D212 Association Rules and Lift Analysis Performance Assessment, Task 3

Alexa R. Fisher

Western Governors University

Degree: M.S. Data Analytics

Table of Contents

Part I: Research Question

A1. Proposal of Question

The research question proposed for this analysis was " What items are associated with the purchase of an Apple pencil?" The analysis was performed utilizing Market Basket Analysis or MBA. It provided a view of what items were commonly purchased with the featured product. The captured listing could offer brainstorming ideas for the telecommunication company as to which items to provide discounts for. It would be an ideal retention incentive for their customers.

A2. Defined Goal

The main goal of this analysis was to examine if there were any positive relationships between the Apple Pencil and other items. Employing the market basket analysis, a data mining technique, allowed the telecommunication company to identify which items were commonly purchased together within the dataset. It would provide a narrower scope for the company to focus discounts for current clientele.

Part II: Market Basket Justification

B1. Explanation of Market Basket

Market Basket Analysis was a data mining technique, which discovered relationships between items by examining previous transactions (Indeed Editorial Team, 2022). The examination could be used to mark any patterns or trends. This analysis technique analyzed a list of frequently purchased itemset or transactions within a dataset. Those transactions were used to identify any association rules among the items by providing a statistical chart. The association rule was a machine learning method utilized to discover the meaningful relationships between the items depending on their concurrence within the dataset (Susan Currie Sivek, 2020). The rule formed a statement that connected an antecedent item with a consequent item. For example, “if a client buys item 1(antecedent), then they are more likely to also buy item 2(consequent)”.

The analysis was completed by first creating itemsets of the transactional data using the Apriori algorithm. The Apriori algorithm generated the itemsets based on the minimum support value noted (Susan Currie Sivek, 2020). The itemsets were divided and recombined over and over. Their support was recalculated for each combination until no more itemsets could be made. After the creation of the itemsets, the association rules were generated by splitting each frequent itemset into antecedents and consequents along with their various association measures. The measures were noted within the support, lift, and confidence metrics. Within the analysis, the association rules were set to a threshold specified based on a noted measure. In the telecommunication dataset, the measure was lift and it was set to a minimum of 1.0.

The expected outcome of the market basket analysis would be a chart of the antecedent and consequents along with their association metrics based on the thresholds specified. In the telecommunication dataset, the expected chart would include the transactions with “Apple Pencil” noted as either an antecedent or consequent and their noted association rules. For example, if it was an antecedent, the resulting rule would be “Apple Pencil”, then Other. On the opposite, if it was a consequent, the rule would be represented as Other, then “Apple Pencil”.

B2. Transaction Example

An example of transactions within the dataset could be shown as follows. The below showed a single transaction within the imported dataset. The transaction was selected at random; transaction number 15 was displayed.

```
# example of a transaction in the dataset
df.iloc[15]
```

```
Item01    10ft iPHone Charger Cable 2 Pack
Item02        Apple Lightning to USB cable
Item03                       HP952XL ink
Item04                               NaN
Item05                               NaN
Item06                               NaN
Item07                               NaN
Item08                               NaN
Item09                               NaN
Item10                               NaN
Item11                               NaN
Item12                               NaN
Item13                               NaN
Item14                               NaN
Item15                               NaN
Item16                               NaN
Item17                               NaN
Item18                               NaN
Item19                               NaN
Item20                               NaN
Name: 15, dtype: object
```

The newly imported dataset allowed up to twenty different items (Item01-Item20). The specific transaction contained 3 items: “10ft iPHone Charger Cable 2 Pack”, “Apple Lightning to USB cable”, and “HP952XL ink”. Each row of the data represented each transaction, where there was at least one item purchased by a customer of the telecommunication company.

B3. Market Basket Assumption

One assumption of Market Basket Analysis was the existence of associations between items purchased together within a transaction (Indeed Editorial Team, 2022). The analysis assumed meaningful or significant relationships existed between the items customers purchased at the same time (Hull, n.d.). The usage of market basket analysis compacted the data to show representations of the transactional items along with the support, confidence, and lift statistics. These statistics would provide insight into the association of the items. The information could provide intuition on the placement of items within a store or even cross-selling opportunities.

Part III: Data Preparation and Analysis

C1. Transforming The Dataset

The dataset needed to be transformed before it would be used in the analysis. The first step in this data preparation was loading the telecommunication market basket csv file into a dataframe within the coding environment. The loaded data was explored via the shape and info functions to show the total variables and entries. The newly loaded data has 15,002 entries and twenty variables.

```
#install libraries and packages to use with environment for analysis

import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import association_rules, apriori
from mlxtend.preprocessing import TransactionEncoder
```

```
#import the telecommunications churn dataset csv file to be used.
#view dataset to ensure proper loading.

df = pd.read_csv('teleco_market_basket.csv')
pd.set_option('display.max_columns', None)
df.head()
```

```
#viewing shape of data
df.shape
```

```
(15002, 20)
```

```
df.info()
```

```
<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
```

The dataset was checked for duplicates and null values. Duplicates were found using the .duplicated() function and removed using the drop_duplicates() function. The missing or null values were found using the isnull().sum() function to provide a count of null values per variable. The null values were removed using the .dropna() function. Due to removing the rows containing null values, there were gaps within the index. This was resolved by resetting the index.

```
# checking for duplicates
df.duplicated()
```

```
0        False
1        False
2         True
3        False
4         True
         ...
14997     True
14998     True
14999     True
15000     True
15001    False
Length: 15002, dtype: bool
```

```
#removing duplicates
df.drop_duplicates()
```

```
#count of null values
df.isnull().sum()
```

```
Item01     7501
Item02     9255
Item03    10613
Item04    11657
Item05    12473
Item06    13138
Item07    13633
Item08    14021
Item09    14348
Item10    14607
Item11    14746
Item12    14848
Item13    14915
Item14    14955
Item15    14977
Item16    14994
Item17    14998
Item18    14998
Item19    14999
Item20    15001
dtype: int64
```

```
#dropping rows with all null values
df.dropna(how='all')
```

```
#resetting index
df.reset_index(drop=True, inplace=True)
df.info()
```

```
<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
```

Once the data cleaning steps were completed, the dataset was transformed into a list of lists. The list was compiled into a list called "temp_list". The "temp_list" was fit and transformed within the TransactionEncoder() function. After it was transformed, it was converted back into a dataframe.

```
#creating a list of lists (Kamara, n.d.)
temp_list=[]
for i in range (0, 7501):
    temp_list.append([str(df.values[i,j]) for j in range(0,20)])
temp_list
```

```
#transactalize dataset to prep for apriori
enc = TransactionEncoder()
#fit transaction encoder to list of lists and then transform
arry = enc.fit(temp_list).transform(temp_list)
```

```
# convert temp array to dataframe
df_clean = pd.DataFrame(arry, columns=enc.columns_)
df_clean
```

	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	Aut V
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
1	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	True	
2	False	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	True	False	False	True	False	False	False	
4	False	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	False	
7497	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False	
7498	False	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	False	
7500	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	

7501 rows × 120 columns

Lastly, the data was explored again via the info() function, and a list of the support for each variable was generated. In this exploration, it was noted there was a column with just null values. This was also removed to leave the dataset with a total of 7,501 entries and 119 variables.

```
#print support for each item
print(df_clean.mean())
```

```
10ft iPhone Charger Cable             0.005733
10ft iPhone Charger Cable 2 Pack      0.027063
3 pack Nylon Braided Lightning Cable  0.002800
3A USB Type C Cable 3 pack 6FT        0.022530
5pack Nylon Braided USB C cables      0.010132
                                        ...
iPhone Charger Cable Anker 6ft        0.012398
iPhone SE case                        0.014131
nan                                   0.999867
nonda USB C to USB Adapter            0.011199
seenda Wireless mouse                 0.005333
Length: 120, dtype: float64
```

```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7501 entries, 0 to 7500
Columns: 120 entries, 10ft iPhone Charger Cable to seenda Wireless mouse
dtypes: bool(120)
memory usage: 879.1 KB
```

```
#dropping column nan
df_clean.drop('nan', axis=1, inplace=True)
```

```
#view shape to confirm column dropped went from 120 to 119 variables
df_clean.shape
```

```
(7501, 119)
```

The cleaned dataset was included with the submission: “AFCodeD212Tk3_clean.csv”.

```
#save to csv without index
df_clean.to_csv(r'AFCodeD212Tk3_clean.csv', index=False)
```

C2. Code Execution

The code used to generate the association rules was completed using two functions. The first was the .apriori() function, which pulled all the associations that meet the minimum support threshold specified. The second function is the .associated_rules() function, which was used to provide further information based on another specified threshold. The Apriori function used the cleaned dataset along with the minimum support of “0.02” along with the use of column names to provide a chart of variables that met those requirements. The use of the minimum support metric of "0.02" for the created "itemsets_freq" table was ideal due to the balance it provided. The use of a "0.05" provided too few items, which seemed very restrictive. Yet the use of the support measure "0.01" provided too many, which was deemed too inclusive. The support metric indicated the percentage of transactions containing the rule (Hull, n.d.).

```
#run apriori to create frequent itemsets
itemsets_freq = apriori(df_clean, min_support = 0.02, use_colnames = True)
itemsets_freq
```

	support	itemsets
0	0.027063	(10ft iPHone Charger Cable 2 Pack)
1	0.022530	(3A USB Type C Cable 3 pack 6FT)
2	0.035329	(Anker USB C to HDMI Adapter)
3	0.043461	(Apple Lightning to Digital AV Adapter)
4	0.093721	(Apple Pencil)
5	0.071724	(Apple USB-C Charger cable)
6	0.032396	(Cat8 Ethernet Cable)
7	0.118917	(Dust-Off Compressed Gas 2 pack)
8	0.034662	(FEIYOLD Blue light Blocking Glasses)
9	0.031862	(Falcon Dust Off Compressed Gas)
10	0.087455	(HP 61 ink)
11	0.026396	(HP 62XL Tri-Color ink)
12	0.039728	(HP 63XL Ink)
13	0.023330	(HyperX Cloud Stinger Headset)
14	0.038528	(Logitech M510 Wireless mouse)
15	0.047327	(Nylon Braided Lightning to USB cable)
16	0.027996	(Premium Nylon USB Cable)
17	0.026663	(SAMSUNG EVO 32GB card)
18	0.024130	(SanDisk 128GB Ultra microSDXC card)
19	0.021997	(SanDisk 128GB microSDXC card)
20	0.032262	(SanDisk Ultra 128GB card)
21	0.049193	(SanDisk Ultra 64GB card)

22	0.066658	(Screen Mom Screen Cleaner kit)
23	0.046660	(Stylus Pen for iPad)
24	0.043061	(Syntech USB C to USB Adapter)
25	0.036662	(TopMate C5 Laptop Cooler pad)
26	0.087055	(USB 2.0 Printer cable)
27	0.023197	(USB Type C Cable)
28	0.040928	(USB Type C to USB-A Charger cable)
29	0.090255	(VIVO Dual LCD Monitor Desk mount)
30	0.026263	(Dust-Off Compressed Gas 2 pack, Apple Pencil)
31	0.021064	(Apple Pencil, USB 2.0 Printer cable)
32	0.027996	(Dust-Off Compressed Gas 2 pack, HP 61 ink)
33	0.026130	(Dust-Off Compressed Gas 2 pack, Screen Mom Sc...
34	0.030263	(Dust-Off Compressed Gas 2 pack, VIVO Dual LCD...
35	0.022130	(VIVO Dual LCD Monitor Desk mount, HP 61 ink)
36	0.020131	(VIVO Dual LCD Monitor Desk mount, Screen Mom ...

Within the associated_rules() function, the "lift" metric was specified to provide an evaluation of the relationship between items. The lift value was used to determine if the purchase of one item affected the purchase of another (Susan Currie Sivek, 2020). The lift value of greater than "1" suggested that customers would be more likely to purchase the first and second items together. They provided an ideal threshold within the associated_rules() function to sift the results further.

```
# Use association_rules with a lift of greater than 1
a_rules = association_rules(itemsets_freq, metric = 'lift', min_threshold = 1.0)
a_rules
```

C3. Association Rules Table

The below chart was provided to show the association rules table. The table computed and visualized the score summary for support, lift, and confidence metrics of each antecedent.

```
# Use association_rules with a lift of greater than 1
a_rules = association_rules(itemsets_freq, metric = 'lift', min_threshold = 1.0)
a_rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(Dust-Off Compressed Gas 2 pack)	(Apple Pencil)	0.118917	0.093721	0.026263	0.220852	2.356488	0.015118	1.163167	0.653332
1	(Apple Pencil)	(Dust-Off Compressed Gas 2 pack)	0.093721	0.118917	0.026263	0.280228	2.356488	0.015118	1.224113	0.635168
2	(Apple Pencil)	(USB 2.0 Printer cable)	0.093721	0.087055	0.021064	0.224751	2.581712	0.012905	1.177615	0.676017
3	(USB 2.0 Printer cable)	(Apple Pencil)	0.087055	0.093721	0.021064	0.241960	2.581712	0.012905	1.195556	0.671081
4	(Dust-Off Compressed Gas 2 pack)	(HP 61 ink)	0.118917	0.087455	0.027996	0.235426	2.691967	0.017596	1.193534	0.713355
5	(HP 61 ink)	(Dust-Off Compressed Gas 2 pack)	0.087455	0.118917	0.027996	0.320122	2.691967	0.017596	1.295942	0.688760
6	(Dust-Off Compressed Gas 2 pack)	(Screen Mom Screen Cleaner kit)	0.118917	0.066658	0.026130	0.219731	3.296404	0.018203	1.196180	0.790663
7	(Screen Mom Screen Cleaner kit)	(Dust-Off Compressed Gas 2 pack)	0.066658	0.118917	0.026130	0.392000	3.296404	0.018203	1.449149	0.746392
8	(Dust-Off Compressed Gas 2 pack)	(VIVO Dual LCD Monitor Desk mount)	0.118917	0.090255	0.030263	0.254484	2.819626	0.019530	1.220290	0.732443
9	(VIVO Dual LCD Monitor Desk mount)	(Dust-Off Compressed Gas 2 pack)	0.090255	0.118917	0.030263	0.335303	2.819626	0.019530	1.325540	0.709367
10	(VIVO Dual LCD Monitor Desk mount)	(HP 61 ink)	0.090255	0.087455	0.022130	0.245199	2.803721	0.014237	1.208988	0.707155
11	(HP 61 ink)	(VIVO Dual LCD Monitor Desk mount)	0.087455	0.090255	0.022130	0.253049	2.803721	0.014237	1.217945	0.704986
12	(VIVO Dual LCD Monitor Desk mount)	(Screen Mom Screen Cleaner kit)	0.090255	0.066658	0.020131	0.223043	3.346089	0.014114	1.201279	0.770703
13	(Screen Mom Screen Cleaner kit)	(VIVO Dual LCD Monitor Desk mount)	0.066658	0.090255	0.020131	0.302000	3.346089	0.014114	1.303360	0.751218

C4. Top Three Rules

The top three rules in the associated rules table were shown in the below table. The top three rules had a lift of over “2.70” and a confidence measure of over “0.30”. The lift indicated an effect of the consequent being included in the transaction with the antecedent. The higher the lift, the stronger the effect. As for the confidence score, it indicated the likelihood of one item being purchased given the presence of another (Susan Currie Sivek, 2020).

```
#top 3 rules based on set conditions
top_3 = a_rules[(a_rules['lift'] >= 2.70) & (a_rules['confidence'] >= 0.30)].sort_values(by=['lift'], ascending=False)
top_3
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
12	(Screen Mom Screen Cleaner kit)	(VIVO Dual LCD Monitor Desk mount)	0.066658	0.090255	0.020131	0.302000	3.346089	0.014114	1.303360	0.751218
7	(Screen Mom Screen Cleaner kit)	(Dust-Off Compressed Gas 2 pack)	0.066658	0.118917	0.026130	0.392000	3.296404	0.018203	1.449149	0.746392
9	(VIVO Dual LCD Monitor Desk mount)	(Dust-Off Compressed Gas 2 pack)	0.090255	0.118917	0.030263	0.335303	2.819626	0.019530	1.325540	0.709367

Part IV: Data Summary and Implications

D1. Significance Of Support, Lift, and Confidence Summary

The final analysis based on my research question was computed by viewing the total count of each antecedent and consequent. This was to ensure that the item, "Apple Pencil" was included in both listings. Once this was confirmed the associated rules table was filtered to only show "Apple Pencil".

```
#checking counts of antecedents and viewing to see if Apple Pencil is included
a_rules.antecedents.value_counts()
```

```
(Dust-Off Compressed Gas 2 pack)      4
(VIVO Dual LCD Monitor Desk mount)    3
(Apple Pencil)                        2
(HP 61 ink)                           2
(Screen Mom Screen Cleaner kit)       2
(USB 2.0 Printer cable)               1
Name: antecedents, dtype: int64
```

```
#checking counts of consequents and viewing to see if Apple Pencil is included.
a_rules.consequents.value_counts()
```

```
(Dust-Off Compressed Gas 2 pack)      4
(VIVO Dual LCD Monitor Desk mount)    3
(Apple Pencil)                        2
(HP 61 ink)                           2
(Screen Mom Screen Cleaner kit)       2
(USB 2.0 Printer cable)               1
Name: consequents, dtype: int64
```

```
ante_df = a_rules[a_rules['antecedents'] == {'Apple Pencil'}]
cons_df = a_rules[a_rules['consequents'] == {'Apple Pencil'}]
apple_pencil = pd.concat([ante_df, cons_df])
apple_pencil
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(Apple Pencil)	(Dust-Off Compressed Gas 2 pack)	0.093721	0.118917	0.026263	0.280228	2.356488	0.015118	1.224113	0.635168
2	(Apple Pencil)	(USB 2.0 Printer cable)	0.093721	0.087055	0.021064	0.224751	2.581712	0.012905	1.177615	0.676017
1	(Dust-Off Compressed Gas 2 pack)	(Apple Pencil)	0.118917	0.093721	0.026263	0.220852	2.356488	0.015118	1.163167	0.653332
3	(USB 2.0 Printer cable)	(Apple Pencil)	0.087055	0.093721	0.021064	0.241960	2.581712	0.012905	1.195556	0.671081

The resulting table provided the statistics for the specified item. There were four rules returned. The first was if "Apple Pencil" then " Dust-Off Compressed Gas 2 pack". The second rule was if "Apple Pencil" then "USB 2.0 Printer cable". The third was if " Dust-Off Compressed Gas 2 pack" then "Apple Pencil". Lastly, the fourth rule was if "USB 2.0 Printer cable" then "Apple Pencil".

The significance of support, lift, and confidence was summarized in the following points.

- Support was the proportion of transactions that contained the rule of the overall transactions within the dataset (Susan Currie Sivek, 2020). Within the resulting table, there were about 2.6% of the 7501 transactions that contained "Apple Pencil" also purchased "Dust-Off Compressed Gas 2 pack". There were about 2.1% of the 7501 transactions containing "Apple Pencil" that also purchased "USB 2.0 Printer cable" within the same transaction.
- Confidence was the proportion of all the transactions which included "Apple Pencil", "Dust-Off Compressed Gas 2 pack", and "USB 2.0 Printer cable" over the proportion of the transactions containing just the antecedents. For example, the rule "Apple Pencil", then " Dust-Off Compressed Gas 2 pack" was a 6% higher confidence metric than the opposite rule of " Dust-Off Compressed Gas 2 pack", then "Apple Pencil". This was calculated by subtracting the confidence metrics of the rules "Apple Pencil", then " Dust-Off Compressed Gas 2 pack" and " Dust-Off Compressed Gas 2 pack" then "Apple Pencil". The calculation would be 0.28 – 0.22 = 0.06 or 6%. The other rules for "Apple Pencil" and "USB 2.0 Printer cable" only showed a 2% difference in confidence. The confidence for those items was higher in the rule "USB 2.0 Printer

cable", then "Apple Pencil". It was shown by the calculation of 0.24 – 0.22 = 0.02 or 2%, as noted in the confidence measurements for each of the noted rules.

- Lift was how much the expectations were exceeded when customers actually bought both items (Susan Currie Sivek, 2020). A lift value greater than "1" indicated that there was a positive relationship. The antecedent was noted to increase the likelihood of the consequent occurring within the transaction. Similarity to support, the lift was noted both ways. For the rules including "Apple Pencil" and "Dust-Off Compressed Gas 2 pack", the lift was about 2.36. For the rules containing "Apple Pencil" and "USB 2.0 Printer cable", the lift was about 2.58.

Although the support for the rule "Dust-Off Compressed Gas 2 pack", then "Apple Pencil" had a higher percentage in comparison to the opposite. The confidence score was higher for "Apple Pencil" being the antecedent with "Dust-Off Compressed Gas 2 pack" being the consequent.

Usually compressed gas was purchased to remove dust from various electronics. Being that an Apple Pencil was considered an electronic item, it would be reasonably considered the rule "Apple Pencil", then "Dust-Off Compressed Gas 2 pack" was correct. If someone was purchasing an Apple Pencil, then they were likely to buy a can of compressed gas.

D2. Practical Significance of Findings

The practical significance of the analysis findings was considered significant. The customers who purchased Apple pencils represented over 9.3 % of the transactions of the dataset. It was noted as a particularly good percentage to evaluate this item as a key product. The key product would be an ideal starting point for the sales and marketing team for identifying discounting opportunities.

D3. Course of Action

The relationship of If “Apple Pencil”, then “Dust-Off Compressed Gas 2 pack” implied that customers who purchased Apple pencils often purchased compressed gas as well. The recommended course of action suggested would be to provide coupons or discounts toward compressed gas for customers with Apple pencils. This recommendation was based on the results from the analysis of the lift, support, and confidence for the specified rule. Having such a high lift metric increased the likelihood of this expectation. It would provide the company with a good starting point to see if discounting could affect retention efforts.

Part V: Attachments

E. Panopto Recording

The below link was a video providing an overview of the Python code used for the data analysis. The recording demonstrated the code’s functionality as well as provide an overview of the Jupyter Notebook programming environment.

Link Found here:

[redacted]

F. Third-Party Web Sources

Kamara, Kesselly.(n.d.). Market Basket Analysis in Python. Webinar Recording [Video] WGU Hosted Panopto:

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=dbe89ddb-e92f-4d40-a87a-af030178abf1

G. References

Hull, I. (n.d.). *Market Basket Analysis in Python*. Retrieved from Datacamp: https://app.datacamp.com/learn/courses/market-basket-analysis-in-python

Indeed Editorial Team. (2022, October 12). *FAQ: What Is Market Basket Analysis? (Types Plus Examples)*. Retrieved from Indeed: https://sg.indeed.com/career-advice/career-development/market-basket-analysis#:~:text=The%20approach%20relies%20on%20the,rules%20or%20if%2Dthen%20statements.

Susan Currie Sivek, P. (2020, November 16). *Market Basket Analysis 101: Key Concepts*. Retrieved from Toward Data Science: https://towardsdatascience.com/market-basket-analysis-101-key-concepts-1ddc6876cd00