**Task 1**

**1 - Pre-processing**

The D1 dataset was loaded with pandas and inspected with the info() method. Column names were converted to lower case for convenience. Pre-processing. was limited to conversion of the Date variable to a pandas datetime object, and the Sales\_ID, and Customer\_ID to string types

A screenshot of a computer code

Description automatically generated

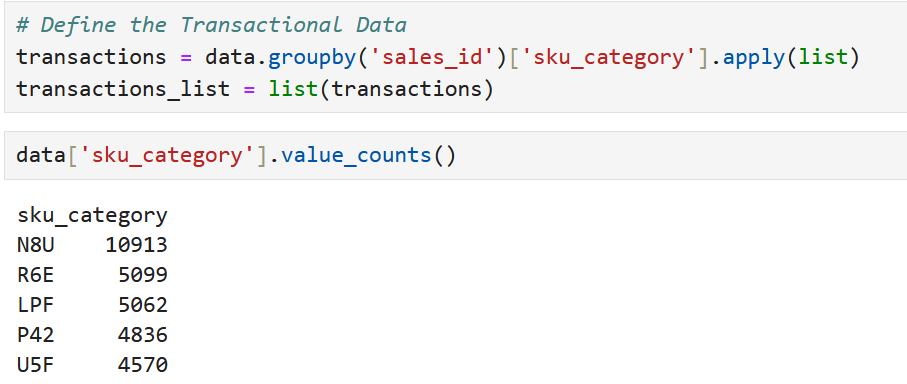
The processed dataset contains the processed variables and no missing data.

A screenshot of a computer program

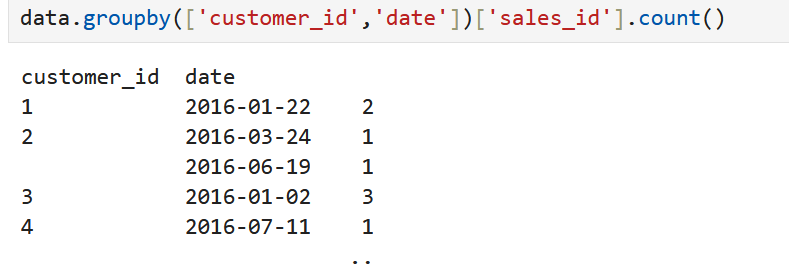
Description automatically generated

**2 - Association mining**

Association mining was carried out to determine which categories of products are bought together in a transaction rather than individually. First, the dataset was transformed to a transactional dataset using the sales\_id and sku\_category variables.



sales\_id was used because some customers had multiple sales on different dates. For example, customer 2 had two purchases, one in March and one in June 2016.



The transactions table containing all sku\_categories purchased for each sales\_id was populated and this was used to generate association rules. The apyori 's apriori function was used to generate the result set with an initial min\_support value of 0.005. This was based on the calculation of the inverse of the number of sku\_categories.

A close-up of a computer screen

Description automatically generated

The min\_confidence was initially set as 0.5 as a pragmatic approach to filter out weak associations without discarding potentially important or useful rules. The results data were converted to pandas dataframe and inspected. The initial support and confidence parameters returned a total of 12 rules. Setting the min\_confidence to a higher level was deemed unnecessary as it would have reduced the number of rules returned. We finally settled on a min\_confidence level of 0.4 as it returned a manageable set of 30 rules, and included a rule with a left value of 9.92 that was not present in the initial set because the confidence level was 0.41.

The code to generate the rules, convert to a pandas dataframe, and the resulting top 5 rules ordered by descending lift values are as follows:

A screenshot of a computer

Description automatically generated

The left-hand side and the right\_side of the association rules represents the antecedent, or the "if", and the consequent, or the "then" parts of the rules respectively.

The support is the proportion of transactions that contain both the left\_side and the right\_side. Or more simply, how frequently the rules occur in the dataset. A higher level of support would indicate that the rule is more common.

The confidence is the proportion of transactions that contain the left\_side where the right\_side also occurs. It tells us how often the rule has been found to be true. A high level of confidence reflects a higher level if reliability of the rule. Higher confidence means that the rule holds true more frequently when the antecedent occurs.

The lift shows whether the occurrence of the left\_side increases the likelihood of the right\_side. Where left is greater than 1 the left\_side and right\_side are positively correlated and that the scale to which the increase in the right\_side would be in the presence of the left\_side rule.

For our dataset, the top 5 rules show that moderately low levels of support, in the range 0.06% for {N8U,OXH} → {IEV} to 1% for {IEV,OXH} → {LPF}.

The min\_confidence was set at 0.4 in our apriori algorithm therefore the support will be at least 40%. The range in our top 5 results is 41% for {N8U,LPF} → {OXH} to 63% for {N8U,OXH} → {LPF}. These two examples are noteworthy as they contain the same items in different orders. In rule 28, OXH is likely to be bought when N8U and LPF are bought together, but it’s not as certain (41% confidence). In rule 29, LPF is much more likely to be bought when N8U and OXH are purchased together (63% confidence), making LPF a more consistently associated item in this context.

All the lift values in for our top 5 rules are 8.6 and above. This means that the presence of the right\_rule is at least 8.6 more likely in the presence of the left\_rule, rather than if the item in the right\_rule was independent. For example, {N8U,LPF} → {OXH} shows that the occurrence of N8U and LPF together increases the likelihood of an additional purchase of OXH is nearly 10 times greater.

**3 – Five common categories purchased with 01F**

To report the five most common product categories that customers bought with the product category ‘01F’ we modified our existing code filter the results such that the left\_side or the right\_side contain the sku\_category 01F. We lowered the min\_confidence parameter to 0.25 to include more rulesets. The values were ordered by support for this set to discover the rulesets with the highest proportions of rulesets where 01F occurs.

A screenshot of a computer

Description automatically generated

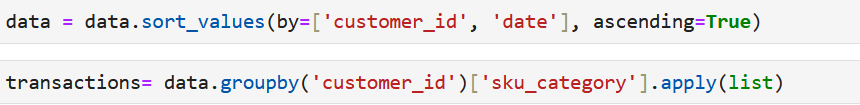
The first two rule sets are the same, but in different orders. The top 5 most common product categories purchased alongside the product category 01F are therefore IEV, LPF, OXH, FU5, and N8U.

**4 - Sequence analysis**

Sequence analysis can be carried out on this data set. The main feature of sequence analysis is that the algorithm aims to identify patterns in the order of transactions over time. In our case, the dataset was used to understand the sequence in which analyse the purchase different SKU categories over time using the date variable allowing us to identify the most frequent sequences of SKU categories that customers typically bought in a particular order.

Steps

1. The data were prepared for sequence analysis by ordering the dataset by customer\_id, then date, both in ascending order to group all transactions of each customer together, ensuring that the sequence analysis was conducted on a per-customer basis.
2. We ran the Philippe Fournier Viger SPMF java library provided in the week 8 tutorial.
3. Use the code provided in the week 8 tutorial to process the java library text outputs as association rules for the sequences into a pandas dataframe
4. The parameter of the association rules were modified for this process. Given the large number of possible rules the min\_support was set to 0.01 and min\_confidence was set to 0.5 to limit the result to a meaningful and manageable set.



A screenshot of a computer

Description automatically generated

The results of the sequence analysis allowed us to determine the likelihood that subsequent product categories would be purchased following purchase of initial product categories. For example, in our results the we showed that if a customer purchased [NTA] then the likelihood of a subsequent purchase of [IEV] is 62%, however this rule is based on 1.4% of all customer sequences meaning it is based on a small proportion of the dataset.

**5 - Relevance to decision makers**

There are several scenarios where these analyses could be actionable or insightful for decision makers. For example, the associating mining algorithm provided information about which product categories are purchased together in single transactions. Decision makers could use this information to drive sales campaigns such as product bundling or offering discounts on grouped sales. Similarly, this information could be used to inform store layouts where categories frequently purchased together are co-located in stores.

Similarly, sequence analysis could target individual customers with personalised product recommendations or targeted promotions based on potential for subsequent purchases. Another potential use case would be to forecast demand for future products based on volume of products sold that have a higher likelihood of follow-up purchases.

**Task 2 – Clustering**

**1 – Pre-processing**