**杭州电子科技大学计算机学院**

**数据仓库与数据挖掘**

**实验3：聚类分析**

时间：2022年1月1日，学号：19151633姓名：应宇杰

# 一、实验目的

1、理解聚类的一般过程和基本原理；

2、巩固聚类算法的算法思想，能够进行聚类操作；

3、学会聚类中的性能评估方法。

# 二、实验原理

**1、常用的聚类算法：K-Means、K-Mediods、凝聚层次聚类和 DBSCAN算法等。**

（1） K-Means(K均值)聚类

算法步骤：

(1) 首先我们选择一些类/组，并随机初始化它们各自的中心点。中心点是与每个数据点向量长度相同的位置。这需要我们提前预知类的数量(即中心点的数量)。

(2) 计算每个数据点到中心点的距离，数据点距离哪个中心点最近就划分到哪一类中。

(3) 计算每一类中中心点作为新的中心点。

(4) 重复以上步骤，直到每一类中心在每次迭代后变化不大为止。也可以多次随机初始化中心点，然后选择运行结果最好的一个。

优点：

速度快，计算简便

缺点：

必须提前知道数据有多少类/组。

（2）K-Mediods

k-均值算法采用簇的质心来代表一个簇，质心是簇中其他对象的参照点。因此，k-均值算法对孤立点是敏感的，如果具有极大值，就可能大幅度地扭曲数据的分布。

k-中心点算法是为消除这种敏感性提出的，它选择簇中位置最接近簇中心的对象（称为中心点）作为簇的代表点，目标函数仍然可以采用平方误差准则。

处理过程：首先，随机选择k个对象作为初始的k个簇的代表点，将其余对象根据其与代表点对象的距离分配到最近的簇； 然后，反复用非代表点来代替代表点，以改进聚类质量，聚类质量用一个代价函数来估计，该函数度量对象与代表点对象之间的平均相异度。

**输入:**n个对象的数据库，期望得到的簇的数目k

**输出:**使得所有对象与其最近中心点的偏差总和最小化的k个簇

**方法：**

选择k个对象作为初始的簇中心

repeat

对每个对象，计算离其最近的簇中心点，并将对象分配到该中心点代表的簇

随机选取非中心点Orandom

计算用Orandom 代替Oj 形成新集合的总代价S

如果S<0，用Orandom代替Oj，形成新的k个中心点的集合

until 不再发生变化

采用k-中心点算法有两个好处：

对属性类型没有局限性；

通过簇内主要点的位置来确定选择中心点，对孤立点的敏感性小

不足：

处理时间要比k-mean更长；

用户事先指定所需聚类簇个数k。

（3）凝聚层次聚类

层次聚类算法分为两类：自上而下和自下而上。凝聚层级聚类(HAC)是自下而上的一种聚类算法。HAC首先将每个数据点视为一个单一的簇，然后计算所有簇之间的距离来合并簇，知道所有的簇聚合成为一个簇为止。

具体步骤：

(1) 首先将每个数据点视为一个单一的簇，然后选择一个测量两个簇之间距离的度量标准。例如使用average linkage作为标准，它将两个簇之间的距离定义为第一个簇中的数据点与第二个簇中的数据点之间的平均距离。

(2) 在每次迭代中，将两个具有最小average linkage的簇合并成为一个簇。

(3) 重复步骤2知道所有的数据点合并成一个簇，然后选择需要多少个簇。

层次聚类优点：（1）不需要知道有多少个簇

（2）对于距离度量标准的选择并不敏感

缺点：效率低

（4）基于密度的聚类方法(DBSCAN)

具体步骤：

(1) 首先确定半径r和minPoints. 从一个没有被访问过的任意数据点开始，以这个点为中心，r为半径的圆内包含的点的数量是否大于或等于minPoints，如果大于或等于minPoints则改点被标记为central point,反之则会被标记为noise point。

(2) 重复1的步骤，如果一个noise point存在于某个central point为半径的圆内，则这个点被标记为边缘点，反之仍为noise point。重复步骤1，知道所有的点都被访问过。

优点：不需要知道簇的数量

缺点：需要确定距离r和minPoints

**2、 评估方法**

TP、True Positive   真阳性：预测为正，实际也为正

FP、False Positive  假阳性：预测为正，实际为负

FN、False Negative 假阴性：预测为负、实际为正

TN、True Negative 真阴性：预测为负、实际也为负。

例如：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | 总计 |
| TP | 2 | 2 | 1 | 5 |
| FP | 0 | 2 | 1 | 3 |
| FN | 2 | 1 | 1 | 4 |

**精确率**：

Precison=TP/(TP+FP)

**召回率**：

Recall=TP/(TP+FN)

**F1分数**（F1-Score），又称为平衡F分数（BalancedScore），它被定义为精确率和召回率的调和平均数。

**F1=（2\*** Precison\* Recall）/**（**Precison+Recall）

**Micro-F1和Macro-F1**

假设已经通过聚类算法，将待分类数据进行了聚类。对每个“类”，计算F1，然后所有类的F1合并起来考虑。这里有两种合并方式：

第一种计算出所有类别总的Precision和Recall，然后计算F1。

例如依照最上面的表格来计算: Precison=5/(5+3)=0.625,Recall=5/(5+4)=0.556，然后代入F1的公式求出F1，这种方式被称为**Micro-F1**微平均。

第二种方式是计算出每一个类的Precison和Recall后计算F1，最后将F1平均。

例如上式A类：P=2/(2+0)=1.0，R=2/(2+2)=0.5，F1=(2\*1\*0.5)/(1+0.5)=0.667。同理求出B类C类的F1，最后求平均值，这种方式叫做**Macro-F1**宏平均。

**轮廓系数：**

假设已经通过聚类算法，将待分类数据进行了聚类。

对于簇中的每个向量。分别计算它们的轮廓系数。对于其中的一个点 i 来说：

计算 a(i) = average(i向量到所有它属于的簇中其它点的距离)

计算 b(i) = min (i向量到与它相邻最近的一簇内的所有点的平均距离) 即：i到不包含i的所有簇的最小平均距离。

那么 i 向量轮廓系数就为：

https://gss1.bdstatic.com/-vo3dSag_xI4khGkpoWK1HF6hhy/baike/pic/item/4610b912c8fcc3ce5691cc0f9745d688d43f20bc.jpg

可见轮廓系数的值是介于 [-1,1] ，越趋近于1代表内聚度和分离度都相对较优。

**将所有点的轮廓系数求平均，就是该聚类结果总的轮廓系数**。

a(i) ：i向量到同一簇内其他点**不相似程度**的平均值

b(i) ：i向量到其他簇的平均**不相似程度**的最小值

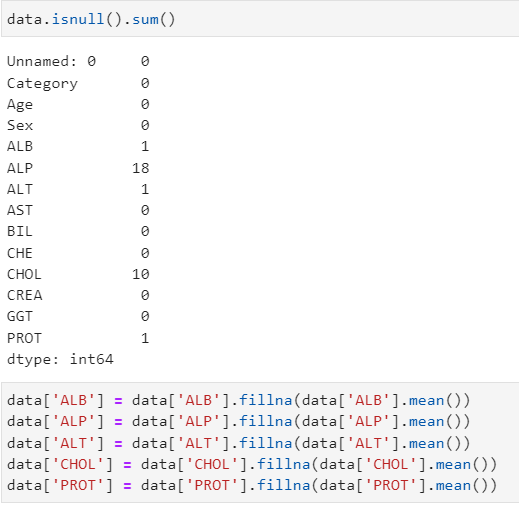
# 三、实验内容：

从**K-means、层次聚类、DBSCAN中选用1种聚类方法对HCV data数据集进行聚类分析， 对得到聚类结果采用**Micro-F1和Macro-F1作为评估指标进行评估。

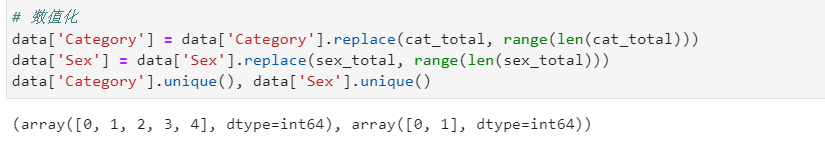
# 四、实验步骤

**1、数据预处理**

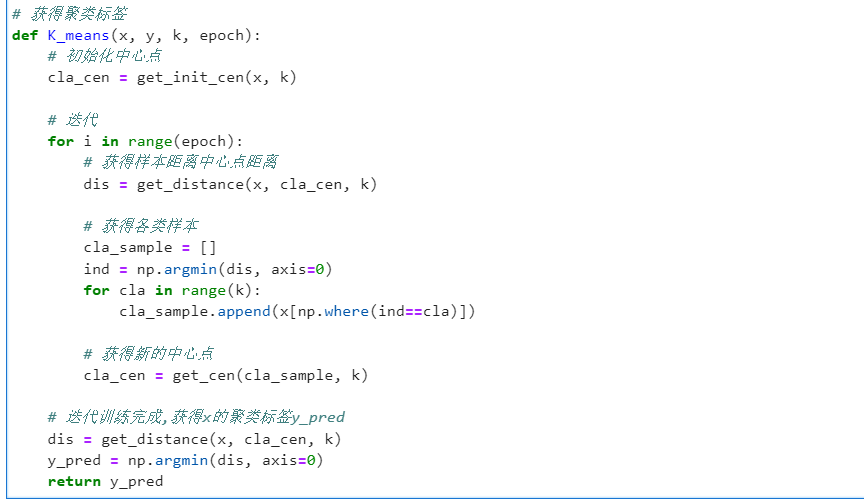
**1.1、缺失值填充，缺失的值使用平均数填充**



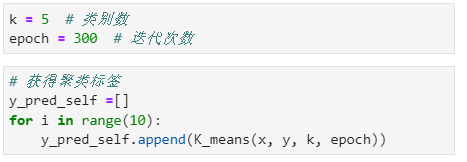
**1.2、字符串数值化**



**2、使用K-means算法进行聚类分析**



但是再实验过程中发现k\_maens聚类算法易受初始化中心点的影响，正确率存在很大的波动。所有我们选择若干次聚类，取正确率最高的作为理想聚类方式。



**2.1、获得有实际意义的聚类标签**

聚类是一种无标签算法，其所得聚类标签没有实际含义，但数据集中提供的标签具有实际含义，因此两者取值可能会有所不同为了便于评价聚类性能，我们希望得到聚类标签经过某种排列能够具有实际含义以便与真实标签比较实现上，考虑类别较小， 我们遍历各种取值排列，选用正确率最高的排列作为我们进行评估的聚类标签



**4、评价指标**

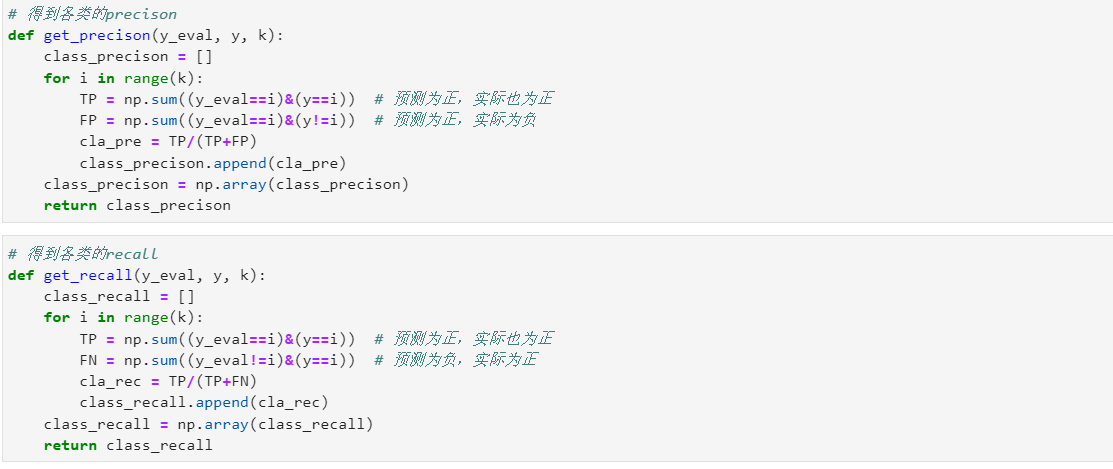
**精确率**：

Precison=TP/(TP+FP)

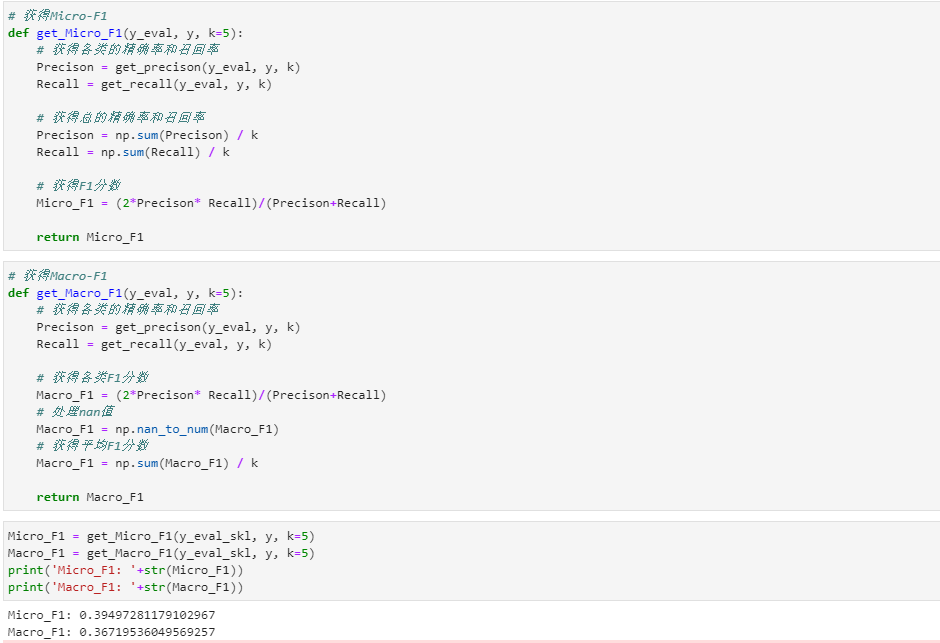
**召回率：**

Recall=TP/(TP+FN)

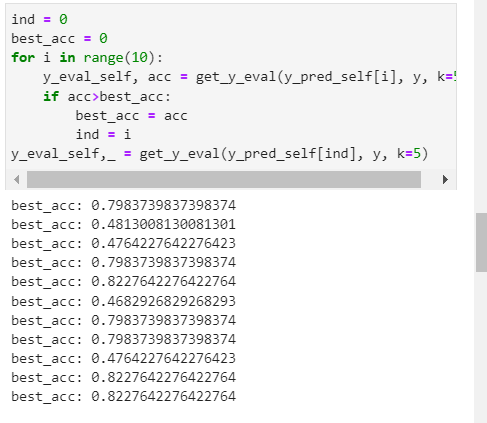
**4.1计算精确率和召回率**

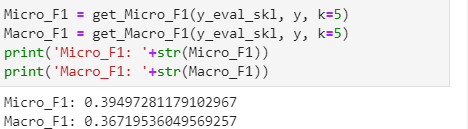


**4.2、计算Micro-F1和Macro F1**



# 五、实验结果





# 我们采用手动实现k-means算法的方法， 分别取得了82.76%,证明我们的聚类能很好的划分类别，另外，两种方法取得的Micro\_F1和Macro\_F1值介于36%-40%之间证明我们的聚类对样本量较小的类别的分类效果并不理想，模型缺乏对小样本类别较好的划分能力

# 六、心得体会

#### 通过这次聚类分析的实验，使我学会了k-means算法的基本算法理念和实现方法，并且最后学习了聚类算法的Micro-F1和Macro F1评估方法，并且在实验中k-means算法取得了不错的正确率。在numpy，pandas等库也在完成这次实验后越发熟练使用。我在求取Macro-F1的实验过程中出现问题：若聚类标签恰好使得某类的精确率Precison和召回率Recall都为0，即TP为0时，则F1公式的分母为0，导致F1值为nan，加入0值判断即可解决。