

- ▶ Creare uno script in python, un file *.py* chiamato «*metrics.py*».

Esercitazione

- Il file avrà il seguente contenuto.

```
1  import sys
2  import numpy as np
3
4  np.random.seed(42)
5
6
7  class Metrics():
8
9  >  def __init__(self, classes : list[str], num_data: int) -> None: ...
18
19      # Calcola la matrice di confusione fra le classi.
20  >  def compute_confusion_matrix(self) -> None: ...
30
31      # Calcola l'accuracy sulla confusion matrix.
32  >  def accuracy(self) -> float: ...
40
41      # Calcola la recall per la classe indicata.
42  >  def recall(self, class_id: int) -> float: ...
59
60      # Calcola la precision per la classe indicata.
61  >  def precision(self, class_id: int) -> float: ...
78
79      # Calcola l'f1-score per la classe indicata.
80  >  def f1_score(self, class_id: int) -> float: ...
86
87      # Calcola il numero di campioni veri per la classe indicata.
88  >  def support(self, class_id: int) -> int: ...
99
100     # Mostra un report di tutte le informazioni.
101  >  def report(self) -> None: ...
115
116     # Verifica se l'indice classe e' valido.
117  >  def __valid_class_id(self, class_id: int) -> bool: ...
119
120  if __name__ == '__main__':
121
122      mt = Metrics(['circle', 'square', 'triangle'], 20)
123      mt.report()
```

Esercitazione

► Completare i metodi nel seguente ordine:

- *__init__*
- *__valid_class_id*
- *compute_confusion_matrix*
- *accuracy*
- *precision*
- *recall*
- *f1-score*
- *support*
- *report*

Esercitazione

- ▶ Al metodo `__init__` sono passati:
 - ▶ `classes`: una lista di etichette di classe.
 - ▶ `num_data`: il numero di dati casuali da generare.
- ▶ Nel metodo `__init__` inizializzare le seguenti variabili di classe:
 - ▶ `classes`: la lista di classi passate.
 - ▶ `num_classes`: il numero totale di classi.
 - ▶ `real_y`: array casuale di `num_data` interi in `[0 - num_classes]`.
 - ▶ `pred_y`: array casuale di `num_data` interi in `[0 - num_classes]`.
 - ▶ `confusion_matrix`: valore **None**.

Esercitazione

- ▶ Il metodo `__valid_class_id`:
 - ▶ Restituisce un booleano.
 - ▶ True se l'indice di classe è **valido**.
 - ▶ False in caso contrario.

Esercitazione

- ▶ Il metodo *compute_confusion_matrix*:
 - ▶ Crea un array $N \times N$ con $N = \text{num_classes}$.
 - ▶ Riempie l'array con i valori della matrice di confusione calcolati fra etichette reali, *real_y*, e previsioni *pred_y*.
 - ▶ Assegna l'array alla variabile di classe *confusion_matrix*.

Esercitazione

- ▶ Il metodo *accuracy*:
 - ▶ Se ancora **None**, calcola la matrice di confusione.
 - ▶ Calcola l'accuratezza complessiva per l'intera matrice di confusione.

Esercitazione

- ▶ Il metodo *recall*:
 - ▶ Verifica che l'id passato sia valido.
 - ▶ In caso contrario esce dallo script: `sys.exit(-1)`
 - ▶ Se ancora **None**, calcola matrice di confusione.
 - ▶ Utilizza la matrice di confusione per calcolare la recall della classe richiesta.

Esercitazione

► Il metodo *precision*:

- Verifica che l'id passato sia valido.
- In caso contrario esce dallo script: `sys.exit(-1)`
- Se ancora **None**, calcola matrice di confusione.
- Utilizza la matrice di confusione per calcolare la precision della classe richiesta.

Esercitazione

- ▶ Il metodo *f1_score*:
 - ▶ Verifica che l'id passato sia valido.
 - ▶ In caso contrario esce dallo script: `sys.exit(-1)`
 - ▶ Se ancora **None**, calcola matrice di confusione.
 - ▶ Utilizza la matrice di confusione per calcolare l'*f1_score* della classe richiesta.

Esercitazione

- ▶ Il metodo *support*:
 - ▶ Verifica che l'id passato sia valido.
 - ▶ In caso contrario esce dallo script: `sys.exit(-1)`
 - ▶ Se ancora **None**, calcola matrice di confusione.
 - ▶ Utilizza la matrice di confusione per calcolare il numero totale di etichette reali per la classe richiesta.

Esercitazione

- ▶ Il metodo *report*:
 - ▶ Se ancora **None**, calcola la matrice di confusione.
 - ▶ Mostra a video la matrice calcolata.
 - ▶ Mostra l'accuratezza globale.
 - ▶ Mostra le metriche per ogni classe:
 - ▶ Precision.
 - ▶ Recall.
 - ▶ F1-score.
 - ▶ Support.

Esercitazione

- ▶ Di seguito un esempio di output all'esecuzione dello script:

```
Confusion matrix:  
  
[[3 2 0]  
 [1 0 5]  
 [2 4 3]]  
  
Accuracy: 0.3  
  
Class "circle":  
|__precision : 0.50  
|__recall    : 0.60  
|__f1_score  : 0.55  
|__support   : 5.00  
  
Class "square":  
|__precision : 0.00  
|__recall    : 0.00  
|__f1_score  : 0.00  
|__support   : 6.00  
  
Class "triangle":  
|__precision : 0.38  
|__recall    : 0.33  
|__f1_score  : 0.35  
|__support   : 9.00
```

Esercitazione

- ▶ Utilizzare le sole librerie presenti nello script.

Esercitazione