## Data Mining HW1

**Kaushal Vinay Nerkar**

Colab Notebook Link :
https://colab.research.google.com/drive/1B5h4cYCkmcqQ0g7jLoh4kWP6DwKfrZkE?authuser=1#scrollTo=vwdimfm8wlZj&uniqifier=13

**Question 1**

Load the dataset file groceries.csv.

| | item_list |
|---|---|
| 0 | [citrus fruit, semi-finished bread, margarine,... |
| 1 | [tropical fruit, yogurt, coffee] |
| 2 | [whole milk] |
| 3 | [pip fruit, yogurt, cream cheese , meat spreads] |
| 4 | [other vegetables, whole milk, condensed milk,... |
| ... | ... |
| 9830 | [sausage, chicken, beef, hamburger meat, citru... |
| 9831 | [cooking chocolate] |
| 9832 | [chicken, citrus fruit, other vegetables, butt... |
| 9833 | [semi-finished bread, bottled water, soda, bot... |
| 9834 | [chicken, tropical fruit, other vegetables, vi... |

9835 rows × 1 columns

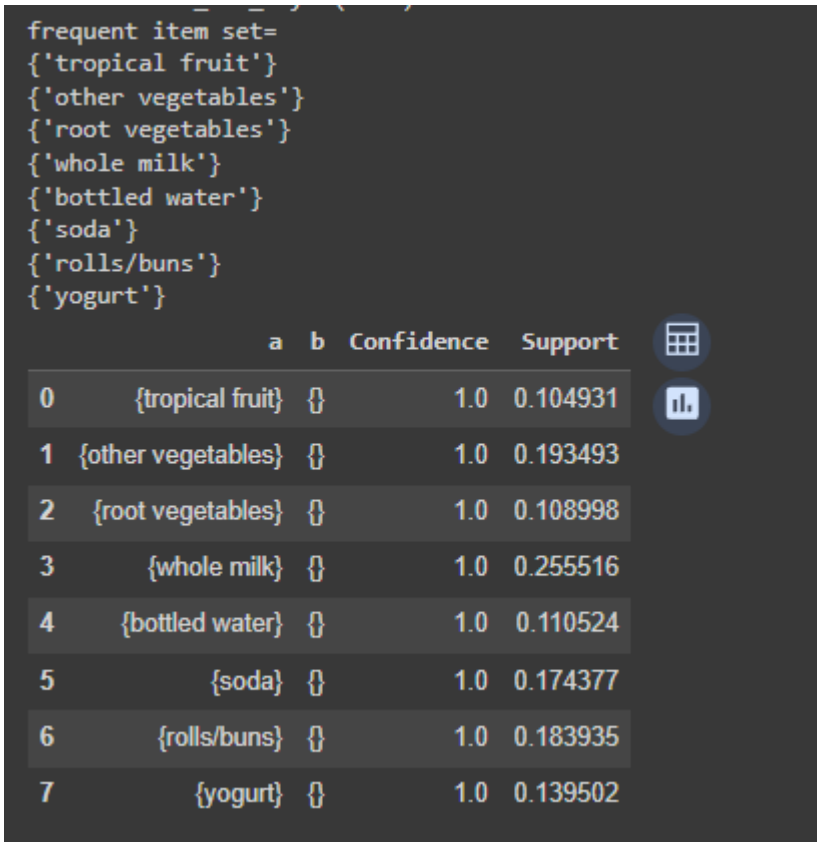
Unique Items or Item List

| | 0 |
|---|---|
| 0 | Instant food products |
| 1 | UHT-milk |
| 2 | abrasive cleaner |
| 3 | artif. sweetener |
| 4 | baby cosmetics |
| ... | ... |
| 164 | white bread |
| 165 | white wine |
| 166 | whole milk |
| 167 | yogurt |
| 168 | zwieback |

169 rows × 1 columns

**Question 2**

Rules and frequent itemset for min_support = 0.1 and Confidence = 0.2



```
frequent item set=
{'tropical fruit'}
{'other vegetables'}
{'root vegetables'}
{'whole milk'}
{'bottled water'}
{'soda'}
{'rolls/buns'}
{'yogurt'}
```

| | a | b | Confidence | Support |
|---|---|---|---|---|
| 0 | {tropical fruit} | {} | 1.0 | 0.104931 |
| 1 | {other vegetables} | {} | 1.0 | 0.193493 |
| 2 | {root vegetables} | {} | 1.0 | 0.108998 |
| 3 | {whole milk} | {} | 1.0 | 0.255516 |
| 4 | {bottled water} | {} | 1.0 | 0.110524 |
| 5 | {soda} | {} | 1.0 | 0.174377 |
| 6 | {rolls/buns} | {} | 1.0 | 0.183935 |
| 7 | {yogurt} | {} | 1.0 | 0.139502 |

**Question 3**

Rules and frequent itemset for min_support = 0.01 and Confidence = 0.2

frequent item set=

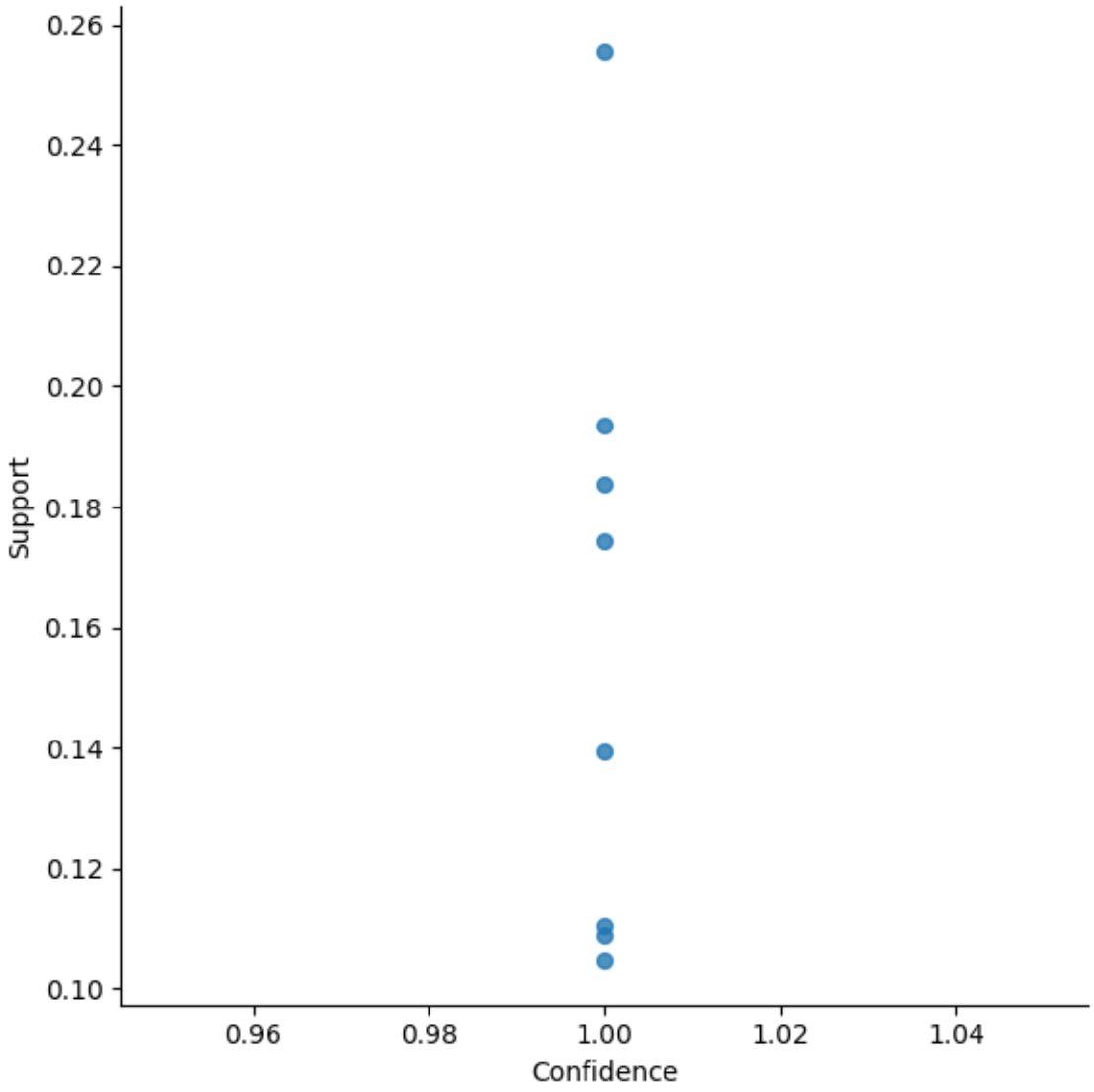
| | itemset |
|---|---|
| 0 | [domestic eggs] |
| 1 | [flower (seeds)] |
| 2 | [cake bar] |
| 3 | [sugar] |
| 4 | [whipped/sour cream] |
| .. | ... |
| 328 | [citrus fruit, root vegetables, other vegetables] |
| 329 | [pork, other vegetables, whole milk] |
| 330 | [root vegetables, other vegetables, yogurt] |
| 331 | [root vegetables, rolls/buns, other vegetables] |
| 332 | [whole milk, rolls/buns, yogurt] |

[333 rows x 1 columns]

| | a | b | Confidence | Support |
|---|---|---|---|---|
| 90 | {napkins} | {whole milk} | 0.376699 | 0.019725 |
| 440 | {frankfurter} | {other vegetables} | 0.279310 | 0.016472 |
| 443 | {domestic eggs} | {rolls/buns} | 0.246795 | 0.015658 |
| 445 | {domestic eggs} | {yogurt} | 0.225962 | 0.014337 |
| 446 | {frozen vegetables} | {rolls/buns} | 0.211416 | 0.010168 |
| ... | ... | ... | ... | ... |
| 287 | {napkins} | {other vegetables} | 0.275728 | 0.014438 |
| 290 | {hamburger meat} | {whole milk} | 0.443425 | 0.014743 |
| 292 | {whipped/sour cream} | {other vegetables} | 0.402837 | 0.028876 |
| 266 | {white bread} | {whole milk} | 0.405797 | 0.017082 |
| 566 | {rolls/buns, whole milk} | {yogurt} | 0.274686 | 0.015557 |

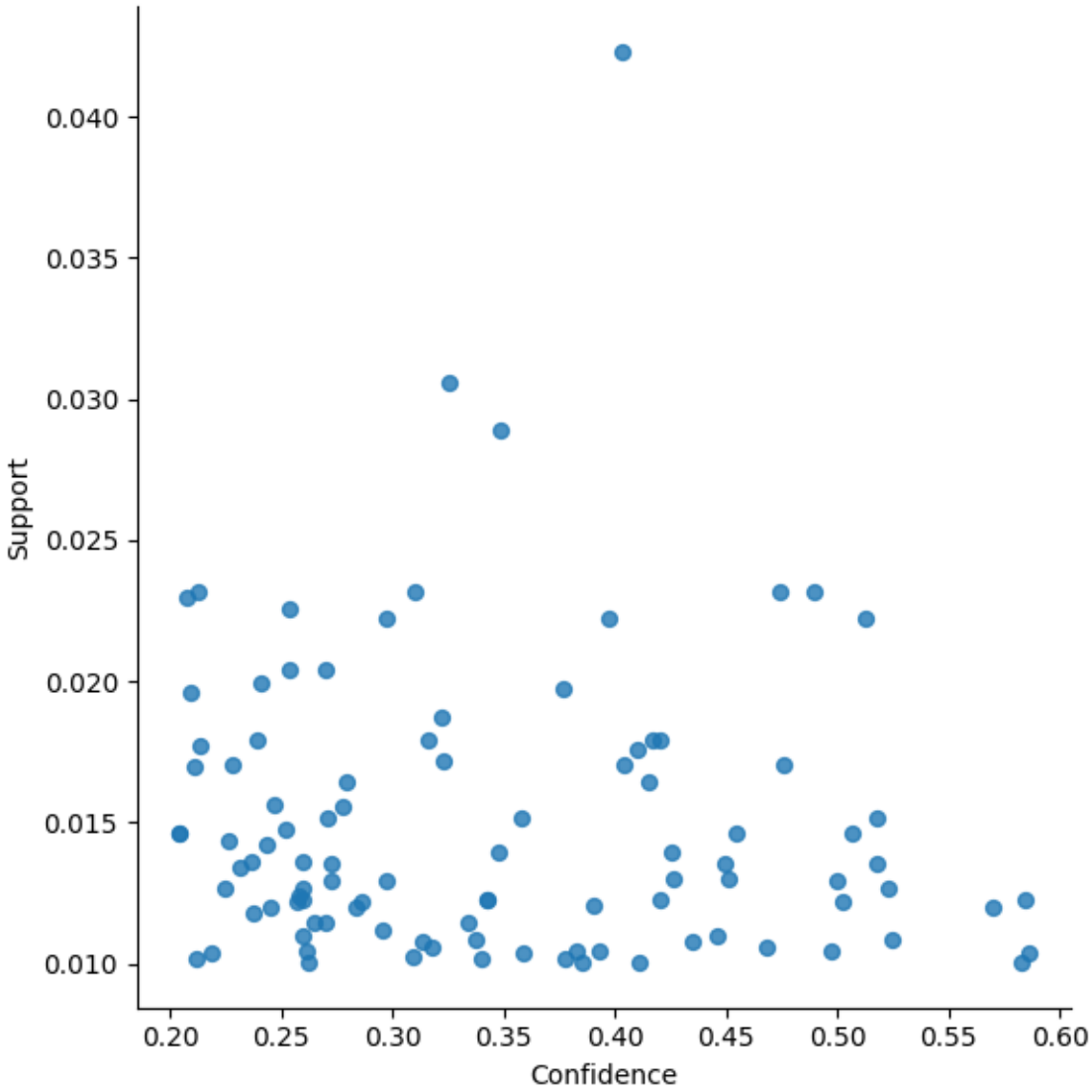
234 rows × 4 columns

**Question 4**




**For min_support = 0.1 and min_confidence = 0.2 from (custom Implementation)**

**Question 5**




for min_support = 0.01 and min_confidence = 0.2

**Question 6**

Using inbuilt Functions mlxtend 0.23.0 Library

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules


df = pd.DataFrame(tn_ary, columns=te.columns_)
#for 0.1 min_sup
frequent_itemsets = apriori(df, min_support=0.1, use_colnames=True)
print("Frequent Itemsets\n", frequent_itemsets)
rules= association_rules(frequent_itemsets, metric="confidence", min_threshold=0.2)
```

```
Frequent Itemsets
     support            itemsets
0  0.110524     (bottled water)
1  0.193493  (other vegetables)
2  0.183935        (rolls/buns)
3  0.108998   (root vegetables)
4  0.174377              (soda)
5  0.104931    (tropical fruit)
6  0.255516        (whole milk)
7  0.139502             (yogurt)
```

```
#for 0.01 supp
frequent_itemsets = apriori(df, min_support=0.01, use_colnames=True)
print("Frequent Itemsets\n", frequent_itemsets)
rules= association_rules(frequent_itemsets, metric="confidence", min_threshold=0.2)
results = pd.DataFrame(rules)
results.head(3000)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the
n during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
Frequent Itemsets
       support                                itemsets
0    0.033452                           (UHT-milk)
1    0.017692                       (baking powder)
2    0.052466                              (beef)
3    0.033249                           (berries)
4    0.026029                         (beverages)
..        ...                                 ...
328  0.011998  (root vegetables, whole milk, tropical fruit)
329  0.014540         (yogurt, root vegetables, whole milk)
330  0.010473                   (yogurt, whole milk, soda)
331  0.015150          (yogurt, whole milk, tropical fruit)
332  0.010880       (yogurt, whipped/sour cream, whole milk)

[333 rows x 2 columns]
```

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction | zhangs_metric |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | (beef) | (other vegetables) | 0.052466 | 0.193493 | 0.019725 | 0.375969 | 1.943066 | 0.009574 | 1.292416 | 0.512224 |
| 1 | (beef) | (rolls/buns) | 0.052466 | 0.183935 | 0.013625 | 0.259690 | 1.411858 | 0.003975 | 1.102329 | 0.307866 |
| 2 | (beef) | (root vegetables) | 0.052466 | 0.108998 | 0.017387 | 0.331395 | 3.040367 | 0.011668 | 1.332628 | 0.708251 |
| 3 | (beef) | (whole milk) | 0.052466 | 0.255516 | 0.021251 | 0.405039 | 1.585180 | 0.007845 | 1.251315 | 0.389597 |
| 4 | (beef) | (yogurt) | 0.052466 | 0.139502 | 0.011693 | 0.222868 | 1.597601 | 0.004374 | 1.107275 | 0.394774 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 229 | (whole milk, yogurt) | (tropical fruit) | 0.056024 | 0.104931 | 0.015150 | 0.270417 | 2.577089 | 0.009271 | 1.226823 | 0.648285 |
| 230 | (yogurt, tropical fruit) | (whole milk) | 0.029283 | 0.255516 | 0.015150 | 0.517361 | 2.024770 | 0.007668 | 1.542528 | 0.521384 |
| 231 | (whole milk, tropical fruit) | (yogurt) | 0.042298 | 0.139502 | 0.015150 | 0.358173 | 2.567516 | 0.009249 | 1.340701 | 0.637483 |
| 232 | (whipped/sour cream, yogurt) | (whole milk) | 0.020742 | 0.255516 | 0.010880 | 0.524510 | 2.052747 | 0.005580 | 1.565719 | 0.523711 |
| 233 | (whipped/sour cream, whole milk) | (yogurt) | 0.032232 | 0.139502 | 0.010880 | 0.337539 | 2.419607 | 0.006383 | 1.298943 | 0.606250 |

234 rows × 10 columns

In build functions are quite faster than my implementation also it shows some info from data like lift , leverage ,conviction etc.