频繁项集Apriori算法实验要求和内容

1. 实验目的
2. 熟悉、理解购物篮模型，理解并掌握Apriori算法过程
3. 能够编程实现Apriori算法，并能计算一些简单实例
4. 实验环境和工具

最好使用Python 3版本，能够用其他程序语言实现也可以

1. 算法过程：
2. 输入：购物篮数据集，支持度阈值
3. 初始化C1，得到所有的单项集
4. 扫描一遍购物篮，结合支持度阈值，得到Ck过滤后的Lk
5. 构造通过Lk构造下一步所需的C(k+1)
6. 重复上面3,4两步，直到没有频繁项出现
7. 输出所有频繁项
8. 实验过程
9. 定义构造C1的函数，完成初始化。储存方式可以设定为购物篮以二维数组储存数据，且数据都是数字，其代码对应为：

def create\_c1(date\_set):  
 c\_1 = set([])  
 for item in date\_set:  
 c\_1 = c\_1.union(set(item))  
 return [frozenset([i]) for i in c\_1] # 因为后面要用它作为字典的元素，所以需要用frozen形式

1. 定义构造Lk的函数，即完成算法中的过滤过程，对应的代码为：

def get\_lk(date\_set, c\_k, support\_threshold): # 建立Lk的函数  
 l\_k = {}  
 for item in date\_set: # 将整个字典扫描一遍  
 for c\_i in c\_k:  
 if c\_i.issubset(item):  
 if c\_i not in l\_k:  
 l\_k[c\_i] = 1  
 else:  
 l\_k[c\_i] += 1  
 l\_k\_return = [] # 用于返回的集合l\_k  
 for l\_i in l\_k:  
 if l\_k[l\_i] >= support\_threshold:  
 l\_k\_return.append(l\_i)  
 return l\_k\_return

1. 建立构造C(k+1)的函数：

def get\_ck1(l\_k):  
 c\_k\_1 = []  
 for i in range(len(l\_k)-1):  
 for j in range(i+1, len(l\_k)):  
 if sorted(list(l\_k[i]))[0:-1] == sorted(list(l\_k[j]))[0:-1]:  
 c\_k\_1.append(l\_k[i] | l\_k[j])  
 return c\_k\_1

1. 以上函数建立好之后，直接在主函数中调用函数，并通过循环完成频繁项集的计算工作，最后输出所有的频繁项集
2. 输入所用到的实例如下：

[[1, 3, 4], [2, 3, 5], [1, 2, 3, 5], [2, 5]]

支持度阈值设为2即可。