Inteligencia Artificial para profesionales TIC

Métricas y evaluación en ML

valgrai.eu

Índice

1. Métricas de error de ML

Cuando la ingeniería llega al ML

Encontrar la técnica de ML adecuada para un problema específico puede ser difícil.

Algunas pautas generales pueden ayudarnos en esta decisión.

Antes de analizar las diferentes técnicas de ML, vamos a prestar atención primero a lo importante que es **evaluar el error de sus predicciones**, ya que **ninguna** de ellas va a ser capaz de encontrar la solución óptima.

Nos vamos a centrar en los algoritmos supervisados y estudiar las diferentes métricas de error que podemos obtener en clasificación y regresión.

1. Métricas de error de ML

Cuando la ingeniería llega al ML

Al ser un **aprendizaje supervisado**, tenemos un conjunto de datos con muestras **completas**.

Para cada muestra conocemos el vector de **características** y su **etiqueta** (una clase en clasificación, un valor numérico en regresión)

Por lo tanto, las métricas de error van, de alguna manera, a comparar de las diferentes muestras del conjunto de datos las **diferencias entre los valores reales** (etiquetas) y las **predicciones** (las que obtenemos después de generar el modelo de ML)

Las métricas de error son tan importantes que las cubriremos incluso antes de estudiar cómo las diferentes técnicas de ML crean los modelos de predicción: las métricas que vamos a estudiar son las mismas en todos los algoritmos de ML que vamos a ver.

La exclusion (holdout) y la validación cruzada también deben tenerse en cuenta a la hora de evaluar cualquier técnica de aprendizaje automático.

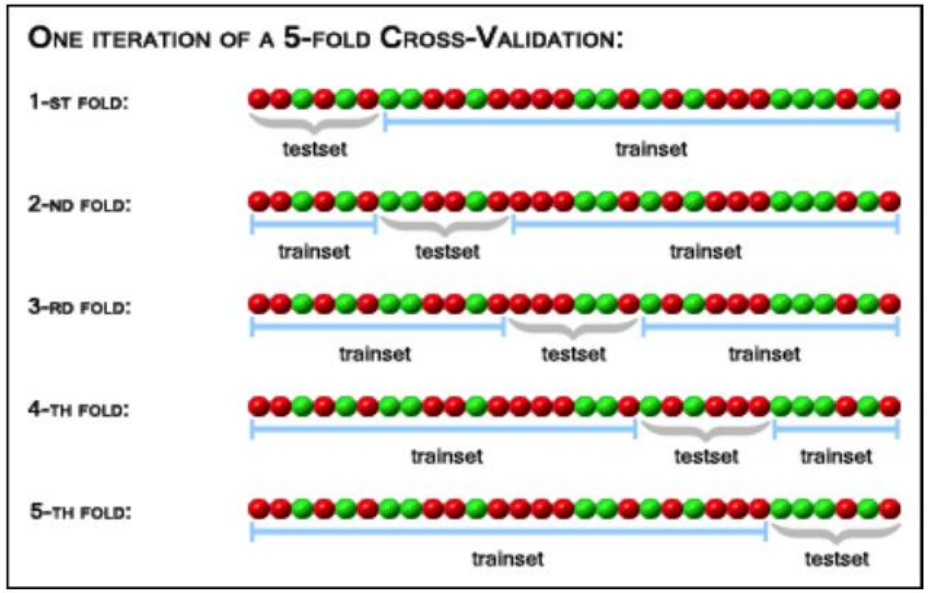
1. Métricas de error de ML

Validación cruzada

Conocemos el holdout, pero ¿**validación cruzada**?cross-validation

Repetimos la técnica de holdout en lo que llamamos validación cruzada.

Esta técnica también se conoce como **K-Fold**, donde K es el número de series.



En el ejemplo anterior hay un **5-fold**, lo que significa **80-20**, donde **mantenemos** la distribución aleatoria original de los elementos del conjunto de datos, por lo que finalmente todos los elementos han estado involucrados en los conjuntos de entrenamiento y prueba (realmente hemos creado 5 modelos diferentes).

Más de 10 folds no proporcionan mejora (LOOCV es el extremo, tal vez con conjuntos de datos muy pequeños)

Promediando la tasa de éxito (en pruebas de validación y entrenamiento) de las divisiones K obtenemos algo **mucho más válido estadísticamente**.

Las bibliotecas ya vienen con esto implementado.

1. Métricas de error de ML

Resultados reales y previstos

A la hora de evaluar la calidad del modelo, dos conceptos son clave, el **resultado real** (normalmente llamado **y**) y el **resultado predicho** (normalmente llamado **ŷ**)

Ambos valores están disponibles ya que estamos utilizando el aprendizaje supervisado.

Un modelo de aprendizaje automático tiene como objetivo asegurarse de que cada vez que se le presenta una muestra, **el resultado predicho intente corresponder con el resultado real**.

Diferentes métricas que miden el error del modelo pueden darnos **información muy crítica** sobre la decisión adecuada de seleccionar una técnica específica de ML.

Cuanto mayor sea la diferencia entre el resultado real **y** y el resultado predicho **ŷ**, más "desviado" estará el modelo de ser una representación precisa del fenómeno.

Ahora vamos a ver diferentes métricas que nos dan diferentes propiedades. Es importante tener en cuenta que la diferencia entre el **f(x) ideal** y el **f(x) estimado** siempre **tiene una parte que no depende de la técnica de ML en sí, sino de una varianza inherente a los datos de entrenamiento** (por ejemplo, porque el vector de características no es lo suficientemente bueno).

2. Regresión

Al predecir valores

El **error cuadrático medio** mide el promedio de los errores al cuadrado. Básicamente, calcula la diferencia entre el valor estimado y el real, eleva al cuadrado estos resultados y luego calcula su promedio. Es una magnitud positiva.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

La **raíz del error cuadrático medio (RMSE)** calcula el promedio de los errores al cuadrado en todas las muestras, pero, además, toma la raíz cuadrada del resultado, tomando efectivamente la raíz cuadrada de MSE. El **Error Absoluto Medio (MAE)** toma el valor absoluto, ya que no estamos interesados en la dirección en la que difieren los valores objetivo estimados y reales (estimados > reales o viceversa), sino en la distancia absoluta

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

2. Regresión

Al predecir valores

R cuadrado (**R^2**) o **coeficiente de determinación** representa la proporción de la varianza de la variable dependiente y que se explica por las variables independientes X.

R^2 explica hasta qué punto la varianza de una variable explica la varianza de la segunda variable.

Por lo tanto, si el R^2 de un modelo es 0,75, entonces aproximadamente el 75% de la variación observada puede explicarse por las características del modelo. Un valor alto significa un buen proceso de ingeniería de características.

R^2 se calcula tomando uno menos la suma de los cuadrados de los residuos dividida por la suma total de los cuadrados.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

3. Clasificación

Al predecir clases o categorías

A la hora de evaluar el rendimiento de un modelo de clasificación, dos conceptos son clave, el resultado real (normalmente llamado **y**) y el resultado predicho (normalmente llamado **ŷ**).

Por ejemplo, se puede entrenar un modelo para predecir si una persona desarrollará una enfermedad en particular.

En este caso, se entrena con muestras, por ejemplo, los datos de una persona, que contienen información predictiva, como la edad, el sexo, etc. y cada persona está etiquetada con una etiqueta que indica si la enfermedad se desarrollará o no.

En este caso, la etiqueta puede ser si la enfermedad ocurrirá (y=1) o no ocurrirá (y=0).

Si nuestro algoritmo de ML nos da **probabilidad**, elegiremos la clase más probable (>0.5 como umbral general).

3. Clasificación

True Positive, True Negative, False Positive and False Negative

Cada predicción del modelo puede ser de uno de los cuatro tipos con respecto al rendimiento:

True Positive, True Negative, False Positive o False Negative.

- **True Positive (TP):** Se predice que una muestra es positiva (ŷ=1, por ejemplo, se predice que la persona desarrollará la enfermedad) y su etiqueta es realmente positiva (y=1, por ejemplo, la persona realmente desarrollará la enfermedad).
- **True Negative (TN):** Se predice que una muestra es negativa (ŷ=0, por ejemplo, se predice que la persona no desarrollará la enfermedad) y su etiqueta es en realidad negativa (y=0, por ejemplo, la persona en realidad no desarrollará la enfermedad).
- **False Positive (FP):** Se predice que una muestra es positiva (ŷ=1, por ejemplo, se predice que la persona desarrollará la enfermedad) y su etiqueta es en realidad negativa (y=0, por ejemplo, la persona en realidad no desarrollará la enfermedad). En este caso, la muestra se predice "falsamente" como positiva.
- **False Negative (FN):** Se predice que una muestra es negativa (ŷ=0, por ejemplo, se predice que la persona no desarrollará la enfermedad) y su etiqueta es realmente positiva (y=1, por ejemplo, la persona realmente desarrollará la enfermedad). En este caso, la muestra se predice "falsamente" como negativa.

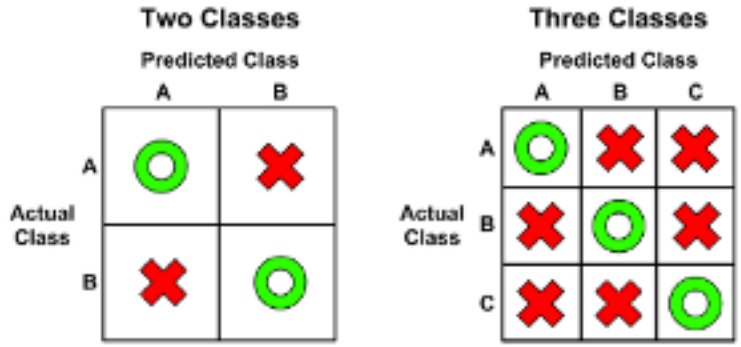
3. Clasificación

Matriz de confusión

Una distribución tabular de TP, TN, FP, FN Los predichos correctos están en la diagonal (lo que queremos maximizar en nuestras técnicas de ML).

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = Σ True Positive / Σ Test Outcome Positive
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = Σ True Negative / Σ Test Outcome Negative
		Sensitivity = Σ True Positive / Σ Condition Positive	**Specificity** = Σ True Negative / Σ Condition Negative	

Precision



3. Clasificación

Exactitud

Accuracy es la fracción de predicciones que nuestro modelo obtuvo de todas las predicciones.

Accuracy oscila entre 0 y 1, estos casos extremos corresponden a fallar por completo las predicciones o tener siempre predicciones correctas.

La accuracy, sin embargo, **no es una gran métrica**, especialmente cuando los datos están **desequilibrados**. Cuando hay una disparidad significativa entre el número de etiquetas positivas y negativas, la accuracy no cuenta la historia completa.

Por ejemplo, imaginemos que tenemos 100 muestras en el conjunto de datos y 95 etiquetadas como clase 0 y 5 como clase 1; ¿Cuál es la precisión si tenemos un modelo de aprendizaje de ML estúpido que solo devuelve 0?

La precision, la recall/sensitivity y la especificidad pueden ayudarnos a resolver los problemas anteriores con accuracy

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

3. Clasificación

Precision, y sensitivity/recall

Precision indica qué proporción de predicciones positivas fue realmente correcta.

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

Sensitivity o Recall con el objetivo de medir qué proporción de positivos reales se identificó correctamente.

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

$$\text{Sensitivity}$$

3. Clasificación

Specificity y F1

Specificity (la simétrica de Sensitivity) tiene como objetivo medir qué proporción de negativos reales se identificó correctamente.

$$\text{Specificity} = \frac{TN}{(FP + TN)}$$

Otra métrica que en este caso resume las anteriores, y que también es bueno para obtener valores cercanos a 1, **is F-measure or F1**.

$$\text{F} - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{recall} + \text{precision}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

Valencian Graduate School and Research Network of Artificial Intelligence

valgrai.eu

valgrai.eu

Valencian Graduate School and Research Network of Artificial Intelligence