Metrics for Multi-class and Multi-label Classification

Maximilian Wenzel
Data Science Seminar Summer Term 2021

Motivation for Metrics in Machine Learning

- Classification: Categorize an instance/sample into a class or multiple classes
- General approach in machine learning:
  - Given: training data, test data,
  - Goal: Classifier predicts class(es) for given instance
  - Train classifier on training data
  - Evaluate classifier over test data
- How to determine the performance of the resulting classifier on the test data?
  - Count the number of correct and incorrect predictions
  - Summarize counts using evaluation metrics
- Finished?

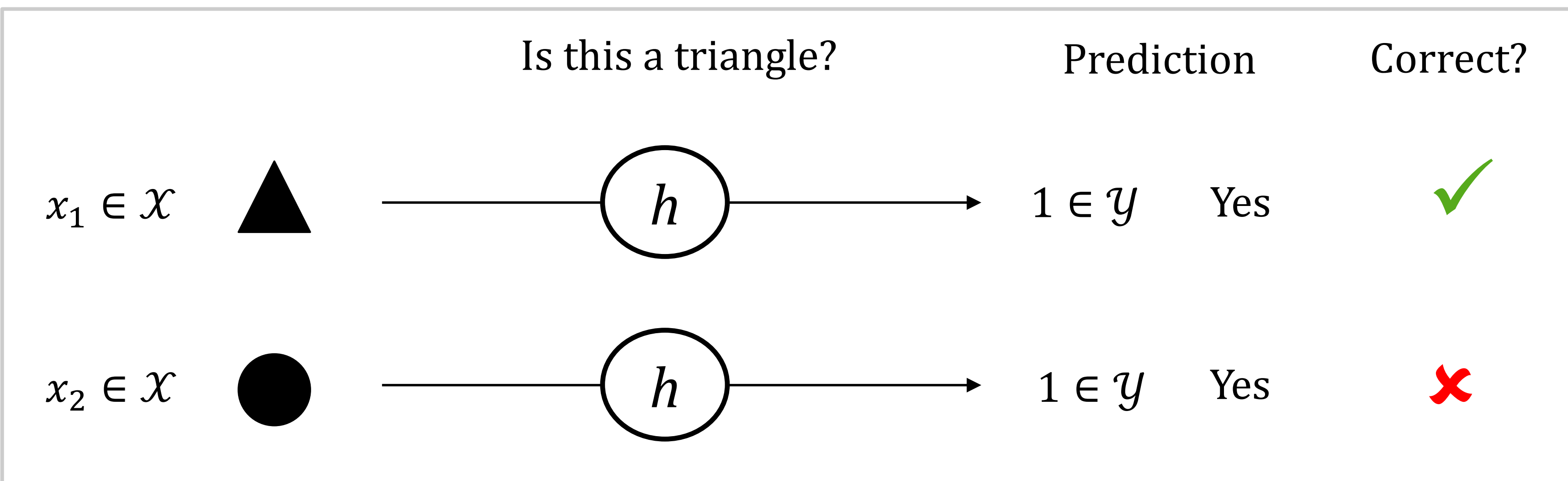
Problems with Evaluation Metrics

- Metrics usually not standardized for application domains
  - There exists no common consent on deployed metrics
- Small variations in metrics may even lead to different classifier rankings
- Number of possibilities to evaluate classifiers for multi-class and multi-label problems increases

➔ Exacerbates the problem!

Binary Classification

- Given:
  - Instance $x \in \mathcal{X} \subseteq \mathbb{R}^d$
  - Binary label space $\mathcal{Y} = \{0, 1\}$, "yes or no", "x or y"
  - Classifier $h: \mathcal{X} \to \mathcal{Y}$



Confusion Matrix

– Counts the number of correct and incorrect predictions of classifier h

		Actual Class: Cat	Actual Class: $Not\ Cat$	Predictions per Class
Predicted Class	Cat	9	2	11
Predicted Class	$Not\ Cat$	1	8	9
Instances per Class		10	10	20

Total number of instances in (test) dataset

– But what is the performance of our classifier?
  - Raw confusion matrix is difficult to interpret

➔ Use metrics to summarize absolute confusion matrix values

Fundamental Metrics

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

– **Recall**: Proportion of instances that have been correctly classified as positive

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

– **Precision**: Proportion of positive predictions that were actually correct

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

– $\boldsymbol{F_1}$**-score**: harmonic mean of **recall** and **precision**

$$F_1 = \frac{2 \cdot p \cdot r}{p + r} = \left(\frac{p^{-1} \cdot r^{-1}}{2}\right)^{-1}$$

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

Issue with Imbalanced Datasets

Balanced Dataset:
Equal amount of instances per class

		Actual Class: Cat	Actual Class: $Not\ Cat$	Predictions per Class
Predicted Class	Cat	9	2	11
	$Not\ Cat$	1	8	9
Instances per Class		10	10	20

$$p = \frac{9}{9+2} = \frac{9}{11} \approx 0.82$$

Imbalanced Dataset:
Different amount of instances per class

		Actual Class: Cat	Actual Class: $Not\ Cat$	Predictions per Class
Predicted Class	Cat	9	4	13
	$Not\ Cat$	1	16	17
Instances per Class		10	20	30

$$p = \frac{9}{9+4} = \frac{9}{13} \approx 0.69$$

- Metrics may be sensitive to imbalanced datasets
  - Although same proportion of $\frac{TP}{FN}$ and $\frac{FP}{TN}$ different results for metric
- Metrics which use values from both "actual class" columns are sensitive to imbalanced datasets

Multi-class Classification

- Given:
  - Instance $x \in \mathcal{X} \subseteq \mathbb{R}^d$
  - Label space $\mathcal{Y} \subseteq \{0,1\}^m$, one-hot-coded vectors
  - Classifier $h: \mathcal{X} \rightarrow \mathcal{Y}$, predicts exactly **one** class per instance

Example: Cat $y_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, Dog $y_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, Mouse $y_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

What kind of animal is this?

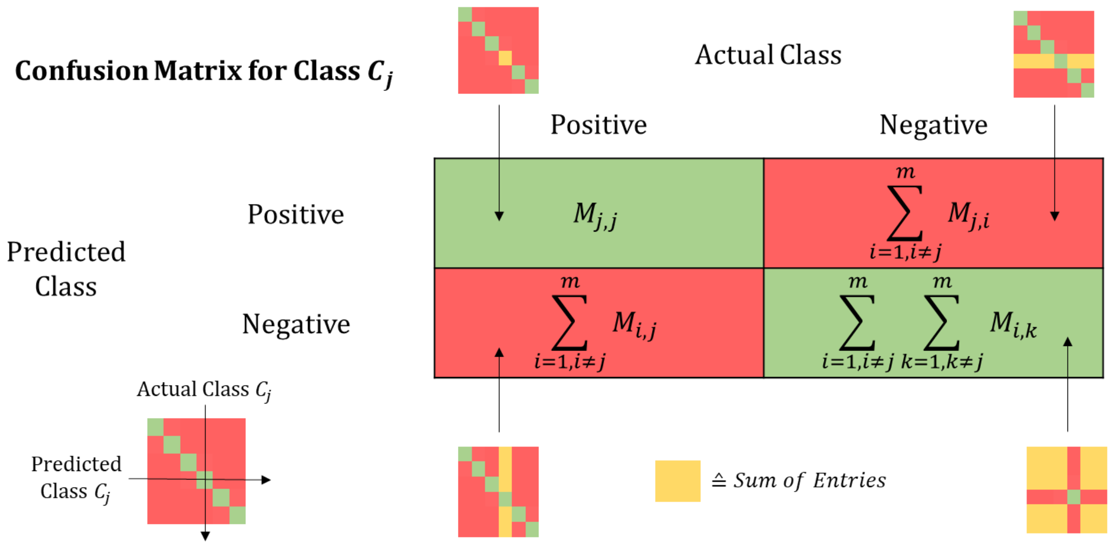
$x \in \mathcal{X}$  $\xrightarrow{\quad h \quad}$ Cat, $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \in \mathcal{Y}$

Multi-class Confusion Matrix

		Actual Class: Cat	Dog	$Mouse$	Predictions per Class
Predicted Class	Cat	9	3	1	13
	Dog	1	6	2	9
	$Mouse$	0	1	7	8
Instances per Class		10	10	10	30

- Confusion matrix becomes more complex: For m classes, $m \times m$ confusion matrix
- How to summarize now the performance of a given classifier?
- Solution:
  - Create for each class C_j a binary confusion matrix
  - Summarize all per-class results using an appropriate averaging strategy

Per-class Confusion Matrix



- Converts the problem into a binary classification problem
  - Class C_j and class "not C_j"
- Previously introduced metrics can thus be computed
- Problem: How to summarize the results for a given metric?

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Averaging Strategies

– **Macro Averaging**: Arithmetic mean of all per-class metrics

$$r_M = \frac{1}{m}\sum_{j=1}^{m}\frac{TP_j}{TP_j + FN_j}$$

All per-class results weighted equally

– **Micro Averaging**: Sum up numerator and denominator separately of the appropriate metric and compute the result

$$r_\mu = \frac{\sum_{j=1}^{m} TP_j}{\sum_{j=1}^{m} TP_j + FN_j}$$

Sensitive to imbalanced datasets

– **Weighted Averaging**: weight the per-class metrics by the number of instances of the appropriate class

$$r_w = \frac{1}{n}\sum_{j=1}^{m}\frac{n_j \cdot TP_j}{TP_j + FN_j}$$

Intentionally weighted by number of instances per class

Per-class Averaging Example

Summary of the per-class confusion matrices

Balanced Dataset

	TP	TN	FP	FN	Sum
Cat	9	16	4	1	30
Dog	6	17	3	4	30
$Mouse$	7	19	1	3	30
Sum	22	52	8	8	90

Imbalanced Dataset

	TP	TN	FP	FN	Sum
Cat	9	41	9	1	60
Dog	12	33	7	8	60
$Mouse$	21	28	2	9	60
Sum	42	102	18	18	180

Averaging Strategy	**Balanced Dataset**	**Imbalanced Dataset**
Macro-Precision p_M	$\frac{1}{3}\left(\frac{9}{9+4}+\frac{6}{6+3}+\frac{7}{7+1}\right) \approx 0.745$	$\frac{1}{3}\left(\frac{9}{9+9}+\frac{12}{12+7}+\frac{21}{21+2}\right) \approx 0.682$
Micro-Precision p_μ	$\frac{9+6+7}{9+4+6+3+7+1} = \frac{11}{15} \approx 0.73$	$\frac{9+12+21}{9+9+12+7+21+2} = 0.7$
Weighted-Precision p_w	$\frac{1}{30}\left(\frac{1}{10}\cdot\frac{9}{9+4}+\frac{1}{10}\cdot\frac{6}{6+3}+\frac{1}{10}\cdot\frac{7}{7+1}\right) \approx 0.745$	$\frac{1}{60}\left(10\cdot\frac{9}{9+9}+20\cdot\frac{12}{12+7}+30\cdot\frac{21}{21+2}\right) \approx 0.750$

Averaging the F_1-score

– Micro-averaged F_1 analogously to the standard approach:

$$F_{1_\mu} = \frac{2 \cdot p_\mu \cdot r_\mu}{p_\mu + r_\mu}$$

– Two distinct approaches to compute the macro-averaged F_1-score

- $\mathcal{F}_1$, the averaged F_1

$$\mathcal{F}_1 = \frac{1}{m} \sum_{j=1}^{m} \frac{2 \cdot p_j \cdot r_j}{p_j + r_j}$$

The standard approach, recommended by Opitz and Burst (2019)

- $\mathbb{F}_1$, the F_1 of averages

$$\mathbb{F}_1 = \frac{2 \cdot p_M \cdot r_M}{p_M + r_M}$$

Individual values p_j and r_j not as much influence
➔ May be overly benevolent

➔ Different strategies also applicable to the weighted-aproach

Averaging the F_1-score

Balanced Dataset

	TP	TN	FP	FN	Sum
Cat	9	16	4	1	30
Dog	6	17	3	4	30
$Mouse$	7	19	1	3	30
Sum	22	52	8	8	90

Imbalanced Dataset

	TP	TN	FP	FN	Sum
Cat	9	41	9	1	60
Dog	12	33	7	8	60
$Mouse$	21	28	2	9	60
Sum	42	102	18	18	180

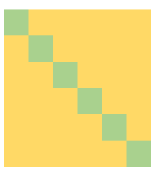
Averaging Strategy	Balanced Dataset	Imbalanced Dataset
$\mathcal{F}_1$, the Averaged F_1	$\mathcal{F}_1 = \frac{1}{m}\sum_{j=1}^{m}\frac{2 \cdot p_j \cdot r_j}{p_j + r_j} = 0.731$	$\mathcal{F}_1 = 0.684$
$\mathbb{F}_1$, the F_1 of Averages	$\mathbb{F}_1 = \frac{2 \cdot p_M \cdot r_M}{p_M + r_M} = 0.739$	$\mathbb{F}_1 = 0.706$
F_{1_μ}, Micro-Averaged F_1	$F_{1_\mu} = \frac{2 \cdot p_\mu \cdot r_\mu}{p_\mu + r_\mu} = \frac{11}{15} \approx 0.733$	$F_{1_\mu} = 0.7$

Micro-Precision, Micro-Recall, and Micro-F_1

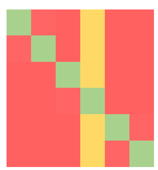
– We have:

$$p_\mu = \frac{\sum_{j=1}^{m} TP_j}{\sum_{j=1}^{m} TP_j + FP_j} = r_\mu = \frac{\sum_{j=1}^{m} TP_j}{\sum_{j=1}^{m} TP_j + FN_j} = F_{1_\mu} = \frac{2 \cdot p_\mu \cdot r_\mu}{p_\mu + r_\mu}$$

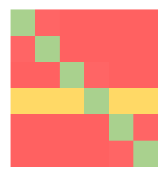
If $p_\mu = r_\mu$, this follows since F_{1_μ} computes harmonic mean



$\sum_{j=1}^{m}$



FP_j



FN_j

$$\sum_{j=1}^{m} FP_j = \sum_{j=1}^{m} FN_j$$

$\triangleq$ *Sum of Entries*

– Given: C_j is predicted class, C_k is actual class
  - From perspective of C_j: $false\ positive\ FP$
  - From perspective of C_k: $false\ negative\ FN$

➔ Each FP is a FN value depending on the viewpoint of the appropriate class

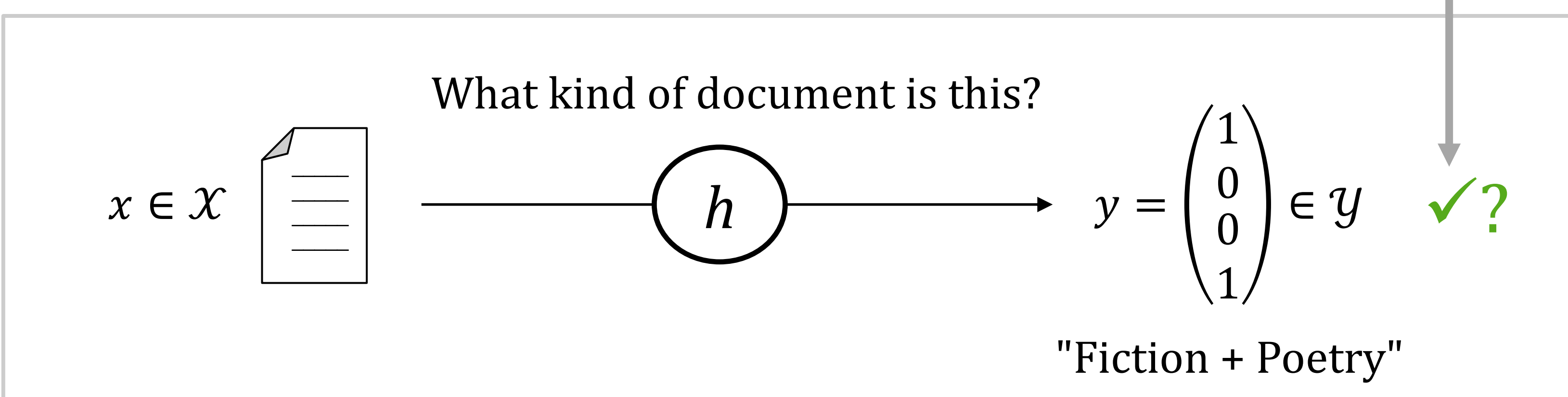
Multi-label Classification

- Given:
  - Instance $x \in \mathcal{X} \subseteq \mathbb{R}^d$
  - Label space $\mathcal{Y} \subseteq \{0,1\}^m$
  - Classifier $h: \mathcal{X} \to \mathcal{Y}$, may predict **multiple** classes/labels per instance

Example: Text classification

$$y = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \begin{array}{l} \leftarrow \text{Fiction} \\ \leftarrow \text{Non-fiction} \\ \leftarrow \text{Drama} \\ \leftarrow \text{Poetry} \end{array}$$

What if prediction is only partially correct?



Multi-label Classification: Viewpoint of Correctness

Evaluation and averaging strategies identical to multi-class case

Multi-label Correctness

Per-class

- Consider each class separately
- Look at all instances of test dataset:
  - How often has a given class been predicted correctly/incorrectly

➔ Derive **per-class confusion matrix**

Per-instance

- Consider each instance separately
- How many labels have been predicted correctly/incorrectly?

➔ Derive **per-instance confusion matrix**

Summarize confusion matrices with averaging strategies

Multi-label Classification: Per-instance evaluation

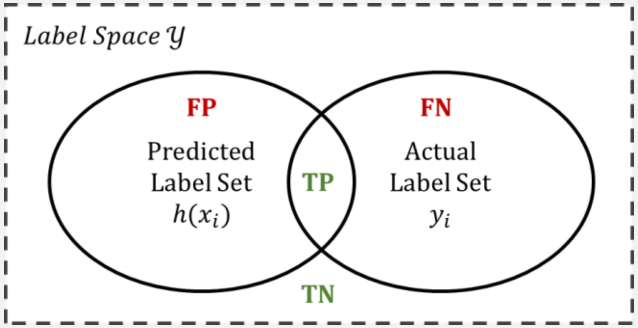
Actual Label Set $y = \begin{pmatrix}1\\0\\0\\1\end{pmatrix}$	Predicted Label Sets	
	Vector Notation	Set Notation
Fully Correct	$h(x) = \begin{pmatrix}1\\0\\0\\1\end{pmatrix}$	$C_{h(x)} = \{C_1, C_4\}$
Partially Correct	$h(x) = \begin{pmatrix}0\\1\\0\\1\end{pmatrix}$	$C_{h(x)} = \{C_2, C_4\}$

Too harsh, espcially if label space $\mathcal{Y}$ becomes large

Exact Match Ratio:

$$MR = \frac{\#fullyCorrect}{\#instances}$$

Per-instance confusion matrix:



Per-instance Evaluation: Which Averaging Strategies?

- Per-instance evaluation makes only sense with **macro averaging strategies → each instance is equally weighted**
  - Micro- and weighted-averaged result would weight instances differently
- **Example**: weighted-average
  - Each per-instance result is weighted by the factor $TP_j + FN_j$ per instance x_j

	Meaning	
	Per-class	**Per-instance**
$\boldsymbol{TP_j + FN_j}$	#instances per class C_j	#labels in actual label set y_j

Accuracy and Error Rate

	Accuracy	Error Rate
Binary	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$	$ERR = 1 - Acc$
Multi-class/Multi-label **Macro-averaged per Class** m: Number of Classes	$Acc_M = \frac{1}{m}\sum_{j=1}^{m}\frac{TP_j + TN_j}{TP_j + TN_j + FP_j + FN_j}$	$ERR_M = 1 - Acc_M$
Multi-class/Multi-label **Micro-averaged per Class**	$Acc_\mu = \frac{\sum_{j=1}^{m} TP_j + TN_j}{\sum_{j=1}^{m} TP_j + TN_j + FP_j + FN_j}$	$ERR_\mu = 1 - Acc_\mu$
Multi-label **Averaged per Instance** Only Macro Averaging Strategy	(Jaccard Similarity) $Acc_M = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i + FN_i}$	(Hamming Loss) $HL = \frac{1}{n}\sum_{i=1}^{n}\frac{FP_i + FN_i}{TP_i + TN_i + FP_i + FN_i}$

TN_i corresponds to labels which are not present in the actual label set C_y → usually large if $\mathcal{Y}$ also large, therefore left out

Corresponds to XOR operation between predicted label vector $h(x)$ and actual label vector y

Best Practice When Dealing with Metrics

BioASQ

- BioASQ: organizes challenges for biomedical semantic indexing and QA systems
  - MESINESP task (2020): implement classifier which assigns labels from the DeCS vocabulary to new medical documents
  - Participants used test dataset to evaluate their classifier
    ➔ Concrete predicted label sets for each instance were recorded as JSON file
  - MESINESP committee computed appropriate metrics centrally
    ➔ Ensures consistent usage of metrics
    - Per-class micro F_1-score
    - Lowest Common Ancestor (LCA) F_1-score

Best Practice When Dealing with Metrics – Paper Writing

- Always explicitly indicate which metric has been deployed
  - Include the metric as equation
  - If the metric has been implemented by a library (e.g. Python SciKit-learn), look up the concrete implementation
  - If possible include the test dataset evaluation
    ➔ Computation of metric can be reproduced

MESINESP task: Structure of the JSON file for test dataset evaluation

```
{ "documents": [
{ "id": "id_test_article_1", "labels": [ "code1", "code2", "code3" ] } ,
{ "id": "id_test_article_2", "labels": [ "code5", "code2", "code21" ] }
] }
```

Source: https://temu.bsc.es/mesinesp2/evaluation/

Conclusion

- Confusion matrix summarizes predictions of a classifier on test data
- Metrics summarize the values from a confusion matrix
- The confusion matrix can be computed...
  - per class ➔ multi-class/multi-label case
  - per instance ➔ only multi-label case
- Averaging strategies: Summary of all per-class/per-instance metrics
  - macro, micro, weighted averaging
- To ensure reproducibility and prevent misconceptions:
  - Always include metric as concrete equation
- Always reflect if the deployed metric makes sense in an application domain

Multi-label Example

$Label\ Space\ \mathcal{Y} = \{a, b, c, d, e, f, g\}$

Instance	*Predicted Label Set*	*Actual Label Set*	*TP*	*TN*	*FP*	*FN*	*Sum*
x_1	$\{a, b, c\}$	$\{a, b, c\}$	3	4	0	0	7
x_2	$\{a, b, d, e\}$	$\{a, b, c, d, e\}$	4	2	0	1	7
x_3	$\{e, f\}$	$\{c, d\}$	0	3	2	2	7
x_4	$\{b, c, d\}$	$\{a, c, d, g\}$	2	2	1	2	7
x_5	$\{a, c, d, f, g\}$	$\{g\}$	1	2	4	0	7
		Sum	10	13	7	5	35

Metric Summary

	Recall	Precision	F_1-score	Accuracy	Error Rate
Binary	$r = \frac{TP}{TP + FN}$	$p = \frac{TP}{TP + FP}$	$F_1 = \frac{2 \cdot p \cdot r}{p + r}$	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$	$ERR = 1 - Acc$
Multi-class/Multi-label Macro-averaged per Class m: Number of Classes	$r_M = \frac{1}{m}\sum_{j=1}^{m}\frac{TP_j}{TP_j + FN_j}$	$p_M = \frac{1}{m}\sum_{j=1}^{m}\frac{TP_j}{TP_j + FP_j}$	$\mathcal{F}_1 = \frac{1}{m}\sum_{j=1}^{m}\frac{2 \cdot p_j \cdot r_j}{p_j + r_j}$ $\mathbb{F}_1 = \frac{2 \cdot p_M \cdot r_M}{p_M + r_M}$	$Acc_M = \frac{1}{m}\sum_{j=1}^{m}\frac{TP_j + TN_j}{TP_j + TN_j + FP_j + FN_j}$	$ERR_M = 1 - Acc_M$
Multi-class/Multi-label Micro-averaged per Class	$r_\mu = \frac{\sum_{j=1}^{m} TP_j}{\sum_{j=1}^{m}(TP_j + FN_j)}$	$p_\mu = \frac{\sum_{j=1}^{m} TP_j}{\sum_{j=1}^{m}(TP_j + FP_j)}$	$F_{1\mu} = \frac{2 \cdot p_\mu \cdot r_\mu}{p_\mu + r_\mu}$	$Acc_\mu = \frac{\sum_{j=1}^{m} TP_j + TN_j}{\sum_{j=1}^{m}(TP_j + TN_j + FP_j + FN_j)}$	$ERR_\mu = 1 - Acc_\mu$
Multi-class/Multi-label Weighted-averaged per Class n: # Instances in Dataset n_j: # Instances in Class C_j	$r_w = \frac{1}{n}\sum_{j=1}^{m}\frac{n_j \cdot TP_j}{TP_j + FN_j}$	$p_w = \frac{1}{n}\sum_{j=1}^{m}\frac{n_j \cdot TP_j}{TP_j + FP_j}$	$\mathcal{F}_1 = \frac{1}{n}\sum_{j=1}^{m}\frac{n_j \cdot 2 \cdot p_j \cdot r_j}{p_j + r_j}$ $\mathbb{F}_1 = \frac{2 \cdot p_w \cdot r_w}{p_w + r_w}$	Not used in literature	Not used in literature
Multi-label Averaged per Instance Only Macro Averaging Strategy	$r_M = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i}$	$p_M = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i}$	$\mathcal{F}_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \cdot p_i \cdot r_i}{p_i + r_i}$ $\mathbb{F}_1 = \frac{2 \cdot p_M \cdot r_M}{p_M + r_M}$	(Jaccard Similarity) $Acc_M = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i + FN_i}$	(Hamming Loss) $HL = \frac{1}{n}\sum_{i=1}^{n}\frac{FP_i + FN_i}{TP_i + TN_i + FP_i + FN_i}$