机器学习评估指标的前世今生

我们先来看一下sklearn中支持哪些机器学习的评估指标：

```
from sklearn.metrics import SCORERS
SCORERS
```

> {'accuracy': make_scorer(accuracy_score), 'adjusted_rand_score': make_scorer(adjusted_rand_score), 'average_precision': make_scorer(average_precision_score, needs_threshold=True), 'f1': make_scorer(f1_score), 'f1_macro': make_scorer(f1_score, average=macro, pos_label=None), 'f1_micro': make_scorer(f1_score, average=micro, pos_label=None), 'f1_samples': make_scorer(f1_score, average=samples, pos_label=None), 'f1_weighted': make_scorer(f1_score, average=weighted, pos_label=None), 'log_loss': make_scorer(log_loss, greater_is_better=False, needs_proba=True), 'mean_absolute_error': make_scorer(mean_absolute_error, greater_is_better=False), 'mean_squared_error': make_scorer(mean_squared_error, greater_is_better=False), 'median_absolute_error': make_scorer(median_absolute_error, greater_is_better=False), 'neg_log_loss': make_scorer(log_loss, greater_is_better=False, needs_proba=True), 'neg_mean_absolute_error': make_scorer(mean_absolute_error, greater_is_better=False), 'neg_mean_squared_error': make_scorer(mean_squared_error, greater_is_better=False), 'neg_median_absolute_error': make_scorer(median_absolute_error, greater_is_better=False), 'precision': make_scorer(precision_score), 'precision_macro': make_scorer(precision_score, average=macro, pos_label=None), 'precision_micro': make_scorer(precision_score, average=micro, pos_label=None), 'precision_samples': make_scorer(precision_score, average=samples, pos_label=None), 'precision_weighted': make_scorer(precision_score, average=weighted, pos_label=None), 'r2': make_scorer(r2_score), 'recall': make_scorer(recall_score), 'recall_macro': make_scorer(recall_score,

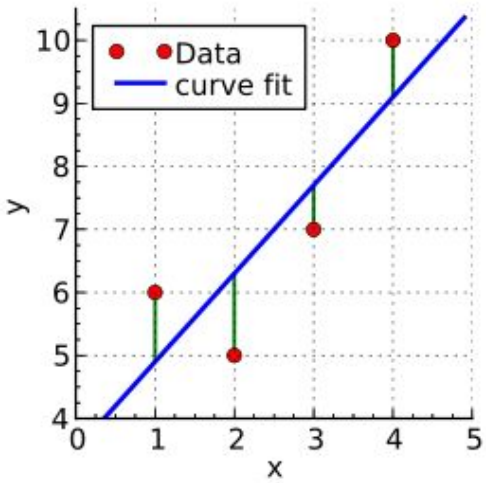
> average=macro, pos_label=None), 'recall_micro': make_scorer(recall_score, average=micro, pos_label=None), 'recall_samples': make_scorer(recall_score, average=samples, pos_label=None), 'recall_weighted': make_scorer(recall_score, average=weighted, pos_label=None), 'roc_auc': make_scorer(roc_auc_score, needs_threshold=True)}

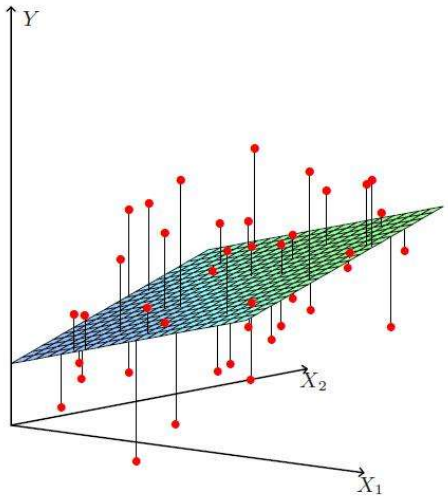
哇，好多，这么多指标，我们都要会用吗？实际上不是这样的，我们平时工作中只会使用一些比较常见的指标，并且根据这些常见的指标（注意不是一个指标！）综合评价我们的模型。下面我们就一一揭晓这些常见评估方法的神秘面纱！

1. 回归（Regression）算法指标大揭秘

- **Mean Absolute Error 平均绝对误差**
- **Mean Squared Error 均方误差**
- **Root Mean Squared Error：均方根误差**
- **Coefficient of determination 决定系数**

以下为一元变量和二元变量的线性回归示意图：





怎样来衡量回归模型的好坏呢？

我们自然而然会想到采用残差（实际值与预测值差值）的均值来衡量，即：

$$\text{residual}(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$

问题①： 用残差的均值合理吗？

当实际值分布在拟合曲线两侧时，对于不同样本而言$y_i - \hat{y}_i$有正有负，相互抵消，因此我们想到采用预测值和真实值之间的距离来衡量。

1.1 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（$l1 - normloss$），

$$\mathrm{MAE}(y,\hat{y}) = \frac{1}{m}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

问题②： MAE有哪些不足?

MAE虽能较好衡量回归模型的好坏，但是绝对值的存在导致函数不光滑，在某些点上不能求导，可以考虑将绝对值改为残差的平方，这就是均方误差。

1.2 均方误差 MSE

均方误差MSE（Mean Squared Error）又被称为 $l2$ 范数损失（$l2-normloss$）。

$$\mathrm{MSE}(y,\hat{y}) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

问题③： 还有没有比MSE更合理一些的指标?

由于MSE与我们的目标变量的量纲不一致，为了保证量纲一致性，我们需要对MSE进行开方。

1.3 均方根误差 RMSE

$$\mathrm{RMSE}(y,\hat{y}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

问题④： 有没有规范化（无量纲化的指标）?

上面的几种衡量标准的取值大小与具体的应用场景有关系，很难定义统一的规则来衡量模型的好坏。比如说利用机器学习算法预测上海的房价RMSE在2000元，我们是可以接受的，但是当四五线城市的房价RMSE为2000元，我们还可以接受吗？有没有一种与应用场景无关的，统一的评价指标呢?

1.4 决定系数 R^2

决定系数又称为 R^2 score，反应因变量的全部变异能通过回归关系被自变量解释的比例。

$\mathrm{SST} = \sum_{i}^{m}(y_i - \bar{y})^2 \qquad \mathrm{SST} = \text{total sum of squares}$

$\mathrm{SSR} = \sum_{i}^{m}(\hat{y}_i - \bar{y})^2 \qquad \mathrm{SSR} = \text{sum of due to regression}$

$\mathrm{SSE} = \sum_{i}^{m}(\hat{y}_i - y_i)^2 \qquad \mathrm{SSE} = \text{sum of due to erros}$

$\mathrm{SST} = \mathrm{SSR} + \mathrm{SSE}$

$$R^2(y,\hat{y}) = \frac{\mathrm{SSR}}{\mathrm{SST}}$$

如果结果是0，就说明模型预测不能预测因变量。 如果结果是1。就说明是函数关系。 如果结果是0-1之间的数，就是我们模型的好坏程度。

化简上面的公式 ,分子就变成了我们的均方误差MSE，下面分母就变成了方差:

$$R^2(y,\hat{y}) = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2/m}{\sum_{i=1}^{m}(y_i - \bar{y})^2/m} = 1 - \frac{\text{MSE}(\hat{\text{y}}, \text{y})}{\text{Var}(\text{y})}$$

2. 分类（Classification）算法指标大揭秘

- **精度 Accuracy**
- **混淆矩阵 Confusion Matrix**
- **准确率（查准率） Precision**
- **召回率（查全率）Recall**
- **F_β Score**
- **AUC Area Under Curve**
- **KS Kolmogorov-Smirnov**

2.1 精度 Acc

预测正确的样本的占总样本的比例，取值范围为[0,1]，取值越大，模型预测能力越好。

$$Acc(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{m} sign(\hat{y}_i, y_i)$$

其中$sign(\hat{y}_i, y_i) = \begin{cases} 1 & \hat{y}_i = y_i \\ 0 & \hat{y}_i \neq y_i \end{cases}$

问题⑤： 什么时候精度指标会失效?

当样本中类别数量严重不均衡的时候，如正样本990个，负样本10个，直接把所有样本分类为正样本，得到识别率为99%，但这显然是没有意义的。单纯根据Accuracy来衡量算法的优劣已经不能表征这种非均衡问题。这个时候就需要对目标变量的真实值和预测值做更深入的分析。

2.2 混淆矩阵 Confusion Matrix

混淆矩阵，在无监督学习中被称为匹配矩阵(matching matrix)，之所以叫混淆矩阵，是因为我们能够很easy从图表中看到学习器有没有将样本的类别给混淆了。矩阵每一列表达分类器预测，每一行表示样本所属真实的类别。

混淆矩阵如下图所示：

		predicted condition	
	total population	prediction positive	prediction negative
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**

这里牵扯到三个方面：真实值，预测值，预测值和真实值之间的关系，其中任意两个方面都可以确定第三个。

通常取预测值和真实值之间的关系、预测值对矩阵进行划分：

- **True positive (TP)**

  真实值为Positive，预测正确（预测值为Positive）

- **True negative (TN)**

  真实值为Negative，预测正确（预测值为Negative）

- **False positive (FP)**

  真实值为Negative，预测错误（预测值为Positive），第一类错误， Type I error

- **False negative (FN)**

  真实值为Positive，预测错误（预测值为 Negative），第二类错误， Type II error

混淆矩阵的衍生指标：

		predicted condition			
	total population	prediction positive	prediction negative	Prevalence = $\frac{\Sigma \text{ condition positive}}{\Sigma \text{ total population}}$	
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)	True Positive Rate (TPR), Sensitivity, Recall, Probability of Detection = $\frac{\Sigma \text{ TP}}{\Sigma \text{ condition positive}}$	False Negative Rate (FNR), Miss Rate = $\frac{\Sigma \text{ FN}}{\Sigma \text{ condition positive}}$
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**	False Positive Rate (FPR), Fall-out, Probability of False Alarm = $\frac{\Sigma \text{ FP}}{\Sigma \text{ condition negative}}$	True Negative Rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ TN}}{\Sigma \text{ condition negative}}$
	Accuracy = $\frac{\Sigma \text{ TP} + \Sigma \text{ TN}}{\Sigma \text{ total population}}$	Positive Predictive Value (PPV), Precision = $\frac{\Sigma \text{ TP}}{\Sigma \text{ prediction positive}}$	False Omission Rate (FOR) = $\frac{\Sigma \text{ FN}}{\Sigma \text{ prediction negative}}$	Positive Likelihood Ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic Odds Ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
		False Discovery Rate (FDR) = $\frac{\Sigma \text{ FP}}{\Sigma \text{ prediction positive}}$	Negative Predictive Value (NPV) = $\frac{\Sigma \text{ TN}}{\Sigma \text{ prediction negative}}$	Negative Likelihood Ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

specificity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

false discovery rate (FDR)

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPV}$$

false omission rate (FOR)

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}} = 1 - \mathrm{NPV}$$

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

F1 score

is the harmonic mean of precision and sensitivity

$$F_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Matthews correlation coefficient (MCC)

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

Informedness or Bookmaker Informedness (BM)

$$\mathrm{BM} = \mathrm{TPR} + \mathrm{TNR} - 1$$

Markedness (MK)

$$\mathrm{MK} = \mathrm{PPV} + \mathrm{NPV} - 1$$

2.3 准确率（查准率） Precision

Precision是分类器预测的正样本中预测正确的比例，取值范围为[0,1]，取值越大，模型预测能力越好。

$$\mathrm{P} = \frac{TP}{TP + FP}$$

2.4 召回率（查全率）Recall

Recall是分类器所识别出的正样本占所有正样本的比例，取值范围为[0,1]，取值越大，模型预测能力越好。

$$\mathrm{R} = \frac{TP}{TP + FN}$$

应用场景：

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。即使有时候放过了一些罪犯，但也是值得的。因此我们希望有较高的Precision值，可以合理地牺牲Recall。

问题⑥： 某一家互联网金融公司风控部门的主要工作是利用机器模型抓取坏客户。互联网金融公司要扩大业务量，尽量多的吸引好客户，此时风控部门该怎样调整Recall和Precision？如果公司坏账扩大，公司缩紧业务，尽可能抓住更多的坏客户，此时风控部门该怎样调整Recall和Precision?

如果互联网公司要扩大业务量，为了减少好客户的误抓率，保证吸引更多的好客户，风控部门就会提高阈值，从而提高模型的查准率Precision，同时，也会放进一部分坏客户，导致查全率Recall下降。如果公司要缩紧业务，尽可能抓住更多的坏客户，风控部门就会降低阈值，从而提高模型的查全率Recall，但是这样会导致一部分好客户误抓，从而降低模型的查准率Precision。

问题⑦： 可不可以用一个指标来权衡Recall和Precision?

Recall和Precision的加权调和平均值作为衡量标准。

2.5 F_β Score

Precision和Recall 是互相影响的，理想情况下肯定是做到两者都高，但是一般情况下Precision高、Recall 就低，Recall 高、Precision就低。为了均衡两个指标，我们可以采用Precision和Recall的加权调和平均（weighted harmonic mean）来衡量，即F_β Score，公式如下：

$$F_\beta = (1+\beta^2) \times \frac{\mathrm{P} \times \mathrm{R}}{\beta^2 \times \mathrm{P} + \mathrm{R}}$$

β表示权重，

$$\begin{aligned} F_\beta &= \frac{(1+\beta^2) \times \mathrm{P} \times \mathrm{R}}{\beta^2 \times \mathrm{P} + \mathrm{R}} \\ &= \frac{1}{\frac{\beta^2}{(1+\beta^2)\times R} + \frac{1}{(1+\beta^2)\times P}} \\ &= \frac{1}{\frac{1}{(1+\frac{1}{\beta^2})\times R} + \frac{1}{(1+\beta^2)\times P}} \end{aligned}$$

当$\beta \to 0$: $F_\beta \approx P$；当$\beta \to \infty$: $F_\beta \approx R$。

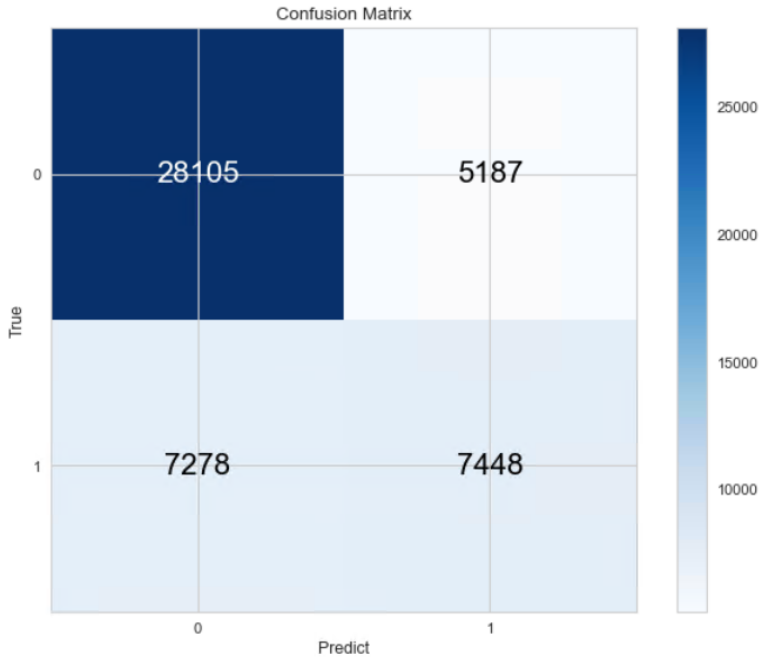
通俗的语言就是：β越大，Recall的权重越大，β越小，Precision的权重越大。

随着如 $\beta = 1$ 为F_1，此时Precision和Recall的权重相等，公式如下：

$$F_\beta = F_1 = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

由于F_β Score 无法直观反映数据的情况，同时业务含义相对较弱，实际工作用到的不多。

问题⑧： 根据以下混淆矩阵计算Recall, Precision，TPR，FPR?



$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN} = \frac{7448}{7448 + 7278} = 0.50577$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{7448}{7448 + 5187} = 0.5895$$

$$\text{FPR} = \frac{FP}{FP + TN} = \frac{5187}{5187 + 58105} = 0.1136$$

问题⑨： 有没有办法观察模型好坏随阈值的变化趋势呢（提示：利用TPR, FPR两个指标）？

有，ROC（Receiver Operating Characteristic）曲线

问题⑩： 以上指标都是基于阈值的，而lr模型输出的是概率，有没有一种指标不依赖于阈值?

有，AUC

2.6 ROC 和 AUC

AUC是一种模型分类指标，且仅仅是二分类模型的评价指标。AUC是Area Under Curve的简称，那么Curve就是ROC（Receiver Operating Characteristic），翻译为"接受者操作特性曲线"。

2.6.1 ROC

曲线由两个变量TPR和FPR组成，这个组合以FPR对TPR，即是以代价(costs)对收益(benefits)。

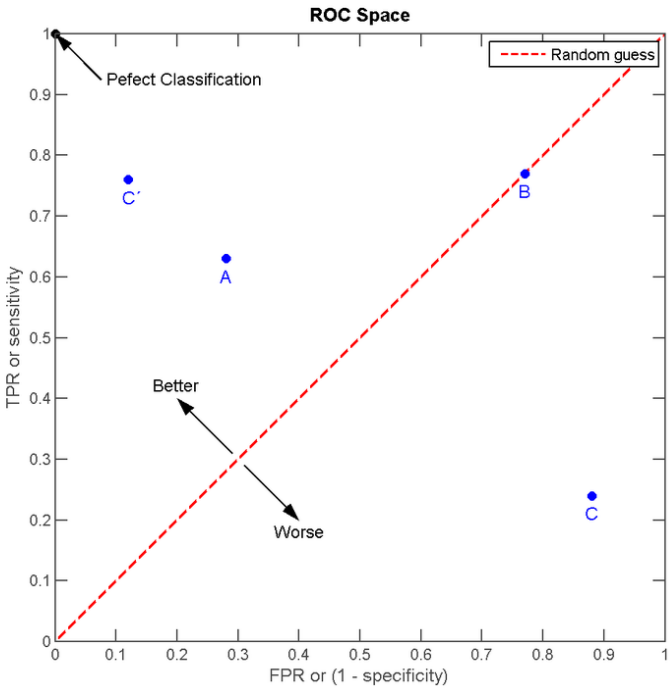
- x轴为假阳性率（FPR）：在所有的负样本中，分类器预测错误的比例
  $FPR = \frac{FP}{FP+TN}$
- y轴为真阳性率（TPR）：在所有的正样本中，分类器预测正确的比例（等于Recall）
  $TPR = \frac{TP}{TP+FN}$

为了更好地理解ROC曲线，我们使用具体的实例来说明：

如在医学诊断中,判断有病的样本。那么尽量把有病的揪出来是主要任务，也就是第一个指标TPR，要越高越好。而把没病的样本误诊为有病的，也就是第二个指标FPR，要越低越好。

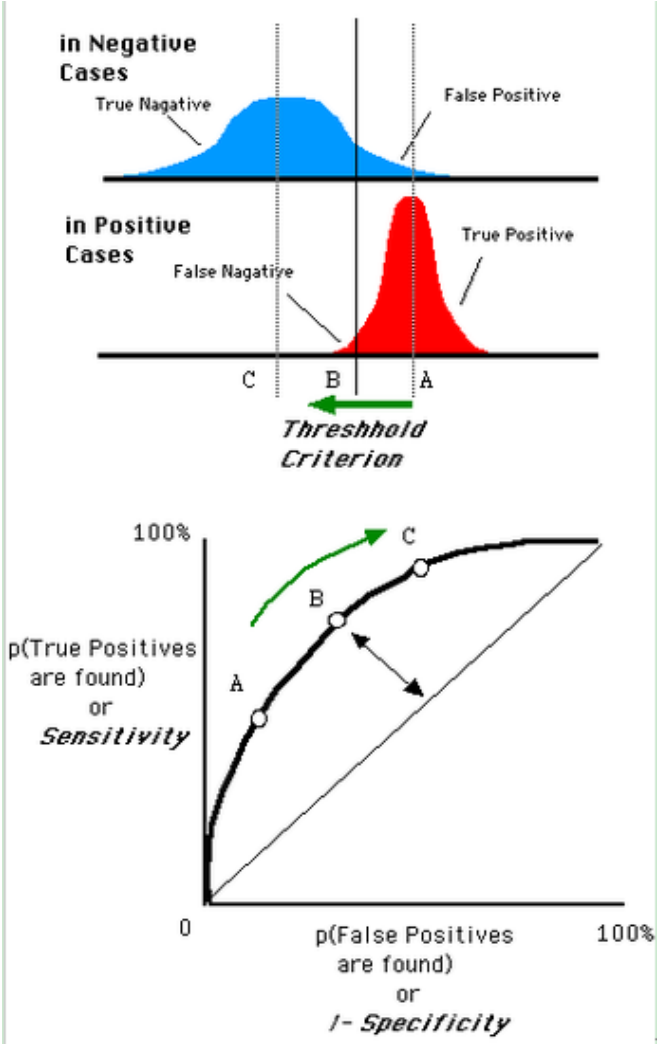
不难发现,这两个指标之间是相互制约的。如果某个医生对于有病的症状比较敏感，稍微的小症状都判断为有病,那么他的第一个指标应该会很高，但是第二个指标也就相应地变高。最极端的情况下,他把所有的样本都看做有病,那么第一个指标达到1,第二个指标也为1。

我们以FPR为横轴,TPR为纵轴,得到如下ROC空间。

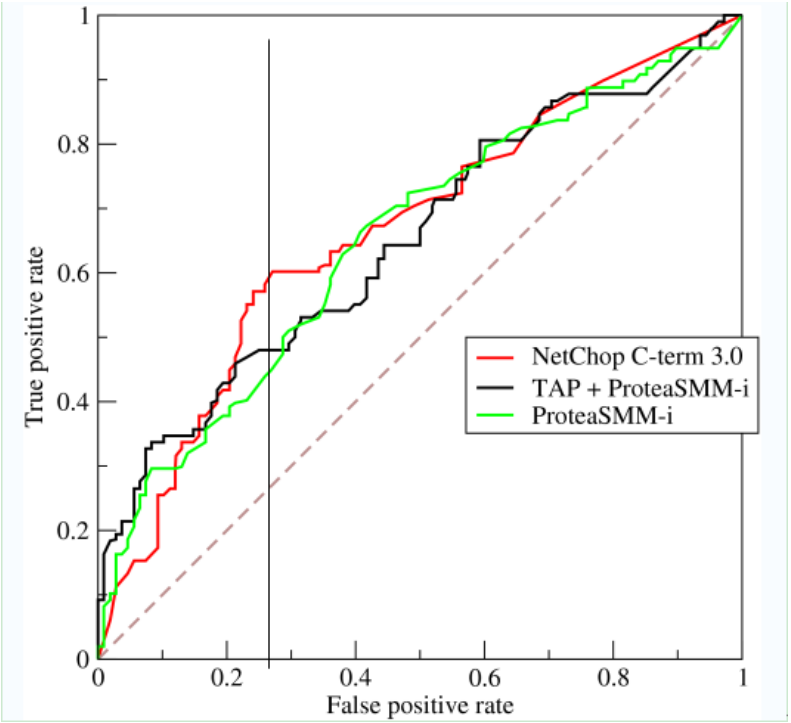


我们可以看出，左上角的点(TPR=1，FPR=0)，为完美分类，也就是这个医生医术高明，诊断全对。点A(TPR>FPR),医生A的判断大体是正确的。中线上的点B(TPR=FPR),也就是医生B全都是蒙的，蒙对一半，蒙错一半；下半平面的点C(TPR<FPR)，这个医生说你有病，那么你很可能没有病，医生C的话我们要反着听，为真庸医。上图中一个阈值，得到一个点。现在我们需要一个独立于阈值的评价指标来衡量这个医生的医术如何，也就是遍历所有的阈值,得到ROC曲线。

假设如下就是某个医生的诊断统计图，直线代表阈值。通过改变不同的阈值$\mathbf{1.0 \rightarrow 0}$，从而绘制出ROC曲线。下图为未得病人群（蓝色）和得病人群（红色）的模型输出概率分布图（横坐标表示模型输出概率，纵坐标表示概率对应的人群的数量）。阈值为1时，不管你什么症状，医生均未诊断出疾病（预测值都为N），此时FPR=TPR=0，位于左下。阈值为0时，不管你什么症状，医生都诊断结果都是得病（预测值都为P），此时FPR=TPR=1，位于右上。



曲线距离左上角越近,证明分类器效果越好。



如上，是三条ROC曲线，在0.23处取一条直线。那么，在同样的低FPR=0.23的情况下，红色分类器得到更高的PTR。也就表明，ROC越往左上，分类器效果越好。我们用一个标量值AUC来量化它。

2.6.2 AUC

AUC定义：

AUC值为ROC曲线所覆盖的区域面积，显然，AUC越大，分类器分类效果越好。

AUC = 1，是完美分类器。绝大多数预测的场合，不存在完美分类器。

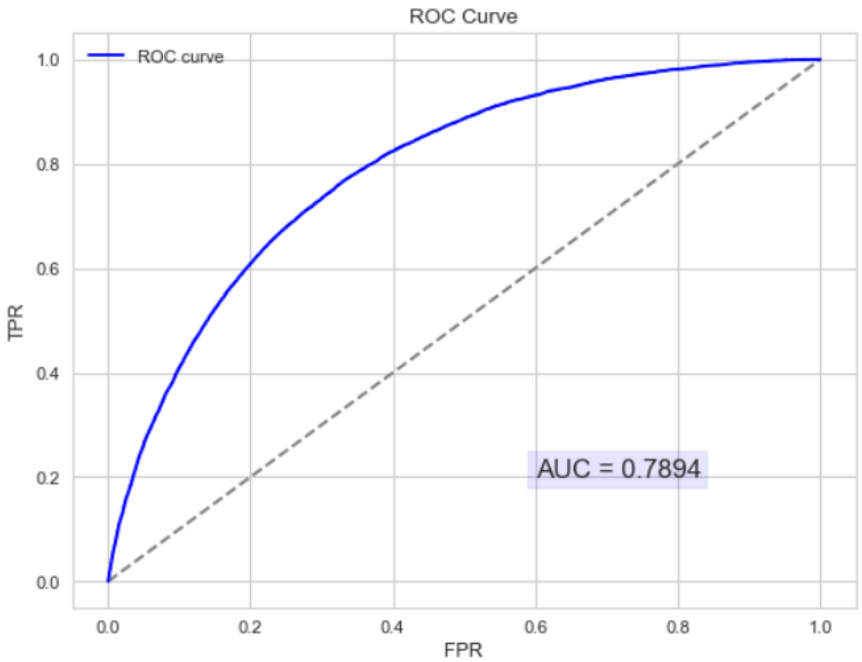
0.5 < AUC < 1，优于随机猜测。这个分类器（模型）妥善设定阈值的话，能有预测价值。

AUC = 0.5，跟随机猜测一样（例：丢铜板），模型没有预测价值。

AUC < 0.5，比随机猜测还差；但只要总是反预测而行，就优于随机猜测。

注：对于AUC小于0.5的模型，我们可以考虑取反（模型预测为positive，那我们就取negtive），这样就可以保证模型的性能不可能比随机猜测差。

以下为ROC曲线和AUC值的实例：



AUC的物理意义

AUC的物理意义正样本的预测结果大于负样本的预测结果的概率。所以AUC反应的是分类器对样本的排序能力。

另外值得注意的是，AUC对样本类别是否均衡并不敏感，这也是不均衡样本通常用AUC评价分类器性能的一个原因。

下面从一个小例子解释AUC的含义：小明一家四口，小明5岁，姐姐10岁，爸爸35岁，妈妈33岁建立一个逻辑回归分类器，来预测小明家人为成年人概率，假设分类器已经对小明的家人做过预测，得到每个人为成人的概率。

1. AUC更多的是关注对计算概率的排序，关注的是概率值的相对大小，与阈值和概率值的绝对大小没有关系

例子中并不关注小明是不是成人，而关注的是，预测为成人的概率的排序。

问题⑪：以下为三种模型的输出结果，求三种模型的AUC。

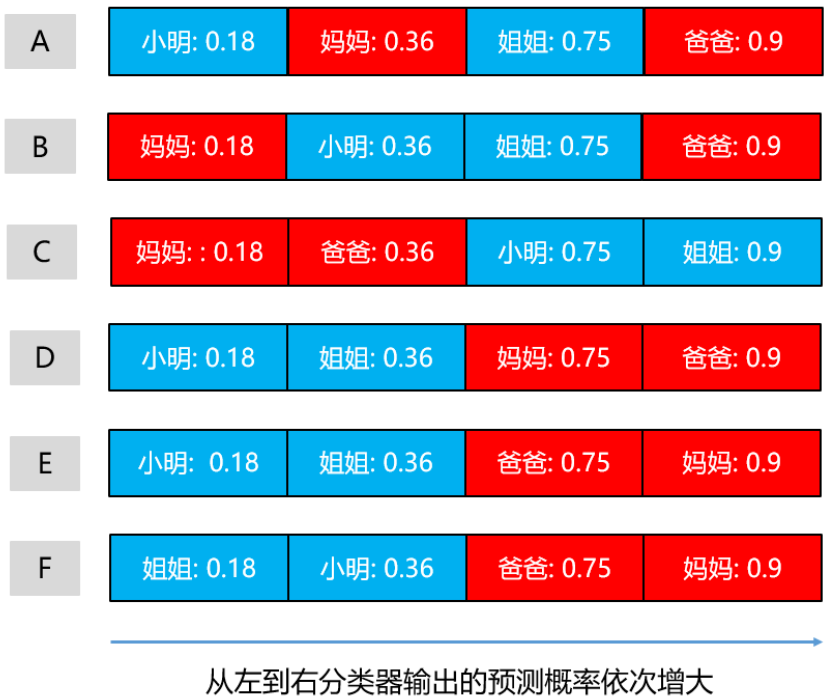
	小明	姐姐	妈妈	爸爸
a	0.12	0.35	0.76	0.85
b	0.12	0.35	0.44	0.49
c	0.52	0.65	0.76	0.85

AUC只与概率的相对大小（概率排序）有关，和绝对大小没关系。由于三个模型概率排序的前两位都是未成年人（小明，姐姐），后两位都是成年人（妈妈，爸爸），因此三个模型的AUC都等于。

1. AUC只关注正负样本之间的排序，并不关心正样本内部，或者负样本内部的排序。这也体现了AUC的本质：任意个正样本的概率都大于负样本的概率的能力

   例子中AUC只需要保证（小明和姐姐）（爸爸和妈妈），小明和姐姐在前2个排序，爸爸和妈妈在后2个排序，而不会考虑小明和姐姐谁在前，或者爸爸和妈妈谁在前。

   问题⑫：以下已经对分类器输出概率从小到大进行了排列，哪些情况的AUC等于1， 情况的AUC为0（其中背景色表示True value，红色表示成年人，蓝色表示未成年人）。



   D 模型, E模型和F模型的AUC值为1，C模型的AUC值为0（爸妈为成年人的概率小于小明和姐姐，显然这个模型预测反了）。

AUC的计算：

- 法1：AUC为ROC曲线下的面积，那我们直接计算面积可得。面积为一个个小的梯形面积（曲线）之和。计算的精度与阈值的精度有关。
- 法2：根据AUC的物理意义，我们计算正样本预测结果大于负样本预测结果的概率。取n1*n0(n1为正样本数，n0为负样本数)个二元组，比较score（预测结果），最后得到AUC。时间复杂度为O(N*M)。

- 法3：我们首先把所有样本按照score排序，依次用rank表示他们，如最大score的样本，rank=n (n=n0+n1，其中n0为负样本个数，n1为正样本个数)，其次为n-1。那么对于正样本中rank最大的样本，rank_max，有n1-1个其他正样本比他score小,那么就有(rank_max-1)-(n1-1)个负样本比他score小。其次为(rank_second-1)-(n1-2)。最后我们得到正样本大于负样本的概率为

$$AUC = \frac{\sum_{正样本} rank(score) - \frac{n_1*(n_1+1)}{2}}{n_0 * n_1}$$

时间复杂度为O(N+M)。

下面有一个简单的例子：

- 真实标签为 (1, 0, 0, 1, 0)
- 预测结果1（0.9, 0.3, 0.2, 0.7, 0.5）
- 预测结果2（0.9, 0.3, 0.2, 0.7, 0.8））
- 分别对两个预测结果进行排序，并提取他们的序号
  - 结果1 (5, 2, 1, 4, 3)
  - 结果2 (5, 2, 1, 3, 4)
- 对正分类序号累加
  - 结果1：$\sum_{正样本} RANK(score) = 5 + 4 = 9$
  - 结果2: $\sum_{正样本} RANK(score) = 5 + 3 = 8$
- 计算两个结果的AUC:
  - 结果1：$AUC = \frac{9-\frac{2*(2+1)}{2}}{2*3} = 1$
  - 结果2：$AUC = \frac{8-\frac{2*(2+1)}{2}}{2*3} = 0.833$

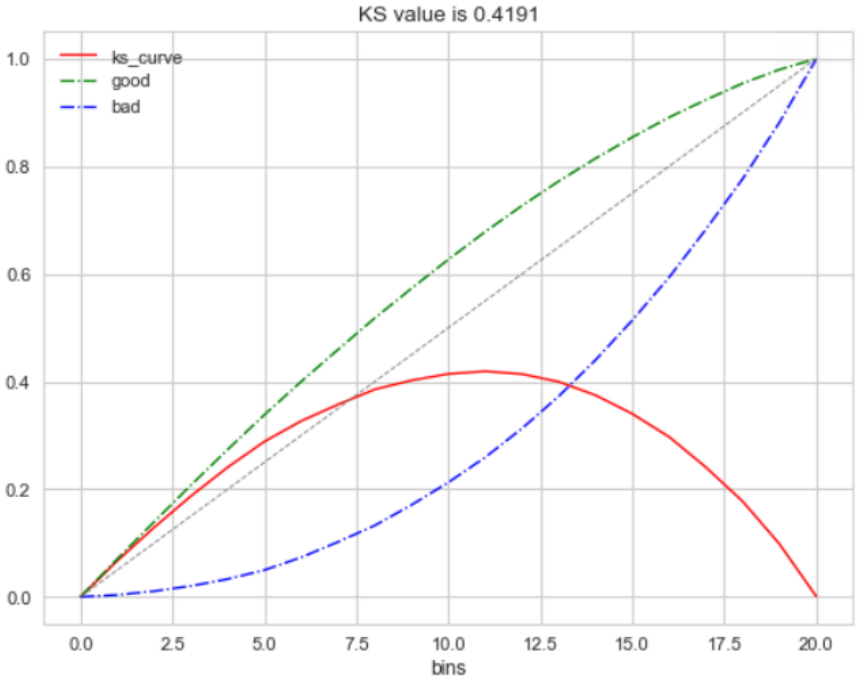
2.7 KS Kolmogorov-Smirnov

KS值是在模型中用于区分预测正负样本分隔程度的评价指标，一般应用于金融风控领域。与ROC曲线相似，ROC是以FPR作为横坐标，TPR作为纵坐标，通过改变不同阈值，从而得到ROC曲线。而在KS曲线中，则是以阈值作为横坐标，以FPR和TPR作为纵坐标，ks曲线则为TPR-FPR，ks曲线的最大值通常为ks值。

为什么这样求KS值呢？我们知道，当阈值减小时，TPR和FPR会同时减小，当阈值增大时，TPR和FPR会同时增大。而在实际工程中，我们希望TPR更大一些，FPR更小一些，即TPR-FPR越大越好，即ks值越大越好。

KS值的取值范围是[0，1]。通常来说，值越大，模型区分正负样本的能力越强（一般0.3以上，说明模型的效果比较好）。

以下为ks曲线的实例 (这里的ks曲线是将score升序排列之后，进行了分组，所以x轴是分组号，而不是阈值)：



3. 评估指标和代价函数是一家人吗?

代价函数：$f(\theta, y)$，又称Cost function，代价函数用来确定模型参数保证模型最优。代价函数越小，模型性能越好。

评判指标：$f(\hat{y}, y)$，用于评估模型好坏。评判指标越大（或越小），模型越好。

本质上代价函数和评判指标都是一家人，只他们的应用场景不同，分工不同。代价函数是用来优化模型参数的，评价指标是用来评判模型好坏的。

作为代价函数所具备的条件：

1. 函数光滑且可导：可用梯度下降求解极值
2. 函数为凸函数：可用梯度下降法求最优解

……

作为评判指标所具备的条件：

1. 直观，可以理解

……

4. 补充小知识点：micro还是macro?

假如我们有n个二分类混淆矩阵，怎样综合评价我们的模型呢？我们通常有两种方式一种叫macro，一种叫micro。

4.1 macro方法

1. 计算出各混淆矩阵的Recall，Precision，记为(P_1, R_1)，(P_2, R_2)，$\cdots$，(P_n, R_n)：

$$P_i = \frac{TP_i}{TP_i + FP_i}$$
$$R_i = \frac{TP_i}{TP_i + FN_i}$$

2. 对各个混淆矩阵的Recall，Precision求平均，然后再根据求得的Recall，Precision计算F1。

$$P_{macro} = \frac{1}{n}\sum_{i=1}^{n} P_i$$
$$R_{macro} = \frac{1}{n}\sum_{i=1}^{n} R_i$$
$$F1_{macro} = \frac{2 \times P_{macro} \times R_{macro}}{P_{macro} + R_{macro}}$$

4.2 micro方法

1. 将各混淆矩阵对应的元素进行平均，得到平均混淆矩阵：

$$\overline{TP} = \frac{1}{n}\sum_{i=1}^{n}(TP)_i$$
$$\overline{TN} = \frac{1}{n}\sum_{i=1}^{n}(TN)_i$$
$$\overline{FP} = \frac{1}{n}\sum_{i=1}^{n}(FP)_i$$
$$\overline{FN} = \frac{1}{n}\sum_{i=1}^{n}(FN)_i$$

2. 再基于平均混淆矩阵计算Recall，Precision，然后再根据求得的Recall，Precision计算F1：

$$P_{micro} = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}$$
$$R_{micro} = \frac{\overline{TP}}{\overline{TP} + \overline{FN}}$$
$$F1_{micro} = \frac{2 \times P_{micro} \times R_{micro}}{P_{micro} + R_{micro}}$$