**Association Project**

**Problem Statement**

To find which products classified as Raw Material are bought more frequently with the products classified as Finished Goods. For example, we need to know how well accessories like ‘PL-02’ (Raw Material) go well with cylinder ‘CP96SDB32-50C’ (Finished Good) or how well accessories like ‘BJ3-1’ (Raw Material) go well with combination of ‘CD85N25-100-B’ (Finished Good) and ‘BM2-025’ (Raw Material).

**What is Association Analysis?**

Association Analysis attempts to find common patterns of items in large data sets. One specific application is often called Market Basket Analysis.

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items. It works by looking for combinations of items that occur together frequently in transactions. To put it another way, it allows retailers to identify relationships between the items that people buy.

Association Rules are widely used to analyze retail basket or transaction data, and are intended to identify strong rules discovered in transaction data using measures of interestingness, based on the concept of strong rules. It is used to predict the likelihood of products being purchased together. Association rules count the frequency of items that occur together, seeking to find associations that occur far more often than expected.

Various Algorithms that use association rules include AIS, SETM and Apriori. The Apriori algorithm is commonly cited by data scientists in research articles about market basket analysis and is used to identify frequent items in the database, then evaluate their frequency as the datasets are expanded to larger sizes.

**An example of Association Rules**

* Assume there are 100 customers
* 10 of them bought Milk, 8 bought Butter and 6 bought both of them.
* bought Milk → bought Butter
* Support = P (Milk & Butter) = 6 / 100 = 0.06
* Confidence = Support / P (Butter) = 0.06 / 0.08 = 0.75

**Why do we need Market Basket Analysis?**

Market Basket Analysis can increase sales and customer satisfaction. Using data to determine that products are often purchased together, retailers can optimize product placement, offer special deals and create new product bundles to encourage further sales of these combinations.

These improvements can generate additional sales for the retailer, while making the shopping experience more productive and valuable for customers. By using market basket analysis, customers may feel a stronger sentiment or brand loyalty toward the company.

**Definition of some commonly occurring Jargons**

**Items**

An individual article or unit, especially one that is part of a transaction.

**Itemsets**

A set of items, collection of items or combination of different items.

**Frequent Itemsets**

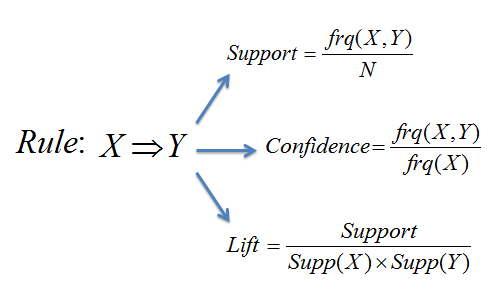
Itemsets which are frequently bought by many customers. For instance, customers of an on-line bookstore could be considered examples, each represented by the set of books he or she has purchased. A set of books, such as {“Machine Learning,” “The Elements of Statistical Learning,” “Pattern Classification,”} is a frequent itemset if it has been bought by sufficiently many customers.

**Association Rules**

Association Rule has the form X → Y, where X and Y are itemsets, and the interpretation is that if set X occurs in an example, then set Y is also likely to occur in the example.

**Support**

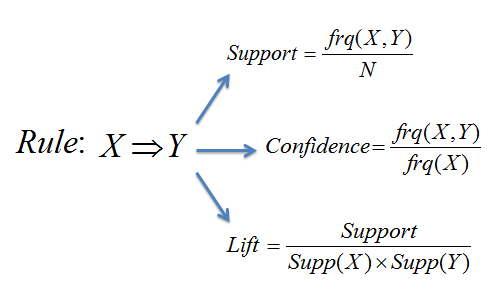
Support refers to how often a given rule appears in the database being mined. For example, how frequently is *{“Bread”, “Butter”}* (rule) is bought together. Support represents the popularity of that product of all the product transactions. Support of the product is calculated as the ratio of the number of transactions includes that product and the total number of transactions.



*N → Total number of transactions*

**Confidence**

Confidence can be interpreted as the likelihood of purchasing both the products X and Y. Confidence is calculated as the number of transactions that include both X and Y divided by the number of transactions includes only product X.

****

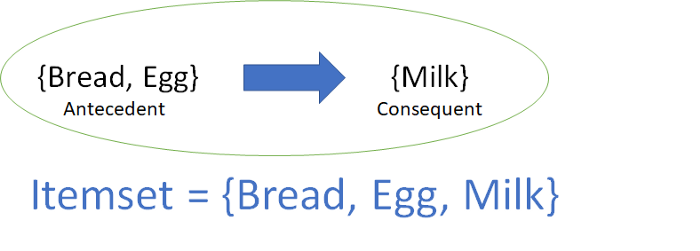
Each association rule is usually associated with two statistics measured from the given data set. The frequency or support of a rule X → Y, denoted frequency (X → Y), is the number (or alternatively the relative frequency) of examples in which X ∪ Y occurs. Its confidence, in turn, is the observed conditional probability P (Y ∣ X) = frequency (X ∪ Y) / frequency (X)

**Antecedents**

An Antecedent is an Item or an Itemset which is found in the data and is supposed to be bought by the customer originally. Within the rules, an antecedent would be bought before any consequent is bought with it.

**Consequents**

A Consequent is an Item or an Itemset which is bought with an Antecedent



An association rule has two parts: antecedent (if) and consequent (then). An antecedent is an item found within the data. A consequent is an item found in combination with the antecedent

**Maximal Itemsets**

A maximal frequent itemset is a frequent itemset for which none of its immediate supersets are frequent.

**Frequent Supersets**

A frequent superset means that it contains more transactions than minimum support threshold

**Frequent Subsets**

A part of an Itemset which contains more transactions than minimum support threshold

**Approach**

The Apriori algorithm is among the first and most popular algorithms for frequent itemset generation (frequent itemsets are then used for association rule mining). However, the runtime of Apriori can be quite large, especially for datasets with a large number of unique items, as the runtime grows exponentially depending on the number of unique items.

In contrast to Apriori, FP-Growth is a frequent pattern generation algorithm that inserts items into a pattern search tree, which allows it to have a linear increase in runtime with respect to the number of unique items or entries.

FP-Max is a variant of FP-Growth, which focuses on obtaining maximal itemsets.

*****“An itemset x is said to maximal if x is frequent and there exists no frequent super-pattern containing x.”*****

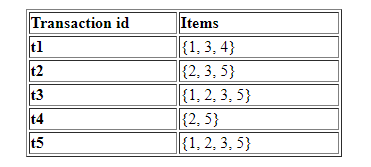
In other words, a frequent pattern X cannot be sub-pattern of larger frequent pattern to qualify for the definition maximal itemset.

We decided to opt for FPMax Algorithm for our Association Rule Mining because of the large dataset and a sheer number of products. As FPMax uses Trie Data Structure for searching the database which is exponentially fast and less hardware extensive.

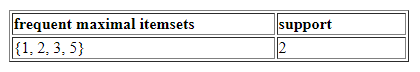
**FPMax**

FPMax was developed as an extension of FP-Growth algorithm to find the MFI’s (Maximal Frequent Itemsets). It initially constructs the Maximal Frequent Itemset Tree (MFI-Tree) to store the MFI’s. Only those frequent itemsets will be inserted into the MFI-Tree that are subsets of itemsets already present in the tree. The algorithm generates the supersets of frequent itemsets and removes the non-maximal frequent itemsets. FPMax is highly scalable and works well with data sets having short average transaction length. However, it spends a lot of time in construction of the MFI-Tree.

A transaction database is a set of transactions. Each transaction is a set of items. For example, consider the following transaction database. It contains 5 transactions (t1, t2, ..., t5) and 5 items (1,2, 3, 4, 5). For example, the first transaction represents the set of items 1, 3 and 4.



Applying **FPMax**on the previous transaction database with a minimum support of 40 % (2 transactions), we get the following result:



**Process**

Data is retrieved from Oracle DB which is parameterized for Company Code and Year. Once we have the relevant Data, we extract all the Part Numbers which are classified as Finished Goods. Once we have all the Parts, We hard code minimum support levels from 0.9 to 0.01.

Some important details about the Processing which is done in Python is undersigned:

1. For every iteration of Company Code, each Year is looped,
2. For every iteration of Year, each Part is looped,
3. For every iteration of Part, each Minimum Support is looped,
4. For every iteration of Minimum Support, FPMax is applied and Association Rules are extracted,
5. If number of Rules < 5, Step 3 is repeated.
6. For every Part, Association Rules are formed and are uploaded to Oracle DB after every iteration for Part,
7. After the completion of Step 2, a .csv file is exported as well in the folder ‘Output’ by the name of ‘V2\_FPMax\_Rules\_for\_CompanyCode\_{ }\_Year\_{ }’.
8. All the Parts without any rules or any exception is stored in two different lists namely
9. part\_exception: for all parts for which, an exception occurred during the runtime.
10. part\_no\_rules: for all parts for which, there were no rules extracted due to less support or less confidence or due to not enough rules being formed.

All the Data preparation and processing is done in Python and the script is end to end, means, it extracts the relevant data and gives the output without any user interference.

*Fig. – Illustration of all the loops in the whole procedure.*

Data Extraction

Extracting Parts

Extracting Unique PO for every Part

Iterating for all possible and plausible Support Values

Applying FPMax for all Unique PO’s for the specific Part keeping the confidence threshold as 0.1

Extracting Association Rules

Manipulating the Table, making it apt for Oracle DB insertion

Oracle DB insertion for each Part

Exporting .csv before the end of Script

*Fig. – Process Tree for every part.*