1. 评估方法

    1. 留出法：划分成两个互斥的集合，一个训练，一个测试。测试集小，评估结果方差大；训练集小，评估结果偏差大
    2. 交叉验证法：分成k个互斥子集，挨个当测试集，训练k次，取k次的均值。划分很重要，一般随机划分p次，即p次k折交叉验证
    3. 留一法：令交叉验证中的k等于样本数量，即测试机只有一个样本。测试结果往往比较准确，但是m次的计算开销太大
    4. 自助法
        1. 留出法和交叉验证都导致训练集变少，偏差上升。自助法可以减少训练样本不同带来的影响。
        2. 自助法会有放回的对m个样本采样m次，生成一个新的可重复的训练集D'。一个样本在m次采样中一直没有被采样的概率为（1-1/m)^m，极限为1/e=0.368，所以D-D'可以用来当测试集
        3. 自助法在数据集小、难以划分训练/测试集时很有用，也适用于集成学习

2. 性能衡量

    1. TP真正例（预测为正例P，测对了T），FN假反例（预测为反例N，测错了F）
    2. 错误率（error）和精度（accuracy）对偶，和为1
    3. 查准和查全
        1. 查准率（precision，准确率）=TP/(TP+FP)，所有预测正例中，猜对的
        2. 查全率（recal，召回率）=TP/(TP+FN)，所有真正正例中被测对的
        3. 查准和查全一般矛盾，一高一低。PR图就是P为纵轴R为横轴
    4. 平衡点（BEP，break event point），查准等于查全

3. ROC全称是"受试者工作特征" (Receiver Operating Characteristic)

    1. 我们根据学习器的预测结果对样例进行排序，按此顺序逐个把样本作为正例进行预测，每次计算出两个重要量的值，分别以它们为横、纵坐标作图
    2. ROC 曲线的纵轴是"真正例率" (True Positive Rate ，简称 TPR)，横轴是"假正例率" (False PositiveRate ，简称 FPR)
    3. TPR=TP/(TP+FN) 所有真实真例中被预测成正例的
    4. FPR=FP/(TN+FP) 所有真实假例中被预测成正例的
    5. ROC的对角线就是就是“随机猜测”，(0,0)->(0,1)->(1,0)则是理想模型
    6. ROC绘制方法如下：
        1. 给定m+个正例和m-个反例，根据学习器预测结果对样例排序
        2. 把分类阈值设最大，即把所有样例均预测为反例，此时真正例率和假正例率均为0，坐标(0,0)标记一个点
        3. 将分类阐值依次设为每个样例的预测值，即依次将每个样例划分为正例
    7. 一个学习器的 ROC 曲线被另一个写习器的曲线完全"包住"，则可断言后者的性能优于前者。若两个ROC曲线发生交叉，则难断言两者孰优孰劣
    8. AUC（area under roc curve）一般在0.5-1，越大越好。AUC和样本预测的排序质量紧密联系
    9. 可以定义排序loss，每一对正反例颠倒，就loss+1，若相反，则loss+0.5。loss + AUC = 1

4. 泛化误差

    1. 泛化误差可分解为偏差、方差与噪声之和.

    1. 偏差：度量了学习算法的期望预测与真实结果的偏离程度，即模型本身的拟合能力
    2. 方差：同样大小的训练集的变动所导致的学习性能的变化，即抗干扰能力
    3. 噪声：当前任务上，任何学习算法所能达到的期望泛化误差的下界，即刻画了学习问题本身的难度
2. 这说明，泛化性能有学习算法的能力（偏差）、数据的充分性（方差）、学习任务本身的难度（噪声）决定
3. 欠拟合时，偏差主导泛化误差。过拟合时，方差主导泛化误差。

5. 时间复杂度

1. KNN：O(knd)，遍历每个训练观测值，并计算机器学习训练集观测值和新观测值之间的距离 d。时间相对于实例数（n）和维数（d）是线性的。空间复杂度 = O（nd）
2. 逻辑回归：训练时间复杂度= O（nd）空间复杂度 = O（d）
3. SVM：训练时间复杂度 = O（n）注意：如果n大，请尽量避免使用支持向量机（SVM）。运行时复杂度 = O（k * d）K =支持向量数，d =数据的维数
4. 决策树：训练时间复杂度= O(n*log(n)*d) n=训练集中的样本数，d=数据的维数，运行时复杂度= O(树的最大深度) 注意:当我们有大量低维数据时，我们使用决策树。
5. 随机森林：
    1. 训练时间复杂度= O(n*log(n)*d*k) k=决策树的数量。
    2. 注:当我们有大量具有合理特征的数据时。然后我们可以使用多核并行化我们的机器学习模型来训练不同的决策树。
    3. 运行时复杂度= O(树的深度* k)
    4. 空间复杂度= O(树的深度*k)
    5. 注:随机森林相对于其他机器学习算法速度更快。
6. 朴素贝叶斯：训练时间复杂度 = O（n * d）运行时复杂度 = O（c * d）我们必须为每个类'c'检索特征