UNIVERSITY OF WISCONSIN – MADISON

Department of Industrial and Systems Engineering
ISyE 521 – Machine Learning in Action for Industrial Engineers

MARKET BASKET ANALYSIS

Supervisors:
Prof. Justin Boultilier
Ari Smith

Team members:
Aravind Varathan
Yogesh Ramakrishnan Mohan

Academic years:
Fall’21

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background

Market Basket Analysis is a data mining method that focusses on uncovering associations between items by looking for combinations of items that occur together frequently in transactions. This is done by analyzing transaction database of many customers using Association Rules (Apriori algorithm). By doing this, decisions like which item to stock more, cross selling, up selling, store shelf arrangement are determined.

1.2 Problem Statement

Grocery is an especially attractive sector. The number of grocery shoppers and the frequency of grocery shopping has been increasing. In order to provide higher customer satisfaction and increase the probability of a product getting sold, it is important to have a good knowledge of customer buying pattern. This will provide insights about product recommendations and product bundling (which products to be grouped on the shelves). Since, the transaction database might be incredibly large, an analytical tool is required to work on the dataset and provide meaningful insights.

1.3 Objective

a. To work on weekly transaction dataset of 'Grupo Bimbo', a bakery retail store which sells baked goods and snack items.
b. Apply Apriori algorithm and generate itemset
c. Prune the insignificant association rules (itemset) formed based on tuning rules (like hyperparameter)
d. Provide product recommendations/product bundles based on best itemset

1.4 Scope

The scope is limited to working on a particular week and train the Apriori model. Test the model on a different week's dataset and identify customer purchase pattern. Based on the results from the former, make recommendations to the company.

CHAPTER 2: METHODOLOGY

2.1 Data Collection

The data was collected from Kaggle and the link to the source is given below https://www.kaggle.com/c/grupo-bimbo-inventory-demand/data

2.2 Exploratory Data Analysis

These are the features of the dataset. The features in Spanish have been converted to English for better interpretation.

Features (Spanish)	Features (English)
Semana	week_number
Producto_ID	product_id
NombreProducto	product_name
Venta_uni_hoy	sales_unit_this_week
Venta_hoy	sales$_this_week (in $)
Dev_uni_proxima	returns_unit_next_week
Dev_proxima	returns$_next_week (in $)
Demanda_uni_equil	adjusted_demand (Sales this week – returns unit next week)
Agencia_ID	sales_depot_id
Canal_ID	sales_channel_id
Ruta_SAK	route_id (routes taken to deliver)
Cliente_ID	client_id (customer ID)
NombreCliente	client_name(customer name)
State	state

Table 1.1., Features of this dataset

The dataset contains 9 weeks data about 74180464 observations, 1799 unique products and 880604 clients.

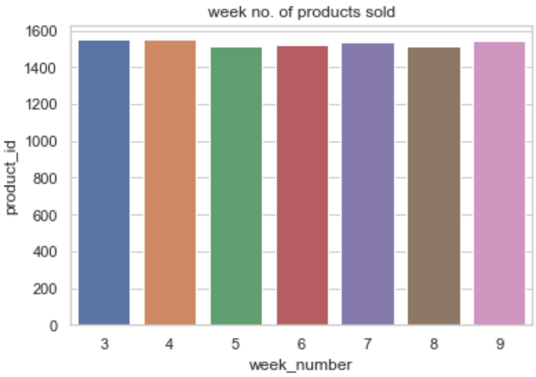


Figure 2.2.1., Weekly sales of all products

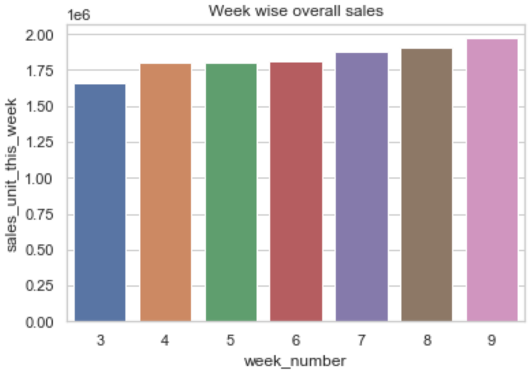


Figure 2.2.2., No. of products sold each week

To limit the size of dataset for the project, weekly sales of the products and number of products sold for each week were observed. From the observations, it was seen that the number of products sold was fairly same and only the sales number was slightly different. Since in Apriori algorithm the frequency of the products is the important parameter, it was decided to work on week 3 transaction dataset to train the model and test the model on week 4. From this any change in customer buying patter can be observed.

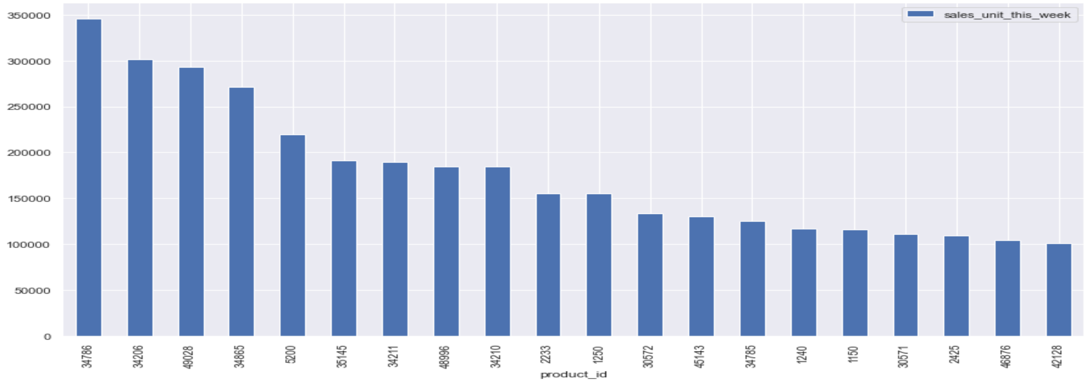


Figure 2.2.3., Top20 products in terms of sales

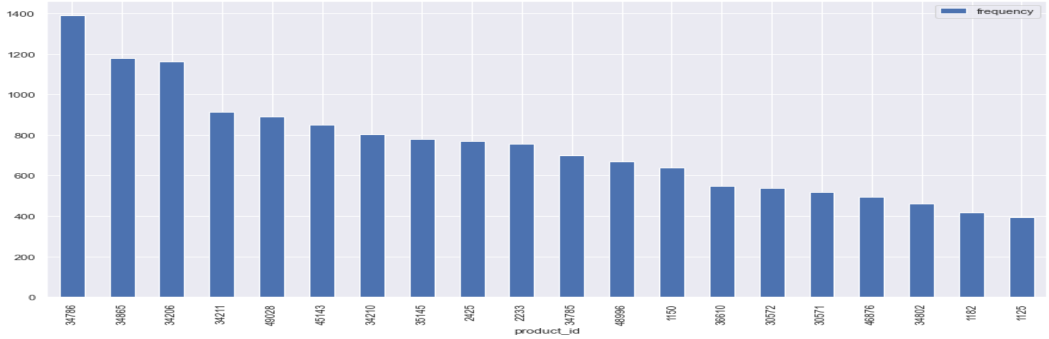


Figure 2.2.4., Top20 products in terms of frequency (frequently bought)

In week 3, Sales of products and frequency of occurrence of Top 20 products were analyzed. It can be seen from the charts that the products with high sales don’t necessarily mean they are frequently bought. This shows that few clients buy some products in large quantities. Since the dataset contains sales to clients (retailers), this relation is logical.

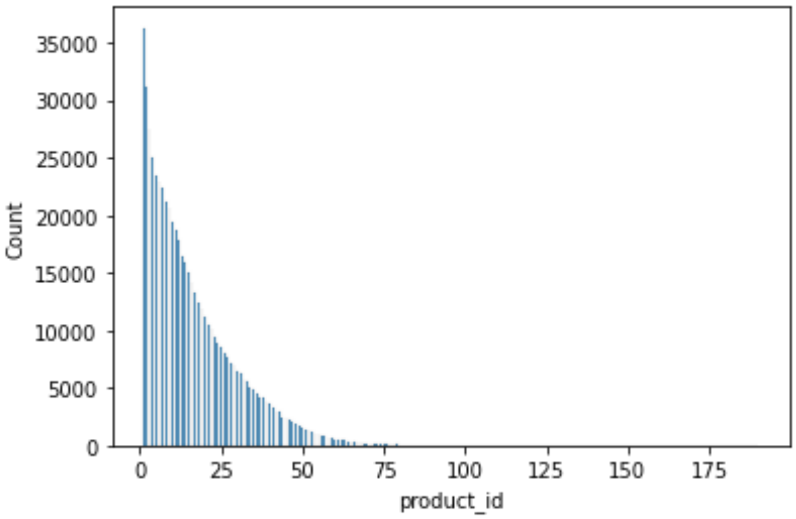


Figure 2.2.5., Histogram of count of products bought by clients

Figure 2.4., shows that most of the clients have bought less than 25 unique products in week 3. A few (<5000 clients) bought 25-50 unique products and very few bought more than 50 unique products of Grupo Bimbo company. This shows chances of increasing the customer base by recommending different products while purchasing.

2.3 Data Preprocessing & Applying Apriori Algorithm

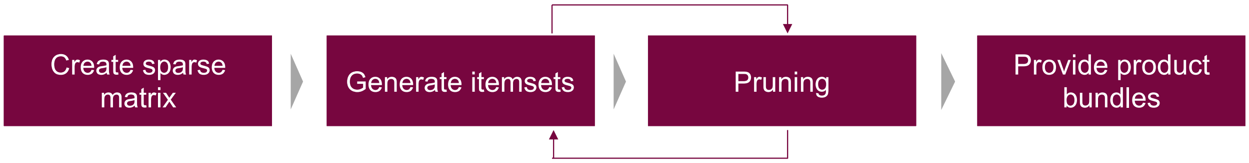
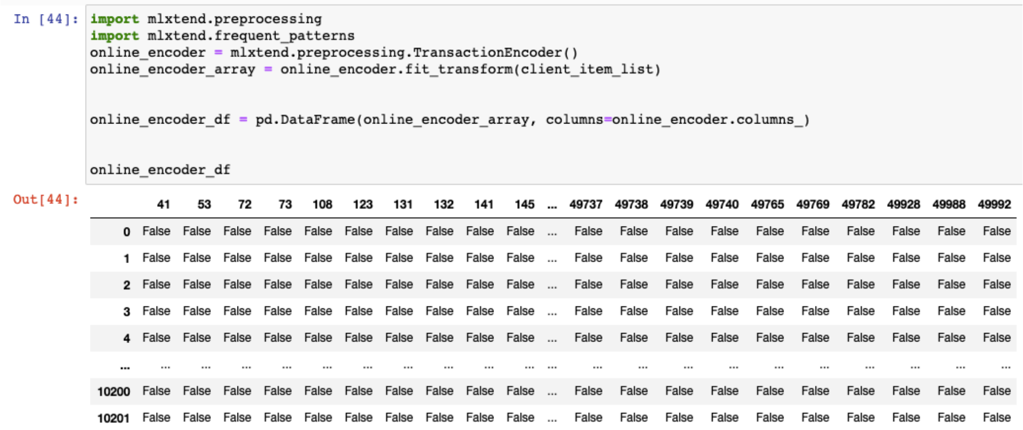


Figure 2.3.1., Process flow in Association Rules

Figure 2.3.1., shows the processes involved in Association rules. The data must first be transformed into a sparse matrix and then itemsets were generated using Association rules. There are three algorithms under Association Rules – Apriori, Eclat and FP-growth. Out of the three, Apriori was chosen considering the size of dataset and simplicity of the concept. The itemsets or the rules were then pruned to remove insignificant rules by tuning rules – Support, Confidence & Lift (like hyperparameter in Apriori). Based on the best itemsets, recommendations were made.

To create sparse matrix, the dataset is transformed into invoice like dataset and then converted into list (list of products each client has bought) and then using the below code shown in Figure 2.3.2., the list is converted into a sparse matrix in which each observation represents a client and product IDs are in the columns.



```
In [44]: import mlxtend.preprocessing
         import mlxtend.frequent_patterns
         online_encoder = mlxtend.preprocessing.TransactionEncoder()
         online_encoder_array = online_encoder.fit_transform(client_item_list)


         online_encoder_df = pd.DataFrame(online_encoder_array, columns=online_encoder.columns_)


         online_encoder_df
```

Out[44]:

	41	53	72	73	108	123	131	132	141	145	...	49737	49738	49739	49740	49765	49769	49782	49928	49988	49992
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	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	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	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10200	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
10201	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

Figure 2.3.2., Sparse matrix

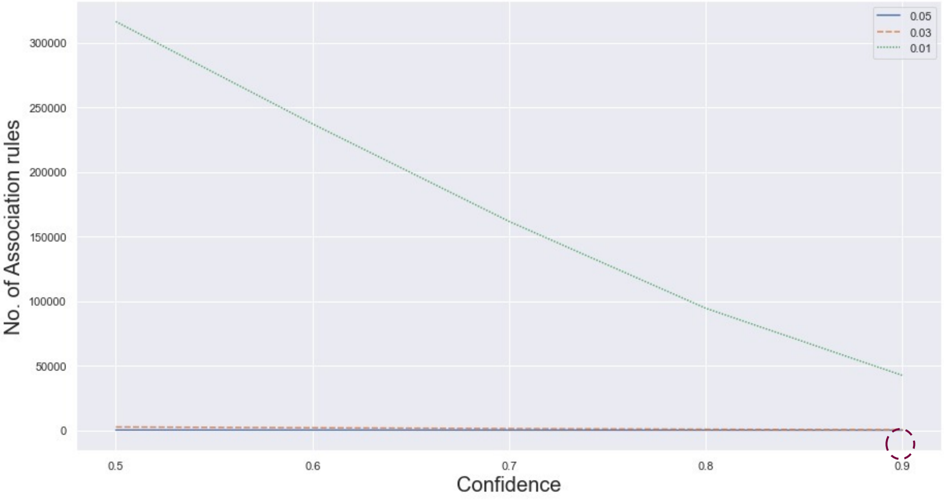
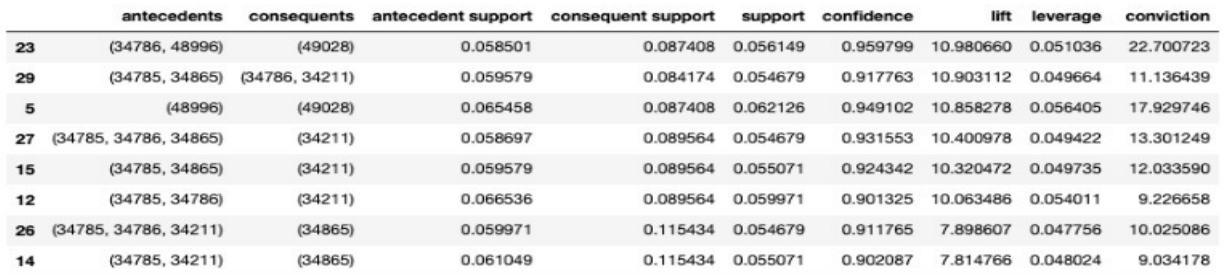


Figure 2.3.3. No. of Association rules formed vs confidence for different support values

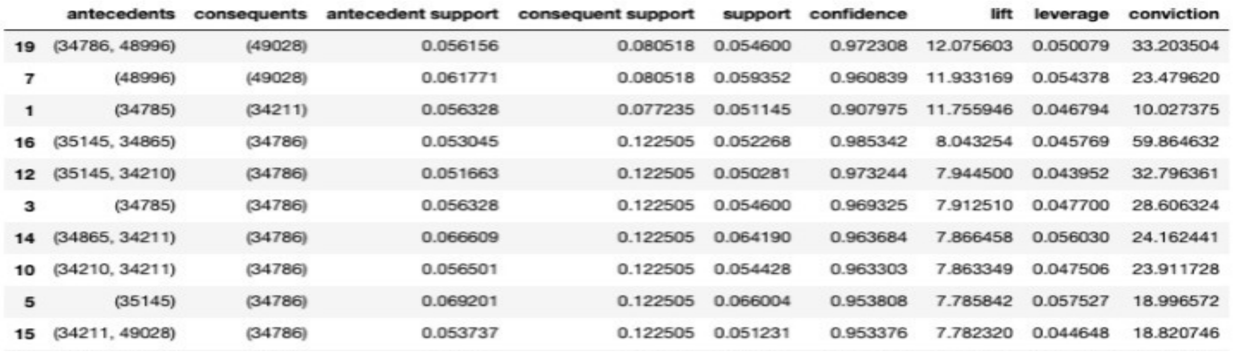
Apriori algorithm is applied on the sparse matrix generated using different rules (Support, Confidence). Figure 2.3.3., shows the number of associations rules formed for different value of Support & Confidence. To have stricter rules, Support value was chosen to be 0.05 (choosing products or itemsets which have appeared at least 5% out of overall transaction) and confidence value of 0.9 (90% probability of consequent occurring if antecedent occurs). By choosing these values, the total



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
23	(34786, 48996)	(49028)	0.058501	0.087408	0.056149	0.959799	10.980660	0.051036	22.700723
29	(34785, 34865)	(34786, 34211)	0.059579	0.084174	0.054679	0.917763	10.903112	0.049664	11.136439
5	(48996)	(49028)	0.065458	0.087408	0.062126	0.949102	10.858278	0.056405	17.929746
27	(34785, 34786, 34865)	(34211)	0.058697	0.089564	0.054679	0.931553	10.400978	0.049422	13.301249
15	(34785, 34865)	(34211)	0.059579	0.089564	0.055071	0.924342	10.320472	0.049735	12.033590
12	(34785, 34786)	(34211)	0.066536	0.089564	0.059971	0.901325	10.063486	0.054011	9.226658
26	(34785, 34786, 34211)	(34865)	0.059971	0.115434	0.054679	0.911765	7.898607	0.047756	10.025086
14	(34785, 34211)	(34865)	0.061049	0.115434	0.055071	0.902087	7.814766	0.048024	9.034178

Figure 2.3.4. Top 10 Association Rules formed with Support - 0.05 & Confidence – 0.9 on train

association rules formed are 30. Figure 2.3.4., shows the top 5 association rules formed on train set.



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
19	(34786, 48996)	(49028)	0.056156	0.080518	0.054600	0.972308	12.075603	0.050079	33.203504
7	(48996)	(49028)	0.061771	0.080518	0.059352	0.960839	11.933169	0.054378	23.479620
1	(34785)	(34211)	0.056328	0.077235	0.051145	0.907975	11.755946	0.046794	10.027375
16	(35145, 34865)	(34786)	0.053045	0.122505	0.052268	0.985342	8.043254	0.045769	59.864632
12	(35145, 34210)	(34786)	0.051663	0.122505	0.050281	0.973244	7.944500	0.043952	32.796361
3	(34785)	(34786)	0.056328	0.122505	0.054600	0.969325	7.912510	0.047700	28.606324
14	(34865, 34211)	(34786)	0.066609	0.122505	0.064190	0.963684	7.866458	0.056030	24.162441
10	(34210, 34211)	(34786)	0.056501	0.122505	0.054428	0.963303	7.863349	0.047506	23.911728
5	(35145)	(34786)	0.069201	0.122505	0.066004	0.953808	7.785842	0.057527	18.996572
15	(34211, 49028)	(34786)	0.053737	0.122505	0.051231	0.953376	7.782320	0.044648	18.820746

Figure 2.3.5. Top 10 Association Rules formed with Support - 0.05 & Confidence – 0.9 on test

From Figure 2.3.4., and 2.3.5., it can be seen that the association rules formed are almost same (only the order is changed) and there is possibility of extrapolation of results for the rest of the weeks.

2.4 What is an association rule?

Association rules are "if-then" statements, that help to show the relationships between data items, within large data sets in various types of databases. One of the applications of association rule mining is to help discover sales correlations in transactional data or in medical data sets. Figure 2.4.1 shows the typical example of an association rule.

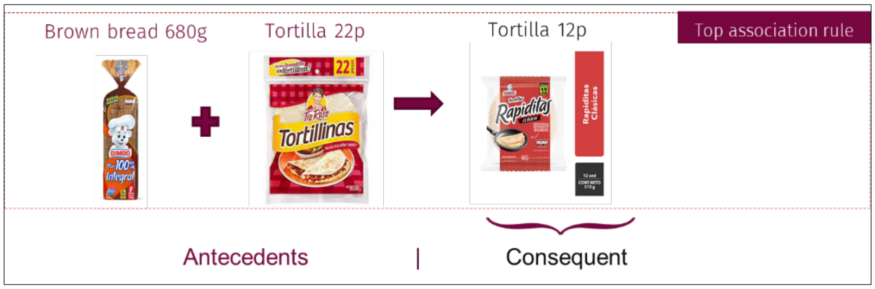
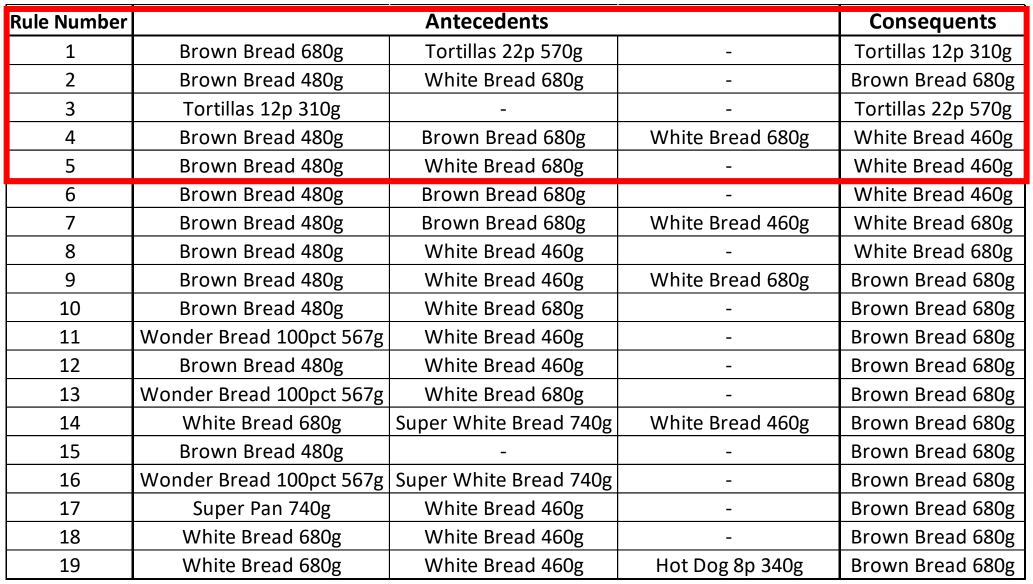


Figure 2.4.1. Example of an association rule

Antecedents are the product(s) that a customer purchases and consequent is the product(s) that the customer will be inclined towards buying. This prediction is made according to the buying patterns recorded by the Apriori algorithm based on the customer's previous transactional data.

CHAPTER 3: RESULTS

3.1 Interpretation of Results



Rule Number	Antecedents			Consequents
1	Brown Bread 680g	Tortillas 22p 570g	-	Tortillas 12p 310g
2	Brown Bread 480g	White Bread 680g	-	Brown Bread 680g
3	Tortillas 12p 310g	-	-	Tortillas 22p 570g
4	Brown Bread 480g	Brown Bread 680g	White Bread 680g	White Bread 460g
5	Brown Bread 480g	White Bread 680g	-	White Bread 460g
6	Brown Bread 480g	Brown Bread 680g	-	White Bread 460g
7	Brown Bread 480g	Brown Bread 680g	White Bread 460g	White Bread 680g
8	Brown Bread 480g	White Bread 460g	-	White Bread 680g
9	Brown Bread 480g	White Bread 460g	White Bread 680g	Brown Bread 680g
10	Brown Bread 480g	White Bread 680g	-	Brown Bread 680g
11	Wonder Bread 100pct 567g	White Bread 460g	-	Brown Bread 680g
12	Brown Bread 480g	White Bread 460g	-	Brown Bread 680g
13	Wonder Bread 100pct 567g	White Bread 680g	-	Brown Bread 680g
14	White Bread 680g	Super White Bread 740g	White Bread 460g	Brown Bread 680g
15	Brown Bread 480g	-	-	Brown Bread 680g
16	Wonder Bread 100pct 567g	Super White Bread 740g	-	Brown Bread 680g
17	Super Pan 740g	White Bread 460g	-	Brown Bread 680g
18	White Bread 680g	White Bread 460g	-	Brown Bread 680g
19	White Bread 680g	White Bread 460g	Hot Dog 8p 340g	Brown Bread 680g

Figure 3.1.1. Rules formed using train set

Model results on the train set for support threshold of 0.05 and confidence threshold of 0.9 gave us the rules in Figure 3.1.1. For example, the first rule tells us that {Brown bread 680g, Tortillas 22p 570g} are the Antecedents and {Tortillas 12p 310g} is the consequent. What this means intuitively is, if a customer purchases the antecedent, the model predicts that the customer is most likely to buy the consequent (tortillas 12p in this case). This prediction is made using past transaction information.

Model results on the test set and train set for support threshold of 0.05 and confidence threshold of 0.9 gave us the same association rules with top three rules being exactly the same

The network diagrams shown in Figure 3.1.1 represents the associations between the products where each line segment represents one rule connecting the antecedent and the consequent. The network diagram shows that product ID 34786, which is the brown bread 680g item was the most frequently occurring product.

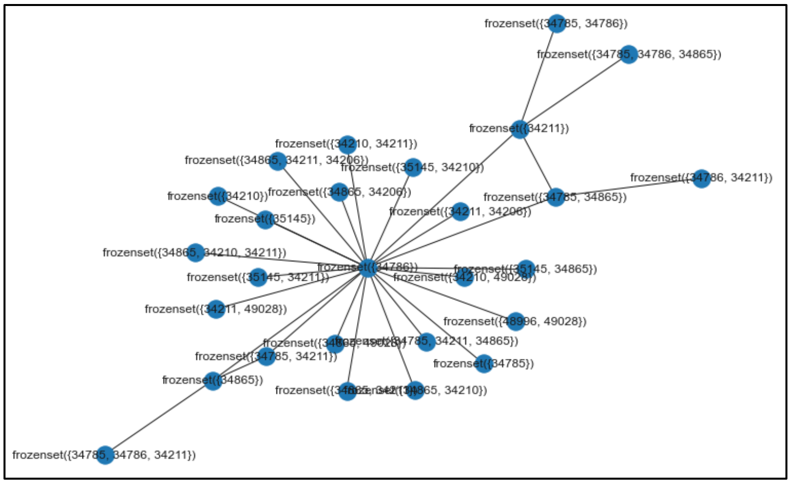

frozenset({34785, 34786})
frozenset({34785, 34786, 34865})
frozenset({34211})
frozenset({34210, 34211})
frozenset({34865, 34211, 34206})
frozenset({35145, 34210})
frozenset({34786, 34211})
frozenset({34210})
frozenset({34865, 34206})
frozenset({34785, 34865})
frozenset({34211, 34206})
frozenset({35145})
frozenset({34865, 34210, 34211})
frozenset({34786})
frozenset({35145, 34865})
frozenset({35145, 34211})
frozenset({34210, 49028})
frozenset({34211, 49028})
frozenset({48996, 49028})
frozenset({34785, 34211, 34865})
frozenset({34785, 34211})
frozenset({34785})
frozenset({34865})
frozenset({34865, 34210})
frozenset({34785, 34786, 34211})

Figure 3.1.1. Network Diagram representing the association rules for train set

3.2 Recommendations to the company

Market basket analysis gives us the most significant rules for a given transactional dataset. These association rules can be used to devise marketing strategies like

- **Creating product bundles:** Stacking the products in an itemset(rule) is one strategy. This is done by combining the antecedents and consequents together on the shelves of a grocery store. Recommending the consequent when they purchase the antecedent in an online space is a parallel strategy that can be used. For example, in Figure 3.1.1, we can recommend bundling {Brown bread 680g, Tortillas 22p 570g, Tortillas 12p 310g} from rule 1. This means that stacking them together on shelves might increase their sales.
- **Upselling:** When a rule contains a higher variant of the same product as the consequent, recommending the higher end product when the customer is interested in buying the lower end version is called upselling. Apple's iPhones follow this strategy. For example, in Figure 3.1.1, from rule 3, we can recommend Tortillas 22p 570g to the customers who buy Tortillas 12p 310g.
- **Cross-selling:** Recommending a different product when the customer is interested in buying variations of the same product is called cross selling. For example, from rule 1, we can recommend Tortillas 12p 310g to customers who buy Brown bread 680g.
- **Combination offers:** Grouping the products in a rule together and selling them at a marginally reduced price pushes the customer to opt for the bundle because it saves them some money. We can combine {Brown bread 480g, White bread 680g, Brown bread 680g} and offer it at a reduced price.

APPENDIX

1. https://taufik-azri.medium.com/recommendation-system-for-retail-customer-3f0f80b84221
2. https://medium.com/swlh/market-basket-analysis-102-alteryx-designer-python-1792228eb0bb
3. https://www.kaggle.com/klospascal/eda-market-basket-analysis/notebook
4. https://goldinlocks.github.io/Market-Basket-Analysis-in-Python/
5. https://www.datacamp.com/community/tutorials/market-basket-analysis-r
6. https://medium.datadriveninvestor.com/how-to-build-a-recommendation-system-for-purchase-data-step-by-step-d6d7a78800b6
7. https://www.kaggle.com/benroshan/market-basket-analysis
8. https://www.statisticshowto.com/market-basket-analysis/
9. https://towardsdatascience.com/a-gentle-introduction-on-market-basket-analysis-association-rules-fa4b986a40ce
10. https://techbusinessguide.com/what-is-market-basket-analysis/
11. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-market-basket-analysis/
12. https://en.wikipedia.org/wiki/Affinity_analysis
13. https://thecleverprogrammer.com/2020/11/16/apriori-algorithm-using-python/
14. https://github.com/anubhav199/Market-Basket-Analysis
15. https://github.com/ashishpatel26/Market-Basket-Analysis/blob/master/Market%20Basket%20Analysis%20Using%20apyori%20%20package.ipynb
16. https://github.com/sharmaroshan/Market-Basket-Analysis
17. https://medium.com/@jihargifari/how-to-perform-market-basket-analysis-in-python-bd00b745b106
18. https://www.youtube.com/watch?v=PYxne5D_32c&t=1758s
19. https://select-statistics.co.uk/blog/market-basket-analysis-understanding-customer-behaviour/
20. Files shared by Prof. Justin Boultilier

Link to our codes:

https://drive.google.com/drive/folders/12w4DELHYqNWzwmhvzf-2L_ya3SnNVLwO?usp=sharing