

# 17. Metriche

Corso di Python per il Calcolo Scientifico

# Outline

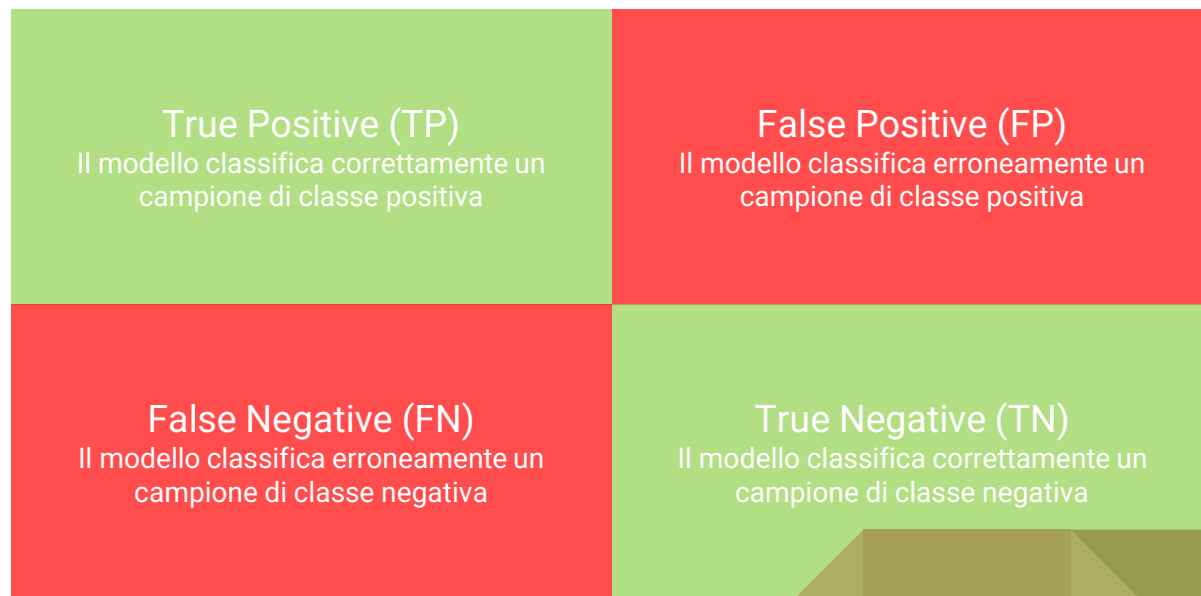
- Soglia di decisione
- Veri positivi, falsi negativi
- L'accuratezza
- La precisione
- Il recall
- Tuning della soglia di decisione
- Metriche per la regressione

# Soglia di decisione

- La regressione logistica **ci restituisce una probabilità**
- Ad esempio:
  - Se  $p(spam) = 0.99 \Rightarrow$  il messaggio è (ragionevolmente) di spam
  - Se  $p(spam) = 0.08 \Rightarrow$  il messaggio è (ragionevolmente) legittimo
- Cosa accade per  $p(spam) = 0.54$ ?
- Occorre impostare una **soglia di decisione**
- Questa **dipende dal problema**
  - La tentazione sarebbe di impostarla a 0.5, **ma non sarebbe corretto!**

# Veri positivi, falsi negativi

- **Spam: classe positiva**
- **Legittima: classe negativa**
- **TP: spam classificato come tale**
- **TN: legittima classificata come tale**
- **FP: legittima classificata come spam**
- **FN: spam classificato come legittima**



# L'accuratezza (1)

- La prima metrica che possiamo usare per valutare un algoritmo è l'**accuratezza (accuracy)**
- È definita come la percentuale di predizioni corrette sul totale delle predizioni, ovvero:

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

- Viene valutata in percentuale
- Dà una prima indicazione (non esaustiva) sulla bontà del nostro algoritmo
- In Scikit Learn si ottiene mediante il metodo **accuracy\_score**

## L'accuratezza (2)

- Numero totale di email: 100
- Spam: 10, di cui 5 individuate come tali
- Legittime: 90, di cui 5 segnalate come spam

$$AC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5 + 85}{5 + 85 + 5 + 5} = 0.90 = 90\%$$

- Tuttavia, il classificatore ha individuato correttamente **soltanto il 50% delle email di spam**
- Di conseguenza, l'accuratezza **può non bastare**

# La precisione

- La **precisione** ci indica la proporzione di valori di classe positiva individuati correttamente

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

- Nel nostro caso:

$$P = \frac{5}{5 + 5} = 0.50 = 50\%$$

- Il classificatore ha quindi una precisione del 50%
- In pratica, **la metà delle mail indicate come spam sono legittime**
- Scikit Learn permette il calcolo della precisione con il metodo `precision_score()`

# Il recall

- Il **recall** ci indica la porzione di veri positivi identificati correttamente dall'algoritmo:

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

- Nel nostro caso:

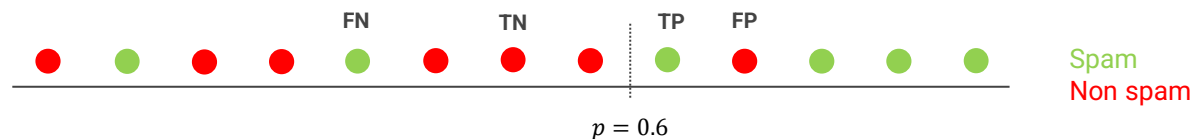
$$R = \frac{5}{5 + 5} = 0.50 = 50\%$$

- Il classificatore ha quindi anche un recall del 50%.
- In pratica, **la metà delle mail di spam sono indicate come legittime.**
- Scikit Learn permette il calcolo della precisione con il metodo `recall_score()`.



# Tuning della soglia di decisione (1)

- L'efficacia del modello è deducibile congiuntamente dall'analisi **congiunta** di recall e precisione.
- Vediamo un esempio:



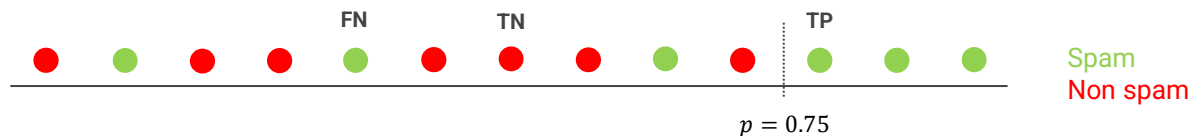
- In questo caso:

$$P = \frac{TP}{TP + FP} = \frac{4}{4 + 1} = 80\%$$

$$R = \frac{TP}{TP + FN} = \frac{4}{4 + 2} = 66\%$$

# Tuning della soglia di decisione (2)

- Proviamo ad innalzare la soglia di decisione, portandola ad esempio al 75%.



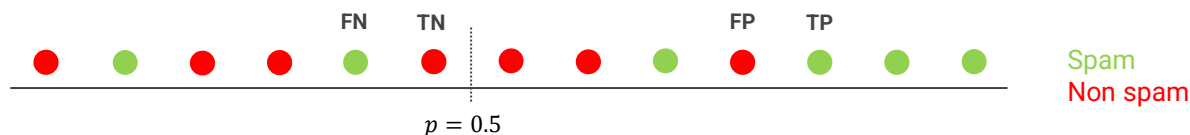
- In questo caso:

$$P = \frac{TP}{TP + FP} = \frac{3}{3} = 100\%$$

$$R = \frac{TP}{TP + FN} = \frac{3}{3 + 3} = 50\%$$

# Tuning della soglia di decisione (3)

- Proviamo ad abbassare la soglia di decisione, portandola ad esempio al 50%.



- In questo caso:

$$P = \frac{TP}{TP + FP} = \frac{4}{4 + 3} \approx 57\%$$

$$R = \frac{TP}{TP + FN} = \frac{4}{4 + 2} = 66\%$$

# Metriche per la regressione (1)

- Il **Mean Squared Error** definisce l'errore quadratico medio della predizione:

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

- È implementato mediante la funzione `mean_squared_error()`.
- È particolarmente sensibile al valore di  $y_i$ .
- Il **Mean Average Percentage Error (MAPE)** va a compensare parte dei limiti dell'MSE dividendo il discostamento tra il valore predetto e quello vero per quest'ultimo.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{\max(\epsilon, y_i)} \%$$

# Metriche per la regressione (2)

- È implementato in Scikit Learn con la funzione `mean_absolute_percentage_error()`.
- Rispetto all'MSE introduce un termine di normalizzazione.
- Il **coefficiente di determinazione**  $R^2$  permette di **spiegare la varianza** del modello.
- Indicando come *varianza* (o *variabilità*) di un fenomeno il modo in cui combina le feature in ingresso per dare un valore di regressione in uscita, un modello ad alto  $R^2$  sarà in grado di modellare la maggior parte di questa variabilità, assicurando performance di fitting migliori rispetto a quelle di un modello a basso  $R^2$ .
- Viene implementato in Scikit Learn con la funzione `r2_score()`.

# Domande?

42