编程作业说明：聚类

任务 1：对地理数据应用二分 k-均值算法聚类

问题：你的朋友 Drew 希望你带他去城里庆祝他的生日。由于其他一些朋友也会过来，所以需要你提供一个大家都可行的计划。Drew 给了你希望去的 69 个地址和相应的经纬度。你要决定将这些地方进行聚类的最佳策略，这样可以安排交通工具抵达这些簇的质心，然后步行到每个簇内地址。

准备数据：用 Python 解析文本文件，根据经纬度信息计算球面距离。

分析数据：使用 Matplotlib 构建一个二维数据图，其中包含簇与地图

聚类算法：应用二分 k-均值算法，最后的输出是包含簇及簇中心的地图。

以下为部分数据集，仅提取最后两列即可：

```
Dolphin II  10860 SW Beaverton-Hillsdale Hwy    Beaverton, OR   45.486502   -122.788346
Hotties 10140 SW Canyon Rd. Beaverton, OR   45.493150   -122.781021
Pussycats   8666a SW Canyon Road    Beaverton, OR   45.498187   -122.766147
Stars Cabaret   4570 Lombard Ave    Beaverton, OR   45.485943   -122.800311
Sunset Strip    10205 SW Park Way   Beaverton, OR   45.508203   -122.781853
Vegas VIP Room  10018 SW Canyon Rd  Beaverton, OR   45.493398   -122.779628
Full Moon Bar and Grill 28014 Southeast Wally Road  Boring, OR  45.430319   -122.376304
505 Club    505 Burnside Rd Gresham, OR 45.507621   -122.425553
Dolphin 17180 McLoughlin Blvd   Milwaukie, OR   45.399070   -122.618893
Dolphin III 13305 SE McLoughlin BLVD    Milwaukie, OR   45.427072   -122.634159
Acropolis   8325 McLoughlin Blvd    Portland, OR    45.462173   -122.638846
Blush   5145 SE McLoughlin Blvd Portland, OR    45.485396   -122.646587
```

二分 k-均值算法思想：

为克服 K-均值算法收敛于局部最小值的问题，因此有人提出二分 k-均值算法，该算法首先将所有点作为一个簇，然后将该簇一分为二。之后选择其中一个簇继续进行划分，选择哪一个簇进行划分取决于对其划分是否可以最大程度降低误差平方和(SSE)的值。上述基于 SSE 的划分过程不断重复，直到得到用户指定的簇数目为止。

二分 k-均值算法伪代码：

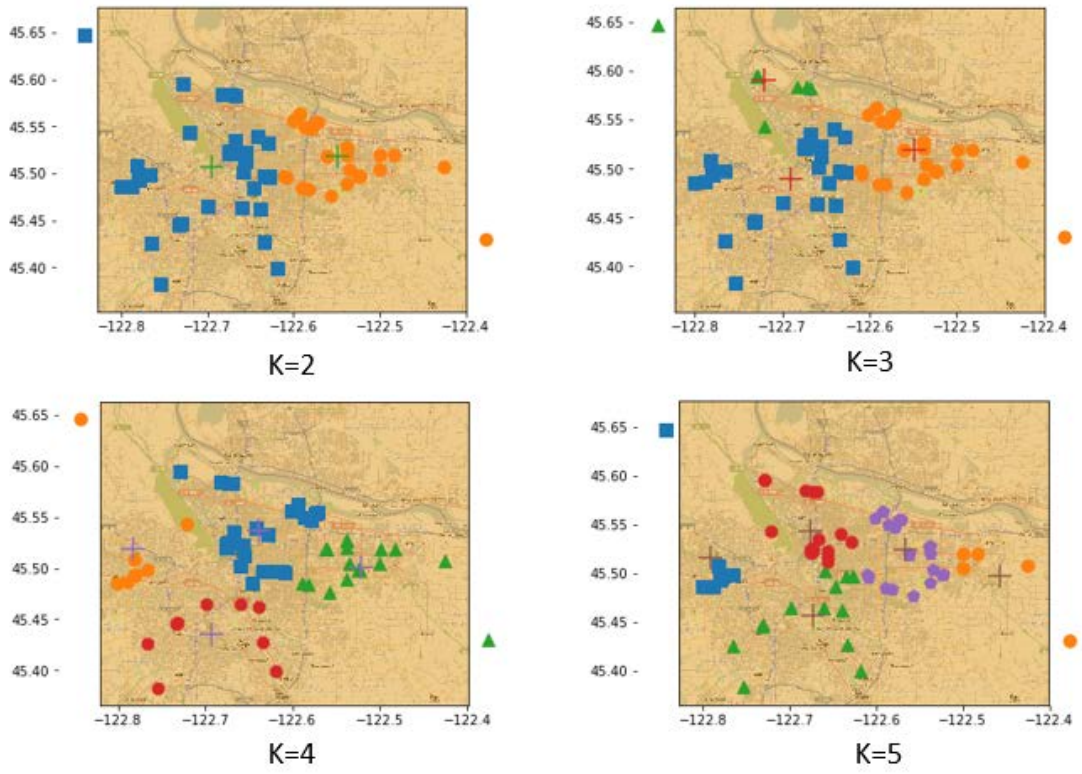
```
将所有点看成一个簇
当簇数目小于 k 时
对于每一个簇
    计算总误差
    在给定的簇上面进行 k-均值聚类（k=2）
```

计算将该簇一分为二之后的总误差

选择使得误差最小的那个簇进行划分操作

实验结果可视化，输出实验结果如下所示：

（由于簇中心具有随机性，因此每次运行结果可能不同）



本个任务给出了 demo 代码给予大家进行启发，希望大家在 demo 代码的基础上，能够加入自己对于 k-means 聚类算法的理解，同时能够自己动手进行编写尝试。

所需提交材料：任务 1 需要完成聚类代码编写，输出包含簇及簇中心的地图，并将其展示在实验报告中，编写实验报告需要简述实验代码编写思路、实验结果、算法流程等。

任务 2：根据用户采集的 WiFi 信息对用户进行聚类

（1）数据集讲解

数据集：数据集存于 DataSetKMeans1.csv 与 DataSetKMeans2.csv 中，两个数据集相互独立。

BSSIDLabel:：SSID 标识符，每个 AP（接入点，如路由器）拥有 1 个或多

个不同的 BSSID，但 1 个 BSSID 只属于 1 个 AP；

RSSLabel：该 BSSID 的信号强度，单位 dbm；

RoomLabel: 该 BSSID 被采集时所属的房间号；

SSIDLabel: 该 BSSID 的名称，不唯一；

finLabel：finLabel 标号相同，表示这部分 BSSID 在同一时刻被采集到；我们将在同一时刻采集的所有 BSSID 及其相应 RSS 构成的矢量称为一个指纹$f_i = [BSSID_1: RSS_1, BSSID_2: RSS_2, \dots]$；由于 BSSID 的 RSS 在不同位置大小不同，因此指纹可以唯一的标识一个位置。

BSSIDLabel	RSSLabel	RoomLabel	SSIDLabel	finLabel
06:69:6c:0a:bf:02	-56	1	HUST_WIRELESS	1
20:76:93:3a:ae:78	-69	1	HC5761	1
4a:69:6c:07:a1:e7	-69	1	HUST_WIRELESS_AUTO	1
0e:69:6c:0a:bf:02	-63	1	HUST_WIRELESS_AUTO	2
4a:69:6c:07:a1:e7	-66	1	HUST_WIRELESS_AUTO	2
2a:69:6c:05:c5:25	-67	1	HUST_WIRELESS_AUTO	2
08:57:00:7b:63:16	-72	1	TL-WTR9500	2

（2）注意

样本构造：一个指纹$f_i = [BSSID_1: RSS_1, BSSID_2: RSS_2, \dots]$为一个样本，共包含 N 个样本；有可能$BSSID_2$未在该样本中出现，但属于特征构造中得到的一个特征，则 RSS 值采用缺失值处理方法进行处理。

特征构造：不同样本中 BSSID 集合不尽相同，因此可以采用所有样本 BSSID 集合的并集作为特征，如指纹f_i的 BSSID 集合为$B_i = \{BSSID_j | BSSID_j \in f_i\}$，则特征可表示为$B_u = \bigcup_{i=1}^{N} B_i$。

缺失值处理：使用特殊值填充，如 0，-100 等。

样本间距离计算：可以采用欧式距离；可以仅计算两个样本中同一特征均为接收到的 RSS 值的平均距离。

举例说明：

$$f_1 = [BSSID_1: -30, BSSID_2: 0, BSSID_3: -45, BSSID_4: -70]$$

$$f_2 = [BSSID_1: -40, BSSID_2: -80, BSSID_3: -35, BSSID_4: 0]$$

则f_1本身只接收到$BSSID_1$、$BSSID_3$、$BSSID_4$共三个 BSSID；f_2本身只接收到$BSSID_1$、$BSSID_2$、$BSSID_3$共三个 BSSID；特征为所有样本 BSSID 的并集$B_U = \{BSSID_1, BSSID_2, BSSID_3, BSSID_4\}$，缺失值用 0 填充。

（上述只是提供一种思路，采用其它方法构造均可以）

（3）实验要求：

编写代码分别对 DataSetMeans1.csv 和 DataSetMeans2.csv 两个数据集完成聚类实验，k（k>=2）取不同的值，评估聚类的内部指标 DB 指数，DB 指数定义如下：

DB 指数（Davies-Bouldin Index，简称 DBI）

$$\mathrm{DBI} = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(\mu_i,\mu_j)}\right)$$

其中，μ代表簇 C 的中心点，avg(C)对应于簇 C 内样本间的平均距离，$d_{cen}(\mu_i,\mu_j)$对应于簇C_i和C_j中心点间的距离。显然，DBI 的值越小越好。

附加：可采用 MDS 对 BSSID 进行降维，可视化聚类。

需要编写实验报告叙述数据处理过程、算法实现思路、实验结果展示等。可以将实验结果截图展示在实验报告中。

（希望同学们得出理想的实验结果，学习到更多的知识！）