



БИБЛИОТЕКА  
ПРОГРАММИСТА

Анналин Ын, Кеннет Су



# ТЕОРЕТИЧЕСКИЙ МИНИМУМ ПО BIG DATA

ВСЁ, ЧТО НУЖНО ЗНАТЬ О БОЛЬШИХ ДАННЫХ



 ПИТЕР®

**Annalyn Ng & Kenneth Soo**

**Numsense!**  
**Data Science for the**  
**Layman**

No Math Added

Анналин Ын, Кеннет Су

# ТЕОРЕТИЧЕСКИЙ МИНИМУМ ПО BIG DATA

ВСЁ, ЧТО НУЖНО ЗНАТЬ О БОЛЬШИХ ДАННЫХ



Санкт-Петербург • Москва • Екатеринбург • Воронеж  
Нижний Новгород • Ростов-на-Дону  
Самара • Минск

2019

ББК 32.973.233-018  
УДК 004.62  
Ы11

## **Ын Анналин, Су Кеннет**

**Ы11** Теоретический минимум по Big Data. Всё, что нужно знать о больших данных. — СПб.: Питер, 2019. — 208 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-4461-1040-7

Сегодня Big Data — это большой бизнес.

Нашей жизнью управляет информация, и извлечение выгоды из нее становится центральным моментом в работе современных организаций. Независимо, кто вы — деловой человек, работающий с аналитикой, начинающий программист или раз-работчик, «Теоретический минимум по Big Data» позволит не утонуть в бушующем океане современных технологий и разобраться в основах новой и стремительно развивающейся отрасли обработки больших данных.

Хотите узнать о больших данных и механизмах работы с ними? Каждому алгоритму посвящена отдельная глава, в которой не только объясняются основные принципы работы, но и даются примеры использования в реальных задачах. Большое количество иллюстраций и простые комментарии позволят легко разобраться в самых сложных аспектах Big Data.

**16+** (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.973.233-018  
УДК 004.62

Права на издание получены по соглашению с Annalyn Ng и Kenneth Soo. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

ISBN 978-9811110689 англ.

Authorized translation from the English language edition, titled «Numsense! Data Science for the Layman: No Math Added» (ISBN 9789811110689) by Annalyn Ng & Kenneth Soo  
© 2017 to present

ISBN 978-5-4461-1040-7

© Перевод на русский язык ООО Издательство «Питер», 2019  
© Издание на русском языке, оформление ООО Издательство «Питер», 2019  
© Серия «Библиотека программиста», 2019  
© Перевод с английского языка Тимохин А. В., 2018

# Краткое содержание

Предисловие.....	12
Введение .....	16
Почему Data Science? .....	18
<b>Глава 1.</b> Об основах без лишних слов .....	21
<b>Глава 2.</b> Кластеризация методом k-средних.....	39
<b>Глава 3.</b> Метод главных компонент .....	51
<b>Глава 4.</b> Ассоциативные правила .....	65
<b>Глава 5.</b> Анализ социальных сетей .....	77
<b>Глава 6.</b> Регрессионный анализ.....	93
<b>Глава 7.</b> Метод k-ближайших соседей и обнаружение аномалий.....	107
<b>Глава 8.</b> Метод опорных векторов .....	117
<b>Глава 9.</b> Дерево решений.....	127
<b>Глава 10.</b> Случайные леса.....	137
<b>Глава 11.</b> Нейронные сети .....	149
<b>Глава 12.</b> A/B-тестирование и многорукие бандиты.....	167
Приложения .....	179
Глоссарий.....	188
Литература и ссылки на источники.....	199
Об авторах .....	204

# Оглавление

<b>Предисловие .....</b>	<b>12</b>
От издательства .....	15
<b>Введение .....</b>	<b>16</b>
<b>Почему Data Science? .....</b>	<b>18</b>
<b>Глава 1. Об основах без лишних слов.....</b>	<b>21</b>
1.1. Подготовка данных.....	22
Формат данных.....	23
Типы переменных .....	24
Выбор переменных .....	25
Конструирование признаков .....	25
Неполные данные .....	26
1.2. Выбор алгоритма .....	27
Обучение без учителя.....	28
Обучение с учителем .....	29
Обучение с подкреплением .....	30
Другие факторы.....	31

---

1.3. Настройка параметров .....	31
1.4. Оценка результатов .....	33
Метрики классификации .....	34
Метрика регрессии .....	35
Валидация.....	36
1.5. Краткие итоги .....	38
<b>Глава 2. Кластеризация методом k-средних .....</b>	<b>39</b>
2.1. Поиск кластеров клиентов.....	40
2.2. Пример: профили кинозрителей .....	41
2.3. Определение кластеров.....	42
Сколько кластеров существует? .....	44
Что включают кластеры? .....	46
2.4. Ограничения .....	48
2.5. Краткие итоги.....	49
<b>Глава 3. Метод главных компонент .....</b>	<b>51</b>
3.1. Изучение пищевой ценности .....	52
3.2. Главные компоненты.....	53
3.3. Пример: анализ пищевых групп.....	56
3.4. Ограничения .....	61
3.5. Краткие итоги.....	64
<b>Глава 4. Ассоциативные правила .....</b>	<b>65</b>
4.1. Поиск покупательских шаблонов.....	66
4.2. Поддержка, достоверность и лифт .....	67

4.3. Пример: ведение продуктовых продаж .....	69
4.4. Принцип Apriori .....	72
Поиск товарных наборов с высокой поддержкой.....	73
Поиск товарных правил с высокой достоверностью или лифтом.....	74
4.5. Ограничения .....	75
4.6. Краткие итоги.....	76

## **Глава 5. Анализ социальных сетей ..... 77**

5.1. Составление схемы отношений.....	78
5.2. Пример: геополитика в торговле оружием.....	80
5.3. Лувенский метод .....	84
5.4. Алгоритм PageRank .....	86
5.5. Ограничения .....	90
5.6. Краткие итоги .....	91

## **Глава 6. Регрессионный анализ ..... 93**

6.1. Выведение линии тренда.....	94
6.2. Пример: предсказание цен на дома .....	95
6.3. Градиентный спуск .....	98
6.4. Коэффициенты регрессии.....	101
6.5. Коэффициенты корреляции.....	102
6.6. Ограничения .....	104
6.7. Краткие итоги.....	106

## **Глава 7. Метод k-ближайших соседей и обнаружение аномалий ..... 107**

7.1. Пищевая экспертиза.....	108
------------------------------	-----



---

7.2. Яблоко от яблони недалеко падает .....	109
7.3. Пример: истинные различия в вине .....	111
7.4. Обнаружение аномалий.....	113
7.5. Ограничения .....	114
7.6. Краткие итоги.....	115
<b>Глава 8. Метод опорных векторов.....</b>	<b>117</b>
8.1 «Нет» или «о, нет!»?.....	118
8.2. Пример: обнаружение сердечно-сосудистых заболеваний .....	118
8.3. Построение оптимальной границы.....	120
8.4. Ограничения .....	124
8.5. Краткие итоги.....	125
<b>Глава 9. Дерево решений.....</b>	<b>127</b>
9.1. Прогноз выживания в катастрофе .....	128
9.2. Пример: спасение с тонущего «Титаника» .....	128
9.3. Создание дерева решений .....	131
9.4. Ограничения .....	133
9.5. Краткие итоги.....	135
<b>Глава 10. Случайные леса.....</b>	<b>137</b>
10.1. Мудрость толпы .....	138
10.2. Пример: предсказание криминальной активности.....	139
10.3. Ансамбли .....	144
10.4. Бэггинг.....	145

10.5. Ограничения.....	147
10.6. Краткие итоги .....	148

**Глава 11. Нейронные сети ..... 149**

11.1. Создание мозга .....	150
11.2. Пример: распознавание рукописных цифр.....	152
11.3. Компоненты нейронной сети.....	156
11.4. Правила активации .....	159
11.5. Ограничения.....	161
11.6. Краткие итоги .....	165

## Глава 12. А/В-тестирование и многорукие бандиты ..... 167

12.1. Основы А/В-тестирования.....	168
12.2. Ограничения А/В-тестирования.....	169
12.3. Стратегия снижения эпсилона.....	169
12.4. Пример: многорукие бандиты .....	171
12.5. Забавный факт: ставка на победителя.....	174
12.6. Ограничения стратегии снижения эпсилона .....	175
12.7. Краткие итоги .....	176

**Приложения ..... 179**

Приложение А. Обзор алгоритмов обучения без учителя .....	180
Приложение В. Обзор алгоритмов обучения с учителем .....	181
Приложение С. Список параметров настройки .....	182

Приложение D. Другие метрики оценки .....	183
Метрики классификации .....	183
Метрики регрессии.....	186
<b>Глоссарий .....</b>	<b>188</b>
<b>Литература и ссылки на источники .....</b>	<b>199</b>
Источники на английском языке .....	199
Литература на русском языке.....	202
<b>Об авторах.....</b>	<b>204</b>

# Предисловие

Сегодня Big Data (большие данные) — это большой бизнес. Информация все больше управляет нашей жизнью, и получение выгод из нее стало центральным моментом в работе почти любой организации. А методы распознавания образов и прогнозирования создают для бизнеса новые измерения. Например, рекомендательные системы выгодны одновременно покупателям и продавцам, так как информируют первых о продукции, которая могла бы их заинтересовать, а вторым позволяют набивать мощну.

Но Big Data — это лишь часть головоломки. Data Science — это многогранная дисциплина, которая охватывает машинное обучение, статистику и связанные с нею разделы математики и при этом дает нам возможность для анализа данных и извлечения из них пользы. Стоит отметить, что машинное обучение занимает в этом описании ведущую позицию, будучи основным двигателем распознавания образов и технологий прогнозирования. Вкупе с данными алгоритмы машинного обучения, направляя науку о них, ведут к бесценным озарениям и новым способам задействования информации, которая уже в нашем распоряжении.

Чтобы по достоинству оценить то, как Data Science двигает сегодняшнюю информационную революцию, непосвященный должен лучше понимать эту сферу деятельности. Несмотря на высокий спрос на грамотность в вопросах данных, опасения некоторых людей в том, что им не хватит навыков для понимания, стали поводом избегать этой области.

Но тут появляется *Теоретический минимум по Big Data*.

Стоит познакомиться с работой Анналин Ын и Кеннета Су, чтобы убедиться, что книга своему названию вполне соответствует. Это *действительно* Data Science для неспециалиста, поэтому математика, местами сложная, которая описывается на отвлеченном уровне, намеренно не освещена подробно. Но не поймите неправильно: это не означает, что содержимое книги размыто. Информация в ней существенная, а вот лаконичность и емкость пошли только на пользу.

Что же хорошего при таком подходе, спросите вы. Вообще, много чего! Я бы утверждал, что для неспециалиста предпочтителен именно такой подход. Подумайте о неспециалисте, которому интересно устройство машины. Абстрактный обзор составных частей автомобиля куда доступнее технического пособия по физике сгорания. То же справедливо и по отношению к Big Data: если вы хотите разобраться в этом, проще начать с общих представлений, не погружаясь сразу в формулы.

Уже в начале книги можно на нескольких страницах познакомиться с фундаментальными понятиями Big Data.

Это гарантирует, что каждый может начать чтение книги, уже зная основы. Важные принципы, например часто опускаемый во вводных материалах выбор алгоритма, также приводятся сразу. Это пробуждает в читателе желание скорее освоить эти области и закладывает фундамент для будущих знаний.

Есть немало концепций, которые Анналин и Кеннет могли бы считать достойными включения в книгу, и существует далеко не один способ их представить. Их подход, при котором они сосредоточились на важнейших для Data Science алгоритмах машинного обучения и описали несколько практических случаев, оказался отличным решением. Но не обделены вниманием и проверенные и испытанные алгоритмы, такие как метод  $k$ -ближайших соседей, дерево принятия решений, метод  $k$ -средних. Хорошо объясняются и более современные алгоритмы классификации и ансамблирования, такие как случайные леса и метод опорных векторов, который нередко отпугивает сложной математикой. Рассмотрены и нейронные сети — движущая сила сегодняшнего помешательства на глубоком обучении.

Другое достоинство книги — описание алгоритмов вместе с интуитивно-понятными примерами использования, будь то объяснение алгоритма случайных лесов в контексте прогнозирования преступлений или метода классификации в применении к кинозрителям. Выбранные примеры обеспечивают ясность и практическое понимание. В то же время избавление от любого намека на высшую математику сохраняет интерес и мотивацию для того, что можно назвать вылазкой читателя в мир Data Science.

Я настоятельно рекомендую *Теоретический минимум по Big Data* новичкам в качестве отправной точки для изучения Data Science и ее алгоритмов. Мне трудно было бы назвать сопоставимый по уровню материал. С этой книгой математика вам больше не мешает оставаться в неведении.

Мэтью Майо,  
дата-сайентист и редактор сайта KDnuggets  
*@mattmayo13*

## От издательства

Мы прекрасно понимаем, что некоторые иллюстрации для лучшего восприятия нужно смотреть в цветном варианте. Мы снабдили их QR-кодами, перейдя по которым, вы можете ознакомиться с цветной версией рисунка.

Ваши замечания, предложения, вопросы отправляйте по адресу [comp@piter.com](mailto:comp@piter.com) (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства [www.piter.com](http://www.piter.com) вы найдете подробную информацию о наших книгах.

# Введение

Эту книгу написали для вас два энтузиаста Data Science, Анналин Ын (Кембриджский университет) и Кеннет Су (Стэнфордский университет).

Мы обратили внимание на то, что, несмотря на растущую роль Data Science в рабочих решениях, многие мало знают об этой области. Поэтому мы составили из руководств книгу, прочитать которую сможет каждый, будь то профессиональный предприниматель, абитуриент, да и просто любой, кому это интересно.

Каждое руководство посвящено важным предпосылкам и функциям одного из методов Data Science и не предполагает математики или научного жаргона. Мы проиллюстрировали эти методы данными и примерами из реального мира.

Мы не сумели бы написать эту книгу одни.

Благодарим нашего редактора и хорошего друга Соню Чан (Sonya Chan) за искусное соединение наших стилей письма и ровность повествования.

Мы признательны нашему талантливому дизайнеру Доре Тань (Dora Tan) за макет книги и обложку.



Благодарим наших друзей Денниса Чу (Dennis Chew), Марка Хо (Mark Ho) и Мишель Фу (Michelle Poh) за бесценные советы о том, как облегчить понимание материала.

Выражаем признательность профессору Лонгу Нгуену (Long Nguyen, Мичиганский университет, г. Анн-Арбор), профессору Перси Ляну (Percy Liang, Стэнфордский университет) и профессору Михалу Косински (Michal Kosinski, Стэнфордский университет) за их терпение во время нашего обучения и за то, что поделились своим экспертным мнением.

Наконец, благодарим друг друга за то, что хотя и ссорились, как это заведено у друзей, но не останавливались, пока не завершили начатое дело.

# Почему Data Science?

Представьте себе, что вы молодой врач. К вам пришел пациент, который жалуется на одышку, боли в груди и периодическую изжогу. Вы убедились, что его давление и показания сердечного ритма в норме и ничего подозрительного у него прежде не замечалось.

Вы также отметили его полноту. Поскольку такие симптомы типичны для людей с избыточным весом, вы заверили его, что все в порядке, и посоветовали найти время для упражнений.

Слишком часто это приводит к неверному диагнозу при сердечно-сосудистых заболеваниях. У пациентов в этом состоянии проявляются симптомы, которые схожи с симптомами ожирения, и врачи прекращают диагностику, которая могла бы обнаружить более серьезное заболевание.

Мы — люди, и наши суждения обусловлены ограниченным субъективным опытом и несовершенными знаниями. Это ухудшает процесс принятия решения и, как в случае с неопытным врачом, удерживает от дальнейших проверок, которые могли бы привести к более точным выводам.

Здесь может помочь Data Science.

Не ограничиваясь суждением одного индивида, методы Data Science позволяют задействовать для принятия лучшего решения информацию из разных источников. Например, мы могли бы свериться со статистикой по пациентам с такими симптомами и обнаружить диагнозы, о которых не подумали.

С современным вычислением и передовыми алгоритмами мы можем:

- обнаружить скрытые тенденции в больших наборах данных;
- воспользоваться этими тенденциями для прогнозирования;
- вычислить вероятность любого возможного исхода;
- получить точные результаты быстро.

Эта книга написана доступным языком (никаких формул!) для легкого введения в Data Science и алгоритмы. Чтобы облегчить понимание ключевых идей, мы будем придерживаться интуитивно-понятных объяснений и иллюстраций.

Каждый алгоритм представлен отдельной главой с реальным случаем для объяснения работы этого алгоритма. Данные этих примеров доступны онлайн, а в разделе ссылок приведены источники.

Для повторения изученного обратитесь к кратким итогам в конце каждой главы. В конце книги вы также найдете

удобные таблицы для сравнения алгоритмов и глоссарий основных терминов.

Мы надеемся дать вам практическое понимание Data Science, чтобы и вы вооружились ею для лучших решений.

Время начинать.

# 1

**Об основах  
без лишних слов**

Чтобы разобраться с тем, как работают алгоритмы Data Science, мы должны начать с основ. Эта глава книги самая длинная. Она вдвое больше остальных, которые останавливаются на каждом алгоритме подробнее. Тем не менее здесь вы найдете обстоятельный обзор основных шагов почти любого исследования с применением Data Science. Эти основные приемы помогут определиться с выбором алгоритмов в соответствии с контекстом и условиями исследования.

Такое исследование включает четыре ключевых шага. Сначала обрабатываются и подготавливаются данные. Потом составляется краткий перечень соответствующих исследованию алгоритмов. Затем для улучшения результатов настраиваются параметры этих алгоритмов. И наконец, строятся модели для выбора лучшей из них.

## **1.1. Подготовка данных**

В Data Science главную роль играют сами данные. Если качество данных низкое, то результаты даже самого изощренного анализа окажутся не ахти какими. В этом разделе мы рассмотрим типичный формат данных, использу-

емый для анализа, и методы их обработки для улучшения результатов.

### Формат данных

Обычно для анализа данных используют табличное представление (табл. 1). Каждая строка представляет собой *элемент данных* с описанием отдельного наблюдения, а каждый столбец несет *переменную* для его описания. Переменные также называются *атрибутами*, *признаками* или *размерностями*.

**Таблица 1.** Вымышленный набор данных о продуктовых покупках животных в магазине. Строки — это покупки, а столбцы — информация о них

← Переменные →						
↑ Элементы данных ↓	ID транзакции	Покупатель	Даты	Куплено фруктов	Куплена рыба	Потрачено
	1	Пингвин	1 янв.	1	да	5,30 \$
	2	Медведь	1 янв.	4	да	9,70 \$
	3	Кролик	1 янв.	6	нет	6,50 \$
	4	Лошадь	2 янв.	6	нет	5,50 \$
	5	Пингвин	2 янв.	2	да	6,00 \$
	6	Жираф	3 янв.	5	нет	4,80 \$
	7	Кролик	3 янв.	8	нет	7,60 \$
	8	Кот	3 янв.	?	да	7,40 \$

В зависимости от цели можно изменить представленный в строках тип наблюдений. Например, выборка в табл. 1

позволяет изучать закономерности, рассматривая покупки.

Но если вместо этого мы хотим исследовать закономерности покупок в зависимости от дня, то нам нужно представить в строках общий итог. Для всестороннего анализа имеет смысл также добавить новые переменные, такие как погода (табл. 2).

**Таблица 2.** Переформатированный набор данных о покупках за день с дополнительными переменными

Переменные				
Дата	Выручка	Число покупателей	Погода	Выходные
1 янв.	21,50 \$	3	солнечно	да
2 янв.	11,50 \$	2	дождливо	нет
3 янв.	19,80 \$	3	солнечно	нет

## Типы переменных

Есть четыре главных типа переменных. Чтобы убедиться, что к ним применимы выбранные алгоритмы, важно понимать разницу.

- **Бинарная.** Это простейший тип переменных только с двумя вариантами значения. В табл. 1 бинарная переменная показывает, брал ли покупатель рыбу.
- **Категориальная.** Если вариантов больше двух, информация может быть представлена категориальной переменной. В табл. 1 категориальная переменная описывает вид покупателя.



- **Целочисленная.** Такой тип используется, когда информация может быть представлена целым числом. В табл. 1 целое число выражает количество купленных каждым покупателем фруктов.
- **Непрерывная** (количественная). Это самая подробная переменная. Она содержит числа со знаками после запятой. В табл. 1 такие переменные показывают количество потраченных покупателем денег.

## **Выбор переменных**

Хотя в нашем первоначальном наборе данных может быть много разных переменных, применение в алгоритме слишком большого их числа ведет к замедлению вычислений или к ошибочным предсказаниям из-за информационного шума. Поэтому нам надо остановиться на коротком списке важнейших переменных.

Выбор переменных часто делается методом проб и ошибок. Их имеет смысл добавлять и убирать, учитывая промежуточные результаты. Для начала мы можем использовать простые графики для выявления корреляций (см. раздел 6.5) между переменными, отбирая самые многообещающие для дальнейшего анализа.

## **Конструирование признаков**

Тем не менее иногда хорошие переменные нужно сконструировать. Например, если мы хотим предсказать, кто из покупателей в табл. 1 не будет брать рыбу, то можем посмотреть на переменную их вида, заключив, что кро-

лики, лошади и жирафы рыбу не покупают. А если мы сгруппируем виды покупателей в более широкие категории — травоядных, хищников и всеядных, — то получим более универсальный вывод: травоядные рыбу не берут.

Вместо переформатирования одной переменной мы можем скомбинировать их методом, называемым *уменьшением размерности* (dimension reduction), который будет рассмотрен в главе 3. Уменьшение размерности может использоваться для извлечения самой полезной информации и ее выражения в небольшом наборе переменных для дальнейшего анализа.

## Неполные данные

Мы не всегда располагаем полными данными. Например, в табл. 1 количество фруктов в последней покупке неизвестно. Неполные данные мешают анализу и при любой возможности с ними нужно разобраться одним из следующих способов:

- **Приближение.** Если пропущено значение бинарного или категориального типа, его можно заменить самым типичным значением (модой) переменной. А для целочисленных или непрерывных переменных используется медиана. Применение этого метода к табл. 1 позволит нам предположить, что кот приобрел 5 фруктов, поскольку, согласно остальным семи записям, именно таково среднее число покупаемых фруктов.
- **Вычисление.** Пропущенные значения также могут быть вычислены с применением более продвинутых

алгоритмов *обучения с учителем* (будут рассмотрены в следующем разделе). Хотя такие вычисления требуют времени, они обычно приводят к более точным оценкам неполных значений. Причина в том, что вместо приближения к самому распространенному значению они оценивают значение по сходным записям. В табл. 1 мы видим, что если покупатели берут рыбу, они склонны приобретать меньше фруктов, а это значит, что кот должен был купить 3–4 фрукта.

- **Удаление.** В качестве последнего средства строки с неполными значениями могут быть удалены. Тем не менее этого обычно избегают, чтобы не уменьшать объем данных, доступных для анализа. Более того, исключение элементов данных может привести к искаженным результатам в отношении отдельных групп. Например, коты могут менее охотно, чем другие, раскрывать информацию о количестве приобретаемых фруктов. Если мы удалим такие покупки, коты будут недостаточно представлены в итоговой выборке.

После того как набор данных обработан, пора заняться его анализом.

## 1.2. Выбор алгоритма

В этой книге мы рассмотрим более десяти алгоритмов, используемых для анализа данных. Выбор алгоритма зависит от задачи, которую мы хотим решить. Существуют три основных класса. В табл. 3 приведены алгоритмы, которые будут рассмотрены в этой книге в соответствии с ними.

Таблица 3. Алгоритмы и их категории

	Алгоритмы
Обучение без учителя	Метод $k$ -средних Метод главных компонент Ассоциативные правила Анализ социальных сетей
Обучение с учителем	Регрессионный анализ Метод $k$ -ближайших соседей Метод опорных векторов Дерево решений Случайные леса Нейросети
Обучение с подкреплением	Многорукие бандиты

## Обучение без учителя

Задача: *найти закономерности в наших данных.*

Когда требуется найти скрытые закономерности в нашем наборе данных, мы можем воспользоваться алгоритмами *обучения без учителя*. Так называют алгоритмы, используемые тогда, когда мы не знаем, какие закономерности искать, и предоставляем их поиск самим алгоритмам.

В табл. 1 такая модель может использоваться либо для изучения товаров, часто покупаемых вместе (с использованием ассоциативных правил, глава 4), либо для груп-

пировки покупателей на основе их приобретений (объясняется в главе 2).

Результаты модели, построенной при обучении без учителя, мы можем подтвердить косвенным образом, если группы соответствуют уже известным категориям (то есть травоядным или хищникам).

## Обучение с учителем

*Задача: использовать для прогнозирования заданные шаблоны.*

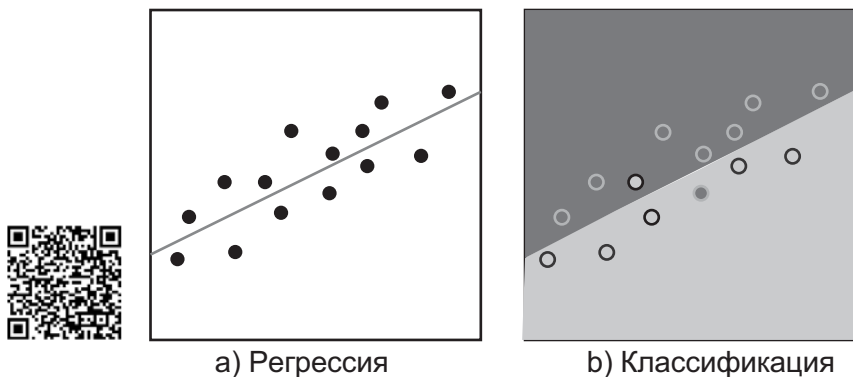
Когда нам требуется прогноз, могут использоваться алгоритмы обучения с учителем. Так называются алгоритмы, предсказания которых основаны на уже существующих шаблонах.

В табл. 1 такая модель может научиться предугадывать количество приобретаемых фруктов (предсказание), исходя из вида покупателя и того, покупает ли он рыбу (*предикторные переменные*).

Мы можем явно проверить точность модели, введя данные о виде покупателя и его склонности брать рыбу, а затем выяснив, насколько предсказание близко к реальному количеству фруктов.

Когда мы предсказываем целые или непрерывные числа, такие как количество фруктов, мы решаем проблему *регрессии* (рис. 1, а). А когда мы предсказываем бинарное или категориальное значение, например, пойдет ли дождь, мы занимаемся проблемой *классификации* (рис. 1, б). Тем

не менее многие алгоритмы классификации способны также выдавать прогноз в виде непрерывного значения, как в высокоточных утверждениях типа «*вероятность дождя 75 %*».



**Рис. 1.** Регрессия предполагает выведение линии тренда, а классификация — разделение элементов данных на группы. Обратите внимание, что ошибки ожидаемы в обеих задачах. При регрессии элементы данных способны отклоняться от линии тренда, в то время как при классификации могут попадать в ошибочные категории

## Обучение с подкреплением

*Задача: использовать закономерности в моих данных, постоянно улучшая прогнозирование по мере появления новых результатов.*

В отличие от обучения с учителем и без, где модели проходят обучение и после применяются без дальнейших изменений, модель *обучения с подкреплением* постоянно развивается, используя результаты обратной связи.

Перейдем от табл. 1 к примеру из реальной жизни. Представьте, что мы сравниваем эффективность двух онлайн-реклам. Изначально мы можем показывать обе с равной частотой, подсчитывая количество людей, кликнувших на каждой из них. Такая модель будет получать эти числа в качестве обратной связи по популярности рекламы, используя их для того, чтобы увеличить долю показа более популярной рекламы. Путем такого циклического процесса модель со временем научится показывать только лучшую рекламу.

## **Другие факторы**

Помимо своей основной задачи алгоритмы отличаются также в других аспектах, таких как их способность анализировать различные типы данных, а также форматом выводимых результатов. Эти моменты раскрыты в дальнейших главах, посвященных каждому алгоритму, а также приведены в сводных таблицах приложения А (обучение без учителя) и приложения В (обучение с учителем).

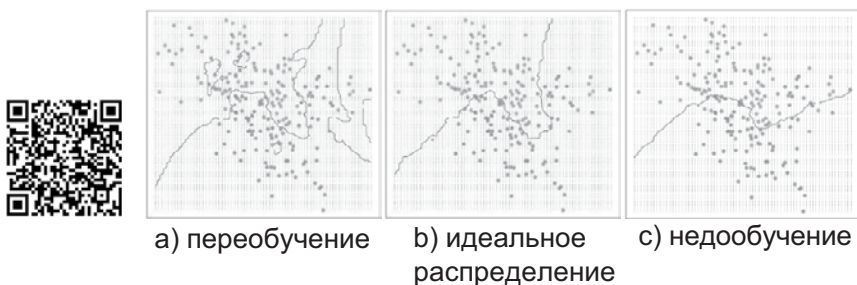
## **1.3. Настройка параметров**

Многочисленные алгоритмы, доступные в Data Science, естественно приводят к огромному числу потенциальных моделей, которые мы можем построить. Но даже один такой алгоритм способен генерировать различные результаты в зависимости от настройки его параметров.

Параметры — это тонкая регулировка алгоритма, похожая на настройку радиоприемника на нужную волну.

У разных алгоритмов свои параметры настройки. Общие параметры алгоритмов можно найти в приложении С.

Излишне говорить, что точность модели проигрывает, если параметры настроены несообразно. Посмотрите на рис. 2, чтобы увидеть, как алгоритм классификации может сгенерировать различные границы для разделения оранжевых и голубых точек.



**Рис. 2.** Сравнение результатов предсказания для одного и того же алгоритма с разными параметрами

На рис. 2, а алгоритм слишком чувствителен и принимает случайные отклонения данных за закономерности. Эта проблема известна как *переобучение* (overfitting). Такая модель точна для прогнозирования по уже имеющимся данным, но меньше подходит для будущей информации.

На рис. 2, в алгоритм, наоборот, слишком нечувствителен и основные закономерности упустил. Эта проблема известна как *недообучение* (underfitting). Такая модель способна пренебрегать важными тенденциями и дает



менее точные предсказания как для текущих, так и для будущих данных.

Но когда параметры настроены хорошо, как на рис. 2, б, алгоритм достигает равновесия, определяя главные тенденции, сбрасывая со счетов мелкие отклонения и предлагая хорошую прогностическую модель.

Чаше всего постоянной задачей становится переобучение. В попытках свести к минимуму ошибки прогнозирования мы можем поддаться искушению увеличить сложность модели. В конечном счете это приводит к результатам, похожим на показанные на рис. 2, а — границы проведены тонко, но избыточно.

Одним из способов держать под контролем сложность модели является введение штрафного параметра в процессе *регуляризации*. Этот новый параметр *штрафует* модель за сложность, искусственно увеличивая погрешность и этим побуждая алгоритм находить оптимальное соотношение точности со сложностью. Тем самым сохраняя простоту модели, мы можем обеспечить ее масштабируемость.

## 1.4. Оценка результатов

После того как модель построена, ее требуется оценить. Для сравнения моделей по степени точности предсказаний используются метрики оценки. Эти метрики определяют типы прогностических ошибок и штрафуют за них по-разному.

Рассмотрим три оценочные метрики, используемые чаще всего. В зависимости от целей нашего исследования, для того чтобы избежать ошибок специфического типа, могут быть разработаны даже новые метрики. В связи с этим перечень приводимых в этой книге оценочных метрик ни в коем случае нельзя считать исчерпывающим. В приложении D рассмотрены другие примеры метрик.

## Метрики классификации

**Процент верных прогнозов.** Простейшая мера точности прогнозирования — это доля достоверно правильных предсказаний. Вернемся к примеру с гастрономическими покупками из табл. 1. Мы можем выразить результаты задачи по предсказанию покупки рыбы в таком утверждении: *Наша модель с точностью 90 % предсказывает, будет ли покупатель брать рыбу*. Хотя эта метрика не так сложна для понимания, она не дает представления о том, где именно происходят ошибки прогнозирования.

**Таблица 4.** Матрица неточностей показывает точность предсказаний о покупке рыбы

		Прогноз	
		Купят	Не купят
Факт	Купили	1 (TP)	5 (FN)
	Не купили	5 (FP)	89 (TN)

**Матрица неточностей.** Матрица неточностей (confusion matrix) дает представление о том, где наша модель прогнозирования преуспела и где она потерпела неудачу.

Посмотрите на табл. 4. Хотя общая точность модели составляет 90 %, она гораздо лучше предсказывает не покупки, чем покупки. Мы также видим, что число прогностических ошибок равномерно (по 5) распределилось между *ложноположительными* (FP, false positives) и *ложноотрицательными* (FN, false negatives).

Разновидности прогностических ошибок могут иметь решающее значение. Ложноотрицательный результат в предсказании землетрясения (то есть землетрясения не ожидалось, но оно произошло) обойдется куда дороже, чем ложноположительный (землетрясение ожидалось, но не случилось).

## Метрика регрессии

**Корень из среднеквадратичной ошибки (Root Mean Squared Error, RMSE).** Поскольку при регрессии используются непрерывные числовые значения, то ошибки обычно измеряют количественно, как разницу между предсказанными и реальными значениями, распределяя штрафы и исходя из величины ошибки. *Корень из среднеквадратичной ошибки* — это популярная метрика регрессии, особенно полезная в случаях, когда мы хотим избежать крупных ошибок: каждая из них возводится в квадрат, что усиливает значимость такой ошибки. Это

делает метрику крайне чувствительной к резко отклоняющимся значениям, за которые она штрафует модель.

## Валидация

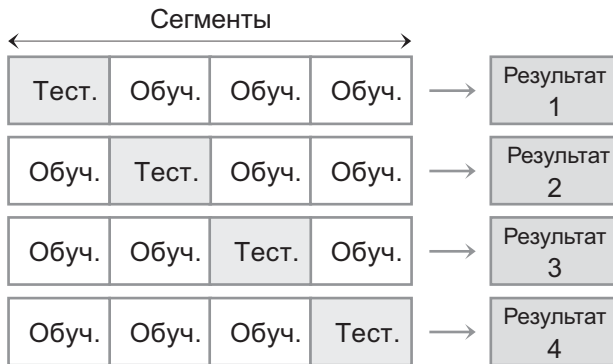
Метрики не дают полной картины эффективности модели. Из-за *переобучения* (см. раздел 1.3) модели, хорошо себя показавшие на уже имеющихся данных, могут не справиться с новыми. Чтобы этого избежать, мы всегда должны подвергать модели оценке, используя надлежащую процедуру валидации.

*Валидация* (validation) — это оценка того, насколько хорошо модель предсказывает новые данные. Тем не менее вместо ожидания новых данных для проверки модели мы можем разбить наш текущий набор данных на два сегмента. Первый выступит в роли нашего *обучающего набора данных* (training dataset), а второй послужит заменой для новой информации в качестве *тестового набора данных* (test dataset) для оценки точности прогностической модели. Лучшей моделью признается та, которая дает самые точные предсказания на тестовом наборе. Чтобы процесс валидации был эффективен, мы должны выбирать элементы для обучающего и тестового набора данных случайно и беспристрастно.

Однако если изначальный набор данных мал, мы не можем позволить себе роскошь отложить их часть для формирования тестового набора, поскольку тогда пришлось бы пожертвовать точностью, которая снижается от сокращения доступного объема данных.

По этой причине, вместо использования двух различных наборов данных для испытания одного набора проверкой другим, мы можем обойтись изначальным набором, устроив перекрестную проверку — кросс-валидацию.

*Кросс-валидация* (cross-validation) позволяет полностью задействовать данные путем разделения их набора на несколько сегментов для поочередной проверки модели. За одну итерацию все сегменты, кроме одного, используются для обучения модели, которая сама проверяется на последнем сегменте. Этот процесс повторяется до тех пор, пока каждый сегмент не отработает в роли тестового (рис. 3).



**Рис. 3.** Кросс-валидация набора данных. Набор данных разделен на четыре сегмента, а итоговая точность прогнозирования — это среднее значение четырех результатов

Поскольку для предсказаний на каждой итерации использовались разные сегменты, их прогнозы могут различаться. Приняв во внимание эту вариативность, мы можем дать

более здравую оценку действительным прогностическим способностям модели. А в качестве итоговой оценки точности модели принимают среднее значение за все итерации.

Если результаты кросс-валидации показывают, что прогностическая точность нашей модели невысока, мы можем вернуться к настройке параметров или обработать данные иначе.

## **1.5. Краткие итоги**

Исследование в рамках Data Science предполагает четыре ключевых шага:

1. Подготовка данных.
2. Выбор алгоритмов для моделирования этих данных.
3. Настройка алгоритмов для оптимизации моделей.
4. Оценка моделей, основанная на их точности.

# 2

## **Кластеризация методом $k$ -средних**

## 2.1. Поиск кластеров клиентов

Давайте поговорим о кинопредпочтениях. Возьмем, к примеру, человека, которому нравится *«50 первых поцелуев»*. Скорее всего, ему придутся по вкусу и другие чикфлики<sup>1</sup> типа *«27 свадеб»*. Так и работает этот метод: определив общие предпочтения или характеристики, можно разделить клиентов на группы, которые затем можно использовать для таргетированной рекламы.

Однако определение таких групп — хитроумная задача. Мы изначально можем не знать, как следует группировать клиентов и сколько групп существует.

Ответить на эти вопросы нам поможет *кластеризация методом  $k$ -средних* (k-means clustering). Этот метод используется для разделения клиентов или товаров на особые кластеры, где  $k$  относится к числу этих найденных кластеров.

---

<sup>1</sup> Чикфлик (англ. Chick flick, кино для девчонок) — термин в западной киноиндустрии, под которым понимают кино- и телефильмы, предназначенные прежде всего для женской аудитории.



## 2.2. Пример: профили кинозрителей

Чтобы определить кластеры клиентов с помощью кластеризации методом  $k$ -средних, нам потребуется информация о клиентах, которую можно соизмерять. Общая переменная — это доход. Группы с высоким доходом более склонны приобретать продукцию известных брендов, чем с низким. В итоге магазины смогут использовать эту информацию, чтобы адресовать рекламу дорогих товаров группам с высоким уровнем дохода.

Особенности личности тоже хороший способ группировки клиентов, который лучше показать на примере пользователей Facebook. Пользователей пригласили пройти опрос, чтобы распределить их, исходя из четырех свойств: *экстраверсии* (насколько им в радость социальные взаимодействия), *добросовестности* (насколько они трудолюбивы), *эмоциональности* (как часто они испытывают стресс) и *открытости* (насколько они восприимчивы к новому).

Первичный анализ показал наличие связи между этими личностными особенностями. Добросовестные люди обычно более экстравертны. Кроме того, хотя это проявляется в меньшей степени, но высокоэмоциональные люди имеют тенденцию быть более открытыми. Поэтому для лучшей визуализации этих свойств мы их объединили (добросовестность с экстраверсией, эмоциональность с открытостью) путем сложения очков для каждой пары. После этого мы получили двумерный график.

Суммарные очки черт характера были соотнесены с информацией о связанных с фильмами страницах, которые пользователь лайкнул на Facebook. Это дало нам возможность соотнести группы кинозрителей с профилями.

На рис. 1 мы видим два больших кластера.

- **Светлый:** добросовестные экстраверты, которым нравятся боевики и романтические фильмы.
- **Темный:** эмоциональные и открытые люди, которым нравится авангардное кино и фэнтези.

Фильмы посередине, по-видимому, фавориты семейного просмотра.

Обладая такой информацией, можно разработать таргетированную рекламу. Если зрителю нравится «*50 первых поцелуев*», владелец магазина может порекомендовать другой фильм этого жанра или даже продавать такие фильмы вместе, предложив скидку.

## **2.3. Определение кластеров**

При определении кластеров нам нужно ответить на два вопроса:

1. Сколько кластеров существует?
2. Что включают в себя кластеры?

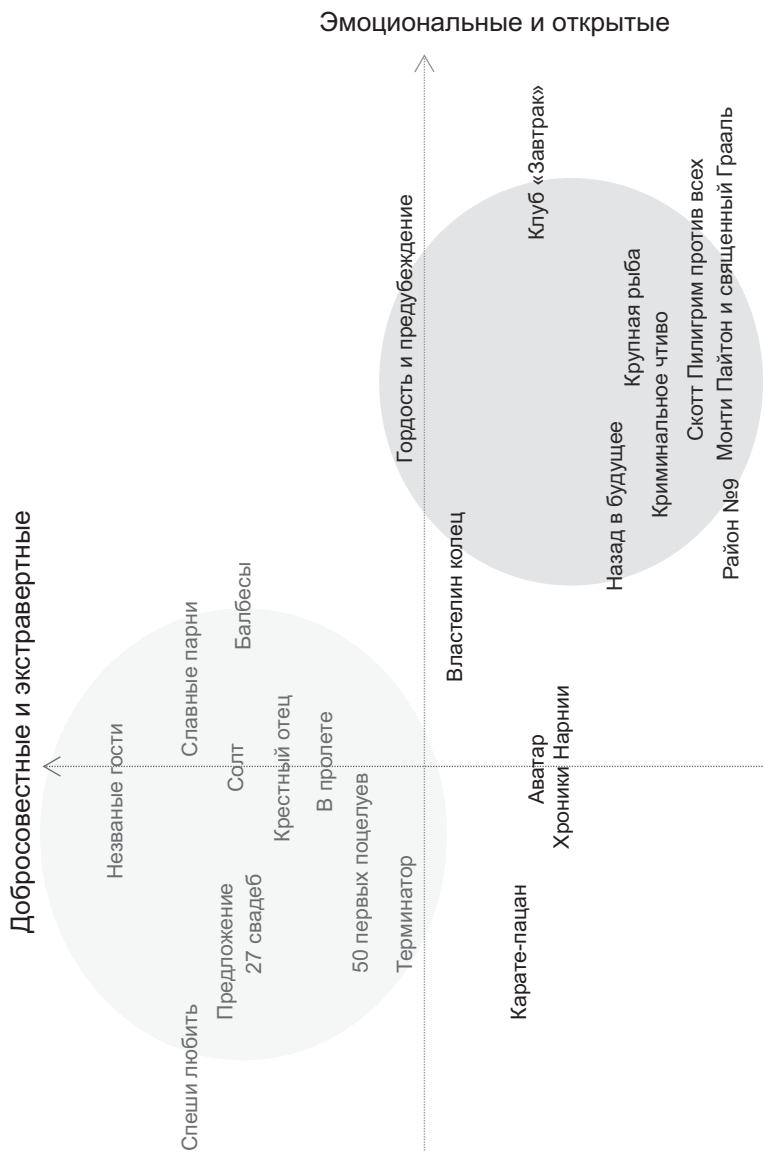


Рис. 1. Персональные профили кинозрителей

## Сколько кластеров существует?

Это субъективно. Хотя на рис. 1 показано два кластера, они могут быть разбиты на кластеры поменьше. Например, темный кластер можно разделить на подкластер «драмы» (включая *Гордость и предубеждение* и *Клуб «Завтрак»*) и подкластер «фэнтези» (включая фильмы *Монти Пайтон и священный Грааль* и *Скотт Пилигрим против всех*).

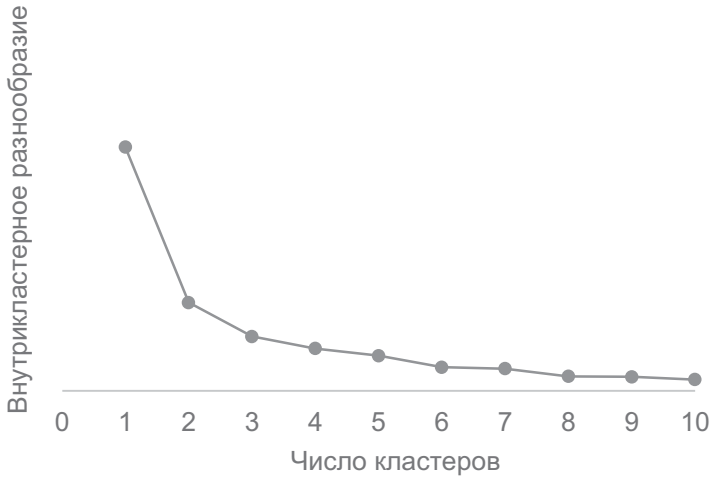
По мере возрастания численности кластеров, члены каждого из них становятся больше похожи друг на друга, но соседние кластеры при этом становятся менее различимы. Если довести это до крайности, то каждый элемент данных окажется в отдельном кластере, что не даст нам никакой полезной информации.

Поэтому нужен компромисс. Число кластеров должно быть достаточно велико, чтобы позволить нам выявить важные для бизнес-решений закономерности, но не слишком, чтобы кластеры сохраняли отчетливые различия.

Одним из способов определить оптимальное количество кластеров является использование так называемого графика каменистой осыпи, или графика Кеттела (scree plot) (рис. 2).

*График осыпи* показывает, насколько снижается разнообразие внутри кластеров при увеличении их числа. Если все члены отнесены к единственному кластеру,

разнообразие максимально. Но по мере увеличения числа кластеров сами они становятся плотнее, а их члены однороднее.



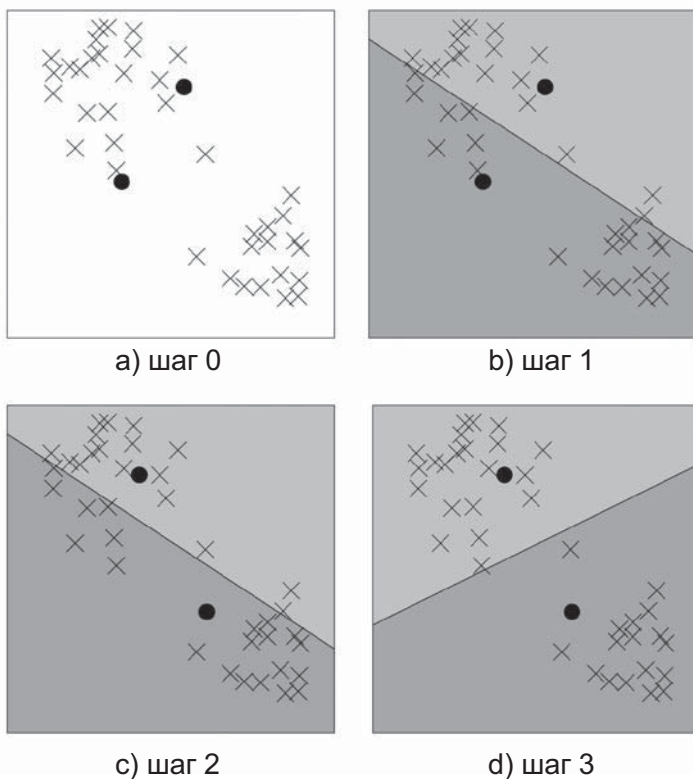
**Рис. 2.** График осыпи показывает изломы, из которых следует, что оптимальное число кластеров от 2 до 3

*Излом* — это острый изгиб на графике осыпи, который предлагает оптимальное число кластеров, исходя из разумной степени внутрикластерного разнообразия. На рис. 2 мы видим излом на двойке, которая соответствует двум кластерам с фильмами на рис. 1. Другой излом, поменьше, находится на тройке, говоря о том, что мы можем ввести третий кластер с семейным кино. А вот введение еще большего их числа уже даст слишком малые кластеры, слабо отличающиеся друг от друга.

После того как мы разобрались с количеством кластеров, можно заняться распределением данных.

### Что включают кластеры?

Данные распределяются по кластерам в итеративном процессе, показанном для двухкластерного примера на рис. 3.



**Рис. 3.** Итеративный процесс кластеризации методом  $k$ -средних

Поскольку хороший кластер содержит похожие элементы данных, мы можем оценить его по тому, как далеко его члены находятся от центра. Но поскольку изначально позиции кластерных центров неизвестны, они берутся приблизительно. Затем элементы данных связывают с ближайшим к ним кластерным центром.

После этого кластерный центр снова вычисляется для своих членов, а для элементов данных процедура повторяется, и если элемент данных окажется ближе к центру другого кластера, его членство будет изменено.

Следующие шаги точно описывают процесс определения членства в кластере и могут использоваться при любом количестве кластеров.

**Шаг 0:** начать с предположения о том, где находятся центры кластеров. Условно можно назвать их псевдоцентрами, поскольку мы пока не знаем, соответствуют ли они в действительности центральному положению.

**Шаг 1:** связать каждый элемент данных с ближайшим псевдоцентром. Сделав это, мы получаем два кластера: светлый и темный.

**Шаг 2:** вычислить новое положение псевдоцентров, ориентируясь на центр отнесенных к кластеру членов.

**Шаг 3:** повторять переназначение членов кластера (шаг 1) и его репозиционирование (шаг 2) до тех пор, пока все изменения в членстве не прекратятся.

Хотя мы рассмотрели только двумерный анализ, группирование в кластеры может быть также выполнено для

трех и более измерений. Этими дополнительными измерениями могут послужить возраст клиента или частота посещения. Хотя такое и трудно визуализировать, мы можем довериться компьютерным программам, которые вычислят за нас многомерные дистанции между элементами данных и кластерными центрами.

## 2.4. Ограничения

Хотя кластеризация методом  $k$ -средних очень полезна, у нее есть ограничения:

**Каждый элемент данных может быть связан только с одним кластером.** Иногда элемент данных находится ровно посередине между двух центров, отчего его включение в эти кластеры равновероятно.

**Предполагается, что кластеры сферичны.** Итеративный процесс поиска ближайшего кластерного центра для элементов данных ограничен его радиусом, поэтому итоговый кластер похож на плотную сферу. Это может стать проблемой, если фактическая форма кластера, например, эллипс. Тогда кластер может быть усечен, а некоторые его члены отнесены к другому.

**Кластеры предполагаются цельными.** Метод  $k$ -средних не допускает того, чтобы они пересекались или были вложены друг в друга.

Вместо принудительного назначения каждого элемента данных в единственный кластер можно воспользоваться



более гибкими методами группировки, которые вычисляют то, с какой вероятностью каждый элемент данных может принадлежать другим кластерам, помогая нам находить несферические или пересекающиеся кластеры.

Несмотря на эти ограничения сила кластеризации методом  $k$ -средних заключается в ее элегантной простоте. Хороший подход состоит в том, чтобы начинать с кластеризации методом  $k$ -средних для изначального понимания структуры данных, а затем привлекать более продвинутые методы, лишенные его недостатков.

## 2.5. Краткие итоги

- Кластеризация методом  $k$ -средних — это способ сгруппировать вместе похожие элементы данных. Число этих кластеров  $k$  должно быть указано заранее.
- Для группировки элементов данных сначала каждый из них соотносится с кластером, а потом обновляются позиции кластерных центров. Эти два шага повторяются до тех пор, пока изменения не будут исчерпаны.
- Кластеризация методом  $k$ -средних лучше работает для сферических, непересекающихся кластеров.



# 3

## **Метод главных компонент**

## 3.1. Изучение пищевой ценности

Представьте, что вы диетолог. Как лучше всего дифференцировать пищевые продукты? По содержанию витаминов? Или белка? Или, может, по тому и другому?



**Рис. 1.** Обычная пирамида питания

Знание о переменных, которые лучше всего дифференцируют ваши данные, может иметь несколько применений:

- **Визуализация.** Отображение элементов на графике с подходящей шкалой может дать их лучшее понимание.

○ **Обнаружение кластеров.** При хорошей визуализации могут быть обнаружены скрытые категории или кластеры. Например, если говорить о пищевых продуктах, мы можем выявить такие широкие категории, как мясо и овощи, а также подкатегории различных видов овощей.

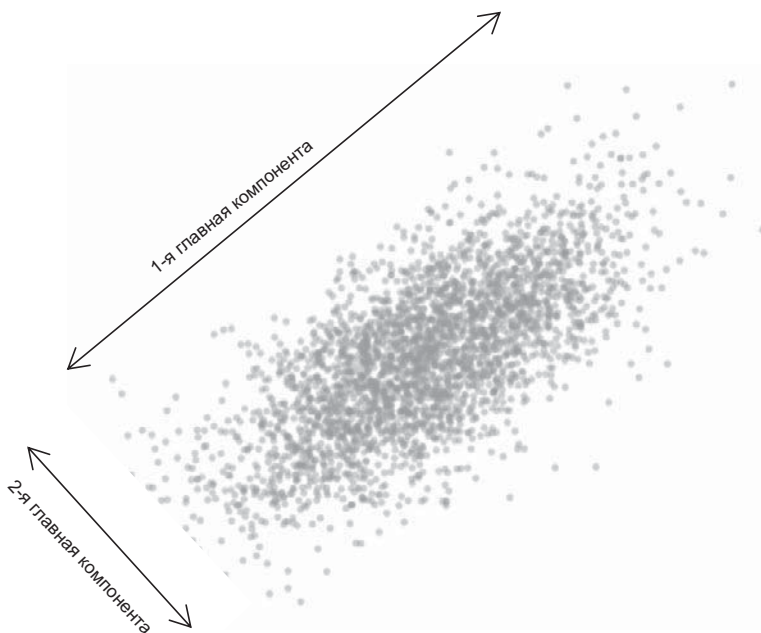
Вопрос в том, как нам получить переменные, которые дифференцируют наши данные наилучшим образом.

## 3.2. Главные компоненты

*Метод главных компонент* (Principal Component Analysis, МГК) — это способ нахождения основополагающих переменных (известных как главные компоненты), которые дифференцируют ваши элементы данных оптимальным образом. Эти главные компоненты дают наибольший разброс данных (рис. 2).

Главная компонента может выражать одну или несколько переменных. Например, мы можем воспользоваться единственной переменной «Витамин С». Поскольку витамин С есть в овощах, но отсутствует в мясе, итоговый график (левая колонка на рис. 3) распределит овощи, но все мясо окажется в одной куче.

Для распределения мясных продуктов мы можем использовать в качестве второй переменной жир, поскольку он присутствует в мясе, но его почти нет в овощах. Тем не менее, поскольку жир и витамин С измеряются в разных единицах, то прежде чем их скомбинировать, мы должны стандартизировать их.



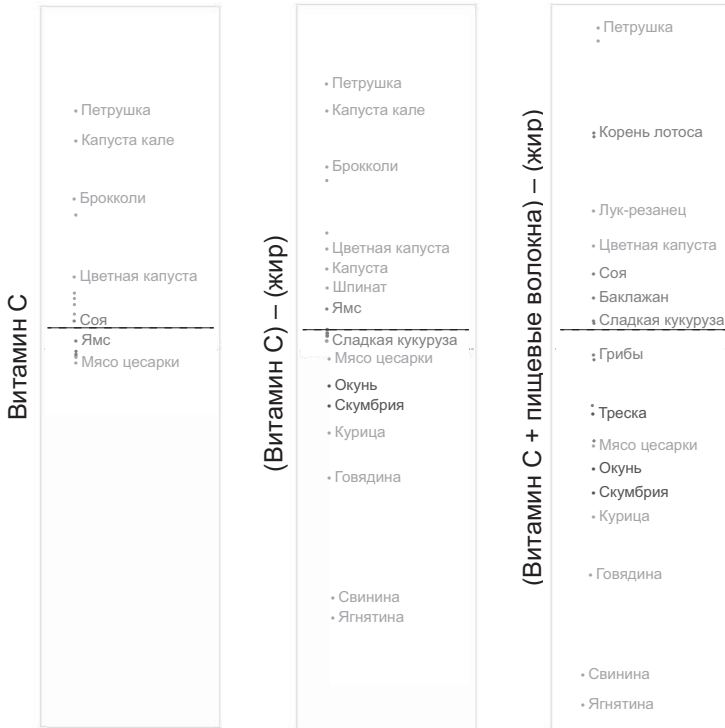
**Рис. 2.** Визуальное представление главных компонент

*Стандартизация* — это выражение каждой переменной в процентилях, которые преобразуют эти переменные в единую шкалу, позволяя нам комбинировать их для вычисления новой переменной:

витамин С – жир

Поскольку витамин С уже распределил овощи вверх, то жир мы вычитаем, чтобы распределить мясо вниз. Комбинирование этих двух переменных поможет нам

распределить как овощи, так и мясные продукты (столбец посередине на рис. 3).



**Рис. 3.** Пищевые продукты, распределенные разными комбинациями переменных

Мы можем улучшить разброс, приняв во внимание пищевые волокна, содержание которых в овощах различается:

(Витамин С + пищевые волокна) – жир

Эта новая переменная дает нам оптимальный разброс данных (правый столбец на рис. 3).

В то время как мы получили главные компоненты в этом примере методом проб и ошибок, МГК может делать это на системной основе. Мы увидим, как это работает, на следующем примере.

### **3.3. Пример: анализ пищевых групп**

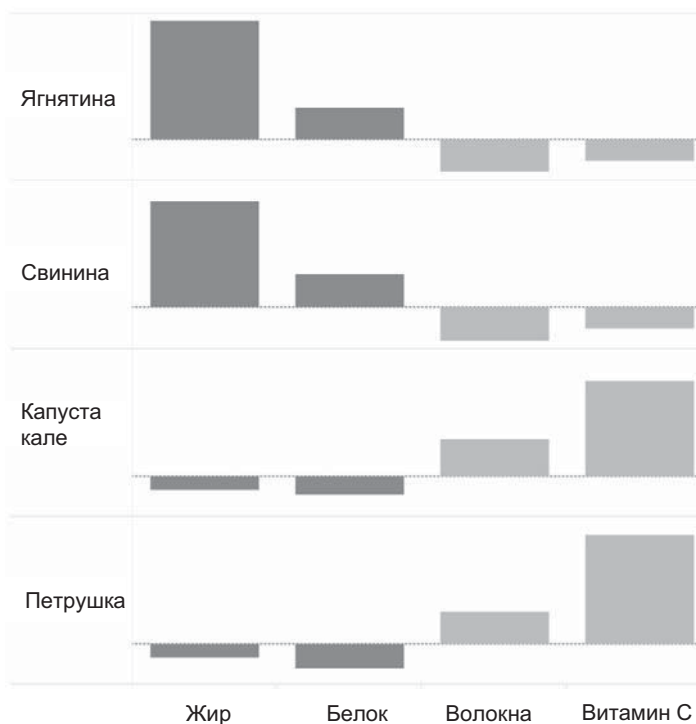
Используя данные Министерства сельского хозяйства США, мы проанализировали питательные свойства случайного набора продуктов, рассмотрев четыре пищевых переменных: жиры, белки, пищевые волокна и витамин С. Как видно на рис. 4, определенные питательные вещества часто встречаются в продуктах вместе.

В частности, уровни содержания жиров и белков возрастают в одном направлении, противоположном тому, в котором растут уровни пищевых волокон и витамина С. Мы можем подтвердить наши предположения, проверив, какие переменные коррелируют (см. раздел 6.5). И действительно, мы находим значимую положительную корреляцию как между уровнями белков и жиров ( $r = 0,56$ ), так и между уровнями пищевых волокон и витамина С ( $r = 0,57$ ).

Таким образом, вместо анализа четырех пищевых переменных по отдельности мы можем скомбинировать высококоррелирующие из них, получив для рассмотрения



всего две. Поэтому метод главных компонент относят к техникам *уменьшения размерности*.



**Рис. 4.** Сравнение пищевой ценности различных продуктов

Применив его к нашему пищевому набору данных, мы получим главные компоненты, изображенные на рис. 5.

Каждая главная компонента — это комбинация пищевых переменных, значение которой может быть положительным, отрицательным или близким к нулю. Например,

чтобы получить компоненту 1 для отдельного продукта, мы можем вычислить следующее:

$$.55(\text{пищевые волокна}) + .44(\text{Витамин С}) - .45(\text{жир}) - .55(\text{белок})$$

	PC1	PC2	PC3	PC4
Жир	-0,45	0,66	0,58	0,18
Белок	0,55	0,21	-0,46	-0,67
Волокна	0,55	0,19	0,43	-0,69
Витамин С	0,44	0,70	-0,52	0,22

**Рис. 5.** Главные компоненты — это комбинации пищевых переменных. Светло-серые ячейки у одной и той же компоненты представляют собой связанные переменные

То есть вместо комбинирования переменных методом проб и ошибок, как мы делали раньше, метод главных компонент сам вычисляет точные формулы, при помощи которых можно дифференцировать наши позиции.

Обратите внимание, что основная для нас главная компонента 1 (PC1) сразу объединяет жиры с белками, а пищевые волокна с витамином С, и эти пары обратно пропорциональны.

В то время как PC1 дифференцирует мясо от овощей, компонента 2 (PC2) подробнее идентифицирует внутренние подкатегории мяса (исходя из жирности) и овощей

(по содержанию витамина С). Лучший разброс данных мы получим, используя для графика обе компоненты (рис. 6).



**Рис. 6.** График продуктов при использовании двух главных компонент

У мясных товаров низкие значения компоненты 1, поэтому они сосредоточены в левой части графика, в противоположной стороне от овощных. Видно также, что среди не

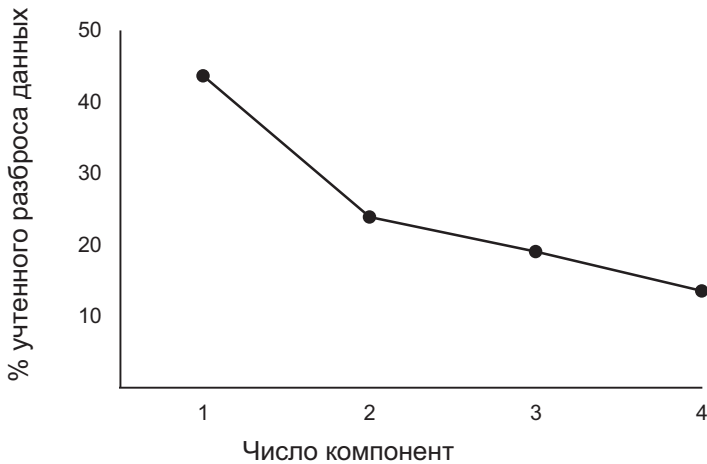
овощных товаров низкое содержание жиров у морепродуктов, поэтому значение компоненты 2 для них меньше, и сами они тяготеют к нижней части графика. Схожим образом у тех овощей, которые не являются зеленью, низкие значения компоненты 2, что видно в нижней части графика справа.

**Выбор количества компонент.** В этом примере созданы четыре главных компоненты по числу изначальных переменных в наборе данных. Поскольку главные компоненты создаются на основе обычных переменных, информация для распределения элементов данных ограничивается их первоначальным набором.

Вместе с тем для сохранения простоты и масштабируемости результатов нам следует выбирать для анализа и визуализации только несколько первых главных компонент. Главные компоненты отличаются по эффективности распределения элементов данных, и первый из них делает это в максимальной степени. Число главных компонент для рассмотрения определяют с помощью *графика осыпи*, который мы рассмотрели в предыдущей главе.

График показывает снижающуюся эффективность последующих главных компонент в дифференцировании элементов данных. Как правило, используется такое количество главных компонент, которое соответствует положению острого *излома* на графике осыпи.

На рис. 7 излом расположен на отметке в две компоненты. Это означает, что хотя три и более главных компонент могли бы лучше дифференцировать элементы данных,



**Рис. 7.** На графике осыпи виден излом, обозначающий, что оптимальное число главных компонент — две

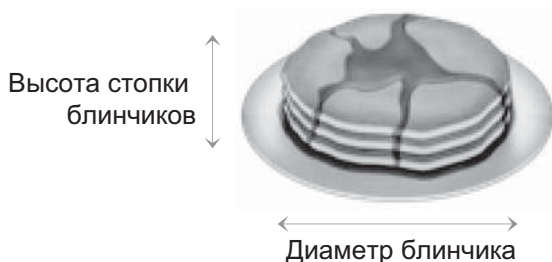
эта дополнительная информация может не оправдать сложности итогового решения. Как видно из графика осыпи, две первые главные компоненты уже дают 70 %-ный разброс. Использование небольшого числа главных компонент для анализа данных дает гарантию того, что схема подойдет и для будущей информации.

## 3.4. Ограничения

Метод главных компонент — это полезный способ анализа наборов данных с несколькими переменными. Однако у него есть и недостатки.

**Максимизация распределения.** МГК исходит из важного допущения того, что наиболее полезны те измерения,

которые дают наибольший разброс. Однако это не всегда так. Известным контрпримером является задача с подсчетом блинчиков в стопке.



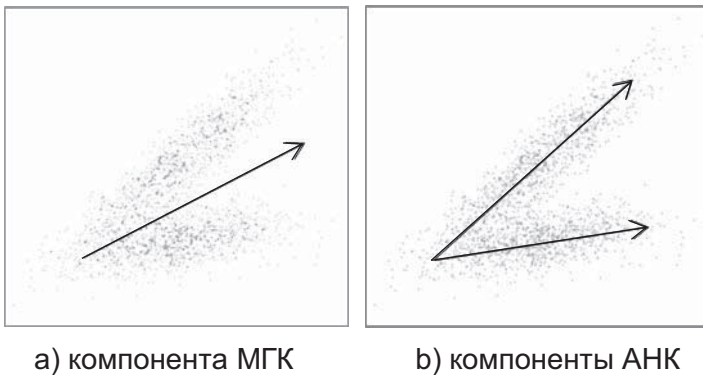
**Рис. 8.** Аналогия с подсчетом блинчиков

Чтобы сосчитать блинчики, мы отделяем один от другого по вертикальной оси (то есть по высоте стопки). Однако если стопка невелика, МГК ошибочно решит, что лучшей главной компонентой будет горизонтальная ось (диаметр блинчиков), из-за того что в этом измерении можно найти больший разброс значений.

**Интерпретация компонент.** Главная трудность с МГК состоит в том, что необходима интерпретация сгенерированных компонент, и иногда нужно очень постараться, чтобы объяснить, почему переменные должны быть скомбинированы именно выбранным способом.

Тем не менее могут выручить предварительные общие сведения. В нашем примере с продуктами скомбинировать пищевые переменные для главных компонент нам помогают именно предварительные знания об их категориях.

**Ортогональные компоненты.** МГК всегда формирует *ортогональные* главные компоненты, то есть такие, которые размещаются в отношении друг друга под углом  $90^\circ$ . Однако это допущение может оказаться излишним при работе с неортогональными информационными измерениями. Для решения этой проблемы мы можем воспользоваться альтернативным методом, известным как *анализ независимых компонент (АНК)*.



**Рис. 9.** Сравнение того, как МГК и АНК определяют главные компоненты

АНК допускает неортогональность компонент, но запрещает ситуации взаимного перекрытия содержащейся информации (рис. 9). Каждая из выделенных им главных компонент будет содержать уникальную информацию о наборе данных. Помимо обхода ортогонального ограничения АНК в поисках главных компонент принимает во внимание не один только разброс данных и поэтому менее подвержен «блинчиковой ошибке».

Хотя АНК может показаться совершенное, самым популярным способом уменьшения размерности остается МГК, и понимание того, как он работает, весьма полезно. В случае сомнений имеет смысл всегда запускать АНК в дополнение к МГК для получения более общей картины.

### 3.5. Краткие итоги

- Метод главных компонент — это способ *уменьшения размерности*, который позволяет выразить наши данные через небольшой набор переменных, называемых *главными компонентами*.
- Каждая главная компонента — это некая сумма из начальных переменных. Лучшие из них могут быть использованы для анализа и визуализации.
- МГК лучше всего работает с теми информационными измерениями, которые дают больший разброс данных и ортогональны друг другу.



# 4

## **Ассоциативные правила**

## 4.1. Поиск покупательских шаблонов

Отправляясь в продуктовый магазин, вы наверняка возьмете с собой список покупок, исходя из ваших потребностей и предпочтений. Домохозяйка, возможно, купит полезные продукты для семейного ужина, а холостяк, скорее всего, возьмет пива и чипсов. Понимание таких закономерностей поможет увеличить продажи сразу несколькими способами. Например, если пара товаров  $X$  и  $Y$  часто покупается вместе, то:

- реклама товара  $X$  может быть направлена на покупателей товара  $Y$ ;
- товары  $X$  и  $Y$  могут быть размещены на одной и той же полке, чтобы побудить покупателей одного товара к приобретению второго;
- товары  $X$  и  $Y$  могут быть скомбинированы в некий новый продукт, такой как  $X$  со вкусом  $Y$ .

Узнать, как именно товары связаны друг с другом, нам помогут *ассоциативные правила*. Кроме увеличения продаж ассоциативные правила могут быть также использованы в других областях. В медицинской диагностике,























например, понимание сопутствующих симптомов может улучшить заботу о пациентах.

## 4.2. Поддержка, достоверность и лифт

Существуют три основные меры для определения ассоциаций.

**Мера 1: поддержка.** Поддержка показывает *то, как часто данный товарный набор появляется*, что измеряется долей покупок, в которых он присутствует. В табл. 1 {яблоко} появляется в четырех из восьми покупок, значит, его поддержка 50 %. Товарные наборы могут содержать и несколько элементов. Например, поддержка

**Таблица 1.** Пример покупок

Покупка 1				
Покупка 2				
Покупка 3				
Покупка 4				
Покупка 5				
Покупка 6				
Покупка 7				
Покупка 8				

набора {яблоко, пиво, рис} — два из восьми, то есть 25 %. Для определения часто встречающихся товарных наборов может быть установлен *порог поддержки*. Товарные наборы, встречаемость которых выше заданного числа, будут считаться частотными.

$$\text{Поддержка } \{\text{яблоко}\} = \frac{4}{8}$$

Рис. 1. Мера «поддержка»

**Мера 2: достоверность.** Достоверность показывает, как часто товар  $Y$  появляется вместе с товаром  $X$ , что выражается как  $\{X \rightarrow Y\}$ . Это измеряется долей их одновременных появлений. Согласно табл. 1, достоверность {яблоко  $\rightarrow$  пиво} соответствует трем из четырех, то есть 75 %.

$$\text{Достоверность } \{\text{яблоко} \rightarrow \text{пиво}\} = \frac{\text{Поддержка } \{\text{яблоко}, \text{пиво}\}}{\text{Поддержка } \{\text{яблоко}\}}$$

Рис. 2. Мера «достоверность»

Одним из недостатков этой меры является то, что она может исказить степень важности предложенной ассоциации. Пример на рис. 2 принимает во внимание только то, как часто покупают яблоки, но не то, как часто покупают пиво. Если пиво тоже довольно популярно, что и видно из табл. 1, то неудивительно, что покупки, включающие яблоки, нередко содержат и пиво, таким образом увеличивая меру достоверности. Тем не менее мы можем

принять во внимание частоту обоих товаров, используя третью меру.

**Мера 3: лифт.** Лифт отражает *то, как часто товары X и Y появляются вместе, одновременно учитывая, с какой частотой появляется каждый из них.*

Таким образом, лифт {яблоко->пиво} равен достоверности {яблоко->пиво}, деленной на частоту {пива}.

$$\text{Лифт } \{\text{яблоко} \rightarrow \text{пиво}\} = \frac{\text{Поддержка } \{\text{яблоко}, \text{пиво}\}}{\text{Поддержка } \{\text{яблоко}\} \times \text{Поддержка } \{\text{пиво}\}}$$

**Рис. 3.** Мера «лифт»

Согласно табл. 1, лифт для {яблоко->пиво} равен единице, что означает отсутствие связи между товарными позициями. Значения лифта больше единицы означают, что товар Y *вероятно* купят вместе с товаром X, а значение меньше единицы — что их совместная покупка *маловероятна*.

## 4.3. Пример: ведение продуктовых продаж

Чтобы продемонстрировать использование мер ассоциации, мы проанализировали данные одного продуктового магазина за 30 дней. Рисунок 4 показывает ассоциации между товарными парами, в которых достоверность выше 0,9 %, а лифт — 2,3. Большие круги означают высокую поддержку, а темные — большой лифт.

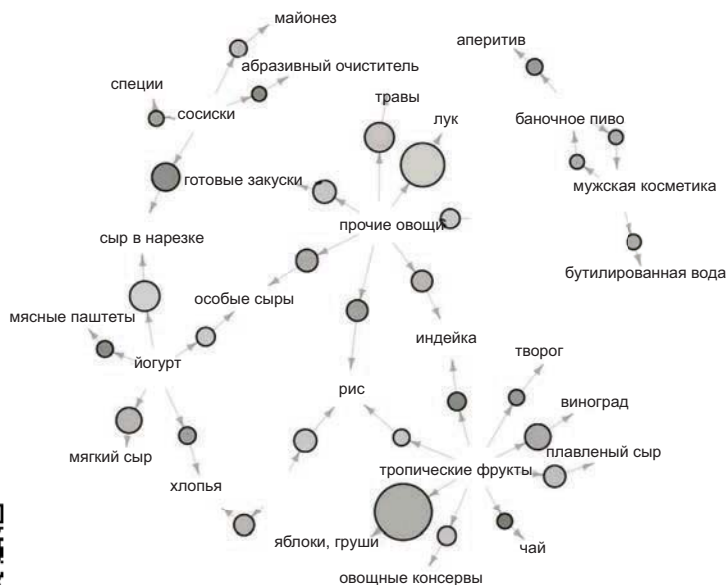


Рис. 4. Граф ассоциаций между товарными позициями

Мы можем наблюдать такие закономерности в покупках:

- чаще всего покупают яблоки и тропические фрукты;
- другая частая покупка: лук и овощи;
- если кто-то покупает сыр в нарезке, он, скорее всего, возьмет и сосиски;
- если кто-то покупает чай, то он, вероятно, возьмет и тропические фрукты.

Вспомним, что одним из недостатков меры «достоверность» является то, что она может создавать искаженное впечатление о значимости ассоциации. Чтобы показать

это, рассмотрим три ассоциативных правила, содержащих пиво.

**Таблица 2.** Ассоциативные метрики для трех правил, связанных с пивом

Покупка	Поддержка	Достоверность	Лифт
Пиво → Газировка	1,38 %	17,8 %	1,0
Пиво → Ягоды	0,08 %	1,0 %	0,3
Пиво → Мужская косметика	0,09 %	1,2 %	2,6

Правило {пиво->газировка} имеет высокую достоверность — 17,8 %. Однако и пиво, и газировка вообще часто появляются среди покупок (табл. 3), поэтому их ассоциация может оказаться простым совпадением. Это подтверждается значением лифта, указывающим на отсутствие связи между газировкой и пивом.

**Таблица 3.** Значение поддержки для отдельных товаров в правилах, связанных с пивом

Покупка	Поддержка
Пиво	7,77 %
Газировка	17,44 %
Ягоды	3,32 %
Мужская косметика	0,46 %

С другой стороны, правило {пиво->мужская косметика} имеет низкую достоверность из-за того, что мужскую

косметику вообще реже покупают. Тем не менее если кто-то покупает ее, он, вероятно, купит также и пиво, на что указывает высокое значение лифта в 2,6. Для пары {пиво-→ягоды} верно обратное. Видя лифт меньше единицы, мы заключаем, что если кто-то покупает пиво, то он, скорее всего, не возьмет ягод.

Хотя несложно определить частотность отдельных товарных наборов, владелец бизнеса обычно заинтересован в получении полного списка часто покупаемых товарных наборов. Для этого потребуется вычислить значения поддержки для каждого возможного товарного набора, после чего выбрать те, поддержка которых выше заданного порога.

В магазине со всего десятью товарами суммарное число возможных конфигураций для анализа составит 1023 (то есть  $2^{10} - 1$ ), и это число экспоненциально возрастает для магазина с сотнями товарных позиций. Ясно, что нам потребуется более эффективное решение.

## 4.4. Принцип Apriori

Одним из способов снизить количество конфигураций рассматриваемых товарных наборов является использование *принципа Apriori*. Если вкратце, то принцип Apriori утверждает, что если какой-то товарный набор редкий, то и большие наборы, которые его включают, тоже должны быть редки. Это значит, что если редким является, скажем, {пиво}, то редким должно быть и сочетание {пиво, пицца}. Таким образом, составляя список



частотных товарных наборов, мы уже не будем рассматривать ни пару {пиво, пицца}, ни какую-либо другую с содержанием пива.

## **Поиск товарных наборов с высокой поддержкой**

С применением принципа Apriori мы можем получить список частотных товарных наборов, используя следующие шаги.

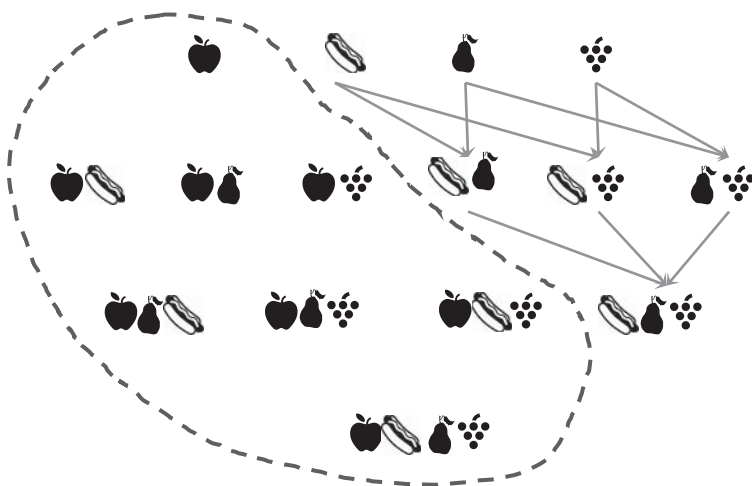
**Шаг 0:** начать с товарных наборов, содержащих всего один элемент, таких как {яблоки} или {груши}.

**Шаг 1:** вычислить поддержку для каждого товарного набора. Оставить наборы, удовлетворяющие порогу, и отбросить остальные.

**Шаг 2:** увеличить размер анализируемого товарного набора на единицу и сгенерировать все возможные конфигурации, используя товарные наборы из предыдущего шага.

**Шаг 3:** повторять шаги 1 и 2, вычисляя поддержку для возрастающих товарных наборов до тех пор, пока они не закончатся.

На рис. 5 показано, как число рассматриваемых товарных наборов может значительно сократиться при использовании принципа Apriori. Если у элемента {яблоки} низкая поддержка, то он будет удален из списка анализируемых товарных наборов вместе со всем, что его содержит, тем самым это сократит число наборов для анализа более чем вдвое.



**Рис. 5.** Товарные наборы в пределах пунктирной линии будут отброшены

## Поиск товарных правил с высокой достоверностью или лифтом

Кроме определения товарных наборов с высокой поддержкой, принцип *Apriori* также может помочь найти товарные ассоциации с высокой достоверностью или лифтом. Поиск этих ассоциаций требует меньше вычислений, поскольку если товарные наборы с высокой поддержкой известны, то достоверность и лифт вычисляются уже с использованием значения поддержки.

Возьмем для примера задачу поиска правил с высокой достоверностью. Если правило {пиво, чипсы → яблоки}

имеет низкую достоверность, то и все другие правила с теми же образующими элементами и яблоком с правой стороны будут тоже иметь низкую достоверность, включая {пиво->яблоки, чипсы} и {чипсы->яблоки, пиво}. Как и прежде, эти правила могут быть отброшены благодаря принципу Apriori, тем самым снижая число потенциально рассматриваемых правил.

## 4.5. Ограничения

**Требует долгих вычислений.** Хотя принцип Apriori и снижает число потенциальных товарных наборов для рассмотрения, оно все еще может быть достаточно значительным, если список товаров большой или указан низкий порог поддержки. В качестве альтернативного решения можно сократить число сравнений, используя расширенные структуры данных, чтобы отобрать потенциальные товарные наборы с большей эффективностью.

**Ложные ассоциации.** В больших наборах данных ассоциации могут быть чистой случайностью. Чтобы убедиться, что обнаруженные ассоциации масштабируемы, их нужно оценить (см. раздел 1.4).

Несмотря на эти ограничения, ассоциативные правила остаются интуитивно-понятным методом обнаружения закономерностей в наборах данных с управляемым размером.

## 4.6. Краткие итоги

- Ассоциативные правила выявляют то, как часто элементы появляются вообще и в связи с другими.
- Есть три основных способа оценки ассоциации:
  1. *Поддержка*  $\{X\}$  показывает, как часто  $X$  появляется.
  2. *Достоверность*  $\{X \rightarrow Y\}$  показывает, как часто  $Y$  появляется в присутствии  $X$ .
  3. *Лифт*  $\{X \rightarrow Y\}$  показывает, как часто элементы  $X$  и  $Y$  появляются вместе по сравнению с тем, как часто они появляются по отдельности.
- *Принцип Apriori* ускоряет поиск часто встречающихся товарных наборов, отбрасывая значительную долю редких.

# 5

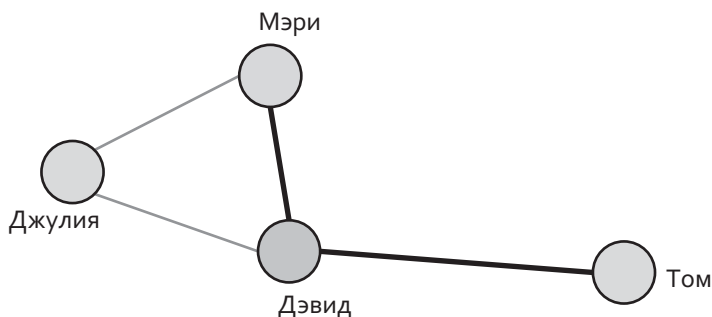
## **Анализ социальных сетей**

## 5.1. Составление схемы отношений

Большинство из нас имеет множество кругов общения, включающих такие категории людей, как родственники, коллеги или одноклассники. Чтобы выяснить, как устроены отношения всех этих людей, определив, например, активных персон и то, как они влияют на групповую динамику, мы можем воспользоваться методом под названием *анализ социальных сетей* (Social Network Analysis). Этот метод можно применять в вирусном маркетинге, моделировании эпидемий и даже для стратегий в командных играх. Тем не менее он больше известен своим использованием для анализа отношений в социальных сетях, что и дало ему название. На рис. 1 пример того, как анализ социальных сетей показывает отношения.

Рисунок 1 показывает сеть из четырех индивидов, также известную как *граф*, в котором каждый из этих персон представлен *узлом* (node). Отношения между узлами представлены линиями, называемыми *ребрами* (edges).

Каждое ребро может иметь *вес*, показывающий силу отношений.



**Рис. 1.** Простая сеть друзей. Более близкие отношения показаны утолщенными линиями

Из рис. 1 мы можем заключить:

- Дэвид имеет больше всех связей, будучи знакомым с остальными тремя персонами;
- Том не знает никого, кроме Дэвида, с которым они близкие друзья;
- Джулия знает Мэри и Дэвида, но не близка с ними.

Кроме отношений анализ социальных сетей может строить схемы и для других сущностей, при условии, что между ними есть связи. В этой главе мы воспользуемся им для анализа международной сети торговли оружием, чтобы выявить доминирующие силы и их сферы влияния.

## 5.2. Пример: геополитика в торговле оружием

Мы получили данные о двусторонних трансферах основных видов обычных вооружений из *Стокгольмского международного института по исследованию проблем мира*. Военные поставки были выбраны в качестве косвенного показателя двусторонних отношений, поскольку должны свидетельствовать о тесной связи стран на международной арене.

В этом анализе мы стандартизировали стоимость оружия на уровне цен 1990 года в долларах США, после чего приняли в расчет только сделки, сумма которых превысила 100 млн долларов. Чтобы учесть флуктуации в торговле оружием, обусловленные производственными циклами новых технологий, мы рассмотрели 10-летний период, с 2006 по 2015 год, построив сеть из 91 узла и 295 ребер.

Для визуализации сети использовался *силовой алгоритм* (force-directed algorithm): узлы без связей отталкиваются друг от друга, а связанные узлы, наоборот, притягиваются с той степенью близости, которая отражает силу их связи (рис. 2). Например, максимальный объем торговли зафиксирован между Россией и Индией (\$ 22,3 млрд), поэтому эти государства соединены толстой линией и близко расположены.

После анализа получившейся сети лувенским методом (Louvain Method, описан в следующем разделе) геополитические альянсы были сгруппированы в три кластера.



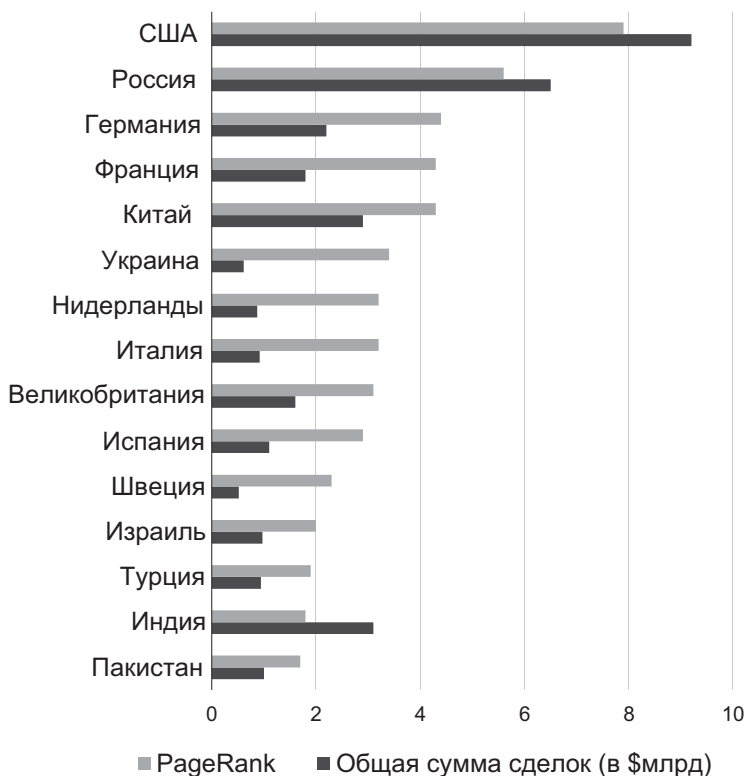


- **Светло-серый:** это крупнейший кластер, в котором доминируют США и который включает их союзников, таких как Великобритания и Израиль.
- **Светлый:** в нем лидирует Германия, и он включает в основном европейские страны, а также тесно связан со светло-серым кластером.
- **Темный:** в этом кластере доминируют Россия и Китай, он дистанцирован от двух других и включает в основном азиатские и африканские государства.

Кластеры отражают геополитические реалии XXI столетия, такие как долгосрочные альянсы между западными нациями, поляризацию между демократическими и коммунистическими странами и возрастающую роль противостояния между США и Китаем.

Кроме группировки в кластеры мы также проранжировали отдельные страны по уровню их влияния, воспользовавшись алгоритмом PageRank (описывается дальше). На рис. 3 представлены 15 самых влиятельных государств, которые также отмечены на рис. 2 более крупными узлами и подписями.

Согласно нашему анализу, в пятерку самых влиятельных стран входят США, Россия, Германия, Франция и Китай. Эти результаты подтверждаются тем обстоятельством, что четыре из пяти этих государств имеют влияние еще и как члены Совета Безопасности ООН.



**Рис. 3.** Самые влиятельные страны в торговле оружием, согласно алгоритму PageRank. Значение PageRank для каждой страны показано светлым, а торговый объем — темным

В следующих разделах мы рассмотрим методы, использованные для выделения кластеров и ранжирования стран.

### 5.3. Лувенский метод

Как видно на рис. 2, можно найти кластеры сети путем группировки узлов. Изучение этих кластеров поможет понять, чем различаются части сети и как они соединены.

*Лувенский метод* — один из способов определения кластеров сети. Он подбирает различные кластерные конфигурации, чтобы: 1) максимизировать число и силу связей между узлами в одном кластере; 2) минимизировать при этом связи между узлами различных кластеров. Степень удовлетворения этим двум условиям известна как *модулярность* (modularity), и более высокая модулярность — признак более оптимальных кластеров.

Чтобы получить оптимальную конфигурацию кластеров, лувенский метод итеративно проходит следующие стадии.

**Стадия 0:** рассматривает каждый узел в качестве кластера, то есть начинает с числа кластеров, равного числу узлов.

**Стадия 1:** меняет кластерное членство узла, если это приводит к улучшению модулярности. Если модулярность больше нельзя улучшить, узел остается на месте. Это повторяется для каждого узла до тех пор, пока изменения кластерного членства не будут исчерпаны.

**Стадия 2:** строит грубую версию сети, в которой каждый кластер, найденный на стадии 1, представлен отдельным

узлом, и объединяет бывшие межкластерные соединения в утолщенные ребра этих новых узлов в соответствии с их весом.

**Стадия 3:** повторяет стадии 1 и 2 до тех пор, пока не закончатся дальнейшие изменения членства и размера связей.

Таким образом, лувенский метод помогает нам выявить более значимые кластеры, начав с обнаружения малых из них, а затем при необходимости соединяя их.

Простота и эффективность делают лувенский метод популярным решением для кластеризации сети. Однако он имеет свои ограничения.

**Важные, но малые кластеры могут быть поглощены.** Итеративный процесс слияния кластеров может привести к тому, что значимые, но небольшие кластеры будут обойдены вниманием. Чтобы избежать этого, мы можем при необходимости проверять идентифицированные кластеры на промежуточных фазах итераций.

**Множество возможных кластерных конфигураций.** Для сетей, содержащих перекрывающиеся или вложенные кластеры, определить оптимальное кластерное решение может оказаться трудным. Тем не менее, когда имеются несколько решений с высокой модулярностью, мы можем сверить кластеры с другими информационными источниками, что мы и проделали на рис. 2, приняв во внимание географическое местоположение и политическую идеологию.

## 5.4. Алгоритм PageRank

Поскольку кластеры выявляют области высокой концентрации взаимодействий, эти взаимодействия могут управляться ведущими узлами, вокруг которых эти кластеры и сформированы. Для определения этих доминирующих узлов мы можем использовать их ранжирование.

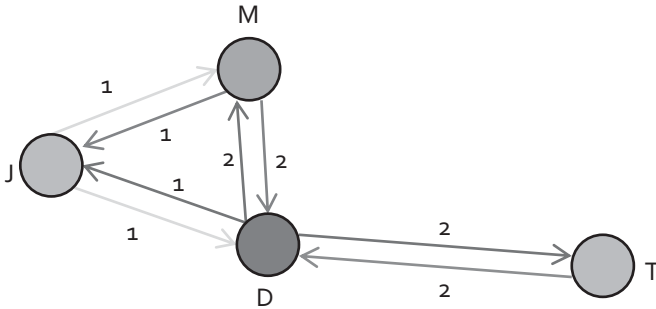
Алгоритм *PageRank*, названный по имени сооснователя Google Ларри Пейджа, стал одним из первых алгоритмов Google для ранжирования веб-сайтов. Хотя мы и опишем PageRank в контексте ранжирования веб-сайтов, он может быть использован для того, чтобы классифицировать узлы любого типа.

Значение PageRank для веб-сайта определяется тремя факторами.

- **Число ссылок.** Если на один веб-сайт ссылаются другие, то он, скорее всего, привлекает больше пользователей.
- **Сила ссылок.** Чем чаще переходят по этим ссылкам, тем больше трафик сайта.
- **Источник ссылок.** Ранг веб-сайта повышается и оттого, что на него ссылаются другие высокоранговые сайты.

Чтобы увидеть работу PageRank, посмотрим на пример сети на рис. 4, где узлы представляют веб-сайты, а ребра — гиперссылки.

Входящая гиперссылка с бóльшим весом означает бóльший объем трафика для сайта. На рис. 4 видно, что посетитель сайта *M* с вдвое большей вероятностью посетит сайт *D*, чем *J*, и может никогда не посетить сайт *T*.

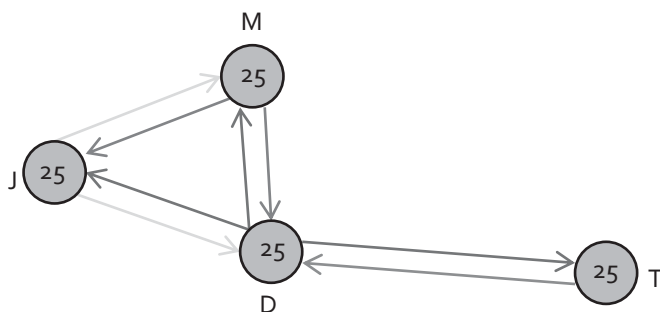


**Рис. 4.** Сеть, в которой узлы — это веб-сайты, а ребра — гиперссылки

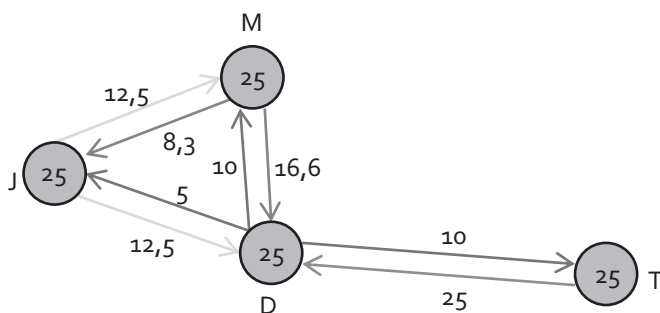
Чтобы понять, какой сайт привлекает больше пользователей, мы можем смоделировать поведение сайта из рис. 4 для 100 пользователей и посмотреть, на какой сайт они в итоге попадут.

Сначала мы равно распределим 100 пользователей по четырем веб-сайтам, как показано на рис. 5.

Затем мы перераспределим пользователей каждого сайта в соответствии с его исходящими ссылками. Например, две трети пользователей сайта *M* отправятся на сайт *D*, в то время как оставшаяся треть посетит сайт *J*. Ребра на рис. 6 показывают число приходящих и уходящих пользователей для каждого сайта.



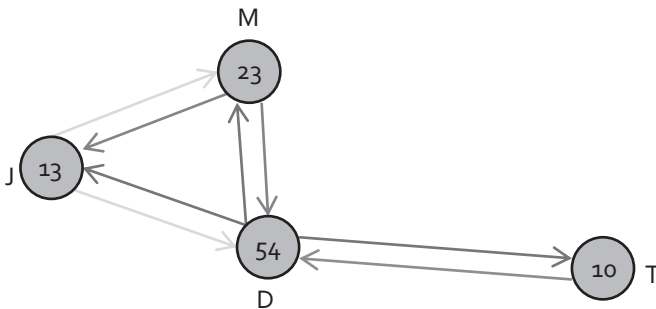
**Рис. 5.** Начальное положение, в котором 100 пользователей распределены по четырем веб-сайтам



**Рис. 6.** Перераспределение пользователей, основанное на силе исходящих ссылок

После перераспределения всех пользователей на сайте *M* оказалось около 23 пользователей, из которых 10 пришли с сайта *D* и 13 с сайта *J*. Рисунок 7 показывает результаты распределения пользователей по каждому сайту, округленные до целого.





**Рис. 7.** Число пользователей на каждом веб-сайте после распределения

Чтобы получить значение PageRank для каждого сайта, нужно повторять этот процесс до тех пор, пока численность пользователей сайта не перестанет меняться. Итоговое число пользователей для каждого веб-сайта будет соответствовать его значению PageRank: чем больше пользователей он привлечет, тем выше его ранг.

Тем же способом с помощью PageRank мы можем измерить и влияние государства. В сети, иллюстрирующей торговлю оружием, страной с высоким значением PageRank будет та, которая участвует во многих значительных торговых сделках с другими высокоранговыми странами, что делает ее влиятельным игроком в мировых военных поставках.

Несмотря на простоту использования, у алгоритма PageRank есть недостаток: **необъективность в отношении старых узлов**. Например, хотя новая веб-страница

и может содержать отличный контент, ее относительная безвестность в момент появления даст ей низкое значение PageRank, что потенциально может привести к исключению из перечней рекомендуемых сайтов. Чтобы избежать этого, значения PageRank могут регулярно обновляться, давая новым сайтам возможность поднимать свои ранги по мере зарабатывания репутации.

Тем не менее такое смещение не всегда критично, особенно при моделировании доминирования за долгие периоды времени, например, когда мы ранжируем страны, исходя из степени их влияния. Это показывает то, как ограничения алгоритмов могут не быть их недостатками, в зависимости от целей исследования.

## 5.5. Ограничения

Хотя методы кластеризации и ранжирования позволяют нам получить очень интересные результаты, интерпретировать их нужно с большой осторожностью.

Возьмем, к примеру, наше использование данных по поставкам оружия для оценки влиятельности государств. У такой упрощенной оценки есть несколько подводных камней.

**Игнорирование дипломатических отношений при отсутствии покупок вооружения.** Большинство ребер проведены между экспортерами и импортерами оружия. Таким образом, дружественные отношения между странами, обе

из которых являются импортерами (либо экспортерами), не отражены.

**Игнорирование других соображений.** Возможно, нужно учесть сложившиеся системы отношений, ограничивающие потенциальных покупателей. Кроме того, страны-экспортеры при принятии решений о продаже оружия могут предпочесть двусторонним отношениям внутренние сделки (например, из экономических соображений). Это могло бы объяснить, почему Украина, значительный экспортер оружия, получила шестой ранг, вопреки отсутствию репутации влиятельной страны.

Поскольку обоснованность наших выводов зависит от того, насколько качественное построение для анализа дают данные, используемые для генерации сети, то они должны выбираться с особой тщательностью. Чтобы убедиться, что наши исходные данные и методы анализа достаточно надежны, мы должны проверять наши результаты по другим источникам информации.

## **5.6. Краткие итоги**

- *Анализ социальных сетей* — это метод, позволяющий строить схему и анализировать отношения между сущностями.
- *Лувенский метод* определяет кластеры внутри сети тем способом, который максимизирует взаимодействие внутри кластеров и минимизирует — между. Он луч-

ше работает, когда кластеры имеют сходный размер и дискретны.

- Алгоритм *PageRank* ранжирует узлы в сети, исходя из числа ссылок, а также из их силы и источника. Хотя он помогает нам идентифицировать ведущие узлы сети, он также имеет необъективность в отношении новых узлов, которые еще не успели обзавестись нужными ссылками.

# 6

## **Регрессионный анализ**

## 6.1. Выведение линии тренда

Линии тренда — популярный инструмент для прогнозирования, поскольку они просты как для вычисления, так и для понимания. Достаточно открыть любую ежедневную газету, чтобы увидеть графики трендов в самых различных областях: от цен на акции до прогноза погоды.

Общие тренды обычно применяют единственный предиктор для предсказания результата, используя, например, время (предиктор) для прогнозирования цен на акции компании (результат). Однако можно улучшить предсказание цен на акции, добавив другие предикторы, такие как уровень продаж.

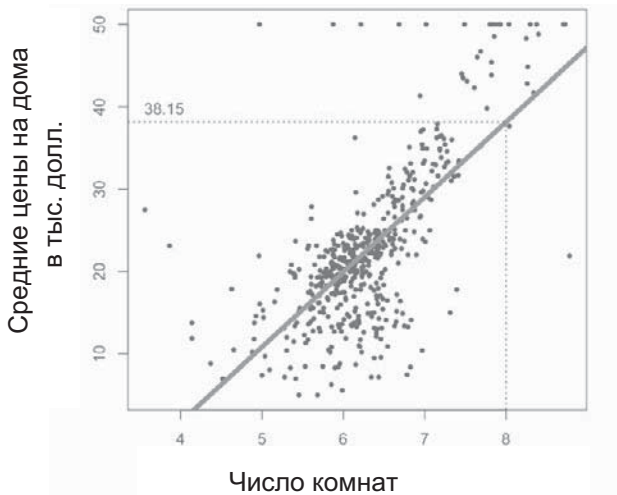
Это становится возможным с *регрессионным анализом*, позволяющим не только улучшать прогнозирование путем учета множества предикторов, но и сравнивать эти предикторы между собой по степени влияния.

Чтобы разобраться с этим, посмотрим на пример с предсказанием цен на дома.

## 6.2. Пример: предсказание цен на дома

Мы использовали данные за 1970-е годы о ценах на дома в Бостоне. Предварительный анализ показывает, что двумя сильнейшими предикторами цен на дома являются число комнат в доме и доля соседей с низким доходом.

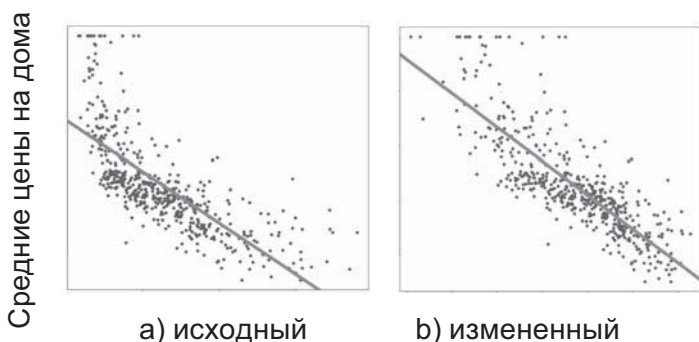
На рис. 1 видно, что у дорогих домов обычно больше комнат. Для предсказания цены дома можно построить линию тренда, известную также как *линия наилучшего соответствия*. Она проходит близко к наибольшему числу элементов данных на графике. Например, если у дома восемь комнат, его цена составит приблизительно \$ 38 150.



**Рис. 1.** Цены на дома в сравнении с числом комнат

Кроме числа комнат на цену дома также влияло его окружение. Дома оказались дешевле там, где была выше пропорция соседей с низким доходом (рис. 2). Поскольку тренд получался немного изогнутым (рис. 2, а), мы применили к предикторам математическую операцию, известную как взятие логарифма. Благодаря этому через элементы данных проще провести прямую линию тренда (рис. 2, б).

Можно заметить, что элементы данных на рис. 2, б плотнее прилегают к линии тренда, чем на рис. 1. Это означает, что фактор соседства оказался более точным предиктором цены дома, чем число комнат.

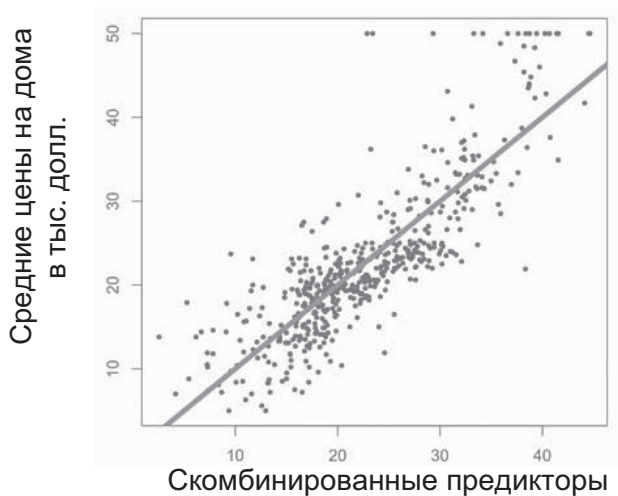


**Рис. 2.** Цены на дома в сравнении с долей соседей с низким доходом

Для улучшения наших расчетов цен на дома мы можем учесть и число комнат, и влияние соседства. Но поскольку выяснилось, что влияние соседства лучше предсказывает цену дома, простое сложение этих двух предикторов



не станет идеальным решением. Вместо этого предиктору соседства нужно задать больший вес.



**Рис. 3.** Цены на дома в сравнении со скомбинированным предиктором из числа комнат и доли соседей с низким доходом

Рис. 3 показывает график цен на дома согласно оптимальной комбинации двух предикторов. Обратите внимание на то, что элементы данных располагаются еще ближе к итоговой линии тренда, чем раньше, поэтому прогноз с использованием такой линии тренда должен оказаться точнее. Чтобы проверить это, можно сравнить погрешность трех линий тренда (табл. 1).

Хотя очевидно, что уравновешенная комбинация предикторов ведет к более точным предсказаниям, возникают два вопроса:

- 1) как вычислить оптимальный вес предикторов;
- 2) как следует их проинтерпретировать.

**Таблица 1.** Средняя прогностическая ошибка при использовании трех разных линий тренда

	Погрешность прогнозирования (в тыс. долл.)
Число комнат	4,4
Влияние окружения	3,9
Число комнат и влияние окружения	3,7

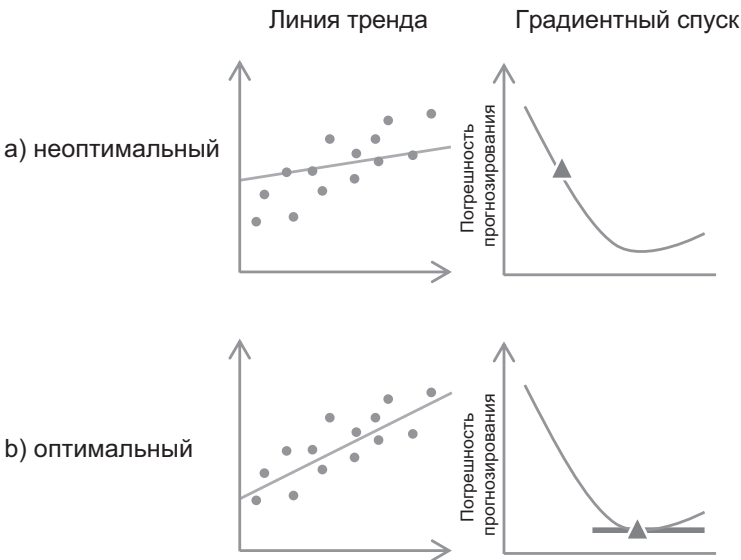
### 6.3. Градиентный спуск

Вес предиктора — главный параметр регрессионного анализа, и оптимальный вес обычно вычисляется путем решения уравнений. Тем не менее, поскольку регрессионный анализ прост и годится для визуализации, мы воспользуемся им для демонстрации альтернативного способа оптимизации параметров. Этот метод называется градиентным спуском (*gradient descent*) и используется в случаях, когда параметры нельзя получить напрямую.

Вкратце: алгоритм *градиентного спуска* делает первоначальное предположение о наборе весовых составляющих, после чего начинается итеративный процесс их применения к каждому элементу данных для прогнозирования,

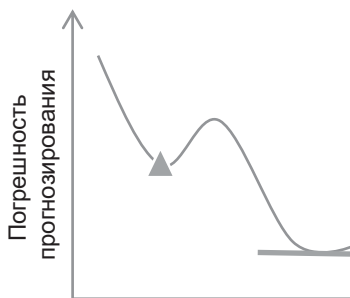
а затем они перенастраиваются для снижения общей ошибки прогнозирования.

Этот процесс можно сравнивать с пошаговым спуском в овраг в поисках дна. На каждом этапе алгоритм определяет, какое направление даст наиболее крутой спуск, и пересчитывает весовые составляющие. В конечном итоге мы достигнем самой нижней позиции, которая представляет собой точку, в которой погрешность прогнозирования минимальна. Рисунок 4 показывает, как оптимальная линия тренда регрессии соответствует нижней точки градиента.



**Рис. 4.** Как линия тренда достигает оптимальности благодаря градиентному спуску

Кроме регрессии градиентный спуск может также использоваться для оптимизации параметров в других моделях, таких как метод опорных векторов (см. главу 8) или в нейронных сетях (см. главу 11). Однако в этих более сложных моделях результаты градиентного спуска могут зависеть от стартовой позиции в овраге (то есть изначальных значений параметра). Например, если нам случится начать в небольшой яме, алгоритм градиентного спуска может ошибочно принять это за оптимальную точку (рис. 5).



**Рис. 5.** Как ближайшая яма может быть ошибочно принята за оптимальную точку (треугольник), хотя истинная оптимальная точка находится ниже ее (черта)

Чтобы снизить риск попадания в такую яму, мы можем воспользоваться *стохастическим градиентным спуском*, при котором вместо использования *всех* элементов данных для регулирования параметров при каждой итерации берется только *один*. Это привносит вариативность, позволяя алгоритму избегать ям. Хотя итоговые значения

параметров после работы стохастического процесса могут оказаться не оптимальными, они, как правило, обеспечивают достаточно высокую точность.

Тем не менее этот «недостаток» относится только к более сложным моделям, и нам не о чем беспокоиться, когда мы используем регрессионный анализ.

## 6.4. Коэффициенты регрессии

После получения оптимального набора регрессионных предикторов их нужно интерпретировать.

Вес регрессионных предикторов называется *коэффициентом регрессии*. Коэффициент регрессии показывает то, *насколько силен предиктор при совместном использовании с другими*. Иными словами, это *значение, добавляемое к предиктору*, а не его собственная предсказательная способность.

Например, если кроме числа комнат использовать для предсказания цены дома его общую площадь, то значимость числа комнат может показаться незначительной. Поскольку и число комнат, и общая площадь дома связаны с его размером, это добавляет к предсказательной силе не так уж и много.

Толковой интерпретации регрессионных коэффициентов мешает также различие в единицах измерения. Например, если предиктор измеряется в сантиметрах, его вес

будет в 100 раз отличаться по весу от предиктора, берущегося в метрах. Чтобы избежать такого, мы должны *стандартизировать* единицы измерения предикторных переменных перед тем, как проводить регрессионный анализ. Стандартизация — это выражение переменных в процентилях. Когда предикторы стандартизированы, то коэффициент, который называется *бета-весом*, может быть использован для более точных сравнений.

В примере с ценами на дома два предиктора (первый — число комнат, второй — соседи с низким доходом) были стандартизированы в соотношении 2,7 к 6,3. Это означает, что доля жильцов с низким доходом является более мощным предиктором цены на дом, чем количество комнат.

Уравнение регрессии будет выглядеть примерно так:

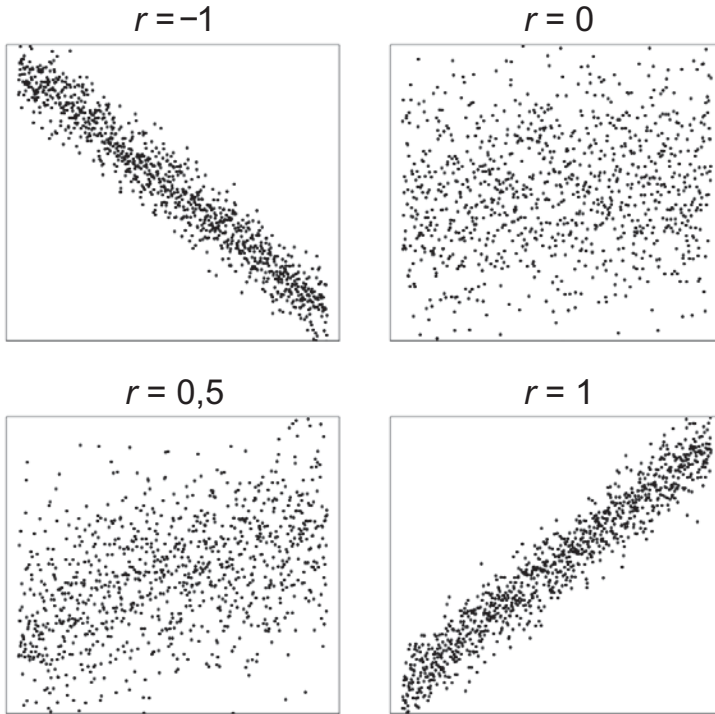
цена = 2,7 (количество комнат) – 6,3(% соседей с низким доходом) .

Обратите внимание, что в этом уравнении доля жильцов с низким доходом имеет отрицательный вес, что выражено знаком «минус». Дело в том, что предиктор имеет обратную корреляцию с ценами на дома, как показано на устремленной вниз линии тренда на рис. 2.

## 6.5. Коэффициенты корреляции

Если предиктор только один, бета-вес такого предиктора называется *коэффициентом корреляции* и обозначается

как  $r$ . Коэффициенты корреляции варьируются от  $-1$  до  $1$  и несут две единицы информации.



**Рис. 6.** Пример распределения данных в соответствии с различными коэффициентами корреляции

**Направление.** При положительных коэффициентах предиктор стремится в том же направлении, что и результат. При отрицательных — в обратном направлении. Цены

домов положительно коррелируют с числом комнат, но отрицательно коррелируют с долей жильцов с низким доходом по соседству.

**Величина.** Чем ближе коэффициент к  $-1$  или  $1$ , тем сильнее предиктор. Например, коэффициент корреляции, показанный линией тренда на рис. 1, равен  $0,7$ , в то время как на рис. 2,  $b$  это  $-0,8$ . Это означает, что достаток соседей — более достоверный предиктор цен на дома, чем число комнат. Нулевая корреляция означала бы отсутствие связи между предиктором и результатом. Коэффициенты корреляции показывают абсолютную силу отдельных предикторов и, следовательно, являются более надежным способом их ранжирования, чем коэффициенты регрессии.

## 6.6. Ограничения

Несмотря на то что регрессионный анализ информативен и не требует долгих вычислений, он имеет недостатки.

**Чувствительность к резко отклоняющимся значениям.** Регрессионный анализ одинаково учитывает все представленные элементы данных. Если среди них будет хотя бы несколько элементов с крайними значениями, это может значительно исказить линию тренда. Чтобы избежать этого, можно использовать диаграмму рассеяния для предварительного выявления таких резко отклоняющихся значений.



**Искажение веса при корреляции предикторов.** Включение в регрессионную модель высокоррелирующих предикторов исказит интерпретацию их веса. Эта проблема называется *мультиколлинеарностью*. Для преодоления мультиколлинеарности нужно либо исключить из анализа коррелирующие предикторы, либо воспользоваться более продвинутым методом, таким как *лассо* или *ридж-регрессия* (или гребневая регрессия).

**Криволинейные тренды.** В нашем примере тренды отображались прямой линией. Тем не менее некоторые тренды могут быть криволинейными, как на рис. 2, а. В этом случае нам потребуется преобразовать значения предикторов или использовать альтернативные алгоритмы, такие как метод опорных векторов (см. главу 8).

**Корреляция не говорит о причинности.** Предположим, была обнаружена положительная корреляция между стоимостью дома и наличием собаки. Понятно, что если просто завести собаку, цена дома от этого не изменится, однако можно предположить, что те, кто могут позволить себе содержать собак, располагают в среднем бóльшим доходом и, вероятно, проживают в районах, где дома стоят дороже.

Несмотря на эти ограничения, регрессионный анализ остается одним из основных, простых в использовании и интуитивно-понятных методов для прогнозирования. Внимательное отношение к способу интерпретации результатов — залог уверенности в точности выводов.

## 6.7. Краткие итоги

- Регрессионный анализ находит линию наилучшего соответствия, тяготеющую к максимально возможному числу элементов данных.
- Линия тренда выводится на основании уравновешенной комбинации предикторов. Вес предиктора называется *коэффициентом регрессии*. Он показывает силу одного предиктора в присутствии других.
- Регрессионный анализ хорошо работает в условиях низкой корреляции между предикторами, отсутствия резко отклоняющихся значений и там, где линия тренда ожидается в виде прямой линии.

# 7

## **Метод k-ближайших соседей и обнаружение аномалий**

## **7.1. Пищевая экспертиза**

Давайте поговорим о вине. Вы когда-нибудь задумывались, в чем различие между красным и белым вином?

Кто-то может считать, что красное вино попросту делают из красного винограда, а белое из белого. Но это не совсем так, поскольку белое вино может быть получено и из красного винограда, хотя из белого винограда красного вина не сделать.

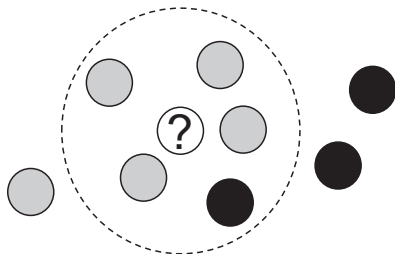
Ключевое отличие состоит в том, при каких условиях виноград подвергается брожению. В случае с красным вином виноградный сок бродит вместе с кожицей, которая выделяет характерный красный пигмент, чего не происходит с белым.

Нетрудно понять, использовалась ли кожица при изготовлении вина, просто взглянув на него, но можно сделать это и не глядя. Дело в том, что кожица значительно меняет химический состав вина, поэтому, располагая такими сведениями, цвет можно вычислить.

Чтобы проверить это предположение, можно воспользоваться одним из простейших алгоритмов машинного обучения: методом  $k$ -ближайших соседей.

## 7.2. Яблоко от яблони недалеко падает

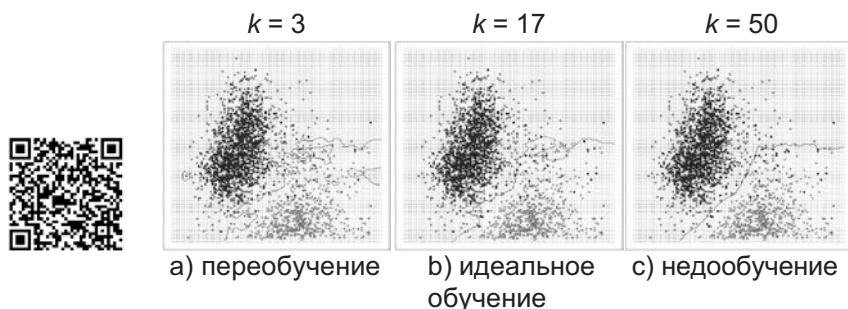
Метод  $k$ -ближайших соседей ( $k$ -Nearest Neighbors) — это алгоритм, который классифицирует элементы данных, исходя из класса соседних. Это означает, что если элемент данных окружен четырьмя серыми точками и одной черной, то, исходя из этого, он, вероятнее всего, серый.



**Рис. 1.** Элемент данных в середине будет сочтен серым, поскольку именно такой цвет преобладает среди его ближайших пяти соседей

В названии метода параметр  $k$  означает количество ближайших соседей, которое нужно учитывать в расчетах. В приведенном примере  $k$  равно пяти. Выбор правильно-

го значения  $k$  является примером настройки параметра (раздел 1.3) и критически важен для точности прогнозирования.



**Рис. 2.** Сравнение моделей настройки при различных значениях  $k$ . Предполагается, что точки в черной зоне должны соответствовать белым винам, а в серой — красным

Если значение  $k$  слишком мало (рис. 2, а), то элементы данных совпадут только для непосредственных соседей, и погрешности, вызванные случайным шумом, усилятся. Если значение  $k$  слишком велико (рис. 2, в), то элементы данных будут классифицироваться слишком неточно, а выявленные закономерности окажутся размытыми. Но когда значение  $k$  выбрано удачно (рис. 2, б) то погрешности в классификации элементов данных взаимопогашаются, выявляя тонкие тренды среди имеющихся данных.

Для достижения наилучшей настройки параметр  $k$  может быть вычислен путем кросс-валидации (раздел 1.4).

В случае с бинарной (двухклассовой) задачей классификации можно избежать проблемы равновероятности распределения, задав для  $k$  нечетное значение.

Вместо классификации элементов данных в группы метод  $k$ -ближайших соседей может также использоваться для прогнозирования непрерывных значений путем агрегирования соседних значений. Помимо того, чтобы рассматривать всех соседей как равноценных, можно улучшить оценку, используя весовой параметр. Значения ближайших соседей могут точнее отражать истинное значение элемента данных, чем отдаленных, поэтому иногда на них стоит ориентироваться в большей степени.

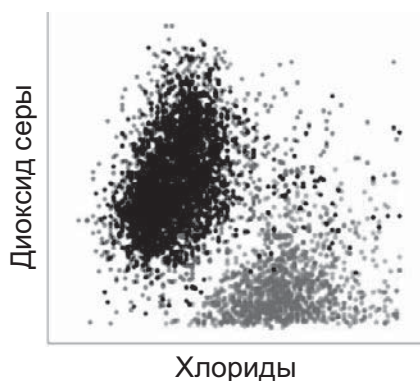
## 7.3. Пример: истинные различия в вине

Вернемся к примеру с вином. Цвет конкретного вина можно вывести из цвета других вин со схожим химическим составом.

Воспользовавшись данными по белым и красным вариантам португальского *винью-верде* («зеленого вина»), мы построили график химического состава 1599 красных и 4898 белых вин, взяв за оси два ингредиента: хлориды и диоксид серы.

Поскольку виноградная кожица содержит более высокую концентрацию таких элементов, как хлорид натрия

(известный как поваренная соль), в красных винах его содержание выше. Кроме того, кожа содержит природные антиоксиданты, препятствующие порче ягод. Из-за их отсутствия белое вино требует большего количества диоксида серы, выступающего в роли консерванта. В силу этих причин красные вина расположились на графике с рис. 3 снизу справа, а белые — сверху слева.



**Рис. 3.** Уровень содержания хлоридов и диоксида серы в белых винах (черным цветом) и красных (серым цветом)

Для определения цвета вина исходя из соответствующих уровней содержания хлоридов и диоксида серы, можно руководствоваться цветом соседних вин, то есть тех, которые обладают похожим содержанием обоих химических компонентов. Сделав это для каждой точки графика, мы получаем границы, отличающие красные вина от белых (см. рис. 2). В случае идеального обучения (см. рис. 2, b), можно предсказать цвет вина с точностью до 98 %.



## 7.4. Обнаружение аномалий

Применимость метода  $k$ -ближайших соседей не ограничивается предсказанием групп или значений элементов данных. Он также может быть использован для обнаружения таких аномалий, как выявление подлогов. Более того, обнаружение аномалий может привести к ценному открытию: нахождению предиктора, который раньше не был замечен.

Обнаружение аномалий становится значительно проще, если данные могут быть визуализированы. Например, на рис. 3 можно сразу увидеть, какие вина сильно отклоняются от кластеров. Однако не всегда возможно визуализировать данные на двумерном графике, особенно в случаях, когда для анализа есть больше двух предикторных переменных. Здесь и помогут такие модели, как метод  $k$ -ближайших соседей.

Поскольку он использует для прогнозирования закономерности среди данных, погрешности прогнозирования служат явным указанием на элементы данных, не укладывающиеся в основные тренды. На самом деле любой алгоритм, строящий прогностическую модель, может быть использован для поиска аномалий. Так, при регрессионном анализе (глава 6) аномальная точка может быть легко найдена, потому что она значительно отклоняется от линии наилучшего соответствия.

Если посмотреть на аномалии в примере с винами (то есть на ошибочные классификации), мы обнаружим,

что красные вина неверно определяются как белые из-за необычно высокого содержания диоксида серы. Если нам известно, что данные вина требуют большего содержания этого консерванта из-за низкого уровня кислотности, то мы можем принять во внимание кислотность вина для улучшения прогнозирования.

Аномалии могут быть вызваны пропущенными предикторами, иногда их причиной является недостаток данных для обучения модели. Чем меньше элементов данных у нас есть, тем сложнее распознать закономерности в данных, из-за чего очень важно убедиться в том, что их объем соответствует задачам моделирования.

Как только аномалии определены, они могут быть удалены из набора данных перед обучением прогностической модели. Это снизит уровень шума в данных и увеличит точность прогнозирования.

## 7.5. Ограничения

Хотя метод  $k$ -ближайших соседей прост и эффективен, нужно учесть, что для некоторых случаев он может оказаться не самым удачным выбором.

**Не классы.** Если имеется множество классов и эти классы существенно отличаются по размеру, то элементы данных, принадлежащие к самому небольшому из них, могут быть ошибочно включены в более крупные. Чтобы улучшить точность, можно и здесь использовать вместо равновесного вычисления весовые параметры, которые

позволят больше ориентироваться на ближайшие элементы данных, а не на отдаленные.

**Избыток предикторов.** Если предикторов слишком много, для определения ближайших соседей в нескольких измерениях могут потребоваться долгие вычисления. Более того, некоторые предикторы могут быть лишними и не улучшать точность прогноза. Чтобы исключить это, для выявления наиболее существенных предикторов для анализа можно воспользоваться уменьшением размерности (см. главу 3).

## 7.6. Краткие итоги

- Метод  $k$ -ближайших соседей представляет собой метод классификации элементов данных путем их сопоставления с ближайшими элементами.
- $k$  — число таких ближайших элементов для расчета, которое определяется с помощью *кросс-валидации*.
- Лучше всего он работает при условиях, когда предикторов немного, а классы примерно одного размера. Неточные классификации могут служить верным признаком возможных аномалий.



# 8

## **Метод опорных векторов**

## 8.1 «Нет» или «о, нет!»?

Медицинский диагноз — сложная задача. Симптомов, которые необходимо принять во внимание, может быть много, а сам процесс не исключает влияния субъективного мнения врачей. Иногда правильный диагноз ставится лишь тогда, когда уже слишком поздно. Системный подход для точного прогноза в сфере диагностики заболеваний заключается в использовании алгоритмов, обученных на медицинских базах данных.

В этой главе мы рассмотрим способ прогнозирования, известный как *метод опорных векторов* (support vector machine). Этот метод выявляет оптимальную границу для классификации, которая может быть использована для разделения пациентов на две группы (то есть здоровых и нездоровых).

## 8.2. Пример: обнаружение сердечно-сосудистых заболеваний

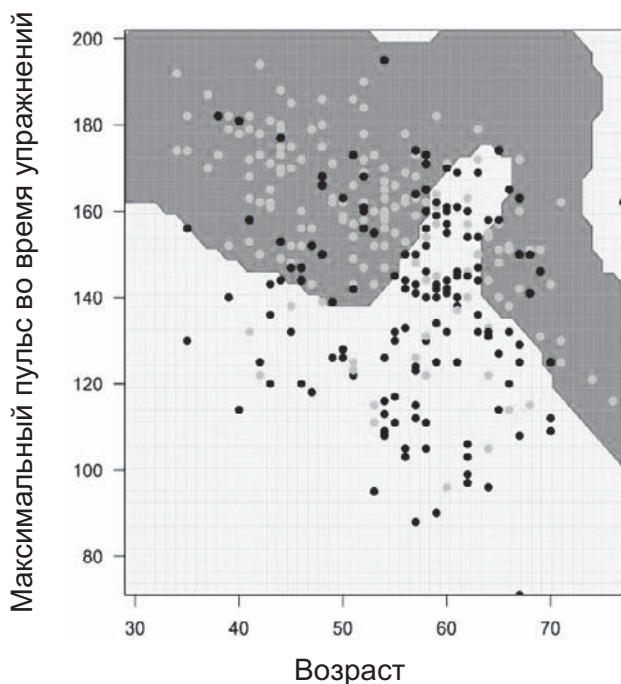
Сердечно-сосудистые заболевания (ССЗ) — одни из самых распространенных в развитых странах. При ССЗ сужение

и закупорка кровеносных сосудов увеличивают риск инфаркта. Заболевание может быть окончательно диагностировано посредством томографии, но ее стоимость не позволяет людям обследоваться регулярно. Альтернативным решением может стать выявление на основе физиологических симптомов пациентов с высокой долей риска, которые более всего нуждаются в таком обследовании.

Для определения того, какие симптомы предшествуют ССЗ, пациентов американской клиники попросили делать упражнения, а затем регистрировали их физическое состояние. Среди учитываемых показателей был и максимальный пульс во время занятий. Вслед за этим для проверки наличия заболеваний использовалась томография. Была построена модель с использованием метода опорных векторов, учитывающая данные о пульсе и возрасте пациентов (рис. 1). С помощью нее удастся с 75 %-ной вероятностью предсказать, если кто-то страдает от ССЗ.

В основном пациенты с ССЗ (черные точки) имели невысокий пульс во время упражнений по сравнению со здоровыми (светлые точки) того же возраста. Заболевания оказались более распространены среди пациентов старше 55 лет.

Хотя пульс обычно снижается с возрастом, пациенты с ССЗ, которым около 60 лет, продемонстрировали более высокий пульс по сравнению со здоровыми молодыми людьми, что показано в виде неожиданной дуги на разделяющей границе. Если бы не способность метода опорных векторов находить криволинейные паттерны, мы могли бы упустить из виду это явление.



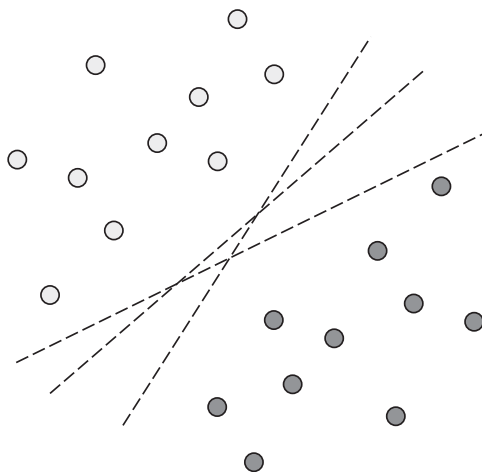
**Рис. 1.** Использование метода опорных векторов для обнаружения сердечно-сосудистых заболеваний. Темная область соответствует здоровым пациентам, а светлая — больным. Светлые и черные точки представляют собой здоровых и нездоровых пациентов соответственно

### 8.3. Построение оптимальной границы

Главная задача метода опорных векторов — построение оптимальной границы, которая отделяет одну группу от



другой. Это не так просто, как кажется, поскольку возможных вариантов очень много (рис. 2).

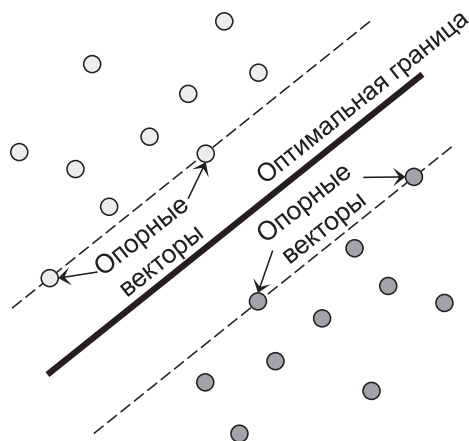


**Рис. 2.** Множество способов разделить две группы

Чтобы найти оптимальную линию разграничения, нужно сначала найти периферийные элементы данных, которые находятся ближе всего к противоположной группе. Оптимальная граница проводится посередине между такими периферийными элементами данных в обеих группах (рис. 3). Поскольку эти элементы данных помогают обнаружить оптимальную линию разграничения, их называют *опорными векторами*.

Одно из преимуществ метода — скорость вычисления. Поскольку линия разграничения определяется только по периферийным элементам данных, для ее получения требуется меньше времени, чем для методов по типу ре-

грессии (глава 6), которые выстраивают линию тренда с учетом всех элементов.



**Рис. 3.** Оптимальная граница находится посередине между периферийными элементами данных из разных противоположных групп

Тем не менее эта манера опираться на отдельные элементы данных имеет оборотную сторону. Разделительная граница становится чувствительнее к положению опорных векторов, а значит, слишком зависит от набора данных, использованного для обучения модели. Более того, элементы данных редко делятся так ровно, как показано на рис. 2 и 3. В реальности они часто перекрываются, как на рис. 1.

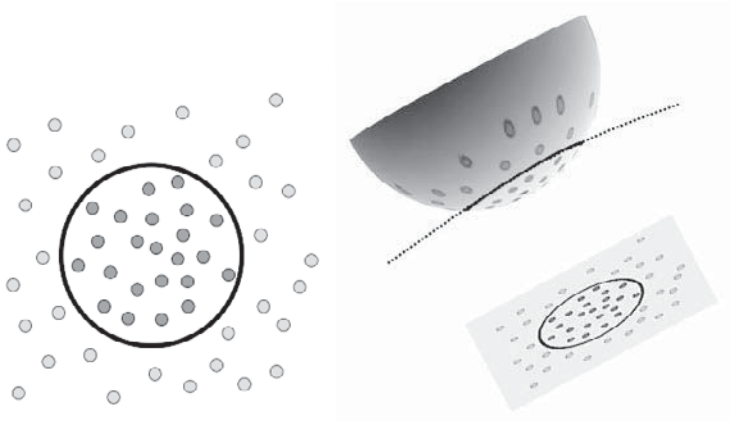
Чтобы справиться с этими проблемами, у метода опорных векторов есть такая ключевая особенность, как

промежуточная область, которая позволяет ограниченному числу элементов данных оказываться по другую сторону границы. В результате получается более «мягкая» граница, которая лучше справляется с резко отклоняющимися значениями и делает модель более масштабируемой.

Промежуточная область задается настройкой *параметр стоимости* (cost parameter), который задает допустимую степень погрешностей классификации. Чем выше параметр стоимости, тем больше допустимый уровень ошибок и тем шире промежуточная область. Чтобы итоговая модель давала точный прогноз как для текущих, так и для новых данных, лучшее значение параметра стоимости можно найти путем кросс-валидации (раздел 1.4).

Существенное достоинство метода опорных векторов состоит в его способности обнаруживать в данных криволинейные паттерны. Хотя на это способны и другие алгоритмы, метод опорных векторов предпочитают из-за сочетания превосходной вычислительной эффективности и умения находить замысловатые криволинейные паттерны с помощью *функции ядра* (kernel trick).

Вместо того чтобы сразу прочерчивать границу на плоскости данных, метод опорных векторов сначала проецирует их на дополнительное измерение, которое может быть отделено прямой линией (рис. 4). Эти прямые линии легче как вычислять, так и преобразовывать в кривые при возврате к изначальной размерности.



**Рис. 4.** Круг темно-серых точек на двумерном листе может быть отображен прямой линией при проекции в виде трехмерной сферы

Способность метода опорных векторов работать с несколькими измерениями обеспечивает его популярность в анализе наборов данных со множеством переменных. Его нередко применяют для расшифровки генетической информации и анализа тональности текста.

## 8.4. Ограничения

Хотя метод опорных векторов является адаптивным и быстрым инструментом, он может не подходить в следующих случаях.

**Малые наборы данных.** Поскольку для определения границ метод опирается на опорные векторы, то небольшой набор данных сокращает их число и отрицательно влияет на точность расчета.

**Множество групп.** Метод опорных векторов способен классифицировать данные только на две группы за раз. Если групп три и более, то необходимо применять итеративно для выявления каждой отдельной группы метод, который называется *многоклассовая классификация* (multi-class SVM).

**Большое перекрытие данных.** Метод опорных векторов классифицирует элементы данных исходя из того, с какой стороны границы разграничения они оказались. Когда элементы данных сильно перекрываются обеими группами, то те из них, которые находятся ближе к границе, могут быть классифицированы ошибочно. Более того, метод не дает информации о вероятности ошибочной классификации для отдельного элемента данных. Тем не менее для оценки точности классификации отдельного элемента можно ориентироваться на расстояние от него до границы разделения.

## 8.5. Краткие итоги

- Метод опорных векторов классифицирует элементы данных на две группы, проводя границу между пери-

ферийными элементами данных (то есть *опорными векторами*) обеих групп.

- Он устойчив к резко отклоняющимся значениям и использует *промежуточную область*, которая позволяет некоторым элементам данных находиться по ту сторону границы разделения. Метод также использует *функцию ядра* для точного получения изогнутых границ.
- Он лучше всего работает с большими наборами данных, которые нужно классифицировать всего по двум группам.

# 9

## **Дерево решений**

## 9.1. Прогноз выживания в катастрофе

Во время катастроф некоторые группы людей, такие как женщины и дети, могут первыми получить помощь, что значительно увеличивает их шансы на спасение. В таких ситуациях мы можем использовать *дерево решений*, чтобы вычислить, какие группы выживут.

Дерево решений предсказывает вероятность выживания исходя из серии бинарных вопросов (рис. 1), на каждый из которых можно ответить только «да» или «нет». Мы начинаем с верхнего вопроса, известного как корень, и движемся по ветвям дерева исходя из ответов, до тех пор пока не достигаем последнего листа, который показывает шансы на выживание.

## 9.2. Пример: спасение с тонущего «Титаника»

Чтобы продемонстрировать работу дерева решений для оценки выживаемости групп пассажиров, мы воспользо-



вались данными о печально известном лайнере «Титанике», собранными британским торговым министерством. Рисунок 2 показывает дерево решений, которое оценивает шансы пассажиров на выживание.

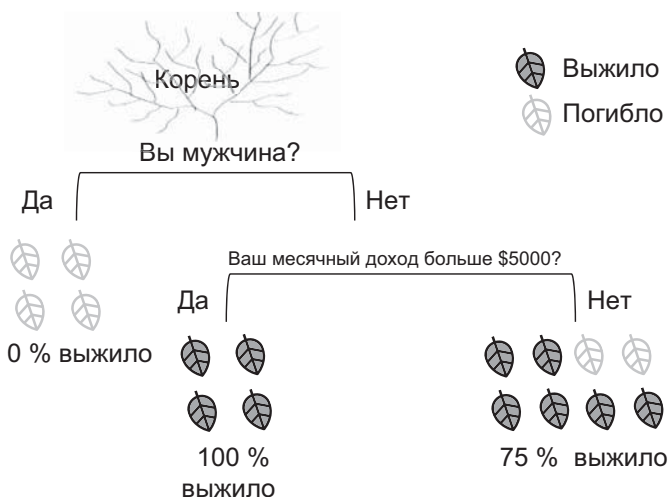


Рис. 1. Пример дерева решений

Можно заметить, что хорошие шансы спастись с «Титаника» были у несовершеннолетних мужчин и у женщин, которые не были пассажирами 3-го класса.

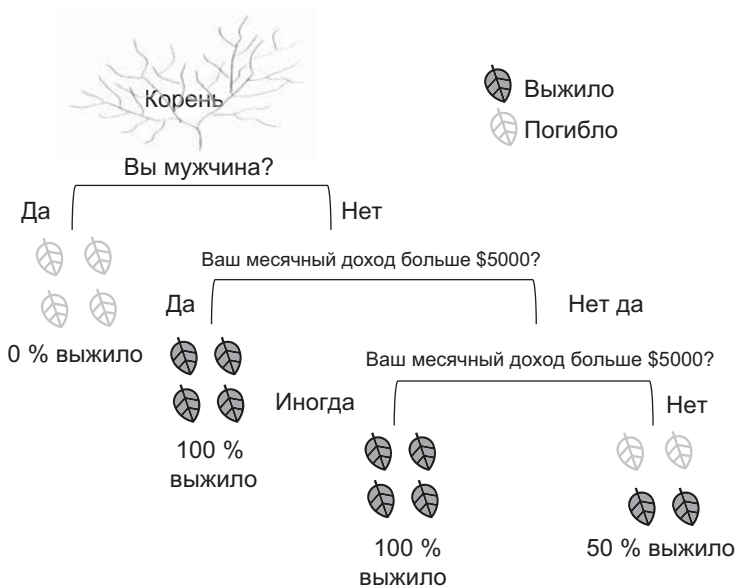
Деревья решений невероятно гибкие и имеют множество применений. Среди них вычисление шансов на выживание при медицинском диагнозе, расчет вероятности увольнения персонала и обнаружение мошеннических транзакций. Деревья решений могут также использо-

ваться и для категориальных переменных (например, мужчины и женщины) или непрерывных (уровень дохода). Обратите внимание, что группами могут быть представлены и непрерывные значения. Если сравнить, например, каждое значение со средним, то оно будет больше или меньше.



**Рис. 2.** Дерево решений, предсказывающее, выжил ли пассажир тонущего «Титаника»

В обычных деревьях решений есть только два возможных ответа на каждом ветвлении: «да» или «нет». Если нужно учесть три и более варианта ответа («да», «нет» и «иногда»), то можно просто добавить больше ветвлений (рис. 3).



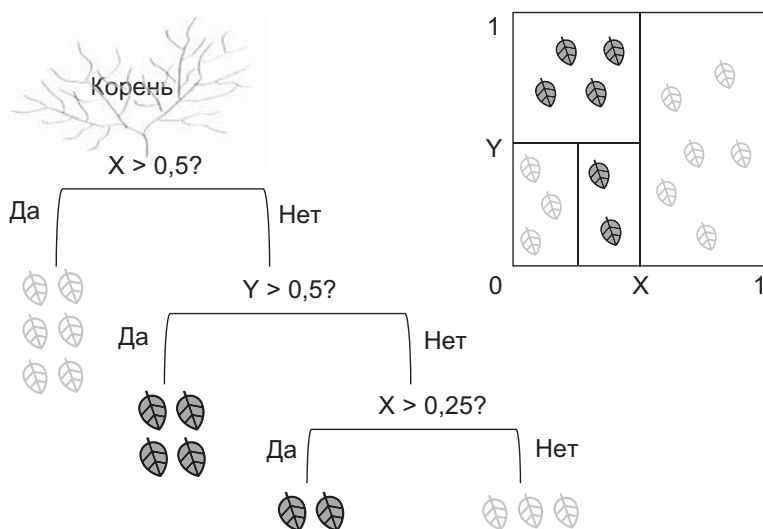
**Рис. 3.** Множественные категории в дереве решений

Деревья решений популярны, поскольку их результат легко интерпретировать. Вопрос только в том, как их создать.

## 9.3. Создание дерева решений

Дерево решений вырастает из деления элементов данных на две группы так, чтобы похожие элементы оказались вместе. Далее этот процесс продолжается для каждой группы.

В результате в каждом следующем листе оказывается меньше элементов данных, но они более однородны. В основе дерева решений лежит идея о том, что элементы данных, проходящие один путь, ближе друг к другу по значению.



**Рис. 4.** Разделение элементов данных на дереве решений и визуализация в виде диаграммы рассеяния

Повторяющийся процесс разбития данных для получения однородных групп называется *рекурсивным делением* (recursive partitioning). Он включает два шага.

**Шаг 1:** найти бинарный вопрос, которым лучше всего разделить элементы данных на две внутренние однородных группы.

**Шаг 2:** повторять шаг 1 для каждого листа, пока критерий остановки не будет достигнут.

Есть много вариантов критерия остановки, выбор среди которых можно сделать при помощи кросс-валидации (см. раздел 1.4). Возможные варианты:

- остановиться, когда элементы данных на каждом листе относятся к одной категории или содержат одно значение;
- остановиться, когда на листе осталось менее пяти элементов данных;
- остановиться, когда дальнейшее ветвление не улучшает однородность на минимальный заданный порог.

Поскольку рекурсивное деление использует только лучшие бинарные вопросы для создания дерева решений, присутствие недостоверных переменных не повлияет на результаты. Более того, бинарные вопросы тяготеют к тому, чтобы разделять элементы данных по средним показателям, поэтому деревья решений устойчивы к резко отклоняющимся значениям.

## 9.4. Ограничения

Несмотря на легкость интерпретации, деревья решений тоже имеют свои недостатки.

**Нестабильность.** Поскольку деревья решений строятся путем деления элементов данных на однородные группы, небольшое изменение в этих данных способно

повлиять на то, как будет выглядеть все дерево. Поскольку деревья решений стремятся к наилучшему способу разделения элементов данных, они восприимчивы к переобучению (раздел 1.3).

**Неточность.** Использование наилучшего бинарного вопроса для разбивки данных не всегда ведет к точным предсказаниям. Иногда для лучшего прогнозирования нужны менее эффективные первоначальные разделения.

Чтобы обойти эти ограничения, можно избежать ориентации на лучшую разбивку данных и использовать различные варианты деревьев решений совместно. То есть мы можем получить более точные и постоянные результаты путем комбинирования прогнозов, полученных от различных деревьев.

Есть два способа сделать это.

- При первом способе сначала различные комбинации бинарных вопросов для создания деревьев выбираются случайным образом, а затем полученные предсказания суммируются. Этот метод известен как построение *случайного леса* (глава 10).
- Вместо того чтобы брать случайные бинарные вопросы, при втором способе они выбираются стратегически, вследствие чего точность прогнозирования последовательно улучшается. Результатом становится взвешенное среднее значение, полученное при помощи всех деревьев решений. Этот метод называется *градиентным бустингом* (gradient boosting).

Хотя случайные леса и градиентный бустинг позволяют делать более точные прогнозы, их сложность мешает визуализации, в связи с этим их прозвали *черными ящиками*. Это объясняет, почему популярным инструментом анализа продолжают оставаться обычные деревья решений. Их наглядность упрощает оценку предикторов и их взаимодействия.

## 9.5. Краткие итоги

- Дерево решений создает прогноз на основании серии бинарных вопросов.
- Набор данных последовательно разбивается на более однородные группы в ходе процесса, названного *рекурсивным делением*. Он продолжается до наступления критерия остановки.
- Несмотря на то что деревья решений понятны и просты в использовании, они подвержены *переобучению*, которое может привести к противоречивым результатам. Чтобы это минимизировать, используют альтернативные методы, такие как *случайные леса*.





# 10

**Случайные леса**

## 10.1. Мудрость толпы

Можно ли из множества неверных ответов получить правильный?

Да!

Хотя это кажется нелогичным, такое возможно. Более того, это ожидается от самых лучших прогностических моделей.

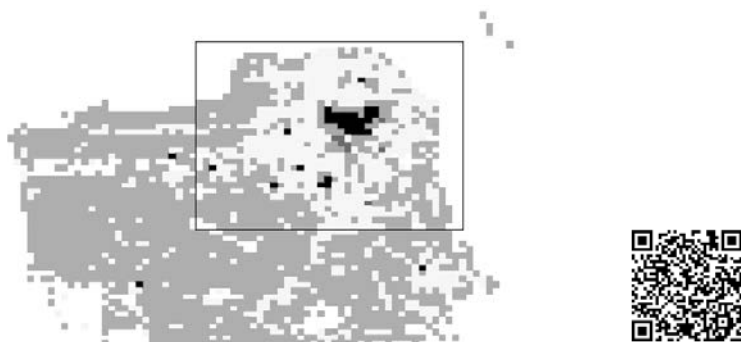
Хитрость в том, что хотя неверных прогнозов может быть много, правильный всегда только один. При комбинировании моделей с различными достоинствами и недостатками оказывается, что точные прогнозы имеют тенденцию подтверждать друг друга, в то время как ошибочные этого не делают. Способ комбинирования моделей для улучшения точности прогноза известен как *ансамблирование* (ensembling).

Мы обнаружим этот эффект в результатах работы *случайного леса*, который представляет собой ансамбль деревьев решений (глава 9). Чтобы показать, насколько случайный лес превосходит деревья решений, мы создали 1000 возможных деревьев, каждое из которых предсказывает пре-

ступление в американском городе, после чего сравнили точность их прогнозирования с точностью случайного леса, построенного на основе тех же 1000 деревьев решений.

## 10.2. Пример: предсказание криминальной активности

Открытая сводка от *полицейского управления Сан-Франциско* предоставляет информацию о месте, времени и тяжести преступлений, совершенных в городе с 2014 по 2016 год. Поскольку анализ показывает, что в жаркие дни уровень преступности обычно растет, мы также взяли метеорологические данные по дневной температуре и осадкам за тот же период (рис. 1).



**Рис. 1.** Тепловая карта Сан-Франциско, которая показывает частоту преступлений: очень низкую (серым), низкую (светлым), среднюю (темно-серым) и высокую (черным)

Мы предположили, что с учетом кадровых и ресурсных возможностей полиция сможет организовать дополнительные патрули в тех местах, где ожидаются преступления. Поэтому мы запрограммировали прогностическую модель находить только 30 % территорий с наиболее высокой вероятностью совершения преступлений с применением насилия.

Предварительный анализ показал, что преступления совершались в основном в северо-восточной части города (выделена прямоугольником). Для дальнейшего анализа мы разделили эту зону на небольшие участки размером  $260 \times 220$  м.

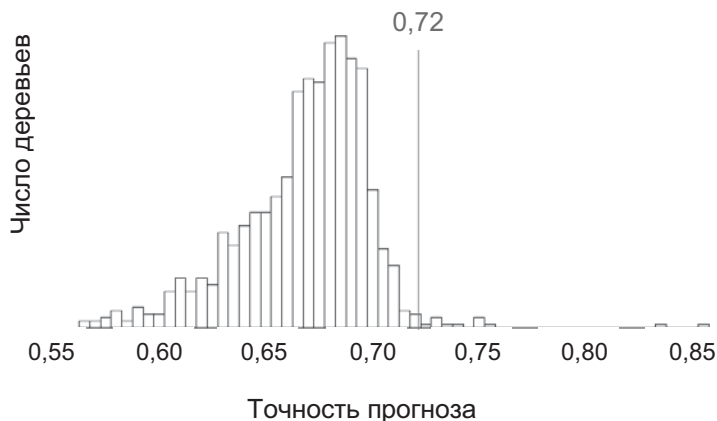
Для предсказания того, где и когда могут случиться преступления, были созданы 1000 возможных деревьев решений, которые учитывали данные по преступности и погоде. После этого мы построили на их основе случайный лес. Мы использовали данные за 2014 и 2015 годы для обучения прогностических моделей, после чего проверяли их точность на данных 2016 года (с января по август).

Так насколько хорошо мы можем предвидеть преступления?

Случайный лес успешно предсказал 72 % (почти три четверти) всех преступлений с применением насилия. Это доказывает превосходство точности его прогноза по сравнению со средней точностью составляющих его деревьев решений, которая равна 67 % (рис. 2).

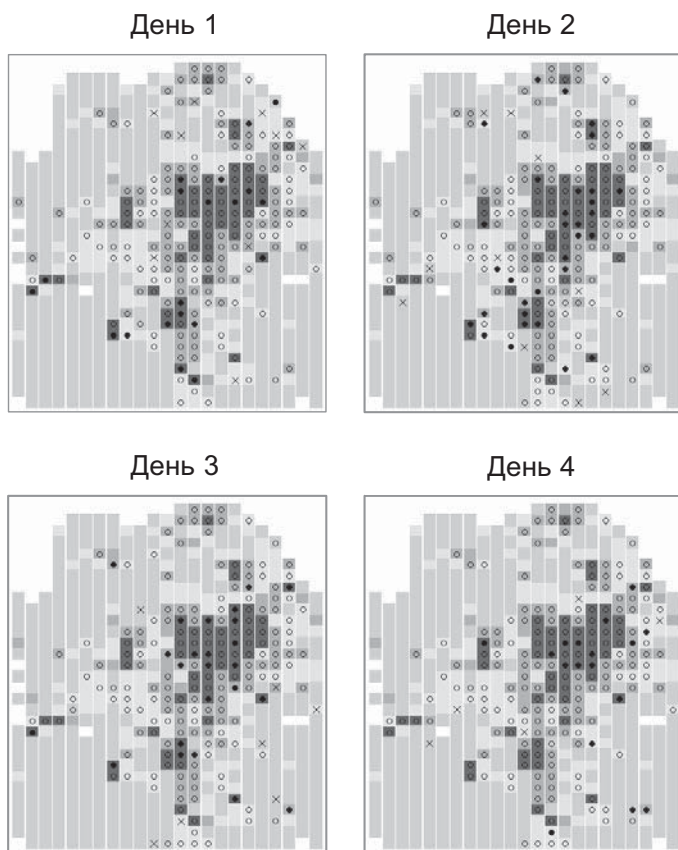
Поскольку лучшую точность показали только 12 из 1000 отдельных деревьев, мы можем располагать 99 %-ной

уверенностью, что случайный лес дает нам более высокую точность, чем отдельно взятое дерево.



**Рис. 2.** Гистограмма прогностической точности 1000 деревьев решений (в среднем 67 %) по сравнению с точностью случайного леса, который получен на их основе (72 %)

На рис. 3 показаны прогнозы случайного леса на четыре дня подряд. Основываясь на наших предсказаниях, полиции следует уделить больше внимания черным участкам и меньше — светлым. Хотя неудивительно, что требуется больше патрулей на территориях, в которых исторически совершается больше преступлений, но модель идет дальше и показывает вероятность совершения преступлений в не черных зонах. Например, для четвертого дня (нижняя правая теплокарта) было верно предсказано преступление в серой зоне, несмотря на отсутствие там криминальной активности за предыдущие три дня.



**Рис. 3.** Прогноз преступлений за четыре дня подряд в 2016 году.

Полыми кругами отмечены участки, в которых ожидалось преступление. Закрашенные круги означают верный прогноз криминальной активности. Крестиками отмечены участки, в которых преступления произошли, но предсказаны не были

Модель случайного леса также позволяет нам увидеть, какие переменные сыграли наибольшую роль в прогнозировании. Согласно диаграмме на рис. 4, о вероятности преступлений лучше всего судить по статистике самих преступлений, месту, дню года и температуре в течение дня.









**Рис. 4.** Главные переменные, участвовавшие в прогнозе преступлений в рамках модели случайного леса

Итак, мы убедились, что случайные леса могут быть очень эффективными в предсказании таких сложных явлений, как преступления. Но как же случайные леса работают?

### 10.3. Ансамбли

Случайный лес — это *ансамбль* деревьев решений. Ансамблем называют прогностическую модель, полученную путем комбинирования предсказаний от других моделей, будь то решение по большинству голосов или учет средних значений.

На рис. 5 показано, как ансамбль, полученный путем большинства голосов, дает более точные результаты, чем отдельные модели, на которых он основывается. Так происходит из-за того, что верные прогнозы склонны подтверждать друг друга, в то время как ошибочные этого не делают. Но для того чтобы эта схема работала, включаемые в ансамбль модели не должны совершать ошибки одного типа. Другими словами, модели должны

Модель 1											70 % верно
Модель 2											70 % верно
Модель 3											60 % верно
Ансамбль											80 % верно

**Рис. 5.** Пример с тремя моделями, предсказывающими десять исходов, каждый из которых может быть темным или светлым. Правильный прогноз для всех десяти — темный. Ансамбль, полученный из трех отдельных моделей путем учета большинства голосов, дал наивысшую точность прогноза: 80 %



быть взаимно независимы, то есть не коррелировать друг с другом.

Метод систематического порождения взаимно независимых деревьев решений известен как бэггинг (bootstrap aggregating).

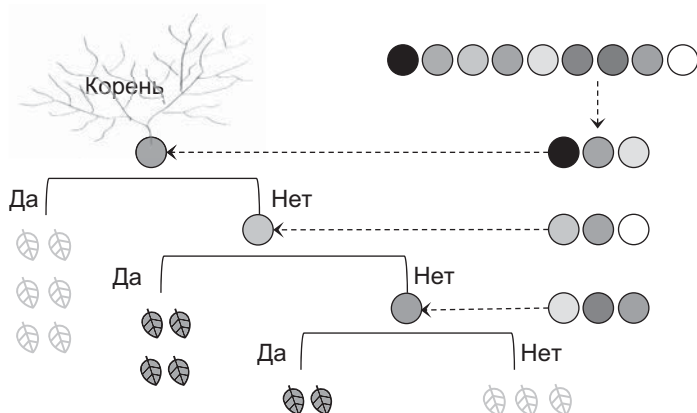
## 10.4. Бэггинг

В прошлой главе мы исходили из того, что дерево решений — это набор данных, который последовательно разбивается на поддеревья с использованием наилучшей комбинации переменных. Тем не менее поиск лучших комбинаций переменных может быть затруднен, поскольку деревья решений склонны к переобучению (раздел 1.3).

Чтобы обойти эту проблему, мы можем сконструировать деревья решений, используя случайные комбинации и порядок переменных, после чего объединить эти деревья для формирования случайного леса.

*Бэггинг* позволяет построить тысячи деревьев решений, которые будут соответствующим образом отличаться друг от друга. Чтобы убедиться в минимальной корреляции между деревьями, каждое из них строится со случайным набором предикторных переменных, а также с использованием случайного фрагмента из обучающего набора данных. Это позволяет строить непохожие деревья, которые при этом сохраняют определенные прогностические способности. Рисунок 6 показывает, как

для построения деревьев используются предикторные переменные.



**Рис. 6.** Создание дерева решений путем бэггинга

На рис. 6 показаны девять предикторных переменных, которые представлены различными оттенками серого. При каждом разбиении набор предикторных переменных случайным образом распределяется, после чего алгоритм дерева решений выбирает для него лучшую переменную.

Ограничивая набор предикторов для каждого разбиения дерева, мы можем получать значительно различающиеся деревья, что позволяет избежать переобучения. Чтобы снизить его влияние еще больше, мы можем увеличить для случайного леса число деревьев решений, в резуль-

тате чего получится более точная и масштабируемая модель.

## **10.5. Ограничения**

Ни одна модель не совершенна. Решение о том, воспользоваться ли данной моделью случайного леса, принимается после соотнесения ее предсказательной силы и интерпретируемости результатов.

**Невозможность интерпретации.** Случайные леса считаются *черными ящиками*, поскольку они состоят из случайно сгенерированных деревьев решений, которые не основаны на ясных прогностических принципах. Например, мы не можем сказать, как именно случайный лес создает свой прогноз, такой как предсказание того, что преступление совершится в определенном месте и в определенное время. Единственное, что мы знаем, это то, что к такому заключению пришло большинство составляющих его деревьев решений. Недостаток ясности, как именно делаются предсказания, создает этические трудности, если этот метод применить к таким областям, как медицинская диагностика.

Тем не менее случайные леса широко используются, поскольку их легко получить. Они очень эффективны в ситуациях, когда точность результатов важнее их интерпретируемости.

## 10.6. Краткие итоги

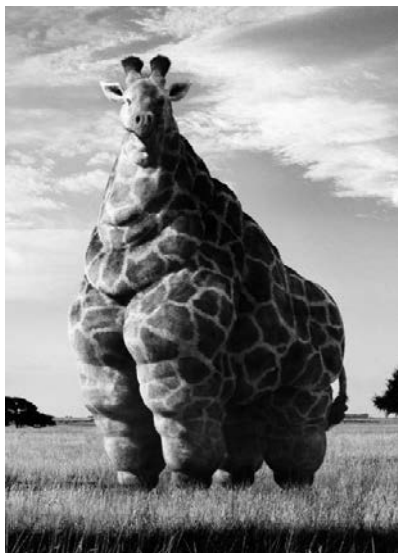
- Случайные леса часто дают более точные предсказания, чем деревья решений, поскольку они задействуют два метода: *бэггинг* и *ансамблирование*.
- Бэггинг подразумевает построение серии взаимно независимых деревьев решений путем случайного ограничения переменных, используемых для разбивки, в то время как ансамблирование комбинирует прогнозы таких деревьев.
- Хотя результаты работы случайного леса не поддаются интерпретации, предикторы могут быть оценены исходя из их вклада в точность прогнозирования.

11

## **Нейронные сети**

## 11.1. Создание мозга

Посмотрите на рис. 1 и догадайтесь, кто на нем изображен.



**Рис. 1.** Догадайтесь, кто это!

Вы должны были распознать жирафа, хотя и весьма необычного, страдающего от избыточного веса. В человеческом мозге соединяются 80 миллиардов нейронов, что позволяет нам легко узнавать объекты, даже если они предстают в ином свете, чем виденные прежде. Нейроны взаимодействуют, преобразуя входные сигналы (картинка с жирафом) в выходные метки (метка «жираф»), что вдохновило на создание метода, известного как нейронные сети.

*Нейронные сети* (neural networks) легли в основу метода автоматического распознавания изображений, и его дальнейшее развитие показывает даже превосходство над людьми по части скорости и точности. Сегодняшняя популярность нейронных сетей связана с тремя ключевыми причинами.

- **Прогресс в передаче и хранении данных.** Это предоставило в наше распоряжение огромные объемы информации, которые можно использовать для обучения нейронных сетей, улучшая тем самым их эффективность.
- **Рост вычислительной мощности.** Графические процессоры (GPU), которые работают почти в 150 раз быстрее, чем центральные (CPU), прежде использовались в основном для отрисовки высококачественной графики в компьютерных играх, но обнаружилось, что они отлично справляются и с обучением нейронных сетей на больших наборах данных.

- **Улучшенные алгоритмы.** Хотя машинам все еще трудно тягаться по производительности с человеческим мозгом, некоторые разработанные методы позволили значительно улучшить их производительность. Часть таких методов будет рассмотрена в данной главе.

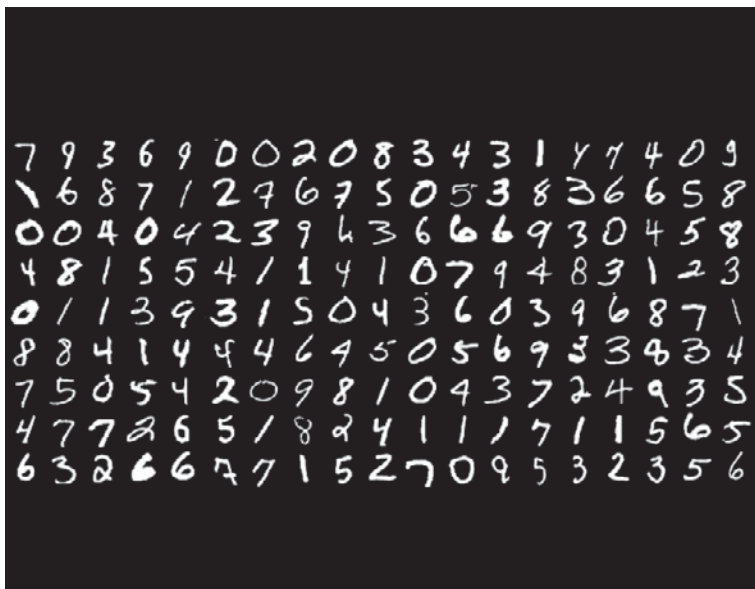
Автоматическое распознавание изображений является поразительным примером способностей нейронных сетей. Его применяют во множестве областей, включая видеонаблюдение и беспилотные транспортные средства. Оно даже используется в приложениях для смартфонов для распознавания рукописного ввода. Давайте посмотрим, как происходит обучение нейронных сетей.

## 11.2. Пример: распознавание рукописных цифр

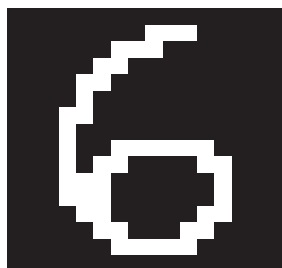
Мы воспользовались рукописными цифрами из базы данных MNIST, предоставляемой американским *Национальным институтом стандартов и технологий*. Примеры цифр показаны на рис. 2.

Чтобы компьютер мог работать с изображениями, их прежде всего нужно представить в виде пикселей. Черным пикселям присваивается значение 0, а белым — 1, как показано на рис. 3. Если изображение цветное, можно было бы работать со значениями цветовой модели RGB (красный, зеленый, синий).





**Рис. 2.** Рукописные цифры из базы данных MNIST



```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 1 1 1 1 1 0 0 0 0
0 0 0 1 0 1 1 0 0 0 0 0 1 0 0 0
0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0
0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0
0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

**Рис. 3.** Перевод изображения в пиксеты

После того как изображение выражено количественно, значения могут быть переданы нейронной сети. Мы загрузили в нее 10 000 рукописных цифр вместе с данными о цифрах, которым эти изображения соответствуют. После того как нейронная сеть научилась связывать изображения цифр с ними самими, мы проверили, сможет ли она распознать 1000 новых изображений рукописных цифр.

Из 1000 рукописных цифр нейронная сеть правильно определила 922, то есть отработала с точностью 92,2 %. Рисунок 4 показывает таблицу сопряженности, которой можно воспользоваться для анализа ошибок идентификации.

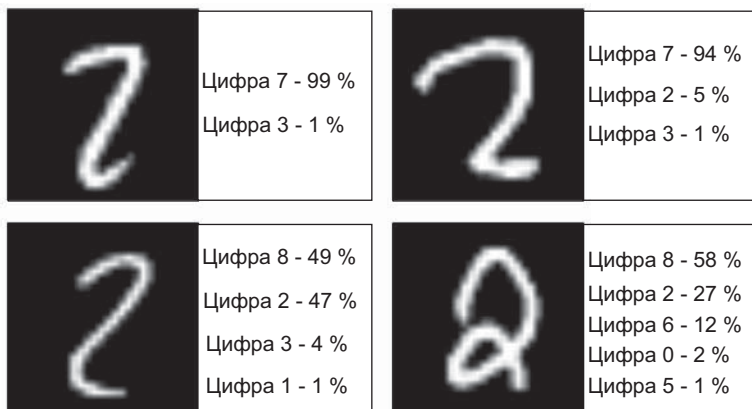
		Распознанное число										Всего	%
		0	1	2	3	4	5	6	7	8	9		
Настоящее число	0	84	0	0	0	0	0	1	0	0	0	85	99
	1	0	125	0	0	0	0	1	0	0	0	126	99
	2	1	0	105	0	0	0	0	4	5	1	116	91
	3	0	0	3	96	0	6	0	1	0	1	107	90
	4	0	0	2	0	99	0	2	0	2	5	110	90
	5	2	0	0	5	0	77	1	0	1	1	87	89
	6	3	0	1	0	1	2	80	0	0	0	87	92
	7	0	3	3	0	1	0	0	90	0	2	99	91
	8	1	0	1	3	1	0	0	2	81	0	89	91
	9	0	0	0	0	1	0	0	6	2	85	94	90
Всего		91	128	115	104	103	85	85	103	91	95	1000	92

**Рис. 4.** Таблица сопряженности иллюстрирует эффективность нейронной сети. Первый ряд показывает, что 84 из 85 изображений нуля были распознаны верно и только одно из них было ошибочно принято за шестерку. Последний столбец показывает точность

Из рис. 4 видно, что рукописные ноль и единица почти всегда определяются верно, в то время как сложнее всего было распознать пятерку. Ознакомимся подробнее с неверно распознанными цифрами.

Примерно в 8 % случаев двойка ошибочно распознавалась как семерка или восьмерка. В то время как человеческий глаз легко опознает цифры из рис. 5, нейронная сеть может быть сбита с толку такими явлениями, как хвостик у двойки. Интересно отметить, что в 10 % случаев нейронная сеть путала тройку с пятеркой (рис. 6).

Несмотря на все эти ошибки, нейронная сеть работает куда быстрее человека, достигая при этом достаточно высокой точности.



**Рис. 5.** Неверное распознавание цифры 2

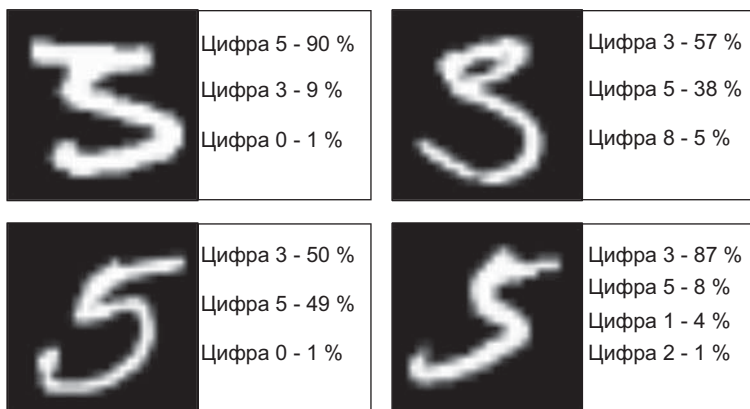


Рис. 6. Неверное распознавание цифр 3 и 5

### 11.3. Компоненты нейронной сети

При распознавании рукописных цифр нейронная сеть использует несколько слоев нейронов, чтобы строить прогноз на основании вводимых изображений.

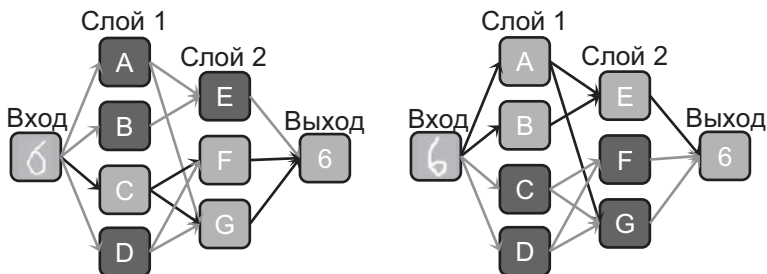


Рис. 7. Пример нейронной сети: разные входные данные с одними выходными. Активные нейроны выделены темным

На рис. 7 показана нейронная сеть, которая, получив различные изображения рукописной шестерки, использует различные нейронные пути для получения одного и того же результата. Хотя каждая комбинация активных нейронов ведет лишь к одному результату, каждый из них может быть достигнут различными комбинациями.

Типичная нейронная сеть состоит из следующих компонентов.

- **Входной слой.** Этот слой обрабатывает каждый пиксел входящего изображения. Поэтому обычно число нейронов в нем совпадает с числом пикселей изображения. Для простоты на рис. 7 множество нейронов показано одним узлом.
- Для улучшения прогнозирования может быть использован и **сверточный слой**. Вместо обработки отдельных пикселей этот слой обнаруживает различные признаки, ориентируясь на комбинации пикселей, такие как присутствие круга или верхнего хвостика у цифры 6. Поскольку такой способ анализа зависит только от присутствия признаков, а не от их расположения, нейронная сеть сможет распознать цифру, даже если ее ключевые признаки будут смещены от центра. Эта способность называется *трансляционной инвариантностью* (translational invariance).
- **Скрытые слои.** После того как пикселы переданы нейронной сети, они претерпевают различные преобразования с целью усиления их похожести на изображении, уже встреченном ранее, благодаря чему

их цифровое значение известно. Хотя задействование большего числа преобразований может привести к предельно высокой точности, оно дается ценой значительного увеличения времени обработки. Как правило, достаточно нескольких слоев. В каждом слое количество нейронов должно быть пропорционально количеству пикселей на изображении. В нашем примере из предыдущего раздела использовался один скрытый слой с 500 нейронами.

- **Выходной слой.** Итоговый прогноз попадает в этот слой, который может состоять либо только из одного нейрона, либо из столько же нейронов, сколько существует возможных выходов.
- **Слой потерь.** Хотя он и не показан на рис. 7, *слой потерь* будет присутствовать в нейронной сети во время обучения. Этот слой, обычно размещаемый последним, дает обратную связь о том, были ли входные данные распознаны верно, и если нет, то о степени погрешности.

Для обучения нейронной сети *слой потерь* жизненно важен. Если сделан верный прогноз, то *слой потерь* подкрепляет приведший к нему путь активации. А если предсказание неправильно, то ошибка возвращается обратно, чтобы нейроны могли перенастроить свои критерии активации для снижения вероятности заблуждения. Этот процесс называется *методом обратного распространения ошибки* (backpropagation).

Путем такого итеративного процесса обучения нейронная сеть учится связывать входные сигналы с правильными

выходными данными, а сами эти ассоциации в дальнейшем становятся *правилами активации* для каждого нейрона. Таким образом, чтобы увеличить точность нейронной сети, нужно настроить компоненты, управляющие правилами активации.

## 11.4. Правила активации

Чтобы построить прогноз, нейроны, в свою очередь, должны быть активированы на протяжении нейронного пути. Активация каждого нейрона управляется *правилом активации*, которое определяет источник и силу входного сигнала, получаемого нейроном перед активацией. Это правило регулируется во время обучения нейронной сети.

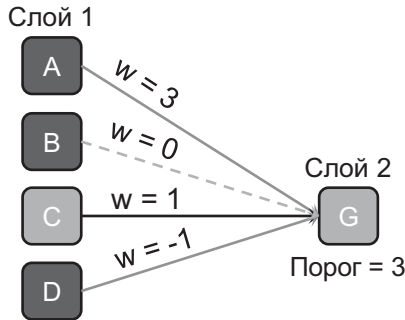


Рис. 8. Пример правила активации

Рисунок 8 иллюстрирует мнимое правило активации для нейрона *G* из случая на рис. 7. После обучения нейрон-

ная сеть выучила, что нейрон  $G$  связан с нейронами  $A$ ,  $C$  и  $D$  из предыдущего слоя. Поэтому любые активации этих трех нейронов превратятся во входные сигналы для нейрона  $G$ .

Ассоциации обладают различным уровнем силы, известным как *вес* ( $w$ ). Например, на рис. 8 видно, что активация нейрона  $A$  передаст более сильный ( $w = 3$ ) сигнал нейрону  $G$ , чем нейрон  $C$  ( $w = 1$ ). Помимо этого, ассоциации обладают направленностью. Так, активация нейрона  $D$  ( $w = -1$ ) приведет к уменьшению входных сигналов, передаваемых нейрону  $G$ .

Чтобы определить итоговый входной сигнал, передаваемый нейрону  $G$ , мы суммируем веса всех активных нейронов предыдущего слоя, с которыми связан нейрон  $G$ . Если итоговый полученный сигнал достигает определенного порога, то нейрон  $G$  будет активирован. На рис. 8 сила итогового сигнала равна  $3 + (-1)$ , то есть 2. Поскольку порог нейрона  $G$  равен 3, он в данном случае не будет активирован.

Изучение правильных значений весов и порогов важно для получения хороших правил активации, которые приведут к точным прогнозам. Кроме того, есть и другие параметры нейронной сети, которые требуют настройки, такие как число скрытых слоев и число нейронов в каждом слое. Для оптимизации этих параметров может быть использован градиентный спуск (раздел 6.3).



## 11.5. Ограничения

Несмотря на теоретическую возможность имитации человеческого разума, нейронные сети не лишены нескольких недостатков. Для борьбы с ними разработаны различные методы.

**Для обучения нужен большой объем данных.** Сложность нейронной сети позволяет распознавать входящие данные по замысловатым признакам, но это возможно только при значительных объемах данных для обучения. Если обучающий сегмент слишком мал, то возможно переобучение (раздел 1.3). Но если получение большего количества данных для обучения затруднительно, то с минимальным риском переобучения можно воспользоваться следующими методами.

- **Подвыборка.** Для того чтобы снизить чувствительность нейронов к шуму, входные данные могут быть «сглажены» путем *подвыборки*. Это достигается путем получения средних значений входного сигнала. Если, например, проделывать это с изображениями, то можно уменьшить их размер или снизить его контрастность.
- **Искажения.** При нехватке данных для обучения можно получить больше данных путем внесения искажений в каждую картинку. Используя каждое искаженное изображение в качестве новых входных данных,

можно увеличить размер обучающего набора. При этом используемые искажения должны соответствовать данным из исходного набора. Например, в случае с рукописными цифрами изображения могут быть повернуты для имитации манеры людей писать под углом, а также просто растянуты или сжаты в отдельных местах (*эластичная деформация*) для имитации колебаний мышц руки.

- **Исключение, или дропаут.** Если данных для обучения немного, нейроны имеют меньше возможностей для формирования связей с другими нейронами, что приводит к переобучению из-за того, что малые нейронные кластеры развивают чрезмерную зависимость друг от друга. Этому можно противопоставить исключение половины нейронов в течение одного цикла обучения. Эти исключенные нейроны будут деактивированы, и оставшиеся будут действовать так, как если бы тех нейронов не было вовсе. Затем на следующей итерации окажется исключен другой набор нейронов. Благодаря этому *исключение* принуждает различные комбинации нейронов к взаимодействию, чтобы они выявили в обучающих примерах больше признаков.

**Требует долгих вычислений.** Обучение нейронной сети, содержащей тысячи нейронов, может занять продолжительное время. Хотя простейшим решением будет модернизация оборудования, она может оказаться слишком дорогой. В качестве альтернативного варианта можно настроить алгоритмы таким образом, чтобы существенно

увеличить скорость обработки за счет небольшого снижения прогностической точности. Есть несколько способов добиться этого.

- **Стохастический градиентный спуск.** При классическом градиентном спуске (раздел 6.3) мы итеративно проходим *весь* обучающий набор для обновления единственного параметра за раз. В случае с большими наборами данных это может слишком затянуться. В качестве альтернативы имеет смысл при обновлении параметра ограничиться только *одним* фрагментом на каждую итерацию. Этот метод называется *стохастическим градиентным спуском*. И хотя в итоге значения параметра могут оказаться не самыми оптимальными, они, как правило, обеспечивают приличную точность.
- **Градиентный спуск Mini-batch.** Хотя использование лишь одного обучающего фрагмента за проход цикла может быть быстрее, но за счет этого работа итогового параметра и всего алгоритма окажется менее точной, то есть параметр отклонится от оптимального значения. Золотой серединой может стать применение для каждого прохода цикла *поднабора* обучающих примеров. Этот метод называется *градиентным спуском Mini-batch*.
- **Полносвязные слои.** С добавлением новых нейронов число возможных нейронных путей возрастает по экспоненте. Чтобы избежать проверки всех возможных комбинаций, можно оставить нейроны в на-

чальных слоях (где обрабатываются низкоуровневые признаки) соединенными лишь частично. И только в финальных слоях (где обрабатываются высокоуровневые признаки) нужно полностью соединить нейроны в смежных слоях.

**Невозможность интерпретации.** Нейронные сети состоят из множества слоев и сотен нейронов, управляемых различными правилами активации. Это делает затруднительным отслеживание комбинации входных сигналов, дающих верный прогноз. Это отличается от методов типа регрессии (глава 6), значимые предикторы которых легко определить и сравнить. Из-за того что нейронная сеть является *черным ящиком*, становится нелегко обосновать ее применение, особенно в этически значимых решениях. Тем не менее продолжаются исследования по анализу процесса обучения в каждом слое, чтобы выяснить, как отдельные входные сигналы влияют на итоговый прогноз.

Несмотря на эти ограничения, эффективность нейронных сетей продолжает побуждать применять их в таких передовых технологиях, как виртуальные помощники и автономное пилотирование. Помимо имитации людей нейронные сети уже превосходили человеческие способности в некоторых областях. Так случилось и в показательном матче 2015 года по игре в го, когда человек проиграл нейронной сети Google. Так как мы продолжаем усовершенствовать алгоритмы и раздвигаем границы вычислительных возможностей, нейронные сети, соединяя и автоматизируя наши повседневные задачи, станут играть ключевую роль в эпоху *интернета вещей*.

## 11.6. Краткие итоги

- Нейронные сети состоят из слоев нейронов. В процессе обучения нейроны первого слоя активируются входными данными, и эта активация передается в следующие слои, в конечном счете попадая в последний слой, где формируется прогноз.
- Будет ли активирован нейрон, зависит от силы и источника полученной активации в соответствии с его *правилом активации*. Правила активации затачиваются в результате обратной связи по точности прогноза. Этот процесс называется *методом обратного распространения ошибки*.
- Нейронные сети работают лучше всего, когда доступны большие наборы данных и производительное оборудование. Однако результаты в значительной степени будут неинтерпретируемыми.



# 12

## **A/B-тестирование и многорукие бандиты**

## 12.1. Основы А/В-тестирования

Представьте, что вы управляете онлайн-магазином и хотите запустить рекламу, информирующую людей о текущих предложениях. Какую фразу вы бы использовали?

- Скидки до 50 % на товары!
- Некоторые товары за полцены.

Хотя обе фразы имеют одинаковый смысл, одна из них может оказаться убедительнее второй. Возникают, к примеру, следующие вопросы. Стоит ли использовать восклицательный знак, чтобы вызвать азарт у покупателей? Окажутся ли «50 %» заманчивее, чем «полцены»?

Чтобы выяснить, что именно сработает, можно в течение пробного периода показывать каждую версию рекламы 100 людям, оценив, сколько раз они кликнули на каждой из них. Реклама, которая соберет больше кликов, привлечет и больше покупателей, и таким образом, она должна использоваться на протяжении всей последующей рекламной кампании. Эта процедура называется *А/В-тестированием*, когда сравнивается эффективность версий *А* и *В*.



## 12.2. Ограничения А/В-тестирования

Метод А/В-тестирования имеет две проблемы.

**Результаты могут быть обычным совпадением.** По чистой случайности неудачная реклама может превзойти лучшую. Для большей уверенности в результатах мы можем увеличить число людей, которым показываются разные версии, но это и приводит нас ко второй проблеме.

**Возможная потеря прибыли.** Увеличивая число людей, которым мы будем показывать разные версии рекламы со 100 до 200, мы удваиваем показ менее успешной рекламы, что может привести к потере покупателей, которых бы убедила лучшая версия.

Две эти проблемы показывают компромисс А/В-тестирования: *эксплорация* против *эксплуатации*. Если увеличить число людей для тестирования рекламы (эксплорация), то можно надежнее выявить, какая из версий лучше, но будут потеряны возможные покупатели, которые могли совершить покупки, увидев лучшую рекламу (эксплуатация).

Как же найти равновесие?

## 12.3. Стратегия снижения эпсилона

В то время как А/В-тестирование подразумевает, что исследование того, какая из версий лучше, предшествует

ее применению, нам в действительности необязательно ждать окончания эксплорации до начала эксплуатации.

Если у первых 100 посетителей реклама *A* собрала больше кликов, чем реклама *B*, то для следующих 100 посетителей мы можем увеличить ее показ на 60 %, снизив показ рекламы *B* до 40 %. Это позволит нам применить уже первоначальные результаты, говорящие о большей эффективности версии *A*, не мешая продолжать исследование на случай, если эффективность версии *B* улучшится. Чем больше результаты будут свидетельствовать в пользу рекламы *A*, тем меньше мы будем показывать рекламу *B*.

Этот подход называется *стратегией снижения эpsilon* (Epsilon-Decreasing Strategy). *Эпсилоном* обозначается доля времени, которая тратится на показ альтернативы для подтверждения ее низкой эффективности. Поскольку мы снижаем эпсилоном по мере укрепления нашей уверенности в том, что одна из версий лучше, этот метод принадлежит к классу алгоритмов *обучения с подкреплением*.



**Рис. 1.** В то время как A/B-тестирование включает один этап эксплорации и один этап эксплуатации, стратегия снижения эpsilon чередует их, постепенно увеличивая эксплуатацию

## 12.4. Пример: многорукие бандиты

Типичным примером для иллюстрации различий между А/В-тестированием и стратегией снижения эpsilon является игровой автомат по типу слот-машины. Предположим, что слот-машины имеют разный коэффициент отдачи, а цель игрока, выбрать ту из них, которая обеспечит лучший выигрыш.



**Рис. 2.** Однорукий бандит

Слот-машины прозвали *однорукими бандитами* за их умение с каждым нажатием рычага опустошать карманы игроков. Выбор того, на какой из слот-машин играть, известен как проблема *многорукого бандита*, как теперь называют любую схожую задачу с распределением ресурсов, например то, какую онлайн-рекламу показывать, какие темы освежить перед экзаменом или какие фармацевтические исследования профинансировать.

Предположим, что выбрать нужно из двух слот-машин, *A* и *B*, а у нас достаточно денег, чтобы сыграть на них 2000 раз. Во время каждой игры мы дергаем рычаг, что может принести нам \$ 1 или не вернуть ничего.

**Таблица 1.** Коэффициенты отдачи слот-машин

Слот-машина	Коэффициент отдачи
А	0,5
В	0,4

Итак, шанс выплаты составляет 50 % для слот-машины А и 40 % для слот-машины В. Тем не менее нам это неизвестно. Вопрос в том, как нам играть, чтобы максимизировать выигрыш.

Давайте сравним возможные стратегии.

**Полная эксплорация.** Если мы играем на слот-машинах по очереди, то получим \$ 900.

**А/В-тестирование.** Если мы применим А/В-тестирование на первых 200 играх, а затем используем это знание для следующих 1800 игр, то сможем выиграть в среднем \$ 976. Но здесь есть подводный камень: поскольку коэффициент отдачи обеих слот-машин схож, есть 8 %-ный шанс, что мы ошибочно сочтем наиболее выгодной слот-машину В.

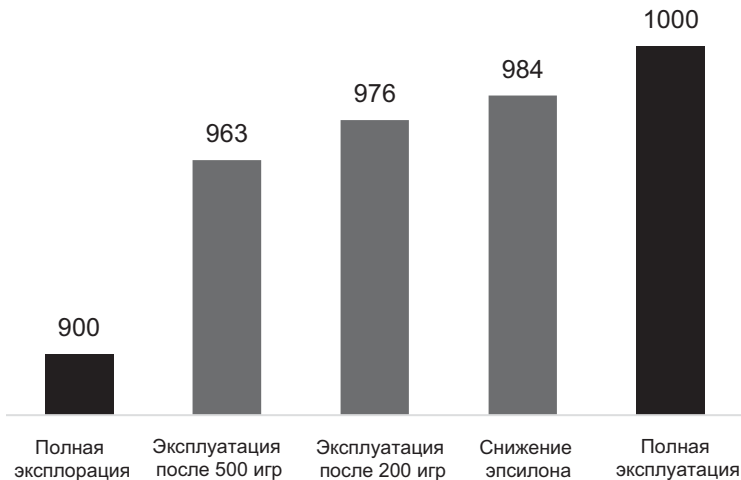
Чтобы снизить риск такой ошибки, мы можем увеличить эксплорацию до 500 игр. Это уменьшит вероятность ошибки до 1 %, но наш выигрыш тоже упадет до \$963.

**Стратегия снижения эpsilon.** Если мы используем стратегию снижения эpsilon, чтобы во время игр отдавать приоритет более щедрой слот-машине, то сможем выиграть в среднем 984 \$ при 4 %-ной вероятности ошиб-

ки. Мы можем снизить риск ошибки путем увеличения доли эксплорации (значения  $\epsilon$ ), но, как и прежде, это повлияло бы на наш выигрыш.

**Полная эксплуатация.** Если мы располагаем инсайдерской информацией о том, что слот-машина  $A$  возвращает больше, мы эксплуатируем ее с самого начала, рассчитывая в среднем на \$ 1000. Но это (почти) недостижимо.

Из рис. 3 видно, что при отсутствии инсайдерской информации стратегия снижения  $\epsilon$  дает наибольший выигрыш. Более того, при большом числе игр математическое свойство, называемое сходимостью, гарантирует, что эта стратегия обязательно выявит лучшую слот-машину.



**Рис. 3.** Сравнение выигрышей при использовании различных стратегий

## 12.5. Забавный факт: ставка на победителя

Интересный случай проблемы многорукого бандита встречается в спорте. Во время работы в английском футбольном клубе «Манчестер Юнайтед» главный тренер Луи ван Гал ввел необычную стратегию для того, чтобы определять порядок игроков во время серии пенальти.

Первый назначенный игрок продолжает бить пенальти, пока не промахнется. Вслед за ним до первого промаха бьет пенальти второй игрок и т. д. Эта стратегия известна как *ставка на победителя*.

Если бы мы применили эту футбольную стратегию в примере со слот-машинами из табл. 1, ставя на слот-машину, которая принесла выигрыш, и сразу переключаясь на другую при проигрыше, наш результат составил бы около \$ 909, что лишь ненамного лучше случайной игры. Если часто менять слот-машину, то получится много эксплорации и слишком мало эксплуатации. Кроме того, ставка на победителя на основе лишь последней игры никак не учитывает результаты других прошлых игр. Становится очевидно, что эта стратегия далека от совершенства.

## 12.6. Ограничения стратегии снижения эпсилона

Хотя стратегия снижения эпсилона кажется превосходной, она также подвержена ограничениям, из-за которых ее труднее применить, чем А/В-тестирование.

При использовании этой стратегии ключевым фактором становится значение эпсилона. Если эпсилон снижается слишком медленно, то можно потерять на том, что используется не лучшая слот-машина. Если же он снижается слишком быстро, можно ошибиться с выбором лучшей слот-машины.

Оптимальное снижение эпсилона зависит от того, насколько сильно различаются коэффициенты отдачи двух слот-машин. Если они довольно близки, как в табл. 1, эпсилон следует снижать медленно. Для вычисления эпсилона можно также использовать метод, называющийся *семплированием Томпсона*.

Стратегия снижения эпсилона также зависит от следующих допущений.

1. **Коэффициент отдачи все время постоянен.** Может оказаться так, что одна реклама популярнее по утрам, а другая пользуется умеренной популярностью в те-

чение дня. Если мы сравниваем их утром, то можем ошибочно заключить, что первая реклама лучше.

2. **Коэффициент отдачи не зависит от предыдущих игр.** Если реклама показана несколько раз, посетитель может вдруг заинтересоваться и все-таки кликнуть на ней. Это значит, что для выявления настоящей отдачи эксплорация должна повторяться.
3. **Между игрой на слот-машине и получением отдачи минимальная задержка.** Если реклама приходит по электронной почте, потенциальные покупатели могут не отвечать в течение нескольких дней. Это не позволяет мгновенно получить результаты эксплорации, и любые попытки одновременной эксплуатации будут основаны на неполной информации.

Тем не менее если при сравнении обеих реклам было нарушено второе либо третье допущение, эффект этих ошибок еще может быть преодолен. Например, если две рекламы рассылались по электронной почте, то задержка произойдет в обоих случаях, и сравнение останется честным.

## 12.7. Краткие итоги

- Проблема многорукого бандита отвечает на вопрос о лучшем распределении ресурсов: использовать ли полученные сведения или искать лучшую альтернативу.
- При одном подходе мы сначала изучаем доступные варианты, после чего тратим все оставшиеся ресурсы на



тот, который сочли лучшим. Эта стратегия называется *A/B-тестированием*.

- При другом подходе мы постепенно увеличиваем долю ресурсов, выделяемых для варианта, который показывает лучший результат. Это называется *стратегией снижения эpsilon*.
- Хотя стратегия снижения эpsilon и работает лучше, чем A/B-тестирование, оптимальную долю ресурсов для перераспределения определить нелегко.



# Приложения

## Приложение А. Обзор алгоритмов обучения без учителя

		Классификация методом $k$ -средних	Метод главных компонент	Ассоциативные правила	Лувенский метод	PageRank
Вход	Бинарные значения			✓		
	Непрерывные значения	✓	✓			
	Узлы и ребра				✓	✓
Выход	Категории	✓	✓		✓	
	Ассоциации			✓		
	Ранги					✓

## Приложение В. Обзор алгоритмов обучения с учителем

		Регрессионный анализ	Метод $k$ -ближайших соседей	Метод опорных векторов	Деревья решений	Случайные леса	Нейронные сети
Прогнозирование	Бинарные переменные	✓	✓	✓	✓	✓	✓
	Категориальные переменные		✓		✓	✓	✓
	Возможные классы	✓	✓		✓	✓	✓
	Непрерывные переменные	✓	✓		✓	✓	✓
	Нелинейные отношения		✓	✓	✓	✓	✓
Анализ	Большое число переменных			✓	✓	✓	✓
	Простота использования	✓	✓		✓	✓	
	Быстрота вычислений	✓			✓		
Результаты	Высокая точность					✓	✓
	Интерпретируемость	✓	✓		✓		

## Приложение С. Список параметров настройки

	Параметры настройки
Регрессионный анализ	<ul style="list-style-type: none"><li>• Параметр регуляризации (для лассо или ридж-регрессии)</li></ul>
Метод $k$ -ближайших соседей	<ul style="list-style-type: none"><li>• Число ближайших соседей</li></ul>
Метод опорных векторов	<ul style="list-style-type: none"><li>• Параметр стоимости</li><li>• Параметры ядра</li><li>• Параметр эластичности</li></ul>
Дерево решений	<ul style="list-style-type: none"><li>• Минимальный размер конечных узлов</li><li>• Максимальное число конечных узлов</li><li>• Максимальная глубина дерева</li></ul>
Случайные леса	<ul style="list-style-type: none"><li>• Все параметры деревьев решений</li><li>• Число деревьев</li><li>• Число переменных для выбора на каждой разбивке</li></ul>
Нейронные сети	<ul style="list-style-type: none"><li>• Число скрытых слоев</li><li>• Число нейронов в каждом слое</li><li>• Число итераций обучения</li><li>• Коэффициент скорости обучения</li><li>• Первоначальные веса</li></ul>

## Приложение D. Другие метрики оценки

Метрики оценки различаются по тому, как они определяют различные типы погрешностей прогнозирования и как штрафуют за них. В этом приложении представлено несколько наиболее типичных метрик в дополнение к рассмотренным в разделе 1.4.

### Метрики классификации

**Площадь под ROC-кривой, AUROC.** *AUROC* (Area Under the Receiver Operating Characteristic Curve) — это метрика, позволяющая выбирать между максимизацией доли истинно положительных результатов и минимизацией доли ложноотрицательных результатов.

- **Доля истинно положительных результатов (TPR)** — это доля правильно определенных положительных результатов среди всех положительных:

$$\text{TPR} = \text{TP} / (\text{TP} + \text{FN}).$$

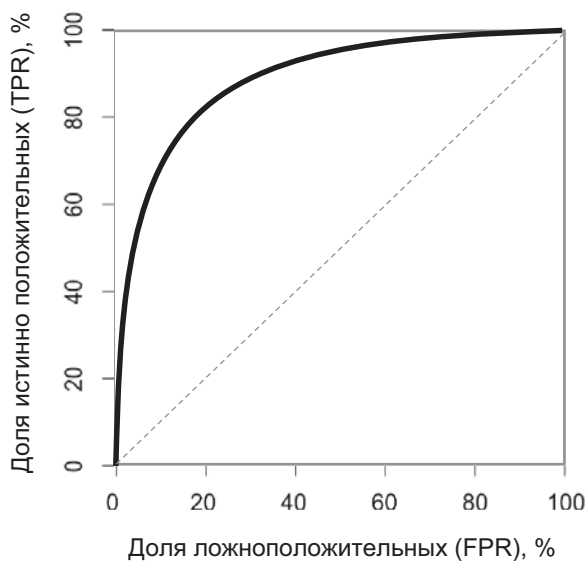
- **Доля ложноположительных результатов (FPR)** — это доля неправильно определенных отрицательных результатов среди всех отрицательных:

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN}).$$

В самом крайнем случае можно пойти по пути максимизации доли истинно положительных результатов ( $\text{TPR} = 1$ ), определяя все значения как положительные. Хотя это

полностью убирает ложноотрицательные результаты, это также значительно увеличивает число ложноположительных. Другими словами, необходимо равновесие между минимизацией ложноположительных и максимизацией истинно положительных результатов.

Этот баланс может быть визуализирован на *ROC-кривой* (рис. 1).



**Рис. 1.** ROC-кривая показывает баланс между максимизацией истинно положительных и минимизацией ложноположительных результатов

Эффективность модели оценивается с помощью площади, охватываемой ROC-кривой, поэтому метрика и на-



зывается *площадью под кривой ошибок* (AUC). Чем точнее модель, тем ближе кривая к верхней левой границе графика. Идеальная модель продемонстрировала бы кривую при  $AUC = 1$ , что эквивалентно всей площади графика. В противоположность ей эффективность модели со случайным прогнозом была бы представлена диагональной пунктирной линией при  $AUC = 0,5$ .

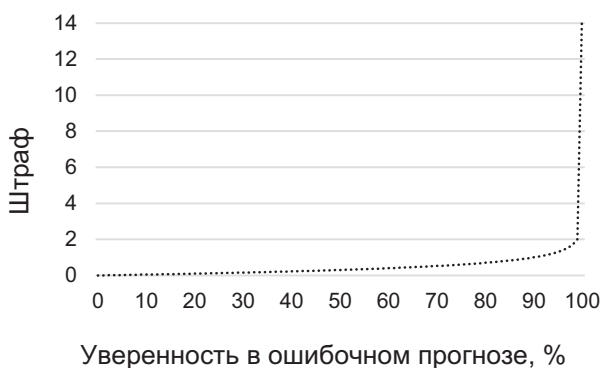
На практике мы можем определить лучшую модель по тому, что она захватывает большую площадь AUC, а ее ROC-кривая использовалась бы для того, чтобы определить подходящий порог TPR и FPR, с которыми мы готовы смириться.

Теперь, когда ROC-кривая позволила нам выбрать тип ошибки, которого мы больше всего хотим избежать, можно применить штрафы ко всем ошибочным предсказаниям с использованием такой метрики, как логарифмическая функция потерь (*logarithmic loss metric*).

**Логарифмическая функция потерь.** При работе с бинарными и категориальными переменными предсказания обычно выражаются в виде вероятности того, что покупатель купит рыбу. Чем ближе вероятность к 100 %, тем *увереннее* модель в том, что покупатель купит рыбу. Логарифмическая функция потерь использует эту уверенность модели для того, чтобы штрафовать за ошибочные прогнозы: чем выше уверенность, тем больше штраф.

На рис. 2 показано, что штраф резко увеличивается, как только модель достигает определенной степени уверенности. Например, если модель предсказывает, что по-

купатель купит рыбу с вероятностью 80 %, но оказалось, что он не купил, то штраф составит 0,7 единиц. Если же модель была уверена на 99 %, то штраф увеличивается до 2 единиц.



**Рис. 2.** Логарифмическая функция потерь возрастает параллельно уверенности модели при ошибочном прогнозе

Из-за своей способности соотносить штрафы с уверенностью модели при прогнозе логарифмическая функция потерь широко используется в случаях, где ошибочные прогнозы весьма критичны.

## Метрики регрессии

**Средняя абсолютная ошибка (Mean Absolute Error, MAE).** Простой способ оценки моделей регрессии заключается в том, чтобы штрафовать за все ошибки одинаково, вычислив среднее отклонение между предсказанным

и действительным значением для всех элементов данных. Эта метрика называется *средней абсолютной ошибкой*.

**Корень из среднеквадратичной логарифмической ошибки (Root Mean Squared Logarithmic Error, RMSLE).**

В разделе 1.4 мы описали метрику *корень из среднеквадратичной ошибки* (RMSE), которая увеличивает штрафы за большие ошибки. Но помимо величины ошибки можно также принять во внимание и ее направление, используя метрику *корень из среднеквадратичной логарифмической ошибки* (RMSLE). RMSLE используется в случаях, когда мы хотим избежать недооценки больше, чем переоценки, например, при предсказании спроса на зонты в дождливый день. Недооценка приведет к недовольству покупателей и упущенной выгоде, в то время как переоценка означала бы только лишние запасы.

# Глоссарий

**А/В-тестирование.** Стратегия сравнения отдачи двух продуктов, *A* и *B*. Процесс начинается с этапа эксплорации (исследования), при котором оба продукта тестируются в равной степени. После этого определяется лучший продукт, и на этапе эксплуатации (применения) на него направляются все ресурсы для максимизации отдачи. Ключевым решением, определяющим поведение А/В-тестирования, является соотношение эксплорации (для нахождения лучшей альтернативы) и эксплуатации (для увеличения возможной отдачи).

**PageRank.** Алгоритм, который определяет доминирующие узлы в сети. Он ранжирует узлы, основываясь на количестве связей, а также на их силе и источнике.

**Ансамблирование.** Метод, комбинирующий различные модели для повышения точности прогнозирования. Такой способ дает хорошие результаты в силу того, что точные прогнозы склонны подтверждать друг друга, чего не делают ошибочные.

**Ассоциативные правила.** Метод обучения без учителя, обнаруживающий ассоциации среди элементов данных,

например товары, которые часто покупают вместе. Есть три типичные ассоциативные метрики:

- *поддержка*  $\{X\}$  показывает, как часто появляется  $X$ ;
- *достоверность*  $\{X \rightarrow Y\}$  показывает, как часто  $Y$  появляется в присутствии  $X$ ;
- *лифт*  $\{X \rightarrow Y\}$  показывает то, как часто  $X$  и  $Y$  появляются вместе, в сравнении с тем, как часто они появляются по отдельности.

**Бэггинг.** Метод, при котором во избежание переобучения создаются тысячи взаимно независимых деревьев решений, от предсказаний которых берутся средние значения. Каждое дерево строится на основе случайного поднабора данных для обучения с использованием столь же случайного поднабора предикторных переменных, выбираемых при каждом ветвлении дерева.

**Валидация.** Оценка того, насколько точно модель строит прогноз для новых данных. Это подразумевает разделение имеющегося набора данных на две части. Первая часть выступает в роли обучающего набора данных, на основании которого создается прогностическая модель. Вторая часть служит тестовым набором данных, который используется для оценки точности модели.

**Градиентный бустинг.** Метод обучения с учителем, при котором строится множество деревьев решений путем использования различных комбинаций бинарных вопросов для каждой ветви. Бинарные вопросы выбираются стратегически (а не случайно, как при использовании случайных лесов), в результате чего прогностическая точ-

ность каждого дерева увеличивается. После этого предсказания отдельных деревьев комбинируются, при этом прогнозы новых деревьев получают больший вес, и процесс повторяется до получения итоговых результатов.

**Градиентный спуск.** Метод настройки параметров модели. При градиентном спуске делается первоначальное предположение о значении параметров, после чего начинается итеративный процесс их применения ко всем элементам данных. В ходе этого процесса значения меняются с целью максимального снижения погрешности прогнозирования.

**График осыпи.** График, позволяющий определить нужное число групп, в роли которых могут выступать, например, кластеры данных или число измерений при уменьшении размерности. Оптимальное число групп обычно определяется по расположению острого изгиба на графике. Большее количество групп может дать менее масштабируемые результаты.

**Дерево решений.** Метод обучения с учителем, который строит прогноз путем формирования последовательности бинарных вопросов, постепенно разбивающих элементы данных на однородные группы. Деревья решений просты для визуализации и понимания, но подвержены переобучению.

**Исключение (дропаут).** Метод, позволяющий избегать переобучения нейронной сети, при котором мы случайным образом исключаем различные поднаборы нейронов при каждой итерации обучения, вынуждая разные

комбинации нейронов к взаимодействию, что позволяет обнаружить больше признаков.

**Классификация.** Класс методов обучения с учителем, при которых предсказываются бинарные или категориальные переменные.

**Корень из среднеквадратичной ошибки.** Метрика, оценивающая точность регрессии. Она особенно полезна в случаях, когда важно избежать больших ошибок. Каждая из них возводится в квадрат, что усиливает значение больших ошибок и делает метрику крайне чувствительной к резко отклоняющимся значениям.

**Корреляция.** Метрика, измеряющая линейную ассоциацию двух переменных. Коэффициенты корреляции варьируются от  $-1$  до  $1$  и несут две единицы информации: а) силу ассоциации, которая максимальна при  $-1$  и  $1$  и минимальна при  $0$ , а также б) направление ассоциации, при котором число положительное, если переменные возрастают в одном направлении, и отрицательное — если в противоположных.

**Кросс-валидация.** Метод максимизации доступных для валидации данных путем разбиения набора данных на несколько сегментов, которые поочередно используются для проверки модели. За одну итерацию все сегменты, кроме одного, используются для обучения прогностической модели, которая затем проверяется на пропущенном сегменте. Этот процесс повторяется до тех пор, пока каждый сегмент не будет использован в качестве проверочного один раз. За итоговую оценку точности

прогностической модели берется средний показатель за все проходы цикла.

**Линия наилучшего соответствия.** Линия тренда, который проходит близко к максимальному числу элементов данных.

**Лувенский метод.** Метод обучения без учителя, который идентифицирует кластеры в сети путем максимизации числа внутрикластерных связей и минимизации связей межкластерных.

**Матрица неточностей.** Метрика, оценивающая точность классификации. Помимо общей оценки точности классификации, матрица выявляет доли ложноположительных и ложноотрицательных предсказаний.

**Метод  $k$ -ближайших соседей.** Метод обучения с учителем, при котором элементы данных классифицируются исходя из близости к соседним элементам. Число ближайших соседей задается  $k$ .

**Метод  $k$ -средних.** Метод обучения без учителя, при котором похожие элементы данных объединяются в группы, число которых задается  $k$ .

**Метод главных компонент.** Метод обучения без учителя, при котором количество переменных для анализа снижается путем комбинирования наиболее информативных из них в новые переменные, называемые главными компонентами.

**Метод обратного распространения ошибки.** Способ обратной связи в нейронной сети, возвращающий ин-



формацию о точности прогноза. Если прогноз неверен, то ошибка передается обратно по нейронному пути, что позволяет нейронам изменить критерии активации, чтобы избегать ее в будущем.

**Метод опорных векторов.** Алгоритм обучения с учителем, который классифицирует элементы данных в две группы, граница между которыми прокладывается между периферийными элементами данных, то есть опорными векторами обеих групп. При работе с изогнутыми границами используют функцию ядра.

**Мультиколлинеарность.** Проблема, возникающая при регрессионном анализе, из-за которой использование высококоррелирующих предикторов приводит к искаженному значению их веса.

**Настройка параметров.** Процесс регулировки параметров алгоритма для повышения его точности, похожий на настройку радиоприемника на нужную волну.

**Недообучение.** Явление, при котором прогностическая модель недостаточно чувствительна и не обнаруживает существующих закономерностей. Недообученная модель склонна упускать из виду важные тренды, из-за чего дает менее точные прогнозы как для текущих, так и для будущих данных.

**Нейронная сеть.** Метод обучения с учителем, который использует слои нейронов для передачи активации, благодаря чему возможно обучение и прогнозирование. Из-за своей сложности результаты не поддаются интерпретации, хотя обладают высокой точностью.

**Обучающий набор данных.** Часть данных, используемая для поиска возможных закономерностей, на основании которой строится прогностическая модель. Такая модель оценивается при помощи тестового набора данных.

**Обучение без учителя.** Класс алгоритмов машинного обучения, используемых для обнаружения скрытых закономерностей в данных. Название обусловлено тем, что они применяются, когда закономерности в данных неизвестны и от алгоритмов ожидается их обнаружение.

**Обучение признаков.** Процесс создания новых переменных путем перекодирования одной из них либо комбинирования нескольких.

**Обучение с подкреплением.** Класс алгоритмов машинного обучения, при которых прогноз строится на основе закономерностей в данных и, кроме того, продолжает улучшаться по мере поступления новых результатов.

**Обучение с учителем.** Класс алгоритмов машинного обучения, используемый для прогнозирования. Название обусловлено тем, что для прогнозирования используются предварительно заданные шаблоны.

**Переменная.** Информация, описывающая элементы данных. Переменные также известны как атрибуты, признаки и размерности. Есть несколько типов переменных:

- **Бинарная.** Простейший тип переменных со всего двумя возможными значениями, например мужское/женское.

- **Категориальная.** Переменная, допускающая более двух значений (например, этническая принадлежность).
- **Целочисленная.** Переменная, используемая для представления целых чисел (например, возраста).
- **Непрерывная.** Наиболее детальный тип переменной, представляющий числа с десятичными дробями (например, цену).

**Переобучение.** Явление, при котором прогностическая модель слишком чувствительна и принимает случайные колебания данных за постоянные закономерности. Переобученная модель может давать высокоточные прогнозы по текущему набору данных, но плохо справляться с новыми.

**Подвыборка.** Метод предотвращения переобучения в нейронной сети, при котором входные данные «разбавляются» средними значениями. Например, при выполнении процедуры над изображениями можно уменьшить размер картинки или снизить контрастность.

**Правило активации.** Критерий, определяющий источник и силу входного сигнала, которые приводят к активации нейрона. Активации нейронов распространяются по нейронной сети для получения прогноза.

**Принцип *Apriori*.** Правило, в соответствии с которым если товарный набор редок, то и включающий его в себя более широкий товарный набор тоже следует считать редким. Этот метод используется для снижения числа

конфигурации для анализа товарных ассоциаций и частоты появления товаров.

**Проблема многорукого бандита.** Термин, используемый для описания любой задачи по распределению ресурсов, похожей на выбор того, на какой слот-машине лучше играть. Название связано с тем, что слот-машины были прозваны однорукими бандитами за их умение с каждым нажатием рычага опустошать карманы игроков.

**Регрессионный анализ.** Метод обучения с учителем, при котором находится линия наилучшего соответствия, пролегающая максимально близко к наибольшему числу элементов данных. Такая линия тренда вычисляется на основе взвешенной комбинации предикторов.

**Регуляризация.** Метод, предотвращающий переобучение прогностической модели путем введения штрафного параметра, который искусственно усиливает значимость любой прогностической ошибки при увеличении сложности модели. Это позволяет учитывать как точность, так и сложность модели при оптимизации ее параметров.

**Рекурсивное деление.** Процесс последовательного разбития данных с целью получения однородных групп, который используется, в частности, в деревьях решений.

**Случайный лес.** Метод обучения с учителем, при котором строится множество деревьев решений. Для формирования каждой ветви дерева используется случайная комби-

нация бинарных вопросов. Затем прогнозы отдельных деревьев суммируются для получения результатов.

**Стандартизация.** Процесс трансформации переменных в единую стандартную шкалу, такую как выражение каждой переменной в процентилях.

**Стратегия снижения эпсилона.** Метод обучения с подкреплением, при котором ресурсы распределяются путем чередования двух этапов: а) поиск лучшей альтернативы; б) применение полученных результатов. Эпсилоном называется доля времени, которая тратится на поиск лучшей альтернативы (экслорацию). По мере накопления информации о том, какая из альтернатив лучше, эпсилон снижается.

**Тестовый набор данных.** Часть данных, используемая для оценки точности и масштабируемости прогностической модели. Во время построения модели используется только обучающий набор данных, а тестовый умышленно пропускается.

**Трансляционная инвариантность.** Свойство сверточной нейронной сети распознавать признаки на изображении вне зависимости от их положения.

**Уменьшение размерности.** Процесс снижения количества переменных, например, путем комбинирования высококоррелирующих.

**Функция ядра.** Метод проекции элементов данных на дополнительное измерение, благодаря чему они могут быть

разделены прямой разграничивающей линией. Такие прямые линии проще вычислить, и затем, при возврате к исходному числу измерений, их можно легко преобразовать в кривые.

**Черный ящик.** Термин, используемый для описания неинтерпретируемой прогностической модели, то есть такой, для которой нет ясной формулы, по которой она строит прогноз.

# Литература и ссылки на источники

## Источники на английском языке

### **Личностные характеристики пользователей Facebook** (*k*-Means Clustering)

Stillwell, D., & Kosinski, M. (2012). myPersonality Project [Data files and description]. Sample dataset can be retrieved from <http://data.miningtutorial.com>

Kosinski, M., Matz, S., Gosling, S., Popov, V., & Stillwell, D. (2015).

Facebook as a Social Science Research Tool: Opportunities, Challenges, Ethical Considerations and Practical Guidelines. *American Psychologist*.

### **Пищевая ценность** (Principal Component Analysis)

Agricultural Research Service, United States Department of Agriculture (2015). USDA Food Composition Databases [Data]. Retrieved from <https://ndb.nal.usda.gov/ndb/nutrients/index>

### **Покупки в магазине** (Association Rules)

Dataset is included in the following R package: Hahsler, M., Buchta, C., Gruen, B., & Hornik, K. (2016). arules: Mining Association Rules and Frequent Itemsets. R package version 1.5-0. <https://CRAN.Rproject.org/package=arules>

Hahsler, M., Hornik, K., & Reutterer, T. (2006). Implications of Probabilistic Data Modeling for Mining Association Rules. In Spiliopoulou, M., Kruse, R., Borgelt, C., Nürnberger, A., & Gaul, W. Eds., *From Data and Information Analysis to Knowledge Engineering, Studies in Classification, Data Analysis, and Knowledge Organization*. pp. 598–605. Berlin, Germany: Springer-Verlag.

Hahsler, M., & Chelluboina, S. (2011). Visualizing Association Rules: Introduction to the R-extension Package arulesViz. R Project Module, 223–238.

### **Торговля оружием (Network Graphs)**

Stockholm International Peace Research Institute (2015). Trade Registers [Data]. Retrieved from [http://armstrade.sipri.org/armstrade/page/trade\\_register.php](http://armstrade.sipri.org/armstrade/page/trade_register.php)

### **Цены на дома (Regression Analysis)**

Harrison, D., & Rubinfeld, D. (1993). Boston Housing Data [Data file and description]. Retrieved from <https://archive.ics.uci.edu/ml/datasets/Housing>

Harrison, D., & Rubinfeld, D. (1978). Hedonic Prices and the Demand for Clean Air. *Journal of Environmental Economics and Management*, 5, 81–102.

### **Состав вина (k-Nearest Neighbors)**

Forina, M., et al. (1998). Wine Recognition Data [Data file and description]. Retrieved from <http://archive.ics.uci.edu/ml/datasets/Wine>

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling Wine Preferences by Data Mining from Physicochemical Properties. *Decision Support Systems*, 47(4), 547–553.

### **Сердечно-сосудистые заболевания (Support Vector Machine)**

Robert Detrano (M.D., Ph.D), from Virginia Medical Center, Long Beach and Cleveland Clinic Foundation (1988). Heart Disease Database



(Cleveland) [Data file and description]. Retrieved from <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>

Detrano, R., et al. (1989). International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease. *The American Journal of Cardiology*, 64(5), 304–310.

### **Выжившие на «Титанике» (Decision Tree)**

British Board of Trade Inquiry (1990). Titanic Data [Data file and description]. Retrieved from <http://www.public.iastate.edu/~hofmann/data/titanic.html>

Report on the Loss of the 'Titanic' (S.S.) (1990). British Board of Trade Inquiry Report (reprint), Gloucester, UK: Allan Sutton Publishing and are discussed in Dawson, R. J. M. (1995). The 'Unusual Episode' Data Revisited. *Journal of Statistics Education*, 3(3).

### **Преступность в Сан-Франциско (Random Forest)**

SF OpenData, City and County of San Francisco (2016). Crime Incidents [Data]. Retrieved from <https://data.sfgov.org/Public-Safety/Map-Crime-Incidents-from-1-Jan-2003/gxxq-x39z>

### **Погода в Сан-Франциско (Random Forest)**

National Oceanic and Atmospheric Administration, National Centers for Environmental Information (2016). Quality Controlled Local Climatological Data (QCLCD) [Data file and description]. Retrieved from <https://www.ncdc.noaa.gov/qclcd/QCLCD?prior=N>

### **Рукописные цифры (Neural Networks)**

LeCun, Y., & Cortes, C. (1998). The MNIST Database of Handwritten Digits [Data file and description]. Retrieved from <http://yann.lecun.com/exdb/mnist>

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradientbased Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Lichman, M. (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Retrieved from <http://archive.ics.uci.edu/ml>

## Литература на русском языке

*Бринк Хенрик, Ричардс Джозеф, Феверолф Марк.* Машинное обучение. — СПб.: Питер, 2018. — 336 с.: ил. — (Серия «Библиотека программиста»).

*Бхаргава А.* Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. — СПб.: Питер, 2018. — 288 с.: ил. — (Серия «Библиотека программиста»).

*Винстон Уэйн.* Бизнес-моделирование и анализ данных. Решение актуальных задач с помощью Microsoft Excel. 5-е издание. — СПб.: Питер, 2018. — 864 с.: ил.

*Клеппман М.* Высоконагруженные приложения. Программирование, масштабирование, поддержка. — СПб.: Питер, 2019. — 640 с.: ил. — (Серия «Бестселлеры O'Reilly»).

*Николенко С., Кадурын А., Архангельская Е.* Глубокое обучение. — СПб.: Питер, 2018. — 480 с.: ил. — (Серия «Библиотека программиста»).

*Плас Дж. Вандер.* Python для сложных задач: наука о данных и машинное обучение. — СПб.: Питер, 2018. — 576 с.: ил. — (Серия «Бестселлеры O'Reilly»).

*Седжвик Р., Уэйн К.* Computer Science: основы программирования на Java, ООП, алгоритмы и структуры данных. — СПб.: Питер, 2018. — 1072 с.: ил. — (Серия «Классика computer science»).

*Силен Дэви, Мейсман Арно, Али Мохамед.* Основы Data Science и Big Data. Python и наука о данных. — СПб.: Питер, 2018. — 336 с.: ил. — (Серия «Библиотека программиста»).

*Феррейра Фило Владстон.* Теоретический минимум по Computer Science. Все, что нужно программисту и разработчику. — СПб.: Питер, 2019. — 224 с.: ил. — (Серия «Библиотека программиста»).

*Шолле Франсуа.* Глубокое обучение на Python. — СПб.: Питер, 2018. — 400 с.: ил. — (Серия «Библиотека программиста»).

*Шолле Франсуа.* Глубокое обучение на R. — СПб.: Питер, 2018. — 400 с.: ил. — (Серия «Библиотека программиста»).

## Об авторах

*Анналин Бл* закончила Мичиганский университет г. Анн-Арбор, где она также была тьютором студенческой группы по статистике. После этого она получила магистерскую степень (MPhil) в Центре психометрии Кембриджского университета, где собирала данные из социальных сетей для таргетированной рекламы и разрабатывала когнитивные тесты для приема на работу. После этого Disney Research пригласил ее в группу поведенческих исследований, где Анналин анализировала психологические портреты потребителей.

*Кеннет Су* получил магистерскую степень (MS) по статистике в Стэнфордском университете. До этого он на протяжении трех лет был лучшим студентом своей группы по курсу «Математика, исследование операций, статистика и экономика» (MORSE) Уорикского университета. Кеннет также работал там научным ассистентом в составе научной группы по исследованию операций и методов управления, занимаясь устойчивой двухкритериальной задачей оптимизации сетей, подверженных случайным сбоям.

---

Анналин и Кеннет долгое время работали в министерстве обороны Сингапура.

Вам понравилась эта книга?

Отметьте ее пятью звездочками на Amazon: <http://getbook.at/numsense>.

Посетите наш блог [www.algobbeans.com](http://www.algobbeans.com). Там есть и другие доступные материалы по Data Science.

*Анналин Ын, Кеннет Су*

**Теоретический минимум по Big Data.**  
**Всё, что нужно знать о больших данных**

*Перевел с английского А. Тимохин*

Заведующая редакцией  
Ведущий редактор  
Литературный редактор  
Художественный редактор  
Корректоры  
Верстка

*Ю. Сергиенко  
К. Тульцева  
А. Бульченко  
Г. Макаров-Якубовский  
С. Беляева, Г. Шкатова  
Л. Егорова*

Изготовлено в России. Изготовитель: ООО «Прогресс книга».  
Место нахождения и фактический адрес: 194044, Россия, г. Санкт-Петербург,  
Б. Сампсониевский пр., д. 29А, пом. 52. Тел.: +78127037373.

Дата изготовления: 01.2019. Наименование: книжная продукция.

Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014, 58.11.12 —

Книги печатные профессиональные, технические и научные.

Импортер в Беларусь: ООО «ПИТЕР М», 220020, РБ, г. Минск, ул. Тимирязева,  
д. 121/3, к. 214, тел./факс: 208 80 01.

Подписано в печать 17.01.19. Формат 60×90/16. Бумага офсетная. Усл. п. л. 13,000.

Доп. тираж. Заказ 0000.



**ИЗДАТЕЛЬСКИЙ ДОМ «ПИТЕР» предлагает профессиональную, популярную и детскую развивающую литературу**

**Заказать книги оптом можно в наших представительствах**

## **РОССИЯ**

**Санкт-Петербург:** м. «Выборгская», Б. Сампсониевский пр., д. 29а  
тел./факс: (812) 703-73-83, 703-73-72; e-mail: sales@piter.com

**Москва:** м. «Электrozаводская», Семеновская наб., д. 2/1, стр. 1, 6 этаж  
тел./факс: (495) 234-38-15; e-mail: sales@msk.piter.com

**Воронеж:** тел.: 8 951 861-72-70; e-mail: hitsenko@piter.com

**Екатеринбург:** ул. Толедова, д. 43а; тел./факс: (343) 378-98-41, 378-98-42;  
e-mail: office@ekat.piter.com; skype: ekat.manager2

**Нижний Новгород:** тел.: 8 930 712-75-13; e-mail: yashny@yandex.ru; skype: yashny1

**Ростов-на-Дону:** ул. Ульяновская, д. 26  
тел./факс: (863) 269-91-22, 269-91-30; e-mail: piter-ug@rostov.piter.com

**Самара:** ул. Молодогвардейская, д. 33а, офис 223  
тел./факс: (846) 277-89-79, 277-89-66; e-mail: pitvolga@mail.ru,  
pitvolga@samara-ttk.ru

## **БЕЛАРУСЬ**

**Минск:** ул. Розы Люксембург, д. 163; тел./факс: +37 517 208-80-01, 208-81-25;  
e-mail: og@minsk.piter.com

**Издательский дом «Питер» приглашает к сотрудничеству авторов:**

тел./факс: (812) 703-73-72, (495) 234-38-15; e-mail: ivanovaa@piter.com  
Подробная информация здезь: <http://www.piter.com/page/avtoru>

**Издательский дом «Питер» приглашает к сотрудничеству зарубежных торговых партнеров или посредников, имеющих выход на зарубежный рынок:** тел./факс: (812) 703-73-73; e-mail: sales@piter.com

---

**Заказ книг для вузов и библиотек:**

тел./факс: (812) 703-73-73, гоб. 6243; e-mail: uchebnik@piter.com

---

**Заказ книг по почте:** на сайте [www.piter.com](http://www.piter.com); тел.: (812) 703-73-74, гоб. 6216;  
e-mail: books@piter.com

---





**Вопросы по продаже электронных книг:** тел.: (812) 703-73-74, гоб. 6217;  
e-mail: kuznetsov@piter.com



**ЗАКАЗАТЬ КНИГИ ИЗДАТЕЛЬСКОГО ДОМА «ПИТЕР»  
МОЖНО ЛЮБЫМ УДОБНЫМ ДЛЯ ВАС СПОСОБОМ:**

- на нашем сайте: **www.piter.com**
- по электронной почте: **books@piter.com**
- по телефону: **(812) 703-73-74**

**ВЫ МОЖЕТЕ ВЫБРАТЬ ЛЮБОЙ УДОБНЫЙ ДЛЯ ВАС СПОСОБ ОПЛАТЫ:**

-  Наложным платежом с оплатой при получении в ближайшем почтовом отделении.
-  С помощью банковской карты. Во время заказа вы будете перенаправлены на защищенный сервер нашего оператора, где сможете ввести свои данные для оплаты.
-  Электронными деньгами. Мы принимаем к оплате Яндекс.Деньги, Webmoney и Kiwi-кошелек.
-  В любом банке, распечатав квитанцию, которая формируется автоматически после совершения вами заказа.

**ВЫ МОЖЕТЕ ВЫБРАТЬ ЛЮБОЙ УДОБНЫЙ ДЛЯ ВАС СПОСОБ ДОСТАВКИ:**

- Псылки отправляются через «Почту России». Отработанная система позволяет нам организовывать доставку ваших покупок максимально быстро. Дату отправления вашей покупки и дату доставки вам сообщат по e-mail.
- Вы можете оформить курьерскую доставку своего заказа (более подробную информацию можно получить на нашем сайте [www.piter.com](http://www.piter.com)).
- Можно оформить доставку заказа через почтоматы (адреса почтоматов можно узнать на нашем сайте [www.piter.com](http://www.piter.com)).

**ПРИ ОФОРМЛЕНИИ ЗАКАЗА УКАЖИТЕ:**

- фамилию, имя, отчество, телефон, e-mail;
- почтовый индекс, регион, район, населенный пункт, улицу, дом, корпус, квартиру;
- название книги, автора, количество заказываемых экземпляров.

**БЕСПЛАТНАЯ ДОСТАВКА:**

- курьером по Москве и Санкт-Петербургу при заказе на сумму **от 2000 руб.**
- почтой России при предварительной оплате заказа на сумму **от 2000 руб.**