**7. ГЕНЕРАЦИЯ И ВЫБОР ПРИЗНАКОВ ДЛЯ ВРЕМЕННЫХ РЯДОВ**

Ранее мы рассмотрели методы анализа временных рядов, в ходе которого обучение модели выполняется с использованием всех точек данных временного ряда. В рамках подготовки к изучению методов анализа временных рядов с помощью технологий машинного обучения в следующей разделе нам предстоит познакомиться с принципами генерации и выбора признаков в временных рядах. Если вы еще не знакомы с подобными концепциями, то узнаете из этой части много нового и интересного. Это интуитивно понятный процесс, знаменующий творческий подход к анализу данных.

Генерация признаков — это процесс поиска количественного способа инкапсуляции наиболее важных характеристик данных временного ряда в виде нескольких числовых значений и категориальных подписей. С помощью набора признаков можно сжимать необработанные данные временного ряда в более короткое представление (небольшой пример приведен далее). Например, существует очень простой способ получения признаков — анализ временного ряда по среднему значению и количеству временных шагов в нем. Это один из способов описания временного ряда без пошагового прохода по всем необработанным данным.

Целью генерации признаков является сжатие как можно большего количества информации о полном временном ряде в нескольких показателях или, как вариант, использование этих показателей для определения наиболее важной информации о временном ряде и отбрасывания остальных факторов. Наибольшее распространение признаки получили в алгоритмах машинного обучения, большинство из которых разрабатывались для данных, не имеющих родственного отношения с временными рядами, но могут быть успешно применены к их обработке при условии представления временного ряда в виде правильно структурированных входных данных. В частности, речь пойдет о пакетах, предназначенных для автоматического генерирования часто используемых признаков временных рядов, что избавит нас от необходимости их изобретения или собственноручной реализации.

После получения в свое распоряжение предположительно актуальных признаков нужно убедиться в их полезности для модели. Хотя вам вряд ли удастся сгенерировать много бесполезных признаков вручную, рано или поздно вы все же столкнетесь с ними, особенно при автоматическом получении для временных рядов, используемых в машинном обучении, с помощью специального кода.

По этой причине всегда нужно проверять избранные признаки на пригодность, чтобы иметь возможность отказаться от их применения до проведения анализа данных.

При разработке модели машинного обучения не предполагалось применять для работы с временными рядами, а потому они плохо адаптируются к приложениям их автоматического исследования. Один из способов заставить такие модели работать с временными данными заключается в описании их признаками.

Например, представляя одномерный временной ряд не последовательностью чисел, задающей направление прохода по данным процесса, а набором признаков, к ним можно применить любые методы обработки перекрестных данных.

В этой части мы сначала рассмотрим очень простой пример генерации признаков для коротких временных рядов. Затем перейдем к изучению пакетов генерации признаков для временных рядов, представленных в языках R и Python. И наконец рассмотрим пример кода автоматической генерации и выбора признаков.

Изучив этот раздел, вы получите все необходимые познания о предварительной обработке данных временных рядов, востребованные при знакомстве с алгоритмами машинного обучения, описанными в следующем раздел.

**Вводный пример**

Представьте, что утренние, полуденные и вечерние температуры на прошлой неделе были такими, как показано в таблице. Внимательно изучив данные, можно увидеть элементы периодичности (ежедневный цикл), а также тенденцию к общему повышению температуры. Но в базе данных невозможно сохранить изображение графика, а большинство методов, принимающих изображения в качестве входных данных, очень ресурсоемкие и стремятся привести их к более сжатым метрикам. Таким образом, нам нужно самостоятельно определить сводные показатели. Вместо представления 21 числа из таблицы временным рядом данные можно описать всего несколькими словами и числами.

• Дневная периодичность

• Тенденция к росту (этот факт определяется количественным путем по наклону графика)

• Средние значения для каждого утра, полудня и вечера

Таблица 8.1. Температура на предыдущей неделе

|  |  |
| --- | --- |
| Время | Температура, °F |
| Полдень понедельника | 52 |
| Вечер понедельника | 15 |
| Утро вторника | 37 |
| Полдень вторника | 52 |
| Вечер вторника | 15 |
| Утро среды | 37 |
| Полдень среды | 54 |
| Вечер среды | 16 |
| Утро четверга | 39 |
| Полдень четверга | 51 |
| Вечер четверга | 12 |
| Утро пятницы | 41 |
| Полдень пятницы | 55 |
| Вечер пятницы | 20 |
| Утро субботы | 43 |
| Полдень субботы | 58 |
| Вечер субботы | 22 |
| Утро воскресенья | 46 |
| Полдень воскресенья | 61 |
| Вечер воскресенья | 35 |

Таким образом, мы свели 21-точечный временной ряд к нескольким числам — довольно сильное сжатие без потери детальных сведений. Это простейший случай генерации признаков. Последующий *подбор признаков* влечет за собой удаление любых признаков, недостаточно информативных для того, чтобы быть включенными в модель. Критерии выбора “правильных” признаков зависят от их дальнейшего назначения.

***Общие принципы выбора признаков***

Как и в случае с любыми другими аспектами анализа данных, при определении признаков для временных рядов некого набора данных необходимо помнить о конечной цели анализа, а также о том, что чрезмерные усилиях, прилагаемые для генерации как можно большего количества признаков, могут привести к переобучению модели, а не к осмысленным результатам.

Наилучшим подходом является разработка набора потенциально полезных признаков на этапах исследования и очистки временных рядов. В ходе визуализации и анализа данных вы научитесь наблюдать различия между временными рядами одного набора данных или разными временными периодами одного и того же временного ряда, постепенно вырабатывая концепции и метрики, на которых основывается разметка или прогнозирование временных рядов. Здесь вам станут полезными любые базовые знания о системе или сведения о рабочих гипотезах, которые требуется учитывать в последующем анализе.

В последующем материале вам предстоит познакомиться с концепциями, которые следует принимать во внимание при генерировании признаков временных рядов.

***Природа временного ряда***

Принимая решение о генерировании тех или иных временных рядов, необходимо помнить об их основных атрибутах, определенных на этапе исследования и очистки данных.

***Стационарность***

Стационарность является одним из важнейших факторов. Многие признаки временных рядов выбираются исходя из стационарности данных и оказываются бесполезными, если исходные временные ряды не оказываются стационарными или по крайней мере эргодическими. Например, использование среднего значения временного ряда в качестве признака будет разумным только тогда, когда оно имеет смысл, т.е. в случае стационарных числовых последовательностей. Среднее значение не имеет особого смысла при анализе нестационарных временных рядов, поскольку в их случае оно будет представлять результат обработки слишком большого числа запутанных процессов — трендов или сезонных циклов.

***Стационарные и эргодические временные ряды***

*Эргодический* временной ряд — это ряд, который одинаково представляет каждая (достаточно большая) его подвыборка. Это более слабое свойство, чем стационарность, требующая, чтобы подвыборки имели одинаковые среднее значение и дисперсию. Эргодичность предполагает “эквивалентность” по содержанию информации в каждом срезе временного ряда без сохранения одинаковых статистических параметров (среднее значение и дисперсия). Детальное описание этого понятия приведено на сайте StackExchange (https://perma.cc/5GW4-ZENE).

***Длина временного ряда***

Еще один фактор, который следует учитывать при генерации признаков, — длина временного ряда. Некоторые признаки прекрасно работают в стационарных временных рядах, но становятся нестабильными по мере увеличения их длины, например, минимальное и максимальное значения ряда. Для одного и того же базового процесса более длинный временной ряд, скорее всего, будет представлять более интенсивные максимальное и минимальное значения, чем те, которые наблюдаются в коротких временных рядах, созданных общим процессом, — только потому, что сбор его данных сопряжен с более широкими возможностями.

***Знания предметной области***

Знания предметной области должны выступать ключевым критерием при генерации признаков для временного ряда. Далее описано несколько примеров того, как знания о предметной области влияют на генерацию ключевых признаков временных рядов, а в этом разделе мы рассмотрим только общие концепции.

Например, обрабатывая временные ряды физических показателей, необходимо определить количество признаков, которые имеют смысл для временной шкалы изучаемой системы, а также убедиться, что на выбранные признаки не оказывают чрезмерного влияния неустранимые критерии, обусловленные, скажем, ошибками измерения датчиков, а не свойствами базовой системы.

Например, представьте, что нужно обработать данные некого финансового рынка. Чтобы обеспечить финансовую стабильность, для рынка устанавливается максимальное изменение цены для каждого отдельного дня. Если цена меняется слишком сильно, то рынок закрывается. Вам решать, нужно ли в такой системе использовать признак, описывающий максимальную цену за выбранный день.

***Независимые факторы***

При выборе признаков немаловажную роль играют вычислительные ресурсы и связанные с ними средства хранения данных. Кроме того, большое значение имеет преследуемая вами цель. Признаки генерируются для дальнейшего использования в модели, что позволяет отказаться от дальнейшего хранения большого объема исходных данных? Или же они рассчитываются только для однократного анализа, а исходные данные подлежат обязательному хранению? Цель, преследуемая при генерации признаков временных рядов, влияет на выбор их количества, а также на важность каждого из них. Она часто зависит от общего размера анализируемого набора данных. В небольшом наборе данных принимаемые решения обычно имеют незначительные последствия, но в случае очень крупных временных рядов задача расчета неправильно подобранных признаков часто сопряжена с напрасными ресурсными затратами —в первую очередь, заключающимися в использовании дорогого вычислительного оборудования и времени, потраченного на написание программного кода.

Таким образом, изучив все требующие учета факторы, попробуйте составить список признаков и протестировать их на небольшом наборе данных, чтобы понять, насколько быстро или медленно выполняется анализ данных. Если небольшой набор данных анализируется слишком медленно, то необходимо сократить временной ряд, прежде чем проводить полный анализ. Аналогичным образом, прежде чем приступать к анализу полного набора данных, рассмотрите возможность проверки вычислительно затратных признаков на подмножестве исходных данных.

***Источники признаков***

Процесс выбора признаков для данных временных рядов ограничен только воображением исследователей, их навыками в написании кода и познаниями предметной области. Для их получения нужно придумать достаточно общий, но в то же время точный вычислительный способ количественного описания поведения временных рядов. Некоторые простые часто используемые признаки временных рядов представляют сводные статистические показатели, используемые в других приложениях.

• Среднее значение и дисперсия

• Максимальное и минимальное значения

• Разница между последним и первым значениями

Кроме того, отдельные важные признаки легче определить визуально, чем вычислить.

• Количество локальных максимумов и минимумов

• Гладкость временного ряда

• Периодичность и автокорреляция временных рядов

В общих случаях необходимо разработать реализационные определения, включающие описания всех возможных способов идентификации часто используемых признаков. Это позволит создать собственную программную библиотеку кодов генерации признаков, но вы всегда можете обратиться к сторонним библиотекам генерации признаков для данных временных рядов, особенно при работе с более требовательным к вычислительным ресурсам методикам. Постарайтесь найти как можно лучше реализацию с надежным и эффективным кодом. Далее мы перейдем к рассмотрению библиотек получения признаков временных рядов, акцентируя внимание на извлечении широкого спектра признаков, генерируемых с помощью автоматизированных инструментов.

***Библиотеки генерации признаков временных* рядов с открытым исходным кодом**

Исследователями было предпринято много попыток автоматизации процесса генерации признаков временных рядов, широко востребованных в задачах, характерных для разных предметных областей и дисциплин.

***Модуль tsfresh в языке Python***

Одним из наиболее успешных инструментов автоматической генерации признаков в языке Python считается модуль tsfresh, обеспечивающий создание большого универсального набора признаков. Чтобы получить представление о количестве поддерживаемых им признаков (<https://perma.cc/2RCC-DJLR>), рассмотрим наиболее общие их категории.

***Описательные статистики***

Они основаны на традиционных методологиях статистического исследования временных рядов, которые изучались ранее, в том числе:

• оценочное значение расширенного критерия Дики-Фуллера;

• коэффициент AR(*k*);

• автокорреляция для запаздывания, *k.*

*Физические показатели нелинейности и сложности системы*

Эта категория включает в себя следующие функции.

• с3(). Через нее рассчитывается ожидаемое значение *L2(X2)L(X)X*(*L* — оператор запаздывания). Была предложена как мера нелинейности временного ряда.

• cid\_ce(). Вычисляет квадратный корень суммы значений от 0 до *п* - 2 х *запаздывание (х. - х*+1)2. Предложена в качестве меры сложности временного ряда.

• friedrich\_coefficients (). Возвращает коэффициенты модели, предназначенной для описания *сложного* нелинейного движения.

*Сжатие истории*

К этой категории относятся такие функции.

• Сумма значений во временном ряду, которые встречаются более одного раза.

• Длина самой длинной последовательной подпоследовательности значений,

которые больше или меньше среднего значения.

• Самое раннее появление во временном ряду минимального или максимального значения.

Модули, подобные tsfresh, призваны сэкономить рабочее время и выбрать наиболее подходящие реализации инструментов подбора признаков. Они также указывают на оптимальные способы описания данных, с которыми вам не приходилось встречаться ранее. Такие модули предоставляют много неоспоримых преимуществ при анализе данных с помощью надежных, хорошо проверенных инструментов с открытым исходным кодом.

• Не стоит изобретать новые методы при работе со стандартными признаками. Используя библиотеку общего назначения, можно быть уверенным в том, что другие пользователи будут подвергать сомнению только точность метода, но не код его реализации и используемые в нем инструменты.

• Подобные общедоступные пакеты являются не обычными библиотеками, а скорее фреймворками вычисления признаков. Например, tsfresh включает отдельный класс для вычисления признаков, который можно использовать для расширения возможностей библиотеки в собственных проектах с сохранением всех ее исходных преимуществ.

• Библиотека предназначена для взаимодействия с целевыми пользователями признаков, среди которых наиболее важным является пакет sklearn, позволяющий применять их в моделях машинного обучения.

Библиотека tsfresh имеет строгую техническую реализацию в том понимании, что многие полученные с ее помощью признаки вычисляются в результате анализа научных данных, собранных экспериментальным путем.

*Платформа анализа временных рядов cesium*

Не столь обширный, но при этом более доступный каталог автоматически генерируемых признаков создается с помощью библиотеки cesium (http://cesium-ml.org/docs/index.html). В настоящее время в документации упоминается следующий список поддерживаемых признаков (http: //cesium-ml. org/docs/feature\_table. html). Далее мы выберем из него и проверим несколько, как нам кажется, наиболее интересных вариантов. Несмотря на то что общие категории признаков детально рассмотрены в исходном коде (<https://perma.cc/8HX4-MXBU>), мы все же подробно изучим следующие из них.

• Признаки, которые описывают общее распределение значений данных без учета временных связей между ними. Эта категория включает самые разнообразные признаки, которые, тем не менее, не учитывают временной контекст данных.

— Сколько локальных пиков в гистограмме данных?

— Какой процент точек данных находится в фиксированном окне значений, близких к медиане данных?

• Признаки, которые описывают распределение данных во времени.

— Признаки, которые описывают распределение времени между измерениями и вычисляют статистику, аналогичную рассмотренной выше, но учитывающую распределение временных разностей, а не значений данных.

— Признаки, которые вычисляют вероятность того, что следующее наблюдение произойдет в течение *n* временных шагов с учетом наблюдаемого распределения.

• Признаки, которые измеряют периодичность поведения данных в пределах временного ряда. Часто эти признаки связаны с периодограммой Ломба-Скарлга.

***Периодограммы***

Периодограмма временного ряда — это оценка величины вклада различных базовых частот во временной ряд. Идеальная периодограмма позволяет преобразовать временной ряд из фазового пространства “время-значение”, с которым приходится работать чаще всего, в частотно-энергетическое пространство, в котором временной ряд рассматривается с точки зрения количества повторяющихся процессов, описываемых в различные моменты времени. Если вы не знакомы с такой концепцией, то ознакомьтесь с результатами ее визуализации, представленными на рисунке 7.1.

Рис. 7.1. Данные одной и той же временной области, в которой они выглядят как сумма синусоидального временного ряда. На оси частоты показано, что вся спектральная масса данных сосредоточена в окрестности трех частот, характеризующих синусоидальные процессы, порождающие данный временной ряд

Существует много способов расчета периодограмм. Один из основных применяемых методов — *преобразование Фурье* автокорреляционной функции. Более интуитивно понятный подход к созданию периодограммы заключается в аппроксимации исходного набора синусоидальных кривых различной частоты по методу наименьших квадратов.

Именно второй вариант лег в основу широко известного метода Ломба-Скаргла, который создает периодограмму по нерегулярной выборке временных рядов в отличие от традиционной периодограммы, которая основана на регулярной выборке временных рядов. Ломб, а затем Скаргл, разработали методы изучения нерегулярных временных рядов и установили, что статистические свойства периодограммы, построенной для таких данных, совпадают с таковыми в традиционной периодограмме, построенной по регулярным выборкам временных рядов. Достижения Ломба и Скаргла позволили решить многие задачи естествознания, неизбежно требующие анализа нерегулярных временных рядов, в таких исследовательских областях, как климатология, астрофизика и геология.

Упомянутые выше признаки могут быть вычислены либо для всего временного ряда, либо только для скользящих или расширяющихся окон. Исходя из полученных ранее знаний о механизмах выбора признаков для скользящих или расширяющихся окон, вы можете самостоятельно написать код реализации соответствующих вычислительных алгоритмов. У вас достаточно навыков, чтобы изучить документацию и разобраться в принципах работы библиотеки. В нашем примере функции скользящего окна используются для суммирования, а не очистки данных. Заметьте, что одни и те же методы анализа временных рядов применяются во многих различных, но одинаково полезных случаях.

Помимо функций генерации признаков, библиотека cesium включает много дополнительных инструментов. Например, она предоставляет веб-интерфейс для функций генерации признаков, а также команды интегрирования с инструментами пакета sklearn.

Протестировав все оговоренные пакеты на своих данных, можно заметить, что генерация признаков для временных рядов отнимает очень много времени. По этой причине необходимо заранее определиться с их количеством в анализируемых данных и также выяснить, в каких случаях автоматическая генерация признаков имеет смысл, а когда лучше отдать предпочтение ручному способу определения признаков.

Зачастую автоматизированное создание признаков с помощью программных библиотек сопряжено с большими вычислительными затратами и, учитывая их большое количество, такой подход не представляет ощутимых преимуществ. В каждой отдельной области знаний можно легко определить признаки, не представляющие интереса, определяемые шумом или не несущие информативной нагрузки. Не стоит тратить время и ресурсы на их вычисление — они совершенно бесполезны в последующем анализе и не добавят ясности в изучение данных.

Несомненно, пакеты автоматической генерации признаков имеют определенную практическую ценность, но должны использоваться разумно, а не во всех без разбора наборах данных.

***Пакет tsfeatures языка R***

Библиотека tsfeatures, разработанная Робом Хиндманом и его коллегами, представляет программный пакет для языка R, предназначенный для генерации большого количества часто используемых и удобных признаков временных рядов. В документации к пакету упомянут ряд полезных функций.

• Функции acf\_features () и pacf\_features (), каждая из которых вычисляет количество связанных значений с учетом общей важности роли автокорреляции в поведении ряда. Согласно документации функция acf\_features() возвращает следующие данные: “Вектор из 6 значений: первый коэффициент автокорреляции и сумма квадратов первых 10 коэффициентов автокорреляции исходного ряда, ряда первых разностей и ряда вторых разностей. Для сезонных данных также возвращается коэффициент автокорреляции для первого сезонного смещения”.

• Функции lumpiness() и stable(), указанные как определяемые окнами, и max\_level\_shift() и max\_var\_shift() — определяемые скользящими окнами. В каждой из вариантов к значениям, измеренным в перекрывающихся (скользящих) или непересекающихся (мозаичных) окнах временного ряда, рассчитываются разности и статистики измерения вариативности.

• Функции unitroot\_kpss() (<https://perma.cc/WF3Y-7MDJ>) и unitroot\_pp() (https://perma.cc/54XY-4HWJ).

В пакете tsfeatures реализованы наработки различных академических проектов по изучению признаков временных рядов. Он является результатом консолидации проектов по совершенствованию инструментов генерации признаков временных рядов, востребованных в различных областях знаний. В частности, пакет включает следующие функции.

• Функция compengine() вычисляет признаки временных рядов, разработанных в рамках проекта comp-engine.org, которые эффективно применяются для решения широкого спектра задач по анализу временных рядов во многих прикладных областях (если вы заинтересованы в более глубоком изучении рассматриваемых принципов, то обратите самое пристальное внимание на описание признаков набора временных рядов Catch22 (https://perma.cc/57AG-V8NP), которые находят широкое применение при решении самых разных задач, связанных с анализом и обработкой данных временных рядов. Сокращение набора признаков с более чем 4000 до 22 позволяет уменьшить время вычислений в 1000 раз — снижение точности классификации при этом составляет всего 7%. Также не лишне познакомиться с конвейером, который применялся для исследования данных и выбора в начальном наборе относительно независимых, но все же точных признаков).

• Ряд функций заимствован из пакета hсtsа, исходно отвечающего за сравнительный анализа временных рядов в Matlab. Приведем только некоторые из них: autocorr\_features(), firstmin\_ac(), pred\_features() и trev\_num (). Описание этих и других функций можно найти в документации. Документация к пакету tsfeatures (https: //perma.cc/Y8E9-9XCK) включает наглядные демонстрации с инструкциями по использованию и описанием результатов применения каждой функции генерации признаков, а также список специальной литературы, в которой вопросы статистического и машинного обучения, основанного на поддерживаемых пакетом признаках временного ряда, рассматриваются более детально.

***ПРИМЕРЫ ПРИЗНАКОВ, ХАРАКТЕРНЫХ ДЛЯ ПРЕДМЕТНЫХ ОБЛАСТЕЙ***

Другим источником вдохновения могут стать специфичные для выбранной предметной области признаки, определенные для различных временных рядов. Часто такие признаки вырабатывались в течение десятилетий и либо являются результатом опыта, даже если они не совсем понятны, либо основываются на научных знаниях о том, как работает система. Далее мы рассмотрим примеры признаков, свойственные временным рядам двух важных предметных областей: финансовой и медицинской.

***Технические индикаторы фондового рынка***

Технические индикаторы фондового рынка, вероятно, являются наиболее широко документированными и формализованными наборами показателей, используемых для анализа временных рядов в отдельных предметных областях. Специалисты по экономике, которые изучают данные временных рядов в течение последнего столетия, располагают обширным списком признаков, используемых для количественной оценки финансовых временных рядов исоставления биржевых прогнозов. Даже не будучи специалистом по финансам, вы будете поражены размером этого списка — он как нельзя лучше показывает, что анализ данных в отдельно взятой предметной области можно выполнять на основе довольно большого количества информативных и очень специфических признаков.

Ниже приведен сильно сокращенный вариант списка таких признаков, но даже он позволяет понять, насколько специфичны и важны указанные показатели для финансовой отрасли. Учитывая их высокую сложность, не удивительно, что многие специалисты по финансам потратили на их изучение всю свою карьеру, пытаясь понять, как по таким “индикаторам” можно предсказать рост и падение финансовых рынков.

*Индекс относительной силы (Relative Strength Index — RSI)*

Этот показатель вычисляется как 100 - 100/(1 + *RS),* где RS — это отношение средней прибыли за период роста цен к среднему убытку за период снижения цен. Входным параметром в таком анализе выступает период изучения роста и снижения цен, так что для разных периодов просмотра можно получить разные значения RSI. Трейдеры разработали эмпирические правила определения пороговых значений RSI, указывающих на недооценку или переоценку активов относительно реальной стоимости. RSI также известен как “индикатор импульса”, поскольку показывает направление движения финансового актива.

*Схождение/расхождение скользящих средних (Moving Average Convergence/Divergence — MACD)*

Этот индикатор основывается на трех временных рядах.

— Временной ряд MACD состоит из значений разности кратковременной (быстрой) и долгосрочной (медленной) экспоненциальными скользящими средними актива.

— Временной ряд “средних” представлен значениями экспоненциальных скользящих средних временного ряда MACD.

— Временной ряд “дивергенции” — это разность между временным рядом MACD и временным рядом “средних”. Это значение, которое обычно используется для составления финансовых прогнозов. Другие временные ряды признака (MACD и “средних”) обычно нужны только для получения временного ряда “дивергенции”.

*Денежный поток по Чайкину (Chaikin Money Flow — CMF)*

Этот индикатор измеряет направление изменения цены. Чтобы вычислить его, необходимо выполнить следующие операции.

— Рассчитать множитель денежного потока:

*((Close - Low) – (High - Close))/(High - Low).*

— Вычислить объем денежных средств, который представляет собой дневной объем торгов, умноженный на множитель денежного потока.

— Просуммировать этот объем денежных средств за определенное количество дней и разделить его на объем за то же количество дней. Этот индикатор является “осциллятором”, который задается в диапазоне значений от -1 до 1. Он указывает на “давление покупки” или “давление продажи”, т.е. показатель направления рынка.

Как можно видеть из приведенного выше краткого описания технических признаков, вычисляемых из простых наборов финансовых данных, существует много способов описания временных рядов. Финансовые рынки являются особенно богатым и хорошо изученным источником данных, который успешно применяется для генерации признаков (Обширный каталог признаков, которые можно использовать в методах машинного обучения, применяемых для анализа финансовых временных рядов, приведен в блоге Kaggle (https://perma.cc/Q84C-44XD). К сожалению, описанный в нем код не получил широкого признания, хотя и выступает отличным примером исчерпывающе правильного подхода к выбору релевантных признаков в рассматриваемой предметной области).

***Медицинские временные ряды***

Здравоохранение — это еще одна область, в которой признаки временных рядов имеют специфические для предметной области значения и даже названия. Как обсуждалось ранее, данные о состоянии здоровья выступают источником для огромного количества временных рядов. Один из наиболее выразительных примеров — данные электрокардиограмм (ЭКГ).



Рис. 7.2. Примеры признаков медицинских временных рядов, которые используются при чтении ЭКГ

Их чтение — это целая наука, в которой признаки устанавливаются врачами и в дальнейшем используются для чтения временных рядов. Если вы планируете подбирать признаки для алгоритмов машинного обучения по данным ЭКГ, то вам, безусловно, следует начать с изучения уже известных вариантов, с описанием и назначением которых вас познакомит любой медицинский сотрудник должной квалификации. Точно так же, анализируя временные ряды с данными высокого разрешения об уровне глюкозы в крови, важно научиться распознавать шаблоны, применяемые для представления ежедневных данных, а также принципы их описания и маркировки медицинскими работниками.

Рис. 7.3. Временной ряд ежедневных данных об уровне глюкозы в крови характеризуется четырьмя признаками, по которым медицинские работники судят о состоянии больного. Один из них не связан с режимом питания и известен под названием феномен утренней зари”. Он оказывает непосредственное влияние на остальные признаки.

Оба временных ряда, показанных на рисунках являются хорошими примерами данных, которые достаточно точно описываются величинами локальных максимумов или расстояний между ними. Следовательно, используя правильные инструменты библиотек tsfresh и cesium и учитывая особенности заданной предметной области, можно легко подобрать соответствующие признаки для целевых медицинских временных рядов.

**ОТБОР АВТОМАТИЧЕСКИ СГЕНЕРИРОВАННЫХ ПРИЗНАКОВ**

Предположим, вам удалось автоматически сгенерировать большое количество признаков, описывающих набор данных большого временного ряда. Скорее всего, вам не удастся оценить их все при первом же просмотре данных, поэтому всегда не лишне дополнить функцию автоматической *генерации* признаков возможностью их автоматического *отбора.* К одним из лучших инструментов отбора признаков относится алгоритм FRESH, который реализован в описанном ранее пакете tsfresh. Этот алгоритм выбирает релевантные признаки на основе масштабируемых критериев проверки гипотез. Необходимость в разработке алгоритма FRESH была вызвана постоянно увеличивающимся объемом данных временных рядов, зачастую подлежащих распределенному хранению с целью последующего распараллеливания вычислений.

Алгоритм оценивает значимость каждого входного признака по отношению к целевой переменной методом вычисления *p-значения* для каждого из них. После этого *p*-значения каждого признака оцениваются в совокупности с помощью *процедуры Бенджамини-Екутиели,* учитывающей такие входные параметры, как уровень допустимых погрешностей, и указывающей, какие из признаков следует сохранить для дальнейшего использования. Процедура *Бенджамини-Екутиели —* это метод ограничения количества ложных срабатываний, обнаруженных во время проверки гипотез, используемый для получения p-значений на начальном этапе алгоритма FRESH.

Самостоятельная реализация таких процедур — это довольно сложная задача, но воспользовавшись пакетом tsfresh, ее можно решить всего в несколько строк кода. В следующем примере использован шаблон кода, приведенный в документации к модулю. Вначале нужно загрузить данные временных рядов, относящиеся к сбоям в работе робота.

*## Python*

from tsfresh.examples.robot\_execution\_fallures import

download\_robot\_execution\_failures,

load\_robot\_execution\_failures

download\_robot\_execution\_failures()

timeseries, у = load\_robot\_execution\_failures()

Признаки извлекаются без дополнительного указания, поэтому пакет автоматически вычисляет их все. В этом смысле он поступает вразрез с предостережениями, приведенными ранее, оставаясь достаточно беспечным относительно экономии вычислительных ресурсов. В тестовом наборе данных не слишком много точек данных, но вы, вероятнее всего, откажетесь от полной их обработки, предварительно не сократив до разумного количества.

*## Python*

from tsfresh import extract features

extracted\_features = extract-features(timeseries,

column\_id = "id",

column\_sort = "time")

Хотя пакет tsfresh позволяет указывать вычисляемые признаки, в настоящем примере мы будем рассматривать все возможные варианты. Заметьте, что всегда можно указать параметры для тех признаков, которые должны приниматься в расчет при вычислении. Все возможности детально описаны и проиллюстрированы в документации.

Проведя полное извлечение для данных — в нашем случае включенных в пакет tsfresh — можно получить огромное количество признаков .

*## Python*

extracted\_features.columns

Index ([' F\_x\_abs\_energy' , 'F\_x\_absolute\_sum\_of\_changes',

'F\_x\_agg\_autocorrelation\_f\_agg\_"mean" ',

'F\_x\_agg\_autocorrelation\_f\_agg\_"median"•,

'F\_x\_agg\_autocorrelation\_f\_agg\_"var"’,

'F\_x\_agg\_1inear\_trend\_fagg\_"max"\_\_ chunk\_len\_l0\_attr\_

"intercept" ',

'F\_x\_agg\_1inear\_trend\_fagg\_"max"\_\_ chunk\_len\_l0\_\_attr\_

"rvalue" ',

'F\_x\_agg\_1inear\_trend\_fagg\_"max"\_\_ chunk\_len\_l0\_\_attr\_

"slope" ',

'F\_x\_agg\_1inear\_trend\_fagg\_"max"\_\_ chunk\_len\_l0\_\_attr\_

"stderr" ',

'F\_x\_agg\_1inear\_trend\_fagg\_"max"\_\_ chunk\_len\_l5\_\_attr\_

"intercept" ',

…

'T\_z\_\_ time\_reversal\_asymmetry\_statistic\_\_ lag\_1' ,

'T\_z\_\_ time\_reversal\_asymmetry\_statistic\_\_ lag\_2' ,

'T\_z\_\_ time\_reversal\_asymmetry\_statistic\_\_ lag\_3' ,

'T\_z\_\_ value\_count\_\_ value\_-inf', 'T\_z\_\_value\_count\_\_value\_0',

'T\_z\_\_ value\_count\_\_ value\_1', 'T\_z\_value\_count\_\_value\_inf',

'T\_z\_\_ value\_count\_\_ value\_nan', 'T\_z\_\_variance',

'T\_z\_\_ variance\_larger\_than\_standard\_deviation],

dtype=' object', name=' variable', length=4764)

Нами получено 4764 столбца значений. Это количество признаков гораздо больше того, которое можно вычислить вручную, но при работе с реальными наборами данных на их вычисление уйдет заметно больше времени. Решившись на развертывание настолько большого набора признаков, постарайтесь реалистично оценить как вычислительную мощность рабочего оборудования, так собственные способности и скорость анализа полученных результатов. Помните, что выбросы данных во временных рядах могут оказаться особенно сложнымии неудобными для последующего анализа. Обязательно убедитесь, что избранные вами признаки лишены их.

Несмотря на способность алгоритма FRESH учитывать зависимость между признаками, разобраться в принципах его работы зачастую весьма непросто. Следовательно, во многих случаях можно обратиться к более традиционному и простому для понимая методу — *рекурсивного удаления признаков* (Recursive Feature Elimination — RFE). Часто метод RFE применяется как дополнение алгоритма FRESH, позволяя упростить понимание различий между признаками, подобранными: и отвергнутыми алгоритмом FRESH.

Метод RFE основан на поэтапном подходе к выбору признаков, в рамках которого элементы постепенно исключаются из более полной модели, что приводит к постепенному ее упрощению до размера с минимальным набором элементов, количество которых определяется в самом начале процедуры выбора.

Этот метод известен как обратный отбор, потому что вы начинаете с наиболее сложной модели и возвращаетесь “обратно” к более простой модели. В противоположность ему в прямом отборе признаки добавляются постепенно, пока не будет получено максимальное возможное их количество или достигнуто предельное значение заранее оговоренного критерия остановки.

Метод RFE можно использовать как для выбора признаков, так и для оценки их важности. В следующем эксперименте объединим 10 случайно выбранных признаков из списка, предоставленного алгоритмом FRESH, с 10 случайно выбранными элементами из списка признаков, отклоненных алгоритмом FRESH.

*## R*

x\_idx = random, sample (range (len (features\_filtered. columns) ), 10)

selX = features\_filtered.iloc[:, x\_idx].values

unselected\_features = list (set(extracted\_features.columns)

.difference (set (features\_filtered. columns)))

unselected\_features = random.sample(unselected\_features, 10)

unsel\_x\_idx = [idx for (idx, val) in enumerate(

extracted\_features.columns) if val in unselected\_features]

unselX = extracted\_features.iloc[:, unsel\_x\_idx].values

mixed\_X = np.hstack( [selX, unselX])

Применим метод RFE к этому набору из 20 признаков, чтобы получить представление об их ранжировании (важности) для набора данных и модели, используемой в методе RFE.

>>svc = SVC(kernel="linear", С=1)

>>rfe = RFE(estimator=svc, n\_features\_to\_select=l, step=l)

>>rfe.fit(mixed\_X, y)

>>rfe.ranking\_

array([ 9, 12, 8, 1, 2, 3, 6, 4, 10, 11,

16, 5, 15, 14, 7, 13, 17, 18, 19, 20])

Итак, мы получили относительные рейтинги 20 признаков, “прогнанных” через алгоритм RFE. Ожидается, что первые 10 признаков, выбранных алгоритмом FRESH, будут иметь более высокий рейтинг, чем те, что были отвергнуты им. В значительной степени это так, но не полностью. Например, легко заметить, что во второй половине массива с рейтингами отвергнутых признаков, указаны 5- и 7-й наиболее важные признаки из всех 20. За исключением этого в отсутствие идеального совпадения полученные результаты согласуются с исходными предположениями. Как видите, метод RFE можно использовать для дальнейшего отбора единожды извлеченных признаков. Кроме того, он прекрасно подходит для проверки работоспособности алгоритма FRESH при точной настройке его входных параметров или количества генерируемых признаков.

Обратите внимание, что алгоритм FRESH по умолчанию работает без указания параметров, поэтому количество генерируемых признаков — это лучшее средство, с помощью которого можно повлиять на его производительность. Другой не менее важный параметр, который приходится устанавливать в алгоритме FRESH — fdr\_level, — определяет процент ненужных признаков, которые отвергаются по завершении процедуры. По умолчанию этот параметр равен .05, но можно задать намного большее значение, чтобы повысить качество отбора признаков, особенно при их генерировании в большом количестве без предварительного учета того, попадают ли они в область рассмотрения.

***Заключение***

Мы обсудили причины, побуждающие исследователей обращаться к признакам в анализе данных, а также рассмотрели простой пример использования признаков для преобразования небольшого временного ряда в ограниченный набор чисел, не уступающий по информативности исходному набору данных. Кроме того, были рассмотрены два программных модуля Python, предназначенных для автоматического выбора и генерации признаков в данных временных рядах, которые способны извлечь несколько тысяч признаков из одного временного ряда. Так как в дальнейшей работе будут полезны не все автоматически полученные признаки, их последующий отбор выполнялся с использованием специальных методик, позволяющих отсеять зашумленные и неинформативные варианты, — только после этого они становятся пригодными к передаче в конвейер анализа данных.

***Генерация признаков позволяет достигнуть ряда целей.***

• Предоставить сводные данные о временных рядах в формате, позволяющем применять к ним алгоритмы машинного обучения, большинство из которых предназначены для обработки наборов признаков для одной точки данных, а не всего временного ряда.

• Сжать данные временных рядов таким образом, чтобы свести временные наблюдения к нескольким числам и качественным показателям. Такой подход применяется не только в анализе, но и в хранении временных рядов в более кратком и удобочитаемом формате в случаях передачи на хранение полного временного ряда.

• Сформировать набор показателей для описания и определения сходства данных, которые получены или измерены в различных условиях. Обобщая данные в более широком плане, можно добиться их лучшего сопоставления, что представляется очень сложной задачей при прочих условиях.

Далее мы будем использовать сгенерированные признаки для подготовки к передаче данных в алгоритмы машинного обучения, работа которых заключается в классификации и прогнозировании по признакам временных рядов, а не по необработанным данным базовых временных рядов.

***Дополнительные источники***

*• Анализ временных рядов по признакам*

*Ben D. Fulcher, “Feature-Based Time-Series Analysis,” eprint arXiv: 1709.08055, 2017,* https://perma. cc/6LZ6-S3NC

Обзорная статья, написанная на доступном языке и принадлежащая перу разработчика пакета tsfresh в Matlab. В ней приведена обширная таксономия различных видов признаков, реализуемых для данных временных рядов. Кроме того, в ней наглядно проиллюстрированы категории всех обсуждаемых признаков. Акцент сделан на описании

важности использования автоматически сгенерированных признаков в исследовании данных — с расширением на их применение в анализе временных рядов.

*• Отбор признаков*

*Maximilian Christ et al., “Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh—A Python package)” Neurocomputing 307 (2018): 72-7,* https://oreil.ly/YDBM8

В этой статье приведен обзор пакета tsfresh для языка Python и алгоритма FRESH, рассматриваемых в этом разделе. Приведены сведения о вычислительной эффективности отдельных признаков и рассмотрены некоторые шаблоны их использования.

*Maximilian Christ et al., “Distributed and Parallel Time Series Feature Extraction for Industrial Big Data Applications,” paper presented at ACML Workshop on Learning on Big Data (WLBD), Hamilton, NZ, November 16 2016,* https : // arxiv.org/pdf/1610.07717.pdf

Это подробное техническое описание алгоритма FRESH содержит более детальную информацию о том, как работает алгоритм и почему он находит применение в промышленных временных рядах, а также обширный список справочных материалов в разделе ссылок.

*• Признаки для отдельных предметных областей*

*Wikipedia, “Technical Analysis for Financial Markets,”* https : //perma.cc/ 8533-XFSZ

Статья в Википедии, в которой приведена история разработки методов технического анализа финансовых рынков, а также содержится описание большого количества популярных индикаторов.

*Amjed S. Al-Fahoum and Ausilah A. Al-Fraihat, “Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains,” ISRN Neuroscience 2014, no. 730218 (2014),* https: //perma.cc/465U-QT53

Здесь рассматривается пример тестирования стандартных признаков временных рядов данных ЭЭГ (электроэнцефалограмм). Существует целая академическая индустрия генерации признаков для описания медицинских временных рядов, и этот документ включает доступное описание принципов выполнения этой задачи.

*Juan Bautista Cabral et al., “From FATS to Feets: Further Improvements to an Astronomical Feature Extraction Tool Based on Machine Learning, Astronomy and Computing 25 (2018),* https: //perma. cc/8ZEM-Y892

В этой статье приведен обзор последнего издания пакета Python, предназначенного для извлечения признаков из данных астрономических временных рядов. Рассчитанная на специалистов по астрономии, знакомых с целевым программным пакетом, она будет полезной всем, кто планирует использовать специальное программное обеспечение для генерации признаков, а также для понимания вызовов и задач, стоящих перед специалистами по анализу астрономических временных рядов.

*Alvin Rajkomar et al., “Supplementary Information for Scalable and Accurate Deep Learning for Electronic Health Records,” npj Digital Medicine 1, no. 18 (2018),* https://perma.cc/2LKM-32 6C

Комментарии к весьма интересной статье, опубликованной Google, в которой проиллюстрированы результаты успешного применения методов глубокого обучения для анализа электронных медицинских записей. Содержат подробную информацию о принципах приведения медицинских записей к количественным входным признакам, которые доступны для обработки нейронной сетью. Если вы не знакомы с технологиями глубокого обучения, то приступайте к изучению этого документа только после изучения Раздела 9. ГЛУБОКОЕ ОБУЧЕНИЕ ДЛЯ ВРЕМЕННЫХ РЯДОВ. Кроме того, перед знакомством с ним неплохо получить представление о вложениях пространств для глубокого обучения (https://perma.cc/3KAZ-9A3Y)

*• Периодограммы*

*“The Periodogram,” lecture notes, Eberly College of Science, Pennsylvania State University,* https : / /perma. cc/5DRZ-VPR9

Заметки из лекционного курса по прикладному анализу временных рядов Университета Penn State, знакомящие студентов с понятием периодограммы, принципами ее интерпретации и построения с помощью инструментов языка R.

*Jacob Т. VanderPlas, “Understanding the Lomb-Scargle Periodogram,” Astrophysical Journal Supplement Series 236, no. 1 (2018),* <https://arxiv.org/> pdf/1703.09824.pdf

Эта обширная статья, знакомящая читателей с общими концепциями оценки периодических зависимостей и, в частности, с методом Ломба- Скаргла.