Assignment 2

Zijie Huang

February 10, 2020

Introduction

Dillard is a major retail chain with several store. The retailer is interested in rearranging floors and change planograms. On the basis of Dillard's points-of-sales(POS) data, this report is going to make suggestions for 20 moves across the entire chain, and provide another 100 stock item candidates to modify the planograms.

1. Methodology

This report uses the points-of-sales (POS) data of Dillard from 08/01/2004 to 08/27/2005, which is available on website: *https://www.dropbox.com/s/5o0v3uoakkqe8u6/Dillards%20POS.7z?dl=0*

Since this dataset contains more than 130 million samples from 453 different stores, the following assumption is made in order to facilitate data processing: The consumption patterns of customers in different stores are similar, and each store has similar products and layout.

Therefore, all the discussion, figures and sample outputs are based on data of one particular store in Mabelvale, Arizona, 72103, whose ID in the dataset is 9806.

1.1 Dataset

There are five tables in the schema of POS data of Dillard—deptinfo, skstinfo, skuinfo, strinfo, transact. The description of these tables is stored in file: *DB Schemas retail.pdf*. The next session will explain some of the important fields among these tables.

In general, Dillard POS dataset contains information of 1564278 stock items, 60 departments, 453 stores and over 130 million transactions.

1.2 Description of Tables

1.2.1 Deptinfo

Table Deptinfo contains information of different departments. Its primary key is DEPT, which are numbers of departments. This table has only one attribute DEPTDESC, which is the description of departments.

1.2.2 Skstinfo

Table Skstinfo contains price information of the stock item. Its primary key is SKU and Store, which describes a specific stock item in a specific store. This table has 2 attributes:

- Cost: The cost of the stock item.
- Retail: The retail price of the stock item.

The profit of the stock item can be calculated by subtracting cost from retail.

1.2.3 Skuinfo

Table Skuinfo contains attributes of the stock item. Its primary key is SKU, which is the stock number of the stock item. This table has 9 attributes, but only one attribute is used in analysis:

- Dept: Foreign key of table Deptinfo, which describe the department of the stock item belongs.

1.2.4 Strinfo

Table Strinfo contains information of stores. Its primary key is Store, which is the number of the store. This table has 3 attributes:

- City: The city of the store.
- State: The state of the store.
- Zipcode: The zip code of the store.

1.2.5 Transact

Table transact contains information of each transactions. Its primary key is composed of the six attributes: SKU, Store, Register, Trannum, Saledate, SEQ. This table has another 6 attributes, but only attribute Quantity and Stype are used in analysis:

- SKU: Stock keeping unit number of the stock item.
- Store: Store number
- Register: Register number of the current transaction.
- Trannum: Transaction code.
- Saledate: Sale date of the stock item.
- SEQ: Sequence number.
- Quantity: Item quantity of the stock item.
- Stype: Type of the transaction (Return or Purchase)

It is noticeable that one transaction has only one transaction number, but different transactions on different dates may have the same transaction number, and different transactions on the same date have different transaction numbers.

1.3 Data Preprocessing

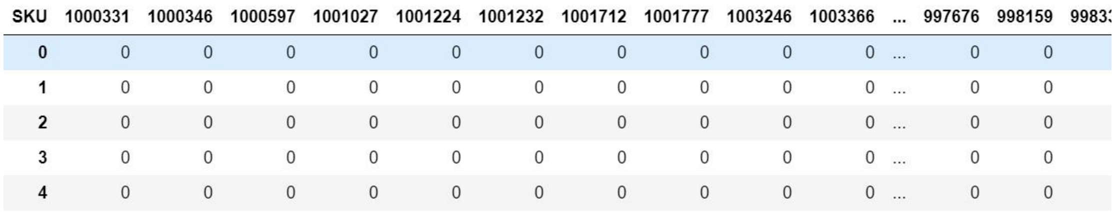
Since the dataset contains 130 million records, file *trnsact.csv* is read in chunk. Each time 10 million records are read, then preprocessed and finally concatenated.
In each chunk, data of table Transact is preprocessed according to following steps:

(1) All records whose store number is not equal to 9804 are filtered.
(2) All records contain missing values are filtered since the number of missing value records is negligible compared to the total number of records.
(3) All records whose type is “returned” are filtered.

(4) Each transaction is identified by its sale date and transaction number. Therefore, a transaction can be regarded as a basket, the number of unit number(SKU) in each transaction can be regarded as the number of items in a basket, and the quantity of each SKU is grouped by the sale date, transaction number and unit number.

(5) Hot encoding method is applied to fit the data for Apriori algorithm. Any value larger or equals to 1.0 is recorded as 1, while any value smaller than 1.0 is recorded as 0.

The following figure is an example of preprocessed data:



SKU	1000331	1000346	1000597	1001027	1001224	1001232	1001712	1001777	1003246	1003366	...	997676	998159	9983
0	0	0	0	0	0	0	0	0	0	0	...	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	

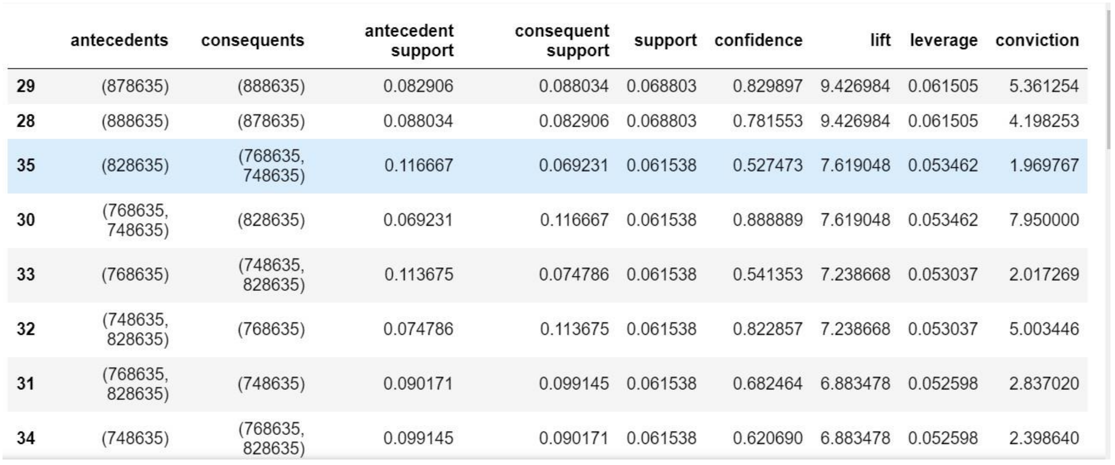
Figure 1.1 Sample of preprocessed data, columns are unit number (SKU), rows are transactions

1.4 Algorithm and Validation

Apriori algorithm is applied in processing data. The support, confidence and lift values of each rule are applied to evaluate the result of Apriori algorithm.

2. Result

The result of Apriori algorithm is saved as file: *Rules9806.csv*. There are 36 rules produced by Apriori, and the antecedents and consequents are displayed as unit number of items since the name of items are missed in this dataset. This file is the suggestion for 20 moves across the entire chain. The following figure is an example of this file:



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
29	(878635)	(888635)	0.082906	0.088034	0.068803	0.829897	9.426984	0.061505	5.361254
28	(888635)	(878635)	0.088034	0.082906	0.068803	0.781553	9.426984	0.061505	4.198253
35	(828635)	(768635, 748635)	0.116667	0.069231	0.061538	0.527473	7.619048	0.053462	1.969767
30	(768635, 748635)	(828635)	0.069231	0.116667	0.061538	0.888889	7.619048	0.053462	7.950000
33	(768635)	(748635, 828635)	0.113675	0.074786	0.061538	0.541353	7.238668	0.053037	2.017269
32	(748635, 828635)	(768635)	0.074786	0.113675	0.061538	0.822857	7.238668	0.053037	5.003446
31	(768635, 828635)	(748635)	0.090171	0.099145	0.061538	0.682464	6.883478	0.052598	2.837020
34	(748635)	(768635, 828635)	0.099145	0.090171	0.061538	0.620690	6.883478	0.052598	2.398640

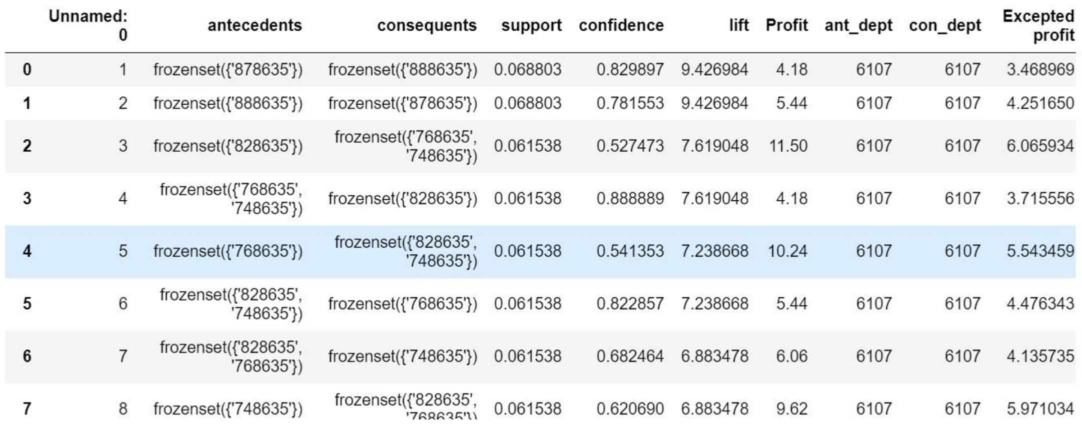
Fig 2.1 A sample of association rules

To further evaluate the benefits of these rules, profits of consequential items are added in the result as column ‘Profit’. The value of this column is calculated by using corresponding ‘Retail’ minus ‘Cost’ in Skstinfo table. It is noteworthy that the primary key of Skstinfo is (SKU, Store), which means one same item can have different values in different stores. As far as store 9806, the retail and cost values of some stock items are missing in Skstinfo table. The following steps are taken to process the missing values:

(1) If SKU number is not included in Skstinfo table, the profit will be labeled as NaN.
(2) If SKU number is included but Store number 9806 is not included, the profit will be calculated by the average of profit of other stores.

In addition, department number of antecedent items and consequential items are added as column 'ant_dept' and 'con_dept', since the schema did not specify the name of stock items. The description can be easily gained by searching in Deptinfo table.

The modified results are saved as file *final_rules.csv*, and the following is an example of this file:



	Unnamed: 0	antecedents	consequents	support	confidence	lift	Profit	ant_dept	con_dept	Excepted profit
0	1	frozenset({'878635'})	frozenset({'888635'})	0.068803	0.829897	9.426984	4.18	6107	6107	3.468969
1	2	frozenset({'888635'})	frozenset({'878635'})	0.068803	0.781553	9.426984	5.44	6107	6107	4.251650
2	3	frozenset({'828635'})	frozenset({'768635', '748635'})	0.061538	0.527473	7.619048	11.50	6107	6107	6.065934
3	4	frozenset({'768635', '748635'})	frozenset({'828635'})	0.061538	0.888889	7.619048	4.18	6107	6107	3.715556
4	5	frozenset({'768635'})	frozenset({'828635', '748635'})	0.061538	0.541353	7.238668	10.24	6107	6107	5.543459
5	6	frozenset({'828635', '748635'})	frozenset({'768635'})	0.061538	0.822857	7.238668	5.44	6107	6107	4.476343
6	7	frozenset({'828635', '768635'})	frozenset({'748635'})	0.061538	0.682464	6.883478	6.06	6107	6107	4.135735
7	8	frozenset({'748635'})	frozenset({'828635', '768635'})	0.061538	0.620690	6.883478	9.62	6107	6107	5.971034

Figure 2.2 Example of modified rules

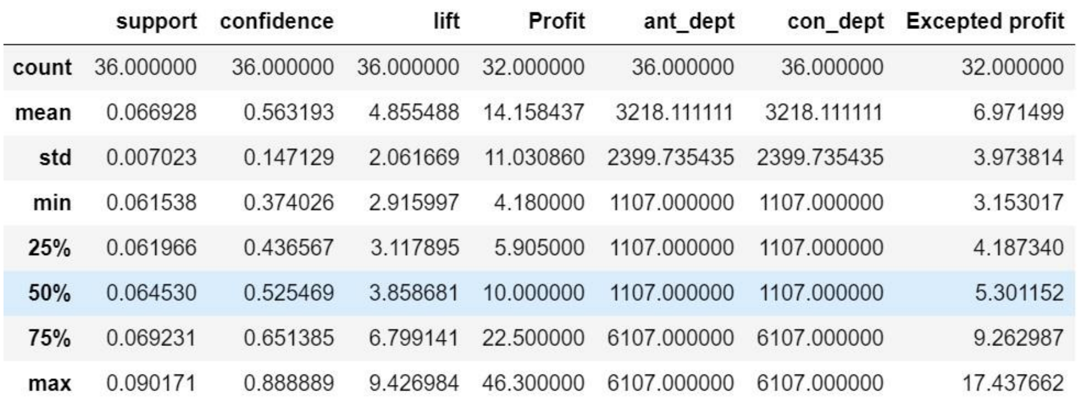
Moreover, rules that related to 100 stock items are saved as file: *100SKU.csv.* These are the 100 SKUs that are best candidates to modify the planograms.

3. Discussion

Since Dillard is interested in rearranging the floors of the stores to gain more profit, the value of profit generated by one association rule is important. The expected profit of one association rule can be derived by using following method:

$$Expected\ profit = Confidence * Profit$$

Considering the huge numbers of transactions, it is acceptable for a rule to have relatively low support value but have a high lift and expected profit value. Figure 3.1 displays the numerical description of the association rules:



	support	confidence	lift	Profit	ant_dept	con_dept	Excepted profit
count	36.000000	36.000000	36.000000	32.000000	36.000000	36.000000	32.000000
mean	0.066928	0.563193	4.855488	14.158437	3218.111111	3218.111111	6.971499
std	0.007023	0.147129	2.061669	11.030860	2399.735435	2399.735435	3.973814
min	0.061538	0.374026	2.915997	4.180000	1107.000000	1107.000000	3.153017
25%	0.061966	0.436567	3.117895	5.905000	1107.000000	1107.000000	4.187340
50%	0.064530	0.525469	3.858681	10.000000	1107.000000	1107.000000	5.301152
75%	0.069231	0.651385	6.799141	22.500000	6107.000000	6107.000000	9.262987
max	0.090171	0.888889	9.426984	46.300000	6107.000000	6107.000000	17.437662

Figure 3.1 Numerical description of association rules

Rules either in top 75% of the lift or in top 75% of expected profit is chosen as the recommendation for Dillard. And the following are the items' SKU involved in these rules:
{888653, 828635, 133108, 878635, 768635, 570033, 1976522, 243947, 1117863, 1117865, 748633, 277805, 617865,977309, 1440299, 615863}. More detailed change in planogram will be covered in next section.

4. Conclusion

Based on the results and discussions above, following conclusions can be drawn:

1. Item 888365 ,828635, 133101 should be put relatively close since there exists a recommended association rule between any two of them.
2. Item 878635, 768635, 748635 should be put relatively close since there exists a recommended association rule between any two of them.
3. Item 277865, 617865 should be put around 1117865.
4. Item 570033, 1976522 should be put around item 1440299.
5. Item 243947, 1117865 and 748635 should be put together since there exists a recommended association rule between any two of them.
6. Item 977309 should be put around 570033.
7. Item 878635 should be put around item 768635.
8. Item 615863 and 570033 should be put around item 748633.
9. Department NOB, ONEIDA and JACQUES should be arranged relatively close.

Appendix: Problems remain to be solved

1. The data is read in chunk, but during each iteration, I noticed a slight difference in the number of records I got. Since the difference is relatively small compared with the size of the whole data, this problem has no significant impact on the result of association rules. I guess it may be caused due to the algorithm of chunk-reading, or maybe my laptop is not powerful enough to run the chunk-reading stably.
2. I randomly selected 300000 records from the dataset as testing data, and these records' store numbers are not 9806. It seems like this random dataset has no similar pattern with store 9806 since the support of the derived rules is relatively low in this random dataset.