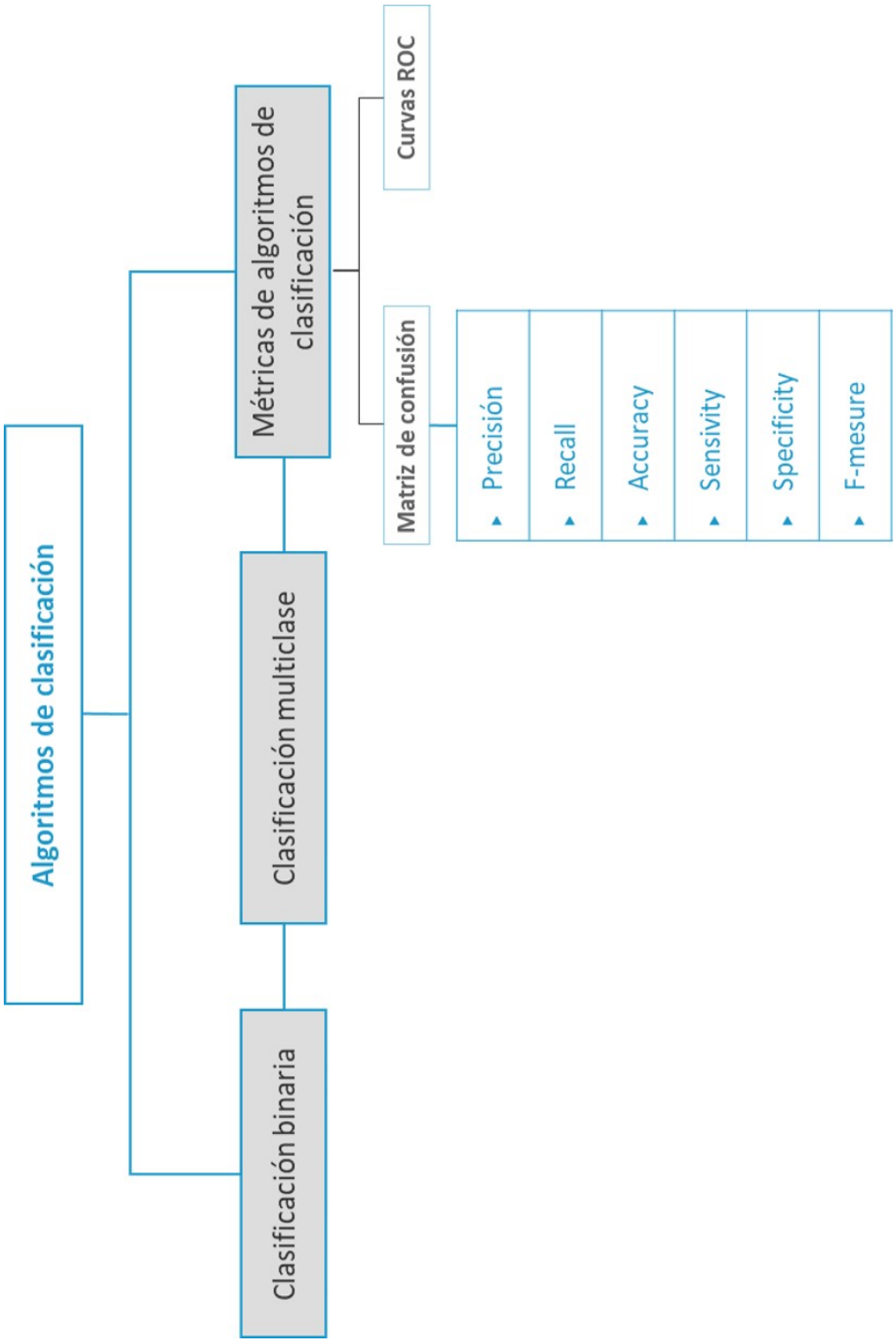
Técnicas de Aprendizaje Automático

Tema 5. Evaluación de algoritmos de clasificación

Índice

Esquema



5.1. Introducción y objetivos

En este tema se pretende proporcionar una comprensión de los modelos de clasificación y evaluar su desempeño.

Los problemas de clasificación en *machine learning* son un tipo de tarea en la que el objetivo es **asignar una etiqueta o categoría a una instancia dada**, basándose en ciertas características o atributos. La idea es aprender un modelo que pueda generalizar a partir de los datos de entrenamiento para predecir la clase correcta para nuevas instancias no vistas. Por lo tanto, en el caso de los problemas de clasificación, se busca aprender una función que pueda mapear correctamente las características de entrada a una de las etiquetas o categorías predefinidas. Dentro de la tarea de clasificación, esta puede ser de dos tipos en base a la naturaleza de las etiquetas o categorías:

- **Clasificación binaria:** el objetivo es predecir una clase entre dos posibles, sí/no, verdadero/falso, positivo/negativo, etc.
- **Clasificación multiclase:** el objetivo es predecir una clase de entre n posibles. Por ejemplo, predecir el tipo de animal (perro, gato, pájaro, etc.) basado en características específicas.

Existe otro tipo de clasificación, que queda fuera de los objetivos de este tema, que es la clasificación multietiqueta, donde una instancia puede pertenecer a múltiples clases simultáneamente. Por ejemplo, en la clasificación de imágenes, una imagen puede tener múltiples etiquetas como "playa", "persona" y "perro" al mismo tiempo.

Para abordar estos problemas de clasificación, se utilizan una variedad de algoritmos de *machine learning*, como árboles de decisión, máquinas de vectores de soporte (SVM), k vecinos más cercanos (KNN), entre otros. Estos algoritmos se presentarán en los siguientes temas.

Ideas clave

Además, se emplean métricas de evaluación específicas para medir el rendimiento de los modelos de clasificación, como la precisión, el *recall*, la F1-score, entre otros. Estas métricas serán el núcleo central de estudio del presente tema.

El objetivo de este tema es obtener una **mayor comprensión de los modelos de clasificación**, entender las métricas existentes para evaluar el rendimiento de estos modelos y ser capaz de explicar el rendimiento de estos. A continuación, se detallan los objetivos específicos:

- Comprensión de los modelos de clasificación y los diferentes tipos de clasificación existentes.
- Comprensión de las métricas de evaluación del rendimiento de los modelos de clasificación.
- Identificar los problemas de rendimiento asociados al desbalanceo de los datos.
- Validar el aprendizaje con la visualización de las métricas de rendimiento.

5.2. Algoritmos de clasificación

Dentro del aprendizaje supervisado, a continuación de los algoritmos de regresión, el siguiente gran grupo de algoritmos son los de **clasificación**. A diferencia de los algoritmos de regresión cuyo objetivo es predecir un valor numérico, los algoritmos de clasificación tienen como objetivo obtener la clase más probable para cada una de las instancias de datos.

Este tipo de técnicas pueden utilizarse para predecir o estimar la probabilidad de pertenencia a una clase de entre dos posibles, lo que se conoce como **clasificación binaria**. O bien se pueden utilizar para predecir o estimar la probabilidad de pertenencia de una clase de entre varias posibles (más de dos), lo cual se conoce como **clasificación multiclase**.

Los problemas de clasificación son muy habituales en el área del *machine learning*, ocurren con frecuencia, quizás incluso más que los problemas de regresión. Algunos de los ejemplos más típicos de tareas de clasificación serían:

- **Clasificación de spam de correo electrónico:** en este problema, el objetivo es clasificar correos electrónicos como spam o no spam. Se pueden utilizar características como la frecuencia de palabras clave, la presencia de ciertas palabras o frases, la estructura del correo electrónico, entre otros, para entrenar un modelo que pueda predecir si un nuevo correo electrónico es spam o no.

- **Diagnóstico médico:** en este caso, se pueden utilizar datos médicos para predecir la presencia o ausencia de una enfermedad. Por ejemplo, utilizando resultados de pruebas médicas, síntomas del paciente, historial médico y otras características relevantes, un modelo de clasificación puede predecir si un paciente tiene una enfermedad específica, como diabetes, cáncer, enfermedades cardíacas, etc.

- **Clasificación de documentos:** en este problema, se busca categorizar documentos en diferentes clases o categorías. Por ejemplo, se podrían clasificar artículos de noticias en categorías como deportes, política, entretenimiento, etc. Esto puede ser útil para organizar grandes volúmenes de información, facilitando la búsqueda y la navegación. Se pueden utilizar características del texto, como palabras clave, frecuencia de términos, análisis de sentimientos, entre otros, para entrenar un modelo de clasificación que pueda asignar automáticamente documentos a las categorías adecuadas.

Al igual que en los problemas de regresión, en las tareas de clasificación tenemos un conjunto de observaciones de entrenamiento $((x_1, y_1)...(x_n, y_n))$ que podemos usar para construir un clasificador. Queremos que nuestro clasificador funcione bien no solo en los datos de entrenamiento, sino también en los datos de test que no se utilizaron para entrenar el clasificador.

5.3. Predicciones de clase

Los modelos de clasificación suelen generar dos tipos de predicciones. Al igual que los modelos de regresión, los modelos de clasificación producen un valor de predicción en forma numérica continua, que generalmente tiene la forma de probabilidad de pertenencia; es decir, los valores predichos de pertenencia a una clase para cualquier muestra individual están entre 0 y 1, y suman 1. Además de una predicción en forma de valor continuo, los modelos de clasificación generan una clase predicha, que se presenta en forma de categoría discreta.

Para la mayoría de las aplicaciones prácticas, se requiere una predicción de categoría discreta para poder tomar una decisión. El filtrado automatizado de spam, por ejemplo, requiere un valor definitivo para cada correo electrónico. Aunque los modelos de clasificación producen ambos tipos de predicciones, a menudo la atención se centra en la predicción discreta en lugar de la predicción continua. Sin embargo, las estimaciones de probabilidad para cada clase pueden resultar muy útiles para medir la confianza del modelo sobre la clasificación predicha. Volviendo al ejemplo del filtro de correo electrónico no deseado, un mensaje de correo electrónico con una probabilidad predicha de ser spam de 0,51 se clasificaría de la misma manera que un mensaje con una probabilidad predicha de ser spam de 0,99. Si bien el filtro trataría ambos mensajes de la misma manera, tendríamos más confianza en que el segundo mensaje fuera, de hecho, verdaderamente spam.

En este tema, y a lo largo de la asignatura, se asumirá que la salida de los modelos de clasificación es un valor de una categoría discreta. Sin embargo, se recomienda la lectura de Kuhn y Johnson (2013) y Gupta *et al.* (2006) para ampliar información sobre cómo los clasificadores producen valores de probabilidad de pertenencia.

5.4. Métricas de evaluación: matriz de confusión

La mejor métrica de rendimiento de un algoritmo de clasificación es ver si el clasificador tiene éxito para su propósito. Para evaluar un clasificador se pueden utilizar los valores predichos de las clases, los valores reales de las clases o bien la probabilidad estimada de la predicción.

Un método habitual para describir el rendimiento de un modelo de clasificación es la **matriz de confusión**. Se trata de una matriz simple de las clases observadas y predichas para los datos. La Tabla 1 muestra un ejemplo cuando el dataset tiene dos clases. Las celdas diagonales indican casos en los que las clases se predicen correctamente, mientras que las celdas fuera de la diagonal ilustran el número de errores para cada caso posible.

	Reales	
Predichos	**Positiva**	**Negativa**
Positiva	TP	FP
Negativa	FN	TN

Tabla 1. La matriz de confusión para dos posibles clases (“positiva” y “negativa”). Fuente: elaboración propia.

Las celdas de la tabla indican el número de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

En esta sección, primero analizaremos las métricas para el caso especial de la clasificación binaria y luego pasaremos a la clasificación multiclase.

Métricas para clasificación binaria

La clasificación binaria es posiblemente la aplicación más común y conceptualmente simple del aprendizaje automático. Sin embargo, todavía hay una serie de advertencias a la hora de evaluar incluso esta sencilla tarea. Antes de sumergirnos en el conjunto de métricas, echemos un vistazo a las formas en que medir la precisión puede ser engañoso. Es importante recordar que, para la clasificación binaria, muchas veces hablamos de una clase positiva y una clase negativa, entendiendo que la clase positiva es la que buscamos.

Si nos fijamos en la matriz de confusión, la primera métrica derivada de ese reparto propuesto entre aciertos y fallos es la **tasa de éxito** (*accuracy rate*), o si lo expresamos en negativo, la **tasa de error**. Esta tasa refleja la concordancia entre lo observado y lo predicho y tiene la interpretación más sencilla posible. Sin embargo, existen algunas desventajas al utilizar esta métrica. En primer lugar, los cálculos generales de precisión no hacen distinción sobre el tipo de errores que se cometen. En el filtrado de spam, el coste de eliminar por error un correo electrónico importante probablemente sea mayor que permitir que un correo electrónico spam pase incorrectamente un filtro. En este tipo de situaciones, donde los costes de los errores son diferentes, es posible que la tasa de éxito del clasificador no mida las características importantes del modelo (Provost *et al.*, 1998).

Incluso en el caso de los clasificadores binarios, que realizan la predicción de una clase en función de dos posibles categorías, es muy útil conocer con qué certeza o confianza se predice cada una de las clases.

Volviendo al ejemplo del filtro de spam, un mensaje de correo electrónico puede ser predicho como spam con una probabilidad de 0,9 o de 0,59, etc. De forma general, si dos algoritmos de clasificación diferentes producen los mismos errores, pero uno de ellos es más capaz de tener en cuenta la incertidumbre, sería un mejor modelo. Por tanto, para evaluar correctamente el resultado de un clasificador es necesario

considerar el valor de probabilidad obtenido en lugar de utilizar únicamente la clase más probable. El objetivo es obtener modelos que tienen mucha confianza en las predicciones correctas y sean cautelosos en las predicciones dudosas. Este **balance entre confianza y prudencia** es la clave de la evaluación de modelos.

Por lo tanto, la tasa de éxito no es una buena medida del rendimiento predictivo, ya que la cantidad de errores que comete el clasificador no contiene toda la información que nos interesa.

Imaginemos una aplicación para la detección precoz del cáncer mediante un test automatizado. Si la prueba es negativa, se asumirá que el paciente está sano, mientras que, si la prueba es positiva, el paciente se tendrá que someterse a exámenes adicionales. En este caso, llamaríamos a una prueba positiva (un indicio de cáncer) la clase positiva, y a una prueba negativa, la clase negativa. No podemos asumir que nuestro modelo siempre funcionará perfectamente, ya que cometerá errores. Para cualquier aplicación, debemos preguntarnos cuáles podrían ser las consecuencias de estos errores en el mundo real.

Para poder evaluar correctamente el rendimiento de un clasificador y evaluar la cantidad y calidad de los errores hay que **volver a la matriz de confusión** y ver todas las posibles métricas derivadas de ella.

Como ya se ha dicho previamente, una matriz de confusión es una tabla que organiza las predicciones en función de los valores reales de los datos. Una de las dimensiones de la tabla hace referencia a la categoría de los valores predichos, la otra dimensión a las categorías reales.

Las instancias clasificadas correctamente caen en la diagonal de la matriz. Los valores fuera de la diagonal indican las instancias clasificadas incorrectamente, se trata de predicciones incorrectas. Las métricas de rendimiento obtenidas con la matriz de confusión se basan en cuantas instancias caen dentro y fuera de la diagonal. La mayoría de las métricas de rendimiento consideran la capacidad que

tiene un clasificador de discernir una categoría respecto de las demás. La categoría de interés se conoce como **clase positiva**, mientas que las otras categorías se conocen como **clase negativa.**

En la Tabla 2 se muestra cómo se obtienen los cuatro valores de una matriz de confusión para el caso de una clasificación binaria, junto con las métricas derivadas de la combinación de cada uno de estos valores. En las filas aparecen los resultados de clases del clasificador, como columnas, las clases reales, y en la intersección (celda) el número de instancias de cada tipo.

		Clases reales		
		Clase positiva	Clase negativa	
Clases predichas	Clase positiva	True Positive (TP)	False Positive (FP)	Positive Predicted Value/Precision $\frac{TP}{TP+FP}$
	Clase negativa	False Negative (FN)	True Negative (TN)	Negative Predicted Value/Precision $\frac{TN}{TN+FN}$
		Sensibilidad $\frac{\sum True\ Positive}{\sum Reales\ positivos}$	Especificidad $\frac{\sum True\ Negative}{\sum Reales\ negativos}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Tabla 2. Matriz de confusión y sus métricas derivadas. Fuente: elaboración propia

Cada uno de estos cuatro valores se pueden **definir** de la siguiente forma:

- **Tasa de verdaderos positivos** (TP, true positives): se trata de las clasificaciones correctas de las instancias que corresponden a la clase positiva.
- **Tasa de falsos positivos** (FP, false positives): se trata de las clasificaciones de la clase negativa que han sido incorrectamente clasificadas como clase positiva.

- **Tasa de verdaderos negativos** (TN, true negatives): se trata de las clasificaciones correctas de las instancias que corresponden a la clase negativa.

- **Tasa de falsos negativos** (FN, false negatives): se trata de las clasificaciones de la clase positiva que han sido incorrectamente clasificadas como clase negativa.

Por medio de la matriz de confusión se pueden obtener también las siguientes métricas:

- **Accuracy:** esta métrica también se conoce como la **ratio de éxito**. Representa la proporción del número de predicciones correctas entre el número total de predicciones y se define así:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- **Precision/recall:** estas métricas hacen referencia a los compromisos hechos en la clasificación. Se utilizan mucho en el campo de *information retrieval* e indican lo interesantes y relevantes que son los resultados de un modelo. La métrica de precisión también se conoce como **positive predictive value (PPV)** e indica la proporción de ejemplos que son verdaderamente positivos. Es decir, cuando el modelo predice la clase positiva, ¿cuántas veces está en lo cierto? Por otro lado, la métrica de *recall* indica qué tan completos son los resultados. La métrica de *recall* equivale a la **métrica de sensibilidad.** Un modelo con gran *recall* captura un gran porcentaje de ejemplos positivos. La definición de ambas métricas es la siguiente:

$$precision = \frac{TP}{TP+FP}$$

$$recall = \frac{TP}{TP+FN}$$

- **F-measure:** también conocida como F1, es una métrica que combina precisión y *recall* utilizando la media armónica. Se utiliza la media armónica porque los valores

indican proporciones entre 0 y 1. Se utiliza con mucha frecuencia, puesto que simplifica el rendimiento de un algoritmo de clasificación a una única métrica. La métrica por defecto asume que precisión y *recall* tienen la misma importancia, pero es posible ponderarlos para darle mayor peso a uno u otro. De forma matemática se define así:

$$F1 = \frac{2 * precision * recall}{recall + precision} = \frac{2 * TP}{2 * TP + FP + FN}$$

- **Sensivity/Specificity (sensibilidad/especificidad):** una clasificación siempre conlleva un balance entre ser conservador y agresivo. Por ejemplo, un sistema de clasificación de spam podría eliminar todos los mensajes de spam a costa de eliminar también los mensajes ham. Por otro lado, es necesario una garantía de que ningún mensaje ham sea clasificado como spam. Este balance se captura con las métricas de sensibilidad y especificidad. La sensibilidad de un modelo es la ratio de los ejemplos positivos correctamente clasificados. La especificidad de un modelo indica la proporción de los ejemplos negativos correctamente clasificados. Estas métricas tienen valores de 0 a 1, siendo 1 lo más deseable, y se definen así:

$$especificidad = \frac{TN}{TN + FP}$$

$$sensibilidad = \frac{TP}{TP + FN}$$

A la hora de evaluar los algoritmos de clasificación es importante diferenciar entre una clasificación binaria o multiclase. Una de las métricas más comunes es la matriz de confusión que nos proporciona información acerca de los aciertos y errores de cada una de las clases. A partir de la matriz de confusión se pueden obtener métricas como *accuracy*, *sensivity/specificity*, *precisión/recall* y *f-measure*.

Métricas para clasificación multiclase

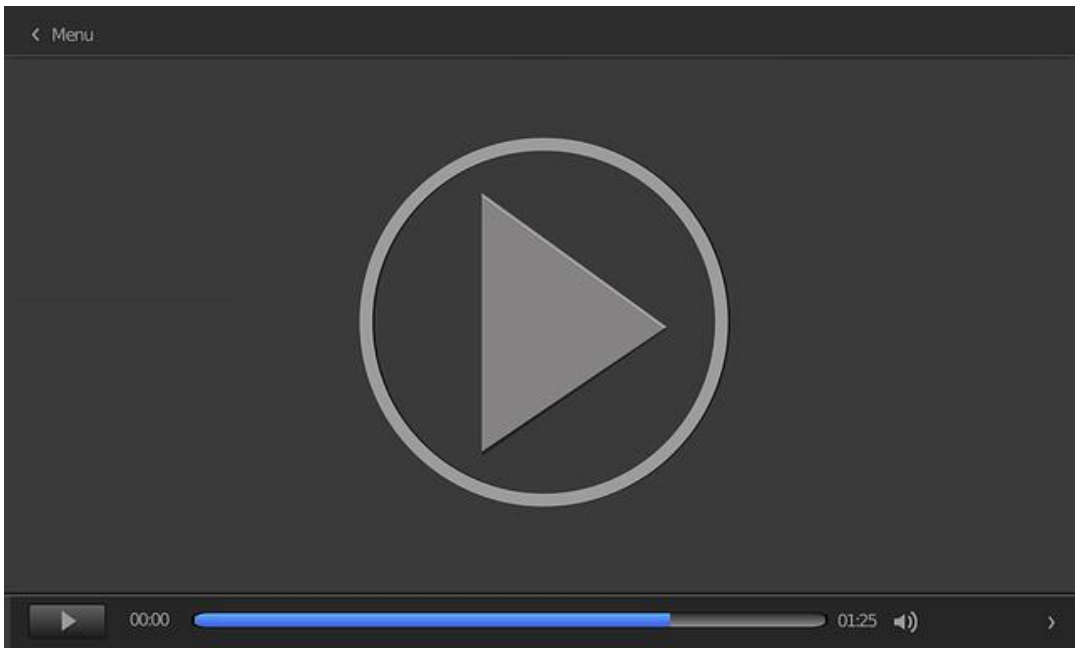
Ahora que hemos analizado en profundidad la evaluación de las métricas de clasificación binaria, pasemos a las métricas para evaluar la clasificación multiclase.

Ideas clave

Básicamente, todas las métricas para la clasificación multiclase se derivan de métricas de clasificación binaria, pero se promedian para todas las clases.

La tasa de éxito para la clasificación multiclase se define nuevamente como la fracción de ejemplos correctamente clasificados. En general, los resultados de la clasificación multiclase son más difíciles de entender que los resultados de la clasificación binaria. Además de la precisión, las herramientas comunes vuelven a ser las métricas derivadas de la matriz de confusión y el informe de clasificación que vimos en el caso binario de la sección anterior, pero esta vez para cada una de las N posibles clases.

Para ilustrar un ejemplo de evaluación de un clasificador multiclase dispones del siguiente vídeo.



Accede al vídeo:

https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=ebbffcb4-886e-422c-b34c-b17e00eb18b2

Dataset desbalanceados

Un dataset desbalanceado en *machine learning* es aquel en el que la distribución de las clases objetivo (etiquetas) no es uniforme; es decir, hay una gran diferencia en el número de instancias entre las diferentes clases. Esto significa que una o más clases están subrepresentadas en comparación con otras.

> Supongamos un problema de clasificación binaria donde una clase representa la presencia de una enfermedad (clase positiva) y la otra clase representa la ausencia de la enfermedad (clase negativa). Si el 95 % de las instancias pertenecen a la clase negativa y solo el 5 % pertenecen a la clase positiva, este sería un dataset desbalanceado.

El desbalanceo en los datos puede causar varios problemas en el proceso de entrenamiento y evaluación de modelos de *machine learning*, entre ellos, el que nos ocupa en este tema, tener métricas de evaluación sesgadas. Cuando se presenta un dataset desbalanceado las métricas globales, como la tasa de éxito del clasificador no pueden ser representativas del rendimiento real del modelo. Esto es debido a que los modelos de *machine learning* tienden a sesgarse hacia la clase mayoritaria, lo que puede llevar a un rendimiento deficiente en la clasificación de la clase minoritaria, sin embargo, esto no se vería reflejado en la ratio de éxito global.

Los tipos de errores que comete el clasificador juegan un papel importante cuando una de dos clases es mucho más frecuente que la otra. Esto es muy común en la industria; un buen ejemplo es la predicción de clics, donde cada punto de datos representa una «impresión», un elemento que se mostró a un usuario. Este elemento puede ser un anuncio, una historia o una persona relacionadas a seguir en una red social. El objetivo es predecir, si se le muestra un elemento en particular, si un usuario hará clic en él (lo que indica que está interesado).

Ideas clave

La mayoría de las cosas que se muestran a los usuarios en Internet (en particular, los anuncios) no generarán un clic. Es posible que convenga mostrarle a un usuario 100 anuncios o artículos antes de que encuentre algo lo suficientemente interesante como para hacer clic. Esto da como resultado un conjunto de datos donde por cada 99 puntos de datos «sin clic», hay 1 punto de datos «en el que se hace clic»; es decir, el 99 % de las muestras pertenecen a la clase «sin clic». Ahora supongamos que se genera un clasificador que tiene una ratio de éxito del 99 % en la tarea de predicción de clics. ¿Qué está indicando esto en términos de ratio de éxito? Una ratio de éxito del 99 % suena impresionante, pero esto no tiene en cuenta el desbalanceo de las clases. Se puede lograr una ratio de éxito del 99 % sin crear un modelo de aprendizaje automático, prediciendo siempre «sin clic». Por otro lado, incluso con datos desbalanceados, un modelo con una precisión del 99 % podría ser bastante bueno. Sin embargo, la ratio de éxito no nos permite distinguir el modelo constante «sin clic» de un modelo potencialmente bueno.

En realidad, los datos desbalanceados son la norma y es raro que los eventos de interés tengan una frecuencia igual o incluso similar en los datos.

En los casos en los que haya que evaluar el rendimiento de un clasificador con un dataset desbalanceado es importante hacer uso de las métricas individuales (para cada clase) que se han presentado a lo largo de esta sección. Las métricas de evaluación estándar como la ratio de éxito (*accuracy*) pueden no ser apropiadas para evaluar el rendimiento del modelo debido al sesgo hacia la clase mayoritaria. En su lugar, es recomendable utilizar métricas que tengan en cuenta el desbalanceo entre las clases y proporcionen una evaluación más precisa del rendimiento del modelo.

5.5. Métricas de evaluación: curvas ROC y AUC

La clave de cualquier modelo de aprendizaje automático es su **capacidad de generalizar situaciones del futuro** en función de los datos históricos observados. Las métricas vistas en el apartado anterior, que se obtienen a partir de la matriz de confusión conllevan que se establezca un punto de corte determinado sobre la distribución de probabilidad para determinar si una observación es clasificada como una clase determinada. Es decir, por ejemplo, a aquellos valores por encima de 0,5 se les asigna la clase positiva (1) y a aquellos valores iguales o por debajo de 0,5, la clase negativa (0). Con el fin de tener una mayor visibilidad de las decisiones que toma el clasificador en esos límites, se puede utilizar la curva ROC (Receiver Operating Characteristics) (Altman y Bland 1994; Brown y Davis 2006; Fawcett 2006).

La curva ROC fue diseñada como un método general que, dada una colección de puntos de datos continuos, determine un umbral efectivo de modo que los valores por encima del umbral sean indicativos de un evento específico.

En el caso de evaluación de los modelos de clasificación, la curva ROC se utiliza para realizar una evaluación cuantitativa del modelo. La curva ROC para modelos de clasificación binaria representa la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) para diferentes umbrales de decisión. La TPR se calcula como el número de verdaderos positivos dividido por el número total de verdaderos positivos y falsos negativos, es decir, es el *recall*, mientras que la FPR se calcula como el número de falsos positivos dividido por el número total de verdaderos negativos y falsos positivos.

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

La curva ROC es útil porque muestra cómo el modelo de clasificación está realizando la distinción entre las clases positiva y negativa en una variedad de umbrales de decisión, sin tener en cuenta un umbral específico. Cuanto más alejada esté la curva ROC del punto de referencia diagonal (que indica una clasificación aleatoria), mejor será el rendimiento del modelo.

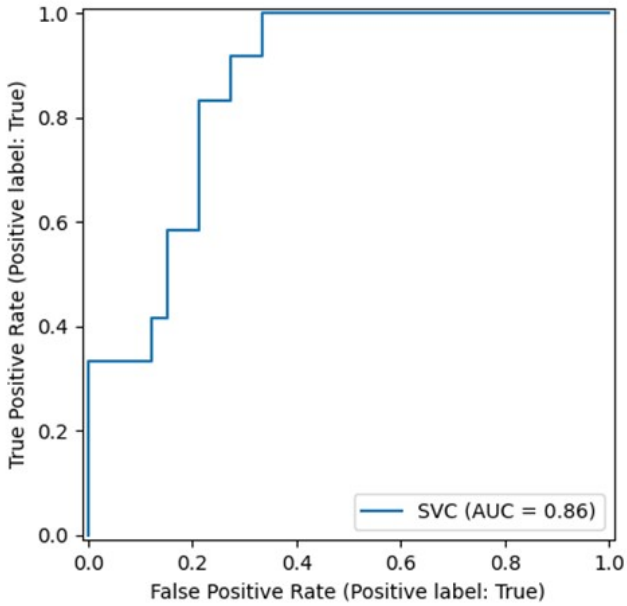


Figura 1. Gráfico de una curva ROC. Fuente: scikit-learn, s.f.

En la Figura 1 se muestra un ejemplo de curva ROC, donde para diferentes umbrales de decisión se muestra el comportamiento del clasificador. En el ejemplo de la Figura 1 se puede observar que la curva ROC se acerca bastante a la esquina superior izquierda del diagrama. Cuanto más se acerque la curva a la esquina superior izquierda del gráfico, mejor clasifica el modelo los datos en categorías.

Un modelo perfecto que separe completamente las dos clases tendría una TPR y FPR de 100 %. Gráficamente, la curva ROC sería un solo paso entre (0, 0) y (0, 1) y permanecería constante desde (0, 1) hasta (1, 1). El área bajo la curva ROC para tal modelo sería 1. Un modelo completamente ineficaz daría como resultado una curva

ROC que sigue de cerca la línea diagonal de 45° y tendría un área bajo la curva ROC de aproximadamente 0,50. Para comparar visualmente diferentes modelos, sus curvas ROC se pueden superponer en el mismo gráfico, un ejemplo de ello se muestra en la Figura 2, donde se han superpuesto las curvas ROC de dos clasificadores diferentes. Comparar curvas ROC puede ser útil para contrastar dos o más modelos con diferentes conjuntos de predictores (para el mismo modelo), diferentes parámetros de ajuste (es decir, dentro de comparaciones de modelos) o clasificadores diferentes completos (es decir, entre modelos).

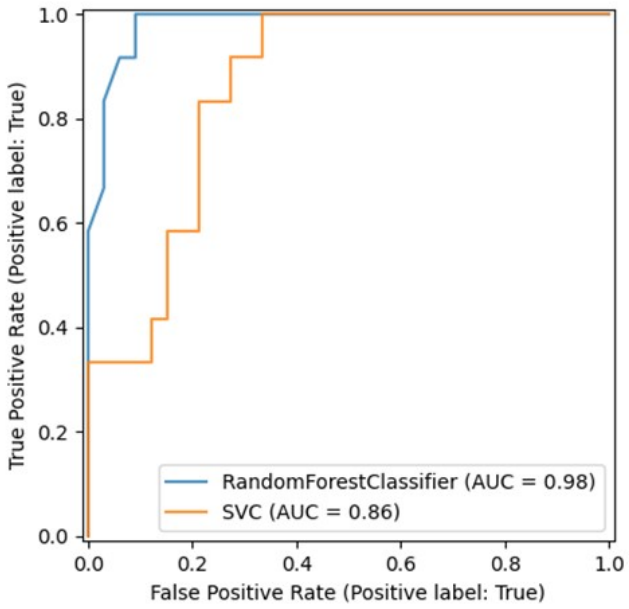


Figura 2. Gráfico de una comparativa entre dos curvas ROC de diferentes modelos de clasificador. Fuente: scikit-learn, s.f.

A partir de estas curvas se puede obtener el **área bajo la curva (AUC)**. Esta métrica va desde valores de 0,5 para un clasificador sin potencia predictiva (completamente aleatorio) hasta 1 para un clasificador perfecto. Cuanto más cercanos están los resultados del clasificador perfecto, mejor. Es posible que para un mismo valor de AUC haya diferentes curvas ROC, por lo que suele ser buena práctica mostrarlas de

forma gráfica.

En general, la curva ROC solo está definida para problemas de dos clases, pero algunos autores han ampliado su definición para manejar tres o más clases. Hand y Till (2001), Lachiche y Flach (2003) y Li y Fine (2008) utilizan diferentes enfoques ampliando la definición de la curva ROC con más de dos clases.

5.6. Cuaderno de ejercicios

- **Ejercicio 1.** Supongamos que hemos utilizado un clasificador, por ejemplo, Naive Bayes, para clasificar documentos con respecto al sentimiento. Las clases son Pos (positivo), Neg (negativo) y Neu (neutro). Probamos nuestro clasificador en 10 documentos para los que conocemos su gold standar (clase real). La prueba tiene los siguientes resultados:

Documento	Clase real	Clase predicha
d1	Pos	Pos
d2	Pos	Pos
d3	Pos	Pos
d4	Pos	Neu
d5	Neg	Neg
d6	Neg	Neu
d7	Neg	Neg
d8	Neu	Pos
d9	Neu	Neu
d10	Neu	Neu

Calcular: precision, recall, accuracy y F1 para estos resultados de clasificación para las tres clases.

SOLUCIÓN

Precicion (Pos, Neg, Neu): array([0.75, 1. , 0.5]),

Recall (Pos, Neg, Neu): array([0.75 , 0.66666667, 0.66666667]),

F1 (Pos, Neg, Neu): array([0.75 , 0.8 , 0.57142857]),

- **Ejercicio 2.** Se evaluó un clasificador binario utilizando un conjunto de 1000 ejemplos de prueba (test) en los que el 50 % de todos los ejemplos son negativos. El clasificador tiene 60 % de sensitivity y 70 % de accuracy. Escribe la matriz de confusión.

	actual 1	actual 0
predicted 1	TP	FP
predicted 0	FN	TN

SOLUCIÓN

	actual 1	actual 0
predicted 1	300	100
predicted 0	200	400

- **Ejercicio 3.** Utilizando la matriz de confusión creada en el ejercicio anterior, calcula la precisión del clasificador, la medida F1 y especificidad.

precision = 0.75

F1 = 0.6666666666666665

Especificidad = 0.8

- **Ejercicio 4.** En Python usaremos el módulo metrics para calcular los valores de la matriz de confusión y las métricas derivadas. En particular, la función https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html nos devuelve la matriz de confusión de un clasificador a partir de los `y_real` , `y_pred` . Dado el siguiente código, calcula la matriz de confusión.

```
from sklearn.metrics import confusion_matrix
```

y_real = ['gato', 'perro', 'gato', 'gato','perro', 'gato']

y_pred = ['perro', 'perro', 'gato', 'gato','perro', 'gato']

Solución: confusion_matrix(y_real, y_pred)

- **Ejercicio 5.** En el código del ejercicio anterior crea las variables `fp`, `fn`, `tp` y `tn`, y calcula sus respectivos valores.

SOLUCIÓN

```
tn, fp, fn, tp = confusion_matrix(y_real, y_pred).ravel()

(tn, fp, fn, tp)

(3, 1, 0, 2)
```

- **Ejercicio 6.** Para el código del ejercicio anterior calcula los valores de precisión, recall, accuracy y f1 para la clase 'gato'.

SOLUCIÓN

```
from sklearn import metrics

precision = metrics.precision_score(y_real, y_pred, pos_label='gato')

recall = metrics.recall_score(y_real, y_pred, pos_label='gato')

accuracy = metrics.accuracy_score(y_real, y_pred)

f1 = metrics.f1_score(y_real, y_pred, pos_label='gato')

(accuracy, precision, recall, f1)

(0.8333333333333334, 1.0, 0.75, 0.8571428571428571)
```

5.7. Referencias bibliográficas

Altman, D. y Bland, J. (1994). Diagnostic Tests 3: Receiver Operating Characteristic Plots. *British Medical Journal, 309*(6948), 188.

Brown, C., Davis, H. (2006). Receiver Operating Characteristics Curves and Related Decision Measures: A Tutorial. *Chemometrics and Intelligent Laboratory Systems, 80*(1), 24–38.

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Gupta, S., Hanssens, D., Hardie, B., Kahn, W., Kumar, V., Lin, N., Ravishanker, N. y Sriram, S. (2006). Modeling Customer Lifetime Value. *Journal of Service Research, 9*(2), 139–155.

Hand, D. y Till, R. (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning, 45*(2), 171–186.

Kuhn, M. y Johnson, K. (2013). *Applied predictive modeling* (Vol. 26, p. 13). Springer.

Lachiche, N. y Flach, P. (2003). Improving Accuracy and Cost of Two–Class and Multi–Class Probabilistic Classifiers using ROC Curves. En *Proceedings of the Twentieth International Conference on Machine Learning*, volume 20, pp. 416–424.

Li, J. y Fine, J. P. (2008). ROC Analysis with Multiple Classes and Multiple Tests: Methodology and Its Application in Microarray Studies. *Biostatistics, 9*(3), 566–576.

Provost, F., Fawcett, T., Kohavi, R. (1998). The Case Against Accuracy Estimation for Comparing Induction Algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453.

Ideas clave

Scikit-learn. (s.f.). *ROC Curve with Visualization API*. . https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_roc_curve_visualization_api.html#sphx-glr-download-auto-examples-miscellaneous-plot-roc-curve-visualization-api-py

Wikipedia. (2024). *Confusion matrix*. https://en.wikipedia.org/wiki/Confusion_matrix

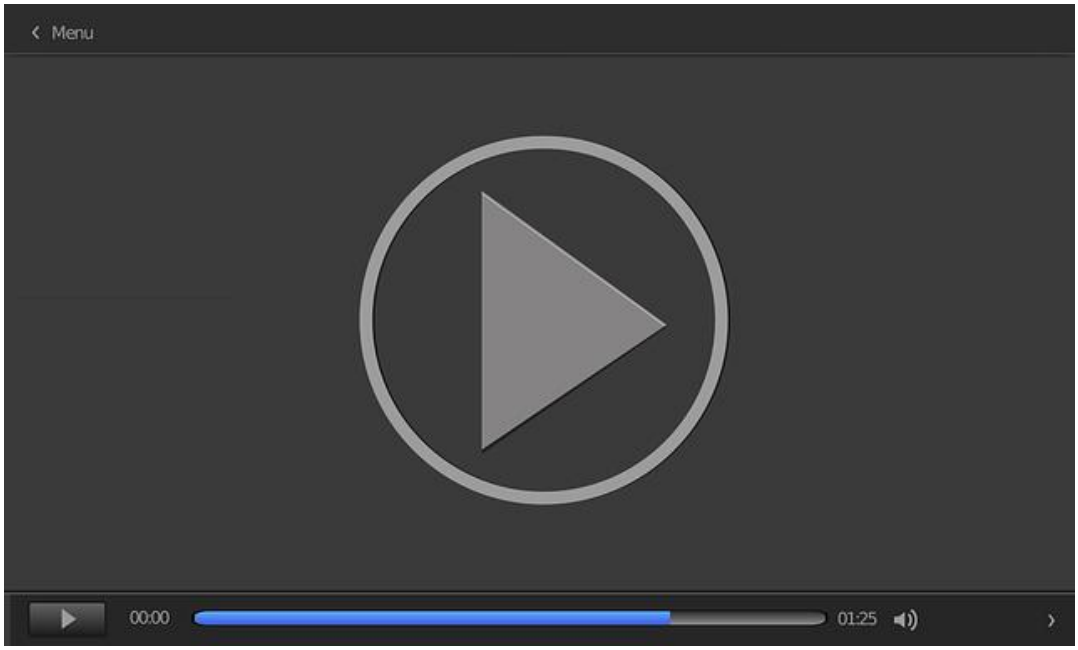
Documentación scikit learn sobre Precision-Recall

Scikit learn. *3.4. Metrics and scoring: quantifying the quality of predictions*. https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics

En la sección de documentación de scikit learn encontrarás una explicación detallada de cada una de estas métricas junto con ejemplos para facilitar su entendimiento.

Explicación de la matriz de confusión

StatQuest with Josh Starmer. (2018, octubre 29). *Machine Learning Fundamentals: The Confusion Matrix* [Vídeo]. YouTube. https://www.youtube.com/watch?v=Kdsp6soqA7o



Accede al vídeo:

https://www.youtube.com/embed/Kdsp6soqA7o

Uno de los conceptos fundamentales en *machine learning* es la matriz de confusión. En el vídeo se explica con un ejemplo y de manera visual cómo interpretar dicha matriz.

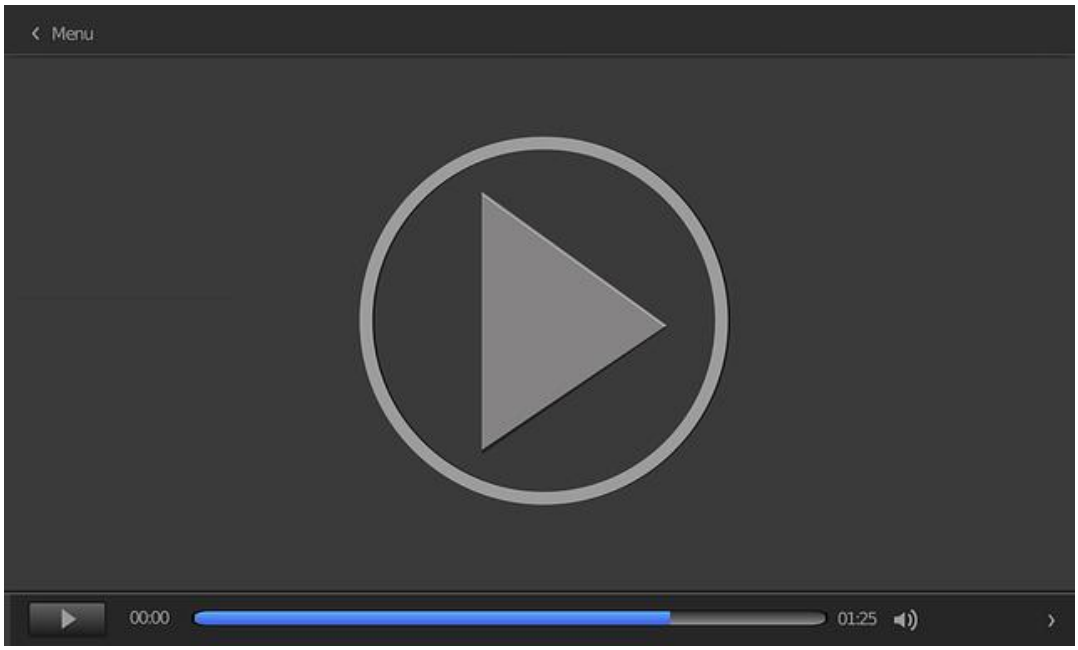
¿Precisión & Recall o Especificidad & Sensibilidad?

Lekhtman, A. (2019). *Data Science in Medicine — Precision & Recall or Specificity & Sensitivit?* Medium. https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1

En este artículo se presentan estas cuatro métricas, con sus características principales, poniendo el foco en qué casos es más relevante o necesario emplear unas frente a otras.

Explicación de las curvas ROC

StatQuest with Josh Starmer. (2019, julio 11). *ROC and AUC, Clearly Explained!* [Vídeo]. YouTube. https://www.youtube.com/watch?v=4jRBRDbJemM



Accede al vídeo:

https://www.youtube.com/embed/4jRBRDbJemM

Los gráficos de la curva ROC y AUC (el área bajo la curva) son útiles para consolidar la información de una tonelada de matrices de confusión en un gráfico único y fácil de interpretar. Este vídeo explica cómo crear e interpretar gráficos ROC paso a paso. Además, se muestra cómo se puede utilizar el AUC para comparar métodos de clasificación.

Test

1. Señala la afirmación correcta: «Los algoritmos de clasificación tienen como objetivo…»:

A. Predecir la clase más probable de entre varias posibles.

B. Predecir la distribución de probabilidad de las clases de cada instancia.

C. Predecir un valor numérico como variable objetivo.

D. Ordenar las etiquetas de clase de un conjunto de datos.

2. Dados dos clasificadores binarios, si se equivocan en las clases más probables de las mismas instancias, significa que:

A. Son igual de buenos.

B. Si uno tiene una mayor incertidumbre que el otro, es peor clasificador.

C. Si uno tiene una mayor incertidumbre que el otro es mejor clasificador.

D. Ninguna de las anteriores.

3. ¿Cuál de las siguientes métricas no es utilizada en los modelos de clasificación?

A. Precisión.

B. Recall.

C. MSE.

D. F-measure.

4. Un matriz de confusión es:

A. Una tabla que organiza las predicciones en función de los valores reales de los datos.

B. Una tabla que tiene las predicciones de los diferentes clasificadores.

C. Una tabla que contiene el valor real y el valor deseado.

D. La matriz de confusión muestra la precisión absoluta de las predicciones.

5. La métrica de accuracy:

A. Se conoce como la ratio de éxito.

B. Representa el número de predicciones correctas entre el total de predicciones.

C. Representa el número de falsos detectados.

D. Ninguna de las anteriores es correcta.

6. Un modelo con gran valor de precisión:

A. Indica que la mayoría de las veces que se predice la clase negativa está en lo cierto.

B. Indica que la mayoría de las veces que se predice la clase positiva está en lo cierto.

C. Indica que el modelo es completamente impreciso y aleatorio.

D. Ninguna de las anteriores.

7. Un modelo con gran valor de recall:

A. Captura un gran porcentaje de ejemplos negativos.

B. Captura un gran porcentaje de ejemplos positivos.

C. Indica que el modelo no captura bien los ejemplos negativos.

D. Indica que el modelo no es capaz de detectar los ejemplos positivos.

8. La métrica F1:

A. Se trata de una forma de calcular el AUC.

B. Combina precisión y recall.

C. Combina sensibilidad y especificidad.

D. Es una medida que se calcula como la suma de la precisión y el recall.

9. El área bajo la curva (AUC):

A. Es un valor comprendido entre 0 y 1.

B. Es un valor comprendido entre 0 y 100.

C. Es un porcentaje entre 0 y 100 %.

D. Es un valor que indica la precisión absoluta del modelo.

10. La curva ROC mide:

A. La ratio de verdaderos positivos y falsos positivos.

B. La ratio de verdaderos positivos y falsos negativos.

C. La ratio de verdaderos negativos y falsos positivos.

D. La ratio de verdaderos positivos y verdaderos negativos.