Содержание

От рецензента

На текущий момент лучшими алгоритмами для работы со структурированными данными являются алгоритмы, основанные на деревьях принятия решений. Исходя из моего опыта, порядка 70% практических задач решаются ими, остальное остается задачам с неструктурированными данными, такими как изображения и текст, – тут побеждают нейронные сети. Также опыт соревнований по машинному обучению показывает, что в соревнованиях со структурированными данными деревья (а точнее, бустинг над решающими деревьями – gradient boosted decision trees) властвуют безраздельно.

Алгоритмы, основанные на деревьях принятия решений, «всеядны» – они умеют работать с числовыми, так и с категориальными признаками, причем данные в современных реализациях не нуждаются в предварительной обработке, такой как нормализация или заполнение пропусков. Они универсальны, например для алгоритма случайного леса есть варианты для решения задач регрессии, классификации, поиска аномалий, кластеризации, селекции признаков и т. д.

Все это делает случайный лес и градиентный бустинг универсальным инструментом, овладеть которым необходимо каждому специалисту.

Эта книга уникальна прежде всего широким охватом программных продуктов, языков программирования и прикладных пакетов, и я совершенно согласен с автором, что не стоит искусственно ограничивать себя чем-то одним, когда можно использовать лучшее отовсюду.

Второй момент, который хочется отметить, – это наличие большого числа примеров и наборов данных, которые автор использует в своей книге.

Я очень рад, что подобные книги начинают издаваться на русском языке, и уверен, что данная книга найдет своего читателя.

Дмитрий Ларько, Kaggle Grandmaster

Предисловие

Данная книга открывает серию пособий, посвященных практическому применению методов машинного обучения на базе популярных статистических пакетов IBM SPSS Statistics и R. В первом выпуске освещаются методы деревьев решений и случайного леса. Деревья решений – это эффективный метод машинного обучения, использующийся в прогнозном моделировании. Кроме того, при решении задач бинарной классификации он нередко дополняет метод логистической регрессии. Аналитики кредитного бюро TransUnion для построения скоринговых моделей используют логистическую регрессию, а для отбора переменных в модель логистической регрессии (рассматриваются сотни переменных) – деревья решений. Наша компания при построении прогнозных моделей на основе логистической регрессии использует метод деревьев решений, чтобы сформировать новые переменные для лучшего прогнозирования дефолта. Аналитики Citibank USA разбивают популяцию заемщиков на сегменты, применяя дерево решений, а затем в каждом сегменте строят модели доходности с помощью линейной регрессии или модели риска с помощью логистической регрессии. Итогом развития метода деревьев решений стал случайный лес, который является на сегодняшний день одним из популярнейших методов машинного обучения. Он представляет собой комитет деревьев решений и за счет использования рандомизации и усреднения позволяет получить более точные прогнозы.

Кому стоит прочитать эту книгу

Данная книга адресована действующим и начинающим специалистам по машинному обучению, решающим задачи прогнозирования риска невыплаты кредита, оттока клиента, отклика покупателя на маркетинговое предложение и т. д.

Часто я становлюсь свидетелем жарких споров по поводу того, какое программное обеспечение лучше использовать для машинного обучения – проприетарное или с открытым программным кодом, что выбрать – неограниченные возможности моделирования, сопряженные с необходимостью постоянной отладки программного кода, или стабильность и удобство, ограниченные набором «кнопок». Эта книга призвана в определенной степени примирить обе стороны. С одной стороны, растет число студентов старших курсов и аспирантов, обучающихся по специальности «компьютерные науки». Они получают все необходимые знания, позволяющие им без труда научиться анализировать данные в R и Python. С другой стороны, в настоящее время машинное обучение активно используется в маркетинге, социологии, медицине и психологии. У действующих маркетологов, социологов, врачей, психологов, не обладающих соответствующими навыками программирования и математическими знаниями, есть настоятельная потребность применять машинное обучение здесь и сейчас, не тратя годы на изучение языков программирования, матричной алгебры, математической статистики и теории вероятностей. Поэтому проприетарное программное обеспечение является в этой ситуации определенным вариантом решения проблемы. Вместе с тем предпринятое в этой книге доступное описание возможностей R и Python позволит пользователям без должного математического бэкграунда оценить всю мощь и гибкость программного обеспечения с открытым исходным кодом.

Эта книга не требует предварительных знаний в области машинного обучения и статистического анализа. Я приложил максимальные усилия, направленные на то,

чтобы дать адаптированное описание математического аппарата методов деревьев решений и случайного леса и сосредоточиться в большей степени на практических аспектах использования данных методов.

Структура книги

В этой книге я детально расскажу о том, как строить модели дерева решений или модели случайного леса, интерпретировать результаты, оценивать качество полученных моделей, улучшать его, сохранять результаты и применять правила классификации/прогноза, полученные с помощью дерева или случайного леса, к новым данным. Также я расскажу о том, как с помощью дерева решений и случайного леса улучшить модель логистической регрессии. Отдельно рассматривается вопрос автоматизированного поиска оптимальных параметров случайного леса.

Глава 1 кратко знакомит с терминологией метода деревьев решений, в ней рассказывается о преимуществах и недостатках деревьев, задачах, которые можно выполнить с их помощью.

Главы 2–6 посвящены построению деревьев решений и случайного леса в IBM SPSS Statistics 24.0. В главе 2 освещается CHAID – один из самых популярных методов деревьев решений. В главе 3 я покажу, как можно менять параметры дерева CHAID, влияя на результаты классификации. Здесь же я расскажу о том, как можно выполнить биннинг переменных для включения в модель логистической регрессии, использовав дерево решений CHAID и случайный лес. Для иллюстрации выбрана конкурсная задача предсказания отклика ОТП Банка. Кроме того, на данном примере я покажу, как выполняется предварительная подготовка данных и решаются вопросы, связанные с автоматизацией построения моделей (для этого будет использован командный синтаксис SPSS). Код, автоматизирующий процесс построения прогнозных моделей, вы можете в дальнейшем использовать в собственных проектах. В этой же главе будет рассмотрена разработка ансамбля модели логистической регрессии и дерева CHAID. Глава 4 посвящена методам деревьев CRT и QUEST. В главе 5 рассказывается о редакторе дерева. Глава 6 посвящена методу случайного леса. В ней я расскажу о методологической основе случайного леса, приведу примеры использования случайного леса для решения задач классификации и регрессии, покажу, как применять модель случайного леса к новым данным.

Главы 7–11 посвящены построению деревьев решений и случайного леса в R.

В главе 7 я подробно рассмотрю процесс построения и интерпретации дерева решений CHAID в пакете `CHAID`. В главе 8 я применю пакет `rpart`, чтобы построить и проанализировать дерево решений CRT. В главе 9 я покажу, как можно построить модель случайного леса, интерпретировать ее и применить к новым данным, используя пакет `randomForest`. В ней же будет рассказано как осуществлять оптимизацию параметров случайного леса с помощью пакета `caret`. Глава 10 посвящена пакету `ranger` – быстрой реализации случайного леса в R, позволяющей работать с большими и высокоразмерными наборами данных. Кроме того, в этом пакете реализована возможность использовать случайный лес не только для решения задач регрессии и классификации, но и для анализа выживаемости. В главе 11 я расскажу о пакете `h2o`, который позволяет использовать в среде R возможности платформы H2O, разработанной для работы с большими данными. Речь пойдет об алгоритме случайного леса, использующего распределенные вычисления и новейшие эвристики, позволяющие в ряде случаев получить лучшее качество модели.

В главе 12 речь пойдет об использовании классов `DecisionTreeClassifier`, `DecisionTreeRegressor`, `RandomForestClassifier` и `RandomForestRegressor`, реализованных в пито-

новской библиотеке `scikit-learn` и предназначенных для построения дерева классификации, дерева регрессии, ансамбля деревьев классификации и ансамбля деревьев регрессии соответственно. В этой же главе будет рассмотрена работа с питоновским модулем `h2o` и построение распределенного случайного леса с помощью класса `H2ORandomForestEstimator`.

В приложениях 1 и 2 дается обзор различных операций по предварительной подготовке данных в Python и R соответственно (создание новых переменных, импутация пропущенных значений, работа со строками и т. д.). Они представляют собой примеры, следующие в строгой последовательности и предназначены для методического выполнения, чтобы помочь вам сформировать базовые навыки по подготовке данных, именно на нее аналитики тратят до 80% своего рабочего времени. В приложении 3 дается обзор возможностей визуализации данных в Python.

Наборы данных и примеры программного кода

Наборы данных и примеры программного кода, использующие в книге, находятся в папке **Trees**. Папку **Trees** можно скачать в заархивированном виде на сайте издательства «ДМК Пресс» и распаковать в корневой каталог диска. Все вопросы, связанные с некорректной работой программного кода, можно направлять по адресу info@gewissta.ru.

Программное обеспечение, используемое в этой книге

Всю необходимую информацию об IBM SPSS Statistics вы найдете на официальном сайте компании IBM (http://www-03.ibm.com/software/products/ru/spss-stats-base). Информацию о программном пакете R можно найти на официальной странице проекта R (https://www.r-project.org/). Если говорить о работе в Python, то я рекомендую пользоваться Anaconda для Python 3.6. Anaconda (https://www.continuum.io/downloads) – это дистрибутив Python, предназначенный для крупномасштабной обработки данных, прогнозной аналитики и научных вычислений. Anaconda *уже включает* библиотеки Python, необходимые для предварительной подготовки данных, машинного обучения и комфортной работы с полученными результатами (NumPy, SciPy, `matplotlib`, `pandas`, IPython, Jupyter Notebook и `scikit-learn`). Есть версии Anaconda для Mac OS, Windows и Linux. Это очень удобное решение, и это тот дистрибутив, который я рекомендую пользователям, у которых еще не установлены вышеупомянутые библиотеки Python.

Благодарности

В заключение я хочу поблагодарить Дмитрия Майорова (Citibank N.A., ArrowModel), Барри Уилка (Google), Дмитрия Ларко (H2O), Максима Савченко (Сбербанк-Технологии) за их ценные советы и замечания, высказанные в ходе подготовки книги. Я также выражаю признательность Антону Вахрушеву (Сбербанк) и сообществу Open Data Science за рекомендации по подготовке главы, посвященной Python. Также я благодарен издательству «ДМК Пресс», в частности, Дмитрию Мовчану и Анне Чанновой за то, что они выдержали мои бесконечные правки, вызванные стремлением максимально улучшить книгу.

Артем Груздев,
генеральный директор ИЦ «Гевисста»

Глава 1

Введение в метод деревьев решений

1.1. Введение в методологию деревьев решений

Как и регрессионный анализ, деревья решений являются методом изучения статистической взаимосвязи между одной зависимой переменной и несколькими независимыми (предикторными) переменными. Базовое отличие метода деревьев решений от регрессионного анализа заключается в том, что взаимосвязь между значением зависимой переменной и значениями независимых переменных представлена не в виде общего прогнозного уравнения, а в виде древовидной структуры, которую получают с помощью иерархической сегментации данных.

Берется весь обучающий набор данных, называемый **корневым узлом**, и разбивается на два или более **узлов (сегментов)** так, чтобы наблюдения, попавшие в разные узлы, максимально отличались друг от друга по зависимой переменной (например, выделяем два узла с наибольшим и наименьшим процентом «плохих» заемщиков). В роли **правил разбиения**, максимизирующих эти различия, выступают значения независимых переменных (пол, возраст, доход и др.). Качество разбиения оценивается с помощью статистических критериев. Правила и статистики отмечаются на **ветвях** – линиях, которые соединяют разбиваемый узел с узлами, полученными в результате разбиения. Для каждого узла вычисляются **вероятности** в виде **процентных долей** категорий зависимой переменной (если зависимая переменная является категориальной) или средние значения зависимой переменной (если зависимая переменная является количественной). В результате выносится **решение** – спрогнозированная категория зависимой переменной (если зависимая переменная является категориальной) или спрогнозированное среднее значение зависимой переменной (если зависимая переменная является количественной).

Аналогичным образом каждый узел, получившийся в результате разбиения корневого узла, разбивается дальше на узлы, то есть узлы внутри узла, и т. д. Этот процесс продолжается до тех пор, пока есть возможность разбиения на узлы. Данный процесс сегментации называется **рекурсивным разделением**. Получившаяся иерархическая структура, характеризующая взаимосвязь между значением зависимой переменной и значениями независимых переменных, называется **деревом**.

Иногда для обозначения разбиваемого узла применяется термин **родительский узел**. Новые узлы, получившиеся в результате разбиения, называются **дочерними**

узлами (или узлами-потомками). Когда впоследствии дочерний узел разбивается сам, он становится родительским узлом. Окончательные узлы, которые в дальнейшем не разбиваются, называются **терминальными узлами** дерева. Их еще называют **листьями**, потому что в них рост дерева останавливается. Лист представляет собой наилучшее окончательное решение, выдаваемое деревом. Здесь мы определяем группы клиентов, обладающие желаемыми характеристиками (например, тех, кто погасит кредит или откликнется на наше маркетинговое предложение).

Обратите внимание, если вы прогнозируете вероятность значения категориальной зависимой переменной по соответствующим значениям предикторов, дерево решений называют **деревом классификации** (рис. 1.1). Например, дерево классификации строится для вычисления вероятности дефолта у заемщика (на основе спрогнозированной вероятности мы относим его к «плохому» или «хорошему» заемщику). Если дерево решений используется для того, чтобы спрогнозировать среднее значение количественной зависимой переменной по соответствующим значениям предикторов, его называют **деревом регрессии** (рис. 1.2). Например, дерево регрессии строится, чтобы вычислить средний размер вклада у клиента.

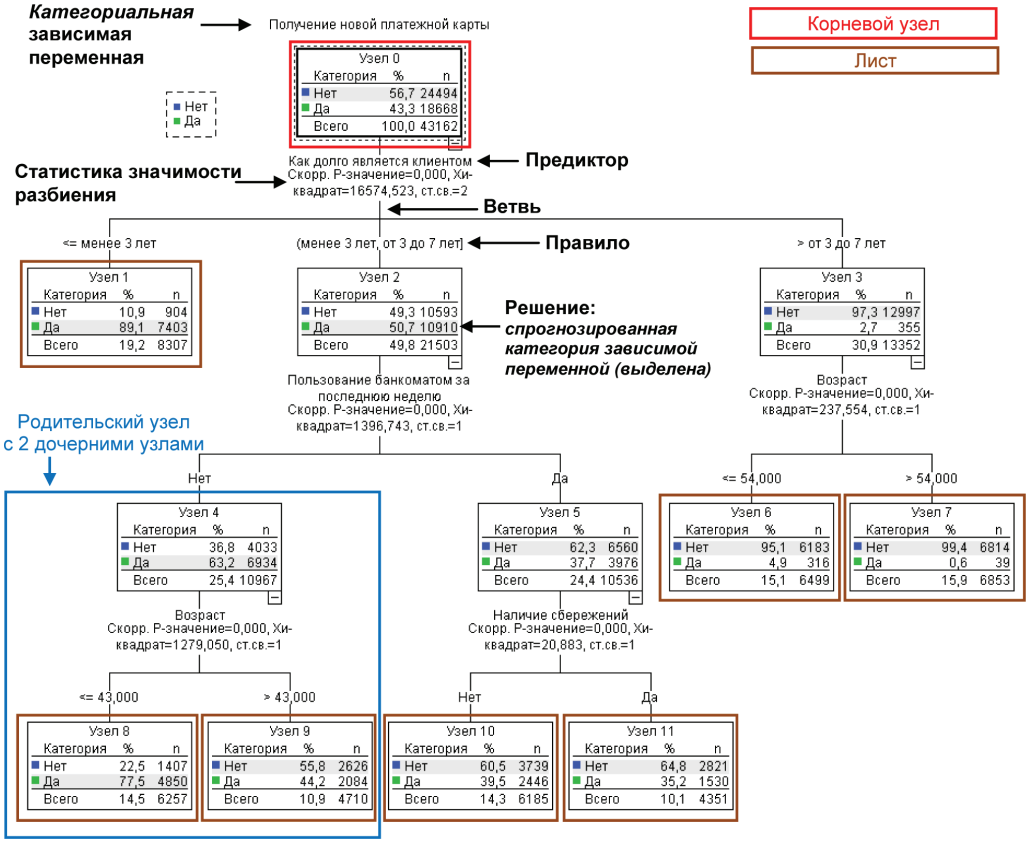

Рис. 1.1 ❖ Пример дерева классификации

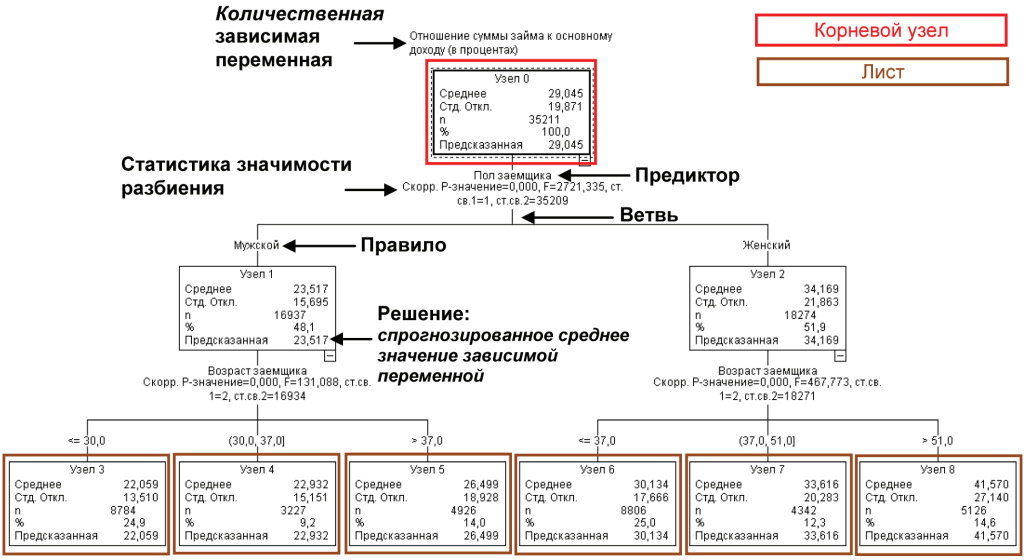

Рис. 1.2 ❖ Пример дерева регрессии

Если визуализировать работу алгоритма дерева решений, то мы увидим, что алгоритм последовательно разбивает данные на прямоугольники, параллельные осям координат. Проиллюстрирум это на примере бинарного дерева решений, то есть когда узел-родитель может иметь только два узла-потомка.

У нас есть набор данных, состоящий из 32 наблюдений, предикторами являются переменные *Длительность звонков в минутах* и *Количество обращений в службу поддержки*, зависимая переменная – *Статус клиента*: 18 клиентов относятся к классу *Ушедший*, а 14 клиентов – к классу *Оставшийся*.

Рис. 1.3 ❖ Визуальное представление набора данных

Первое разбиение набора данных происходит по предиктору *Количество обращений в службу поддержки* (рис. 1.4).

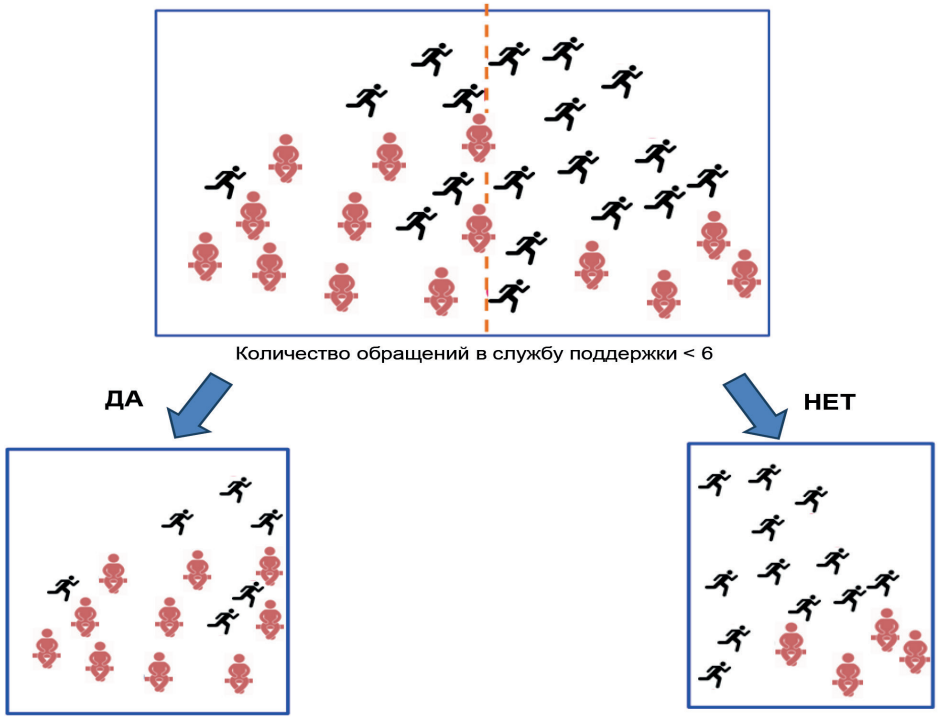

Рис. 1.4 ❖ Разбиение набора данных по предиктору *Количество обращений в службу поддержки*

Затем каждый из полученных узлов разбивается по предиктору *Длительность звонков в минутах* и дерево останавливается в росте (рис. 1.5).

Таким образом, можно сделать вывод, что если клиент обращался в службу поддержки 6 раз и более и при этом делал звонки длительностью 15 минут и более, его можно отнести к ушедшему клиенту. Вполне возможно, что постоянно возникающие проблемы в оказании услуг (об этом свидетельствует факт частого обращения в службу поддержки) при высокой интенсивности использования сотового телефона стали причиной оттока. Ниже приводится рисунок, на котором показаны все границы принятия решений, предложенные деревом для нашего набора данных.

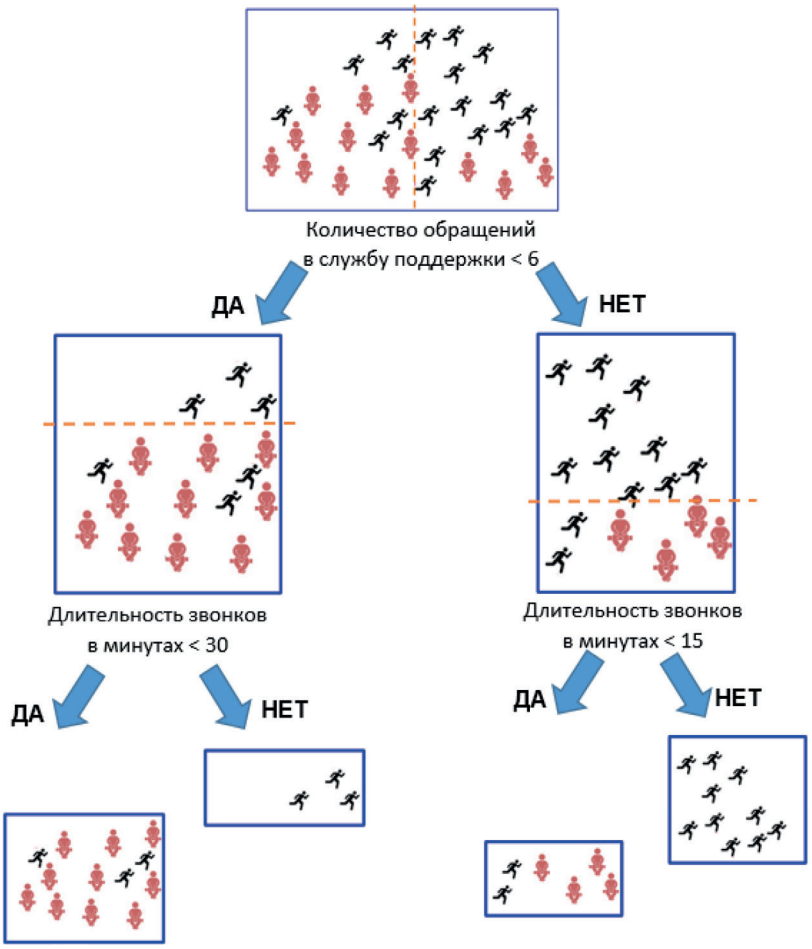

Рис. 1.5 ❖ Процесс последовательного разбиения набора данных

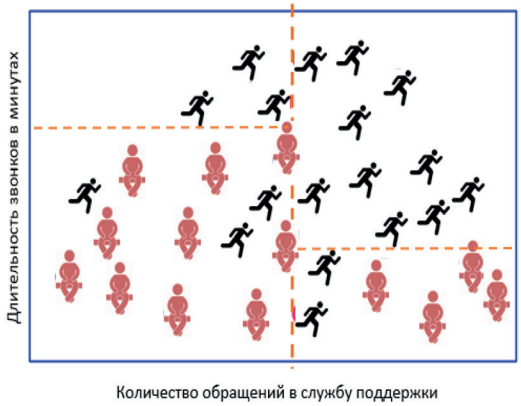

Рис. 1.6 ❖ Границы принятия решений, предложенные деревом

1.2. Преимущества и недостатки деревьев решений

Метод деревьев решений обладает рядом преимуществ. Главное из них – это наглядность представления результатов (в виде иерархической структуры дерева). Деревья решений позволяют работать с большим числом независимых переменных. На вход можно подавать все существующие переменные, алгоритм сам выберет наиболее значимые среди них, и только они будут использованы для построения дерева (автоматический отбор предикторов). Деревья решений способны выявлять нелинейные взаимосвязи, сложные взаимодействия, которые нелегко обнаружить в рамках стандартных статистических моделей. Они могут работать с любым типом переменной, таким образом, зависимые и независимые переменные могут быть количественными, порядковыми и номинальными. Деревья решений устойчивы к выбросам, поскольку разбиения основаны на количестве наблюдений внутри диапазонов значений, выбранных для расщепления, а не на абсолютных значениях. Перед построением модели необязательно импутировать пропущенные значения, поскольку деревья используют собственные процедуры обработки пропусков. Требования, выдвигаемые методом деревьев решений к распределению переменных, не являются строгими.

К недостаткам метода деревьев решений можно отнести отсутствие простого общего прогнозного уравнения, выражающего модель (в отличие от регрессионного анализа). Другой недостаток заключается в том, что некоторым методам деревьев решений (например, CRT) свойственно переобучение. Речь идет о ситуации, когда деревья получаются слишком детализированными, имеют много узлов и ветвей, сложны для интерпретации, что требует специальной процедуры отсечения ветвей (она называется прунинг). В отличие от линейной регрессии дерево решений не умеет экстраполировать или делать прогнозы для наблюдений, лежащих вне диапазона обучающих данных (данных, на которых строится или обучается модель). Например, обучающие данные содержит предиктор x и зависимую переменную y. Диапазон переменной x включает в себя значения от 30 до 70. Если новые данные содержат наблюдение со значением предиктора x, равным 200, дерево выдаст неправильный прогноз. Впрочем, именно неспособность экстраполировать позволяет избежать появления экстремальных значений в случае обработки выбросов. Наконец, для методов одиночных деревьев характерна проблема множественных сравнений. Перед расщеплением узла дерево сравнивает различные варианты разбиения, число этих вариантов зависит от числа уровней предикторов, как правило, происходит смещение выбора в пользу переменных, у которых большее количество уровней. Все это обусловливает определенную нестабильность результатов. Небольшие изменения в наборе данных могут приводить к построению совершенно другого дерева. В силу иерархичности дерева изменения в верхних узлах ведут к изменениям во всех узлах, расположенных ниже. Отмечу, что в большей степени вышесказанное относится к методу CRT. Чтобы достигнуть удовлетворительной прогностической способности CRT, один из его разработчиков, Лео Брейман, пришел к идее случайного леса, когда из обучающего набора извлекаются случайные выборки (того же объема, что и исходный обучающий набор) с возвращением, по каждой строится дерево с использованием случайно отобранных предикторов, и затем результаты, полученные по каждому дереву, усредняются. Однако при таком подходе теряется главное преимущество деревьев решений – простота интерпретации.

1.3. Задачи, выполняемые с помощью деревьев решений

Прежде всего деревья решений используются в маркетинге для сегментации клиентской базы. Например, деревья позволяют определить, какие демографические группы имеют максимальный показатель отклика. Эту информацию можно использовать, чтобы максимизировать отклик при будущей прямой рассылке.

Кроме того, деревья применяются для задач прогнозирования и классификации, когда моделируется взаимосвязь между зависимой переменной и предиктором. С этой точки зрения деревья решений сравнивают с логистической регрессией и линейной регрессией. Деревья решений более эффективны, по сравнению с регрессионным анализом, в тех случаях, когда взаимосвязи между предикторами и зависимой переменной являются нелинейными, переменные имеют несимметричные распределения, наблюдается большое количество коррелирующих между собой переменных, взаимодействия высоких порядков, аномальные значения. Если же предпосылки регрессионного анализа выполняются, то логистическая регрессия (когда зависимая переменная является категориальной) или линейная регрессия (когда зависимая переменная является количественной) может дать лучший результат. Это обусловлено тем, что деревья пытаются описать линейную связь между переменными путем многократных разбиений по предикторам. CHAID делает это за счет расщепления сразу на несколько категорий, CRT и QUEST пытаются уловить эту связь посредством серии бинарных делений, и это может быть менее эффективно, по сравнению с подбором параметров в регрессионном анализе. Однако проблема заключается в том, что данные обычно содержат как линейные, так и нелинейные зависимости, переменные с симметричными и асимметричными распределениями. Поэтому опытный моделер может построить ансамбль логистической регрессии и дерева решений (при условии, что оно использует строгие статистические критерии для отбора предикторов разбиения узлов), чтобы скомпенсировать недостатки обеих методов. Ансамбли дерева решений CHAID и логистической регрессии используются в моделях оттока в телекоме, где данные часто характеризуются переменными с несимметричными распределениями.

В банковском скоринге деревья решений используются как вспомогательный инструмент при разработке модели логистической регрессии. Приведем конкретные примеры такого применения дерева.

В кредитном скоринге использование нескольких скоринговых карт для одного портфеля обеспечивает лучшее дифференцирование риска, чем использование одной скоринговой карты. Это характерно, когда нам приходится работать с разнородной аудиторией, состоящей из различных групп, и одна и та же скоринговая карта не может работать достаточно эффективно для всех. Например, в скоринге кредитных карточек выделяют сегменты «активные клиенты» и «неактивные клиенты», «клиенты в просрочке» и «клиенты, не имеющие просрочек». Переменные в таких сегментах будут сильно отличаться. Например, для активных кредитных карт утилизация будет сильной переменной, а для неактивных – слабой. И наоборот, может оказаться, что время неактивности для активных клиентов равно 0, а для неактивных клиентов время неактивности окажется сильной переменной. Для этих целей выполняют сегментацию клиентов. Первый способ сегментации – деление на группы на основе опыта и отраслевых знаний с последующей аналитической проверкой. Второй способ – это

сегментация с помощью статистических методов типа кластерного анализа или деревьев решений. При этом, по сравнению с кластерным анализом, деревья решений обладают преимуществом: они формулируют четкие правила выделения сегментов, а сами выделенные сегменты статистически значимо отличаются между собой по зависимой переменной. В дальнейшем для каждого из сегментов можно построить собственную модель логистической регрессии, разработать скоринговую карту и сформулировать кредитные правила. В Citibank USA является стандартной практикой делать дерево с двумя-тремя уровнями и в каждом узле подгонять свою модель логистической регрессии. В основе скорингового балла FICO также лежит сегментация на основе деревьев решений. Об эффективности использования сегментации в кредитном скоринге пишет в своей книге «Скоринговые карты для оценки кредитных рисков» известный эксперт по управлению рисками Наим Сиддики[1], а также один из разработчиков алгоритмов скоринга компании FICO Брюс Ходли[2].

С помощью деревьев решений из большого числа предикторов можно выбрать переменные, полезные для построения модели логистической регрессии. Например, из 100 переменных дерево включило в модель 25 переменных, таким образом, у нас появляется информация о том, какие переменные наверняка можно включить в модель логистической регрессии. Методы CRT и случайный лес позволяют вычислить важность переменных, использованных в модели дерева. Мы уже можем ранжировать переменные по степени полезности.

Деревья решений можно использовать для биннинга – перегруппировки категориального предиктора или дискретизации количественного предиктора с целью лучшего описания взаимосвязи с зависимой переменной. Например, при построении модели логистической регрессии часто обнаруживается, что взаимосвязи между количественным предиктором и интересующим событием являются нелинейными. Уравнение логистической регрессии, несмотря на нелинейное преобразование своего выходного значения (логит-преобразование), все равно моделирует линейные зависимости между предикторами и зависимой переменной. Возьмем пример нелинейной зависимости между стажем работы в банке и внутренним мошенничеством. Допустим, рассчитанный регрессионный коэффициент в уравнении логистической регрессии получился отрицательным. Это значит, что вероятность совершения внутреннего мошенничества с увеличением стажа работы уменьшается. Однако, выполнив разбивку переменной с помощью дерева CHAID на категории до 12 месяцев, от 12 до 36 месяцев, от 36 до 60 месяцев и больше 60 месяцев, стало видно, что зависимость между стажем и внутренним мошенничеством нелинейная. Первая (до 12 месяцев) и последняя (больше 60 месяцев) категории склонны к внутреннему мошенничеству, а промежуточные сегменты, наоборот, не склонны к внутреннему мошенничеству. После правильной разбивки переменной, проведенной с помощью дерева, связь между предиктором и зависимой переменной становится больше похожа на реальную.

Строя модель логистической регрессии, нередко приходится работать с предикторами, у которых большое количество категорий. Как правило, речь идет о географических переменных (регион, область регистрации, область фактического пребывания заемщика, область торговой точки, где клиент брал кредит) и переменных, фиксиру-

---

[1] *Сиддики Н.* Скоринговые карты для оценки кредитных рисков. М.: Манн, Иванов и Фабер, 2014.

[2] *Breiman L.* (2001). Statistical modeling: The two cultures.

ющих профессию или сферу занятости заемщика. Если включить такие переменные в модель логистической регрессии, то переменная с k категориями будет преобразована в $k - 1$ дамми-переменных, которые станут в модели логистической регрессии статистически незначимыми. Только представьте, сколько будет дамми-переменных, если у вас 4 географические переменных с 89 категориями. Исключение таких переменных из анализа также нерационально, поскольку они могут дать ценную информацию. Поэтому можно выполнить биннинг с целью укрупнения категорий, а можно построить по этим четырем переменным дерево решений. В результате дерево укрупнит категории переменных и скомбинирует переменные так, чтобы полученные комбинации характеристик максимизировали различия по зависимой переменной. Такую переменную, где категориями являются терминальные узлы дерева, можно включить в модель логистической регрессии.

Вопросы к главе 1

1. Терминальный узел – это:
   a) самый верхний узел, представляющий всю выборку наблюдений;
   b) узел, в котором рост дерева останавливается;
   c) любой расщепляемый узел;
   d) новый узел, появившийся в результате расщепления узла.

2. Деревья классификации строятся для:
   1) количественной зависимой переменной;
   2) категориальной зависимой переменной;
   3) порядковой зависимой переменной;
   4) номинальной зависимой переменной;
   5) бинарной зависимой переменной;
   6) любой зависимой переменной.

3. Деревья регрессии строятся для:
   a) количественной зависимой переменной;
   b) категориальной зависимой переменной;
   c) номинальной зависимой переменной;
   d) бинарной зависимой переменной;
   e) любой зависимой переменной.

4. В качестве правил разбиения используются значения:
   1) независимых переменных;
   2) зависимых переменных;
   3) категориальных переменных;
   4) количественных переменных.