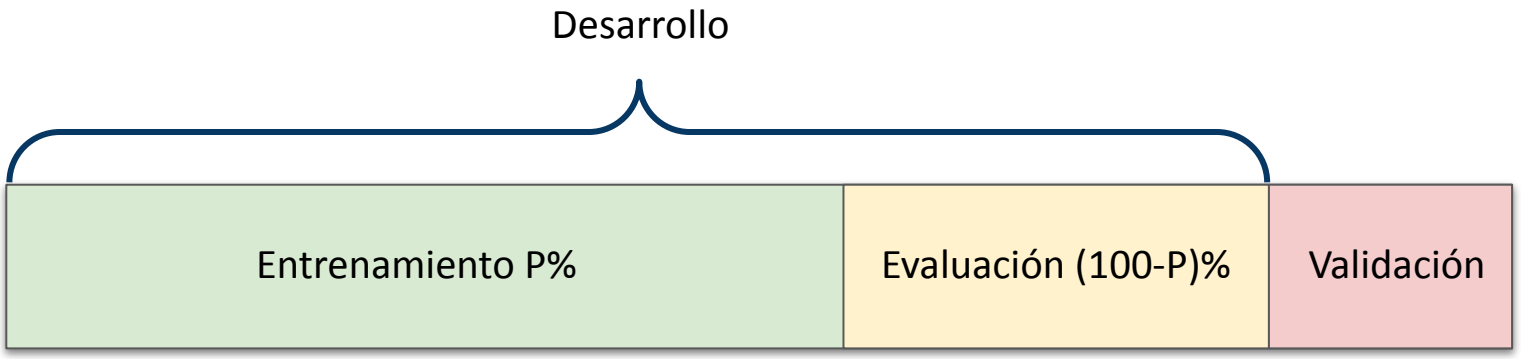
Validación cruzada

Entrenamos nuestro modelo con **algunos** de nuestros datos, y vemos cómo funciona en los otros datos.

- al azar (elijo algunos, evalúo en otros)
- al azar pero repitiendo varias veces (y promediando)
- leave one out
- k-folding

Validación cruzada

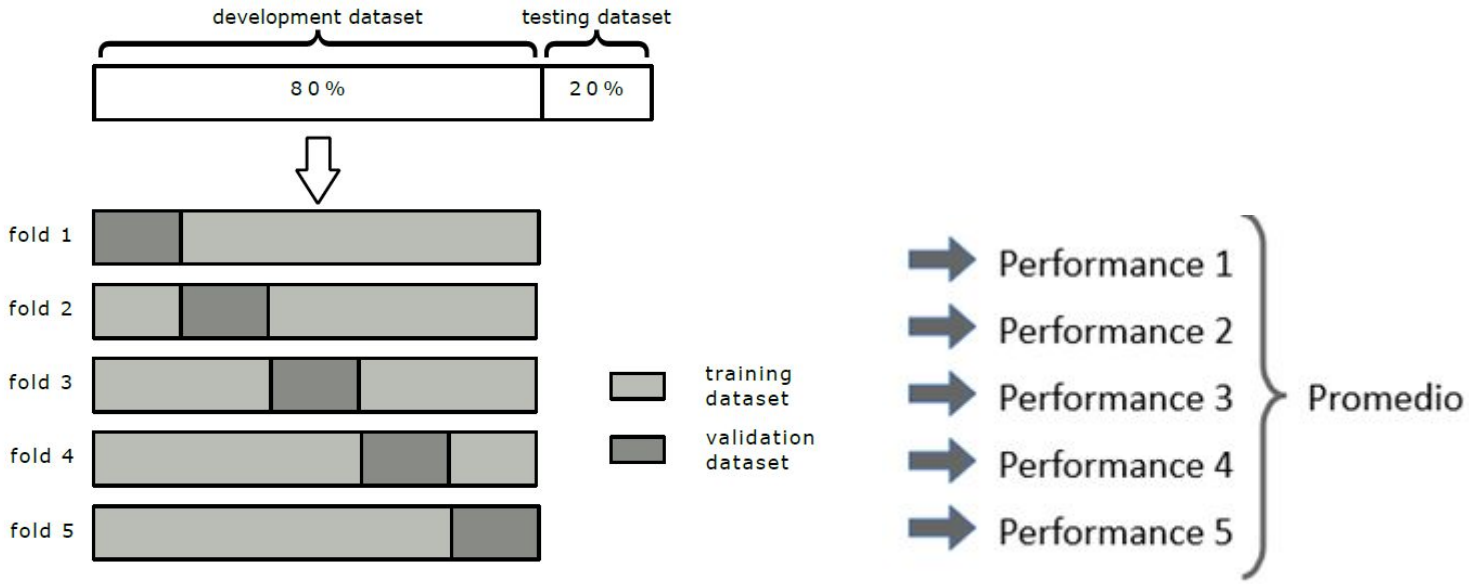
En realidad… si después queremos reportar la performance de nuestro mejor modelo.



K-Fold cross-validation

1. Desordenar los datos (o no).
2. Realizar una partición del conjunto en k folds disjuntos del mismo tamaño.
3. Para i = 1 ... k:
    - entrenar considerando todos los folds menos i.
    - predecir en el fold restante y medir la performance.
4. Promediar las métricas.

K-Fold cross-validation



Selección de modelos

Modelo: Cómo planeo modelar el proceso.
Por ejemplo, ¿árbol de decisión o SVM?

Hiperparámetros: Valores que se especifican previo a realizar el aprendizaje, es decir determinan el proceso de aprendizaje automático.

Por ejemplo, en árboles: el criterio de elección de atributos en cada nodo (Gini Gain, Information Gain), cantidad de hijos permitidos (árboles binarios vs. n-arios), criterio de parada (ej: max_depth), estrategia de poda, etc.

Parámetros: Valores internos del modelo resultante luego del aprendizaje, que se ajustan a partir de ejecutar el algoritmo sobre un dataset. Representan las reglas aprendidas.

Por ejemplo, en árboles: las preguntas a realizar en cada nodo del árbol.

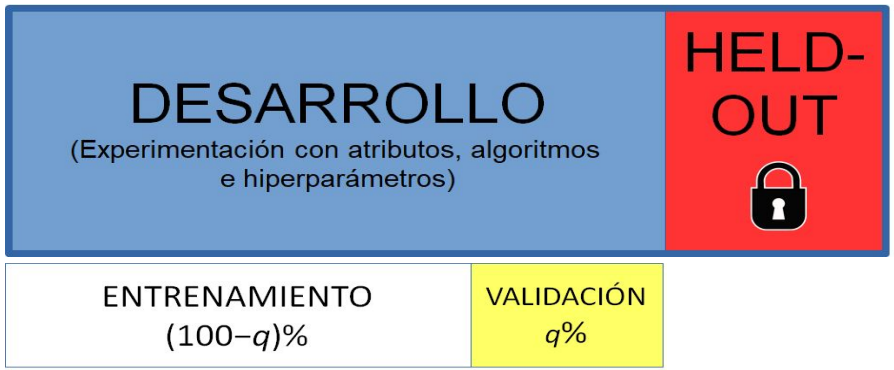
Selección de modelos

Al realizar validación cruzada estamos entrenando k veces un mismo modelo con mismos hiperparámetros, ajustado en distintos conjuntos, lo que da lugar a distintos parámetros. Y luego promediamos la performance, para tener una idea de la performance de la elección modelo+hiperparámetros.

Cuando hablamos de selección de modelos, pensamos en comparar un mismo modelo pero configurado y ajustado con distintas configuraciones de hiperparámetos, o quizás (menos usual) distintos modelos. Si estamos utilizando cross-val, evaluamos todos los modelos en la misma partición.

Conjunto held-out (o control o test)

- Al comenzar hay que **separar un conjunto de datos** (Held-Out, Control o **Test**) y **NO TOCARLOS** hasta el final
- Todas las pruebas y ajustes se realizan sobre el conjunto de **Desarrollo**
- Al terminar todas las pruebas, se evalúa el modelo obtenido con el conjunto Held-out



Ejercicios

- Utilizar los datos ‘seleccion_modelos.csv’. Se trata de un problema binario que queremos modelar utilizando un árbol de decisión.
- Separar un 10% de datos para el test final (held-out).
- Con los datos de desarrollo queremos utilizar k-fold cross-val para comparar distintas elecciones del hiperparámetro de altura, es decir ‘max_depth’.
- Hacer k-folding con k = 10, probar con alturas 1, 2, 3, 5, 8, 13, 21. Determinar cuál fue la mejor opción. Anotar la altura seleccionada y el score obtenido.
- Utilizando la altura seleccionada, entrenar un árbol de decisión ajustando al conjunto entero de los datos, evaluarlo en el mismo conjunto total de datos y anotar este valor.
- Probar el modelo recién entrenado en el conjunto held-out y reportar su performance.

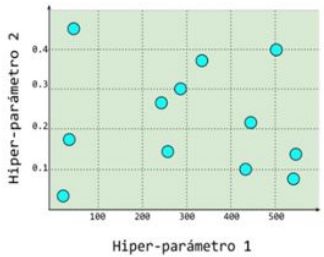
Selección de modelos

¿Cómo buscar la mejor combinación de modelo + hiperparámetros?

- Exploramos un espacio de búsqueda, **usando k-fold CV** para medir el desempeño de cada combinación.

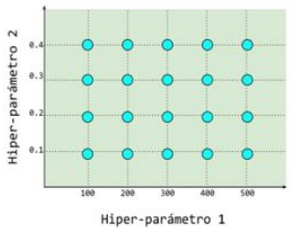
Random search (best guess, 1 factor at a time)

Explorar opciones y combinaciones al azar



Grid search

Plantear opciones y explorar todas las combinaciones

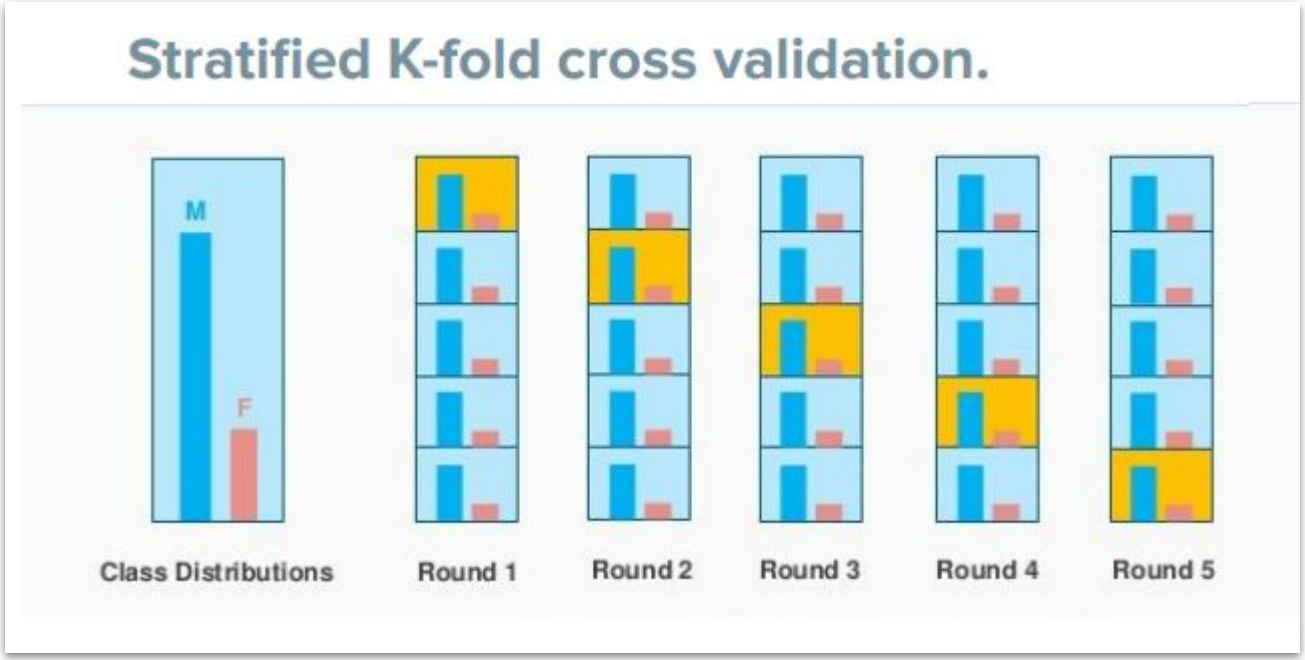


Al terminar, nos quedamos con la combinación con **mejor desempeño**,
y **entrenamos un único modelo usando todos los datos**

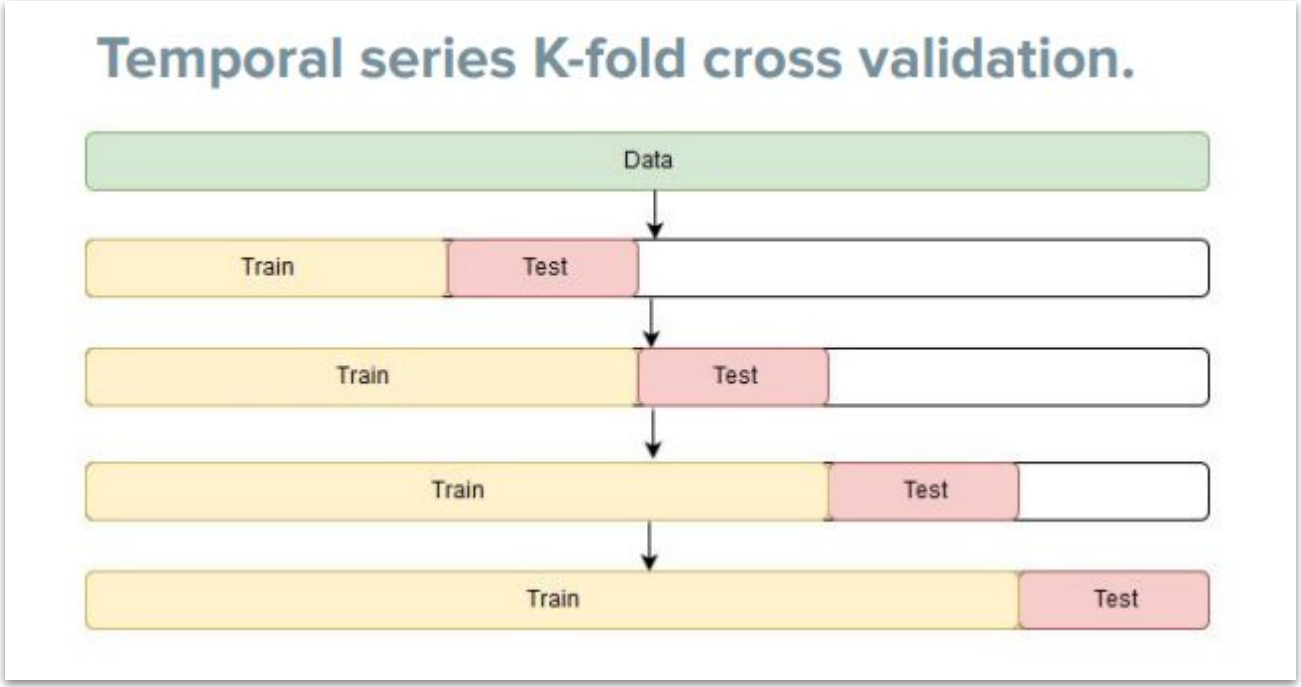
K-Fold cross-validation

Algunas preguntas

- Es necesario mezclar al azar?
- Es bueno mezclar al azar?
- Cómo están ordenados los datos?
- Cómo están balanceadas las clases?

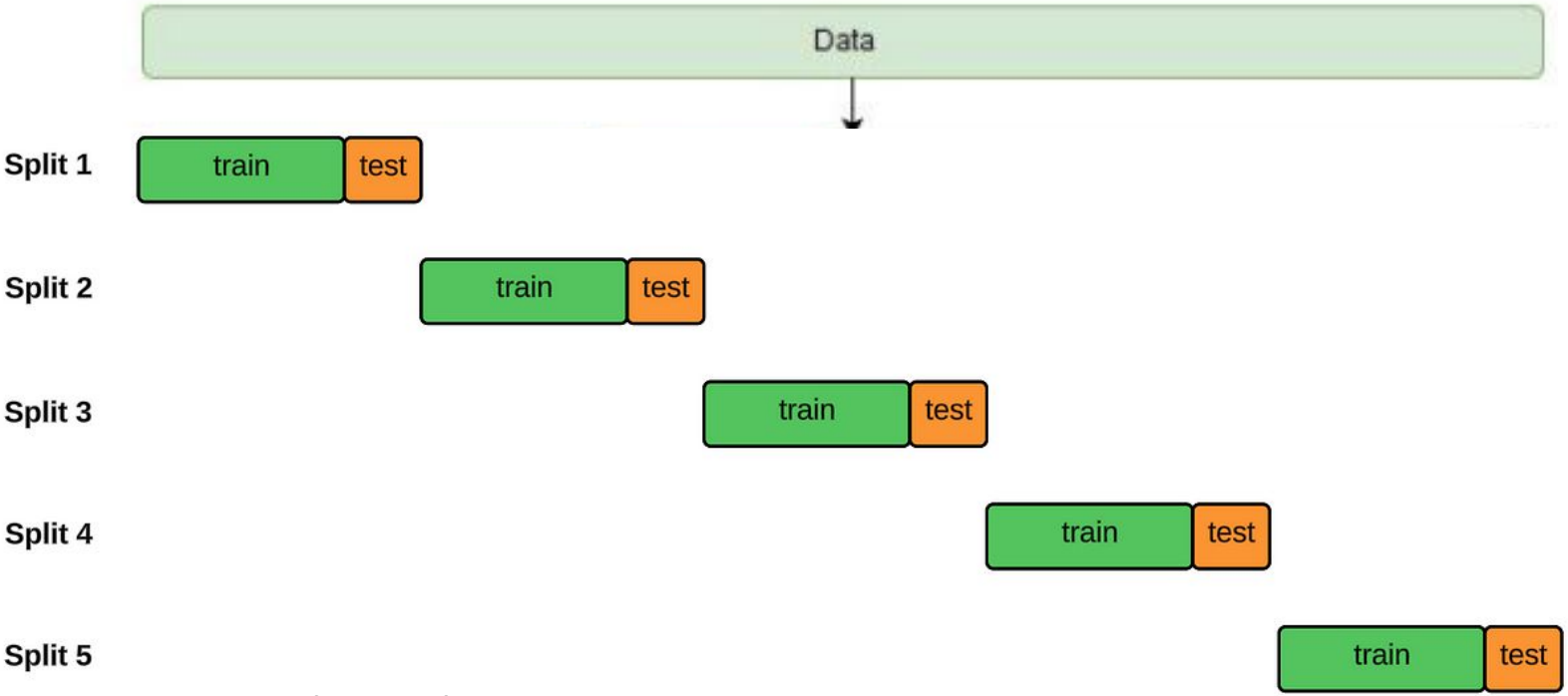


En cada fold, respetar las proporciones de ocurrencia de casos

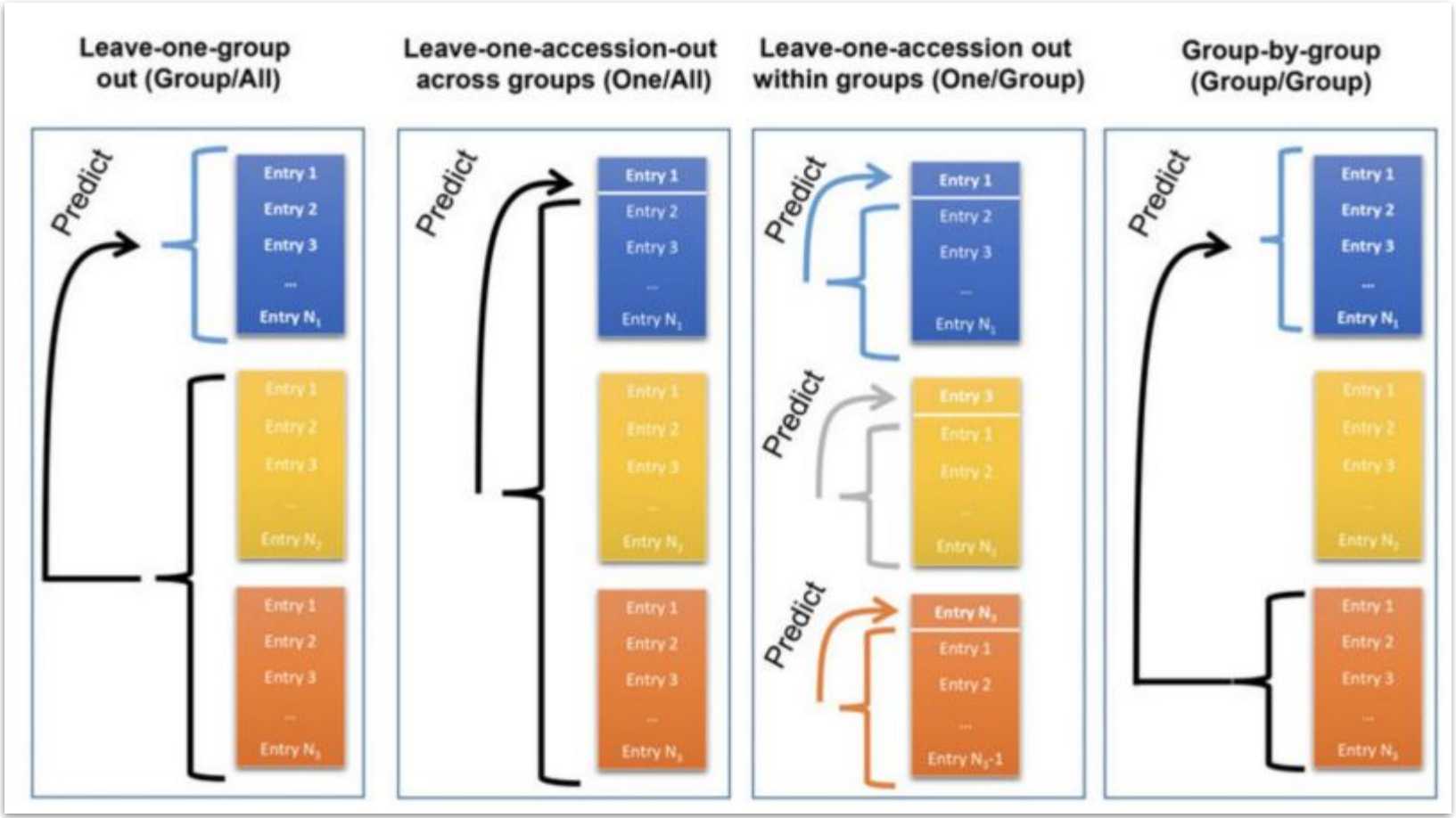


Quitás los datos “del futuro” para que no influyan en el entrenamiento del modelo

Blocked time series cross validation



Esta versión podría dar un mejor entrenamiento en caso de que los datos “actuales” dependan solamente del pasado reciente (agregar todo el pasado podría meter ruido)


Leave-one-group out (Group/All)
Leave-one-accession-out across groups (One/All)
Leave-one-accession out within groups (One/Group)
Group-by-group (Group/Group)
Predict
Entry 1
Entry 2
Entry 3
...
Entry N1
Entry N2
Entry N3
Entry N3-1

Ejercicios

Para el experimento antes realizado, probar las siguientes variantes y repetirlo.

- Probar realizar el experimento con otro valor de k para el k-folding.
- Probar variando otro hiperparámetro, por ejemplo el criterio.
- Probar variando más de un hiperparámetro a la vez.