MARKET&BASKET

INSIGHT

DATA MINING: MARKET BASKET ANALYSIS WITH APRIORI ALGORITHM

DATA-DRIVEN STRATEGIES:

HUGE RETAILERS PIVOT ON A DETAILED MARKET BASKET ANALYSIS TO UNCOVER ASSOCIATIONS BETWEEN ITEMS. USING THIS VALUABLE INFORMATION, THEY ARE ABLE TO CARRY OUT VARIOUS STRATEGIES TO IMPROVE THEIR REVENUE:

- ASSOCIATED PRODUCTS ARE PLACED CLOSE TO EACH OTHER, SO THAT BUYERS OF ONE ITEM WOULD BE PROMPTED TO BUY THE OTHER.
- DISCOUNTS CAN BE APPLIED TO ONLY ONE OF THE ASSOCIATED PRODUCTS.

ASSOCIATION RULE MINING

- BUT HOW EXACTLY IS A MARKET BASKET ANALYSIS CARRIED OUT?

- DATA SCIENTISTS ARE ABLE TO CARRY OUT MARKET BASKET ANALYSIS BY IMPLEMENTING ASSOCIATION RULE MINING. ASSOCIATION RULE MINING IS A RULE-BASED MACHINE LEARNING METHOD THAT HELPS TO UNCOVER MEANINGFUL CORRELATIONS BETWEEN DIFFERENT PRODUCTS ACCORDING TO THEIR CO-OCCURRENCE IN A DATA SET.

- HOWEVER, ONE OF THE MAJOR PITFALLS IS THAT IT CONSISTS OF VARIOUS FORMULAS AND PARAMETERS THAT MAY MAKE IT DIFFICULT FOR PEOPLE WITHOUT EXPERTISE IN DATA MINING. THEREFORE, BEFORE SHARING YOUR RESULTS WITH STAKEHOLDERS, MAKE SURE THAT THE UNDERLYING DEFINITIONS ARE WELL-UNDERSTOOD.

STEPS IN THE APRIORI ALGORITHM

- THE DIAGRAM BELOW ILLUSTRATES HOW THE APRIORI ALGORITHM STARTS BUILDING FROM THE SMALLEST ITEMSET AND FURTHER EXTENDS FORWARD.

- THE ALGORITHM STARTS BY GENERATING AN ITEMSET THROUGH THE JOIN STEP, THAT IS TO GENERATE (K+1) ITEMSET FROM K-ITEMSETS. FOR EXAMPLE, THE ALGORITHM GENERATES COOKIE, CHOCOLATE AND CAKE IN THE FIRST ITERATION.

- IMMEDIATELY AFTER THAT, THE ALGORITHM PROCEEDS WITH THE PRUNE STEP, THAT IS TO REMOVE ANY CANDIDATE ITEM SET THAT DOES NOT MEET THE MINIMUM SUPPORT REQUIREMENT. FOR EXAMPLE, THE ALGORITHM WILL REMOVE CAKE IF SUPPORT(CAKE) IS BELOW THE PREDETERMINED MINIMUM SUPPORT.

- IT ITERATES BOTH OF THE STEPS UNTIL THERE ARE NO POSSIBLE FURTHER EXTENSIONS LEFT.

- NOTE THAT THIS DIAGRAM IS NOT THE COMPLETE VERSION OF THE TRANSACTIONS TABLE ABOVE. IT SERVES AS AN ILLUSTRATION TO HELP PAINT THE BIGGER PICTURE OF THE FLOW.

GROUPING INTO TRANSACTIONS

- THE DATA SET RECORDS INDIVIDUAL ITEM PURCHASES IN A ROW. WE WILL HAVE TO GROUP THESE PURCHASES INTO BASKETS OF ITEMS.
- AFTER THAT, WE WILL USE TRANSACTIONENCODER TO ENCODE THE TRANSACTIONS INTO A FORMAT THAT IS SUITABLE FOR THE APRIORI FUNCTION.

CODE:

```
BASKET.ITEMDESCRIPTION =
BASKET.ITEMDESCRIPTION.TRANSFORM(LA
MBDA X: [X])
BASKET =
BASKET.GROUPBY(['MEMBER_NUMBER','DA
TE']).SUM()
['ITEMDESCRIPTION'].RESET_INDEX(DROP=
TRUE)

ENCODER = TRANSACTIONENCODER()
TR[illegible]NSA[illegible]IONS =
[illegible]RAME(ENCODER.FIT(BASKET).TR
[illegible]ORM(BASKET),
[illegible]OLUMNS=ENCODER.COLUMNS_)
DI[illegible]LAY(TRANSACTIONS.HEAD())
```

output

	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics	bags	baking powder	bathroom cleaner	beef	berries	...
0	False	False	False	False	False	False	False	False	False	False	...
1	False	False	False	False	False	False	False	False	False	False	...
2	False	False	False	False	False	False	False	False	False	False	...
3	False	False	False	False	False	False	False	False	False	False	...
4	False	False	False	False	False	False	False	False	False	False	...

APRIORI AND ASSOCIATION RULES

THE APRIORI ALGORITHM WILL BE USED TO GENERATE FREQUENT ITEM SETS. WE WILL BE SPECIFYING THE MINIMUM SUPPORT TO BE 6 OUT OF TOTAL TRANSACTIONS. THE ASSOCIATION RULES ARE GENERATED AND WE FILTER FOR LIFT VALUE > 1.5

code:

```
frequent_itemsets = apriori(transactions, min_support=
6/len(basket), use_colnames=True, max_len = 2)
rules = association_rules(frequent_itemsets, metric="lift",
min_threshold = 1.5)
display(rules.head())
print("Rules identified: ", len(rules))
```

output:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(UHT-milk)	(butter milk)	0.021386	0.017577	0.000601	0.028125	1.600131	0.000226	1.010854
1	(butter milk)	(UHT-milk)	0.017577	0.021386	0.000601	0.034221	1.600131	0.000226	1.013289
2	(cream cheese)	(UHT-milk)	0.023658	0.021386	0.000869	0.036723	1.717152	0.000363	1.015922
3	(UHT-milk)	(cream cheese)	0.021386	0.023658	0.000869	0.040625	1.717152	0.000363	1.017685
4	(soda)	(artif. sweetener)	0.097106	0.001938	0.000468	0.004818	2.485725	0.000280	1.002893

Rules identified: 190

Visualizations

To visualize our association rules, we can plot them in a 3D scatter plot. Rules that are closer to top right are the rules that can be the most meaningful to be further dived in.

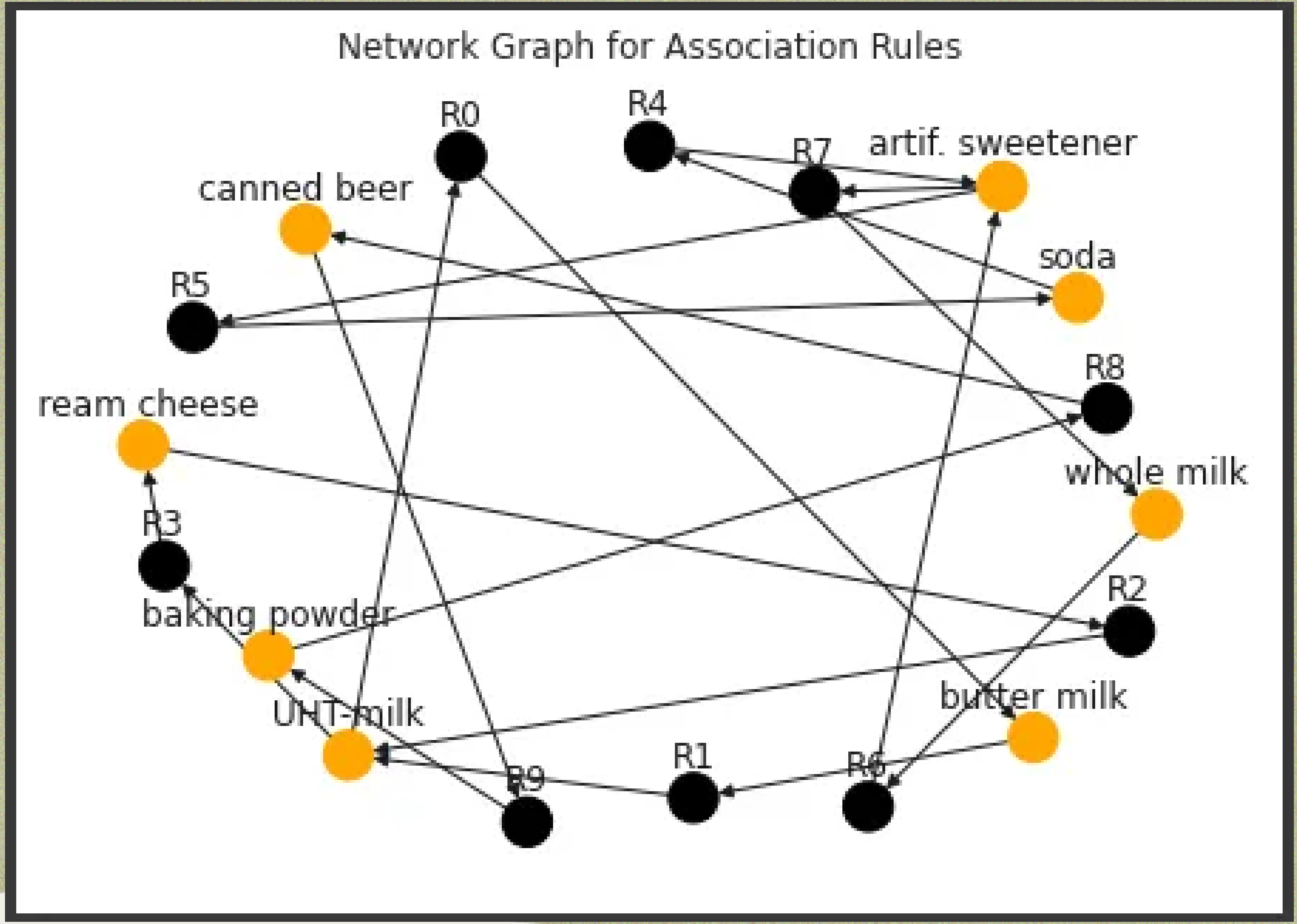
Code:

```
def draw_network(rules, rules_to_show):
# Directional Graph from NetworkX
network = nx.DiGraph()

# Loop through number of rules to show
for i in range(rules_to_show):

# Add a Rule Node
network.add_nodes_from(["R"+str(i)])
for antecedents in rules.iloc[i]['antecedents']:
# Add antecedent node and link to rule
network.add_nodes_from([antecedents])
network.add_edge(antecedents, "R"+str(i),  weight = 2)

for consequents in rules.iloc[i]['consequents']
# Add consequent node and link to rule
network.add_nodes_from([consequents])
network.add_edge("R"+str(i), consequents,  weight = 2)
```

code:

```
color_map=[]

# For every node, if it's a rule, colour as Black, otherwise Orange
for node in network:
if re.compile("^[R]\d+$").fullmatch(node) != None:
color_map.append('black')
else:
color_map.append('orange')

# Position nodes using spring layout
pos = nx.spring_layout(network, k=16, scale=1)
# Draw the network graph
nx.draw(network, pos, node_color = color_map, font_size=8)

# Shift the text position upwards
for p in pos:
pos[p][1] += 0.12

nx.draw_networkx_labels(network, pos)
plt.title("Network Graph for Association Rules")
plt.show()

draw_network(rules, 10)
```

output:


Network Graph for Association Rules
R0
R4
R7
artif. sweetener
canned beer
soda
R5
R8
ream cheese
whole milk
R3
R2
baking powder
butter milk
UHT-milk
R1
R6
R9

Business Application

Let's say the grocery has bought up too much Whole Milk and is now worrying that the stocks will expire if they cannot be sold out in time. To make matters worse, the profit margin of Whole Milk is so low that they cannot afford to have a promotional discount without killing too much of their profits.

For instance, we can apply a promotional discount on Brandy, Softener, Canned Fruit, Syrup and Artificial Sweetener. Some of the associations may seem counter-intuitive, but the rules state that these products do drive the sales of Whole Milk.

Takeaway

By implementing the Apriori Algorithm and analyzing the association measures, businesses can derive dozens of data-driven strategies to boost their revenue and profits. These association rules are critical in data mining for analyzing consumer’s purchasing behavior. Some of the most important strategies of a retailer, such as Customer analytics, Market Basket analysis and Product Clustering are able to derive valuable insights through association rule mining.

Thank you!

By Claudia Alves