

УДК 004.4
ББК 32.972
К62

Кольер Р., Монтонен К., Азарми Б.

К62 Машинное обучение в Elastic Stack / пер. с англ. В. С. Яценкова. – М.: ДМК Пресс, 2021. – 380 с.: ил.

ISBN 978-5-93700-107-8

В книге подробно рассматривается работа с Elastic Stack – обширной экосистемой компонентов, которые служат для сбора, поиска и обработки данных. Вы ознакомитесь с общими принципами машинного обучения, узнаете о методах автоматического обнаружения аномалий, проверке целостности и анализа данных из разрозненных источников, научитесь истолковывать результаты обнаружения и прогнозирования аномалий и использовать их в своих целях, а также выполнять анализ временных рядов для различных типов данных.

Издание адресовано специалистам, которые работают с данными и хотят интегрировать машинное обучение с эффективными приложениями для мониторинга, обеспечения безопасности и аналитики в области данных.

УДК 004.4
ББК 32.972

First published in the English language under the title 'Machine Learning with the Elastic Stack. Second Edition (978-1-80107-003-4). The Russian-Language 1st edition Copyright © 2021 by DMK Press Publishing under license by No Starch Press Inc. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-80107-003-4 (англ.)
ISBN 978-5-93700-107-8 (рус.)

© Packt Publishing, 2021
© Перевод, оформление, издание,
ДМК Пресс, 2021

Содержание

От издательства.....	12
Об авторах	13
О рецензентах.....	14
Предисловие.....	15
Часть I. ЗНАКОМСТВО С МАШИНЫМ ОБУЧЕНИЕМ И ELASTIC STACK	18
Глава 1. Машинное обучение в информационных технологиях.....	19
Преодоление исторических вызовов в ИТ.....	19
Что нам делать с потоком данных?.....	20
Причины появления автоматического обнаружения аномалий.....	21
Машинное обучение без учителя и с учителем.....	23
Использование машинного обучения без учителя для обнаружения аномалий.....	24
Что такое необычность?.....	24
Изучение того, что является нормой.....	26
Вероятностные модели.....	26
Обучение моделей.....	27
Выявление и устранение тенденций.....	30
Оценка степени необычности.....	31
Роль времени.....	32
Применение машинного обучения с учителем в аналитике фреймов данных.....	33
Процесс обучения с учителем.....	33
Заключение.....	35
Глава 2. Подготовка и использование Elastic ML.....	36
Технические требования.....	36
Включение функций Elastic ML.....	36
Включение машинного обучения в собственном кластере.....	37
Включение машинного обучения в облаке – Elasticsearch Service.....	39
Обзор операционализации Elastic ML.....	46
Узлы ML.....	46
Задания.....	47
Сегментирование данных в анализе временных рядов.....	48

Загрузка данных в Elastic ML	49
Служебные хранилища	51
.ml-config.....	51
.ml-state-*	51
.ml-notifications-*	52
.ml-annotations-*	52
.ml-stats-*	52
.ml-anomalies-*	52
Оркестровка обнаружения аномалий.....	52
Снимки модели обнаружения аномалий.....	53
Заключение	54

Часть II. АНАЛИЗ ВРЕМЕННЫХ РЯДОВ – ОБНАРУЖЕНИЕ И ПРОГНОЗИРОВАНИЕ АНОМАЛИЙ

Глава 3. Обнаружение аномалий	56
Технические требования.....	56
Типы заданий Elastic ML.....	56
Устройство детектора	58
Функция	59
Поле.....	59
Поле partition	59
Поле by	60
Поле over	60
Формула детектора.....	60
Обнаружение изменений частотности событий.....	61
Подробнее о функциях count	61
Другие функции подсчета	73
Ненулевой подсчет	73
Раздельный подсчет.....	74
Обнаружение изменений значений показателей.....	75
Метрические функции	76
min, max, mean, median и metric	76
varp.....	76
sum и non-null sum.....	76
Обзор расширенных функций детектора.....	77
Функция rare.....	78
Функция freq_rare	79
Функция info_content	79
Функции геолокации.....	79
Функции времени.....	80
Разделение анализа по категориальным признакам.....	80
Настройка поля разделения	80
Разница между разделением с использованием partition и by_field	82
Является ли двойное разделение пределом возможного?.....	83
Обзор временного и популяционного анализов.....	84

Категоризация в анализе неструктурированных сообщений.....	86
Типы сообщений, подходящие для категоризации.....	88
Предварительная категоризация.....	88
Анализ категорий.....	89
Пример задания по категоризации.....	90
Когда не следует использовать категоризацию.....	94
Управление Elastic ML через API.....	95
Заключение.....	97
Глава 4. Прогнозирование.....	98
Технические требования.....	98
Ключевое различие между предсказаниями и пророчествами.....	98
Для чего применяется прогнозирование?.....	100
Как работает прогнозирование?.....	100
Прогнозирование одиночного временного ряда.....	103
Просмотр результатов прогнозирования.....	114
Прогнозирование нескольких временных рядов.....	119
Заключение.....	122
Глава 5. Интерпретация результатов.....	123
Технические требования.....	123
Просмотр хранилища результатов Elastic ML.....	123
Оценка аномалий.....	128
Оценка на уровне сегмента.....	129
Нормализация.....	131
Оценка на уровне фактора влияния.....	131
Факторы влияния.....	133
Оценка на уровне записи.....	135
Описание схемы хранилища результатов.....	136
Результаты на уровне сегмента.....	137
Результаты на уровне записи.....	140
Результаты на уровне факторов влияния.....	143
Аномалии в нескольких сегментах.....	145
Пример аномалии в нескольких сегментах.....	145
Оценка аномалии в нескольких сегментах.....	146
Результаты прогноза.....	148
Запрос результатов прогноза.....	149
API результатов Elastic ML.....	151
Конечные точки API результатов.....	152
API обобщения сегментов.....	152
API категорий.....	153
Пользовательские панели мониторинга и рабочие панели Canvas.....	155
Панель инструментов встраивания.....	155
Аномалии как аннотации в TSVB.....	156
Настройка рабочих панелей Canvas.....	159
Заключение.....	162

Глава 6. Создание и использование оповещений	163
Технические требования.....	163
Определение и принцип работы оповещений.....	164
Аномалии не обязательно нуждаются в оповещениях.....	164
Точное время имеет значение.....	165
Создание оповещений из интерфейса машинного обучения.....	168
Определение заданий по обнаружению аномалий.....	168
Создание оповещений для пробных заданий.....	174
Моделирование аномального поведения в реальном времени.....	179
Получение и просмотр оповещений.....	180
Создание оповещений с помощью Watcher.....	183
Использование устаревшего варианта watch.....	183
trigger.....	184
input.....	184
condition.....	187
action.....	188
Пользовательские шаблоны watch с уникальной функциональностью.....	189
Связанный ввод и сценарий условий.....	189
Передача информации между связанными входами.....	190
Заключение.....	191
Глава 7. Выявление истинных причин аномалий	192
Технические требования.....	192
Настоящее значение термина AIOps.....	192
Значимость и ограничения KPI.....	194
Выходя за рамки KPI.....	197
Организация данных для анализа.....	198
Настраиваемые запросы для каналов данных.....	199
Дополнение получаемых данных.....	202
Использование контекстной информации.....	203
Аналитическое разделение.....	203
Статистические факторы влияния.....	204
Анализ первопричин аномалии.....	205
История проблемы.....	205
Корреляция и общие факторы влияния.....	207
Заключение.....	212
Глава 8. Другие приложения Elastic Stack для обнаружения аномалий	213
Технические требования.....	213
Обнаружение аномалий в Elastic APM.....	213
Включение обнаружения аномалий для APM.....	214
Просмотр результатов задания по обнаружению аномалий.....	219
Создание заданий машинного обучения с помощью распознавателя данных.....	222
Обнаружение аномалий в приложении Logs.....	224

Категории журналов.....	224
Журнал аномалий	225
Обнаружение аномалий в приложении Metrics	227
Обнаружение аномалий в приложении Uptime	230
Обнаружение аномалий в приложении Elastic Security	233
Готовые задания по обнаружению аномалий	233
Оповещения на основе заданий обнаружения аномалий	235
Заключение	237

Часть III. АНАЛИЗ ФРЕЙМОВ ДАННЫХ 238

Глава 9. Введение в анализ фреймов данных..... 239

Технические требования.....	240
Основы преобразования данных.....	240
Чем полезны преобразования?.....	240
Анатомия преобразований	241
Использование преобразований для анализа заказов интернет-магазина.....	242
Более сложные конфигурации сводной таблицы и агрегирования	246
Различие между пакетными и непрерывными преобразованиями.....	248
Анализ данных социальных сетей с помощью непрерывных преобразований	250
Использование Painless для расширенных конфигураций преобразования.....	253
Знакомство с Painless	254
Переменные, операторы и управление выполнением	255
Функции.....	260
Совместное использование Python и Elasticsearch	263
Краткий обзор клиентов Python Elasticsearch.....	264
Зачем нам нужен Eland?	266
Знакомство с Eland	267
Заключение	269
Дополнительная литература	270

Глава 10. Обнаружение выбросов 272

Технические требования.....	273
Принцип работы механизма обнаружения выбросов.....	273
Обзор четырех методов обнаружения выбросов	274
Методы, основанные на расстоянии	274
Методы, основанные на плотности	275
Оценка влияния характеристики.....	276
Как рассчитывается оценка влияния характеристик для каждой точки?.....	277
Чем обнаружение выбросов отличается от обнаружения аномалий?	278
Сравнение вероятностных моделей и экземпляров	278
Подсчет оценок.....	279

Характеристики данных.....	279
Потоковая и пакетная обработка.....	279
Применение обнаружения выбросов на практике.....	280
Оценка качества обнаружения выбросов с помощью API Evaluate.....	285
Настройка гиперпараметров для обнаружения выбросов.....	290
Заключение.....	293
Глава 11. Классификационный анализ.....	294
Технические требования.....	295
Классификация: от данных к обученной модели.....	295
Классифицирующие модели учатся на данных.....	296
Конструирование признаков.....	298
Оценка модели.....	299
Простой пример классификации.....	300
Деревья решений с градиентным усилением.....	307
Введение в деревья решений.....	308
Градиентное усиление.....	309
Гиперпараметры.....	309
Интерпретация результатов.....	313
Вероятность класса.....	314
Оценка класса.....	314
Важность признака.....	314
Заключение.....	316
Дополнительная литература.....	317
Глава 12. Регрессия.....	318
Технические требования.....	318
Использование регрессионного анализа для прогнозирования цен на жилье.....	319
Использование деревьев решений в регрессионном анализе.....	326
Заключение.....	329
Дополнительная литература.....	329
Глава 13. Логический вывод моделей.....	330
Технические требования.....	330
Поиск, импорт и экспорт обученных моделей с помощью API.....	331
Обзор API обученных моделей.....	331
Экспорт и импорт обученных моделей с помощью API и Python.....	333
Обработчики логического вывода и конвейеры данных.....	336
Обработка отсутствующих или поврежденных данных в конвейерах.....	345
Получение развернутой информации о прогнозах.....	347
Импорт внешних моделей с помощью eLand.....	348
Кратко о поддержке внешних моделей в eLand.....	349
Обучение DecisionTreeClassifier и импорт в Elasticsearch с помощью eLand.....	349
Заключение.....	353

Приложение. Советы по обнаружению аномалий	354
Технические требования.....	354
Роль факторов влияния в разделенных и неразделенных заданиях	354
Использование односторонних функций	361
Исключение определенных интервалов времени	363
Исключение наступающего (известного) интервала времени	364
Создание события календаря	364
Остановка и запуск потока данных в нужное время	365
Исключение интервала времени постфактум.....	366
Клонирование задания и повторный запуск исторических данных.....	366
Возврат задания к предыдущему снимку модели.....	366
Использование настраиваемых правил и фильтров	368
Создание собственных правил	369
Использование настраиваемых правил для оповещения «сверху вниз».....	370
Соображения относительно пропускной способности заданий.....	371
О вреде излишней сложности сценариев.....	372
Обнаружение аномалий в вычисляемых полях	373
Заключение	376
Предметный указатель	377

Об авторах

Рич Кольер (Rich Collier) – архитектор решений в Elastic. Он присоединился к команде Elastic после приобретения Prekert. Рич имеет более чем 20-летний опыт работы в качестве архитектора решений и системного инженера предпродажной подготовки программного обеспечения, оборудования и сервисных решений. Профессиональные интересы Рича включают аналитику больших данных, машинное обучение, обнаружение аномалий, обнаружение угроз, обеспечение безопасности, управление производительностью приложений, веб-приложения и технологии контакт-центров. Рич проживает в Бостоне, штат Массачусетс.

Камилла Монтонен (Camilla Montonen) – старший инженер по машинному обучению в Elastic.

Бахаалдин Азарми (Bahaaldine Azarmi), или коротко Баха, – архитектор решений в Elastic. До этого Баха был соучредителем ReachFive, платформы маркетинговых данных, ориентированной на поведение пользователей и социальную аналитику. Баха также сотрудничал с различными поставщиками программного обеспечения, такими как Talend и Oracle, где он занимал должности архитектора решений и системного архитектора. Баха является автором нескольких книг, в том числе *Learning Kibana 5.0*, *Scalable Big Data Architecture* и *Talend for Big Data*. Живет в Париже и имеет степень магистра компьютерных наук Парижского технологического института.

О рецензентах

Апурва Джоши (Apoorva Joshi) в настоящее время работает специалистом по безопасности данных в Elastic (ранее Elasticsearch), где она занимается применением машинного обучения для обнаружения вредоносных программ в конечных точках системы. До Elastic работала научным сотрудником в FireEye, где исследовала применение машинного обучения в задачах безопасности электронной почты. У нее разностороннее инженерное образование: бакалавр электротехники и магистр информационных технологий (с акцентом на машинное обучение).

Лицзюань Чжун (Lijuan Zhong) – опытный инженер в области технологий Elastic и облачных вычислений. У нее степень магистра информационных технологий и почти 20-летний опыт работы в сфере информационных технологий и телекоммуникаций. Сейчас сотрудничает с Netnordic – основным партнером Elastic в Швеции. Она начала свой путь в Elastic в 2019 году и стала сертифицированным инженером Elastic, также прошла курс машинного обучения Стэнфордского университета. Возглавляет множество образовательных программ и проектов, связанных с Elastic и машинным обучением, и клиенты всегда очень довольны результатом. Она была соорганизатором конференции Elastic Stockholm в 2020 г., приняла участие в конференции сообщества Elastic в 2021 г. и выступила с докладом о машинном обучении с помощью Elastic Stack. В 2021 году была удостоена бронзовой медали за вклад в развитие Elastic.

Предисловие

Elastic Stack, ранее известный как ELK Stack, представляет собой комплексное решение для анализа журналов, которое помогает пользователям эффективно получать, обрабатывать и анализировать данные поиска. Благодаря применению машинного обучения – ключевой особенности решения – Elastic Stack делает этот процесс еще более эффективным. Эта книга содержит всесторонний обзор функций машинного обучения Elastic Stack как для анализа данных временных рядов, так и для классификации, регрессии и обнаружения выбросов.

Знакомство с экосистемой Elastic Stack начинается с интуитивно понятного объяснения концепций машинного обучения. Затем под руководством авторов вы выполните анализ временных рядов для различных типов данных, таких как файлы журналов, сетевые потоки, показатели приложений и финансовые данные. По мере прочтения глав вы научитесь использовать машинное обучение в Elastic Stack для ведения журнала, обеспечения безопасности и отслеживания показателей. Наконец, вы узнаете, как анализ фреймворка открывает доступ к совершенно новым сценариям использования данных, в которых вам поможет машинное обучение.

После прочтения этой книги вы приобретете практический опыт совместного использования технологии машинного обучения и Elastic Stack, а также знания, необходимые для включения машинного обучения в вашу платформу распределенного поиска и анализа данных.

Для кого эта книга

Если вы профессионал в области данных и хотите получить представление о технологиях Elasticsearch, не прибегая к помощи специалиста по машинному обучению и не разрабатывая собственные решения, то эта книга про совместное применение машинного обучения и Elastic Stack для вас. Вы также найдете эту книгу полезной, если хотите интегрировать машинное обучение с вашими приложениями для мониторинга, обеспечения безопасности и аналитики. Чтобы извлечь из данной книги максимальную пользу, необходимо знать и уметь применять на практике Elastic Stack.

Какие темы охватывает эта книга

Глава 1 служит введением в тему и справочным пособием по историческим проблемам ручного анализа данных в IT и технологиях безопасности. В этой главе также представлен всесторонний обзор базовых принципов работы

машинного обучения Elastic (Elastic ML), чтобы читатель получил полное представление о том, что происходит «за кулисами».

В *главе 2* объясняется, каким образом применяются возможности машинного обучения в Elastic Stack, а также подробно описывается теория работы алгоритмов Elastic ML. Кроме того, в этой главе дается подробное объяснение логики операций машинного обучения применительно к Elastic.

В *главе 3* подробно рассматриваются методы автоматического обнаружения аномалий с обучением без учителя, которые лежат в основе анализа временных рядов.

В *главе 4* показано, что сложные модели временных рядов Elastic ML можно использовать не только для обнаружения аномалий. Возможности прогнозирования позволяют пользователям экстраполировать тенденции и поведение в будущем, чтобы помочь в решении таких задач, как планирование мощности.

В *главе 5* рассказано, как детально истолковать результаты обнаружения и прогнозирования аномалий и использовать их в своих целях в визуализации данных, информационных панелях и инфографике.

В *главе 6* рассмотрены различные методы интеграции возможностей упреждающего уведомления Elastic и данных, полученных с помощью машинного обучения, чтобы сделать обнаружение аномалий еще более эффективным.

В *главе 7* показано, как использование Elastic ML для проверки целостности и анализа данных из разрозненных источников в коррелированных представлениях дает аналитику преимущество с точки зрения наследования подходов.

В *главе 8* объясняется, как обнаружение аномалий используется другими приложениями в Elastic Stack для повышения эффективности анализа данных.

В *главе 9* рассмотрен анализ фрейма данных, его отличие от обнаружения аномалий временных рядов и рассказано, какие инструменты доступны пользователю для загрузки, подготовки, преобразования и анализа данных с помощью Elastic ML.

В *главе 10* описано применение анализа фреймов в сочетании с Elastic ML в аналитике данных.

В *главе 11* говорится о возможности классификационного анализа фреймов данных в сочетании с Elastic ML.

В *главе 12* рассмотрено использование регрессионного анализа фреймов данных в сочетании с Elastic ML.

Глава 13 описывает использование обученных моделей машинного обучения для логического вывода – прогнозирования выходных значений во время реальной работы системы.

Приложение включает в себя множество практических советов, которые отчасти выходят за рамки других глав. Эти полезные советы помогут вам максимально эффективно использовать Elastic ML.

КАК ПОЛУЧИТЬ МАКСИМАЛЬНУЮ ОТДАЧУ ОТ ЭТОЙ КНИГИ

Чтобы получить максимальную отдачу от этой книги, вам понадобится компьютер с хорошим подключением к интернету и учетной записью Elastic.

СКАЧИВАНИЕ ИСХОДНОГО КОДА ПРИМЕРОВ

Вы можете скачать файлы примеров кода для этой книги с GitHub по адресу <https://github.com/PacktPublishing/Machine-Learning-with-Elastic-Stack-Second-Edition>. Если выйдет обновление кода, оно появится в репозитории GitHub.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ И СОГЛАШЕНИЯ, ПРИНЯТЫЕ В КНИГЕ

В книге используются следующие типографские соглашения.

Курсив – используется для смыслового выделения важных положений, новых терминов, имен команд и утилит, а также слов и предложений на естественном языке.

Моноширинный шрифт – применяется для листингов программ, а также в обычном тексте для обозначения имен переменных, функций, типов, объектов, баз данных, переменных среды, операторов, ключевых слов и других программных конструкций и элементов исходного кода.

Моноширинный полужирный шрифт – используется для обозначения команд или фрагментов текста, которые пользователь должен ввести дословно без изменений, а также в листингах программ, если необходимо обратить особое внимание на фрагмент кода.

Моноширинный курсив – применяется для обозначения в исходном коде или в командах шаблонных меток-заполнителей, которые должны быть заменены соответствующими контексту реальными значениями.



Советы оформлены так.



Примечания оформлены так.



Важные примечания оформлены так.

Часть I

ЗНАКОМСТВО С МАШИНЫМ ОБУЧЕНИЕМ И ELASTIC STACK

В этой части представлено обобщенное описание Elastic ML – не только с точки зрения алгоритмов, но и с точки зрения организации работы программного обеспечения в Elastic Stack.

Эта часть книги состоит из следующих глав:

- главы 1 «Машинное обучение в информационных технологиях»;
- главы 2 «Подготовка и использование Elastic ML».

Глава 1

Машинное обучение в информационных технологиях

Десять лет назад идея использования технологий машинного обучения (ML) в IT-структурах или IT-безопасности казалась чем-то вроде научной фантастики. Однако сегодня это одно из самых популярных модных словечек, используемых поставщиками программного обеспечения. Очевидно, что за минувшее десятилетие произошел серьезный сдвиг как в осознании потребности в технологии ML, так и в возможностях, которые она предоставляет. Эта эволюция важна для понимания того, как появился инструмент Elastic ML и для решения каких проблем он был разработан.

Эта глава посвящена обзору истории и концепций, лежащих в основе работы Elastic ML. В ней также представлены различные виды анализа, которые можно провести, и задачи, которые можно решить с помощью Elastic ML. В частности, мы рассмотрим следующие темы:

- преодоление исторических вызовов в IT;
- что нам делать с потоком данных;
- причины появления автоматического обнаружения аномалий;
- машинное обучение без учителя и с учителем;
- использование машинного обучения без учителя для обнаружения аномалий;
- применение машинного обучения с учителем в аналитике фреймов данных.

ПРЕОДОЛЕНИЕ ИСТОРИЧЕСКИХ ВЫЗОВОВ В IT

К специалистам по поддержке IT-структур и архитекторам решений предъявляют высокие требования. Их роль не ограничивается внедрением новых передовых проектов и технологий для бизнеса; они также должны поддерживать безопасную и бесперебойную работу развернутых в настоящее время приложений. Сегодняшние приложения значительно сложнее, чем

когда-либо прежде, – они разбиты на множество компонентов, распределены и, возможно, виртуализированы/контейнеризированы. Они могут быть разработаны с использованием Agile-методики или аутсорсинговой командой. Вдобавок они, скорее всего, постоянно меняются. Некоторые команды DevOps заявляют, что обычно они вносят более 100 изменений в день в действующую производственную систему. Пытаться понять состояние и поведение современного приложения уровня предприятия – все равно что механику пытаться отремонтировать автомобиль, пока он едет по шоссе.

Аналитики по безопасности в области IT тоже с трудом справляются с повседневной работой, но, очевидно, у них другой фокус внимания – обеспечение безопасности предприятия и устранение возникающих угроз. Хакеры, вредоносные программы и инсайдеры-мошенники стали настолько распространенными и изощренными, что по мнению большинства специалистов по безопасности сегодня вопрос заключается не в том, будет ли взломана организация, а в том, насколько быстро она узнает об этом (если вообще узнает). Очевидно, что узнать о взломе как можно раньше (до того, как будет нанесен слишком большой ущерб) предпочтительнее, чем услышать об этом впервые от правоохранительных органов или из вечерних новостей.

Но что же нам делать? Возможно, проблема в том, что экспертам по приложениям и аналитикам службы безопасности не хватает данных, которые помогли бы им эффективно выполнять свою работу? На самом деле в большинстве случаев ситуация противоположная. Многие IT-специалисты и организации тонут в данных.

Что нам делать с потоком данных?

IT-отделы десятилетиями вкладывали силы и средства в инструменты мониторинга, и нередко в их распоряжении есть дюжина или более инструментов, активно собирающих и архивирующих данные, объем которых измеряется в терабайтах или даже петабайтах в день. Источники этих данных чрезвычайно вариативны – от элементарной статистики на уровне инфраструктуры и сети до результатов глубокой диагностики и/или файлов журналов системы и приложений.

Ключевые показатели эффективности (key performance indicators, KPI) на уровне бизнеса также можно отслеживать, иногда включая данные об опыте конечного пользователя. Глубина и широта охвата доступных данных сегодня больше, чем когда-либо. Для обнаружения возникающих проблем или угроз, скрытых в этих данных, традиционно использовалось несколько основных подходов к преобразованию «сырых» данных в информационные объекты:

- **Фильтрация/поиск:** некоторые инструменты предоставляют пользователю возможность фильтрации или поиска, чтобы сократить данные до более удобоваримого ограниченного набора. Хотя эта возможность чрезвычайно полезна, она чаще всего используется бессистемно, в основном когда возникает подозрение в наличии проблемы. Даже в этом

случае успех обычно зависит от способности пользователя понять, что он ищет, и от его уровня опыта – как от знания предыдущих ситуаций, так и от навыков использования самой технологии поиска;

- **визуализация:** панели мониторинга, диаграммы и виджеты также чрезвычайно полезны для понимания того, что происходит с данными, и выявления тенденций. Однако визуализации пассивны по своей сути и требуют постоянного наблюдения на предмет обнаружения значимых отклонений. Когда количество собираемых и отображаемых на экране показателей превышает возможности человеческого восприятия (или даже площадь экрана для их отображения), полезность визуализации резко снижается;
- **пороговые значения/правила:** чтобы обойти физические ограничения визуализации и внести в наблюдение за данными активный компонент (т. е. реакцию на изменение данных), многие инструменты позволяют пользователю определять правила или действия, которые запускаются при соблюдении определенных условий или возникновении определенных зависимостей между элементами данных. Однако маловероятно, что вы сможете реалистично определить все подходящие рабочие диапазоны или смоделировать все возможные зависимости в современных сложных и распределенных приложениях. Кроме того, количество и скорость изменений в приложении или среде могут быстро сделать любой статический набор правил бесполезным.

Аналитики обнаружили, что погрязли в ложных срабатываниях предупреждающих систем – широко известная проблема мальчика, который кричал о несуществующих волках, – что приводит к возмущению пользователей по поводу инструментов, генерирующих предупреждения, и скептицизму по поводу ценности, которой обладает такое предупреждение.

В конечном счете стало ясно, что нужен другой подход – такой, который не обязательно был бы полным отказом от прошлых методов, но мог бы обеспечить уровень автоматизации и значимого увеличения эмпирической ценности данных. Посмотрим правде в глаза: люди несовершенны – у нас есть скрытые предубеждения и ограничения способности запоминать информацию, и мы легко отвлекаемся и утомляемся. Алгоритмы машинного обучения при правильном использовании могут легко восполнить эти недостатки.

ПРИЧИНЫ ПОЯВЛЕНИЯ АВТОМАТИЧЕСКОГО ОБНАРУЖЕНИЯ АНОМАЛИЙ

Машинное обучение, будучи очень обширной темой, охватывающей многие области, от беспилотных автомобилей до компьютерных программ, приносящих выигрыш в играх, стало естественным кандидатом на роль эффективного решения. Если вы понимаете, что большинство задач мониторинга приложений или поиска угроз безопасности представляют собой всего лишь вариации на тему *поиска отличий от обычного хода событий*, тогда дисципли-

на обнаружения аномалий становится естественным местом для применения методов машинного обучения IT-специалистами.

Однако в науке об обнаружении аномалий, безусловно, нет ничего нового. Многие очень умные люди в течение долгих лет исследовали и применяли различные алгоритмы и методы. Но на практике обнаружение аномалий в IT-данных сопровождается некоторыми специфическими ограничениями, которые делают интересные с академической точки зрения алгоритмы непригодными для работы. Речь о следующих ограничениях:

- **своевременность.** Уведомление об отключении, нарушении или другой существенной аномальной ситуации должно стать известно как можно быстрее, чтобы смягчить последствия. Стоимость простоя или риск продолжения нарушения безопасности сводятся к минимуму, если быстро устранить проблему. Алгоритмы, которые не успевают отслеживать сегодняшние IT-данные в реальном времени, имеют ограниченную ценность;
- **масштабируемость.** Как упоминалось ранее, в современных IT-средах объем, скорость и вариативность IT-данных продолжают стремительно расти. Алгоритмы, которые занимаются мониторингом и анализом огромных данных, должны иметь возможность линейного масштабирования сообразно с данными, иначе спустя какое-то время они утрачат применимость;
- **эффективность.** Бюджеты IT-подразделений часто подвергаются тщательной проверке на предмет нерациональных расходов, и многие организации постоянно пытаются их урезать. Закупка дополнительного парка суперкомпьютеров для запуска неэффективных алгоритмов вряд ли будет утверждена руководством компании. Скорее, в качестве частичного решения придется использовать скромное стандартное оборудование с типичными характеристиками;
- **обобщаемость.** Хотя узкоспециализированная наука о данных часто является лучшим способом решения конкретной информационной проблемы, разнообразие данных в IT сформировало потребность в подходе, который применим в большинстве случаев. Повторное использование одних и тех же методов намного более рентабельно в долгосрочной перспективе;
- **адаптивность.** Постоянно меняющаяся IT-среда быстро делает жесткий алгоритм бесполезным. Обучение и переподготовка модели ML превращаются в бесконечное занятие и фактически в пустую трату времени, чего мы не можем себе позволить;
- **точность.** Мы уже говорили, что усталость от ложных предупреждений из-за устаревших пороговых и основанных на правилах систем является реальной проблемой. Замена одного генератора ложных тревог на другой никого не впечатлит;
- **простота использования.** Даже если все ранее упомянутые ограничения могут быть удовлетворены, любое решение, для реализации которого потребуются армия специалистов по данным, окажется слишком дорогостоящим и будет немедленно отвергнуто.

Итак, мы подошли к сути задачи – созданию быстрого, масштабируемого, точного и недорогого решения для обнаружения аномалий, которое все будут охотно использовать, потому что оно работает безупречно. Без проблем!

Как бы пугающе это ни звучало на самом деле, основатель и технический директор Prekert Стив Додсон принял этот вызов еще в 2010 году. Хотя Додсон, несомненно, заложил в фундамент компании свои академические знания, технология, которая в конечном итоге превратилась в Elastic ML, зародилась в муках попыток устранения реальных сбоев приложений уровня предприятия. Первая из них – досадный периодический сбой в работе торговой платформы в крупной лондонской финансовой компании. Додсон и несколько инженеров, присоединившихся к этому предприятию, помогли команде банка использовать технологию обнаружения аномалий для автоматического поиска «иголки в стоге сена», что позволило аналитикам сосредоточиться на небольшом наборе соответствующих показателей и записей в журналах событий, которые вызывали подозрения. Выявление первопричины (отказ службы, восстановление которой вызывало каскадный сбой других служб и причиняло ущерб) в конечном итоге обеспечило стабильную работу приложения и избавило банк от необходимости тратить много денег на другое решение – дорогостоящее обновление сетевой инфраструктуры.

Однако со временем стало ясно, что даже этот показательный успех был только началом. Спустя несколько лет и несколько тысяч примеров использования в реальном мире возник союз Prekert и Elastic – сочетание платформы, делающей большие данные легкодоступными, и технологий, которые помогли преодолеть ограничения традиционных методов анализа.

Перенесемся в 2021 год, спустя полные 5 лет после объединения усилий, когда Elastic ML прошел долгий путь в развитии и расширении возможностей платформы ML. Это второе издание книги включает в себя обновления, внесенные в Elastic ML за прошедшие годы, в том числе интеграцию с некоторыми решениями Elastic, касающимися наблюдаемости и безопасности. Во второе издание мы добавили введение в аналитику фреймов данных, которая подробно обсуждается в третьей части книги. Чтобы получить обновленное и глубокое понимание того, как работает Elastic ML, нам сначала нужно рассмотреть терминологию и идеи, а потом двигаться дальше.

МАШИННОЕ ОБУЧЕНИЕ БЕЗ УЧИТЕЛЯ И С УЧИТЕЛЕМ

Хотя существует множество подтипов машинного обучения, два самых известных (и относящихся к Elastic ML) – это *обучение без учителя* и *обучение с учителем*.

В машинном обучении без учителя нет внешнего руководства или указаний со стороны людей. Другими словами, алгоритмы должны изучать (и моделировать) шаблоны данных исключительно самостоятельно. В целом самая большая проблема здесь состоит в том, чтобы алгоритмы точно выявляли отклонения от нормальных шаблонов входных данных, дабы обеспечить

вывод модели, значимый для пользователя. Если алгоритм не может этого сделать, то он бесполезен и непригоден для использования. Следовательно, алгоритмы должны быть достаточно надежными и способны учитывать все тонкости поведения входных данных.

В машинном обучении с учителем для моделирования желаемого результата используются размеченные входные данные (часто многомерные). Ключевое отличие от машинного обучения без учителя состоит в том, что человек априори решает, какие переменные использовать в качестве входных данных, а также предоставляет «достоверные» примеры ожидаемой целевой переменной – *обучающие данные*. Затем алгоритмы машинного обучения изучают, как входные переменные взаимодействуют и влияют на известную выходную цель. Чтобы точно получить желаемый результат (например, прогноз), алгоритм должен иметь набор «правильных данных», которые не только отражают зависимость выхода от входа, но и достаточно разнообразны, чтобы модель смогла изучить максимально широкий спектр сочетаний переменных на входе.

Таким образом, в обоих случаях требуются качественные входные данные, хорошие алгоритмические подходы и хороший механизм, позволяющий ML как изучать поведение данных, так и применять это обучение для оценки последующих наблюдений за этими данными. Давайте посмотрим, как Elastic ML использует машинное обучение без учителя и с учителем.

ИСПОЛЬЗОВАНИЕ МАШИННОГО ОБУЧЕНИЯ БЕЗ УЧИТЕЛЯ ДЛЯ ОБНАРУЖЕНИЯ АНОМАЛИЙ

Чтобы сформировать у вас более полное представление о том, как работает обнаружение аномалий Elastic ML с использованием машинного обучения без учителя, мы рассмотрим следующие темы:

- строгое определение необычности в контексте технологии;
- наглядный пример обучения без учителя;
- описание того, как технология ML моделирует, устраняет тенденции и оценивает данные.

Что такое необычность?

Обнаружение аномалий – это то, чем мы регулярно занимаемся в повседневной жизни, поэтому имеем интуитивное представление о сути процесса. Люди довольно хорошо работают с визуальной информацией, поэтому не удивительно, что если бы я спросил у 100 человек на улице, что необычного в графике на рис. 1.1, подавляющее большинство (включая далеких от техники людей) указало бы на всплеск линии.

Аналогично мы можем спросить у людей, что необычного на следующей фотографии (рис. 1.2).

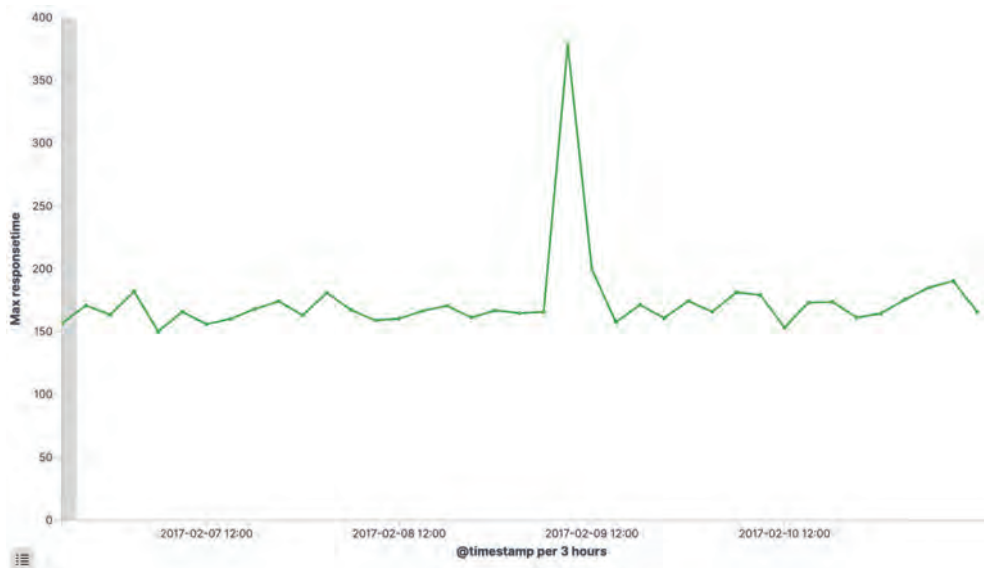


Рис. 1.1 ❖ Этот график содержит визуально заметную аномалию



Рис. 1.2 ❖ На этой фотографии запечатлен тюлень среди пингвинов

Большинство опрошенных наверняка ответят, что тюлень в окружении пингвинов – явление весьма необычное. Но людям бывает сложно сформулировать в явных терминах эвристику, которая лежит в основе таких выводов.

Есть две разные эвристики, которые мы могли бы использовать для определения различных видов аномалий, показанных на этих изображениях:

- сущность необычна, если ее поведение значительно отклоняется от установленного шаблона или диапазона, основанного на исторических данных;

- сущность необычна, если некоторые характеристики этой сущности значительно отличаются от тех же характеристик других членов группы или популяции.

Эти ключевые определения будут иметь прямое отношение к обнаружению аномалий Elastic ML, поскольку они образуют два основных фундаментальных режима работы алгоритмов обнаружения аномалий (временной и популяционный анализ, о которых пойдет речь в главе 3). Как вы увидите, пользователь будет решать, какой режим работы использовать для определенной задачи.

Изучение того, что является нормой

Как мы уже говорили, механизм обнаружения аномалий в Elastic ML основан на обучении без учителя, так как обучение происходит без привлечения специальных обучающих данных. Человек не помогает модели обучаться; она делает это самостоятельно, путем изучения представленных данных. Это немного похоже на изучение иностранного языка путем погружения, вместо того чтобы сидеть за словарями и учебниками грамматики.

Чтобы перейти от совершенно наивного состояния, когда о ситуации ничего неизвестно, к состоянию, в котором можно делать уверенные прогнозы, необходимо построить модель ситуации. Способ создания этой модели чрезвычайно важен, поскольку эффективность всех последующих действий, предпринятых на основе этой модели, будет сильно зависеть от точности модели. Модель должна быть гибкой и постоянно обновляться на основе новой информации, и это главное, что она должна делать в соответствии с парадигмой обучения без учителя.

Вероятностные модели

Вышеупомянутой цели вполне могут служить распределения вероятностей. Существует много фундаментальных типов распределений (в Elastic ML используются различные типы распределений, такие как пуассоновское, гауссово, логнормальное или даже сочетание моделей), но распределение Пуассона лучше всего подходит для обсуждения в первую очередь, потому что оно уместно в ситуациях, когда есть дискретные события («отсчеты») относительно времени (рис. 1.3).

Здесь показаны три различных варианта распределения, каждый с различным средним (λ) и максимальным ожидаемым значением k . Мы можем провести аналогию, которая гласит, что эти распределения моделируют ожидаемое количество почтовых отправлений, которые ежедневно приносят человеку домой, представленное знаком k на оси x :

- при $\lambda = 1$ вероятность того, что ежедневно будут доставлять ноль или одно почтовое отправление, составляет около 37 %. Возможно, это справедливо для студента колледжа, который не получает много писем;

- при $\lambda = 4$ вероятность получения трех или четырех писем составляет около 20 %. Это может быть хорошей моделью для молодого специалиста;
- при $\lambda = 10$ вероятность того, что в день будут доставлять 10 писем, составляет около 13 %, что может соответствовать молодой семье или домашнему хозяйству, которое каким-то образом оказалось во многих списках рассылки.

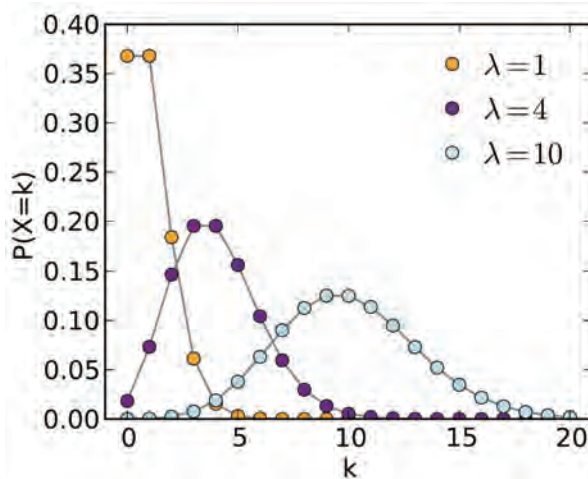


Рис. 1.3 ❖ График, демонстрирующий распределения Пуассона
(источник: https://en.wikipedia.org/wiki/Poisson_distribution#/media/File:Poisson_pmf.svg)

Дискретные точки на каждой кривой также дают вероятность других значений k . Поэтому модель может быть информативной и отвечать на такие вопросы, как «Вероятно ли получение 15 писем?». Как видите, это маловероятно для студента ($\lambda = 1$) или молодого специалиста ($\lambda = 4$), но более вероятно для большой семьи ($\lambda = 10$). В данном случае мы просто заявили, что представленные модели подходят для определенных категорий людей, — но совершенно очевидно, что в реальной жизни мы должны опираться на механизм обучения модели конкретным ситуациям, а не ограничиваться голословными утверждениями. Процесс обучения интуитивно понятен.

Обучение моделей

Если продолжить аналогию с доставкой писем, то становится ясно, что наилучшую модель для конкретного получателя можно определить путем ежедневного дежурства у почтового ящика и записи данных о том, сколько писем почтальон бросил в ящик. Также достаточно очевидно, что чем больше наблюдений сделано, тем выше будет ваша уверенность в точности модели. Другими словами, если провести у почтового ящика всего 3 дня, это даст менее полную информацию и уверенность, чем 30 дней или даже 300.

Фактически мы можем разработать алгоритм выбора соответствующей модели на основе наблюдений. Частью этого процесса самоотбора должна быть тщательная проверка выбора алгоритмом как самого типа модели (то есть распределение Пуассона, Гаусса, логнормальное и т. д.), так и конкретных коэффициентов этого типа модели (как в предыдущем примере λ). Для этого проводится постоянная оценка соответствия модели. Для оценки вероятных значений параметров модели по набору данных в целом также используются байесовские методы, но с учетом того, сколько информации было просмотрено до определенного момента времени. Алгоритмы машинного обучения делают это автоматически.

❗ Для тех, кто хочет более глубоко погрузиться в некоторые типичные математические процессы, происходящие за кулисами, рекомендуем научную статью по адресу <http://www.ijmlc.org/papers/398-LC018.pdf>.

Самое главное, что процесс уточнения модели является непрерывным, так что новая информация рассматривается вместе со старой, с экспоненциальным взвешиванием более свежей информации. Такая модель после 60 наблюдений могла бы напоминать график, представленный на рис. 1.4.

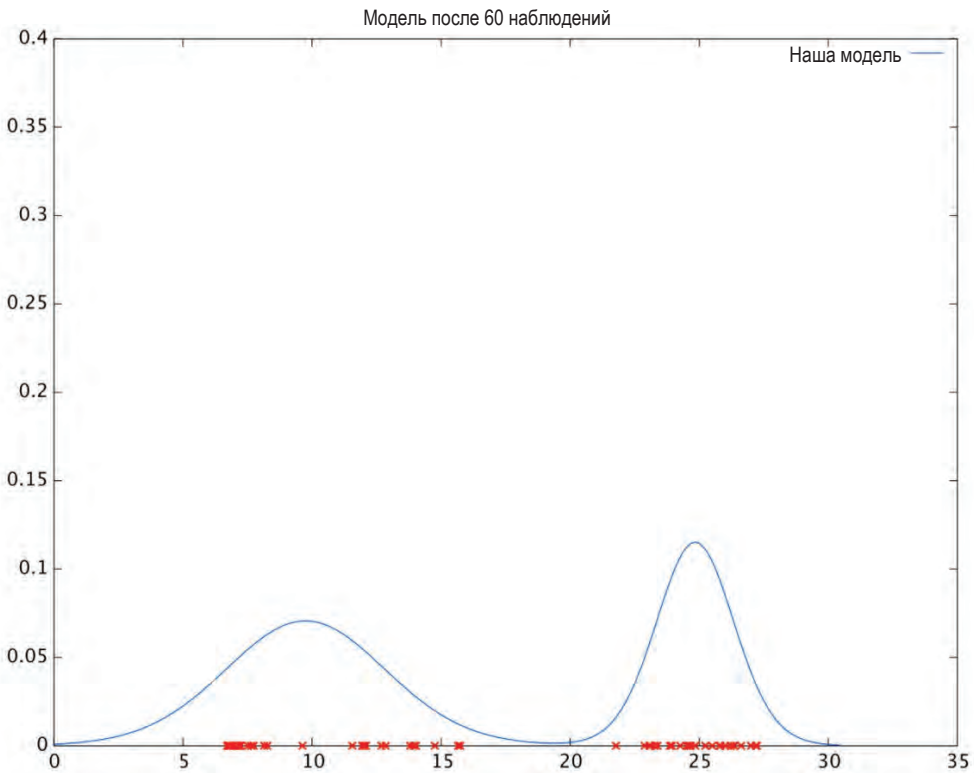


Рис. 1.4 ❖ Пример модели после 60 наблюдений

После 400 наблюдений модель будет выглядеть совсем иначе, поскольку данные представляют собой множество новых наблюдений со значениями от 5 до 10.

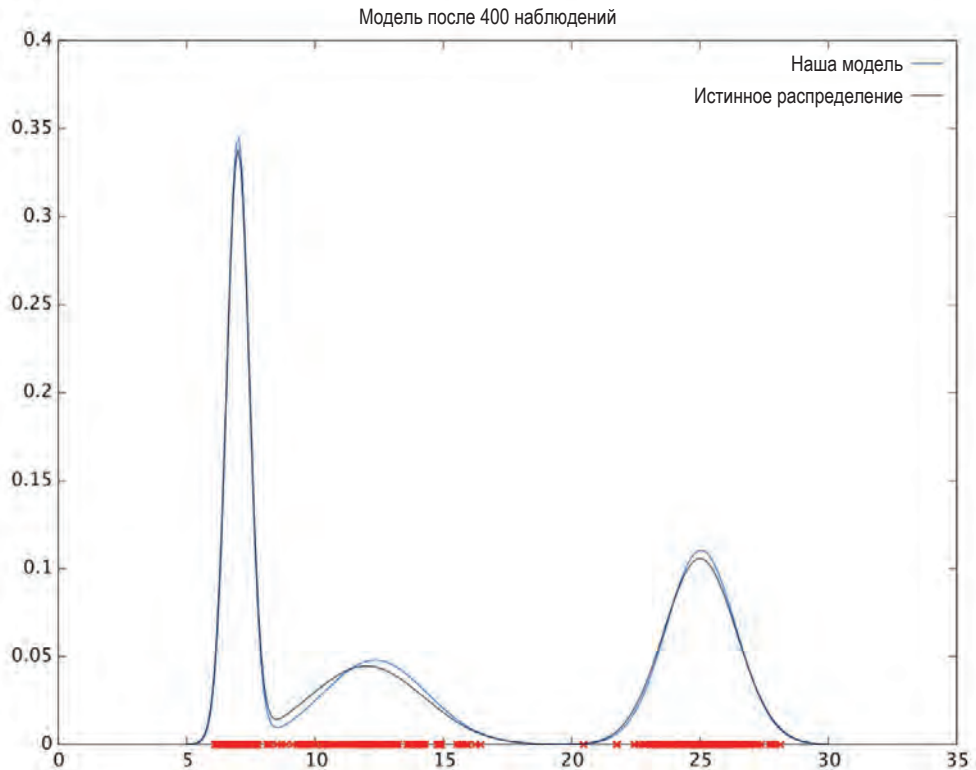


Рис. 1.5 ❖ Пример модели после 400 наблюдений

Также обратите внимание, что модель может иметь несколько модов, или областей/кластеров, с более высокой вероятностью. Сложность и точность совпадения изученной модели (показана синей кривой) с теоретической идеальной моделью (выделена черным цветом) имеет большое значение. Чем точнее модель, тем лучше представление состояния нормы для этого набора данных и, таким образом, в конечном итоге тем точнее прогноз того, как будущие значения совпадают с этой моделью.

Непрерывный характер моделирования также требует, чтобы эту модель можно было сериализовать и поместить в долгосрочное хранилище, а затем восстановить и возобновить обучение или использование позже. Как мы увидим, реализация этого процесса создания, хранения и использования модели представляет собой сложную оркестровку, которая, к счастью, автоматически выполняется средствами Elastic ML.

Выявление и устранение тенденций

Еще одним важным аспектом точного моделирования реальных данных является учет явных тенденций и скрытых закономерностей, которые возникают естественным образом. Например, наблюдаются ли у вас приливы и спады объема данных ежечасно и/или ежедневно при большей активности в рабочие часы или в рабочие дни? Если да, то это необходимо учитывать. Elastic ML автоматически выявляет основные тенденции в данных (линейный рост, периодические гармоника и т. д.) и учитывает их. Рассмотрим следующий график (рис. 1.6).

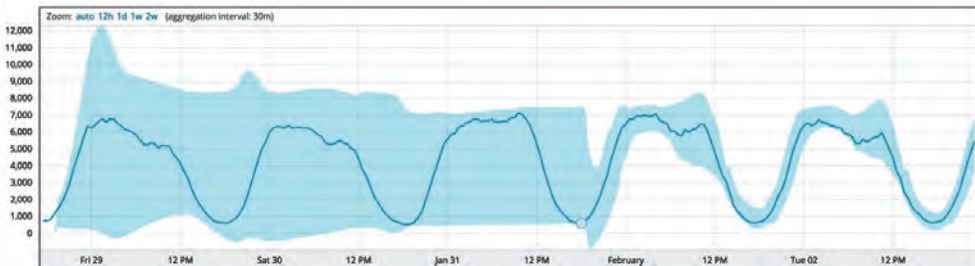


Рис. 1.6 ❖ Пример обнаружения периодичности

На этом графике показано, как мы изучаем, а затем выводим суточный цикл. Границы прогноза модели (представленные в виде голубой оболочки вокруг темно-синего сигнала) резко изменяются после автоматического обнаружения трех последовательных итераций этого цикла.

Таким образом, по мере того как с течением времени набирается больше данных, модели становятся более точными как с точки зрения более зрелой функции распределения вероятностей, так и за счет автоматического распознавания и устранения тенденций других стандартных шаблонов (например, рабочих дней, выходных и т. д.), которые могут не появляться в течение нескольких дней или недель. В следующем примере со временем обнаруживаются несколько тенденций, включая ежедневный и еженедельный цикл и общий линейный наклон (рис. 1.7).

Эти изменения модели записываются в виде системных аннотаций. Аннотации как общая концепция будут рассмотрены в следующих главах.

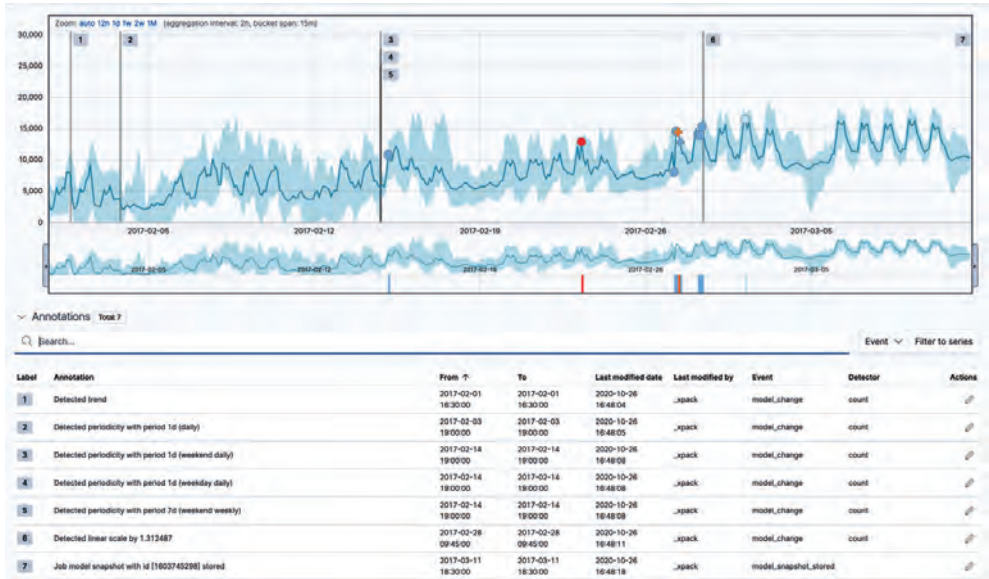


Рис. 1.7 ❖ Обнаружение нескольких тенденций

Оценка степени необычности

После построения модели вероятность любого будущего наблюдаемого значения может быть найдена в имеющемся распределении вероятностей. Ранее мы задавали вопрос: «Вероятно ли получение 15 писем?» На этот вопрос теперь можно дать эмпирический ответ в зависимости от модели, с числом от 0 (невозможно) до 1 (абсолютно достоверно). Elastic ML будет использовать модель для вычисления этого дробного значения с точностью примерно до 300 значащих цифр (что может быть полезно при работе с очень низкими вероятностями). Рассмотрим график, представленный на рис. 1.8.

В этом примере вероятность наблюдения значения 921 вычислена как $1,444e - 9$ (или, как чаще записывают, всего лишь 0,0000001444 %). Это очень маленькое значение, которое люди с трудом воспринимают и оценивают. ML будет использовать вычисленное значение вероятности и с помощью процесса квантильной нормализации отображать его на шкалу значимости от 0 до 100, где 100 – это наивысший уровень необычности, возможный для этого конкретного набора данных. В данном случае результат расчета вероятности $1,444e - 9$ нормализован до оценки 94. Эта нормализованная оценка пригодится позже в качестве средства для оценки серьезности аномалии в целях предупреждения и/или сортировки.

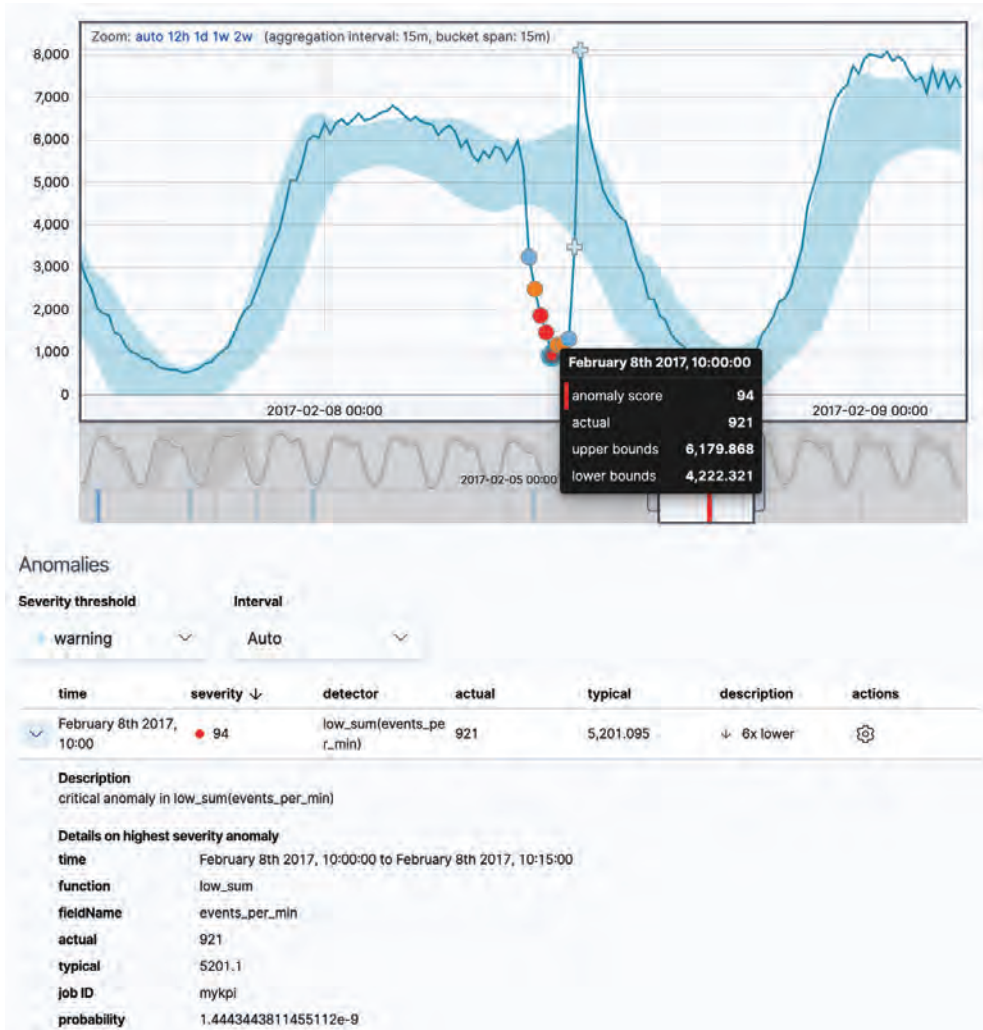


Рис. 1.8 ❖ Оценка аномалий

Роль времени

В Elastic ML все действия по обнаружению аномалий, которые мы будем обсуждать в оставшейся части книги, будут учитывать фактор времени, связанный с данными и анализом. Другими словами, при поиске аномалий Elastic ML полагает, что входные данные являются данными временных рядов, и они будут проанализированы с приращениями времени. Это ключевой момент, который помогает различать обнаружение аномалий и аналитику фреймов данных в дополнение к парадигме «обучение с учителем/без учителя».

Вы увидите, что есть небольшое, но важное различие между анализом совокупности (глава 3) и обнаружением выбросов (глава 10). Хотя оба они эффек-

тивно выявляют сущности, которые заметно отличаются от своих сородичей, анализ совокупности при обнаружении аномалий выполняется в рамках временного ряда, тогда как анализ с целью обнаружения выбросов не ограничен временем. Более подробно мы поговорим об этом в следующих главах.

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ С УЧИТЕЛЕМ В АНАЛИТИКЕ ФРЕЙМОВ ДАННЫХ

За исключением обнаружения выбросов (описанного в главе 10), которое основано на обучении без учителя, остальная часть аналитики фреймов данных использует обучение с учителем. В частности, есть два основных типа задач, которые позволяет решить аналитика фреймов данных Elastic ML:

- **регрессия.** Используется для прогнозирования непрерывного числового значения (цены, продолжительности, температуры и т. д.);
- **классификация.** Используется для прогнозирования того, относится ли что-то к определенному классу (например, является ли транзакция мошеннической и т. д.).

В обоих случаях модели строятся с использованием обучающих данных для сопоставления входных переменных (которые могут быть числовыми или категориальными) с выходными прогнозами посредством обучающих деревьев решений. Конкретная реализация, используемая Elastic ML, представляет собой особый вариант XGBoost, фреймворка дерева решений с градиентным усилением с открытым исходным кодом, который недавно приобрел определенную известность среди специалистов по данным благодаря своей способности выигрывать соревнования Kaggle.

Процесс обучения с учителем

Процесс машинного обучения с учителем значительно отличается от такового без учителя. При обучении с учителем вы четко отделяете этап обучения от этапа прогнозирования. Очень упрощенная версия процесса выглядит, как показано на рис. 1.9.

Здесь мы видим, что на этапе обучения признаки извлекаются из необработанных обучающих данных для создания *матрицы признаков* (также называемой *фреймом данных*) для передачи в алгоритм машинного обучения и создания модели. Модель может быть проверена на части данных (валидация модели), чтобы увидеть, насколько хорошо она работает, и могут быть предприняты последующие уточняющие шаги, чтобы скорректировать извлечение признаков или уточнить параметры алгоритма машинного обучения, используемого для повышения точности прогнозов модели.

Как только пользователь решает, что модель достаточно эффективна, эта модель «перемещается» в рабочий процесс прогнозирования, где она при-

меняется к новым данным. На основе модели поочередно *выводится* один новый вектор признаков, чтобы сформировать прогноз.

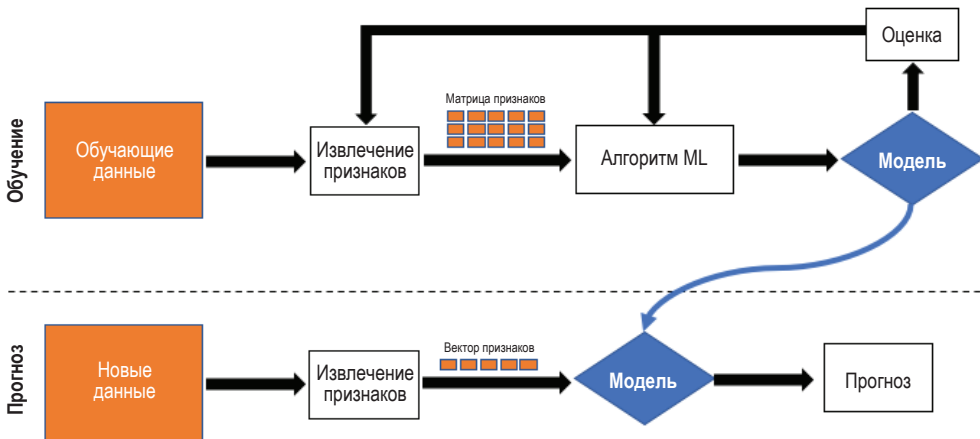


Рис. 1.9 ❖ Машинное обучение с учителем

Чтобы получить интуитивное представление о том, как это работает, представьте сценарий, в котором вы хотите продать свой дом, но не знаете, по какой цене его выставить. Вы изучаете предыдущие продажи в вашем районе и отмечаете для себя разницу в стоимости домов в зависимости от различных факторов (количество спален, количество ванных комнат, площадь в квадратных метрах, близость к школам/магазинам, возраст дома и т. д.). Эти факторы являются «признаками», которые учитываются вместе (а не по отдельности) для каждой предыдущей продажи.

Этот корпус исторических данных продаж составляет ваш набор обучающих данных. Он полезен, потому что вы точно знаете, сколько стоит каждый дом (и это параметр, значение которого вы в конечном итоге хотите спрогнозировать для своего дома). Если вы достаточно хорошо изучите этот набор, то можете интуитивно понять, что цены на дома сильно зависят от некоторых признаков (например, количества спален) и что другие признаки (возможно, возраст дома) не так сильно влияют на ценообразование. Это понятие, так называемую *важность признака* (feature importance), мы еще раз обсудим в следующей главе.

Обладая достаточным количеством обучающих данных, вы можете иметь хорошее представление о том, какова должна быть стоимость вашего дома, учитывая, что это дом 30-летней давности с тремя спальнями, двумя ванными комнатами и площадью 1700 квадратных футов. Другими словами, вы построили в уме модель на основе исследования сопоставимых домов, проданных за последний год или около того. Если прошлые продажи – это обучающие данные, то признаки вашего дома (спальни, ванные комнаты и т. д.) – это *векторы признаков* (feature vector), которые будут определять ожидаемую цену с учетом модели, которую вы «обучили» у себя в голове.

Ваша простая ментальная модель, очевидно, не настолько строгая, как та,

которую можно было бы построить на основе регрессионного анализа с использованием машинного обучения и принимая во внимание десятки соответствующих входных признаков, но эта простая аналогия должна сформировать у вас представление о процессе обучения на предыдущих, ранее известных ситуациях и затем применить это знание к настоящей, новой ситуации.

ЗАКЛЮЧЕНИЕ

В этой главе мы рассмотрели историю применения машинного обучения в прикладных задачах IT в связи с необходимостью автоматизировать анализ огромного и постоянно растущего объема данных в корпоративных средах. Вы получили более глубокое понимание различных типов машинного обучения, задействованных в Elastic ML, включая как обнаружение аномалий (обучение без учителя), так и анализ фреймов данных (обучение с учителем).

По мере продвижения по оставшимся главам мы будем часто сопоставлять варианты прикладных задач, которые пытаемся решить, с различными режимами работы Elastic ML.

Если ваши данные представлены временными рядами, т. е. они появляются регулярно с течением времени (данные показателей/производительности, файлы журналов, транзакции и т. д.), вполне возможно, что вы сможете обойтись только механизмом обнаружения аномалий Elastic ML. Как вы увидите дальше, он невероятно гибкий и простой в использовании и позволяет выполнять множество сценариев использования с широким спектром данных. Это что-то вроде швейцарского армейского ножа! Большая часть этой книги (главы с 3 по 8) будет посвящена тому, как использовать обнаружение аномалий и дополнительные возможности прогнозирования, чтобы получить максимальную отдачу от данных временных рядов, обработанных в Elastic Stack.

Если вас больше интересует поиск необычных объектов в популяции/когорте (необычное поведение пользователей/объектов), у вас может возникнуть непростой выбор между использованием анализа популяции для обнаружения аномалий или обнаружением выбросов в аналитике фрейма данных. Решение в первую очередь зависит от того, нужно ли вам делать это почти в реальном времени, – если да, то вы наверняка выберете анализ популяции. Если работа в реальном времени не требуется и/или если вам требуется одновременно рассматривать несколько признаков, вы должны предпочесть обнаружение выбросов. В главе 10 более подробно говорится о сравнении и преимуществах каждого подхода.

Остается много других вариантов использования, требующих многопланового подхода к моделированию. Кроме вышеупомянутого примера ценообразования на недвижимость, к ним относятся такие задачи, как распознавание языка, анализ оттока клиентов, обнаружения вредоносных программ и т. д. Все они относятся к области машинного обучения с учителем и аналитике фреймов данных и будут рассмотрены в главах с 11 по 13.

В следующей главе вы узнаете, как настроить Elastic ML и применять его на практике. Садитесь поудобнее и приготовьтесь наслаждаться новыми знаниями!