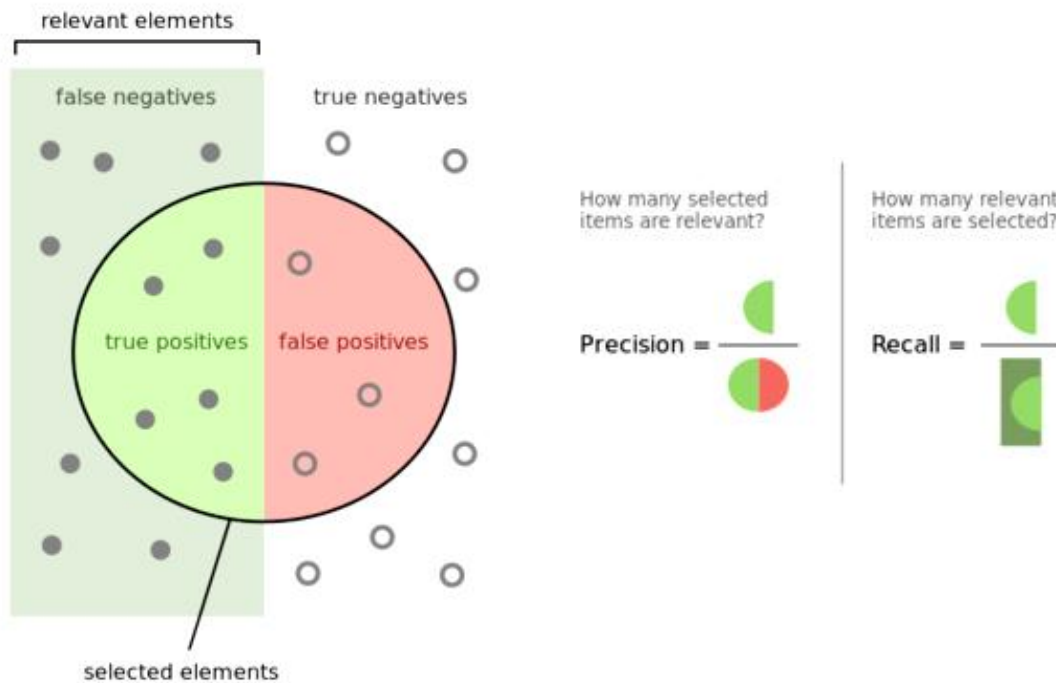


Practice 9

Metrics for multi-class classification

Precision, Recall and F1. These are vital metrics utilized for unbalanced test sets, regardless of the standard type and error rate. For example, most of the test samples have a class label. Metrics *Precision* and *Recall* are defined as:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$



F1 the harmonic average of *Precision* and *Recall* is defined:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}.$$

If we express it in terms of True Positive (TP), False Positive (FP), and False Negative (FN), we get this equation:

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}.$$

The desired results will be obtained when the accuracy, $F1$ and $Recall$ value reach 1. On the contrary, when the values become 0, the worst result is obtained. For the multi-class classification problem, the precision and recall value of each class can be calculated separately, and then the performance of the individual and whole can be analyzed.

A more general F_β score, that uses a positive real factor β , where is chosen such that recall is considered β times as important **as precision**:

$$F1 = (1 + \beta^2) \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}.$$

Multi-label metrics. Compared with single-label text classification, multi-label text classification divides the text into multiple category labels, and the number of category labels is variable. Thus, there are some metrics designed for multi-label text classification for an « N »-class dataset.

Micro-F1 is a measure that considers the overall accuracy and recall of all labels. The *Micro-F1* is defined as:

$$Micro - F1 = \frac{2P_t \cdot R_t}{P + R}.$$

$$P = \frac{\sum_{t \in N} TP_t}{\sum_{t \in N} TP_t + FP_t}, \quad R = \frac{\sum_{t \in N} TP_t}{\sum_{t \in N} TP_t + FN_t}$$

Macro-F1 calculates the average $F1$ of all labels. Unlike *Micro-F1*, which sets even weight to every example, *Macro-F1* sets the same weight to all labels in the average process. Formally, *Macro-F1* is defined as:

$$Macro - F1 = \frac{1}{N} \sum_{t \in N} \frac{2P_t \cdot R_t}{P_t + R_t}.$$

$$P_t = \frac{TP_t}{TP_t + FP_t}, \quad R_t = \frac{TP_t}{TP_t + FN_t}$$

Sample-weighted F1 score is ideal for computing the net $F1$ score for class-imbalanced data distribution. As the name suggests, it is a weighted average of the class-wise $F1$ scores, the weights of which are determined by the number of samples available in that class.

$$Weighted - F1 = \sum_{i=1}^N w_i \cdot F1_i,$$

$$w_i = \frac{\text{Number of objects in class } i}{\text{Total number of objects}}.$$

To illustrate the concepts of averaging F1 scores, we will use the following example. Imagine we have trained an **image classification model** on a **multi-class** dataset containing images of **three** classes: **Airplane**, **Boat**, and **Car**.

We use this model to **predict** the classes of **ten** test set images. Here are the **raw predictions**:

No	Actual	Predicted	Match
1	Airplane	Airplane	✓
2	Car	Boat	✗
3	Car	Car	✓
4	Car	Car	✓
5	Car	Boat	✗
6	Airplane	Boat	✗
7	Boat	Boat	✓
8	Car	Airplane	✗
9	Airplane	Airplane	✓
10	Car	Car	✓

Upon running `sklearn.metrics.classification_report`, we get the following classification report:

	precision	recall	f1-score	support
Aeroplane	0.67	0.67	0.67	3
Boat	0.25	1.00	0.40	1
Car	1.00	0.50	0.67	6
accuracy			0.60	10
macro avg	0.64	0.72	0.58	10
weighted avg	0.82	0.60	0.64	10







Per-Class
F1 scores

Average
F1 scores




The columns (in orange) with the **per-class** scores (i.e., score for each class) and **average** scores are the focus of our discussion.

We can see from the above that the dataset is **imbalanced** (only one out of ten test set instances is 'Boat'). Thus the **proportion of correct matches** (aka accuracy) would be ineffective in assessing model performance.

Instead, let us look at the **confusion matrix** for a holistic understanding of the model predictions.

		Predicted		
		 Airplane	 Boat	 Car
Actual	 Airplane	2	1	0
	 Boat	0	1	0
	 Car	1	2	3




The confusion matrix above allows us to compute the critical values of True Positive (TP), False Positive (FP), and False Negative (FN), as shown below.

Label	True Positive (TP)	False Positive (FP)	False Negative (FN)
 Airplane	2	1	1
 Boat	1	3	0
 Car	3	0	3

The above table sets us up nicely to compute the **per-class** values of **precision**, **recall**, and F1 score for each of the three classes.

It is important to remember that in **multi-class classification**, we calculate the **F1 score for each class in a One-vs-Rest (OvR)** approach instead of a single overall F1 score, as seen in binary classification.

In this **OvR** approach, we determine the metrics for each class separately, as if there is a different classifier for each class. Here are the per-class metrics (with the F1 score calculation displayed):

Label	True Positive (TP)	False Positive (FP)	False Negative (FN)	Precision	Recall	F1 Score
 Airplane	2	1	1	0.67	0.67	$2 * (0.67 * 0.67) / (0.67 + 0.67)$ = 0.67
 Boat	1	3	0	0.25	1.00	$2 * (0.25 * 1.00) / (0.25 + 1.00)$ = 0.40
 Car	3	0	3	1.00	0.50	$2 * (1.00 * 0.50) / (1.00 + 0.50)$ = 0.67

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$

However, instead of having multiple per-class F1 scores, it would be better to **average** them to obtain a **single number** to describe overall performance.




Now, let's discuss the **averaging** methods that led to the classification report's **three different average F1 scores**.

Macro Average

Macro averaging is perhaps the most straightforward among the numerous averaging methods.

The macro-averaged F1 score (or macro F1 score) is computed using the arithmetic mean (**unweighted** mean) of all the per-class F1 scores.

This method treats all classes equally regardless of their **support** values.

Label	Per-Class F1 Score	Macro-Averaged F1 Score
 Airplane	0.67	$\frac{0.67 + 0.40 + 0.67}{3}$ = 0.58
 Boat	0.40	
 Car	0.67	

The value of **0.58** we calculated above matches the macro-averaged F1 score in our classification report.



	precision	recall	f1-score	support
Aeroplane	0.67	0.67	0.67	3
Boat	0.25	1.00	0.40	1
Car	1.00	0.50	0.67	6
accuracy			0.60	10
macro avg	0.64	0.72	0.58	10
weighted avg	0.82	0.60	0.64	10

Weighted Average

The **weighted-averaged** F1 score is calculated by taking the mean of all per-class F1 scores **while considering each class's support**.

Support refers to the number of actual occurrences of the class in the dataset. For example, the support value of 1 in **Boat** means that there is only one observation with an actual label of Boat.

The ‘weight’ essentially refers to the proportion of each class’s support relative to the sum of all support values.

Label	Per-Class F1 Score	Support	Support Proportion	Weighted Average F1 Score
 Airplane	0.67	3	0.3	$(0.67 * 0.3) + (0.40 * 0.1) + (0.67 * 0.6) = \mathbf{0.64}$
 Boat	0.40	1	0.1	
 Car	0.67	6	0.6	
Total	-	10	1.0	

With weighted averaging, the output average would have accounted for the contribution of each class as weighted by the number of examples of that given class.




The calculated value of **0.64** corresponds to the weighted-averaged F1 score in our classification report.

	precision	recall	f1-score	support
Aeroplane	0.67	0.67	0.67	3
Boat	0.25	1.00	0.40	1
Car	1.00	0.50	0.67	6
accuracy			0.60	10
macro avg	0.64	0.72	0.58	10
weighted avg	0.82	0.60	0.64	10

Micro Average

Micro averaging computes a **global average** F1 score by counting the **sums** of the True Positives (TP), False Negatives (FN), and False Positives (FP).

We first sum the respective TP, FP, and FN values across all classes and then plug them into the F1 equation to get our micro F1 score.

Label	True Positive (TP)	False Positive (FP)	False Negative (FN)	Micro-Averaged F1 Score
 Airplane	2	1	1	$\frac{TP}{TP + \frac{1}{2}(FP + FN)} = \frac{6}{6 + \frac{1}{2}(4 + 4)} = \mathbf{0.60}$
 Boat	1	3	0	
 Car	3	0	3	
TOTAL	6	4	4	




In the classification report, you might be wondering why our micro F1 score of **0.60** is displayed as ‘accuracy’ and why there is **NO row stating ‘micro avg’**.

	precision	recall	f1-score	support
accuracy			0.60	10
macro avg	0.64	0.72	0.58	10
weighted avg	0.82	0.60	0.64	10

This is because micro-averaging essentially computes the **proportion** of **correctly classified** observations out of all observations so that there is no TN in multi-class classification (TN = TP). If we think about this, this definition is what we use to calculate overall **accuracy**.

$$\begin{aligned}
 \text{multiclass Accuracy} &= \frac{TP + TP}{TP + TP + FP + FN} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \\
 &= \frac{TP}{TP + \frac{1}{2}(FP + FN)} = F1
 \end{aligned}$$

Furthermore, if we were to do micro-averaging for precision and recall, we would get the same value of **0.60**.

Label	True Positive (TP)	False Positive (FP)	False Negative (FN)	Micro-Averaged Values
 Airplane	2	1	1	Precision = $\frac{6}{6+4} = \mathbf{0.60}$ Recall = $\frac{6}{6+4} = \mathbf{0.60}$ F1 Score = $\frac{6}{6 + \frac{1}{2}(4+4)} = \mathbf{0.60}$
 Boat	1	3	0	
 Car	3	0	3	
TOTAL	6	4	4	

These results mean that in multi-class classification cases where each observation has a **single label**, the **micro-F1**, **micro-precision**, **micro-recall**, and **accuracy** share the **same** value (i.e., **0.60** in this example).

And this explains why the classification report **only needs to display a single accuracy value** since micro-F1, micro-precision, and micro-recall also have the same value.

$$\mathbf{micro-F1} = \mathbf{accuracy} = \mathbf{micro-precision} = \mathbf{micro-recall}$$

Which average should you choose?

In general, if you are working with an imbalanced dataset where all classes are equally important, using the **macro** average would be a good choice as it treats all classes equally.

It means that for our example involving the classification of airplanes, boats, and cars, we would use the macro-F1 score.

If you have an imbalanced dataset but want to assign greater contribution to classes with more examples in the dataset, then the **weighted** average is preferred.

This is because, in weighted averaging, the contribution of each class to the F1 average is weighted by its size.

Suppose you have a balanced dataset and want an easily understandable metric for overall performance regardless of the class. In that case, you can go with accuracy, which is essentially our **micro** F1 score.

Task 1: Create one `y_true` list and `y_pred0`, `y_pred1`, `y_pred2`, `y_pred3`, `y_pred4`, according to the table below.

Indexes of a list	[0..49]	[50..99]	[100..1000]	[155..10099]
<code>y_true</code>	0	0	1	2
<code>y_pred0</code>	0	0	2	0
<code>y_pred1</code>	1	2	0	2
<code>y_pred2</code>	0	0	0	0
<code>y_pred3</code>	1	0	1	2
<code>y_pred4</code>	2	2	2	2

Task 2: Calculate accuracy, precision, recall and f-measure for every `y_pred_`. Paste the results to the table below. Analyse the results.

	Accuracy	Macro avg (precision)	Macro avg (recall)	Macro avg (F1)
<code>y_pred0</code>				
<code>y_pred1</code>				
<code>y_pred2</code>				
<code>y_pred3</code>				
<code>y_pred4</code>				