PES University, Bangalore

Established under Karnataka Act No. 16 of 2013

UE22AM343AB4 - Advanced Data Analytics

Designed by Sathwik HJ

Student Details

- Name : **Shusrith S**
- SRN : **PES1UG22AM155**

Market Basket Analysis

```
+ Apriori Algorithm
+ Association Rule mining
```

Prerequisites

- Revise the following concepts
  - Apriori Algorithm
    - Suport
    - Confidence
    - Lift
- Install the following software
  - pandas
  - apyori

Problems

1. Problem 1: Preprocessing
2. Problem 2: Item set detection and analaysis
3. Problem 3: Association rule minning

- Note: Make sure to include justifications for each of your approaches and also provide a brief observations of the results obtained

Context

Welcome to "Shocking Electronics", a budding Electronic Store. As a data analyst working with "Shocking Electronics" , your mission is to uncover meaningful patterns within customer transactions to enhance their shopping experience and help us compete with our competitor "Circuit Circus Electronics".

```
!wget https://raw.githubusercontent.com/MBUYt0n/ada/main/ADA_Worksheet_C/Electr
```

```
--2024-09-20 16:25:37--  https://raw.githubusercontent.com/MBUYt0n/ada/main
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199
HTTP request sent, awaiting response... 200 OK
Length: 9593 (9.4K) [text/plain]
Saving to: ‘Electronics.csv’

Electronics.csv     100%[===================>]   9.37K  --.-KB/s    in 0s

2024-09-20 16:25:37 (103 MB/s) - ‘Electronics.csv’ saved [9593/9593]
```

About the dataset

- "Electronics.csv"
- Each record of the dataset represents a transaction made by a customer at "Shocking Electronics store".
- The transaction contains the items bought in that transaction

```
# Install apyori package
# Use % or ! based on the environment to install
%pip install apyori
%pip install mlxtend
import apyori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=595
  Stored in directory: /root/.cache/pip/wheels/c4/1a/79/20f55c470a50bb3702a
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
Requirement already satisfied: mlxtend in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: scipy>=1.2.1 in /usr/local/lib/python3.10/di
Requirement already satisfied: numpy>=1.16.2 in /usr/local/lib/python3.10/d
Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.10/
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python
Requirement already satisfied: matplotlib>=3.0.0 in /usr/local/lib/python3.
Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.10/
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.1
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/di
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.1
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/pytho
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/di
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/pytho
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-p
```

apyori documentation: https://pypi.org/project/apyori/ (Refer API Usage)

```
import pandas as pd
from apyori import apriori
```

Problem 1 - Preprocessing

Load the transactions data from the provided csv file. **Transform the data** to a suitable format (Hint: **List of lists**[internal list contains the items of the transaction]). **Make sure to clean the data** (Hint: NA values).

```
df = pd.read_csv('Electronics.csv', names=["item1", "item2", "item3", "item4",
df
```

	item1	item2	item3	item4	item5	item6
0	smartphone	earbuds	screen protector	power bank	USB cable	phone case
1	laptop	mouse	keyboard	laptop stand	external hard drive	NaN
2	smartwatch	fitness tracker	wireless charger	NaN	NaN	NaN
3	TV	soundbar	HDMI cable	streaming device	NaN	NaN
4	tablet	stylus	tablet case	screen cleaner	NaN	NaN
...	...	...	...	...	...	...
212	desktop computer	mechanical keyboard	gaming mouse	NaN	NaN	NaN

213	smartwatch	fitness tracker	charging dock	wireless headphones	NaN	NaN
214	laptop	webcam	external hard drive	USB hub	NaN	NaN

```
df.isna().sum()
```

	0
item1	0
item2	0
item3	5
item4	86
item5	213
item6	215

dtype: int64

```
df.fillna("", axis=1, inplace=True)
df
```

	item1	**item2**	**item3**	**item4**	**item5**	**item6**
0	smartphone	earbuds	screen protector	power bank	USB cable	phone case
1	laptop	mouse	keyboard	laptop stand	external hard drive	
2	smartwatch	fitness tracker	wireless charger			
3	TV	soundbar	HDMI cable	streaming device		
4	tablet	stylus	tablet case	screen cleaner		
...	...	...	...	...	...	...
212	desktop computer	mechanical keyboard	gaming mouse			
213	smartwatch	fitness tracker	charging dock	wireless headphones		
214	laptop	webcam	external hard drive	USB hub		

```
import numpy as np
len(set(np.ravel(df.values)))
```

```
    93
```

```
items = df.apply(lambda x: [i for i in x if i != ''], axis=1).tolist()
items
```

```
    [['smartphone',
      'earbuds',
      'screen protector',
      'power bank',
      'USB cable',
      'phone case'],
     ['laptop', 'mouse', 'keyboard', 'laptop stand', 'external hard drive'],
     ['smartwatch', 'fitness tracker', 'wireless charger'],
     ['TV', 'soundbar', 'HDMI cable', 'streaming device'],
     ['tablet', 'stylus', 'tablet case', 'screen cleaner'],
     ['smartphone', 'smartwatch', 'earbuds'],
     ['gaming console', 'game controller', 'HDMI cable', 'gaming headset'],
     ['laptop', 'wireless mouse', 'backpack', 'laptop sleeve'],
     ['smartphone', 'phone case', 'screen protector', 'USB cable'],
     ['smartphone', 'smartwatch', 'power bank'],
     ['laptop', 'external hard drive', 'USB hub'],
     ['TV', 'soundbar', 'HDMI cable'],
     ['smartphone', 'earbuds', 'power bank'],
     ['smartwatch', 'fitness tracker', 'charging dock'],
     ['gaming console', 'game controller', 'gaming headset'],
     ['laptop', 'keyboard', 'mouse', 'monitor'],
     ['tablet', 'stylus', 'tablet case'],
     ['smartphone', 'screen protector', 'phone case'],
     ['TV', 'streaming device', 'HDMI cable', 'soundbar'],
     ['smartphone', 'earbuds', 'wireless charger', 'power bank'],
     ['smartwatch', 'fitness tracker'],
     ['laptop', 'external hard drive', 'wireless mouse'],
     ['smartphone', 'screen protector', 'USB cable', 'phone case'],
     ['TV', 'soundbar', 'streaming device'],
     ['gaming console', 'gaming headset', 'game controller'],
     ['laptop', 'laptop sleeve', 'backpack'],
     ['tablet', 'stylus', 'screen cleaner'],
     ['smartwatch', 'charging dock', 'fitness tracker'],
     ['smartphone', 'phone case', 'USB cable', 'power bank'],
     ['laptop', 'wireless mouse', 'keyboard', 'monitor'],
     ['TV', 'HDMI cable', 'soundbar'],
     ['smartphone', 'screen protector', 'phone case', 'earbuds'],
     ['gaming console', 'gaming headset', 'HDMI cable'],
     ['tablet', 'tablet case', 'stylus'],
     ['smartphone', 'screen protector', 'USB cable'],
     ['laptop', 'external hard drive', 'wireless mouse', 'USB hub'],
     ['TV', 'streaming device', 'soundbar', 'HDMI cable'],
     ['smartwatch', 'fitness tracker', 'charging dock'],
     ['smartphone', 'earbuds', 'phone case'],
     ['laptop', 'monitor', 'keyboard', 'wireless mouse'],
     ['gaming console', 'gaming headset'],
     ['tablet', 'screen cleaner', 'stylus'],
     ['TV', 'soundbar', 'HDMI cable'],
     ['smartphone', 'screen protector', 'power bank'],
     ['laptop', 'external hard drive', 'USB hub'],
     ['smartwatch', 'wireless charger', 'fitness tracker'],
     ['smartphone',
      'earbuds'
```

```
   'earbuds',
   'screen protector',
   'power bank',
   'USB cable',
   'phone case'],
  ['laptop', 'mouse', 'keyboard', 'laptop stand', 'external hard drive'],
```

Problem 2 - Item sets

1. Print out the frequent item sets along with their support values also display the count of item sets.(**min_support=0.08**)
2. "Shocking Electronics" wants to replenish its stocks, help find the top 5 most popular(higher buying frequency) items/item_sets to replenish. **Explain and justify** the process followed to come to the conclusion.

```
from apyori import apriori

frequent_itemsets = apriori(items, min_support=0.08)

frequent_itemsets_list = list(frequent_itemsets)
for itemset in frequent_itemsets_list:
    print(f"Itemset: {itemset.items}, Support: {itemset.support}")
```

```
Itemset: frozenset({'HDMI cable'}), Support: 0.15668202764976957
Itemset: frozenset({'TV'}), Support: 0.1382488479262673
Itemset: frozenset({'earbuds'}), Support: 0.1152073732718894
Itemset: frozenset({'external hard drive'}), Support: 0.1382488479262673
Itemset: frozenset({'fitness tracker'}), Support: 0.11981566820276497
Itemset: frozenset({'gaming console'}), Support: 0.12903225806451613
Itemset: frozenset({'gaming headset'}), Support: 0.08755760368663594
Itemset: frozenset({'laptop'}), Support: 0.1889400921658986
Itemset: frozenset({'phone case'}), Support: 0.09216589861751152
Itemset: frozenset({'screen protector'}), Support: 0.15207373271889402
Itemset: frozenset({'smartphone'}), Support: 0.2350230414746544
Itemset: frozenset({'smartwatch'}), Support: 0.14285714285714285
Itemset: frozenset({'soundbar'}), Support: 0.12442396313364056
Itemset: frozenset({'streaming device'}), Support: 0.0967741935483871
Itemset: frozenset({'stylus'}), Support: 0.09216589861751152
Itemset: frozenset({'tablet'}), Support: 0.08755760368663594
Itemset: frozenset({'wireless mouse'}), Support: 0.09216589861751152
Itemset: frozenset({'TV', 'soundbar'}), Support: 0.12442396313364056
Itemset: frozenset({'TV', 'streaming device'}), Support: 0.0967741935483871
Itemset: frozenset({'smartphone', 'earbuds'}), Support: 0.10138248847926268
Itemset: frozenset({'laptop', 'external hard drive'}), Support: 0.133640552
Itemset: frozenset({'fitness tracker', 'smartwatch'}), Support: 0.115207373
Itemset: frozenset({'gaming console', 'gaming headset'}), Support: 0.082949
Itemset: frozenset({'laptop', 'wireless mouse'}), Support: 0.09216589861751
Itemset: frozenset({'smartphone', 'phone case'}), Support: 0.09216589861751
Itemset: frozenset({'smartphone', 'screen protector'}), Support: 0.11981566
Itemset: frozenset({'streaming device', 'soundbar'}), Support: 0.0921658986
Itemset: frozenset({'tablet', 'stylus'}), Support: 0.08294930875576037
Itemset: frozenset({'TV', 'streaming device', 'soundbar'}), Support: 0.0921
```

```
item_support = {}
for itemset in frequent_itemsets_list:
    for item in itemset.items:
        item_support[item] = item_support.get(item, 0) + itemset.support

top_items = sorted(item_support.items(), key=lambda x: x[1], reverse=True)[:5]

print("Top 5 items to replenish:")
for item, support in top_items:
    print(f"Item: {item}, Support: {support}")
```

```
Top 5 items to replenish:
Item: smartphone, Support: 0.5483870967741935
Item: TV, Support: 0.45161290322580644
Item: soundbar, Support: 0.43317972350230416
Item: laptop, Support: 0.4147465437788018
Item: streaming device, Support: 0.3778801843317972
```

These are the top 5 items to replenish.

The process to identify popular itemsets for replenishment is as follows:

1. **Frequent Itemset Mining:** The Apriori algorithm, with a minimum support threshold of 0.08, identified itemsets appearing in at least 8% of transactions.
2. **Support Calculation:** Support values, indicating the percentage of transactions containing each itemset, were computed.
3. **Sorting by Support:** Itemsets were sorted by support in descending order to prioritize popularity.
4. **Top 5 Selection:** The top 5 itemsets with the highest support were chosen for replenishment.

This approach ensures that "Shocking Electronics" stocks the most popular products and combinations, maximizing customer satisfaction and sales.

Problem 3 - Association Rules

- Items on the left side of the association rule are called : **Antecedent items** and the right side of the association rule are called : **Consequent Items**.

1. Print out the association rules along with their confidence and lift. (Analyse the output structure of apriori())
    - **(min_support=0.01, min_confidence = 0.08, min_lift=1.7, min_length=3)**
2. As the festive season is approaching, "Shocking Electronics" is considering to provide discounts and offers on some of their products. Help them identify the top 5 popular **pairs/sets** of items/item_sets bought, considering probability of consequent item being

purchased when antecedent item is bought.

3. Also help them identify the top 5 popular **pairs/sets** of items/item_sets bought together, considering the popularity of consequent and antecedent items.

- (Consequent and antecedent items together form the **pairs/sets** specified in the question)

```
# Let's get association rules along with confidence and lift
association_rules = []
for result in frequent_itemsets_list:
    for ordered_statistic in result.ordered_statistics:
        antecedent = list(ordered_statistic.items_base)
        consequent = list(ordered_statistic.items_add)
        support = result.support
        confidence = ordered_statistic.confidence
        lift = ordered_statistic.lift

        association_rules.append((antecedent, consequent, support, confidence,

rules_df = pd.DataFrame(association_rules, columns=['Antecedent', 'Consequent',
rules_df
```

	Antecedent	Consequent	Support	Confidence	Lift
0	[]	[HDMI cable]	0.156682	0.156682	1.000000
1	[]	[TV]	0.138249	0.138249	1.000000
2	[]	[earbuds]	0.115207	0.115207	1.000000
3	[]	[external hard drive]	0.138249	0.138249	1.000000
4	[]	[fitness tracker]	0.119816	0.119816	1.000000
5	[]	[gaming console]	0.129032	0.129032	1.000000
6	[]	[gaming headset]	0.087558	0.087558	1.000000
7	[]	[laptop]	0.188940	0.188940	1.000000
8	[]	[phone case]	0.092166	0.092166	1.000000
9	[]	[screen protector]	0.152074	0.152074	1.000000
10	[]	[smartphone]	0.235023	0.235023	1.000000
11	[]	[smartwatch]	0.142857	0.142857	1.000000
12	[]	[soundbar]	0.124424	0.124424	1.000000
13	[]	[streaming device]	0.096774	0.096774	1.000000
14	[]	[stylus]	0.092166	0.092166	1.000000
15	[]	[tablet]	0.087558	0.087558	1.000000
16	[]	[wireless mouse]	0.092166	0.092166	1.000000

17	[]	[TV, soundbar]	0.124424	0.124424	1.000000
18	[TV]	[soundbar]	0.124424	0.900000	7.233333
19	[soundbar]	[TV]	0.124424	1.000000	7.233333
20	[]	[TV, streaming device]	0.096774	0.096774	1.000000
21	[TV]	[streaming device]	0.096774	0.700000	7.233333
22	[streaming device]	[TV]	0.096774	1.000000	7.233333
23	[]	[smartphone, earbuds]	0.101382	0.101382	1.000000
24	[earbuds]	[smartphone]	0.101382	0.880000	3.744314
25	[smartphone]	[earbuds]	0.101382	0.431373	3.744314
26	[]	[laptop, external hard drive]	0.133641	0.133641	1.000000
27	[external hard drive]	[laptop]	0.133641	0.966667	5.116260
28	[laptop]	[external hard drive]	0.133641	0.707317	5.116260
29	[]	[fitness tracker, smartwatch]	0.115207	0.115207	1.000000
30	[fitness tracker]	[smartwatch]	0.115207	0.961538	6.730769
31	[smartwatch]	[fitness tracker]	0.115207	0.806452	6.730769
32	[]	[gaming console, gaming headset]	0.082949	0.082949	1.000000
33	[gaming console]	[gaming headset]	0.082949	0.642857	7.342105
34	[gaming headset]	[gaming console]	0.082949	0.947368	7.342105
35	[]	[laptop, wireless mouse]	0.092166	0.092166	1.000000
36	[laptop]	[wireless mouse]	0.092166	0.487805	5.292683
37	[wireless mouse]	[laptop]	0.092166	1.000000	5.292683
38	[]	[smartphone, phone case]	0.092166	0.092166	1.000000
39	[phone case]	[smartphone]	0.092166	1.000000	4.254902
40	[smartphone]	[phone case]	0.092166	0.392157	4.254902
41	[]	[smartphone, screen protector]	0.119816	0.119816	1.000000
42	[screen protector]	[smartphone]	0.119816	0.787879	3.352347
43	[smartphone]	[screen protector]	0.119816	0.509804	3.352347
44	[]	[soundbar, streaming device]	0.092166	0.092166	1.000000
45	[soundbar]	[streaming device]	0.092166	0.740741	7.654321

46	[streaming device]	[soundbar]	0.092166	0.952381	7.654321
47	[]	[tablet, stylus]	0.082949	0.082949	1.000000
48	[stylus]	[tablet]	0.082949	0.900000	10.278947
49	[tablet]	[stylus]	0.082949	0.947368	10.278947
50	[]	[TV, soundbar, streaming device]	0.092166	0.092166	1.000000
51	[TV]	[soundbar, streaming device]	0.092166	0.666667	7.233333
52	[soundbar]	[TV, streaming device]	0.092166	0.740741	7.654321
53	[streaming device]	[TV, soundbar]	0.092166	0.952381	7.654321

```
min_confidence = 0.08
min_lift = 1.7
filtered_rules_df = rules_df[(rules_df['Confidence'] >= min_confidence) & (rule
filtered_rules_df
```

	Antecedent	Consequent	Support	Confidence	Lift
18	[TV]	[soundbar]	0.124424	0.900000	7.233333
19	[soundbar]	[TV]	0.124424	1.000000	7.233333
21	[TV]	[streaming device]	0.096774	0.700000	7.233333
22	[streaming device]	[TV]	0.096774	1.000000	7.233333
24	[earbuds]	[smartphone]	0.101382	0.880000	3.744314
25	[smartphone]	[earbuds]	0.101382	0.431373	3.744314
27	[external hard drive]	[laptop]	0.133641	0.966667	5.116260
28	[laptop]	[external hard drive]	0.133641	0.707317	5.116260
30	[fitness tracker]	[smartwatch]	0.115207	0.961538	6.730769
31	[smartwatch]	[fitness tracker]	0.115207	0.806452	6.730769
33	[gaming console]	[gaming headset]	0.082949	0.642857	7.342105
34	[gaming headset]	[gaming console]	0.082949	0.947368	7.342105
36	[laptop]	[wireless mouse]	0.092166	0.487805	5.292683
37	[wireless mouse]	[laptop]	0.092166	1.000000	5.292683
39	[phone case]	[smartphone]	0.092166	1.000000	4.254902
40	[smartphone]	[phone case]	0.092166	0.392157	4.254902
42	[screen protector]	[smartphone]	0.119816	0.787879	3.352347
43	[smartphone]	[screen protector]	0.119816	0.509804	3.352347
45	[soundbar]	[streaming device]	0.092166	0.740741	7.654321
46	[streaming device]	[soundbar]	0.092166	0.952381	7.654321

48	[stylus]	[tablet]	0.082949	0.900000	10.278947
49	[tablet]	[stylus]	0.082949	0.947368	10.278947
51	[TV]	[soundbar, streaming device]	0.092166	0.666667	7.233333
52	[soundbar]	[TV, streaming device]	0.092166	0.740741	7.654321
53	[streaming device]	[TV, soundbar]	0.092166	0.952381	7.654321
54	[TV, soundbar]	[streaming device]	0.092166	0.740741	7.654321
55	[TV, streaming device]	[soundbar]	0.092166	0.952381	7.654321
56	[streaming device, soundbar]	[TV]	0.092166	1.000000	7.233333

```
rules = []
for itemset in frequent_itemsets_list:
    for ordered_stat in itemset.ordered_statistics:
        if len(ordered_stat.items_base) > 0 and len(ordered_stat.items_add) > 0
            rules.append({
                'antecedent': tuple(ordered_stat.items_base),
                'consequent': tuple(ordered_stat.items_add),
                'support': itemset.support,
                'confidence': ordered_stat.confidence,
                'lift': ordered_stat.lift
            })

rules_sorted_by_confidence = sorted(rules, key=lambda x: x['confidence'], rever
rules_sorted_by_lift = sorted(rules, key=lambda x: x['lift'], reverse=True)[:5]

print("Top 5 popular pairs/sets based on confidence:")
for rule in rules_sorted_by_confidence:
    print(f"Antecedent: {rule['antecedent']}, Consequent: {rule['consequent']},

print("\nTop 5 popular pairs/sets based on lift:")
for rule in rules_sorted_by_lift:
    print(f"Antecedent: {rule['antecedent']}, Consequent: {rule['consequent']},
```

```
Top 5 popular pairs/sets based on confidence:
Antecedent: ('soundbar',), Consequent: ('TV',), Confidence: 1.00, Lift: 7.2
Antecedent: ('streaming device',), Consequent: ('TV',), Confidence: 1.00, L
Antecedent: ('wireless mouse',), Consequent: ('laptop',), Confidence: 1.00,
Antecedent: ('phone case',), Consequent: ('smartphone',), Confidence: 1.00,
Antecedent: ('streaming device', 'soundbar'), Consequent: ('TV',), Confiden

Top 5 popular pairs/sets based on lift:
Antecedent: ('stylus',), Consequent: ('tablet',), Confidence: 0.90, Lift: 1
Antecedent: ('tablet',), Consequent: ('stylus',), Confidence: 0.95, Lift: 1
Antecedent: ('streaming device',), Consequent: ('soundbar',), Confidence: 0
Antecedent: ('streaming device',), Consequent: ('TV', 'soundbar'), Confiden
Antecedent: ('TV', 'streaming device'), Consequent: ('soundbar',), Confiden
```

Problem 4 - Theory

Explain **Rule**, **Lift** and **Support**, taking examples from the results obtained from **your analysis**

1. **Rule**:

    - An **association rule** is a relationship between two or more items. It helps in discovering how items are associated with one another in a dataset. A rule is structured in the form `Antecedent → Consequent`, where the antecedent is the item(s) purchased first, and the consequent is the item(s) purchased after.

    Example:

    - **Rule**: `('soundbar',) → ('TV',)`
        - This rule indicates that customers who buy a soundbar also tend to buy a TV.

2. **Support**:

    - **Support** measures the frequency of occurrence of an itemset in the dataset. It is the ratio of transactions that contain the itemset to the total number of transactions. It helps determine how popular an itemset is.

Example:

- If the **support** for the rule `('soundbar',) → ('TV',)` is `0.15`, it means that 15% of all transactions include both a soundbar and a TV.

3. **Lift**:

    - **Lift** measures how much more likely the consequent is to be purchased when the antecedent is purchased, compared to the chance of purchasing the consequent independently. It indicates the strength of the association.

    - A **lift** greater than 1 indicates a strong positive association between the antecedent and the consequent.

    Example:

    - **Lift** for the rule `('soundbar',) → ('TV',)` is `7.23`, meaning that customers who buy a soundbar are 7.23 times more likely to buy a TV compared to customers who don't buy a soundbar.

```
Start coding or generate with AI.
```