Big Data Integration Final Paper

Sidney Schaeper, Aman Rastogi, Keya Satpathy, Bhawna Saini, Prathiba Swamykannu

2/20/2020

### Understanding Our Topic

We had a variety of questions we wanted to answer in this project. We formualted these questions after evalauting our 84.51 datasets and the twitter data. The following are the questions we hoped to answer.

* Should we continue to do business with our five worst performing brands based on the perspective of customers?
* Does location of the product within the store and ad impact sales of the product? What is the optimal location within the store and ads?
* What amounts of each product should we plan to have in inventory on average daily based on these sales?

For the first question, we thought that the twitter dataset, product\_lookup dataset, and transactions dataset would provide us insight into answering this question. For the second question, we thought that the transactions dataset and casual\_lookup dataset would provide us information into answering this question. For the third question, we thought that the transactions dataset, store\_lookup dataset, and product\_lookup dataset would help us discover the answer to this question.

We thought the first question would be insightful to the business, because a company should know what customers think of the brands they partner with. A company likely wouldn’t want to work with a brand that has a bad image to its customers, because they likely wouldn’t sell many of the products associated with that brand. In addition, we considered the second question to be a great question to answer, because we thought this could help Kroger’s marketing strategy. The marketing strategy would be improved with this knowledge, because the marketers would have more knowledge of the optimal locations for high product sales in the weekly ad and stores. The final question would be great for a busines to know, because this knowledge would help improve a company’s inventory strategy. This strategy would improve, because you possibly could improve your inventory prediction levels for each store from using the data instead of just guessing.

### Describe The Tools Used

The tool that we used for this project was R Studio.

#Packages that needed to be installed for the project#  
install.packages("readr")  
install.packages("dplyr")  
install.packages("tidyr")  
install.packages("kableExtra")

#Packages that need to be found for the project#  
library(readr)  
library(dplyr)  
library(tidyr)  
library(kableExtra)

### 84.51 Data Source

The data that we wanted to evaluate for this project came from two sources. The first source was from [84.51’s website](https://www.8451.com/). The folder of data that we used from 84.51 are called the Carbo Loading datasets. There are a total of four datasets within this folder. The four datasets are called the casual\_lookup, product\_lookup, store\_lookup, and transactions. According to the Carbo Loading guide provided, these datasets came from a relational database. Also, these datasets contain purchases at a household level over two years, and the datasets are filtered to only contain products from four categories. These four categories are pasta, pasta sauce, syrup, and pancake mix. In addition, the guide contained information pertaining to the variables within each of the datasets. The tables below show these variables and their description.

**Transactions Dataset Table**

|  |  |
| --- | --- |
| Variable | Description |
| upc | It is a standard 10 digit code assigned to products. This is the product that was purchased. |
| dollar\_sales | The amount of money spent on this product by the customer. These are recorded in dollars. |
| units | The quantity of this product purchased. |
| time\_of\_transaction | The time the transaction occurred. This is recorded in military time. |
| geography | This label tells you where it was purchased out of the two large regions. These two regions consist of multiple states. The value can either be 1 or 2. |
| week | This notifies the week that the transaction occured. The range of values is from 1 to 104. These numbers are assigned chronologically. |
| household | This value is a unique number assigned to a household. This is the purchaser of the product. |
| store | This value is a unique number assigned to each store. This is where the product was purchased. |
| basket | This is a unique number assigned to a trip to the store. This is the trip that this product was assigned to. |
| day | This is the day that this product was purchased. The range of values is 1 to 728. |
| coupon | This is dummy variable to notify whether a coupon was used. The possible value is 1 or 0. 1 means a coupon was used. 0 means a coupon wasn’t used. |

**Store Lookup Dataset Table**

|  |  |
| --- | --- |
| Variable | Description |
| store | This value is a unique number assigned to each store. |
| store\_zip\_code | This is the 5 digit zip code for the store. |

**Product Lookup Dataset Table**

|  |  |
| --- | --- |
| Variable | Description |
| upc | This is the standard 10 digit code assigned to this product. |
| product\_description | This details the product. This likely contains the name of the product. |
| commodity | This is the category of the product. The four possibly values are pasta, pasta sauce, pancake mix, or syrup. |
| brand | This is the brand name of the product. |
| product\_size | This is the size of the product. These aren’t all in the same measurement. |

**Casual Lookup Dataset Table**

|  |  |
| --- | --- |
| Variable | Description |
| upc | This is the standard 10 digit code assigned to this product. |
| store | This value is a unique number assigned to each store. |
| week | This notifies the week that the transaction occured. The range of values is from 1 to 104. These numbers are assigned chronologically. |
| feature\_desc | This is where the product is located on the weekly ad. |
| display\_desc | This is where the product is displayed in the store. |
| geography | This label tells you where it was purchased out of the two large regions. These two regions consist of multiple states. The value can either be 1 or 2. |

#Set the working directory#  
setwd("~/8451\_Carbo-Loading/Carbo-Loading CSV")  
  
#Read the datasets into R#  
causal <- read\_csv("causal\_lookup.csv")  
transaction <- read\_csv("transactions.csv")

#View snippets of the dataset#  
head(causal)

## # A tibble: 6 x 6  
## upc store week feature\_desc display\_desc geography  
## <chr> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 7680850108 1 68 Wrap Interior Feature Not on Display 1  
## 2 5100001212 1 66 Wrap Back Feature Not on Display 1  
## 3 5100002792 1 72 Interior Page Feature Not on Display 1  
## 4 3620000300 1 55 Wrap Interior Feature Not on Display 1  
## 5 4112907742 1 68 Wrap Interior Feature Not on Display 1  
## 6 3620000250 1 55 Wrap Interior Feature Not on Display 1

head(transaction)

## # A tibble: 6 x 11  
## upc dollar\_sales units time\_of\_transac~ geography week household store  
## <chr> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 7680~ 0.8 1 1100 2 1 125434 244  
## 2 3620~ 3.59 1 1100 2 1 125434 244  
## 3 1800~ 2.25 1 1137 2 1 108320 244  
## 4 9999~ 0.85 1 1148 2 1 162016 244  
## 5 9999~ 2.19 1 1323 2 1 89437 244  
## 6 5100~ 2.19 1 1323 2 1 89437 244  
## # ... with 3 more variables: basket <dbl>, day <dbl>, coupon <dbl>

### Twitter Data Source

The final source of data we used for our projet was from [Twitter](http://apps.twitter.com). We had to create a developer account in order to get access to these tweets. In this project, we were hoping to pull this data manually to get customer opinions about brands. The way we pulled these tweets down manually was through using the search tweets package in R, and we searched through twitter to find these tweets by searching on the brand name and the food category of the brand. Next, we cleaned these results by focusing just on the tweet, removing emojis, removing urls, lowering the case of the tweets, removing the punctation, removing the numbers, removing stop words, removing white space, and removing additional words that we don’t want to evaluate. After we cleaned the tweets, we separated the tweets into words, counted these words, and filtered on the top 25 words. After collecting these results, we created a data frame from all these results. If a brand and category didn’t show a result, we put NA for these word results. Below is a table of the dataset we created manually, and this table shows the variables and description.

**Twitter Dataset Table**

|  |  |
| --- | --- |
| Variable | Description |
| brand | This is the name of the brand. |
| word | These are the words that showed up in the top 25 filter. |
| freq | These are the counts of the word showing up in the tweets. |

### Data Integration Process

#Clean the files before the inner join for Question 2#  
causal\_select <- causal %>%  
 select(upc, store, week, feature\_desc, display\_desc)  
transaction\_clean <- transaction %>%  
 filter(week > 42) %>%  
 select(upc, dollar\_sales, units, week, store)  
  
#Inner join for Question 2#  
causal\_trans <- inner\_join(causal\_select, transaction\_clean, by = c("upc", "store", "week"))  
tibble(causal\_trans)

## # A tibble: 381,746 x 1  
## causal\_trans$upc $store $week $feature\_desc $display\_desc $dollar\_sales  
## <chr> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 3620000300 1 55 Wrap Interio~ Not on Displ~ 1.5  
## 2 4112907742 1 68 Wrap Interio~ Not on Displ~ 2   
## 3 4420979129 1 66 Interior Pag~ Not on Displ~ 2.5  
## 4 3620001375 1 56 Interior Pag~ Not on Displ~ 1.5  
## 5 4112907700 1 68 Wrap Interio~ Not on Displ~ 2   
## 6 3620001368 1 56 Interior Pag~ Not on Displ~ 3   
## 7 7680851613 1 67 Wrap Interio~ Not on Displ~ 0.8  
## 8 7680851613 1 67 Wrap Interio~ Not on Displ~ 1.6  
## 9 3620000488 1 56 Interior Pag~ Not on Displ~ 1.5  
## 10 3620000482 1 56 Interior Pag~ Not on Displ~ 1.5  
## # ... with 381,736 more rows, and 1 more variable: $units <dbl>

### Schema Alignment

#### Record Linkage

#### Data Fusion

### Final Results and Analyzing These Results

#Editing of the Combined Dataset to Find the Answers#  
#Feature Description#  
causal\_trans\_feature\_dollar <- causal\_trans %>%  
 select(feature\_desc, dollar\_sales)%>%  
 group\_by(feature\_desc) %>%  
 summarise(Total\_Dollar\_Sales = sum(dollar\_sales, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Dollar\_Sales))  
causal\_trans\_feature\_dollar

## # A tibble: 8 x 2  
## feature\_desc Total\_Dollar\_Sales  
## <chr> <dbl>  
## 1 Interior Page Feature 240347.  
## 2 Not on Feature 172678.  
## 3 Front Page Feature 105429.  
## 4 Wrap Interior Feature 22507.  
## 5 Back Page Feature 17707.  
## 6 Wrap Front Feature 11098.  
## 7 Wrap Back Feature 9492.  
## 8 Interior Page Line Item 9378.

causal\_trans\_feature\_units <- causal\_trans %>%  
 select(feature\_desc, units)%>%  
 group\_by(feature\_desc) %>%  
 summarise(Total\_Units = sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Units))  
causal\_trans\_feature\_units

## # A tibble: 8 x 2  
## feature\_desc Total\_Units  
## <chr> <dbl>  
## 1 Interior Page Feature 196841  
## 2 Not on Feature 136526  
## 3 Front Page Feature 106489  
## 4 Wrap Interior Feature 19338  
## 5 Back Page Feature 17476  
## 6 Wrap Front Feature 11210  
## 7 Wrap Back Feature 7467  
## 8 Interior Page Line Item 5965

causal\_trans\_feature\_units\_dollar <- causal\_trans %>%  
 select(feature\_desc, units, dollar\_sales)%>%  
 group\_by(feature\_desc) %>%  
 summarise(Sales\_Divided\_By\_Units = sum(dollar\_sales, na.rm = TRUE)/sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Sales\_Divided\_By\_Units))  
causal\_trans\_feature\_units\_dollar

## # A tibble: 8 x 2  
## feature\_desc Sales\_Divided\_By\_Units  
## <chr> <dbl>  
## 1 Interior Page Line Item 1.57   
## 2 Wrap Back Feature 1.27   
## 3 Not on Feature 1.26   
## 4 Interior Page Feature 1.22   
## 5 Wrap Interior Feature 1.16   
## 6 Back Page Feature 1.01   
## 7 Front Page Feature 0.990  
## 8 Wrap Front Feature 0.990

#Display\_desc#  
causal\_trans\_display\_dollar <- causal\_trans %>%  
 select(display\_desc, dollar\_sales)%>%  
 group\_by(display\_desc) %>%  
 summarise(Total\_Dollar\_Sales = sum(dollar\_sales, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Dollar\_Sales))  
causal\_trans\_display\_dollar

## # A tibble: 11 x 2  
## display\_desc Total\_Dollar\_Sales  
## <chr> <dbl>  
## 1 Not on Display 324395.  
## 2 Rear End Cap 79471.  
## 3 In-Shelf 63270.  
## 4 Front End Cap 31889.  
## 5 Secondary Location Display 23581.  
## 6 In-Aisle 21124.  
## 7 Promo/Seasonal Aisle 13619.  
## 8 Mid-Aisle End Cap 13288.  
## 9 Store Rear 10016.  
## 10 Store Front 6147.  
## 11 Side-Aisle End Cap 1835.

causal\_trans\_display\_units <- causal\_trans %>%  
 select(display\_desc, units)%>%  
 group\_by(display\_desc) %>%  
 summarise(Total\_Units = sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Units))  
causal\_trans\_display\_units

## # A tibble: 11 x 2  
## display\_desc Total\_Units  
## <chr> <dbl>  
## 1 Not on Display 281389  
## 2 Rear End Cap 68814  
## 3 In-Shelf 52084  
## 4 Front End Cap 28086  
## 5 Secondary Location Display 16422  
## 6 In-Aisle 16219  
## 7 Promo/Seasonal Aisle 12359  
## 8 Mid-Aisle End Cap 11084  
## 9 Store Rear 8349  
## 10 Store Front 5256  
## 11 Side-Aisle End Cap 1250

causal\_trans\_display\_units\_dollar <- causal\_trans %>%  
 select(display\_desc, units, dollar\_sales)%>%  
 group\_by(display\_desc) %>%  
 summarise(Sales\_Divided\_By\_Units = sum(dollar\_sales, na.rm = TRUE)/sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Sales\_Divided\_By\_Units))  
causal\_trans\_display\_units\_dollar

## # A tibble: 11 x 2  
## display\_desc Sales\_Divided\_By\_Units  
## <chr> <dbl>  
## 1 Side-Aisle End Cap 1.47  
## 2 Secondary Location Display 1.44  
## 3 In-Aisle 1.30  
## 4 In-Shelf 1.21  
## 5 Store Rear 1.20  
## 6 Mid-Aisle End Cap 1.20  
## 7 Store Front 1.17  
## 8 Rear End Cap 1.15  
## 9 Not on Display 1.15  
## 10 Front End Cap 1.14  
## 11 Promo/Seasonal Aisle 1.10

### Future Steps

### Bibliography

### Additonal Significant Code