Dillard's POS Analysis

Bruno Zecchi

For this assignment we were tasked with analyzing Dillard's POS data to find the potential 100 SKUs whose movement in the planogram would cause the most increase in benefit for the retail chain. The task of the problem is complicated by the fact that the data is very large (over 11 gigabytes of data), which required the use of subset selection throughout the analysis process. A detailed description of the methodology used can be found below. After the data was cut to a reasonable size, association rules could now be formed. We could now analyze the antecedents with the highest lift values and selected the top one hundred SKUs to be the candidates for Dillard's rearrangement plan. By finding the SKUs antecedents with the highest lift, we identified products that tend to lead customers to purchase other products as well. We believe moving these items into highly viewed areas of stores is beneficial for stores. These SKUs would replace previous items that did not provide such an incentive to customers to purchase other items. This effect is further amplified by the fact that all of our 100 potential SKUs are highly popular items by themselves, meaning there is a less of a worry that customers might not want to purchase the initial item itself (and thus not purchase the consequents).

Analysis Process

Data Loading:

Since the data files to analyze are so large, we utilized Python's Dask package, which can store very large datasets into a computer's memory. We looked at Dillard's transaction data.

Subset Selection:

In order to narrow down the number of columns in the data and permit faster analysis, we decided to subset the transaction data so that data only from Illinois stores is considered. We believe this will also permit more benefits to individual stores because this analysis can be run for any state. Performing analysis for each individual state separately will provide more tailor-made results to its stores, thus increasing the probability that each state's strategy will work better on its customers. The data from Illinois spans from August 2004 to August 2005.

Storing only relevant columns:

After subset selection, irrelevant columns were removed from the dataframe. We are trying to identify variables that define individual baskets and we determined that the Store Number, Register Number, Transaction Code, and Sale Data values are what determine a basket. All other columns (apart from SKU) are irrelevant for further analysis and are removed.

Creating Basket IDs:

With the four relevant variables identified above, basket IDs could be formed for each SKU. This will assign each transaction row to the basket it belongs to.

Data Filtering:

There is still too much data in the dataframe. We first remove baskets that only have 1 item in them because they are useless for the association algorithm. After that we further cut down the dataframe by only keeping the 1000 most popular SKUs. Apart from reducing the data, this also ensures that our SKU recommendations are not unpopular items. Since we can now potentially have baskets in the dataframe that again only have one SKU in them, we again remove those one-item baskets. We are now left with 48,000 transactions rows in our data with only 1000 SKUs, which can now be easily processed by the association algorithms.

Final Steps:

We now run through the apriori and association_rules algorithms in Python, which prints out all antecedents and their lift values. We store the 100 individual SKUs antecedents that have the highest lift values and present those as our final recommendation.

SKU Recommendations

The final 100 recommended SKUs number are displayed below:

'8142644'	'2726578'	'6072521'	'8966664'	'6656135'	'3898011'	'3898011'	'8142644'	'6696135'	'6656135'
'8122644'	'2716578'	'6062521'	'8646664'	'6642521'	'3968011'	'3968011'	'8132644'	'2141939'	'8146822'
'8132644'	'3988011'	'5846627'	'8618636'	'6742521'	'803921'	'4440924'	'8122644'	'6318344'	'7668092'
'7568362'	'3908011'	'5826627'	'8798636'	'6752521'	'3690654'	'3898011'	'8142644'	'8156822'	'3998011'
'7848362'	'3898011'	'5786627'	'8618636'	'6320353'	'3968011'	'3968011'	'8122644'	'1671939'	'9590684'
'7468362'	'3968011'	'4898362'	'8718362'	'6340353'	'9576432'	'3690654'	'8132644'	'3978011'	'6032521'
'7596135'	'2716578'	'5168362'	'8308357'	'6300353'	'9716432'	'3978011'	'7739879'	'8146822'	'8166822'
'7636135'	'2001637'	'4648362'	'8288357'	'6208362'	'9469364'	'3898011'	'9452485'	'7596135'	'4648362'
'6656135'	'5487088'	'4898362'	'8156822'	'6738362'	'9600684'	'3524026'	'7693786'	'803921'	'1751939'
'6706135'	'1821637'	'4318362'	'8166822'	'6998362'	'9459364'	'3908011'	'7793786'	'4108011'	'7596135'