

**TDS3301 – DATA MINING**

**Lecturer: Dr. Ho Chiung Ching**

**Tutorial Section: TT01**

**Assignment #2: Association Rule Mining**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| 1132701439 | Tay Tiong Soon |
| 1132700475 | Teo Yong Zhi |
| 1131122661 | Tazeek Bin Abdur Rakib |
| 1132701076 | Khairul Azmi bin Aminuddin |

**Introduction**

The domain to be investigated involves a bakery chain, which has a menu of a total of 50 items: 40 different types of pastries and 10 types of drinks. By utilizing association rule mining, the bakery chain can gain the possible benefits, based on their transaction history:

* Increase the stock of the items that have a high selling rate.
* Discover unexpected patterns of items sold together and take action, such as ‘Buy 1 Get 1 Free’ or a promotional package.

However, the bakery chain must wary of some of the drawbacks when using association rule mining:

* Redundant rules appearing
* Rules that have a very low **lift ratio**, hence, less usefulness.
* Parameterizing the **support** and **confidence** threshold can give a lot of unnecessary rules, or in the worst case: no rules at all.

**Dataset**

Two datasets will be used:

* **goods.csv:** This dataset contains the information of all the items in the bakery chain’s menu. It consists of five columns:
  + **item\_id:** the id of the item in the dataset.
  + **flavor:** the flavor of the item, such as Apple and Chocolate
  + **food:** Examples include cookie and tart.
  + **price:** the price of the item
  + **type:** the type of item. It can be either Food or Drink.
* **75000\_items.csv:** This dataset contains the past 75,000 transactions made by the bakery chain. It consists of three columns:
  + **transaction\_id:** the id of the particular transaction
  + **item\_count:** the quantity of a specific item per transaction.
  + **item\_id:** the id of the item that was sold. Only present once per transaction.

Before association rule mining can take place, the following preprocessing were done to make it meaningful and appropriate:

1. **Merging:** Firstly, the flavor column and food column, from the **goods.csv,** were combined as one column to create the full name of the item. After that, the two dataframes (created from the two csv files) were merged based on the specific item’s id.

1. **Histogram Analysis:** The merged dataframe is then aggregated to find the total counts of items sold throughout the dataset. A picture shown below (Created using ggplot2) and the items are sorted, from the most-sold to the least-sold.

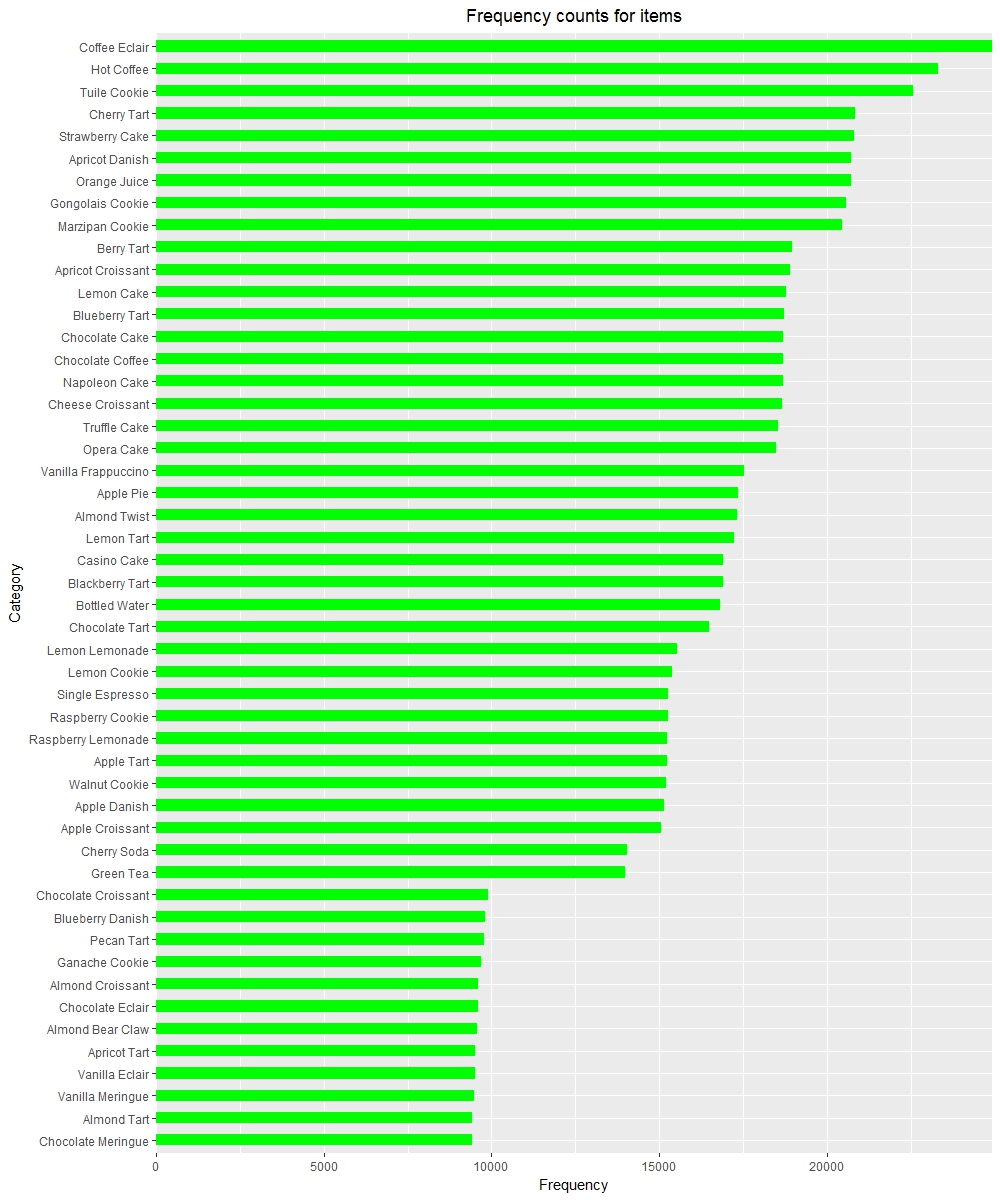


Figure 1: Frequency Distribution of Items Sold

1. **Transaction Object:** Lastly, the merged dataframe is transformed into a transaction object that contains:
   * **transaction\_id:** the id of the transactions. This increases sequentially and is unique.
   * **items:** the list of items sold in the transaction. Do note that the counts are not taken into consideration. An example of the transaction object, showing the first 10 rows is shown below:

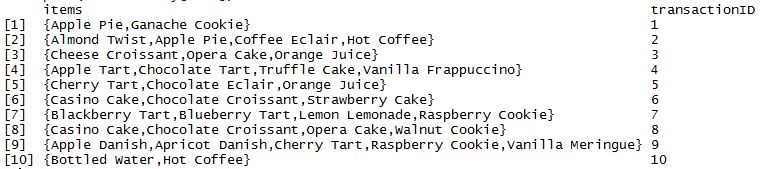


Figure 2: First ten transactions of the dataset

**Association Rule Mining**

**Apriori** algorithm was chosen for association rule mining, with the following main parameter settings:

* **Support:** This value denotes the set of items that occur frequently. Since our dataset consists of a high number of transactions, a support value of **0.005** would be appropriate.
* **Confidence:** This value denotes the frequency of subsets of supported itemsets. Again, due to the size of our dataset, using a confidence value of **0.65** would be ideal.

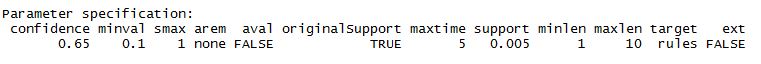
****

Figure 3: Parameter settings for Apriori Algorithm

The algorithm generated **131 rules** in about **80ms**. However, most of the rules are redundant, as shown below.

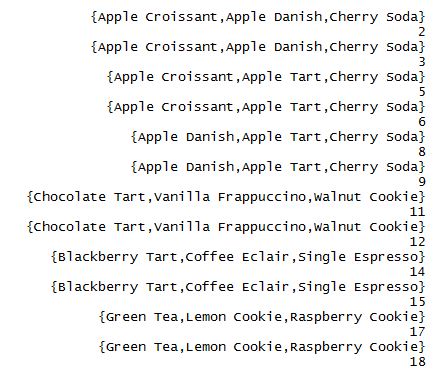


Figure 4: Redundant rules

Hence, **we** need to get rid of the redundant rules. After removal, the total number of rules drops to **31 rules**, which is about a decrease of 75% from the original. A scatterplot of the newly pruned rules are shown below

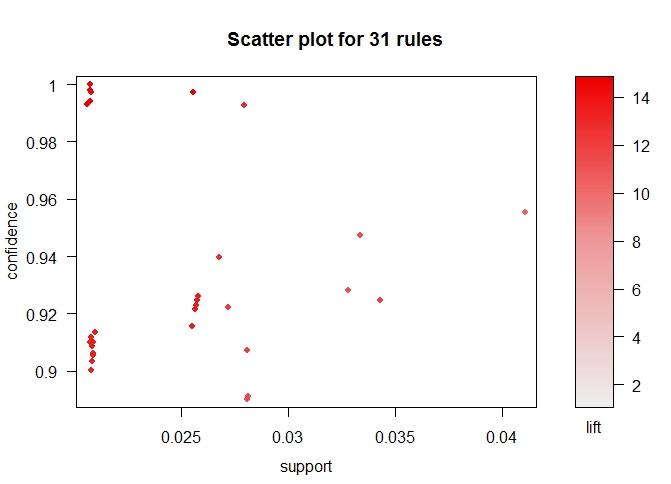


Figure 5: Scatterplot of unique rules

From the scatterplot, we can see that 26 of our rules have a support of less than 0.03 but have a high enough confidence, with a high lift ratio complimenting it. This justifies our parameter choices and removal of redundant rules.

**Results and Recommendations**

Before visualizing our results, we sort our rules based on the **lift ratio.** This is because the strength of the association (rule) varies with the lift ratio: the larger the lift ratio, the stronger the rule. After sorting by the lift ratio, we generate the following force-directed graph:

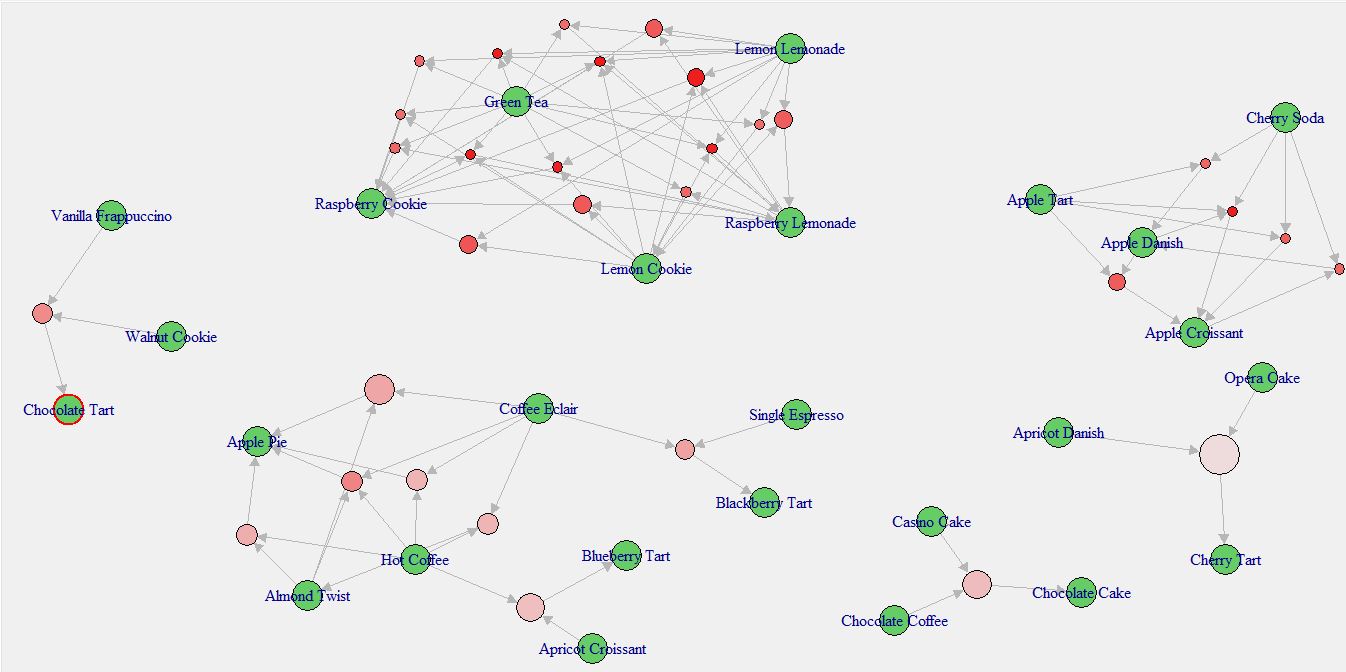


Figure 6: Force-Directed graph of rules. Note that **size** corresponds to **support** and **color** corresponds to **lift ratio**

Based on the force-directed graph, we can observe, for example, that a lot of customers take the initiative to buy **Lemon Lemonade and Green Tea,** since it consists of entirely out-going arrows and a high lift ratio. The recommended choice would be to put **Lemon Lemonade** and **Green Tea** on the same shelf, along with the **Raspberry Lemonade.** Also, a promotional package involving **Lemon Lemonade and/or Green Tea** together with **Lemon Cookie or Raspberry Cookie** would also be ideal, as there are a lot of incoming arrows. Another ideal choice would be to promote, as well as increase the stock, **Coffee Eclair** since it also has a lot of outgoing arrows and sold in large quantities, as observed from Figure 1. In terms of pair, it has a strong support with **Hot Coffee** and **Almond Twist**.

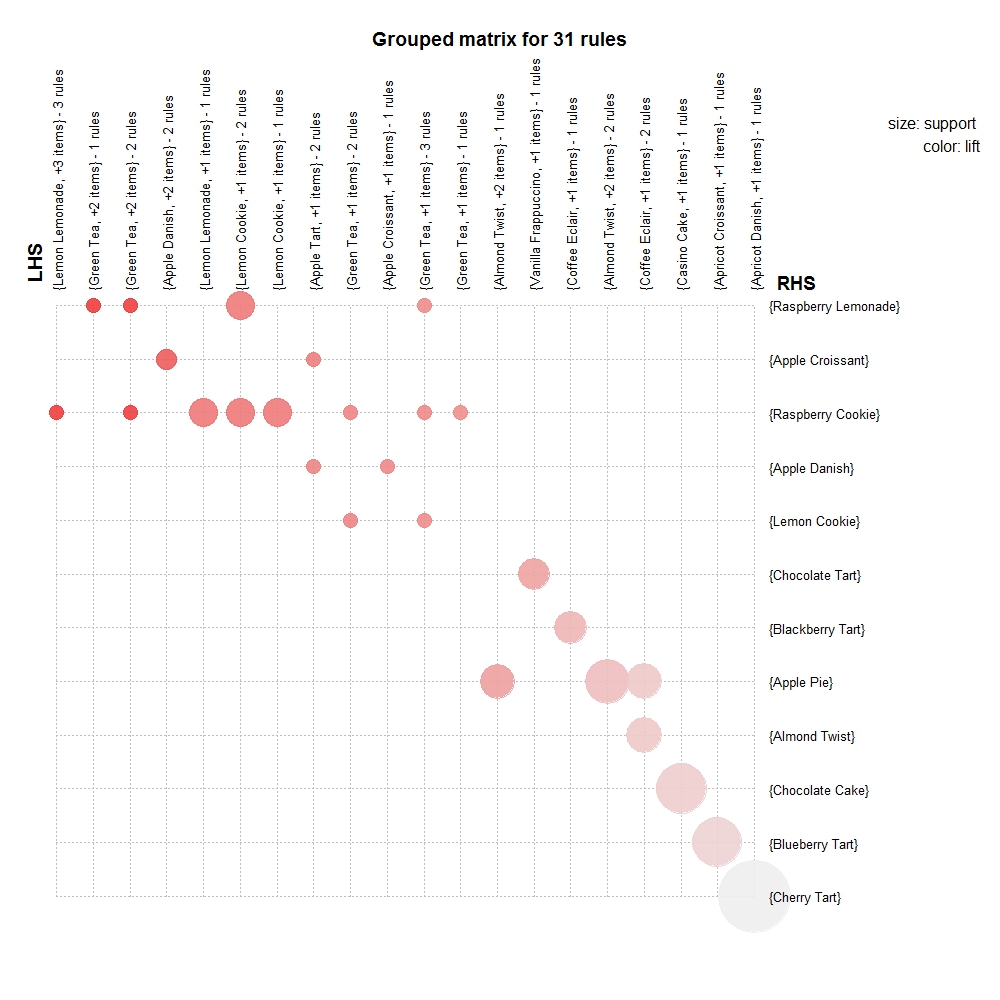


Figure 7: Group Matrix for rules. Note that **size** corresponds to **support** and **color** corresponds to **lift.**

From the figure above, **Raspberry Cookie** seems to be the ideal choice based on both support and lift ratio. However, two items that should be taken into consideration would be **Apple Croissant** and **Apple Danish,** since they both also have good support and confidence. Taking information from Figure 5 and Figure 6, a promotional package of **Cherry Soda** along with an **Apple-**flavored item would be an ideal choice, while another option would be to stack all **Apple-**flavored items on the same shelf.