

بسمه تعالی

گزارش عملکرد مند multi_metrics()

مهدی جواهری صابر
۹۲۴۳۰۸۸۰۱۷

```
from nltk.metrics import masi_distance, precision, recall
```

برای ارزیابی مدل های Mutilabel از معیارهای Precision و Recall میتوانیم استفاده کنیم ولی نمیتوانیم از Accuracy استفاده کنیم ، زیرا اگر به عنوان مثال یک Feats با سه لیبل داشته باشیم و دوتا از لیبل ها صحیح و سومی نادرست باشد معیار accuracy برای این مدل لیبل نادرست را به آن ست میکند ، برای همین منظور از معیار masi_distance استفاده میکنیم masi_distance هر چقدر به صفر نزدیکتر باشد مدل ما دقیق تر است

```
def multi_metrics(multi_classifier, test_feats):
```

ورودی های متد داده های تست و classifier به ازای هر کلاس از لیبل ها میباشد

```
mds = []
```

از این متغیر برای ذخیره ی masi_distance هر لیبل استفاده میکنیم که در انتها مقدار میانگین آن را برمیگردانیم

```
refsets = collections.defaultdict(set)
```

این متغیر یک دیکشنری از خروجی واقعی سیستم میباشد و در صورتی که کلیدی از آن مقدار نگرفته باشد ، مقدار پیشفرض آن یک مجموعه با set خالی است

```
testsets = collections.defaultdict(set)
```

از این متغیر برای ذخیره ی داده های تست که در قالب یک دیکشنری از set هاست استفاده میکنیم. در صورتی که کلیدی از این دیکشنری مقدار نگرفته باشد ، مقدار پیشفرض یک set خالی است

```
for i, (feat, labels) in enumerate(test_feats):
```

این حلقه روی داده های تست حرکت میکند و ا نشانگر ایندکس لیست داده های تست ، feat دیکشنری از کلمات موجود (که به صورت پیشفرض bag_of_words است) و labels لیستی از لیبل های آن feat مذکور است

```
for label in labels:  
    refsets[label].add(i)
```

به ازای تمامی لیبل ها ، مقدار مربوط به هر لیبل را به اندازه ی اینکس فعلی افزایش میدهد

```
guessed = multi_classifier.classify(feat)
```

مدل ساخته شده از آن feat مد نظر (از این مدل برای ارزیابی استفاده میشود)

```
for label in guessed:  
    testsets[label].add(i)
```

این حلقه تعداد هر لیبل در داده های تست را به دیکشنری testsets اضافه میکند

```
mds.append(masi_distance(set(labels), guessed))
```

معیار masi را برای مدل ساخته شده از لیبل های مجموعه فعلی در این حلقه محاسبه میکند

```
avg_md = sum(mds) / len(mds)
```

بعد از اتمام حلقه میانگین معیار masi را محاسبه میکنیم (زیرا به ازای هر مدل یک معیار وجود دارد)

```
precisions = {}  
recalls = {}
```

از این دو دیکشنری برای نگه داری precision و recal هر لیبل استفاده میکنیم

```
for label in multi_classifier.labels():
```

به ازای هر لیبل داخل classifier محاسبه میکند

```
precisions[label] = precision(refsets[label], testsets[label])
```

precision هر لیبل را بر اساس کلمات تست و لیبل های مربوط به آن محاسبه میکند

```
recalls[label] = recall(refsets[label], testsets[label])
```

recall هر لیبل را بر اساس کلمات تست و لیبل های مربوط به آن محاسبه میکند

```
return precisions, recalls, avg_md
```

در انتها دو دیکشنری precisions و recalls که هر کدام به ازای هر لیبل یک مدخل دارند ، و

avg_md = average_masi_distance که میانگین تمامی msd لیبل ها را محاسبه کرده برمیگرداند