# Оглавление

От издательства ... 20

**Предисловие ... 22**

Спрос на квалификацию в области data science ... 23
Модульная структура ... 23
Ключевые особенности ... 24
Ответы на вопросы ... 39
Поддержка Jupyter ... 40
Приложения ... 40
Как связаться с авторами книги ... 41
Благодарности ... 41
Об авторах ... 43
О компании Deitel® & Associates, Inc. ... 44

**Приступая к работе ... 45**

Загрузка примеров кода ... 45
Структура папки examples ... 46
Установка Anaconda ... 46
Обновление Anaconda ... 47
Менеджеры пакетов ... 47
Установка программы статического анализа кода Prospector ... 48
Установка jupyter-matplotlib ... 48

Установка других пакетов ... 48
Получение учетной записи разработчика Twitter ... 49
Необходимость подключения к интернету в некоторых главах ... 49
Различия в выводе программ ... 49
Получение ответов на вопросы ... 50

**Глава 1.** Компьютеры и Python ... 51
1.1. Введение ... 52
1.2. Основы объектных технологий ... 53
1.3. Python ... 57
1.4. Библиотеки ... 59
1.4.1. Стандартная библиотека Python ... 60
1.4.2. Библиотеки data science ... 61
1.5. Первые эксперименты: использование IPython и Jupyter Notebook ... 63
1.5.1. Использование интерактивного режима IPython как калькулятора ... 63
1.5.2. Выполнение программы Python с использованием интерпретатора IPython ... 65
1.5.3. Написание и выполнение кода в Jupyter Notebook ... 68
1.6. Облачные вычисления и «интернет вещей» ... 73
1.6.1. Облачные вычисления ... 73
1.6.2. «Интернет вещей» ... 75
1.7. Насколько велики большие данные? ... 76
1.7.1. Анализ больших данных ... 83
1.7.2. Data Science и большие данные изменяют ситуацию: практические примеры ... 84
1.8. Практический пример: использование больших данных в мобильном приложении ... 87
1.9. Введение в data science: искусственный интеллект — на пересечении компьютерной теории и data science ... 89
1.10. Итоги ... 93

**Глава 2.** Введение в программирование Python ... 95
2.1. Введение ... 96
2.2. Переменные и команды присваивания ... 96
2.3. Арифметические операторы ... 97
2.4. Функция print и строки, заключенные в одинарные и двойные кавычки ... 103

2.5. Строки в тройных кавычках ... 105
2.6. Получение ввода от пользователя ... 107
2.7. Принятие решений: команда if и операторы сравнения ... 109
2.8. Объекты и динамическая типизация ... 116
2.9. Введение в data science: основные описательные статистики ... 117
2.10. Итоги ... 120

**Глава 3.** Управляющие команды ... 121
3.1. Введение ... 122
3.2. Управляющие команды ... 122
3.3. Команда if ... 123
3.4. Команды if...else и if...elif...else ... 125
3.5. Команда while ... 129
3.6. Команда for ... 130
3.6.1. Итерируемые объекты, списки и итераторы ... 131
3.6.2. Встроенная функция range ... 132
3.7. Расширенное присваивание ... 132
3.8. Повторение, управляемое последовательностью; отформатированные строки ... 133
3.9. Повторение, управляемое контрольным значением ... 135
3.10. Подробнее о встроенной функции range ... 137
3.11. Использование типа Decimal для представления денежных сумм ... 138
3.12. Команды break и continue ... 143
3.13. Логические операторы and, or или not ... 144
3.14. Введение в data science: параметры, характеризующие положение центра распределения, — математическое ожидание, медиана и мода ... 148
3.15. Итоги ... 150

**Глава 4.** Функции ... 151
4.1. Введение ... 152
4.2. Определение функций ... 152
4.3. Функции с несколькими параметрами ... 156
4.4. Генератор случайных чисел ... 158
4.5. Практический пример: игра «крэпс» ... 161
4.6. Стандартная библиотека Python ... 165
4.7. Функции модуля math ... 167

4.8. Использование автозаполнения IPython ... 168
4.9. Значения параметров по умолчанию ... 170
4.10. Ключевые аргументы ... 171
4.11. Произвольные списки аргументов ... 172
4.12. Методы: функции, принадлежащие объектам ... 173
4.13. Правила области видимости ... 174
4.14. Подробнее об импортировании ... 177
4.15. Подробнее о передаче аргументов функциям ... 179
4.16. Рекурсия ... 183
4.17. Программирование в функциональном стиле ... 187
4.18. Введение в data science: дисперсионные характеристики ... 190
4.19. Итоги ... 192

**Глава 5.** Последовательности: списки и кортежи ... 194
5.1. Введение ... 195
5.2. Списки ... 195
5.3. Кортежи ... 201
5.4. Распаковка последовательностей ... 204
5.5. Сегментация последовательностей ... 207
5.6. Команда del ... 210
5.7. Передача списков функциям ... 211
5.8. Сортировка списков ... 213
5.9. Поиск в последовательностях ... 214
5.10. Другие методы списков ... 217
5.11. Моделирование стека на базе списка ... 220
5.12. Трансформации списков ... 221
5.13. Выражения-генераторы ... 223
5.14. Фильтрация, отображение и свертка ... 224
5.15. Другие функции обработки последовательностей ... 227
5.16. Двумерные списки ... 230
5.17. Введение в data science: моделирование и статические визуализации ... 232
5.17.1. Примеры диаграмм для 600, 60 000 и 6 000 000 бросков ... 233
5.17.2. Визуализация частот и процентов ... 236
5.18. Итоги ... 244

**Глава 6.** Словари и множества ... 246
6.1. Введение ... 247
6.2. Словари ... 247
6.2.1. Создание словаря ... 248
6.2.2. Перебор по словарю ... 249
6.2.3. Основные операции со словарями ... 249
6.2.4. Методы keys и values ... 252
6.2.5. Сравнения словарей ... 254
6.2.6. Пример: словарь с оценками студентов ... 254
6.2.7. Пример: подсчет слов ... 255
6.2.8. Метод update ... 258
6.2.9. Трансформации словарей ... 258
6.3. Множества ... 259
6.3.1. Сравнение множеств ... 262
6.3.2. Математические операции с множествами ... 263
6.3.3. Операторы и методы изменяемых множеств ... 265
6.3.4. Трансформации множеств ... 267
6.4. Введение в data science: динамические визуализации ... 267
6.4.1. Как работает динамическая визуализация ... 268
6.4.2. Реализация динамической визуализации ... 271
6.5. Итоги ... 275

**Глава 7.** NumPy и программирование, ориентированное на массивы ... 277
7.1. Введение ... 278
7.2. Создание массивов на основе существующих данных ... 279
7.3. Атрибуты array ... 280
7.4. Заполнение array конкретными значениями ... 282
7.5. Создание коллекций array по диапазонам ... 283
7.6. Сравнение быстродействия списков и array ... 285
7.7. Операторы array ... 288
7.8. Вычислительные методы NumPy ... 290
7.9. Универсальные функции ... 292
7.10. Индексирование и сегментация ... 295
7.11. Представления: поверхностное копирование ... 296
7.12. Глубокое копирование ... 299

7.13. Изменение размеров и транспонирование ... 300
7.14. Введение в data science: коллекции Series и DataFrame библиотеки pandas ... 303
7.14.1. Коллекция Series ... 304
7.14.2. DataFrame ... 309
7.15. Итоги ... 319

**Глава 8.** Подробнее о строках ... 322
8.1. Введение ... 323
8.2. Форматирование строк ... 324
8.2.1. Типы представлений ... 324
8.2.2. Ширины полей и выравнивание ... 326
8.2.3. Форматирование чисел ... 327
8.2.4. Метод format ... 328
8.3. Конкатенация и повторение строк ... 329
8.4. Удаление пропусков из строк ... 330
8.5. Изменение регистра символов ... 331
8.6. Операторы сравнения для строк ... 331
8.7. Поиск подстрок ... 332
8.8. Замена подстрок ... 334
8.9. Разбиение и объединение строк ... 334
8.10. Символы и методы проверки символов ... 337
8.11. Необработанные строки ... 338
8.12. Знакомство с регулярными выражениями ... 339
8.12.1. Модуль re и функция fullmatch ... 341
8.12.2. Замена подстрок и разбиение строк ... 345
8.12.3. Другие функции поиска, обращение к совпадениям ... 346
8.13. Введение в data science: pandas, регулярные выражения и первичная обработка данных ... 350
8.14. Итоги ... 356

**Глава 9.** Файлы и исключения ... 358
9.1. Введение ... 359
9.2. Файлы ... 360
9.3. Обработка текстовых файлов ... 361
9.3.1. Запись в текстовый файл: команда with ... 361
9.3.2. Чтение данных из текстового файла ... 363

9.4. Обновление текстовых файлов ... 364
9.5. Сериализация в формат JSON ... 367
9.6. Вопросы безопасности: сериализация и десериализация pickle ... 370
9.7. Дополнительные замечания по поводу файлов ... 371
9.8. Обработка исключений ... 372
9.8.1. Деление на нуль и недействительный ввод ... 372
9.8.2. Команды try ... 373
9.8.3. Перехват нескольких исключений в одной секции except ... 377
9.8.4. Какие исключения выдают функция или метод? ... 377
9.8.5. Какой код должен размещаться в наборе try? ... 377
9.9. Секция finally ... 378
9.10. Явная выдача исключений ... 380
9.11. Раскрутка стека и трассировка (дополнение) ... 381
9.12. Введение в data science: работа с CSV-файлами ... 384
9.12.1. Модуль csv стандартной библиотеки Python ... 384
9.12.2. Чтение CSV-файлов в коллекции DataFrame библиотеки pandas ... 387
9.12.3. Чтение набора данных катастрофы «Титаника» ... 389
9.12.4. Простой анализ данных на примере набора данных катастрофы «Титаника» ... 391
9.12.5. Гистограмма возраста пассажиров ... 392
9.13. Итоги ... 393

**Глава 10.** Объектно-ориентированное программирование ... 395
10.1. Введение ... 396
10.2. Класс Account ... 399
10.2.1. Класс Account в действии ... 399
10.2.2. Определение класса Account ... 401
10.2.3. Композиция: ссылка на объекты как компоненты классов ... 404
10.3. Управление доступом к атрибутам ... 404
10.4. Использование свойств для доступа к данным ... 405
10.4.1. Класс Time в действии ... 405
10.4.2. Определение класса Time ... 408
10.4.3. Замечания по проектированию определения класса Time ... 412
10.5. Моделирование «приватных» атрибутов ... 414

10.6. Практический пример: моделирование тасования и сдачи карт ..... 416
10.6.1. Классы Card и DeckOfCards в действии ..... 416
10.6.2. Класс Card — знакомство с атрибутами класса ..... 418
10.6.3. Класс DeckOfCards ..... 421
10.6.4. Вывод изображений карт средствами Matplotlib ..... 423
10.7. Наследование: базовые классы и подклассы ..... 426
10.8. Построение иерархии наследования. Концепция полиморфизма ..... 429
10.8.1. Базовый класс CommissionEmployee ..... 430
10.8.2. Подкласс SalariedCommissionEmployee ..... 433
10.8.3. Полиморфная обработка CommissionEmployee и SalariedCommissionEmployee ..... 438
10.8.4. Объектно-базированное и объектно-ориентированное программирование ..... 439
10.9. Утиная типизация и полиморфизм ..... 439
10.10. Перегрузка операторов ..... 441
10.10.1. Класс Complex в действии ..... 443
10.10.2. Определение класса Complex ..... 444
10.11. Иерархия классов исключений и пользовательские исключения ..... 446
10.12. Именованные кортежи ..... 448
10.13. Краткое введение в новые классы данных Python 3.7 ..... 449
10.13.1. Создание класса данных Card ..... 450
10.13.2. Использование класса данных Card ..... 454
10.13.3. Преимущества классов данных перед именованными кортежами ..... 456
10.13.4. Преимущества класса данных перед традиционными классами ..... 456
10.14. Модульное тестирование с doc-строками и doctest ..... 457
10.15. Пространства имен и области видимости ..... 462
10.16. Введение в data science: временные ряды и простая линейная регрессия ..... 466
10.17. Итоги ..... 477

**Глава 11.** Обработка естественного языка (NLP) ..... 479
11.1. Введение ..... 480
11.2. TextBlob ..... 481
11.2.1. Создание TextBlob ..... 484

11.2.2. Разбиение текста на предложения и слова ... 484
11.2.3. Пометка частей речи ... 485
11.2.4. Извлечение именных конструкций ... 486
11.2.5. Анализ эмоциональной окраски с использованием анализатора TextBlob по умолчанию ... 487
11.2.6. Анализ эмоциональной окраски с использованием NaiveBayesAnalyzer ... 489
11.2.7. Распознавание языка и перевод ... 490
11.2.8. Формообразование: образование единственного и множественного числа ... 492
11.2.9. Проверка орфографии и исправление ошибок ... 493
11.2.10. Нормализация: выделение основы и лемматизация ... 494
11.2.11. Частоты слов ... 495
11.2.12. Получение определений, синонимов и антонимов из WordNet ... 496
11.2.13. Удаление игнорируемых слов ... 498
11.2.14. n-граммы ... 500
11.3. Визуализация частот вхождения слов с использованием гистограмм и словарных облаков ... 501
11.3.1. Визуализация частот вхождения слов средствами Pandas ... 501
11.3.2. Визуализация частот слов в словарных облаках ... 505
11.4. Оценка удобочитаемости с использованием Textatistic ... 508
11.5. Распознавание именованных сущностей с использованием spaCy ... 511
11.6. Выявление сходства средствами spaCy ... 513
11.7. Другие библиотеки и инструменты NLP ... 514
11.8. Машинное обучение и NLP-приложения с глубоким обучением ... 515
11.9. Наборы данных естественных языков ... 516
11.10. Итоги ... 517

**Глава 12.** Глубокий анализ данных Twitter ... 519
12.1. Введение ... 520
12.2. Обзор Twitter APIs ... 522
12.3. Создание учетной записи Twitter ... 524
12.4. Получение регистрационных данных Twitter — создание приложения ... 525
12.5. Какую информацию содержит объект Tweet? ... 527
12.6. Tweepy ... 532
12.7. Аутентификация Twitter с использованием Tweepy ... 533

12.8. Получение информации об учетной записи Twitter ... 535
12.9. Введение в курсоры Tweepy: получение подписчиков и друзей учетной записи ... 537
12.9.1. Определение подписчиков учетной записи ... 538
12.9.2. Определение друзей учетной записи ... 540
12.9.3. Получение недавних твитов пользователя ... 541
12.10. Поиск недавних твитов ... 542
12.11. Выявление тенденций: Twitter Trends API ... 545
12.11.1. Места с актуальными темами ... 546
12.11.2. Получение списка актуальных тем ... 547
12.11.3. Создание словарного облака по актуальным темам ... 549
12.12. Очистка / предварительная обработка твитов для анализа ... 550
12.13. Twitter Streaming API ... 553
12.13.1. Создание подкласса StreamListener ... 553
12.13.2. Запуск обработки потока ... 557
12.14. Анализ эмоциональной окраски твитов ... 559
12.15. Геокодирование и вывод информации на карте ... 564
12.15.1. Получение твитов и нанесение их на карту ... 566
12.15.2. Вспомогательные функции tweetutilities.py ... 571
12.15.3. Класс LocationListener ... 573
12.16. Способы хранения твитов ... 574
12.17. Twitter и временные ряды ... 575
12.18. Итоги ... 575

**Глава 13.** IBM Watson и когнитивные вычисления ... 577
13.1. Введение: IBM Watson и когнитивные вычисления ... 578
13.2. Учетная запись IBM Cloud и консоль Cloud ... 580
13.3. Сервисы Watson ... 581
13.4. Другие сервисы и инструменты ... 586
13.5. Watson Developer Cloud Python SDK ... 588
13.6. Практический пример: приложение-переводчик ... 589
13.6.1. Перед запуском приложения ... 590
13.6.2. Пробный запуск приложения ... 592
13.6.3. Сценарий SimpleLanguageTranslator.py ... 594
13.7. Ресурсы Watson ... 607
13.8. Итоги ... 610

**Глава 14.** Машинное обучение: классификация, регрессия и кластеризация ... 611
14.1. Введение в машинное обучение ... 612
14.1.1. Scikit-learn ... 613
14.1.2. Типы машинного обучения ... 615
14.1.3. Наборы данных, включенные в поставку scikit-learn ... 618
14.1.4. Последовательность действий в типичном исследовании data science ... 619
14.2. Практический пример: классификация методом k ближайших соседей и набор данных Digits, часть 1 ... 620
14.2.1. Алгоритм k ближайших соседей ... 622
14.2.2. Загрузка набора данных ... 623
14.2.3. Визуализация данных ... 627
14.2.4. Разбиение данных для обучения и тестирования ... 629
14.2.5. Создание модели ... 631
14.2.6. Обучение модели ... 631
14.2.7. Прогнозирование классов для рукописных цифр ... 632
14.3. Практический пример: классификация методом k ближайших соседей и набор данных Digits, часть 2 ... 634
14.3.1. Метрики точности модели ... 634
14.3.2. K-проходная перекрестная проверка ... 639
14.3.3. Выполнение нескольких моделей для поиска наилучшей ... 641
14.3.4. Настройка гиперпараметров ... 643
14.4. Практический пример: временные ряды и простая линейная регрессия ... 644
14.5. Практический пример: множественная линейная регрессия с набором данных California Housing ... 651
14.5.1. Загрузка набора данных ... 651
14.5.2. Исследование данных средствами Pandas ... 654
14.5.3. Визуализация признаков ... 656
14.5.4. Разбиение данных для обучения и тестирования ... 661
14.5.5. Обучение модели ... 661
14.5.6. Тестирование модели ... 663
14.5.7. Визуализация ожидаемых и прогнозируемых цен ... 664
14.5.8. Метрики регрессионной модели ... 665
14.5.9. Выбор лучшей модели ... 666

14.6. Практический пример: машинное обучение без учителя, часть 1 — понижение размерности ... 667
14.7. Практический пример: машинное обучение без учителя, часть 2 — кластеризация методом k средних ... 672
14.7.1. Загрузка набора данных Iris ... 674
14.7.2. Исследование набора данных Iris: описательная статистика в Pandas ... 676
14.7.3. Визуализация набора данных функцией pairplot ... 678
14.7.4. Использование оценщика KMeans ... 682
14.7.5. Понижение размерности методом анализа главных компонент ... 684
14.7.6. Выбор оптимального оценщика для кластеризации ... 687
14.8. Итоги ... 690

**Глава 15.** Глубокое обучение ... 692
15.1. Введение ... 693
15.1.1. Практическое применение глубокого обучения ... 696
15.1.2. Демонстрационные приложения глубокого обучения ... 696
15.1.3. Ресурсы Keras ... 697
15.2. Встроенные наборы данных Keras ... 697
15.3. Нестандартные среды Anaconda ... 699
15.4. Нейронные сети ... 701
15.5. Тензоры ... 704
15.6. Сверточные нейронные сети для распознавания образов; множественная классификация с набором данных MNIST ... 706
15.6.1. Загрузка набора данных MNIST ... 709
15.6.2. Исследование данных ... 709
15.6.3. Подготовка данных ... 712
15.6.4. Создание нейронной сети ... 715
15.6.5. Обучение и оценка модели ... 726
15.6.6. Сохранение и загрузка модели ... 733
15.7. Визуализация процесса обучения нейронной сети в TensorBoard ... 734
15.8. ConvnetJS: глубокое обучение и визуализация в браузере ... 738
15.9. Рекуррентные нейронные сети для последовательностей; анализ эмоциональной окраски с набором данных IMDb ... 740
15.9.1. Загрузка набора данных IMDb ... 741
15.9.2. Исследование данных ... 742

15.9.3. Подготовка данных ... 746
15.9.4. Создание нейронной сети ... 747
15.9.5. Обучение и оценка модели ... 750
15.10. Настройка моделей глубокого обучения ... 752
15.11. Модели сверточных нейронных сетей с предварительным обучением на ImageNet ... 753
15.12. Итоги ... 755

**Глава 16.** Большие данные: Hadoop, Spark, NoSQL и IoT ... 758
16.1. Введение ... 759
16.2. Реляционные базы данных и язык структурированных запросов (SQL) ... 765
16.2.1. База данных books ... 767
16.2.2. Запросы SELECT ... 773
16.2.3. Секция WHERE ... 773
16.2.4. Условие ORDER BY ... 774
16.2.5. Слияние данных из нескольких таблиц: INNER JOIN ... 776
16.2.6. Команда INSERT INTO ... 777
16.2.7. Команда UPDATE ... 778
16.2.8. Команда DELETE FROM ... 780
16.3. Базы данных NoSQL и NewSQL: краткое введение ... 781
16.3.1. Базы данных NoSQL «ключ-значение» ... 782
16.3.2. Документные базы данных NoSQL ... 782
16.3.3. Столбцовые базы данных NoSQL ... 783
16.3.4. Графовые базы данных NoSQL ... 784
16.3.5. Базы данных NewSQL ... 785
16.4. Практический пример: документная база данных MongoDB ... 786
16.4.1. Создание кластера MongoDB Atlas ... 787
16.4.2. Потоковая передача твитов в MongoDB ... 789
16.5. Hadoop ... 801
16.5.1. Обзор Hadoop ... 801
16.5.2. Получение статистики по длине слов в «Ромео и Джульетте» с использованием MapReduce ... 805
16.5.3. Создание кластера Apache Hadoop в Microsoft Azure HDInsight ... 805
16.5.4. Hadoop Streaming ... 808
16.5.5. Реализация сценария отображения ... 809
16.5.6. Реализация сценария свертки ... 810

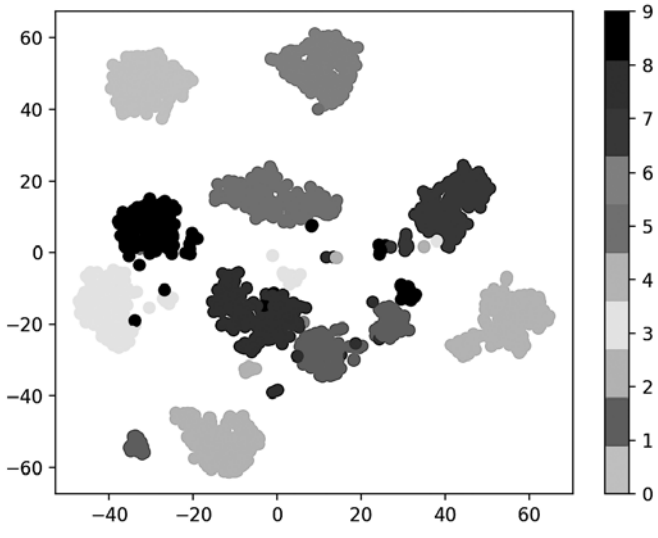
16.5.7. Подготовка к запуску примера MapReduce ... 811
16.5.8. Выполнение задания MapReduce ... 812
16.6. Spark ... 816
16.6.1. Краткий обзор Spark ... 816
16.6.2. Docker и стеки Jupyter Docker ... 818
16.6.3. Подсчет слов с использованием Spark ... 823
16.6.4. Подсчет слов средствами Spark в Microsoft Azure ... 827
16.7. Spark Streaming: подсчет хештегов Twitter с использованием стека Docker pyspark-notebook ... 831
16.7.1. Потоковая передача твитов в сокет ... 832
16.7.2. Получение сводки хештегов и Spark SQL ... 836
16.8. «Интернет вещей» ... 844
16.8.1. Публикация и подписка ... 846
16.8.2. Визуализация живого потока PubNub средствами Freeboard ... 846
16.8.3. Моделирование термостата, подключенного к интернету, в коде Python ... 849
16.8.4. Создание информационной панели с Freeboard.io ... 853
16.8.5. Создание подписчика PubNub в коде Python ... 854
16.9. Итоги ... 860

# От издательства

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 (издательство «Питер», компьютерная редакция). Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.




## 14.7. Практический пример: машинное обучение без учителя, часть 2 — кластеризация методом k средних

В этом разделе будет представлен, пожалуй, самый простой из алгоритмов машинного обучения без учителя — *кластеризация методом k средних*. Алгоритм анализирует *непомеченные* образцы и пытается объединить их в кластеры. Поясним, что *k* в «методе *k* средних» представляет количество кластеров, на которые предполагается разбить данные.

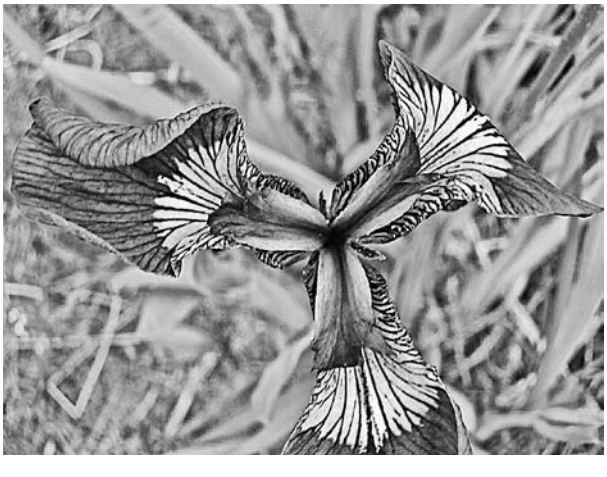
Алгоритм распределяет образцы на заранее заданное количество кластеров, используя метрики расстояния, сходные с метриками алгоритма кластеризации *k* ближайших соседей. Каждый кластер группируется вокруг *центроида* — центральной точки кластера. Изначально алгоритм выбирает *k* случайных центроидов среди образцов набора данных, после чего остальные образцы распределяются по кластерам с ближайшим центроидом. Далее выполняется итеративный пересчет центроидов, а образцы перераспределяются по кластерам, пока для всех кластеров расстояние от заданного центроида до образцов, входящих в его кластер, не будет минимизировано. В результате выполнения алгоритма формируется одномерный массив меток, обозначающих кластер,

к которому относится каждый образец, а также двумерный массив центроидов, представляющих центр каждого кластера.

## Набор данных Iris

Поработаем с популярным *набором данных Iris*[1], входящим в поставку scikit-learn. Этот набор часто анализируется при классификации и кластеризации. И хотя набор данных помечен, мы не будем использовать эти метки, чтобы продемонстрировать кластеризацию. Затем метки будут использованы для определения того, насколько хорошо алгоритм *k* средних выполняет кластеризацию образцов.

Набор данных Iris относится к «игрушечным» наборам данных, поскольку состоит только из 150 образцов и четырех признаков. Набор данных описывает 50 образцов трех видов цветов ириса — *Iris setosa*, *Iris versicolor* и *Iris virginica* (см. фотографии ниже). Признаки образцов: длина наружной доли околоцветника (sepal length), ширина наружной доли околоцветника (sepal width), длина внутренней доли околоцветника (petal length) и ширина внутренней доли околоцветника (petal width), измеряемые в сантиметрах.



*Iris setosa. Credit: Courtesy of Nation Park services*

[1] Fisher, R.A., «The use of multiple measurements in taxonomic problems», Annual Eugenics, 7, Part II, 179–188 (1936); также «Contributions to Mathematical Statistics» (John Wiley, NY, 1950).

*Iris versicolor. Credit: Courtesy of Jefficus*

*Iris virginica. Credit: Christer T Johansson*

## 14.7.1. Загрузка набора данных Iris

Запустите IPython командой `ipython --matplotlib`, после чего воспользуйтесь *функцией* `load_iris` модуля `sklearn.datasets` для получения объекта `Bunch` с набором данных:

```
In [1]: from sklearn.datasets import load_iris

In [2]: iris = load_iris()
```

Атрибут `DESCR` объекта `Bunch` показывает, что набор данных состоит из 150 образцов (`Number of Instances`), каждый из которых обладает четырьмя признаками (`Number of Attributes`). В наборе данных нет отсутствующих значений. Образцы классифицируются целыми числами 0, 1 и 2, представляющими *Iris setosa*, *Iris versicolor* и *Iris virginica* соответственно. *Проигнорируем* метки и поручим определение классов образцов алгоритму кластеризации методом *k* средних. Ключевая информация `DESCR` выделена жирным шрифтом:

```
In [3]: print(iris.DESCR)
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
    ============== ==== ==== ======= ===== ====================

    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988
...
```

### Проверка количества образцов, признаков и целевых значений

Количество образцов и признаков можно узнать из атрибута `shape` массива `data`, а количество целевых значений — из атрибута `shape` массива `target`:

```
In [4]: iris.data.shape
Out[4]: (150, 4)

In [5]: iris.target.shape
Out[5]: (150,)
```

Массив `target_names` содержит имена числовых меток массива. Выражение `target — dtype='<U10'` означает, что его элементами являются строки длиной не более 10 символов:

```
In [6]: iris.target_names
Out[6]: array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

Массив `feature_names` содержит список строковых имен для каждого столбца в массиве `data`:

```
In [7]: iris.feature_names
Out[7]:
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

## 14.7.2. Исследование набора данных Iris: описательная статистика в Pandas

Используем коллекцию `DataFrame` для исследования набора данных Iris. Как и в случае с набором данных California Housing, зададим параметры pandas для форматирования столбцового вывода:

```
In [8]: import pandas as pd

In [9]: pd.set_option('max_columns', 5)

In [10]: pd.set_option('display.width', None)
```

Создадим коллекцию `DataFrame` с содержимым массива `data`, используя содержимое массива `feature_names` как имена столбцов:

```
In [11]: iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
```

Затем добавим столбец с названием вида для каждого из образцов. Трансформация списка в следующем фрагменте использует каждое значение в массиве `target` для поиска соответствующего названия в массиве `target_names`:

```
In [12]: iris_df['species'] = [iris.target_names[i] for i in iris.target]
```

Воспользуемся pandas для идентификации нескольких образцов. Как и прежде, если pandas выводит \ справа от имени столбца, это означает, что в выводе остаются столбцы, которые будут выведены ниже:

```
In [13]: iris_df.head()
Out[13]:
   sepal length (cm)  sepal width (cm)  petal length (cm)  \
0                5.1               3.5                1.4
1                4.9               3.0                1.4
2                4.7               3.2                1.3
3                4.6               3.1                1.5
4                5.0               3.6                1.4
   petal width (cm) species
0               0.2  setosa
1               0.2  setosa
2               0.2  setosa
3               0.2  setosa
4               0.2  setosa
```

Вычислим некоторые показатели описательной статистики для числовых столбцов:

```
In [14]: pd.set_option('precision', 2)

In [15]: iris_df.describe()
Out[15]:
       sepal length (cm)  sepal width (cm)  petal length (cm)  \
count             150.00            150.00             150.00
mean                5.84              3.06               3.76
std                 0.83              0.44               1.77
min                 4.30              2.00               1.00
25%                 5.10              2.80               1.60
50%                 5.80              3.00               4.35
75%                 6.40              3.30               5.10
max                 7.90              4.40               6.90
```

```
       petal width (cm)
count            150.00
mean               1.20
std                0.76
min                0.10
25%                0.30
50%                1.30
75%                1.80
max                2.50
```

Вызов метода `describe` для столбца `'species'` подтверждает, что он содержит три уникальных значения. Нам заранее известно, что данные состоят из трех классов, к которым относятся образцы, хотя в машинном обучении без учителя это и не всегда так.

```
In [16]: iris_df['species'].describe()
Out[16]:
count        150
unique         3
top       setosa
freq          50
Name: species, dtype: object
```

## 14.7.3. Визуализация набора данных функцией pairplot

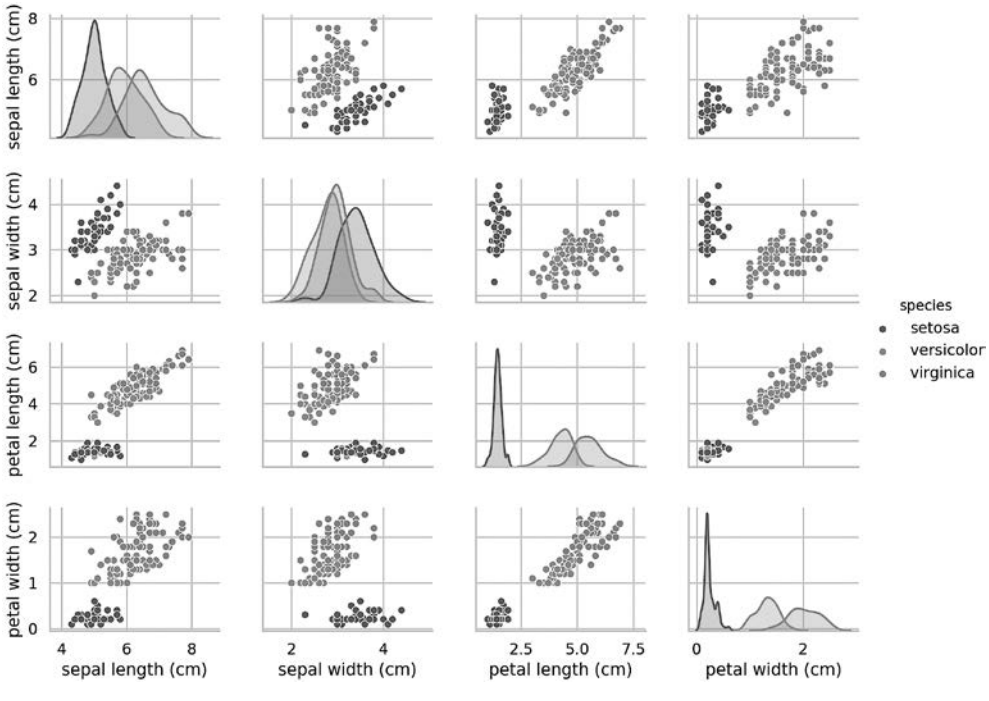
Проведем визуализацию признаков в этом наборе данных. Один из способов извлечь информацию о ваших данных — посмотреть, как признаки связаны друг с другом. Набор данных имеет четыре признака. Мы не сможем построить диаграмму соответствия одного признака с тремя другими на одной диаграмме. Тем не менее можно построить диаграмму, на которой будет представлено соответствие между двумя признаками. Фрагмент [20] использует функцию `pairplot` библиотеки Seaborn для создания таблицы диаграмм, на которых каждый признак сопоставляется с одним из других признаков:

```
In [17]: import seaborn as sns

In [18]: sns.set(font_scale=1.1)

In [19]: sns.set_style('whitegrid')

In [20]: grid = sns.pairplot(data=iris_df, vars=iris_df.columns[0:4],
    ...:     hue='species')
    ...:
```

Ключевые аргументы:

- `data` — коллекция `DataFrame`[1] с набором данных, наносимым на диаграмму;
- `vars` — последовательность с именами переменных, наносимых на диаграмму. Для коллекции `DataFrame` она содержит имена столбцов. В данном случае используются первые четыре столбца `DataFrame`, представляющие длину (ширину) наружной доли околоцветника и длину (ширину) внутренней доли околоцветника соответственно;
- `hue` — столбец коллекции `DataFrame`, используемый для определения цветов данных, наносимых на диаграмму. В данном случае данные окрашиваются в зависимости от вида ирисов.

Предыдущий вызов `pairplot` строит следующую таблицу диаграмм 4 × 4:




[1] Также может использоваться двумерный массив или список.