- Ранжирование — целевая переменная y представляет собой порядок элементов внутри группы, например порядок страниц на странице результатов поиска. Проблема ранжирования часто возникает в таких областях, как поиск и рекомендации, но эта тема выходит за рамки данной книги, и мы не будем рассматривать ее подробно.

Каждую задачу контролируемого обучения можно решить с помощью различных алгоритмов. Нам доступно множество типов моделей. Эти модели определяют, как именно функция g учится прогнозировать y на основе X. Модели включают в себя:

- линейную регрессию для решения задачи регрессии (описывается в главе 2);
- логистическую регрессию для решения задачи классификации (описывается в главе 3);
- древовидные модели для решения задач как регрессии, так и классификации (описываются в главе 6);
- нейронные сети для решения как регрессионных задач, так и задач классификации (описываются в главе 7).

Глубокому обучению и нейронным сетям в последнее время уделяется особое внимание, в основном благодаря прорыву в методах компьютерного зрения. Эти сети решают такие задачи, как классификация изображений, намного лучше, чем это делали более ранние методы. *Глубокое обучение* — подобласть машинного обучения, в которой функция g представляет собой нейронную сеть со многими слоями. Мы узнаем больше о нейронных сетях и глубоком обучении, начиная с главы 7, где обучаем модель глубокого обучения для классификации изображений.

1.2. ПРОЦЕСС МАШИННОГО ОБУЧЕНИЯ

Создание системы машинного обучения включает в себя больше, чем просто выбор модели, ее обучение и применение к новым данным. Обучение модели — лишь часть, небольшой шаг в этом процессе.

Будет много других шагов, таких как определение проблемы, которую может решить машинное обучение, и использование прогнозов модели для воздействия на конечных пользователей. Более того, процесс является итеративным. Обучая модель и применяя ее к новому набору данных, мы часто выявляем случаи, в которых модель работает недостаточно хорошо. Мы используем их для переобучения модели таким образом, чтобы новая версия лучше справлялась с подобными ситуациями.

Определенные методы и фреймворки помогают нам организовать проект машинного обучения так, чтобы он не выходил из-под контроля. Одним из таких фреймворков служит CRISP-DM, который расшифровывается как Cross-Industry Standard Process for Data Mining — *межотраслевой стандартный процесс интеллектуального анализа данных*. Он был изобретен довольно давно, в 1996 году, но, несмотря на возраст, все еще применим к сегодняшним задачам.

Согласно CRISP-DM (рис. 1.9) процесс машинного обучения состоит из шести этапов:

1. Бизнес-анализ.
2. Анализ данных.
3. Подготовка данных.
4. Моделирование.
5. Оценка.
6. Развертывание.

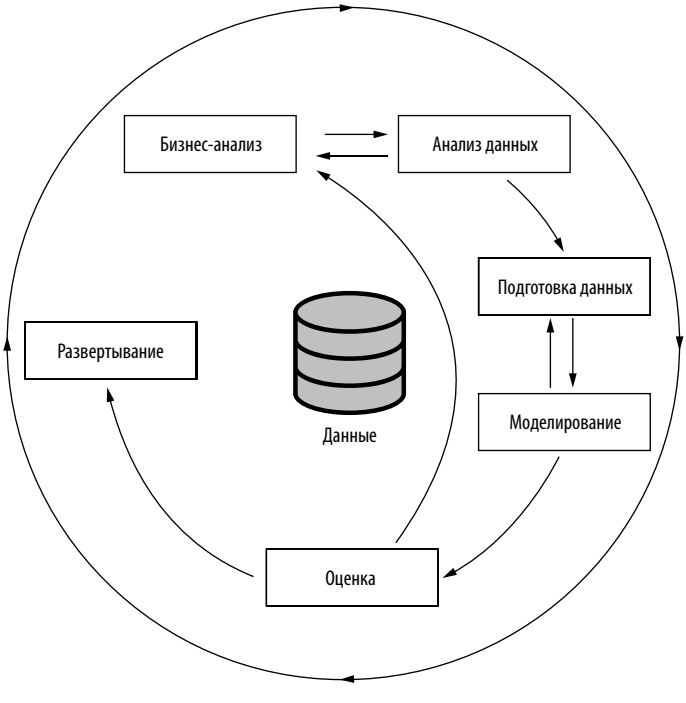

Рис. 1.9. Процесс CRISP-DM. Проект машинного обучения начинается с понимания проблемы, а затем переходит к подготовке данных, обучению модели и оценке результатов. Наконец модель добирается до этапа развертывания. Процесс является итеративным, и на каждом шаге можно вернуться к предыдущему

Каждый этап охватывает типичные задачи:

- на этапе бизнес-анализа мы пытаемся выразить задачу, понять, как мы можем ее решить, и определить, поможет ли нам в этом машинное обучение;
- на этапе анализа данных мы анализируем доступные наборы данных и решаем, нужно ли нам собирать больше данных;
- на этапе подготовки данных мы преобразуем данные в табличную форму, которую можно использовать в качестве входных данных для модели машинного обучения;
- когда данные подготовлены, мы переходим к этапу моделирования, на котором обучаем модель;
- после определения наилучшей модели наступает этап оценки, на котором мы оцениваем модель, чтобы понять, решает ли она исходную бизнес-задачу, и оцениваем ее успешность на этом поприще;
- наконец на этапе развертывания мы развертываем модель в производственной среде.

1.2.1. Бизнес-анализ

Рассмотрим пример обнаружения спама для поставщика услуг электронной почты. Мы видим больше спам-сообщений, чем когда-либо прежде, и наша нынешняя система не может с этим справиться. К данной задаче мы обращаемся на этапе бизнес-анализа: анализируем проблему и существующее решение, после чего пытаемся определить, поможет ли внедрение машинного обучения в эту систему остановить спам-сообщения. Мы также определяем цель и способы ее измерения.

Целью может быть, например, «уменьшить количество полученных спам-сообщений» или «уменьшить количество жалоб на спам, которые служба поддержки клиентов получает за день». На данном этапе мы также можем решить, что машинное обучение не поможет, и предложить более простой способ решения задачи.

1.2.2. Анализ данных

Следующий шаг — анализ данных. Здесь мы попытаемся определить источники данных, которые можем использовать для решения задачи. Например, если на нашем сайте есть кнопка «Сообщить о спаме», то мы можем получить данные, сгенерированные пользователями, которые отметили свои входящие электронные письма как спам. Затем мы смотрим на данные и пытаемся понять, подходят ли они для решения нашей проблемы.

Однако эти данные могут не в полной мере подходить по целому ряду причин. Одной из них может быть то, что набор слишком мал, чтобы извлечь какие-либо полезные закономерности. Другой причиной может быть то, что данные слишком зашумлены. Пользователи могут неправильно использовать кнопку, поэтому она будет бесполезна для обучения модели машинного обучения, или же процесс сбора данных может быть нарушен, и в итоге будет собрана лишь небольшая часть нужных нам данных.

Если мы придем к выводу, что имеющихся у нас в настоящее время данных недостаточно, то нам потребуется найти способ получить более качественные данные, независимо от того, получаем мы их из внешних источников или совершенствуем способ их сбора внутри компании. К тому же открытия, которые мы совершим на данном этапе, могут повлиять на цель, поставленную на этапе, повлияют на цель, которую мы поставили на этапе бизнес-анализа, поэтому нам, возможно, придется вернуться к этому шагу и скорректировать цель в соответствии с выводами.

Когда у нас есть надежные источники данных, мы переходим к этапу подготовки данных.

1.2.3. Подготовка данных

Здесь мы очищаем данные, преобразуя их так, чтобы использовать в качестве входных для модели машинного обучения. В примере со спамом мы преобразуем набор данных в набор признаков, которые позже вводим в модель.

После того как данные подготовлены, мы переходим к этапу моделирования.

1.2.4. Моделирование

На этом этапе мы решаем, какую модель машинного обучения использовать и как убедиться, что мы извлекаем из нее максимум пользы. Например, чтобы решить проблему спама, мы можем попробовать логистическую регрессию и глубокую нейронную сеть.

Нам нужно знать, как измерить производительность моделей и выбрать наилучшую из них. Что касается модели спам-фильтра, то мы можем посмотреть, насколько хорошо модель прогнозирует спам-сообщения, и выбрать ту, которая делает это лучше других. Для этой цели важно установить надлежащую структуру проверки, поэтому несколько позже мы рассмотрим данную задачу более подробно.

Весьма вероятно, что на текущем этапе нам придется вернуться и скорректировать способ подготовки данных. Возможно, мы выделили отличный признак, поэтому возвращаемся к этапу подготовки данных, чтобы написать некий код для вычисления этого признака. Когда код готов, мы снова обучаем модель, чтобы проверить, подходит ли признак. Например, мы могли бы добавить признак «длина темы письма», переобучить модель и проверить, улучшает ли это изменение производительность модели.

Выбрав наилучшую из возможных моделей, мы переходим к этапу оценки.

1.2.5. Оценка

На данном этапе мы проверяем, соответствует ли модель ожиданиям. Ставя цель на этапе бизнес-анализа, мы также продумываем способ определения того, будет ли цель достигнута. Как правило, мы делаем это, просматривая некую важную бизнес-метрику и убеждаясь, что модель перемещает метрику в нужном направлении. В случае обнаружения спама метрикой может служить количество людей, которые нажимают кнопку «Сообщить о спаме», или количество жалоб на решаемую проблему, полученных службой поддержки. В обоих случаях мы надеемся, что использование модели сократит их количество.

В настоящее время этот шаг тесно связан со следующим — развертыванием.

1.2.6. Развертывание

Лучший способ оценить модель — устроить ей боевое крещение: протестировать ее на небольшом количестве пользователей, а затем проверить, изменилась ли наша бизнес-метрика для этих пользователей. Например, если мы хотим, чтобы наша модель уменьшила количество зарегистрированных спам-сообщений, то ожидаем увидеть меньше сообщений от этой группы по сравнению с остальными пользователями.

После развертывания модели мы используем все, что узнали на предыдущих этапах, и возвращаемся к первому шагу, чтобы поразмышлять о достигнутом (или недостигнутом). Может оказаться, что наша первоначальная цель была неправильной и что на самом деле мы хотим добиться *не* сокращения количества жалоб, а повышения вовлеченности клиентов за счет уменьшения количества спама. Поэтому мы возвращаемся к этапу бизнес-анализа и переопределяем нашу цель. Затем, при повторной оценке модели, мы уже используем другую бизнес-метрику для измерения ее качества.

1.2.7. Повтор

Как видите, CRISP-DM делает упор на итеративный характер процессов машинного обучения: по окончании последнего шага от нас ожидается возвращение к первому, уточнение исходной задачи и изменение ее на основе полученной информации. Мы никогда не останавливаемся на последнем шаге; вместо этого мы переосмысливаем проблему и стараемся понять, что можно улучшить на следующей итерации.

Распространенное заблуждение состоит в том, что инженеры по машинному обучению и специалисты по обработке данных целыми днями только и делают, что обучают модели машинного обучения. В действительности это не так, что можем легко заметить на схеме CRISP-DM (см. рис. 1.9). До и после этапа моделирования выполняется множество шагов, и все они важны для успешного проекта машинного обучения.

1.3. МОДЕЛИРОВАНИЕ И ПРОВЕРКА МОДЕЛИ

Как мы видели ранее, обучение моделей (этап моделирования) — только один шаг во всем процессе. Но он важен, поскольку именно здесь мы на самом деле используем машинное обучение для обучения моделей.

Собрав все необходимые данные и убедившись, что они достаточно хороши, мы ищем способ обработки данных, а затем приступаем к обучению модели машинного обучения.

В нашем примере со спамом это происходит после того, как мы получим все отчеты о спаме, обработаем электронные письма и подготовим матрицу для использования в модели.

На этом этапе может встать вопрос, что использовать — логистическую регрессию или нейронную сеть. Если мы решим использовать нейронную сеть, поскольку где-то услышали, что это лучшая модель, то как нам убедиться, что она действительно лучше любой другой?

Цель данного этапа — создать модель таким образом, чтобы она обеспечивала наилучшие прогностические характеристики. Для этого нам нужен способ надежно измерить производительность каждой возможной модели-кандидата, а затем выбрать наилучшую.

Один из возможных подходов заключается в обучении модели, запуске ее в живой системе и наблюдении за тем, что произойдет. В примере со спамом мы решили использовать нейронную сеть для обнаружения спама, поэтому мы обучаем ее и внедряем в нашу производственную систему. Затем наблюдаем,

как модель ведет себя при новых сообщениях, и регистрируем случаи, когда система работает некорректно.

Однако такой подход в нашем случае не идеален: мы не можем проделать это для каждой имеющейся у нас модели-кандидата. Что еще хуже, мы можем ненароком развернуть действительно плохую модель и увидеть, что она плохая, только после того, как она опробована на живых пользователях нашей системы.

ПРИМЕЧАНИЕ

Тестирование модели в живой системе называется онлайн-тестированием и служит важным этапом оценки качества модели на реальных данных. Однако этот подход относится к этапам оценки и развертывания процесса, а не к этапу моделирования.

Лучший способ выбрать наилучшую модель перед развертыванием — эмуляция сценария запуска в эксплуатацию. Мы получаем наш полный набор данных, отбираем из него часть и обучаем модель на остатке. Когда обучение завершено, мы делаем вид, что сохраненный набор данных — это новые данные, и используем его для оценки производительности наших моделей. Эту часть данных часто называют *проверочным набором*, а процесс удаления части набора данных и использования его для оценки производительности называется *проверкой* (рис. 1.10).

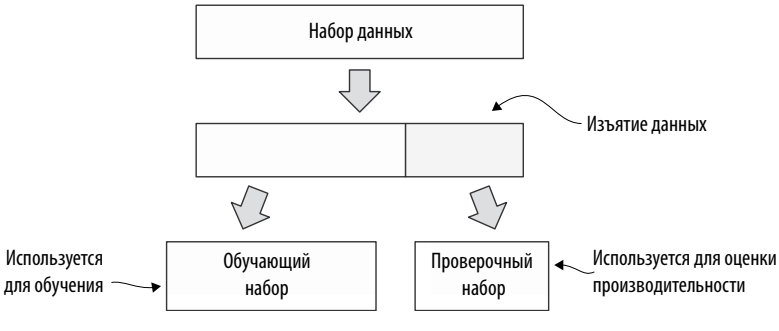

Рис. 1.10. Чтобы оценить производительность модели, мы отделяем часть данных и используем их только для проверки

В наборе данных о спаме мы можем изъять каждое десятое сообщение. Таким образом, мы сохраняем 10 % данных, которые используем только для проверки моделей, а остальные 90 % применяем для обучения.

Далее на основе обучающих данных мы обучаем как логистическую регрессию, так и нейронную сеть. Когда модели обучены, мы применяем их к проверочному набору данных и выясняем, какая из них более точно прогнозирует спам.

Если после применения моделей к проверочному набору мы видим, что логистическая регрессия справляется с прогнозированием спама только в 90 % случаев, тогда как нейронная сеть — в 93 % случаев, то приходим к выводу, что модель нейронной сети более результативна, чем логистическая регрессия (рис. 1.11).

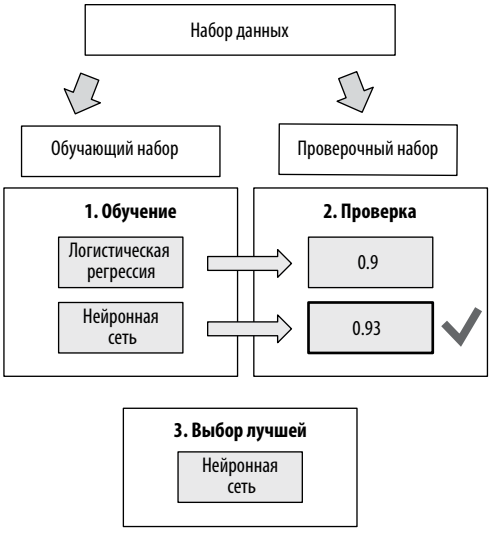

Рис. 1.11. Процесс проверки. Мы разделяем набор данных на две части, обучаем модели на обучающей части и оцениваем производительность на проверочной. Используя результаты оценки, мы можем выбрать наилучшую модель

Часто у нас в наличии оказываются не две модели, а гораздо больше. Логистическая регрессия, например, имеет параметр C, и в зависимости от заданного значения, результаты могут сильно различаться. Аналогично нейронная сеть имеет множество параметров, и каждый из них может очень сильно повлиять на прогностические характеристики конечной модели. Кроме того, есть и другие модели, каждая со своим набором параметров. Как нам выбрать лучшую модель с наилучшими параметрами?

Для этого мы используем ту же схему оценки. Мы обучаем модели с различными параметрами на обучающих данных, применяем их к проверочным данным, а затем выбираем модель и ее параметры на основе наилучших результатов этапа проверки (рис. 1.12).

Однако у этого подхода есть один нюанс. Если мы раз за разом повторяем процесс оценки модели и используем для этой цели один и тот же проверочный набор данных, то хорошие цифры, которые мы наблюдаем в наборе данных проверки, могут оказаться лишь следствием случайности. Другими словами, «лучшей» модели могло просто повезти с прогнозированием результатов для этого конкретного набора данных.

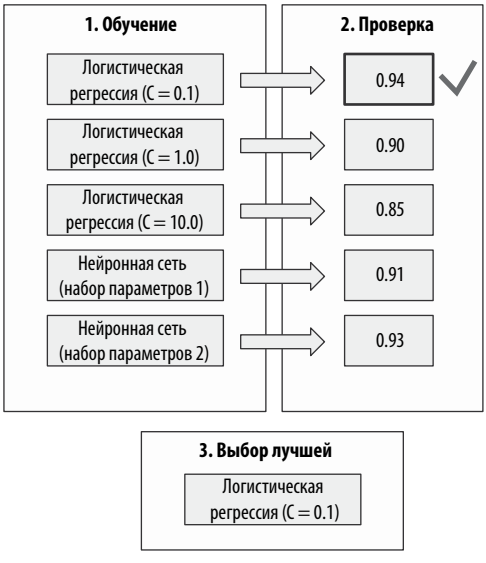

Рис. 1.12. Использование проверочного набора данных для выбора наилучшей модели с наилучшими параметрами

ПРИМЕЧАНИЕ

В статистике и других областях эта проблема известна как проблема множественных сравнений или проблема множественных тестов. Чем больше раз мы выполняем прогнозы на одном и том же наборе данных, тем больше вероятность того, что мы случайно увидим хорошую производительность.

Чтобы избежать этой проблемы, мы используем ту же идею: снова изымем часть данных. Эту часть назовем *тестовым* набором данных. Мы будем изредка использовать его для тестирования модели, которую выбрали как лучшую (рис. 1.13).

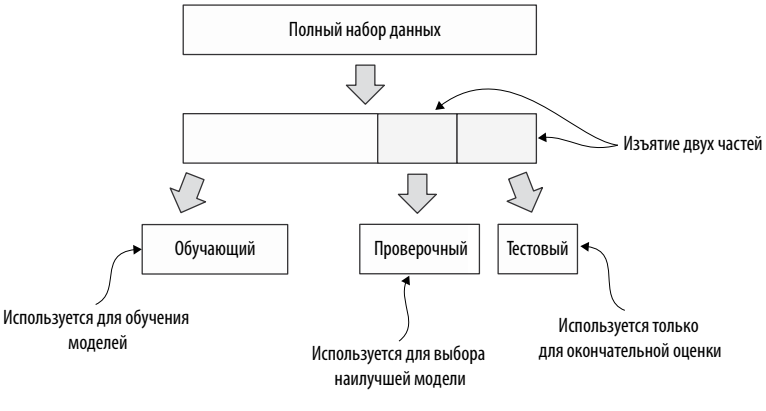

Рис. 1.13. Разделение данных на обучающую, тестовую и проверочную части

Чтобы применить этот подход к примеру со спамом, сначала мы сохраним 10 % данных в качестве тестового набора, после чего сохраним еще 10 % в качестве проверочного. Мы опробуем несколько моделей на проверочном наборе данных, выберем лучшую и применим ее уже к тестовому набору. Если при этом разница в производительности между проверкой и тестированием невелика, то можно утверждать, что эта модель действительно наилучшая (рис. 1.14).

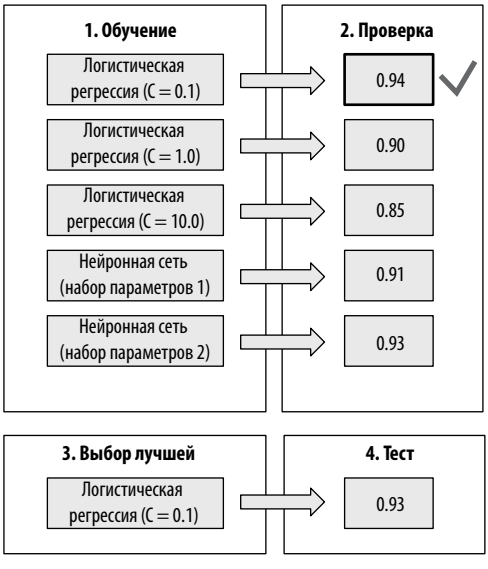

Рис. 1.14. Использование тестового набора данных для подтверждения того, что производительность наилучшей модели на проверочном наборе является хорошей

ВАЖНО

Настройка процесса проверки — наиболее важный шаг в машинном обучении. Без этого нет надежного способа выяснить, является ли только что обученная модель хорошей, бесполезной или даже вредной.

Процесс выбора наилучшей модели и наилучших параметров для модели называется *выбором модели*. Мы можем обобщить выбор модели следующим образом (рис. 1.15).

1. Мы разделяем данные на части для обучения, проверки и тестирования.
2. Сначала мы обучаем каждую модель на обучающей части, а затем оцениваем на проверочной.

3. Каждый раз, обучая новую модель, мы записываем результаты оценки, используя проверочную часть.
4. В конце мы определяем, какая модель является лучшей, и тестируем ее на тестовом наборе данных.

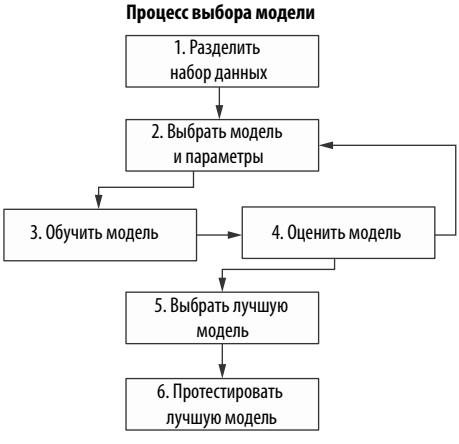

Рис. 1.15. Сначала мы разбиваем набор данных, выбираем модель и обучаем ее только на обучающей части данных. Затем, на этапе проверки, мы оцениваем модель. Мы повторяем этот процесс многократно, пока не обнаружим лучшую модель

Важно использовать процесс выбора модели, прежде всего проверив и протестировав модели в автономном режиме, чтобы убедиться, что обучаемые модели вполне пригодны. Если модель хорошо работает в автономном режиме, то мы можем принять решение о переходе к следующему шагу и развернуть ее, чтобы оценить ее производительность на реальных пользователях.

РЕЗЮМЕ

- В отличие от традиционных систем разработки программного обеспечения на основе правил, в которых правила извлекаются и кодируются вручную, системы машинного обучения можно научить извлекать значимые закономерности из данных автоматически. Это в разы увеличивает уровень гибкости и облегчает адаптацию к изменениям.
- Для успешной реализации проекта машинного обучения требуется структура и набор руководящих принципов. CRISP-DM — платформа для организации проекта машинного обучения, которая разбивает процесс на шесть этапов, от бизнес-анализа до развертывания. Фреймворк ориенти-

рован на итеративный характер машинного обучения и помогает нам с его организацией.

- Моделирование — важный шаг в проекте машинного обучения: та часть, где мы фактически используем машинное обучение для обучения модели. На данном этапе рождаются модели, которые обеспечивают наилучшую прогностическую производительность.
- Выбор модели представляет собой процесс выбора наилучшей модели для решения задачи. Мы разделили все доступные данные на три части: обучающую, проверочную и тестовую. Мы обучаем модели на обучающем наборе и выбираем лучшую с помощью проверочного. Когда лучшая модель выбрана, мы используем этап тестирования, чтобы выполнить окончательную проверку и убедиться, что лучшая модель работает хорошо. Этот процесс помогает создавать полезные модели, которые хорошо работают, не преподнося неприятных сюрпризов.