Homework 1. Frequent itemset

Double Click here to edit this cell

- Name: 전기범
- Student ID: 201703091
- Submission date: 2019/03/18

Remark. Do not import numpy, pandas, sklearn, or any module implementing the solution directly

Frequent itemset

- ***Support*** is an indication of how frequently the itemset X appears in the dataset T.
- The support of X with respect to T is defined as the proportion of transactions t in the dataset which contains the itemset X.

$$\text{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

- Frequent itemset is an itemset whose support $\geq$ ***min_sup***.

Data set ¶

- Each line in the following can be imagined as a market basket, which contains items you want to buy.

In [1]:
```
# DO NOT EDIT THIS CELL
data_str = 'apple,beer,rice,chicken\n'
data_str += 'apple,beer,rice\n'
data_str += 'apple,beer\n'
data_str += 'apple,mango\n'
data_str += 'milk,beer,rice,chicken\n'
data_str += 'milk,beer,rice\n'
data_str += 'milk,beer\n'
data_str += 'milk,mango'
```

Problem 1 (2 pts)

- Define a function ***record_gen*** generating a list of items each ***next***.
- It must be a generator.
- Use ***yield*** instead of ***return***

In [2]:
```
# YOUR CODE MUST BE HERE

def gen_record(s):
    an = ', '.join(s.split(",")).split("\n")
    for i in range(len(an)):
        yield an[i].split(', ')
```

In [3]:
```
# DO NOT EDIT THIS CELL
test = gen_record(data_str)
next(test)
```

Out[3]:
```
['apple', 'beer', 'rice', 'chicken']
```

Your output must be:

```
['apple', 'beer', 'rice', 'chicken']
```

In [4]:
```
# DO NOT EDIT THIS CELL
next(test)
```

Out[4]:
```
['apple', 'beer', 'rice']
```

Your output must be:

```
['apple', 'beer', 'rice']
```

Problem 2 (10 pts)

- Define a function ***gen_frequent_1_itemset*** generating 1-itemset.
- It must be a generator.
- We want to find frequent 1-itemset (itemset containing only 1 item)

In [5]:
```
# YOUR CODE MUST BE HERE

def gen_frequent_1_itemset(dataset, min_sup=0.5):
    word_cnt={}
    for i in range(len(dataset)):
        for word in dataset[i]:
            try:
                word_cnt[word]+=1
            except KeyError:
                word_cnt[word]=1
    for key,cnt in word_cnt.items():
        if cnt>=len(data_str.split("\n"))*min_sup:
            yield key
```

In [6]:
```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
for item in gen_frequent_1_itemset(dataset, 0.5):
    print(item)
print('No more items')
```

```
apple
beer
rice
milk
No more items
```

Your output must be:

```
rice
beer
milk
apple
No more items
```

In [7]:
```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
for item in gen_frequent_1_itemset(dataset, 0.7):
    print(item)
print('No more items')
```

```
beer
No more items
```

Your output must be:

```
beer
No more items
```

In [8]:
```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
for item in gen_frequent_1_itemset(dataset, 0.2):
    print(item)
print('No more items')
```

```
apple
beer
rice
chicken
mango
milk
No more items
```

Your output must be:

```
rice
chicken
beer
mango
milk
apple
No more items
```

Problem 3 (10 pts)

- Define a function ***gen_frequent_2_itemset*** generating 2-itemset.
- It must be a generator.
- We want to find frequent 2-itemset (itemset containing only 2 items)

In [9]:
```
# YOUR CODE MUST BE HERE

def gen_frequent_2_itemset(dataset, min_sup=0.5):
    d=set()
    data_set=[]
    for i in range(len(dataset)):
        a=dataset[i]
        for j in range(len(a)):
            d.add(a[j])
    d=list(d)
    for x in range(len(d)):
        for y in range(x+1,len(d)):
            data_set.append((d[x],d[y]))
    set_cnt = {}
    for _ in range(len(dataset)):
        for a in range(len(data_set)):
            cnt = 0
            if data_set[a][0] in dataset[_]:
                cnt+=1
            if data_set[a][1] in dataset[_]:
                cnt+=1
            if cnt==2:
                try:
                    set_cnt[data_set[a]] += 1
                except KeyError:
                    set_cnt[data_set[a]] = 1
    for key, cnt in set_cnt.items():
        if cnt >= len(data_str.split("\n")) * min_sup:
            yield key
```

In [10]:
```
# DO NOT EDIT THIS CELL
data = list(gen_record(data_str))
for item in gen_frequent_2_itemset(data, 0.5):
    print(item)
print('No more items')
```

```
('beer', 'rice')
No more items
```

Your output must be:

```
('beer', 'rice')
No more items
```

In [11]:
```
# DO NOT EDIT THIS CELL
data = list(gen_record(data_str))
for item in gen_frequent_2_itemset(data, 0.3):
    print(item)
print('No more items')
```

```
('apple', 'beer')
('beer', 'rice')
('milk', 'beer')
No more items
```

Your output must be:

```
('beer', 'rice')
('beer', 'milk')
('apple', 'beer')
No more items
```

In [12]:
```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
for item in gen_frequent_2_itemset(dataset, 0.2):
    print(item)
print('No more items')
```

```
('chicken', 'beer')
('chicken', 'rice')
('apple', 'beer')
('apple', 'rice')
('beer', 'rice')
('milk', 'beer')
('milk', 'rice')
No more items
```

Your output must be:

```
('chicken', 'rice')
('beer', 'rice')
('beer', 'chicken')
('beer', 'milk')
('milk', 'rice')
('apple', 'rice')
('apple', 'beer')
No more items
```

Ethics:

If you cheat, you will get negatgive of the total points. If the homework total is 22 and you cheat, you get -22.

What to submit

- Run all cells
- Goto "File -> Print Preview"
- Print the page
- Submit in class
- No late homeworks accepted
- Your homework will be graded on the basis of correctness and programming skills

Deadline: 3/18