

Предисловие	11
Об авторе.....	12
О рецензентах.....	13
Введение	15
 Глава 1. Наделение компьютеров способностью обучаться на данных	 25
Построение интеллектуальных машин для преобразования данных в знания	25
Три типа машинного обучения.....	26
Выполнение прогнозов о будущем на основе обучения с учителем.....	26
Задача классификации – распознавание меток классов	27
Задача регрессии – предсказание непрерывных результатов.....	28
Решение интерактивных задач на основе обучения с подкреплением	29
Обнаружение скрытых структур при помощи обучения без учителя	30
Выявление подгрупп при помощи кластеризации	30
Снижение размерности для сжатия данных	31
Введение в основополагающую терминологию и систему обозначений	32
Дорожная карта для построения систем машинного обучения.....	33
Предобработка – приведение данных в приемлемый вид.....	34
Тренировка и отбор прогнозной модели.....	35
Оценка моделей и прогнозирование на ранее не встречавшихся экземплярах данных.....	36
Использование Python для машинного обучения.....	36
Установка библиотек Python	37
Записные книжки Jupyter/IPython.....	38
Резюме	40
 Глава 2. Тренировка алгоритмов машинного обучения для задачи классификации.....	 42
Искусственные нейроны – краткий обзор ранней истории машинного обучения.....	42
Реализация алгоритма обучения персептрона на Python.....	48
Тренировка персептронной модели на наборе данных цветков ириса	50
Адаптивные линейные нейроны и сходимость обучения.....	54
Минимизация функций стоимости методом градиентного спуска	55
Реализация адаптивного линейного нейрона на Python	57
Крупномасштабное машинное обучение и стохастический градиентный спуск	62
Резюме	67
 Глава 3. Обзор классификаторов с использованием библиотеки scikit-learn.....	 68
Выбор алгоритма классификации.....	68

Первые шаги в работе с scikit-learn.....	69
Тренировка персептрона в scikit-learn.....	69
Моделирование вероятностей классов логистической регрессии.....	73
Интуитивное понимание логистической регрессии и условные вероятности.....	74
Извлечение весов логистической функции стоимости.....	77
Тренировка логистической регрессионной модели в scikit-learn	79
Решение проблемы переподргонки при помощи регуляризации.....	81
Классификация с максимальной маржой на основе машин опорных векторов.....	84
Интуитивное понимание максимальной маржи.....	85
Обработка нелинейно разделимого случая при помощи ослабленных переменных	86
Альтернативные реализации в scikit-learn.....	88
Решение нелинейных задач ядерным методом SVM	88
Использование ядерного трюка для нахождения разделяющих гиперплоскостей в пространстве более высокой размерности	90
Обучение на основе деревьев решений	93
Максимизация прироста информации – получение наибольшей отдачи	94
Построение дерева решений	98
Объединение слабых учеников для создания сильного при помощи случайных лесов.....	100
k ближайших соседей – алгоритм ленивого обучения.....	103
Резюме	106

Глава 4. Создание хороших тренировочных наборов – предобработка данных

Решение проблемы пропущенных данных	107
Устранение образцов либо признаков с пропущенными значениями	109
Импутация пропущенных значений.....	110
Концепция взаимодействия с эстиматорами в библиотеке scikit-learn	110
Обработка категориальных данных	112
Преобразование порядковых признаков.....	112
Кодирование меток классов.....	113
Прямое кодирование на номинальных признаках.....	114
Разбивка набора данных на тренировочное и тестовое подмножества	116
Приведение признаков к одинаковой шкале.....	117
Отбор содержательных признаков	119
Разреженные решения при помощи L1-регуляризации	119
Алгоритмы последовательного отбора признаков.....	125
Определение важности признаков при помощи случайных лесов	130
Резюме	132

Глава 5. Сжатие данных путем снижения размерности.....

Снижение размерности без учителя на основе анализа главных компонент	133
Общая и объясненная дисперсия.....	135
Преобразование признаков	138
Анализ главных компонент в scikit-learn	140
Сжатие данных с учителем путем линейного дискриминантного анализа.....	143

Вычисление матриц разброса.....	145
Отбор линейных дискриминантов для нового подпространства признаков.....	147
Проецирование образцов на новое пространство признаков.....	149
Метод LDA в scikit-learn	150
Использование ядерного метода анализа главных компонент для нелинейных отображений.....	151
Ядерные функции и ядерный трюк.....	152
Реализация ядерного метода анализа главных компонент на Python	156
Пример 1. Разделение фигур в форме полумесяца.....	157
Пример 2. Разделение концентрических кругов	159
Проецирование новых точек данных.....	162
Ядерный метод анализа главных компонент в scikit-learn.....	165
Резюме	166

Глава 6. Изучение наиболее успешных методов оценки моделей и тонкой настройки гиперпараметров

Оптимизация потоков операций при помощи конвейеров.....	167
Загрузка набора данных Breast Cancer Wisconsin.....	167
Совмещение преобразователей и эстиматоров в конвейере.....	169
Использование k -блочной перекрестной проверки для оценки работоспособности модели.....	170
Метод проверки с откладыванием данных	171
k -блочная перекрестная проверка.....	172
Отладка алгоритмов при помощи кривой обучения и проверочной кривой.....	176
Диагностирование проблем со смещением и дисперсией при помощи кривых обучения.....	176
Решение проблемы перепогонки и недопогонки при помощи проверочных кривых	179
Тонкая настройка машинно-обучаемых моделей методом сеточного поиска	181
Настройка гиперпараметров методом поиска по сетке параметров	181
Отбор алгоритмов методом вложенной перекрестной проверки	183
Обзор других метрик оценки работоспособности.....	184
Прочтение матрицы несоответствий	185
Оптимизация точности и полноты классификационной модели	186
Построение графика характеристической кривой.....	188
Оценочные метрики для многоклассовой классификации	191
Резюме	192

Глава 7. Объединение моделей для методов ансамблевого обучения

Обучение при помощи ансамблей.....	193
Реализация простого классификатора с мажоритарным голосованием	197
Объединение разных алгоритмов классификации методом мажоритарного голосования.....	202
Оценка и тонкая настройка ансамблевого классификатора	205
Бэггинг – сборка ансамбля классификаторов из бутстрап-выборок.....	210
Усиление слабых учеников методом адаптивного бустинга	214
Резюме	221

Глава 8. Применение алгоритмов машинного обучения

в анализе мнений	222
Получение набора данных киноотзывов IMDb	222
Концепция модели мешка слов.....	224
Преобразование слов в векторы признаков	225
Оценка релевантности слова методом tf-idf.....	226
Очистка текстовых данных	228
Переработка документов в лексемы.....	229
Тренировка логистической регрессионной модели для задачи классификации документов	232
Работа с более крупными данными – динамические алгоритмы и обучение вне ядра.....	234
Резюме	237

Глава 9. Встраивание алгоритма машинного обучения

в веб-приложение	239
Сериализация подогнанных эстиматоров библиотеки scikit-learn	239
Настройка базы данных SQLite для хранения данных	242
Разработка веб-приложения в веб-платформе Flask.....	244
Наше первое веб-приложение Flask.....	245
Валидация и отображение формы	246
Превращение классификатора кинофильмов в веб-приложение.....	249
Развертывание веб-приложения на публичном сервере.....	256
Обновление классификатора киноотзывов.....	258
Резюме	259

Глава 10. Прогнозирование непрерывных целевых величин

на основе регрессионного анализа	260
Введение в простую линейную регрессионную модель	260
Разведочный анализ набора данных Housing	261
Визуализация важных характеристик набора данных	263
Реализация линейной регрессионной модели обычным методом наименьших квадратов	266
Решение уравнения регрессии для параметров регрессии методом градиентного спуска	267
Оценивание коэффициента регрессионной модели в scikit-learn.....	270
Подгонка стабильной регрессионной модели алгоритмом RANSAC.....	272
Оценивание работоспособности линейных регрессионных моделей	274
Применение регуляризованных методов для регрессии.....	277
Превращение линейной регрессионной модели в криволинейную – полиномиальная регрессия	278
Моделирование нелинейных связей в наборе данных Housing.....	280
Обработка нелинейных связей при помощи случайных лесов	283
Регрессия на основе дерева решений.....	283
Регрессия на основе случайного леса.....	285
Резюме	287

Глава 11. Работа с немаркированными данными –

кластерный анализ	289
Группирование объектов по подобию методом k средних	289
Алгоритм k -средних++	292
Жесткая кластеризация в сопоставлении с мягкой	294
Использование метода локтя для нахождения оптимального числа кластеров	296
Количественная оценка качества кластеризации методом силуэтных графиков	298
Организация кластеров в виде иерархического дерева	302
Выполнение иерархической кластеризации на матрице расстояний	303
Прикрепление дендограмм к теплокарте	307
Применение агломеративной кластеризации в scikit-learn	308
Локализация областей высокой плотности алгоритмом DBSCAN	309
Резюме	313

Глава 12. Тренировка искусственных нейронных сетей

для распознавания изображений	315
Моделирование сложных функций искусственными нейронными сетями	315
Краткое резюме однослойных нейронных сетей	317
Введение в многослойную нейросетевую архитектуру	318
Активация нейронной сети методом прямого распространения сигналов	320
Классификация рукописных цифр	322
Получение набора данных MNIST	323
Реализация многослойного персептрона	328
Тренировка искусственной нейронной сети	339
Вычисление логистической функции стоимости	339
Тренировка нейронных сетей методом обратного распространения ошибки	341
Развитие интуитивного понимания алгоритма обратного распространения ошибки	344
Отладка нейронных сетей процедурой проверки градиента	345
Сходимость в нейронных сетях	350
Другие нейросетевые архитектуры	351
Сверточные нейронные сети	352
Рекуррентные нейронные сети	354
Несколько последних замечаний по реализации нейронной сети	355
Резюме	355

Глава 13. Распараллеливание тренировки нейронных сетей

при помощи Theano	356
Сборка, компиляция и выполнение выражений в Theano	356
Что такое Theano?	358
Первые шаги с библиотекой Theano	359
Конфигурирование библиотеки Theano	360
Работа с матричными структурами	362
Завершающий пример – линейная регрессия	364

Выбор функций активации для нейронных сетей с прямым распространением сигналов.....	367
Краткое резюме логистической функции	368
Оценивание вероятностей в многоклассовой классификации функцией softmax	370
Расширение выходного спектра при помощи гиперболического тангенса.....	371
Эффективная тренировка нейронных сетей при помощи библиотеки Keras	373
Резюме	378
Приложение А.....	380
Оценка моделей.....	380
Что такое перепогонка?.....	380
Как оценивать модель?	381
Сценарий 1. Элементарно обучить простую модель.....	381
Сценарий 2. Натренировать модель и выполнить тонкую настройку (оптимизировать гиперпараметры)	382
Сценарий 3. Построить разные модели и сравнить разные алгоритмы (например, SVM против логистической регрессии против случайных лесов и т. д.)	383
Перекрестная проверка. Оценка работоспособности эстиматора.....	384
Перекрестная проверка с исключением по одному	386
Пример стратифицированной k -блочной перекрестной проверки.....	387
Расширенный пример вложенной перекрестной проверки.....	387
А. Вложенная кросс-валидация: быстрая версия	388
Б. Вложенная кросс-валидация: ручной подход с распечаткой модельных параметров	388
В. Регулярная k -блочная кросс-валидация для оптимизации модели на полном наборе тренировочных данных	389
График проверочной (валидационной) кривой.....	389
Настройка типового конвейера и сеточного поиска.....	391
Машинное обучение	393
В чем разница между классификатором и моделью?.....	393
В чем разница между функцией стоимости и функцией потерь?	394
Обеспечение персистентности моделей scikit-learn на основе JSON.....	395
Глоссарий основных терминов и сокращений.....	400
Термины	400
Сокращения	406
Предметный указатель	408

Мы живем в потоке данных. Согласно недавним оценкам, ежедневно генерируется 2,5 квинтиллиона (10¹⁸) байт данных. Это такой огромный объем данных, что более 90% информации, которую мы храним в наши дни, было сгенерировано за все прошедшее десятилетие. К сожалению, люди не способны воспользоваться подавляющей частью этой информации. Данные либо лежат за пределами возможностей стандартных аналитических методов, либо они просто слишком обширны, чтобы наши ограниченные умы смогли их понять.

Благодаря методам машинного обучения мы наделяем компьютеры способностями обрабатывать большие объемы данных, которые в противном случае стояли бы непроницаемой стеной, даем им возможность обучаться на этих данных и извлекать из них практические выводы. От массивных суперкомпьютеров, которые обеспечивают работу поисковых движков компании Google, до смартфонов, которые мы носим в наших карманах, – везде мы опираемся на машинное обучение, которое приводит в действие значительную часть окружающего нас мира, и нередко мы об этом даже не догадываемся.

Являясь первооткрывателями наших дней дивного нового мира больших данных, нам надлежит узнать еще больше о машинном обучении. Что же такое машинное обучение и как оно работает? Каким образом его применять, чтобы заглянуть в неизведанное, привести в действие свой бизнес либо просто узнать, что Интернет в целом думает о моем любимом фильме? Все это и даже больше будет охвачено в следующих главах, созданных моим добрым другом и коллегой Себастьяном Рашка.

Когда Себастьян не занят одомашниванием моей вспыльчивой собаки, он неустанно посвящает свое свободное время сообществу специалистов, работающих с открытым исходным кодом в области машинного обучения. В течение последних нескольких лет Себастьян разработал десятки популярных учебных руководств, в которых затрагиваются различные темы из области машинного обучения и визуализации данных на Python. Он также является автором и соавтором разработок ряда библиотек Python с открытым исходным кодом, и некоторые из них теперь являются составной частью базового потока операций по машинному обучению на Python.

В силу его обширных экспертных познаний в этой области я уверен, что глубокое понимание Себастьяном мира машинного обучения на Python будет неоценимо для пользователей всех уровней квалификации. Я искренне рекомендую эту книгу любому, кто находится в поисках более широкого и более практического понимания принципов машинного обучения.

Доктор Рандал С. Олсон,
*исследователь в области искусственного интеллекта
и машинного обучения, Университет шт. Пенсильвания, США*

Себастьян Рашка – аспирант докторантуры в Мичиганском университете, США, занимающийся разработкой новых вычислительных методов в области вычислительной биологии. Веб-сайтом Analytics Vidhya (<https://www.analyticsvidhya.com/>) сообщества увлеченных профессионалов в области науки о данных отмечен первым местом среди наиболее влиятельных аналитиков данных на GitHub. За его плечами многолетний опыт программирования на Python; он также проводит ряд семинаров по практическому применению науки о данных и машинного обучения. Регулярные выступления и публикации на тему науки о данных, машинного обучения и языка Python на деле мотивировали его написать эту книгу, с тем чтобы помочь людям разрабатывать управляемые данными решения без обязательного наличия предварительной квалификации в области машинного обучения.

Он также является активным соавтором проектов с открытым исходным кодом и автором собственных методов, которые теперь успешно применяются в конкурсах по машинному обучению, таких как Kaggle. В свое свободное время он работает над моделями для спортивного прогнозирования, и если не сидит перед компьютером, то любит проводить время, занимаясь спортом.

Хотел бы поблагодарить своих профессоров, Арун Росса и Панг-Нинг Тана, а также многих других, кто вдохновил меня и сформировал у меня огромный интерес к исследованиям в области классификации образов, машинного обучения и добычи данных.

Хотел бы воспользоваться представившейся возможностью и поблагодарить огромное сообщество пользователей и разработчиков библиотек Python с открытым исходным кодом, которые помогли мне создать совершенную среду для научных исследований и науки о данных.

Особую благодарность передаю базовым разработчикам библиотеки scikit-learn. В качестве одного из соавторов этого проекта было приятно работать вместе с замечательными людьми, которые не только превосходно осведомлены в области машинного обучения, но и являются превосходными программистами.

Наконец, хочу поблагодарить всех за проявление интереса к этой книге и искренне надеюсь, что смогу передать весь свой энтузиазм по поводу моего присоединения к огромным сообществам программистов на Python и в области машинного обучения.

Ричард Даттон начал заниматься программированием компьютера ZX Spectrum в возрасте 8 лет, и с тех пор это увлечение направляет его по противоречивому мас-сиву технологий и ролей в области промышленности и финансов.

Работал в Microsoft и управляющим в Barclays, его текущим увлечением является гибрид из Python, машинного обучения и цепочки блоков транзакций.

Если он не сидит перед компьютером, то его можно найти в спортзале либо дома с бокалом вина перед смартфоном iPhone. Он называет это равновесием.

Дэйв Джулиан – ИТ-консультант и преподаватель с 15-летним стажем. Работал техником, проектным инженером, программистом и веб-разработчиком. Его текущие проекты состоят из разработки инструмента для анализа урожайности в составе интегрированных стратегий по борьбе с сельскохозяйственными вредителями в теплицах. Его большой интерес пролегает на пересечении биологии и технологии с уверенностью, что умные машины способны помочь решить самые важные глобальные задачи.

Вахид Мирджалили получил звание доктора наук Мичиганского университета по машиностроению, где он разработал новые методы рафинирования белковых структур с использованием молекулярно-динамического имитационного моделирования. Объединив свои знания из областей статистики, добычи данных и физики, разработал мощные управляемые данными подходы, которые помогли ему и его исследовательской группе одержать победу в двух недавних мировых конкурсах по прогнозированию и рафинированию протеиновых структур, CASP, в 2012 и 2014 гг.

Работая над докторской диссертацией, решил присоединиться к факультету информатики и инженерного дела в Мичиганском университете с целью специализации в области машинного обучения. Его текущие исследовательские проекты включают разработку алгоритмов машинного обучения без учителя для добычи массивных наборов данных. Он также является страстным поклонником программирования на Python и делится своими реализациями алгоритмов кластеризации на своем личном веб-сайте <http://vahidmirjalili.com>.

Хамидреза Саттари – ИТ-профессионал, участвовавший в ряде областей, связанных с разработкой программного обеспечения, от программирования до архитектуры и управления. Владеет степенью магистра в области разработки программного обеспечения Университета Хериота-Уатта, Соединенное Королевство, и степенью бакалавра по электротехнике (электронике) Тегеранского университета Азад, Иран. В последние годы его области интереса составляли большие данные и машинное обучение. Является соавтором книги *«Веб-службы Spring 2. Книга рецептов»* (Spring Web Services 2 Cookbook); ведет свой собственный блог по адресу <http://justdeveloped-blog.blogspot.com/>.

Дмитрий Тарановский – разработчик программного обеспечения с заинтересованностью и квалификацией в Python, Linux и машинном обучении. Родом из Киева, Украина, он переехал в США в 1996 г. С раннего возраста страстно увлекался наукой и знаниями, побеждая на конкурсах по физике и математике. В 1999 г. был избран членом команды США по физике. В 2005 г. окончил Мичиганский Технологический институт со специализацией по математике. Позже работал программным

инженером над системой трансформации текста для компьютерных медицинских транскрипций (eScription). Изначально работая на Perl, он по достоинству оценил мощь и ясность Python, который позволил ему масштабировать систему до данных больших объемов. Впоследствии работал инженером программного обеспечения и аналитиком на алгоритмическую трейдинговую фирму. Он также внес значительный вклад в математические основы, в том числе создание и совершенствование расширения языка теории множеств и его связи с аксиомами множеств большой мощности, разработав понятийный аппарат конструктивной истины и создав систему порядковой индексации с реализацией на Python. Он также любит читать, быть за городом и старается сделать мир лучше.

Наверное, не стоит и говорить, что машинное обучение стало одной из самых захватывающих технологий современности. Такие крупные компании, как Google, Facebook, Apple, Amazon, IBM, и еще многие другие небезосновательно вкладывают значительный капитал в разработку методов и программных приложений в области машинного обучения. Хотя может показаться, что термин «машинное обучение» сегодня уже набил оскомину, совершенно очевидно, что весь этот ажиотаж не является результатом рекламной шумихи. Эта захватывающая область исследования открывает путь к новым возможностям и стала неотъемлемой частью нашей повседневной жизни. Разговоры с речевым ассистентом по смартфону, предоставление рекомендаций относительно подходящего продукта для клиентов, предотвращение актов мошенничества с кредитными картами, фильтрация спама из входящих сообщений электронной почты, обнаружение и диагностирование внутренних заболеваний – и этот список можно продолжать.

Если вы хотите стать практиком в области машинного обучения, более основательным решателем задач или, возможно, даже обдумываете карьеру в научно-исследовательской области, связанной с машинным обучением, то эта книга для вас! Однако новичка теоретические идеи, лежащие в основании машинного обучения, нередко могут подавлять своей сложностью. И все же многие из опубликованных в последние годы практических изданий способны помочь вам приступить к работе с машинным обучением на основе реализации мощных алгоритмов обучения. По моему мнению, использование практических примеров программного кода служит важной цели. В них идеи иллюстрируются путем приведения изученного материала непосредственно в действие. Однако помните, что огромная мощь влечет за собой большую ответственность! Идеи, лежащие в основании машинного обучения, слишком красивы и важны, чтобы их прятать в черном ящике. Поэтому моя личная миссия состоит в том, чтобы предоставить вам иную книгу: книгу, в которой обсуждаются важные подробности относительно идей и принципов машинного обучения, предлагаются интуитивные и одновременно информативные объяснения по поводу того, каким образом алгоритмы машинного обучения работают, как их использовать и, самое главное, как избежать наиболее распространенных ловушек.

Если в поисковой системе Академия Google¹ в качестве поискового запроса набрать «машинное обучение», то она вернет обескураживающе большое количество публикаций. В английском сегменте их число составляет 1 800 000 публикаций (для сравнения, в русском – 16 400). Разумеется, мы не сможем обсудить мельчайшие подробности всех основных алгоритмов и приложений, которые появились в течение предыдущих 60 лет. Однако в этой книге мы предпримем увлекательное путешествие, которое затронет все существенные темы и понятия, чтобы дать вам преимущество в этой области. Если вы считаете, что ваша потребность в знаниях из области машинного обучения не удовлетворена, то можно воспользоваться много-

¹ Академия Google (Google Scholar) (<https://scholar.google.ru/schhp?hl=ru>) – бесплатная поисковая система по полным текстам научных публикаций всех форматов и дисциплин. Проект работает с ноября 2004 г. – *Прим. перев.*

численными полезными ресурсами, чтобы отслеживать существенные прорывы в этой области.

Если вы уже подробно изучили теорию машинного обучения, то эта книга покажет вам, каким образом претворить ваши знания на практике. Если вы использовали методы машинного обучения прежде и хотите получить более глубокое понимание того, каким образом машинное обучение работает в действительности, то эта книга для вас! Не переживайте, если вы абсолютно плохо знакомы с машинным обучением; у вас гораздо больше причин испытывать предвкушение. Я обещаю вам, что машинное обучение изменит ваш способ мышления относительно задач, которые вы хотите решать, и покажу вам, как справляться с ними путем высвобождения мощи данных.

Прежде чем мы погрузимся с головой в область машинного обучения, отвечу на ваш сакраментальный вопрос – «почему, собственно, Python?» Ответ прост: он – мощный и одновременно очень доступный. Python стал самым популярным языком программирования для науки о данных, потому что он позволяет забыть об утомительных сторонах программирования и предлагает нам среду, где мы можем быстро набросать наши мысли и привести идеи непосредственно в действие.

Размышляя над тем путем, который я прошел лично, могу сказать вам совершенно искренне, что изучение методов машинного обучения сделало меня более основательным ученым, мыслителем и решателем задач. В этой книге я хочу поделиться этими знаниями с вами. Знание достигается в процессе исследования, ключом к нему является наш энтузиазм, а истинное освоение навыков может быть достигнуто только практикой. Местами продвижение может быть ухабистым, и некоторые темы могут оказаться более сложными для понимания, чем другие, но я надеюсь, что вы воспользуетесь этой возможностью и сконцентрируетесь на вознаграждении. Помните, что мы отправляемся в это путешествие вместе, и по ходу изложения мы будем пополнять ваш арсенал большим количеством мощных методов, которые помогут нам решать даже самые трудноразрешимые задачи на основе управляемого данными подхода.

О чем эта книга рассказывает

Глава 1 «Наделение компьютеров способностью обучаться на данных» знакомит с основными подобластями машинного обучения, в которых решаются самые разные практические задачи. Кроме того, в ней обсуждаются принципиальные шаги, которые необходимо предпринять, чтобы создать типичную машинно-обучаемую модель, и создается конвейер, который будет направлять нас в последующих главах.

Глава 2 «Тренировка алгоритмов машинного обучения для задачи классификации» обращается к истокам области исследований, связанной с машинным обучением, и знакомит с бинарными (двуклассовыми) классификаторами на основе персептрона и адаптивных линейных нейронов. Эта глава является осторожным введением в основополагающие принципы классификации образов, где основное внимание уделено взаимодействию машинного обучения с алгоритмами оптимизации.

Глава 3 «Обзор классификаторов с использованием библиотеки `scikit-learn`» описывает принципиальные алгоритмы машинного обучения, предназначенные для зада-

чи классификации, и предлагает практические примеры с использованием одной из самых популярных и всеобъемлющих библиотек машинного обучения с открытым исходным кодом `scikit-learn`.

Глава 4 «Создание хороших тренировочных наборов – предобработка данных» посвящена обсуждению того, как обходиться с наиболее распространенными трудностями, возникающими в работе с исходными наборами данных, такими как пропущенные данные. В ней также обсуждается ряд подходов к идентификации в наборах данных наиболее информативных признаков и будут продемонстрированы способы подготовки переменных различных типов для ввода в алгоритмы машинного обучения.

Глава 5 «Сжатие данных путем снижения размерности» посвящена описанию принципиальных методов, необходимых для сведения числа признаков в наборе данных к наборам меньшего объема при сохранении большей части их полезной и отличительной информации. В ней обсуждается стандартный подход к снижению размерности на основе главных компонент, который сравнивается с методами нелинейного преобразования с учителем.

Глава 6 «Изучение наиболее успешных методов оценки моделей и тонкой настройки гиперпараметров» посвящена обсуждению плюсов и минусов методик оценки работоспособности прогнозных моделей. Кроме того, в ней обсуждаются различные метрики, применяемые для измерения работоспособности моделей, и приемы тонкой настройки алгоритмов машинного обучения.

Глава 7 «Объединение моделей для методов ансамблевого обучения» знакомит с различными принципами эффективного объединения двух и более алгоритмов обучения. В ней будет продемонстрировано создание ансамблей экспертов с целью преодоления недостатков отдельных алгоритмов обучения, которые в результате приводят к более точным и надежным прогнозам.

Глава 8 «Применение алгоритмов машинного обучения в анализе мнений» посвящена обсуждению принципиальных шагов, необходимых для преобразования текстовых данных в содержательные представления для алгоритмов машинного обучения, с целью прогнозирования мнений людей на основе их письменной речи.

Глава 9 «Встраивание алгоритма машинного обучения в веб-приложение» продолжает прогнозную модель из предыдущей главы и проведет вас по основным шагам разработки веб-приложений со встроенными машинно-обучаемыми моделями.

Глава 10 «Прогнозирование непрерывных целевых величин на основе регрессионного анализа» посвящена обсуждению принципиальных методов, необходимых для моделирования линейных связей между целевыми переменными и переменной отклика для создания прогнозов на непрерывной шкале. После ознакомления с различными линейными моделями в ней также обсуждаются подходы на основе параболической регрессии и на основе деревьев.

Глава 11 «Работа с немаркированными данными – кластерный анализ» смещает акцент в другую подобласть машинного обучения – обучение без учителя. Мы применим алгоритмы из трех фундаментальных семейств алгоритмов кластеризации с целью нахождения групп объектов, в которых присутствует определенная степень подобия.

Глава 12 «Тренировка искусственных нейронных сетей для распознавания изображений» расширяет принцип градиентной оптимизации, который был впервые представлен в главе 2 «Тренировка алгоритмов машинного обучения для задачи

классификации» для создания мощных, многослойных нейронных сетей на основе популярного алгоритма обратного распространения ошибки.

Глава 13 «Распараллеливание тренировки нейронных сетей при помощи Theano» опирается на знания, полученные в предыдущих главах, и предоставляет практическое руководство по более эффективной тренировке нейронных сетей. В центре внимания данной главы находится библиотека Python с открытым исходным кодом Theano, которая предоставит возможность использовать ядра современных многоядерных графических процессоров.

Что требуется для этой книги

Исполнение прилагаемых к данной книге примеров программ требует установки Python версии 3.4.3 или более поздней в Mac OS X, Linux или Microsoft Windows. На протяжении всей книги мы часто будем использовать ключевые библиотеки Python для научных вычислений, в том числе SciPy, NumPy, scikit-learn, matplotlib и pandas.

В первой главе будут предложены инструкции и полезные подсказки по поводу инсталляции среды Python и указанных ключевых библиотек. Мы пополним наш репертуар дополнительными библиотеками, а инструкции по их инсталляции будут предоставлены в соответствующих главах: библиотека NLTK для обработки естественного языка (глава 8 «Применение алгоритмов машинного обучения в анализе мнений»), веб-платформа Flask (глава 9 «Встраивание алгоритма машинного обучения в веб-приложение»), библиотека seaborn для визуализации статистических данных (глава 10 «Прогнозирование непрерывных целевых величин на основе регрессионного анализа») и Theano для эффективной тренировки нейронных сетей на графических процессорах (глава 13 «Распараллеливание тренировки нейронной сети при помощи Theano»).

Для кого эта книга

Если вы хотите узнать, как использовать Python, чтобы начать отвечать на критические вопросы в отношении ваших данных, возьмите книгу «Python и машинное обучение» – и не важно, хотите вы приступить к изучению науки о данных с нуля или же намереваетесь расширить свои познания в этой области, – это принципиальный ресурс, которого нельзя упускать.

Условные обозначения

В этой книге вы найдете ряд текстовых стилей, которые выделяют различные виды информации. Вот некоторые примеры этих стилей и объяснение их значения.

Кодовые слова в тексте, имена таблиц баз данных, папок, файлов, расширения файлов, пути, ввод данных пользователем и дескрипторы социальной сети Twitter показаны следующим образом: «И уже установленные пакеты могут быть обновлены при помощи флага `--upgrade`».

Фрагмент исходного кода оформляется следующим образом:

```
import matplotlib.pyplot as plt
import numpy as np
y = df.iloc[0:100, 4].values
y = np.where(y == 'Iris-setosa', -1, 1)
X = df.iloc[0:100, [0, 2]].values
plt.scatter(X[:50, 0], X[:50, 1],
            color='red', marker='x', label='setosa')
plt.scatter(X[50:100, 0], X[50:100, 1],
            color='blue', marker='o', label='versicolor')
plt.xlabel('длина чашелистика')
plt.ylabel('длина лепестка')
plt.legend(loc='upper left')
plt.show()
```

При этом если исходный код содержит результат выполнения, то он предваряется значком консоли интерпретатора Python (>>>):

```
>>>
df.isnull().sum()
A      0
B      0
C      1
D      1
dtype: int64
```

Любой ввод или вывод в командной строке записывается следующим образом:

```
> dot -Tpng tree.dot -o tree.png
```

Новые термины и важные слова показаны полужирным шрифтом. Слова, которые вы видите на экране, например в меню или диалоговых окнах, выглядят в тексте следующим образом: «После щелчка по кнопке **Dashboard** в верхнем правом углу мы получим доступ к панели управления, показанной в верхней части страницы».



Предупреждения или важные примечания появляются в этом поле.



Подсказки и приемы появляются тут.



Дополнения к тексту оригинала книги.

Отзывы читателей

Отзывы наших читателей всегда приветствуются. Сообщите нам, что вы думаете об этой книге – что вам понравилось, а что нет. Обратная связь с читателями для нас очень важна, поскольку она помогает нам формировать названия книг, из которых вы действительно получите максимум полезного.

Отзыв по общим вопросам можно отправить по адресу dmkpress@gmail.com, упомянув заголовок книги в теме вашего электронного сообщения.

Если же речь идет о теме, в которой вы профессионально осведомлены, и вы интересуетесь написанием либо содействием в написании книги, то также напишите письмо по адресу dmkpress@gmail.com.

Служба поддержки

Теперь, когда вы являетесь довольным владельцем книги издательства «ДМК Пресс», мы предложим вам ряд возможностей с целью помочь вам получать максимум от своей покупки.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф на странице с описанием соответствующей книги.

Опечатки

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки всё равно случаются. Если вы найдёте ошибку в одной из наших книг — возможно, ошибку в тексте или в коде — мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдёте какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательство ДМК Пресс и Раскт очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в Интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, и помогающую нам предоставлять вам качественные материалы.

Вопросы

Если у вас есть вопрос по каким-либо аспектам этой книги, то вы можете связаться с нами по электронному адресу dmkpress@gmail.com. и мы приложим все усилия, чтобы решить ваш вопрос.

Комментарий переводчика

Весь материал книги приведен в соответствие с последними действующими версиями библиотек (время перевода книги – октябрь-ноябрь 2016 г.), дополнен свежей информацией и протестирован в среде Windows 10 и Fedora 24. При тестировании исходного кода за основу взят Python версии 3.5.2.

Большинство содержащихся в книге технических терминов и аббревиатур для удобства кратко определено в сносках, а для некоторых терминов в силу отсутствия единой терминологии приведены соответствующие варианты наименований или пояснения. Книга дополнена «Глоссарием основных терминов и сокращений». В приложении к книге особое внимание уделено оценке работоспособности машинно-обучаемых моделей.

Несколько замечаний по поводу терминологии. В русскоязычной литературе по машинному обучению термин «обучение» нередко перегружен. Он обозначает сразу 5 англоязычных терминов – learning, train, fitting и производные от fitting – underfitting и overfitting, что затрудняет восприятие материала. В тексте перевода эти термины разделены – соответственно обучение, тренировка, подгонка, недоподгонка и переподгонка – и даны пояснения. Такой перевод не является каким-то нововведением и широко используется. «Learning» – это целенаправленное получение знаний извне, извлечение знаний, направленность на себя, «training» – передача знаний вовне, направленность на объект. В настоящем переводе они переведены соответственно как «обучение» и «тренировка». Термины «overfitting» и «underfitting» нередко переводятся как «переобучение» и «недообучение», что неверно. Согласно толковому словарю Webster, ближайшим по смыслу переводом термина «fitting» будет «подгонка, адаптация, настройка, аппроксимация», и соответственно «overfitting» и «underfitting» – это «переподгонка» и «недоподгонка»; речь не об обучении, а об излишней или недостаточной адаптации модели под тренировочные данные, ее аппроксимации.

На веб-сайте Github имеется веб-страница книги, где содержатся обновляемые исходные коды прилагаемых к книге программ, много дополнительных материалов и ссылок, а также обширный раздел часто задаваемых вопросов (<https://github.com/rasbt/python-machine-learning-book/tree/master/faq>), наиболее интересная, по моему мнению, и лишь незначительная часть ответов на которые сведена в *приложении А* к данной книге.

Прилагаемый к книге адаптированный и скорректированный исходный код примеров должен находиться в подпапке домашней папки пользователя (`/home/ваши_проекты_Python` или `C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\ваши_проекты_Python`). Ниже приведена структура папки с прилагаемыми примерами и дополнительной информацией:

- bonus** Дополнительные материалы, документация и другие примеры.
- ch01-ch13** Исходный код примеров в виде записных книжек Jupyter и сценарных файлов Python.
- data** Наборы данных, используемых в книге и примерах.

- faq** Копия раздела часто задаваемых вопросов по книге и машинному обучению репозитория Github на английском языке.
- fonts** Шрифты, используемые для оформления записных книжек, книги и графиков.

Для просмотра исходного кода примеров лучше всего пользоваться записными книжками Jupyter. Они более читабельны, содержат графики, цветные рисунки и расширенные пояснения.

Для оформления графиков и диаграмм использовался шрифт Ubuntu Condensed, который прилагается к книге. Его можно найти в папке fonts. В Windows для установки шрифта в операционную систему следует правой кнопкой мыши нажать на файле шрифта и выбрать из контекстного меню **Установить**.

Далее приведены особенности установки некоторых используемых программных библиотек Python.

Особенности программных библиотек

Numpy+MKL привязана к библиотеке Intel® Math Kernel Library и включает в свой состав необходимые динамические библиотеки (DLL) в каталоге numpy.core. Для работы SciPy и Scikit-learn в Windows требуется, чтобы в системе была установлена библиотека Numpy+MKL. Ее следует скачать из репозитория whl-файлов (<http://www.lfd.uci.edu/~gohlke/pythonlibs/>) и установить (например, `pip3 install numpy-1.11.2+mkl-cp35-cp35m-win_amd64.whl` для 64-разрядного компьютера) как whl.

- ☞ **NumPy** – основополагающая библиотека, необходимая для научных вычислений на Python.
- ☞ **Matplotlib** – библиотека для работы с двумерными графиками. Требуется наличие numpy и некоторых других.
- ☞ **Pandas** – инструмент для анализа структурных данных и временных рядов. Требуется наличие numpy и некоторых других.
- ☞ **Scikit-learn** – интегратор классических алгоритмов машинного обучения. Требуется наличие numpy+mkl.
- ☞ **SciPy** – библиотека, используемая в математике, естественных науках и инженерном деле. Требуется наличие numpy+mkl.
- ☞ **Jupyter** – интерактивная вычислительная среда.

Факультативно:

- ☞ **Spyder** – инструментальная среда программирования на Python.
- ☞ **Mlxtend** – библиотека модулей расширения и вспомогательных инструментов для программных библиотек Python, предназначенных для анализа данных и машинного обучения (автор С. Рашка) для решения каждодневных задач в области науки о данных. Используется в приложении и дополнительных материалах к книге (<https://github.com/rasbt/mlxtend>). Документацию по библиотеке можно найти в папке bonus.
- ☞ **PyQt5**, библиотека программ для программирования визуального интерфейса, требуется для работы инструментальной среды программирования Spyder

Протокол установки программных библиотек

```
python -m pip install --upgrade pip

pip3 install numpy
    либо как whl: pip3 install numpy-1.11.2+mkl-cp35-cp35m-win_amd64.whl
pip3 install matplotlib
pip3 install pandas
pip3 install scikit-learn
    либо как whl: pip3 install scikit_learn-0.18.1-cp35-cp35m-win_amd64.whl
pip3 install scipy
    либо как whl: pip3 install scipy-0.18.1-cp35-cp35m-win_amd64-any.whl
pip3 install jupyter
pip3 install theano
pip3 install keras
pip3 install nltk
pip3 install seaborn
pip3 install flask
pip3 install pyprind
pip3 install wtforms
    факультативно:
pip3 install mlxtend
pip3 install pyqt5
pip3 install spyder
```

Примечание: в зависимости от базовой ОС, версий языка Python и версий программных библиотек устанавливаемые вами версии whl-файлов могут отличаться от приведенных выше, где показаны последние на декабрь 2016 г. версии для 64-разрядной ОС Windows и Python версии 3.5.2

Установка библиотек Python из whl-файла

Библиотеки для Python можно разрабатывать не только на чистом Python. Довольно часто библиотеки пишутся на C (динамические библиотеки), и для них пишется обертка Python, или же библиотека пишется на Python, но для оптимизации узких мест часть кода пишется на C. Такие библиотеки получаются очень быстрыми, однако библиотеки с вкраплениями кода на C программисту на Python тяжелее установить ввиду банального отсутствия соответствующих знаний либо необходимых компонентов и настроек в рабочей среде (в особенности в Windows). Для решения описанных проблем разработан специальный формат (файлы с расширением .whl) для распространения библиотек, который содержит заранее скомпилированную версию библиотеки со всеми ее зависимостями. Формат whl поддерживается всеми основными платформами (Mac OS X, Linux, Windows).

Установка производится с помощью менеджера пакетов pip. В отличие от обычной установки командой `pip3 install <имя_библиотеки>`, вместо имени библиотеки указывается путь к whl-файлу `pip3 install <путь/к/whl_файлу>`. Например:

```
pip3 install C:\temp\networkx-1.11-py2.py3-none-any.whl
```

Откройте окно командой строки и при помощи команды `cd` перейдите в каталог, где размещен ваш whl-файл. Просто скопируйте туда имя вашего whl-файла. В этом случае полный путь указывать не понадобится. Например:

```
pip3 install networkx-1.11-py2.py3-none-any.whl
```

При выборе библиотеки важно, чтобы разрядность устанавливаемой библиотеки и разрядность интерпретатора совпадали. Пользователи Windows могут брать whl-файлы на веб-странице <http://www.lfd.uci.edu/~gohlke/pythonlibs/> Кристофа Голька из Лаборатории динамики флуоресценции Калифорнийского университета в г. Ирвайн. Библиотеки там постоянно обновляются, и в архиве содержатся все, какие только могут понадобиться.

Установка и настройка инструментальной среды Spyder

Spyder – это инструментальная среда для научных вычислений для языка Python (Scientific PYthon Development EnviRonment) для Windows, Mac OS X и Linux. Это простая, легковесная и бесплатная интерактивная среда разработки на Python, которая предлагает функционал, аналогичный среде разработки на MATLAB, включая готовые к использованию виджеты **PyQt5** и **PySide**: редактор исходного кода, редактор массивов данных **NumPy**, редактор словарей, консоли Python и IPython и многое другое.

Чтобы установить среду Spyder в Ubuntu Linux, используя официальный менеджер пакетов, нужна всего одна команда:

```
sudo apt-get install spyder3
```

Чтобы установить с использованием менеджера пакетов `pip`:

```
sudo apt-get install python-qt5 python-sphinx  
sudo pip3 install spyder
```

И чтобы обновить:

```
sudo pip3 install -U spyder
```

Установка среды Spyder в Fedora 24:

```
dnf install python3-spyder
```

Во всех вышеперечисленных случаях речь идет о версии Spyder для Python 3 (на момент инсталляции это был Python 3.5.2).

Установка среды Spyder в Windows:

```
pip3 install spyder
```

Конец ознакомительного фрагмента.

Приобрести книгу можно
в интернет-магазине «Электронный универс»
(e-Univers.ru)