西安交通大学
XI'AN JIAOTONG UNIVERSITY

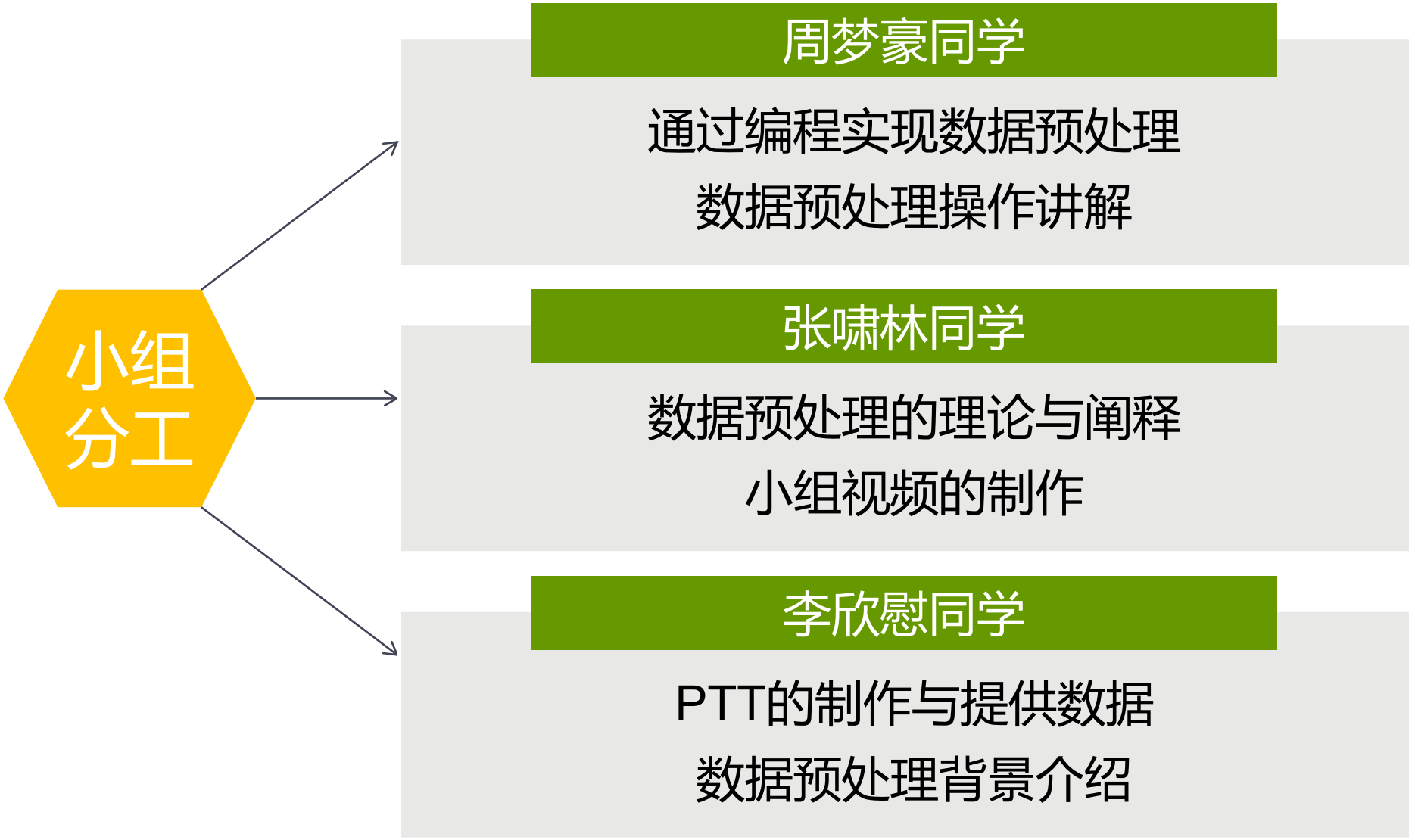
数 据 预 处 理
在材料化学领域的应用

工业工程71 周梦豪
光信61 张啸林
材化61 李欣慰

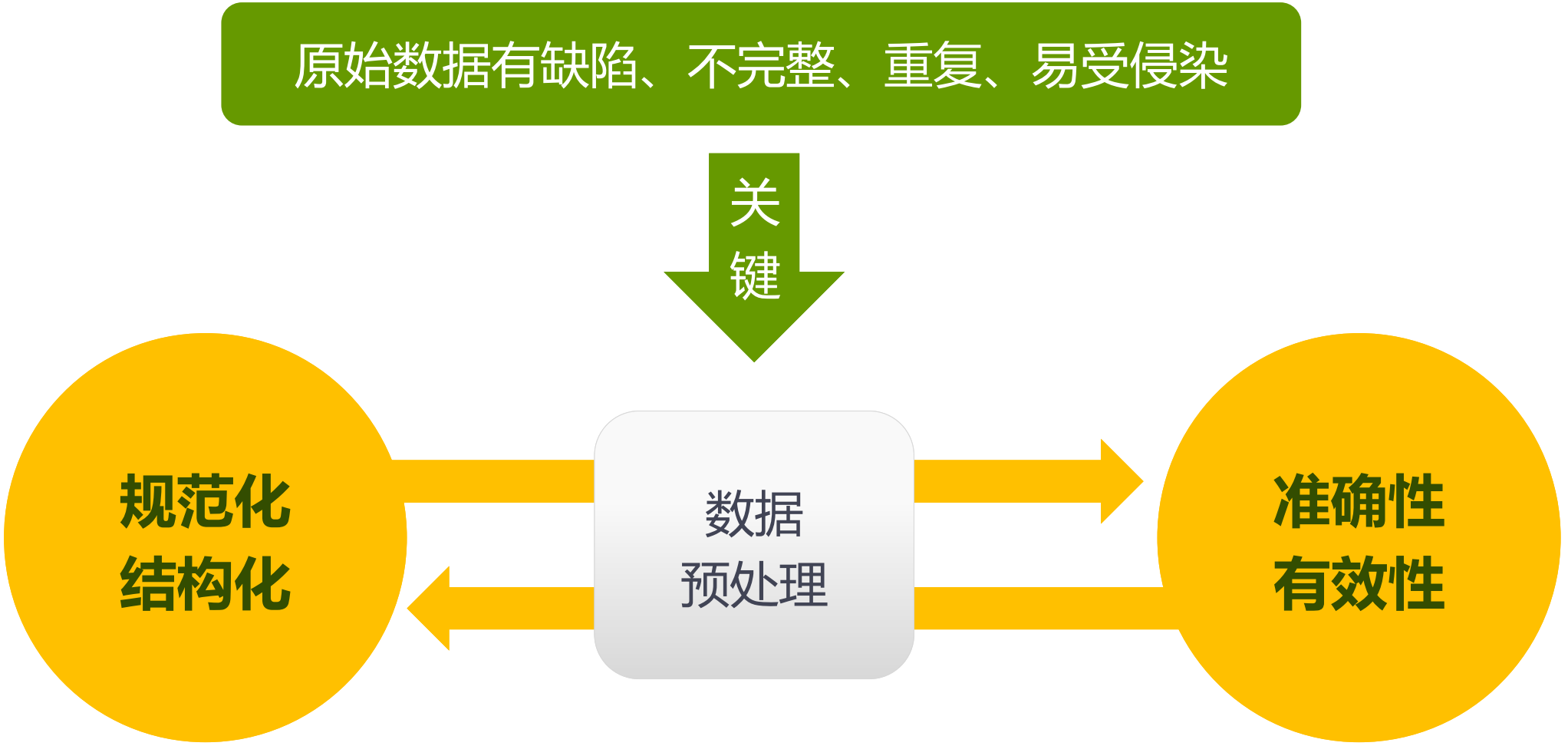
2020.03.24

目录
CONTENTS

Part 1 小组分工


小组分工
周梦豪同学
通过编程实现数据预处理
数据预处理操作讲解
张啸林同学
数据预处理的理论与阐释
小组视频的制作
李欣慰同学
PTT的制作与提供数据
数据预处理背景介绍

Part 2 背景介绍



Part 2 背景介绍

数据清洗

处理奇异值、离群点、
重复信息、噪声干扰

数据集成

数据集中、匹配、统一

数据变换

数据光滑、聚集、概化、
规范、特征构造

数据归约

维归约、样本归约、
数据压缩、离散化

典型问题与方法

缺失值处理	异常值判别	噪声处理	重复值处理
删除 均值插补 就近补齐 多重插补 回归 极大似然估计	简单统计分析 3δ原则 基于模型判别 基于密度判别	分箱法 聚类法	降低权重 混合删除机制

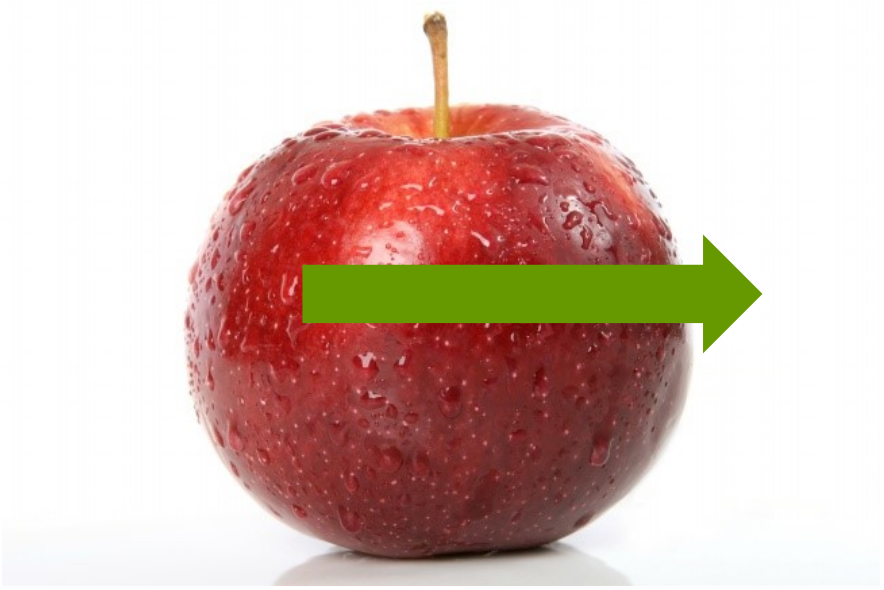
Part 3 数据预处理

什么是数据预处理？

本质

臣好食，请以食喻。

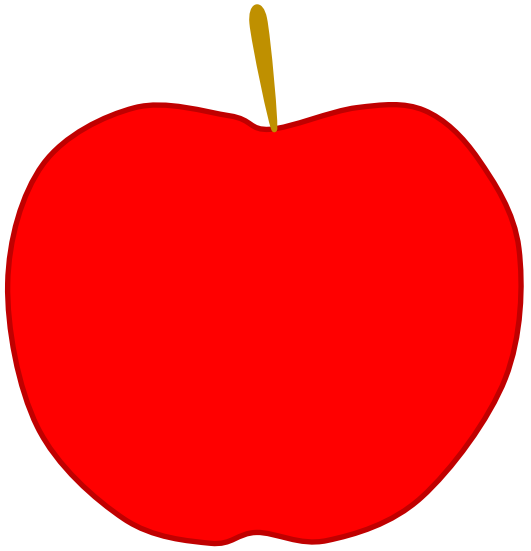
数据？



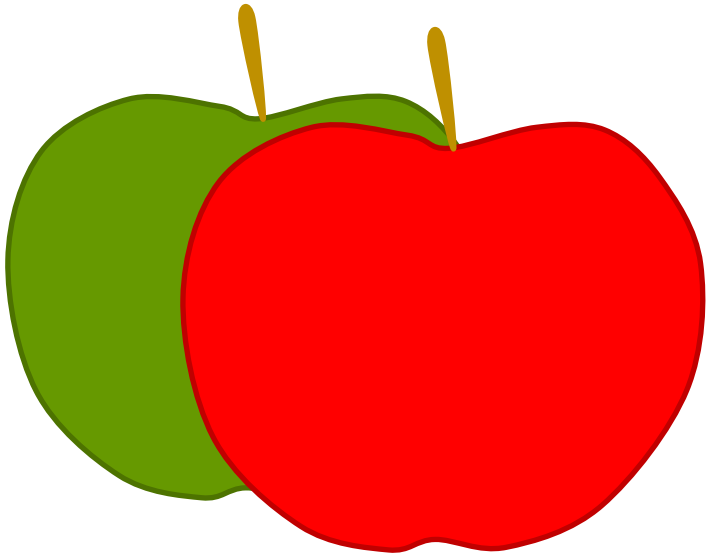
The Apple

数据清洗

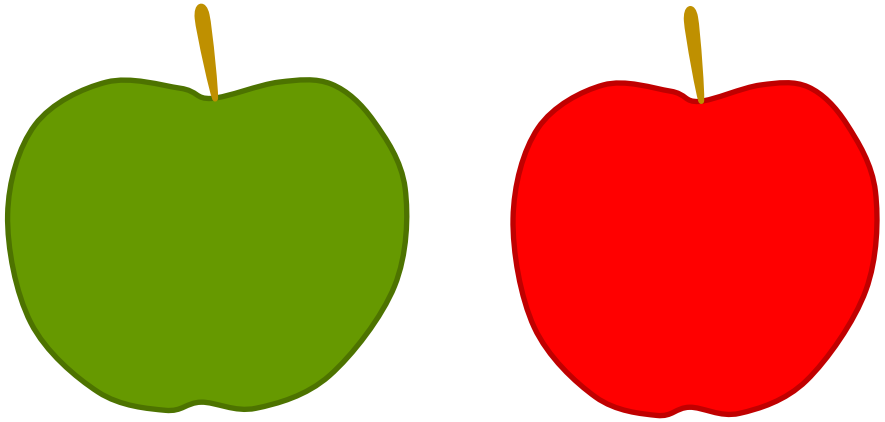
苹果的本质在一定范围内稳定存在



数据集成

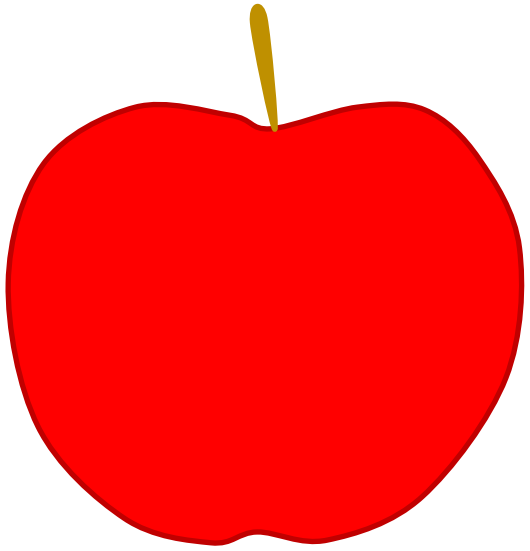


编年体
记录对比



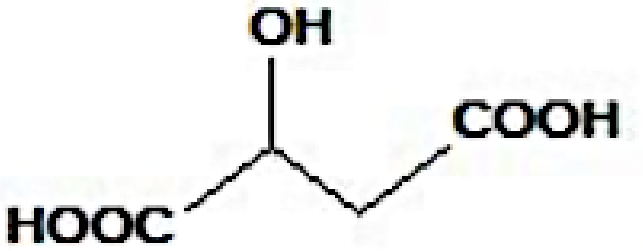
纪传体
动态变化

数据变换



被子植物门蔷薇科苹果属

苹果酸：



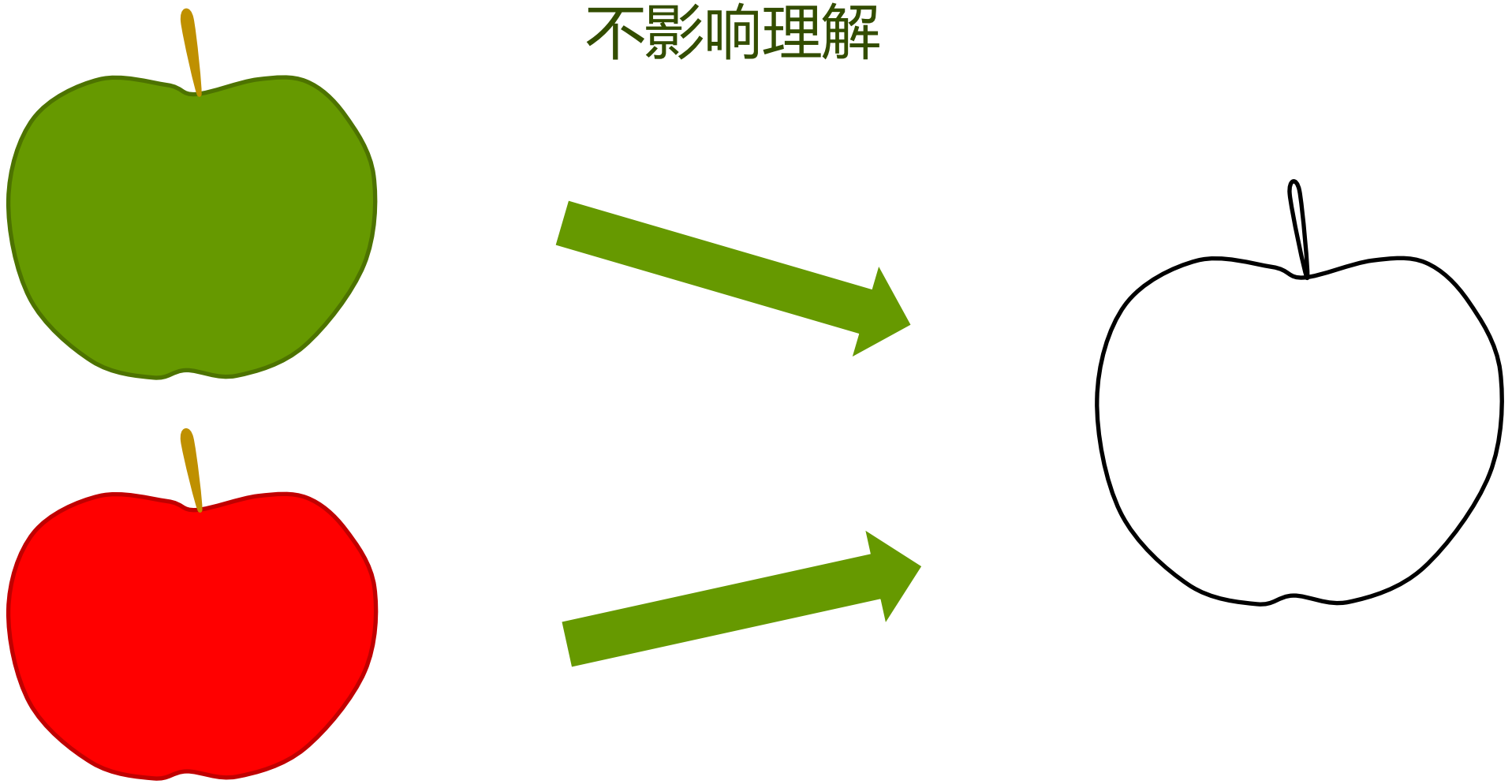
营养成分：维C, 维B, 视黄醇等

日常生活数据

计算机/专业人员处理数据

数据规约

不影响理解



对介电温谱数据进行预处理

数据描述
数据集成

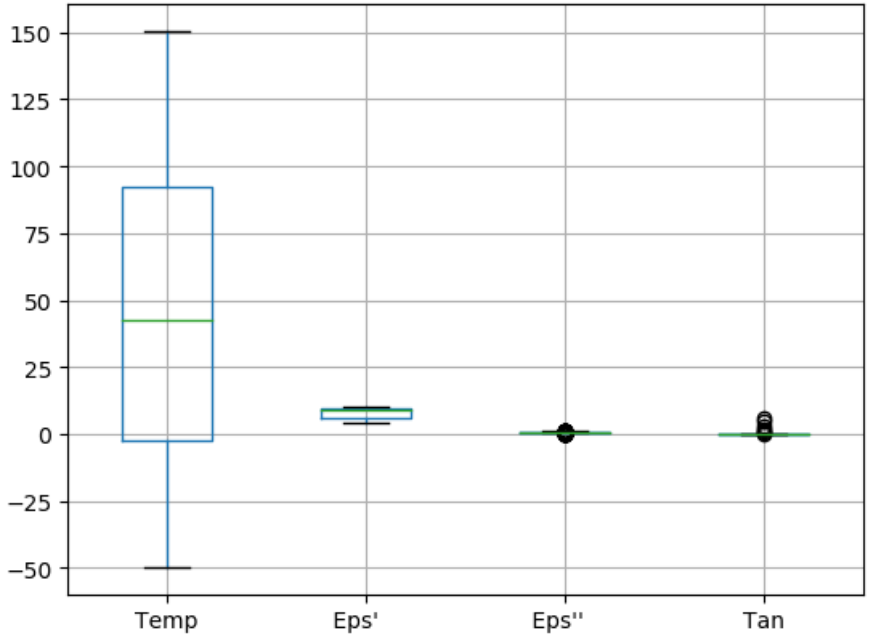
三个维度：
- Temp.，温度
- Eps’，介电常数
- Tan(Delta)，介电损耗

	Temp	Eps'	Eps''	Tan
count	305.000000	315.000000	315.000000	299.000000
mean	44.775166	7.953906	0.631273	0.119506
std	57.603877	2.090945	0.212408	0.295083
min	-49.687300	4.206410	0.180805	0.010000
25%	-2.500000	5.966475	0.547586	0.062356
50%	42.688900	8.955120	0.625427	0.064709
75%	92.579000	9.788255	0.730798	0.108160
max	150.492000	9.960800	1.029360	3.000000

Part 4 实例分析

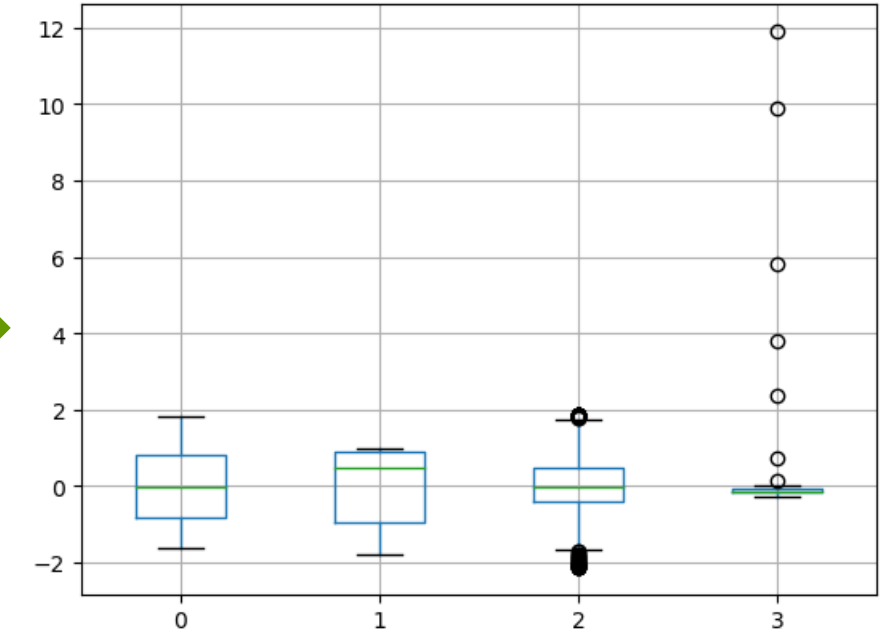
异常值处理

寻找异常值



均值-标准差缩放

数据变换

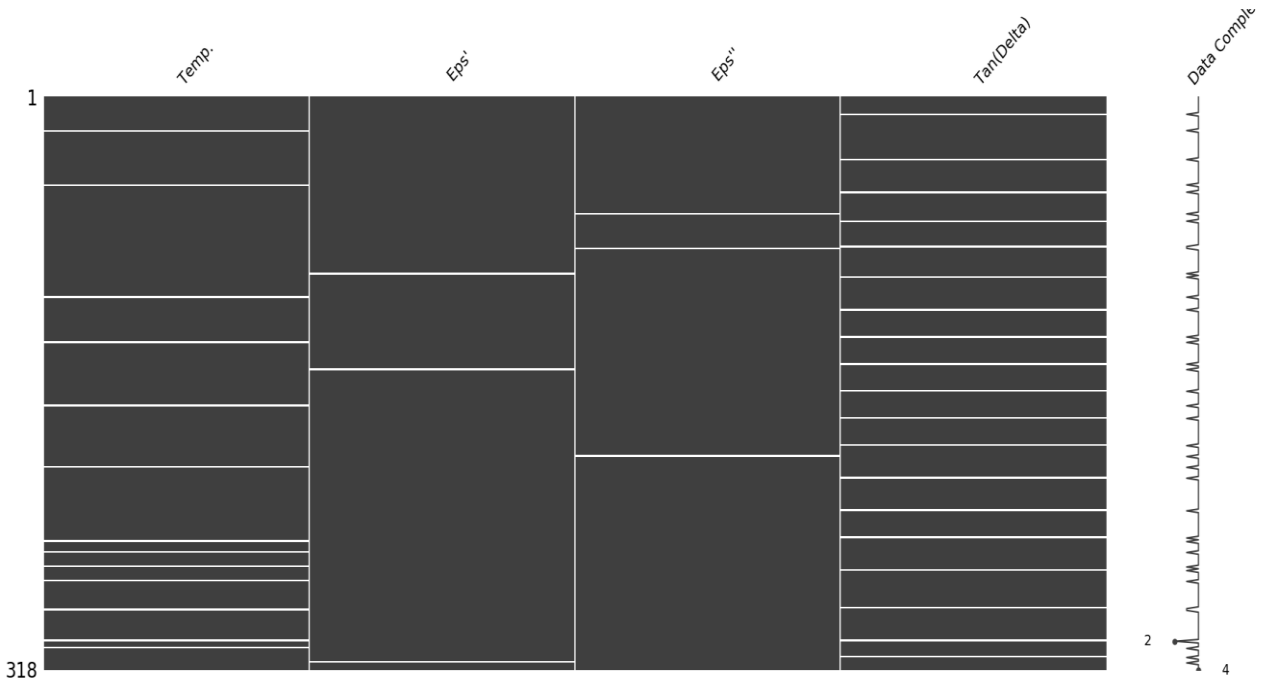


进行修正

Part 4 实例分析

缺失值处理

寻找缺失值




```
(array([ 10,  19,  35,  49,  53,  65,  69,  83,  84,  98, 100, 111, 118,
        133, 136, 148, 151, 163, 171, 178, 193, 199, 205, 211, 229, 244,
        246, 252, 260, 262, 268, 283, 284, 301, 301, 305, 310, 313],
       dtype=int64), array([3, 0, 3, 0, 3, 2, 3, 3, 2, 1, 3, 0, 3, 3, 0, 3, 1, 3, 0, 3, 3, 2,
        0, 3, 3, 3, 0, 0, 0, 3, 0, 3, 0, 0, 3, 0, 3, 1], dtype=int64))
```

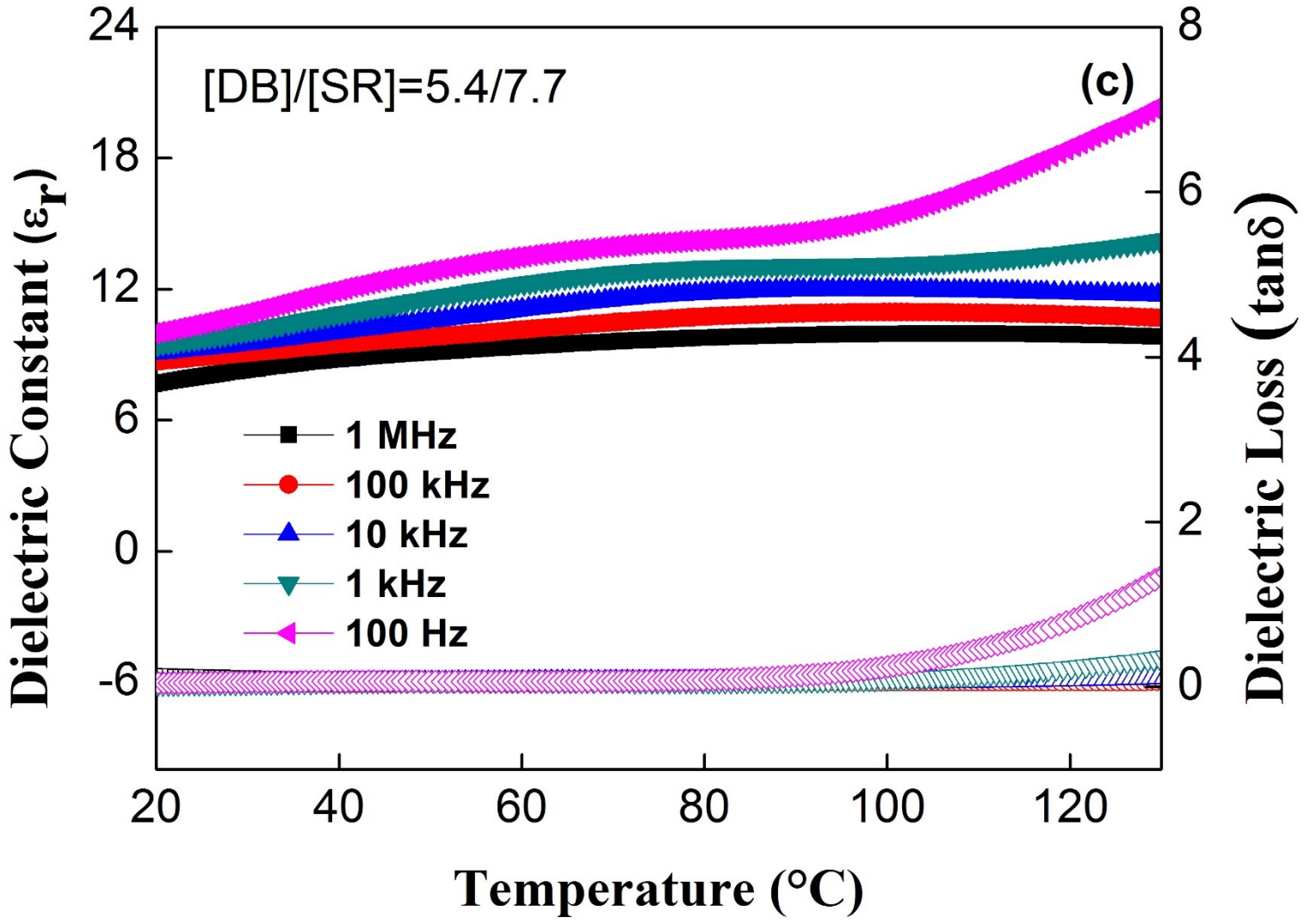
缺失值处理

填充　　删除

介电损耗数据

填充方式	训练集(75%)	测试集(25%)
均值	12.69%	27.41%
前后均值	23.51%	13.70%
中位数	46.28%	42.48%
KNN-3	78.39%	68.37%
KNN-6	71.07%	65.49%
KNN-10	68.47%	61.60%

数据规约


[DB]/[SR]=5.4/7.7
(c)
1 MHz
100 kHz
10 kHz
1 kHz
100 Hz
Dielectric Constant (εr)
Dielectric Loss (tanδ)
Temperature (°C)
24
18
12
6
0
-6
8
6
4
2
0
20
40
60
80
100
120

Part 5 总结与展望

周梦豪：拓展Python编程知识，提高讲解能力

张啸林：讲解生动形象，合理讨论问题，个人理解

李欣慰：PPT准确表达思想，团队合作能力

参考文献

[1]周泉锡.常见数据预处理技术分析[J].通讯世界,2019,26(01):17-18.
[2]周党生.大数据背景下数据预处理方法研究[J].山东化工,2020,49(01):110-111+122.
[3] Python特征缺失值填充. https://www.cnblogs.com/Allen-rg/p/9488249.html
[4]K-近邻算法分类与回归.
https://blog.csdn.net/luckyflyyy/article/details/89463692?depth_1utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task

附录

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
data = pd.read_csv('C:\\Users\\Angle豪\\Desktop\\预处理\\介电温谱\\LXY-3-S0.csv')
data.dropna(axis=0, how='any', inplace=True)

plt.plot(data['Tan'], data['Temp'], "b.-")

plt.show()
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl

from fancyimpute import KNN

inputfile = "C:\\Users\\Angle豪\\Desktop\\预处理\\电滞回线
\\252.csv"



data1 = pd.read_csv(inputfile)



def wherena(data=data1):
    print(np.where(np.isnan(data)))

def deletehang(data=data1):
    cleaned1 = data.dropna()
    print(cleaned1)
    # cleaned1.to_csv("C:\\Users\\Angle豪\\Desktop\\预处理\\电滞
回线\\252拉格朗日插值.csv", sep=',', header=True, index=False)
    return cleaned1
```

```
def chazhi1(data=data1):
    data['Measured Polarization'] = data['Measured
Polarization'].interpolate()
    data.to_csv("C:\\Users\\Angle豪\\Desktop\\预处理\\电滞回线
\\252相邻插值.csv", sep=',', header=True, index=False)
def chazhi2(data=data1):
    # KNN
    fill_knn = KNN(k=3).fit_transform(data)
    data = pd.DataFrame(fill_knn)
    print(data.head())
def figure(data=data1):
    plt.figure(figsize=(8, 6))
    plt.scatter(data['Time (ms)'], data['Measured Polarization'],
marker='o', color='g', alpha=0.7, label='1.0')
    # plt.title('dataset')
    # plt.ylabel('variable Y')
    # plt.xlabel('Variable X')
    # plt.legend(loc='upper right')
    plt.show()

if __name__ == '__main__':
    # wherena()
    # deletehang()
    # figure()
    chazhi2()
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
# from fancyimpute import KNN
# import os
# os.environ['FANCYIMPUTE_BACKEND'] = 'tensorflow'
inputfile = "C:\\Users\\Angle豪\\Desktop\\预处理\\介电温谱\\LXY-3-S0.csv"
data1 = pd.read_csv(inputfile)
def jichutongji(data=data1):
    print(data.describe())
def wherena(data=data1):
    print(np.where(np.isnan(data)))
def deletehang(data=data1):
    cleaned1 = data.dropna()
    print(cleaned1)
    # cleaned1.to_csv("C:\\Users\\Angle豪\\Desktop\\预处理\\电滞回线\\252拉格朗日插值.csv", sep=',', header=True, index=False)
    return cleaned1
def chazhi1(data=data1):
    data['Measured Polarization'] = data['Measured Polarization'].interpolate()
    data.to_csv("C:\\Users\\Angle豪\\Desktop\\预处理\\LXY-3-S0.csv", sep=',', header=True, index=False)
def chazhi2(data=data1):
    # KNN
    fill_knn = KNN(k=3).fit_transform(data)
    data = pd.DataFrame(fill_knn)
    print(data.head())
def figure(data=data1):
    plt.figure(figsize=(8, 6))
    plt.scatter(data['Time (ms)'], data['Measured Polarization'], marker='o', color='g', alpha=0.7, label='1.0')
    # plt.title('dataset')
    # plt.ylabel('variable Y')
    # plt.xlabel('Variable X')
    # plt.legend(loc='upper right')
    plt.show()
def box(data=data1):
    plt.figure()  # 建立图像
    p = data.boxplot()  # 画箱线图，直接使用DataFrame的方法
    plt.show()
def guifanhua(data=data1):
    (data['Tan(Delta)'] - data['Tan(Delta)'].min())/(data['Tan(Delta)'].max() - data['Tan(Delta)'].min()) #最小-最大规范化
    print(data)
if __name__ == '__main__':
    wherena()
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model.logistic import LogisticRegression
from fancyimpute import BiScaler, KNN,
NuclearNormMinimization, SoftImpute
from sklearn import neighbors
def countF1(train):
    count = 0 # 统计预测的正确的正样本数
    stdd = np.std(train)
    for i in range(2, len(train)-2):
        predict = (train[i-1]+train[i+1])/2
        if  predict>= train[i]-0.05*stdd and predict <=
train[i]+0.05*stdd:
            count += 1
    pre = count * 1.0 / sum(train) # 准确率
    recall = count * 1.0 / sum(train) # 召回率
    print(stdd)
    return 2 * pre * recall / (pre + recall)
data = pd.read_csv('C:\\Users\\Angle豪\\Desktop\\预处理\\介电温
谱\\LXY-3-S01.csv')
# 1000,85
data.dropna(axis=0, how='any', inplace=True)
filter_feature = ['Tan']  # 过滤无用的维度
```

```
features = []
for x in data.columns:  # 取特征
    if x not in filter_feature:
        features.append(x)
train_data_x = data[features]

# train_data_x =
pd.DataFrame(KNN(k=5).fit_transform(train_data_x),
columns=features)

train_data_y = data['Tan']
X_train, X_test, y_train, y_test = train_test_split(train_data_x,
train_data_y, random_state=1, test_size=0.25)  # 划分训练集、测
试集

predict = []

# linreg = LogisticRegression()
# linreg.fit(X_train, y_train.astype('int')) # 模型训练
predict2 = np.mean(y_test)

print("训练集", countF1(y_train.values))

print("测试集", countF1(y_test.values))
```

西安交通大学
XI'AN JIAOTONG UNIVERSITY

THANKS!

工业工程71 周梦豪
光信61 张啸林
材化61 李欣慰

2020.03.24