# 标准参数：

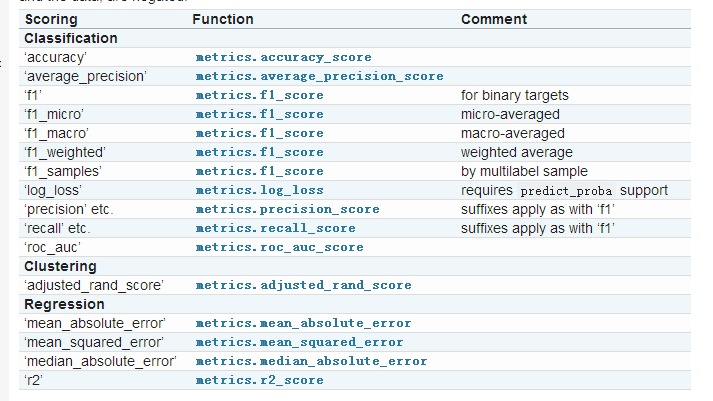
**multilabel case ：**hamming\_loss(np.array([[0, 1], [1, 1]]), np.zeros((2, 2)))>>0.75

**Normalize**: True(百分比)，False(数目)

**average\_precision**:precision\_recall\_curve曲线下方的面积

****probas\_pred****:就是y\_score,分类器对每个样本给出的评分，在多分类问题中，会给出每个类的评分[n\_samples, n\_classes]

**Scoring**:计分函数



**Multioutput**:回归问题中，多维样本回归，每一维都有预测输出。String or array-like of shape(n\_outputs)定义每一维的权重

**Multilable**:一个样本有多个标签

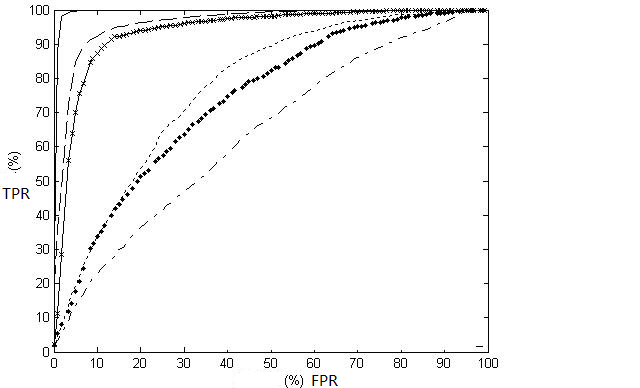
# 自定义score:

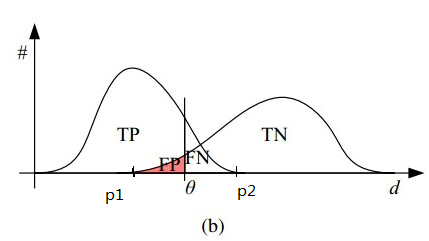
sklearn.metrics.make\_scorer:返回可供estimator调用的scorer

* score\_func,
* greater\_is\_better=True:score类越大越好，error类越小越好（False）
* needs\_proba=False,
* needs\_threshold=False:for classification metrics only: whether the python function you provided requires continuous decision certainties (needs\_threshold=True).For example average\_precision or the area under the roc curve can not be computed using discrete predictions alone.
* \*\*kwargs:score\_func中需要的参数

# 分类问题的评价函数

## ROC曲线

[](http://photo.blog.sina.com.cn/showpic.html#blogid=493b40e10100jps5&url=http://s13.sinaimg.cn/orignal/493b40e1t8b0049b73f9c)



TPR(recall)=TP/(TP+FN)，正例中多少被分对（正例的全面性）；

FPR=FP/(FP+TN)，负例中有多少被错分

ROC曲线可以用于评价一个分类器，ROC曲线上的每一个点对应于一个threshold，对于一个分类器，每个threshold下会有一个TPR（纵坐标）和FPR（横坐标）：

* 随着阈值theta增加，TP和FP都增加；TP增速减慢，FP增速加快。
* 反映到ROC中TPR,FPR正相关，且有一个中间的凸点值。
* AUC(area under curve):处于ROC curve下方的那部分面积的大小。通常，AUC的值介于0.5到1.0之间，较大的AUC代表了较好的Performance
* 若正负例是图b的交叉方式：

1. 0<<p1,FPR=0，TPR从0递增
2. p1<<p2,先TPR增长快，后FPR增速快
3. >p2,TPR=1,FPR从某个值开始递增到1

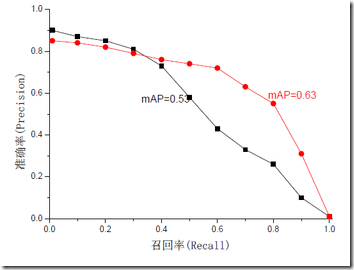
## Precision\_recall\_curve:

，预测正例的可信度,预测正例总数

，预测正例的全面度，实际正例数目

IMG_256

统计学概念，假阳性（FP）和假阴性（FN）不可能同时减小，一个增大，另一个就会减小。这意味着，precision和recall是两个相互权衡的概念。预测正例全面了，混入的负例（FP）就越多;预测正例的可信度提高了，漏掉的正例（FN）就增多了.



如果一个系统的性能较好，

1. 其曲线应当尽可能的向上突出。
2. 曲线与坐标轴之间的面积应当越大。
3. 最理想的系统， 其包含的面积应当是1，而所有系统的包含的面积都应当大于0。
4. 面积，平均准确率mAP(mean Average Precision)定义如下:(其中P，R分别为准确率与召回率)，sklearn中average\_precision就是这个

IMG_256

Example:

from sklearn.metrics import average\_precision\_score

y\_true = np.array([0, 0, 1, 1])

y\_scores = np.array([0.1, 0.4, 0.35, 0.8])

verage\_precision\_score(y\_true, y\_scores) 0.79...

默认第0类是正类，计算各个阈值区间，precision的积分：

## F1

* F1分数（F1 Score），是统计学中用来衡量二分类模型精确度的一种指标。它同时兼顾了分类模型的[准确率](http://baike.baidu.com/view/1799098.htm" \t "http://baike.baidu.com/_blank)和[召回率](http://baike.baidu.com/view/29947.htm" \t "http://baike.baidu.com/_blank)。
* F1分数可以看作是模型[准确率](http://baike.baidu.com/view/1799098.htm" \t "http://baike.baidu.com/_blank)和[召回率](http://baike.baidu.com/view/29947.htm" \t "http://baike.baidu.com/_blank)的一种加权平均，它的最大值是1，最小值是0。
* F1分数认为召回率和准确率同等重要， F2分数认为召回率的重要程度是准确率的2倍，而 F0.5分数认为召回率的重要程度是准确率的一半。

TPR(precision)=TP/(TP+FP):假阳性越少，值越大

Recall = TP/(TP+FN)：假阴性越少，值越大

4610b912c8fcc3ceb409adb89045d688d53f206b

c2cec3fdfc039245afef86848594a4c27c1e25e6

API:

sklearn.metrics.f1\_score(y\_true,y\_pred,labels=None,pos\_label=1,average='binary',sample\_weight=None)

* pos\_label: binary avarage中指定哪个类是正类(只有两分类才有正类)
* average :除了’binary’，其他都是针对muticalss。If None, 输出每个类的score

1. 'binary':Only report results for the class specified by pos\_label. This is applicable only if targets (y\_{true,pred}) are binary.
2. 'micro':Calculate metrics globally by counting the total true positives, false negatives and false positives.



1. 'macro':



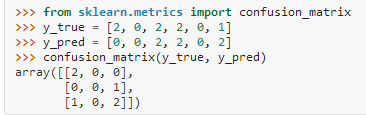
1. 'weighted':针对’macro’：考虑各类样本不均衡情况，为每类样本加权平均
2. 'samples':改用accuracy\_score计算样本的分类准确率

* sample\_weight : 每个样本的权重。array-like of shape = [n\_samples],

**variation:**

**fbeta\_score**(*y\_true*, *y\_pred*, *beta*, *labels=None*, *pos\_label=1*, *average='binary'*,*sample\_weight=None*)

## 混淆矩阵（confusion\_matrix）：



## 逻辑回归的损失函数或交叉熵损失（Log\_loss: negative log-likelihood）

逻辑回归借助sigmod函数，最终输出的是一个条件概率值：

Log\_loss评估的是这个条件概率的优劣。

* 二分类，一般=1，表示属于类1，反之表示属于类2



* 多分类，表示实际属于某一类的实际概率，通常情况属于某一个类概率为1，其他为0：



**log\_loss**(*y\_true*, *y\_pred*, *eps=1e-15*, *normalize=True*, *sample\_weight=None*)

* *y\_true*, *y\_pred*,
* *eps=1e-15，防止ln0出现*
* *normalize=True*, 如果是False,返回每个样本的损失函数之和
* *sample\_weight=None*

## 准确率（accuracy\_score）

**accuracy\_score**(*y\_true*, *y\_pred*, *normalize=True*, *sample\_weight=None*)

Normalize:分类正确的数目（False）或者比例（TRue）

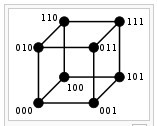
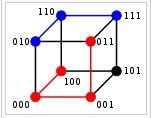
## 对每一类输出结果(Classification\_report):

classification\_report(y\_true, y\_pred, labels=None, target\_names=None, sample\_weight=None, digits=2)

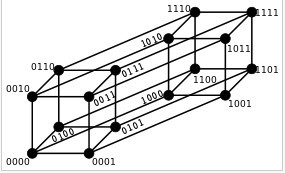
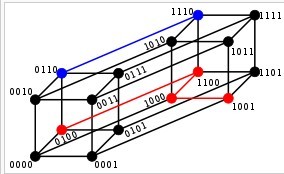
* target\_names:从0类开始，每一类的名称（选择哪些类输出）

# 海明距离（hamming\_loss）：

在信息论中，两个等长字符串之间的海明距离是两个字符串对应位置的不同字符的个数。换句话说，它就是将一个字符串变换成另外一个字符串所需要替换的字符个数。

三bit位海明距离立方体  
[](http://photo.blog.sina.com.cn/showpic.html#blogid=663457ed01017bry&url=http://s4.sinaimg.cn/orignal/663457edtd18695c18603)[](http://photo.blog.sina.com.cn/showpic.html#blogid=663457ed01017bry&url=http://s14.sinaimg.cn/orignal/663457edt7b5a422e772d)

100->011 海明距离为3， 010->111海明距离为2  
4bit位的超正方体模拟的海明距离

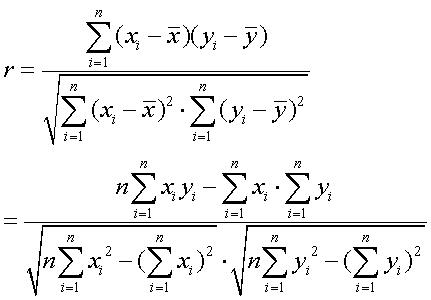
[](http://photo.blog.sina.com.cn/showpic.html#blogid=663457ed01017bry&url=http://s2.sinaimg.cn/orignal/663457edtd18695dc6931)

0110-》1110 的海明距离为1，0100-》1001的海明距离为3.

在分类问题中：海明距离是y\_true和y\_pred之间不相同的个数，占总数的比例

IMG_256

## 皮尔森相关系数（pearson correlation coefficient）：判断两个变量的线性相关性



相比较相关系数，多了一步零均值化步骤：

IMG_256

# 二、回归问题的损失函数

## 绝对值平均损失Mean\_absolute\_error

IMG_256

参数类似均方损失

## 均方损失MSE(mean\_squared\_error)

IMG_256

mean\_squared\_error(y\_true, y\_pred, sample\_weight=None, multioutput='uniform\_average')

* y\_true, y\_pred,
* sample\_weight=None:shape=(n\_samples)
* Multioutput=’Uniform\_avarage,针对多维输出的回归问题。[‘raw\_values’（所有维的误差）, ‘uniform\_average（所有维的误差平均）’]或者array-like of shape(n\_outputs)（所有维误差的加权平均）

## 绝对值的中位数损失（median absolute error）

IMG_256

R2(r2\_score):和平均预测结果相比,回归的准确率。等于0表示预测结果和平均数回归差不多。越接近1表示回归结果越好（y是真实值），也可能是负数。

IMG_256

r2\_score(y\_true, y\_pred, sample\_weight=None, multioutput=None)

* *y\_true*, *y\_pred*, *sample\_weight=None*,
* *multioutput=None,*[‘raw\_values’（所有维的误差）, ‘uniform\_average（所有维的误差平均）’]或者array-like of shape(n\_outputs)（所有维误差的加权平均）

1. <http://www.cnblogs.com/heaad/archive/2011/03/08/1977733.html> 机器学习中各种距离
2. <http://www.xuebuyuan.com/1208150.html> Recall, Precision, and Average Precision 信息检索（IR）的评价指标介绍