Оглавление

Формулировка задачи машинного обучения

Допустим, вы управляете несколькими сотнями кинотеатров по всей стране и хотите спрогнозировать, сколько билетов будет продано за определенное время показа в конкретном кинотеатре, — это может очень пригодиться при планировании графика показа фильмов. При наличии статистических данных о показе фильмов задачу машинного обучения можно сформулировать следующим образом: на основе информации о прокате, имеющейся в наборе данных, узнать, сколько билетов было продано на каждый сеанс в каждом кинотеатре, затем применить полученную модель к фильму-кандидату и определите, каким будет спрос на этот фильм в конкретное время.

Атрибуты фильма, которые мы используем в качестве входных данных в модели машинного обучения, называются *признаками* модели. Метка — это прогнозное значение, которое требуется узнать, и в данном случае метка — это количество проданных билетов. Ниже приводятся некоторые примеры признаков, которые вы можете включить в свою модель:

- Оценка содержимого кинофильма[1] (например, PG-13 означает, что фильм рекомендуется для детей младше 13 лет).
- Фильм будет показан в рабочие или выходные дни?
- В какое время дня будет демонстрироваться фильм (днем, вечером, ночью)?
- Жанр фильма (комедия, триллер и т. д.).
- Как давно снят фильм (в днях).
- Средний рейтинг фильма по оценкам критиков (по шкале от 1 до 10).
- Общая сумма кассовых сборов за предыдущий фильм этого же режиссера, если применимо.
- Общая сумма кассовых сборов за предыдущий фильм с этим же актером в главной роли, если применимо.
- Местоположение кинотеатра.
- Тип кинотеатра (например, большой кинотеатр с несколькими залами, кинотеатр под открытым небом для автомобилистов, кинозал в торговом центре и т. д.).

Обратите внимание, что название фильма само по себе не является хорошим обучающим признаком.[2] Даже если фильм «Шпион, выйди вон!» («Tinker Tailor

[1] См. https://en.wikipedia.org/wiki/Motion_picture_content_rating_system.

[2] Отдельные слова в названии фильма могут оказаться неплохими признаками, если применить к названиям фильмов методы обработки естественного языка, такие как лексемизация, морфологический поиск и получение векторных представлений слов. Также могут пригодиться признаки, вычисленные из названий фильмов; например,

Soldier Spy») 2011 года будет присутствовать в наборе обучающих данных, нам будет неинтересно прогнозировать прокат именно этого фильма (потому что он уже был показан в наших кинотеатрах). Гораздо интереснее спрогнозировать сборы, скажем, от проката фильма «Глубоководный горизонт» («Deep Water Horizon»), еще одного триллера с похожими отзывами критиков, выпущенного в 2016 году.

То есть модель машинного обучения должна опираться на особенности фильма (характеризующие фильм), а не на признаки, однозначно его идентифицирующие. В этом случае модель сможет догадаться, что при одинаковых условиях показа фильм «Глубоководный горизонт» принесет такую же прибыль, как и «Шпион, выйди вон!», потому что они относятся к одному жанру и имеют похожий рейтинг.

Первые четыре признака (рейтинг, дни показа, время показа, жанр) являются категориальными, то есть принимают одно из конечного числа возможных значений. В BigQuery категориальным считается любой признак, являющийся строкой. Если в базе данных категориальные признаки представлены значениями другого типа (например, признак времени показа может быть числом, например 1430, или отметкой времени), в запросе они должны преобразовываться в строки. Следующие четыре признака (время с момента выпуска, рейтинги критиков, кассовые сборы за предыдущий фильм этого же режиссера и за предыдущий фильм с этим же актером в главной роли) — числовые, то есть имеют значимое числовое выражение. Последние два признака (тип и местоположение кинотеатра) должны быть представлены особым образом; возможные варианты мы обсудим далее в этой главе.

Меткой, или правильным прогнозом, является количество проданных билетов. Во время обучения модели машинного обучения BigQuery передаются входные признаки и соответствующие метки, на основе которых создается модель, обобщающая эту информацию (рис. 9.1). Затем, во время прогнозирования, обученную модель можно применить к новому набору входных признаков и получить оценку количества билетов, которое можно продать, если запланировать показ фильма в определенное время и в определенном месте.

Типы задач машинного обучения

Мы часто используем разные модели и методы машинного обучения в зависимости от характера входных признаков и меток. В этом подразделе мы кратко перечислим поддерживаемые типы задач машинного обучения, а в оставшейся части главы подробно рассмотрим способы их решения.

хорошим прогнозирующим элементом может послужить длина названия или наличие в нем слова «шпион».

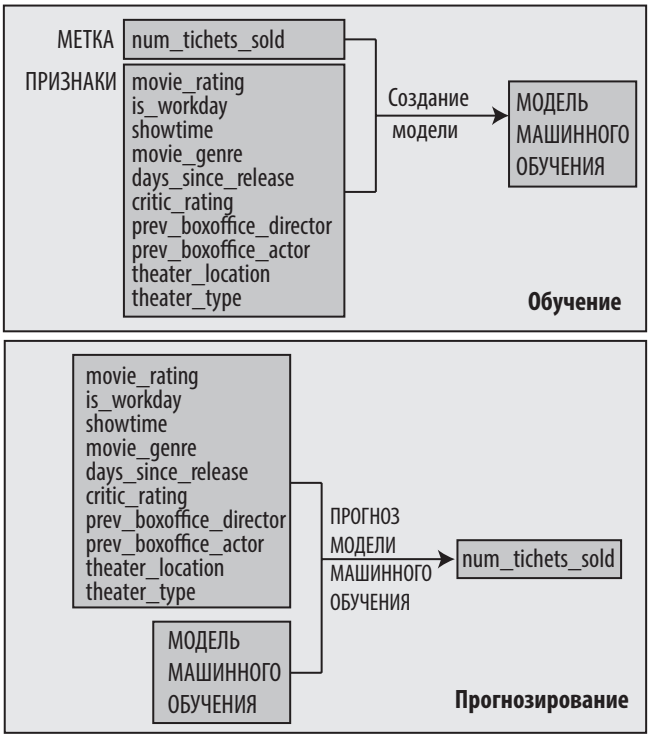

Рис. 9.1. Во время обучения модели машинного обучения передаются входные признаки и соответствующие метки. Затем обученную модель можно использовать для прогнозирования. На основе набора входных признаков модель способна спрогнозировать метку

Регрессия

В примере, описанном в предыдущем разделе, требовалось спрогнозировать количество проданных билетов на конкретный фильм. В данном случае метка является числом, и задачи машинного обучения подобного типа называют *регрессией*.

Классификация

Если результатом (меткой) задачи машинного обучения является категориальная переменная, такие задачи относятся к разряду задач *классификации*. Модели классификации возвращают вероятность принадлежности записи к категории (классу), определяемой меткой. Например, чтобы получить модель машинного обучения, прогнозирующую успех шоу, вы могли бы создать модель классификации, возвращающую вероятность успеха.

Многие задачи классификации имеют два класса, например: успех или провал шоу, высокий или низкий спрос на товар, прибытие рейса вовремя или с опозданием. Это задачи так называемой *бинарной классификации*. В таких случаях столбец метки должен иметь значение True или False либо 1 или 0. Прогнозом такой модели является вероятность, что метка будет иметь значение True. Чтобы определить наиболее вероятный класс, мы обычно устанавливаем пороговую вероятность равной 0.5.

Задача классификации может иметь несколько классов. Например, возвращаясь к нашему сценарию проката велосипедов, может понадобиться предсказать пункт проката, куда будет возвращен велосипед, и поскольку для этой категориальной метки существуют сотни возможных значений, эта задача относится к разряду задач *мультиклассовой (множественной) классификации*. Результатом такой модели машинного обучения является набор вероятностей, по одной для каждого пункта проката, и сумма этих вероятностей будет равна 1.0. В задачах множественной классификации, как правило, наибольший интерес представляют три или пять наибольших предсказаний, а не фактические значения вероятностей.

Рекомендательная система

Особый случай многоклассовой классификации, когда цель состоит в том, чтобы рекомендовать «следующий» продукт на основании рейтингов или предыдущих покупок, называется *рекомендательной системой*. Как и любую другую задачу множественной классификации, задачу подбора рекомендаций можно решить с помощью стандартного подхода. Тем не менее для таких задач были созданы специальные типы моделей машинного обучения, и для реализации рекомендаций лучше использовать именно их. Рекомендательные системы также лучше подходят для решения задач таргетинга — поиска клиентов, которым понравится продукт или рекламное предложение.

Кластеризация

Если в наборе исходных данных вообще нет меток, к ним не получится применить методы машинного обучения с учителем. Зато можно попробовать выявить сложившиеся группы в данных; задачи этого вида называются *кластеризацией (группировкой)*. Например, клиентов можно разделить на кластеры (группы) в зависимости от характеризующих их признаков. Также можно воспользоваться службой Cloud Data Labeling Service и прибегнуть к помощи людей, осуществляющих маркировку данных, перед проведением обучения с учителем.

Неструктурированные данные

До сих пор мы исходили из того, что данные могут быть структурированными или полуструктурированными. Если какие-то входные признаки не структу-

рированы (например, являются изображениями или текстом на естественном языке), их можно обработать с помощью готовых моделей, таких как Cloud Vision API или Cloud Natural Language, и использовать полученные результаты в виде числовых или категориальных входных признаков. Например, с помощью Cloud Natural Language API можно определить ключевые сущности в электронных письмах клиентов или эмоциональную окраску их отзывов и использовать полученные сущности в качестве категориальных переменных, а оценки эмоциональной окраски — в качестве числовых признаков.

Преобразовать неструктурированные данные в структурированные можно с помощью строковых функций или машинного обучения. На практике часто используется прием под названием *«мешок слов»,* суть которого состоит в том, чтобы разделить текстовое поле на отдельные слова и интерпретировать наличие/отсутствие отдельных слов как признаки. Например, из названия фильма «Шпион, который меня любил» («The Spy Who Loved Me») можно выделить два признака со значением True: `has_spy` и `has_love`. Все остальные признаки будут иметь значение False (слова «the», «Who» и «Me» лучше отбросить, потому что они слишком часто встречаются в тексте и не имеют большого значения для прогноза). Также в качестве признака можно использовать количество слов в названии (фильмы с большим количеством слов в названии, скорее всего, окажутся авторским кино и могут быть привлекательными для разных аудиторий).

Если не структурирована сама метка (например, требуется, чтобы модель генерировала идеальные ответы на вопросы клиентов, опираясь на набор архивных ответов), такие задачи относят к разряду задач генерирования естественного языка — они не поддерживаются в BigQuery.

Краткая сводка по типам моделей

В табл. 9.1 перечисляются типы задач машинного обучения. В следующем разделе мы обсудим типы моделей, поддерживаемые в BigQuery.

Таблица 9.1. Типы моделей машинного обучения и как они реализуются в BigQuery

Описание задачи	Тип задачи машинного обучения	Тип модели в BigQuery
Метки отсутствуют, данные нельзя обеспечить метками	Кластеризация	`kmeans`

Числовые метки	Регрессия	`linear_reg`
		`dnn_regressor`
		`boosted_tree_regressor`

Рекомендация продуктов пользователям	Рекомендации	`matrix_factorization`

Таблица 9.1 (*окончание*)

Описание задачи	Тип задачи машинного обучения	Тип модели в BigQuery
Выбор целевой аудитории для предложения продукта	Таргетинг клиентов	`matrix_factorization`

Метки имеют значения 1/0, True/False (две категории)	Бинарная классификация	`logistic_reg`
		`dnn_classifier`
		`boosted_tree_classifier`

Метки представлены фиксированным набором строк	Многоклассовая классификация	`logistic_reg`
		`dnn_classifier`
		`boosted_tree_classifier`

Входные признаки не структурированы	Классификация изображений, классификация текста, анализ эмоциональной окраски, извлечение сущностей	Используйте выходные данные Cloud Vision API или Cloud Natural Language API в качестве входных данных для любой из стандартных моделей BigQuery, перечисленных выше
Метки не структурированы	Ответы на вопросы, аннотирование текста, генерирование подписей к изображениям	Используйте продукты Cloud AutoML

Построение регрессионной модели

В примере построения регрессионной модели мы используем набор данных `london_bicycles`. Предположим, что у нас есть два типа велосипедов: тяжелые и надежные городские велосипеды и быстрые, но менее надежные шоссейные велосипеды. Для клиентов, совершающих длительные поездки, у нас должны быть в наличии шоссейные велосипеды, а для коротких поездок — обычные городские. Чтобы организовать правильное распределение велосипедов, необходимо спрогнозировать продолжительность поездок.

Выбор метки

Первым шагом к решению задачи машинного обучения является ее формулирование — определение признаков и меток для модели. Поскольку задача первой модели состоит в том, чтобы на основе набора архивных данных предсказать продолжительность поездки, меткой будет служить продолжительность поездки.

Но насколько верно мы определили цель задачи? Должна ли модель прогнозировать продолжительность каждой поездки или общую продолжительность

всех поездок для каждого пункта проката, например, в течение часа? Если мы выберем последний вариант, тогда меткой должна быть сумма продолжительностей всех поездок за определенный час. Допустим, по опыту ведения бизнеса мы знаем, что пункт проката, выдающий 1000 велосипедов на 20 минут каждый, должен выдавать городские велосипеды, тогда как пункт проката, выдающий ежедневно 100 велосипедов на 200 минут, должен выдавать шоссейные велосипеды. То есть прогнозирование общей продолжительности не поможет принять правильное решение, а вот предсказание продолжительности каждой отдельной поездки будет весьма кстати.

Другой вариант — оценка вероятности, что поездка продлится меньше 30 минут. В этом случае метка будет иметь два значения — True/False — в зависимости от длительности поездки (больше или меньше 30 минут). Это еще больше поможет бизнесу, потому что вероятность может определять относительную пропорцию городских и шоссейных велосипедов, имеющихся в каждом пункте проката.

На практике довольно часто приходится выбирать между несколькими целями. Иногда можно создать метку в виде взвешенной комбинации из целей и обучить единственную модель. Иногда полезнее обучить несколько моделей, по одной для каждой цели, и использовать разные модели в разных сценариях. А иногда лучше представить конечному пользователю результаты всех моделей и дать ему возможность выбора. Все зависит от того, чем вы занимаетесь.

В этом примере мы построим две модели: одну для прогнозирования продолжительности поездки, а другую для прогнозирования вероятности того, что поездка займет больше 30 минут. Затем мы дадим конечному пользователю возможность принять решение на основе двух прогнозов.

Выбор признаков в наборе данных

Если предположить, что продолжительность поездок зависит от пункта проката, дня недели и времени суток, эти параметры могут послужить нам входными признаками. Прежде чем продолжить и создать модель с этими тремя признаками, желательно убедиться, что эти факторы действительно влияют на метку.

Выбор и формирование признаков для модели машинного обучения называется *конструированием признаков* (feature engineering). Конструирование признаков часто является наиболее важным условием создания точных моделей машинного обучения, и этот шаг может гораздо сильнее повлиять на точность прогнозирования, чем выбор алгоритма или настройка гиперпараметров. Чтобы получить хороший набор признаков, необходимо глубоко понимать данные и предметную область. Часто на этом этапе проверяется множество гипотез; у вас есть идея относительно признака, вы проверяете ее обоснованность (влияние признака на метку), а затем добавляете этот признак в модель. Если эта идея не подтверждается, вы проверяете следующую.