Metrics.

Iván Alejandro García Amaya

1 Multiclass and multilabel classification

In multiclass and multilabel classification task, the notions of precision, recall, and F-measures can be applied to each label independently. There are a few ways to combine results across labels, specified by the average argument to the average_ precision_score (multilabel only), f1_ score, fβ_ score, precision_ recall_ fscore_ support, precision_ score and recall_ score functions, as described above. Note that if all labels are included, "micro"-averaging in a multiclass setting will produce precision, recall and that are all identical to accuracy. Also note that "weighted" averaging may produce an F-score that is not between precision and recall. [2]

To make this more explicit, consider the following notation:

y the set of predicted $(sample, label)$ pairs

$\hat{y}$ the set fo true $(sample, label)$ pairs

L the set of labels

S the set of samples

y_s the subset of y with sample s, i.e. $y_s := \{(s^{'}, l) \in y | s^{'} = s\}$

y_l the subset of y with label l

similary, $\hat{y}s$ and $\hat{y}l$ are subsets of $\hat{y}$

$P(A, B) := \frac{|A \cap B|}{|A|}$ for some sets A nad B

$R(A, B) := \frac{|A \cap B|}{|B|}$ (Convections vary on handling $B = \emptyset$; this implementation uses) $R(A, B) := 0$, and similar for P.)

$f_\beta(A, B) := (1 + \beta^2) \frac{P(A,B) x R(A,B)}{\beta^2 P(A,B) + R(A,B)}$

1.1 Precision weighted.

$$\frac{1}{\sum_{l \in L} \hat{y}l} \sum_{l \in L} |\hat{yl}| P(yl, \hat{y}l) \tag{1}$$

$$WeightedAveragePrecision = \frac{\sum_{i=1}^{n} |y_i| \frac{Tp_i}{Tp_i + Fp_i}}{\sum_{i=1}^{n} |y_i|} \tag{2}$$

$$WeightedAveragePrecision = \frac{|y_1|}{|y|} * Precision_1 + \frac{|y_2|}{|y|} * Precision_2 \tag{3}$$

$$Precision = \frac{Tp}{Tp + Fp} \tag{4}$$

1.1.1 Example Precision.

$y\_true = [0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0]$
$y\_pred = [1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0]$

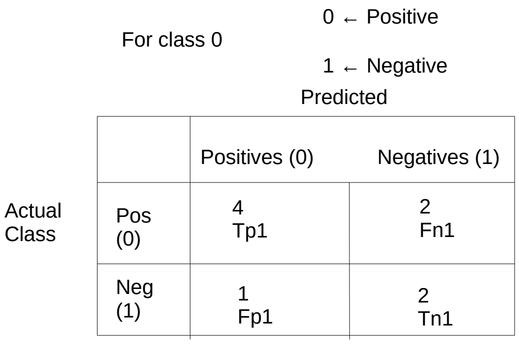
y $= |y| = 9$

$|y_1| = 6$

$|y_2| = 3$

$Precision_1 = \frac{Tp_1}{Tp_1 + Fp_1} = \frac{4}{4+1} = 0.8$

$Precision_2 = \frac{Tp_2}{Tp_2 + Fp_2} = \frac{2}{2+2} = 0.5$



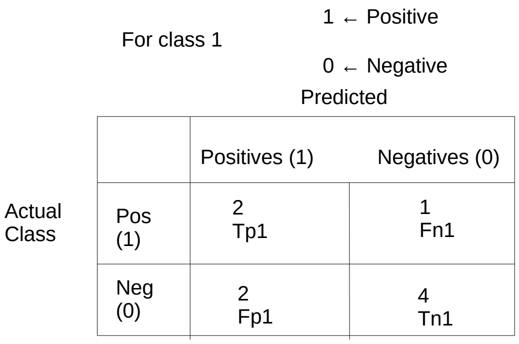
For class 0

0 ← Positive

1 ← Negative

		Predicted	
		Positives (0)	Negatives (1)
Actual Class	Pos (0)	4 Tp1	2 Fn1
	Neg (1)	1 Fp1	2 Tn1

Figure 1: Confusion matrix 1.



For class 1

1 ← Positive

0 ← Negative

		Predicted	
		Positives (1)	Negatives (0)
Actual Class	Pos (1)	2 Tp1	1 Fn1
	Neg (0)	2 Fp1	4 Tn1

Figure 2: Confusion matrix 2.

$$WeightedAveragePrecision = \frac{6}{9} * 0.8 + \frac{3}{9} * 0.5 \tag{5}$$

1.2 Recall weighted.

(Sensitivity)

$$\frac{1}{\sum_{l\in L}\hat{y}l}\sum_{l\in L}|\hat{yl}|R(yl,\hat{y}l) \tag{6}$$

$$WeightedAverageRecall = \frac{\sum_{i=1}^{n}|y_i|\frac{Tp_i}{Tp_i+Fn_i}}{\sum_{i=1}^{n}|y_i|} \tag{7}$$

$$WeightedAverageRecall = \frac{|y_1|}{|y|} * Recall_1 + \frac{|y_2|}{|y|} * Recall_2 \tag{8}$$

$$Recall = \frac{Tp}{Tp+Fn} \tag{9}$$

1.2.1 Example Recall

$Recall_1 = \frac{Tp_1}{Tp_1+Fn_1} = \frac{4}{6} = 0.66$

$Recall_2 = \frac{Tp_2}{Tp_2+Fn_2} = \frac{2}{3} = 0.66$

$$WeightedAverageRecall = \frac{6}{9} * 0.66 + \frac{6}{9} * 0.66 \tag{10}$$

1.3 F_beta.

$$\frac{1}{\sum_{l\in L}\hat{y}l}\sum_{l\in L}|\hat{yl}|F_\beta(yl,\hat{y}l) \tag{11}$$

$$WeightedAveragef1score = \frac{\sum_{i=1}^{n}|y_i|f1score}{|y_i|} \tag{12}$$

$$WeightedAveragef1score = \frac{|y_1|f1score_1}{|y|} + \frac{|y_2|f1score_2}{|y|} \tag{13}$$

$$f1score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{14}$$

1.3.1 Example F_beta.

$$f1score_1 = \frac{2 * Precision_1 * Recall_1}{Precision_1 + Recall_1} = \frac{2 * 0.8 * 0.66}{0.8 + 0.66} \tag{15}$$

$$f1score_2 = \frac{2 * Precision_2 * Recall_2}{Precision_2 + Recall_2} = \frac{2 * 0.5 * 0.66}{0.5 + 0.66} \tag{16}$$

1.4 Accuracy.

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \tag{17}$$

[4]

$$accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{18}$$

[5]

1.5 Specificity.

$$Specificity = \frac{Tn}{Tn + Fp} \tag{19}$$

2 Code.

2.1 main code.

```
import pandas as pd
from weightedmetrics import *
class_names = ['No␣Finding', 'Enlarged␣Cardiomediastinum',
                'Cardiomegaly', 'Lung␣Opacity', 'Lung␣Lesion',
                'Edema', 'Consolidation', 'Pneumonia',
        'Atelectasis', 'Pneumothorax', 'Pleural␣Effusion',
        'Pleural␣Other', 'Fracture', 'Support␣Devices',
        'Hernia', 'Mass', 'Fibrosis', 'Infiltration',
        'Nodule', 'Emphysema', 'Pleural_Thickening']

filename = './gtfile.csv'
filename2 = './predfile.csv'
df = pd.read_csv(filename)
df2 = pd.read_csv(filename2)
gt = df.to_numpy()
pred = df2.to_numpy()
for i, (name) in enumerate(class_names):
    print(name)
    weighted_average_precision = weighted_precision(gt[:, i], pred[:, i])
    weighted_average_recall = weighted_recall(gt[:, i], pred[:, i])
    f1score = weighted_f1score(gt[:, i], pred[:, i])
    accuracy_1 = accuracy_sc(gt[:, i], pred[:, i])
    weighted_average_specificity = \
        weighted_specificity(gt[:, i], pred[:, i])

    print('weighted_average_precision␣:␣', weighted_average_precision)
    print('weighted_average_recall␣:', weighted_average_recall)
    print('f1score␣:', f1score)
    print('accuracy', accuracy_1)
    print('weighted_average_specificity', weighted_average_specificity)
```

2.2 weighted metrics.

```
from sklearn.metrics import confusion_matrix
import numpy as np
```

```
def weighted_precision(gt, pred):
    # for class 1
    tn, fp, fn, tp = confusion_matrix(gt, pred).ravel()
    precision = tp/(tp + fp)
    y = len(gt)
    y1 = np.count_nonzero(gt == 1)
    w1 = (y1/y)
    weighted_precision_1 = w1 * precision

    # for class 0
    tp, fn, fp, tn = confusion_matrix(gt, pred).ravel()
    precision_2 = tp / (tp + fp)
    y1_2 = np.count_nonzero(gt == 0)
    w2 = (y1_2 / y)
    weighted_precision_2 = w2 * precision_2

    weighted_average_precision = weighted_precision_1 + \
                weighted_precision_2

    return weighted_average_precision


def weighted_recall(gt, pred):
    # for class 1
    tn, fp, fn, tp = confusion_matrix(gt, pred).ravel()
    recall = tp/(tp + fn)
    y = len(gt)
    y1 = np.count_nonzero(gt == 1)
    w1 = (y1/y)
    weighted_recall_1 = w1 * recall

    # for class 0
    tp, fn, fp, tn = confusion_matrix(gt, pred).ravel()
    recall_2 = tp/(tp + fn)
    y1_2 = np.count_nonzero(gt == 0)
    w2 = (y1_2 / y)
    weighted_recall_2 = w2 * recall_2

    weighted_average_recall = weighted_recall_1 + weighted_recall_2
    return weighted_average_recall


def weighted_f1score(gt, pred):
    weighted_average_precision = weighted_precision(gt, pred)
    weighted_average_recall = weighted_recall(gt, pred)
    f1score = (2*weighted_average_precision * weighted_average_recall)\
        / (weighted_average_precision + weighted_average_recall)

    return f1score

def accuracy_sc(gt, pred):
    tn, fp, fn, tp = confusion_matrix(gt, pred).ravel()
    accuracy_1 = (tp + tn)/(tp + tn + fp + fn)
    return accuracy_1


def weighted_specificity(gt, pred):
    # for class 1
    tn, fp, fn, tp = confusion_matrix(gt, pred).ravel()
    specificity_1 = tn/(tn + fp)
    y = len(gt)
    y1 = np.count_nonzero(gt == 1)
    w1 = (y1/y)
    weighted_specificity_1 = w1 * specificity_1

    # for class 0
    tp, fn, fp, tn = confusion_matrix(gt, pred).ravel()
    specificity_2 = tn/(tn + fp)
    y1_2 = np.count_nonzero(gt == 0)
    w2 = (y1_2 / y)
    weighted_specificity_2 = w2 * specificity_2

    weighted_average_specificity = weighted_specificity_1 +\
     weighted_specificity_2
    return weighted_average_specificity
```

3 Results.

Results obtained using the codes described in the previous section.

Medical condition	Accuracy	f1score	Weighted Precision	Weighted Recall	Weighted Specificity
No Finding	0.74	0.77	0.81	0.74	0.81
Enlarged Cardiomediastinum	0.70	0.76	0.84	0.70	0.86
Cardiomegaly	0.83	0.85	0.86	0.83	0.72
Lung Opacity	0.89	0.90	0.91	0.89	0.93
Lung Lesion	0.66	0.79	0.99	0.66	0.0015
Edema	0.81	0.87	0.93	0.81	0.97
Consolidation	0.56	0.68	0.86	0.56	0.73
Pneumonia	0.49	0.66	0.97	0.49	0.88
Atelectasis	0.68	0.74	0.81	0.68	0.78
Pneumothorax	0.76	0.85	0.95	0.76	0.73
Pleural Effusion	0.83	0.85	0.87	0.83	0.80
Pleural Other	0.75	0.85	0.99	0.75	0.99
Fracture	0.67	nan	1.0	nan	nan
Support Devices	0.86	0.87	0.88	0.86	0.86
Hernia	0.88	0.93	0.98	0.88	0.0061
Mass	0.55	0.70	0.97	0.55	0.86
Fibrosis	0.55	0.71	0.99	0.55	0.99
Infiltration	0.56	0.71	0.96	0.56	0.98
Nodule	0.56	0.71	0.97	0.56	0.98
Emphysema	0.58	0.73	0.98	0.58	0.99
Pleural thickening	0.54	0.70	0.99	0.54	0.99

Table 1: Metrics results.

Results obtained using the Sklearn library.

Medical condition	Accuracy	f1score	Weighted Precision	Weighted Recall	Weighted Specificity
No Finding	0.74	0.74	0.81	0.74	0.81
Enlarged Cardiomediastinum	0.70	0.72	0.84	0.70	0.86
Cardiomegaly	0.83	0.84	0.86	0.83	0.72
Lung Opacity	0.89	0.89	0.91	0.89	0.93
Lung Lesion	0.66	0.79	0.99	0.66	0.0015
Edema	0.81	0.84	0.93	0.81	0.97
Consolidation	0.56	0.64	0.86	0.56	0.73
Pneumonia	0.49	0.64	0.97	0.49	0.88
Atelectasis	0.68	0.71	0.81	0.68	0.78
Pneumothorax	0.76	0.84	0.95	0.76	0.73
Pleural Effusion	0.83	0.84	0.87	0.83	0.80
Pleural Other	0.75	0.85	0.99	0.75	0.99
Fracture	0.67	0.80	1.0	0.67	0.0
Support Devices	0.86	0.87	0.88	0.86	0.86
Hernia	0.88	0.93	0.98	0.88	0.0061
Mass	0.55	0.69	0.97	0.55	0.86
Fibrosis	0.55	0.70	0.99	0.55	0.99
Infiltration	0.56	0.69	0.96	0.56	0.98
Nodule	0.56	0.69	0.97	0.56	0.98
Emphysema	0.58	0.72	0.98	0.58	0.99
Pleural thickening	0.54	0.69	0.99	0.54	0.99

Table 2: Metrics results sklearn.

4 Reference.

[1] https://datascience.stackexchange.com/questions/40900/whats-the-difference-between-sklearn-f1-score-micro-and-weighted-for-a-mult

[2] https://scikit-learn.org/stable/modules/model_ evaluation.html# precision-recall-f-measure-metrics

[3] https://www.youtube.com/watch?v=5ySAKEzZTZA

[4] https://scikit-learn.org/stable/modules/model_evaluation.html# accuracy-score

[5] https://developers.google.com/machine-learning/crash-course/classification/accuracy

[6] https://academic.oup.com/bjaed/article/8/6/221/406440