Aprendizaje Automático

Segundo Cuatrimestre de 2018

Evaluación y Selección de Modelos

DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Aproximación de Funciones

Marco del problema:

- Conjunto de instancias X. Cada instancia $x \in X$ tiene atributos.
- Función objetivo desconocida $f: X \rightarrow Y$
- Espacio de hipótesis $H = \{ h \mid h : X \rightarrow Y \}$

Entrada del algoritmo de aprendizaje:

- Datos de entrenamiento $\{<x^{(i)}, y^{(i)}>\}$.

Salida del algoritmo de aprendizaje:

- Hipótesis (o modelo) $h \in H$ que aproxima a la función f.

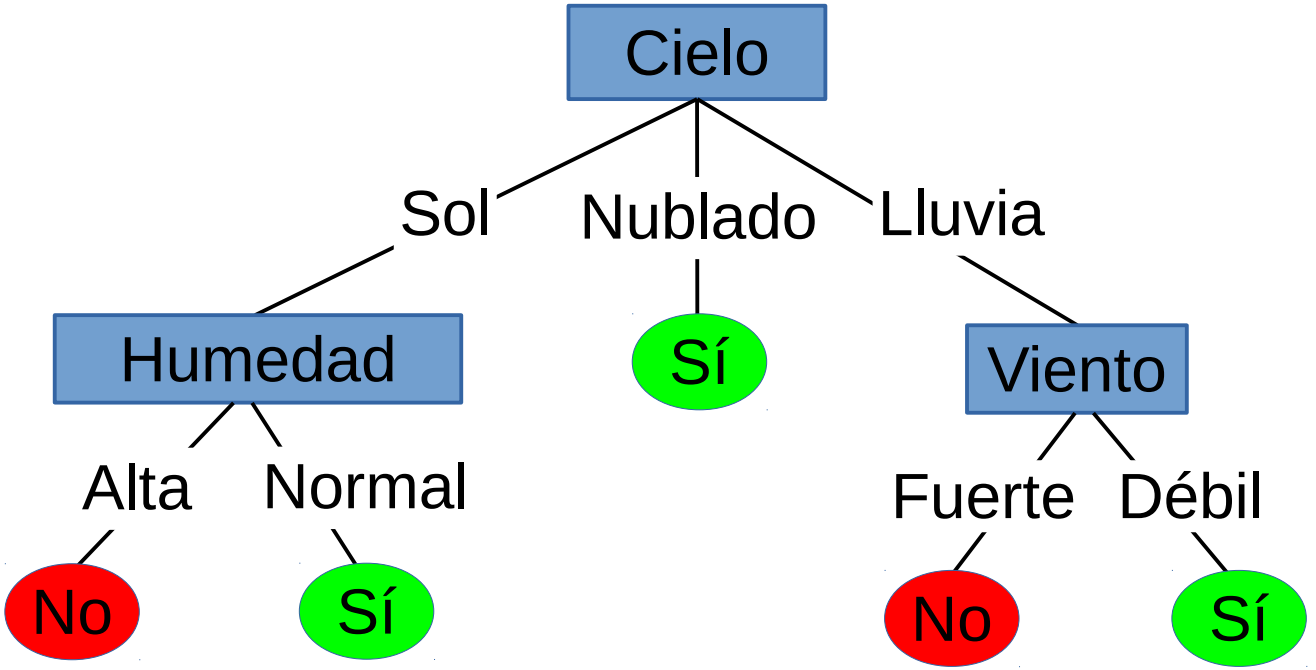
atributos

clase

instancias

Cielo	Temperatura	Humedad	Viento	¿Camina?
Sol	Calor	Alta	Débil	No
Sol	Calor	Alta	Fuerte	No
Nublado	Calor	Alta	Débil	Sí
Lluvia	Templado	Alta	Débil	Sí
Lluvia	Frío	Normal	Débil	Sí
Lluvia	Frío	Normal	Fuerte	No
Nublado	Frío	Normal	Fuerte	Sí
Sol	Templado	Alta	Débil	No
Sol	Frío	Normal	Débil	Sí
Lluvia	Templado	Normal	Débil	Sí
Sol	Templado	Normal	Fuerte	Sí
Nublado	Templado	Alta	Fuerte	Sí
Nublado	Calor	Normal	Débil	Sí
Lluvia	Templado	Alta	Fuerte	No

Árboles de Decisión



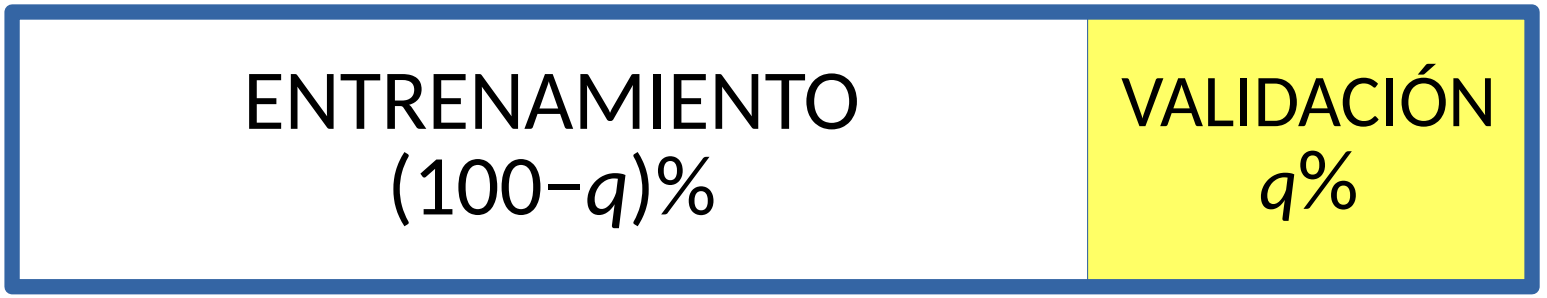
- $h : \langle X_1, \ldots, X_n \rangle \rightarrow Y$
- Cada nodo interno evalúa un atributo discreto X_i
- Cada rama corresponde a un valor para X_i
- Cada hoja predice un valor de Y

Evaluación de Modelos

- ¿Cómo sabemos cuán bueno es nuestro modelo?
  - El concepto es desconocido. ¿Qué usamos cómo referencia?
- Primera idea:
  - *Accuracy* (eficacia): Porcentaje de datos de entrenamiento clasificados correctamente.
  - Mala idea.
    - Un modelo podría simplemente memorizar los casos de entrenamiento y tener *accuracy* de 100%.
    - **Medir performance sobre los datos de entrenamiento tiende a sobre-estimar los resultados.**

Validación Cruzada

- ¿Cómo estimamos la performance de nuestro modelo?
- Medir accuracy sobre datos de entrenamiento → mala idea.
- Surge la necesidad de separar un q% de datos, para validar los modelos: datos de validación (o test).



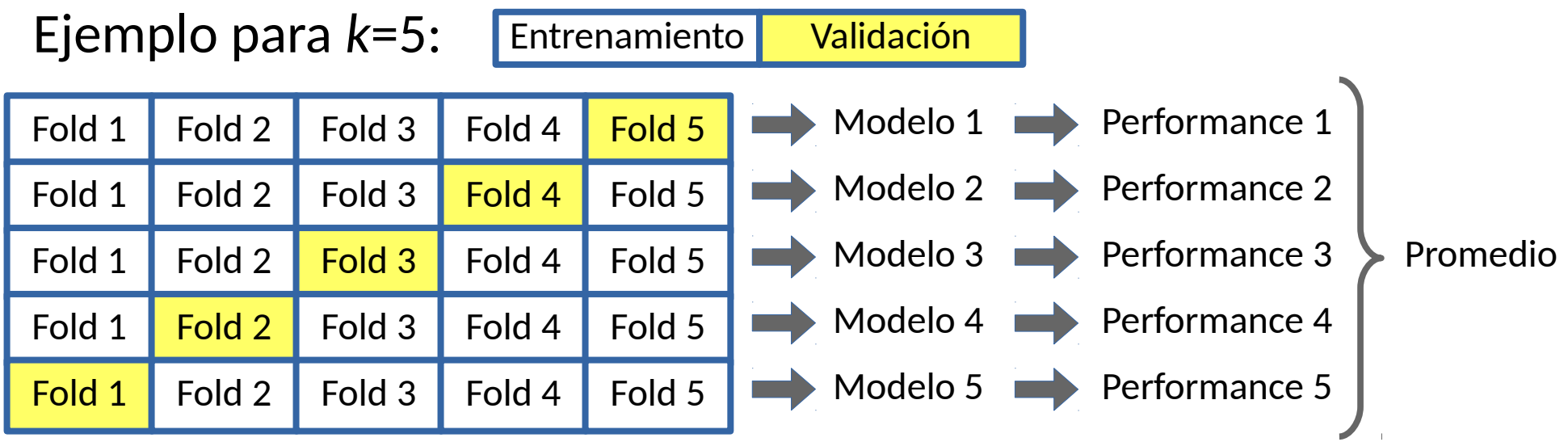
La expresión “test” es ambigua. En esta materia preferimos hablar de datos de “validación”, y evitar “test”.

- Los datos se deben separar al azar (*), para evitar cualquier orden o estructura subyacente en los datos.

(*) Esto no siempre es así. Los datos de entrenamiento y validación deben ser independientes entre sí (ej: en ASR una persona no debe aparecer en ambos); los datos pueden estar desabalanceados, tener orden temporal, etc.

Validación Cruzada

- ¿Qué puede pasar si tenemos mala suerte al separar los datos para entrenamiento/validación?
  - La estimación de performance del modelo podría no ser realista.
- Para disminuir este riesgo: *k*-Fold Cross Validation
  1) Desordenar los datos.
  2) Separar en *k* folds del mismo tamaño.
  3) Para *i* = 1 .. *k*: entrenar sobre todos menos *i*; evaluar sobre *i*.
- Ejemplo para *k*=5:



Comparando Modelos

- Queremos comparar 2 modelos: M_1, M_2
- Opción 1: Comparar la accuracy media de cada modelo.
  - Contra: No tenemos forma de saber la significancia estadística de las diferencias halladas. ¿Podría ser una consecuencia del azar?
- Opción 2 (mejor): Comparar los vectores de resultados de los *k* folds con un test estadístico:

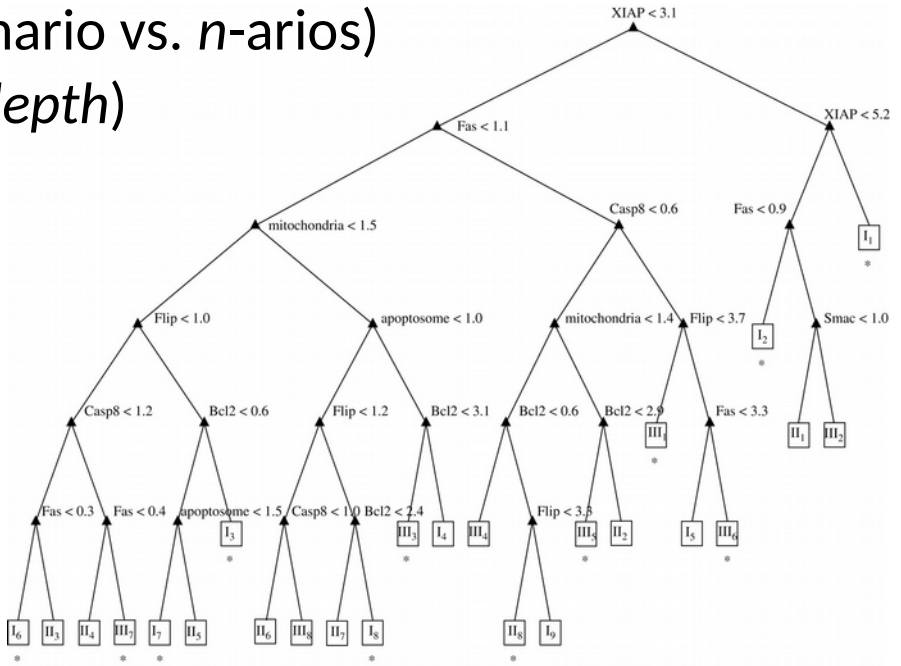
  1) *k*-fold CV para M_1: $\mathbf{Acc_1} = \langle Acc_{1,1}, Acc_{1,2}, \ldots, Acc_{1,k} \rangle$
  2) *k*-fold CV para M_2: $\mathbf{Acc_2} = \langle Acc_{2,1}, Acc_{2,2}, \ldots, Acc_{2,k} \rangle$

  (Deben usarse los mismos folds para cada modelo.)

  3) Test apareado entre $\mathbf{Acc_1}$ y $\mathbf{Acc_2}$.
     - *Wilcoxon signed-rank test* (versión no paramétrica del *paired t-test*).
     - Output del test: *p*-valor, que nos dice el grado de significancia estadística de la diferencia entre la performance de ambos modelos. Por ejemplo: $p<0.05 \rightarrow$ diferencia significativa

Selección de Modelos

- ¿Por qué tendríamos distintos modelos para comparar?
  - Distintos atributos (selección y transformación de atributos)
  - Distintos algoritmos (árboles, LDA, NB, KNN, SVM, DNN, ...)
  - Distintos hiperparámetros de cada algoritmo.
    - Ejemplo: hiperparámetros de los árboles de decisión
      - Criterio de elección de atributos en cada nodo (GiniGain, ...)
      - Cantidad de hijos (árboles binario vs. *n*-arios)
      - Criterio de parada (ej: *max_depth*)
      - Estrategia de poda
      - ...

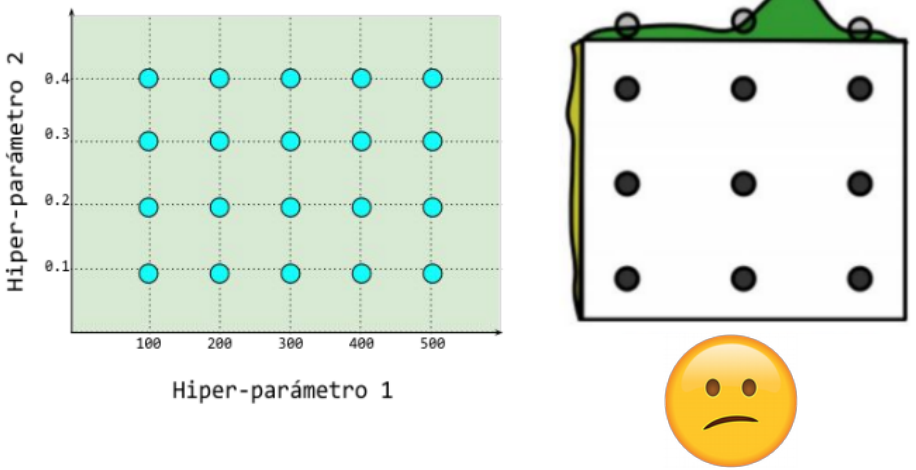


Selección de Modelos

- ¿Cómo buscar la mejor combinación de atributos + algoritmos + hiperparámetros?
  - Exploramos un espacio de búsqueda, usando *k*-fold CV para medir el desempeño de cada combinación.

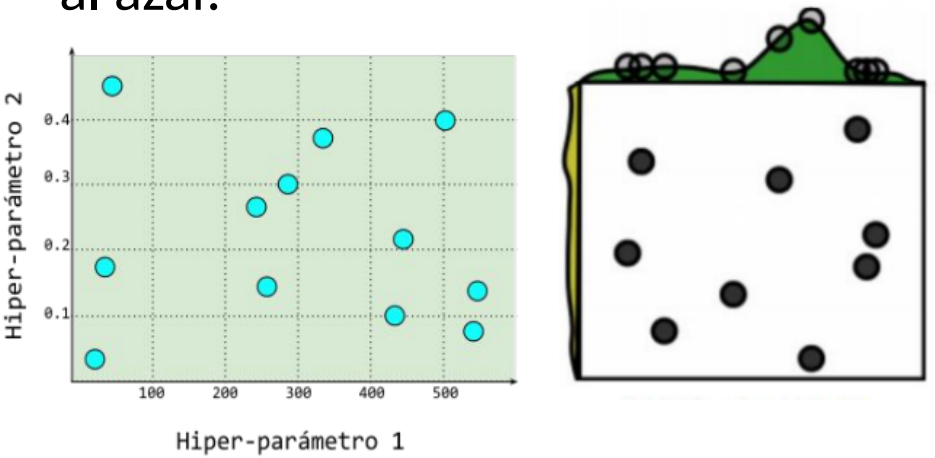
Grid search

Plantear opciones y explorar todas las combinaciones.



Random search

Explorar opciones y combinaciones al azar.



Bergstra & Bengio. "Random search for hyper-parameter optimization." Journal of Machine Learning Research 2012.

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

Selección de Modelos

- Escenario frecuente:
  - Construimos nuestro modelo con la combinación de atributos + algoritmos + hiperparámetros con mejor desempeño.
  - Lo ponemos a funcionar con datos nuevos, y los resultados son peores.
- ¿Qué falló?
- Otra vez sopa… A otro nivel, repetimos el mismo error de antes. Evaluamos un modelo sobre los mismos datos que usamos para construirlo.

- Entonces, sobreestimamos la performance de nuestro modelo.
- ¿Solución?

Conjunto Held-Out (o de Control)

- Lo antes posible, hay que separar un conjunto held-out de datos (o de control), y NO TOCARLOS hasta el final.
- Todas las pruebas y ajustes se hacen sobre el conjunto de datos de desarrollo (*dev set*).
- Cuando termina el desarrollo, se evalúa sobre los datos held-out. La estimación de performance será más realista.
- ¡No volver atrás!

Medidas de Performance

- Un modelo tiene una *accuracy* (eficacia) del 95%.
  - O sea, de cada 100 instancias, clasifica bien 95.
- ¿Qué significa esto?
- Según la tarea y la distribución de clases en el dominio, 95% puede ser muy bueno o pésimo.
- No dice nada sobre el tipo de aciertos y errores que comete el modelo.
- Ejemplos:
  - **Filtro de spam:** descarta directamente los mails sospechosos.
  - **Detección de fraude:** prepara un listado de casos sospechosos para ser revisados por humanos.
  - Identificación de meteoritos. xkcd.com/1723 :-)
- Veamos otras medidas de performance…

Matriz de Confusión: (Clasificación binaria)

tp: true positives
tn: true negatives
fp: false positives
fn: false negatives

	SPAM (predicho)	NO SPAM (predicho)
SPAM (real)	2739 **tp**	56 **fn**
NO SPAM (real)	4 **fp**	1042 **tn**

Precisión y Recall *("exhaustividad")*:

(Terminología de Recuperación de la Información)

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

De los documentos recuperados, qué porcentaje son relevantes.

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

De los documentos relevantes, qué porcentaje fueron recuperados.

Documento recuperado = Positivo predicho (ej: mail clasificado como spam por el modelo)

Documento relevante = Positivo real (ej: mail anotado como spam por el usuario)

Ejemplos de Aprendizaje Automático:

¿Cuál medida (p/r) debería priorizar cada uno de estos sistemas?

- Filtro de spam: descarta directamente los mails sospechosos.
- Detección de fraude: prepara un listado de casos sospechosos para ser revisados por humanos.

Matriz de Confusión: (Clasificación binaria)

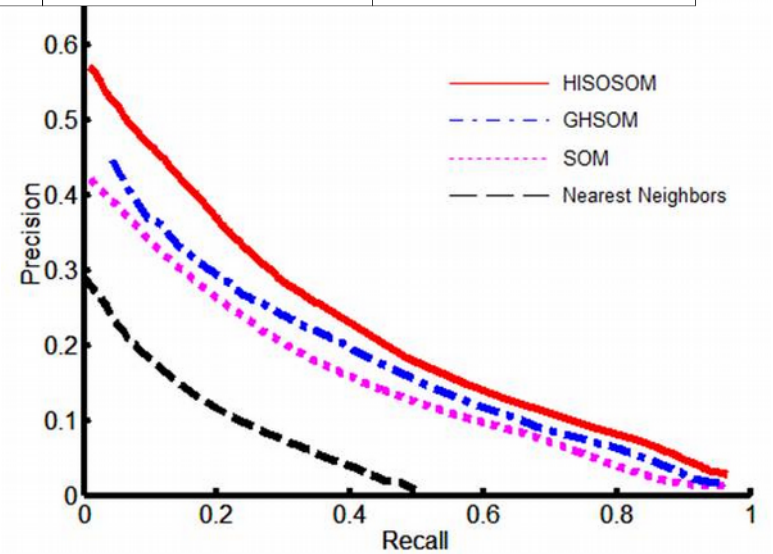
tp: true positives
tn: true negatives
fp: false positives
fn: false negatives

	Positivo (predicho)	Negativo (predicho)
Positivo (real)	**tp**	**fn**
Negativo (real)	**fp**	**tn**

Precisión y Recall *("exhaustividad")*:

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

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



$$F\text{-measure} = 2 \cdot \frac{\text{precisión} \cdot \text{recall}}{\text{precisión} + \text{recall}}$$

Media armónica. También llamada F_1 score.

Fórmula general: $F_\beta = (1 + \beta^2) \cdot \frac{\text{precisión} \cdot \text{recall}}{(\beta^2 \cdot \text{precisión}) + \text{recall}}$

F_2 enfatiza recall; $F_{0.5}$ enfatiza precisión.

Matriz de Confusión: (Clasificación binaria)

tp: true positives
tn: true negatives
fp: false positives
fn: false negatives

	Positivo (predicho)	Negativo (predicho)
Positivo (real)	**tp**	**fn**
Negativo (real)	**fp**	**tn**

Sensibilidad y Especificidad (Medicina, Biología):

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

$$\text{Recall} = \frac{tp}{tp+fn} = \text{Sensitivity o bien True Positive Rate}$$

$$\frac{tn}{tn+fp} = \text{Specificity o bien True Negative Rate}$$

Sensitivity: Porcentaje de pacientes enfermos correctamente diagnosticados.
Specificity: Porcentaje de pacientes sanos correctamente diagnosticados.

"...the use of repeatedly reactive enzyme immunoassay followed by confirmatory Western blot or immunofluorescent assay remains the standard method for diagnosing HIV-1 infection. A large study of HIV testing in 752 U.S. laboratories reported a sensitivity of 99.7% and specificity of 98.5% for enzyme immunoassay"

Chou R et al., "Screening for HIV: A review of the evidence for the U.S. Preventive Services Task Force", Annals of Internal Medicine, 143 (1): 55-73. 2005.

Matriz de Confusión: (Clasificación binaria)

tp: true positives
tn: true negatives
fp: false positives
fn: false negatives

	Positivo (predicho)	Negativo (predicho)
Positivo (real)	**tp**	**fn**
Negativo (real)	**fp**	**tn**

Curva ROC:

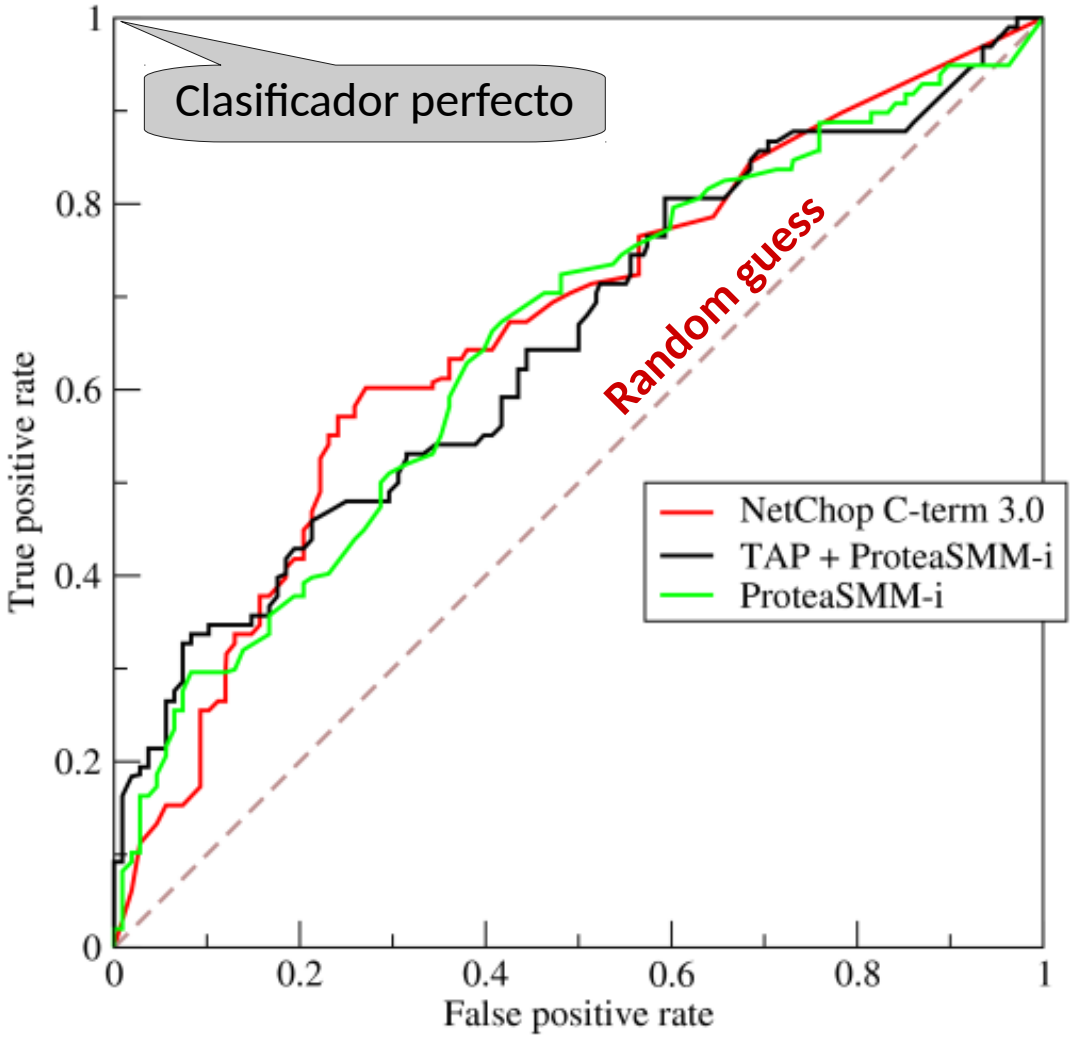
"Receiver operating characteristic"

Gráfico TPR (recall) vs. FPR.

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

$$\text{FPR} = \frac{\text{fp}}{\text{fp}+\text{tn}}$$

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)

Entre 0 y 1. Random = 0.5.

Matriz de Confusión: (Clasificación *n*-aria)

	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. solo pueden formularse en forma binaria: cada clase contra el resto.

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

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

Resumen

- ¡Cuidado con la sobreestimación de resultados!
- Validación cruzada (datos de entrenamiento vs. validación).
- *k*-fold cross validation.
- Selección de modelos: grid search y random search.
- Conjuntos de datos: desarrollo, held-out.
- *Accuracy* (eficacia).
- Matriz de confusión, precision, recall.
- Sensibilidad y especificidad.
- Curva ROC y área bajo la curva.