假设我们有一个二维的dataset长这个样子：



k means算法的实质就是找出一个中心使该类到这个中心的距离之和最小

伪代码：

随机设置k个中心点（从这么多点里选3个）

loop：

根据到中心点的距离将所有的点归为k类

平均一下，计算k类的新中心点

update中心直到中心不变为止

算下来的结果（运气好）：



运气不好：



所以对初始值还是比较敏感的，会有local minimum的问题，改进方法为二分k means

伪代码：

cluster数=1

while cluster数 < k：

选出到中心距离之和较大的cluster

对该cluster 进行 2-means

cluster数++

算下来的结果（运气一直都很好）：



然后我们能看出这个里面是三类，如果高维的没办法看的话，选择k的办法是plot一下total cost function（在这里是所有cluster点到中心的总距离之和），选择拐点处的值：



显然这里是3个比较好

三维的情况类似：

 

写的小代码：https://github.com/daiguochen007/Machine-Learning-Implementation