

# Market Basket Recomender Analysis - Apriori

Patrick Herlihy

01/30/23

The goal of this analysis is to uncover hidden relationships between items. Knowing which items are commonly bought together can help us recommend additional items to customers which might be useful to their purchases. Market basket analysis uses data mining to define the strength of a relationship between items commonly grouped together and the frequency which they occur together. These metrics of strength can then be used to determine association rules to analyze transaction data. These rules can then be used to pair items which are most likely to be grouped together.

Dataset:

```
In [2]:  
import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
from mlxtend.preprocessing import TransactionEncoder  
from mlxtend.frequent_patterns import apriori  
from mlxtend.frequent_patterns import association_rules  
from pandas.plotting import parallel_coordinates
```

## Data Cleaning/Preparation Steps

- Load data into a Pandas dataframe
- Review the head of all the data, data types
- Check for rows entirely filled with null values and remove
- Create a 2D Array for all purchases by each customer
- Use Transaction Encoder to encode whether or not (True/False) a customer purchased each item
- Check for any empty column items and remove
- Export the cleaned/prepared dataset to csv

```
In [3]: purchases=pd.read_csv( '/Users/herlihpj/Desktop/Data Analytics/D212 Data Mining II/Task 3/Teleco/teleco_market_basket.csv' )
```

```
In [4]: purchases.info()
purchases.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15002 entries, 0 to 15001
Data columns (total 20 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   Item01    7501 non-null   object  
 1   Item02    5747 non-null   object  
 2   Item03    4389 non-null   object  
 3   Item04    3345 non-null   object  
 4   Item05    2529 non-null   object  
 5   Item06    1864 non-null   object  
 6   Item07    1369 non-null   object  
 7   Item08    981 non-null    object  
 8   Item09    654 non-null    object  
 9   Item10    395 non-null    object  
 10  Item11    256 non-null    object  
 11  Item12    154 non-null    object  
 12  Item13    87 non-null    object  
 13  Item14    47 non-null    object  
 14  Item15    25 non-null    object  
 15  Item16    8 non-null     object  
 16  Item17    4 non-null     object  
 17  Item18    4 non-null     object  
 18  Item19    3 non-null     object  
 19  Item20    1 non-null     object  
dtypes: object(20)
memory usage: 2.3+ MB
```

Out[4]:	Item01	Item02	Item03	Item04	Item05	Item06	Item07	Item08	Item09	Item10	Item11	Item12	Item13	Item14
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Logitech M510 Wireless mouse	HP 63 Ink	HP 65 ink	nonda USB C to USB Adapter	10ft iPHONE Charger Cable	HP 902XL ink	Creative Pebble 2.0 Speakers	Cleaning Gel Universal Dust Cleaner	Micro Center 32GB Memory card	YUNSONG 3pack 6ft Nylon Lightning Cable	TopMate C5 Laptop Cooler pad	Apple USB-C Charger cable	HyperX Cloud Stinger Headset	TONOR USB Gaming Microphone
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Apple Lightning to Digital AV Adapter	TP-Link AC1750 Smart WiFi Router	Apple Pencil	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [5]: `purchases.dropna(how='all', inplace=True)`  
`purchases.info()`  
`purchases.head()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7501 entries, 1 to 15001
Data columns (total 20 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   Item01    7501 non-null    object  
 1   Item02    5747 non-null    object  
 2   Item03    4389 non-null    object  
 3   Item04    3345 non-null    object  
 4   Item05    2529 non-null    object  
 5   Item06    1864 non-null    object  
 6   Item07    1369 non-null    object  
 7   Item08    981 non-null    object  
 8   Item09    654 non-null    object  
 9   Item10    395 non-null    object  
 10  Item11    256 non-null    object  
 11  Item12    154 non-null    object  
 12  Item13    87 non-null    object  
 13  Item14    47 non-null    object  
 14  Item15    25 non-null    object  
 15  Item16    8 non-null    object  
 16  Item17    4 non-null    object  
 17  Item18    4 non-null    object  
 18  Item19    3 non-null    object  
 19  Item20    1 non-null    object  
dtypes: object(20)
memory usage: 1.2+ MB
```

Out[5]:	Item01	Item02	Item03	Item04	Item05	Item06	Item07	Item08	Item09	Item10	Item11	Item12	Item13	Item14
1	Logitech M510 Wireless mouse	HP 63 Ink	HP 65 ink	nonda USB C to USB Adapter	10ft iPhone Charger Cable	HP 902XL ink	Creative Pebble 2.0 Speakers	Cleaning Gel Universal Dust Cleaner	Micro Center 32GB Memory card	YUNSONG 3pack 6ft Nylon Lightning Cable	TopMate C5 Laptop Cooler pad	Apple USB-C Charger cable	HyperX Cloud Stinger Headset	TONOR USB Gaming Microphone
3	Apple Lightning to Digital AV Adapter	TP-Link AC1750 Smart WiFi Router	Apple Pencil	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	UNEN Mfi Certified 5-pack Lightning Cable	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	Cat8 Ethernet Cable	HP 65 ink	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	Dust-Off Compressed Gas 2 pack	Screen Mom Screen Cleaner <small>Lit+</small>	Moread HDMI to VGA Adapter	HP 62XL Tri-Color ink	Apple USB-C Charger cable	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [6]:

```
num_rows=purchases.shape[0]
num_cols=purchases.shape[1]
purchases.shape
```

Out[6]:

```
(7501, 20)
```

In [7]:

```
items = []
for i in range (0, num_rows):
    items.append([str(purchases.values[i,j]) for j in range(0, num_cols)])
#print second transaction for the assessment
print('Second transaction in the set: ', items[1])
```

Second transaction in the set: ['Apple Lightning to Digital AV Adapter', 'TP-Link AC1750 Smart WiFi Router', 'Apple Pencil', 'nan', 'nan']

```
In [8]: encoder=TransactionEncoder()
onehot=encoder.fit(items).transform(items)
prepared_df=pd.DataFrame(onehot, columns=encoder.columns_)
prepared_df
```

Out[8]:

	10ft iPhone Charger Cable	10ft iPhone Charger Cable 2 Pack	3 pack Nylon Braided Lightning Cable	3A USB Type C Cable 3 pack 6FT	5pack Nylon Braided USB C cables	ARRIS SURFboard SB8200 Cable Modem	Anker 2-in-1 USB Card Reader	Anker 4-port USB hub	Anker USB C to HDMI Adapter	Apple Lightning to Digital AV Adapter	...	iFixit Pro Tech Toolkit	iPhone 11 case	iPhone 12 Charger cable	iPhone 12 Pro case	iPhone 13 Pro case
0	True	False	False	True	False	False	False	False	False	False	...	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	True	...	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7496	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
7497	False	False	False	False	False	True	False	False	False	True	...	False	False	False	False	False
7498	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
7499	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
7500	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False

7501 rows × 120 columns

```
In [9]: #Check for empty columns, each item is now a column after encoding
for col in prepared_df.columns:
    print(col)
#one column Labeled 'nan'
#Drop the empty column
df_cleaned=prepared_df.drop(['nan'], axis=1)
print('Shape of cleaned data: ', df_cleaned.shape)
```

10ft iPHONE Charger Cable  
10ft iPHONE Charger Cable 2 Pack  
3 pack Nylon Braided Lightning Cable  
3A USB Type C Cable 3 pack 6FT  
5pack Nylon Braided USB C cables  
ARRIS SURFboard SB8200 Cable Modem  
Anker 2-in-1 USB Card Reader  
Anker 4-port USB hub  
Anker USB C to HDMI Adapter  
Apple Lightning to Digital AV Adapter  
Apple Lightning to USB cable  
Apple Magic Mouse 2  
Apple Pencil  
Apple Pencil 2nd Gen  
Apple Power Adapter Extension Cable  
Apple USB-C Charger cable  
AutoFocus 1080p Webcam  
BENGOO G90000 headset  
Blue Light Blocking Glasses  
Blue Light Blocking Glasses 2pack  
Brother Genuine High Yield Toner Cartridge  
Cat 6 Ethernet Cable 50ft  
Cat8 Ethernet Cable  
CicTsing MM057 2.4G Wireless Mouse  
Cleaning Gel Universal Dust Cleaner  
Creative Pebble 2.0 Speakers  
DisplayPort ot HDMI adapter  
Dust-Off Compressed Gas  
Dust-Off Compressed Gas 2 pack  
FEEL2NICE 5 pack 10ft Lightning cable  
FEIYOLD Blue light Blocking Glasses  
Falcon Dust Off Compressed Gas  
HOVAMP Mfi 6pack Lightning Cable  
HOVAMP iPhone charger  
HP 61 2 pack ink  
HP 61 Tri-color ink  
HP 61 ink  
HP 62XL Tri-Color ink  
HP 62XL ink  
HP 63 Ink  
HP 63 Tri-color ink  
HP 63XL Ink  
HP 63XL Tri-color ink  
HP 64 Tri-Color ink  
HP 64 ink

HP 65 ink  
HP 902XL ink  
HP 952 ink  
HP ENVY 5055 printer  
HP952XL ink  
HooToo USB C Hub  
HyperX Cloud Stinger Headset  
Jelly Comb 2.4G Slim Wireless mouse  
Leader Desk Pad Protector  
Logitech M510 Wireless mouse  
Logitech MK270 Wireless Keyboard/Mouse  
Logitech MK345 Wireless combo  
Logitech USB H390 headset  
M.2 Screw kit  
Mfi-Certified Lightning to USB A Cable  
Micro Center 32GB Memory card  
Microsot Surface Dock 2  
Moread HDMI to VGA Adapter  
Mpow HC6 USB Headset  
NETGEAR CM500 Cable Modem  
NETGEAR Nighthawk WiFi Router  
NETGEAR Orbi Home Mesh WiFi System  
Nylon Braided Lightning to USB cable  
PS4 Headset  
Premium Nylon USB Cable  
RUNMUS Gaming Headset  
SAMSUNG 128GB card  
SAMSUNG 256 GB card  
SAMSUNG EVO 32GB card  
SAMSUNG EVO 64GB card  
Sabrent 4-port USB 3.0 hub  
SanDisk 128GB Ultra microSDXC card  
SanDisk 128GB card  
SanDisk 128GB microSDXC card  
SanDisk 32GB Ultra SDHC card  
SanDisk 32GB card  
SanDisk Extreme 128GB card  
SanDisk Extreme 256GB card  
SanDisk Extreme 32GB 2pack card  
SanDisk Extreme Pro 128GB card  
SanDisk Extreme Pro 64GB card  
SanDisk Ultra 128GB card  
SanDisk Ultra 256GB card  
SanDisk Ultra 400GB card  
SanDisk Ultra 64GB card

Screen Mom Screen Cleaner kit  
Stylus Pen for iPad  
Syntech USB C to USB Adapter  
TONOR USB Gaming Microphone  
TP-Link AC1750 Smart WiFi Router  
TP-Link AC4000 WiFi router  
TopMate C5 Laptop Cooler pad  
UNEN Mfi Certified 5-pack Lightning Cable  
USB 2.0 Printer cable  
USB C to USB Male Adapter  
USB Type C Cable  
USB Type C to USB-A Charger cable  
VIVO Dual LCD Monitor Desk mount  
VicTsing Mouse Pad  
VicTsing Wireless mouse  
Vsco 70 pack stickers  
Webcam with Microphone  
XPOWER A-2 Air Pump blower  
YUNSONG 3pack 6ft Nylon Lightning Cable  
hP 65 Tri-color ink  
iFixit Pro Tech Toolkit  
iPhone 11 case  
iPhone 12 Charger cable  
iPhone 12 Pro case  
iPhone 12 case  
iPhone Charger Cable Anker 6ft  
iPhone SE case  
nan  
nonda USB C to USB Adapter  
seenda Wireless mouse  
Shape of cleaned data: (7501, 119)

```
In [10]: #Export cleaned to csv
df_cleaned.to_csv('/Users/herlihpj/Desktop/Data Analytics/D212 Data Mining II/Task 3/Teleco/prepared.csv')
print('Prepared Data has been exported to CSV')
```

Prepared Data has been exported to CSV

## Begin Analysis

### 1. Runs the Apriori Function to mine the rules on the support metric and review info

```
In [11]: support_rules=apriori(df_cleaned, min_support=0.05, use_colnames=True)
support_rules.head(20)
```

Out[11]:

	<b>support</b>	<b>itemsets</b>
<b>0</b>	0.050527	(10ft iPHONE Charger Cable 2 Pack)
<b>1</b>	0.068391	(Anker USB C to HDMI Adapter)
<b>2</b>	0.087188	(Apple Lightning to Digital AV Adapter)
<b>3</b>	0.179709	(Apple Pencil)
<b>4</b>	0.132116	(Apple USB-C Charger cable)
<b>5</b>	0.062525	(Cat8 Ethernet Cable)
<b>6</b>	0.238368	(Dust-Off Compressed Gas 2 pack)
<b>7</b>	0.065858	(FEIYOLD Blue light Blocking Glasses)
<b>8</b>	0.059992	(Falcon Dust Off Compressed Gas)
<b>9</b>	0.163845	(HP 61 ink)
<b>10</b>	0.058526	(HP 62XL Tri-Color ink)
<b>11</b>	0.079323	(HP 63XL Ink)
<b>12</b>	0.071457	(Logitech M510 Wireless mouse)
<b>13</b>	0.095321	(Nylon Braided Lightning to USB cable)
<b>14</b>	0.051060	(Premium Nylon USB Cable)
<b>15</b>	0.052393	(SAMSUNG EVO 32GB card)
<b>16</b>	0.063325	(SanDisk Ultra 128GB card)
<b>17</b>	0.098254	(SanDisk Ultra 64GB card)
<b>18</b>	0.129583	(Screen Mom Screen Cleaner kit)
<b>19</b>	0.095054	(Stylus Pen for iPad)

## 2. Set the additional metrics and filter the results by those rules

In [16]:

```
ass_r=association_rules(support_rules, metric='lift', min_threshold=1)
ass_r.head(10)
#Strong indication that we can reccomend 'Dust-Off Compressed Gas 2 pack'
#to users buying apple pencil, Hp 61 or Vivo dual LCD monitor
```

Out[16]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Apple Pencil)	(Dust-Off Compressed Gas 2 pack)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815
1	(Dust-Off Compressed Gas 2 pack)	(Apple Pencil)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158
2	(HP 61 ink)	(Dust-Off Compressed Gas 2 pack)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
3	(Dust-Off Compressed Gas 2 pack)	(HP 61 ink)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
4	(VIVO Dual LCD Monitor Desk mount)	(Dust-Off Compressed Gas 2 pack)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
5	(Dust-Off Compressed Gas 2 pack)	(VIVO Dual LCD Monitor Desk mount)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008

Only six results remain with a lift of over 1 and confidence .28, confident we can reccomend items based on rules

Strong indication that we can reccomend 'Dust-Off Compressed Gas 2 pack' to users buying apple pencil, Hp 61 or Vivo dual LCD monitor

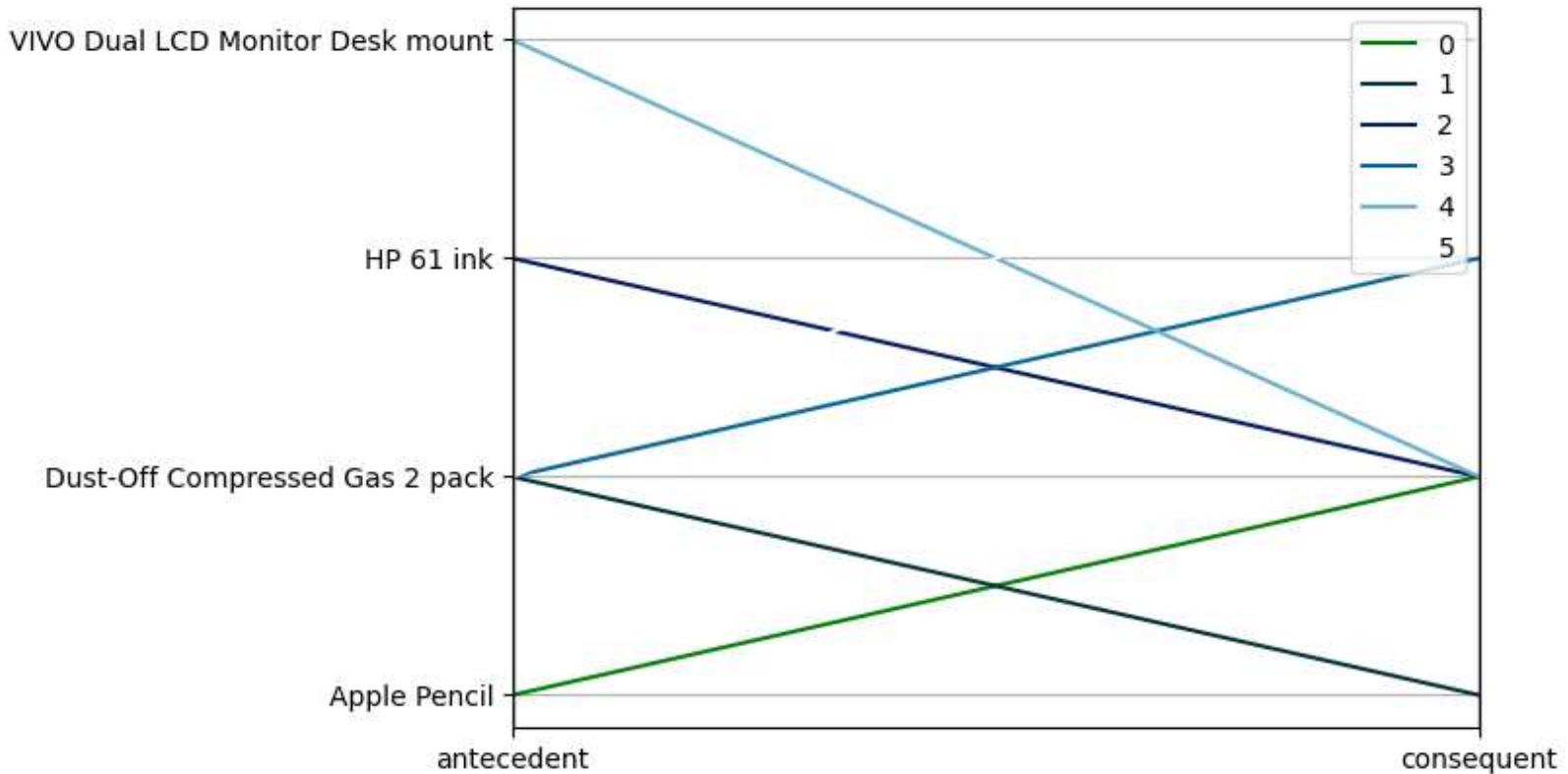
### 3. Generate Parallel Coordinates plot to help visualize findings

In [18]:

```
#Generate Parallel Coordinate plot
# Convert rules to coordinates.
ass_r['antecedent'] = ass_r['antecedents'].apply(lambda antecedent: list(antecedent)[0])
ass_r['consequent'] = ass_r['consequents'].apply(lambda consequent: list(consequent)[0])
ass_r['rule'] = ass_r.index
# Define coordinates and Label
coords = ass_r[['antecedent','consequent','rule']]
# Generate parallel coordinates plot
parallel_coordinates(coords,'rule', colormap = 'ocean')
```

Out[18]:

&lt;AxesSubplot:&gt;



#### 4. Further filter the association rules with lift of 1.2 and confidence of 0.25 thresholds

```
In [19]: #Determine rules based on set conditions for the top 3  
ass_r=ass_r[(ass_r['lift'] >= 1.2) & (ass_r['confidence'] >= 0.25)]  
ass_r.to_csv('/Users/herlihpj/Desktop/Data Analytics/D212 Data Mining II/Task 3/Teleco/results.csv')  
ass_r[0:3]
```

Out[19]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent	consequent	rule
2	(HP 61 ink)	(Dust-Off Compressed Gas 2 pack)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357	HP 61 ink	Dust-Off Compressed Gas 2 pack	2
4	(VIVO Dual LCD Monitor Desk mount)	(Dust-Off Compressed Gas 2 pack)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314	VIVO Dual LCD Monitor Desk mount	Dust-Off Compressed Gas 2 pack	4
5	(Dust-Off Compressed Gas 2 pack)	(VIVO Dual LCD Monitor Desk mount)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008	Dust-Off Compressed Gas 2 pack	VIVO Dual LCD Monitor Desk mount	5



**Final Result: We can very confidently reccomend the dustoff gas to VIVO dual LCD monitor and HP61 ink customers!**