Introduction:

Market Basket analysis is a data mining method focusing on discovering purchase patterns of the customers by extracting association or co-occurrences from a store's transactional data. For example, when the person checkout items in a supermarket all the details about their purchase goes into the transaction database. Later, this huge data of many customers are analyzed to determine the purchasing pattern of customers. Also decisions like which item to stock more, cross selling, up selling, store shelf arrangement are determined.

Association rule mining (ARM) identifies the association or relationship between a large set of data items and forms the base for market basket analysis. Association rule mining has been widely used in various industries besides supermarkets, such as mail order, telemarketing production, fraud detection of credit card and e-commerce.

One of the challenges for companies that have invested heavily in customer data collection is how to extract important information from their vast customer databases and product feature databases, in order to gain competitive advantage. Market basket analysis has been intensively used in many companies as a means to discover product associations.

A retailer must know the needs of customers and adapt to them. Market basket analysis is one possible way to find out which items can be put together

Import Libraries

For market basket analysis I'm going to use mlxtend. For other purposes (reading data, working with data, visualizing data) I'll use all well-known libraries like numpy, pandas etc.

```
In [1]:
import numpy as np
import pandas as pd
import squarify
import matplotlib.pyplot as plt

# for market basket analysis
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEnc
```

Algorithms Used in Market Basket Analysis

There are multiple data mining techniques and algorithms used in Market Basket Analysis. One of the important objectives is "*to predict the probability of items that are being bought together by customers.*"

- **Apriori Algorithm**
- **AIS**
- **SETM Algorithm**
- **FP Growth**

1. Apriori Algorithm

Apriori Algorithm is a widely-used and well-known Association Rule algorithm and is a popular algorithm used in market basket analysis. It is also considered accurate and overtop AIS and SETM algorithms. It helps to find frequent itemsets in transactions and identifies association rules between these items. The limitation of the Apriori Algorithm is *frequent itemset generation*. It needs to scan the database many times, leading to increased time and reduced performance as a computationally costly step because of a large dataset. It uses the concepts of Confidence and Support.

AIS Algorithm

The AIS algorithm creates multiple passes on the entire database or transactional data. During every pass, it scans all transactions. As you can see, in the first pass, it counts the support of separate items and determines then which of them are frequent in the database. Huge itemsets of every pass are enlarged to generate candidate itemsets. After each scanning of a transaction, the common itemsets between the itemsets of the previous pass and the items of this transaction are determined. This algorithm was the first published algorithm which is developed to generate all large itemsets in a transactional database. It focused on the enhancement of databases with the necessary performance to process decision support. This technique is bounded to only one item in the consequent

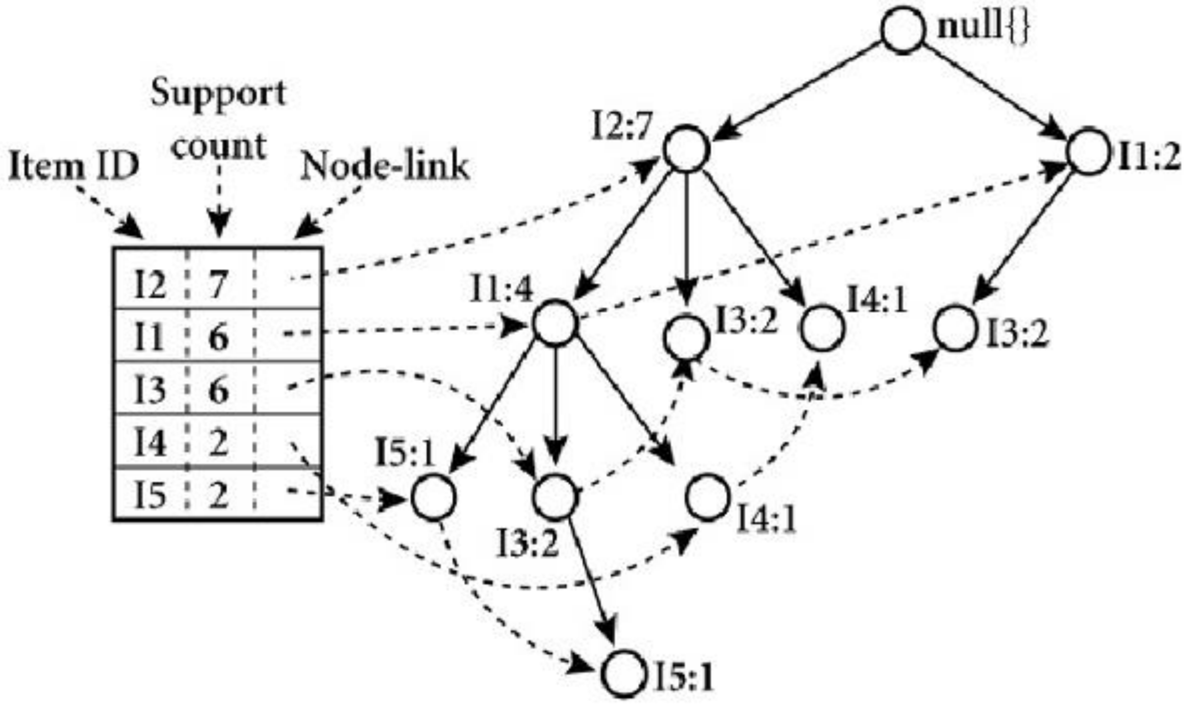
SETM Algorithm

This Algorithm is quite similar to the AIS algorithm. The **SETM** algorithm creates collective passes over the database. As you can see, in the first pass, it counts the support of single items and then determines which of them are frequent in the database. Then, it also generates the candidate itemsets by enlarging large itemsets of the previous pass. In addition to this, the SETM algorithm recalls the TIDs(transaction ids) of the generating transactions with the candidate itemsets

FP Growth

FP Growth is known as Frequent Pattern Growth Algorithm. FP growth algorithm is a concept of representing the data in the form of an FP tree or Frequent Pattern. Hence FP Growth is a method of *Mining Frequent Itemsets*. This algorithm is an advancement to the **Apriori Algorithm**. There is no need for candidate generation to generate a frequent pattern. This frequent pattern tree structure maintains the association between the itemsets.

A Frequent Pattern Tree is a tree structure that is made with the earlier itemsets of the data. The main purpose of the FP tree is to mine the most frequent patterns. Every node of the FP tree represents an item of that itemset. The root node represents the null value, whereas the lower nodes represent the itemsets of the data. The association of these nodes with the lower nodes, that is, between itemsets, is maintained while creating the tree.



Item ID	Support count	Node-link
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	

Advantages of Market Basket Analysis

There are many advantages to implementing Market Basket Analysis in marketing. Market Basket Analysis (MBA) can be applied to data of customers from the point of sale (**PoS**) systems.

It helps retailers in the following ways:

- Increases customer engagement
- Boosts sales and increases RoI
- Improves customer experience
- Optimizes marketing strategies and campaigns
- Helps in demographic data analysis
- Identifies customer behavior and pattern

The Dataset

This dataset contains a total of 7501 transaction records, where every record consists of a list of items sold in just one transaction.

Implementing Market Basket Analysis Using the Apriori Method

The Apriori algorithm is frequently used by data scientists. We are required to import the necessary libraries. Python provides the ***apyori*** as an API that is required to be imported to run the Apriori Algorithm.

Pseudocode

```
//Find all frequent itemset
Apriori(database D of transaction, min_support)
{
F1={frequent 1-itemset}
K=2
While Fk-1≠ Empty Set
Ck=AprioriGeneration (Fk-1)
//Generate candidate item sets.
For each transaction in the database D
 {
Ct=subset (Ck, t)
For each candidate c in Ct
{
Count c++
}
Fk={c in Ck such that countc&gt;min_support}
K++
}
F=U K&gt;Fk
}
//prune the candidate item sets
Apriori generation (Fk-1) {
//Insert into Ck all combination of elements in Fk-1 obtained by self-joining item sets in Fk-1
//Delete all item sets c in Ck such that some (K-1) subset of c is not in Lk-1 }

//find all subsets of candidate contained in t

Subset (Ck, t)
}
```

Conclusion

In this tutorial, we discussed Market Basket Analysis and learned the steps to implement it from scratch using Python. We then implemented Market Basket Analysis using Apriori Algorithm. We also looked into the various uses and advantages of this algorithm and learned that we could also use FP Growth and AIS algorithms to implement Market Basket Analysis.