# Cart树和随机森林

# CART树

Cart模型是一种决策树模型，它即可以用于分类，也可以用于回归，其学习算法分为下面两步：

1. 决策树生成：用训练数据生成决策树，生成树尽可能大
2. 决策树剪枝：基于损失函数最小化的剪枝，用验证数据对生成的数据进行剪枝。

分类和回归树模型采用不同的最优化策略。Cart回归树使用平方误差最小化策略，Cart分类生成树采用的基尼指数最小化策略。

**Scikit-learn中有两类决策树，他们均采用优化的Cart决策树算法。一个是DecisionTreeClassifier一个是DecisionTreeRegressor回归。**

## 1.1Cart树介绍

分类回归树(CART,Classification And Regression Tree)[算法](http://lib.csdn.net/base/datastructure" \o "算法与数据结构知识库" \t "http://blog.csdn.net/u014568921/article/details/_blank)是一种决策树分类方法。

它采用一种二分递归分割的技术，分割方法采用基于最小距离的基尼指数估计函数，将当前的样本集分为两个子样本集，使得生成的的每个非叶子节点都有两个分支。因此，CART算法生成的决策树是结构简洁的二叉树。

**其核心思想**与ID3和C4.5相同，主要的不同处在于CART在每一个节点上都采用二分法，即每个节点都只能有两个子节点，最后构成的是二叉树。

划分方法：剪枝

**Cart算法步骤：**

1. 决策树的生成：基于训练数据集生成决策树，生成的决策树要尽量大。
2. 决策树的剪枝：用验证数据集对以生成的树进行剪枝并选择最优子树。这时用损失函数最小作为剪枝标准。

## 1.2 Cart树生成

决策树的生成就是递归地构建二叉决策树的过程，对回归树用平方误差最小准则，对分类树用基尼指数最小化原则，进行特征选择，生成二叉树。

## 1.3回归树

### 1.3.1回归树原理

**回归树的原理：**

如果目标是连续变量，则是Regression Tree回归树。CART树是二叉树，不像多叉树那样形成过多的数据碎片。

对于连续变量X（x1…xn）如何处理？

首先将值排序，分别取其两相邻值的平均值点作为分隔点，将树一分成左枝和右枝，不断扫描，进而判断最佳分割点。特征值大于分裂值就走左子树，或者就走右子树。

**下面从数学层面做推导：**

假设X与Y分别为输入和输出变量，并且Y是连续变量，给定训练数据集：



考虑如何生成回归树。

一个回归树对用着一个特征空间的一个划分及在划分单元上的是输出值。假设已输入特征空间划分为M个单元R1，R2.....Rn,并且在每个单元上有一个固定的输出类别Cm，于是我们把回归树表示为：



损失函数：



表示回归树对于训练数据的预测误差，用平方误差最小的准则求解每个单元上的最有输出值。我们考虑特征空间的第m个单元上的的最优值，它是上的所有输入实例对应输出的的均值：（连续值通常的最优值都是取均值）



问题是怎么样对输入空间进行划分，我们采用启发式的方法；选择第j个变量和它去的值s，作为切分变量(splitting variable)和切分点(splitting point)，并定义两个区域：



然后我们需要寻求切分变量j和最有优切分点s，求解如下方程：



对固定输入变量j可以寻找最优切分点s：



遍历所有的输入变量，找到最优的切分变量j，构成一个对(j,s),依次将输入空间划分为两个区域。

接着对每个区域重复上述过程，直到满足停止条件为主。这样就生成了一颗回归树，也称为最小二乘回归树(least square regression tree)。

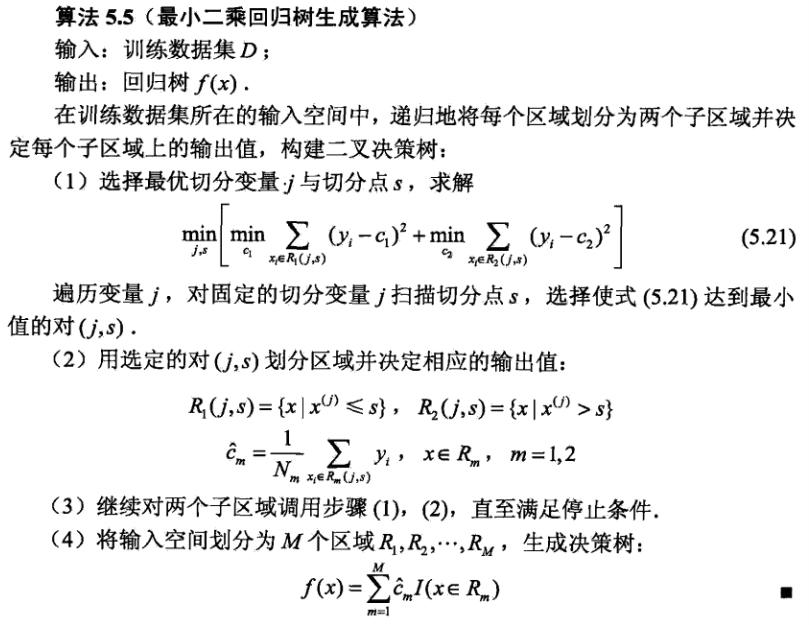
为什么是平均值？

**均方误差的目标函数最优值（取定值）是均值**



**绝对误差的目标函数的最优解是中位数**

### 1.3.2 回归树算法详解



## 1.4分类树

### 1.4.1分类树原理

如果目标变量是离散变量，则是classfication Tree分类树。

分类树是使用树结构算法将**数据分成离散类**的方法。

**（1） 分类树两个关键点：**

➀将训练样本进行递归地划分自变量空间进行建树

➁用验证数据进行剪枝。

**（2）对于离散变量X（x1…xn）处理：**

分别取X变量各值的不同组合，将其分到树的左枝或右枝，并对不同组合而产生的树，进行评判，找出最佳组合。如果只有两个取值，直接根据这两个值就可以划分树。取值多于两个的情况就复杂一些了，如变量年纪，其值有“少年”、“中年”、“老年”，则分别生产{少年，中年}和{老年}，{上年、老年}和{中年}，{中年，老年}和{少年}，这三种组合，最后评判对目标区分最佳的组合。因为CART二分的特性，当训练数据具有两个以上的类别，CART需考虑将目标类别合并成两个超类别，这个过程称为**双化**。这里可以说一个公式,n个属性，可以分出(2^n-2)/2种情况。

1. **例子**



上例是属性有8个，每个属性又有多个离散的值可取。在决策树的每一个节点上我们可以按任一个属性的任一个值进行划分。比如最开始我们按：

1）表面覆盖为毛发和非毛发

2）表面覆盖为鳞片和非鳞片

3）体温为恒温和非恒温

要产生树的左右两个孩子，按哪种划分最好呢？一般我们采用GINI指数，作为划分标准。总体内包含的类别越杂乱，GINI指数就越大（跟熵的概念很相似）。

分类树用基尼指数选择最优特征，同时决定该特征的最优二值切分点。

### 1.4.2变量和最佳切分点选择原则

树的生长，总的原则是，让枝比树更纯，而度量原则是根据不纯对指标来衡量，对于分类树，则用GINI指标、Twoing指标、Order Twoing等；如果是回归树则用，最小平方残差、最小绝对残差等指标衡量。

### （1）利用基尼指数求解划分

**GINI指数**

分类问题中，假设有k个类，样本点属于第i类的概率为pi，则基尼指数定义为：



体温为恒温时包含哺乳类5个、鸟类2个，体温为非恒温时包含爬行类3个、鱼类3个、两栖类2个。



体温为**恒温**时包含哺乳类5个、鸟类2个，则：



体温为**非恒温**时包含爬行类3个、鱼类3个、两栖类2个,则：



### （2）集合的基尼指数

如果样本集合D根据特征A是否取某一可能值a被分割成D1和D2两部分，则在特征A的条件下，集合D的基尼增益定义为：



如果按照“体温为恒温和非恒温”进行划分的话，我们得到GINI的增益：



集合的基尼指数表示集合D的不确定性，基尼指数值越大，样本属于某类的不确定性也就越大，这点与熵相似。我们总希望获得更多信息，减少不确定性。因此，最好的选取特征划分就是使得集合的**基尼指数GINI最小的划分**。

**GINI指数补充：**介于0~1之间的数，0-完全相等，1-完全不等。总体内包含的类别越杂乱，GINI指数就越大（和熵的概念类似）。

### 1.4.3 分类树算法步骤

CART分类树生成算法：

输入：训练数据集D，停止计算的条件；

输出：CART决策树

根据训练数据集，从根节点开始，递归地对每个节点进行以下操作，构建二叉决策树：

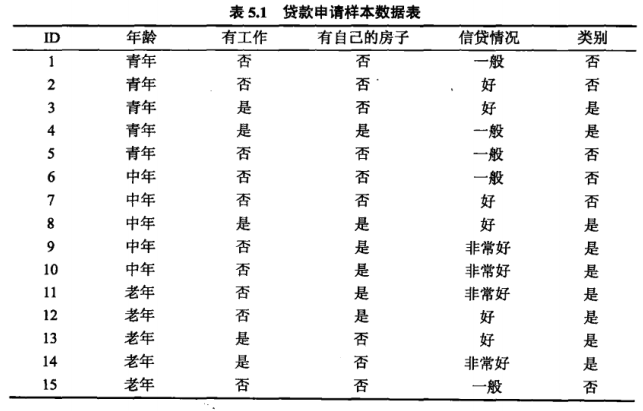
1. 设结点的训练数据集为D，计算现有特征对该数据集的基尼指数，此时对每一个特征A，对其可能取得每个值a，根据样本点对A=a的测试为“是”或“否”将D分割成D1和D2两部分，利用集合的基尼指数公式计算A=a的基尼指数。



1. 在所有可能的特征A(**切分变量**)以及他们的所有可能的**切分点**a中，选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点。依照最优特征和最优切分点，从现结点生成两个子节点，将训练数据集依特征分配到两个子节点中去。
2. 对两个子节点递归地调用（1），（2），直到满足停止条件。
3. 生成CART决策树。

### 1.4.4 分类决策树的例子

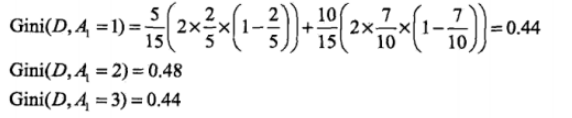
数据集：



应用上述数据集，应用CART算法生成决策树。

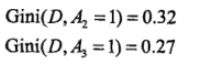
分析：首先计算各特征的基尼指数，选择最优特征以及其最优切分点。我们以A1,A2,A3,A4表示**年龄、有工作、有自己的房子和信贷**4个特征，并以1，2，3表示年龄的值为青年、中年、老年，以1、2表示有工作和有自己的房子的值为是和否，以1、2、3表示信贷情况的值为非常好、好和一般。

求特征A1的基尼指数：



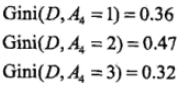
优于Gini(D,A1=1)和Gini(D,A1=3)相等，且最小，所以A1=1和A1=3都可以选作最优切分点。

求特征A2和A3的基尼指数：



由于A2和A3只有一个切分点，所以他们就是最优切分点。

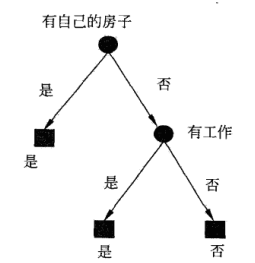
求解特征4的基尼指数：



Gini(D,A4=3)最小，所以A4=3为A4的最优切分点。

在A1，A2，A3，A4几个特征中，Gini(D,A3=1)=0.27最小，所以**选择A3为最优特征**，A3=1为其最优切分点。于是根节点生成两个子节点，一个是叶节点，对另一个结点继续使用以上方法在A1，A2，A3中选择最优特征及其最优切分点，结果是A2=1，依次计算得知，所有结点都是叶节点。

该例子按照CART算法和ID3算法生成的决策树完全一致的。下图我们只拿A3和A2作为最优划分来切分数据集。



## 1.2剪枝

### （1）剪枝

当CART树划分得太细时，会对噪声数据产生过拟合作用。因此我们要通过剪枝来解决。剪枝又分为前剪枝和后剪枝。

前剪枝是指在构造树的过程中就知道哪些节点可以剪掉，于是干脆不对这些节点进行分裂。后剪枝是指构造出完整的决策树之后再来考查哪些子树可以剪掉。

CART剪枝算法从“完全生长”的**决策树的底端剪去一些子树**，使决策树变小(模型变简单)，从而能够对未知数据有更准确的预测。

CART剪枝算法由两步组成：

1.首先从生成算法产生的决策树T0底端开始不断剪枝，直到T0的根节点，形成一个子树序列；

2.然后通过**交叉验证法**在独立的**验证数据集**上对子树序列进行测试，从中选择最优子树。

### （2）误差率增益值

CART树中的每一个非叶子节点的表面**误差率增益值α**(误差增加的速率，越小越好)



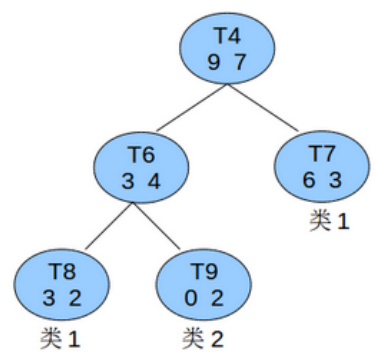
是子树中包含的叶子节点个数。

是节点t的**误差代价**，如果该节点被剪枝：，r(t)是节点t的误差率；p(t)是节点t上的数据占所有数据的比例；

 是子树Tt的误差代价，如果该节点不被剪枝，它等于子树Tt上所有叶子节点的误差代价之和。

**例子讲解：**

有个非叶子节点t4如图所示：



已知所有的数据总共有60条，则节点t4的节点误差代价为：

* r(t)是节点t的误差率
* p(t)是节点t上的数据占所有数据的比例



注意:叶子节点的类定义为**覆盖的样本占多数的类**，即分正确的为多数，分错的为少数。

子树误差代价为：



注： 是子树Tt的误差代价，如果该节点不被剪枝，它等于子树Tt上所有叶子节点的误差代价之和。

以t4为根节点的子树上叶子节点有3个，最终：



找到α值最小的非叶子节点，令其左右孩子为空，即该节点成为叶子节点，即剪枝。

## 1.3 Cart树总结

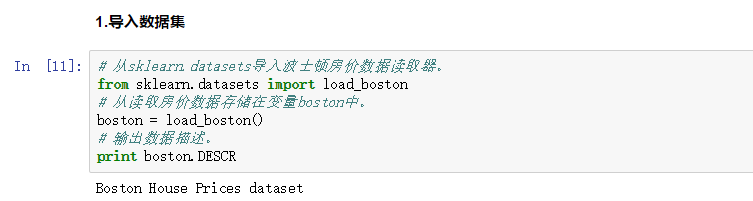
创建分类树递归过程中，CART每次都选择当前数据集中具有**最小Gini信息增益**的特征作为结点划分决策树。ID3算法和C4.5算法虽然在对训练样本集的学习中可以尽可能多地挖掘信息，但其生成的决策树分支、规模较大，CART算法的**二分法**可以简化决策树的规模，提高生成决策树的效率。对于连续特征，CART也是采取和C4.5同样的方法处理。为了避免过拟合(Overfitting)，CART决策树需要**剪枝（后剪枝）**。预测过程当然也就十分简单，根据产生的决策树模型，延伸匹配特征值到最后的叶子节点即得到预测的类别。

# 2 Scikit-learn实战Cart树代码(回归)

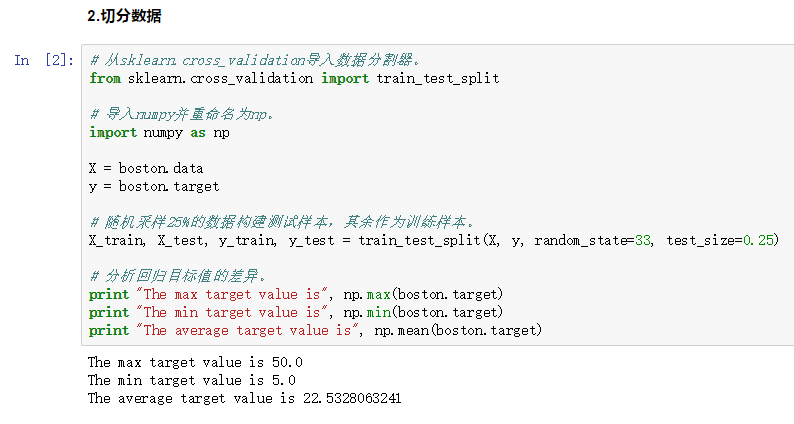
Scikit-learn中有两类决策树，他们均采用优化的Cart决策树算法。一个是DecisionTreeClassifier，一个是DecisionTreeRegressor回归。

这里引用波斯顿房价的回归预测问题。

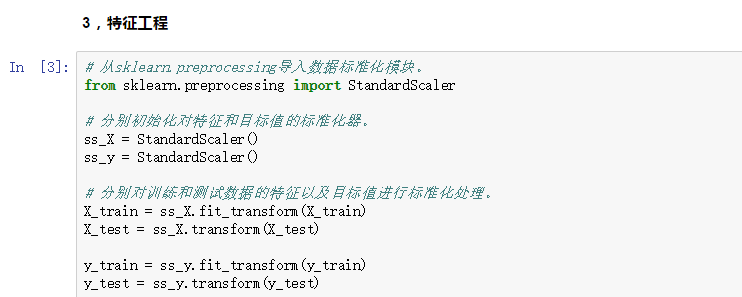
## 2.1导入数据



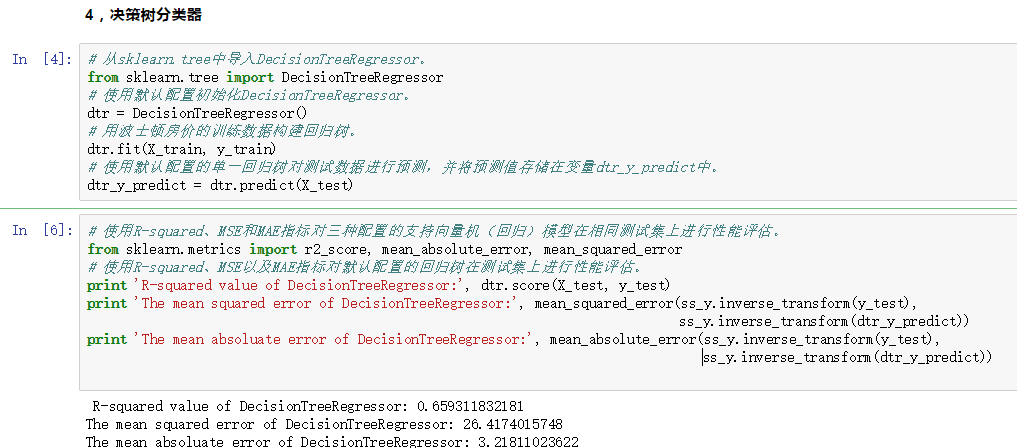
## 2.2切分数据集



## 2.3特征工程



## 2.4决策树分类器



# 3.随机森林（Random Forest Classifier）

## 3.0集成学习简介

随机森林是集成模型中的一种，常言道：“一个篱笆三个桩，一个好汉三个帮”。集成分类模型便是综合考量多个分类器的预测结果，从而做出决策，集成学习分两种：

1. 利用相同的训练数据同时搭建多个独立的分类模型，然后通过投票的方式，以少数服从多数的原则做出最终的分类决策。今天学习的随机森林就是这种方式，即在相同训练数据上同时搭建多颗决策树。

在决策树中学到过一颗标准的决策树是根据每维特征对预测结果的影响程度进行排序，进而决定不同特征从上到下构建分裂节点的顺序；如果这里还按照这种方式随机森林会因为这一策略影响而构建的所有树都一致，从而丧失了多样性。因此随机森林在构建的过程中，每一颗决策树都会放弃这一个固定的算法，转而随机选取特征。

（2）按一定的次序搭建多个分类模型。这些模型之间彼此存在依赖关系。一般后一个模型的加入都需要对现有的集成模型有一定贡献，进而不断提高更新过后的集成模型性能，并**借助多个弱分类器搭建出强分类器**。代表行的有Bossting（AdaBoost）算法。该算法与第一种的随机森林主要区别在于每一颗决策树在生成的过程中都会尽可能降低模型在训练集上的拟合或训练误差。

## 3.1算法介绍

随机森林就是建立很多决策树，组成一个决策树的“森林”，通过多棵树投票来进行决策。这种方法能够有效地提高对新样本的分类准确度。

随机森林在以决策树为基学习器构建**Bagging集成（样本的随机选取）**的基础上，进一步在决策树的训练过程中引入**随机属性**选择。具体来说，传统决策树在选择划分属性时是在当前节点的属性集合（假设有d个属性）中选择一个最优属性；而在RF随机森林中，对基决策树的每个节点，先从该节点的属性集合中随机选择一个包含K个属性的子集，然后在从这个子集中选择一个最优属性用于划分。K=d就是传统决策树，K=1则是随机选取一个属性用于划分，一般情况。

## 3.2随机森林的步骤

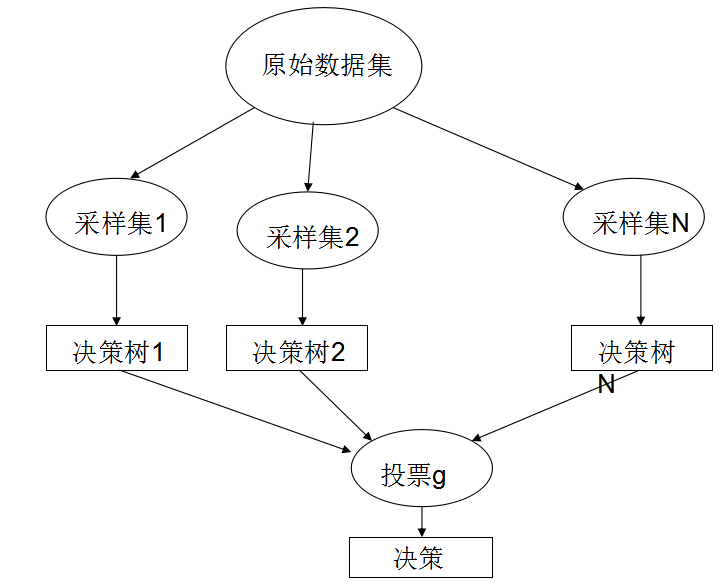
首先，对样本数据进行有放回的抽样，得到多个样本集。具体来讲就是每次从原来的N个训练样本中有放回地随机抽取N个样本(包括可能重复样本)。

然后，从候选的特征中随机抽取m个特征，作为当前节点下决策的备选特征，从这些特征中选择最好地划分训练样本的特征。用每个样本集作为训练样本构造决策树。单个决策树在产生样本集和确定特征后，使用CART算法计算，不剪枝。

最后，得到所需数目的决策树后，随机森林方法对这些树的输出进行投票，以得票最多的类作为随机森林的决策。

说明：

（1）随机森林的方法即对训练样本进行了采样，又对特征进行了采样，充分保证了所构建的每个树之间的独立性，使得投票结果更准确。



（2）随机森林的随机性体现在每棵树的训练样本是随机的，树中每个节点的分裂属性也是随机选择的。有了这2个随机因素，即使每棵决策树没有进行剪枝，随机森林也不会产生过拟合的现象。

随机森林中有两个可控制参数：

1. 森林中树的数量（一般选取值较大）
2. 抽取的属性值m的大小。

## 3.3随机森林的优点

随机森林简单、容易实现、计算开销小，被誉为“代表集成学习计数水平的方法”。可以看出随机森林只是对Bagging做了很小的改动。Bagging的多样性只是体现在样本的随机性，随机森林的基学习器的多样性不仅来自于样本的随机性，还来自于属性的随机性。随机森林随着学习器数目的增加，随机森林通常会收敛到更低的泛化误差。

（1）分类结果更加准确

（2）可以处理高维度的属性，并且不用做特征选择

（3）即使有很大部分数据遗失，仍可以维持高准确度

（4）学习过程快速

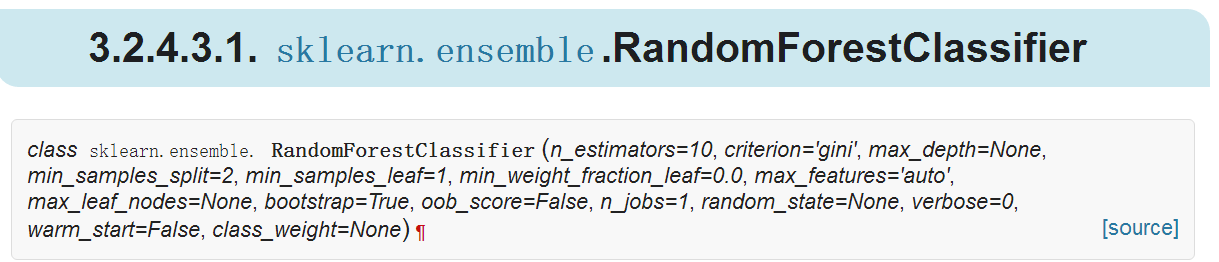
（5）在训练完成后，能够给出哪些属性比较重要

（6）容易实现并行化计算

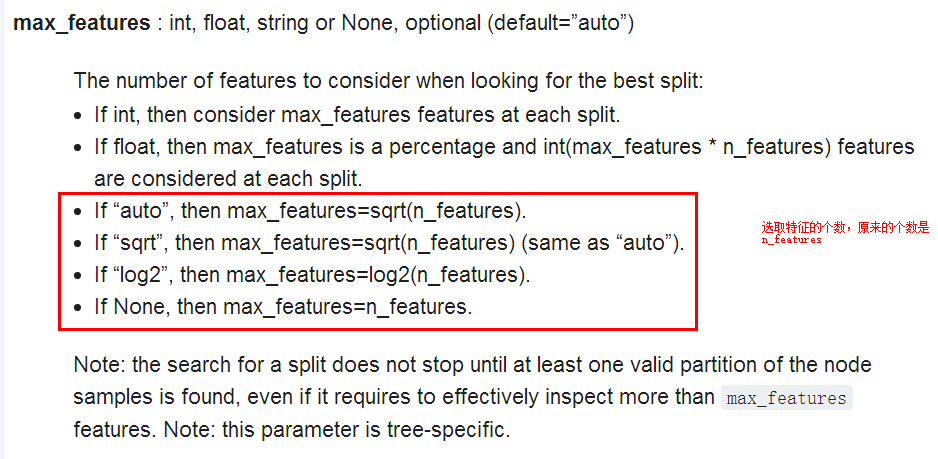
（7）在训练过程中，能够检测到属性之间的相互影响

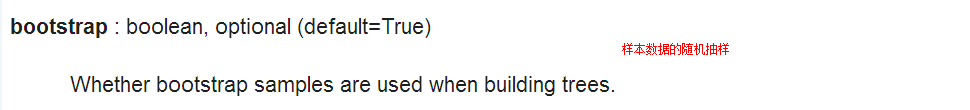
## 3.4随机森林的API

<http://scikit-learn.org/0.17/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier>

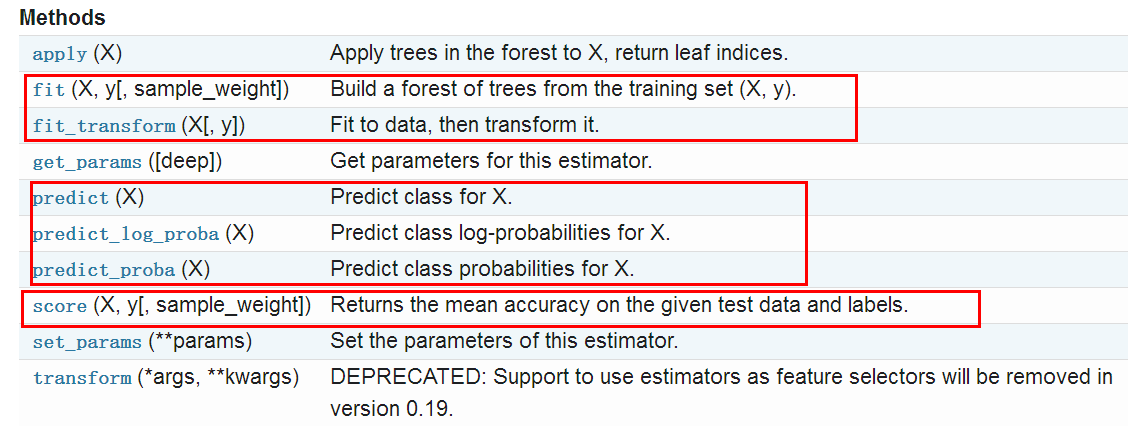


反应随机性的参数：





方法：

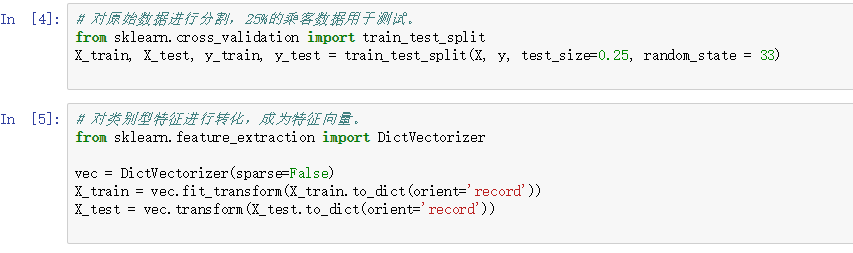


# 4.随机森林预测泰坦尼克号获救人员

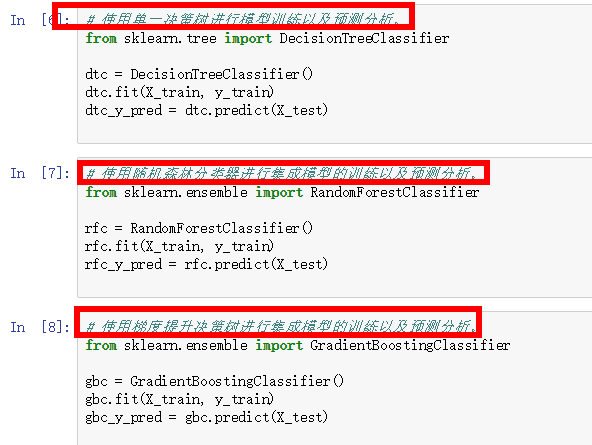
### 4.1导包并选取特征



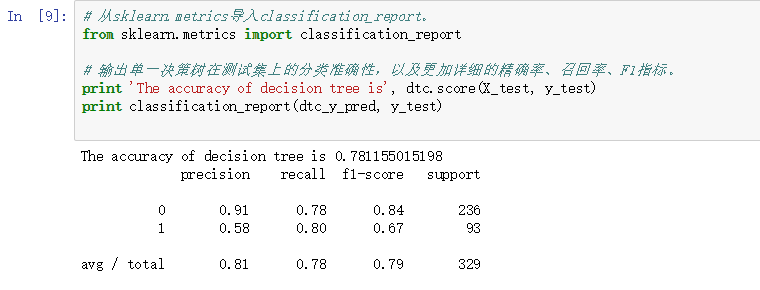
### 4.2切分数据及特征处理

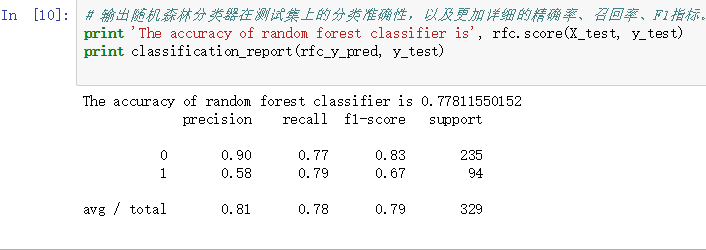


### 4.3三种分类器训练及预测



### 4.4三种分类器性能评估





# 5.总结

## 5.1Cart树

简称：分类和回归树

ID3：信息增益作为建立树的标准

C4.5:信息增益率为建立树的标准

Cart树：以基尼系数为建树标准（分类）

使用MSE（平方误差最小原则）（回归）

Cart树是基于二叉树来划分我们的特征

Python的sklearn中：

内部实现均采用Cart树的方式

分类：DecisionTreeClassfier 回归：DecisionTreeRegression

## 5.2随机森林

集成学习：两种（1）串行方式：AdaBoost和Boosting（2）并行方式：随机森林

随机森林：若干颗决策树构成的森林

注意：这里的决策树和我们之前单颗决策树是有差别的，这里我们采用了两个随机性：（1）特征（属性）的随机性m=log2 n（2）样本的随机性Bagging

随机森林有两个重要的变量：

1. 随机森林中的树的个数（越多越精确）
2. 特征选择的个数m=log2 n（2）

SKlearn中也实现了随机森林：

from sklearn.ensemble import RandomForestClassifier