Алгоритм CART [M.154]

CART (Classification and Regression Tree) — популярный алгоритм построения деревьев решений, предложенный в 1984 г. (L. Breiman, J. Friedman, R. Olshen и Ch. Stone).

Деревья решений, построенные с помощью CART, являются бинарными, то есть содержат только два потомка в каждом узле.

Пусть задано обучающее множество, содержащее K примеров и N классов. Введем в рассмотрение показатель, который позволит оценить эффективность разбиения, полученного на основе конкретного атрибута. Обозначим его $Q(s \mid t)$, где s — идентификатор разбиения, t — идентификатор узла. Тогда можно записать:

$$Q(s \mid t) = 2 \cdot P_L \cdot P_R \sum_{j=1}^{N} \left(P(j \mid t_L) - P(j \mid t_R) \right) \quad (1)$$

где t_L и t_R — левый и правый потомки узла t соответственно;

$P_L = K_L / K$ — отношение числа примеров в левом потомке узла t к общему числу примеров;

$P_R = K_R / K$ — отношение числа примеров в правом потомке узла t к общему числу примеров;

$P(j \mid t_L) = K_j^L / K_t$ — отношение числа примеров j-го класса в t_L к общему числу примеров в t_L;

$P(j \mid t_R) = K_j^R / K_t$ — отношение числа примеров j-го класса в t_R к общему числу примеров в t_R.

Тогда наилучшим разбиением в узле t будет то, которое максимизирует показатель .

В качестве примера рассмотрим задачу предсказания кредитного риска потенциального клиента банка (таблица 1).

Таблица 1– Набор данных

№ клиента	Сбережения	Другие активы (недвижимость, автомобиль и т. д.)	Годовой доход (тыс. у. е.)	Кредитный риск
1	Средние	Высокие	75	Низкий
2	Низкие	Низкие	50	Высокий
3	Высокие	Средние	25	Высокий
4	Средние	Средние	50	Низкий
5	Низкие	Средние	100	Низкий
6	Высокие	Высокие	25	Низкий
7	Низкие	Низкие	25	Высокий
8	Средние	Средние	75	Низкий

Все восемь примеров обучающего множества поступают в корневой узел дерева. Поскольку алгоритм CART создает только бинарные разбиения, возможные кандидаты, которые будут оцениваться на начальном этапе, представлены в таблице 2. Непрерывный атрибут **Доход** был предварительно квантован с помощью порогов 25, 50 и 75 тыс.

Таблица 2 – Потенциальные разбиения

Разбиение	Левый потомок, t_L	Правый потомок, t_R
1	Сбережения = *Низкие*	Сбережения ∈ {*Высокие*, *Средние*}
2	Сбережения = *Средние*	Сбережения ∈ {*Высокие*, *Низкие*}
3	Сбережения = *Высокие*	Сбережения ∈ {*Средние*, *Низкие*}
4	Активы = *Низкие*	Активы ∈ {*Высокие*, *Средние*}
5	Активы = *Средние*	Активы ∈ {*Высокие*, *Низкие*}
6	Активы = *Высокие*	Активы ∈ {*Средние*, *Низкие*}
7	Доход ≤ 25	Доход > 25
8	Доход ≤ 50	Доход > 50
9	Доход ≤ 75	Доход > 75

Для каждого потенциального разбиения вычислим значения составляющих выражения (1) и исследуем их влияние на значение всего показателя Q.

Можно увидеть, что $Q(s \mid t)$ увеличивается, когда оба сомножителя в выражении (1) $2 \cdot P_L \cdot P_R$ и $\sum_{j=1}^{N} \left(P(j \mid t_L) - P(j \mid t_R) \right)$ также увеличиваются.

Обозначим сумму в выражении (1) через $W(s \mid t)$, то есть

$$W(s \mid t) = \sum_{j=1}^{N} \left(P(j \mid t_L) - P(j \mid t_R) \right)$$

Компонент $W(s \mid t)$ будет расти при увеличении разности в скобках. Данная сумма максимальна, когда количества примеров, относящихся к одному классу, в обоих потомках максимально различается. Следовательно, значение окажется наибольшим тогда, когда оба потомка вообще не будут содержать примеров одинаковых классов. В частности, если классов только два, то оба потомка будут чистыми. Теоретически максимальное значение $W(s \mid t)$ равно числу классов в обучающем множестве. Поскольку в нашем примере только два класса — Высокий и Низкий, максимальное значение $W(s \mid t)$ равно 2. В таблице 3 приведены результаты расчета компонентов выражения (1).

Таблица 3 – Расчет значений меры Q для потенциальных разбиений

№	P_L	P_R	$P(j \mid t_L)$		$P(j \mid t_R)$		$2 \cdot P_L \cdot P_R$	$W(s\|t)$	$Q(s\|t)$
			Низкий	Высокий	Низкий	Высокий			
1	0,375	0,625	0,333	0,667	0,8	0,2	0,46875	0,934	0,4378
2	0,375	0,625	1	0	0,4	0,6	0,46875	1,2	0,5625
3	0,25	0,75	0,5	0,5	0,667	0,333	0,375	0,334	0,1253
4	0,25	0,75	0	1	0,833	0,167	0,375	1,667	0,6248
5	0,5	0,5	0,75	0,25	0,5	0,5	0,5	0,5	0,25
6	0,25	0,75	1	0	0,5	0,5	0,375	1	0,375
7	0,375	0,625	0,333	0,667	0,8	0,2	0,46875	0,934	0,4378
8	0,625	0,375	0,4	0,6	1	0	0,46875	1,2	0,5625
9	0,875	0,125	0,571	0,429	1	0	0,21875	0,858	0,1877

Произведение $P_L \cdot P_R$ возрастает с увеличением значений сомножителей. Это происходит, когда доли записей одного класса в левом и правом потомках оказываются равны, Следовательно, мера $Q(s \mid t)$ имеет тенденцию давать сбалансированные разбиения, которые будут делить исходное множество на подмножества, содержащие примерно одинаковое количество записей. Теоретически максимальное значение $2 \cdot P_L \cdot P_R = 2 \cdot 0{,}5 \cdot 0{,}5 = 0{,}5$.

В нашем примере только потенциальное разбиение № 5 (см. таблицу 2) дает произведение $P_L \cdot P_R$, достигающее теоретического максимума 0,5, поскольку в результате записи были разделены на две равные группы по четыре в каждой.

Наибольшее из наблюдаемых значений $Q(s \mid t)$ было получено для разбиения № 4 (см. таблицу 2), где $Q(s \mid t) = 0{,}6248$. Значит, для начального разбиения CART выберет условие, являются ли активы клиента низкими. Тогда в результате разбиения будут созданы два потомка: в одном окажутся записи, в которых атрибут **Активы** принимает значение *Низкие*, во втором — записи, в которых этот же атрибут принимает значения *Высокие* и *Средние*. Полученное в результате данного разбиения дерево представлено на рисунке 1.

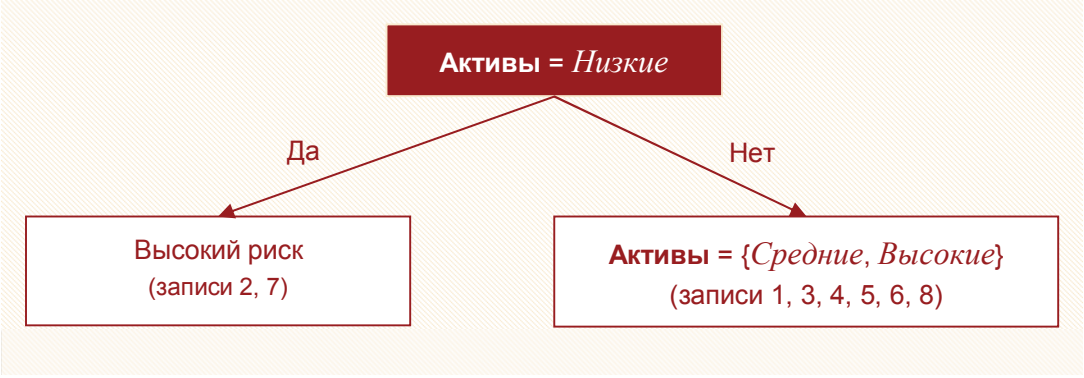

Рисунок 1 – Дерево после первого разбиения

Обе записи, в которых активы клиента являются низкими и по этой причине оказавшиеся в левом узле, содержат одну и ту же целевую переменную, указывающую на высокий кредитный риск. Таким образом, узел является чистым. Узел будет объявлен листом, и дальнейшее разбиение по данной ветви производиться не будет. Записи в правом узле относятся к различным классам. Потребуется дальнейшее разбиение, поэтому мы снова рассчитаем

$Q(s \mid t)$ и представим полученные результаты в таблице 4. Обратим внимание на то, что ранее использованное разбиение № 4 больше не рассматривается.

Таблица 4– Второй этап расчета значений меры Q

№	P_L	P_R	$P(j \mid t_L)$		$P(j \mid t_R)$		$2 \cdot P_L \cdot P_R$	$W(s\|t)$	$Q(s\|t)$
			Низкий	Высокий	Низкий	Высокий			
1	0,167	0,833	1	0	0,8	0,2	0,2782	0, 4	0,1112
2	0,5	0, 5	1	0	0,667	0,333	0,5	0,6666	0,3333
3	0,333	0,667	0,5	0,5	1	0	0,4444	1	0,4444
4	0,667	0,333	0,75	0,25	1	0	0,4444	0,5	0,2222
5	0,333	0,667	1	0	0,75	0,25	0,4444	0,5	0,2222
6	0,333	0,667	0,5	0,5	1	0	0,4444	1	0,4444
7	0,5	0, 5	0,667	0,333	1	0	0,5	0,6666	0,3333
8	0,167	0,833	0,8	0,2	1	0	0,2782	0,4	0,1112
9	0,167	0,833	1	0	0,8	0,2	0,2782	0, 4	0,1112

Наибольшее значение меры $Q(s \mid t) = 0{,}4444$ было получено для разбиений № 3 и 7.

Произвольным образом выберем разбиение № 3 **Сбережения** = *Высокие*. В результате дерево будет дополнено двумя новыми узлами. В левом потомке окажутся записи, в которых атрибут **Сбережения** принимает значение *Высокие*. Таких записей в исходном множестве всего две (см. таблицу 1, записи № 3 и 6). Однако они имеют разное значение целевой переменной: для записи № 2 кредитный риск высокий, а для записи № 3 — низкий. Следовательно, данный узел содержит два класса, и в нем возможно дальнейшее ветвление. Во второй узел этого разбиения будут отобраны оставшиеся записи с номерами 1, 4, 5 и 8. Как можно увидеть из таблицы 1, все они имеют одну и ту же метку класса, указывающую на низкий риск кредитования заемщиков. Поскольку записи, попавшие в данный узел, относятся к одному классу, узел объявляется листом. Результирующее дерево представлено на рисунке 2.

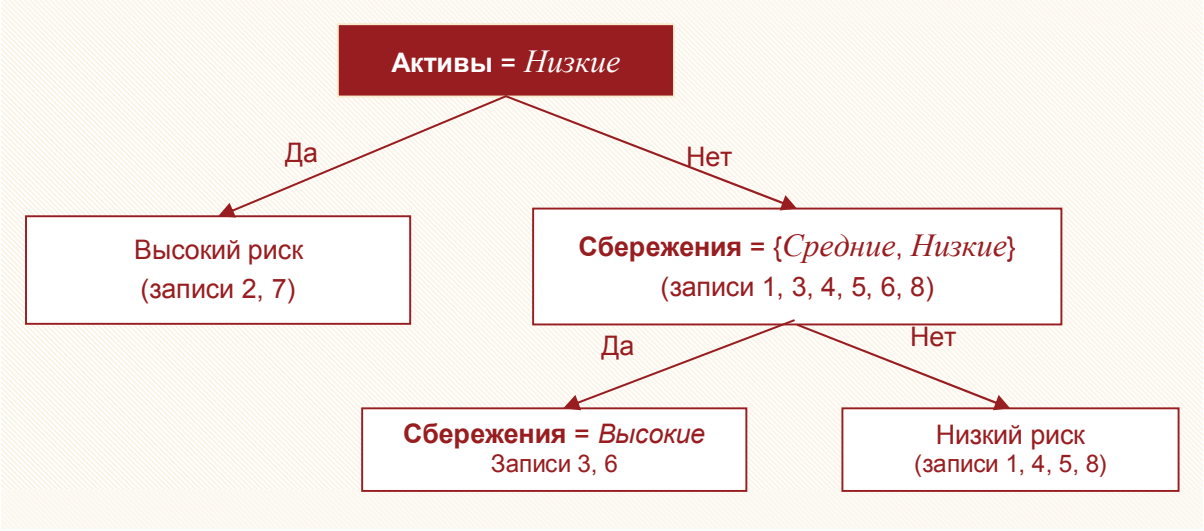

Рисунок 2 – Дерево после второго разбиения

Новое разбиение возможно только для узла, содержащего записи № 3 и 6, относящиеся к различным классам. Для разбиения можно использовать два ранее не применявшихся атрибута — **Доход** и **Другие активы**. Однако, поскольку в обеих записях сумма дохода одна и та же (25 000 у. е.), это ничего не даст. В то же время значения атрибута **Другие активы**

различаются: для записи № 3 — *Средние*, а для записи № 6 — *Высокие*, поэтому его можно использовать для разбиения. Результирующее дерево представлено на рисунке 3.

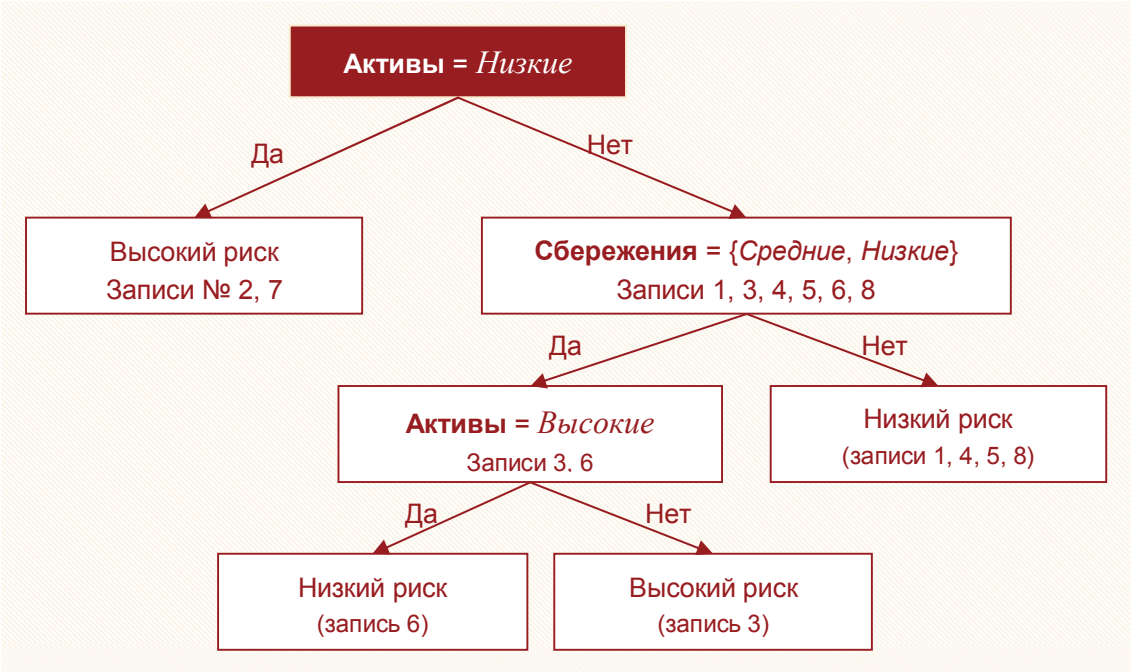

Рисунок 3 – Дерево после третьего разбиения

В общем случае алгоритм будет рекурсивно продолжаться, пока в дереве остаются узлы, содержащие примеры, которые относятся к различным классам. Когда узлов, для которых можно выполнить разбиение, не останется (то есть все подветви будут заканчиваться листьями), будет построено полное дерево. Но при работе с реальными наборами данных часто возникает ситуация, когда получить абсолютно чистые узлы не удается, что ведет к возникновению ошибки классификации. Рассмотрим набор данных, представленный в таблице 5.

Таблица 5 – Набор данных

№ клиента	Сбережения	Другие активы (недвижимость, автомобиль и т. д.)	Годовой доход (тыс. у. е.)	Кредитный риск
1	Высокие	Низкие	< 30	Низкий
2	Высокие	Низкие	< 30	Низкий
3	Высокие	Низкие	< 30	Высокий
4	Высокие	Низкие	< 30	Высокий
5	Высокие	Низкие	< 30	Высокий

Визуальный анализ таблицы 5 показывает, что все представленные в ней потенциальные заемщики характеризуются высокими сбережениями, но низкими (менее 30 тыс. у. е.) доходами. В то же время несмотря на одинаковые значения атрибутов целевые переменные для идентичных записей окажутся разными. Как известно, записи, где одному и тому же набору значений входных переменных соответствуют разные значения выходных, называются *противоречивыми*, и от них стремятся избавиться путем очистки данных. На практике подобная ситуация может сложиться, если атрибуты **Сбережения** и **Другие активы** получены путем квантования соответствующих непрерывных атрибутов.

Очевидно, что в данном случае невозможно получить чистые узлы, так как одинаковые значения независимых атрибутов не позволят разделить объекты по классам. В результате в листьях окажутся «смеси» нескольких классов. В такой ситуации можно отнести всех клиентов к категории высокого кредитного риска. Тогда вероятность того, что случайно выбранная из данного набора запись будет отнесена к соответствующему классу, составит 3/5 = 0,6, или 60 %. В то же время вероятность неправильной классификации составит 0,4, или 40 %, Данное значение называется *ошибкой классификации* (classification error rate). Если узлов, в которых допущена ошибка классификации, несколько, то полная ошибка дерева есть средневзвешенная ошибка по всем узлам. При этом в качестве весов используются доли записей в каждом листе относительно общего количества записей в обучающем множестве.

Регрессионное дерево решений

Как следует из названия алгоритма CART — классификационные и регрессионные деревья решений — он позволяет строить не только классификационные, но и регрессионные модели. В основном процесс построения регрессионного дерева аналогичен процессу построения классификационного, но вместо меток классов в листьях будут расположены числовые значения. Фактически регрессионные деревья реализуют кусочно-постоянную функцию входных переменных (рисунок 4).

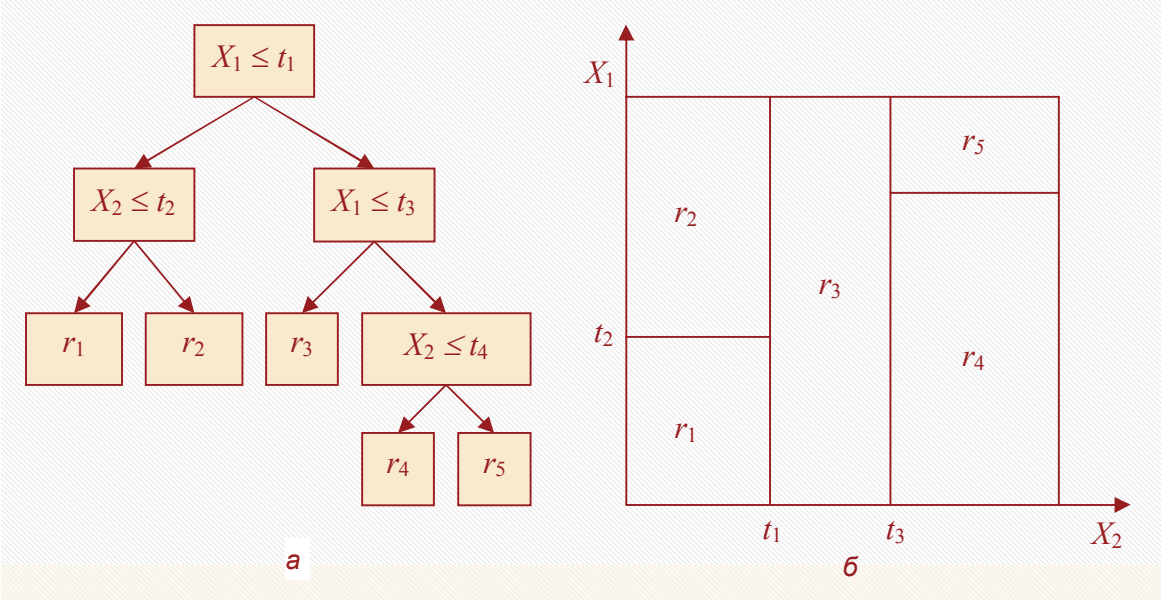

Рисунок 4 – Регрессионное дерево решений

На рисунке 4а представлено дерево решений, построенное для двух входных переменных X_1 и X_2 и содержащее 5 листьев. На рисунке 4б показано разбиение (кусочно-постоянная функция двух переменных), которое реализует данное дерево.

В результате построения регрессионного дерева в каждом листе должны оказаться примеры с близкими значениями выходной переменной. Чем ближе будут эти значения, тем меньше будет их дисперсия. Поэтому дисперсия является хорошей мерой чистоты узла.

Для минимизации квадратичной ошибки на обучающем множестве результат в листе определяется как среднее значений выходных переменных обучающих примеров, распределенных в данный лист. При этом мера «загрязненности» листа I пропорциональна дисперсии D_y значений выходной переменной примеров в узле:

$$I = D_y = E\left\{\left(y_n - E_y\right)^2\right\}.$$

Тогда наилучшим разбиением в узле будет то, которое обеспечит максимальное уменьшение дисперсии выходной переменной.

Следует отметить, что для регрессионных деревьев решений упрощение важнее, чем для классификационных. Это связано с тем, что регрессионные деревья, как правило, получаются более сложными, чем классификационные, поскольку количество значений непрерывной целевой переменной намного разнообразнее, чем категориальной. Например, если значения непрерывной целевой переменной уникальны для каждого примера обучающего множества, то полное дерево будет содержать число листьев, равное числу примеров. Процедура упрощения дерева путем отсечения ветвей основана на анализе квадратичной ошибки на тестовом множестве: отсекаются все узлы, удаление которых не приводит к росту ошибки,