一 、RF原理解释：

首先，从给定的训练集通过多次随机的可重复的采样得到多个 bootstrap 数据集。接着，对每个 bootstrap 数据集构造一棵决策树，构造是通过迭代的将数据点分到左右两个子集中实现的，这个分割过程是一个搜索分割函数的参数空间以寻求最大信息增量意义下最佳参数的过程。然后，在每个叶节点处通过统计训练集中达到此叶节点的分类标签的直方图经验的估计此叶节点上的类分布。这样的迭代训练过程一直执行到用户设定的最大树深度（随机森林提出者Breiman采用的是ntree=500）或者直到不能通过继续分割获取更大的信息增益为止，网上的代码中作者都是对树的最大深度设置了最大值。

二、函数，参数的中文解释

function model = classRF\_train(X,Y,ntree,mtry, extra\_options)

随机森林中模型的训练

X,表示输入的数据矩阵

Y输出

Ntree 设置的树的数目

Mtry的默认值为

floor(sqrt(size(X,2))，表示不超过矩阵X列数的二次开根值的整数。

extra\_options 包含很多控制RF的项

extra\_options.replace 取值为1或0，默认值为1，表示是否做变量替换

extra\_options.classwt 表示预先知道的类，函数首先得到一个升序排列的标签然后给先前的类同样的排序。

extra\_options.cutoff 只在分类器中使用的一个向量，长度等于类的数目。对类的观察值是取对cutoff投票占的的最大比例的一个。

extra\_options.strata 用于分层抽样

extra\_options.sampsize 样本的长度

extra\_options.nodesize 表示终端节点的最小值，这个参数设置得越大会使更小的树生长，耗时更少。

extra\_options.importance 判断是否需要对预测器的importance进行评估

extra\_options.localImp 决定是否对casewise的重要性度量进行计算

extra\_options.proximity 判别是否计算行之间的距离

extra\_options.oob\_prox 判断是否计算out-of-bag

extra\_options.do\_trace 如果设定为TRUE，当随机森林运行的时候输出更多冗长的数据。如果设置为一些整数，输出每个追踪树。

extra\_options.keep\_inbag 通过树的数目矩阵跟踪每个样本在树上的in-bag。

norm\_votes 统计每一类的投票数

importance 对于分类器来说是一个列数等于类别数加二，第一列计算精度下降值。在ncalss+1列表示所有类平均精度减少值。最后一列表示Gini指数平均减小值。在随机森林用于回归的应用中importance表示的含义又不一样，我们只用到分类的作用，所以对回归的含义不做介绍。

importanceSD 标准差

localImp 包含importance标准化残差测量值的矩阵

ntree 生长的树的数目

mtry 在每一个分裂节点处样本预测器的个数。

Votes 一个矩阵，行表示每一个输入数据点，列表示每一类，记录投票数。

Proximity 如果proximity=TRUE表示随机森林被调用。

Combine 把树结合起来

getTree(rfobj, k=1, labelVar=FALSE)，这个函数的作用是从随机森林中提取树的结构，rfob表示对应的随机森林，k表示要提取的那棵树，, labelVar表示是否是最好的标签用于变量分离和类的预测。在分类器预测中分裂点用整数表示，比如预测器如果有四个参数，分裂点是十三，就表示成（1.0.1.1）因为

if labelVar=TRUE) 一个列数为六，行数等于树中节点总数的矩阵。六列分别为：left daughter，如果是末端左子女节点取值为0。right daughter，如果这个右子节点是末端节点则取值为零。

split var表示哪个变量用来分裂节点，如果是末端节点则取零。split point 表示最佳分裂的位置。

Status 是否是终端节点-1/1. Prediction 对节点的预测 用0表示还没有到达终端节点。

grow(x, how.many, ...) 给现有的树的群体增加新的额外的树。X表示随机森林的类，how.many表示增加到随机森林x中的树的个数。

importance(x, type=NULL, class=NULL, scale=TRUE, ...) x表示随机森林中的类，type表示测量的类型，取值为1的时候表示不准确度，取值为2的时候表示节点不纯度测量。Class 返回值表示哪一个具体的类被测量了。Scale 对测量值进行变换，求他们的标准差。对于分类器来说，节点不纯度是通过Gini系数求解的。返回值为一个矩阵，行表示每个预测变量，列表示不同的importance测量值。

margin(x, ...) 从随机森林分类器中计算或绘制预测器的边缘。

MDSplot(rf, fac, k=2, palette=NULL, pch=20, ...) 随机森林中相似矩阵的多维测量。rf 表示包含近邻成分的随机森林的类。Fac 训练rf的一个响应值。k表示伸缩坐标的维数。Palette用颜色要区分不同的类。

outlier(x, cls=NULL, ...) 基于相似矩阵计算无关测量值。x指的是相似矩阵。Cls表示相似矩阵所在的行属于的类。如果没有给出，则默认为所有数据来自同一个类。

partialPlot(x, pred.data, x.var, which.class, w, plot = TRUE, add = FALSE, n.pt = min(length(unique(pred.data[, xname])), 51), rug = TRUE, xlab=deparse(substitute(x.var)), ylab="", main=paste("Partial Dependence on", deparse(substitute(x.var))), ...)

用图表描述局部的一个变量对类的概率的边缘效应。

plot(x, type="l", main=deparse(substitute(x)), ...)计算错误率

predict(object, newdata, type="response",norm.votes=TRUE, predict.all=FALSE, proximity=FALSE, nodes=FALSE,cutoff, ...) 用随机森林预测测试数据。Object表示随机森林类的对象，由随机森林函数创建。Newdata表示包含新数据的数据帧或者矩阵。可表示响应，概率，票数，指输出的类型。Nodes判断是否是终点。Proximity判断是否需要进行近邻测量。predict.all判断是否保留所有的预测器。

randomForest(x, y=NULL, xtest=NULL, ytest=NULL, ntree=500, mtry=if (!is.null(y) && !is.factor(y)) max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))), replace=TRUE, classwt=NULL, cutoff, strata, sampsize = if (replace) nrow(x) else ceiling(.632\*nrow(x)), nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1, maxnodes = NULL, importance=FALSE, localImp=FALSE, nPerm=1, proximity, oob.prox=proximity, norm.votes=TRUE, do.trace=FALSE, keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE, keep.inbag=FALSE, ...)

data是一个可选的数据帧，包含模型中的变量。Subset是一个指示量表示哪一行应该被使用。

Xtest测试集，

ytest表示对测试集的输出，

Cutoff 是一个长度等于类的数目的向量。被预测的类是投票给cutoff投票率最高的一个类。

treesize(x, terminal=TRUE)表示数的宽度。