

Оглавление

Предисловие	11
Благодарности	12
О книге	14
Структура книги	14
Для кого написана эта книга	15
Условные обозначения и загружаемые файлы	15
Об авторах	16
От издательства	17
Глава 1. Data science в мире больших данных	18
1.1. Область применения data science и больших данных и их преимущества . . .	19
1.2. Грани данных	21
1.2.1. Структурированные данные	21
1.2.2. Неструктурированные данные	22
1.2.3. Данные на естественном языке.	22
1.2.4. Машинные данные	23
1.2.5. Графовые, или сетевые, данные	24
1.2.6. Аудио, видео и графика	25
1.2.7. Поточковые данные	26
1.3. Процесс data science	26
1.3.1. Назначение цели исследования	27
1.3.2. Сбор данных.	27
1.3.3. Подготовка данных.	27
1.3.4. Исследование данных.	27

1.3.5. Моделирование данных или построение модели	27
1.3.6. Отображение и автоматизация	28
1.4. Экосистема больших данных и data science	28
1.4.1. Распределенные файловые системы	30
1.4.2. Инфраструктура распределенного программирования.	30
1.4.3. Инфраструктура интеграции данных.	31
1.4.4. Инфраструктуры машинного обучения	31
1.4.5. Базы данных NoSQL	32
1.4.6. Инструменты планирования	33
1.4.7. Инструменты сравнительного анализа.	33
1.4.8. Развертывание системы	33
1.4.9. Программирование служб	34
1.4.10. Безопасность	34
1.5. Вводный пример использования Hadoop	34
1.6. Итоги	40

Глава 2. Процесс data science 42

2.1. Обзор процесса data science	42
2.1.1. Не будьте рабом процесса	45
2.2. Этап 1: Определение целей исследования и создание проектного задания	46
2.2.1. Выделите время на то, чтобы разобраться в целях и контексте исследования	46
2.2.2. Создайте проектное задание.	47
2.3. Этап 2: Сбор данных	47
2.3.1. Начните с данных, хранимых в компании	48
2.3.2. Не бойтесь покупок во внешних источниках	49
2.3.3. Проверьте качество данных сейчас, чтобы предотвратить проблемы в будущем	50
2.4. Этап 3: Очистка, интеграция и преобразование данных	50
2.4.1. Очистка данных	51
2.4.2. Исправляйте ошибки как можно раньше	58
2.4.3. Комбинирование данных из разных источников	59
2.4.4. Преобразование данных	62
2.5. Этап 4: Исследовательский анализ данных	66
2.6. Этап 5: Построение моделей	70
2.6.1. Выбор модели и переменных	71
2.6.2. Выполнение модели	72

2.6.3. Диагностика и сравнение моделей	77
2.7. Этап 6: Представление результатов и построение приложений на их основе	78
Итоги	79

Глава 3. Машинное обучение 81

3.1. Что такое машинное обучение, и почему оно важно для вас?	82
3.1.1. Применение машинного обучения в data science	83
3.1.2. Применение машинного обучения в процессе data science.	84
3.1.3. Инструменты Python, используемые в машинном обучении	85
3.2. Процесс моделирования	87
3.2.1. Создание новых показателей и выбор модели	88
3.2.2. Тренировка модели	89
3.2.3. Проверка адекватности модели	90
3.2.4. Прогнозирование новых наблюдений	91
3.3. Типы машинного обучения	92
3.3.1. Контролируемое обучение	92
3.3.2. Неконтролируемое обучение	100
3.4. Частично контролируемое обучение	111
3.5. Итоги	112

Глава 4. Работа с большими данными на одном компьютере 114

4.1. Проблемы при работе с большими объемами данных	115
4.2. Общие методы обработки больших объемов данных	116
4.2.1. Правильный выбор алгоритма.	117
4.2.2. Правильный выбор структуры данных.	126
4.2.3. Правильный выбор инструментов	128
4.3. Общие рекомендации для программистов при работе с большими наборами данных	131
4.3.1. Не повторяйте уже выполненную работу	131
4.3.2. Используйте все возможности оборудования	132
4.3.3. Экономьте вычислительные ресурсы.	133
4.4. Пример 1: Прогнозирование вредоносных URL-адресов	134
4.4.1. Этап 1: Определение цели исследования	134
4.4.2. Этап 2: Сбор данных URL	135
4.4.3. Этап 4: Исследование данных.	136
4.4.4. Этап 5: Построение модели	137

4.5. Пример 2: Построение рекомендательной системы внутри базы данных	139
4.5.1. Необходимые инструменты и методы	139
4.5.2. Этап 1: Вопрос исследования	142
4.5.3. Этап 3: Подготовка данных	142
4.5.4. Этап 5: Построение модели	147
4.5.5. Этап 6: Отображение и автоматизация	148
4.6. Итоги	150

Глава 5. Первые шаги в области больших данных 151

5.1. Распределение хранения и обработки данных в инфраструктурах	152
5.1.1. Nadoop: инфраструктура для хранения и обработки больших объемов данных	152
5.1.2. Spark: замена MapReduce с повышенной производительностью	156
5.2. Учебный пример: Оценка риска при кредитовании	157
5.2.1. Этап 1: Цель исследования	159
5.2.2. Этап 2: Сбор данных	160
5.2.3. Этап 3: Подготовка данных	164
5.2.4. Этап 4: Исследование данных и Этап 6: построение отчета	169
5.3. Итоги	182

Глава 6. Присоединяйтесь к движению NoSQL 183

6.1. Введение в NoSQL	186
6.1.1. ACID: базовые принципы реляционных баз данных	186
6.1.2. Теорема CAP: проблема баз данных, распределенных по многим узлам	187
6.1.3. Принципы BASE баз данных NoSQL	190
6.1.4. Типы баз данных NoSQL	192
6.2. Учебный пример: Диагностика болезней	199
6.2.1. Этап 1: Назначение цели исследования	201
6.2.2. Этапы 2 и 3: Сбор и подготовка данных	202
6.2.3. Этап 4: Исследование данных	211
6.2.4. Этап 3 (снова): Подготовка данных для профилирования болезни	220
6.2.5. Этап 4 (повторно): Исследование данных для профилирования болезни	223
6.2.6. Этап 6: Отображение и автоматизация	224
6.3. Итоги	226

Глава 7. Графовые базы данных 227

7.1. Связанные данные и графовые базы данных	227
7.1.1. Когда и почему используются графовые базы данных?	231
7.2. Neo4j: графовая база данных	234
7.2.1. Cypher: язык запросов к графам	235
7.3. Пример использования связанных данных: рекомендательная система	242
7.3.1. Этап 1: Определение цели исследования	242
7.3.2. Этап 2: Сбор данных.	244
7.3.3. Этап 3: Подготовка данных.	245
7.3.4. Этап 4: Исследование данных.	248
7.3.5. Этап 5: Моделирование данных	251
7.3.6. Этап 6: Отображение	254
7.4. Итоги	255

Глава 8. Глубокий анализ текста 257

8.1. Глубокий анализ текста в реальном мире	259
8.2. Методы глубокого анализа текста	263
8.2.1. Набор слов.	264
8.2.2. Выделение основы и лемматизация	266
8.2.3. Классификатор на базе дерева принятия решений	267
8.3. Учебный пример: классификация сообщений Reddit	269
8.3.1. NLTK.	270
8.3.2. Обзор процесса data science и этап 1: назначение цели исследования	272
8.3.3. Этап 2: Сбор данных.	273
8.3.4. Этап 3: Подготовка данных.	277
8.3.5. Этап 4: Исследование данных.	280
8.3.6. Этап 3 (повторно): Подготовка данных (адаптированная)	283
8.3.7. Этап 5: Анализ данных	287
8.3.8. Этап 6: Отображение и автоматизация	291
8.4. Итоги	293

Глава 9. Визуализация данных для конечного пользователя 295

9.1. Способы визуализации данных	296
9.2. Crossfilter, библиотека MapReduce для JavaScript	300
9.2.1. Подготовка необходимых компонентов	300
9.2.2. Использование Crossfilter для фильтрации набора данных.	305

9.3. Создание информационной панели с использованием dc.js	309
9.4. Средства разработки	315
9.5. Итоги	317
Приложение А. Настройка Elasticsearch	319
A.1. Установка в Linux	319
A.2. Установка в Windows	321
Приложение Б. Установка Neo4j	325
Б.1. Установка в Linux	325
Б.2. Установка в Windows	326
Приложение В. Установка сервера MySQL	328
В.1. Установка в Windows	328
В.2. Установка в Linux	330
Приложение Г. Установка Anaconda в виртуальной среде	332
Г.1. Установка в Linux	332
Г.2. Установка в Windows	332
Г.3. Настройка среды	333

2.5. Этап 4: Исследовательский анализ данных

В фазе исследовательского анализа данных происходит углубленное изучение данных (рис. 2.13). В графическом виде информация воспринимается намного проще, поэтому для понимания данных и взаимодействий переменных применяются в основном графические методы. Целью этой фазы является исследование данных, поэтому в фазе исследовательского анализа данных необходимо сохранять объективность и смотреть в оба. Очистка данных непосредственной целью не является, однако в этой фазе нередко обнаруживаются аномалии, упущенные ранее; в таком случае отступите на шаг назад и исправьте их.

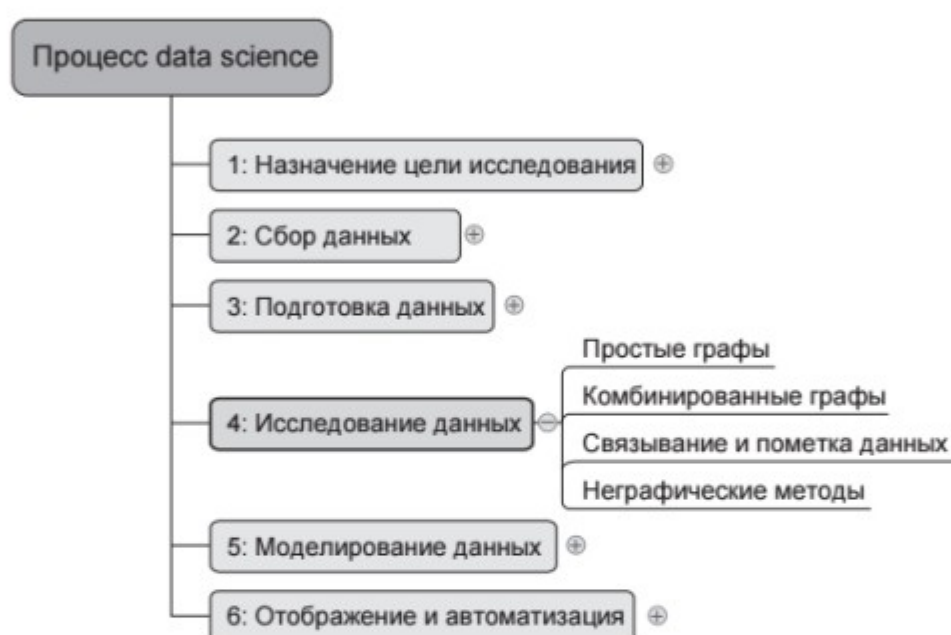


Рис. 2.13. Этап 4: Исследование данных

В этой фазе применяется широкий спектр методов визуализации, от простых графиков или столбцовых диаграмм, как на рис. 2.14, до более сложных диаграмм Сэнки и сетевых графов. Иногда бывает полезно составить из нескольких простых диаграмм одну сложную, чтобы еще лучше разобраться в сути данных. Также возможно построение анимированных или интерактивных диаграмм — с такими диаграммами работать проще (и, откровенно говоря, гораздо интереснее). Пример интерактивной диаграммы Сэнки доступен по адресу <http://bost.ocks.org/mike/sankey/>.

Майк Босток приводит интерактивные примеры почти для всех разновидностей диаграмм. Его сайт заслуживает внимания, хотя большинство примеров ориентировано скорее на отображение данных, нежели на их исследование.

Объединение этих диаграмм еще лучше раскрывает суть данных (рис. 2.15).

Наложение диаграмм также часто применяется на практике. На рис. 2.16 несколько простых диаграмм объединяются в диаграмму Парето («диаграмма 80/20»).

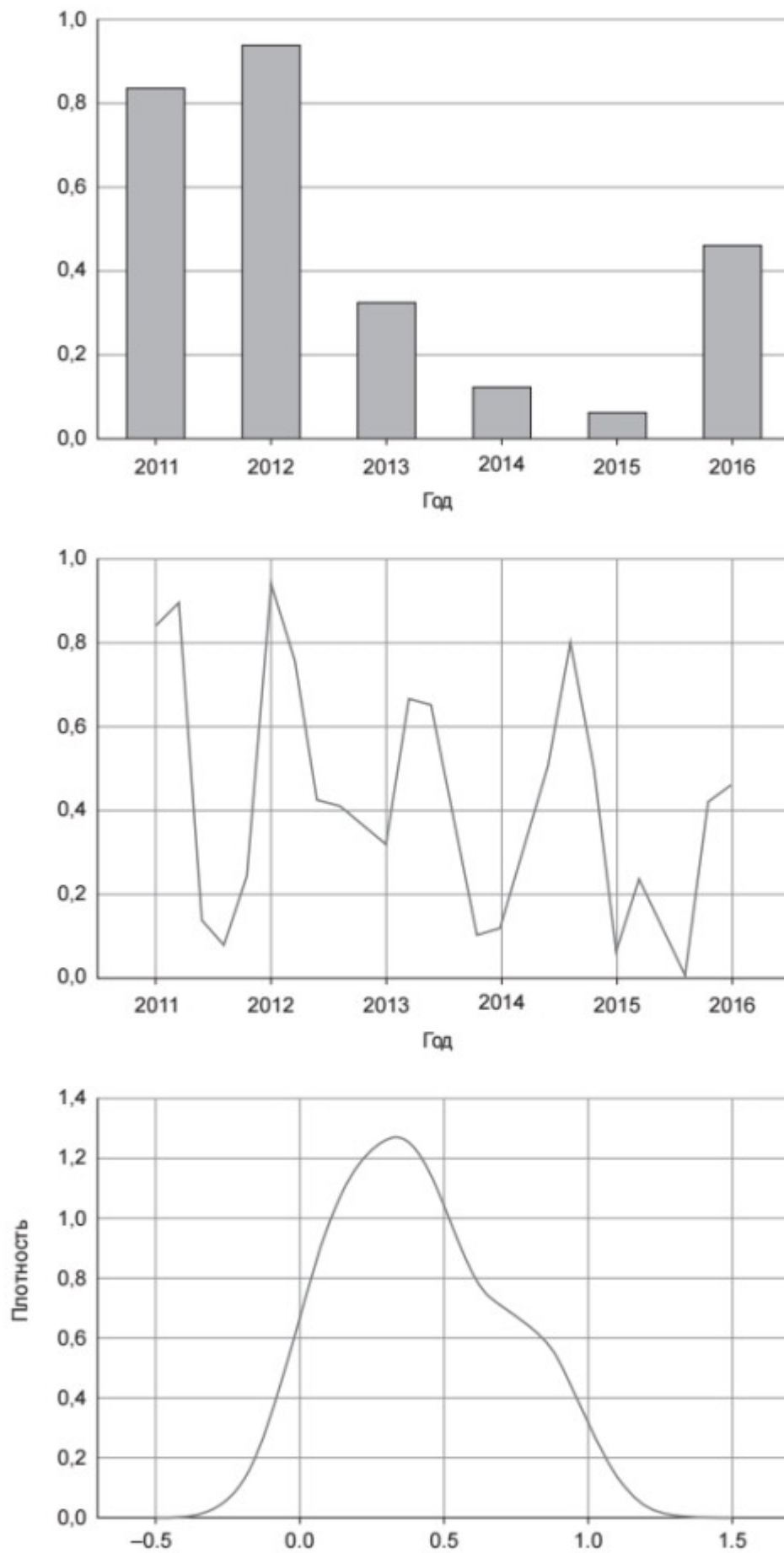


Рис. 2.14. Сверху вниз: столбцовая диаграмма, линейный график, кривая распределения — примеры диаграмм, используемых в исследовательском анализе

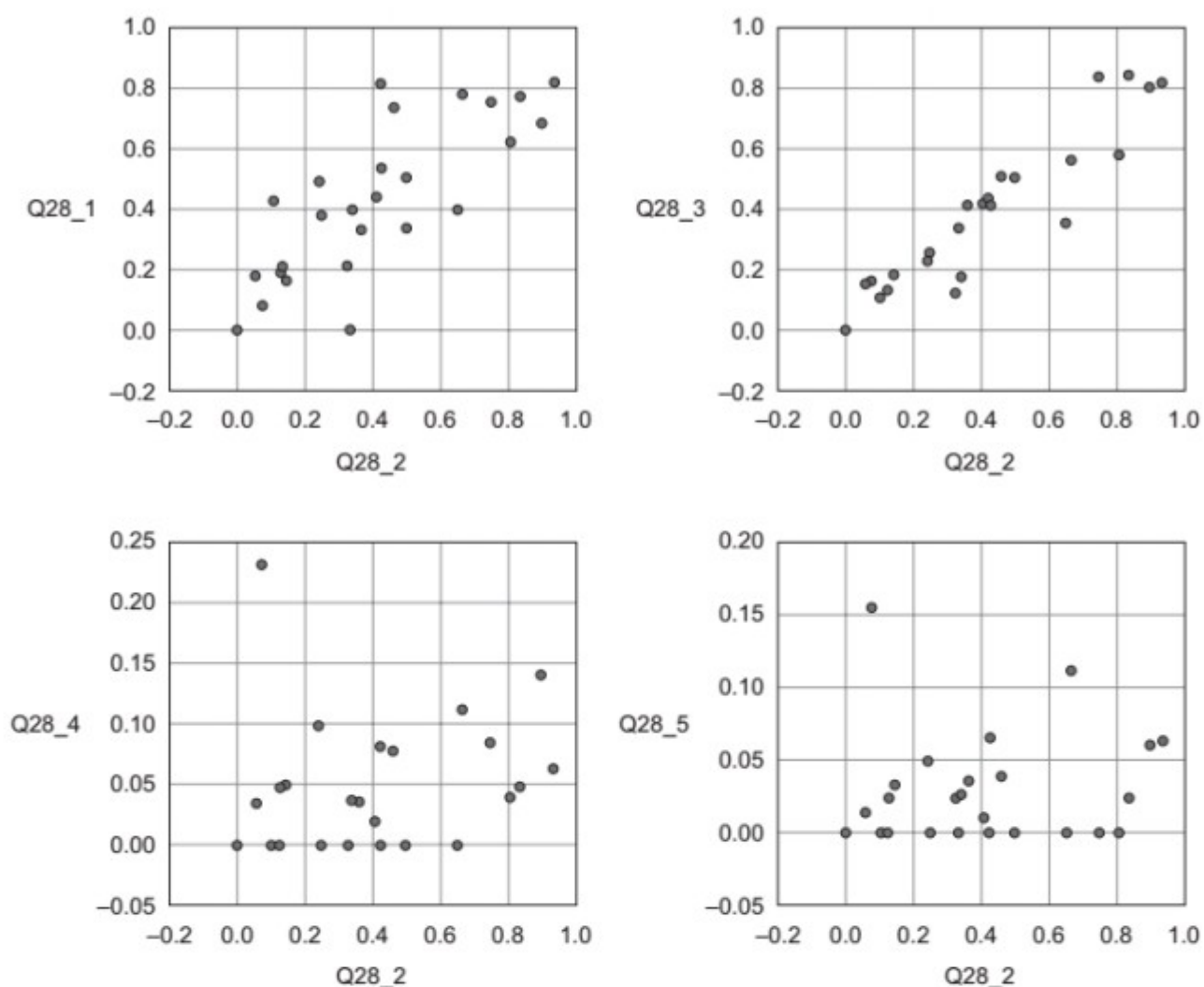


Рис. 2.15. Размещение диаграмм вблизи друг от друга помогает лучше понять структуру данных с несколькими переменными

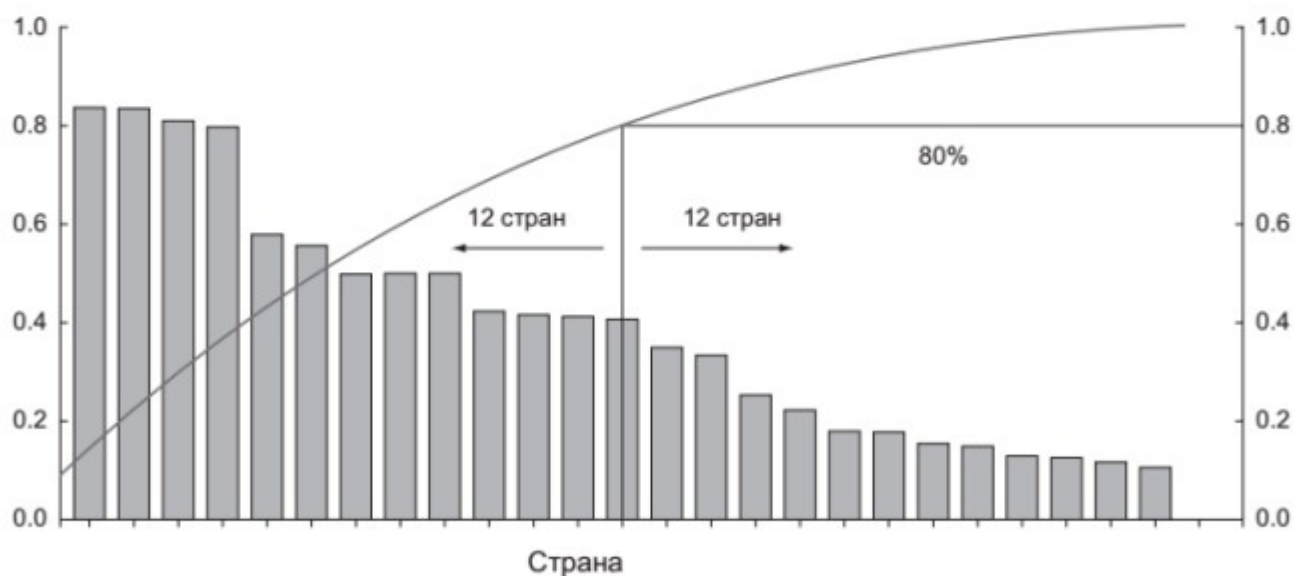


Рис. 2.16. Диаграмма Парето представляет собой комбинацию значений и кумулятивного распределения. Диаграмма наглядно показывает, что на первые 50% стран приходится чуть менее 80% общего вклада. Если бы на диаграмме была представлена покупательная способность, а вы занимались продажей дорогостоящих товаров, вероятно, тратить маркетинговый бюджет на все страны было бы неэффективно, разумнее начать с первых 50%

На рис. 2.17 представлен другой метод: *связывание и пометка данных* (brushing and linking). Разные диаграммы и таблицы (или представления) объединяются и связываются таким образом, что изменения в одной диаграмме автоматически переносятся на другие. Нетривиальный пример такого рода приведен в главе 9. Подобные интерактивные исследования данных упрощают выявление новых глубинных причин и взаимосвязей.

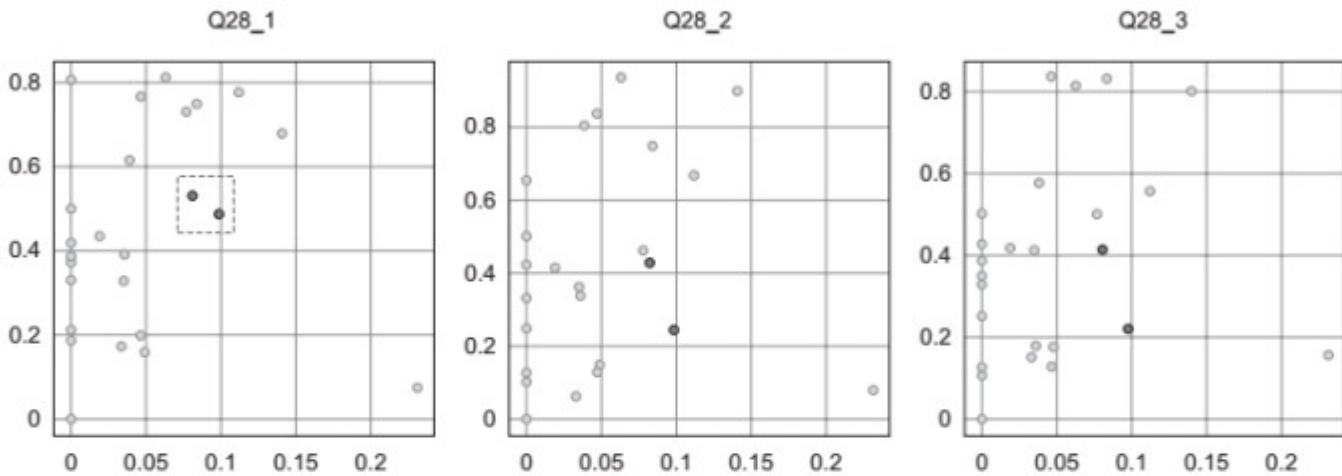


Рис. 2.17. Метод связывания и пометки данных позволяет выбрать наблюдения на одной диаграмме с выделением тех же наблюдений на другой диаграмме

На рис. 2.17 представлены средние баллы по странам. Диаграмма не только обозначает высокую степень корреляции между ответами, но и позволяет увидеть, что при выделении нескольких точек на одной диаграмме эти точки соответствуют похожим точкам на других диаграммах. В данном случае выделенные точки на левой диаграмме соответствуют точкам на средней и правой диаграммах, хотя между средней и правой диаграммами это соответствие более очевидно.

Две другие важные разновидности диаграмм — гистограмма на рис. 2.18 и коробчатая диаграмма на рис. 2.19.

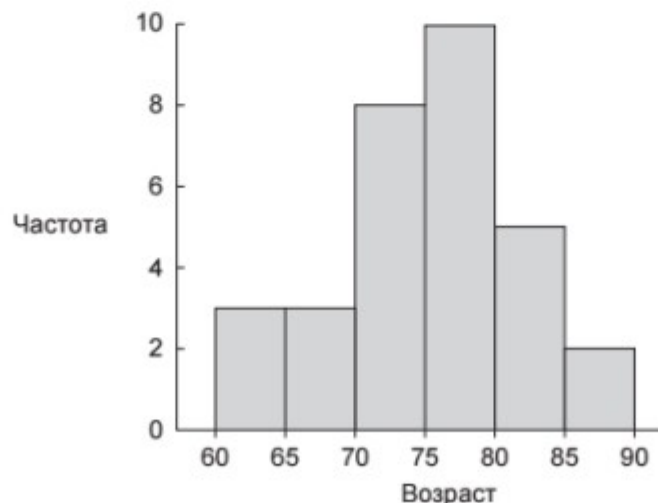


Рис. 2.18. Пример гистограммы: численность людей в возрастных группах с интервалом в 5 лет

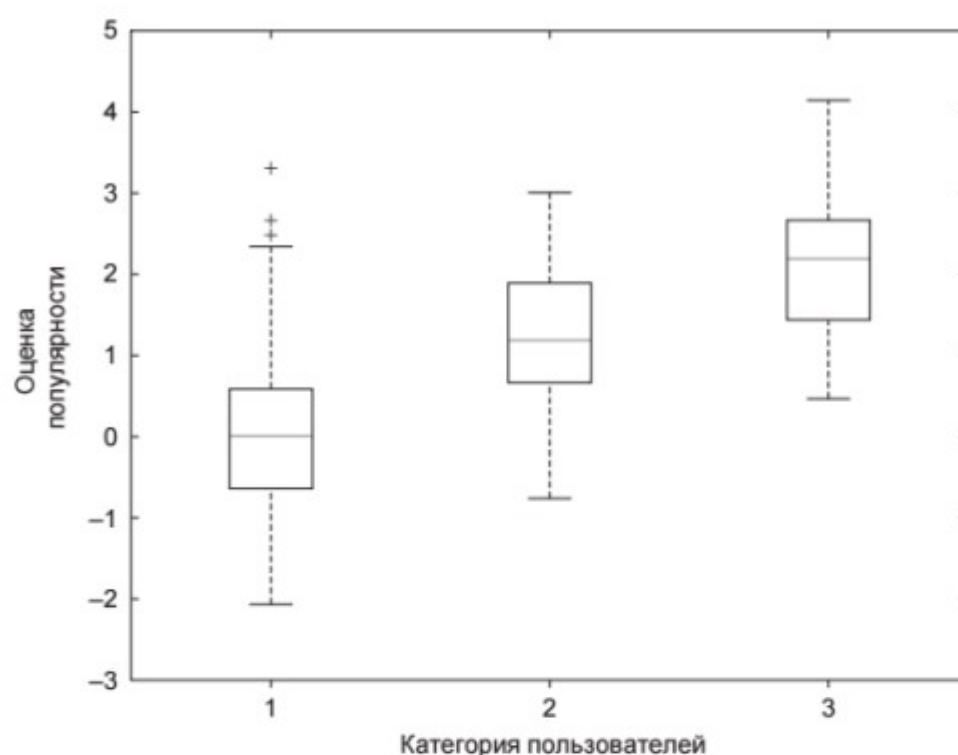


Рис. 2.19. Пример коробчатой диаграммы: у каждой категории пользователей существует распределение оценок, выставленных за определенное изображение на сайте фотографий

На гистограмме переменная делится на дискретные категории, количества вхождений в каждую категорию суммируются и отображаются на диаграмме. С другой стороны, коробчатая диаграмма не показывает количество наблюдений, но дает представление о распределении внутри категорий. На ней могут одновременно отображаться минимум, максимум, медиана и другие характеристики распределения.

Методы, упомянутые в этой фазе, в основном имеют визуальную природу, но на практике анализ не ограничивается методами визуализации. Сведение в таблицы, кластеризация и другие методы моделирования также могут быть частью исследовательского анализа. Даже построение простых моделей может быть частью этого шага.

Итак, фаза исследования данных завершена, а вы получили хорошее представление о своих данных. Пора переходить к следующей фазе: построению моделей.

2.6. Этап 5: Построение моделей

При наличии очищенных данных и хорошем понимании контента вы готовы к построению моделей с целью улучшения прогнозов, проведения классификации объектов или лучшего понимания моделируемой системы. Эта фаза является намного более целенаправленной, чем этап исследовательского анализа, потому что вы знаете, что ищете и каким должен быть результат. На рис. 2.20 представлены основные компоненты построения модели.



Рис. 2.20. Этап 5: Моделирование данных

Методы, которые будут использоваться, позаимствованы из области машинного обучения, обработки/анализа данных и статистики. В этой главе рассматривается лишь малая часть существующих методов, в главе 3 они будут представлены более подробно. Давать здесь нечто большее концептуального введения значило бы выйти за рамки книги, но и этой информации достаточно для начала; 20% методов помогут вам в 80% случаев, потому что методы перекрываются в отношении предполагаемой цели. Часто они достигают своих целей в основном похожими, но немного различающимися способами.

Построение модели является итеративным процессом. Способ построения модели зависит от того, принадлежите ли вы к школе классической статистики или же к несколько более современной школе машинного обучения, а также от типа применяемого метода. В любом случае процесс построения большинства моделей состоит из следующих шагов:

1. Выбор метода моделирования и переменных для включения в модель.
2. Выполнение модели.
3. Диагностика и сравнение моделей.

2.6.1. Выбор модели и переменных

Вам нужно выбрать переменные, которые должны быть включены в модель, и метод моделирования. Результаты, полученные в ходе исследовательского анализа, должны были уже дать достаточно четкое представление о том, какие переменные позволят построить хорошую модель. Известно много методов моделирования, и выбор правильной модели для задачи — ваша обязанность. Вы должны учесть

качество модели и соответствие проекта всем требованиям для использования модели, а также другие факторы:

- ❑ Должна ли модель быть вынесена в производственную среду, и, если должна, насколько просто она будет реализовываться?
- ❑ С какими трудностями связано сопровождение модели: долго ли она останется актуальной, если не менять ее?
- ❑ Должна ли модель быть простой для объяснения?

Когда предварительные размышления будут завершены, наступает время действовать.

2.6.2. Выполнение модели

После того как модель будет выбрана, ее необходимо реализовать в программном коде.

ПРИМЕЧАНИЕ

Здесь мы впервые будем заниматься выполнением кода Python, поэтому убедитесь в том, что виртуальная среда настроена и готова к использованию. Умение настраивать виртуальную среду относится к числу обязательных навыков, но если вы занимаетесь этим впервые, обратитесь к приложению Г. Весь код этой главы можно загрузить по адресу <https://www.manning.com/books/introducing-data-science>. К этой главе прилагаются файлы `ipython (.ipynb)` и `Python (.py)`.

К счастью, в большинстве языков программирования (таких, как Python) уже существуют специализированные библиотеки, например `StatsModels` или `Scikit-learn`. Эти пакеты поддерживают многие популярные методы моделирования. Программирование модели во многих случаях является делом нетривиальным, так что наличие таких библиотек ускорит процесс. Как видно из следующего кода, использовать линейную регрессию (рис. 2.21) с `StatsModels` или `Scikit-learn` до-

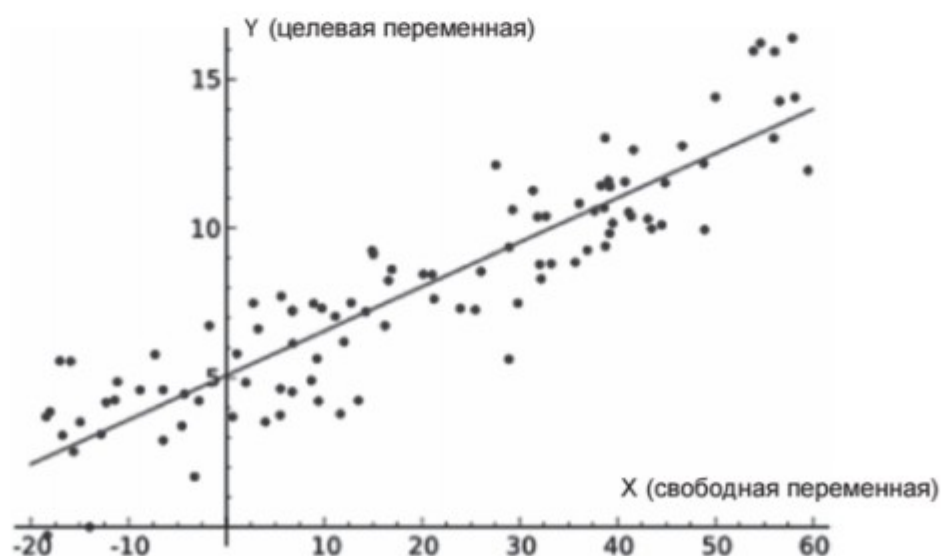


Рис. 2.21. Метод линейной регрессии пытается подобрать линию с минимальным расстоянием до каждой точки

статочно просто. Самостоятельная реализация потребует существенно больших усилий даже для простых методов. В листинге 2.1 приведен пример выполнения модели линейного прогнозирования на программном уровне.

Листинг 2.1. Выполнение модели линейного прогнозирования для полуслучайных данных

```
import statsmodels.api as sm      | Импортирование необходимых
import numpy as np              | модулей Python.
predictors = np.random.random(1000).reshape(500,2)
target = predictors.dot(np.array([0.4, 0.6])) + np.random.random(500)
lmRegModel = sm.OLS(target,predictors)
result = lmRegModel.fit()
result.summary() ← Вывод статистики
                  | соответствия модели.
```

Подбор линейной регрессии для данных.

Создание случайных данных для свободных (x) и целевых переменных (y) модели. Прогностические параметры используются для создания целевых значений, чтобы создать корреляцию.

Ладно, мы здесь смухлевали, и довольно основательно. Мы создали свободные значения, которые вроде бы должны прогнозировать поведение целевых переменных. Линейная регрессия предполагает линейное отношение между x (свободная переменная) и y (целевая переменная) (см. рис. 2.21).

Однако при этом мы создали целевую переменную на основании значения независимой, добавив к ней небольшую долю случайности. Разумеется, в результате получилась модель с высокой степенью соответствия. Вызов `results.summary()` выводит таблицу на рис. 2.22. Разумеется, точные результаты зависят от сгенерированных случайных значений.

Пока не будем обращать внимание на большую часть выводимых данных и сосредоточимся на самом важном:

- *Степень соответствия модели* — для оценки степени соответствия используется коэффициент детерминации (R-квадрат) или скорректированный (adjusted) коэффициент детерминации. Эта метрика обозначает степень разброса данных, отраженного в модели. Разность между скорректированным и простым коэффициентом детерминации в данном случае минимальна, потому что скорректированный коэффициент равен сумме простого коэффициента и штрафа за сложность модели. Модель усложняется с введением большого количества переменных. При наличии простой модели сложная модель не нужна, поэтому скорректированный коэффициент детерминации «наказывает» за излишнее усложнение. В любом случае значение 0,893 достаточно большое (так и должно быть, потому что мы смухлевали). Существуют разные эмпирические правила, но для экономических моделей значения свыше 0,85 обычно считаются хорошими. Если вы хотите однозначной победы, потребуются значения свыше 0,90. Впрочем, в ходе исследований часто встречаются модели с очень низкой степенью соответствия (даже $<0,2$). В данном случае важнее влияние введенных свободных переменных.

Dep. Variable:	y	R-squared:	0.893
Model:	OLS	Adj. R-squared:	0.893
Method:	Least Squares	F-statistic:	2088.
Date:	Fri, 30 Oct 2015	Prob (F-statistic):	7.13e-243
Time:	12:44:31	Log-Likelihood:	-176.74
No. Observations:	500	AIC:	357.5
Df Residuals:	498	BIC:	365.9
Df Model:	2		
Covariance Type:	nonrobust		

Степень соответствия модели данным: высокие значения лучше низких, но слишком высокие выглядят подозрительно.

P-значение сообщает, оказывает ли свободная переменная значимое влияние на целевую. Низкие значения предпочтительны, и значение <0,005 часто считается «значимым».

	coef	std err	t	P> t	[95.0% Conf. Int.]
x1	0.7658	0.040	19.130	0.000	0.687 0.844
x2	1.1252	0.039	28.603	0.000	1.048 1.202

Omnibus:	34.269	Durbin-Watson:	1.943
Prob(Omnibus):	0.000	Jarque-Bera (JB):	13.480
Skew:	-0.125	Prob(JB):	0.00118
Kurtosis:	2.235	Cond. No.	2.51

Коэффициенты линейного уравнения.
 $y = 0,7658 x_1 + 1,1252 x_2$.

Рис. 2.22. Выходная информация модели линейной регрессии

- *Коэффициенты свободных переменных* — в линейной модели эти значения интерпретируются легко. В нашем примере увеличение x_1 на 1 приводит к изменению y на 0,7658. Легко видеть, что удачный выбор свободной переменной может проложить путь к Нобелевской премии, даже если модель в целом никуда не годится. Если, например, вы определили, что некий ген является значимым фактором возникновения рака, эта информация крайне важна, даже если ген сам по себе не определяет, заболит человек раком или нет. В данном примере мы имеем дело с классификацией, а не регрессией, но суть остается неизменной: обнаружить влияние в научных исследованиях важнее (не говоря уже о реалистичности), чем найти модель с идеальным соответствием. Но как определить, что ген оказывает влияние? Характеристика для его оценки называется значимостью.
- *Значимость свободных переменных* — коэффициенты удобны и понятны, но в некоторых случаях не существует убедительных доказательств наличия влияния. Для оценки этой величины применяется *p-значение* (p-value). Здесь можно было бы долго объяснять, что такое ошибки 1-го и 2-го типа, но вкратце ситуация выглядит так: если p-значение меньше 0,05, то переменная обычно считается значимой. Откровенно говоря, значение выбрано произвольно — оно

указывает на то, что с 5%-ной вероятностью свободная переменная не оказывает влияния. Вас устраивает 5%-ная вероятность ошибки? Дело ваше. Некоторые специалисты вводили понятие чрезвычайно значимых ($p < 0,01$) и минимально значимых порогов ($p < 0,1$).

Линейная регрессия работает, если нужно спрогнозировать значение, но что, если потребуется провести классификацию? Тогда на помощь приходят *модели классификации*, из которых наибольшей известностью пользуется модель *k ближайших соседей*.

Как видно из рис. 2.23, модель *k ближайших соседей* ищет помеченные точки рядом с непомеченной и на основании полученных результатов прогнозирует, какой должна быть метка.

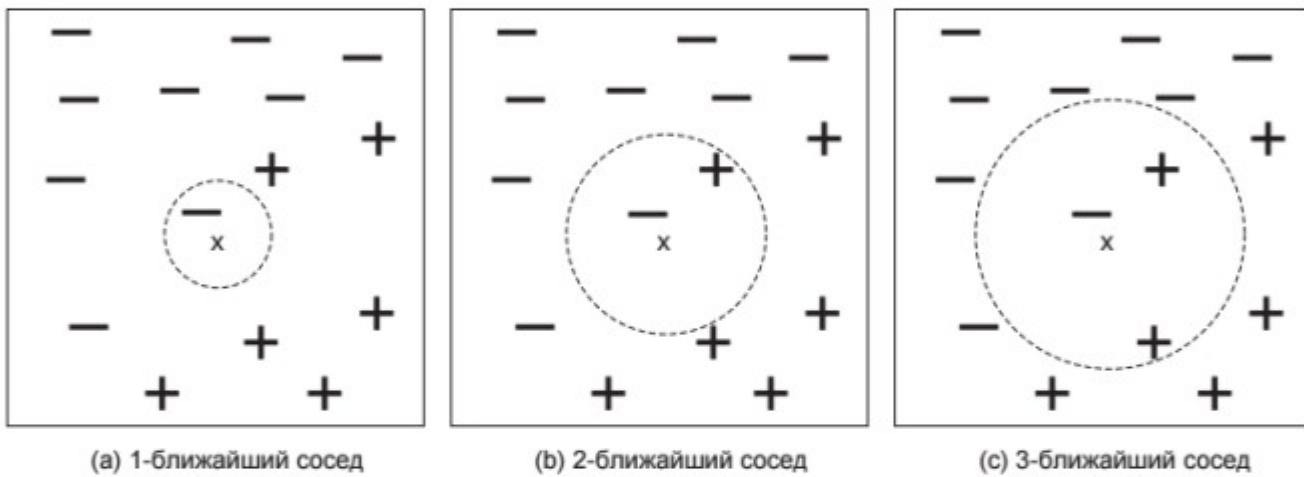


Рис. 2.23. Метод *k*-ближайших соседей проверяет до *k* ближайших точек для формирования прогноза

Попробуем применить этот метод в коде Python при помощи библиотеки Scikit, как показано в следующем листинге.

Листинг 2.2. Выполнение классификации методом *k* ближайших соседей для полуслучайных данных

```

from sklearn import neighbors ← Импортирование
predictors = np.random.random(1000).reshape(500,2) ← модулей.
target = np.around(predictors.dot(np.array([0.4, 0.6])) + ← Создание случайных
    np.random.random(500)) ← свободных данных
                                и полуслучайных
                                целевых данных на
                                основании свободных.
clf = neighbors.KNeighborsClassifier(n_neighbors=10) ← Классификация по модели
knn = clf.fit(predictors, target) ← 10 ближайших соседей.
knn.score(predictors, target) ← Получение метрики
                                соответствия модели:
                                какой процент
                                классификации был
                                правильным?

```