

Resumen del capítulo: métricas de clasificación

Equilibrio y desequilibrio de las clases

Hay un fuerte desequilibrio de clases en algunos problemas, lo que afecta negativamente la forma en que se entrena el modelo. Las clases están desequilibradas cuando su proporción está lejos de 1:1. Se observa equilibrio de clase si su número es aproximadamente igual.

La exactitud (accuracy) no elimina este desequilibrio de clases. ¡Necesitamos una nueva métrica!

Ya sabes que una clase con la etiqueta "1" se llama positiva y una clase con la etiqueta "0" se llama negativa.

Si combinamos estas respuestas con predicciones, obtendremos la siguiente división:

Respuestas verdaderas positivas (VP): el modelo etiquetó un objeto como "1", y su valor real también es "1"

Respuestas verdaderas negativas (VN): el modelo etiquetó un objeto como "0", y su valor real también es "0"

Respuestas falsas positivas (FP): el modelo etiquetó un objeto como "1", pero su valor real es "0"

Respuestas falsas negativas (FN): el modelo etiquetó un objeto como "0", pero su valor real es "1"

Matriz de confusión

Cuando VP, FP, VN y FN se recopilan en una tabla, se denomina **matriz de confusión**. La matriz se forma de la siguiente manera:

- Las etiquetas del algoritmo (0 y 1) se colocan en el eje horizontal ("Predicciones").
- Las etiquetas verdaderas de la clase (0 y 1) se colocan en el eje vertical ("Respuestas").

Lo que obtienes:

1. Las predicciones correctas están en la diagonal principal (desde la esquina superior izquierda):
 - VN en la esquina superior izquierda
 - VP en la esquina inferior derecha
2. Las predicciones incorrectas están fuera de la diagonal principal:
 - FP en la esquina superior derecha
 - FN en la esquina inferior izquierda

Respuestas	0	1
	Verdadera Negativa	Falsa Positiva
1	Falsa Negativa	Verdadera Positiva
Predicciones		

La matriz de confusión se encuentra en el módulo `sklearn.metrics`, que ya conoces. La función `confusion_matrix()` toma *respuestas* y *predicciones correctas* y devuelve una *matriz de confusión*.

```
from sklearn.metrics import confusion_matrix

print(confusion_matrix(target, predictions))
```

Recall

La matriz de confusión te ayudará a crear nuevas métricas. Comencemos con recall.

Recall (sensibilidad) revela qué porción de respuestas positivas ha identificado el modelo entre todas las respuestas. Recall se calcula usando esta fórmula:

$$Recall = \frac{VP}{VP + FN}$$

Recall es una métrica de evaluación que mide la proporción de respuestas VP entre todas las respuestas que realmente tienen una etiqueta 1. Queremos que el valor de recall esté cerca de 1. Esto significaría que el modelo es bueno para identificar verdaderos positivos. Si está más cerca de cero, es necesario revisar y reparar el modelo.

La función `recall_score()` se encuentra en el módulo `sklearn.metrics`. La función toma respuestas y predicciones correctas y devuelve el valor de recuperación.

```
from sklearn.metrics import recall_score

print(recall_score(target, predictions))
```

Precisión

Otra métrica para evaluar la calidad de una predicción de clase objetivo es la **precisión**.

La precisión mide cuántas respuestas negativas encontró el modelo mientras buscaba respuestas positivas. Cuantas más respuestas negativas se encuentren, menor será la precisión.

La *precisión* se calcula usando esta fórmula:

$$Precisión = \frac{VP}{VP + FP}$$

Queremos que el valor de precisión esté cerca de 1. La función `precision_score()` se encuentra en el módulo `sklearn.metrics`. La función toma respuestas y predicciones correctas y devuelve el valor de precisión.

```
from sklearn.metrics import precision_score

print(precision_score(target, predictions))
```

Valor F1

Por separado, *recall* y *precisión* no son muy informativas. Debes aumentar simultáneamente los valores de ambas métricas... ¿O podemos recurrir a una nueva métrica que las combine!

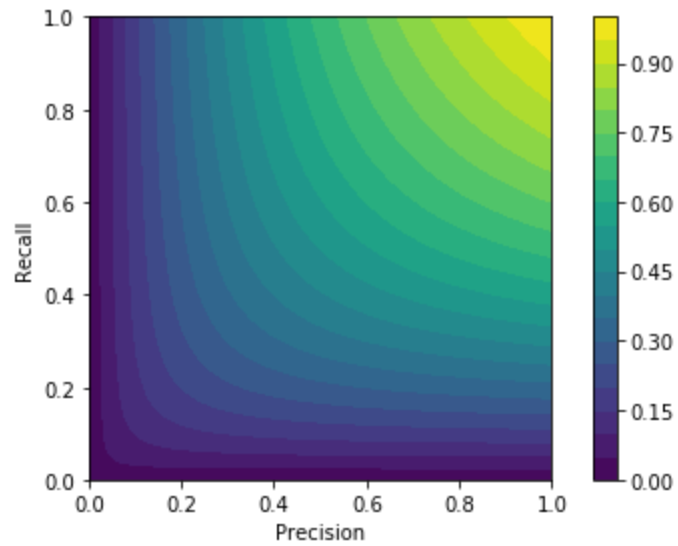
Recall y *precisión* evalúan la calidad de las predicciones de la clase positiva desde diferentes ángulos. *Recall* describe qué tan bien entendió el modelo las propiedades de esta clase y qué tan bien reconoció la clase. *Precisión* detecta si el modelo se está excediendo al asignar demasiadas etiquetas positivas.

Ambas métricas son importantes. Las métricas de agregación, una de las cuales es el **valor F1**, ayudan a controlarlas de forma simultánea. Esta es la media armónica de *recall* y *precisión*. En F1, 1 significa que la relación de *recall* a *precisión* es 1:1.

$$F1 = \frac{2 \cdot \textit{Precisión} \cdot \textit{Recall}}{\textit{Precisión} + \textit{Recall}}$$

Es importante comprender que cuando *recall* o *precisión* están cerca de cero, la media armónica misma se acerca a 0.

La gráfica muestra los valores del valor F1 con diferentes valores de *recall* y *precisión*. El color azul corresponde a cero y el color amarillo corresponde a uno.



Si una clase positiva se pronostica de manera deficiente en una de las escalas (recall o precisión), un valor F1 cercano a cero indicará que la predicción de la clase 1 ha fallado.

La *función* `f1_score()` se encuentra en el módulo `sklearn.metrics`. La función toma respuestas y predicciones correctas y devuelve la media armónica de *recall* y *precisión*.

```
from sklearn.metrics import f1_score  
  
print(f1_score(target, predictions))
```