**注意：此习题仅供参考，如有错误，欢迎指正！**

习题

**2.1 数据集包含1000个样本，其中500个正例，500个反例，将其划分为包含70%样本的训练集和30%样本的测试集用于留出法评估，试估算共有多少种划分方式。**

答：这是一个排列组合的问题。要保证训练集和测试集中正反例之比为1:1,则在正例中选择30%，在反例中选择30%。（与正例中选择70%和反例中选择70%结果一样）。所以所有的划分方式共有：种。

**2.2 数据集包含100个样本，其中正反例各一半，假定学习算法所产生的模型是将新样本预测为训练样本数较多的类别（训练样本数相同时进行随机猜测），试给出用10折交叉验证法和留一法分别对错误率进行评估所得的结果**

答：如果采用的是10折交叉验证法，那么由于每次的训练样本正反例的数目是一样的，所以结果判断为正和为负的概率也是一样，为50%，所以最后的错误率的期望也是为50%。

如果采用的是留一法：如果留下的是正例，那么对应的反例的样本会比正例的样本多一个，于是会被判定为反例。同样如果留下的是反例，那么会被判定为正例，所以错误率是100%。

**2.3 若学习器A的F1值比学习器B高，试析A的BEP值是否也比B高。**

答：查准率 P = TP / (TP + FP) 查全率 R = TP / (TP + FN)

BEP：平衡点，Break-Event Point，是“查准率 = 查全率”时P-R曲线上的的取值。

F1度量：F1 = 2 \* P \* R / (P + R) = 2 \* TP / (样例总数 + TP - TN) 是基于查准率和查全率的调和平均。

分类器将所有训练样本按自己认为是正例的概率排序，排在越前面的分类器更可能将它判断为正例。按照顺序逐个把样本标记为正，当查准率与查全率相等时，BEP=P=R。分类器的真实输出时在这个寻猎种的选择一个位置，前面的标记为正，后面的标记为负，这时的查准率与查全率用来计算F1值。可以看出有两个同样的BEP值的两个分类器在不同位置截断可能有不同的F1值，所以F1值高的BEP不一定也高。如下：

**1/+ 2/+ 3/+ 4/+ 5/+ 6/- 7/- 8/- 9/- 10/-**

**1/+ 2/+ 3/+ 4/+ 6/- 5/- 7/- 8/- 9/- 10/-**

**1/+ 2/+ 3/+ 4/+ 6/+ 5/- 7/- 8/- 9/- 10/-**

第一行时真实的测试样本的编号与分类，第二三行时两个分类器对所有样本为正例可能行的排序，以及判断的结果。显然这两个分类器有相等的BEP，但是一个F1为0.89，一个为0.8.

**2.4 试述真正例率（TPR）、假正例率（FPR）与查准率（P）、查全率（R）之间的联系。**

答：**查全率**: 真实正例被预测为正例的比例

**真正例率:** 真实正例被预测为正例的比例

显然查全率与真正例率是相等的。

**查准率**:预测为正例的实例中真实正例的比例

**假正例率**: 真实反例被预测为正例的比例

两者并没有直接的数值关系。

**2.5 试证明式(2.22)。**

**式(2.22) AUC=1−lrank**

答：略

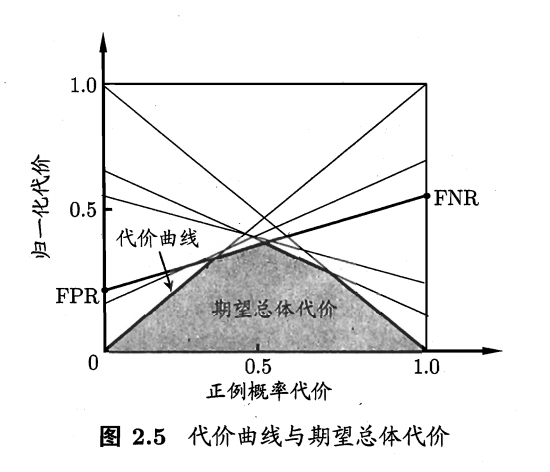
**2.6 试述错误率与ROC曲线之间的关系**

答：ROC曲线的纵坐标为TPR，横坐标为FPR。那么曲线上的每一个点就对应着一个TPR和FPR，也相应的对应着一个错误率：

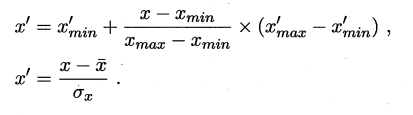
C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\0_HI~(BY`3OWGJJZ}RBSHRY.tmp

**2.7 试证明任意一条ROC曲线都有一条代价曲线与之对应，反之亦然。**

答：代价曲线的绘制：ROC曲线上的每一点对应了代价平面上的一条线段，设ROC曲线上点的坐标为(TPR,FPR)，则可以对应的求出FNR = 1 – TPR，然后在代价平面上绘制一条从（0，FPR）到（1，FNR）的线段，线段下的面积即表示了该条件下的期望的总体代价；如此将ROC曲线上的每一个点转化为代价平面上的一条线段，然后取所有线段的下界，围城的面积即为在所有条件下学习器的期望总体代价。如下图：



**2.8 Min-max规范化和z-score规范化是两种常用的规范化方法。令x和x’分别表示变量在规范化前后的取值，相应的，令xmin和xmax表示规范化钱的最小值和最大值，x’min和x’max表示规范后的最小值和最大值，****分别表示规范化前的均值和标准差，则min-max规范化，z-score规范化分别如式(2.43)和(2.44)所示，试析二者优缺点。**



答：Min-max规范化的方法较为简单，它可以保证规范化之后所有的元素均为正，只有当新加入的数比最大值大或者比最小值小才需要重新计算全部元素。但是它会收到极端值的影响，当存在一个极大（小）值时，会导致其他元素变的特别小（大）。

Z-score规范化对极端元素则有很高的鲁棒性，而且使得所有元素分布在0的周围，每新加入一个元素都要重新的计算方差和均值。

**2.9 试述检验过程。**

答：1）分均值已知与均值未知两种情况，求得卡方检验统计量

2）根据备选假设以及α，求得所选假设对应的拒绝域

3）根据1）中求得的卡方统计量与2）中求得的拒绝域，判断假设成立与否。

**2.10 试述在Friedman检验中使用式(2.34)与(2.35)的区别。**

答：原始Friedman检验要求k较大时，平均序值ri近似于正态分布，均值(k+1)/2，方差(k^2-1)/12。即：C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\ENB_`Z[Q]HSBKM~7AZ]}@ET.tmp

所以C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\K9_Q_Y`)I2@D%L6BBBB11A9.tmp

统计量C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\PAB$SIF$6I[_JVFRB$NN`05.tmp由于k个模型的平均序值ri是有关联的，知道k-1个就能推导出最后一个，所以自由度为k-1，所以前面要乘上(k-1)/k，得到：C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\OXXBILA5N36N9]))ZWQ`@~9.tmp

猜测:由于Friedman统计量只考虑了不同算法间的影响，而没去考虑不同数据集(其他方差)所带来的影响，所以书上说这个Friedman统计量太保守。 (k较大才行，比如k>30)所以对序值表做方差分析：总方差C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\(G3V[NJ$(WR{_2RH[T{@13I.tmp，

自由度 N∗(k−1)算法间方差C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\(7L79RHU9BFRXK[PLL)U3YA.tmp，自由度 k−1

其他方差 SSE = SST − SSA 自由度 (N−1)∗(k−1)做统计量C:\Users\visg\AppData\Roaming\Tencent\QQ\Temp\TempPic\$ZJM4X%8{JY{Q6K`6`~S9LE.tmp，f服从(k−1)和(N−1)∗(k−1)的F分布