Lab 7 实验题目

学 号	陈政培	学 院	专 业
姓 名	17363011	智能工程学院	智能科学与技术

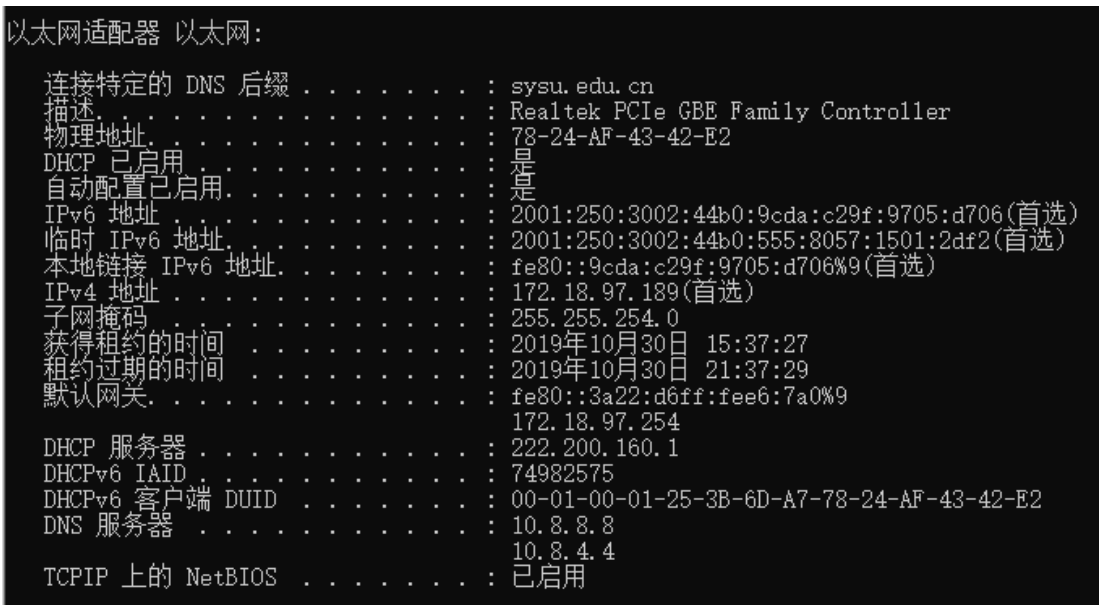
1、实验目的： **k-means 应用实践**

2、实验环境：vs code、python3.6.8、百度 Web API

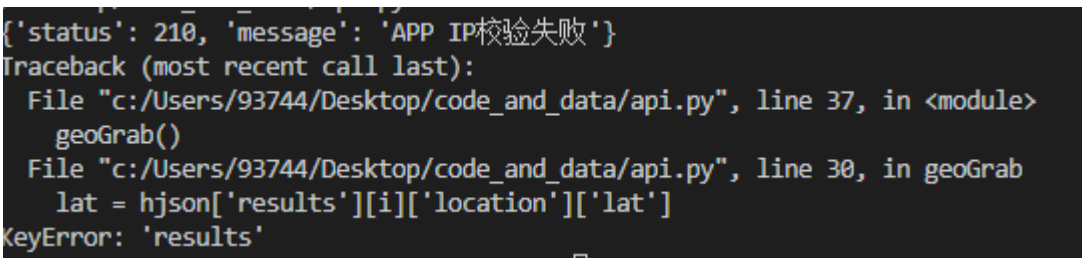
3、实验步骤：

① 获取地图数据

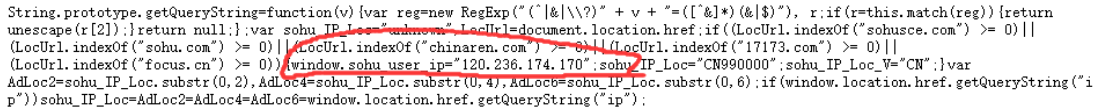
在百度地图开放平台中创建应用时填写访问方的公网 IP，一开始在 cmd 中输入 ipconfig /all 获取本机 ip 地址



误以为在 IP 白名单处填写对应的网关 IP 就可以，结果出现报错



发现 APP IP 校验失败，后来发现原来此处 cmd 获取的 ip 仍然是对于学校内网的 ip。为了获取公网 ip，我打开了 http://txt.go.sohu.com/ip/soip 以检查我的实际访问公网 ip



原来我的实际公网 ip 是中山大学的 120.236.174.170

② 将应用 AK 放到 python 代码中，成功获取了 Restaurant_Data_Beijing.txt

③ 测试 loadDataSet()、randCent()和 distEclud()函数

④ kMeans 算法实现

在未修改 kmeans.py 文件前，可以直接显示测试结果，但是当把语句“from numpy import *”修改成“import numpy as np”后代码出现了报错

```
Traceback (most recent call last):
  File "c:/Users/93744/Desktop/code_and_data/main.py", line 14, in <module>
    myCentroids, clustAssing = kmeans.kMeans(dataset, 4)
  File "c:\Users\93744\Desktop\code and data\kmeans.py", line 86, in kMeans
    if distJI < minDist:
ValueError: The truth value of an array with more than one element is ambiguous.
 Use a.any() or a.all()
```

本以为是语句

```
if distJI < minDist:
```

是数据结构之间的比较出现了问题，后来才发现 python 中除了 numpy 库中有一个 sum 函数，python 本身也有一个 sum 函数。但是两者实现的功能和输出结果并不相同，所以出现了数据结构的错误

```
    return np.sqrt(np.sum(np.power(vecA - vecB, 2)))
```

需要将 kmeans.py 文件 19 行的代码中 sum 改成 np.sum 即可正常运行

⑤ 用 kMeans 算法对地图上的点聚类

原理和之前使用 testSet 一样

4、实验结果与分析：

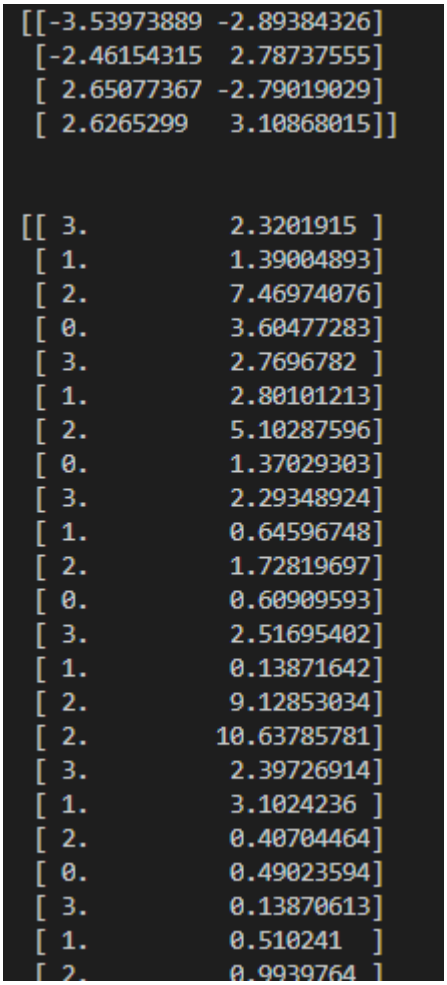
① 从文本文件构建矩阵并建立辅助函数

```
[[-5.379713]]
[[-4.232586]]
[[5.1904]]
[[4.838138]]
[[ 3.5067209  -0.69521633]
 [ 0.29318168  3.87091094]]
5.184632816681332
```

输出结果分别为

```
print(min(dataset[:,0]))
print(min(dataset[:,1]))
print(max(dataset[:,1]))
print(max(dataset[:,0]))
print(kmeans.randCent(dataset, 2))
print(kmeans.distEclud(dataset[0], dataset[1]))
```

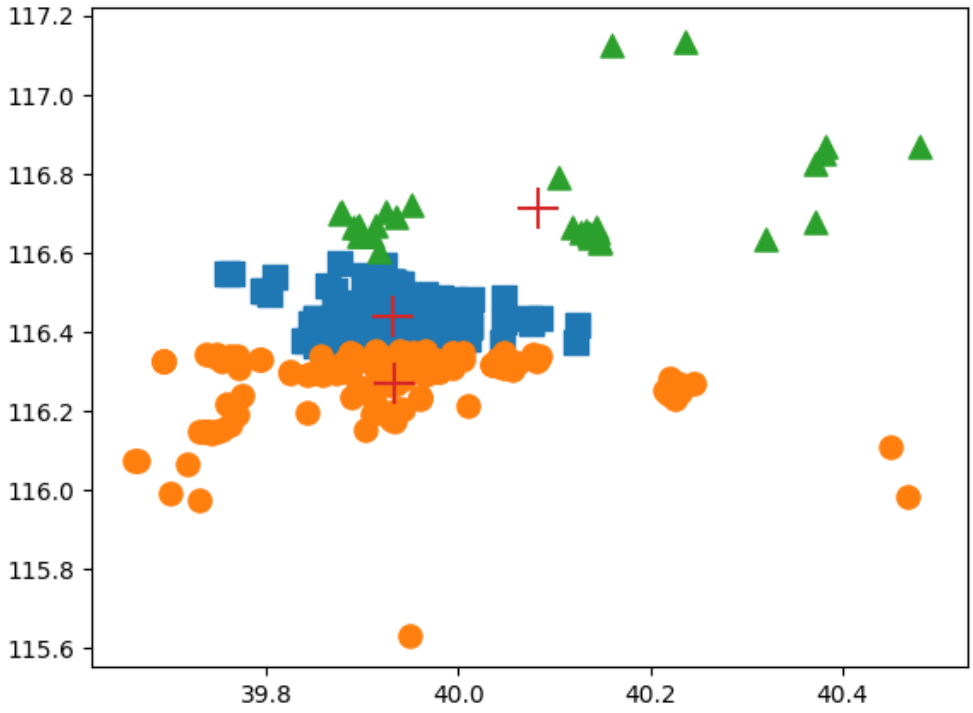
② kMeans 算法实现



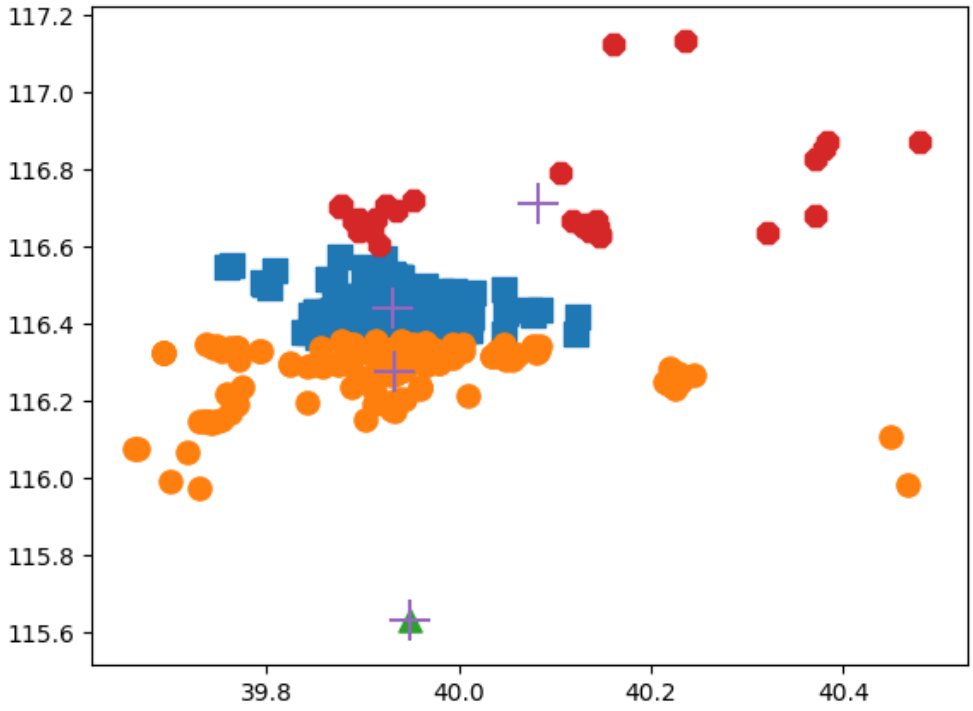
得到了 4 个质心的坐标和各个簇的分配结果（结果截图了一小部分）

③ 用 kMeans 算法对地图上的点聚类

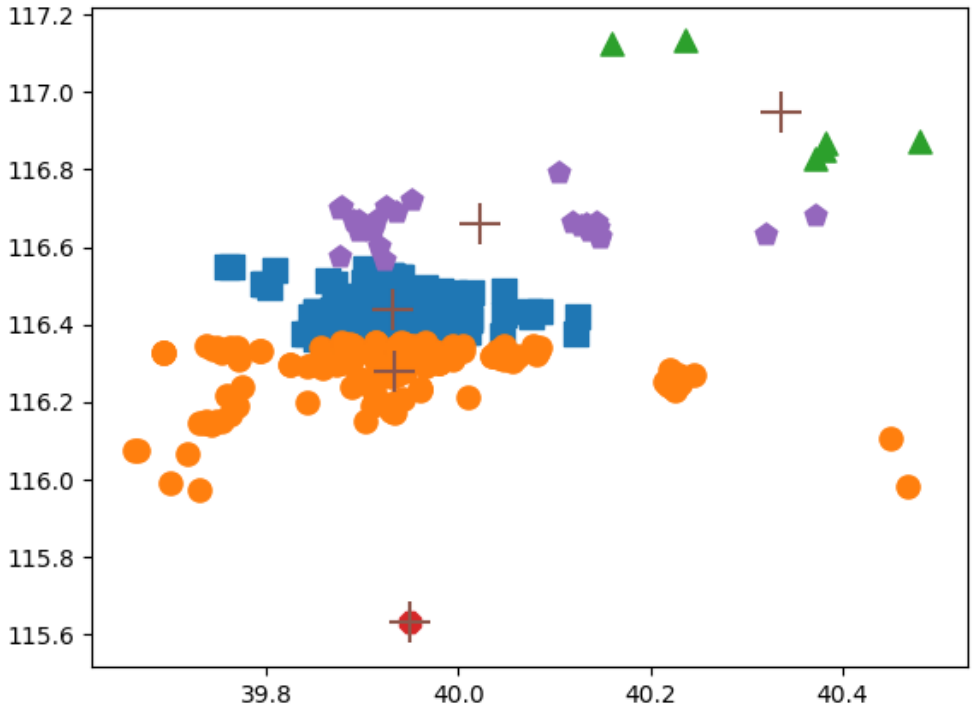
Kmeans.clusterPlaces(3)



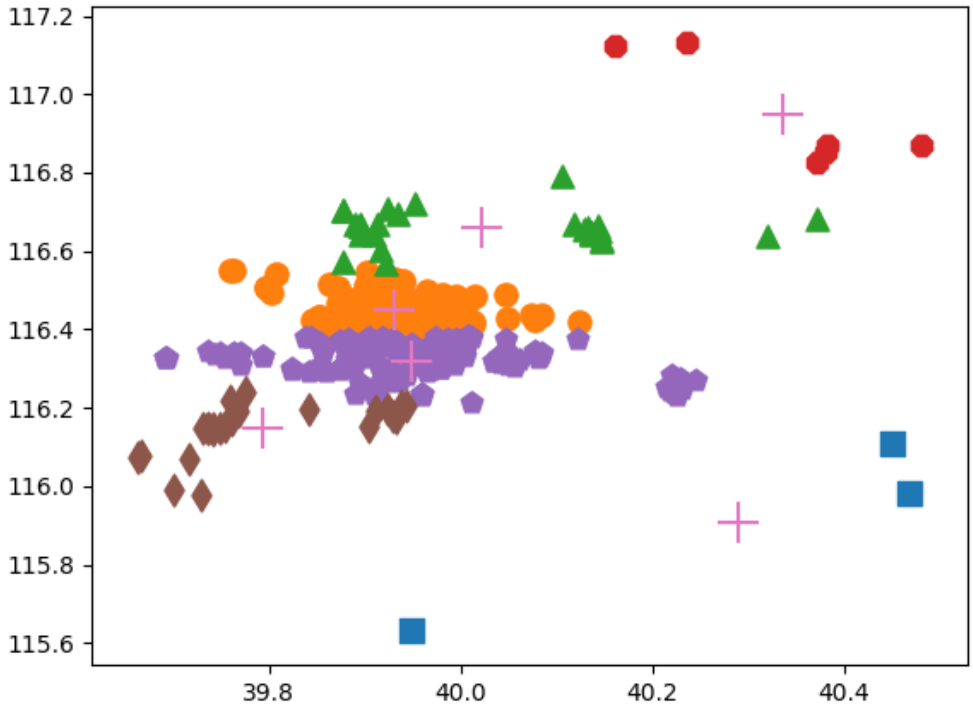
Kmeans.clusterPlaces(4)



Kmeans.clusterPlaces(5)



Kmeans.clusterPlaces(6)



5、作业：

① 对聚类结果可视化（包含样本点和簇中心，用不同颜色、记号标记）

4 结果展示中已经实现了

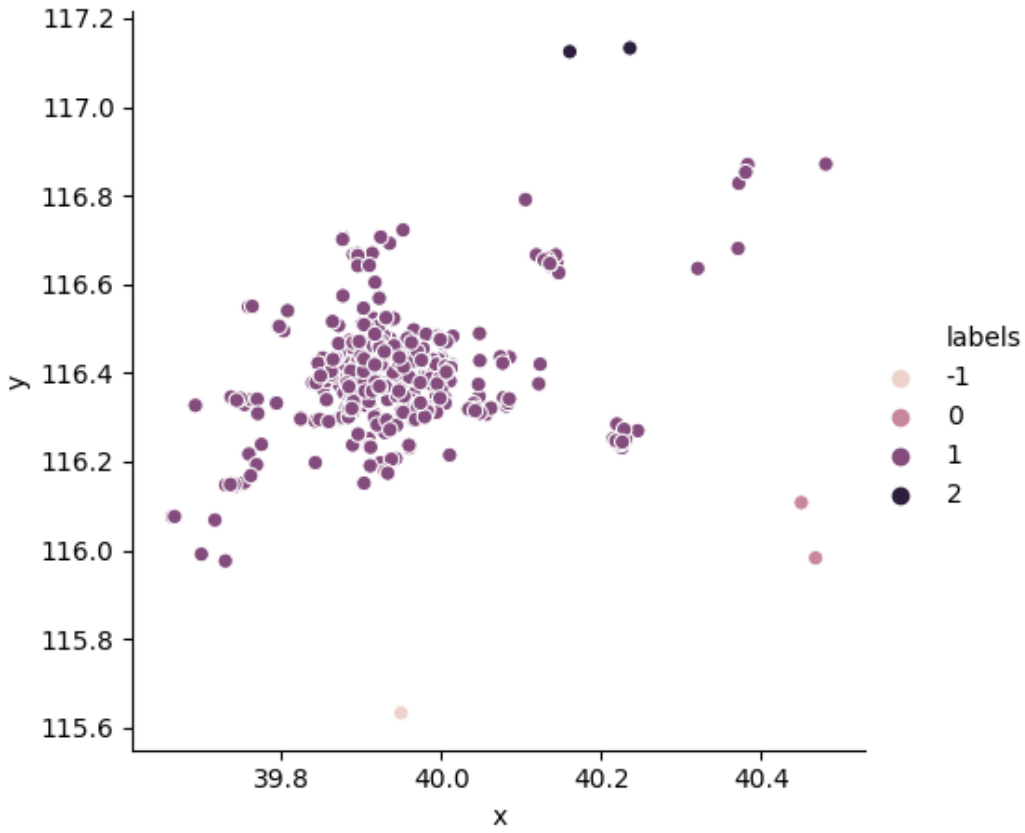
② Kmeans.py 中的语句代替并修改代码使其正常执行

修改结果见代码，遇到的问题和 bug 修正过程详见 3.④中描述

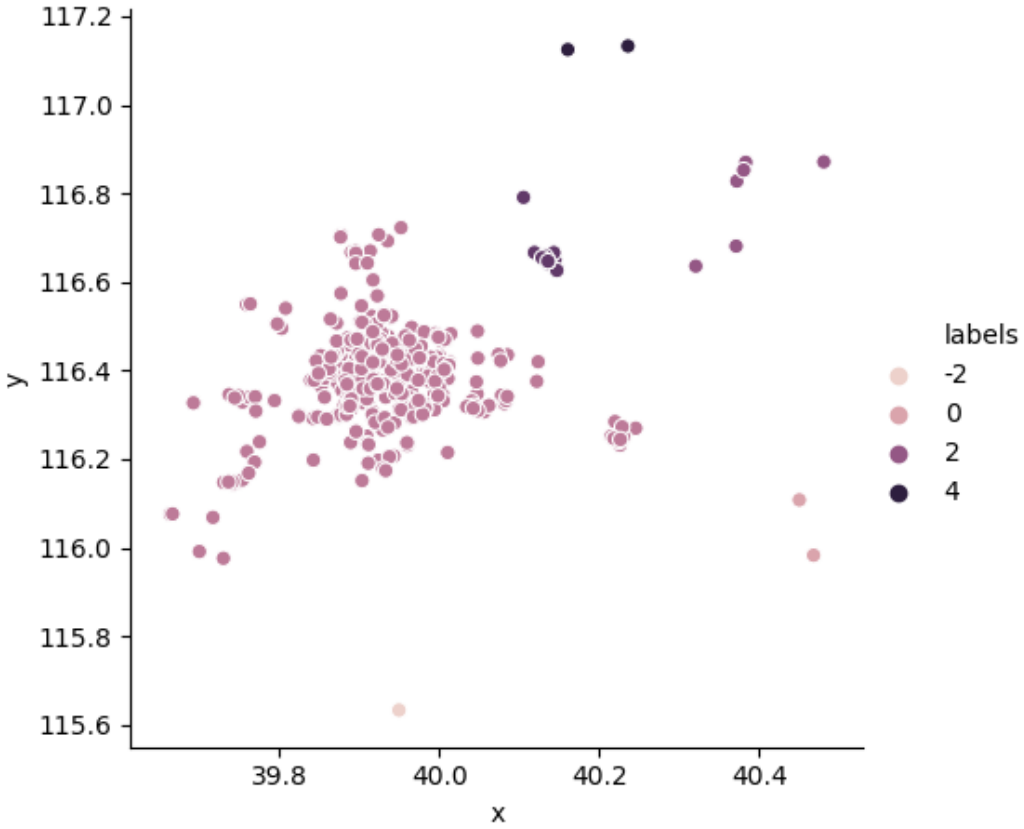
③ 尝试用 DBSCAN 聚类该数据集

使用了 sklearn 库中的 cluster 里的函数 KMeans 和 DBSCAN 进行聚类

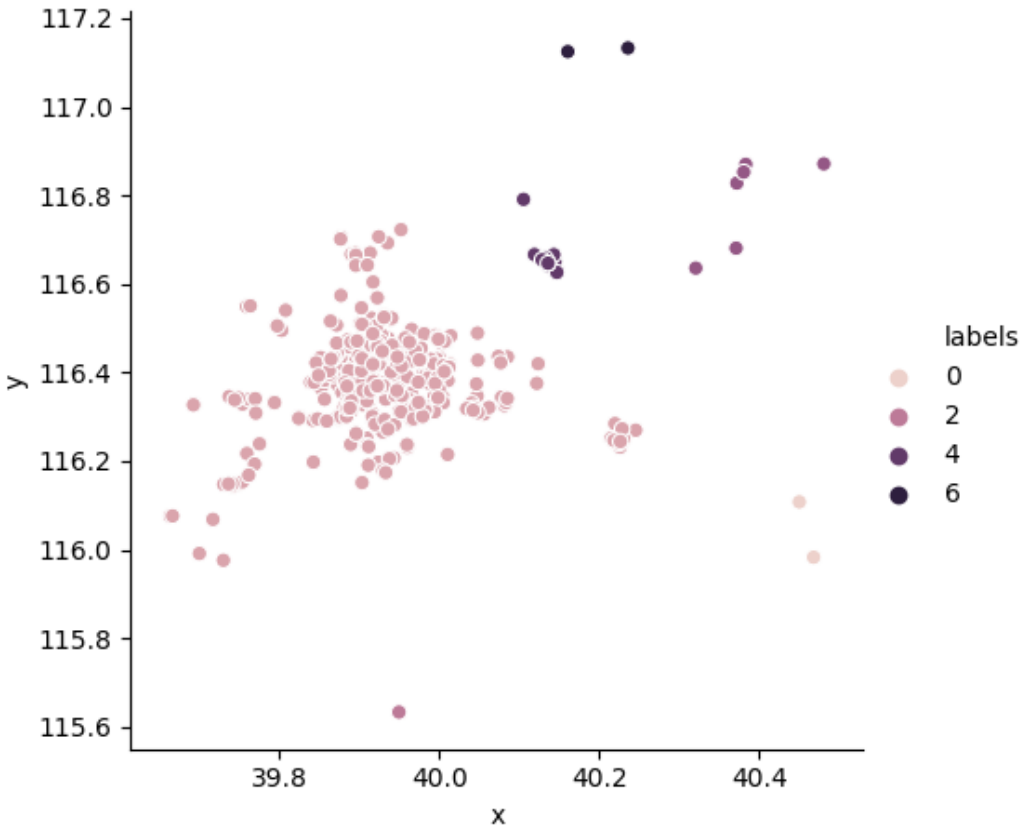
- 因为 DBSCAN 聚类函数是基于密度的聚类函数，所以不需要预先指定簇的个数，最终的簇的个数不确定。分类的依据是聚类的半径 eps 和邻域内最小点数目 min_samples
- 想要获得和 KMeans 相似的聚类效果则需要调试 eps 和 min_samples 的大小，以得到合适的簇数目
- Eps = 0.2， min_samples = 2 时获得 3 个簇



- Eps = 0.15， min_samples = 1.5 时获 5 个簇



- Eps = 0.15， min_samples = 1 时获得 6 个簇



④ 利用轮廓系数评估 kMeans 和 DBSCAN 对该数据集的聚类效果

通过 sklearn 中 metrics 里的函数 silhouette_score 来对两种聚类算法的结果进行轮廓系数评估

- DBSCAN 函数在实际计算时把一部分餐厅作为噪声屏蔽了
- 簇为 3 时

```
KMeans:
噪声比: 0.00%
分簇的数目: 3
轮廓系数: 0.863
DBSCAN:
噪声比: 0.25%
分簇的数目: 3
轮廓系数: 0.833
```

- 簇为 5 时

```
KMeans:
噪声比: 0.00%
分簇的数目: 5
轮廓系数: 0.893
DBSCAN:
噪声比: 0.25%
分簇的数目: 5
轮廓系数: 0.855
```

- 簇为 6 时

```
KMeans:
噪声比: 0.00%
分簇的数目: 6
轮廓系数: 0.904
DBSCAN:
噪声比: 0.00%
分簇的数目: 6
轮廓系数: 0.858
```

- 整体结论为在轮廓系数上 DBSCAN 相对更优，但是有一定噪声损失

⑤ 尝试在 Tensorflow 环境下实现 kMeans

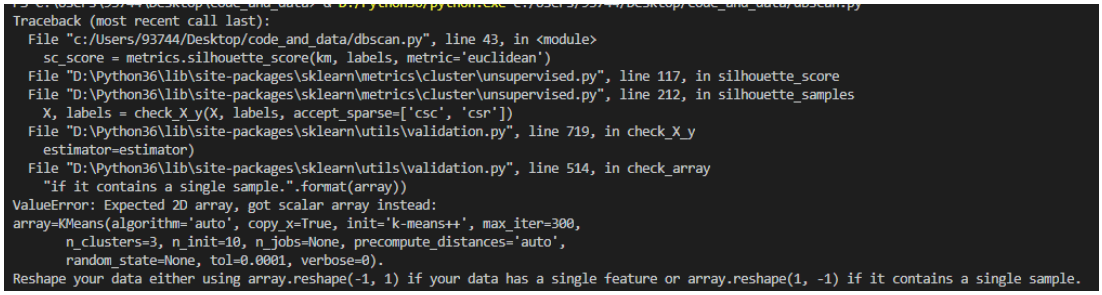
详见代码

```
the first cluster division is :  [1 1 2 1 1 1 1 1 1 1 1 1 0 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1
 2 2 2 2 2 1 2 1 2 1 2 2 2 2 1 2 0 1 1 1 1 1 2 2 2 0 2 1 2 1 1 1 1 1 2 1 1
 1 1 2 1 2 1 1 1 2 2 0 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 2 0 1 2 2 1 1 1
 2 2 1 2 2 2 1 1 2 2 2 2 2 1 1 2 1 2 1 0 1 1 2 2 2 2 2 2 1 2 1 2 1 1 1 2 2 2
 1 2 2 0 1 2 1 2 0 0 2 1 1 1 1 2 1 0 1 2 1 1 1 1 1 1 1 2 1 2 1 1 1 1 2 0 2
 2 1 1 2 1 2 0 1 2 1 2 0 2 2 2 2 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1
 1 2 2 1 1 1 0 1 1 1 2 1 2 1 2 2 1 1 2 2 1 1 0 1 2 1 1 0 1 1 1 1 1 1 1 1 2
 2 2 1 1 1 2 1 0 1 2 1 2 1 2 1 2 1 2 1 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 2 1
 1 1 2 1 2 2 2 1 1 1 1 2 1 0 1 1 0 1 1 1 2 2 2 1 2 1 2 1 1 1 1 1 2 1 1 1 1
 1 1 1 2 1 0 2 1 2 2 2 1 1 1 1 2 1 1 2 2 1 1 2 1 0 1 1 1 1 2 1 0 2 1 0 1 2 2
 1 2 1 0 1 2 1 1 2 1 1 1 1 2 2 1 0 1 1 2 1 1 1 1 2 1 1 1 1 0]
```

可以实现聚类结果，但是由于时间原因可视化未进行颜色分类

6、实验总结：

- APP IP 校验失败，后来发现原来此处 cmd 获取的 ip 仍然是对于学校内网的 ip。为了获取公网 ip，我打开了 http://txt.go.sohu.com/ip/soip 以检查我的实际访问公网 ip
- python 中除了 numpy 库中有一个 sum 函数，python 本身也有一个 sum 函数。但是两者实现的功能和输出结果并不相同，所以出现了数据结构的错误
- 调试 DBSCAN 代码时报错



原因是 silhouette_score 函数接受的参数中需要的 list 类型是（1，-1）的，而我们传进去的数据 shape 是错误的，对相应数据进行 reshape 即可

- 在网上找到的教程使用的 tensorflow 版本是 1.0，所以很多 tf 函数都不能直接使用，必须通过 tf.compat.v1.来访问老版本的函数

7、参考文献：

① https://www.jianshu.com/p/82db047eea13

② https://blog.csdn.net/u012967763/article/details/79149703

③ https://github.com/aliceleee/KMeans_Tensorflow/tree/master/code