2022 秋季学期《大数据分析》期末复习

from 20 级 LeeHITsz

（一）数据分析必备技能

1. 常见的数据来源与分类

- 商业数据:

  贸易数据、财务数据、金融数据

- 政府数据:

  税务数据、经济数据、居民数据、银行数据

- 科学数据:

  实验数据、观测数据、地球数据、空间数据、气象数据、基因数据

- 人类行为数据:

  交通数据、网络数据、医学数据、消费数据、人口学数据。

2. 数据相似度的度量[1-3]

- 余弦距离:

$$sim\left(\vec{x},\vec{y}\right)=\cos\left(\vec{x},\vec{y}\right)=\frac{\vec{x}\cdot\vec{y}}{|\vec{x}|\cdot|\vec{y}|}$$

- 皮尔逊相关系数（调整余弦距离）:

$$sim\left(\vec{x},\vec{y}\right)=\frac{\sum_{k=1}^{n}\left(x_k-\overline{x}\right)\left(y_k-\overline{y}\right)}{\sqrt{\sum_{k=1}^{n}\left(x_k-\overline{x}\right)^2}\sqrt{\sum_{k=1}^{n}\left(y_k-\overline{y}\right)^2}}$$

- 斯皮尔曼相关系数:

$$sim(\vec{x},\vec{y}) = 1 - \frac{6\sum_{k=1}^{n}(rank(x_k) - rank(y_k))^2}{n(n^2-1)}$$

- 欧式距离：

$$sim(\vec{x},\vec{y}) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

- 马氏距离：

$$sim(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T \Sigma^{-1}(\vec{x}-\vec{y})}$$

3. 预测与分类的概念

- 缺失值的填充可视为预测任务
- 预测任务的类型：
  - 预测实际值：回归
  - 预测 YES/NO 值：二元分类
  - 预测类别：分类
- 分类的过程：
  ① 查找描述实体的特征
  ② 使用需要预测的类别示例
  ③ 学习一个预测的模型（函数）

4. 深度学习的概念

机器学习系统使用多层神经网络，并在大量数据上进行训练。

5. 网络图与节点的重要性

网络图中最重要的节点：**如果一个节点被其他重要节点指向，则该节点很重要**。

6. 数据挖掘的概念

- 数据挖掘是对观察到的数据集（海量数据）进行分析，以发现未预料到的关系，并且用对数据分析师来说既易于理解又有用的、新颖的方式来总结数据。
- 数据挖掘是为数据发现模型：
  - 解释数据的模型
  - 预测未来数据实例的模型
  - 汇总数据的模型
  - 提取数据中最突出的特征的模型
- 数据挖掘是关于收集、处理、分析数据，并从数据中获得有用见解的研究
- 从数据中提取有用信息是商业化数据的关键
- 由于**数据量大，特征维度多**，数据的数量和复杂性不允许手动处理数据，需要自动化技术，不能使用传统的分析方法。

7. 数据科学、大数据、AI/机器学习/深度学习的侧重点

- 数据科学：专注于更直接的应用和洞察。
- 大数据：面向更多系统
- AI/机器学习/深度学习：更重视科学上的突破。

8. 数据主权的概念

数据主权是与领土、领海和领空权等相当，是信息和大数据时代的新权利范畴。他是指网络空间中的国家主权，是一个国家对本国数据进行管理和利用的独立自主性，不受他国干涉和侵扰的自由权，

体现了国家作为控制数据权的主体地位。数据主权包括数据所有权和数据管辖权两个方面，所有权是国家对于本国数据排他性占有的权利，管辖权是国家对其本国数据享有的管理和利用的权利。

9. 中央处理器（CPU）的主要参数

(1) **主频**:

CPU 内核的时钟频率，即 CPU 运算时的工作频率。虽然 CPU 的主频不代表 CPU 的速度，但提高主频对于提高 CPU 的运算速度非常重要，可以减少运算的时钟周期占用时间。

(2) **核数**:

一个 CPU 芯片上核心的数量，用来完成所有的计算、接受存储命令、处理数据等，是数字处理核心。

(3) **线程**:

CPU 线程数是指逻辑上的处理单元数，即模拟出的 CPU 核心数，在 Intel 超线程技术下，一个 CPU 可以模拟出多于核数的线程数。线程数越多，CPU 能同时并行处理的任务数越多，能够提高处理器运算部件的利用率和运算效率。

(4) **架构**:

CPU 的架构简单来说是 CPU 核心的设计方案，就如房屋的布局一样，架构的设计对内存缓存访问，各核心间数据交换等都有影响。架构越先进，相同频率下 CPU 处理效率就越高。

(5) **缓存**:

CPU 缓存是用于减少处理器访问内存所需平均时间的部件，其容量小于

内存但速度接近处理器的顷率。由于缓存的运行效率极高，缓存容量的增大，可以大幅度提升 CPU 内部读取数据的命中率，而不用再到内存或者硬盘上寻找，以此提高系统性能。

10.物理 CPU 数、核数与线程的概念[4]

- 物理 CPU 个数:

  指主板上实际插入的 CPU 硬件个数。(这一概念经常被泛泛的说成是 CPU 数,容易与核数和线程数等概念混淆,所以此处强调是物理 CPU 数)

- 核数:

  早期的 CPU 只有一个核（core)，对操作系统而言，也就是只能同时运行线程。为了提高性能，CPU 厂商开始在单个物理 CPU 上增加核数，所以，就出现了双核和多核 CPU，这样的 CPU 就可以同时运行两个或更多线程，有几个 core 就可并行运行同样数量的线程。

- 线程（多线程技术和超线程技术):

  运行中线程在等待的时候运行其他线程，这样提高 CPU 的并行性能。本质一样，是为了提高单个 core 同时执行多线程数的技术，充分利用单个 core 的计算能力。

11.线程与进程的对比

线程	进程
线程没有数据段或堆	一个进程中含有代码、数据、堆和其他数据段
线程不能独立存在，它必须存在于进程中	一个进程中必须至少包括一个线程
进程中可以不止有一个线程，其中第	一个进程中的线程共享代码、数据、

一个线程称作主线程，拥有进程的堆栈	堆，共享 I/O，但是每个线程有独立的堆栈与寄存器
创建代价较小	创建代价较大
切换代价较小	切换代价较大
如果线程结束，它的堆栈将被回收	如果一个进程结束，它的资源将被回收，并且所有线程都将结束

12.分布式计算与并行式计算的对比

	分布式计算	并行式计算
概念	在分布式计算中，许多统一的计算机通过消息传递相互通信，同时致力于一个共同的任务	在并行计算中，一个任务被划分为多个子任务，然后分配给同一个 CPU 上的不同处理器
涉及计算机系统数量	多个物理计算机系统存在于同一个计算机系统中	一个物理计算机系统承载多个处理器
进程间的依赖关系	进程之间可能没有太多的依赖性	过程之间有更多的依赖性。一个的输出可能是另一个的输入
可伸缩性	系统很容易扩展，因为没有限制可以添加多少系统到一个网络。	实现并行计算的系统具有有限的可伸缩性
资源共享	系统有自己的内存和处理器	所有处理器共享相同的内存
同步	网络中的计算机必须执行同步算法	所有处理器使用相同的主时钟进行同步
使用	通常在需要高可伸缩性的地方首选	在需要更快的速度和更好的性能的地方通常是首选

（二）数据属性

1. 数据的度量水平[5]

(1) 二元数据：取值仅有两个类别的定类数据

- 例如：考试结果、归类、调查结果、事件发生与否、是否具备某属性

(2) 定类数据：给数据定义一个类别。这种数据类型能将所研究的对象区分开，**没有等级或好坏的差异**。

- 例如：血型、类别、材料属性、颜色种类

(3) 定序数据：不仅能够代表事物的分类，还能代表事物按某种特性的排序，但**各个定序变量的值之间没有确切的间隔距离**，只能排列出它们的顺序，而不能反映出大于或小于的数量或距离。

- 例如：收入水平、喜爱程度、信用卡分级、酒店星级

(4) 定距数据：定距变量的值之间可以比较大小，两个值的差有实际意义。**加减运算有效，但乘除运算无意义。有单位，但无绝对零点**。

- 例如：温度、时间

(5) 定比数据：数据的最高级别。在定距数据的基础上，**具有绝对零点，由此也可以做乘除运算**。

- 例如：质量、压强、能量、长度、电荷

2. 结构化数据、半结构化数据与无结构数据的对比

	结构化数据	无结构数据	半结构化数据
特点	(1) 预定义数据类型	(1) 没有预定义的数	(1) 松散组织的元级

	(2) 容易搜索 (3) 基于文本 (4) 展示了发生了什么	据类型 (2) 不易搜索 (3) 基于文本，pdf，图片、视频 (4) 展示了原因	结构，可以包含非结构化的数据 (2) 基于 html，xml，json
存储位置	关系型数据库、数据仓库	应用程序、数据仓库、数据湖、云文件夹	关系型数据库、Tagged-text 格式
存储格式	行、列	各种形式	摘要与图形
举例	日期，电话号码，社保号，客户姓名，交易信息	文件、电子邮件和信息、对话记录、图像文件、开放式调查的答案	服务器日志、按标签组织的推文、按文件夹分类的电子邮件

3. 数据分箱[6-7]

- 等距分箱：根据测量的距离均等划分类别。
  - 问题：有些分箱可能会非常稀疏
- 等频分箱：根据样本的数量均等划分类别。
  - 问题：有些分箱可能会非常小
- 指数等距分箱：以 10 的幂（或其他常数的幂）作为区间划分点。
  - 适用于测量范围跨度很大的数据
- 聚类分箱：根据数据聚集情况进行分箱。

4. 数据对象与数据属性的概念

- 属性是对象的特征，也称特征、变量、域等。

- 属性都有特定的值。
- 对象由一组属性组合描述，又称记录、数据点、样本、实例、元素等。

5. 数据矩阵化（向量化）

(1) 关系数据的矩阵化：

概念：如果数据可以用固定的属性描述，数据对象就可以用多维空间的数据点或者向量定义，每个维度代表一种特征。数据集表征为 n 行 k 列的数据矩阵，其中 n 为样本总数，k 为特征总数。

例如：

	Temperature	Humidity	Pressure
O1	30	0.8	90
O2	32	0.5	80
O3	24	0.3	95

(2) 图像的矩阵化[8]：

灰度图矩阵：$F(n_x, n_y)$

彩色图矩阵：$C(n_x, n_y, 3)$

其中 n_x, n_y 分别为两个维度上像素的个数，矩阵中元素取值为 $[0, 255]$，对于灰度图来说，0 表示黑色，255 表示白色。

(3) 时间序列的矩阵化：

p 阶自回归模型：

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$

可以改写为矩阵形式的一阶自回归模型：

$$\xi_t = F\xi_{t-1} + \varepsilon_t$$

$$\begin{bmatrix} y_t \\ y_{t-1} \\ y_{t-2} \\ \vdots \\ y_{t-p+1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \dots & \phi_{p-1} & \phi_p \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \\ y_{t-3} \\ \vdots \\ y_{t-p} \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

(4) 文本数据的矩阵化：

文本文件可用词汇组合的频率代表，词汇代表维度，出现的次数代表值。

特点：多个文本可合成矩阵；稀疏矩阵很多；可以任意排列。

例如：

Doc Id	Words
1	the, dog, follows, the, cat
2	the, cat, chases, the, cat
3	the, man, walks, the, dog

转化为：

Doc Id	the	dog	follows	cat	chases	man	walks
1	2	1	1	1	0	0	0
2	2	0	0	2	1	0	0
3	1	1	0	0	0	1	1

(5) 购物数据的矩阵化：

与文本文件的矩阵化类似。例如：

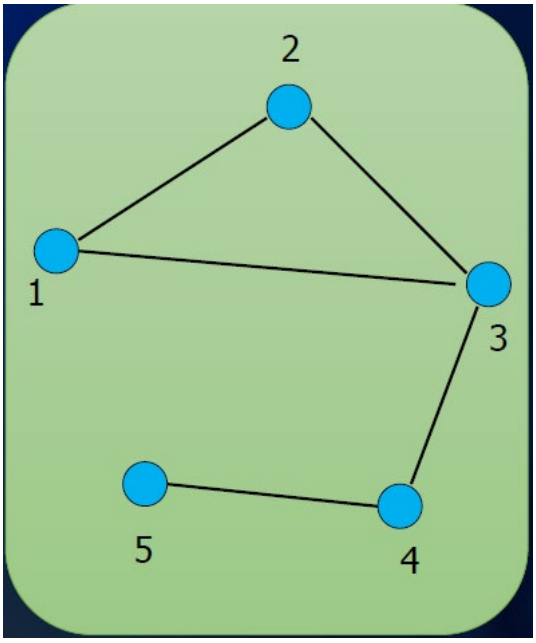
TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

转化为：

TID	Bread	Coke	Milk	Beer	Diaper
1	1	1	1	0	0
2	1	0	0	1	0
3	0	1	1	1	1
4	1	0	1	1	1
5	0	1	1	0	1

(6) 图形数据的矩阵化（构建邻接矩阵）：

例如：



对应的邻接矩阵为：

$$A=\begin{bmatrix}0&1&1&0&0\\1&0&1&0&0\\1&1&0&1&0\\0&0&1&0&1\\0&0&0&1&0\end{bmatrix}$$

注意：无向图的邻接矩阵是对称矩阵；有向图的邻接矩阵可以不是对称矩阵。

6. 文本挖掘的基本流程

(1) 初步统计词汇

- 简单处理：去除标点符号，调整为小写字体，删除空白等；
- 分割为词语，保留最频繁用语。

(2) 删除停用词[9]：

- 停用词：在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言数据（或文本）之前或之后会自动过滤掉某些字或词，这

些字或词即被称为停用词。这些停用词都是人工输入、非自动化生成的，生成后的停用词会形成一个停用词表。但是，并没有一个明确的停用词表能够适用于所有的工具。甚至有一些工具是明确地避免使用停用词来支持短语搜索的。

(3) 按算法排列

7. 关联性数据

- 有序或时序数据：时间上前后关联。
  - 例如：基因序列、股票价格、计算机日志
- 空间数据：空间位置存在相对关联性
  - 例如：福岛核泄漏之后的辐射强度分布
- 时空数据：数据存在时空关联性
  - 例如：大型天文望远镜拍摄的天文图像
- 网络或图形数据：数据点存在相互连接

8. 文本文件的类型

- 文本文件：只由字符原生编码组成，人类可读。
- 二进制文件：由字符二进制编码组成，计算机可读，人类不可读。
- 富文本文件：具有风格、排版等信息，如颜色、式样（黑体、斜体等）、字体尺寸、特性。

9. 矩阵的存储格式

矩阵可以按行下标连续存储或按列下标连续存储。例如：

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

其存储方式可以是:

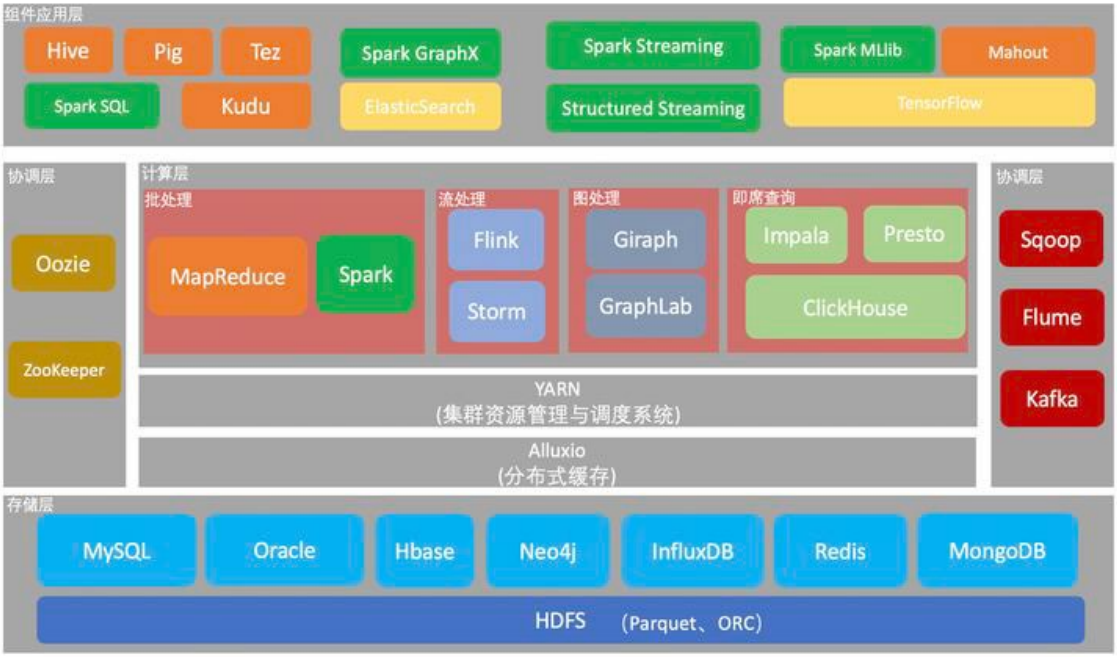
地址	行连续存储	列连续存储
0	a_{11}	a_{11}
1	a_{12}	a_{21}
2	a_{13}	a_{12}
3	a_{21}	a_{22}
4	a_{22}	a_{13}
5	a_{23}	a_{23}

对于多维数据: $N_1*N_2*N_3*...*N_d$,

行连续存储：最后一个维度的下标在内存中连续。

列连续存储：第一个维度的下标在内存中连续。

10.大数据架构图



（三）数据预处理

1. 数据分析与挖掘架构系统

(1) 需求调研

(2) 框架定位

(3) 数据准备

- 数据采集
- 数据库模型
- 数据入仓
- 数据预处理
  - 数据集成
  - 数据清洗
  - 数据规约
  - 特征提取
  - 数据转化

(4) 数据挖掘

(5) 测试评审

2. 抽样的概念

抽样是根据总体的子集（样本）统计数据，获取有关总体的信息的方法，无需调查所有个体。

3. 抽样的特性

- 相关性
- 可靠性

- 有效性

4. 抽样的原因

- 无法研究总体，综合考虑时间和资源限制
- 总体无法获得
- 样本能代表总体的特征
- 特殊抽样不可恢复

5. 抽样的步骤

(1) 明确抽样目的

(2) 定义总体的范畴

(3) 制定抽样条件

(4) 确定抽样大小

6. 概率抽样方法[10]

(1) 简单随机抽样：从总体中逐个抽取单元并且无放回，每次都在所有尚未进入样本的单元中等概率地抽取，直到单元全部抽完。

- 适用场景：当总体数量较小或者总体方差与任意局部方差基本相当的情况。

(2) 整群抽样：将总体的各单位归并为互不交叉、不重叠的群，然后以群为单位抽样。

- 适用场景：适用于群间差异小、群内各个体差异大、可以依据某种特征差异来划分的群体。

(3) 系统抽样：先将总体中的抽样单元按某种次序排列，在规定范围内随机抽取一个初始单元，然后按事先规定的规则抽取其他

样本单元。特别地，如果在抽取初始单元后按相等的间距抽取其余样本单元，则称为等距抽样 。

- 适用场景：适用于容量很大且个体的排列是按照随机顺序排列的总体。

(4) 分层抽样：先按照某种规则把总体划分为不同的层，然后在层内再进行抽样，各层的抽样之间是独立进行的。特别地，如果各层内是简单随机抽样，则称为分层随机抽样。

- 适用场景：适用于层间有较大的异质性，而每层内的个体具有同质性的总体

7. 非概率抽样方法[11]

(1) 便利抽样：抽样时，以方便为原则，容易获取的对象更大概率成为样本。

- 例如：某影评人为收集观众对某部电影的评分情况，可以随机在电影院出口进行抽样采访。

(2) 选择抽样：以采样者的主观经验选择总体中具有代表性的样本。

- 例如：对福建省旅游市场状况进行调查，有关部门选择厦门、武夷山、泰宁金湖等旅游风景区做为样本调查。

(3) 配额抽样：将总体样本按一定标志分类或分层，确定各类（层）单位的样本数额，在配额内任意抽选样本的抽样方式。

- 与分层随机抽样的区别在于，各层内部为调查人员主观选取样本

(4) 滚雪球抽样：先随机选择一些被访者并对其实施访问，再请他们提供另外一些属于所研究目标总体的调查对象，根据所形成的线索选择此后的调查对象，往往用于对稀少群体的调查。

- 例如：小红被老师提问，小红回答错误后，推荐小丽回答。

8. 数据抽样存在误差的原因

- 样本选择偏差：样本被选的概率不同。
- 抽样误差：样本无法代表总体。采样方案的设计或者样本数不够大，很难完全反映总体的特征。

9. 数据集成的概念

- 数据集成是将互相关联的分布式异构数据源集成在一起，使用户能够以透明的方式访问数据源。
- 集成是指维护数据源整体的一致性，提高信息共享利用的效率;
- 透明的方式是指用户无需关心如何实现对异构数据源数据的访问,只关心以何种方式访问何种数据。

10.数据集成的难点

- 异构性：被集成的数据源通常是独立开发的，数据模型异构，给数据集成带来很大困难。

——数据语义、相同语义数据的表达形式、数据源的使用环境等。

- 分布性：数据源是异地分布的，依赖网络传输数据，这就存在网络传输的性能和安全性等问题。
- 自治性：各个数据源有很强的自治性，它们可以在不通知集成系统的前提下改变自身的结构和数据，给数据集成系统的鲁棒性提出挑战。

11.数据集成的模式

- 数据仓库模式：解决了数据异构性的问题，但是很难应对经常更新的数据，ETL（Extract、Transform、Load）需多次重复以保持数据的同步性。
- 中介模式：直接访问元数据。
- 数据集成模式：无需备份数据或者衍生数据。

12.数据规约的概念

数据规约就是缩小数据挖掘所需的数据集规模。

13.数据规约的作用

- 降低无效或错误数据对建模的影响，提高建模的准确性；
- 通过采用规模小且具有代表性的数据，降低数据挖掘时间；
- 节约数据存储的成本。

14.数据规约的方法

(1) 维度规约（属性规约）：减少所需自变量的个数

- 例如：小波变换、主成分分析、奇异值分解、特征集选择

(2) 数量规约（数值规约）：用较小的数据表示形式替换原始数据

- 例如：合并、聚类、抽样、参数回归

(3) 特征值规约：映射到特征空间

- 例如：回归、聚类、选样

15.量纲分析（无量纲化）[12-15]

- 概念：通过一个合适的变量替代，将一个涉及物理量的方程的部分或全部的单位移除，以求简化实验或者计算的目的，是科学研究中一种重要的处理思想。

下面，我们以流体力学中的纳维-斯托克斯方程（Navier-Stokes 方程）为例，介绍无量纲化的流程。

N-S 方程如下：

$$\frac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla\mathbf{u})=\frac{-\nabla p}{\rho}+\nu\nabla^2\mathbf{u}+\mathbf{g}$$

其中，ρ 为流体的密度，p 为流体的压强，$\boldsymbol{u}$ 为流体的速度向量，ν 是流体的运动粘度，$\boldsymbol{g}$ 为重力场，t 为时间，∇ 为梯度算子。

① 自由变量的选取与物理量的无量纲化：

常用的基本量纲为时间、长度、质量等。在这里，我们选取长度、密度和速率作为基本量纲，以长度L_0，密度ρ_0和速率u_0作为自由变量的常数。各个物理量的无量纲化如下：

$$\nabla=L_0^{-1}\hat{\nabla}$$
$$\rho=\rho_0\hat{\rho}$$
$$\boldsymbol{u}=u_0\hat{\boldsymbol{u}}$$
$$t=t_0\hat{t}$$
$$p=p_0\hat{p}$$
$$\nu=\nu_0\hat{\nu}$$
$$\boldsymbol{g}=g_0\hat{\boldsymbol{g}}$$

其中，t_0，p_0由L_0，ρ_0，u_0表示：

$$t_0=\frac{L_0}{u_0}$$
$$p_0=\rho_0 u_0^2$$

这里下标为 0 的物理量表示该物理量的某个有量纲数值，带^的物理量表示提取量纲后剩余的无量纲数值。值得注意的是，这里没有将ν_0，g_0用自由变量

的常数表示是因为，运动粘度与物质的种类相关，重力加速度与物体的位置有关，不是常数。

② 物理方程的无量纲化：

不难看出，N-S 方程中的每一项，量纲均为 $\frac{u_0^2}{L_0}$，因此，我们在原方程左右两侧同时乘以 $\frac{L_0}{u_0^2}$，并将各个物理量用无量纲化的形式代替，得到下式：

$$\frac{\partial \hat{\mathbf{u}}}{\partial \hat{t}}+\left(\hat{\mathbf{u}}\cdot\hat{\nabla}\hat{\mathbf{u}}\right)=\frac{-\hat{\nabla}\hat{p}}{\hat{\rho}}+\frac{\nu_0}{L_0u_0}\hat{\nabla}^2\hat{\mathbf{u}}+\frac{g_0L_0}{u_0^2}\hat{\boldsymbol{g}}$$

其中，$\frac{L_0u_0}{\nu_0}$，$\frac{g_0L_0}{u_0^2}$ 均为无量纲数。

(1) 结合物理意义，调整无量纲数的形式：

雷诺数（Re）与弗劳德数（Fr），是流体力学中常用的无量纲数。定义如下：

$$\mathrm{Re}=\frac{L_0u_0}{\nu_0}$$

$$\mathrm{Fr}=\frac{u_0}{\sqrt{g_0L_0}}$$

其中，雷诺数是一种用来表征流体流动情况的无量纲数，弗劳德数是表征流体惯性力和重力相对大小的一个无量纲数，表示惯性力和重力量级的比。将雷诺数与弗劳德数代入无量纲化后的 N-S 方程，得到下式：

$$\frac{\partial \hat{\mathbf{u}}}{\partial \hat{t}}+\left(\hat{\mathbf{u}}\cdot\hat{\nabla}\right)\hat{\mathbf{u}}=\frac{-\hat{\nabla}\hat{p}}{\hat{\rho}}+\frac{\hat{\nabla}^2\hat{\mathbf{u}}}{Re}+\frac{\hat{\mathbf{g}}}{Fr^2}$$

对于不可压缩流体，即 ρ 为常数 ρ_0，此时无量纲的密度为 $\hat{\rho}=1$，方程可进一步改写为如下形式：

$$\frac{\partial \hat{\mathbf{u}}}{\partial \hat{t}}+\left(\hat{\mathbf{u}}\cdot\hat{\nabla}\hat{\mathbf{u}}\right)=-\hat{\nabla}\hat{p}+\frac{\hat{\nabla}^2\hat{\mathbf{u}}}{Re}+\frac{\hat{\mathbf{g}}}{\mathrm{Fr}^2}$$

注：数据分析中的无量纲化，通常指将不同规格的数据转换到同一规格，也就是数据标准化与归一化。

16.数据归一化的方法[16-19]

(1) Standard Scaler 标准化：

$$x_{i,scaled} = \frac{x_i - \mu_i}{\sigma_i}$$

- 标准化为非线性特征缩放

(2) MinMax Scaler 归一化：

$$x_{i,scaled} = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}}$$

(3) MaxAbs Scaler 归一化：

$$x_{i,scaled} = \frac{x_i}{|x_{i,\max}|}$$

(4) Robust Scaler：

RobustScaler 函数使用对异常值鲁棒的统计信息来缩放特征。这个标量去除中值，并根据分位数范围(默认为 IQR 即四分位数范围)对数据进行缩放。IQR 是第 1 个四分位数和第 3 个四分位数之间的范围。公式如下：

$$x_{i,scaled} = \frac{x_i - Q_1(x_i)}{Q_3(x_i) - Q_1(x_i)}$$

(5) Log transform 对数变换：

$$x_{i,scaled} = \log(1 + x_i)$$

在图像处理中，对数变换主要用于将图像的低灰度值部分扩展，将其高灰度值部分压缩，以达到强调图像低灰度部分的目的。变换由下式给出：

$$s = c \cdot \log_{v+1}(1 + v \cdot r) \quad r \in [0, 1]$$

（四）数据分析平台

1. Shell、Terminal、Console 与 Kernel 的概念[20]

- Shell：命令行解释器，能够接收用户输入的命令，并对命令进行处理，处理完毕后再将结果反馈给用户。
- Terminal：终端，能够控制和运行 Shell 功能的封装程序。
- Console：控制台，是一种特殊的终端。

Terminal 和 Console 的区别是什么?

Belleve
计算机科学等 2 个话题下的优秀答主

+ 关注

166 人赞同了该回答

这个是 LCM 里面那个运行象棋程序的 PDP-8 小型计算机的控制台（在棋盘后面）以及连入它的 ADM-3A 视频终端（就是把 vi 的方向键搞成 hjkl 的那个）。对于早期的计算机来说，控制台往往并没有打字机功能，反而是一大堆 LED 灯和开关，用来显示和干预机器的内部状态。

PDP-8 的主机长这样。在 1965 年这台机器「仅售」$16000 而且「体积小巧」，因此广受好评。

- Kernel：操作系统内核，能够直接控制计算机硬件（CPU、内存、显示器等）。

2. Markdown 的基本语法[21]

- 标题：
  - # 一级标题
  - ## 二级标题
  - ### 三级标题
  - #### 四级标题
  - ##### 五级标题
  - ###### 六级标题
- 空格与空行：
  - 标题前的“#”与标题内容之间需空一格：

✅ Do this	❌ Don't do this
`# Here's a Heading`	`#Here's a Heading`

  - 标题与内容之间需空一行：

✅ Do this	❌ Don't do this

`Try to put a blank line before...`
 	
`# Heading`
 	
`...and after a heading.`
	`Without blank lines, this might not look right.`
	`# Heading`
	`Don't do this!`

- 段落：

  段落之间需空一行，段落开头无需空两格：

✅ Do this	❌ Don't do this

`Don't put tabs or spaces in front of your paragraphs.`
 	
`Keep lines left-aligned like this.`
	`  This can result in unexpected formatting problems.`
	
	`  Don't add tabs or spaces in front of paragraphs.`

- 字体：
  - *斜体*，_斜体_
  - **加粗**，__加粗__
  - ***斜体并加粗***，___斜体并加粗___

    斜体并加粗，_**斜体并加粗**_
- 注释（引用）：
  - >一级引用
  - >>二级引用
- 图片：
  - ![图片名称](图片相对/绝对/网络引用地址)表格：
- 表格：
  - 第一列表头 | 第二列表头

    ---------- | ----------

    第一列内容 | 第二列内容

    第一列内容 | 第二列内容

    第一列内容 | 第二列内容
- 程序：
  - ```python（这里可替换为其他语言名称）

    print(“Hello, World!”)

    ```
- 数学公式：
  - 行内公式：$E=mc^2$
  - 行间公式：$$E=mc^2$$
    ```

- 列表:
  - 1. One
    2. Two
    3. Three
  - * Start a line with a star
    * Profit
  - - Dashes work just as well
    - And if you have sub points, put two spaces before the dash or star:
      - Like this
      - And this

（五）爬虫原理

1. 爬虫的基本步骤

(1) 向目标网页（URL，服务器）发送 GET 请求。

(2) 目标服务器响应请求，并发送网页信息，返回网页内容。

(3) 本地接受 HTML 的源代码并解析数据。

(4) 本地的结构化数据中，搜索目标内容（文本、表格、数据、链接、多媒体资料等）。

2. HTML 的基本语法

- 基本结构：
  - HTML 网页必须包含<html >、<head>、<title>与<body>标签：

```
<html>
<head>
<title>文档标题</title>
</head>
<body>
可见文本
</body>
</html>
```

- 文本标签：

```
<b>粗体</b>
<i>斜体</i>
<u>下划线</u>
<p>新段落</p>
<br>下一行</br>
```

- 字体与颜色：

```
<font color="red">红色</font>
<font face＝"fontname">字体</font>
<font size＝n>字号</font>
```

- 多级标题：

```
<h1>这是标题 1</h1>
<h2>这是标题 2</h2>
```

```
<h3>这是标题 3</h3>
<h4>这是标题 4</h4>
<h5>这是标题 5</h5>
<h6>这是标题 6</h6>
```

- 注释:

```
<!-- 注释 -->
```

- 图片:

```
<img height=n width=n border=n src="URL">
```

- 超链接:

```
<a href="URL">
```

- 无序列表:

```
<ul>
    <li>项目</li>
    <li>项目</li>
</ul>
```

- 有序列表:

```
<ol>
    <li>第一项</li>
    <li>第二项</li>
</ol>
```

- 定义列表:

```
<dl>
  <dt>项目 1</dt>
    <dd>描述项目 1</dd>
  <dt>项目 2</dt>
    <dd>描述项目 2</dd>
</dl>
```

- 表格:

```
<table border="1">
  <tr>
    <th>表格标题</th>
    <th>表格标题</th>
  </tr>
  <tr>
    <td>表格数据</td>
    <td>表格数据</td>
  </tr>
```

```
10. </table>
```

3. 统一资源定位符（URL）[23]

以 URL：

http://www.aspxfans.com:8080/news/index.asp?boardID=5&ID=24618&page=1#r_70732423 为例，完整的 URL 可分为七个部分：

(1) **协议**：“http:”，代表网页使用的是 HTTP 协议。后面的“//”为分隔符。

(2) **域名**：“www.aspxfans.com”， 一个 URL 中，也可以使用 IP 地址作为域名。

(3) **端口**：“8080”，域名和端口之间使用“:”作为分隔符。端口不是一个 URL 必需的部分，如果省略端口部分，将采用默认端口。

(4) **虚拟路径**：“/news/”， 从域名后的第一个“/”开始到最后一个“/”为止，是虚拟路径部分。虚拟路径也不是一个 URL 必需的部分。

(5) **文件名**：“index.asp”， 从域名后的最后一个“/”开始到“?”为止，是文件名部分，如果没有“?”，则是从域名后的最后一个“/”开始到“#”为止，是文件部分，如果没有“?”和“#”，那么从域名后的最后一个“/”开始到结束，都是文件名部分。文件名部分也不是一个 URL 必需的部分，如果省略该部分，则使用默认的文件名。

(6) **请求参数**：“boardID=5&ID=24618&page=1”， 从“?”开始到“#”为止之间的部分为参数部分，又称搜索部分、查询部分。参

数可以允许有多个参数，参数与参数之间用“&”作为分隔符。

(7) **锚**：“r_70732423”，从“#”开始到最后，都是锚部分。锚部分也不是一个 URL 必需的部分。

4. JS 与 CSS 的概念[24]

- JS：一种客户端编程语言，有助于鼓励网站上的交互性。该脚本运行在用户浏览器而不是服务器上，可以增加网站的功能，而无需从头编写代码。
- CSS：一种用来表现 HTML 或 XML 等文件样式的计算机语言。CSS 不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。

5. 正则表达式的基本语法[25]

- 匹配单个字符：

字符	功能
.	匹配任意一个字符（除换行符\n）
[aeiou]	匹配列出的集合中的单个字符
[^XYZ]	匹配不在列出的集合中的单个字符
[a-z0-9]	匹配范围内的字符
\d	匹配数字，等价于[0-9]
\D	匹配非数字
\s	匹配空白，包括 space 与 tab
\S	匹配非空白
\w	匹配单词字符，等价于[0-9a-zA-Z]

\W	匹配非单词字符

- 匹配多个数量的单字符:

字符	功能
*	匹配前一个字符出现零次或若干次
+	匹配前一个字符出现一次或若干次
?	匹配前一个字符出现零次或一次
{m}	匹配前一个字符出现 m 次
{m,}	匹配前一个字符至少出现 m 次
{m,n}	匹配前一个字符出现 m 到 n 次

- 匹配边界:

字符	功能
^	匹配字符串开头
$	匹配字符串结尾

- 匹配分组:

字符	功能
|	匹配左右任意一个表达式
(	表示从哪里开始提取字符串
)	表示在哪里停止提取字符串

（六）时间序列模型

时间序列基本概念

- 时间序列是按照时间排序的一组随机变量序列：

$$\{X_t\} = X_1, X_2, ..., X_N$$

- 每个X_t的观测值序列，是时间序列的一个实现：

$$\{x_t\} = x_1, x_2, ..., x_N$$

2. 统计基础概念

(1) 总体期望：

- 连续形式：

$$\mu = E[X] = \int_{-\infty}^{\infty} x f(x) dx$$

- 离散形式：

$$\mu = \mathrm{E}[X] = \sum_{i=1}^{k} x_i p_i = x_1 p_1 + x_2 p_2 + \cdots + x_k p_k$$

(2) 总体方差：

- 连续形式：

$$\sigma^2 = Var[X] = \int (x-\mu)^2 f(x) dx = \int x^2 f(x) dx - \mu^2$$

- 离散形式：

$$\sigma^2 = Var[X] = \sum_{i=1}^{n} p_i \cdot (x_i - \mu)^2 = \sum_{i=1}^{n} (p_i \cdot x_i^2) - \mu^2$$

(3) 总体相关系数：

$$\rho_{X,Y} = corr[X,Y] = \frac{cov[X,Y]}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

(4) 样本均值：

$$\overline{x}=\frac{1}{n}\sum_{k=1}^{n}x_i$$

(5) 样本方差：

令：

$$\sigma_x^2=\frac{1}{n}\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2=\left(\frac{1}{n}\sum_{i=1}^{n}X_i^2\right)-\overline{X}^2$$

则无偏的样本方差为：

$$s^2=\frac{n}{n-1}\sigma_X^2=\frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2\right)=\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2$$

(6) 总体协方差矩阵：

$$Var[X]=\begin{bmatrix}Var[X_1] & \dots & Cov[X_1,X_k]\\ \vdots & \ddots & \vdots\\ Cov[X_k,X_1] & \dots & Var[X_k]\end{bmatrix}$$

$$=\begin{bmatrix}\sigma_1 & 0 & \cdots & 0\\ 0 & \sigma_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & \cdots & \cdots & \sigma_n\end{bmatrix}\times\begin{bmatrix}1 & \rho_{12} & \cdots & \rho_{1n}\\ \rho_{21} & 1 & \cdots & \rho_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \rho_{n1} & \cdots & \cdots & 1\end{bmatrix}\times\begin{bmatrix}\sigma_1 & 0 & \cdots & 0\\ 0 & \sigma_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & \cdots & \cdots & \sigma_n\end{bmatrix}$$

3. 总体均值与自协方差函数[26]

- 对所有$t\in T$，时间序列的总体均值为：

$$\mu_t=E(X_t)$$

- 对所有的$s,t\in T$，自协方差函数为：

$$\gamma_{s,t}=Cov[X_s,X_t]=\mathbb{E}[X_sX_t]-\mathbb{E}[X_t]\mathbb{E}[X_s]$$

4. 自相干系数与偏自相干函数

- 自相干系数（ACF），即变量在 s 时刻与 t 时刻的相关系数：

$$\rho_{s,t}=Corr[X_s,X_t]=\frac{E[(X_s-\mu_s)(X_t-\mu_t)]}{\sigma_s\sigma_t}$$

考虑弱平稳条件：

$$\gamma_{s,t} = Cov[X_s, X_t] = Cov[X_{s+r}, X_{t+r}] = \gamma_{s+r,t+r}$$

在给定弱平稳的条件下，X_t 的均值与方差与时间无关，此时自相干系数可以写成：

$$\rho_t = Corr[X_t, X_{t+\tau}] = \frac{Cov[X_t, X_{t+\tau}]}{\sigma_t \sigma_{t+\tau}} = \frac{\gamma_\tau}{\gamma_0}$$

- 偏自相干系数（PACF）：

$$\begin{cases} \alpha(1) = corr(z_{t+1}, z_t), & k = 1 \\ \alpha(k) = corr(z_{t+k} - P_{t,k}(z_{t+k}), z_t - P_{t,k}(z_t)), & k \geq 2 \end{cases}$$

其中P是z在希尔伯特空间(z_{t+1}, z_{t+k-1})的投影。

5. 时间序列的应用

- 股票收益 EPS（股票的每股收益）
- 语音振幅（音节 aaa-hhh）
- 道·琼斯工业指数
- 德国零售额

6. 时间序列的平稳性

- 严格平稳（强平稳）：
  - 概念：给定时间序列$\{X_t | t \in T\}$，对于所有的$t_1 ... t_k$与r，如果联合分布$f(X_{t_1}, ..., X_{t_k})$与$f(X_{t_{1+\tau}}, ..., X_{t_{k+\tau}})$相同，则该序列是严格平稳的。
  - 换句话说，**强平稳时间序列的概率分布不随时间变化而变化**。
  - 严格平稳是非常严格的，而真实过程很少符合。一般只有纯粹的随机过程严格平稳，因此使用的更多的是弱平稳。

- 弱平稳：
  - 给定时间序列$\{X_t|t\in T\}$，如果满足以下三个条件：

    (1) 均值为有限常数：即$\mu_t=\mathbb{E}[X_t]=\mu<\infty$

    (2) 方差为有限常数：即$\sigma_t^2=Var[X_t]=\sigma^2<\infty$

    (3) 自协方差和自相关函数仅取决于滞后τ，即

    $$\gamma_{t,t+\tau}=Cov[X_t,X_{t+\tau}]=\gamma_\tau$$

    $$\rho_{t,t+\tau}=Corr[X_t,X_{t+\tau}]=\rho_\tau$$

    则该序列是弱平稳的。
  - 换句话说：**弱平稳时间序列的一阶矩、二阶矩不随时间变化而变化**。
  - 分别令强平稳性中的$k=1,2$，即可由强平稳性推出弱平稳条件。
- 强平稳性与弱平稳性的区别：
  - 平稳性定义时间序列的统计特征不随时间变化。
  - 强平稳性是指时间序列的变量的联合概率分布不随时间变化，由于在实际使用中，联合概率分布很难直接测量，所以该定义在实践中很少使用。
  - 弱平稳性为了弥补强平稳性在实践中测量和检验的实操性，检验时间序列的一阶和二阶动量，即平均值和方差不随时间变化。这样的定义不但方便测量，也便于检测。

7. 平稳性检验[27]：

- 增强迪基-富勒检验（Augmented Dickey-Fuller Testing）：用于检验时间序列是否平稳（可以检验序列包含高阶滞后相关的情景）

8. 时间序列的分解

- 时间序列可分解为：**趋势、周期性、不规律特征、噪音**

考虑加法模型：$y_t = d_t + c_t + \varepsilon_t$，其中$d_t$，$c_t$，$\varepsilon_t$分别为趋势分量、周期分量与无规律分量，则时间序列分解的原理，就是对分量建模，提取出分量d_t，c_t，**使得无规律分量**$\varepsilon_t = y_t - d_t - c_t$**具有平稳性，接近白噪声**。

- 常见的趋势模型：
  - 线性趋势：$y_t - y_{t-1}$
  - 二次趋势：$(y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$
  - 指数趋势：$\dfrac{y_t - y_{t-1}}{y_t}$
- 周期分量：时间序列和趋势之间的差异，可以取经济学周期（周、月、季、年等），并不总是出现在时间序列中。

9. 白噪声[28]

- 概念：白噪声是一种特殊的弱平稳随机过程$\{e_t | t \in T\}$，且满足以下三个条件：

  (1) $E(e_t) = 0$

  (2) $Var(e_t) = \sigma^2$

  (3) 当$\tau \neq 0$时，有$Cov(e_t, e_{t+\tau}) = 0$
- 换句话说，**白噪声的一阶矩为常数 0，二阶矩互不相关**。
- 当白噪声服从高斯分布时，才能推出e_t，$e_{t+\tau}$彼此独立，否则e_t，$e_{t+\tau}$仅仅是互不相关。这种白噪声称为高斯白噪声[29]。
- 白噪声可应用于时间序列的噪声检测，也应用于科学、金融领域的随机过程模拟。

10.随机漫步模型

- 随机漫步：

$$y_t = y_{t-1} + \varepsilon_t$$

- 带漂移项的随机漫步：

$$y_t = c + y_{t-1} + \varepsilon_t = ct + \sum_{j=1}^{t} \varepsilon_t$$

11.自回归模型[30]（AR(p)）

- AR(p)：

$$X_t = \varphi_0 + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + ... + \varphi_p X_{t-p} + \varepsilon_t$$

其中，φ_0为常数项，ε_t是独立同分布的，均值为0标准差为ε的随机误差。

  - AR(p)模型可以理解为，X_t**的期望值等于**q**个落后期的线性组合，加上常数项与随机误差**。
  - 应用场景：适用于短期预测

- AR(1)：

$$X_t = c + \varphi X_{t-1} + \varepsilon_t = \frac{c}{1-\varphi} + \sum_{i=0}^{t-1} \varphi^i \varepsilon_{t-i}$$

  - AR(1)的稳定性条件是：$|\varphi| < 1$
  - AR(1)具有以下形式的均值、方差、自协方差函数和自相干系数：

$$\mathrm{E}(X_t) = \frac{c}{1-\varphi}$$

$$Var(X_t) = \frac{\sigma^2}{1-\varphi^2}$$

$$Cov(X_t, X_{t-\tau}) = \frac{\sigma^2}{1-\varphi^2}\varphi^\tau$$

$$Corr(X_t, X_{t-\tau}) = \varphi^\tau$$

- AR(1)的单次脉冲将影响后续所有取值。从自相干系数中可以看出，这种影响呈指数衰减，衰减率取决于回归系数φ。

- AR(p)的矩阵化：见 2.5.3 时间序列的矩阵化。
- AR(p)的稳定性条件[31]：

  必要条件：

$$\begin{cases}\sum_{i=1}^{p}\varphi_i<1\\ |\varphi_i|<1\end{cases}$$

  充分条件：

$$\sum_{i=1}^{p}|\varphi_i|<1$$

12.移动平均模型[32]（MA(p)）

- MA(q)：

$$X_t=c+\varepsilon_t+\theta_1\varepsilon_{t-1}+\theta_2\varepsilon_{t-2}+.....+\theta_q\varepsilon_{t-q}$$

  - MA(q)模型可以理解为：**当前值是由过去误差的线性组合组成的，其中误差是服从正态分布并且相互独立的。**
  - MA(q)模型对于任意参数组合均满足稳定性条件。
  - 容易看出，在滞后 q 期后，MA(q)的自协方差函数为 0
  - 应用场景：移动平均模型用于预测未来值，移动平均平滑用于估计过去值的趋势周期；

- MA(1)：

$$X_t=\theta_0+\varepsilon_t+\theta_1\varepsilon_{t-1}$$

  - MA(1)具有以下形式的均值、方差、协方差和自相干系数：

$$E(X_t) = \theta_0$$

$$Var(X_t) = (1+\theta_1^2)\sigma^2$$

$$Cov(X_t, X_{t-\tau}) = \begin{cases} \theta_1\sigma^2 & \text{if } \tau = 1 \\ 0 & \text{if } \tau > 1 \end{cases}$$

$$Corr(X_t, X_{t-\tau}) = \begin{cases} \dfrac{\theta_1}{1+\theta_1^2} & \text{if } \tau = 1 \\ 0 & \text{if } \tau > 1 \end{cases}$$

13.移动平均自回归模型（ARMA(p,q)）

- ARMA(p,q):

$$X_t = \varphi_0 + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

  - **ARMA(p,q)模型就是自回归与移动平均的结合。**
  - ARMA(p,q)模型的平稳性取决于 AR 项。
  - 滞后 q 阶后，ARMA(p,q)的自相干系数取决于 AR 项。

（七）聚类分析

1. 聚类的相关概念

- 数据集群化：
  - 相似的对象在同一个集群
  - 不同的对象在不同的集群
- 集群的属性：
  - 连通性
  - 分离程度
  - 密度

2. 聚类的应用

- 生物学：

  生物分类：界、门、纲、目、科、属和种
- 信息检索：文档聚类
- 土地利用：土地分类、分割、工程管理
- 营销：发现客户群，制定有针对性的营销计划
- 城市规划：根据房屋类型、价值和地理位置确定房屋组
- 经济科学：市场研究
- 地图信息：

  利用遥感卫星彩色成像数据，色相 RGB 矢量聚类；

  将卫星图像分为不同的土地用途：森林、农业、沙漠、建筑物等。

3. 聚类的基本步骤

(1) 特征选择：选择相关特征或变量

(2) 相似度测量：选择合适的公式度量两个特征向量的相似度

(3) 聚类准则：建立成本函数评估优劣

(4) 聚类算法：算法的选择

(5) 结果验证：验证测试

4. 聚类的具体算法

- 层次聚类：
  - 方法：合并法、分解法、绘制树状图。
  - 流程（合并法）：

    (1) 每一个样本点视为一个簇;

    (2) 计算各个簇之间的距离，最近的两个簇聚合成一个新簇;

    (3) 重复以上过程，直至最后只有一个簇
  - 时间复杂度：$O(n^3)$
  - 空间复杂度：$W(n^2)$
- 基于质心的聚类（K-Means）：
  - 流程：

    (1) 随机选择 K 个簇中心，按照距离最近原则，簇类各自圈选簇内数据点;

    (2) 根据圈选的簇内数据，更新 K 个簇的中心;

    (3) 不断重复(1)和(2)，直到距离的平方和收敛

    (4) 重复多次 K-means 聚类，排除步骤(1)中初始化的随机性，确保该方法能够收敛。

- 为什么 K-Means 一般需要重复多次:

  由于在计算 K-means 聚类的时候，聚的质心的初始值随机产生。在计算 K-means 聚类时，虽然可能已经获得正确的聚类中心，但是不排除除一定收敛到全局最小值，所以为了排斥随机质心导致的影响，需要多次重复该过程，以确保收敛到全局最小值。

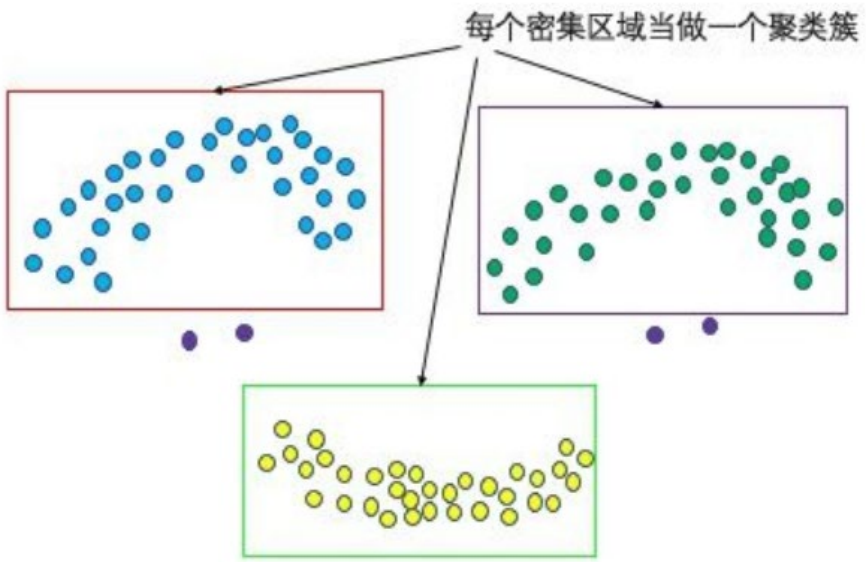
- 基于分布（模型）的聚类（高斯混合模型）[33]:

高斯混合模型是一种聚类算法，它假设数据点是由具有未知参数的高斯分布的混合生成的。该算法的目标是估计高斯分布的参数，以及来自每个分布的数据点的比例。相比之下，K-means 不对数据点的潜在分布做出任何假设，它只是将数据点划分为 K 个集群，其中每个集群由其质心定义。

- 虽然高斯混合模型更灵活，但它们可能比 K-means 更难训练。K-means 通常收敛速度更快，因此在运行时是一个重要考虑因素的情况下可能是首选。
- 一般来说，当数据集较大且聚类分离良好时，K-means 会更快、更准确。当数据集较小或聚类分离不充分时，高斯混合模型会更准确。
- 高斯混合模型考虑了数据的方差，而 K-means 没有。
- 高斯混合模型在簇的形状方面更加灵活（但是不适用于非凸多边形），而 K-means 仅限于球形簇。
- 高斯混合模型可以处理缺失数据，而 K-means 不能。这种差异可以使高斯混合模型在某些应用中更有效，例如具有大量噪声的数据或未明确定义的数据。

- 基于密度的聚类（DBSCAN）[34]:

直观效果上看，DBSCAN 算法可以找到样本点的全部密集区域，并把这些密集区域当做一个一个的聚类簇。



DBSCAN的1个核心思想

和传统的 k-means 算法相比，DBSCAN 算法不需要输入簇数 k 而且可以发现任意形状的聚类簇，同时，在聚类时可以找出异常点。

- DBSCAN 算法的主要优点：

(1) 可以对任意形状的稠密数据集进行聚类，而 k-means 之类的聚类算法一般只适用于凸数据集。

(2) 可以在聚类的同时发现异常点，对数据集中的异常点不敏感。

(3) 聚类结果没有偏倚，而 k-means 之类的聚类算法的初始值对聚类结果有很大影响。

- DBSCAN 算法的主要缺点：

(1) 样本集的密度不均匀、聚类间距差相差很大时，聚类质量较差，这时用 DBSCAN 算法一般不适合。

(2) 样本集较大时，聚类收敛时间较长。此时可以对搜索最近邻时建立的 KD 树或者球树进行规模限制来进行改进。

(3) 调试参数时比较复杂，主要是对距离阈值 Eps，邻域样本数阈值 MinPts 进行联合调参，不同的参数组合对最后的聚类效果有较大影响。

(4) 对于整个数据集只采用了一组参数。如果数据集中存在不同密度的簇或者嵌套簇，则 DBSCAN 算法不能处理。

(5) DBSCAN 算法可过滤噪声点，这同时也是其缺点，这造成了其不适用于某些领域，如对网络安全领域中恶意攻击的判断。

5. 距离函数与相似度

- 距离函数:
  - 概念：定义集合的每对元素之间距离的函数。
  - 特性:

非负性：$d(x,y) \geq 0$

唯一性：$d(x,y) = 0 \Leftrightarrow x = y$

对称性：$d(x,y) = d(y,x)$

满足三角不等式：$d(x,z) \leq d(x,y) + d(y,z)$

  - 常用的距离函数:

欧氏距离：$d(x,y) = |x-y|_2 = \left(\sum (x_i - y_i)^2\right)^{1/2}$

曼哈顿距离：$d(x,y) = |x-y|_1 = \sum |x_i - y_i|$

闵可夫斯基距离：$d(x,y) = |x-y|_2 = \left(\sum (x_i - y_i)^2\right)^{1/2}$

- 相似度:
  - 相似度与距离的度量相反：距离越远，相似度越低；距离越近，相

似度越高。

- 特性：

$$\text{非负性：}\ s(x,y) \geqslant 0$$
$$\text{唯一性：}\ s(x,y) = 1 \Leftrightarrow x = y$$
$$\text{对称性：}\ s(x,y) = s(y,x)$$

- 相似度与距离的关系：

$$s(x,y) = \frac{1}{1+d(x,y)}$$

- 距离函数和相似度的关联与差异：
  - 距离函数一般有定义数据点的高维度矢量之差的函数定义，代表矢量之间的距离。
  - 相似度一般定义为距离的倒数，归一化在 0 和 1 之间，描述数据点之间的相似度。相似度越低，距离越近；相似度越高，距离越远。

6. 聚类的内部评估指标[35]

内部评估指标主要基于数据集的集合结构信息从紧致性、分离性、连通性和重叠度等方面对聚类划分进行评价，即**基于数据聚类自身进行评估**的。

- 轮廓系数：

对于聚类后的数据，给定其中一个样本点 i，则关于该样本的轮廓系数计算公式如下：

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

其中$a(i)$分别为该样本与簇内其他样本点的平均距离，$b(i)$为该样本与距离最近的簇的簇外样本点平均距离。

- 对于一个样本集合，它的轮廓系数是所有样本轮廓系数的平均值。显然，轮廓系数的取值范围为$(-1,1)$，**轮廓系数越大，说明聚类效果越好**。

- Calinski-Harabaz 指数：

$$CH = \left[\frac{\sum_{k=1}^{K} n_k \|c_k - c\|^2}{K-1}\right] / \left[\frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k} \|d_i - c_k\|^2}{N-K}\right]$$

其中，k 为簇数，c 全局中心点，N 为样本总数；d_i 为数据点；c_k 为簇 k 的中心；n_k 为簇 k 的数量。

  - CH 指数是分离度与紧密度的比值。**CH 指数越大，代表类自身越紧密，簇间距越大，聚类结果越好**。
  - 优点：计算时间短；当簇的样本点密集且分离较好时，CH 指数高。
  - 缺点：凸的簇 CH 指数通常高于其他类型的簇。例如，通过 DBSCAN 等算法得到的基于密度聚类的簇，不适合使用 CH 指数评估。

- Davies-Bouldin 指数：

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

其中，s_i 表示簇的每个点与该簇的质心之间的平均距离，也称为簇直径。d_{ij} 表示聚类 i 和 j 的质心之间的距离。

  - DB 指数是计算任意两类别的类内距离平均距离之和除以两聚类中心距离求最大值。**DB 指数越小，意味着类内距离越小，类间距离越大，聚类效果越好**。

7. 聚类的外部评估指标[35]

外部评估是指**当数据集的外部信息可用**时，通过比较聚类划分与外部准则的匹配度，可以评价不同聚类算法的性能。

- 兰德指数：

$$\mathrm{RI}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{C}_{\mathrm{N}}^{2}}$$

其中：

TP 为将相似的样本归为同一个簇（同-同）的数据点对数；

TN 将不相似的样本归入不同的簇（不同-不同）的数据点对数；

FP 将不相似的样本归为同一个簇（不同-同）的数据点对数；

FN 将相似的样本归入不同的簇（同-不同）的数据点对数；

  - RI 是正确决策的比率（精确率），因此 RI 的取值范围是$[0,1]$，**取值越大，聚类效果越好**。

- 调整兰德指数：

$$\mathrm{ARI}=\frac{\mathrm{RI}-\mathrm{E}[\mathrm{RI}]}{\max(\mathrm{RI})-\mathrm{E}[\mathrm{RI}]}$$

  - 对于随机结果，RI 并不能保证分数接近零。为了实现“在聚类结果随机产生的情况下，指标应该接近零”，使用 ARI 将具有更高的区分度。
  - ARI 取值范围为$[-1,1]$，**取值越大，聚类效果越好**。

- 标准化互信息：

$$\mathrm{NMI}(\mathrm{U},\mathrm{V})=\frac{\mathrm{MI}(\mathrm{U},\mathrm{V})}{mean(\mathrm{H}(\mathrm{U}),\mathrm{H}(\mathrm{V}))}$$

其中：

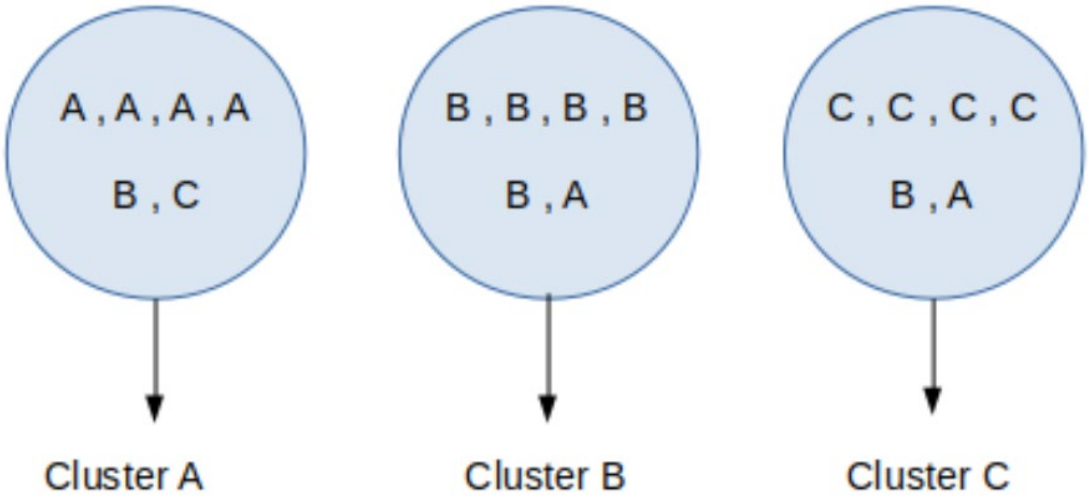
$$\mathrm{MI}(U,V)=\sum_{i=1}^{|U|}\sum_{j=1}^{|V|}\frac{|U_i\cap V_j|}{N}\log\left(\frac{N|U_i\cap V_j|}{|U_i||V_j|}\right)$$

$$\mathrm{H}(U)=-\sum_{i=1}^{|U|}\mathrm{P}'(i)\log(\mathrm{P}'(i))$$

$$\mathrm{H}(\mathrm{V})=-\sum_{\mathrm{j}=1}^{|\mathrm{V}|}\mathrm{P}'(\mathrm{j})\log(\mathrm{P}'(\mathrm{j}))$$

- MI 与 NMI 取值范围均为$[0,1]$，它们都表示**取值越大，聚类效果越好**。

- 聚类纯度:



$$\text{purity}=\frac{(\text{cluster}\,A+\text{cluster}\,B+\text{cluster}\,C)}{\text{total}}$$

其中，$\text{cluster}\,A$，$\text{cluster}\,B$，$\text{cluster}\,C$均为正确分类到对应类别的样本点的个数。purity 的**取值越大，聚类效果越好**。

8. 图片类信息的聚类分析

在地图图片中，水域、林地、土壤等元素的颜色各不相同，假设图片表示为$I(x,y,3)$，可将图片的像素展开为$R(x,y)$，$G(x,y)$，$B(x,y)$的三个维度的强度信息。将像素视为 RGB

三原色的向量。一般同类颜色的元素在三维 RGB 空间中距离较近，形成聚类。这样通过聚类算法，在三维空间内能够区分颜色相同的区域，从而标记不同特征的区域信息，进行聚类分析。

（八）神经网络基本原理

1. 一维权重的批量梯度下降法：

定义神经网络的模型为$\hat{y}=wx$，取均方误差损失函数为$L=\|y-\hat{y}\|_2$，

由梯度下降的定义，有：

$$w(k+1)=w(k)-\Delta w(k)=w(k)-\eta\frac{\partial L}{\partial w}$$

考虑批量梯度下降的损失函数：

$$L=\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2$$

我们使用链式求导法则计算梯度：

$$\frac{\partial L_i}{\partial \hat{y}_i}=2(\hat{y}_i-y_i),\quad \frac{\partial \hat{y}_i}{\partial w}=x_i$$

$$\frac{\partial L}{\partial w}=\sum_{i=1}^{N}\frac{\partial L_i}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial w}=\frac{1}{N}\sum_{i=1}^{N}2x_i(\hat{y}_i-y_i)$$

2. 多维权重逻辑分类的随机梯度下降法：

定义神经网络的模型为$\hat{y}=\sigma(\boldsymbol{x}\boldsymbol{w}^T)$，其中$\sigma(z)=\frac{1}{1+e^{-z}}$，

取交叉熵损失函数为$L(y_i,\hat{y}_i)=-y_i\log\hat{y}_i-(1-y_i)\log(1-\hat{y}_i)$

由梯度下降的定义，有：

$$\mathbf{w}(k+1)=\mathbf{w}(k)-\Delta\mathbf{w}(k)=\mathbf{w}(k)-\eta\frac{\partial L}{\partial \mathbf{w}}$$

我们使用链式求导法则计算梯度：

$$\frac{\partial L_i}{\partial \hat{y}_i} = -\left[\frac{y_i}{\hat{y}_i} - \frac{1-y_i}{1-\hat{y}_i}\right] = \frac{\hat{y}_i - y_i}{\hat{y}_i(1-\hat{y}_i)}$$

$$\frac{\partial \hat{y}_i}{\partial z_i} = \hat{y}_i(1-\hat{y}_i)$$

$$\frac{\partial z_i}{\partial \mathbf{w}} = \frac{\partial \mathbf{x}_i \cdot \mathbf{w}}{\partial \mathbf{w}} = \mathbf{x}_i$$

$$\frac{\partial L_i}{\partial \mathbf{w}} = \frac{\partial L_i}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial z_i}\frac{\partial z_i}{\partial \mathbf{w}} = \mathbf{x}_i(\hat{y}_i - y_i)$$

3.梯度下降方法的优缺点以及改进[36]

- 批量梯度下降法的每一步都把整个训练集载入进来进行计算，时间花费和内存开销都非常大，无法应用于大数据集、大模型的场景。
- 随机梯度下降法则放弃了对梯度准确性的追求，每步仅仅随机采样一个(或少量)样本来估计当前梯度，计算速度快，内存开销小。但由于每步接受的信息量有限，随机梯度下降法对梯度的估计常常出现偏差，造成目标函数曲线收敛得很不稳定，伴有剧烈波动，有时甚至出现不收敛的情况。
- 解决方案： Momentum 方法、AdaGrad 方法、Adam 方法

4. 监督性学习、无监督性学习，强化学习

- 监督学习：

监督学习具有特征和标签。即使数据是没有标签的，也可以通过学习特征和标签之间的关系，判断出标签（分类）。

简言之：**提供数据，预测标签**，比如对动物猫和狗的图片进行预则，预测 label 为 cat 或者 dog。

通过已有的一部分输入数据与输出数据之间的对应关系，生成一个函数，将输入映射到合适的输出，例如分类和回归问题。

- 无监督学习:

无监督学习即只有特征，没有标签。只有特征，没有标签的训练数据集中，通过数据之间的内在联系和相似性将他们分成若干类（聚类），根据数据本身的特性，从数据中根据某种度量学习出一些特性。

比如一个人没有见过恐龙和鲨鱼，如果给他看了大量的恐龙和鲨鱼，虽然他没有恐龙和鲨鱼的概念，但是他能够观察出每个物种的共性和两个物种间的区别的，并对这两种动物予以区分。

简言之：**给出数据，寻找隐藏的关系**。

- 强化学习:

强化学习与半监督学习类似，均使用未标记的数据，但是强化学习通过算法学习是否距离目标越来越近，可以理解为激励与惩罚函数。类似生活中，女朋友不断调教直男男友变成暖男。

简言之：**通过不断激励与惩罚，达到最终目的**。

- 区别:
  - 监督学习有反馈，无监督学习无反馈，强化学习是执行多步之后才反馈。
  - 强化学习的目标与监督学习的目标不一样，即强化学习看重的是行为序列下的长期收益，而监督学习往往关注的是和标签或已知输出的误差。
  - 强化学习的奖惩概念是没有正确或错误之分的，而监督学习标签就

是正确的，并且强化学习是一个学习+决策的过程，有和环境交互的能力(交互的结果以惩罚的形式返回)，而监督学习不具备。

5. 过拟合与欠拟合

- 欠拟合:
  - 根本的原因是特征维度过少，导致拟合的函数无法满足训练集，误差较大。
  - 欠拟合问题可以通过增加特征维度来解决。
- 过拟合:

  根本的原因则是特征维度过多，导致拟合的函数完美的经过了训练集。
- 解决过拟合问题的两个途径:
  - **减少特征维度**：可以人工选择保留的特征，或者模型选择算法。
  - **正则化**：保留所有的特征，通过降低参数θ的值，来影响模型。

参考文献

[1] 余弦相似度、欧氏距离、斯皮尔曼相关系数:

https://blog.csdn.net/y990041769/article/details/77837915

[2] 皮尔逊相关系数:

https://blog.csdn.net/yanyanwenmeng/article/details/111033499

[3] 马氏距离:

https://blog.csdn.net/weixin_39910711/article/details/113985520

[4] CPU 个数、CPU 核心数、CPU 线程数:

http://t.zoukankan.com/kimsimple-p-7787018.html

[5] 定类，定序，定距，定比四种数据类型:

https://blog.csdn.net/qq_33254766/article/details/115111191

https://blog.csdn.net/YYIverson/article/details/100068865

https://blog.csdn.net/mjfppxx/article/details/120506923

https://blog.csdn.net/shenbo2030/article/details/20040455

[6] 等距分箱、等频分箱:

https://blog.csdn.net/sweet1194695742/article/details/116794173

[7] 指数分箱:

https://blog.csdn.net/weixin_40631985/article/details/102168512

[8] 灰度图与彩色图:

https://zhuanlan.zhihu.com/p/145728538

[9] 停用词:

https://baike.baidu.com/item/%E5%81%9C%E7%94%A8%E8%AF%8D/4531676

[10] 概率抽样方法：

https://blog.csdn.net/qq_41080850/article/details/86593220

[11] 非概率抽样方法：

https://blog.csdn.net/jackyrongvip/article/details/104115620

[12] 无量纲化的基本概念：

https://baike.baidu.com/item/%E6%97%A0%E9%87%8F%E7%BA%B2%E5%8C%96/10689984?fr=aladdin

[13] Navier-Stokes 方程：

https://baike.baidu.com/item/%E7%BA%B3%E7%BB%B4%EF%BC%8D%E6%96%AF%E6%89%98%E5%85%8B%E6%96%AF%E6%96%B9%E7%A8%8B?fromModule=lemma_inlink

[14] 雷诺数：

https://baike.baidu.com/item/%E9%9B%B7%E8%AF%BA%E6%95%B0/2691284?fr=aladdin

[15] 弗劳德数：

https://baike.baidu.com/item/%E5%BC%97%E5%8A%B3%E5%BE%B7%E6%95%B0/228891?fr=aladdin

[16] Standard Scaler：

https://blog.csdn.net/wzyaiwl/article/details/90549391

[17] MinMaxScale、MaxAbsScaler：

https://blog.csdn.net/weixin_40683253/article/details/81508321

[18] Robust Scaler：

https://blog.csdn.net/qq_41185868/article/details/108521209

[19] Log transform:

https://blog.csdn.net/zhoufan900428/article/details/12709361

[20] Shell、Terminal 与 Console 的区别:

https://www.zhihu.com/question/20388511/answer/990303033

https://www.zhihu.com/question/21711307/answer/56056972

[21] Markdown 语法:

https://blog.csdn.net/m0_52316372/article/details/125724052

[22] HTML 基础:

https://blog.csdn.net/zong596568821xp/article/details/83277729

[23] URL:

https://blog.csdn.net/m0_53592673/article/details/111668038

[24] CSS:

https://baike.baidu.com/item/CSS/5457

[25] 正则表达式:

https://blog.csdn.net/qq_44159028/article/details/120575621

[26] 自相干系数与平稳性:

https://zhuanlan.zhihu.com/p/424609116

[27] Augmented Dickey-Fuller Testing:

https://blog.csdn.net/TimeFuture/article/details/120690931

[28] 白噪声:

https://blog.csdn.net/jason_cuijiahui/article/details/87517127

https://baike.baidu.com/item/%E7%99%BD%E5%99%AA%E5%A3%B0/4062223?fr=aladdin

[29] 高斯白噪声：

https://baike.baidu.com/item/%E9%AB%98%E6%96%AF%E7%99%BD%E5%99%AA%E5%A3%B0/3547261

[30] 自回归模型：

https://baike.baidu.com/item/%E8%87%AA%E5%9B%9E%E5%BD%92%E6%A8%A1%E5%9E%8B/1037587?fr=aladdin

[31] AR(q)模型的稳定性条件：

https://blog.csdn.net/m0_37422217/article/details/105474653

[32] 移动平均模型：

https://zhuanlan.zhihu.com/p/46886003

https://zhuanlan.zhihu.com/p/435441628

[33] 高斯混合模型：

https://blog.csdn.net/bashendixie5/article/details/124891359

[34] DBSCAN：

https://blog.csdn.net/hansome_hong/article/details/107596543

[35] 聚类的评价指标：

https://blog.csdn.net/scgaliguodong123_/article/details/121303457

[36] 梯度下降方法的优缺点：

https://blog.csdn.net/qq_44614524/article/details/114241259