Precision, Recall & F1 for Multi-class Classification

Multi-class Classification Model (C classes)

Confusion Matrix for Each Class

	Predicted	
Actual	TP_1	FN_1
	FP_1	TN_1

	Predicted	
Actual	TP_2	FN_2
	FP_2	TN_2

..

	Predicted	
Actual	TP_C	FN_C
	FP_C	TN_C

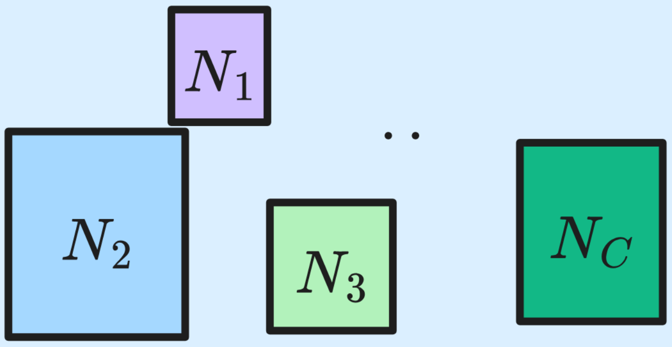
$$\text{Let } P_i = \frac{TP_i}{TP_i + FP_i} \quad \text{be the precision for the } i^{\text{th}} \text{ class}$$

$$\text{Let } R_i = \frac{TP_i}{TP_i + FN_i} \quad \text{be the recall for the } i^{\text{th}} \text{ class}$$

$$\text{Let } F_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \quad \text{be the F1-score for the } i^{\text{th}} \text{ class}$$

$$\text{Let } N_i = \text{Number of true instances of class-i (support)}$$

$$N = N_1 + N_2 + N_3 + .. + N_C$$



@AIinMinutes

Precision, Recall & F1 for Multi-class Classification

Macro-averaged Metrics

Idea: The metric is computed per class, then all values are averaged (unweighted).

Feature: Treats all classes equally, so minority class performance matters just as much.

$$\text{Macro Precision} = \frac{1}{C}\sum_{i=1}^{C} P_i = \frac{1}{C}\sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i}$$

$$\text{Macro Recall} = \frac{1}{C}\sum_{i=1}^{C} R_i = \frac{1}{C}\sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i}$$

$$\text{Macro F1-score} = \frac{1}{C}\sum_{i=1}^{C} F_i$$

Micro-averaged Metrics

@AIinMinutes

Idea: Combine all confusion matrices (sum), then compute the metric on the grand matrix.

Feature: Focuses on overall performance, not per-class. Can be high even if minority class performance is poor.

$$\text{Micro Precision} = \text{Recall} = \text{F1} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FP_i}$$

Precision, Recall & F1 for Multi-class Classification

Weighted Metrics

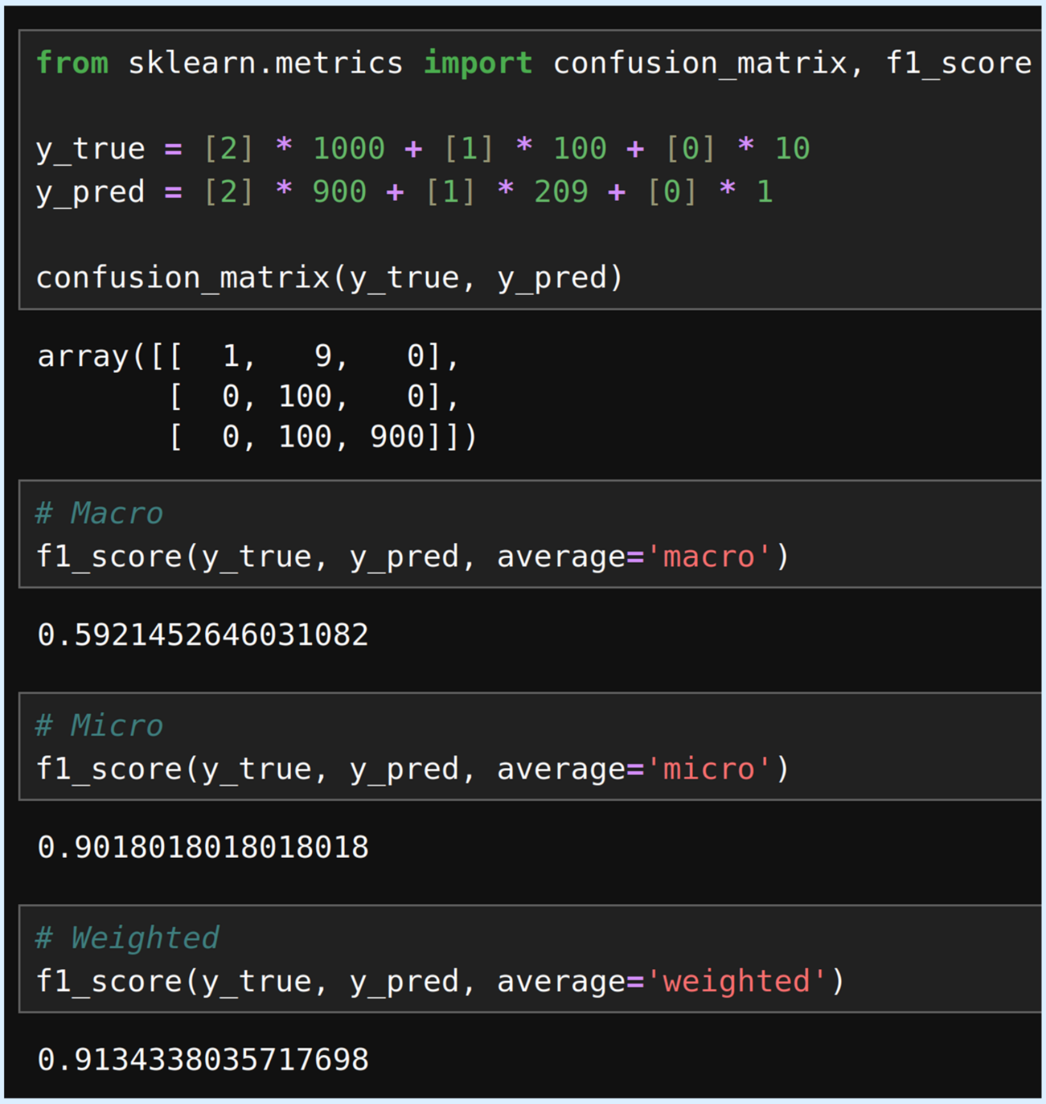
Idea: Compute the metric per class,
then all values are averaged (weighted by class support).

Feature: Reflects the representative importance
of each class (by its support).

$$\text{Weighted Precision} = \sum_{i=1}^{C} \frac{N_i}{N} \cdot P_i = \sum_{i=1}^{C} \frac{N_i}{N} \cdot \frac{TP_i}{TP_i + FP_i}$$

$$\text{Weighted Recall} = \sum_{i=1}^{C} \frac{N_i}{N} \cdot R_i = \sum_{i=1}^{C} \frac{N_i}{N} \cdot \frac{TP_i}{TP_i + FN_i}$$

$$\text{Weighted F1-score} = \sum_{i=1}^{C} \frac{N_i}{N} \cdot F_i$$



```
from sklearn.metrics import confusion_matrix, f1_score

y_true = [2] * 1000 + [1] * 100 + [0] * 10
y_pred = [2] * 900 + [1] * 209 + [0] * 1

confusion_matrix(y_true, y_pred)
```

```
array([[  1,   9,   0],
       [  0, 100,   0],
       [  0, 100, 900]])
```

```
# Macro
f1_score(y_true, y_pred, average='macro')
```

```
0.5921452646031082
```

```
# Micro
f1_score(y_true, y_pred, average='micro')
```

```
0.9018018018018018
```

```
# Weighted
f1_score(y_true, y_pred, average='weighted')
```

```
0.9134338035717698
```

@AIinMinutes