计算机学院 数据科学概论 课程实验报告

实验题目：K-Means 聚类算法		学号：202200130041
日期：2023/4/13	班级： 22 级数据	姓名： 左景萱
Email：zuojingxuan1130@mail.sdu.edu.cn		

实验目的：

Kmeans 是一种无监督的机器学习方法，用于把数据分成 k 个类簇。 本实验首先安装配置好 Anaconda 以及 Python 环境，通过在一个二维数组上用 K-Means 算法实现聚类操作，从而使同学们了解 K-Means 算法的原理并学会将其运用到实践中。

实验软件和硬件环境：

1. 操作系统：Windows10；
2. Anaconda 版本：5.3.0；
3. Python 版本：3.6.8；
4. matplotlib 包；
5. sklearn 包；

实验步骤与内容：

1. 首先通过 sklearn 包中的 cluster 中的 KMeans 算法进行了 KMeans 算法的简明实现，知道了 KMeans 聚类算法的功能和函数中每一个参数所代表的意思。接着利用 plt 画出了聚类之后的点的分布，直观的展现了 KMeans 分类的准确性。

2. 根据 KMeans 的原理自己实现 KMeans 算法，填补框架中的空白。

```
def randCent(dataSet, k):
    np.random.seed(0)#使得随机数据可预测，当我们设置相同的seed，每次生成的随机数相同，这里是为了方便验收结果
    """
    待填空
   PS：可用np.random.choice(listA,k,replace=False) 从listA中随机出不重复的k个index
   注意：为了对应下面的程序，这里centroids中的每个质心应返回二维position，而不是index

   """


    centroids_choice=np.random.choice(dataSet.shape[0],k,replace=False)
    centroids=dataSet[centroids_choice,:]
    return centroids
```

（1）. 首先设置随机数种子为 0 以获得稳定的结果，然后用 centroids_choice 记录随机抽取的几个初始化质心位置的 index，再从 dataset 中选出这几个质心的坐标进行返回。

（2）. 首先定义返回的存有分簇类和欧式距离的数组 clusterAssment, 质心 centroids 和更新状态 clusterChange。

```python
def KMeans(dataSet, k):
    m = np.shape(dataSet)[0]  # 行的数目
    # 第一列存样本属于哪一簇
    # 第二列存样本的到簇的中心点的误差
    clusterAssment = np.mat(np.zeros((m, 2)))
    clusterChange = True

    # 第1步 初始化centroids
    centroids = randCent(dataSet, k)
```

（3）. 接下来进行更新操作：按行遍历样本对于每一行样本遍历质心 array 找到最靠近的簇中心点并且记录中心点和距离。对于质心 array 进行遍历之后，更新最靠近的质心。如果需要更新，则认为这次仍不是最优。设置更新状态为需要更新，否则对于这行样本即为最优。

```python
# 遍历所有的样本（行数）
for i in range(m):
    minDist = 100000.0
    minIndex = -1

    # 遍历所有的质心
    # 第2步 找出最近的质心
    for j in range(k):
        """
        待填空：
        计算该样本到质心的欧式距离
        更新minDist & minIndex
        """
        dis=distEclud(centroids[j,:],dataSet[i,:])
        if dis<minDist:
            minDist=dis
            minIndex=j
    # 第 3 步：更新每一行样本所属的簇
    if clusterAssment[i, 0] != minIndex:
        """
        待填空：
        更新每一样本所属的簇clusterAssment
        判断是否clusterChange->是否要继续循环
        """
        clusterAssment[i,0]=minIndex
        clusterAssment[i,1]=minDist**2
        clusterChange=True
```

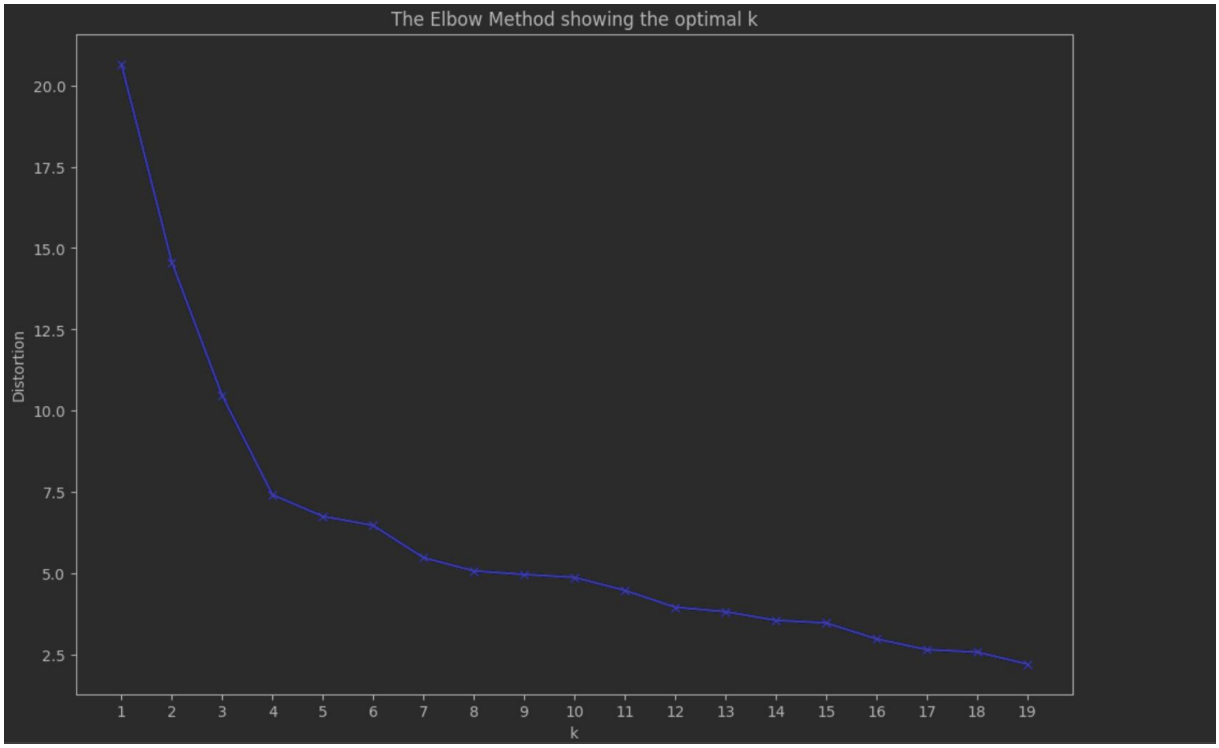
（4）. 将数据样本按照最近的质心进行分组，更新质心的坐标为其对应的所有样本的坐

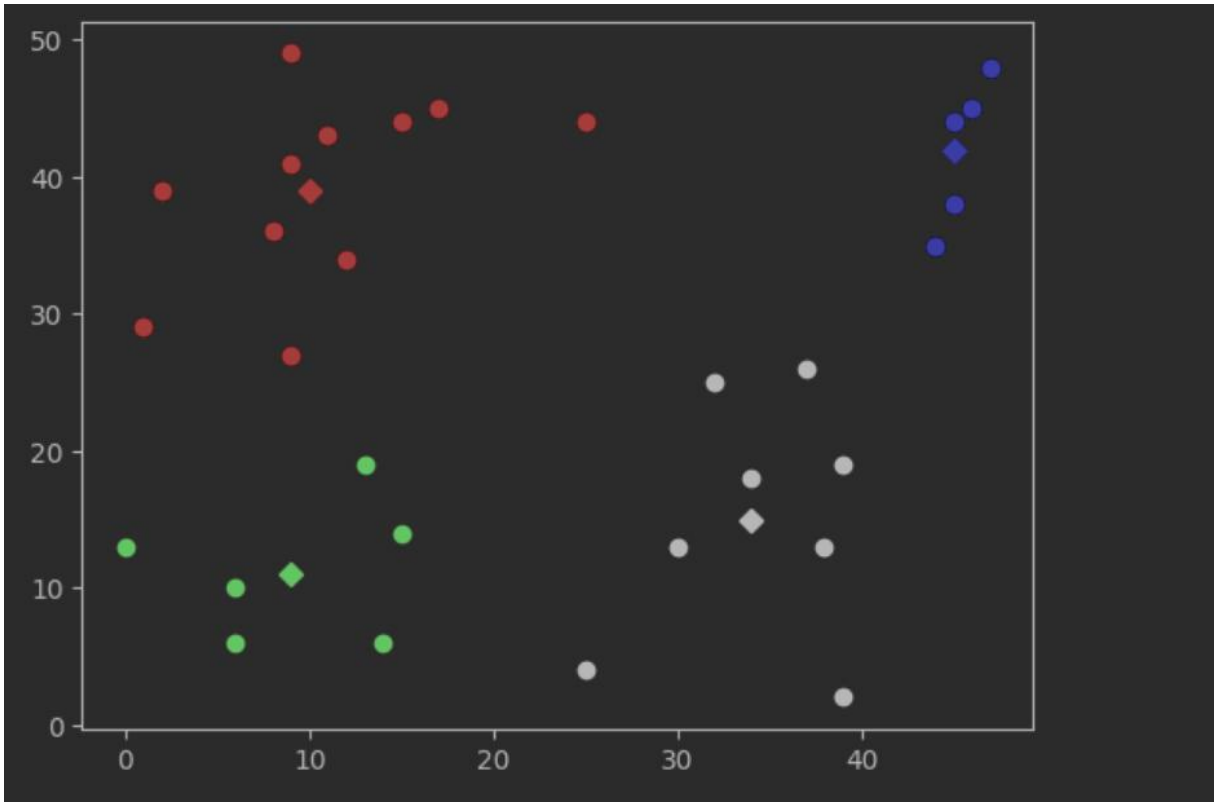
标的平均值。

```python
# 第 4 步：更新质心：
for j in range(k):
    """
    待填空：
    获取簇类所有的点
    对类中所有点求均值更新质心
    """
    pointsInCluster = dataSet[np.nonzero(clusterAssment[:, 0] == j)[0]]
    if len(pointsInCluster) > 0:
        # 对类中所有点求均值更新质心
        centroids[j, :] = np.mean(pointsInCluster, axis=0)
```

（5）. 循环更新操作，直至不需要更新为止。

（6）. 使用自己写的 KMeans 函数进行样本数据的绘图，选取最优的分类簇数量，再将分类之后的图像画出。





结论分析与体会：

1. KMeans 聚类算法可以较优的将多组数据分为若干类，但是对比 sklearn 中提供的 KMeans 算法的结果可以发现 sklearn 中提供的簇质心在均方误差的衡量标注之下是更优的。这说明我完成的算法还有很大的改进空间（比如将聚类中心的更新由平均值改为均方误差最小的点等）。

2. numpy 在类型和维度上的要求比较严格，运用的时候要注意数组的维度以及各种操作的返回值的维度和类型。

就实验过程中遇到的问题及解决处理方法，自拟 1—3 道问答题：

1. 下图的代码该如何修改？

```
pointsInCluster = dataSet[np.nonzero(clusterAssment[:, 0] == j)]
if len(pointsInCluster) > 0:
    # 对类中所有点求均值更新质心
    centroids[j, :] = np.mean(pointsInCluster, axis=0)
```

答：应该进行如下修改：

```
pointsInCluster = dataSet[np.nonzero(clusterAssment[:, 0] == j)[0]]
if len(pointsInCluster) > 0:
    # 对类中所有点求均值更新质心
    centroids[j, :] = np.mean(pointsInCluster, axis=0)
```

将 np.nonzero(clusterAssment[:, 0] == j)改为 np.nonzero(clusterAssment[:, 0] == j)[0]。
因为 np.nonzero()返回的是一个 tuple 类型的数据，必须转为别的类型（如 ndarray）才能进行接下来的操作，所以需要用[0]进行数据的取出。否则会出现一些别的问题。