**聚类与KMeans**

与分类、序列标注等任务不同，聚类是在事先并不知道任何样本标签的情况下，通过数据之间的内在关系把样本划分为若干类别，使得同类别样本之间的相似度高，不同类别之间的样本相似度低（即增大类内聚，减少类间距）。

聚类属于非监督学习，K均值聚类是最基础常用的聚类算法。它的基本思想是，通过迭代寻找K个簇（Cluster）的一种划分方案，使得聚类结果对应的损失函数最小。其中，损失函数可以定义为各个样本距离所属簇中心点的误差平方和：

**具体步骤**

KMeans的核心目标是将给定的数据集划分成K个簇（K是超参），并给出每个样本数据对应的中心点。具体步骤非常简单，可以分为4步：

（1）数据预处理。主要是标准化、异常点过滤。

（2）随机选取K个中心

（3）定义损失函数J

（4）令t=0,1,2,... 为迭代步数，重复如下过程直到J收敛：

（4.1）对于每一个样本，将其分配到距离最近的中心

（4.2）对于每一个类中心k，重新计算该类的中心

**KMeans最核心的部分就是先固定中心点，调整每个样本所属的类别来减少J；再固定每个样本的类别，调整中心点继续减小J。两个过程交替循环，J单调递减直到最（极）小值，中心点和样本划分的类别同时收敛。**

**优缺点与优化方法**

KMenas的优点：

* 高效可伸缩，计算复杂度O（NKt）为接近于线性（N是数据量，K是聚类总数，t是迭代轮数）。
* 收敛速度快，原理相对通俗易懂，可解释性强。

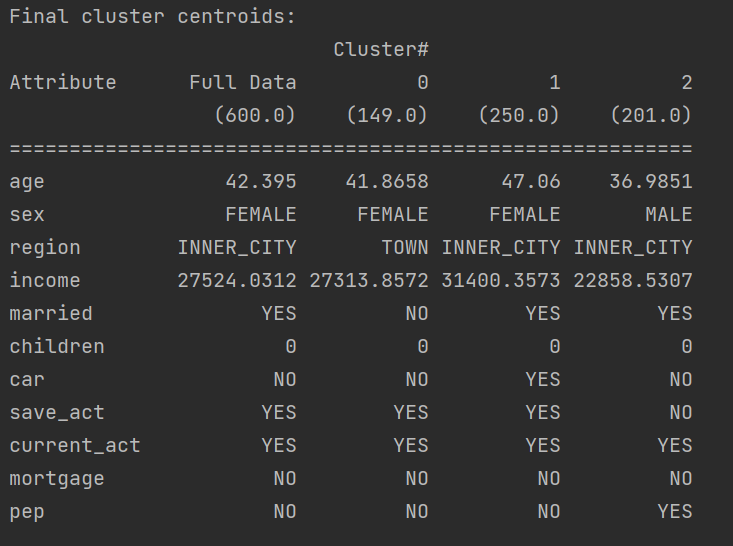
KMeans也有一些明显的缺点：

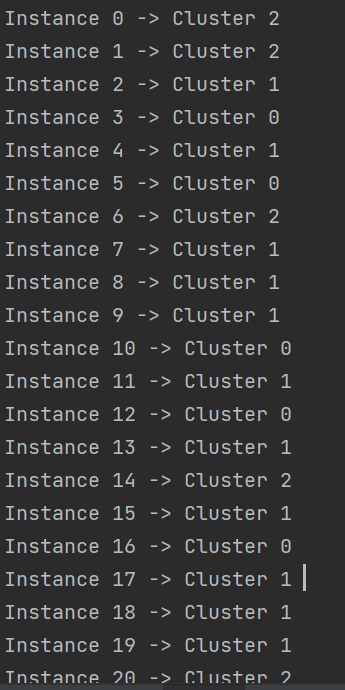
* 受初始值和异常点影响，聚类结果可能不是全局最优而是局部最优。
* K是超参数，一般需要按经验选择
* 样本点只能划分到单一的类中

**数据集处理**

对数据集进行聚类，希望将其聚为3类，将数据集读取为Instances对象，创建SimpleKMeans对象，并setNumCluster(3)，通过setPreserveInstancesOrder(true);来保留顺序。考虑输出聚类的同时，输出每个数据分别被分进了哪个类。

**运行结果**

****

****