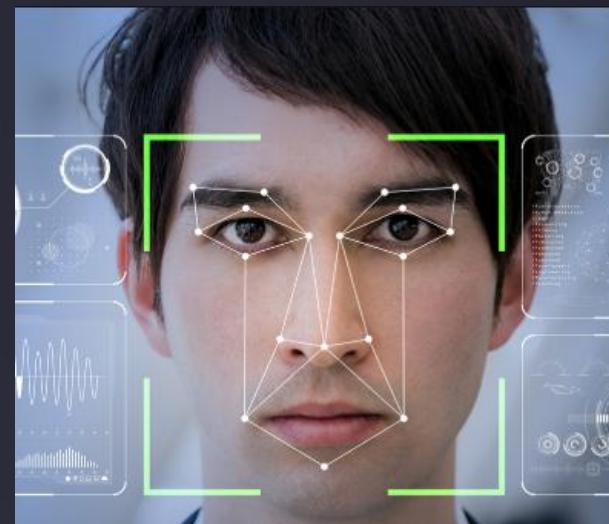




Python3人工智能入门+实战提升：机器学习

Chapter 1 课程介绍及环境配置

赵辛



本门课程的特点

网络AI课程

1、学员基础要求高：
微积分、线性代数、
计算机编程...

2、内容太专：卷积
神经网络、循环神经网
络、tensorflow...

3、概念堆积：80%
以上都是抽象的概念、
缺乏实战

4、实战内容单一：
iris鸢尾花、mnist手
写字...

学员情况

1、大部分学员的数
学、编程基础薄弱

2、期望由浅入深地
了解各项技术

3、希望内容能与现
实案例结合，加深理
解

4、渴望掌握实战工
具，能够解决实际问
题...

本门课程

1、以AI知识与实战为主、降低学习难度
课程安排：现实案例+知识干货+实战

2、生活案例出发、覆盖主流技术，侧重
对不同技术的应用理解

3、现实案例驱动知识讲解、手把手带你
编程，帮助加深知识点理解

4、实战案例与现实场景相结合，帮助加
强解决实际问题的能力

知识加餐

毕业设计规划
实战项目开发
综合能力提升

讲师介绍



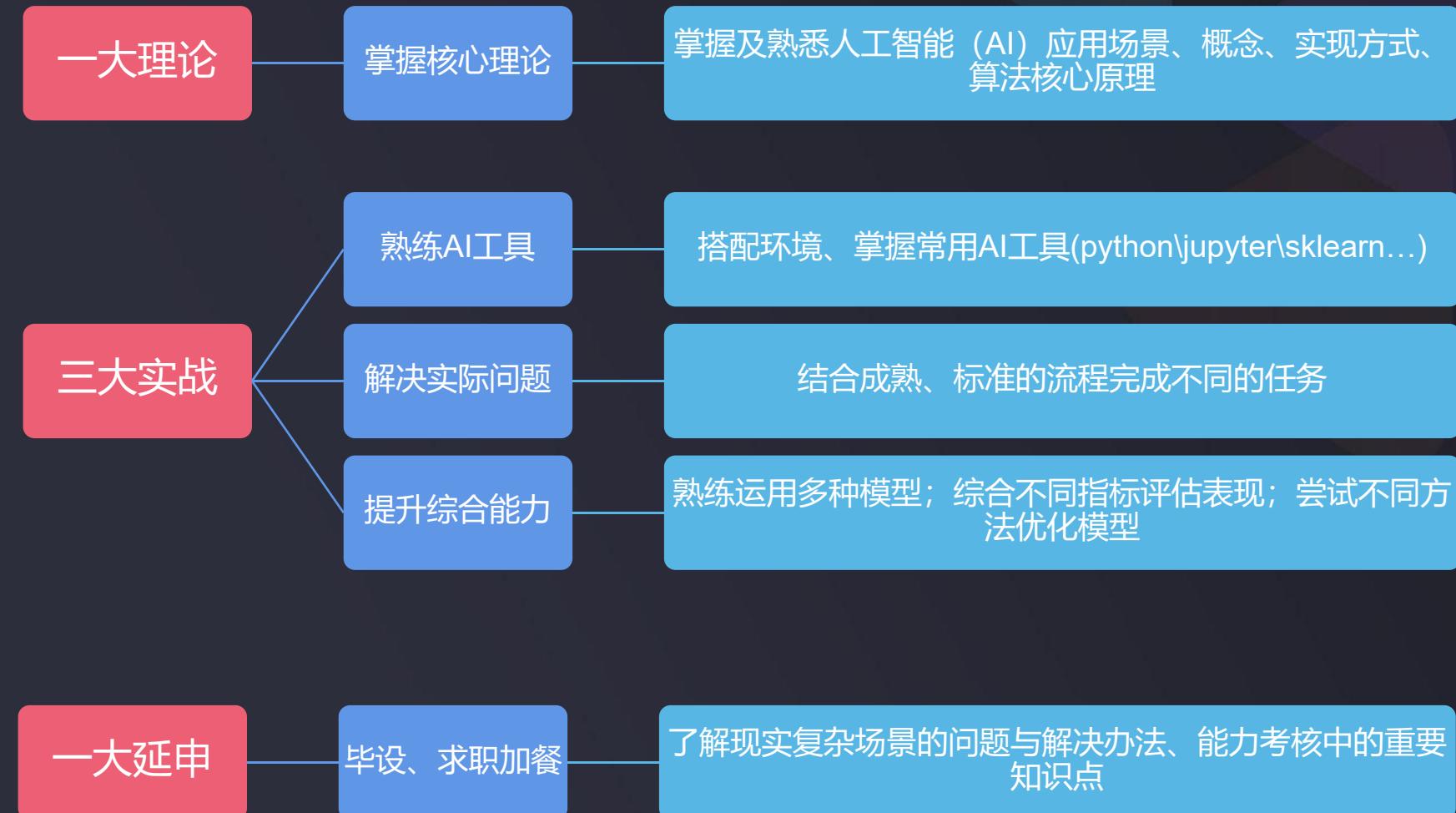
人工智能算法科学家

- 2019年福布斯精英科技榜U30
- 深圳市海外高层次人才
- 澳大利亚新南威尔士大学全奖博士
- 国际SCI收录学术文章十篇
- CSDN人工智能精英讲师（机器学习、深度学习）

Chapter 1 课程介绍及环境配置

-
- 1 --课程目标
 - 2 --课程内容概览
 - 3 --人工智能核心概念
 - 4 -- AI开发工具及实战基础

课程目标



课程目标

1、掌握核心知识点



课程目标

2、熟练使用AI工具



课程目标

3、掌握利用AI解决实际场景问题的能力

全面、
规范的
实践训
练流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

部分
案例

评估合理房价

异常消费检测

工作胜任与否划分

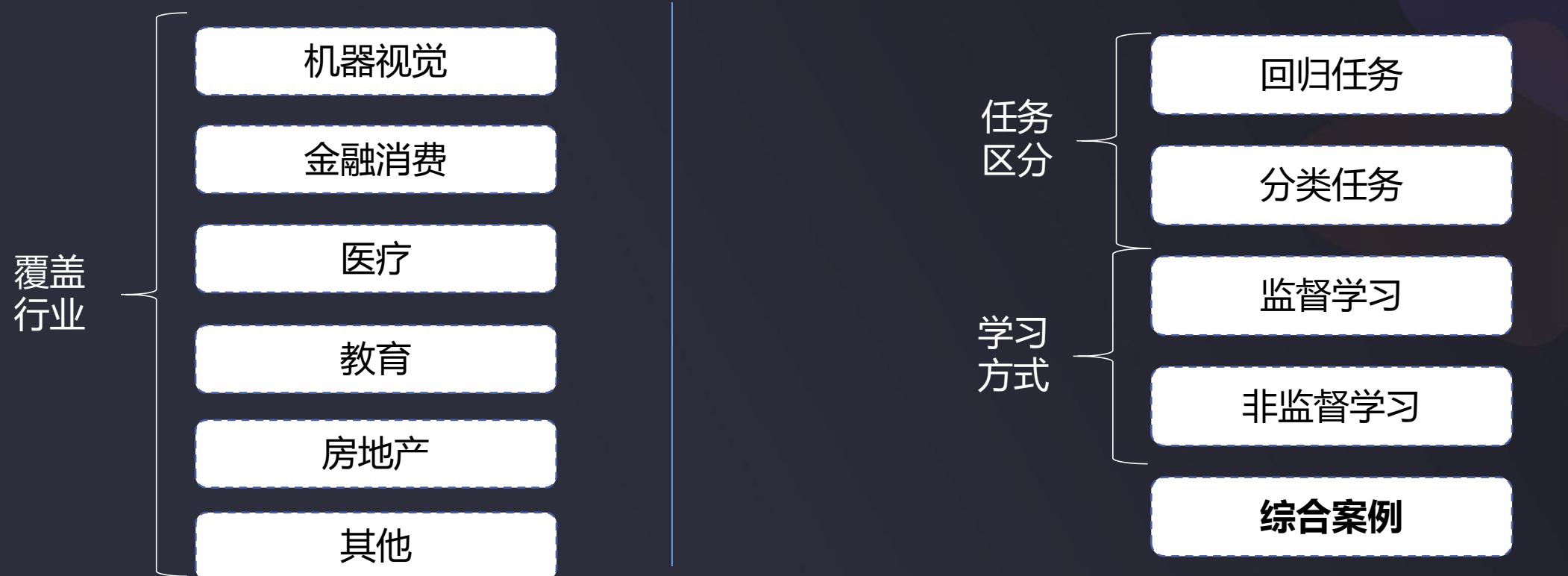
学生录取概率预测

日常图像分割

糖尿病病例筛选

课程目标

3、掌握利用AI解决实际场景问题的能力



课程目标

4、提升AI综合能力



课程目标

4、提升AI综合能力

表现优化

获取数据后，提前做哪些工作？

不同的场景，应该选什么模型合适？

如模型性能不佳，该如何优化？

模型评估

在实现回归功能时，怎样判断模型的性能？

在实现分类功能时，怎样判断模型的性能？

怎样将模型预测的结果可视化？

| Python3人工智能入门+实战提升：机器学习

下一个改革是人工智能！本课程是一门为AI新手量身定做的入门课

简单易懂的AI课程！不同行业的学员均适合，让你拥有AI解决问题的技能！

亲自整理与使用的
学习资料，
内容不定期更新，包含：
行业状况
知识干货
实战资料

(资料) Python3 掌握人工智能：入门与综合

目录

介绍

行业发展与就业情况

人工智能产业人才发展报告（2020
工信部最新）

工程开发

环境配置及软件安装常见问题汇总

常用pip安装源：
https://blog.csdn.net/dfly_zx/article/details/108060652

知识学习

课程>>Python3人工智能入门+实战提升：机器学习

模型评估、选择与优化

具体算法

行业应用

常用链接

(资料) Python3 掌握人工智能：入门与综合

介绍

本文档为围绕五门核心课程(《零基础学Python》、《Python3人工智能入门+实战提升：机器学习》、《Python3人工智能入门+实战提升：深度学习》、《资深人工智能面试官 教你拿下AI工作OFFER的5大秘决》、《深度学习与计算机视觉》)，整理的重要资料入口，不定期更新，如果无法访问，可加小助手微信反馈问题（微信号：ai_flare）。

备注：

- 1、一次给你一个压缩包，大概率会成垃圾。老师会整理出重要资料，附带简介，不定期更新，希望能帮到每位学习的同学
- 2、如有好的学习资料，欢迎添加小助手微信投稿

行业发展与就业情况

人工智能产业人才发展报告（2020工信部最新）

简介：国家工信部的AI行业发展报告，包含行业发展概览、现状、就业情况（薪资、能力要求）、趋势等，帮你明确行业格局，先人一步。

链接：<https://pan.baidu.com/s/1OpDdcrLy5oTGz0DqEzY7nQ>



Python3人工智能入门+实战提升：机器学习

Chapter 1 课程介绍及环境配置

赵辛

Chapter 1 课程介绍及环境配置

-
- 1 --课程目标
 - 2 --课程内容概述
 - 3 --人工智能核心概念
 - 4 -- AI开发工具及实战基础

课程概述

1、课程介绍及环境配置

课程目标和概述

人工智能的定义

所处阶段和实现方法

项目实战工具介绍

实战：搭配环境与测试



课程概述

2、机器学习之线性回归

机器学习介绍

回归分析

线性回归模型

核心原理讲解

代码实战

根据房子的信息来评估房子的合理价格

房源信息：
片区居住人口、人均收入、
房龄、房子面积、房间数量



课程概述

3、机器学习之逻辑回归

分类问题介绍

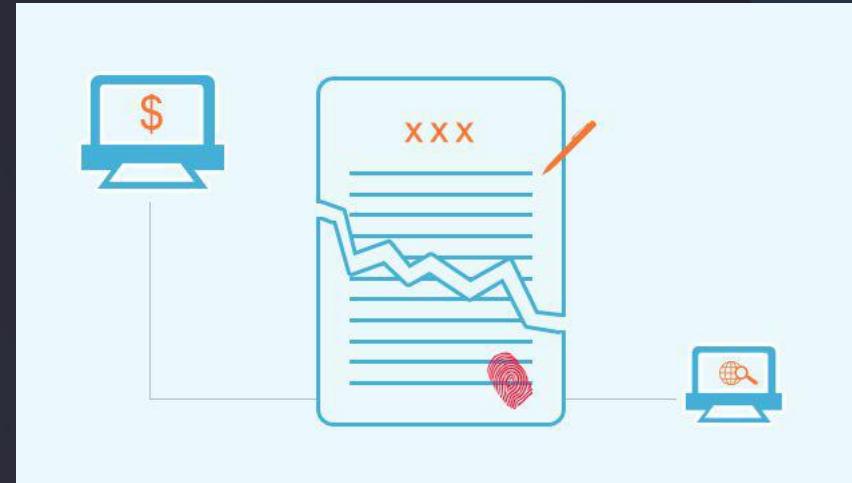
分类模型框架

逻辑回归分析

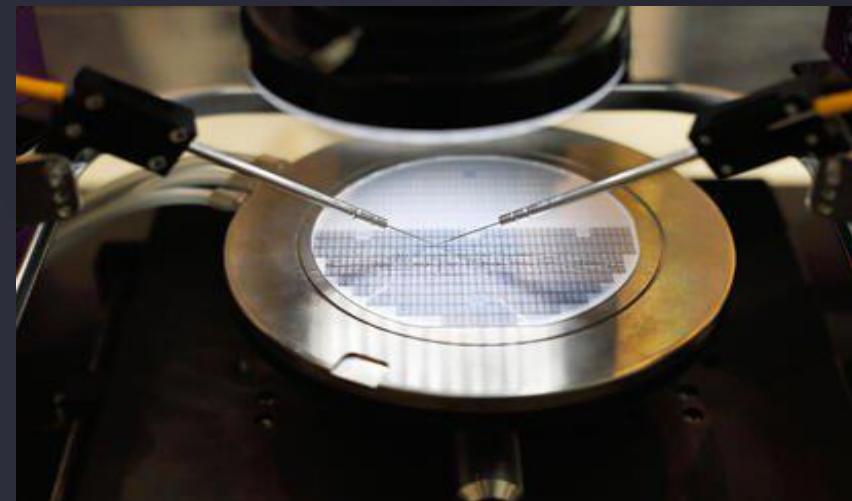
核心原理讲解

代码实战

综合项目知识加餐



预测消费是否异常



预测芯片质量

课程概述

4、其他常用分类技术

K近邻

决策树

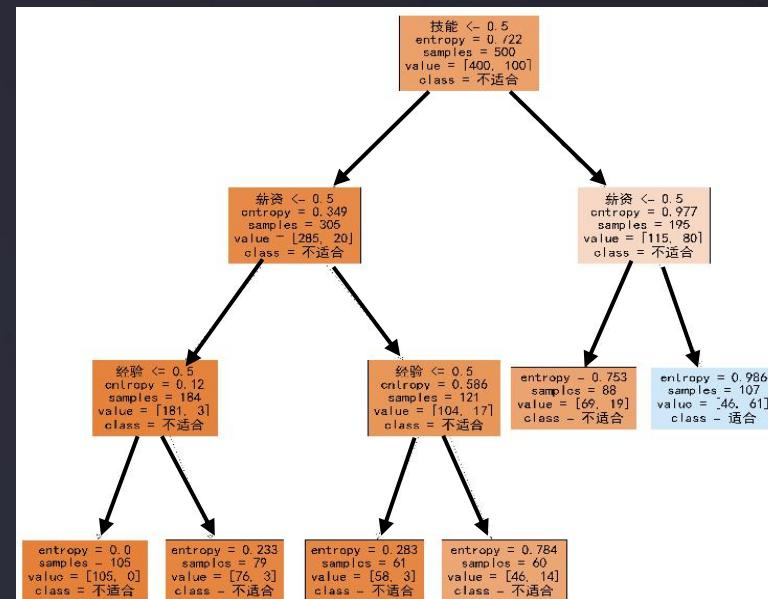
朴素贝叶斯

代码实战

技术对比与总结

朴素贝叶斯预测学生录取结果

学员信息 (测试样本)					预测概率			预测结果
成绩	学校	获奖	性别	英语	未录取	无奖学金录取	带奖学金录取	
2	1	1	1	1	0.152	0.346	0.502	带奖学金录取
2	1	1	1	0	0.203	0.400	0.397	无奖学金录取
2	1	1	0	0	0.158	0.455	0.387	无奖学金录取
2	1	0	0	0	0.388	0.447	0.166	无奖学金录取
2	0	0	0	0	0.595	0.293	0.112	未录取



决策树筛选合适求职者

课程概述

5、无监督学习与聚类分析

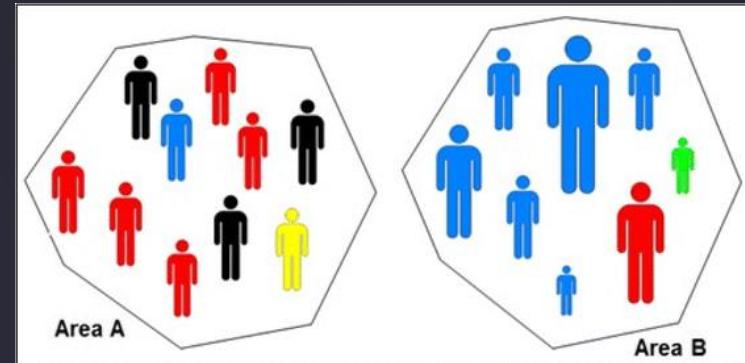
无监督学习

聚类分析

K均值聚类

代码实战

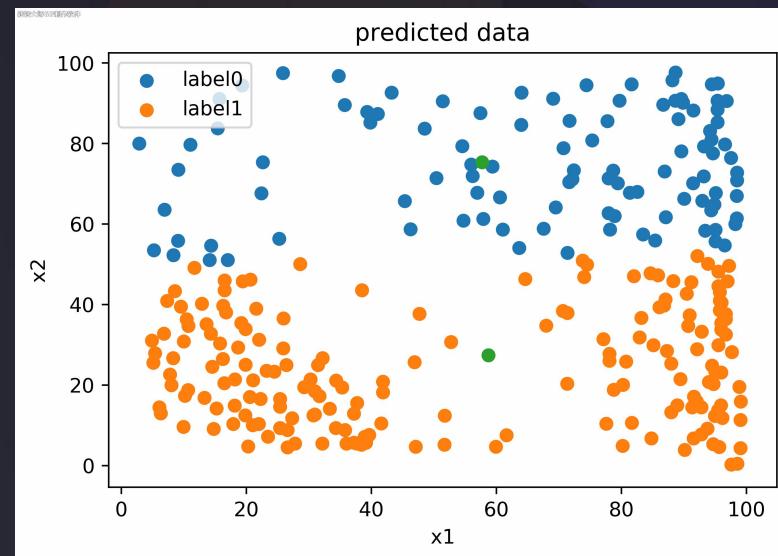
监督真的重要吗



目标用户的群体分类



图像分割



无监督数据聚类

课程概述

6、异常检测与数据降维

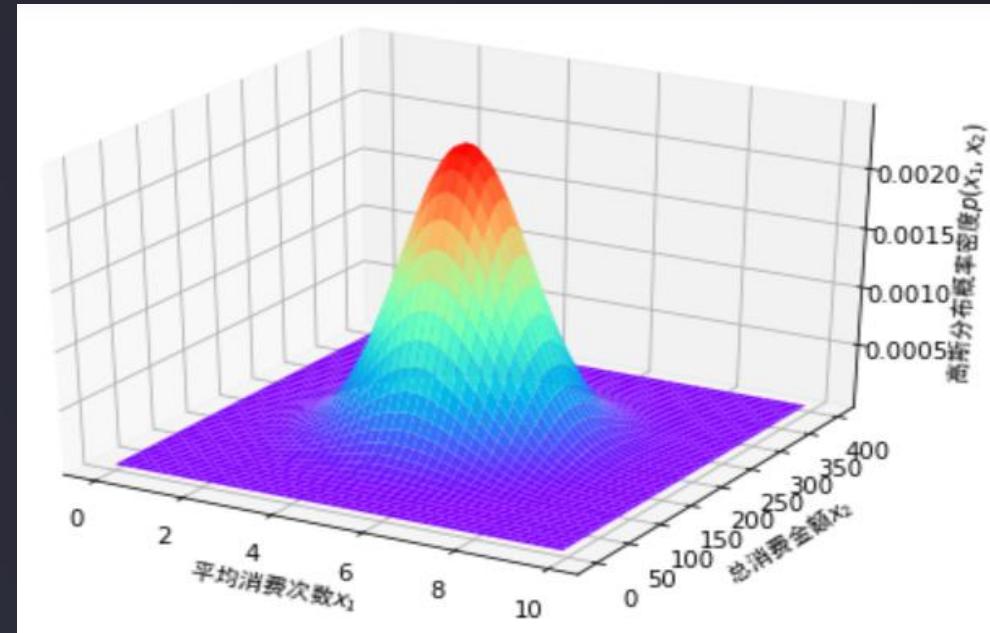
异常数据检测

高斯分布的概率密度

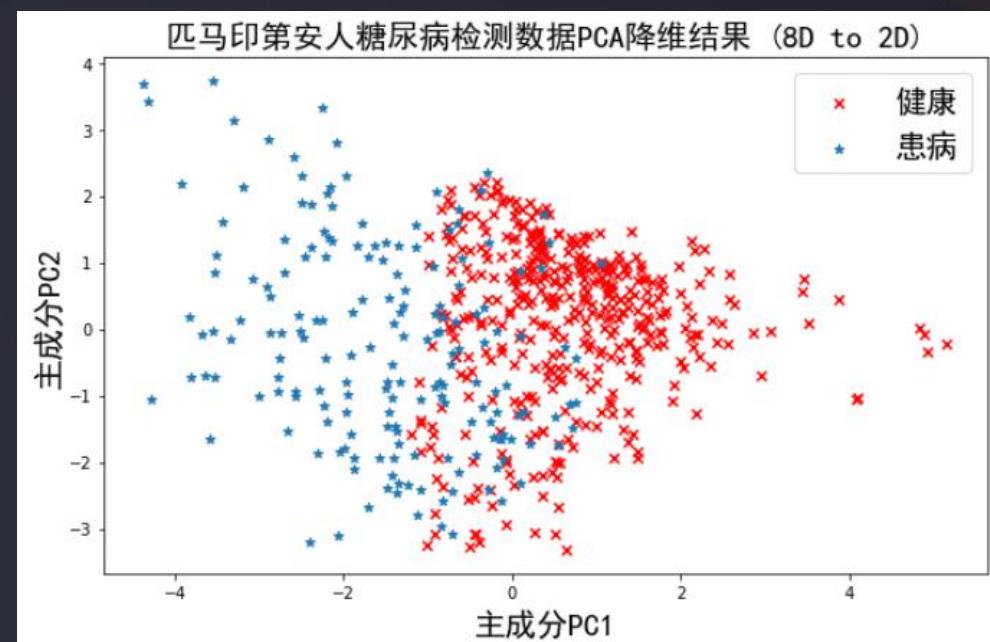
数据降维

主成分分析(PCA)

代码实战



商业异常消费检测



PCA降维实现糖尿病
检测

课程概述

7、模型评价与优化

数据预处理

上游决定下游，建模前五检查：

- 1、样本代表性
- 2、标签统一化
- 3、数据合理性
- 4、数据重要性
- 5、属性差异性

课程概述

7、模型评价与优化

数据预处理

三大核心问题

三大核心问题：

选用什么算法？

核心结构、参数如何设置？

模型表现不好，怎么办？

课程概述

7、模型评价与优化

数据预处理

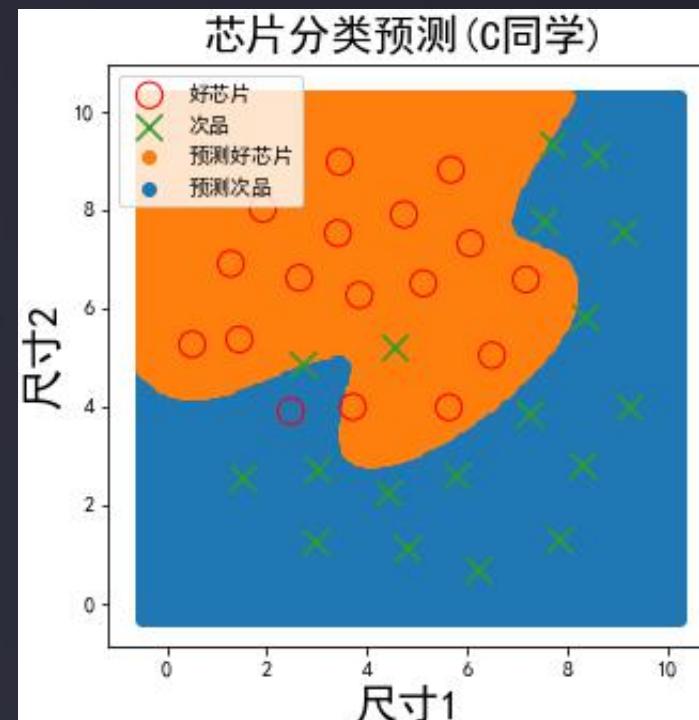
三大核心问题

数据分离

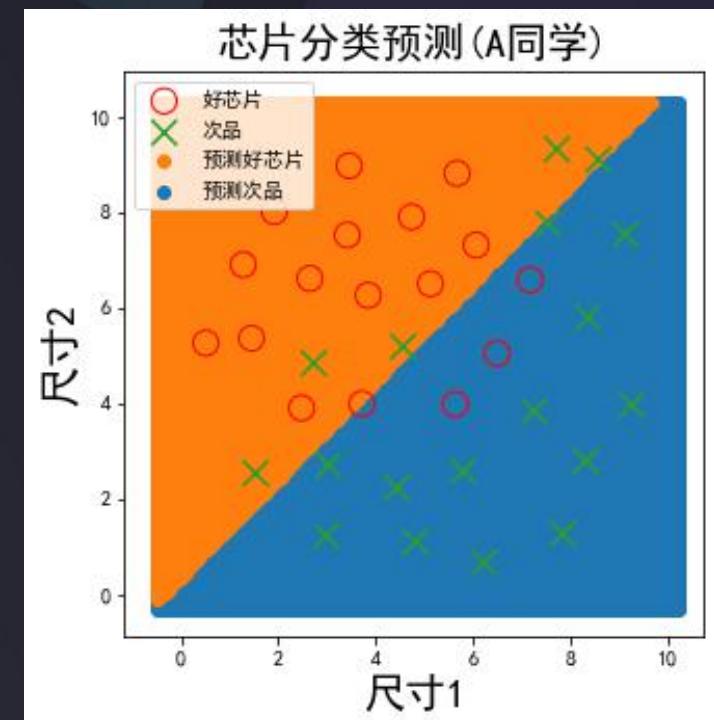
混淆矩阵

过拟合与欠拟合

代码实战



过拟合



欠拟合

知识点巩固

问题：本门课程实战将涉及哪些实战工具，它们的作用分别是什么？



Python3人工智能入门+实战提升：机器学习

Chapter 1 课程介绍及环境配置

赵辛

Chapter 1 课程介绍及环境配置

-
- 1 --课程目标
 - 2 --课程内容概览
 - 3 --人工智能核心概念
 - 4 -- AI开发工具及实战基础

|生活案例出发：身边的人工智能



微信聊天中的人工智能

生活案例出发：身边的人工智能



日常股票交易

	名称	品种类型	行业	多空	操作时间	操作类型	操作价格	数量
1	中兴通讯	股票	通信	-	2018-06-28 15:0	买入	12.92	77300
2	中兴通讯	股票	通信	-	2018-11-23 15:0	卖出	19.77	77300
3	中国神华	股票	采掘	-	2018-12-21 15:0	买入	18.91	20100
4	中国平安	股票	非银	-	2018-12-21 15:0	买入	58.34	6500
5	建设银行	股票	银行	-	2018-12-21 15:0	买入	6.30	60600
6	大族激光	股票	电子	-	2018-12-21 15:0	买入	30.09	12600
7	中国神华	股票	采掘	-	2019-03-26 15:0	卖出	19.42	20100
8	中国平安	股票	非银	-	2019-03-26 15:0	卖出	72.69	6500
9	建设银行	股票	银行	-	2019-03-26 15:0	卖出	6.81	60600
10	大族激光	股票	电子	-	2019-03-26 15:0	卖出	39.64	12600
11	隆基股份	股票	电气	-	2019-05-27 15:0	买入	23.90	12400
12	中兴通讯	股票	通信	-	2019-05-27 15:0	买入	29.29	10100
13	五粮液	股票	食品	-	2019-05-27 15:0	买入	102.00	2900
14	新希望	股票	农林	-	2019-05-27 15:0	买入	17.61	16800
15	贵州百灵	股票	医药	-	2019-05-27 15:0	买入	11.62	25500
16	国药股份	股票	医药	-	2019-05-27 15:0	买入	99.88	2200

	2017/12/29	2019/12/31	收益率	年化收益率
上证指数	3307	2857	-13.6%	-7.1%
沪深300指数	4030	4096	1.6%	0.8%
AI策略净值	1	1.968	96.8%	40.3%

|生活案例出发：身边的人工智能

人工智能应用

人脸识别，自动驾驶，医学图像诊断

智能机器人，AlphaGO

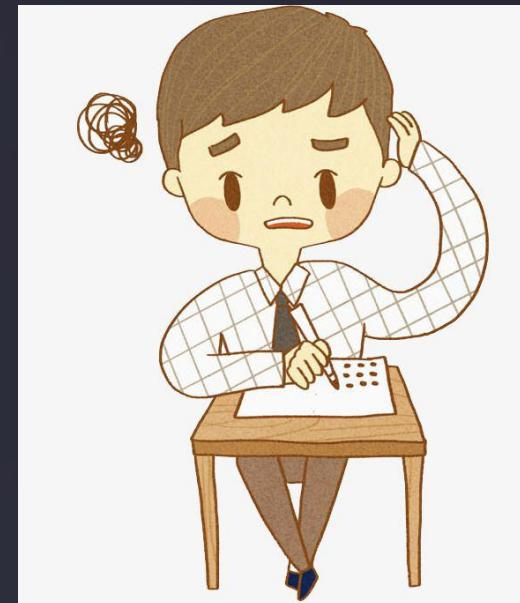
多语言翻译，智能客服，情感分类



|人工智能—维基百科定义

人工智能，亦称**智机器智能**，指由人制造出来的机器所表现出来的智能。

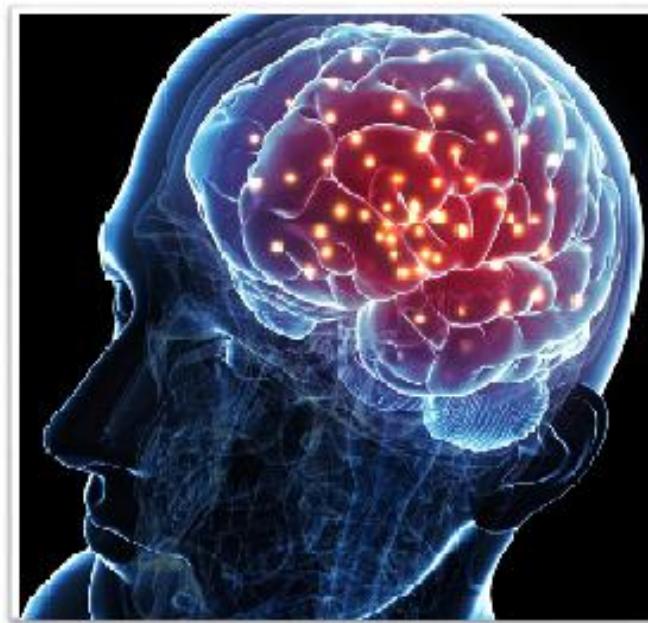
人工智能的核心问题包括建构能够跟人类似甚至超卓的推理、知识、规划、学习、交流、感知、移物、使用工具和操控机械的能力等



ARTIFICIAL



INTELLIGENCE



ARTIFICIAL
INTELLIGENCE



Intelligence: “The capacity to learn and solve problems”
(自主学习及解决问题的能力)

Artificial Intelligence: The simulation of human intelligence by machines
(机器对人类智能的模仿)

人工智能定义

人工智能就其本质而言，是机器对人的思维或行为过程的模拟，让它能像人一样思考或行动

遇到事件或任务

思考与学习

做出决策

从过去的信息中寻找规律（经验），将规律或经验吸收，并为未来的判断或决策提供依据。

学习、优化、决策

部分应用场景



AI 安防：利用计算机视觉技术和大数据分析犯罪嫌疑人生活轨迹及可能出现的场所



AI 金融：利用复杂的 AI 系统能极其迅速的做出交易决策。



AI 工业制造：机器人代替工人在危险场所完成工作，在流水线上高效完成重复工作

医疗、交通、教育...

人工智能发展阶段

通用人工智能（“强”人工智能）：

- 具备与人类同等智慧、或超越人类的人工智能，能表现正常人类所具有的所有智能行为。
 - 目前AI技术无法达到通用人工智能
 - 无法判断离通用人工智能还有多远

非通用人工智能（“弱”人工智能）

- 不需要具有人类完整的认知能力，甚至是完全不具有人类所拥有的感官认知能力
- 可处理特定问题，在特定应用中很厉害
 - 目前AI处于此阶段

人工智能实现方法

人工智能实现方法

符号人工智能&机器学习
(Symbolic Artificial Intelligence & Machine learning)

符号学习

基于逻辑与规则的学习方法，用一些特定的符号来表示现实的事物或者观念(符号不局限于图像文字，还包括了既定的逻辑、规则等)

$$a = 10, b = 20$$

- 根据既定的逻辑和顺序告诉机器接下来做什么
- 遵循if...then...原则

$$c = \begin{cases} a + b, & \text{if } a > b \\ a - b, & \text{if } a < b \end{cases}$$



$$c = ??$$

需要先知道或假设信息的逻辑、规律

机器学习

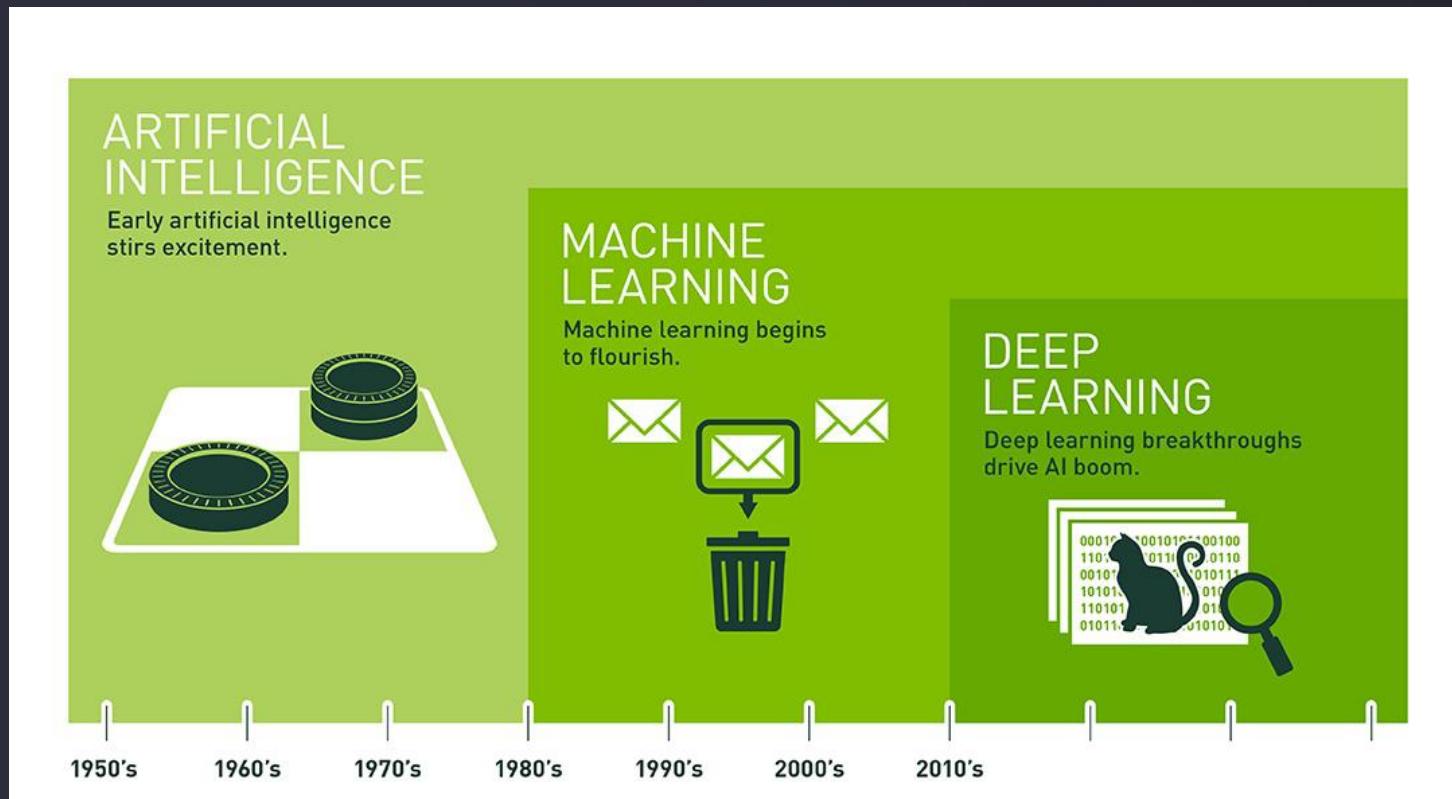
从数据中自动分析获得规律，并利用规律对未知数据进行预测或用于解决实际问题的方法。

- 从**数据**中学习规律，实现对原有推理的更新，实现“自我优化”
- 现阶段主流的AI学习算法
- 本门课程的核心

a	b	c
10	3	13
8	5	13
5	6	-1
3	7	-4
2	4	-2
1	9	-8

未来发展趋势：符号学习+机器学习

机器学习与深度学习的关系



机器学习是一种实现人工智能的方法，
深度学习是一种实现机器学习的技术。

机器学习：从数据中自动分析获得规律，并利用规律对未知数据进行预测或用于解决实际问题的方法。

深度学习：机器在对数据进行分析时，将引入类人类神经结构模型，实现对复杂数据的理解与推理，通常可应用于更为复杂的任务中。

知识点巩固

问题：思考以下三个案例最有可能使用了哪种学习方法。

- A.计算机检测到设备温度超过50度，发出报警信号
- B.计算机根据用户对历史邮件的归类，自动发现并过滤垃圾邮件
- C.计算机根据用户输入的图像，能够自动识别图像中的动物

|延伸阅读资料

1、浅谈人工智能：现状、任务、构架与统一

<https://blog.csdn.net/dingyahui123/article/details/78446329>

2、5分钟了解人工智能

<https://www.youtube.com/watch?v=2ePf9rue1Ao>

3、比较符号学习与机器学习

<https://analyticsindiamag.com/understanding-difference-symbolic-ai-non-symbolic-ai/>



Python3人工智能入门+实战提升：机器学习

Chapter 1 课程介绍及环境配置

赵辛

Chapter 1 课程介绍及环境配置

-
- 1 --课程目标
 - 2 --课程内容概览
 - 3 --人工智能核心概念
 - 4 -- AI开发工具及实战基础

核心开发
环境



| Python

Python是一种解释型的、面向对象的、移植性强的高级程序设计语言。

开发者：吉多·范罗苏姆 (Guido van Rossum)

解释型：不需要编译成二进制代码，直接从源代码运行程序

面向对象：Python同时支持面向过程和面向对象编程。

可移植性：Python可以跨操作平台无差别的运行

高层语言：无须考虑诸如如何管理程序使用的内存一类的底层细节

www.python.org/



Python



1. 简单易用，学习成本低

- 开发效率高
- 高级语言，功能强大
- 解释型语言，能跨平台
- 可扩展性强
- 可嵌入性

● 运行速度慢
代码加密困难

缺陷

Rank	Language	Type	Score
1	Python	🌐💻⚙️	100.0
2	Java	🌐💻⚙️	96.3
3	C	💻⚙️	94.4
4	C++	💻⚙️	87.5
5	R	💻	81.5
6	JavaScript	🌐	79.4
7	C#	🌐💻⚙️	74.5
8	Matlab	💻	70.6
9	Swift	💻	69.1
10	Go	🌐💻	68.0

IEEE Spectrum (电气和电子工程师协会)
编程语言 Top 10

Anaconda

Anaconda是一个方便的python包管理和环境管理软件

- 支持 Linux, Mac, Windows
- 可以很方便地实现多版本python并存、切换以及各种第三方包的快速安装



特点:

- 使用方便、安装过程简单
- 兼容不同系统、可同时实现包管理、环境管理的功能

www.anaconda.com/

Jupyter notebook

Jupyter Notebook (此前被称为 IPython notebook) 是一个开源的 Web 应用程序，允许开发者方便的创建、共享和执行代码。

- 可以实时写代码、运行代码、查看结果，并可视化数据

特点：

- 极其适合数据分析（分块执行、方便调试）
- 远程运行
- 交互式展现

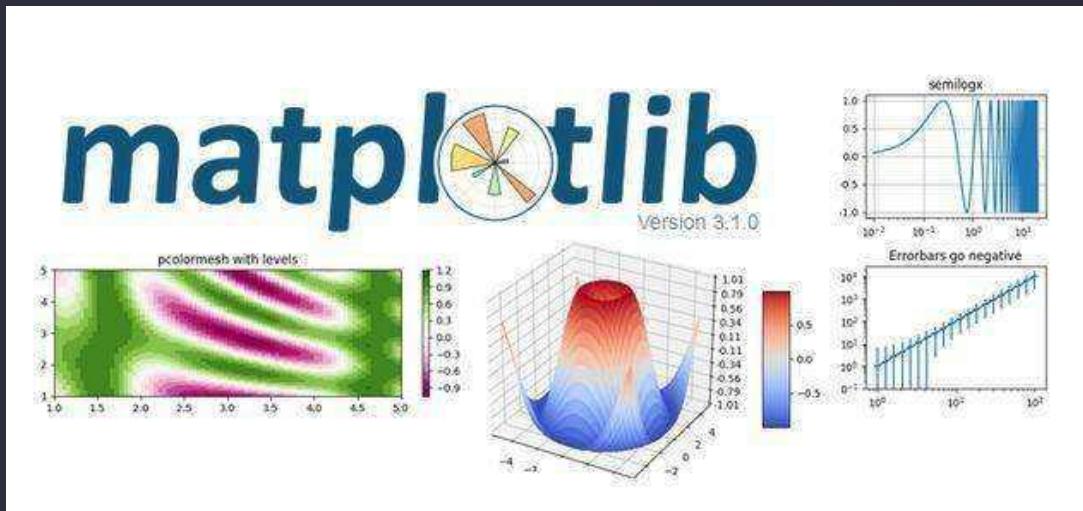
<https://jupyter.org/>



|基础工具包



Pandas提供了大量能使我们快速便捷地处理数据的函数和方法，可用于快速实现数据导入/出、索引。
www.pypandas.cn/



通过 Matplotlib，开发者可以仅需要几行代码，便可以生成绘图、散点图、曲线/曲面图、直方图等。
www.matplotlib.org.cn/



可用来存储和处理大型矩阵，支持大量的维度数组与矩阵运算，对数组运算提供大量的数学函数库。
www.numpy.org.cn/

AI开发实战基础

1. 下载、安装Anaconda
2. 新建开发环境、指定安装python版本
 - 配置：`conda create -n env_name python=3.7.0`
3. 安装jupyter-notebook
4. Python基本语法；numpy、pandas、matplotlib安装与测试
 - 配置：`pip/conda install package_name`

国内镜像源：https://blog.csdn.net/dfly_zx/article/details/108060652

|AI开发实战基础

1. Python基本语法：基本运算、列表生成、函数、模块引入
2. Matplotlib：安装、引入、使用
3. Numpy：安装、引入、使用
4. Pandas：安装、引入、使用



Python3人工智能入门+实战提升：机器学习

Chapter 2 回归分析与线性回归

赵辛

Chapter 2 机器学习之线性回归

-
- 1 --机器学习
 - 2 --回归分析
 - 3 --模型求解与线性回归
 - 4 --梯度下降法
 - 5 --实战准备
 - 6 --实战（一）基于面积的单因子房价预测
 - 7 --实战（二）现实多因子房价预测

现实问题思考--营业额预测

店铺A第一周营业额为5000，每周增长10%，第10周是多少？

$$y = 5000 \times 1.1^{x-1}$$

第x周	营业额y
1	5,000
2	5,500
3	6,050
4	6,655
5	7,321
6	8,053
7	8,858
8	9,744
9	10,718
10	???

传统算法：

第一周营业额 y_0

$$y = y_0 \times 1.1^{x-1}$$



结果

机器学习：

第几周

营业额



$F(x)$ → 结果

现实问题思考--股票怎么买

小张是个金融小白，想在股市中赚钱，他该怎么做？

传统办法：

找一位有经验的前辈、老师，学习经验与知识，然后再选股

机器学习：

股票基本信息

股票涨跌数据



判断每只股票接下来一段时间会上涨，还是下跌

名称	品种类型	行业	多空	操作时间	操作类型	操作价格	数量
汇顶科技	股票	电子	-	2019-11-20 15	买入	215.70	1300
华友钴业	股票	有色	-	2019-11-20 15	买入	51.40	5700
步长制药	股票	医药	-	2019-11-20 15	买入	39.44	7400
潍柴动力	股票	汽车	-	2019-11-20 15	卖出	248.22	500
泸州老窖	股票	食品	-	2019-11-20 15	卖出	2820.76	100
长安汽车	股票	汽车	-	2019-11-20 15	买入	67.42	4300
五粮液	股票	食品	-	2019-11-20 15	卖出	2391.77	100
石基信息	股票	计算机	-	2019-11-20 15	买入	696.53	400
大华股份	股票	电子	-	2019-11-20 15	卖出	759.44	200
奥飞娱乐	股票	传媒	-	2019-11-20 15	买入	70.34	4100
大北农	股票	农林	-	2019-11-20 15	卖出	42.26	2600
海康威视	股票	电子	-	2019-11-20 15	卖出	691.33	100
欧菲光	股票	电子	-	2019-11-20 15	买入	263.90	500
赣锋锂业	股票	有色	-	2019-11-20 15	买入	241.84	1200
天齐锂业	股票	有色	-	2019-11-20 15	买入	187.67	1500
三一重工	股票	机械	-	2019-11-20 15	买入	921.00	400

|现实问题思考—自动图像分类

目标：

以下六组图片，按照自己喜爱的方式分成两组



分组一：站着或非站着

分组二：白色或黄色

分组三：吐舌头或不吐舌头

机器学习应用与概念——机器学习

机器学习是一种实现人工智能的方法。
从数据中寻找规律、建立关系，根据建立的关系去解决问题。



| 机器学习应用与概念——机器学习应用场景

- ✓ 数据挖掘
- ✓ 无人驾驶
- ✓ 机器视觉
- ✓ 机器人
- ✓ 语言理解
- ✓ 病例分析

实现人工智能的
主流方法!

机器学习应用与概念——学习框架

训练数据x, y

第x周	营业额y
1	5,000
2	5,500
3	6,050
4	6,655
5	7,321
6	8,053
7	8,858
8	9,744
9	10,718
10	???

自动求解x, y数据关系

$$y = f(x)$$

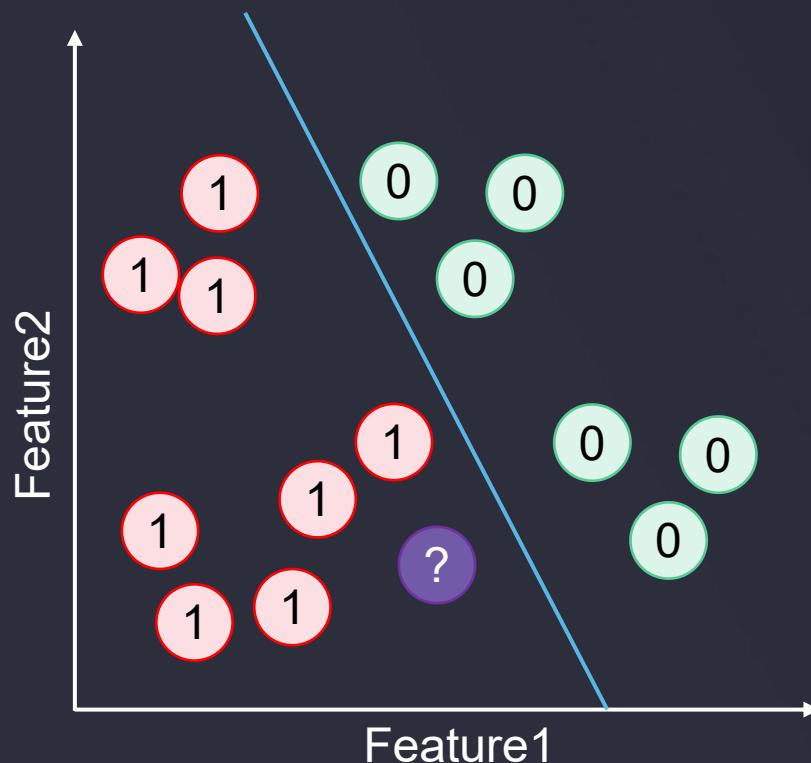
新数据预测

机器学习应用与概念——四大学习方法类别

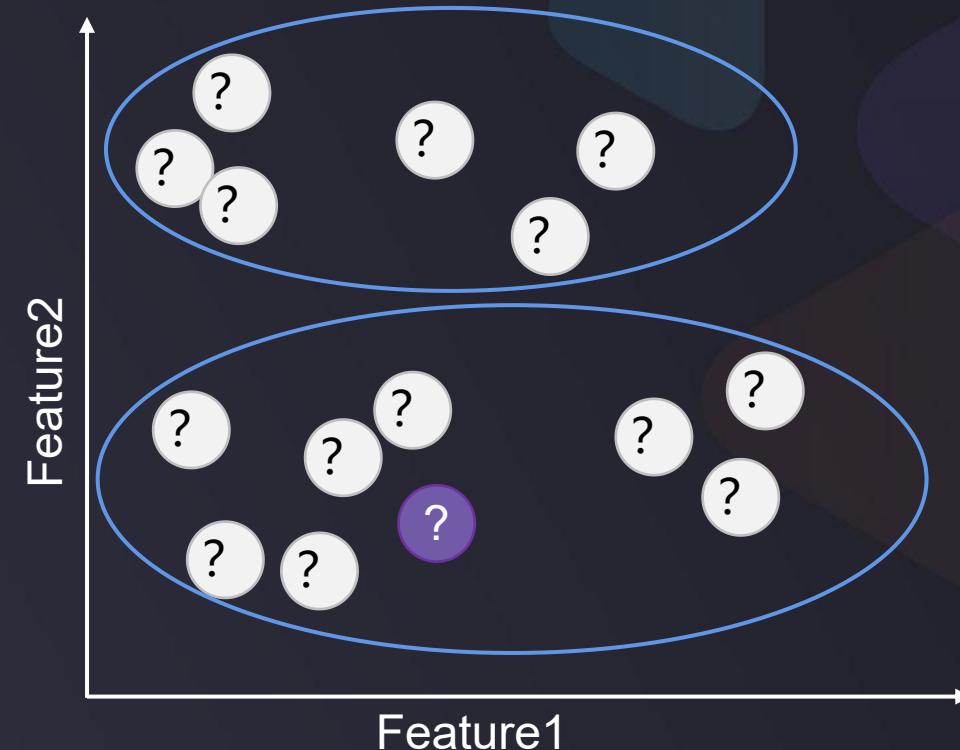
- 监督学习 (Supervised Learning)
- 无监督学习 (Unsupervised Learning)
- 半监督学习 (Semi-supervised Learning)
- 强化学习 (Reinforcement Learning)

训练是否有
正确的结果
(标签-label)

机器学习应用与概念——监督、无监督学习

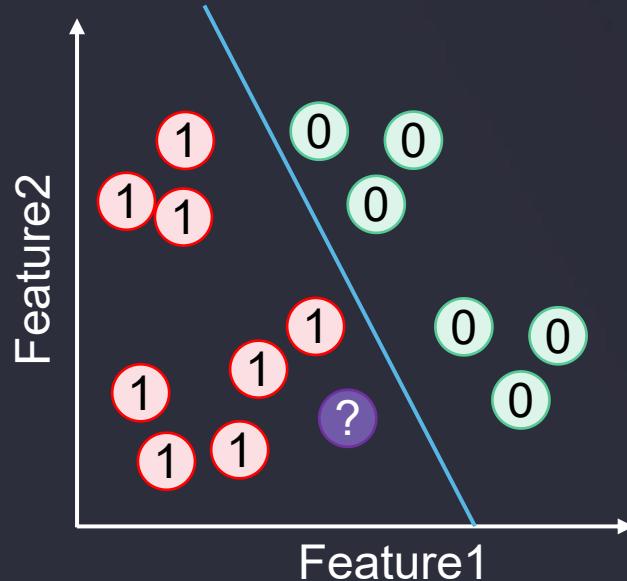


监督学习
包括正确的结果

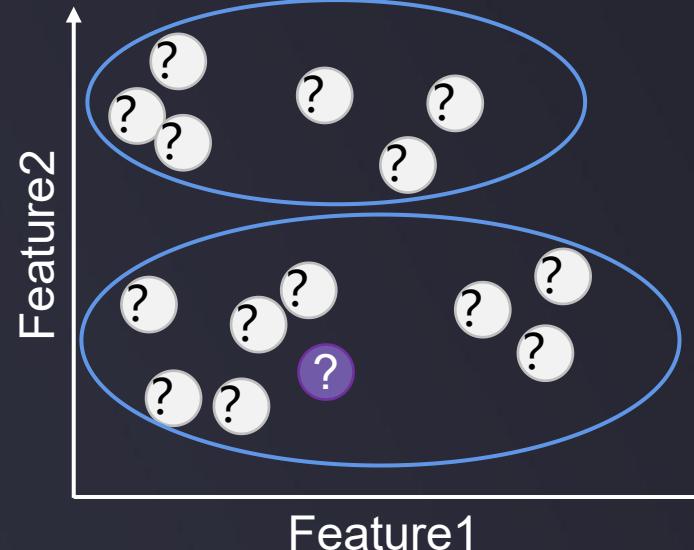


无监督学习
不包括正确的结果

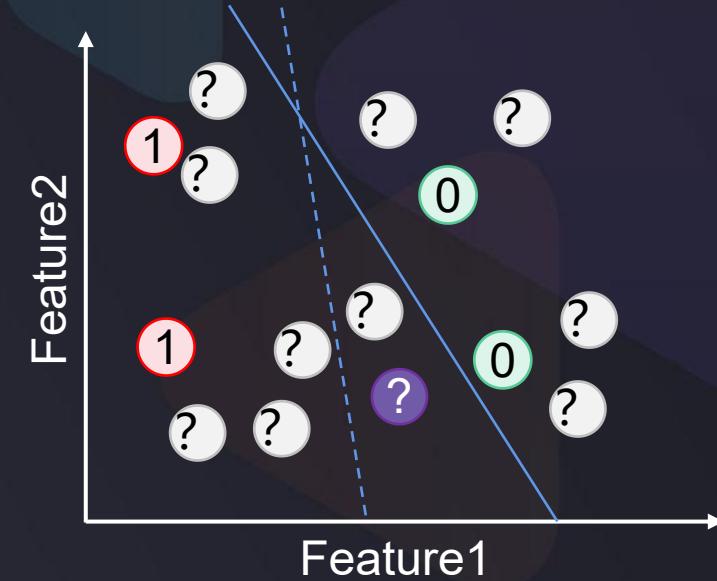
机器学习应用与概念——半监督学习



监督学习
包括正确的结果

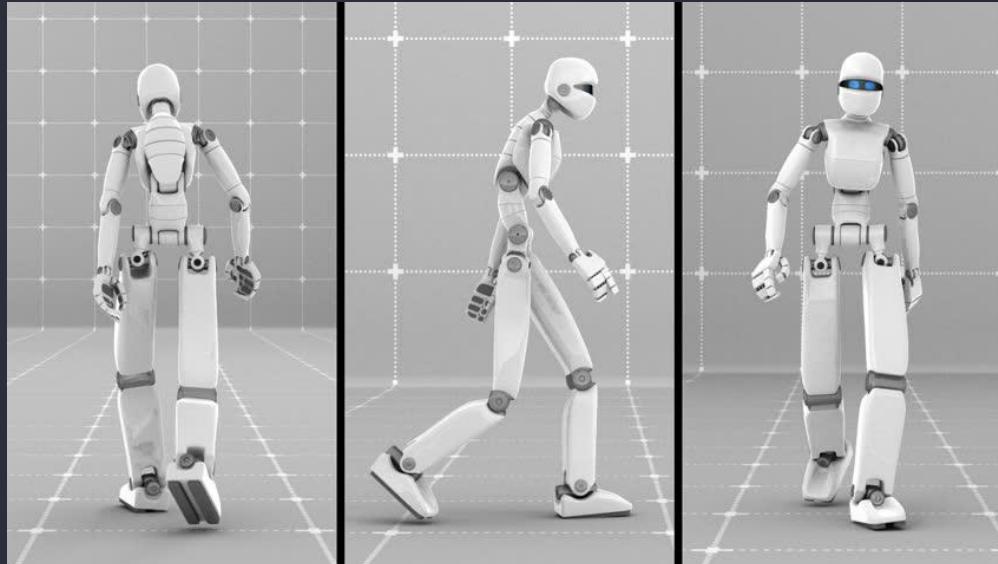


无监督学习
不包括正确的结果



半监督学习
包括少量正确的结果

机器学习应用与概念——强化学习



左, 右, 直行, 左, 右, 左, 直行

GOOD

左, 直行, 左, 左, 右, 直行, 直行

BAD

左, 右, 直行, 左, 左, 左, 直行

+3

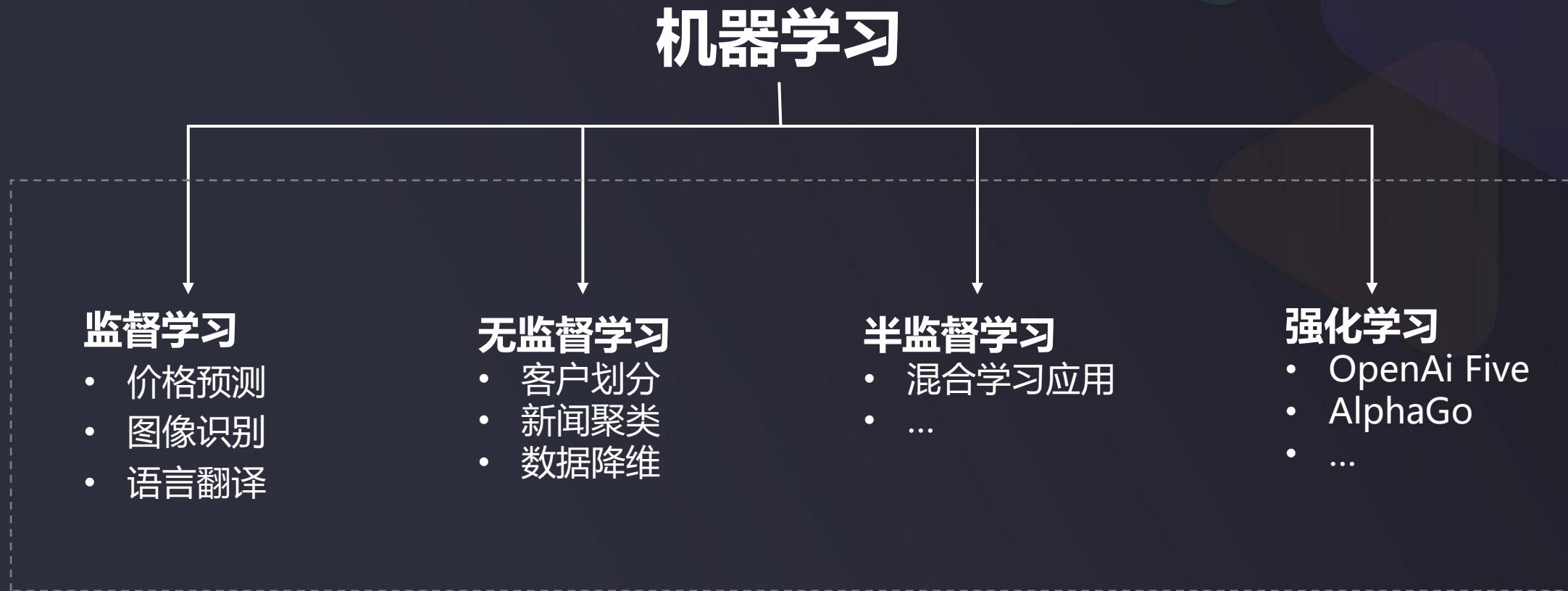
左, 直行, 左, 左, 右, 直行, 直行

-3

程序初始化

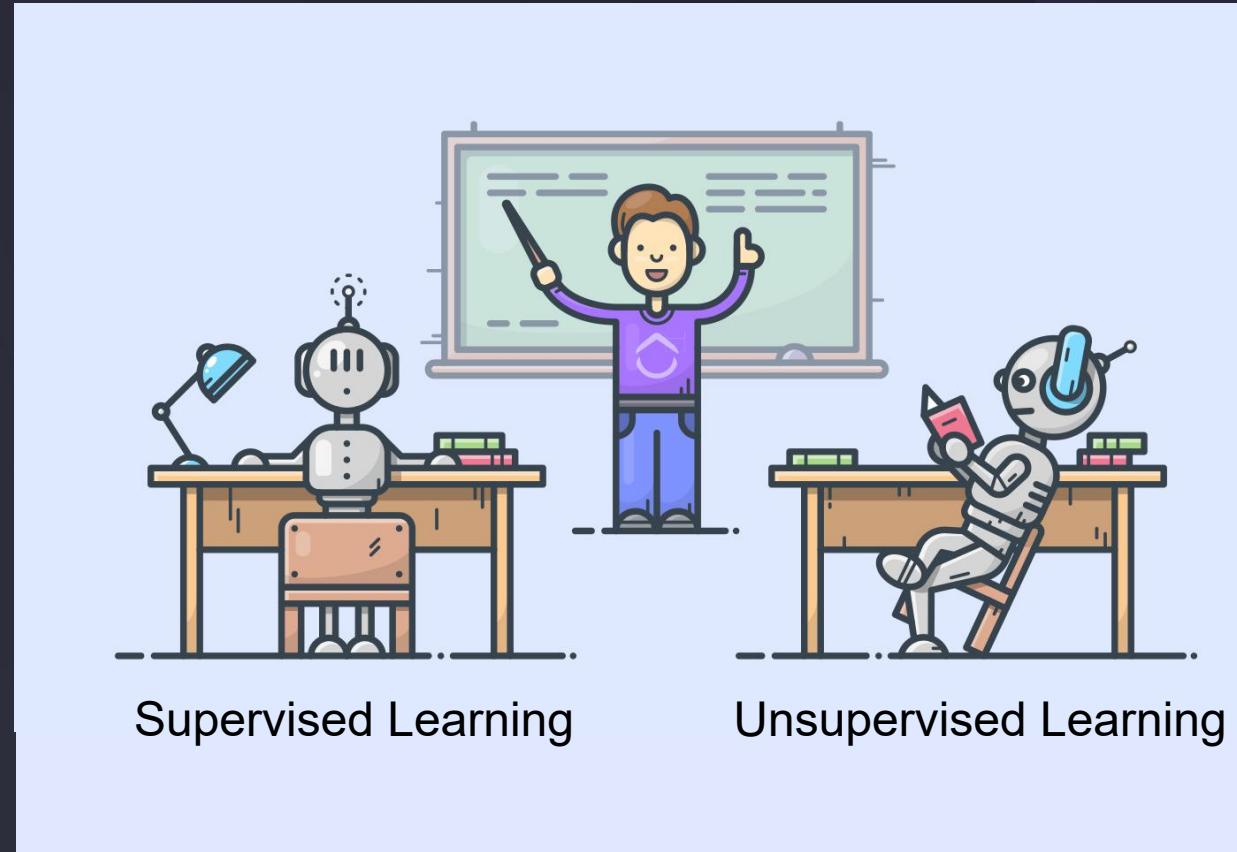
- 根据执行效果给与奖励/惩罚（分数）
- 程序逐步寻找获得高分的方法

机器学习应用与概念——四大学习方法



机器学习应用与概念——即将学习

- 监督学习
 - 线性回归
 - 逻辑回归
 - 决策树
 - 朴素贝叶斯
 - KNN
- 无监督学习
 - 聚类算法
 - PCA降维
 - 异常检测



| 机器学习应用与概念——知识巩固

问题：以下哪些应用采用了机器学习？

- A.医疗患者CT照片诊断
- B.谷歌翻译
- C.AlphaGo
- D.机器工作温度监控与报警



Python3人工智能入门+实战提升：机器学习

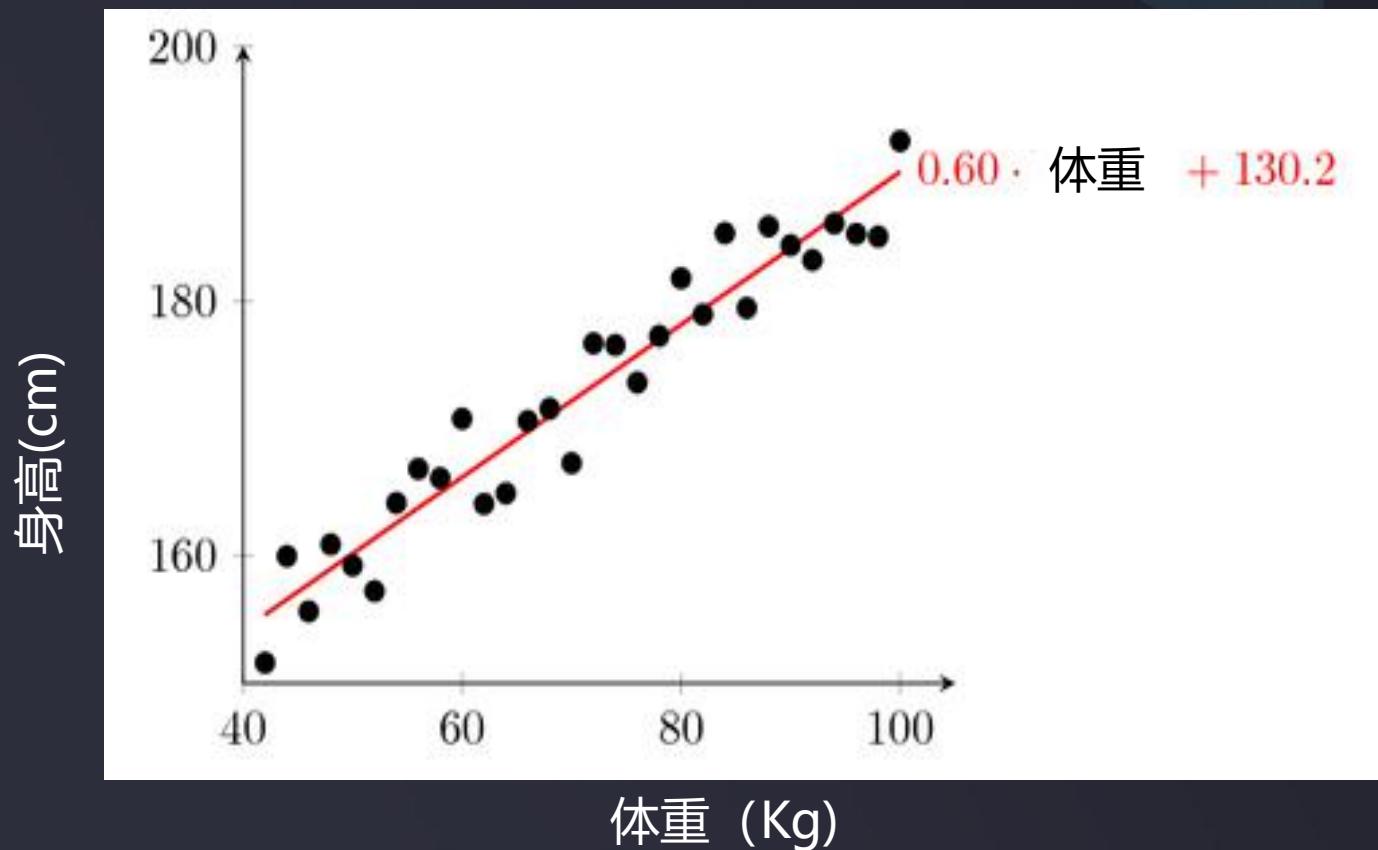
Chapter 2 回归分析与线性回归

赵辛

Chapter 2 回归分析与线性回归

-
- 1 --机器学习
 - 2 --回归分析
 - 3 --模型求解与线性回归
 - 4 --梯度下降法
 - 5 --实战准备
 - 6 --实战（一）基于面积的单因子房价预测
 - 7 --实战（二）现实多因子房价预测

现实问题思考--体重预测身高



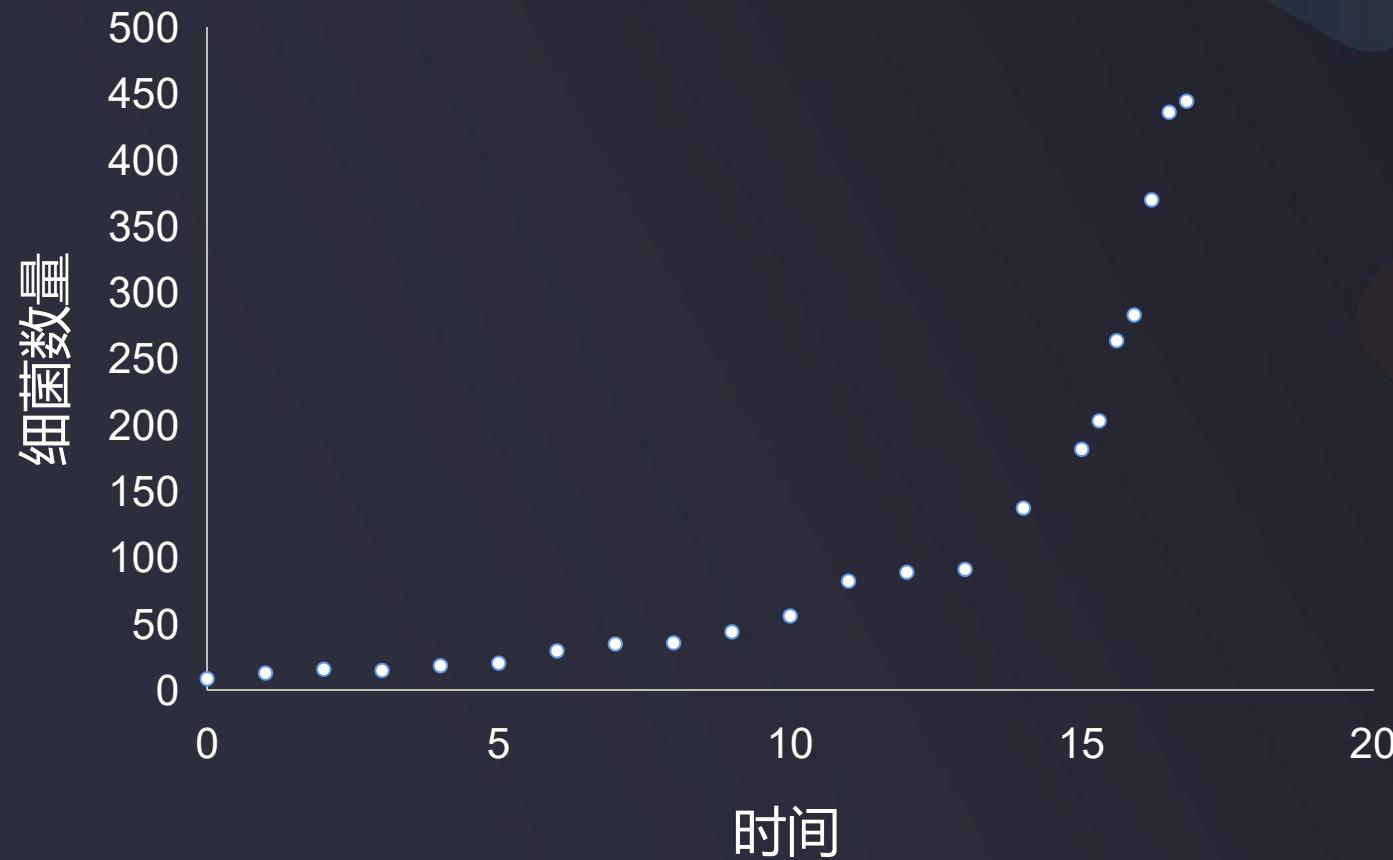
更多因素：性别、所在城市、父母身高等

| 现实问题思考--住宅面积预测售价



更多因素：房间数量、房屋年龄、人口密度、交通便利程度等

现实问题思考--细菌增长数量预测



更多因素：环境温度、营养液余量

回归分析

基于输入数据，确定变量间相互依赖的定量关系

输入数据： x, y

$x_1, x_2 \cdots x_n$

}

$$y = f(x_1, x_2 \cdots x_n)$$

举例：

面积	房间数	房龄	其他因子	售价y
80	2	5	...	1,000,000
50	1	10	...	800,000
90	3	6	...	1,000,000
90	2	1	...	110,000
40	1	7	...	700,000

$$\text{合理售价} = f(\text{面积}, \text{房间数}, \text{房龄}, \text{其他因子})$$

回归分析

$$y = f(x_1, x_2 \dots x_n)$$

函数关系:

- 线性回归: $y = ax + b$
- 非线性回归: $y = ax^2 + bx + c$

举例: 小明开始工资1000, 每周增长, 第t周的工资是多少(第一周开始算增长) ?

线性回归: 每周增长100

非线性回归: 每周是上周的1.1倍

回归分析

$$y = f(x_1, x_2 \dots x_n)$$

变量数量:

- 一元回归: $y = f(x)$
- 多元回归: $y = f(x_1, x_2 \dots x_n)$

合理售价 = $f(\text{面积})$

合理售价 = $f(\text{面积}、\text{房间数}、\text{房龄}、\text{其他因子})$

回归问题求解

问题：面积100平米售价120万是否值得投资？

面积(x)	售价(y)
68	414,592
95	956,877
...	...
102	1,123,582
115	???
130	893,667
...	...

1. 确定x、y间的定量关系

$$y = f(x)$$

2. 根据关系预测合理价格

$$y(x = 100) = f(x = 100)$$

3. 做出判断

$$y(x = 100) \quad 1200000$$



Python3人工智能入门+实战提升：机器学习

Chapter 2 回归分析与线性回归

赵辛

Chapter 2 回归分析与线性回归

-
- 1 --机器学习
 - 2 --回归分析
 - 3 --模型求解与线性回归
 - 4 --梯度下降法
 - 5 --实战准备
 - 6 --实战（一）基于面积的单因子房价预测
 - 7 --实战（二）现实多因子房价预测

回归问题求解

问题：面积100平米售价120万是否值得投资？

面积(x)	售价(y)
68	414,592
95	956,877
...	...
102	1,123,582
115	???
130	893,667
...	...

1. 确定x、y间的定量关系

$$y = f(x)$$

2. 根据关系预测合理价格

$$y(x = 100) = f(x = 100)$$

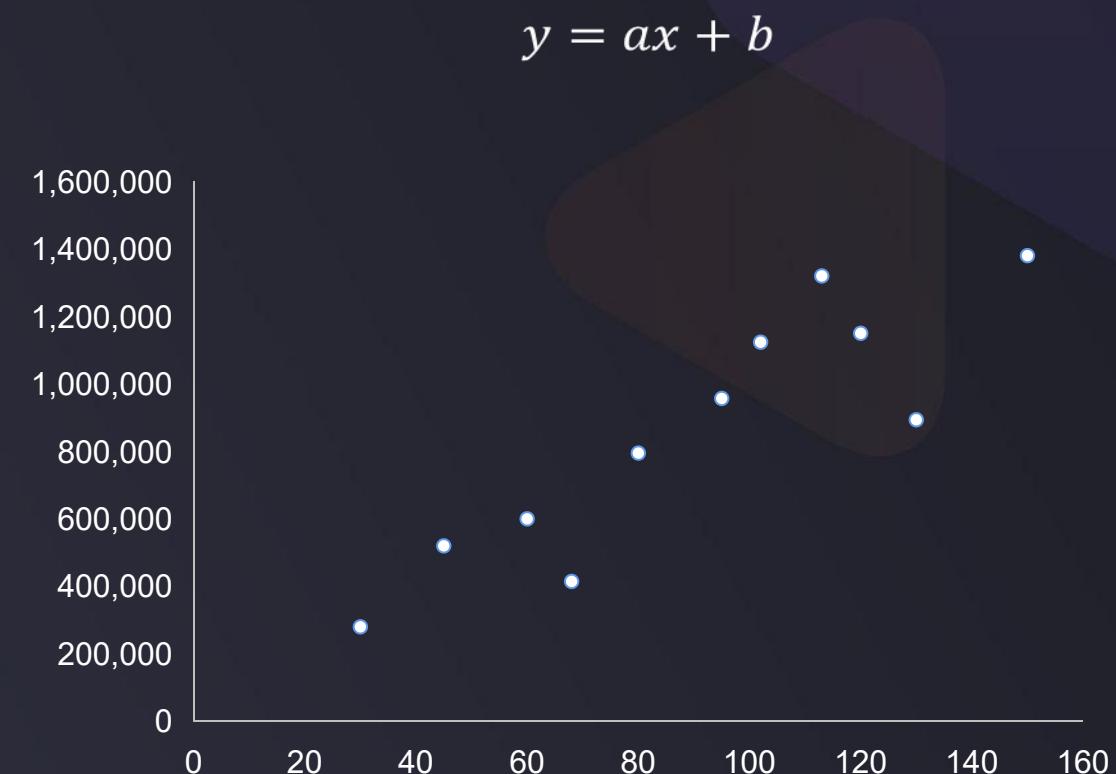
3. 做出判断

$$y(x = 100) \quad 1200000$$

线性回归问题求解

问题：面积100平米售价120万是否值得投资？

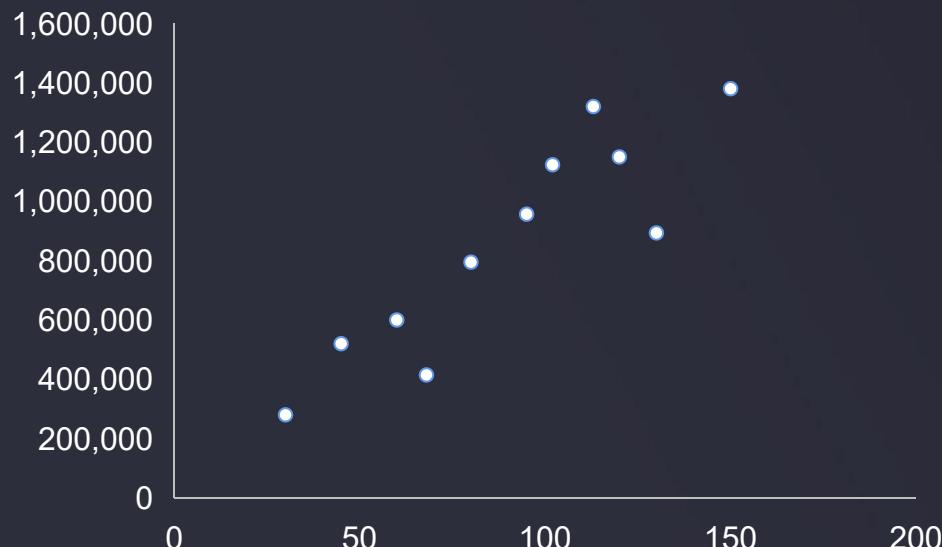
面积(x)	售价(y)
68	414,592
95	956,877
...	...
102	1,123,582
115	???
130	893,667
...	...



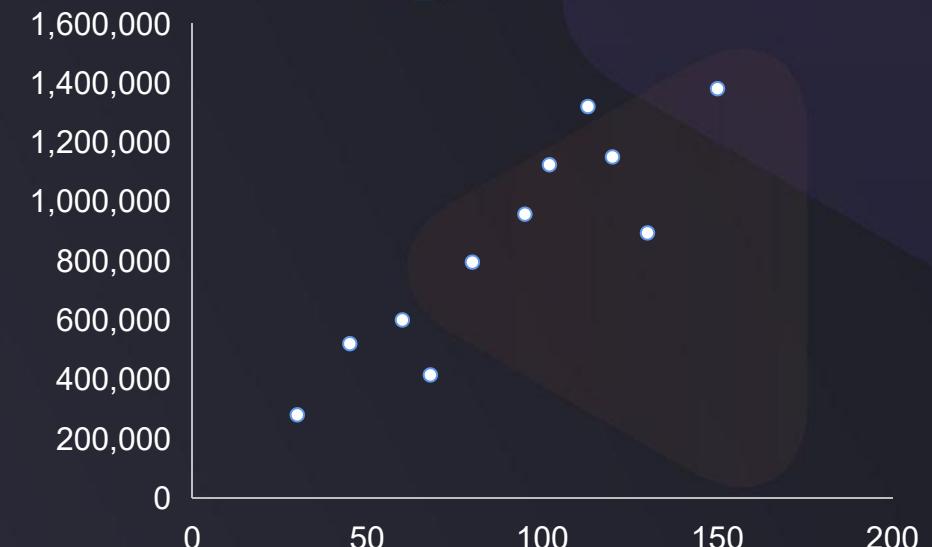
寻找合理的a 和 b

线性回归问题求解

$$y = ax + b$$



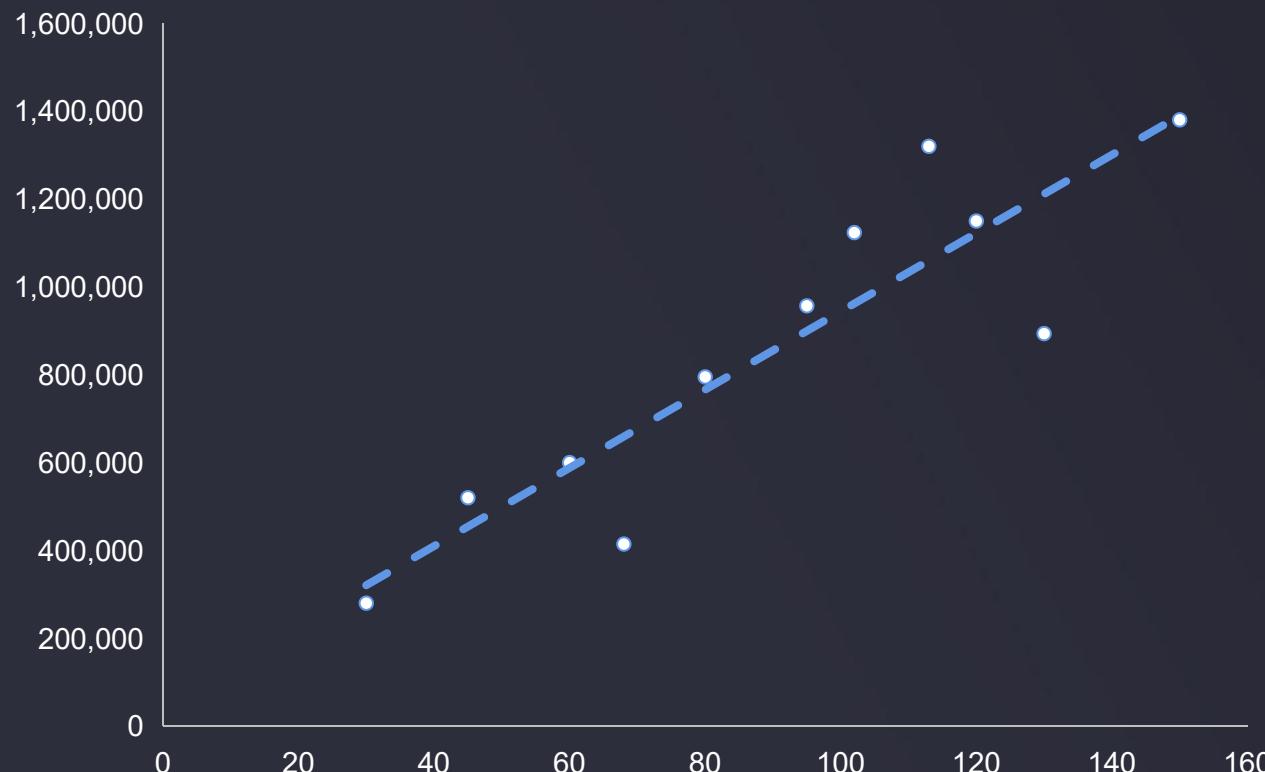
$$a = 0 ; b = 1000000$$



$$a = -100000 ; b = 1500000$$

线性回归

假设有 m 套房子每套房子面积 x_i 对应的实际售价为 y_i
线性模型预测的售价为 y'_i



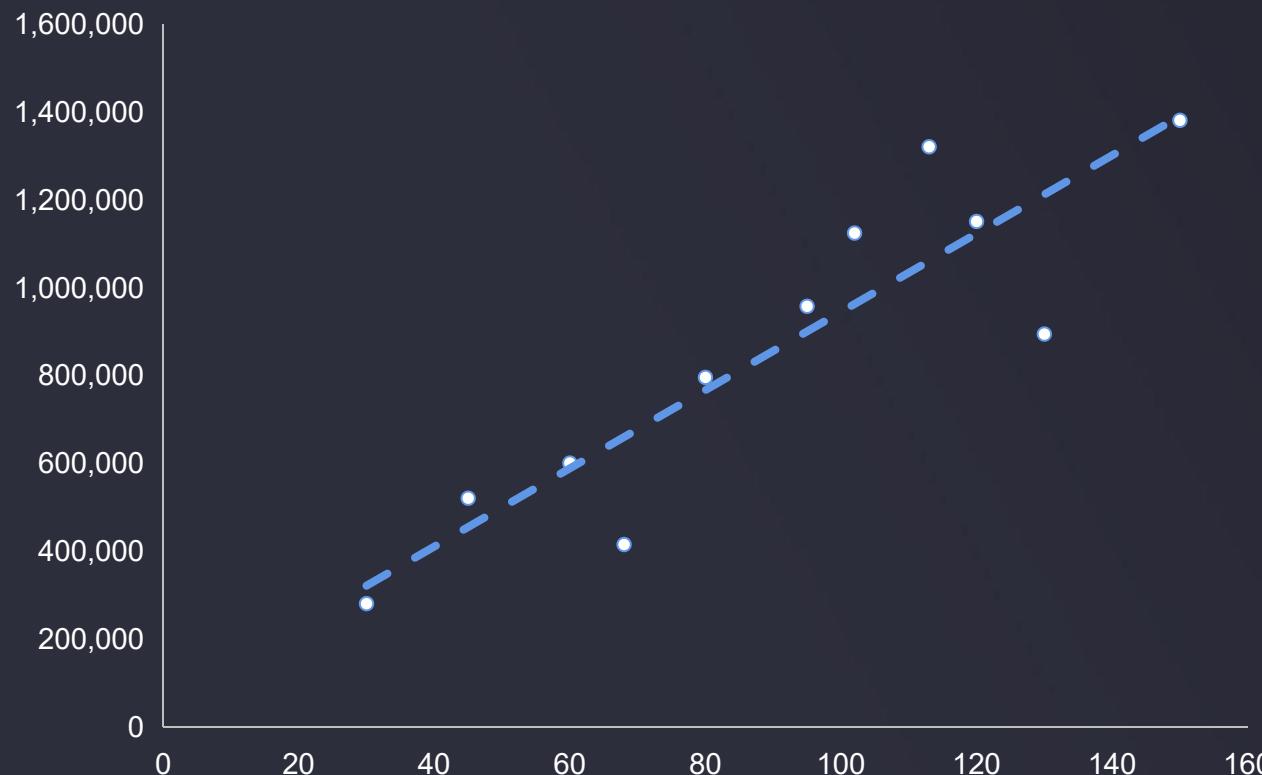
希望 y_i 与 y'_i 尽可能接近

$$\text{minimize} \left\{ \sum_{i=1}^m (y'_i - y_i)^2 \right\}$$

m 为样本数

线性回归

假设有m套房子每套房子面积 x_i 对应的实际售价为 y_i
线性模型预测的售价为 y'_i

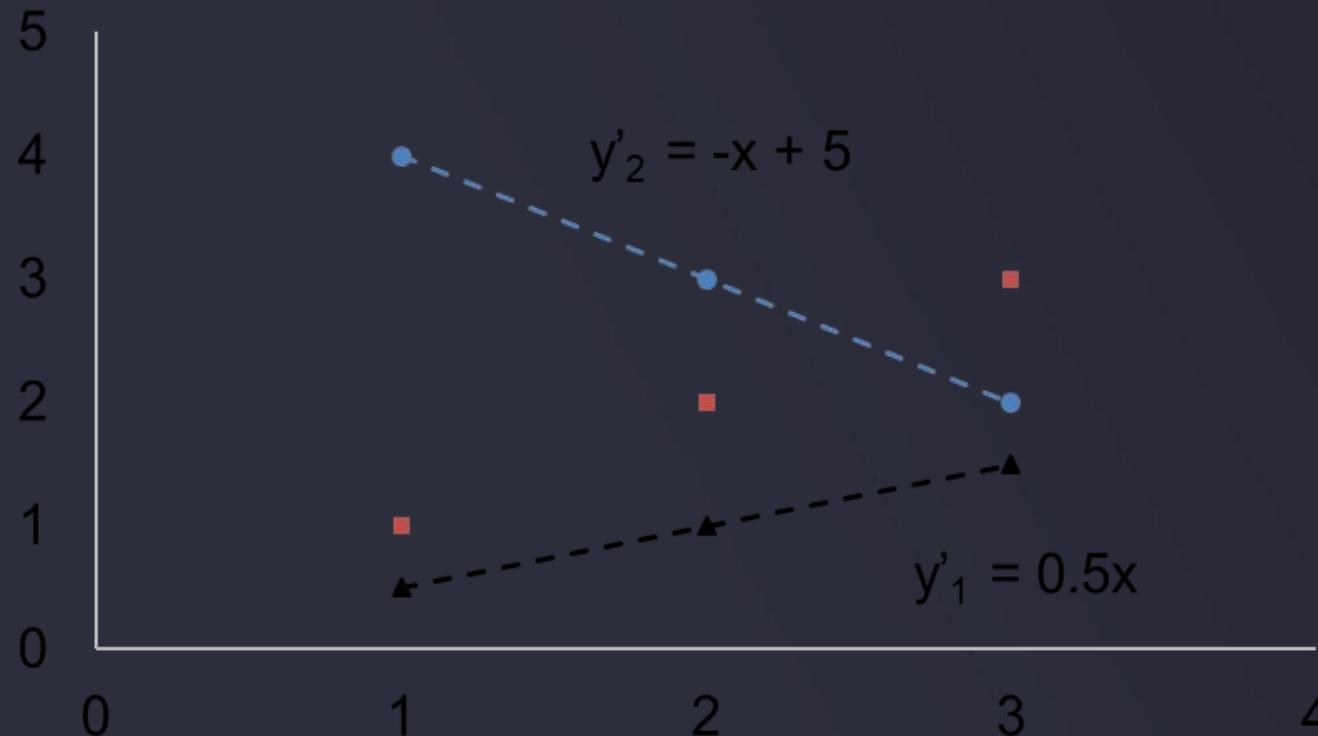


希望 y_i 与 y'_i 尽可能接近

$$\text{minimize} \left\{ \frac{1}{2m} \sum_{i=1}^m (y'_i - y_i)^2 \right\}$$

损失函数 J

回归问题求解



x	y	y'_1	y'_2
1	1	0.5	4
2	2	1	3
3	3	1.5	2

$$J_1 = \frac{1}{2m} \sum_{i=1}^m (y'_1 - y)^2 = \frac{1}{2 \times 3} \times ((0.5 - 1)^2 + (1 - 2)^2 + (1.5 - 3)^2) = 0.583$$

$$J_2 = \frac{1}{2m} \sum_{i=1}^m (y'_2 - y)^2 = \frac{1}{2 \times 3} \times ((4 - 1)^2 + (3 - 2)^2 + (2 - 3)^2) = 1.83$$



Python3人工智能入门+实战提升：机器学习

Chapter 2 回归分析与线性回归

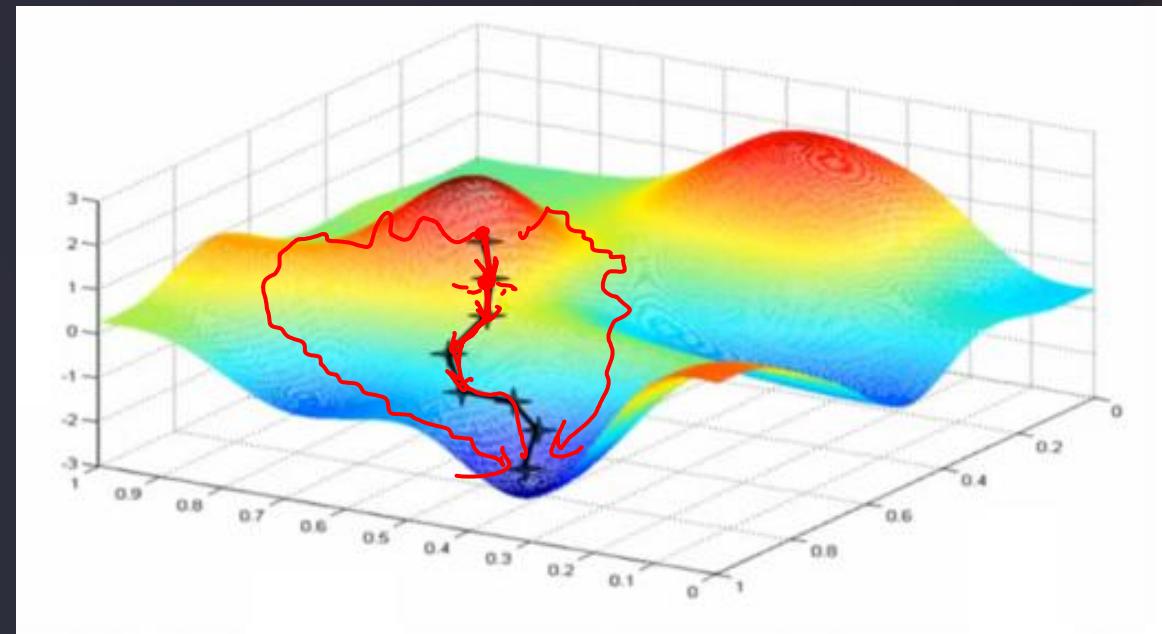
赵辛

Chapter 2 回归分析与线性回归

-
- 1 --机器学习
 - 2 --回归分析
 - 3 --模型求解与线性回归
 - 4 --梯度下降法
 - 5 --实战准备
 - 6 --实战（一）基于面积的单因子房价预测
 - 7 --实战（二）现实多因子房价预测

| 现实问题思考

从山上某点出发，找到最快的下山路径



梯度下降法

寻找函数极小值的一种方法。

核心：计算开始点 x_i 对应梯度，以一定步长向梯度反方向到达新的点 x_{i+1} ，重复此过程，直到 x_i 、 x_{i+1} 几乎不再变化。

$$y = f(x) \longrightarrow$$

搜索方法

梯度下降法

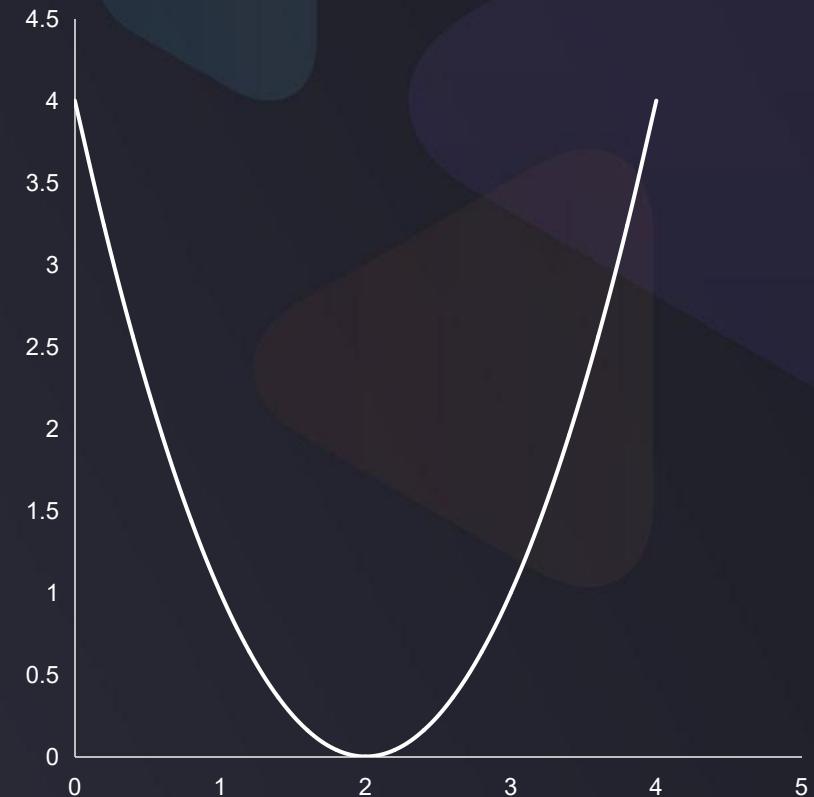
梯度下降法举例

$$y = f(x) = x^2 - 4x + 4$$

$$x_i = 1, \alpha = 0.01$$

$$x_{i+1} = ??$$

逐渐接近极小值点 ($x=2$)



梯度下降法

minimize(J)

$$J = \frac{1}{2m} \sum_{i=1}^m (y' - y)^2 = \frac{1}{2m} \sum_{i=1}^m (ax + b - y)^2 = g(a, b)$$

对每个系数分别使用梯度下降法，重复计算直到收敛

$$\left\{ \begin{array}{l} temp_a = a - \alpha \frac{\partial}{\partial a} g(a, b) \\ temp_b = b - \alpha \frac{\partial}{\partial b} g(a, b) \\ a = temp_a \\ b = temp_b \end{array} \right\}$$

梯度下降法

$\boxed{\text{minimize}(J)}$

$$J = \frac{1}{2m} \sum_{i=1}^m (y' - y)^2 = \frac{1}{2m} \sum_{i=1}^m (ax + b - y)^2 = g(a, b)$$

重复计算直到收敛

$$\text{temp}_a = a - \alpha \frac{\partial}{\partial a} g(a, b)$$

梯度下降法

minimize(J)

$$J = \frac{1}{2m} \sum_{i=1}^m (y' - y)^2 = \frac{1}{2m} \sum_{i=1}^m (ax + b - y)^2 = g(a, b)$$

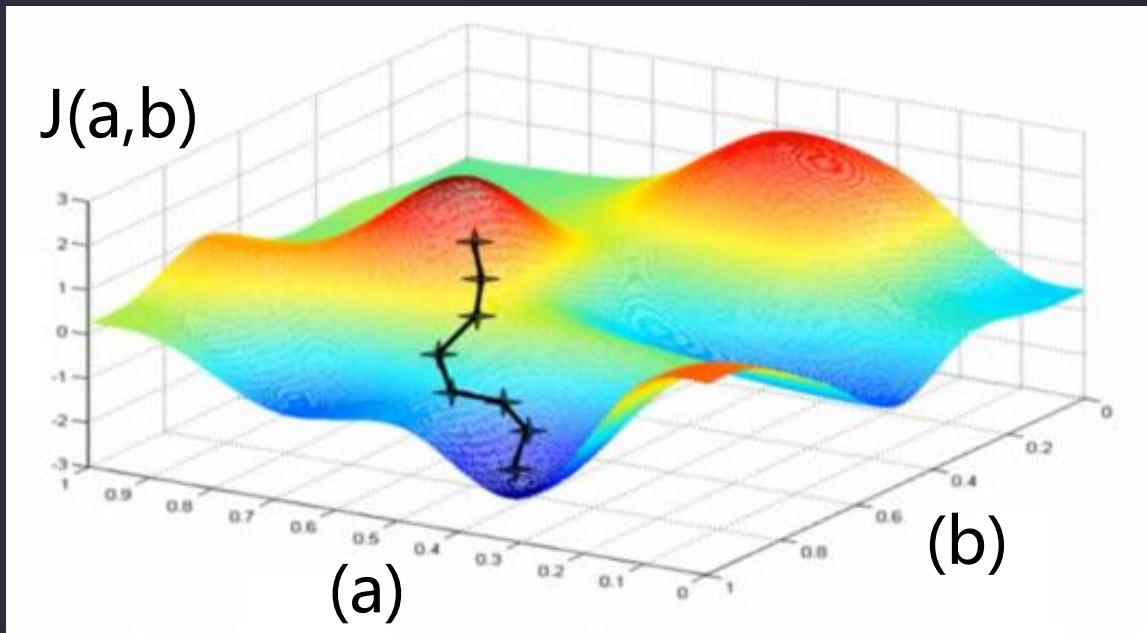
重复计算直到收敛

$$\left\{ \begin{array}{l} temp_a = a - \alpha \frac{\partial}{\partial a} g(a, b) = a - \alpha \frac{1}{m} \sum_{i=1}^m (ax_i + b - y_i) \\ temp_b = b - \alpha \frac{\partial}{\partial b} g(a, b) = b - \alpha \frac{1}{m} \sum_{i=1}^m (ax_i + b - y_i) \\ a = temp_a \\ b = temp_b \end{array} \right\}$$

梯度下降法

$\boxed{\text{minimize}(J)}$

$$J = \frac{1}{2m} \sum_{i=1}^m (y' - y)^2 = \frac{1}{2m} \sum_{i=1}^m (ax + b - y)^2 = g(a, b)$$

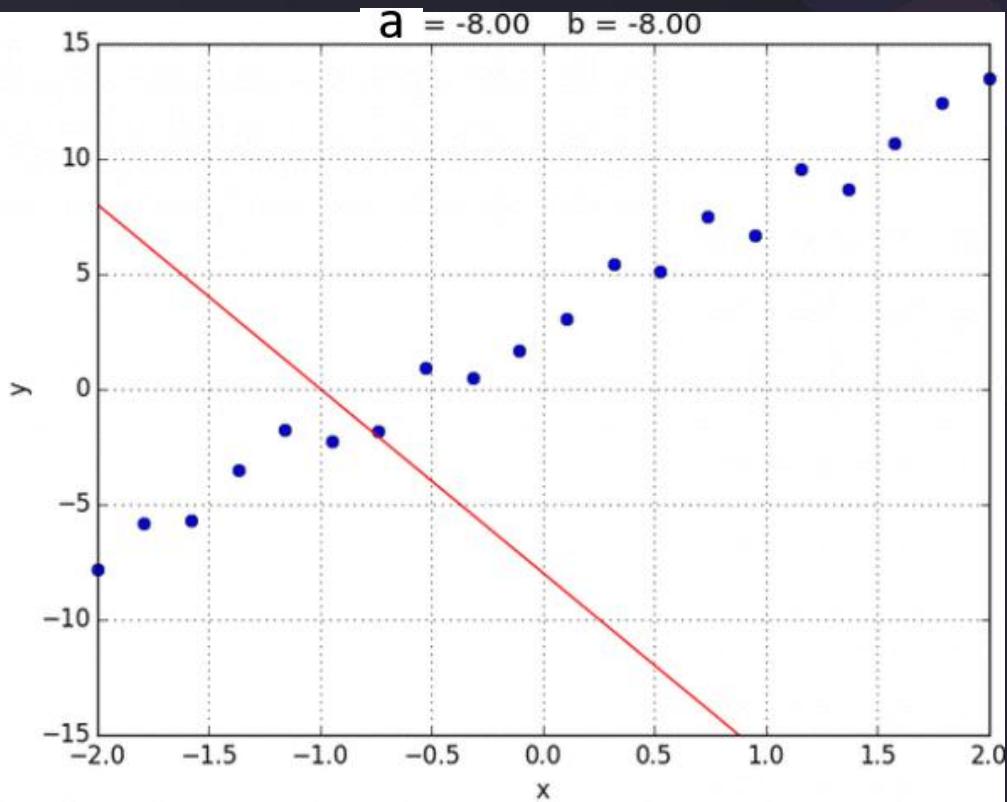
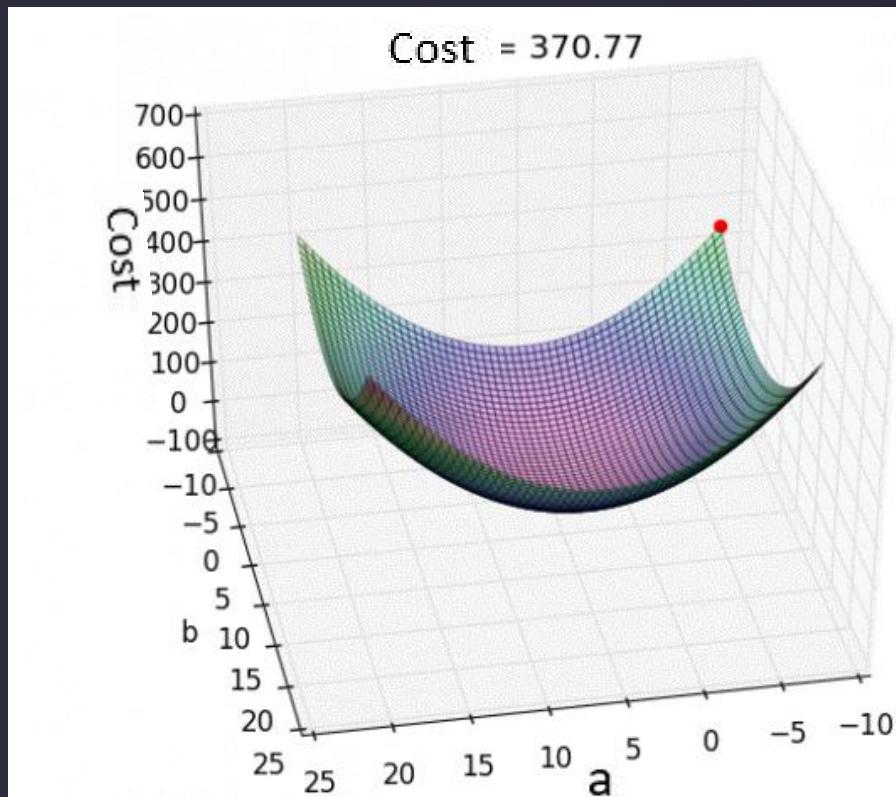


通过梯度下降法，从山丘某点出发，
可以找到下降到山底最快的线路

梯度下降法

$$Cost = J = J(a, b)$$

$$y = ax + b$$



| 梯度下降法

问题：以面积预测房价为例，写出核心的四个步骤（提示关键词：模型、损失函数、梯度下降法、预测）。



Python3人工智能入门+实战提升：机器学习

Chapter 2 回归分析与线性回归

赵辛

Chapter 2 回归分析与线性回归

-
- 1 --机器学习
 - 2 --回归分析
 - 3 --模型求解与线性回归
 - 4 --梯度下降法
 - 5 --实战准备
 - 6 --实战（一）基于面积的单因子房价预测
 - 7 --实战（二）现实多因子房价预测

|Scikit-learn

针对机器学习应用而开发的算法库

编程语言：python

常用功能： 数据预处理、分类、回归、降维、模型选择等常用的机器学习算法

三大优点：

- 丰富的算法模块；
- 易于安装与使用；
- 样例丰富、教程文档详细

官网：<https://scikit-learn.org/stable/index.html>



| 任务一：基于面积的单因子房价预测

基于课程中的房价预测案例与task1_data.csv数据，建立单因子线性回归模型，预测面积100平方米的房子售价100万是否值得投资。

面积	房价
68	414592
95	956877
102	1123582
130	893667
60	600000
45	520000
30	280000
80	795000
120	1150000
113	1320000
150	1380234

- 1、完成数据加载与可视化
 - 2、进行数据预处理: X、y赋值、格式转化、维度确认
 - 3、建立单因子线性回归模型，训练模型
 - 4、评估模型表现，可视化线性回归预测结果
- 拓展任务：手动计算模型输出a、b后对应的损失函数，尝试其他ab组合，对比损失函数大小

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#数据加载  
import pandas as pd  
import numpy as np  
data = pd.read_csv('task1_data.csv')  
data.head()
```

	面积	房价
0	68	414592
1	95	956877
2	102	1123582
3	130	893667
4	60	600000

线性回归实战流程

数据加载及展示

数据预处理

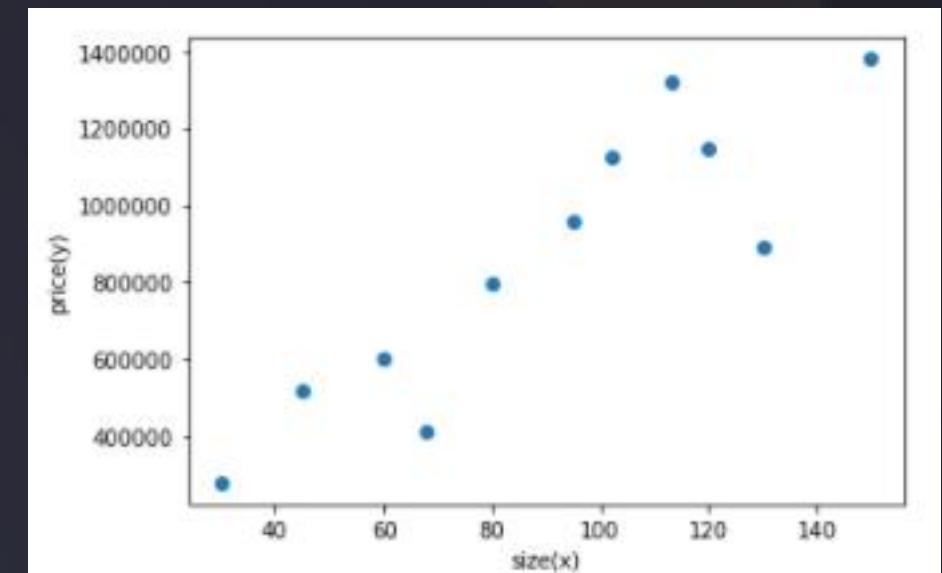
模型建立及训练

模型预测

结果展示及表现评估

#数据可视化

```
from matplotlib import pyplot as plt  
fig1 = plt.figure()  
plt.scatter(x,y)  
plt.xlabel('size(x)')  
plt.ylabel('price(y)')  
plt.show()
```



线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#x y赋值  
x = data.loc[:,‘面积’]  
y = data.loc[:,‘房价’]  
print(x,y)
```

```
#数据格式转化  
x = np.array(x)  
y = np.array(y)  
print(x.shape,y.shape)
```

```
x = x.reshape(-1,1)  
y = y.reshape(-1,1)  
print(x.shape,y.shape)
```

```
<class ‘numpy.ndarray’>  
<class ‘numpy.ndarray’>  
(11, 1) (11, 1)
```

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#创建模型实例  
from sklearn.linear_model import  
LinearRegression  
model = LinearRegression()
```

```
#模型训练  
model.fit(x,y)
```

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#获取线性回归模型系数  
a = model.coef_  
b = model.intercept_  
print(a,b,"y=f(x)={}*x+{}".format(a[0][0],b[0]))
```

```
[[8905.69177214]] [53690.91547905]  
y=f(x)=8905.691772135347*x+53690.91547905456
```

```
#结果预测  
y_predict = a[0][0]*x+ b[0]  
print(y_predict)
```

```
[[ 659277.95598426]  
[ 899731.63383191]  
[ 962071.47623686]  
[1211430.84585665]  
[ 588032.42180718]  
[ 454447.04522515]  
[ 229961.559512141]
```

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#获取线性回归模型系数  
a = model.coef_  
b = model.intercept_  
print(a,b,"y=f(x)={}*x+{}".format(a[0][0],b[0]))
```

```
#结果预测  
y_predict = a[0][0]*x+ b[0]  
print(y_predict)
```

```
#第二种预测的方法  
y_predict2 = model.predict(x)  
print(y_predict2)
```

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#预测面积为100时，对应的价格  
X_test = np.array([[100]])  
y_test_p = model.predict(X_test)  
print(y_test_p)
```

```
[[944260.09269259]]
```

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#模型评估

```
from sklearn.metrics import  
mean_squared_error,r2_score  
MSE = mean_squared_error(y,y_predict)  
R2 = r2_score(y,y_predict)  
print(MSE,R2)
```

y 与 y' 的均方误差 (MSE):

$$MSE = \frac{1}{m} \sum_{i=1}^m (y'_i - y_i)^2$$

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#模型评估

```
from sklearn.metrics import  
mean_squared_error,r2_score  
MSE = mean_squared_error(y,y_predict)  
R2 = r2_score(y,y_predict)  
print(MSE,R2)
```

R²分数(R²):

$$R^2 = 1 - \frac{\sum_{i=1}^m (y'_i - y_i)^2}{\sum_{i=1}^m (y_i - \bar{y})^2} = 1 - \frac{MSE}{\text{方差}}$$

MSE越小越好， R²分数越接近1越好

线性回归实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

模型评估指标的官方补充：

Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y , disregarding the input features, would get a R^2 score of 0.0.

参考链接：https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error

线性回归实战流程

数据加载及展示

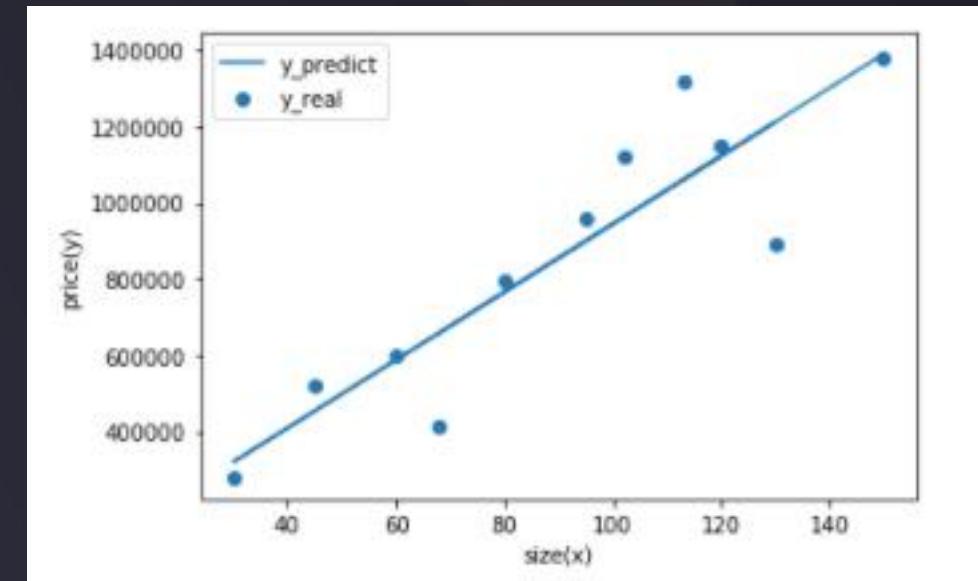
数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#预测结果可视化  
from matplotlib import pyplot as plt  
fig1 = plt.figure()  
plt.scatter(x,y,label='y_real')  
plt.plot(x,y_predict,label='y_predict')  
plt.xlabel('size(x)')  
plt.ylabel('price(y)')  
plt.legend()  
plt.show()
```



任务二：现实多因子房价预测

基于task2_data.csv数据，建立多因子线性回归模型，与只使用因子x1进行建模预测的结果进行对比。

面积	人均收入	平均房龄	价格
188.6	79245.6	4.9	1096850.0
164.2	78936.7	4.7	1455588.5
232.9	63237.0	4.9	1051696.1
150.6	65122.3	3.6	1373963.5
153.9	63628.6	5.9	623122.2
165.5	78251.3	6.3	962417.2
181.8	63814.9	5.3	1540738.9
172.1	75195.1	3.7	1442916.5
169.2	63762.7	6.2	857747.1

- 1、以面积为输入变量，建立单因子模型，评估模型表现，可视化线性回归预测结果
 - 2、以面积、人均收入、平均房龄为输入变量，建立多因子模型，评估模型表现
 - 3、预测面积=160, 人均收入=70000, 平均房龄=5的合理房价
- 拓展任务：尝试以人均收入、平均房龄作为单因子的模型，思考因子与价格的关系

| 拓展延伸：为毕业与工作做准备

如果想以房价预测作为一个综合项目，完成毕业设计，应该如何做计划，过程中可能遇到哪些问题，我们需要做什么工作？

数据获取

模型选择

结果优化

综合报告

区域、考虑指标（参数属性）；如何去获取到这些数据；
获取到的数据是否有代表性

数据中有异常数据怎么办？有数据缺失如何处理？

选用什么模型？模型表现并不好怎么办。 . .



Python3人工智能入门+实战提升：机器学习

Chapter 2 回归分析与线性回归

赵辛

Chapter 2 回归分析与线性回归

-
- 1 --机器学习
 - 2 --回归分析
 - 3 --模型求解与线性回归
 - 4 --梯度下降法
 - 5 --实战（一）基于面积的单因子房价预测
 - 6 --实战（二）现实多因子房价预测
 - 7

| 任务一：基于面积的单因子房价预测

基于课程中的房价预测案例与task1_data.csv数据，建立单因子线性回归模型，预测面积100平方米的房子售价100万是否值得投资。

面积	房价
68	414592
95	956877
102	1123582
130	893667
60	600000
45	520000
30	280000
80	795000
120	1150000
113	1320000
150	1380234

- 1、完成数据加载与可视化
 - 2、进行数据预处理: X、y赋值、格式转化、维度确认
 - 3、建立单因子线性回归模型，训练模型
 - 4、评估模型表现，可视化线性回归预测结果
- 拓展任务：手动计算模型输出a、b后对应的损失函数，尝试其他ab组合，对比损失函数大小



Python3人工智能入门+实战提升：机器学习

Chapter 2 回归分析与线性回归

赵辛

Chapter 2 回归分析与线性回归

-
- 1 --机器学习
 - 2 --回归分析
 - 3 --模型求解与线性回归
 - 4 --梯度下降法
 - 5 --实战（一）基于面积的单因子房价预测
 - 6 --实战（二）现实多因子房价预测
 - 7

任务二：现实多因子房价预测

基于task2_data.csv数据，建立多因子线性回归模型，与只使用面积单因子进行建模预测的结果进行对比。

面积	人均收入	平均房龄	价格
188.6	79245.6	4.9	1096850.0
164.2	78936.7	4.7	1455588.5
232.9	63237.0	4.9	1051696.1
150.6	65122.3	3.6	1373963.5
153.9	63628.6	5.9	623122.2
165.5	78251.3	6.3	962417.2
181.8	63814.9	5.3	1540738.9
172.1	75195.1	3.7	1442916.5
169.2	63762.7	6.2	857747.1

- 1、以面积为输入变量，建立单因子模型，评估模型表现，可视化线性回归预测结果
 - 2、以面积、人均收入、平均房龄为输入变量，建立多因子模型，评估模型表现
 - 3、预测面积=160, 人均收入=70000, 平均房龄=5的合理房价
- 拓展任务：尝试以人均收入、平均房龄作为单因子的模型，思考因子与价格的关系

| 拓展延伸：为毕业与工作做准备

如果想以房价预测作为一个综合项目，完成毕业设计，应该如何做计划，过程中可能遇到哪些问题，我们需要做什么工作？

数据获取

模型选择

结果优化

综合报告

区域、考虑指标（参数属性）；如何去获取到这些数据；
获取到的数据是否有代表性

数据中有异常数据怎么办？有数据缺失如何处理？

选用什么模型？模型表现并不好怎么办。 . .



Python3人工智能入门+实战提升：机器学习

Chapter 3 分类任务与逻辑回归

赵辛

Chapter 3 分类任务与逻辑回归

-
- 1 --分类任务
 - 2 --分类预测的实现
 - 3 --逻辑回归
 - 4 --实战准备
 - 5 --实战（一）逻辑回归实现数据二分类
 - 6 --实战（二）商业异常消费数据预测

现实案例思考：垃圾短信检测



任务：
自动判断收到的信息是否为垃圾信息

现实案例思考：垃圾短信检测



如何实现:

收集一些样本，告诉计算机哪些是垃圾信息

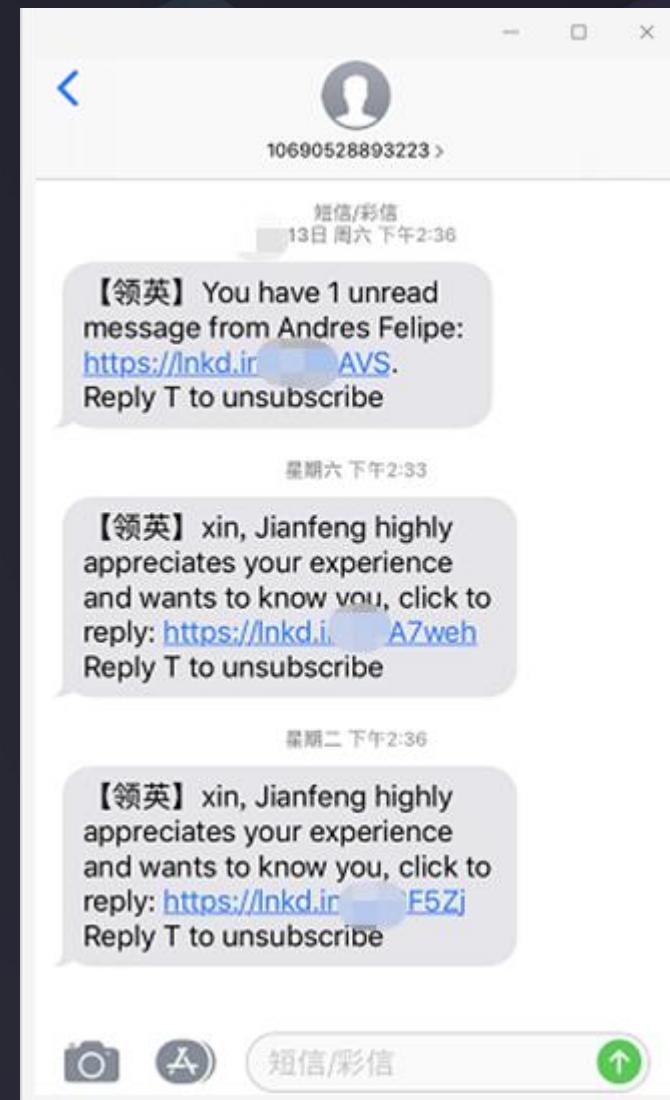
计算机自动寻找垃圾信息共同特征

在新信息中检测是否包含垃圾信息特征内容，判断其是否为垃圾邮件

部分特征：发件人、是否群发、网址、元、赢、微信、免费

现实案例思考：垃圾短信检测

部分特征：发件人、是否群发、网址、元、赢、微信、免费



现实案例思考：图像识别

图片

标签

新图片

预测



西瓜



橙子



桃子



草莓



葡萄

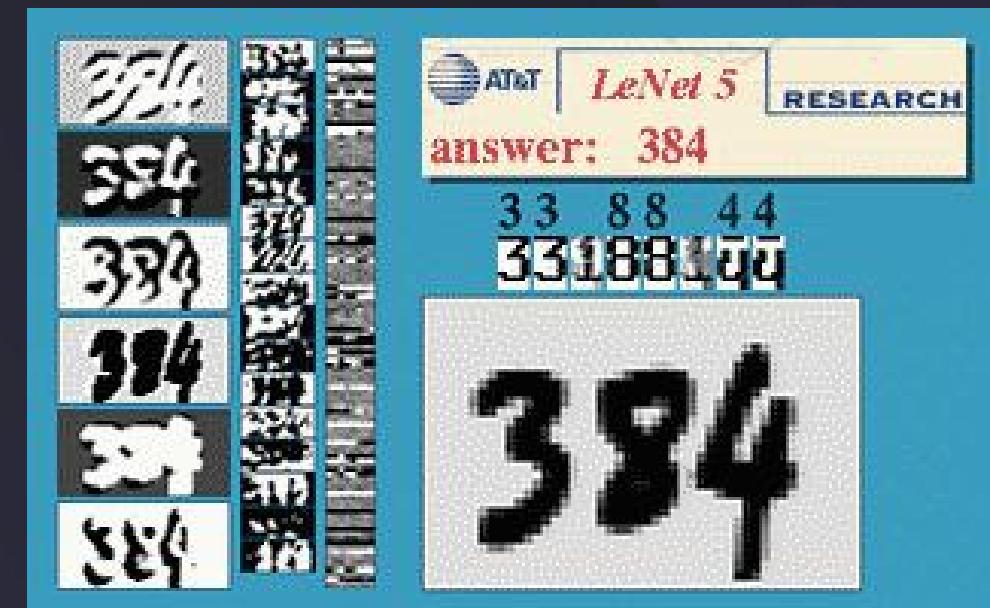
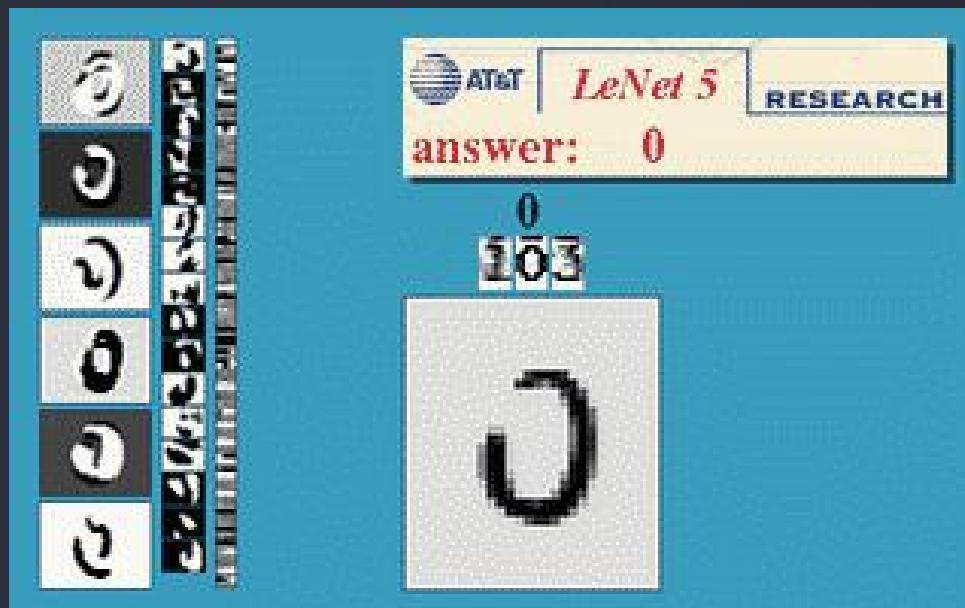


荔枝



草莓

现实案例思考：手写数字识别



参考链接：<http://yann.lecun.com/exdb/lenet/>

现实案例思考：股票涨跌预测

名称	品种类型	行业	多空	操作时间	操作类型	操作价格	数量
汇顶科技	股票	电子	-	2019-11-20 15	买入	215.70	1300
华友钴业	股票	有色	-	2019-11-20 15	买入	51.40	5700
步长制药	股票	医药	-	2019-11-20 15	买入	39.44	7400
潍柴动力	股票	汽车	-	2019-11-20 15	卖出	248.22	500
泸州老窖	股票	食品	-	2019-11-20 15	卖出	2820.76	100
长安汽车	股票	汽车	-	2019-11-20 15	买入	67.42	4300
五粮液	股票	食品	-	2019-11-20 15	卖出	2391.77	100
石基信息	股票	计算	-	2019-11-20 15	买入	696.53	400
大华股份	股票	电子	-	2019-11-20 15	卖出	759.44	200
奥飞娱乐	股票	传媒	-	2019-11-20 15	买入	70.34	4100
大北农	股票	农林	-	2019-11-20 15	卖出	42.26	2600
海康威视	股票	电子	-	2019-11-20 15	卖出	691.33	100
欧菲光	股票	电子	-	2019-11-20 15	买入	263.90	500
赣锋锂业	股票	有色	-	2019-11-20 15	买入	241.84	1200
天齐锂业	股票	有色	-	2019-11-20 15	买入	187.67	1500
三一重工	股票	机械	-	2019-11-20 15	买入	921.00	400

判断每只股票接下来
一段时间会上涨，还
是下跌

分类预测

根据数据类别与部分特征信息，自动寻找类别与特征信息的关系，
判断一个新的样本属于哪种类别

特征信息	数据类别	寻找关系
$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$	$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$	$y = f(x_1, x_2 \cdots x_n)$

如果 $y_{test} = i$, 判断为类别 I

分类预测

特征信息

$$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

数据类别

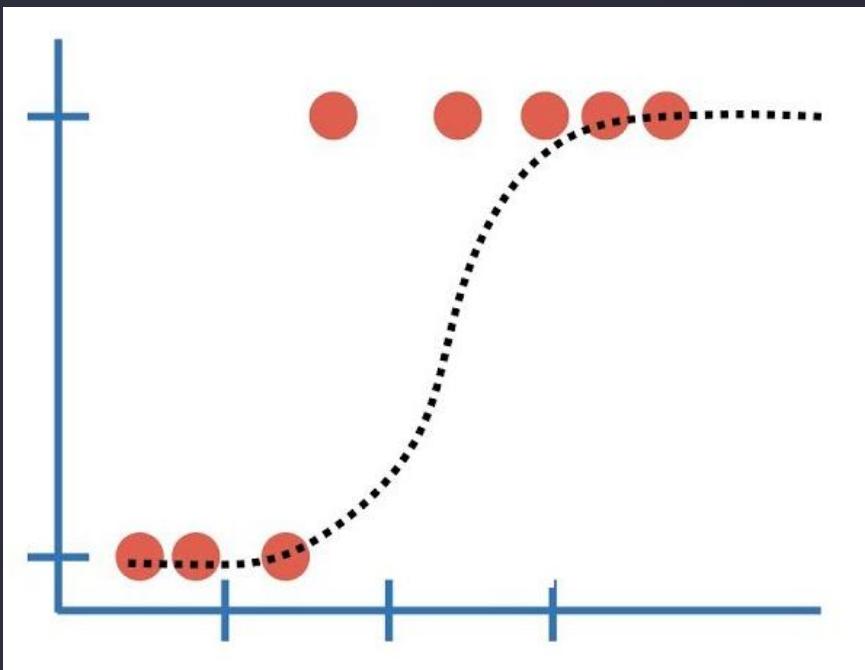
发件人 现金 赢 $y_{test} = 1$

$y_{test} = 1$, 判断为垃圾信息!



分类预测

逻辑回归



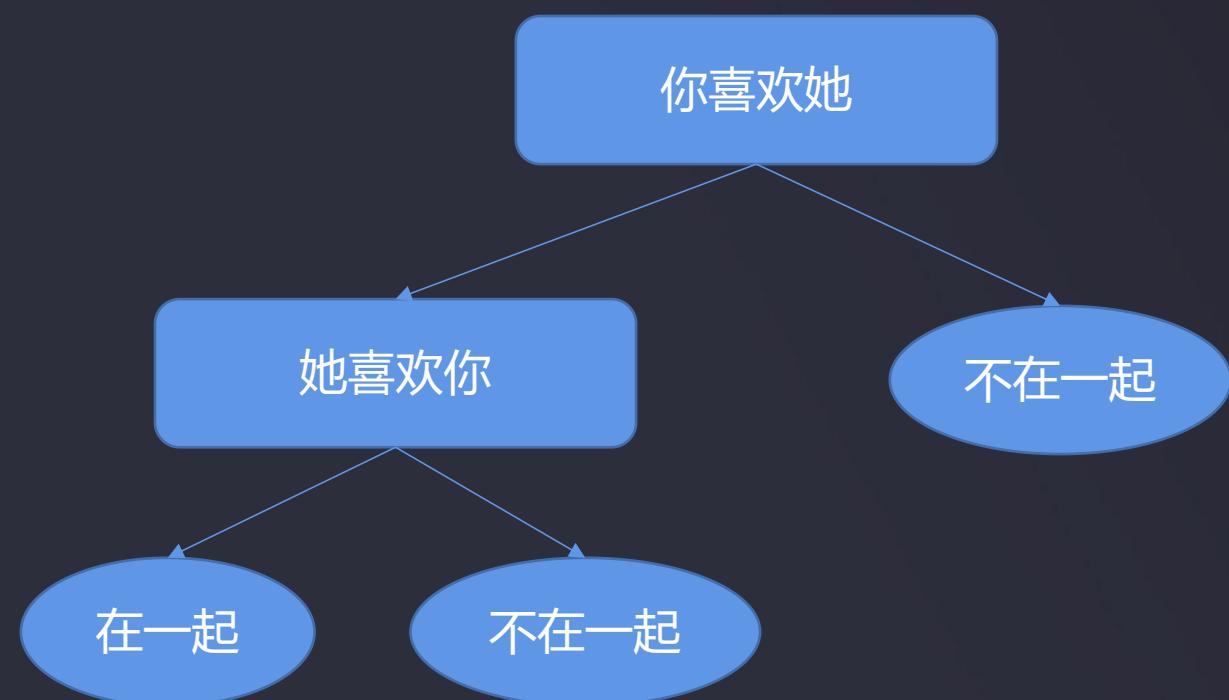
朴素贝叶斯

$$P(Y|X) = P(Y) * \frac{P(X|Y)}{P(X)}$$

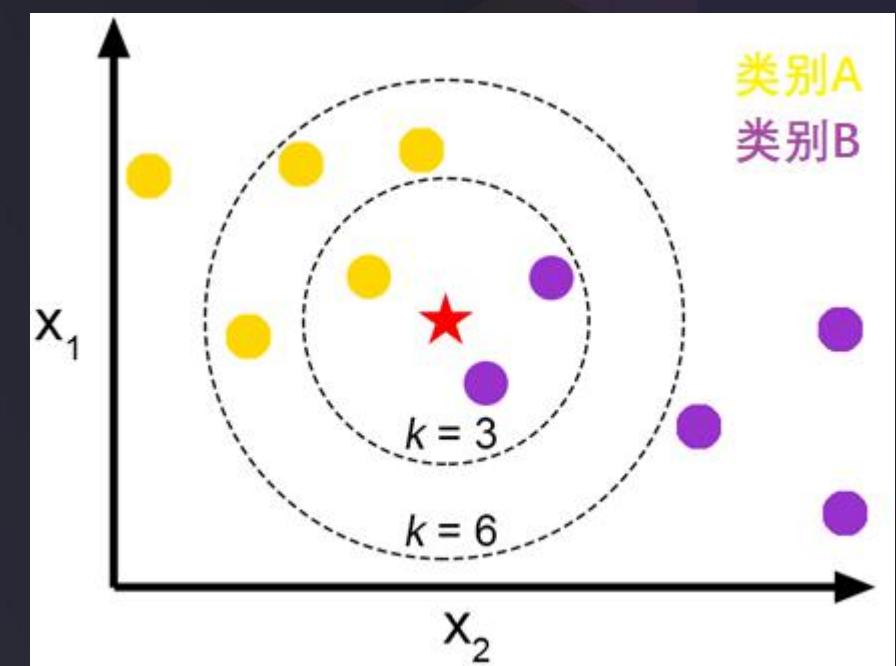
$$P(y_i|x_1, x_2 \dots, x_n) = \frac{P(y_i) \prod_{j=1}^n P(x_j|y_i)}{\prod_{j=1}^n P(x_j)}$$

分类预测

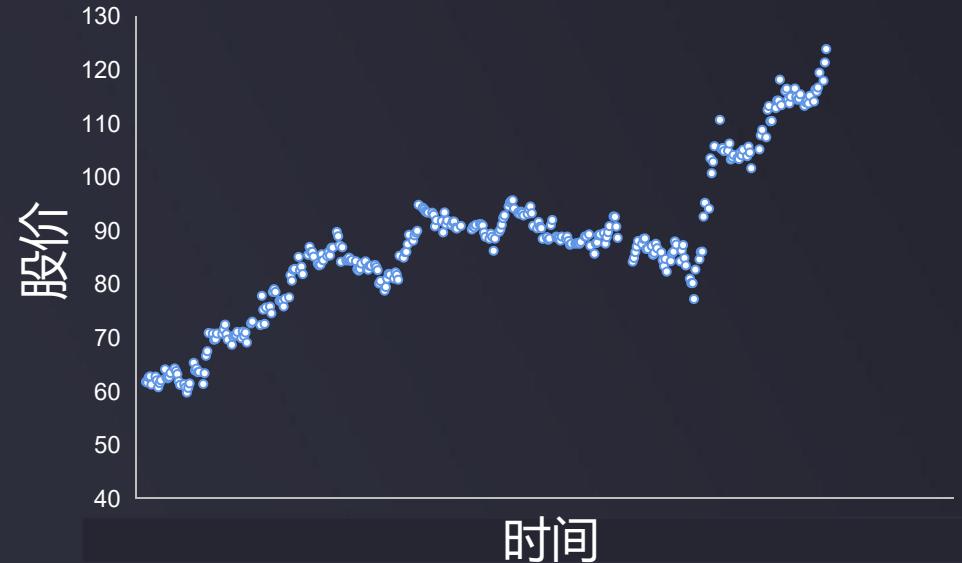
决策树



KNN近邻模型



通过股价预测任务区分回归任务与分类任务



回归：连续性数值预测
模型输出：连续型数值
(明天股价预测为：125.1)

分类：非连续性判断类别
模型输出：非连续型标签
(明天股价预测为：上涨)

| 知识巩固

问题：

- 1、检查自己的手机短信或邮箱，观察垃圾邮件发件人、标题、内容，列举垃圾邮件的共同点。

- 2、以下两种情况，属于什么任务，对比这两种任务的不同点：
A.预测中国平安明天股价
B.预测中国平安明天股价将会上涨还是下跌



Python3人工智能入门+实战提升：机器学习

Chapter 3 分类任务与逻辑回归

赵辛

Chapter 3 分类任务与逻辑回归

-
- 1 --分类任务
 - 2 --分类预测的实现
 - 3 --逻辑回归
 - 4 --实战准备
 - 5 --实战（一）逻辑回归实现数据二分类
 - 6 --实战（二）商业异常消费数据预测

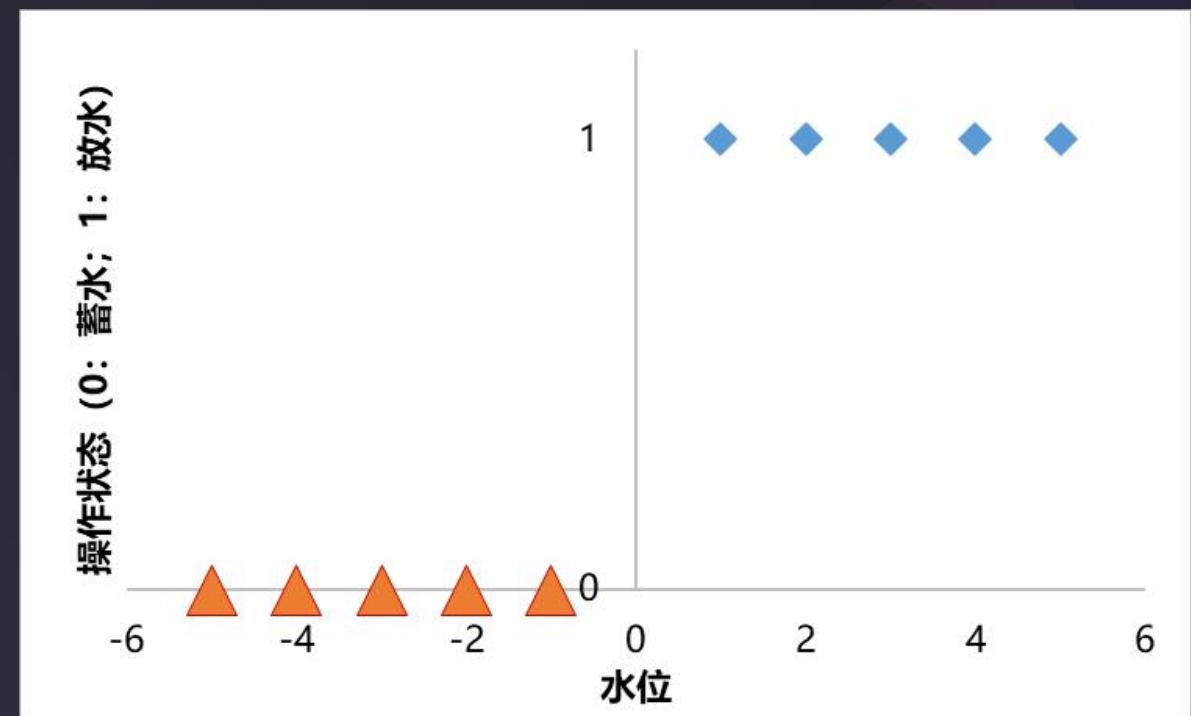
分类预测

任务：根据水位，判断水池是否需要蓄水或放水

训练数据

水位为-1、-2、-3、-4、-5：
水不足，待蓄水（负样本）

水位为1、2、3、4、5：
水过量，待放水（正样本）



分类预测

任务：根据水位，判断水池是否需要蓄水或放水

特征信息	数据类别	寻找关系
$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$	$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$	$y = f(x_1, x_2 \cdots x_n)$

如果 $y_{test} = i$, 判断为类别 I

特征信息：水位数据； 数据类别：待蓄水（0）、放水（1）

分类预测

任务：根据水位，判断水池是否需要蓄水或放水

特征信息

$$\mathbf{A} = \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{m1} \end{bmatrix}$$

数据类别

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

寻找关系

$$y = f(x_1)$$

$$y_{test} = \begin{cases} 0, & \text{待蓄水} \\ 1, & \text{待放水} \end{cases}$$

特征信息：水位数据、数据类别：待蓄水（0）、放水（1）

分类预测

任务：根据水位，判断水池是否需要蓄水或放水

训练数据

水位为-1、-2、-3、-4、-5：
水不足，待蓄水（负样本）

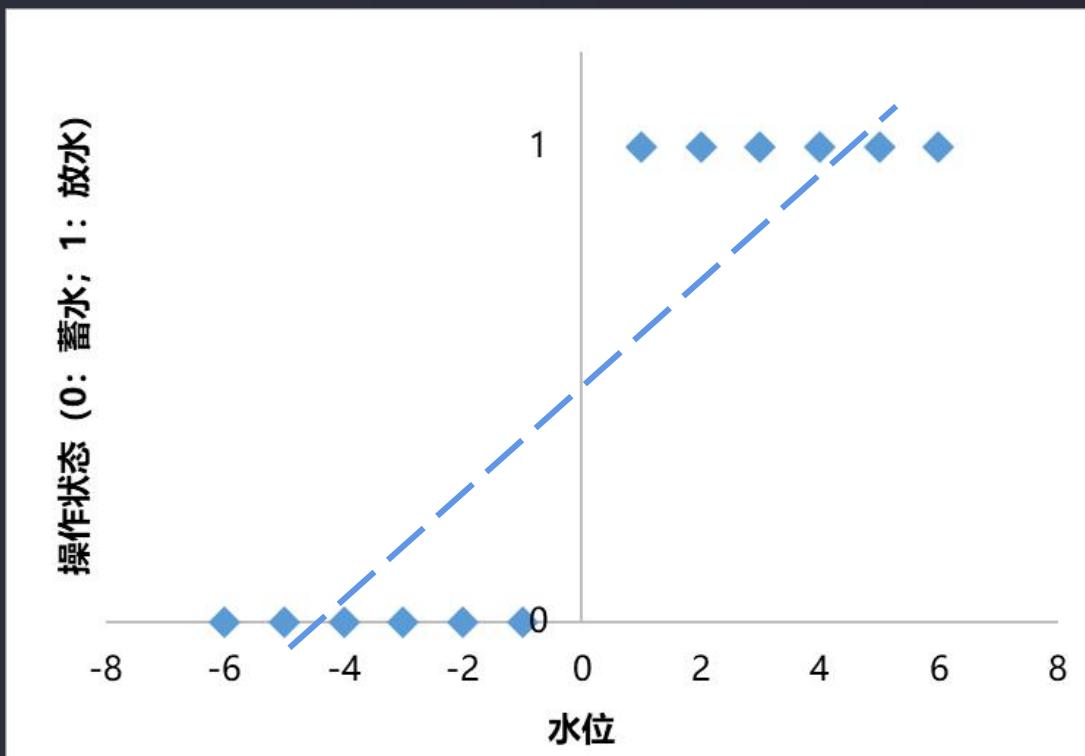
水位为1、2、3、4、5：
水过量，待放水（正样本）



x	实际y
-6	0
-5	0
-4	0
-3	0
-2	0
-1	0
1	1
2	1
3	1
4	1
5	1
6	1

分类预测

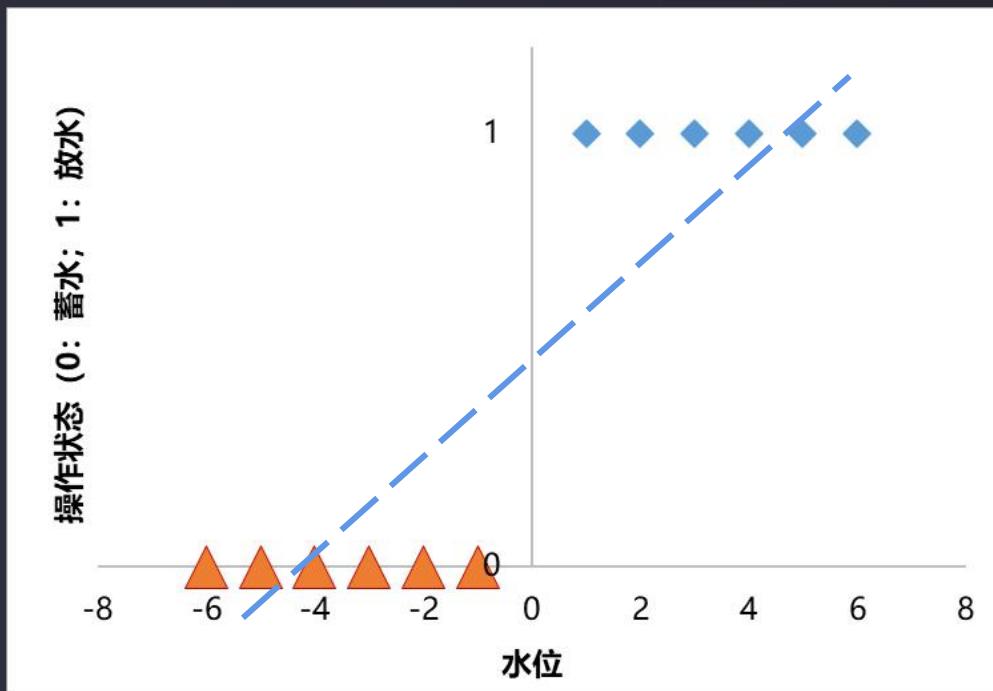
任务：根据水位，判断水池是否需要蓄水或放水



x	实际y	Y(x)
-6	0	-0.19
-5	0	-0.08
-4	0	0.04
-3	0	0.15
-2	0	0.27
-1	0	0.38
1	1	0.62
2	1	0.73
3	1	0.85
4	1	0.96
5	1	1.08
6	1	1.19

分类预测

任务：根据水位，判断水池是否需要蓄水或放水



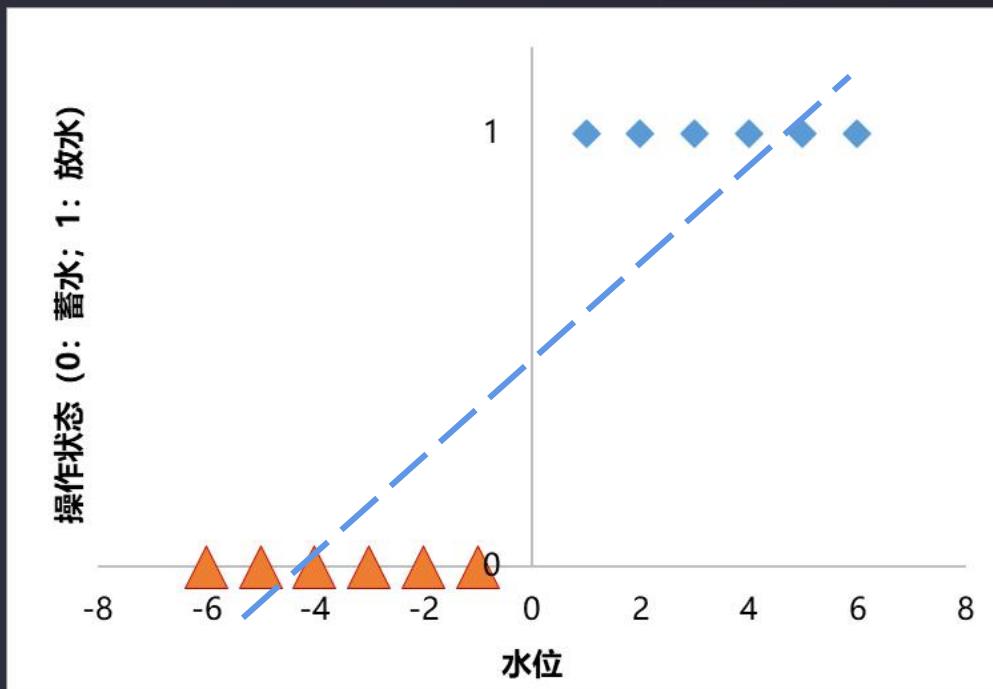
$$(1) Y = 0.1154x + 0.5$$

$$(2) y = f(x) = \begin{cases} 1, & Y \geq 0.5 \\ 0, & Y < 0.5 \end{cases}$$

分类预测

任务：根据水位，判断水池是否需要蓄水或放水

线性回归+逻辑判断效果似乎很不錯！



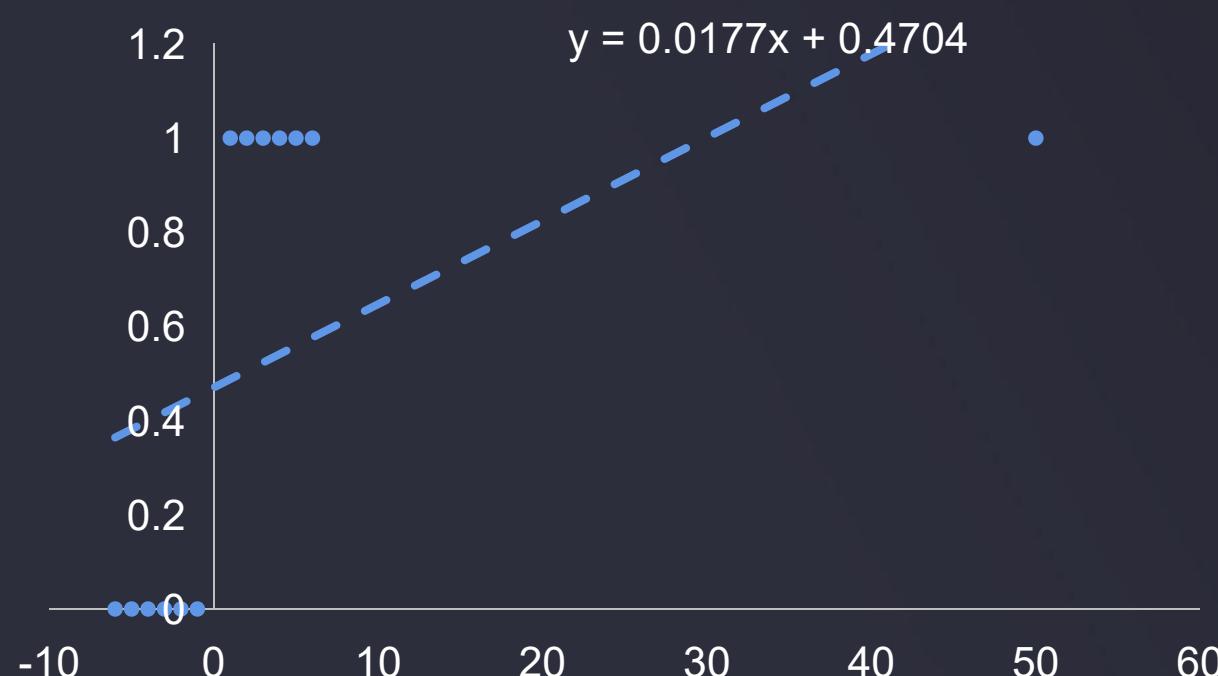
x	实际y	$\hat{Y}(x)$	预测 $y(x)$
-6	0	-0.19	0
-5	0	-0.08	0
-4	0	0.04	0
-3	0	0.15	0
-2	0	0.27	0
-1	0	0.38	0
1	1	0.62	1
2	1	0.73	1
3	1	0.85	1
4	1	0.96	1
5	1	1.08	1
6	1	1.19	1

$$(1) \hat{Y} = 0.1154x + 0.5$$

$$(2) y = f(x) = \begin{cases} 1, & \hat{Y} \geq 0.5 \\ 0, & \hat{Y} < 0.5 \end{cases}$$

分类预测

增加一个样本数据： $x=50, y=1$



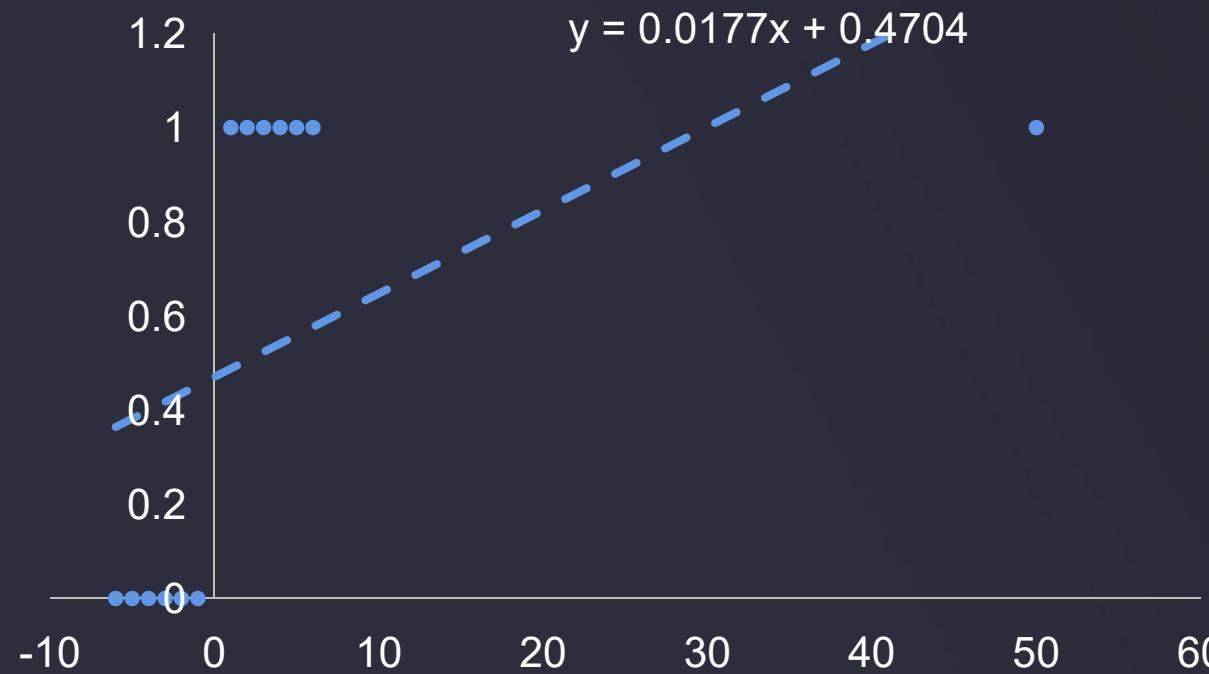
数据样本复杂度增加，模型准确率下降明显

x	实际y	$\hat{Y}(x)$	预测 $y(x)$
-6	0	0.36	0
-5	0	0.38	0
-4	0	0.40	0
-3	0	0.42	0
-2	0	0.44	0
-1	0	0.45	0
1	1	0.49	0
2	1	0.51	1
3	1	0.52	1
4	1	0.54	1
5	1	0.56	1
6	1	0.58	1
50	1	1.36	1

思考：需要更适合于分类场景的模型

知识巩固

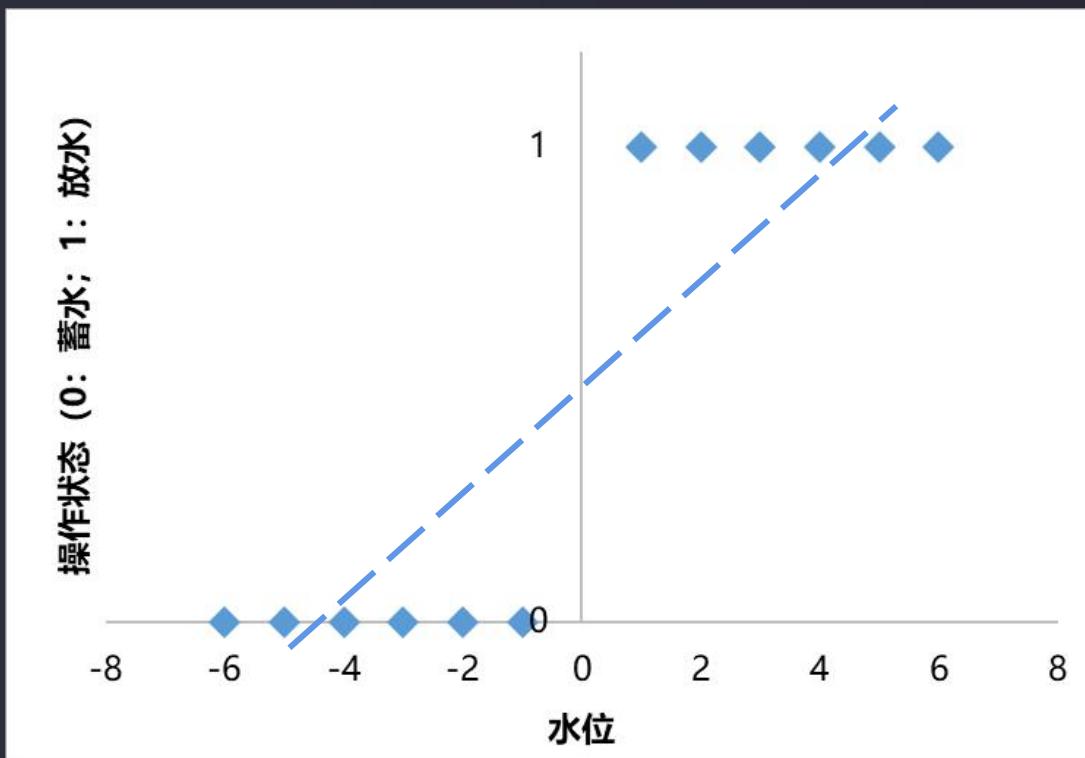
基于表格数据，结合第二章内容，建立线性回归模型，计算拟合数据分布的线性回归模型。



x	实际y
-6	0
-5	0
-4	0
-3	0
-2	0
-1	0
1	1
2	1
3	1
4	1
5	1
6	1
50	1

分类预测

任务：根据水位，判断水池是否需要蓄水或放水



x	实际y	Y(x)
-6	0	-0.19
-5	0	-0.08
-4	0	0.04
-3	0	0.15
-2	0	0.27
-1	0	0.38
1	1	0.62
2	1	0.73
3	1	0.85
4	1	0.96
5	1	1.08
6	1	1.19



Python3人工智能入门+实战提升：机器学习

Chapter 3 分类任务与逻辑回归

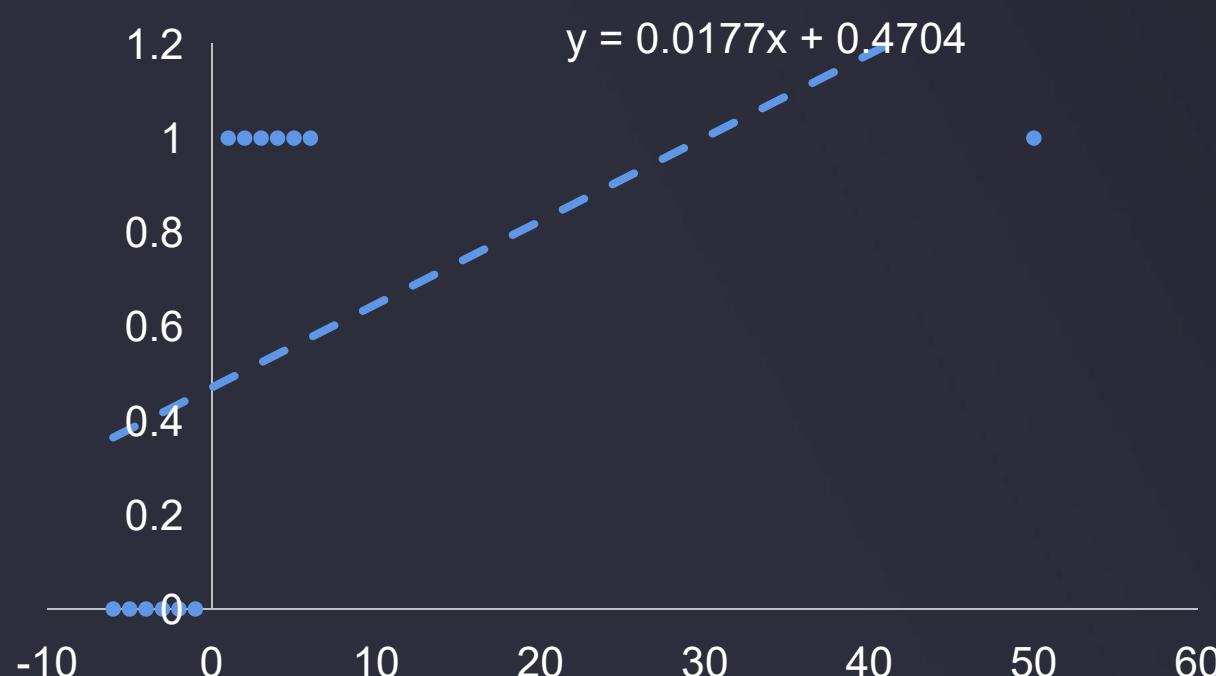
赵辛

Chapter 3 分类任务与逻辑回归

-
- 1 --分类任务
 - 2 --分类预测的实现
 - 3 --逻辑回归
 - 4 --实战准备
 - 5 --实战（一）逻辑回归实现数据二分类
 - 6 --实战（二）商业异常消费数据预测

分类预测

增加一个样本数据： $x=50, y=1$



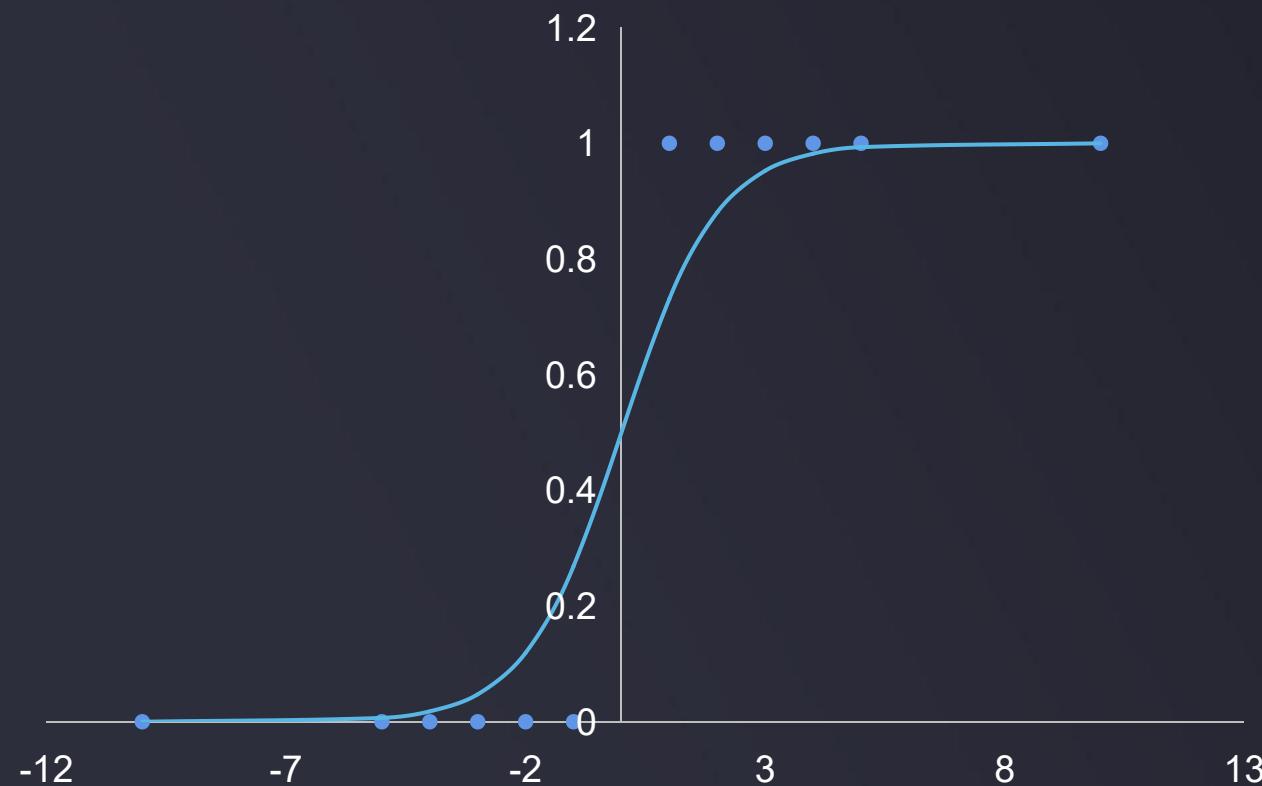
数据样本复杂度增加，模型准确率下降明显

x	实际y	$\hat{Y}(x)$	预测 $y(x)$
-6	0	0.36	0
-5	0	0.38	0
-4	0	0.40	0
-3	0	0.42	0
-2	0	0.44	0
-1	0	0.45	0
1	1	0.49	0
2	1	0.51	1
3	1	0.52	1
4	1	0.54	1
5	1	0.56	1
6	1	0.58	1
50	1	1.36	1

思考：需要更适合于分类场景的模型

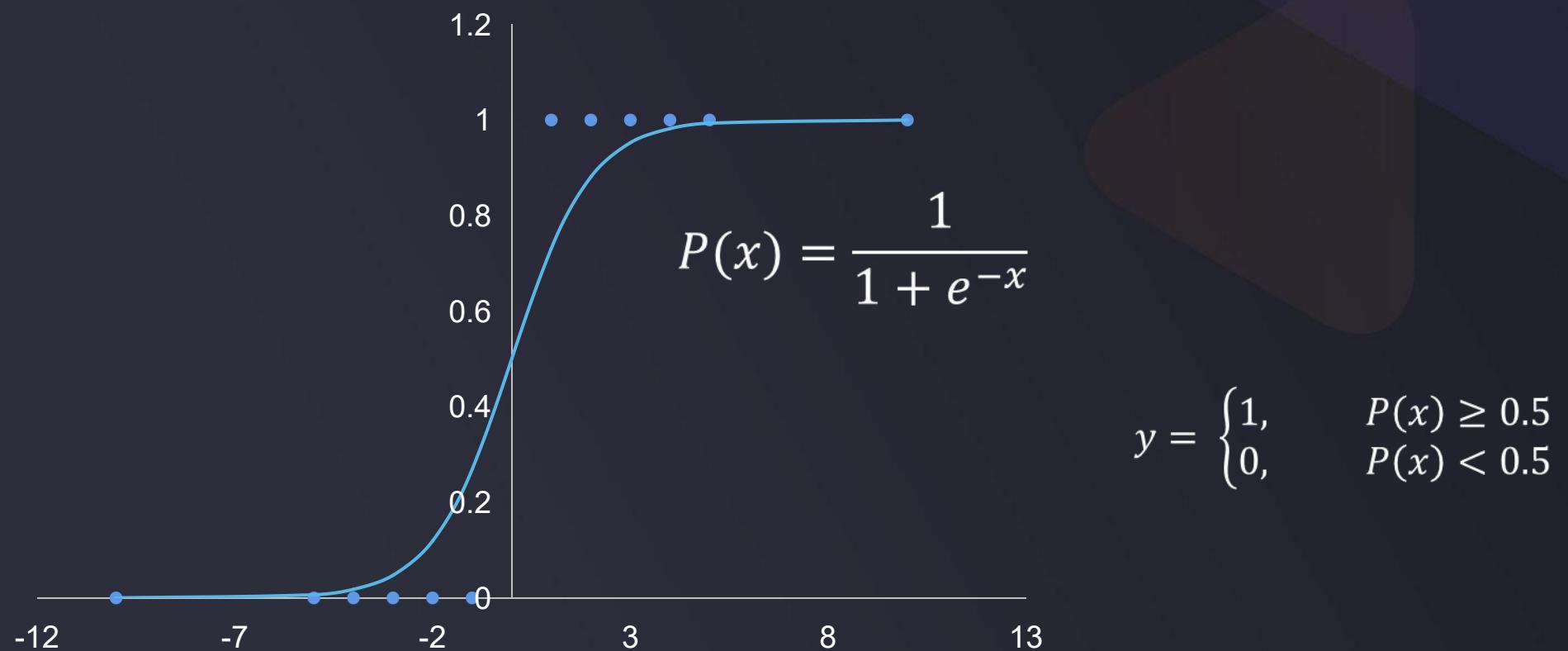
| 更适合于分类场景的模型

- 1、更符合类别数据分布特点；
- 2、连续的函数关系



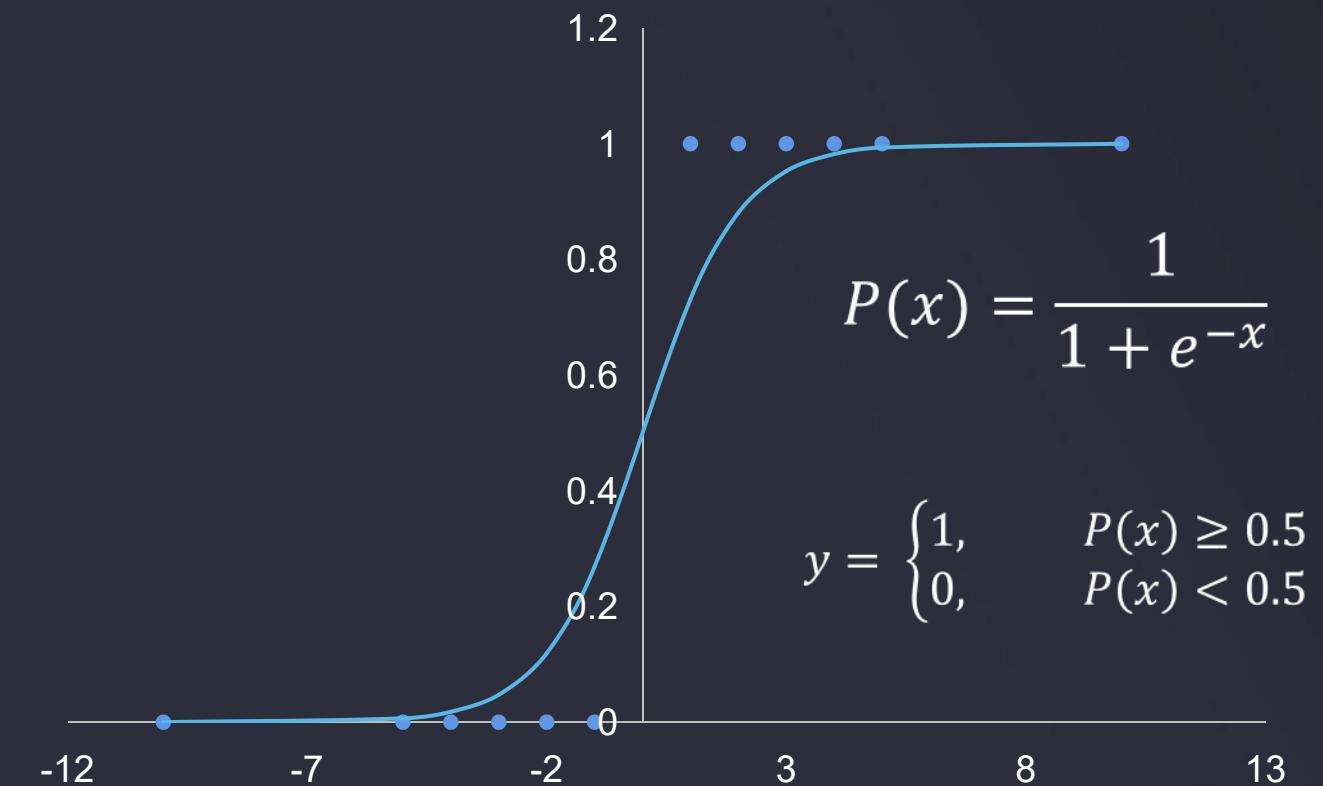
逻辑回归

根据数据特征，计算样本归属于某一类别的概率 $P(x)$ ，根据概率数值判断其所属类别。



逻辑回归

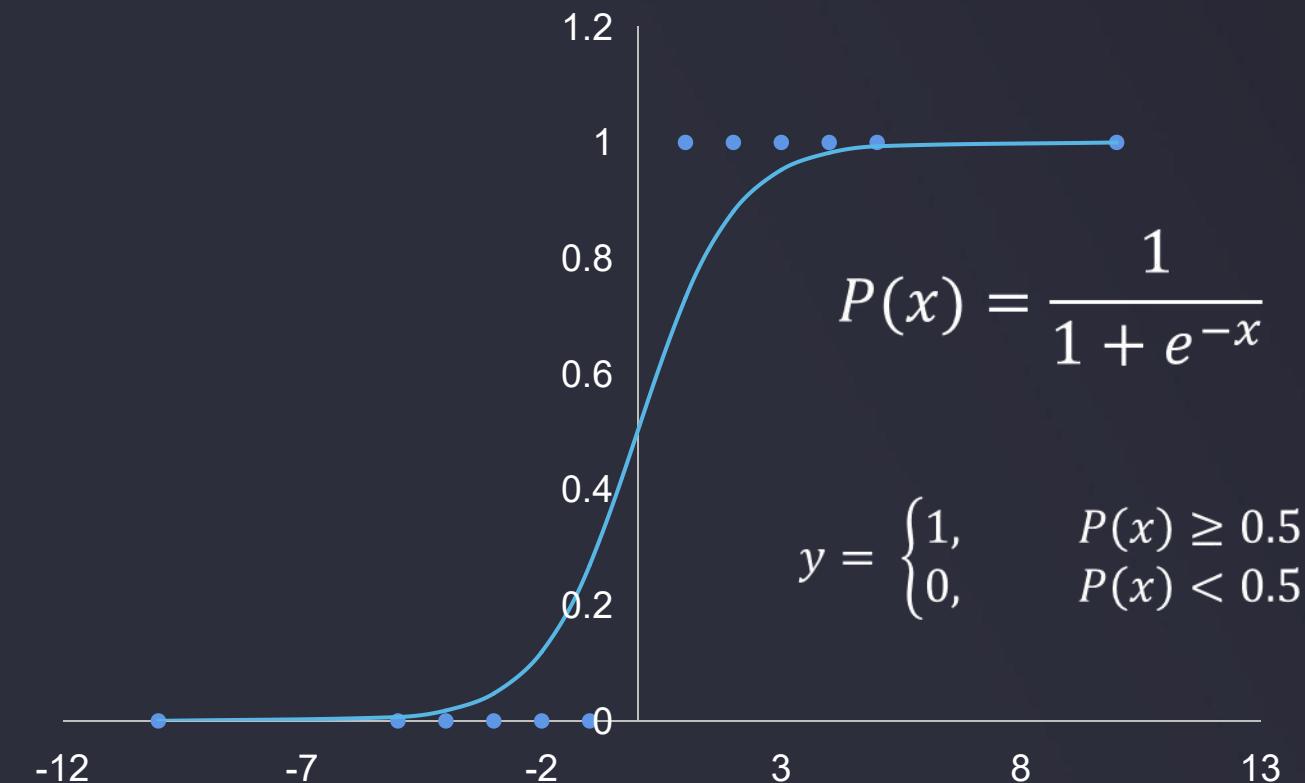
计算 $x = -20, 1, 100$ 对应的y值



逻辑回归

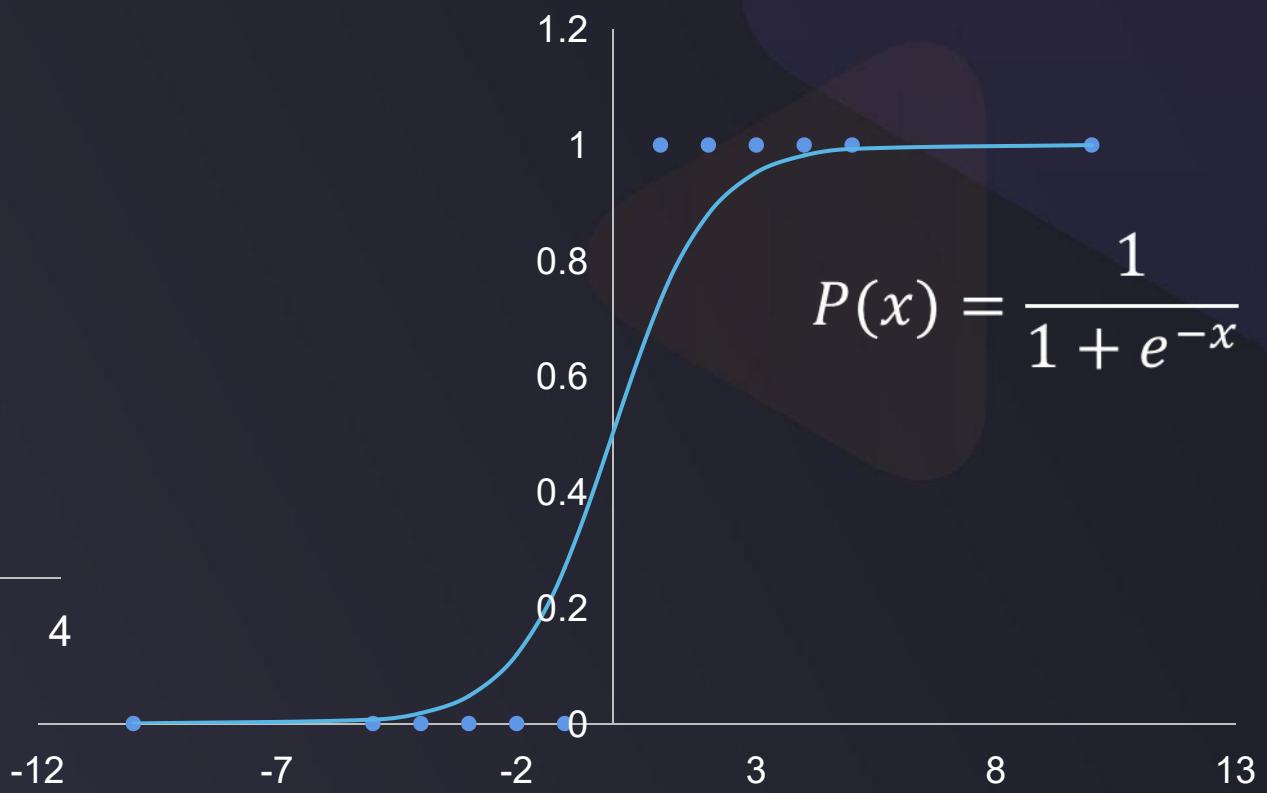
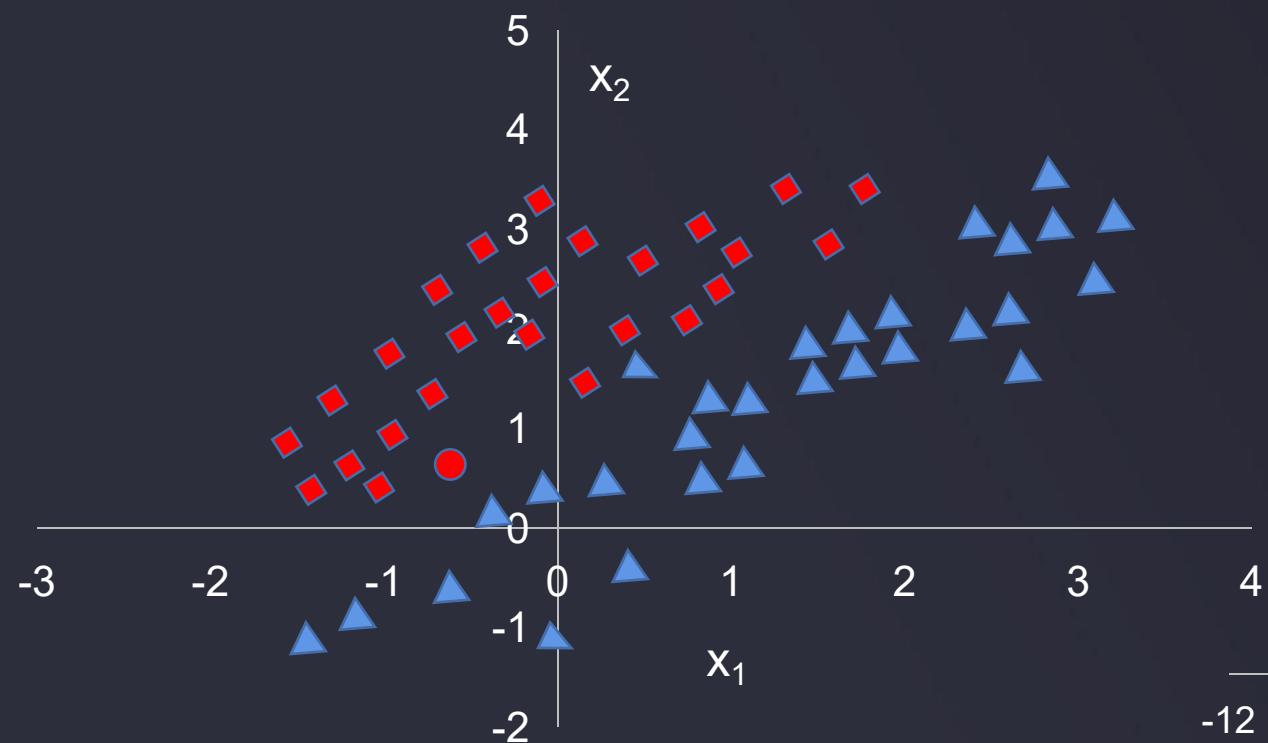
$Y(x)$ 界线明显，分类效果好！

计算 $x = -20, 1, 100$ 对应的 y 值

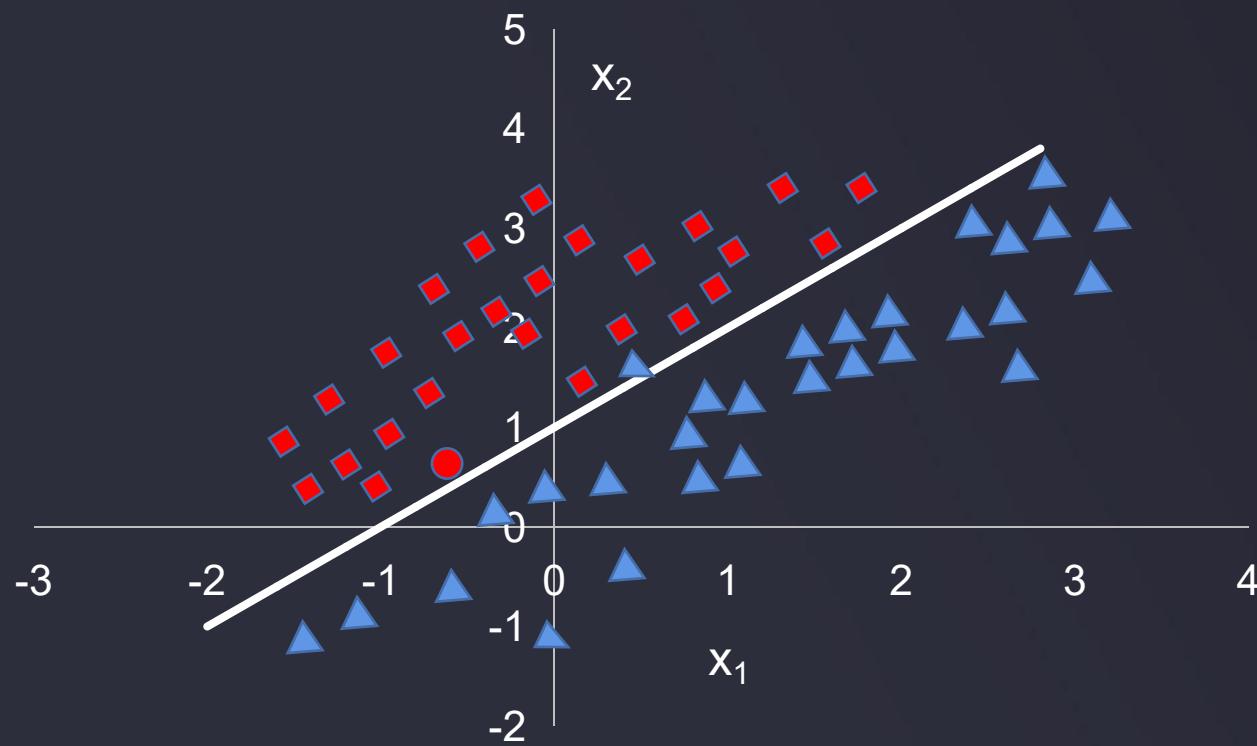


x	实际y	$Y(x)$	预测 $y(x)$
-100	0	0.00	0
-50	0	0.00	0
-10	0	0.00	0
-5	0	0.01	0
-2	0	0.12	0
-1	1	0.27	0
1	1	0.73	1
2	1	0.88	1
5	1	0.99	1
10	1	1.00	1
50	1	1.00	1
1000	1	1.00	1

逻辑回归处理更复杂的分类任务



逻辑回归处理更复杂的分类任务



$$P(x) = \frac{1}{1 + e^{-x}}$$

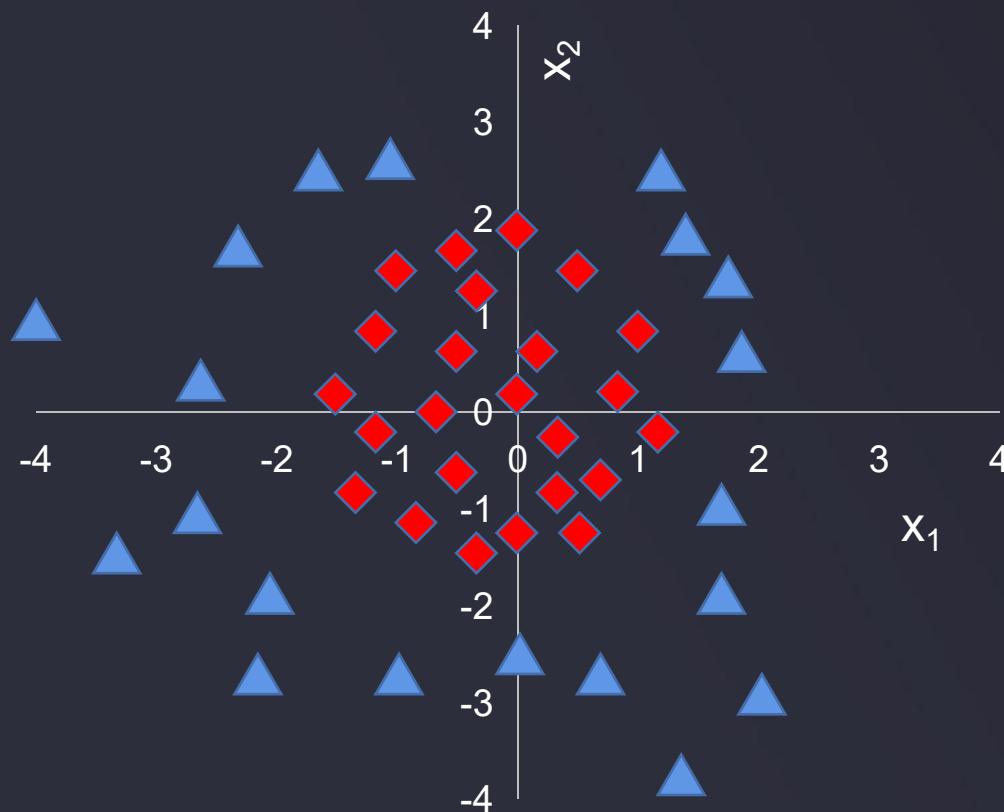
$$P(x) = \frac{1}{1 + e^{-g(x)}}$$

$$g(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$g(x) = x_2 - x_1 - 1$$

$g(x) = x_2 - x_1 - 1 > 0$: 方形
 $g(x) = x_2 - x_1 - 1 < 0$: 三角形

逻辑回归处理更复杂的分类任务



$$g(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$$

$$g(x) = x_1^2 + x_2^2 - 4$$

$x_1^2 + x_2^2 - 4 = 0$ 决策边界 (Decision Boundary)

逻辑回归处理更复杂的分类任务

- 逻辑回归结合多项式边界函数可解决复杂的分类问题
- 模型求解的核心，在于寻找到合适的多项式边界函数

$$P(x) = \frac{1}{1 + e^{-g(x)}}$$

$$g(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 \dots$$

寻找 $g(x)$ ，回忆回归问题求解方法！

逻辑回归模型求解

- 寻找损失函数极小值点
- 分类问题，结果为离散数据，需要对损失函数进行调整以适应梯度下降法求解

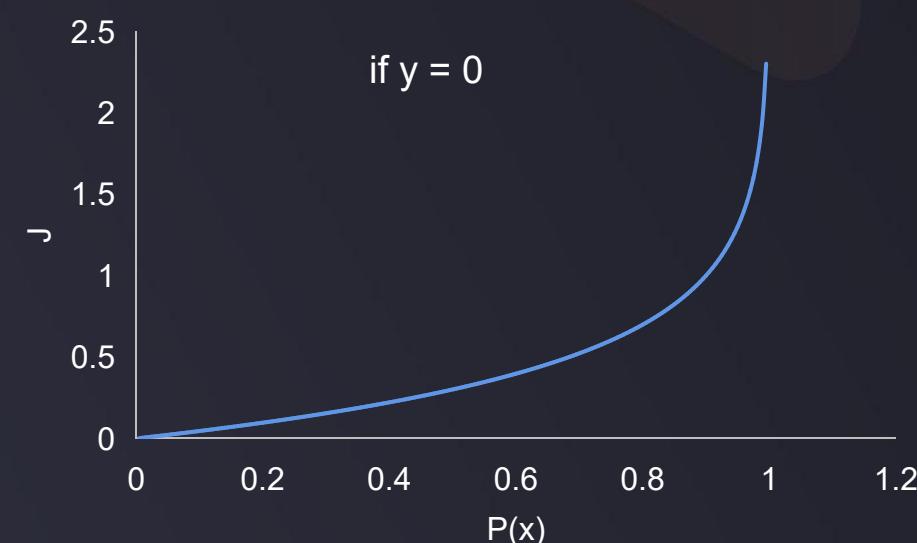
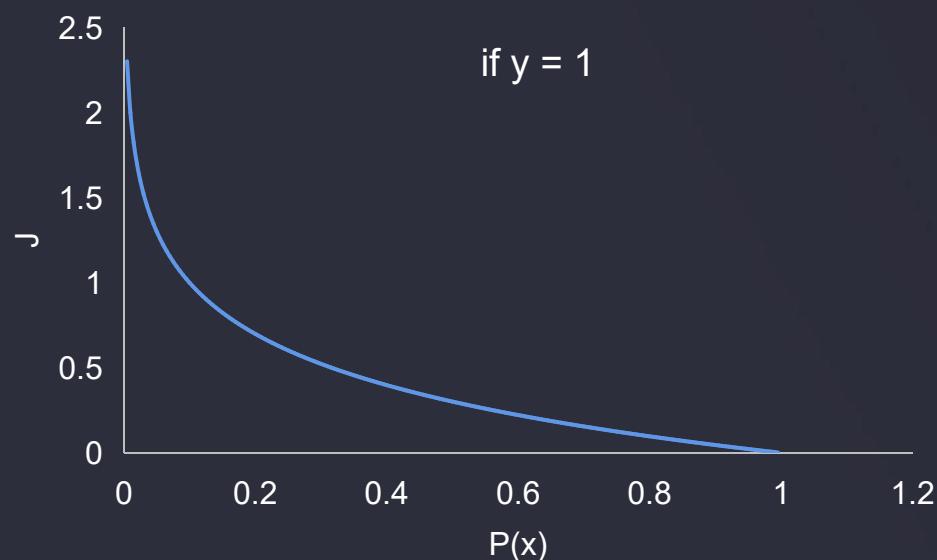
损失函数 (J) :

$$J_i = \begin{cases} -\log(P(x_i)), & \text{if } y_i = 1 \\ -\log(1 - P(x_i)), & \text{if } y_i = 0 \end{cases}$$

逻辑回归模型求解

损失函数 (J) :

$$J_i = \begin{cases} -\log(P(x_i)), & \text{if } y_i = 1 \\ -\log(1 - P(x_i)), & \text{if } y_i = 0 \end{cases}$$



逻辑回归模型求解

损失函数 (J) :

$$J_i = \begin{cases} -\log(P(x_i)), & \text{if } y_i = 1 \\ -\log(1 - P(x_i)), & \text{if } y_i = 0 \end{cases}$$

$$J = \frac{1}{m} \sum_{i=1}^m J_i = -\frac{1}{m} \left[\sum_{i=1}^m (y_i \log(P(x_i)) + (1 - y_i) \log(1 - P(x_i))) \right]$$

$$P(x) = \frac{1}{1 + e^{-g(x)}}$$

$$g(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 \dots$$

逻辑回归模型求解 (梯度下降法)

$$y = f(x) \longrightarrow \begin{aligned}x_{i+1} &= x_i - \alpha \frac{\partial}{\partial x_i} f(x_i) \\&\text{搜索方法}\end{aligned}$$

重复计算直到收敛

$$\left\{ \begin{array}{l} temp_{\theta_j} = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\ \theta_j = temp_{\theta_j} \end{array} \right\}$$

知识巩固

问题：逻辑回归如果要应用于多分类，该如何实现？



Python3人工智能入门+实战提升：机器学习

Chapter 3 分类任务与逻辑回归

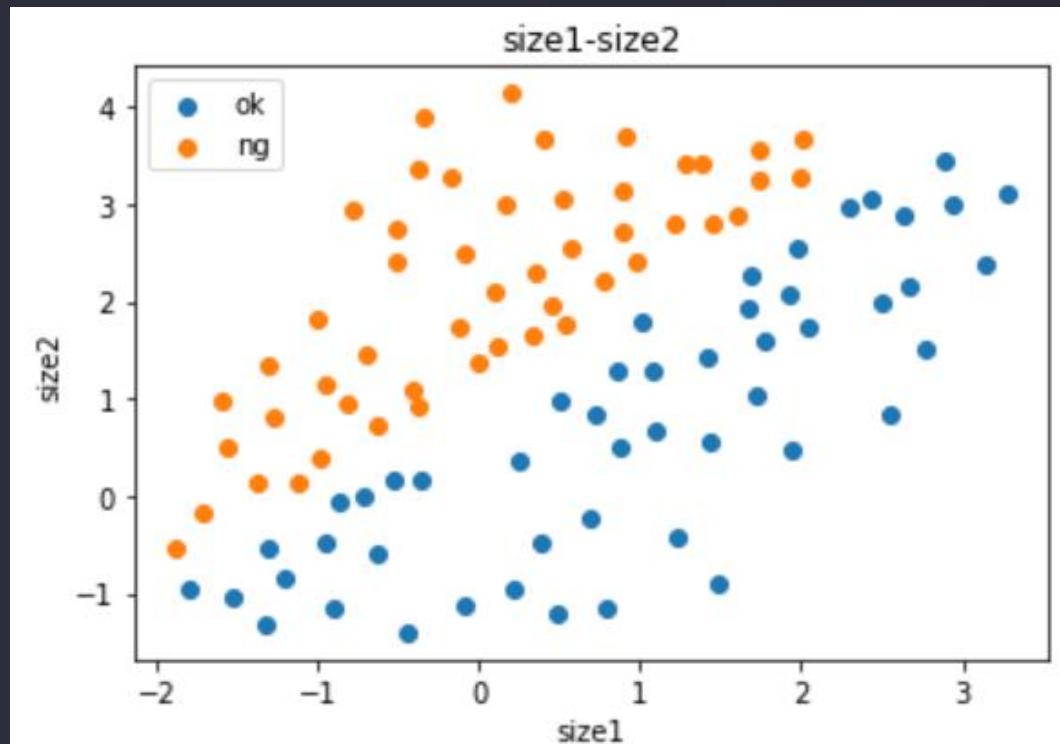
赵辛

Chapter 3 分类任务与逻辑回归

-
- 1 --分类任务
 - 2 --分类预测的实现
 - 3 --逻辑回归
 - 4 --实战准备
 - 5 --实战（一）逻辑回归实现产品分类
 - 6 --实战（二）商业异常消费数据预测

任务一：逻辑回归实现产品分类

基于课程中的线性二分类案例与task1_data数据，建立逻辑回归模型，计算并绘制边界曲线，并预测 $x_1=1, x_2=10$ 数据点属于什么类别。



- 1、基于task1_data.csv数据，建立逻辑回归模型，评估模型表现；
- 2、预测 $x_1=1, x_2=10$ 时，该产品是良品（ok）还是次品
- 3、获取边界函数参数、绘制边界函数

逻辑回归实现产品分类

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#数据加载  
import pandas as pd  
import numpy as np  
data = pd.read_csv('task1_data.csv')  
data.head()
```

	尺寸1	尺寸2	y
0	-1.78680	-0.943606	1
1	-1.52284	-1.048610	1
2	-1.31980	-1.324320	1
3	-1.29949	-0.536819	1
4	-1.19797	-0.845908	1

逻辑回归实现产品分类

数据加载及展示

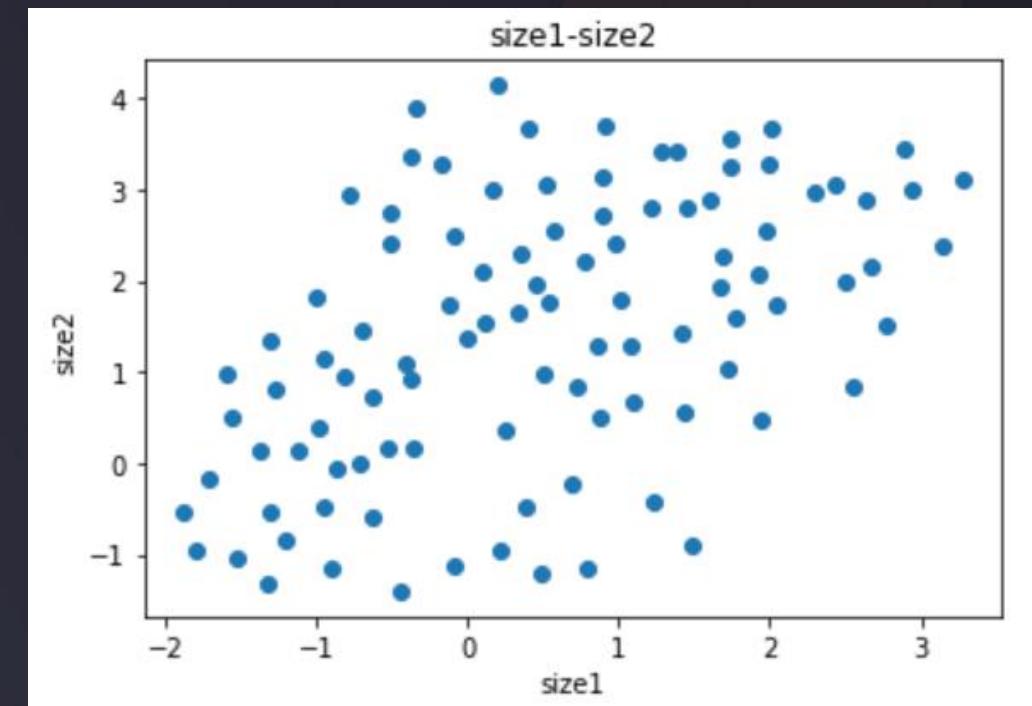
数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#数据可视化  
from matplotlib import pyplot as plt  
fig1 = plt.figure()  
plt.scatter(data.loc[:, '尺寸1'], data.loc[:, '尺寸2'])  
plt.title('size1-size2')  
plt.xlabel('size1')  
plt.ylabel('size2')  
plt.show()
```



逻辑回归实现产品分类

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#建立一个用于筛选类别的变量  
mask=data.loc[:, 'y']==1  
print(mask)
```

```
0      True  
1      True  
2      True  
3      True  
4      True  
      ...  
95     False  
96     False  
97     False  
98     False  
99     False  
Name: y, Length: 100, dtype: bool
```

逻辑回归实现产品分类

数据加载及展示

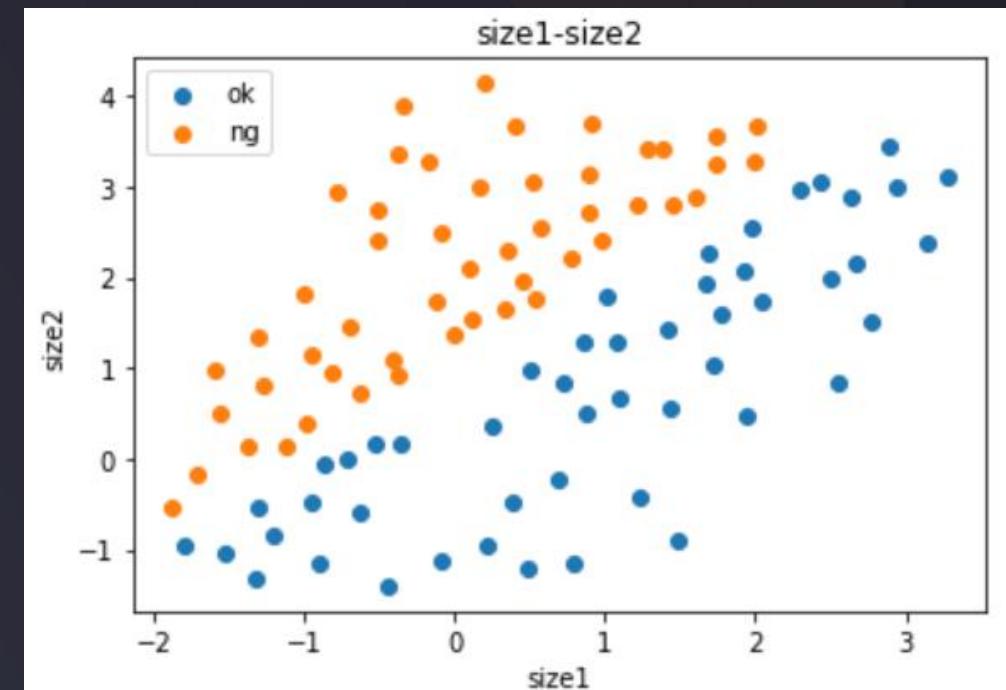
数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#数据可视化  
ok=plt.scatter(data.loc[:, '尺寸1'][mask], data.loc[:, '尺寸2'][mask])  
ng=plt.scatter(data.loc[:, '尺寸1'][~mask], data.loc[:, '尺寸2'][~mask])  
plt.title('size1-size2')  
plt.xlabel('size1')  
plt.ylabel('size2')  
plt.legend((ok,ng), ('ok', 'ng'))  
plt.show()
```



逻辑回归实现产品分类

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#X,y赋值

X = data.drop(['y'],axis=1)

y = data.loc[:, 'y']

X.head()

	尺寸1	尺寸2
0	-1.78680	-0.943606
1	-1.52284	-1.048610
2	-1.31980	-1.324320
3	-1.29949	-0.536819
4	-1.19797	-0.845908

逻辑回归实现产品分类

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#创建模型实例

```
from sklearn.linear_model import  
LogisticRegression  
model = LogisticRegression()  
print(model)
```

#模型训练

```
model.fit(x,y)
```

参考链接:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

逻辑回归实现产品分类

数据加载及展示

```
y_predict = LR.predict(X)  
print(y_predict2)
```

数据预处理

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

模型建立及训练

```
y_test = LR.predict([[1,10]])  
print('ok' if y_test==1 else 'ng')
```

模型预测

结果展示及表现评估

ng

逻辑回归实现产品分类

如何判断模型好坏？

准确率（类别正确预测的比例）：

预测y	实际y	是否预测正确
0	0	是
0	0	是
0	0	是
0	0	是
0	0	是
0	1	否
1	1	是
1	1	是
1	1	是
1	1	是

左表准确率：

$$\text{Accuracy} = \frac{\text{正确预测样本数量}}{\text{总样本数量}}$$

$$\text{Accuracy} = \frac{9}{10} = 0.9$$

准确率越接近1越好

逻辑回归实现产品分类

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#模型评估
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y,y_predict)
print(accuracy)

1.0

逻辑回归实现产品分类

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

$$\text{边界函数: } \theta_0 + \theta_1 X_1 + \theta_2 X_2 = 0$$

```
#通过边界函数计算边界  
X2_new = -(theta0+theta1*X1)/theta2  
print(X2_new)
```

```
0      -0.739752  
1      -0.484664  
2      -0.288448  
3      -0.268820  
4      -0.170713  
      ...  
95     2.537527  
96     2.671652  
97     2.672851  
98     2.907007  
99     2.924924  
Name: 尺寸1, Length: 100, dtype: float64
```

逻辑回归实现产品分类

数据加载及展示

数据预处理

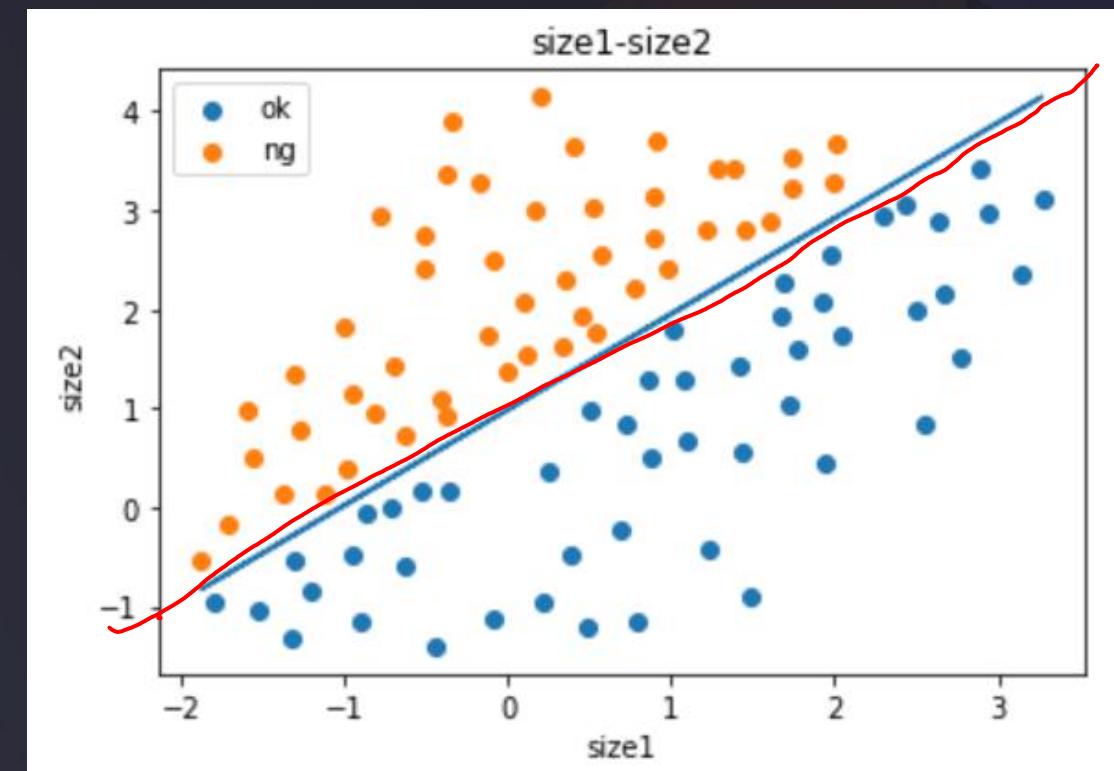
模型建立及训练

模型预测

结果展示及表现评估

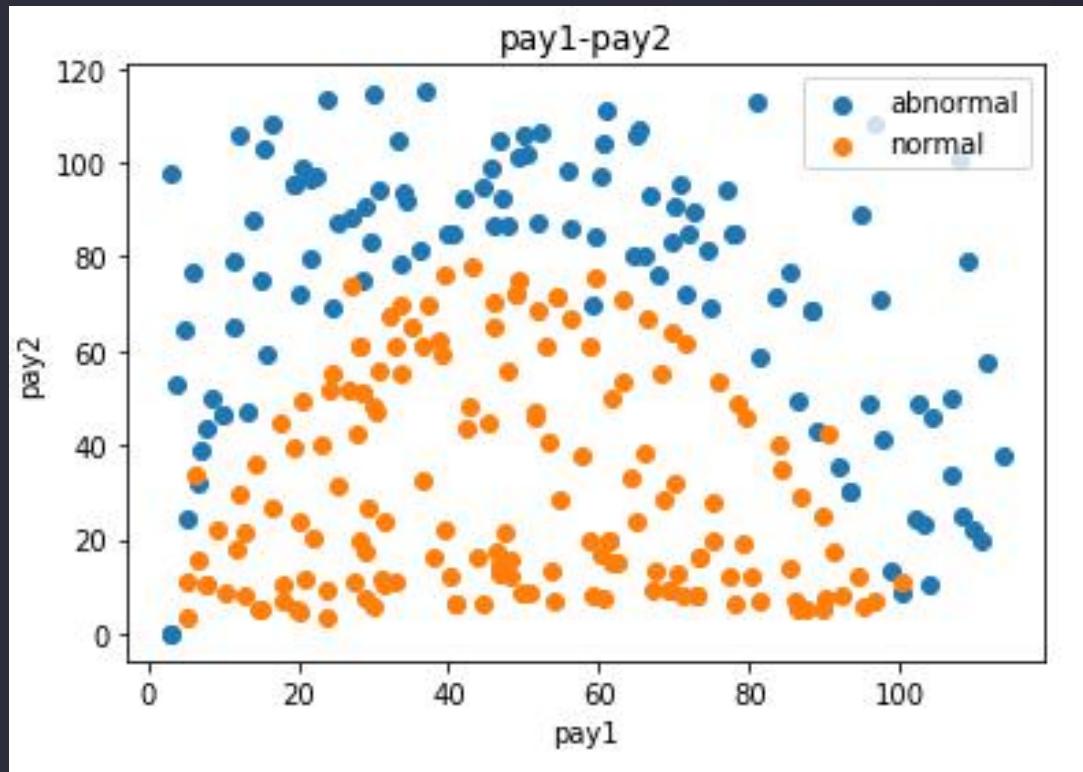
#可视化边界曲线

```
X2_new = -(theta0+theta1*X1)/theta2  
plt.plot(X1,X2_new)
```



| 任务二：商业异常消费数据预测

基于task2_data.csv数据，建立二阶多项式逻辑回归模型实现异常消费数据预测，与线性逻辑回归模型结果进行对比。



- 1、建立线性边界的逻辑回归模型，评估模型表现；
- 2、建立二阶多项式边界的逻辑回归模型，对比其与线性边界的表现
- 3、预测 $\text{pay1}=80, \text{pay2}=20$ 时对应消费是否为异常消费
- 4、获取边界函数参数、绘制边界函数

商业异常消费数据预测

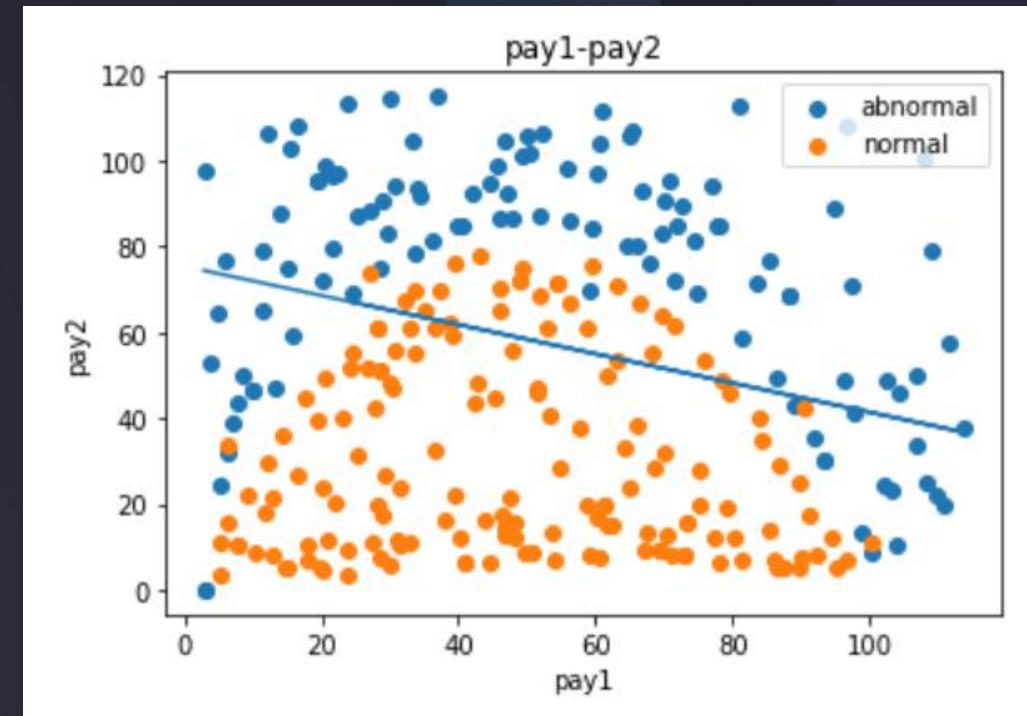
数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估



使用一阶线性边界分类效果不理想

解决办法：增加二阶项

商业异常消费数据预测

解决办法：增加二阶项

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

边界函数: $\theta_0 + \theta_1 X_1 + \theta_2 X_2 = 0$

二阶边界函数: $\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_1^2 + \theta_4 X_2^2 + \theta_5 X_1 X_2 = 0$

```
#生成二阶数据  
X1_2 = X1*X1  
X2_2 = X2*X2  
X1_X2 = X1*X2  
print(X1[0],X2[0],X1_2[0],X2_2[0],X1_X2[0])
```

2.89738 0.0579476 8.3948108644 0.00335792434576 0.167896217288

商业异常消费数据预测

解决办法：增加二阶项

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#创建新的输入数据

```
X_new =  
{'X1':X1,'X2':X2,'X1_2':X1_2,'X2_2':X2_2,'X1_X2':X1_X2}  
X_new = pd.DataFrame(X_new)  
print(X_new)
```

	X1	X2	X1_2	X2_2	X1_X2
0	2.89738	0.057948	8.394811	0.003358	0.167896
1	10.38230	8.777660	107.792153	77.047315	91.132299
2	14.00400	87.967200	196.112016	7738.228276	1231.892669
3	104.30600	45.734200	10879.741636	2091.617050	4770.351465
4	80.88530	113.010600	6542.431756	12771.395712	9140.896284
..
268	64.94970	106.091800	4218.463530	11255.470027	6890.630582
269	16.66000	108.059800	277.555600	11676.920376	1800.276268
270	19.31590	39.515800	373.103993	1561.498450	763.283241
271	63.01810	70.852400	3971.280928	5020.062586	4464.983628
272	24.38630	55.457000	594.691628	3075.478849	1352.391039

[273 rows x 5 columns]

商业异常消费数据预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

二阶边界函数: $\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_1^2 + \theta_4 X_2^2 + \theta_5 X_1 X_2 = 0$

$ax^2 + bx + c = 0 : x1 = (-b + (b^2 - 4ac)^{.5})/2a, x2 = (-b - (b^2 - 4ac)^{.5})/2a$

$\theta_4 X_2^2 + (\theta_5 X_1 + \theta_2)X_2 + (\theta_0 + \theta_1 X_1 + \theta_3 X_1^2) = 0$

商业异常消费数据预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

$$\begin{aligned} \text{二阶边界函数: } & \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_1^2 + \theta_4 X_2^2 + \theta_5 X_1 X_2 = 0 \\ ax^2 + bx + c = 0 : & x_1 = (-b + (b^2 - 4ac)^{.5})/2a, x_2 = (-b - (b^2 - 4ac)^{.5})/2a \\ \theta_4 X_2^2 + (\theta_5 X_1 + \theta_2)X_2 + (\theta_0 + \theta_1 X_1 + \theta_3 X_1^2) = 0 \end{aligned}$$

```
#获取边界函数系数、生成新的边界数据
theta0 = LR2.intercept_
theta1,theta2,theta3,theta4,theta5 =
LR2.coef_[0][0],LR2.coef_[0][1],LR2.coef_[0][2],LR2.coef_[0]
[3],LR2.coef_[0][4]
a = theta4
b = theta5*X1_new+theta2
c = theta0+theta1*X1_new+theta3*X1_new*X1_new
X2_new_boundary = (-b+np.sqrt(b*b-4*a*c))/(2*a)

print(X2_new_boundary)
```

商业异常消费数据预测

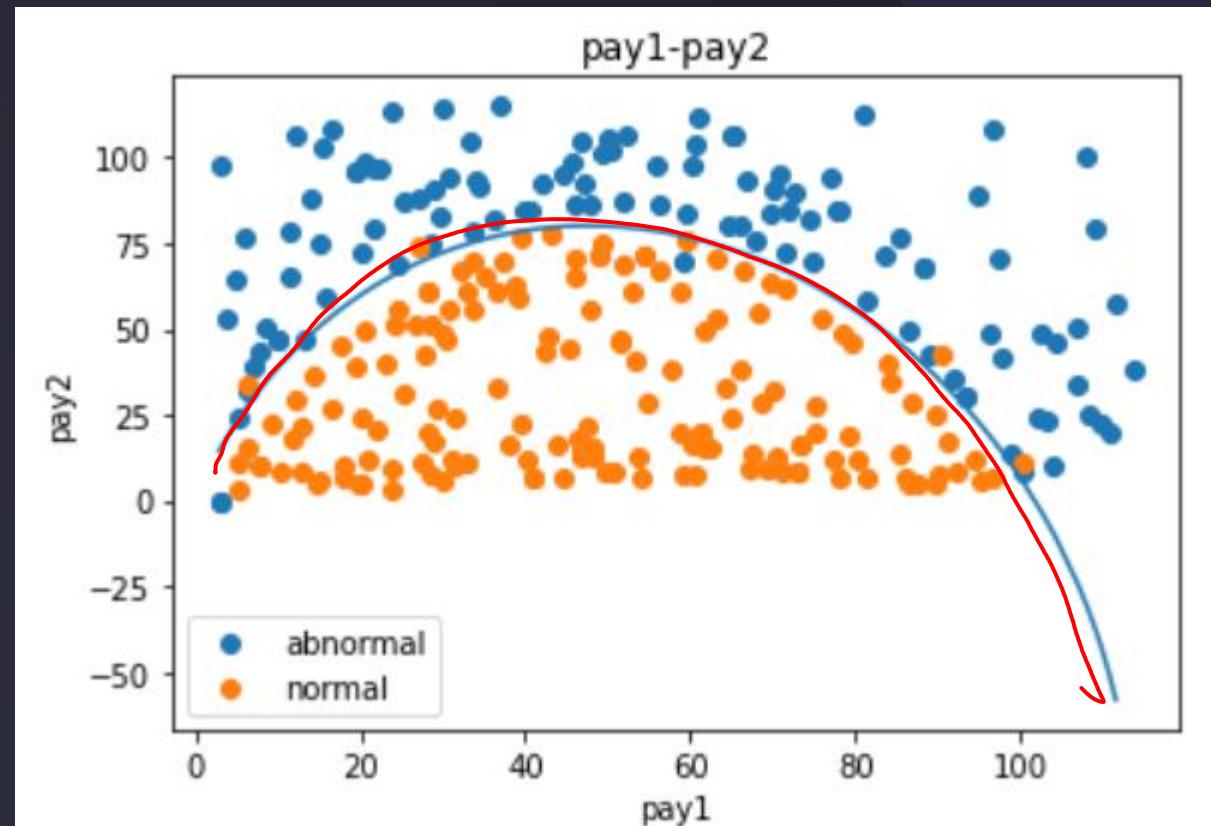
数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估



| 任务三：为毕业与工作做准备

如果毕业设计是搭建分类模型，预测消费者是否会购买商品，我们通常做些什么？

- 1、前期工作：调研、讨论并确认影响购买意愿的因素；数据采集
- 2、数据预处理：异常数据处理、信息量化
- 3、建模与训练：从简单到复杂的决策边界模型
- 4、预测、评估、优化：引入不同的评估指标、尝试不同的模型
- 5、总结与汇报：结果整理并分析、输出项目报告



Python3人工智能入门+实战提升：机器学习

Chapter 3 分类任务与逻辑回归

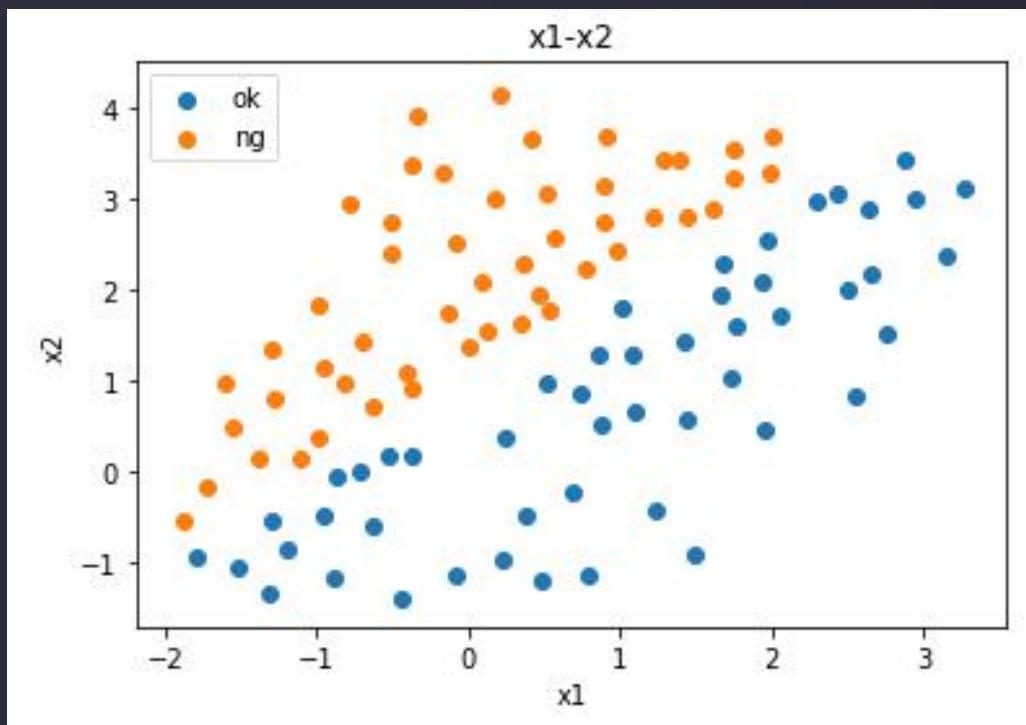
赵辛

Chapter 3 分类任务与逻辑回归

-
- 1 --分类任务
 - 2 --分类预测的实现
 - 3 --逻辑回归
 - 4 --实战准备
 - 5 --实战（一）逻辑回归实现产品分类
 - 6 --实战（二）商业异常消费数据预测

任务一：逻辑回归实现产品分类

基于课程中的线性二分类案例与task1_data数据，建立逻辑回归模型，计算并绘制边界曲线，并预测 $x_1=1, x_2=10$ 数据点属于什么类别。



- 1、基于task1_data.csv数据，建立逻辑回归模型，评估模型表现；
- 2、预测 $x_1=1, x_2=10$ 时，该产品是良品（ok）还是次品
- 3、获取边界函数参数、绘制边界函数



Python3人工智能入门+实战提升：机器学习

Chapter 3 分类任务与逻辑回归

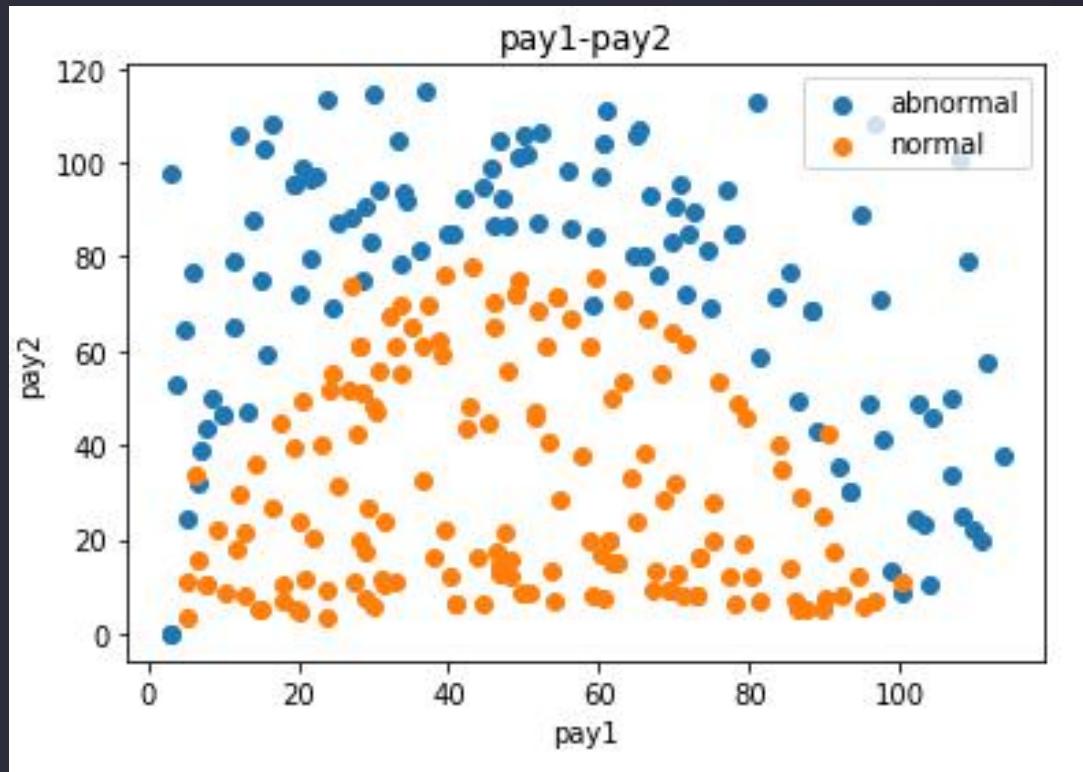
赵辛

Chapter 3 分类任务与逻辑回归

-
- 1 --分类任务
 - 2 --分类预测的实现
 - 3 --逻辑回归
 - 4 --实战准备
 - 5 --实战（一）逻辑回归实现产品分类
 - 6 --实战（二）商业异常消费数据预测

| 任务二：商业异常消费数据预测

基于task2_data.csv数据，建立二阶多项式逻辑回归模型实现异常消费数据预测，与线性逻辑回归模型结果进行对比。



- 1、建立线性边界的逻辑回归模型，评估模型表现；
- 2、建立二阶多项式边界的逻辑回归模型，对比其与线性边界的表现
- 3、预测 $\text{pay1}=80, \text{pay2}=20$ 时对应消费是否为异常消费
- 4、获取边界函数参数、绘制边界函数

| 毕设、工作综合项目的思考与建议

如果毕业设计是搭建分类模型，预测消费者是否会购买商品，我们通常做些什么？

- 1、前期工作：调研、讨论并确认影响购买意愿的因素；数据采集
- 2、数据预处理：异常数据处理、信息量化
- 3、建模与训练：从简单到复杂的决策边界模型
- 4、预测、评估、优化：引入不同的评估指标、尝试不同的模型
- 5、总结与汇报：结果整理并分析、输出项目报告



Python3人工智能入门+实战提升：机器学习

Chapter 3 分类任务与逻辑回归

赵辛

Chapter 3 分类任务与逻辑回归

-
- 1 --分类任务
 - 2 --分类预测的实现
 - 3 --逻辑回归
 - 4 --实战准备
 - 5 --实战（一）逻辑回归实现产品分类
 - 6 --实战（二）商业异常消费数据预测

| 毕设、工作综合项目的思考与建议

如果毕业设计是搭建分类模型，预测消费者是否会购买商品，我们通常做些什么？

- 1、前期工作：调研、讨论并确认影响购买意愿的因素；数据采集
- 2、数据预处理：异常数据处理、信息量化
- 3、建模与训练：从简单到复杂的决策边界模型
- 4、预测、评估、优化：引入不同的评估指标、尝试不同的模型
- 5、总结与汇报：结果整理并分析、输出项目报告



Python3人工智能入门+实战提升：机器学习

Chapter 4 其他常用分类技术

赵辛

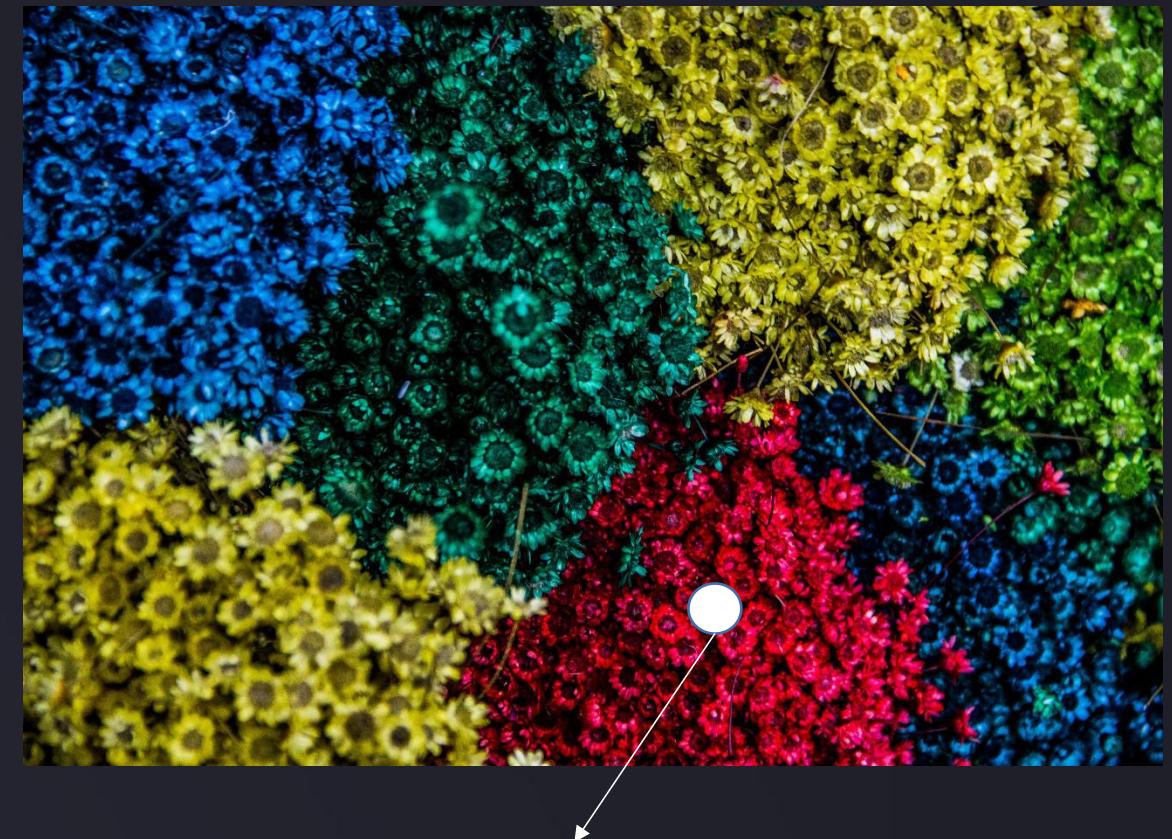
Chapter 4 其他常用分类技术

-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

|现实问题思考

“物以类聚，人以群分”

同类的东西常聚在一起，志同道合的人相聚成群



如果有一束花在这个位置，最有可能是什么颜色的花？

| K近邻分类模型(K-nearest neighbors)

通过计算新数据与训练数据之间的距离，然后选取K ($K \geq 1$) 个距离最近的邻居进行分类判断 (K个邻居) ， 这K个邻居的多数属于某个类，就把该新数据实例分类到这个类中。

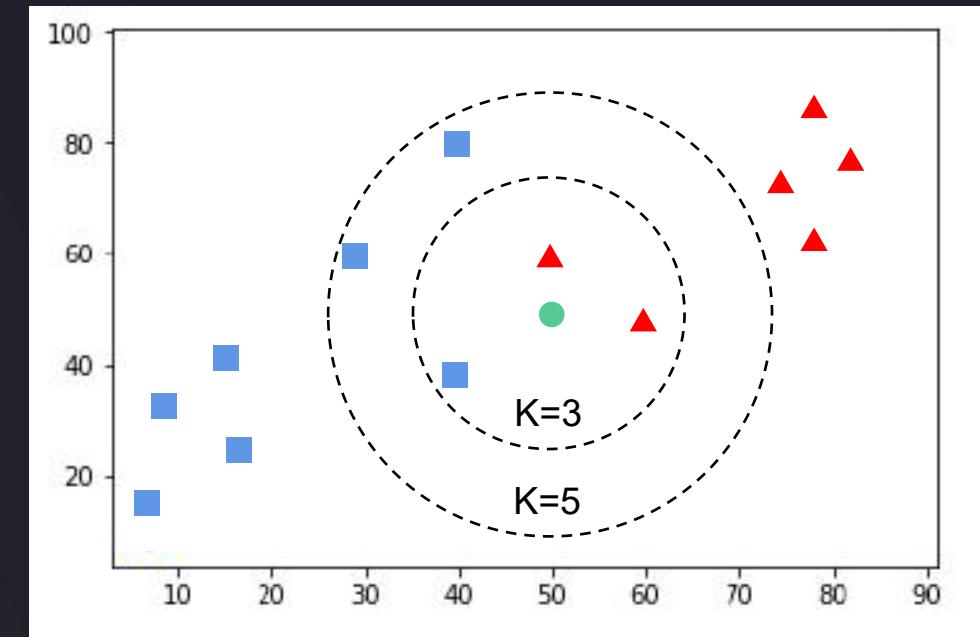
- 最简单的机器学习算法之一

K近邻分类模型(K-nearest neighbors)

举例：判断图中圆圈属于哪个类别

$K=3$, 绿色圆点(50,50)的最近的3个邻居是2个红色小三角形(60,50)、(50,60)和1个蓝色小正方形(40,40), 判定其属于红色的三角形一类。

$K=5$, 绿色圆点的最近的5个邻居是2个红色三角形(60,50)、(50,60)和3个蓝色的正方形(40,40)、(40,80)、(30,60), 判定其属于蓝色的正方形一类。



我和谁一队？看看你周围哪个队的人多！

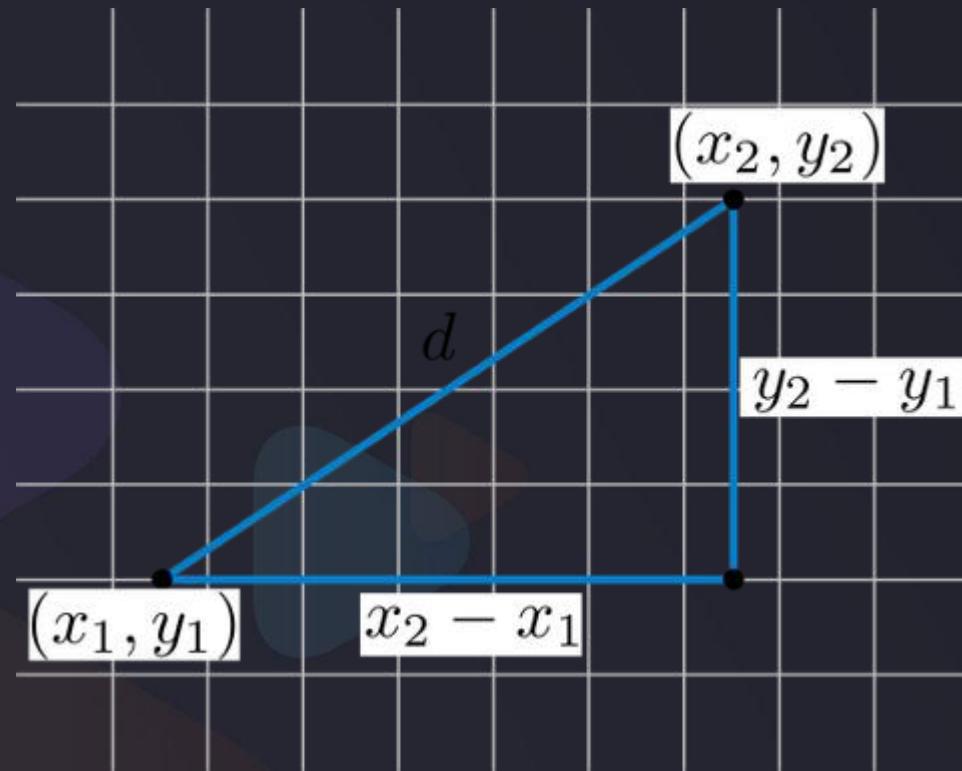
K近邻分类模型算法步骤

输入：训练数据集 $D = \{(x_1, y_1), (x_2, y_2) \dots (x_i, y_i) \dots (x_m, y_m)\}$, 其中 x_i 为数据的特征向量, y_i 代表数据所属类别, 对于新的样本数据 x_{test} :

- (1) 计算训练数据集每个样本 x_i 与新的样本数据 x_{test} 的距离 d_{i-test} ;
- (2) 将计算出的距离按照升序排列, 并取出前 K 个距离最小的样本;
- (3) 统计这 K 个样本的标签值 y , 并找出出现频率最高的标签;
- (4) 新的样本数据 x_{test} 的标签值 y_{test} 即为该频率最高的标签值。

K近邻分类模型-距离计算方式

欧氏距离：两点之间的直线距离，KNN计算中应用最多距离。

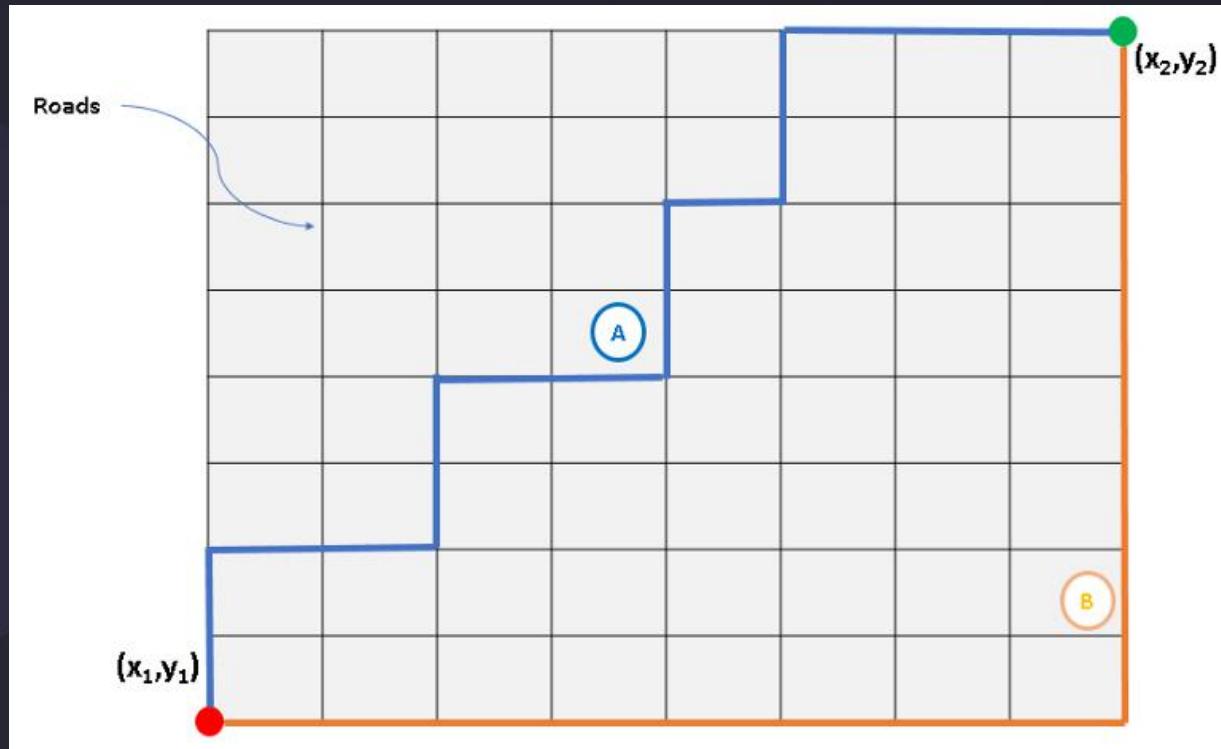


$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

x, y 均为数据特征，不是结果标签

K近邻分类模型-距离计算方式

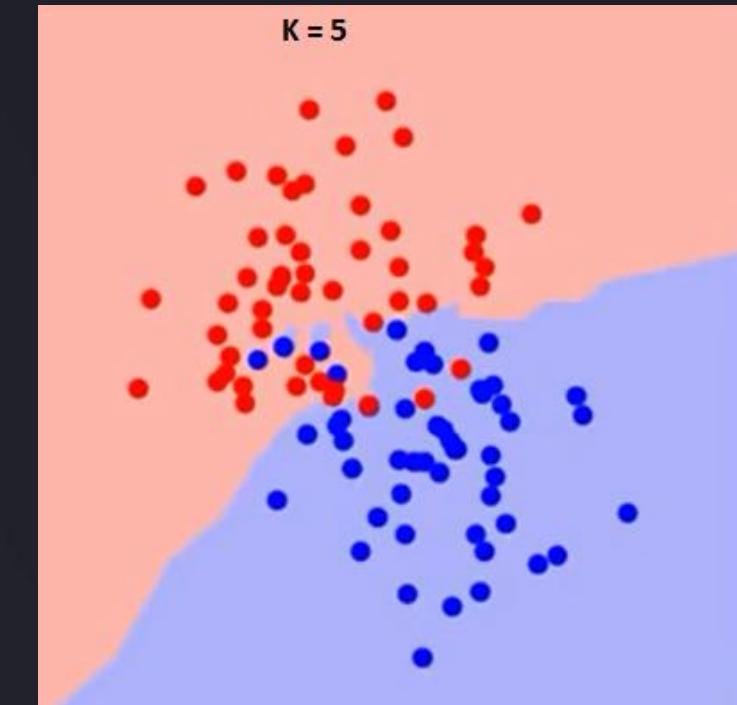
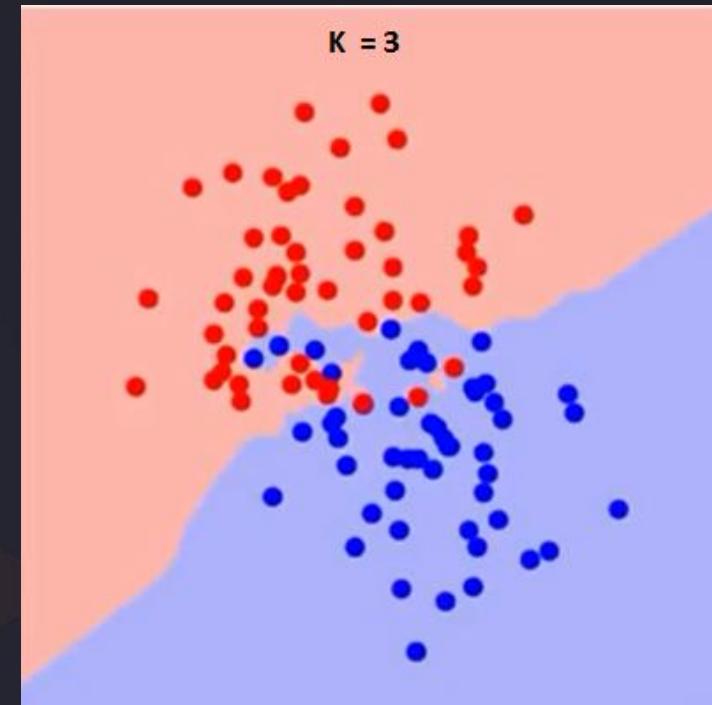
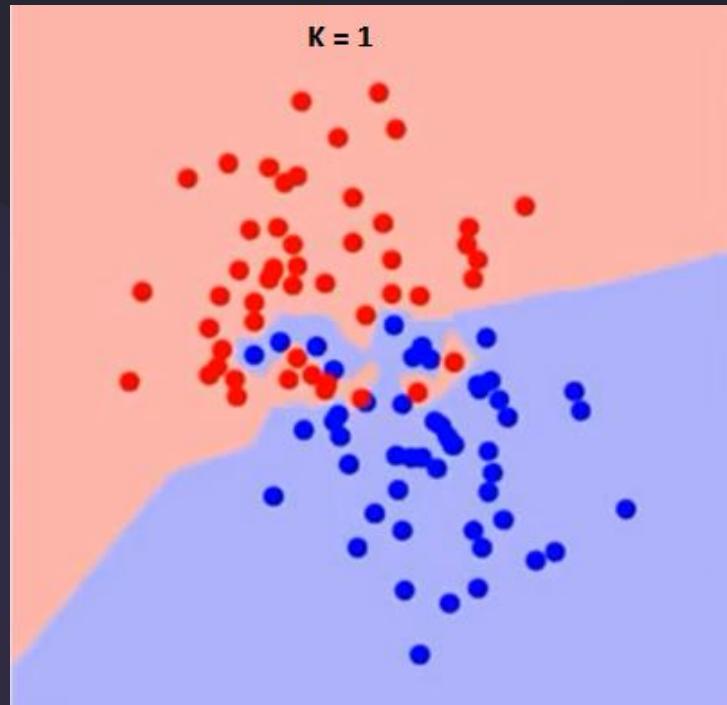
曼哈顿距离：两个点在标准坐标系上的绝对轴距总和



$$|x_2 - x_1| + |y_2 - y_1|$$

x, y 均为数据特征，不是结果标签

| KNN分类图



K 值越小，分类边界越曲折，抗干扰性更弱（噪声数据影响结果明显）

知识巩固

问题：训练数据集如下表所示

x1	x2	x3	x4	y
1	5	-3	10	1
5	7	6	0	0
4	0	1	3	0
7	6	8	2	1

新数据 $(3, 2, 7, -1)$ ，计算其与训练数据各点的欧氏距离、曼哈顿距离，并基于KNN算法原理判断其类别 ($K=2$)



Python3人工智能入门+实战提升：机器学习

Chapter 4 其他常用分类技术

赵辛

Chapter 4 其他常用分类技术

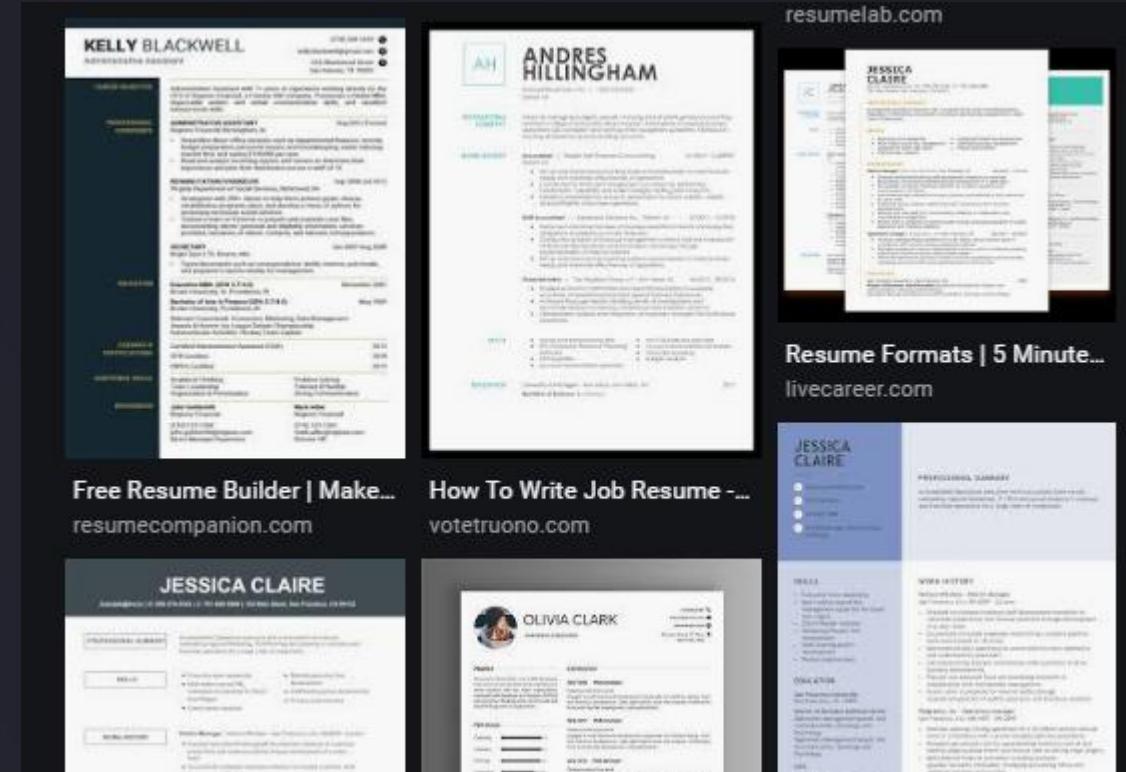
-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

|现实问题思考

“求职简历太多，给不给面试机会？”

简历上有什么：

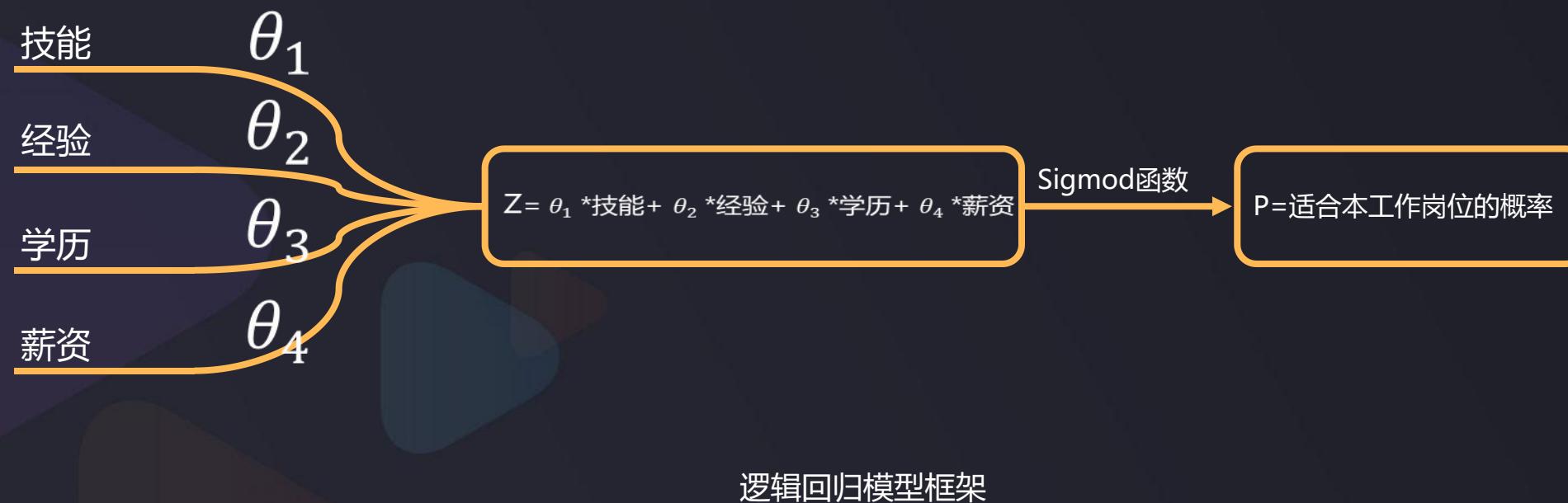
个人技能、工作经验、学校学历、期望薪资等



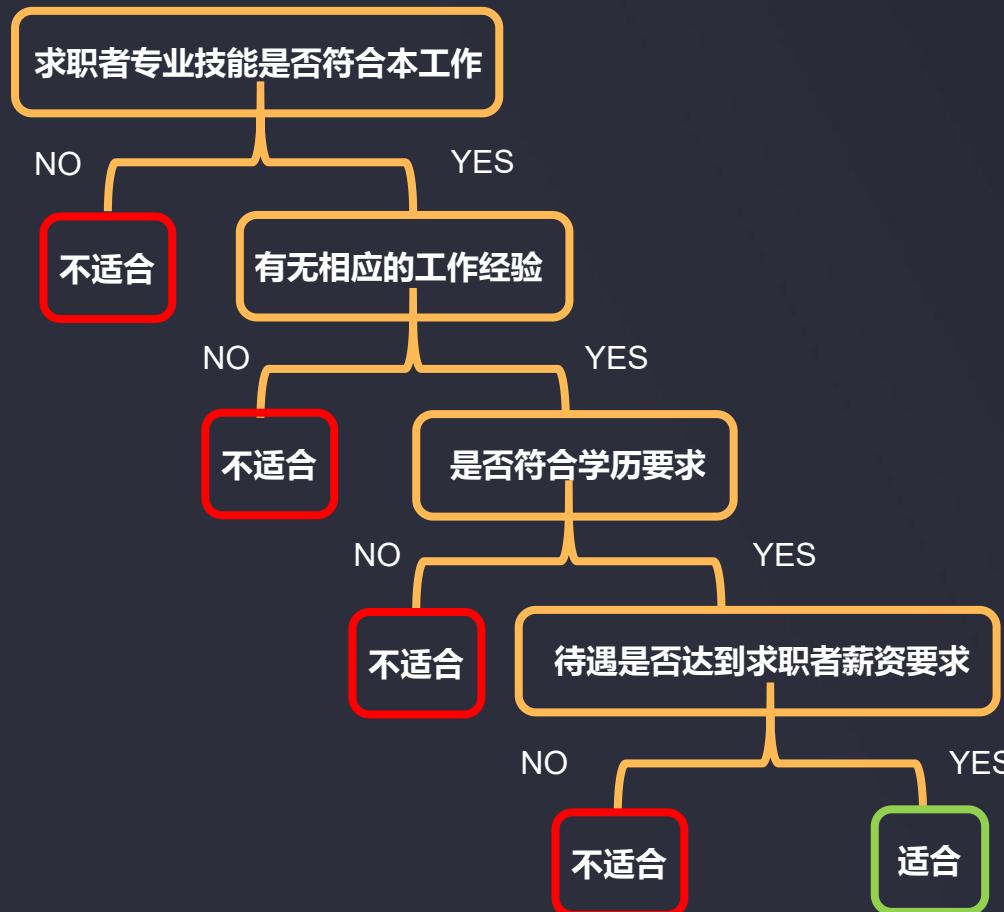
|逻辑回归VS决策树

任务：

根据求职者的相应技能、工作经验、学历背景和薪资要求判断能否安排该求职者面试。



逻辑回归VS决策树



问：求职者是否有本岗位相应的专业技能？

答：有

问：求职者是否有本岗位相关的工作经验？

答：有

问：求职者是否符合学历要求？

答：符合

问：公司给出的待遇是否达到求职者薪资要求？

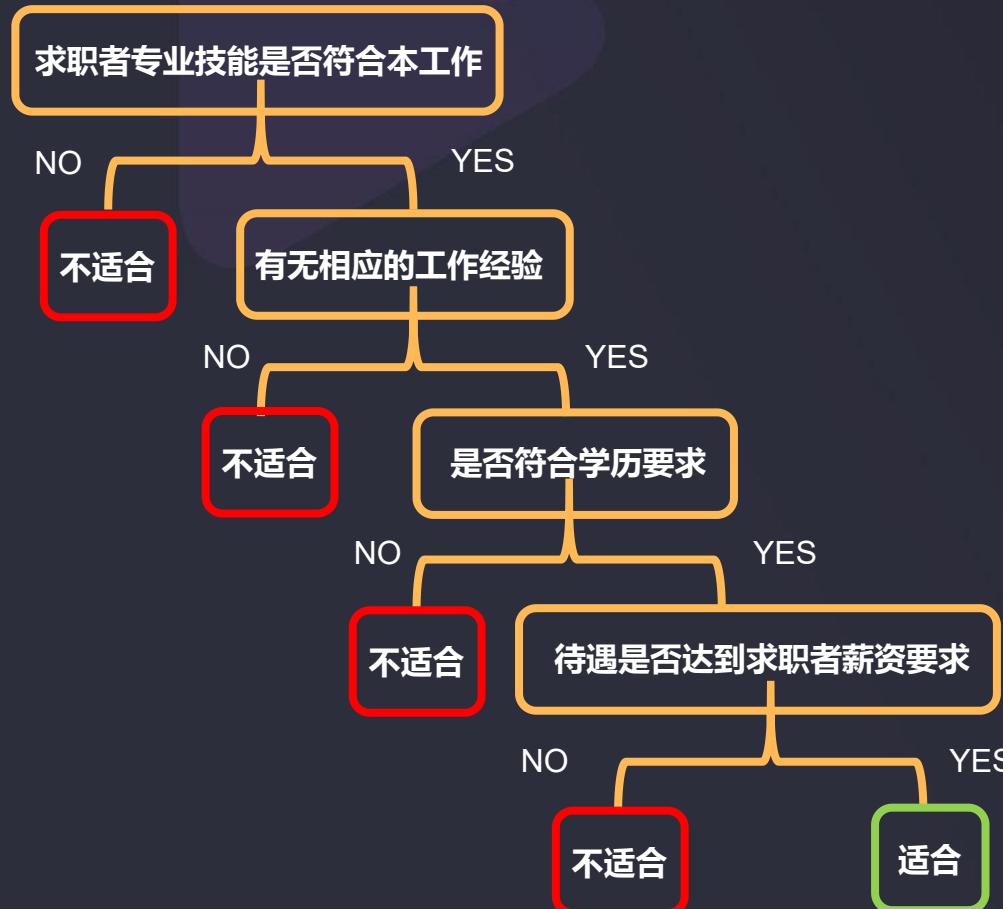
答：达到

结论：该求职者可以安排面试

决策树模型框架

决策树

一种基于样本分布概率，以树形结构的方式，实现多层判断从而确定目标所属类别



假设给定训练数据集

$$D = \{(x_1, y_1), (x_2, y_2) \dots (x_i, y_i) \dots (x_m, y_m)\}$$

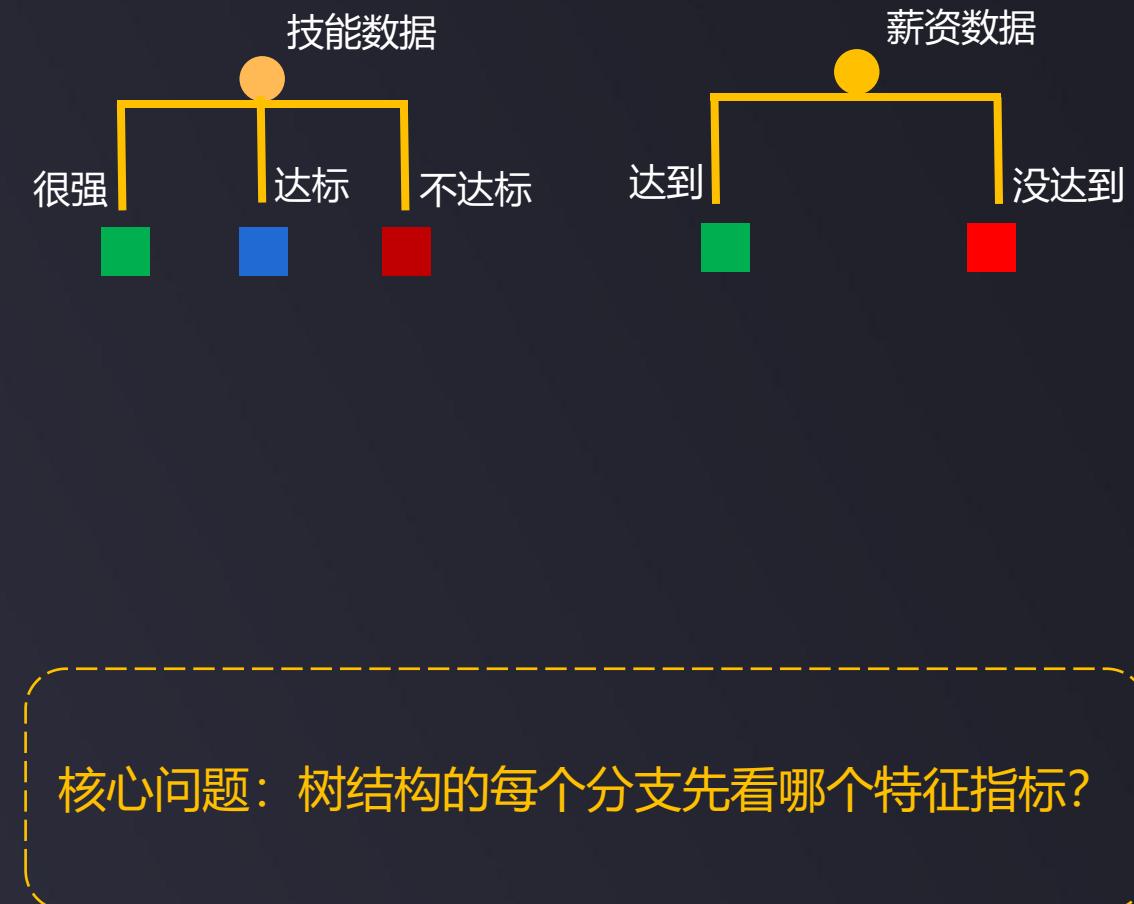
其中， $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})^T$ 为输入实例，n为特征个数，
 $y_i \in \{1, 2, 3, \dots, K\}$ 为类标记， $i = 1, 2, \dots, m$, m为样本容量。

根据数据集D的分布，生成树形结构，实现最终类别判断

决策树求解

员工背景与工作匹配度统计数据

ID	技能	经验	学历	薪资	类别
1	达标	无	不符合	达到	不适合
2	达标	无	符合	没达到	不适合
3	很强	有	符合	达到	适合
4	达标	无	不符合	达到	不适合
5	达标	无	不符合	没达到	不适合
6	达标	有	不符合	没达到	不适合
7	达标	有	符合	达到	适合
8	达标	有	符合	达到	适合
9	很强	有	符合	达到	适合
10	不达标	无	不符合	没达到	不适合



| 决策树求解

三种求解方法：

ID3、C4.5、CART

参考资料：

1.

https://blog.csdn.net/dfly_zx/article/details/107797695

2.

https://blog.csdn.net/dfly_zx/article/details/107797864

利用信息熵原理选择信息增益最大的属性作为分类属性，依次确定决策树的分枝，
完成决策树的构造

决策树求解

信息熵 (entropy) 是度量随机变量不确定性的指标，**熵越大，样本的不确定性就越大**。假定当前样本集合D中第k类样本所占的比例为 p_k ，则D的信息熵为：

$$Ent(D) = - \sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

$Ent(D)$ 的值越小，样本分布的不确定性越小。

$p_k = 1$ 或 $p_k = 0$ 时（样本分布完全确定）：

$$Ent(D) = 0$$

决策树求解

根据信息熵，可以计算以属性a进行样本划分带来的信息增益：

$$Gain(D, a) = Ent(D) - \sum_{v=1}^V \frac{D^v}{D} Ent(D^v)$$

V 为根据属性a划分出的类别数、 D 为当前样本总数， D^v 为类别 v 样本数

$Ent(D)$ ：划分前的信息熵

$\sum_{v=1}^V \frac{D^v}{D} Ent(D^v)$ ：划分后的信息熵

目标：划分后样本分布不确定性尽可能小，即划分后信息熵小，信息增益大

决策树求解

$$Ent(D) = - \sum_{k=1}^{|Y|} p_k \log_2 p_k$$

$$Gain(D, \alpha) = Ent(D) - \sum_{v=1}^V \frac{D^v}{D} Ent(D^v)$$

员工背景与工作匹配度统计数据

ID	技能	经验	学历	薪资	类别
1	达标	有	符合	没达到	不适合
2	达标	无	符合	没达到	不适合
3	很强	有	符合	达到	适合
4	达标	无	不符合	达到	不适合
5	达标	无	不符合	没达到	不适合
6	达标	有	不符合	没达到	不适合
7	达标	有	符合	达到	适合
8	达标	有	符合	达到	适合
9	很强	有	符合	达到	适合
10	不达标	有	不符合	没达到	不适合

决策树求解

$$Ent(D) = - \sum_{k=1}^{|Y|} p_k \log_2 p_k$$

$$Gain(D, \alpha) = Ent(D) - \sum_{v=1}^V \frac{D^v}{D} Ent(D^v)$$

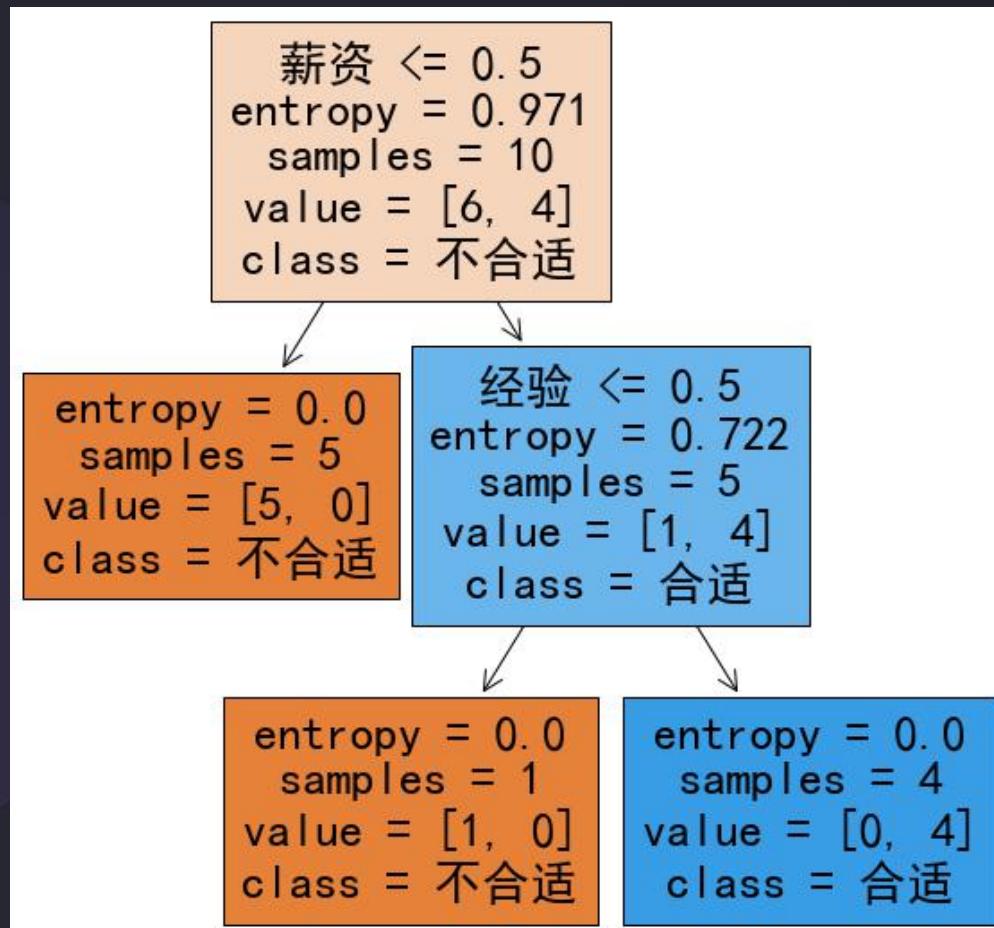
员工背景与工作匹配度统计数据

	技能	经验	学历	薪资	
Ent	0.6	0.69	0.55	0.36	
Gain	0.37	0.28	0.42	0.61	

ID	技能	经验	学历	薪资	类别
1	达标	有	符合	没达到	不适合
2	达标	无	符合	没达到	不适合
3	很强	有	符合	达到	适合
4	达标	无	不符合	达到	不适合
5	达标	无	不符合	没达到	不适合
6	达标	有	不符合	没达到	不适合
7	达标	有	符合	达到	适合
8	达标	有	符合	达到	适合
9	很强	有	符合	达到	适合
10	不达标	有	不符合	没达到	不适合

决策树求解

决策树



员工背景与工作匹配度统计数据

ID	技能	经验	学历	薪资	类别
1	达标	有	符合	没达到	不适合
2	达标	无	符合	没达到	不适合
3	很强	有	符合	达到	适合
4	达标	无	不符合	达到	不适合
5	达标	无	不符合	没达到	不适合
6	达标	有	不符合	没达到	不适合
7	达标	有	符合	达到	适合
8	达标	有	符合	达到	适合
9	很强	有	符合	达到	适合
10	不达标	有	不符合	没达到	不适合

| 知识巩固

问题：

根据课程中判断面试者工作合适与否的案例数据，计算使用ID3算法建立决策树时各节点的信息增益



Python3人工智能入门+实战提升：机器学习

Chapter 4 其他常用分类技术

赵辛

Chapter 4 其他常用分类技术

-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

现实问题思考

为什么赌博总是“输多赢少”？



骰宝

骰宝是赌场里最简单的游戏之一，俗称押大小。三个骰子加起来的点数小于等于10为小，大于等于11为大。看似有50%胜率的游戏，游戏还有一条额外的规则，叫围骰（三个骰一样的点数，比如3个6），这个时候不管玩家押大押小，都算玩家输，赌场赢。
实际上玩家胜率只有48.61%。

| 概率(Probability)

概率是一个在0到1之间的实数，是对随机事件发生可能性的度量，反映某种情况出现的可能性 (likelihood)大小。



|机器学习中的概率

分类任务中，逻辑回归模型直接预测的结果是某种情况对应的概率。



Apple: 0.97
Orange: 0.03



Apple: 0.01
Orange: 0.99



Apple: 0.02
Orange: 0.98



Apple: 0.99
Orange: 0.01

机器学习中的概率

市场交易预测中，操作建议 基于 股票价格的涨、跌的概率。



预测为1 (涨幅>3%)概率： 0.9
预测为0 (涨幅<=3%)概率： 0.1

预测为1 (涨幅>3%)概率： 0.2
预测为0 (涨幅<=3%)概率： 0.8

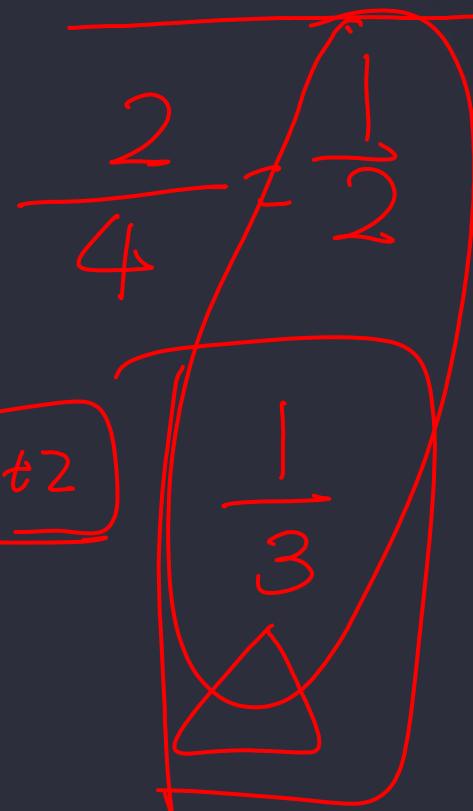
参考链接：

https://blog.csdn.net/dfly_zx/article/details/104461097

现实问题思考

$1 \rightarrow \frac{1+1.1}{1}$

两个白球、两个黑球，从中抽取一个，如果为白球，退还下注并奖励1.1倍，玩家是否应该下注？如果抽取的第一个为白球并且不放回，游戏继续，玩家是否应该下注？



$$\begin{aligned} & \frac{1}{2} \times 1.1 + \left(\frac{1}{2}\right) \times (-1) \\ &= \frac{1}{2} \times 0.1 = 0.05 > 0 \end{aligned}$$

$$\begin{aligned} & \frac{1}{3} \times 1.1 + \frac{2}{3} \times (-1) \\ &= \frac{1.1}{3} - \frac{2}{3} < 0 \end{aligned}$$

条件概率

第1次白

第2次白

定义：事件A已经发生的条件下事件B发生的概率，表示为 $P(B|A)$

$$P(B|A) = \frac{P(AB)}{P(A)}$$

$$P(AB) = \frac{2}{4} \times \frac{1}{3} = \frac{1}{6}$$

$$P(A) = \frac{2}{4} = \frac{1}{2}$$

$$\Rightarrow P(B|A) = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3}$$

|现实问题思考

两个箱子中，第一箱装有4个黑球1个白球，第二箱装有3个黑球2个白球，现任取一箱，再从该箱中任取一球，试求：取出的球是白球的概率。



条件概率公式变形

$$P(A|B_1) = \frac{P(AB_1)}{P(B_1)} \Rightarrow P(AB_1) = P(A|B_1) \times P(B_1)$$

$$P(A|B_2) = \frac{P(AB_2)}{P(B_2)} \quad P(AB_2) = P(A|B_2) \times P(B_2)$$

对于事件 i :

$$P(A|B_i) = \frac{P(AB_i)}{P(B_i)} \Rightarrow P(AB_i) = P(A|B_i) \times P(B_i)$$

全概率公式

$$P(AB_1) = P(A|B_1) \times P(B_1)$$

$$P(AB_2) = P(A|B_2) \times P(B_2)$$

$$P(AB_i) = P(A|B_i) \times P(B_i)$$

前提：各个B事件互相独立，不会同时发生

A事件发生的概率为：

$$\frac{P(A)}{\sum_{i=1}^n P(A|B_i)P(B_i)} = P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + \cdots + P(A|B_n)P(B_n)$$

将复杂事件A的概率求解问题，转化为在不同情况下发生的简单事件的概率的求和问题。

|现实问题思考

两个箱子中，第一箱装有4个黑球1个白球，第二箱装有3个黑球2个白球，现任取一箱，再从该箱中任取一球，试求：取出的球是白球的概率。



| 知识巩固

问题：有三个盒子甲乙丙，甲装了两个红球，乙装了一红一蓝两个球，丙装了两个蓝球。随机取一个盒子，从该盒子中随机取一个球，计算是红球的概率。如果第一个球确实是红球，求该盒子中另一个球也是红球的概率？



Python3人工智能入门+实战提升：机器学习

Chapter 4 其他常用分类技术

赵辛

Chapter 4 其他常用分类技术

-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

|现实问题思考

两个箱子中，第一箱装有4个黑球1个白球，第二箱装有3个黑球2个白球，现任取一箱，再从该箱中任取一球为白球，试求：取出的球是第一个箱子的概率。

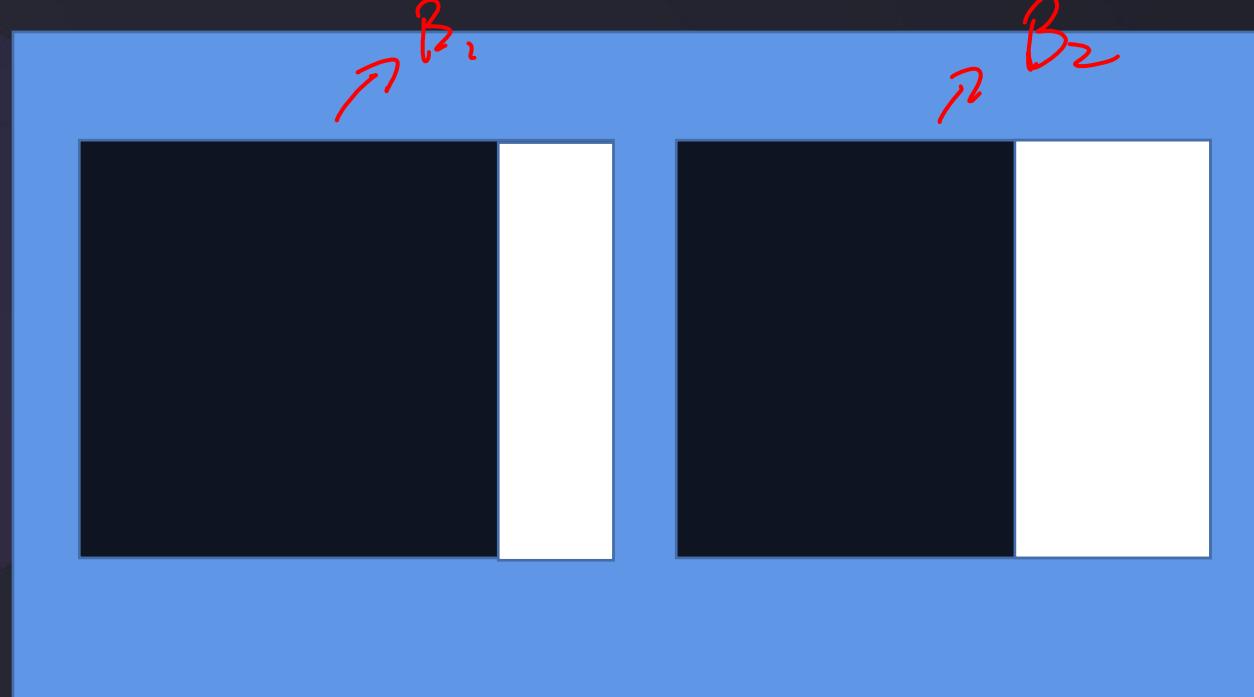


现实问题思考

$$P(B_1|A) = \frac{P(B_1) \cdot P(A|B_1)}{P(A)}$$

两个箱子中，第一箱装有4个黑球1个白球，第二箱装有3个黑球2个白球，现任取一箱，再从该箱中任取一球为白球，试求：取出的球是第一个箱子的概率。

案例转化：该箱子为第一个箱子的事件记为事件 B_1 ，取出来的球为白球的事件记为事件 A ，相当于计算 $P(B_1|A)$



↑

$$P(B_1|A) = \frac{P(B_1 A)}{P(A)}$$

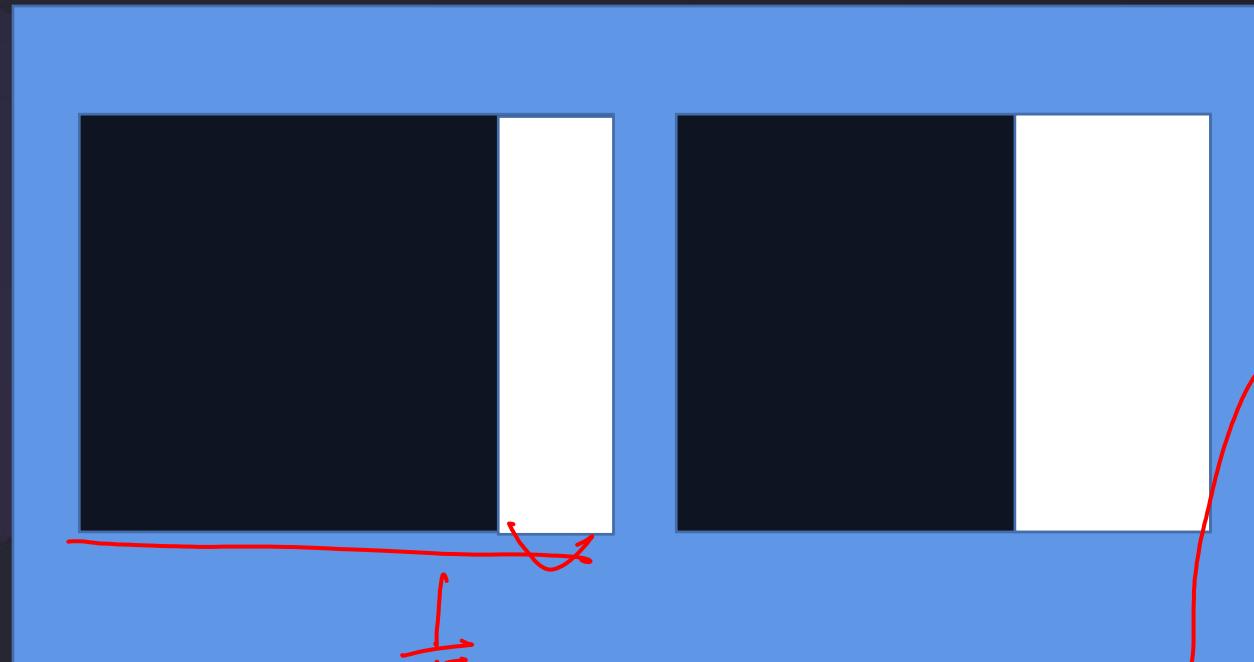
$$= \frac{P(A|B_1)}{P(A)}$$

$$P(A|B_1) = \frac{P(AB_1)}{P(B_1)}$$

现实问题思考

两个箱子中，第一箱装有4个黑球1个白球，第二箱装有3个黑球2个白球，现任取一箱，再从该箱中任取一球为白球，试求：取出的球是第一个箱子的概率。

案例转化：该箱子为第一个箱子的事件记为事件 B_1 取出来的球为白球的事件记为事件A，相当于计算 $P(B_1|A)$



$$P(B_1|A) = \frac{P(AB_1)}{P(A)} = \frac{P(A|B_1)P(B_1)}{P(A)}$$

$$\Rightarrow P(B_1|A) = \frac{\frac{1}{2} \times \frac{1}{5}}{\frac{3}{10}} = \frac{1}{3}$$

贝叶斯公式

在已知一些条件下（部分事件发生），实现对目标事件发生概率更准确的预测，公式为：

$$P(B|A) = P(B) * \frac{P(A|B)}{P(A)}$$

公式的延伸：

$$P(B_i|A) = \frac{P(B_i) * P(A|B_i)}{P(A)} = \frac{P(B_i) * P(A|B_i)}{\sum_{j=1}^n P(A|B_j) P(B_j)}$$

核心：基于事件先验概率，及可能性函数（事件发生的约束条件），得到特定情况下事件发生的概率
(后验概率)

现实问题思考

$$P(B|A)$$

猫对你叫，猫喜欢你的概率是多少？

已知：猫喜欢一个人的概率是0.1，它对喜欢的人叫的概率是0.4，它平时叫的概率是0.2

$$P(B) = 0.1$$

$$P(A|B) = 0.4$$

$$P(A) = 0.2$$

$$P(B|A) = \frac{P(B) \cdot P(A|B)}{P(A)} = \frac{0.1 \times 0.4}{0.2} = \boxed{0.2}$$

朴素贝叶斯用于机器学习

基于训练数据集 (X, Y) 与贝叶斯概率公式 机器学习从输入到输出的 概率分布，计算求出使得后验概率最大的类别作为预测输出。

x	y
1	1
0	0
1	0
1	1
0	0

$$P(Y_i|X) = P(Y_i) * \frac{P(X|Y_i)}{P(X)}$$

$$P(y=1|x=1) = P(y=1) * \frac{P(x=1|y=1)}{P(x=1)} = \frac{2}{5} * \frac{1}{3}$$

$$P(y=0|x=1) = \frac{3}{5} * \frac{1}{3} = \frac{1}{5}$$

$$\Rightarrow y_{\text{best}} = 1$$

假设新的输入 $x = 1$ ，预测其对应 y 的类别

朴素贝叶斯用于机器学习

$$P(Y|X) = P(Y) * \frac{P(X|Y)}{P(X)}$$

现实案例的输入特征高于1维，假设特征之间相互独立：

朴素贝叶斯公式：

$$P(X|Y = y_i) = \prod_{j=1}^m P(x_j|Y = y_i)$$

$$P(y_i|x_1, x_2 \dots, x_m) = \frac{P(y_i) \prod_{j=1}^m P(x_j|y_i)}{P(x_1, x_2 \dots, x_m)} = \frac{P(y_i) \prod_{j=1}^m P(x_j|y_i)}{\prod_{j=1}^m P(x_j)}$$

现实问题思考

$$P(y_i|x_1, x_2 \dots, x_m) = \frac{P(y_i) \prod_{j=1}^m P(x_j|y_i)}{P(x_1, x_2 \dots, x_m)} = \frac{P(y_i) \prod_{j=1}^m P(x_j|y_i)}{\prod_{j=1}^m P(x_j)}$$

$y=0$; $P(y=0 | x_1=0, x_2=1, x_3=0) = \frac{\frac{1}{2} \times \left(\frac{1}{2} \times 0 \times 1\right)}{\frac{1}{2} \times \frac{2}{4} \times \frac{3}{4}}$

学历	能力	经验	胜任与否 (y)
0	1	0	1
1	2	0	0
0	0	0	0
1	1	1	1

$y=1$:

$$P(y=1 | x_1=0, x_2=1, x_3=0)$$

$$= \frac{\frac{1}{2} \times \left(\frac{1}{2} \times 1 \times \frac{1}{2}\right)}{\frac{1}{2} \times \frac{2}{4} \times \frac{3}{4}}$$

$$= \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{3}{8}} = \frac{2}{3}$$

$y=1$

F:

$$\frac{1}{2} \times \frac{2}{4} \times \frac{3}{4}$$

小结

条件概率公式:

$$P(B|A) = \frac{P(AB)}{P(A)}$$

全概率公式:

$$P(A) = \sum_{i=1}^n P(A|B_i)P(B_i)$$

贝叶斯公式:

$$P(B|A) = P(B) * \frac{P(A|B)}{P(A)}$$

概率是反映随机事件出现的可能性大小的量度,而条件概率则是给定某事件A的条件下,另一事件B发生的概率。全概率公式则是利用条件概率将复杂事件A分割为若干简单事件概率的求和问题。贝叶斯公式则是利用条件概率和全概率公式计算后验概率。

知识巩固

问题：计算第一行输入预测 $y=0$ 的概率，并思考结果与 $y=1$ 之和不为1的原因。

学历	能力	经验	胜任与否 (y)
0	1	0	1
1	2	0	0
0	0	0	0
1	1	1	1



Python3人工智能入门+实战提升：机器学习

Chapter 4 其他常用分类技术

赵辛

Chapter 4 其他常用分类技术

-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

任务一：决策树判断员工是否适合相关工作

基于课程中决策树案例与task1_data数据，基于信息熵原理建立决策树模型。

Skill	Experience	Degree	Income	y
2	0	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
1	0	1	1	1
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	1

属性数值意义：

Skill技能

0: 不达标; 1: 达标; 2: 超强

Experience经验

0: 无相关经验; 1: 有相关经验

Degree学位

0: 不符合; 1: 符合

Income收入期望

0: 未达到期望; 1: 达到期望

Y结果

0: 不适合该工作; 1: 适合该工作

任务一：决策树判断员工是否适合相关工作

基于课程中决策树案例与task1_data数据，基于信息熵原理建立决策树模型。

Skill	Experience	Degree	Income	y
2	0	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
1	0	1	1	1
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	1

- 1、建立决策树模型、计算准确率
 - 2、预测申请者skill=1,experience=0,degree=1,income=1是否适合该工作；
 - 3、可视化模型结构
 - 4、修改min_samples_leaf参数，对比模型结果
- 能力拓展：基于ID3原理计算每个节点信息熵增益，画出决策树结构，与实战模型结构对比

决策树实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

```
#数据加载  
import pandas as pd  
import numpy as np  
data = pd.read_csv(job_data.csv')  
data.head(10)
```

	Skill	Experience	Degree	Income	y
0	2	0	1	0	0
1	0	0	1	0	0
2	0	1	1	0	0
3	0	0	1	0	0
4	0	1	1	0	0
5	0	0	1	0	0
6	1	0	1	1	1
7	0	1	0	0	0
8	0	0	1	0	0
9	1	0	1	0	0

决策树实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

```
#x y赋值  
X = data.drop(['y'],axis=1)  
y = data.loc[:, 'y']  
print(X.shape,y.shape)
```

(500, 4) (500,)

决策树实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

```
#创建模型实例  
from sklearn import tree  
dc_tree =  
tree.DecisionTreeClassifier(criterion='entropy',min_samples_leaf=5)  
dc_tree.fit(X,y)
```

criterion='entropy'：以信息熵的变化作为建立树结构的标准

min_samples_leaf=5：建立树结构最小分支的样本数

决策树实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

```
#新数据预测  
X_test = np.array([[1,0,1,1]])  
y_test = dc_tree.predict(X_test)  
print('适合' if y_test==1 else '不适合')
```

适合

```
#预测准确率  
y_predict = dc_tree.predict(X)  
from sklearn.metrics import accuracy_score  
accuracy = accuracy_score(y,y_predict)  
print(accuracy)
```

0.85

决策树实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

#可视化决策树结构

%matplotlib inline

from matplotlib import pyplot as plt

fig = plt.figure(figsize=(200,200))

tree.plot_tree(dc_tree,filled='True',

feature_names=['Skill', 'Experience',

'Degree', 'Income'],

class_names=['Un-qualified','Qualified'])

entropy = 0.722
samples = 5
value = [4, 1]
class = Un-qualified

entropy = 0.98
samples = 12
value = [7, 5]
class = Un-qualified

entropy = 0.961
samples = 13
value = [5, 8]
class = Qualified

决策树实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

#设置字体

```
import matplotlib as mpl  
mpl.rcParams['font.family'] = 'SimHei'
```

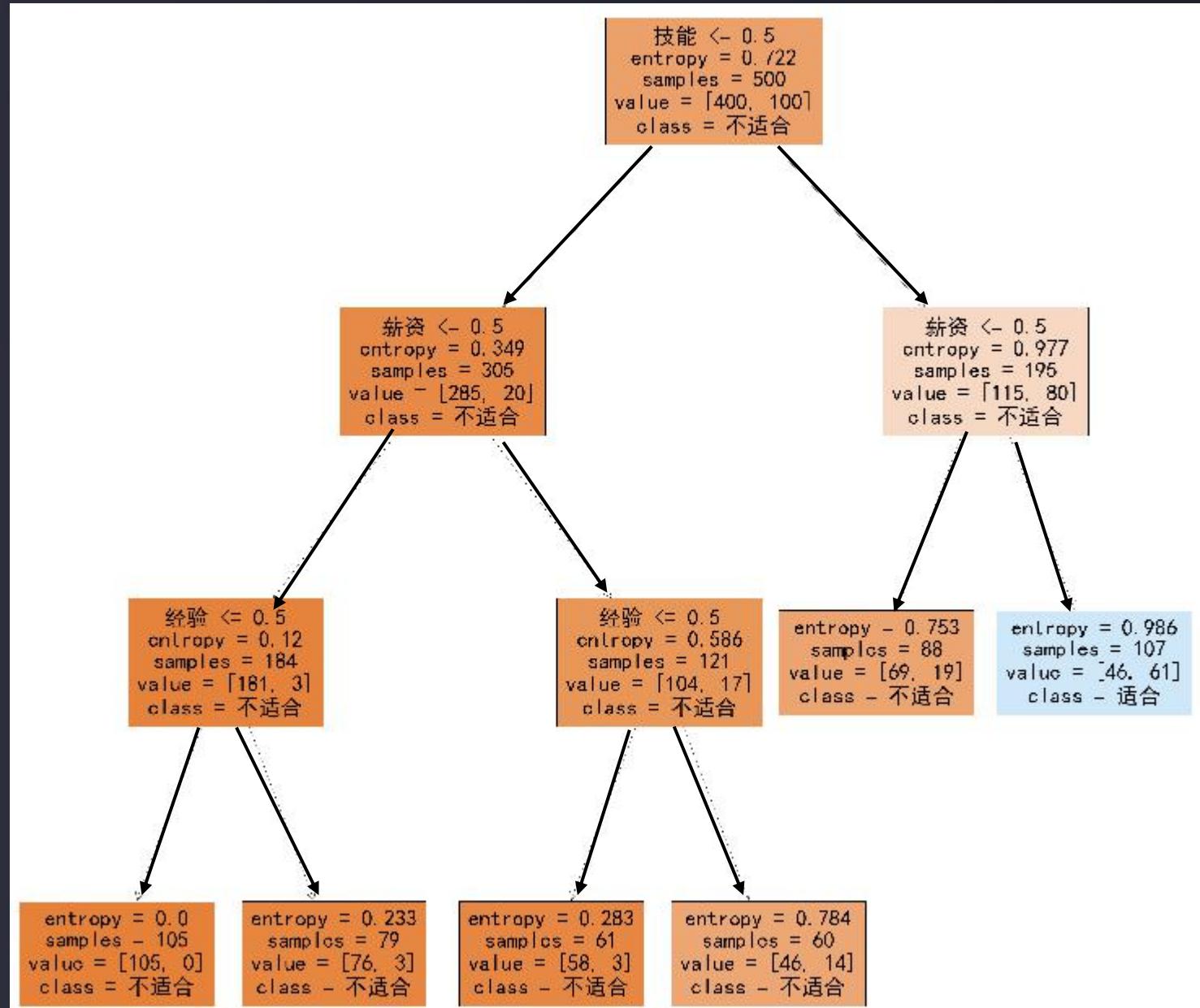
#可视化决策树结构

```
fig = plt.figure(figsize=(80,80))  
tree.plot_tree(dc_tree,filled='True',feature_names=['技能',  
'经验', '学历', '薪资'],class_names=['不适合','适合'])
```

#模型结果存储

```
fig.savefig('test.png')
```

结果展示



任务二：朴素贝叶斯预测学生录取及奖学金情况

基于task2_data数据，建立朴素贝叶斯模型预测学生申请结果。

成绩	学校	获奖	性别	英语	y
0	1	0	1	1	0
1	0	0	0	0	0
0	1	1	0	1	1
1	1	0	1	0	1
0	0	0	1	0	0
2	1	0	0	0	1
1	1	1	0	1	0
2	1	0	1	1	0
0	1	0	1	0	0

属性数值意义：

成绩

0: 不及格; 1: 及格; 2: 优秀

学校

0: 普通; 1: 重点

获奖

0: 无; 1: 有

性别

0: 女; 1: 男

英语

0: 普通; 1: 优异

y结果

0: 未录取; 1: 录取; 2: 带奖学金录取

任务二：朴素贝叶斯预测学生录取及奖学金情况

基于task2_data数据，建立朴素贝叶斯模型预测学生申请结果。

成绩	学校	获奖	性别	英语	y
0	1	0	1	1	0
1	0	0	0	0	0
0	1	1	0	1	1
1	1	0	1	0	1
0	0	0	1	0	0
2	1	0	0	0	1
1	1	1	0	1	0
2	1	0	1	1	0
0	1	0	1	0	0

- 1、计算模型对训练数据各样本预测各类别的概率及输出类别结果、计算模型准确率；
- 2、观察测试样本数据并主观预测每个样本的结果，然后结合模型计算对应类别概率、与结果，将两个结果进行对比
- 3、将测试样本数据、预测概率、结果以csv格式存储到本地

学员信息 (测试样本)				
成绩	学校	获奖	性别	英语
2	1	1	1	1
2	1	1	1	0
2	1	1	0	0
2	1	0	0	0
2	0	0	0	0

朴素贝叶斯实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

#创建模型实例

```
from sklearn.naive_bayes import  
CategoricalNB
```

```
model = CategoricalNB()  
model.fit(X, y)
```

参考链接: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html

朴素贝叶斯实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

#概率预测

```
y_predict_prob = model.predict_proba(X)  
print(y_predict_prob)
```

```
[[0.72279958 0.21735277 0.05984765]  
 [0.79655931 0.16840549 0.0350352 ]  
 [0.37146649 0.40763466 0.22089885]  
 [0.70511475 0.23857787 0.05630738]  
 [0.88207671 0.09877993 0.01914336]  
 [0.38753488 0.44680162 0.1656635 ]
```

朴素贝叶斯实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

#类别预测

```
y_predict = model.predict(X)  
print(y_predict)
```

朴素贝叶斯实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

```
#测试数据集预测  
X_test =  
np.array([[2,1,1,1,1],[2,1,1,1,0],[2,1,1,0,0],[2,  
1,0,0,0],[2,0,0,0,0]])  
y_test_proba = model.predict_proba(X_test)  
print(y_test_proba)
```

学员信息 (测试样本)					预测概率			预测结果
成绩	学校	获奖	性别	英语	未录取	无奖学金录取	带奖学金录取	
2	1	1	1	1	0.152	0.346	0.502	带奖学金录取
2	1	1	1	0	0.203	0.400	0.397	无奖学金录取
2	1	1	0	0	0.158	0.455	0.387	无奖学金录取
2	1	0	0	0	0.388	0.447	0.166	无奖学金录取
2	0	0	0	0	0.595	0.293	0.112	未录取

结果展示

学员信息 (测试样本)					预测概率			预测结果
成绩	学校	获奖	性别	英语	未录取	无奖学金录取	带奖学金录取	
2	1	1	1	1	0.152	0.346	0.502	带奖学金录取
2	1	1	1	0	0.203	0.400	0.397	无奖学金录取
2	1	1	0	0	0.158	0.455	0.387	无奖学金录取
2	1	0	0	0	0.388	0.447	0.166	无奖学金录取
2	0	0	0	0	0.595	0.293	0.112	未录取

结果输出

```
#结果组合输出  
test_data_result = np.concatenate((X_test, y_test_proba,np.array(y_test).reshape(5,1)), axis=1)  
test_data_result = pd.DataFrame(test_data_result)  
test_data_result.head()
```

	0	1	2	3	4	5	6	7	8
0	2.0	1.0	1.0	1.0	1.0	0.152466	0.345591	0.501943	2.0
1	2.0	1.0	1.0	1.0	0.0	0.202700	0.399898	0.397402	1.0
2	2.0	1.0	1.0	0.0	0.0	0.157867	0.454823	0.387310	1.0
3	2.0	1.0	0.0	0.0	0.0	0.387535	0.446802	0.165663	1.0
4	2.0	0.0	0.0	0.0	0.0	0.594723	0.293380	0.111897	0.0

```
#修改各列名称  
test_data_result.columns=['score','school','award','gender','english','p0','p1','p2','y_predict' ]  
#数据存储到本地  
test_data_result.to_csv('test_data_result.csv')
```



Python3人工智能入门+实战提升：机器学习

Chapter 4 其他常用分类技术

赵辛

Chapter 4 其他常用分类技术

-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

任务一：决策树判断员工是否适合相关工作

基于课程中决策树案例与task1_data数据，基于信息熵原理建立决策树模型。

Skill	Experience	Degree	Income	y
2	0	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
1	0	1	1	1
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	1

属性数值意义：

Skill技能

0: 不达标; 1: 达标; 2: 超强

Experience经验

0: 无相关经验; 1: 有相关经验

Degree学位

0: 不符合; 1: 符合

Income收入期望

0: 未达到期望; 1: 达到期望

Y结果

0: 不适合该工作; 1: 适合该工作

任务一：决策树判断员工是否适合相关工作

基于课程中决策树案例与task1_data数据，基于信息熵原理建立决策树模型。

Skill	Experience	Degree	Income	y
2	0	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
1	0	1	1	1
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	1

- 1、建立决策树模型、计算准确率
 - 2、预测申请者skill=1,experience=0,degree=1,income=1是否适合该工作；
 - 3、可视化模型结构
 - 4、修改min_samples_leaf参数，对比模型结果
- 能力拓展：基于ID3原理计算每个节点信息熵增益，画出决策树结构，与实战模型结构对比



Python3人工智能入门+实战提升：机器学习

Chapter 4 其他常用分类技术

赵辛

Chapter 4 其他常用分类技术

-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

任务二：朴素贝叶斯预测学生录取及奖学金情况

基于task2_data数据，建立朴素贝叶斯模型预测学生申请结果。

成绩	学校	获奖	性别	英语	y
0	1	0	1	1	0
1	0	0	0	0	0
0	1	1	0	1	1
1	1	0	1	0	1
0	0	0	1	0	0
2	1	0	0	0	1
1	1	1	0	1	0
2	1	0	1	1	0
0	1	0	1	0	0

属性数值意义：

成绩

0: 不及格; 1: 及格; 2: 优秀

学校

0: 普通; 1: 重点

获奖

0: 无; 1: 有

性别

0: 女; 1: 男

英语

0: 普通; 1: 优异

y结果

0: 未录取; 1: 录取; 2: 带奖学金录取

任务二：朴素贝叶斯预测学生录取及奖学金情况

基于task2_data数据，建立朴素贝叶斯模型预测学生申请结果。

成绩	学校	获奖	性别	英语	y
0	1	0	1	1	0
1	0	0	0	0	0
0	1	1	0	1	1
1	1	0	1	0	1
0	0	0	1	0	0
2	1	0	0	0	1
1	1	1	0	1	0
2	1	0	1	1	0
0	1	0	1	0	0

- 1、计算模型对训练数据各样本预测各类别的概率及输出类别结果、计算模型准确率；
- 2、观察测试样本数据并主观预测每个样本的结果，然后结合模型计算对应类别概率、与结果，将两个结果进行对比
- 3、将测试样本数据、预测概率、结果以csv格式存储到本地

学员信息 (测试样本)				
成绩	学校	获奖	性别	英语
2	1	1	1	1
2	1	1	1	0
2	1	1	0	0
2	1	0	0	0
2	0	0	0	0



Python3人工智能入门+实战提升：机器学习

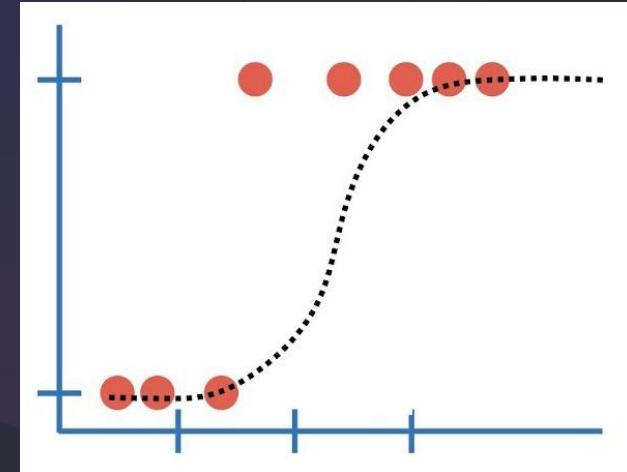
Chapter 4 其他常用分类技术

赵辛

Chapter 4 其他常用分类技术

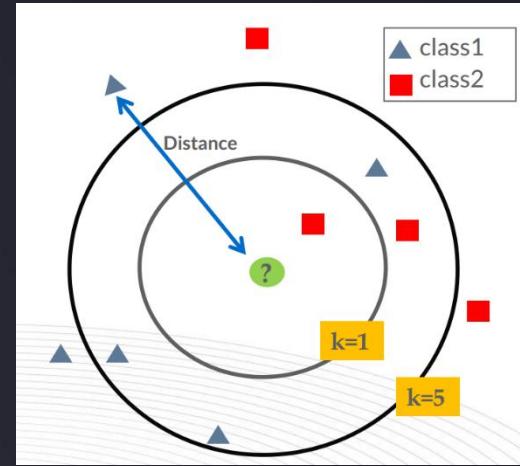
-
- 1 --K近邻分类 (KNN)
 - 2 --决策树
 - 3 --朴素贝叶斯 (一)
 - 4 --朴素贝叶斯 (二)
 - 5 --实战准备
 - 6 --实战 (一) 决策树判断员工是否适合相关工作
 - 7 --实战 (二) 朴素贝叶斯预测学生录取及奖学金情况
 - 8 --综合能力体现：技术对比与总结

| 常用分类方法

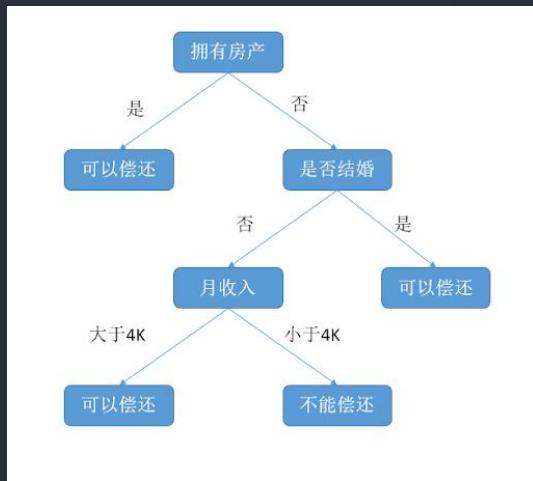


逻辑回归(logistics regression)

没有最好的分类器，只有最合适的选择器



KNN近邻模型(K-nearest neighbors)



决策树(decision tree)

朴素贝叶斯

$$P(Y|X) = P(Y) * \frac{P(X|Y)}{P(X)}$$

$$P(y_i|x_1, x_2, \dots, x_n) = \frac{P(y_i) \prod_{j=1}^n P(x_j|y_i)}{\prod_{j=1}^n P(x_j)}$$

逻辑回归

根据数据特征，计算样本归属于某一类别的概率 $P(x)$ ，根据概率数值判断其所属类别。

核心：基于逻辑回归方程，计算类别概率

优点：

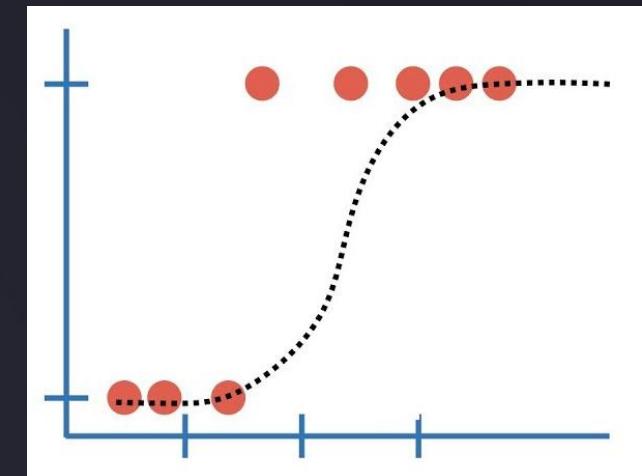
- 输出值自然地落在0到1之间，并且有概率意义
- 参数代表每个特征对输出的影响，可解释性强
- 实施简单，非常高效（计算量小、存储占用低），可以在大数据场景中使用

缺点：

- 本质上是一个线性的分类器，对于特征相关度高的情况效果不是很好
- 特征空间很大时，性能不好

适用场景：

需要较为清晰地理解每个属性对结果的影响



逻辑回归(logistics regression)

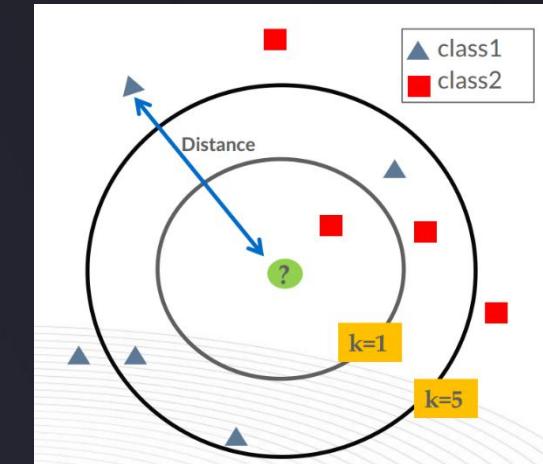
KNN

通过计算新数据与训练数据之间的距离，然后选取K ($K \geq 1$) 个距离最近的邻居进行分类判断 (K个邻居) ， 这K个邻居的多数属于某个类，就把该新数据实例分类到这个类中。

类比：物以类聚，人以群分

优点：

- 简单、易于理解、易于实现、无需估计参数、无需求解训练。
- 适合对稀有事件进行分类；
- 在多分类场景中也不会增加训练复杂度，特别适合多分类问题



KNN近邻模型(K-nearest neighbors)

缺点：

- 懒惰算法，对测试样本分类时的计算量大，需要扫描所有训练样本，内存开销大，评分慢
- 完全跟着数据走，没有数学模型可言，无法检查不同属性对结果的影响
- 当样本不平衡时，如一个类的样本容量很大，而其他类样本容量很小时，有可能导致当输入一个新样本时，该样本的K个邻居中大容量类的样本占多数。

适用场景：

需要一个特别容易解释的模型的时候，比如需要向用户解释原因的推荐算法

决策树

一种对实例进行**分类的树形结构**，通过**多层判断**区分目标所属类别

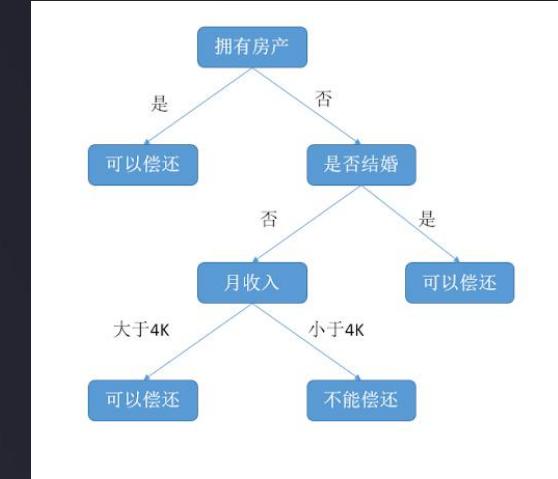
本质：通过多层判断，从训练数据集中归纳出一组分类规则

优点：

- 计算量小，运算速度快
- 判断步骤清晰，易于理解

缺点：

- 忽略属性间的相关性
- 样本类别分布不均匀时，容易影响模型表现



决策树(decision tree)

适用场景：

需要清晰地描述类别判断的前后逻辑（先依据哪个指标判断，接下来使用哪个指标）

朴素贝叶斯

基于训练数据集 (X, Y) 与贝叶斯概率公式，机器学习从输入到输出的概率分布，计算求出使得后验概率最大的类别作为预测输出。

朴素贝叶斯

假设：各属性之间相互独立

优点：

➤ 源于古典数学理论，分类逻辑清晰

➤ 可清晰查看各个类别对应概率，观察数据改变的概率变化，
帮助理解预测过程

$$P(Y|X) = P(Y) * \frac{P(X|Y)}{P(X)}$$

$$P(y_i|x_1, x_2 \dots, x_n) = \frac{P(y_i) \prod_{j=1}^n P(x_j|y_i)}{\prod_{j=1}^n P(x_j)}$$

缺点：

➤ 在属性个数比较多或者属性之间相关性较大时，分类效果不好
➤ 对先验概率依赖度高，样本类别分布不均匀时，容易影响模型表现

适用场景：

数据不同维度之间相关性较小

小结

不同模型之间没有绝对地优劣，根据场合选择合适的

思考应用场景，尝试并对比不同的模型表现

掌握其他的模型与方法，比如：SVM、神经网络、集成方法等

参考资料：

https://blog.csdn.net/dfly_zx/article/details/108126034

或者扫描二维码，从学习资料中获取。





Python3人工智能入门+实战提升：机器学习

Chapter 5 无监督学习与聚类分析

赵辛

Chapter 5 无监督学习与聚类分析

-
- 1 --无监督学习 (Unsupervised Learning)
 - 2 --k均值聚类算法(Kmeans)
 - 3 --实战准备
 - 4 --实战 (一) KMeans实现数据聚类
 - 5 --实战 (二) KMeans实现图像分割
 - 6 --现实问题思考：监督真的重要吗

|现实问题思考

目标：

以下六组图片，按照自己喜爱的方式分成两组



|现实问题思考

方式一：站着或非站着



|现实问题思考

方式二：白色或黄色



|现实问题思考

方式三：吐舌头或不吐舌头



|现实问题思考



分组一：站着或非站着

分组二：白色或黄色

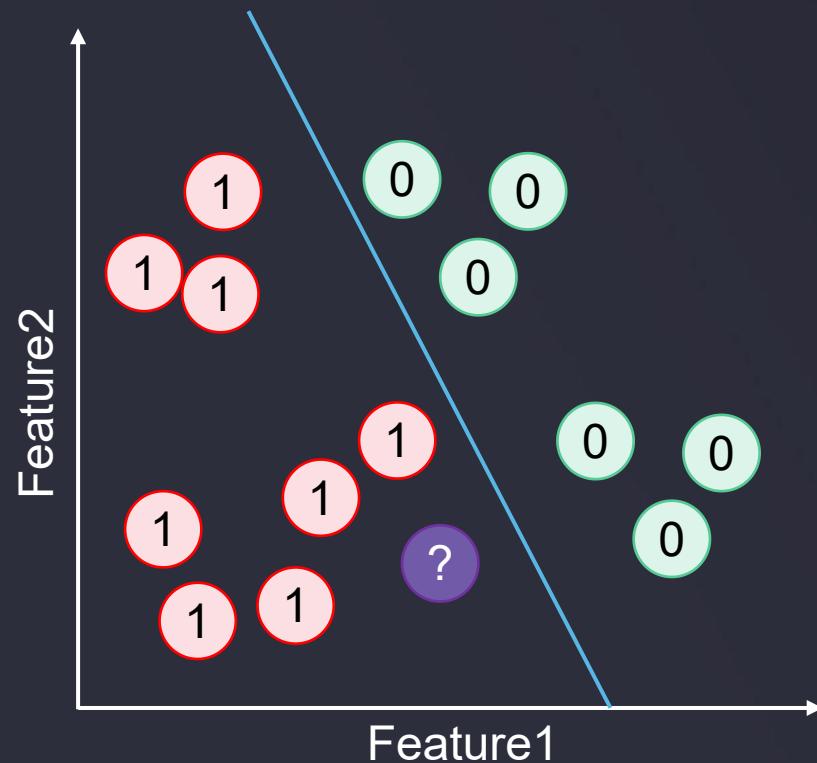
分组三：吐舌头或不吐舌头

- 没有绝对的对错标准
- 寻找数据特征的相似性

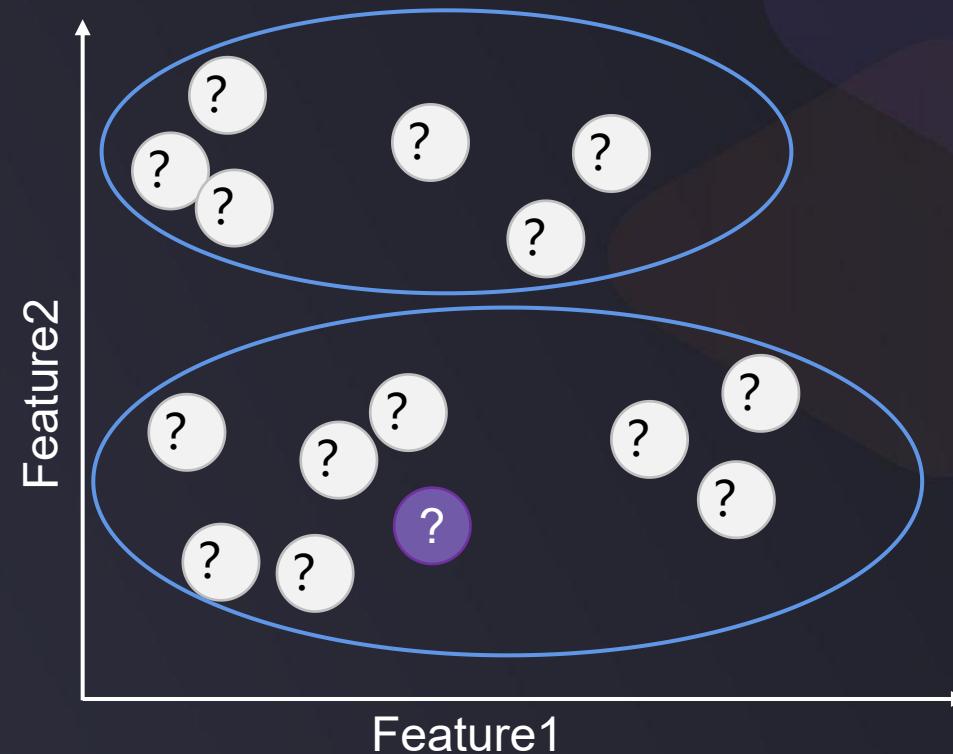
无监督学习

无监督学习 (Unsupervised Learning)

定义：机器学习的一种方法，训练数据中不带标签，让机器自动寻找数据规律并完成任务。

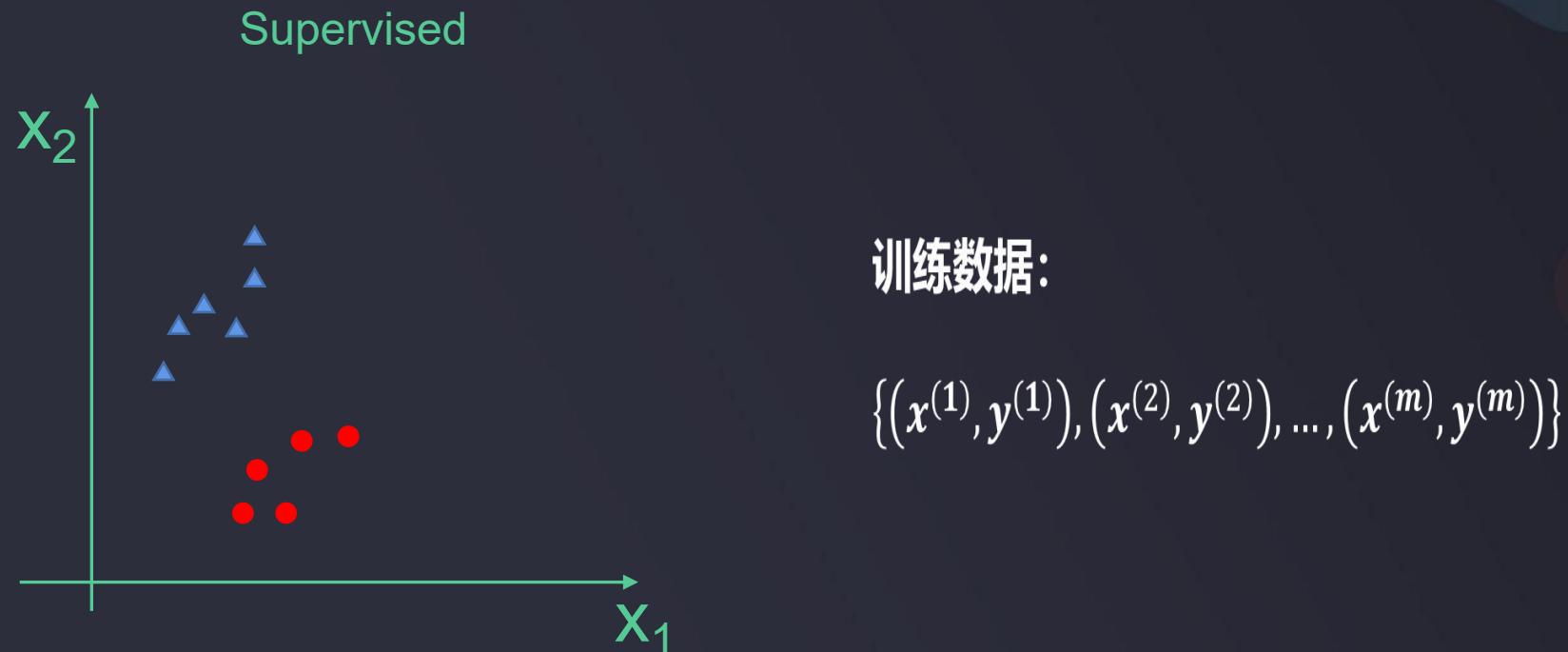


监督学习 包括正确的结果

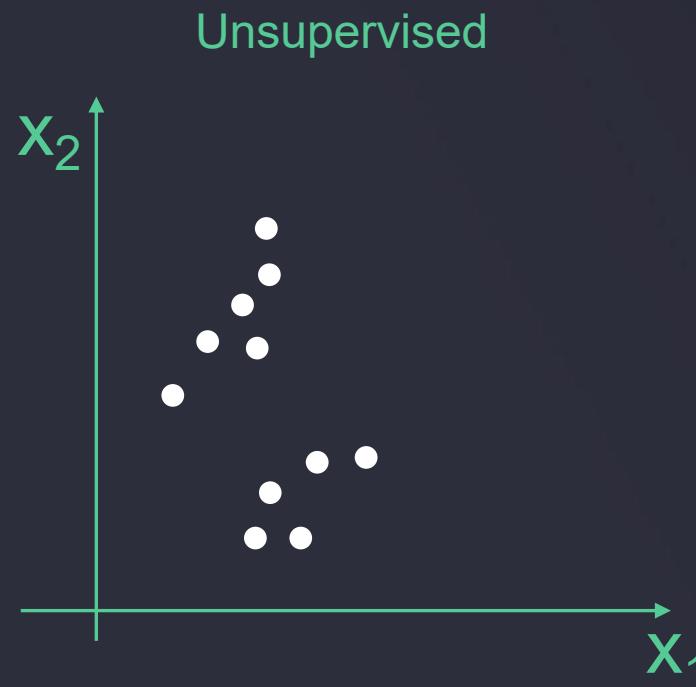


无监督学习 不包括正确的结果

监督学习 (Supervised Learning)



无监督学习 (Unsupervised Learning)



训练数据:

$$\{(\boldsymbol{x}^{(1)}), (\boldsymbol{x}^{(2)}), \dots, (\boldsymbol{x}^{(m)})\}$$

没有标签 y !

无监督学习 (Unsupervised Learning)

特点:

- 数据不需要标签
- 算法不受监督信息（偏见）约束

优点:

- 降低数据采集难度，极大程度扩充样本量
- 可能发现新的数据规律、被忽略的重要信息

主要运用:

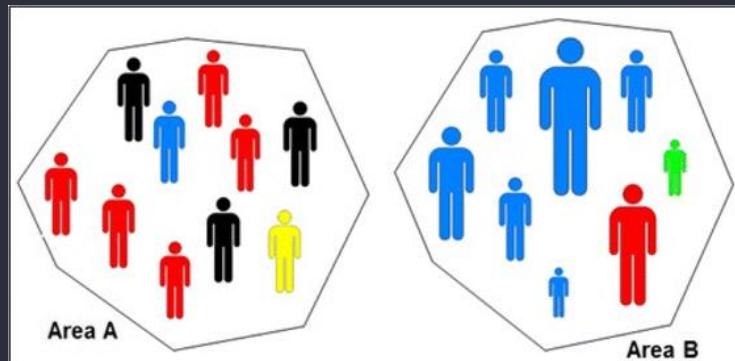
聚类分析

关联规则

维度缩减

|聚类分析 (Cluster analysis)

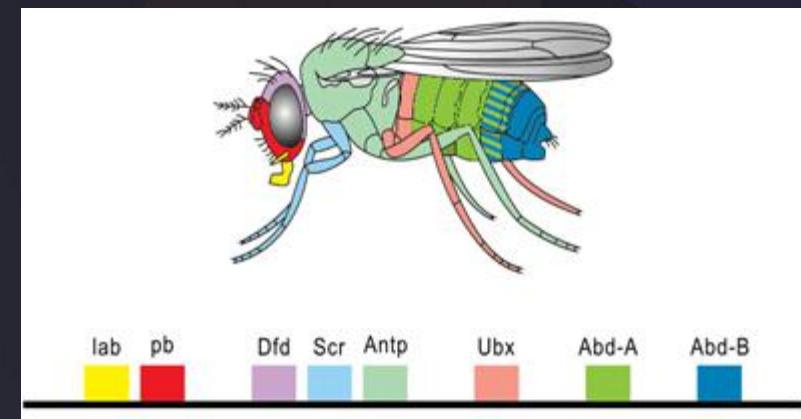
把数据样本按照一定方法分成不同的组别，这样让在同一个组别的成员对象都有相似的一些属性



目标用户的群体分类



图像切割



基因聚类

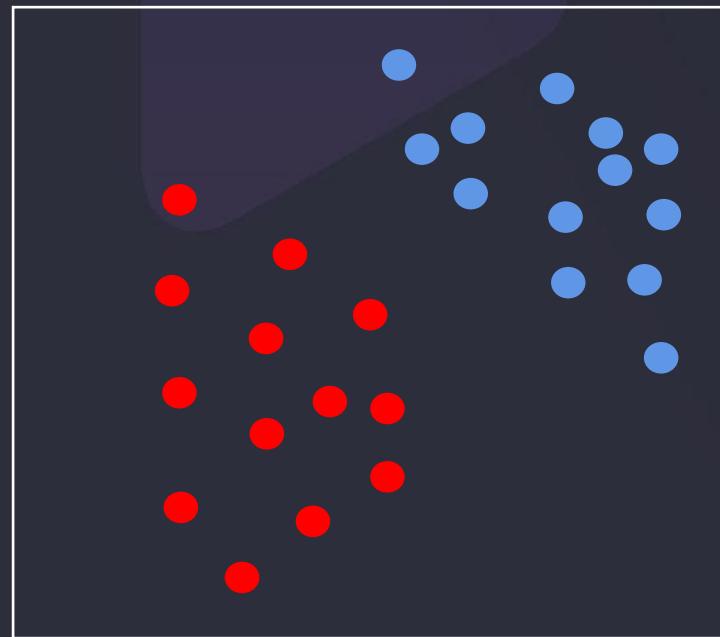
| 知识巩固

问题：无监督学习的一大特点是数据样本不要提前标注输出结果，思考这个特点带来的影响。

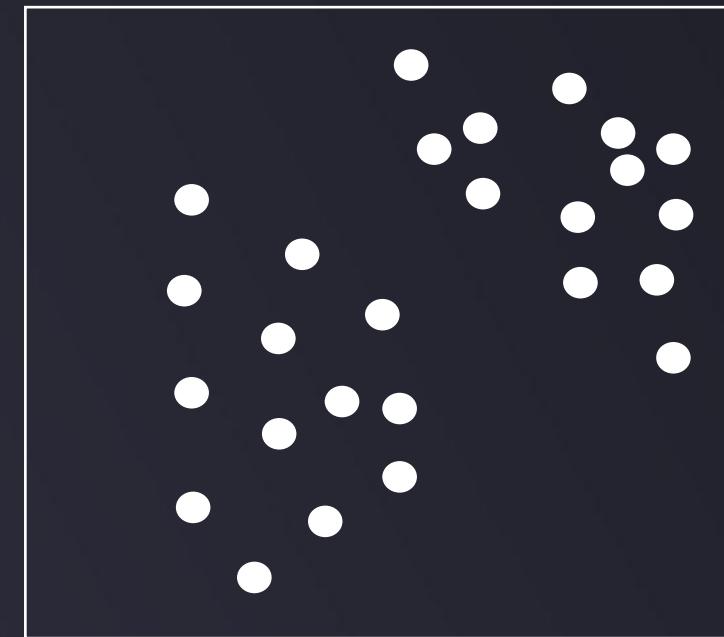
Chapter 5 无监督学习与聚类分析

-
- 1 --无监督学习 (Unsupervised Learning)
 - 2 --k均值聚类算法(Kmeans)
 - 3 --实战准备
 - 4 --实战 (一) KMeans实现数据聚类
 - 5 --实战 (二) KMeans实现图像分割
 - 6 --现实问题思考：监督真的重要吗

现实问题思考



带标签数据分类

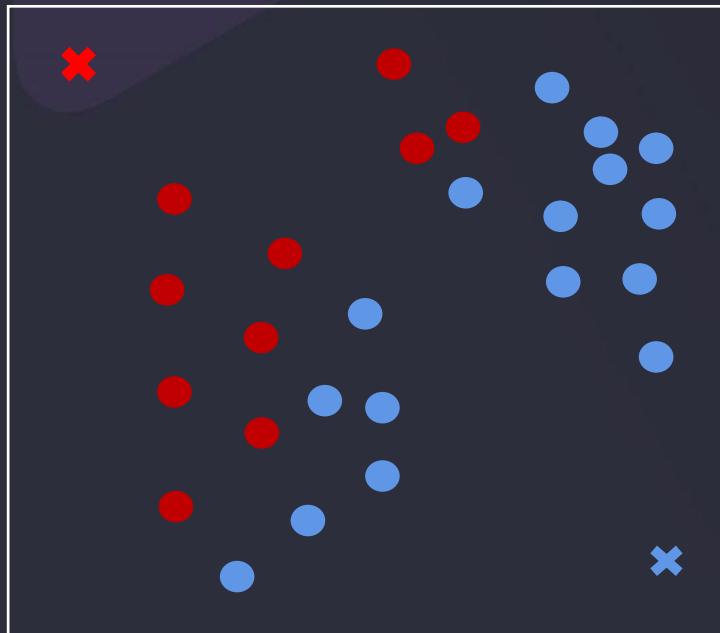


不带标签数据聚类

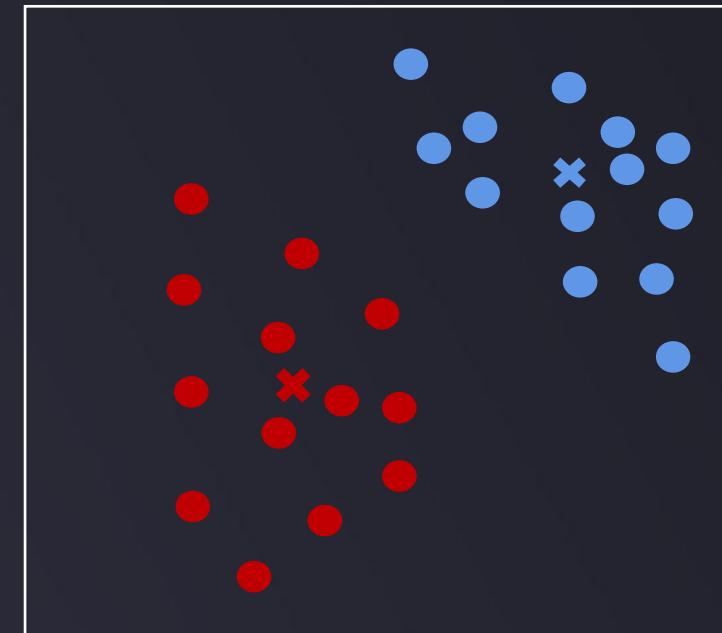
没有标签的情况下实现聚类的一个思路：给出中心点，根据数据到中心点距离判断类别

K均值聚类

在样本数据空间中选取K个点作为中心，计算每个样本到各中心的距离，根据距离确定数据类别，是聚类算法中最为基础但也最为重要的算法。



不带标签数据聚类



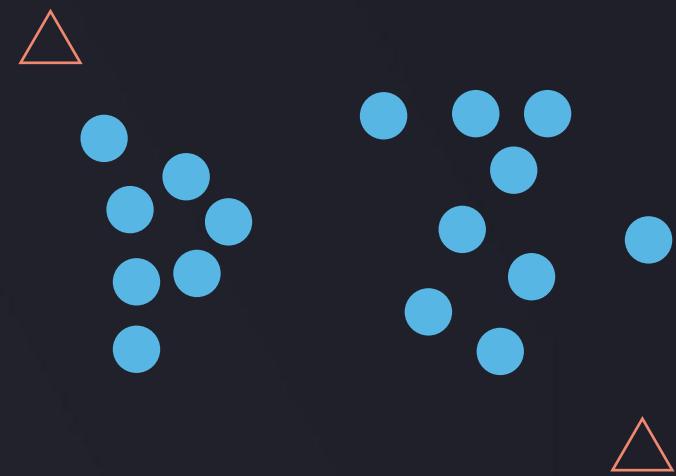
不带标签数据聚类

中心点会根据类别内样本数据分布进行更新

K均值聚类

核心流程：

- 1、基于要求、观察或经验确定聚类的个数k
- 2、确定k个中心
- 3、计算样本到各中心点距离
- 4、根据距离确定各个样本点所属类别
- 5、计算同类别样本的中心点，并将其设定为新的中心
- 6、重复步骤3-5直到收敛（中心点不再变化）



K均值聚类

核心流程：

- 1、基于观察或经验确定聚类的个数k
- 2、确定k个中心
- 3、计算样本到各中心点距离
- 4、根据距离确定各个样本点所属类别
- 5、计算同类别样本的中心点，并将其设定为新的中心
- 6、重复步骤3-5直到收敛（中心点不再变化）

核心公式：

$$u_j^t$$

数据点与各簇中心点距离： $dist(x_i, u_j^t)$

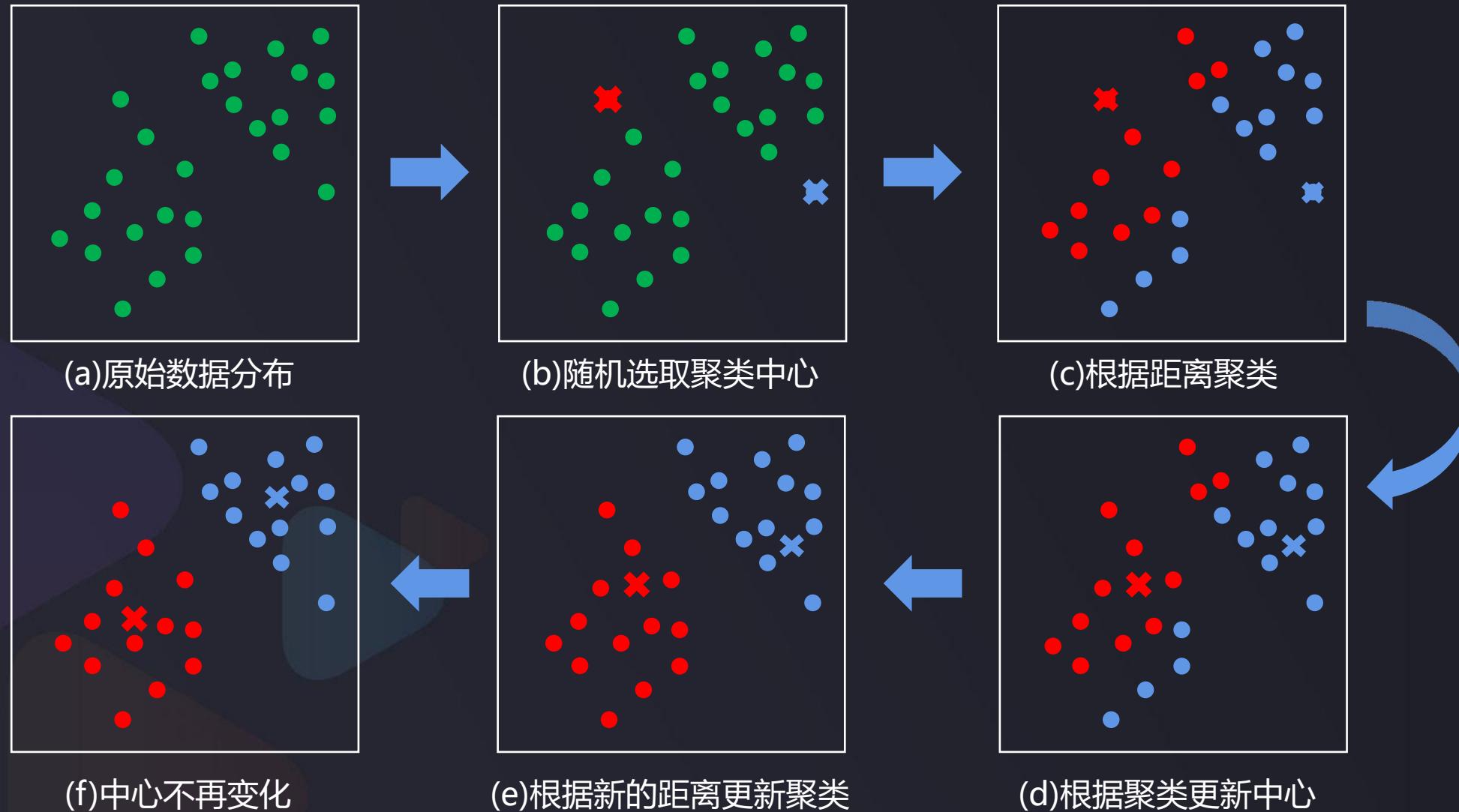
根据距离归类： $x_i \in u_{nearest}^t$

$$\text{中心更新: } u_j^{t+1} = \frac{1}{c} \sum_{x_i \in S_j} (x_i)$$

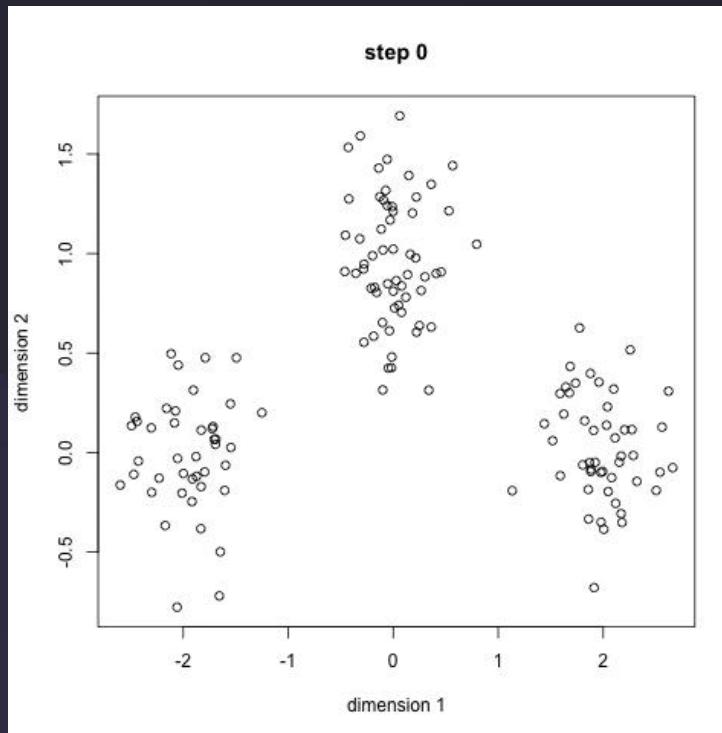
$$u_j^{t+1} = u_j^t$$

S_j :t时刻第j个区域簇； c:包含在 S_j 范围内点的个数； x_i :样本数据点； u_j^t 为t状态下第j区域中心

K均值聚类

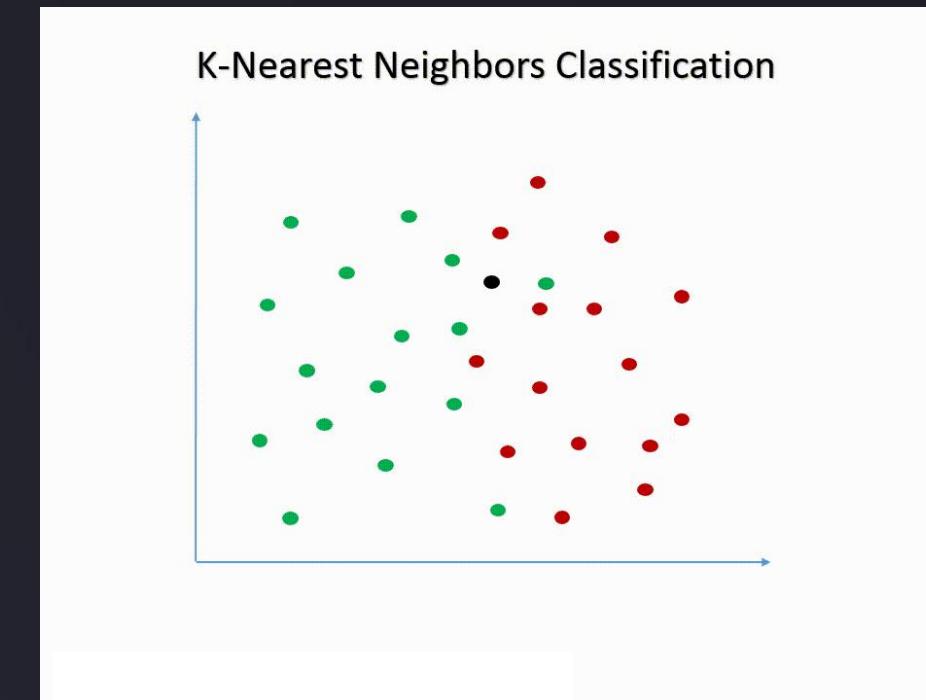


K均值聚类 (KMeans) VS K近邻分类 (KNN)



KMeans

- 无监督学习
- 聚类算法
- 无Label的数据集
- 计算数据与中心点距离



KNN

- 监督学习
- 分类算法
- 带Label的数据集
- 计算数据与其他数据的距离

K均值聚类实现图像分割



(1)原图



(2) $k=3$



(3) $k=8$

图像分割就是把图像分成若干个特定的、具有独特性质的区域的技术，是由图像处理到图像分析的关键步骤。

最基础的实现方法：灰度阈值分割

$$g(i,j) = \begin{cases} 1, & \text{if } f(i,j) \geq T \\ 0, & \text{if } f(i,j) < T \end{cases}$$

T 为阈值

K均值聚类实现图像分割



(1)原图



(2) $k=3$



(3) $k=8$

不得不面对的问题：

阈值如何确定？需要确定几个阈值？

只用阈值是否会遗漏其他重要信息？

K均值聚类帮你快速
实现图像分割

K均值聚类实现图像分割



(1)原图



```
[[[254 255 250]
 [255 255 251]
 [246 247 241]
 ...
 [255 255 246]
 [255 255 250]
 [254 251 246]]]
```

(140, 140, 3)



```
[[254 255 250]
 [255 255 251]
 [246 247 241]
 ...
 [240 246 246]
 [246 252 248]
 [250 255 250]]]
```

(19600, 3)



(2)k=3



(3)k=8

```
[[1 1 1 ... 1 1 1]
 [1 1 2 ... 2 1 1]
 [1 2 0 ... 0 2 1]
 ...
 [1 2 0 ... 0 2 1]
 [1 1 2 ... 2 2 1]
 [1 1 1 ... 1 1 1]]]
```

(140, 140)

```
[[1]
 [1]
 [1]
 ...
 [1]
 [1]
 [1]]]
```

(19600, 1)

| 知识巩固

思考：三张分割图哪张k值最大，哪张最小？观察并思考其差异及原因。





Python3人工智能入门+实战提升：机器学习

Chapter 5 无监督学习与聚类分析

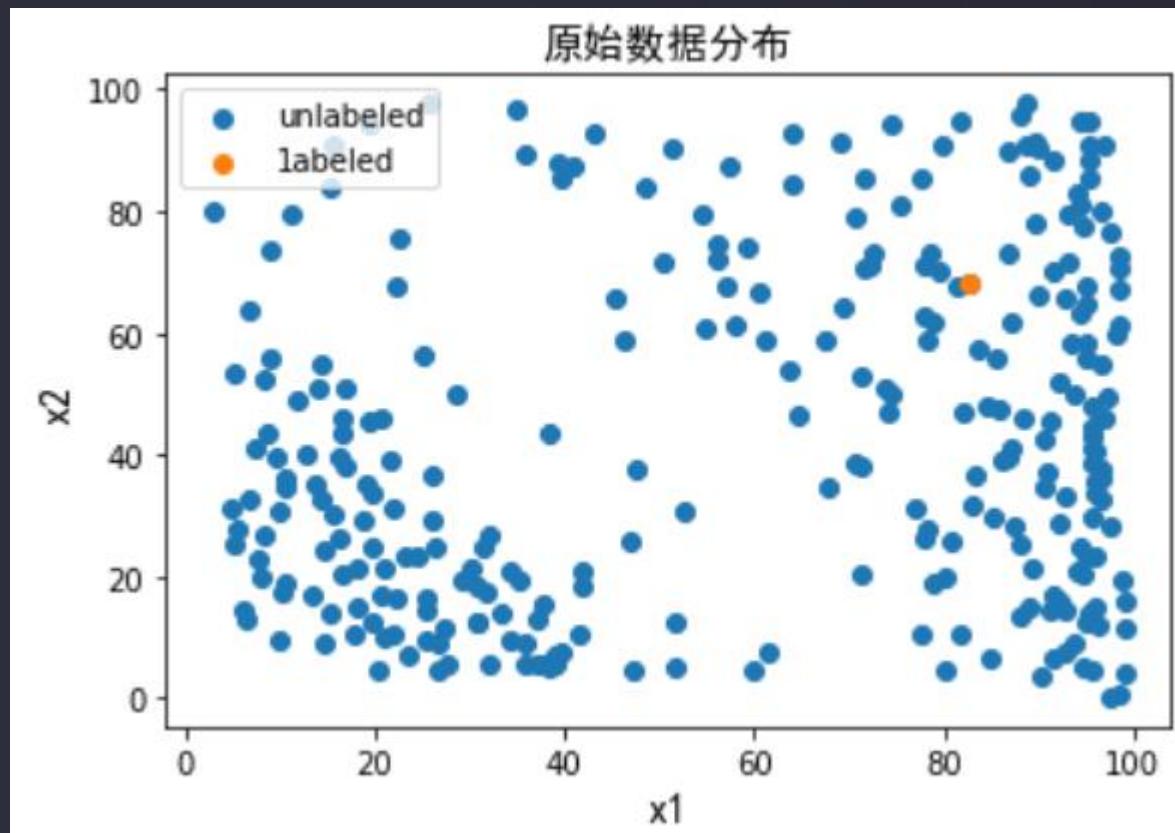
赵辛

Chapter 5 无监督学习与聚类分析

-
- 1 --无监督学习 (Unsupervised Learning)
 - 2 --k均值聚类算法(Kmeans)
 - 3 --实战准备
 - 4 --实战 (一) KMeans实现数据聚类
 - 5 --实战 (二) KMeans实现图像分割
 - 6 --现实问题思考：监督真的重要吗

任务一：KMeans实现数据聚类

基于task1_data1数据，建立Kmeans模型，实现数据聚类。



- 1、 $K=2$ ，实现数据聚类，可视化聚类结果、聚类中心；
- 2、已知第一个样本点 $X_1=82.5, X_2=67.9$ 属于类别0，对聚类结果进行矫正；
- 3、基于task1_data2建立KNN模型，思考其与聚类结果的差异
- 4、修改Kmeans迭代次数与初始化参数，查看模型迭代过程中的结果变化

task1_data1：包含了无标签数据及一个带有标签的样本点；

task1_data2：包含了正确类别标签结果的数据，可用于模型评估与监督学习

KMeans数据聚类实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#数据加载  
import pandas as pd  
import numpy as np  
data = pd.read_csv('task1_data1.csv')  
data_result = pd.read_csv('task1_data2.csv')  
data.head()
```

	x1	x2	y
0	82.5302	67.9939	0.0
1	14.3821	54.6641	NaN
2	88.9239	14.9664	NaN
3	78.0811	26.0769	NaN
4	78.1597	58.6068	NaN

KMeans数据聚类实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#获取唯一一个有标签的数据点  
X_labeled = data.iloc[0,:]  
print(X_labeled)  
print(type(X_labeled))
```

```
x1      82.5302  
x2      67.9939  
y       0.0000  
Name: 0, dtype: float64  
<class 'pandas.core.series.Series'>
```

```
#获取用于模型评估的正确结果  
y = data_result.loc[:, 'y']  
y.head()
```

```
0      0  
1      1  
2      0  
3      0  
4      0  
Name: y, dtype: int64
```

KMeans数据聚类实战流程

数据加载及展示

数据预处理

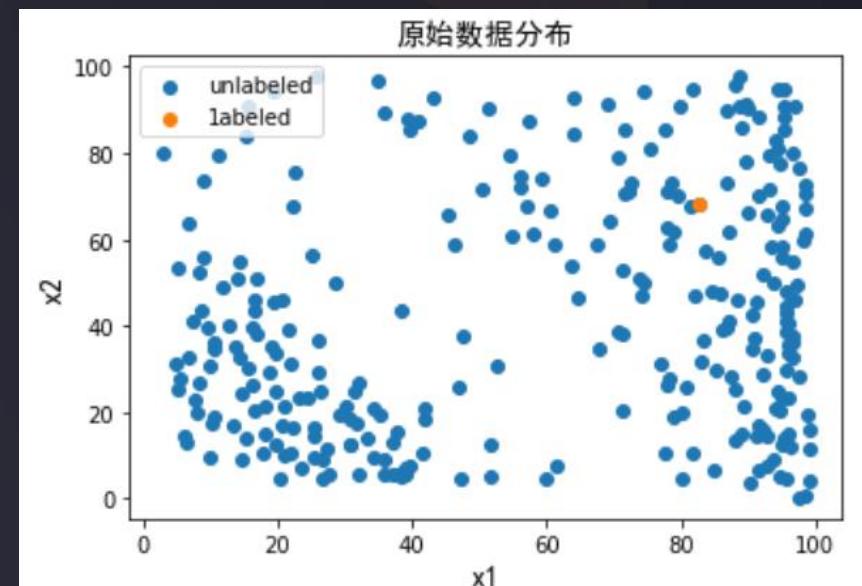
模型建立及训练

模型预测

结果展示及表现评估

#数据可视化

```
from matplotlib import pyplot as plt
import matplotlib as mlp
font2 = {'family':'SimHei','weight':'normal','size':14}
fig1 = plt.figure()
plt.scatter(X.loc[:, 'x1'], X.loc[:, 'x2'], label='unlabeled')
plt.scatter(X_labeled['x1'], X_labeled['x2'], label='labeled')
plt.title("原始数据分布", font2)
plt.xlabel('x1', font2)
plt.ylabel('x2', font2)
plt.legend(loc='upper left')
plt.show()
```



KMeans数据聚类实战流程

数据加载及展示

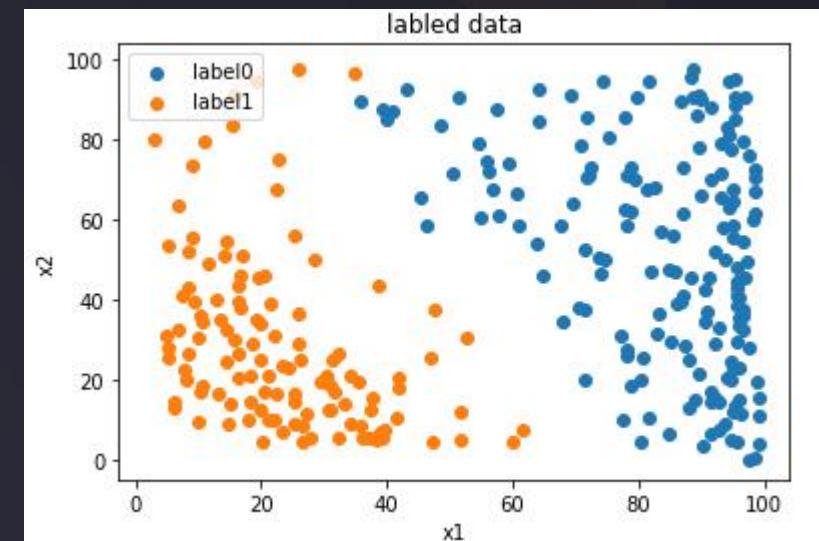
数据预处理

模型建立及训练

模型预测

结果展示及表现评估

```
#带标签数据可视化  
fig1 = plt.figure()  
label0 = plt.scatter(X.loc[:, 'x1'][y==0], X.loc[:, 'x2'][y==0])  
label1 = plt.scatter(X.loc[:, 'x1'][y==1], X.loc[:, 'x2'][y==1])  
  
plt.title("labeled data")  
plt.xlabel('x1')  
plt.ylabel('x2')  
plt.legend((label0,label1),('label0','label1'), loc='upper left')  
plt.show()
```



KMeans数据聚类实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#模型建立与训练

```
from sklearn.cluster import KMeans
```

```
KM =
```

```
KMeans(n_clusters=2,init='random',random_state=0)
```

```
KM.fit(X)
```

```
KMeans(algorithm='auto', copy_x=True, init='random', max_iter=300, n_clusters=2,  
       n_init=10, n_jobs=None, precompute_distances='auto', random_state=0,  
       tol=0.0001, verbose=0)
```

参考链接: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

KMeans数据聚类实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

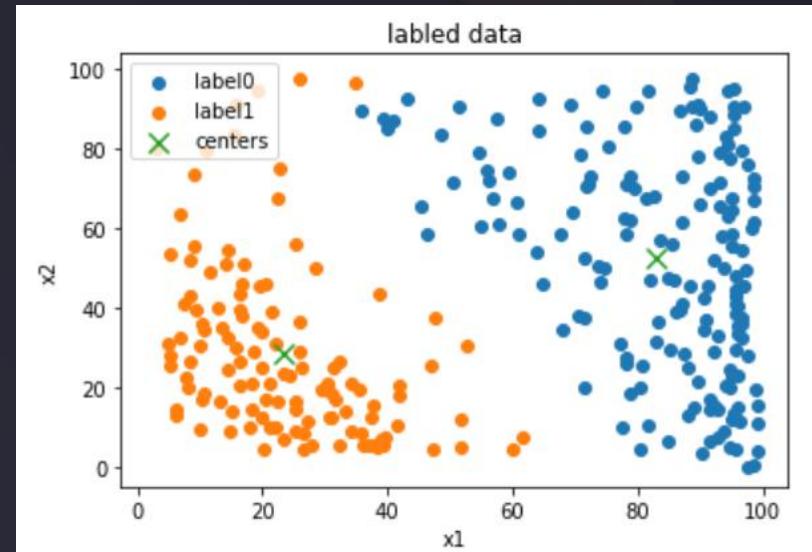
结果展示

#查看聚类中心

```
centers = KM.cluster_centers_
print(centers)
```

```
[[ 23.23572487 28.61664752]
 [82.88964583 52.50966869]]
```

```
plt.scatter(centers[:,0],centers[:,1],100,marker='x',label='centers')
```



KMeans数据聚类实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测及表现评估

结果展示

#结果预测

```
y_predict = KM.predict(X)
```

```
print(pd.value_counts(y_predict),pd.value_
counts(y))
```

```
1    168
0    117
dtype: int64
0    167
1    118
Name: y, dtype: int64
```

#准确率计算

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y,y_predict)
print(accuracy)
```

```
0.0035087719298245615
```

KMeans实战

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

#可视化预测结果

```
plt.figure(figsize=(20,10))
```

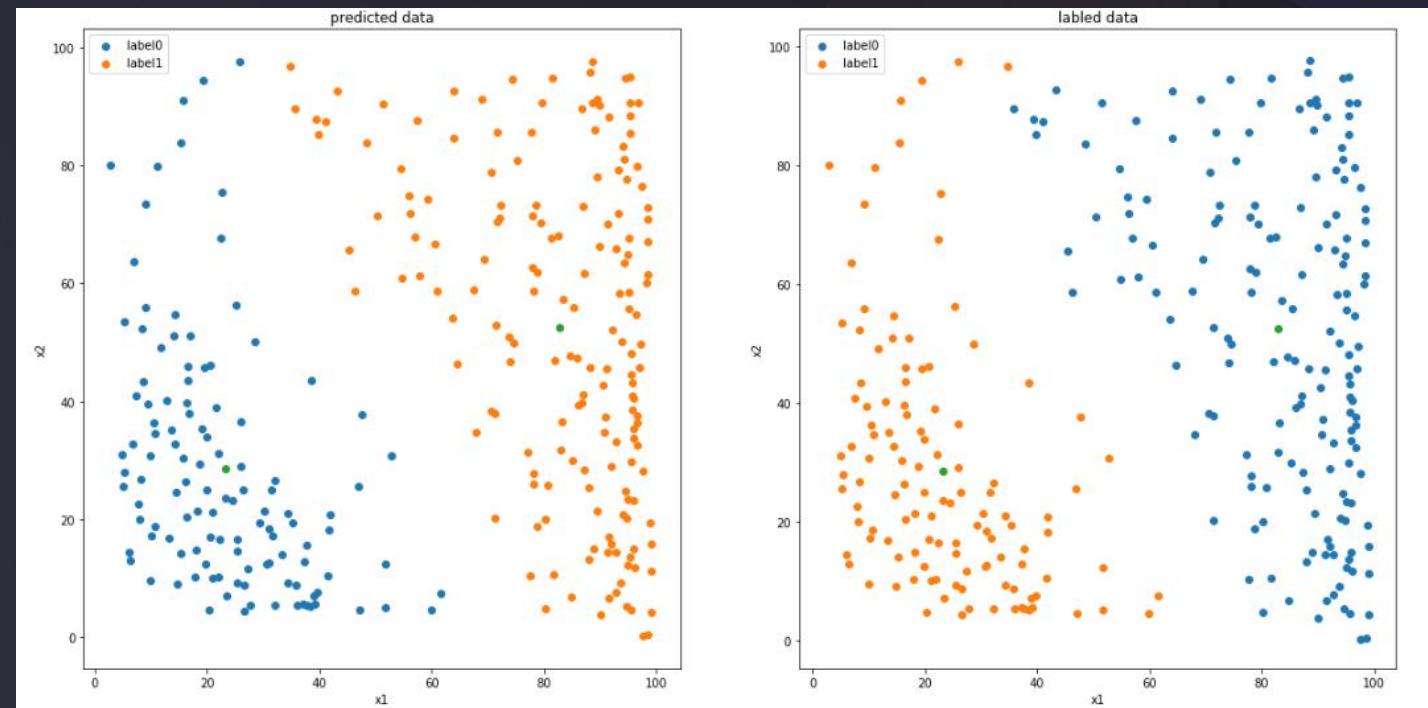
```
fig4 = plt.subplot(121)
```

```
label0 =
```

```
plt.scatter(X.loc[:, 'x1'][y_predict==0], X.loc[:, 'x2'][y_predict==0])
```

```
label1 =
```

```
plt.scatter(X.loc[:, 'x1'][y_predict==1], X.loc[:, 'x2'][y_predict==1])
```



KMeans实战

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

```
#查看并对比第一个样本点  
print(X_labeled)  
print(y_predict[0])
```

```
x1      82.5302  
x2      67.9939  
y       0.0000  
Name: 0, dtype: float64  
1
```

KMeans数据聚类实战流程

结果矫正

```
#结果矫正
y_corrected = []
for i in y_predict:
    if i==0:
        y_corrected.append(1)
    elif i==1:
        y_corrected.append(0)

print(pd.value_counts(y_corrected),pd.value_counts(y))
```

```
print(accuracy_score(y,y_corrected))
```

```
1      168
0      117
dtype: int64
0      167
1      118
Name: y, dtype: int64
```

```
0      168
1      117
dtype: int64
0      167
1      118
Name: y, dtype: int64
```

```
0.9964912280701754
```

KMeans数据聚类实战流程

查看模型迭代过程中的结果变化

#迭代一次的结果

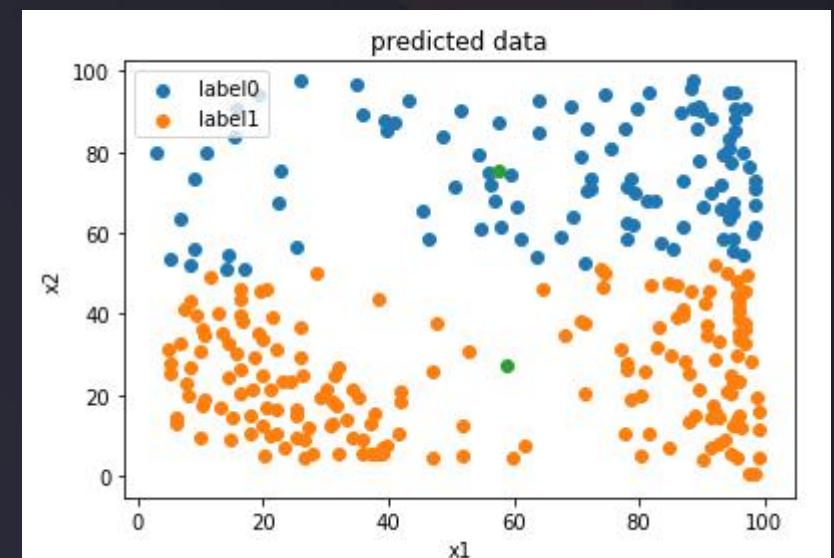
```
KM = KMeans(n_clusters=2,init='random',random_state=1,n_init=1,max_iter=1)  
KM.fit(X)
```

#迭代1-9次的结果

```
for i in range(1,10):  
    KM = KMeans(n_clusters=2,random_state=1,  
    init='random',n_init=1,max_iter=i)  
    KM.fit(X)
```

`n_init` : Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of `n_init` consecutive runs in terms of inertia.

`max_iter` : Maximum number of iterations of the k-means algorithm for a single run.



KMeans数据聚类实战流程

查看模型迭代过程中的结果变化



Kmeans_batch_
code.ipynb

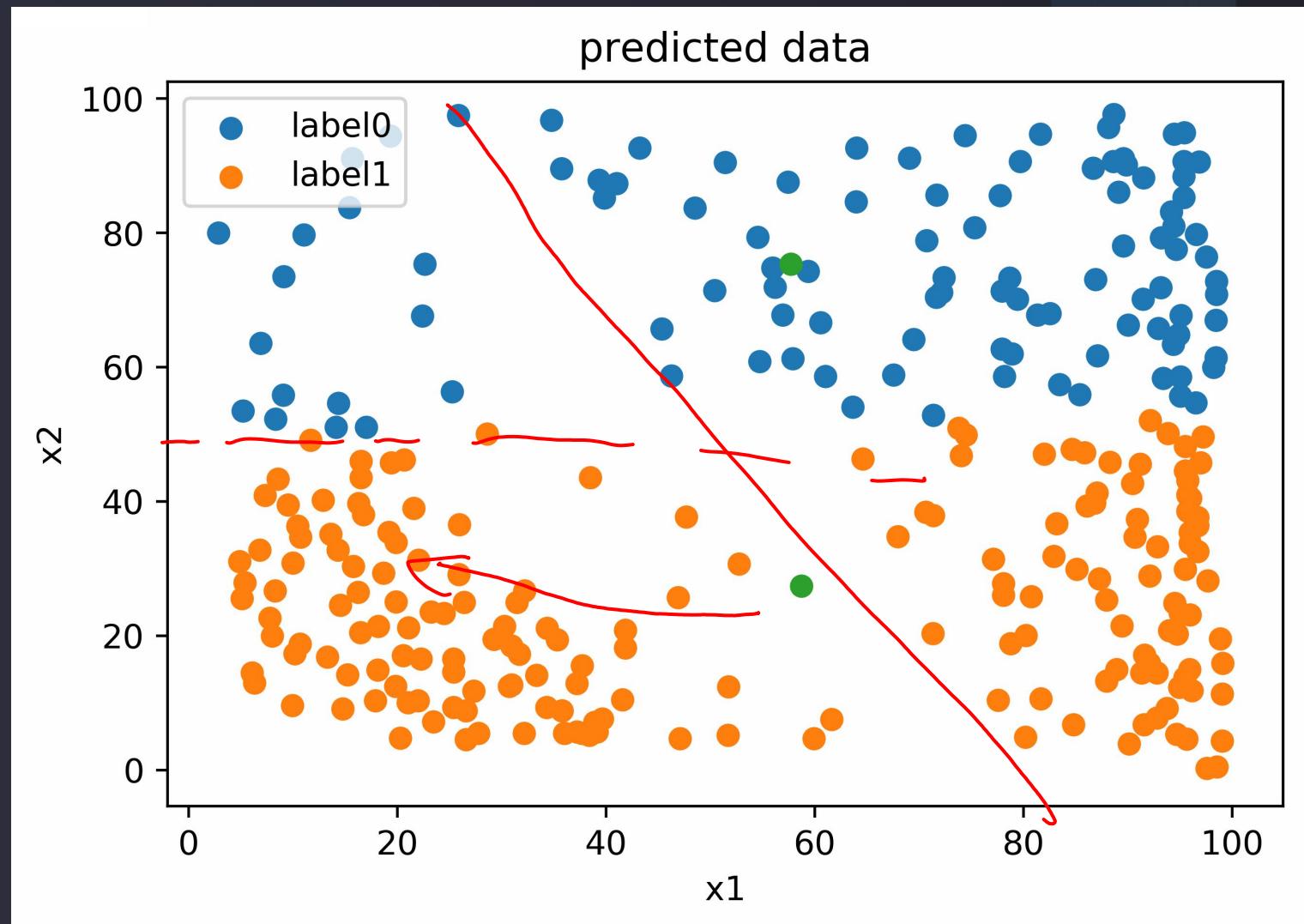
```
#逐步迭代查看KMeans模型训练效果
centers = np.array([[0,0,0,0]])
for i in range(1,10):
    KM = KMeans(n_clusters=2,random_state=1,init='random',n_init=1,max_iter=i)
    KM.fit(X)

    centers_i = KM.cluster_centers_
    centers_i_temp = centers_i.reshape(1,-1)
    centers = np.concatenate((centers,centers_i_temp),axis=0)
#predict based on training data
y_predict = KM.predict(X)

#visualize the data and results
fig_i = plt.figure()
label0 = plt.scatter(X.loc[:, 'x1'][y_predict==0],X.loc[:, 'x2'][y_predict==0])
label1 = plt.scatter(X.loc[:, 'x1'][y_predict==1],X.loc[:, 'x2'][y_predict==1])

plt.title("predicted data")
plt.xlabel('x1')
plt.ylabel('x2')
plt.legend([label0,label1],('label0','label1'), loc='upper left')
plt.scatter(centers_i[:,0],centers_i[:,1])
fig_i.savefig('2d_output/{}.png'.format(i),dpi=500,bbox_inches = 'tight')
```

KMeans数据聚类实战流程



KMeans数据聚类实战流程

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#建立KNN模型及训练

```
from sklearn.neighbors import KNeighborsClassifier  
KNN = KNeighborsClassifier(n_neighbors=3)  
KNN.fit(X,y)
```

y_predict_knn = KNN.predict(X)

```
print('knn accuracy:',accuracy_score(y,y_predict_knn))
```

knn accuracy: 1.0

| 任务二：KMeans实现图像分割

加载本地图像1.jpg，建立Kmeans模型实现图像分割。



- 1、实现图像加载、可视化、维度转化，完成数据的预处理；
- 2、 $K=3$ 建立Kmeans模型，实现图像数据聚类；
- 3、对聚类结果进行数据处理，展示分割后的图像；
- 4、尝试其他的 K 值 ($K=4, 8$)，对比分割效果，并思考导致结果不同的原因；
- 5、使用新的图片，对其实现图像分割

KMeans实现图像分割

数据加载及展示

数据预处理

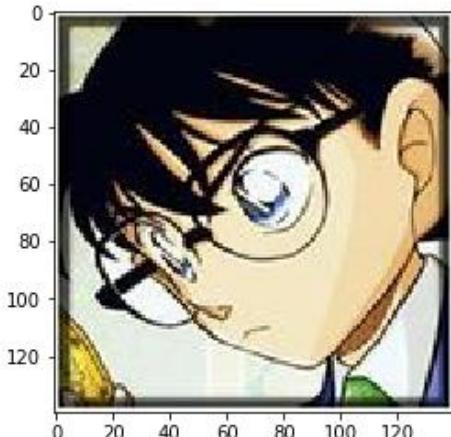
模型建立及训练

模型预测

结果展示及表现评估

```
#图像加载与展示  
import numpy as np  
import matplotlib.pyplot as plt  
from skimage import io as io  
img = io.imread("1.jpg")  
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x142b9216208>
```



KMeans实现图像分割

数据加载及展示

数据预处理

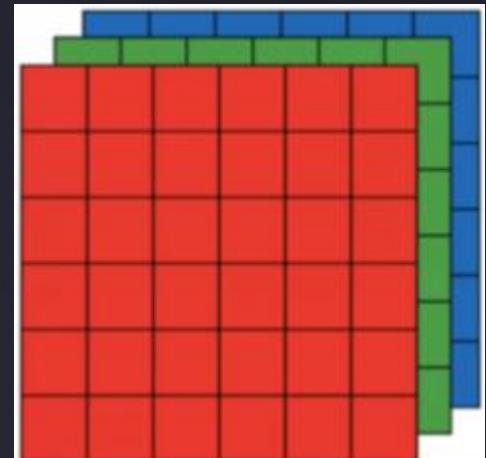
模型建立及训练

模型预测

结果展示及表现评估

```
#查看数据结构及维度  
print(type(img))  
print(img.shape)
```

```
<class 'numpy.ndarray'>  
(140, 140, 3)
```



KMeans实现图像分割

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#数据维度转化

```
img_data = np.reshape(img,(-1,3))  
print(img_data.shape)  
print(img_data)
```

```
(19600, 3)  
[[254 255 250]  
 [255 255 251]  
 [246 247 241]  
 ...  
 [240 246 246]  
 [246 252 248]  
 [250 255 250]]
```

KMeans实现图像分割

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示及表现评估

#模型建立与训练

```
from sklearn.cluster import KMeans  
KM = KMeans(n_clusters=3,random_state=0)  
#聚类中心的个数为3  
KM.fit(img_data)
```

参考链接: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

KMeans实现图像分割

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

```
#聚类结果预测  
label=KM.predict(img_data)  
print(label.shape)  
print(label)
```

```
(19600, )  
[1 1 1 ... 1 1 1]
```

```
#维度度转化  
label = label.reshape([img_height,img_width])  
print(label.shape)  
print(label)
```

```
(140, 140)  
[[1 1 1 ... 1 1 1]  
 [1 1 2 ... 2 1 1]  
 [1 2 0 ... 0 2 1]  
 ...  
 [1 2 0 ... 0 2 1]  
 [1 1 2 ... 2 2 1]  
 [1 1 1 ... 1 1 1]]
```

KMeans实现图像分割

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

```
#聚类类别结果转灰度值  
label = 1/(label+1)  
print(label.shape)  
print(label)
```

```
(140, 140)  
[[0.5 0.5 0.5 ... 0.5 0.5 0.5 ]]  
[0.5 0.5 0.33333333 ... 0.33333333 0.5 0.5 ]]  
[0.5 0.33333333 1. ... 1. 0.33333333 0.5 ]]  
...  
[0.5 0.33333333 1. ... 1. 0.33333333 0.5 ]]  
[0.5 0.5 0.33333333 ... 0.33333333 0.33333333 0.5 ]]  
[0.5 0.5 0.5 ... 0.5 0.5 0.5 ]]
```

KMeans实现图像分割

数据加载及展示

```
#图片保存到本地  
io.imsave("test4.png",label)
```

数据预处理

模型建立及训练

模型预测

结果展示





Python3人工智能入门+实战提升：机器学习

Chapter 5 无监督学习与聚类分析

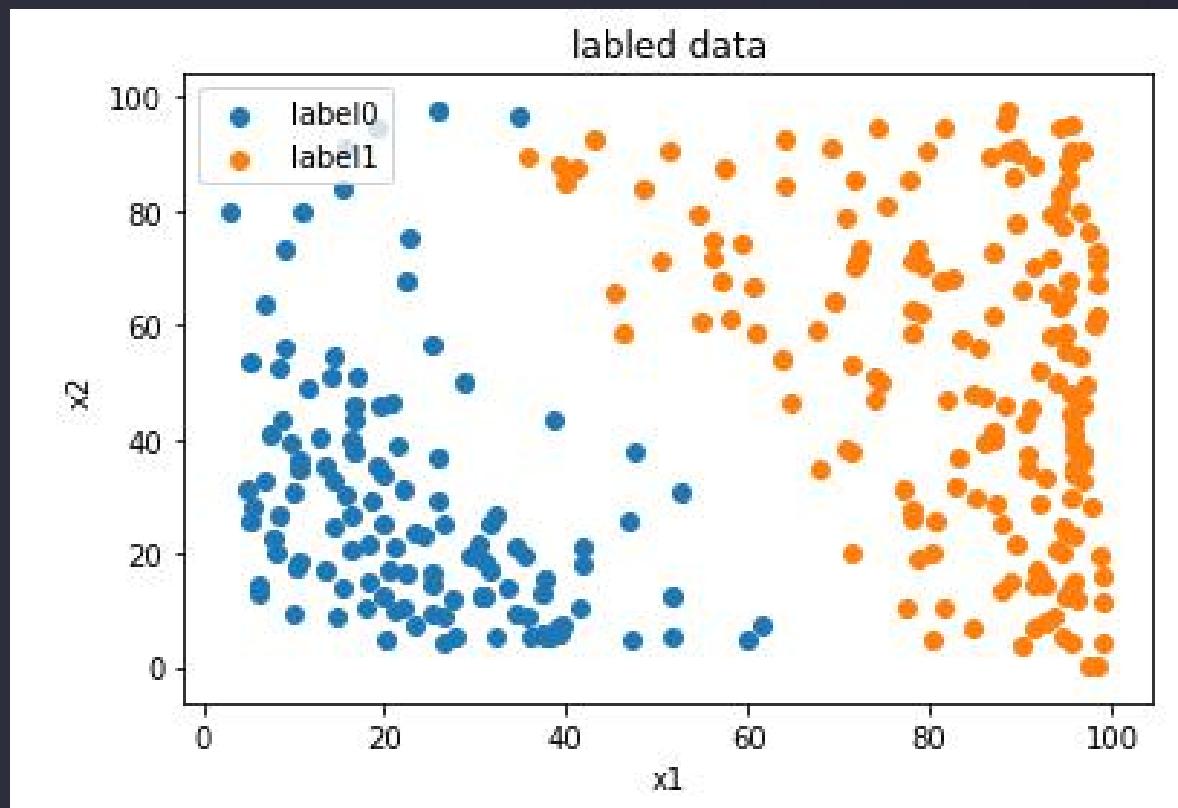
赵辛

Chapter 5 无监督学习与聚类分析

-
- 1 --无监督学习 (Unsupervised Learning)
 - 2 --k均值聚类算法(Kmeans)
 - 3 --实战准备
 - 4 --实战 (一) KMeans实现数据聚类
 - 5 --实战 (二) KMeans实现图像分割
 - 6 --现实问题思考：监督真的重要吗

| 任务一：KMeans实现数据聚类

基于2D_data数据，建立Kmeans模型。



- 1、K=2，实现数据聚类，可视化聚类结果、聚类中心；
- 2、对比聚类结果与实际类别，进行结果矫正
- 3、建立KNN模型，对比分类结果，思考其与聚类结果的差异
- 4、修改Kmeans迭代次数与初始化参数，查看模型迭代过程中的结果变化



Python3人工智能入门+实战提升：机器学习

Chapter 5 无监督学习与聚类分析

赵辛

Chapter 5 无监督学习与聚类分析

1 --无监督学习 (Unsupervised Learning)

2 --k均值聚类算法(Kmeans)

3 --实战准备

4 --实战 (一) KMeans实现数据聚类

5 --实战 (二) KMeans实现图像分割[CSDN](#)

6 --现实问题思考：监督真的重要吗

| 任务二：KMeans实现图像分割

加载本地图像1.jpg，建立Kmeans模型实现图像分割。



- 1、实现图像加载、可视化、维度转化，完成数据的预处理；
- 2、 $K=3$ 建立Kmeans模型，实现图像数据聚类；
- 3、对聚类结果进行数据处理，展示分割后的图像；
- 4、尝试其他的 K 值 ($K=4, 8$)，对比分割效果，并思考导致结果不同的原因；
- 5、使用新的图片，对其实现图像分割



Python3人工智能入门+实战提升：机器学习

Chapter 5 无监督学习与聚类分析

赵辛

Chapter 5 无监督学习与聚类分析

-
- 1 --无监督学习 (Unsupervised Learning)
 - 2 --k均值聚类算法(Kmeans)
 - 3 --实战准备
 - 4 --实战 (一) KMeans实现数据聚类
 - 5 --实战 (二) KMeans实现图像分割
 - 6 --现实问题思考：监督真的重要吗

|现实问题思考：监督真的重要吗



不同科目的知识学习

数学乘法： $2*3=6$

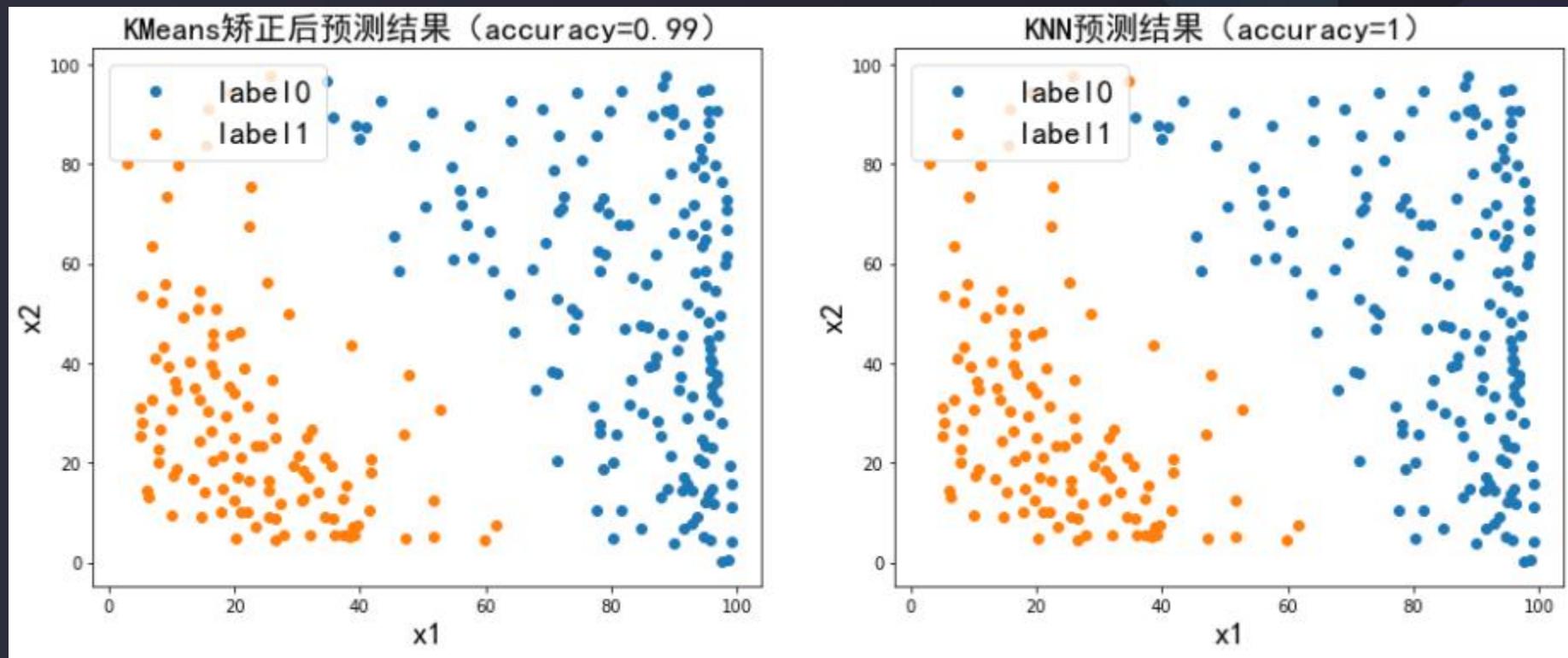
语文近义词：喜欢-喜爱

英语翻译： I 我

物理公式： $S=vt$

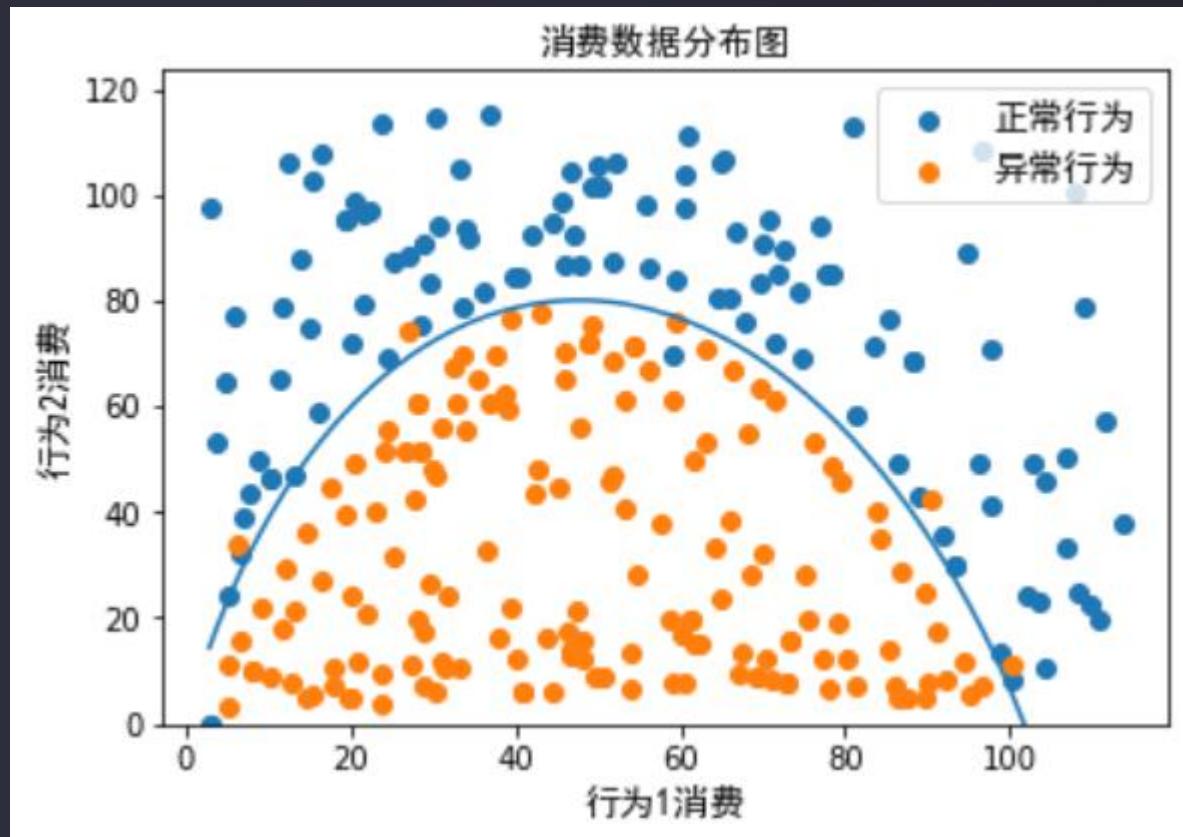
...

现实问题思考：监督真的重要吗



某些场景，无监督学习+一定的矫正方法，也可以达到很好的效果，而且无监督对数据标签要求低，可以极大程度降低数据采集难度。

现实问题思考：监督真的重要吗



场景变复杂以后，必须要通过监督学习才能达到好的分类效果。

现实场景：

- 1、任务复杂；
- 2、采集大量数据有难度

现实问题思考：监督真的重要吗

现实场景：

- 1、任务复杂；
- 2、采集大量数据有难度

解决办法：

- 1、大部分场景都需要监督学习；
- 2、条件允许的情况下尽可能收集足够的样本；
- 3、无法收集足够样本的情况下，考虑标签样本+无标签样本实现监督学习与无监督学习的结合，即半监督学习

|现实问题思考：监督真的重要吗

- 1、大部分应用场景中，条件允许情况下，优先考虑监督学习；
- 2、部分特定场景，无监督学习能够帮我们找到“惊喜”（预料之外的数据关系）；
- 3、未来的一大方向：监督+无监督，实现少量标签样本下的数据学习，在保证精度的同时极大降低数据采集难度



Python3人工智能入门+实战提升：机器学习

Chapter 6 异常检测与数据降维

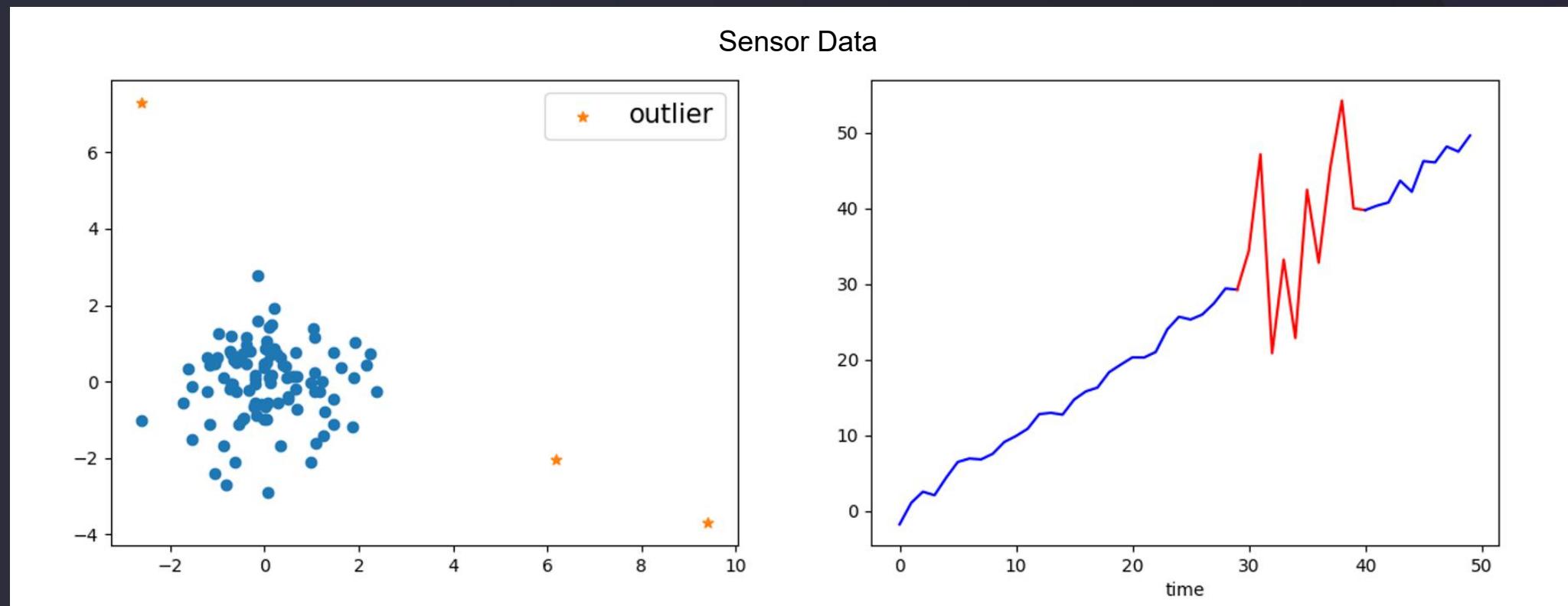
赵辛

Chapter 6 异常检测与数据降维

-
- 1 --异常检测（一）
 - 2 --异常检测（二）
 - 3 --主成分分析（PCA）
 - 4 --实战准备
 - 5 --实战（一）异常消费检测
 - 6 --实战（二）PCA降维之糖尿病检测

现实问题思考

以下为监控传感器检测的设备工作数据，如何让机器在接收到异常数据时自动报警？



|现实问题思考

当相机画面中突然出现异常目标，如何实现自动识别？



更多案例

欺诈检测：盗刷信用卡检测

入侵检测：检测网络入侵或计算机入侵行为

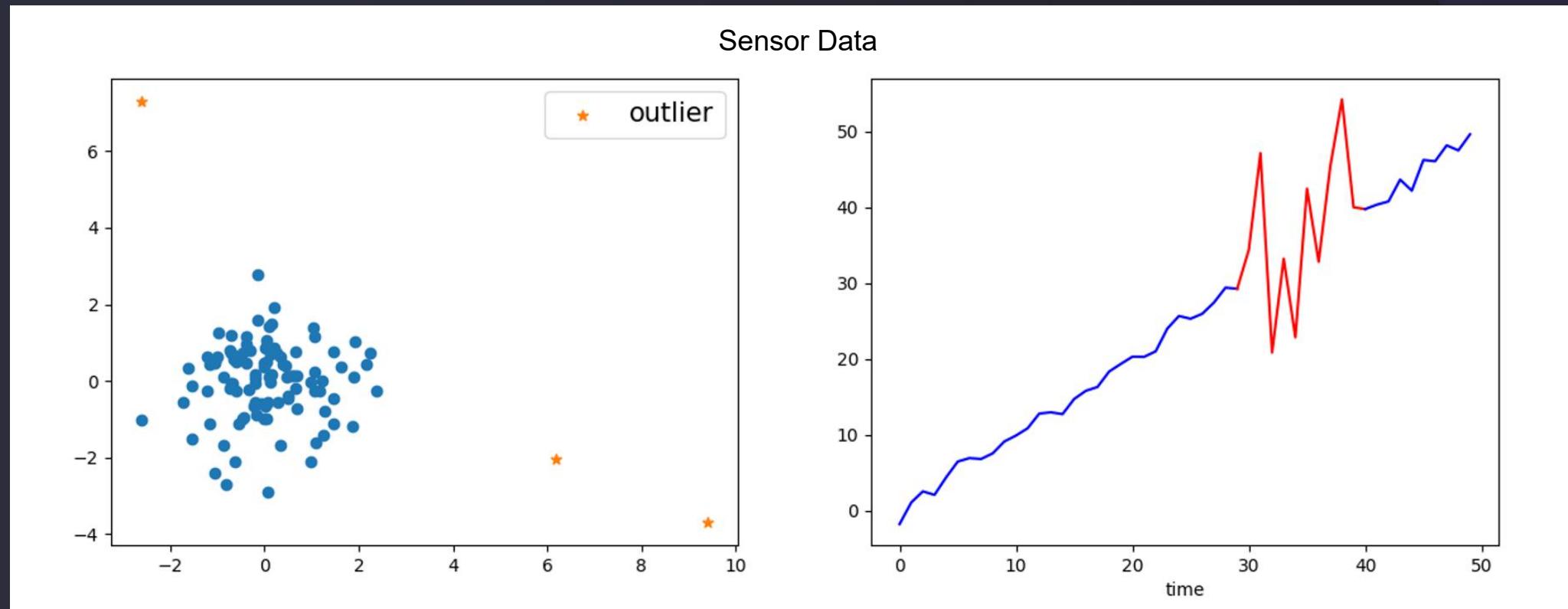
医疗：缺陷基因检测

生态系统：预测飓风、洪水、干旱、热浪和
火灾的发生

...

异常检测

根据输入数据，对不匹配预期模式的数据进行识别



| 异常检测

监督式异常检测：提前使用带“正常”与“异常”标签的数据对模型进行训练，机器基于训练好的模型判断新数据是否为异常数据



“正常”

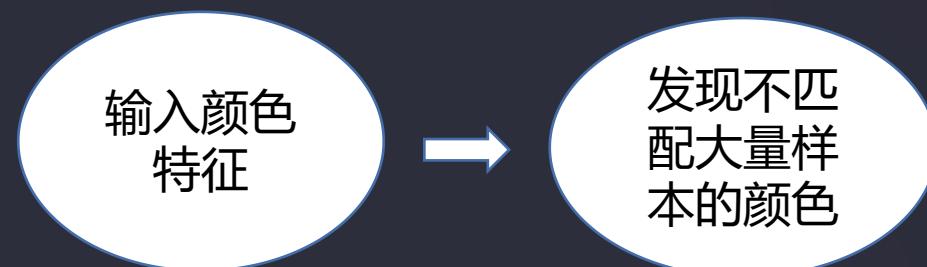
机器

“异常”



| 异常检测

无监督式异常检测：通过寻找与其他数据最不匹配的实例来检测出未标记测试数据的异常



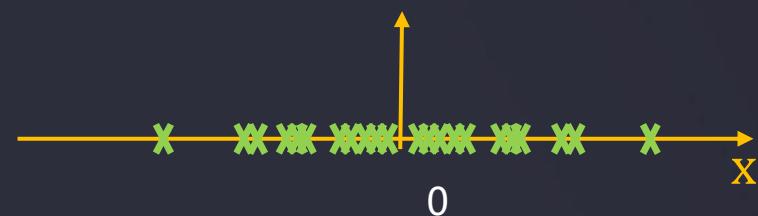
| 异常检测

基于数据分布，寻找与其他数据最不匹配的实例

一维数据集：

$$\{x^{(1)}, x^{(2)}, \dots x^{(m)}\}$$

寻找发生可能性低的
数据（事件）



| 概率(Probability)

概率是一个在0到1之间的实数，是对随机事件发生可能性的度量，反映某种情况出现的可能性 (likelihood)大小。



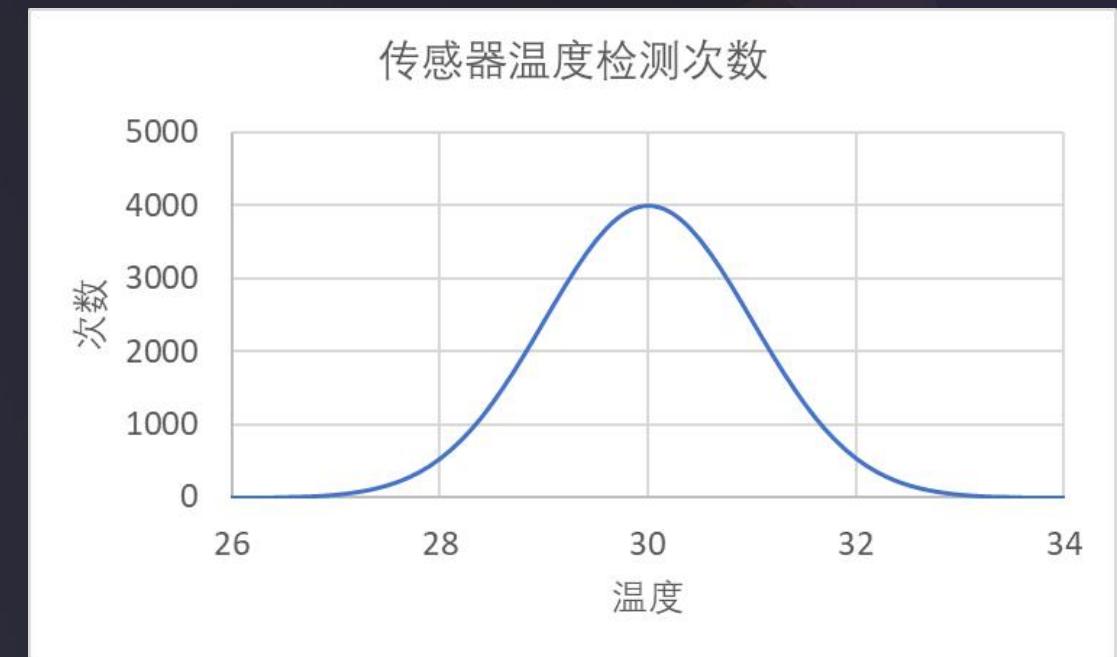
| 概率(Probability)

不连续分布事件



结果类别有限，比如1、2、3、4、5、6

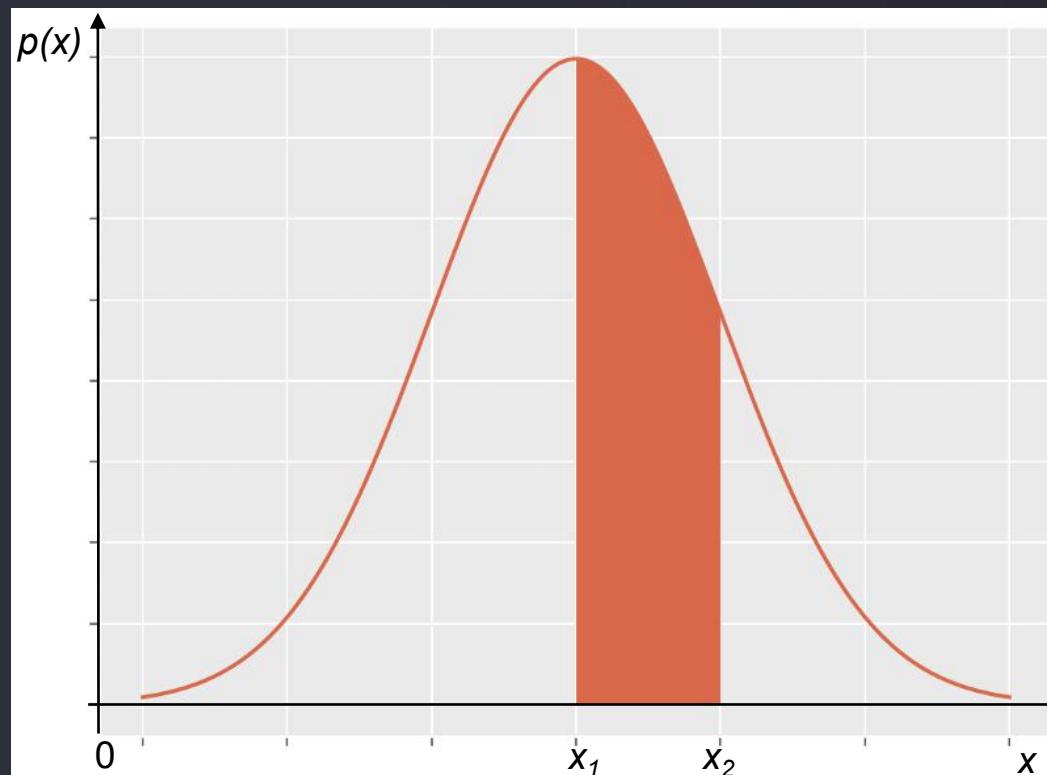
连续分布事件



结果类别无限，传感器温度：26-34度任意值

概率密度函数

在连续分布事件中，用于描述连续随机变量的输出值在某个确定的取值点附近的可能性的函数，通过其可计算取值点附近区间发生事件的概率。



x 分布概率密度图

连续分布事件 x 发生的概率密度函数为 $p(x)$ ，则

区间 (x_1, x_2) 发生的概率为：

$$P(x_1, x_2) = \int_{x_1}^{x_2} p(x)dx$$



Python3人工智能入门+实战提升：机器学习

Chapter 6 异常检测与数据降维

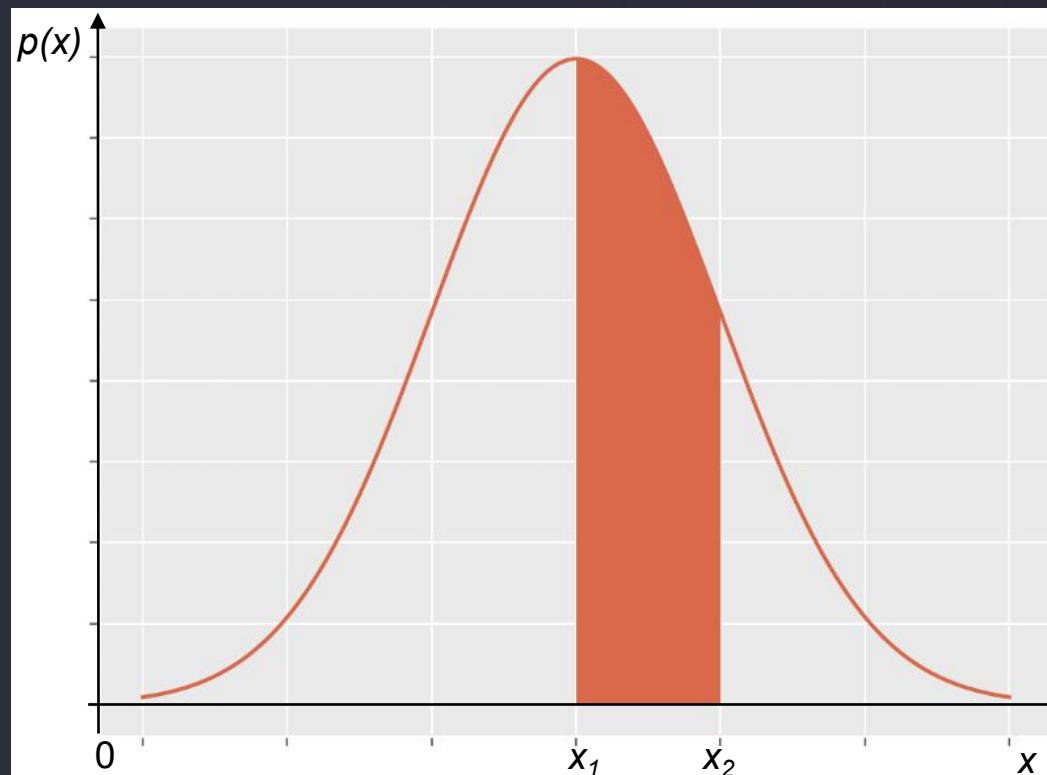
赵辛

Chapter 6 异常检测与数据降维

-
- 1 --异常检测（一）
 - 2 --异常检测（二）
 - 3 --主成分分析（PCA）
 - 4 --实战准备
 - 5 --实战（一）异常消费检测
 - 6 --实战（二）PCA降维之糖尿病检测

概率密度函数

在连续分布事件中，用于描述连续随机变量的输出值在某个确定的取值点附近的可能性的函数，通过其可计算取值点附近区间发生事件的概率。



X分布概率密度图

连续分布事件x发生的概率密度函数为 $p(x)$ ，则

区间 (x_1, x_2) 发生的概率为：

$$P(x_1, x_2) = \int_{x_1}^{x_2} p(x)dx$$

| 基于高斯分布的概率密度函数

高斯分布 (正态分布)的概率密度函数是：

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

其中， μ 为数据均值， σ 为标准差

概率论中最重要的分布

现实生活中，很多事件发生的频率都符合高斯分布，比如：工业产品的强力、抗压强度、口径、长度等指标；人体的身高、体重等指标；同一品种种子的重量；某个地区的年降水量，等等。

基于高斯分布的概率密度函数

μ 决定中间轴位置， σ 为决定分布集中度

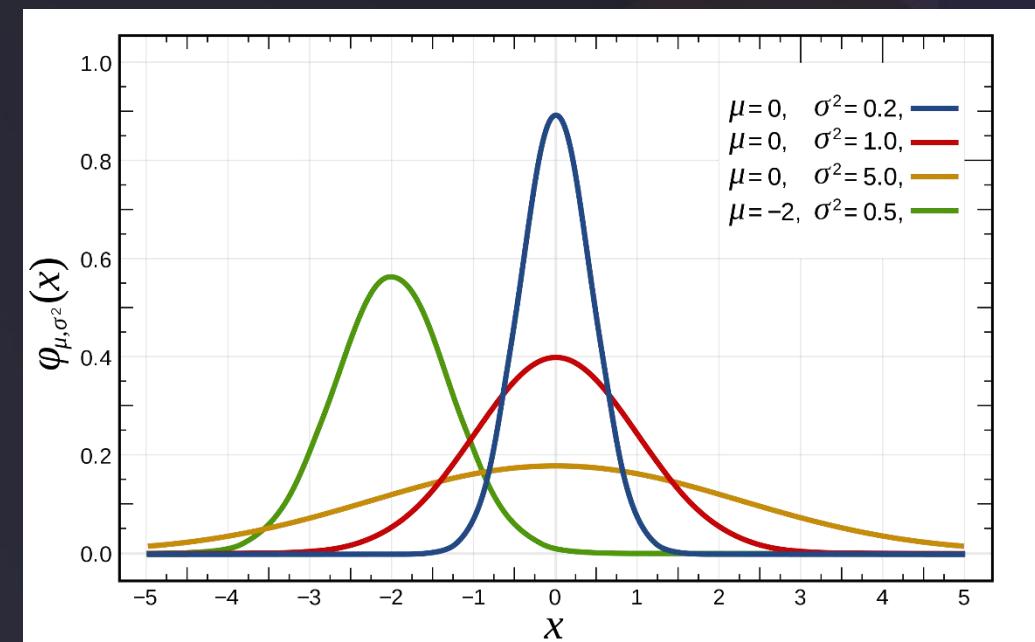
高斯分布（正态分布）的概率密度函数是：

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

其中， μ 为数据均值， σ 为标准差

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)},$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$$



高斯分布概率密度图

| 基于高斯分布的概率密度函数

高斯分布(正态分布)的概率密度函数是:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

其中, μ 为数据均值, σ 为标准差

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)},$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$$

$$P(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

示例数据

x_1	x_2	x_3	x_4
-3	0	2	1

$$\underline{\mu} = \frac{1}{4} (-3+0+2+1) = \boxed{0}$$

$$\underline{\sigma^2} = \frac{1}{4} [(-3-0)^2 + 0^2 + (2-0)^2 + (1-0)^2] \\ = \frac{1}{4} \times (9+4+1) = \frac{14}{4} = \boxed{3.5}$$

| 基于高斯分布概率密度函数实现异常检测

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
-10	-7	-6.5	-6	-2.5	-2	-1	-1
X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}
1	1	1.5	2	4.5	5	6	9



1、根据输入数据 x , 计算均值 μ , 标准差 σ

3、根据数据点概率密度, 进行判断

2、根据 μ 、 σ 得到对应的高斯分布概率函数:

如果 $p(x) < \text{阈值}\varepsilon$:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

判断该点为异常点

| 基于高斯分布概率密度函数实现异常检测

当数据维度高于一维怎么办

(m个样本，每个样本有n个特征):

$$\begin{cases} x_1^{(1)}, x_1^{(2)}, \dots x_1^{(n)} \\ \dots \\ x_m^{(1)}, x_m^{(2)}, \dots x_m^{(n)} \end{cases}$$

1、计算数据均值 $\mu_1, \mu_2, \dots, \mu_n$, 标准差 $\sigma_1, \sigma_2, \dots, \sigma_n$

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_i^{(j)}, \quad \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_i^{(j)} - \mu_j)^2$$

2、计算概率密度函数 $p(x)$

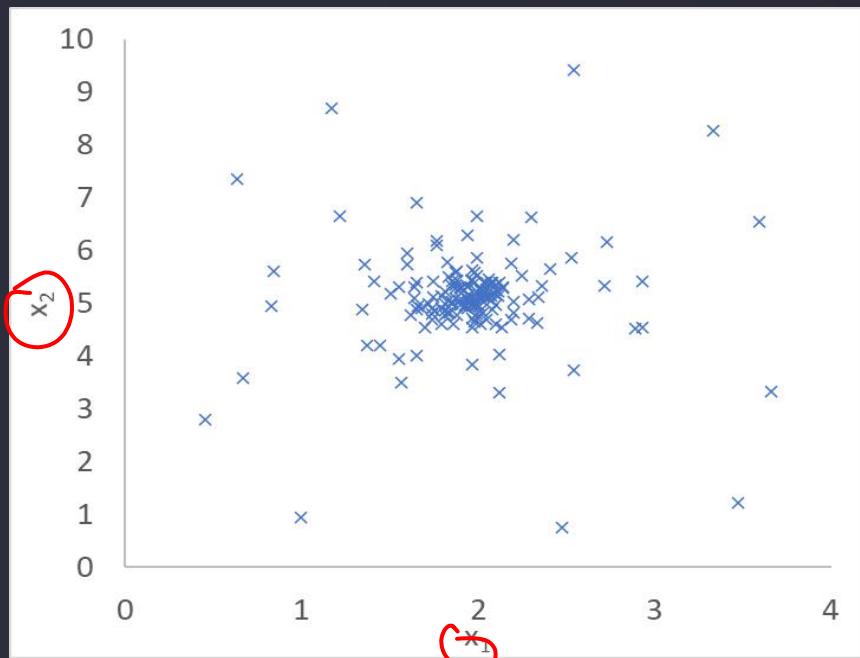
$$p(x) = \prod_{j=1}^n p(x^{(j)}; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(x^{(j)} - \mu_j)^2}{2\sigma_j^2}}$$

3、如果 $p(x) < \text{阈值}\varepsilon$:

判断该点为异常点

| 基于高斯分布概率密度函数实现异常检测

数据分布



$$\rightarrow \mu_1 = 2, \sigma_1 = 1$$

$$\rightarrow \mu_2 = 5, \sigma_2 = 0.5$$

$$\varepsilon = 0.01 \quad x^{(1)} = 1, x^{(2)} = 3$$

$$p(x) = \prod_{j=1}^n p(x^{(j)}; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(x^{(j)} - \mu_j)^2}{2\sigma_j^2}}$$

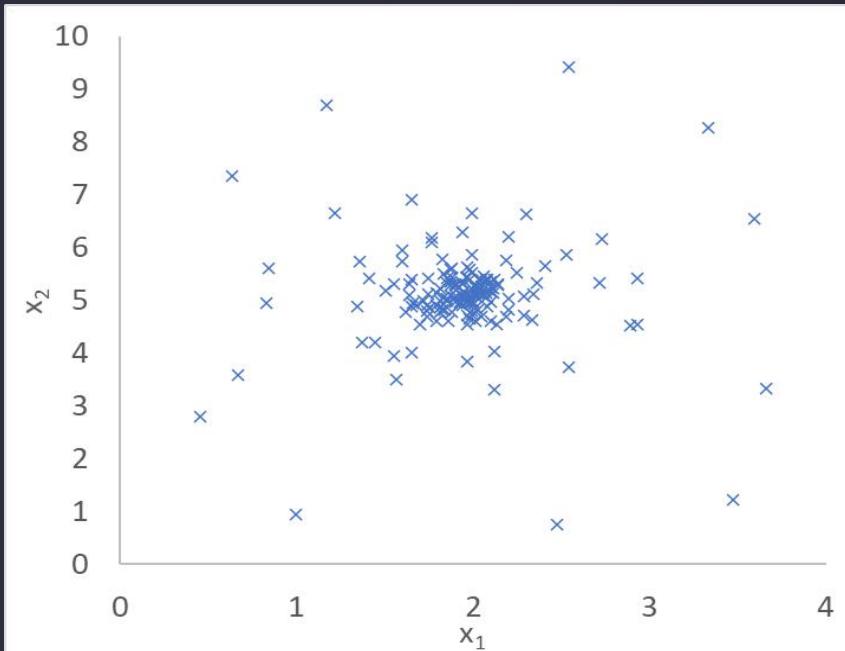
$$\overline{P(x_1)} = \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}} = \frac{1}{1 \sqrt{2\pi}} e^{-\frac{(1 - 2)^2}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}}$$

$$P(x_2) = \frac{1}{0.5 \sqrt{2\pi}} e^{-\frac{(3 - 5)^2}{2 \times 0.25}} = 0.000268$$

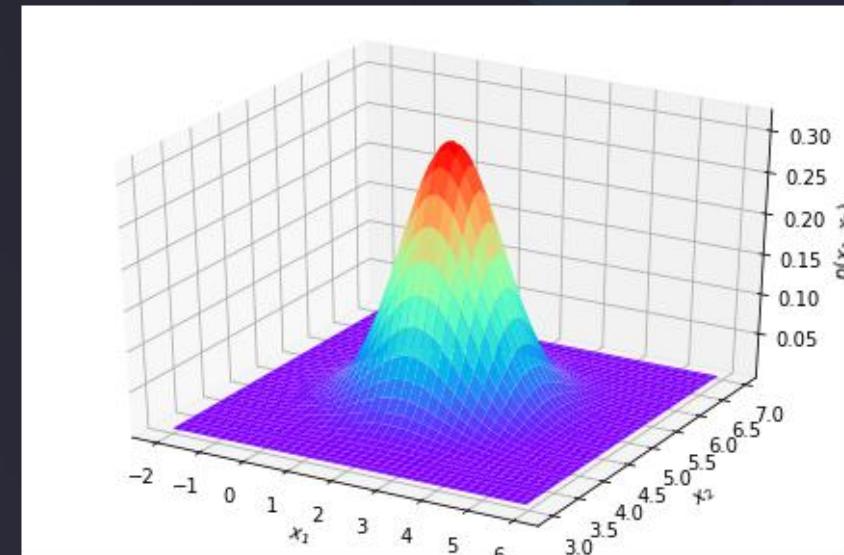
$$\Rightarrow P(x_1, x_2) = 0.2417 \times 0.000268 = \frac{6.5 \times 10^{-5}}{\varepsilon} = 0.001$$

| 基于高斯分布概率密度函数实现异常检测

数据分布



概率密度函数



$$\mu_1 = 2, \sigma_1 = 1$$

$$p = 0.000065 < \epsilon :$$

$$\mu_2 = 5, \sigma_2 = 0.5$$

异常点!

$$\epsilon = 0.01, x^{(1)} = 1, x^{(2)} = 3$$

知识巩固

问题：计算一下样本分布的均值、标准差、基于高斯分布的概率密度函数

x1	x2
0	3
3	4.5
3.5	4.7
3.5	5
4	5.1
4.2	5
3.6	5.4
5	7
7	6
3	5.6



Python3人工智能入门+实战提升：机器学习

Chapter 6 异常检测与数据降维

赵辛

Chapter 6 异常检测与数据降维

-
- 1 --异常检测（一）
 - 2 --异常检测（二）
 - 3 --**主成分分析（PCA）**
 - 4 --实战准备
 - 5 --实战（一）异常消费检测
 - 6 --实战（二）PCA降维之糖尿病检测

现实问题思考



简化机器学习框架

想建立一个AI模型，筛选金融股票，
潜在数据指标：

价格、交易量、换手率、股东人数、最近N日
涨跌幅、RSI指标、威廉指标、市值、营业额、
净利润、负债率、利润增长率...多达几百、上
千个因子

两大问题：
求解困难、模型过拟合

数据降维

在一定的限定条件下，按照一定的规则，尽可能保留原始数据集重要信息的同时，降低数据集特征的个数。

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$



$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mk} \end{bmatrix}$$

m个样本，每个样本n维特征

m个样本，每个样本k维特征

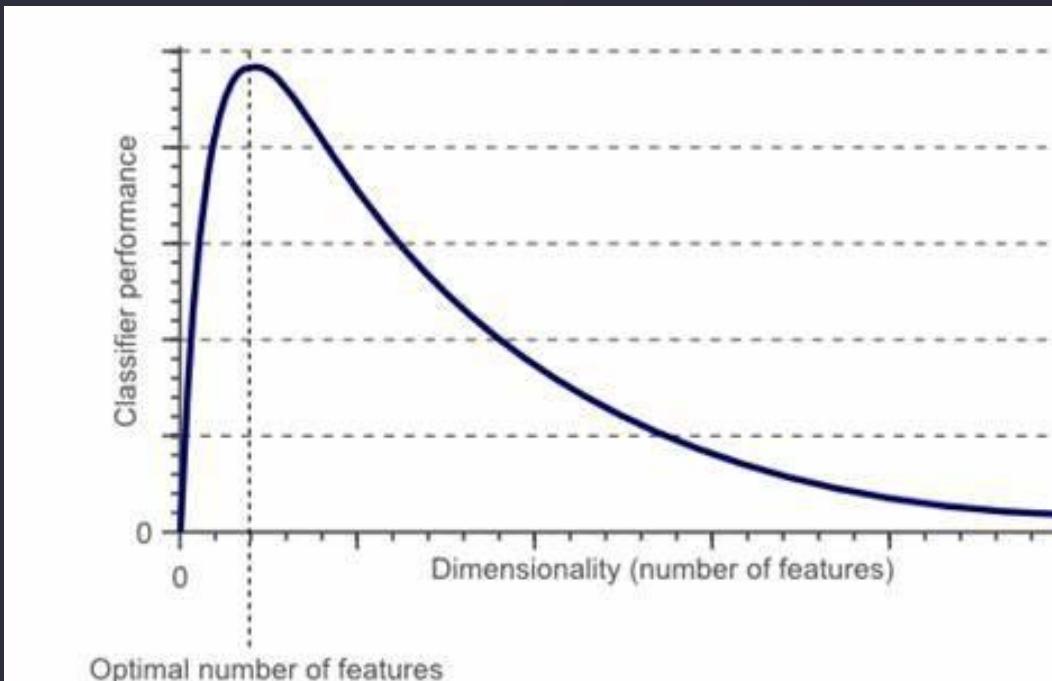
$$k < n$$

为什么需要数据降维

■ Curse of dimensionality - 维数灾难

随着特征数量越来越多，为了避免过拟合，对样本数量的需求会以**指数**速度增长。

任务：通过一封邮件的上百个特征，预测这封邮件是不是垃圾邮件



当样本数量确定时，特征数量并不是越多越好。

数据降维可以降低我们对样本数量的需求，同时简化学习过程。

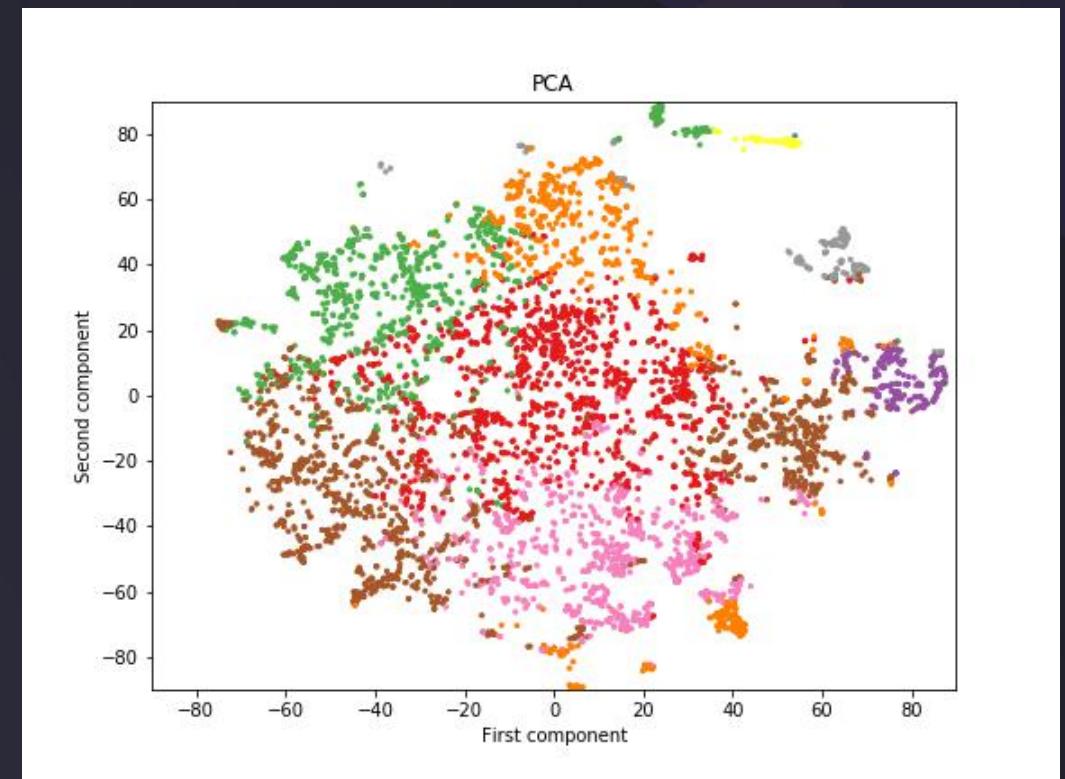
为什么需要数据降维

■ 数据可视化

高维数据不能可视化，只有降低到二维或三维才能可视化。

任务：输入是包含14个特征的脑电波数据^[1]，用来预测被测试者的状态。

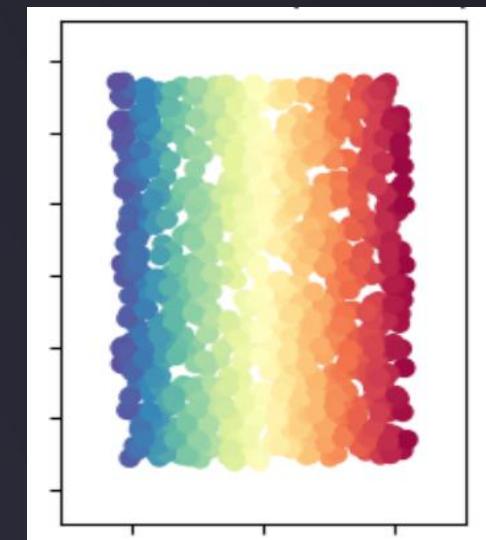
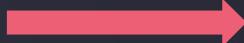
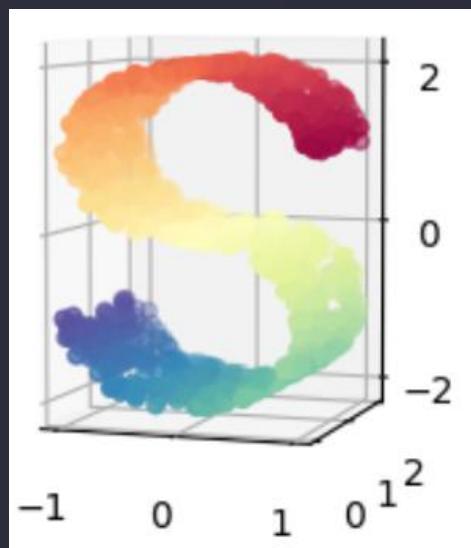
使用PCA将14维数据降成2维，实现了可视化。



[1] EEG Eye State Dataset. <http://archive.ics.uci.edu/ml/datasets/EEG+Eye+State#>

| 数据降维举例

3D数据降维到2D数据



| 数据降维最常用的方法：主成分分析（PCA）

- 也称主分量分析，按照一定规则把数据变换到一个新的坐标系统中，使得任何数据投影后尽可能可以分开（新数据尽可能不相关、分布方差最大化）。



主成分分析 (PCA)

核心：投影后的数据尽可能分得开（即不相关）

如何实现？

使投影后数据的方差最大，因为方差越大数据也越分散

计算过程：

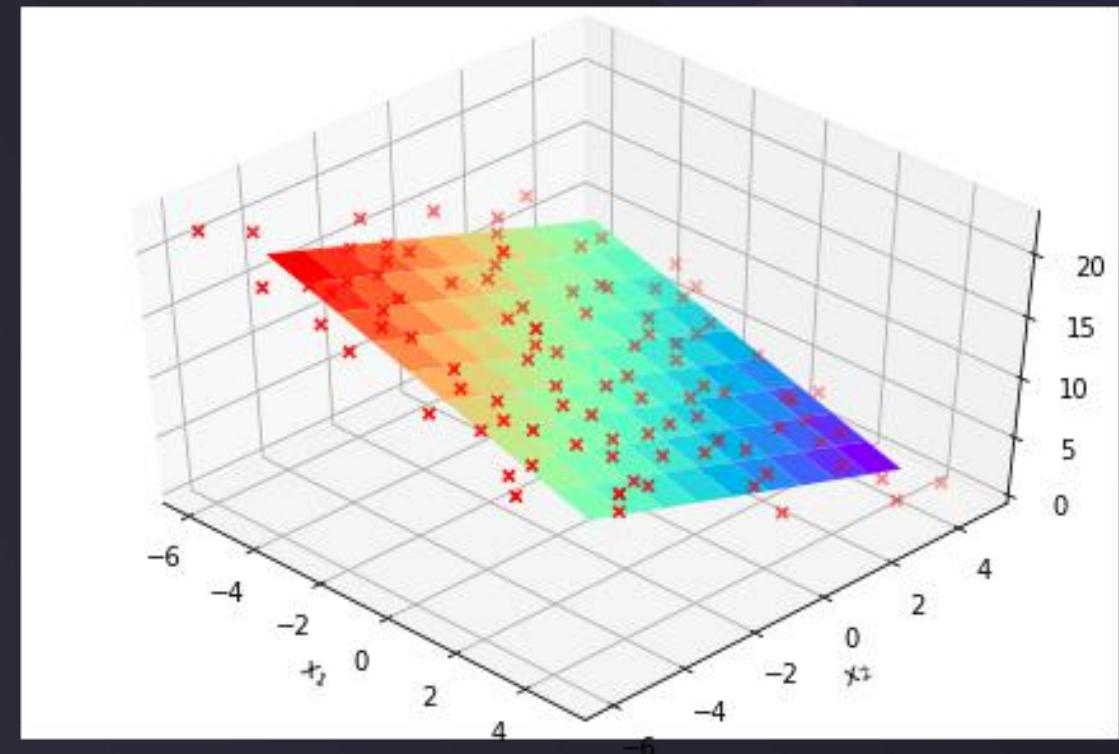
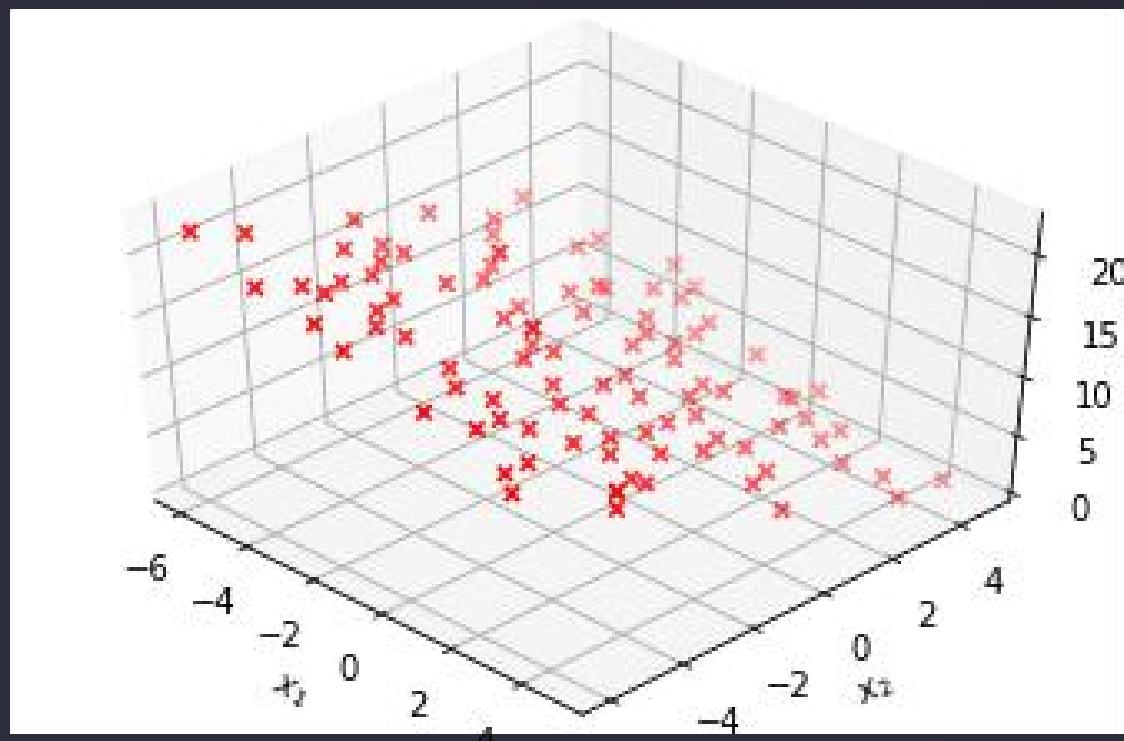
1. 数据预处理（数据分布标准化： $\mu = 0, \sigma = 1$ ）
2. 计算协方差矩阵特征向量、及数据在各特征向量投影后的方差
3. 根据需求（任务指定或方差比例）确定降维维度k
4. 选取k维特征向量，计算数据在其形成空间的投影

参考资料：

- 1、https://blog.csdn.net/dfly_zx/article/details/107908497

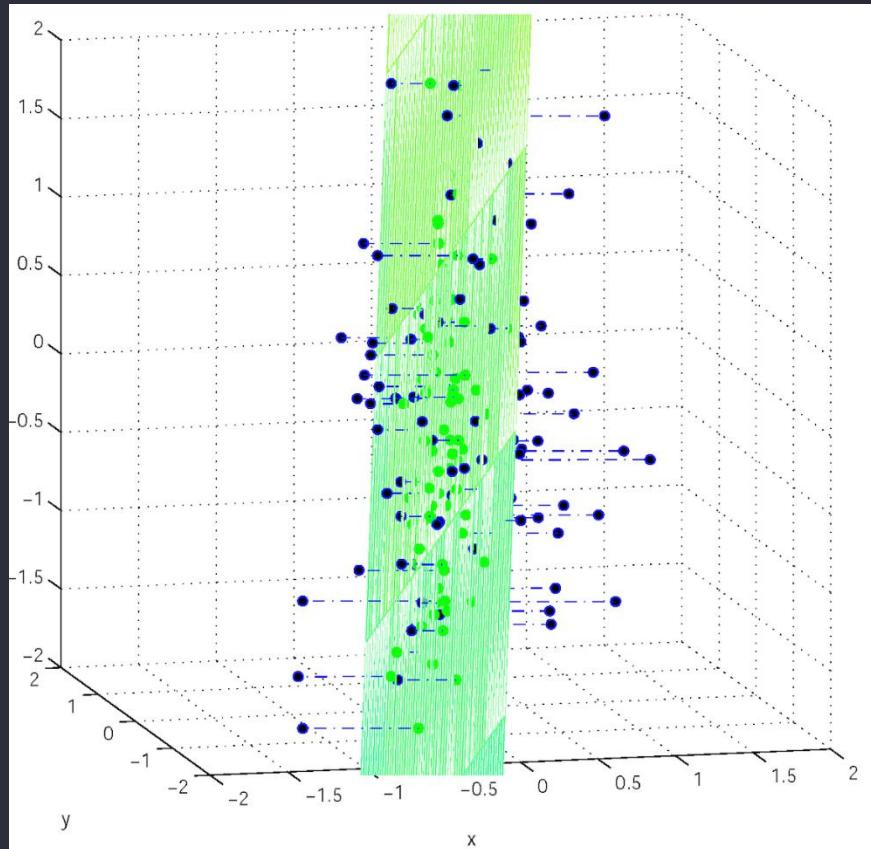
主成分分析 (PCA)

3维数据PCA降维到2维：
投影到 u_1 、 u_2 形成的平面



n维数据PCA降维到k维：投影到 u_1 、 u_2 ... u_k 形成的空间

| 主成分分析 (PCA)



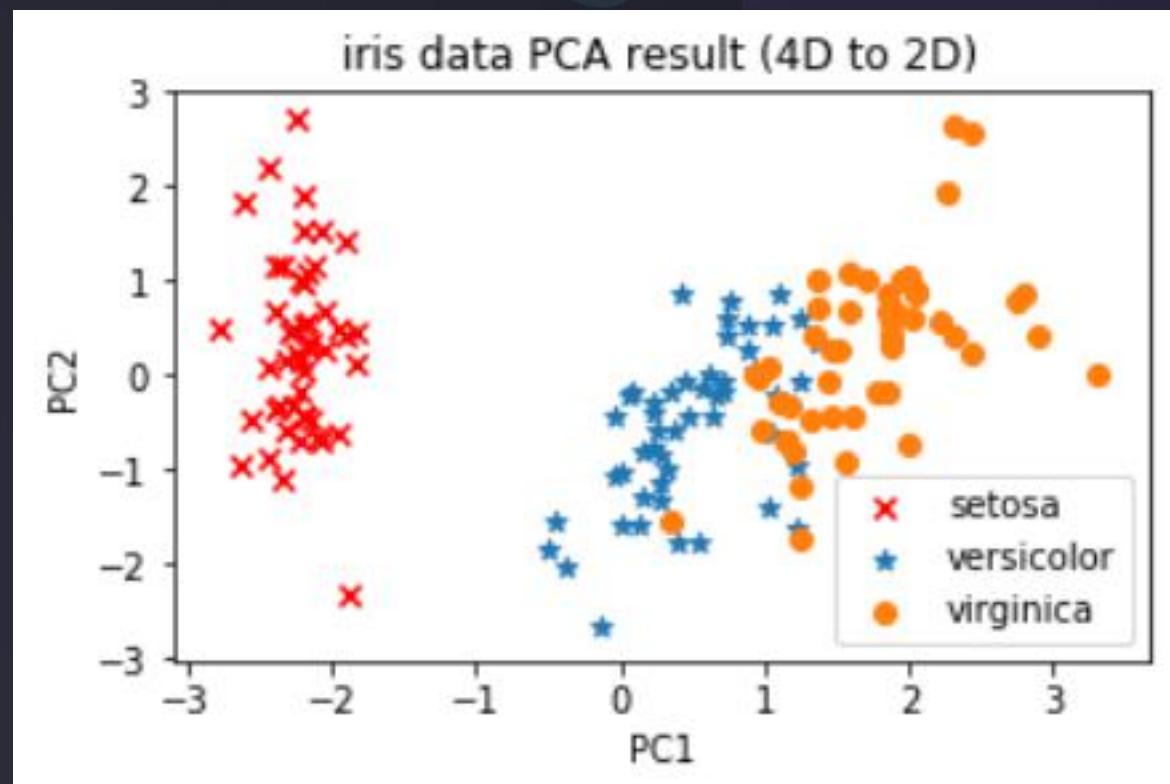
3维到2维：
投影到 u_1 、 u_2 形成的平面

n维到k维：
投影到 u_1 、 u_2 ... u_k 形成的空间

主成分分析 (PCA)

iris 鸢尾花数据经过PCA降维后的结果

萼片长	萼片宽	花瓣长	花瓣宽	类别
6.1	2.8	4.7	1.2	Iris-versicolor
5.8	2.8	5.1	2.4	Iris-virginica
5	3.4	1.5	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
6.7	3.3	5.7	2.1	Iris-virginica
5.5	2.6	4.4	1.2	Iris-versicolor
5.1	3.4	1.5	0.2	Iris-setosa
4.4	3.2	1.3	0.2	Iris-setosa
6.8	2.8	4.8	1.4	Iris-versicolor
5	3.5	1.6	0.6	Iris-setosa



核心：投影后的数据尽可能分得开（即不相关）

知识巩固

问题：我们常认为信息越多越有助于做出正确判断，在机器学习过程中，数据特征信息在很多、很少的情况下分别会导致什么问题，如何解决这些问题？



Python3人工智能入门+实战提升：机器学习

Chapter 6 异常检测与数据降维

赵辛

Chapter 6 异常检测与数据降维

-
- 1 --异常检测（一）
 - 2 --异常检测（二）
 - 3 --主成分分析（PCA）
 - 4 --实战准备
 - 5 --实战（一）异常消费检测
 - 6 --实战（二）PCA降维之糖尿病检测

任务一：异常消费行为检测

基于task1_data数据，基于高斯分布的概率密度函数实现异常消费行为检测。

平均消费次数	总消费金额
5.0	197.9
5.1	198.0
5.1	194.8
5.0	189.6
4.8	189.5
4.8	194.7
4.8	201.0
5.1	202.1
5.1	199.0
4.8	184.4

- 1、可视化消费数据、数据分布次数、及其对应高斯分布的概率密度函数；
 - 2、设置概率密度阈值0.03，建立模型，实现异常数据点预测
 - 3、可视化异常检测处理结果
 - 4、修改概率密度为0.1、0.2，查看阈值改变对结果的影响
- 能力拓展：修改概率密度阈值为0-0.2，以0.01为递增间隔，查看保存结果、生成动态gif图

异常消费行为检测

数据加载及展示

数据预处理

模型建立及训练

模型预测

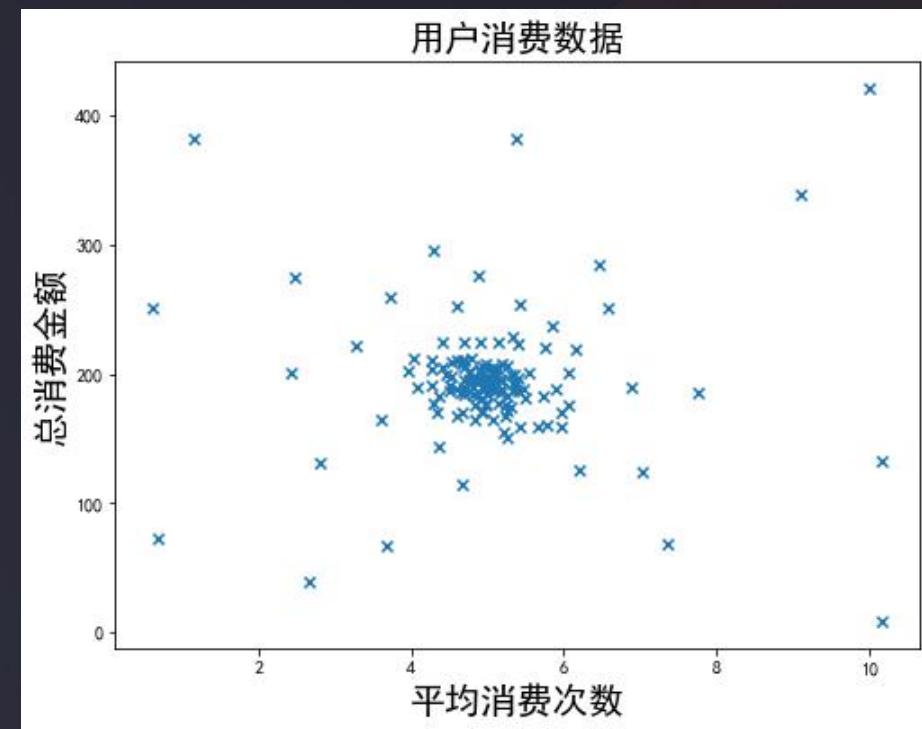
结果展示

#数据可视化

```
from matplotlib import pyplot as plt
```

```
fig1 = plt.figure(figsize=(8,6))
```

```
plt.scatter(data.loc[:, 'frequency'], data.loc[:, 'payment'], marker='x')
```



异常消费行为检测

数据加载及展示

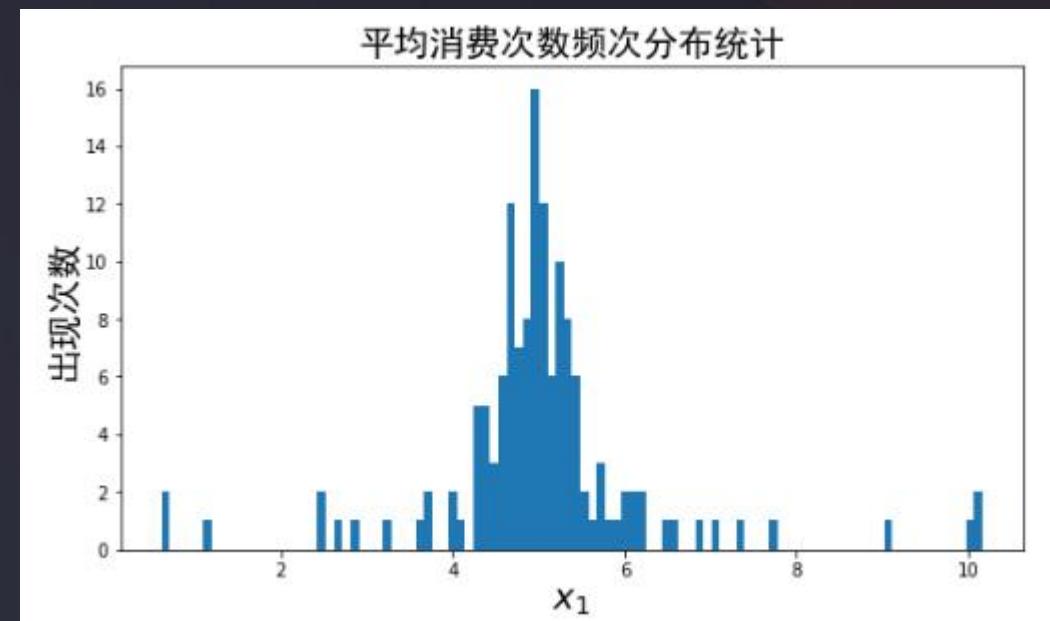
数据预处理

模型建立及训练

模型预测

结果展示

```
#数据赋值  
x1 = data.loc[:, 'frequency']  
x2 = data.loc[:, 'payment']  
#数据分布频次图  
fig2 = plt.figure(figsize=(20,5))  
plt.hist(x1, bins=100)
```



异常消费行为检测

计算数据均值、标准差：

```
x1_mean = x1.mean()  
x1_sigma = x1.std()
```

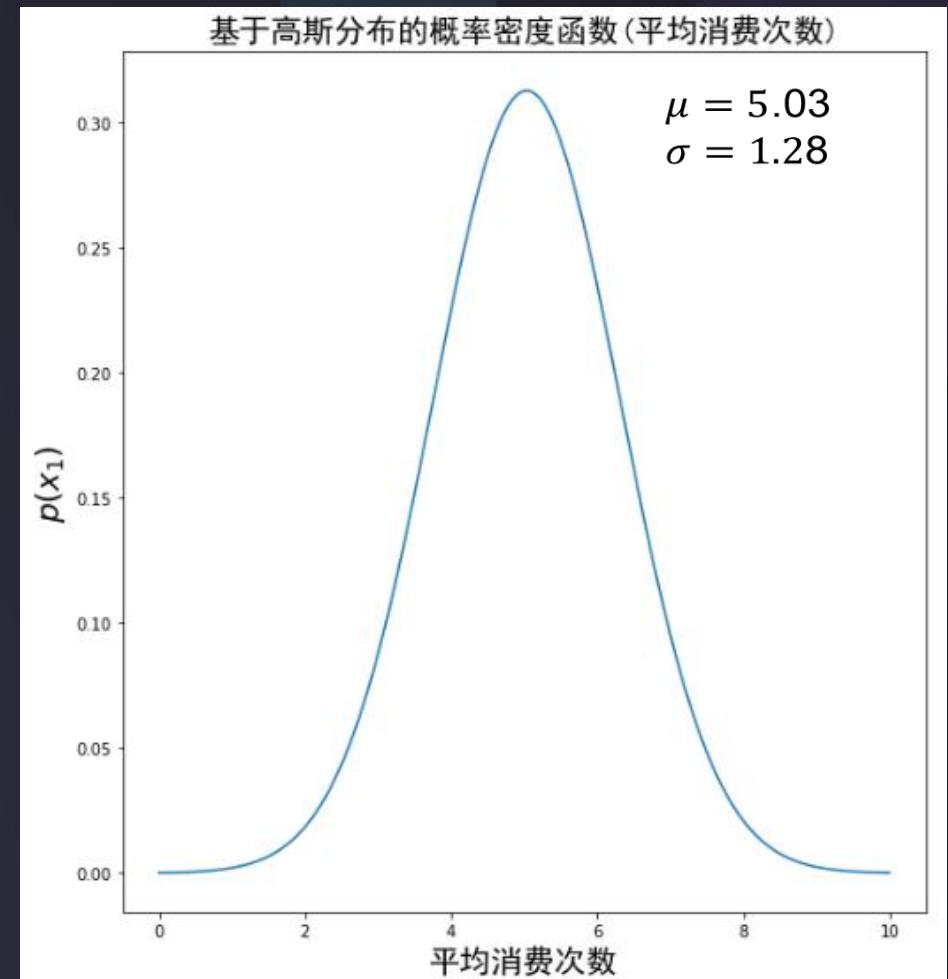
计算对应的高斯分布数值：

```
from scipy.stats import norm  
x1_range = np.linspace(0,10,300)  
normal1 = norm.pdf(x1_range,  
x1_mean, x1_sigma)
```

可视化高斯分布曲线：

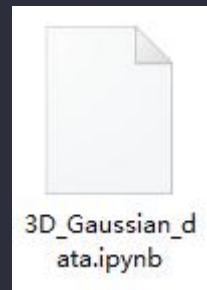
```
plt.plot(x1_range,normal1)
```

<https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.norm.html>



| 异常消费行为检测

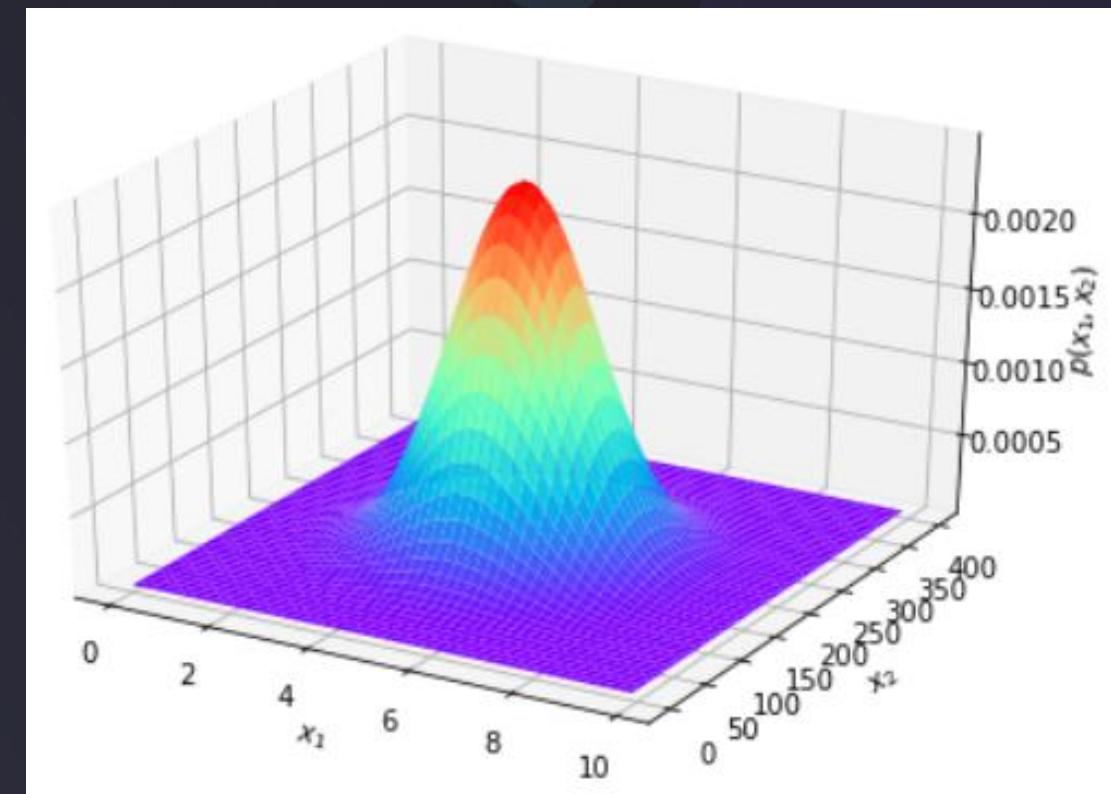
#生成用于绘制3D高斯分布图的数据：



#3D高斯分布概率密度函数图形可视化

```
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
fig = plt.figure()
axes3d = Axes3D(fig)
```

```
axes3d.plot_surface(xx,yy,p_2d,cmap=cm.rainbow)
```



异常消费行为检测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

#模型建立及训练

```
from sklearn.covariance import EllipticEnvelope  
ad_model = EllipticEnvelope(contamination=0.03)  
ad_model.fit(data)
```

EllipticEnvelope(contamination=0.03)

#模型预测

```
y_predict = ad_model.predict(data)  
print(pd.value_counts(y_predict))
```

1	137
-1	5
dtype: int64	

异常消费行为检测

数据加载及展示

数据预处理

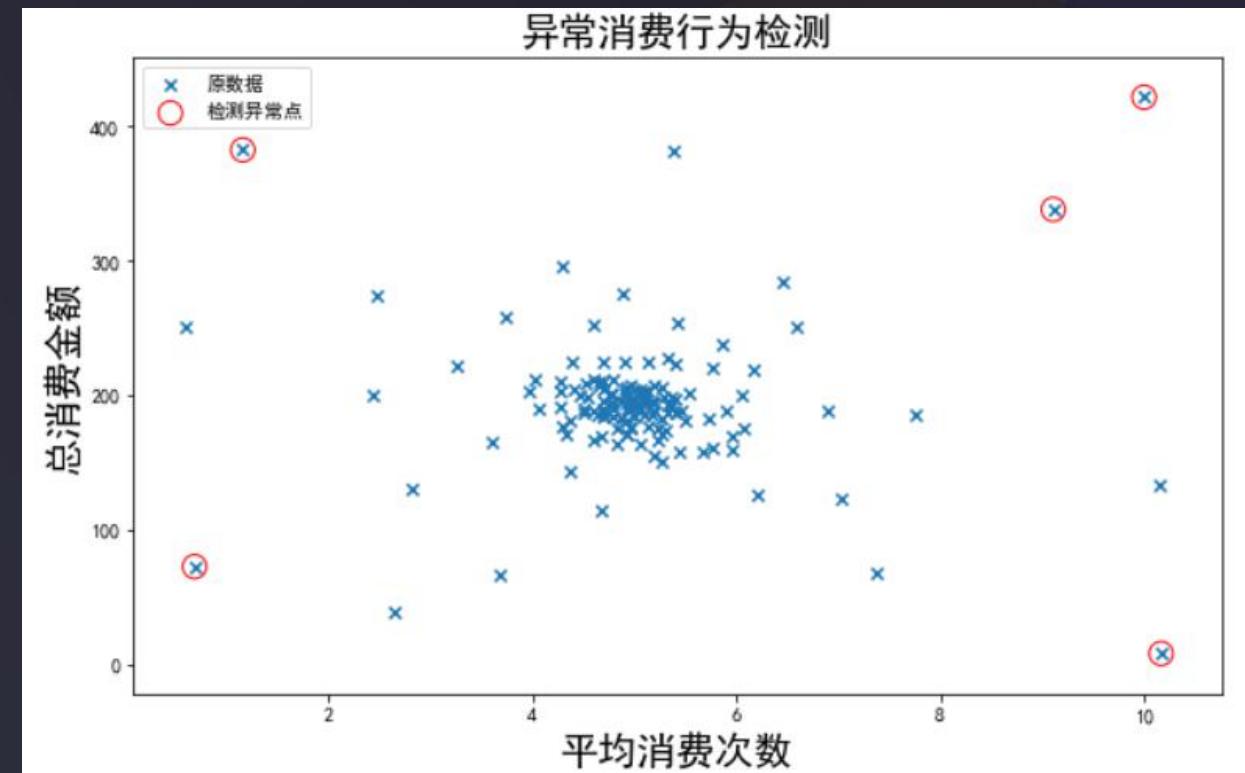
模型建立及训练

模型预测

结果展示

#异常数据可视化

```
anomaly_data=plt.scatter(data.loc[:, 'frequency'][y_predict== -1], data.loc[:, 'payment'][y_predict== -1], marker='o', facecolor='none', edgecolor='red', s=150)
```



任务二：PCA+逻辑回归预测检查者是否患糖尿病

基于task2_data数据，结合PCA降维技术与逻辑回归预测检查者患病情况。

pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
9	122	56	0	0	33.3	1.114	33	1
2	112	66	22	0	25	0.307	24	0
1	103	30	38	83	43.3	0.183	33	0
1	138	82	0	0	40.1	0.236	28	0
9	72	78	25	0	31.6	0.28	38	0
4	76	62	0	0	34	0.391	25	0
3	163	70	18	105	31.6	0.268	28	1
0	94	0	0	0	0	0.256	25	0
5	77	82	41	42	35.8	0.156	35	0
1	88	30	42	99	55	0.496	26	1
1	81	72	18	40	26.6	0.283	24	0
13	152	90	33	29	26.8	0.731	43	1

- 1、对原数据建立逻辑回归模型，计算模型预测准确率；
- 2、对数据进行标准化处理，选取glucose维度数据可视化处理后的效果；
- 3、进行与原数据等维度PCA，查看各主成分的方差比例；
- 4、保留2个主成分，可视化降维后的数据；
- 5、基于降维后数据建立逻辑回归模型，与原数据表现进行对比，思考结果变化原因

PCA+逻辑回归预测检查者是否患糖尿病

数据加载及展示

数据预处理

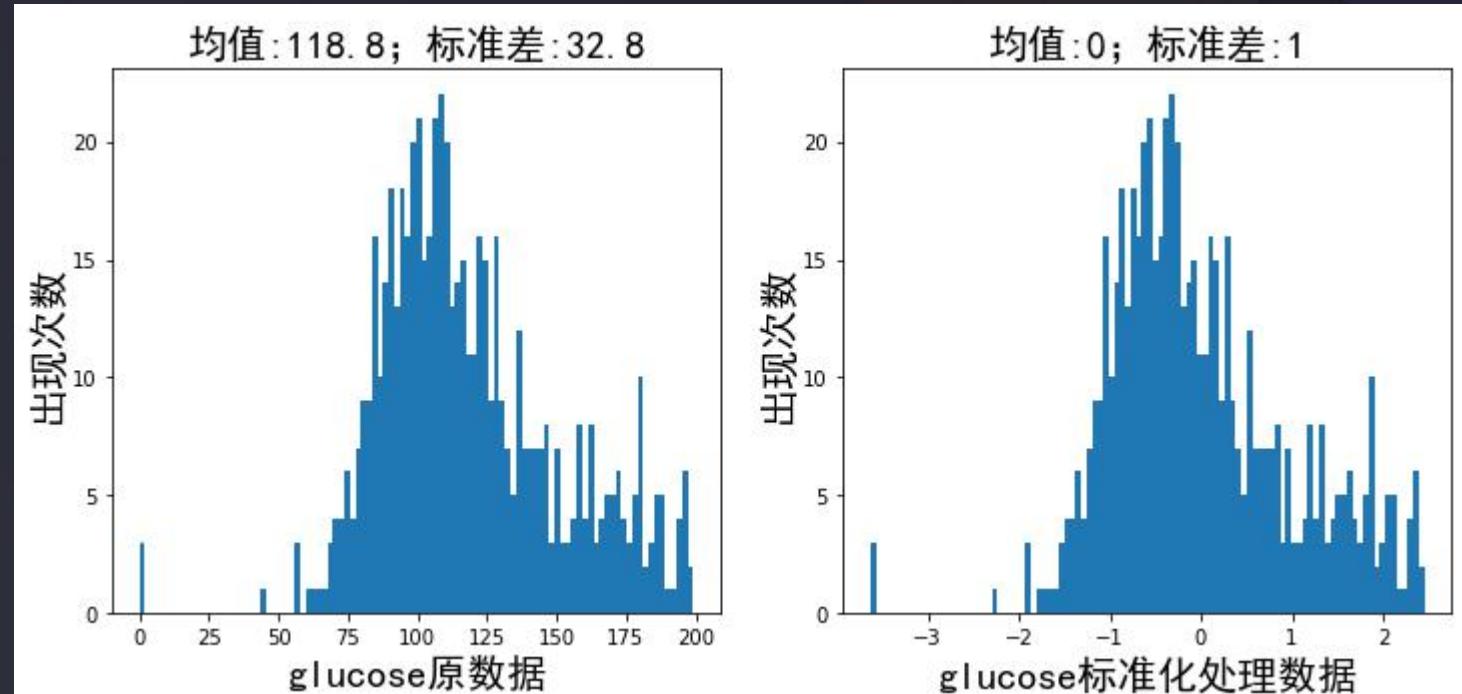
PCA降维

模型训练与预测

结果展示及表现评估

#数据标准化处理

```
from sklearn.preprocessing import StandardScaler  
X_norm = StandardScaler().fit_transform(X)  
print(X_norm)
```



PCA+逻辑回归预测检查者是否患糖尿病

数据加载及展示

数据预处理

PCA降维

模型训练与预测

结果展示及表现评估

```
#pca分析
from sklearn.decomposition import PCA
pca = PCA(n_components=8)
X_pca = pca.fit_transform(X_norm)
#计算各成分投影数据方差比例
var = pca.explained_variance_
var_ratio = pca.explained_variance_ratio_
print(var)
print(var_ratio)
```

```
[2.15669029 1.74035262 1.03817655 0.87890467 0.75186028 0.61865346
 0.42446137 0.40425636]
[0.26913698 0.2171815  0.12955578 0.10967998 0.0938259  0.0772028
 0.05296924 0.05044782]
```

PCA+逻辑回归预测检查者是否患糖尿病

数据加载及展示

数据预处理

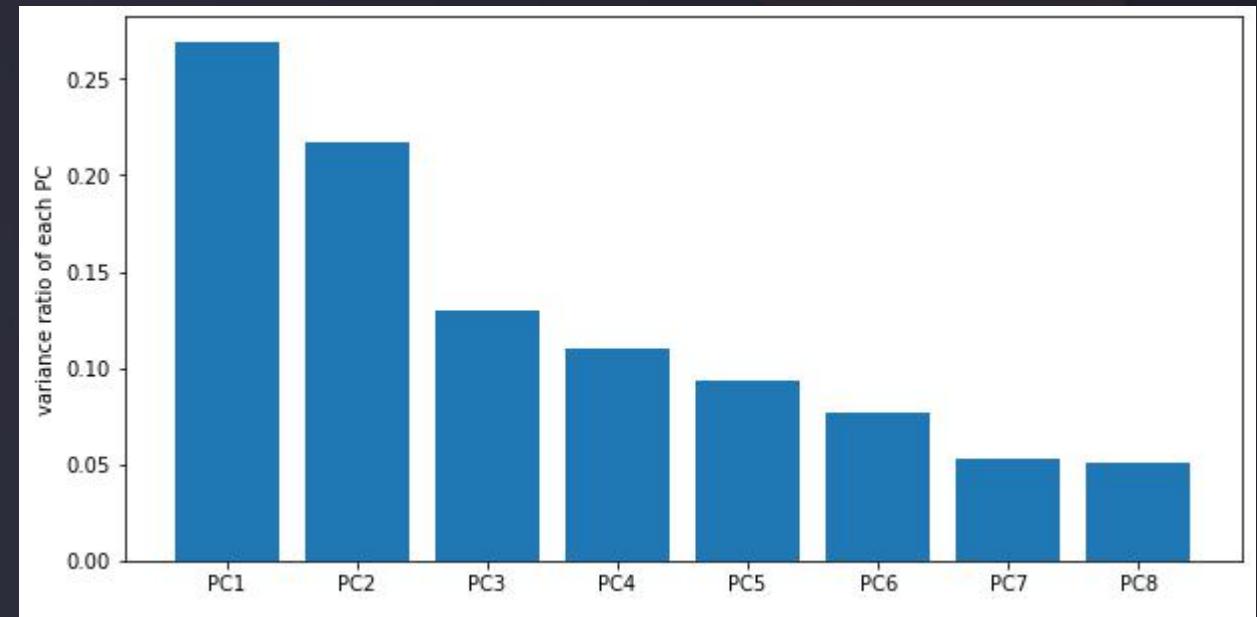
PCA降维

模型训练与预测

结果展示及表现评估

#可视化方差比例

```
fig2 = plt.figure(figsize=(20,5))
plt.bar([1,2,3,4,5,6,7,8],var_ratio)
plt.xticks([1,2,3,4,5,6,7,8],['PC1','PC2','PC3','PC4','PC5',
'PC6','PC7','PC8'])
plt.ylabel('variance ratio of each PC')
plt.show()
```



PCA+逻辑回归预测检查者是否患糖尿病

数据加载及展示

数据预处理

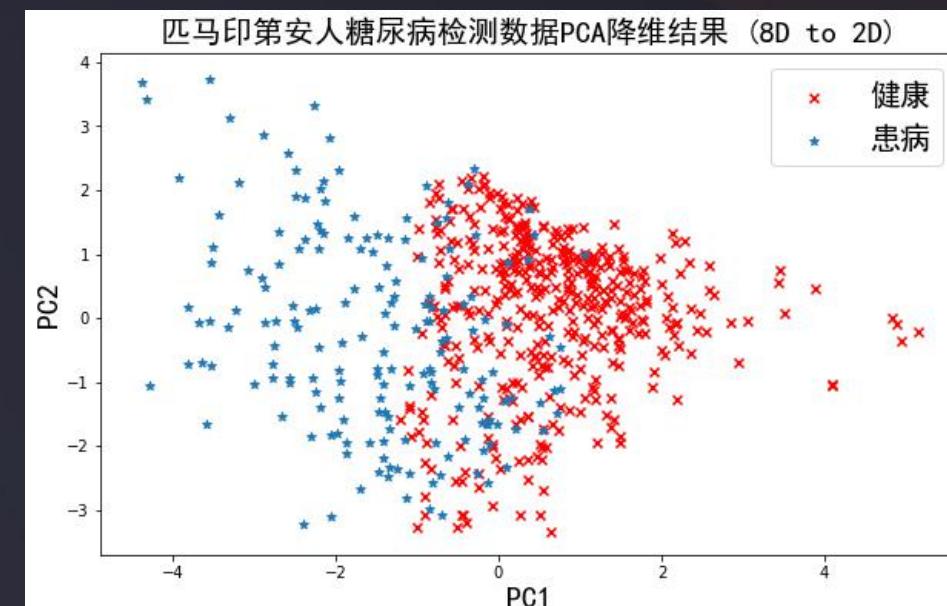
PCA降维

模型训练与预测

结果展示及表现评估

```
#数据降维到2维  
pca = PCA(n_components=2)  
X_pca = pca.fit_transform(X_norm)  
X_pca.shape
```

```
#可视化降维后的数据  
negative=plt.scatter(X_pca[:,0][y==0],X_pca[:,1][y==0],c='r',marker='x')  
positive=plt.scatter(X_pca[:,0][y==1],X_pca[:,1][y==1],marker='*')
```



PCA+逻辑回归预测检查者是否患糖尿病

数据加载及展示

数据预处理

PCA降维

模型训练与预测

结果展示及表现评估

```
#降维后逻辑回归预测  
logreg2 = LogisticRegression()  
logreg2.fit(X_pca, y)  
y_pca_p = logreg2.predict(X_pca)  
#计算准确率  
print(metrics.accuracy_score(y, y_pca_p))
```

原数据模型预测准确率: 0.92



PCA降维后模型预测准确率: 0.88



Python3人工智能入门+实战提升：机器学习

Chapter 6 异常检测与数据降维

赵辛

Chapter 6 异常检测与数据降维

-
- 1 --异常检测（一）
 - 2 --异常检测（二）
 - 3 --主成分分析（PCA）
 - 4 --实战准备
 - 5 --实战（一）异常消费检测
 - 6 --实战（二）PCA降维之糖尿病检测

任务一：异常消费行为检测

基于ad_data数据，基于高斯分布的概率密度函数实现异常消费行为检测。

平均消费次数	总消费金额
5.0	197.9
5.1	198.0
5.1	194.8
5.0	189.6
4.8	189.5
4.8	194.7
4.8	201.0
5.1	202.1
5.1	199.0
4.8	184.4

- 1、可视化消费数据、数据分布次数、及其对应高斯分布的概率密度函数；
- 2、设置概率密度阈值0.03，建立模型，实现异常数据点预测
- 3、可视化异常检测处理结果
- 4、修改概率密度为0.1、0.2，查看阈值改变对结果的影响
- 5、能力拓展：修改概率密度阈值为0-0.2，以0.01为递增间隔，查看保存结果、生成动态gif图



Python3人工智能入门+实战提升：机器学习

Chapter 6 异常检测与数据降维

赵辛

Chapter 6 异常检测与数据降维

-
- 1 --异常检测（一）
 - 2 --异常检测（二）
 - 3 --主成分分析（PCA）
 - 4 --实战准备
 - 5 --实战（一）异常消费检测
 - 6 --实战（二）PCA降维之糖尿病检测

任务二：PCA+逻辑回归预测检查者是否患糖尿病

基于diabetes_data数据，结合PCA降维技术与逻辑回归预测检查者患病情况。

pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
9	122	56	0	0	33.3	1.114	33	1
2	112	66	22	0	25	0.307	24	0
1	103	30	38	83	43.3	0.183	33	0
1	138	82	0	0	40.1	0.236	28	0
9	72	78	25	0	31.6	0.28	38	0
4	76	62	0	0	34	0.391	25	0
3	163	70	18	105	31.6	0.268	28	1
0	94	0	0	0	0	0.256	25	0
5	77	82	41	42	35.8	0.156	35	0
1	88	30	42	99	55	0.496	26	1
1	81	72	18	40	26.6	0.283	24	0
13	152	90	33	29	26.8	0.731	43	1

- 1、对原数据建立逻辑回归模型，计算模型预测准确率；
- 2、对数据进行标准化处理，选取glucose维度数据可视化处理后的效果；
- 3、进行与原数据等维度PCA，查看各主成分的方差比例；
- 4、保留2个主成分，可视化降维后的数据；
- 5、基于降维后数据建立逻辑回归模型，与原数据表现进行对比，思考结果变化原因

| 任务三：为毕业与工作做准备

待添加?



Python3人工智能入门+实战提升：机器学习

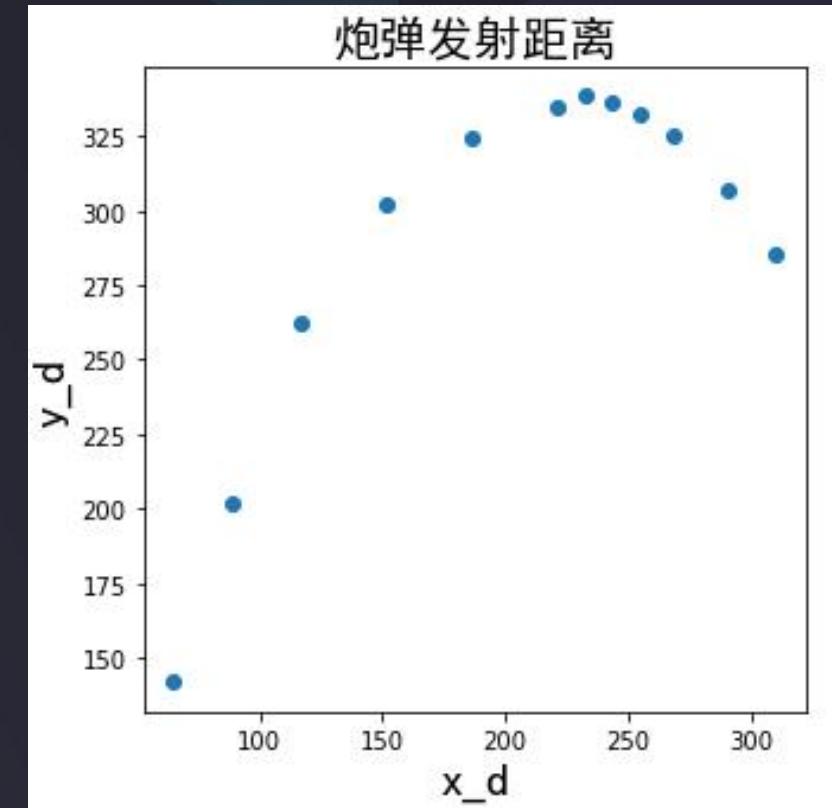
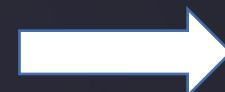
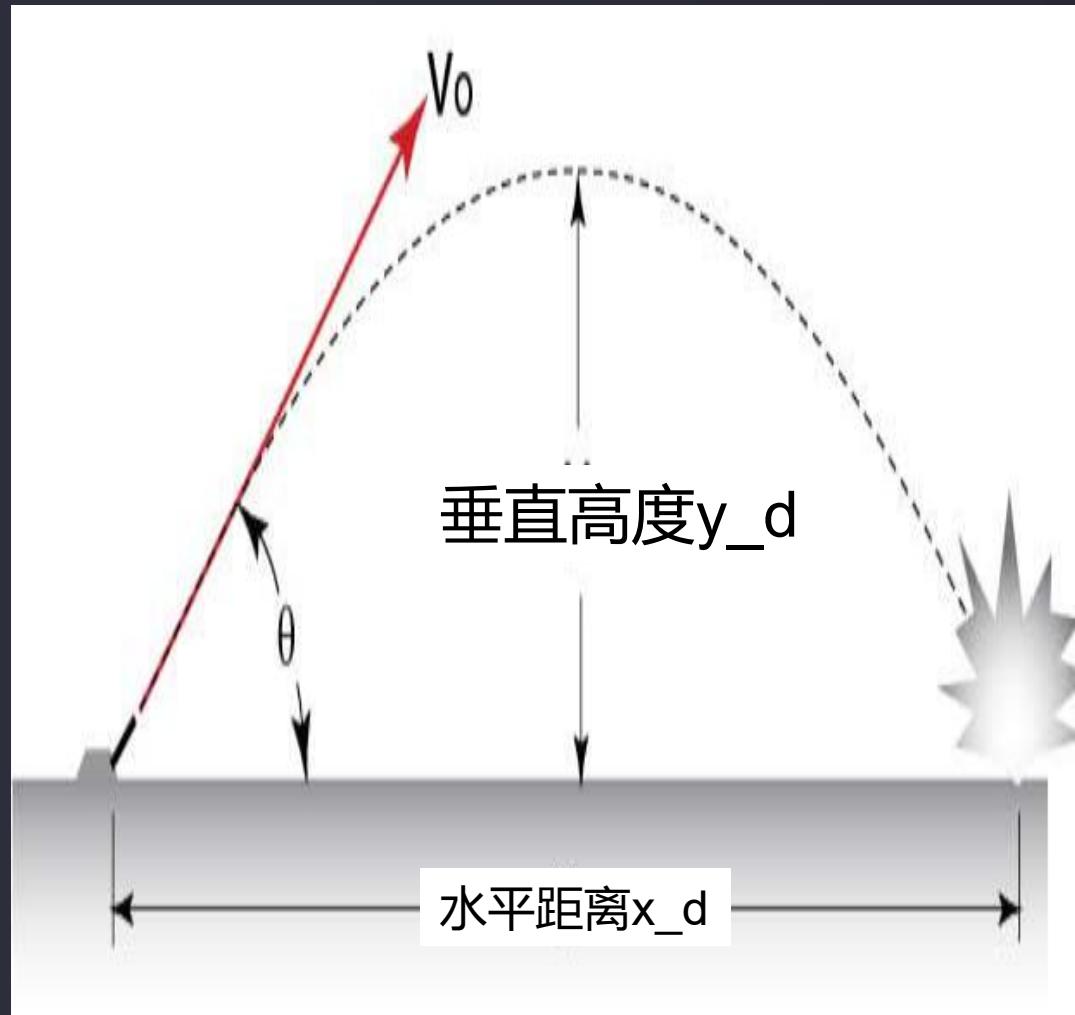
Chapter 7 综合能力提升之模型选择与优化

赵辛

Chapter 7 综合能力提升之模型选择与优化

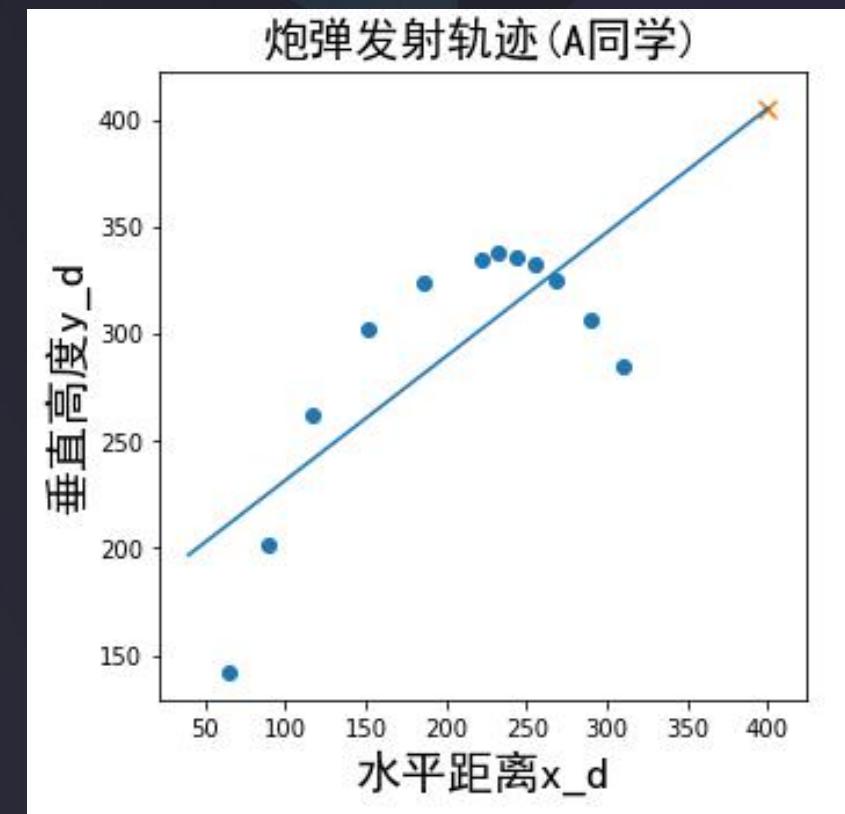
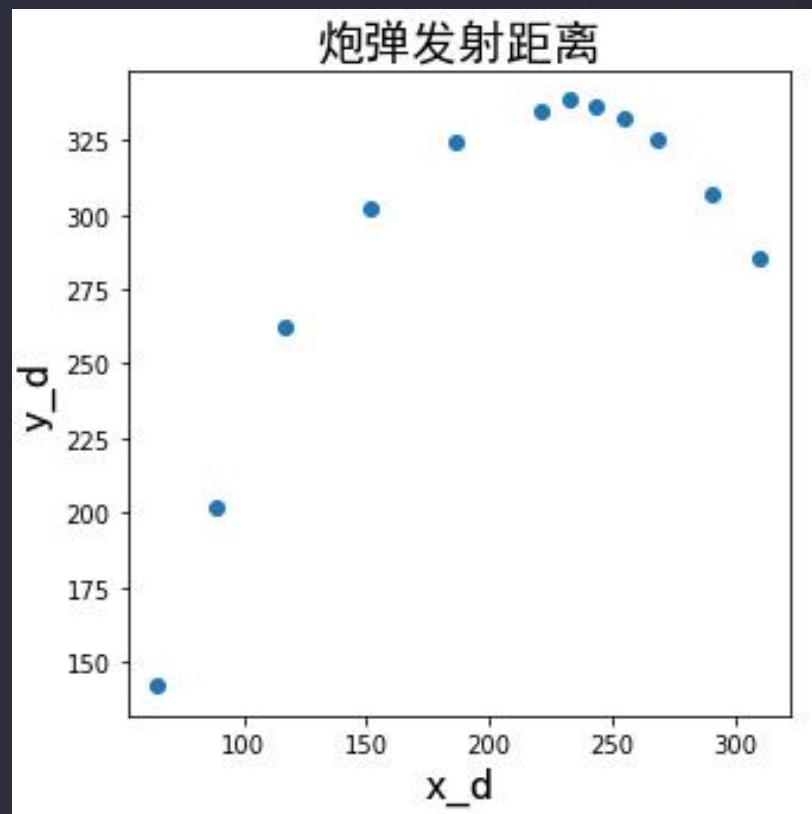
-
- 1 --模型过拟合与欠拟合
 - 2 --模型过拟合解决
 - 3 --数据分离与混淆矩阵
 - 4 --模型选择与优化
 - 5 --实战准备
 - 6 --实战（一）综合提升之炮弹发射轨迹预测
 - 7 --实战（二）综合提升之芯片品质预测

现实问题思考



根据收集到的部分炮弹发射数据，
推测在 $x_d = 400$ 时，炮弹高度 y_d

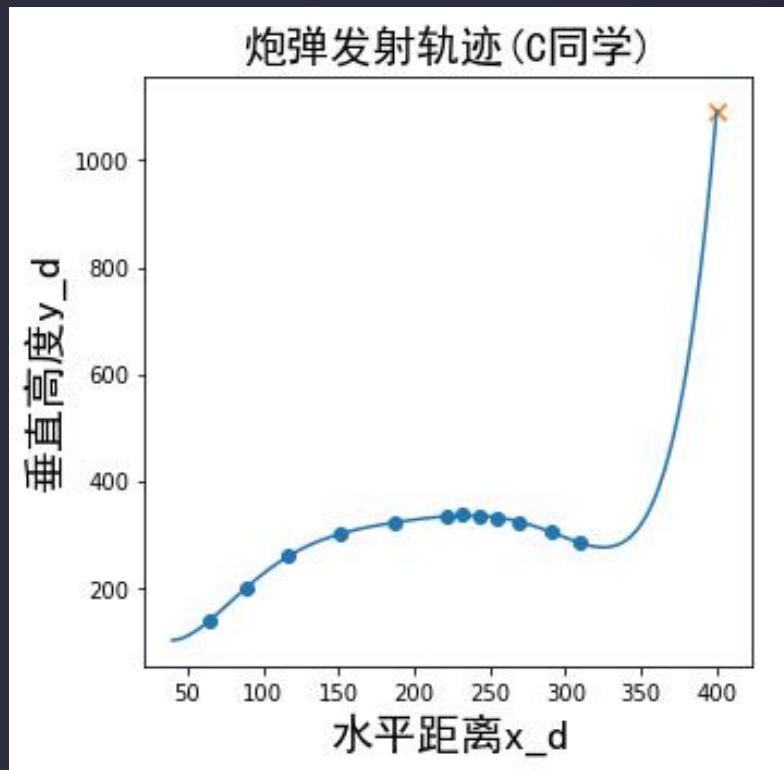
现实问题思考



根据收集到的部分炮弹发射数据，
推测在 $x_d=400$ 时，炮弹高度 y_d

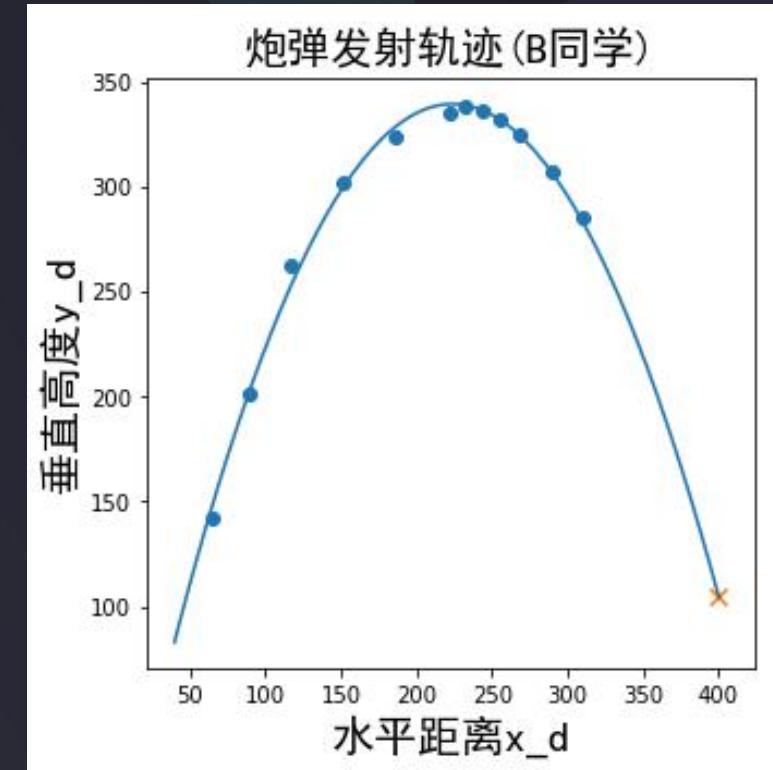
$$y_d = \theta_0 + \theta_1 x_d \quad \rightarrow \quad y_d(x_d=400) = 405$$

现实问题思考



$$y_d = \theta_0 + \theta_1 x_d + \theta_2 x_d^2 + \theta_3 x_d^3 + \theta_4 x_d^4 + \theta_5 x_d^5 + \theta_6 x_d^6$$

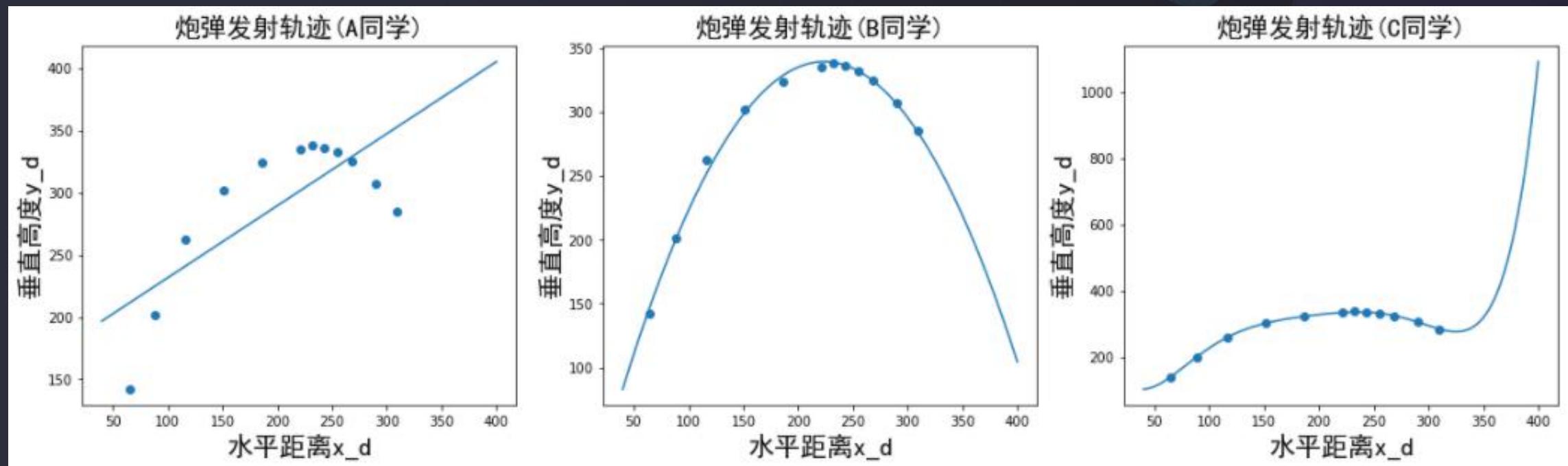
→ $y_d(x_d=400) = 1092$



$$y_d = \theta_0 + \theta_1 x_d + \theta_2 x_d^2$$

→ $y_d(x_d=400) = 104$

现实问题思考



欠拟合结果

训练数据、预测数据效果都不好

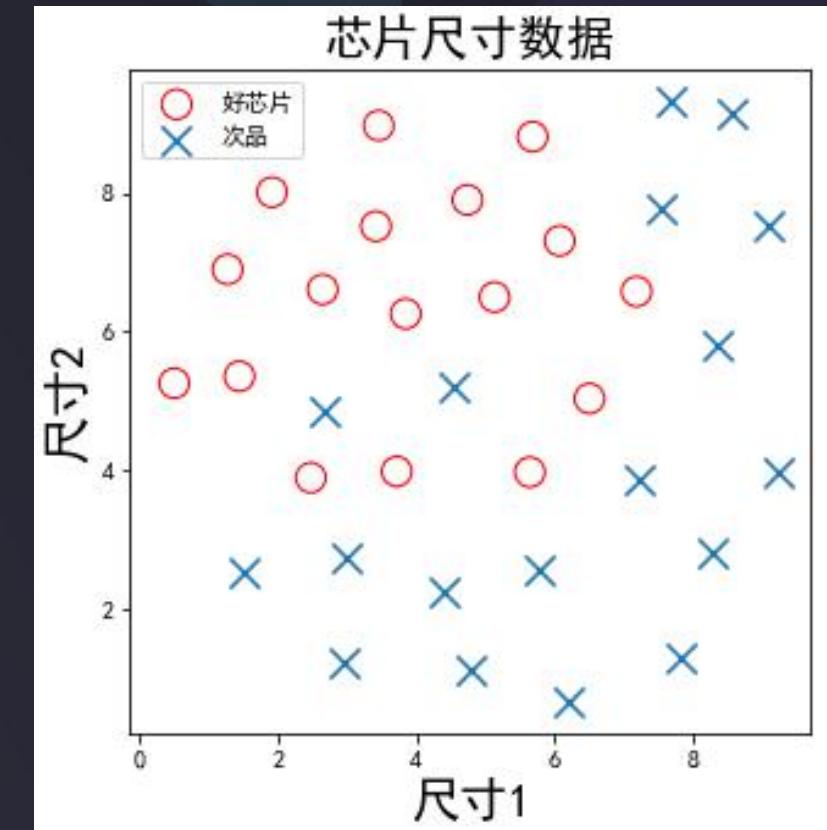
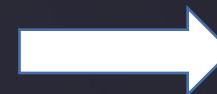
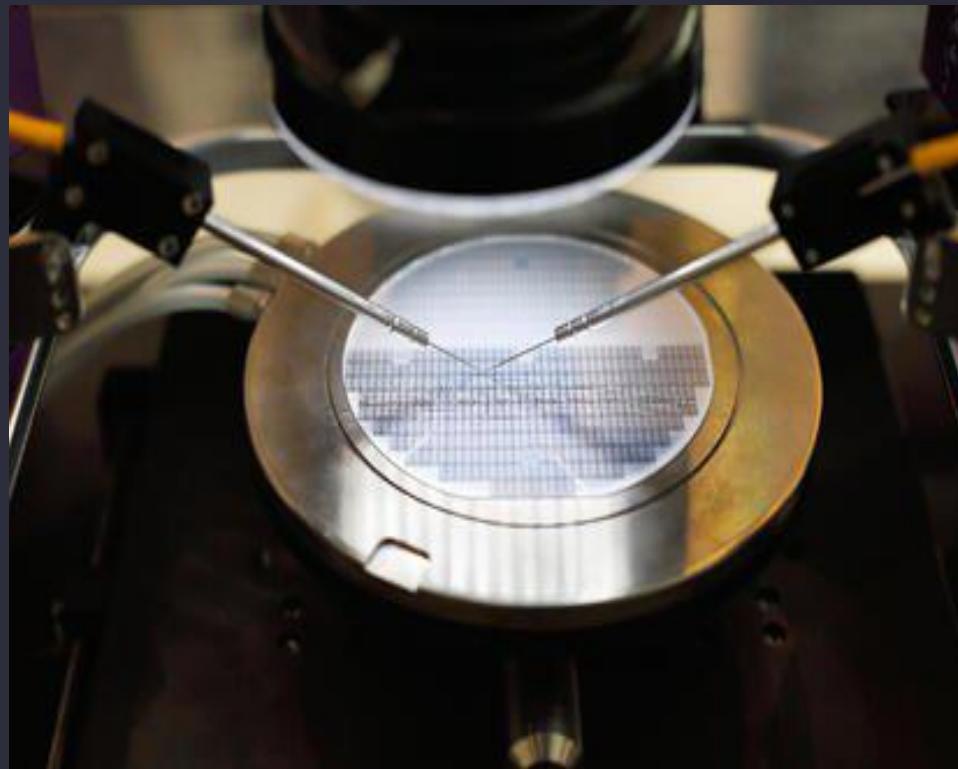
好模型结果

训练数据、预测数据效果都很不错

过拟合结果

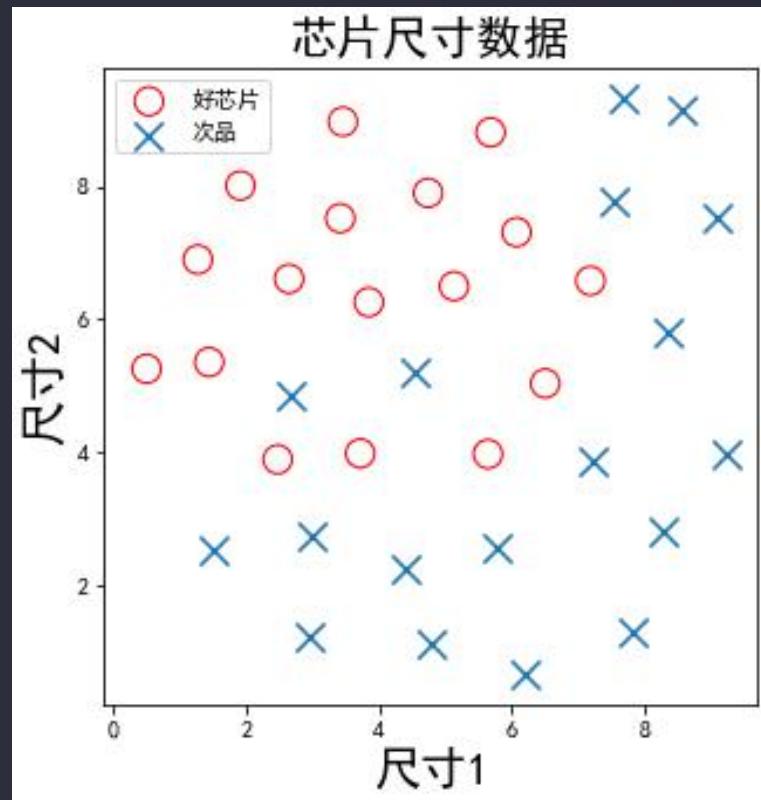
训练数据效果很好、但预测数据效果不好

现实问题思考

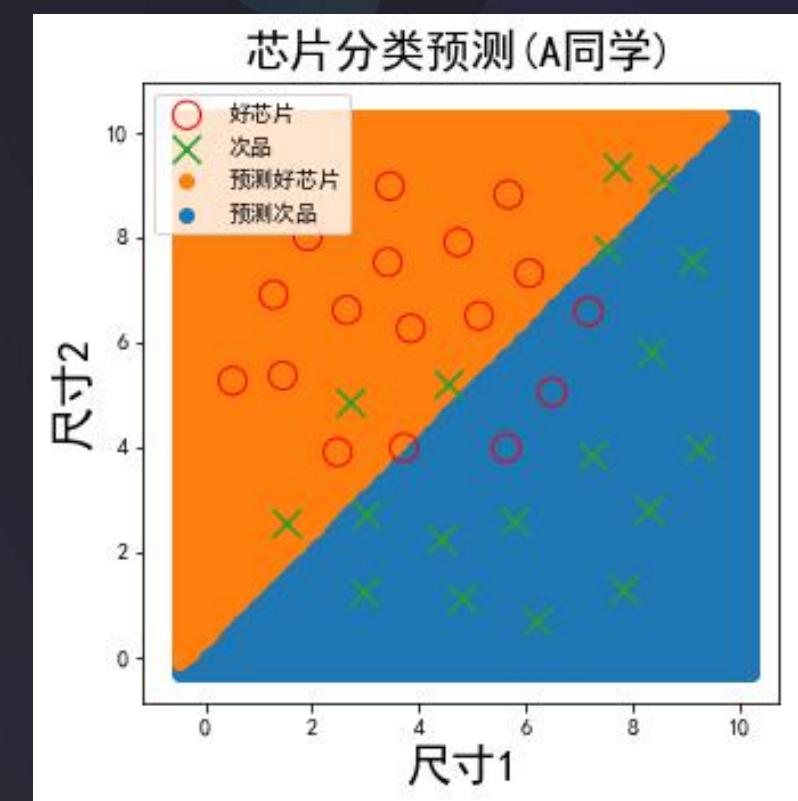


根据芯片尺寸1、尺寸2参数识别次品

现实问题思考



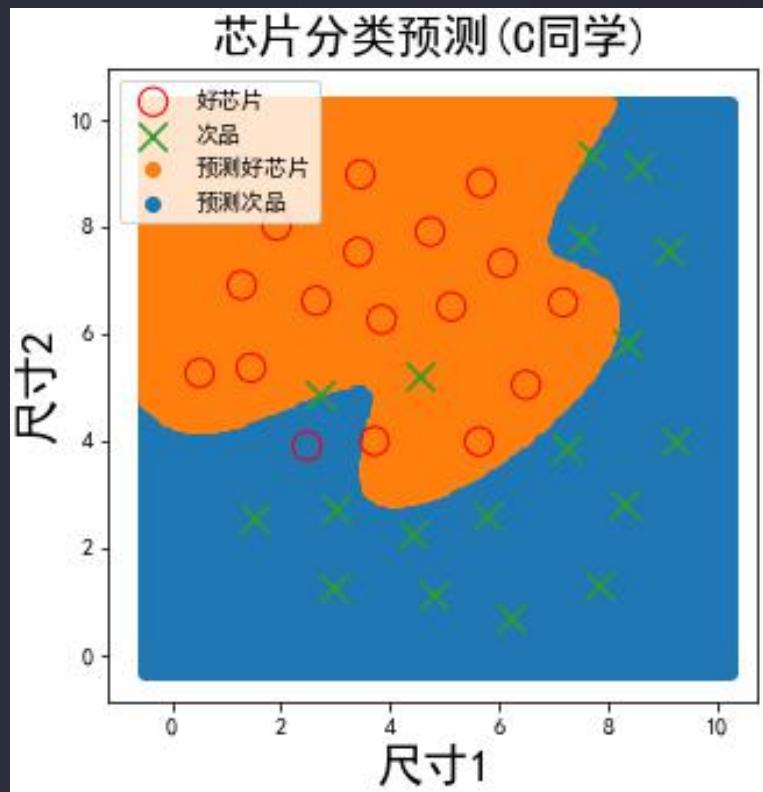
根据芯片尺寸1、尺寸2参数识别次品



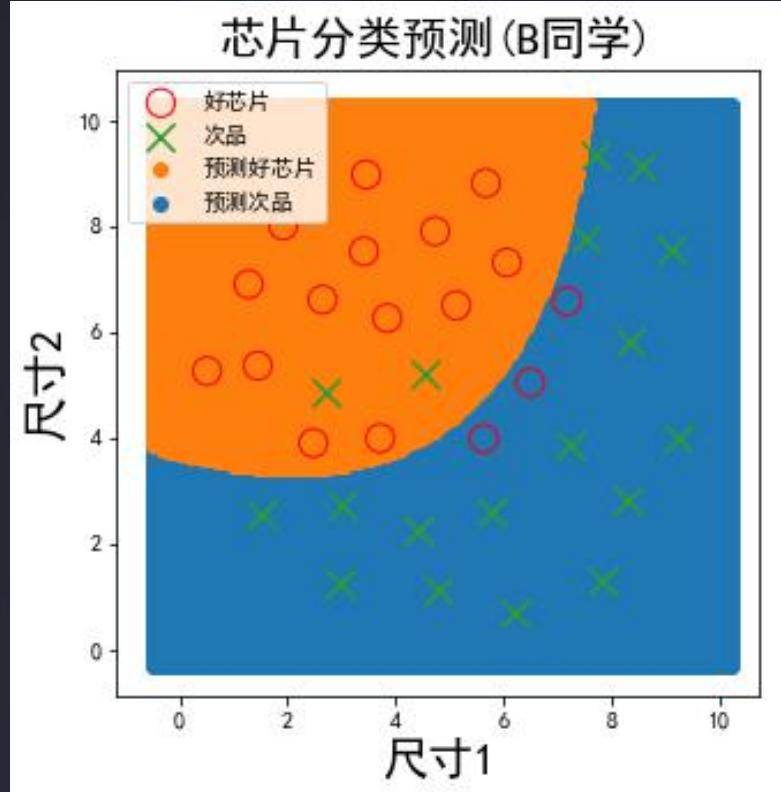
线性决策边界： $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$

现实问题思考

准确率：94.3%



准确率：85.7%



四阶决策边界：

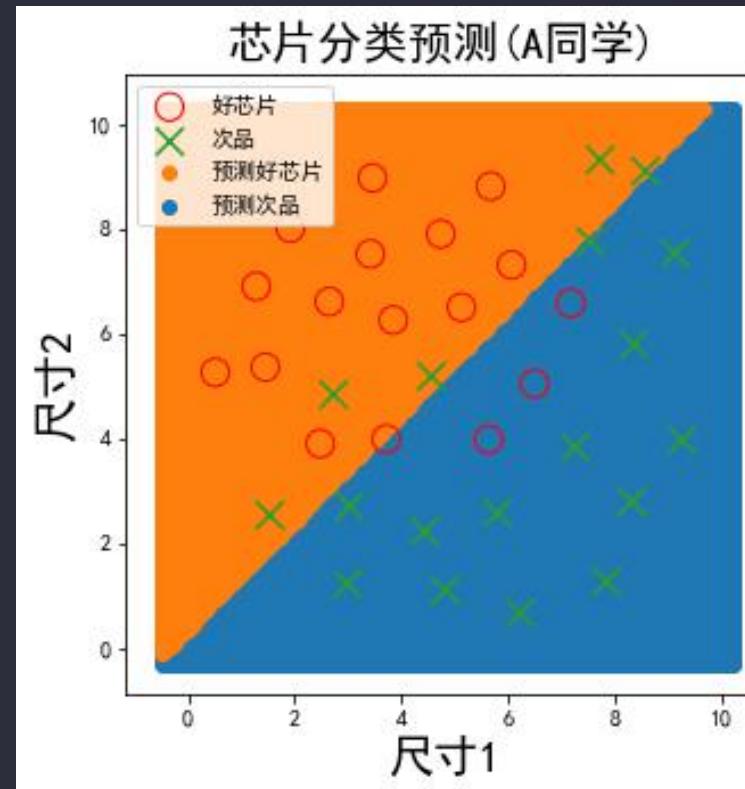
$$f\left((\theta_0 \dots \theta_{14}), (x_1 \dots x_1^4), (x_2 \dots x_2^4)\right) = 0$$

二阶决策边界：

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 = 0$$

现实问题思考

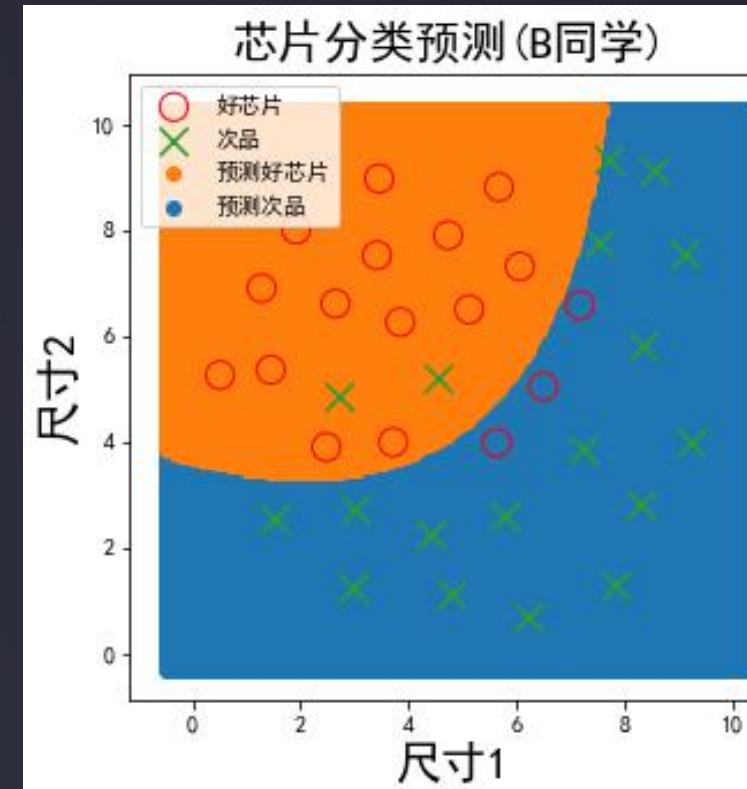
准确率: 77.1%



欠拟合结果

训练数据、预测数据效
果都不好

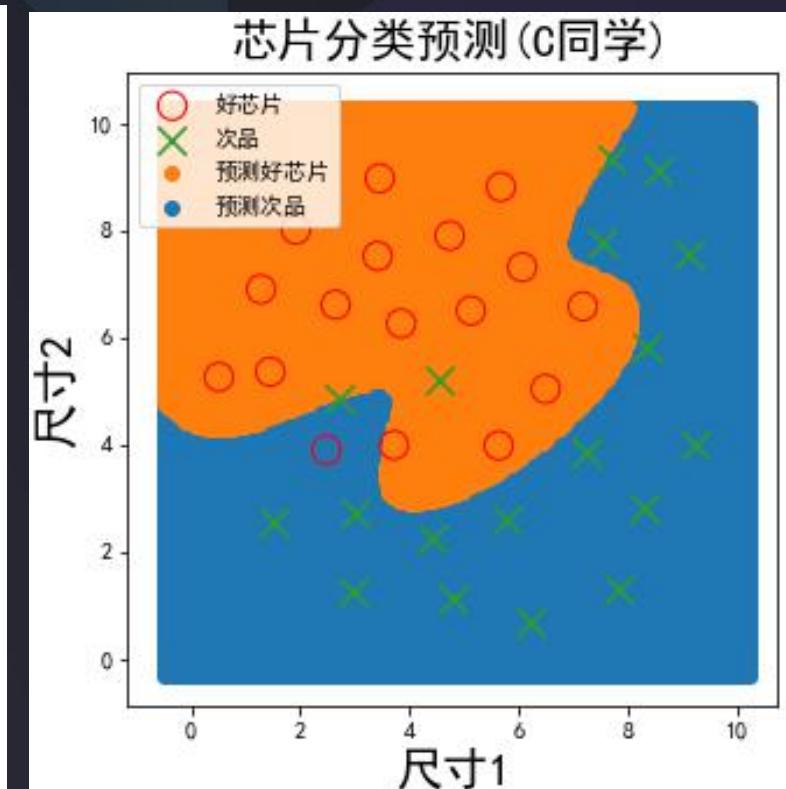
准确率: 85.7%



好模型结果

训练数据、预测数据效
果都很不错

准确率: 94.3%



过拟合结果

训练数据效果很好、
但预测数据效果不好

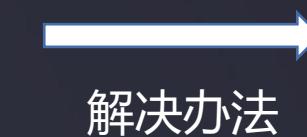
欠拟合与过拟合

由于模型不合适，致使其无法对数据进行准确的预测。

模型对数据的预测情况

	训练数据	预测数据
欠拟合	不准确	不准确
过拟合	准确	不准确
合适模型	准确	准确

通常来说，欠拟合可通过观察训练数据的预测结果发现



选用其他模型、增加模型复杂度、增加数据样本、采集新的维度数据

| 解决过拟合问题的方法

原因：

- 使用了过于复杂的模型结构（比如高阶决策边界）
- 训练数据不足，有限的训练数据（训练样本只有总体样本中的小部分、不具备代表性）
- 样本里的噪音数据干扰过大，模型学习到了噪音信息（使用过多与结果不相关属性数据）

解决办法：

- 简化模型结构（降低模型复杂度，能达到好的效果情况下尽可能选择简单的模型）
- 数据增强（按照一定的规则扩充样本数据）
- 数据预处理，保留主成分信息（数据PCA处理）
- 增加正则化项（regularization）

参考链接：

https://blog.csdn.net/dfly_zx/article/details/107954860



Python3人工智能入门+实战提升：机器学习

Chapter 7 综合能力提升之模型选择与优化

赵辛

Chapter 7 综合能力提升之模型选择与优化

-
- 1 --模型过拟合与欠拟合
 - 2 --模型过拟合解决
 - 3 --数据分离与混淆矩阵
 - 4 --模型选择与优化
 - 5 --实战准备
 - 6 --实战（一）综合提升之炮弹发射轨迹预测
 - 7 --实战（二）综合提升之芯片品质预测

| 解决过拟合问题的方法

原因：

- 使用了过于复杂的模型结构（比如高阶决策边界）
- 训练数据不足，有限的训练数据（训练样本只有总体样本中的小部分、不具备代表性）
- 样本里的噪音数据干扰过大，模型学习到了噪音信息（使用过多不相关属性数据）

解决办法：

- 简化模型结构（调小模型复杂度，能达到好的效果情况下尽可能选择简单的模型）
- 数据增强（按照一定的规则扩充样本数据）
- 数据预处理，保留主成分信息（数据PCA处理）
- 增加正则化项（regularization）

参考链接：

https://blog.csdn.net/dfly_zx/article/details/107954860

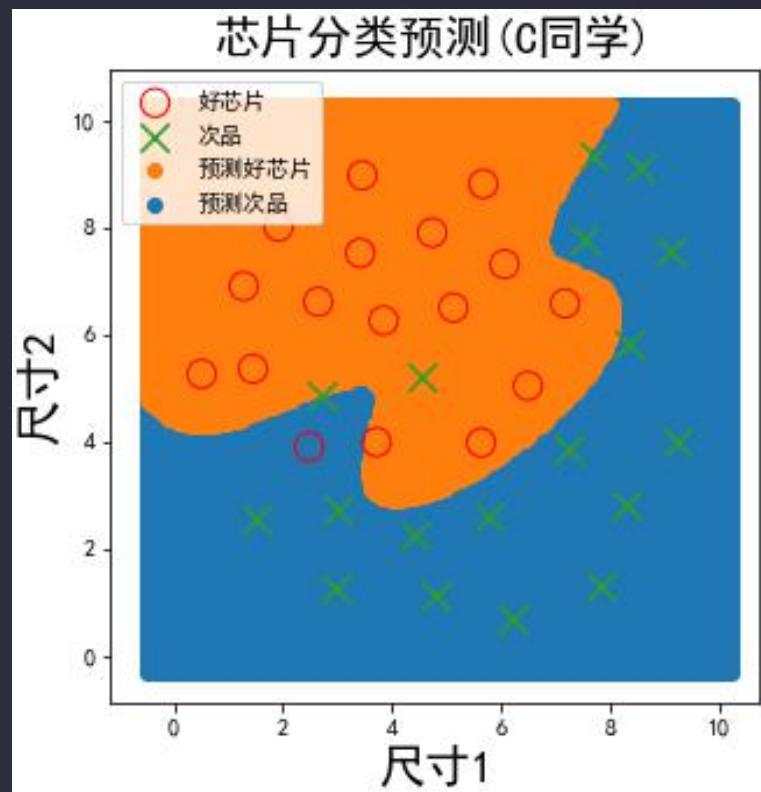
数据增强



平移、翻转、旋转、镜像...

数据PCA处理

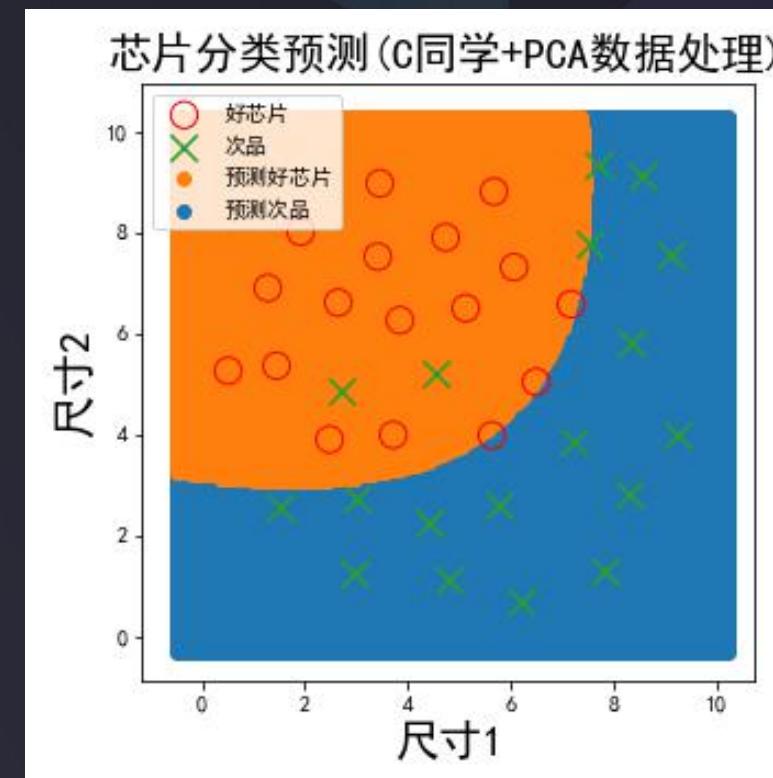
准确率：94.3%



过拟合结果

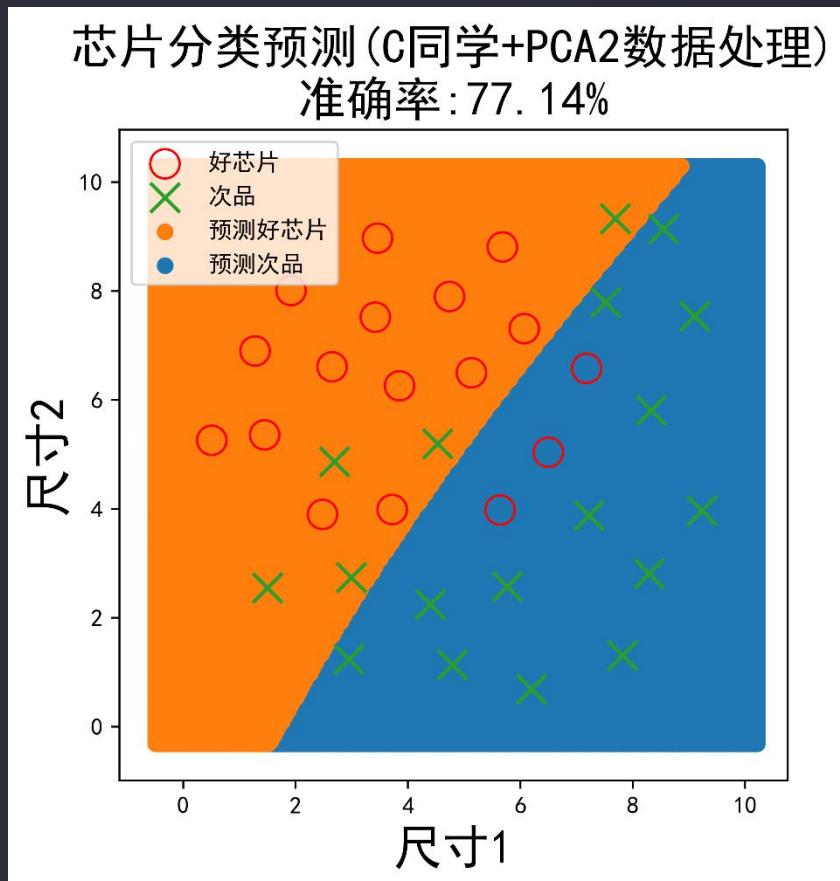
15维属性变量通过PCA降到5维

准确率：85.7%

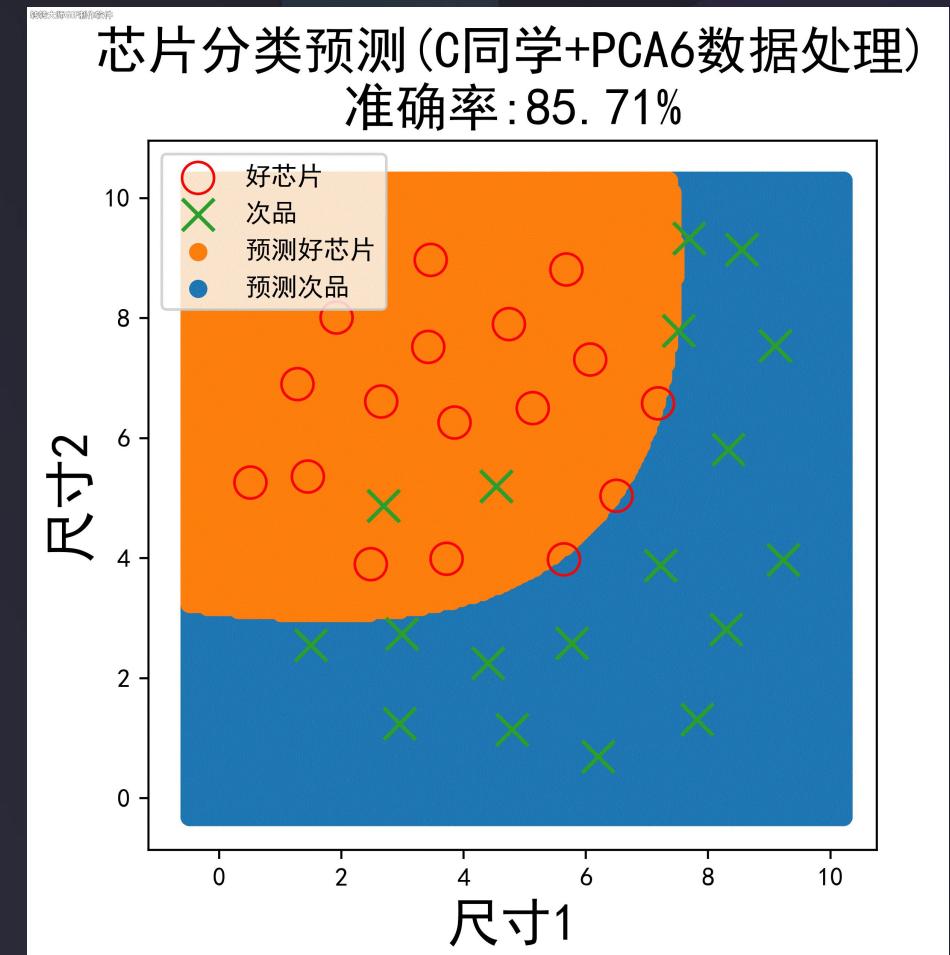


尝试降到更低维度？

数据PCA处理



降低到2维后，效果和线性边界分类接近，准确率下降明显



| 增加正则项

机器学习过程中，模型求解的核心目标就是最小化损失函数，增加正则项是指在损失函数中添加一个额外项，实现对求解参数的数值约束，防止模型过拟合。

回归任务中，损失函数 (J) :

$$J = \frac{1}{2m} \sum_{i=1}^m (y'_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^m (g(\theta, x_i) - y_i)^2$$

正则项:

$$+ \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

思考：如果 λ 是很大的数值（比如1000000），将对损失函数、 θ 造成什么影响？

增加正则项

机器学习过程中，模型求解的核心目标就是最小化损失函数，正则化是指在损失函数中添加一个额外项，实现对求解参数的数值约束，防止模型过拟合。

逻辑回归中，损失函数 (J) :

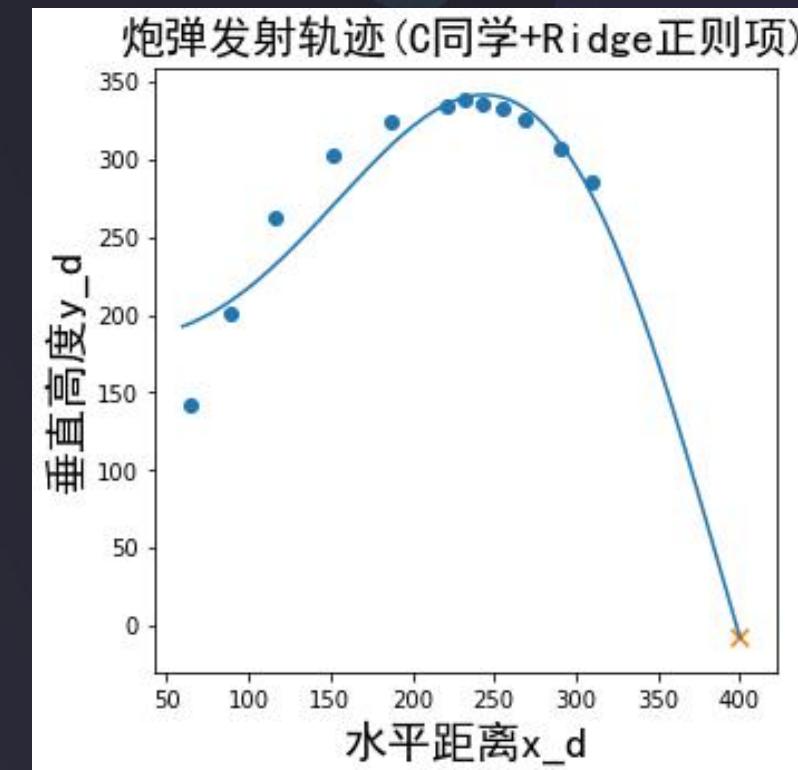
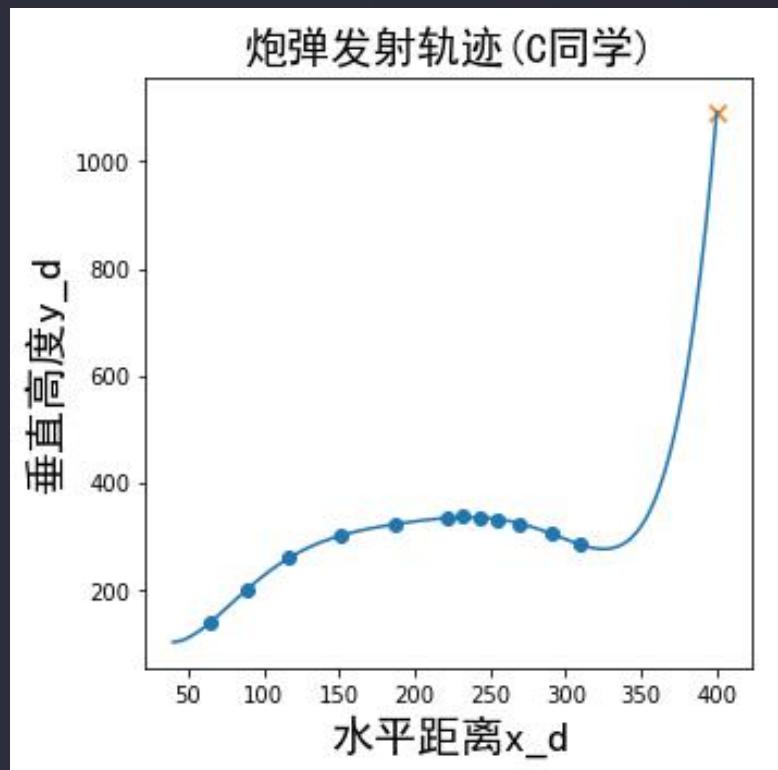
正则项:

$$J = -\frac{1}{m} \left[\sum_{i=1}^m (y_i \log(g(\theta, x_i)) + (1 - y_i) \log(1 - g(\theta, x_i))) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

如果 λ 是很大的数值（比如1000000），那各个 θ 取值就不能过大，其意义则是各个属性数据的系数受到约束（有效控制各个属性数据的影响）。

增加正则项

正则项: $\frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$

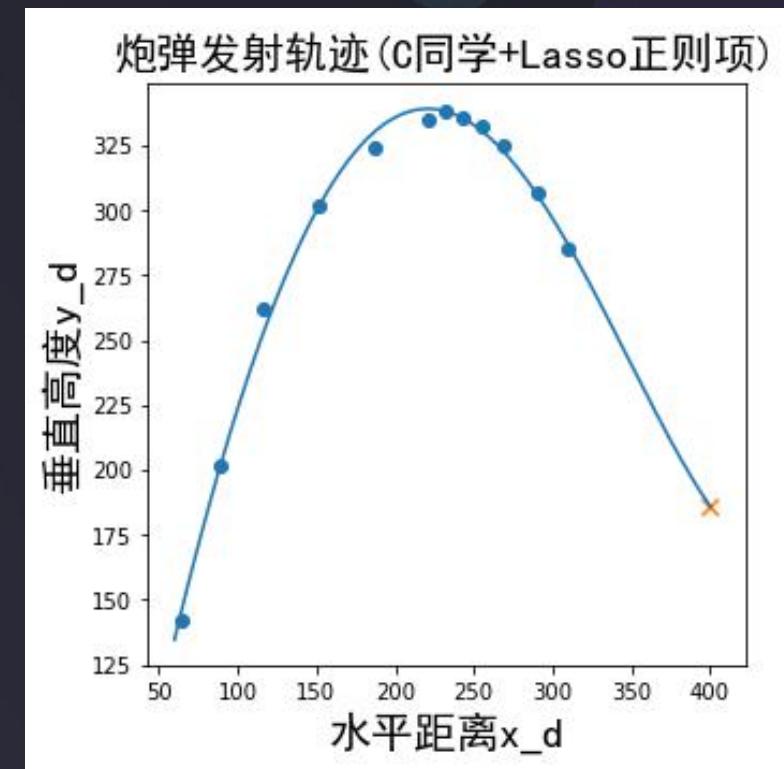
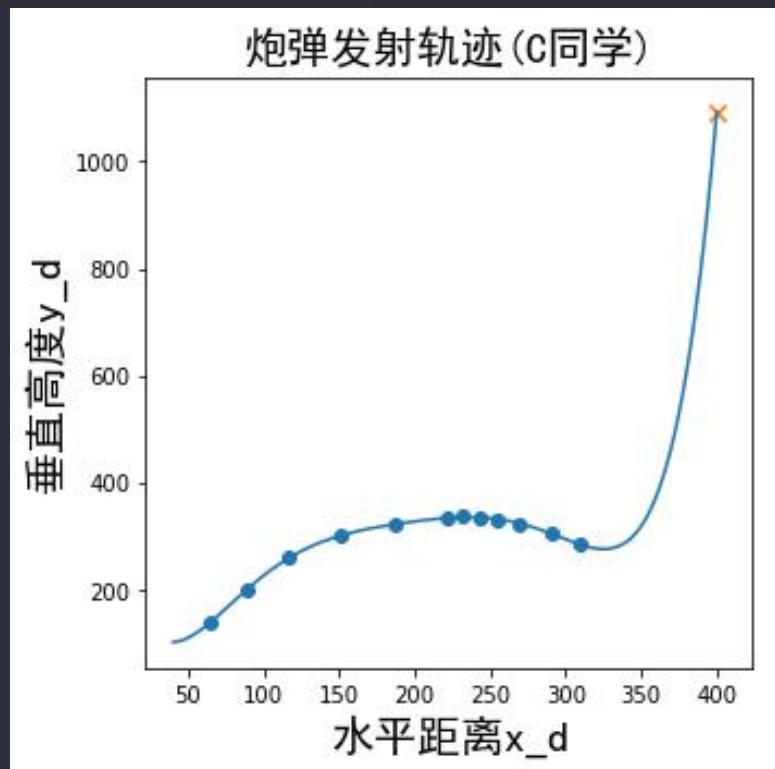


回归模型，引入Ridge回归正则项

增加正则项

回归模型，引入Lasso回归正则项

$$\frac{\lambda}{2m} \sum_{j=1}^n |\theta_j|$$



参考链接：

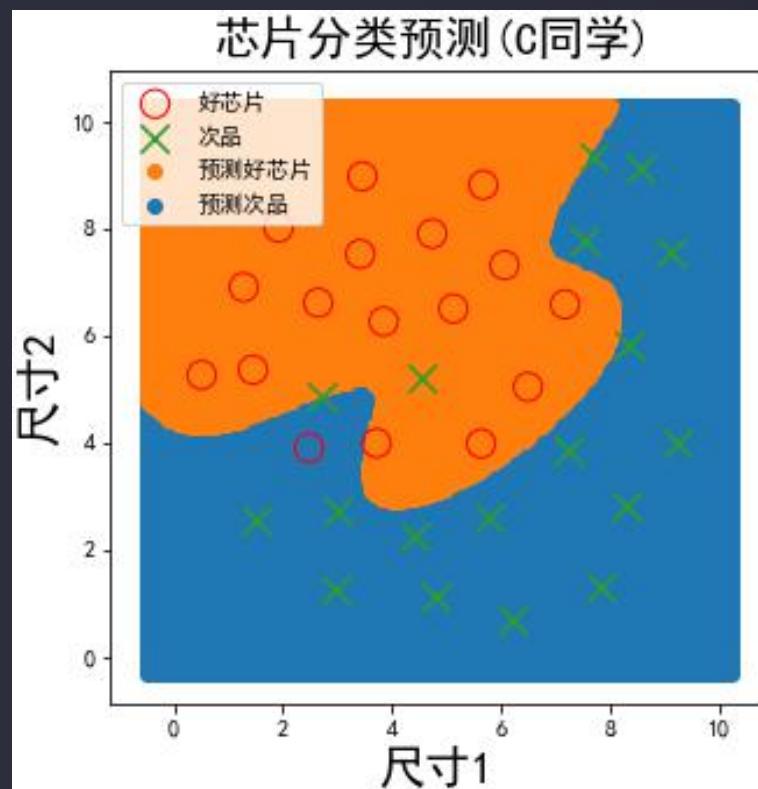
https://blog.csdn.net/dfly_zx/article/details/1079578

增加正则项

回归模型，引入Lasso回归正则项

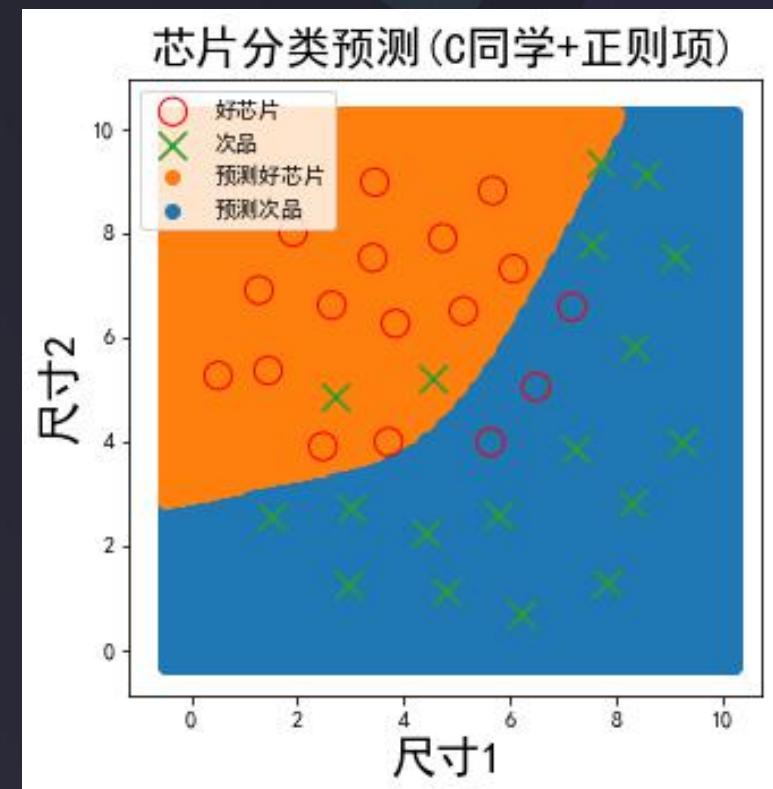
$$\frac{\lambda}{2m} \sum_{j=1}^n |\theta_j|$$

准确率：94.3%



过拟合结果

准确率：85.7%



参考链接：

https://blog.csdn.net/dfly_zx/article/details/1079578

| 知识巩固

思考：在日常的机器学习建模任务中，数据会决定模型表现的上限，因此应该尽可能多收集数据，属性也越多越好，这样能达到更好的效果。这是正确的还是错误的，为什么？



Python3人工智能入门+实战提升：机器学习

Chapter 7 综合能力提升之模型选择与优化

赵辛

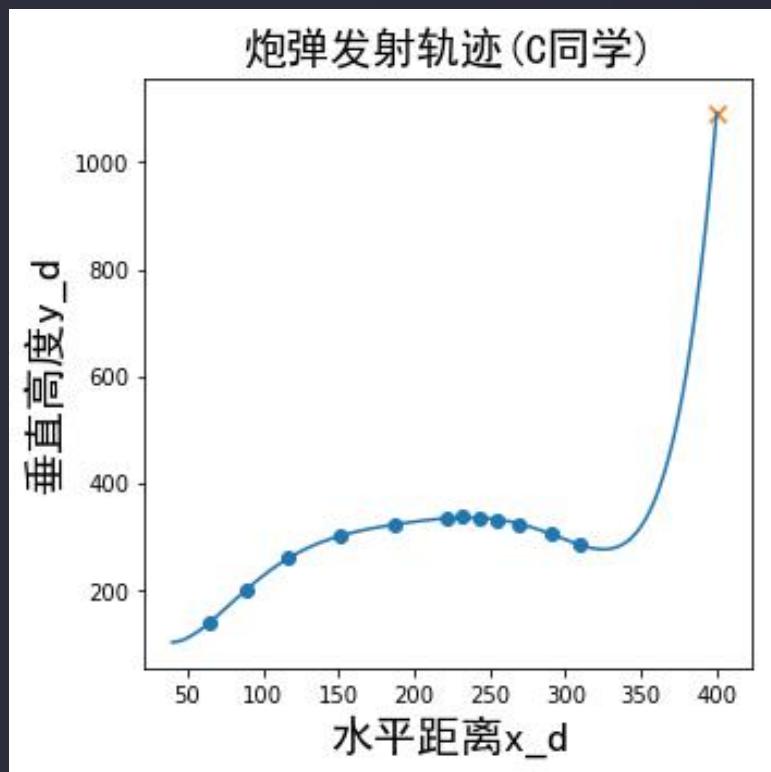
Chapter 7 综合能力提升之模型选择与优化

-
- 1 --模型过拟合与欠拟合
 - 2 --模型过拟合解决
 - 3 - 数据分离与混淆矩阵
 - 4 --模型选择与优化
 - 5 --实战准备
 - 6 --实战（一）综合提升之炮弹发射轨迹预测
 - 7 --实战（二）综合提升之芯片品质预测

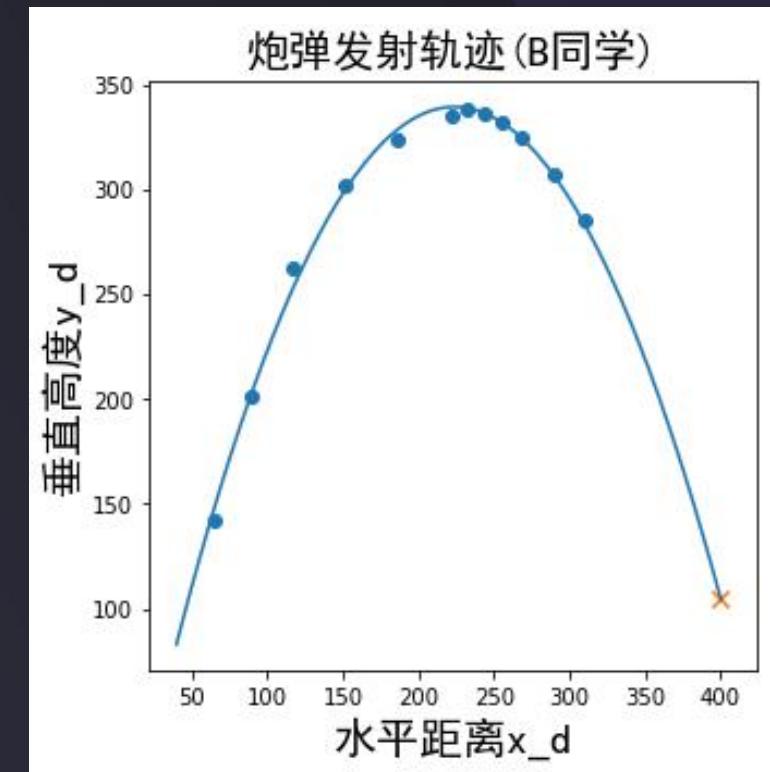
现实问题思考

根据收集到的部分炮弹发射数据，推测在 $x_d=400$ 时，炮弹高度 y_d

思考：仅仅通过训练数据的预测效果，是否足以评判模型的表现？



$$y_d(x_d=400) = 1092$$

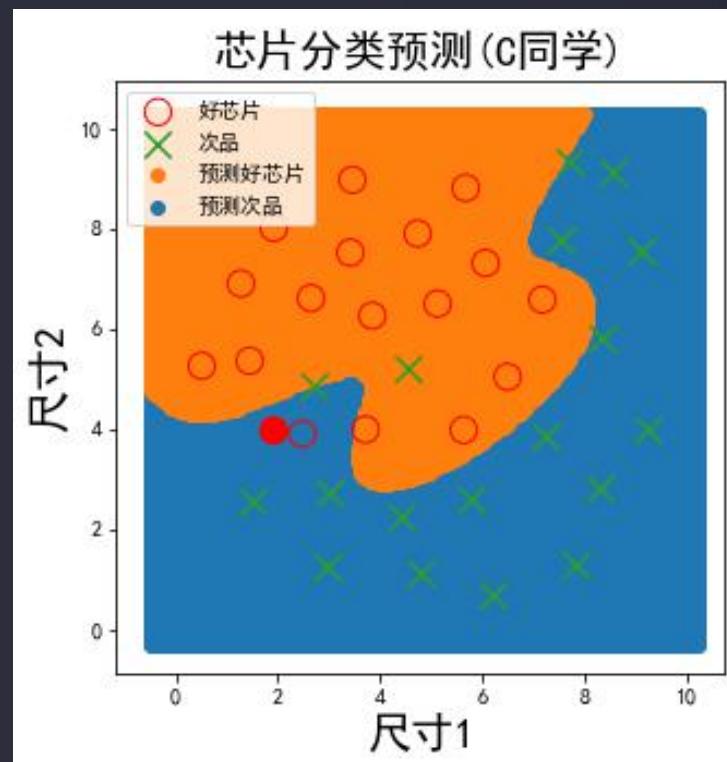


$$y_d(x_d=400) = 104$$

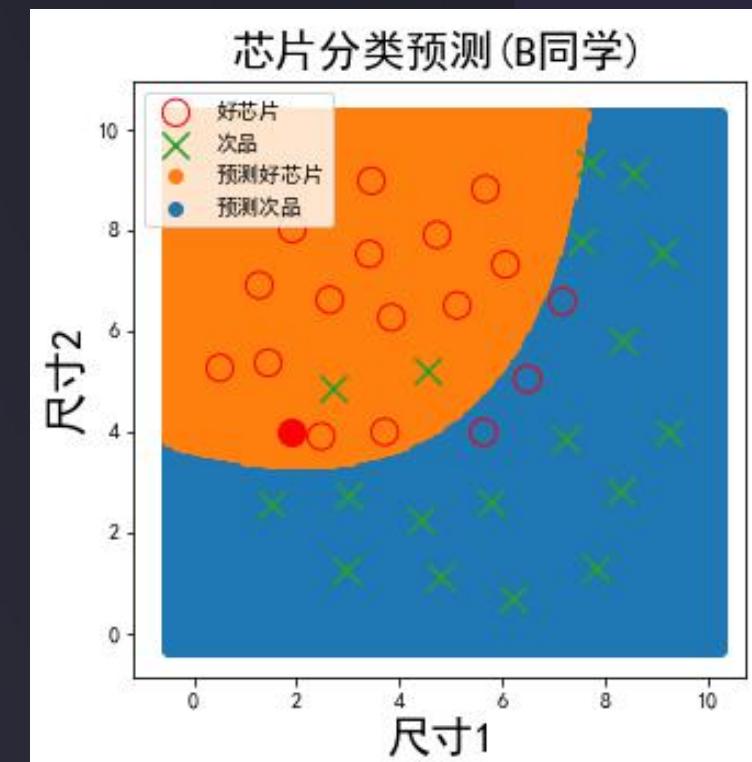
现实问题思考

一个好的模型，能通过对新数据样本做出准确预测

根据收集到的芯片质量数据，推测在尺寸1=1.9，尺寸2=4的芯片，其质量是否达标



预测为次品，质量不达标



预测为良品，质量达标

回顾模型训练与评估流程

数据载入 → 数据可视化与预处理 → 模型创建 → 全数据用于模型训练 → 模型评估



分离训练数据与测试数据

- 1、把全部数据分成两组：训练集、测试集
- 2、用训练集里的数据输入模型进行训练
- 3、用测试集里的数据输入模型进行预测，能有效评估此模型预测新的输入数据的表现

特征A	特征B	结果
2	4	8
3	5	15
4	6	24
5	7	35
6	8	48

全数据集



特征A	特征B	结果
2	4	8
3	5	15
6	8	48

训练数据

特征A	特征B	结果
5	7	35
4	6	24

测试数据

|混淆矩阵 (Confusion Matrix)

混淆矩阵
(Confusion Matrix)

|只用准确率作为模型评估指标的局限性

数据载入 → 数据可视化与预处理 → 模型创建 → 部分数据用于模型训练
→ 部分数据用于模型评估 → 模型评估



分类任务模型评估指标：准确率（accuracy）

局限性：不能全面或真实表达模型对各类别结果的预测准确度

现实问题思考：只用准确率作为模型评估指标的局限性

案例：奢侈品公司在投放广告前，根据部分高档消费客户的数据作为训练集和测试集，训练测试了高档消费客户的分类模型。该模型的准确率达到了95%。但是在实际广告投时，发现模型输出预测都为普非高档消费客户（非目标用户群体），其结果无法帮助决策。



高档消费客户只占所有消费客户的一小部分（5%），
模型预测所有用户都为非高档消费用户，准确率就
高达95%了，其实并没有找到目标用户群。

|只用准确率作为模型评估指标的局限性

以准确率作为分类问题的评价指标是有明显缺陷的，假如不同样本的比例非常不均衡，占大比例的类别会成为影响准确率的主要原因。

例如有100个样本，95个负样本，只有5个正样本，如果测试所有的样本结果都是负样本，也可以说准确率是95%

以上场景没有一个正样本识别出来！

局限性表现在：

- 没有体现数据子类别的预测效果（如：0、1分别预测的准确率）
- 没有体现模型**错误预测的类型**（如：5%的错误率是什么预测错误）

混淆矩阵

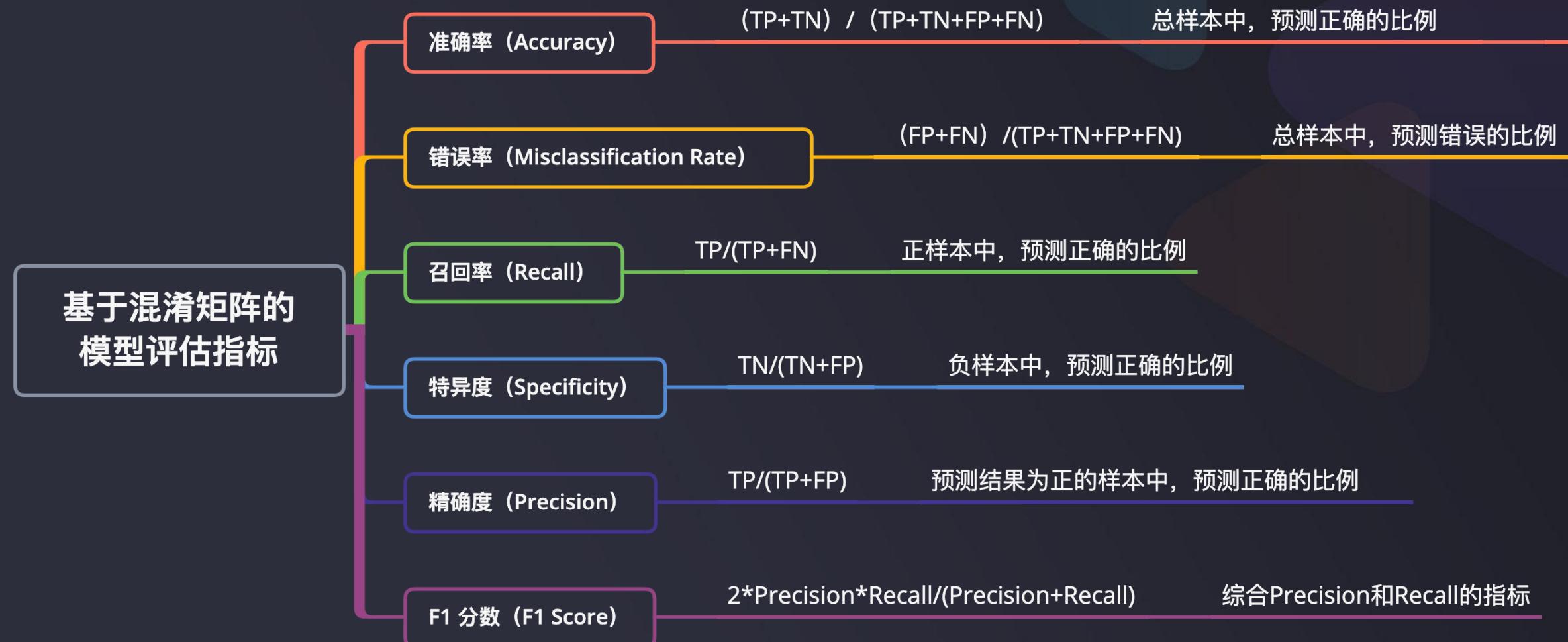
混淆矩阵也称误差矩阵，用于统计各类别样本预测正确与错误的数量，能帮助用户更全面地评估模型表现。

		Predicted	Predicted
		0	1
Actual	0	TN	FP
	1	FN	TP

- **True Positives (TP):** 预测准确、预测为正样本的数量（实际为1，预测为1）
- **True Negatives (TN):** 预测准确、预测为负样本的数量（实际为0，预测为0）
- **False Positives (FP):** 预测错误、预测为正样本的数量（实际为0，预测为1）
- **False Negatives (FN):** 预测错误、预测为负样本的数量（实际为1，预测为0）

(预测结果正确或错误，预测结果为正样本或负样本)

基于混淆矩阵计算评估指标



现实问题思考

广告投放情况：10000个用户，9500个非目标用户（负样本），500个目标用户（正样本）

模型A预测：10000个用户全都为非目标用户（负样本），0个目标用户（正样本）

模型B预测：9500个非目标用户中预测正确9025个，500个目标用户预测正确475个

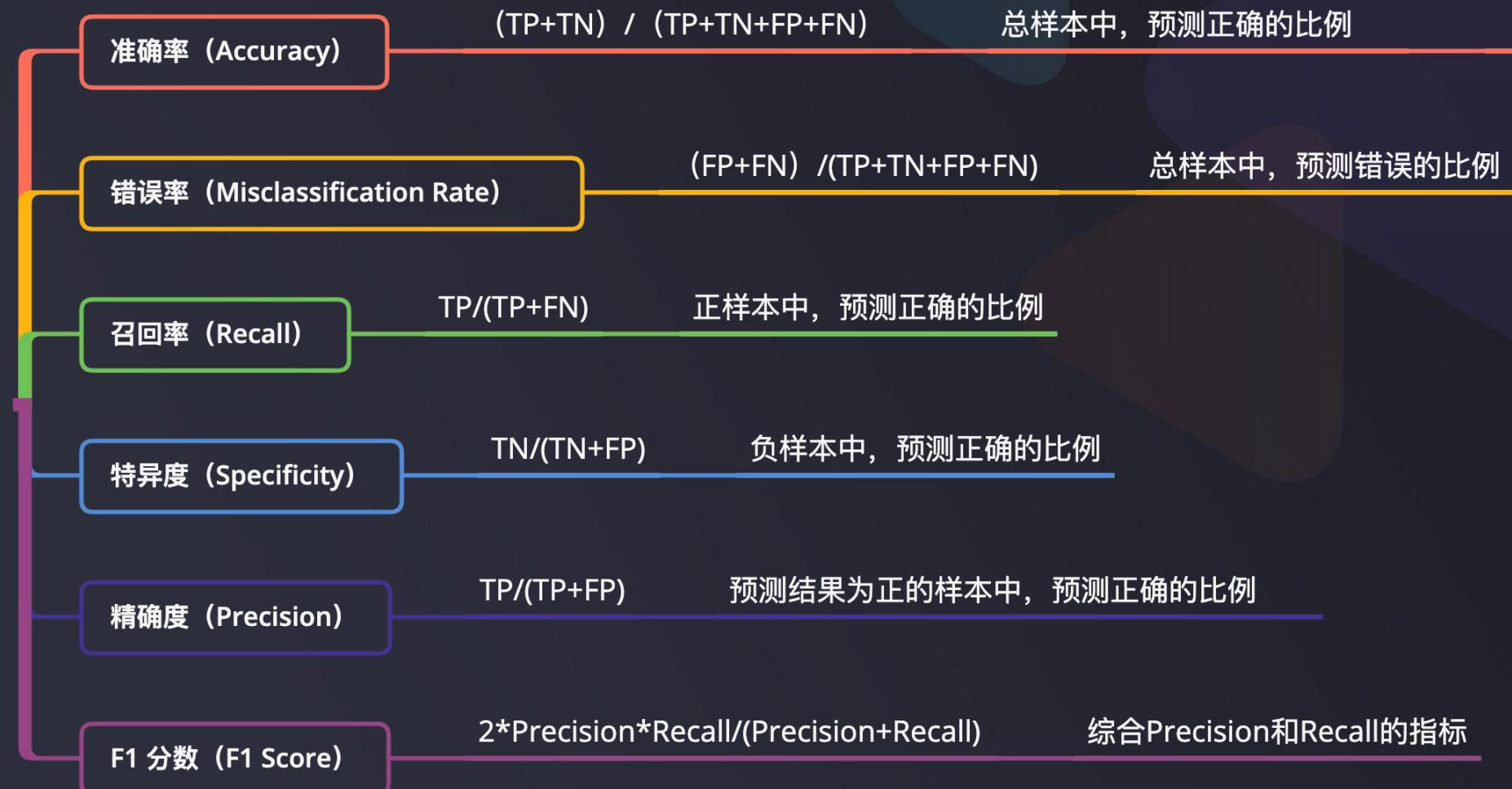
模型A混淆矩阵

		预测结果	
		0	1
实际 结果	0	TN=	FP=
	1	FN=	TP=

模型B混淆矩阵

		预测结果	
		0	1
实际 结果	0	TN=	FP=
	1	FN=	TP=

现实问题思考



现实问题思考

广告投放情况：10000个用户，9500个非目标用户（负样本），500个目标用户（正样本）

模型A预测：10000个用户全都为非目标用户（负样本），0个目标用户（正样本）

模型B预测：9500个非目标用户中预测正确9025个，500个目标用户预测正确475个

指标	模型A	模型B
准确率	0.95	0.95
错误率	0.05	0.05
召回率	0	0.95
特异度	1	0.95
精确率	0	0.95
F1分数	0	0.95

核心关注：能否把这500个目标用户（正样本）都尽可能找出来？

实际正样本中，预测正确的比例，即召回率！

混淆矩阵小结

优点:

- 分类任务中，相比单一的准确率指标，混淆矩阵提供了**更全面的模型评估信息** (TP\TN\FP\FN)
- 基于混淆矩阵，我们可以计算出**多样的模型表现衡量指标**，从而实现模型的综合评估

应用场景决定了衡量指标的重要性:

- **广告精准投放**(正样本为“目标用户”): 希望目标用户尽可能都被找出来、即实际正样本预测正确，需要关注召回率；同时，希望预测的正样本中实际都尽可能为正样本，需要关注**精确率**
- **异常消费检测** (正样本为“异常消费”): 希望判断为正常的消费（负样本）中尽可能不存在异常消费，还需要关注**特异度**

知识巩固

ID	实际类别	预测类别
1	0	1
2	0	0
3	1	1
4	1	0
5	1	1
6	1	0
7	1	1
8	1	1
9	0	0
10	1	1

问题：

- 1、根据数据计算混淆矩阵、再基于混淆矩阵计算准确率、错误率、召回率、特异度、精确度、F1分数；
- 2、思考这个模型在什么场景下是表现较好的，什么场景中表现不够好。



Python3人工智能入门+实战提升：机器学习

Chapter 7 综合能力提升之模型选择与优化

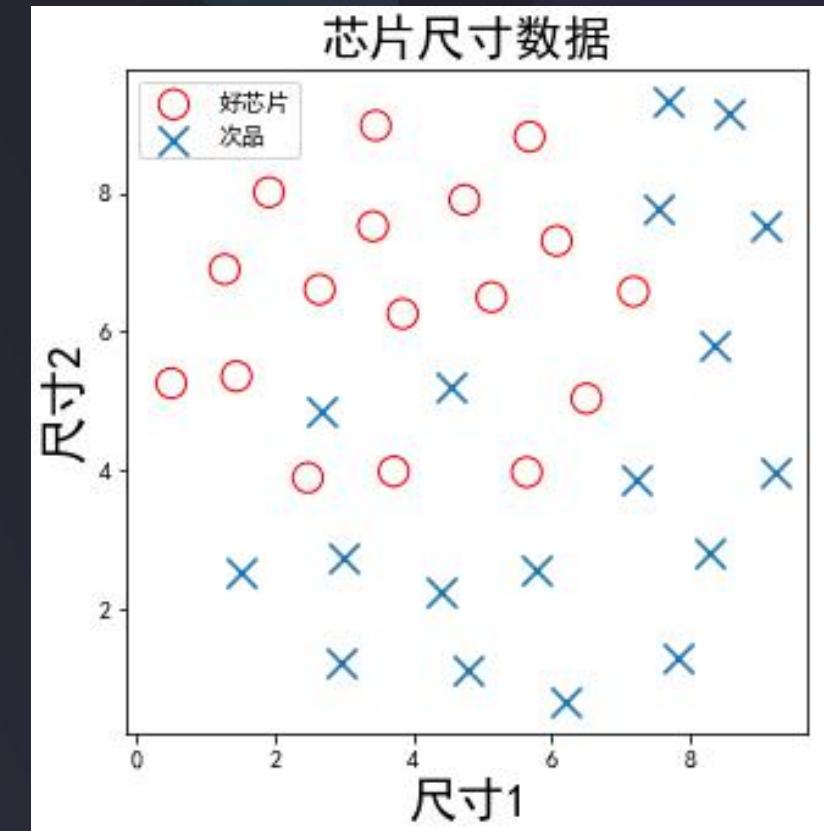
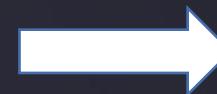
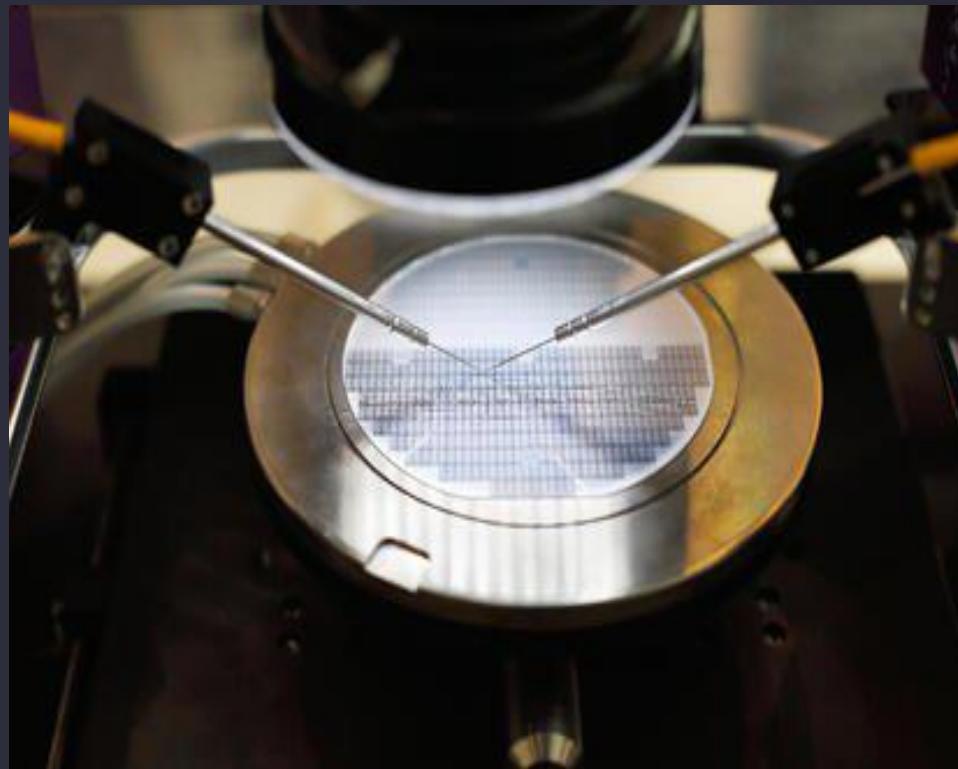
赵辛

Chapter 7 综合能力提升之模型选择与优化

-
- 1 --模型过拟合与欠拟合
 - 2 --模型过拟合解决
 - 3 --数据分离与混淆矩阵
 - 4 --模型选择与优化
 - 5 --实战准备
 - 6 --实战（一）综合提升之炮弹发射轨迹预测
 - 7 --实战（二）综合提升之芯片品质预测

现实问题思考

根据芯片尺寸1、尺寸2参数识别次品



根据收集到的芯片质量数据，推测在尺寸1=1.9，尺寸2=4的芯片，其质量是否达标

现实问题思考

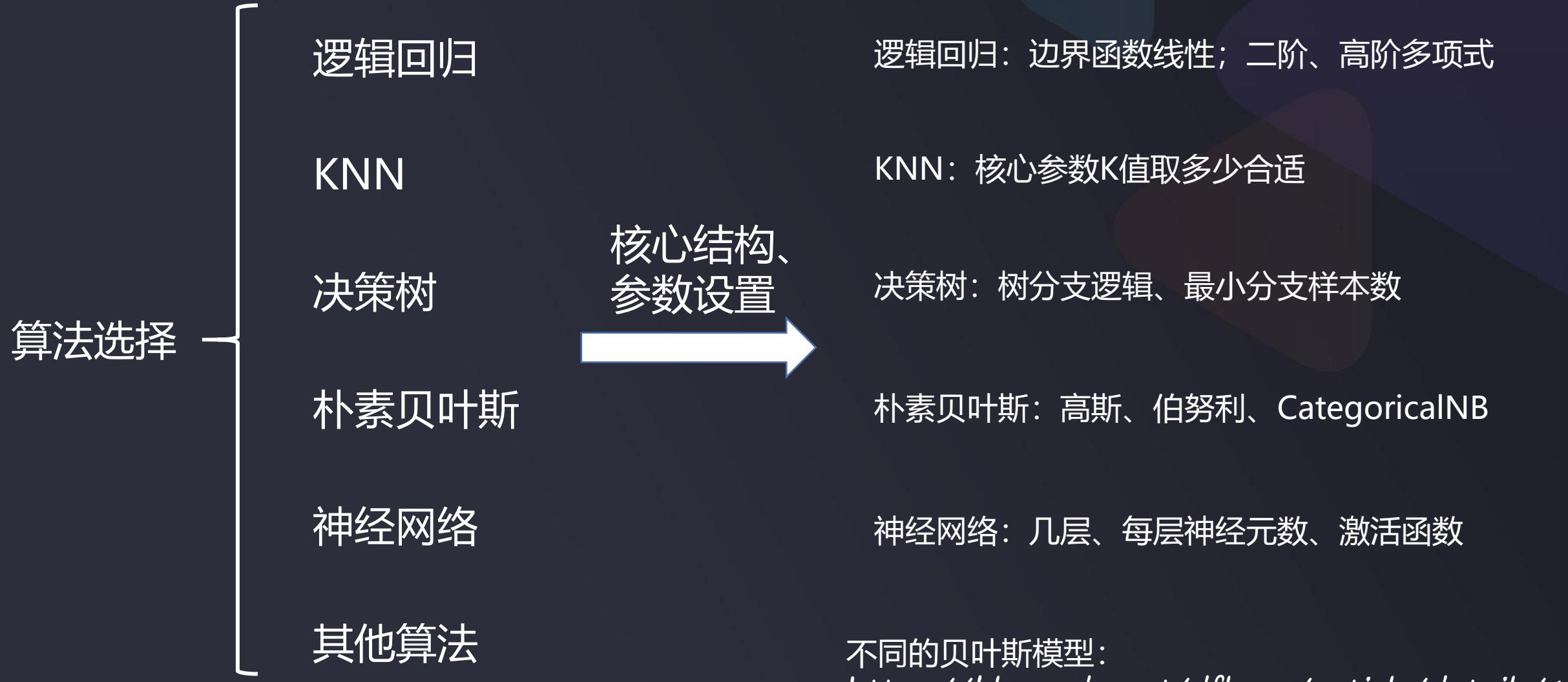
数据载入 → 数据可视化与预处理 → 模型创建 → 部分数据用于模型训练
部分数据用于模型评估 → 模型评估



三大核心问题： [

- 选用什么算法?
- 核心结构、参数如何设置?
- 模型表现不好，怎么办?

| 模型建立与优化



| 模型建立与优化

算法选择 → 核心结构、参数设置 → 模型表现

模型表现属于结果，如表现不好，需要从前往后找问题

→ 数据是否有问题、算法选的是不合适、核心结构与参数是否合理



训练样本预测准确率太低

测试样本准确率下降明显

召回率/特异度/精确率低

| 数据质量决定模型表现的上限！



上游决定下游，建模前五检查**：**

- 1、样本代表性：采集数据的方法是否合理，采集到的数据是否有代表性
- 2、标签统一化：对于样本结果，要确保每个样本都遵循一样的标签规则
- 3、数据合理性：样本中的异常数据点是否合理、如何处理
- 4、数据重要性：数据属性的意义，是否为无关数据
- 5、属性差异性：不同属性数据的数量级差异性如何

| 数据质量决定模型表现的上限！



尝试以下方法：

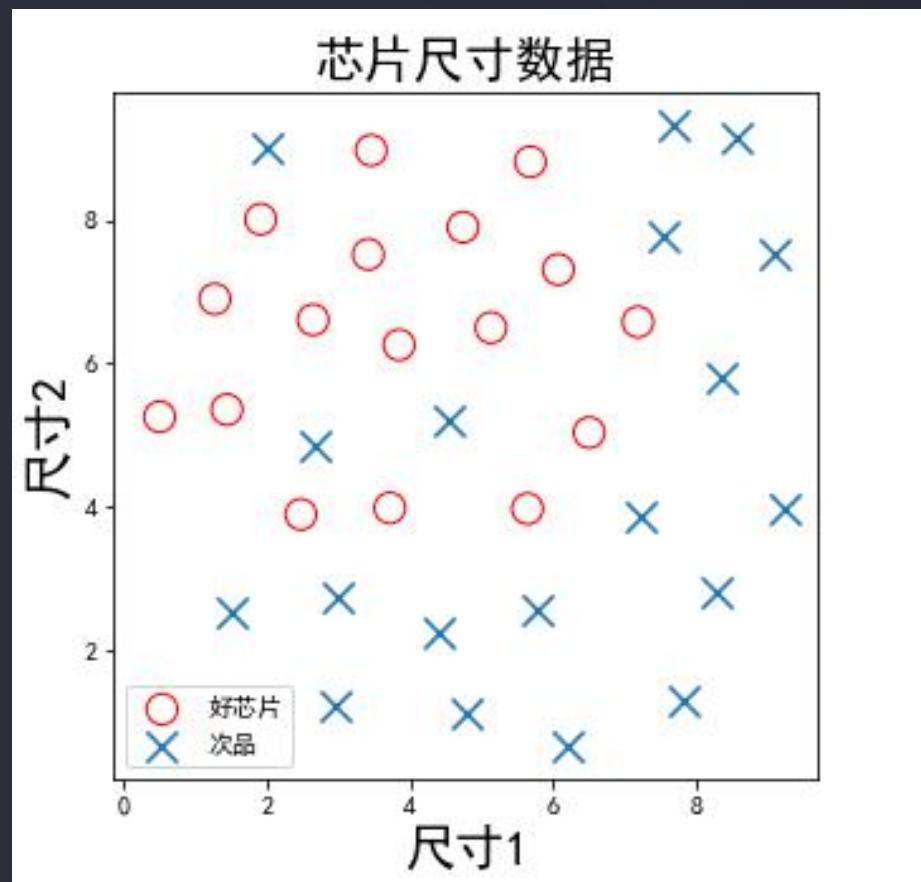
- 1、根据实际场景扩充或减少样本
- 2、对不合理标签数据进行预处理
- 3、删除不重要的属性数据、数据降维
- 4、对数据进行归一化或标准化
- 5、过滤掉异常数据

好处：

- 1、数据质量提升，有助于提高模型表现
- 2、帮助模型学习到正确信息（合理的“监督”）
- 3、降低噪声影响、减少过拟合、节约运算时间
- 4、平衡数据影响，加快训练收敛
- 5、降低噪声影响、提高鲁棒性

现实问题思考

根据芯片尺寸1、尺寸2参数识别次品



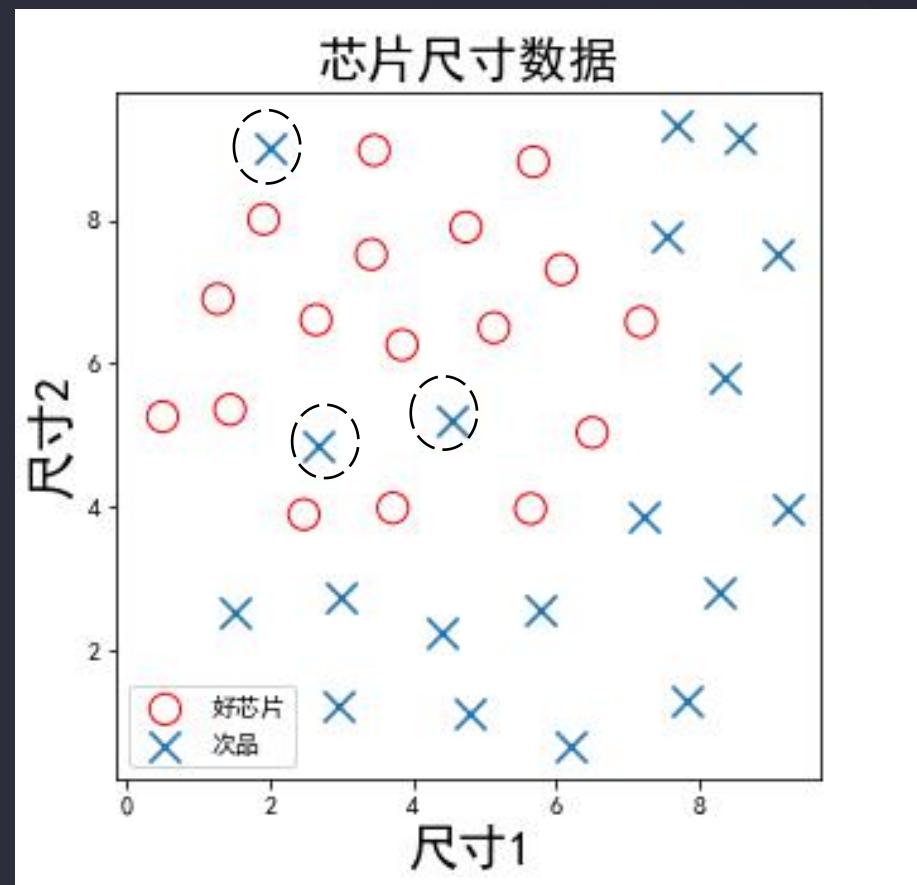
是否有异常数据点、对其是否要保留或删除？

不同属性数据量级差异如何？

是否有不重要的属性、是否需要降低数据维度？

现实问题思考：异常数据处理

根据芯片尺寸1、尺寸2参数识别次品



低维度数据集，对其进行可视化，可能的异常点：

$x_1=2, x_2=9, y=0$

$x_1=2.69, x_2=4.87, y=0$

$x_1=4.53, x_2=5.2, y=0$

计算数据基于高斯分布的概率密度，帮助决策：

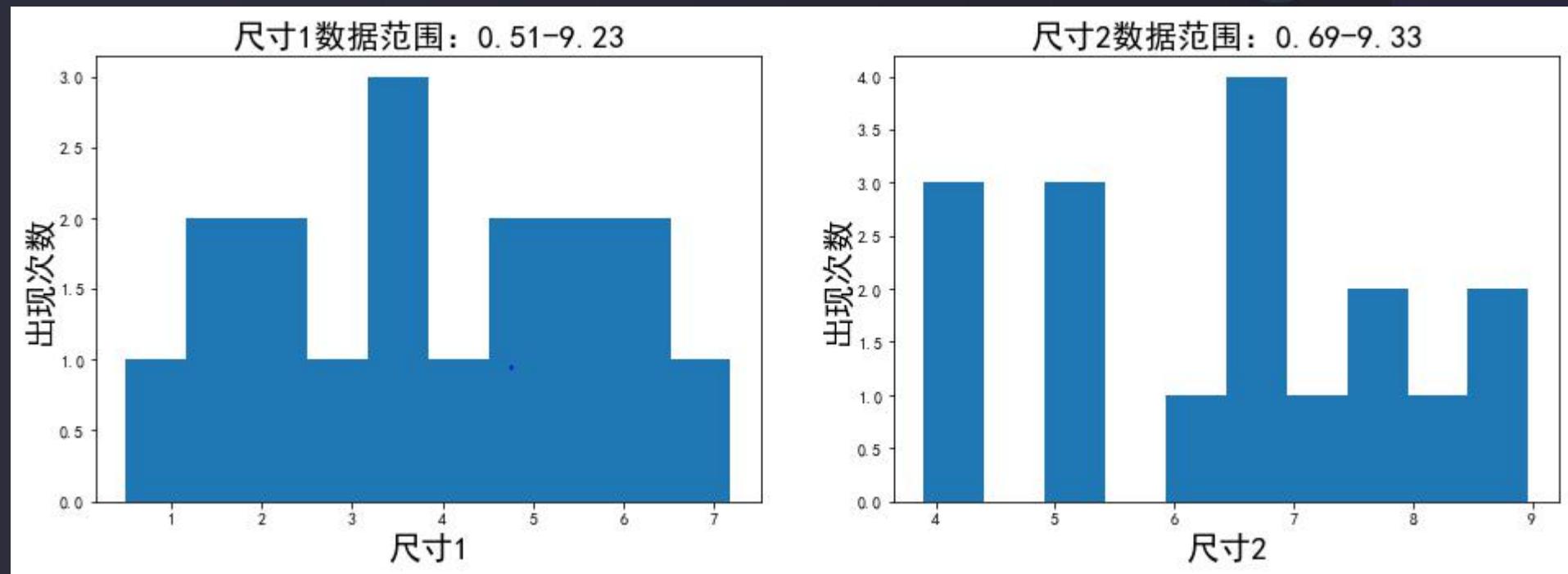
0.001980

0.009740

0.017871

现实问题思考：不同属性数据量级比较

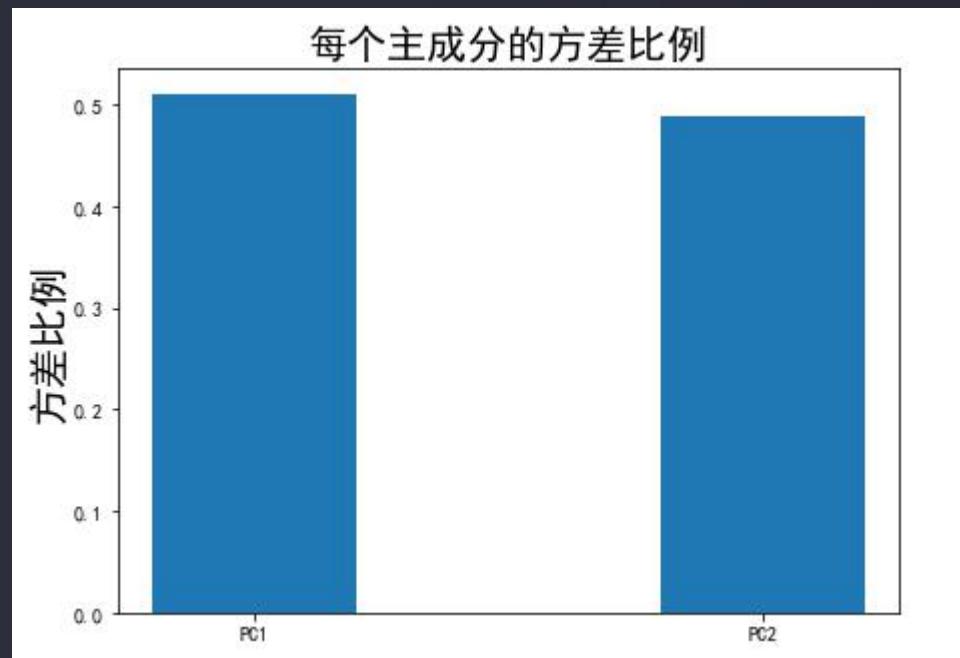
根据芯片尺寸1、尺寸2参数识别次品



两个属性的数据范围接近

现实问题思考：数据降维分析

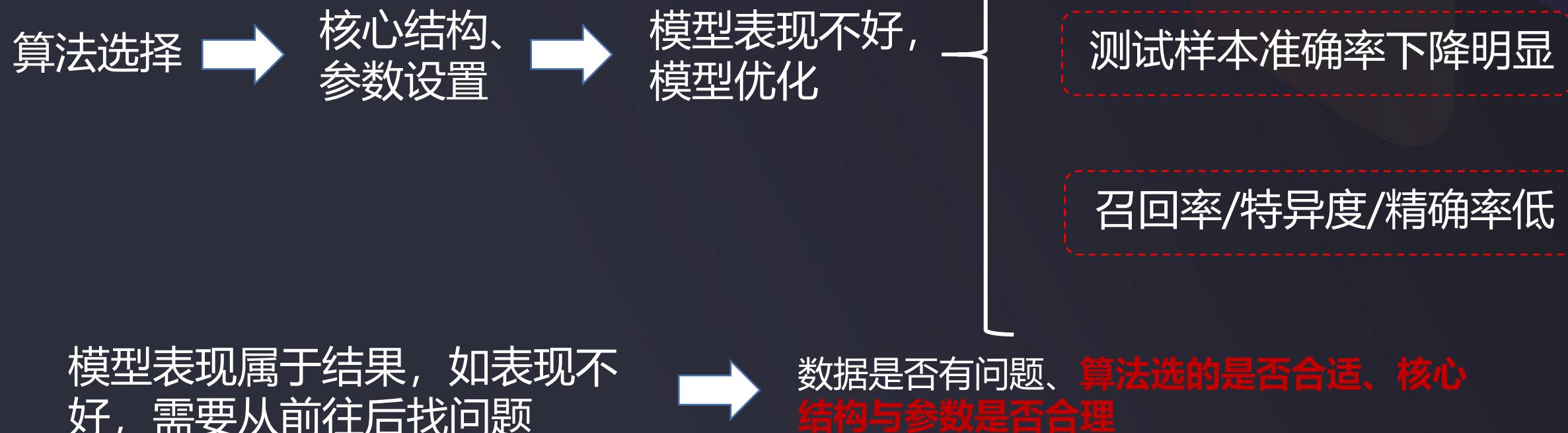
根据芯片尺寸1、尺寸2参数识别次品



对数据进行PCA分析，发现
需要保留两个维度的数据

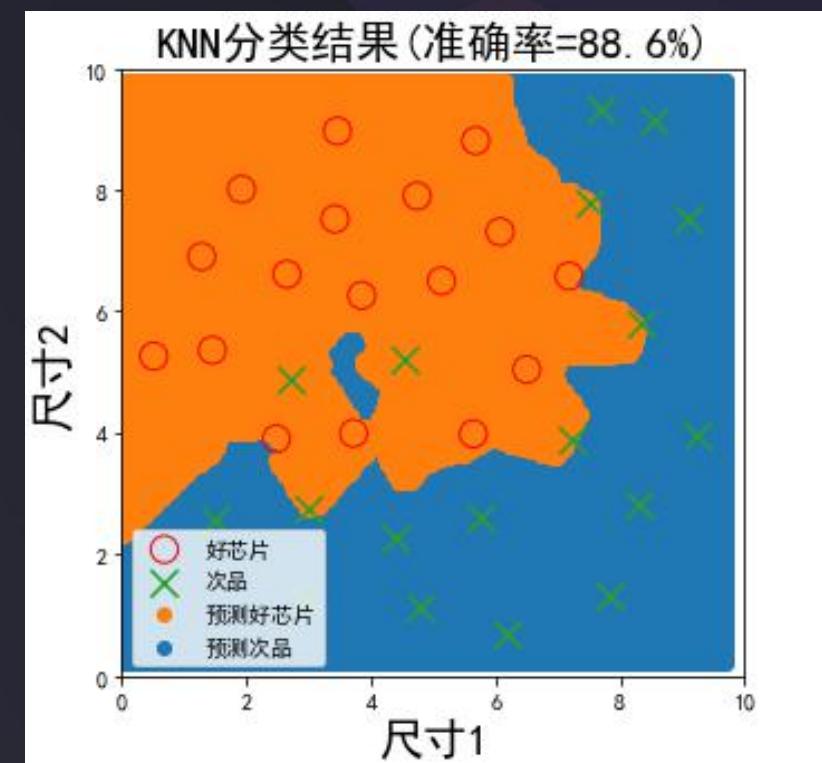
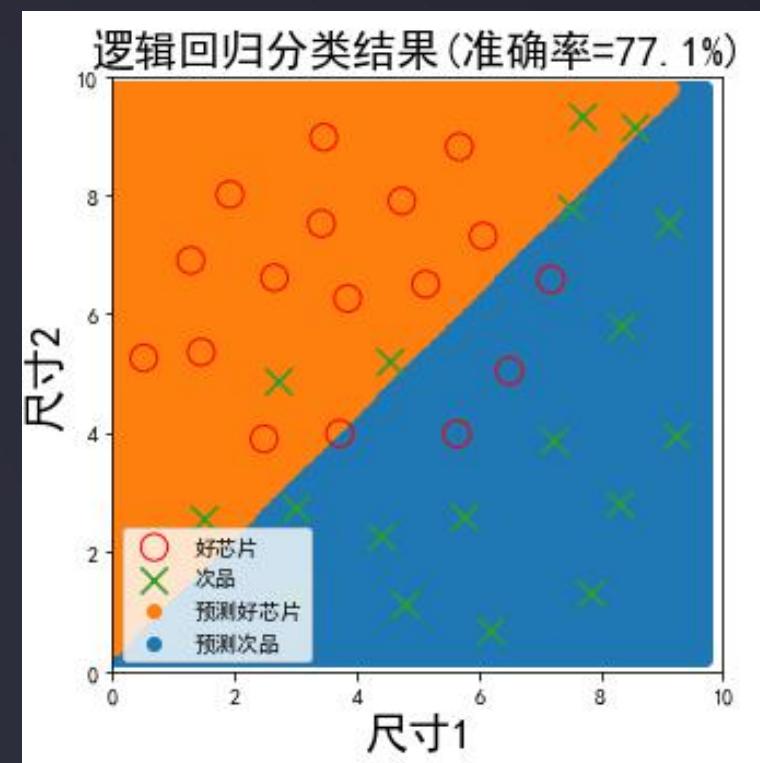
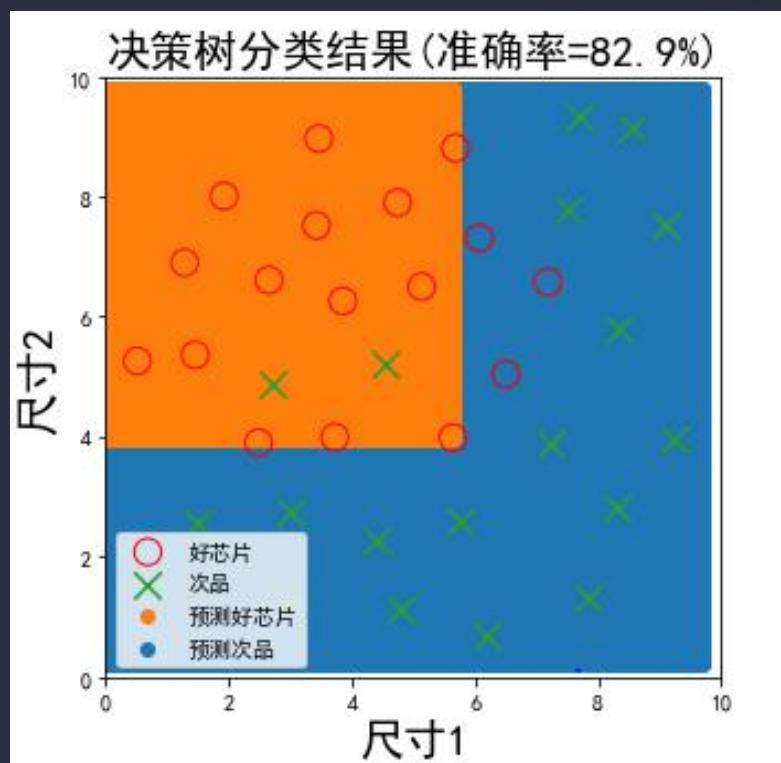
保留 x_1 、 x_2
也可尝试使用PC1、PC2

| 模型建立与优化



现实问题思考：多模型对比

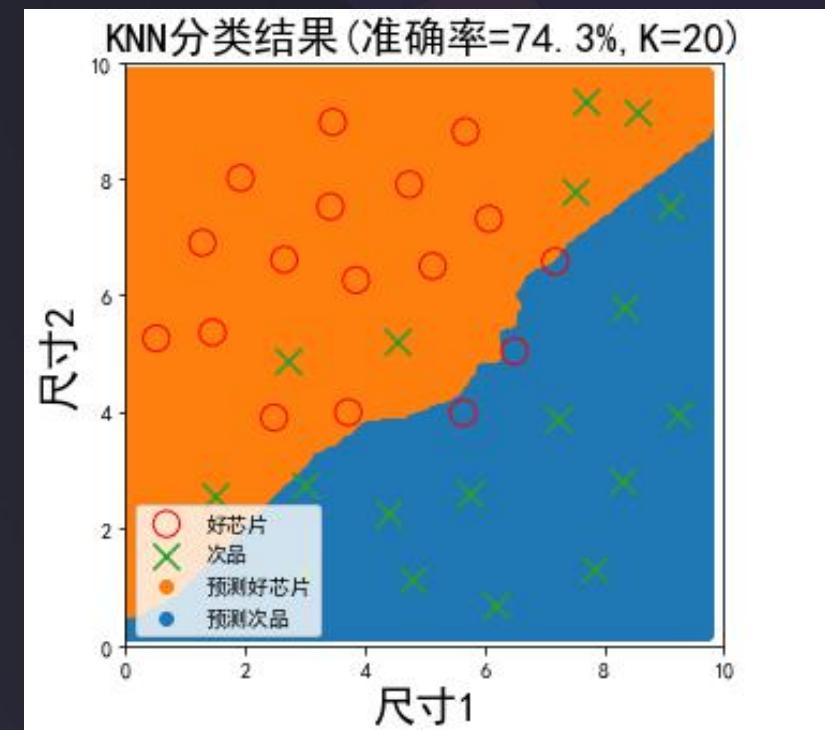
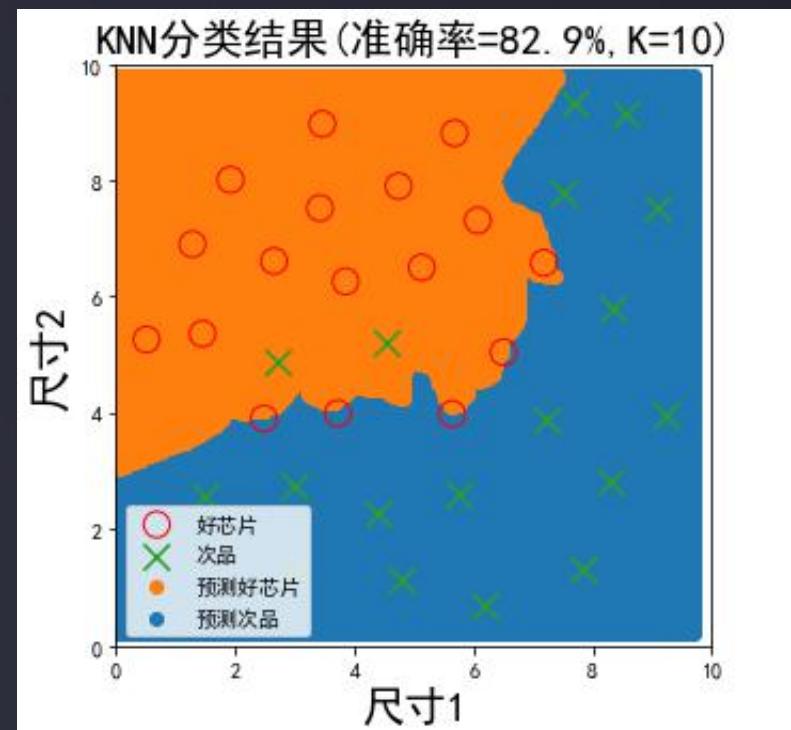
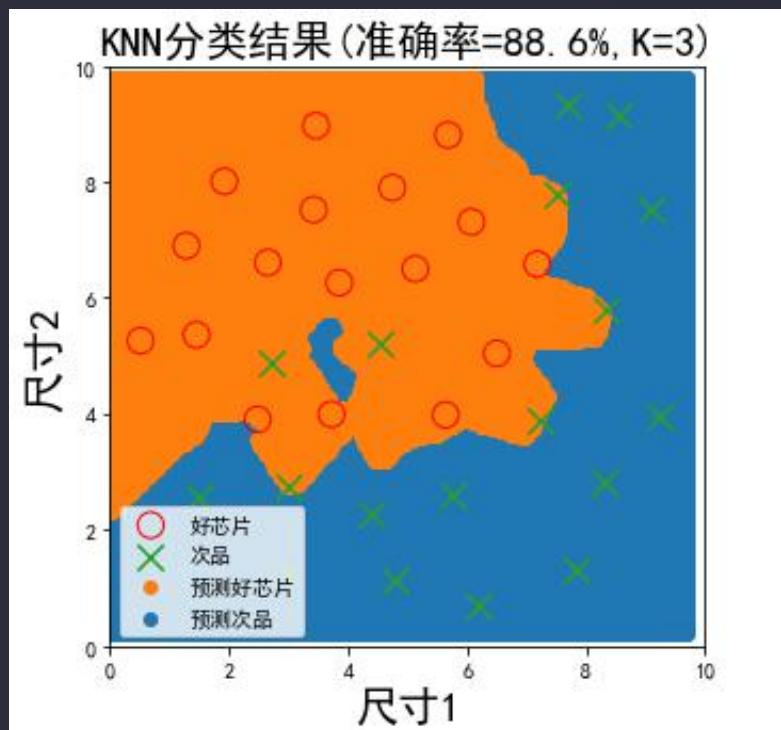
尝试不同的模型，对比模型表现



KNN模型 (K=3) 在训练数据上的准确率最高，边界也最为复杂

现实问题思考：单一模型的核心参数优化

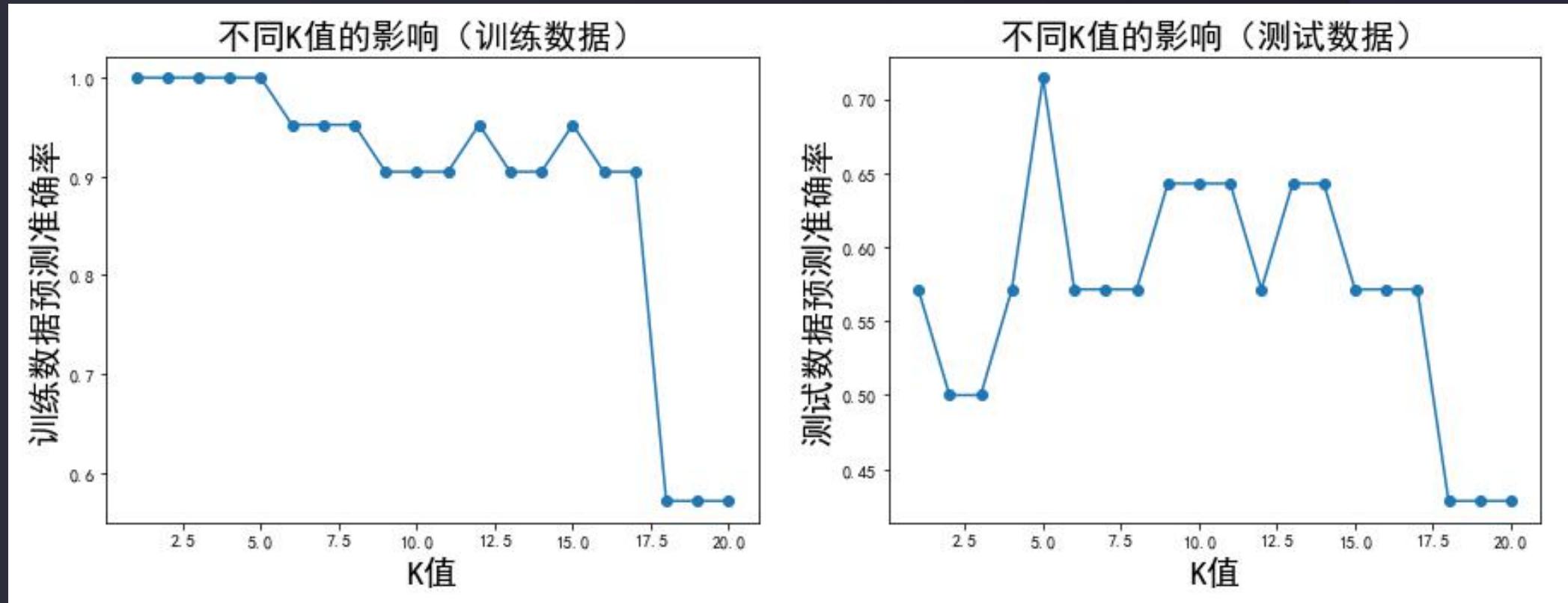
KNN模型，可以尝试不同的K值



K越小，训练数据的预测准确率越高，但分类边界也越复杂

现实问题思考：单一模型的核心参数优化

KNN模型，可以尝试不同的K值



K越小，训练数据的预测准确率越高，但测试数据准确率可能下降（过拟合）

|提高模型表现的四要素

数据预处理

- 扩大数据样本
- 增加/减少属性
- 数据降维、标准化
- 异常数据剔除...

模型选择

- 尝试不同的模型
- 通过不同指标、基于训练/测试数据评估表现

结构参数优化

- 不同的结构、求解方法
- 核心参数修改

其他方法

- 增加正则项
- 多模型结合.....

| 知识巩固

问题：以下哪些方法可能有助于提高模型表现，思考其具体的帮助？

- A. 收集更多的样本数据
- B. 对数据进行降维处理
- C. 尝试不同的模型
- D. 在模型损失函数计算中，增加正则项
- E. 尝试调整模型的核心参数
- F. 对计算机的配置进行升级



Python3人工智能入门+实战提升：机器学习

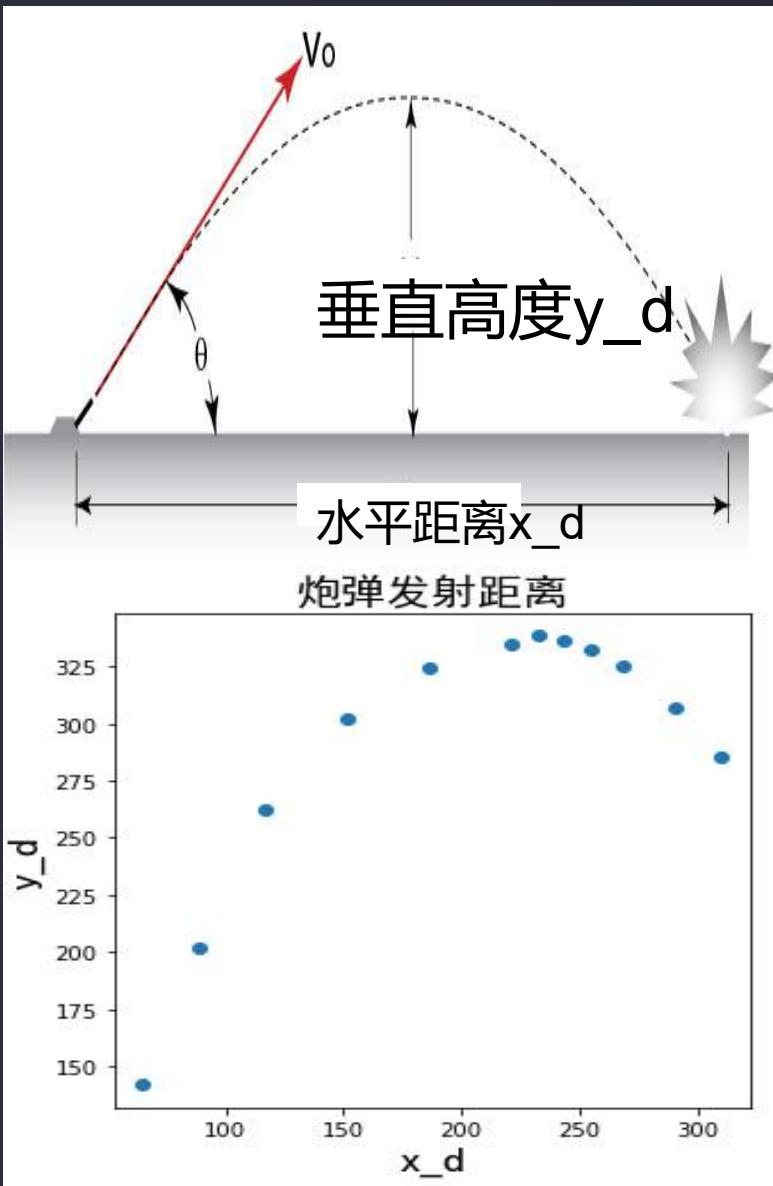
Chapter 7 综合能力提升之模型选择与优化

赵辛

Chapter 7 综合能力提升之模型选择与优化

-
- 1 --模型过拟合与欠拟合
 - 2 --模型过拟合解决
 - 3 --数据分离与混淆矩阵
 - 4 --模型选择与优化
 - 5 --实战准备
 - 6 --实战（一）综合提升之炮弹发射轨迹预测
 - 7 --实战（二）综合提升之芯片品质预测

任务一：炮弹发射轨迹预测



基于task1_train_data数据，建立回归模型，预测炮弹高度。

- 1、基于task1_train_data数据，建立线性回归模型，计算其在task1_test_data数据上的r2分数，可视化模型预测结果
- 2、分别引入2次、6次多项式属性数据，建立回归模型
- 3、对比三个模型对训练数据、测试数据集做预测的r2分数，判断哪个模型预测更准确
- 4、可视化三个模型的预测曲线，判断哪个模型预测更准确

炮弹发射轨迹预测

数据加载及展示

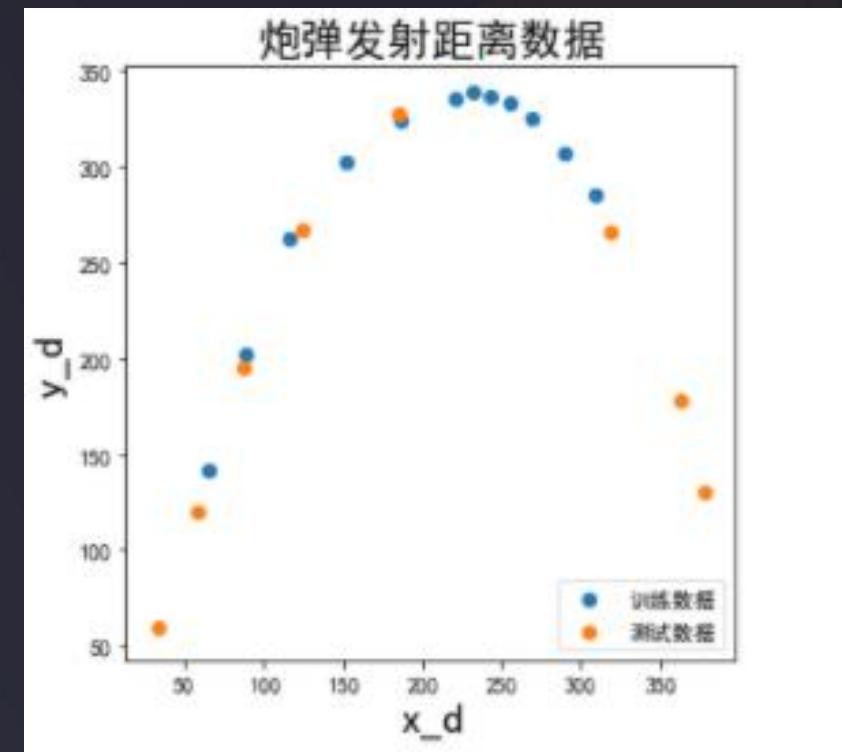
数据预处理

模型建立及训练

模型预测

结果展示

```
#训练数据加载  
import pandas as pd  
import numpy as np  
data_train = pd.read_csv('task1_train_data.csv')  
#测试数据加载  
data_test = pd.read_csv('task1_test_data.csv')
```



炮弹发射轨迹预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

#线性回归模型训练

```
from sklearn.linear_model import LinearRegression  
lr1 = LinearRegression()  
lr1.fit(X_train,y_train)
```

#模型预测

```
y_train_predict = lr1.predict(X_train)  
y_test_predict = lr1.predict(X_test)  
from sklearn.metrics import r2_score  
r2_train = r2_score(y_train,y_train_predict)  
r2_test = r2_score(y_test,y_test_predict)
```

```
training r2: 0.5756251457400435  
test r2: -1.6553718247964797
```

炮弹发射轨迹预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

```
#生成新数据  
X_range = np.linspace(40,400,300).reshape(-1,1)  
print(X_range.shape,min(X_range),max(X_range))  
print(X_range)  
y_range_predict = lr1.predict(X_range)
```

```
(300, 1) [40.] [400.]  
[[ 40.  
[ 41.20401338]  
[ 42.40802676]  
[ 43.61204013]  
[ 44.81605351]  
[ 46.02006689]
```

炮弹发射轨迹预测

数据加载及展示

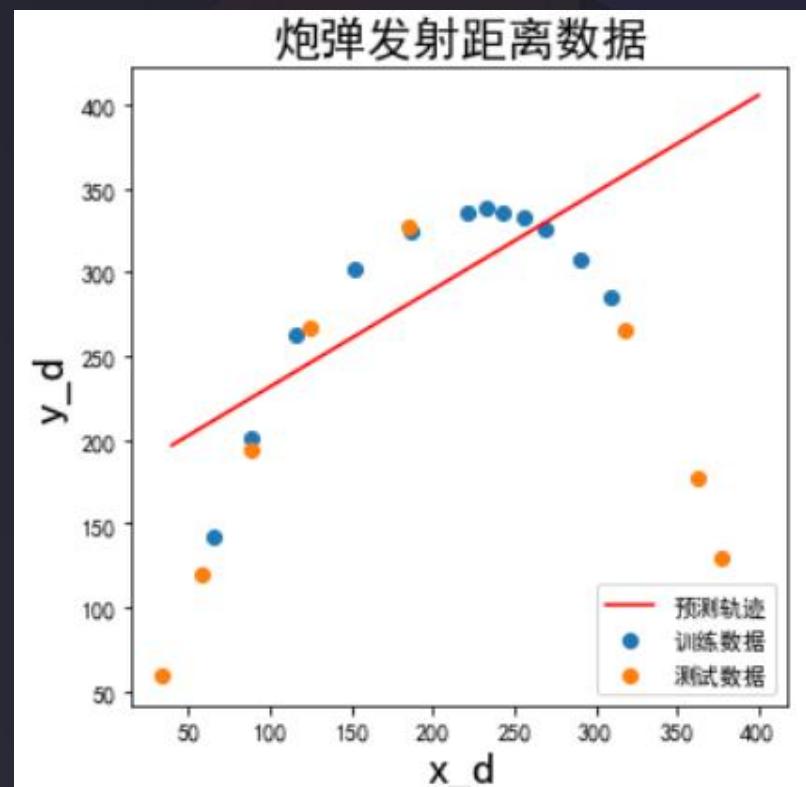
数据预处理

模型建立及训练

模型预测

结果展示

```
#数据与结果可视化  
fig2 = plt.figure(figsize=(5,5))  
curve_p = plt.plot(X_range,y_range_predict,'r',label='预测  
轨迹')  
data_train = plt.scatter(X_train,y_train,label='训练数据')  
data_test = plt.scatter(X_test,y_test,label='测试数据')  
plt.legend(loc='lower right')  
plt.title('炮弹发射距离数据',font2)  
plt.xlabel('x_d',font2)  
plt.ylabel('y_d',font2)  
plt.show()
```



炮弹发射轨迹预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

```
#生成二次多项式数据
from sklearn.preprocessing import PolynomialFeatures
poly2 = PolynomialFeatures(degree=2)
X_2_train = poly2.fit_transform(X_train)
X_2_test = poly2.transform(X_test)
print(X_train.shape,X_2_train.shape)
print(X_train[0:5],'\n',X_2_train[0:5,:])
```

```
(12, 1) (12, 3)
[[232.274 ]
 [ 64.8744]
 [ 88.8854]
 [116.517 ]
 [151.444 ]]
[[1.00000000e+00 2.32274000e+02 5.39512111e+04]
[1.00000000e+00 6.48744000e+01 4.20868778e+03]
[1.00000000e+00 8.88854000e+01 7.90061433e+03]
[1.00000000e+00 1.16517000e+02 1.35762113e+04]
[1.00000000e+00 1.51444000e+02 2.29352851e+04]]
```

炮弹发射轨迹预测

数据加载及展示

training r2: 0.575625
test r2: -1.655371824

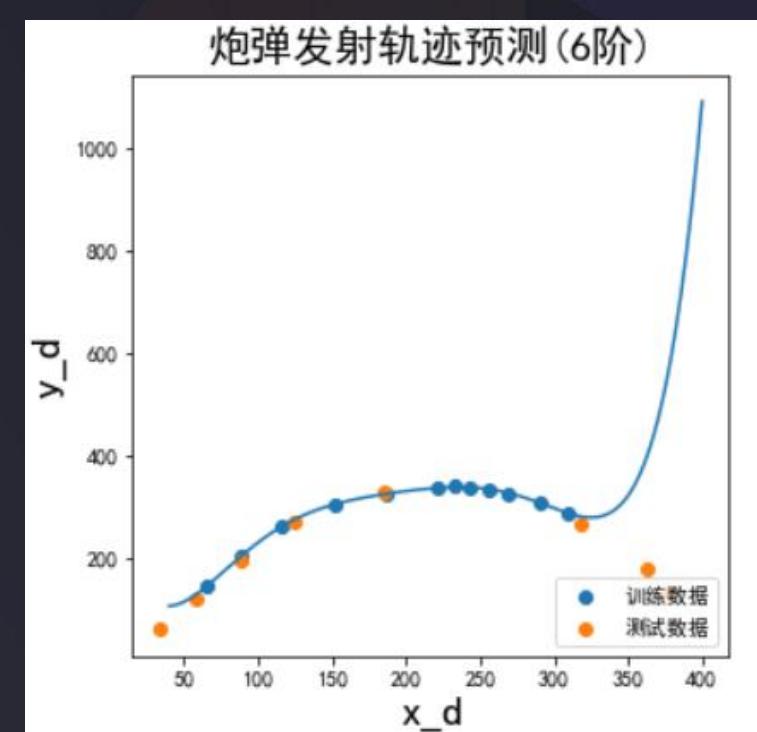
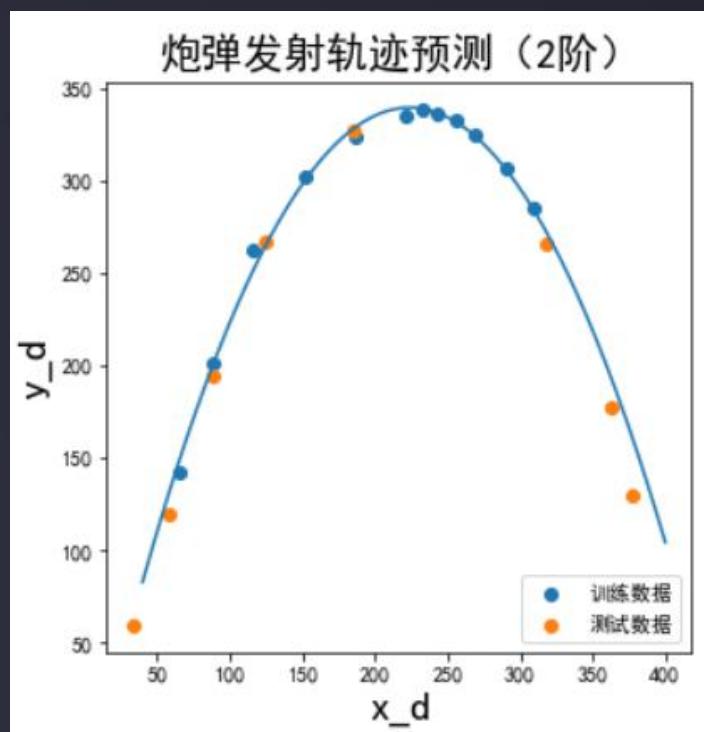
training r2_2: 0.9954725354913146
test r2_2: 0.9738532191575348
training r2_6: 0.9997685987915845
test r2_6: -3.296834183909657

数据预处理

模型建立及训练

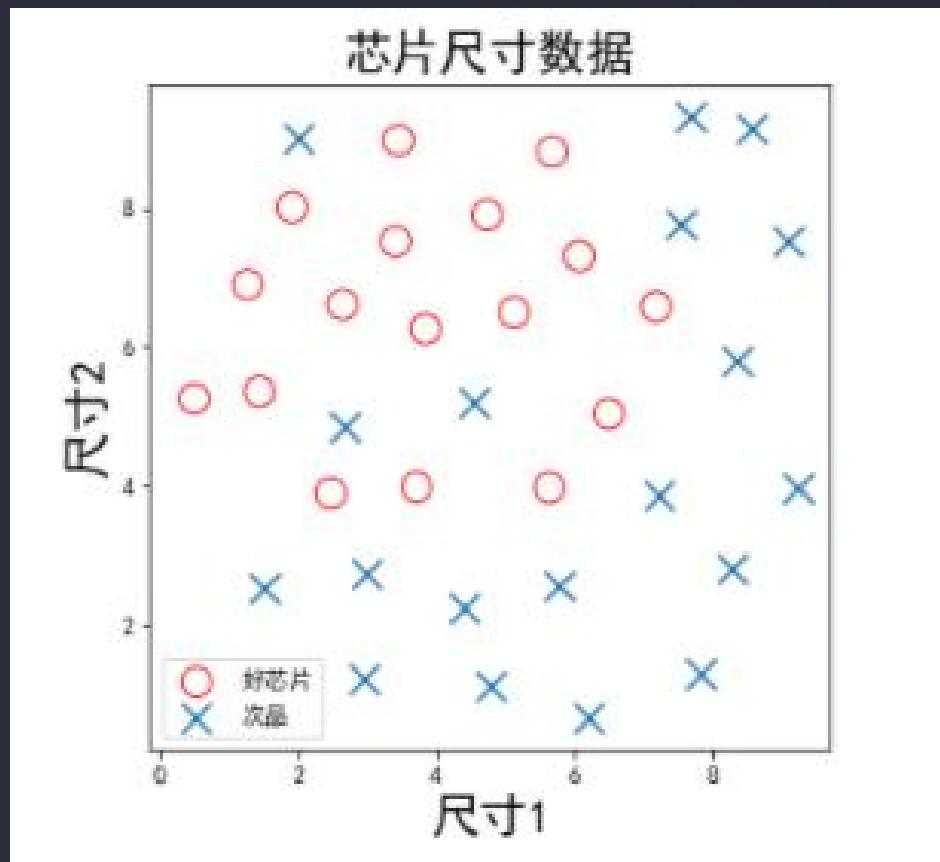
模型预测

结果展示



任务二：芯片品质预测

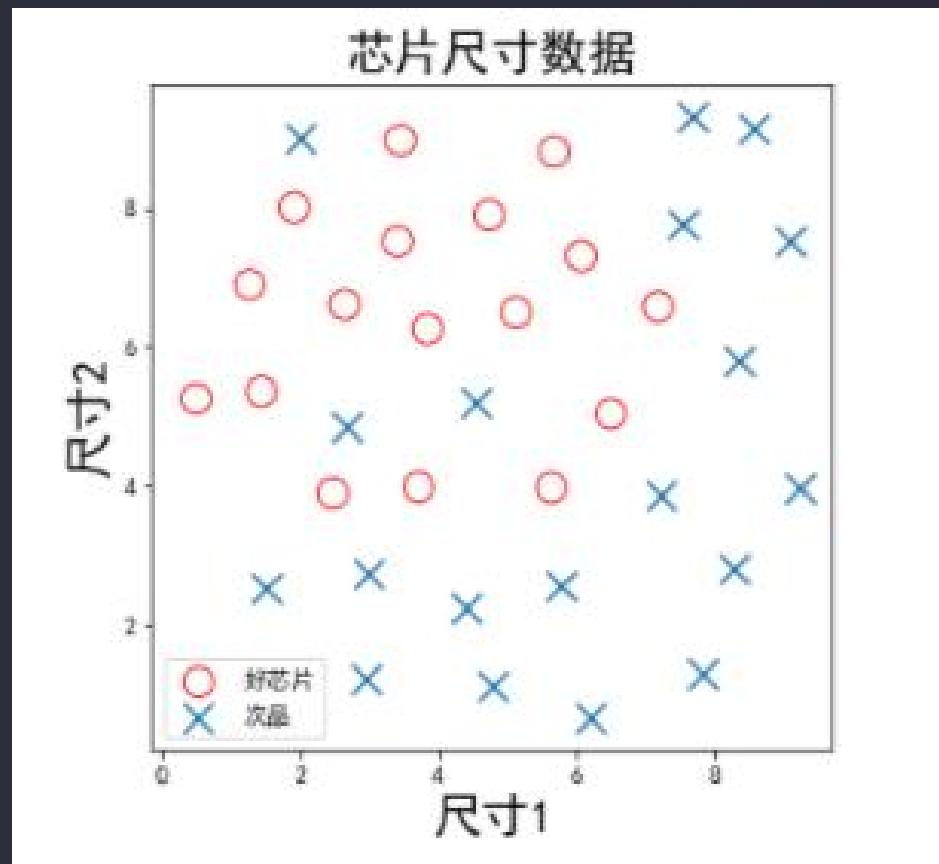
基于task2_data数据，综合异常数据检测、PCA降维、数据分离、KNN等技术完成芯片品质预测。



- 1、基于高斯分布概率密度函数，对两个维度数据进行分析、计算概率密度函数，寻找异常点并剔除
- 2、统计分析各维度数据分布
- 3、对数据进行主成分分析，计算各维度方差比例
- 4、数据分离，数据分离参数：
`random_state=1,test_size=0.4`

任务二：芯片品质预测

基于task2_data数据，综合异常数据检测、PCA降维、数据分离、KNN等技术完成芯片品质预测。



5、建立KNN模型 ($K=3$) 完成分类，可视化分类边界

6、计算测试数据集对应的混淆矩阵，准确率、召回率、特异度、精确率、F1分数

7、尝试不同的K值 (1-20)，计算其在训练数据集、测试数据集上的准确率并作图

任务拓展：尝试其他模型完成预测：决策树、逻辑回归、朴素贝叶斯

芯片品质预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

```
#计算高斯分布概率  
p1 = 1/sigma1/math.sqrt(2*math.pi)*np.exp(-np.power((x1-  
u1),2)/2/math.pow(sigma1,2))
```

```
p2 = 1/sigma2/math.sqrt(2*math.pi)*np.exp(-np.power((x2-  
u2),2)/2/math.pow(sigma2,2))
```

```
p = np.multiply(p1,p2)  
print(p)  
print(max(p),min(p),max(p)/min(p))
```

```
max p: 0.01820586881510458  
min p: 0.001979598419462152  
max/min: 9.196748510261507
```

```
0    0.008934  
1    0.011729  
2    0.009050  
3    0.009334  
4    0.010360  
5    0.017426  
6    0.018206  
7    0.011982  
8    0.005428  
9    0.003318  
10   0.003993  
11   0.008846  
12   0.013695  
13   0.006155  
14   0.009603  
15   0.004068  
16   0.009740  
17   0.017871  
35   0.001980  
dtype: float64
```

芯片品质预测

数据加载及展示

数据预处理

模型建立及训练

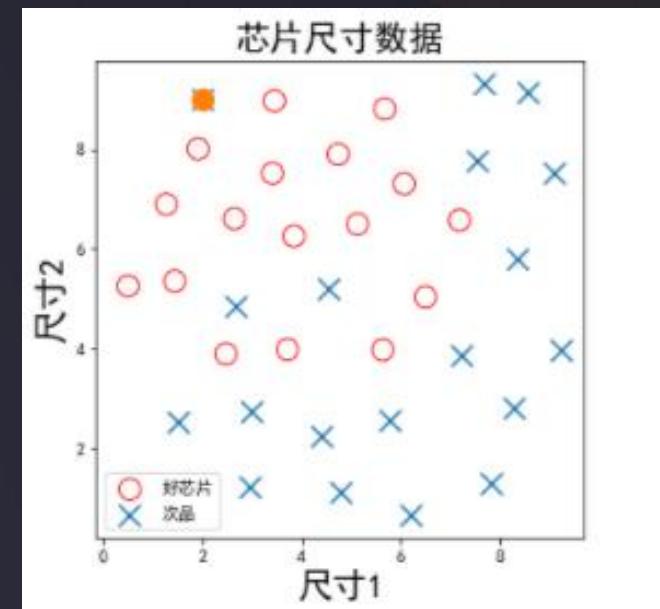
模型预测

结果展示

#异常数据点检测

```
from sklearn.covariance import EllipticEnvelope  
ad_model = EllipticEnvelope(contamination=0.02)  
ad_model.fit(X[y==0])  
y_predict_bad = ad_model.predict(X[y==0])  
print(y_predict_bad)
```

```
[ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1]
```



芯片品质预测

数据加载及展示

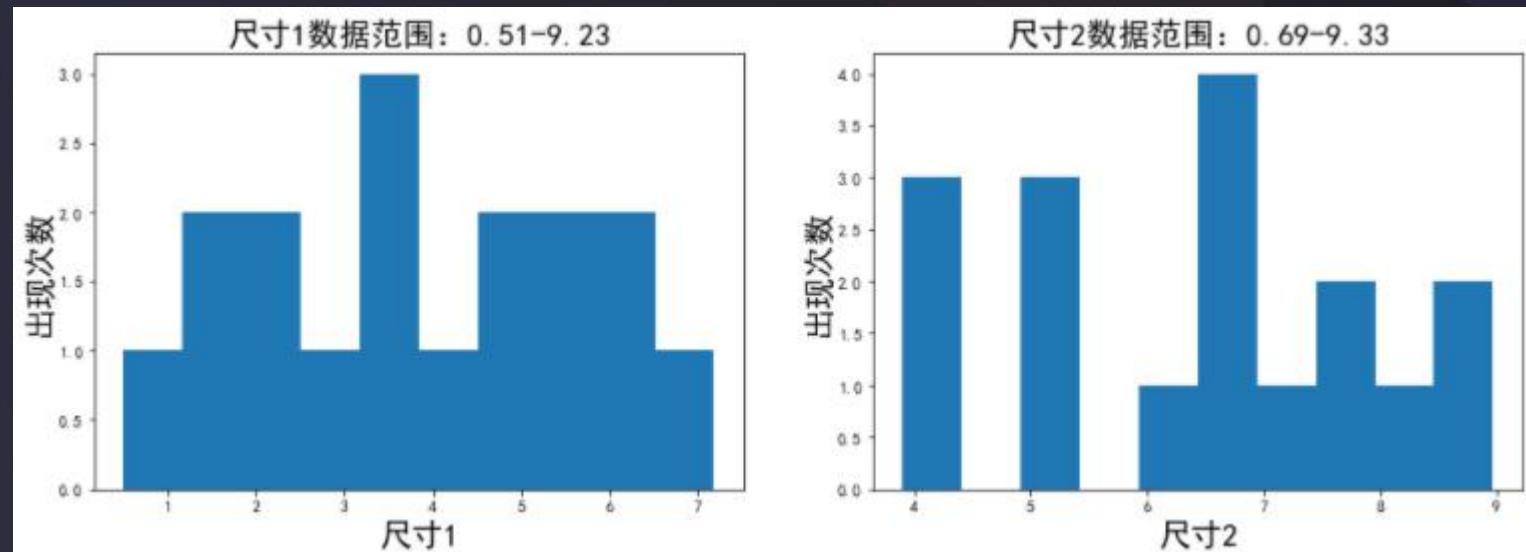
数据预处理

模型建立及训练

模型预测

结果展示

```
#各维度数据分布  
fig2 = plt.figure(figsize=(16,5))  
plt.subplot(121)  
plt.hist(x1,bins=10)  
plt.title('尺寸1数据范围：0.51-9.23',font2)  
plt.xlabel('尺寸1',font2)  
plt.ylabel('出现次数',font2)
```



芯片品质预测

数据加载及展示

数据预处理

模型建立及训练

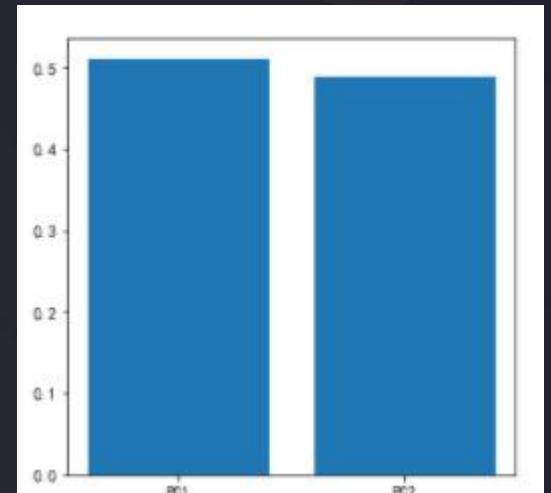
模型预测

结果展示

#主成分分析

```
from sklearn.preprocessing import StandardScaler  
from sklearn.decomposition import PCA  
X_norm = StandardScaler().fit_transform(X)  
pca = PCA(n_components=2)  
X_reduced = pca.fit_transform(X_norm)  
var_ratio = pca.explained_variance_ratio_  
print(var_ratio)  
fig4 = plt.figure(figsize=(5,5))  
plt.bar([1,2],var_ratio)  
plt.xticks([1,2],['PC1','PC2'])  
plt.show()
```

[0.5106415 0.4893585]



芯片品质预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

```
# 数据分离: random_state=4,test_size=0.4
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X,y,random_state=1,test_size=0.4)
print(X_train.shape,X_test.shape,X.shape)
print(X_test)
```

(21, 2) (14, 2) (35, 2)

	x1	x2
30	3.42	7.52
34	0.51	5.26
28	5.68	8.81
3	6.20	0.69
19	6.50	5.04
17	4.53	5.20
21	3.72	3.99
23	1.45	5.36
29	4.74	7.90
26	5.13	6.50
27	6.07	7.31
33	1.28	6.90
24	2.65	6.61
25	3.85	6.26
1	6.79	1.16

芯片品质预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

#knn模型

```
from sklearn.neighbors import KNeighborsClassifier  
knn_3 = KNeighborsClassifier(n_neighbors=3)  
knn_3.fit(X_train,y_train)  
y_train_predict = knn_3.predict(X_train)  
y_test_predict = knn_3.predict(X_test)
```

#计算准确率

```
from sklearn.metrics import accuracy_score  
accuracy_train = accuracy_score(y_train,y_train_predict)  
accuracy_test = accuracy_score(y_test,y_test_predict)  
print("trianing accuracy:",accuracy_train)  
print('testing accuracy:',accuracy_test)
```

```
trianing accuracy: 0.8095238095238095  
testing accuracy: 0.7857142857142857
```

芯片品质预测

数据加载及展示

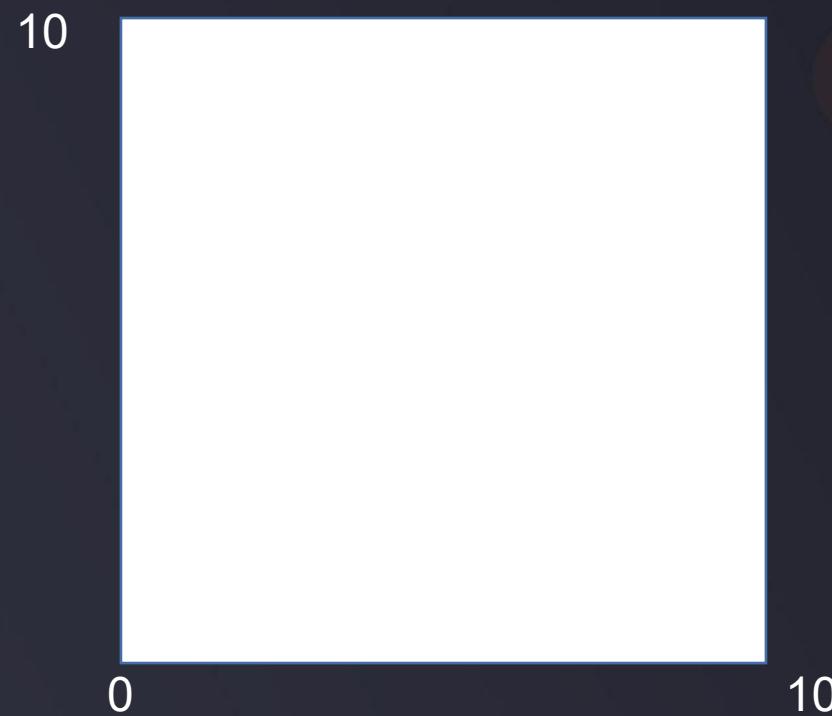
数据预处理

模型建立及训练

模型预测

结果展示

```
### 生成边界数据集  
xx, yy = np.meshgrid(np.arange(0,10,0.05),np.arange(0,10,0.05))  
#数据组合  
x_range = np.c_[xx.ravel(),yy.ravel()]  
print(x_range.shape, x_range)
```



```
(40000, 2)  
[[0.  0. ]  
 [0.05 0. ]  
 [0.1  0. ]  
 ...  
 [9.85 9.95]  
 [9.9  9.95]  
 [9.95 9.95]]
```

芯片品质预测

数据加载及展示

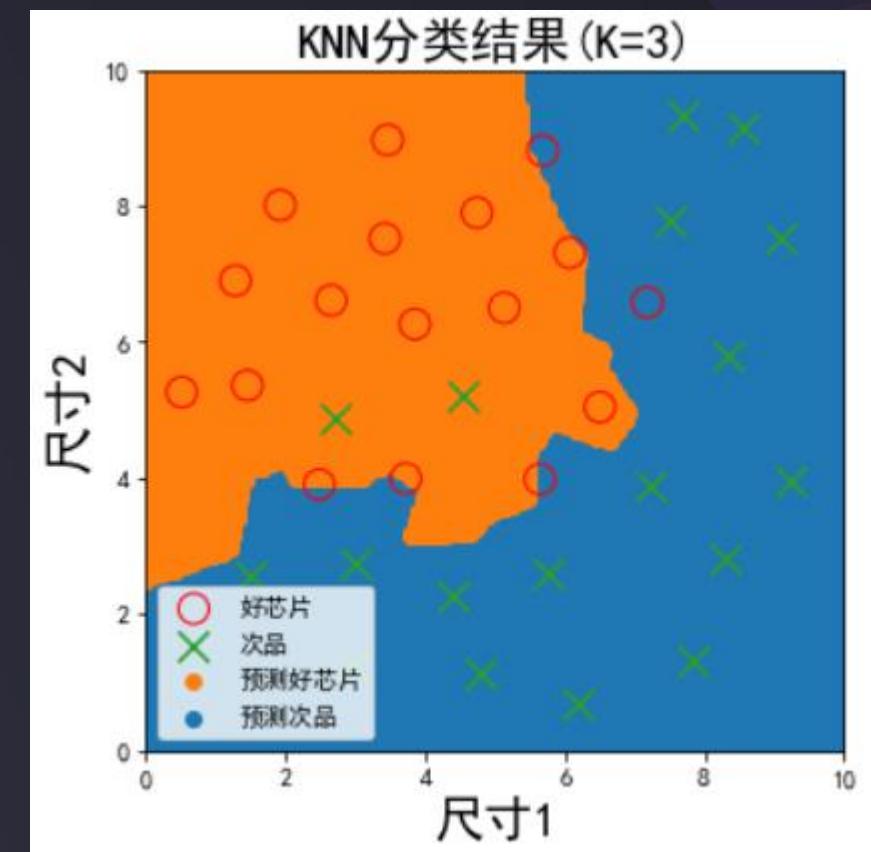
数据预处理

模型建立及训练

模型预测

结果展示

```
#预测新数据对应类别  
y_range_predict = knn_3.predict(x_range)
```



芯片品质预测

数据加载及展示

数据预处理

模型建立及训练

模型预测

结果展示

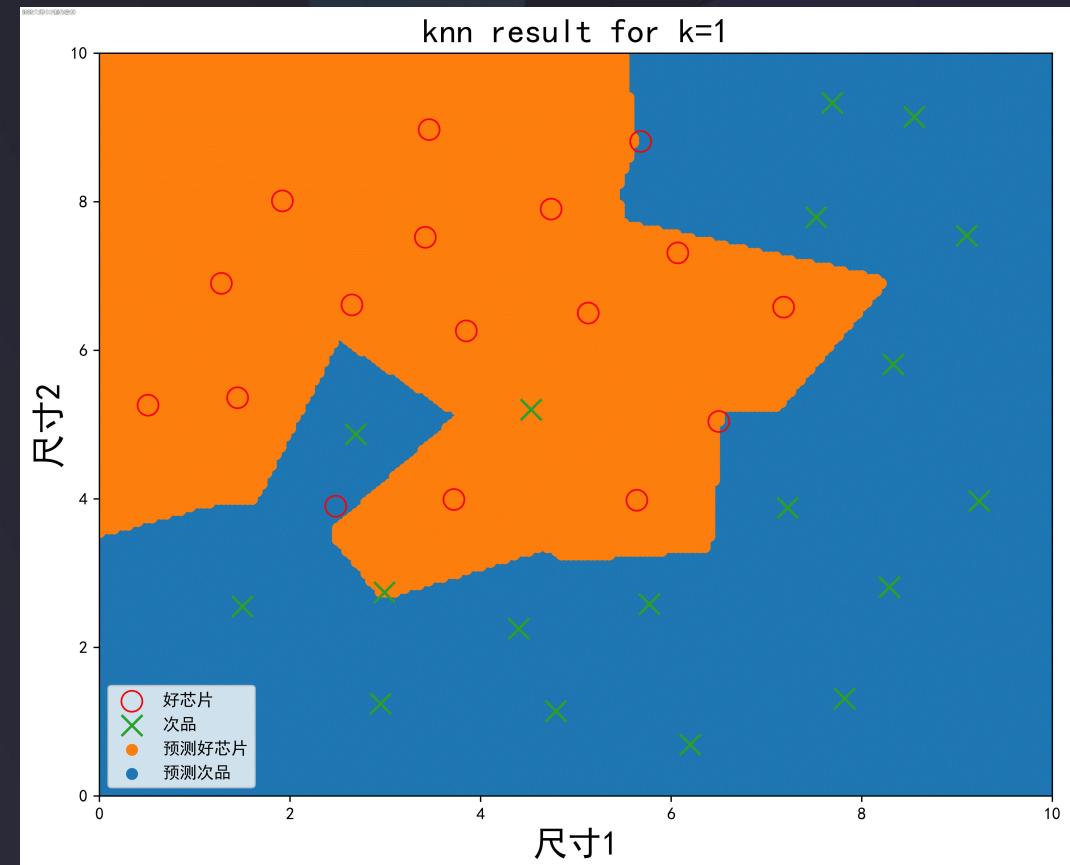
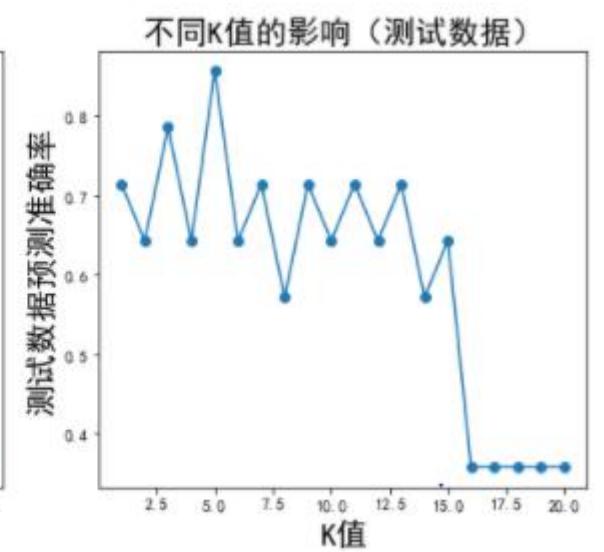
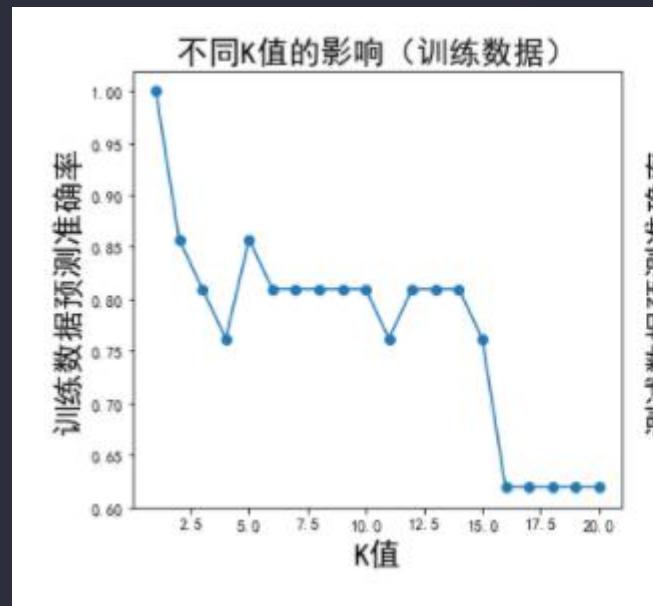
```
#计算混淆矩阵  
from sklearn.metrics import confusion_matrix  
cm = confusion_matrix(y_test,y_test_predict)  
print(cm)
```

```
[[4 1]  
 [2 7]]
```

```
#获取混淆矩阵各元素  
TP = cm[1,1]  
TN = cm[0,0]  
FP = cm[0,1]  
FN = cm[1,0]  
print(TP,TN,FP,FN)
```

```
7 4 1 2
```

芯片品质预测





Python3人工智能入门+实战提升：机器学习

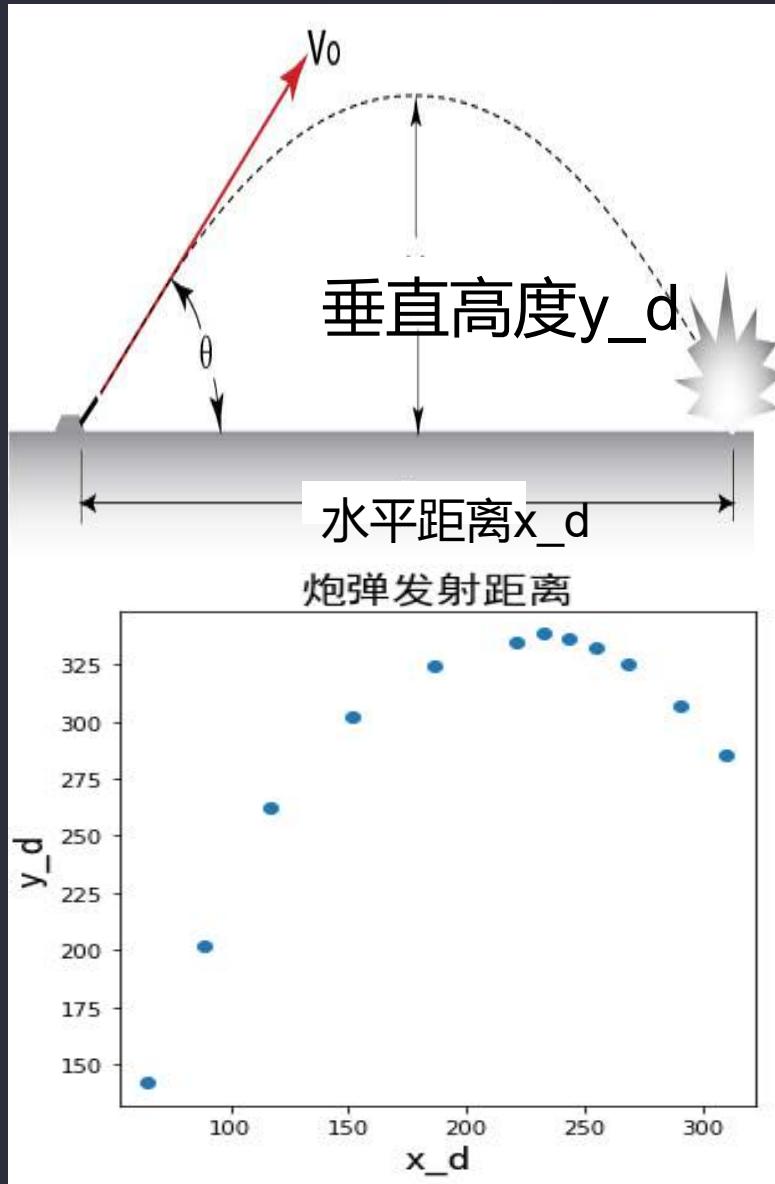
Chapter 7 模型选择与优化

赵辛

Chapter 7 模型选择与优化

-
- 1 --模型过拟合与欠拟合
 - 2 --模型过拟合解决
 - 3 --数据分离与混淆矩阵
 - 4 --模型选择与优化
 - 5 --实战准备
 - 6 --实战（一）
 - 7 --实战（二）

| 任务一：炮弹发射轨迹预测



基于task1_train_data数据，建立回归模型，预测炮弹高度。

- 1、基于task1_train_data数据，建立线性回归模型，计算其在task1_test_data数据上的r2分数，可视化模型预测结果
- 2、分别引入2次、6次多项式属性数据，建立回归模型
- 3、对比三个模型对训练数据、测试数据集做预测的r2分数，判断哪个模型预测更准确
- 4、可视化三个模型的预测曲线，判断哪个模型预测更准确



Python3人工智能入门+实战提升：机器学习

Chapter 7 模型选择与优化

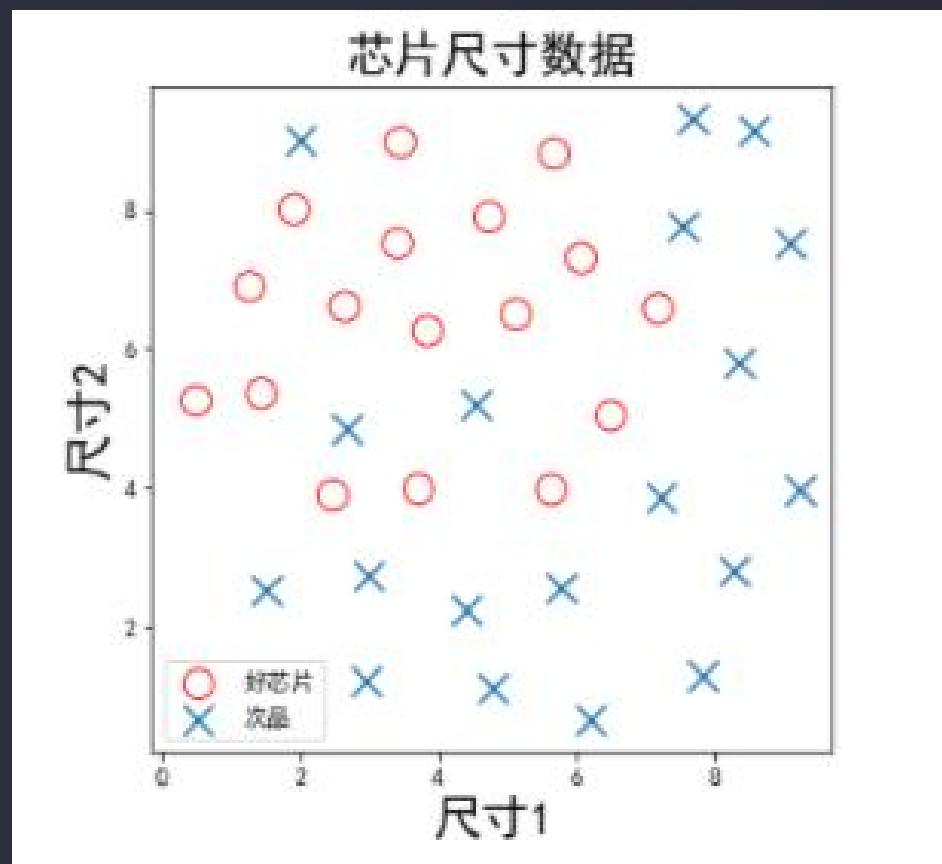
赵辛

Chapter 7 模型选择与优化

-
- 1 --模型过拟合与欠拟合
 - 2 --模型过拟合解决
 - 3 --数据分离与混淆矩阵
 - 4 --模型选择与优化
 - 5 --实战准备
 - 6 --实战（一）
 - 7 --实战（二）

任务二：芯片品质预测

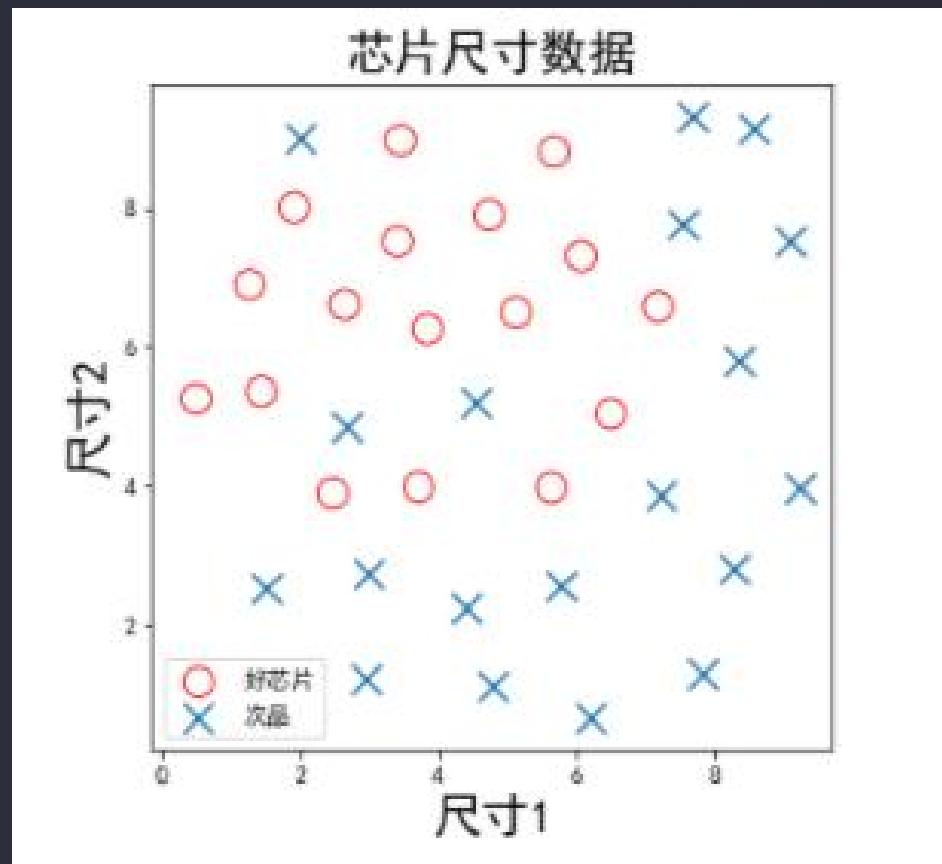
基于task2_data数据，综合异常数据检测、PCA降维、数据分离、KNN等技术完成芯片品质预测。



- 1、基于高斯分布概率密度函数，对两个维度数据进行分析、计算概率密度函数，寻找异常点并剔除
- 2、统计分析各维度数据分布
- 3、对数据进行主成分分析，计算各维度方差比例
- 4、数据分离，数据分离参数：
`random_state=1,test_size=0.4`

任务二：芯片品质预测

基于task2_data数据，综合异常数据检测、PCA降维、数据分离、KNN等技术完成芯片品质预测。



5、建立KNN模型 ($K=3$) 完成分类，可视化分类边界

6、计算测试数据集对应的混淆矩阵，准确率、召回率、特异度、精确率、F1分数

7、尝试不同的K值 (1-20)，计算其在训练数据集、测试数据集上的准确率并作图

任务拓展：尝试其他模型完成预测：决策树、逻辑回归、朴素贝叶斯

| 任务三：为毕业与工作做准备

待添加?

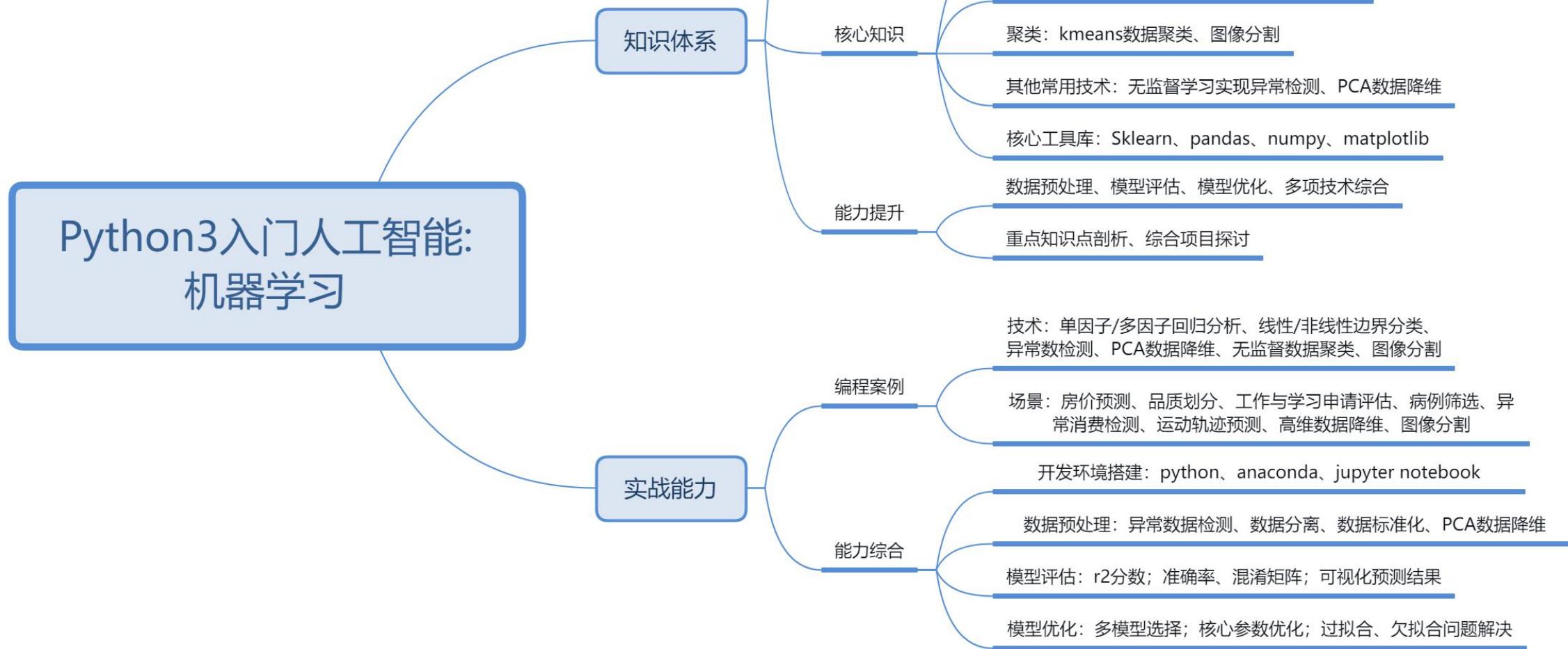


Python3人工智能入门+实战提升：机器学习

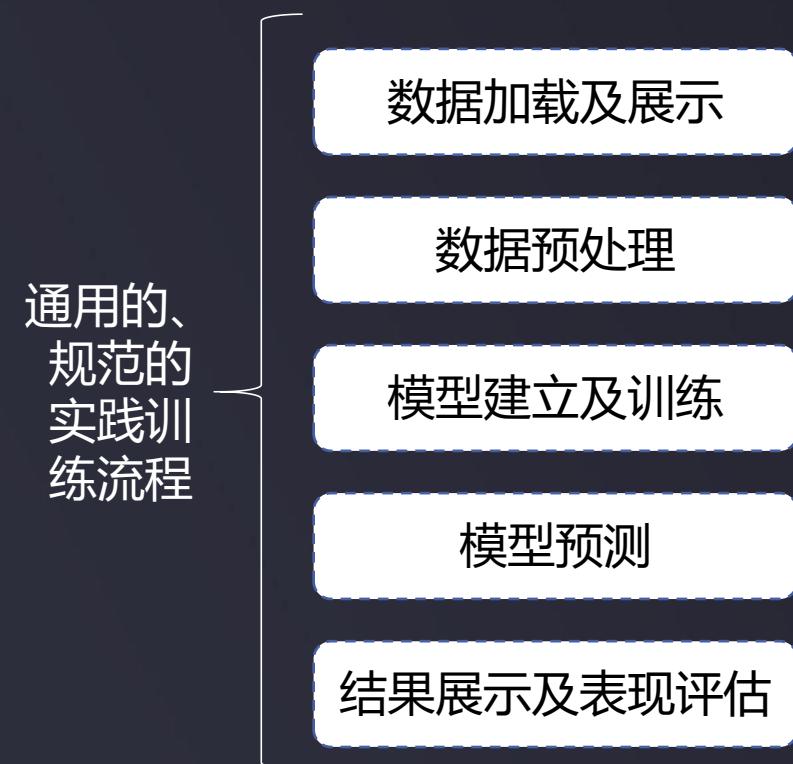
课程总结

赵辛

核心知识点与工具



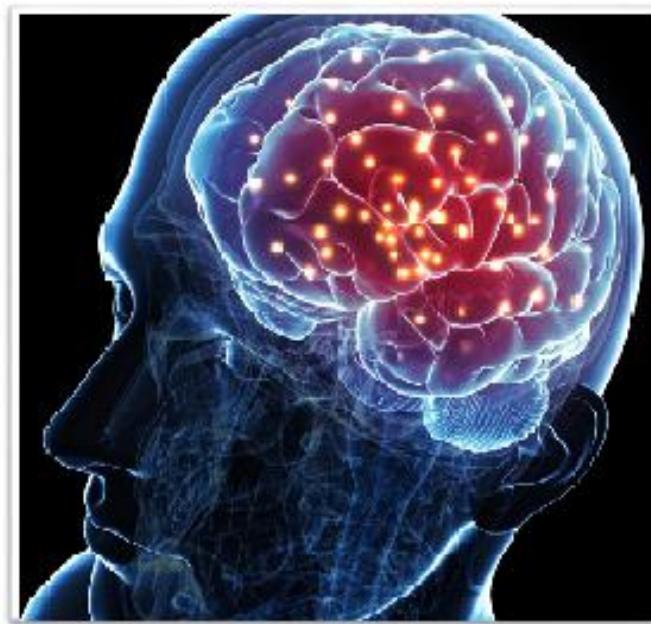
核心流程



ARTIFICIAL



INTELLIGENCE



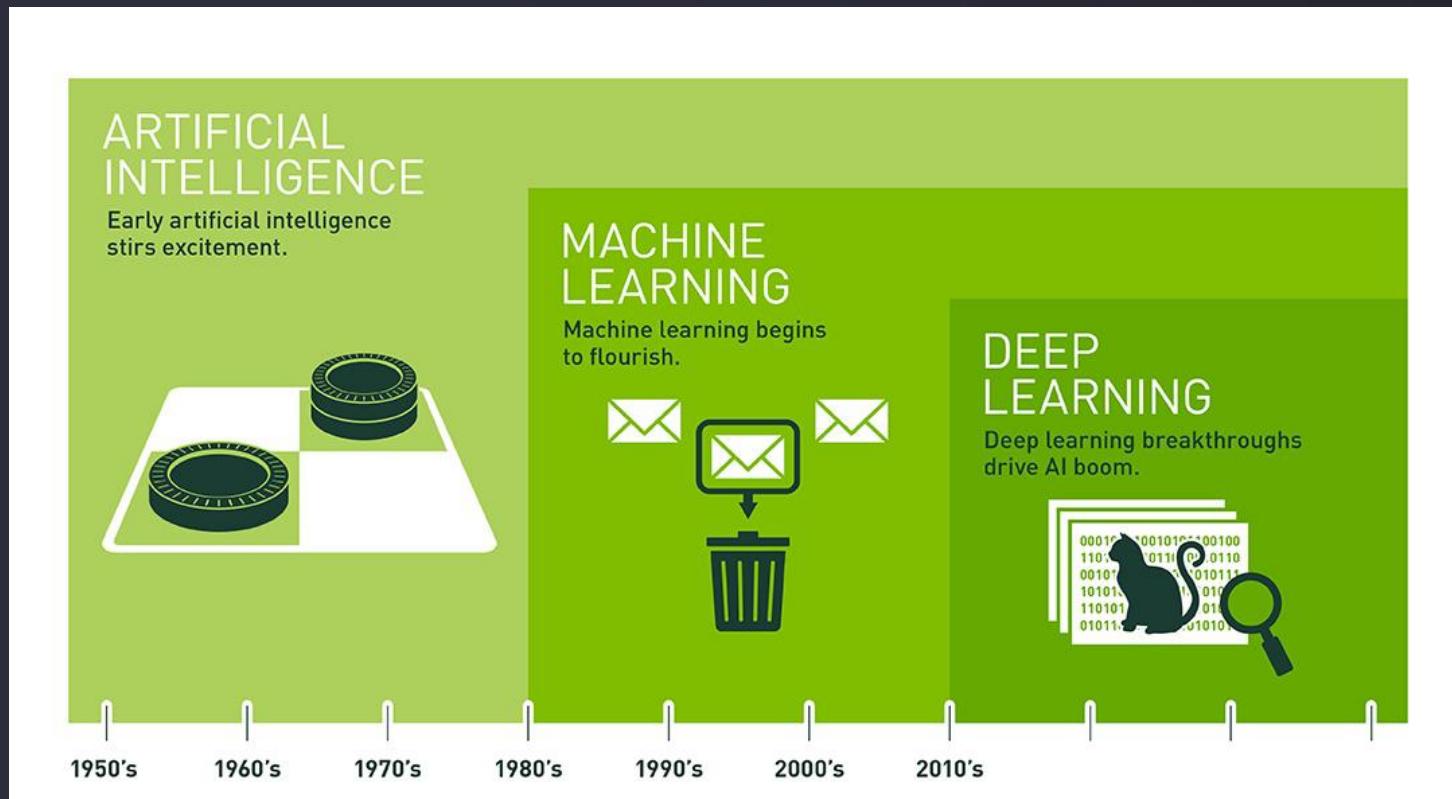
ARTIFICIAL
INTELLIGENCE



Intelligence: “The capacity to learn and solve problems”
(自主学习及解决问题的能力)

Artificial Intelligence: The simulation of human intelligence by machines
(机器对人类智能的模仿)

机器学习与深度学习的关系



机器学习是一种实现人工智能的方法，
深度学习是一种实现机器学习的技术。

机器学习：从数据中自动分析获得规律，并利用规律对未知数据进行预测或用于解决实际问题的方法。

深度学习：机器在对数据进行分析时，将引入类人类神经结构模型，实现对复杂数据的理解与推理，通常可应用于更为复杂的任务中。

| 回归分析与线性回归

传统算法：



机器学习：



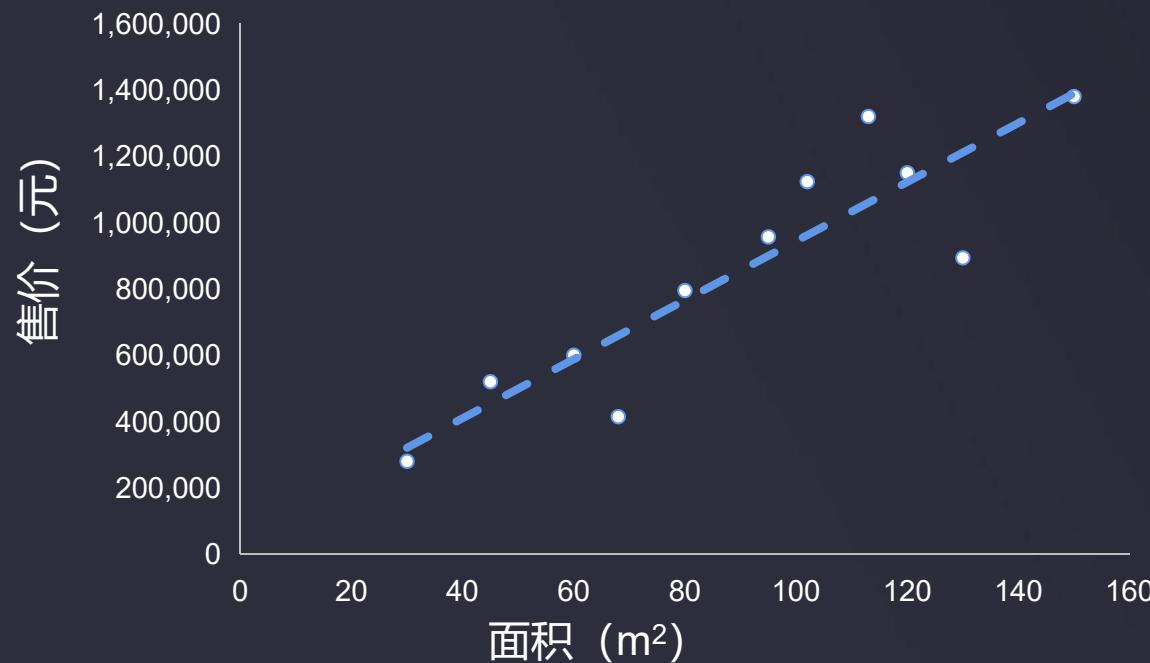
机器学习应用与概念——四大学习方法

- 监督学习 (Supervised Learning)
- 无监督学习 (Unsupervised Learning)
- 半监督学习 (Semi-supervised Learning)
- 强化学习 (Reinforcement Learning)

训练是否有
正确的结果
(标签-label)

回归分析与线性回归

问题：面积100平米售价120万是否值得投资？



$$J = \frac{1}{2m} \sum_{i=1}^m (y' - y)^2 = \frac{1}{2m} \sum_{i=1}^m (ax + b - y)^2 = g(a, b)$$

$\boxed{\text{minimize}(J)}$

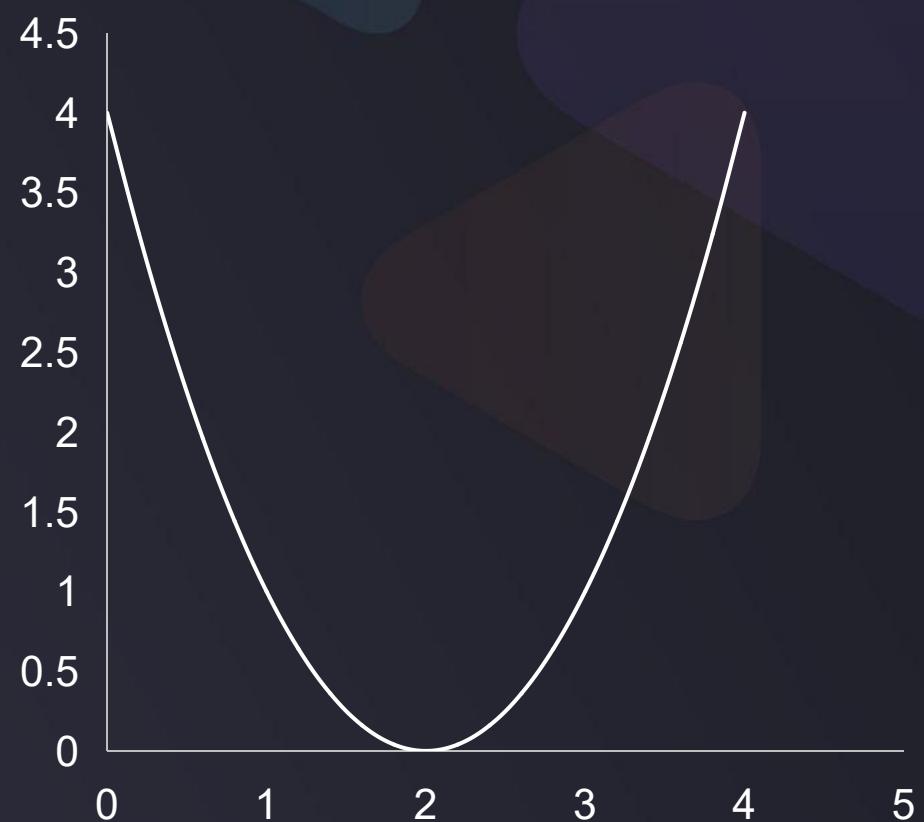
回归分析与线性回归

$$J = \frac{1}{2m} \sum_{i=1}^m (y' - y)^2 = \frac{1}{2m} \sum_{i=1}^m (ax + b - y)^2 = g(a, b)$$

minimize(J)

梯度下降法：

$$y = f(x) \quad \xrightarrow{\text{搜索方法}} \quad x_{i+1} = x_i - \alpha \frac{\partial}{\partial x_i} f(x_i)$$



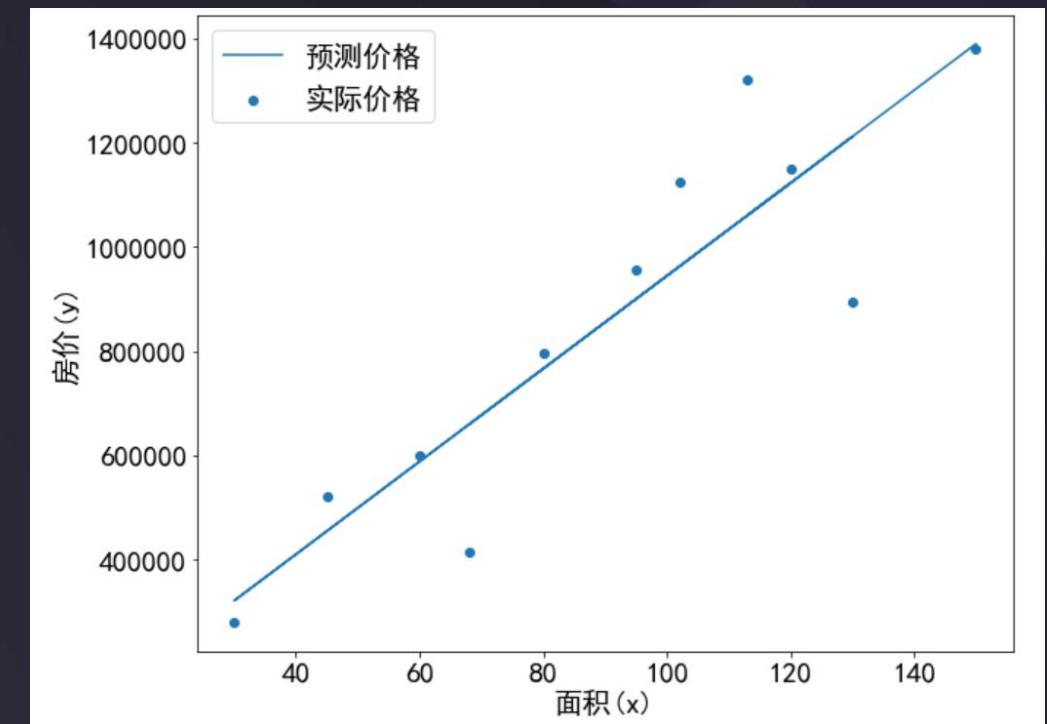
回归分析与线性回归

$$J = \frac{1}{2m} \sum_{i=1}^m (y' - y)^2 = \frac{1}{2m} \sum_{i=1}^m (ax + b - y)^2 = g(a, b)$$

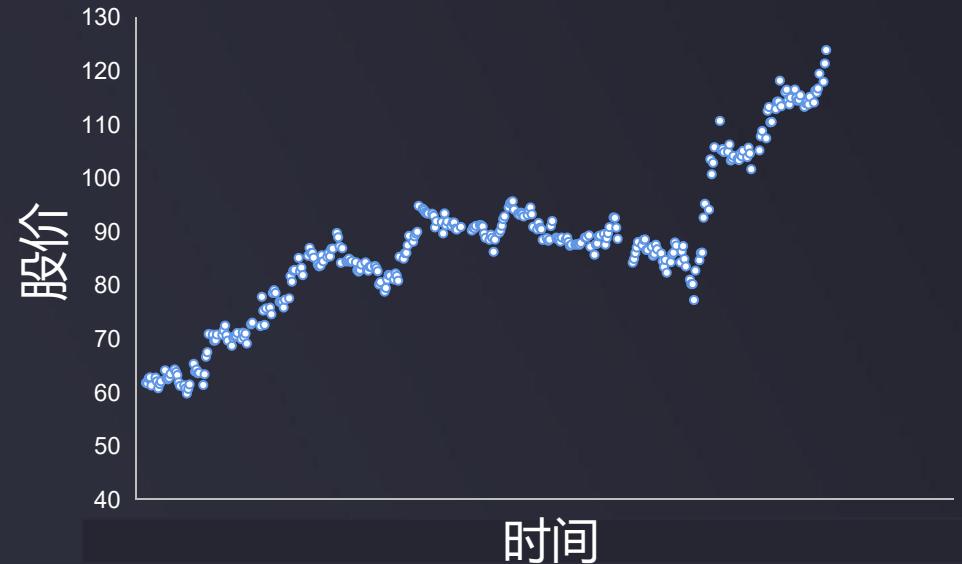
$\boxed{\text{minimize}(J)}$

对每个系数分别使用梯度下降法，重复计算直到收敛

$$\begin{cases} temp_a = a - \alpha \frac{\partial}{\partial a} g(a, b) \\ temp_b = b - \alpha \frac{\partial}{\partial b} g(a, b) \\ a = temp_a \\ b = temp_b \end{cases}$$



通过股价预测任务区分回归任务与分类任务

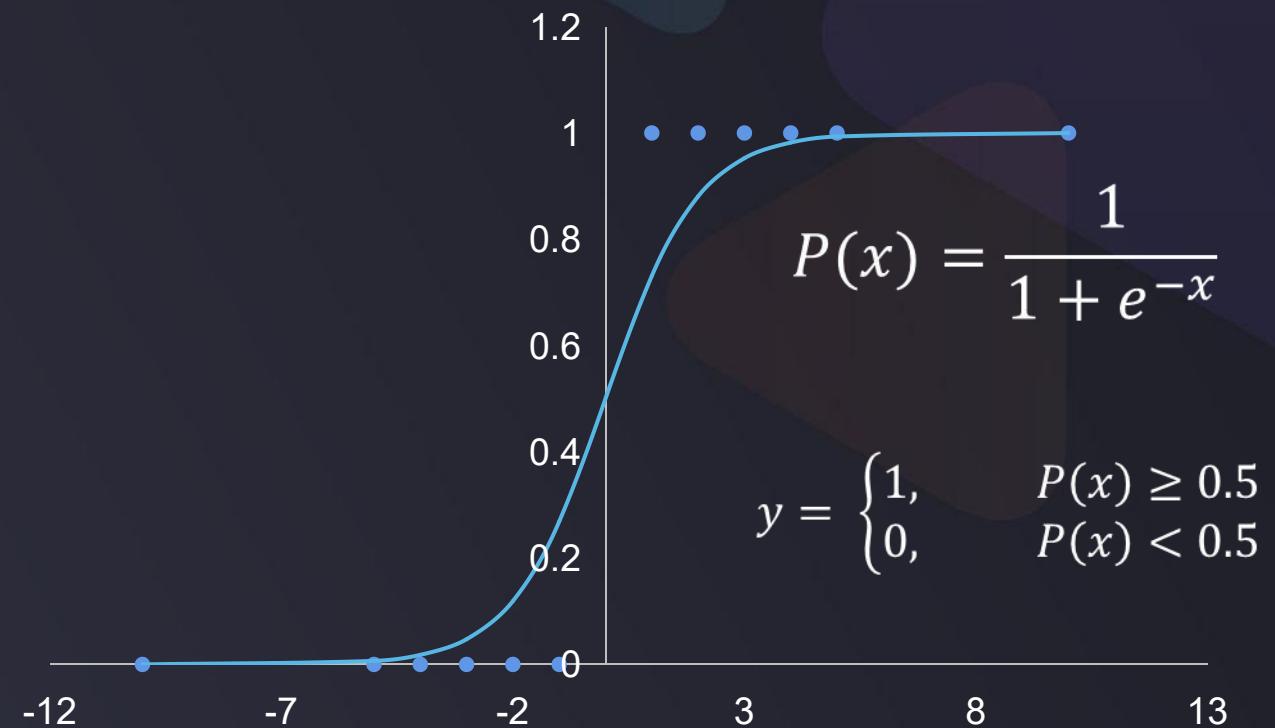
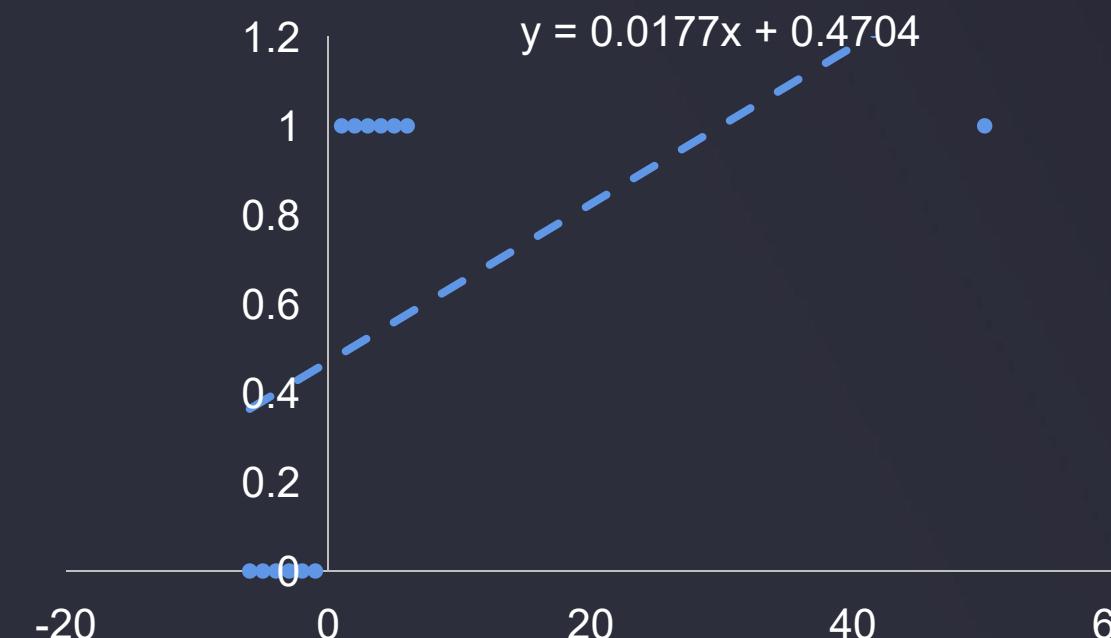


回归：连续性数值预测
模型输出：连续型数值
(明天股价预测为：125.1)

分类：非连续性判断类别
模型输出：非连续型标签
(明天股价预测为：上涨)

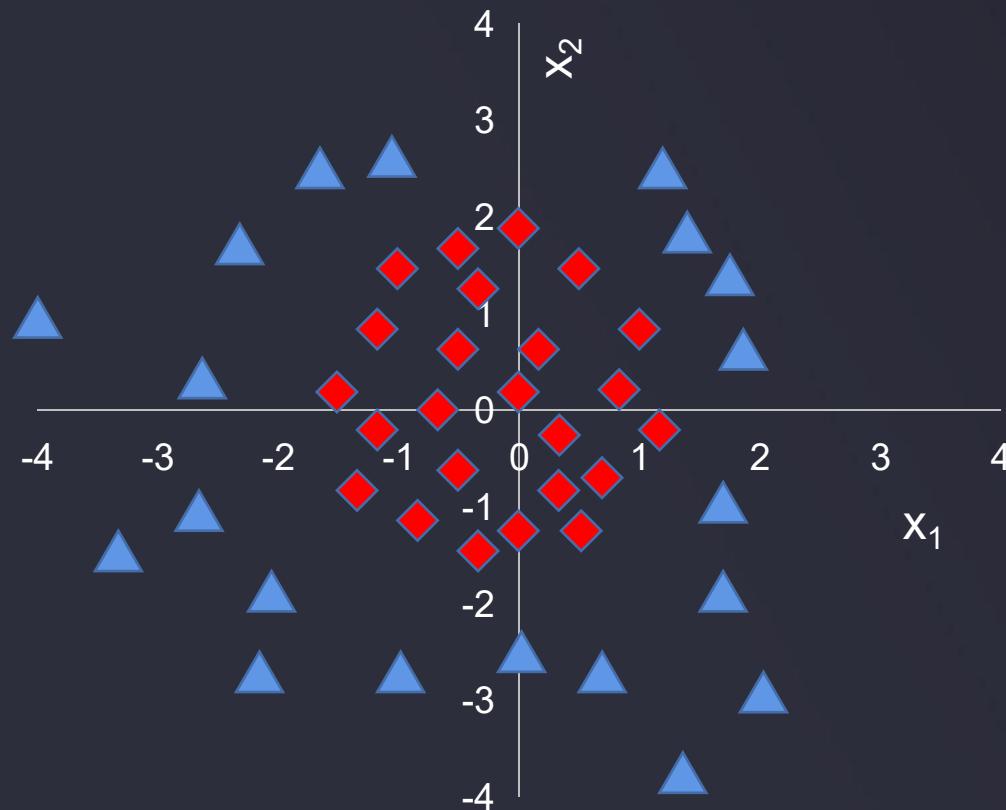
分类预测

任务：根据水位，判断水池是否需要蓄水或放水



思考：需要更适合于分类场景的模型

分类问题与逻辑回归



➤ 逻辑回归结合多项式边界函数可解决复杂的分类问题

$$P(x) = \frac{1}{1 + e^{-g(x)}}$$

$$g(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 \dots$$

➤ 模型求解的核心，在于寻找到合适的多项式边界函数

$$g(x) = x_1^2 + x_2^2 - 4$$

分类问题与逻辑回归

- 寻找损失函数极小值点
- 分类问题，结果为离散数据，需要对损失函数进行调整以适应梯度下降法求解

逻辑回归求解，最小化损失函数 (J) ：

$$J_i = \begin{cases} -\log(P(x_i)), & \text{if } y_i = 1 \\ -\log(1 - P(x_i)), & \text{if } y_i = 0 \end{cases}$$

$$J = \frac{1}{m} \sum_{i=1}^m J_i = -\frac{1}{m} \left[\sum_{i=1}^m (y_i \log(P(x_i)) + (1 - y_i) \log(1 - P(x_i))) \right]$$

分类问题与逻辑回归

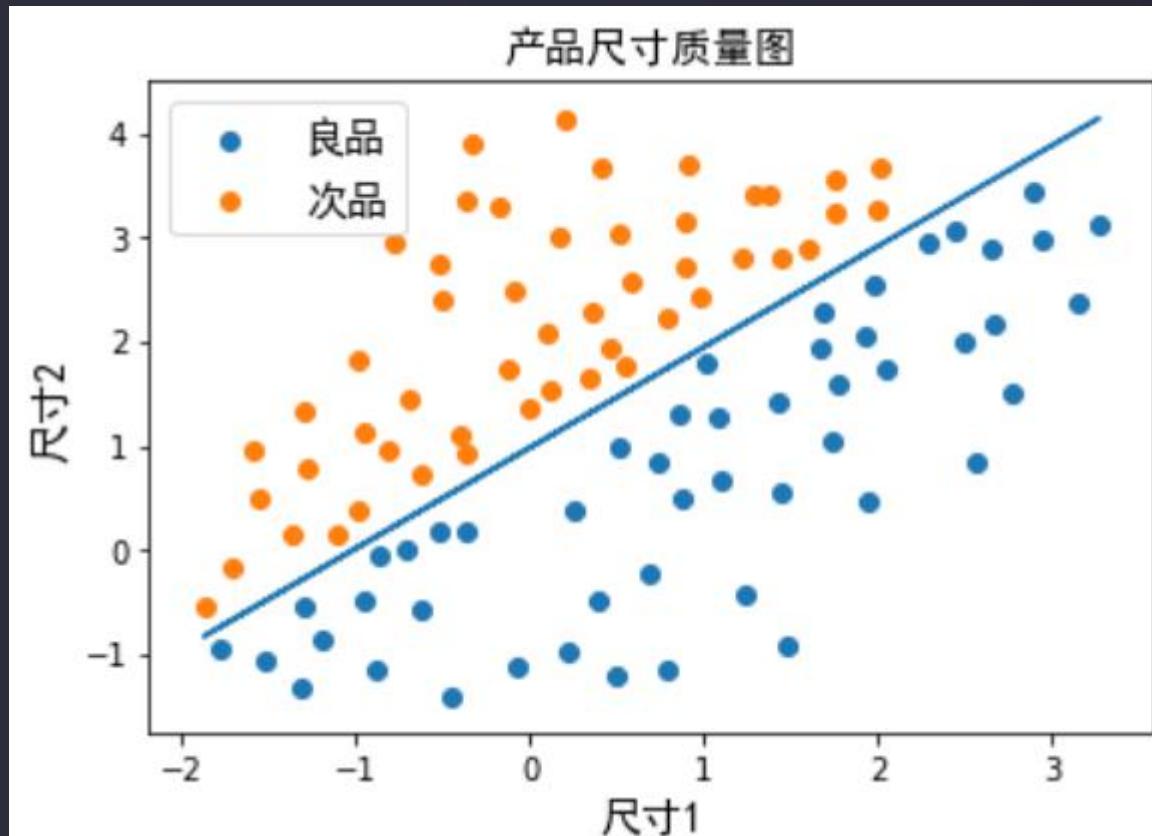
$$P(x) = \frac{1}{1 + e^{-g(x)}}$$

$$g(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 \dots$$

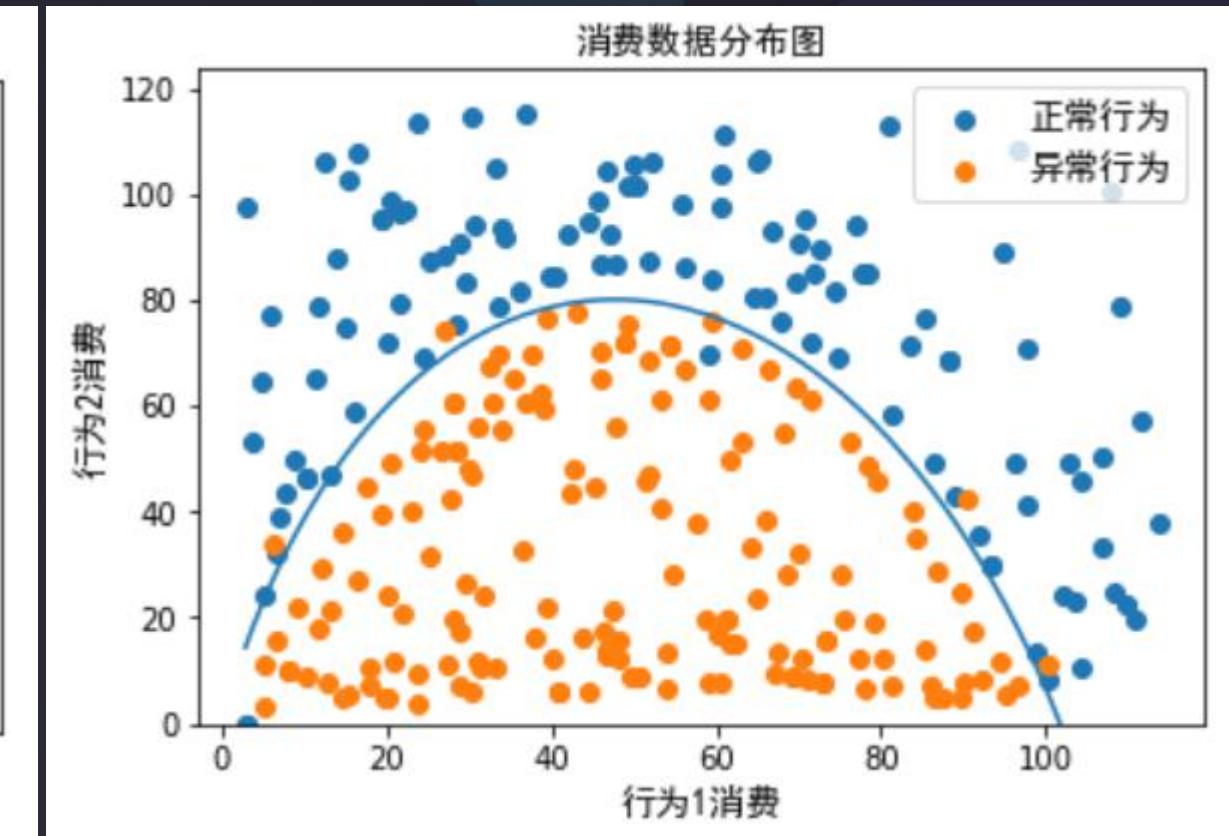
重复计算直到收敛

$$\left\{ \begin{array}{l} temp_{\theta_j} = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\ \theta_j = temp_{\theta_j} \end{array} \right\}$$

分类问题与逻辑回归



逻辑回归产品质量预测



逻辑回归商业异常消费行为预测

| 毕设、工作综合项目的思考与建议

如果毕业设计是搭建分类模型，预测消费者是否会购买商品，我们通常做些什么？

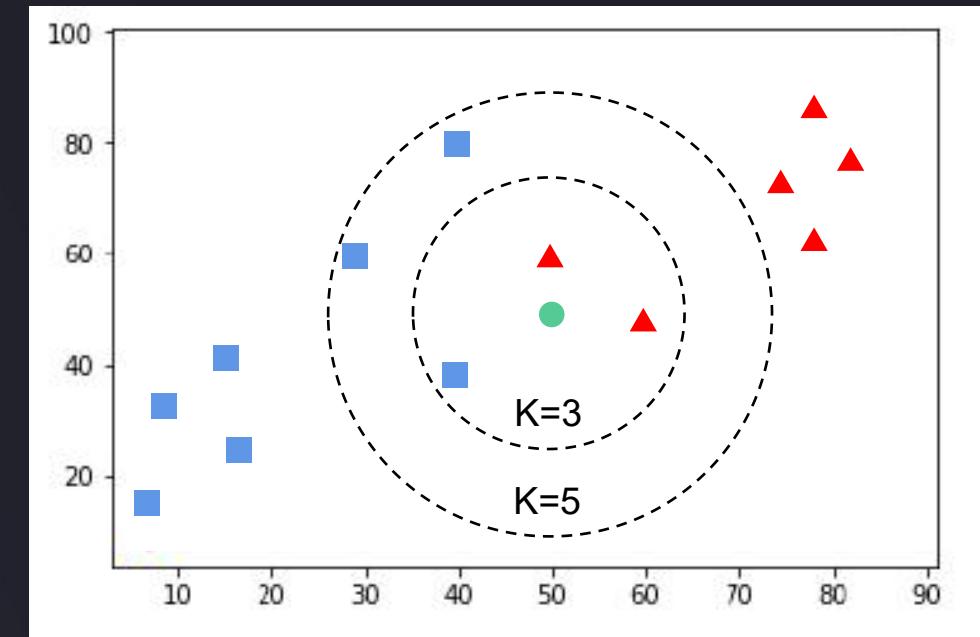
- 1、前期工作：调研、讨论并确认影响购买意愿的因素；数据采集
- 2、数据预处理：异常数据处理、信息量化
- 3、建模与训练：从简单到复杂的决策边界模型
- 4、预测、评估、优化：引入不同的评估指标、尝试不同的模型
- 5、总结与汇报：结果整理并分析、输出项目报告

K近邻分类模型(K-nearest neighbors)

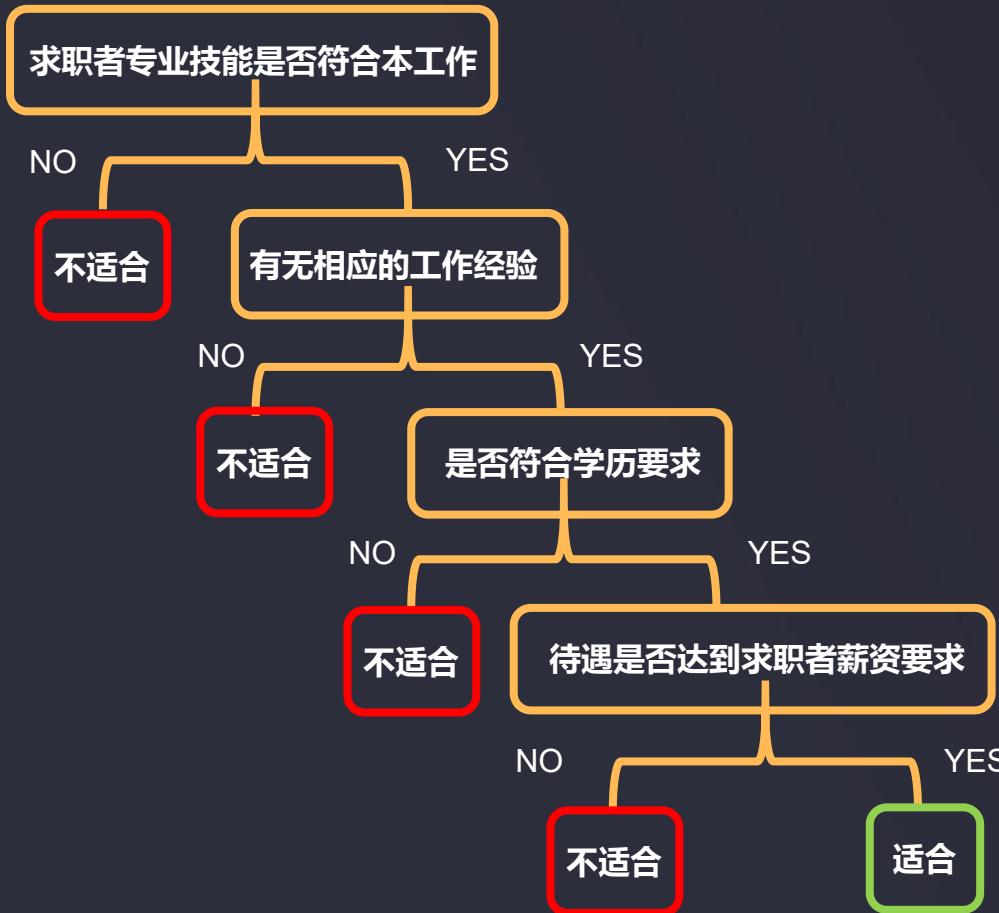
举例：判断图中圆圈属于哪个类别

$K=3$, 绿色圆点(50,50)的最近的3个邻居是2个红色小三角形(60,50)、(50,60)和1个蓝色小正方形(40,40), 判定其属于红色的三角形一类。

$K=5$, 绿色圆点的最近的5个邻居是2个红色三角形(60,50)、(50,60)和3个蓝色的正方形(40,40)、(40,80)、(30,60), 判定其属于蓝色的正方形一类。



决策树



决策树模型框架

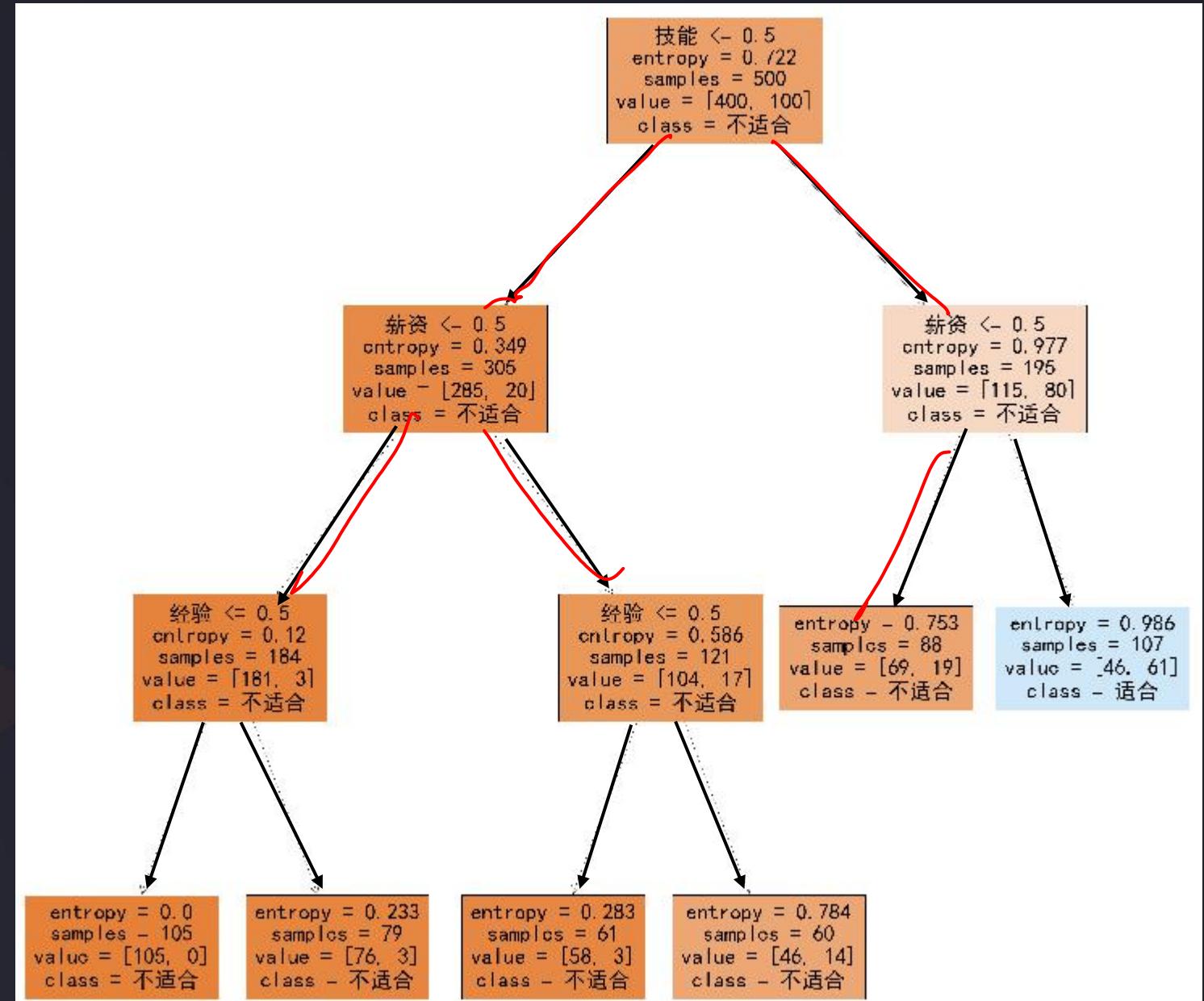
信息熵 (entropy) 是度量随机变量不确定性的指标，**熵越大，样本的不确定性就越大**。假定当前样本集合D中第k类样本所占的比例为 p_k ，则D的信息熵为：

$$Ent(D) = - \sum_{k=1}^{|D|} p_k \log_2 p_k$$

Skill	Experience	Degree	Income	y
2	0	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
1	0	1	1	1
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	1

决策树判断员工是否适合相关工作

决策树-预测-面试者是否适合岗位



朴素贝叶斯

条件概率公式：

$$P(B|A) = \frac{P(AB)}{P(A)}$$

全概率公式：

$$P(A) = \sum_{i=1}^n P(A|B_i)P(B_i)$$

贝叶斯公式：

$$P(B|A) = P(B) * \frac{P(A|B)}{P(A)}$$

概率是反映随机事件出现的可能性大小的量度,而条件概率则是给定某事件A的条件下,另一事件B发生的概率。全概率公式则是利用条件概率,将复杂事件A分割为若干简单事件概率的求和问题。贝叶斯公式则是利用条件概率和全概率公式计算后验概率。

朴素贝叶斯用于机器学习

$$P(Y|X) = P(Y) * \frac{P(X|Y)}{P(X)}$$

现实案例的输入特征高于1维，假设特征之间相互独立：

朴素贝叶斯公式：

$$P(X|Y = y_i) = \prod_{j=1}^m P(x_j|Y = y_i)$$

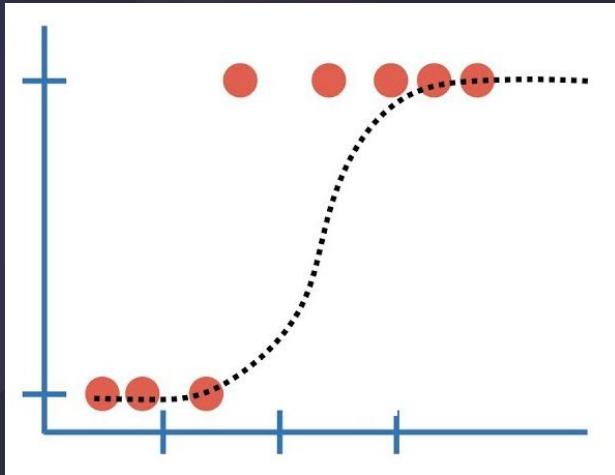
$$P(y_i|x_1, x_2 \dots, x_m) = \frac{P(y_i) \prod_{j=1}^m P(x_j|y_i)}{P(x_1, x_2 \dots, x_m)} = \frac{P(y_i) \prod_{j=1}^m P(x_j|y_i)}{\prod_{j=1}^m P(x_j)}$$

朴素贝叶斯-预测-学生录取及奖学金情况

学员信息 (测试样本)					预测概率			预测结果
成绩	学校	获奖	性别	英语	未录取	无奖学金录取	带奖学金录取	
2	1	1	1	1	0.152	0.346	0.502	带奖学金录取
2	1	1	1	0	0.203	0.400	0.397	无奖学金录取
2	1	1	0	0	0.158	0.455	0.387	无奖学金录取
2	1	0	0	0	0.388	0.447	0.166	无奖学金录取
2	0	0	0	0	0.595	0.293	0.112	未录取

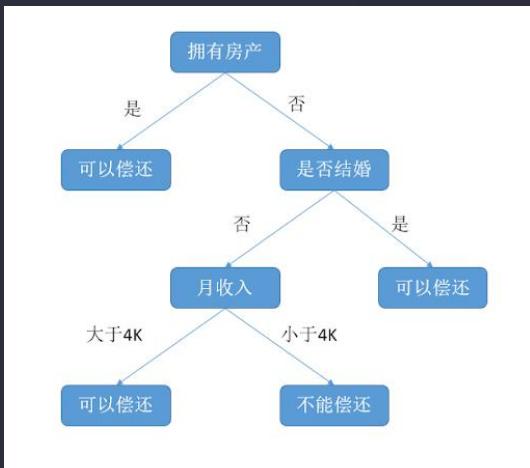
常用分类方法

适用场景：
需要较为清晰地理解每个属性对结果的影响



逻辑回归(logistics regression)

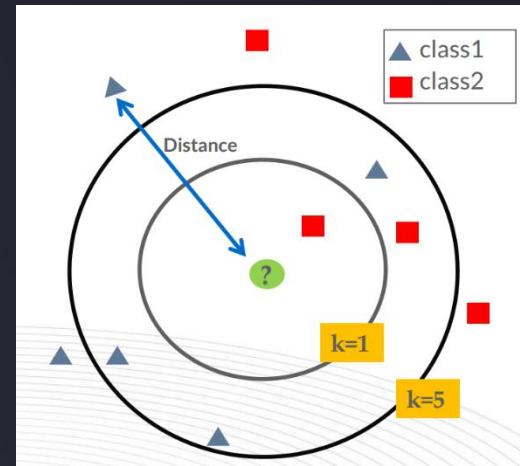
适用场景：
需要清晰地描述类别判断的前后逻辑 (先依据哪个指标判断, 接下来使用哪个指标)



决策树(decision tree)

没有最好的分类器, 只有最合适的分类器

适用场景：
需要一个特别容易解释的模型的时候, 比如需要向用户解释原因的推荐算法



KNN近邻模型(K-nearest neighbors)

朴素贝叶斯

$$P(Y|X) = P(Y) * \frac{P(X|Y)}{P(X)}$$

适用场景：
数据不同维度之间相关性较小

$$P(y_i|x_1, x_2, \dots, x_n) = \frac{P(y_i) \prod_{j=1}^n P(x_j|y_i)}{\prod_{j=1}^n P(x_j)}$$

无监督学习与聚类

目标：

以下六组图片，按照自己喜爱的方式分成两组



分组一：站着或非站着

分组二：白色或黄色

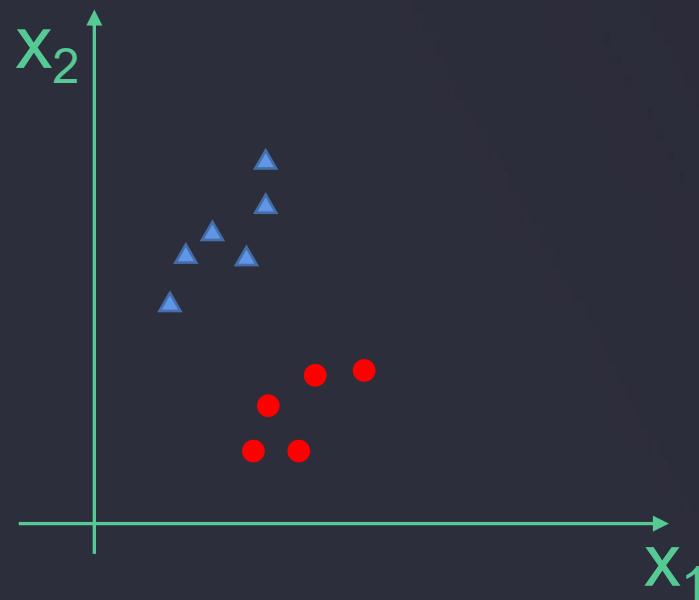
分组三：吐舌头或不吐舌头

- 没有绝对的对错标准
- 寻找数据特征的相似性

无监督学习

无监督学习与聚类

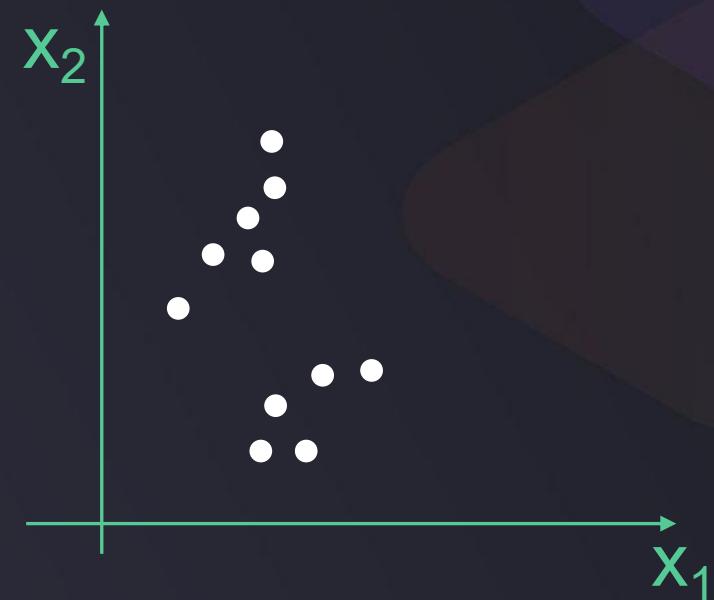
Supervised



训练数据：

$$\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \dots, (\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\}$$

Unsupervised



训练数据：

$$\{(\mathbf{x}^{(1)}), (\mathbf{x}^{(2)}), \dots, (\mathbf{x}^{(m)})\}$$

无监督学习 (Unsupervised Learning)

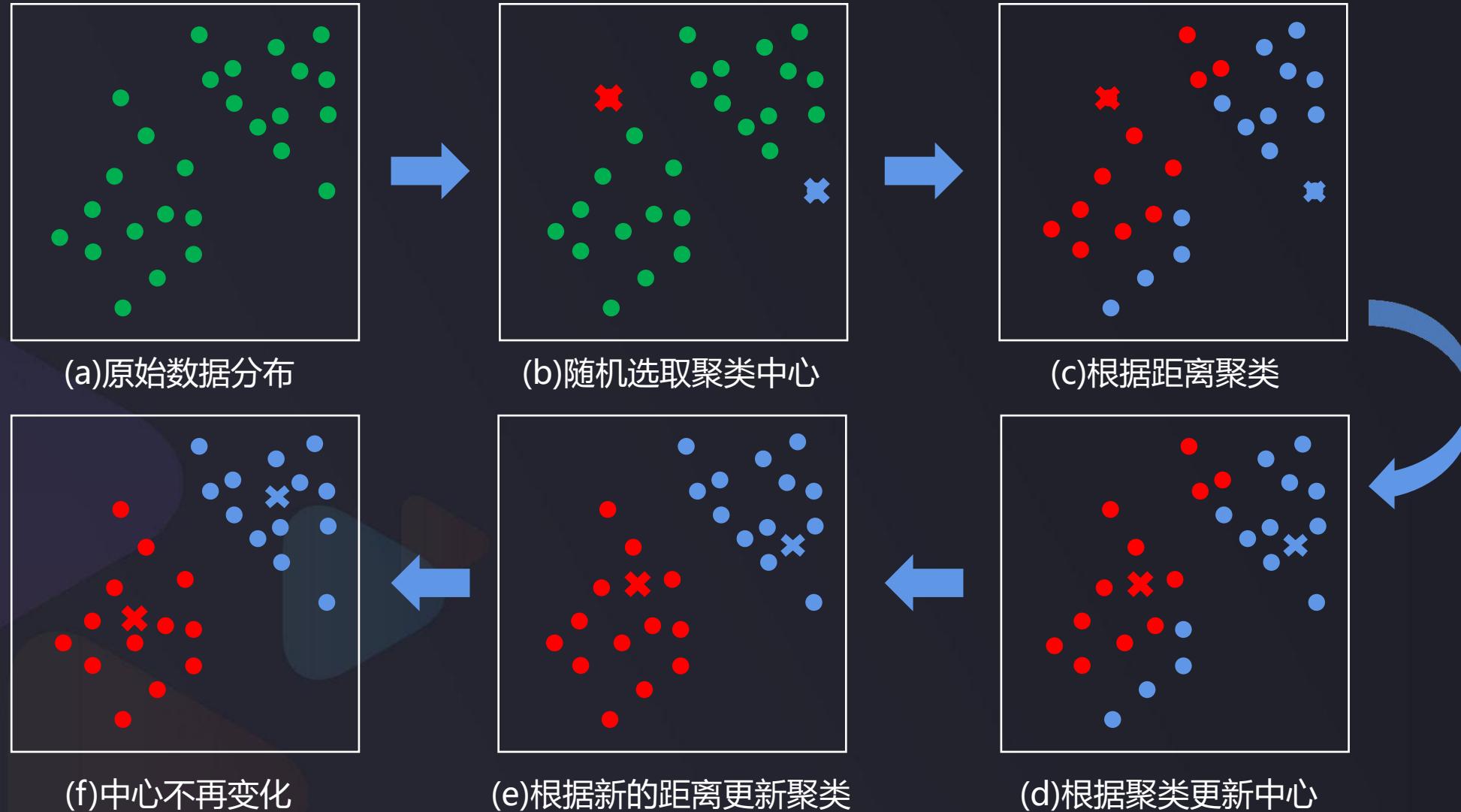
特点:

- 数据不需要标签
- 算法不受监督信息（偏见）约束

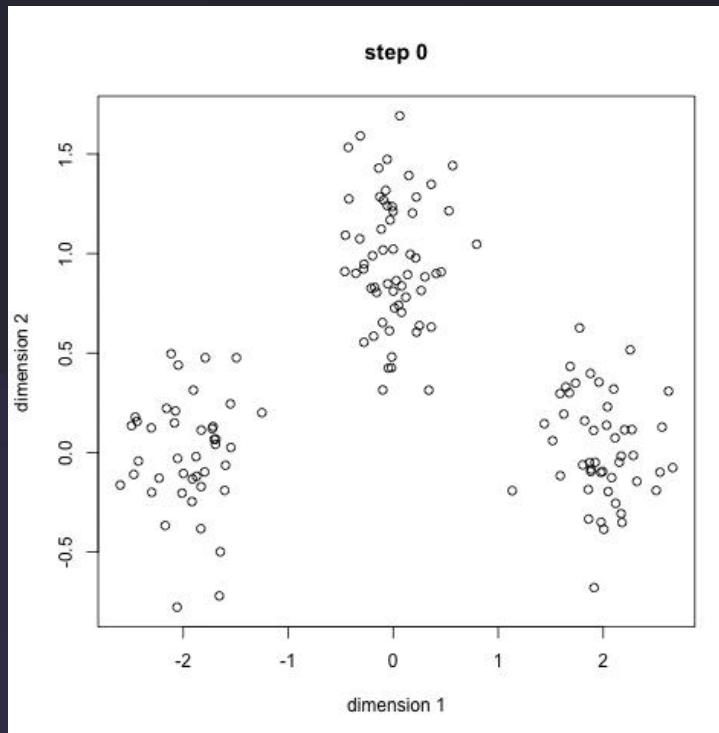
优点:

- 降低数据采集难度，极大程度扩充样本量
- 可能发现新的数据规律、被忽略的重要信息

K均值聚类

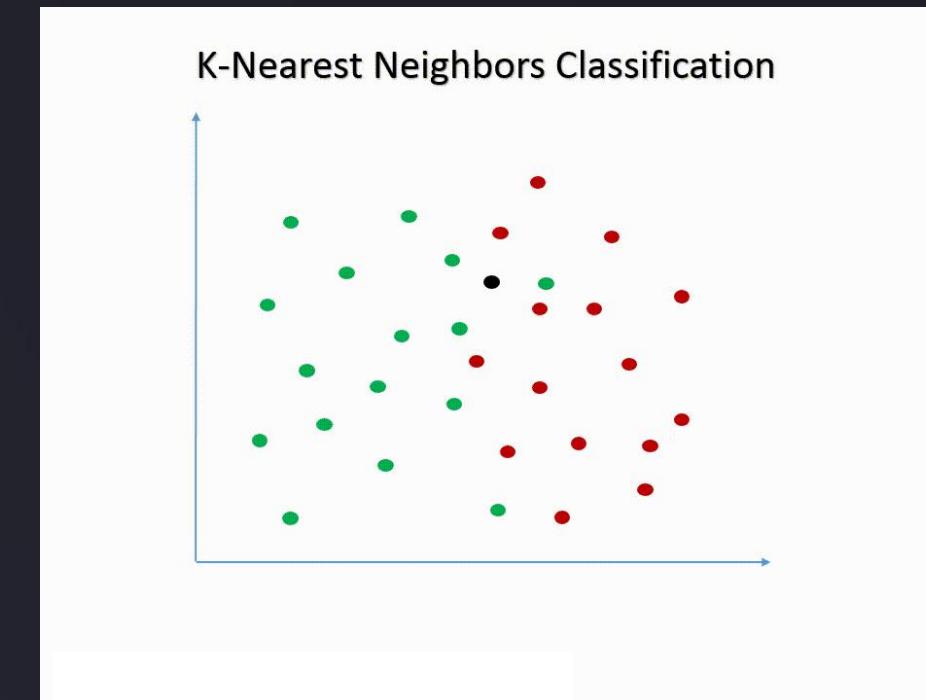


K均值聚类 (KMeans) VS K近邻分类 (KNN)



KMeans

- 无监督学习
- 聚类算法
- 无Label的数据集
- 计算数据与中心点距离



KNN

- 监督学习
- 分类算法
- 带Label的数据集
- 计算数据与其他数据的距离

K均值聚类实现图像分割



(1)原图



```
[[[254 255 250]
 [255 255 251]
 [246 247 241]
 ...
 [255 255 246]
 [255 255 250]
 [254 251 246]]]
```

(140, 140, 3)



```
[[254 255 250]
 [255 255 251]
 [246 247 241]
 ...
 [240 246 246]
 [246 252 248]
 [250 255 250]]]
```

(19600, 3)



(2)k=3



(3)k=8

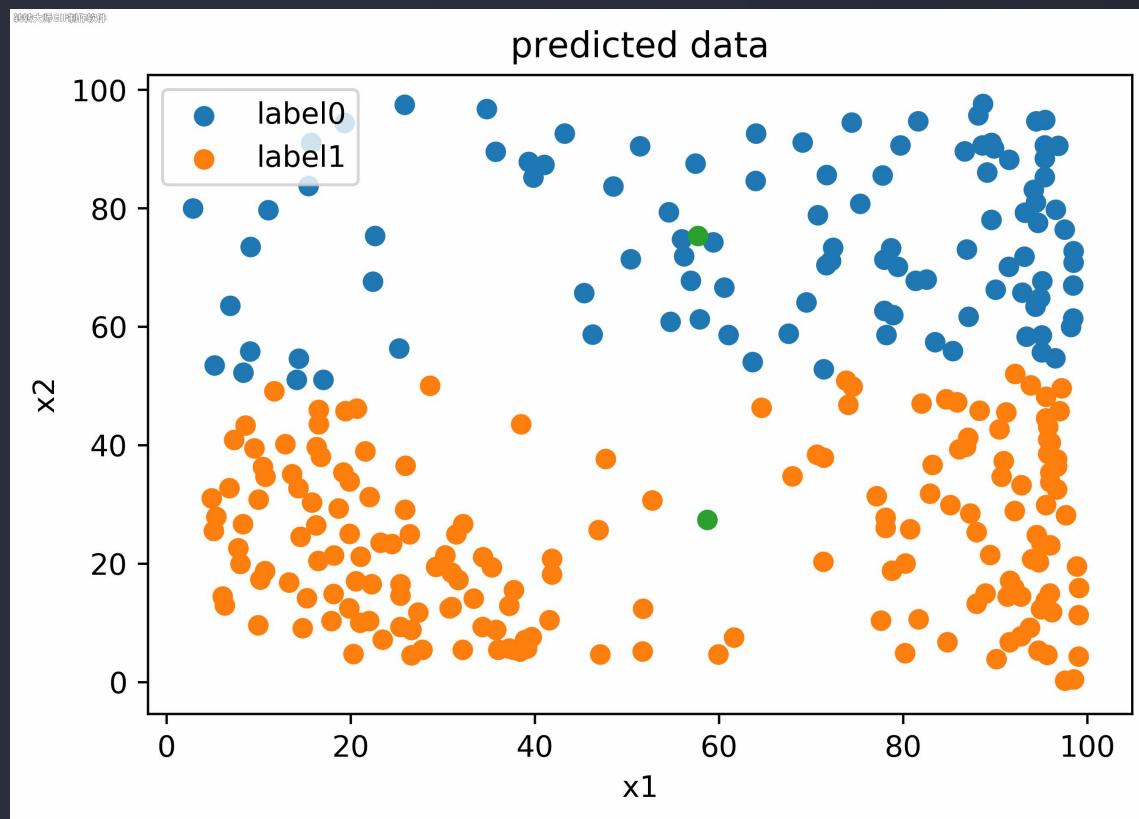
```
[[1 1 1 ... 1 1 1]
 [1 1 2 ... 2 1 1]
 [1 2 0 ... 0 2 1]
 ...
 [1 2 0 ... 0 2 1]
 [1 1 2 ... 2 2 1]
 [1 1 1 ... 1 1 1]]]
```

(140, 140)

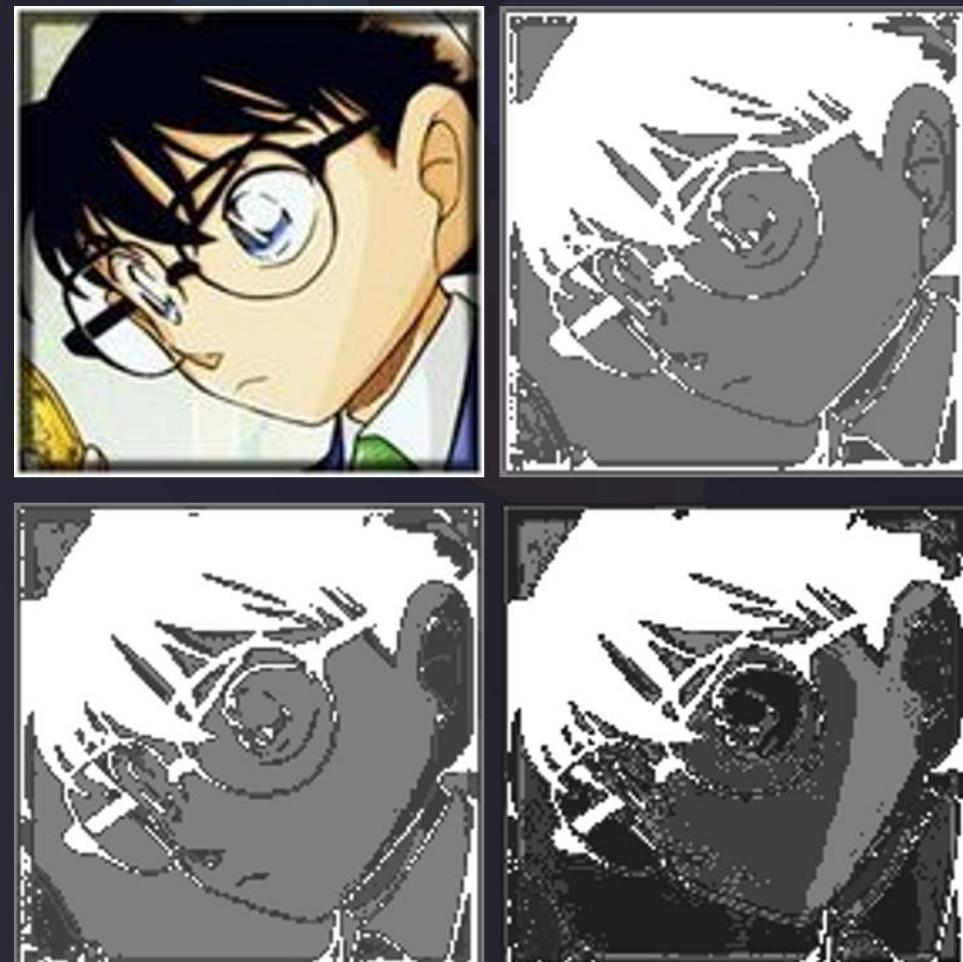
```
[[1]
 [1]
 [1]
 ...
 [1]
 [1]
 [1]]]
```

(19600, 1)

无监督学习与聚类



KMeans实现数据聚类



KMeans实现图像分割

现实问题思考：监督真的重要吗

现实场景：

- 1、任务复杂；
- 2、采集大量数据有难度

解决办法：

- 1、大部分场景都需要监督学习；
- 2、条件允许的情况下尽可能收集足够的样本；
- 3、无法收集足够样本的情况下，考虑标签样本+无标签样本实现监督学习与无监督学习的结合，即半监督学习

| 异常检测

监督式异常检测：提前使用带“正常”与“异常”标签的数据对模型进行训练，机器基于训练好的模型判断新数据是否为异常数据



“正常”

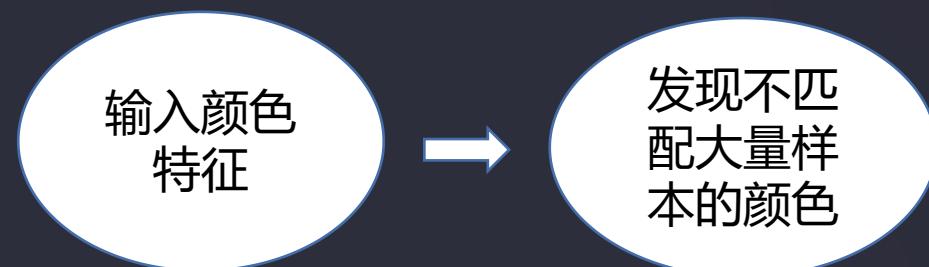
机器

“异常”



| 异常检测

无监督式异常检测：通过寻找与其他数据最不匹配的实例来检测出未标记测试数据的异常



基于高斯分布的概率密度函数

μ 决定中间轴位置， σ 为决定分布集中度

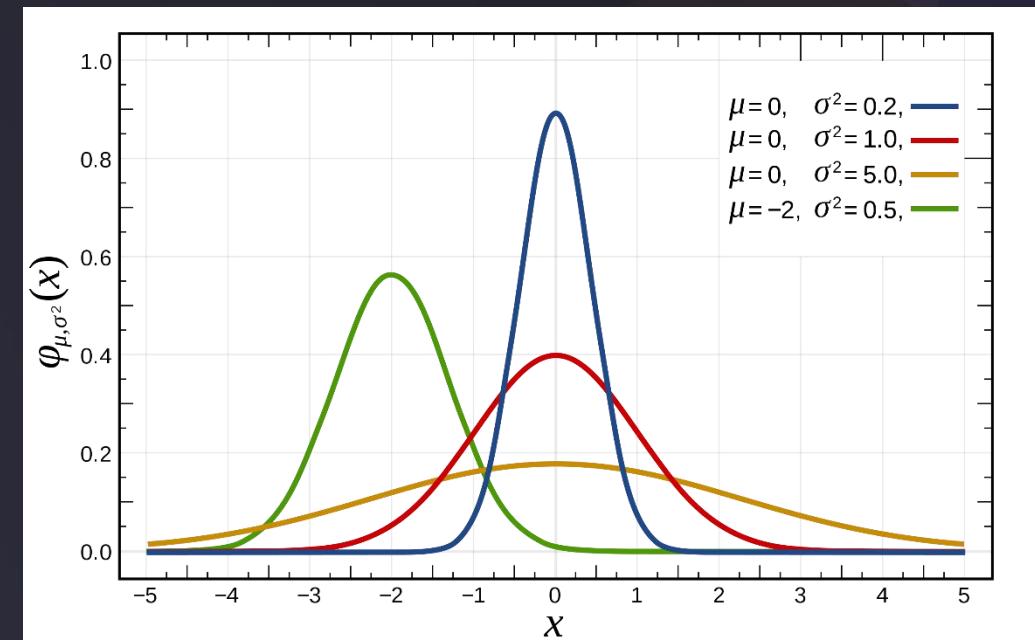
高斯分布（正态分布）的概率密度函数是：

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

其中， μ 为数据均值， σ 为标准差

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)},$$

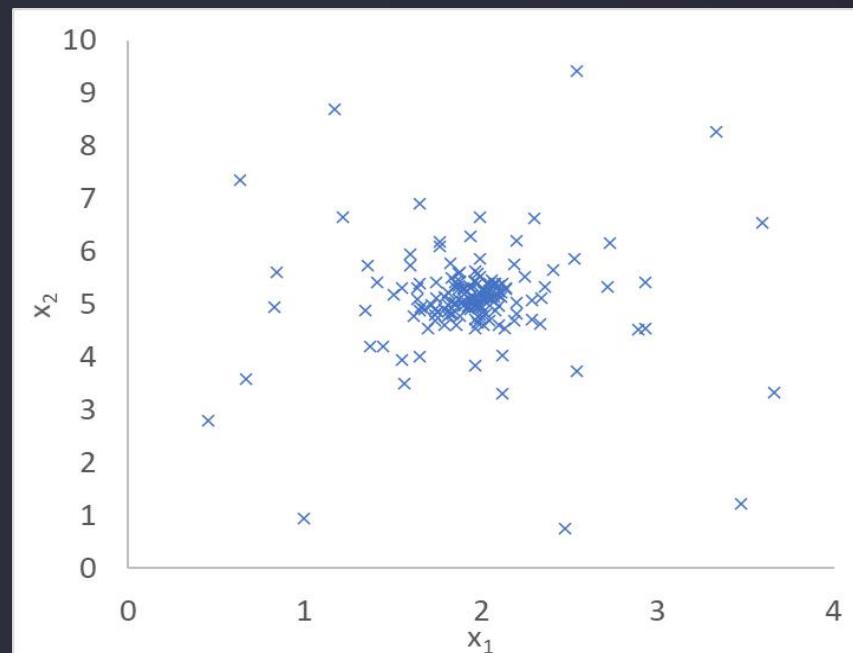
$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$$



高斯分布概率密度图

| 基于高斯分布概率密度函数实现异常检测

数据分布



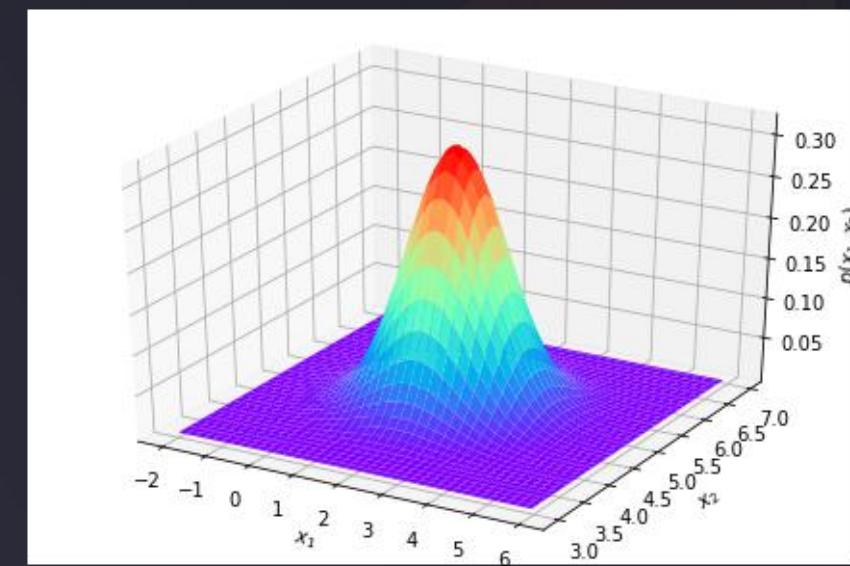
$$\mu_1 = 2, \sigma_1 = 1$$

$$\mu_2 = 5, \sigma_1 = 0.5$$

$$\epsilon = 0.01, x^{(1)} = 1, x^{(2)} = 3$$

$$p(x) = \prod_{j=1}^n p(x^{(j)}; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(x^{(j)} - \mu_j)^2}{2\sigma_j^2}}$$

概率密度函数

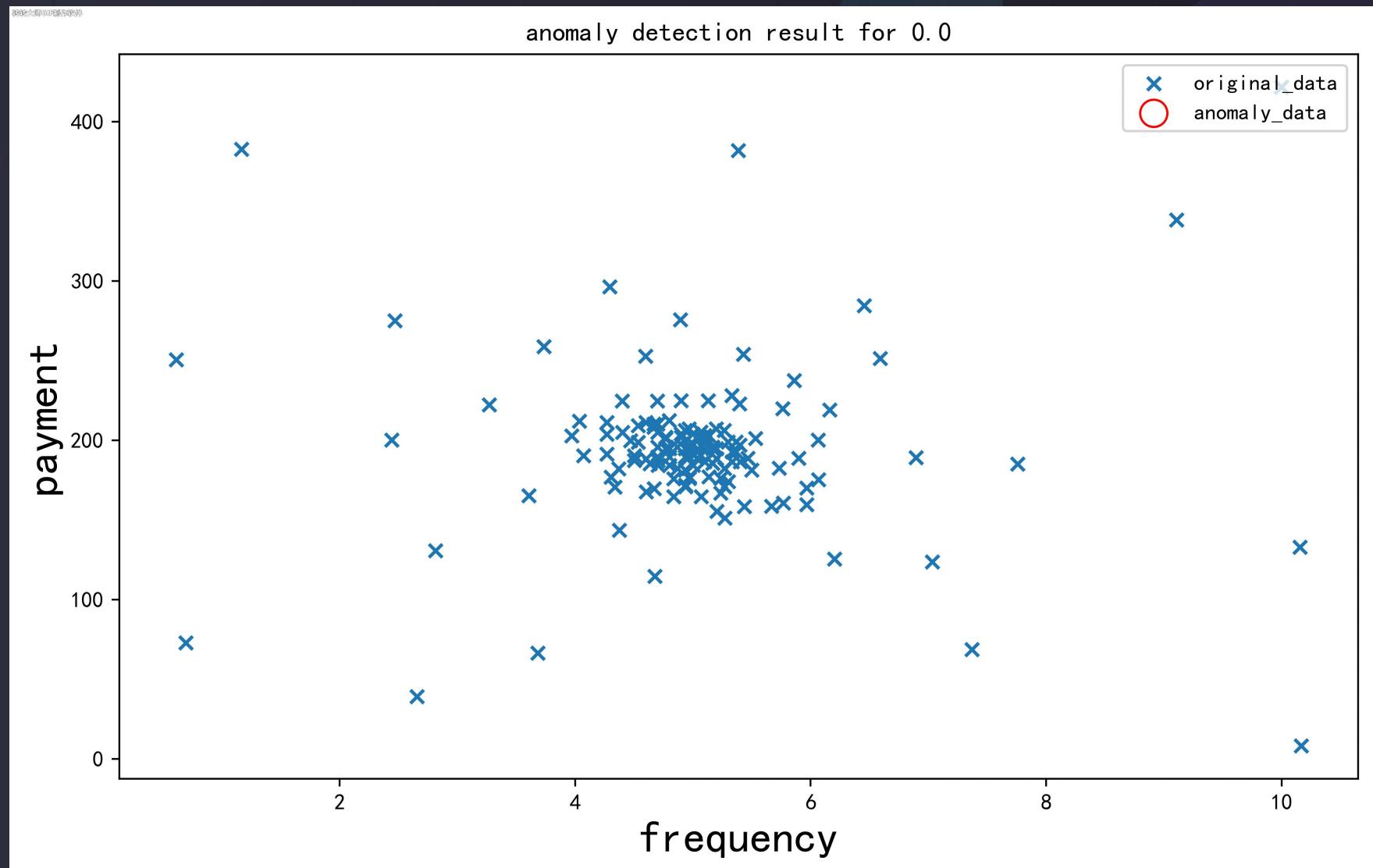


$$p = 0.000065 < \epsilon :$$

异常点!

| 基于高斯分布概率密度函数实现异常检测

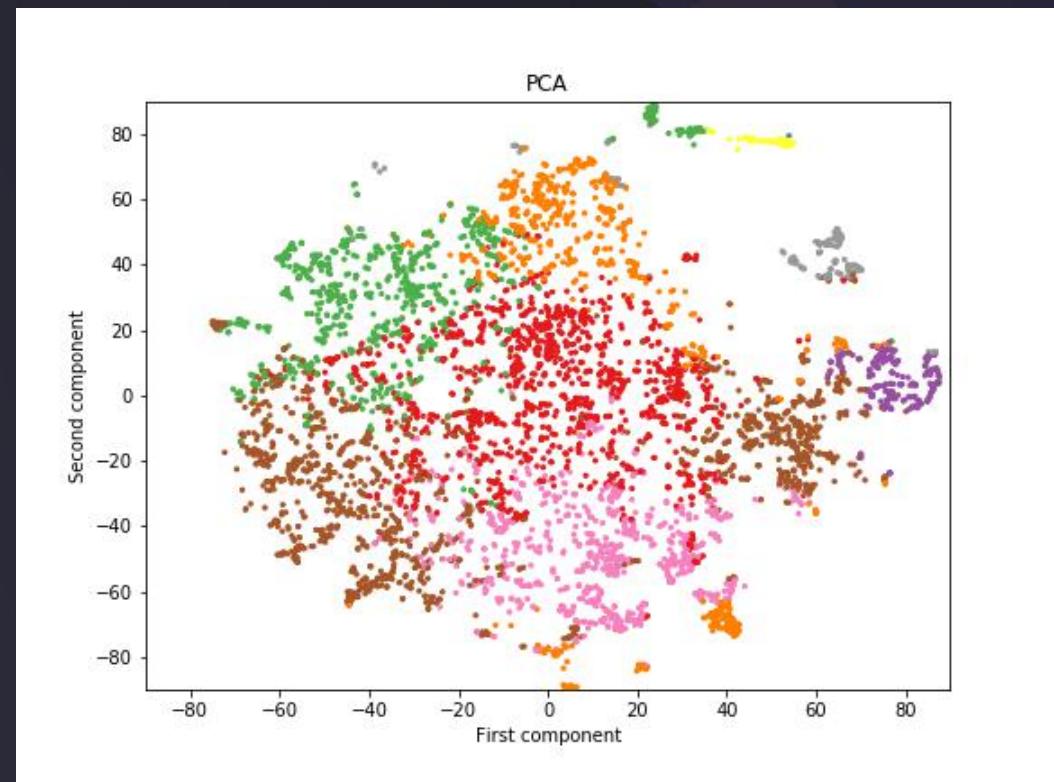
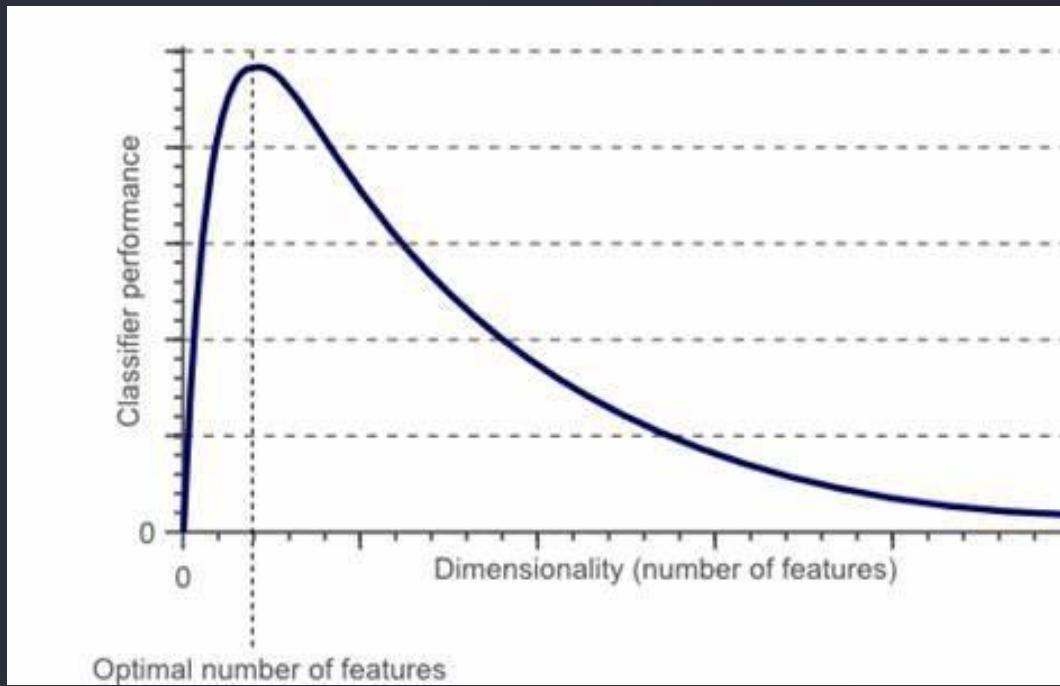
商业异常消费行
为预测



为什么需要数据降维

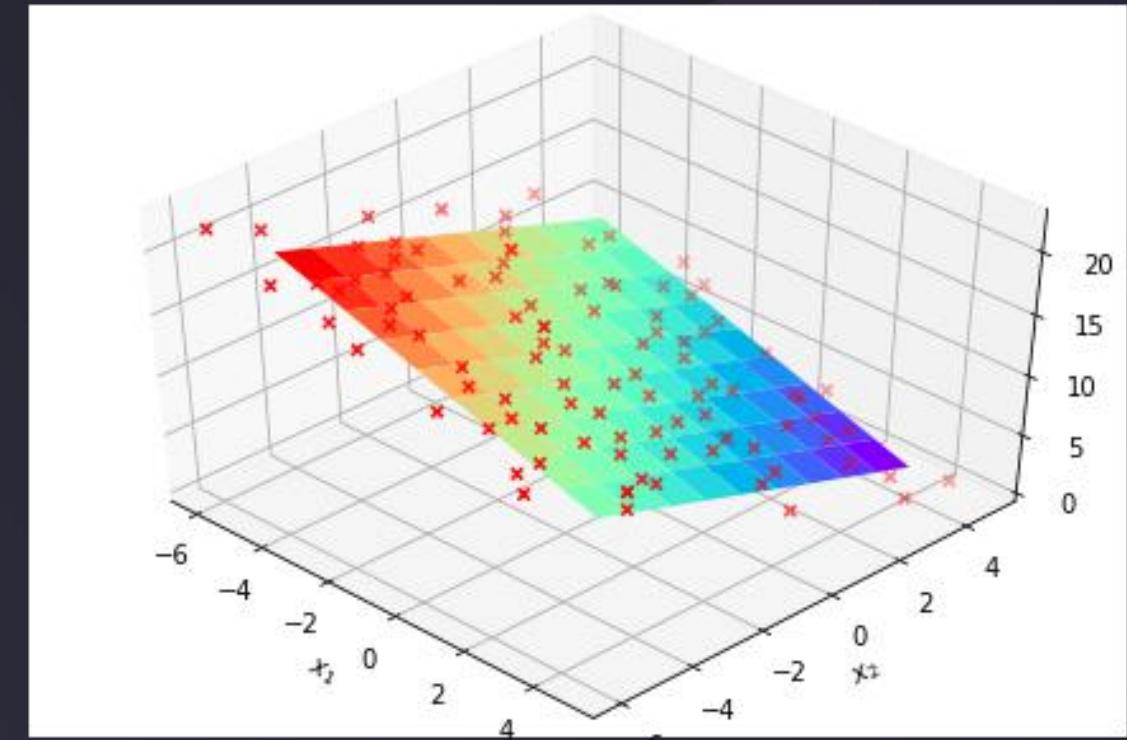
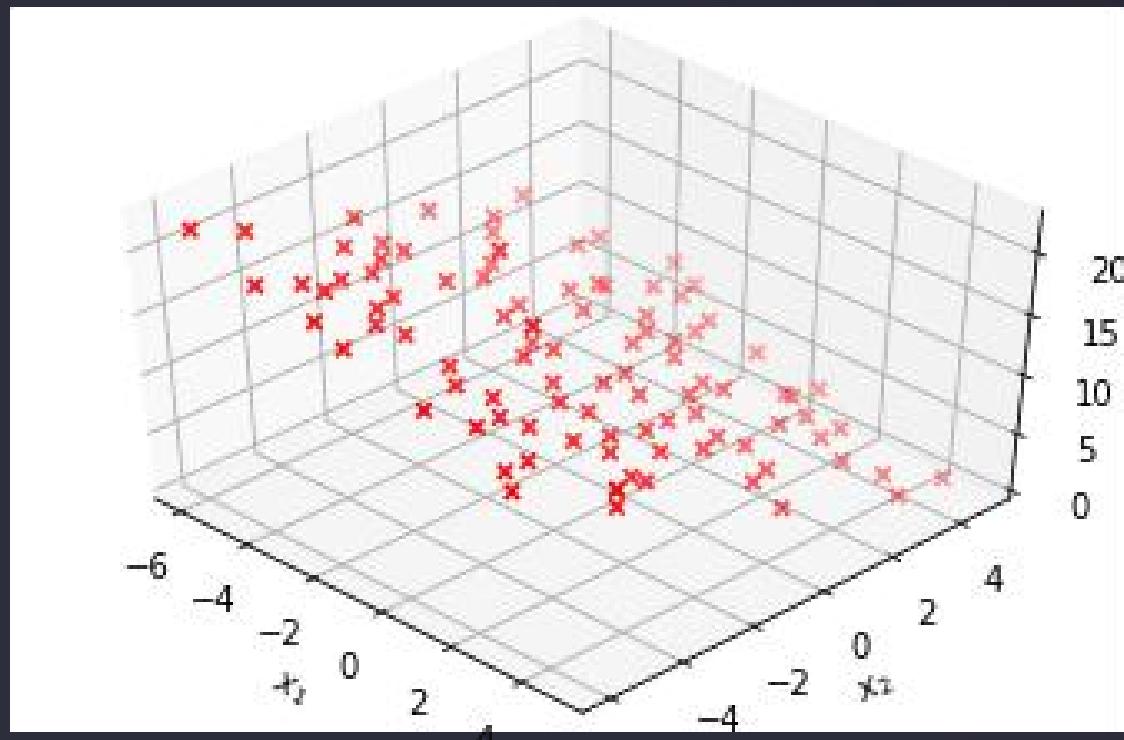
■ Curse of dimensionality - 维数灾难

数据可视化



| 数据降维最常用的方法：主成分分析（PCA）

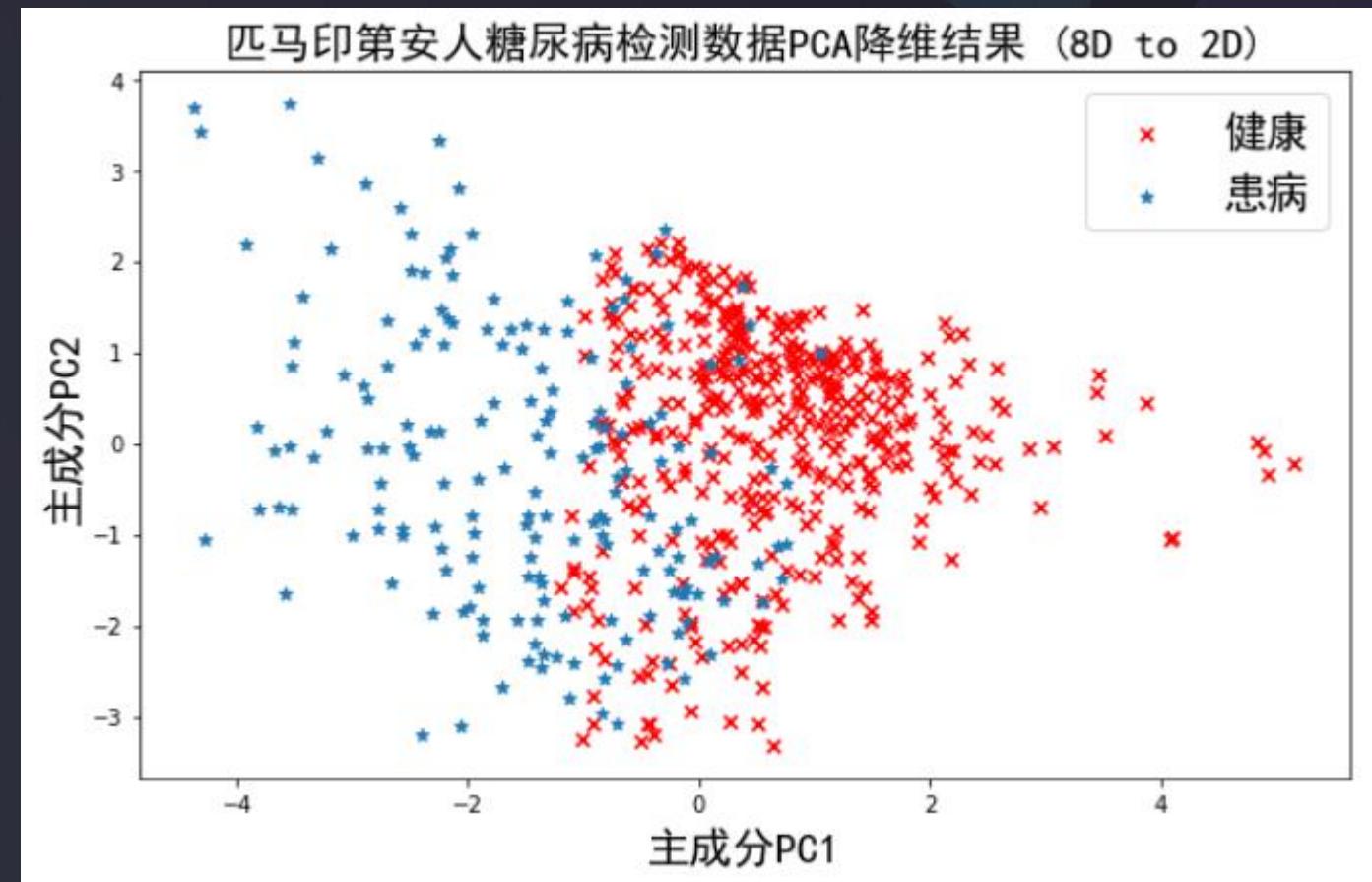
- 也称主分量分析，按照一定规则把数据变换到一个新的坐标系统中，使得任何数据投影后尽可能可以分开（新数据尽可能不相关、分布方差最大化）。



n维数据PCA降维到k维：投影到 u_1 、 u_2 ... u_k 形成的空间

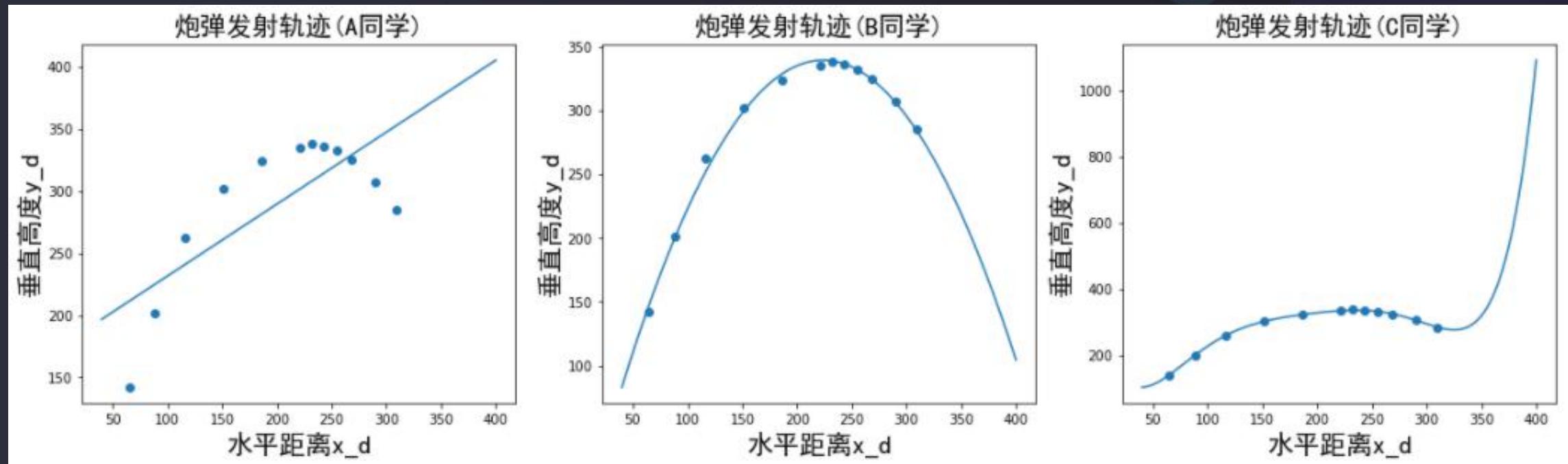
| 数据降维最常用的方法：主成分分析（PCA）

PCA数据降维+
逻辑回归实现
匹马印第安人
糖尿病检测



现实问题思考

炮弹轨迹预测



欠拟合结果

训练数据、预测数据效果都不好

好模型结果

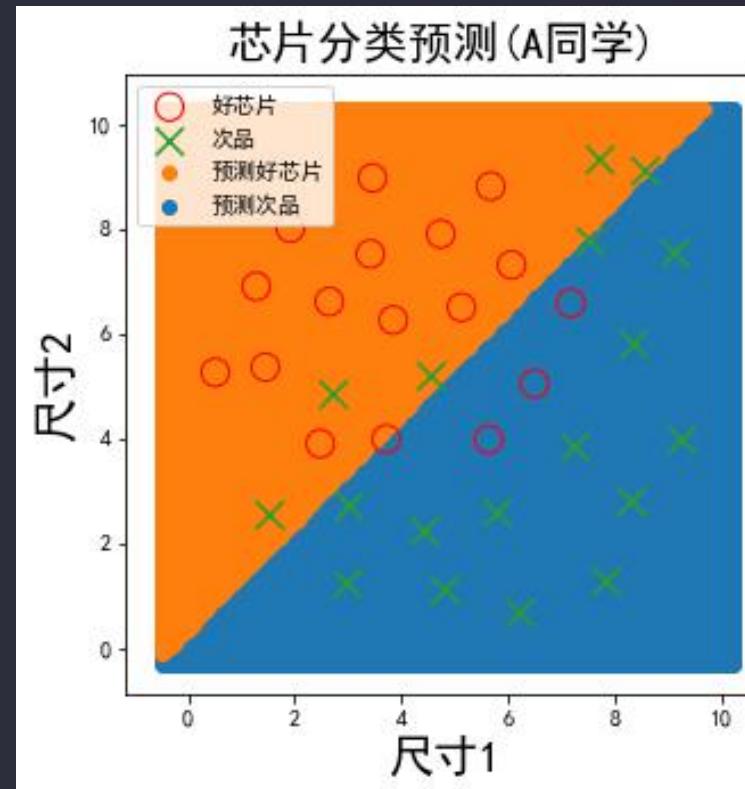
训练数据、预测数据效果都很不错

过拟合结果

训练数据效果很好、但预测数据效果不好

现实问题思考

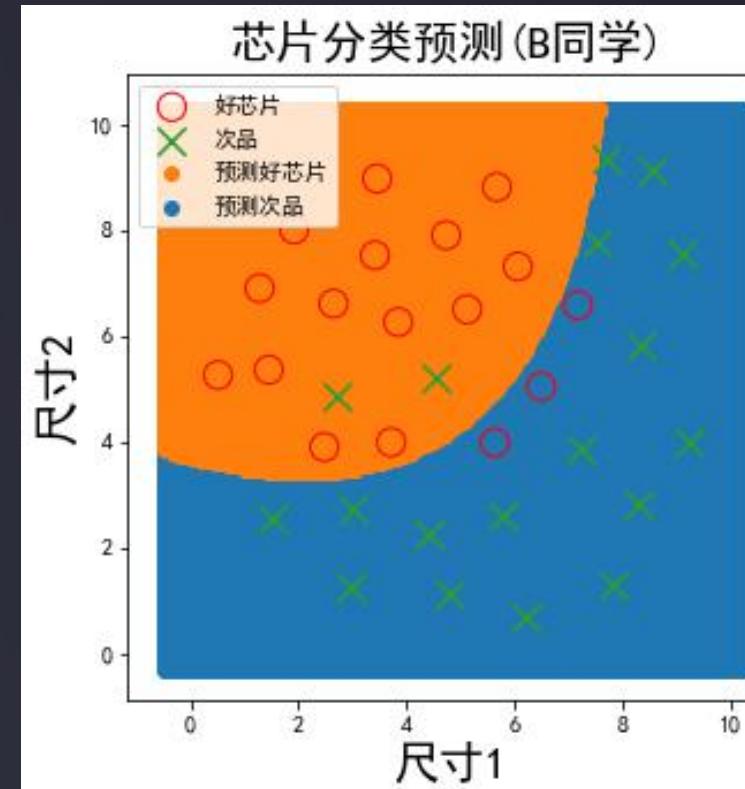
准确率: 77.1%



欠拟合结果

训练数据、预测数据效
果都不好

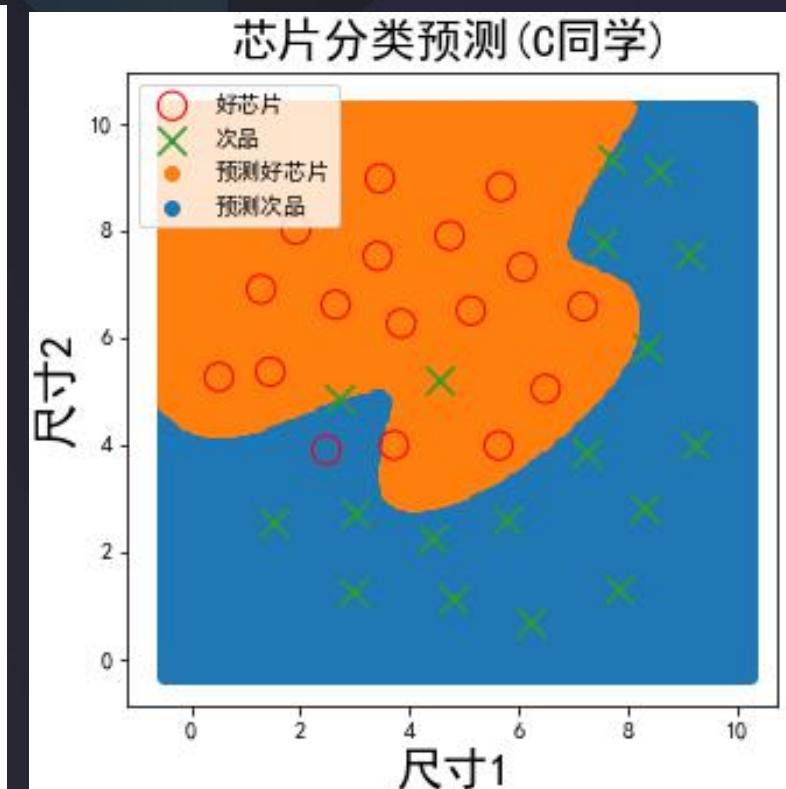
准确率: 85.7%



好模型结果

训练数据、预测数据效
果都很不错

准确率: 94.3%



过拟合结果

训练数据效果很好、
但预测数据效果不好

欠拟合与过拟合

由于模型不合适，致使其无法对数据进行准确的预测。

模型对数据的预测情况

	训练数据	预测数据
欠拟合	不准确	不准确
过拟合	准确	不准确
合适模型	准确	准确

通常来说，欠拟合可通过观察训练数据的预测结果发现



选用其他模型、增加模型复杂度、增加数据样本、采集新的维度数据

| 解决过拟合问题的方法

原因：

- 使用了过于复杂的模型结构（比如高阶决策边界）
- 训练数据不足，有限的训练数据（训练样本只有总体样本中的小部分、不具备代表性）
- 样本里的噪音数据干扰过大，模型学习到了噪音信息（使用过多与结果不相关属性数据）

解决办法：

- 简化模型结构（降低模型复杂度，能达到好的效果情况下尽可能选择简单的模型）
- 数据增强（按照一定的规则扩充样本数据）
- 数据预处理，保留主成分信息（数据PCA处理）
- 增加正则化项（regularization）

回顾模型训练与评估流程

数据载入 → 数据可视化与预处理 → 模型创建 → 全数据用于模型训练 → 模型评估



现实问题思考：只用准确率作为模型评估指标的局限性

案例：奢侈品公司在投放广告前，根据部分高档消费客户的数据作为训练集和测试集，训练测试了高档消费客户的分类模型。该模型的准确率达到了95%。但是在实际广告投时，发现模型输出预测都为普非高档消费客户（非目标用户群体），其结果无法帮助决策。



高档消费客户只占所有消费客户的一小部分（5%），
模型预测所有用户都为非高档消费用户，准确率就
高达95%了，其实并没有找到目标用户群。

混淆矩阵

混淆矩阵也称误差矩阵，用于统计各类别样本预测正确与错误数量，能更全面地评估模型表现。

		Predicted	Predicted
		0	1
Actual	0	TN	FP
	1	FN	TP

- **True Positives (TP):** 预测准确、预测为正样本的数量（实际为1，预测为1）
- **True Negatives (TN):** 预测准确、预测为负样本的数量（实际为0，预测为0）
- **False Positives (FP):** 预测错误、预测为正样本的数量（实际为0，预测为1）
- **False Negatives (FN):** 预测错误、预测为负样本的数量（实际为1，预测为0）

(预测结果正确或错误, 预测结果为正样本或负样本)

混淆矩阵小结

优点:

- 分类任务中，相比单一的准确率指标，混淆矩阵提供了**更全面的模型评估信息** (TP\TN\FP\FN)
- 基于混淆矩阵，我们可以计算出**多样的模型表现衡量指标**，从而实现模型的综合评估

应用场景决定了衡量指标的重要性:

- **广告精准投放**(正样本为“目标用户”): 希望目标用户尽可能都被找出来、即实际正样本预测正确，需要关注召回率；同时，希望预测的正样本中实际都尽可能为正样本，需要关注**精确率**
- **异常消费检测** (正样本为“异常消费”): 希望判断为正常的消费（负样本）中尽可能不存在异常消费，还需要关注**特异度**

| 数据质量决定模型表现的上限！



上游决定下游，建模前**五检查**：

- 1、样本代表性：采集数据的方法是否合理，采集到的数据是否有代表性
- 2、标签统一化：对于样本结果，要确保每个样本都遵循一样的标签规则
- 3、数据合理性：样本中的异常数据点是否合理、如何处理
- 4、数据重要性：数据属性的意义，是否为无关数据
- 5、属性差异性：不同属性数据的数量级差异性如何

| 数据质量决定模型表现的上限！



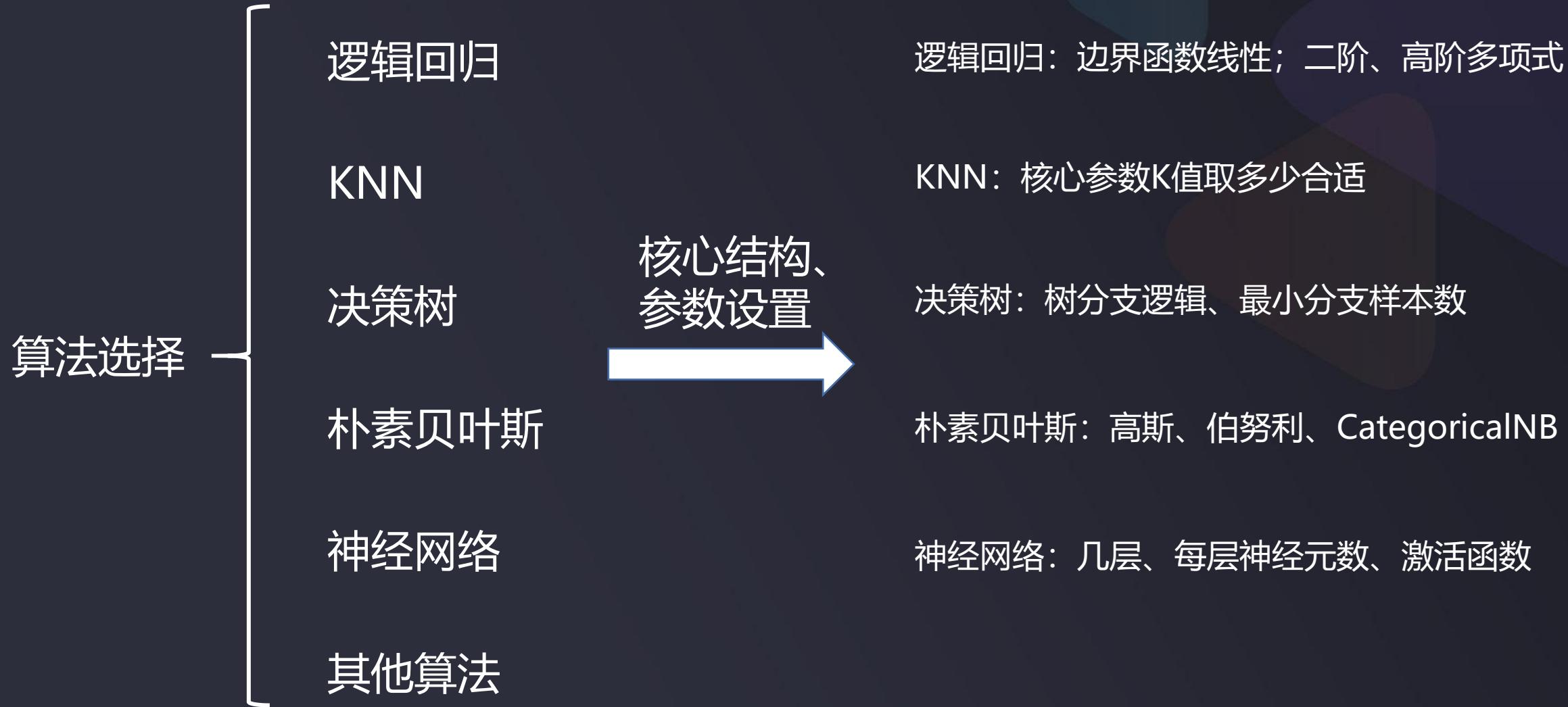
尝试以下方法：

- 1、根据实际场景扩充或减少样本
- 2、对不合理标签数据进行预处理
- 3、删除不重要的属性数据、数据降维
- 4、对数据进行归一化或标准化
- 5、过滤掉异常数据

好处：

- 1、数据质量提升，有助于提高模型表现
- 2、帮助模型学习到正确信息（合理的“监督”）
- 3、降低噪声影响、减少过拟合、节约运算时间
- 4、平衡数据影响，加快训练收敛
- 5、降低噪声影响、提高鲁棒性

| 模型建立与优化



|提高模型表现的四要素

数据预处理

- 扩大数据样本
- 增加/减少属性
- 数据降维、标准化
- 异常数据剔除...

模型选择

- 尝试不同的模型
- 通过不同指标、基于训练/测试数据评估表现

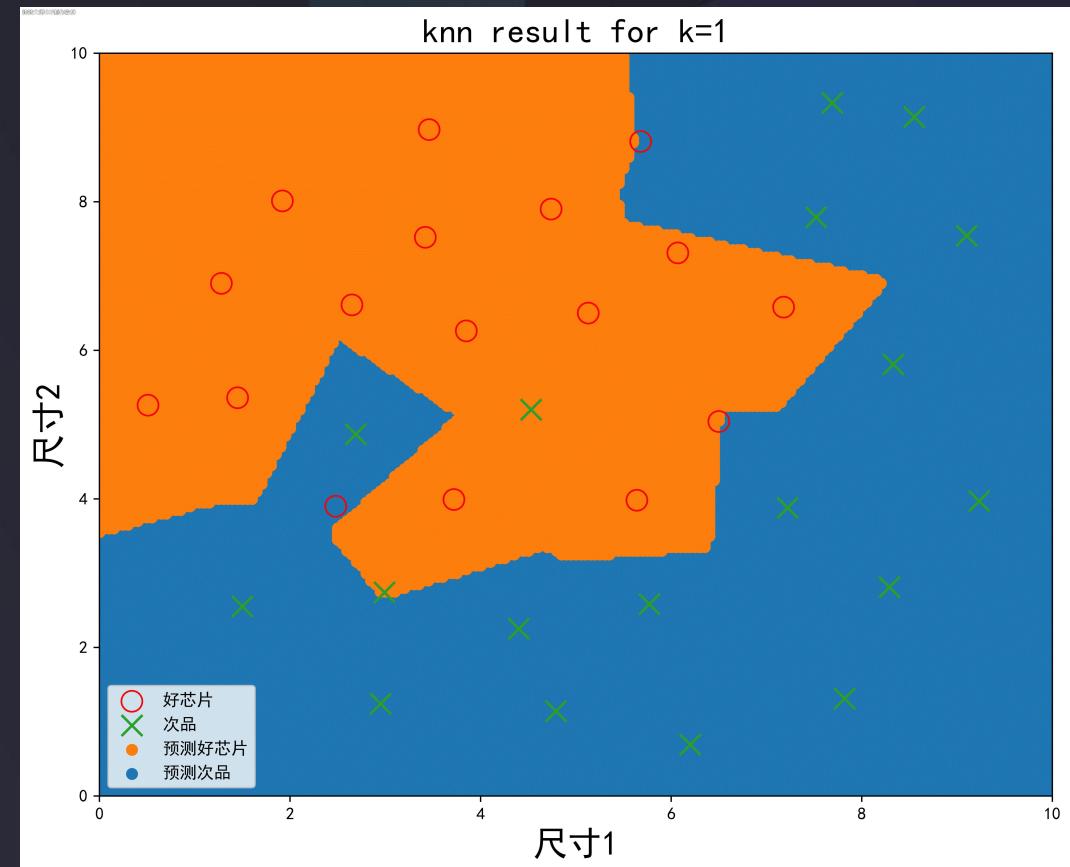
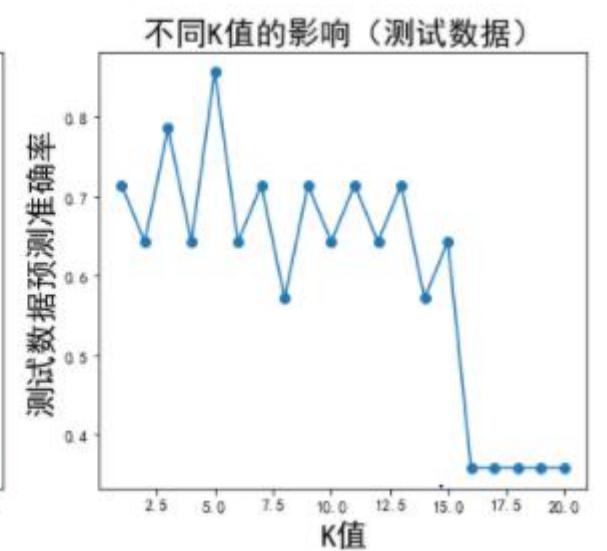
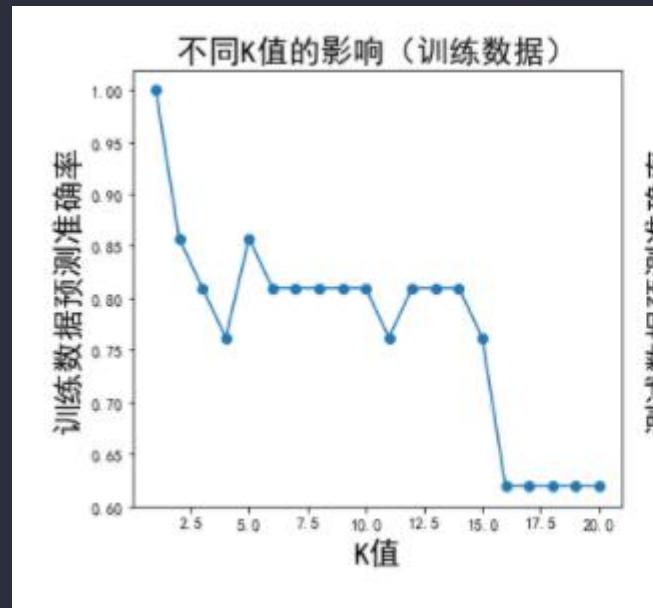
结构参数优化

- 不同的结构、求解方法
- 核心参数修改

其他方法

- 增加正则项
- 多模型结合.....

芯片品质预测



异常数据检测、PCA分析、数据分离、混淆矩阵、核心参数选择



C S D N 学院 IT 实战派

专属资料

