Обучение

ИЦ «Гевисста»

Артем Груздев

Прогнозное моделирование в R и Python

Модуль 3. Построение деревьев решений CART с помощью пакета R rpart

Москва — 2017

СОДЕРЖАНИЕ

Модуль 3 Построение деревьев решений CART с помощью пакета R rpart

Лекция 3.1 Знакомство с методом CART

Алгоритм метода CART был опубликован в 1984 году в книге «Classification and regression trees» («Деревья классификации и регрессии»), авторами которой являлись Лео Брейман (Калифорнийский университет в Беркли), Джером Фридман (Стэнфордский университет), Ричард Олшен (Калифорнийский университет в Беркли) и Чарльз Стоун (Стэнфордский университет).

Как уже ясно из названия, CART включает два вида анализа: деревья классификации для категориальных зависимых переменных и регрессионные деревья для количественных зависимых переменных. Эти виды анализа отличаются в деталях, но используют общий принцип. CART выполняет последовательные бинарные разбиения данных на основе выбранного критерия. В отличие от CHAID, в CART для выбора предиктора расщепления не применяются статистические критерии. Вместо этого в каждом узле при расщеплении данных используется предиктор, обеспечивающий наибольшее улучшение по выбранному критерию. Ключевой элемент в методе CART – это отсечение ветвей дерева, известный под названием «отсечение с минимизацией стоимости-сложности» (minimal cost-complexity pruning). Деревья, построенные с помощью метода CART, имеют тенденцию быть слишком большими, а результаты не повторяются с необходимой степенью устойчивости. Авторы метода пришли к выводу, что если разрешить построение максимально большого дерева, но затем отсечь его ветви, используя более сложный критерий, то в результате будет построено меньшее дерево лучшего качества. Построение дерева с последующим отсечением его ветвей стало основой метода CART.

3.1.1 Описание алгоритма

Алгоритм CART строит дерево, итеративно применяя к каждому узлу, начиная с корневого, процедуры выбора наилучшего расщепления предиктора, выбора наилучшего расщепления узла и остановки. В качестве критерия расщепления алгоритм использует уменьшение неоднородности, для категориальных зависимых переменных – уменьшение меры Джини, для количественных зависимых переменных – уменьшение девианса. Наилучшее расщепление – это расщепление, максимально уменьшающее меру Джини или девианс.

Этап 1. Выбор наилучшего расщепления предиктора

1. Алгоритм начинает с поиска наилучшего расщепления для каждого предиктора. Для количественных и порядковых предикторов алгоритм выполняет сортировку значений в порядке возрастания. Затем алгоритм разбивает предиктор по всем возможным точкам расщепления (разделяющим значениям). Если имеются 6 различных значений возраста, они будут упорядочены и для них будет созданы 5 точек расщепления.
Например, есть значения возраста 18, 35, 20, 16, 11, 10. Они будут упорядочены: 10, 11, 16, 18, 20 и 35. Будет рассмотрено пять расщепляющих значений:
< 11 s ≥ 11
< 16 s ≥ 16
< 18 s ≥ 18
< 20 s ≥ 20
< 35 s ≥ 35

Кроме того, часто применяется стратегия, когда в качестве возможной точки расщепления рассматривается среднее по каждой паре упорядоченных смежных значений. В данном случае точки расщепления будут выглядеть так:

$<10{,}5$ s $\geq 10{,}5$
$<13{,}5$ s $\geq 13{,}5$
<17 s ≥ 17
<19 s ≥ 19
$<27{,}5$ s $\geq 27{,}5$

Именно эта стратегия используется в пакете R `rpart`.
В каждой возможной точке расщепления переменной вся выборка наблюдений может быть разбита на два дочерних узла: левый и правый. Все наблюдения, у которых значение предиктора меньше разделяющего значения, относятся в левый дочерний узел. Все наблюдения, у которых значение предиктора больше или равно разделяющему значению, относятся в правый дочерний узел.
Для номинального предиктора его категории делятся всеми возможными способами на две группы.
2. Алгоритм вычисляет уменьшение неоднородности для каждого варианта расщепления (см. *раздел 3.1.2* «*Неоднородность*»).
3. В качестве наилучшего расшепления предиктора алгоритма выбирается расщепление, лучше всего минимизирующее неоднородность.
Вышеописанные шаги повторяются для всех остальных переменных.

Этап 2. Выбор наилучшего расщепления узла

1. Из наилучших расщеплений предикторов, полученных на первом этапе, алгоритм выбирает расщепление предиктора, максимизирующее критерий расщепления для корневого узла или узла-потомка (проще говоря, из лучших расщеплений выбираем лучшее).
2. Алгоритм расщепляет узел, используя найденное наилучшее расщепление для него, если это позволяют правила остановки. Обратите внимание, что каждый предиктор может неоднократно использоваться для расщепления в ветви дерева. Например, может быть выполнено расщепление по переменной *Возраст* в значении 60 лет, а затем в узле-потомке снова может быть выполнено расщепление по этой переменной. Таким образом, могут моделироваться сложные зависимости между непрерывным предиктором и зависимой переменной, несмотря на то что выполняются только бинарные расщепления.

Этап 3. Остановка

Алгоритм проверяет, нужно ли прекратить построение дерева, в соответствии со следующими правилами остановки.

1. Если узел стал однородным, то есть все наблюдения в узле имеют одинаковые значения зависимой переменной, узел не разбивается.
2. Если при выполнении разбиения уменьшение кроссвалидационной ошибки модели меньше порогового значения, процесс построения дерева останавливается.
3. Если количество наблюдений в родительском узле меньше заданного пользователем минимума наблюдений в родительском узле, узел не разбивается.
4. Если минимальное количество наблюдений в терминальном узле меньше заданного пользователем минимума наблюдений в терминальном узле, узел не разбивается.
5. Если текущая глубина дерева достигает заданной пользователем максимальной глубины дерева, процесс построения дерева останавливается.

ПРИМЕЧАНИЕ

В пакете R `rpart` с помощью ряда параметров вспомогательной функции `rpart.control` можно изменить некоторые вышеперечисленные правила остановки:

- `cp` задает штраф за сложность, если разбиение уменьшает ошибку модели на значение, меньшее значения `cp`, оно не принимается и дерево останавливается в росте;
- `minsplit` задает минимальное количество наблюдений в родительском узле перед расщеплением, по умолчанию 20;
- `minbucket` задает минимальное количество наблюдений в терминальном узле, по умолчанию используется округленное значение `minsplit/3`;
- `maxdepth` задает максимальную глубину дерева (количество уровней дерева, лежащих ниже корневого узла), по умолчанию равна 30.

3.1.2. Неоднородность

Критерий расщепления, используемый для построения дерева CART, называется **неоднородность (impurity)**.

Применительно к дереву классификации CART под неоднородностью понимается неоднородность распределения классов зависимой переменной в узлах-потомках. Однородным узлом является тот, в котором все наблюдения относятся к одному и тому же классу зависимой переменной, в то время как узел с максимальной неоднородностью содержит равное количество наблюдений во всех классах зависимой переменной. Допустим, есть узел, и он разбит на два класса. Максимальная неоднородность в узле будет достигнута при разбиении его на два подмножества по 50 примеров, а минимальная неоднородность – при разбиении на 100 и 0 примеров.

Наиболее популярная мера неоднородности для деревьев классификации – **мера Джини**. В основе меры Джини лежат возведенные в квадрат вероятности, с которыми наблюдения будут отнесены к каждому классу зависимой переменной.

Общая формула для вычисления меры Джини выглядит так:

$$Gini(t) = 1 - \sum_{k=1}^{K} p_k^2$$

где:

K – количество классов зависимой переменной;

k – класс зависимой переменной;

p_k – вероятность k-того класса зависимой переменной в t-ом узле.

Для бинарной зависимой переменной мера Джини принимает вид:

$$Gini(t) = 1 - p_1^2 - p_0^2$$

где:

p_1 – вероятность класса 1 (положительного класса) в t-ом узле;

p_0 – вероятность класса 0 (отрицательного класса) в t-ом узле.

Когда наблюдения в узле равномерно распределены по категориям, мера Джини принимает свое максимальное значение (для бинарной зависимой переменной максимальное значение меры Джини равно 0,5). Когда все наблюдения в узле принадлежат к одной и той же категории, мера Джини равна 0.

Узел с распределением (1, 0)	Мера Джини = $1 - 1^2 - 0^2 = 0$
Узел с распределением (0,5, 0,5)	Мера Джини = $1 - 0{,}5^2 - 0{,}5^2 = 0{,}5$
Узел с распределением (0,7, 0,3)	Мера Джини = $1 - 0{,}7^2 - 0{,}3^2 = 0{,}42$

Применительно к дереву регрессии CART под неоднородностью понимается степень разброса значений количественной зависимой переменной вокруг среднего значения. Более точно, речь идет о **девиансе** – сумме квадратов остатков или разностей между фактическими значениями зависимой переменной и ее средним значением в конкретном узле.

$$Deviance(t) = \sum_i (y_{it} - \bar{y}_t)^2$$

где:

y_{it} – фактическое значение зависимой переменной для i-того наблюдения в t-ом узле;

$\bar{y}_t$ – среднее значение зависимой переменной для в t-ом узле;

n_t – количество наблюдений в t-ом узле.

При построении дерева CART расщепляет узел на дочерних узла по предиктору, который обеспечивает наибольшее уменьшение неоднородности, для этого неоднородность родительского узла сравнивается со средним взвешенным значением неоднородностей дочерних узлов:

$$\Delta i = i(par) - \left\{ \frac{n_{left}}{n_{par}} i(left) + \frac{n_{right}}{n_{par}} i(right) \right\}$$

где:

Δi – уменьшение неоднородности;

$i(par)$ – неоднородность родительского узла;

$\left\{ \frac{n_{left}}{n_{par}} i(left) + \frac{n_{right}}{n_{par}} i(right) \right\}$ – среднее взвешенное значение неоднородностей дочерних узлов:

n_{left} – количество наблюдений в левом дочернем узле;

n_{right} – количество наблюдений в правом дочернем узле;

n_{par} – количество наблюдений в родительском узле;

$i(left)$ – неоднородность левого дочернего узла;

$i(right)$ – неоднородность правого дочернего узла.

Изменение неоднородности от родительского узла к дочерним узлам называется *улучшением* и отображается на диаграмме дерева.

3.1.3. Метод отсечения ветвей на основе меры стоимости-сложности с перекрестной проверкой

При использовании критерия неоднородности для построения дерева возникает следующая проблема. Увеличивая размеры дерева, почти всегда можно уменьшить неоднородность. Любое дерево будет иметь нулевую неоднородность, если оно построено достаточно большим. В частности, если в каждом узле имеется только одно наблюдение, то неоднородность равна нулю. По мере увеличения размеров дерево становится более сложным, а ошибка модели уменьшается. Однако в процессе построения дерева возникает момент, когда при расщеплении узла добавление переменной увеличивает сложность, но не уменьшает ошибку модели. Это ведет к построению излишне сложного, детализированного дерева, происходит переобучение модели. В итоге, несмотря на высокий процент правильных прогнозов, большие деревья из-за сложности дают статистически неустойчивые результаты. Поэтому необходимо найти баланс между сложностью и оценкой риска. Для решения этих проблем разработчики CART ввели меру стоимости-сложности, которая включает штраф, возрастающий с увеличением размера дерева. Эта функция для дерева (или его ветви) обычно выражается как

$$R_\alpha(T) = R(T) + \alpha |T|,$$

где $R(T)$ – оценка риска, рассчитанная по тем же данным, по которым строилось дерево; α – коэффициент штрафа; $|T|$ – количество терминальных узлов дерева (или ветви) T.

Дерево б*о*льшего размера будет иметь б*о*льшую меру стоимости-сложности за счет слагаемого $\alpha|T|$. Для того чтобы мера стоимости-сложности улучшилась, ошибка модели должна уменьшиться в большей степени, чем штраф за сложность (в пакете R `rpart` штраф за сложность регулируется параметром `cp`).

Мера стоимости-сложности была протестирована в качестве критерия построения дерева, однако авторы (Брейман, Фридман и др.) констатировали, что построенные таким способом деревья все еще не вполне удовлетворительны – они недостаточно стабильны. Решение этой проблемы привело, в свою очередь, к методу отсечения ветвей на основе

критерия максимального уменьшения меры стоимости-сложности (cost-complexity pruning). Его суть сводится к следующему. Сначала строим максимально большое дерево (с небольшим числом наблюдений в узлах – от 1 до 5). Затем отсекаем у него ветви на основе меры стоимости-сложности. Выбираем простейшее дерево с наименьшим числом узлов, риск которого находится в пределах одной стандартной ошибки от минимальной меры риска, достигнутой на этапе построения дерева.
В дальнейшем этот метод был вновь усовершенствован авторами. В ходе экспериментов было установлено, что управление отсечением и отбором необходимо осуществлять с помощью перекрестной проверки. Для каждого шага построения дерева, то есть для каждого разбиения рассчитывается стоимость-сложность и кросс-валидационная ошибка, помогающие определить, когда достигнуто максимальное качество модели и дерево можно остановить в росте. Для дерева классификации кросс-валидационная ошибка – это ошибка классификации, усредненная по всем контрольным блокам перекрестной проверки. Для дерева регрессии кроссвалидационная ошибка – это усредненная по всем контрольным блокам перекрестной проверки сумма квадратов остатков. Кросс-валидационная ошибка оценивает способность модели выдавать правильные прогнозы на новых данных, не входивших в состав обучения. Затем строится график зависимости кросс-валидационных ошибок от числа расщеплений и сложности модели. В пакете `rpart` как раз реализовано отсечение ветвей на основе меры стоимости-сложности с перекрестной проверкой. Общая логика построения дерева в пакете rpart состоит в том, чтобы построить полное дерево с максимальным количеством расщеплений (то есть переобученное), а затем обрезать его до нужного размера, выбрав оптимальное сочетание значений кросс-валидационной ошибки и стоимости-сложности.

3.1.4. Обработка пропущенных значений

В методе CART пропущенные значения обрабатываются с использованием переменных-суррогатов. Таким образом, если наблюдение имеет пропущенное значение в переменной, по которой осуществляется разбиение узла, то для выбора дочернего узла, к которому относится данное наблюдение, используется его значение для наилучшей переменной-суррогата. Наилучшей переменной-суррогатом является альтернативная предикторная переменная, дающая наиболее близкое (с использованием меры связи) разбиение к тому, которое дает исходный предиктор.

3.1.5 Иллюстрация работы метода CART на конкретном примере

Предположим, есть данные по клиентам микрофинансовой организации и известно, выплатили они займ или нет (категориальная зависимая переменная *Просрочка*). Для удобства расчетов представим, что наш набор данных состоит всего из 5 наблюдений. В качестве потенциальных предикторов фигурируют две переменные: *Возраст* и *Пол*. Переменная *Пол* является номинальной, переменная *Возраст* является количественной. Необходимо выяснить, какие группы клиентов с большей вероятностью выйдут в просрочку, чтобы сосредоточить внимание на них. Схематично наши исходные данные представлены на рис. 3.1.

Имеется набор данных (корневой узел)

Возраст	Пол	Наличие просрочки
70	Мужской	Да
64	Мужской	Да
69	Женский	Да
68	Мужской	Нет
65	Женский	Нет

УЗЕЛ 0

Просрочка		
Нет	2	40%
Да	3	60%

Рис. 3.1 Исходные данные перед началом работы CART (корневой узел)

На первом этапе (рис. 3.2) алгоритм CART ищет наилучшую точку расщепления по количественному предиктору *Возраст* (отсортировав значения по возрастанию) и номинальному предиктору *Пол*. В каждой рассматриваемой точке расщепления родительский узел разбивается на два дочерних узла (в левый записываются наблюдения со значениями, которые меньше точки расщепления, в правый – наблюдения со значениями, которые больше или равны точке расщепления) и вычисляется среднее взвешенное значение неоднородностей дочерних узлов (по умолчанию для категориальной зависимой переменной используется мера Джини). Наилучшей точкой расщепления для каждого предиктора будет такая точка, которая дает наименьшее взвешенное среднее значение неоднородностей дочерних узлов.

CART ищет по каждому предиктору наилучшую точку расщепления, дающую наименьшее среднее взвешенное значение неоднородностей дочерних узлов

Наблюдения со значениями < точка расщепления отправляются в левый узел

1

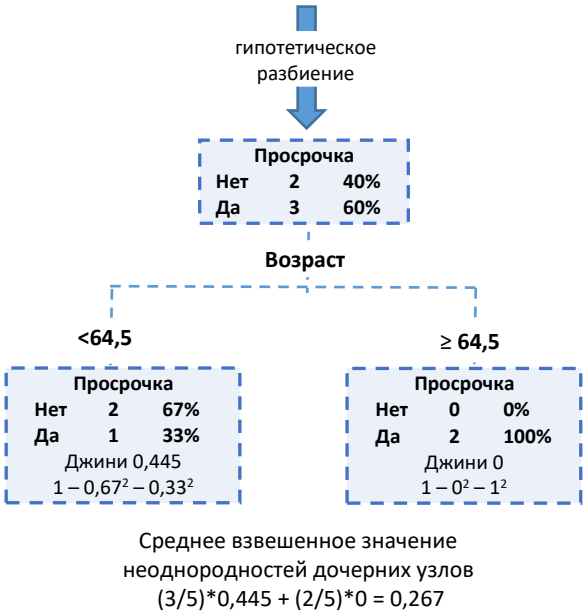
Наблюдения со значениями ≥ точка расщепления отправляются в правый узел

вычисление средних взвешенных значений неоднородностей дочерних узлов

Предиктор *Возраст*

Значения	**64**	**65**	**68**	**69**	**70**
Точки расщепления	**64,5**	**66,5**	**68,5**	**69,5**	

Точки расщепления	<64,5	≥ 64,5	<66,5	≥ 66,5	<68,5	≥ 68,5	<69,5	≥ 69,5
Категория *Нет*	0	2	1	1	2	0	2	0
Категория *Да*	1	2	1	2	1	2	2	1
Взвешенное среднее значение неоднородностей дочерних узлов	0,4		0,464		**0,267**		0,4	



Среднее взвешенное значение неоднородностей дочерних узлов
(3/5)*0,445 + (2/5)*0 = 0,267

Предиктор *Пол*

Значения	Мужской	Женский
Категория *Нет*	1	1
Категория *Да*	2	1
Взвешенное среднее значение неоднородностей дочерних узлов	**0,467**	

Рис. 3.2 Поиск наилучшего расщепления предиктора

Взвешенное среднее значение неоднородностей дочерних узлов рассчитывается следующим образом. Сначала вычисляются оценки неоднородности дочерних узлов. Затем оценка неоднородности левого дочернего узла умножается на вес левого дочернего узла (долю наблюдений, попавших в дочерний левый узел, взятую от общего количества наблюдений в родительском узле). Потом оценка неоднородности правого дочернего узла умножается на вес правого дочернего узла (долю наблюдений, попавших в дочерний правый узел, взятую от общего количества наблюдений в родительском узле). Наконец, произведения суммируются и получается взвешенное среднее неоднородностей дочерних узлов. Все вышесказанное можно проиллюстрировать формулой:

$$\Delta i = i(par) - \left\{ \frac{n_{left}}{n_{par}} i(left) + \frac{n_{right}}{n_{par}} i(right) \right\}$$

где:

Δi – уменьшение неоднородности;
$i(par)$ – неоднородность родительского узла;
$\left\{ \frac{n_{left}}{n_{par}} i(left) + \frac{n_{right}}{n_{par}} i(right) \right\}$ – среднее взвешенное значение неоднородностей дочерних узлов:
n_{left} – количество наблюдений в левом дочернем узле;
n_{right} – количество наблюдений в правом дочернем узле;
n_{par} – количество наблюдений в родительском узле;
$i(left)$ – неоднородность левого дочернего узла;
$i(right)$ – неоднородность правого дочернего узла.

Для предиктора *Возраст* наилучшей точкой расщепления становится точка 68,5, которая дает наименьшее среднее взвешенное значене неоднородностей дочерних узлов, равное 0,267. Для предиктора Пол такой наилучшей точкой расщепления автоматически становится разбиение на мужчин и женщин.
На втором этапе (рис. 3.3) алгоритм CART выбирает наилучшую точку расщепления из набора наилучших точек расщепления, вычисленных по каждому предиктору на первом этапе, и разбивает по ней узел. Наилучшей точкой расщепления считается точка, минимизирующая среднее взвешенное значение неоднородностей дочерних узлов. В данном

случае такой точкой будет точка 68,5, полученная для предиктора *Возраст*. Алгоритм вычисляет улучшение. Улучшение рассчитывается как разность между неоднородностью родительского узла и минимальным взвешенным средним неоднородностей дочерних узлов.

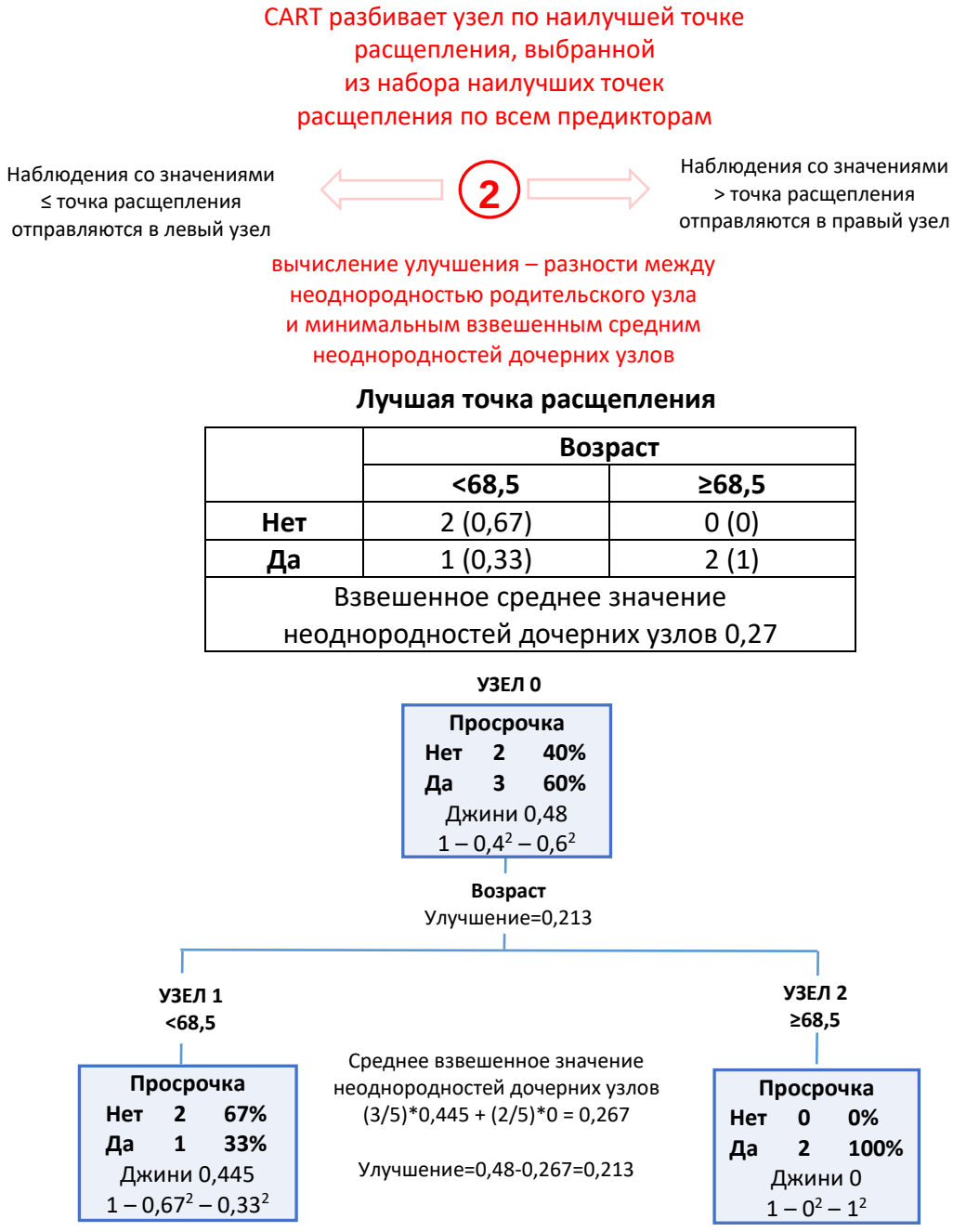


Рис. 3.3 Выбор наилучшего расщепления родительского узла

Алгоритм спускается на уровень ниже и для полученных узлов 1 и 2 повторяет вышеописанные шаги. Сначала он рассматривает узел 1 (рис. 3.4).

Имеется поднабор данных (узел 1)

Возраст	Пол	Наличие просрочки
64	Мужской	Да
68	Мужской	Нет
65	Женский	Нет

УЗЕЛ 1

Просрочка		
Нет	2	67%
Да	1	33%

Рис. 3.4 Поднабор данных (узел 1)

Снова для каждой точки расщепления по каждому предиктору вычисляется среднее взвешенное значение неоднородностей дочерних узлов. Для каждого предиктора выбираем точку расщепления, которое дает наименьшее взвешенное среднее значение неоднородностей дочерних узлов.

Предиктор *Возраст*

Значения	**64**	**65**	**68**
Точки расщепления	**64,5**	**66,5**	

Точки расщепления	<64,5	≥ 64,5	<66,5	≥ 66,5
Категория *Нет*	0	2	1	1
Категория *Да*	1	0	1	0
Взвешенное среднее значение неоднородностей дочерних узлов	**0**		0,33	

Предиктор *Пол*

Значения	Мужской	Женский
Нет	1	1
Да	1	0
Взвешенное среднее значение неоднородностей дочерних узлов	**0,33**	

Рис. 3.5 Поиск наилучшего расщепления предиктора

Снова выбираем наилучшую точку расщепления из набора наилучших точек расщепления, вычисленных по всем предикторам, и разбиваем по ней узел. Алгоритм вновь вычисляет улучшение – изменение неоднородности от родительского узла к дочерним узлам.

Лучшая точка расщепления

	Возраст	
	<64,5	**≥64,5**
Нет	0 (0)	2 (1)
Да	1 (1)	0 (0)
Взвешенное среднее значение неоднородностей дочерних узлов 0		

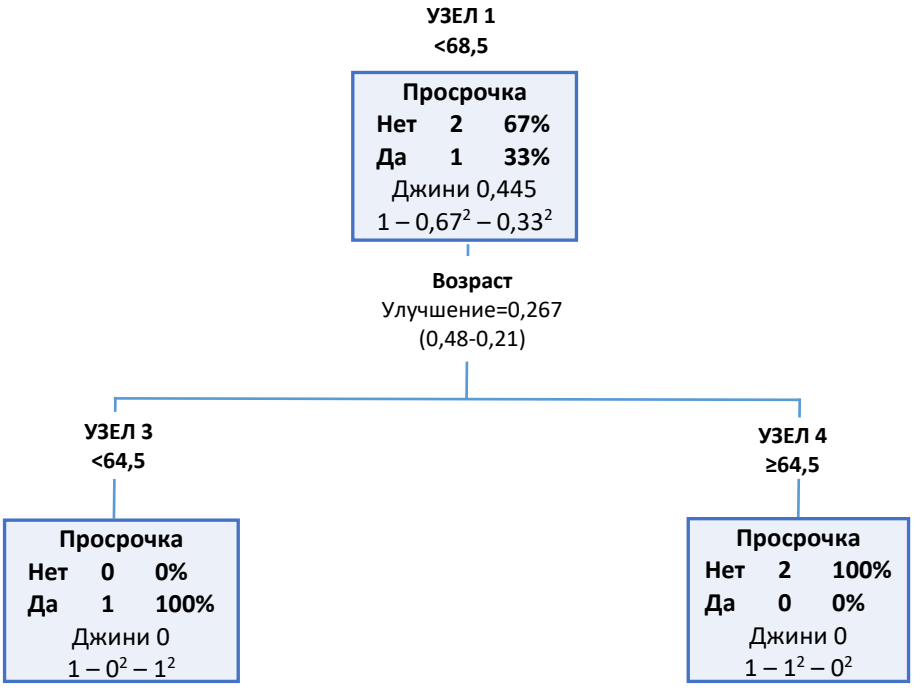


Рис. 3.6 Выбор наилучшего расщепления узла 1

Затем алгоритм рассматривает узел 2. Поскольку он является однородным, он не разбивается (срабатывает правило остановки) и становится терминальным узлом.

Алгоритм спускается на уровень ниже и рассматривает узлы 3 и 4. Поскольку узлы 3 и 4 тоже являются однородными, они не разбиваются (вновь срабатывает правило остановки) и становятся терминальными узлами.

Итоговое дерево CART показано на рис. 3.7.

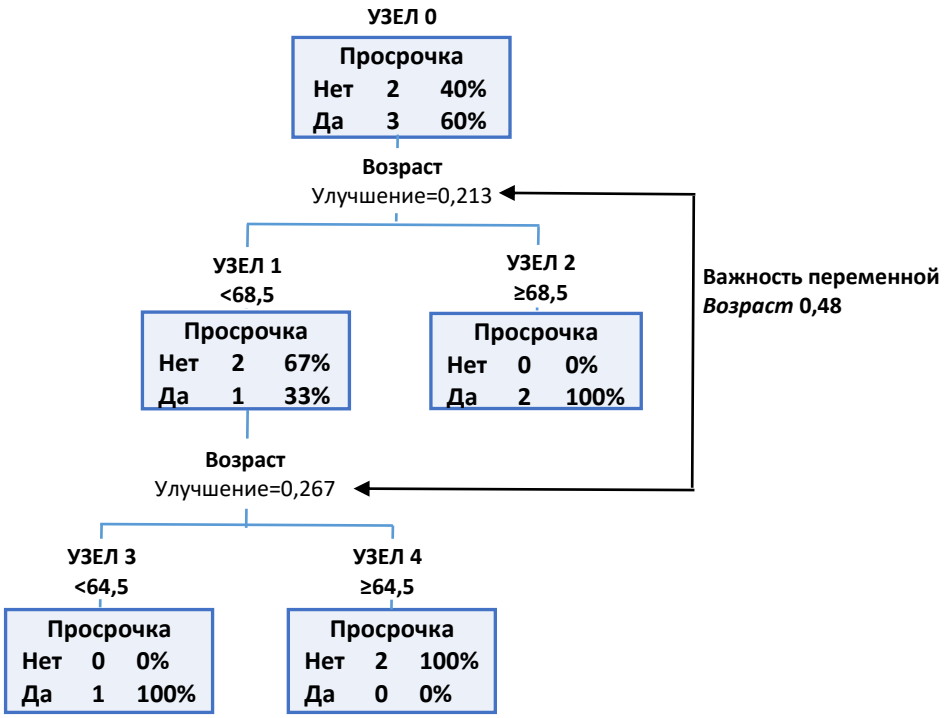


Рис. 3.7 Итоговое дерево CART

Обратите внимание, метод CART позволяет вычислить важности предикторов. Данная информация может быть полезна, когда у вас имеется большое количество переменных и необходимо выбрать наиболее важные для включения в прогнозную модель (например, в модель логистической регрессии). Важность предиктора – это сумма улучшений, вызванных применением данного предиктора в качестве основной расщепляющей переменной или переменной-суррогата. Каждый из предикторов при каждом расщеплении берется с весом. Вес зависит от того, применялся ли предиктор в качестве основной расщепляющей переменной (предиктор, по которому был расщеплен родительский узел, имеет вес 1) или в качестве суррогата (вес зависит от ранга суррогата). Если допускается использовать 5 суррогатов, то первый суррогат на расщепление будет иметь наибольший вес среди прочих, а пятый – самый низкий.
Кроме того, часто вычисляют нормализованную важность для переменной, которая определяется формулой:

Нормализованная важность = 100 x (Важность/Максимальная важность)

Поэтому наиболее важный предиктор имеет нормализованное значение важности, равное 100.

Давайте вычислим важность предиктора *Возраст* в рамках нашего игрушечного примера. Она будет равна сумме улучшений – уменьшений неоднородности, когда в качестве предиктора использовалась переменная *Возраст*. Складываем наши улучшения 0,213 и 0,267, получаем 0,48. В данном случае важность предиктора *Возраст* равна неоднородности корневого узла. Это обозначает, что неоднородность корневого узла удалось полностью снизить за счет использования одной переменной *Возраст*.

Нетрудно увидеть недостаток важности. По сути важность складывается из частоты использования переменной в качестве предиктора разбиения, то есть чаще наиболее важными будут переменные, по которым можем быть рассмотрено больше вариантов разбиения и у них больше шансов стать предиктором разбиения. Поэтому наиболее важными переменными чаще будут переменные с большим количеством уникальных значений, традиционно это количественные переменные.

3.1.6 Особенности реализации метода CART в пакете R rpart

В пакете `rpart` на первом этапе для каждой рассматриваемой точки расщепления вместо взвешенного среднего значения неоднородностей дочерних узлов алгоритм вычисляет взвешенное уменьшение неоднородности при разбиении родительского узла на дочерние узлы. Сначала вычисляем разницу между неоднородностью родительского узла и неоднородностью левого дочернего узла, затем разницу между неоднородностью родительского узла и неоднородностью правого дочернего узла. Затем эти разницы умножаем на размеры узлов и полученные результаты складываем.

$$\Delta i = n_{left} \times \left[i(par) - i(left) \right] + n_{right} \times \left[i(par) - i(right) \right]$$

В итоге в качестве наилучшей рассматривается точка расщепления, которая дает наибольшее взвешенное уменьшение неоднородности.

CART ищет по каждому предиктору наилучшую точку расщепления, дающую наибольшее взвешенное уменьшение неоднородности

Наблюдения со значениями < точка расщепления отправляются в левый узел

1

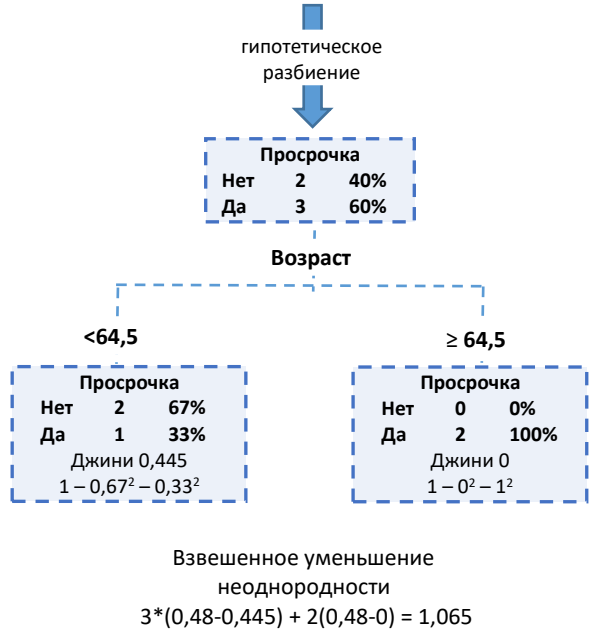
Наблюдения со значениями ≥ точка расщепления отправляются в правый узел

вычисление взвешенных уменьшений неоднородности

Предиктор *Возраст*

Значения	**64**	**65**	**68**	**69**	**70**
Точки расщепления	**64,5**	**66,5**	**68,5**	**69,5**	

Точки расщепления	<64,5	≥ 64,5	<66,5	≥ 66,5	<68,5	≥ 68,5	<69,5	≥ 69,5
Категория *Нет*	0	2	1	1	2	0	2	0
Категория *Да*	1	2	1	2	1	2	2	1
Взвешенное уменьшение неоднородности	0,4		0,065		**1,065**		0,4	



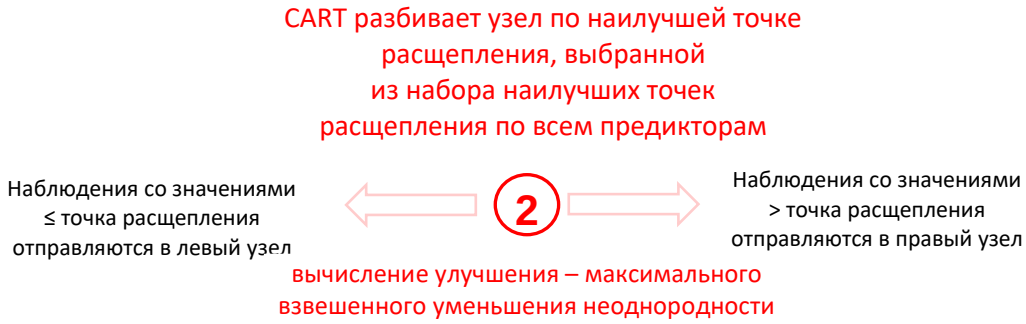
Взвешенное уменьшение неоднородности
3*(0,48-0,445) + 2(0,48-0) = 1,065

Предиктор *Пол*

Значения	Мужской	Женский
Категория *Нет*	1	1
Категория *Да*	2	1
Взвешенное уменьшение неоднородности	**0,065**	

Рис. 3.8 Поиск наилучшего расщепления предиктора в пакете R rpart

На втором этапе алгоритм CART выбирает наилучшую точку расщепления из набора наилучших точек расщепления, вычисленных по всем предикторам на первом этапе, и разбивает по ней узел. Наилучшей точкой расщепления считается точка, максимизирующая взвешенное уменьшение неоднородности. Максимальное взвешенное уменьшение неоднородности будет считаться улучшением. Кроме того, вычисляются т.н. альтернативные улучшения – взвешенные уменьшения неоднородности, которые можно было получить при использовании других переменных в качестве предиктора расщепления.



Лучшая точка расщепления

	Возраст	
	<68,5	**≥68,5**
Нет	2 (0,67)	0 (0)
Да	1 (0,33)	2 (1)
Взвешенное уменьшение неоднородности 1,065		

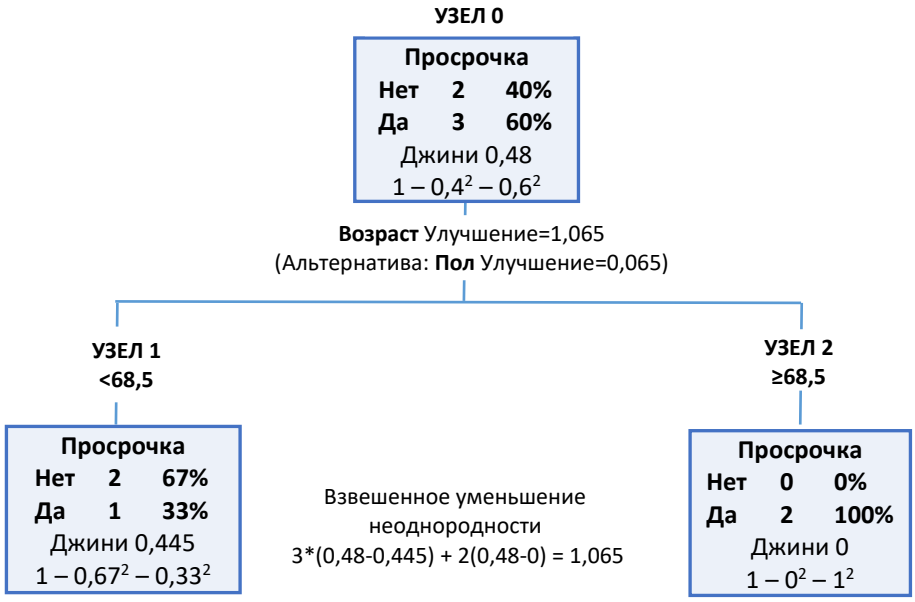


Рис. 3.9 Выбор наилучшего расщепления родительского узла в пакете R rpart

Для полученных узлов алгоритм повторяет вышеописанные шаги, и так до тех пор, пока не сработают правила остановки. Итоговое дерево CART показано в адаптированном виде на рис. 3.10.

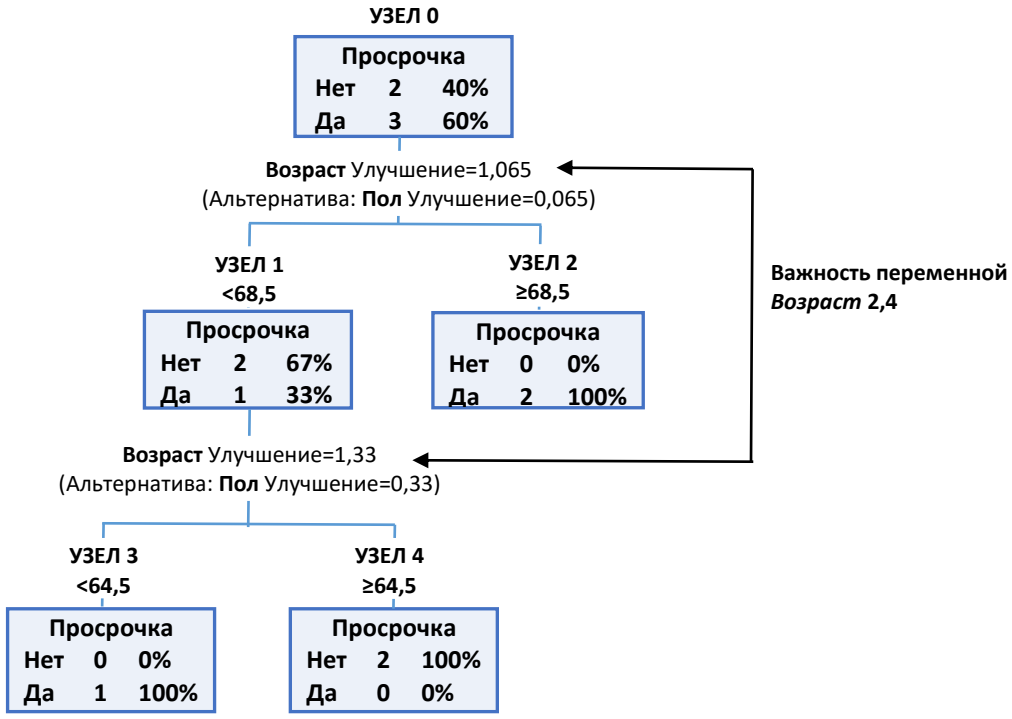


Рис. 3.10 Итоговое дерево CART в пакете rpart

Как видно, построенное дерево идентично дереву, которое строится с помощью классического алгоритма CART, за исключением лишь того, что теперь улучшение является максимальной суммой взвешенных уменьшений неоднородности, а не разностью между неоднородностью родительского узла и минимальным взвешенным средним неоднородностей дочерних узлов. Кроме того, выводится информация об альтернативных улучшениях.