**Evaluación de modelos de clasificación**

Completado100 XP

* 4 minutos

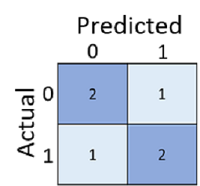
La precisión del entrenamiento de un modelo de clasificación es mucho menos importante que lo bien que funcionará ese modelo cuando se le den nuevos datos no vistos. Después de todo, entrenamos modelos para que se puedan usar en los nuevos datos que se encuentran en el mundo real. Por lo tanto, después de entrenar un modelo de clasificación, debemos evaluar cómo funciona con un conjunto de datos nuevos no vistos.

En las unidades anteriores, creamos un modelo que predecía si un paciente tenía diabetes o no en función de su nivel de glucosa en sangre. Ahora, cuando se aplica a algunos datos que no formaban parte del conjunto de entrenamiento, obtenemos las siguientes predicciones:

| **x** | **y** | **ŷ** |
| --- | --- | --- |
| 83 | 0 | 0 |
| 119 | 1 | 1 |
| 104 | 1 | 0 |
| 105 | 0 | 1 |
| 86 | 0 | 0 |
| 109 | 1 | 1 |

Recordemos que x se refiere al nivel de glucosa en sangre, y se refiere a si son realmente diabéticos, e ŷ se refiere a la predicción del modelo sobre si son diabéticos o no.

El simple cálculo de cuántas predicciones fueron correctas es a veces engañoso o demasiado simplista para que comprendamos el tipo de errores que cometerá en el mundo real. Para obtener información más detallada, podemos tabular los resultados en una estructura denominada *matriz de confusión*, como la siguiente:



La matriz de confusión muestra el número total de casos en los que:

* El modelo predijo 0 y la etiqueta real es 0 (*verdaderos negativos*; arriba a la izquierda)
* El modelo predijo 1 y la etiqueta real es 1 (*verdaderos positivos*, abajo a la derecha)
* El modelo predijo 0 y la etiqueta real es 1 (*falsos negativos*, abajo a la izquierda)
* El modelo predijo 1 y la etiqueta real es 0 (*falsos positivos*, arriba a la derecha)

Las celdas de la matriz de confusión se suelen sombrear para que los valores más altos tengan un sombreado más profundo. De este modo, resulta más fácil ver una tendencia diagonal fuerte de la parte superior izquierda a la inferior derecha, resaltando las celdas en las que el valor predicho y el valor real son los mismos.

A partir de estos valores principales, puede calcular un rango de otras métricas que pueden ayudarle a evaluar el rendimiento del modelo. Por ejemplo:

* **Exactitud**: (VP+VN)/(VP+VN+FP+FN): de todas las predicciones, ¿cuántas son correctas?
* **Coincidencia**: VP/(VP+FN): de todos los casos que *son* positivos, ¿cuántos detectó el modelo?
* **Precisión**: VP/(VP+FP): de todos los casos en los que el modelo predijo ser positivo, ¿cuántos realmente *son* positivos?

**Siguiente unidad: Ejercicio: Realización de la clasificación con métricas alternativas**

**Creación de modelos de clasificación multiclase**

Completado100 XP

* 4 minutos

También es posible crear modelos de clasificación *multiclase*, en los que hay más de dos clases posibles. Por ejemplo, el consultorio médico podría ampliar el modelo de diabetes para clasificar a los pacientes como:

* No diabéticos
* Diabéticos de tipo 1
* Diabéticos de tipo 2

Los valores de probabilidad de las clases individuales aún sumarían un total de 1 (el paciente definitivamente está en una de las tres clases), y el modelo prediría la clase más probable.

**Uso de modelos de clasificación multiclase**

La clasificación multiclase puede considerarse una combinación de varios clasificadores binarios. Hay dos maneras de abordar el problema:

* **Una frente al resto (OVR)**, en el que se crea un clasificador para cada valor de clase posible, con un resultado positivo para los casos en los que la predicción es esta clase y predicciones negativas para los casos en los que la predicción es cualquier otra clase. Por ejemplo, un problema de clasificación con cuatro clases de formas posibles (cuadrado, círculo, triángulo, hexágono) requeriría cuatro clasificadores que predigan:
  + cuadrado o no
  + círculo o no
  + triángulo o no
  + hexágono o no
* **Uno frente a uno (OVO)**, en el que se crea un clasificador para cada par de clases posible. El problema de clasificación con cuatro clases de formas requeriría los siguientes clasificadores binarios:
  + cuadrado o círculo
  + cuadrado o triángulo
  + cuadrado o hexágono
  + círculo o triángulo
  + círculo o hexágono
  + triángulo o hexágono

En los dos enfoques, el modelo general debe tener en cuenta todas estas predicciones para determinar a qué categoría única pertenece el elemento.

Afortunadamente, en la mayoría de los marcos de aprendizaje automático, incluido Scikit-learn, la implementación de un modelo de clasificación multiclase no es significativamente más compleja que la clasificación binaria y, en la mayoría de los casos, los estimadores utilizados para la clasificación binaria admiten implícitamente la clasificación multiclase abstrayendo un algoritmo OVR o un algoritmo OVO, o permitiendo la elección de cualquiera de ellos.

**Siguiente unidad: Ejercicio: Entrenamiento y evaluación de modelos de clasificación multiclase**

**Resumen**

Completado100 XP

* 1 minuto

En este módulo, ha aprendido cómo se puede usar la clasificación para crear un modelo de aprendizaje automático que predice categorías o *clases*. Después, ha usado el marco de **Scikit-learn** en Python para entrenar y evaluar un modelo de clasificación.

Aunque Scikit-learn es un marco popular para escribir código con el fin de entrenar modelos de clasificación, también puede crear soluciones de aprendizaje automático para clasificación con las herramientas gráficas de Microsoft Azure Machine Learning. Para obtener más información sobre el desarrollo sin código de modelos de clasificación mediante Azure Machine Learning, consulte el módulo [Creación de un modelo de clasificación con el diseñador de Azure Machine Learning](https://learn.microsoft.com/es-es/learn/modules/create-classification-model-azure-machine-learning-designer/).

**Desafío: Clasificación de vinos**

¿Le parece difícil entrenar un modelo de clasificación? Pruebe el desafío en el cuaderno [/challenges/03 - Wine Classification Challenge.ipynb](https://github.com/MicrosoftDocs/ml-basics/blob/master/challenges/03%20-%20Wine%20Classification%20Challenge.ipynb) para ver si puede clasificar los vinos en sus variedades de uva.