**Dillard’s Point of Sale Data Association Rules Analysis**

Charles Marshall

Prof. Klabjan

IEMS 308

13 February 2020

**Introduction:**

The premise of this business problem is to help Dillard’s, a struggling retail department store, rearrange the floors of the store in a way which will place items likely to be bought together next to one another. This movement will promote sales by making it easier for customers to access items which they will be statistically likely to buy together. The analysis of this report will present potential improvements to Dillard’s current store format by suggesting items which should be grouped together and provide Dillard’s with data about what items are commonly bought together which might be useful when devising sales or inventory plans.

**Data Exploration:**

One of the most important parts of this project was to understand and explore the data. I sought to initially understand the data by viewing the relational database schema and understanding how certain tables were connected to one another. I found the two most important tables in this project were the transaction data and the SKU information. The transaction data gave raw point of sale data and was used to do the market basket association rules analysis. The SKU information was used to find out more about items included in the rules, including the style, color, size and brand. This was useful for drawing conclusions from the rules.

Additionally, the most challenging part of data exploration for this project was dealing with the large datasets. The transaction data was so large that it could not be loaded into Jupyter Notebook in its entirety, instead it was read in using only the first 7 columns including all the primary keys and the SType, the type of the transaction. The other columns, such as AMT and MIC, were left out because they are not relevant parameters to creating market baskets and doing association rules analysis.

Because the transaction dataset was so large, approximately 121 million rows, it needed to be broken up in a creative way. I decided to subset the data using specific SKU values. Instead of using random SKUs, I decided to use SKUs which corresponded to a specific department, more specifically the Polo Men’s Departments. I chose this department because I was familiar with the brand and thought I would be able to more effectively analyze the rules. Additionally, this department had a large number of unique SKU values (about 142 thousand unique values). A large number of unique SKUs in this department allows for a large number of choices. In the end, I ended up using the 100 most frequently purchased SKUs in order to effectively perform one hot encoding by not overload the apriori algorithm. This action also greatly reduced the size of the data set from over 5 million data points to 250 thousand data points. The 100 unique SKUs and their description are included in Appendices A.

Besides departments, I analyzed many other options to potentially use to subset the data. For instance, I decided against using specific store locations as a parameter because I wanted to create rules which could be applied to all locations, not just one store. Additionally, I considered subsetting the data by only considering a specific time period, specifically the summer of 2005, because I know retail stores often change their configurations when new products come in in different seasons. However, I found that using the summer still left the data set with 23 million data points, and it would therefore need to be broken up more. Both of these approaches are valid and could have been used, but in the end, I decided to go in a separate direction with the business question I answered.

**Solve the Problem:**

After data exploration, the first step in defining the problem was defining a market basket, which I defined as the variable orderID. The unique market basket is supplied by the composite primary key in the transaction table. In this table, no two rows can have the same SKU, store, register, transaction number, sequence number, and date combination. So, the orderID was created by adding the store, register, transaction number, sequence number, and date. From the composite primary key, I knew that if two unique SKUs had the same orderID then they had to be a part of the same market basket. The two SKUs could not have the same value because then it would break the rules of a primary key having a unique value for each row. The concept of the market basket could be scaled to any sized basket by grouping by the orderID and summing the rows after all the SKUs were one hot encoded. After dropping some no longer relevant columns, the data set is ready to be modeled with association rules.

The apriori function was used to create a set of frequent items based on their support. The important part of this function is to pick a minimum support value which is not too small so that it will blow up, but large enough so that it will not restrict too many rules. After apriori, association\_rules was run on the set of frequent items with a minimum lift of one, which indicates independent items, in order to generate rules.

**Analysis:**

The association\_rules function lead to the creation of nearly 440 rules. The description of these rules is as follows:

support confidence lift

count 438.000000 438.000000 438.000000

mean 0.000471 0.073210 9.391543

std 0.000512 0.078111 9.109040

min 0.000104 0.002935 1.015415

25% 0.000129 0.015443 2.189526

50% 0.000251 0.038598 5.583817

75% 0.000531 0.107968 14.065761

max 0.002423 0.344828 37.511159

I found the most important values in this table to be the mean lift, confidence and support. With this information, I chose to look at a subset of rules which had a lift greater than 15 and a confidence greater than 0.25. These values were chosen by using round values which were greater than the mean of both the lift and confidence. The 15 rules from this subset are included in full in Appendix B. The description of these rules is as follows:

support confidence lift

count 15.000000 15.000000 15.000000

mean 0.000326 0.293849 29.343860

std 0.000169 0.033044 6.350264

min 0.000104 0.253521 17.870856

25% 0.000135 0.257898 25.106399

50% 0.000427 0.295238 31.298819

75% 0.000474 0.321598 34.271343

max 0.000493 0.344828 37.511159

From looking at the rules, it is evident that some of the rules “repeat” in a way. For instance, rules 1, 2, and 3 in Appendix A are just different permutations of the same group of SKUs. Using this logic, the rules were broken down into different groups or market baskets:

Basket 1 SKUs: {1049441,1589441,439441}

A screenshot of a cell phone

Description automatically generated

Basket 2 SKUs: {1869441,459441,1069441}

A screenshot of a cell phone

Description automatically generated

Basket 3 SKUs: {2078810,1571243,5901243,4912330,9972087,2108810}A screenshot of a cell phone

Description automatically generated

Basket 4 SKUs: {5709431,3269431,6149446}A screenshot of a cell phone

Description automatically generated

Basket 5 SKUs: {3279431,5719431,6169446}A screenshot of a cell phone

Description automatically generated

Basket 6 SKUs: {4898351,5228351,5549233,5748351} **A screenshot of a cell phone

Description automatically generated**

From the descriptions of the SKUs, one thing that can easily be seen is that it is often hard to decipher exactly what each item is because there is no intuitive label. Because of this, we are often left to make an educated guess based on the sizes. For instance, I would guess that Basket 3 items are socks, ties, or some other item which is one size fits all. I would guess that basket 6 items are shirts because of the L label and basket 5 items are shorts because of the size 38.

Another thing that is evident is that the items that are grouped together and form rules together are items which are the same in every dimension except for color or style. This would suggest that people shop for similar items at the same time. For instance, someone would go to the Polo Men’s Department store looking for shorts and would be likely to buy multiple pairs of shorts in different colors or styles, but the same size. This would suggest that the department should attempt to keep similar clothing items (i.e. shorts style 1 and style 2) close to one another in the store. Additionally, the department should try to make sure similar sized items are near one another in different colors and styles so to increase the ease for the customer to look at different options.

**Appendix A: 100 SKUs used for Polo Men’s Department**

****

**Appendix B: Association Rules for Polo Men’s Department**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table of Rules for Polo Men's Department** | | | | | |
|  | **antecedents** | **consequents** | **support** | **confidence** | **lift** |
| **1** | ({'SKU\_1049441', 'SKU\_439441'}) | ({'SKU\_1589441'}) | 0.000474093 | 0.275482094 | 33.56855144 |
| **2** | ({'SKU\_1049441', 'SKU\_1589441'}) | ({'SKU\_439441'}) | 0.000474093 | 0.284900285 | 31.29881885 |
| **3** | ({'SKU\_439441', 'SKU\_1589441'}) | ({'SKU\_1049441'}) | 0.000474093 | 0.344827586 | 37.51115932 |
| **4** | ({'SKU\_1869441', 'SKU\_459441'}) | ({'SKU\_1069441'}) | 0.000493057 | 0.318042813 | 31.74843947 |
| **5** | ({'SKU\_459441', 'SKU\_1069441'}) | ({'SKU\_1869441'}) | 0.000493057 | 0.258064516 | 29.82646045 |
| **6** | ({'SKU\_1869441', 'SKU\_1069441'}) | ({'SKU\_459441'}) | 0.000493057 | 0.255528256 | 26.12618488 |
| **7** | ({'SKU\_2078810', 'SKU\_1571243'}) | ({'SKU\_5901243'}) | 0.0001043 | 0.328358209 | 35.40913531 |
| **8** | ({'SKU\_4912330', 'SKU\_5901243'}) | ({'SKU\_1571243'}) | 0.000251269 | 0.325153374 | 34.97413364 |
| **9** | ({'SKU\_4912330', 'SKU\_1571243'}) | ({'SKU\_5901243'}) | 0.000251269 | 0.335443038 | 36.17314139 |
| **10** | ({'SKU\_9972087', 'SKU\_1571243'}) | ({'SKU\_5901243'}) | 0.000146969 | 0.295238095 | 31.83756451 |
| **11** | ({'SKU\_4912330', 'SKU\_2108810'}) | ({'SKU\_9972087'}) | 0.000123264 | 0.254901961 | 29.62325933 |
| **12** | ({'SKU\_5709431', 'SKU\_3269431'}) | ({'SKU\_6149446'}) | 0.000426684 | 0.253521127 | 18.2570699 |
| **13** | ({'SKU\_3279431', 'SKU\_5719431'}) | ({'SKU\_6169446'}) | 0.000450389 | 0.305466238 | 24.08661237 |
| **14** | ({'SKU\_4898351', 'SKU\_5549233'}) | ({'SKU\_5228351'}) | 0.000109041 | 0.315068493 | 21.8465096 |
| **15** | ({'SKU\_5549233', 'SKU\_5748351'}) | ({'SKU\_5228351'}) | 0.000118523 | 0.257731959 | 17.87085612 |