Оглавление

5 Разделяй и властвуй: классификация с использованием деревьев решений и правил

Выбирая между предложениями о работе с разными уровнями оплаты и бонусами, многие взвешивают все за и против, руководствуясь простыми правилами, например: «Если придется добираться на работу больше часа, мне это не понравится» или «Если я заработаю менее 50 тысяч долларов, то не смогу содержать семью». Как видите, сложно сделать правильный выбор, но в этом может помочь несколько простых решений.

В этой главе будут рассмотрены деревья решений и алгоритмы обучения на основе правил — два метода машинного обучения, которые позволяют принимать сложные решения, выбирая из множества простых вариантов. Эти методы представляют решение в виде логических схем, которые можно понять без специальных знаний. Благодаря этому аспекту такие модели особенно полезны для построения бизнес-стратегий и улучшения процессов.

В этой главе вы:

- узнаете, как деревья и правила «жадно» делят данные на интересные сегменты;
- познакомитесь с самыми распространенными обучающими алгоритмами на основе дерева решений и правил классификации, включая алгоритмы C5.0, 1R и RIPPER;
- научитесь использовать эти алгоритмы для выполнения реальных задач классификации, таких как выявление рискованных банковских кредитов и распознавание ядовитых грибов.

Начнем с изучения деревьев решений, а затем перейдем к правилам классификации. После этого подведем итоги и рассмотрим краткий обзор последующих глав, в которых приведены технологии, использующие деревья и правила в качестве основы для более расширенных методов машинного обучения.

Деревья решений

Обучающие алгоритмы на базе дерева решений — это мощные классификаторы, в которых используется *древовидная структура* для моделирования отношений между признаками и возможными результатами. Как показано на рис. 5.1, эта структура получила такое название благодаря своему сходству с реальным деревом: начинается с широкого ствола, который разделяется на ветви — чем выше, тем тоньше. Примерно таким же образом в классификаторе, построенном на основе дерева решений, используется структура ветвящихся решений, по которой примеры перенаправляются к окончательному спрогнозированному значению класса.

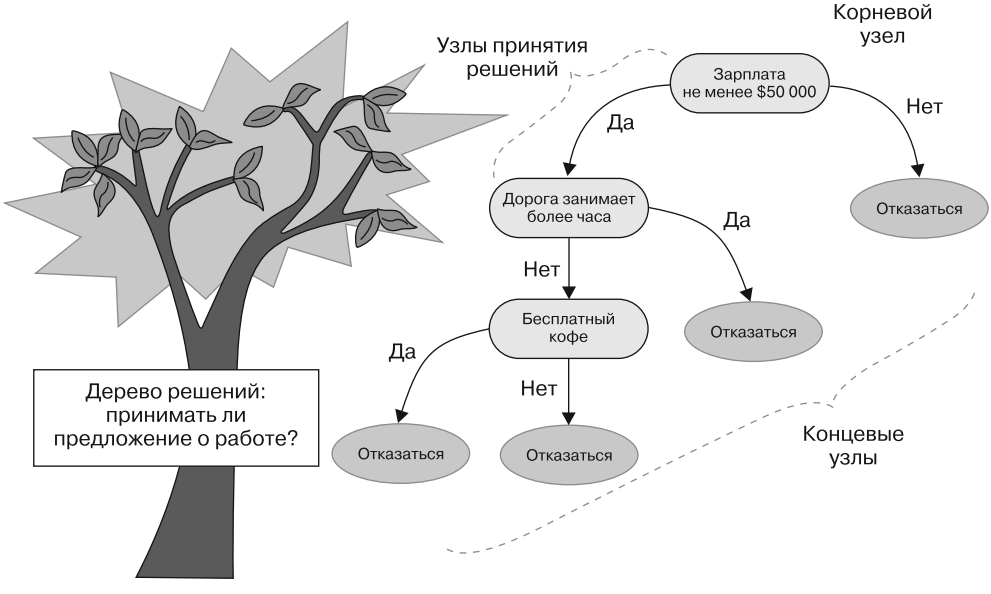

Рис. 5.1. Дерево решений, изображающее процесс принятия предложения о работе

Чтобы лучше понять, как это работает, рассмотрим дерево, которое позволяет решить, следует ли принимать предложение о работе. Рассмотрение предложения начинается с *корневого узла*; затем оно проходит через *узлы принятия решений*, в каждом из которых делается выбор на основе условий задачи. Эти точки выбора делят данные на *ветви*, которые указывают на возможные окончательные решения. В данном случае эти решения изображены как ответы «да» и «нет», но в других случаях возможно более двух вариантов.

В точке, где может быть принято окончательное решение, дерево заканчивается *концевыми узлами* (также известны как *терминальные узлы*). Эти узлы обозначают действие, которое должно быть выполнено в результате последовательности решений. В случае модели прогнозирования концевые узлы предоставляют ожидаемый результат для данной последовательности событий в дереве.

Огромное преимущество алгоритмов, основанных на дереве принятия решений, состоит в том, что древовидная структура, подобная блок-схеме, предназначена не только для внутреннего использования машиной. После создания модели многие алгоритмы, основанные на дереве принятия решений, выводят полученную структуру в формате, удобном для восприятия человеком. Это дает представление о том, как и почему работает или не работает данная модель для конкретной задачи. Благодаря этому деревья решений особенно хорошо подходят для тех случаев, когда механизм классификации должен быть прозрачным по юридическим причинам или если результаты должны быть переданы в другие инстанции для использования в будущем. С учетом этого данные алгоритмы применяются:

- для составления рейтинга кредитоспособности, в котором причины отказа заявителю в кредите должны быть четко документированы и свободны от предвзятости;
- в маркетинговых исследованиях поведения клиентов, такого как удовлетворенность или отток, которые будут переданы руководству или рекламным агентствам;
- для диагностики заболеваний на основе лабораторных исследований, симптомов или скорости прогрессирования заболевания.

Хотя указанные области применения достаточно хорошо демонстрируют ценность деревьев для процессов обоснованного принятия решений, это не означает, что их полезность этим ограничивается. На практике деревья решений являются, пожалуй, наиболее широко используемой технологией машинного обучения и могут применяться для моделирования практически любого типа данных — часто с превосходной эффективностью без дополнительной настройки.

Тем не менее, невзирая на их широкое применение, стоит отметить, что существуют случаи, когда деревья решений далеко не идеальный вариант. Сюда входят задачи, в которых данные имеют много именованных признаков с большим количеством уровней или числовых признаков. Эти случаи могут приводить к огромному количеству решений и чрезмерно сложным деревьям. Они также иногда способствуют склонности деревьев решений к сверхчувствительности данных, хотя, как мы скоро увидим, даже этот недостаток можно преодолеть, введя некоторые простые параметры.

Разделяй и властвуй

Деревья решений строятся с использованием эвристики, называемой *рекурсивным сегментированием*. Этот подход также широко известен как метод *«разделяй и властвуй»* (divide and conquer), так как он разбивает данные на подмножества, которые затем снова разделяются на еще меньшие подмножества и так далее, до тех пор, пока процесс не остановится. Это произойдет, когда алгоритм определит, что данные в подмножествах являются достаточно однородными, или выполнится другое условие остановки.

Чтобы проследить, как при разделении набора данных строится дерево решений, представьте один только корневой узел, который затем вырастает в целое дерево. Сначала корневой узел представляет собой весь набор данных, поскольку расщепления не произошло. В этот момент алгоритм построения дерева должен выбрать признак разделения; в идеале он выбирает наиболее легко предсказуемый признак целевого класса. Затем примеры (объекты) разбиваются на группы в соответствии с различными значениями этого признака и формируется первое множество ветвей дерева.

Продвигаясь по каждой ветви, алгоритм продолжает разделять и властвовать над данными, всякий раз выбирая наилучший признак-кандидат для создания очередного узла принятия решения, пока не будет достигнут критерий остановки. Метод «разделяй и властвуй» может остановиться на узле, если:

- все (или почти все) примеры в данном узле относятся к одному классу;
- не осталось признаков, по которым можно было бы разделить оставшиеся примеры;
- размер дерева достиг заданной предельной величины.

Для того чтобы лучше разобраться в процессе построения дерева, рассмотрим простой пример. Представьте, что вы работаете на голливудской киностудии. Ваша задача — решить, должна ли студия продолжать снимать фильмы по сценариям, предлагаемым молодыми многообещающими авторами. Вы вернулись из отпуска, и ваш стол завален предложениями. Чтобы не тратить время на чтение каждого предложения от начала до конца, вы решаете создать алгоритм построения дерева решений, который бы позволил спрогнозировать, попадет ли потенциальный фильм в одну из следующих трех категорий: «Успех у критиков», «Успех в прокате» или «Полный провал».

Чтобы построить дерево решений, вы обращаетесь к архивам студии с целью изучить причины успеха и провала 30 последних фильмов компании. Вы быстро замечаете связь между бюджетом фильма, количеством знаменитостей, снявшихся в главных ролях, и успешностью фильма. Придя в восторг от этого открытия, вы строите диаграмму рассеяния (рис. 5.2).

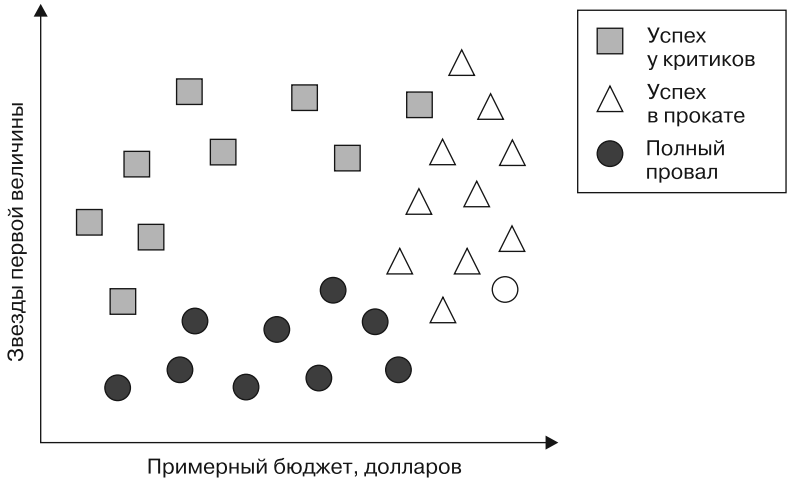

Рис. 5.2. Диаграмма рассеяния, отражающая взаимосвязь между бюджетом фильма и количеством снявшихся в нем знаменитостей

Используя стратегию «разделяй и властвуй», на основе данных можно построить простое дерево решений. Прежде всего, чтобы создать корневой узел дерева, разделим признак, указывающий на количество знаменитостей, разбивая фильмы на группы: те, где снималось много звезд первой величины, и те, где знаменитостей было мало (рис. 5.3).

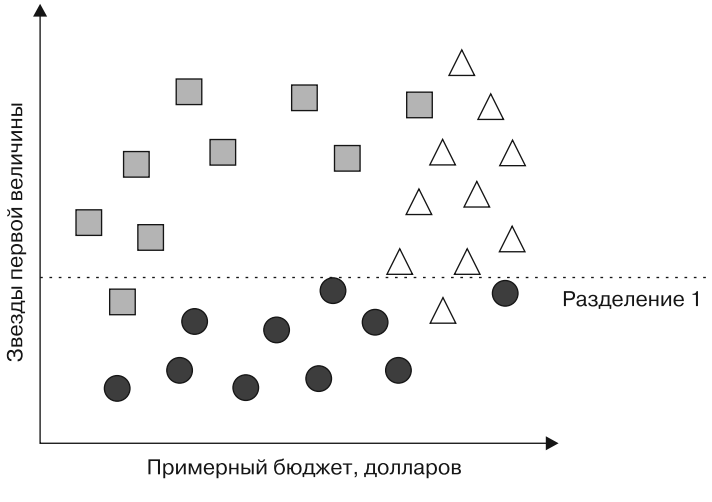

Рис. 5.3. Первое разделение дерева решений: фильмы разделены на те, в которых снималось много и мало знаменитостей

Затем группу фильмов с большим количеством знаменитостей можно разделить на еще одну: фильмы с большим и маленьким бюджетом (рис. 5.4).

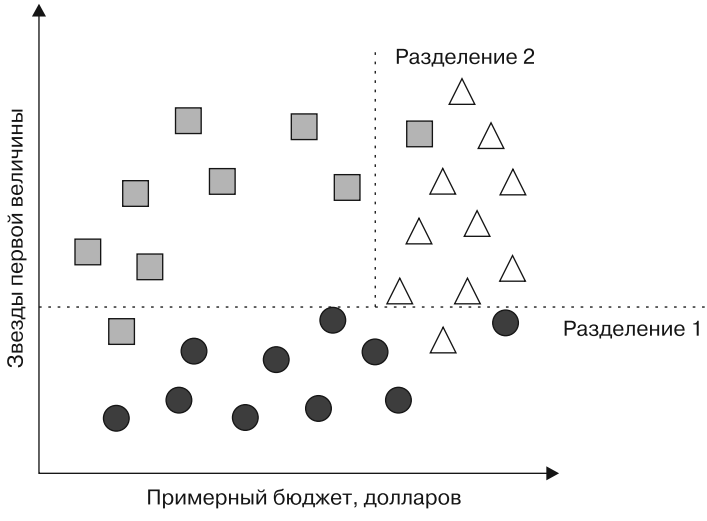

Рис. 5.4. Второе разделение дерева решений: фильмы с большим количеством знаменитостей делятся на фильмы с большим и маленьким бюджетом

В этой точке мы разделили данные на три группы. Группа, попадающая в верхний левый угол диаграммы, полностью состоит из фильмов, получивших признание у критиков. Эта группа отличается большим количеством знаменитостей и относительно низким бюджетом. В верхнем правом углу большинство фильмов — кассовые хиты с большим бюджетом и большим количеством знаменитостей. К последней группе относятся провальные фильмы, в которых снималось немного звезд, но бюджеты которых варьируются от маленьких до больших.

Если бы мы захотели, то могли бы продолжить разделять данные и властвовать над ними, разбивая их на все более специфичные диапазоны по бюджету и количеству знаменитостей, пока все неправильно классифицированные значения не будут классифицированы правильно в своем маленьком разделе. Однако не рекомендуется делать дерево решений таким сверхчувствительным. Конечно, ничто не мешает алгоритму разбивать данные до бесконечности, однако слишком специфичные решения не всегда позволяют делать более широкие обобщения. Чтобы избежать переобучения, мы остановим алгоритм на этом уровне, так как более 80 % примеров в каждой группе принадлежат к одному классу. Это будет основой для критерия остановки алгоритма.

Возможно, вы заметили, что диагональные линии могли бы разделить данные еще более четко. Но это одно из ограничений представления знаний в виде дерева решений: здесь используются только *разбиения, параллельные осям координат*. Тот факт, что при каждом разбиении учитывается только один признак, не позволяет дереву формировать более сложные границы решений. Например, диагональная линия могла бы соответствовать следующему решению: «Превышает ли количество знаменитостей предполагаемый бюджет?» И если да, то результатом будет «успех у критиков».

Модель для прогнозирования успешности фильмов может быть представлена в виде простого дерева, как показано на рис. 5.5.

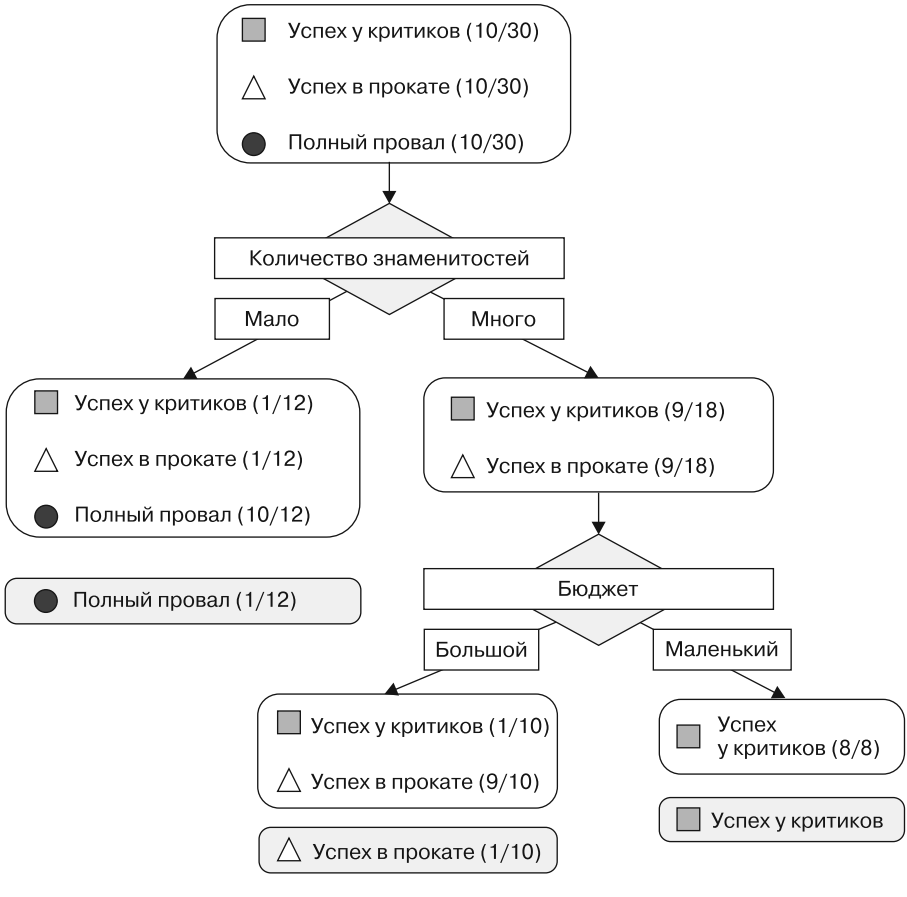

Рис. 5.5. Дерево решений, построенное на данных об уже снятых фильмах, позволяет спрогнозировать успешность будущих фильмов

Каждый шаг в дереве показывает, какая доля примеров попадает в соответствующий класс. На каждом этапе видно, как данные становятся все более однородными по мере приближения ветвей к концевым узлам. Для того чтобы оценить новый сценарий фильма, достаточно следовать указаниям по каждому решению, пока не будет спрогнозирована успешность или неудача сценария. Используя этот подход, можно быстро определить наиболее перспективные варианты среди всех сценариев и вернуться к более важной работе, такой как написание речи для выступления на церемонии вручения «Оскара».

Поскольку реальные данные содержат более двух признаков, деревья решений быстро становятся намного сложнее, чем с большим количеством ветвей, промежуточных и концевых узлов. В следующем разделе мы познакомимся с популярным алгоритмом автоматического построения моделей дерева решений.

Алгоритм дерева решений C5.0

Существует множество реализаций деревьев решений. Одним из самых известных является *алгоритм C5.0*. Этот алгоритм разработан специалистом по компьютерным системам Дж. Россом Квинланом (J. Ross Quinlan). Это усовершенствованная версия его предыдущего алгоритма *C4.5*, который, в свою очередь, является улучшенным вариантом его же алгоритма *Iterative Dichotomiser 3 (ID3)*. Квинлан продает C5.0 коммерческим клиентам (подробности вы найдете на сайте http://www.rulequest.com/), однако исходный код однопоточной версии этого алгоритма опубликован и поэтому включен в такой программный продукт, как R.

Чтобы еще больше сбить вас с толку, в R-пакет RWeka входит J48 — популярная альтернатива C4.5 с открытым исходным кодом на основе Java. Поскольку различия между C5.0, C4.5 и J48 незначительны, принципы, изложенные в этой главе, применимы к любому из этих трех методов и алгоритмы следует считать синонимичными.

Алгоритм C5.0 стал отраслевым стандартом для построения деревьев решений, поскольку он без дополнительной настройки хорошо подходит для решения большинства типов задач. По сравнению с другими расширенными моделями машинного обучения, такими как описанные в главе 7, деревья решений, построенные на базе C5.0, обычно работают почти так же хорошо, но их гораздо легче понять и реализовать. Кроме того, как показано в табл. 5.1, недостатки этого алгоритма сравнительно невелики и их по большей части можно избежать.

Таблица 5.1

Преимущества	Недостатки
• Универсальный классификатор, который хорошо справляется со многими типами задач. • Высокоавтоматизированный процесс обучения, позволяющий обрабатывать числовые и номинальные характеристики, а также отсутствующие данные. • Исключает неважные признаки. • Может использоваться и для маленьких, и для больших наборов данных. • Приводит к построению модели, которая может быть интерпретирована без математического образования (для относительно небольших деревьев). • Эта модель эффективнее, чем другие, более сложные модели	• Модели дерева решений часто смещены в сторону разделения по признакам, имеющим большое количество уровней. • Легко получить как переобученную, так и недостаточно обученную модель. • При моделировании некоторых отношений возможны проблемы из-за разделений только вдоль осей координат. • Незначительные изменения тренировочных данных могут привести к значительным изменениям в логике принятия решений. • Большие деревья бывает трудно интерпретировать, а решения, которые они рекомендуют, могут показаться нелогичными

Для простоты в предыдущем примере дерева решений игнорировалась математика, связанная с тем, как машина использует стратегию «разделяй и властвуй». Рассмотрим это более подробно, чтобы изучить, как эвристика работает на практике.

Выбор лучшего разделения

Первая проблема, с которой сталкивается дерево решений, — определение того, по какому признаку осуществлять разделение. В предыдущем примере мы искали способ разделения данных таким образом, чтобы в полученных разделах содержались примеры, главным образом относящиеся к одному классу. Степень, в которой подмножество примеров содержит только один класс, называется *чистотой*, а подмножество, состоящее только из одного класса, называется *чистым*.

Существуют различные показатели чистоты, которые можно использовать для определения наилучшего кандидата для разделения дерева решений. В C5.0 используется *энтропия* — концепция, заимствованная из теории информации, которая количественно определяет случайность, или беспорядочность, множества значений класса. Множества с высокой энтропией очень разнообразны и предоставляют мало информации о других элементах, которые также могут принадлежать к этому множеству, поскольку между ними нет очевидной общности. Дерево решений старается найти такие разделения, которые уменьшают энтропию, в итоге повышая однородность внутри групп.

Обычно энтропия измеряется в *битах*. Если существует только два возможных класса, то энтропия принимает значения 0 и 1. Для n классов энтропия принимает значения от 0 до $\log_2(n)$. В каждом случае минимальное значение указывает на то, что выборка является полностью однородной, а максимальное — на то, что данные настолько разнообразны, насколько это возможно, и ни у одной группы нет ничего общего.

В математическом представлении энтропия (Entropy) определяется следующим образом:

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \log_2(p_i).$$

В этой формуле для выбранного сегмента данных *(S)* c означает число уровней классов, а p_i — долю значений, попадающих в класс уровня i. Например, предположим, что у нас есть раздел данных с двумя классами: красный (60 %) и белый (40 %). Тогда энтропию можно рассчитать следующим образом:

```
> -0.60 * log2(0.60) - 0.40 * log2(0.40)
[1] 0.9709506
```

Мы можем визуализировать энтропию для всех возможных вариантов, состоящих из двух классов. Если известно, что доля примеров, относящихся к одному классу, равна x, то доля примеров, принадлежащих к другому классу, равна $(1 - x)$. Используя функцию `curve()`, можно построить график энтропии для всех возможных значений x:

```
> curve(-x * log2(x) - (1 - x) * log2(1 - x),
    col = "red", xlab = "x", ylab = "Entropy", lwd = 4)
```

В результате получим следующий график (рис. 5.6).

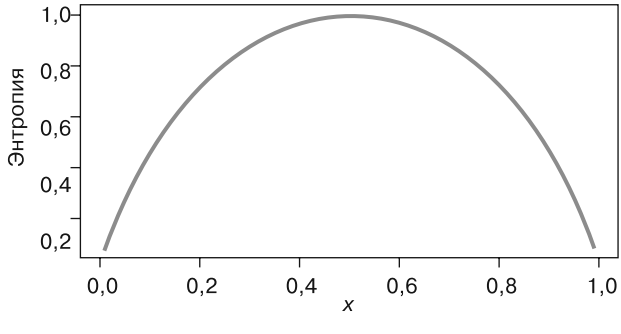

Рис. 5.6. Общая энтропия как пропорция изменений одного класса к результату двух классов

Как видно на графике, энтропия достигает пикового значения при $x = 0{,}50$: разделение на уровне 50–50 приводит к максимальной энтропии. Поскольку один класс все больше доминирует в другом, энтропия снижается до нуля.

Чтобы использовать энтропию для определения оптимального признака разделения, алгоритм рассчитывает изменение однородности, которое может возникнуть при разделении по всем возможным признакам. Эта мера называется *приростом информации*. Прирост информации (infogain) для признака F вычисляется как разность между энтропией в данном сегменте до разделения (S_1) и в сегментах, полученных в результате разделения (S_2):

$$\text{InfoGain}(F) = \text{Entropy}(S_1) - \text{Entropy}(S_2).$$

Сложность этого подхода заключается в том, что в результате разделения создается несколько сегментов данных. Таким образом, функция вычисления энтропии $\text{Entropy}(S_2)$ должна учитывать общую энтропию для всех сегментов. Ее можно получить путем взвешивания энтропии каждого сегмента в соответствии с долей всех записей, попадающих в этот сегмент. Это можно выразить в формуле:

$$\text{Entropy}(S) = \sum_{i=1}^{n} w_i \text{Entropy}(P_i).$$

Проще говоря, общая энтропия, полученная в результате разделения, представляет собой сумму энтропии каждого из n сегментов, взвешенную в соответствии с долей примеров, попадающих в этот сегмент (w_i).

Чем больше прирост информации, тем лучше данный признак подходит для создания однородных групп после разделения по этому признаку. Если прирост информации равен нулю, то при разделении по этому признаку энтропия не уменьшается. Однако максимальный прирост информации равен энтропии до разделения. Таким образом, если энтропия после разделения равна нулю, это означает, что разделение привело к созданию полностью однородных групп.

Приведенные выше формулы рассчитаны на номинальные признаки, однако деревья решений также позволяют использовать прирост информации для разделения по числовым признакам. Для этого обычно проверяют различные варианты разделений, при которых значения делятся на группы, где значения данного признака больше или меньше некоего порогового значения. Это позволяет свести числовой признак к двухуровневому категориальному признаку, после чего вычислять прирост информации как обычно. Для разделения выбирается числовая точка среза, обеспечивающая максимальный прирост информации.

Прирост информации, используемый в C5.0, не единственный критерий разделения, который можно использовать для построения деревьев решений. Другими часто используемыми критериями являются индекс Джини, критерий хи-квадрата и коэффициент усиления. Обзор этих (и многих других) критериев вы найдете в статье: Mingers J. An Empirical Comparison of Selection Measures for Decision-Tree Induction. Machine Learning, 1989. Vol. 3. P. 319–342.

Сокращение дерева решений

Как уже отмечалось, дерево решений может расти бесконечно, выбирая признаки разделения и ветвясь на все меньшие сегменты, пока не будут полностью классифицированы все примеры или пока в алгоритме не исчерпаются возможности для разделения. Но, когда дерево становится слишком большим, многие принимаемые им решения оказываются чрезмерно конкретными, и модель становится переобученной. Процесс *сокращения* дерева решений означает уменьшение его размера таким образом, чтобы дерево лучше обобщало новые данные.

Одним из решений этой проблемы является остановка роста дерева при достижении определенного числа точек принятия решений или когда узлы принятия решений будут содержать лишь небольшое количество примеров. Это называется *ранней остановкой* или *ранним сокращением* дерева решений. Поскольку при построении дерева алгоритм стремится избегать ненужной работы, это привлекательная стратегия. Одним из недостатков такого подхода является то, что не существует способа узнать, не будут ли при этом пропущены неявные, но важные паттерны, которые были бы изучены, если бы дерево выросло до большего размера.

Альтернатива, называемая *поздним сокращением*, подразумевает построение намеренно слишком большого дерева и сокращение концевых узлов, чтобы уменьшить размер дерева до приемлемого. Такой подход часто оказывается более эффективным, чем ранняя остановка, поскольку бывает довольно сложно определить оптимальную глубину дерева решений заранее, без предварительного увеличения. Последующее сокращение дерева позволяет алгоритму гарантировать, что были обнаружены все важные структуры данных.

Подробности реализации операций сокращения дерева очень специфичны и выходят за рамки этой книги. Сравнение некоторых существующих методов вы найдете в статье: Esposito F., Malerba D., Semeraro G. A Comparative Analysis of Methods for Pruning Decision Trees // IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997. Vol. 19. P. 476–491.

Одним из преимуществ алгоритма C5.0 является то, что сокращение в нем сильно автоматизировано: алгоритм принимает многие решения самостоятельно, используя весьма разумные значения, предлагаемые по умолчанию. Общая стратегия этого алгоритма заключается в позднем сокращении дерева. Сначала строится большое дерево, соответствующее тренировочным данным. Затем узлы и ветви, которые мало влияют на ошибки классификации, удаляются. В некоторых случаях целые ветви перемещаются вверх по дереву или заменяются более простыми реше-

ниями. Эти процессы пересадки ветвей называются *поднятием поддерева* и *заменой поддерева* соответственно.

Достижение правильного баланса между переобучением и недообучением дерева — это в некотором роде искусство. Однако, если точность модели жизненно важна, возможно, стоит потратить некоторое время на исследование различных вариантов сокращения, чтобы посмотреть, не повысится ли от этого эффективность распознавания тестового набора данных. Как вы вскоре увидите, одним из преимуществ алгоритма C5.0 является то, что он очень легко позволяет настроить параметры обучения.

Пример: распознавание рискованных банковских кредитов с помощью деревьев решений C5.0

Мировой финансовый кризис 2007–2008 годов показал, как важна прозрачность и строгость в принятии банковских решений. Когда кредиты стали менее доступными, банки ужесточили систему кредитования и обратились к машинному обучению для более точного определения рискованных кредитов.

Благодаря высокой точности и возможности формулировать статистическую модель на понятном человеку языке деревья решений широко применяются в банковской сфере. Поскольку правительства многих стран тщательно следят за справедливостью кредитования, руководители банков должны быть в состоянии объяснить, почему одному заявителю было отказано в получении займа, в то время как другому одобрили выдачу кредита. Эта информация полезна и для клиентов, желающих узнать, почему их кредитный рейтинг оказался неудовлетворительным.

Похоже, что автоматические модели оценки кредитоспособности используются для рассылок по кредитным картам и мгновенных онлайн-процессов одобрения кредитов. В этом разделе мы разработаем простую модель принятия решения о предоставлении кредита с использованием алгоритма построения деревьев решений C5.0. Вы также узнаете, как настроить параметры модели, чтобы свести к минимуму ошибки, которые могут привести к финансовым потерям.

Шаг 1. Сбор данных

Цель кредитной модели — выявить факторы, связанные с более высоким риском невозвращения кредита. Для этого необходимо получить данные о большом количестве предыдущих банковских кредитов и информацию о получателях этих кредитов.