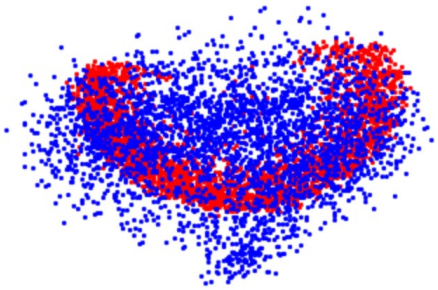
实践题结果解析

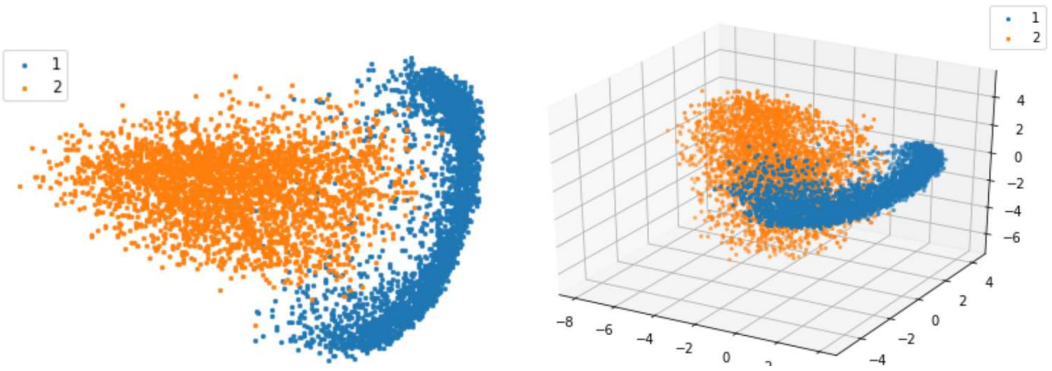
T1：编程实现 MDS“求解方法二”的算法，并分别对 MNIST12 数据集做二维和三维降维

编程思路为：①读取数据集，筛选出 1 类与 2 类的样本；②计算 D 矩阵，然后计算 B 矩阵；③进行特征值分解，选择前 k 大的特征值及其特征向量，计算降维后的样本矩阵。

在计算 D 矩阵与 B 矩阵时，我最开始是考虑对于两类样本，分别计算两组 D 矩阵与 B 矩阵，再分别进行降维，最终的降维效果如下图：



如图所示，两类样本分别降维效果并不好，正如助教姐姐分析说到的那样：“如同用两个矩阵的话，两类数据降维后的相对位置就无法保证了，可能本不混杂的数据会发生混杂。”所以应当将所有样本一起计算 D 矩阵与 B 矩阵，这样可以保留两类间的距离关系特征。最终降至两维和三维的可视化效果如下：



由于电脑能力限制，我只选择了 1 类和 2 类各 3000 个点进行降维展示，降维结果保存为 npy 文件一起上交，助教检查时节约时间可以直接运行 notebook 中最后一个 block，可以读取 npy 文件直接展示。

T2：实现自顶向下或自下往上层级聚类，分别使用平均欧氏距离和 Ncut 值作为两类 Ci 和 Cj 的距离/相似性度量，并对西瓜 3.0 做聚类分析

答：编程思路：①读取数据，并用 one-hot 对离散属性进行编码；②编写计算平均欧氏距离的函数与计算 Ncut 值的函数；③采取自底向上的层次聚类算法，即先将每一个样本单独认作一类，逐步合并。

为了展示聚类过程，我对原来的样本进行编号如下（最左侧一列）：

0	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	1
1	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	1
2	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	1
3	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	1
4	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	1
5	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	1
6	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	1
7	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	1
8	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	0
9	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	0
10	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	0
11	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	0
12	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	0
13	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	0
14	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	0
15	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	0
16	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	0

聚类过程在程序运行中并不方便展示，我输出了每次被聚合在一起的新类别的密度与糖度用于鉴别是哪些样本被聚合了，最终我将聚类过程描述如下：

欧氏距离：

(1)2 号与 4 号；(2) 5 号与 14 号；(3)0 号与 3 号；(4)11 号与 15 号；(5)7 号与（5 号、14 号）；(6)1 号与（0 号、3 号）；(7)（2 号、4 号）与（1 号、0 号、3 号）；(8)6 号与（7 号、5 号、14 号）；(9)12 号与 13 号；(10)8 号与 16 号；(11)10 号与（11 号、15 号）；(12)（8 号、16 号）与（12 号、13 号）；(13)（8 号、16 号、12 号、13 号）与（2 号、4 号、1 号、0 号、3 号）；(14)（6 号、7 号、5 号、14 号）与（8 号、16 号、12 号、13 号、2 号、4 号、1 号、0 号、3 号）；(15) 9 号与（10 号、11 号、15 号）

最终两类分别包含样本：

1. （6 号、7 号、5 号、14 号、8 号、16 号、12 号、13 号、2 号、4 号、1 号、0 号、3 号）　2. （9 号、10 号、11 号、15 号）

Ncut:

(1)8 号与 9 号；(2)10 号与 14 号；(3)3 号与 6 号；(4)1 号与 11 号；(5)5 号与 15 号；(6)16 号与（10 号、14 号)；(7)4 号与（8 号、9 号)；(8)0 号与 13 号；(9)12 号与（1 号、11 号)；(10)2 号与（16 号、10 号、14 号)；(11)7 号与（0 号、13 号)；(12)（3 号、6 号）与（5 号、15 号)；(13)（4 号、8 号、9 号）与（12 号、1 号、11 号)；(14)（7 号、0 号、13 号）与（3 号、6 号、5 号、15 号)；(15) （2 号、16 号、10 号、14 号）与（4 号、8 号、9 号、12 号、1 号、11 号)

最终两类分别包含样本:

1. （2 号、16 号、10 号、14 号、4 号、8 号、9 号、12 号、1 号、11 号)

2. （7 号、0 号、13 号、3 号、6 号、5 号、15 号)

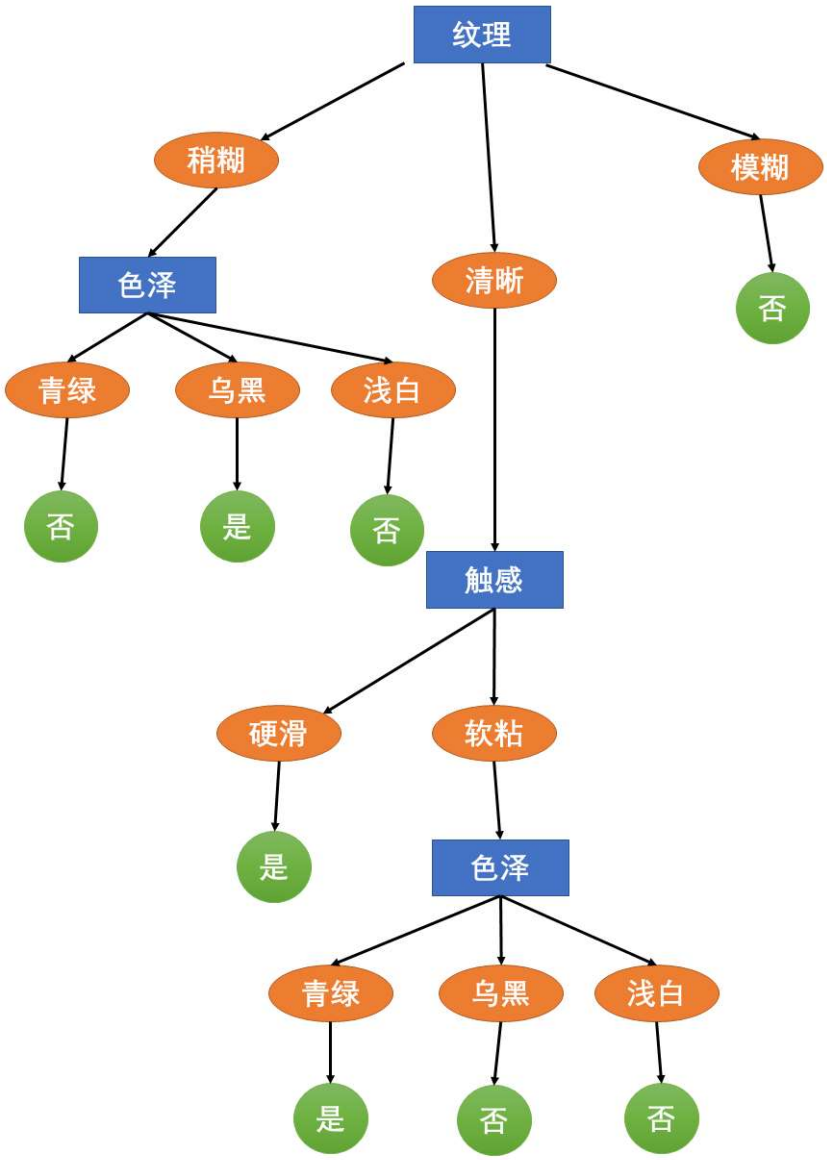
T3：附加题：实现基于信息增益率的决策树，并对西瓜 3.0 数据集进 行 70%训练-30%测试

答：编程思路：①读取数据，打乱顺序后划分训练集与测试集；②编写计算信息熵的程序；③编写计算某特征的信息增益程序，其中会调用②的程序；④对比每一个特征的信息增益，由此选出最优的特征；⑤递归函数建造决策树，用字典保存树结构；⑥在测试集上检测精度。

最终由训练集生成的决策树字典结构如下:

```
the DTree is:
{'纹理': {'稍糊': {'色泽': {'青绿': '否', '乌黑': '是',
'浅白': '否'}}, '清晰': {'触感': {'硬滑': '是', '软粘':
{'色泽': {'青绿': '是', '乌黑': '否', '浅白': '否'}}}},
'模糊': '否'}}
```

画出决策树如下:



最终在测试集上的正确率为 66.7%

理论题简答

(1)试把 k 均值的目标函数进行变换，使得表达式中每项只包含$\mathbf{x}^{\mathrm{T}}\mathbf{x}$形式；

(2) 如果 k 均值聚类过程中出现了错误提示某个类是空集，试画图例分析什么情况下会造成这种错误以及对应的改正方法；

（1）类别划分步骤：

$$j = \arg min \|\mathbf{x} - \boldsymbol{\mu}_{\mathbf{i}}\|^2 = \arg min (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{i}}) \cdot (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{i}})^{\mathbf{T}}$$

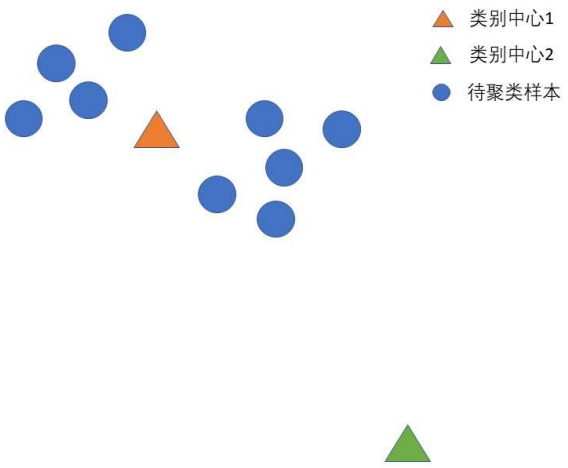
$$j = \arg min \left(\mathbf{x}\mathbf{x}^{\mathbf{T}} - \boldsymbol{\mu}_{\mathbf{i}}\mathbf{x}^{\mathbf{T}} - \mathbf{x}\boldsymbol{\mu}_{\mathbf{i}}^{\mathbf{T}} + \boldsymbol{\mu}_{\mathbf{i}}\boldsymbol{\mu}_{\mathbf{i}}^{\mathbf{T}}\right)$$

$$= \arg min \left(\mathbf{x}\mathbf{x}^{\mathbf{T}} - 2\boldsymbol{\mu}_{\mathbf{i}}\mathbf{x}^{\mathbf{T}} + \boldsymbol{\mu}_{\mathbf{i}}\boldsymbol{\mu}_{\mathbf{i}}^{\mathbf{T}}\right)$$

计算均值步骤：

$$\boldsymbol{\mu}_{\mathbf{i}} = \frac{1}{|\mathrm{C}_i|} \sum_{x_j \in \mathrm{C}_i} \mathbf{x}_j$$

（2）出现空集的情况：如果选取某一点为聚类中心时，出现没有样本归属到这一类时，会造成出现某一类为空集的情况，此时没有办法进行聚类类别中心位置的更新。该情况如下图所示：



从图中可以看到，用 Kmeans 聚类可以分为两类。但由于初始类别中心选取不恰当，导致所有的样本都会被分为橙色类别中心的类别，也就是说绿色的中心代表的类别为空集，这样就没办法进行类别中心的更新了。在更高维度的聚类问题中，在聚类过程中也有可能会出现某一类别中心没有样本较为接近的情况，这也会导致出现某类别为空集。

该问题的改正办法为：（1）在选取初始聚类中心时，采用 k 个样本的所在位置为起始类别中心。（2）在聚类过程中若出现空集，则选择在点最多的一簇类别中，选择离该类别中心最远的样本所在位置作为空集类别的类别中心。