举例：一个人的年收入和他的年龄Age、性别Gender、教育程度Education、居住地Residence、从事行业Industry的关系如下表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | Age | Gender | Education | Residence | Industry | Salary |
| 1 | 17 | Male | High School | Non-Metro | Services | $10,000 |
| 2 | 27 | Female | Bachelors | Non-Metro | Manufacturing | $40,000 |
| 3 | 32 | Female | Master | Metro | Manufacturing | $120,000 |
| 4 | 50 | Male | Bachelors | Metro | Finance | $110,000 |
| 5 | 29 | Male | Master | Non-Metro | Finance | $180,000 |
| 6 | 35 | Female | High School | Non-Metro | Services | $30,000 |
| 7 | 48 | Female | High School | Metro | Services | $25,000 |
| 8 | 27 | Male | Master | Metro | Manufacturing | $90,000 |
| 9 | 31 | Male | Bachelors | Non-Metro | Services | $100,000 |
| 10 | 44 | Female | Bachelors | Metro | Services | $160,000 |

用随机森林回归算法建立回归森林。

假设森林中树的数目为3，最大树深maxDepth为3，maxBins为4(一个feature的最多分箱数)。

其中Gender, Residence, Industry为无序特征，

因此numbins=[4,2,3,2,3], 对于连续特征来说，每个split有split+1个bin，

因此numSplits=[3,2,3,2,3]。

为连续特征Age选取split，numSplit=3，numExamples=10，则步长为10/4=2.5，Age排序为[(17:1), (27:2), (29:1), (31:1), (32:1), (35:1), (44:1), (48:1), (50:1)]。

计算过程：

index=1,courrentCount=1,targetCount=2.5

previousCount=courrentCount=1,

currentCount = 1+2 = 3,

previousGap= |1-2.5|=1.5

currentGap=|3-2.5|=0.5

index=2

previousCount=courrentCount=3,

currentCount = 3+1 = 4,

previousGap= |3-2.5|=0.5

currentGap=|4-2.5|=1.5

previousGap<currentGap:

splitsBuilder = [27.0]

targetCount = 2.5 +2.5 = 5.0

index=3

previousCount=courrentCount=4,

currentCount = 4+1 = 5,

previousGap= |4-5.0|=1

currentGap=|5-5.0|=0

index=4

previousCount=courrentCount=5,

currentCount = 5+1 = 6,

previousGap= |5-5.0|=0

currentGap=|6-5.0|=1

previousGap<currentGap:

splitsBuilder = [31.0]

targetCount = 5.0 +2.5 = 7.5

index=5

previousCount=courrentCount=6,

currentCount = 6+1 = 7,

previousGap= |6-7.5|=1.5

currentGap=|7-7.5|=0.5

index=6

previousCount=courrentCount=7,

currentCount = 7+1 =8,

previousGap= |7-7.5|=0.5

currentGap=|8-7.5|=0.5

index=7

previousCount=courrentCount=8,

currentCount = 8+1 =9,

previousGap= |8-7.5|=0.5

currentGap=|9-7.5|=1.5

currentGap=|6-5.0|=1

previousGap<currentGap:

splitsBuilder = [44.0]

targetCount = 7.5 +2.5 = 10.0

index=8

previousCount=courrentCount=9,

currentCount = 9+1 =10,

previousGap= |9-10.0|=1.0

currentGap=|10-10.0|=0.0

Splits=[[27.0,31.0,44.0]]

其他特征:

Gender: Male/0, Female/1 无序类别

Education: High School/0, Bachelors/1, Master/2 无序类别

Residence: Non-Metro/0, Metro/1 无序类别

Industry: Services/0, Finance/1, Manufacturing/2 无序类别

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | Age | Gender | Education | Residence | Industry | Salary |
| 1 | 17 | 0 | 0 | 0 | 0 | $10,000 |
| 2 | 27 | 1 | 1 | 0 | 2 | $40,000 |
| 3 | 32 | 1 | 2 | 1 | 2 | $120,000 |
| 4 | 50 | 0 | 1 | 1 | 1 | $110,000 |
| 5 | 29 | 0 | 2 | 0 | 1 | $180,000 |
| 6 | 35 | 1 | 0 | 0 | 0 | $30,000 |
| 7 | 48 | 1 | 0 | 1 | 0 | $25,000 |
| 8 | 27 | 0 | 2 | 1 | 2 | $90,000 |
| 9 | 31 | 0 | 1 | 0 | 0 | $100,000 |
| 10 | 44 | 1 | 1 | 1 | 0 | $160,000 |

为3棵树随机取10个样本(采样比例为1.0)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | Salary | subSampledWeight tree0 | subSampledWeight tree1 | subSampledWeight tree2 |
| 1 | $10,000 | 2 | 2 | 0 |
| 2 | $40,000 | 2 | 0 | 1 |
| 3 | $120,000 | 0 | 0 | 2 |
| 4 | $110,000 | 2 | 1 | 0 |
| 5 | $180,000 | 1 | 0 | 1 |
| 6 | $30,000 | 1 | 2 | 1 |
| 7 | $25,000 | 4 | 0 | 0 |
| 8 | $90,000 | 0 | 0 | 1 |
| 9 | $100,000 | 1 | 1 | 1 |
| 10 | $160,000 | 1 | 0 | 0 |

森林中3棵数的根节点Node0均为全部样本

1. 为第1棵树随机选取onethird(5)=2个特征：Education, Residence

为第2棵树随机选取onethird(5)=2个特征：Gender, Education

为第2棵树随机选取onethird(5)=2个特征：Education, Residence

1. 计算每个箱的impurity(variance)：

第1棵树Node0(全部样本)：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | Education | Residence | Salary |
| 1 | 0 | 0 | $10,000 |
| 1 | 0 | 0 | $10,000 |
| 2 | 1 | 0 | $40,000 |
| 2 | 1 | 0 | $40,000 |
| 4 | 1 | 1 | $110,000 |
| 4 | 1 | 1 | $110,000 |
| 5 | 2 | 0 | $180,000 |
| 6 | 0 | 0 | $30,000 |
| 7 | 0 | 1 | $25,000 |
| 7 | 0 | 1 | $25,000 |
| 7 | 0 | 1 | $25,000 |
| 7 | 0 | 1 | $25,000 |
| 9 | 1 | 0 | $100,000 |
| 10 | 1 | 1 | $160,000 |

Node0: count=14.0, sum=890,000.0, sumSquare=9.9E10

DTStatsAggregator[0]

statsSize: 3(每个bin的统计信息个数 count, sum, sumSquare)

每个featire的numbins=[3,2],

每个feature的偏移量featuresOffsets: [0, 9, 15]

allStats: double[15]

Education=0: count=7.0, sum=150,000.0, sumSquare=3.6E9

Education=1: count=6.0, sum=560,000.0, sumSquare=6.3E10

Education=2: count=1.0, sum=180,000.0, sumSquare=3.24E10

Residence=0: count=7.0, sum=410,000.0, sumSquare=4.67E10

Residence=1: count=7.0, sum=480,000.0, sumSquare=5.23E10

选取最佳划分bestSplit: gain最小。

gain = impurity - leftWeight \* leftImpurity - rightWeight \* rightImpurity

leftWeight = leftCount / totalCount

rightWeight = rightCount / totalCount

impurity = (sumSquares - (sum \* sum) / count) / count

以Education划分：

CategoriesSortedByCentriod(0,21428) (1,93333) (2,180000.)

Education=0：

左子树：

VarianceAggregator(cnt = 7.0, sum = 150000.0, sum2 = 3.6E9)

右子树：

VarianceAggregator(cnt=7.0, sum = 740000.0, sum2 = 9.54E10)

gain = 1.7760204081632655E9, impurity = 3.0301020408163266E9, left impurity = 5.510204081632655E7, right impurity = 2.453061224489796E9

Education=2：

gain = 1.0427394034536896E9, impurity = 3.0301020408163266E9, left impurity = 2.1402366863905323E9, right impurity = 0.0

... ...

第2棵树：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | Education | Gender | Salary |
| 1 | 0 | 0 | $10,000 |
| 1 | 0 | 0 | $10,000 |
| 4 | 1 | 1 | $110,000 |
| 6 | 0 | 0 | $30,000 |
| 6 | 0 | 0 | $30,000 |
| 9 | 1 | 1 | $100,000 |

Node0: count=6.0, sum=290,000.0, sumSquare=2.41E10

Education=0: count=4.0, sum=80,000.0, sumSquare=2.0E9

Education=1: count=2.0, sum=210,000.0, sumSquare=2.21E10

Education=2: count=0.0, sum=0.0, sumSquare=0.0

Gender=0: count=4.0, sum=230,000.0, sumSquare=2.23E10

Gender=1: count=2.0, sum=60,000.0, sumSquare=1.8E9

以Education特征划分：

CategoriesSortedByCentriod：

(0,20000.0) (1,105000.0) (2,1.7976931348623157E308)

以Education=0划分时：

根节点：

impurity =1.0083333333333334E10/6.0=1.6055555555555556E9

左子树：VarianceAggregator(cnt = 4.0, sum = 80000.0, sum2 = 2.0E9)

left impurity=4.0E8/4.0= 1.0E8

右子树：VarianceAggregator(cnt = 2.0, sum = 210000.0, sum2 = 2.21E10)

right impurity=5.0E7/2.0=2.5E7

leftWeight = leftCount / totalCount = 0.6666667

rightWeight = rightCount / totalCount = 0.33333333

gain = impurity - leftWeight \* leftImpurity - rightWeight \* rightImpurity

= 1.6055555555555556E9

以Education=2划分时：

gain = -1.7976931348623157E308, impurity = 1.6805555555555556E9, left impurity = -1.0, right impurity = -1.0

以Education=1划分时：

gain = 1.6805555555555558E8, impurity = 1.6805555555555556E9, left impurity = 0.0, right impurity = 2.26875E9

以Gender划分：

CategoriesSortedByCentriod：

(0,57500.0)(1,30000.0)

以Gender=0划分时：

gain = 1.6805555555555558E8, impurity = 1.6805555555555556E9, left impurity = 0.0, right impurity = 2.26875E9

... ...

第3棵树：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | Education | Residence | Salary |
| 2 | 1 | 0 | $40,000 |
| 3 | 2 | 1 | $120,000 |
| 3 | 2 | 1 | $120,000 |
| 5 | 2 | 0 | $180,000 |
| 6 | 0 | 0 | $30,000 |
| 8 | 2 | 1 | $90,000 |
| 9 | 1 | 0 | $100,000 |

Node0: count=7.0, sum=680,000.0, sumSquare=8.18E10

Education=0: count=1.0, sum=30,000.0, sumSquare=9.0E8

Education=1: count=2.0, sum=140,000.0, sumSquare=1.16E10

Education=2: count=4.0, sum=510,000.0, sumSquare=6.93E10

Residence=0: count=4.0, sum=350,000.0, sumSquare=4.49E10

Residence=1: count=3.0, sum=330,000.0, sumSquare=3.69E10

以Education划分：

CategoriesSortedByCentriod：

(0,30000.0)(1,70000.0)(2,127500.0)

Education=2

gain = 1.228741496598639E9, impurity = 2.2489795918367343E9, left impurity = 9.555555555555553E8, right impurity = 1.06875E9

... ...

以Residence划分:

CategoriesSortedByCentriod：

(0,87500.0)(1,110000.0)

Residence=0

gain = 1.2397959183673443E8, impurity = 2.2489795918367343E9, left impurity = 3.56875E9, right impurity = 2.0E8

... ...

最佳划分：Education=2

Node0(全部样本)

Node1

Education=2

Node2

Education!=2

森林中得到的树为：

树1：

Node0

Node1

Education=0

Node2

Education!=0

树2：

Node0

Node1

Education=1

Node2

Education!=1

树3：

Node0(全部样本)

Node1

Education=2

Node2

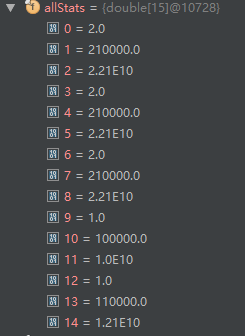
Education!=2

为每个第二层的节点选择features，重复上述过程，得到森林。

如：树2中的Node1。

选择features为Education, Residence

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | Education | Residence | Salary |
| 4 | 1 | 1 | $110,000 |
| 9 | 1 | 0 | $100,000 |



计算按照每种划分的gain：

Residence=0：

gain = 2.5E7, impurity = 2.5E7, left impurity = 0.0, right impurity = 0.0

最终得到森林：

不同的树\*3：

Node0

Node1

Education=1

Node2

Education!=1

110,000

$100,000

Residence=1

Residence=0

Age<31

Age>31

$30,000

10,000