Data Mining Homework 1
Apriori Algorithm

M10915201
陳牧凡

Association Rules:

min_support=0.005
confidence, min_threshold=0.2
cofidence分數前六名:

antecedents	consequents
frozenset({'Organic Raspberries', 'Organic Hass Avocado'})	frozenset({'Bag of Organic Bananas'})
frozenset({'Organic Hass Avocado', 'Organic Strawberries'})	frozenset({'Bag of Organic Bananas'})
frozenset({'Organic Avocado', 'Large Lemon'})	frozenset({'Banana'})
frozenset({'Organic Navel Orange'})	frozenset({'Bag of Organic Bananas'})
frozenset({'Yellow Bell Pepper'})	frozenset({'Orange Bell Pepper'})
frozenset({'Limes', 'Organic Avocado'})	frozenset({'Large Lemon'})

1. 使用 Apriori algorithm，並自己設定 confidence 跟 support 試看看能 mine 出哪些 rule。 挑選其中兩條 rule 並說明該 rule 所代表的意義。

   apriori: min_support=0.005
   association rules: metric=confidence, min_threshold=0.2
   rule1:
   Organic Raspberries, Organic Hass Avocado -> Bag of Organic Bananas
   通常買水果的人通常不會只買一樣水果，而是一次買很多的種類，所以可以知道這條rule相關的。事實上，mine出來的rule中，水果類就佔了大多數。
   rule2:
   Yellow Bell Pepper -> Orange Bell Pepper
   黃甜椒和橘色甜椒經常一起買，因為甜椒在料理中經常被拿來當作是裝飾顏色用的食材，所以常常是一次購買各種顏色的甜椒。另外還有一個可能的原因就是黃色和橘色因為色系相近，顧客在選購的時候很可能會混淆，結果就拿了一顆黃的跟一顆橘的。

2. 在 confidence 設為最低的情況下（即為 0%），將 support 設定高於多少會剛好 mine 不出任何 rule？而這個門檻的 support 值代表什麼意義？

   0.03。經觀察，在apriori中找出來的item sets中size>=2(即一組set中包含兩樣以上的商品)，的最大support值為0.029，因此若將min support設為0.03的話，則每一樣商品都是獨立存在於set中，那麼在接下來計算rules時就會一條也找不到。

3. 在以 confidence 為 metric 且 support 設定為 0.01 的情況下，前 10 條最佳 rule 中，是 否發現某個商品常常出現在這些 rule 中？

antecedents	consequents
frozenset({'Organic Large Extra Fancy Fuji Apple'})	frozenset({'Bag of Organic Bananas'})
frozenset({'Organic Hass Avocado'})	frozenset({'Bag of Organic Bananas'})
frozenset({'Organic Fuji Apple'})	frozenset({'Banana'})
frozenset({'Honeycrisp Apple'})	frozenset({'Banana'})
frozenset({'Organic Raspberries'})	frozenset({'Bag of Organic Bananas'})
frozenset({'Organic Lemon'})	frozenset({'Bag of Organic Bananas'})
frozenset({'Strawberries'})	frozenset({'Banana'})
frozenset({'Organic Avocado'})	frozenset({'Banana'})
frozenset({'Seedless Red Grapes'})	frozenset({'Banana'})
frozenset({'Large Lemon'})	frozenset({'Banana'})

BANANA。經統計，BANANA光是單獨出現的機率就高達了17.8%，因此很容易出現BANANA搭配任一商品出現的組合。

support	itemsets
0.178267	frozenset({'Banana'})

4. 試著調整 support 及更改 metric，mine 出跟 Q3 結果較不一樣的 rule。

support就一樣維持0.005，再調高或調低都不太合適。
metric部分使用lift，threshold設定為3。
lift分數前六名:

antecedents	consequents
frozenset({'Yellow Bell Pepper'})	frozenset({'Orange Bell Pepper'})
frozenset({'Orange Bell Pepper'})	frozenset({'Yellow Bell Pepper'})
frozenset({'Lime Sparkling Water'})	frozenset({'Sparkling Water Grapefruit'})
frozenset({'Sparkling Water Grapefruit'})	frozenset({'Lime Sparkling Water'})
frozenset({'Green Bell Pepper'})	frozenset({'Red Peppers'})
frozenset({'Red Peppers'})	frozenset({'Green Bell Pepper'})

lift metric計算上會多考慮到Consequent是不是獨立的。從上面用confidence metric的結果找出來的rules中很多都是本身單獨出現機率都很高的商品，因此有時候未必能夠說明這個Antecedents和Consequents是具有關聯性的。而lift metric則避開了這樣的問題，即便這個item set本身的support不高，計算lift時還是有機會得到較高的分數。如上面所找出來lift分數最高的幾個組合，其support值均不高，甚至都落在min support邊界值的0.005附近，不過透過lift metric便可以找出此類出現頻率不高但較相互依賴的商品組合。

5. 簡單描述實作本次作業的過程。

我是自行使用python並且透過已實作的套件:mlxtend.frequent_patterns中的apriori、association_rules兩個function來找到關聯的rules。因為這兩個function輸入都吃DataFrame，而作業一開始給的資料集就是一個csv檔，於是就可以直接透過panda.read_csv直接將檔案讀進來，然後直接跑出我們要的答案。

參數設定上面，由於mlxtend上的apriori以及association rules的function預設的min support以及min threshold分別是0.5和0.8，但直接跑的話是沒有任何結果的，原因是因為在總計7500筆交易紀錄中，每個商品出現的頻率並不會太高，下面統計了出現次數最高的商品前三名，可以看到就連最常出現的商品也只出現1337，機率相當於17.8%。

```
7 import pandas as pd
8
9 records = pd.read_csv('records.csv')
10
11 count = records.sum() #count each item fq
12
13 #%%
```

Index	0
Banana	1337
Bag of Organic Bana…	1143
Organic Strawberries	799

由此可見，min support設定要遠小於0.1，這樣在用apriori找frequent sets的時候才找的到support值符合的sets。經嘗試過後apriori min support設定在0.005~0.01間較為合適。

而association rules則要根據選用的metric不同而設定不同值，如metric為confidence時threshold設定至少大於min support才有意義；而metric為lift時至少要設定大於1 mine出來的rules才是有意義的。