Review

1.Introduction

什么是数据工程?

What is Data Engineering?

数据工程是设计和构建数据收集、存储和分析系统的实践。

Data engineering is the practice of designing and building systems for collecting, storing, and analyzing data.

为什么需要数据工程?

Why need Data Engineering?

大数据正在改变我们的业务方式，并产生了对能够收集和管理大量数据的数据工程师的需求。

Big data is changing the way we do business and creating a need for data engineers who can collect and manage large quantities of data.

数据工程的目的

The purpose of Data Engineering

数据工程的目标是提供有组织的标准数据流，以实现数据驱动模型，如机器学习模型和数据分析。

The Goal of Data Engineering is to provide organized, standard data flow to enable data driven models such as machine learning models, data analysis.

Data engineering pipeline

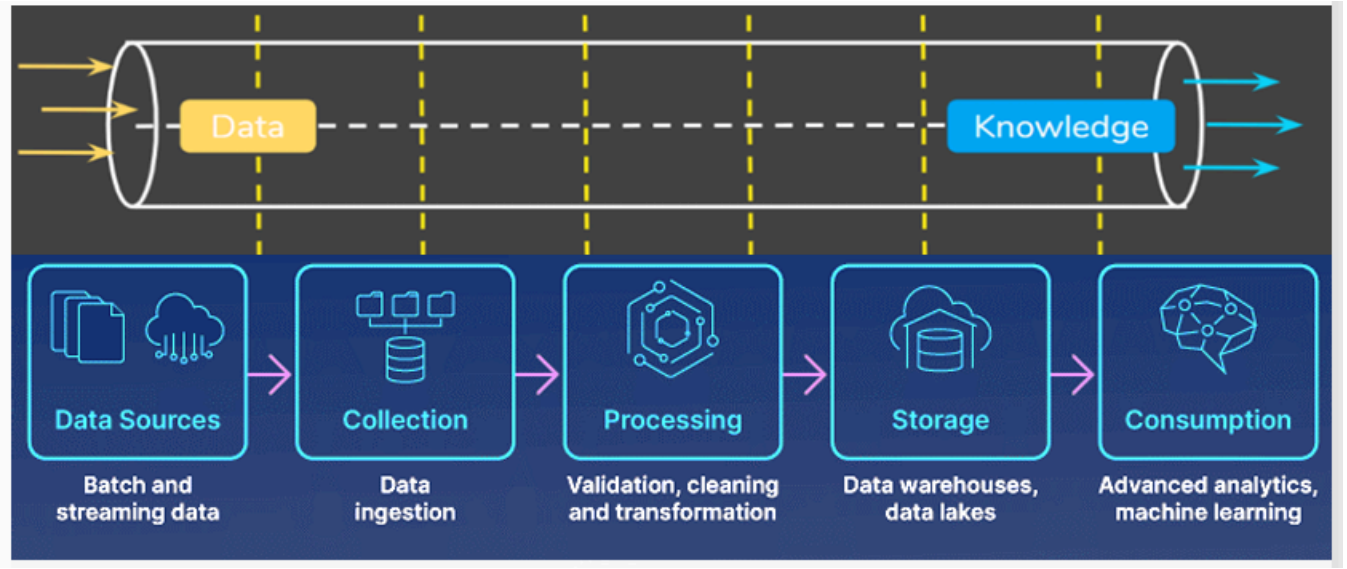
数据工程管道

Data engineering pipeline is the process of collecting data, processing data and send it to destination.

数据工程管道是收集数据、处理数据并将其发送到目的地的过程。

Including data collection, data processing, data storage and data consumption.

包括数据收集、数据处理、数据存储和数据消费。



数据类型

Data types

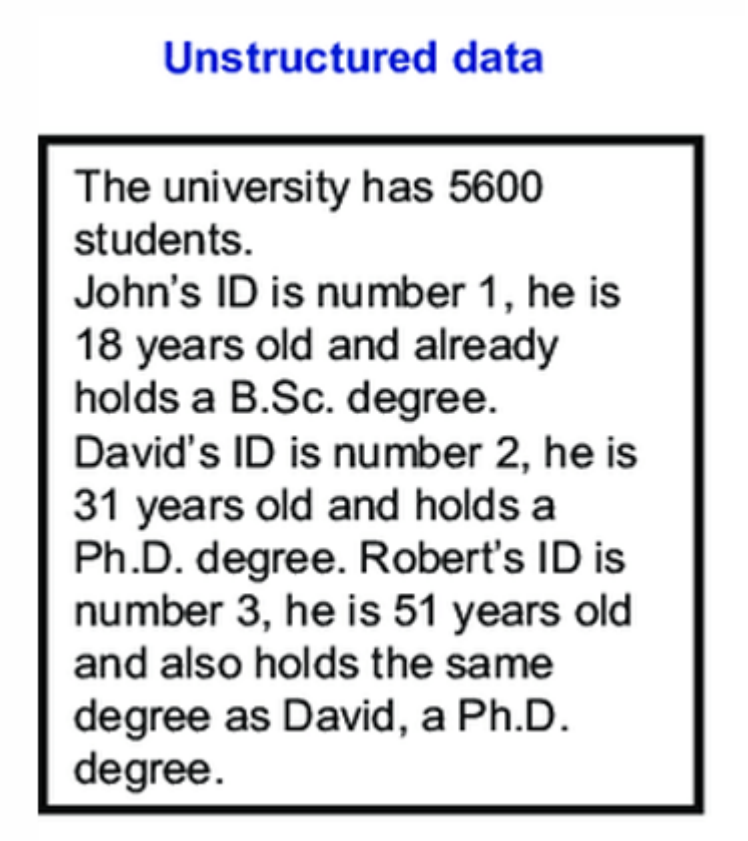
非结构化数据、半结构化数据、结构化数据

Unstructured Data, Semi-structured Data, Structured Data

Unstructured Data

数据复杂，多为定性信息，无法按行和列进行结构化处理

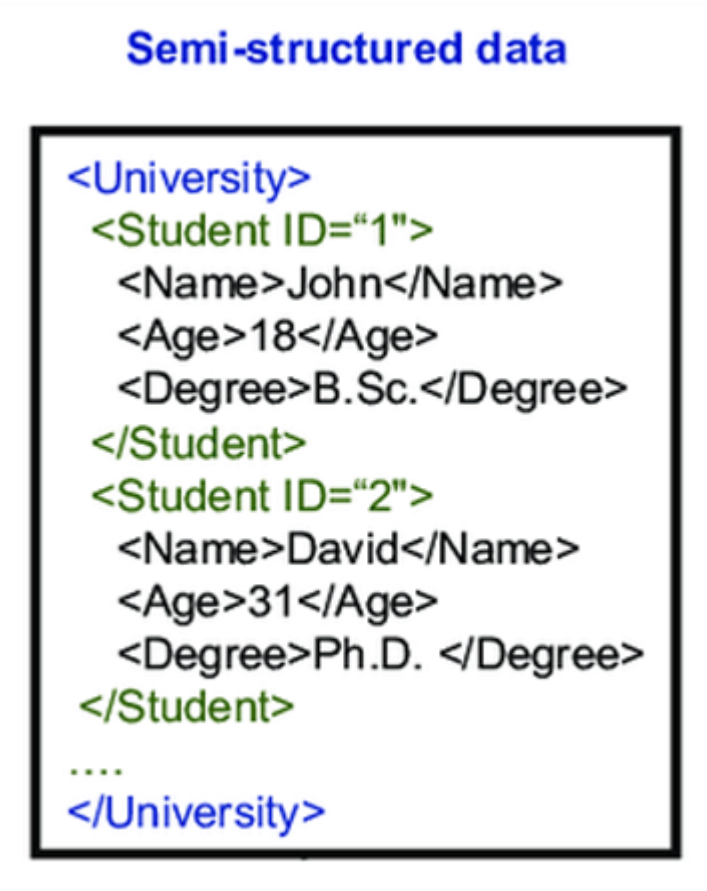
Data that is complex and mostly qualitative information that cannot be structured into rows and columns



Semi-structured Data

具有一致特征的数据和不符合严格结构的数据的混合

Mix of data that has consistent characteristics and data that does not conform to a rigid structure



Structured Data

格式严格、可按行列组织的数据

Data that follows a rigid format and can be organized into rows and columns

数据专业人员使用的语言

Languages for data professionals

查询语言、编程语言以及 Shell 和脚本语言

Query Languages(MySQL), Programming Languages(Python) and Shell and Scripting Languages(cmd Shell)

2.Data Acquisition

数据来源

Sources of data

关系数据库、平面文件和 XML 数据集、应用程序接口和网络服务

Relational databases, Flat files and XML datasets, APIs and web services

Relational databases

使用表格组织数据，每个表格由行（记录）和列（字段）组成，每列都有固定的类型。不同表格通过外键建立连接

Use tables to organize data, each table consists of rows (records) and columns (fields), and each column has a fixed type. Different tables establish connection with foreign keys

Flat files and XML datasets

以纯文本格式存储数据。最常见的平面文件格式是 .CSV

Store data in plain text format.Most common flat file format is .CSV

APIs and web services

应用程序接口和网络服务通常会侦听传入的请求，这些请求可以是用户的网络请求，也可以是应用程序的网络请求，并以纯文本、XML、HTML、JSON 或媒体文件的形式返回数据。

APIs and Web Services typically listen for incoming requests, which can be in the form of web requests from users or network requests from applications, and return data in plain text, XML, HTML, JSON, or media files.

Web scraping

构建一个代理，以自动方式从网上下载、解析和整理数据

The construction of an agent to download, parse, and organize data from the web in an automated manner

识别数据->抓取->存储

Identify data->scrape->store

HTML is a good data source for data scraping

- HTML is a standard markup language for creating web pages.
- HTML provides the building blocks to provide structure and formatting to documents.
- Python ‘requests’ library could get the html content from a webpage.
- HTML 是一种用于创建网页的标准标记语言。
- HTML 提供了为文档提供结构和格式的构件。
- Python 的 "requests "库可以从网页中获取 HTML 内容。

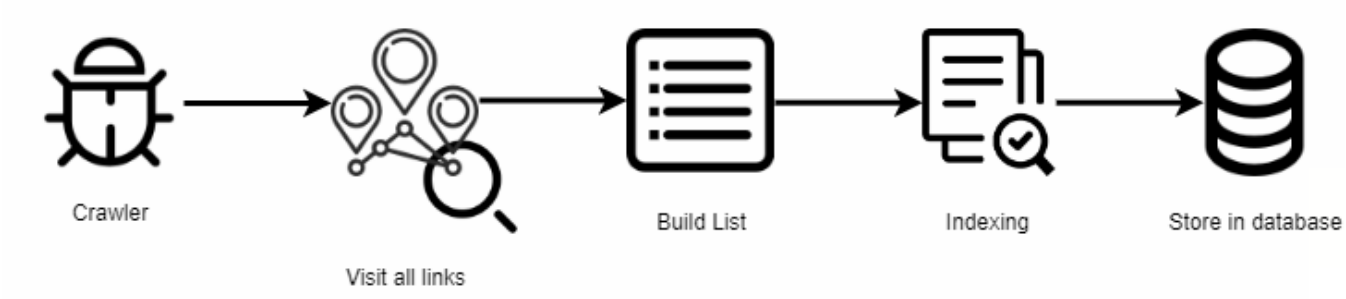
网络搜刮与网络爬行的区别

Differences between web scraping and web crawling

网络搜刮与网络爬行的区别

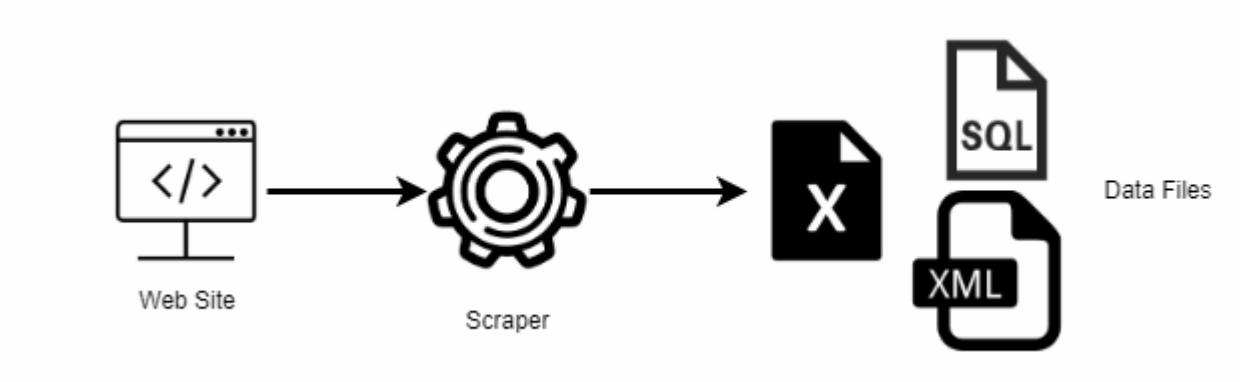
网络抓取： 使用工具读取、复制和存储网站内容，以便存档或编制索引。 抓取通常涉及网页的网络（多个页面）

Web Crawling: Using tools to read, copy and store the content of the websites for archiving or indexing purposes. Crawling usually deals with a network of webpages



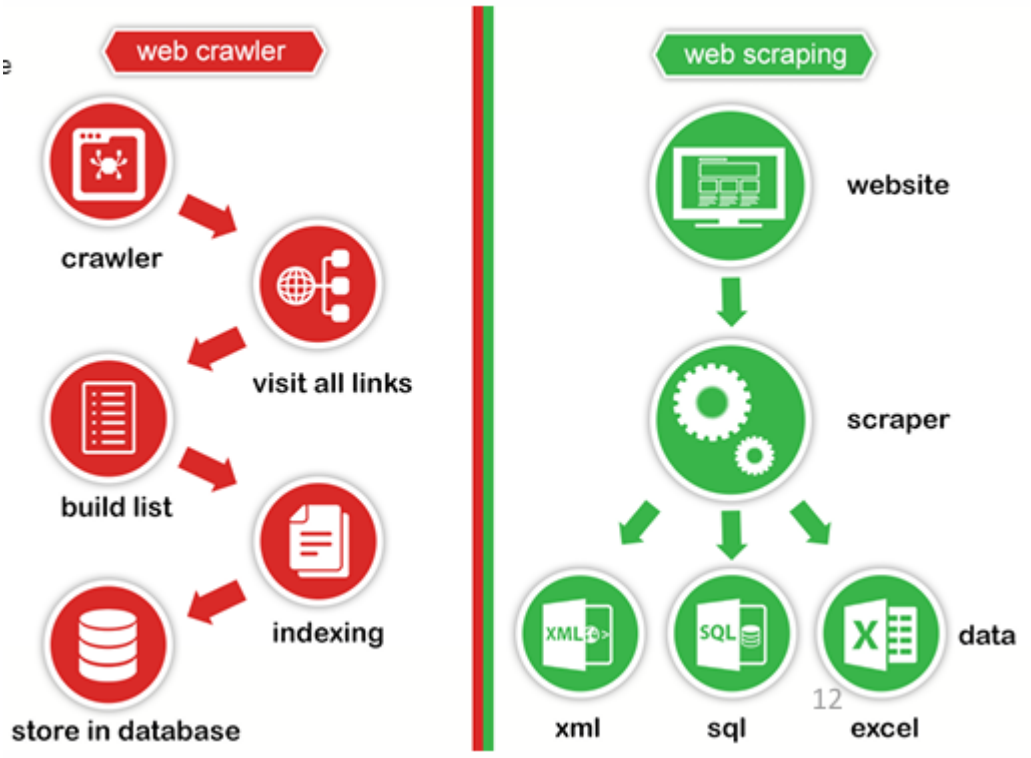
网络抓取： 通常从单个网页或单个网站中提取大量特定数据

Web Scraping: Extracting a large amount of specific data usually from a single webpage or a single website



最明显的区别在于获取数据的范围，网络搜刮通常关注单个页面，而网络爬虫则关注一组网站。

The most obvious difference is the range of acquiring data, Web scraping usually focus on single page while Web Crawling pays attention to a set of websites.

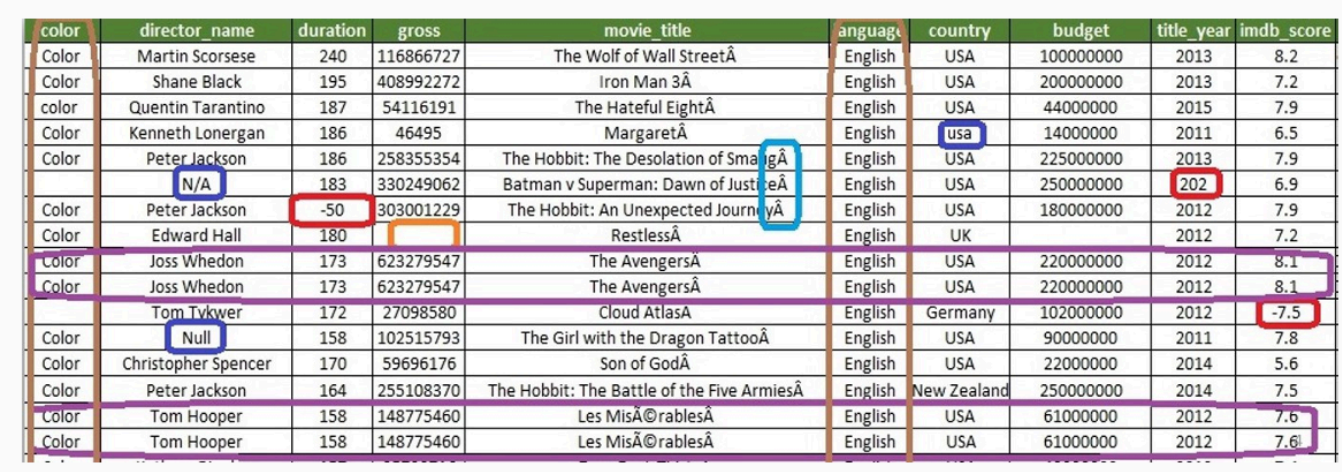


3. Data Preprocessing

Why need preprocessing?

因为现有的数据很脏：不完整、有噪音、不一致、有意为之

Because the are dirty data existing: incomplete, noisy, inconsistent, intentional



color	director_name	duration	gross	movie_title	anguag	country	budget	title_year	imdb_score
Color	Martin Scorsese	240	116866727	The Wolf of Wall StreetÂ	English	USA	100000000	2013	8.2
Color	Shane Black	195	408992272	Iron Man 3Â	English	USA	200000000	2013	7.2
color	Quentin Tarantino	187	54116191	The Hateful EightÂ	English	USA	44000000	2015	7.9
Color	Kenneth Lonergan	186	46495	MargaretÂ	English	usa	14000000	2011	6.5
Color	Peter Jackson	186	258355354	The Hobbit: The Desolation of SmaugÂ	English	USA	225000000	2013	7.9
	N/A	183	330249062	Batman v Superman: Dawn of JusticeÂ	English	USA	250000000	202	6.9
Color	Peter Jackson	-50	303001229	The Hobbit: An Unexpected JourneyÂ	English	USA	180000000	2012	7.9
Color	Edward Hall	180		RestlessÂ	English	UK		2012	7.2
Color	Joss Whedon	173	623279547	The AvengersÂ	English	USA	220000000	2012	8.1
Color	Joss Whedon	173	623279547	The AvengersÂ	English	USA	220000000	2012	8.1
	Tom Tykwer	172	27098580	Cloud AtlasÂ	English	Germany	102000000	2012	-7.5
Color	Null	158	102515793	The Girl with the Dragon TattooÂ	English	USA	90000000	2011	7.8
Color	Christopher Spencer	170	59696176	Son of GodÂ	English	USA	22000000	2014	5.6
Color	Peter Jackson	164	255108370	The Hobbit: The Battle of the Five ArmiesÂ	English	New Zealand	250000000	2014	7.5
Color	Tom Hooper	158	148775460	Les MisÃ©rablesÂ	English	USA	61000000	2012	7.6
Color	Tom Hooper	158	148775460	Les MisÃ©rablesÂ	English	USA	61000000	2012	7.6

Major tasks in data preprocessing:

数据清理： 填补缺失值，平滑噪声数据，识别或删除异常值，解决不一致问题。

Data cleaning: Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.

数据集成： 整合多个数据库、数据集、文件或注释。

Data integration: Integration of multiple databases, data cubes, files, or notes.

数据转换： 归一化（缩放至特定范围）、汇总。

Data transformation: Normalization (scaling to a specific range), aggregation.

减少数据： 获得体积缩小的表示，但产生相同或相似的分析结果, 消除冗余数据。

Data reduction: Obtains reduced representation in volume but produces the same or similar analytical results, eliminating redundant data.

Data Cleaning

处理失踪的 3 种方法：

1. 忽略：当缺少类别标签时
2. 手动填写：相当繁琐且不可行
3. 自动填写：

- 使用全局常量，如 "不知道
- 使用同一类别的平均值
- 使用最有可能的值：基于推理。

3 ways to deal with missing:

1. Ignore: when the class label is missing
2. Manually fill: quite tedious and infeasible
3. automatically fill:
    - use global constant like "unknow"
    - use the mean value of the same class
    - use the most probable value: based on inference.

应对噪音的 4 种方法：

1. 二进制法（也可用于离散化）：分类并归入二进制->通过均值、中值和边界等进行平滑处理。
2. 聚类： 检测并移除异常值
3. 半自动方法：计算机检测，人工检查
4. 回归法：将数据拟合为回归函数。

4 ways to deal with noisy:

1. Binning method(can also be used for discretization): sort and classify into bin-> smooth by means, median and boundaries, etc.
2. Cluster: Detect and remove outliers
3. Semi-automated method: detect by computer, check by human
4. Regression: fit the data into regression functions.

Regex

正则表达式是在文本中指定搜索模式的字符序列。

A regular expression is a sequence of characters that specifies a search pattern in text.

• Regex examples

- `.at` matches any three-character string ending with "at", including "hat", "cat", "bat", "4at", "#at" and " at" (starting with a space).
- `[hc]at` matches "hat" and "cat".
- `[^b]at` matches all strings matched by `.at` except "bat".
- `[^hc]at` matches all strings matched by `.at` other than "hat" and "cat".
- `^[hc]at` matches "hat" and "cat", but only at the beginning of the string or line.
- `[hc]at$` matches "hat" and "cat", but only at the end of the string or line.
- `\[.\]` matches any single character surrounded by "[" and "]" since the brackets are escaped, for example: "[a]", "[b]", "[7]", "[@]", "[]]", and "[ ]" (bracket space bracket).
- `s.*` matches s followed by zero or more characters, for example: "s", "saw", "seed", "s3w96.7", and "s6#h%(>>>m n mQ".

Data Transformation

归一化：按比例缩小到指定的小范围内

- 最小-最大归一化
- z 分数归一化
- 十进制缩放归一化

Normalization: scaled to fall within a small, specified range

- min-max normalization
- z-score normalization
- decimal scaling normalization

属性/特征结构

- 根据给定属性构建新属性

Attribute/feature construction

- New attributes constructed from the given ones

Data reduction

获得一个较小的数据集，并能产生相同的分析结果。

obtain a smaller data set that is able to produce the same analysis result.

降维

dimensionality reduction:

选择概率分布与原始特征接近的最小特征集。

Select a minimum set of features that the probability distribution is as close as the original features.

更易于理解

Make it easier to understand

Data discretization

将连续数据转换为离散数据

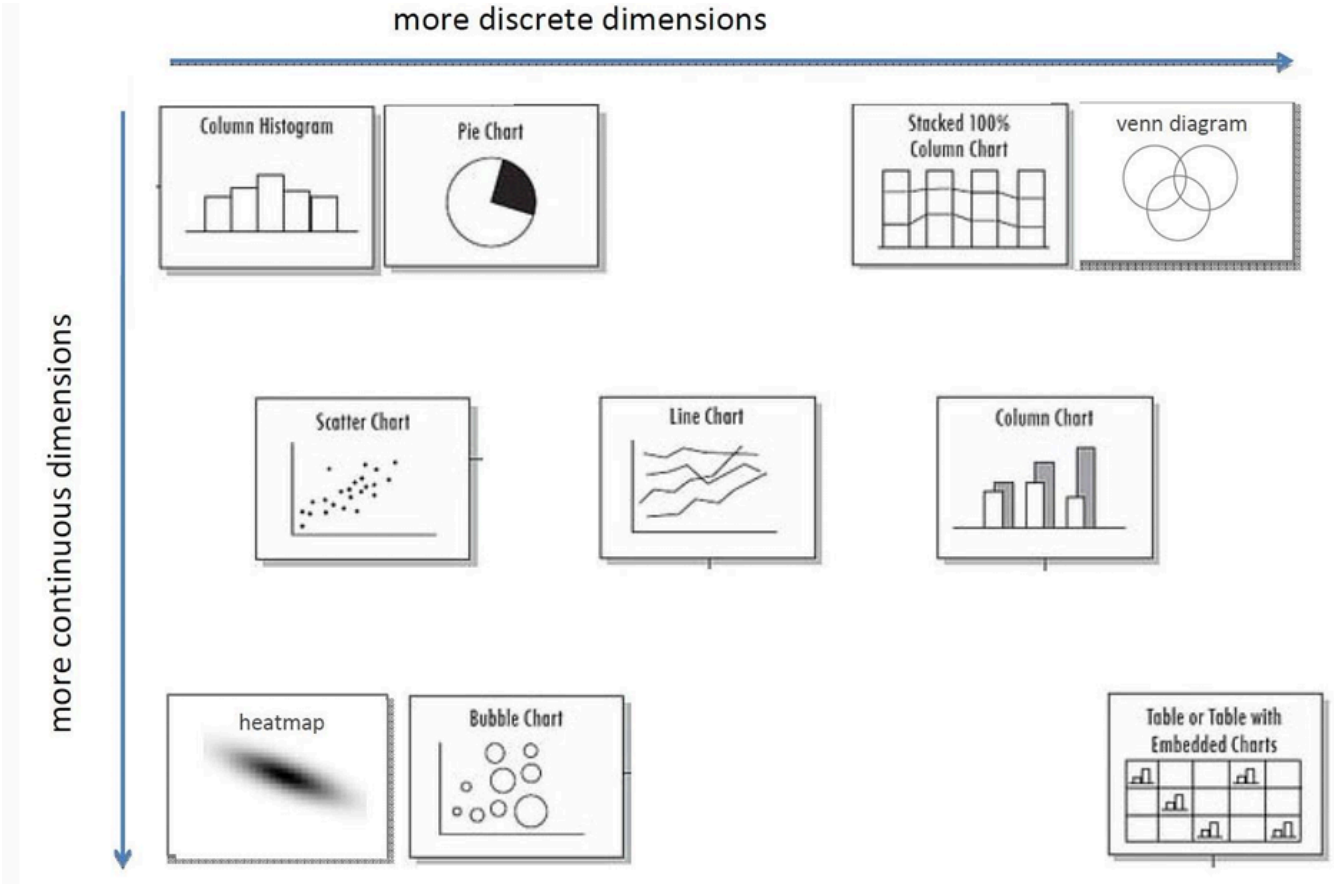
Converting continuous data to discrete data

1. 分层和递归分解
    - 分选
    - 直方图
    - 聚类
2. 基于熵的离散化
3. 基于自然分区的分割
4. Hierarchical and recursive decomposition:
    - Binning
    - Histogram
    - Clustering
5. Entropy-based discretization
6. Segmentation by natural partitioning

4. Data visualization

数据可视化是对抽象数据进行可视化表示的研究，它使信息易于理解、解释和保留。

Data visualization is the study of visual representations of abstract data, making information easy to understand, interpret and retain .



动态图：看起来像直方图，另外还能显示变化信息。

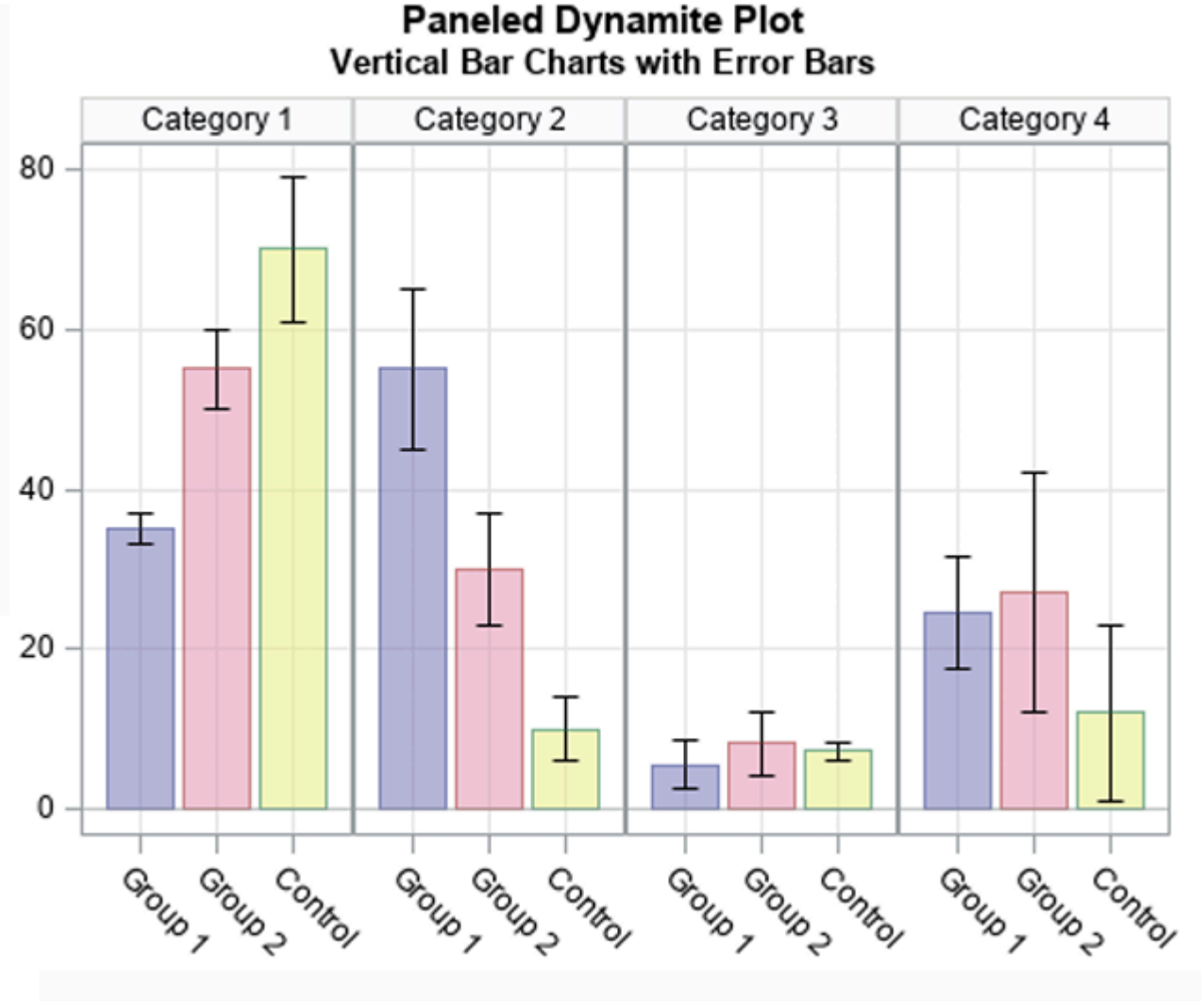
作为一种离散的可视化和分组方法，它很容易理解，并能揭示数据是否可信。

但有些数据是隐藏的。

dynamic plot: looks like histogram additionally telling variability information.

It's easy to understand as a discrete visualization and grouping method, and reveals whether the data is trustworthy or not.

But some data is concealed.



在选择使用哪种方式时，有 6 个因素需要考虑：

比较：将一组数值与另一组数值进行比较。 (柱状图)

分布：显示一组数值的分布。 (散点图)

整体的各个部分：显示各部分如何构成整体（饼图）

随时间变化的趋势：了解某个变量随时间变化的趋势。

找出偏差：查看哪些数值偏离了正常值。 (折线图)

了解关系：确定（或显示）两个（或多个）变量之间的关系。

When choosing what way to use, there are 6 factors to consider:

Compare: to compare one set of value(s) with another. (Column chart)

Distribution: to show the distribution of a set of values. (Scatter chart)

Parts of whole: to show how various parts comprise the whole.(Pie chart)

Trend over time: to understand the trend over time of some variable.(Line chart)

To find out the deviations: to see which values deviate from the norm. (Line chart)

To understand the relationship: to establish (or show) relationship between 2 (or more) variables.(Table or Line chart)

2项权衡

2 trade offs:

信息量与可读性：信息量过多会使人难以阅读，而信息量过少则会使数据隐蔽。

Informativeness vs. readability: too much information makes it hard to read, while too little information makes data concealed.

可能的解决方案：分级组织

possible solution: hierarchical organization

以数据为中心与以观众为中心

Data-centric vs. viewer-centric: when considering visualization method, care more about viewer or data?

观众更倾向于阅读他们熟悉的可视化数据。

但新颖的可视化方式可以展示更多的数据信息。

Viewers are more tend to read data visualized by visualization they are familiarized with.

But novel visualization can show more information of data.

5. 6. LLMs for data engineering

ChatGPT 具有强大的文本处理能力。

ChatGPT has powerful textual processing abilities.

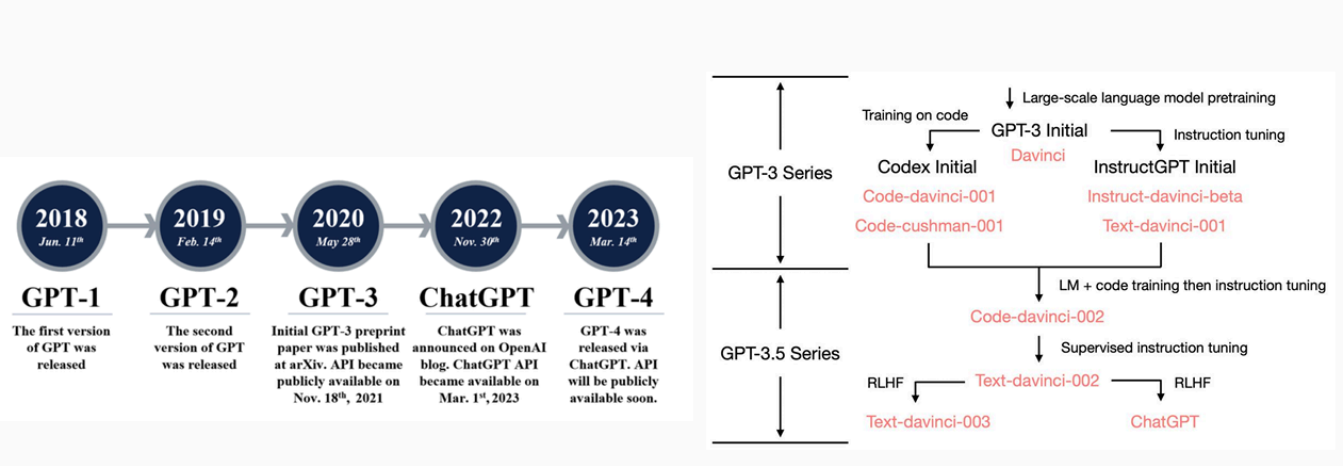
ChatGPT 功能多样。

ChatGPT is versatile.

语言模型是词序列的概率分布。 它用于确定给定的单词排序是否听起来像自然语言，从而预测下一个单词。

A language model is a probability distribution over sequences of words. It's used to determine whether a given ordering of words sounds like natural language, thus, to predict the next word.

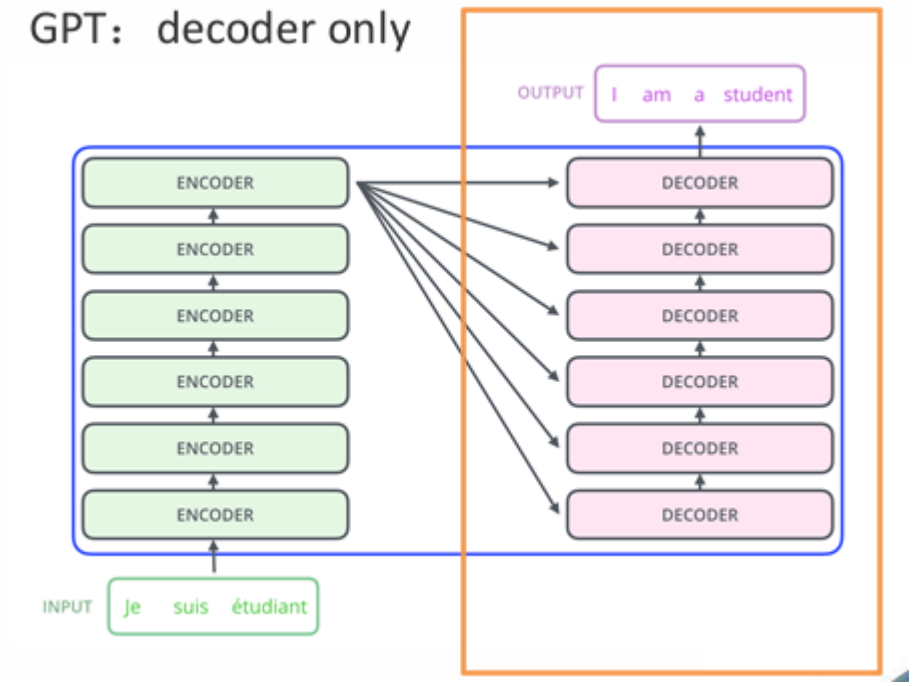
The evolution of GPT models



ChatGPT's core technique: Transformer

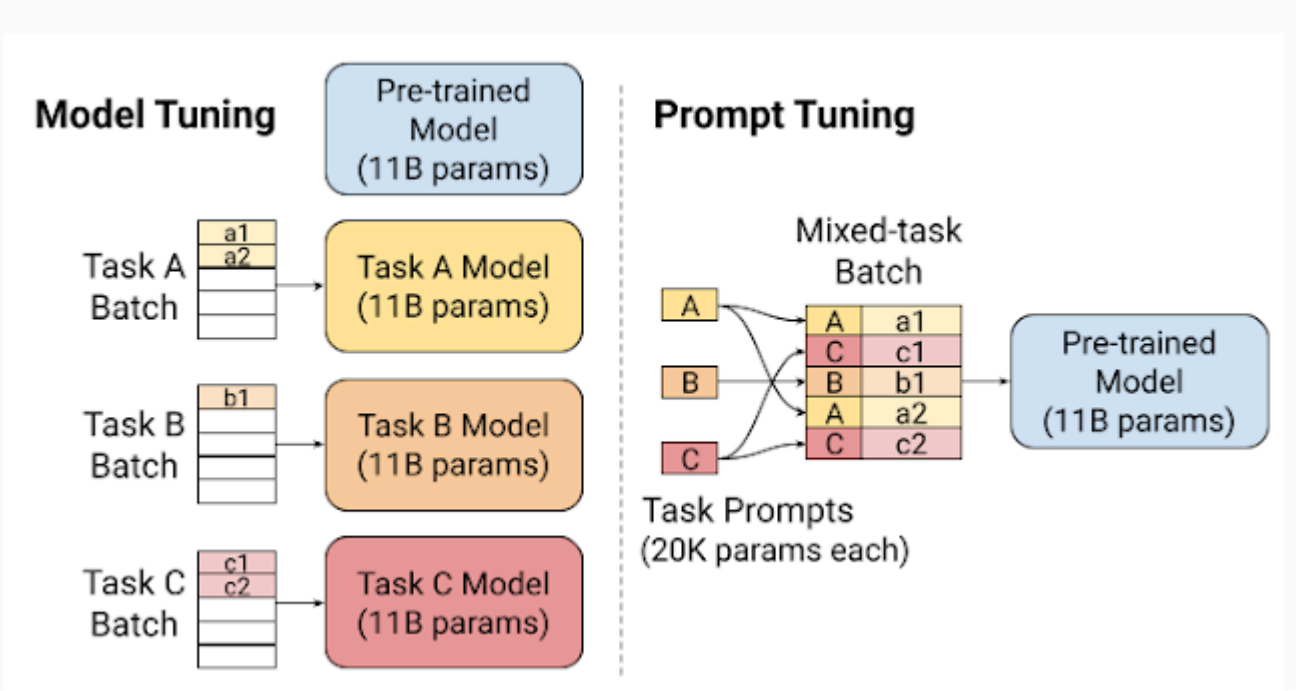
Transformer based on multi-head self attention mechanism, to solve sequence to sequence understanding and generation tasks.

After that, GPT is just a decoder, used to generate text.



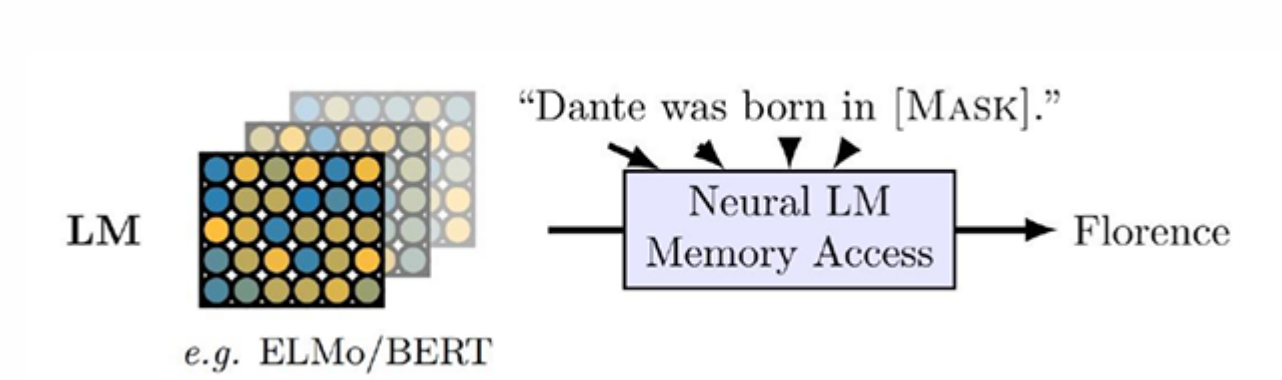
在传统模型中，预训练和微调是必要的。 但对于大型语言模型来说，微调成本太高。 降低成本的一个好方法是提示词调整

In traditional model, pretrain and finetuning is necessary. But for large language model, finetuning is to expensive. A good way to reduce cost is prompt tuning

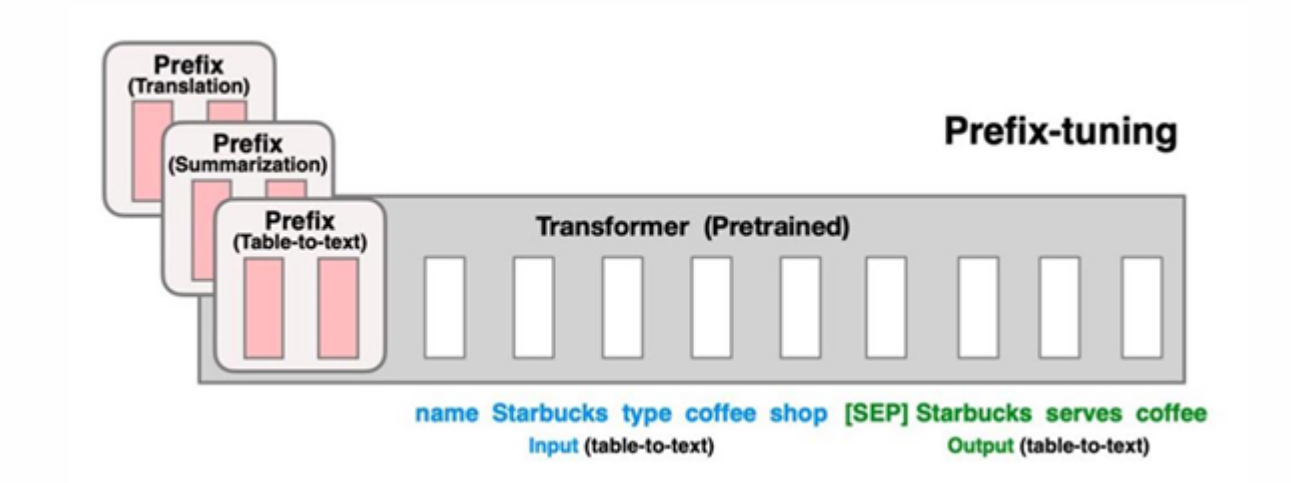


Prompt shape：

- Cloze 提示：在文字串中填空
- Cloze prompts: fill in the blanks of a textual string



- 前缀提示：继续字符串前缀
- Prefix prompts: continue a string prefix



Automated prompt searching

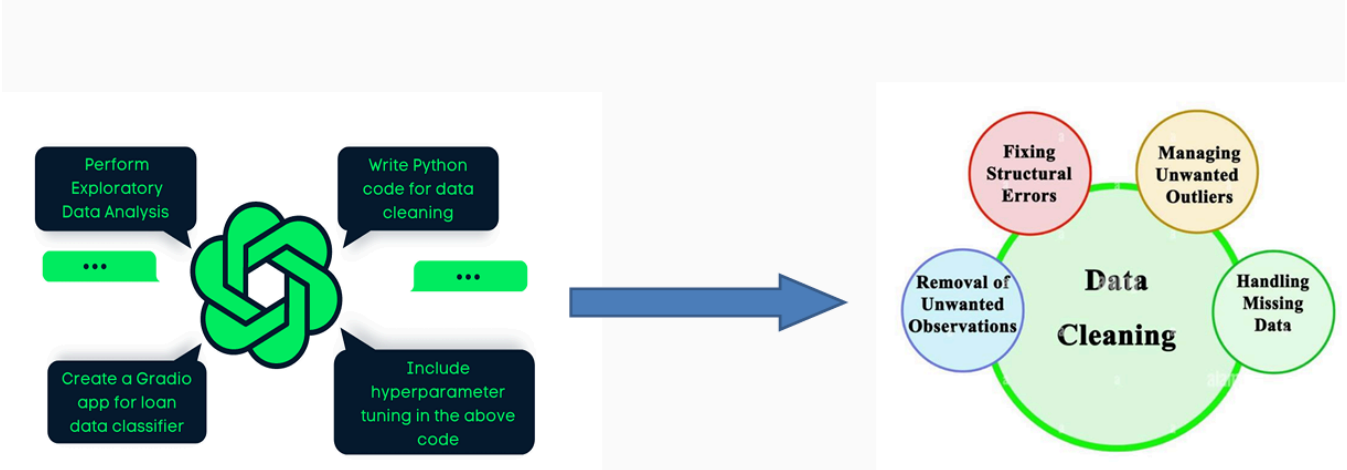
离散提示（又称硬提示）：在离散空间中描述的模板，通常与自然语言短语相对应。 通常由人类以自然语言的形式给出。

连续提示（又称软提示）： 连续提示直接在模型的嵌入空间中以向量或点的形式进行提示。

Discrete prompts (a.k.a. hard prompts): templates described in a discrete space, usually corresponding to natural language phrases. Usually given by human, in the form of natural language.

Continuous prompts (a.k.a. soft prompts): Continuous prompts perform prompting directly in the embedding space of the model, in the form of vector or dots.

LLMs could be used for data preprocessing



LLM 增强数据：

LLM augmented data:

LLM 需要高质量的数据来改进，但来自书籍、论文的高质量数据却很少。 而低质量的数据在训练中表现糟糕。 敏感信息也不能用于训练。

LLM needs high-quality data to improve, but high quality data from books, papers is scarce. While low quality data performs badly in training. And sensitive information could not be used for training.

一个可行的解决方案是创建用于训练的合成/增强数据。

A possible solution is creating synthetic/augmented data for training purpose.

由 LLM 指导的数据收集：

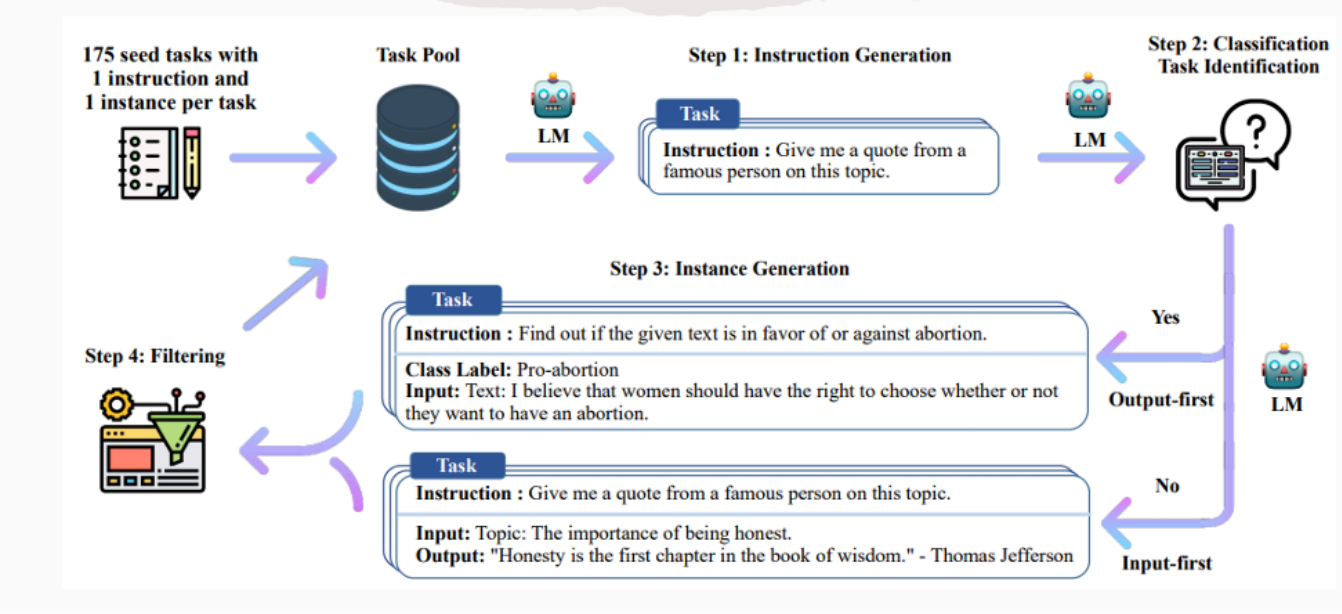
LLM-guided data Collection:

模型提炼： 使用 LLM 输出作为标签来训练小型模型。 这通常比在原始数据上训练小型模型更好。

Model distillation: Use the LLM output as a label to train small models. This is often better than training a small model on the original data.

自我指令：使用 LLM 生成的指令对其指令进行微调，可视为一个迭代过程。

Self-Instruct: use the LLM-generated instruction to finetune its instruction, could be seen as an iterative process.



LLM could also be used for text analysis:

- 情感分析文本
- 分类信息
- 提取器
- 翻译
- sentiment analysis text
- classification information
- extraction machine
- translation

7. Data Indexing

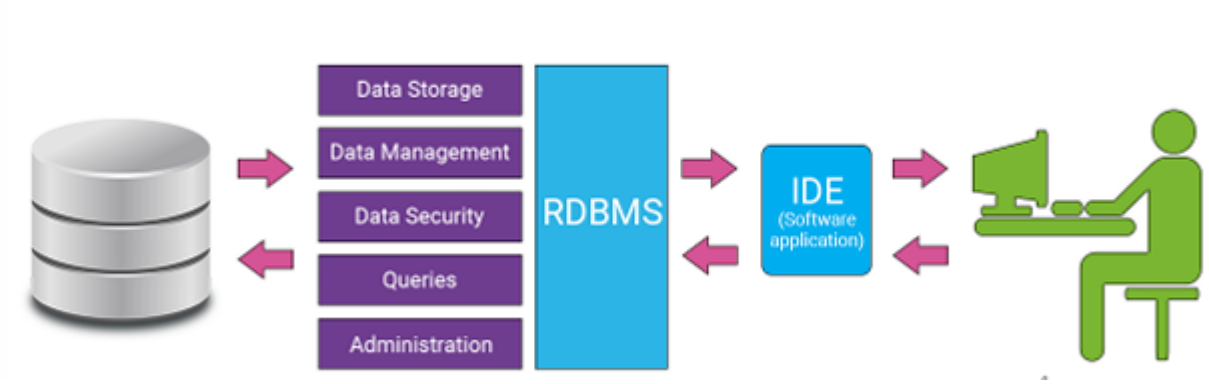
Relational databases

关系数据库：

- 基于关系型数据模型
- 使用关系数据库管理系统 (RDBMS) 管理和维护数据
- 使用 SQL（结构化查询语言）进行查询和维护

Relational databases：

- Based on relational model of data
- Use relational database management system (RDBMS) to manage and maintain data
- use SQL(Structured Query Language) for querying and maintaining



NoSQL databases

所有不遵循 RDBMS 原则的数据库都可以是 NoSQL 数据库，包括

- 无关系型
- 无 RDBMS
- 不只有 SQL

文档数据库、图数据库、键值数据库等

All databases don't follow RDBMS principle could be NoSQL databases, including:

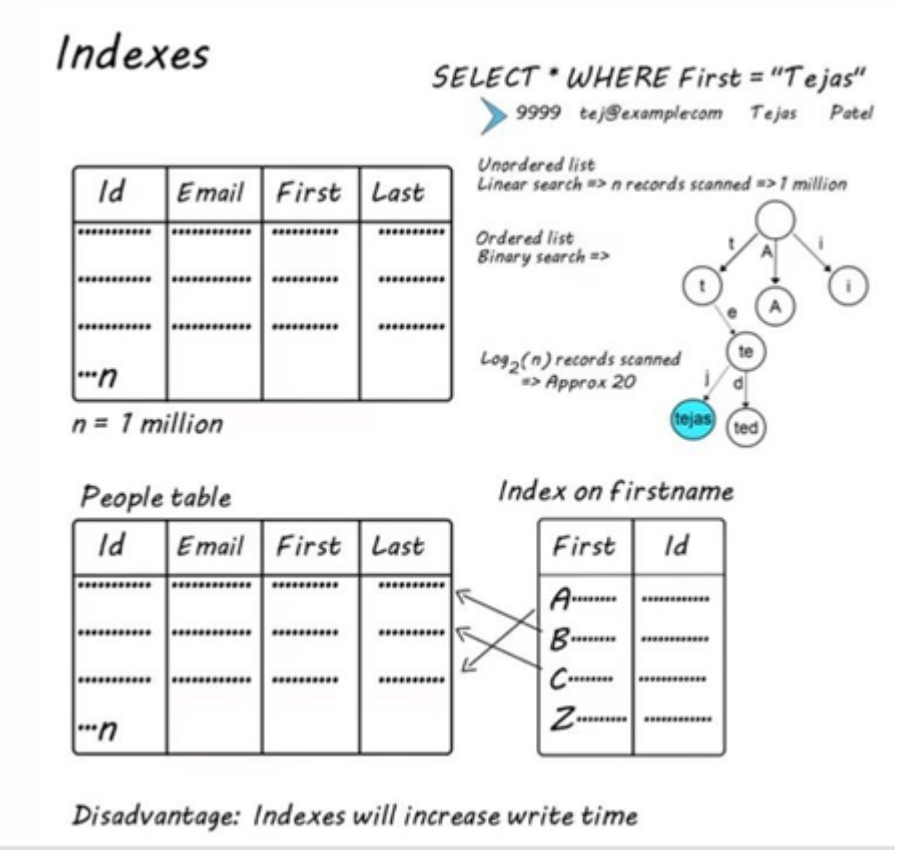
- no relational
- no RDBMS
- no only SQL

Document databases, Graph databases, Key-value databases, etc.

Data indexing

数据索引是一种用于加快检索速度的数据结构，但它需要额外的编写时间和存储空间。

Data indexing is a kind of data structure used for faster retrieving, but it needs extra time for writing and extra space for and storing.



B+ Tree

树的基本特征：

- 除根节点外，树中的每个节点都有一个父节点和零个或多个子节点。
- 没有任何子节点的节点称为叶节点。
- 非叶节点称为内部节点。
- 节点的子树由节点及其所有子节点组成。
- 如果叶节点处于不同的层级，则该树称为不平衡树。

Basic characteristics of Tree:

- Except the root node, each node in the tree has one parent node and zero or more child node.
- A node that does not have any child nodes is called a leaf node.
- A non-leaf node is called internal node.
- A sub-tree of a node consists of the node and all its descendants.
- If the leaf nodes are at different levels, the tree is called unbalanced.

B+ 树：

- 数据指针只存储在树的叶节点上
- 内部节点中的指针是树指针
- 叶节点中的指针是数据指针
- 内部节点中没有数据指针，因此留给树指针的空间更大，层次更少。
- 叶节点连接在一起，提供有序的数据访问。

B+ tree:

- data pointers are stored only at the leaf nodes of the tree
    - The pointers in internal nodes are tree pointers
    - The pointers in leaf nodes are data pointers
- There's no data pointer in internal nodes, so more space is left for tree pointer thus fewer levels.
- Leaf nodes are linked together to provide ordered access to data.

对于阶数为 p 的 B+ 树

- 每个节点最多有 p 个指针
- 内部节点至少有ceil（p/2）个树指针
- 每个叶节点至少有 ceil((p-1)/2) 个值。

For B+ tree with order p:

- Each node has at most p pointers
- Internal nodes have at least ceil(p/2) tree pointers
- Each leaf nodes has at least ceil((p-1)/2) values.

B+ 树形插入：

插入满叶节点时，记住两个步骤：

1. 复制中间值并将其插入父节点。
2. 将节点拆分为 2 个新的叶节点，中间值移至右侧。

B+ Tree insertion:

When inserting into full leaf node, remember 2 steps:

1. Copy the middle value and insert it to parent node.
2. Split the node to 2 new leaf nodes, the middle value go to right side.

Insertion Samples

Insertion Tutorial

B+ 树删除：

删除时，考虑该节点是否在其父节点中。 如果是，则考虑:

- 删除后，该叶节点是否为半满?
  - 是，使用右侧值替换父节点中删除的值
  - 否，同级节点是否超过半满?
    - 是，重新分配兄弟节点值和本节点值，创建一个新的父节点
    - 否，将此节点与其同级节点（先右侧）合并为一个新节点，修复其父节点。

B+ Tree Deletion:

When deleting, consider whether this node is in its parent node. If yes, consider:

- after deletion whether this leaf node is half-full?:
  - yes, use the right side value to replace the value deleted in parent node
  - no, whether sibling node is more than half-full?
    - yes, redistribute sibling node values and this node value, create a new parent node
    - no, merge this node with its sibling node(right side first) to a new node, repair its parent node.

Deletion Samples

Deletion Tutorial

注：分割非叶节点时，无需复制中间值

Note:

1. when spliting non-leaf node, no need to copy the middle value
2. Each value can't exist in non-leaf nodes twice

8. Data Querying

数据查询既可以是操作查询，也可以是选择查询：

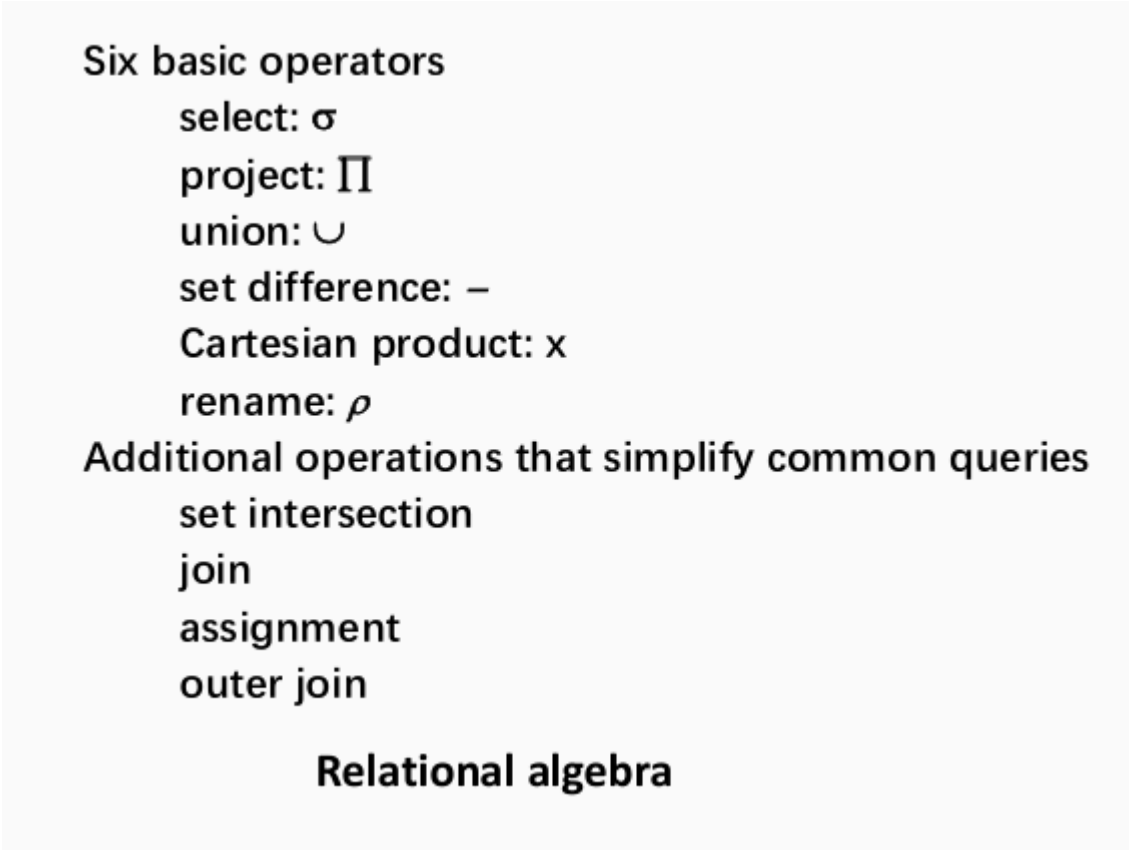
- 选择查询是从数据库中检索数据的查询。
- 操作查询要求对数据进行其他操作，如插入、更新、删除或其他形式的数据操作。

A data query is either an action query or a select query：

- A select query is one that retrieves data from a database.
- An action query asks for additional operations on data, such as insertion, updating, deleting or other forms of data manipulation.

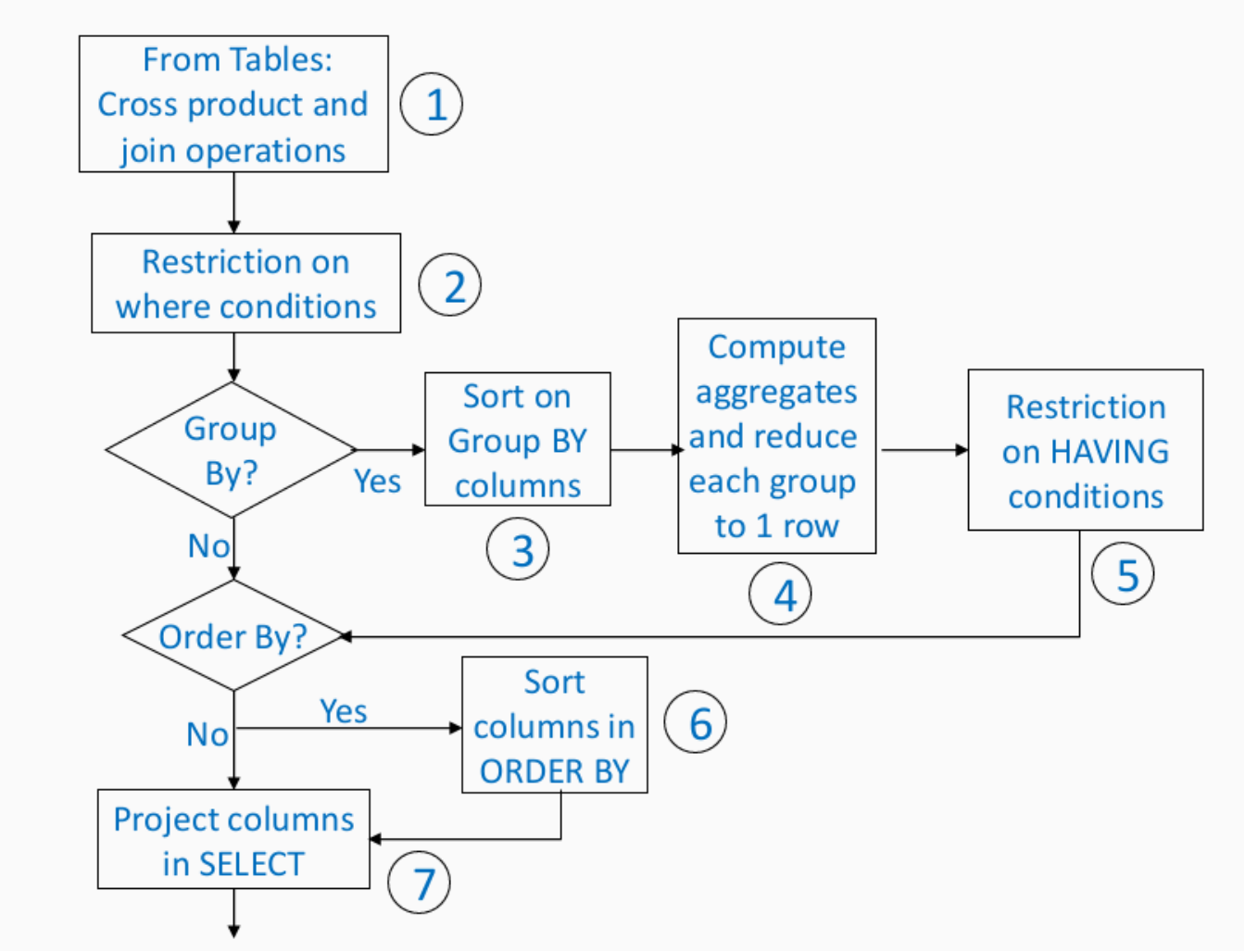
Relational Algebra

Use operators to do select queries in database.


Six basic operators
select: σ
project: Π
union: ∪
set difference: –
Cartesian product: x
rename: ρ
Additional operations that simplify common queries
set intersection
join
assignment
outer join
Relational algebra

Algebra

SQL select



Select

NoSQL querying

E.g., MongoDB querying: Mongo query language

- Targets a specific collection of documents
- Specifies criteria that identify the returned documents
- May include a projection to specify returned fields • -
- May impose limits, sort, orders

- E.g.,

```
db.inventory.find({ type: "snacks" })
```

All documents from collection inventory where the type field has the value snacks

```
db.inventory.find({ type: { $in: [ 'food', 'snacks' ] } } )
```

All inventory docs where the type field is either food or snacks

```
db.inventory.find( { type: 'food', price: { $lt: 9.95 } } )
```

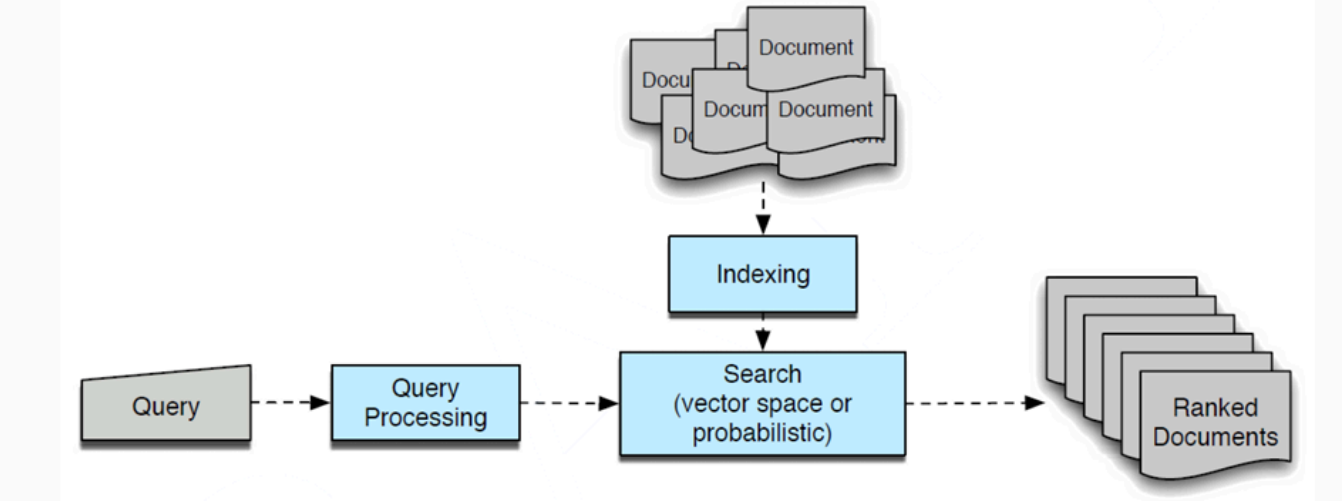
All ... where the type field is food and the price is less than 9.95

9.10.data-driven applications

Information retrieval

- 它是 "搜索"：主要是搜索文件，但也可以是其他文件。
- 它是一门计算机科学学科，设计并实施算法和工具，帮助人们找到想要的信息。
- It is 'search': mostly searching for documents, but can also be others.
- It is a computer science discipline that designs and implements algorithms and tools to help people find information that they want.

Process:



1. 产生查询需求
2. 根据索引和查询关键词之间的相关性，通过特殊算法对存储的文档进行索引和搜索。
3. 返回排序文档
4. Need generated so query
5. Stored documents are indexed and searched by special algorithm, based on the relevance between index and query key words.
6. Return ranked documents

如何找到相关性?

- 通过关键词匹配：布尔模型
- 通过相似性：向量空间模型
- 通过想象查询是如何产生的：概率模型和语言模型
- 通过其他网页的想法：网页链接权重
- 通过用户资料：了解用户的行为和反馈

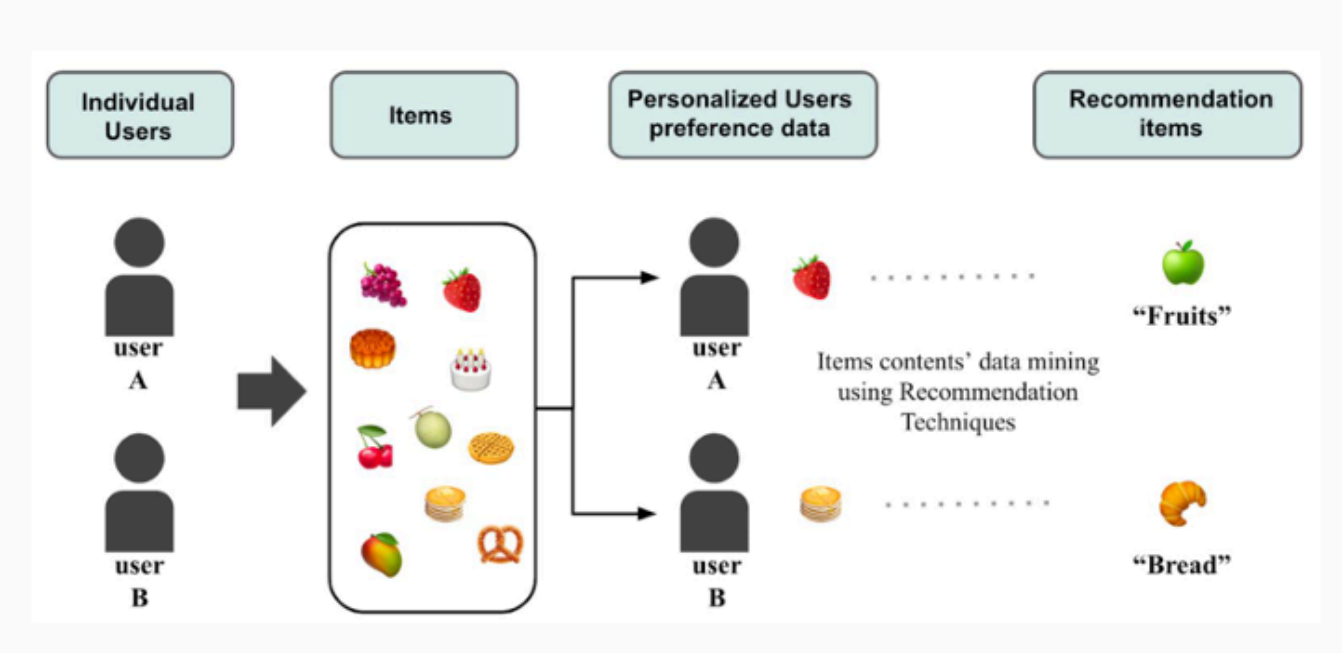
How to find relevance?

- by key word matching: boolean model
- by similarity: vector space model
- by imagining how the query comes: probability model and language model
- by other pages' thoughts: webpages link weight
- by users profile: learn users' action and feedback

Recommender systems

它是信息过滤系统的一个子类，根据用户可能感兴趣的内容向用户提供建议。

A subclass of information filtering system, it gives suggestions to users based on their possible interest.



Collaborative filtering

- 基于项目/用户的协同过滤：查找相似项目/用户并推荐相似项目。

  矩阵因式分解：使用向量表示用户和项目，计算他们的偏好分数。
- Item/User based collaborative filtering: find similar items/users and recommend similar items.
- Matrix factorization: use vector to represent users and items, calculate their preference score.

基于项目/用户的协同过滤步骤：

根据与活跃用户的相似度对所有用户加权

选择部分邻居作为预测因子

根据邻居的评分进行归一化处理并计算得分

根据预测得分进行排名

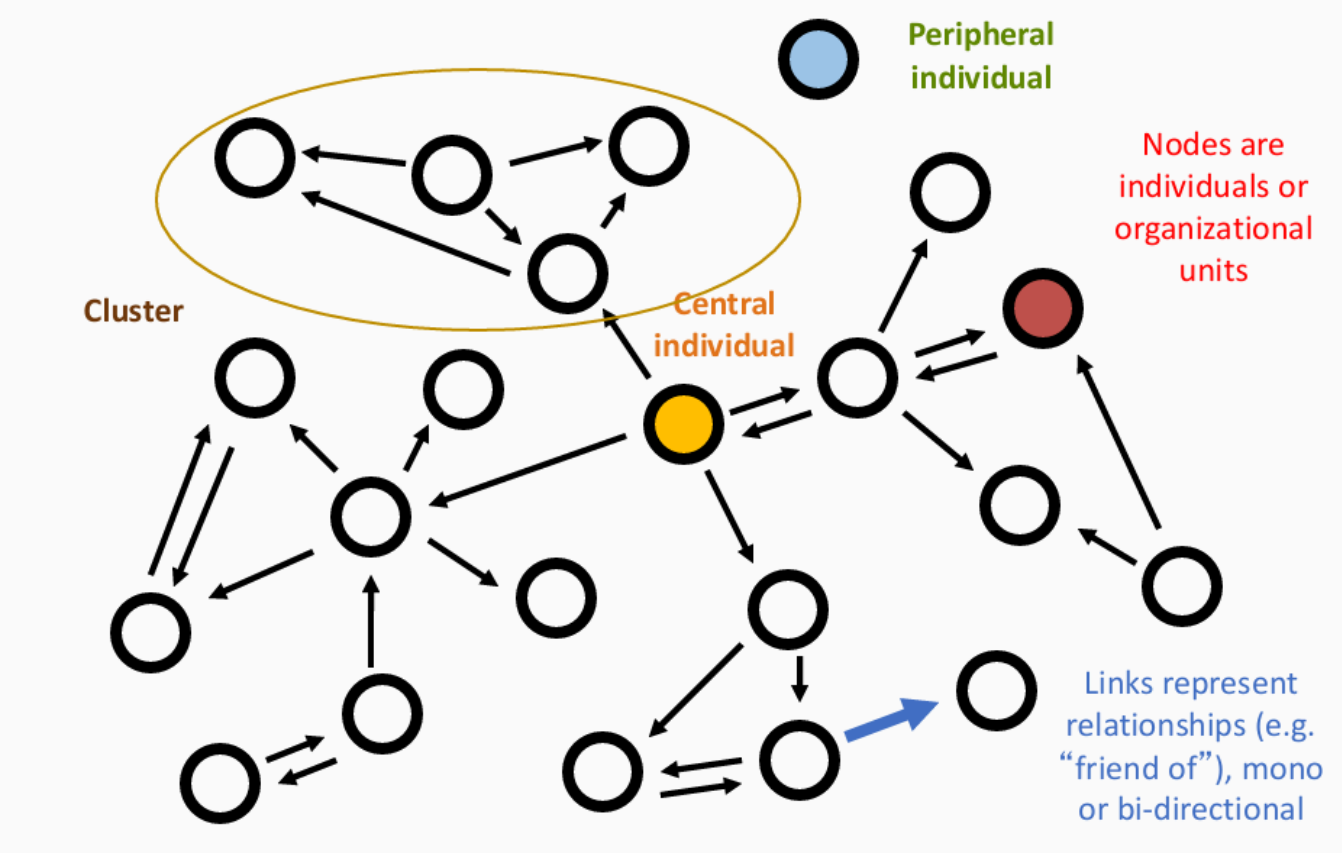
Item/User based collaborative filtering steps:

1. Weight all users by similarity with active user
2. Select some neighbors as predictors
3. Normalize and calculate prediction based on neighbors' ratings
4. Rank items based on predicted score.

Social network analysis

图由节点和边组成，我们可以用它来表示社交网络

Graphs(a branch of discrete mathematic) are consist of nodes and edges, we could use it to represent social network



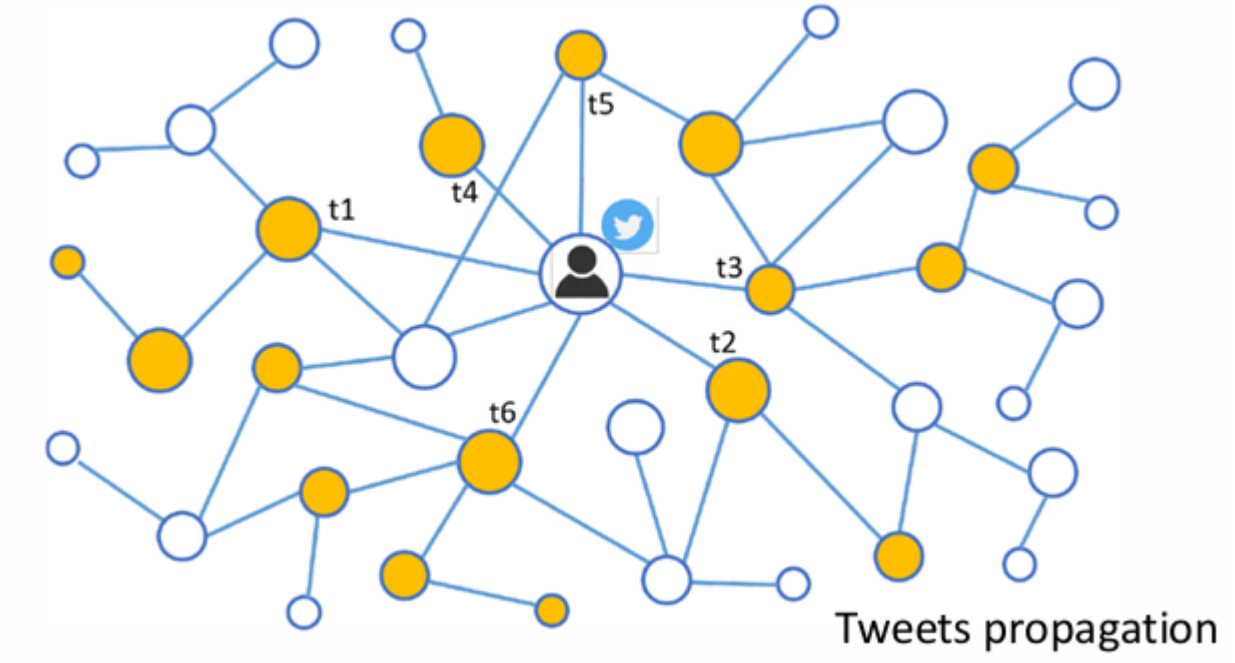
Network Propagation

一类将输入数据信息整合到给定网络中各连接节点的算法。

并研究输入数据如何影响整个网络的现有数据

A class of algorithms that integrate information from input data across connected nodes in a given network.

And study how input data affect existing data across network



Anomaly detection

检测不常见的事物，包括项目、事件和用户。

detect uncommon things including items, events and users.

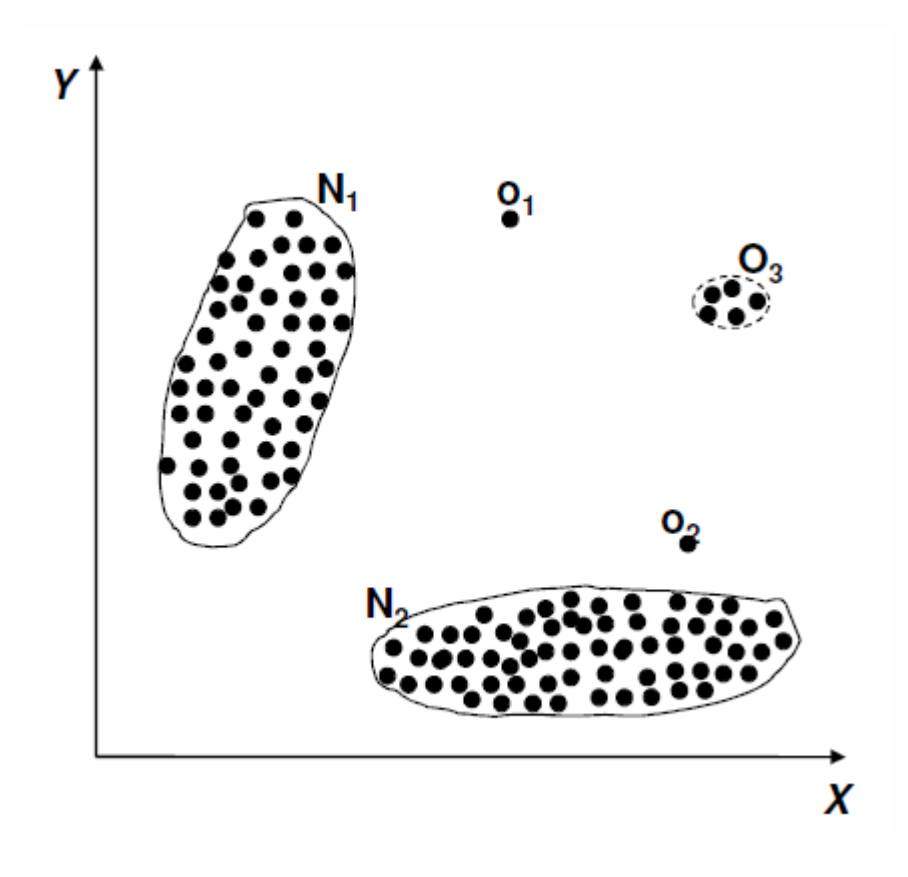
从历史上看，我们使用统计数据来查找和剔除异常值。

Historically, we use statistics to find and remove outliers.

异常的结构：

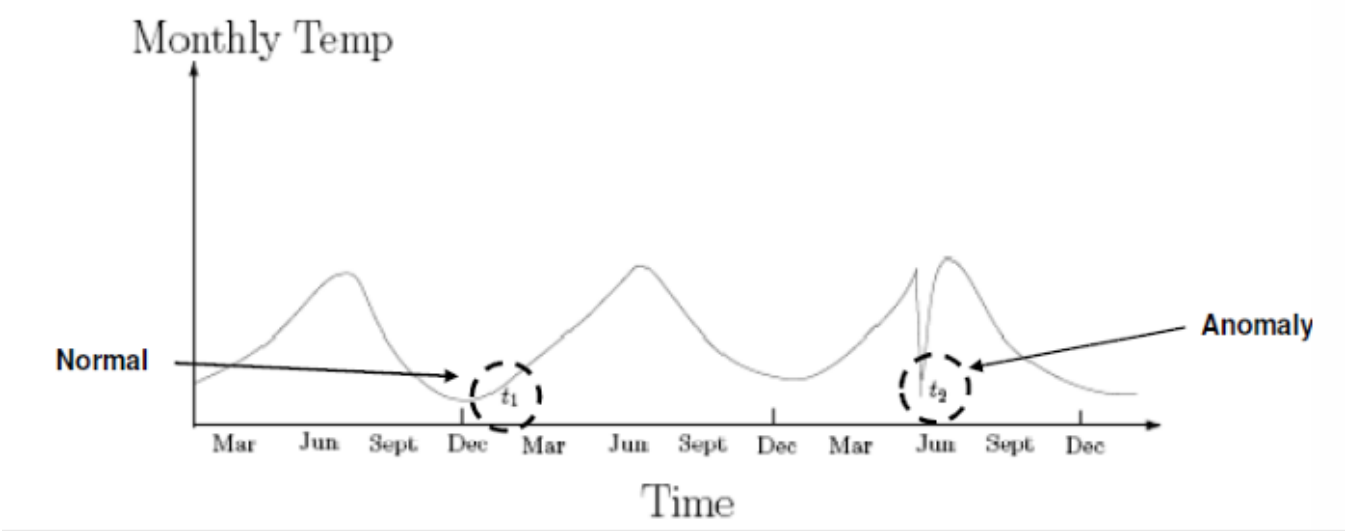
Structure of anomalies:

1. 点异常：群体外的单个数据实例
2. Point anomalies: an individual data instance out of groups



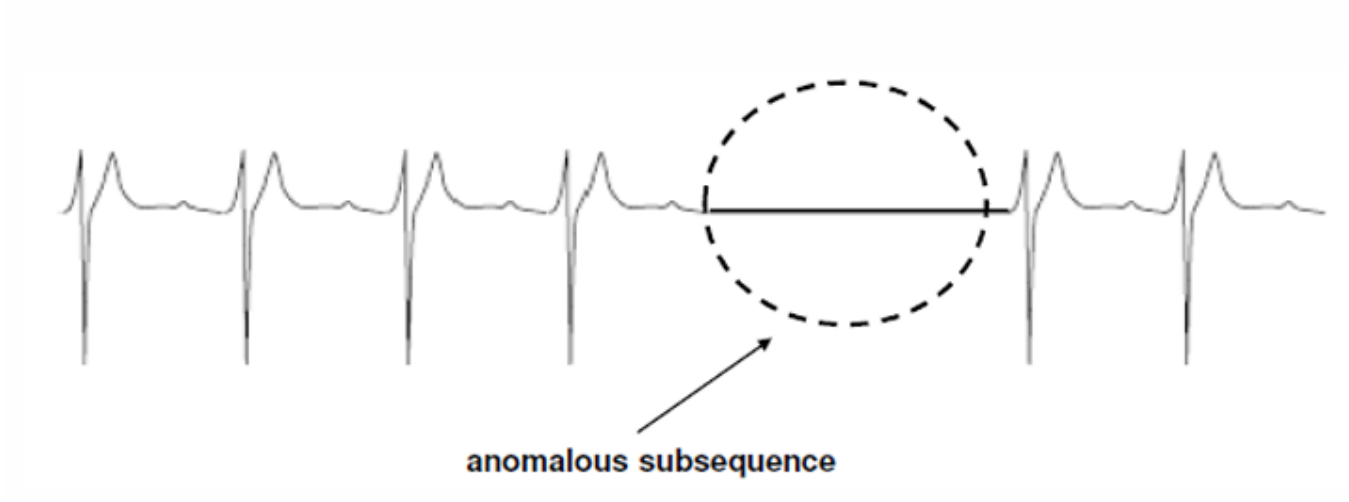
2. Contextual anomalies： an individual data instance is anomalous based on content

3. 上下文异常: 单个数据实例的内容异常



3. Collective anomalies: a collection of data is anomalous based on relationship

4. 集体异常：基于关系的数据集合异常



Data labels for anomalies detection:

根据是否有标签，我们有

- 有监督的异常检测： 正常数据和异常数据都有标签，我们可以将其视为分类问题。 如果异常数据很少，则尝试通过数据增强来生成异常数据。
- 半监督异常检测： 只有正常数据才有标签，我们可以学习正常模式的样子，如果任何数据不符合该模式，我们就将其视为异常数据。
- 无监督异常检测： 没有标签，我们需要进行一些统计，如计算平均值和方差，以推断哪些数据是异常数据。

based on whether there is label, we have:

- Supervised anomaly detection: Both normal data and anomalous data have labels, we could take it as classification problem. If anomalous data is rare, try to generate by Data augmentation
- Semi-supervised anomaly detection: Labels available only for normal data, we could learn how normal pattern looks like, if any data doesn't follow that pattern, we treat them as anomalous data.
- Unsupervised anomaly detection: No labels, we need to do some statistics such as calculate mean and variance to infer what data is anomaly.

11.Data Management

什么是数据管理：

数据管理是指安全、高效、经济地收集、保存和使用数据。

数据管理变得越来越重要，因为数据的创建和消耗速度都非常快。

What's data management:

Data management is the practice of collecting, keeping, and using data securely, efficiently, and cost effectively.

Data management has become more and more important since data is creating and consuming at a rapid rate.

数据质量维度：

准确性：数据应反映真实情况，并可验证

完整性：数据应提供所有必要的值。

一致性： 跨网络传输时，数据不应发生变化。

有效性： 数据收集应遵循正式规则，并以正确的格式和范围存储。

唯一性： 数据不应重复或重叠，数据清理或重复数据删除可帮助解决问题。

及时性： 数据应可随时访问和使用，并及时更新。

Data Quality Dimensions:

Accuracy: data should reflect real scenarios, and could be verified

Completeness: data should provide all the required values.

Consistency: When transfered across network, data shouldn't change.

Validity: Data should be collected with formal rule, and be stored in the right format and range.

Uniqueness: Data shouldn't repeat or overlap, data cleaning or data deduplication could help solve the problem

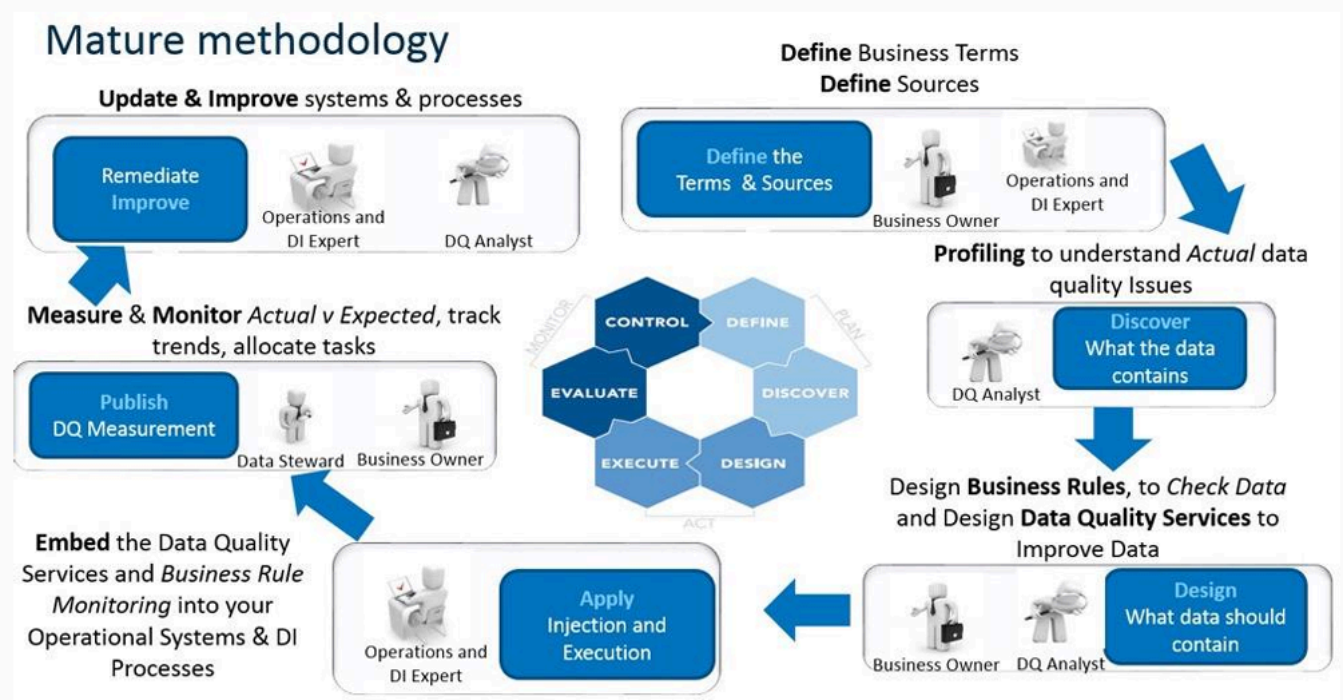
Timeliness: Data should be accessible and available at anytime, and be updated in time.

如何提高数据质量?

- 数据预处理
- 数据标准化： 使来自不同数据集的数据具有相同的格式。
- 数据质量监控： 经常检查数据质量，尝试将质量监控软件与机器学习相结合，自动监控数据质量。

How to improve data quality?:

- Data preprocessing
- Data Standardization: Make data from different sets into the same format.
- Data quality monitoring: Frequently check data quality, try to combine quality monitoring software with machine learning to automatically monitor data quality.



数据安全：

数据安全的目的是在数据的整个生命周期内保护数据免遭损坏、窃取和未经授权的访问。

我们可以使用安全工具和服务，通过掩码和加密等方式保护数据。

Data security:

Data security's purpose is to protect data from corruption, stealing and unauthorized access throughout its whole life cycle.

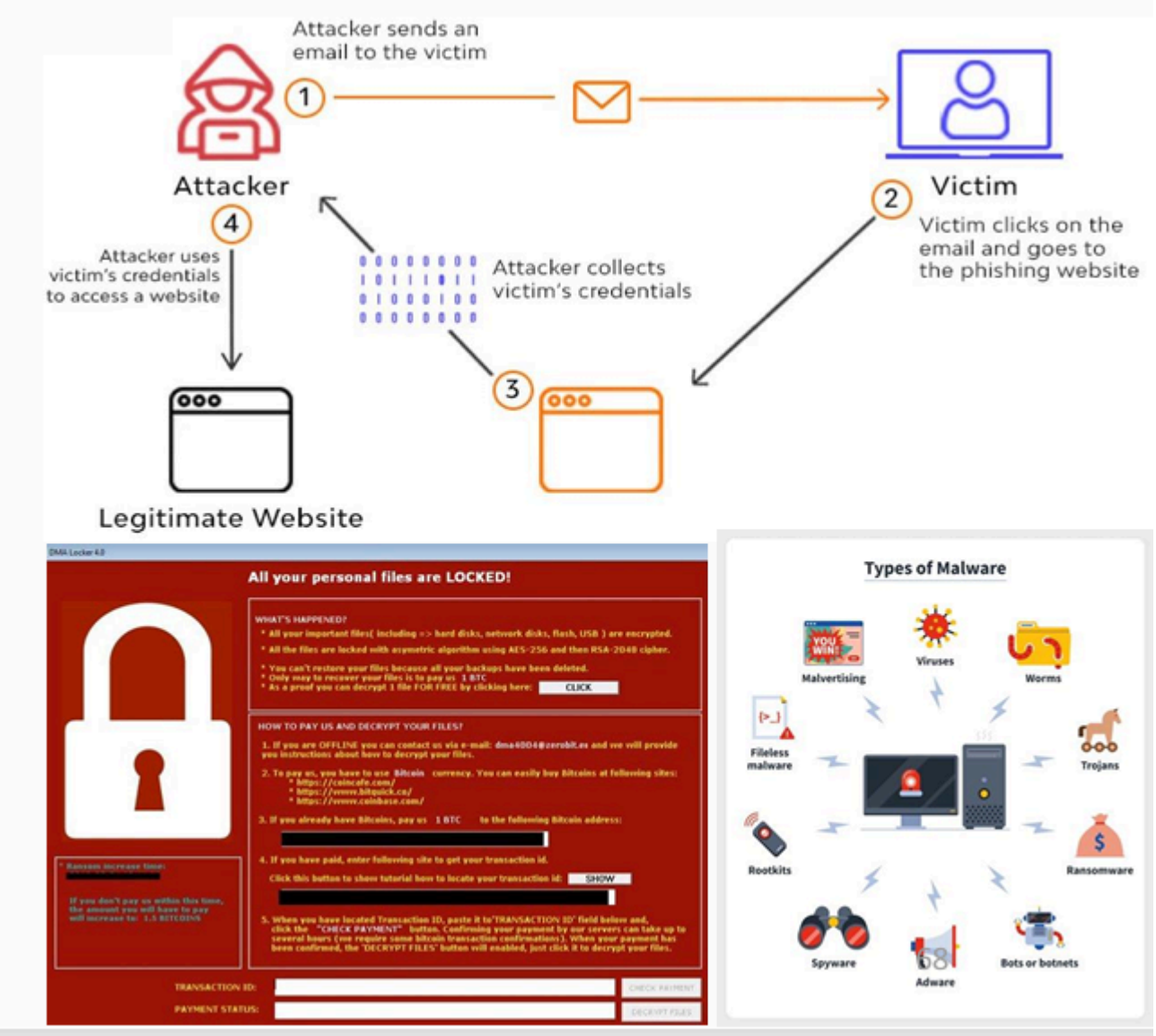
We could use security tools and services to protect data by masking and encryption etc.

一些数据安全风险：

- 网络钓鱼攻击：网站或电子邮件中包含的恶意链接假装成正常链接，如果用户点击，很容易窃取隐私数据。
- 恶意软件： 通过网络或电子邮件传播，可能感染整个公司的计算机或网络，导致严重的安全事件。

Some data security risks:

- Phishing attack: malicious links contained in websites or emails which pretend to be normal ones, if user clicks it, it could easily steal privacy data.
- Malware: Spread through Web or emails, could infect computers or network of the entire company, lead to serious security event.



如何防止?

- 数据屏蔽：通过遮盖和替换特定字母或数字来隐藏数据
- 数据加密：使用算法扰乱数据并隐藏其真实值。

How to prevent?

- Data masking: hide data by obscuring and replacing specific letters or numbers
- Data encryption: use algorithms to scramble data and hide its true value.

数据隐私：

个人应决定其个人信息在多大程度上可被他人访问或使用。

主要重点是第三方应如何收集、储存、管理和使用个人数据，法律应对此做出规定

Data privacy:

A person should decide to what extend its personal information could be visited or be used by others.

The mainly focus is on how personal data should be collected, stored, managed and used by the third parties, which should be provided by laws

A method to preserve: Data perturbation

User deliberately provide data with some noise. But this method may cause lower performance.