文档对接：

1. 聚类获取：（get cluster）

Base version：K-means

基本算法过程：

1、随机选取k个元素作为聚类中心

2、地图中的所有元素，计算它到k个聚类中心的距离并将其分到距离最小的聚类中心所对应的类中。

3、针对每个类别，更新计算它的聚类中心

4、迭代第2、3步直至中止。

聚类效果差的原因：

聚类簇数K值的初始化问题：会对聚类效果造成较大影响

Improved Version：Kmeans PlusPlus

基本算法过程：

1、随机选取一个元素作为第一个聚类中心

2、计算数据到之前n个聚类中心距离。

3、基于概率选择新的聚类中心 。

3、迭代第2步直至中止。

效果提升：

确定化聚类初始化，降低由于随机初始化聚类中心而引来的偏差。

KmeansPlusPlus Realization

1、封装ClusterNode作为Kmeans PlusPlus的聚类的节点单元，通过利用Apache Open Source Machine Learning Model进行聚类。

2、

maxIterations：最大迭代次数

DistanceMeasures：两个向量之间的距离

RandomGenerator：设置种子等

EmptyClusterStrategy：

* FARTHEST\_POINT距离最远的创造一个聚类中心
* LARGEST\_POINTS\_NUMBER分割点数最多的聚类
* LARGEST\_VARIANCE插入掉距离距离方差最大的聚类

3、通过不断迭代调试，获取最优参数列表。

1. 智能体分配：（assign agent）

匈牙利算法[hungarian algorithm]：任务分配(二分图最大匹配)算法

利用代价矩阵表示智能体与聚类的之间的代价，通过欧式定律作为代价矩阵中的代价值，代价矩阵的横纵两边即为二分图最大匹配的对应类别的元素，最后返回一个分配结果,将智能体分配到对应的聚类。

匈牙利算法运行过程

a.每一行减去该行最小值

b.通过寻找增广路径的方式，尝试进行不带权重的匹配，只有值为0的项为可匹配项

c.如果最大匹配数为n，则结束，当前匹配为最优匹配，否则执行第4步

d.用最少的横线和竖线覆盖所有零值，取mv=未被覆盖的矩阵块的最小元素，所有未被覆盖的矩阵块的元素减去mv，所有被横竖线都覆盖的矩阵块的元素加上mv，然后回到第2步