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.

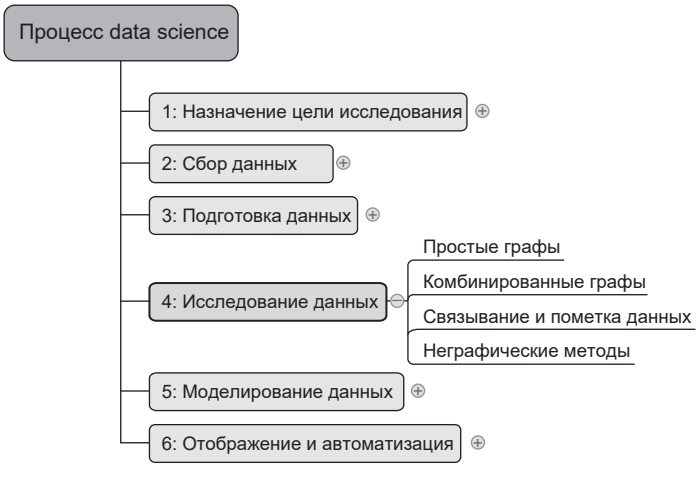


Рис. 2.13. Этап 4: Исследование данных

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 Бостоk 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»).

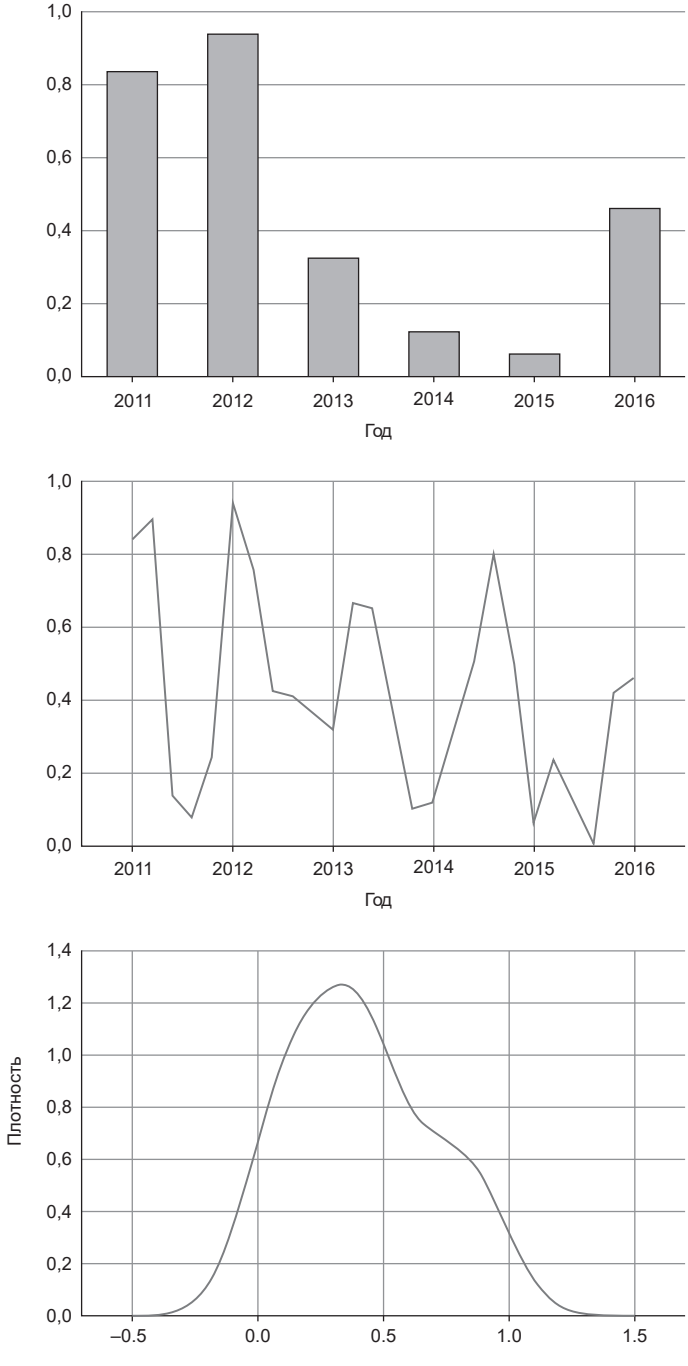


Рис. 2.14. Сверху вниз: столбцовая диаграмма, линейный график, кривая распределения — примеры диаграмм, используемых в исследовательском анализе

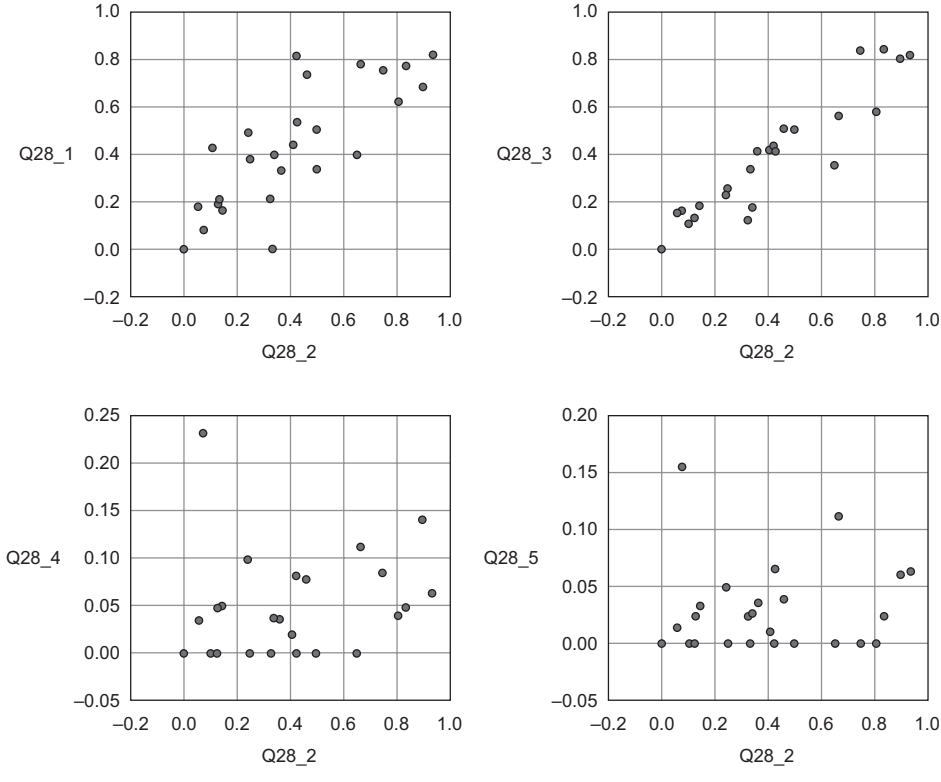


Рис. 2.15. Размещение диаграмм вблизи друг от друга помогает лучше понять структуру данных с несколькими переменными

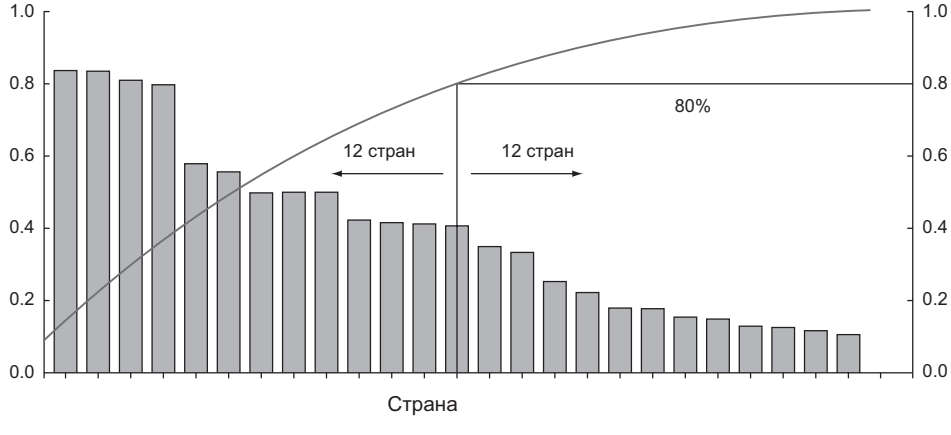


Рис. 2.16. Диаграмма Парето представляет собой комбинацию значений и кумулятивного распределения. Диаграмма наглядно показывает, что на первые 50% стран приходится чуть менее 80% общего вклада. Если бы на диаграмме была представлена покупательная способность, а вы занимались продажей дорогостоящих товаров, 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.

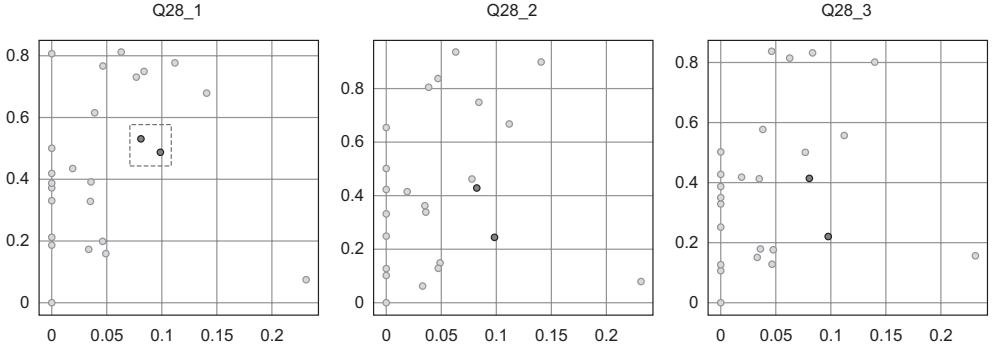


Рис. 2.17. Метод связывания и пометки данных позволяет выбрать наблюдения на одной диаграмме с выделением тех же наблюдений на другой диаграмме

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.

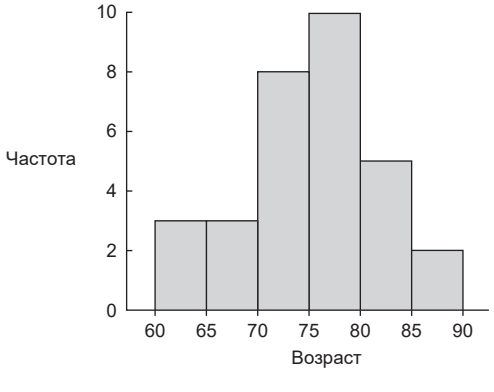


Рис. 2.18. Пример гистограммы: численность людей в возрастных группах с интервалом в 5 лет

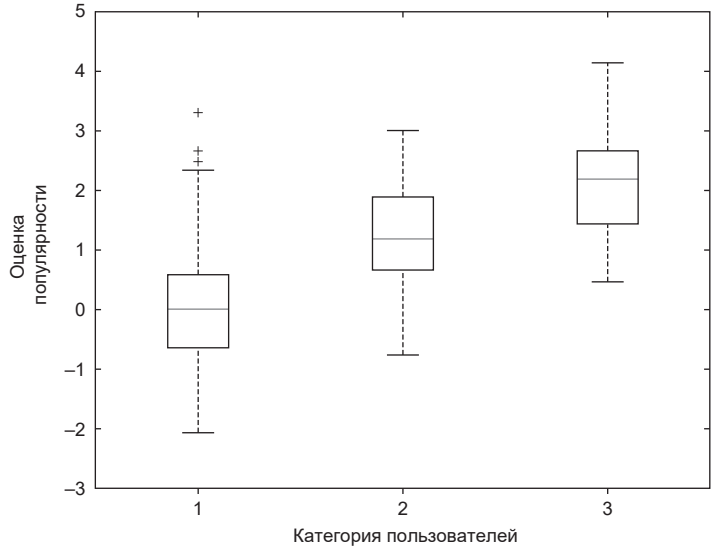


Рис. 2.19. Пример коробчатой диаграммы: у каждой категории пользователей существует распределение оценок, выставленных за определенное изображение на сайте фотографий

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.

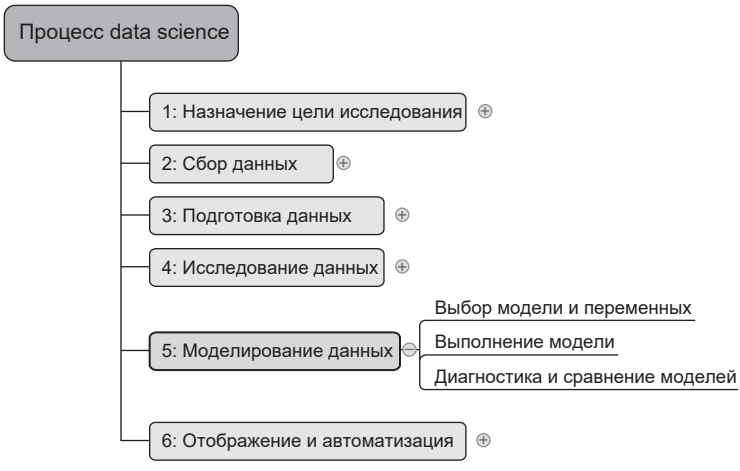


Рис. 2.20. Этап 5: Моделирование данных

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-

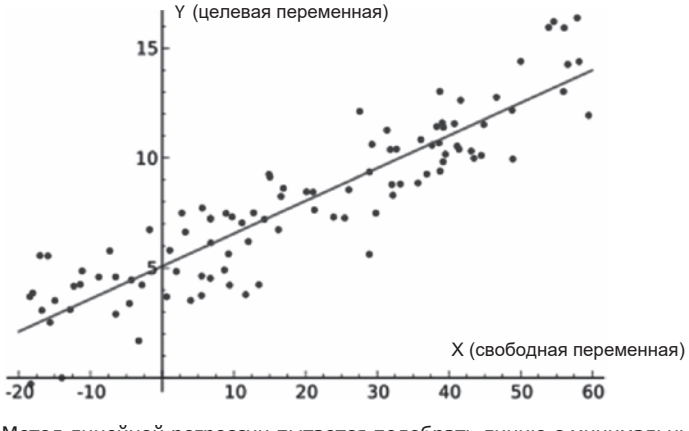


Рис. 2.21. Метод линейной регрессии пытается подобрать линию с минимальным расстоянием до каждой точки

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
import numpy as np
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()
```

Импортирование необходимых модулей Python.

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

Подбор линейной регрессии для данных.

Вывод статистики соответствия модели.

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.

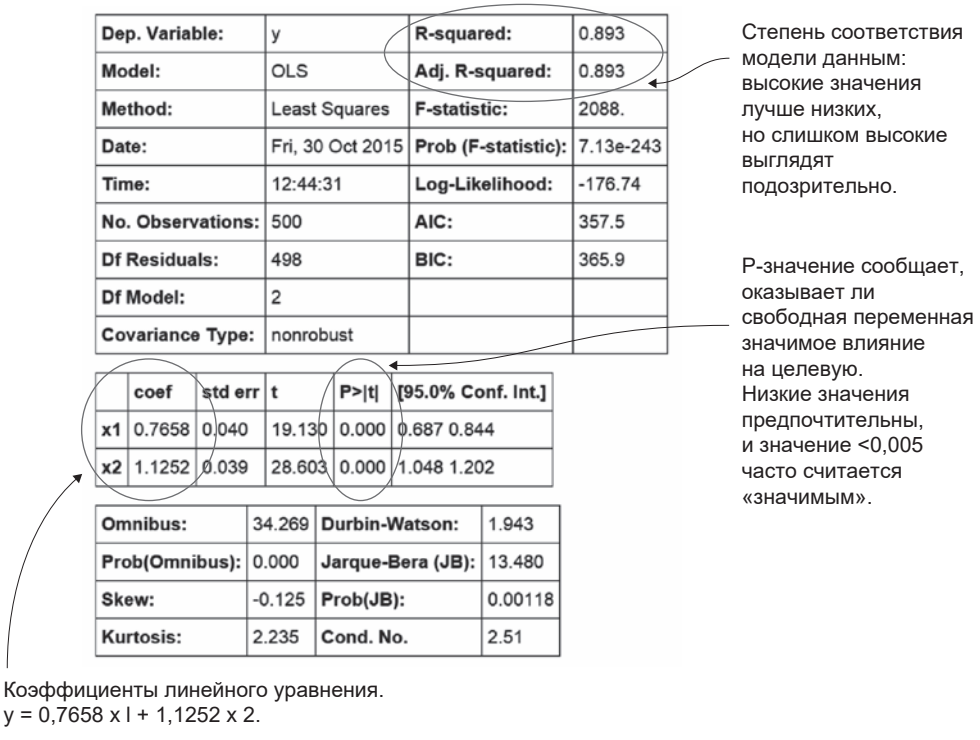


Рис. 2.22. Выходная информация модели линейной регрессии

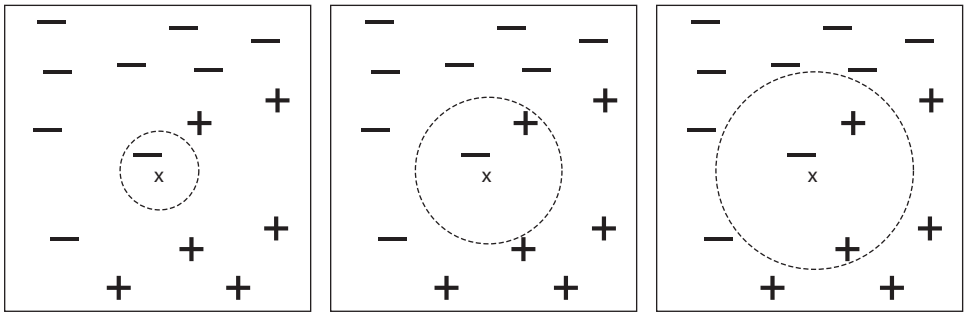
- *Коэффициенты свободных переменных* — в линейной модели эти значения интерпретируются легко. В нашем примере увеличение x_1 на 1 приводит к изменению y на 0,7658. Легко видеть, что удачный выбор свободной переменной может проложить путь к Нобелевской премии, даже если модель в целом никуда не годится. Если, например, вы определили, что некий ген является значимым фактором возникновения рака, эта информация крайне важна, даже если ген сам по себе не определяет, заболеет 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.



(a) 1-ближайший сосед (b) 2-ближайший сосед (c) 3-ближайший сосед

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
predictors = np.random.random(1000).reshape(500,2)
target = np.around(predictors.dot(np.array([0.4, 0.6])) +
         np.random.random(500))
clf = neighbors.KNeighborsClassifier(n_neighbors=10)
knn = clf.fit(predictors,target)
knn.score(predictors, target)
```

Импортирование модулей.

Создание случайных свободных данных и полуслучайных целевых данных на основании свободных.

Классификация по модели 10 ближайших соседей.

Получение метрики соответствия модели: какой процент классификации был правильным?