Второе издание,
обновленное
и дополненное

ДМК издательство
ИЦ «Гевисста»

Артем Груздев

Прогнозное моделирование в IBM SPSS Statistics, R и Python

Метод деревьев решений и случайный лес

УДК 519.7:004.9IBM SPSS Statistics
ББК 21.18с
Г90

Груздев А. В.
Г90 Прогнозное моделирование в IBM SPSS Statistics, R и Python: метод деревьев решений и случайный лес. – М.: ДМК Пресс, 2018. – 642 с.: ил.

ISBN 978-5-97060-539-4

Данная книга представляет собой практическое руководство по применению метода деревьев решений и случайного леса для задач сегментации, классификации и прогнозирования. Каждый раздел книги сопровождается практическим примером. Кроме того, книга содержит программный код SPSS Syntax, R и Python, позволяющий полностью автоматизировать процесс построения прогнозных моделей. Автором обобщены лучшие практики использования деревьев решений и случайного леса от таких компаний, как Citibank N.A., Transunion и DBS Bank.

Издание будет интересно маркетологам, риск-аналитикам и другим специалистам, занимающимся разработкой и внедрением прогнозных моделей.

УДК 519.7:004.9IBM SPSS Statistics
ББК 21.18с

ISBN 978-5-97060-539-4

Содержание

Глава 2

Основы прогнозного моделирования с помощью деревьев решений CHAID

2.1. Запуск процедуры *Деревья классификации*

Дерево решений в IBM SPSS Statistics можно построить с помощью процедуры **Деревья классификации**. Для вызова процедуры **Деревья классификации** необходимо в меню **Анализ** выбрать **Классификация ⇒ Деревья классификации**.

Вы оказываетесь в главном диалоговом окне **Дерево решений** (рис. 2.1). В поле **Зависимая переменная** необходимо перенести одну зависимую переменную. Кнопка **Категории** позволяет включить/исключить из анализа категории зависимой переменной или задать их как целевые. Указание одной или нескольких категорий как целевых не влияет на модель дерева, оценки рисков и результаты классификации. В поле **Независимые переменные** необходимо перенести одну или несколько независимых переменных.

Параметр **Первая переменная принудительно** позволяет задать первую переменную из списка независимых переменных как первую переменную расщепления.

Поле **Переменная влияния** позволяет указать переменную, которая будет определять, насколько большое влияние данное наблюдение оказывает на процесс построения дерева. Наблюдения с более низкими значениями переменной влияния будут иметь меньшее влияние, а наблюдения с более высокими значениями – большее влияние. При этом значения переменной влияния должны быть положительными.

Выпадающий список **Метод построения** позволяет выбрать метод построения дерева.

В правой части окна находятся пять кнопок, используемых для настройки процедуры **Деревья классификации**.

Кнопка **Вывод** задает появление дерева решений и генерацию таблиц. Можно запросить дополнительную статистическую информацию о модели, графическую интерпретацию соответствующих статистик, также можно запросить генерацию правил классификации для модели в SPSS синтаксисе, в SQL или в обычном текстовом формате.

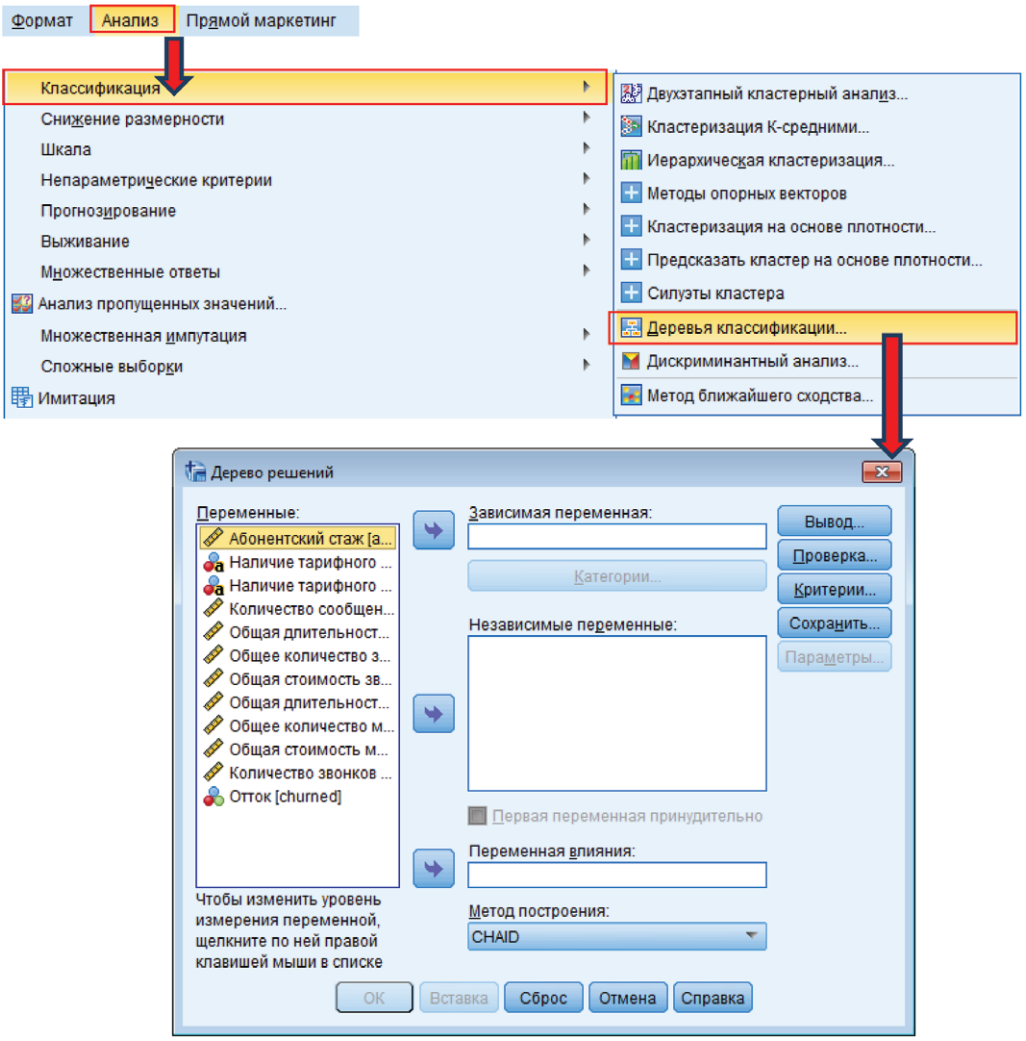

Рис. 2.1 ❖ Запуск процедуры **Деревья классификации**

Кнопка **Проверка** позволяет построить модель, отобрав только часть данных, и затем посмотреть, как она работает на оставшейся части, которая была исключена при построении модели.

Кнопка **Критерии** задает значения, которые используются в построении модели, такие как минимальное количество наблюдений в каждой группе или сегменте и уровень значимости, используемый в статистических тестах.

Кнопка **Сохранить** добавляет в активный набор данных переменные результатов анализа для каждого наблюдения:

- номер узла, к которому относится наблюдение;
- спрогнозированное значение зависимой переменной (для количественной зависимой переменной сохраняется спрогнозированное среднее значение, для категориальной зависимой переменной – спрогнозированная категория);

- спрогнозированные вероятности категорий зависимой переменной (только для категориальной зависимой переменной);
- принадлежность к обучающей или контрольной выборке.

Кнопка **Параметры** позволяет задать стоимости ошибочной классификации, априорные вероятности, прибыль и затраты по результатам классификации.

2.2. Четыре метода деревьев решений

Выпадающий список **Метод построения** в диалоговом окне **Дерево решений** (рис. 2.2) позволяет вам выбрать четыре метода деревьев решений: **CHAID** (используется по умолчанию), **Исчерпывающий CHAID**, **CRT**, **QUEST**.

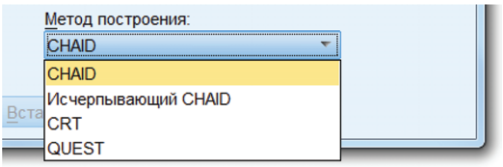

Рис. 2.2 ❖ Выпадающий список **Метод построения**

CHAID (расшифровывается как *Chi-square Automatic Interaction Detector – Автоматический обнаружитель взаимодействий*) используется процедурой **Деревья классификации** по умолчанию. Он был разработан Гордоном Каасом в 1980 году и представляет собой метод на основе дерева решений, который исследует взаимосвязь между предикторами и зависимой переменной с помощью статистических тестов.

Каждый раз для разбиения узла выбирается предиктор, сильнее всего взаимодействующий с зависимой переменной. При этом категории каждого предиктора объединяются, если они не имеют между собой статистически значимых отличий по отношению к зависимой переменной, остальные категории рассматриваются как отдельные. Для количественной зависимой переменной используется F-тест, для категориальной зависимой переменной – хи-квадрат Пирсона или хи-квадрат отношения правдоподобия.

Зависимая переменная и предикторы могут быть измерены в номинальной, порядковой и количественной шкалах, при этом количественные предикторы преобразовываются в порядковые переменные. CHAID позволяет осуществлять многомерные расщепления узлов. Каждый узел при разбиении может иметь более 2 потомков, поэтому CHAID имеет тенденцию выращивать более раскидистые деревья, чем бинарные методы. Вместе с тем из-за жестких статистических критериев расщепления нередко дерево CHAID получается нереалистично коротким и тривиальным («грубое» дерево), поэтому требуется тонкая настройка уровней значимости для объединения категорий и разбиения узлов. По сравнению с другими методами, CHAID характеризуется умеренным временем вычислений.

Помимо прочего, метод CHAID обладает собственным способом обработки пропущенных значений. Пропуски рассматриваются как отдельная фактическая категория. В ряде случаев это имеет смысл. Например, отказ отвечать на вопрос о доходе или занятости может оказаться предсказательной категорией для зависимой переменной.

Исчерпывающий CHAID является модификацией метода CHAID, предложенной Дэвидом Биггсом, Барри Де Виллем и Эдом Суеном в 1991 году. Он был разработан для устранения недостатка CHAID – ограниченного набора расщеплений для предиктора.

CHAID прекращает объединение категорий, когда обнаруживает, что все оставшиеся категории статистически различаются между собой. Исчерпывающий CHAID исправляет это, продолжая объединять категории предиктора до тех пор, пока не останутся только две суперкатегории. Таким образом, он позволяет найти наилучшее расщепление для каждого предиктора и затем выбрать, какой предиктор нужно расщепить.

Исчерпывающий CHAID идентичен CHAID с точки зрения используемых зависимой переменной и предикторов, статистических тестов значимости взаимодействия и способа обработки пропущенных значений. Вместе с тем, поскольку объединение категорий осуществляется более тщательно, чем в методе CHAID, исчерпывающий CHAID требует большего времени вычислений. Надежность результатов исчерпывающего CHAID выше, чем у CHAID.

CRT (расшифровывается как Classification and Regression Tree – Деревья классификации и регрессии) был разработан в 1974–1984 годах профессорами статистики Лео Брейманом (Калифорнийский университет в Беркли), Джеромом Фридманом (Стэнфордский университет), Ричардом Олшеном (Калифорнийский университет в Беркли) и Чарльзом Стоуном (Стэнфордский университет).

Для построения дерева метод CRT использует принцип уменьшения неоднородности в узле. Расщепление узла происходит так, чтобы узел-потомок был более однородным, чем его узел-родитель. В абсолютно однородном узле все наблюдения имеют одно и то же значение целевой переменной (все объекты принадлежат к одной и той же категории целевой переменной). Такой узел еще называют «чистым».

Зависимая переменная может быть измерена в номинальной, порядковой и количественной шкалах. Предикторы могут быть измерены в номинальной, порядковой и количественной шкалах (подробнее о типах шкал читайте в разделе 2.3 «*Шкалы переменных*»). CRT позволяет только одномерные расщепления узлов. Каждый узел при разбиении может иметь лишь 2 потомков. Поэтому CRT имеет тенденцию выращивать высокие деревья с большим количеством уровней. Часто деревья CRT получаются слишком детализированными, имеют много узлов и ветвей, сложны для интерпретации, при этом усложнение дерева не приводит к повышению прогностической способности дерева. Для упрощения структуры дерева и повышения качества модели в методе CRT предусмотрена возможность отсечения ветвей (прунинг). Прунинг позволяет получить дерево «подходящего размера», избежать построения ветвистых, усложненных деревьев и при этом достичь лучшего качества модели.

Для обработки наблюдений, у которых пропущено значение в предикторе, используются суррогаты – другие предикторы, имеющие сильную корреляцию с исходной независимой переменной. Таким образом, разбиение, задаваемое суррогатом, будет наиболее близко к разбиению, задаваемому исходным предиктором, по которому имеются пропуски. Метод CRT требует большее время вычислений, по сравнению с другими методами.

QUEST (расшифровывается как *Quick, Unbiased, Efficient Statistical Tree – Быстрое, несмещенное, эффективное статистическое дерево*) был предложен в 1997 году профессорами статистики Вэй Ин Ло (Университет Висконсина-Мэдисона) и Ю Шан Ши (Национальный университет Чун Чен, Тайвань).

Метод QUEST строит дерево следующим образом: для отбора предикторов используются статистические тесты значимости взаимодействия между зависимой переменной и предиктором, а разбиение узлов задается путем выполнения квадратичного дискриминантного анализа с использованием отобранного предиктора. Зависимая переменная может быть измерена только в номинальной шкале. Предикторы могут быть измерены в номинальной, порядковой и количественной шкалах.

QUEST имеет схожие с CRT характеристики:

- позволяет только одномерные расщепления узлов;
- каждый узел при разбиении может иметь лишь 2 потомков;
- есть возможность отсечения ветвей (прунинг);
- для обработки наблюдений, у которых пропущено значение в предикторе, используются суррогаты – другие предикторы, имеющие сильную корреляцию с исходной независимой переменной.

Ниже на рис. 2.3 приводится таблица сходств и различий между четырьмя методами деревьев решений, предлагаемых процедурой **Деревья классификации**.

Характеристика метода	CHAID	Exhaustive CHAID	CRT	QUEST
Категориальная зависимая переменная	Да	Да	Да	Да, только номинальная
Категориальные предикторы	Да	Да	Да	Да
Количественная зависимая переменная	Да	Да	Да	Нет
Количественные предикторы	Да, преобразуются в порядковые	Да, преобразуются в порядковые	Да	Да
Тип разбиения	Множественный	Множественный	Бинарный	Бинарный
Цены ошибочной классификации (Построение дерева)	Нет	Нет	Да	Да
Статистические тесты (Отбор предикторов)	Да	Да	Нет	Да
Статистические тесты (Разбиение)	Да	Да	Нет	Нет
Время вычислений	Умеренное	Умеренное	Большое	Умеренное/Большое
Использование априорных вероятностей	Нет	Нет	Да	Да
Пропущенные значения в предикторах	Да, как категория	Да, как категория	Нет, для разбиения используется заменитель	Нет, для разбиения используется заменитель

Рис. 2.3 ❖ Четыре метода деревьев решений

2.3. Шкалы переменных

В зависимости от шкалы (уровня измерения) зависимой переменной и независимых переменных деревья решений применяют различные критерии для отбора предикторов и разбиения узлов. Поэтому важно задать правильную шкалу переменной. В IBM SPSS Statistics существуют три типа шкалы: количественная, порядковая, номинальная.

Количественная шкала содержит информацию о расстояниях между уровнями переменной, порядке уровней и количестве объектов в уровнях. Пример предиктора с количественной шкалой – переменная *Возраст*. Например, мы знаем, что расстояние между 25 и 30 в два раза меньше, чем расстояние между 30 и 40, 30-летний на 5 лет старше 25-летнего. Мы можем упорядочить уровни по нарастанию или убыванию интенсивности определенного признака (например, по увеличению возраста): после 25 следует 30, и 30-летний старше 25-летнего. Наконец, мы можем сказать, сколько в выборке человек с тем или иным уровнем (значением) возраста.

Порядковая шкала содержит информацию о порядке уровней и количестве объектов в уровнях. Пример предиктора с порядковой шкалой – переменная *Доход*, разбитая на уровни *низкий*, *средний*, *высокий*. Здесь уже нельзя сказать, что расстояние между уровнями *низкий* и *средний* больше или меньше в определенное количество раз расстояния между уровнями *средний* и *высокий*. Мы не можем утверждать, что человек со средним доходом на *n*-ное количество единиц богаче, чем человек с низким доходом. Однако можно упорядочить уровни по нарастанию или убыванию интенсивности определенного признака: сначала следует уровень *низкий*, затем уровень *средний*, и потом уровень *высокий*. Респонденты, относящиеся к уровню *средний*, обладают меньшим доходом, по сравнению с респондентами, относящимися к уровню *высокий*, то есть демонстрируют меньшую интенсивность признака. Также мы можем сказать, сколько в выборке человек с тем или иным уровнем дохода.

Номинальная шкала содержит только информацию о количестве объектов в уровнях. Пример предиктора с номинальной шкалой – переменная *Регион*, который имеет уровни *Алтайский край*, *Новосибирская область*, *Красноярский край*, *Кемеровская область*. Мы ничего не можем сказать о расстояниях между уровнями, о порядке уровней. Мы можем лишь судить о количестве респондентов, проживающих в каждом регионе.

Для изменения шкалы непосредственно в диалоговом окне **Дерево решений** щелкните правой кнопкой мыши по переменной, уровень измерения которой вы хотите изменить. В появившемся контекстном меню, изображенном на рис. 2.4, можно объявить переменную как номинальную, порядковую или количественную. Обратите внимание, что вы можете менять шкалу переменной только до ее перемещения в область **Зависимая переменная** или в область **Независимые переменные**.

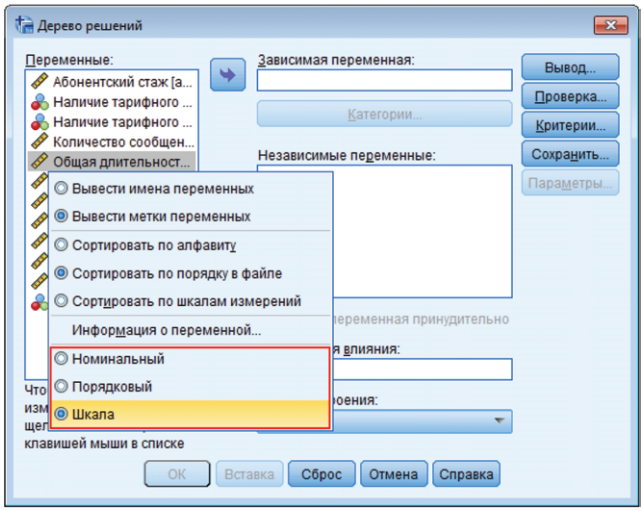

Рис. 2.4 ❖ Изменение шкалы переменной в диалоговом окне **Дерево решений**

Также обратите внимание, что пиктограммы, сопровождающие каждую переменную в диалоговом окне, показывают текущую шкалу этой переменной. Пиктограммы и соответствующие им шкалы показаны ниже на рис. 2.5.

Рис. 2.5 ❖ Три пиктограммы переменных (слева направо: количественная, номинальная, порядковая)

Кроме того, задать шкалу можно в Редакторе переменных (рис. 2.6). Для этого перейдите в Редактор переменных, щелкнув в левом нижнем углу Редактора данных вкладку **Представление Переменные**. Затем щелкните левой клавишей мыши по полю **Шкала** напротив интересующей переменной и выберите нужную шкалу.

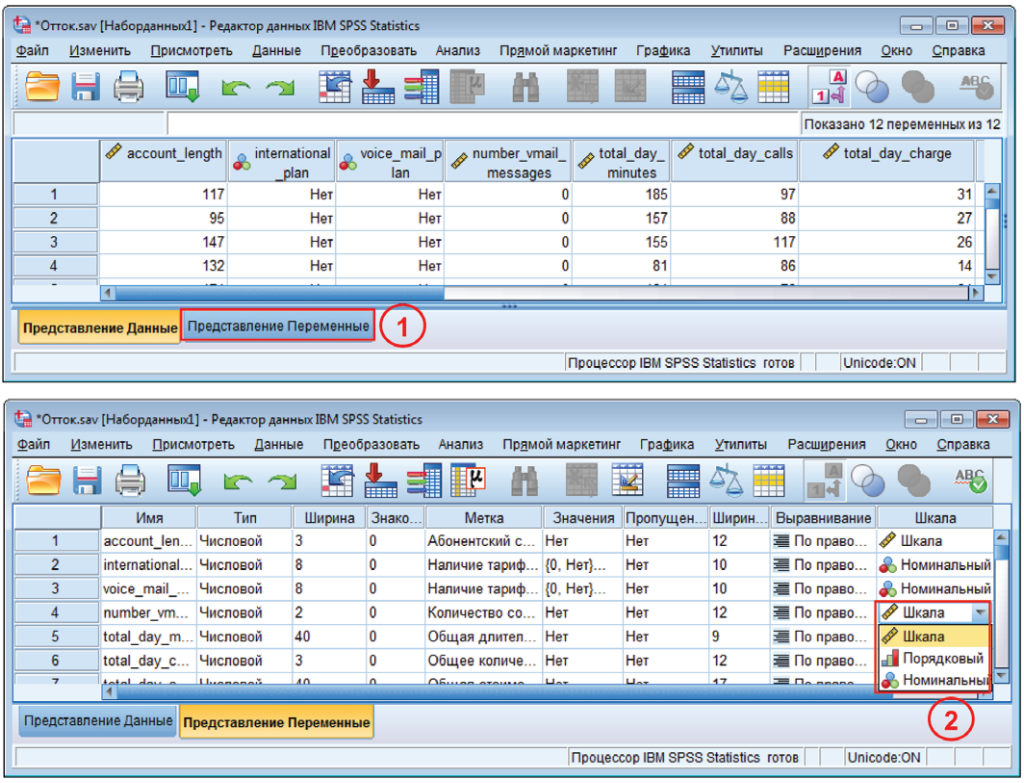

Рис. 2.6 ❖ Изменение шкалы переменной

2.4. Определение необходимого размера выборки

Формирование выборки в ходе разработки прогнозной модели обусловлено двумя задачами, а именно разбиением исторической выборки на обучающую и контрольную и распределением наблюдений в категориях зависимой переменной.

Предположим, мы хотим быть уверенным на 95%, что соотношение «плохих» и «хороших» заемщиков в выборке должно отражать генеральную популяцию заемщиков. Зная долю интересующего признака (например, долю «плохих») в популяции, задав z-статистику, определяемую в зависимости от уровня значимости, и ширину доверительного интервала (точность оценки), вы можете найти минимальный объем выборки по нижеприведенной формуле:

$$n = (z^1 p(1 - p))/d^2,$$

где z – z-статистика для уровня значимости (например, 1,96 для уровня значимости 0,05); p – доля признака в популяции (например, доля «плохих»); d – половина ширины доверительного интервала для доли признака (точность оценки доли признака).

Например, предположив наихудший сценарий 50/50 (задающий максимально возможный размер выборки) и точность оценки 5%, или 0,05, вычислим минимальный объем выборки:

$n = (1{,}962 \times 0{,}5 \times (1 - 0{,}5))/0{,}05^2 = 385.$

Однако нужно помнить, что эти методы дают минимальные размеры выборки. Необходимо придерживаться такого метода формирования выборки, который считается наиболее обоснованным до тех пор, пока размеры выборки позволяют убедиться в удовлетворительных статистических и практических результатах.

На практике обычно используют правило 20 EPV (Event Per Variable), сформулированное американским стастистиком Фрэнком Харреллом[1]. Оно связывает минимальный объем выборки с количеством наблюдений в миноритарной (наименьшей по размеру) категории зависимой переменной и количеством предикторов, поданным на вход модели.

Согласно этому правилу, необходимо взять количество наблюдений в исторической выборке, относящихся к миноритарной категории зависимой переменной (в кредитном скоринге это «плохие» заемщики). Это число наблюдений нужно разделить на количество заданных предикторов. На один предиктор должно приходиться не менее 20 наблюдений. Если это правило выполняется, то объем выборки достаточный.

Например, у нас есть выборка в 2000 наблюдений (1700 «хороших» и 300 «плохих») и 16 независимых переменных. Проверяя выполнение правила 20EPV, мы получаем 300/16 = 18,75. Наша выборка не обеспечивает достаточного количества наблюдений в миноритарной категории зависимой переменной.

Если планируется проверка модели с разбиением набора данных на обучающую и контрольную выборки (например, 70%:30%), необходимо руководствоваться еще более жестким правилом: взять число наблюдений в миноритарной категории зависимой переменной по контрольной выборке и разделить на количество заданных предикторов.

Например, имеется историческая выборка объемом 5000 наблюдений (4000 «хороших» и 1000 «плохих») и 10 независимых переменных. Теперь разбиваем историческую выборку случайным образом на обучающую и контрольную. В обучающую выборку попали 3500 наблюдений (2800 «хороших» и 700 «плохих»), а в контрольную выборку попали 1500 наблюдений (1200 «хороших» и «300 плохих»). Проверяя выполнение правила 20EPV, мы получаем 300/10 = 30. Историческую выборку данного объема можно использовать для моделирования.

2.5. Знакомство с методом CHAID

2.5.1. Описание алгоритма

Перед началом работы алгоритм CHAID преобразует все имеющиеся количественные предикторы в порядковые переменные, имеющие по умолчанию 10 приблизительно равных по объему категорий (число категорий можно устанавливать самостоятельно).

Затем алгоритм приступает к построению дерева, итеративно применяя к каждому узлу, начиная с корневого, процедуры объединения категорий, расщепления узла и проверки правил остановки.

---

[1] http://biostat.mc.vanderbilt.edu/wiki/Main/ManuscriptChecklist.

Объединение категорий

1. Для каждого предиктора с числом категорий больше двух[1] алгоритм ищет пару категорий (для порядковых переменных можно брать лишь две смежные категории, для номинальных переменных – любые две категории), меньше всего статистически различающихся по зависимой переменной. Для этого он выполняет статистические тесты. Выбор теста определяется типом шкалы зависимой переменной. Для номинальной зависимой переменной используется тест хи-квадрат Пирсона, или хи-квадрат отношения правдоподобия. Алгоритм строит двухвходовую таблицу сопряженности с категориями предиктора в качестве строк и категориями зависимой переменной в качестве столбцов. Он проверяет нулевую гипотезу о том, что категории предиктора не отличаются друг от друга с точки зрения распределения категорий зависимой переменной. Для количественной зависимой переменной используется F-тест. Алгоритм осуществляет дисперсионный анализ и проверяет нулевую гипотезу о том, что средние значения зависимой переменной для различных категорий предиктора не различаются между собой. Для порядковой зависимой переменной используется тест хи-квадрат отношения правдоподобия. Алгоритм подгоняет модель эффектов строк, где строки являются категориями предиктора, а столбцы – категориями зависимой переменной. В рамках каждого теста алгоритм вычисляет *p*-значение – вероятность того, что случайная величина с распределением тестовой статистики при нулевой гипотезе примет значение, не меньшее, чем фактическое значение тестовой статистики. Таким образом, задача алгоритма сводится к тому, чтобы для каждого предиктора найти пару категорий с наибольшим *p*-значением, поскольку именно такие категории будут меньше всего статистически различаться по зависимой переменной.

2. Найдя наибольшее *p*-значение для пары категорий, алгоритм сравнивает его с заданным уровнем значимости для объединения категорий.

Если *p*-значение:

- меньше или равно заданному уровню значимости для объединения категорий – алгоритм переходит к вычислению скорректированных *p*-значений для полученного набора категорий (шаг 3);
- больше уровня значимости для объединения категорий – эта пара объединяется в отдельную составную категорию, в результате формируется новый набор категорий предиктора и процесс начинается заново с поиска пары категорий с наибольшим *p*-значением.

ПРИМЕЧАНИЕ

Чтобы задать уровень значимости для объединения категорий, необходимо выполнить следующие действия:

- в главном диалоговом окне **Дерево решений** нажмите кнопку **Критерии**;
- в открывшемся диалоговом окне **Деревья решений: Критерии** откройте вкладку **CHAID**;
- в панели **Уровни значимости для** выберите для параметра **Объединения категорий** необходимое значение (значение по умолчанию 0,05).

Более подробно об этом читайте в разделе 3.1 «***Построение деревьев CHAID с измененными критериями***».

[1] Если предиктор имеет одну категорию, он исключается из анализа. Если предиктор имеет две категории, переходит к шагу 3.

(Опционно) Если новая составная категория состоит из трех и более исходных категорий, алгоритм находит внутри этой составной категории наилучшее бинарное расщепление, которое дает наименьшее *p*-значение. Алгоритм выполняет бинарное расщепление, если его *p*-значение не превышает уровня значимости для разбиения объединенных категорий.

ПРИМЕЧАНИЕ

Чтобы задать разбиение объединенных категорий, необходимо выполнить следующие действия:

- в главном диалоговом окне **Дерево решений** нажмите кнопку **Критерии**;
- в открывшемся диалоговом окне **Деревья решений: Критерии** откройте вкладку **CHAID**;
- отметьте параметр **Допускать разбиение объединенных категорий в узле** (по умолчанию не используется).

Более подробно об этом читайте в разделе 3.1 «***Построение деревьев CHAID с измененными критериями***».

3. Для сформированного набора категорий предиктора алгоритм вычисляет скорректированное *p*-значение как *p*-значение, умноженное на поправку Бонферонни. Поправка Бонферрони представляет собой число возможных способов, с помощью которых исходные категории предиктора могут быть объединены в итоговые категории.

ПРИМЕЧАНИЕ

Чтобы отменить применение поправки Бонферрони, необходимо выполнить следующие действия:

- в главном диалоговом окне **Дерево решений** нажмите кнопку **Критерии**;
- в открывшемся диалоговом окне **Деревья решений: Критерии** откройте вкладку **CHAID**;
- деактивируйте параметр **Корректировать уровни значимости с использованием поправки Бонферрони** (по умолчанию активирован).

Расщепление узла

После вычисления скорректированных *p*-значений для итоговых наборов категорий по всем предикторам алгоритм переходит к этапу расщепления узла.

1. На этапе расщепления алгоритм выбирает, какой предиктор обеспечит наилучшее разбиение узла. Для этого предиктор должен иметь наименьшее скорректированное *p*-значение (то есть является наиболее статистически значимым).

2. Найдя предиктор с наименьшим скорректированным *p*-значением, алгоритм сравнивает его с заданным уровнем значимости для расщепления.

Если *p*-значение:

- меньше или равно заданному уровню значимости для расщепления – алгоритм разбивает узел с использованием данного предиктора (категории предиктора становятся дочерними узлами);
- больше заданного уровня значимости для расщепления, то алгоритм не расщепляет узел и узел рассматривается как терминальный.