

CONFUSION MATRIX

È una matrice dove vengono contati il numero di:

- 1) **TRUE POSITIVE**: tuple correttamente classificate
- 2) **FALSE NEGATIVE**: tuple classificate erroneamente con una classe diversa da quella sotto esame
- 3) **FALSE POSITIVE**: tuple che hanno un'altra classe ma sono classificate con la classe sotto esame.
- 4) **TRUE NEGATIVE**: Tuple che hanno una classe che non è quella in esame, e sono classificate con una classe ancora diversa da quella sotto esame.

PREDETTA ATTUALI	C_1	$\neg C_1$
C_1	TP	FN
$\neg C_1$	FP	TN

La confusion matrix può essere fatta anche con tutte le classi etichette del dataset.

Un classificatore perfetto è una matrice diagonale

ACCURATEZZA

$$\text{Accuracy} = \frac{TP + TN}{|D|}$$

ERROR RATE

$$\text{Error} = (1 - \text{Accuracy})$$

$$\text{Error} = \frac{FP + FN}{|D|}$$

SENSITIVITY o RECALL Valore perfetto: 1

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

Quale percentuale di tuple che hanno classe C sono effettivamente classificate con C dal nostro modello di classificazione.

SPECIFICITY

$$\text{specificity} = \frac{TN}{TN + FP} = \frac{TN}{N}$$

Ci dà indicazione su quanto il modello sia bravo a riconoscere la classe negativa o le altre classi diverse dalla corrente.

PRECISION Valore perfetto: 1

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

Ci dà indicazione sul numero di tuple che vengono classificate con C poi effettivamente hanno C come classe.

F-MEASURES o F-SCORES = F_1

$$F = \frac{2 \times \text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}}$$

Di fatti è la media armonica fra precision e recall.

F_β :

Misura pesata di precision e recall

$$F_\beta = \frac{(1 + \beta) \text{Precision} \times \text{recall}}{\beta \cdot (\text{Precision} + \text{recall})}$$

Se per esempio usiamo F_2 allora pesiamo recall più di precision, $F_{0.5}$ fa il contrario

HOLDOUT METHOD

Dividere in training e test e allenare e testare il modello K volte, e ogni volta si cambiano i campioni dentro training e test.

CROSS-VALIDATION

Dividere il dataset in K partizioni, $K-1$ vengono considerate training e una come test. Ripetere l'allenamento e il test K volte, e ogni volta scegliere un test set diverso fra le K fold disponibili.

BOOTSTRAP

L'idea è di usare il campionamento con rimpiazzamento.

.632 method

Crea il training campionando d tuple d volte con rimpiazzamento. Così circa il 63,2% dei dati va nel training e il restante 36,8% nel test. Questo perché la probabilità che un oggetto non venga mai scelto è $(1 - 1/d)^d$. Per d grandi questa probabilità si aggira a $e^{-1} = 0,368$. Ripetendo il campionamento K volte abbiamo l'accuracy:

$$Acc(M) = \frac{1}{K} \sum_{i=1}^K \left(0,632 \cdot Acc(M_i)_{\text{TEST}} + 0,368 \cdot Acc(M_i)_{\text{TRAIN}} \right)$$