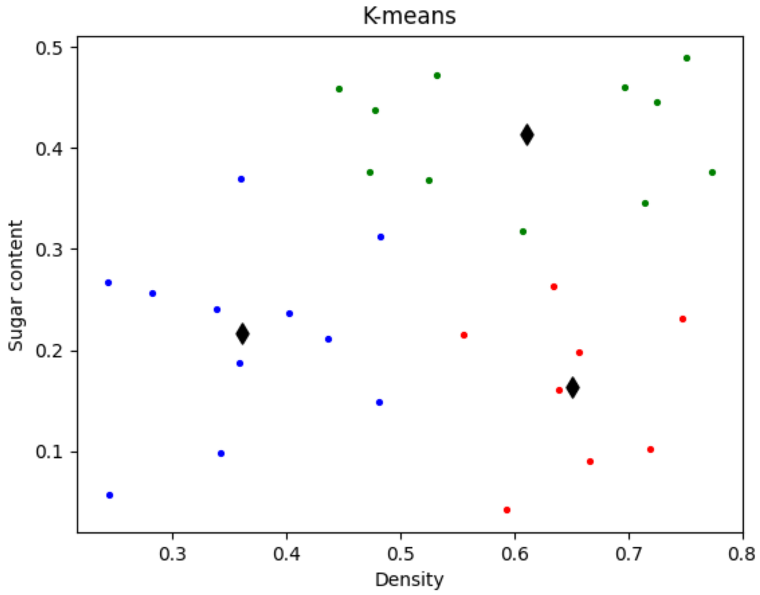
机器学习作业——聚类

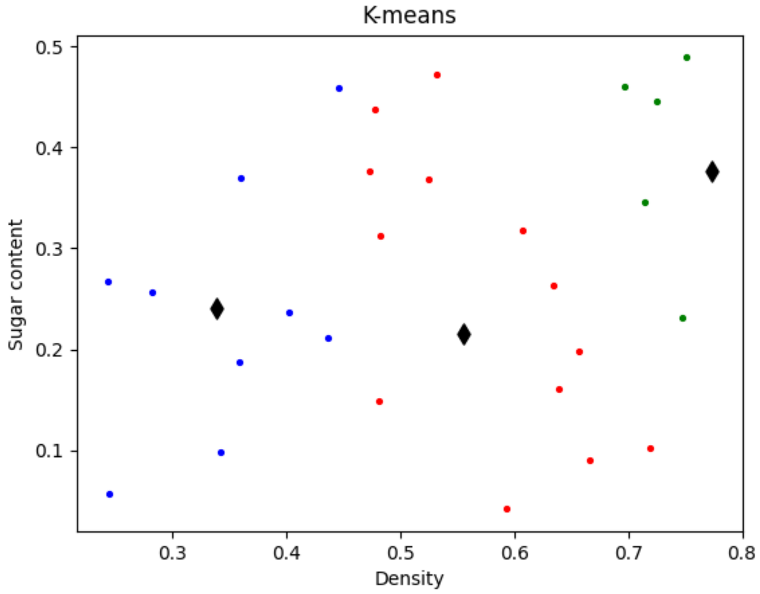
161910126 赵安

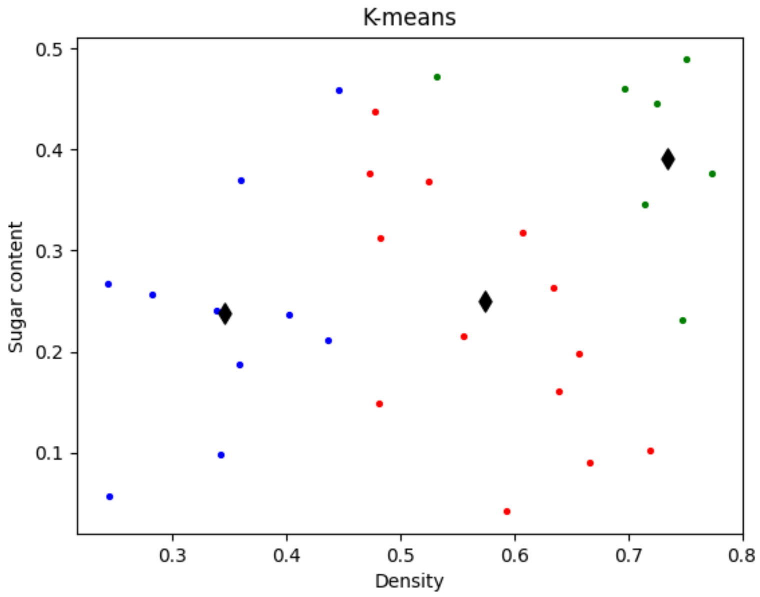
1、K均值算法：

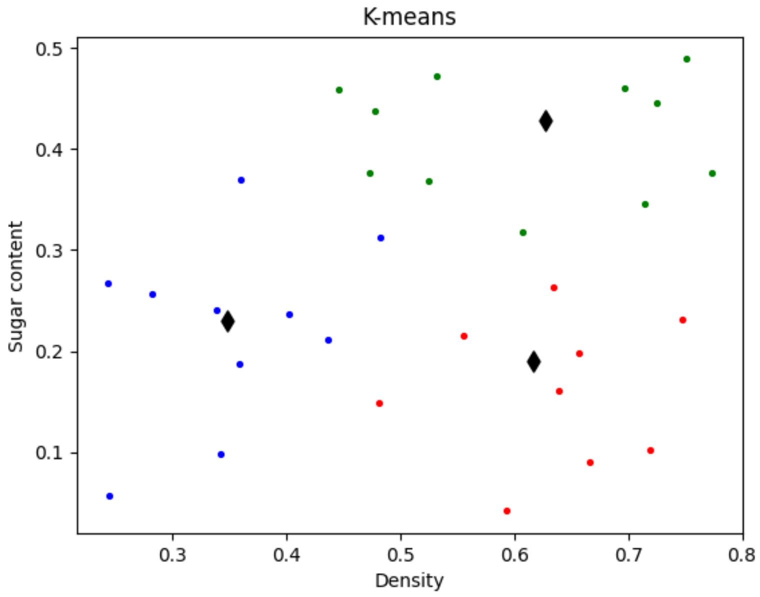
```
def K_means(dataset, k):
    # 初始化k个簇
    n_cluster = []
    for i in range(k):
        temp = []
        n_cluster.append(temp)

    # 计算总样本数
    m = len(data)

    # 随机生成k个随机数
    k_random = random.sample(range(0, m), k)
    # 存储均值向量
    mean = []
    for i in range(k):
        mean.append(dataset[k_random[i]])
    flag = 1

    while flag:
        for i in range(m):
            # 初始化距离列表
            dist = []
            # 计算每个样本到均值向量的距离
            for j in range(k):
                dist.append(e_dist(dataset[i], mean[j]))
            # 返回最小距离的下标
            min_index = dist.index(min(dist))
            # 将该元素添加到对应簇类中（若已存在，则不添加）
            if dataset[i] not in n_cluster[min_index]:
                n_cluster[min_index].append(dataset[i])
            # 如果该元素已经在其他簇类中，就将其删除
            for a in range(k):
                if a != min_index:
                    if dataset[i] in n_cluster[a]:
                        n_cluster[a].remove(dataset[i])
        # 初始化更新后的均值向量
        mean_update = []
        # 计算更新后的均值向量
        for i in range(k):
            sum0 = 0.0
            sum1 = 0.0
            if n_cluster[i]:
                for j in range(len(n_cluster[i])):
                    sum0 += n_cluster[i][j][0]
                    sum1 += n_cluster[i][j][1]
                mean_update.append([sum0 / len(n_cluster[i]), sum1 / len(n_cluster[i])])
        # 判断更新后的均值向量是否与上次相同，若相同，则结束循环，分类完毕；若不同，则继续循环
        if mean_update == mean:
            flag = 0
        else:
            mean = mean_update
```

结果如下：（以**k = 3** 为例）









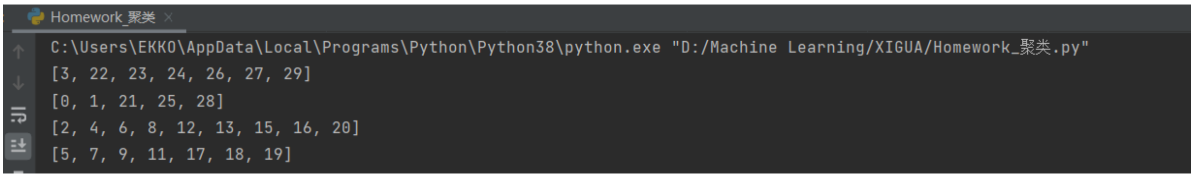
其中 ♦ 为每次更新后的均值向量，由图可以看到经过4次迭代后，各个簇分类完成。

2、DBSCAN算法

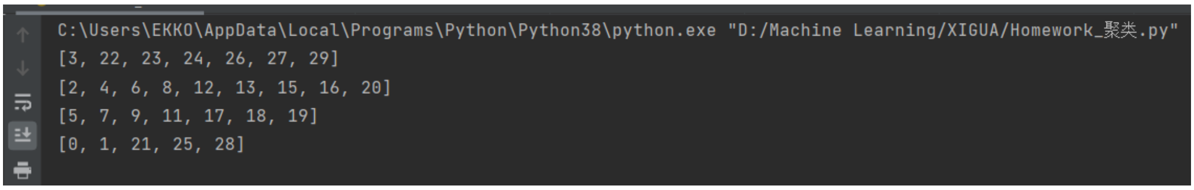
```
# 以下代码参考西瓜书上的伪代码，注释不再另写
def DBSCAN(dataset, minpts, epsilon):
    m = len(dataset)
    core_obj = []
    contain = []
    cluster = [[]]

    for i in range(m):
        contain.append([])
        for j in range(m):
            if e_dist(dataset[i], dataset[j]) <= epsilon:
                contain[i].append(j)
        if len(contain[i]) >= minpts:
            core_obj.append(i)
```

```
no_visited = list(range(30))

while len(core_obj) > 0:
    old_v = copy.deepcopy(no_visited)
    o = random.sample(core_obj, 1)[0]
    Q = []
    Q.append(o)
    no_visited.remove(o)
    while len(Q) > 0:
        q = Q[0]
        Q.remove(q)
        if len(contain[q]) >= minpts:
            temp = [x for x in contain[q] if x in no_visited]
            for x in temp:
                Q.append(x)
            for x in temp:
                no_visited.remove(x)
    for x in no_visited:
        old_v.remove(x)
    cluster.append(old_v)
    for x in old_v:
        if x in core_obj:
            core_obj.remove(x)

for i in range(1,len(cluster)):
    print(cluster[i])
```

运行结果：



```
C:\Users\EKKO\AppData\Local\Programs\Python\Python38\python.exe "D:/Machine Learning/XIGUA/Homework_聚类.py"
[3, 22, 23, 24, 26, 27, 29]
[0, 1, 21, 25, 28]
[2, 4, 6, 8, 12, 13, 15, 16, 20]
[5, 7, 9, 11, 17, 18, 19]
```



```
C:\Users\EKKO\AppData\Local\Programs\Python\Python38\python.exe "D:/Machine Learning/XIGUA/Homework_聚类.py"
[3, 22, 23, 24, 26, 27, 29]
[2, 4, 6, 8, 12, 13, 15, 16, 20]
[5, 7, 9, 11, 17, 18, 19]
[0, 1, 21, 25, 28]
```

多次运行，结果相同。