Yusiqi Wang

Professor Diego Klabjan

Data Science and Analytics

Feb 13, 2020

# Executive Summary

**Purpose:**

The report is going to find the 100 SKUs that are best candidates to modify the planograms for retail store Dillard’s.

**Data Set:**

Data Set is selected as SKUs of the Department No.800 in Store No.8204.

**Conclusion:**

We put priority to finding larger lift, and also consider larger confidence and support to some extent. According to these filtering conditions, we can get the 100 SKUs. For the reason that their lift is larger, plus confidence and support are conditionally larger than others, we think movement of these SKUs will be helpful to increase sales.

The following table is the 100 SKUs candidates (most optimum 20 SKUs in red.)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1310252** | **2258366** | **3161221** | **4108011** | **5108107** | **6318344** | **726718** | **8942943** | **9277426** |
| **1832285** | **2698353** | **3524026** | **4628597** | **5978084** | **6063159** | **7808101** | **803921** | **9402188** |
| **106343** | **2072671** | **3978011** | **4208011** | **5098107** | **6328344** | **7064350** | **8401343** | **9667426** |
| **108507** | **2288366** | **3559555** | **4112626** | **5957568** | **6458364** | **7077566** | **8123420** | **9526376** |
| **1400555** | **2726578** | **3998011** | **4138348** | **539951** |  | **7407566** | **8391343** | **9073382** |
| **1467737** | **264715** | **3968011** | **4062567** | **5036322** |  | **7915** | **8401343** | **9168271** |
| **1588107** | **2708353** | **3864099** | **4072567** | **5079809** |  |  |  | **9257426** |
| **1658851** | **208362** | **3968011** | **4198011** | **5098107** |  |  |  | **9594893** |
| **17915** | **2571221** | **348498** | **4218011** | **5128107** |  |  |  | **9667426** |
|  | **2688353** | **3013129** | **4440924** | **5509179** |  |  |  | **9633** |
|  | **2716578** | **3448186** | **4498011** | **5758109** |  |  |  |  |
|  | **2726578** | **3589483** | **4686413** | **5778109** |  |  |  |  |
|  | **2784759** | **3631365** | **4928011** | **5923159** |  |  |  |  |
|  | **2938210** | **3751221** | **4938011** |  |  |  |  |  |
|  |  | **3844099** | **4976322** |  |  |  |  |  |
|  |  | **3854099** | **4992993** |  |  |  |  |  |
|  |  | **3864099** | **4999530** |  |  |  |  |  |
|  |  | **3868338** |  |  |  |  |  |  |
|  |  | **3894099** |  |  |  |  |  |  |
|  |  | **3898011** |  |  |  |  |  |  |
|  |  | **3908011** |  |  |  |  |  |  |

**Report for Dillard’s Planograms Modification**

# Business Context

# Dillard’s is a major retail chain with several stores. The retailer is interested in changing the planograms. It is highly unlikely that SKUs are already appropriately close to each other. The report is going to find the 100 SKUs that are best candidates to modify the planograms.

# Data Exploration

After reading data from 5 csv file, we matched Column name with Key name out of understanding of the Database relationship.

According to the data in TRNSACT table, use value\_counts() to find the “Store” number with the most frequent occurrence. We think that the sales of this store are the best, which is more representative for people’s purchase preference. We decide to use it as a subset of TRNSACT table, named dataTrnsactFinal.

In dataTrnsactFinal, we further judge what kind of product is the bestselling. Similarly, we use value\_counts() to find the SKU of the most purchased goods in **Store No.8042**. Thus, the **SKU No.4628597** obtained. We go to the DEPT table to find the Department of the goods. Then, we take the products of **Department No.800** as a set to judge which goods are related.

A screenshot of a cell phone

Description automatically generated

This step requires the method of isin. We take out the commodities belonging to Dept No.800 in dataTrnsactFinal separately and conclude a new dataframe DeptProduce.

Next, we extract the features in DeptProduce and put the extracted SKUs into onehot as a new dataframe. By using concat(), we can get a dataframe containing the data of TRNSACT table and SKU features.

Delete the data we don't need. Leave only the Trannum column.

Next, use groupby () to group in terms of Trannum column, and merge the same transaction number. We define a new dataframe named finalData. We get **in total 1085 different kinds of SKUs in the Dept No.800.**

Later, I found that the data in finalData should be 0 and 1, but in the process of merge, there is data greater than 1. For those data greater than 1, the item must have been purchased, then it could be replaced by 1, which is represented the meaning of “true.”

The following table present **what specific SKU appear in certain basket**. And the sum of number “1” present **the number of items per basket.**

A picture containing screenshot

Description automatically generated

Use the Apriori algorithm to get the support in each basket. Use association rule to get a dataframe named rules. At this time, we get the final dataframe we will use to calculate support, confidence and lift and filter potential SKUs.

The describe() function helps us get various summary statistics. We get Q3 (upper quartile) of the Confidence in the rules dataframe. For the reason that we expect all of support, confidence and lift to be large, we expect the **lift to be greater than Q3 (0.93)** for preliminary filtering. We put priority to Lift. Thus, we expect the **lift as close to Max value (2.59) as better.**

A screenshot of a cell phone

Description automatically generated

# Business Insight

In the rules table, there are relatively more SKU No.4108011, No.4628597 and No.352406 compare to others. From the perspective of retailers, when customers buy these three items, there is a greater probability for them to add other goods in their basket. Therefore, when modifying the planograms, these three SKUs should be given priority.

1. For example, from the perspective of customer shopping experience, put them in the center of other items, making it more convenient for customers to grab other items in different directions.
2. Or from the perspective of making more profit, put other items in the front position, and put these three items in the back position. In this way, when getting these three items, they will go through more shelves and probably buy other items that were not planned before.
3. At the same time, in order to improve efficiency, items related can be put together. When customers buy one item, if they immediately see the other item related to this one (say Complementary good,) they may also take it, which can also promote the sales volume of groceries. **Following Business Solution will consider this kind of strategy.**

A screenshot of a cell phone

Description automatically generated

# Business Solution (for 100 candidates)

According to lift greater than or equal to 2.51 and confidence greater than or equal to 0.93, we can get the relationship between several SKUs. Based on the data in following table, we can filter out the SKUs appear most times. Namely: 4108011, 3978011, 5108107, 3161221, 3524026, 4628597. But this is not enough to find enough data sets to modify the planograms.  
A screenshot of a cell phone

Description automatically generated

Next, try to modify the confidence (from 0.93 to 0.9) with lift unchanged. But according to data in the following table, we find that these are actually the new permutations and combinations for previous data, which cannot help to find new SKUs.A screenshot of a cell phone

Description automatically generated

It should be noted that the data with the confidence less than 0.9 is not needed. So next we only change the lift to further increase the range of filtering. Following table is part of the data of 2.5 > lift > 2.51 and confidence > 0.9. From this set of data, we can get new SKU (in red) and some new corresponding combinations. Add SKU\_2698353 to the candidates.

A screenshot of a cell phone

Description automatically generated

|  |  |
| --- | --- |
| **antecedents** | **consequents** |
| SKU\_3524026, SKU\_2698353 | SKU\_3978011, SKU\_5108107 |
| SKU\_3161221, SKU\_3978011 | SKU\_4108011, SKU\_2698353, SKU\_4628597 |
| SKU\_3978011, SKU\_4628597, SKU\_5108107 | SKU\_3524026, SKU\_2698353 |
| SKU\_3524026, SKU\_2698353, SKU\_4628597 | SKU\_3978011, SKU\_5108107 |
| SKU\_3524026, SKU\_4108011, SKU\_5108107 | SKU\_3161221, SKU\_3978011 |
| SKU\_3524026, SKU\_4108011, SKU\_4628597, SKU\_5108107 | SKU\_3161221, SKU\_3978011 |
| SKU\_3161221, SKU\_3978011, SKU\_4628597 | SKU\_3524026, SKU\_4108011, SKU\_5108107 |

Next, we will continue to narrow the scope of lift, and no new SKU will appear in the dataset between 2.42 and 2.5. We found a new SKU between the lift range of 2.41 and 2.42. Add SKU\_4208011 to the candidates.

A screenshot of a cell phone

Description automatically generated

|  |  |
| --- | --- |
| **antecedents** | **consequents** |
| SKU\_3524026, SKU\_4208011 | SKU\_4108011, SKU\_3978011 |
| SKU\_3524026, SKU\_4628597, SKU\_4208011 | SKU\_4108011, SKU\_3978011 |
| SKU\_4108011, SKU\_3978011, SKU\_4628597 | SKU\_3524026, SKU\_4208011 |

Next, when the lift is between 2.36 and 2.37, we find new SKU\_3559555 and SKU\_726718.

A screenshot of a cell phone

Description automatically generated

|  |  |
| --- | --- |
| **antecedents** | **consequents** |
| SKU\_3524026, SKU\_3559555 | SKU\_726718 |
| SKU\_3524026, SKU\_3559555, SKU\_4628597 | SKU\_726718 |
| SKU\_3161221, SKU\_3524026 | SKU\_3978011, SKU\_2698353 |
| SKU\_3161221, SKU\_4108011 | SKU\_3978011, SKU\_2698353 |
| SKU\_3524026, SKU\_3978011 | SKU\_2698353, SKU\_5108107 |
| SKU\_3161221, SKU\_3524026, SKU\_4628597 | SKU\_3978011, SKU\_2698353 |
| ..... | ..... |

In terms of efficiency, the 0.1 interval is a little slow. Let's try to enlarge the lift interval. Next is the data with lift values between 2.35 and 2.30. Add new SKU\_2258366 to the candidates.A screenshot of a cell phone

Description automatically generated

|  |  |
| --- | --- |
| **antecedents** | **consequents** |
| SKU\_3524026, SKU\_5108107 | SKU\_2258366 |
| SKU\_3524026, SKU\_726718 | SKU\_3559555, SKU\_4628597 |
| SKU\_3524026, SKU\_4628597, SKU\_5108107 | SKU\_2258366 |
| SKU\_3524026, SKU\_5108107 | SKU\_2258366, SKU\_4628597 |
| SKU\_3161221, SKU\_3524026 | SKU\_4108011, SKU\_2698353 |
| SKU\_3161221, SKU\_4108011 | SKU\_3524026, SKU\_2698353 |
| SKU\_3524026, SKU\_2698353 | SKU\_3161221, SKU\_4108011 |
| SKU\_3524026, SKU\_5108107 | SKU\_3161221, SKU\_2698353 |
| .... | .... |
| SKU\_3524026, SKU\_4108011 | SKU\_3978011, SKU\_4628597, SKU\_5108107 |

Repeat these steps until finding 100 candidates.

We put priority to finding larger lift, and also consider larger confidence and support to some extent. According to these filtering conditions, we can get the 100 SKUs. For the reason that their lift is larger, plus confidence and support are conditionally larger than others, we think movement of these SKUs will be helpful to increase sales.

The following table is the 100 SKUs candidates (most optimum 20 SKUs in red.)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1310252** | **2258366** | **3161221** | **4108011** | **5108107** | **6318344** | **726718** | **8942943** | **9277426** |
| **1832285** | **2698353** | **3524026** | **4628597** | **5978084** | **6063159** | **7808101** | **803921** | **9402188** |
| **106343** | **2072671** | **3978011** | **4208011** | **5098107** | **6328344** | **7064350** | **8401343** | **9667426** |
| **108507** | **2288366** | **3559555** | **4112626** | **5957568** | **6458364** | **7077566** | **8123420** | **9526376** |
| **1400555** | **2726578** | **3998011** | **4138348** | **539951** |  | **7407566** | **8391343** | **9073382** |
| **1467737** | **264715** | **3968011** | **4062567** | **5036322** |  | **7915** | **8401343** | **9168271** |
| **1588107** | **2708353** | **3864099** | **4072567** | **5079809** |  |  |  | **9257426** |
| **1658851** | **208362** | **3968011** | **4198011** | **5098107** |  |  |  | **9594893** |
| **17915** | **2571221** | **348498** | **4218011** | **5128107** |  |  |  | **9667426** |
|  | **2688353** | **3013129** | **4440924** | **5509179** |  |  |  | **9633** |
|  | **2716578** | **3448186** | **4498011** | **5758109** |  |  |  |  |
|  | **2726578** | **3589483** | **4686413** | **5778109** |  |  |  |  |
|  | **2784759** | **3631365** | **4928011** | **5923159** |  |  |  |  |
|  | **2938210** | **3751221** | **4938011** |  |  |  |  |  |
|  |  | **3844099** | **4976322** |  |  |  |  |  |
|  |  | **3854099** | **4992993** |  |  |  |  |  |
|  |  | **3864099** | **4999530** |  |  |  |  |  |
|  |  | **3868338** |  |  |  |  |  |  |
|  |  | **3894099** |  |  |  |  |  |  |
|  |  | **3898011** |  |  |  |  |  |  |
|  |  | **3908011** |  |  |  |  |  |  |

The following table shows most optimum 20 SKUs and their potential combinations.

|  |  |  |
| --- | --- | --- |
| **Antecedents** | **Consequents** | **Lift** |
| SKU\_3524026, SKU\_5108107 | SKU\_2258366, SKU\_4628597 | 2.32 |
| SKU\_3524026, SKU\_3559555 | SKU\_726718 | 2.36 |
| SKU\_3524026, SKU\_3559555, SKU\_4628597 | SKU\_726718 | 2.36 |
| SKU\_3978011, SKU\_5108107 | SKU\_3161221, SKU\_3524026, SKU\_4108011 | 2.59 |
| SKU\_3978011, SKU\_4628597, SKU\_5108107 | SKU\_3161221, SKU\_3524026, SKU\_4108011 | 2.59 |
| SKU\_3524026, SKU\_2698353 | SKU\_3978011, SKU\_5108107 | 2.505 |
| SKU\_3524026, SKU\_4208011 | SKU\_4108011, SKU\_3978011 | 2.41 |
| SKU\_3161221, SKU\_4108011, SKU\_4628597 | SKU\_3998011 | 2.22 |
| SKU\_6318344 | SKU\_726718 | 2.07 |
| SKU\_3524026 | SKU\_8942943 | 1.97 |
| SKU\_3559555 | SKU\_803921 | 2.14 |
| SKU\_2698353 | SKU\_9277426 | 1.96 |
| SKU\_2072671 | SKU\_3524026 | 1.86 |
| SKU\_3524026 | SKU\_2288366 | 1.90 |
| SKU\_2698353 | SKU\_7808101 | 2.00 |