MKT 680 - Marketing Analytics

Project 3: Pricing Model

Group 4: Carl Xi, Frank Fan, Jie Zhu, Yaping Zhang

With over 10,000 products in 400+ categories, Pernalonga is an undisputed leader in the Lunitunia retail space. Our project aims to help Pernalonga conduct segmentation analysis on their customers, products, and stores for marketing analytics purposes. Below is an excerpt that speaks about Pernalonga's partnership and sales strategies:

" Pernalonga regularly partners with suppliers to fund promotions and derives about 30% of its sales on promotions. While a majority of its promotion activities are in-store promotions, it recently started partnering with select suppliers to experiment on personalized promotions. In theory, personalized promotions are more efficient as offers are only made to targeted individuals who require an offer to purchase a product. In contrast, most in-store promotions make temporary price reductions on a product available to all customers whether or not a customer needs the incentive to purchase the product. The efficiency of personalized promotion comes from an additional analysis required on customer transaction data to determine which customers are most likely to purchase a product to be offered in order to maximize the opportunity for incremental sales and profits."

1. Background and Assumptions

1.1 Problem Statement

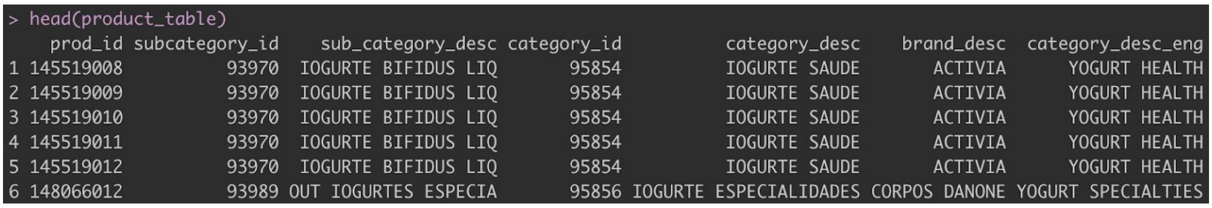
Our analytics consulting firm first successfully delivered a segmentation analysis of Pernalonga's customers, stores and products. We then created a recommender system for Pernalonga's inventory of Nestlé chocolate confectioneries to different customer segments. Now, we are here to improve Pernalonga's revenue by revising list (shelf) prices. Pernalonga is constrained in resources to only implement pricing change to products effective the week of Apr. 13th, 2020. Additionally, Pernalonga will only be able to select the same 100 products within 2 product categories to adjust prices across a meager 10 stores. As such, this is more of a trial run for Pernalonga, and we hope to use our analytics results to further encourage the store chain to implement analytics-driving decision making and pricing.

1.2 Setup

We first assumed that timing and product temporary price reduction levels (essentially the promotion schedule) for the week of Apr. 13th, 2020 would be the same as the one for Apr.13th, 2017. This is because supplier-funded in-store temporary price reductions on products are still under negotiation. We have devised a pricing model that will noticeably improve Pernalonga's expected revenue while still maintaining overall profitability.

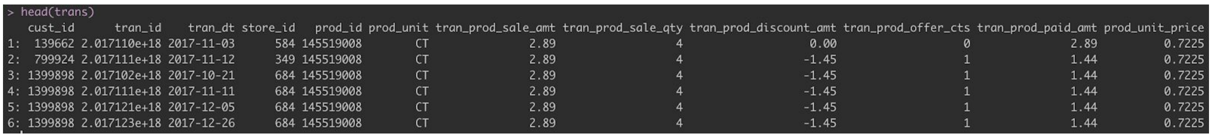
Just like the previous two projects, we will be using the same two datasets. *transaction_table.csv* contains transaction data (purchase history) of approximately 8000 customers between 2016 and 2017, while *product_table.csv* contains product data, including their categories, subcategories, and descriptions for about 11,000 products. The transaction dataset has about 29.62 million entries and 12 variables, while the product table dataset has about 10.8k entries and 7 variables. Below is a snapshot of how the datasets look like:

Product Table:



```
> head(product_table)
    prod_id subcategory_id    sub_category_desc category_id          category_desc    brand_desc  category_desc_eng
1 145519008          93970  IOGURTE BIFIDUS LIQ       95854          IOGURTE SAUDE       ACTIVIA      YOGURT HEALTH
2 145519009          93970  IOGURTE BIFIDUS LIQ       95854          IOGURTE SAUDE       ACTIVIA      YOGURT HEALTH
3 145519010          93970  IOGURTE BIFIDUS LIQ       95854          IOGURTE SAUDE       ACTIVIA      YOGURT HEALTH
4 145519011          93970  IOGURTE BIFIDUS LIQ       95854          IOGURTE SAUDE       ACTIVIA      YOGURT HEALTH
5 145519012          93970  IOGURTE BIFIDUS LIQ       95854          IOGURTE SAUDE       ACTIVIA      YOGURT HEALTH
6 148066012          93989 OUT IOGURTES ESPECIA       95856 IOGURTE ESPECIALIDADES CORPOS DANONE YOGURT SPECIALTIES
```

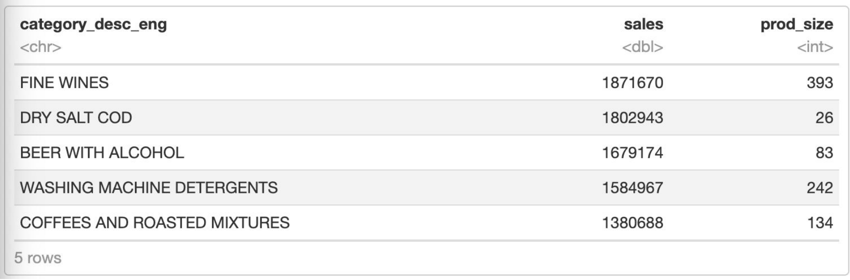
Transactions Table:



```
> head(trans)
   cust_id      tran_id    tran_dt store_id   prod_id prod_unit tran_prod_sale_amt tran_prod_sale_qty tran_prod_discount_amt tran_prod_offer_cts tran_prod_paid_amt prod_unit_price
1:  139662 2.017110e+18 2017-11-03      584 145519008        CT               2.89                  4                   0.00                   0               2.89          0.7225
2:  799924 2.017111e+18 2017-11-12      349 145519008        CT               2.89                  4                  -1.45                   1               1.44          0.7225
3: 1399898 2.017102e+18 2017-10-21      684 145519008        CT               2.89                  4                  -1.45                   1               1.44          0.7225
4: 1399898 2.017111e+18 2017-11-11      684 145519008        CT               2.89                  4                  -1.45                   1               1.44          0.7225
5: 1399898 2.017121e+18 2017-12-05      684 145519008        CT               2.89                  4                  -1.45                   1               1.44          0.7225
6: 1399898 2.017123e+18 2017-12-26      684 145519008        CT               2.89                  4                  -1.45                   1               1.44          0.7225
```

2. Identify Target Products and Categories

Before we dive straight into modeling, we have to make sure to clean our data and select only the subset necessary. The first and most important thing would be to identify and exclude all 'fresh' products from our following analysis, as Pernalonga explicitly told us to not target these products for price changes. However, we decided to retain the data within our dataset as they could act as complements or substitutes for other goods in the future. Additionally, we also calculated the total sales of the different categories and focused our analysis on only the top 5, as they account for a significant share of the sales (which would have the biggest reaction to price changes), as well as Pernalonga's 2 category constraint as mentioned in part 1. Below is a table of the top 5 categories:

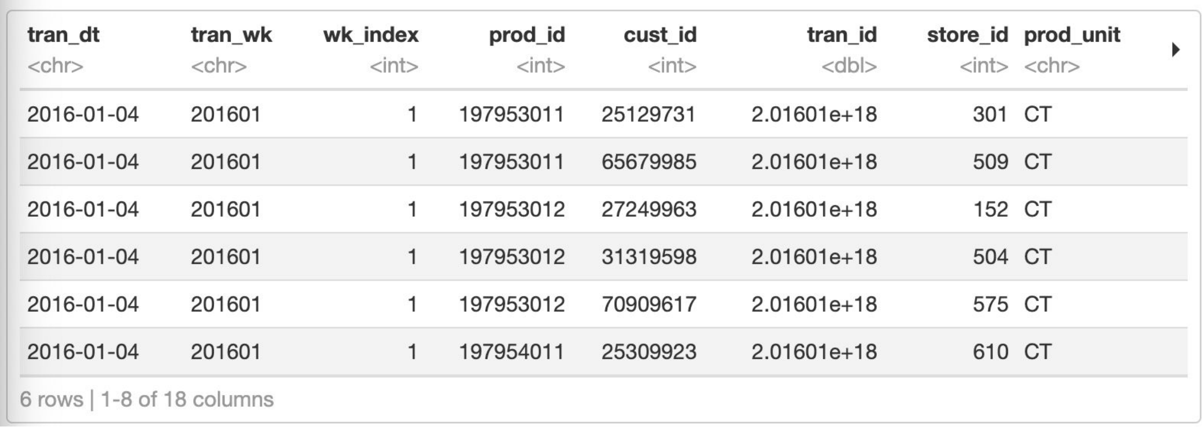


category_desc_eng	sales	prod_size
<chr>	<dbl>	<int>
FINE WINES	1871670	393
DRY SALT COD	1802943	26
BEER WITH ALCOHOL	1679174	83
WASHING MACHINE DETERGENTS	1584967	242
COFFEES AND ROASTED MIXTURES	1380688	134

5 rows

2.1 Adding Week Indexes

There are two important problems with our pricing data. Firstly, price adjustments typically happened once in a while rather than daily. Secondly, we do not have the demand data for every single data within our time period. Even when we did have daily demand data the data was very noisy. To solve all these problems, we have calculated the weekly average price of products, which will smooth out the noise in the daily data. Calculating price elasticity on a weekly basis will ultimately allow us to better observe price changes. The created variable ‘wk_index’ tells us which week the transaction belongs to, as you can see in the header below:



tran_dt	tran_wk	wk_index	prod_id	cust_id	tran_id	store_id	prod_unit
<chr>	<chr>	<int>	<int>	<int>	<dbl>	<int>	<chr>
2016-01-04	201601	1	197953011	25129731	2.01601e+18	301	CT
2016-01-04	201601	1	197953011	65679985	2.01601e+18	509	CT
2016-01-04	201601	1	197953012	27249963	2.01601e+18	152	CT
2016-01-04	201601	1	197953012	31319598	2.01601e+18	504	CT
2016-01-04	201601	1	197953012	70909617	2.01601e+18	575	CT
2016-01-04	201601	1	197954011	25309923	2.01601e+18	610	CT

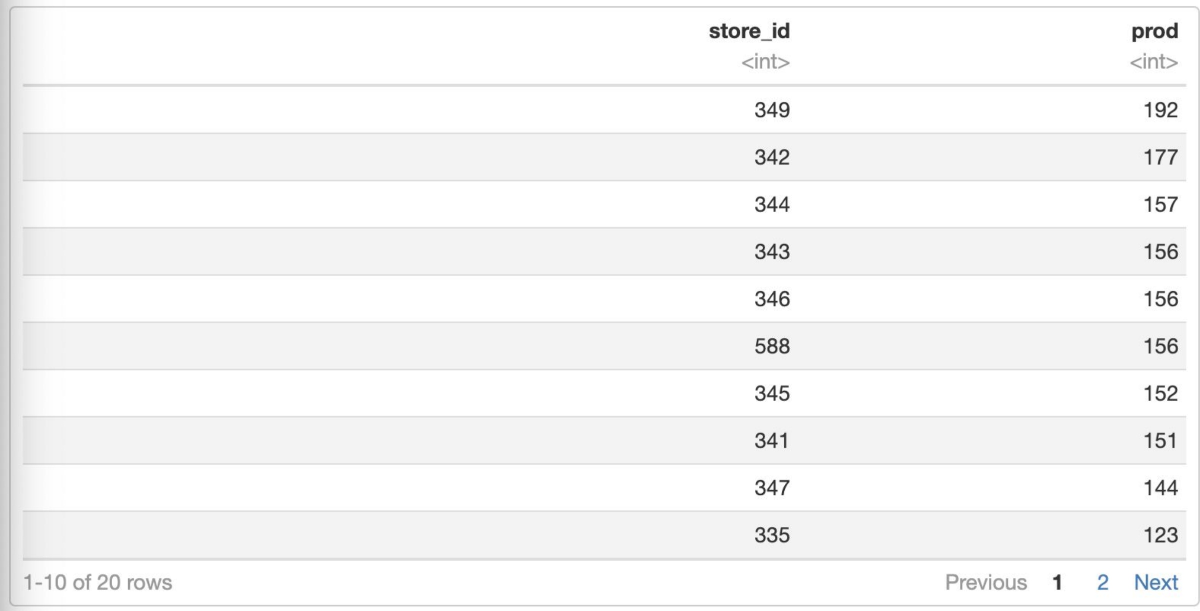
6 rows | 1-8 of 18 columns

2.2 Removing products with fixed prices

Some of the products within our dataset do not have weekly price variations, which make them difficult to work with. We simply cannot estimate the change in demand under the influence of price changes of products with no price change, so we will be excluding them from our calculations, analysis and pricing model/campaign. As we cannot calculate their cross elasticity, we are also unable to analyze them as complements or substitutes. Thus, we have deleted all store-product combinations that do not have weekly price variations. Of the 333,757 store-product combinations, we will only be analyzing 14,660 of them due to this filter. This equals to about 494 unique products across 408 stores.

3. Identify Target Stores

As stated at the very beginning, Pernalonga's constrained resources allow us to only focus on our pricing campaign on 100 products across 10 stores. As we want to maximize the number of common products that our target 10 stores share, we decided to first identify the store with the highest number of target products, then find the other stores that have the most number of common target products with our 'top' store. This will allow us to concentrate our efforts on the stores that will be impacted the most by our pricing model, effectively maximizing 'bang for the buck'. The store with the most number of target products had 192 products, identified by the store id of 349:



store_id	prod
<int>	<int>
349	192
342	177
344	157
343	156
346	156
588	156
345	152
341	151
347	144
335	123

1-10 of 20 rows

Previous
1
2
Next

We then identified the stores with the most number of common items with store 349, which were stores 342, 343, 346, 344, 341, 345, 588, 347, 395, 335, 320, 157, 348, 525, 572, 627, 331, 194, 398. We decided to keep 20 stores in total for our analysis moving forwards. By filtering to only include data on these 20 stores, our dataset only has about 3.28 million entries, with 42560 unique store and product combinations, 2,552 of which are for target products. We have 381 actual target products across the 20 stores that we have chosen. To narrow down our product and store count down to our constraints of 100 products and 10 stores, we will next model the optimal price change and potential profits.

4. Data Preparation

In order to build a model to estimate demand and revenue for our chosen products, we need to begin preparing the training dataset and generate necessary features. We want to incorporate the following aspects within our model:

- Weekly price
- Weekly discount
- Seasonality

- Holiday effects
- Price of complements and substitutes

4.1 Weekly Price and Weekly Discount

Both price and discount are important features that could influence demand. We have talked about why we want to use weekly aggregated data. We took the weekly average price and weekly average discount as variables in the response functions. Later, using the model's result, we will vary the weekly price over a range to find what is the optimal price change that maximizes Pernalonga's revenue.

4.2 Seasonality and Holiday Effects

In order to capture seasonality, we decided to introduce a variable week index denoted by 'wk_index' to our response function to capture seasonal trends. Week index serves as a control variable in our model. By introducing it into our model, we can isolate the effect of the price change on demand from seasonal demand fluctuations.

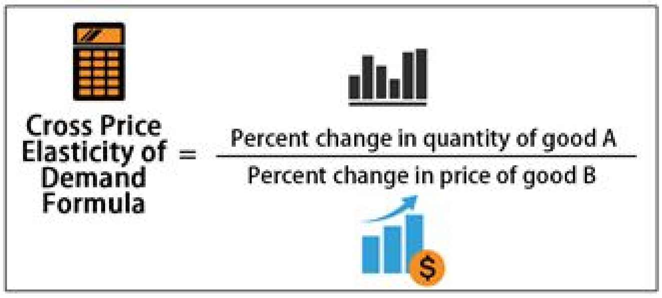
Similarly, holidays have significant effects on demand. We need to account for this effect in the response function as well. Because holiday effects will generally last for a few days, we aggregate them to a weekly level variable as well. For example, because 2017-12-25 is Christmas Day, we have labeled all records that happened in that week as 'wkly_holiday=1'.

The above two variables 'wk_index' and 'wkly_holiday' will be used in our model as controls.

4.3. Price of complements and substitutes

Complements and substitutes are also important factors when we are trying to optimize prices. The price of complementary goods and the price of substitute goods have a significant influence on products' demand. As the price of complementary goods decreases, the demand for a product will increase; as the price of substitute goods decreases, the demand for a product will decrease. So, we need to account for the price of both complementary goods and substitute goods in our response function.

Then how could we find the complements and substitutes for each product? We decided to use the concept of cross elasticity.


Cross Price Elasticity of Demand Formula =
Percent change in quantity of good A
Percent change in price of good B

As denoted by the above formula, cross-price elasticity between our target product A and product B is the percentage change in the quantity of good A divided by the percentage change in the price of good B. If product B is a complementary good of product A, A and B will have negative cross elasticity; if product B is a substitute good of product A, A and B will have positive cross elasticity.

For example, in Store 342, we have a target product 999168803, which belongs to the category 'BEER WITH ALCOHOL'. By calculation, we identified the product 999746995 has the lowest negative cross elasticity with it, so these two products can be seen as complements.

By calculating the cross elasticity between all target products and other products in the dataset, we have ranked the cross elasticity and found the complement and substitute for each target product. Thus, we are able to incorporate the price of complement and the price of a substitute in our model.

5. Modeling

5.1 Response Function

We have tested out three different price response functions: linear response function, constant elasticity price response function and logit response function. We chose the logit response function for our final model because: first, it gives the best cross-validation result, measured by Root Mean Squared Error. Second, the elasticity of the logit response function is non-constant and increases in magnitude as price increases. Because we are trying to predict the demand for different potential new prices, it makes more sense for us to use the logit price response function.

5.2 Formula

From the previous step of data preparation, we have all the variables we need for modeling. After the logit transformation of quantity demanded, we defined the linear regression formula as below:

t_volume ~ wkly_price + wkly_dct + wk_index + wkly_holiday + comp_price + sub_price

- **t_volume**: the target variable. The logit transformation of demand
- **wkly_price**: aggregated average weekly price for each product
- **wkly_dct**: aggregated average discount level for each product. A control variable

- **wkly_index**: week index for each record. A control variable for seasonal trends
- **wkly_holiday**: whether a particular week has holidays or not. A control for holiday effects
- **comp_price**: the price of complementary goods for this product
- **sub_price**: the price of substitute goods for this product

5.3 Test Dataset and Prediction

The next question becomes "at what price level we should predict the demand?". The team ultimately decided that it is not wise to vary the price a lot. In retail, if the price falls below the cost, profitability cannot be guaranteed; if the price rises too high, customer loyalty will potentially be hurt. In this case, we decided to allow the price to vary between 70% of the product's most recent price and 130% of the most recent price, taking the step of 1% at a time. By doing so, we are able to create the test dataset.

For other variables in the test dataset, because we are predicting the week of '2020-04-13 to 2020-04-19', the week index is 16 and 'wkly_holiday' equals 1 (Easter). And we used the same discount information during the corresponding period last year as 'wkly_discount'.

In the end, we are able to predict the demand for each target product at each store at 60 different potential price levels. Using these results, in the next step, we will determine what is the optimal price for each product and decide the final selection of products to target.

6. Results

6.1. Final selection of target products

As mentioned before, we have 381 products across 20 stores as potential candidates for price change. After we ran the model and got the predicted demand for all potential new price levels for our product, we took the following steps to narrow down to the final products and stores to target:

1. We estimated the cost for each product in our dataset to be 95% of the product's historical lowest shelf price, based on the assumption that grocery stores typically have less than 5% sales revenue as margins.
2. We calculated the projected profits after the price change and selected the optimal new suggested price for each product that maximizes incremental profits.
3. We found the optimal 10 stores and 2 categories combination that brings the most incremental profits. We also made sure that the 2 categories that we have chosen derive the highest average incremental revenue of each product in those categories.
    - The selected two categories are: "**Dry Salt Cod**" and "**Beer with Alcohol**".
4. We finalized the top 100 products based on projected profits increase.
    - The selected 10 stores are: **342, 349, 331, 343, 344, 346, 395, 588, 345, 341**

To save space, we will not type out all 100 selected products here, but please refer to the code output that is a part of our submission for the detailed list.

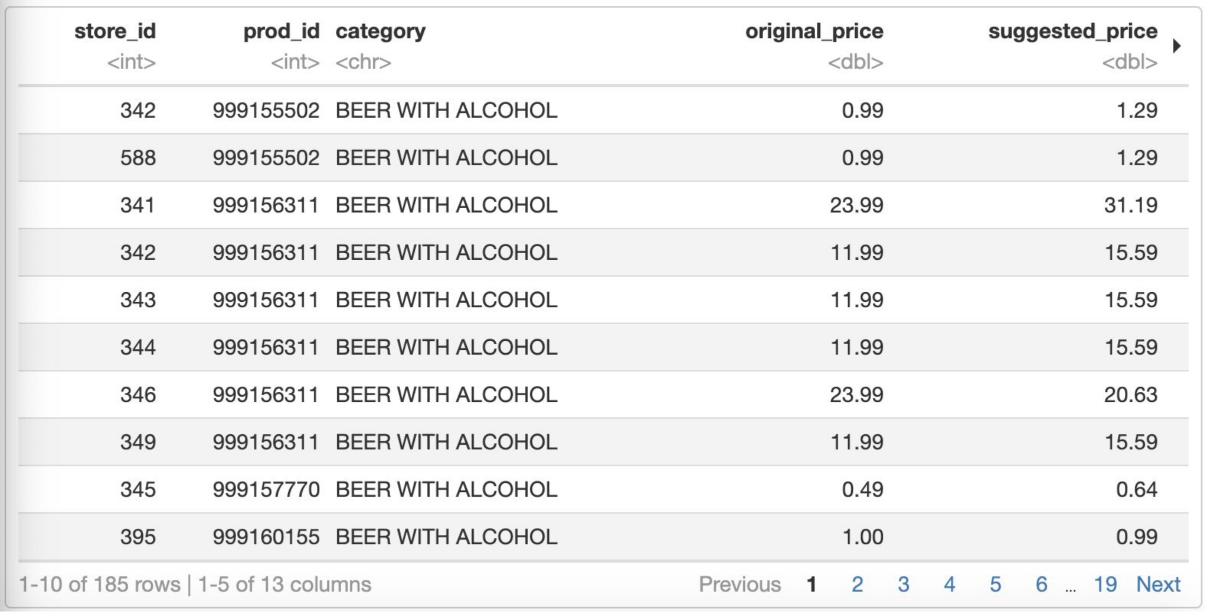
6.2. Estimated Achievements

After we optimized the prices for our selected 100 products across 2 categories in ten stores, the following results are estimated to be achieved (Please refer to our code to see how we calculated the estimated achievements):

Store_id	Increase in Sales	Increase in Revenue	Increase in Profit
331	69.40	1147.11	676.90
341	55.50	313.72	183.13
342	93.36	886.06	495.63
343	76.82	233.97	159.49
344	81.35	472.69	330.83
345	14.57	342.23	198.64
346	90.74	284.97	152.40
349	105.12	454.71	192.46
395	64.90	342.96	195.52
588	34.38	216.71	119.61

Increase in Total Sales Volume	686 Units
Increase in Total Revenue	4,695.17 Price Units
Increase in Total Profit	2,704.62 Price Units

Our final results, which shows every store-product combination post-pricing adjustment, looks like the screenshot below. You can find a full list within our submissions.



store_id	prod_id	category	original_price	suggested_price
<int>	<int>	<chr>	<dbl>	<dbl>
342	999155502	BEER WITH ALCOHOL	0.99	1.29
588	999155502	BEER WITH ALCOHOL	0.99	1.29
341	999156311	BEER WITH ALCOHOL	23.99	31.19
342	999156311	BEER WITH ALCOHOL	11.99	15.59
343	999156311	BEER WITH ALCOHOL	11.99	15.59
344	999156311	BEER WITH ALCOHOL	11.99	15.59
346	999156311	BEER WITH ALCOHOL	23.99	20.63
349	999156311	BEER WITH ALCOHOL	11.99	15.59
345	999157770	BEER WITH ALCOHOL	0.49	0.64
395	999160155	BEER WITH ALCOHOL	1.00	0.99

1-10 of 185 rows | 1-5 of 13 columns

6.3. Future Improvements

As Pernalonga is heavily constrained by its resources during this pricing experimentation, we hope to expand our analysis to even more products, across more categories, across more stores. As well, we hope to examine the effects of our price changes on each customer segment's sales quantity, revenue and profitability. We were also constrained to fresh products for this project, but we hope to extend our analysis to more product types in the future. We were also forced to make many assumptions, such as how timing and the promotion schedule for the week we are targeting is the same as the one from three years ago. With more data, it would be interesting to see trends in promotion schedules over weeks and years in order to utilize a better promotional schedule for our model. On this note, we also hope to extend our pricing modeling to beyond the one week. We are confident that our results will prove to Pernalonga that analytics-driven decision making and pricing is worth the investment in the long run.