REPORT

과목명 | **빅데이터최신기술**

담당교수 |

학과 | **소프트웨어학부**

학년 |

학번 | **20181703**

이름 | **평선호**

제출일 |

<후보자와 basket을 만드는 함수>

```python
def filter(candidates,k,s):
    itemsets_cnt_k = {}
    with open ("groceries.csv","r") as f:
        for line in f:
            basket = line.strip().split(',')
            for comb in combinations(basket,k):
                comb = frozenset(comb)
                if comb in candidates:
                    if comb not in itemsets_cnt_k:
                        itemsets_cnt_k[comb] = 0
                    itemsets_cnt_k[comb] +=1
    freq_itemsets = set(itemset for itemset, cnt in itemsets_cnt_k.items() if cnt >= s)

    return freq_itemsets
```

```python
def make_candidate(freq_itemsets,k):
    candidates = set()
    for itemset1 in freq_itemsets:
        for itemset2 in freq_itemsets:
            union = itemset1 | itemset2
            if len(union) == k:
                for item in union:
                    if union - {item} not in freq_itemsets:
                        break
                    else:
                        candidates.add(union)
    return candidates
```

<실행 코드 및 실행 결과>

```
# L1
freq_itemsets = set(frozenset([item]) for item, cnt in item_cnt.items() if cnt >= s)

freq_itemsets_all = freq_itemsets.copy()



candidates_all = set(make_candidate(freq_itemsets,2))

k = 2
while len(freq_itemsets) > 0:
    candidates = make_candidate(freq_itemsets,k)
    candidates_all |= set(candidates)
    freq_itemsets = filter(candidates,k,s)
    freq_itemsets_all |= freq_itemsets
    print(k,len(candidates),len(freq_itemsets))
    k += 1

# association_rule(list(freq_itemsets_all),1,candidates_all)

for fi in freq_itemsets_all:
    print(set(fi))
```

```
{'long life bakery product', 'other vegetables'}
{'pastry', 'whole milk', 'rolls/buns'}
{'bottled water', 'tropical fruit'}
{'salty snack', 'fruit/vegetable juice'}
{'newspapers', 'root vegetables'}
{'margarine', 'other vegetables'}
{'butter', 'whole milk', 'other vegetables'}
{'salty snack', 'other vegetables'}
{'sausage', 'frozen vegetables'}
{'brown bread', 'newspapers'}
{'bottled water'}
{'whole milk', 'citrus fruit', 'other vegetables'}
{'frozen meals', 'other vegetables'}
{'fruit/vegetable juice', 'other vegetables'}
{'whole milk', 'sugar', 'other vegetables'}
{'coffee', 'whole milk'}
{'bottled beer', 'yogurt', 'whole milk'}
{'tropical fruit', 'margarine'}
{'whole milk', 'whipped/sour cream', 'rolls/buns'}
{'long life bakery product', 'root vegetables'}
{'whole milk', 'margarine'}
{'yogurt', 'frankfurter'}
{'salty snack', 'whipped/sour cream'}
{'berries'}
{'domestic eggs', 'whole milk', 'root vegetables'}
{'whole milk', 'pork'}
{'yogurt', 'whole milk', 'other vegetables'}
{'hard cheese', 'rolls/buns'}
```

<Association 함수>

```python
def association_rule(freq_itemsets_all,cnt,candidate_all):

    while(True):
        for i in freq_itemsets_all:
            a = frozenset(list(itertools.combinations(i,cnt)))
            if len(a) == 1:
                continue
            elif (len(a)) != 1:
                a = a - frozenset(i)
            for k in a: # 부분집합
                k = frozenset(k)
                support_i = 0
                support_j = 0
                for j in candidate_all:
                    if k == j.intersection(k):
                        support_i +=1
                    if i == i.intersection(j):
                        support_j +=1
                confidence = support_j/ support_i

                print(f'{set(k)} -> {set(i)} = confidence : {confidence}, support_i = {support_i}, support_j = {support_j}')
        else:
            cnt+=1
```

freq_itemset_all에 있는 집합원소 하나하나들에 대하여 association rule을 적용한다. 이때 집합원소에 대하여 진부분집합을 구한 후 진행하였다.

```
{'brown bread'} -> {'root vegetables', 'brown bread'} = confidence : 0.06956521739130435, support_i = 115, support_j = 8
{'root vegetables'} -> {'root vegetables', 'brown bread'} = confidence : 0.03571428571428571, support_i = 224, support_j = 8
{'whole milk'} -> {'whole milk', 'curd'} = confidence : 0.023715415019762844, support_i = 253, support_j = 6
{'curd'} -> {'whole milk', 'curd'} = confidence : 0.058823529411764705, support_i = 102, support_j = 6
{'citrus fruit'} -> {'citrus fruit', 'tropical fruit'} = confidence : 0.08441558441558442, support_i = 154, support_j = 13
{'tropical fruit'} -> {'citrus fruit', 'tropical fruit'} = confidence : 0.06735751295336788, support_i = 193, support_j = 13
{'whole milk'} -> {'citrus fruit', 'whole milk'} = confidence : 0.05138339920948617, support_i = 253, support_j = 13
{'citrus fruit'} -> {'citrus fruit', 'whole milk'} = confidence : 0.08441558441558442, support_i = 154, support_j = 13
{'soda'} -> {'soda', 'chocolate'} = confidence : 0.018518518518518517, support_i = 216, support_j = 4
{'chocolate'} -> {'soda', 'chocolate'} = confidence : 0.043010752688172046, support_i = 93, support_j = 4
{'newspapers'} -> {'newspapers', 'bottled water'} = confidence : 0.06956521739130435, support_i = 115, support_j = 8
{'bottled water'} -> {'newspapers', 'bottled water'} = confidence : 0.04819277108433735, support_i = 166, support_j = 8
{'whole milk'} -> {'tropical fruit', 'whole milk', 'root vegetables'} = confidence : 0.007905138339920948, support_i = 253, support_j = 2
{'tropical fruit'} -> {'tropical fruit', 'whole milk', 'root vegetables'} = confidence : 0.010362694300518135, support_i = 193, support_j = 2
{'root vegetables'} -> {'tropical fruit', 'whole milk', 'root vegetables'} = confidence : 0.008928571428571428, support_i = 224, support_j = 2
{'pip fruit'} -> {'pip fruit', 'other vegetables'} = confidence : 0.07857142857142857, support_i = 140, support_j = 11
{'other vegetables'} -> {'pip fruit', 'other vegetables'} = confidence : 0.043478260869565216, support_i = 253, support_j = 11
{'tropical fruit'} -> {'tropical fruit', 'soda'} = confidence : 0.08808290155440414, support_i = 193, support_j = 17
{'soda'} -> {'tropical fruit', 'soda'} = confidence : 0.0787037037037037, support_i = 216, support_j = 17
{'whole milk'} -> {'whole milk', 'margarine'} = confidence : 0.02766798418972332, support_i = 253, support_j = 7
{'margarine'} -> {'whole milk', 'margarine'} = confidence : 0.06481481481481481, support_i = 108, support_j = 7
{'whole milk'} -> {'whole milk', 'yogurt', 'other vegetables'} = confidence : 0.019762845849802372, support_i = 253, support_j = 5
{'yogurt'} -> {'whole milk', 'yogurt', 'other vegetables'} = confidence : 0.022026431718061675, support_i = 227, support_j = 5
{'other vegetables'} -> {'whole milk', 'yogurt', 'other vegetables'} = confidence : 0.019762845849802372, support_i = 253, support_j = 5
{'rolls/buns'} -> {'beef', 'rolls/buns'} = confidence : 0.021645021645021644, support_i = 231, support_j = 5
{'beef'} -> {'beef', 'rolls/buns'} = confidence : 0.05154639175257732, support_i = 97, support_j = 5
{'newspapers'} -> {'yogurt', 'newspapers'} = confidence : 0.06956521739130435, support_i = 115, support_j = 8
```

Association rule에 대한 출력물 중 일부분을 가져왔다.

I -> J 라 가정하면, candidate_all에서 i를 가지고 있는 집합을 counting하고 candidate_all에서 j를 가지고 있는 집합을 counting한다. 이후 support_j에서 support_i를 나눈다. 나온 값이 confidence이다.