1 Табличные данные

В этой главе

- ✓ Что такое табличные данные
- ✓ Почему они так важны
- ✓ Различие между глубоким обучением и другими методами анализа табличных данных
- ✓ Что думают специалисты об использовании глубокого обучения для табличных данных
- ✓ Что отличает табличные данные от других типов данных, таких как изображения, звук или текст

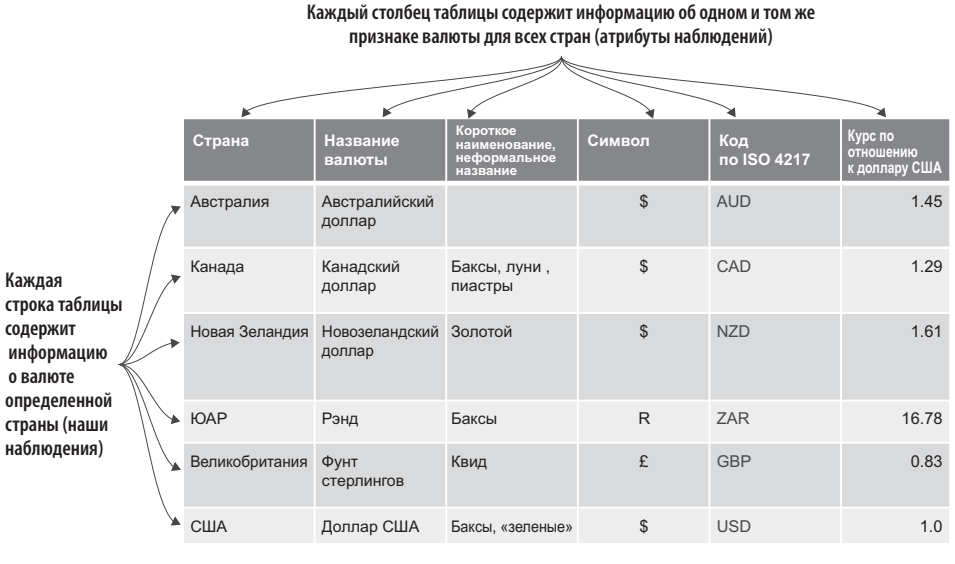
Табличные данные — основа современной жизни, а для многих из нас — часть нашей повседневной работы. Они присутствуют в CSV-файлах и реляционных базах данных, служат основой для аналитических отчетов и используются для обучения ML-моделей. А ML-модели, обученные на бизнес-данных, в свою очередь, могут успешно решать множество полезных задач, например прогнозирование потребностей в запасах для розничных точек или предсказание рыночных цен на товары.

В этой главе мы расскажем, как выбрать подходящий метод моделирования для задач с табличными данными. Мы представим два главных подхода: глубокое и машинное обучение. Затем рассмотрим уникальные особенности, характерные именно для применения табличных данных в ML-моделях.

1.1. Что такое табличные данные?

Для целей этой книги *табличные данные* — это просто данные, организованные в строки и столбцы. Коллекция табличных данных может быть названа *табличным набором данных*, *табличным датасетом* или просто *таблицей*. Все элементы в строке связаны с одной точкой данных, также называемой наблюдением. Каждая строка является автономной, не зависит от других строк и полностью описывает определенное состояние или условие. Столбцы представляют собой атрибуты для этой точки данных, и их часто называют переменными (это термин из статистики) или признаками (что более типично для машинного обучения). Все записи в столбце имеют один и тот же тип данных, например целое число (integer), строка (string) или число с плавающей точкой (float).

Рассмотрим для примера таблицу, в которой содержится информация о валютах, используемых в ряде стран (рис. 1.1).



Страна	Название валюты	Короткое наименование, неформальное название	Символ	Код по ISO 4217	Курс по отношению к доллару США
Австралия	Австралийский доллар		$	AUD	1.45
Канада	Канадский доллар	Баксы, луни , пиастры	$	CAD	1.29
Новая Зеландия	Новозеландский доллар	Золотой	$	NZD	1.61
ЮАР	Рэнд	Баксы	R	ZAR	16.78
Великобритания	Фунт стерлингов	Квид	£	GBP	0.83
США	Доллар США	Баксы, «зеленые»	$	USD	1.0

Рис. 1.1. Пример табличных данных

Столбцы в этой таблице содержат значения разных типов.

- Страна, название и символ валюты, код ISO 4217 являются *категориальными столбцами,* поскольку допустимые значения для них берутся из конечного, относительно небольшого набора значений.
- Неформальные названия валют представляют собой столбцы с текстовыми данными в свободной форме, поскольку они могут содержать ряд значений или не содержать никаких в зависимости от страны.
- Курс по отношению к доллару США — это столбец с непрерывными данными, поскольку он содержит действительные числовые значения.

Более подробно характеристики табличных данных мы рассмотрим в главе 2.

Табличные данные могут храниться в различных физических форматах.

- *Отдельные файлы*, включая файлы CSV и электронные таблицы, такие как Excel и Google Таблицы.
- *Таблицы в реляционных базах данных*:
  - базы данных с открытым исходным кодом, например Postgres (https://www.postgresql.org/) и MySQL (https://www.mysql.com/);
  - локальные проприетарные (on-premise vendor) базы данных, например SQL Server (https://mng.bz/MD2W) и Oracle (https://www.oracle.com/ca-en/database/);
  - облачные нативные (cloud-native) базы данных, например Google Cloud Spanner (https://cloud.google.com/spanner), AWS Aurora (https://aws.amazon.com/rds/aurora/) и Snowflake (https://www.snowflake.com/).

ПРИМЕЧАНИЕ Вы могли слышать термин *структурированные данные*, используемый наряду с *табличными данными*. Однако эти два понятия — не абсолютные синонимы. Например, иногда речь может идти о данных, которые имеют определенную степень структурированности, но при этом не являются таблицами, допустим вложенные записи в формате JSON. Структурированные данные могут включать реляционные, графические, пространственные данные и временны́е ряды, которые также могут быть представлены в табличной форме. Чтобы избежать путаницы, в этой книге мы будем использовать исключительно термин *табличные данные*.

Теперь, когда мы определились, что представляют собой табличные данные, попробуем разобраться, что включают в себя нетабличные. Это важная тема, поскольку различия между ними помогают прояснить один из ключевых вопросов, рассматриваемых в этой книге: существуют ли ситуации, в которых при анализе таблиц следует использовать глубокое обучение? Ниже приведены некоторые примеры данных, которые не являются табличными:

- изображения;
- видео;
- аудио;
- текст;
- данные датчиков в формате JSON, например генерируемые устройствами интернета вещей (IoT);
- стриминговые данные в социальных сетях.

Можете ли вы назвать одно общее свойство, которое объединяет их? Если вы ответите: «Все они очень успешно использовались для обучения моделей глубокого обучения», то будете абсолютно правы. Действительно, за последние 10 лет одна за другой были созданы принципиально новые модели с использованием различных нетабличных датасетов. В этой книге мы выясним, почему глубокое обучение не произвело такой же революции для табличных данных и когда имеет смысл применять его к ним.

1.2. Весь мир работает на табличных данных

Согласно статье «Structured vs Unstructured Data» (https://mng.bz/5g7Z), до 90 % всех цифровых данных в мире являются нетабличными, и их доля увеличивается с каждым годом. Если это так, то зачем читать книгу о применении методов машинного обучения к таблицам? Хотя только небольшая часть данных в мире является табличной, но без них невозможно представить себе современную жизнь. Каждый банк, страховая компания, государственное учреждение, ретейлер и производитель — все они ведут свою основную деятельность с помощью таблиц. Это продиктовано, во-первых, тем, что такая структура данных, организованных в строки и столбцы, делает их удобными для ввода, извлечения, управления и анализа. Во-вторых, такой формат поддерживается многими бизнес-программами и приложениями, такими как электронные таблицы, базы данных и инструменты бизнес-аналитики.

Помимо своей основной деятельности, организации зависят от табличных данных в части мониторинга своего прогресса и обнаружения проблем. Поскольку в современном мире каждый из нас является потребителем, сотрудником и гражданином, наши ежедневные действия обновляют сотни и даже тысячи таблиц.

В течение трех лет один из авторов этой книги имел честь руководить техподдержкой одной из крупнейших реляционных баз данных в мире. Работа 24/7 продемонстрировала масштабы организаций, которые работают с табличными данными, а также последствия сбоев в таких системах: покупатели на целом континенте не могли использовать свои кредитные карты, фуры стояли

в многокилометровых пробках на границе, грузовые поезда останавливались, сайты интернет-магазинов обваливались в «черную пятницу», а производство кардиостимуляторов прекращалось. Это не преувеличение: весь мир работает на табличных и в целом структурированных данных.

Табличные данные повсюду, и они критически важны. Работа многих из нас зависит от них, так что умение эффективно применять к ним машинное (и, если это уместно, глубокое) обучение является очень полезным навыком. В этой книге вы узнаете о методах, позволяющих раскрыть весь огромный потенциал таких данных.

1.3. Сравнение машинного и глубокого обучения

И глубокое, и классическое машинное обучение решают задачу сопоставления входных данных и предсказания (прогноза). Однако они используют разные подходы: методы глубокого обучения разработаны для имитации поведения биологического мозга, тогда как машинное обучение обычно основано на статистических оптимизациях или установлении сходства. Однако, кроме того, эти два подхода различаются способом работы с данными.

В классическом машинном обучении преобразование и генерация признаков (feature engineering) играют важнейшую роль, поскольку, независимо от выбранной модели, необходимо выполнить соответствующие преобразования входных данных на основе их характеристик и предметной области, к которой они относятся. (Это бизнес-данные? Отражают ли они какие-либо социальные, экономические или физические явления?) Вот некоторые причины, почему генерация признаков так важна в классическом машинном обучении.

- *Извлечение релевантной информации.* Далеко не всегда первичные данные хорошо отражают специфику решаемой задачи. Генерация признаков помогает определить и извлечь наиболее информативные параметры данных, одновременно исключая незначащие или зашумленные. Сориентировав модель на наиболее значимые признаки, можно обучить ее находить наиболее важные закономерности, что улучшает ее обобщающую способность и производительность.
- *Представление данных.* Различные модели имеют разные требования к представлению данных. В процессе генерации признаков необходимо преобразовать данные в нужный формат, учитывающий требования и ограничения конкретной модели. Этот шаг гарантирует, что модель сможет эффективно обучаться на данных и делать точные прогнозы.
- *Решение проблемы нелинейности.* Во многих реальных задачах взаимосвязи между признаками и целевой переменной могут быть нелинейными.

Генерация признаков помогает преобразовать данные для работы с нелинейностями, упрощая линейным моделям аппроксимацию сложных зависимостей.

- *Учет предметной области*. В некоторых случаях эксперты в предметной области обладают ценной информацией, которую можно использовать при генерации признаков. Применение специфических знаний может значительно улучшить производительность модели при решении конкретных задач.

С другой стороны, подходы глубокого обучения полагаются на *обучение представлений* (representation learning), которое заключается в их способности автоматически обрабатывать данные внутри модели, приводя их в осмысленную форму для решения поставленной задачи. Возможности обучения представлений позволяют моделям глубокого обучения преобразовывать данные в более компактный и осмысленный формат, который фиксирует соответствующие признаки и выявляет закономерности для конкретной задачи. Фактически в процессе обучения, благодаря тому что все входные признаки нелинейно взаимодействуют с другими, модели глубокого обучения сами обнаруживают сложные закономерности и зависимости в данных, которые могут быть неочевидны при ручной генерации признаков, и выстраивают иерархические представления входных данных, начиная с базовых признаков и постепенно переходя к более сложным и абстрактным.

Таким образом, при анализе табличных данных с помощью классического машинного обучения внимание в первую очередь сосредоточено на тщательной и эффективной генерации признаков, тогда как в моделях глубокого обучения отводят гораздо большее значение архитектуре слоев нейронов и характеристикам отдельных нейронов. Это и есть основная мысль нашей книги. В следующих главах будет не только проиллюстрировано различие между классическим машинным и глубоким обучением, но и представлены варианты постановки задачи в контексте обработки данных и поиска решений на основе этого фундаментального различия между двумя подходами.

Мы понимаем, что упрощение неизбежно привносит неточность. Однако на протяжении всей книги мы будем использовать общий термин *«машинное обучение»* или *«классическое машинное обучение»* для всех подходов, за исключением нейронных сетей, а термин *«глубокое обучение»* будет применяться для подходов, основанных на нейронных сетях.

Мы рассмотрим как базовые, так и более продвинутые модели машинного обучения, имеющиеся в популярных библиотеках.

- Базовые модели машинного обучения, доступные в библиотеке scikit-learn (https://scikit-learn.org/) и специализированных библиотеках для графических

процессоров, таких как NVIDIA Rapids (https://developer.nvidia.com/rapids), для следующих алгоритмов:

  - линейная регрессия;
  - логистическая регрессия;
  - обобщенные линейные модели.

- Алгоритмы на основе деревьев из scikit-learn:
  - бэггинг на основе слабых предикторов;
  - случайный лес;
  - особо случайные деревья (extremely randomized trees).
- Градиентный бустинг на основе гистограмм:
  - XGBoost eXtreme Gradient Boosting (https://github.com/dmlc/xgboost);
  - LightGBM от Microsoft (https://github.com/Microsoft/LightGBM);
  - HistGradientBoosting из scikit-learn.

В качестве моделей глубокого обучения мы рассмотрим ряд архитектур, реализованных во фреймворках глубокого обучения TensorFlow и PyTorch, которые показали свою эффективность при работе с табличными данными:

- неглубокие сети с категориальными эмбеддингами (непосредственно реализованные с использованием одного из доступных фреймворков глубокого обучения);
- fastai tabular (https://docs.fast.ai/tabular.model.html);
- PyTorch Tabular (https://github.com/manujosephv/pytorch_tabular);
- TabNet (https://arxiv.org/abs/1908.07442);
- SAINT (https://arxiv.org/abs/2106.01342);
- DeepTables (https://github.com/DataCanvasIO/deeptables).

Вы сами сможете наблюдать различие между машинным и глубоким обучением на протяжении всей книги, поскольку мы рассмотрим оба подхода и дадим рекомендации по использованию каждого из них для решения задач анализа табличных данных.

1.4. В чем особенность табличных данных?

Мы знаем, что подходы глубокого обучения преобладают в решении проблем, связанных со многими типами данных, которые мы могли бы назвать нетабличными или неструктурированными вследствие большого разнообразия их

характеристик, размеров и модальностей и которые нельзя заключить в узкие рамки формата «строки/столбцы». Типичные примеры неструктурированных данных, с которыми успешно справляются модели глубокого обучения, это:

- аудио;
- видео;
- изображения;
- текст.

Здесь, в отличие от задач анализа структурированных табличных данных, нет ничего похожего на типичный табличный формат, а есть разные файлы или экземпляры, содержащие множество информации в неупорядоченном виде. До того как глубокое обучение произвело революцию в способе моделирования неструктурированных данных, для их применения в предиктивной модели их нужно было преобразовывать в структурированный формат путем тщательного создания четко и однозначно определенных признаков. Эта процедура называется генерацией признаков (feature engineering). Для каждого типа задач с неструктурированными данными исследователям и практикам требовались годы, чтобы найти оптимальные признаки для извлечения данных, чтобы затем передать их в модель машинного обучения и получить предсказание приемлемого качества.

Благодаря умению работать с представлениями, модели глубокого обучения могут сами производить все необходимые преобразования, чтобы превратить неструктурированные данные в адекватный прогноз в сквозном (end-to-end) режиме, то есть напрямую от входа к решению. Учитывая это, можно было бы ожидать, что модели глубокого обучения будут еще эффективнее на табличных данных, но пока этого не произошло.

По правде говоря, существуют различные объяснения трудностей, с которыми сталкивается глубокое обучение при работе с табличными данными. Первое связано с фактическими направлениями академических исследований и частными инвестициями в новые технологии и методологии. Как мы уже упоминали, в прошлом исследователи посвящали все свое время и тратили усилия на поиск наилучшего способа преобразования неструктурированных данных в структурированные, чтобы соответствовать парадигме машинного обучения своего времени. В настоящее время те же усилия прилагаются к продвижению глубокого обучения, по большей части на неструктурированных данных, поскольку они более доступны в публичных репозиториях и более однородны (uniform), чем таблицы, что позволяет надеяться на больший успех проводимых исследований.

Хранилища изображений, такие как ImageNet (https://image-net.org/index.php), и открытые текстовые корпуса, такие как Wikipedia или веб-архив Common Crawl (https://commoncrawl.org/), легкодоступны как академическим исследователям, так и практикам для обучения или доработки моделей глубокого обучения. Что касается таблиц, то для них нет аналогичного общего репозитория с открытым

исходным кодом. Напротив, они рассредоточены по множеству частных баз данных, каждая из которых демонстрирует еще более высокую степень изменчивости, чем неструктурированные данные, поскольку имеет собственные правила сбора данных и структуру признаков.

Мало того что табличные датасеты с открытым исходным кодом, представляющие реальные бизнес-задачи, как правило, сложнее найти; они также обычно меньше по размеру и часто сильно отличаются от данных, которые находятся в собственности предприятий и государственных учреждений. Как следствие, отсутствие достаточно большого объема открытых данных приводит к тому, что нейронные сети становятся неэффективны. Кроме того, не существует общепринятых метрик для оценки достигнутого успеха, поскольку использование определенного типа данных ограничено одной конкретной задачей из обширной области проблем табличных данных. Для любого исследователя гораздо сложнее применить лучшие практики к табличному датасету или некоторому ограниченному их подмножеству, чем сделать то же самое, используя изображения, аудио или тексты, которые общедоступны и для которых есть общепризнанные метрики качества и производительности.

Поскольку табличные датасеты труднодоступны и чрезвычайно разнообразны по типу содержащейся в них информации, они представляют собой еще одно ограничение для решений на основе глубокого обучения: нельзя придумать никакие универсальные, предварительно обученные модели, поскольку нельзя получить доступ ко всем видам табличных задач. Как только вы разработаете модель глубокого обучения для задач с изображениями и текстом, вы можете сделать ее общедоступной и подождать, когда другие ученые или инженеры смогут применить ее для своих задач после небольшой дополнительной настройки или дообучения. Технический термин для этого процесса звучит как *трансферное обучение (transfer learning)*: нейросеть, обученная на одной задаче, «переводится» на другую похожую задачу с минимальными усилиями. Распространение методов глубокого обучения во многом произошло именно благодаря возможности использовать предобученные модели для дообучения на похожих задачах.

В заключение следует отметить, что отсутствие обобщаемых табличных примеров, большое разнообразие существующих таблиц и повышенное внимание ученых к неструктурированным данным привели к тому, что генерация признаков в машинном и глубоком обучении стала играть различную роль.

- В машинном обучении удачная генерация признаков может обеспечить гораздо более мощную предсказательную способность табличным данным, чем сами алгоритмы, и обычно рассматривается скорее как искусство, нежели как наука.

- В глубоком обучении, напротив, ученые и практики склонны слишком полагаться на обучение представлений и позволять сети справляться со всем самостоятельно, вместо того чтобы самим использовать генерацию признаков и продемонстрировать, как глубокое обучение — при тех же входных данных, что и у алгоритмов машинного обучения, — может находить решение альтернативным и эффективным способом.

Действительно, как показали недавние исследования, типичные свойства табличных данных, такие как избыточные признаки, смещенные распределения и нерегулярные закономерности целевой переменной, представляют собой проблему для нейронных сетей. Мы обсудим это более подробно в главе 5, когда будем рассматривать модели градиентного бустинга. Тем не менее мы утверждаем, что как машинное, так и глубокое обучение являются жизнеспособными методами решения задач табличных данных, и нейросети приобретут большее влияние в будущем, поскольку в настоящее время учеными прилагается много усилий для тестирования архитектур и решений на более реалистичных табличных данных.

1.5. Генеративный ИИ и табличные данные

Генеративный ИИ, в частности большие языковые модели (large language models, LLM), стал незаменимым помощником в различных задачах, связанных с созданием и обработкой текста. В целом LLM оказались прорывным решением для определенного круга задач, среди которых следует назвать следующие.

- *Генерация* — генерация текста, например очередного токена, для получения законченной фразы или целого текста на основе инструкции, или промпта.
- *Извлечение* — распознавание именованных сущностей (named entities), сегментация предложений, извлечение ключевых слов, тематическое моделирование, определение семантических связей.
- *Классификация* — задачи классификации текстов по языкам, на которых они написаны, а также классификации намерений, тональности, семантики и даже такие нетривиальные задачи, как распознавание сарказма, иронии или отрицания.
- *Преобразование текста* — переводы, исправления ошибок, изменения стиля, перефразирование, резюмирование.
- *Понимание* — диалоги в форме «вопрос — ответ», рассуждения, дополнение знаний.

Многие из этих задач могут распространяться на сферу занятий специалистов data science или инженеров по данным (data engineer). LLM могут поддерживать пользователя при решении таких задач, как генерация признаков, написание

кода и визуализация (например, с использованием команд из пакета matplotlib), снабжать его аналитическими рекомендациями, помогая интерпретировать результаты и представляя итоговые выводы для диаграмм и отчетов. Одним из заметных практических применений LLM в задачах обработки табличных данных является автоматизация операций работы с текстом. При работе с текстовыми столбцами LLM могут генерировать новые признаки, реферировать тексты, классифицировать их и определять их тематику. Они также могут писать код обработки на Python, например создавая функции или формируя корректное регулярное выражение для обработки текста.

Помимо поддержки пользователя и предоставления помощи, LLM также могут играть более явную и активную роль в аналитике. Последние приложения для ChatGPT (например, API Advanced Data Analytics) обеспечивают непосредственный анализ данных в формате CSV, а также решают смежные задачи, связанные с данными, включая реферирование, предварительную обработку, анализ, визуализацию и создание отчетов. На каждом этапе инструмент может предоставить код Python для выполнения и получения результатов, самостоятельно запускать его и визуализировать результаты в виде диаграмм и таблиц. Это соответствует возможностям TableGPT или других инструментов, таких как MediTab. TableGPT (https://arxiv.org/pdf/2307.08674.pdf) — это новый фреймворк, который использует LLM для улучшения взаимодействия человека с табличными данными. Он позволяет пользователям с помощью команд, выраженных на естественном языке, выполнять различные задачи: отвечать на вопросы, работать с данными, создавать визуализации, генерировать отчеты и даже строить прогнозы. MediTab (https://arxiv.org/pdf/2305.12081.pdf) работает с медицинскими табличными данными: объединяет семплы, адаптирует информацию из внешних источников к целевой задаче и расширяет объем обучающих данных. На материале задач прогнозирования на текстовых данных MediTab продемонстрировал производительность, превосходящую классические алгоритмы машинного обучения, такие как XGBoost.

В целом LLM не обеспечивают сопоставимой эффективности в задачах прогнозирования на табличных данных, как показано в тесте TABLET (https://arxiv.org/pdf/2304.13188.pdf). Для оценки производительности LLM относительно моделей машинного обучения с учителем (fully supervised) в статье сравнивались модели Flan-T5 11b и ChatGPT (с использованием четырех примеров с подсказками) с моделью XGBoost, обученной на всем датасете. Модель XGBoost, примененная ко всем данным, достигла среднего показателя F1 0.94 в задачах прогнозирования. Для сравнения, ChatGPT набрала в среднем 0.68 балла, а Flan-T5 11b — 0.66. Этот анализ показал, что производительность LLM все еще имеет потенциал для роста, в том числе в задачах с разнородными данными (текст и числа), в то время как эти инструменты по-прежнему превосходно выполняют инструкции, особенно при обработке текстовых входных данных

и генерации текстовых выходных. Такой инструмент, как llm-classifier (https://github.com/lamini-ai/llm-classifier), может использовать информацию, уже заложенную в LLM, но не умеет извлекать дополнительную, типичную для табличных задач, и это вызывает удивление.

Подводя итог, можно сказать, что генеративный ИИ пока не является безусловно хорошим решением для работы с табличными данными не только из-за производительности, но и по другим важным причинам, таким как:

- *затратность* — генеративные модели требуют значительных ресурсов графического процессора, что приводит к более высоким эксплуатационным расходам;
- *масштабируемость* — ресурсоемкость моделей генеративного ИИ, особенно их зависимость от графических процессоров, может препятствовать их масштабируемости;
- *задержка и пропускная способность* — с ростом модели, как правило, увеличивается время обработки каждого запроса, что негативно влияет на задержку и пропускную способность;
- *смещение*, или *предвзятость* (bias), — генеративные модели могут наследовать смещение обучающих данных, потенциально закрепляя или усиливая его;
- *гибкость* — адаптация моделей генеративного ИИ к конкретным задачам часто требует масштабного дообучения, что ограничивает их гибкость;
- *детерминированность* — сложность архитектуры, присущая генеративным моделям, может затруднить контроль и прогнозирование результатов, что влияет на их устойчивость и воспроизводимость;
- *интерпретируемость* — сложность генеративных моделей может терять объяснимость, затрудняя понимание того, как они работают и как приходят к результатам.

Учитывая эти ограничения, мы сосредоточимся на основных классических методах машинного и глубокого обучения, а также на том, как правильно и тщательно подготовить данные для анализа. Однако мы также уделим внимание инструментам генеративного ИИ, таким как ChatGPT, Google Gemini и Gemini для Google Cloud, поскольку признаем перспективность этих технологий для анализа табличных данных. Основываясь на нашем опыте в этой области, мы не видим LLM в качестве полноценной замены классическим алгоритмам машинного обучения или архитектурам глубокого обучения в силу преимуществ, которые предлагают традиционные инструменты — и с точки зрения эффективности, и в плане контролируемости. Однако в качестве вспомогательного средства LLM и другие модели генеративного ИИ могут дополнить обработку

таблиц, их анализ и моделирование, повышая квалификацию специалистов и производительность проектов.

Итоги

- Табличные данные — это данные, организованные в строки и столбцы, как, например, в CSV-файлах или таблицах реляционных баз.
- Структурированные данные иногда используются как альтернативный термин для табличных, но это более широкое понятие, включающее, например, данные в формате JSON.
- Табличные данные составляют небольшую часть всех цифровых данных в мире, но оказывают огромное влияние на нашу жизнь.
- В отличие от иных типов данных (например, изображения, видео, текст, аудио), таблицы являются наиболее распространенными в бизнесе, поэтому изучение того, как эффективно применять к ним машинное и глубокое обучение, — полезный для многих людей навык.
- В этой книге мы называем *классическим машинным обучением* или просто *машинным обучением* все, от линейной регрессии до градиентного бустинга, исключая нейронные сети, чтобы различать эти две группы методов.
- По сравнению с глубоким обучением для других типов данных (например, изображений, видео, текста, аудио) глубокое обучение для табличных данных привлекает меньше внимания со стороны исследователей.
- Традиционно с табличными данными применяется метод градиентного бустинга, такой как XGBoost.
- В социальных сетях развернулась оживленная дискуссия о том, есть ли место глубокому обучению в решении задач, связанных с табличными данными. Мы не встаем ни на чью сторону в этом споре. Вместо этого мы попытаемся объективно обосновать, почему можно использовать машинное или глубокое обучение для конкретной задачи, а также познакомим читателя с лучшими практиками использования каждого подхода.
- Табличные данные имеют некоторые особенности, которые не свойственны другим типам данных, таким как изображения, видео или текст, а именно: отсутствие больших датасетов с открытым исходным кодом, аналогичных тем, которые можно увидеть в реальных бизнес-задачах.
- Генеративный ИИ, особенно LLM, существенно влияет на то, как в целом воспринимается искусственный интеллект, как он распространяется среди людей и организаций и как используется. LLM могут помочь автоматизировать различные задачи, связанные с анализом и моделированием табличных данных, особенно когда это касается моделей с текстовым входом и выходом.