Programming 1：

ISOMAP：代码见 isomap.py

生成的 Z 形状流形如图 1 所示，共有 2000 个数据点。

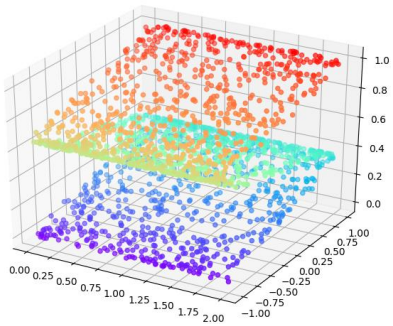
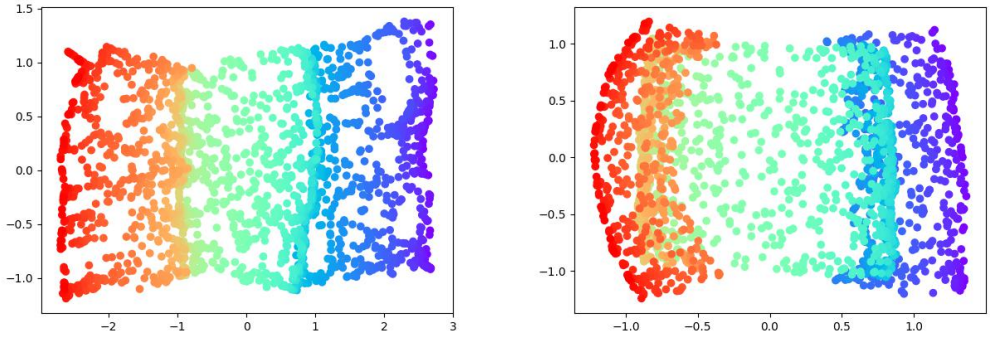


图 1：三维空间中 Z 形状数据

在 isomap 编程中遇到了一些问题。采用 floyd 求图的最短路径，但是返回的距离矩阵仍有 inf 值，即某些点更新完之后仍不可达。可能是 k 过小使得在 k 近邻连接后，仍有一些数据点孤立，不是任何一个点的近邻。将 k 改大后即解决这一问题。另外 floyd 三层循环，时间复杂度较高。2000 个数据点 k=10，单核需要跑一个多小时。

得到的结果如下图 2 所示，k=10 比较理想。



k=10　　k=30

图 2：isomap 变换结果图

LLE：具体代码见 LLE.py

生成的 3 形状流形如图 3 所示，共有 2000 个数据点。

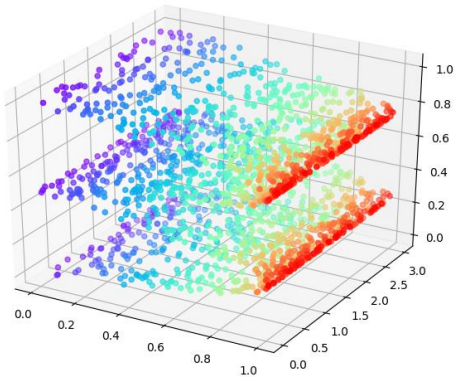
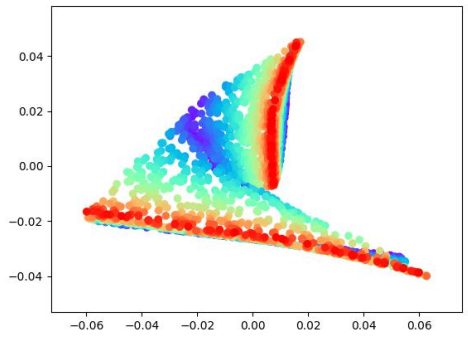
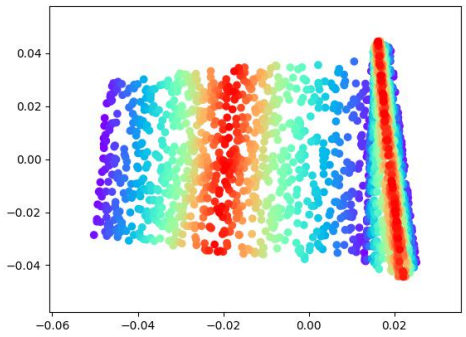


图 3：三维空间中 3 形状数据

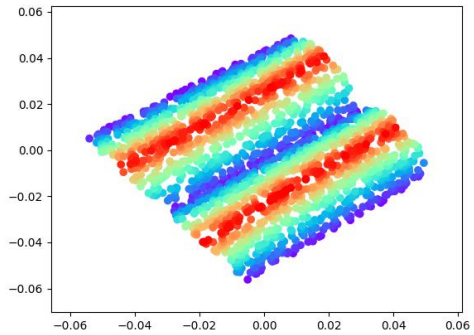
得到的部分结果如下图 4 所示，可以看到均为一个流形，k=40 效果已经很不错了，k=80 时相比 k=40，数据点更分散了。



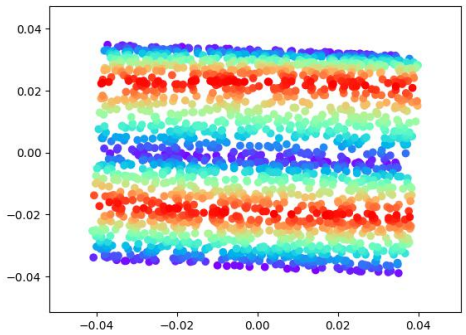
k=10



k=20



k=40



k=80

图 4：LLE 结果图

Programming 2：DecisionTree

Sogou corpus 数据集包含 14400 个文档的特征矩阵 wordMat（14400*1200），共有 1200 个关键词特征，0-1 表示。所有文档被分成 9 类，用 0-9 表示，记录在 doclabel 里。

用 train_test_split 将训练集和测试集以 4：1 分割，再从训练集中以 3：1 分割出交叉验证集。

由于错误估计作业量，没有完成代码的调试工作，具体代码见 decisiontree.py，下面简单说明各个函数的思路。

GenerateTree(thresh):

用递归方法创建决策树。

划分节点 SplitNode(samplesUnderThisNode, thresh)

根据最优特征的标签生成节点，删除已经使用的特征标签。

遍历特征，生成树。

SplitNode(samplesUnderThisNode, thresh)：#对当前节点进行分支

按照 SelectFeature 得到的待分特征 index，对节点进行二分支，返回左、右节点。

停止分支的条件：所有的候选分支的不纯度下降量都小于 thresh，则停止分支。已经达

到最小节点数，或者最大深度等。

SelectFeature(samplesUnderThisNode)：#选择待分特征

遍历特征，选择 Impurity 信息增益最大的那个特征做查询，返回 index。

Impurity(samples): #计算不纯度

$$i(N) = -\sum_j P(w_j) \log_2 P(w_j)$$

采用熵不纯度来表示，，进一步用信息增益。

Decision(GeneratedTree, XToBePredicted)：

将测试数据输入生成的决策树 featMat，返回分类结果 label。

Prune(GeneratedTree, CrossValidationDataset)：#后剪枝

对已经生成好的树在验证集上进行剪枝。遍历所有相邻的叶子节点，如果将他们消去可以增加验证集上的正确率，则减去两叶子节点，将他们的共同祖先作为新的叶子节点。返回剪枝后的树。

main(): #主函数

```
loaddata() # import data
for thresh in [a1,a2,a3,a4]: #测试不同的超参数，选择正确率最高的那个模型
    myTree = GenerateTree(thresh)
    PrunedTree = Prune(myTree, CrossValidationDataset)
    predicted_label = Decision(myTree, XToBePredicted)
    accuracy = np.sum(testlabel == predicted_label) / len(testlabel)
```

用 sklearn 中 decisionTree 包进行结果的讨论，具体代码见 sklearn_decisionTree.py
对几个重要参数进行讨论：

1. criterion：特征选择标准，默认为 gini，正确率 73.5%；若为 entropy，则正确率为 71.4%。二者相近。

2. Splitter: 特征划分点选取标准，默认为 best，正确率为 73.5%；若为 random，则正确率为 73.2%，二者相近。

3. max_features：划分时考虑的最大特征数，可选参数，默认是 None，主要影响决策树的生成时间。若为 1200，正确率为 73.5%，默认是全部使用；若为 sqrt(n_features)，正确率为 59.4%，下降很多。

4. max_depth：决策树最大深度，可选参数，默认是 None。是某种意义上的剪枝操作。

max_depth	3	10	50	100	1000
正确率	32.0%	57.7%	71.9%	72.2%	72.3%

5. min_impurity_split：节点划分最小不纯度,可选参数，默认是 1e-7。这个阈值限制了决策树的增长，如果某节点的不纯度(基尼系数，信息增益，均方差，绝对差)小于这个阈值，则该节点不再生成子节点。在 0.1 及以下，正确率基本上差不多，0.1 正确率最高。

min_impurity_split	0.0001	0.001	0.01	0.1	1
正确率	73.4%	73.3%	72.8%	75%	11%