

# Practice of AI

## C2: Machine learning & Data analyze

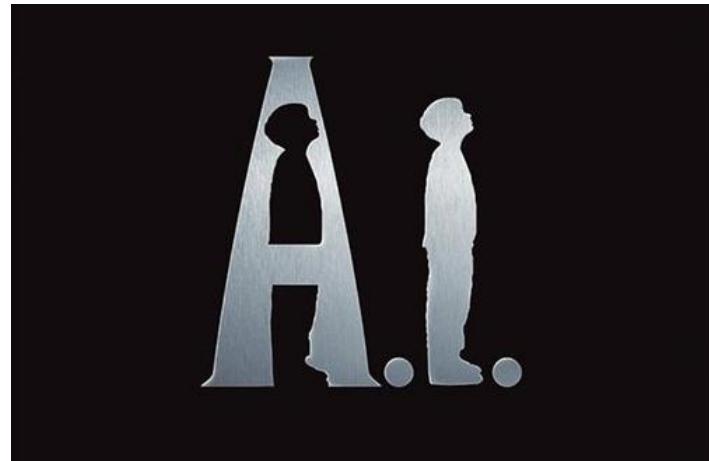
Jim Xie

2020/3/6



# Goal

---



了解ML流程，进行数据探索分析和特征选取

训练一个BMI分级模型

# 试验样本

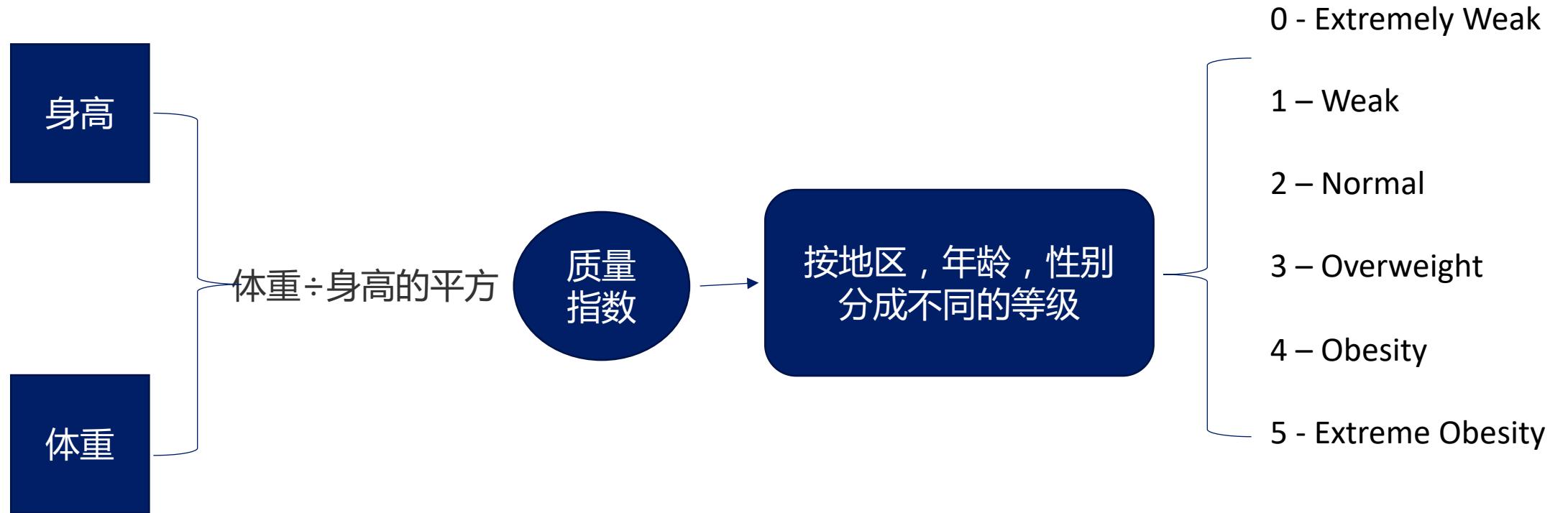
	Gender	Height	Weight	Level
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
5	Male	189	104	3



- 性别
- 身高
- 体重
- 健康等级

{ 0 - Extremely Weak  
1 – Weak  
2 – Normal  
3 – Overweight  
4 – Obesity  
5 - Extreme Obesity

# 健康等级



[http://10.206.67.123:8888/edit/dataset/500\\_Person\\_Gender\\_Height\\_Weight\\_Index.csv](http://10.206.67.123:8888/edit/dataset/500_Person_Gender_Height_Weight_Index.csv)

# BMI

---

Body Mass Index 分类识别  
( 简化版 )

## Demo

<http://localhost:8888/notebooks/C2/BMI-draft.ipynb>

# 感受

---

1. 这么简单？感觉结果稀里糊涂就出来了？
2. 靠谱吗？敢在product环境使用吗？
3. 样本里只有身高，体重数据，如果无效怎么办？

[识别无效的例子](#)

# 简化流程

---



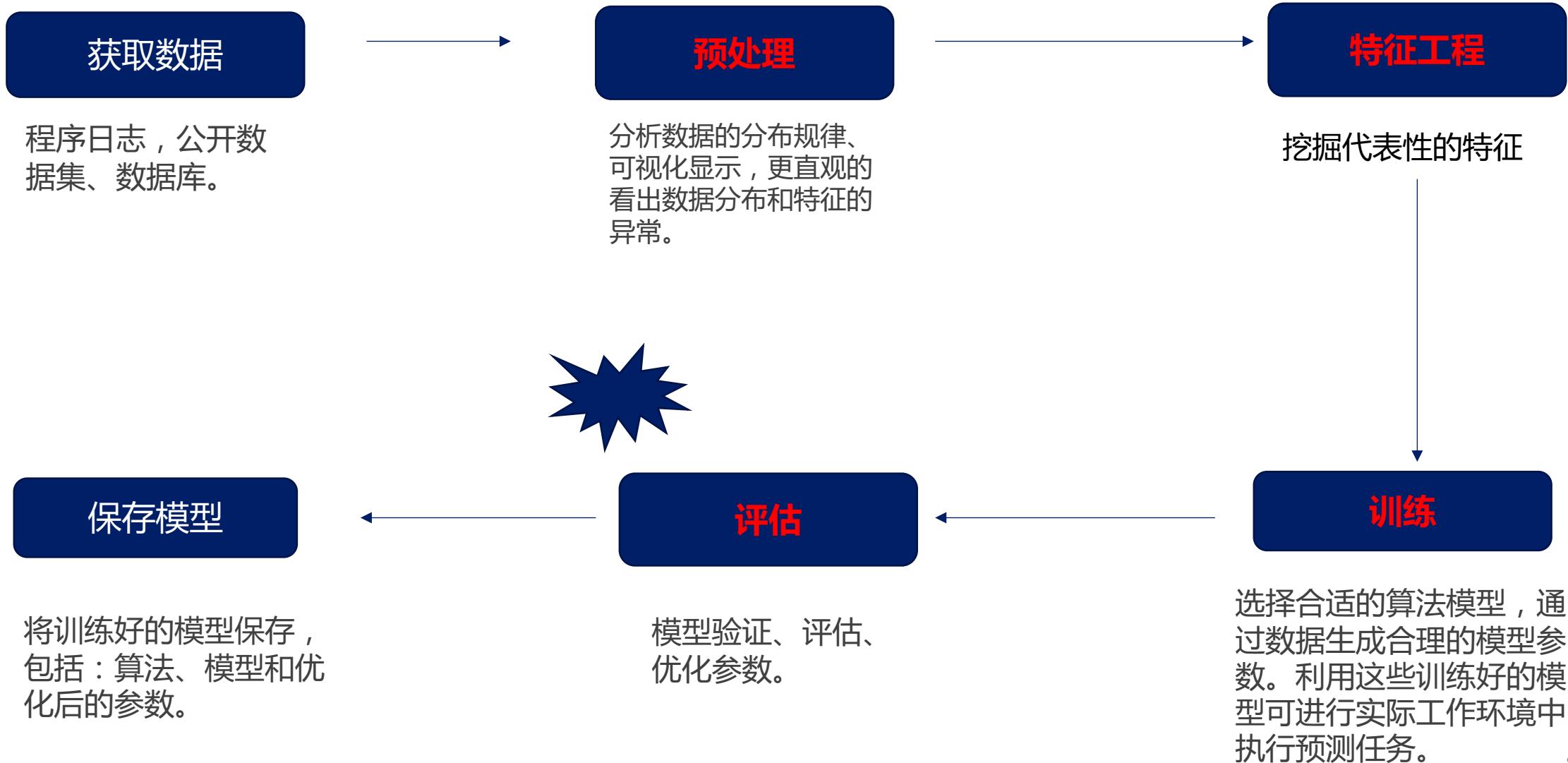
获取数据

数据预处理

模型训练

模型评估

# 细化流程



# 模型评估

---

**举例：**

- 现在要在机场装个摄像头，对恐怖分子进行识别；
- 怎样建一个模型，达到90%+的精准度？

# 基本概念

---

- 样本，特征，数据集分别是什么？

# 结构数据的特征和样本

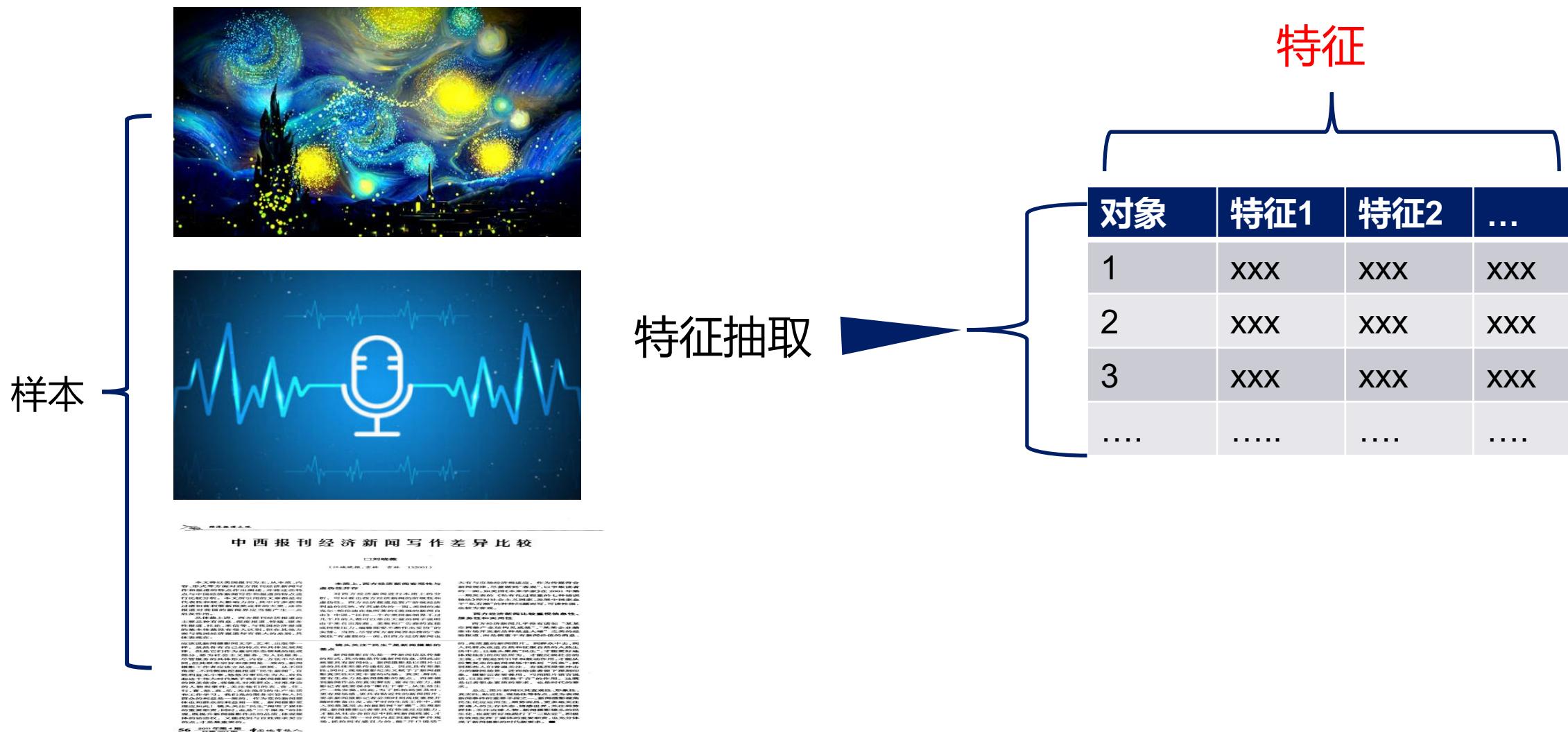
**特征** (输出特征，输入特征)

	Gender	Height	Weight	Level
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
5	Male	189	104	3

**样本**  
(训练样本  
测试样本)

标注标签

# 非结构数据的特征和样本



# 数据集划分

## 数据集: 样本的集合

类型	类比	作用	经验比例 ( 小样本 )	经验比例 ( 百万级大样本 )
训练集	学生课本	训练模型	60% 或 70%	98%
验证集	作业	寻找最佳参数	20% 或 0%	1%
测试集	考试	评估模型性能 ( 不会改变学习算法或参数 )	20% 或 30%	1%

1. 训练集上训练 → 2. 验证集上评估参数 → 3. 测试集上测试结果

# 数据预处理

---

- 样本扫描
- 样本异常值/缺失值处理
- 数据变换
- 数据可视化

# 样本扫描

---

拿到样本后，先进行整体扫描，对数据产生一个宏观的印象。

- 读取样本
- 样本大小
- 数据类型
- 缺失值或空值
- 有哪些值
- 统计信息
- .....

常用函数示例：<http://127.0.0.1:8888/notebooks/C2/数据集.ipynb>

# 缺失值/异常值处理

	Gender	Height	Weight
0	Male	174	96
1	Male		87
2	Female	185	110
3	Female	1197	104
4	Male	149	61

替换成编码  
( 0,1 )

删除 or 插值

```
import pandas as pd
df1 = pd.read_csv('./data/500_Person_Gender_Height_Weight_Index.csv')
df1['Height'].fillna(df1.Height.mean())
#df1['Height'].fillna(df1.Height.mode())
#df1['Height'].fillna(df1.Height.median())
df1.dropna()
df1.dropna(axis=1)
df1.replace("Male", 0,inplace=True)
df1.replace("Female", 1,inplace=True)
```

插值方法	样本中的值	pandas函数
平均数	169.00	mean()
中位数	170.00	median()
众数	168.00	mode()
回归数	通过模型预测得到	

# 可视化

---

## 可视化的作用和常用方法

# 可视化常用方法

---

可视化 = 作图

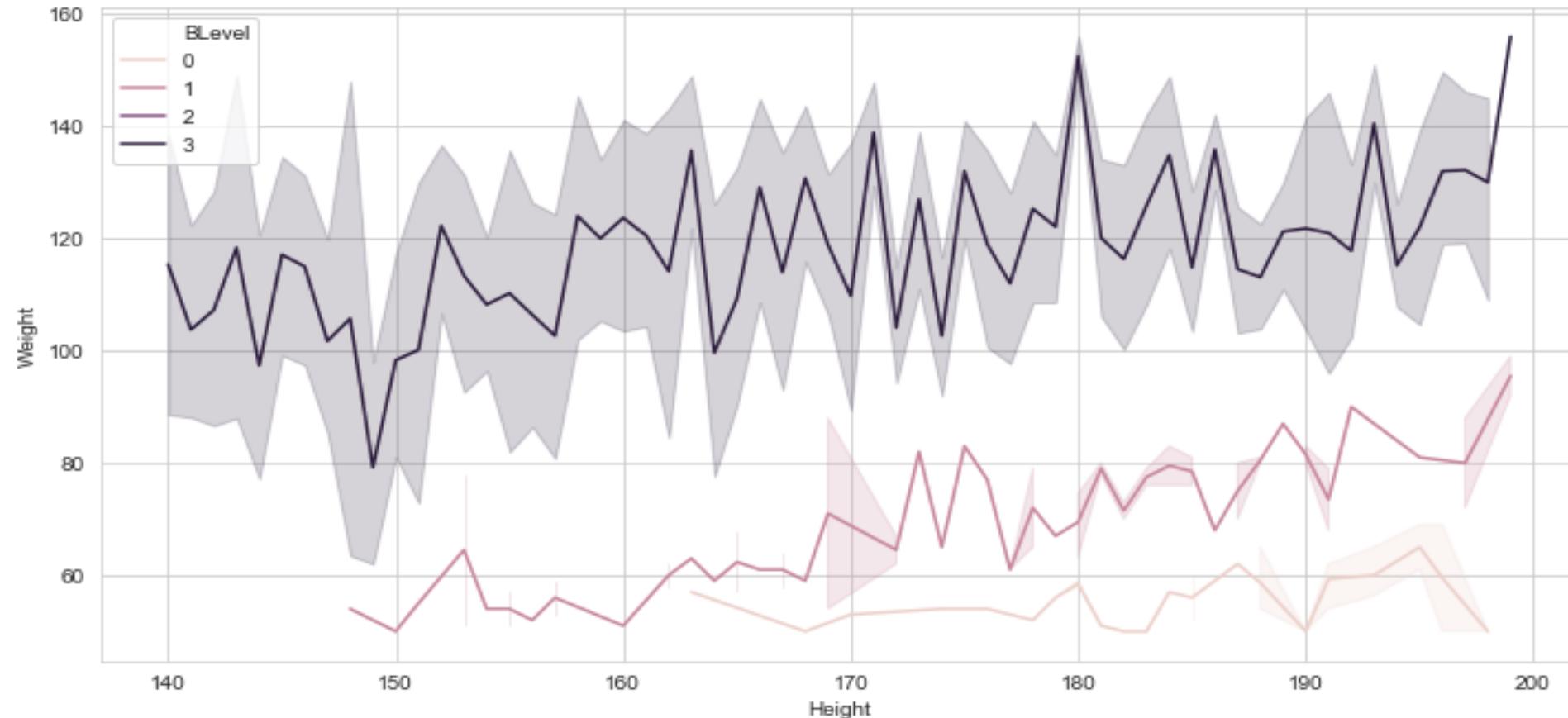


数据转换

- 趋势图
- 散点图
- 直方图
- 热力图
- .....
- 归一化
- 标准化
- .....

# 趋势图 : Good case

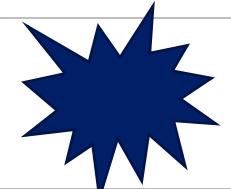
观察走势：试验样本中，身高和体重是否有关联？



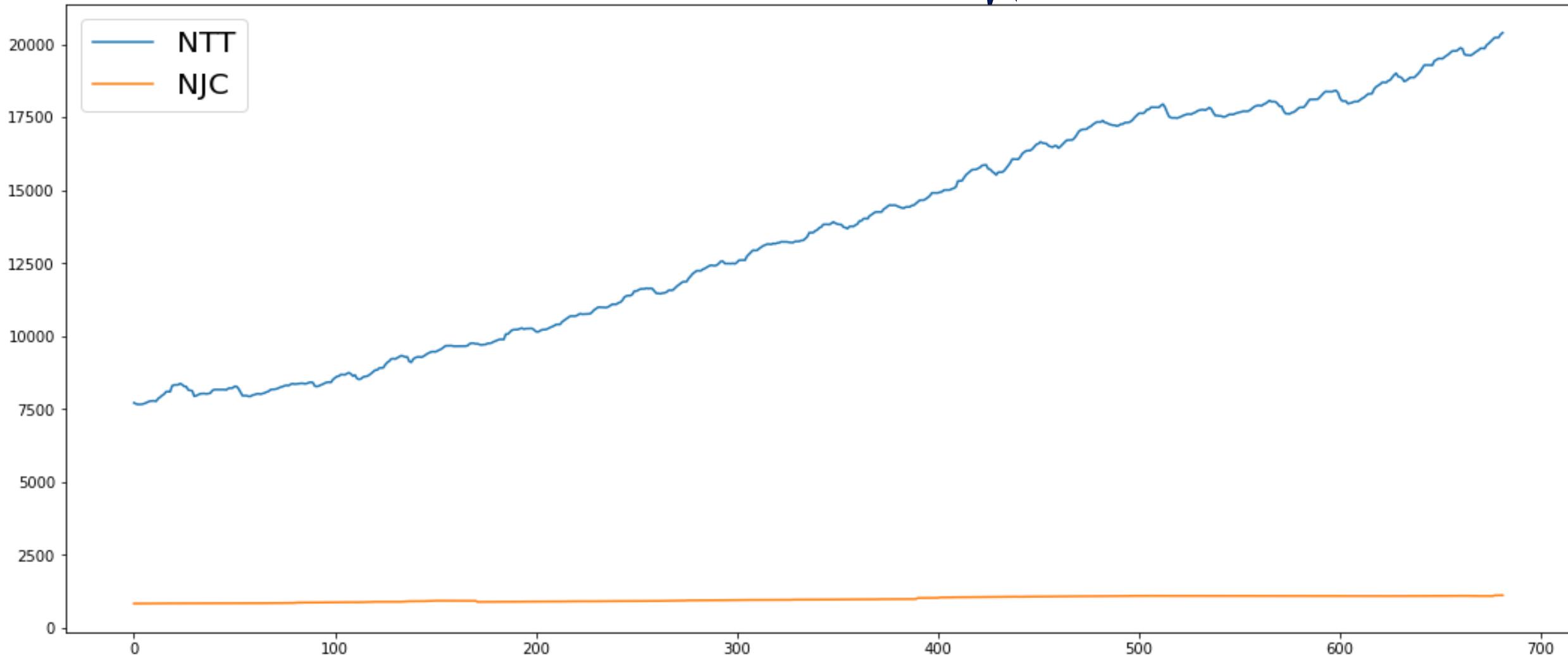
```
plt.figure(figsize=(12,6))
sns.lineplot(x="Height",y="Weight",hue='Level',ci=95,data=df1)
```

# 趋势图 : Bad case

观察走势: 产品销售增长是否有规律?



数据差别大，看不清楚



# 数据变换：归一化

---

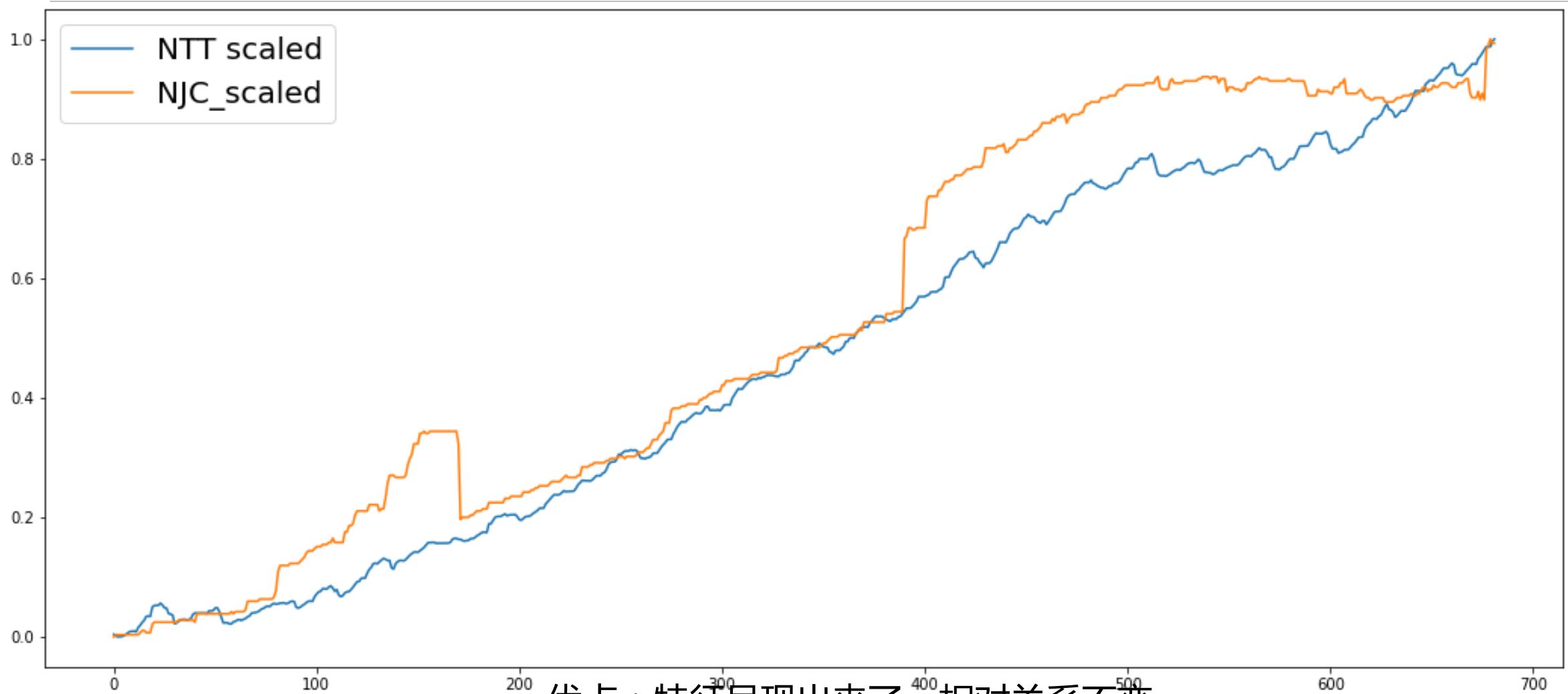
产生原因： 单位不同导致 (如白细胞个数和身高数据 )

解决方法： 将样本中的数据压缩至0 ~ 1之间。 (去量纲)

最值归一化 :  $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$  ( x为当前值 )

Method	Data					
Raw data	[ 529    578    466    437    318 ]					
MaxMinScal	[ 0.81    1.    0.569    0.45769    0. ]					

# 归一化后

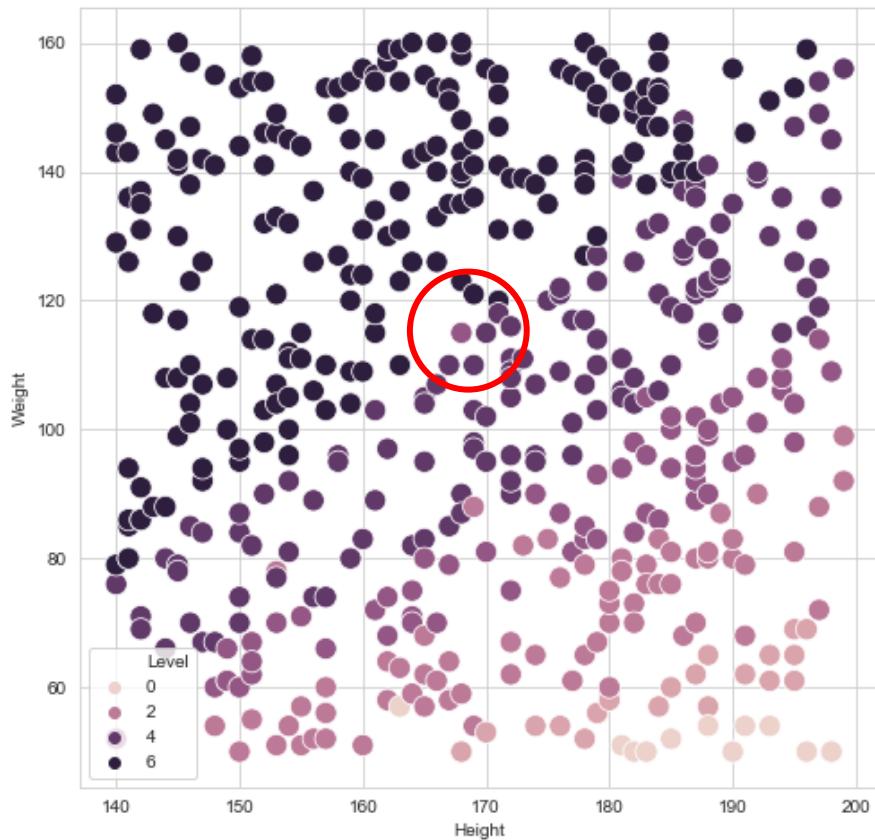


优点：特征呈现出来了，相对关系不变

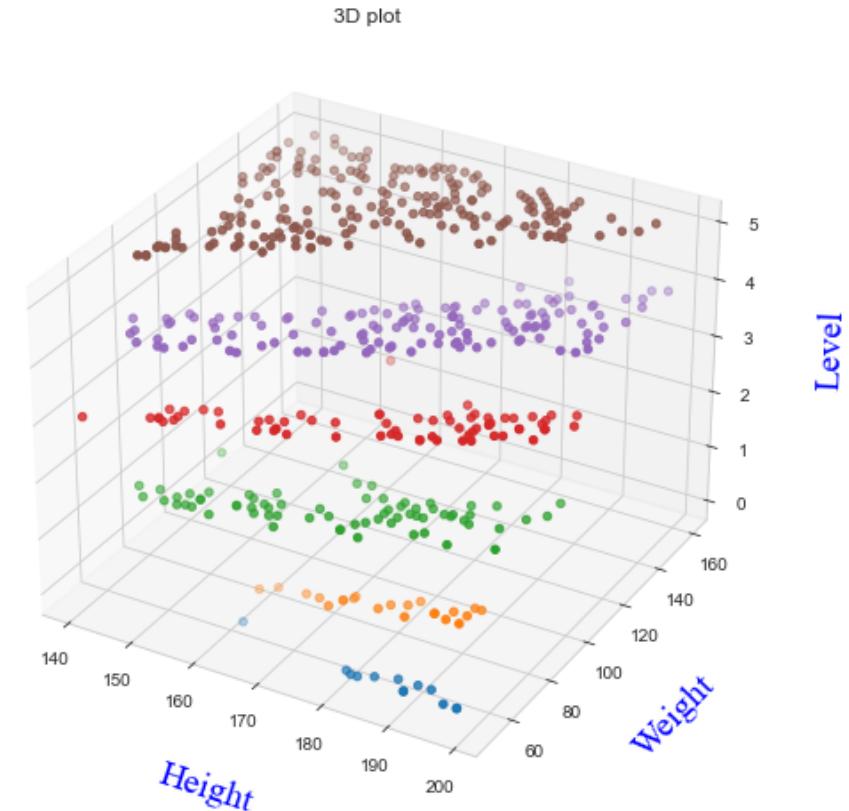
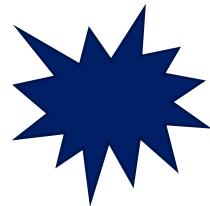
缺点：需要预先知道最大/最小值，异常值不敏感

# 散点图: Bad case

观察数据的分布：肥胖程度，身高，体重的分布



- 异常数据不明显
- 类别多，辨识度不够



```
ax = sns.scatterplot(x='Height', y="Weight", s=150, hue="Level", data=df3)
```

# 数据变换：标准化

---

作用1：去量纲

作用2：异常值检测

作用3：更容易显现特征

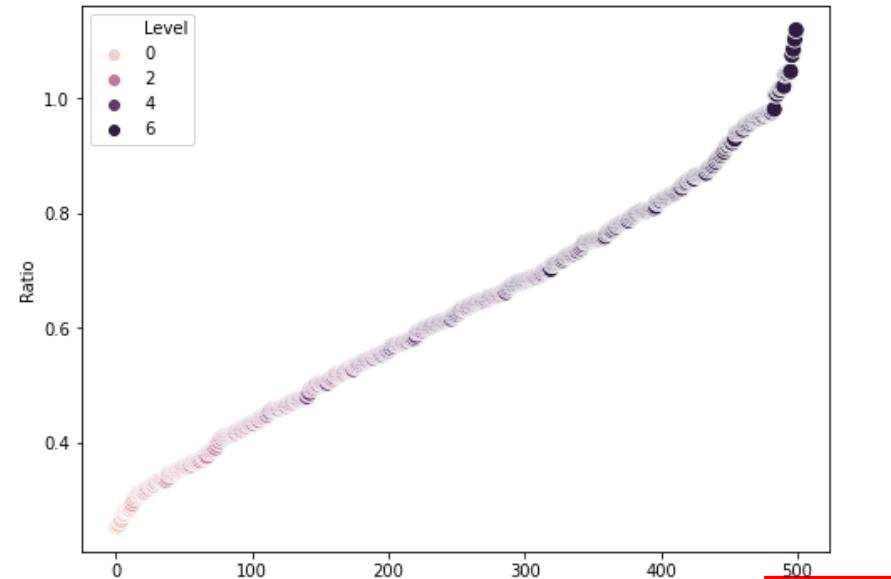
变换方法： $x' = \frac{x - \mu}{\sigma}$  <http://10.206.67.123:8888/notebooks/JimXie/excise/excise-1.ipynb>

Method	Data					
Raw data	[ 529        578        466        437        318 ]					
Z-score	[ 0.71550817    1.26850345    0.00451425    -0.32276867    -1.6657572 ]					

# 散点图: Good case

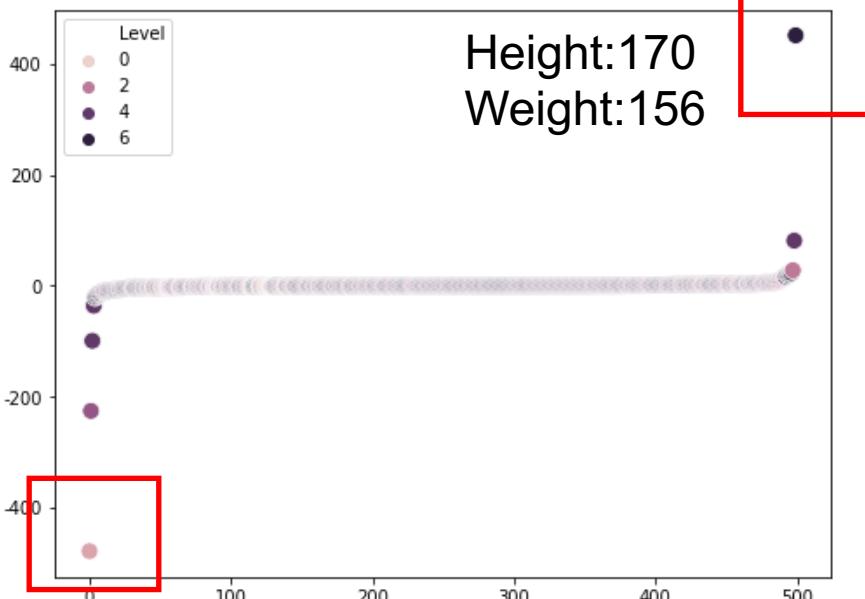
	Gender	Height	Weight	Ratio
490	Male	140	143	1.02
491	Male	143	149	1.04
492	Female	140	146	1.04
493	Male	140	146	1.04
494	Female	151	158	1.05
495	Male	148	155	1.05

身高 , 体重比  
( 原始 )



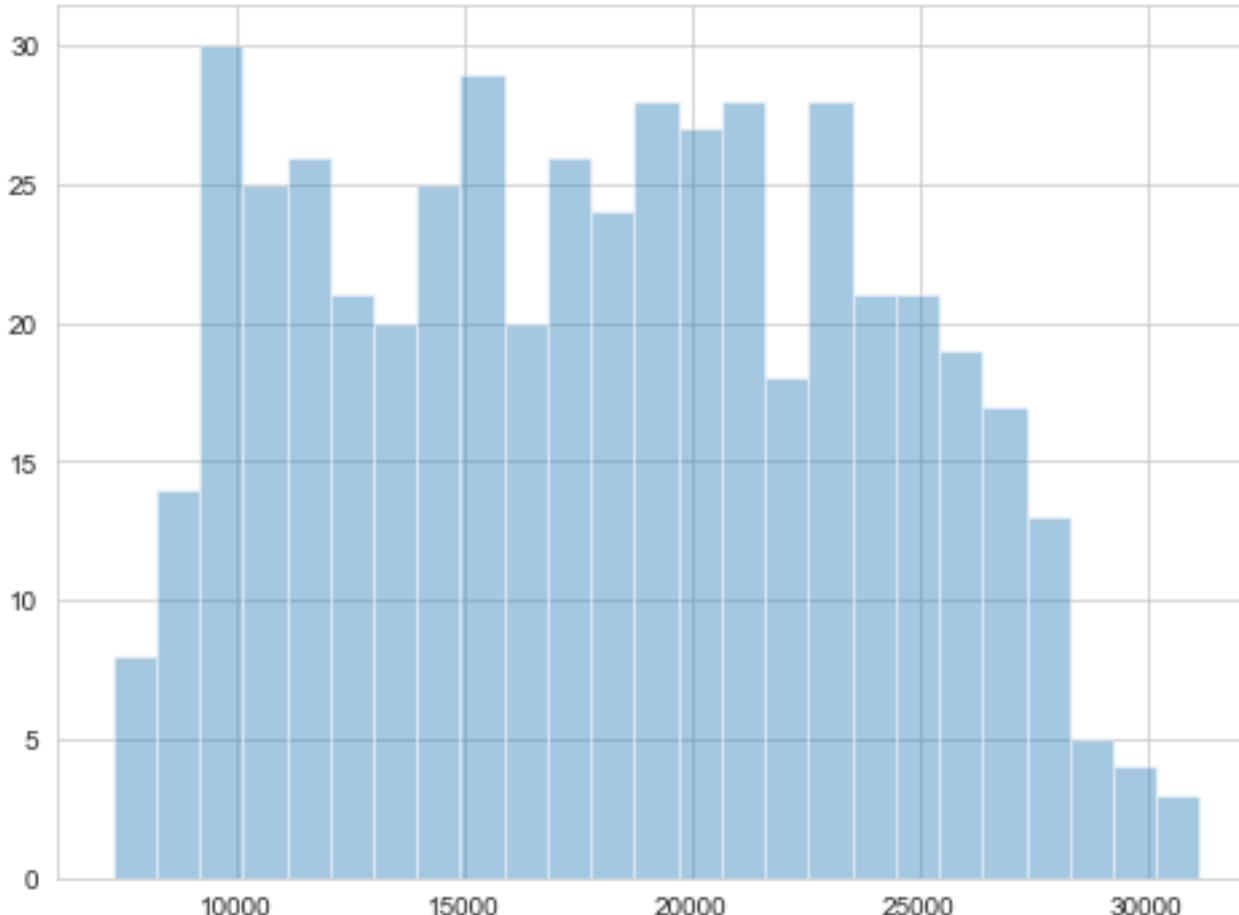
	Gender	Height_scaled	Weight_scaled	Height	Weight	Ratio
490	Female	-0.12	-1.45	168	59	12.23
491	Male	-0.12	-1.73	168	50	14.57
492	Male	0.06	1.08	171	141	16.76
493	Male	0.06	1.27	171	147	19.63
494	Male	0.06	1.27	171	147	19.63
495	Female	0.06	1.42	171	152	22.03
496	Female	0.06	1.51	171	155	23.46
497	Female	-0.06	-1.61	169	54	27.86
498	Male	0.00	0.28	170	115	81.27
499	Female	0.00	1.54	170	156	451.50

身高 , 体重比  
( 标准化后 )

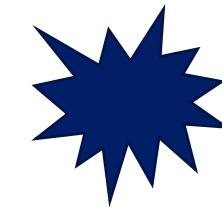


# 直方图 : Bad case

频次统计 : 组合特征 体重\*身高 在不同区间统计



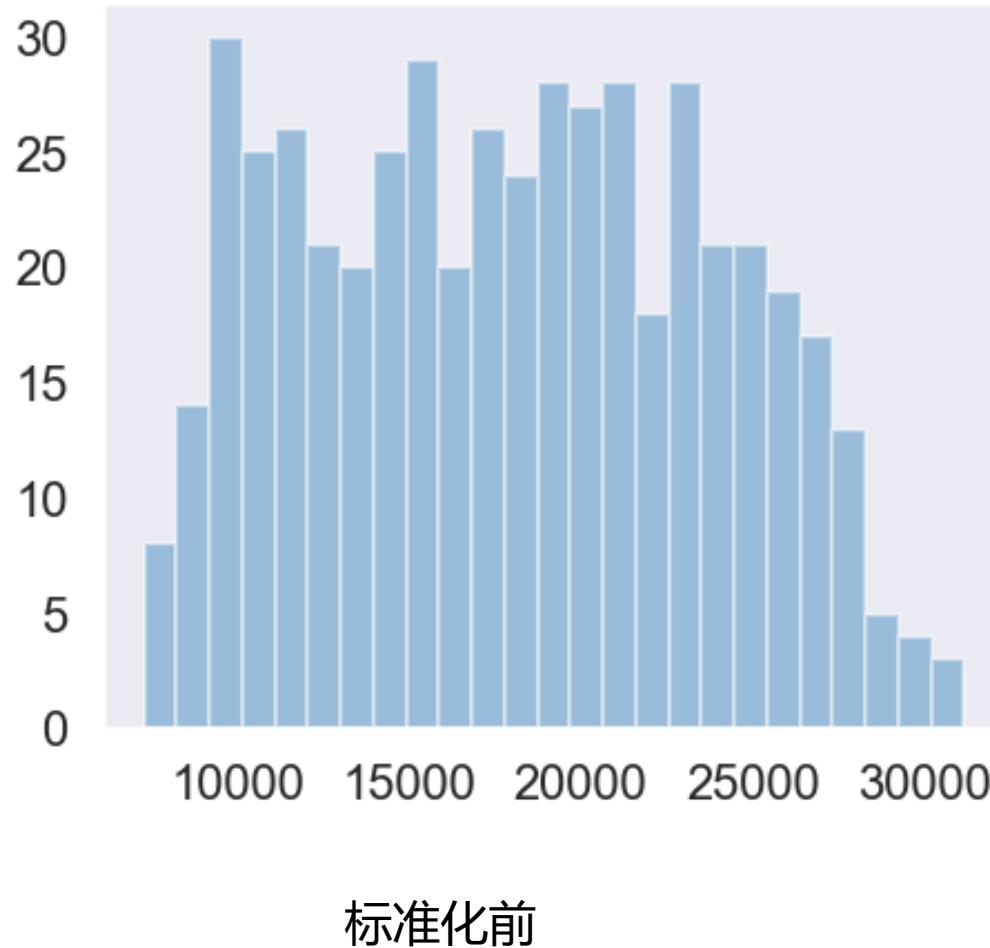
```
sns.distplot(df3['Height']*df3['Weight'], bins=25)
```



没发现什么规律

# 直方图 : Good case

频次统计 : 体重\*身高 在不同区间统计

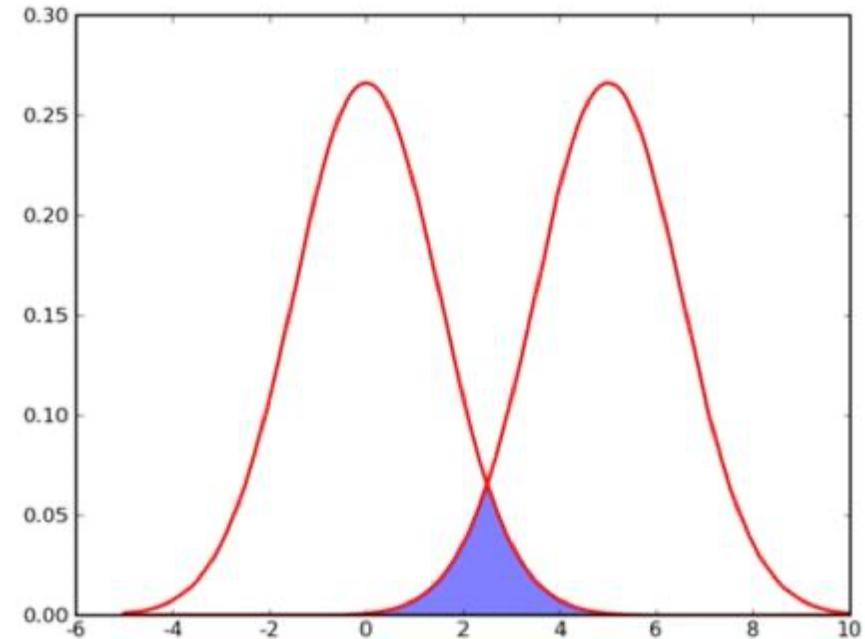


正态分布 ( 特征显现 )

# 正太分布的应用

---

- 样本质量检验
- 发现新特征
- 模型需要
- 有效性检验
- .....

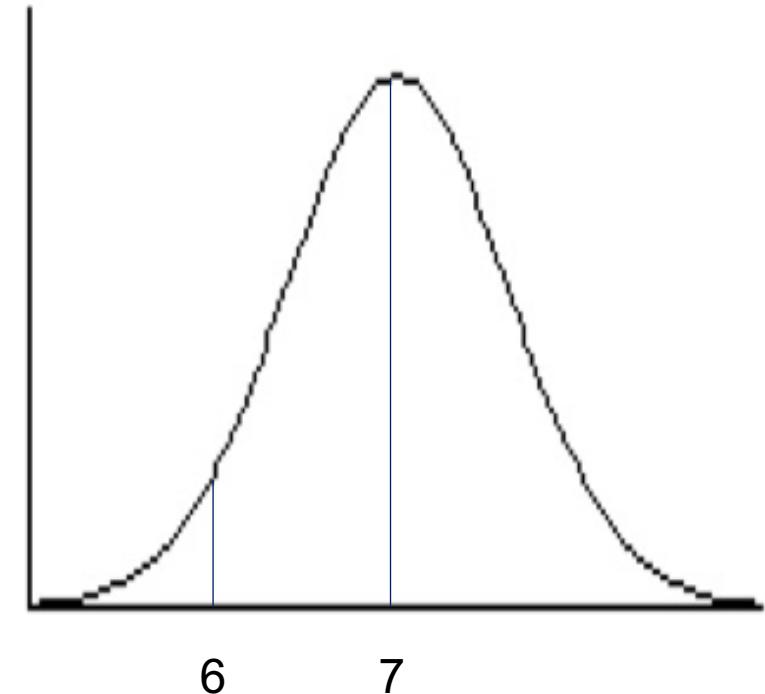


# 正太分布的应用：有效性检验

---

举例：

- 正常人感冒平均7天会自愈；
- 新发明了一种新药，在试验组平均6天恢复；
- 药品是否有效？

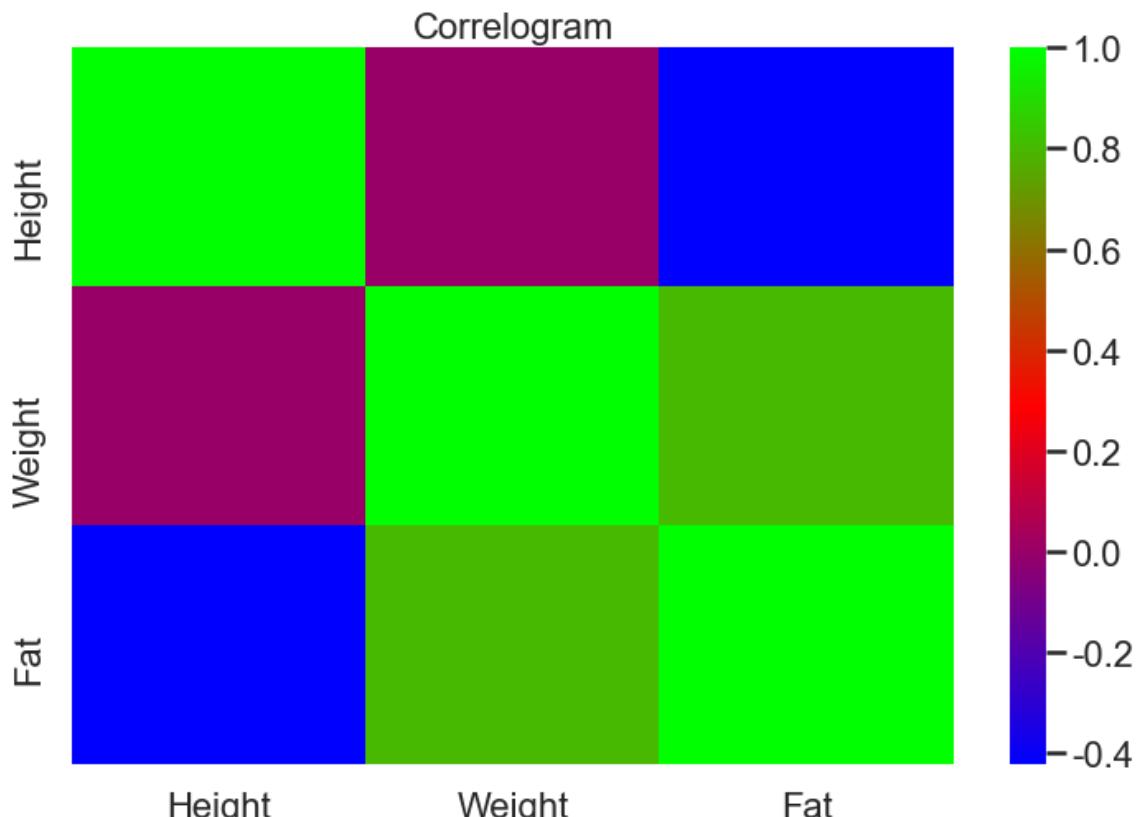


# 热力图

观察多维数据的分布区间

样本的关联系数矩阵

	Height	Weight	Level
Height	1.00	0.00	-0.42
Weight	0.00	1.00	0.80
Level	-0.42	0.80	1.00



```
sns.heatmap(corr,cmap='brg', annot=False)
```

# 小结

---



- 常见的作图方法
- 归一化和标准化？
- 通过标准化发现异常值

# 特征工程

---

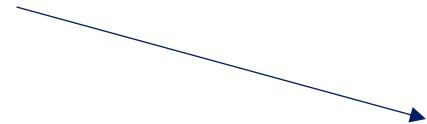
## 特征工程的作用与方法

数据样本有很多基础特征（属性），特征工程就是凝练一些有价值的特征。



# 为什么需降维？

特征过多坏处



计算缓慢  
模型复杂  
增加干扰项  
造成维度灾难



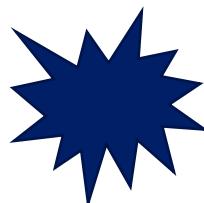
维度灾难指什么？

# 维度灾难：现象

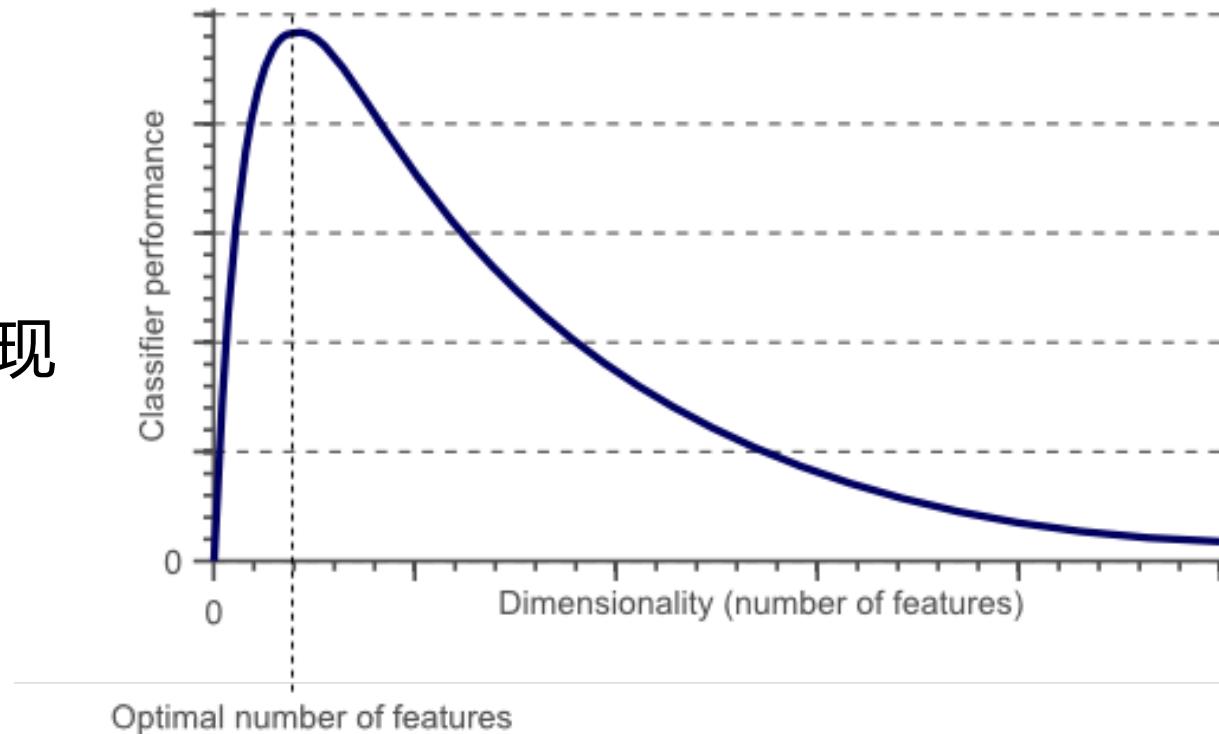
高维情况下，很多模型表现会急剧下降

猫狗识别的例子

模型性能在不同特征数量下的表现



我应该选择在多少个特征呢？



# 维度灾难：背景

人脸比对方法：<http://localhost:8888/notebooks/C3/face/demo.ipynb>

```
[ 6.45104647e-02,  2.84561459e-02, -7.34514371e-02, -1.72763225e-02,
  1.13835242e-02, -1.63212884e-02, -5.34696281e-02,  5.32028750e-02,
 -1.44749098e-02,  3.59034464e-02,  1.85653958e-02,  5.90951927e-02,
  1.47780543e-02, -2.17742771e-02,  1.50234271e-02,  2.69742161e-02,
  9.79379192e-03,  6.51236475e-02, -4.73314198e-03,  1.31187811e-02,
  1.93645861e-02,  7.57417455e-02, -2.74137277e-02,  1.94943510e-02,
 -5.75501844e-03,  7.68118026e-03, -1.67309050e-03, -4.68759052e-02,
 -7.83467572e-03,  1.86435506e-02,  6.26649186e-02, -2.79785935e-02,
  6.60447478e-02,  7.25395456e-02,  3.40714306e-02, -3.05157658e-02,
  2.18674112e-02, -3.11621651e-02,  6.31265417e-02,  1.90785695e-02,
 -3.54125351e-02, -8.98810178e-02, -8.88275821e-03,  8.27952754e-03,
 -7.42505789e-02,  6.32991940e-02, -5.85555136e-02, -3.11240144e-02,
  7.45614385e-03,  3.36035006e-02, -3.36385928e-02, -3.62228416e-02,
 -2.25111637e-02, -3.21121365e-02,  1.06237046e-02, -3.13168541e-02,
  2.29744464e-02, -7.73122311e-02, -2.71368353e-03, -2.71095615e-02,
 -3.94065753e-02,  1.24143705e-01, -3.52797844e-02,  3.50417607e-02,
 -2.03845110e-02, -8.30745846e-02, -1.34285409e-02, -1.15193380e-02,
  8.32579955e-02,  6.08327519e-03,  1.97861549e-02, -4.40394022e-02,
 -4.70835753e-02,  3.52543071e-02,  3.41317244e-02, -1.77111034e-03,
```

**D**

```
[ 7.10385218e-02, -3.22032999e-03, -1.24619424e-03,  1.91412363e-02,
 -5.42471372e-03,  4.97934222e-02, -4.62391647e-03,  2.44413707e-02,
 -4.37092967e-02,  1.28905633e-02,  4.58273627e-02,  1.60606683e-03,
 -3.24402517e-03, -3.11120655e-02,  5.20463549e-02,  2.78522763e-02,
  1.21331312e-01,  5.84986359e-02, -2.24856790e-02,  6.56152982e-03,
  2.72995494e-02,  2.95747090e-02,  6.11840189e-02, -5.89872478e-03,
 -1.83749925e-02,  1.91699155e-02,  2.81136055e-02,  1.05330022e-02,
  2.49693021e-02,  1.23054804e-02,  3.61388363e-02,  6.87964931e-02,
  1.34759527e-02,  4.35568206e-02,  2.19232421e-02, -7.88312331e-02,
 -6.39629960e-02, -6.75716326e-02, -5.15788188e-03,  1.35627938e-02,
  4.90935482e-02, -7.30361789e-02, -6.72618253e-03,  3.84790963e-03,
 -3.24839801e-02,  3.11814086e-03,  1.42189832e-02, -2.38011386e-02,
 -4.32611741e-02,  2.29026433e-02, -3.68273519e-02, -2.12987065e-02,
  4.82287668e-02, -1.86451513e-03,  2.75416374e-02, -2.81859729e-02,
  2.01441273e-02, -5.03548756e-02,  5.85663095e-02,  5.18572219e-02,
 -3.08055952e-02,  6.18212484e-02, -3.48523930e-02,  6.40194491e-02,
  2.44354028e-02, -1.25454785e-02, -1.40200751e-02, -9.67387408e-02,
  1.91903841e-02, -4.41153497e-02, -6.71385787e-03,  1.30572531e-03,
 -3.06758154e-02,  3.16419564e-02, -1.22185666e-02,  9.03905705e-02,
 -8.64211842e-03, -4.80578952e-02,  9.80059337e-03, -3.30699869e-02,
  5.56746162e-02, -5.78230619e-02, -1.10484585e-01,  1.57824636e-03,
```

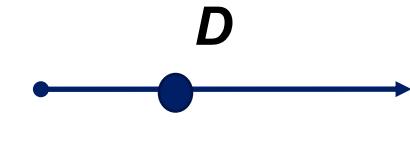
1. 计算距离得到相似度 **D** →

2. 根据相似度 **D** 判断是哪个人

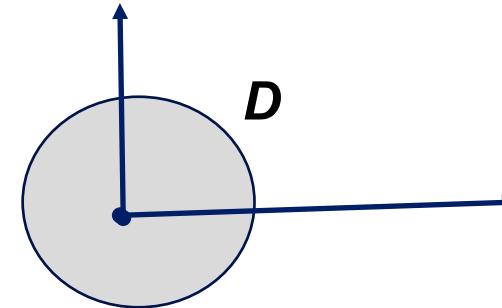
# 维度灾难：原因

维度越高，数据碰撞概率越高

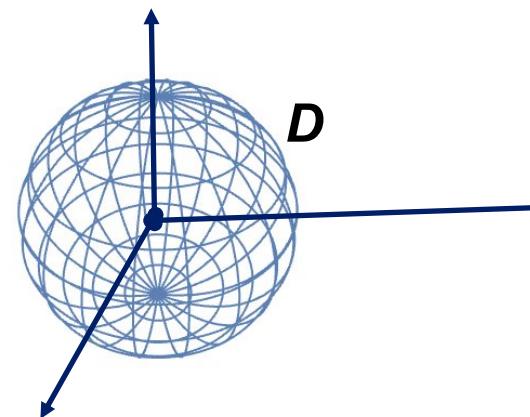
一维：根据距离 $D$ ，可以反向确定唯一的点(人)



二维：根据距离 $D$ ，只能确定一个范围圆上的点(人)



三维：根据距离 $D$ ，只能确定球面上的点(人)



我应该选择在多少个特征呢？

# 解决方法：特征工程

就是找到合适的特征组合



Filter过滤法

根据特征重要程度，过滤掉一些没用的特征，  
主要方法：方差、相关系数等。



Wrapper包装法

穷举全部特征组合，评估包括训练器的综合性能，得出最佳组合。



Embedded嵌入法

输入全部特征，在学习模型的过程中，挑选出那些对模型训练有重要性的特征。

# 问题 : Filter过滤法

---

要分析Disk usage情况

哪些特征是有用的 ?

#	Column	Non-Null Count	Dtype
0	MSP	34635 non-null	object
1	Company name	34635 non-null	object
2	Serial number	34635 non-null	object
3	AU enable	34635 non-null	bool
4	Frenquecy	34635 non-null	object
5	Version	34635 non-null	object
6	Status	34635 non-null	object
7	Register time(UTC)	34101 non-null	object
8	Last heartbeat(UTC)	34635 non-null	object
9	lastHeatbeat2now days	34635 non-null	object
10	User shutdown	34635 non-null	object
11	ActCode	34635 non-null	object
12	SSD version	34635 non-null	object
13	Hardware Model	34634 non-null	object
14	Mail Scan mode	34635 non-null	object
15	CompanyId	34635 non-null	object
16	DeviceId	34635 non-null	object
17	Disk Usage	34635 non-null	float64

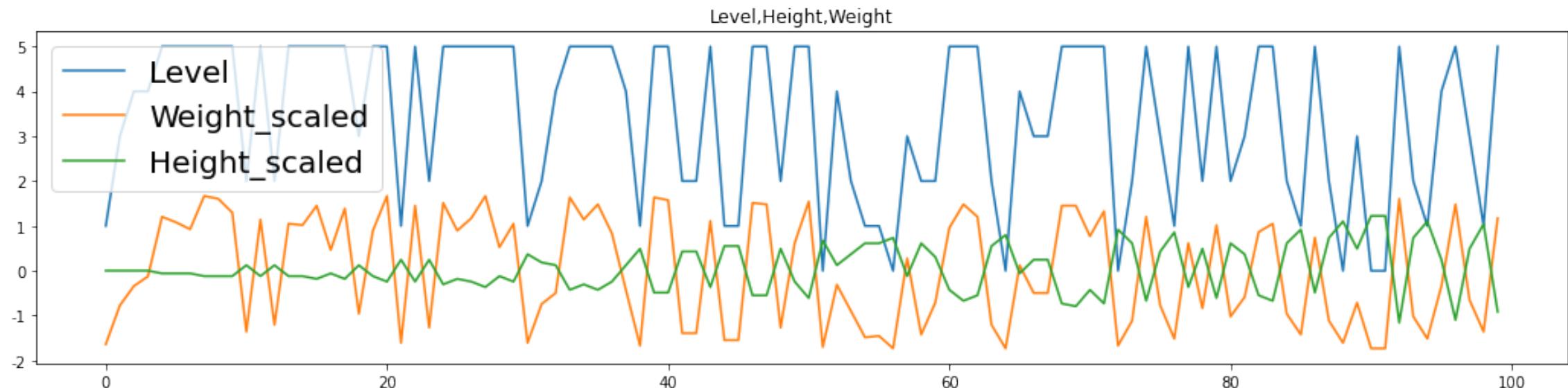
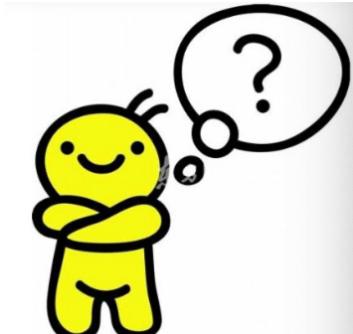
# Filter过滤法:相关系数

1. 将Level , Height , Weight放到一张图上(如下图);

如何度量这种关联程度?

2. 发现:相比于Height走势, Level和Weight走势更趋近;

3. 认为: Level 和Weight关联度更大, 和Height关联度更小;



# 相关系数: 方差

衡量数据的稳定性 ( 偏离均值的程度 )

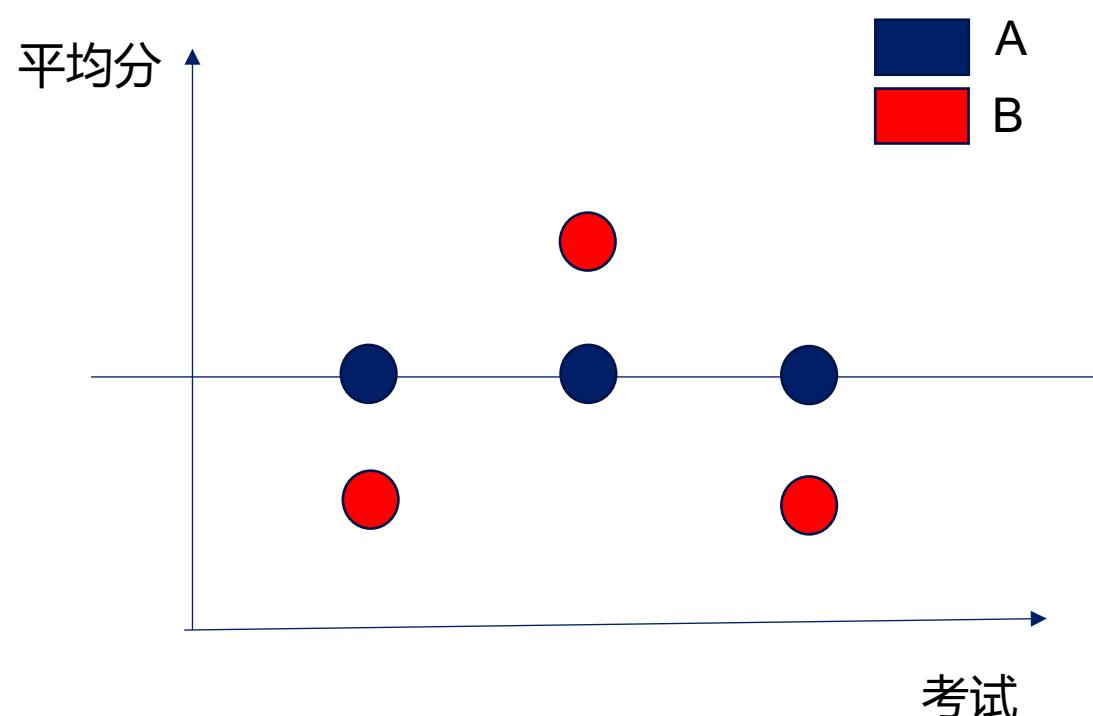
$$Var(x) = \frac{\sum(x_i - \bar{x})^2}{n}$$

考试成绩

学生	考试1	考试2	考试3	平均分
A	90	90	90	270
B	85	90	95	270

A的方差是 0  
B的方差是  $50/3$

学生A的成绩更稳定



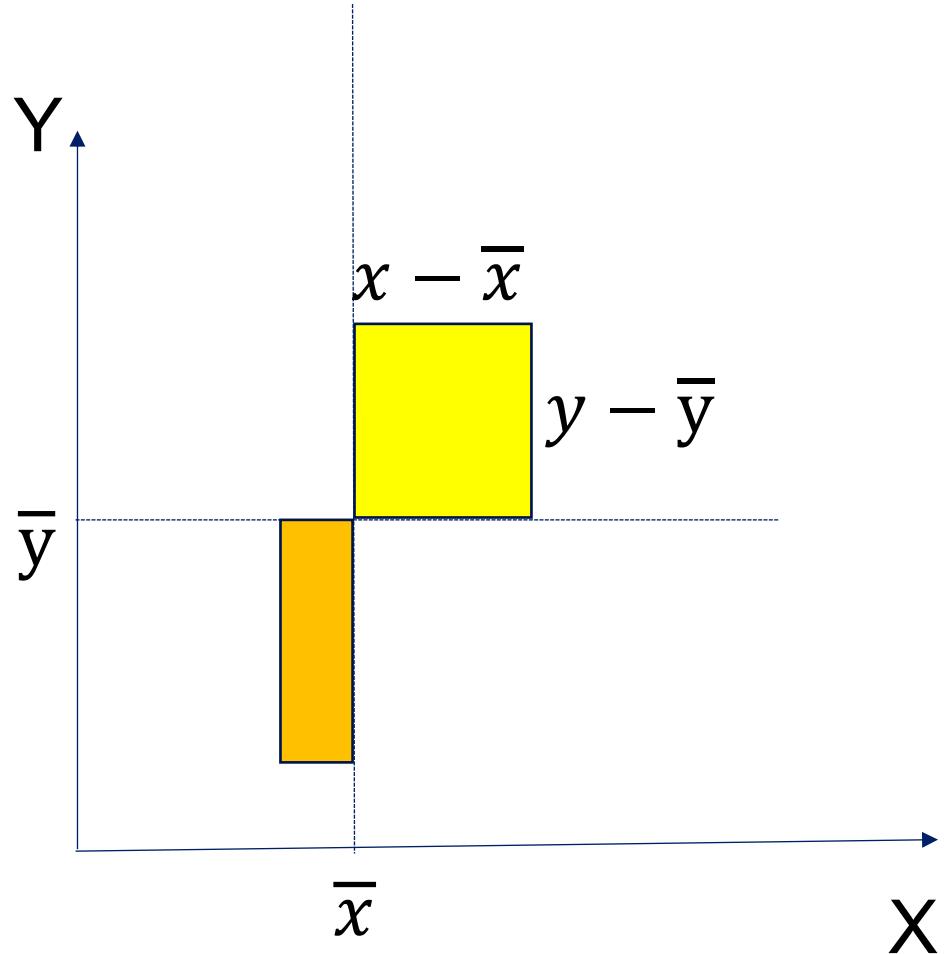
# 相关系数: 协方差

衡量两个变量的相关度

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1}$$

绝对值大有两个原因

- ①: 共振 ( $x, y$  关联度大)
- ② :  $X, Y$ 自身的方差大

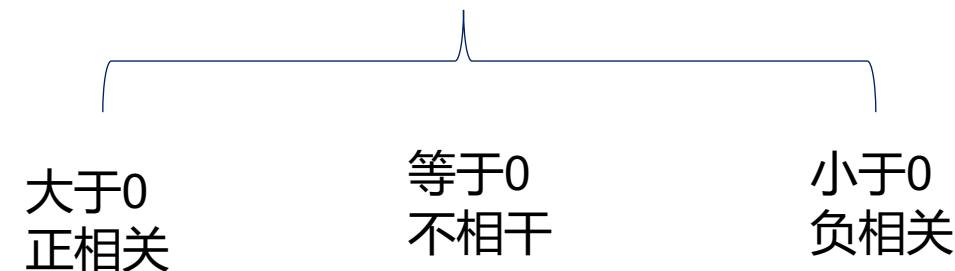


# 关联程度的度量：相关系数

衡量两个变量的相关度

剔除X,Y方差的影响，将协方差分别除以X,Y的标准差

$$\rho_{x,y} = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X) \text{Var}(Y)}}$$



# 样本中的方差和协方差

方差法 : df.var()

Height	268.15
Weight	1048.63
Fat	1.84
..	.. ..

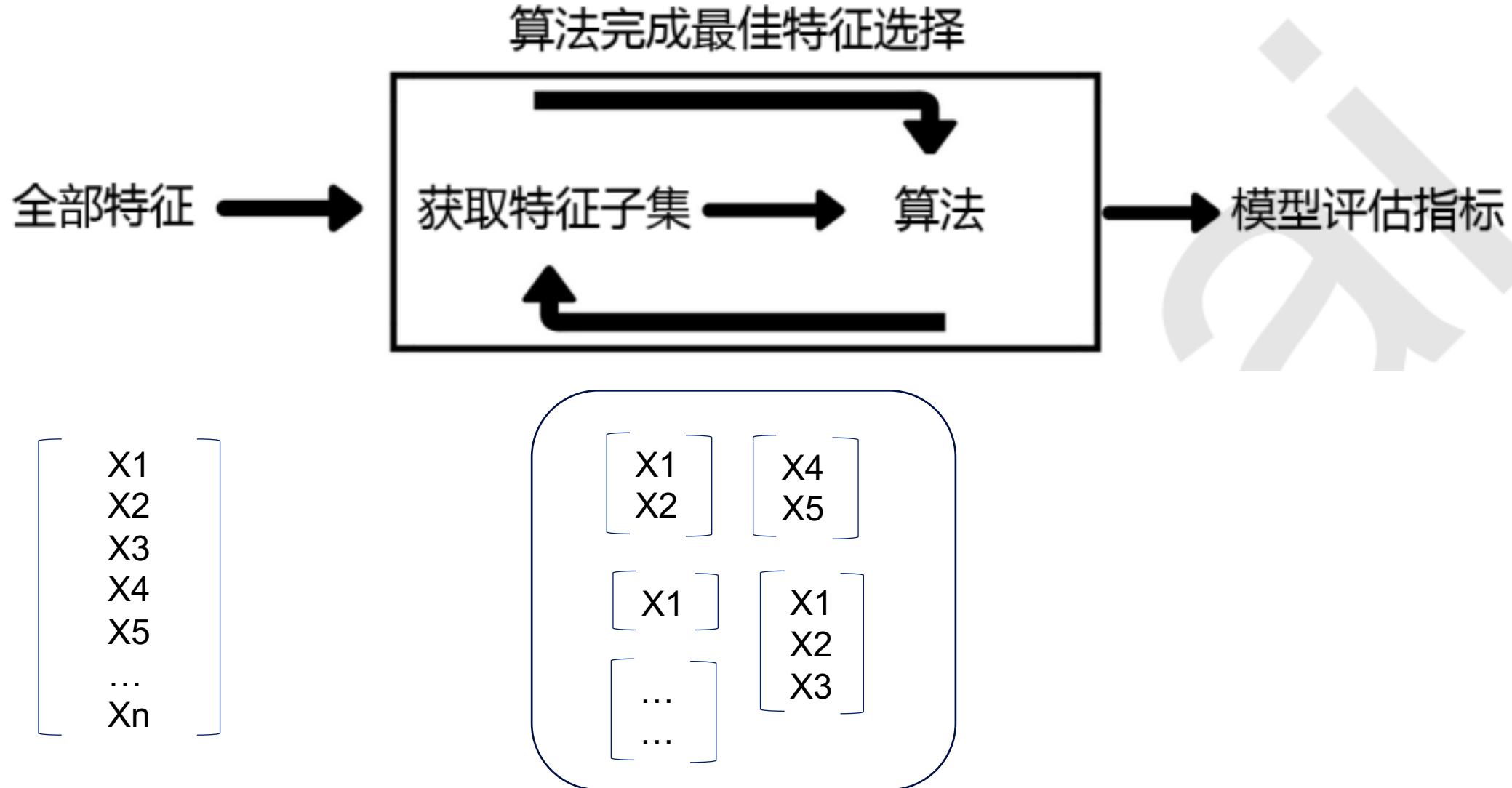
取最大，选特征Weight

相关系数法 : df.corr()

	Height	Weight	Level
Height	1.00	0.00	-0.42
Weight	0.00	1.00	0.80
Level	-0.42	0.80	1.00

取最大，取特征Weight

# Wrapper包装法



# Embedded嵌入法：问题

---

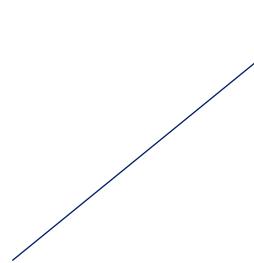
## 举例：

- 现有一个task，领导要求评估effort；
- 评估方法：总effort = 研发人数\*develop 时间 + 测试人数\*test时间；
- 实际上，项目不同，需要的RD或QA的数量也不同
- 如何设计改进评估方法，弹性更好，效率更高？

# Embedded嵌入法

---

- 总effort = 研发人数\*develop 时间 + 测试人数\*test时间；
- 原来算法不变，同时要求研发人数 + 测试人数 最小；



在学习模型的过程中，利用正则化思想，将部分特征属性的权重变成零。常见的正则化有L1的Lasso，L2的Ridge和混合的Elastic Net。

什么是正则化，Ridge, LASSO, Elastic Net ?



# 正则化

举例：考试成绩

学生	考试1	考试2	考试3	平均分
A	90	90	90	90
B	85	90	95	90

学生A的成绩更稳定

模型  $Y = W_1 * X_1 + W_2 * X_2 + W_3 * X_3$

模型	$W_1$	$W_2$	$W_3$	准确率
A	90	90	90	95%
B	85	90	95	95%

模型A的表现更稳定

使用正则项( $W_1^n + W_2^n + W_3^n$ 表示稳定性)

- $\sum W_i^n$  越小越稳定 (n为自然数)
- n=1 称L1正则
- n=2 称L2正则

# 线性回归增强：LASSO回归模型

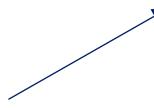
---

其实就是线性回归加上正则项 ( $\lambda$ 为惩罚系数)

预测函数与线性回归相同 : 
$$Y = \omega_1 x_1 + \omega_2 x_2 + b$$

总误差 ( 线性回归 ) : 
$$J = \frac{1}{2} \sum (y_i - (w_1 x_i + w_2 x_i + b))^2$$

总误差 ( LASSO ) : 
$$J = \frac{1}{2} \sum (y_i - (w_1 x_i + w_2 x_i + b))^2 + \lambda(w_1 + w_2)$$



求总误差最小值时，会使得部分  $\omega$  值很小或为 0；

从而使某些特征失效，有特征选择作用；

# 总结：三化的常用场景

1 归一化

数据探索阶段

消除数据在  
量级上差别

2 标准化

特征工程阶段

让数据特征  
更容易显现

3 正则化

模型训练阶段

让数据拟  
合更稳定

# PCA:问题

## 文件信息

文件大小(kb)  
文件类型  
文件扩展名  
文件版本  
生产厂商  
磁盘占用(mb)  
hash值  
.....

(文件大小和占用空间意义相同)



哪些是有用信息？

如何将用处不大的信息剔除掉？

## 程序日志

```
logformat = [
    'CLF_CompanyID',
    'CLF_MsgType',
    'CLF_ProductID',
    'CLF_DeviceName',
    'CLF_UTCTimeStamp',
    'CLF_TimeZone',
    'CLF_UserName',
    'CLF.UserID',
    'CLF_GroupName',
    'CLF_Department',
    'CLF_Location',
    'CLF_URL',
    'CLF_Size',
    'CLF_AppID',
    'CLF_Channel',
    'CLF_TransportProtocol',
    'CLF_MimeType',
    'CLF_ClientIP',
    'CLF_ServerIP',
    'CLF_Domain',
    'CLF_FileName',
    'CLF_ProtocolAction',
    'CLF_Connection',
    'CLF_URLCategory',
    'CLF_CustomURLCategory',
    'CLF_AdvancedCategory',
    'CLF_OnlineTime',
    'CLF_MalwareName',
    'CLF_RuleName',
    'CLF_MatchedContent',
    'CLF_Action',
    'CLF_PolicyName',
    'CLF_Phase',
    'CLF_SessionPackets',
    'CLF_SessionBytes',
    'CLF_InByte',
    'CLF_OutByte',
    'CLF_WRSScore',
    'CLF_SessionStartTime',
    'CLF_SessionEndTime',
    'CLF_ActionTime',
    'CLF_Direction',
    'CLF_SourceInterface',
    'CLF_DestinationInterface',
    'CLF_SourcePort',
    'CLF_DestinationPort',
    'CLF_SourceZone',
    'CLF_DestinationZone',
    'CLF_IPSRule',
    'CLF_ErsCategory',
    'CLF_MailSender',
    'CLF_MailReceiver',
    'CLF_MailSubject',
    'CLF_RiskLevel'
]
```

(存在大量冗余的信息)

# 降维:PCA



不是简单掉某些特征，而是重新构造特征

降维的主要目的是减少特征个数、确保这些特征变量相互独立。

主成分分析(Principal Component Analysis)简称PCA，PCA主要应用在将高维度空间压缩降维到低维度空间，同时尽可能多的保留原始特征，在低维空间更容易进行可视化展示。

PCA算法流程：



# PCA：协方差矩阵的特征值和特征向量

协方差： $cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1}$

协方差矩阵：

$$\begin{bmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) & Cov(X_1, X_3) \\ Cov(X_2, X_1) & Cov(X_2, X_2) & Cov(X_2, X_3) \\ Cov(X_3, X_1) & Cov(X_3, X_2) & Cov(X_3, X_3) \end{bmatrix}$$

可以找到  $\lambda, X$  使得

$$AX = \lambda X$$

$x$  为向量，是矩阵 A 的特征向量  
 $\lambda$  为实数，是矩阵 A 的特征值

# PCA : 过程

步骤1 : 每列减去其平均值得到new\_A

	x1	x2	x3	x1-1.64	x2-1.98	x3-1.82	
A	2.3	2.4	2	0.66	0.42	0.18	
	0.4	0.6	0.7	-1.24	-1.38	-1.12	
	1.5	2.9	2	-0.14	0.92	0.18	
	1.9	2.7	2.4	0.26	0.72	0.58	
	2	3	1.5	new_A	0.36	1.02	-0.32
	3	2.9	2.7		1.36	0.92	0.88
	1	1.6	2.3		-0.64	-0.38	0.48
	1.5	1	2.1		-0.14	-0.98	0.28
	2.2	1.8	1.7		0.56	-0.18	-0.12
	0.6	0.9	0.8		-1.04	-1.08	-1.02

x1,x2,x3的平均值分布为1.64 , 1.98和1.82

步骤2 : 计算协方差矩阵B

0.65	0.18	0.06
0.18	0.76	0.30
0.06	0.30	0.38

步骤3 : 计算矩阵B的特征值和特征向量

1.03	0.56	0.21
特征值		
0.45	0.89	0.08
0.79	-0.36	-0.50
0.41	-0.29	0.86

特征向量

# PCA : 过程

步骤4：取前两个特征值和特征向量

1.03	0.56	0.45	0.89
0.79	-0.36		
0.41	-0.29		

特征值

特征向量



K值通过实验来确定，看特征值占据所有特征值的比例。本例中特征值有三个：

1.03, 0.56, 0.21

$$1.03/(1.03+0.56+0.21)=0.573$$

$$0.56/(1.03+0.56+0.21)=0.311$$

前两个特征所占百分比为0.884即88.4%可以认为用前两个特征值对应的特征的话，会占据全部能信息量的88.4%，付出的代价是少了一列特征。

步骤5： New\_A 乘以新特征向量得到最终数据

0.70561734	0.38275488
0.98901302	-1.91965147
0.73682807	-0.50698361
0.92611898	-0.19545067
0.83601875	0.04496213
1.7076029	0.62227539
-0.39199146	-0.56972824
-0.72108241	0.14742213
0.06266525	0.59629137
-1.74642821	-0.24038943

# PCA : 实现

```
1 from sklearn.decomposition import PCA
2 x1=df2['Height_scaled'].tolist()
3 x2=df2['Weight_scaled'].tolist()
4 X = []
5 for xx1,xx2 in zip(x1,x2):
6     X.append([xx1,xx2,xx1*xx1,xx2*xx2,xx1*xx2,xx1+xx2])
7 X = np.array(X)
8
9 pca = PCA(n_components=4)
10 newX = pca.fit_transform(X)
11 print("特征贡献率",pca.explained_variance_ratio_)
12 print("4主元特征贡献率",np.sum(pca.explained_variance_ratio_))
13 print("特征数量对比",X.shape,newX.shape)
```

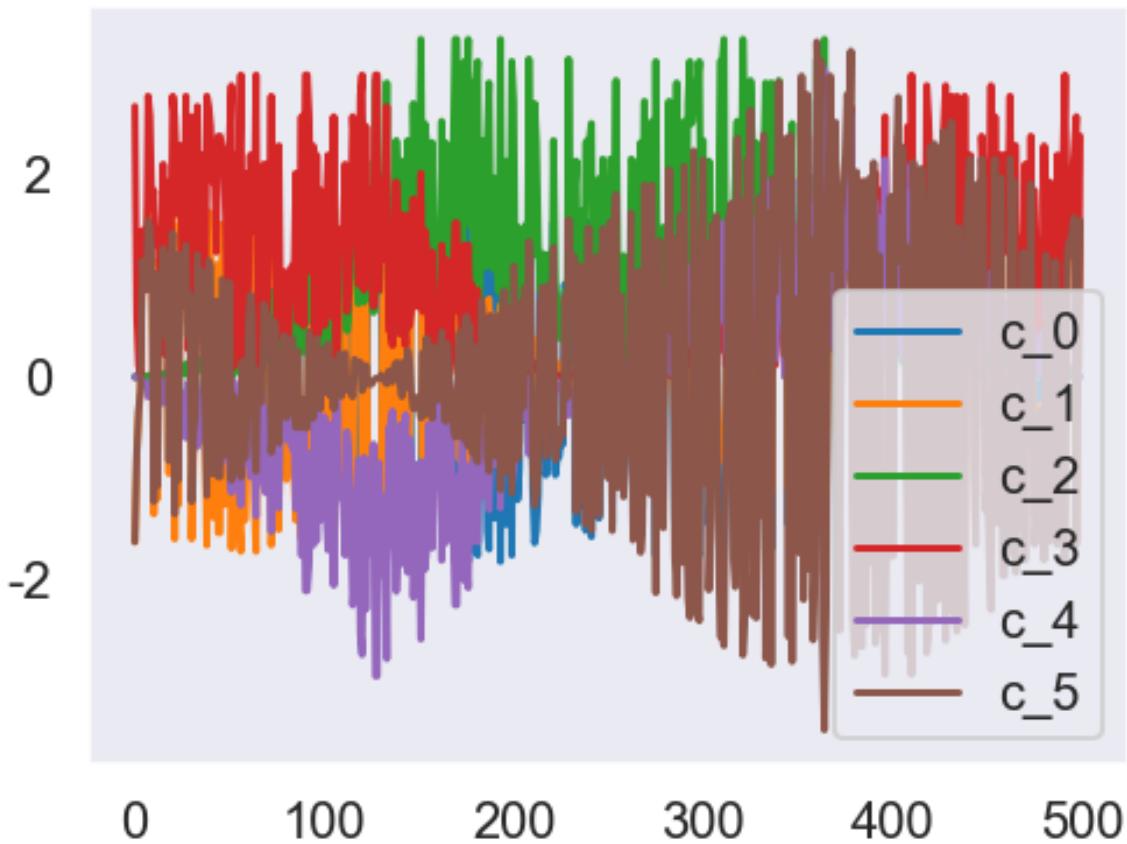
特征贡献率 [0.45314742 0.15557184 0.14965998 0.12586158]

4主元特征贡献率 0.8842408317248092

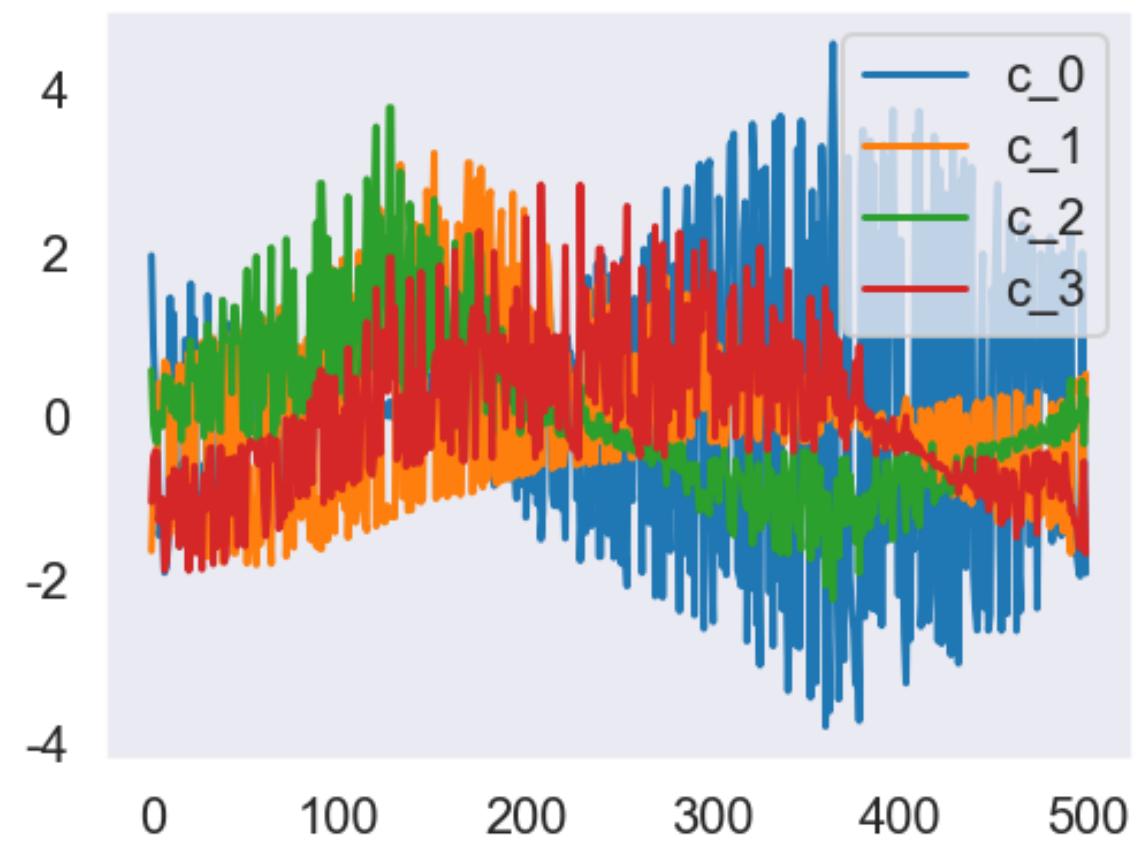
特征数量对比 (500, 6) (500, 4)

# PCA：给试验样本降维

重新构造特征



降维前



降维后

# PCA：在试验样本中的应用

---

通过上面的试验，在试验样本中；

重新组合后，我们可以用4个特征，可以表示原来样本88%的信息量；

我们可以将这4个特征输入模型，进行训练。（原来需要6个特征）

# 小结

---



- 特征提取方法
- 协方差与相关系数
- 正则化与线性模型的扩展
- PCA降维

# Train and Evaluation

---

[TBD]

# Thanks

2020-8-15

