Елена Капаца

Машинное обучение доступным языком

Елена Капаца

Машинное обучение доступным языком

«Автор»
2023

Капаца Е.

Машинное обучение доступным языком / Е. Капаца — «Автор», 2023

ISBN 978-5-0060-1962-1

Краткий гайд для новичков по машинному и глубокому обучению с разбором кода. Здесь вы найдете необходимый минимум по предмету, истолкованный языком, понятным школьнику. Некоторые разделы написаны с помощью chatGPT. По прочтении вы избавитесь от страха перед технологией и освоите базовый инструментарий подготовки данных, их загрузке в модель и ее донастройки. Подходит студентам технических специальностей.

ISBN 978-5-0060-1962-1

Содержание

Елена Капаца
Машинное обучение доступным языком

Введение

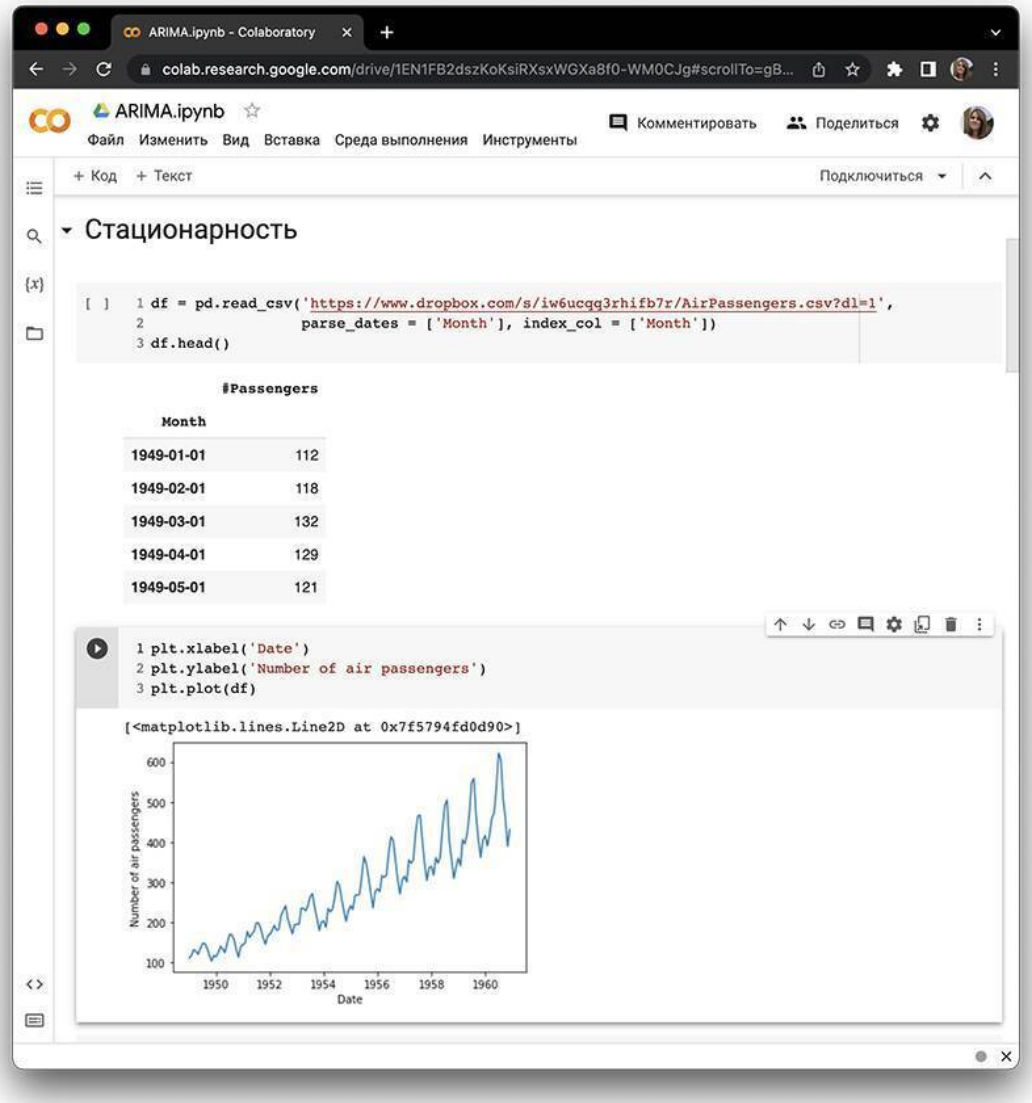
Приступая к изучению машинного обучения, студенты легко и непринужденно добиваются… запутанности. Пара-тройка непонятных терминов или неясностей при расчете – и все: мозг теряет нить и начинает воспринимать “по диагонали”. Продираться через непонятное довольно трудоёмко. Не каждый день у нас есть ресурс догугливать непонятное. Не каждый запрос в Google даст лаконичный понятный ответ.

Моя задача – описать детали этой мозаики языком, понятным старшекласснику. Я намеренно буду избегать формул, потому что знаю: каждая из них сокращает число читателей. Однако в книге будет код, и он будет расширенно комментироваться.

Минимальное требование к читателю – знание основ Python. Книга фокусируется на машинном обучении, и потому останавливаться на терминах вроде “переменной” и “списка” я не буду.

Если вы чувствуете, что пересиливаете себя при чтении, лучше сделайте перерыв. В Data Science будет предостаточно информации, однако в этой книге я постаралась собрать повторяющиеся в работе термины. Добиться их понимания особенно важно.

Некоторые главы будут базироваться на полноценных моделях и скриптах. В машинном обучении принято использовать так называемые ноутбуки – наборы ячеек с исполняемыми кусками кода:



Все используемые в дальнейшем ноутбуки можно открыть, запустить и скопировать себе для дальнейших экспериментов. Инструменты ML имеют свойство совершенствоваться, а это значит, что спустя 3-4 года после выхода книги некоторые участки кода вам придется отлаживать с помощью поисковиков.

Машинное обучение – это абстрактная концепция. Ее основные компоненты стоит описывать просто, пускай даже это вызовет раздражение профессионалов. Эта книга – серия взаимосвязанных статей. Их основная цель – осветить основные и популярные термины во взаимосвязи друг с другом. Ключевые понятия при первом упоминании я буду дополнять англоязычным термином. Так вы всегда сможете с легкостью отыскать дополнительные материалы.

Немалое влияние на меня оказал бестселлер Максима Ильяхова и Людмилы Сарычевой “Пиши, сокращай”. Потому эта книга написана в информационном стиле[1] и изобилует упрощениями. Если вы сохраните по прочтении ощущение удобства чтения и желание взбираться на эту познавательную гору дальше, то моя цель достигнута.

[1] Информационный стиль – инструмент для очистки текста от лишнего, для обнажения самой сердцевины текста.

Вы всегда можете “напитаться” полноценными зубодробительными статьями на моем сайте helenkapatsa.ru.

Приятного чтения! Я буду рада вашим предложениям и фидбэку в целом (kapatsahelen@gmail.com). Вы также можете запросить PDF-версию с цветовой разметкой кода. Это упростит восприятие материала.

Машинное обучение

Что же это такое? Машинное обучение (machine learning, ML) – наука о том, как заставить компьютеры выполнять объемную вычислительную задачу без явного программирования.

Классическим алгоритмам дают точные и полные правила для выполнения задачи, моделям Машинного обучения – данные. Мы говорим, что «подгоняем модель к данным» или «модель обучена на данных».

Проиллюстрируем это на простом примере. Предположим, мы хотим спрогнозировать цену дачного дома на основе:

- площади
- размера придомового участка
- количества комнат.

Мы могли бы попытаться построить классический алгоритм, который решает эту проблему. Этот алгоритм возьмет три вышеупомянутых признака (feature) и выдаст прогнозируемую цену на основе явного правила. Но на практике эта формула часто неочевидна.

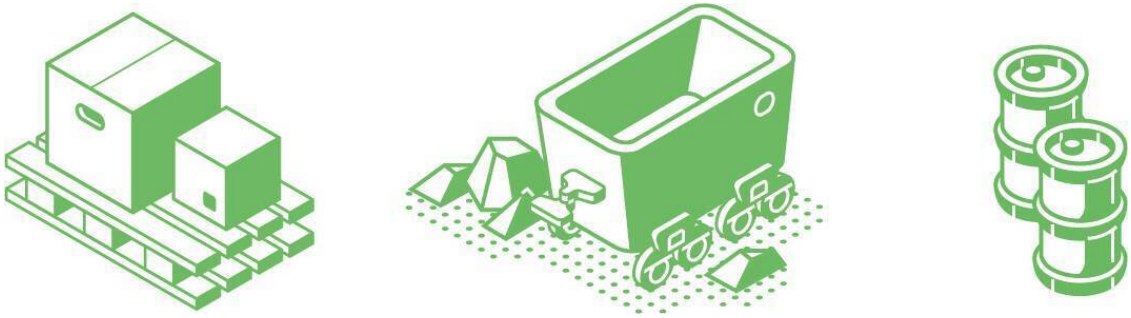
Однако мы хотим автоматизировать этот процесс и построить модель. Она будет корректировать формулу сама каждый раз, когда появляются новые примеры цен на жилье. В целом, ML невероятно полезно для задач, когда мы располагаем неполной или слишком обильной информацией для программирования вручную. В этих случаях мы можем предоставить имеющиеся сведения и позволить ей «изучить» недостающую. Затем алгоритм будет использовать статистические методы для извлечения недостающих знаний.

Машинное обучение способно выполнять широкий спектр задач:

- оценки стоимости чего угодно
- изменение изображений
- помощь на письме
- обработка звука
- генерация текста и многие другие.

Представьте, что Машинное обучение – это конвейер по сборке автомобилей. И первое, что потребуется для его работы – металл, различные композитные материалы, и в конечном итоге, топливо. Вся эта троица олицетворяет данные.

Данные



Данные – основа основ в ML. В контексте науки принято рассматривать два типа: традиционные и большие (big data).

Традиционные данные структурированы и хранятся в базах, управляемых с одного компьютера. На самом деле, эпитет «традиционный» введен для ясности: это помогает подчеркнуть различия с большими.

Большие данные, в свою очередь, массивнее, чем традиционные, по ряду характеристик:

- типы (числа, текст, изображения, аудио, видео и проч.)
- скорость извлечения и вычисления
- объем (тера-, пета-, эксабайты и проч.).

Набор однотипных данных, выделенный с целью обучения модели, называют датасетом (dataset). Их разделяют на следующие категории:

Классическая таблица

Здесь каждая строка имеет одинаковый набор характеристик-столбцов. Такие таблицы – датафреймы (dataframe) обычно хранятся либо в файлах форматов .csv, .parquet, либо в базах данных:

ID	ЗАБИТЫЕ МЯЧИ	ПР
Арсенал	1	
Ювентус	4	
Саутгемптон	5	
Манчестер Юнайтед	0	
Селтик	3	

Датасет о результативности футбольных команд

Текстовый документ

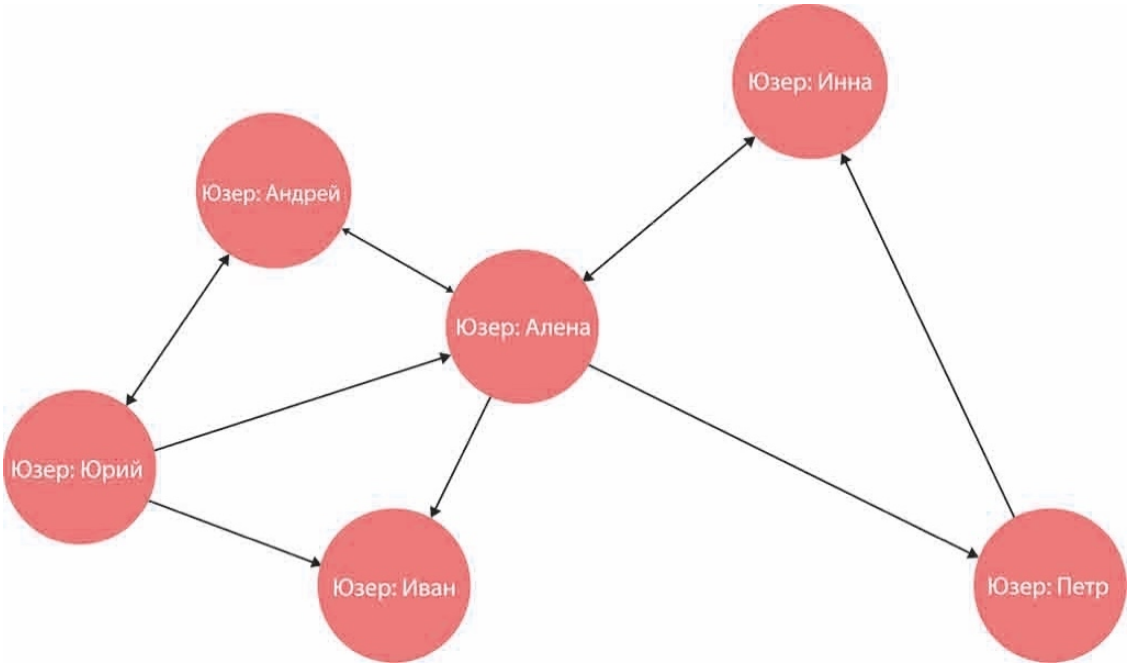
(document) Отдельно взятой единицей здесь является блок (corpus). Например, книгу можно рассматривать как датасет, состоящий из абзацев – корпусов.

"… После обучения в Университете Вашингтона Болл опубликовала статью в Journal of the American Chemical Society и отправилась на Гавайи, чтобы стать магистром химии. В 1915 г. она впервые среди женщин и афроамериканцев получила степень магистра в Гавайском колледже, где осталась преподавать".

Корпус из книги-датасета Рейчел Свайби "52 упрямые женщины"

Графы

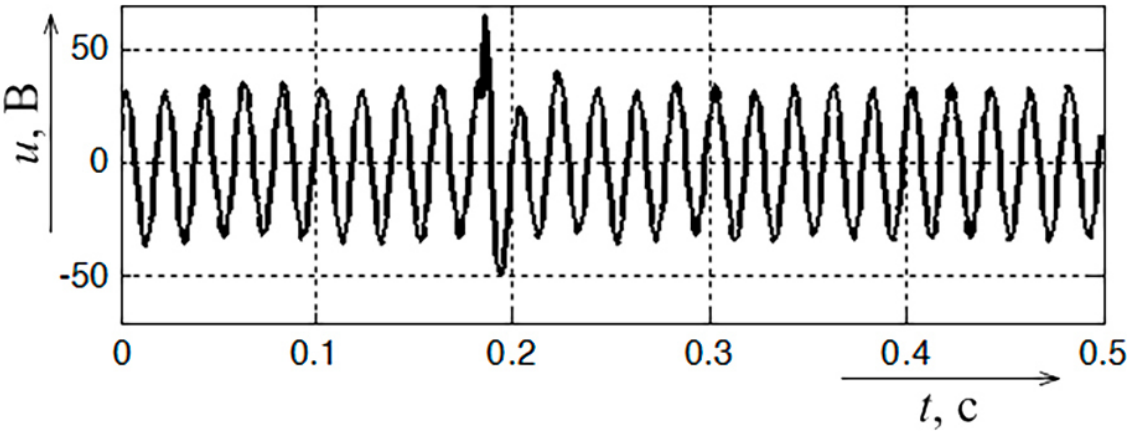
(graph) Здесь отдельно взятая единица – это связь между объектами:



Граф социальной сети

Аудиодорожки

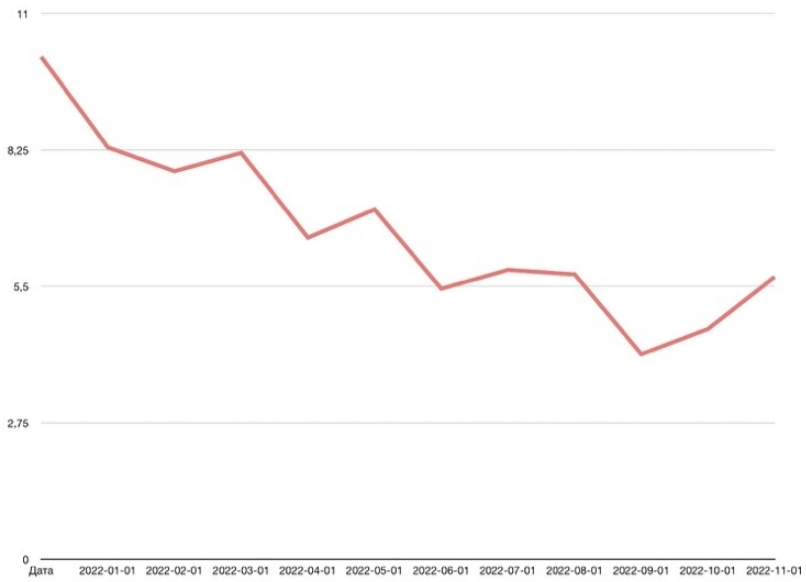
Здесь довольно очевидно: аудиозаписи. Помимо распознавания речи ML решает обширный спектр задач с помощью таких данных: очистка от шумов, написание музыки.



Временной ряд

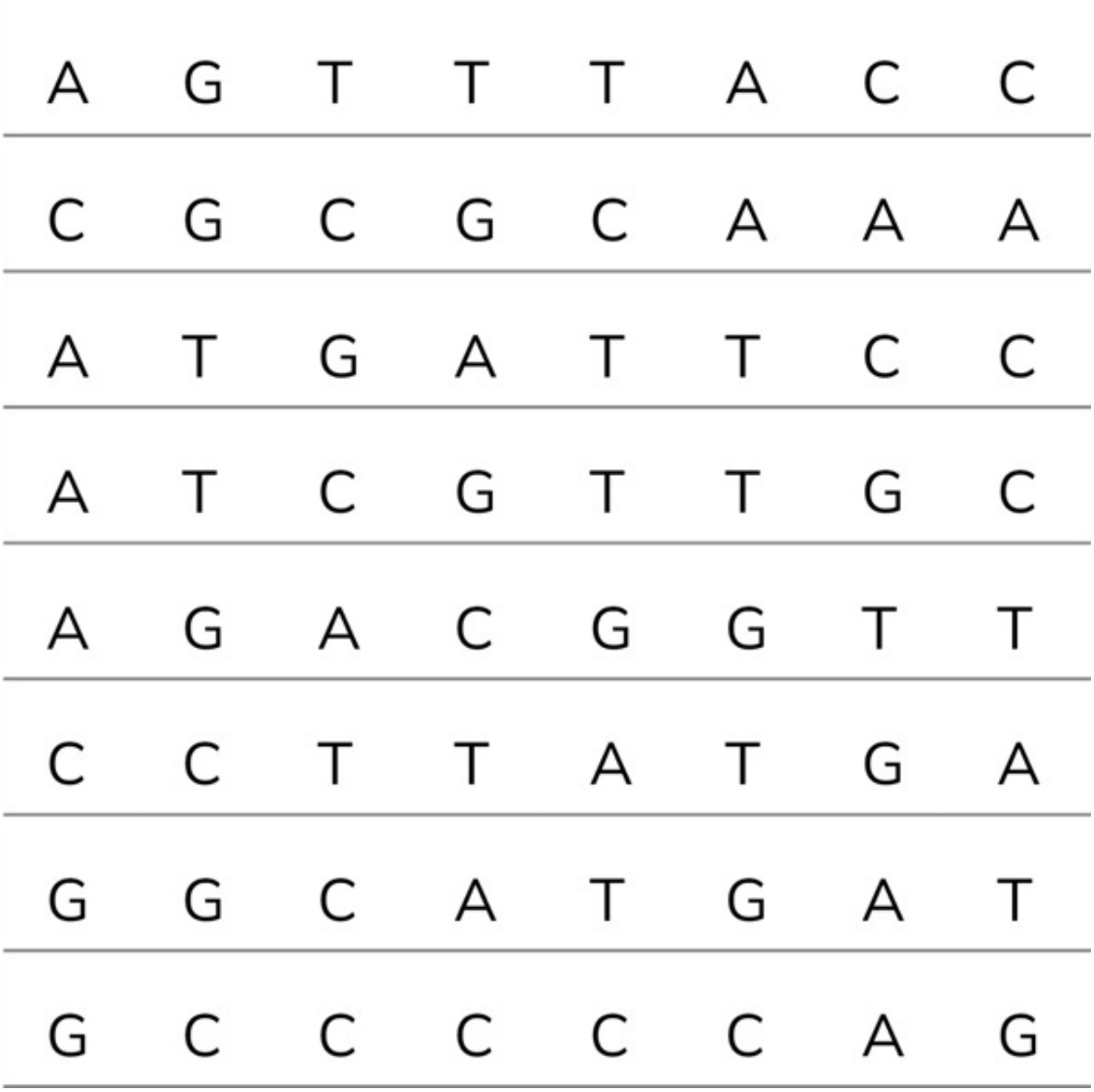
(time series) Здесь каждая точка привязана к временной оси x и, как правило, взаимосвязана с окружающими ее соседями.

ДАТА	ЦЕНА ОТКРЫТИЯ
2022-01-01	10,130
2022-02-01	8,300
2022-03-01	7,820
2022-04-01	8,190
2022-05-01	6,480
2022-06-01	7,050
2022-07-01	5,450
2022-08-01	5,830
2022-09-01	5,740
2022-10-01	4,130
2022-11-01	4,640
2022-12-01	5,690



Цена акции LG на момент открытия биржи на протяжении года

Последовательные данные

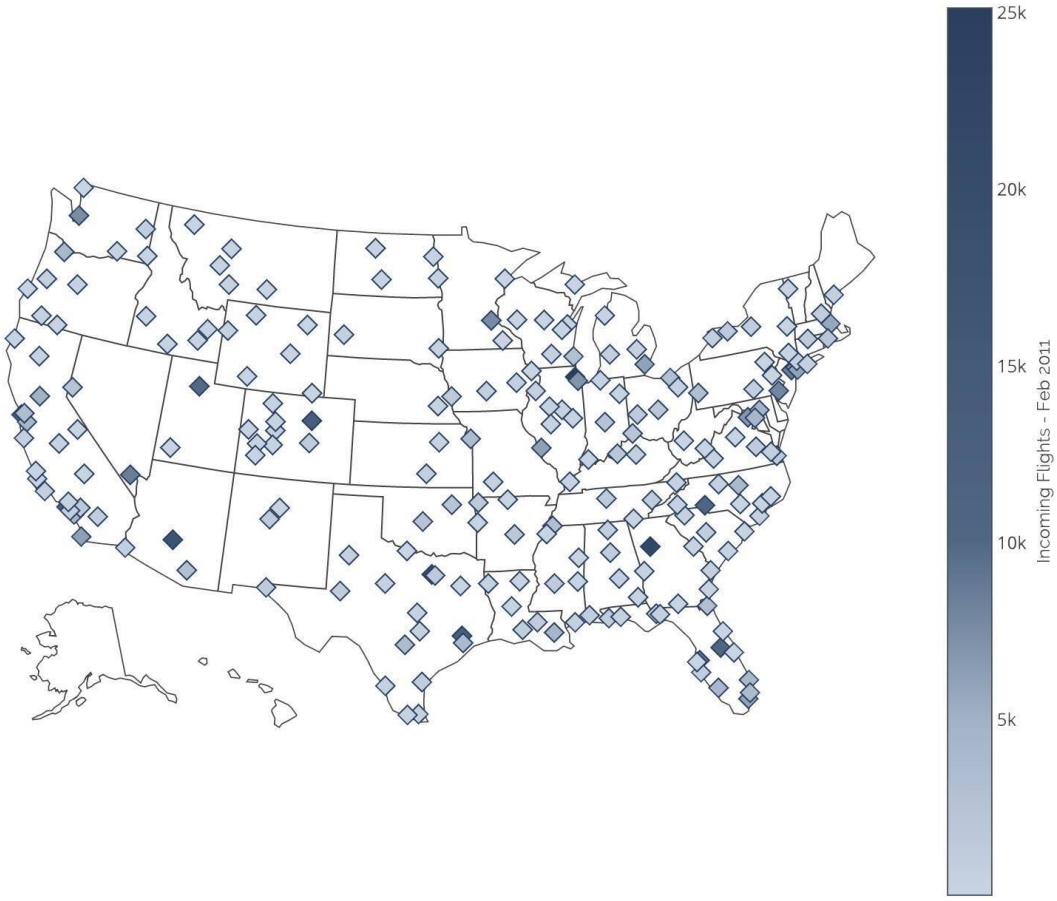


(sequence data) Состоят из набора отдельных объектов, таких как слова или буквы. Здесь нет временных меток; вместо этого есть позиции в упорядоченной последовательности:

На картинке справа яркий пример: геном – набор генов в хромосоме.

Пространственные данные

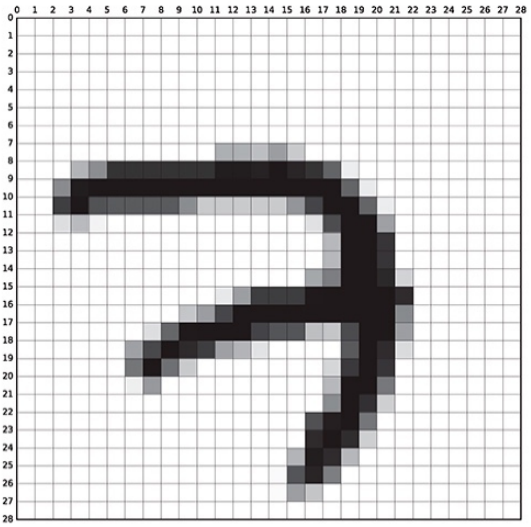
(geospatial data) Здесь каждая точка имеет координаты:



Трафик аэропортов США

Изображения

Здесь единицей является отдельная картинка. Видео рассматривается как набор картинок.



(**a**) MNIST sample belonging to the digit '7'.

(**b**) 100 samples from the MNIST training set.

Датасет рукописных цифр

Перед дата-сайентистами часто встает вопрос: где взять данные?

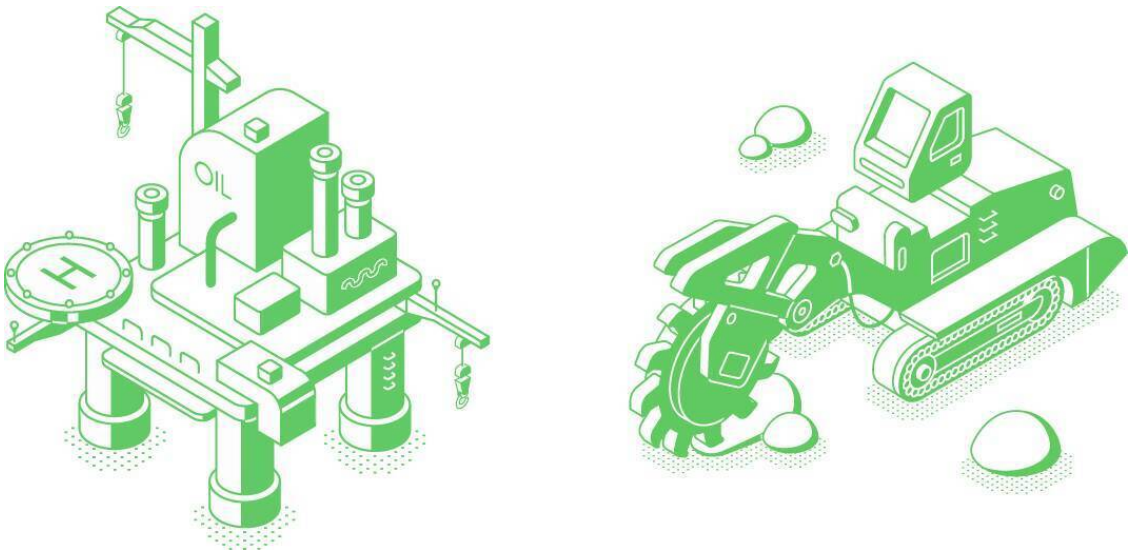
Студентам проще: у некоторых обширных библиотек вроде Scikit-learn встречаются собственные встроенные датасеты, прекрасно подходящие для обучения:

```
from sklearn.datasets import load_digits
digits = load_digits()
```

Помимо таких встроенных коллекций, данные предоставляют бесплатно еще и ресурсы вроде kaggle.com.

А вот на рабочей ниве требования к информации куда специфичнее. Порой проще и лучше собрать свой набор, и в таком случае мы обращаемся к инструментам ETL.

ETL



(extract, transform, load – извлечь, преобразовать и загрузить) группа процессов, происходящих при переносе данных из нескольких систем в одно хранилище.

Если у вас есть данные из нескольких источников, вам необходимо:

• Извлекать данные из исходного источника

• Преобразовывать информацию путем очистки, объединения и других способов подготовки

• Загружать результат в целевое хранилище

Как правило, один инструмент ETL выполняет все три шага. Пожалуй, самый популярный сегодня представитель такого программного обеспечения – это Hadoop.

ETL уходит своими корнями в 1970-е годы к появлению централизованных хранилищ данных. Но только в конце 1980-х и начале 1990-х годов, когда они заняли центральное место, мир ощутил потребность в специализированных загрузочных инструментах. Первым пользователям нужен был способ извлекать информацию из разрозненных систем, преобразовывать ее в целевой формат и загружать в конечное место хранения. Первые инструменты ETL были примитивными, и объем данных, которые они обрабатывали, был скромным по сегодняшним меркам.

По мере роста объема данных росли и хранилища данных, а программные инструменты ETL множились и становились все более сложными. Но до конца 20-го века хранение и преобразование данных осуществлялось в основном в локальных хранилищах. Однако произошло нечто, навсегда изменившее наш взгляд на хранение и обработку.

Облачные вычисления

Объем данных, которые мы генерируем и собираем, продолжает расти с экспоненциальной скоростью. У нас есть все более сложные инструменты, которые позволяют нам использовать все наши данные для получения представления о исследуемом предмете в режиме онлайн.

Традиционная инфраструктура не может масштабироваться для хранения и обработки большого объема данных. Это неэффективно с точки зрения затрат. Если мы хотим выполнять

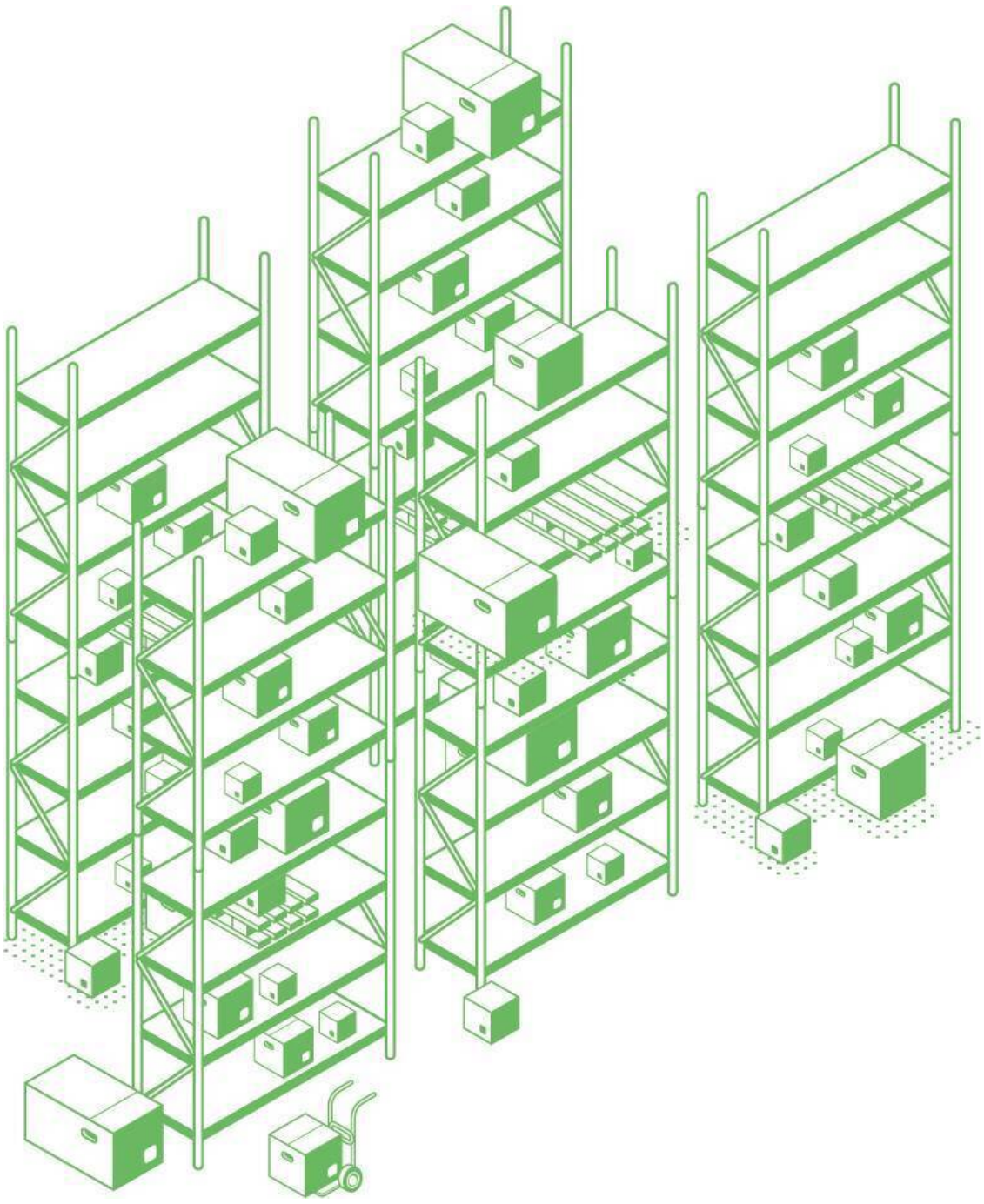
высокоскоростную, сложную аналитику и строить подобные модели, облако – оптимальное решение.

Облачные хранилища могут бесконечно масштабироваться для размещения практически любого объема данных. Облачное хранилище также позволяет координировать огромные рабочие нагрузки между группами вычисляющих серверов.

Преобразования и моделирование данных часто выполняются с помощью SQL – языка запросов к базе данных.

Конечная точка ETL – хранилище данных (DWH).

DWH



(data warehouse – хранилище данных) предназначено исключительно для выполнения запросов и часто содержит большие объемы исторических данных. Данные в хранилище обычно поступают из широкого круга источников, таких как:

- Логи приложений
- Сведения, собираемые с форм на сайте
- Записи различных устройств, вроде видеокамер и датчиков температуры

Хранилище объединяет большие объемы данных из нескольких источников. Это позволяет генерировать ценные инсайты[2] и улучшать процесс принятия решений. С ростом объема и качества DWH становится бесценным объектом для бизнес-аналитики. Типичное хранилище данных часто включает следующие элементы:

- Реляционная база данных
- ПО для ETL
- Инструменты анализа и визуализации
- Модели машинного обучения

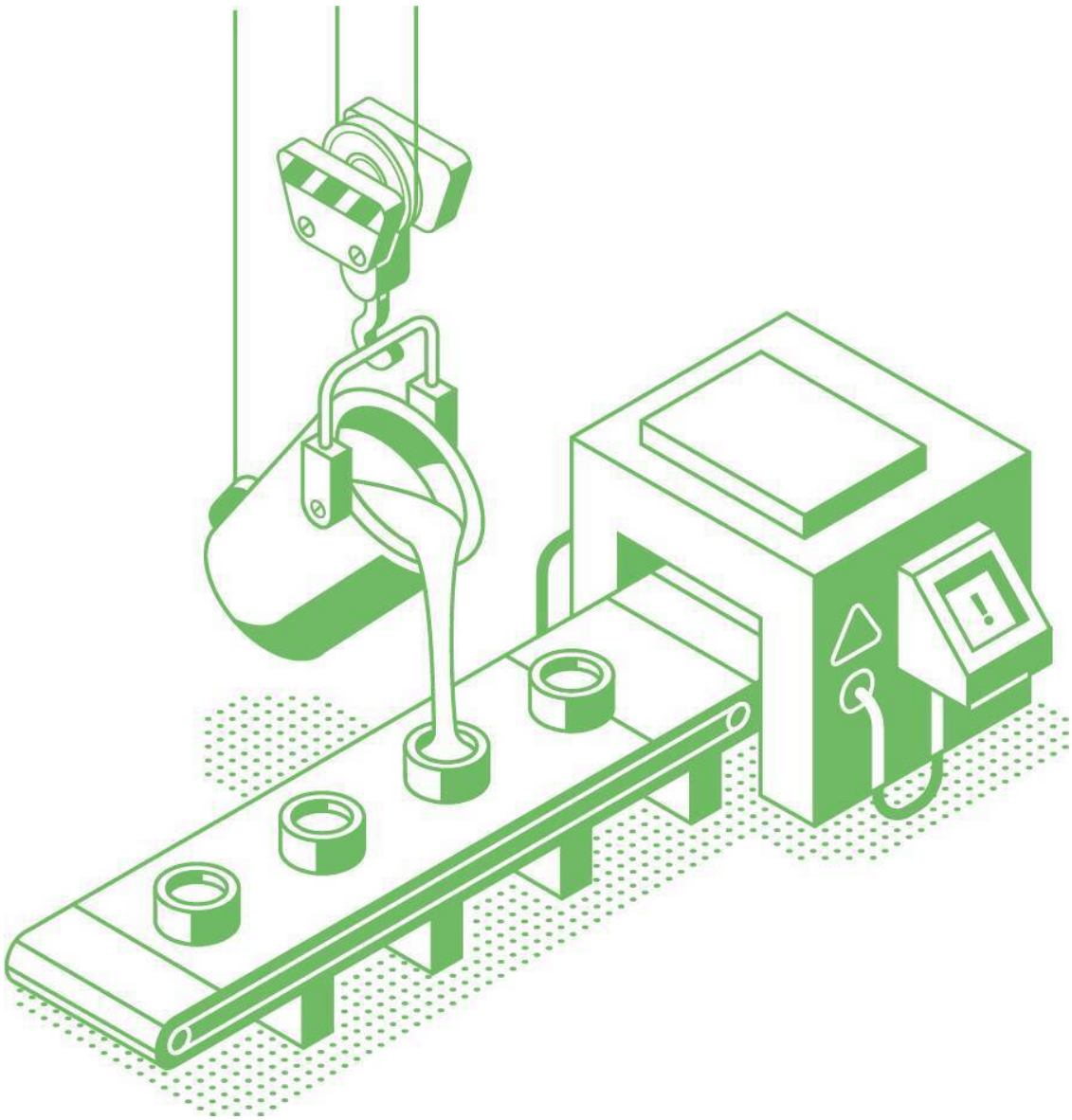
К популярным хранилищам можно отнести Amazon Redshift, Google BigQuery и Greenplum.

Стоит отличать DWH от так называемого озера данных (data lake). Хранилище содержит очищенные и структурированные данные, готовые к анализу на основе предопределенных потребностей бизнеса. В озере же все содержится в необработанном, неструктурированном виде.

Когда команда ML получает доступ к такому хранилищу, то предваряет моделирование целой серией действий – разведочным анализом данных (EDA).

[2] Решение задачи

EDA



(exploratory data analysis – разведочный анализ данных) предварительное исследование датасета с целью определения его основных характеристик, взаимосвязей между признаками, а также сужения набора методов, используемых для создания модели.

Давайте рассмотрим, на какие этапы разбивают EDA. Для этого мы используем данные[3] банка, который продает кредитные продукты своим клиентам. Возьмет ли клиент кредит или нет?

Мы располагаем обширным набором переменных (столбцов):

[3] Исходная англоязычная версия датасета: kaggle.com/datasets/volodymyrgavrysh/bank-marketing-campaigns-dataset

ПРИЗНАК	КРАТКОЕ ОПИСАНИЕ	ТИП ДАННЫХ	ПРИМЕР
Возраст		Числовой	
Работа	Профессия	Категориальный	Менеджер, рабочий, предприниматель, домработница
Семейный статус		Категориальный	в разводе, женат / замужем, не женат / не замужем, неизвестно
Образование		Категориальный	Базовое (4 класса), неграмотный, университетское образование
Кредитный дефолт	Было ли невыполнение договора займа?	Булевый	Нет, да, неизвестно
Ипотека	Есть ли ипотека?	Булевый	Нет, да, неизвестно
Займ	Есть ли займ?	Булевый	Нет, да, неизвестно
Контакт	Контактный вид связи	Категориальный	Сотовый телефон, городской телефон
Месяц	Последний контактный месяц года	Категориальный	Янв, фев, мар, ..., ноя, дек
День недели	Последний контактный день недели	Категориальный	Пн, вт, ..., сб, вс
Длительность	Продолжительность последнего звонка в секундах	Числовой	
Кампания	Количество контактов, выполненных во время этой кампании	Числовой	
День	Количество дней, прошедших с момента последнего обращения	Числовой	
Предыдущий контакт	Количество контактов, выполненных до этой кампании	Числовой	
Доходность	Результат предыдущей маркетинговой кампании	Категориальный	Отсутствует, присутствует, неизвестно
Колебание уровня безработицы	Отклонение от базового коэффициента занятости	Числовой	
Индекс потребительских цен	Измерение среднего уровня изменения цен на товары и услуги за определённый период в экономике	Числовой	
Индекс потребительской уверенности	Степень оптимизма относительно состояния экономики	Числовой	
Европейская межбанковская ставка предложения на три месяца	Усреднённая процентная ставка по межбанковским кредитам	Числовой	
Количество сотрудников в компании	Количество работников	Числовой	
Y	Взял ли клиент кредитный продукт	Булевый	

Это не сам датасет, а только описание столбцов

Столбец Y назван так неслучайно: это общепринятое обозначение целевой переменной (target variable). Изучив 40 тысяч записей о клиентах, модель автоматически сможет предсказывать, возьмет новый клиент кредит или не возьмет.

Довольно увесистый датасет: записей в нем более 40 тысяч. Для начала[4] импортируем датасет и посмотрим на "шапку". С помощью метода head() мы отобразим шапку датафрейма и первые пять записей:

```
df = pd.read_csv('https://www.dropbox.com/s/62xm9ymoaunnfg6/bank-full.csv?dl=1', sep=';')
df.head()
```

Параметр sep используется, чтобы задать нестандартный разделитель данных по столбцам, в данном случае – точку с запятой.

№	ВОЗРАСТ	РАБОТА	СЕМЕЙНЫЙ СТАТУС	...	ЕВРОПЕЙСКАЯ МЕЖБАНКОВСКАЯ СТАВКА	КОЛИЧЕСТВО СОТРУДНИКОВ В КОМПАНИИ	Y
1	27	Самозанятый	Не женат / не замужем	...	5,045	5195,8	Нет
2	30	Предприниматель	Женат / замужем	...	5,045	5195,8	Да
3	39	Голубой воротничок	Женат / замужем	...	5,045	5195,8	Да
4	42	Менеджер	Женат / замужем	...	5,045	5195,8	Да
5	42	Самозанятый	Женат / замужем	...	5,045	5195,8	Да

Все столбцы мы отображать здесь, конечно, не будем

[4] Здесь и далее ячейка с импортом библиотек будет пропущена. С полной версией кода можно ознакомиться в конце главы по QR-коду со ссылкой.

Удаление дубликатов

(duplicates removing) Повторяющиеся записи искажают статистические показатели. Всего несколько повторов – и среднее значение столбца сместится в их пользу. Дубликаты также снижают качество обучения модели. Для начала уточним, сколько у нас строк с помощью df.shape. Затем удалим повторы с помощью drop_duplicates() и обновим данные о размере данных:

```
print(df.shape)
df.drop_duplicates(inplace=True)
print(df.shape)
```

Библиотека pandas вообще сопровождает любителей и профессионалов на каждом шагу, так что у некоторых ее компонентов параметры одинаковые. Чтобы удалить повторы “на месте”, без излишнего перекопирования датафрейма, дополняем drop_duplicates() параметром inplace, равным True.

Ячейка выдает, что удалила 41188 – 41176 = 12 дубликатов:

```
(41188, 21)
(41176, 21)
```

Хоть число и небольшое, все же качество набора мы повысили.

Обработка пропусков

(omission handling) Если пропусков у признака-столбца слишком много (более 70%), такой признак удаляют. Проверим, насколько разрежены наши признаки:

df.isnull().mean() * 100

Метод isnull() пройдется по каждой ячейке каждого столбца и определит, кто пуст, а кто нет. Метод mean() определит концентрацию пропусков в каждом столбце. На 100 мы умножаем, чтобы получить значение в процентах:

Конец ознакомительного фрагмента.