MKT-680-4100
Marketing Analytics

Pernalonga-Segmentation

Group 5

Group Members:

Rush Bharadwaj

Yi Cai

Sameera Vattikuti

Nicholas Yang

Zeyu Zhu

Table of Contents

Chapter 1: Introduction

1.1 Business Background

Pernalonga, a leading supermarket chain with 421 stores which sells ~10K products in 430 categories with a consumer base of ~7900.

Currently, 30% of their sales are through promotions (mostly in-store) executed in partnership with suppliers.

In order to increase efficiency of spends, client intends to shift to a more targeted and personalized approach to promotions. To this end, an analysis is required on customer transaction data to determine which customers are more likely to buy which item given a particular offer.

1.2 Problem Statement

While the overall problem statement is to develop a marketing campaign to experiment on personalized promotions, in this project, we intend to first focus on the segmentation aspect of the problem.

In summary, the problem statement is as follows:

"Divide the customers, products/product categories and stores into segments so that the transactional details and behavioral traits associated with them are homogenous within a segment and heterogeneous across segments. "

This segmentation will help us identify and implement ideal promotional activities for a given intersection of customer, product and store segments.

Chapter 2: Data Overview

For the purpose of our analysis, we have transactional data related to ~2.69 Million unique transactions and 29.6 Million line items. This data corresponds to 421 stores, selling ~10000 unique products belonging to 429 categories. The stores serviced 7920 unique customers in the given time period.

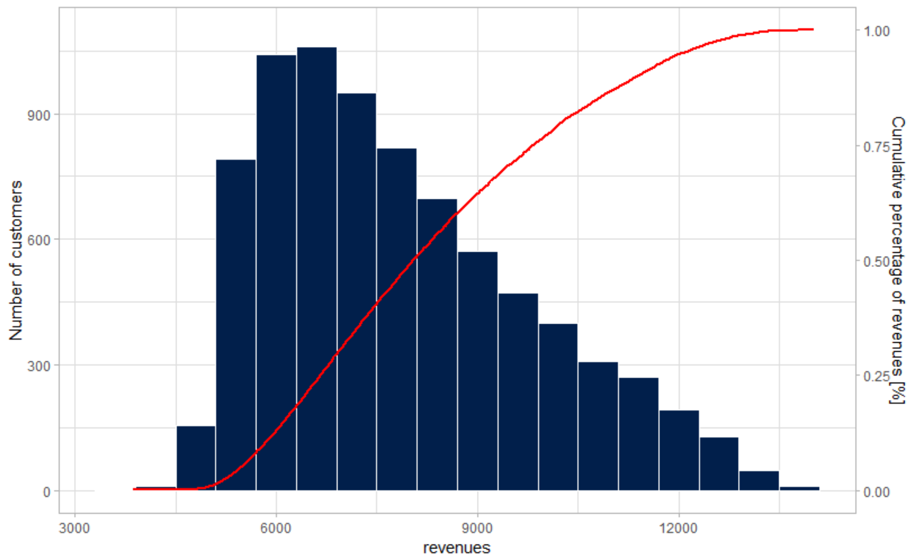
Note: *In the process of exploring the data, it was identified that the number of unique transactions were less than the number of unique customers which is erroneous in the real world. In order to rectify this, we created a unique transaction id key using the date and unique customer id. This key will operate under the assumption that a customer only purchases once a day from the store.*

We first grouped the data by customer_id, product_ id and store_id to analyze the measurements such as revenue generated, transactions, volumes, customer visits respectively. From the summarized statistics we can drive a few insights: 1) the range between the extreme values are large. We can identify certain products and stores are only bought or visited as low as one time. 2)when we take a closer look at those extreme values, they are not outliers that can be simply excluded from the analysis. The big gap between the minimum and maximum values indicates the great different between the groups and a single promotion strategy cannot fit all.

Then, we want to identify the top performers in each entity class in terms of the revenues, transactions, traffics and so on.

2.1 Customer Overview

Graph 1: Revenue Distribution by Customers



Graph 1 is a histogram of the revenue distribution by customers. The bin size is set at $600 and the distribution showed in the graph is right skewed. The graph indicates that most customers are spending $5700- $6900 and less than 300 customers are spending 12000 dollars or more. The red line shows the cumulative percentage of revenues. The slope of the line reflects how much each bin of customers contributed to the total sales revenue. We can infer that the groups of customers that are contributing the most to the total revenues are customers who are spending around $6500, which is less than the average $7855 customer spending.

2.2 Product Overview

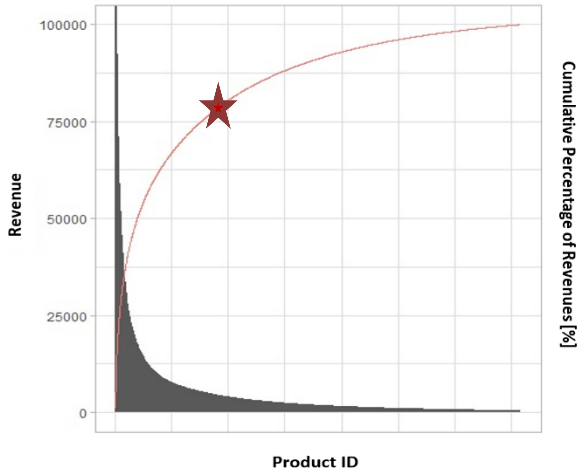
To explore which products or product categories are top sellers, we use Recency- Frequency- Monetary method to rate product categories. The recency measures the numbers of days between the last transaction and 2018-01-01, which is one day after the latest transaction data in the data. The frequency and monetary evaluates how many times a category of products was bought and how much revenue it generates. Then we assign scores (range from 1 to 4) to the recency, frequency and revenues.

category_desc_eng	recency	frequency	monetary_value	RFMScore
FRESH PORK	1	632046.0	2525657.92	111
FRESH BEEF	1	408244.0	2468050.23	111
FRESH POULTRY MEAT	1	597559.0	2443876.26	111
FINE WINES	1	333855.0	1871669.82	111
BEER WITH ALCOHOL	1	189459.0	1631996.92	111
WASHING MACHINE DETERGENTS	1	121845.0	1584967.11	111
COFFEES AND ROASTED MIXTURES	1	279025.0	1380687.93	111
FROZEN FISH SERVICE	1	154693.0	1242226.18	111
FINE WAFERS	1	663634.0	1188805.62	111
OLIVE OIL	1	207899.0	1141254.51	111

Graph 2: Top-selling Product Categories by RFM

Graph 2 indicates that customers most likely to buy fresh meat, followed by wines and alcohols. Understand the business context, fresh meat and alcohols are not applicable for promotions, so we can cultivate the marketing strategy with a focus on products such as detergents and coffee.

Graph 3 is a bar chart that visualize the revenue for each product, ranked from high to low. The red line indicates the cumulative percentage of revenues. The red star highlights that the top 25% of the best-selling product accounts for 75% of the total revenue.



Graph 3: Revenue Distribution by Product ID

2.3 Store Overview

To evaluate the store performance, we group the measurements by store_id and sum up the revenue and count the number of transactions, customers and products sold in each store. The best performing store generates a revenue of $786521, with 30357 transactions happened among 567 customers. With the analysis, the top 25% of the stores generates sales revenues ranging between $182016.7 and $786521, which add up to 48% of the total revenues generated by all stores combined. Similarly, 46% of the transaction happens in the top 25% of stores

The fundamental analysis of the data gives us some basic insights on the customers, products, and stores. We identify in each entity which are driving the business and generating the revenues. Yet simply designing promotion strategy aiming at those top-performers is not optimal, because first, we cannot make sure that the most valuable customers are buying top-selling product in most popular store, and second, the market size will be very limited. We want to cluster the customers into distinguish groups and understand that for different groups of customers, what kind of products they are buying and in which stores. We need more detailed segmentation on each entity to cultivate tailored promotion strategy by answering the question: promote what product to which customer group buying in what store.

Chapter 3: Segmentation Modeling

3.1 Customer Segmentation

In order to conduct the segmentation on customers, we first need to examine the data. Data contains two tables - Transaction and Product. Transaction table describes the information related to each transaction (customer and product information). Product table stores the information describing what each product is (category & brand). Since our goal is conducting a segmentation on customers with the hope of extracting potential patterns for customer shopping habits/behaviors, we merged these two tables into one on their product_id. After merging the data, we have 29617585 rows of data with 18 attributes (columns). Aiming to set customer as the dependent variable for our segmentation model, we grouped the dataset by customer_id, which means that each row of data represents the aggregated information for each unique customer (customer_id). The aggregation information is listed below:

'tran_id' - number of unique data, number of all count

'tran_dt' - number of unique data

'store_id' - number of unique data, mode

'tran_prod_sale_qty' - sum

'tran_prod_sale_amt' - sum

'tran_prod_discount_amt' - sum, average

'tran_prod_paid_amt' - sum

'tran_prod_offer_cts': - sum

'prod_unit_price': - average

'subcategory_id': - number of unique data

'category_id': - number of unique data

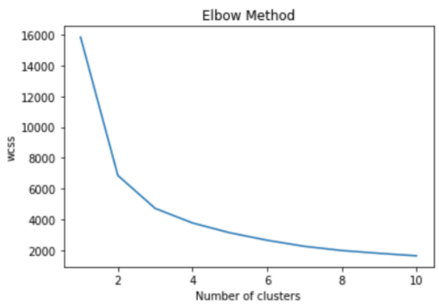
'category_id': - number of unique data

'brand_desc': - number of unique data

Since our ultimate task is providing a solution for customized promotion strategy, we created a new attribute with the aggregated level attribute to describe the discount ratio (discount_ratio) as :

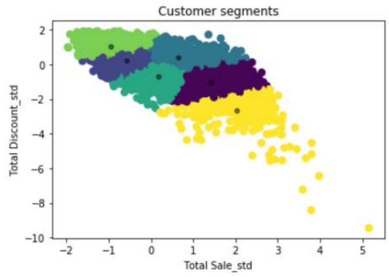
abs(discount_sum)/tran_prod_sale_amt

The most important parameter for the K-means algorithm is the value for K. We conducted a WCSS score test over the number of clusters can be potentially adopted. As shown below, we



decided to set K as 6 with a consideration of efficiency. Since selection of the number of clusters (K value) has risks between the trade-off off model complexity and segmentation performance, we consider 6 is an optimal number as it outputs great results without sacrificing the efficiency of the model. Also, since the K-means algorithm is very sensitive to the scale of the data, we performed standardization process over the aggregated level of data.

The summary statistics for each group can be found in the appendix. The example visualization of clustering is showing below.



For demonstration purpose, we will use Cluster 3 as an example to conduct RFM (Recency Frequency Monetary) on for further segmentation examination. Summary statistics, as well as RFM results for Cluster 3, can be found below.

```
customer[customer.kmeans_seg==3].describe().round(2)
```

	tran_unique	tran_cnt	tran_date_unique	store_unique	store_mode	prod_cnt	qty_sum	gross_sales	discount_sum	discount_avg	net_sales
count	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00
mean	386.73	4760.27	390.64	6.34	444.93	1084.21	8173.99	10108.96	-1560.56	-0.34	8548.39
std	107.77	859.30	108.26	4.68	206.36	229.60	1207.77	887.98	492.29	0.14	751.64
min	149.00	2194.00	149.00	1.00	137.00	430.00	6203.39	7374.24	-3008.41	-1.16	6507.79
25%	304.00	4183.00	307.00	3.00	288.00	932.00	7370.16	9462.72	-1898.59	-0.43	7993.49
50%	370.00	4693.00	374.00	5.00	388.00	1089.00	7906.90	10215.69	-1542.75	-0.33	8580.70
75%	449.00	5232.00	453.00	8.00	588.00	1239.00	8657.50	10815.85	-1188.77	-0.24	9115.22
max	717.00	8581.00	717.00	46.00	999.00	1816.00	16148.74	11840.03	-351.64	-0.05	10489.31

Summary Statistics for Cluster 3

	recency	frequency	monetary_value	r_quartile	f_quartile	m_quartile	RFMScore	category_id	category_desc_eng
0	1	74424.0	492746.25	1	1	1	111	95894.0	FRESH BEEF
74424	1	111400.0	488346.68	1	1	1	111	95890.0	FRESH PORK
185824	1	107025.0	464418.89	1	1	1	111	95888.0	FRESH POULTRY MEAT
292849	1	49800.0	246964.72	1	1	1	111	95797.0	FINE WINES
342649	1	35171.0	225464.93	1	1	1	111	95977.0	WILD FRESH FISH
377820	1	26808.0	216615.68	1	1	1	111	95978.0	FRESH FISH AQUACULTURE
404628	1	18453.0	212611.23	1	1	1	111	95788.0	WASHING MACHINE DETERGENTS
423081	1	45546.0	209627.76	1	1	1	111	96026.0	COFFEES AND ROASTED MIXTURES
468627	1	23968.0	200657.38	1	1	1	111	95974.0	FROZEN FISH SERVICE
492595	1	30185.0	188128.44	1	1	1	111	95800.0	BEER WITH ALCOHOL

RFM Statistics for Cluster 3

The table lists the top 10 product categories that the Cluster 3 customers are mostly buying in terms of the recency, frequency and money dollar. We can see customers in Cluster 3 mostly are buying household groceries, especially meat.

Since we are trying to find interesting patterns among customers and stores, we conducted a separate clustering on top of the data preparation process. Instead of grouping by just customer_id, we added store_id. Hence, every row of the data represents the information of the customer and store combination. Hence, a specific customer can be segmented into different groups with different store combinations. Sample data can be seen below:

```
customer_store.head(100)
```

		tran_id		tran_dt	prod_id	tran_prod_sale_qty	tran_prod_sale_amt	tran_prod_discount_amt		tran_prod_paid_amt	tran_prod_offer_cts
		nunique	count	nunique	nunique	sum	sum	sum	mean	sum	sum
cust_id	store_id										
29568	**168**	6	23	6	20	39.118	32.40	-6.95	-0.302174	25.45	11
	188	3	20	3	17	57.322	48.95	0.00	0.000000	48.95	0
	192	243	3777	243	1021	7621.444	10945.85	-1250.13	-0.330985	9695.72	1002
	288	2	25	2	25	44.810	92.09	-4.30	-0.172000	87.79	3
	311	20	49	20	23	84.564	183.09	-43.35	-0.884694	139.74	7
	341	1	4	1	4	8.000	19.67	-4.30	-1.075000	15.37	1
	478	25	275	25	203	464.140	792.19	-76.10	-0.276727	716.09	70
	602	8	142	8	123	267.062	335.86	-56.43	-0.397394	279.43	73
29909	**176**	21	231	21	179	390.345	502.49	-48.56	-0.210216	453.93	52
	188	220	3032	220	1105	4549.803	7005.92	-944.60	-0.311544	6061.32	887
	192	10	360	10	256	624.079	995.16	-185.69	-0.515806	809.47	182
	293	1	5	1	5	9.969	10.08	-0.60	-0.120000	9.48	1
	308	4	187	4	168	254.263	497.97	-81.50	-0.435829	416.47	92
	514	64	443	64	259	686.373	941.71	-113.37	-0.255914	828.34	94
	523	1	5	1	5	13.007	8.50	0.00	0.000000	8.50	0
	549	3	12	3	11	15.446	26.77	-4.81	-0.400833	21.96	5
	575	75	1648	75	700	2656.947	3499.37	-345.47	-0.209630	3153.90	555
	606	1	56	1	56	79.818	155.18	-20.18	-0.360357	135.00	9
39774	**258**	269	3102	269	1125	5678.077	10884.30	-2280.14	-0.735055	8604.16	1637

Similarly, 6 clusters were segmented. Segmentation information is listed below.And their summary statistics are attached in the appendix.

Segmentation interpretation

Group1: High life quality customers

Group2: Skin-care liker
large transaction of face care, cream waxes, etc…

Group3: Frequent customers

Group4: Cherry-picker

Group5: Cooked food liker
large transaction of different dishes, pre fried savory, etc…

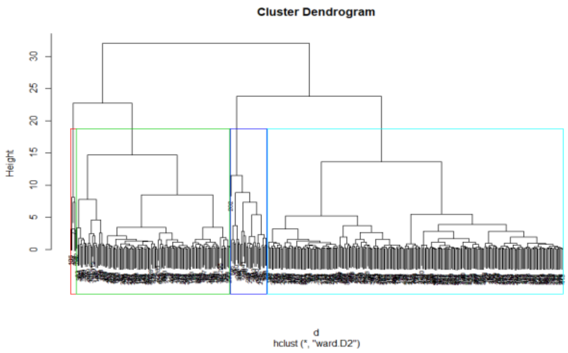
Group6: Infrequent customers

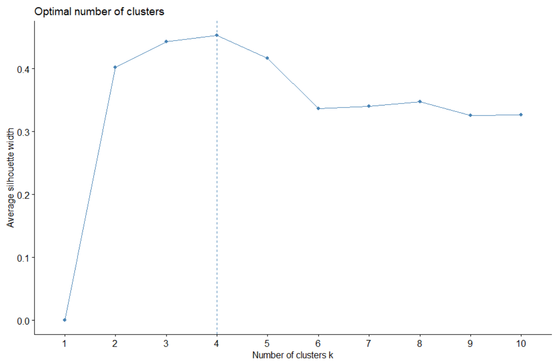
3.2 Product Segmentation

As for products, we applied k-means and hierarchy clustering algorithms on product categories to see if there are interesting groups of products that could answer questions like which categories are most frequently promoted.

The attributes used in this algorithm include 7 variables that provide information about revenue generated per purchase, average volume, number of transactions, number of customers, unit price, discount occurrence ratio and discount amount ratio.

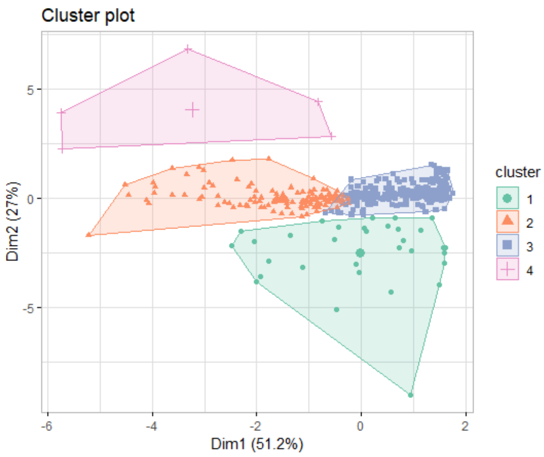
Using function hclust in R, we were able to create the dendrogram as graph000. After examining the average silhouette width by different number of clusters, we set the value of k to be 4 because it gives the best performance on how similar an object is to its own cluster compared to other clusters.



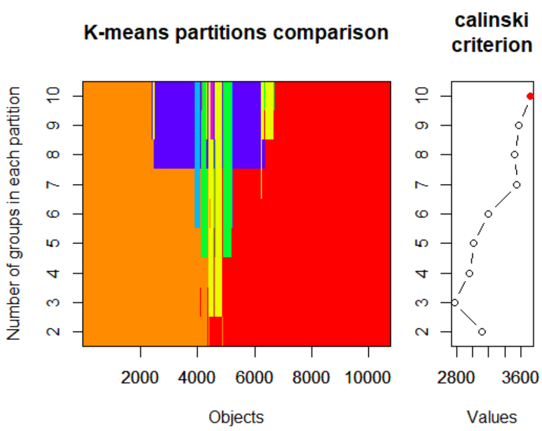


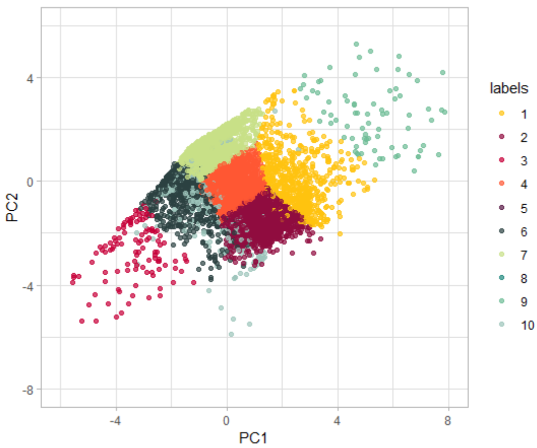
We can see interesting findings from table000 like that group 4 are the most expensive categories, group 3 are the least frequently discounted categories and group 1 are most frequently purchased.

sub_grp	avg_qty	avg_price	avg_num_offer	avg_discount	avg_freq	cat_count
1	3.575772	1.468134	0.359629	0.109601	35.11937	32
2	1.156033	8.878205	0.587305	0.211783	7.997577	134
3	1.134045	3.592852	0.196358	0.048579	12.69549	258
4	1.022935	122.5209	0.48537	0.186719	1.035978	5



Likewise, we also applied k-means on these variables. Based on calinski criterion, which also measures the similarity within each group, we chose the best k as 10.





This following table tells us that group 8 is the most expensive group of categories, group 2 is most frequently discounted, and group 10 has the highest average volume. Mapping back to categories, we can find that group 8 actually is the electronic devices, group 10 are mostly liquids with units as karat, and group 2 are a mix of daily commodities like shampoo or toothpaste, which all make sense with its transaction data.

group	revenues	volumes	transactions	customers	unit_price	discount_ratio	discount_amt_ratio	unique
1	15.17531	1.502391	169623	7889	16.54978	0.717248	0.327527	84
2	2.725235	1.299111	1377866	7920	3.51125	0.721546	0.376197	191
3	1.751908	1.737353	2189962	7920	1.752239	0.278151	0.085913	59
4	2.582167	1.299756	1412836	7920	2.489329	0.47132	0.161792	254
5	0.517205	1.291718	1103374	7913	0.467816	0.147141	0.064972	2
6	1.907129	1.303251	2253193	7920	1.888089	0.271406	0.107222	194
7	2.326048	1.225061	1930753	7920	2.363056	0.113195	0.02639	366
8	291.9626	1.05	80	75	339.9749	0.6375	0.184291	3
9	42.30265	1.622377	5635	3279	49.66208	0.575409	0.283411	34
10	1.664565	5.203327	1168739	7919	0.414825	0.329586	0.175833	41

When we have both product as well as customers, we can combine this two to find if there is interesting insights about right promotion to right people.

We can see from this table that customer1 (high-life-quality customers) spent much greater money on group1, 3 and 9, which is high-price commodities, machine and accessories and food ingredients.

And group 4 (cherry-pickers) are more into buying group 2, group 8 and group 10 (frequent-discounted goods, electronics and large-volume goods). And not buy much on products in group 7 (seldom discounted products).

So it's reasonable for us to promote or give offers on these items to specifically group 4 who are sensitive to discounts.

Group	Product 1	Product 2	Product 3	Product 4	Product 5	Product 6	Product 7	Product 8	Product 9	Product 10
1	7.1185	3.3669	52.3757	2.6937	33.5235	14.5540	3.9864	0.9253	2.2192	8.3686
2	4.2185	5.4492	26.7690	3.9673	45.9086	10.6104	4.5751	0.5937	0.7871	10.5182
3	4.1763	5.0113	26.2326	3.1613	38.3339	9.7450	3.5085	1.3321	0.8747	9.9404
4	4.3758	6.0446	26.5937	3.6376	41.9740	10.2625	3.6799	1.1083	1.0087	11.7930
5	5.0004	6.7409	27.8304	4.3238	44.5687	11.4519	4.5328	0.2253	0.8884	12.2688
6	3.3098	4.8357	25.0934	3.4765	41.3047	9.6831	4.7321	0.3618	0.7945	10.9742

3.3 Store Segmentation

The analysis conducted on store information was from the transaction data and product data. Both the tables were merged and put together and then summarized using data.tables on R Studio. This led to seeing that there were 421 stores in the data and had various different trends throughout. Some of the interesting trends can be observed can be seen in table below:

	Store Number	Descriptive value
Store with highest volume of sales	164	670291
Store with highest revenue	164	$786,521
Store with most transactions	342	363011

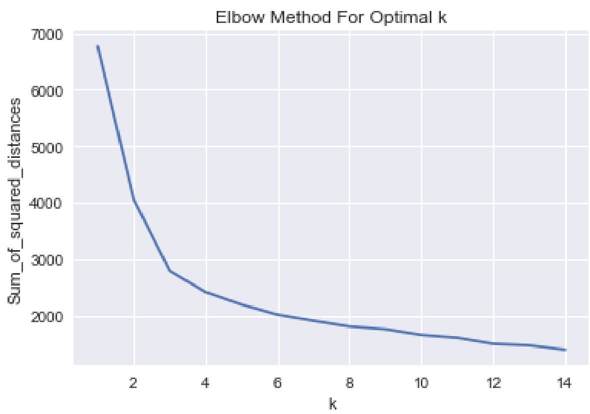
For further analysis K means clustering technique was used to find the segments within the store data. For this analysis the average values were computed per store and were clubbed together with the segments obtained from products and customer segmentations. To conduct the following analysis python's scikit-learn package was used. To find the most optimal K, the elbow method was used, and the most optimal k was 5. Using 5, we obtained our analysis with the below segmentation as in table below:

	Average distribution of stores within a segment								
Segments	Sales	Discounted Sales	Discounts offered	Quantity Sold	Offers Used	Unique Products Sold	Unique Categories Sold	Unique Customers	Transactions
A	$60,965	$51,881	14.90%	41,044	7,909	3,453	305	81	593
B	$403,503	$337,251	16.40%	271,743	55,915	7,772	387	188	723
C	$136,120	$114,705	15.70%	89,781	18,406	5,585	359	112	715
D	$241,178	$202,927	15.90%	162,303	32,644	6,823	374	141	725
E	$776,119	$630,256	18.80%	505,160	112,163	9,242	409	482	719

The descriptive statistics are mentioned in table below:

		Count
Segment A	Neighborhood retailer	89
Segment B	Supply Retailer	24
Segement C	Mega supermarket	191
Segment D	Supermarket	108
Segment E	Wholesale Supermarket	9

It can be seen there is a very clear differentiation in store segmentations making them M.E.C.E. This is a good sign for the modeling approach as we can see clear boundaries. The model selection criterion for store segmentation was the BIC score. When we used k = 5 and just the summarized information of the store to model the segments, a high BIC score was seen than when the product and customer segmentation information was added to the model. This was a clear sign of how segmentations of customers and products can help improve segmenting the stores. Also, a point to be noted, as the elbow for the segments can be seen between 3 to 4 in figure here. The value of 5 was decided when the first segmentation was done when stores were just summarized, and no customer and product segmentations were added to the model.



Chapter 4: Conclusion

4.1 Customer

Based on the above segmentation exercise, we have 6 segments of customers derived from their shopping preferences, with most of them choosing products with around 15% discount rate as opposed to the highest discount rate. This could mean that most customers do have a level of brand preference instead of opting to purchase a good with the highest discount. It implies that the discounts/promotions merely prepone or hasten the purchase. The following table gives an overview of different customer segments, their interpretations and possible business goals which could prompt promotions in those segments.

Segment	Segment Interpretation	Business Goals & Potential Promotions
1	High life quality customers	These are the consumers that have the highest average revenue and volume as well as the least amount of discounted promotions. With the implementation of personalized promotions, this set of consumers could be excluded from most generic offers and could result in savings for the retail stores
2	Skin-care liker	Promotions offered to these consumers can be focused on cross-category bundling.
3	Frequent customers	These are the consumers with the highest number of transactions with a reasonably high percent of average discount. Our interpretation is that since these customers visit stores with a high frequency, they have a higher exposure to the sales being offered in store and so take advantage of that. Further analysis of the purchases of this segment of consumers can help in providing only relevant promotions to them
4	Cherry-picker	These are the set of consumers whose purchases seem to be motivated by discounts. Further analysis on the stores visited and product categories purchased would help in planning for personalized promotions
5	Cooked food liker	Most of the transactions from this set of customers involve cooked and frozen foods. So the stores could look at cross-category promotions to expand the variety of products in this customers' basket as a means to increase revenue
6	Infrequent customers	In case a store wants to increase the footfall from their infrequent customers as a strategy to increase sales, promotions could be targeted specifically at this segment of customers

4.2 Product

Based on the segmentation exercise, we have 10 segments of product categories. Out of these, there are 5 segments which operate on an average discount of more than 40% and each of these 5 segments also generate more than the median revenues of the 10 segments.st discount. It implies that the discounts/promotions merely prepone or hasten the purchase. The following table gives an overview of different product segments, their interpretations and possible business goals which could prompt promotions in those segments.

Segment	Segment Interpretation	Business Goals & Potential Promotions
1	High-price necessities	This consists of several categories or products all of which are characterized by a high price. Further exploration needs to be done determine the correct type of personalized promotion.

2	Frequent discounted daily commodities	They account for a lot of transactions with a relatively low revenue and high discount. Analysis of this segment in the context of customer segmentation could give further insights on what kinds of promotions are to be designed
3	Food ingredients	Usually fresh foods - a segment that doesn't require too many promotions unless for the sake of clearance
4	Less Expensive daily commodities	High number of transactions with relatively low value as well as low discounts
5	Shipping bags & bananas	Extremely low value products and the shopping bags are more of a necessity for shoppers who do not carry their own bags. No promotions needed for this segment
6	Mix of daily commodities	Low value products necessary for sustenance with a high number of transactions. Typically no discounts needed unless targeted at slow-moving products
7	Seldom discounted commodities	Low value products with extremely low discounts with a high number of transactions. No discounts needed unless targeted at slow-moving products
8	Expensive Electronics	High value extremely low frequency purchases originating from very few customers. So extremely personalized promotions can be designed for customers.
9	Small machines & accessories	Segment with the second highest average revenue and low frequency of purchases. Promotions could be offered for specific time periods on a small assortment of products.
10	Mostly Liquids (unit as carat)	Low value products with a mix of perishable and non-perishable consumable liquids. Promotions could be rolled out for the necessary brands or in the case of clearance

4.3 Store

Based on the segmentation exercise, we have 5 segments of stores. The following table gives an overview of different stores, their interpretations and possible business goals which could prompt promotions in those segments.

Segment	Segment Interpretation	Business Goals & Potential Promotions
1	Neighborhood retailer	Promotions aimed at prompting more consumer visits and increasing number of transactions
2	Supply retailer	Similar to a wholesale supermarket but availability and promotions encouraging bulk purchase of a smaller and more specialized assortment of product categories
3	Mega supermarket	Promotions to upsell to consumers in order to increase revenue per transaction
4	Supermarket	More analysis on product assortment and customer segments to create personalized promotions

5	Wholesale supermarket	Revenues generated through bulk sales & promotions could be designed to encourage bulk buying behavior

4.4 Business Implementation

Based on the strategy and need of the retailer, promotions can be designed for an appropriate combination of customer, product and store.

Chapter 5: Appendix

5.1 Customer Segmentation Summary Statistics:

In [109]:
```
customer[customer.kmeans_seg==0].describe().round(2)
```
Out[109]:

	tran_unique	tran_cnt	tran_date_unique	store_unique	store_mode	prod_cnt	qty_sum	gross_sales	discount_sum	discount_avg	net_sales	discount_c
count	1791.00	1791.00	1791.00	1791.00	1791.00	1791.00	1791.00	1791.00	1791.00	1791.00	1791.00	1791.
mean	343.42	3655.95	346.26	6.34	449.24	952.08	5979.88	7989.72	-1250.69	-0.36	6739.03	1200.
std	94.60	651.83	94.83	4.60	201.59	203.99	906.34	612.29	378.94	0.14	547.48	319.
min	117.00	1499.00	119.00	1.00	137.00	279.00	3917.28	5857.98	-2803.77	-1.13	4969.44	335.
25%	278.00	3217.00	281.00	3.00	292.00	819.00	5331.17	7562.46	-1498.10	-0.44	6375.90	971.
50%	325.00	3598.00	328.00	5.00	399.00	956.00	5881.74	8055.16	-1231.67	-0.34	6767.33	1176.
75%	396.00	4031.50	399.00	8.00	588.00	1088.50	6467.26	8480.97	-971.34	-0.25	7138.42	1391.
max	712.00	6408.00	718.00	54.00	999.00	1665.00	9993.21	9224.66	-298.50	-0.07	8207.42	3166.

In [110]:
```
customer[customer.kmeans_seg==1].describe().round(2)
```
Out[110]:

	tran_unique	tran_cnt	tran_date_unique	store_unique	store_mode	prod_cnt	qty_sum	gross_sales	discount_sum	discount_avg	net_sales	discount_c
count	975.00	975.00	975.00	975.00	975.00	975.00	975.00	975.00	975.00	975.00	975.00	975.
mean	346.17	4216.55	348.46	6.98	447.95	1091.28	7209.12	12488.03	-2033.24	-0.53	10454.79	1492.
std	93.14	888.30	93.72	4.83	198.00	256.34	1035.57	956.82	644.73	0.40	882.73	437.
min	99.00	631.00	99.00	1.00	143.00	246.00	2565.65	10905.03	-4986.84	-6.89	7906.98	270.
25%	282.00	3658.50	284.50	4.00	307.00	927.00	6545.30	11784.01	-2431.48	-0.61	9844.15	1186.
50%	329.00	4315.00	331.00	6.00	393.00	1115.00	7219.66	12300.71	-2006.09	-0.48	10303.59	1485.
75%	400.00	4777.00	403.50	9.00	588.00	1263.00	7940.60	12965.94	-1578.04	-0.35	10980.84	1798.
max	715.00	6644.00	715.00	51.00	999.00	1881.00	10149.92	18276.60	-428.09	-0.10	13929.00	3196.

In [111]:
```
customer[customer.kmeans_seg==2].describe().round(2)
```
Out[111]:

	tran_unique	tran_cnt	tran_date_unique	store_unique	store_mode	prod_cnt	qty_sum	gross_sales	discount_sum	discount_avg	net_sales	discount_c
count	1400.00	1400.00	1400.00	1400.00	1400.00	1400.00	1400.00	1400.00	1400.00	1400.00	1400.00	1400.
mean	321.58	3438.83	323.80	7.12	454.89	977.19	5684.46	9869.25	-1647.23	-0.53	8222.02	1232.
std	85.20	710.89	85.56	5.13	203.05	229.64	822.61	723.37	536.88	0.69	670.02	372.
min	38.00	174.00	38.00	1.00	137.00	113.00	1468.25	8527.79	-4241.41	-24.09	6508.40	138.
25%	265.00	3021.00	267.00	4.00	301.00	829.00	5169.43	9293.04	-1978.47	-0.60	7737.20	981.
50%	308.00	3485.50	310.00	6.00	478.00	990.00	5767.81	9743.30	-1623.64	-0.47	8133.10	1212.
75%	369.00	3919.25	373.00	9.00	592.00	1130.25	6301.33	10378.19	-1269.51	-0.35	8635.95	1477.
max	676.00	5589.00	685.00	57.00	999.00	1602.00	7632.86	13674.28	-373.97	-0.10	10873.54	2597.

In [112]:
```
customer[customer.kmeans_seg==3].describe().round(2)
```
Out[112]:

	tran_unique	tran_cnt	tran_date_unique	store_unique	store_mode	prod_cnt	qty_sum	gross_sales	discount_sum	discount_avg	net_sales	discount_c
count	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.00	1037.
mean	386.73	4760.27	390.64	6.34	444.93	1084.21	8173.99	10108.96	-1560.56	-0.34	8548.39	1543.
std	107.77	859.30	108.26	4.68	206.36	229.60	1207.77	887.98	492.29	0.14	751.64	430.
min	149.00	2194.00	149.00	1.00	137.00	430.00	6203.39	7374.24	-3008.41	-1.16	6507.79	479.
25%	304.00	4183.00	307.00	3.00	288.00	932.00	7370.16	9462.72	-1898.59	-0.43	7993.49	1240.
50%	370.00	4693.00	374.00	5.00	388.00	1089.00	7906.90	10215.69	-1542.75	-0.33	8580.70	1503.
75%	449.00	5232.00	453.00	8.00	588.00	1239.00	8657.50	10815.85	-1188.77	-0.24	9115.22	1813.
max	717.00	8581.00	717.00	46.00	999.00	1816.00	16148.74	11840.03	-351.64	-0.05	10489.31	4016.

In [113]:
```
customer[customer.kmeans_seg==4].describe().round(2)
```
Out[113]:

	tran_unique	tran_cnt	tran_date_unique	store_unique	store_mode	prod_cnt	qty_sum	gross_sales	discount_sum	discount_avg	net_sales	discount_c
count	2078.00	2078.00	2078.00	2078.00	2078.00	2078.00	2078.00	2078.00	2078.00	2078.00	2078.00	2078.
mean	294.69	2683.53	297.09	6.46	449.47	823.42	4232.08	6986.31	-1119.48	-0.44	5866.83	916.
std	75.81	539.83	76.25	4.50	197.78	193.10	687.82	689.78	376.36	0.20	588.82	285.
min	83.00	537.00	81.00	1.00	137.00	151.00	1379.59	4744.80	-2964.66	-2.23	3867.10	128.
25%	248.00	2346.00	251.00	3.00	303.00	696.00	3779.51	6492.99	-1350.64	-0.54	5435.15	716.
50%	288.00	2692.50	290.00	5.00	398.00	823.00	4266.86	6932.34	-1089.32	-0.41	5818.19	889.
75%	333.00	3041.75	335.00	9.00	589.00	959.00	4701.01	7452.73	-859.79	-0.31	6244.13	1106.
max	678.00	4424.00	678.00	41.00	999.00	1435.00	6488.68	9196.79	-299.94	-0.09	8353.19	2077.

In [115]:
```
customer[customer.kmeans_seg==5].describe().round(2)
```
Out[115]:

	tran_unique	tran_cnt	tran_date_unique	store_unique	store_mode	prod_cnt	qty_sum	gross_sales	discount_sum	discount_avg	net_sales	discount_
count	639.00	639.00	639.00	639.00	639.00	639.00	639.00	639.00	639.00	639.00	639.00	639
mean	388.38	5682.29	392.23	6.95	445.81	1263.22	10062.24	13828.38	-2278.99	-0.49	11549.40	1971
std	114.81	1164.09	115.73	5.33	207.75	268.07	1697.54	1305.87	779.43	1.27	939.77	530
min	62.00	234.00	61.00	1.00	137.00	166.00	7006.47	10344.45	-7425.23	-31.73	8787.77	333
25%	303.00	5003.00	305.00	4.00	296.50	1084.00	8938.49	12970.99	-2692.54	-0.50	10863.68	1596
50%	368.00	5664.00	374.00	6.00	379.00	1285.00	9710.53	13805.82	-2219.94	-0.39	11602.11	1927
75%	463.00	6373.00	466.00	9.00	588.00	1448.50	10842.55	14622.13	-1723.82	-0.29	12247.30	2294
max	724.00	10790.00	724.00	76.00	999.00	1972.00	19887.35	21444.94	-601.36	-0.10	14019.71	4405

5.2 Customer-Store Segmentation Summary Statistics:

In [137]:
```
customer_store[customer_store.kmeans_seg==0].describe().round(2)
```
Out[137]:

	tran_id		tran_dt	prod_id	tran_prod_sale_qty	tran_prod_sale_amt	tran_prod_discount_amt		tran_prod_paid_amt	tran_prod_offer_cts	prod_uni
	nunique	count	nunique	nunique	sum	sum	sum	mean	sum	sum	
count	1048.00	1048.00	1048.00	1048.00	1048.00	1048.00	1048.00	1048.00	1048.00	1048.00	1
mean	370.30	4758.96	370.30	1075.85	8244.03	12092.19	-1921.86	-0.45	10170.33	1609.30	
std	120.49	1192.31	120.49	255.63	1850.98	1352.07	665.37	0.37	1110.42	491.08	
min	87.00	394.00	87.00	139.00	2965.08	7370.35	-6022.57	-6.46	6898.59	151.00	
25%	284.00	4021.50	284.00	904.00	7024.78	11095.41	-2311.12	-0.53	9372.70	1273.00	
50%	348.00	4669.00	348.00	1071.00	7939.26	11950.65	-1864.19	-0.40	10018.95	1565.50	
75%	447.00	5437.00	447.00	1236.00	9061.07	12932.03	-1465.47	-0.29	10926.56	1892.00	
max	724.00	9655.00	724.00	1849.00	17707.71	17936.85	-471.76	-0.06	13339.11	4245.00	

In [144]:
```
customer_store[customer_store.kmeans_seg==1].describe().round(2)
```
Out[144]:

	tran_id		tran_dt	prod_id	tran_prod_sale_qty	tran_prod_sale_amt	tran_prod_discount_amt		tran_prod_paid_amt	tran_prod_offer_cts	prod_uni
	nunique	count	nunique	nunique	sum	sum	sum	mean	sum	sum	
count	4128.00	4128.00	4128.00	4128.00	4128.00	4128.00	4128.00	4128.00	4128.00	4128.00	4
mean	75.50	715.80	75.50	342.56	1193.33	1856.22	-316.47	-0.60	1539.75	256.52	
std	50.69	292.74	50.69	109.58	458.60	617.84	167.23	3.72	512.98	126.92	
min	3.00	4.00	3.00	4.00	112.84	773.61	-1784.54	-209.58	651.46	0.00	
25%	39.00	491.00	39.00	264.00	826.74	1323.80	-407.90	-0.60	1097.74	163.00	
50%	62.00	668.00	62.00	333.50	1130.62	1768.66	-285.91	-0.42	1467.84	234.00	
75%	100.00	911.25	100.00	415.00	1504.32	2326.68	-194.18	-0.30	1936.68	328.25	
max	383.00	2129.00	383.00	763.00	3150.26	4138.35	0.00	0.00	3157.71	856.00	

In [145]:
```
customer_store[customer_store.kmeans_seg==2].describe().round(2)
```
Out[145]:

	tran_id		tran_dt	prod_id	tran_prod_sale_qty	tran_prod_sale_amt	tran_prod_discount_amt		tran_prod_paid_amt	tran_prod_offer_cts	prod_unit
	nunique	count	nunique	nunique	sum	sum	sum	mean	sum	sum	
count	2908.00	2908.00	2908.00	2908.00	2908.00	2908.00	2908.00	2908.00	2908.00	2908.00	2
mean	273.42	2714.74	273.42	779.81	4418.20	6728.88	-1065.78	-0.45	5663.11	911.23	
std	97.76	631.33	97.76	185.75	885.87	781.34	370.60	1.17	677.95	290.11	
min	15.00	28.00	15.00	19.00	1591.87	4496.61	-3282.95	-61.75	3975.00	13.00	
25%	209.00	2308.00	209.00	653.00	3804.55	6140.35	-1290.36	-0.52	5141.57	710.00	
50%	269.00	2703.00	269.00	783.00	4353.06	6709.55	-1031.80	-0.38	5637.11	887.00	
75%	329.00	3099.25	329.00	903.00	4950.00	7291.13	-806.28	-0.28	6146.64	1092.25	
max	711.00	5385.00	711.00	1392.00	8572.50	9842.25	-124.62	-0.06	8309.59	2533.00	

In [146]:
```
customer_store[customer_store.kmeans_seg==3].describe().round(2)
```
Out[146]:

	tran_id		tran_dt	prod_id	tran_prod_sale_qty	tran_prod_sale_amt	tran_prod_discount_amt		tran_prod_paid_amt	tran_prod_offer_cts	prod_unit
	nunique	count	nunique	nunique	sum	sum	sum	mean	sum	sum	
count	2064.00	2064.00	2064.00	2064.00	2064.00	2064.00	2064.00	2064.00	2064.00	2064.00	2
mean	332.01	3722.52	332.01	921.38	6224.79	9005.18	-1409.39	-0.42	7595.79	1234.51	
std	114.83	869.75	114.83	220.79	1237.27	972.25	493.06	0.26	831.59	382.99	
min	58.00	525.00	58.00	124.00	2445.94	6249.74	-3757.72	-6.01	5242.27	162.00	
25%	253.75	3182.75	253.75	781.00	5406.63	8279.73	-1714.15	-0.51	6967.23	973.00	
50%	314.00	3701.50	314.00	931.50	6085.56	8952.15	-1367.42	-0.38	7531.50	1210.00	
75%	400.00	4250.00	400.00	1065.25	6928.44	9690.80	-1046.28	-0.26	8163.75	1454.25	
max	716.00	7091.00	716.00	1684.00	13690.79	13015.42	-279.64	-0.06	10606.21	3685.00	

In [147]:
```
customer_store[customer_store.kmeans_seg==4].describe().round(2)
```
Out[147]:

	tran_id		tran_dt	prod_id	tran_prod_sale_qty	tran_prod_sale_amt	tran_prod_discount_amt		tran_prod_paid_amt	tran_prod_offer_cts	prod_u
	nunique	count	nunique	nunique	sum	sum	sum	mean	sum	sum	
count	39929.00	39929.00	39929.00	39929.00	39929.00	39929.00	39929.00	39929.00	39929.00	39929.00	
mean	5.90	48.34	5.90	36.83	80.72	124.38	-20.94	-0.53	103.44	17.09	
std	10.69	76.95	10.69	49.97	127.74	193.53	37.93	1.70	160.60	30.22	
min	1.00	1.00	1.00	1.00	0.09	0.06	-473.09	-105.00	0.00	0.00	
25%	1.00	6.00	1.00	6.00	8.72	12.60	-22.49	-0.54	10.76	1.00	
50%	2.00	16.00	2.00	15.00	26.71	40.93	-5.29	-0.27	34.42	5.00	
75%	6.00	53.00	6.00	46.00	88.51	137.29	-0.68	-0.08	114.77	18.00	
max	248.00	900.00	248.00	337.00	1372.62	1208.30	0.00	0.00	1171.45	372.00	

In [148]:
```
customer_store[customer_store.kmeans_seg==5].describe().round(2)
```
Out[148]:

	tran_id		tran_dt	prod_id	tran_prod_sale_qty	tran_prod_sale_amt	tran_prod_discount_amt		tran_prod_paid_amt	tran_prod_offer_cts	prod_unit
	nunique	count	nunique	nunique	sum	sum	sum	mean	sum	sum	
count	2491.00	2491.00	2491.00	2491.00	2491.00	2491.00	2491.00	2491.00	2491.00	2491.00	2
mean	166.50	1672.73	166.50	592.13	2774.97	4288.86	-708.34	-0.80	3580.52	590.32	
std	80.57	483.38	80.57	153.37	710.94	767.55	276.88	13.68	656.59	220.79	
min	1.00	3.00	1.00	3.00	540.00	2501.40	-2098.59	-669.23	2125.10	11.00	
25%	105.00	1337.00	105.00	489.50	2235.61	3665.42	-871.02	-0.57	3040.27	434.00	
50%	157.00	1642.00	157.00	588.00	2716.79	4264.80	-675.85	-0.41	3574.17	563.00	
75%	217.00	1992.50	217.00	693.50	3230.47	4915.01	-512.09	-0.30	4091.42	722.00	
max	535.00	3403.00	535.00	1152.00	10481.86	6313.84	-48.92	-0.05	5859.75	1581.00	