

Aprendizaje Automático

Evaluación de modelos

Viviana Cotik

Gran parte del material tomado de
**Evaluación y Selección de Modelos: Materia Aprendizaje
automático, DC, FCEyN, UBA.**

Profesor: Agustín Gravano



1er cuatrimestre 2019

Aproximación de funciones

Problema:

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

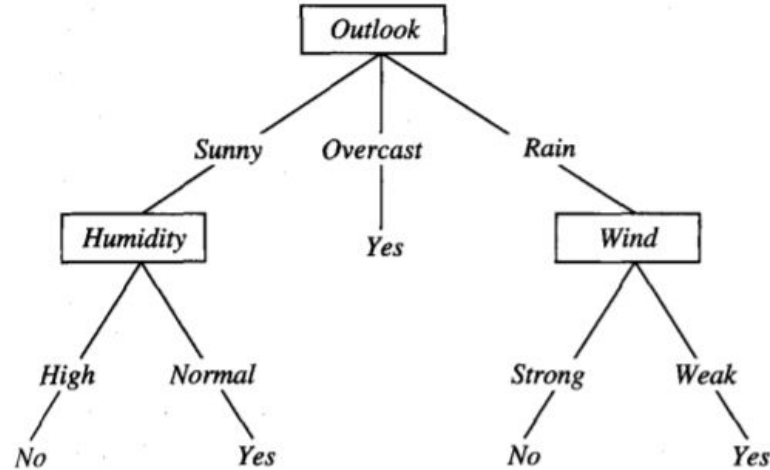
Entrada del algoritmo de aprendizaje:

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

Salida del algoritmo de aprendizaje:

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

Árboles de decisión



Machine Learning, Mitchel

- $h : \langle X_1, \dots, X_n \rangle \rightarrow Y$
- 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?
- ¿Sobre qué conjunto de datos lo medimos?

Primera idea:

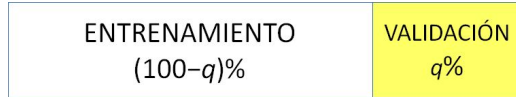
- **Accuracy (eficacia) sobre cjto. entrenamiento:** Porcentaje de datos de entrenamiento clasificados correctamente.:
- Mala idea:
 - El modelo puede **memorizar** los datos de entrenamiento y tener **accuracy de 100%**.
 - Medir **performance sobre los datos de entrenamiento** tiende a **sobreestimar los resultados**.

Conjunto Held-Out (Control, Test)

- Al comenzar hay que **separar un conjunto de datos** (Held-Out, 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



Estimación de performance



Se prueban distintas configuraciones.

Los datos de desarrollo se separan en:

- $(100-q)$ % para entrenamiento
- q % para validación del modelo

Una vez definido cuál es el mejor modelo se entrena con cjto. de **desarrollo** (también llamado de **entrenamiento**)

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; los datos pueden estar desbalanceados, tener orden temporal, etc.

Validación cruzada (o cross-validation)

¿Y 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**

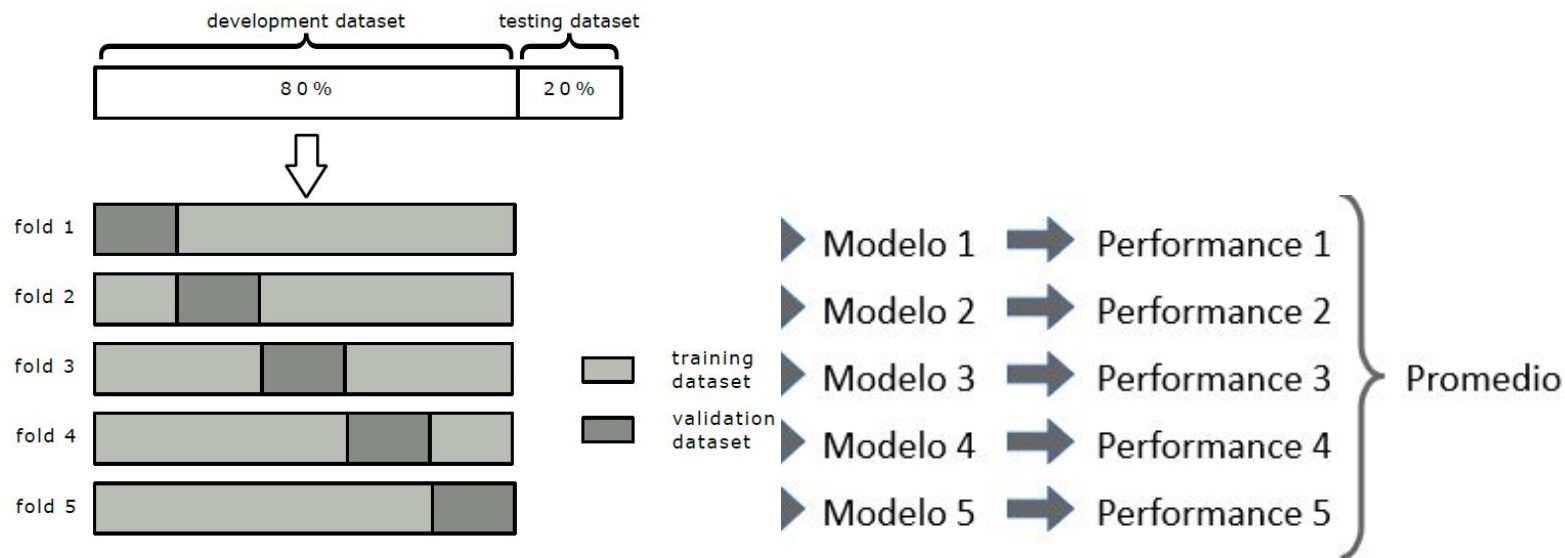
1. Desordenar los datos.
2. Separar en k folds disjuntos del mismo tamaño.
3. Para $i = 1 \dots k$: entrenar sobre todos menos i ; evaluar sobre i .

Ejemplo para $k = 5$.

Valores usuales de D/T: 80-20, 70-30, $\frac{2}{3}$ - $\frac{1}{3}$



Validación cruzada



Selección de modelos

¿Por qué tendríamos distintos modelos para comparar?

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

Ejemplo: hiperparámetros de los árboles de decisión

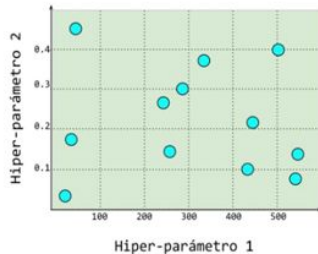
- Criterio de elección de atributos en cada nodo (Information Gain, Gini Gain...)
- Criterio de parada (ej: max_depth)
- Estrategia de poda

Selección de modelos

¿Cómo buscar la mejor combinación de **atributos + algoritmos + hiperparámetros**?

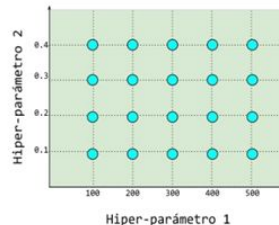
- Exploramos un espacio de búsqueda, **usando k-fold CV** para medir el desempeño de cada combinación.

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**

Medidas de performance

Matriz de confusión (clasificación binaria)

		valores reales	
		positivo	negativo
predicción	positivo	TP	FP
	negativo	FN	TN

TP: true positives

FP: false positives

TN: true negatives

FN: false negatives

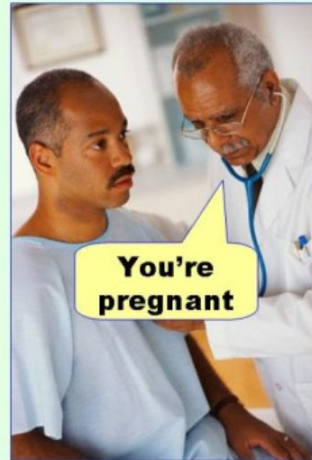
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

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

Medidas de performance

Type I error
(false positive)



Type II error
(false negative)



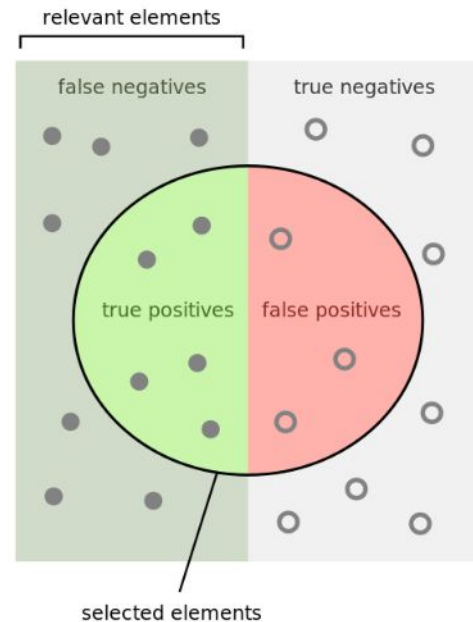
Medidas de performance

Precisión = $\frac{TP}{TP + FP}$ de las instancias clasificadas

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

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

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



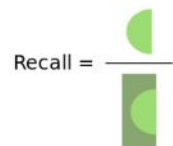
How many selected items are relevant?



Precision =

Wikipedia

How many relevant items are selected?



Recall =

Medidas de performance

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

(cuán útiles son los resultados de búsqueda)

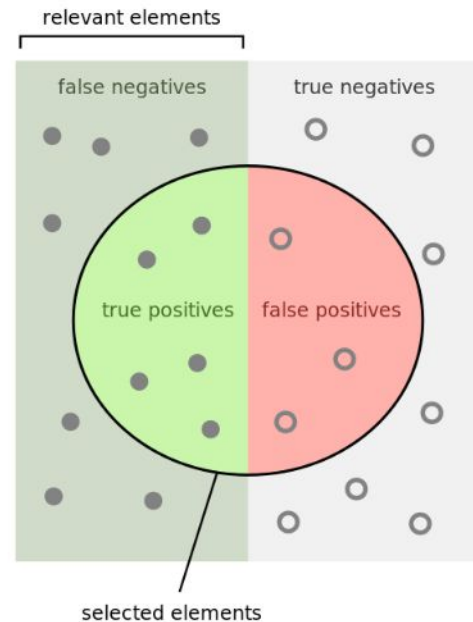
$$\text{Recall} = \frac{TP}{TP + FN}$$

(cuán completos son los resultados)



Se clasifican 4 como gatos

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



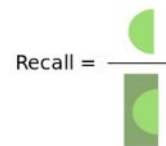
How many selected items are relevant?



Precision =

Wikipedia

How many relevant items are selected?



Recall =

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

Media armónica:

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

También llamada **F₁ score**.

Fórmula general:

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

Medidas de performance

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

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

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

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

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

FPR: Proporción de impostores que aceptamos erróneamente.

Medidas de performance

CURVA ROC (Receiver operating characteristic)

- Gráfico TPR (Recall) vs. FPR

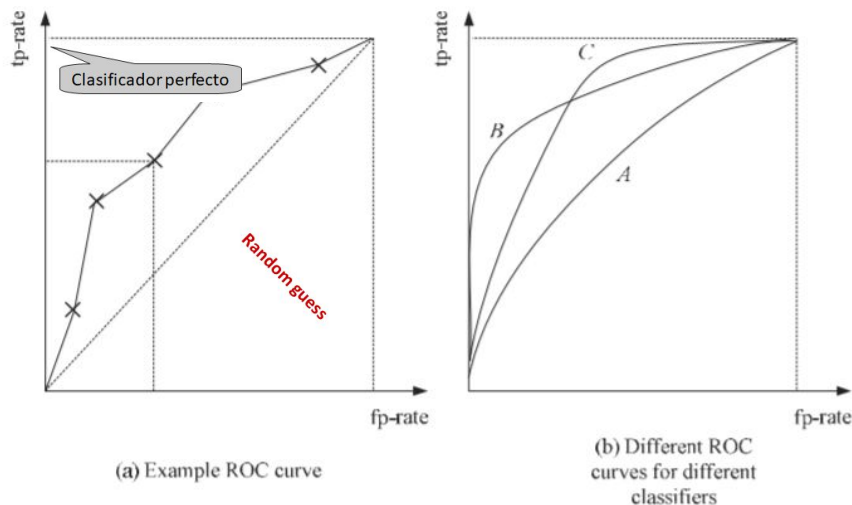
$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN}$$

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

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

	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.

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

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

Factores a considerar en la elección de modelos

- **Tasa de error**
- **Velocidad de entrenamiento y velocidad de test**
- **Interpretabilidad** (¿el conocimiento extraído del modelo puede ser validado por expertos?)
- **Facilidad de desarrollo**

Experimentos de aprendizaje automático

1. **Establecer objetivo de estudio** (error de un algoritmo, comparación de dos algoritmos, etc.)
2. **Seleccionar la métrica para evaluar performance**
3. Seleccionar **factores** (dependen de 1): hiperparámetros de un algoritmo, comparación de algoritmos: algoritmos a comparar)
4. **Elegir diseño experimental** (división de conjunto en entrenamiento y test, cross validation, hiperparámetros: modificaciones aleatorias vs. grid search)
5. **Realización de experimento** (uso de código testeado, reproducibilidad de resultados)
6. Realizar **análisis estadísticos** de los datos
7. **Conclusiones:** son sobre los datos utilizados. Realizar análisis de errores.

Resumen

Armado de conjunto de datos:

- Dejar **apartado un conjunto de test** o held-out
- Seleccionar modelos con **conjunto de desarrollo** (datos de entrenamiento y validación):
 - k-fold cross validation
 - grid search y random search
- Medidas de performance:
 - Accuracy
 - Precision, Recall, F1
 - Sensibilidad, especificidad
 - TPR, FPR, curva ROC, AUC (área bajo la curva)

Bibliografía

Capítulos de libros:

.ISLR, Cap. 2 (2.2)

.Alpaydin, Cap. 19 (hasta 19.7 inclusive)