УДК 004.4
ББК 32.972
Х12

Хапке Х., Нельсон К.
Х12 Разработка конвейеров машинного обучения / пер. с анг. Н. Б. Желновой. – М.: ДМК Пресс, 2021. – 346 с.: ил.

ISBN 978-5-97060-886-9

Машинное обучение становится важным элементом почти во всех отраслях. В этой книге представлено четкое и понятное руководство по автоматизации развертывания, управления и повторного использования моделей машинного обучения. Шаг за шагом описывается конкретный пример проекта, на котором можно отработать основные навыки в этой сфере. Благодаря множеству примеров кода и ясным, лаконичным объяснениям вы сможете создать свой собственный конвейер машинного обучения и запустите его в кратчайшие сроки.

Книга поможет ученым и инженерам, специализирующимся в области машинного обучения и искусственного интеллекта, выйти за рамки работы с единичной моделью и успешно реализовать свои проекты в области науки о данных. Также издание будет полезно менеджерам проектов в области науки о данных, разработчикам программного обеспечения и инженерам DevOps, которые хотят, чтобы их организация ускорила свои проекты, использующие технологии машинного обучения и искусственного интеллекта.

Читателю понадобится знание основных концепций машинного обучения и хотя бы одного из фреймворков, используемых в машинном обучении (например, PyTorch, TensorFlow, Keras).

УДК 004.4
ББК 32.972

ISBN (анг.) 978-1492053194
ISBN (рус.) 978-5-97060-886-9

Оглавление

Предисловие

Когда в 1913 году компания Генри Форда построила свой первый сборочный конвейер для производства своей легендарной Model T, время, необходимое для сборки каждой машины, сократилось с 12 до 3 часов. Затраты на производство резко снизились, что позволило Model T стать первым доступным автомобилем в истории. Это также сделало возможным массовое производство: вскоре Model T стала королевой автомобильных дорог.

Поскольку производственный процесс теперь представлял собой четкую последовательность четко определенных шагов (он же конвейер), стало возможно автоматизировать некоторые из этих шагов, сэкономив еще больше времени и денег. Сегодня производство автомобилей невозможно без автоматики.

Но дело не только во времени и деньгах. При выполнении многих повторяющихся задач машина будет производить гораздо более стабильный результат, чем люди, в результате чего конечный продукт станет более предсказуемым, последовательным и надежным. Наконец, избавляя людей от тяжелого физического труда, автоматика позволяет значительно повысить безопасность. Многие рабочие от монотонной физической работы перешли к работе, требующей более высокого уровня квалификации (хотя, честно говоря, многие просто потеряли работу).

Если посмотреть на автоматизацию с другой стороны, установка автоматизированной сборочной линии может занять много времени и стать дорогостоящим проектом. Кроме того, сборочный конвейер – не идеальное решение, если вы хотите производить небольшие партии или продукцию по индивидуальному заказу. Форд сказал: «Цвет автомобиля может быть любым, при условии что он будет черным».

История автомобилестроения повторилась в индустрии программного обеспечения за последние пару десятилетий: каждая значительная часть программного обеспечения в настоящее время создается, тестируется и развертывается с использованием таких инструментов автоматизации, как Jenkins или Travis. Однако метафоры Model T уже недостаточно. Программное обеспечение не просто развертывается и работает как есть; его необходимо регулярно контролировать, поддерживать и обновлять. Программные конвейеры теперь больше похожи на динамические циклы, чем на статические производственные линии. Крайне важно иметь возможность быстро обновлять программное обеспечение (или сам конвейер), не нарушая его целостности. А программное обеспечение гораздо более вариативно, чем когда-либо была Model T: программное обеспечение можно раскрасить в любой цвет (попробуйте, например, подсчитать количество существующих вариантов для MS Office).

К сожалению, «классические» инструменты автоматизации не подходят для создания полноценного конвейера машинного обучения. Действительно, модель машинного обучения не является обычным программным обеспечением.

Во-первых, большая часть его поведения определяется данными, на которых он обучается. Следовательно, сами обучающие данные должны рассматриваться как код (и, соответственно, иметь версии). Это довольно сложная проблема, потому что новые данные появляются каждый день (часто в больших количествах), изменяются и дрейфуют с течением времени, часто включают персональные данные; также новые данные должны быть размечены, прежде чем вы сможете передать их в работу, которую выполняют алгоритмы машинного обучения.

Во-вторых, поведение модели нередко бывает довольно непрозрачным: она может пройти все тесты для одних данных, но полностью потерпеть неудачу для других. Таким образом, вы должны убедиться, что ваши тесты охватывают все области данных, на которых ваша модель будет использоваться в производстве. В частности, вы должны убедиться, что она не проявляет дискриминацию в отношении какой-либо группы ваших пользователей.

По этим (и другим) причинам специалисты по обработке данных и инженеры-программисты сначала начали создавать и обучать модели машинного обучения вручную, так сказать, «в своем гараже», и многие из них до сих пор это делают. Но за последние несколько лет были разработаны новые инструменты автоматизации, которые решают задачи конвейеров машинного обучения, такие как TensorFlow Extended (TFX) и Kubeflow. Все больше и больше организаций начинают использовать эти инструменты для создания конвейеров машинного обучения, которые автоматизируют большую часть (или все) этапов построения и обучения моделей машинного обучения. Преимущества этой автоматизации в основном те же, что и для автомобильной промышленности: экономия времени и денег; возможность создавать более качественные, надежные и безопасные модели и тратить больше времени на выполнение более полезных задач, чем на копирование данных или изучение кривых обучения. Однако построить конвейер машинного обучения непросто. Так с чего же начать?

Начать с этой книги!

В этой книге Ханнес и Кэтрин дают четкое и понятное руководство по автоматизации конвейеров машинного обучения. Как твердому сторонникупрактического подхода, особенно для такой технической темы, мне особенно понравилось то, как эта книга шаг за шагом проведет вас через конкретный пример проекта от начала до конца. Благодаря множеству примеров кода и ясным, лаконичным объяснениям вы сможете создать свой собственный конвейер машинного обучения и запустить его в кратчайшие сроки, а также все концептуальные инструменты, необходимые для адаптации этих конвейеров машинного обучения к вашим собственным вариантам использования. Я настоятельно рекомендую вам взять свой ноутбук и попробовать что-то во время чтения: так вы научитесь намного быстрее.

Я впервые встретился с Ханнесом и Кэтрин в октябре 2019 года на конференции TensorFlow World в Санта-Кларе, Калифорния, где я делал доклад о создании конвейеров машинного обучения с использованием TFX. Они работали над этой книгой по той же теме, и у нас был один редактор, так что, естественно, нам было о чем поговорить. Некоторые слушатели задавали технические вопросы о TensorFlow Serving (который является частью TFX), и у Ханнеса и Кэтрин были все ответы, которые я искал. Ханнес даже любезно принял мое при-

глашение выступить с докладом о расширенных функциях TensorFlow Serving в конце моего курса в очень короткие сроки. Его выступление было сокровищницей идей и полезных советов, которые вы найдете в этой книге, а также во многих, многих других.

Пришло время приступить к созданию профессиональных конвейеров машинного обучения!

– *Орелиен Жерон*,
бывший руководитель группы классификации видео YouTube,
автор книги «Практическое машинное обучение
с использованием Scikit-Learn, Keras и TensorFlow» (O'Reilly)
Окленд, Новая Зеландия, 18 июня 2020 г.

Введение

Все говорят о машинном обучении. Из академической дисциплины оно превратилось в одну из самых удивительных технологий. Машинное обучение используется везде – от обработки видеопотока, регистрируемого в автомобилях с автоматическим управлением, до персонализации назначений лекарств. Оно становится важным элементом в каждой отрасли. В то время как моделям архитектуры и концепциям уделялось большое внимание, машинное обучение еще не прошло стадию стандартизации процессов, появлявшихся в отрасли программного обеспечения в последнее десятилетие. В этой книге мы хотели бы показать вам, как создать стандартизированную систему машинного обучения, которая была бы автоматизированной и воспроизводимой.

За последние несколько лет разработки в области машинного обучения достигли впечатляющих результатов. Благодаря широкой доступности графических процессоров (Graphical Processing Units, GPU) и разработке новых концепций глубокого обучения, таких как Transformers (например, BERT) или Generative Adversarial Networks (например, DCGAN), количество проектов в области искусственного интеллекта, или ИИ (Artificial Intelligence, AI), резко возросло. Количество стартапов в сфере ИИ огромно. Корпорации применяют новейшие технологии машинного обучения для решения всех видов бизнес-задач. В этом стремлении к наиболее эффективному решению для задач машинного обучения мы наблюдали несколько вещей, которым уделялось меньше внимания. Мы увидели, что специалистам в области ИИ – ученым, специализирующимся в области машинного обучения и искусственного интеллекта, и инженерам по машинному обучению – не хватает хороших источников информации для создания концепций и инструментов для ускорения, повторного использования, управления и развертывания своих разработок. Необходима стандартизация конвейеров машинного обучения.

Конвейеры машинного обучения – это процессы для ускорения, повторного использования, управления и развертывания моделей машинного обучения. Примерно десять лет назад разработка программного обеспечения претерпела такие же изменения благодаря внедрению непрерывной интеграции (Continuous Integration, CI) и непрерывного развертывания (Continuous Deployment, CD). Когда-то это был длительный процесс тестирования и развертывания веб-приложения. В наши дни эти процессы были значительно упрощены с помощью нескольких инструментов и концепций. Ранее для развертывания веб-приложений требовалось сотрудничество между инженером DevOps и разработчиком программного обеспечения. Сегодня приложение можно надежно протестировать и развернуть за считанные минуты. Специалисты по обработке данных и инженеры машинного обучения могут заимствовать концепции рабочих процессов из программной инженерии.

Исходя из нашего личного опыта, большинство проектов в области науки о данных, нацеленных на внедрение моделей в производство, не могут позволить себе роскошь создавать большие команды, что затрудняет построение всего конвейера собственными силами с нуля. Это может означать, что проекты машинного обучения превращаются в одиночные попытки построения моделей, и их результативность ухудшается со временем; специалист тратит большую часть своего времени на исправление ошибок, когда меняются данные, лежащие в основе решения, или модель не находит широкого применения. Автоматизированный конвейер, позволяющий построить воспроизводимый сквозной рабочий процесс, уменьшает усилия, необходимые для развертывания модели. Конвейер машинного обучения должен включать процессы, которые:

- эффективно отслеживают изменение версий исходных данных и запускают новое обучение моделей;
- эффективно выполняют предварительную обработку данных для обучения и проверки модели;
- следят за версиями контрольных точек модели во время обучения;
- отслеживают ваши эксперименты по обучению моделей;
- анализируют и проверяют обученные и настроенные модели;
- выполняют развертывание проверенных моделей;
- масштабируют развернутую модель;
- собирают новые данные для обучения и регистрируют показатели точности модели, используя петли обратной связи.

В этом списке пропущен один важный момент: обучение и настройка модели. Мы предполагаем, что вы уже обладаете достаточными знаниями и опытом для выполнения этого шага. Если же вы только начинаете погружаться в тему машинного или глубокого обучения, следующие книги, опубликованные O'Reilly, станут отличной отправной точкой для знакомства с машинным обучением:

- Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms 1st Edition by Nikhil Buduma, Nicholas Locascio (*Никил Будума*, *Николас Локасио*. Основы глубокого обучения: разработка алгоритмов искусственного интеллекта следующего поколения. 1-е изд.);
- Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition by Aurélien Géron (*Аурелиен Жерон*. Практическое машинное обучение с использованием Scikit-Learn, Keras и TensorFlow: концепции, инструменты и методы для построения интеллектуальных систем. 2-е изд.).

Для кого предназначена эта книга

Основная аудитория этой книги – ученые, специализирующиеся в области машинного обучения и искусственного интеллекта и инженеры по машинному обучению, которые хотят выйти за рамки обучения единичной модели машинного обучения и успешно реализовать свои проекты в области науки о данных. Вы должны быть знакомы с основными концепциями машинного обучения и хотя бы с одним из фреймворков, используемых в машинном

обучении (например, PyTorch, TensorFlow, Keras). Примеры в этой книге основаны на TensorFlow и Keras, но основные концепции могут быть применены к любой среде.

Также эта книга может быть полезна менеджерам проектов в области науки о данных, разработчикам программного обеспечения или инженерам DevOps, которые хотят, чтобы их организация ускорила свои проекты, использующие технологии машинного обучения и искусственного интеллекта. Если вы заинтересованы в лучшем понимании жизненного цикла разработки систем на основе автоматизированного машинного обучения и хотите понять, как это может принести пользу вашей организации, то в следующих главах будет представлен набор инструментов для достижения данной цели.

Почему мы используем TensorFlow и TensorFlow Extended

В этой книге во всех наших примерах конвейера будут использоваться инструменты из экосистемы TensorFlow, в частности TensorFlow Extended (TFX). Мы выбрали этот фреймворк по ряду причин:

- экосистема TensorFlow является наиболее доступной для машинного обучения на момент написания статьи. Она включает в себя несколько полезных проектов и библиотек поддержки, таких как TensorFlow Privacy и TensorFlow Probability;
- она популярна и широко используется как в малых, так и на крупных производственных предприятиях, и существует активное сообщество заинтересованных пользователей;
- поддерживаемые варианты использования простираются от академических исследований до машинного обучения в производственной среде. TFX тесно интегрирован с базовой платформой TensorFlow для использования в производственных процессах;
- и TensorFlow, и TFX являются инструментами с открытым исходным кодом, и нет никаких ограничений на их использование.

Однако все принципы, которые мы описываем в этой книге, применимы и к другим инструментам и фреймворкам.

Обзор глав

В следующих главах мы представим конкретные шаги для построения конвейеров машинного обучения и продемонстрируем, как они работают, на примере демонстрационного проекта.

Глава 1 «Введение». В этой главе представлен обзор конвейеров машинного обучения, обсуждается, когда их следует использовать, и описываются все этапы, входящие в состав конвейера. Также мы представляем здесь пример проекта, который будем использовать на протяжении всей книги.

Глава 2 «Введение в TensorFlow Extended» знакомит читателей с экосистемой TFX, объясняет, как задачи взаимодействуют друг с другом, и описывает внутреннюю работу компонентов TFX. Мы также рассмотрим ML MetadataStore,

покажем, как он используется в контексте TFX, и как Apache Beam запускает компоненты TFX без вмешательства пользователя.

Глава 3 «Загрузка данных» посвящена процессу систематической загрузки данных в наши конвейеры, а также в ней обсуждается концепция управления версиями данных.

Глава 4 «Проверка данных» объясняет, каким образом организовать эффективную проверку данных, поступающих в ваш конвейер, с помощью инструмента TensorFlow Data Validation. Система предупредит вас, если новые данные существенно изменятся по сравнению с предыдущими данными, что может повлиять на точность вашей модели.

Глава 5 «Предварительная обработка данных» посвящена подготовке данных (конструированию признаков) с использованием TensorFlow Transform для преобразования необработанных данных в признаки, подходящие для обучения модели.

Глава 6 «Обучение модели» объясняет, как обучать модели в рамках задачи машинного обучения. В этой главе мы также объясним концепцию настройки модели.

Глава 7 «Анализ и проверка модели» содержит полезные метрики для понимания, как работает ваша модель в производственной среде. В число этих метрик входят и те, которые могут позволить вам выявить ошибки в предсказаниях модели. В разделе «Анализ и проверка модели в TFX» объясняется, как управлять версиями вашей модели при улучшении одного из показателей в новой версии. Модель в конвейере может быть автоматически обновлена до новой версии.

Глава 8 «Развертывание моделей с помощью TensorFlow Serving» посвящена тому, как эффективно развернуть модель машинного обучения. Начиная с простой реализации Flask мы подчеркиваем ограничения в использовании таких пользовательских моделей в приложениях. Мы расскажем о TensorFlow Serving и о том, как настроить ваши исполнительные экземпляры. Мы также обсуждаем его пакетный режим работы и демонстрируем настройку клиентов для запроса результатов прогнозирования на основе модели.

Глава 9 «Расширенные варианты развертывания моделей с помощью TensorFlow Serving» показывает, как оптимизировать ваши варианты развертывания модели и как их контролировать. Мы обсуждаем стратегии оптимизации ваших моделей TensorFlow для повышения производительности. Мы также продемонстрируем базовый сценарий настройки развертывания с использованием Kubernetes.

Глава 10 «Расширенные концепции TensorFlow Extended» представляет концепцию специальных компонентов для ваших конвейеров машинного обучения, так что вы не ограничены стандартными компонентами в TFX. Если вы хотите добавить дополнительные этапы загрузки данных или преобразовать экспортированные модели в TensorFlow Lite (TFLite), мы проведем вас через все этапы создания таких компонентов.

Глава 11 «Конвейеры, часть 1: Apache Beam и Apache Airflow» собирает воедино все, что обсуждалось в предыдущих главах. Мы обсудим, как превратить ваши компоненты в конвейеры и как их настроить для выбранной платформы оркестровки. Мы также проведем вас от начала до конца, через все шаги конвейера, работающего на Apache Beam и Apache Airflow.

Глава 12 «Конвейеры, часть 2: Конвейеры Kubeflow Pipelines» является продолжением предыдущей главы. В ней рассматривается построение конвейера с использованием Kubeflow и платформы искусственного интеллекта Google.

Глава 13 «Петли обратной связи» посвящена тому, как превратить ваш модельный конвейер в цикл, позволяющий вносить улучшения на основе отзывов пользователей конечного продукта. Мы обсудим, какие типы данных необходимо собирать, чтобы улучшить модель в будущих версиях, и как передать эти данные обратно в конвейер.

Глава 14: «Приватность данных, используемых для машинного обучения» представляет концепцию быстро меняющейся области машинного обучения с сохранением приватности данных. В ней обсуждается три важных элемента этой концепции: дифференцированная приватность, федеративное обучение и зашифрованное машинное обучение.

Глава 15 «Будущее конвейеров машинного обучения и следующие шаги» содержит обзор технологий, которые повлияют на будущие конвейеры машинного обучения и на то, как мы будем смотреть на машинное обучение в ближайшие годы.

Приложение A «Введение в инфраструктуру машинного обучения» содержит краткое введение в Docker и Kubernetes.

Приложение B «Настройка кластера Kubernetes в Google Cloud» содержит дополнительные материалы по настройке Kubernetes в Google Cloud.

Приложение C «Советы по работе с конвейерами Kubeflow» содержит несколько полезных советов по работе с настройками конвейеров Kubeflow, включая обзор интерфейса командной строки TFX.

Условные обозначения, используемые в этой книге

В этой книге используются следующие условные обозначения.

Курсив

Обозначает новые термины, URL-адреса, адреса электронной почты, имена файлов и расширения файлов.

`Моноширинный шрифт`

Используется для оформления листингов программ, а также в тексте для обозначения фрагментов программы, таких как имена переменных или функций, базы данных, типы данных, переменные среды, операторы и ключевые слова.

`Жирный моноширинный шрифт`

Показывает команды или другой текст, который пользователь должен набирать точь-в-точь так, как показано в книге.

`Моноширинный шрифт, выделенный курсивом`

Показывает текст, который следует заменить пользовательскими значениями или значениями, определяемыми контекстом.

Данный элемент означает совет или предложение.

Данный элемент обозначает общее примечание.

Так обозначаются предупреждения и предостережения.

Использование примеров кода

Дополнительные материалы (примеры кода и т. п.) доступны для загрузки по ссылке https://oreil.ly/bmlp-git.

Если у вас есть технический вопрос или проблема с использованием примеров кода, отправьте электронное письмо на адрес bookquestions@oreilly.com и buildingmlpipelines@gmail.com.

Эта книга предназначена для того, чтобы помочь вам в работе над проектами. Вы можете использовать приведенные в этой книге примеры кода в своих программах и документации. Вам не нужно связываться с нами для получения разрешения на использование этих примеров, если вы используете незначительную часть кода. Например, для написания программы, использующей несколько фрагментов кода из этой книги, не требуется разрешения. Однако для коммерческого использования или распространения примеров из книг O'Reilly потребуется разрешение. Для ответа на вопрос с использованием цитат из этой книги и примеров кода разрешение не требуется. Но для включения значительного количества примеров кода из этой книги в документацию к вашему программному продукту потребуется разрешение.

Мы ценим, но не требуем указания авторства. Обычно при цитировании указывается название, автор, издатель и ISBN, например: «Создание конвейеров машинного обучения», Ханнес Хапке и Кэтрин Нельсон (O'Reilly). © 2020 Ханнес Хапке и Кэтрин Нельсон, 978-1-492-05319-4».

Если вы считаете, что ваше использование примеров кода выходит за рамки условий добросовестного использования или разрешений, приведенных выше, обращайтесь к нам по адресу электронной почты permissions@oreilly.com.

Онлайн-обучение O'Reilly

На протяжении более 40 лет O'Reilly Media разрабатывает технологические и бизнес-тренинги, продвигая знания и развивая идеи, чтобы помочь компаниям добиться успеха.

Наша уникальная сеть экспертов и новаторов делится своими знаниями и опытом через книги, статьи и нашу платформу онлайн-обучения. Платформа онлайн-обучения O'Reilly предоставляет по запросу доступ к курсам обучения в режиме реального времени, схемам углубленного обучения, интерактивным средам программирования и обширной коллекции текстов и видео от O'Reilly и более 200 других издателей. Для получения дополнительной информации перейдите по ссылке http://oreilly.com.

Как с нами связаться

Оба автора этой книги хотели бы поблагодарить вас за уделенное ей внимание. Если вы хотите связаться с ними, вы можете сделать это через их веб-сайт www.buildingmlpipelines.com или по электронной почте buildingmlpipelines@gmail.com. Авторы желают вам всяческих успехов в создании собственных конвейеров машинного обучения.

Комментарии и вопросы, касающиеся этой книги, просьба направлять издателю:

O'Reilly Media, Inc.
1005 Gravenstein Highway Северный Севастополь, CA 95472
800-998-9938 (в США или Канаде) 707-829-0515 (международный или местный)
707-829-0104 (факс)

Для этой книги разработана веб-страница, где мы публикуем исправления, примеры, а также другую дополнительную информацию. Для того чтобы ознакомиться с материалами, размещенными на этой странице, перейдите по ссылке: https://oreil.ly/build-ml-pipelines.

Для отправки комментариев или технических вопросов по содержанию данной книги используйте адрес электронной почты bookquestions@oreilly.com.

Новости и информацию о наших книгах и курсах можно найти на сайте http://oreilly.com.

Наша страница в Facebook: http://facebook.com/oreilly.

Мы в Twitter: http://twitter.com/oreillymedia.

Наш канал YouTube: http://www.youtube.com/oreillymedia.

Благодарности

В процессе написания этой книги нас поддерживали многие замечательные люди. Большое спасибо всем, кто помог воплотить в жизнь наши замыслы! Мы хотели бы выразить особую благодарность всем вам.

Со всей командой O'Reilly было здорово работать все время, пока готовилась эта книга. Спасибо нашим редакторам Мелиссе Поттер (Melissa Potter), Николь Таше (Nicole Taché) и Амелии Блевинс (Amelia Blevins) за постоянную поддержку и вдумчивые отзывы. Спасибо также Кэти Тозер (Katie Tozer) и Джонатану Хасселлу (Jonathan Hassell) за оказанную нам поддержку.

Спасибо Орелиену Жерону (Aurélien Géron), Роберту Кроу (Robert Crowe), Маргарет Мейнард-Рид (Margaret Maynard-Reid), Сергею Хоменко (Sergii Khomenko) и Викраму Тивари (Vikram Tiwari), которые внимательно просмот-

рели всю книгу и дали множество полезных советов и ценных комментариев. Вы много сделали для того, чтобы ее окончательный вариант стал лучше. Спасибо за часы, потраченные вами на внимательное изучение материалов книги.

Спасибо Янну Дюпису (Yann Dupis), Джейсону Манкузо (Jason Mancuso) и Мортену Далю (Morten Dahl) за ваш тщательный и всесторонний обзор главы о конфиденциальности машинного обучения.

Мы получили фантастическую поддержку от многих замечательных людей в Google. Благодарим вас за помощь в поиске и исправлении ошибок, а также за то, что вы выпускаете инструменты машинного обучения в виде пакетов с открытым исходным кодом! Помимо сотрудников Google, мы хотим особо поблагодарить Эми Унру (Amy Unruh), Анушу Рамеш (Anusha Ramesh), Кристину Грир (Christina Greer), Клеменс Мевальд (Clemens Mewald), Дэвида Заца (David Zats), Эдда Уайлдера-Джеймса (Edd Wilder-James), Ирен Джаннумис (Irene Giannoumis), Ярека Вилькевича (Jarek Wilkiewicz), Джайи Чжао (Jiayi Zhao), Джири Симсу (Jiri Simsa), Константиноса Катсиаписана (Konstantinos Katsiapis), Лак Лакшана (Lak Lakshmanan), Майка Древеса (Mike Dreves), Пейджа Бейли (Paige Bailey), Педрама Пеймана (Pedram Pejman), Сару Робинсон (Sara Robinson), Сунсон Квон (Soonson Kwon), Тею Ламкин (Thea Lamkin), Триса Варкентина (Tris Warkentin), Варшу Наганатан (Varshaa Naganathan), Чжитао Ли (Zhitao Li) и Зохара Яхава (Zohar Yahav).

Мы выражаем огромную благодарность сообществу TensorFlow и Google Developer Expert и его замечательным участникам. Спасибо за поддержку в этом начинании.

Спасибо другим участникам, которые помогали на разных этапах: Барбаре Фусинска (Barbara Fusinska), Хамелю Хусайну (Hamel Husain), Михалу Ястшембскому (Michał Jastrzębski) и Яну Хенселю (Ian Hensel).

Спасибо сотрудникам Concur Labs (прошлым и настоящим) и другим сотрудникам SAP Concur за плодотворные обсуждения и полезные идеи. В частности, спасибо Джону Дитцу (John Dietz) и Ричарду Пакетту (Richard Puckett) за неоценимую поддержку, оказанную авторам книги.

Ханнес

Я хочу поблагодарить мою замечательную партнершу Уитни за ее огромную поддержку на протяжении всего времени работы над этой книгой. Спасибо за постоянную поддержку, за отклики и за терпение, когда я провожу долгие часы за написанием статей. Спасибо моей семье, особенно моим родителям, которые позволили мне следовать за моей мечтой по всему миру.

Эта книга не появилась бы, не будь у меня замечательных друзей. Спасибо, Коул Ховард (Cole Howard), за то, что вы прекрасный друг и учитель. Наше сотрудничество подтолкнуло меня к этой работе и размышлениям о конвейерах машинного обучения. Моим друзьям Тимо Мецгеру (Timo Metzger) и Аманде Райт (Amanda Wright): спасибо за то, что вы улучшили язык этой книги. И спасибо Еве и Килиану Рамбах (Eva Rambach, Kilian Rambach), а также Деб и Дэвиду Хаклеманам (Deb Hackleman, David Hackleman). Без вашей помощи я бы не добрался до Орегона.

Я хотел бы поблагодарить моих предыдущих работодателей, Cambia Health, Caravel и Talentpair, за то, что они позволили мне реализовать концепции этой публикации в производственных условиях, несмотря на то что эти идеи были новаторскими.

Эта книга не вышла бы без моего соавтора Екатерины. Спасибо за дружбу, поддержку и бесконечное терпение. Я рад, что мы познакомились по чистой случайности в жизни. Я очень рад, что мы вместе написали эту книгу.

Екатерина

Я написала много слов в этой книге, но нет слов, чтобы выразить, насколько я ценю поддержку, которую оказал мне мой муж Майк. Спасибо за теплую поддержку, за приготовленную еду, полезные обсуждения, сарказм и ценные отзывы. Спасибо моим родителям за то, что давным-давно поддержали мой интерес к программированию, – чтобы вырасти, мне было нужно время, но вы всегда были правы!

Спасибо всем замечательным сообществам, частью которых мне посчастливилось быть. Я встретила множество замечательных людей через Seattle PyLadies, Women in Data Science и более широкое сообщество Python. Я очень ценю вашу поддержку.

И спасибо Ханнесу за то, что он пригласил меня в это путешествие! Без тебя ничего бы не случилось! Ваши знания, внимательность и настойчивость сделали весь этот проект успешным. И это было очень весело!

Глава 1

Введение

В первой главе этой книги мы познакомимся с конвейерами машинного обучения и рассмотрим все этапы их создания. Мы объясним, что должно произойти, чтобы превратить вашу модель машинного обучения из экспериментальной в надежную производственную систему. Мы также представим наш пример проекта, который будем использовать в оставшейся части книги, чтобы продемонстрировать описанные нами принципы.

Почему и где используются конвейеры машинного обучения

Ключевое преимущество конвейеров машинного обучения заключается в автоматизации этапов жизненного цикла модели. Когда становятся доступными новые обучающие данные, должен быть запущен рабочий процесс, который включает проверку данных, предварительную обработку, обучение модели, анализ и развертывание. Мы наблюдали, как слишком много рабочих групп, занятых анализом данных, вручную выполняют эти шаги. Это обходится очень дорого и часто является источником ошибок. Давайте подробно рассмотрим некоторые из преимуществ конвейеров машинного обучения.

Возможность сосредоточиться на новых моделях, а не на поддержке существующих моделей

Автоматизированные конвейеры машинного обучения освободят специалистов по обработке данных от необходимости поддерживать существующие модели. Мы видели, как множество специалистов по данным тратят человекодни на обновление ранее разработанных моделей. Они вручную запускают сценарии для предварительной обработки своих обучающих данных, они пишут одноразовые сценарии развертывания или вручную настраивают свои модели. Автоматизированные конвейеры позволяют специалистам по обработке данных разрабатывать новые модели – ведь это гораздо более интересно. В конечном итоге это будет способствовать повышению удовлетворенности работой и удержанию персонала на конкурентном рынке труда.

Предотвращение ошибок

Автоматизированные конвейеры машинного обучения могут предотвратить ошибки. Как мы увидим в следующих главах, вновь созданные модели

будут привязаны к набору версионированных данных, а шаги предварительной обработки будут привязаны к разработанной модели. Это означает, что при сборе новых данных будет сгенерирована новая модель. При изменении шагов предварительной обработки обучающие данные станут недействительными, и будет сгенерирована новая модель. В рабочих процессах ручного машинного обучения обычным источником ошибок является изменение этапа предварительной обработки после обучения модели. В этом случае мы бы развернули модель с инструкциями обработки, отличными от тех, при помощи которых мы обучали модель. Подобные ошибки может быть действительно сложно отладить, модель все еще выдает результат, но этот результат просто неверен. Такие ошибки можно предотвратить с помощью автоматизированных рабочих процессов.

Полезная документация

Средства отслеживания эксперимента и управления версиями модели генерируют протоколы, в которых хранится история изменений модели. Отслеживание эксперимента включает в себя отслеживание изменений гиперпараметров модели, используемых наборов данных и результирующих метрик модели (таких как, например, потери или точность). Инструменты управления версиями модели будут помогать отслеживать, какая модель была в конечном итоге выбрана и развернута. Такая документация особенно полезна, если команде специалистов в области машинного обучения и искусственного интеллекта необходимо повторно создать модель или отследить качество модели.

Стандартизация

Стандартизированные конвейеры машинного обучения улучшают работу команды специалистов в области машинного обучения и искусственного интеллекта. Благодаря стандартизированным настройкам инженеры, занимающиеся обработкой данных, могут быстро подключаться к работе или переходить из одной группы в другую и работать с одной и той же средой разработки. Это повышает их эффективность и сокращает время, затрачиваемое на вход в новый проект. Время, затраченное на настройку конвейеров машинного обучения, также может способствовать повышению коэффициента удержания персонала.

Бизнес-модель для конвейеров машинного обучения

Внедрение конвейеров машинного обучения позволит командам, занимающимся разработкой в области машинного обучения, достичь следующих результатов:

- уменьшение времени разработки новых моделей;
- упрощение процессов обновления существующих моделей;
- уменьшение затрат времени на воспроизведение моделей.

Все эти аспекты ведут к уменьшению стоимости проектов, реализуемых в области машинного обучения и искусственного интеллекта. Более того, конвейеры машинного обучения обладают следующими преимуществами:

- помогают обнаружить потенциальные смещения в наборах данных или в обученных моделях. Обнаружение таких смещений может предотвратить нанесение ущерба людям, которые используют результаты построения модели. Например, система автоматического просмотра резюме на базе машинного обучения, созданная Amazon, как оказалось, проявляла «предвзятость» в отношении женщин в результате смещения[1];
- протоколы, полученные в результате отслеживания эксперимента и управления версиями модели, пригодятся, если возникнут вопросы, касающиеся соответствия Генеральному регламенту о защите персональных данных (General Data Protection Regulation, GDPR);
- автоматизация обновлений модели высвободит время специалистов в области машинного обучения и искусственного интеллекта и повысит их удовлетворенность работой.

Когда следует подумать о конвейерах машинного обучения?

Конвейеры машинного обучения предоставляют множество преимуществ, но не каждый проект в области обработки данных, машинного обучения или искусственного интеллекта нуждается в автоматизации процессов при помощи конвейера. Иногда ученые, работающие с данными, просто хотят поэкспериментировать с новой моделью, испытать новую архитектуру модели или воспроизвести недавно опубликованный результат. Для подобных случаев конвейеры не принесут никакой пользы. Однако как только у модели появляются пользователи, например она становится частью пользовательского приложения, вам потребуется постоянно обновлять модель и выполнять ее тонкую настройку. В этих ситуациях мы возвращаемся к сценариям, которые обсуждали ранее, когда говорили о непрерывном обновлении моделей и уменьшении бремени этих задач, которое ложится на специалистов-исследователей в области машинного обучения и искусственного интеллекта.

Роль конвейеров возрастает по мере роста проекта машинного обучения. Если требования к набору данных или ресурсам велики, обсуждаемые нами подходы позволяют легко масштабировать инфраструктуру. Если важна повторяемость, это обеспечивается за счет автоматизации и ведения контрольного журнала конвейеров машинного обучения.

Обзор этапов конвейера машинного обучения

Работа конвейера машинного обучения начинается со сбора новых данных для обучения и заканчивается получением определенной обратной связи о том, как работает ваша недавно обученная модель. В качестве вариантов такой обратной связи может рассматриваться метрика достоверности модели или же отзывы пользователей вашего продукта. Конвейер включает в себя множество этапов, включая предварительную обработку данных, обучение и анализ мо-

[1] Статья Reuters от 9 октября 2018 г.

дели, а также ее развертывание. Вы можете себе представить, что выполнение этих шагов вручную чрезвычайно обременительно, и вероятность допустить ошибку при таком сценарии очень велика. В этой книге мы представим инструменты и решения для автоматизации жизненного цикла вашей модели.

Как вы можете видеть на рис. 1.1, жизненный цикл модели фактически является циклическим процессом. Данные могут собираться непрерывно, и поэтому модели машинного обучения могут обновляться. Большее количество данных обычно означает улучшение модели. И из-за этого постоянного притока данных именно автоматизация является ключевым фактором. В реальных приложениях вы часто хотите повторно обучать свои модели. Если это ручной процесс, когда необходимо вручную проверить новые данные обучения или проанализировать обновленные модели, у специалиста в области машинного обучения или работающего с данными инженера просто не останется времени для разработки новых моделей для новых бизнес-задач.

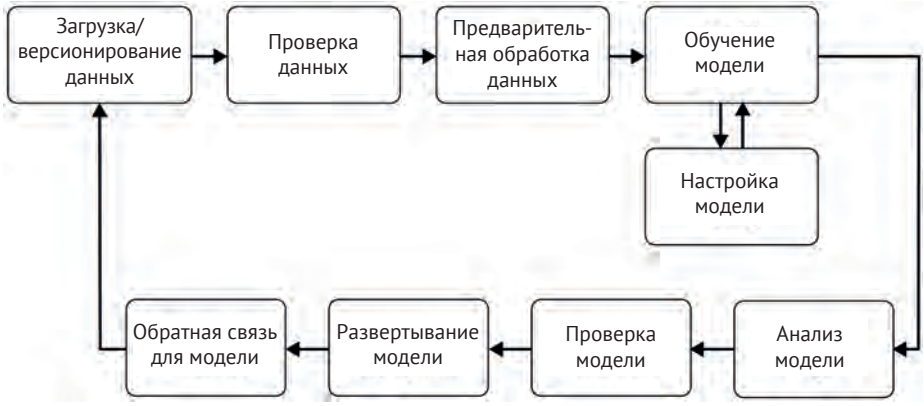

Рис. 1.1. Жизненный цикл модели

Жизненный цикл модели обычно включает в себя следующие этапы.

Этап загрузки данных и управление версиями данных

Загрузка данных – это начальный этап каждого жизненного цикла модели. На этом этапе работы конвейера мы обрабатываем данные, приводя их к формату, который могут воспринимать и обрабатывать следующие этапы. На этапе загрузки данных не выполняется никакое конструирование признаков (это происходит после этапа проверки данных). Это также хороший момент для создания новой версии поступающих данных, чтобы связать снимок данных с обученной моделью в конце рабочего процесса конвейера.

Проверка данных

Перед обучением новой версии модели нам нужно проверить новые данные. При проверке данных (о которой рассказывается в главе 4) мы фокусируемся на проверке статистических характеристик новых данных (таких как диапазон, количество категорий и распределение категорий), убеждаясь, что эти характеристики соответствуют ожиданиям, и предупреждаем специалиста, работа-

ющего с данными, в случае если обнаружены какие-либо аномалии. Например, если вы обучаете модель бинарной классификации, ваши данные, используемые для обучения, могут содержать 50 % выборок класса X и 50 % выборок класса Y. Инструменты проверки данных выдают предупреждения в случае, если пропорция в распределении выборок между этими классами изменяется, например когда вновь собранные данные распределяются между двумя классами в соотношении 70/30. Если модель обучается на таком несбалансированном обучающем наборе и разработчик не скорректировал функцию потерь модели или избыточно/недостаточно представленную в выборке категорию X или Y, прогнозы модели могут быть смещены в сторону преобладающей категории.

Стандартные инструменты проверки данных также позволят вам сравнивать различные наборы данных. Если у вас есть набор данных с преобладающей меткой и вы разбили исходные данные на обучающий и контрольный наборы, вы должны убедиться, что распределение меток между двумя этими наборами данных примерно одинаково. Инструменты проверки данных позволят вам сравнить наборы данных и выделить отклонения.

Если при проверке обнаруживается что-то необычное, жизненный цикл может быть остановлен на данном этапе, и разработчик может получить предупреждение о такой ситуации. Если обнаружено смещение выборки данных, ученый или инженер, специализирующийся на машинном обучении, может либо изменить критерии отбора отдельных выборок меток (например, выбрать только такое же количество выборок меток), либо изменить функцию потерь модели, запустить новый цикл конвейера для построения модели и перезапустить жизненный цикл модели.

Предварительная обработка данных

Вряд ли вы сможете использовать недавно собранные данные без обработки и напрямую обучать на них свою модель машинного обучения. Почти во всех случаях вам необходимо предварительно обработать данные, чтобы использовать их для обучения. Метки часто необходимо преобразовать в унитарные или многофакторные векторы[1]. То же самое относится к входным данным модели. Если вы обучаете модель на текстовых данных, вам нужно преобразовать текстовые символы в индексы или текстовые маркеры в векторы слов. Поскольку предварительная обработка требуется только перед обучением модели, а не для каждого цикла обучения, имеет смысл выполнять предварительную обработку как отдельный этап жизненного цикла, перед обучением модели.

Для обработки данных существует множество различных инструментов. Список возможных вариантов решений поистине бесконечен – от простого сценария Python до сложных графовых моделей. Хотя большинство специалистов по работе с данными предпочитают использовать функции обработки данных, встроенные в любимые инструменты, также важно, чтобы изменения этапов предварительной обработки можно было связать с обработанными

[1] В задачах классификации с обучением, использующих несколько классов в качестве выходных данных, часто необходимо преобразовать категорию в вектор, например в унитарный вектор (0,1,0), или из списка категорий в вектор, например в многофакторный вектор (1,1,0).

данными, и наоборот. Это означает, что если кто-то изменяет этап обработки (например, вводит дополнительную метку в преобразование унитарного вектора), предыдущие обучающие данные должны стать недействительными, что, в свою очередь, приведет к перезапуску рабочего цикла конвейера машинного обучения. Об этом этапе конвейера рассказывается в главе 5.

Обучение и настройка модели

Этап обучения модели (см. главу 6) является основным этапом в работе конвейера машинного обучения. На этом этапе мы обучаем модель принимать входные данные и прогнозировать выходные данные с наименьшей возможной ошибкой. Для больших моделей, и в особенности для больших обучающих выборок, этот этап может быстро стать малоуправляемым. Поскольку память, как правило, является конечным ресурсом при выполнении вычислений, при обучении модели решающую роль играет эффективное распределение ресурсов.

В последнее время большое внимание уделяется настройке модели, поскольку она может привести к значительному улучшению качества и обеспечить конкурентное преимущество. В зависимости от того, как устроен ваш проект машинного обучения, вы можете настроить свою модель, прежде чем начнете думать о конвейерах машинного обучения, или же вы можете настроить ее как часть своего конвейера. Поскольку наши конвейеры масштабируемы, благодаря их базовой архитектуре мы можем запускать большое количество моделей параллельно или последовательно. Это позволяет выбрать оптимальные гиперпараметры модели для нашей результирующей промышленной модели.

Анализ модели

Как правило, мы будем использовать такие параметры, как точность или потери, для определения оптимального набора параметров модели. Но как только мы определились с окончательной версией модели, крайне полезно провести более глубокий анализ качества этой модели. Для этого может потребоваться вычислить другие показатели, такие как точность, полнота отклика и площадь под кривой (Area Under Curve, AUC), или же вычисление точности модели на наборе данных большего размера, нежели контрольный набор, используемый при обучении.

Еще одна причина для углубленного анализа модели заключается в проверке правильности прогнозов модели. Невозможно сказать, как модель будет работать для разных групп пользователей, если набор данных не будет разделен на части, и точность модели не будет рассчитываться для каждого такого поднабора. Мы также можем исследовать зависимость модели от признаков, используемых в обучении, и исследовать, как изменится предсказание модели, если мы изменим признаки в одном из обучающих наборов.

Подобно этапу настройки модели и окончательному выбору наиболее эффективной модели, на этом этапе рабочего процесса для модели требуется проверка специалиста по машинному обучению и искусственному интеллекту. Тем не менее мы покажем, как может быть автоматизирован весь процесс анализа, чтобы участие человека требовалось лишь на заключительном этапе. Автоматизация будет поддерживать анализ моделей в согласованном состоянии, пригодном для сравнения с другими процедурами и результатами анализа.

Управление версиями модели

Цель этапа управления версиями и проверки модели состоит в том, чтобы отслеживать, какая модель, какой набор гиперпараметров и какие наборы данных были выбраны в качестве следующей версии модели.

В основанном на семантическом подходе управлении версиями в разработке ПО требуется увеличивать основной номер версии, когда вы вносите несовместимое изменение в свой API или добавляете новые функциональные возможности, внося существенные изменения. В противном случае вы бы увеличили дополнительный номер версии. Управление выпуском модели имеет еще одну степень свободы: набор данных. Существуют ситуации, когда вы можете достичь значительно отличающихся характеристик модели, не меняя определенный параметр модели или описание архитектуры, но предоставляя значительно больше данных для процесса обучения. Так что же, заметное повышение качества модели – повод для увеличения номера основной версии?

Хотя ответ на этот вопрос может быть разным для разных команд, работающих над проектами в области обработки данных и машинного обучения, важно документировать все входные данные для новой версии модели (гиперпараметры, наборы данных, архитектуру) и отслеживать их как часть этого этапа релиза.

Развертывание модели

После того как вы обучили, настроили и проанализировали свою модель, она готова к дальнейшему использованию. К сожалению, слишком много моделей развернуто как уникальные варианты реализации, и это делает процесс обновления моделей уязвимым местом.

Современные серверы моделей позволяют развертывать модели без разработки кода веб-приложений. Часто они предоставляют вам несколько интерфейсов API, таких как протоколы передачи представительного состояния (Representational State Transfer, REST) или удаленного вызова процедур (Remote Procedure Call, RPC), и позволяют одновременно размещать несколько версий одной и той же модели. Имея в своем распоряжении несколько версий одновременно, вы сможете выполнить A/B-тестирование на ваших моделях и получить ценную информацию об улучшениях вашей модели.

Серверы моделей также позволяют обновлять версию модели без повторного развертывания приложения, что сокращает время простоя вашего приложения и уменьшает необходимость в коммуникациях между разработчиками приложений и группами специалистов машинного обучения. Вопросы развертывания модели подробно обсуждаются в главах 8 и 9 этой книги.

Петли обратной связи

О последнем этапе жизненного цикла машинного обучения часто забывают, но он имеет решающее значение для успеха проектов в области науки о данных. Нам нужно замкнуть цикл и измерить эффективность и точность недавно развернутой модели.

На этом этапе мы можем собирать ценную информацию о качестве модели. В некоторых случаях мы можем собрать новые обучающие данные для увеличения наших наборов данных, чтобы обновить нашу модель. В петлях обратной связи может участвовать человек, либо этот этап может выполняться автоматически. Про петли обратной связи подробно рассказывается в главе 13.

За исключением двух шагов ручного просмотра, мы можем автоматизировать весь жизненный цикл. Специалисты по машинному обучению и искусственному интеллекту должны иметь возможность сосредоточиться на разработке новых моделей, а не на обновлении и поддержке существующих моделей.

Приватность данных

На момент написания этой книги вопросы приватности данных выходят за рамки стандартного жизненного цикла модели. Мы ожидаем, что это изменится в будущем, поскольку возрастает озабоченность пользователей в связи с использованием их данных и вводятся новые законы, ограничивающие использование персональных данных. Это приведет к интеграции методов сохранения приватности для машинного обучения в инструменты для построения конвейеров.

В главе 14 этой книги обсуждается несколько текущих вариантов повышения приватности в моделях машинного обучения:

- дифференцированная приватность, которая математически гарантирует, что предсказания модели не раскрывают данные пользователей;
- федеративное обучение, когда первичные данные не покидают пользовательское устройство;
- зашифрованное машинное обучение, когда либо весь процесс обучения может выполняться в зашифрованном пространстве, либо модель, обученная на первичных данных, может быть зашифрована.

Оркестровка конвейера

Все компоненты конвейера машинного обучения, описанные в предыдущем разделе, должны быть выполнены или, как мы говорим, оркестрованы, чтобы порядок их выполнения был строго определен и не нарушался. Входные данные для каждого компонента должны вычисляться до того, как этот компонент будет выполнен. Эти шаги выполняются при помощи таких инструментов, как Apache Beam, Apache Airflow (эти инструменты обсуждаются в главе 11) или Kubeflow Pipelines для инфраструктуры Kubernetes (обсуждается в главе 12).

В то время как инструменты конвейера данных координируют этапы конвейера машинного обучения, хранилища артефактов конвейера, такие как TensorFlow ML MetadataStore, сохраняют выходные данные отдельных процессов. В главе 2 мы представим обзор MetadataStore TFX и заглянем за кулисы TFX и его компонентов конвейера.

Для чего нужна оркестровка конвейера

В 2015 году группа инженеров машинного обучения в Google пришла к выводу, что одна из причин, по которой проекты машинного обучения терпят неудачу, заключается в том, что большинство проектов используют пользова-

тельский код для интеграции между отдельными этапами конвейера машинного обучения[1]. Однако этот нестандартный код нелегко перенести из одного проекта в другой. Исследователи обобщили свои выводы в статье «Скрытый технический долг в системах машинного обучения»[2]. В этой статье авторы утверждают, что связующий код между этапами конвейера часто бывает уязвим и что, за редкими исключениями, пользовательские сценарии не масштабируются. Со временем были разработаны такие инструменты, как Apache Beam, Apache Airflow или Kubeflow Pipelines. Эти инструменты можно использовать для управления задачами конвейера машинного обучения; они обеспечивают стандартизированную оркестровку и абстракцию связующего кода.

Хотя на первый взгляд изучение нового инструмента (такого как Apache Beam или Airflow), новой интегрированной среды (например, Kubeflow) и настройка дополнительной инфраструктуры машинного обучения (например, Kubernetes) могут показаться слишком сложными, эти вложения времени окупятся очень быстро. Не применяя стандартизированные конвейеры машинного обучения, команды, использующие технологии машинного обучения и искусственного интеллекта, неизбежно столкнутся с такими препятствиями, как уникальные настройки проекта, произвольное расположение файлов журнала, уникальные шаги отладки и т. д. Список сложностей, с которыми потенциально можно столкнуться при такой организации проектов, может быть поистине бесконечным.

Направленные ациклические графы

Инструменты конвейеризации, такие как Apache Beam, Apache Airflow и Kubeflow, управляют потоком задач, используя представление зависимостей задач в виде графов.

Пример графа, приведенный на рис. 1.2, иллюстрирует наличие направленности шагов конвейера. Работа конвейера начинается с задачи A и заканчивается задачей E. Это означает, что путь выполнения четко определен зависимостями задач. Направленные графы помогают избежать ситуаций, когда некоторые задачи запускаются в то время, когда еще не все зависимые от них задачи определены. Поскольку мы знаем, что нам необходимо предварительно обработать наши обучающие данные перед обучением модели, представление конвейера в виде направленного графа не позволяет выполнить задачу обучения до завершения шага предварительной обработки данных.

Графы рабочего процесса конвейера также должны быть ациклическими, то есть в графе должны отсутствовать связи задач с ранее выполненными задачами. Если бы это условие не выполнялось, это означало бы существование бесконечных циклов работы конвейера и, следовательно, невозможности завершить рабочий процесс.

---

[1] Google запустил внутренний проект под названием Sibyl в 2007 году для управления внутренним промышленным конвейером машинного обучения. Однако в 2015 году эта тема привлекла более широкое внимание, когда коллективом авторов была опубликована статья «Скрытый технический долг в системах машинного обучения» (D. Sculley et al. «Hidden Technical Debt in Machine Learning Systems», https://papers.nips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf).

[2] D. Sculley et al., «Hidden Technical Debt in Machine Learning Systems», Google, Inc. (2015).

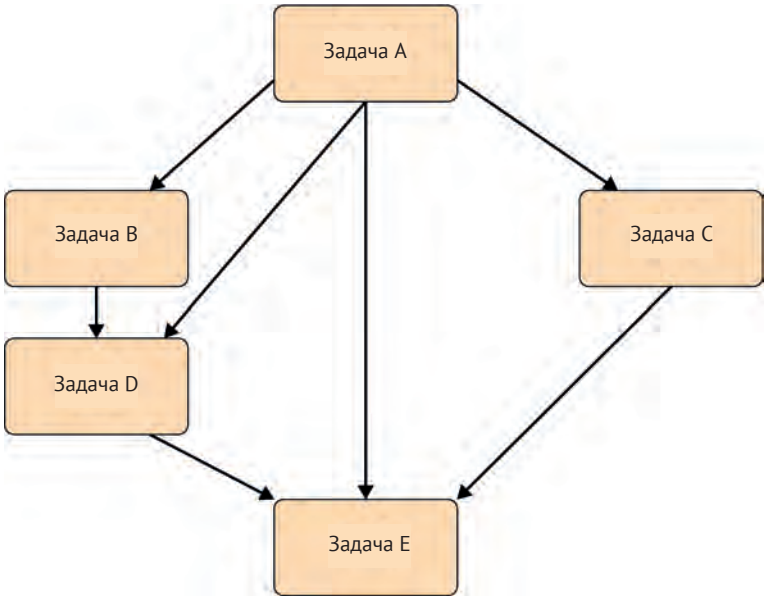

Рис. 1.2. Пример направленного ациклического графа

Из-за этих двух условий графы рабочего процесса конвейера являются направленными ациклическими графами (Directed Acyclic Graphs, DAGs). Далее вы сможете убедиться, что DAG – это основная концепция большинства инструментов рабочего процесса. Исполнение графов более подробно обсуждается в главах 11 и 12.

Наш демонстрационный проект машинного обучения

Чтобы пояснить содержимое глав этой книги на практических примерах, мы создали демонстрационный проект с использованием открытых данных и ПО с открытым исходным кодом. Используемый набор данных представляет собой жалобы потребителей на финансовые продукты в США и содержит комбинацию структурированных данных (категориальных/числовых данных) и неструктурированных данных (текста). Источником данных является Бюро по защите прав потребителей.

На рис. 1.3 показан фрагмент из этого набора данных.

	product	issue	consumer_complaint_narrative	company	state	company_response	timely_response	consumer_disputed
0	Mortgage	Loan servicing, payments, escrow account	My mortgage servicing provider (XXXX) transf...	SunTrust Banks, Inc.	TX	Closed with non-monetary relief	Yes	No
1	Debt collection	Cont'd attempts collect debt not owed	I HAVE NEVER RECEIVED ANY FORM OF NOTIFICATION...	ERC	CA	Closed with non-monetary relief	Yes	No
2	Debt collection	Disclosure verification of debt	I contacted walmart and the manager there said...	Synchrony Financial	MA	Closed with non-monetary relief	Yes	No
3	Credit reporting	Credit reporting company's investigation	I have filed multiple complaints XXXX on this ...	TransUnion Intermediate Holdings, Inc.	NY	Closed with explanation	Yes	Yes
4	Bank account or service	Account opening, closing, or management	Sofi has ignored my request to stop sending me...	Social Finance, Inc.	TX	Closed with explanation	Yes	No

Рис. 1.3. Образец данных

Задача машинного обучения заключается в том, чтобы, основываясь на данных о жалобе, предсказать, будет ли жалоба оспорена потребителем. В этом наборе данных оспаривается 30 % жалоб, поэтому набор данных не сбалансирован.

Структура проекта

Наш демонстрационный проект размещен в репозитории GitHub, и вы можете копировать его обычным способом, используя следующую команду:

```
$ git clone https://github.com/Building-ML-Pipelines/
          \ building-machine-learning-pipelines.git
```

Версии пакета Python

Для создания нашего демонстрационного проекта мы использовали Python 3.6–3.8. Мы использовали версии TensorFlow 2.2.0 и TFX 0.22.0. Мы сделаем все возможное, чтобы обновить репозиторий GitHub, загрузив в него последующие версии, однако мы не можем гарантировать, что проект будет работать с другими языками или версиями пакетов.

Структура нашего демонстрационного проекта:

- каталог *chapters*, который содержит блокноты для отдельных примеров из глав 3, 4, 7 и 14;
- каталог *components*, который содержит код для общих компонентов, таких как определение модели;
- завершенный интерактивный конвейер;
- пример эксперимента машинного обучения, который является отправной точкой для конвейера;
- законченные примеры конвейеров, оркестрованные с помощью Apache Beam, Apache Airflow и Kubeflow Pipelines;
- каталог *utility*, который содержит сценарий для загрузки данных.

В следующих главах мы проведем вас через шаги, необходимые для того, чтобы превратить пример эксперимента с машинным обучением (в нашем случае это блокнот Jupyter Notebook со структурой модели Keras) в законченный конвейер непрерывного машинного обучения.

Наша модель машинного обучения

Ядром нашего демонстрационного проекта глубокого обучения является модель, созданная функцией `get_model` в сценарии `components/module.py` нашего демонстрационного проекта. Модель предсказывает, оспорит ли потребитель жалобу, используя следующие признаки:

- финансовый продукт;
- подпродукт;
- ответ компании на жалобу;
- проблему, на которую пожаловался потребитель;
- штат США;
- почтовый индекс;
- текст жалобы (описание).

Для создания конвейера машинного обучения мы предполагаем, что проект архитектуры модели завершен, и мы не будем изменять модель. Мы обсудим архитектуру модели более подробно в главе 6; однако для этой книги архитектура модели – второстепенный вопрос. Здесь рассказывается о том, что вы можете делать с уже имеющейся у вас моделью.

Цель демонстрационного проекта

В ходе этой книги мы продемонстрируем необходимые структуры, компоненты и элементы инфраструктуры для непрерывного обучения нашей модели машинного обучения. Мы будем использовать стек, приведенный на схематической иллюстрации архитектуры на рис. 1.4.

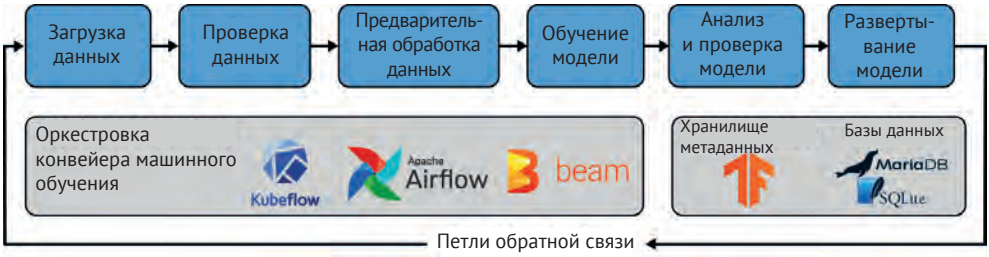

Рис. 1.4. Архитектура конвейера машинного обучения для нашего демонстрационного проекта

Мы попытались реализовать общую задачу машинного обучения, которую можно легко заменить вашей конкретной задачей машинного обучения. Структура и базовая настройка конвейера машинного обучения остаются прежними и могут быть перенесены на ваш вариант использования. Для каждого компонента потребуется выполнить некоторые настройки (например, указать, откуда брать данные), но, как мы увидим, объем этих настроек будет ограничен.

Резюме

В этой главе мы представили концепцию конвейеров машинного обучения и показали отдельные этапы рабочего процесса конвейера. Мы также продемонстрировали преимущества автоматизации этого процесса. Кроме того, мы подготовили стартовую площадку для следующих глав, приведя краткое описание содержания каждой главы, и представили наш демонстрационный проект. В следующей главе мы приступим к построению нашего конвейера!