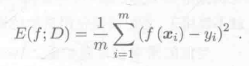
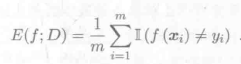
1. **模型性能度量**
2. 回归任务

均方误差： 

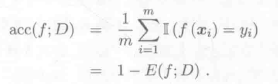
更一般的形式：对于数据分布D和概率密度函数p



1. 分类任务
2. 错误率：分类错误的样本数占样本总数的比例



精确度：分类正确的样本数占总样本数的比例



1. 查准率：预测为真时实际为真的概率



查全率：实际为真时预测为真的概率





F1值：查准率与查全率的调和平均值





Fβ：查准率与查全率的加权平均值，衡量模型对查准率与查全率的偏好





其中，β>0度量了查全率对查准率的相对重要性，

β=1时，为标准的F1；

β>1时，查全率有更大影响；

β<1时，查准率有更大影响

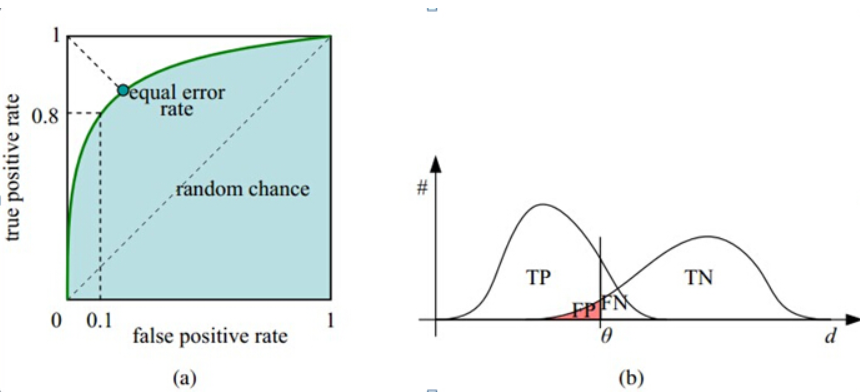
1. ROC与AUC

**真正类率(True Postive Rate)TPR**: TP/(TP+FN),代表分类器预测的正类中实际正实例占所有正实例的比例。Sensitivity

**负正类率(False Postive Rate)FPR**: FP/(FP+TN)，代表分类器预测的正类中实际负实例占所有负实例的比例。1-Specificity

**真负类率(True Negative Rate)TNR**: TN/(FP+TN),代表分类器预测的负类中实际负实例占所有负实例的比例，TNR=1-FPR。Specificity

假设采用逻辑回归分类器，其给出针对每个实例为正类的概率，那么通过设定一个阈值如0.6，概率大于等于0.6的为正类，小于0.6的为负类。对应的就可以算出一组(FPR,TPR),在平面中得到对应坐标点。随着阈值的逐渐减小，越来越多的实例被划分为正类，但是这些正类中同样也掺杂着真正的负实例，即TPR和FPR会同时增大。阈值最大时，对应坐标点为(0,0),阈值最小时，对应坐标点(1,1)。



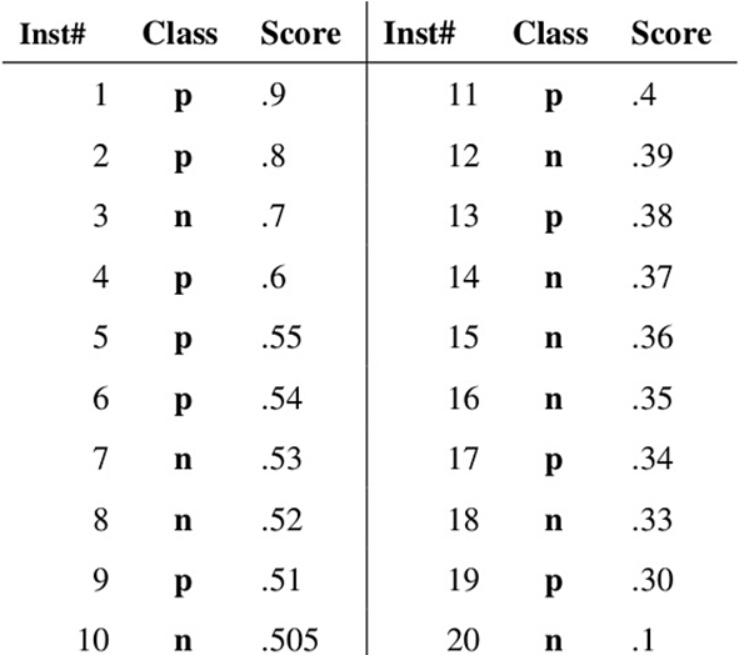
横轴FPR:1-TNR,1-Specificity，FPR越大，预测正类中实际负类越多。

纵轴TPR：Sensitivity(正类覆盖率),TPR越大，预测正类中实际正类越多。

理想目标：TPR=1，FPR=0,即图中(0,1)点，故ROC曲线越靠拢(0,1)点，越偏离45度对角线越好，Sensitivity、Specificity越大效果越好。

**画ROC曲线：**

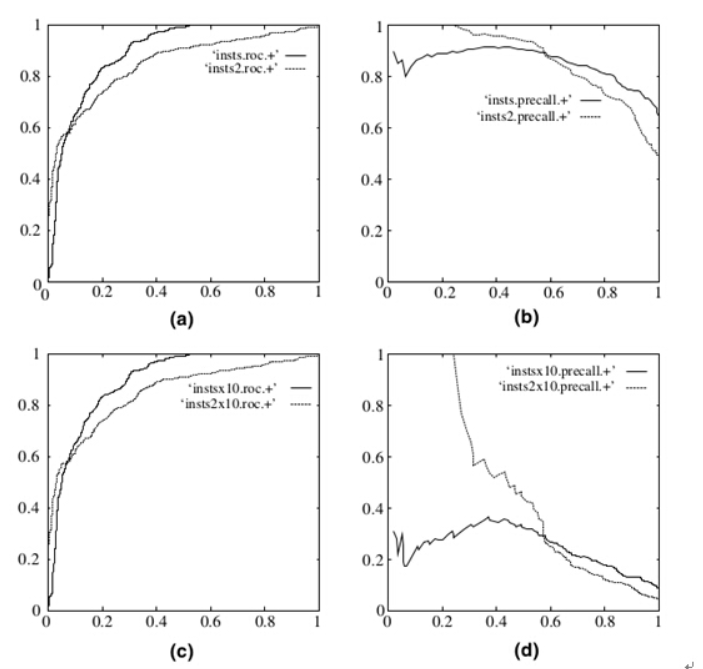
假设已经得出一系列样本被划分为正类的概率，然后按照大小排序，下图是一个示例，图中共有20个测试样本，“Class”一栏表示每个测试样本真正的标签（p表示正样本，n表示负样本），“Score”表示每个测试样本属于正样本的概率。



AUC(Area under Curve)：Roc曲线下的面积，介于0.1和1之间。AUC作为数值可以直观的评价分类器的好坏，值越大越好。首先AUC值是一个概率值，当你随机挑选一个正样本以及负样本，当前的分类算法根据计算得到的Score值将这个正样本排在负样本前面的概率就是AUC值，AUC值越大，当前分类算法越有可能将正样本排在负样本前面，从而能够更好地分类。

**为什么用ROC：**

因为ROC曲线有个很好的特性：当测试集中的正负样本的分布变换的时候，ROC曲线能够保持不变。在实际的数据集中经常会出现样本类不平衡，即正负样本比例差距较大，而且测试数据中的正负样本也可能随着时间变化。下图是ROC曲线和Presision-Recall曲线的对比：



在上图中，(a)和(c)为Roc曲线，(b)和(d)为Precision-Recall曲线。

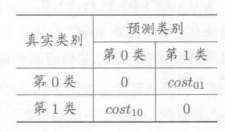
(a)和(b)展示的是分类其在原始测试集(正负样本分布平衡)的结果，(c)(d)是将测试集中负样本的数量增加到原来的10倍后，分类器的结果，可以明显的看出，ROC曲线基本保持原貌，而Precision-Recall曲线变化较大。

**sklearn的调用：**

函数包sklearn.metrics.roc\_curve

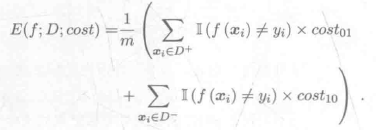
1. 代价敏感错误率与代价曲线

**非均等代价**：不同错误类型造成的损失不同，权重用两个cost调节



在非均等代价下，不是简单地最小化错误次数，而是最小化总体代价

**代价敏感错误率**：



**代价曲线：**

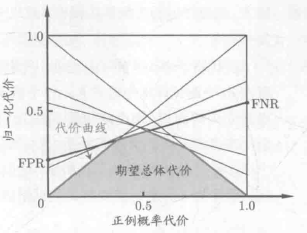
横轴：正例概率代价，取值范围[0,1]



纵轴：归一化代价，取值范围[0,1]



其中，FPR是假正例率，FNR=1-TPR是假反例率



1. 偏差与方差

偏差：期望输出与真实标记的差别





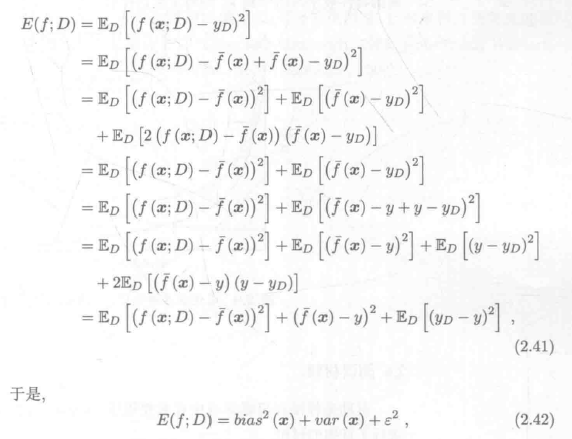
方差：



噪声：

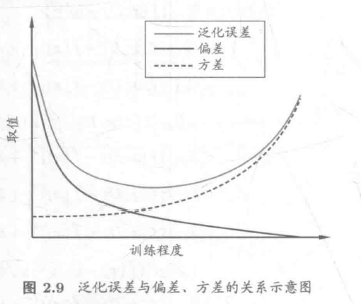


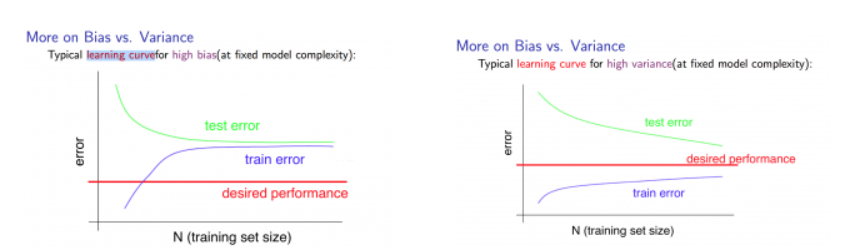
泛化误差：偏差、方差与噪声之和



偏差度量了学习算法的期望预测与真实结果的偏离程度；就是在样本上拟合的好不好，越低越好，相当于越准确

方差度量了同样大小的训练集的变动导致学习性能的变化，相当于数据分布是不是集中，低集中，高分散。





左图对应高偏差（欠拟合），右图对应过拟合。可以看出当模型属于高偏差时，随着样本数据规模增大，性能不会有什么改善，过拟合中的误差则在持续减小。这个很好理解，欠拟合一般是模型比较简单，不能准确的描述数据特征，因此盲目增大数据量是没用的；而过拟合是模型比较复杂，描述数据过于准确了，因此增加一些数据量可以减小过拟合。

过拟合：增大数据规模、减小数据特征数（维数）、增大正则化系数λ

欠拟合：增多数据特征数、添加高次多项式特征、减小正则化系数λ

1. **模型比较**
2. **决策树**
3. 度量指标

信息熵：