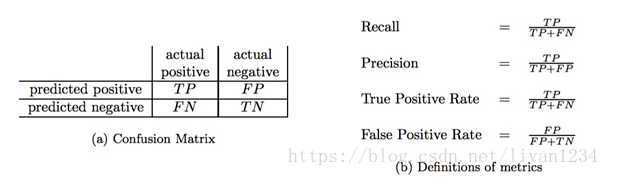
**分类模型评估**

从线性回归到逻辑回归  
线性回归解决的是连续型数值的预测问题，例如预测房价，产品销量等。  
逻辑回归解决的是分类问题，从分类数量上看，有二项分类和多项分类。

sklearn库的metrics模块提供各种评估方法，包括分类评估、回归评估、聚类评估和交叉验证等，评估分类是判断预测值时否很好的与实际标记值相匹配。正确的鉴别出正样本（True Positives）或者负样本（True Negatives）都是True。同理，错误的判断正样本（False Positive，即一类错误）或者负样本（False Negative，即二类错误）。  
注意：True和False是对于评价预测结果而言，也就是评价预测结果是正确的(True)还是错误的(False)。而Positive和Negative则是样本分类的标记。

  
metrics模块分类度量有6种方法，如表1所示：

| **指标** | **描述** | **metrics方法** |
| --- | --- | --- |
| Accuracy | 准确度 | from sklearn.metrics import accuracy\_score |
| Precision | 查准率 | from sklearn.metrics import precision\_score |
| Recall | 查全率 | from sklearn.metrics import recall\_score |
| F1 | F1值 | from sklearn.metrics import f1\_score |
| Classification Report | 分类报告 | from sklearn.metrics import classification\_report |
| Confusion Matrix | 混淆矩阵 | from sklearn.metrics import confusion\_matrix |
| ROC | ROC曲线 | from sklearn.metrics import roc\_curve |
| AUC | ROC曲线下的面积 | from sklearn.metrics import auc |

**from sklearn.metrics import accuracy\_score, precision\_score,**

**recall\_score, f1\_score, classification\_report, confusion\_matrix**

**accuracy\_score(y\_test, y\_pred)**

**precision\_score(y\_test, y\_pred)**

**recall\_score(y\_test, y\_pred)**

**f1\_score(y\_test,y\_pred)**

**classification\_report(y\_test,y\_pred)**

**confusion\_matrix(y\_test, y\_pred)**

# 准确度(accuracy)

准确度是预测正确的数（包括正样本和负样本）占所有数的比例。利用accuracy\_score函数对预测数据进行模型评估，其中第一个参数是测试标记，第二个参数是预测标记值

ACC = (TP+TN)/(TP+TN+FP+FN)

# 查准率/精确度（Precision）和查全率/召回率（Recall）

sklearn的metrics模块分别提供了precision\_score和recall\_score函数用来评估分类模型的查全率和查准率。

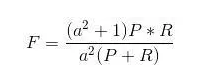
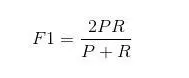
精确度：precision，正确预测为正的，占全部预测为正的比例，TP / (TP+FP)

召回率：recall，正确预测为正的，占全部实际为正的比例，TP / (TP+FN)

假设有一个大小为1000的带布尔标签数据集，里面的“真”样本只有100不到，剩下都是假样本。假设训练一个模型，不管输入什么数据，它只给出“假”的预测，那么正确率依旧是90%以上，很明显，这个时候准确率accuracy就失去它的作用。因此，查全率和查准率一般用在倾斜数据集的时候。

# F1值（F1-Measure）

Precision和Recall指标有的时候是矛盾的，F-Measure综合这二者指标的评估指标，用于综合反映整体的指标。F-Measure是Precision和Recall加权调和平均, a为权重因子，当a = 1时，F值变为最常见的F1了，代表精确率和召回率的权重一样 (fl\_score)

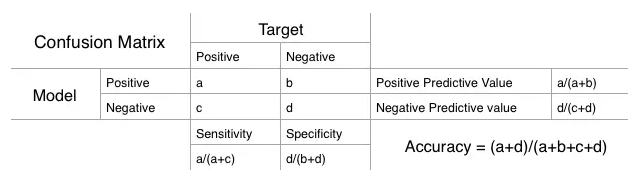
 

# 分类报告(Classification Report)

metrics模块的classification\_report方法，综合提供了查准率（precision）、查全率（recall）和f1值三种评估指标。

# 混淆矩阵(Confusion Matrix)

混淆矩阵是一个N X N矩阵，N为分类的个数。假如我们面对的是一个二分类问题，也就是N＝2，我们就得到一个2 X 2矩阵。在学习这个矩阵之前，我们需要知道一些简单的定义。  
Accuracy(准确度):预测正确的数占所有数的比例。  
Positive Predictive Value(阳性预测值) or Precision(查准率)：阳性预测值被预测正确的比例。  
Negative Predictive Value(阴性预测值)：阴性预测值被预测正确的比例。  
Sensity(灵敏度) or Recall(查全率)：在阳性值中实际被预测正确所占的比例。  
Specificity(特异度)：在阴性值中实现被预测正确所占的比例。



# ROC(receiver operating characteristic curve)

曲线指受试者工作特征曲线或者是接收器操作特性曲线, 虑一个二分问题，即将实例分成正类（positive）或负类（negative）。对一个二分问题来说，会出现四种情况。如果一个实例是正类并且也被 预测成正类，即为真正类（True positive）,如果实例是负类被预测成正类，称之为假正类（False positive）。相应地，如果实例是负类被预测成负类，称之为真负类（True negative）,正类被预测成负类则为假负类（false negative）。

**TP**：正确肯定的数目；

**FN**：漏报，没有正确找到的匹配的数目；

**FP**：误报，给出的匹配是不正确的；

**TN**：正确拒绝的非匹配对数；

从列联表引入两个新名词。其一是真正类率(true positive rate ,TPR), 计算公式为TPR=TP/ (TP+ FN)，刻画的是分类器所识别出的 正实例占所有正实例的比例。另外一个是假正类率(false positive rate, FPR),计算公式为FPR= FP / (FP + TN)，计算的是分类器错认为正类的负实例占所有负实例的比例。还有一个真负类率（True Negative Rate，TNR），也称为specificity,计算公式为**TNR=TN/ (FP+ TN) = 1-FPR。**

**FPR = FP/(FP + TN)** 负样本中的错判率（假警报率）

**TPR = TP/(TP + TN)** 判对样本中的正样本率（命中率）

**ACC = (TP + TN) / P+N** 判对准确率

　　在一个二分类模型中，对于所得到的连续结果，假设已确定一个阀值，比如说 0.6，大于这个值的实例划归为正类，小于这个值则划到负类中。如果减小阀值，减到0.5，固然能识别出更多的正类，也就是提高了识别出的正例占所有正例 的比类，即TPR,但同时也将更多的负实例当作了正实例，即提高了FPR。为了形象化这一变化，在此引入ROC，ROC曲线可以用于评价一个分类器。

　　ROC曲线其实就是从混淆矩阵衍生出来的图形，其横坐标为1-Specificity，纵坐标为Sensitivity。1-specificity=FPR(False positive rate)，即假正类率。Sensitivity=TPR(True positive rate)，即是真正类率。

# 

# 理想情况下，TPR应该接近1，FPR应该接近0。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和FPR。比如Threshold最大时，TP=FP=0，对应于原点；Threshold最小时，TN=FN=0，对应于右上角的点(1,1)。在ROC空间，ROC曲线越凸向左上方向效果越好；越靠近对角线，分类器越趋向于随机分类器。

# 利用metrics计算roc曲线，roc曲线有三个属性：fpr，tpr，和阈值，因此该函数返回这三个变量。

import numpy as np

from sklearn.metrics import roc\_curve

y = np.array([1,1,2,2])

pred = np.array([0.1, 0.4, 0.35, 0.8])

fpr, tpr, thresholds = roc\_curve(y, pred, pos\_label=2)

# AUC（ROC曲线下面积）

# ROC曲线是根据与那条参照线进行比较来判断模型的好坏，但这只是一种直觉上的定性分析，如果我们需要精确一些，就要用到AUC，也就是ROC曲线下面积。其判定方法是AUC应该大于0.5。

# 

# 参考线的面积是0.5，ROC曲线与它偏离越大，ROC曲线就越往左上方靠拢，它下面的面积(AUC)也就越大，这里面积是0.869。我们可以根据AUC的值与0.5相比，来评估一个分类模型的预测效果。如果AUC的值达到0.80，那说明分类器分类非常准确；如果AUC值在0.60～0.80之间，那分类器有优化空间，可以通过调节参数得到更好的性能；如果AUC值小于0.60，那说明分类器模型效果比较差。

# 利用刚才求得的fpr和tpr的结果作为参数，可以求得AUC的值，

from sklearn.metrics import auc

auc(fpr, tpr)

# PR曲线

# PR曲线的横坐标是精确率P，纵坐标是召回率R。评价标准和ROC一样，先看平滑不平滑（蓝线明显好些）。一般来说，在同一测试集，上面的比下面的好（绿线比红线好）。当P和R的值接近时，F1值最大，此时画连接(0,0)和(1,1)的线，线和PRC重合的地方的F1是这条线最大的F1（光滑的情况下），此时的F1对于PRC就好像AUC对于ROC一样。一个数字比一条线更方便调型

# 

**有时候模型没有单纯的谁比谁好（比如图二的蓝线和青线），所以选择模型还是要结合具体的使用场景。下面是两个场景：**

**1，地震的预测 对于地震的预测，我们希望的是RECALL非常高，也就是说每次地震我们都希望预测出来。这个时候我们可以牺牲PRECISION。情愿发出1000次警报，把10次地震都预测正确了，也不要预测100次对了8次漏了两次。**

**2，嫌疑人定罪 基于不错怪一个好人的原则，对于嫌疑人的定罪我们希望是非常准确的。即时有时候放过了一些罪犯（recall低），但也是值得的。**

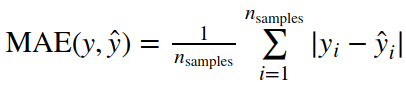
**对于分类器来说，本质上是给一个概率，此时，我们再选择一个CUTOFF点（阀值），高于这个点的判正，低于的判负。那么这个点的选择就需要结合你的具体场景去选择。反过来，场景会决定训练模型时的标准，比如第一个场景中，我们就只看RECALL=99.9999%（地震全中）时的PRECISION，其他指标就变得没有了意义。  
当正负样本数量差距不大的情况下，ROC和PR的趋势是差不多的，但是在正负样本分布极不均衡的情况下，PRC比ROC更能真实的反映出实际情况，因为此时ROC曲线看起来似乎很好，但是却在PR上效果一般。**

**二、回归问题**

拟合（回归）问题比较简单，所用到的衡量指标也相对直观。假设yiyi是第ii个样本的真实值，ŷ iy^i是对第ii个样本的预测值。

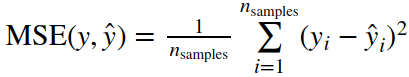
## 1. 平均绝对误差（MAE）

平均绝对误差MAE（Mean Absolute Error）又被称为l1范数损失（l1-norm loss）：



## 2. 平均平方误差（MSE）

平均平方误差MSE（Mean Squared Error）又被称为l2范数损失（l2-norm loss）：

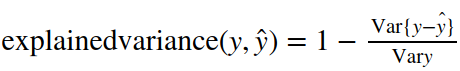


**3、均方根误差（RMSE）**

RMSE虽然广为使用，但是其存在一些缺点，因为它是使用平均误差，而平均值对异常点（outliers）较敏感，如果回归器对某个点的回归值很不理性，那么它的误差则较大，从而会对RMSE的值有较大影响，即平均值是非鲁棒的。

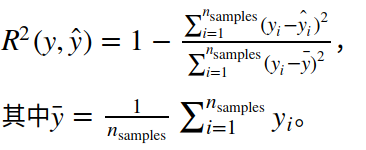
**4、解释变异**

解释变异（ Explained variance）是根据误差的方差计算得到的：



## 5、决定系数

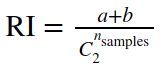
决定系数（Coefficient of determination）又被称为R2分数：



**三、聚类**

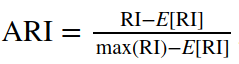
## 1 . 兰德指数

兰德指数（Rand index）需要给定实际类别信息C，假设K是聚类结果，a表示在C与K中都是同类别的元素对数，b表示在C与K中都是不同类别的元素对数，则兰德指数为：



其中https://images2018.cnblogs.com/blog/1180120/201808/1180120-20180806180839468-1245828502.png数据集中可以组成的总元素对数，RI取值范围为[0,1]，值越大意味着聚类结果与真实情况越吻合。

对于随机结果，RI并不能保证分数接近零。为了实现“在聚类结果随机产生的情况下，指标应该接近零”，调整兰德系数（Adjusted rand index）被提出，它具有更高的区分度：

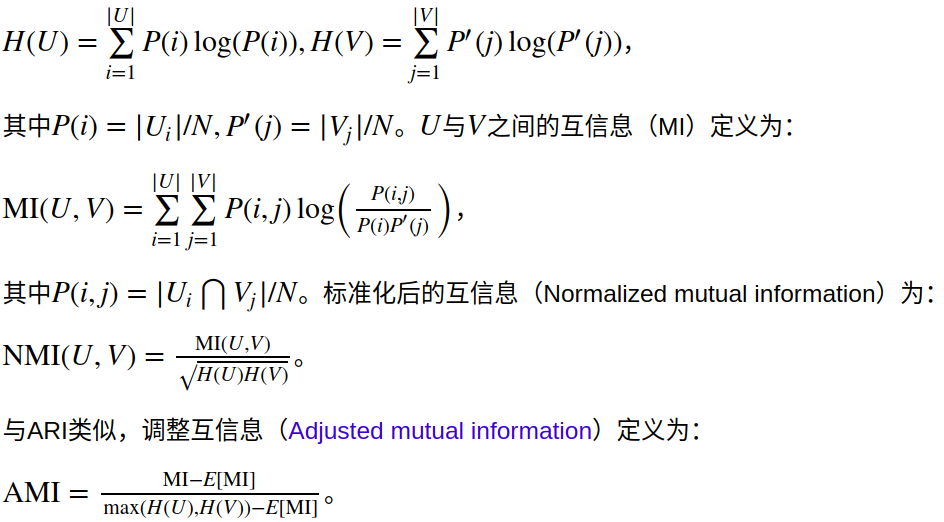


具体计算方式参见[Adjusted Rand index](https://en.wikipedia.org/wiki/Rand_index#Adjusted_Rand_index)。

ARI取值范围为[−1,1]，值越大意味着聚类结果与真实情况越吻合。从广义的角度来讲，ARI衡量的是两个数据分布的吻合程度。

## 2. 互信息

互信息（Mutual Information）也是用来衡量两个数据分布的吻合程度。假设UU与VV是对NN个样本标签的分配情况，则两种分布的熵（熵表示的是不确定程度）分别为：



利用基于互信息的方法来衡量聚类效果需要实际类别信息，MI与NMI取值范围为[0,1]，AMI取值范围为[−1,1]，它们都是值越大意味着聚类结果与真实情况越吻合。

## 3. 轮廓系数

轮廓系数（Silhouette coefficient）适用于实际类别信息未知的情况。对于单个样本，设aa是与它同类别中其他样本的平均距离，bb是与它距离最近不同类别中样本的平均距离，轮廓系数为：

https://images2018.cnblogs.com/blog/1180120/201808/1180120-20180806181107733-987187523.png

对于一个样本集合，它的轮廓系数是所有样本轮廓系数的平均值。

轮廓系数取值范围是[−1,1][−1,1]，同类别样本越距离相近且不同类别样本距离越远，分数越高。

# 四、信息检索

信息检索评价是对信息检索系统性能（主要满足用户信息需求的能力）进行评估，与机器学习也有较大的相关性，感兴趣的可以参考[这篇](http://blog.sina.com.cn/s/blog_72995dcc01013oo9.html)不错的博文。

# 

# 在机器学习模型训练过程中，由于偏差过大导致的模型欠拟合，以及由于方差过大导致的过拟合，都严重影响着模型的好坏，为了解决这两个问题，我们需要一整套方法及评价指标，来评价所训练模型的效果，其中评估方法用于评估模型的泛化能力，而性能指标则用于评价单个模型性能的高低。

# 模型的泛化性能是由学习算法的能力，数据的充分性及学习任务本身的难度所决定的，良好的泛化性能代表了较小的偏差，即算法的期望预测结果与真实结果的偏离程度，同时还要有较小的方差，即随训练样本的变化算法本身的学习能力变化不大。

# *泛化能力（generalization ability）是指机器学习算法对新鲜样本的适应能力。学习的目的是学到隐含在数据背后的规律，对具有同一规律的学习集以外的数据，经过训练的网络也能给出合适的输出，该能力称为泛化能力。*

# 集成算法的预测：<https://blog.csdn.net/sinat_29957455/article/details/79777187>

# 关于宏平均（Macro-averaging）和微平均（Micro-averaging）:

# <https://blog.csdn.net/chenpe32cp/article/details/102519255>

# <https://www.jianshu.com/p/e19f557f07af>

# ROC深入理解：<https://blog.csdn.net/u012837965/article/details/85761159>

# sklearn—评价指标大全：<https://blog.csdn.net/Monk_donot_know/article/details/86614558>

# <https://www.cnblogs.com/guoyaohua/p/classification-metrics.html>