

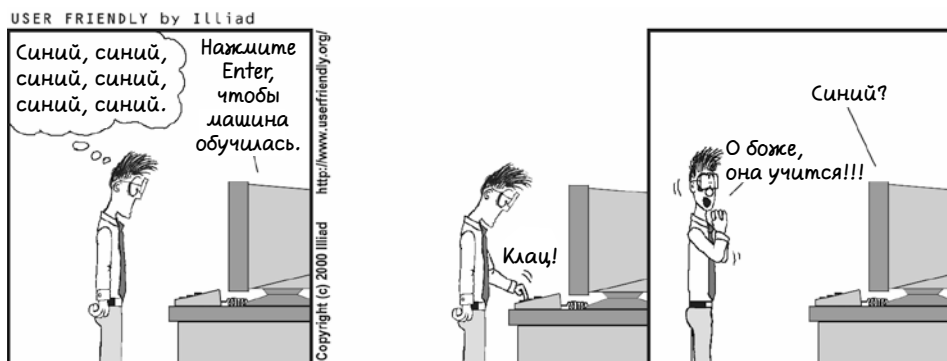
1

Что такое машинное обучение? Это здравый смысл, проявляемый компьютером



В этой главе

- ✓ Что такое машинное обучение.
 - ✓ Сложно ли машинное обучение (спойлер: нет).
 - ✓ Чему мы научимся в этой книге.
 - ✓ Что такое искусственный интеллект и чем он отличается от машинного обучения.
 - ✓ Как мыслят люди и как можно внедрить эти идеи в машину.
 - ✓ Некоторые базовые примеры машинного обучения из реальной жизни.
-



Я с радостью присоединяюсь к вам на пути к знаниям!

Приветствую вас на страницах этой книги! Я искренне рад присоединиться к вам на пути к пониманию машинного обучения. На высоком уровне машинное обучение — это процесс, в ходе которого компьютер решает проблемы и принимает решения. Почти как человек.

Через эту книгу я хочу донести до вас одну мысль: машинное обучение — это просто! Чтобы разобраться с ним, не нужно быть великим математиком или программистом. Базовая математика, конечно, понадобится, но основными составляющими послужат здравый смысл, хорошая визуальная интуиция и желание изучать и применять указанные методы ко всему, чем вы увлечены, и в области, где хотите что-то улучшить. Я получил огромное удовольствие от написания книги, так как мне понравилось углубляться в данную тему, и я надеюсь, что вы получите огромное удовольствие, прочитав ее и погрузившись в машинное обучение!

Машинное обучение повсюду

Машинное обучение можно встретить везде. Это утверждение с каждым днем кажется все более верным. Мне трудно представить себе хоть один аспект жизни, который нельзя было бы так или иначе улучшить с помощью машинного обучения. В любом деле, требующем повторения, анализа данных и сбора выходных данных, вам поможет машинное обучение. За последние несколько лет благодаря достижениям в области вычислительных мощностей и повсеместному сбору данных в сфере машинного обучения наблюдался поистине взрывной рост. Назову лишь несколько областей применения машинного обучения: системы рекомендаций, распознавание изображений, обработка текста, автопилот в автомобилях, выявление спама, постановка медицинских диагнозов... Список можно продолжать бесконечно. Возможно, у вас есть задача или область,

в которой требуется ощутимый прогресс. Весьма вероятно, что машинное обучение вполне применимо в этой области — и именно это привело вас к этой книге. Давайте вместе с этим разберемся!

Нужен ли большой опыт в математике и программировании, чтобы понять машинное обучение?

Нет. Машинное обучение требует воображения, творческого подхода и визуального мышления. Оно заключается в том, чтобы улавливать закономерности, которые появляются в мире, и использовать их для прогнозирования событий будущего. Если вам нравится находить закономерности и выявлять корреляции, тогда машинное обучение — ваша область. Если бы я сказал вам, что бросил курить, стал есть больше овощей и занялся спортом, то что, согласно вашему прогнозу, произойдет с моим здоровьем через год? Возможно, оно улучшится. Если бы я сказал вам, что перехожу с красных свитеров на зеленые, то что, по-вашему, случится с моим здоровьем через год? Скорее всего, мало что изменится (может, случатся грандиозные перемены, но не из-за того, о чем я сообщил). Выявление таких корреляций и закономерностей — вот что такое машинное обучение. Единственное различие машинного обучения состоит в том, что здесь мы привязываем к этим шаблонам формулы и цифры, чтобы компьютеры могли их распознавать.

Для изучения машинного обучения необходимы некоторые знания математики и программирования, но экспертом быть вовсе не обязательно. Если вы *уже* эксперт в любой из этих областей (или даже в обеих), то эти навыки наверняка вам пригодятся. Но если пока это не так, вы все равно сможете изучить машинное обучение и восполнить пробелы в математике и программировании по ходу дела. В этой книге мы представим все необходимые математические концепции в тот момент, когда они нам понадобятся. Если говорить о кодировании, то объем кода, который потребуется в машинном обучении, зависит только от вас. В машинном обучении есть задания как для тех, кто программирует весь день напролет, так и для тех, кто не программирует вообще. Многие пакеты, API и инструменты помогают нам реализовать машинное обучение с минимальным объемом кода. С каждым днем машинное обучение становится все более общедоступным, и я рад, что вы не остались в стороне!

Формулы и код — это весело, если их рассматривать как язык

В большинстве книг по машинному обучению алгоритмы объясняются математически с использованием формул, производных и т. д. Хотя такие точные описания методов хорошо работают на практике, сама по себе формула может быть скорее запутанной, чем иллюстративной. Однако, подобно музыкальной партитуре, за запутанностью формулы может скрываться чудесная мелодия.

Например, взглянем на формулу $\sum_{i=1}^4 i$. На первый взгляд она выглядит некрасиво, но представляет собой простейшее суммирование, а именно $1 + 2 + 3 + 4$.

А как насчет $\sum_{i=1}^n w_i$? Это просто сумма множества (n) чисел. Но когда я думаю о сумме многих чисел, то представляю что-то вроде $3 + 2 + 4 + 27$, а не $\sum_{i=1}^n w_i$. Всякий раз, когда я вижу формулу, мне сразу же нужен небольшой пример, и тогда картина в моем сознании проясняется. Что приходит на ум, когда я вижу что-то вроде $P(A|B)$? Это условная вероятность, поэтому я думаю о чем-то вроде: «Вероятность того, что событие A произойдет, с учетом того, что другое событие, B , уже произошло». Например, если A представляет собой сегодняшний дождь, а B — жизнь в тропическом лесу Амазонии, то формула $P(A|B) = 0,8$ попросту означает: «Вероятность того, что сегодня пойдет дождь, учитывая, что мы живем в тропическом лесу Амазонии, составляет 80 %».

Если вы действительно любите формулы, не волнуйтесь — в книге их достаточно. Но появляться они будут сразу после примера, который их иллюстрирует.

То же самое будет и с кодом. Если мы смотрим на код со стороны, он может казаться сложным, и бывает трудно поверить, что кто-то может уместить все это в своей голове. Однако код — это лишь последовательность шагов, и, как правило, каждый из них прост. В этой книге мы будем писать код, но разобьем его на простые шаги, каждый из которых подробно объясним на примерах либо с помощью иллюстраций. В первых нескольких главах мы будем кодировать свои модели с нуля, чтобы лучше понимать, как они работают. Однако в последующих главах модели усложнятся. Для них мы станем использовать такие пакеты, как Scikit-Learn, Turi Create или Keras, в которых точно и эффективно реализовано большинство алгоритмов машинного обучения.

Так что же такое машинное обучение?

Чтобы дать определение машинному обучению, сначала определим более общий термин — искусственный интеллект (ИИ).

Что такое искусственный интеллект

Это общий термин, который мы определяем так.

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ — набор всех задач, в которых компьютер может принимать решения.

Во многих случаях компьютер принимает эти решения, имитируя принятие решения человеком. Также он может имитировать эволюционные, генетические или физические процессы. Но в целом каждый раз, когда мы видим, что компьютер решает проблему самостоятельно, будь то вождение автомобиля, поиск маршрута между двумя точками, постановка диагноза пациенту или рекомендация фильма, мы наблюдаем работу искусственного интеллекта.

Что такое машинное обучение

Машинное обучение похоже на искусственный интеллект, и часто их определения путают.

Машинное обучение (МО) является частью искусственного интеллекта, определяют его следующим образом.

МАШИННОЕ ОБУЧЕНИЕ — набор всех задач, в которых компьютер может принимать решения на *основе данных*.

Что это значит? Позвольте проиллюстрировать на рис. 1.1.



Рис. 1.1. Машинное обучение — это область искусственного интеллекта

Вернемся к рассмотрению того, как именно люди принимают решения. В общих чертах мы принимаем решения двумя путями:

- с помощью логики и рассуждений;
- задействуя собственный опыт.

Представьте, что мы принимаем решение о покупке автомобиля. Можно внимательно изучить его характеристики, такие как цена, расход топлива и навигация, после чего попытаться определить наилучшую комбинацию, которая соответствует нашему бюджету. Это использование логики и принятие разумного решения. Если же вместо этого мы спросим всех своих друзей, какими автомобилями они владеют и что им в них нравится или не нравится, составим список аргументов за и против, после чего с его помощью примем решение, то это будет применением опыта — в данном случае опыта друзей.

Машинное обучение представляет собой второй метод — принятие решений с использованием своего опыта. На компьютерном языке термин, обозначающий *опыт*, звучит как *данные*. Следовательно, при машинном обучении компьютеры принимают решения на основе данных. Таким образом, каждый раз, когда мы заставляем компьютер решать задачи или принимать решения с помощью только данных, мы занимаемся машинным обучением. Простыми словами машинное обучение можно описать так — это рассудительность, за исключением того факта, что проявляет ее компьютер.

Переход от решения проблем с использованием любых необходимых средств к решению проблем с применением только данных может показаться небольшим шагом для компьютера, но это огромный шаг для человечества (рис. 1.2). Когда-то давно, если мы хотели заставить компьютер выполнять какую-то задачу, нам нужно было писать программу, а именно целый набор инструкций, которым компьютер должен следовать. Этот процесс хорош для простых задач, но некоторые из них слишком сложны для подобной концепции. Например, рассмотрим задачу по определению того, есть ли на изображении яблоко. Начав писать компьютерную программу для решения этой задачи, мы быстро поймем, что это крайне сложно.

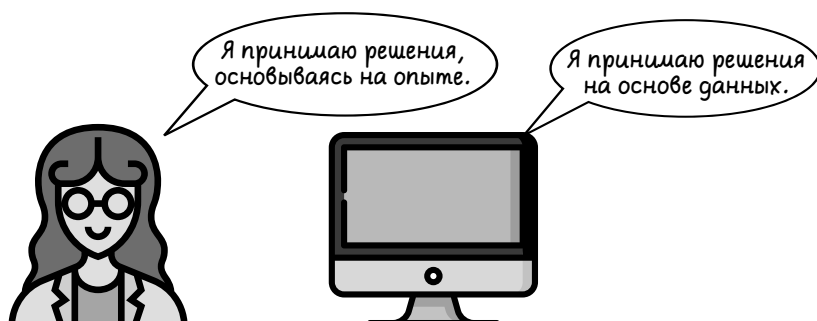


Рис. 1.2. Машинное обучение охватывает все задачи, в которых компьютеры принимают решения на основе данных

Давайте сделаем шаг назад и зададим себе вопрос: как мы, люди, узнали, как выглядит яблоко? Большинство слов мы выучили не благодаря тому, что кто-то объяснил нам, что они означают, а путем повторения. В детстве мы видели много предметов, и взрослые сообщали нам, что это за предметы. Прежде чем узнать, что такое яблоко, мы видели множество яблок на протяжении многих лет и слышали слово «яблоко», пока однажды в голове что-то не щелкнуло и мы не поняли, что такое яблоко. Это как раз то, что мы заставляем делать компьютер в рамках машинного обучения. Мы демонстрируем ему множество изображений и сообщаем, какие из них содержат яблоко (что составляет наши данные). Мы повторяем этот процесс до тех пор, пока компьютер не начнет улавливать верные формы и атрибуты, которые и составляют яблоко. В конце процесса, когда мы передаем компьютеру новое изображение, он уже может использовать эти шаблоны, чтобы определить, содержит ли изображение яблоко. То есть точно так же, как люди принимают решения на основе предыдущего опыта, компьютеры могут принимать решения на основе предыдущих данных. Конечно, нам все равно нужно запрограммировать компьютер так, чтобы он мог улавливать эти закономерности. В этом поможет ряд приемов, которым мы обучимся в книге.

Ну и раз уж мы сделали первый шаг — что такое глубокое обучение?

Точно так же, как машинное обучение является частью искусственного интеллекта, глубокое обучение — часть машинного обучения. В предыдущем разделе мы узнали, что в нашем распоряжении есть несколько методов, чтобы заставить компьютер учиться на основе данных. Один из этих методов работает настолько хорошо, что обзавелся собственной областью исследований, называемой *глубоким обучением* (ГО) (рис. 1.3), которую мы определяем следующим образом.

ГЛУБОКОЕ ОБУЧЕНИЕ — область машинного обучения, использующая объекты, называемые *нейронными сетями*.



Рис. 1.3. Глубокое обучение является частью машинного обучения

Что такое нейронные сети? О них мы узнаем в главе 10. Глубокое обучение — это, возможно, наиболее часто используемый тип машинного обучения, потому что он действительно хорошо работает. Видя какое-то из передовых приложений, таких как распознавание изображений, генерация текста, игра в го или беспилотные автомобили, скорее всего, мы в той или иной степени наблюдаем глубокое обучение.

Другими словами, глубокое обучение — это часть машинного обучения, которое, в свою очередь, является частью искусственного интеллекта. Если бы эта книга была о транспорте, то ИИ был бы совокупностью транспортных средств, МО — автомобилями, а ГО — Ferrari.

Как заставить машины принимать решения с помощью данных? Концепция вспоминания — формулирования — прогнозирования

В предыдущем разделе говорилось, что машинное обучение состоит из набора методов, которые мы используем, чтобы заставить компьютер принимать

решения на основе данных. В этом разделе узнаем, что подразумевается под принятием решений на основе данных и как работают некоторые из таких методов. Для этого еще раз проанализируем процесс, с помощью которого люди принимают решения, основанные на опыте. Это то, что называется *концепцией воспоминания — формулирования — прогнозирования* (рис. 1.4). Составляющие ее шаги сформулированы в следующем разделе. Цель машинного обучения состоит в том, чтобы научить компьютеры мыслить одинаково, следуя одним и тем же принципам.



Рис. 1.4. Воспоминание — формулирование — прогнозирование — основная концепция, используемая в книге

Как мыслят люди?

Когда нам, людям, нужно принять решение, основанное на собственном опыте, мы обычно задействуем следующую концепцию.

1. **Вспоминаем** похожие ситуации в прошлом.
2. **Формулируем** общее правило.
3. Используем его, чтобы **спрогнозировать** то, что может произойти в будущем.

Например, если вопрос звучит так: «Будет ли сегодня дождь?», то процесс предсказания состоит в следующем.

1. Мы **вспоминаем**, что на прошлой неделе большую часть времени шел дождь.
2. Мы **формулируем** правило о том, что в этом месте большую часть времени идет дождь.
3. И **прогнозируем**, что сегодня будет дождь.

Мы можем быть правы или неправы, но по крайней мере пытаемся сделать максимально точный прогноз, основываясь на имеющейся информации.

Некоторые лингвомодели и алгоритмы машинного обучения

Прежде чем обратиться к дополнительным примерам, иллюстрирующим методы, используемые в машинном обучении, дадим определения кое-каких полезных терминов, которые будем применять на протяжении всей книги. Мы знаем, что при машинном обучении заставляем компьютер учиться решать проблему с помощью данных, используя их для построения *модели*. Что такое модель? Определим ее следующим образом.

МОДЕЛЬ — набор правил, которые представляют наши данные и могут быть задействованы для прогнозирования.

Мы можем думать о модели как о представлении реальности, использующем набор правил, максимально точно имитирующих существующие данные. В примере с дождем в предыдущем разделе нашим представлением реальности была модель, описывающая мир, где большую часть времени идет дождь. Это простой мир с одним правилом: большую часть времени идет дождь. Представление может быть точным, а может — нет, но, согласно нашим данным, это наиболее точное представление реальности, которое мы способны сформулировать. Позже мы применяем это правило для составления прогнозов на основе неявных данных.

Алгоритм — это процесс, который использовался для построения модели. В текущем примере он прост: мы посмотрели, сколько дней шел дождь, и поняли, что большинство. Конечно, алгоритмы машинного обучения могут быть намного сложнее, но в итоге они всегда состоят из набора шагов. Наше определение алгоритма таково.

АЛГОРИТМ — процедура или набор шагов, используемых для решения задачи или выполнения вычисления. Здесь цель алгоритма — построение модели.

Короче говоря, модель — это то, что мы применяем для прогнозирования, а алгоритм — то, с помощью чего строим модель. Эти два определения легко спутать, и часто они взаимозаменяемы, так что для ясности рассмотрим несколько примеров.

Примеры используемых людьми моделей

В этом разделе мы сосредоточимся на распространенном применении машинного обучения — обнаружении спама. В следующих примерах будем выявлять как спам, так и письма, им не являющиеся. Письма, не являющиеся спамом, называются также *полезными*¹.

СПАМ И ПОЛЕЗНЫЕ ПИСЬМА

Спам — это общий термин для обозначения ненужной или нежелательной электронной почты, такой как письма-рассылки, рекламные акции и пр.

¹ В оригинале ham — антоним к слову spam. — *Примеч. пер.*

Термин берет начало в скетче Монти Пайтона 1972 года, в котором каждый пункт меню ресторана содержал спам в качестве ингредиента. Разработчики используют термин «ветчина» (ham) для обозначения электронных писем, не являющихся спамом.

Пример 1. Раздражающий друг

В этом примере мы встречаемся со своим другом Бобом, любителем отправлять электронные письма. Многие его письма — это спам в виде писем счастья. Они уже начинают раздражать. Сегодня суббота, и мы только что получили очередное уведомление по электронной почте от Боба. Можно ли угадать, спам ли это письмо, не глядя на него?

Попробуем использовать метод вспоминания — формулирования — прогнозирования. Во-первых, **вспомним**, скажем, последние десять электронных писем, полученных от Боба. Это наши данные. Мы помним, что шесть из них были спамом, а остальные четыре — полезными. Исходя из этой информации, можем **сформулировать** следующую модель.

Модель 1. Шесть из каждых десяти электронных писем от Боба — спам.

Это правило станет нашей моделью. Обратите внимание на то, что правило не обязательно должно быть истинным. Более того, оно может быть бессовестно неверным. Но, учитывая наши данные, это лучшее, что мы можем придумать, так что придется работать с тем, что есть. Далее в этой книге мы узнаем, как оценивать модели и улучшать их, когда это необходимо.

Теперь, когда у нас есть правило, можем использовать его, чтобы **спрогнозировать**, является ли электронное письмо спамом. Если шесть из десяти писем Боба — спам, то можно предположить, что новое электронное письмо с вероятностью 60 % будет спамом и с вероятностью 40 % — нет. Согласно этому правилу, с несколько большей вероятностью электронное письмо является спамом. Таким образом, прогнозируем, что электронное письмо — спам (рис. 1.5).



Рис. 1.5. Очень простая модель машинного обучения

Повторюсь, наше предсказание может оказаться неверным. Мы можем открыть электронное письмо и понять, что оно все-таки полезное. Но мы сделали прогноз на основе *известных данных*. Вот в чем суть машинного обучения! Возможно, вы уже думаете о том, можно ли внести в процесс какие-то улучшения. Мы оцениваем каждое электронное письмо от Боба сходным образом, но можно ли получить больше информации, которая поможет отделить спам? Попробуем немного глубже проанализировать эти электронные письма. Например, можно выяснить, когда именно Боб отправлял их, и попробовать выявить закономерность.

Пример 2. Раздражающий друг по расписанию

Давайте более внимательно рассмотрим электронные письма, которые Боб отправлял нам месяцем ранее. Точнее, посмотрим, в какой именно день он это делал. Вот электронные письма с названием дня недели и информацией о том, являются ли они спамом:

- понедельник — полезное;
- вторник — полезное;
- суббота — спам;
- воскресенье — спам;
- воскресенье — спам;
- среда — полезное;
- пятница — полезное;
- суббота — спам;
- вторник — полезное;
- четверг — полезное.

Все заиграло новыми красками, не так ли? Закономерность налицо! Похоже, что каждое электронное письмо, отправленное Бобом в течение недели, полезное, а отправленное в выходные — спам. В этом есть логика: возможно, в течение недели он отправляет нам рабочую электронную почту, а в выходные у него есть время для рассылки спама, и он решает ни в чем себе не отказывать. Итак, мы можем **сформулировать** более обоснованное правило или модель.

Модель 2. Каждое электронное письмо, которое Боб отправляет в течение недели, — это полезное письмо, а отправленное в выходные — спам.

Теперь посмотрим, какой сегодня день. Если воскресенье и мы только что получили электронное письмо от Боба, то с большой долей уверенности можем **спрогнозировать**, что это спам (рис. 1.6). Сделав прогноз, сразу отправляем его в мусорную корзину и продолжаем наш день.



Рис. 1.6. Несколько более сложная модель машинного обучения

Пример 3. Все усложняется!

Теперь предположим, что мы продолжаем следовать этому правилу и однажды встречаем Боба на улице. Он спрашивает: «Почему ты не пришел на мой день рождения?» Мы понятия не имеем, о чем он говорит. Оказывается, в прошлое воскресенье он рассылал приглашения на свой день рождения, а мы его пропустили! Почему? Потому что он отправил письмо в выходной и мы предположили, что это спам. Похоже, нам требуется модель получше. Давайте вернемся к просмотру электронных писем Боба — тому шагу, когда мы **вспоминаем**, — и выясним, нет ли там других закономерностей:

- 1 Кбайт — полезное;
- 2 Кбайт — полезное;
- 16 Кбайт — спам;
- 20 Кбайт — спам;
- 18 Кбайт — спам;
- 3 Кбайт — полезное;
- 5 Кбайт — полезное;
- 25 Кбайт — спам;
- 1 Кбайт — полезное;
- 3 Кбайт — полезное.

Что же мы видим? Похоже, что объемные электронные письма обычно являются спамом, в то время как более мелкие — как правило, нет. Это вполне логично, потому что спам-письма часто содержат большие вложения.

Итак, мы можем **сформулировать** следующее правило.

Модель 3. Любое электронное письмо размером 10 Кбайт или больше — это спам, а менее 10 Кбайт — нет.

Теперь, когда правило сформулировано, можем сделать **прогноз**. Мы смотрим на электронное письмо, которое получили сегодня от Боба, видим, что его размер составляет 19 Кбайт, и закономерно приходим к выводу — это спам (рис. 1.7).



Рис. 1.7. Еще одна чуть более сложная модель машинного обучения

Ну что, задача *решена*? Как бы не так!

Прежде всего обратите внимание на то, что для составления прогнозов мы использовали день недели и размер электронного письма. Это примеры *признаков* — одного из самых важных понятий в книге.

ПРИЗНАК — любое свойство или характеристика данных, которые модель может применять для прогнозирования.

Легко догадаться, что существует еще множество признаков, которые могут указывать на то, является электронное письмо спамом или нет. Есть идеи? В следующих разделах мы выявим еще несколько.

Пример 4. Еще?

Наши два классификатора вполне рабочие, потому что исключают большие электронные письма и электронные письма, отправленные в выходные дни. Каждый из них использует только один из этих двух признаков. Но что, если нам нужно правило, которое работало бы с обоими признаками? Вполне допустимы правила, подобные следующим.

Модель 4. Если размер электронного письма более 10 Кбайт или оно было отправлено в выходные дни, то оно классифицируется как спам. В противном случае — как полезное.

Модель 5. Если электронное письмо отправлено в течение недели, то оно должно быть больше 15 Кбайт, чтобы попасть в категорию спама. Если оно отправлено в выходные дни, то должно быть больше 5 Кбайт, чтобы попасть в категорию спама. В противном случае оно классифицируется как полезное.

Можно усложнять и дальше.

Модель 6. Присвоим дням недели порядковые номера: понедельник — 0, вторник — 1, среда — 2, четверг — 3, пятница — 4, суббота — 5, воскресенье — 6. Если мы сложим номер дня и размер электронного письма (в килобайтах) и в результате получим 12 или более, то электронное письмо классифицируется как спам (рис. 1.8). В противном случае — как полезное.



Рис. 1.8. Еще более сложная модель машинного обучения

Все это рабочие модели. Мы можем продолжать создавать все больше и больше моделей, добавляя уровни сложности или рассматривая все новые признаки. Теперь вопрос в том, какая же модель будет лучшей. Вот тут-то нам и понадобится помощь компьютера.

Примеры моделей, используемых машинами

Наша задача заключается в том, чтобы заставить компьютер думать так, как думаем мы, а именно применять структуру воспоминания — формулирования — прогнозирования. Если упростить, то вот что должен сделать компьютер на каждом из этапов.

- **Вспомнить** — обратиться к огромной таблице данных.
- **Сформулировать** — создать модели, просматривая множество правил и формул, и проверить, какая модель лучше всего соответствует данным.
- **Спрогнозировать** — использовать эту модель для прогнозирования будущих данных.

Этот процесс немногим отличается от того, что мы проделывали в предыдущем разделе. Улучшением здесь является то, что компьютер может быстро создавать модели, просматривая множество формул и комбинаций правил до тех пор, пока не обнаружит то, что хорошо соответствует имеющимся данным. Например, мы можем создать классификатор спама с такими признаками, как

отправитель, дата и время суток, количество слов, количество орфографических ошибок и появление определенных слов, таких как «*купить*» или «*выиграть*». Модель вполне может выглядеть как следующее логическое утверждение.

Модель 7

- Если в электронном письме есть две орфографические ошибки или более, то оно классифицируется как спам.
- Если оно содержит вложение размером более 10 Кбайт, оно классифицируется как спам.
- Если отправителя нет в нашем списке контактов, письмо классифицируется как спам.
- Если в нем есть слова «*купи*» и «*выиграй*», оно классифицируется как спам.
- В противном случае оно классифицируется как полезное.

Все это можно записать следующей формулой.

Модель 8. Если (размер) + 10 (количество орфографических ошибок) – (количество появлений слова «мама») + 4 (количество появлений слова «купить») > 10, то мы классифицируем сообщение как спам (рис. 1.9). В противном случае классифицируем его как «ветчину».

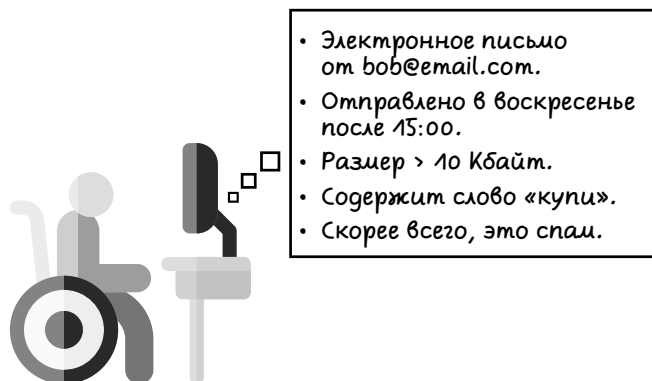


Рис. 1.9. Гораздо более сложная модель машинного обучения, выработанная компьютером

Теперь вопрос: какое правило лучшее? Быстрый ответ: то, которое лучше всего соответствует данным. А настоящий ответ: то, которое лучше всего обобщает новые данные. В конце концов, мы можем прийти к сложному правилу, но компьютер способен сформулировать его и использовать для быстрого принятия решений. Следующий вопрос: как создать наилучшую модель? Именно об этом и пойдет речь в книге.