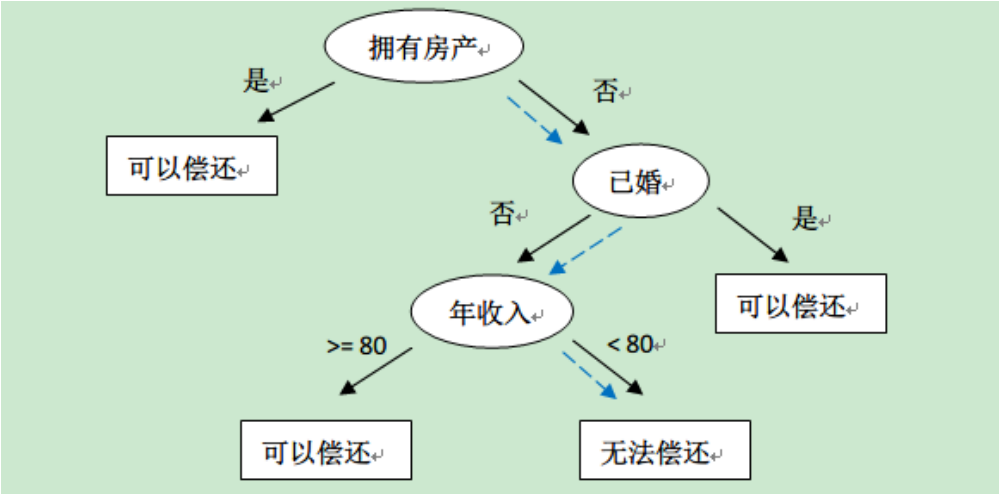
分类决策树

原理

决策树（Decision Tree）是一种简单但是广泛使用的分类器。通过训练数据构建决策树，对未知的数据进行分类。如何预测，先看看下面的数据表格：

ID	拥有房产	婚姻情况	年收入：千元	无法偿还债务
1	是	单身	125	否
2	否	已婚	100	否
3	否	单身	70	否
4	是	已婚	120	否
5	否	离婚	95	是
6	否	已婚	60	否
7	是	离婚	220	否
8	否	单身	85	是
9	否	已婚	75	否
10	否	单身	90	是

上表根据历史数据，记录已有的用户是否可以偿还债务，以及相关的信息。通过该数据，构建的决策树如下：



如新来一个用户：无房产，单身，年收入 55K，那么根据上面的决策树，可以预测他无法偿还债务（蓝色虚线路径）。从上面的决策树，还可以知道是否拥有房产可以很大的决定用户是否可以偿还债务，对借贷业务具有指导意义。

决策树构建的基本步骤如下：

1. 开始所有记录看作一个节点
2. 遍历每个变量的每一种分割方式，找到最好的分割点
3. 分割成两个节点 N1 和 N2

4. 对 N1 和 N2 分别继续执行 2-3 步，直到每个节点足够“纯”为止

构建决策树的变量可以有两种：

1） 连续型：如前例中的“年收入”。用“>=”，“>”，“<”或“<=”作为分割条件（排序后，利用已有的分割情况，可以优化分割算法的时间复杂度）。

2） 分类型：如前例中的“婚姻情况”，使用“=”来分割。

如何评估分割点的好坏？如果一个分割点可以将当前的所有节点分为两类，使得每一类都很“纯”，也就是同一类的记录较多，那么就是一个好分割点。比如上面的例子，“拥有房产”，可以将记录分成了两类，“是”的节点全部都可以偿还债务，非常“纯”；“否”的节点，可以偿还贷款和无法偿还贷款的人都有，不是很“纯”，但是两个节点加起来的纯度之和与原始节点的纯度之差最大，所以按照这种方法分割。构建决策树采用贪心算法，只考虑当前纯度差最大的情况作为分割点。

纯度计算

前面讲到，决策树是根据“纯度”来构建的，如何量化纯度呢？这里介绍三种纯度计算方法。如果记录被分为 n 类，每一类的比例 P(i)=第 i 类的数目/总数目。还是拿上面的例子，10 个数据中可以偿还债务的记录比例为 P(1) = 7/10 = 0.7，无法偿还的为 P(2) = 3/10 = 0.3，N = 2。

Gini 不纯度：

$$Gini = 1 - \sum_{i=1}^{n} P(i)^2$$

熵（Entropy）：

$$Entropy = -\sum_{i=1}^{n} P(i) * \log_2^{P(i)}$$

错误率：

$$Error = 1 - \max\{p(i) | \ i \text{ in } [1, n]\}$$

上面的三个公式均是值越大，表示越 “不纯”，越小表示越“纯”。三种公式只需要取一种即可，对最终分类准确率的影响并不大，一般使用熵公式。

纯度差，也称为信息增益（Information Gain），公式如下：

$$\Delta = I(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} * I(v_j)$$

其中，I 代表不纯度（也就是上面三个公式的任意一种），K 代表分割的节点数，一般 K = 2。vj 表示子节点中的记录数目。上面公式实际上就是当前节点的不纯度减去子节点不纯度的加权平均数，权重由子节点记录数与当前节点记录数的比例决定。

停止条件

决策树的构建过程是一个递归的过程，所以需要确定停止条件，否则过程将不会结束。一种最直观的方式是当每个子节点只有一种类型的记录时停止，但是这样往往会使得树的节点过多，导致过度拟合（Overfitting）。另一种可行的方法是当前节点中的记录数低于一个最小的阀值，那么就停止分割，将 max(P(i))对应的分类作为当前叶节点的分类。

过度拟合

采用上面算法生成的决策树在事件中往往会导致过度拟合。也就是该决策树对训练数据可以得到很低的错误率，但是运用到测试数据上却得到非常高的错误率。过度拟合的原因有以下几点：

噪音数据：训练数据中存在噪音数据，决策树的某些节点有噪音数据作为分割标准，导致决策树无法代表真实数据。

缺少代表性数据：训练数据没有包含所有具有代表性的数据，导致某一类数据无法很好的匹配，这一点可以通过观察混淆矩阵（Confusion Matrix）分析得出。

多重比较（Mulitple Comparition）：举个列子，股票分析师预测股票涨或跌。假设分析师都是靠随机猜测，也就是他们正确的概率是 0.5。每一个人预测 10 次，那么预测正确的次数在 8 次或 8 次以上的概率为$(C_{10}^{8}+C_{10}^{9}+C_{10}^{10})/2^{10} = 0.0547$，只有 5%左右，比较低。但是如果 50 个分析师，每个人预测 10 次，选择至少一个人得到 8 次或以上的人作为代表，那么概率为$(C_{10}^{8}+C_{10}^{9}+C_{10}^{10})/2^{10} = 0.0547$，概率十分大，随着分析师人数的增加，概率无限接近 1。但是，选出来的分析师其实都是非专业的，他对未来的预测不能做任何保证。上面这个例子就是多重比较。这一情况和决策树选取分割点类似，需要在每个变量的每一个值中选取一个作为分割的代表，所以选出一个噪音分割标准的概率是很大的。

优化方案 1：修剪枝叶

决策树过度拟合往往是因为太过“茂盛”，也就是节点过多，所以需要裁剪（Prune Tree）枝叶。裁剪枝叶的策略对决策树正确率的影响很大。主要有两种裁剪策略。

前置裁剪：在构建决策树的过程时，提前停止。那么，会将切分节点的条件设置的很苛刻，导致决策树很短小。结果就是决策树无法达到最优。实践证明此策略无法得到较好的结果。

后置裁剪：决策树构建好后，然后才开始裁剪。采用两种方法：1）用单一叶节点代替整个子树，叶节点的分类采用子树中最主要的分类；2）将一个子树完全替代另外一颗子树。后置裁剪有个问题就是计算效率，有些节点计算后就被裁剪了，导致有点浪费。

优化方案 2：K-Fold Cross Validation

首先计算出整体的决策树 T，叶节点个数记作 N，设 i 属于[1,N]。对每个 i，使用 K-Fold Validataion 方法计算决策树，并裁剪到 i 个节点，计算错误率，最后求出平均错误率。这样可以用具有最小错误率对应的 i 作为最终决策树的大小，对原始决策树进行裁剪，得到最优决策树。

优化方案 3：Random Forest

Random Forest 是从训练数据随机地重抽样，计算出许多决策树，形成了一个森林。然后用这个森林对未知数据进行预测，选取投票最多的分类。实践证明，此算法的错误率

得到了经一步的降低。这相当于“三个臭皮匠顶一个诸葛亮”。一颗树预测正确的概率可能不高，但是集体预测正确的概率却很高。

本模块功能：

给定变量 Y（分布类型可以是 gaussian, binomial, poisson, Survival），如果 Y 是生存状态数据，需要给定时间变量，一组自变量,可以是连续型与分类型，本模块自动进行分类决策树分析，自动采用复杂度损失修剪（CP）的修剪方法，根据分裂到每一层，CP 是多少，平均相对误差是多少（“xerror”），以及标准误差(“xstd”)，平均相对误差=xerror±xstd，选择具有 xerror 小（次小）的 CP。自动对 Y 缺失的数据进行预测。判别结果输出到数据文件中。

例 1：R rpart 程序包练习数据 stagec (146 patients with stage C prostate cancer, from a study exploring the prognostic value of flow cytometry. pgtime:Time to progression or last follow-up (years); pgstat: 1=progression observed, 0=censored; age:age in years; eet: early endocrine therapy, 1 = no, 2 = yes; g2:percent of cells in G2 phase found by flow cytometry; grade:grade of the tumor, Farrow system; gleason: grade of the tumor, Gleason system; ploidy:the ploidy status of the tumor (diploid, tetraploid, and aneuploidy), from flow cytometry.（下载数据：http://www.empowerstats.com/empowerStats/exdata/stagec.xls）分类决策树分析输入界面如下：

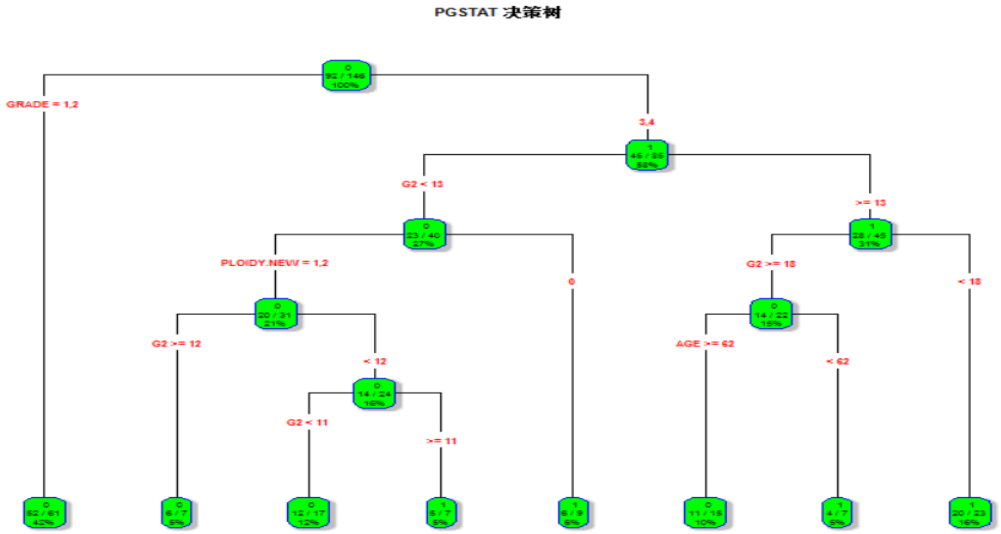
输出结果：

PGSTAT~AGE+EET+G2+GRADE+GLEASON+PLOIDY.NEW

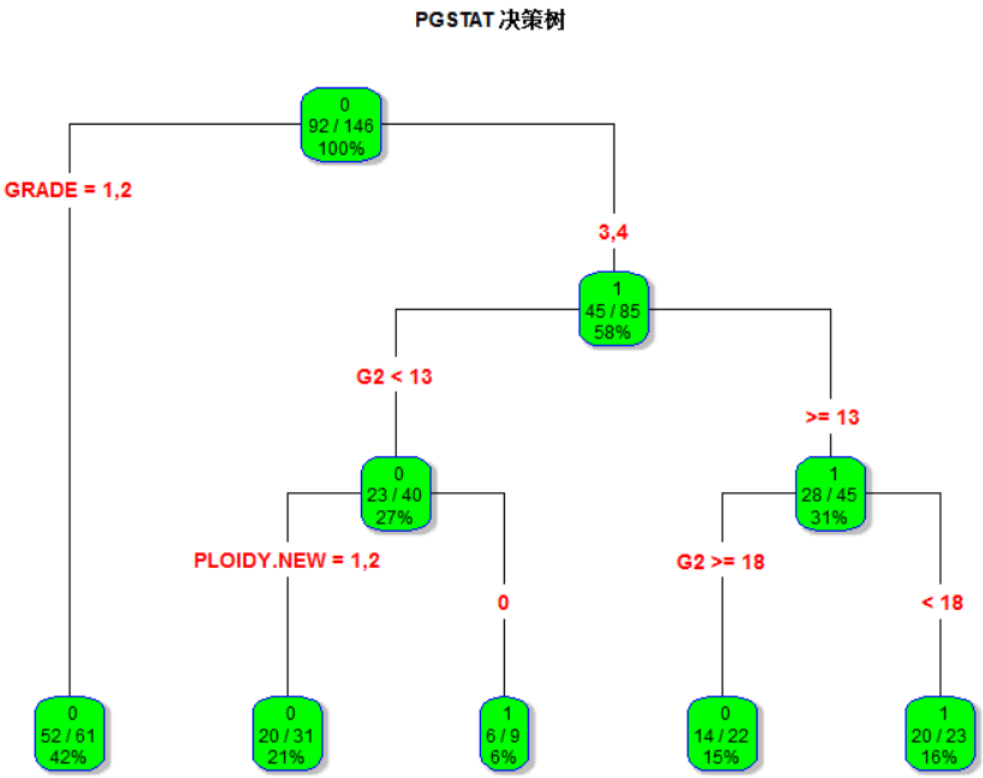
Variable	Importance
G2	29.7
GRADE	27.8
GLEASON	19.9
PLOIDY.NEW	13.4
AGE	7.3
EET	2

```
Recursive partitioning analysis
n= 146
node), split, n, loss, yval, (yprob)
      * denotes terminal node
 1) root 146 54   0 (0.6301370 0.3698630)
   2) GRADE=1,2 61  9   0 (0.8524590 0.1475410) *
   3) GRADE=3,4 85 40   1 (0.4705882 0.5294118)
     6) G2< 13.2 40 17   0 (0.5750000 0.4250000)
      12) PLOIDY.NEW=1,2 31 11   0 (0.6451613 0.3548387)
        24) G2>=11.845 7  1   0 (0.8571429 0.1428571) *
        25) G2< 11.845 24 10   0 (0.5833333 0.4166667)
          50) G2< 11.005 17  5   0 (0.7058824 0.2941176) *
          51) G2>=11.005 7  2   1 (0.2857143 0.7142857) *
      13) PLOIDY.NEW=0 9  3   1 (0.3333333 0.6666667) *
     7) G2>=13.2 45 17   1 (0.3777778 0.6222222)
      14) G2>=17.91 22  8   0 (0.6363636 0.3636364)
        28) AGE>=62.5 15  4   0 (0.7333333 0.2666667) *
        29) AGE< 62.5 7  3   1 (0.4285714 0.5714286) *
      15) G2< 17.91 23  3   1 (0.1304348 0.8695652) *

Classification tree:
rpart(formula = formula(fml), data = wdtmp, method = mthi)

Variables actually used in tree construction:
[1] AGE        G2         GRADE      PLOIDY.NEW
Root node error: 54/146 = 0.36986
n= 146

        CP nsplit rel error  xerror    xstd
1 0.104938      0   1.00000 1.00000 0.10802
2 0.055556      3   0.68519 0.96296 0.10715
3 0.027778      4   0.62963 0.90741 0.10566
4 0.018519      6   0.57407 0.90741 0.10566
5 0.010000      7   0.55556 0.94444 0.10668

 cp= 0.0277777777777778
Prune tree analysis
n= 146
node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 146 54   0 (0.6301370 0.3698630)
```

```
2) GRADE=1,2 61  9   0 (0.8524590 0.1475410) *
3) GRADE=3,4 85 40   1 (0.4705882 0.5294118)
  6) G2< 13.2 40 17   0 (0.5750000 0.4250000)
   12) PLOIDY.NEW=1,2 31 11   0 (0.6451613 0.3548387) *
   13) PLOIDY.NEW=0 9  3   1 (0.3333333 0.6666667) *
  7) G2>=13.2 45 17   1 (0.3777778 0.6222222)
   14) G2>=17.91 22  8   0 (0.6363636 0.3636364) *
   15) G2< 17.91 23  3   1 (0.1304348 0.8695652) *
```



修剪后的决策树：



例 2：上例中引进时间变量 PGTIME 做生存状态决策树分析。

输入界面：

输出结果：

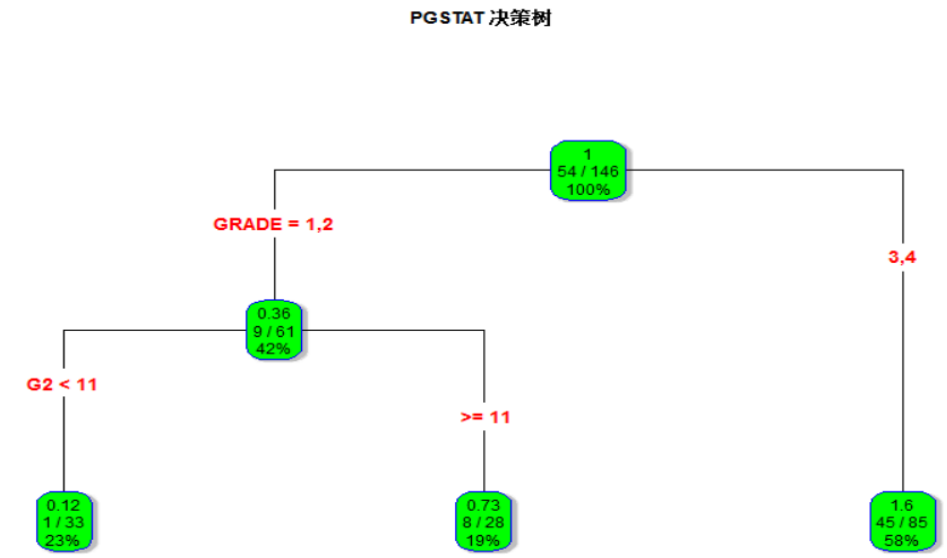
```
Surv(PGTIME,PGSTAT)~AGE+EET+G2+GRADE+GLEASON+PLOIDY.NEW
```

Variable	Importance
GRADE	27.2
GLEASON	25.4
G2	22
PLOIDY.NEW	13.3
AGE	10.6
EET	1.5

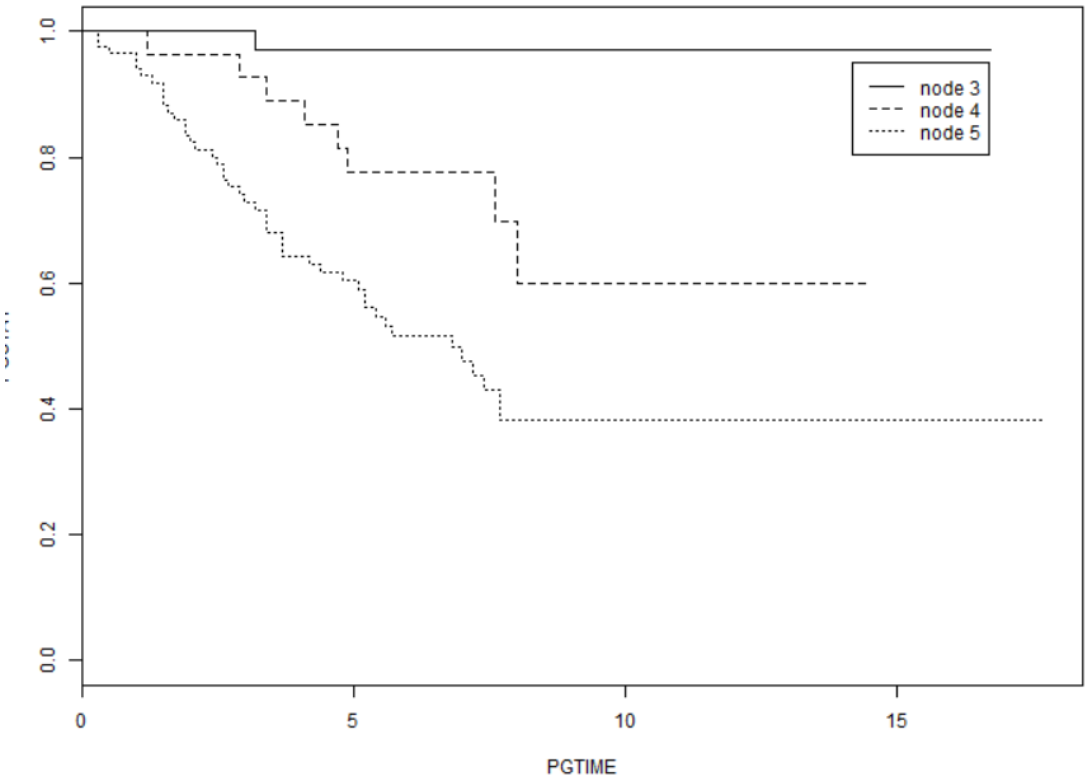
```
Recursive partitioning analysis
n= 146
node), split, n, deviance, yval
      * denotes terminal node
 1) root 146 192.111100 1.0000000
   2) GRADE=1,2 61  44.799010 0.3634439
     4) G2< 11.36 33   9.117405 0.1229835 *
     5) G2>=11.36 28  27.602190 0.7345610
```

```
      10) GLEASON< 5.5 20  14.297110 0.5304115 *
      11) GLEASON>=5.5 8  11.094650 1.3069940 *
   3) GRADE=3,4 85 122.441500 1.6148600
     6) AGE>=56.5 75 103.062900 1.4255040
      12) GLEASON< 7.5 50  66.119800 1.1407320
        24) G2< 13.475 24  27.197170 0.8007306 *
        25) G2>=13.475 26  36.790960 1.4570210
          50) G2>=17.915 15  20.332740 0.9789825 *
          51) G2< 17.915 11  13.459010 2.1714480 *
      13) GLEASON>=7.5 25  33.487250 2.0307290
        26) G2>=15.29 10  11.588480 1.2156230 *
        27) G2< 15.29 15  18.939150 2.7053610 *
     7) AGE< 56.5 10  13.769010 3.1822320 *

Survival regression tree:
rpart(formula = formula(fml), data = wdtmp, method = mthi)

Variables actually used in tree construction:
[1] AGE     G2      GLEASON GRADE

Root node error: 192.11/146 = 1.3158
n= 146
        CP nsplit rel error  xerror     xstd
1 0.129460      0   1.00000 1.01119 0.071126
2 0.042056      1   0.87054 0.88835 0.073791
3 0.029200      2   0.82848 0.94849 0.077837
4 0.017989      3   0.79928 1.02987 0.094239
5 0.015406      4   0.78130 1.06191 0.095477
6 0.013354      5   0.76589 1.05927 0.095565
7 0.011506      7   0.73918 1.06702 0.096775
8 0.010000      8   0.72768 1.07575 0.097248

 cp= 0.0291998567529351

 Prune tree analysis
n= 146
node), split, n, deviance, yval
      * denotes terminal node
1) root 146 192.111100 1.0000000
  2) GRADE=1,2 61  44.799010 0.3634439
    4) G2< 11.36 33   9.117405 0.1229835 *
    5) G2>=11.36 28  27.602190 0.7345610 *
  3) GRADE=3,4 85 122.441500 1.6148600 *
```

修剪后的决策树：



按节点分组的 KM 曲线图：



对照决策树：

```
  2) GRADE=1,2 61  44.799010 0.3634439
    4) G2< 11.36 33   9.117405 0.1229835 *
    5) G2>=11.36 28  27.602190 0.7345610 *
  3) GRADE=3,4 85 122.441500 1.6148600 *
```

查出 node3 表示 GRADE=3 或 4，有 85 人；node4 表示 GRADE=1 或 2，G2<11.36，有 33 人；node5 表示 GRADE=1 或 2，G2>=11.36 有 28 人。KM 曲线看出这三组人生存曲线有明显差异。