《机器学习》读书笔记

1. 算法概述（算法名称及原理）

* apriori算法

算法背景：

从大规模数据集中寻找物品间的隐含关系被称作关联分析（association analysis）或者关联规则学习（association rule learning）。

关联分析的任务包括两项：发现频繁项集和从频繁项集中发现关联规则。

* 频繁项集（frequent item sets）：是经常出现在一块的物品的集合，例如香烟和打火机。Apriori 算法是发现频繁项集的一种方法
* 关联规则（association rules）：暗示两种物品之间可能存在很强的关系，通常表示物品之间的“如果…那么”关系，例如“如果购买香烟，那么有很大概率会购买打火机”。

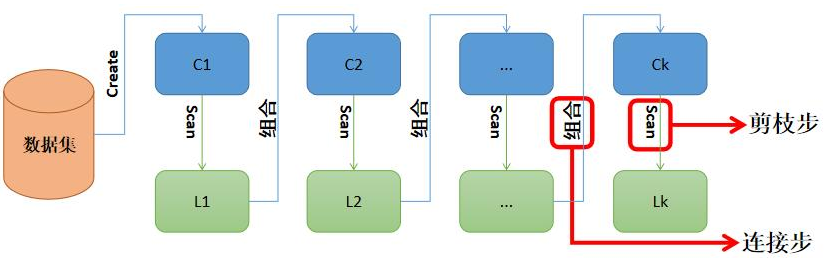
定量地衡量一物品集合是否频繁的概念：

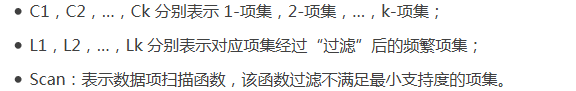
* 项集：项的集合，项可以是商品，那么项集就是商品的集合。
* 支持度：数据集中包含该项集的记录所占的比例，也就是该项集在数据集中的出现频率，用以衡量项集的频繁程度。
* 可信度：又称置信度，是针对关联规则来定义的，表示某项集在指定条件下的出现概率，用以衡量物品之间的关系。

支持度和可信度是用来量化关联分析是否成功的方法，但当物品成千上万时，蛮力计算不仅耗时也有非常高的计算代价（对于包含 N 种物品的数据集共有 2^N−1 种项集组合），这时可以选择Apriori 原理减少关联规则学习时所需的计算量

* apriori算法原理
* 如果某个项集是频繁的，那么它的所有子集也是频繁的
* 如果某个项集是非频繁的，那么它的所有超集也是非频繁的（超集可以理解为父集，和子集相应）

1. 算法设计（流程图及主要分段代码，附详细代码注释）



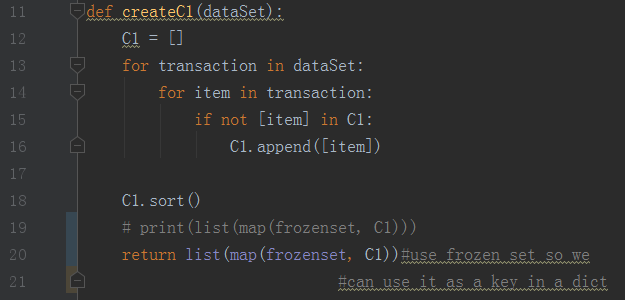


连接步：分为两种情况，第一是从数据集中生成 C1，第二是根据 *Lk*−1生成 Ck。简单地说，连接步就是产生项集的过程。

剪枝步：剔除不满足最小支持度的项集。从图中来看就是从 Ck 到 Lk 的过程

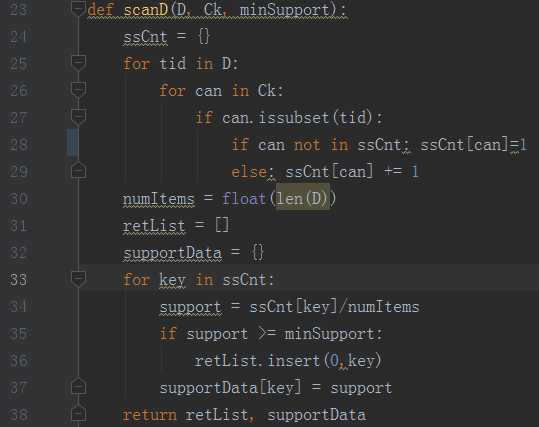
频繁项集代码-1

创建初始C1用于存放候选项集



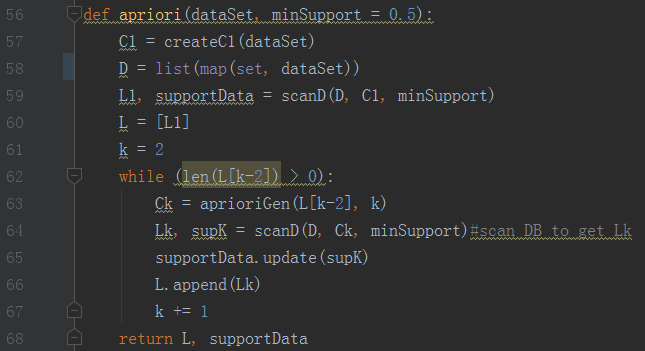
频繁项集代码-2

scan部分，用于从C1生成L1



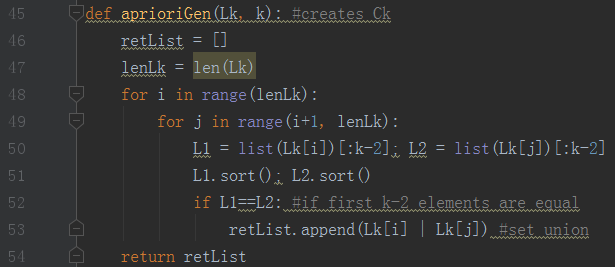
频繁项集代码-3

apriori主体部分代码



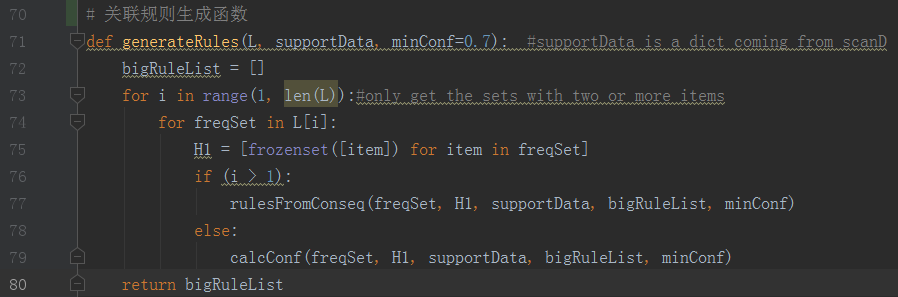
频繁项集代码-4

aprioriGen()函数用于创建候选项集Ck



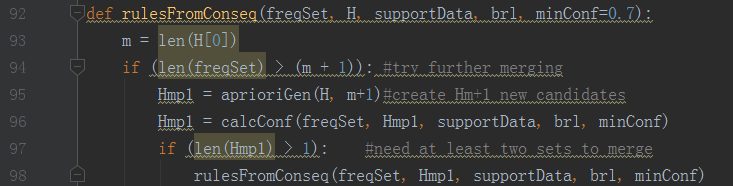
关联规则代码-1

关联规则生成函数



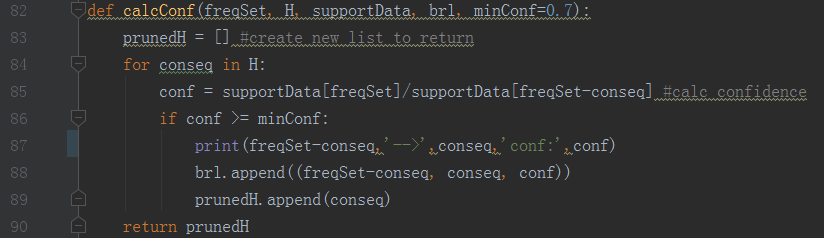
关联规则代码-2

生成候选规则集合



关联规则代码-3

对规则进行评估



1. 选用数据（数据集描述，包括来源，行数，列数，格式等）

数据集一：



数据集二：

iris鸢尾花，行数：150，列数：4

列属性：

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

类别：  
Iris Setosa ，Iris Versicolour ，Iris Virginica

数据集三：

car，行数：1728，列数：6

列属性及取值：  
1) buying: vhigh, high, med, low.   
2) maint: vhigh, high, med, low.   
3) doors: 2, 3, 4, 5more.   
4) persons: 2, 4, more.   
5) lug\_boot: small, med, big.   
6) safety: low, med, high.

类别：

unacc, acc, good, vgood

数据集四：

wine，行数：178，列数：13

属性：

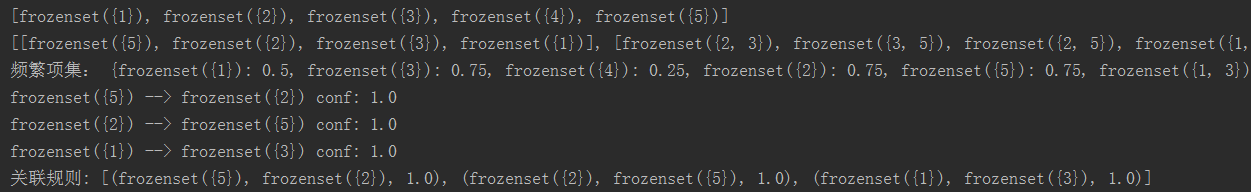
1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

类别：

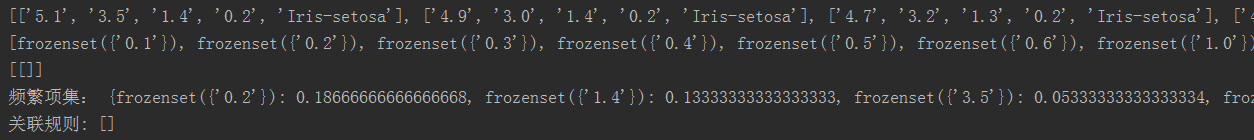
Alcohol 1，2，3

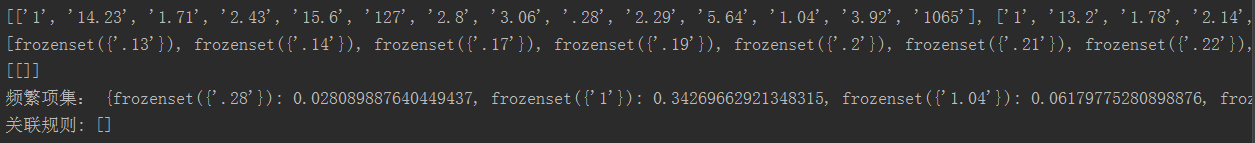
1. 评价方法（说明训练集和测试集分配方法及评价指标）
2. 实验结果截图

数据集一实验结果：

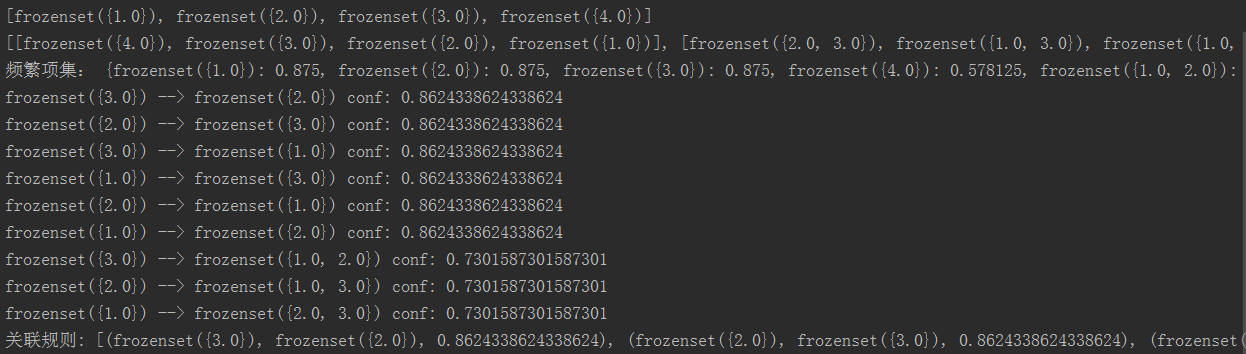


iris实验结果：



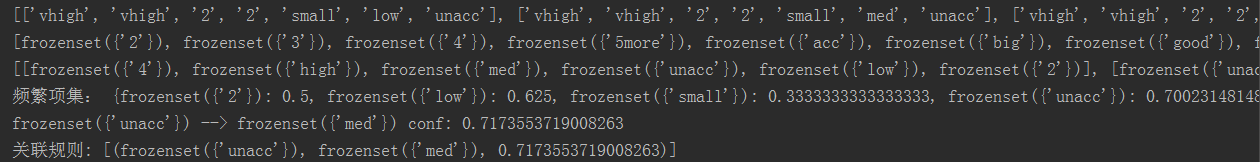
wine实验结果：

car实验结果（未使用分类属性进行实验）：



car实验结果（加上分类属性）

没有选择数值化数据，如果数值化数据的话因为car的属性值几乎都是1，2，3，4，会对实验结果造成非常大的影响，或者可以刻意避免属性值相同的情况即赋予新值，比如毒蘑菇的例子，第一个属性分类是1，2，第二个属性的值就从3开始



从关联结果可以看出，如果一个汽车不被接受，那么它很大可能med属性值比较多，但是这个实验存在一个比较大的不足在于，它把不同属性相同的值作为一个值处理，这对于实验的结果肯定会造成一定的影响

1. 实验结果分析及比较

因为本次算法是用于解决关联分析问题，所以选用的数据集一般要有一定的关联才能看出实验的效果，观察到鸢尾花数据，葡萄酒数据的分布都十分分散（iris，wine是数值型数据），若硬要将属性数据集应用于apriori算法，得到的实验结果十分差强人意，我们发现频繁项集的支持度非常低，几乎没有关联的可能；

对于汽车数据集（car是标称型数据，标称型目标的结果只有在有限目标集中取值），由于数值化后的数据比较集中，所以使用apriori算法模拟实验效果良好，实际上，关联分析是发现大数据集合元素间关系的工具集，可以用在不同的物品上，主要是在样本处理上需要将样本转换成特征集合，也就是将所有维的特征统一编码

1. 遇到的问题及解决方法，实践心得

问题一：

对C1进行排序后将其中的每个单元素列表映射到 frozenset()，最后返回 frozenset 的列表。需要注意的是这里使用的数据结构是 Python 中的 frozenset 类型。frozenset 是指被“冰冻”的集合，就是说它们是不可改变的，即用户不能修改它们。这里必须要使用 frozenset 而不是 set 类型，因为之后必须要将这些集合作为字典键值使用，使用 frozenset 可以实现这一点，而 set 却做不到





需要注意 Python 3.x 和 Python 2.x 不同，map() 函数返回的是可迭代对象，而不是列表

问题二：

为什么要取列表的前 k - 2 个数据



假设现在要将 {0，1}、{0，2}、{1，2} 来创建三元素项集，如果仅仅将两个项集合并，就会得到三个 {0，1，2}。也就是说，同样的结果会重复 3 次，我们要做的是确保遍历列表的次数最少。现在，如果只比较集合的第 1 个元素，并且只对第 1 个元素相同的集合求并操作，同样可以得到 {0，1，2}，且只需要一次操作

问题三：

在用鸢尾花等三个数据集进行实验的时候，若是只选择属性值进行实验可能会由于属性值的离散分布导致没有太大的关联分析

解决办法：

加上分类的属性值赋予实验新的意义，典型示例：发现毒蘑菇的相似特征

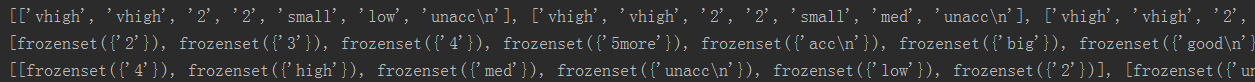
问题四：

如何将样本转换成特征集合，也就是将所有维的特征统一编码

问题五：

数据显示问题，观察到每一行末尾都有\n

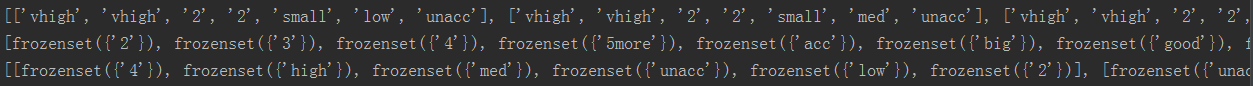




解决办法，加上strip()方法



数据正确显示



实验总结：

1：对于鸢尾花数据集，红酒数据集可以先离散化在进行实验，离散化的方法可见周师严同学以及张文楷同学的代码

2：为了避免出现不同属性同一值当成一个值进行处理可以加上前缀进行区分